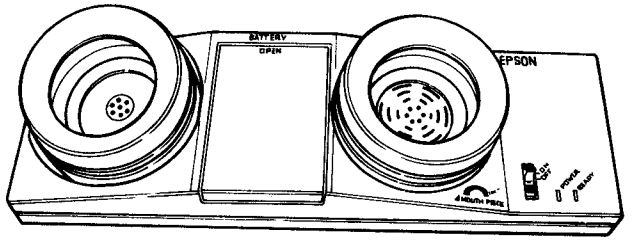
4.10 Acoustic Coupler

With an acoustic coupler, you can connect the PX-8 to a computer (such as another PX-8, Epson QX-10 or a large computer) in another place using the telephone line. Data is transfered through the RS-232C interface.



Acoustic Coupler

The TERM and FILINK utility programs described in Chapter 3, enable communications to be handled with a minimum of effort. More information is given under these programs.
Such communication can also be achieved in BASIC.

Cable #724 is required to attach to an acoustic coupler with a DB25 connector.

The acoustic coupler models available differ from country to country. See your Epson dealer for detailed information.

Chapter 5

THE SYSTEM INTERFACE

The operating system of the PX-8 is an expanded version of the CP/M operating system, version 2.2. This chapter deals with the more advanced aspects of using the PX-8 and will be of more use to programmers than any other users. It contains information on subroutines which can handle peripheral devices and files. Familiarity with assembler programming is a prerequisite for using these subroutines and functions. You must purchase an assembler such as MACRO-80 and other program development tools from your dealer for using these BDOS and BIOS subroutines.

Users who are not interested in assembler programming may skip this chapter.

Further information on the operating system will be published as a separate manual.

5.1 The CP/M Configuration

The configuration of the expanded CP/M is as follows:

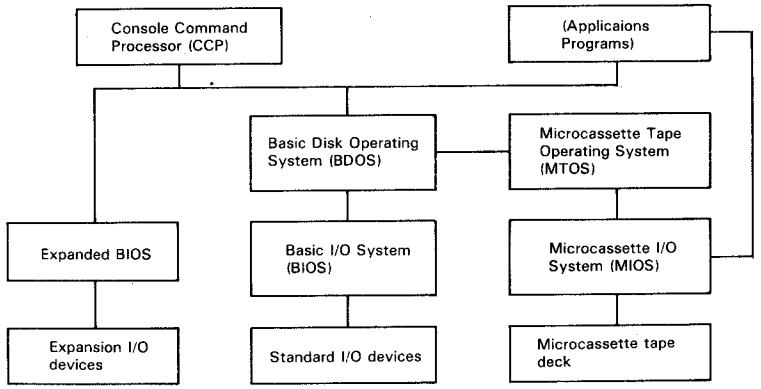


Fig. 5-1

ClibPDF - www.fastio.com

5.2 The IOBYTE

The PX-8 operating system supports 4 logical I/O devices, CON:, RDR:, PUN: and LST: as described in Chapter 3. In order to make it easy for the user, the CP/M Utilities STAT enable devices to be assigned using these names.

In software terms the logical and physical devices are assigned using the address 0003H in the main memory bank which contains a byte whose contents decide the logical to physical device assignments. This byte is known as the IOBYTE. The contents of the IOBYTE assigns the logical to physical devices as shown in the following table.

Table 5.1

Logical device	LST:	PUN:	RDR:	CON:	
Bit position	7 or 6	5 or 4	3 or 2	1 or 0	
Direction / Bit pair value (Binary)	Output	Output	Input	Output	Input
00	Serial (printer)	—	Keyboard	RS-232C	Keyboard
01	LCD	LCD	—	*LCD	*Keyboard
10	*RS-232C	*RS-232C	*RS-232C	LCD	RS-232C
11	—	—	—	RS-232C	RS-232C

The default setting of the IOBYTE is 10101001. (Hexadecimal A9) This gives the default assignments as shown by the asterisks in the table.

Further information on the assignments can be found in Chapter 3, sections 3.6 and 3.8.

5.3 The File Control Block (FCB)

During file operations, the operating system obtains file information from a table called the file control block (FCB). The information included in the FCB is as shown below.

Dr	F1	F2	/ /	F8	T1	T2	T3	Ex	S1	S2	Rc	D0	/ /	D15	Cr	R0	R1	R2
00	01	02	...	08	09	10	11	12	13	14	15	16	...	31	32	33	34	35

Dr: Decimal Drive code
- 0: use logged-in drive
- 1: use Drive A:
- 2: use Drive B:
- 3: use Drive C:
- 4: use Drive D:
- 5: use Drive E:
- 6: use Drive F:
- 7: use Drive G:
- 8: use Drive H: the microcassette drive

F1 — F8: contain the file name in upper case ASCII

T1 — T3: contain the file type in upper case ASCII

Ex: Current extent number

S1: Reserved for system
S2: Reserved for system

Rc: Record count

D0 — D15: Assigned by operating system.

Cr: Record counter

R0 — R2: Random access record counter

Fig. 5.2

The CCP provides a FCB in the system work area which is called a TFCB (temporary FCB); this is used as the default FCB.

ClibPDF - www.fastio.com

5.4 BDOS Function Calls

BDOS includes many utility subroutines which input or output data to and from peripherals and handle files. These subroutines are referred to as BDOS functions, and can easily be used in your own programs as shown below. There are 39 BDOS functions which can be utilised; any of these can be used by calling address 0005H with a function number in register C and a parameter in register pair DE. Single byte values will be returned in register A while double byte values are returned in register pair HL.

The BDOS functions are listed below. In this list, the entry parameters are those which are passed to BDOS from the user program which calls the function, and the return parameters are those which are passed to the user program from the function called.

