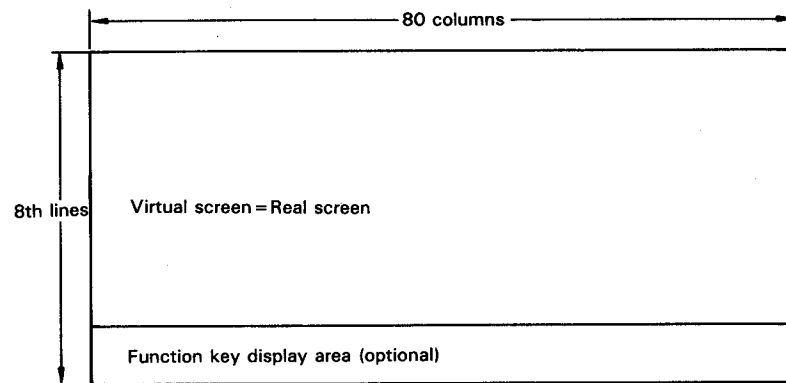


## Chapter 3

# OPERATING THE COMPUTER UNDER CP/M



The screen allows individual dots of the screen to be lit (bit image mode). This is only possible by means of software. The most convenient way to understand this mode is to use BASIC. If this mode is used other than in BASIC, it will be with special applications software and the appropriate manual should be consulted for its use.

Chapter 2 discussed the operation of the computer at the simplest level. This chapter deals with the operation on a day to day level. It mainly covers using the operating system CP/M, housekeeping of files and using the utility and applications programs. For a complete guide to the operation of an applications program, see the appropriate manual for that program.

### 3.1 What Is CP/M?

CP/M is the most popular operating system for microcomputers. An operating system is a collection of computer programs which have been assembled to make it easy for the user to run programs, handle saving and loading of data and generally allow easy transfer of information between software and hardware. The popularity of CP/M is due to the fact that it can be used on a large number of machines.

To understand what an operating system does, consider the analogy of a telephone system linked to the switchboard of a large company. If anyone dials into the company the public telephone network handles the connection to the company. The switchboard in the company then directs the caller to the person to which he wishes to speak. Similarly, if a person inside the company wishes to call out, he again goes through the switchboard to reach the public telephone system.

CP/M is very much like this except that there are a number of 'switchboards' for different parts of the hardware, e.g. the disk, Microcassette and devices such as the keyboard and screen. Anyone writing software for a CP/M based machine only has to use the equivalent of the public telephone system in this analogy. This means that having written a program such as a word-processor, it requires little if any further programming to allow the program to be run on a computer other than the one for which it was originally designed, providing both computers use the CP/M operating system. All interfaces between the

software and hardware are handled by the different versions of CP/M written for the different computers. This is the reason why CP/M has been chosen as the operating system for the PX-8. The huge library of CP/M based software can be installed on the PX-8, making it easy for the user to find the particular application he requires. This increases the power and versatility of the computer since it is only a machine to run software - a tool to extend the mind much as a hammer is a tool to extend the body.

If you are a beginner to computing, CP/M can be a little frightening. You will find that the terms used to communicate ideas to the computer are not as near to normal human communication as one would like. There are a number of jargon words which will have to be learnt. This chapter has been written in such a way that as far as possible you only have to think about the current task. If necessary, use the index to direct you to the appropriate part of the chapter to find out about anything you do not understand. Any difficulties you experience will disappear with practice, and you will find using CP/M will soon be second nature to you.

The only way to learn to use CP/M is to try out the examples given. This manual cannot hope to cover the finer points of the system, but fortunately there are a number of books which describe all aspects of CP/M at all levels, from the absolute beginner to the most advanced user. A list of some of these are given at the end of Chapter 3.

## 3.2 Files And File Names

In a standard CP/M system each drive holds floppy disks which are used to store programs. The PX-8 has some drives which do not contain floppy disks. For the purposes of using them under CP/M, the RAM disks, Microcassette drive and the ROM drives behave as if they were standard disk drives. There are a few specific differences, e.g. the ROM drives cannot be used to store information. The use of the Microcassette drive is described in Chapter 4.

The programs and data are stored in files. A file is simply a collection of information which is related as a unit. Files fall into two main categories. Program files are a set of instructions, for example a wordprocessing program. Data files store information which is used by a program, for example text used by the wordprocessor or the information used by an accounts program. Files are stored by name. To make it easier to understand which files are which, the name is divided into two parts. The first part consists of up to eight characters which are used to give a distinctive name to the file.

All of the printable characters can be used except:

< > , . ; : = ? \* [ ] and the space character

The secondary part of the name is used to describe what type of file is stored. Secondary names are frequently referred to as file types or extensions. This part of the name is used in the MENU to select the type of file to be displayed on the MENU in the file name area. It is possible to store a file without an extension, but this can cause problems. Applications programs frequently add the extension to the primary name you have given the file so that you know which files belongs to that program. The extension is separated from the primary filename by a full stop.

Examples of file names with extensions are:

**BASIC.COM**  
**STAT.COM**  
**SAMPLE.ASM**  
**MEMO.DAT**

Note that all filenames are given in upper case letters. Even if you type them in lower case, they are stored on the disk as upper case. If you are using the filename, CP/M will understand if you type the file you want in lower case or even a mixture of upper and lower case. Note also that the separator in front of the extension is printed next to the primary file name. This may not always

be so if the computer prints it out.

Some extensions are standardised and are recognised by CP/M or particular programming languages or applications languages.

Examples of common extensions are:

<b>.ASM</b>	ASseMbly language source file
<b>.BAK</b>	BACkUp file
<b>.BAS</b>	BASIC program source file
<b>.COM</b>	Directly executable COMmand file
<b>.DAT</b>	DATA file
<b>.DOC</b>	DOCument file
<b>.MSG</b>	A MeSsaGe file
<b>.OBJ</b>	OBJect code for machine code program
<b>.SUB</b>	Command file for use with the SUBMIT program
<b>.TXT</b>	TeXT file
<b>.\$\$\$</b>	Temporary file

Some terms used in this list which are commonly met are:

**SOURCE CODE.** This is text which is converted or interpreted by a program to create machine readable codes. It is in a form which is understandable provided you can understand the language it was written in.

**OBJECT CODE.** This is code directly readable by the computer as a set of instructions.

**COMMAND FILE or COM FILE.** This is the most important file you will meet under the CP/M operating system. It is basically a program file which is loaded into memory to execute a set of instructions. There are a number of commands built in to CP/M. A COM file is sometimes known as a TRANSIENT COMMAND since it is temporarily loaded into the TRANSIENT PROGRAM AREA or TPA, i.e. the part of the computer's memory set aside to run programs.

### 3.2.1 Referring to file names - (ambiguous/unambiguous file names and wildcards)

In many cases CP/M or a program requires you to use the complete, exact file name. In this case there is no doubt which file is being referred to. The name is UNAMBIGUOUS as there is only one file of that name. Sometimes you may

wish to refer to several similar file names. For example they may be all the files beginning with the letter "E" or all the files with extension ".COM". In this case the file name is AMBIGUOUS. In references to CP/M you will frequently see these terms abbreviated:

ufn	means	Unambiguous File Name
afn	means	Ambiguous File Name

In using ambiguous file names, you need to tell CP/M some further information, so that it can have a precise idea what you mean. In order to do this you can use the "?" characters to denote "any character" and the "\*" to denote "any group of characters". The "?" and "\*" characters are known as WILDCARD characters.

For example:

<b>*.COM</b>	means any file name with the extension .COM
<b>NAME.*</b>	means all the files with the primary name "NAME" but with any extension, eg. NAME.COM, NAME.BAS, NAME.DAT
<b>E*.COM</b>	means any file name beginning with E which has the extension .COM
<b>E*.*</b>	means all files beginning with E no matter what their extension



**Warning:**  
you cannot use:

**\*P.\*** to mean all the files ending in the letter "P". CP/M will treat this as meaning every single file. This is important to remember if you are erasing files from a disk.

<b>?????????.COM</b>	means any file with the extension .COM because there are eight (i.e. the maximum) wildcard characters.
<b>?????.COM</b>	means any file of not more than four characters in length, which has the extension .COM.
<b>A???.*</b>	means any file beginning with the letter "A" which has not more than four characters.
<b>?S???.BAS</b>	means any file whose second character is "S", which has five or less characters, and has the .BAS extension.
<b>?????ROP.*</b>	means any file ending in the characters "ROP".
<b>*.B??</b>	means all files whose extensions begin with the character "B".

### 3.3 Starting To Use CP/M

Although the MENU is an aid to using CP/M, it is not a replacement. To use the MENU effectively, a working knowledge of CP/M is required. An understanding of how to use CP/M is also required to make effective use of the CP/M utility programs.

#### 3.3.1 Disk drive names

Switch off the MENU from the System Display (section 2.2.3) and the screen will clear to show the system prompt "A>". All drives are given an identification letter. It is possible to change the relationship between the actual physical drives and the identifying letter on some of the drives. This can be done with the CONFIG program described in section 3.8. When you initialize your PX-8 the drives are specified as follows:

Identifying letter	Physical Drive
A:	RAM disk
B:	ROM 1
C:	ROM 2
D:	Floppy Disk Drive 1
E:	Floppy Disk Drive 2
F:	Floppy Disk Drive 3
G:	Floppy Disk Drive 4
H:	MICROCASSETTE DRIVE

The default drive is set to drive A: which is why the system prompt said "A>". Notice that apart from the system prompt the drive name is followed by a colon(:). Since file names and commands consist of letters, CP/M has to have some means of recognising when you are talking about a disk drive identifier. The colon is always used in CP/M to denote a drive. The system prompt always includes the default or currently "logged in" drive as a reminder to you which drive is currently logged in. Unless you specify another drive, all access to drives (saving and loading or simply inspecting the contents of the drive) will be made to that drive. To use the currently logged in drive, you need not type in its identifier.

This can be seen with the following sequence of operations:

Type the command DIR next to the system prompt and press the **RETURN** key. This means "show me all the files stored on the currently logged in drive". If there are no files on the disk the screen will show:

```
A>DIR
NO FILE
A>
```

You can specify another drive by adding the drive name after the DIR command as follows when the screen will show:

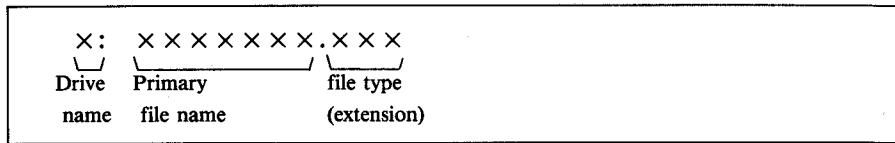
```
A>DIR C:
C: PIP          COM : STAT      COM : SUBMIT  COM : XSUB    COM
C: FILINK      COM : TERM      COM : CONFIG  COM
A>
```

Rather than type the command DIR, you can press the Programmable Function Key **PF1**. This will print the command "dir", including the required space.

To change the logged in drive, type the name of the new drive and press the **RETURN** key. For example to change the logged in drive to drive C: type the two characters "C:" and press **RETURN**. The prompt then changes to "C>". Now simply using the DIR command will give a directory of drive C:, and then return to show drive C: is the currently logged in drive. If you do this the screen will show:

```
A>C:
C>DIR
C: PIP          COM : STAT      COM : SUBMIT  COM : XSUB    COM
C: FILINK      COM : TERM      COM : CONFIG  COM
C>
```

Drive name can be used in combination with the file name described in section 3.2 as follows.



**Example**

B: BASIC.COM

or

A: SAMPLE.BAS

### 3.3.2 Changing disks, tapes and ROMs — warm starts

On a conventional CP/M system the disks in the drives could, and indeed would be changed regularly. If you have additional disk drives you will experience the same problems as on a conventional CP/M system if you do not change the disks in a correct manner. On the PX-8, unless you have additional floppy disk drives, only Microcassette tapes and ROMs will be changed. It is more important to obey the rules in changing tapes, as data you have saved may not be readable at all unless you remove the tape in the correct manner.

When you change disks, you must let CP/M know you have done so. This is because certain disk directory information is stored in memory at all times, which is used to allocate space on the disk. When a disk is changed, this information must be replaced by the information from the new disk. If you change a disk without telling CP/M you have done so, it is possible to lose data from the new disk, and also files may not be closed properly on the old disk making it difficult to read them.

IF YOU CHANGE FLOPPY DISKS always press **CTRL** - **C** or the **STOP** key after you have done so.

ALWAYS EXIT A PROGRAM IN AN ORDERLY MANNER by taking the EXIT option. In some cases there will be a specific command to do so, in others you can use **CTRL** - **C** .

A WARM START restores the internal information which CP/M uses to reorientate itself. This is normally when disks have been changed or you have exited from a program. If the MENU is switched on, and you exit from a program if it does a warm start, the MENU will be displayed. If a program leaves the CP/M prompt on display and the MENU is switched on you should carry out a warm start by pressing **CTRL** - **C** or the **STOP** key. There may be a reason for the CP/M command line being displayed, most probably because there is information to be read from the screen which would disappear before you could read it, if the MENU came up straight away.

### 3.3.3 Changing ROMs

Details of physically changing over ROMs are given in Chapter 4. Since ROMs behave like write protected disks, when changing ROMs you must also carry out a warm start.

### 3.3.4 Changing microcassette tapes

The Microcassette drive has been designed to operate as a disk drive. However, because it is physically different it cannot be treated quite the same. It should become a habit to change a Microcassette tape through the System Display. Otherwise it is possible to not only make the removed tape unreadable, but also the one onto which you are writing the new data. Full details of using the Microcassette Drive are given in Chapter 4.



**WARNING:**

*A Warm start does not update the directory on a Microcassette tape. Always change tapes through the System Display. This can be done in the middle of an applications program by pressing **CTRL** and the **HELP** key. On completing any changes via the System Display, control will be returned to the applications program at the point where you left off.*



## 3.4 Using The Keyboard Under CP/M

Since CP/M has been designed as a universal operating system, it does not necessarily allow all the special keys on the PX-8 to function as they do in BASIC or in some applications programs. In particular the cursor keys are not supported by CP/M. Some of the special keys behave differently. The following summarises the use of the different functions available from the keyboard when the PX-8 is on the CP/M command line. They do not hold for the MENU command line.

### 3.4.1 Using the programmable function keys

The keys **PF1** to **PF5** and their shifted counterparts can be used to print the common CP/M command words, thus preventing errors and saving time. When setting them up, it is worth ensuring that trailing spaces are included to conform with the requirements of CP/M. The key display is normally absent from the lowest line of the screen but can be switched on using the CONFIG program. Another alternative in deciding which key provides which string is to use the attached Programmable Function key label. The CONFIG program also allows alteration of the strings given by each key.

### 3.4.2 Line termination commands

When you have entered a command, you must tell CP/M that you have finished. There are three key commands to do this:

The **RETURN** key is the simplest way to enter a command.

A **CTRL** - **M** is equivalent to a pressing the **RETURN** key.

A **CTRL** - **J** is equivalent to a line feed. On the CP/M command line it also causes the command to be entered as though you had pressed the **RETURN** key.

