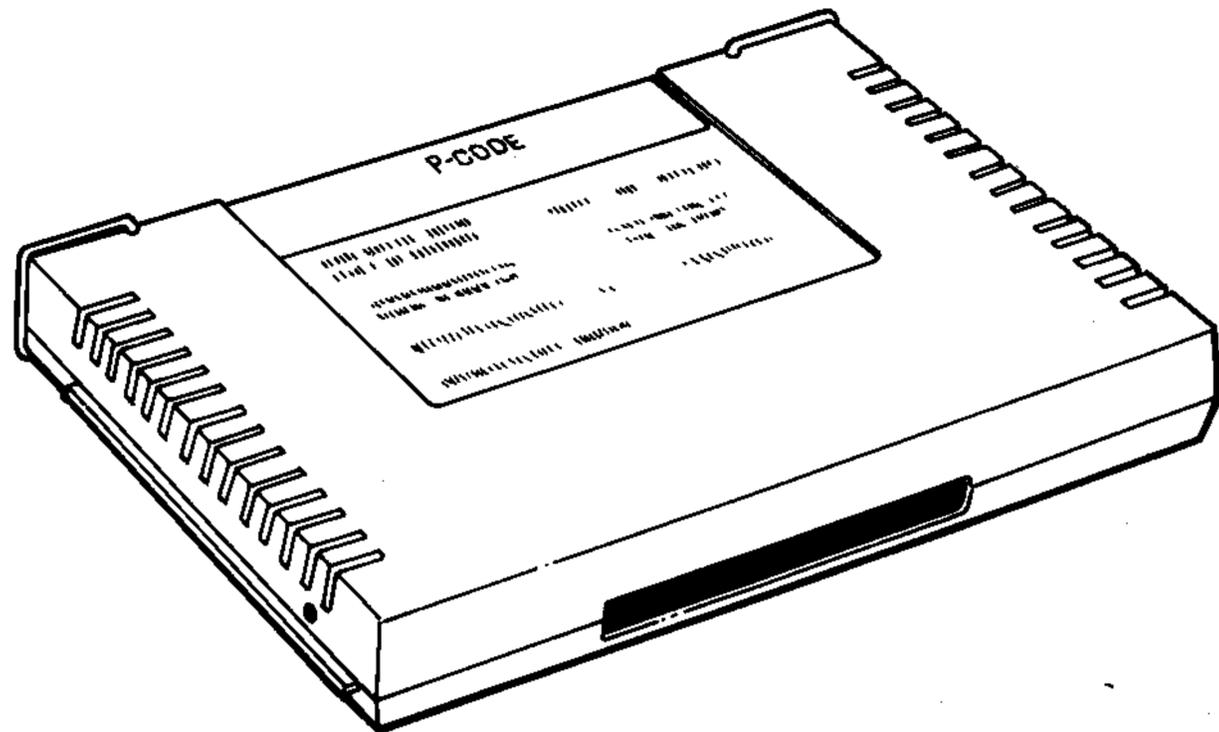


UCSD p-System*

P-Code Card

Model No. PHP1270



IMPORTANT

Record the serial number from the label on the P-Code Card and the purchase date in the space below. The serial number is identified by the words "SERIAL NO." printed on the label. Always reference this information in any correspondence.