Table 5.2

Function No.	Explanation	Entry Parameter	Returned value
0:	This function returns control to the CP/M command level.	C: 00H	
1	This function reads a character from CON:	C: 01H	A: Read character
2	This function outputs a character to CON:	C: 02H E: Character to be output	
3	This function reads a character from RDR:	C: 03H	A: Read character
4	This function outputs a character to PUN:	C: 04H E: Character to be output	
5	This function outputs a character to LST:	C: 05H E: Character to be output	
6	This function provides direct console input and output operation. It bypasses all CP/M normal control character functions such as "CTRL" + "P.".	C: 06H E: For input, 0FFH For output, the character to be output.	A: For input, the character read 00H is returned when CON: is not ready.
7	This function returns the current contents of IOBYTE.	C: 07H	A: Contents of IOBYTE
8	This function sets a new value in IOBYTE.	C: 08H E: Value to be set in IOBYTE	
9	This function outputs the character string starting at the specified address and ending with "$" to CON:	C: 09H DE: Starting address of the memory area in which the character string is stored	
10	This function reads a character string from CON: into the buffer area starting at the specified address.	C: 0AH DE: Starting address of the buffer area	Buffer: Character string read from CON:
11	This function reads the status of CON:	C: 0BH	A: CON: status FFH — CON: ready 00H — CON: not ready
12	This function returns the version number of the CP/M system currently in use.	C: 0CH	H: CP/M or MP/M L: Version number
13	This function sets all disks to read/write, selects drive A and sets the default DMA address to 0080H.	C: 0DH	
14	This function selects the specified disk drive.	C: 0EH E: Name of drive to be selected	

(Continued)

ClibPDF - www.fastio.com

Function No.	Explanation	Entry Parameter	Returned value
15	This function opens a file.	C: 0FH DE: FCB address Positions 1 through14must contain the name of the file to be opened.	A: Directory code 0FFH when the file cannot be found. This function fills the FCB in the main memory with the contents of the corresponding FCB in the directory stored on the disk.
16	This function closes a file.	C: 10H DE: FCB address	A: Directory code 0FFH when the file cannot be found. This function writes the contents of the FCB in the main memory to the directory on the disk.
17	This function searches for a file.	C: 11H DE: FCB address	A: Directory code 0FFH when the file cannot be found.
18	This function is used following function 17 to find the file whose name matches that specified.	C: 12H	A: Directory code 0FFH when the file cannot be found.
19	This function deletes the specified file.	C: 13H DE: FCB address	A: Directory code 0FFH when the file cannot be found.
20	This function reads the next record from the file into memory at the current DMA address.	C: 14H DE: FCB address	A: 00H when the read operation is completed. Other than 00H when the next record contains no data.
21	This function writes 128 bytes of data at the current DMA address to a record of the file specified by the FCB.	C: 15H DE: FCB address	A: 00H when the write operation is completed. Other than 00H when the disk is full.
22	This function generates a new file and catalogs it in the directory.	C: 16H DE: FCB address	A: Directory code 0FFH when the directory is full.
23	This function changes the file name.	C: 17H DE: FCB	A: Directory code 0FFH when the file cannot be found
24	This function returns the log-in vector which indicates drives which are currently on line.	C: 18H	HL: Log-in vector The least significant bit of L corresponds to drive A and the most significant bit of H corresponds to drive P. A "1" bit indicates that the corresponding drive is on line. A: Contains the same value as register L.
25	This function returns the currently logged-in drive.	C: 19H	A: Currently logged-in drive 01H - Drive A 02H - Drive B 05H - Drive E 06H - Drive F

(Continued)

Function No.	Explanation	Entry Parameter	Return Parameter
26	This function changes the DMA address.	C: IAH DE: DMA address	
27	This function returns the base address of the allocation vector.	C: IBH	HL: ALLOC address
28	This function sets the read only attribute for the currently logged-in drive.	C: 1CH	
29	This function returns the R/O vector which indicates drives which are set to read only.	C: IDH	HL: R/O vector
30	This function sets the file attributes. The R/O and system attributes can be set or reset with this function.	C: IEH DE: FCB address	A: Directory code A: FFH (no file)
31	This function returns the BIOS resident disk parameter block (DPB).	C: 1FH	HL: DPB address
32	This function sets or gets the user number.	C: 20H E: 0FFH for get E: User code for set	A: User number (GET)
33	This function is similar to function 20; however, a particular record is read according to the contents of positions R0 through R2 in the FCB.	C: 21H DE: FCB address	A: Return code 00H - Normal completion Non-zero - Abnormal completion
34	This function is initiated similarly to function 33. However, the data at the DMA address is written to the disk.	C: 22H DE: FCB address	A: Return code 00H - Normal completion Non-zero - Abnormal completion
35	This function returns the virtual file size to the random record bytes (R0 to R2) of the FCB.	C: 23H DE: FCB address	
36	This function returns the random record position to the random record bytes of the FCB after a series of sequential reads or writes.	C: 24H DE: FCB address	
37	This function resets the specified drives according to the 16-bit drive vector indicating drives to be reset; the least significant bit of the vector corresponds to drive A, and so forth.	C: 25H DE: Drive vector	A: 00H
40	This function is similar to function 34. However, the data written is all 00H.	C: 28H DE: FCB address	A: Return code 00H - Normal completion

ClibPDF - www.fastio.com

5.5 Zero Page Locations

CP/M holds a number of parameters in page zero of the main memory bank. The IOBYTE (section 5.2) is held in location 3H. The following are important locations in page zero. Further information is included in the books referred to in Chapter 3, and other such books.

Location in hex		Contents
from	to	
0000	0002	A jump instruction followed by the warm start entry address. Thus locations 0001 and 0002 contain WBOOT.
0003		The IOBYTE. See section 5.2 for further details.
0004		The current default drive number (0=A:, 1=B: etc.)
0005	0007	A jump instruction followed by the location of BDOS. Thus locations 006 and 007 contain the lowest address used by CP/M.

5.6 The BIOS Interface

The Basic Input/Output System (the BIOS) of the operating system for the PX-8 includes many useful subroutines which can be used by calling their entry addresses from user programs after setting parameters (if required) in applicable registers. Care must be taken to ensure that the entry addresses are correctly specified when calling these subroutines.

ClibPDF - www.fastio.com

5.7 The Entry Address of the BIOS Subroutines

Table 5.3 lists the entry addresses of BIOS subroutines.