### 3.4.3 Character deletion commands

If a mistake is made, it is necessary to correct it. This can be done in two ways, but the CURSOR KEYS CANNOT BE USED.

**CTRL** - **H** or the BS (backspace) characters delete the character to the left of the cursor from the screen and move the cursor one step back.

The **DEL** key causes the key to the left of the cursor to be deleted, but also repeats the deleted character on the screen. If the **BS** or **CTRL** - **H** is used

to delete another character it will place the cursor one place to the left of this deleted character because any characters deleted by the **DEL** key do not exist except on the screen.

### 3.4.4 Line deletion commands

Sometimes it is necessary to delete a whole line when a mistake is made or for other reasons. There are two ways to do this:

**CTRL** - **X** will backspace to the beginning of the line, erasing all of the text.

**CTRL** - **U** will place a “#” character at the end of the line and then place the cursor on the next line ready to start over again. This is useful, because the incorrect command is still present on the screen, although it has been erased.

### 3.4.5 Display control commands

There are a number of control key sequences which can be used to handle the screen display. Again not all the special keys work, although some can be made to perform their function to a limited extent.

The **CLR** key can be used to clear the screen, although it is not supported by CP/M. It is equivalent to a **CTRL** - **L**. Typing either of these commands causes the CP/M command line to show:

A > ^L

Pressing the **RETURN** key will clear the screen, printing a question mark in the top left hand corner. The system prompt will be placed on the third line. This is because if you give CP/M a command it does not understand it will repeat the command followed by a question mark and return the system prompt.

**CTRL** - **E** can be used to overflow on to the next line before the user has pressed the **RETURN** key. This is useful if the screen is split (screen mode 1 or 2), as it enables a new word to be written on to a new line so that it can be read more easily. The “#” character is placed at the end of a line when using the **CTRL** - **U** or **CTRL** - **R** command to show it is a terminated line, to distinguish it from a **CTRL** - **E**.

**CTRL** - **R** is used to repeat a line with the errors removed if you are a bad typist and using the **DEL** key frequently. For example if you type an “R” instead of a “T” in the “STAT” command, and had deleted the “R” using DEL, and pressed **CTRL** - **R** keys after typing “T” again the screen would show:

A>STARRT #  
STAT

The **INS** key performs the same function as **CTRL** - **R** .

The **PAUSE** or **CTRL** - **S** key can be used to temporarily halt the screen scrolling. For example if a DIRectory has too many files to fit in the screen window, pressing either the **PAUSE** or **CTRL** - **S** keys will stop the display until another key is pressed, enabling the text to be read. The shifted cursor keys can also help with this problem as described below.

The SHIFTED CURSOR KEYS ( **↑** and **↓** ) can be used to scroll the real screen window up and down the virtual screen. Thus if a long DIRectory has scrolled off the real screen, these keys can be used to bring it on again.

The **CTRL** and CURSOR KEYS ( **←** and **→** ) can be used to change the displayed virtual screen in modes 0 and 1 and the cursor position in screen mode 2. For example a DIRectory can be stored on one virtual screen while a program is used on another. Care must be taken in using this facility. Full details are given in section 2.2.6.

The **SCRN** key can be used to lock the real screen window on part of the virtual screen. This is described in section 2.2.6.

### 3.5 Printing When On The CP/M Command Line

There are two ways to print when on the CP/M command line.

The **CTRL** and **PF5** key can be used to dump the contents of the screen to the printer.

**CTRL** - **P** can be used to toggle the echoing of text to the printer. The first time **CTRL** - **P** is pressed anything output to the screen is simultaneously sent to the printer. Pressing **CTRL** - **P** a second time only sends input to the screen. This can be used in utility programs, but may not work in applications programs.

## 3.6 The Concept of Logical And Physical Devices

As a universal operating system, CP/M has to have a means of communicating with all the devices associated with a computer in a general way. The software in each computer is then connected to the devices which are unique to that particular computer. This is achieved through the use of the concept of logical (symbolic) device names in CP/M being assigned to physical (actual) devices supported by the computer. Such a concept considerably simplifies the use of the computer.

### NOTE:

*In practice assignments are carried out by the use of the CONFIG program described in section 3.8. This allows various devices to be assigned using common names, rather than following the normal CP/M system. The description outlined below is presented for completeness, and for those familiar with CP/M, who wish to use the STAT program (section 3.8) to change assignments.*

### 3.6.1 The logical devices

There are four logical devices in CP/M which correspond to the types of devices which one would expect with a computer. They are given the symbols CON:, RDR:, PUN:, and LST: from the names associated with their functions.

**CON:** denotes the CONsole device, which is the means of allowing data to be input by and output to the user. In the case of the PX-8 it would correspond to the keyboard and LCD screen.

**RDR:** denotes the ReaDeR device, which receives information from another source. The name originally derives from a paper tape reader, but in the PX-8 is normally the RS-232C device.

**PUN:** denotes a device to output information. Its name derives from a paper tape PUNch. In the PX-8 it is normally assigned to the RS-232C port.

**LST:** denotes the output LiST device, normally a printer.

### 3.6.2 The physical devices

There are many possible physical devices which can be attached to a computer. Examples of such devices are printers, plotters, modems, and video monitors. They all have one thing in common, they involve the transfer of information to or from the computer. The physical devices are given symbols in CP/M which like the logical symbols relate to their function. A computer manufacturer may not support each device, and different manufacturers may give different assignments to the same device. The symbols corresponding to the physical devices are:

**TTY:** denotes a slow speed terminal. Its name is derived from the TeleTypewriter, which has a keyboard and printer as input and output devices for the user to communicate with the computer.

**CRT:** represents a high speed console device. Its name is derived from the commonest type of device a Cathode Ray Tube.

**BAT:** denotes a BATch processing device, which historically relates to devices such as punched card readers.

**UC1:** denotes a User defined Console device, which allows special console equipment to be used.

**PTR:** corresponds to a Paper Tape Reader.

**UR1: and UR2:** correspond to two different User defined Reader devices. They enable a computer manufacturer to provide two different high speed devices for input of data.

**PTP:** denotes a Paper Tape Punch device for output of data to paper-punch equipment.

**UP1: and UP2:** correspond to two different User defined Punch devices. They enable a computer manufacturer to output data to two high speed output devices.

**LPT:** is a Line PrinTer which allows data to be printed for a permanent record.

**UL1:** is a User defined List device. It enables the computer manufacturer to provide an interface for communication to special devices such as plotters.

### 3.6.3 Physical Devices Supported by the PX-8

The PX-8 does not have all the physical devices which could be supported by CP/M. The physical devices which the PX-8 has are:

- 1) The LCD display.
- 2) The keyboard
- 3) High speed serial port, which is used to communicate with floppy disk drives, printers and other special devices.
- 4) RS-232 serial port, which is used to communicate with other computers and also printers.

A computer manufacturer has to modify CP/M so that a particular physical device corresponds to one or more of the names allowed by CP/M. A device can correspond to more than one name, and each of the logical devices can be connected to more than one physical device. The four logical devices can be assigned to the physical devices as follows:

**Table 3.1**

Logical Device	I/O	Physical Device			
LST:	O	TTY: Serial (printer)	CRT: LCD display	LPT: RS-232C	UL1 not implemented
PUN:	O	TTY: not implemented	PTP LCD display	UP1: RS-232C	UP2: not implemented
RDR:	I	TTY: keyboard	PTR: not implemented	UR1: RS-232C	UR2: not implemented
CON: output input	O I	TTY RS-232C keyboard	CRT: LCD keyboard	BAT: RS-232C RS-232C	UC1: RS-232C RS-232C



## 3.7 The CP/M Built-In Commands

Built-in commands are commands which can be executed simply by typing in their name on the command line. They do not show up as files on the directory (as do STAT and PIP, for example) but are part of CP/M itself. They are:

<b>DIR</b>	Display a list of files on a particular disk.
<b>ERA</b>	Erase a specified file or files.
<b>REN</b>	Rename a specified file.
<b>SAVE</b>	Save a portion of memory as a file.
<b>TYPE</b>	Display the contents of a file on the screen.
<b>USER</b>	Assign files to different users

### 3.7.1 DIR (Directory)

This command is used to display a list of file names stored on a disk. There are various ways of specifying the files depending on the results you want. These are achieved using wildcards (section 3.2.1). On a conventional CP/M computer the CP/M operating system is loaded into the computer from disk. It is not displayed on the directory. It is also possible to make other files non-directory files. This can be done in two ways. Files can be set (and reset) to 'SYS' files using the STAT program. These files do not appear on the directory of any user. Files can also be assigned to different users using the USER command, and then only appear on the directory when the particular user to which they belong is using the disk.

Points worth bearing in mind when using the DIR command are:

- There must be a space between DIR and the filename (if there is one).
- No space must be put between the drive name and filename if this is used.
- The files are not sorted into alphabetical order, and are usually displayed in the order in which they were first put on to the disk. If you require an alphabetical list it is best to use the STAT command.

#### a) List all files on the currently logged-in drive

The simplest way to do this is to type DIR on the command line:

```
C>DIR
C : PIP      COM : STAT  COM : SUBMIT  COM : XSUB  COM
C : CONFIG  COM : FILINK COM : TERM   COM
```

The same result can be obtained by being more long-winded and using the drive-name or wildcards:

```
C>DIR C:
C>DIR C:????????.*
```

#### b) List all the files on a drive other than that currently logged in

This can be done simply by adding the required drivename to the command:

```
C>DIR A:
A : PIP      COM
```

```
C>DIR B:
B : BASIC      COM
```

c) List all the files with the same extension on any drive  
Wildcards can be very useful when this is required. For example, to show all the COM files on the current drive:

```
C>DIR *.COM
C : PIP      COM : STAT  COM : SUBMIT  COM : XSUB  COM
C : CONFIG  COM : FILINK COM : TERM   COM
```

```
C>DIR A:*.BAS
NO FILE
```

This message informs you that there are no files conforming to that name on the A: drive. The message will appear whenever you ask for a DIRECTORY of a drive on which no files with the specified name reside.

```
C>DIR D:*.BAS
D : TEST      BAS : DEMO1  BAS : DEMO2  BAS
```

There may be other files on drive D: but as the request was only for those with the extension BAS so they are the only ones listed.

d) Check that a particular file is on a particular drive  
If you ask for a particular file by name it will be displayed on the screen. If it is not present the "NO FILE" message will be printed.

For example:

```
A>DIR B:TEXT1.DAT
B:TEXT1.DAT
A>
```

shows the file is on disk B :

```
A>DIR B:TEXT1.DAT
NO FILE
A>
```

shows the file is not present.

e) List all the files with the same name on any drive  
It may be that there are files with the same name but different extensions on the same drive, for instance assembler program files, so the asterisk wildcard can again be used:

```
D>DIR START.*
D : START    ASM : START    HEX : START    PRN
```

```
C>DIR D:START.*
D : START    ASM : START    HEX : START    PRN
```

f) List all files with similar names on any drive  
This is where you can really start to use the wildcards! For instance:

```
D>DIR S??????.*
```

will list all files on drive D: whose names begin with S of any extension;

```
D>DIR C : *.C??
```

will list all the files on drive C: whose extension begins with C;

```
A>DIR D:DEM??.BAS
```

will list all those files on drive D: whose names consist of five letters beginning with DEM and whose extensions are BAS.

g) Microcassette Tape DIRECTORIES

The Microcassette drive is always named drive H: and is used in the same way as the other drives.

```
H>DIR
H : TEST.BAS
```

If the tape is mounted, the directory will be read from the directory stored in memory, and so will be read immediately.

If the drive is unmounted when an access is made the drive will wind the tape to the beginning and mount the tape. The directory is then stored in memory. It may take a short while for the tape to be mounted.

It is better to keep to the habit of mounting and removing the tape through the System Display. Before carrying out a directory operation on the Microcassette drive, use the System Display to check if the tape is mounted. If it is not mounted, mount it through the System Display.



**WARNING:**

*NEVER try to obtain the directory of a Microcassette tape when another tape is already mounted. The directory of the second tape must be loaded into memory before the directory can be printed. This can only be done by removing the first tape and then mounting the second tape.*

**h) Pausing a DIRectory**

Either the **PAUSE** key or **CTRL** - **S** can be used to pause a DIRectory as it is displayed. If a directory has scrolled off the screen in screen modes 0,1 or 2, it can be seen by using the shift cursor keys ( **↑** and **↓** ).

**i) Printing a DIRectory**

To obtain a hardcopy of a DIRectory, use **CTRL** - **P** before issuing the DIR command or use the screen dump facility ( **CTRL** and the **PF5** key).

**j) Stopping the DIR**

If any key is pressed during execution of this command it is terminated and the CP/M command line reappears.

### 3.7.2 ERA (Erasing files)

Files can be erased at any time using the ERA command. The command is used in a similar way to the DIR command and the rules about spaces apply equally. It is possible to set files to be Read/Only files. Files which have been set to Read/Only cannot be erased. If you really do want to erase them you can use the STAT program to set them to Read/Write. In fact ERA does not really erase the file from the disk - it simply removes the directory entry for it.

Files on the ROM drives cannot be erased. If an attempt is made to erase a file on a drive which is set to Read/Only, one of the following error messages will be displayed:

**BDOS ERROR ON dr:FILE R/O**  
**BDOS ERROR ON dr:R/O**

will occur. The particular drive name "dr:" will be printed as the drive name accessed. If this does occur you should press **CTRL** - **C** , **STOP** or **RETURN** .

**a) Erase a specific file on any drive**

To do this you simply specify the full filename and extension of the file you wish to erase:

```
D>ERA TESTING.COM  
D>
```

```
C>ERA A:WASTE.FIL  
C>
```

If the file is not present the "NO FILE" message will be printed.

```
C>ERA DUDFILE.COM  
NO FILE  
C>
```



**WARNING:**  
**USING WILDCARDS WITH THE ERA COMMAND**

*It is very easy to erase files you do not wish to erase from a drive by using wildcards with the ERA command.*

*TO AVOID ACCIDENTS use the DIR command with the same wildcard option you wish to use. The files displayed by the DIRectory will then be the ones ERAsed. This will ensure that you know what your action will be without causing anything drastic to happen.*

*IF YOU ARE USING 'SYS' FILES these will be deleted by the ERA command, even though you did not see them on the directory. To avoid accidents use the STAT program to check the complete list of file names if you are using 'SYS' files.*

Alternatively delete files one at a time.

**b) Erase all files on a drive**

This is achieved by using the asterisk wildcard. Since this is drastic action, it results in a question from the PX-8 asking if you are sure you want to erase all the files:

```
A>ERA *.*  
ALL (Y/N)?
```

If you answer N execution will be terminated, and answering Y will result in all the files in the directory being ERAsed. Any files NOT in the directory will not be erased, that is system files and those with the SYS attribute set.