PHP1270	115277	11/14/83
Model No.	Serial No.	Purchase Date
	ATA5282	

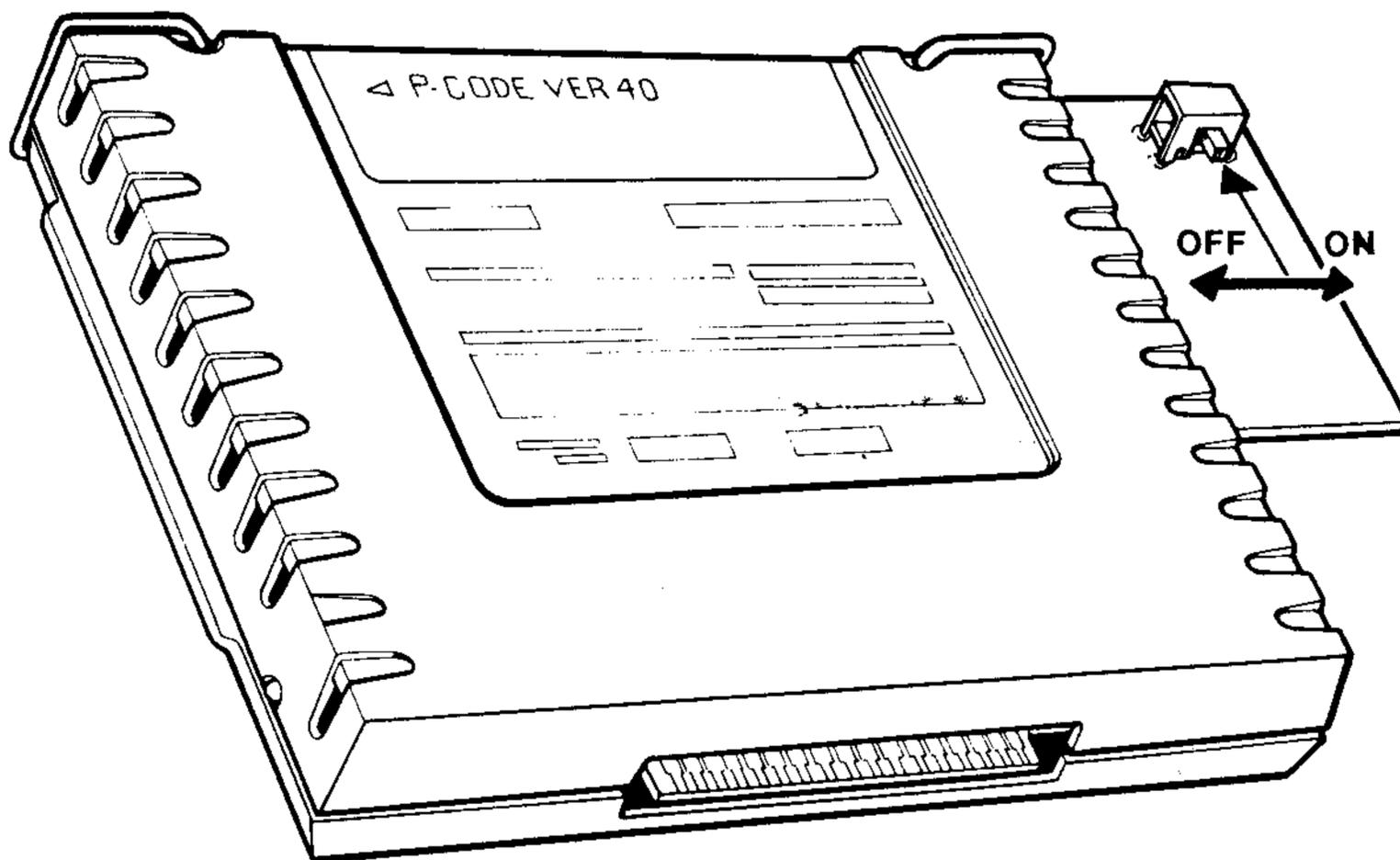
ADDENDUM

P-Code Peripheral Card ON/OFF Switch

An **ON/OFF** switch has been added to the P-Code Peripheral Card. This switch allows you to turn the card off without physically removing it from the Peripheral Expansion System unit.

When the card is switched **ON**, the console powers up in the UCSD p-System* mode, with the Command promptline displayed at the top of the screen. If the switch is **OFF** when the console is turned on, the normal title screen appears on the display.

The slide switch is located on a tab on the back of the P-Code card. The tab extends through the opening in the back of the expansion unit when the P-Code Card is in place. To turn the card **ON**, slide the switch toward the back edge of the tab. To turn the card **OFF**, slide the switch toward the Peripheral Expansion System unit.



*UCSD p-System is a trademark of the Regents of the University of California.

This manual was developed by staff members of the Texas Instruments Education and Communications Center.

This software is copyrighted 1979, 1981 by the Regents of the University of California, SofTech Microsystems, Inc., Texas Instruments Incorporated, and other copyright holders as identified in the program code. No license to copy this software is conveyed with this product. Additional copies for use on additional machines are available through Texas Instruments Incorporated. No copies of the software other than those provided for in Title 17 of the United States Code are authorized by Texas Instruments Incorporated.

UCSD Pascal and UCSD p-System are trademarks of the Regents of the University of California. Item involved met its quality assurance standards applicable to Version IV.0.

TABLE OF CONTENTS

GENERAL INFORMATION

1.1 Set-Up Instructions 4
 1.1.1 Inserting the P-Code Card 5
 1.1.2 Testing the p-System 6
 1.1.3 Removing the P-Code Card 6
 1.1.4 Using the TIDisk Memory System 6
1.2 Special Keys 7

USING THE SYSTEM COMMANDS

2.1 Promptlines 9

P-SYSTEM COMMANDS

3.1 A(ssem) 11
3.2 C(omp) 11
3.3 E(dit) 11
3.4 F(ile) 12
3.5 H(alt) 12
3.6 I(nitialize) 12
3.7 L(ink) 12
3.8 M(onitor) 13
 3.8.1 B(egin) 13
 3.8.2 E(nd) 13
 3.8.3 A(bort) 13
 3.8.4 S(uspend) 13
 3.8.5 R(esume) 13
3.9 R(un) 14
3.10 U(ser Restart) 14
3.11 X(ecute) 14
 3.11.1 Prompts for Filenames 14
 3.11.2 Execution Option Strings 15

LOADING AN EXISTING PROGRAM

4.1 Loading a Diskette-Based Program 18
4.2 Loading a Cassette-Based Program 18

APPENDICES

5.1 Appendix A—Execution Error Messages 19
5.2 Appendix B—I/O Error Messages 19

SERVICE INFORMATION

6.1 In Case of Difficulty 19
6.2 Exchange Centers (Local Service Options) 20
6.3 If You Have Questions or Need Assistance 20

THREE-MONTH LIMITED WARRANTY 21

SECTION 1: GENERAL INFORMATION

The Texas Instruments P-Code Card allows the computer to access the UCSD p-System* and a variety of programming languages, including UCSD Pascal*, BASIC, and Pilot. The P-Code Card is designed to work with the TI Home Computer, a TI Color Monitor (or the TI Video Modulator attached to a television set), the TI Peripheral Expansion System, the TI Memory Expansion Card, the TI Disk Memory System with up to three disk drives, and a diskette or cassette tape that contains source programs, object programs, raw data, and/or document text. A printer can also be attached to your system to increase its capabilities and applicability.

The P-Code Card allows your computer to execute programs written in several high-level languages. With the P-Code Card, the UCSD p-System's high-level languages, such as Pascal and BASIC, are compiled to an intermediate language called pseudo-code or p-code. The P-Code Card interprets the p-code instructions, which are then executed by the computer.

The simplest configuration for running the UCSD p-System requires the TI Home Computer, a monitor, the TI Peripheral Expansion System, the Memory Expansion Card, the P-Code Card, and either a cassette recorder or the Disk Drive Controller Card with a Disk Memory Drive. With this equipment you can either develop programs of your own or run existing programs. **Note:** To develop a program, the Disk Memory System (the Disk Drive Controller Card and up to three Disk Memory Drives) must be attached to your system.

After the System is "booted," or turned on, a promptline of the currently available System Commands appears at the top of the display. (Booting, or bootstrapping, the System simply means turning it on.) The System Commands are the communications interface for interacting with the various System functions. Each command is accessed by a single keystroke. Typing a key causes either an action to be performed or another promptline to be displayed, detailing new commands available at a different level.

The Editor, Filer, Compiler, Assembler, Linker, and Utilities capabilities (all sold separately) require the use of the TI Disk Memory System and a diskette containing the program.

1.1 SET-UP INSTRUCTIONS

The steps involved in inserting the P-Code Card into the Peripheral Expansion System and then checking its operation are included in this section. Please read this material completely before proceeding.