Since the BIOS routines are relocatable, the entry addresses of BIOS subroutines are indicated relative to the warm boot routine (WBOOT) address. The WBOOT address can be found from locations 0001 and 0002 in page zero.

Table 5.3

ADDRESS	ENTRY NAME
WBOOT −03H	BOOT
WBOOT	WBOOT
WBOOT +03H	CONST
+06H	CONIN
+09H	CONOUT
+0CH	LIST
+0FH	PUNCH
+12H	READER
+15H	HOME
+18H	SELDSK
+1BH	SETTRK
+1EH	SETSEC
+21H	SETDMA
+24H	READ
+27H	WRITE
+2AH	LISTST
+2DH	SECTRAN
+30H	PSET
+33H	SCRNDUMP
+36H	BEEP
+39H	RSOPEN
+3CH	RSCLOSE
+3FH	RSINST
+42H	RSOUTST

ADDRESS	ENTRY NAME
+45H	RSIN
+48H	RSOUT
+4BH	TIMDAT
+4EH	MEMORY
+51H	RSIOX
+54H	LIGHTPEN
+57H	MASKI
+5AH	LOADX
+5DH	STORX
+60H	LDIRX
+63H	JUMPX
+66H	CALLX
+69H	GETPFK
+6CH	PUTPFK
+6FH	ADCVRT
+72H	SLAVE
+75H	RDVRAM
+78H	MCMTX
+7BH	POWEROFF
+7EH	USERBIOS

5.7.1 Functions of BIOS Subroutines

In the following explanations, entry parameters are those which must be assigned by user programs calling the BIOS routines. The contents of registers other than those to which the system returns parameters may be changed unless otherwise specified.

5.7.2 Subroutines Concerned With Power-Up and Initialization

BOOT

Entry Point: WBOOT − 03H

BOOT is the entry point for the cold start loader, which runs only when system initialization is made or the 7508 sub-CPU is reset or SHIFT + NUM GRAPH keys are pressed together with RESET switch. This routine is not used by the user.

WBOOT

Entry Point: See locations 0001H and 0002H in page zero.

The WBOOT location is used as the point from which all other BIOS routines are given in the following sections. WBOOT can alter if the configuration of the system changes, e.g. the RAM Disk size alters or the USER BIOS size changes.

WBOOT is the entry point for the warm start bootstrap; this routine loads CCP and BDOS into memory. The MENU or CCP module is activated after a warm start.

Return Parameters:
The currently selected drive is returned in register C.

POWEROFF

Entry Point: WBOOT + 7BH

POWEROFF is the entry point for the subroutine which turns off the main system power supply, after saving the current status.

Entry parameters
Register C = 00H Sets continue mode when switching on.
Register C = 01H Sets restart mode upon power-up.

5.7.3 BIOS Subroutines for use with the console

CONST

Entry Point: WBOOT + 03H

ClibPDF - www.fastio.com

CONST is the entry point for the subroutine which reads the console status and sets it in register A.

Return Parameter
Register A = 00H Indicates that the console input buffer is empty.
Register A = FFH Indicates that the console input buffer contains characters.

CONIN

Entry Point: WBOOT + 06H

CONIN is the entry point for the subroutine which enables a character to be input from the keyboard. It loops indefinitely until a character is entered when the buffer is empty. Because it is also possible to press the Programmable Function keys, special provision is made for them as follows.

Return Parameters
Register A contains the ASCII code of the input character if a key other than a PF key is pressed.

The FUNKFLG (address F108H) setting determines the contents of register A when a PF key is pressed.

If FUNKFLG is set to FFH, registers A and C contain the following data:

	Register C	Register A
PF key pressed	FFH	Code as in table of right
OTHER KEY pressed	00H	ASCII code

Code returned by the PF keys			
PF1	E0H	PF6	E5H
PF2	E1H	PF7	E6H
PF3	E2H	PF8	E7H
PF4	E3H	PF9	E8H
PF5	E4H	PF10	E9H

If FUNCFLG is not set to FFH (normally it will be 0), then register A will contain the ASCII code of the key pressed if a key other than a PF key is pressed, and will contain the first character of the PF key string if a PF key is pressed. Continuous polling of the keyboard will cause the other characters to be entered from the PF key string as if they were typed from the keyboard.

CONOUT

Entry Point: WBOOT + 09H

CONOUT is the entry point for the subroutine which outputs a character to the console from register C. Note that a number of functions can be obtained by using the control codes in Appendix E and also the ESC control sequences in Appendix A.

Entry parameter
Register C = ASCII code of character to be output to the console

5.7.4 BIOS subroutines for use with devices

LIST

Entry point: WBOOT + 0CH

LIST is the entry point for the subroutine which outputs a character to the logical device LST:. It will be output if both DSR and TxRDY are "1", otherwise will wait until these conditions are satisfied.

Entry parameter
Register C = ASCII code of character to be output to LST:

PUNCH

Entry Point: WBOOT + 0FH

PUNCH is the entry point for the subroutine which outputs a character to the logical device PUN:.

Entry parameter
Register C = ASCII code of character to be output

READER

Entry Point: WBOOT + 12H

READER is the entry point for the subroutine which reads a character from the logical device RDR:. This subroutine loops until a character is input.

Return Parameter
Register A = ASCII code of character input from RDR:

ClibPDF - www.fastio.com

LISTST

Entry Point: WBOOT + 2AH

LISTST is the entry point for the subroutine which reads the status of the logical device LST:.

Return Parameter
Register A = 00H Indicates that LST: is busy.
Register A = FFH Indicates that LST: is ready.

This subroutine returns FFH when DSR = 1; otherwise, it returns 00H.

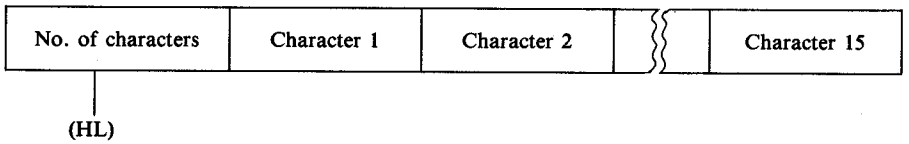
GETPFK

GETPFK is the entry point for the subroutine which accesses the character strings assigned to the PF keys.