To erase all files on another drive the drivename must be specified:

```
C>ERA D: *.*  
ALL (Y/N)?
```

**c) Erase all files with the same filename on any drive**

The asterisk wildcard can be used to erase all the files on a particular drive, which have the same name but whose extensions differ.

```
D>ERA START.*
```

will erase all those files on drive D: whose filename is START and whose extension is anything, for instance, START.ASM, START.BAS or START.DAT.

```
C>ERA D:START.*
```

will have exactly the same effect but is executed when C: is the logged-in drive instead of D:.

**d) Erase all files with the same extension on any drive**

To do this you must specify the extension in full but use the asterisk wildcard to allow any filename:

```
C>ERA *.COM
```

will erase all the files on drive C: having the extension COM.

```
C>ERA D:*.BAS
```

will erase all files on drive D: with the extension BAS, that is, all BASIC program files.

**e) Erase all files with similar names or extensions on any drive**

Here you can use both the asterisk and the question mark wildcards. The asterisk replaces an entire filename or extension and the question mark replaces individual characters in the filename or extension:

```
C>ERA S?????.*
```

will erase all files on drive C: with any extension and whose filename consists of six characters beginning with S.

```
C>ERA D:TES?????.BAS
```

will erase all those files on drive D: whose extension is BAS and whose filename begins with TES.

```
D>ERA A:DEMO???.?A
```

will erase all files on drive A: whose filename consists of six letters beginning with DEMO and whose extension ends in A.

**f) Erasing Files on a Microcassette drive.**

On a conventional drive when a file is removed from the directory, the disk is opened up for saving other files. On a Microcassette tape the files are saved sequentially as a single block. This means that the removing a series of files from a tape directory does not free up the tape. It will simply removes the name

from the directory. If the space were freed, it would be possible for a new file to overwrite the start of a current file. If the file is the last file on the tape, the file space will be set free for further use.

To remove a block of files from the middle of a tape, and free the tape for new files,

- o Use the PIP program to transfer the files at the end of the tape into the RAM disk.
- o ERASE all files from the block you want to delete to the end.
- o Use the PIP program to replace the end files back on the tape.



**WARNING:**

*The directory is only updated in memory when disk access is carried out. Only when the tape is removed will the directory be written to tape. It is especially important to remove a tape using the correct procedure if the ERA command has been used on a Microcassette tape.*

### 3.7.3 **REN** (Rename a file)

**Format**    **REN** new filename = old filename

A file can be renamed at any time by giving this command followed by the new filename, an equals sign and the old filename. There must be a space between **REN** and the new filename. although it is optional either side of the equals sign.

```
A>REN NEW.COM = OLD.COM
A>
```

has changed the filename "OLD.COM" to "NEW.COM".

If the new filename you have chosen already exists on that drive, the message **FILE EXISTS** is displayed and the command line reappears:

```
A>REN TESTING.BAS = DEMO.BAS
FILE EXISTS
```

This indicates that there is already a file on drive A: called TESTING.BAS. This can also happen if you specify the names the wrong way round, since a test is carried out on the new name before carrying out the renaming.

If a drive name is specified it must be placed before new filename:

```
C>REN A:START.COM = TESTING.COM
```

This example will have the result of renaming a file on drive A: called TESTING.COM to be called START.COM.

Bear in mind that this command only renames the file, so the old filename disappears from the directory and is replaced by the new name while the file itself remains exactly the same. Another important thing to remember is that you can only rename files on one drive, for example:

```
C>REN A:DEMO.COM = B:TESTING.COM
```

is not allowed and will result in the message:

```
B:TESTING.COM?
```



If you want to copy a file from one drive to another and rename it at the same time use the PIP program.

### 3.7.4 **SAVE** (Save an area of memory)

First of all, remember that this is completely different to the SAVE command in BASIC. The only feature they have in common is that they store programs on disk, but there the resemblance ends. It enables blocks of memory to be saved as a file on a disk. Normally only programmers will use this command, but there is one case where it may be useful to even BASIC programmers which is described below.

The format for the SAVE command is very simple.

#### **SAVE n filename**

The filename consists of a drivename (if necessary), filename and extension, in the usual way.

The 'n' parameter denotes the number of "memory pages" to be saved and must have a space either side of it. The value is specified in decimal notation. One page consists of 256 bytes of data, so the basic method of calculating n is to divide the number of bytes you want to save by 256 and round the answer up to the next integer. For instance, if you want to save 300 bytes the number of pages (n) would be 2 because 300 divided by 256 is greater than one but less than two.

A frequent use of SAVE is to write to disk a program which has been patched using the DDT program (see section 3.10). When the program is loaded with DDT the address of the next available byte is shown as NEXT. You can calculate the number of pages to SAVE in the following way:

- 1) Round the value shown as NEXT up to the next highest page. Taking as an example if 2C5A were returned by NEXT in DDT, this would give 2D00 (remembering that you are working in hexadecimal notation).
- 2) Drop the zeros from the end of the rounded value and calculate the value of the remaining digits in decimal. In the example 2D becomes 45 decimal.
- 3) Since DDT loads programs at the beginning of the first page of memory, subtract 1 to allow for this, giving 44 in the example.

If the file we have just patched is called FREDDY.COM and lives on the D:drive, the command given to write it back to disk will be

**C>SAVE 44 D:FREDDY.COM**

#### a) The "SAVE # GO.COM" Trick

There is a special trick you can play on the PX-8 using the SAVE command which can be very helpful in various circumstances. When you exit a program it is still present in memory. For example when exiting BASIC using the 'system' command, the BASIC interpreter and the programs in the five program areas are still existant in memory as far as MENU is on. You may then remember that one of the programs should have been saved, and will be lost if you try to restart BASIC. Also you might load a program such as PIP, then having used it once to transfer a single file done nothing except a DIR or ERA or other minor operation, and then wish to use it again. In either case the program (the BASIC interpreter, PIP or whatever) can be recalled instantly if you know how, instead of having to be reloaded from disk or even tape. In the case of BASIC the five programs stored in memory can be recovered just as if you had not exited to CP/M. First you have to create a file called GO.COM then execute it when you wish to continue with the previous program by typing GO. To create this file all you have to do is type:

```
A>SAVE # GO.COM
```

or

```
A>SAVE # D:GO.COM
```

If you want it on the D: drive, for instance. what you are actually doing with this command is making a file of length 0 bytes and saving it on the specified drive. When you type GO at the prompt the program GO.COM (of length 0 bytes) is loaded into memory and executed. Because the file contains 0 bytes it does not overwrite the program already in memory, so in fact the program already there is executed instead.

### 3.7.5 TYPE (Displaying the contents of a file)

**Format**    **TYPE** filename

This command is used to display a file on the screen. There must be a space between TYPE and the filename. This is usually an ASCII file because any other data format results in a screenful of rubbish and is totally incomprehensible. This is because as the computer reads through the file it tries to convert all the hexadecimal numbers it finds into printable ASCII characters, and the only files which produce anything recognisable are those which are in ASCII format. For example, if you wanted to look at a BASIC program file using TYPE, it would have to have been SAVED in ASCII format using the "A" option in BASIC. Normally, BASIC programs are stored in compressed binary format which will produce some very strange results if it is TYPEd to the screen.

#### a) Displaying the contents of a file to the screen

The command is invoked either to show the contents of a file from the current drive:

```
A>TYPE TEXT.TXT
```

or for example:

```
C>TYPE D:DATAFILE.DOC
```

will display the contents of the file DATAFILE.DOC which is held on drive D:.

The only files which you can be sure will contain data in ASCII format are those with the extensions ASM, PRN and DOC or TXT. Others may be in this format, and it doesn't do any harm to look at them anyway if you want to in order to check. WILDCARDS cannot be used with the TYPE command.

If the file is not on the disk the filename will be repeated on the next line with a question mark after it, for example:

```
C>TYPE B:TEXT1.DOC  
B:TEXT1.DOC?
```

This would also happen if an attempt was made to use wildcard characters.

**b) Pausing the display of a text file.**

When the text is displayed on the screen, it may be going too fast to be read. You can pause the display by pressing the **PAUSE** or **CTRL** - **S** key. If you wish to see some of the text which has scrolled out of the real screen window, but is still present on the virtual screen, the shifted cursor keys ( **↑** and **↓** ) can be used when the display has been paused. To continue the display press any key other than the shifted cursor keys.

**c) Stopping the Display**

Pressing any key other than the **PAUSE** , **CTRL** - **S** or a shifted cursor key will cause the TYPE command to cease displaying the file. The PX-8 will then return to show the CP/M prompt.

**d) Printing a File using TYPE.**

It is possible to echo the file to the printer by typing **CTRL** - **P** before invoking the TYPE command, and so produce a hard copy of the text.

Small passages (i.e. of screen size) can be printed using the screen dump, by pressing **CTRL** and the **PF5** key.

**e) Problems with TYPE using non ASCII files.**

If an attempt is made to view a non text file, the screen may be covered with a garble of non displayable characters. When stopped, the CP/M prompt may not appear. In this case you will have to press the reset button, and in extreme cases initialize the system. See section 2.1.3 if this occurs.

### 3.7.6 **USER** (selecting a different user areas)

**Format**      **USER n**

If more than one person will be using the PX-8 it is a good idea to give each a user number. This has the effect of dividing the drives into different areas, each of which contains files belonging to a different user. Thus two users may label their areas 0 and 1, and they can both have files with the same names because their numbers are 'invisibly' added to the filenames in their respective directories. When the PX-8 is switched on it defaults to user area 0, that is, it assumes you are user number 0 unless you tell it otherwise. If you are user number 2, simply give the command

**C>USER 2**

and you will have immediate access to your own files. There must be a space between USER and the area number. If you want to look at a file in another user area you can either give the USER command again with the relevant number, or use the PIP program using the [G] option.

To find out which user numbers are active on a particular disk, use the STAT program.

## 3.8 Using Utilities and Application Program with CP/M

Computers are normally used to run applications programs, for example to write text using a word processor. These programs are loaded from the disk into the memory of the computer when they are needed. For this reason they are often called transient commands to contrast them with the built-in commands. Such programs are given the file extension .COM meaning a command type of file. They are executed by typing the name of the file without the extension as if they were a command. For example to use the file TERM.COM which is a program to use the PX-8 as a terminal, you would simply type the word TERM after the CP/M prompt.

### 3.8.1 Utilities

The PX-8 is provided with a ROM which contains a number of programs which are used for common day to day management of the computer, for example transferring files and changing parameters. They perform the following functions:

PIP is a program for copying files.

STAT is a program to determine the status of the disks and perform general housekeeping functions.

SUBMIT is used to process a sequence of commands.

XSUB is used to allow input from the keyboard with SUBMIT.

CONFIG is used to change system parameters which are not changed often.

TERM is a program to allow the use of the PX-8 as a terminal.

FILINK is used to transfer files between the PX-8 and other computers.

These programs are discussed in detail in the following pages.

Other programs are often provided with CP/M. These are discussed in section 3.10. Please consult your dealer for further information.

### 3.8.2 Application programs

You will also use programs such as Portable WordStar™ and Portable Calc™. They will have instruction manuals which will give you details of how to use them. Whatever media they are provided on, ROM, Microcassette tape or floppy disk, they will have a file in the directory which is a .COM file to run them. They are executed in the same manner as the utility programs. They are transient commands of a different kind.

### 3.8.3 PIP

The PIP program is used to copy files between peripherals, e.g. from disk to disk, disk to Microcassette tape, from a disk to a printer etc. The name comes from the initial letters of the words Peripheral Interchange Program.

PIP can be used with wildcards (see section 3.2.1) to transfer all types of files. It can also be used to perform other valuable functions such as:

- Remove part of a file either from the beginning, end or the middle.
- Convert all characters into upper case or lower case
- Join a number of files together
- Make a backup of a file under a different name
- Add sequential numbers to each logical line of text
- Reform the page length of a text file

There are two ways PIP can be used.

- 1) When only a single operation is required, PIP can be used by following the command 'PIP' with the command string of the operation you wish to carry out. On completion of the operation the CP/M prompt will be returned or the MENU displayed if it is switched on. When using PIP from the MENU the command string can be typed on the MENU command line, when PIP has been selected.
- 2) When a number of operations are to be performed, the PIP program can be loaded into memory. The CP/M prompt (A>, B> etc ) will be replaced by a PIP prompt indicated by an asterisk. The use of PIP in this manner is essentially the same as when only one operation is required.

When PIP is chosen from the MENU, unless an additional command string is given, the PIP screen will clear and the PIP asterisk prompt will appear.

Pressing the **RETURN** key, the **STOP** key or **CTRL** - **C** will return to the MENU screen or the CP/M system prompt whichever is set.

In using PIP the command is written to perform the operation to a particular drive or device, from a particular drive or device. The location of the file being moved always comes last.

#### 1. COPYING A FILE FROM DRIVE TO DRIVE

Any file with a read/write attribute can be copied from one disk to another. For example,

```
C>PIP H:=A:INFO.DAT
```

This copies the file DEMO.DAT from drive A : to drive H: then returns to the system prompt. The same filename will appear on the new drive. The filename can be changed, for example:

```
C>PIP H:NEWNAME.DAT=A:INFO.DAT
```

By using this option to copy the file to the same drive, it can be used to make a backup of a file under a different name.

Various options can be added to the end of the PIP command string. These are listed starting from item 5. For example specifying the [V] option causes the file to be verified as it is copied:

```
C>PIP H:=A:INFO.DAT[V]
```

#### 2. COPYING A FILE FROM DISK TO PRINTER

This has almost the same effect as pressing **CTRL** - **P** then asking the computer to TYPE a file:

```
C>PIP LST:=A:LETTER.TXT
```

The file will be printed character by character, and words which come at the end of a line are likely to be split in the middle of the word.

This command can also be used to list a BASIC file to a printer if you are not in BASIC and do not have the BASIC ROM installed.

```
C>PIP LST:=A:DEMO.BAS
```

The BASIC program is sent to the printer as if you had LLISTed it while in BASIC. You MUST have saved the BASIC program as an ASCII file using the "A" option in BASIC.

A refinement of this command is to add the option [Pn] where the printer will



execute a form feed every n lines. This can be used to prevent lines being printed on the perforations. For example:

```
C>PIP LST:=A:INFO.DAT[P60]
```

will force a form feed every 60 lines giving the same effect as a one-inch skip-over on paper which takes 60 lines.

Another addition could be to echo the file to the screen at the same time as it is printed. This is achieved using the [E] option:

```
C>PIP LST:=A:INFO.DAT[E]
```

If both options are required, the command would be:

```
C>PIP LST:=A:INFO.DAT[P60E]
```

### 3. COPYING A FILE FROM DISK TO SCREEN

The use of this command will have the same effect as the [E] option when copying from and to any other device. It is also the same as using TYPE command. The format is:

```
C>PIP CON:=A:MEMO.DOC
```

One of the options that can be specified here to good effect is [N] or [N2]. This causes line number to be added at the beginning of each line in the form 01 if [N] is specified and 000001 if [N2] is specified:

```
C>PIP CON:=A:MEMO.DOC[N2]
```

In addition, all lower case characters can be converted to upper case, and vice versa, using the [U] and [L] options:

```
A>PIP CON:=A:MEMO.DOC[N2U]
```

will give

```
C>PIP CON:=A:MEMO.DOC[N2U]
000001 THIS IS A LINE OF TEXT
000002 HAVE A NICE DAY
```

```
C>
```

and

```
A>PIP CON:=A:MEMO.DOC[NL]
```

will give

```
C>PIP CON:=A:MEMO.DOC[NL]
1: this is a line of text
2: have a nice day
```

```
C>
```

### 4. COPYING A FILE TO AN EXTERNAL DEVICE

Files can be copied to external devices such as printers, disk drives and other computers by specifying the relevant output port as the destination. For instance:

```
A>PIP TTY:=A:DEMO.BAS
```

will copy the file to a printer or any other device connected to the high-speed serial output port.

