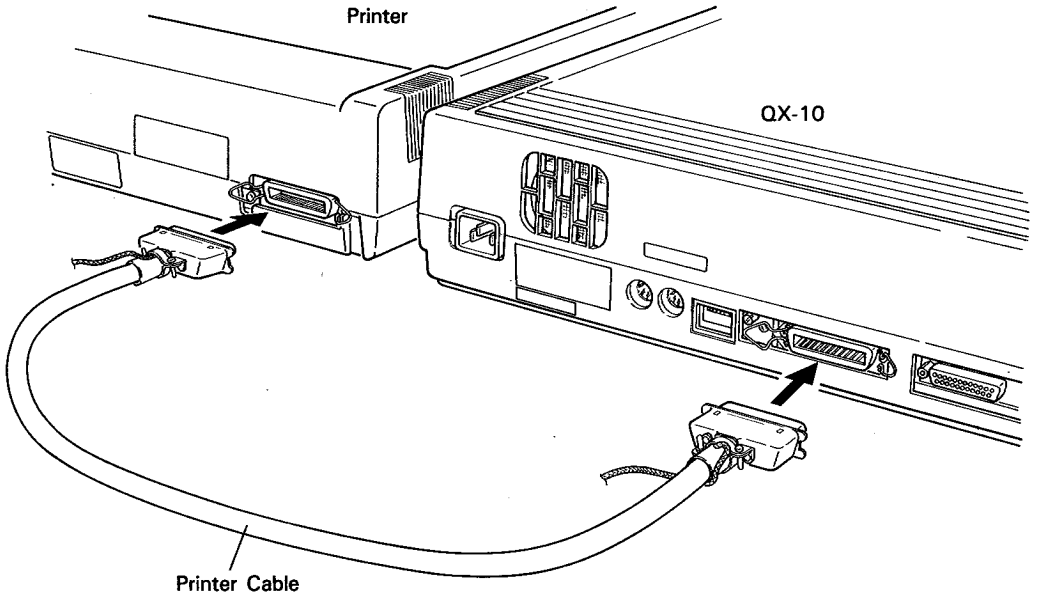
Chapter 6 PERIPHERAL INTERFACES AND OPTIONS

Standard interfaces built into the QX-10 include the printer interface, the light pen interface, the keyboard inteface, and a serial communication interface. Models featuring a color display monitor also include a color display interface. Various options can be added to the system for expanded system configuration. This chapter describes the functions of these interfaces and options, and the manner in which they are used.

6.1 Printer Interface

The printer interface built into the QX-10 allows the system to be connected to any Centronics-standard printer through the standard 36-pin Centronics connector provided on the back panel.



Pin assignments of the printer connector are as follows.

Pin No.	Signal Symbol	Signal Direction	Description of Signal
1	$\overline{STB}$	OUT	Strobe
2	DB0	OUT	Data line 0
3	DB1	OUT	Data line 1
4	DB2	OUT	Data line 2
5	DB3	OUT	Data line 3
6	DB4	OUT	Data line 4
7	DB5	OUT	Data line 5
8	DB6	OUT	Data line 6
9	DB7	OUT	Data line 7
10	$\overline{ACK}$	IN	Acknowledge
11	$\overline{RDY}$	IN	Ready
12	NPA	IN	No paper
13	$\overline{SLO}$	IN	Select out
14	$\overline{ALF}$	OUT	Auto line feed
15	—	—	NC (unused)
16	GL	—	Signal ground
17	FG	—	Frame ground
18	—	—	NC (unused)
19-30	GL	—	Signal ground
31	$\overline{RST}$	OUT	Reset
32	$\overline{ERR}$	IN	Error
33	GL	—	Signal ground
34	—	—	NC (unused)
35	PWF	IN	Power failure
36	—	—	NC (unused)

Note: The direction of signal is as viewed from the Q10SYM board.

The meanings of the various signals are as follows.

$\overline{STB}$
The strobe signal used by the printer to read data. Active low.

DB0 through DB7
Parallel data signals. Active high.

$\overline{ACK}$
This signal indicates that data has been received by the printer and that the printer is ready to receive more data. Active low.

$\overline{\text{RDY}}$
This signal indicates that the printer is ready to receive data. Active low.

NPA
When high, this signal indicates that the printer is out of paper.

$\overline{\text{SLO}}$
When high, this signal indicates that the printer is in the selected state.

$\overline{\text{ALF}}$
This signal directs the printer to automatically make a line feed after each line has been printed. Active low.

$\overline{\text{RST}}$
This signal initializes the printer. Active low.

$\overline{\text{ERR}}$
This signal indicates that a printer error has occurred. Active low.

$\overline{\text{PWF}}$
This signal indicates that the printer power is off. Active low.

It should be noted that some printers which are advertised as Centronics-compatible do not use the full Centronics communication protocol; such printers cannot be used with the QX-10. Printers supported by the QX-10 include all printers of the EPSON MX and FX series and the RX-80 shown in "6.7 Printers." Refer to the individual printer manual or consult your EPSON dealer for details. See Appendix D for control of printers.

NOTE:
All EPSON MX/FX series printers (except the MX-80) and the RX-80 are capable of printing multiple font characters and international character sets.