Note: The Peripheral Expansion System has eight slots into which accessory cards can be inserted. The Peripheral Expansion Card must occupy slot number 1. (For information on setting up the peripheral system, refer to the Peripheral Expansion System owner's manual.) If you have a Disk Memory Drive in the Peripheral Expansion System, the Disk Drive Controller Card must be in slot number 8. Other cards can be inserted in any of the remaining slots.

CAUTION

These electronic components can be damaged by static electricity discharges. To avoid damage, do not touch the connector contacts.

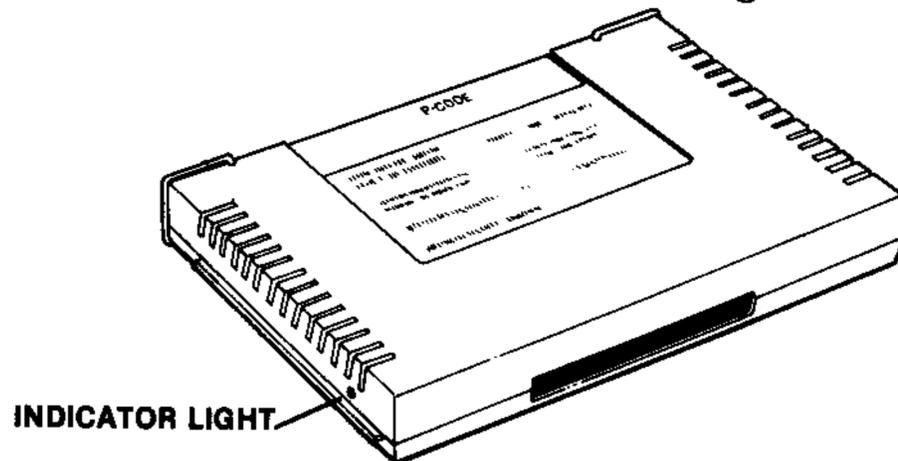
Once you've unpacked the unit, you're ready to insert the P-Code Card into the Peripheral Expansion System. (Save the packing material for storing or transporting the unit.)

* trademark of the Regents of the University of California

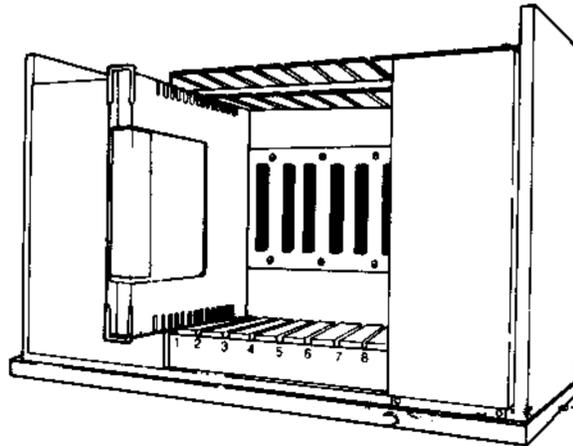
P-Code Card

1.1.1 Inserting the P-Code Card

1. First turn off the computer console and all attached devices.
2. **WARNING: TO AVOID DAMAGING ACCESSORY CARDS, WAIT TWO (2) MINUTES AFTER TURNING OFF THE UNIT FOR THE POWER TO DISCHARGE BEFORE PROCEEDING.**
3. Remove the top from the peripheral system by lifting the back edge of the top and pulling up.
4. The label identifying the P-Code Card is on the top of the card. On the front of the card is an indicator light that is seen from the front of the peripheral system when the card is active. Hold the P-Code Card with the indicator light facing the front of the peripheral system.



5. Carefully align the card with the desired slot and press the card firmly down into the slot.



6. Make sure the Memory Expansion Card is inserted in the peripheral system.
7. Replace the top on the Peripheral Expansion System by sliding the front edge under the extension on the front of the unit and firmly pressing down on the back edge of the top. Do not run the system without the top in place because the top ensures proper ventilation. **Note:** If the top does not fit properly, remove the card and realign it in the slot, remembering to press down firmly until the connection is made.

CAUTION

Always disconnect the Peripheral Expansion System before moving the computer console. The cable connecting the console and peripheral system is not designed to support the weight of the units. To prevent damage, always disconnect all devices before moving any part of your Home Computer system. For long distance moves, remove all cards from the Peripheral Expansion System. Then, repack the devices in their original packing material.

1.1.2 Testing the P-Code Card

1. The power switch is located on the front of the Peripheral Expansion System in the lower left-hand corner. Turn on the peripheral system, monitor, and console in that order.
2. A light should briefly come on in the position where you have inserted the P-Code Card. Each time the computer system accesses the P-Code Card, its corresponding light comes on. Note that the intensity and duration of the light varies, depending on the operation being performed within the system.
3. The p-System now begins its power-up sequence which requires 30-60 seconds. You will hear six beeps. The first beep indicates that the console is turned on. The second beep indicates that the interpreter is initializing. The third beep indicates that the System is initializing. The next three beeps indicate that the screen and keyboard are initializing. The "Welcome to UCSD p-System" title screen and the p-System promptline appear on the display.
4. If the light does not come on or if the System title screen does not appear, the corresponding card may not be properly inserted. Repeat the "Set-Up Instructions" procedure. If you still have difficulty, see "In Case of Difficulty" on page 19.

1.1.3 Removing the P-Code Card

1. Turn OFF the computer console, Peripheral Expansion System, monitor, and any other attached accessories.
2. Wait two minutes and then remove the top from the peripheral system.
3. Pull up the wire handles on the ends of the card.
4. Firmly pull on the handles to remove the card from its slot in the peripheral system.

1.1.4 Using the P-Code Card with the TI Disk Memory System

To ensure that the computer accesses the disk drive containing either a program or data diskette, perform the following steps.