### 5. OPTIONAL PARAMETERS AVAILABLE WITH PIP

There are nineteen options which can be used with the PIP command, including those which have been mentioned already:

#### 1. Block mode transfer

**Format** [B]

This option causes the data to be copied across in blocks, that is, 256 bytes at a time. When the buffer has received a block it sends the data to the destination device before allowing the source device to send it the next block. The actual character it recognises as an end-of-block marker is ASCII code 19, that is, **CTRL** - **S** or X-OFF.

**Example** A>PIP TEST.FIL=CRT:[B]

#### 2. Echo to screen

**Format** [E]

Giving this option will cause the screen to display all the data being transferred so that you can see what is being copied across during the process.

**Example** A>PIP LST:=TEST.FIL[E]

It is advisable to use the [V] entry, i.e. to verify the file when using the [E] option in copying files.

### 3. Form feed insertion

**Format** [Pn]

This makes it possible to force a form feed every n lines. It is particularly useful when sending files to the printer, allowing it to skip the perforations in continuous stationery to improve legibility. If n is omitted or given as 1 PIP assumes a form feed is required every 60 lines (that is, the default setting). If your paper requires 66 lines per page use the parameter as in the following example:

```
A>PIP LST:=REPORT.TXT[P66]
```

If used with the next parameter, [F], the [F] should come first:

```
A>PIP LST:=REPORT.TXT[FP66]
```

### 4. Form feed suppression

**Format** [F]

This is most useful when using PIP with a printer. It suppresses the form feed character (ASCII code 12 or hex 0C) which would otherwise cause the printer to feed a sheet through when it encounters the code. It is as well to remember that this code is also the 'clear screen' code, so it is advisable to take care when using this option.

**Example** A>PIP TTY:=TEXT.DOC[F]

In conjunction with [Pn] it can be used to reform page lengths, by copying a file, removing previously added form feeds (e.g. ones inserted by PORTABLE WORDSTAR) and adding them at a different place.

### 5. Hex format

**Format** [H]

When this option is specified the data is checked to ensure it is in Intel HEX format. If there is a discrepancy it terminates the transfer.

**Example** A>PIP DEMO2.HEX = DEMO1.HEX[H]

### 6. Ignore NULL records

**Format** [I]

This can be used as an alternative to [H] - it causes PIP to ignore NULL records (hex 00) and ensures the data is in Intel HEX format. It is therefore an extension of the [H] option.

**Example** A>PIP DEMO2.HEX = DEMO1.HEX[I]

### 7. Lower case conversion

**Format** [L]

Using this option will convert all upper case characters into lower case as they are transferred.

**Example** A>PIP LST:=C:LITTLE.DOC[L]

### 8. Numbering lines

**Format** [N]  
[N2]

When sending programs to another device it may be useful to have the lines numbered. PIP regards a line as a series of characters terminated by a carriage return (ASCII code 13 or 0DH). Specifying [N] will begin the file at column 9, with a colon(:) at column 7 followed by a space. The number of the line will be placed before the colon, as the following example shows:

If the file TEST.DOC contains the following information:

```
This is line one  
This is line two  
*  
This is line ten  
*  
This is line one hundred
```

where the text would also correspond to the numbers of the lines as seen by PIP, and the asterisks refer to further lines of text. If the file is transferred to from drive D: to drive A: using the [N] option, using:

```
C> PIP A:=D: TEST.DOC[N]
```

The file on drive A: will have the lines numbered as follows:

```
1: This is line one
2: This is line two
   *
10: This is line ten
   *
100: This is line one hundred
```

Similarly specifying [N2] fills in all leading spaces with a zero and replaces the colon with a space. For example if the following command is given:

```
C>PIP A:=D: TEST.DOC[N2]
```

the file on drive A: will have the lines numbered as follows:

```
000001 This is line one
000002 This is line two
      *
000010 This is line ten
      *
000100 This is line one hundred
```

If you wish to display the file with numbered lines on the screen, use:

```
C>PIP CON:=D: TEST.DOC[N]
```

or

```
C>PIP CON:=D: TEST.DOC[N2]
```

## 9. Object file transfer

**Format** [O]

Normally PIP can only copy standard ASCII or HEX files, but using this option allows it to transfer other types of files. Its effect is to ignore the physical end-of-file code a CTRL-Z (ASCII 26 or 1A hex) wherever it occurs in the object file, because in this context it will not be signalling the end of the file.

**Example** A>PIP B:=OBJECT.FIL[O]

It is not necessary to use this optional parameter with a COM file, because PIP will automatically add it. However, when using it with other machine code or object files which do not have the file extension COM, the [O] parameter MUST be used.

## 10. Read system files

**Format** [R]

This option makes it possible to read and copy system files, that is, files which do not appear in the directory and those with a filetype of SYS. It automatically sets the [W] option.

**Example** A>PIP B:=A:OSTAB.SYS[R]

## 11. Stop copying at specified string

**Format** [Qstring^Z]

(^Z means type a **CTRL** - **Z** by holding the **CTRL** key down while pressing the Z key.)

If you only want to transfer part of a file, giving this option will cause PIP to copy the file until it finds the specified string. Only text before the string will be copied.

**Example** A>PIP LST:=B:REPORT.DOC[QTHE END^Z]

If you want to search for a string containing lower case characters this can only be used from the PIP \* prompt. This is because CP/M converts everything typed on the command line into upper case including the string you have given and PIP will not be able to find it, because it is searching for the upper case equivalent of your string.

**Example** \*LST:=B:REPORT.DOC[QThe End^Z]

It can be used with the next option to copy a section from the middle of a file.

## 12. Start copying at a specified string

**Format** [Sstring^Z]

This behaves in much the same way as the [Qstring^Z] option, except that it starts copying from the end of the specified string. The same conditions apply

if you want to detect lower case characters.

**Example** A>PIP CHAPTER1.DOC=CHAPTER1.DOC[SIN-TRODUCTION^Z]

The two options can be used together if you want to copy a piece of data from the middle of a file.

**Example** \*CHAPTER1.DOC=CHAPTER2.DOC[SIntroduction ^ZQlast.^Z]

### 13. Tab settings

**Format** [Tn]

The tab settings on the copied file can be changed from those of the original using this option. The number given by n puts tabs at columns n, 2n, 3n etc. For instance, [T9] will give tabs at columns 9, 19, 29, 39 and so on. These settings will be used wherever PIP comes across a TAB character in the file it is copying. The TAB character is **CTRL** - **T** (ASCII code 9).

If the wordprocessor, or whatever created the text file, inserts spaces instead of using the TAB character, this option will have no effect.

**Example** A>PIP CON:=B:PROGRAM.ASM[T10]

### 14. Transfer between user areas

**Format** [Gn]

Normally it is not possible to use files from another user area. However, specifying this option will allow transfer of files from user area n to the current area.

**Example** C>PIP A:TEST.DOC=A:DEMO.DOC[G3]

### 15. Truncate lines of data

**Format** [Dn]

This option allows truncation of lines past column n, that is, PIP will delete all characters between column n and the next carriage return.

**Example** A>PIP LST:=PROGRAM.BAS[D80]

### 16. Upper case conversion

**Format** [U]

This works in the opposite sense to the [L] option in that it converts all lower case characters to upper case.

**Example** A>PIP TTY:=BIGTYPE.TXT[U]

### 17. Verify the copy

**Format** [V]

Verifying a file as it is copied acts as a double check on its integrity. When given this option PIP compares the copy it has made with the original as it goes along, ensuring a faithful reproduction containing only those errors that were in the original.

**Example** A>PIP B:=PERFECT.COM[V]

### 18. Write to a read/only file

**Format** [W]

If you have files which have been set to read-only using STAT, you can copy over them if you give this option without the computer asking you if you want the existing file erased and overwritten. As it is not possible to reverse the process if you make a mistake, use with care!

**Example** A>PIP A:=B:SECURE.COM[W]

### 19. Zero parity setting

**Format** [Z]

The leftmost bit of a byte is usually the parity bit, and this option sets all these high-order bits to zero. It results in converting 8-bit ASCII bytes to 7-bit ASCII bytes. It is as well to ensure none of the characters you are sending use this bit because otherwise you could get some unexplained results. For example graphics characters would be changed, and some console ESC codes would be changed.

**Example** A>PIP C:=B:ORDINARY.FIL[Z]

## USING THE PIP \* PROMPT

If you want to transfer a large number of files or transfer files between two data disks, it may be more convenient to give the PIP command on its own, producing the \* prompt:

The command string as described above can then be typed for each operation. For example to copy all the COM files to drive A: from drive C:, the display would show:

```
C>PIP
*E:=C:*.COM

COPYING -
PIP.COM
STAT.COM
SUBMIT.COM
XSUB.COM
FILINK.COM
TERM.COM
CONFIG.COM
*
```

Or to output a file to the screen:

```
C>PIP
*CON:=MEMO.DOC
```

At the \* prompt you can type in any PIP commands you like without having to load PIP on every operation. This can save a great deal of time if there are a number of operations to be carried out.

Once you have finished, pressing the **RETURN** key, the **STOP** key or **CTRL - C** will return you to the system prompt on the drive form which you entered PIP (the default drive), or to the MENU if it is switched on.

## 3.8.4 STAT

The STAT program is used to find out the STATistics or STATus of the various disk drives. This makes it the most frequently used of the utility programs. Many CP/M users do not use the full facilities provided by STAT. The following information is therefore provided as a list of operations, with a summary at the end.

### STAT and the Microcassette Drive

Because the Microcassette drive is not a standard CP/M disk device, but is a sequential tape device, the STAT program cannot determine the correct data for space on the Microcassette Drive. Also file name lengths may not be accurate.



## USE OF THE STAT PROGRAM

### 1. CHANGE DEVICE ASSIGNMENTS

**Format** STAT logical: = physical:

The device assignments can be altered with this command. However, it is more likely that the CONFIG program would be used instead since the devices are named in real terms rather than as codes.

For those familiar with CP/M who are used to using the STAT command, the following table shows the correspondence between the physical devices and those implemented:

Logical device	Physical device			
LST:	TTY: Serial (printer)	CRT: LCD display	LPT: RS-232C	UL1 not implemented
PUN:	TTY: not implemented	PTP: LCD display	UP1: RS-232C	UP2: not implemented
RDR:	TTY: Keyboard	PTR: not implemented	UR1: RS-232C	UR2: not implemented
CON: Output Input	TTY: RS-232C Keyboard	CRT: LCD Keyboard	BAT: RS-232C RS-232C	UC1: RS-232C RS-232C

For instance, to tell the computer that the printer is now attached to the serial port instead of the RS232-C port, the command

**C>STAT LST : = TTY :**

is given, after which all output destined for the printer is sent to the serial port instead of the RS-232C port. Then, if the STAT DEV: command is given, the result will be:

```
C>STAT DEV:
CON: 1s CRT:
RDR: 1s UR1:
PUN: 1s UP1:
LST: 1s TTY:

C>
```

Further details of the physical devices assigned to the logical devices are given in section 3.6.

### 2. DISK CHARACTERISTICS

**Format** STAT DSK:

The complete status of a disk is displayed using this command. It shows the status of both the current disk and that of any others that have been accessed during the same session:

This command displays the characteristics of the disks accessed, such as its capacity for example:

**A>C:STAT DSK:**

<b>A:</b> Drive characteristics	(drive name)
<b>72:</b> 128 Byte Record Capacity	(no. of 128 byte records allowed)
<b>9:</b> Kilobyte Drive Capacity	(formatted capacity of drive)
<b>16:</b> 32 Byte Directory Entries	(no. and size of directory entries)
<b>0:</b> Checked Directory Entries	(no. of checked directory entries)
<b>128:</b> Records/Extent	(no. of records per extent)
<b>8:</b> Records/Block	(no. of records per block)
<b>64:</b> Sectors/Track	(no. of sectors per track)
<b>0:</b> Reserved Tracks	(no. of tracks reserved for CP/M)

### 3. HELP

**Format** STAT VAL:

This acts as a sort of HELP command. It shows the formats of the various STAT commands which can be given to obtain information or alter device attributes and assignments:

C>STAT VAL:

Temp R/O Disk: d: = R/O  
Set Indicator: d:filename.typ \$R/O \$R/W \$SYS \$DIR  
Disk Status: DSK: d:DSK:  
User Status: USR:  
I/O byte Assign:  
CON: = TTY: CRT: BAT: UC1:  
RDR: = TTY: PTR: UR1: UR2:  
PUN: = TTY: PTP: UP1: UP2:  
LST: = TTY: CRT: LPT: UL1:

#### 4. DISPLAY DEVICE ASSIGNMENTS

**Format** STAT DEV:

This gives the details showing which logical devices have which physical devices assigned to them. To check the devices by name, use the CONFIG program.

```
C>STAT DEV:  
CON: is CRT:  
RDR: is UR1:  
PUN: is UP1:  
LST: is LPT:  
C>
```

#### 5. READ/ONLY - PROTECT ALL THE FILES ON A DISK

**Format** STAT drivename: = R/O

An entire disk can be set to read/only with this command. It remains effective until either a warm or a cold start is made:

C>STAT A: = R/O

The Read/Only command protects the files on a disk so that they cannot be erased or written to. If you try to write or erase a file with a R/O attribute set the error message:

```
A>ERA LETTER.DOC  
BDOS ERROR ON A:R/O
```

will be displayed.

Pressing either the **RETURN** key or the **STOP** key or **CTRL** - **C** will return

you to the prompt.

It is also possible to protect single files and unprotect them using STAT.

#### 6. READ/ONLY - PROTECT A SPECIFIED FILE

**Format** STAT drivename:filename.filetype \$R/O

Any file can be set to read/only using this format. This will prevent any alteration of the file until it is reset to read/write:

```
C>STAT A:DOCUMENT.TXT $R/O
```

DOCUMENT.TXT set to R/O

#### 7. READ/WRITE - REMOVING PROTECTION

**Format** STAT drivename:filename.filetype \$R/W

This resets a file to allow it to be written to as well as read from.

```
C>STAT A:DOCUMENT.TXT $R/W
```

DOCUMENT.TXT set to R/W

If a file is protected, and subsequently the whole disk is protected, the file will still be protected when a warm or cold boot is made to remove the protection from the disk.