6.2 Light Pen Interface

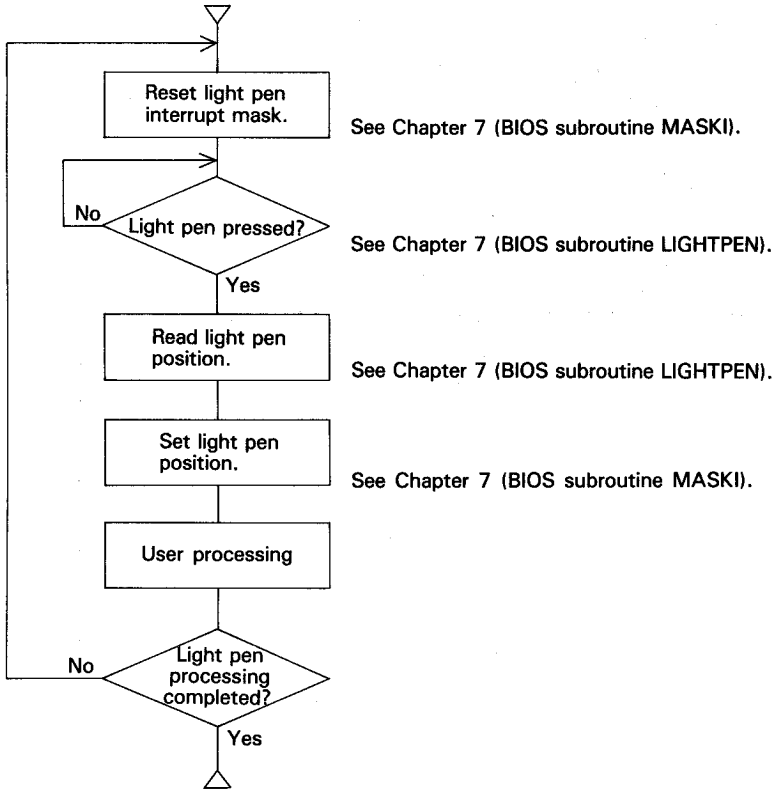
The light pen interface built into the QX-10 allows the optional light pen to be used. The light pen is connected through the DIN connector provided on the back panel. Pin assignements of this connector are as follows.

Pin No.	Signal Symbol	Signal Direction	Description of Signal
1	+5V	—	+5V
2	SIG	IN	Light pen signal
3	+12V	—	+12V
4	SW	IN	Light pen switch
5	GND	—	Ground
E	FG	—	Frame ground

Note: The direction of signal is as viewed from the Q10SYM board.

The screen position at which the light pen is pressed can be obtained by calling BIOS subroutines.

Sample flowchart



6.3 Serial Communication Interface

The built-in serial communication interface supports both synchronous and RS-232C asynchronous communication protocol, with the mode of operation selected by changing the setting of a jumper cable inside the main unit. Synchronous communication (either SDLC or BYSYNC) capability is provided to allow the QX-10 to be used as an intelligent terminal with large computer systems; asynchronous protocol is that which is used for communication with devices using RS-232C protocol, such as serial printers, acoustic couplers, or other QX-10 computers. The jumper is set for asynchronous communication at the time of shipment; therefore this will be referred to hereafter simply as the RS-232C interface. Consult your dealer concerning use of the synchronous serial interface.

See the discussion of the CONFIG and TERM commands for procedures for changing the RS-232C configuration and using the interface under CP/M; procedures for using asynchronous communications protocol with BASIC programs are discussed in the **QX-10 MFBASIC Reference Manual.**

Use of optional RS-232C interface boards allows the system to be connected to up to 5 serial communication lines. An optional RS-232C card can be inserted in any of the five option slots.

6.3.1. Communication speed

One standard RS-232C interface is built into the main unit of the QX-10 and an additional four interfaces can be installed in the option slots for a total of five. The maximum communication speed of interfaces mounted in the option slots is different from that built into the main unit. The main unit RS-232C interface provides a maximum communication speed of 9600 bps; however, that of interfaces mounted in the option slots is 19,200 bps. Further, each of the interfaces can be set to a different communication speed.

