Appendix A

HARDWARE SPECIFICATIONS

CPU and Memory

Main CPU:	Z-80 compatible CMOS CPU Clock - 3.68 MHz RAM - 64KB ROM - 32KB ROM capsule - 64KB (Max.)
Slave CPU:	7508 (4-bit CMOS CPU) Clock - 270 kHz Internal ROM - 4KB

Power Supply

Main power supply:	Four AA-size dry cells (optional) (not chargeable) Ni-Cd battery pack Capacity - 450 mAH Nominal voltage - 4.8 V Charging current - 70 mA Charging time - Approx. 8 hours (when power is off) AC adapter
Backup power supply:	Ni-Cd battery pack Capacity - 90 mAH Nominal voltage - 4.8 V Charging current - 1 mA

Keyboard

Standard keyboard	
Total number of keys	72
Mode indicators	3 LEDs
Item keyboard	
Total number of keys	58
Mode indicators	3 LEDs

Item keyboard
  Total number of keys 58
  Mode indicators 3 LEDs

LCD (Liquid Crystal Display)

Character mode: 40 characters × 8 lines
Graphic mode: 240 × 64 dot matrix
Character matrix: 6 × 8 dots

Buzzer

Piezoelectric buzzer
  Volume control impossible
  External buzzer connector pin assignment:

Pin No.	Symbol	I/O	Name
1	GND	—	Signal ground
2	EX	Out	Signal output
3	EX	Out	Signal output

ROM Capsule

Number of slots: 2
Capacity: 32KB/slot (Max.)
Type of ROM usable: PROM
  27C64
  27C256
MASK-ROM
  61364
  613128
  613256

Serial Interface

Signal level: RS-232C level (±5 V)
Bit rate: 38400, 19200, 9600, 4800, 2400, 1200, 600, 300, 200, 150, 110 bps
1200/75 or 75/1200 bps for Tx/Rx
Format: 1 start bit plus 8 data bits followed by 1 or 2 stop bits
Parity: Even, odd, none
Mode: Full duplex

External connector pin assignment:
See "3.4 Serial Interface."

RS-232C Interface

Signal level: RS-232C level (±5 V)
Bit rate: 38400, 19200, 9600, 4800, 2400, 1200, 600, 300, 200, 150, 110 bps
1200/75 or 75/1200 bps for Tx/Rx
Format: 1 start bit plus 8 data bits followed by 1 or 2 stop bits
Parity: Even, odd, none
Mode: Full duplex
External connector pin assignment:
See "3.5 RS-232C Interface."

External Audio Cassette

Bit rate: Approx. 1300 bps
Remote control feature
External connector pin assignment:
See "3.6 Printer Interface."

Printer Interface

Conforming to Centronics Standard
External connector pin assignment:
See "3.6 Printer Interface."

Bar Code Reader Interface

External connector pin assignment:

Pin No.	Symbol	I/O	Name
1	GND	—	Signal ground
2	BRDT	In	Data read
3	+5	—	+5 V

Cartridge Interface

This interface is provided for optional cartridge devices such as RAM and ROM cartridges, microcassette drive, A/D converter, cartridge printer. This interface is equipped with both serial and parallel I/O ports. See "3.8 Cartridge Interface" for external connector pin assignment.

System Bus

Address bus: 16 bit parallel
Data bus: 8 bit parallel
Connector pin assignment: See next page.

Pin No.	Symbol	I/O	Signal name	Signal level
1	VCH	P/S	Battery charger line	6 to 8V
2	VB1	P/S	Battery power	4.8 to 6.5V
3	VB2	P/S	Battery power	4.8 to 6.5V
4	VBK	P/S	Battery power	4.8 to 6.5V
5	GND	—	Signal ground	—
6	VL	P/S	Power for logics	4.5 to 6V
7	DB7		Data 7	
≀	≀	TTL	≀	TTL
14	DB0		Data 0	
15	$\overline{\text{MEN}}$	In	Internal memory enable	TTL
16	$\overline{\text{INTE}}$	In	External interrupt	TTL
17	$\overline{\text{WAIT}}$	In	Z-80 wait	TTL
18	$\overline{\text{PON}}$	Out	Power ON	TTL
19	$\overline{\text{CLK}}$	Out	Z-80 main clock	TTL
20	BUAK	Out	Bus acknowledge	TTL
21	$\overline{\text{RD}}$	Out	Z-80 data read	TTL
22	$\overline{\text{IORQ}}$	Out	Z-80 data request	TTL
23	$\overline{\text{WR}}$	Out	Z-80 data write	TTL
24	$\overline{\text{MRQ}}$	Out	Z-80 memory request	TTL
25	$\overline{\text{HLTA}}$	Out	Halt acknowledge	TTL
26	BURQ	In	Z-80 bus request	TTL
27	CG	—	Protective ground	—
28	CG	—	Protective ground	—
29	$\overline{\text{RS}}$	Out	System reset	TTL
30	$\overline{\text{M1}}$	Out	Z-80 machine cycle 1	TTL
31	AB0		Address 0	
≀	≀	Out	≀	TTL
46	AB15		Address 15	
47	OFF	Out	Gate array control at power switching	TTL
48	DW	Out	Expansion D-RAM RF control signal	TTL
49	DCAS	Out	Expansion D-RAM RF control signal	TTL
50	$\overline{\text{RSI}}$	In	Reset input	TTL

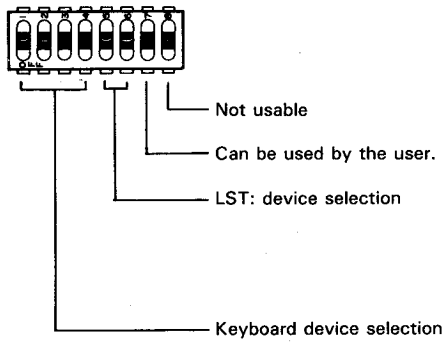
System Bus Pin Assignment

AC Adapter

Input voltage: 100 VAC ±10%
Frequency: 49 to 61 Hz
Output voltage: 6 V (400 mA)
External connector pin assignment:

Pin No.	Symbol	I/O	Name	Level
1	Vch	In	AC adapter output	6 to 8V
2				
3	GND	—	Ground	

DIP switch (in ROM capsule compartment)



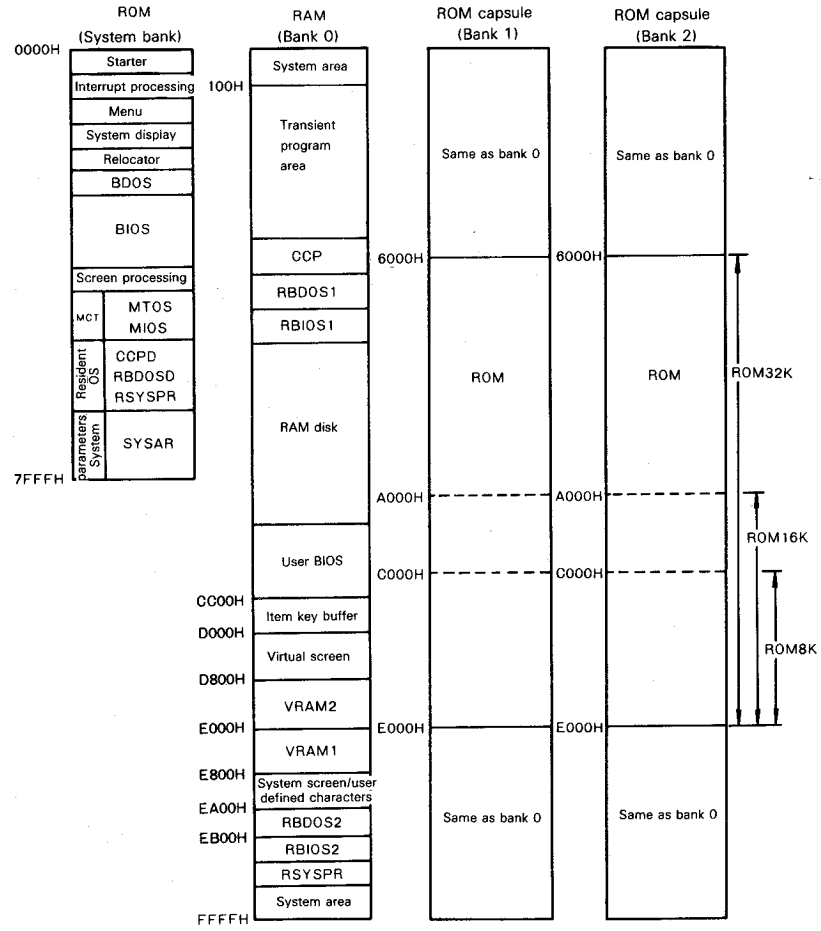
Device name	SW-5	SW-6
Serial interface	OFF	OFF
RS-232C interface	OFF	ON
Cartridge printer	ON	OFF
Printer interface	ON	ON

Keyboard type	SW-1	SW-2	SW-3	SW-4
ASCII	ON	ON	ON	ON
France	OFF	ON	ON	ON
German	ON	OFF	ON	ON
England	OFF	OFF	ON	ON
Denmark	ON	ON	OFF	ON
Sweden	OFF	ON	OFF	ON
Italy	ON	OFF	OFF	ON
Spain	OFF	OFF	OFF	ON
Norway	OFF	ON	ON	OFF

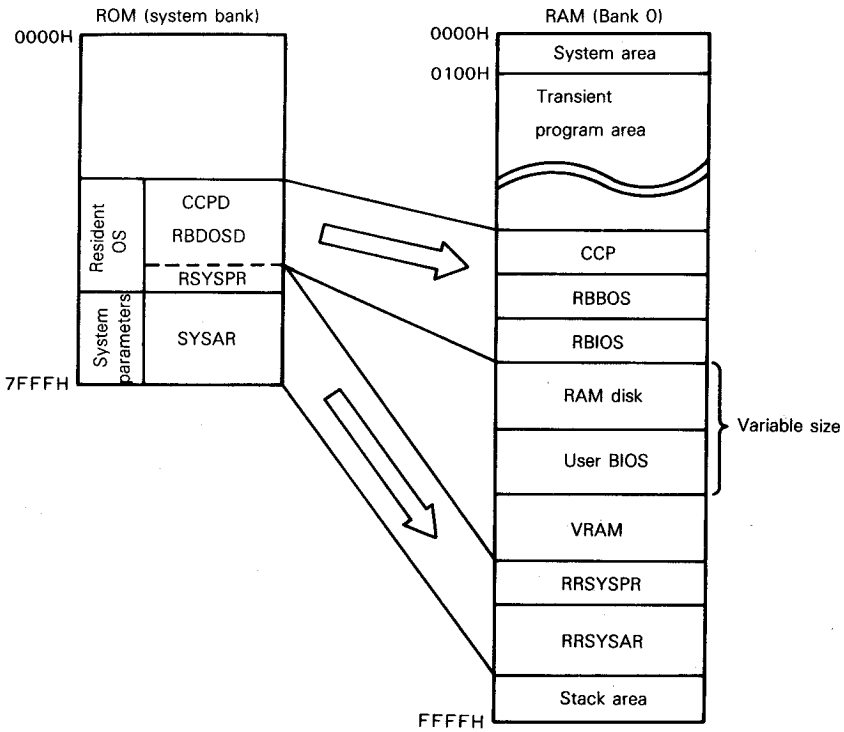
Appendix B

MEMORY MAPS AND MICROCASSETTE TAPE FORMAT

Memory map



Part of the OS program and system parameters in RAM (bank 0) is relocated by the relocato as shown below.