#### 8. HIDE FROM DIRECTORY - SPECIFIED FILE

**Format** STAT drivename:filename.filetype \$SYS

It is possible to give a file SYStem status using this command. This effectively removes its name from the directory so it cannot be used by anyone who does not know it exists:

```
C>STAT A:DOCUMENT.TXT $SYS
```

DOCUMENT.TXT set to SYS

If a STATus is carried out on the file when it is set to a SYS file it shows with the filename in brackets

C>STAT A:DOCUMENT.TXT

Recs	Bytes	Ext	Acc	
41	6k	1	R/O	A:(STAT.COM)

Bytes Remaining On A: 3k

### 9. REPLACE IN DIRECTORY - SPECIFIED FILE

**Format** STAT drivename:filename.filetype \$DIR

This countermands the previous format, resetting the file so that it shows on the directory:

C>STAT A:DOCUMENT.TXT \$DIR  
DOCUMENT.TXT set to DIR

### 10. SIZE AND ATTRIBUTES - SPECIFIED FILE

**Format** STAT drivename:filename.filetype

The size and attributes of the specified file are displayed using this command. It gives specific information about the number of records (Recs), number of bytes (Bytes), number of extents (Ext) and read/write status (Acc) of each file on the disk, then the total number of unused bytes remaining. A complete file name can be given, or wildcards can be used to specify a number of files:

#### a) Information on a particular file

C>STAT A:TESTING.COM

Recs	Bytes	Ext	Acc	
16	4k	2	R/O	A:TESTING.COM

Bytes Remaining On A: 4k

#### b) Information on all files

C>STAT C:\*. \*

Recs	Bytes	Ext	Acc	
64	8k	1	R/W	C:CONFIG.COM
22	3k	1	R/W	C:FILINK.COM
58	8k	1	R/W	C:PIP.COM
41	6k	1	R/W	C:STAT.COM

10	2k	1	R/W	C:SUBMIT.COM
24	3k	1	R/W	C:TERM.COM
6	1k	1	R/W	C:XSUB.COM

Bytes remaining on C: 33k

#### c) All files with a particular extension

C>STAT A:\*.COM

Recs	Bytes	Ext	Acc	
58	8k	1	R/W	A : PIP.COM

Bytes remaining on A: 0k

#### d) Files containing particular characters

A>STAT D:DEMO??.BAS

Recs	Bytes	Ext	Acc	
20	3k	1	R/W	D:DEMO1.BAS
16	2k	1	R/O	D:DEMO13.BAS
14	2k	1	R/W	D:DEMO1A.BAS

Bytes remaining on D: 258k

### 11. SIZE AND ATTRIBUTES - SPECIFIED FILE

**Format** STAT drivename:filename.filetype \$\$

Using this form of the command will give the same information as without the \$\$ option, with the addition of the size of the file. This value is the same as the number of records for sequential access files, and is generally used to show the amount of space that has been reserved for a random access file. This is because a sequential file simply takes up space as it is added to, and a random access file has an amount of space allocated to it when it is created:

```
C>STAT A:DOCUMENT.* $$
Size Recs Bytes Ext Acc
1 1 1k 1 R/W A:DOCUMENT.BAS
1 1 1k 1 R/W A:DOCUMENT.TXT
Bytes Remaining On A: 6k
C>
```

## 12. SPACE REMAINING ON DISK

**Format** STAT

Giving this command at the prompt will result in a display of the amount of space available on the current drive and on any other drives which have been accessed during the current session. It also shows the read/write attribute:

a) C>STAT

C: R/O, Space: 33k

b) C>STAT

A:R/W, Space: 0k

C:R/O, Space: 33k

H:R/W, Space: 24k

## 13. SPACE REMAINING ON SPECIFIED DISK

**Format** STAT drivename:

This form of the command gives the amount of space remaining on specified drive:

C>STAT A:

Bytes Remaining On A: 0k

## 14. USER STATUS

**Format** STAT USR:

This format is used to display the number of the current user and the numbers of the users who have active files on the disk:

A>STAT USR:

Active User : 0

Active Files : 0 1

This gives the information that the current user is number 0 and that users 0 and 1 both have active files on the disk.

## SUMMARY OF STAT COMMANDS

STAT	SPACE REMAINING ON DISK
STAT DEV:	LOGICAL TO PHYSICAL ASSIGNMENTS
STAT drivename:	SPACE REMAINING ON SPECIFIED DISK
STAT drivename: = R/O	READ/ONLY - SET SPECIFIED DISK
STAT drivename:filename.filetype	SIZE AND ATTRIBUTES - SPECIFIED FILE
STAT drivename:filename.filetype \$DIR	REPLACE IN DIRECTORY - SPECIFIED FILE
STAT drivename:filename.filetype \$R/O	READ-ONLY - SET SPECIFIED FILE
STAT drivename:filename.filetype \$R/W	READ/WRITE - SET SPECIFIED FILE
STAT drivename:filename.filetype \$\$	SIZE AND ATTRIBUTES - SPECIFIED FILE
STAT drivename:filename.filetype \$\$SYS	REMOVE FROM DIRECTORY - SPECIFIED FILE
STAT DSK:	DISK STATUS
STAT logical: = physical:	CHANGE DEVICE ASSIGNMENTS
STAT USR:	USER STATUS
STAT VAL:	HELP

## 3.8.5 SUBMIT/XSUB

Two very useful commands available in CP/M are SUBMIT and XSUB. Suppose you wish to execute the same commands in a certain sequence at various times. Instead of having to type in the commands one by one you can put them all into a special file with the extension .SUB. Then when you want that series of commands executed you can simply type

```
C>SUBMIT filename
```

and they will all be executed one by one. You can use the XSUB command to allow for input from the keyboard when running programs listed in the SUBMIT file by inserting it at the beginning of the file.

The most useful aspect of having a SUBMIT file facility in CP/M on the PX-8 is to enable a whole sequence of files to be run from an AUTOSTART or WAKE string. Section 2.2.3 shows how to set up these strings.

### 1. SUBMIT

First of all you need to make a list of the programs you want to run. These must all be programs which will return to the CP/M system prompt when they end because otherwise the next command in the sequence cannot be executed. It is possible to run other types of programs, or ones which require input from the keyboard, providing you are present to type in at the keyboard when necessary.

The files containing the commands can be created in a variety of ways - any word processing program such as Portable WordStar™, a BASIC program or using the CP/M ED program if you have it.

Here is an example of a simple batch of jobs to be carried out:

1. Display a directory of the disk on the D: drive;
2. Show the sizes of the files;
3. Erase all those with the extension .BAK (that is, back-up files);
4. Move all BASIC files to the disk in the E: drive.

The sequence of commands to do this would be:

```
DIR D:  
STAT D:*.*
```

```
ERA D:*.BAK  
PIP E:=D:*.BAS
```

If this series of commands were held in a file called START.SUB the command to execute them will be:

```
A>SUBMIT START
```

The commands will be executed one by one, and each one will be shown on the screen before it is run, just as if you had typed it in from the keyboard.

If you want to run a BASIC program from SUBMIT, remember that when such a program ends it returns to the BASIC 'Ok' prompt instead of to CP/M, so the next command in the SUBMIT file will not be executed. One way round this is to insert a line in the program:

```
65000 SYSTEM
```

This will ensure a return to CP/M at the end of the program.

If one or more of the programs you want to be executed may vary, it is possible to insert variables into the SUBMIT file which can be substituted when the submit program is run. The variables symbols are inserted in the form "\$1", "\$2", "\$3" etc, where "\$1" is the variable symbol used for the first substitution, "\$2" for the second and so on. The values of the variables are substituted according to the following syntax:

```
C>SUBMIT filename X1 X2 X3.....
```

where X1 will substitute all values of "\$1", X2 all values of "\$2" and so on.

For instance if the SUBMIT file BEGIN.SUB contained

```
DIR D:*.BAS  
B:BASIC $1  
STAT D:*.BAS
```

the command given could be

```
C>SUBMIT BEGIN D:PROG1
```

This has the effect of inserting the filename you have given (D:PROG1) into the variable "\$1" in the SUBMIT file. The result would be to display a directory of the disk in drive B: and then load BASIC from drive B: before running the BASIC program PROG1 on the disk in drive D: and then showing the BASIC files on drive D:.

There can be any number of variables in the SUBMIT file, and you must ensure that the filenames you give are in the correct order to be substituted at the appropriate time. For instance, if you have a SUBMIT file TESTING.SUB containing the following commands:

```
DIR
$1
DIR
$2
STAT
$3
DIR
```

the command you give to execute it with the .COM files TESTA, TESTB and TESTC in that order should be:

```
C>SUBMIT TESTING TESTA TESTB TESTC
```

so that they are substituted in the correct order.

Some CP/M programs (PIP for example) use the "\$" character in their subcommands, and if this will mean they will be interpreted as variables. Consequently care must be taken as follows to overcome this by adding an extra "\$" whenever the subcommand uses a "\$". SUBMIT will then ignore the first one and then treat the second one as normal. For example to insert the command to cause STAT to set all files on disk A: as Read/Only use:

```
STAT A:*. * $$R/O
```



**WARNING:**

*The SUBMIT program writes a temporary file when it runs. This file will be written to drive A: and if the disk is write protected, an error will be generated and you will not be able to proceed further. Similarly if the RAM disk A: is set to zero bytes a "DIRECTORY FULL" error will be generated.*

### 1.1. Creating a SUBMIT file in BASIC

Since the PX-8 is not supplied with the CP/M program ED, and you may not have a wordprocessor program, BASIC may be the only means of creating a SUBMIT file. This can be done very simply by creating a sequential text file and writing the commands to it as data. Chapter 5 of the BASIC Reference Manual gives details of creating text files. For instance, the series of commands given above could be put into the file called START.SUB with the following program:

```
10 OPEN "O", #1, "D:START.SUB"
20 PRINT #1, "DIR D:"
30 PRINT #1, "STAT D:*. * "
40 PRINT #1, "ERA D:*.BAK"
50 PRINT #1, "PIP E:=D:*.BAS"
60 CLOSE
70 END
```

Or, put another way:

```
10 OPEN "O", #1, "D:START.SUB"
20 READ A$
30 IF A$="END" THEN 60
40 PRINT #1, A$
50 GOTO 20
60 CLOSE
70 END
80 DATA "DIR D:","STAT D:*. *","ERA D:*.BAK";PIP E:=D:*.BAS";END"
```

Either of these programs will create a file called START.SUB containing the commands you want which can then be executed from the CP/M prompt by typing:

```
C>SUBMIT d:START
```

You could even write a utility program in BASIC to allow you to create a SUBMIT file at any time by putting in the commands as you go and then storing them in a .SUB file of your choice:

```
10 INPUT "Type in drivename and filename ";F$
20 OPEN "O", #1, F$ + ".SUB"
30 INPUT "Enter command or END ";A$
```



```

40 IF A$ = "END" THEN 70
50 PRINT #1, A$
60 GOTO 30
70 CLOSE
80 END

```

### 1.2 Creating a SUBMIT file with Portable WordStar™

You can create a SUBMIT file with any word processing program, but the example given here deals specifically with Portable WordStar™. In fact the procedure will be almost identical regardless of which word processor you use.

First of all you must create a new document, giving it the name.

Filename.SUB

If you do not give the extension .SUB the program will not allocate the extension .SUB and the file you have created will not work with SUBMIT.

Now all you have to do is type in the commands you want, one on each line and each followed by a carriage return:

```

A:FILENAME.SUB PAGE 1 LINE 5 COL 01
L-----!-----!-----!-----!-----!-----!-----R
DIR D : 
STAT D:*. * 
ERA D:*.BAK 
PIP E:=D:*.BAS 

```

Once you have saved this back to disk you will be able to execute it in the normal way using the SUBMIT command.

### 1.3 Adding Comments to SUBMIT Files

In many cases SUBMIT files will be short and easily understood. As you use SUBMIT more, you will realise how powerful a program it is and build up more complex files. It is often difficult to understand why they were created in a specific fashion, and also if someone else wishes to understand what you have done, they may not be able to follow your file. This may be overcome by adding comments to the SUBMIT file which are not executed, but can be found by loading the SUBMIT file into Portable WordStar™ or another word processor, or using the TYPE command of CP/M to read the file. Comments can be added

by preceding the comment by a semicolon.

For example:

```

DIR D: ;first show a directory of disk in drive d:
STAT D:*. * ;then show the statistics of the files on the disk
ERA D:*.BAK ;and erase all backup files on drive d:

```

## 2. XSUB

Some CP/M programs expect information to be typed in at the keyboard when the program has been loaded. The CP/M utilities DDT and ED are two such programs. They can only be used in this way. In order to use them with SUBMIT it is necessary to use the XSUB program with SUBMIT.

If the PIP program is to be used to carry out a number of operations, it is normally loaded first and then the individual operations typed from the keyboard after the PIP asterisk prompt (\*). These subcommands can be included in a SUBMIT if the XSUB program is included also. To use XSUB it is typed into the SUBMIT file as the first command. For example, the operations

1. Move the files to be updated from the D: drive to the E: drive;
2. Execute the program A:UPDATE.COM;
3. Move the updated files to the H: drive;
4. Move the original files to the E: drive as back-up files;
5. Erase the original files from the D: drive.

would be executed by a SUBMIT file containing the following commands:

```
XSUB
PIP
E: = D:RECORDS.DAT
^C
A:UPDATE
PIP
H: = E:RECORDS.DAT
E:RECORDS.BAK = D:RECORDS.DAT
^C
ERA D:RECORDS.DAT
```

While this file is executing you would see the PIP \* prompt appear when PIP was being executed, and the names of the files being moved exactly as you entered them in the SUBMIT file, as would be the case if you typed them from the keyboard. The fourth and ninth commands are 'C' which means "CTRL-C" — this is the command given to exit from PIP, and has to be given in the SUBMIT file to allow return to the CP/M prompt. An alternative is 'M' (CTRL-M) which is a carriage return and has the same effect of leaving PIP and going back to CP/M. When PIP has exited to the CP/M command line, the program UPDATE will be executed. Such a program MUST exit normally to the CP/M command line when being used in a SUBMIT file. When the program UPDATE has finished, PIP can carry out a further transfer of records.

## 3.8.6 CONFIG

The CONFIG program is used to set those system parameters which are not changed very often. It is complementary to the System Display, and the current values of some of the parameters changed by the CONFIG program are shown on the System Display. The CONFIG program can also be used to check the settings of parameters not shown on the System Display.



### WARNING:

*Do not switch off the PX-8 (either manually or by allowing the auto power off to be activated) after changing the RAM disk or USER BIOS size without exiting from the CONFIG program. The full configuration can only be carried out by CP/M. If the PX-8 is switched off, the RAM disk could be destroyed and it may also be necessary to re-initialise the system. You will see repeated message of "RAM disk format (Y/N)" until you press Y(es) key.*

When you have entered CONFIG, either from the MENU or CP/M command line, the screen will show the following:

```
*** MAIN MENU ***                CONFIG V1.0
      Select alphanumeric or ESC to exit.
1=auto power off                 7=RAM disk
2=CP/M function key              8=RS-232C
3=cursor & function key display  9=screen mode
4=date & time                     A=serial
5=disk drives                    B=user BIOS
6=printer                         C=country
```

The CONFIG program is used in a similar way to the System Display, but with 12 options. As with the System Display the ESC key is used to move back to the main menu and to exit.