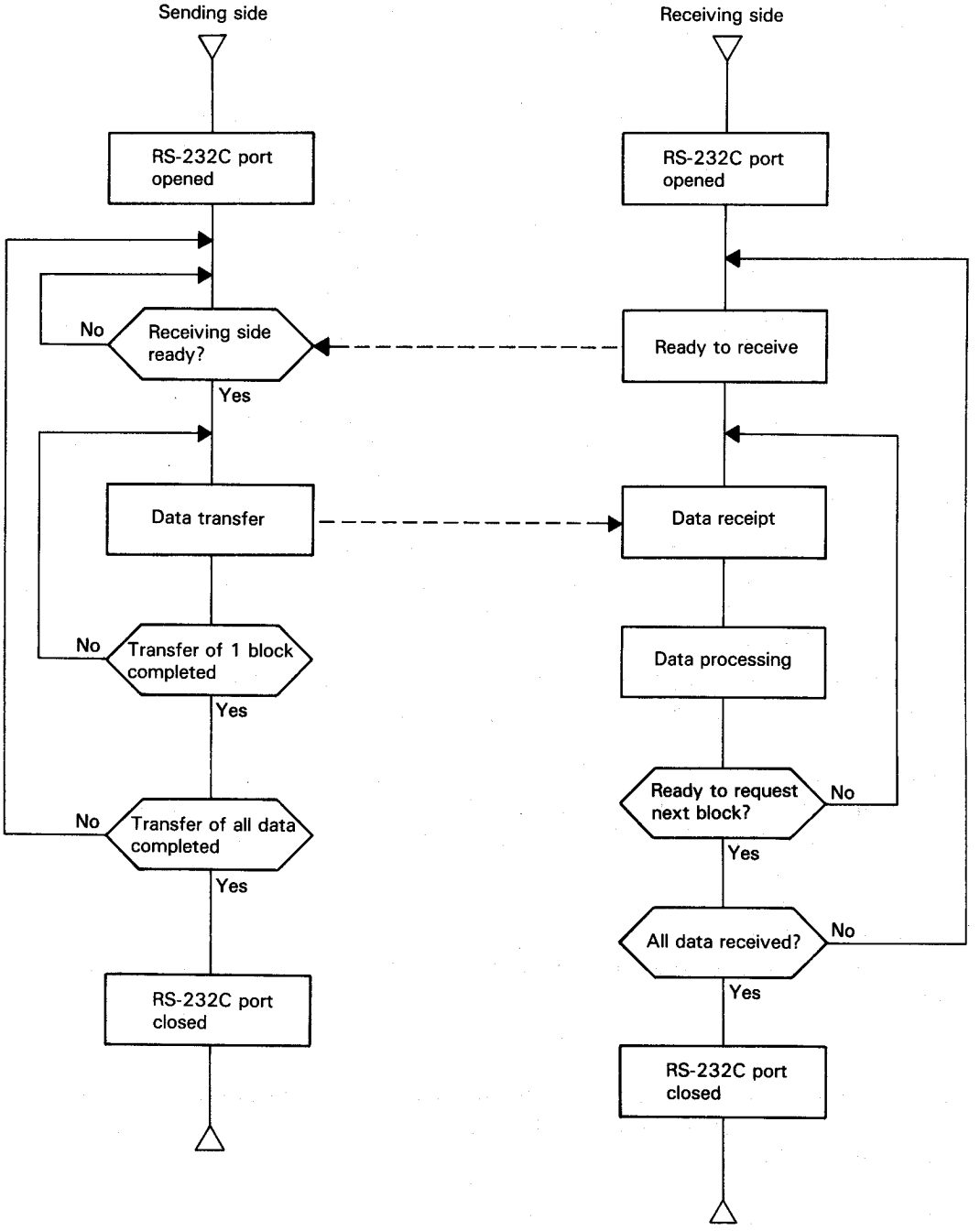
This makes it possible to perform low speed collection processing of data, then to transfer the data at high speed to another device. (There is a high probability that a receive buffer overflow will occur if collection processing of data is performed at a speed greater than or equal to the speed at which collected data is transferred.)

6.3.2. Data reception

The receive buffer may overflow during receipt of data; this is particularly likely to occur when data is being transferred at high speed, but receive data processing is performed slowly. (This will not result in a problem if the receive buffer is larger than the number of data bytes received.) Use the following measures to avoid receive buffer overflows.

Prepare as large a buffer as possible on the receiving QX-10 side. On the send side, the size of one block of data should be set equal to that of the buffer on the receive side, then the sending side waits for the receiving side to become ready. When the receiving side is ready, an entire block is sent without pause. After transmission of each block has been completed, the sending side waits for the receiving side to become ready again, then sends the next block. This is repeated until transmission of all data has been completed. The receiving side takes the data out of the receive buffer and processes it as each block is received. The number of bytes of data processed is counted and, when the number of bytes remaining to be processed has been reduced to a certain level, the receiving side requests transmission of the next block from the sending side. The timing at which this request is sent varies according to the speed of data communication and the speed with which received data is processed; the higher the communication speed and the lower the processing speed, the fewer the number of unprocessed bytes at which the request for transmission is sent. This cycle is repeated until all data has been processed.

Receive buffer overflows are unlikely if the communication speed is set to 1200 bps or less and no significant data processing time is involved. When other interrupt processing or processing with all interrupts disabled is required during data transfer, it is best to have the receiving side send the request for transmission of the next block only after processing has been completed for the entire block received previously.



6.3.3 Standard RS-232C interface

The standard RS-232C interface is connected to external devices through a standard DB-25 connector. Pin assignments of this connector are as follows.

Pin No.	Signal Symbol	Signal Direction	Description of Signal
1	FG	–	Frame ground
2	TxD	OUT	Transmitted data
3	RxD	IN	Receive data
4	RTS	OUT	Request to send
5	CTS	IN	Clear to send
6	DSR	IN	Data set ready
7	SG	–	Signal ground
8	CD	IN	Carrier detect
20	DTR	OUT	Data terminal ready

Note: The direction of signal is as viewed from the Q10SYM board.

Pins other than those indicated in the table are not used.
The meanings of the various signals are as follows.

FG (Frame ground)
This terminal is connected to the chassis of the QX-10; ordinarily, it is also connected via the external cable to the corresponding terminal on the other device.

TxD (Send data)
TxD is the signal used from transmitting data from the QX-10 to the device (acoustic coupler, etc.) with which the QX-10 is connected. This is possible when the Clear to send signal is on.

RxD (Receive data)
RxD is the data signal from the acoustic coupler or other RS-232C compatible device to the QX-10.

RTS (Request to send)
RTS is the signal which controls the communication function of the device (acoustic coupler, etc.) connected to the QX-10. The connected device becomes ready to send when this signal is ON.