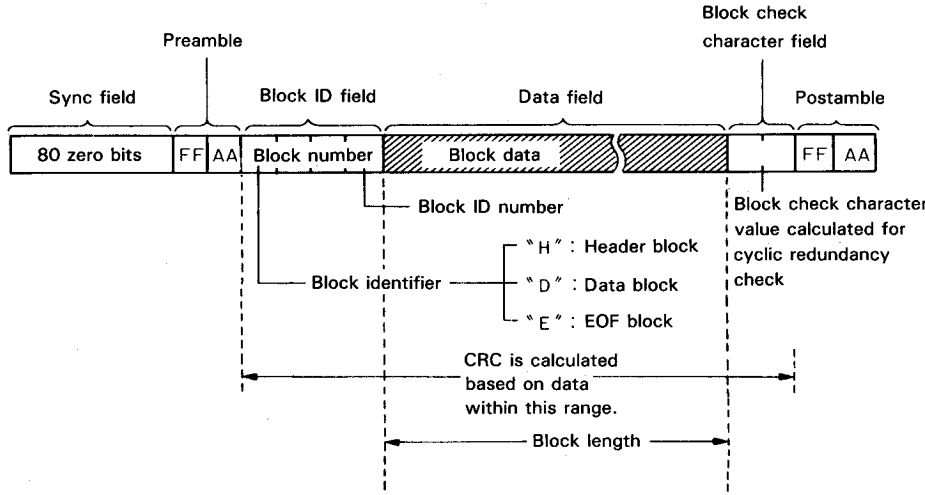


The memory map of ROM capsule (banks 1 and 2) differs according to the size of ROM (8KB, 16KB and 32KB).

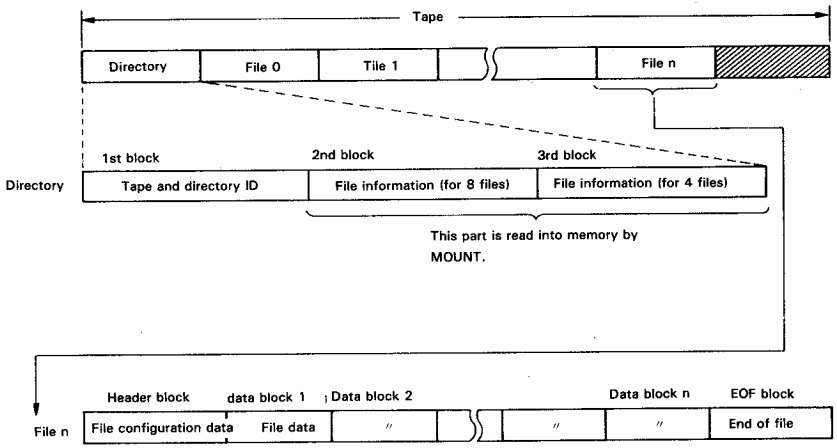
ROM size	ROM space	RAM space
8KB	C000H to E000H	0 to BFFFH, E000H to FFFFH
16KB	A000H to E000H	0 to 9FFFH, E000H to FFFFH
32KB	6000H to E000H	0 to 5FFFH, E000H to FFFFH

Microcassette tape format

Files are recorded on the microcassette tape in 256-byte blocks. Therefore, they are accessed in 256-byte blocks. The tape format is shown below.



One tape (one volume) contains one directory and one or more data files. The directory consists of three blocks: the first block contains the tape and directory ID and the other two blocks contain file location and other file information. One file consists of one header block, one or more data blocks and one EOF block. Tape structure is shown below.



Appendix C

BIOS SUBROUTINES

CP/M BIOS (Basic Input and Output System) has many useful subroutines which can be used by application programs. To use a BIOS subroutine, set parameters in appropriate registers and call the entry address for that subroutine. The following table lists the entry addresses for BIOS subroutines which can be called by application programs.

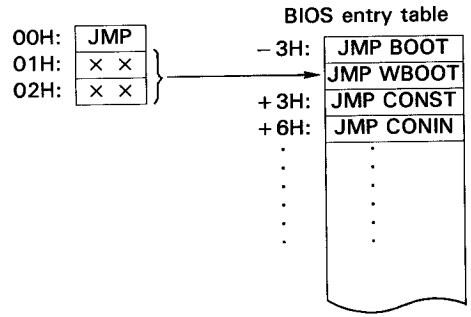
Address (Hex.)	Entry name	
WBOOT−3	BOOT	
+0	WBOOT	
+3	CONST	
+6	CONIN	
+9	CONOUT	
+C	LIST	
+F	PUNCH	
+12	READER	
+15	HOME	
+18	SELDSK	
+1B	SETTRK	
+1E	SETSEC	
+21	SETDMA	
+24	READ	
+27	WRITE	
+2A	LISTST	
+2D	SECTRN	
+30	PSET	
+33	SCRNDUMP	
+36	BEEP	
+39	(RSOPEN)	X
+3C	(RSCLOSE)	X
+3F	(RSINST)	X
+42	(RSINST)	X

Address (Hex.)	Entry name	
WBOOT+45	(RSIN)	X
+48	(RSOUT)	X
+4B	TIMDAT	
+4E	MEMORY	
+51	RSIOX	
+54	(LIGHTPEN)	X
+57	MASKI	
+5A	LOADX	
+5D	STORX	
+60	LDIRX	
+63	JUMPX	
+66	CALLX	
+69	GETPFK	
+6C	PUTPFK	
+6F	READSW	
+72	(SLAVE)	X
75	RDVRAM	
+78	MCMTX	
+7B	POWEROFF	
+7E	(USERBIOS)	X
+81	AUTOST	
+84	RESIDENT	
+87	CONTINUE	

X: Not supported by PX-4 OS. (Entry address only)

Although the absolute address of WBOOT varies according to the sizes of RAM disk and user BIOS area, locations 01H and 02H always contain the absolute address of WBOOT. Therefore, you can obtain the absolute addresses of other entry points easily.

Example 1: CONIN can be called as follows.



Program sample

```
                    : SET CONIN ENTRY
    LD   HL. (01H)  : GET (JMP WBOOT) ADDRESS FROM 0 PAGE
    ADD  HL. BC     : ADD 6
    LD   (ADRS). HL : STORE (JMP CONIN) ADDRESS
    DB   0CDH       : CALL (JMP CONIN)
ADRS: DB   0,0      : ADDRESS AREA
```

Example 2: CONIN can also be called from a BASIC program as follows.

```
10 STAD=&H50000
20 FOR I=1 TO 14
30 READ A
40 POKE STAD. A
50 STAD=STAD+1
60 NEXT I
70 DEF USR1=&H5000
80 X=USR1(X)
90 DATA &H01,&H06,&H00,&H2A,&H01,&H00,&H09,&H22,
&H0B,&HA0,&HCD,&H00,&H00,&HC9
```

Store the machine language program shown in Example 1 starting at 5000H.

The BIOS functions and user interfaces are described in the following. The contents of registers other than those to which a BIOS routine returns information are not assured after that routine has been executed unless otherwise specified. Therefore, the required contents of registers should be saved before calling a BIOS routine. Parameter blocks which are passed to or from a BIOS routine must be in the same memory bank as that from which the BIOS routine is called.

BOOT

Function: CP/M cold start bootstrap loader
Entry address: WBOOT - 3H
Entry parameters: None
Return parameters: C=00H
NOTE: This routine is used only when system initialization is made, [SHIFT] + [GRPH] are pressed together with the reset switch or the 7508 sub-CPU is reset. This routin is not used by the user.

WBOOT

Function: CP/M warm start bootstrap loader
Entry address: WBOOT + 0H
Entry parameters: None
Return parameters: C=drive number
NOTE: This routine is used when JUMP 0 is executed in an application program or power is turned on in the restart mode.

CONST

Function: Reads the console status.
Entry address: WBOOT + 3H
Entry parameters: None
Return parameters: A=00H Indicates that the console input buffer is empty.
A=FFH Indicates that the console input buffer contains characters.

CONIN

Function: Reads a character from the console.
Entry address: WBOOT + 6H
Entry parameters: None
Return parameters: A=ASCII code for the character read
(Register C is used if necessary.)

CONOUT

Function: Outputs a character to console
Entry address: WBOOT+9H
Entry parameters: C=ASCII code for character to be output
Return parameters: None.

LIST

Function: Outputs a character to the LST: device.
Entry address: WBOOT + 0CH
Entry parameters: C=character to be output
Return parameters: None.
NOTE: If the device is not ready, this routine loops until it becomes ready.

PUNCH

Function: Outputs a character to the PUN: device.
Entry address: WBOOT + 0FH
Entry parameters: C=character to be output
Return parameters: None.

READER

Function: Reads a character from the RDR: device.
Entry address: WBOOT + 12H
Entry parameters: None.
Return parameters: A=character read
NOTE: This routine loops until a character is read from the device which is assigned to RDR:.

HOME

Function: Sets the track to zero.
Entry address: WBOOT + 15H
Entry parameters: None.
Return parameters: None.

SELDSK

Function: Specifies a disk drive.
Entry address: WBOOT + 18H
Entry parameters: C=

0:	Drive A	(built-in/external RAM disk)
1:	Drive B	(ROM capsule)
2:	Drive C	(ROM capsule)
3:	Drive D	(Floppy disk drive 1)
4:	Drive E	(Floppy disk drive 2)
5:	Drive F	(Floppy disk drive 3)
6:	Drive G	(Floppy disk drive 4)
7:	Drive H	(Microcassette drive)
8:	Drive I	(RAM cartridge)
9:	Drive J	(ROM cartridge 1)
10:	Drive K	(ROM cartridge 2)

Return parameters: HL=00H: Parameter error
HL≠00H: disk parameter block address

SETTRK

Function: Selects the track to be read or written.
Entry address: WBOOT + 1BH
Entry parameters: BC=track number
Return parameters: None.
NOTE: The track number which can be specified is as follows.

Drive	Range
A	0 — 15
B	0 — 8
C	0 — 8
D	0 — 39
E	0 — 39
F	0 — 39
G	0 — 39
H	0 — 4
I	0 — 7
J	0 — 8
K	0 — 8

NOTE: If the track number specified is out of the range, an error will result when a READ or WRITE is performed. (In practice, the maximum value of the track number is limited by the capacity of the device installed.)

SETSEC

Function: Selects the sector to be read or written.
Entry address: WBOOT + 1EH
Entry parameters: BC = sector number (0 to 63)
Return parameters: None.
NOTE: The range of the sector number is from 0 to 63 (3FH). If a value out of this range is specified, an error will result when a READ or WRITE is performed.