Entry Point: WBOOT + 69H

Entry parameters
Register C = PF key number -1 (i.e. 0 to 9 for keys 1 to 10)
Register HL = Starting address of the character string buffer

Result
The format of the character string buffer is as follows.



Whether or not any of the PF keys has been pressed can be determined using the CONST and CONIN routines. The maximum capacity of the character string buffer is 15 characters. The contents of register pair HL are not affected by execution of this routine.

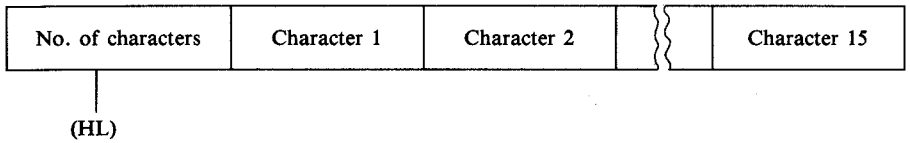
PUTPFK

PUTPFK is the entry point for the subroutine which assigns a character string to the PF keys.

Entry Point: WBOOT + 6CH

Entry parameters
Register C = PF key number − 1 (i.e. 0 to 9 for keys 1 to 10)
Register HL = Starting address of the character string buffer

The format of the character string buffer is as follows.



A maximum of 15 characters can be assigned to one PF key. The contents of register pair HL are not affected by execution of this routine.

PSET

Entry Point: WBOOT + 30H

PSET is the entry point for the subroutine which replaces graphic screen data at a specified address in accordance with a specified logical operation. The graphics screen is sequential in memory, with each dot corresponding to one bit. The graphic screen memory starts at 8380H and has a length of F00H bytes. Each byte thus corresponds to eight dots across the screen. Since there are 480 dots, the second row of the screen starts at byte 83BC. Logical operation is required if individual bytes need to be set.

Entry parameters
Register B = Data
Register C = 1 (AND)
C = 2 (OR)
C = 3 (XOR)
C = Others (NOP)
Register HL = graphic screen address

Return Parameters
Register A = 00H — Normal completion
Register A = FFH — Character screen mode

ClibPDF - www.fastio.com

Register A = other than 00H and FFH — Indicates that the address in HL is not within the VRAM area.
The contents of register B and register pair HL are not changed.

SCRNDUMP

Entry Point: WBOOT + 33H

SCRNDUMP is the entry point for the subroutine which prints a hard copy of the LCD screen image on the printer. It is terminated by pressing the CTRL-STOP key.

Return Parameters
F67EH (LSTERR) = 00H — normal operation
F67EH (LSTERR) = FFH — Terminated by CTRL-STOP

5.7.5 BIOS Subroutines related to the Disk operation

HOME

Entry Point: WBOOT + 15H

HOME is the entry point for the subroutine which sets the track to zero.

SELDSK

Entry Point: WBOOT + 18H

SELDSK is the entry point for the subroutine which specifies a disk drive.

Entry parameter
Register C = 00H Drive A:
Register C = 01H Drive B:
Register C = 02H Drive C:
Register C = 03H Drive D:
Register C = 04H Drive E:
Register C = 05H Drive F:
Register C = 06H Drive G:
Register C = 07H Drive H:

HL register includes the disk parameter header address.

Return Parameters
Register HL = 00H Indicates that a parameter error has occurred.
Register HL is not 00H Normal operation

SETTRK

Entry Point: WBOOT + 1BH

SETTRK is the entry point for the subroutine which selects the track to be read from or written to.

Entry parameter

Register BC = Track number

The track number varies according to the drive as shown below. It can be any number from zero to that shown in the following table:

Drive	Default Track number	Default Device
A:	Ø to 1 (Max. 2) Ø to 6 Ø to 13	Internal RAM disk 60K RAM Disk Unit 120K RAM Disk Unit
B: C:	Ø to 3	ROM 1 (32K ROM) ROM 2 (32K ROM)
D: E: F: G:	Ø to 39	FDD 1 FDD 2 FDD 2 FDD 4
H:	Ø to 4	Microcassette drive

If a value which is not within the above range is specified, an error occurs when a read or write is performed. These values assume that the default settings for the drive assignments hold. See section 3.8.6 for further details of drive assignments.

NOTE:
Maximum track number changes in accordance with the sign of RAM disk set by CONFIG.

SETSEC

Entry Point: WBOOT + 1EH

ClibPDF - www.fastio.com

SETSEC is the entry point for the subroutine which sets the sector number to be read from or written to.

Entry parameter
Register BC = Sector number to be accessed (00H — 3FH)

If a value other than 00H to 3FH is specified, an error will occur when a read or write is performed.

SETDMA

Entry Point: WBOOT + 21H

SETDMA is the entry point for the subroutine which sets the starting address of the 128 byte DMA buffer area used for disk access.

Entry parameter
Register BC = DMA address

READ

Entry Point: WBOOT + 24H

READ is the entry point for the subroutine which reads data from a disk drive into the DMA buffer according to parameters set by the SELDSK, SETTRK, SETSEC and SETDMA subroutines.

Return Parameter
Register A = 0 Normal completion
Register A = Other than 0 Abnormal completion

WRITE

Entry Point: WBOOT + 27H

WRITE is the entry point for the subroutine which writes data to a disk drive according to parameters set by the SELDSK, SETTRK, SETSEC and SETDMA subroutines.

Entry parameter
Register C = 00H Write standard format data.
Register C = 01H Write unblocked data.
Register C = 02H Write sequential file.

Return Parameter
Register A = 00H Normal completion
Register A = Other than 00H Abnormal completion

SECTRAN

Entry Point: WBOOT + 2DH

SECTRAN is the entry point for the subroutine which converts a logical sector number into the corresponding physical sector number.