CTS (Clear to send)
CTS is the signal which indicates whether the connected device is ready to accept data transmissions. Transmission is enabled when this signal is ON and disabled when it is OFF.

DSR (Data set ready)
DSR is the signal which indicates whether the connected device is ready for operation. When this signal is ON, the applicable device is connected to the interface cable and is ready to accept data transmission/reception control signals.

SG (Signal ground)
The SG terminal provides a electrical reference potential which is common to all signal lines. It is connected to the corresponding terminal on the connected device.

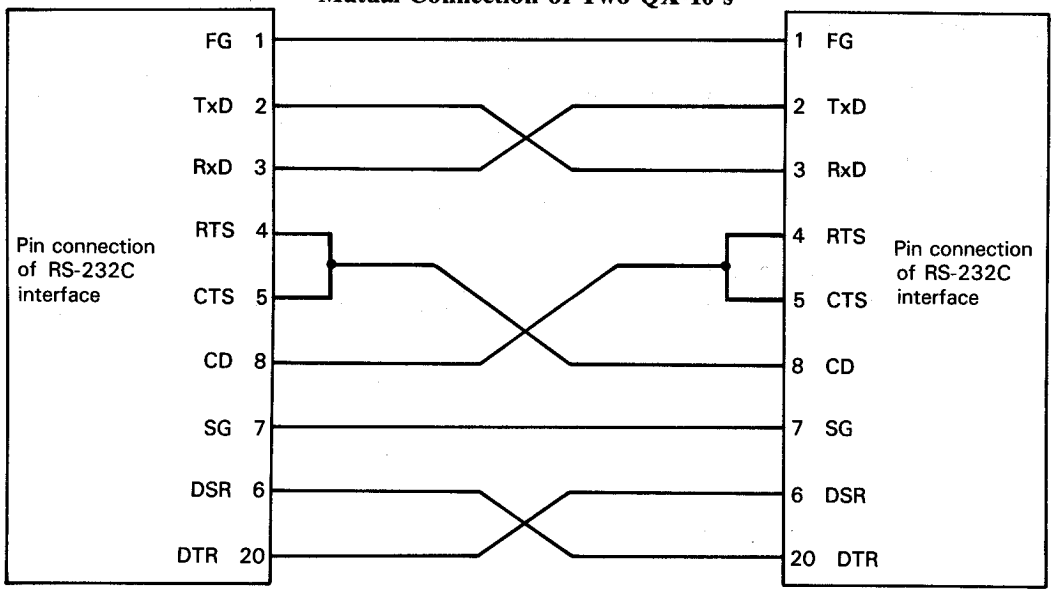
CD (Carrier detect)
The CD terminal is used for detecting the carrier signal from the connected device.

DTR (Data terminal ready)
DTR is the signal output by the QX-10 to the connected device to indicate that it is ready to receive data.

6.3.4 Example of connection of the RS-232C interface

An example of connection of two QX-10's through the RS-232C interface is given below.

Mutual Connection of Two QX-10's



6.3.5 Electrical characteristics of RS-232C signals

Two types of signals are used in RS-232C communications; these are referred to as control signals and data signals. The control signals are DTR, RTS, CTS, DSR, and CD. The data signals are TxD and RxD. The voltage levels of these signals are as follows.

(1) Control signals
ON: +5V to +15V
OFF: −5V to −15V

(2) Data signals
Mark (logical 1): −5V to −15V
Space (logical 0): +5V to +15V

6.3.6 Programming for the RS-232C interface

The RS-232C interface is supported by all versions of CP/M and BASIC supplied by EPSON for the QX-10. With CP/M, the CONFIG and TERM commands make it possible to configure the interface for various communication formats. See the discussions of these commands for details.

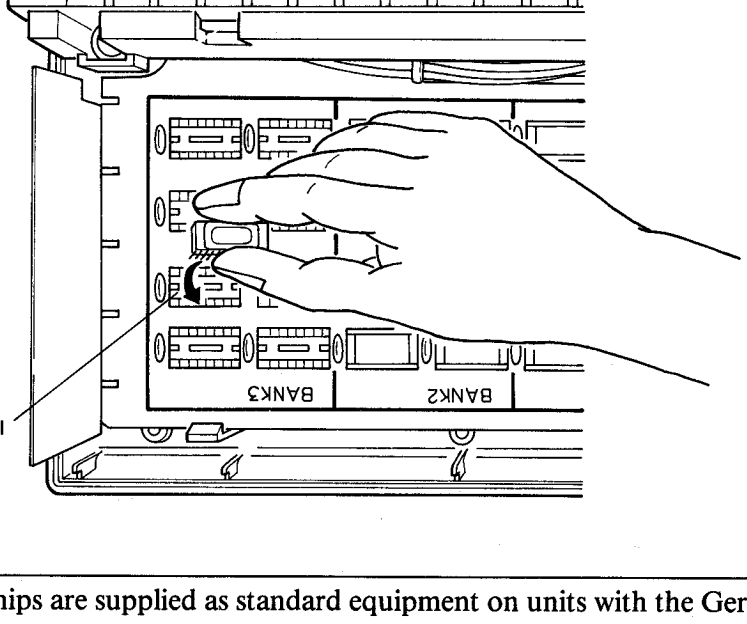
MFBASIC provide a variety of commands which make it easy to communicate through the RS-232C interface. For details, see Chapter 5 of the **QX-10 MFBASIC Reference Manual.**

6.4 Color Display Interface

The color display interface is an optional circuit board which makes it possible to use the QX-10 with the EPSON high-resolution color display monitor. (Note that installation of this interface by anyone other than an authorized EPSON dealer will void your warranty.) The board provides a 640 – 400 dot display in eight colors, and allows graphics and color to be controlled on a pixel basis. Since this interface uses the same graphic display controller as the interface for the monochrome display, all graphic control functions (such as zoom display, circle drawing, and point-to-point interpolation) are supported.