SETDMA

Function: Specifies the DMA address for disk access.
Entry address: WBOOT + 21H
Entry parameters: BC = DMA address
Return parameters: None.

READ

Function: Reads 128 bytes of data from a disk.
Entry address: WBOOT + 24H
Entry parameters: None.
Return parameters: A = 00H - Normal completion
A = other than 0 - Abnormal end

WRITE

Function: Writes 128 bytes of data to a disk.
Entry address: WBOOT + 27H
Entry parameters: C = 00H - Write standard format data
C = 01H - Write unblocked data
C = 02H - Write sequential file
Return parameters: A = 00H - Normal completion
A = other than 0 - Abnormal end

LISTST

Function: Reads the status of the LST: device.
Entry address: WBOOT + 2AH
Entry parameters: None.
Return parameters: A = 00H - Busy
A = FFH - Ready.

SECTRN

Function: Converts a logical sector number into a physical sector number.
Entry address: WBOOT + 2DH
Entry parameters: BC = logical sector number
Return parameters: HL = physical sector number

PSET

Function: Performs various operations on VRAM data.
Entry address: WBOOT + 30H
Entry parameters: B = data
C = type of operation 01H: AND
02H: OR
03H: XOR
Other values: no operation
HL = VRAM relative address
$0 \leqq HL \leqq 1919$
Return parameters: A = 00H - Normal completion
A = 01H - Address out of range
C = result of operation
NOTE: This routine performs the operation specified by register C between data at the VRAM address specified by register pair HL and data in register B, then places the result at that VRAM address and register C. The contents of register B and register pair HL are not changed.

SCRNDUMP

Function: Outputs the contents of VRAM to LST:.
Entry address: WBOOT + 33H
Entry parameters: None.
Return parameters: (LSTERR) = 00H - Normal completion
(LSTERR) = FFH - Ended by [CTRL] + [STOP].
Location of LSTERR is 0F773H.

BEEP

Function: Sounds the speaker.
Entry address: WBOOT + 36H
Entry parameters: B = Note (13 to 60) 0: turns the speaker off
C = Interval (1 – 255) × 100 ms
Return parameters: A = 00H - Normal completion
= FFH - Ended by CTRL + STOP
NOTE: The relationship between the contents of register B and note is as follows. (No sound is generated when register B = 0.)

Note \ Octave	1	2	3	4	5
C		13	25	37	49
C#		14	26	38	50
D		15	27	39	51
D#		16	28	40	52
E		17	29	41	53
F		18	30	42	54
F#		19	31	43	55
G		20	32	44	56
G#		21	33	45	57
A		*22	34	46	58
A#		23	35	47	59
B		24	36	48	60

*: 440 Hz

NOTE: The pitch of sound can be specified as a frequency. Refer to the OS Reference Manual.

TIMDAT

Function: Performs the clock functions.
Entry address: WBOOT + 4BH
Explanation: This routine performs one of the six functions according to the contens of register C. The six functions are setting and reading the time, enabling and disabling the alarm function and reading and setting the alarm time. This routine uses a parameter block (referred to as the time descriptor) to obtain and return various information relating to the clock functions. This block is referred to as the time descriptor.

Time descriptor

Year (lower 2 digits)	2 BCD digits	1 byte
Month	2 BCD digits	1 byte
Day	2 BCD digits	1 byte
Hour	2 BCD digits	1 byte
Minute	2 BCD digits	1 byte
Second	2 BCD digits	1 byte
Weekday		1 byte
Type		1 byte
Address		2 bytes
Status		1 byte

Year: 00 - 99
Weekday: 0 - Sunday, 1 - Monday, ..., 6 - Saturday
Address: Indicates the starting address of the area which contains the alarm message or wake string.

33 bytes max.
Length
Message or string
(1 byte)
The address indicates the address of this byte.

Type: 1 byte
= 0 (not defined)
= 1 Alarm
= 2 Wake

The address and type parameters must be set by the user before calling this routine.
When reading the alarm/wake setting, it is not necessary to set the type parameter.

Status: The system sets the status parameter as follows.
Set to 0 when an alarm/wake time is set. (function No. = 82)
Set to 1 when an alarm/wake interrupt occurs.
Set to 0 after the alarm/wake time has been read.
As shown above, the status is set to 1 only when an alarm/wake interrupt occurs; otherwise, it is always set to 0.

NOTE: *Calling side bank information is automatically processed by BIOS.*

A) TIMDAT (Read Time)

Function: Reads the time.

Entry parameters: C = 00H - Time read function
DE = Starting address of a 7 byte block for the time descriptor.

Return parameters: DE = Not changed
Time descriptor

Year (lower 2 digits)	2 BCD digits	1 byte
Month	2 BCD digits	1 byte
Day	2 BCD digits	1 byte
Hour	2 BCD digits	1 byte
Minute	2 BCD digits	1 byte
Second	2 BCD digits	1 byte
Weekday		1 byte

Weekday: 0 - Sunday, 1 - Monday, ... ,
6 - Saturday

B) TIMDAT (Set Time)

Function: Sets the time.

Entry parameters: C = FFH - Time set function
DE = Starting address of a 7 byte block for the time descriptor

Time descriptor

Year (lower 2 digits)	2 BCD digits	1 byte
Month	2 BCD digits	1 byte
Day	2 BCD digits	1 byte
Hour	2 BCD digits	1 byte
Minute	2 BCD digits	1 byte
Second	2 BCD digits	1 byte
Weekday		1 byte

Weekday: 0 - Sunday, 1 - Monday, ... ,
6 - Saturday

Return parameters: None. (DE = Not changed).

C) TIMDAT (Alarm/Wake enable)

Function: Enables the alarm/wake function.
Entry parameters: C = 80H - Alarm enable
Return parameters: None.

D) TIMDAT (Alarm/Wake disable)

Function: Disables the alarm/wake function.
Entry parameters: C = 81H - Alarm disable
Return parameters: None.

E) TIMDAT (Set Alarm/Wake)

Function: Sets the alarm/wake time.
Entry parameters: C = 82H - Alarm set
DE = Starting address of the time descriptor

Time descriptor

Year (lower 2 digits)	2 BCD digits	1 byte
Month	2 BCD digits	1 byte
Day	2 BCD digits	1 byte
Hour	2 BCD digits	1 byte
Minute	2 BCD digits	1 byte
Second	2 BCD digits	1 byte
Weekday		1 byte
Type		1 byte
Address		2 bytes

Year: 00 - 99

Weekday: 0 - Sunday, 1 - Monday, ... ,
6 - Saturday

Address: Indicates the starting address of the area which contains the alarm message or wake string.

Type: 1 byte

= 0 (not defined)
= 1 Alarm
= 2 Wake

Return parameters: None. (DE = Not changed.)

F) TIMDAT (Read Alarm/Wake)

Function: Reads the alarm/wake time.
Entry parameters: C = 84H
DE = Starting address of the time descriptor

Time descriptor

Year (lower 2 digits)	2 BCD digits	1 byte
Month	2 BCD digits	1 byte
Day	2 BCD digits	1 byte
Hour	2 BCD digits	1 byte
Minute	2 BCD digits	1 byte
Second	2 BCD digits	1 byte
Weekday		1 byte
Type		1 byte
Address		2 bytes
Status		1 byte

Year: 00 - 99

Weekday: 0 - Sunday, 1 - Monday, ..., 6 - Saturday

Address: Indicates the starting address of the area which contains the alarm message or wake string.

Type: 1 byte
= 0 (not defined)
= 1 Alarm
= 2 Wake

Alarm: 1 byte
= 0 Not yet sounded
= 1 Sounded

Return parameters: None. (DE = Not changed)
NOTE: FFH is set to the year parameter byte and FH is set to the lower 4 bits for the second parameter byte.

To reset the alarm/wake function, to disable the function with function D, then zero all parameters with function E.

MEMORY

Function: Reads the current bank.
Entry address: WBOOT + 4EH
Entry parameters: None.
Return parameters: C = FFH - System bank
= 00H - Bank 0
= 01H - Bank 1
= 02H - Bank 2

RSIOX

Function: Performs serial I/O operation.
Entry address: WBOOT + 51H
Entry parameters: B = FXH
NOTE: This routine performs serial communication through the RS-232C, SIO and cartridge serial interfaces. The routine supports 12 functions and the setting of entry parameter FX for these functions are shown below.

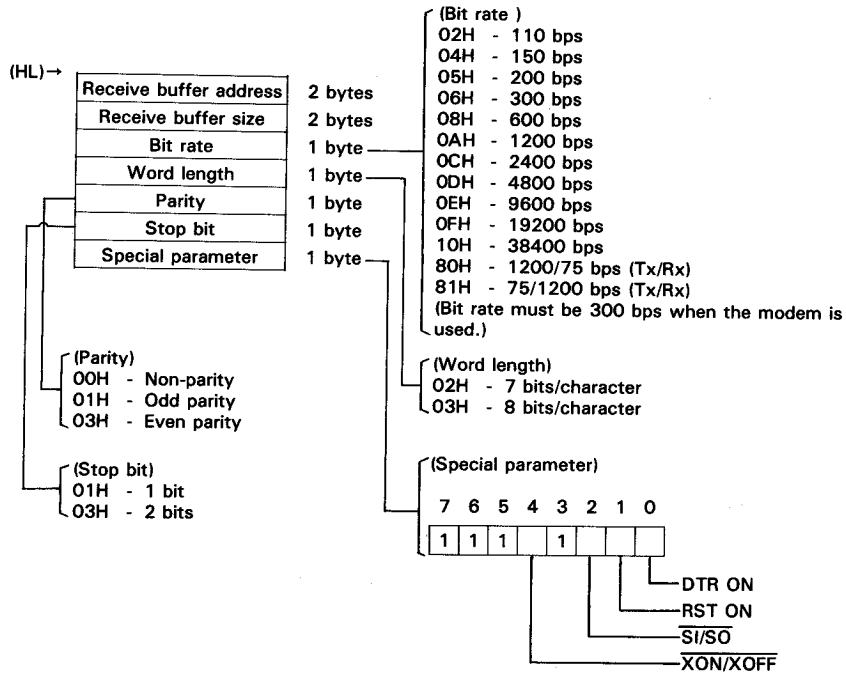
FXH

- Device (OPEN only)
  - 0: RS-232C
  - 1: SIO
  - 2: RS-232C input
  - 3: Cartridge serial
- Function code
  - 1: OPEN
  - 2: CLOSE
  - 3: INSTS
  - 4: OUTST
  - 5: GET
  - 6: PUT
  - 7: CTRLIN
  - 8: SETCTRL
  - 9: ERSTS
  - F: SENSE

Explanation:

A) OPEN

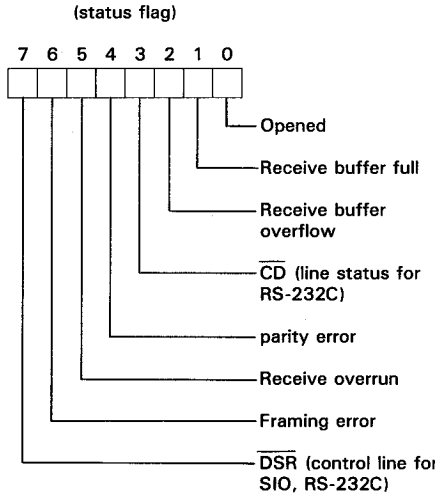
Function: Opens the specified device.
Entry parameters: B = 1XH (X = 0 to 3)
HL = Starting address of the parameter block



Return parameters: A = 00H - Normal open, Z = 1
A = 02H - The interface already open, Z = 0
HL = Not changed

(HL)→

Status flag	1 byte
Receive buffer get point	2 bytes
Receive buffer put point	2 bytes
*Receive buffer address	2 bytes
*Receive buffer size	2 bytes



Bits 0 to 2 are used by the system. CD is active when "1" and DSR is active when "0".