1. Be sure that the Memory Expansion, P-Code, and Disk Drive Controller cards are inserted into the Peripheral Expansion System and that the disk drives are attached to your system.
2. Turn on the disk drives, peripheral system, and monitor. Next, insert a p-System diskette into each drive you plan to use with your system. Then turn on the computer console. Now the system checks each disk drive sequentially. If a diskette is in a drive, that drive is regarded as being "on-line" (accessible). If a diskette is not in a drive, the system will not check that drive for information. Rebooting or I(nitializing the System, or using the V(olumes command in the Filer program, tells the computer to recheck all of the disk drives.

P-Code Card

1.2 SPECIAL KEYS

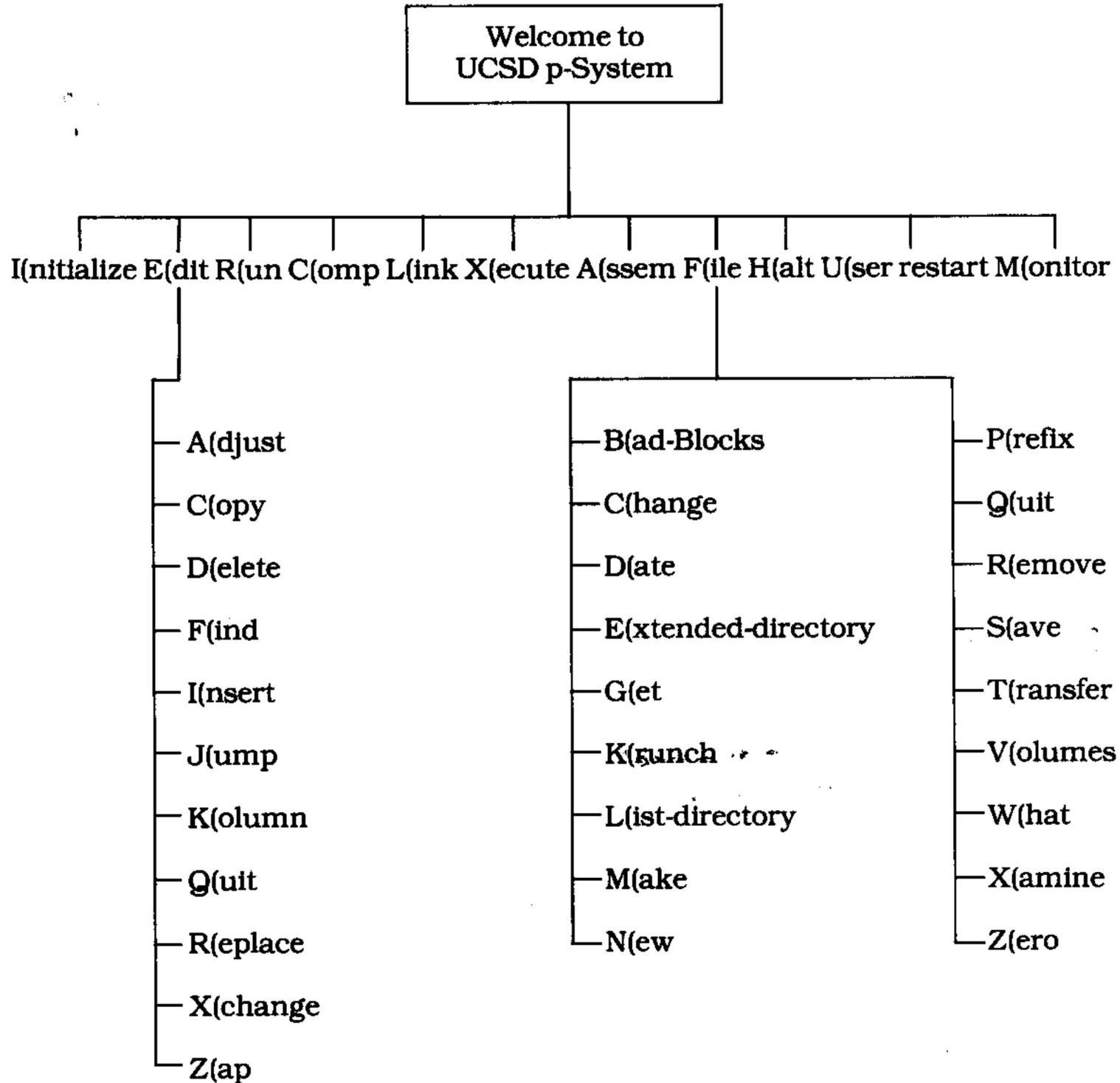
A keyboard overlay for the TI-99/4 console and a two-level strip overlay for the TI-99/4A console are included to help you more easily identify certain keys that are used with the p-System. On the TI-99/4 console, certain keys are used in combination with the SHIFT and SPACE keys. On the TI-99/4A console, certain keys are used in combination with the FCTN and CTRL keys.

<u>Name</u>	<u>TI-99/4</u>	<u>TI-99/4A</u>	<u>Action</u>
	SHIFT F	FCTN 1	Deletes a character in the Editor.
<ins>	SHIFT G	FCTN 2	Inserts a character in the Editor.
<flush>	SPACE 3	FCTN 3	Stops writing the output to the screen.
<break>	SPACE 4	FCTN 4	Stops the program and initializes the System.
<stop>	SPACE 5	FCTN 5	Suspends the program until this key is pressed again.
<alpha lock>	SPACE 6	FCTN 6	Acts as a toggle to convert upper-case letters to lower-case and back again.
<screen left>	SPACE 7	FCTN 7	Moves the text displayed on the screen to the left 20 columns at a time.
<screen right>	SPACE 8	FCTN 8	Moves the text displayed on the screen to the right 20 columns at a time.
<line delete>	SHIFT Z	FCTN 9	Deletes the current line of information.
{	SPACE 1	FCTN F	Types the left brace {.
}	SPACE 2	FCTN G	Types the right brace }.
[SPACE 9	FCTN R	Types the left bracket [.
]	SPACE 0	FCTN T	Types the right bracket].
<etx/eof>	SHIFT C	CTRL C	Indicates the end of a file and fills that file with zeros in the Editor.
<esc>	SPACE .	CTRL .	Tells the program to ignore previous text.
<tab>	SHIFT A	CTRL I	Moves the cursor to the next tab.
<up-arrow>	SHIFT E	FCTN E	Moves the cursor up one line.
<left-arrow>	SHIFT S	FCTN S	Moves the cursor to the left one line.
<right-arrow>	SHIFT D	FCTN D	Moves the cursor to the right one line.
<down-arrow>	SHIFT X	FCTN X	Moves the cursor down one line.
<return>	ENTER	ENTER	Tells the computer to accept the information you type.