6.5 Optional RAM Chips

Eight optional RAM chips are inserted in the IC sockets for user memory bank #3 in the option slot compartment to expand the memory size to 256K bytes. The location of the IC sockets is shown below.
Installing the RAM chips makes it possible to use 56K virtual disk drive E (disk image RAM).



IC sockets for optional 64KB RAM

These RAM chips are supplied as standard equipment on units with the German keyboard.

6.6 MultiFont Character Generator ROM Card

This card includes six 128K bits mask ROMs which generate fonts consisting of 14×17 dots. The multifont characters include information necessary for proportional spacing.

This card is provided with the QX-10. The MFONT command and some statements of MFBASIC cannot be used when this card is not installed. This card is installed in option slot 5 when the QX-10 is shipped from the factory. However, it can be inserted into any of the five option slots.
See Appendix I.

6.7 Printers

Eleven types of printers can be used with QX-10s which use the MultiFont version of CP/M. These are as listed below.

EPSON	MX-80
	MX-80/II
	MX-80/III
	MX-82
	MX-82/III
	MX-100
	MX-100/III
	FX-80
	FX-100
	RX-80

(1) EPSON MX-80
With this printer, a total of four different character sets can be selected by DIP switch; these are as listed below.

US ASCII
France
England
Germany

This printer cannot prints MultiFont characters and user defined characters since it does not support bit image printing.

Character sets are as specified in the specifications of individual printers.

Proportional printing is not possible.

(2) EPSON MX-80/82/100 (types II and III), FX-80/100, and RX-80
BIOS accomodates these printers by sending the following escape sequence to the printer when it is entered from the console, when transient command CONFIG is executed, or when the OPTION COUNTRY command is executed from MFBASIC.

```
ESC ''R'' CHR$ (n)
```

These printers can be used to print up to 16 different styles of Multifont characters by entering the appropriate escape sequence from the console, or by means of the OPTION STYLE command or STYLE$ function of MFBASIC. They can also be used to print user defined characters.

• Bit image printing

(i) In the BIT ON mode, character sets are as indicated in the specifications of individual printers.

(ii) In the BIT OFF mode, the bit image printing function of the printer is used to print graphic characters (7FH to 9FH) and, in the MFBASIC mode, 2-byte characters.
Therefore, all control characters other than the following are invalid.

BEL (07H)
LF (0AH)
FF (0CH)
CR (0DH)
ESC (1BH)

Further, escape sequences are limited to those indicated in this publication; those built into the printer itself cannot be used.

• Proportional printing

(i) Proportional printing is not possible in the BIT ON mode.

(ii) These printers contain proportional spacing information which determines the print width for multiple font characters and other 2-byte characters. Characters with widths of from 4 to 17 dots are printed in accordance with this information. It is also possible to disregard the proportional spacing and print characters at fixed spacings (see Appendix G.1). Further, the spacing between characters can be varied and spaces of any desired width can be printed (see Appendix G.3).

Chapter 7 CP/M SYSTEM INTERFACE

As was described in Chapter 4, the CP/M operating system consists of CCP, BDOS and BIOS. BDOS and BIOS include many subroutines which handle peripheral devices and files, and which can be called by user programs. This chapter describes how to use these subroutines.
Familiarity with assembler programming is a prerequisite for using these subroutines and functions.

7.1 IOBYTE and File Control Block (FCB)

The CP/M operating system supports 4 logical I/0 devices, CON:, RDR:, PUN: and LST:. These names are assigned to physical devices according to the contents of a logical-to-physical assignment table called IOBYTE. IOBYTE is located at address 0003H in main memory. The contents of IOBYTE can be read and changed with the STAT command (STAT DEV:).

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

Relationship between physical devices and logical devices

Logical device Bit position	LST: 7 6	PUN: 5 4	RDR: 3 2	CON: 1 0
Physical device	0 0 none	0 0 printer	0 0 keyboard	0 0 keyboard (O) keyboard (I)
	0 1 CRT	0 1 none	0 1 none	0 1 CRT (O) keyboard (I)
	1 0 printer	1 0 RS-232C	1 0 RS-232C	1 0 printer (O) keyboard (I)
	1 1 RS-232C	1 1 none	1 1 none	1 1 RS-232C (O) RS-232C (I)

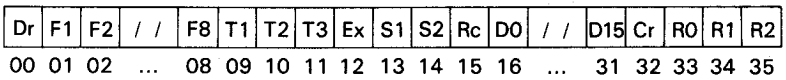
The initial setting of IOBYTE is 10101001, that is LST: corresponds to the printer, PUN: to the RS-232C port, RDR: to the RS-232C port and CON: to both the CRT display (output) and keyboard (input).

(Reference)

The IOBYTE assignments are listed as follows when STAT VAL: is executed. Note that the physical device names indicated do not represent the names of actual physical devices.

