LINK

A PC Disc, Monitor and Printer for your HP–41, 71 or 75.

Southern
Software

LINK

CONNECTING PC's TO HP HANDHELDS

EduCALC
27953 CABOT RD.
LAGUNA NIGUEL, CA 92677

LICENSE AGREEMENT

By accepting this package you are granted a license to use all of the programs contained herein subject to the following terms:

1 - You have the right to make copies of this software for the sole purpose of backing up your original copy to protect yourself from loss of investment.

2 - This software (or copies of it) may be used on any number of machines by any number of people provided that it is never in use on two different machines at the same time.

3 - You do not have the right to modify the software without the express written consent of Southern Software.

4 - You do not have the right to distribute part or all of this package, including copies of part or all of the documentation, to any other parties without the express written cnsent of Southern Software.

WARRANTY

Southern Software warrants this software and accompanying manual to be free from defects in material and workmanship for a period of 60 days from date of purchase. In the event of notification of defects in material or workmanship within the warranty period, Southern Software will promptly replace the faulty disc or documentation.

The remedy for breach of this warraty shall be limited to replacement and shall not encompass any other damages, including but not limited to loss of profit, special, incidental, consequential, or other similar claims.

This statement shall be construed, interpreted and governed by the laws of the state of Louisiana.

If you find these terms to be unacceptable, return the disc and docmentation for a prompt refund of your purchase price.

OUR PROMISE

These programs have been written to operate on a variety of computer systems and have been tested extensively to minimize the possibility of "bugs". We will make a good faith effort to ensure that the programs perform as stated in the documentation. Should you find one or more "bugs" in the programs, send the disc you are having a problem with to us along with a complete description of the

problem and we will try to determine the source of the problem in an expedient manner. Should we find it, we will return the disc with a corrected version free of charge. Note that this applies only to items stated in the current documentation and cannot be construed to cover program upgrades.

Southern Software
2853 Ramsey Drive
New Orleans
Louisiana 70114

EduCALC
27953 CABOT ROAD
LAGUNA NIGUEL, CA 92677 U.S.A.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

ACKNOWLEDGEMENT

This package would not have been in existence if it had not been for Mr. Ed Keefe of FastAid who was gracious enough on many occasions to pretend to believe the line "I think all of the bugs are out of it this time." and then proceed to discover that they were not.

Chapter 1
USING LINK

Until now, little support existed to connect various HP computers such as series 80 or the HP 41, 71 or 75 with MS-DOS computers such as the IBM PC (and compatibles), HP Vectra, HP Portable/Plus, and HP 150. What support existed was restricted primarily to data communications through RS-232 connectors.

Hewlett Packard developed a system known as the "Hewlett Packard Interface Loop", or simply HP-IL, to allow handheld and portable computers to be able to access printers, disc drives, plotters, etc. via an inexpensive two-wire system. This concept was expanded to other MS-DOS computers primarily to allow the HP Portable/Plus to transfer files to/from IBM PC's and HP 150's. However, the two applications for HP-IL have not been merged.

LINK provides a major step forward in connecting non-MS-DOS computers such as the handhelds into the MS-DOS world and gives the handhelds complete access to all devices which may be connected to the PC. The only requirement for the MS-DOS computer is that it have HP-IL capability. This is built into the Portable/Plus and is available as an add-on card for the HP 150 and the IBM PC family. For less than the cost of a single peripheral for your handheld computer you have access to and control over any device that can be connected to your PC.

LINK provides the following capabilities to HP's family of handheld computers and any other computer with HP-IL capability (such as HP Series 80):

1 - The MS-DOS computer can appear as a disc drive (or digital cassette drive) to the handheld allowing the handheld access to any disc drive on the PC, including hard discs and electronic (i.e. RAM) discs. There is virtually no limit to the amount of data which can be stored on the PC. Best of all, the handheld "sees" the PC as a normal mass storage device and can read/write with no special programming on the handheld. The data are actually stored in the PC's disc drive in MS-DOS format which makes it possible for the PC to share some of the data if necessary (see item 4 below for direct file transfer between the handheld and the PC which is simpler for most applications where the handheld and the PC need to share data). The most significant fact regarding this feature is that it will no longer be necessary for handheld owners to invest in disc drives for the PC as well as the handheld. Even for those handheld owners who have disc

drives, this gives access to much larger and faster drives than are currently available for the handhelds, and at virtually no additional cost.

2 - The PC can appear to the handheld as a video display such as the HP or Mountain Computer video interface/monitor combinations provide. An emulation mode is provided which mimics the response of these two video interfaces. As with the disc drive emulator above, the handheld "sees" the PC not as a PC but as a video display.

Enhancements have been added to the video interfaces that LINK emulates. You can obtain the cursor position at any time and you can make any character blink. Also, if you are using an IBM PC (or compatible) you have easy control over screen color, both foreground and background.

In addition, if you have a 71 or 75 with the KEYBOARD LEX file, you can use the PC keyboard as a remote input device and enjoy the advantages of a full-sized keyboard. By setting the PC as both the DISPLAY IS and KEYBOARD IS devices you can not only use the keyboard but you can see what you are typing as well.

3 - The PC can appear as a printer. Any data sent to it from the handheld will be routed to the PC's printer. This can include text or graphics data.

4 - The PC can appear as an "interface" device to the handheld that can be written to and/or read from. When LINK comes up on the PC, you tell it what device on the PC you wish to access and all input/output from the handheld is directed to this device. Thus, you have access to any device which can be connected to the PC including (but not limited to) the following:

a) RS-232 (i.e. serial) communications ports. These may in turn be connected to modems, plotters, printers, mainframe computers, etc.

b) Paralllel printer ports (i.e. LPT1, LPT2 or LPT3)

c) Modems

d) Console (i.e. the PC screen and keyboard)

e) Graphics tablets and digitizers

f) Mice

g) Joysticks

h) PC clock

i) Files. MS-DOS version 2.0 and later treats files and devices similarly. When LINK asks you for a device name you can give it a filename (on any drive and in

any subdirectory) and the handheld can read/write directly to it.

Any other devices supported by MS-DOS can be accessed by the handheld. This gives the handheld a tremendously wide range of peripherals which can be accessed at virtually no additional cost. Full support is given for control strings to/from the devices (for example, you can power up an RS-232 port, set the baud rate and other communication parameters and then dial a phone number). Block mode devices are supported.

The type of device that the PC is emulating is chosen by the user when LINK first starts. During program execution the device type can be changed by the user at the PC or even from the handheld.

Extensive error handling has been included so that the program will not "crash" but rather will react in a manner similar to current devices available for HP-IL systems. For example, when emulating a disc drive the PC will return the same type of error messages as the digital cassette drive or the 9114 disc drive when the disc has been removed, the end of the disc has been reached, etc. These are fully described later.

It is anticipated that LINK will open up a number of new horizons for handheld computing applications due to the fact that a whole group of peripheral devices are now available to the handhelds. For those who desire the technical details of the response of the PC in the various emulation modes (i.e. response to individual HP-IL messages) please refer to Appendix A of the documentation.

Please refer to the file on your disc named README for the latest information regarding LINK. If you prefer, set your printer to top of form and use the TYPE or PRINT commands to route the file listing to your printer. For example, if README is on your C: drive you can enter:

```
TYPE C:README >PRN
```

or

```
PRINT C:README
```

NOTES

Throughout this manual all entries made to the PC or to the handhelds will be shown in uppercase letters but unless otherwise noted they can be in lowercase also (i.e. "PRINT" is equivalent to "print").

When instructed to "enter" something it means to key in the item and then press "Return" (on some PC's this will be the "Enter" key).

If you are instructed to "press" a key then you do not press "Return" but rather only press the key itself.

Also many times inputs will be shown for quotes to separate it from the neighboring text. Do not enter the quotes unless instructed to do so.

A thorough knowledge of MS-DOS or PC-DOS is not required to use LINK but the user is expected to understand how to start programs, copy files, etc. If you are new to DOS please refer to your DOS manual prior to continuing with this manual.

GETTING STARTED

There are several versions of LINK on your disc to accomodate the various types of computers upon which the program will be run:

1 - IBM PC (and compatibles): To use LINK you must first know something about the Portable/Desktop Link Card you have in your PC. In the installation documentation for that product is a discussion of setting the address of the card so the PC will know where to "find" it. This address is preset by HP to be 1700 but it can be set to whatever you want. The majority of cards have been installed at this address and a version of LINK is on the disc which uses this address named DOSLINK.COM. If this is the address you are using for your card, copy DOSLINK.COM to your working disc, renaming it to LINK.COM when you copy it.

   If the address of your card has been set to something other than 1700 you will copy the file named GENLINK.COM to your working disc but first you must run the program named ADDRESS which will ask you for the address of the card (in hexadecimal) and then it will modify GENLINK accordingly. After you run ADDRESS you can copy GENLINK to any disc, renaming it to LINK.COM in the process. It functions the same as the other versions of LINK. Should you ever change the address of the HP-IL card go back to your master disc and repeat the process of running ADDRESS etc.

2 - HP 150: For this computer you should use your normal INSTALL utility to copy the LINK program to your working disc and get it into PAM. Alternatively, LINK.COM on the root directory is the version used for the 150 and you can copy it to any other disc you want using the DOS COPY command.

3 - HP Portable/Plus: Insert the disc into your C: drive and go to your main PAM screen. A choice will appear for "Install LINK" which you should select by moving the cursor to it and pressing function key "f1" or Return. You will then be asked two questions, a)where you want to install LINK and b)do you want

a REMOVE utility also. If you want to copy LINK to your A: drive, enter A: to the first question and it will be installed on your electronic disc and a new PAM listing will appear on your main PAM screen which you can use to run LINK. If you want to copy LINK to another disc enter C: at the prompt and it will be copied to another disc (you will be prompted to change discs first).

The question of the REMOVE utility is for whether you want an extra file copied to your specified disc such that a new entry for "Remove LINK" will appear in your PAM screen in addition to the one to run LINK. The purpose of the Remove utility is that if you ever want to remove the program from the disc it will be deleted and the PAM entries will also be removed from the PAM screen. We recommend that you do not use this utility if you install LINK on your electronic disc because with limited memory you waste over 2k for the REMOVE utility and it clutters up your PAM screen. You can accept the option by pressing "Y" or refuse it by pressing "N".

For those of you who want to use the LINK program without going through the INSTALL routine, you can find it in the HP110 subdirectory as the file LINK.COM.

RUNNING LINK

Introduction

Prior to discussing the program itself, we should consider what the PC will be when LINK is running. The HP-IL is a loop-type system with all of the devices in the loop controlled by one controller which may be a PC, a handheld, etc. An important point to remember is that there can be only one controller at a time in the loop. The controller determines which devices send and receive data at any point in time and also it sets up the devices' actions when they receive data. The controller can also send and receive data just like all of the other devices in the loop. For example, the controller may tell a disc drive to go to a certain spot on a disc, read some data from the disc and send it on the loop back to the controller or some other device.

Most computers that have HP-IL capability have the ability to act as the loop controller. Some computers such as the HP-41 must be controller in the loop. The LINK program sets up your PC to act as a device on the interface loop. When you run LINK you have several choices as to what type of device you want the PC to act like (i.e.

emulate). Some other computer such as a handheld can then "tell" the PC what to do, such as send and/or receive data. To start the program enter LINK (or if you have an HP computer with PAM point to it and press "f1"). A main screen will appear giving you several options as choices for what you want the PC to do when it is a device in the HP-IL loop. When you exit LINK the PC will no longer be a device on the loop and is able to control the loop if the old controller is taken out of the loop or it is reconfigured as a device instead of controller. Appendix A gives a detailed listing of the PC responses as a device. For most users, this degree of knowledge is unnecessary, especially if the user only uses the device as an emulator of existing HP-IL devices such as disc drives, video displays, etc.

The way this system works is that the controller typically assigns addresses to all of the devices in the loop, typically starting with one. If there are five devices in the loop, the first one is given an address of 1, the second is 2, and so on. Each device remembers its address and responds to the controller's commands. For example, the controller may tell the device at address 2 to become a talker and then send some data on the loop. Each device checks its assigned address and if it matches it makes itself a talker and gets ready to send data (there can be only 1 talker at a time so it is the controller's responsibility to ensure that there is only one device with an address of 2). When the controller issues a command to send data, the talker (in this case the device at address 2) begins sending data. The type of data it sends and how much it sends can be set by the controller. These data are received by all listeners (which the controller sets up and can include the controller).

A detailed knowledge of HP-IL is not required to use LINK. A discussion of the full HP-IL system is beyond the scope of this manual. However if you desire more detailed information, please refer to the following:

"The HP-IL System: An Introductory Guide to the Hewlett Packard Interface Loop" Kane, Harper, and Ushijima, Osborne/McGraw-Hill, 1982

"The HP-IL Interface Specification", Hewlett Packard, 1982

Program Execution

Please note that in all of the discussions below the term "handheld" will be used to designate the loop controller but it actually can be any controller such as another PC or an HP Series 80 computer, etc.

Also, the discussions below are general in nature and cover the actions from the PC side mainly. Please refer to the appropriate chapter for discussions of the uses of a specific handheld with LINK. Examples are given there for copying files, using the video display, etc.

You can start LINK from any drive and any subdirectory. You can even start it from a drive on HP-IL since the PC will not become an HP-IL device until after LINK starts. It does not make any difference to the IBM PC versions whether or not you used HPIL.SYS in the configuration boot file. LINK will ultimately leave the HP-IL circuitry in essentially the same state when it ends as it had when it started.

One important note for those of you who run LINK from a drive on HP-IL - Do not put your handheld in the HP-IL loop until after you load LINK (i.e. when the first menu appears). If the HP-41 or 75 are in the loop you will not even be able to use the HP-IL disc drive since they will be competing with the PC to be the "loop controller". If you have an HP-71 your PC will turn it into a device (which you probaly do not want) and then read LINK from the disc. If you have a 71 and this happens to you, wait until after you have chosen the desired mode in LINK and then enter the command: CONTROL ON to fix it.

This caution applies even to HP Portable/Plus users who start LINK from PAM with a 71. Before PAM transfers control of the PC to any application it will first check out the devices on the loop which will also force the 71 to be a device rather than the controller.

To start LINK, type the name of the drive that has LINK in it (if it is different from the one you are currently in) followed by a colon and "LINK". For example, if you have LINK.COM on your C: drive and you are presently in A: then you would enter the following line:

```
C:LINK
```

The following menu will appear:

```
Choose device to have the [your computer] computer emulate:
        D - Disc drive (such as 9114A or 82161A Digital Cassette Drive)
        P - Printer (uses the PC printer)
        V - Video display
        M - MS-DOS device
        E - End program
```

From this menu you press the key corresponding to your desired choice. Pressing "E" ends the program and returns you to DOS. The other choices are described below. Following the discussion below two techniques are presented which enable you to bypass the menus: 1) by entering the necessary information from the command line when you start LINK, and 2) by commands from the handheld.

For whatever mode you have chosen for LINK you will be informed on the screen that to return to the main menu you must press the F10 key (SELECT on the HP computers) and then you can either exit or

choose another mode. When you change modes in this manner your loop address remains the same.

D - Selection of this option causes your PC to emulate an HP disc drive. Data read from or written to the PC will actually be held in a file named HPILDISC in MS-DOS but this file has all of the HP-type data in it in HP format including the directory, all files, etc. Please note the following enhancements which have been made to the HPILDISC file which are not available using HP-type drives (note that these are totally transparent to the computer reading/writing the HPILDISC file):

1) the size of the HPILDISC file is imited only by the PC's memory up to a bit under 8.4 megabytes. The handheld can use any/all of this available space and if it is not used it is released when LINK ends to avoid wasting disc space when you go back to MS-DOS operation with the PC.

2) there can only be one HPILDISC file in any directory on any drive but you can have an HPILDISC on as many disc drives as you have connected to the PC and one in each subdirectory. Whereas each HPILDISC is limited to 8.4 megabytes, by having more than one HPILDISC on a hard disc, you can effectively use all of the hard disc for HPILDISC storage. Thus you are not limited to any particular amount of HP-IL data you can store on your PC.

3) input and output are buffered by the PC to/from HPILDISC to minimize actual disc access and increase speed. The digital cassette drive requires almost 4 minutes to format a tape and the HP9114 disc drive takes well over a minute; LINK does it in roughly 2 seconds. Because of the buffers though, what is on the disc at any point in time is not necessarily what has been written (some new data may be in the buffers). Therefore it is important that you DO NOT REMOVE the disc from the drive while in this mode or your HPILDISC file will be severely messed up and you will probably lose some or all of the data on it.

NOTE - The file named HPILDISC mentioned above is only one file as far as the PC is concerned but for the handheld it contains all of the files you would normally put on a disc or tape in a conventional HP-IL mass storage device. For example, one HPILDISC file could contain 50 programs, 15 data files, etc. The only reasons for allowing different HPILDISCS in different MS-DOS subdirectories are a) space (e.g. if you used the 8.4 megabytes max. for one HPILDISC

but your 30 meg hard disc had more room) or b) file separation (i.e. the same reason you use different discs or cassette tapes).

The important thing to remember is that HPILDISC is one DOS file that can hold a large number of handheld files.

From a technical standpoint, the abovementioned HPILDISC file does not necessarily have to hold HP-type files. It can hold anything, even another MS-DOS "disc" since it appears to all outside devices as a disc drive and not a file.

After you select the disc drive emulator you will asked for the path to use to find HPILDISC. This can be any valid MS-DOS pathname with one exception, you cannot specify a disc drive that is on the HP-IL because the PC must "control" reads/writes on the physical disc drive (i.e. the one that contains HPILDISC) and the PC and the handheld cannot both control the loop at the same time. Some built-in error checking is done. For example, LINK checks to ensure that the specified disc drive and directory exist before proceeding.

You will be informed whether HPILDISC exists in the specified directory and if so how much memory is currently in use. Also you will be told how much total available disc space remains for the handheld to use. From this point until you exit the program (discussed below) the screen will be frozen with the abovementioned data on it.

If HPILDISC does not currently exist, your handheld must "format" it first before it can be used just like when using the digital cassette drive or 9114. For example NEWM on the HP-41 and INITIALIZE on the HP-71 or 75 perform the "format" function. Refer to the documentation for your handheld to see how to use the new "disc drive". You can copy files, read/write data, etc. to HPILDISC just as you would to a normal HP-IL disc drive.

A technique you can use to change discs is to exit the current program run by pressing "f10" (or "Select" on HP PC's) to have a new device prompted for (this will keep the PC with the same loop address as HPILDISC had). This allows you to switch discs, change subdirectories etc. which is analogous to changing discs in a "normal" disc drive. If you want, you can even change to any other device such a printer, display, etc.

Special notes to HP Portable/Plus users -

1) DO NOT let the computer turn itself off while in any of these device modes since no data can pass through the PC when it is off which causes a "Broken Loop" error. This can be accomplished by setting the display timeout value to a number greater than the time you will be in LINK if you are using your batteries ("infinite" is safe) or keep the AC adaptor plugged in.