(Receive buffer get point)
The pointer used to read data from the buffer.

(Receive buffer put pointer)
The pointer used to store received data in the buffer.

B) CLOSE

Function: Closed the currently used serial device.
Entry parameters: B = 20H
Return parameters: None.

C) INSTS

Function: Checks whether there is any data in the receive buffer.
Entry parameters: B = 30H
HL = Starting address of the 9 byte block which is used to store return information.
Return parameters: Z = 1 - Normal completion
A = FFH - Data has been received
= 00H - No data in the receive buffer
BC = Number of bytes received
HL = Not changed
The parameter block is the same as for OPEN.
Z = 0 - Abnormal end
A = 03H - Device not opened

D) OUTST

Function: Checks whether output is enable.
(Checks whether the Tx buffer is empty.)
Entry parameter: B = 40H
HL = Starting address of the 9 byte block which is used to store return information.
Return parameters: Z = 1 - Normal completion
A = FFH - Output enabled
= 00H - Output disabled
HL = Not changed
The parameter block is the same as for OPEN.
Z = 0 - Abnormal end
A = 03H - Device not opened

E) GET

Function: Reads 1 byte data from the receive buffer.
Entry parameters: B = 50H
HL = Starting address of the 9 byte block which is used to store return information.
Return parameters: Z = 1 - Normal completion
A = Receive data
HL = Not changed

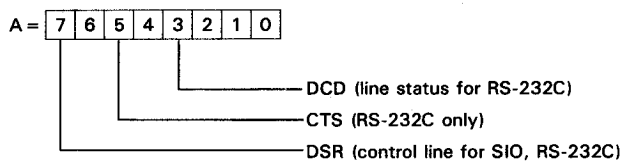
The parameter block is the same as for OPEN.
Z = 0 - Abnormal end
A = 03H - Device not open
A = 04H - CTRL/STOP
A = 05H - receive buffer overflow

F) PUT

Function: Sends 1 byte data.
Entry parameters: B = 60H
C = Send data
HL = Starting address of the 9 byte block which is used to store return information.
Return parameters: Z = 1 - Normal completion
HL = Not changed (return information address)
Z = 0 - Abnormal end
A = 03H - Device not open
= 04H - CTRL + STOP pressed.
The parameter block is the same as for OPEN.

G) CTLIN

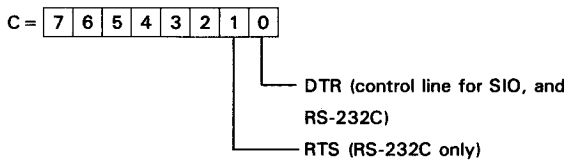
Function: Reads the status of the control line.
Entry parameters: B = 70H
Return parameters: Z = 1 - Normal completion



Z = 0 - Abnormal end
A = 03H - Device not open

H) SETCTL

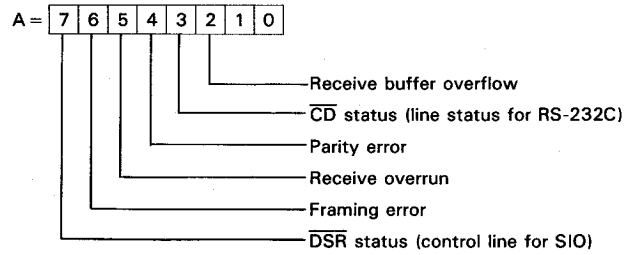
Function: Sets the control lines.
Entry parameters: B = 80H



Return parameters: Z = 1 - Normal completion
Z = 0 - Abnormal end
A = 03H - Device already open

I) ERSTS

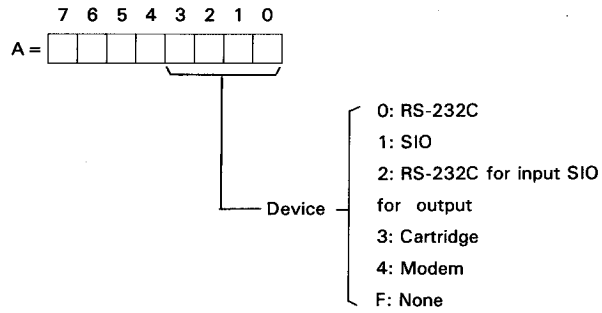
Function: Reads the error status and clears the error flags.
Entry parameters: B = 90H
Return parameters: Z = 1 - Normal completion



Z = 0 - Abnormal end
A = 03H - Device not open

J) SENS

Function: Reads the status of the serial device and RSBYTE.
Entry parameters: B = F0H
Return parameters:

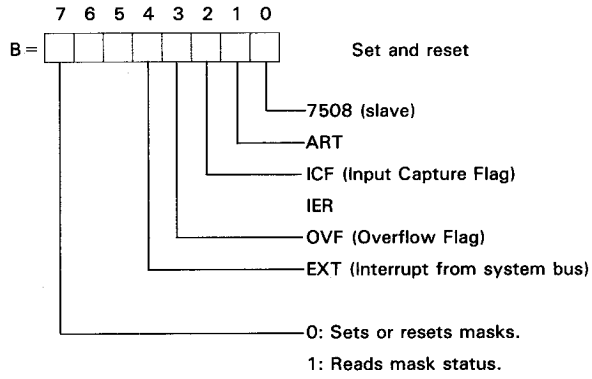


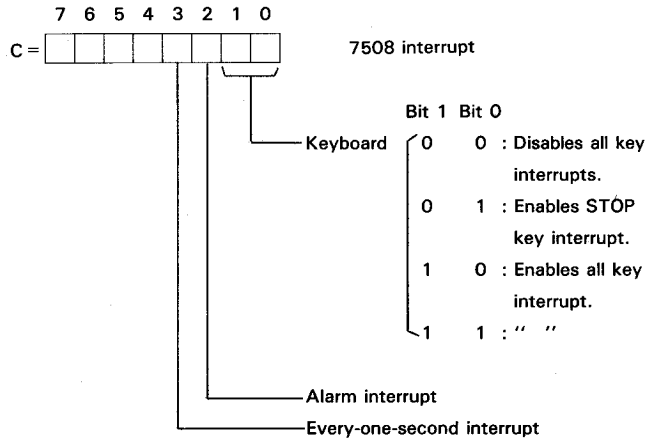
MASKI

Function: Sets and resets interrupt masks and enables and disables 7508 interrupts.

Entry adress: WBOOT + 57H

Entry parameters:





Each bit enables the corresponding interrupt when it is "1" and disables the interrupt when it is "0"

Return parameters: B = Previous IER status
C = Previous 7508 interrupt mask status
The contents of all registers other than registers B and C are not changed.

LOADX

Function: Reads one byte of data from the specified memory bank.

Entry address: WBOOT + 5AH

Entry parameters: C = FFH - System bank
C = 00H - Bank 0
C = 01H - Bank 1
C = 02H - Bank 2
HL = Data address (must be in the bank in which the calling program is present.)

Return parameters: A = read data
The contents of other registers are not changed.

STORX

Function: Writes one byte of data in the specified bank.

Entry address: WBOOT + 5DH

Entry parameters: C = FFH - System bank
C = 00H - Bank 0
C = 01H - Bank 1
C = 02H - Bank 2
A = Data to be written
HL = Address to which data is written (Must be in the called bank.)

Return parameters: None.
The contents of all registers are not changed.

LDIRX

Function: Transfers data from the specified bank to bank 0.

Entry address: WBOOT + 60H

Entry parameters: A = destination bank
A = FFH - System bank
A = 00H - Bank 0
A = 01H - Bank 1
A = 02H - Bank 2
HL = Starting address of the memory area in the source bank from which data is transferred
BC = Number of bytes transferred
DE = Starting address of the memory area in bank 0 to which data is transferred.

Return parameters: None.

JUMPX

Function: Jumps to the specified bank.
Entry address: WBOOT + 63H
Entry parameters: IX = destination address
ADRS = FFH - System bank
= 00H- Bank 0
= 01H- Bank 1
= 02H- Bank 2
ADRS is at 0F52EH.
Return parameters: None.
(Control is not returned to the calling program.)
NOTE: If a stack is used by the destination routine, new one must be set.

CALLX

Function: Calls a subroutine in the specified bank.
Entry address: WBOOT + 66H
Entry parameters: IX = destination address
ADRS = FFH - System bank
= 00H - Bank 0
= 01H - Bank 1
= 02H - Bank 2
ADRS is at 0F52EH.
Return parameters: Depending on the user subroutine called (except IX and IY).
NOTE: If a stack is used in the subroutine, new one must be set in the subroutine. The previous stack must be restored when control is returned to the calling program.

GETPFK

Function: Reads the string assigned to a function key.
Entry address: WBOOT + 69H
Entry parameters: C = Function key number
0-9: Function keys on the standard keyboard
40-7E: Item function key position code
HL = Starting address of the meory block into which the string is read.
Return parameters: None.
HL = Not changed.
NOTE: The format of the read buffer is as follows.

No. of char.	Char. 1	Char. 2		Char. n

(HL): Number of characters are that actually stored in the buffer and its maximum value is 15.
This routine allows the user who has the standard keyboard to use the functions assigned to item functions 40 to 7E. Bank information of the calling program is automatically processed by this routine.

PUTPFK

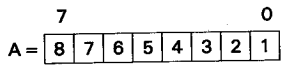
Function: Assigns a character string to the specified function key.
Entry address: WBOOT + 6CH
Entry parameters: C = Function key number or item function key control information
0H - 9H: Standard function key number
40-7EH: Item function key position code
FFH: Reset all function key assignments
FEH: Resets the item flag.
FDH: Sets the item flag.
HL = Starting address of the memory area which contains the string to be assigned to the specified key.
Return parameters: None.
HL = Not changed
NOTE: The format of the character string is as follows.

No. of char.	Char. 1	Char. 2		Char. n

(HL): The maximum number of characters is 15.
The item function specified is valid when the item flag is set.
This routine allows the user who has the standard keyboard to use the functions assigned to item functions 40 to 7E. Bank information of the calling program is automatically processed by this routine.