SECTION 2: USING THE SYSTEM COMMANDS

This section includes a description of each command at the System level. The System command level (the "outer level") is the chief control for the entire System. These outer level commands invoke basic System functions, such as calling the Compiler, the Editor, the Filer, and so forth. The System diagram of command levels, with the System commands as the outer level, is shown here.

COMMAND DIAGRAM



P-Code Card

To better understand the organization of the System, think of the outer level as the communications interface between the functions. Thus, the Filer program initializes a workfile which the Editor program uses to create a textfile. The Compiler then uses the textfile to create codefiles. Next, the Linker and the Library utility combine the codefiles to create a usable file. The X(ecute command then sets it into operation. This sequence of events is controlled by the System commands. It is done "by hand," since the System was started as an interactive environment. Therefore, you must use the System commands to accomplish interaction between the various System components.

At the System command level, typing a command letter:

- Calls a program, such as the F(ile;
- Performs an operation on the workfile, such as C(omp or R(un;
- Begins a file operation which prompts you directly for one or more filenames, such as X(ecute or L(ink; or
- Alters the status of the System, such as I(nitalize or H(alt.

2.1 PROMPTLINES

A promptline shows the command options available at any given level of the System. Each command is invoked by typing a single letter. When any portion of the promptline is displayed, all of the commands are available by typing the appropriate single-letter key. All promptlines include:

- The name of the "level" (System module) at the beginning;
- A list of available commands, with the calling letter capitalized and separated from the rest of the word by "(";
- The version number of the program in square brackets at the end of the line; and
- The capability of containing up to 40 characters.

A promptline usually is displayed at the top of the screen, letting you know your available options. Note that it is not always visible when you are using the Editor to insert text, and it is never visible while a program is running. Typing unintelligible commands at any level may cause the promptline to disappear. If this occurs, pressing the <spacebar> clears the display and causes the correct promptline to be displayed.

Some promptlines include a "?," indicating that there are more commands than can fit onto one line of your display. Type a question mark (?) to display the additional commands. For example, the following promptlines appear at various stages as you use the System.

- Command: E(dit, R(un,? [IV.0 B3H]
- Command: F(ile, C(omp,? [IV.0 B3H]
- Command: L(ink, X(ecute,? [IV.0 B3H]
- Command: A(sssem, D(ebug,? [IV.0 B3H]
- Command: H(alt,? [IV.0 B3H]
- Command: I(nitalize,? [IV.0 B3H]
- Command: U(ser restart,? [IV.0 B3H]
- Command: M(onitor, [IV.0 B3H]

(Note: The "B3H" displayed at the end of the promptline is the System version number. Future releases of the System, if any, may display a different version number.)

SECTION 3: P-SYSTEM COMMANDS

The commands on the promptlines provide access not only to program development but also can be used for special-purpose applications, such as word processing, computer-aided instruction, interactive business data processing, communications, process control, and scientific analysis.

Although commands for the Editor, Filer, Linker, Assembler, and Compiler appear on the promptline, the program capabilities are only available when the appropriate diskette is on line. That means you must insert that diskette in a disk drive *before* booting your System, or you must initialize your System *after* the appropriate diskette is inserted in a disk drive.

The commands available from the p-System promptline are explained in this section in alphabetical order according to the letter you type to access them. For your convenience, the commands are listed here in the order they appear on the promptline with the corresponding section. **Note:** A Debug utility is not available in this version of the p-System.

<u>Command</u>	<u>Section</u>
E(dit	3.3
R(un	3.9
F(ile	3.4
C(omp	3.2
L(ink	3.7
X(ecute	3.11
A(ssem	3.1
D(ebug	not available
H(alt	3.5
I(nitalize	3.6
U(ser restart	3.10
M(onitor	3.8

P-Code Card

3.1 A(ssem)

The A(ssem (assemble) command lets you access the Assembler program⁺ to assemble code. Pressing **A** causes SYSTEM.ASSMBLER to execute. If a workfile is present, either *SYSTEM.WRK.TEXT or the designated .TEXT file assembles to a codefile of a generated machine language (depending on which of the assemblers has been named SYSTEM.ASSMBLER). If there is no workfile, you are prompted for a source filename, a codefile, and a listing file. The defaults for these are *SYSTEM.WRK.CODE, with the listing file being omitted.

When an error is detected in the program being assembled with the Assembler program, you have the choice of continuing assembly, stopping assembly, or returning directly to the Editor to correct the source file.

See the UCSD p-System Assembler manual for further details.

3.2 C(omp)

The C(omp (compile) command lets you access the Compiler program⁺ to compile a Pascal program. Pressing **C** causes SYSTEM.COMPILER to execute. If a workfile is present, either *SYSTEM.WRK.TEXT or the designated .TEXT file is compiled to p-code. If there is no workfile, you are prompted for a source file and a code filename. The default for this is *SYSTEM.WRK.CODE.

If an error is detected in your program, the Compiler program gives you the choice of continuing compilation, stopping compilation, or returning directly to the Editor to correct the source file. In the last instance, the cursor is positioned at the point of error detection, and if the file *SYSTEM.SYNTAX is present, an error message is displayed (see *Appendix A*).

See the UCSD p-System Compiler manual for further details.

3.3 E(dit)

The E(dit command lets you access the Editor program⁺ to edit a workfile. Pressing **E** causes SYSTEM.EDITOR to execute. If a .TEXT workfile is present, it is displayed and available for editing. If no workfile is present, you are prompted for a filename and you have the additional options of either escaping the Editor or entering the Editor with no file at all (with the intent of creating a new one).