### 1. AUTO POWER OFF FUNCTION

Press the 1 key from the main CONFIG menu to change the auto power off time. The screen will clear and change to:

```
*** AUTO POWER OFF ***
Set time(1 to 255) in minutes or ESC to return (0 disables auto power of
Auto power off time ?

auto power off      : 10
```

The value shown against the semi-colon on the fifth line is the current setting in minutes for the auto power off time. This is the time for which the PX-8 will wait before switching itself off in continue mode if no key is pressed. It is advisable to keep the time short, otherwise the battery will run down. The reason for having such a system is to conserve the battery.

As can be seen from the display, the time can be set in intervals of one minute (up to 255 minutes) or switched off altogether using the value 0. When you have entered the time you require, press RETURN. The chosen time will be displayed on the fifth line, and you can use the ESC key to return to the main CONFIG menu.

## 2. The CP/M FUNCTION KEY ASSIGNMENTS

When the 2 key is pressed on the main CONFIG menu, the display changes to:

```
*** CP/M FUNCTION KEY ***
Select function key number (PF10=0) or ESC to return.

PF1  dir
PF2  type
PF3  stat
PF4  pip
PF5  config^M

PF6  dump
PF7  submit
PF8  ddt
PF9  term^M
PF10 filink^M
```

These are the strings assigned to the programmable function keys on the top of the keyboard. A description of the use of the keys is given in Chapter 2 section 2.2.1c. The screen above shows the default setting, i.e. the strings which will be assigned when the system is initialized or a reset is performed.

Note that when shown on the eighth line of the CP/M screen, some of the strings terminate in the ↵ character. In setting them up with the CONFIG program, this carriage return is shown as “^M”. These strings have a carriage return (CTRL-M) added to them because there is no possibility that any more characters need to be typed after the string. For example the PF5 key can be used to run the CONFIG program as follows. On the CP/M command line type the drive name where the CONFIG program is located (e.g. C:) and then press the PF5 key. All the letters of the word CONFIG will appear, and the carriage return will be ‘typed’ also. In a few seconds the CONFIG main menu will appear on the screen.

The CTRL-M is not added to all commands since some of them might possibly require extending. For example with DIR you might want to add the name of

a drive other than the current one to find out the files on that particular drive instead of changing the logged in drive first. When the [PF1] key is pressed, the letters “DIR” are printed, and the PX-8 will wait for any further input. If you simply press the [RETURN] key, the directory of the current drive will be printed. If you press the space bar and then the two keys “C:” followed by the [RETURN] key, the directory of the C: drive will be printed on the screen.

To change the assigned string, press the number corresponding to the PF key (i.e. 1 for PF1, 2 for PF2 etc, remembering that 0 is used for PF10). The second line of the screen will display the message:

### Terminate function key string with HELP

and the third line of the screen will then display the name of the key whose function is to be changed, with the cursor to the right of it awaiting input. Up to 15 characters can be inserted into each PF string. If a control key is to be added, e.g. a carriage return (CTRL-M), this MUST be added by pressing the [CTRL] and the appropriate alphabetic key. However, in the special case of the carriage return simply pressing the [RETURN] key will add the characters “^M” to denote the CTRL-M for the carriage return. This counts as one character of the string, although it is displayed as two (“^” and “M”) in this section of the CONFIG program. If a control character has been added correctly, the backspace ([BS]) key will remove it as a pair if the [BS] key is pressed once. When a carriage return is inserted correctly it will be seen on the function key display line as a ↵ character. If the characters “^M” appear on the function key display line, they are represented as those characters and DO NOT REPRESENT A CARRIAGE RETURN. For example if the function key display line shows “DIR^M” for one of the keys when the corresponding key is pressed, all five characters will be displayed and the cursor will be placed at the right of the last one awaiting further input.

Although the string can consist of up to 15 characters, if more than seven are assigned to the string, only the first seven will be seen on the function key display line. However, if the function key is pressed the complete number of characters will be printed to the screen.

If a wrong character is input, the backspace key ([BS]) can be used to erase the previous character.

Because the [RETURN] key can be used to enter the carriage return as a “^M”, the [HELP] key is used to terminate the string. The [ESC] key is used to return to the CONFIG main menu.

### 3. SETTING THE CURSOR AND THE FUNCTION KEY DISPLAY

If key 3 is pressed from the CONFIG main menu, the display changes to:

```
*** CURSOR & FUNCTION KEY DISPLAY ***
  Select alphanumeric or ESC to return.

cursor tracking   : on           1=on           2=off
cursor display   : on           3=on           4=off
cursor type      : block,blink  5=block,blink  6=block,nonblink
function key display: off       7=underline,blink 8=underline,nonblink
                                          9=on           A=off
```

The left of the screen shows the current status of the various parameters which can be altered by this section of the CONFIG program. The right part of the screen shows which keys will change them. The parameter setting is altered simply by pressing the key. For example if the 9 key is pressed, the left hand side of the screen will change to show the function key display to be 'on' and whenever the CP/M command line is displayed, the assignments to the function keys will be shown on the base of the screen. The display can be switched off using the A key on this option of the CONFIG program.

This option also allows the type of cursor displayed to be changed, by typing any of the keys 5 to 8. These keys do not affect the cursor on the command line of the MENU and System Display.

Keys 3 and 4 switch the cursor on and off. This affects both the cursor on the CP/M command line and the cursor on the MENU command line.

Keys 1 and 2 switch the tracking mode on and off. The tracking mode means that the cursor follows the window as the real screen moves over the virtual screen. The non-tracking mode means that the window is locked on a particular part of the virtual screen and the cursor (where input is made, or the PX-8 prints the next character) will move along the virtual screen, and once it moves outside the window will not be visible. The tracking and non-tracking modes can also be set using the **SCRN** key. Setting the non tracking mode with the CONFIG program will set the window at the top of the screen. The **SCRN** key can be used to set the window anywhere on the virtual screen.

Futher details of using the tracking and non tracking modes are given in section 2.2.6.

#### NOTE:

*In the screen mode 3, the type of cursor is always set to "underline, nonblink" and cannot be changed with the CONFIG program.*

### 4. SETTING THE DATE AND TIME

Setting the date and time can be achieved by taking option 4 of the CONFIG main menu. The time can be set more accurately than in the initialization question when using either the sub-CPU or initialization reset. The time is entered the moment the **RETURN** key is pressed. When the date is input calculation of the day of the week is carried out automatically.

When option 4 is taken from the CONFIG main menu, the display changes to:

```
*** DATE & TIME ***
  Set date and time or ESC to return.
  Date as MM/DD/YY ?

Date      : 00/00/00 (SUN)
Time     : 18:13:19
```

Although the PX-8 is waiting for the date to be input, simply pressing the **RETURN** key will switch to asking for the time to be input. As soon as the time option is selected, the time is no longer updated on the sixth line. Change the time by inputting the hours, minutes and seconds separated by a colon. If any of the data is a single digit number, it does not have to be preceded by a zero; however, data must be input for the hours minutes and seconds or an error will be detected and the input line will be cleared. When the correct time as required has been entered press the **RETURN** key to enter it into the memory of the computer. The time will be updated when the **RETURN** key is pressed. If an error is made the **BS** key can be used to erase the characters to the left of the cursor.

When the **RETURN** key is pressed after setting the time, or directly on selecting option 4 from the CONFIG main menu, the date can be input as month and day followed by the last two digits of the year, each item being separated by a backslash "/" character. The date will then be entered when the **RETURN** key is pressed. The input will then be entered into the memory of the PX-8 to update the date. If an error is made, such as trying to enter a day of the month greater then the number of days in that month, the input line clears and awaits a correct input.

When a date has been entered on the screen the display changes to input a time. Neither the time or date is updated at this stage. To see if the correct date has been entered, press **RETURN** once more to change the input line to date input.

The date will be updated on the screen when the time is first updated as the seconds change.

Pressing the **[ESC]** key at any time returns to the CONFIG main menu.

## 5. ALLOCATION OF DISK DRIVES

Selecting option 5 on the CONFIG main menu will cause the screen to change to:

```
*** DISK DRIVES ***      Select number or ESC to return.
disk drives : 1          1= A: RAM disk   2= A: FDD1       3= A: FDD1
                        B: ROM1       B: RAM disk   B: FDD2
                        C: ROM2       C: ROM1       C: RAM disk
                        D: FDD1       D: ROM2       D: ROM1
                        E: FDD2       E: FDD2       E: ROM2
                        F: FDD3       F: FDD3       F: FDD3
                        G: FDD4       G: FDD4       G: FDD4
```

This menu makes it possible to allocate the logical disk drives to the physical disk drives in one of three ways. The three tables on the right show the ways in which the allocations are made. The default drives are set as in table 1. The number to the right of the first colon on line two shows which table is selected. Pressing the 1, 2 or 3 keys will change the selected table, and the selected set will then be shown on line 2. The ESC key is used to return to the CONFIG main menu, and keys other than 1,2 3 and ESC are ignored.

It is only possible to reallocate the drives in these three ways. The Microcassette drive is always drive H: and so is not shown on this menu. The terms FDD1, FDD2, FDD3 and FDD4 refer to additional Floppy Disk Drives which are connected to the PX-8 via the serial interface. The ROM drives are shown in section 4.1.2.

## 6. PRINTER INTERFACING

It is possible to drive a serial printer from either of the interface ports. Selecting option 6 on the CONFIG main menu changes the display to:

```
*** PRINTER ***
Select number or ESC to return.

printer I/F      :RS-232C      1=RS-232C      2=serial
```

The fourth line shows which option is selected. The 1 and 2 keys can be used to change the selection. Details of using a printer with the PX-8 are given in Chapter 4.

As with the other sub-menus, after checking which interface is selected, or changing the selection, the **[ESC]** key is used to return to the CONFIG main menu.

## 7. RAM DISK

This option can be used to change the RAM disk size if part of the memory is being used for file storage.

If option 7 is taken from the CONFIG main menu the screen display changes to:

```
*** RAM DISK ***
Set RAM disk size or ESC to return.
RAM disk size ?

RAM disk size      : 9 kb      Max. RAM disk size is 24 kb.
```

The current RAM disk size is shown on the fifth line, and can be changed by typing in the new size. If the size is decreased any files already on the RAM disk will be destroyed. If the RAM disk size is increased the files will remain. The value of the RAM disk size is entered into the memory of the PX-8 by pressing **[RETURN]**. If the value is greater than allowed, the input line will be cleared. If the value input is less than the current value the fourth line will display the message:

**RAM disk files will be destroyed (Y/N) ?**

followed by a flashing cursor. If the Y key is pressed, the new size RAM disk will be set up, destroying the files currently saved there. If the N key is pressed, the fourth line will clear and place the cursor next to the RAM size input message.



The maximum size the RAM disk can be extended to is shown on the right of the fifth line. The above display shows the maximum possible (24 kb) which occurs when no RAM is reserved for USER BIOS.

If a USER BIOS area has been reserved, the value of the maximum value will reflect this. The total USER BIOS and RAM disk must not exceed 24 kb.

If an Intelligent RAM disk is connected it is not possible to use part of the memory of the PX-8 to extend the size of this RAM disk. For example, if the RAM Disk Unit 120 is connected, when option 7 is chosen from the CONFIG main menu, the screen changes to show:

```
*** RAM DISK ***
An external RAM disk is connected. ESC to return.

RAM disk size      : 120 kb
```

The ESC key can then be used to return to the CONFIG main menu. Further details of the Intelligent RAM disk can be found in Chapter 4.



**WARNING:**

*Do not switch off the PX-8 (either manually or by allowing the auto power off to be activated) after changing the RAM disk size without exiting from the CONFIG program. The full configuration can only be carried out by CP/M. If the PX-8 is switched off, the RAM disk could be destroyed and it may also be necessary to re-initialise the system.*

## 8. CHANGING THE RS-232C INTERFACE PARAMETERS

Option 8 on the CONFIG main menu allows the settings of the RS-232C interface to be set.

The RS-232C interface is used to transmit data to and from the PX-8. For example if a text file has been written on the PX-8 and the data needs to be used on a desk top computer such as the EPSON QX-10, the file can be sent to the other computer using the TERM or FILINK program in the CP/M UTILITY ROM. This transmission can be by cable directly to the other computer, or over a telephone line using an acoustic coupler.

The RS-232C interface or the serial interface can be used to connect to a printer.

The printer must have a serial interface in order to receive the information. This allows program listings as well as letters/reports or other text data from applications programs to be printed for a permanent record.

In both these cases the parameters of the interface have to be matched. This option on the menu allows the parameters to be set. In general they should be set or checked with this option before using the communications program (e.g. TERM or FILINK). Further details of the RS-232C interface are given in Chapter 4.

When option 8 is chosen from the CONFIG main menu the screen changes to display:

```
*** RS-232C ***
Select alphanumeric or ESC to return.

bit rate : 4800      1=19200    2=9600    3=4800    4=2400    5=1200    6=600
              7=300      8=150     9=110     A=75/1200 B=1200/75 (Tx/Rx)
data bits : 8
parity  : none      E=none     F=odd      G=even
stop bits : 2       H=1       I=2
```

The bit rate (bits per second or baud rate) currently set is shown on the fourth line. It can be changed to various settings using keys 1 to 9 and keys A and B. The settings produced by keys 1 to 9 set transmission rates in both directions (transmit and receive). The A and B keys set a rate which is different for transmitting and receiving. The A key sets 75 baud transmit (Tx) and 1200 receive (Rx) with the reverse for key B.

The number of data bits is shown on the sixth line and can be changed using the C and D keys.

The parity is shown on the seventh line and can be changed using the E, F and G keys.

The number of stop bits is shown on the eighth line and can be changed using the H and I keys.

When the parameters have been set or checked, the ESC key returns the PX-8 to the main menu.

## 9. Screen Mode

This option can be used to change the screen configuration. Details of the types



of screen and its use are given in Chapter 2 (section 2.2.7). There is also a detailed detailed practical guide to using the different screen modes in the BASIC Reference Manual (Chapter 2 section 2.14).

When option 9 is chosen from the main CONFIG menu, the screen changes to:

```
*** SCREEN ***
  Set screen configuration or ESC to return.
  Screen mode (0,1,2,3) ?
screen mode      : 0
virtual screen 1 : 80 x 25
virtual screen 2 : 80 x 23
selected screen  : 1
separation character:
```

The cursor is located next to the prompt on the third line of the screen. The rest of the screen shows the current configuration of the screen.