READSW

Function: Reads the status of switches.
Entry address: WBOOT + 6FH
Entry parameters: C = 02H - Read DIP SW status
C = 04H - Read power SW status
C = Other than 2 and 4 - No operation
Return parameters: 1) Read DIP SW status:



When each DIP switch position is ON, the corresponding bit is set to "1".

2) Read power switch status:
A = 00H: Power switch OFF
= 01H: Power switch ON

RDVRAM

Function: Reads the contents of the virtual screen.
Entry address: WBOOT + 75H
Entry address: B = Column at which read is to start (1 - m)
C = Line at which read is to start (1 - n)
DE = Number of characters to be read
HL = Starting address of the area in which data read is to be stored
Return parameters: A = 00H - Notmal completion
A = 01H - Read error
A = FFH - Parameter error

NOTE: M and n are determined by the virtual screen size and m must be 40 or 80 and n must be equal to or less than 25. If the screen end is encounterred during read, the remaining area is filled with blanks (20H) and register A is set to 01H.
The address specified and the calling program must be in the same bank.
All characters are returned in ASCII codes.

MCMTX

Function: Performs processings relating to the microcassette.
Entry address: WBOOT + 78H
NOTE: See the OS Reference Manual (sold separately).

POWEROFF

Function: Turns off the system's power.
Entry address: WBOOT + 7BH
Entry parameters: C = 00H - Turns off power in the continue mode.
C = 01H - Turns off power in the restart mode.
Return parameters: None.

USERBIOS

Function: BIOS entry for user coded BIOS routine
Entry address: WBOOT + 7EH
Entry parameters: Defined by the user.
Return parameters: Defined by the user.
Explanation: This entry is provided so that the user can code his own BIOS routine. The user must reserve the area for his routine (at the time of system initialize or using the CONFIG command) and write the starting address of the routine at the entry address. For details, refer to the OS Reference Manual (sold separately).

AUTOST

Function: Sets and cancels the auto start string.
Entry address: WBOOT + 81H
Entry parameters: C = 00H - Cancels the auto start string.
C = 01H - Sets an auto start string.
HL = Starting address of the area in which the string to be set is stored (Valid when C = 01H)

Length	

(HL) Max. length is 32 bytes.

Return parameters: None.

RESIDENT

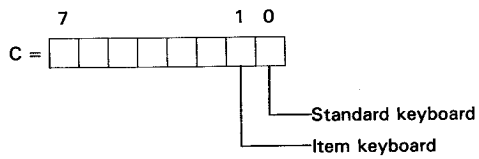
Function: Sets and resets the resident flag.
Entry address: WBOOT + 84H
Entry parameters: C = 00H - Resets the resident flag.
C = 01H - Sets the resident flag.
Return parameters: None.

CONTINUE

Function: Sets and resets the continue flag.

Entry address: WBOOT + 87H

Entry parameters:



"1": Set

"0": Reset

Return parameters: None.

Appendix D

CHARACTER CODE TABLE

Higher bits

Lower bits	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	HELP		SP	0	@	P	'	p	+	○	SP				▵	
1			!	1	A	Q	a	q	┴	♠					↵	
2		INS	"	2	B	R	b	r	┬	♥						
3	STOP	PAUSE	#	3	C	S	c	s	┤	♦						
4			$	4	D	T	d	t	├	♣						
5			%	5	E	U	e	u	─	♪						
6			&	6	F	V	f	v	│	▵						
7			'	7	G	W	g	w	┌	±						
8	BS		(	8	H	X	h	x	┐	◘						
9	TAB		)	9	I	Y	i	y	└	○						
A			*	:	J	Z	j	z	┘	◙						
B	HOME	ESC	+	;	K	[	k	{	▒	↑						
C	CLR	→	,	<	L	¥	l	¦	■	↓						
D	RET	←	−	=	M	]	m	}	▬	×						
E		↑	.	>	N	^	n	~	▮	÷						
F		↓	/	?	O	_	o	△	●	±						

SP = space

Appendix E

CONSOLE CONTROL CODES

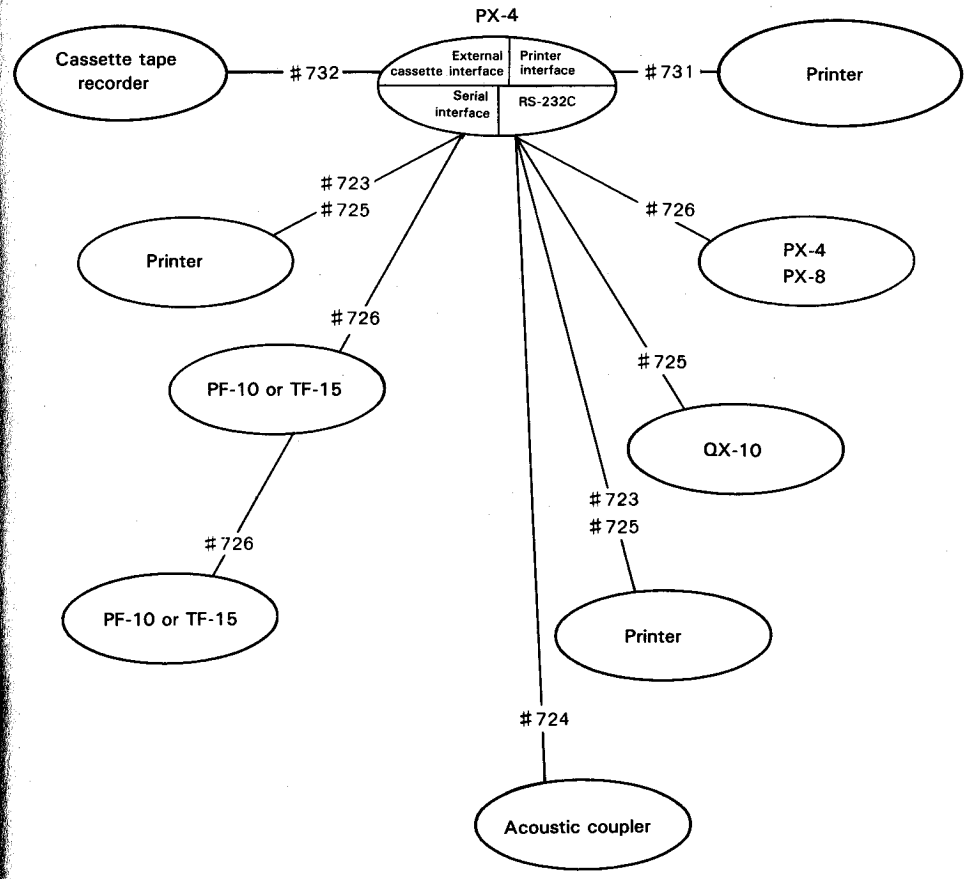
Code		Function
Decimal	Hexadecimal	
2	02	Moves the window 40 characters to the left.
5	05	Deletes characters to the end of the line from the cursor position.
6	06	Moves the window 40 characters to the right.
7	07	Sounds the speaker at 880 Hz.
8	08	Deletes the character to the left of the cursor position.
9	09	Moves the cursor to the next tab position.
10	0A	Moves the cursor down one line.
11	0B	Moves the cursor to the home position.
12	0C	Clears the currently selected virtual screen.
13	0D	Executes a command or statement.
16	10	Moves the window up one screen (7 or 8 lines).
17	11	Moves the window down one screen (7 or 8 lines).
26	1A	Erases all characters to the end of the virtual screen from the cursor position.
27	1B	Escape code
28	1C	Moves the cursor one character to the right.
29	1D	Moves the cursor one character to the left.
30	1E	Moves the cursor up one line.
31	1F	Moves the cursor down one line.

NOTE:
An escape code, followed by one or more characters is referred to as a escape sequence. The escape sequences are listed and explained in the OS Reference Manual (sold separately).

Appendix F

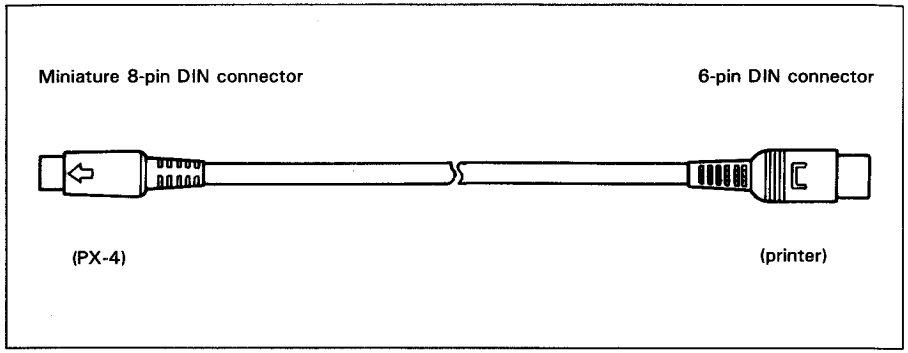
INTERFACE CABLES

The following cables are used to connect optional devices to the PX-4.



Each cable is explained in the following.

Cable set # 723

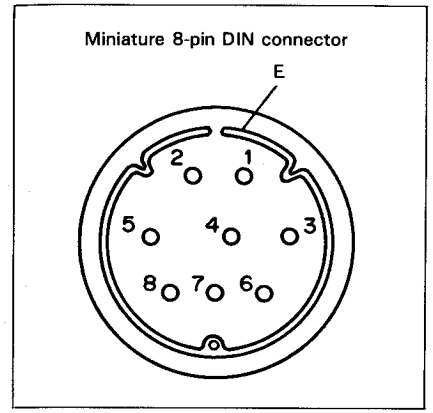


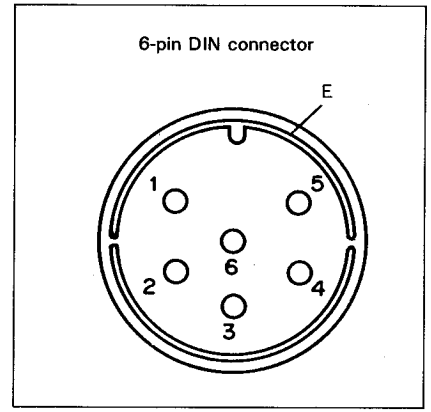
Purpose: Connects P-40 or P-80 printer to the PX-4.

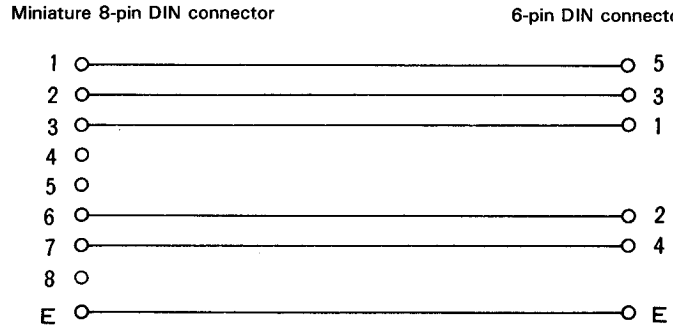
Connection: **Connecting a printer to the PX-4**

Connect the miniature 8-pin DIN connector to the RS-232C or serial interface connector of the PX-4 and the 6-pin DIN connector to the 6-pin DIN connector of the printer.