With the Editor program, you can create program or document textfiles or alter and add to existing files.

See the UCSD p-System Editor manual for further details.

⁺ sold separately

3.4 F(ile

The F(ile command lets you access the Filer program⁺ for file manipulation. Pressing **F** causes SYSTEM.FILER to execute. The Filer program provides commands for maintaining the workfile, moving files, and maintaining disk directories.

See the UCSD p-System Filer manual for further details.

3.5 H(alt

The H(alt command lets you exit from the p-System. Pressing **H** causes the System to stop execution and the master title screen to be displayed. You can then select an option shown on the master selection list, such as TI BASIC. To reset the system for Pascal, remove all diskettes from the disk drives and then turn off the console and the Peripheral Expansion System. Wait 10 seconds and then turn the peripheral system on again. Next, insert the System diskettes into the disk drives and turn on the console.

3.6 I(nitialize

The I(nitialize command lets you reboot your System. Pressing **I** causes the file *SYSTEM.STARTUP, if present, to execute. (**Note:** SYSTEM.STARTUP must be a codefile.) *SYSTEM.STARTUP also executes automatically after a bootstrap.

You may create your own SYSTEM.STARTUP. Some applications of this might be displaying reminders for the next session with the System or creating a program that runs automatically when the System is turned on. To create a SYSTEM.STARTUP, you must create a .CODE file with the Compiler and then change its name to SYSTEM.STARTUP with the Filer.

All execution errors that are not "fatal" (see *Appendix A*) cause the System to initialize. Note that although an I(nitialize command does not clear any redirections, any runtime error does.

3.7 L(ink

The L(ink command lets you access the Linker program⁺ to link programs or program segments. Pressing **L** causes the file SYSTEM.LINKER to execute. The Linker program allows you to link code-assembled routines into host compilation units which have been compiled from a high-level language. The Linker also allows you to link machine language code routines together.

See the UCSD p-System Linker manual for further details.

⁺ sold separately

P-Code Card

3.8 M(onitor

The M(onitor command keeps a record of keystrokes. One convenient way to create a series of keystrokes is with the M(onitor command. Press **M** to enter the monitor mode. You may use the System in a normal manner, but all input is saved in a file. Thus, to automate a sequence of System commands, you begin a monitor and go through all the commands that are to be remembered. Then you end the monitor and all the input is saved as a file. This file can be used by redirecting System input to the monitor file with the I = execution option string (see Section 3.11.2).

The monitor file can be either a .TEXT file or a datafile. If it is .TEXT, you can use the Editor program to alter it. (**Note:** Some special characters cannot be read by the Editor. Monitor files containing these characters cannot be edited with the Editor.)

The M(onitor command itself cannot be recorded in a monitor file.

When **M** is pressed, the following prompts are displayed:

Monitor: B(egin, E(nd, A(bort,?
Monitor: S(uspend, R(esome

To select a command, type the initial letter of that command.

3.8.1 B(egin

B(egin starts a monitoring session. When prompted, type a filename up to 15 characters in length, and press <return>. Then select R(esome to return to the System promptline. To specify the length of a monitoring file, type the filename followed by [1]. This instructs the computer to designate the first free single block of memory for the monitoring file. Although one block of memory is normally sufficient for a monitoring file, you may specify a larger block of memory by changing the number in the brackets.

3.8.2 E(nd

E(nd stops a monitoring session and saves the monitor file. To return to the System promptline, select S(uspend. If no monitor file is open, an error message is displayed (see Appendix A).

3.8.3 A(bort

A(bort ends a monitoring session without saving the monitor file. To return to the System promptline, select R(esome or S(uspend.

3.8.4 S(uspend

S(uspend turns off the monitoring capability but does not close the monitor file. The System returns to the promptline where you can now type commands without recording them. The monitor file remains open, and more can be added to it with the R(esome command.

3.8.5 R(esome

R(esome starts the monitoring capability again and returns to the System promptline. If the monitoring capability has not been suspended, nothing happens.

3.9 R(un

The R(un command lets you run an existing program or one you have created. Pressing **R** causes the current workfile to execute. If the workfile does not contain a codefile, R(un calls the Compiler. If the compilation is successful, the resulting code is run. If there is no workfile, R(un again calls the Compiler, which then prompts you for the name of a textfile to compile.

If the codefile requires linking to one or more external codefiles, the Linker is called automatically to search *SYSTEM.LIBRARY. If the external files cannot be found there, an error message is displayed (see *Appendix A*).

3.10 U(ser restart

Pressing **U** causes the last System or application program executed to execute again, with all file parameters equal to what they were previously. U(ser Restart does *not* restart the Compiler or Assembler program. The command has such uses as making multiple runs of a user program or returning to the Editor after a workfile update.

3.11 X(ecute

The X(ecute (execute) command lets you call programs that have already been compiled. It may also be used simply to take advantage of the execution options. Pressing **X** causes the computer to prompt you, as shown here, to enter a filename.

What file?

Respond with an execution option string (see Section 3.11.2). In the simplest case, this string contains nothing but the name of a codefile you wish to execute.

If the codefile cannot be found, the message "No (filename) file" is displayed. If all of the code necessary to execute the codefile has not been linked in, the message "Must L(ink first" is displayed. If the codefile contains no program (all its segments are units or segment routines), the message "No program in (filename)" is displayed.

If the execution option string contains *only* option specifications, they are treated as explained in the Section 3.11.2. If the execution option string contains both option specifications *and* a codefile name, the options are performed first, and then the codefile executes (unless one of the errors named in the preceding paragraph occurs). **Note:** The codefile name must be listed first in the execution option string.

Also, the codefile *must* have been created with a .CODE suffix, even if its name has been changed subsequently.

3.11.1 Prompts for Filenames

Several of the System commands prompt for filenames. The conventions are the same for all responses to filename prompts throughout the System. A filename can be up to 15 characters in length and is followed by pressing <return>. Before <return> is pressed, the name may be corrected by pressing <backspace> or <delete line> and then retyping the name. Prompts often expect .TEXT or .CODE files, and these standard suffixes may be omitted from the filename since the System programs append them automatically. To prevent this automatic appending, follow the filename with a ".".