	LST:	PUN:	RDR:	CON:
0	TTY:	TTY:	TTY:	TTY:
1	CRT:	PTP:	PTR:	CRT:
2	LPT:	UP1:	UR1:	BAT:
3	UL1:	UP2:	UR2:	UC1:

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



DR:	Drive code
0:	Logged-in drive
1:	Drive A
2:	Drive B
5:	Drive E
6:	Drive F
F1 - F8:	File name in upper case ASCII
T1 - T3:	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 CP/M.
Cr:	Record counter
R0 - R2:	Random access record counter

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

7.2 BDOS Function Call

BDOS includes many utility subroutines which input or output data from/to peripherals and handle files. These subroutines are referred to as BDOS functions, and can easily be used by user programs as shown below.

There are 39 BDOS functions which can be used by user programs; 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.

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 01H — 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	

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

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 getting the current user number	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

For an example of BDOS function call, see Appendix E.

7.3 BIOS Interface

The basic input/output system of CP/M for the QX-10 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.

7.3.1. Entry Address

The entry addresses of BIOS subroutines are listed below.

BIOS ENTRY

ADDRESS	ENTRY NAME
F600	BOOT
F603	WBOOT
F606	CONST
F609	CONIN
F60C	CONOUT
F60F	LIST
F612	PUNCH
F615	READER
F618	HOME
F61B	SELDSK
F61E	SETTRK
F621	SETSEC
F624	SETDMA
F627	READ
F62A	WRITE
F62D	LISTST
F630	SECTRN
F633	PSET
F636	HCOPY
F639	BEEP
F63C	RSOPEN
F63F	RSCLOSE
F642	RSINST
F645	RSOUTST
F648	RSIN
F64B	RSOUT
F64E	TIMDAT
F651	MEMORY
F654	RSIOX
F657	LIGHTPEN
F65A	MASKI
F65D	LOADX
F660	STORX
F663	LDIRX
F666	JUMPX
F669	CALLX
F66C	GETPFK
F66F	PUTPFK

7.3.2 Functions of BIOS subroutines

In the following explanations, entry parameters are those which must be assigned by user programs calling the BIOS routines.

BOOT F600

BOOT is the entry point for the cold start loader, which runs only when the power is turned on and is not used by the user.

WBOOT F603

WBOOT is the entry point for the warm start bootstrap; this routine loads CCP and BDOS into memory and initializes the system.

CONST F606

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

Result

A: 0	Indicates that the console input buffer is empty.
A: FFH	Indicates that the console input buffer contains a character.

CONIN F609

CONIN is the entry point for the subroutine which sets a character input from the console into register A.

Result

A: Input character

CONOUT F60C

CONOUT is the entry point for the subroutine which outputs a character to the console from register C.

Entry parameter

C: Character to be output to the console

LIST F60F

LIST is the entry point for the subroutine which outputs characters to LST: from register C.

Entry parameter

C: Character to be output to LST:

PUNCH F612

PUNCH is the entry point for the subroutine which outputs a character to PUN: from register C.

Entry parameter

C: Character to be output to PUN:

READER F615

READER is the entry point for the subroutine which read a character from RDR: and sets it in register A.

Result A:

Character input from RDR:

HOME F618

HOME is the entry point for the subroutine which sets the C register to 0 and jumps to SETTRK. This routine is used to set the head to track 0.

SELDSK F61B

SELDSK is the entry point for the subroutine which selects a disk drive in accordance with the entry parameter stored in register C.

Entry parameter

C: 0 — Drive A
C: 1 — Drive B
C: 4 — Drive E (disk image RAM)
C: 5 — Drive F (disk image RAM)

SETTRK F61E

SETTRK is the entry point for the subroutine which selects the track to be accessed.

Entry parameter

BC: 0 - 27 (For drives A and B)
BC: 0 - 6 (For disk image RAM)

SETSEC F621

SETSEC is the entry point for the subroutine which sets the sector number to be accessed.

Entry parameter

C: Sector number to be accessed (0 - 3FH)

SETDMA F624

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

BC: DMA address

READ F627

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.

Result

A: 0 — Normal completion
A: Other than 0 — Abnormal completion

WRITE F62A

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

C: 0 — Write standard format data.
C: 1 — Write unblocked data.
C: 2 — Write sequential file.

Result

A: 0 — Normal completion
A: Other than 0 — Abnormal completion

LISTST F62D

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

Result

A: 0 — Printer not ready
A: FFH — Printer ready

SECTRN F630

SECTRN is the entry point for the subroutine which converts a logical sector number into the corresponding physical sector number.
The entry parameter is stored in register pair BC and the result is returned to register pair HL. With the QX-10, the logical sector number equals the physical sector number, and therefore this entry point is not required.

PSET F633

PSET is the entry point for the subroutine which reads data at the location on the CRT screen specified by the contents of register pair HL, performs the operation (AND, OR, or XOR) specified in register C on the data read and the contents of register B, then sets the result at the original location in the color specified by the contents of register E.

Entry parameters

HL: CRT screen address (0 - 31999)
B: Data
C: 1 = AND, 2 = OR, 3 = XOR
E: Color

0	1	2	3	4	5	6	7
Black	Blue	Red	Violet	Green	Light blue	Yellow	White

Result

B: Same as the input data
C: Resultant data (red)
D: Resultant data (green)
E: Resultant data (blue)
HL: Same as the input data

NOTE:
With the green CRT, the contents of register pair DE may be destroyed.