2) Since the B: drive is in ROM, the PC cannot make a file named HPILDISC on it and selection of this will cause the program to terminate with an error message.

3) The Portable and Plus are configured to utilize external discs (in MS-DOS format) on HP-IL. If you try to use one of these through this disc drive option, a problem would exist since the PC must control the loop to read from the drive but in the LINK program the PC is a device, not a controller. The program will reject any specification for a drive B: or greater. This is no great loss since the handheld can directly use the drive without going through the PC anyway.

4) You can start LINK from an HP-IL drive (such as C:) since the PC is still a controller until LINK gets underway. After it starts then you cannot use the MS-DOS disc in the HP-IL drive but you can put an HP-type in it for the handheld to use. Just remember to put your handheld in the loop after you have started LINK.

P - Selection of this option makes the PC act like a printer. All data "printed" to it on the interface loop is sent to the PC printer (technically the PRN device). One note of caution - the print device must not be on the HP-IL for the same reason that the disc file HPILDISC above cannot be on HP-IL. This is no great loss since if the printer is on HP-IL, the handheld can use it directly and does not need to go through the PC.

LINK uses the PRN device and you can redirect that on the IBM PC family. Use the DOS MODE command to make it either COM1 or COM2. If you want to use either LPT2 or LPT3 you will have to get more sophisticated and use (or write) some software to change the BIOS address of the parallel printer cards. It is easier to use the MS-DOS device emulator described below to access LPT2 or LPT3.

Please note that how the printer responds to data sent to it through the PC is totally a function of the printer. For example, if you have a program for the HP-41 that was written to print to one of HP's 24-column thermal printers, it may not do what you expect when it is sent to an Epson printer connected to the PC

especially if anything other than basic printing is done (graphics, bold, underline, etc.) since the printers require different control codes to do these "special" things. You should refer to the owner's manual for your PC printer if this is a concern and be prepared to adjust your handheld program accordingly.

You may be asking why there is a printer option if you can get data to the printer through the option using DOS devices? Essentially they both do the same thing and no special control modes are provided by LINK since there are so many different types of printers which can be connected. But there is one important difference, the accessory ID. The MS-DOS device options have an ID of 68 (as an interface device) but in the printer mode described above it is 36. This does not sound like much of a difference until you consider that all the handhelds look for an HP-IL device with an ID from 32-47 as a printer and normal printing functions can go to it. To get output to go to a device with an ID of 68 requires more effort and may not be compatible with existing programs which have been written for the handhelds.

V - This option is for video display emulation. In the default state, it emulates the HP82163A Video Interface but with 80 columns instead of 32. This is provided primarily for compatibility with existing programs which have been developed for the handhelds but several new features have been added.

Characters having ASCII numbers greater than 127 are displayed as the inverse video characters of the character number less 128. For example to display the inverse of character "A" (ASCII character 65) you would send the ASCII character 65 + 128 = 193. Note that if your PC has a color monitor you will not have true inverse video but your text will be a different color (white vs. the normal yellow).

The PC screen also will respond to various escape sequences to allow control of such things as cursor position, clearing, etc. just like the video interface it emulates. These are given below. Note that "ESC" refers to the escape character (ASCII 27). The top left corner of the screen is row 0, column 0.

ESC A - Cursor moves up one row stopping at row 0.

ESC B - Cursor moves down one row stopping at row 15 on the HP Portable and row 24 on all others.

ESC C - Cursor moves right one column. When it gets to column 79, it goes to column 0 of the next row. When it gets to the bottom right corner it will then go home (row 0, column 0).

ESC D - Cursor moves left one column. If it is in column 0, it moves to column 79 of the previous row. It stops in row 0, column 0.

ESC E - Screen is cleared, and cursor goes home. Color goes to initial setting and blink mode is cleared.

ESC H - Homes cursor (row 0, column 0).

ESC J - Clears screen from cursor position down to the end of the screen.

ESC Q - Sets an underscore cursor.

ESC R - Sets a box cursor (default).

ESC S - Screen rolls up one line.

ESC T - Screen rolls down one line.

ESC < - Turns cursor off but it still functions normally.

ESC > - Turns cursor back on.

ESC % X Y - Sets cursor position. X is a character whose ASCII number determines the column (0-79) and Y is a character whose ASCII number determines the row (0-24 or 15 for HP Portable). Range protection is provided by using X mod 80 and Y mod 25 (Y mod 16 for the HP Portable).

NOTE - The spaces above (such as the one between ESC and A) are NOT to be sent. They are included above only for ease of reading.

A feature which is implemented in LINK which is not in the video interface which LINK emulates is the ability to change colors on the screen. This feature is only available on the IBM PC versions and is ignored on HP computers. You also have the capability to generate blinking text and this feature is available on all versions. Color and blink can be set at any time. On all machines, blink and inverse video can be "on" at the same time.

The escape sequences for setting color and blink are similar to those given above except they use lowercase letters after the ESC character:

ESC a - Black (i.e. invisible)

ESC b - Blue

ESC c - Green

ESC d - Cyan

ESC e - Red

ESC f - Magenta

ESC g - Brown

ESC h - Light Grey

ESC i - Dark Grey

ESC j - Light Blue

ESC k - Light Green

ESC l - Light Cyan

ESC m - Light Red

ESC n - Light Magenta

ESC o - Yellow (default)

ESC p - White

Depending upon your monitor and video adapter for your PC, you may not get the colors on the right (they may look like their "dark" equivalents) or you may get shades of grey, or nothing. The HP computers ignore these escape sequences so there is no danger of having an application use them even if it may run using several different PC's.

Background color is set as follows:

ESC q - Black (initial)	ESC u - Red
ESC r - Blue	ESC v - Magenta
ESC s - Green	ESC w - Brown
ESC t - Cyan	ESC x - Light Grey

Note that the background color is only in effect from the point where you change it. To change the entire screen, go to row 0, column 0 and then clear the screen with ESC J. The drawback is that any data on the screen are erased (i.e. set your desired background color first then output). This sequence would be as follows to set a brown background:

```
ESC w ESC H ESC J
```

The IBM monochrome monitor obviously does not have color but it will not give an error for color setings either. The right column colors above are a bit brighter and the only effect color has is that blue and light blue will be underlined. Inverse video on a monochrome monitor is difficult to see but it is there.

To turn blink on, use ESC y; use ESC z to turn it off.

NOTES

- Color and blink settings remain in effect for all subsequent characters displayed (even if they are on different lines) until you either change the settings or issue ESC E to clear the screen.
- When you end LINK or go to another device, LINK will reset the screen for you.
- The HP 150 display apparently uses the last character enhancement set on a line for all subsequent characters on that line. If you put some text in inverse blink in line 1, go off to some other line and set normal characters and then come back to line 1 and add some more characters, they will be in inverse blink. As a good programming practice you should turn off any enhancements prior to leaving a line if you plan to possibly come back to it before it is cleared.

A special feature is included in LINK in this video display emulator mode. Since it has been established to allow you to use the PC screen,

why not take advantage of the keyboard facility since it is right there by the screen? LINK keeps the keyboard live and saves whatever you have typed in it so that the handheld can call for it. LINK will save up to 255 characters in its input buffer before it starts ignoring characters you type.

If you have a 71 or 75 with the KEYBOARD IS LEX file then you can use the PC as the keyboard for your handheld. By setting the PC as both the KEYBOARD IS and also as the DISPLAY IS device you will set it up so that whatever you type at the PC will be entered at the handheld and also displayed at the PC. This makes program editing etc. extremely nice since you get to use the PC and bypass all of the inconveniences of the small display and small keys on the handheld.

Appendix B has a discussion of the KEYBOARD IS option and also a listing of what keys are returned when you press certain special keys on the PC that do not return normal characters (for example the cursor keys). The active keyboard in this mode can be turned off with DDT 0 and turned on with DDT 1. Initially it is on.

M - This option provides essentially complete access to all MS-DOS devices in the PC. The choice of devices is simple. A list of typical devices is given and then you are prompted for the name of a device or file (remember that MS-DOS 2.0 or greater treats files and devices similarly). If the device does not exist it is assumed to be a file and is created. If it is a new or existing file the file pointer is set to the beginning of the file. All devices and files are opened as read/write.

NOTES - Writing to or reading from a file as described in this section is done on a byte-by-byte basis. You cannot copy the whole file at once. For example, on the HP-41 you cannot use the functions WRTP, READP, etc. and on the 75 you cannot use COPY. On the 41 you can use OUTA, INA and IND; on the 71 you can use COPY, OUTPUT and ENTER and on the 75 you cannot use anything without the Extended I/O module (you can "print" to the file with the 71 or 75 by making it the PRINTER IS device but you cannot read from it with the basic 75). If you want to store programs, key assignments, etc. you must use the disc drive emulator option discussed previously.

In this mode, the PC does not emulate any particular device but rather acts like an "interface" device (with a device accessory ID = 68 for a device, 69 for an existing file, or 70 for a new file). Please refer to your handheld's owner's manual for information on methods of outputting to these types of devices. Chapter 2 has some examples also.

This type of operation is much more technical than the three "emulator" options described previously and is discussed in detail in

Chapter 2, "Using Devices" for those users needing this feature. Below is a brief of some of the more important features of this option.

Input/output to the file/device is done on a byte-by-byte basis. One simple example of this is if the "CON" device has been chosen. When the handheld requests data to be sent from the device, whatever characters have been typed on the keyboard are sent from the PC (a null string is sent if no keys have been pressed) and when the handheld sends data to the PC it goes to the screen. The only character that is not sent "as-is" for any device is a carriage return from the PC keyboard ("Return" on HP computers and "Enter" on IBM PC's) for which a carriage return/linefeed is sent.

Some MS-DOS devices have the capability of accepting "I/O Control" data. This provision is provided on a read/write basis through the use of an HP-IL command to "Remote Enable" the device, make it a listener or talker, and then send or request data. This capability uses DOS function call 44h and is described in the DOS Technical Reference Manual. An example of this would be where the handheld would send data to turn on a modem and set the baud rate, parity, etc. before actual data transfer. Note that the IBM PC family does not generally use this feature but HP computers often do. Check the manual for your device to see if it accepts IOCTL and what data bytes you have to send or receive. LINK will inform you whether the device is capable of IOCTL but it cannot detemine what data are required.

Another feature provided is the ability to set block data transfers from the device. For example on the HP Portable and HP 150 the RS-232 communications ports act a bit differently than on the HP Portable Plus or IBM PC. If you request data from an RS-232 port on an IBM PC when no data have come into it, the device will return with no characters but on the HP Portable or 150 the PC will "lock up" until a character comes into the port and the only way to get out without getting a character is to reset the machine. LINK automatically handles this for these machines by first checking the device to see if any characters are in the input buffer before requesting any from it (at the expense of speed however). This feature can be set for other devices via a Device Dependent Talker command for those devices that use the same character as HP does for input checking (one byte with a value of BFh).

NOTE - Technically speaking, the "CON" option is the only one in this package that allows I/O Redirection (refer to your DOS manual) and this only for output (i.e. the keyboard is always the input but the data sent to the PC on the HP-IL can be redirected). This feature should not really be necessary since the concept of LINK is to provide you access to all MS-DOS devices and I/O redirection is not necessary since it accomplishes the same thing.

If you are using a file rather than a physical device the action of the PC when it is remote enabled is different. It does not set IOCTL as does the physical device option. With files, you can set the file pointer or read the current position of the pointer.

If the file is Remote Enabled and a listener, it will accept a string of up to 3 characters that will be used to set the file pointer (firtst byte in the file is 0, not 1). This string of characters must end with CR/LF and if more than 3 characters are sent (up to 250) only the last three will be used. If you send the string "EOF" in upper or lowercase then the file pointer will be moved to the end of the file. If you want to set a specific position you must send ASCII characters representing the pointer position in powers of 256 with the most significant one first. Some examples will clarify this:

Top of file - The string can be one ASCII character 0.

To byte position 300 - This is 255 + 45 so the string is two bytes, an ASCII 1 (for the 255) and ASCII 45.

To byte position 2560 - This is 10 multiples of 256 plus 1 since the first one is 0. The string, then, is ASCII 10 followed by ASCII 1.

Once you have set the file pointer, remove the file from being Remote Enabled (i.e. Not Remote Enabled) and then any data you send to it will be written starting at that position. A common mistake is to set the file pointer and then forget to remove it from Remote Enabled so that further data sent to it are not written to the file but rather only move the file pointer.

Alternatively, if you have the file Remote Enabled and ask it to send data it will not send data from the file but will send three characters representing the position of the file pointer. This string is not terminated with CR/LF. It is in the same form as described above for setting the file pointer position.

One other feature is included for files which may be useful. If you have specified an existing DOS file, any data sent to it from the handheld will overwrite any data that are in the file starting at the file pointer position (initially 0). If you are at the end of the file it will be extended. But if you had a file of 100 bytes and you wanted to reuse the file but only with 50 bytes, after you wrote the 50 bytes to it the last 50 would still be in there. To allow for cases such as this where you want to have the file truncated or "cut off" at any point, just set the PC to be Remote Enabled and then send it the string "CUT" (lowercase is okay too). This will cut off the file at that point. If you desire you can then close the file. If you set it back to Local (i.e. Not Remote Enabled) and then write to it, the file will grow from that point.

Other Methods of Starting LINK

I. Command Line Invocation

Once you become familiar with the LINK menus, you may wish to bypass them and go straight to the program. This is possible by using the "command line" approach when you start LINK. To do this you just enter the same choices separated by spaces after the word LINK in the same order that you would respond to the menu prompts. Examples are given below assuming that you are starting LINK which is on the C: drive:

Disc Drive emulator - For this you selected device "D" and then gave it a path where HPILDISC could be found or made. If HPILDISC is on drive A: and is in the root directory you would enter:

```
C:LINK D A:\
```

Printer - This is device choice "P" so it would be:

```
C:LINK P
```

Video Display - This is device choice "V":

```
C:LINK V
```

DOS Device - This is device choice "M" followed by a device/filename. Assume that you have chosen to use the CLOCK device (if your PC has such a device) :

```
C:LINK M CLOCK
```

That is all there is to it! This feature makes it very easy to use and also does well in a batch (.BAT) file.

II. Handheld Control (Advanced topic)

Once LINK is running, you can change devices from the handheld using DDL 31. The data you will send it will be in the same form as described above for the command line invocation. Note that this requires the Extended I/O module for the HP-41 or HP-75 handhelds. An example is given below for the HP-71B to change the PC to a video display assuming that it is currently device number 1 in the HP-IL loop:

```
100 SEND UNT UNL LISTEN 1 SDC DDL 31 MTA DATA 'V' EOL
```

What the above program line did was to send Device Dependent Listener command 31 to the PC (if it is at HP-IL address 1) followed by the data to reset it as a device. In this case we wanted it to be a video display. If we were doing this on the PC we would only have to input the device choice for the display which is "v". Note that the PC will accept characters for this until either 45 characters have been received or CR/LF is received.

CHAPTER 2
USING DEVICES

LINK provides "conventional" HP-IL device emulation of a disc drive, printer, and video display. But it also enables the handheld user complete access to all devices that may be connected to the PC. This chapter covers the potential uses of LINK in this capacity and offers some examples of several uses.

Version 2.0 and greater of MS-DOS allows devices to be connected to the PC and accessed in a relatively simple manner. A wide variety of devices have been made for the PC including serial (i.e. RS-232) ports, parallel (i.e. Centronics) ports, modems, joysticks, light pens, graphics tablets, clocks, etc. Note that LINK only addresses the device itself, not what may be connected to it. For example, with LINK you could output data to a serial port but LINK cannot know what is on the other end of the port. It is up to the user to know whether a printer, plotter, modem, etc. is connected to the port.

One major feature of this MS-DOS system is that files can be treated the same as physical devices. This means that if you specify a filename when LINK wants a device, you can read/write to a file. Data can be ASCII, binary, etc.

SETTING UP THE PC AS A DEVICE

The first step in accessing devices on the PC is to get LINK running as a device. There are three methods of doing this: 1) running LINK and responding to the input menus, 2) running LINK with the required parameters entered on the command line, and 3) sending the appropriate data from the handheld to the PC once LINK is running. Each method is described in detail below.

1 - LINK with menu prompts : In this method, you start LINK in the normal manner by entering "LINK" (with drive specifier if necessary). The program will prompt you for your desired mode of operation to which you press "M" (or "m") to indicate that you wish LINK to access MS-DOS devices. Then you will be prompted for the type of device you wish LINK to access. You merely type in the name of the device or filename (with optional path) and press Return. Note that if you press Return

with no input, LINK will assume COM1 as your input. If the device does not exist, LINK checks to see if a file by the same name exists and if it does it will be used. If it does not, then a new file by that name will be created.

LINK will tell you a few things about your choice. First it will tell you if your specified device is a physical device or a file.

If you have specified a physical device, LINK checks to see if you can perform standard DOS input/output control (i.e IOCTL) to the device (technically DOS function call 44h). LINK will tell you whether IOCTL capability exists. You are probably asking yourself what IOCTL is used for. An example of IOCTL is the serial ports on the HP Portable/Plus. You can use IOCTL to specify baud rate, parity, stop bits, etc. Note that not all devices have IOCTL capability. For example, the serial ports on the IBM PC (i.e COM1 and COM2) do not have IOCTL capability and baud rate etc. must be set through the DOS MODE command or through BIOS calls, both of which are currently beyond the scope of LINK. For these and similar devices you must set the appropriate parameters on your PC prior to executing LINK. However for those devices capable of IOCTL, these can be set by the handheld (more on how later).

If you have specified a file, LINK will tell you whether the file is an existing one or a new one. Note that the file does not have to be in the same directory or even on the same drive as the one where LINK was started from. You can read and write the file at any location. You can determine whether the file exists by the device accessory ID LINK assumes (see below).

2 - LINK with command line input : This method is the same as the one above except that the inputs are entered at the same time as LINK is run and the input menus are skipped. For example, to use the console device "CON" (i.e. the keyboard and screen) you would start LINK with the following line:

```
LINK D CON
```

All of the command line can be upper or lower-case or a combination. LINK will give you the same data regarding the physical device or file as the method described above.

3 - Changing LINK while it is running : If LINK is already running you can control it from the handheld and make it into any of the options supported by LINK. With this technique you can access any devices from the handheld. To do this you must send Device Dependent Listener command 31 to the PC while it is a listener. After that, the PC will accept data bytes (up to 45 of them) until terminated by a carriage return/linefeed sequence. These data bytes will be interpreted just as if they had been keyed into the PC and they must be in the form described above.

NOTE - This method requires the Extended I/O module for either the HP41 or HP75 computers but can be done directly from the HP71.