Since different screen modes show and require different configurations they will be treated individually as follows:

#### Screen Mode 0

Pressing the 0 key followed by **RETURN**, will cause the third line of the screen to change to:

##### Number of lines of virtual screen 1 ?

You can then enter the number of lines of the screen you require for this screen in the range 8 to 40. If you attempt to enter value outside this range, the line will clear after the prompt, and you can then enter a correct value. When the **RETURN** key is pressed the prompt line alters to read:

##### Number of lines of virtual screen 2 ?

Again the number of lines of the screen is in the range 8 to 40, but the total of the number of lines on the two virtual screens MUST be the less than 48. Also each screen must have a minimum of eight lines, so the maximum lines on a screen is 40 if the minimum is used on either screen. When entering a value for the second screen if the total of the two screens would be greater than 48 with the value you have entered for the second screen, the prompt line will clear so that you can enter a correct value. When **RETURN** is pressed and a valid entry has been made, the prompt line will display:

##### Select virtual screen (1,2) ?

You can then choose which screen will be displayed.

If you wish to change only one of the parameters, simply pressing **RETURN** will enter the current value. Thus if you only wanted to change the virtual screen, the **RETURN** key would be pressed until the required prompt string was displayed.

When the selected screen is entered, the display changes to show the values selected.

Since screen mode 0 can only display one virtual screen at a time, the separation character parameter will not be shown.

#### Screen mode 1

In screen mode 1, both virtual screens have the same number of lines. Thus in answer to the prompt:

##### Number of lines of virtual screen 1 ?

you are in fact choosing the number of lines on both virtual screens. Since also each the virtual screen when displayed is split into two columns of 39 characters with a separation character, a minimum of 16 lines is required on each virtual screens. Also the maximum number of lines on each screen is 48. If a value outside the range 16 to 48 to entered the line will clear from the prompt in order that a correct value can be inserted.

When the **RETURN** key has been pressed and a correct value has been entered, the prompt line changes to:

##### Select virtual screen (1,2) ?

One of the two virtual screens can then be chosen to be displayed as split on either side of the separator. Since the separator is fixed, the screen changes to show the changed values.

#### Screen mode 2

In screen mode 2 both virtual screens are displayed at once. It is also possible

to alter the number of columns of the display of each screen. The number of lines on each screen must be the same.

When the number of lines in the virtual screens have been chosen correctly in the range 8 to 48, the prompt line changes to show:

#### Number of columns of virtual screen 1 ?

It is thus possible to alter the width of the two halves of the display. The number of columns must be in the range 1 to 79. When the width of the left hand side of the screen is chosen the width of the right hand side is set to a value of 79 minus the size of the left hand side.

When the screen width has been set the prompt line changes to:

#### Select virtual screen (1,2) ?

and it is possible to set which screen the cursor lies in. When this has been chosen, the final input required is the separator character. If the **RETURN** key is pressed, the default character is used; this is the same character as is used as separator for the screen mode 1.

On pressing **RETURN** when the separator character has been chosen, the screen shows all the parameters chosen, and it is then possible to use the **ESC** key to return to the CONFIG main menu.

#### Screen mode 3

There is only one setting for the parameters of screen mode 3. The screen has only one virtual screen of 8 lines and 80 columns. Thus entering screen mode 3 and pressing **RETURN** causes the prompt line to revert to the same prompt line, awaiting either the screen to be changed again or the **ESC** key to be pressed to return to the main CONFIG menu.

### 10. THE SERIAL INTERFACE

The Serial interface is used for communication with other devices (in particular printers and floppy disk drives) but not for general communication in the way that the RS-232C interface is. If an EPSON disk drive is connected, the transmission rate is set by the PX-8. If a printer is connected, the transmission rate can be adjusted using CONFIG command.

Selecting option A from the CONFIG main menu shows the following display:

```
*** SERIAL ***
Select number or ESC to return.

bit rate (printer) : 4800          1=4800    2=600    3=150
bit rate (FDD)    : 38400
```

If a printer is being used check that the bit rate setting is the same otherwise unintelligible data will be printed on the printer. Three settings are provided, and they can be changed using the 1, 2 and 3 keys. If the printer you are using does not have a serial port which can be set to any of these values, you will only be able to print using the RS-232C interface of the PX-8, as this has the standard settings from 19200 to 110 baud. Other protocols are set to 8-bit, non-parity, 1 stop-bit.

The Floppy Disk Drive bit rate is shown on the sixth line for reference.

When the parameters have been set or checked, the ESC key returns the PX-8 to the main menu.

### 11. USER BIOS SETTING

The USER BIOS is used by some applications programs. It can also be used by advanced programmers. If you are such a user, please refer to the OS Reference Manual.

An applications program will normally change the USER BIOS size automatically. The manual which describes the program may tell you to alter the USER BIOS manually in which case follow the instructions given.

Normally the setting of the USER BIOS will be 0 pages. If you find it is not zero it will probably have been set by an application program. For example Portable Scheduler™ uses the USER BIOS area to store the alarm times. If you change the size of the USER BIOS area in this case, the values set up will no longer be actioned.



#### WARNING:

*Do not switch off the PX-8 (either manually or by allowing the auto power off to be activated) after changing the USER BIOS size without exiting from the CONFIG program. The full configuration can only be carried out by CP/M. If the PX-8 is switched off, the RAM disk could be destroyed and it may also be necessary to re-initialize the system.*

## 12. CHANGING THE CHARACTER SET BY COUNTRY

The DIP switch is normally used to set the keyboard layout. Occasionally, it is useful to be able to temporarily change the characters but not the keyboard layout so that, for example, a word processed file written in French can be read on the screen with the correct characters. Taking the last option by pressing the key marked C, will enable the character set for the different countries to be displayed. The menu shows:

```

*** COUNTRY ***
  Select number or ESC to return.

country          : ASCII          1=ASCII          6=Italy
                                     2=Denmark         7=Norway
                                     3=England        8=Spain
                                     4=France         9=Sweden
                                     5=Germany
  
```

The county whose characters are displayed is changed by pressing the key corresponding to the country in the table on the right of the screen. The currently chosen country is displayed on the fourth row on the left hand half of the screen.

International Character Sets

Country Dec. Code	United States	France	Germany	England	Denmark	Sweden	Italy	Spain	Norway
35	#	#	#	£	#	#	#	₣	#
36	\$	\$	\$	£	¤	¤	₣	₣	¤
64	@	@	§	@	£	£	@	@	£
91	[	°	ä	[	£	ä	°	ı	£
92	\	¢	ö	\	¢	ö	\	¢	¢
93	]	§	ü	]	¤	¤	£	ı	¤
94	^	^	^	^	ü	ü	^	^	ü
96	'	'	'	'	£	£	ü	'	£
123	<	£	ä	<	¤	ä	ü	ı	¤
124	ı	ü	ö	ı	¢	ö	ö	₣	¢
125	>	£	ü	>	ä	ä	£	>	ä
126	~	ı	ß	~	ü	ü	ı	~	ü

## 3.9 Communications

The PX-8 can be used to communicate with other computers, either directly or using telephone lines. This means that program and data files can be sent from one computer to another directly along a simple pair of wires - the cables and telephone lines connecting the computers are really no more than this. Connecting a printer to a computer is a very simple form of communication, as is the addition of disk drives, so it is not really much more complicated to connect another computer. The length of the pair of wires between the computers is irrelevant so long as the correct equipment is used. For instance, if you want to connect your PX-8 at home to your QX-10 desktop microcomputer at the office, all you would need is an acoustic coupler at each end into which to plug the telephone handset and the appropriate sending and receiving software to run on each computer. It wouldn't matter if one computer was in New York and the other in London - the system would still work.

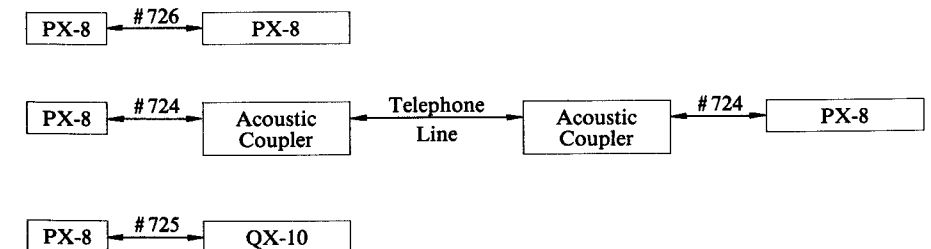
The examples given here relate mainly to communication with the EPSON QX-10 desktop microcomputer, the EPSON acoustic coupler and EPSON printers, although any other computers and printers may be used given the appropriate cables and utility software.

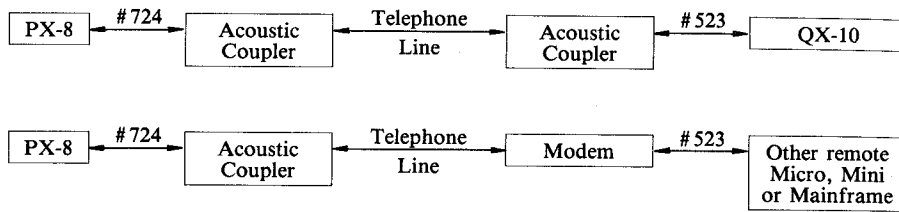
### HARDWARE

When the PX-8 is required to 'talk' to other hardware, the following cables are necessary to connect via the RS232C I/O ports:

	Cable Type
PX-8 to PX-8	# 726
QX-10	# 725
Acoustic Coupler	# 724

There are various configurations possible of which the following are a few:





For an explanation of RS232C communication see Chapter 4.

## SOFTWARE

The two utility programs available on the PX-8 are TERM and FILINK. Various other communications programs are available which work in roughly the same way, and you should consult their respective manuals for instructions as to their use.

### 3.9.1. TERM

The TERM utility is used when it is necessary to 'talk' to a remote computer, using the PX-8 as a terminal, sending single files in either direction or simply using both as 'dumb' messengers. In the following descriptions the second computer is assumed to be an EPSON QX-10 (MultiFont CP/M version) for the sake of clarity, but equivalent software may be used on other machines for this type of communication.

The first thing to do is make sure the PX-8 is connected to the required external devices using the relevant cables, then load the TERM program from the appropriate drive.

When TERM is run the screen shows the following:

```
The RS-232C status is :
bit rate = 4800  data bits = 8  stop bits = 2  parity = NONE
Use CONFIG.COM program to change the RS-232C status.
Modes of TERM
1 = Normal
2 = Delete LF after CR (send)/ Insert LF after CR (receive)
3 = Insert ETX and Delete LF after CR (send)
Select a mode 1
```

The top line of the screen shows the default RS232C settings (they may not be the same as the ones shown here). These can be changed using the CONFIG program and should be set before running the TERM program. Make sure the receiving computer has the same settings, and change one or other machine's settings so that they agree.

The three options given on the screen refer to how data is sent and received.

- i) The default is option 1 which merely requires you to press the key, and sets the computer to send and receive data without any modification. This is the most commonly used setting.
- ii) Some computers automatically add a line feed to any carriage return in the data they receive, so in this case you would select Option 2 (delete LF). This will have the effect of deleting line feed codes (ASCII code 10 decimal; hexadecimal 0A) following carriage returns (hex 0D) during transmission, that is, when sending data. It also adds a line feed code immediately after any carriage return code in received data. Remember that a carriage return merely moves the cursor to the beginning of a line - it is the line feed code that moves it to the next line down.

iii) Option 3 (add ETX and delete LF) has the same effect on line feeds as Option 2, but it also adds an 'end of text' code (ASCII code 03 decimal; hexadecimal 03) to each transmitted carriage return after deleting LF code there. When receiving, data is processed as is. This allows communication with computers which require the ETX code instead of the LF code.

When you have selected which mode you wish to use on the PX-8 by pressing **RETURN** key, the screen will display:

```

Modes of TERM
1 = Normal
2 = Delete LF after CR (send)/ Insert LF after CR (receive)
3 = Insert ETX and Delete LF after CR (send)
Select a mode 1

display/      print/      send/      receive/      /exit
  
```

The bottom line of the screen shows the functions assigned to the **PF** keys. When these functions are inactive they appear in lower case characters, and when activated by pressing the relevant key they appear in upper case. These have the following effect:

**PF1 - DISPLAY ON/OFF**

This switches the display ON or OFF while transmission is in progress. Normally, transmitted data does not show on the PX-8's screen, so switching the display on allows you to see what is transmitted as it is sent. If the display is switched on while data is being received (either from the keyboard or from an external source) the data will show twice on the screen. The effect is that anything sent from an outside source instead of appearing on the screen as 'Hello' will appear as 'HHeellloo'. This would also be the case if the display were switched on while using the PX-8 as a terminal.

**PF2 - PRINTER ON/OFF**

Printer output can be switched on or off during reception or transmission to produce a hard copy of the data to be transferred. Thus, whatever appears on the screen will also appear on the printer. Remember that because the TERM program uses the RS232C port the printer must be connected to the serial port. The **PF3** and **PF4** assignments at the bottom of the screen disappear when PRINTER is ON.

**PF3 - SEND A FILE/TERMINATE TRANSMISSION**

A file can be sent to the external device and its transmission ended using this key. This procedure is fully described in 1.4 below.

**PF4 - RECEIVE A FILE/TERMINATE RECEPTION**

Pressing the **PF4** key will make the PX-8 prompt for the name of the file to be received. A full description of this can be found in 1.3 below.

**PF10 - EXIT**

Pressing **PF10** and will cause the PX-8 to exit from the TERM program at any time.

**HELP - OUTPUT BREAK SIGNAL**

Pressing the **HELP** key outputs the BREAK signal to the receiving computer.

*Note:*

*When using TERM only the function displayed on the 8th line can be selected.*

**1.1 SENDING AND RECEIVING MESSAGES**

When transmitting messages between computers it is necessary to have a program to handle the data being transmitted. Since there are many such programs and computers, the EPSON QX-10 is used as an example with the QX-10 TERM program as the executing software.

First of all, execute the TERM programs on both the PX-8 and the QX-10. The QX-10 screen will show:

```

QX-10 TERMinal or Remote ver 1.8

rs232c : bit rate = 300 parity = no stop bit = 1 data char = 8bit

Terminal mode..... 1
Remote mode..... 2
Normal mode..... 3
change rs232c..... X

select 1-3 or X?
  
```

Comparison of the RS232C parameters with the default parameters of TERM on the PX-8 will show that the handshaking does not match. The quickest way to make an alteration is to change the QX-10 parameters by selecting 'X'. When the screen changes to that of fig. below press 'B' and '5' to change the bit rate to 4800 and the stop bits to 2:



### QX-10 TERMinal or Remote ver 1.8

rs232c : bit rate = 300 parity = no stop bit = 1 data char = 8bit  
select A-9 or RETURN ?

bit rate	parity	stop bit	data char
9600 ..... A	no ..... 0	1 ..... 3	0-5bit ..... 6
4800 ..... B	yes even ..... 1	1.5 ..... 4	6bit ..... 7
2400 ..... C	yes odd ..... 2	2 ..... 5	7bit ..... 8
1800 ..... D			8bit ..... 9
1200 ..... E			
900 ..... F			
600 ..... G			
400 ..... H			
300 ..... I			
200 ..... J			
150 ..... K			
135 ..... L			
110 ..... M			
75 ..... N			
50 ..... O			

The parameters given on the PX-8 screen are the default settings, as are those for the QX-10. Either computer's settings can be changed using the CONFIG command, but only the QX-10's can be changed from within the TERM program. However, the QX-10 parameters are only temporarily affected by the TERM program and the QX-10 CONFIG program must be used to alter permanently the QX-10 default parameters. If it is necessary to change the parameters on the PX-8 press **RETURN** then **PF10** (shifted **PF5**) to exit from TERM. Now you can use CONFIG to alter the parameters.