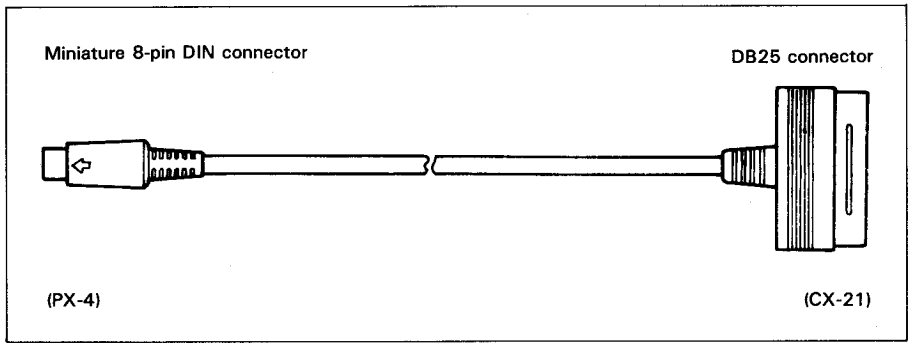
Pin assignments:







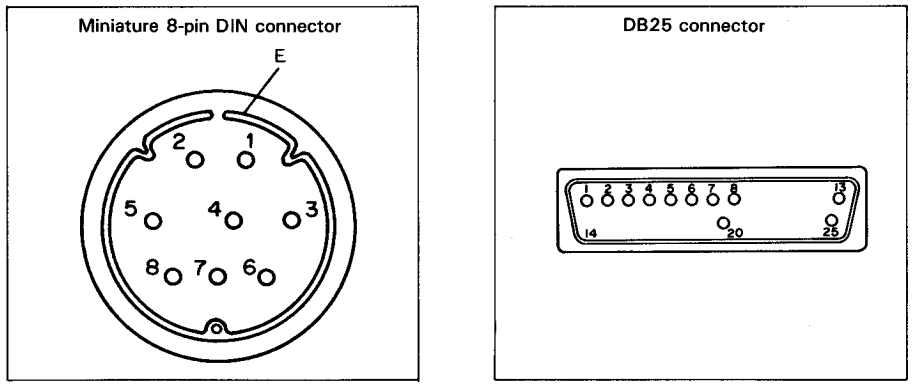
Cable set #724

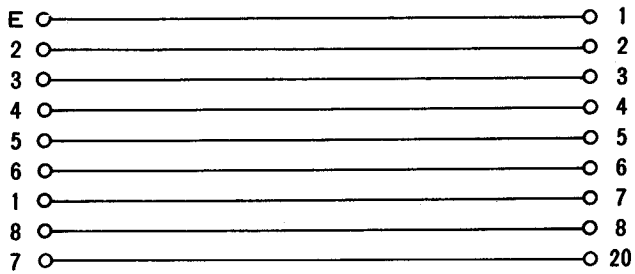


Purpose: Connects a CX-21 acoustic coupler to the PX-4.

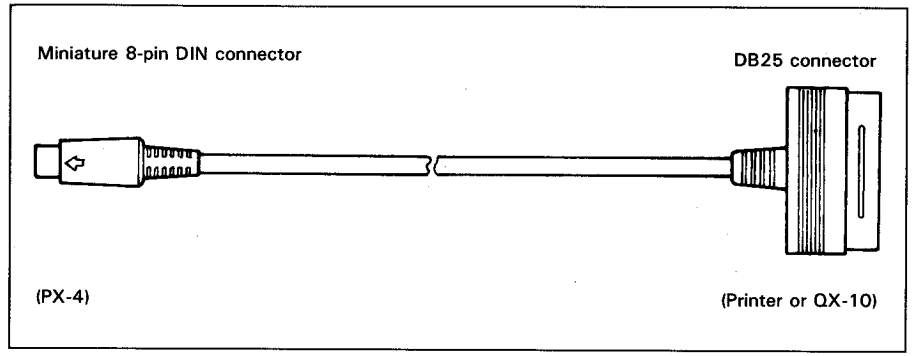
Connection: Connect the miniature 8-pin DIN connector to the RS-232C interface connector of the PX-4 and the DB25 connector to the CX-21.

Pin assignments:





Cable set #725



Purpose: Connects a printer with an RS-232C interface or QX-10 to the PX-4.

Connection:

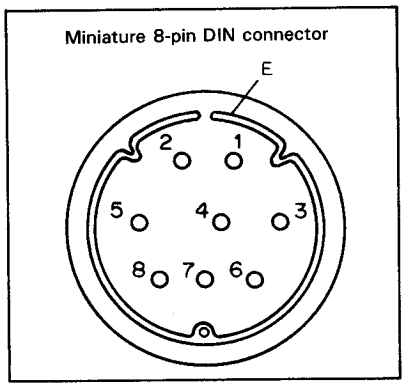
Connecting a printer to the PX-4.

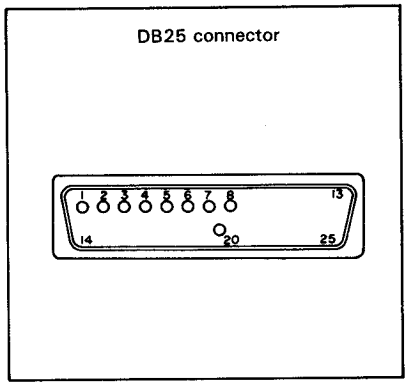
Connect the miniature 8-pin DIN connector to the RS-232C interface or serial connector (which is used is determined by the DIP switch in the ROM capsule compartment), then connect the DB25 connector to the RS-232C interface connector of the printer.

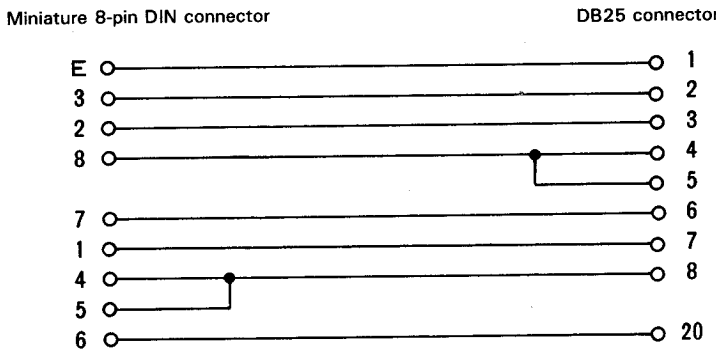
Connecting QX-10 to the PX-4

Connect the miniature 8-pin DIN connector to the RS-232C interface connector of the PX-4 and the DB25 connector to the RS-232C interface connector of the QX-10.

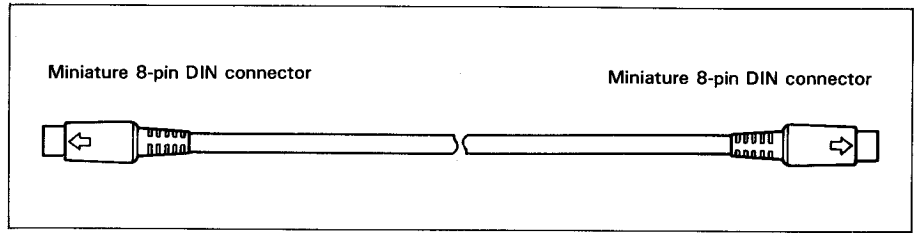
Pin assignment:







Cable set #726



Purpose: Connects the PX-4 to another PX-4 or PX-8.
Connects a PF-10 or TF-15 floppy disk unit to the PX-4.
Connects a PF-10 to another PF-10 or a TF-15 to another TF-15.
Connects a TF-15 to a PF-10.

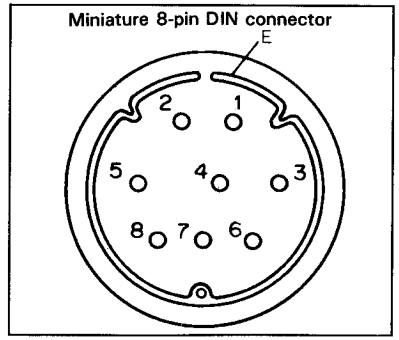
Connection:

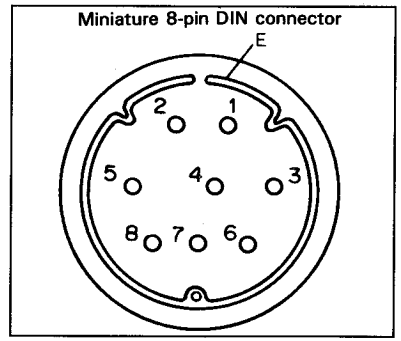
Connecting two PX-4s or connecting PX-4 to PX-8
Connect the miniature connectors to the RS-232C interface connectors of the computers.

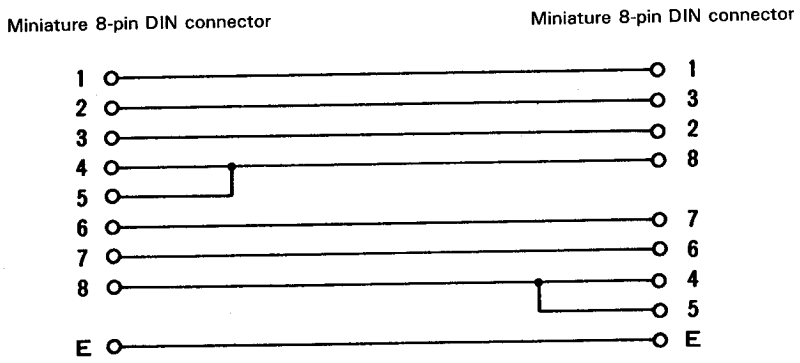
Connecting a PF-10 or TF-15 to PX-4
Connect one miniature connector to the serial interface of the PX-4 and the other to the disk unit.

Connecting PF-10s and TF-15s
Connect each miniature connector to the 8-pin connector of each disk unit.