For example, assume that LINK is running (as anything) and the HP71 is the system controller which wishes to access the COM1 port to use a PC plotter. The following code would accomplish this (in the HP71):

```
1000 FOR I=1 to 32 @ A$=DEVID$(I) @ IF A$="" THEN 1030
1010 IF POS('PO PL TU DO',A$[1,2])<>0 THEN 1040
1020 NEXT I
1030 BEEP @ DISP 'PC not found!' @ END ! Stop if necessary
1040 SEND UNT UNL LISTEN I DDL 31 MTA DATA "M COM1" EOL
```

FINDING THE DEVICE

Now that you have configured the PC as a device you have access to it. Probably the first thing you will do with a handheld program (or from the keyboard) is to locate the device on the loop. This is normally done by one of two methods, 1) finding the device accessory ID (a number), or 2) finding the device ID (a name). Either method will work and each is discussed below. Each method involves having the handheld "ask" each device on the loop for it's ID (starting with the first device in the HP-IL loop) and then stopping when it has found the correct one.

1 - Device Accessory ID: HP has established some standard ranges of device numbers for devices in the loop. For example, mass storage devices such as tape and disc drives must have an accessory ID in the range 16-31, printers from 32-47, and so on. Since in effect the PC is acting as an interface between the handheld and PC devices such as serial ports, modems, etc. LINK sets up accessory ID's in the range for interface devices, which is from 64-79.

   For non-file devices such as serial ports, modems, etc. LINK sets up an accessory ID of 68. If you specify a new file, LINK sets the accessory ID to 69 and if you have specified an existing file, LINK sets the accessory ID to 70. These accessory ID's are unique to the best of our knowledge and should not cause confusion with other interface devices on HP-IL. With these accessory ID's you should be able to locate the PC in the loop although if you have specified a physical device for LINK to access this method will not tell you what device in particular you are accessing.

2 - Device ID : Any device on HP-IL can optionally return a character string when requested by the handheld (or any loop

controller for that matter). Some of the early HP-IL devices such as the 24-column thermal printer and the digital cassette drive sent no characters when requested but most new devices do. For example, the HP9114A Portable Disc Drive returns the string "HP9114A" followed by CR/LF. LINK will send a string that contains two pieces of data which may be of use to the handheld: 1)the first 2 characters indicate which computer is running LINK, and 2)the remaining characters indicate the device or file name. PO is for the HP Portable, PL is for the HP Portable Plus, TU is for the HP 150 Touchscreen (it does not have to have a touchscreen), and DO is for a DOS computer of the IBM PC family. For example if LINK is running on an IBM PC (or compatible) and is set up to access the COM1 device, the string that is returned is "DOCOM1" followed by CR/LF.

One note of caution if you have specified a file as your LINK device: some standard handheld functions for obtaining HP-IL device ID's will cut it off at 8 characters regardless of the number the HP-IL device is trying to send. For example, if you are using an HP-150 to run LINK and access a file named "C:\TEXTFILE\TEST" and you use an HP-71 to ask for the ID you will only get "TUC:\TEX" instead of "TUC:\TEXTFILE\TEST" with a CR/LF. If you need the whole string, you will have to get more sophisticated (using SEND for example) and play some tricks on the loop.

As an example, assume that we want to find an IBM PC in the HP-IL and also find what it is accessing. Below is a segment of an HP-71 program that is looking for a COM2 device to access a printer.

```
2000 A=DEVADDR('%68') ! Look for interface device
2010 IF A<0 THEN DISP 'Device not in loop!' @ BEEP @ END
2020 D$=DEVID$(A) ! Get the ID
2030 IF D$[1,2]<>'DO' THEN DISP 'PC is not IBM!' @ BEEP @ END
2040 IF D$[3]<>'COM2' THEN DISP 'PC not COM2 device' @ BEEP @
     END
      .
      .(Continue with program)
      .
```

This could be done in one statement as follows:

```
2000 IF DEVID$('%68')<>'DOCOM2' THEN DISP 'Device not
     found!' @ END
```

The same thing for the HP-41 (alpha indicator is ^):

```
01 LBL^TEST
02 AUTOIO
03^DOCOM2        Put device name in alpha.
04 FINDID        See if it exists on the loop.
05 X=0?          If not on loop go to routine HALT which ends
06 GTO^HALT      execution with an appropriate error message.
07 SELECT        Make this the primary device for output.
      .
      .                          .(Rest of program. Send data with OUTA fun
      .
```

ACCESSING THE DEVICE

Now that you have found the device you want to make it do something. You have only two choices, sending data to it or receiving data from it. Within each there can be straight data transfers such as to a printer or, if your device supports it, it can be IOCTL.

SENDING DATA

Sending data can be done on any of the handhelds but receiving data cannot be done on the HP-75 without the Extended I/O module. In addition, to set or clear raw mode (normally not done), to set/clear block mode, or to send/receive IOCTL data requires the Extended I/O module for both the 41 and 75.

To send data with the 41, make the PC the primary device and then merely send the desired data using the OUTA function. Continuing with the example above, further assume that you want to print the phrase "SOUTHERN SOFTWARE" on the printer which is connected to the COM2 port of the PC. Continuing with the example:

```
08 CF 17          (send End-of-Lines with each OUTA)
09^SOUTHERN SOFTWARE
10 OUTA           (print the string)
 .
 .
```

On the HP-71 the following would suffice:

```
2010 OUTPUT '%68';'Southern Software'
```

On the HP-75 without the Extended I/O module, all you can do is assign a device code to the PC and then either make it the display or the printer. We recommend that regardless of what device you have the PC interfacing to except as a video display) that you do NOT use the DISPLAY option since everything you type at the keyboard as well as all data DISPlayed is sent to the device, including cursor movements and mistakes, etc. Once you have set it to be the PRINTER IS device then you can merely PRINT to the device. With the Extended I/O module, you have much greater control over sending and receiving data to HP-IL devices. These will not be covered in detail here since that module gives you commands very similar to the 71 and the 71 is covered fairly well here.

Please note that it is not feasible to cover here what your chosen device may do in response to receiving data from a handheld. If you choose the CON device, the data will go to the screen (unless it is redirected by DOS) but different displays on different computers respond to different characters. HP computers (except the Vectra) generally respond to many different "escape sequences" to perform various screen functions and IBM PC-types may respond to some others (if ANSI.SYS is used) or none at all and just display each character.

Various plotters, printers, modems, etc. respond differently. To determine what will be the response of the devices you wish to use, you will have to refer to the Owner's Manuals for each device. LINK only provides you with the ability to get data to/from the device; the device itself determines what it does with the data.

DEVICE STATUS

There are certain errors that will cause LINK to abort, although these are rare. In this case, your handheld will usually give a "Loop Broken" error the next time you attempt to use it. Examples of this are unplugging a printer or plotter you are trying to use, removal of a disc you are writing to, etc. When these types of things occur, DOS on the PC will generally put up a message on the PC screen and ask you if you want to "Abort, Retry or Ignore?" and you can press A, R, or I appropriately (be careful with "I" !). If you press "A", then LINK will end and your handheld cannot communicate with it until you restart LINK. If you are in the middle of writing to a DOS file, you will most probably lose the entire file. Fortunately, these events are rare. The majority of errors are trapped by LINK and reported to you through the use of the HP-IL command "SST" (Send Status).

If you want to get the status from the PC, you must request it. The way LINK works, a 0 status means no error has occurred, and a non-zero value indicates an error. The value of the status byte is the error

number reported by DOS. Refer to Appendix A, "Response to HP-IL Messages" for a table of the various errors. Typical DOS error codes are in the range from 1-18 although DOS 3.0+ has error codes all the way up to 88 but most of the higher-numbered ones are very unlikely to occur in LINK.

On the HP-41 you use the INSTAT function (p. 49 of the HP-IL module manual) which returns the status byte into the x-register and also bit-by-bit into flags 0-7. If you have the Extended I/O module you have the SRQ? (p. 36) and STAT (p. 39) functions available to detect and monitor loop status.

On the HP-71 you can use the the SPOLL function to get the status byte from a device.

NOTE - LINK never generates a service request in device mode. The only situation where LINK generates a service request is in the video emulator mode when the keyboard is live and keys have been pressed on the PC but not yet read by the handheld.

RAW MODE

DOS allows physical devices connected to it to function in either "raw" or ASCII mode. In ASCII mode each byte sent to or received from the device is checked for Ctrl-C, Ctrl-S, Ctrl-P and Ctrl-Z which all mean special things to DOS. If you want to maximize speed of data transfer to the device you do not want this check to occur so you would set the device to "raw" mode. This is the default condition LINK establishes for physical devices (it cannot be set for files).

There may be occasions where you do not want the device to act in raw mode. LINK allows you to set ASCII mode with Device Dependent Listener 0 and set raw mode with DDL 1. Note that LINK will reset the device to whatever it originally was when it terminates.

One possible use of ASCII mode might be where you were using the CON device which is the keyboard/screen combination and you want the Ctrl keys to be active to toggle echo of data to a printer etc. Please refer to your DOS manual for a discussion of possible uses of the Ctrl characters. It is not expected to be a common practice and will not be covered in detail here.

RECEIVING DATA

Some devices such as printers do not have the capability to send data. But others such as plotters, the keyboard, modems, etc. can and often do. For this case you have the capability to receive data from the device into the handheld. What the data mean are totally dependent

upon the device. For example, you could use a PC keyboard to input data into an HP-71 file or the HP-41 alpha register. You could get the pen position from a plotter, incoming data from a modem, etc. There are few limits to what you can do.

Below is a program segment for the HP-71 for getting data from a PC text file until end-of-file is reached (i.e. the PC quits sending data). This data is put into a TEXT file named PCDOWN on the 71.

```
2960 SF -1 @ PURGE PCDOWN @ CF -1 ! Purge it if it exists
2970 CREATE TEXT PCDOWN @ ASSIGN #1 TO PCDOWN ! Open
     the file
2980 A=DEVADDR(':INTRFCE') ! Find the PC
2990 IF A<0 THEN BEEP @ DISP 'PC not in loop.' @ END
3000 DIM A$[255],B$[510] @ A$='' @ B$='' @ SFLAG -23 @ CFLAG 0
3010 ENTER :A ;A$
3020 IF A$='' THEN GOTO 3110
3030 B$=B$&A$
3040 S=SPOLL(A)
3050 IF S THEN BEEP @ DISP 'DOS File READ Error : ';(S-32) @ WAIT
     2 @ GOTO 3110
3060 B=POS(B$,CHR$(13)&CHR$(10))
3070 IF B<256 THEN 3090
3080 PRINT #1;B$ @ BEEP @ DISP 'Line too long' @ WAIT 2 @ GOTO
     3110
3090 IF B=0 THEN GOTO 3010
3100 PRINT #1;B$[0,B-1] @ B$=B$[B+2] @ IF B$<>'' THEN 3060
     ELSE 3010
3110 ASSIGN #1 TO * @ DIM A$,B$ @ CFLAG -23 @ BEEP @ DISP
      'Done'
```

By slightly adjusting the program above you can use it to get text from the PC keyboard and put it into the same file. With the adjustment given below all you have to do is press any key other than the ATTN key to terminate input to the 71 and end the program.

```
3040 IF KEY$<>'' THEN 3110 ELSE IF A$="" THEN 3010
```

Technical comments for HP-71 users- LINK does not add CR/LF to any data being transferred between a DOS file/device and the handheld. Pages 118-126 of the HP-71 HP-IL Interface Manual has a good discussion of the use of flag -23 in determining when the 71 will quit requesting data. Essentially if flag -23 is clear the 71 will keep asking for data until either a CR/LF or LF is received or all variables are full. This can cause the 71 to lock up in some cases with LINK. An example would be where a string variable is dimensioned to a larger number than there are characters left in the file and

there is not a CR/LF sequence left. The 71 will keep asking for more data to be sent but the PC will keep returning a null string because there are no more data to send.

It is for this reason that the program above has flag -23 set. In this mode the 71 will quit requesting data when the variables are full or no more data can be sent (i.e. an ETO is sent). This bypasses the potential problem discussed above but it complicates programming somewhat since you cannot normally just read a line of text from the PC into a 71 variable since the 71 will not quit requesting data when the PC sends the CR/LF. It does prevent unexpected lock-ups though. In general this is not necessary for DOS text files since DOS uses CR/LF to separate lines of text. Two cases you need to be aware of: 1)some word processors do not use CR/LF at the end of each line of text (known as soft carriage returns) when the file has been saved in "document mode", and 2)some text files do not have CR/LF on the last line since Ctrl-Z is used as an end-of-file marker it is not strictly required. This occurs most frequently in batch files (i.e. those with extensions of .BAT). It is for this reason that we recommend that in general you use the method shown in the example above.

Also, to make the routine above "bullet-proof" you might replace the top two lines with some code to find the PC for whatever mode it is in (example presented earlier) and then change it from the 71 program to whatever file you want to download. The program presented here assumes that the PC side has been set up properly already.

Note that when reading from a file, if you reach the end of the file you do not get an error message; you just read 0 bytes.

One other point to discuss on receiving data concerns "block" input of data. Basically, if you instruct a device to fetch some data it has two methods of doing it, a)it can send you however many it has received up to the number you requested, or b)it can put you "on hold" until the number of characters you asked for comes in and then send that many characters. Method (b) is known as block data transfer. The way you can deal with this problem is to first ask how many characters the device has and then only ask for that many. LINK automatically sets up the serial ports of the HP 150 and HP Portable (not the Plus) as block devices, using the character BFh to send to the port which causes the port to send back one character representing how many characters have been received. The IBM PC and the HP Portable Plus do not block data within their serial ports so this is not a concern for them.

Please note that the block mode used within LINK requires the device on the PC to be able to handle IOCTL (see discussion). The character BFh (ASCII 191) is sent via IOCTL and the return number of bytes available is also read via IOCTL. If you use the IBM PC you should not generally need this feature and if you use the standard

serial ports on the HP 150 and Portable this is already set up automatically (unless you change it deliberately).

If you have block mode devices that respond in block mode and also respond to the BFh character as described above, you can set LINK to block mode by sending Device Dependent Talker 1 to it. If a device needs to be taken out of block mode, send it DDT 0. Examples are given below:

HP-41 (requires Ext. I/O module)

```
01 68
02 FINDAID          Look for interface device
03 X=0?
04 GTO^NOFIND       Exit if not found
05 SELECT           Make it the primary device
06 1
07 DEVT             Make it block mode by sending DDT 1 to it
```

HP-71

```
10 A=DEVADDR('%68') @ IF A<0 THEN DISP 'Not found' @ END
20 SEND UNT UNL LISTEN A DDT 1 UNT UNL ! Sends DDT 1 to
       interface device
```

SENDING DATA TO A FILE/DEVICE

Writing to a file or device is even simpler than reading from one. The simplest way is to make the PC the printer (or primary device on the HP-41) and then PRINT to it (or use OUTA on the HP-41). Assume that you have an HP-41 and you want to set it up to take data you will output from the ALPHA register into a file. First set up LINK in DOS device mode with a filename rather than a physical device. The steps below will allow you to OUTA whatever you desire to the file. Note that flag 17 is cleared to provide for CR/LF at the end of the line output. If you have a line of text more than 24 characters long (the APLHA register capacity) just set flag 17 for however many ALPHA registers you want to output and clear it for the last one. The filename on the PC is A1.

```
01 LBL^TOFIL
02 AUTIO
03^DOA1
04 FINDID
05 X=0?
06 GTO^NOFIND
07 SELECT
```

```
08 MANIO
09 CF 17
```

On the HP-71 it is also simple:

```
10 A=DEVADDR('DOA1')
20 IF A<0 THEN BEEP @ DISP 'PC not found' @ END
30 PRINTER IS :A
40 A$=              (whatever you want)
50 PRINT A$
```

It is all very easy! A note of caution though; the handhelds can return only eight characters when requesting the device ID name on the loop. Since LINK uses 2 for the computer name, you can only check for the first 6 characters of the filename. In general the best thing to do is check for Device AID (69 for a new file, 70 for an existing file and 68 for a physical DOS device) or use the technique presented earlier to "restart" LINK from the handheld where you can set the filename yourself.

If you own an HP-71 please refer to p. 187 of the HP-IL Interface Owner's Manual for a discussion of why you cannot use "SEND" to directly obtain a long string of an accessory ID from a device without playing some tricks on the loop.

FILE POINTERS (DOS Files Only)

From time to time you may find it necessary to change the point in a file where you read or write data. LINK gives you this capability using the HP-IL commands for Remote Enable and Not Remote Enable. These will set all devies in the loop to the specified condition. Many devices do not respond to the commands. Note that these commands are available only with the Extended I/O module on the HP-41 and 75.

To determine where you are in the file at any point, set Remote Enable, make the PC a talker then request data. It will send three bytes of data indicating the current file pointer position. Then you can move it to wherever you want to by sending it some data (see below) or you can make it Not Remote Enabled and then read/write normally. Note that the first byte in the file is always at position 0, not 1 as you might expect. The data that come back are in powers of 256. For example, if the ASCII numbers 0 10 1 came in you would interpret them as follows (note the most significant byte is first):

```
0x(256^2) + 10x256 + 1
```

To set the position you must send a similar string to the PC while it is Remote Enabled and a listener. The string you send must end with CR/LF. If you send more than three bytes, only the last three will be used. If you send less than three characters (you have to send at least one) the first ones are assumed to be 0. For example if you send ASCII 1 then ASCII 0 followed by CR/LF the file pointer will be at byte number 255 (remember that the first one is 0). With this technique you can set the file pointer anywhere from 0 to 16,843,008!

Two special character strings are allowed by LINK to help with file handling, both using the Remote Enable command like the file pointer position. If you send the string "EOF" to the PC then it will move the file pointer to the end of the file so that any further writes to the file will be appended. If you send "CUT" to it then the file will be cut off at that point and any data in the file beyond the current file pointer position are deleted (i.e. truncated). CUT is most useful when you are overwriting an existing file and you do not want any of the old data to remain past the point where you have written your new data. Note that both "CUT" and "EOF" can be upper or lowercase or even a combination.

Note that you can place the pointer beyond the end of the file. If you do this and then start writing to it the intervening bytes may contain anything so this is not generally advised.

An example follows for the HP-71 which appends the TEXT file "UPLOAD" to the PC file:

```
100 A=DEVADDR(':INTRFCE') @ IF A<0 THEN BEEP @ END
110 IF DEVAID(A)<69 THEN BEEP @ DISP 'Not a File' @ END
120 REMOTE A @ OUTPUT A ;'EOF' @ LOCAL
130 ASSIGN #1 TO UPLOAD
140 ON ERROR GOTO 160 @ DIM A$[255]
150 READ #1;A$ @ OUTPUT A ;A$ @ GOTO 150
160 ASSIGN #1 TO * @ BEEP @ DISP I-1;'lines uploaded.' @ WAIT 2
170 REMOTE @ SFLAG -23 @ ENTER A;A$ @ LOCAL
180 X=NUM(A$[3])+256*(NUM(A$[2])+256*NUM(A$[1]))
190 DISP 'PC filesize now';X:'bytes' @ DIM A$ @ CFLAG -23 @ END
```

As with any program this one could have features to dimension variables, save and restore flag settings, etc. but it is presented merely to show the ease with shich you can append a file, not to demonstrate detailed programming methods.

For the HP-41 and 75 you can easily write to a PC file and with the 41 you can even read from it. To read with the 75 or to move the file pointer with either machine requires the Extended I/O module. With the basic 41 however you can perform a crude file appending by reading from the file until no more characters come in (which occurs at the end of the DOS file) and then start writing to it. It is pretty slow for long files but it will work.