HCOPY F636

HCOPY is the entry point for the subroutine which prints a hard copy of the CRT screen image on the printer. This subroutine is effective with the combinations of mode, CRT and printer indicated by the circles in the table below.

CRT / Mode / Printer	Green CRT				Color CRT			
	I	II	III	IV	I	II	III	IV
MX-80	○	×	×	×	×	×	×	×
TYPE 2, MX-82 or MX-100	○	○	○	○	○	○	○	○
TYPE 3, MX-82 or MX-100	○	○	○	○	○	○	○	○
FX-80	○	○	○	○	○	○	○	○
FX-100	○	○	○	○	○	○	○	○
RX-80	○	○	○	○	○	○	○	○

○...Hard copy
×...No operation

I: Non-MFBASIC Normal mode
II: Non-MFBASIC MF mode
III: MFBASIC width 80 mode
IV: MFBASIC width 40 mode

With a color CRT, colors other than the background color are printed out.

BEEP F639

BEEP is the entry point for the subroutine which sounds the keyboard buzzer.

Entry parameter

C: 0 = OFF
FFH = ON
1 to FEH = Buzzer sounds for the specified interval (in 10 msec units)

Result

This subroutine changes the contents of all registers.

RSOPEN F63C

RSOPEN is the entry point for the subroutine which initializes the RS-232C interface for communication according to the condition set with the CONFIG command. This routine changes the contents of all registers.

RSCLOSE F63F

RSCLOSE is the entry point for the subroutine which disables communication through the RS-232C interface. This routine changes the contents of all registers.

RSINST F642

RSINST is the entry point for the subroutine which checks whether a character has been received in the RS-232C interface's receive buffer.

Result

A: 00H Receive buffer is empty.
A: FFH A character is received in the receive buffer.

RSOUTST F645

RSOUTST is the entry point for the subroutine which checks whether the RS-232C interface is ready to send.

Result

A: 00H The RS-232C interface is not ready to send.
A: FFH The RS-232C interface is ready to send.

RSIN F648

RSIN is the entry point for the subroutine which reads in a character from the RS-232C interface.

Result

A: Character received
This subroutine changes the contents of the other registers.

RSOUT F64B

RSOUT is the entry point for the subroutine which sends a character to the RS-232C port.

Entry parameter

C: Character to be sent

This subroutine changes the contents of all registers.

NOTE:
RSOPEN must be called to open the RS-232C port before RSIN or RSOUT is called. Characters which overflow the receive buffer are lost.

TIMDAT F64E

TIMDAT is the entry point for the subroutine which resets or reads the clock (time and date). This routine is called every time warm boot is performed.

Entry parameter

C: 0FFH — Sets the clock with data stored in addresses 0FEF8 through 0FEFFH.

C: 0 — Reads the clock and stores the data read in addresses 0FEF8 through 0FEFFH.

Result

This subroutine changes the contents of all registers.

Memory contents

All data is coded in BCD code.

Address		Data (BCD)
0FEF8H	Year	(19)XX
F9H	Month	01 - 12
FAH	Day	01 - 31
FBH	Hour	00 - 23
FCH	Min.	00 - 59
FDH	Sec.	00 - 59
FEH	Day of week	00 - 06
FFH	Not used	00

F8	F9	FA	FB	FC	FD	FE	FF
Year	Month	Day	Hour	Min.	Sec.	Day of week	Not used

MEMORY F651

MEMORY is the entry point for the subroutine which switches the memory bank (56KB from 0 to ODFFFH) excluding the common area.

Entry parameter

C: 0 Selects the main memory bank.
C: 1 Selects user memory bank #1.
C: 2 Selects user memory bank #2.
C: -1 Selects the system memory bank.

Result

A: 0 Normal completion
A: Other than 0 RAM is not installed.
C: Bank number selected previously (-1 to 2)
The contents of other registers are not changed.

NOTE:
The main memory bank is always selected after cold start or warm boot.

RSIOX F654

RSIOX is the entry point for the subroutine which initializes a RS-232C port in an option slot, disables interrupt from it, inputs and outputs data throught it, and checks its status.
This routine can be used for communications through the RS-232C port on the main board when the required parameters are set.
The functions of this subroutine are explained below.