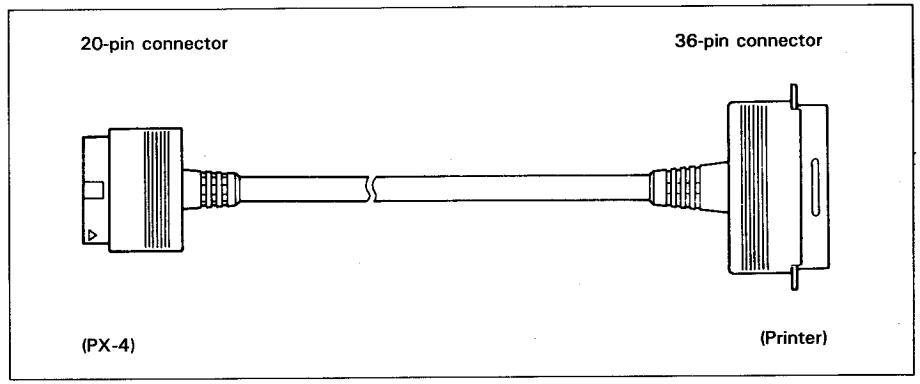
Pin assignment:







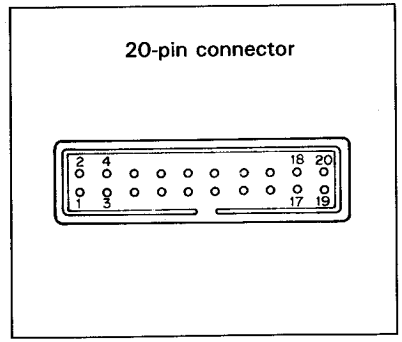
Cable set #731

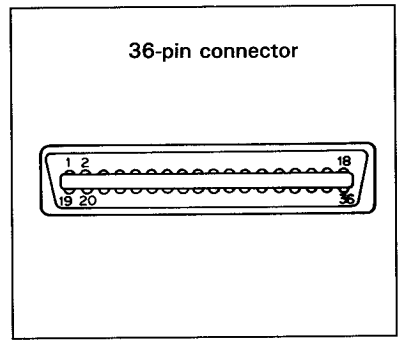


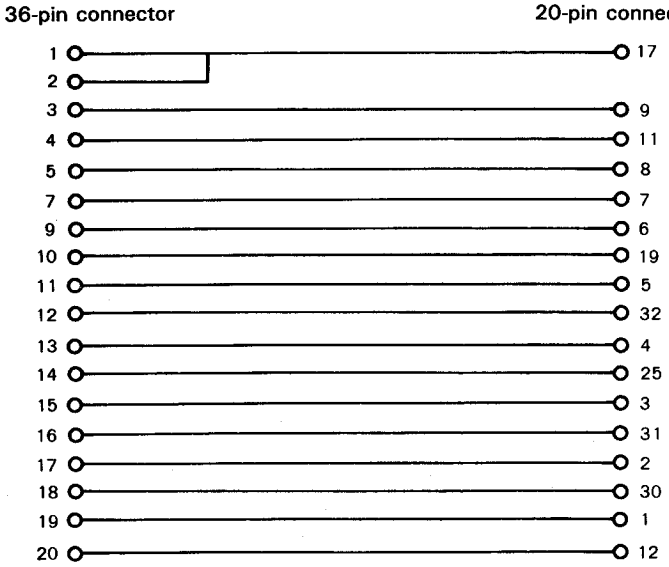
Purpose: Connects a parallel printer to the PX-4.

Connection: Connect the 20-pin connector to the printer interface connector of the PX-4 and the 36-pin connector to the parallel interface connector of the printer.

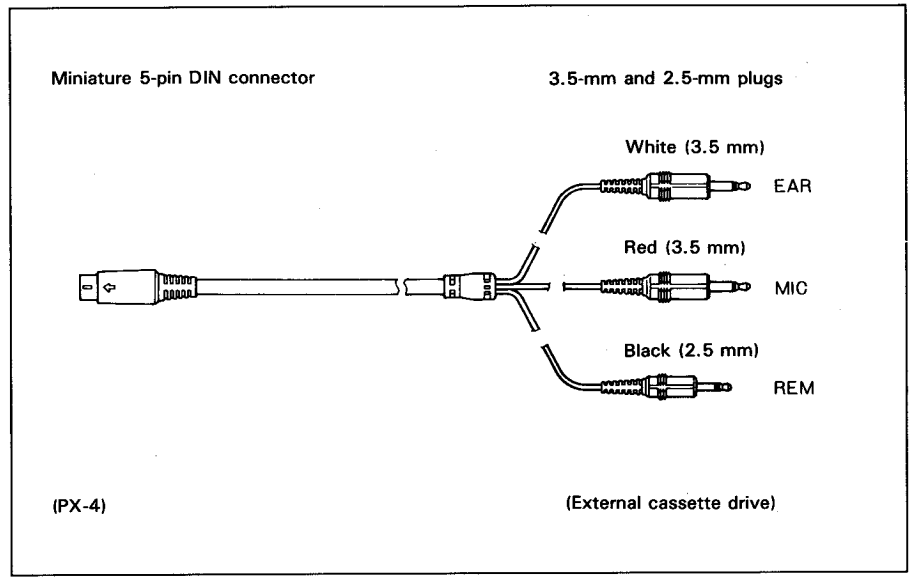
Pin assignment:







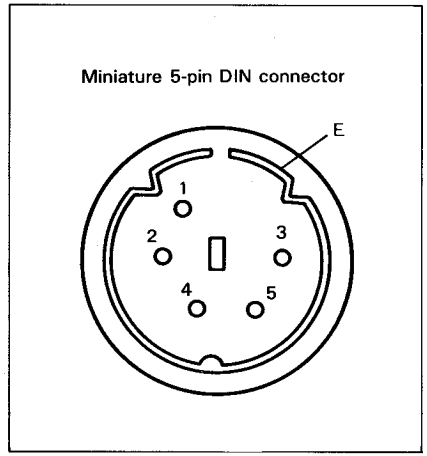
Cable set # 732

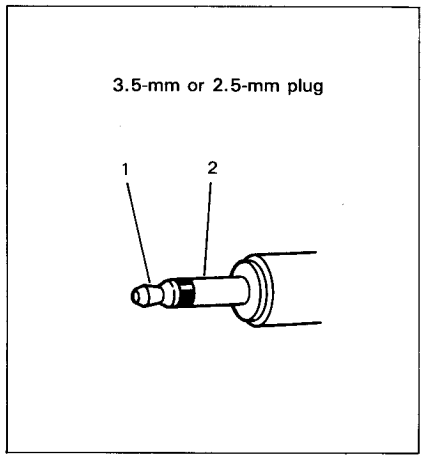


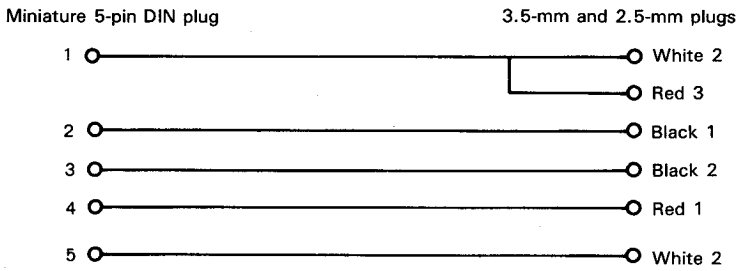
Purpose: Connects an external cassette drive to the PX-4.

Connection: Connect the miniature DIN connector to the external audio cassette interface connector. Insert the white plug to the EAR jack of the cassette drive, the red plug to the MIC jack and the black plug to the REM jack.

Pin assignment:







Appendix G

CP/M Errors and Messages

When using CP/M and the associated utilities, many possible errors can occur. Messages can come from different sources. They can be dislayed when there are errors in calls to the Basic Disk Operating System (BDOS). CP/M also displays errors when there are errors in command lines. The following list of error messages and sources of error covers errors in CP/M and the standard utilities. Some of these utilities may only be supplied on disk, but the error messages are presented as a single table to cover all these cases. Other application programs and the TERM and FILINK utility programs have their own error messages. Please consult the sections in this manual approprite to these utilities or the manual provided with the application program, when using such programs.

Message	Meaning
?	This message has four possible meanings: 1) DDT does not understand the assembly language instruction. 2) The file cannot be opened. 3) A checksum error occurred in a HEX file. 4) The assembler/disassembler was overlayed.
ABORTED	You stopped a PIP operation by pressing a key.
ASM Error Messages	D Data error: data statement element cannot be placed in specified data area. E Expression error: expression cannot be evaluated during assembly. L Label error: label cannot appear in this context (might be duplicate label). N Not implemented: unimplemented features, such as macros, are trapped. O Overflow: expression is too complex to evaluate. P Phase error: label value changes on two passes through assembly. R Register error: the value specified as a register is incompatible with the code. S Syntax error: improperly formed expression.

Message	Meaning
	U Underlined label: label used does not exist. V Value error: improperly formed operand encountered in an expression.
BAD DELIMITER	Check command line for typing errors.
Bad Load	CCP error message, or SAVE error message.
Bdos Err On d:	Basic Disk Operating System Error on the designated drive: CP/M replaces d: with the drive specification of the drive where the error occurred. This message is followed by one of the four phrases in the situations described below.
Bdos Err On d: Bad Sector	This message appears when CP/M finds no disk in the drive, when the disk is improperly formatted, when the drive latch is open, or when power to the drive is off. Check for one of these situations and try again. This could also indicate a hardware problem or a worn or improperly formatted disk. Press CTRL-C to terminate the program and return to CP/M, or press the return key to ignore the error.
Bdos Err On d: File R/O	You tried to erase, rename, or set file attributes on a Read-Only file. The file should first be set to Ready-Write (RW) with the command: "STAT filespec $R/W."
Bdos Err On d: R/O	Drive has been assigned Read Only status with a STAT command, or the disk in the drive has been changed without being initialized with a CTRL-C. CP/M terminates the current program as soon as you press any key.
Bdos Err on d: Select	CP/M received a command line specifying a nonexistent drive. CP/M terminates the current program as soon as you press any key. Press return key or CTRL-C to recover.
Break "x" at c	"x" is one of the symbols described below and c is the command letter being executed when the error occurred. # Search failure. ED cannot find the string specified in an F, S, or N command. ? Unrecognized command letter c. ED does not recognize the indicated command letter, or an E, H, Q, or O command is not alone on its command line. O The file specified in an R command cannot be found.

Message	Meaning
	> Buffer full. ED cannot put any more characters in the memory buffer, or the string specified in an F, N, or S command is too long. E Command aborted. A keystroke at the console aborted command execution. F Disk or directory full. This error is followed by either the disk or directory full message. Refer to the recovery procedures listed under these messages.
CANNOT CLOSE DESTINATION FILE-{filespec}	An output file cannot be closed. You should take appropriate action after checking to see if the correct disk is in the drive and that the disk is not write protected.
Cannot close, R/O CANNOT CLOSE FILES	CP/M cannot write to the file. This usually occurs because the disk is write-protected. An output file cannot be closed. This is a fatal error that terminates ASM execution. Check to see that the disk is in the drive, and that the disk is not write-protected. The disk file written by a W command cannot be closed. This is a fatal error that terminates DDT execution. Check if the correct disk is in the drive and that the disk is not write-protected. This error can occur during SUBMIT file processing. Check if the correct system disk is in the A drive and that the disk is not write-protected. The SUBMIT job can be restarted after rebooting CP/M.
CANNOT READ	PIP cannot read the specified source. Reader may not be implemented.
CANNOT WRITE	The destination specified in the PIP command is illegal. You probably specified an input device as a destination.
Checksum error	A hex record checksum error was encountered. The hex record that produced the error must be corrected, probably by recreating the hex file.
CHECKSUM ERROR LOAD ADDRESS hhhh ERROR ADDRESS hhhh BYTES READ: hhhh:	File contains incorrect data. Regenerate hex file from the source.
Command Buffer Overflow	The SUBMIT buffer allows up to 2048 characters in the input file.