Also with the Extended I/O module you can upload and download HP-41 program files directly into PC files. Pages 53-55 of the manual describe two functons, INP and OUTP which allow you to send an ASCII equivalent of your program to any HP-IL device which includes LINK acting as a DOS file. This should open the door for some enterprising individuals to devise a program editor for 41 programs on your PC. With a little knowledge of the hexadecimal representations of the 41 functions and a special editor it would not be too difficult.

IOCTL Capabilities (Physical Devices Only)

If your physical device has IOCTL capability you can send/receive IOCTL strings using your handheld using the Remote Enable command. The meaning of IOCTL is Input/Output Control and not all devices within DOS implement this feature. The common IBM PC devices such as parallel and serial ports do not. HP Portable and Portable Plus serial devices do. Usually IOCTL is used to set the operating parameters for the device. It is analagous to the buttons and knobs on the front and side of many devices. LINK allows you to set these from the handheld.

LINK determines whether this capability exists when you select your physical device and displays a message regarding its capabilities. Files do not have IOCTL capability at all.

Please refer to the owner's manuals for your DOS devices to determine if they support IOCTL, the format it should be in, and also what you can do with it. An example is given below for sending IOCTL to a serial port (COM1) on an HP Portable Plus to set it's baud rate, parity, stop bits, etc. The meanings are as follows:

M1 : Turn on the RS-232 port and turn the built-in modem off.
P4 : No parity.
SBE : Set 9600 baud.
SL6 : Observe DSR hardware handshake.
SS0 : Use 1 stop bit.
SW0 : Word length is 8 bits.

Sample for the HP-71 (assume device is at HP-IL address A):

```
100 REMOTE @ OUTPUT A;'M1;SBE;P4;SL6;SS0;SW0;' @ LOCAL
```

Note that the IOCTL string is teminated with a CR/LF. LINK will continue reading characters into a buffer until it gets a CR/LF sequence then it will send this to the device as the IOCTL string. Up to 255 characters will be stored in the buffer. If a device does not have IOCTL capability, this data will be ignored.

Chapter 3
LINK and HP-41

This chapter presents some sample code and offers some references for using LINK with the HP-41 handheld computer family. Those programs which require extended functions or the Extended I/O module will be noted.

NOTE - Anywhere the ALPHA register or an alpha label is used, this documentation will use a caret ("^") to denote it.

- Unless otherwise noted any program can also be executed manually.

OPTION 1 - Operation as a disc drive

References - "HP-IL Module Owner's Manual", pp. 33-40.

- "Extended Functions/Memory Module Owner's Manual", page 33.
- "Extended I/O Module Owner's Manual", pp. 10-18, 30-66.

Sample 1: Format a disc on the PC disc drive. Assume the PC has been set up already as a disc drive.

This is always done manually since there is a chance of overwriting data which may already exist on the drive. The function needed is NEWM which is not programmable. CAUTION - if you have more than one mass storage device on the loop, make sure the PC is the first or that you have used SELECT and MANIO to specify it so that your other drives are not erased.

The NEWM function erases the HPILDISC on the PC if it exists so if you have any anything on that from previous operations that you want to save, copy them off first to another disc or the 41 memory. When LINK first starts it will tell you if HPILDISC exists and if so, how much space it currently occupies. Remember that the size of this DOS file will grow as you write HP files to it.

A question comes up regarding how many files to specify with NEWM since it will prompt you for it. Check how much room on your PC disc is available (LINK displays this on the screen). If it is not enough, exit LINK and erase a few files or change discs. It most

probably will be much more room than you will need though. Remember that you do not waste any space on the PC disc if you do not use it since LINK will return all unused records back to DOS when it is through. If you do not know how many files to specify estimate the average length of your files (1,000 bytes is not too bad) and allocate enough for them.

For example if you have 300,000 bytes left you will have more than the basic 41 can address anyway since without the Extended I/O module and a few tricks you cannot use more than 131,072 bytes anyway. In this case you could probably specify 120 or so and still have plenty of room.

Space for the directory (which holds the filenames, types, sizes, locations on the disc, etc.) is 32 bytes per file and it is in "chunks" of 256 bytes (called records). A directory of 120 files will use 3,840 bytes on your disc. The section referenced in the HP-IL book gives a fairly good overview of all of the types of things you can do with the disc drive. LINK supports all of the functions in that book.

Note that you can copy ASCII files between extended memory (with an Extended Functions/Memory module or an HP-41CX) and the PC using the SAVEAS and GETAS functions.

If you have the Extended I/O module, you can do much more with the 41 and your PC. A couple of examples follow but please note that you can do everything with the LINK disc drive emulator that you can with standard HP-drives.

If you have sufficient room on your PC disc you can copy an entire medium to the HPILDISC file. You can check how much room you have on HPILDISC with Device Dependent Talker command 7 (this is not available with the digital cassette drive but is with the HP9114 disc drive). You can use the MCOPY function (pp.11-12) to duplicate your disc or tape to the PC.

The routine CHECK below will return the loop address of the digital cassette drive in the Y-register and the PC file HPILDISC loop address in the X-register. This program requires the Extended I/O module.

```
01 LBL^CHECK
02 LBL 01
03 1
04 AUTIO
05 SELECT          Make the starting address 1.
06 16
07 FINDAID         Look for a mass storage device.
08 X#0?
09 GTO 02
10^NO DRIVE        Display message then stop.
11 AVIEW
```

```
12 PROMPT
13 GTO 01
14 LBL 02
15 STO 00          Save the address.
16 SELECT
17 SF 25
18 CF 01
19 ID              Get the name of the device.
20 FC?C 25         If it is a cassette drive, it won't have an ID.
21 SF 01           Set if a cassette drive.
22 1
23 +
24 SELECT          Start loop check with next device in the loop.
25 16
26 FINDAID         Look for next mass storage device
27 X#0?
28 GTO 03
29^ONE DRIVE       Display error message and stop.
30 PROMPT
31 GTO 01
32 LBL 03
33 SELECT
34 SF 25
35 ID              Get ID of this device.
36 FS?01
37 GTO 05          If the first one was the digital cassette, go to 5.
38 FS?C 25
39 GTO 04
39^NO CASSETTE          Both drives had an ID, no cassette drive.
40 PROMPT
41 GTO 01
42 LBL 04          We are at the address of the PC drive.
43 RCL 00          Now address of PC in Y, cass. drive in X.
44 END
45 LBL 05          This one should have an ID.
46 FS?C 25
47 GTO 06
48^NO PC DRIVE     Neither one had an ID.
49 PROMPT
50 GTO 01
51 LBL 06
52 RCL 00
53 X<>Y            PC address in Y, drive in X.
54 END
```

The routine below uses CHECK from above and COPYFL to make copies of individual files you may have on your cassette drive to the PC in those situations where you do not want to copy the whole disc.

```
01 LBL^SIMPLE
02 XEQ^CHECK      Same routine from a while back.
03 LBL 01
04 SELECT
05 X<>Y
06^FILE ?
07 AON
08 PROMPT
09 AOFF
10 COPYFL
11 GTO 01
```

To run this one, just keep entering the filenames and then press R/S. When you run out of filenames, do not press R/S but do press ALPHA to get out of ALPHA mode since the program switches you in and out of ALPHA and you stopped it before it got you back out.

NOTE - One of the major advantages of LINK is that you have access to a very large amount of disc space which ranges from 360k to 8.4 megabytes if you use a hard disc on your PC. One of the major drawbacks of the 41 is that it came out when the digital cassette drive was the only mass storage device available other than magnetic cards. Since the cassette drive was limited to 131,072 bytes you will not be able to fully utilize all of the space on a DOS disc (unless you use subdirectories since each subdirectory can contain an HPILDISC file). A utility is available from the HP User's Library (#41-09114) which will give you access to all of the HPILDISC file. If you own a 9114A disc drive a listing of the program is given on pp.12-19 of "Using Your Disc Drive" which accompanies that disc drive.

If you want to read data from a TEXT file on an HP-formatted disc or tape drive (LINK included) you should be aware of the fact that the lines of text are not stored in a normal fashion (i.e. a carriage return/linefeed or CR/LF at the end of each line) by either the 71 or 75 handhelds. They use a technique of using ASCII character 0 as a separator of text lines followed by two characters whose ASCII values indicate the length of the line of text. If you use the 41 to read/write these files to/from your ALPHA register you will get strange results unless you look

for these and adjust accordingly. This requires a lot of programming effort and will not be covered here.

OPTION 2 - Operation as a Printer

References - "HP-IL Module Owner's Manual", pp.9-29.

LINK acceses the PRN device in this mode which is the "standard" DOS printer. When you first boot your PC it is LPT1, your first parallel printer connector. If you use either an HP 150 or an HP Portable/Plus you can easily change this through your system configuration menus to be anything you want (within reason). If you are using an IBM PC you can change it to either the COM1 or COM2 ports via a form of the MODE command but to change it to LPT2 or LPT3 requires more sophistication (you have to change the base addresses of the cards which cannot be done from DOS). Thus, in the printer mode of LINK you can easily access LPT1, COM1, or COM2 and with the appropriate software also LPT2 or LPT3.

LINK allows you to do most of the things discussed regarding printer devices in the HP-IL manual. The only things that cannot be performed are those functions which were specific to the HP82162A Thermal Printer as follows;

Flags 12, 13 for double-wide and underline (pp. 9, 10, 13).

SKPCHR - Accumulating skipped characters (page 18).

FMT - Centering lines (page 19).

All graphics/plotting functions (pp. 20-28)

Flags 21 and 55 concerning printer existence and use as well as flags 15 and 16 regarding manual, normal and trace modes all function properly since they are internal to the 41 and determine what gets sent to the print device and do not have anything to do with the physical printer itself.

One feature regarding printers not really addressed in that book is that the PRA and ACA functions used for printing will only print characters in the ASCII range from 0-127. If you try to use these functions with an ASCII number from 128-255, the character that is 128 less will be set (technically the most significant bit is set to 0). With the thermal printers that originally came out with the 41 family this was not a problem since they did not have a higher ASCII character set anyway and the graphics were done in such a way that these characters were not needed.

However with the wide range of printers which can be connected to PC's there is a much larger need for these characters. Do not despair. There is another way to get printed output with these characters

(several actually). The easiest way to print these characters is to use the OUTA function if your character is all or part of a string in ALPHA or, if you have an Extended I/O Module, you can output individual bytes with the OUTXB function (p. 47).

Getting the character into alpha is difficult and if you don't have an Extended I/O module you have to trick it using the BLDSPEC function in a manner other than that for which it was intended. See the following secton on the video interface for this "poor boy" method. Below is a simple routine to print a string in ALPHA to the printer in lowercase (requires Ext. I/O):

```
01 LBL^LOWER
02 1
03 SELECT
04 32
05 CHS          Make it negative to do a "class" search.
06 FINDAID      Find anything in the printer class (LINK printer
                is 36).
07 X=0?
08 GTO^NOFIND   NOFIND would be your error handler.
09 SELECT
10 LBL 01
11 AON
12^STRING ?
13 PROMPT
14 AOFF
15 ALENGIO
16 X=0?         Quit if a null string is input.
17 END
18 LBL 02
19 ATOXL
20 X<0?
21 GTO 03       If no more characters, jump out.
22 65
23 X>Y?
24 GTO 04
25 CLX
26 96
27 X<Y?
28 GTO 04
29 CLX          We have found a character between 65 and 96 (i.e.
                uppercase)
30 32
31 +            Adding 32 to the ASCII number makes it
                lowercase.
32 LBL 04
```

```
33 ACCHR
34 GTO 02
35 LBL 03
36 PRBUF
37 GTO 01
```

Since you have a PC to run LINK on and you have read through some or all of this manual you have an idea of what the carriage return/linefeed (i.e. CR/LF) is in the world of ASCII. After you go back and read the complex descriptions in the HP-IL book regarding the simple printing functions, note that the functions that begin with "AC" such as ACA (accumulate alpha), ACCHR (accumulate character), and ACX (accumulate x-register) are just like PRA, PRBUF and the rest of the "PR" functions except the "AC" functions do not have CR/LF sent with them and "PR" functions do. In fact, the HP-IL does not have a true "print buffer"; it was part of the thermal printers. So what does PRBUF do if there is no print buffer and no thermal printer? It just prints CR/LF.

How your printer responds to character sent to it is a function of the printer. Many wait until they get a whole line of text into their buffers before printing but not all do so do not think that something is wrong when you execute PRBUF and nothing happens or if you use ACA and the characters print immediately. The 41 sends these characters immediately and LINK passes them on. Check your printer's manual to see how it is supposed to act before you assume your system has a problem. Especially be aware that strange things may happen if you use any of the functions from the table above on a non-HP thermal printer system.

OPTION 3 - Video Display

References - "HP-IL Module Owner's Manual", pp. 9-30.
- "Extended I/O Module Owner's Manual", pp. 19-53.

In principle, accessing the video screen is the same as using a printer. The HP-41 treats them the same as far as devices go. One point to remember when you have two or more printers and/or displays in the loop is that the 41 will normally access only the first one in the loop. If you want to use one of the others you must find its address (or actually just 1 greater than the first printer or display) and use SELECT to make it the primary device.

LINK gives you some features not available from printers or from the other video interfaces on the market: 1) you can find out where the cursor is at any point in time, 2) you can make any character blink, and 3) you can set colors if you are using an IBM PC (or

compatible). These are fully discussed in the preceeding documentation.

A feature which is compatible with other video interfaces is the ability to use "inverse video". Normal characters are displayed with light color on a dark background (LCD displays such as the HP Portable or Portable Plus are reversed) and inverse video is dark characters on a light background. Most monitors will only display the block that the character is in in the inverse and not the whole screen or line.

Following is a routine that displays "Cat" in inverse video in the middle of line 10 of the screen. We have covered several times how to find and select an HP-IL device so we will assume that it is the SELECTed device.

```
01 LBL^INV
02 27          Escape character
03 ACCHR       Send it to the video interface
04 37          %
05 ACCHR
06 39          Column 39
07 ACCHR
08 10          Row 10
09 ACCHR
10 CLA         Clear alpha so we can "build" our string.
11 67          C
12 XEQ 01
13 97          a
14 XEQ 01
15 116         t
16 XEQ 01
17 OUTA        Send the string to the video interface
18 END
19 LBL 01
20 STO 00      Save the number
21 CLX         Clear x-register
22 1
23 BLDSPEC     Sets most significant byte to "1".
24 RCL 00      Get back our character code.
25 BLDSPEC     Build the character (ASCII number + 128)
26 ARCL X      Tack it onto the ALPHA register
27 RTN
```

With the extended functions XTOA and ALENG you can easily expand this routine to handle any string in alpha being displayed anywhere on the screen. Note that since we are outputting characters with ASCII numbers greater than 127 we must use OUTA rather

than PRA and the setting of flag 17 will determine where the cursor ends up. If flag 17 is clear, a CR/LF is sent so the cursor will be in column 0 of row 11 after this display but if flag 17 is set, CR/LF is not sent so the cursor will be in column 42 of row 10, just after the three characters displayed.

OPTION 4 - Operation as a Device

This is the most difficult subject for which to give examples merely because of the wide variety of devices which can be connected to a PC. Due to the relatively limited memory capacity of the 41, data transfer to/from the handheld will be somewhat limited. However, some probable uses include:

1 - Transfer of field-gathered data (numeric and/or string) to a text file on the PC for further processing and even as input to spreadsheets.

2 - Driving printers and/or plotters. If you have a printer connected to LPT2 or LPT3 on your IBM PC it is not very easy to make it the PRN device so this is an easy way of outputting to that printer. If it is hooked up to one of the COM ports, you can effectively change it to PRN using one of the forms of the MODE command so it is not quite so critical there.

3 - Upload of HP-41 programs for editing on the PC followed by downloading to the 41 for execution using OUTP and INP of the Extended I/O module.

4 - Use of the 41 as a time manager using both time and extended functions for appointments, diary, expense records, etc. with the data sent to the PC for archival.

Below is a program that uploads data from ten registers to the PC where the PC has already been made the primary device through SELECT:

```
01 LBL^UPLOAD
02^FIRST REG ?
03 PROMPT
04^LAST REG ?
05 PROMPT
06 1000
07 /
08 +          X-register now has the loop counter for the
                 registers.
09 CF 17      This will make each line end with a CR/LF
10 LBL 01
```

```
11 CLA
12 ARCL X        Have to put it in ALPHA since it is sent as
                       ASCII data.
13 OUTA          Send it to the PC.
14 ISG X         Any more to send?
15 GTO 01
16^DONE          No, so finish.
17 BEEP
18 END
```

You may want to add some error-checking for such things as loop status, verification of valid start and stop registers, etc.

In DOS device mode LINK only sends one byte of data for status so you can easily check that with the standard HP-IL function INSTAT. If it is 0, everything is okay otherwise the value returned in the x-register is the DOS error number.

Below is a subroutine to check the status.

```
01 LBL^DEVST
02 INSTAT
03 X=0?
04 RTN
05 BEEP
06^ERROR :
07 ARCL X
08 PROMPT
```

Another potentially useful thing is to be able to read data from the PC, whether it is acting as a physical device or a DOS file. The following code accomplishes this (assuming the PC is the primary device again):

```
01 LBL^READ
02 CF 17
03 LBL 01
04 INA
06 ALENG         This is an extended function.
07 X=0?
08 END
 .
 .
 .       (Whatever you want to do with the incoming string.)
30 GTO 01
```

This code will work fine as long as each string sent from the PC has CR/LF at the end since with flag 17 clear it is looking for that.

As a last item we will ask the PC (which we will assume is acting as a DOS file and has been assigned the primary device) to tell us where in the file it is and then we will set the file pointer to the end of the file so that we can append data. This routine requires the Extended I/O module. To keep things simple, assume that the file is never longer than 64k (65,535 bytes) long so that we can ignore the first byte of the file pointer position.

```
01 LBL^FPOINT
02 REMOTE       Set PC to send/receive file pointer info.
03 INA          Get the file pointer position.
04 ATOXL        Throw away the highest-order bit.
05 ATOXL
06 256
07 *
08 ATOXL        Low order bit.
09 +            File pointer position.
10^POINTER :
11 ARCL X
12 BEEP
13 PROMPT       Display answer and wait for R/S to resume.
14^EOF          The string to set the file to end-of-file.
15 CF 17        Need to put a CR/LF at the end of this.
16 OUTA         Send the pointer position.
17 NOTREM       Put loop devices out of remote enabled mode.
```

Chapter 4
LINK and HP-71

The HP-71 HP-IL module has virtually all of the features built-in to fully access LINK. No other modules or programs are required to run any of the examples presented in this book which will cover methods for "finding" the PC on the loop, setting control parameters, writing to it, reading from it, and restarting it as another device. Several page references will be given throughout this chapter. Unless otherwise noted, they refer to the "HP-IL Interface Owner's Manual".