3.11.2 Execution Option Strings

The X(ecute command allows you to specify some options to modify the System's environment. These modifications include redirecting standard program input/output (I/O) or standard System I/O, changing the default prefix (for example, the volume name part of a filename), and changing the default library textfile. These options are also available from within a user program.

All of these options are specified by means of "execution option strings," which are strings that contain one optional filename and zero or more option specifications. An option specification consists of one or two letters followed by an equal sign (=) and possibly followed by a filename or literal string.

These are the possible execution options, with a summary of their uses:

<u>OPTION</u>	<u>ACTION</u>
L	Changes the default library textfile
P	Changes the default prefix
PI	Redirects program input
PO	Redirects program output
I	Redirects System input
O	Redirects System output

Note: On both TI Home Computer consoles, upper- and lower-case letters may be used interchangeably with an execution option string.

Several different option strings may be entered at a single time as long as they are separated by one or more spaces. Also, there may be a single optional space between the "=" and the following filename or string.

These options are invoked with the X(ecute command. Causing redirection from within a user program requires the use of the REDIRECT intrinsic (and possibly the EXCEPTION intrinsic); refer to the descriptions of these intrinsics in the UCSD p-System Compiler manual. The intrinsic CHAIN also makes use of execution option strings.

Redirecting System input to come from a file or main memory amounts to driving the System from a series or script of commands. This is a useful tool, especially in testing or applications that load automatically. One way to create a script for the System is to use the M(onitor command (see Section 3.8), which records keystrokes by writing them to a file while they are performed.

Note: Redirection applies only to the input and output of standard files, and therefore has no effect on low-level device I/O intrinsics such as UNITWRITE or BLOCKREAD.

3.11.2.1 Alternate Prefixes and Libraries

You can change the default prefix (the name of the diskette) with the P = execution option string. After this is done, all filenames that do not explicitly name a volume are prefixed by the default prefix. This is equivalent to using the P(refix command in the Filer.

To change the default prefix, press **X** when any portion of the promptline is displayed. Next, the program prompts you for an execution option string. Respond to the "What file?" prompt by typing P = and the volume name, up to seven characters in length. **Note:** Since the program automatically provides the colon at the end of the of the volume name, you do not need to type it.

For example, if you type P = "ABC", the default prefix becomes ABC:.

Similarly, the default "library textfile" can also be changed. The library textfile is a file that contains the names of several user libraries. When a program with separately compiled units is run, the System searches for them first in the files named in the library textfile, and then in *SYSTEM.LIBRARY. When the System is booted, the default library textfile is *USERLIB.TEXT. For more information about libraries, see the Linker manual.

To change the default library textfile, press **X** when any portion of the promptline is displayed. Next, the program prompts you for an execution option string. Respond to the "What file?" prompt by typing L = and the filename, up to 15 characters in length.

For example, typing L = MYLIB makes the file MYLIB.TEXT the new default library textfile.

Note: The execution option strings are performed in the following order.

1. Change prefix, if the P = (option) is present.
2. Change library textfile, if the L = (option) is present.
3. Do the I/O redirections, if any are present.

3.11.2.2 Redirection

The following execution option strings control redirection.

PI = (filename)

PI = (string)

PO = (filename)

I = (filename)

I = (string)

O = (filename)

P-Code Card

PI = redirects program input and overrides any previously input redirection. PI = (filename) causes the input to a program to come from the file named. PI = (string) causes the input to a program to come from the program's input buffer and adds the string given to the input buffer. Using PI = without a filename makes program input the same as System input.

PO = redirects program output and overrides any previous output redirection. PO = (filename) causes program output to be sent to the file named. Using PO = without a filename makes program output the same as System output.

I = redirects System input and overrides any previous I = . I = (filename) causes System input to come from the file named. I = (string) causes System input to come from the System's input buffer and adds the string to the input buffer. Using I = without a filename resets System input to CONSOLE:.

O = redirects System output and overrides any previous O = . O = (filename) causes System output to be sent to the file named. Using O = without a filename resets System output to CONSOLE:.

For PI = (filename) and I = (filename), the filename may specify either a disk file or an input device that sends characters. If the file is a disk file, redirection ends with an EOF and the System performs the equivalent of an input redirection with no filename, thus resetting input. If the file is a device, redirection continues until you explicitly change it. This allows you to control the System from a remote port, such as REMIN:.

For PO = (filename) and O = (filename), the (filename) may specify either a disk file or an output device that receives characters. If the file is a disk file, it is named literally as shown (to make it a textfile, you must explicitly type .TEXT). Whenever output redirection is changed, the file is closed and locked.

For PI = (string) and I = (string), the (string) may be any sequence of characters enclosed in double quotes. Any double quote embedded in the string must be typed as two single quotes. Input buffers are located in the main memory. Program or System input may be redirected to come from *both* a file and the appropriate input buffer. However, if this is the case, the buffer is used *first* (until it is empty). Strings are always added to input buffers so that they are read in order (first in, first out). Commas in scratch input buffers are treated as carriage returns.

Program redirection ends when the program terminates. Any characters still in the program's input buffer are lost.

System redirection ends when the System terminates with a H(alt command or an execution error.

Note: Redirection applies only to the standard files, called "input" and "output." Redirection affects file-level operations and intrinsics but *not* device-level intrinsics, such as UNITREAD, UNITWRITE, BLOCKREAD, BLOCKWRITE, and so on. It also cannot affect calls of the following form.

```
REWRITE(MY__FILE, "CONSOLE:")    or    WRITE(MY__FILE, LOTS__OF__TEXT)
```

These calls do not involve the standard input and output files.

A user program can also access redirection with the intrinsic REDIRECT and clear redirection with the intrinsic EXCEPTION. The CHAIN intrinsic allows you to "queue" an execution option string for execution after the program that contains it has finished executing. For further details, see the UCSD p-System Compiler manual.