Entry parameters
Register BC = Logical sector number

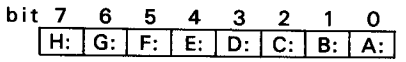
Register HL = Physical sector number

DISKTBL AND DISKROV — CHANGING DRIVES

The CONFIG program enables the logical and physical drives to be assigned in three ways. It is not possible to have the Microcassette Drive assigned to any drive other than drive H:. The logical and physical drives are assigned in a table of 7 bytes in length beginning at address F1D2H (DISKTBL). The table at default contains the following data:

Address	Logical Drive	Code	Physical Drive
F1D2H	A:	00	RAM Disk
F1D3H	B:	01	ROM Capsule 1
F1D4H	C:	02	ROM Capsule 2
F1D5H	D:	03	Floppy Disk Drive 1
F1D6H	E:	04	Floppy Disk Drive 2
F1D7H	F:	05	Floppy Disk Drive 3
F1D8H	G:	06	Floppy Disk Drive 4

If the code 07H to FFH is contained in this table, the drive cannot be selected. If the table is changed, the two bytes at address F1DAH (DISKROV) must also be changed. This is the vector table which sets the READ/WRITE status of the drives. If the corresponding bit is "1" the drive is set to R/O, and if to R/W it is set to "0". F1DAH (DISKROV) includes the drive in the following order.



ClibPDF - www.fastio.com

NOTE:
Please not that device assignments changed by this address may not be reflected in CONFIG main menu 5. Do not set the same value as the code.

5.7.6 Bios Subroutine for the Speaker

BEEP

Entry Point: WBOOT + 36H

BEEP is the entry point for the subroutine which sounds the speaker.

Entry parameters
Register C = 0 turns the speaker off.
C = 1 to FF The speaker sounds for an interval of [C] × 0.1 secs
Register DE sets the frequency of the sound according to the expression:

$$(DE) = \frac{10^6}{3.2 \times \text{Frequency (Hz)}}$$

5.7.7 BIOS Subroutines for the RS232 Port

The following routines can be used to operate the RS-232C interface. Details of the RS-232C interface are given in Chapter 4.

The interface supports all commonly used transmission rates, up to 19,200 bps. In general it is not possible to transmit and receive at different rates. Since some databases (e.g. the European Videotext databases) use 75 bps transmit and 1200 bps receive this option is available together with the reverse 1200 bps transmit/75 bps receive.

The interface is initialized using the RSIOX routine, but can also be initialized by BASIC and partially by the CONFIG program. The operating system uses a 261 byte buffer. Characters overflowing this buffer are lost. The size of the buffer can be increased by the RSIOX routine.

The interface should be opened before attempting to send data.

The interface also supports XON/XOFF and SI/SO in communication. These are described in Chapter 4.

The SI/SO function only applies in 7 bit transmission.

When $\overline{\text{SI/SO}}$ = 1, no action is taken.
When $\overline{\text{SI/SO}}$ = 0, if a SI character (0FH) is received, subsequent characters will have a "0" added as the MSB to complete the byte. When $\overline{\text{SI/SO}}$ = 0, if a SO character (0EH) is received, subsequent characters in the range 20H to 7EH will have a "1" added as the MSB to complete the byte. Characters 00H to 1FH and character 7FH will remain unchanged.

The $\overline{\text{XON/XOFF}}$ protocol allows the receiving device to tell the sender to wait while it catches up on processing.

When $\overline{\text{XON/XOFF}}$ = 1 no action is taken.
When $\overline{\text{XON/XOFF}}$ = 0 a transmission off character CTRL-S (13H) is sent when the buffer becomes more than three-quarters full, so that processing can catch up. A transmission on character CTRL-Q (011H) is sent when the buffer becomes less than one-quarter full, so that transmission can begin again. This makes it possible to prevent the buffer overflowing and the loss of characters.

RSOPEN

Entry Point: WBOOT + 39H

RSOPEN is the entry point for the subroutine which initializes the RS232C interface for communication according to the conditions set with the CONFIG program. This subroutine turns on the power supply to the RS-232C interface (and the serial interface) and performs processing to prevent noise from being output over the line. When this routine is called, the former RS-232C conditions are reset; therefore, any data remaining in the receive buffer at the time of the call is lost.

RSCLOSE

Entry Point: WBOOT + 3CH

RSCLOSE is the entry point for the subroutine which disables communication through the RS232C interface.

RSINST

Entry Point: WBOOT + 3FH

ClibPDF - www.fastio.com

RSINST is the entry point for the subroutine which checks whether any characters have been received.

Return Parameters
Register A = 00H No data received
Register A = FFH Data received

RSOUTST

Entry Point: WBOOT + 42H

RSOUTST is the entry point for the subroutine which checks whether transmission is enabled.

Return Parameters
Register A = 00H Transmission disabled
Register A = FFH Transmission enabled

RSIN

Entry Point: WBOOT + 45H

RSIN is the entry point for the subroutine which receives one character. When the receive buffer is empty, it waits until data is received.

Return Parameter
Register A = Character received
Z Flag = 01H Normal completion
Z Flag = 00H CTRL-STOP was used to terminate transmission.

RSOUT

Entry Point: WBOOT + 48H

RSOUT is the entry point for the subroutine which outputs a character via the RS232C interface.

Entry parameter
Register C = ASCII code of character to be sent.
Z Flag = 01H Normal completion
Z Flag = 00H [CTRL] - [STOP] was used to terminate transmission.

Buffer address and size are as specified by the user with the RSOPEN routine.

NOTE:
RSOPEN must be called to open the RS232C port before RSIN or RSOUT is called.