Before we get into the specifics of LINK we must discuss something that you will see quite often in examples in this chapter, subprograms. The ability to use these is one of the most powerful and unique features of the HP-71 and its implementation of BASIC. Some of the features are: 1) they can be in any program in RAM and accessed by any other program which can minimize coding, 2) you can pass parameters back and forth easily, and 3) any variables used in the subprogram not passed as parameters are unique to the subprogram and will not affect any others by the same name anywhere else in your programs and they "go away" when the subprogram ends. The main disadvantage of them is speed. A subprogram call can take as long as .4 seconds to set up and begin execution so if you have intensely iterative calculations it is best to use a subroutine or a user-defined function. The ones that follow fall into the category of those that are not called repetitively.

Finding the PC on the loop can range from very easy for "custom" applications where the loop configuration is known to a bit more involved if you want to find it under any circumstances. Following is a subprogram that has two parameters, "A" the loop address of the PC if it is found, and "A$" the string you want to send to the PC to have it restart as which will be returned as the first two characters of the name of the PC. The reason for specifying the PC device type is that if you are writing a program to use LINK as some device it may be possible that someone (maybe even you) will start LINK in the wrong configuration. This little routine takes care of any possible errors.

```
100 SUB FINDPC(A,A$)
110 ON ERROR GOTO 190 @ IF A$="" THEN BEEP @ ENDALL ! Check
        the string.
120 STANDBY OFF @ RESET HPIL @ CONTROL ON @ RESTORE IO
        ! Set up loop.
```

```
130 FOR A=1 TO 32 @ I=DEVAID(A)
140 IF I<0 THEN 180 ! No AID mean we have gone around the loop.
150 B$=DEVID$(A) @ IF B$="" THEN 170
160 IF POS('PO PL DO TU',B$[1,2]) THEN 200 ! Found it.
170 NEXT A
180 BEEP @ DISP 'PC not in loop!' @ ENDALL ! Did not find so ended
      everything.
190 BEEP @ DISP ERRM$ @ ENDALL ! Error handler.
200 SEND UNT UNL LISTEN A DDL 31 UNT UNL ! Command to
      reconfigure the PC
210 ENDLINE @ OUTPUT :A;A$ ! Send the new device type to the PC.
220 ON ERROR GOTO 230 @ I=0 @ WAIT 2 ! Give the PC time to reset.
230 I=I+1 @ IF I>5 THEN 190
240 RESET HPIL @ RESTORE IO ! Establish new addresses and clear
      loop.
250 A$=B$[1,2] @ END
```

Without the comments this file takes up only 351 bytes.

Option 1 - Operation as a Disc Drive

Please refer to pages 28-33 of the HP-IL manual as well as pp. 109-119 and 247-264 of the "HP-71 Owner's Manual" for a discussion of how to work with files. You can do any standard function with a file on the PC (in the DOS file HPILDISC) that you can do with a file in RAM. There is a minor difference in the command RUN. If you RUN a file from disc it is first read into RAM and then executed.

The PC will emulate a standard HP-type disc or cassette drive and the 71 is unable to distinguish it from them. The only significant differences are speed (HPILDISC is much faster than either HP drive especially when using a hard disc on the PC), and media size. The cassette drive is limited to 128k (131,072 bytes) and the HP9114 is limited to 616k (630,784 bytes) but HPILDISC can hold up to 8,192k (8,388,608 bytes) for a hard disc of 10 megabytes or more. Also the size of HPILDISC is dynamically adjusted by LINK so that when LINK terminates the unused disc space is returned to DOS. For virtually all normal operations though there is no difference even down to the error messages that are returned.

LINK treats the disc drive emulator mode just like a normal tape or disc on HP-IL. You cannot use the disc until you format it (INITIALIZE on the 71). See pp.127-128 of the HP-IL manual for a discussion of this operation. Please note that when you INITIALIZE a disc you will erase everything that was on it if it had been previously formatted and written to.

To copy a file named MYFILE from RAM to HPILDISC if there is no other mass storage device in the loop:

```
COPY MYFILE TO :TAPE
```

To read it back into RAM:

```
COPY MYFILE:TAPE
```

One of the more beneficial features of LINK is that data are buffered into and out of the actual disc. The speed increase is very noticeable especially for sequential and random access of DATA and TEXT files. Data transfer rates are limited by HP-IL rather than the drive and data transfer rates of 5,000 bytes per second are not uncommon.

CAUTION - When you use the 71 command INITIALIZE to "format" a tape you have an optional entry for number of directory entries. If you do not enter a number of entries, the 71 assumes that you want the default which it calculates to be the number of records on the medium divided by 4. This is a reasonable number where an entire disc is dedicated to HP files but with LINK you usually share the disc with DOS files and if you blindly accept the default value you can waste quite a lot of space without knowing it. For example, the HP digital cassette drive will have a default directory size of 4k (128 entries) but if you have a hard disc with the 8.4 megabytes the default directory size will be 256k (8,192 entries)! If you intend for the hard disc to be primarily HP files then this may be okay but otherwise you tie up a tremendous amount of storage that you will probably never use. As a general guideline you should supply your own directory size.

OPTION 2 - Operation as a Printer

This option of LINK emulates an HP-IL printer fairly completely. Please refer to pp. 22-26 of the HP-IL manual and also pp. 224-239 of the Owner's Manual for additional information regarding printing and printers.

Using the subprogram FINDPC presented earlier in this chapter, the following line can be used to set the PC as a printer and make it the PRINTER IS device.

```
CALL FINDPC(A,'P') @ PRINTER IS :A
```

Whenever you use the PRINT or PLIST commands, the data will be printed on the PC printer. You do not have to designate the PC as the PRINTER IS device to send data to its printer. If A is the loop address (from FINDPC) you can use the OUTPUT command. For example, to print the string A$ just enter:

```
OUTPUT :A;A$
```

One important point to consider when you use software that accesses the printer is that most programs assume that either the HP 82162A Thermal printer or the HP ThinkJet printer is connected on the loop. LINK however allows any type of printer which can be connected to a PC to be used as it just passes data from the 71 to the printer. If you are printing text with no enhancements you should have no problem but if your printed output includes graphics or other enhancements such as bold, underline, double wide, etc. then you need to check your owner's manual for your printer to determine what character sequences you need to send to it to achieve the desired results and then modify your software if possible. In general if your PC is connected to another HP printer you should have minimal problems but if you are using an Epson, IBM, etc. you will need some changes for these enhancements.

OPTION 3 - Operation as a Video Display

In this mode LINK emulates other HP-IL video interfaces by displaying data on the PC screen. LINK also provides several enhancements not available on any other video interface. Please refer to pp. 22-26 of the HP-IL manual and also pp. 224-239 of the Owner's Manual for additional information regarding displays.

Typically you will designate a display device either as the system printer or the system display. If it is designated the system printer (via a PRINTER IS assignment) then all normal printing functions such as PRINT and PLIST will go to it. Normally this is used when you run a program for which you do not want a hard copy of but that you do not want to match the 71 display. For example, you could prompt for data on the 71 and output answers to the screen.

If it is the system display device (via a DISPLAY IS assignment) then all displaying functions such as DISP, LIST, and CAT go to it. In addition, the system display has everything that happens in the 71 display. Not all devices support the same control sequences that the 71 display does but LINK does. When you backspace, delete characters, insert characters, etc. it will appear on the screen as well

as the 71 display. The obvious advantages are that you can see a full line rather than a window and you have a full screen of lines.

In addition to being able to match the 71 display, LINK has the capability to have the cursor moved around the screen so that you can put any characters anywhere on the screen. Also you can turn the cursor off and on although it still functions normally either way. Another feature LINK provides is what is called inverse video. On monochrome monitors the characters are normally light against a dark background but this can be flipped for any character so that it is a dark color against a light background. On a color monitor (IBM PC family only) inverse video produces a grey character as opposed to the normal yellow.

LINK provides a feature not found on any other video interface. For the IBM PC family you can specify color for each character or set it and it will remain for all characters (HP PC's ignore these commands). Not only can you specify the character color, you can also specify background color.

On all PC's LINK provides another unique capability that enables you to make any character blink, regardless of whether it is in inverse video or another color. One other unique feature which may be useful is the ability to read the current location of the cursor on the screen at any time.

As a demonstration, we present below a program that will display the phrase "Who is John Galt?" (refer to "Atlas Shrugged" by Ayn Rand) on line 10 of the PC screen in inverse video starting at column 30. At the end we will find out the cursor position. This program assumes that there is only one video interface in the loop.

```
110 PRINTER IS :DISPLAY ! Sets the PC to receive printed data.
120 PRINT CHR$(27);'E'; ! Clear the screen.
130 PRINT CHR$(27)&'%'&CHR$(30)&CHR$(10); ! Position the cursor.
140 A$='Who is John Galt?'
150 FOR I=1 TO LEN(A$)
160 PRINT CHR$(NUM(A$[I])+128); ! Inverse is normal ASCII number +
      128
170 NEXT I
180 I=SPOLL(:DISPLAY) DIV 256 ! Get the cursor position (discard first
      byte)
190 DISP 'Cursor col:';MOD(I,256);' row:';I DIV 256
```

The second example is a bit more sophisticated. We want to have a subprogram that will print a message on the bottom line of the screen and then wait until the user presses a key so that the message can be cleared. Assume that we know nothing about what PC we are connected to (which will make a difference because if it is an IBM we want to put the message in red but for all others we want just inverse video). Also, the message should blink. One minor problem to

resolve is that all of the PC's that can run LINK have 25 lines or more on their screens except the HP Portable which only has 16. The call parameters to the subprogram are A and M$, the address of the PC and the message.

```
100 SUB MESAGE(A,M$) ! A is the address of the PC and M$ is the
      message.
110 A$=DEVID$(A)[1,2] ! Use the first two characters of PC device name.
120 IF A$='PO' THEN B=15 ELSE B=24 ! Max. row on HP Portable is
      15
130 GOSUB 'CLEAR' ! This puts the cursor on the last line and clears it.
140 OUTPUT :A;CHR$(27)&'Y'; ! Specify blinking.
150 IF A$='DO' THEN OUTPUT :A;CHR$(27)&'e'&M$&CHR$(27)
      &'o' @ GOTO 170
160 FOR B=1 TO LEN(M$0 @ DISP CHR$(NUM(M$[B])+128); @ NEXT
      B
170 OUTPUT :A;CR$(27)&'%'&CHR$(54)&CHR$(B)&CHR$(27)&'z';
180 BEEP @ OUTPUT :A; 'Press any key to continue';
190 A$=KEY$ @ IF A$="" THEN 130 ! Wait for a key to be pressed
200 GOSUB 'CLEAR' @ END ! Clear line and end.
210 'CLEAR' : OUTPUT :A; CHR$(27)&'%'&CHR$(0)&CHR$(B)&CHR
      $(27)&'J'; @ RETURN
```

This sub also shows that you can send data to any HP-IL device without having to specify it as a PRINTER IS or DISPLAY IS device.

Note that in line 190 the loop with KEY$ could be replaced by the LEX KEYWAIT$ but if you are writing routines for general use you must assume that not everyone has it.

You can use ERRM$ as the M$ parameter when you call the subprogram and also you can easily set A by using the subprogram FINDPC which was presented at the start of this chapter. In addition to displaying error messages this subprogram has uses where you want to display multiple lines of data and then give the user time to read it (Help info, results, etc.).

If you have an IBM PC you may want to try the following line with the PC as the DISPLAY IS device (or use OUTPUT and the loop address):

```
FOR I=97 TO 112 @ DISP CHR$(27)&CHR$(I);'Color' @ NEXT I @ DISP
      CHR$(27);'o'
```

This will print the word "Color" in 16 diffeent colors then reset the color to the default of yellow. It will also run on monochrome monitors but instead of color you will see two of them underlined.

If you want to change the background color from the default of black:

```
DISP CHR$(27);'x';CHR$(27);'H';CHR$(27);'J';
```

The sample uses 'x' as the background color which is light grey but it can be anything from 'q' to 'x'. The background color is set and then the cursor is sent home (i.e. row 0, column 0) and then the screen is erased which makes the background color take effect for the entire screen.

There is one other feature of LINK you may find worthwhile if you have either the FORTH/Assembler ROM or the Keyboard Is LEXfile from the HP User's Library. When in the video interface mode, the PC's keyboard is active and any keys you press are held in an input buffer until requested. Up to 255 characters can be held in the buffer before additional ones are ignored. If you set the PC as both the DISPLAY IS and the KEYBOARD IS device then not only will your keystrokes on the PC be sent to the handheld but also will show back up on the PC display. It will appear to you that the PC is acting like a 71 but it really is not. Please refer to Appendix B for a more complete discussion of the characters sent by the PC in this respect. Below is a sample line you could use to set the PC in video mode and to set it as both DISPLAY IS and KEYBOARD IS:

```
CALL FINDPC(A,'V') @ DISPLAY IS :A @ KEYBOARD IS :A
```

If for some reason you do not want the PC keyboard active (note that it always checks for SELECT or F10) then you can send Device Dependent Talker command 0 to the PC after you make it a talker. To restore the PC keyboard to it's normal active state, send Device Depedent Talker command 1 to it. Below is a sample line to turn it off assuming the PC is the only :DISPLAY device in the loop:

```
100 SEND UNT UNL TALK ":DISPLAY" DDT 0 UNT UNL
```

Note that when you change between active and inactive modes that the input buffer is cleared of any keystrokes that you have done on the PC that have not been sent to the handheld.

OPTION 4 - Operation as a Device

This mode of operation is the most complex of all but it also offers the most opportunity for additional growth of new applications for the 71 (and all of the handhelds for that matter). Fortunately, no additional modules other than the HP-IL are required for this mode of operation.

In the main body of this manual a program is presented which enables data transfer from any PC text file to the 71. Also with minor modifications noted in the text it will work as a means of transferring data from the PC keyboard. In this section will be presented examples of methods of writing to PC files as well as reading/writing to physical devices.

One of the potential problems with using devices is that these are new functions for the 71 and there are not built-in error sensing mechanisms in place like there are for the disc drive. To fully ensure that all of the data you send and/or receive from LINK are successfully transferred you should periodically use the 71 function SPOLL to check the status of the PC. If you check the status of the PC and no error has occurred then the character 0 is returned which the 71 will report as a numeric value of 0 in SPOLL. If an error has occurred the PC will send one byte, the ASCII number of which corresponds to the DOS error number. The 71 converts this to the same decimal number as the byte number.

One suggestion regarding the use of SPOLL (or any other HP-IL command that may be frequently used) - The format of the device specifier can affect the speed of your program. If possible, specify the device specifier as a loop address rather than a device class or device name since if you do it by name, class etc. the 71 has to go on the loop each time to "find" the device before it can send the appropriate data or commands to it but if you have specified a loop address the HP-IL message can go right to it. This requires a bit more code but if you iterate a lot then it will save a lot of time. Below are two versions of a section of code that accomplish essentially the same thing but the second one will function much more rapidly if you use it a lot. The routine just asks the PC for it's status.

```
500 A=SPOLL(':INTRFCE') ! This is the simpler but slower method in
      loops.

100 Z9=DEVADDR(':INTRFCE')
 .
 . (other program steps)
 .
500 A=SPOLL(Z9) ! Message goes right to device at loop address ''Z9''.
```

Uploading a 71 TEXT file to the PC is a simple matter. We will use the sub FINDPC to shorten the code. It will prompt for a filename and then make a file on the PC by the same name. If one already exists on the PC it will prompt the user for whether to abort, overwrite, or append the file. At termination, the PC is set to a "null device" so that future loop operations will not go to the file. Note that in the world of DOS the NUL device is the "great byte heaven in the sky"; anything written to NUL will simply disappear.

```
90 SUB UPTEXT
100 DIM A$,T$[500],A,B,C,I
110 INPUT 'TEXT Filename:';A$ @ IF A$='' THEN END
120 ASSIGN #1 TO A$
130 A=POS(UPRC$(A$),':TAPE') ! Check to see if it is on a tape/disc
140 IF A<>0 THEN A$=A$[1,MIN(8,A-1)] ! DOS filename is limited to 8
      characters
150 A$='M '&A$ @ CALL FINDPC(A,A$) ! Reconfigure the PC to access
      the file.
160 B=DEVAID(A)
180 IF B=68 THEN DISP A$;' is a phys. device' @ GOTO 'NUL'
190 IF B=69 THEN 260 !   AID of 69 implies a new file.
200 BEEP @ DISP 'Ovrwrt,Append,End? OAE'
210 B$=UPCASE(KEY$) @ IF B$="" THEN 210 ! Could use KEYWAIT$
      here instead.
220 IF B$="E" THEN GOTO 'NUL'
230 IF B$="O" THEN B$='CUT' ELSE IF B$='A' THEN B$='EOF'
      ELSE 210
250 REMOTE @ OUTPUT :A ;B$ @ LOCAL ! Truncate or file pointer to
      end-of-file.
260 ON ERROR GOTO 'ERR'
270 READ #1;B$ @ OUTPUT :A ;B$
280 B=SPOLL(A) @ IF NOT B THEN GOTO 270
290 BEEP @ DISP 'DOS Error #';STR$(B-32) @ WAIT 2
300 'NUL': LOCAL @ OUTPUT :A ;'M NUL' @ DISP 'PC is NUL
      device'
310 ASSIGN #1 TO * @ DIM B$ @ END !        Clean up and end
320 'ERR': IF ERRN=255027 OR ERRN=54 THEN DISP 'OK-'; @
      GOTO 'NUL'
321 ! Error numbers above are for end-of-file for tape/disc drive or RAM.
330 BEEP @ DISP ERRM$ @ WAIT 2 @ GOTO 'NUL'
```

Note that to fully make the program error-proof you could insert some code in place of line 20 to check if the file existed prior to ASSIGNing it and also to ensure that it is a TEXT file rather than DATA, SDATA, BASIC, etc. The program could easily be modified to upload individual lines entered at the keyboard by replacing line 270 with an input prompt with an appropriate check to send the program to the NUL routine at the end (could be sensed off of a null input, a certain character, etc.).

Now that you have seen how to read and write to DOS files/devices you can fairly easily write a program to access either a modem or an RS-232 port connected to a modem and have true data communications. You can even fairly easily set up TEXT files to automatically dial the phone number for you and log on. It would be

simple to go one step further to have the 71 automatically transfer a given file then log off. Please refer to your PC modem for a discussion of it's auto-dial capabilities, etc. In fact, existing software for the 71 will work in this capacity as long as it looks on the loop for a general interface type HP-IL device rather than an acoustic coupler modem (AID=65) or an HP-IL/RS-232 interface (AID=66) specifically.