Now select Option 1 (default) on the PX-8 and Option 1 on the QX-10. The QX-10's screen will clear and the message

### QX-10 Terminal Mode

will appear at the top with the flashing cursor beneath it. The PX-8's cursor will move to the beginning of the line above the function key assignment display. Typing a message on the PX-8 will result in the message appearing on the screen of the QX-10. At the end of a sentence or paragraph the **RETURN** key can be pressed which returns the cursor to the beginning of the same line and a line feed can be sent by pressing the **↓** key. Exactly the same thing will happen in reverse if you type in a message from the keyboard of the QX-10. The characters will appear on the screen of the PX-8 - pressing the **RETURN** key will move the cursor back to the beginning of the current line and pressing the **↓** key will move the cursor down a line. Pressing **PF1** on the PX-8 will turn the display on and result in characters input from the PX-8's keyboard showing on both screens instead of just the QX-10 screen. Remember that both computers are completely 'dumb' in this mode and cannot perform any processing. All they do is display on their screens whatever has been typed in on the keyboard of the other computer.

If the printer is switched on using the **PF2** key all output from the QX-10 keyboard will be echoed to the printer.



#### WARNING

*If the printer is not switched on, off line or wrongly connected, the function key display of 'PRINT' will flash.*

*To exit from this mode press **PF10** (shifted **PF5**) on the PX-8 and the **BREAK** key on the QX-10.*



## 1.2 THE PX-8 AS A TERMINAL

This is a very useful function which can be used, among other things, for remote data processing. One instance would be if the QX-10 were in the office and you were in another part of the country. You could leave the QX-10 set up in Remote Mode, plug your PX-8 into an acoustic coupler and thence into the public telephone system, then use the QX-10 with the PX-8 behaving as the keyboard and screen. In this way you could enter data, retrieve it and process it from anywhere you happen to be, at any time!

First of all, make sure the RS232C settings match (see previous Section). then run the TERM program on both computers. On the QX-10 choose Option 2 to enter the Remote Mode. Wait a few seconds and the A> prompt will appear on the screen of the PX-8. This means that the PX-8 is now acting as a 'dumb' terminal to the host QX-10 computer. You can now use the PX-8 exactly as if it were really the QX-10 and perform file manipulation, input and output, programming or any other functions just as though you were using the QX-10's own keyboard and screen. The only constraints are that the screens are a different size (the PX-8 screen behaves like a window on the QX-10 screen) and the keys do not behave in quite the same way - the **PF** key of the PX-8 retain their assignments as part of the TERM program, but the other keys behave as you would expect on the QX-10. In fact, the first draft of this paragraph was typed on a PX-8 acting as a terminal to a QX-10 which was running a powerful word processing program!

If the printer is switched on with the **PF2** key, all output from the host computer will be printed as hard copy as though you had pressed **CTRL - P**.

In order to terminate this host-terminal condition the following operations on the PX-8 must be carried out:

- a) Return to the A> system prompt given by the QX-10;
- b) Make sure the disk containing the TERM.COM program is on the A: drive of the QX-10;
- c) Type TERM to re-enter the Terminal program;
- d) Select Option 3 on the menu - the QX-10 will now return to normal mode;
- e) Press **PF10** to exit from the TERM program on the PX-8.

## 1.3 RECEIVING FILES

Program and data files can be downloaded from an external device to the PX-8 by using the RECEIVE facility. Enter the TERM program in the normal manner, then press **PF4** to select RECEIVE. You will then be prompted for a filename. This should be the filename you wish the file to have on the PX-8, for instance, to send it to the RAM disk call it A:filename.extension or to send it to the microcassette drive call it H:filename.extension. The computer will then wait for input from the RS232C port and direct the file to wherever you have specified.

After all the file was received, press **PF4** key so that the received file may be closed.

It is possible to print the file being received on the printer connected to the serial port, but not at the same time as it is being transferred. If PRINTER ON is selected the printing will not begin until the file has been fully transferred to the PX-8.

If you are using the QX-10 or another CP/M based computer for output, an easy way of demonstrating this is to use the PIP command, for instance

```
A>PIP PUN: = A:FRED.BAS
```

This command tells the QX-10 to send the file A:FRED.BAS to the device called PUN: which is the default output device name for the RS-232C port. The **RETURN** key must not be pressed until the PX-8 is ready and waiting for input. If you have told it to expect a file called H:SALLY.BAS it will receive the file A:FRED.BAS sent by the QX-10 and store it on microcassette tape as H:SALLY.BAS. The contents of the file will appear on the screen of the PX-8 as it is transmitted, and transmission can be terminated by pressing the HELP key. This outputs a Break signal to the transmitting device and tells it to stop sending.

## 1.4 SENDING FILES

The PX-8 can send files to an external device by working the opposite way round to the procedure for receiving a file.

Enter the TERM program in the normal way, then press **PF3** to enter the transmission mode. First the display of assignments for keys **PF4** and **PF2** are suppressed, then the prompt

### Enter file name

appears. Type in the complete filename of the file you wish to send, e.g. H:CHARLIE.BAS.

Next comes a series of three questions relating to time delays to be inserted between blocks of data, as follows:

```
Enter file name      A:DOCUMENT.TXT
Set transmission delay time in 10 ms (Max. 255)
After each character  0
After CR.LF          0
After 128 bytes      0
```

The defaults for these delays are 0 milliseconds, i.e. no delay, and this is the most commonly used setting. Consult the manual for the receiving device and software before deciding if these defaults should be altered. If you find characters are lost, you can try empirically adding a delay, but in normal circumstances you should not need to change the default values. At the receiving end you can use the same method as described in 1.3 above except that this time the RS232C port should be prepared for input. The command

**A> PIP B:KATE.BAS = RDR:**

will read in the PX-8 file H:CHARLIE.BAS through the RS232C port of the other computer and write it to disk drive B: on that computer as B:KATE.BAS.

Another possibility is that you may wish to look at a large screenful of data. To do this you can use the command

**A> PIP CRT: = RDR:**

which will direct the input to the screen.

When data transfer is completed the TERM program on the PX-8 can be exited in the normal way by pressing the **PF10** key.

## Error messages

All errors (except 'Printer not ready', 'File not found' and 'File already exists') which require action will necessitate the TERM program being restarted after the error has been dealt with.

- a) **PRINT** ( **PF2** key display) **blinks** - if you have selected the printer option make sure that:
  - i) It is connected to the serial port;
  - ii) The printer is switched on and on-line.
- b) **RS-232C is not ready** - make sure that RS-232C cable is connected correctly.
- c) **File not found** - the file you have specified for transmission cannot be found. You are given the opportunity to specify the file again - there was probably a typing mistake the first time.
- d) **No file name specified** - specify the file name
- e) **Bad File descriptor** - specify the correct drive and file name.
- f) **Drive select error** - you have specified the drive name other than A: to H:.
- g) **Overwrite (Y/N)?** - This means that the file you have specified already exists but gives you the chance to either overwrite it (by answering Y to the question) or to specify a different filename (by answering N).

- h) **Communication error** - if the RS232C ports of the two computers are not configured with the same parameters the PX-8 will not receive recognisable characters. When data is received from the transmitting computer the screen will show this error message and the prompt:

**Press ESC to restart, STOP to exit from TERM**

You can then take the appropriate action.

### **NOTE:**

*The QX-10 TERM program will cause meaningless characters to appear on the screen if an attempt is made to send to it with non-matching RS232C parameters.*

- i) **Directory full** - this means that the directory on the device to which you want the file written is full, that is, contains the maximum number of entries. You can delete a file, change the disk or tape or specify a different device, and then start TERM again.
- j) **Disk full** - this lets you know that the device to which you are writing the file has no more space on it. In the case of a disk or a tape change it for another one with more room on it. In the case of the RAM disk you will either have to redirect the file to another device or risk losing what is already on it by expanding it with CONFIG. In any case you will have to start the TERM program again.

## 3.9.2 FILINK

FILINK is a more specialised program than TERM in that it supports specific protocols between machines and is used solely for sending and receiving files. It can be used to communicate with the following machines using the specified software:

PX-8 FILINK.COM  
Portable WordStar™ using the 'T' and 'C' commands

QX-10 FILINK.COM

File names can be specified in their entirety or using wildcard characters (\* and ?), unlike TERM which will only support full filenames.

Refer to the OS Reference Manual for details of the communications protocol.

The cables used for communication are the same as those specified in Section 3.9.1 for the TERM command.

### Using FILINK

When the program is run the following screen appears:

```
A file transfer program via RS-232C port.

The RS-232C status is :
bit rate = 4800  data bits = 8  stop bits = 2  parity = NONE
Use CONFIG.COM program to change the RS-232C status.

Press ESC to restart, STOP to exit from FILINK or CTRL/STOP to abort.
Send or Receive (S/R) ?
```

The second line of the screen gives the RS232C settings. These can be changed using the CONFIG program. The fourth and fifth lines give information on special key usage:

**ESC** Pressing the **ESC** key while the program is waiting for key input restarts the program from the beginning.

**STOP** Pressing **STOP** while the program is waiting for key input stops the program and returns control to CP/M.

**CTRL** - Pressing **CTRL** and **STOP** together at any time results in termination of the program and return of control to CP/M.

The sixth line describes the wildcard characters allowed when specifying file names, and the seventh line asks whether you wish to send or receive. When you have typed in 'S' or 'R' you will be prompted for a filename. This can be typed in as a complete name or using wildcard characters (\* and ?). When receiving files you will have to specify a full filename because in this case you are actually specifying the file into which the data being received is to be put. When sending files, a whole family of files can be sent using the wildcard options, for instance

**D:\*.BAS**

will ensure that all files on drive D: having the extension BAS are sent down the line.

### Error messages

- a) **Directory full** - this means that the directory on the device to which you want the file written is full, that is, contains the maximum number of entries. You can delete a file, change the disk or tape or specify a different device, and then start FILINK again.
- b) **Disk full** - this lets you know that the device to which you are writing the file has no more space on it. In the case of a disk or a tape change it for another one with more room on it. In the case of the RAM disk you will either have to redirect the file to another device or risk losing what is already on it by expanding it with CONFIG. In any case you will have to start the FILINK program again.
- c) **Overwrite (Y/N)?** - This means that the file you have specified already exists but gives you the chance to either overwrite it (by answering Y to the question) or to specify a different filename (by answering N).
- d) **File not found** - the file you have specified for transmission cannot be found. You are given the opportunity to specify the file again - there was probably a typing mistake the first time.
- e) **Receiver is not ready** - make sure you have connected the PX-8 correctly to the external device and that the receiving computer is ready to receive the file.
- f) **No file name specified** - Specify the file name.

- g) **Bad file descriptor** - Specify the correct file name.
- h) **Drive select error** - Specify the drive name from A: to H:.
- i) **Communication error** - if the RS-232C ports of the two computers are not configured with the same parameters the PX-8 will not receive recognisable characters. When data is received from the transmitting computer the screen will show this error message and the prompt:

**Press ESC to restart, STOP to exit from FILINK**

You can then take the appropriate action.

**NOTE:**

*The QX-10 TERM program will cause meaningless characters to appear on the screen if an attempt is made to send to it with non-matching RS-232C parameters.*

- j) **Close error** - replace the new disk and restart from the beginning.
- k) **Sender is not ready** - make sure that the sending computer is ready to send the file.
- l) **RS-232C is not ready** - make sure that RS-232C cable is connected correctly.

## 3.10 Other CP/M Transient Programs

Apart from the utility programs or commands in CP/M which have already been described, there are some others which will be available to you as soon as you connect a disk drive to your PX-8. Some are disk utility programs and some allow you to carry out all sorts of operations on the computer which are either difficult or impossible with the machine on its own. Please consult your dealer for information on obtaining these programs.

### 1. Disk utilities

#### a) **FORMAT** (Preparing a floppy disk for use)

This program allows you to **FORMAT** a floppy disk in a specified drive. **FORMATting** is a process whereby the disk is divided up into sections which help the computer to find its way around. It also has the effect of erasing all the data already on the disk. This function may be included in **COPYDISK**.

#### b) **COPYDISK** (Making an exact copy of a floppy disk)

Instead of using **PIP** you can use **COPYDISK** to copy the entire contents of one disk on to another. This is particularly useful when making back-up copies of program and data disks. This program incorporates a **VERIFY** facility so that it compares what is on the disk from which you are copying with what is being written to the disk on which the copy is being made, thus ensuring a perfect copy.

### 2. Program utilities

#### a) **ED** (Editing a file)

This is an **EDitor** incorporating a powerful set of subcommands which can be used for on-screen creation and editing of files. It is most useful when creating **SUBMIT** files or when typing in or correcting assembler programs.

#### b) **DDT** (Dynamic Debugging Tool)

Programs written in Intel 8080 in assembler or machine code can be quickly debugged or altered with **DDT**. It contains a number of subcommands which can be used to look at, alter and assemble programs in memory. Files changed in this way can be written back to disk using the CP/M **SAVE** command.

c) **ASM** (8080 Assembler)

When an assembler program has been written and given the filename extension .ASM, running this program will assemble it and produce object and list files with the filename extension .HEX and .PRN which are automatically written to disk.

d) **LOAD** (Producing COM files)

When an assembler program has been assembled using the ASM program, the object code file (type .HEX) can be converted into an executable machine code file with the filename extension .COM. You can use this to create your own commands on disk.

e) **DUMP** (Hexadecimal file dump)

Any file on a disk can be shown on the screen in hexadecimal form using this program. Each line contains a 16-byte section of the file with the starting address given at the left hand end.

### 3.11 Finding Out More About CP/M

This manual cannot cover all aspects of using CP/M particularly the finer points. The following books are among many available from your computer dealer or bookshop.

**CP/M and the Personal Computer** by Thos A. Dwyer and Margot Critchfield. This covers CP/M from the beginners point of view as well as containing much information on the internal workings of CP/M and advanced uses. (Published by Addison Wesley. ISBN 0-201-10355-9)

**CP/M Handbook** by Rodney Zaks. This is a general guide to using CP/M. (Published by Sybex. ISBN 089588-048-2)

**Osborne CP/M User Guide** by Thom Hogan. Another general guide. (Published by Osborne/McGraw Hill)

**CP/M BIBLE** - The Authorative Reference Guide to CP/M by Mitchell Waite and John Anglermeyer. As the title suggests, a reference guide to the use of CP/M. (Published by Howard Sams. ISBN 0-672-22015-6)

**Soul of CP/M** by Mitchell Waite and Robert Lafore. This is a guide for the advanced user and programmer. (Published by Howard Sams. ISBN 0-672-22030-X)

**CP/M The Software Bus - A Programmers Companion** by A.Clarke, J.M.Eaton and D.Powys-Lybbe. A further manual for advanced programmers. (Published by Sigma Technical Press. ISBN 0905104-18-8)