RSIOX

Entry Point: WBOOT + 51H

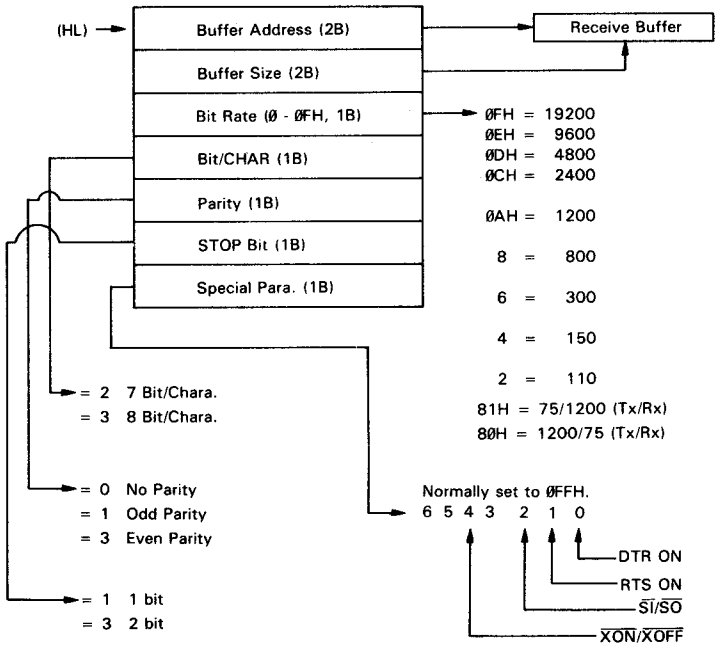
RSIOX is the entry point for the subroutine which determines the I/O configuration via the RS-232C interface and opens or closes the interface. The functions of this subroutine are explained below.

RSIOX (OPEN)

Function: Opens the RS232C interface

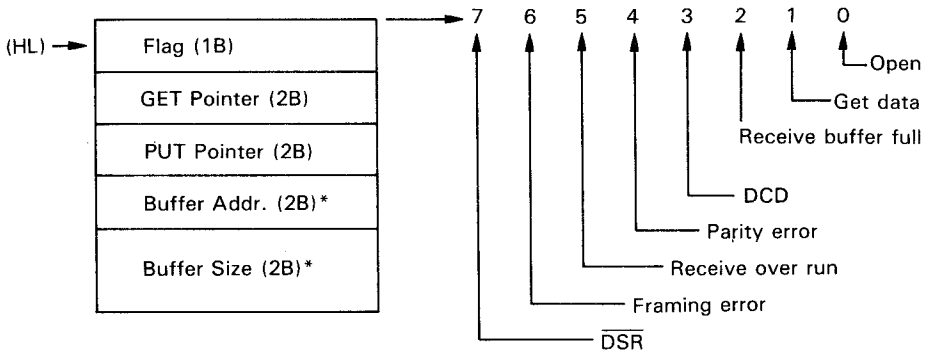
Entry parameters
Register B = 10H
Register HL= Parameter block address

The contents of the parameter block are shown below.



ClibPDF - www.fastio.com

Return parameters
Register A = 00H Normal open — Z flag = 1
A = 02H Busy (i.e., used by another program) — Z flag = 0
A = 04H Parameter error — Z flag = 0
Register HL contents are not changed



RSIOX (CLOSE)

Function: Closes the RS232C interface.

Entry parameter
Register B = 20H

RSIOX (INSTS)

Function: Checks whether there is any data in the receive buffer.
Entry parameters
Register B = 30H
Register HL= 9 byte block address which is used to store return information.

Return parameters
Z flag = 01H Normal completion
Register A = 00H No data in the receive buffer
A = FFH Data has been received
Register BC = LOC — Number of bytes of received data
Register HL = Return information (See Open)

NOTE:
The meaning of LOC relates to the value of PUT(Pointer) and GET(Pointer) (see diagram under RSIOX (OPEN)).

If PUTP ≧ GETP then LOC = PUTP − GETP
If PUTP < GETP then LOC = PUTP − GETP + Buffer size

Z flag = 0 Abnormal end
Register A = 03H — Interface not open
Register BC Unknown
Register HL Unknown

RSIOX (OUTST)

Function: Checks whether output is enabled.

Entry parameters
Register B =40H
Register HL =Address of the block which is used to store reurn information

Return parameters
Z flag = 1 Normal completion
Register A = 00H Output disabled
Register A = FFH Output enabled

Z flag = 0 Abnormal end
Register A = 03H Interface not open
Register HL= Not changed (For the HL contents, refer to the OPEN function.)

RSIOX (GET)

Function: Reads in one byte of data from the receive buffer.

Entry parameters
Register B = 50H
Register HL= Starting address of the block which is used to store return information

Return parameters
Z flag = 1 Normal completion
Register A = Receive data
Register HL= Refer to the OPEN function.

Z flag = 0 Abnormal end
Register A = 03H Interface not open
A = 00H CTRL-STOP was pressed
Register HL= not changed

ClibPDF - www.fastio.com

RSIOX (PUT)

Function: Sends one byte of data.

Entry parameters
Register B = 60H
Register C = Send data
Register HL= Address of the block which is used to store return information

Return parameters
Z flag = 1 Normal completion
Register HL= Refer to the OPEN function.

Z flag = 0 Abnormal end
Register A = 03H Interface not open
A = 00H CTRL-STOP was pressed
The contents of register HL is not changed.

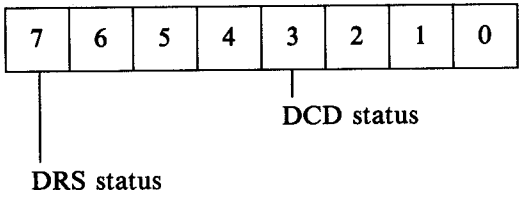
RSIOX (CTLIN)

Function: Reads the status of the control line.

Entry parameter
Register B = 70H

Return parameters
Z flag = 1 Normal completion

Register A shows the conditions of the control line flags as follows:



Z flag = 00H Abnormal end
Register A = 03H Interface not open

RSIOX (SETCTL)

Function: Sets the control lines.