As for transferring DATA or SDATA files, just change the READ statement above to be one that reads a number then converts it to a string with the STR$ function. One of the more difficult things you can do is to send data from a DATA file that has both strings and numbers in it. If the DATA file follows standard HP-format (called HPAF for HP Application File format) then the following program will work to upload a DATA file on the 71 to a file that can be imported into a Lotus 1-2-3 spreadsheet. (there is no error checking shown here to make sure it really is an HPAF DATA file).

```
1000 SUB UPDATA
1010 DIM T$[500] @ STD
1020 INPUT 'DATA Filename:';A$ @ IF A$='' THEN END
1030 ASSIGN #1 TO A$
1040 A=POS(UPRC$(A$),':TAPE') ! Check to see if it is on a tape/disc
1050 IF A<>0 THEN A$=A$[1,MIN(8,A-1)] ! DOS filename is limited to 8
       characters
1060 A$='M '&A$&'.PRN' @ CALL FINDPC(A,A$) ! Reconfigure the
       PC
1070 B=DEVAID(A) ! We will check if it is a new file by the accessory ID
1080 IF B=68 THEN DISP A$;' is a phys. device' @ BEEP @ WAIT 2 @
       GOTO 'NUL'
1090 IF B=69 THEN 1180 ! AID of 69 implies a new file so we can just
       write.
1100 BEEP @ DISP 'Ovrwrt,Append,End? OAE' ! Existing file options.
1110 B$=UPRC$(KEY$) @ IF B$="" THEN 1110
1120 IF B$="E" THEN GOTO 'NUL'
1130 IF B$="O" THEN B$='CUT' ELSE IF B$='A' THEN B$=
       'EOF' ELSE 1110
1140 REMOTE @ OUTPUT :A ;B$ @ LOCAL ! Either cut off file or go to
       the end.
1150 READ #1;B$ @ B$=UPRC$(B$[5]) @ C=LEN(B$)
1160 FOR I=1 TO C @ IF B$[I,I]='N' THEN SFLAG(I) ELSE CFLAG(I)
1170 NEXT I
1180 READ #1,1;K ! This is the number of records to transfer
1190 ON ERROR GOTO 'ERR'
1200 FOR I=3 TO (K+2) @ READ #1,I @ IF C=1 THEN 1240
1210 FOR J=1 TO (C-1)
```

```
1220 IF FLAG(I) THEN READ #1;B @ B$=STR$(B) ELSE READ #1;B$
@
     B$='""'&B$&'""
1230 OUTPUT :A ;B$;' '; @ GOSUB 'CHK' NEXT J
1240 IF FLAG(I) THEN READ #1;B @ B$=STR$(B) ELSE READ #1;B$
1250 OUTPUT :A ;B$ @ GOSUB 'CHK'
1260 NEXT I @ DISP 'OK-';
1270 'NUL': LOCAL @ OUTPUT :A ;'M NUL' @ DISP 'PC is NUL
     device'
1280 ASSIGN #1 TO * @ DIM B$ @ FOR I=1 TO C @ CFLAG I @ END !
     Clean up and end
1290 'ERR': BEEP @ DISP ERRM$ @ WAIT 2 @ GOTO 'NUL'
1300 'CHK': B=SPOLL(A) @ IF NOT B THEN RETURN ! Spoll=0
     means no error.
1310 BEEP @ DISP 'DOS Error #';STR$(B) @ WAIT 2 @ POP @ GOTO
     'NUL'
```

This program will take each set of data in a record and output it to the PC with all of the items separated by spaces and the strings are in quotes (" "). The filename you specify is limited to 8 characters (truncated if necessary) and ".PRN" is added to it for the DOS name. This is the format required by Lotus 1-2-3 for importing text and numbers into a spreadsheet (using the /File Import Numbers command). The numbers will come into the spreadsheet as-is and the strings will be unquoted. You can upload a file from mass storage but do not specify a filename to upload which is on an HPILDISC on the same PC that you will upload to since LINK cannot act as a disc drive and a file at the same time.

Uploading HP-71 program files in ASCII format for editing on the PC is very simple with LINK's device mode. There are two ways: 1)TRANSFORM it into a TEXT file and then use the program given above, or 2)just make the PC the PRINTER IS device and use PLIST. The second method is quite a lot quicker for long programs but you should remember to set PRINTER IS to something other than the PC when you finish to avoid possibly sending more data later.

Downloading ASCII versions of program files requires that you transfer into a TEXT file and then TRANSFORM it into BASIC.

Note that if you want to directly transfer any file (most of which are in binary format) you can use the standard 71 COPY command to upload or download a file to or from the PC. Any 71 file can be COPY'd directly and in this respect the PC can be used in this mode as a "disc drive" for the 71. Note that any file you upload in this manner will not be directly usable on the PC since HP-formatted files are very different from MS-DOS files. In general, if you want to store files on the PC you should use the disc drive emulator. However, if you need to have a separate DOS file for your 71 file, the

mechanism is in-place for the transfer. All you have to do is to set up LINK in device mode and select a file. Then you use the standard COPY command to copy the file to the PC. When it is stored in the PC file, the file will be in HP-format and the first 32 bytes of the file are a directory entry (filename, size, type, etc.).

TEXT files are the closest files on the 71 to typical MS-DOS files but there is a difference nonetheless. In a typical MS-DOS file, each line of text ends in a carriage return/linefeed (CR/LF). In a 71 TEXT file though, CR/LF is not present. Instead, two bytes precede each line of text which indicate the number of bytes in the line and the line terminates with an ASCII 0. In addition, some MS-DOS files use ASCII character 26 (Ctrl-Z) as the end-of-file indicator but HP uses two ASCII 255's.

You can also directly COPY any file on the PC into the 71 but keep in mind that the 71 is expecting the first 32 bytes to be a directory entry. If you are interested in these types of transfers and want to know how these 32 bytes are supposed to be, please refer to Appendix D of the HP-IL Interface Owner's Manual page 228 for a table of directory information requirements.

The subprogram below illustrates how to copy any file in the 71 to the PC with the extension ".71" added to it.

```
2000 SUB TODOS
2010 INPUT 'Upload file : ':A$ @ IF A$="" THEN END
2020 CAT A$ ! If the file does not exist this will generate an error to stop
2030 FINDPC(A,A$&'.71") ! Same subprogram as before to configure the
       PC.
2040 COPY A$ TO :A ! Copy it to the PC
2050 END
```

HP-71 GENERAL COMMENTS

LINK is designed so that only certain severe errors in DOS can cause it to abort. These are rare and the majority of errors are trapped in LINK and just affect the status bytes you request via SPOLL. Should you get a 'Loop Broken' error on the 71 it could be due to one of the severe errors.

Unless there is a power failure in the PC there should never be a terminating error on the PC while it is in disc drive emulaton mode or video display mode. You could get an error message on the PC in printer emulation mode if you send data to the PC and the printer is not connected, turned off, out of paper, etc. The familiar "Abort, Retry Ignore?" message we have come to know and hate with DOS

will appear to give you a chance to correct it before it crashes (which will not hurt the PC, only terminate the current run of LINK).

There are a few occasions when LINK may appear to "lock up". These can only occur when it is sending data and are unusual. The way HP-IL works the device sending data must get whatever data it sent back (i.e. it has to go all the way around the loop) before it can send any more. If the loop is broken while this data sending is in progress LINK can appear to "lock up". Prior to rebooting your PC give it a chance to work properly by keying ATTN on the 71 twice to interupt the HP-IL system. Then enter RESET HPIL @ RESTORE IO.

Also there is a unique feature about the 71 compared to the other hanhelds that may at first seem strange when it happens. After you get LINK running on the PC you try to use it or anything else on the loop and you get an error message saying "Invalid Mode" and you cannot seem to get rid of it with RESET HPIL, RESTORE IO or any other normal commands.

What has happened is that the 71 is not the loop controller so it cannot do whatever you are trying to do (most things the 71 does such as read/write files, display and print are done as controller). The 41 and 75 must always be controller of the loop so they cannot have this occur. This causes a lot of other limitations for these machines but is somewhat offset by this problem. In general what can/does hapen to the 71 is that some other HP-IL device took control of the loop and the 71 automatically reverted to a role as a device on the loop. Unfortunately one of the ways the 71 can be forced to give up control is in response to an "Interface Clear" command which is very common to controllers, including 71's, trying to set up the loop and do some things. If you have the 71 in the loop and try to do something on the loop with your PC before you run LINK it will unwittingly turn the 71 into a device.

The single-most common occurrence of this is when the PC is trying to load LINK from a disc drive on HP-IL such as the HP9114A disc drive and it can occur very easily on the HP Portable/Plus which rely on HP-IL for all external disc drives. The solution to the problem is to enter the command CONTROL ON after LINK is running which will re-establish the 71 as the loop controller (no conflict with the PC at this time since when LINK is running it is a device). One prevention of the problem is to keep the handheld out of the loop until LINK is up and running.

Chapter 5
LINK and HP-75

The HP-75 has an HP-IL module built-in and can easily utilize the disc drive, printer, and video display modes of LINK just as described in the "HP-75 Owner's Manual" pp. 124-140. In addition there are several programs and modules which can greatly increase the ease of use of the loop and increase its use especially in the MS-DOS device mode of LINK.

The biggest drawbacks of the HP-75 with the HP-IL system as it is in the "bare bones" 75 are: 1) the HP-IL devices must be manually assigned device codes each time the 75 is turned on or the loop configuration changes, 2) DATA and TEXT files cannot be read from/written to on a disc but must first copied to RAM, 3) no methods exist for obtaining any data or status from any device on the loop, and 4) the 75 cannot act as a device on HP-IL but rather must always be the controller.

The AUTOLOOP program available from the HP User's Library (also in the I/O module) gives you the luxury of not having to assign device codes manually. The Extended I/O module overcomes many of the shortcomings with general data interchange on the loop and gives you much more control over loop operations. LINK avoids the fourth problem above since it makes the PC a device and the 75 can remain as the controller although only one 75 or 41 can be in the loop at any one time.

This chapter will cover only the basic 75 and briefly touch upon the AUTOLOOP program. For those of you who have the Extended I/O module and would like to apply its capabilities to LINK please refer to the chapter for the HP-71 since its HP-IL capabilities and programming language are very similar to the 75.

If you do not have software for automatically assigning loop address and device codes you must follow the insructions on pp. 126-127 of the owner's manual. This chapter assumes that you or your software have used the following nomenclature for device codes:

First letter - M : mass storage device
P : printer
D : video display
I : interface device (including LINK MS-DOS device mode)

Second letter will be "1" for the first device in the loop, "2" for the second, and so on. For example, P2 is the second printer in the loop. These are the same codes used by the AUTOLOOP program.

OPTION 1 - Operation as a Disc Drive

In this mode LINK emulates a standard HP-IL disc drive. Refer to pp. 132-138 of the owner's manual for details on how to use disc drives on HP-IL. Do not forget that LINK is just like a normal disc drive in that you cannot use it to store data and programs until you have formatted the disc (INITIALIZE is the command on the 75). Failure to do so will result in error number 96, "Invalid Medium". Regardless of how much room you have on your HPILDISC with LINK the basic 75 can access only 131,072 bytes (128k). With the Extended I/O module you can write software to access the full amount HPILDISC may have which is up to 8.4 megabytes. Regardless of how much disc space is available on HPILDISC when LINK terminates HPILDISC will only be as large as the number of bytes you (or the 75) have actually written to it. The rest is given back to DOS.

The 75 owner's manual does a good job of describing how to use mass storage devices and includes some examples. Since LINK emulates a normal disc drive essentially completely from a user's standpoint, there is no need to duplicate the owner's manual's efforts.

OPTION 2 - Operation as a Printer

In this mode LINK makes the PC appear as a printer device on the loop. All data sent to LINK will be routed to the standard PC printer. With the basic 75 you can designate the PC either the PRINTER IS or DISPLAY IS device. If you designate it the DISPLAY IS device then all displaying commands such as DISP and LIST will be directed to the PC printer as well as everything that you do in the display. It is generally not recommended that you use the PC printer as the DISPLAY IS device when you are editing in the 75 display since everything goes to the printer including backspaces, character deletion, etc. and the codes used by the 75 are not the same as any printer to do the same functions so you may get strange output.

If you designate the PC as the PRINTER IS device (done automatically by AUTOLOOP if it is the first or only printer in the loop) then all printing functions and commands are directed toward the PC such as PRINT and PLIST. Please note that until now there have been few opportunities to use printers other than HP's since

most had to be connected via HP-IL and not too many other printers had this type connector. As a result, your software may send HP-type printer codes if it does unusual things such as bold, underline, graphics, etc. Due to the wide variety of printers which may be connected to PC's it is virtually impossible for LINK to decipher HP-type commands and escape sequences and transform them into the appropriate codes for all the various printers. LINK cannot tell what type of printer is connected (in fact it does not even know if there is a true printer connected); all it does is pass the print information to the port selected on the PC device as the PRN device.

If your software does not do unusual things then most probably it will print okay with your PC printer. You should refer to the manuals for your software and your printer if problems develop (strange characters etc.) to determine how to modify the software to properly control your printer.

OPTION 3 - Operation as a Video Display

The PC may be designated as either the DISPLAY IS or PRINTER IS device, or both. Refer to the preceeding section for the effects of each. Since LINK emulates the HP video interface you will be able to use the PC screen very effectively if you (or your software) have designated it as the DISPLAY IS device. The actions on the screen will match your display even while you are editing things and using DEL, I/R, BACK, etc.

Another good application for the PC as a video display is to set DISPLAY IS * and PRINTER IS ':D1' for program execution. In this mode your keystrokes will not be displayed on the PC screen but all printed output from your program will be on the PC screen. This is worthwhile for those cases where you want to look at the output but you do not need a hard copy.

If you have a member of the IBM PC family with a color monitor you can easily change the foreground and background colors from the keyboard or from a program as shown below in a sample program extract:

```
1000 DISP CHR$(27)&'x'&CHR$(27)&'H'&CHR$(27)&'J';
1010 ! Above sets light grey as the background color and clears screen.
1020 ! If using a non-color monitor it just clears the screen.
1030 DISP CHR$(27)&'c'; ! Sets blue as the character color.
```

Below is a routine that will place random color blocks on the screen until you press any key.

```
100 DISP CHR$(27);'E'; ! Clear screen and reset colors to defaults.
```

```
110 DISP CHR$(27);'%';CHR$(RND*79);CHR$(RND*23);
120 ! Set the cursor randomly on the screen from rows 0 to 23.
130 DISP CHR$(27);CHR$(RND*7+113);' ';
140 ! Set random background color and print a space.
150 IF KEY$='' THEN GOTO 110
160 DISP CHR$(27)&'E'; ! Clear screen and reset colors to defaults.
```

You may notice that when DISPlaying data to the PC screen that it seems to take a long time for the characters beyond the 32nd one to get on the screen. This is not a problem with LINK but rather is a characteristic of the method the 75 uses to DISP data. The next time this happens to you notice that the characters are appearing on the screen at the same rate they are scrolling across the 75 display and the reason this effect does not begin until 32 characters have been displayed is that the 75 display is 32 characters wide so scrolling in the 75 display does not start until the 33rd character.

If you have the Extended I/O module you can request the current cursor position from the PC via the SPOLL$ function. The second character will be the column number and the third character is the row number.

Another feature of the video interface mode is the live keyboard. If you have the Extended I/O module you can set the PC to be the KEYBOARD IS device and use it as a remote keyboard for the 75. In addition if it is also set as the DISPLAY IS device then everything in the 75 display will also show up on the PC screen. From the PC standpoint it will appear as if the "brains" of the PC have been replaced by the 75. Appendix B has a more detailed discussion of the keyboard option.

OPTION 4 - Operation as a Device

As described in the main documentation, LINK gives you the ability to access any physical device or file on the PC. Unfortunately, with the basic 75 all you can do is set the PC as the DISPLAY IS or PRINTER IS device and then output data or program listings to it. With this you have the ability to upload information to the PC into a file but nothing can be sent back to the 75.

You can also access printers and plotters with this mode. Note that if you choose this mode on the PC and use a program such as AUTOLOOP on the 75, the PC will not be designated as either the PRINTER IS or DISPLAY IS device since in this mode LINK appears as an interface device and not a printer or video display.

If you have the Extended I/O module you can send data to the PC and also read data from it so that you can download data,

communicate through modems, etc. Please refer to the section for the 71 use of this feature. The functions in the 71 are very similar to those offered by the 75 Extended I/O module. There are a couple of differences worth noting: a)most of the functions require that you use the device code (e.g. D1, M2, etc.) to specify the loop device rather than a device type (e.g. :PRINTER) or loop address, and b)the 75 does not use flags as the 71 does so the method of controlling the end of input of data from a device on the loop is not determined by flag -23 but rather by some formatting control in the ENTER statement (refer to the discussion beginning on page 25 of the Extended I/O manual).

APPENDIX A
HP-IL Message Responses

In general, the loop controller has a command set that actually sends the appropriate HP-IL commands around the loop. This command set is usually very simplified for the user. For example, when an HP handheld is given the command to get the directory of a tape/disc ("DIR" on the HP-41 or "CAT :TAPE" on the HP-71B, etc.) the user sees this as only one command. However, several hundred messages are passed around the loop as the controller causes the tape drive to clear itself, position itself to the right spot on the tape, read some data, send the data, error check, etc. For those who need the degree of detail on the response of the PC to individual HP-IL messages, the following is presented.

COMMON TO ALL OPTIONS

IFC - Clear the PC from listener and talker status.

SDC - If listener, PC cleared as with DCL command.

UNL - PC removed from listener statuus.

UNT - PC removed from talker status.

LAD - If address matches, PC becomes a listener.

TAD - If address matches, PC becomes a talker. If address does not match, PC is removed from talker status if it is a talker.

RFC - Passes on to next device.

SST - If talker, sends one or two status bytes (refer to individual tables).

SDI - If talker, sends device ID string composed of two parts, a PC-dependent part and option-dependent part followed by CR/LF as follows:

Computer	PC part	Option	Option part	Access ID	AAU Address
HP Portable	PO	Disc drive	DRV	16	2
HP Plus	PL	Printer	PRN	36	1
HP 150 (Touchscreen)	TU	Display	DSP	48	31
		DOS Device	devicename	68	4
IBM PC et al (DOS PC)	DO	New DOS File	filespec	69	4
		Existing File	filespec	70	4

Ex. - IBM PC acting as a disc drive, the string is DODRV with CR/LF.

HP 150 accessing TEST.PRN, the string is TUTEST.PRN with CR/LF.

SAI - Sends one byte according to option type (see above table).

NRD - If talker, makes previous data byte the last one sent and when it is returned, ETO is sent for no errors or ETE for transmission errors.

ETO - During data transmission LINK continuously checks for transmission errors

ETE and if an error is detected, transmission halts and an ETE is sent. If no error is detected, an ETO is sent at the end of the transmission.

AAD - Takes address as its own, increments address by 1 and passes on.

AAU - PC reverts to its address as given in the table on the preceeding page.

DDL - If listener and DDL number is 31, the PC is set to a mode such that subsequent Data Bytes are retained in a special buffer. This continues until 45 characters have been received or a CR/LF is received, at which point the PC resets itself as a device based upon the data in the buffer while retaining the same loop address. Note that after the change, the PC is neither a talker or listener until so instructed. If the DDL number is not 31, refer to each device option for the response, if any.

IDY - No response. See video display mode for only exception.

All messages and commands not listed here or below cause no response.