SECTION 4: LOADING AN EXISTING PROGRAM

4.1 LOADING A DISKETTE-BASED PROGRAM

1. Be sure all the accessories are attached to the console and turned on in the proper order (see Section 1.1).
2. Insert the program diskette in the disk drive.
3. Turn on the console. The System boots and displays the promptline automatically.
4. Select the X(ecute command by pressing **X**. The computer then displays the following prompt:

What file?

The filename can be entered in the following ways:

TEST:HI

where TEST is the volume name and HI is the filename, or,

#n:HI

where #n is the device number of the drive containing the program diskette and HI is the filename. The following chart lists the available disk drive device numbers.

<u>Device</u>	<u>Number</u>
first disk drive	#4
second disk drive	#5
third disk drive	#9

See the UCSD p-System Filer manual for details concerning device names and numbers.

5. The computer then runs the program indicated by the filename.

Note: If you do not know the filename, refer to the UCSD p-System Filer manual for details on obtaining a directory of the files on a diskette.

4.2 LOADING A CASSETTE-BASED PROGRAM

1. Be sure all the accessories are attached to the console and turned on in the proper order (see Section 1.1).
2. Insert the program cassette in the cassette recorder.
3. Turn on the console. The System boots and displays the promptline automatically.
4. Select the X(ecute command by pressing **X**. The computer then displays the following prompt:

What file?

The filename can be entered by typing TAPE: and pressing <return>. Then, follow the instructions that appear on the display.

To load an additional file from a cassette tape, position the tape at the start of the file. Then type TAPE: and press <return>.

5. The computer now runs the program that has been loaded into the system.
-

SECTION 5: APPENDICES

5.1 APPENDIX A — Execution Error Messages

If an execution error is detected, the appropriate error number message is displayed. The following chart lists the execution error number and the message. To reinitialize the system after an error, press the <spacebar>.

0	System error. . . FATAL	10	User I/O error (see <i>Appendix B</i>)
1	Value range error	11	Unimplemented instruction
2	No procedure in segment	12	Floating point error
3	Exited uncalled procedure	13	String overflow
4	Stack overflow	14	HALT
5	Integer overflow	15	Illegal heap operation
6	Divide by 0	16	Breakpoint
7	NIL pointer reference	17	Incompatible real number size
8	Program interrupted by user	18	Set too large
9	System I/O error. . . FATAL		

5.2 APPENDIX B — I/O Error Messages

If an I/O error is detected, the appropriate error number message is displayed. The following chart lists the I/O error number and the message. To reinitialize the system after an error, press the <spacebar>.

1	Parity error	10	File not found
2	Illegal unit number	11	Duplicate directory entry
3	Illegal request	12	File already open
4	Data-com timeout	13	File not open
5	Volume went off-line	14	Bad input format
6	File lost in directory	15	Ring buffer overflow
7	Bad file name	16	Write protect
8	No room on volume	17	Illegal block
9	Volume not found	18	Illegal buffer

SECTION 6: SERVICE INFORMATION

6.1 IN CASE OF DIFFICULTY

If the P-Code Card does not appear to be working properly, check the following:

1. *Power* — Be sure the peripheral system is plugged in and turned on. Also, check to be sure that the fuse is not blown by looking at the fuse to see if the wire is unbroken.
2. *Card Position* — Turn the power OFF, wait two minutes, and remove the top from the peripheral system. Verify that all cards are inserted properly and then replace the top.
3. *Program Errors* — Be sure all commands and statements are used as described in the appropriate manual. If the P-Code Card works properly when tested but not with a program, the problem is probably with the program or with the indicated device number where the program diskette is located.
4. *Home Computer* — Check to see that the Home Computer works properly with all accessories disconnected.
5. *Peripheral Expansion System* — Check for proper connection between the console and peripheral system.
6. If none of the above procedures corrects the difficulty, consult "If You Have Questions or Need Assistance" or see the "Maintenance and Service Information" section of the *User's Reference Guide*.

6.2 EXCHANGE CENTERS (LOCAL SERVICE OPTIONS)

If your P-Code Card requires service, instead of returning the unit to a service facility for repair or replacement, you may elect to exchange the unit for a factory-reconditioned P-Code peripheral card of the same model (or equivalent model specified by TI) by going in person to one of the exchange centers which have been established across the United States. A handling fee will be charged by the exchange center for in-warranty exchanges of the P-Code peripheral card. Out-of-warranty exchanges will be charged at the rates in effect at the time of the exchange.

To determine if there is an exchange center in your area, look for Texas Instruments Exchange Center in the white pages of your telephone directory, or look under the Calculator and Adding Machine heading in the yellow pages. Please call the exchange center for availability and exchange fee information. Write the Consumer Relations Department for further details and the location of the nearest exchange center.

6.3 IF YOU HAVE QUESTIONS OR NEED ASSISTANCE

If you have questions concerning P-Code peripheral card repair, or peripheral or software purchase, please call our Customer Relations Department at 800-858-4565 (toll free within the contiguous United States except Texas) or 800-692-4279 within Texas. The operators at these numbers cannot provide technical assistance.

For technical questions, such as programming information, specific applications, etc., you can call 806-741-2663. Please note that this is not a toll-free number, and collect calls cannot be accepted.

As an alternative, you can write to:

Consumer Relations Department
Texas Instruments Incorporated
P.O. Box 53
Lubbock, Texas 79408

Because of the number of suggestions which come to Texas Instruments from many sources containing both new and old ideas, Texas Instruments ~~will consider~~ such suggestions only if they are freely given to Texas Instruments. It is the policy of Texas Instruments to refuse to receive any suggestions in confidence. Therefore, if you wish to share your suggestions with Texas Instruments, or if you wish us to review any programs which you have developed, please include the following statement in your letter:

"All of the information forwarded herewith is presented to Texas Instruments on a nonconfidential, nonobligatory basis; no relationship, confidential or otherwise, expressed or implied, is established with Texas Instruments by this presentation. Texas Instruments may use, copyright, distribute, publish, reproduce, or dispose of the information in any way without compensation to me."