Entry parameters
Register B = 80H
Register C = Data to be set
BIT 0 = DTR (H for "1" and L for "0")
BIT 1 = RTS (H for "1" and L for "0")

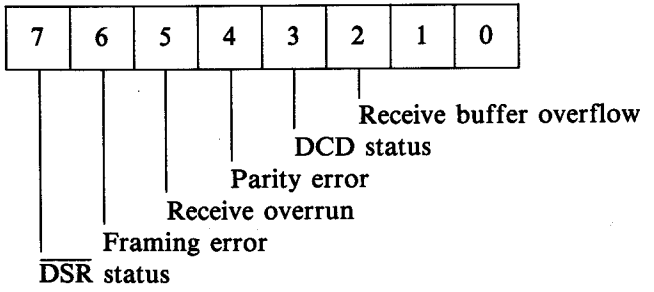
Return parameters
Z flag = 1 Normal completion
Z flag = 0 Abnormal end
Register A = 3 Interface not open

RSIOX (ERSTS)

Function: Checks the error status and clears the error flags.

Entry parameter
Register B = 90H

Return parameters
Z flag = 1 Normal completion
Register A contains the conditions of the control line flags



Z flag = 00H Abnormal end
Register A = 03H Interface not open

RSIOX (SENS)

Function: Checks the status of the RS232C interface.

ClibPDF - www.fastio.com

Entry parameter
Register B = F0H

Return parameters
Z flag = 1 The RS232C interface can be opened.
Z flag = 0 Busy (i.e., the port is being used by another program.)
Register A = 02H

5.7.8 BIOS Subroutines related to the Clock

TIMDAT

Entry Point: WBOOT + 4BH

TIMDAT is the entry point for the subroutine which has 6 functions relating to the clock (time and date), according to the contents of the C register. The TIMDAT routine also uses a series of bytes called the Time Descriptor to read or write the description of the time. There are a total of 11 bytes in the Time Descriptor, although not always all of them may be used by or for each function. The date and time are coded in BCD. The order of the bytes are as follows.

Year	— 1 byte, 2 BCD digits
Month	— 1 byte, 2 BCD digits
Day	— 1 byte, 2 BCD digits
Hour	— 1 byte, 2 BCD digits
Minute	— 1 byte, 2 BCD digits
Second	— 1 byte, 2 BCD digits
Day of week	— 1 byte (Ø = Sunday, 6 = Saturday)
Type	— 1 byte (Ø = none, 1 = alarm, 2 = wake, 3 = wake subroutine).
Address	— 2 bytes (message address for alarm if type = 1, wake string address if type = 2 or subroutine address if type = 3).
Alarm	— 1 byte (Ø = alarm not sounded, 1 = alarm sounded).

It is important to make sure that the number of bytes of the Time Descriptor required by the function are in fact present. It is also important that the order described above is adhered to.

TIMDAT (READ TIME)

Function: Reads the time and sets data in time descriptor address.

Entry parameters
Register C = 00H — Time read function
Register DE = Starting address for 7 bytes of Time Descriptor information

Return Parameters
The contents of register pair DE are not changed.
The Year Month Day Weekday and time are placed in memory starting at the address placed in the DE register.

TIMDAT (SET TIME)

Function: Allows the date, weekday and time to be set. If any bytes of the Time Descriptor are set to FFH they will not be updated. It is also up to the programmer to check the contents of the Time Descriptor before calling this routine as no checks are made.

Entry Parameters
Register C = FFH — set time
Register DE = starting address of the 7 bytes of the Time Descriptor

TIMDAT (ALARM ENABLE)

Function: Enables the alarm/wake function.

Entry parameter
Register C = 80H — Alarm enable

TIMDAT (ALARM DISABLE)

Function: Disables the alarm/wake function.

Entry parameter
Register C = 81H — Alarm disable

TIMDAT (SET ALARM)

Function: Sets the alarm/wake time, whether it is an alarm or a wake function and the address of the location of the message/wake string. The time can be set in increments of 10 seconds. If the byte 0FH is specified, applicable digits will be regarded as wild-card positions. For example, setting FH for the year, month, and day will cause the alarm/wake function to operate at the specified hour, minute, and second every day.

ClibPDF - www.fastio.com

Entry parameters
Register C = 82H — Alarm set
Register DE = Time descriptor address 10 bytes including message string address.

Return Parameters
The contents of register pair DE are not changed.

TIMDAT (READ ALARM)

Function: Reads the Time Descriptor into the address required. Note that the year byte (byte 1) will contain FFH and the seconds byte (byte 6) contains FH in the lowest nibble (lowest 4 bits). This is because neither the year nor the 1s place of the seconds (i.e. it can only be set in 10 second intervals) is relevant for alarm setting.
Entry parameters
Register C = 84H
Register DE = Starting address to load the 11 bytes of the Time Descriptor

5.7.9 BIOS Subroutines related to the Serial Port

The Serial Port is not supported by the BIOS.

5.7.10 BIOS Subroutines related to memory

The memory of the PX-8 consists of 64K bytes of RAM and 32K of ROM. By means of bank switching it is possible to read addresses 0000H to 7FFFH from either RAM or ROM.

In the following routines when references are made to system bank and user bank they correspond to addresses as follows:

System bank: 0000H to 7FFF (ROM)
8000H to FFFF (RAM)

User bank: 0000H to FFFF (RAM)

LOADX

LOADX is the entry point for the subroutine which reads 1 byte of data from the specified memory bank.

Entry address: WBOOT + 5AH

Entry parameters
Register C = −1 — System bank
C = 0 — User bank
Register HL= Data address

Return Parameter
Register A = Data read

This routine does not change the contents of the other registers.

STORX (Ordinarily, this entry is not used by the user.)

STORX is the entry point for the subroutine which writes 1 byte of data into the user bank.

Entry Point: WBOOT + 5DH

Entry parameters
Register C = 0 — User bank
Register A = Data
Register HL= Data address into which data is to be written.

Return Parameters
This routine will not change the contents of the other registers.

LDIRX (Ordinarily, this entry is not used by the user.)

LDIRX is the entry point for the subroutine which transfers data from the specified memory bank to the user bank.

Entry Point: WBOOT + 60H

Entry parameters
Register HL= Starting address of the memory area from which data is transferred

ClibPDF - www.fastio.com

Register DE = Starting address of the memory area to which data is transferred
Register BC = Number of bytes of data to be transferred
Register A = 0 — Data is transferred to the user bank.