MESSAGES SPECIFIC TO "DISC DRIVE" MODE

LINK maintains two internal buffers for data read/write storage. The first internal buffer always contains records 0-23 (each record is 256 bytes long). This "lower" buffer is only written to the PC file HPILDISC when the user exits disc drive mode. The other internal buffer contains 48 records and the start record floats as necessary but is never less than 24. It is written to the disc only when necessary which is when other data is sent to or read from a record number greater than 23 (i.e. not stored in the lower buffer) but outside the current range of the upper buffer or when the user exits disc drive mode. For example, if the range is currently from 60-107 and a write is made to record 35, records 60-107 are written to HPILDISC and the range of the upper buffer is set to 35-82, with a read from HPILDISC for the data to fill the buffer.

NOTE - 1) Regardless of the range of the upper and lower internal buffers, they may only be partially written to HPILDISC depending on the maximum record number ever written. For example, if the maimum record written is 2 (typical just after "formatting") if the disc drive mode is exited, only records 0,1 and 2 from the lower buffer are written to HPILDISC and the upper buffer is not written at all. This saves DOS disc space.

2) If the specified record to write is greater than the largest record number that has ever been written to that file, the records between the previous largest record and the new largest record are filled with ASCII character 255 and the upper buffer is now set to start at the new largest record if it is greater than 23.

3) Since HP handhelds use record 0 as the "volume label" record of mass storage which contains, among other things the number of surfaces, tracks/surface, and records/track and this disc drive emulator has those as variables (dependent upon actual DOS disc space available), LINK will set these in record 0, bytes 24-35, whenever the disc is formatted or when the disc drive mode is started if it is already formatted. The actual values of these 12 bytes are the same as determined by DDT 6 below. If HPILDISC is formatted but not written to by the user or is not formatted by the user, it will not exist on the DOS disc when disc drive emulation is ended. If it exists but the user reformats it (DDL 5) it is first erased in DOS and then recreated.

Note that the HP disc and cassette drives do not implement the above features but these features will not affect existing programs in any way unless the program reads the tracks/surface etc. and expects

to see only those values that the disc or cassette drives generate (we have not seen any like this although some may exist).

The net effect to the user is that HPILDISC functions just as a tape in the HP82161A digital cassette drive or a disc in the HP9114A or HP9114B disc drives with the exception that the actual number of records available on the medium is a variable from 4 to 32,768 (i.e. 1,024 to 8,388,608 bytes) depending upon available space on the physical DOS disc drive and that unused records are not physically written to the DOS disc file HPILDISC. All of the discussion above regarding the internal buffers is fairly complicated and very few applications, if any, will be affected by it. The sole purpose of the internal buffers is to minimize actual disc read/writes and rely on the electronic type which is faster and causes less disc and drive wear. It does mean however that the disc should NOT BE REMOVED at any time during the time that the PC is emulating a drive.

NOTE - Buffers 0 and 1 below are NOT the internal buffers described above which are 6k and 12k bytes. Buffers 0 and 1 are 256 byte buffers from which data will be transferred to/from the interface loop. These buffers interact with the larger buffers but the user only "sees" buffers 0 and 1, not the internal buffers. Buffers 0 and 1 are provided for compatibility with the Digital Cassette Drive and 9114 disc drive.

DCL - Recording mode cleared, HPILDISC file pointer set to byte 0, record 0.

DDL - If listener, response determined by DDL number below:

0 : Subsequent data bytes are stored in buffer 0 starting at the byte pointer position. When the buffer fills (i.e. byte pointer is 255) the buffer is sent to the appropriate internal buffer according to the current recording mode (see DDL 2 and 6). Additional data bytes fill buffer 0 starting at byte position 0 and are recorded when the buffer fills or an End byte is received. {Write Buffer 0}

1 : Subsequent data bytes fill buffer 1 starting at the current byte pointer. When buffer 1 fills, the byte pointer is reset to 0 and additional data bytes replace those in buffer 1. This command also clears the Partial Record Mode set up by DDL 6. {Write Buffer 1}

2 : Sets continuous recording mode. Subsequent data bytes will fill buffer 0 starting at byte position 0. When the buffer fills the record is sent to the internal buffers and additional data bytes fill buffer 0 and are sent to the next record in the internal buffer. This continues until the maximum allowable record has been written or an End byte is received or DDL 8 is received. {Write}

3 : Subsequent data bytes reset the byte pointer from 0-255 but only the last one is used. {Set Byte Pointer}

4 : The next 2 bytes determine the record number to go to. The number is an integer with most significant byte first. For example, the bytes 2,0 would position the current record to be 512. This also clears Partial Record Mode set up by DDL 6. {Seek}

5 : If HPILDISC exists, it is erased and recreated. The current record is set to 0 and the Partial Record Mode set up by DDL 6 is cleared. {Format}

6 : Sets up Partial Write Mode. The current record is copied into buffer 0 and the current record does not change. Subsequent data bytes replace those in buffer 0 starting at the byte pointer position. When buffer 0 fills the data are sent to the current record in the internal buffers but the current record does not change. This continues until a Close Record command (DDL 8) is received or an End byte is received which cause buffer 0 to be copied to the current record. If the End byte fills buffer 0, the current record is incremented by one and the next record is copied to buffer 0. {Partial Write}

7 : Clears partial recording mode set up by DDL 6. {Rewind}

8 : The contents of buffer 0 are copied to the current record. Following a Write (DDL 2) the current record is incremented. Following a Partial Write (DDL 6) the current record is unchanged. {Close Record}

9 : Buffer 0 is copied into buffer 1 and the byte pointer is set to 0. {Copy Buffer 0}

10 : Buffers 0 and 1 are exchanged. The byte pointer is set to 0. {Exchange Buffers}

11-30 : Subsequent data bytes are ignored.

DDT - If talker, responses indicated below:

0 : A subsequent Send Data message causes buffer 0 to be sent on the loop from the byte pointer. When the end of the buffer is reached the next record becomes the current one and the data are copied to buffer 0. This continues until the maximum record is exceeded (sets status byte 17), an NRD is received, or a data transfer error is detected. This also clears the Partial Write Mode set by DDL 6. {Send Buffer 0}

1 : A subsequent Send Data message causes buffer 1 to be sent on the loop from the current byte pointer. This

continues until the end of the buffer is reached or an NRD is received. {Send Buffer 1}

2 : The next record is copied to buffer 0 and the current record is incremented. A subsequent Send Data message causes buffer 0 to be sent on the loop from the first cell in the buffer. When the end of the buffer is reached, the next one is copied to the buffer and sent. This continues until the maximum record is exceeded (sets status byte 17), an NRD is received, or a data transmission error is detected. This clears the Partial Write Mode set up by DDL 6. {Read}

3 : A subsequent Send Data message causes 3 bytes to be sent. The first two are the most and least significant bytes of the current record and the third is the byte pointer position. {Send Position}

4 : Same as DDL 10. {Exchange Buffers}

6 : Emulates a feature found in the HP9114 drive. Twelve bytes of data are sent to indicate the tracks/surface, number of surfaces, and the records/track. LINK allows these to be variable since it makes available all room on the DOS disc up to 32,768 records for a single HPILDISC. Bytes 1-4 indicate the records/track, 5-8 indicate the number of surfaces, and 9-12 indicate the records/track.

LINK sets the values upon the free disc space available including any used by HPILDISC and the numbers are therefore variable between runs of LINK. The program sets the number of surfaces to 2 if more than 4 records are available and the tracks/surface are set as follows: if more than 256 records are available per surface, the tracks/surface is MSB of the records per surface, otherwise it is 2. Note that LINK does nothing with this data other than report it. It is normally used by a handheld when it formats a tape/disc to put in the volume label section to override default values set up for the digital cassette drive since that device does not respond to this DDT command.

The HP-41 does not use this data. The HP-71 and 75 use it in a check of the available disc space while executing the COPY command.

7 : A subsequent Send Data command causes 2 bytes of data to be sent that represent the maximum number of records available for HPILDISC less 1. This DDT is not part of the Digital Cassette Drive but is part of the 9114 disc drive. For comaptibility with the 71 and 75, during

formatting this DDT is changed to 6 (i.e. after DDL 5 is received).

SST - Sends one status byte as follows depending on the current state:

0 : Everything okay.

17 : Current record exceeds the maximum allowable (i.e. "end of disc").

23 : Disc has not been positioned via DDL 4 (i.e. "new disc error")

24 : Disc has not been formatted (i.e. "no data on disc")

26 : General MS-DOS read/write error (i.e. "checksum error")

28 : Specified seek record greater than maximum allowable ("size error")

32 : Busy. This is not an error. There is a physical difference of operation between this emulator and the cassette or disc drives it emulates. They wait until receiving an RFC after a command (standard HP-IL protocol for RFC) to execute the previous command and can then respond to other HP-IL mesages while they are executing the specified command. This includes responding to a Send Status command which is why there is a status for "busy". However, this emulator holds up loop operation until the command is executed (other than letting IDY messages through) so by definition it cannot be busy when it responds to a Send Status command because it won't respond until it is not busy. However, for some reason the HP-41 and 71 handhelds expect at least one "busy" status response after issuing a DDT 2 command so it is included here. Other than that, LINK will never report a "busy" condition.

MESSAGES SPECIFIC TO PRINTER

SST - The printer will return two 0 bytes if no error is detected. Two bytes will be returned at any error condition the first of which is decimal 65 and the second is 0. If an error occurs on the PC with the printer, the error message will be displayed there. This includes not turned on, out of paper, etc.

DAB END - If a listener, the PC will send the character to the print device. Note that if the system printer is an HP-IL priner, an error will occur since the PC is trying to control the HP-

IL to make the printer print at the same time it is acting as a device.

MESSAGES SPECIFIC TO DISPLAY

SST - LINK sends three bytes. The first is 0 if no data have been put in the keyboard input bufer (i.e. no keys pressed or keyboard inactive) or it is 32 if data are ready to be sent from the input buffer. The second byte is the cursor column position (0-79) and the third is the cursor row (0-24 for all but the HP Portable which is 0-15).

IDY - If the keyboard is active and data are in the keyboard input buffer then the service request bit in the IDY frame is set.

SDA - If data are in the keyboard input buffer and the keyboard is active, each byte is sent on the loop until an error is detected, an NRD is received or the buffer is emptied. Otherwise the null string (i.e. no characters) is sent.

DDL - If 31 and the PC is a listener, see the discussion above regarding general responses to individual HP-IL commands.

DDT - If talker the response is as follows:

0 : Keyboard is removed from active status and input buffer is cleared.

1 : Keyboard is activated and input buffer is cleared.

DAB END - If the keyboard is active and keys are in the buffer, the service request bit in the data or end byte is set even if the PC is not a listener.

If the PC is a listener, the following apply.

For characters with ASCII codes from 32-127 (i.e. normal characters), the same character is displayed on the screen. If the character is in column 80, the cursor goes to column 1 of the next line. If the cursor is in column 80 of the bottom line of the screen, the screen "rolls up" one line and the cursor goes to column 1 on the new bottom line.

ASCII 8 causes a backspace.

ASCII 10 causes a linefeed.

ASCII 13 causes a carriage return.

If the ASCII value is between 128-255 the character with an ASCII number that is 128 less is displayed in inverse video. What inverse video looks like is a function of the type of screen you are using, whether it has color or not, etc. For example, since 193 - 128 = 65 and 65 is ASCII character "A", if the PC receives ASCII 193, inverse "A" is displayed.

ASCII 27 is the escape character (denoted below as "ESC") and initiates an escape sequence for screen control depending on the next character received. If the next character is not in the tables below, both it and the ESC are ignored.

ESC A -Cursor moves up one row stopping at row 0.

ESC B - Cursor moves down one row stopping at row 15 on the HP Portable and row 24 on all others.

ESC C - Cursor moves right one column. When it gets to column 79, it goes to column 0 of the next row. When it gets to the bottom right corner it will then go home (row 0, column 0).

ESC D - Cursor moves left one column. If it is in column 0, it moves to column 79 of the previous row. It stops in row 0, column 0.

ESC E - Clears screen, homes cursor, blink off, color yellow.

ESC H -Homes cursor (row 0, column 0).

ESC J - Clears screen from cursor position down.

ESC Q - Sets cursor to underscore.

ESC R - Restores cursor to a box (default).

ESC S - Screen rolls up one line.

ESC T - Screen rolls down one line.

ESC < - Turns cursor off but it still functions normally.

ESC > - Turns cursor back on.

ESC % X Y - Sets cursor position. X is a character whose ASCII numberdetermines the column (0-79) and Y is a character whose ASCII number determines the row (0-24 or 15 for HP Portable). Out of range protection is provided by using X mod 80 and Y mod 25 (or Y mod 16 for the HP Portable).

NOTE - The spaces above (such as the one betweenn ESC and A) are NOT to be sent. They are only for ease of reading.

On IBM PC's and compatibles, text color can be set using the following two-character escape sequences. They are ignored on the HP Portable/Plus and HP 150. Note that black is always the background color.

ESC a - Black (i.e. invisible)

ESC b - Blue

ESC i - Dark Gray

ESC j - Light Blue

ESC c - Green	ESC k - Light Green
ESC d - Cyan	ESC l - Light Cyan
ESC e - Red	ESC m - Light Red
ESC f - Magenta	ESC n - Light Magenta
ESC g - Brown	ESC o - Yellow (default)
ESC h - Light Gray	ESC p - White

The background color is set according to the following:

ESC q - Black (Default)	ESC u - Red
ESC r - Blue	ESC v - Magenta
ESC s - Green	ESC w - Brown
ESC t - Cyan	ESC x - Light Gray

The response of a given PC to color will depend upon monitor type and video adapter. On an IBM PC, the light colors are slightly brighter than the "regular" colors. On a Compaq, the colors get brighter from the top left (ESC a) to the bottom right (ESC p). For all monochrome monitors, blue and light blue will cause underline.

Blink is set/cleared on all PC's as follows:

ESC y - Blink On	ESC z - Blink Off

MESSAGES SPECIFIC TO DOS DEVICE (including files)

SST - Sends one byte depending on the status. A non-zero value indicates that an error condition has occurred. The non-zero value is the DOS Error Code that results from DOS function calls. The meaning of the error is dependent upon the device. Refer to the DOS Technical Reference Manual for the exact meaning of this byte. A summary table is given below for some errors which can occur:

DOS Error Number	Meaning
2	File not found
3	Path not found
4	Too many open files
5	Access denied
6	Invalid handle
7	Memory control blocks destroyed
8	Insufficient memory
9	Invalid memory block address
12	Invalid access code
15	Invalid drive was specified

16	Attempted to remove the current directory
17	Not same device
18	No more files

Several of the above are very unlikely to occur. For example, error 2 is unlikely since if a file is specified which does not exist, it will be created by LINK.

NOTE - If you are using DOS 3.0+ it is possible although unlikely that you could get a number greater than 18 since DOS 3.0+ has 88 error codes. Most higher-numbered ones deal with network access and some really nasty ones that LINK does not trap (examples from the DOS Tech. Ref. Manual are : "General Failure", "Network name not found", "Too many redirections", etc.). If you get an unusual error code, refer to pp.6-40 and 41 in the Tech. Ref. Manual for IBM PC-DOS 3.10). In general, if you encounter an error, you should either retry or abort, regardless of the error.

DDL - If the PC is a listener, the response is as follows for various DDL's:

0 : If the PC is a DOS Device, "raw" mode is disabled. Raw mode is faster than ASCII mode because ASCII mode checks each character to see if an End-of-File (Ctrl-Z), Ctrl-s, Ctrl-P, and/or Ctrl-C has occurred. Note that LINK initially puts all devices in raw mode at startup and returns it to whatever state it was in at termination. This does nothing for DOS files.

1 : If the PC is a DOS device, "raw" mode is enabled. This does nothing for DOS files. This is LINK's initial setting.

2-30 : No response. Reserved for future versions.

31 : See the discussion above regarding programmable restart of LINK using DDL 31.

DDT - If a DOS device and DDT 0, block mode is disabled. If a DOS device and DDT 1, block mode is enabled. Note that your device may or may not use HP's block mode check character (refer to Chapter 2).

NOTE - LINK automatically sets the COM1, COM2, and AUX devices on the HP Portable and HP 150 computers to block mode enabled.

DCL - Input/output buffers are cleared. Device is not remote enabled.

REN - Remote Enables device.

NRE - Makes file/device not Remote Enabled.

GTL - If a listener, makes it not Remote Enabled.

DAB - If a listener, the response is given below:
END

If the PC is not Remote Enabled, the data byte is sent to the device or file on a byte-for-byte basis.

If the PC is Remote Enabled, data bytes are accumulated until a carriage return/linefeed (CR/LF) is is received (up to 255 characters). The response after CR/LF is received depends upon whether the device is a physical device or a file. If it is a physical device, the entire string is sent to the PC device in a control string. If it is a file, the last 3 characters (not including CR/LF and with ASCII 0's filling from the left side if fewer than 3 characters) are sent to the PC to set the current file pointer. Note that if the string is sent as "EOF", "eof", "Eof", etc. it is interpreted to mean move to the end of the file. The bytes are multiples of 256, with the most significant first. For example, if the third character from the end is ASCII 1 then the file pointer would be set to (256*256*256)-1 and if only the last character was non-zero it would be in the range of 0-255. Note that file position starts with 0, not 1.

If the PC is Remote Enabled and a file is being accessed, sending "CUT" in upper or lowercase truncates the file at the current file pointer.

SDA - If the PC is a talker, the response is given below:

If Remote Enabled, the response depends upon whether the PC is accessing a physical device or a file. If it is a physical device, the PC polls the device for device I/O control information and if anything is returned, it is sent on the loop until NRD, a transmission error or the end of the string is sensed. If it is a file, 3 bytes are sent (with no CR/LF) which indicate the current position of the file pointer). The equivalent decimal values are the same as discussed above under DAB except that "EOF" and "CUT" are not used.

If not Remote Enabled, the file or device is read from and data are sent on the loop until NRD is received, a transmission error, or the device quits feeding the PC characters.

APPENDIX B
Live Keyboard Data

In the main documentation is a discussion of the use of your PC's keyboard for data entry to your handheld which occurs when you select the video display emulator and also in the device mode with the CON device. This appendix covers the types of ASCII characters which will be returned when you press keys in these modes since several of the keys do not return normal ASCII characters.

The most obvious examples of those keys that do not return ASCII characters are the function keys. On the IBM PC these do not include F10 since LINK continuously checks input keys to see if F10 has been pressed to determine if it should restart. For the rest of the function keys, the response of LINK is determined by whether you have redefined the keys to return other strings when you press them. Any decent application should not leave these keys defined when it ends so the problem of unintentional key assignments hanging around when LINK starts is minimal but it can exist. If it is a problem, refer to your PC's owner's manual for the techniques to redefine function keys and reset the definitions (HP computers use escape sequences and IBM PC's use ANSI.SYS functions if ANSI.SYS is in the configuration file).