Message	Meaning
Command too long	A command in the SUBMIT file cannot exceed 125 characters.
CORRECT ERROR, TYPE RETURN OR CTRL-Z	A hex record checksum was encountered during the transfer of a hex file. The hex file with the checksum error should be corrected, probably by recreating the hex file.
DESTINATION IS R/O, DELETE (Y/N)?	The destination file specified in a PIP command already exists and it is Read Only. If you type Y, the destination file is deleted before the file copy is done.
Directory full	There is not enough directory space for file being written to the destination disk. You can use the OX filespec command to erase any unnecessary files on the disk without leaving the editor.
	There is not enough directory space to write the $$$.SUB file used for processing SUBMITs. Erase some files or select a new disk and retry.
Disk full	There is not enough disk space for the output file. This error can occur on the W, E, H, or X commands. If it occurs with X command, you can repeat the command prefixing the filename with a different drive.
DISK READ ERROR-{filespec}	The input disk file specified in a PIP command cannot be read properly. This is usually the result of an unexpected end-of-file. Correct the problem in your file.
DISK WRITE ERROR-{filespec}	A disk write operation cannot be successfully performed during a W command, probably due to a full disk. You should either erase some unnecessary files or get another disk with more space.
	A disk write operation cannot be successfully performed during a PIP command, probably due to a full disk. You should either erase some unnecessary files or get another disk with more space and execute PIP again.
	The SUBMIT program cannot write the $$$.SUB file to the disk. Erase some files, or select a new disk and try again.
ERROR: BAD PARAMETER	You entered an illegal parameter in a PIP command. Retype the entry correctly.

Message	Meaning
ERROR: CANNOT OPEN SOURCE, LOAD ADDRESS hhhh	Displayed if LOAD cannot find the specified file or if no filename is specified.
ERROR: CANNOT CLOSE FILE, LOAD ADDRESS hhhh	Caused by an error code returned by a BDOS function call. Disk may be write protected.
ERROR: CANNOT OPEN SOURCE, LOAD ADDRESS hhhh	Cannot find source file. Check disk directory.
ERROR: DISK READ, LOAD ADDRESS hhhh	Caused by an error code returned by a BDOS function call.
ERROR: DISK WRITE, LOAD ADDRESS hhhh	Destination Disk is full.
ERROR: INVERTED LOAD ADDRESS, LOAD ADDRESS hhhh	The address of a record was too far from the address of the previously-processed record. This is an internal limitation of LOAD, but it can be circumvented. Use DDT to read the hexfile into memory, then use a SAVE command to store the memory image file on disk.
ERROR: NO MORE DIRECTORY SPACE, LOAD ADDRESS hhhh	Disk directory is full.
Error on line nnn message	The SUBMIT program displays its messages in the format shown above, where nnn represents the line number of the SUBMIT file. Refer to the message following the line number.
FILE ERROR	Disk or directory is full, and ED cannot write anything more on the disk. This is a fatal error, so make sure there is enough space on the disk to hold a second copy of the file before invoking ED.
FILE EXISTS	You have asked CP/M to create or rename a file using a file specification that is already assigned to another file. Either delete the existing file or use another file specification.
	The new name specified is the name of a file that already exist. You cannot rename a file with the name of an existing file. If you want to replace an existing file with a newer version of the same file, either rename or erase the existing file, or use the PIP utility.

Message	Meaning
File exists, erase it	The destination filename already exists when you are placing the destination file on a different disk than the source. It should be erased or another disk selected to receive the output file.
FILE IS READ/ONLY	The file specified in the command to invoke ED has the Read Only attribute. ED can read the file so that the user can examine it, but ED cannot change a Read Only file.
File Not Found	CP/M cannot find the specified file. Check that you have entered the correct drive specification or that you have the correct disk in the drive. ED cannot find the specified file. Check that you have entered the correct drive specification or that you have the correct disk in the drive. STAT cannot find the specified file. The message might appear if you omit the drive specification. Check to see if the correct disk is in the drive.
FILE NOT FOUND-{filespec}	An input file that you have specified does not exist.
Filename required	You typed the ED command without a filename. Reenter the ED command followed by the name of the file you want to edit or create.
hhhh?? = dd	The ?? indicates DDT does not know how to represent the hexadecimal value dd encountered at address hhhh in 8080 assembly language. dd is not an 8080 machine instruction opcode.
Insufficient memory	There is not enough memory to load the file specified in an R or E command.
Invalid Assignment	You specified an invalid drive or file assignment, or misspelled a device name. This error message might be followed by a list of the valid file assignments that can follow a filename. If an invalid drive assignment was attempted the message "Use: d: = RO" is displayed, showing the proper syntax for drive assignments.
Invalid control character	The only valid control characters in the SUBMIT files of type SUB are ^A through ^Z. Note that in a SUBMIT file the control character is represented by typing the cir-

Message	Meaning
	cumflex, ^, not by pressing the control key.
INVALID DIGIT-{filespec}	An invalid hex digit has been encountered while reading a hex file. The hex file with the invalid hex digit should be corrected, probably by recreating the hex file.
Invalid Disk Assignment	Might appear if you follow the drive specification with anything except = R/O.
INVALID DISK SELECT	CP/M received a command line specifying a nonexistent drive, or the disk in the drive is improperly formatted. CP/M terminates the current program as soom as you press any key.
INVALID DRIVE NAME (Use A, B, C, or D)	SYSGEN recognizes only drives A, B, C and D as valid destinations for system generation.
Invalid File Indicator	Appears if you do not specify RO, RW, DIR, or SYS.
INVALID FORMAT	The format of your PIP command is illegal. See the description of the PIP command.
INVALID HEX DIGIT LOAD ADDRESS hhhh ERROR ADDRESS hhhh BYTES READ: hhhh	File contains incorrect hex digit.
INVALID MEMORY SIZE	Specify a value less than 64K or your computer's actual memory size.
INVALID SEPARATOR	You have placed an invalid character for a separator between two input filenames.
INVALID USER NUMBER	You have specified a user number greater than 15. User numbers are in the range 0 to 15.
n?	You specified a number greater than fifteen for a user area number. For example, if you type USER [←], the screen displays 18?.
NO DIRECTORY SPACE	The disk directory is full. Erase some files to make room for PRN and HEX files. The directory can usually hold only 64 filenames.
NO DIRECTORY SPACE-{filespec}	There is not enough directory space for the output file. You should either erase some unnecessary files or get

Message	Meaning
	another disk with more directory space and execute PIP again.
NO FILE-{filespec}	CP/M cannot find the specified file, or no files exist.
	The indicated source or include file cannot be found on the indicated drive.
	The file specified in an R or E command cannot be found on the disk.
NO INPUT FILE PRESENT ON DISK	The file you requested does not exist.
No memory	There is not enough (buffer?) memory available for loading the program specified.
NO SOURCE FILE ON DISK	SYSGEN cannot find CP/M either in CPMxx.com form or on the system tracks of the source disk.
NO SOURCE FILE PRESENT	The assembler cannot find the file you specified. Either you mistyped the filespecification in you command line, or the file is not type ASM.
NO SPACE	Too many files are already on the disk, or no room is left on the disk to save the information.
No SUB file present	For SUBMIT to operate properly, you must create a file with filetype of SUB. The SUB file contains usual CP/M commands. Use one command per line.
NOT A CHARACTER SOURCE	The source specified in your PIP commands is illegal. You have probably specified an output device as a source.
NOT DELETED	PIP did not delete the file, which may have had the R/O attribute.
NOT FOUND	PIP cannot find the specified file.
OUTPUT FILE WRITE ERROR	You specified a write-protected diskette as the destination for the PRN and HEX files, or the diskette has no space left. Correct the program before assembling your program.
Parameter error	Within the SUBMIT file of type sub, valid parameters are $0 through $9.

Message	Meaning
PARAMETER ERROR, TYPE RETURN TO IGNORE	If you press return, SYSGEN proceeds without processing the invalid parameter.
QUIT NOT FOUND	The string argument to a Q parameter was not found in you input file.
Read error	An error occurred when reading the file specified in the type command. Check the disk and try again. The STAT filespec command can diagnose trouble.
READER STOPPING	Reader operation interrupted.
Record Too Long	PIP cannot process a record longer than 128 bytes.
START NOT FOUND	The string argument to an S parameter cannot be found in the source file.
SOURCE FILE INCOMPLETE	SYSGEN cannot use your CP/M source file.
SOURCE FILE NAME ERROR	When you assemble a file, you cannot use the wildcard characters * and ? in the filename. Only one file can be assembled at a time.
SOURCE FILE READ ERROR	The assembler cannot understand the information in the file containing the assembly language program Portions of another file might have been written over your assembly language file, or information was not properly saved on the diskette. Use the TYPE command to locate the error. Assembly language files contain the letters, symbols, and numbers that appear on your keyboard. If your screen displays unrecognizable output or behaves strangely, you have found where computer instructions have crept into your file.
SYNCHRONIZATION ERROR	The MOVCPM utility is being used with the wrong CP/M system.
"SYSTEM" FILE NOT ACCESSIBLE	You tried to access a file set to SYS with the STAT command.
TOO MANY FILES	There is not enough memory for STAT to sort the files specified, or more than 512 files were specified.
UNEXPECTED END OF HEX FILE-{filespec}	An end-of-file was encountered prior to a termination hex record. The hex file without a termination record should be corrected, probably by recreating the hex file.

Message	Meaning
Unrecognized Destination	Check command line for valid destination.
Use: STAT d: = RO	An invalid STAT drive command was given. The only valid drive assignment in STAT is STAT d: = RO.
VERIFY ERROR: -{filespec}	When copying with the V option, PIP found a difference when rereading the data just written and comparing it to the data in its memory buffer. Usually this indicates a failure of either the destination disk or drive.
XSUB ACTIVE	XSUB has been invoked.
XSUB ALREADY PRESENT	XSUB is already active in memory.
Your input?	If CP/M cannot find the command you specified, it returns the command name you entered followed by a question mark. Check that you have typed the command line correctly, or that the command you requested exists as a .COM file on the default or specified disk.

GLOSSARY

acoustic coupler

A device which converts digital electrical signals into acoustic signals for transmission over a telephone line, and which converts acoustic signals received over a telephone line into digital signals in a form which can be input by a computer.

address

A number which is used during data input/output to indicate a location in memory, an input/output port, or a position in an auxiliary storage device. Ordinarily, each byte in memory is indicated by a unique address. There are a variety of addressing schemes, including absolute addresses, relative addresses, and base addresses.

application program

A program which is used to do work for the user. Contrast with system program. Application programs may be prepared either by software manufacturers or the user himself, and may be written in any language supported by the computer. auxiliary storage - Devices which are used to supplement the computer's main storage, such as floppy disk units, ROM capsules, and magnetic tape.

ASCII code

An abbreviation for "American National Standard Code for Information Interchange." Uses a data word consisting of 7 bits (8 bits including parity) to represent the letters of the alphabet, numerals, and symbols.

bank

In this book, a 64K byte segment of main memory.