The operating system ROM cannot be changed or modified using this routine.

JUMPX

JUMPX is the entry point for the subroutine which jumps to the specified address in the specified memory bank.

Entry Point: WBOOT + 63H

Entry parameters
Register IX = Jump address
If Address F539H (DISBNK) = −1 System bank
If Address F539H (DISBNK) = 0 User bank

NOTE:
When a stack is to be used at the destination address, a new stack must be established.

CALLX

CALLX is the entry point for the subroutine which calls the specified address in the specified bank.

Entry Point: WBOOT + 66H

Entry parameters
Register IX = Call address
If Address F539H (DISBNK) = −1 — System bank
If Address F539H (DISBNK) = 0 — User bank

NOTE:
CALLX uses one level in the user stack. Therefore, the stack pointer must be used in the common area in RAM (8000H to 0FFFFH), and there must be at least one free level of stack space.

SLAVE

SLAVE is the entry point for the subroutine which sends commands and data to the 6301 slave CPU and returns results and data from the slave. The use of this command is beyond the scope of this manual. Full details are shown in the OS Reference Manual and Technical Reference Manual.

RDVRAM

RDVRAM is the entry point for the subroutine which reads the contents of the virtual character screen.

Entry Point: WBOOT + 75H

Entry parameters
Register B = Column at which the read is to start (1 to 80)
Register C = Line at which the read is to start (1 to virtual screen line size)
Register DE = Number of characters to be read
Register HL = Starting address of the area in which data read is to be stored.

Return Parameters
Register A = 00H — Normal completion
A = 01H — Read error
A = FFH — Parameter error or graphic screen read attempted

If the end of the screen is encountered before the specified number of characters has been read, remaining addresses in the storage area are padded with spaces and 1 is returned in register A.

5.7.11 Miscellaneous Subroutines

MEMORY

Entry Point: WBOOT + 4EH

This is not supported. It will simply RETURN.

ClibPDF - www.fastio.com

LIGHTPEN

Entry Point: WBOOT + 54H

This is not supported. It will simply RETURN.

MASKI

MASKI is the entry point for the subroutine which sets and resets interrupt masks.

Entry Point: WBOOT + 57H

Entry parameters
Register B = 00H Interrupt disable
B = 01H Interrupt enable
B ≧2 Interrupt enable register (IER) read
Register C Bit 0 — 7508
C Bit 1 — 8251
C Bit 2 — DCD (RS-232C carrier detect)
C Bit 3 — ICF (Input capture flag)
C Bit 4 — OVF (Timer overflow)
C Bit 5 — EXT (Interrupt from system bus)

Return Parameters
Register A = Contents of IER register (I/O port 4)

ADCVRT

ADCVRT is the entry point for the subroutine which reads the A/D converter input, Bar Code reader, DIP switch settings, battery voltage, or power switch status.

Entry Point: WBOOT + 6F

Entry parameters
RegisterC=00H — A/D channel 1 (input from the analog jack)
RegisterC=01H — A/D channel 2 (input from the bar code reader connector)
RegisterC=02H — DIP SW1
RegisterC=03H — Battery voltage
RegisterC=04H — Power switch status (TRIG status of the analog connector)

When any other value is set in register C, the routine returns without doing anything.

Return Parameters
When Register C contains 00H or 01H on calling ADCVRT
Register A contains A/D converter data in the 6 MSB as follows:

Bit	7	6	5	4	3	2	1	0
Data	MSB					LSB		

Bits 2 to 7 will all be 1 if the input voltage is greater than 2.0 V. If the input voltage is negative, bits 2 to 7 will all be 0.

When Register C contains 2 on calling ADCVRT
Register A contains contains the Switch information as follows.

Bit	7	6	5	4	3	2	1	0
SW	8	7	6	5	4	3	2	1

When Register C contains 3 on calling ADCVRT
Register A contains data from the A/D converter

Full scale=5.7 V

When register C contains 4 on calling ADCVRT

Register A, bit 0=1 — Power switch ON
Register A, bit 0=0 — Power switch OFF
Register A, bit 1=1 — TRIG ON
Register A, bit 1=0 — TRIG OFF

ClibPDF - www.fastio.com

5.7.12 Subroutines for using the MICROCASSETTE DRIVE

MCMTX

MCMTX is the entry point for the subroutine which calls the MTOS functions.

Entry point: WBOOT + 78H

The use of this routine is similar to using a BDOS function call. Full details are given in the OS reference manual.

5.7.13 Subroutines for the USER BIOS

The BIOS can be extended to include routines which are not supported by CP/M.

USERBIOS

Entry Point: WBOOT + 7EH

USERBIOS is the routine which allows USER BIOS entry points to be changed. The table of entry points is reinitialized if a reset is carried out or the system is initialized. Thus it is necessary to reload the USER BIOS entry points if any type of reset is carried out.

The use of the USER BIOS is explained in the OS Reference Manual.

Appendix A

PX-8 CONSOLE ESCAPE SEQUENCES

The PX-8 uses a separate CPU to handle the console. Rather than have many calls to specific routines for particular console functions, the functions are executed by sending a sequence of characters to the console. This appendix deals with the use of these command sequences. It is also possible to use certain ASCII control codes. These are described in Appendix E.

The sequences involve the ESCAPE character ASCII code 27 decimal (1B in hexadecimal), followed by one or more characters, the values of which determine the command to be carried out. In the remainder of this appendix the ESCAPE character is denoted by the letters "ESC". The ESC code would normally be used as part of a machine code routine, by using the CONOUT routine described in section 5.7.3, although they could also be input directly from the keyboard on the CP/M command line in many cases. Since it may be necessary to enter them from the keyboard, the key sequence is given where this is possible. The character sequence can also be executed as part of a BASIC PRINT statement. The BASIC MANUAL contains further information on using the sequences in BASIC programs. Not all the commands are supported in BASIC because they interact with the screen editor.

ClibPDF - www.fastio.com