Other than F10 on the IBM PC, LINK does not change any key assignments (or labels on HP PC's although the labels have been turned off) so that if you want to you can redefine them to be anything you want which can be a typing aid when you are using the PC as a remote keyboard for the 71 or 75. An example is given below for the HP 150 and HP Portable/Plus (not IBM PC's) for a batch file (.BAT) that when executed would redefine the F1 key to be a typing aid for a catalog listing of RAM files in either the 71 or 75. The items in the brackets { } are comments and should not be entered.

```
ECHO ^[&f0a1k7d7LCAT ALLCAT ALL  {Label and string are set to
                                  CAT ALL for F1}
C:\LINK                          {Invoke LINK assuming it is in C:\}
ECHO ^[&f0a1k0d2L^[p             {Reset F1 to ESC p which is the
                                  default}
```

Please note that the symbols "^[" are the DOS representation of the escape character that you get by pressing the ESC key followed by &

(shift 7). Also, if you want to see your functin key labels at any time during LINK, it can be toggled on/off with the "Menu" key.

NOTE- If you normally use F10 on your IBM PC as a redefined key in other applications you should NOT have it redefined when LINK runs or LINK will "see" the redefined character string and not the default string so that the only way out of the program is to reset the PC. With an HP PC you cannot redefine the SELECT key anyway so this is not a consideration.

"Normal" keys on your keyboard are treated as single characters in LINK. For example, if you press the "1" key, LINK puts the character "1" into its input buffer. Caps lock is functional so if you want all of the letter keys to be in uppercase, use that key to toggle it on/off.

On the following page are tables for selected keys that do not produce normal characters such as cursor keys, tab, backspace, etc. The function keys given are based upon no key reassignments.

LINK generates two-character sequences for the keys below. The first is the escape character (ASCII 27) and the second is given from the table below. The numbers in the table refer to standard ASSCII numbers (e.g. A = 65).

HP 150 and Portable/Plus Keys

F1	112	(p)	ClrLine	75	(K)	Home	104	(h)
F2	113	(q)	ClrDsp	74	(J)	End	70	(F)
F3	114	(r)	+Line	76	(L)	Prev	86	(V)
F4	115	(s)	-Line	77	(M)	Next	85	(U)
F5	116	(t)	+Char	81	(Q)			
F6	117	(u)	-Char	80	(P)			
F7	118	(v)						
F8	119	(w)						
Left	68	(D)	Tab	73	(I)			
Right	67	(C)	ShiftTab	105	(i)			
Up	65	(A)	BackSpace	8				
Down	66	(B)						

IBM PC Keys

F1	59	(;)	Left	68	(D)	Home	104	(h)
F2	60	(<)	Right	67	(C)	End	70	(F)
F3	61	(=)	Up	65	(A)	PgUp	86	(V)
F4	62	(>)	Down	77	(B)	PgDn	85	(U)
F5	63	(?)						
F6	64	(@)	DEL	80	(P)	Tab	73	(I)
F7	65	(A)	INS	81	(Q)	ShiftTab	105	(i)
F8	66	(B)				Backspace	8	
F9	67	(C)						

If you need ASCII characters from 0-31, hold down the CTRL key and press the key corresponding to an ASCII number which is 64 higher. ASCII 1 is CTRL @, ASCII 2 is CTRL A, etc. CTRL M is the same as the Return key, CTRL I is the same as the Tab key, and CTRL H is the same as the Backspace key.

Also, the Return key generates just ASCII 13 (i.e. CR) in video mode but in device mode it returns CR/LF since MS-DOS uses CR/LF as an end-of-line indicator and the keyboard is treated like other files/devices in this mode.

In device mode LINK returns ASCII 9 for Tab and ASCII 8 for backspace.

If you have a need to use other special keys but you do not know what key or keys will be returned, you can use the following program to find the ASCII numbers of a key on the 71 or 75 assuming that the PC is NOT the KEYBOARD IS device but it is either in video display mode or device mode using CON as the DOS device. This assumes the handheld has set the PC as the DISPLAY IS device.

HP-71

```
4000 SUB KEYCODE
4010 I=FLAG(-23,1) ! Set flag -23 and save old setting
4020 ENTER ':DISPLAY';A$ @ IF A$="" THEN 4020
4030 IF A$=CHR$(3) THEN GOTO 'END' ! Exit if Ctrl-C pressed on PC
4040 FOR J=1 TO LEN(A$) @ DISP NUM(A$[J]); ! Show ASCII numbers
       of string
4050 NEXT J @ DISP @ GOTO 4020
4060 'END': I=FLAG(-23,I) @ END ! Reset flag -23 and quit.
```

HP-75

```
4000 ENTER ':D1' USING '#!%,K';A$ ! For LINK as DOS device use
       :I1
4010 IF A$="" THEN GOTO 4000
4020 IF A$=CHR$(3) THEN DISP 'Done' @ END
4030 FOR I=1 TO LEN(A$) @ DISP NUM(A$[I]); ! Show ASCII numbers of
string
4040 NEXT I @ GOTO 4000
```

Both of the programs above will continue to display the ASCII numbers of each string until Ctrl-C is pressed on the PC. The 71 program affects no variables and resets flag -23 to its original value.

Both the 71 and the 75 have the capability to "map" the special two-character escape sequences generated by LINK into whatever keys you want on the handheld. Typically these will be the editing keys so that backspace, character replacement, etc. can function by your pressing the keys you normally do on the PC. However the mapping is not limited to that. You can have any two-character sequence generated by LINK map into any key on the handheld, even those that you have redefined.

For the 75, you have to edit a text file named KEYMAP and place in it the mapping information as described in the Extended I/O module on page 91. Below is a sequence of lines you can use to put some typical editing keys from the PC into the KEYMAP file. Items in { } are our comments.

```
EDIT 'KEYMAP', TEXT
132 A    {Up arrow is up arrow on PC}
133 B    {Down arrow is down arrow on PC}
134 C    {Left arrow is left arrow on PC}
135 D    {Right arrow is right arrow on PC}
136 Q    {I/R toggle is INS on IBM and +Char on HP}
138 P    {DEL is DEL on PC and -Char on HP}
142 I    {TAB is Tab on PC}
139 K    {CLR is ClrLine on HP PC}
166 h    {Shift-LeftArrow is Home on PC}
167 F    {Shift-RightArrow is End on PC}
164 V    {Shift-UpArrow is PgUp on IBM and Prev on HP}
165 U    {Shift-DownArrow is PgDn on IBM and Next on HP}
```

With some modifications to the above file you can make whatever keys on your PC totally suit your 75 editing needs. Note that the 75 key "BACK" has no representation and cannot be mapped into.

On the 71 you use the ESCAPE command to map incoming escape sequences. The following program could rerun after LINK is running to switch it to video display mode and then to remap the PC keys into their 71 equivalents.

```
5000 SUB USEPC
5010 FINDPC(A,'V')          ! See the 71 chapter for a description of this
                              subprogram.
5020 KEYBOARD IS :A @ DISPLAY IS :A
5030 ESCAPE 'A',50          ! Up arrow is up arrow on PC
5040 ESCAPE 'B',51          ! Down arrow is down arrow on PC
5050 ESCAPE 'C',48          ! Right arrow is right arrow on PC
5060 ESCAPE 'D',47          ! Left arrow is left arrow on PC
5130 ESCAPE 'K',107         ! -LINE is -Line on HP
5070 ESCAPE 'Q',105         ! I/R is INS on IBM and +Char on HP
5080 ESCAPE 'P',104         ! -Char is DEL on IBM and -Char on HP
5090 ESCAPE 'h',159         ! g-LeftArrow is Home on PC
5100 ESCAPE 'F',160         ! g-RightArrow is End on PC
5110 ESCAPE 'V',162         ! g-UpArrow is PgUp on IBM and Prev on HP
5120 ESCAPE 'U',163         ! g-DownArrow is PgDn on IBM and Next on HP
```

```
5140 ESCAPE CHR$(8),103! BACK is BackSpace on PC (deleting, not just
                         cursor)
5150 ESCAPE 'I',102      ! SST is now Tab on the PC (optional)
5160 ESCAPE 'i',109      ! USER is now ShiftTab on the PC (optional)
5170 END
```

Whenever you are finished you can execute "RESET ESCAPE" to cancel all key definitions. If you want to cancel one of them, just set it to character 0 (e.g. ESCAPE 'A',0 removes ESC A from being mapped into a 71 key).

APPENDIX C
Future Plans

We believe that LINK will greatly increase the applications for all of the handhelds if for no other reason that the number of devices available to the handheld community is limited only by the advances in the MS-DOS world. Southern Software is commited to providing the tools necessary to allow this to happen. We will provide the following services:

1 - We will be available for technical consultation on those items pertaining to the goal of LINKing the handhelds and the MS-DOS family. We do not pretend to have all of the answers but if we cannot help directly then perhaps at least we can steer you in the proper direction or to the proper people. If you call us with questions or problems please have the following available:

   a) handheld you are using (e.g. 41CX, 75D, etc.)
   b) modules currently in the handheld
   c) program listing or copy of keystrokes being attempted
   d) handheld settings (flags, variables, loop assignments, etc.)
   e) PC which is running LINK (e.g. Compaq, HP 150, etc.)
   f) pertinent devices connected to the PC (no. of disc drives, ports, etc.)
   g) owners manuals (as many as you can have your hands on)

   Since we are dealing with two completely different systems (handheld and PC) the potential problems can be very complex and the more data you have on hand when you call the lower your phone bill will be.

   We do not want to get into normal program development for handheld programs but will assist where possible in those aspects where the PC and the handheld machines are LINKed.

   Southern Software can be reached by phone or mail at the following:

   Southern Software
   2853 Ramsey Drive
   New Orleans, La. 70114
   504/391-9465

   Telephone calls will be taken from 7am-7pm Central Time, Monday-Saturday. Ask to speak to Mr. Paul Grimmer, the author

of LINK. Also please fill out the accompanying registration card with your comments. We will respond to as many questions and comments as possible as well as alert you to upgrades to LINK.

2 - Heaven forbid that there are any bugs in LINK but should you find any bugs let us know and we will fix it as soon as possible, and then send you a free copy of the corrected program. A bug is defined as something that does not work as this documentation states but is not just a feature you would like to see implemented.

3 - If there are features you would like to see implemented on future versions of LINK please let us know of it and we will evaluate it. If we decide to implement it you will get a free copy of the new version. Some of the things currently under consideration include:

a) graphics capability in the video display emulator. The various PC's that LINK is able to run on all do graphics very differently so this is more of an effort than it appears. This exists now in a crude form for those PC's that have escape sequence graphics capability (use the device mode of LINK and specify the CON device) but no standard currently exists.

b) user-definable accessory ID's. This might be useful in patching LINK with existing applications that look for a specific accessory ID. For example you have access to your PC plotter but your software may look either for the plotter directly on the loop or look for an RS-232 interface. If your software is not in ROM it is relatively easy to modify but if it is in ROM then it may not be quite so easy to get it to work...

c) a utility to extract any file from HPILDISC and copy it to an MS-DOS file. We are uncertain as to the true utility of this since LINK allows you to write anything you want to an MS-DOS file directly so its use would be restricted to those types of files that you cannot make copyable representations of (i.e. ASCII or hex) which would be key assignments, etc.

d) a translator that would intercept HP-type printer controls and convert them into Epson, IBM, etc. printer codes so that your programs written for HP printers would run using the PC printer. Some HP functions have no counterparts on the other printers and the way HP does graphics is very different from the other major printer manufacturers so this is much more difficult than it appears on the surface.

e) access to DOS and BIOS calls. Implementation of this is fairly simple but the documentation requirement would be horrendous. It is also along the same lines as opening Pandora's box.

4 - Until software marketers for handheld products can begin supplying their programs on MS-DOS discs for those of you who have LINK without an HP disc drive, we will be glad to take the HP disc that came with your software and copy it to an MS-DOS disc for you. Send us the HP disc, a formatted DOS disc and $5. If we have any public domain "goodies" for your machine we will put a few of them on the disc also.

One application for LINK that has not previously discussed concerns the use of bulletin boards which are widespread for MS-DOS PC's but non-existent for the handhelds. With the disc drive emulator that LINK provides this barrier is broken. Any and all handheld files can be in your HPILDISC file and since it is an MS-DOS file it can be passed around bulletin boards. The only requirement is that the BBS provides 8-bit file transfer using XMODEM, Kermit, etc. and most do. Also, handheld program documentation could easily be put on the DOS disc in the same subdirectory that holds the HPILDISC file with the program(s) which should have a net effect of lowering distribution costs for handheld programs since printing costs can be eliminated.

APPENDIX D
More From Southern Software

SIDEWINDER $69.95 (Utilities $10)

For the same PC's as LINK, SIDEWINDER enables sideways printing of text files such as Lotus 1-2-3 print files, word processing files, etc. It is specifically for HP printers such as the ThinkJet and QuietJet. Text can be printed at either 6 or 8 lnes/inch which is up to 53 lines/page on the ThinkJet or 105 on the QuietJet. The character set is as close as possible to the built-in normal character set and not some arbitrary set that has to fit all types of printers as for other commercially-available packages. SIDEWINDER recognizes standard HP printer controls for printing double wide and underline and a utility is included for easily inserting these in your text (did you know that 1-2-3 ignores your setup specification when you print to a file?).

A utility is available to enable you to define your own characters. Any that you redefine will override the built-in set that SIDEWINDER has. With this you can make scientific characters (delta, integral signs, etc.), foreign characters, or italics. Your imagination is the limit!

It is the only sidways printing program available for the HP Portable/Plus and is the only one that will address the HP-IL on the IBM PC family through the Portable/Desktop link.

LIFDOS $60

This program enables movement of files from an HP disc which has been used by a series 40, 70, 80, 12x, 200, etc. (called LIF for Logical Interchange Format) to MS-DOS. It runs on the same PC's as LINK. We have a working version now and are finishing the documentation.

GRPRINT $60

Where SIDEWINDER allows you to create your own sideways character sets GRPRINT allows you to graphically create your own

normal printing character sets. Whatever character you redefine will be printed instead of the built-in character for the ThinkJet, QuietJet and LaserJet printers. In addition, a utility is provided which will allow these characters to be downloaded to the QuietJet.

The program includes other features such as header/footers, wide, bold, underline, mail merge, auto page numbering, etc. to facilitate a wide range of printing needs. It will be ready in Novemeber 1986 for the same PC's as LINK.

PLOT $60

The only plotting package for the HP-71. It uses the ThinkJet, QuietJet or LaserJt printers. You can plot up to 5 sets of data points, each with its own symbol, or up to 10 user-defined functions (via subprograms you define) or both. Plots can be linear or semi-log (from 1-3 cycles). Provides several new 71 functions via LEX keywords.

DED $50

A Data EDitor for the HP-71 which fits well with the plotting package or any other application that can use DATA files in standard HP format. Allows up to 50 labelled columns of numbers and an unlimited number of rows. Includes extensive editing functions and enables movement of columns/rows, sorting, printing, etc. Also gives you 6 new functions for the 71 for working with DATA files (LEX keywords).

FIT $50

A curve fitting program for the HP-71 which has a simplified version of the above data editor. It fits linear, log, power and exponential curves and will pick the best one for you. It is much faster than the Curve Fitting ROM and is much easier to use. Easy calculation of actual vs. observed points.

OTHER HP-71 PROGRAMS

All of the programs below are for the HP-71 and are in the User's Library and have been written by Southern Software. The User's Library Catalog number is given in brackets. All are available on a 5" DOS disc (i.e. an HPILDISC to use with LINK), 3" DOS disc (HPILDISC formatted for HP PC's), 3" HP-formatted disc, or

cassette tape as are all of the above. Please specify when ordering. Prices are 1 for $25, 2 for $45, and $15 for each over 2 in a single order.

TEXT File Editor {#3357}

A comprehensive TEXT file editor. Use it to keep notes, write memos, etc.

PROPERTIES OF PETROLEUM FLUIDS {#3359}

Duplicates the calculations of the HP-41 Petroleum Fluids Pac. All of the inputs, outputs, and calculations are in subprograms for use by other programs.

TUBING EROSIONAL VELOCITY CALCULATIONS {#3356}

Petroleum engineering program that calculates the minimum allowable pipe size or the maximum allowable flowrate for two-phase in pipe.

INPUT/OUTPUT SUBPROGRAMS {#3360}

A set of I/O subprograms that greatly simplify writing programs that may be run by yourself or others under a wide variety of peripheral connections on HP-IL. Your programs will automatically optimize input prompts, menus, output, titles, etc. based upon the current HP-IL configuration (or no loop) detecting and using printers and/or video displays.

DIRECT EXECUTION OF HP-41 PROGRAMS FROM TAPE/DISC {#3471}

Uses the HP-41 Translator Pac and allows you to read HP-41 program files directly from the tape or disc that they are on and converts them into the 71 FORTH environment provided with the Translator Pac. As an aside, a large complaint about PC's is that they are not good calculators. With LINK's KEYBOARD IS option and the Translator Pac your PC acts like an HP HP-41!

FLANGE TAP ORIFICE CALCUATIONS FOR NATURAL GAS {#3422}

Calculates flowrates, orifice sizes, and differential pressures for natural gas orifice meters using industry-standard correlations.

GAS PIPELINE HYDRAULICS {#3425}

Calculates flowrates, pressures, line size, etc. using either the Weymouth or Panhandle equations for natural gas flow.

MASS STORAGE UTILITIES {#3423}

Enables total access to your disc/tape drive. Sorted catalog listings, restoration of purged files, making private files unprivate, set/change voume label, duplication of tapes/discs, and editing byte-by-byte are some of the options available.

CATALOG AND FILE LISTING UTILITIES {#3460}

Three utilities that enable 1)time and date program listing of program and key files to the ThinkJet printer with wrap and perforation skip, 2)listing of RAM catalog to video display with available memory, and 3)time and date stamped catalog listing of tape/disc with volume label on the ThinkJet.

TEXT FILE PRINTER {#3470}

If you have a TEXT editor this program will give you total control over printing to your ThinkJet. You just embed your controls in the text (e.g. \U is underline) and the program will print with margins, headers/footers, wide, bold, underline, subscripts, centering, page numbering, etc. You can even make your own graphics characters. Has automatic wordwrap so you don't have to worry how long each line of text is in your file.

CURVE FITTING ROM UTILITY {#3488}

If you have HP's Curve Fitting ROM, you'll get three additional features that it needed: 1)the coefficients of the fit are placed in 71 variables or in a file for your use, 2)the process of predicting new y-

values for given x-values is greatly simplified, and 3)a table of observed versus calculated y-values can be printed or displayed.

CATALOG UTILITY {#3484}

Give the utility the filetype you want (BASIC, FORTH, etc.) and it will display or print a catalog of all files of that type in RAM and/or tape/disc. Especially useful when you have many different file types on a disc or tape.

PIANO {#3485}

This program configures the 71 as a musical keyboard with 3 octaves (with sharps/flats) available at a time out of 7 total. You have easy control over pitch and length. An editor is provided with playback and merge capabilities. This requires NOISE and KEYWAIT LEX functions but a utility is provided which POKEs them into memory. A simplified version for children is included.

EduCALC
27953 CABOT ROAD
LAGUNA NIGUEL, CA 92677 U.S.A.