1. Opening the RS-232C port (OPEN)

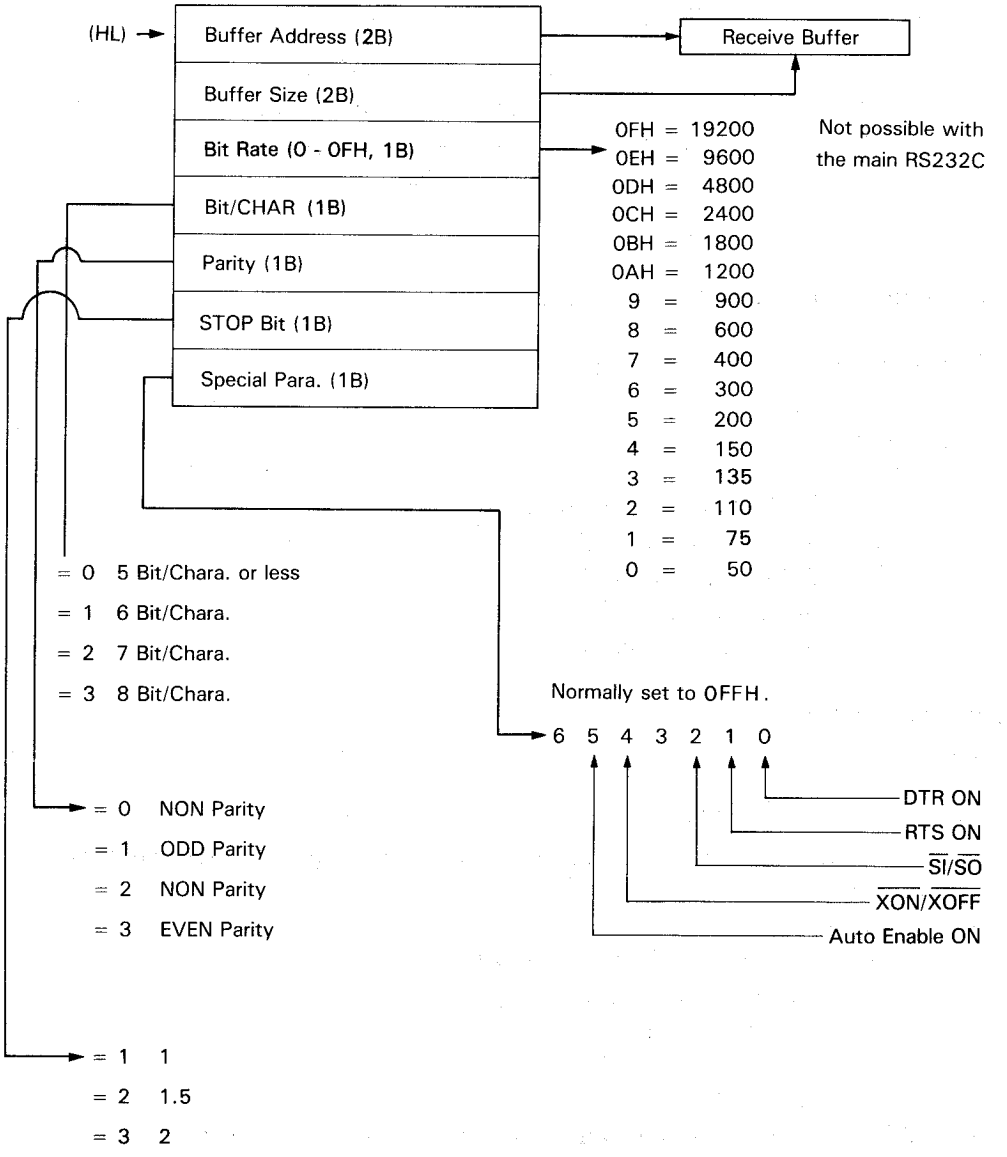
Entry parameters

B: 1XH

X = 0 Main board RS-232C port
1 Option 1, channel A
2 Option 1, channel B
3 Option 2, channel A
4 Option 2, channel B

HL: Parameter block address

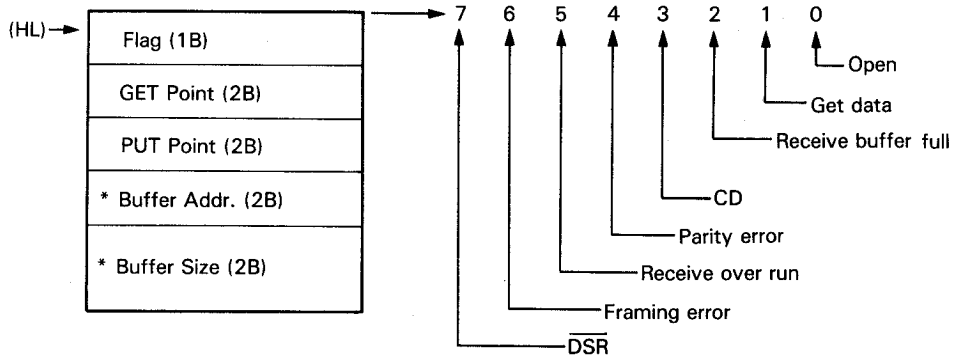
The contents of the parameter block is shown below.



Return parameters

A: 0	Normal open	Z flag = 1
1	No option board	= 0
2	Busy (i.e., used by another program)	= 0
4	Parameter error	= 0
HL:	Not changed	

The contents of the parameter block is as follows.



*: The same as the value set by the user.

2. Closing the RS-232C port (CLOSE)

Entry parameter

B: 2XH

X = 0	RS-232C on the main board
1	Option 1, channel A
2	Option 1, channel B
3	Option 2, channel A
4	Option 2, channel B

Return parameter

None.

3. Checking status (INSTS)

The routine checks whether a character has been received in the receive buffer.

Entry parameters

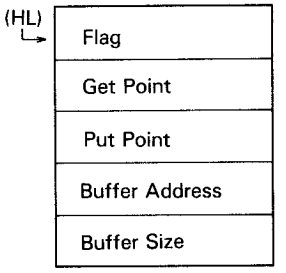
B: 3XH

X = 0	RS-232C on the main board
1	Option 1, channel A
2	Option 1, channel B
3	Option 2, channel A
4	Option 2, channel B

HL: 9 byte block address which is used to store return information.

Return parameter

Z flag = 1 Normal completion
A: 00H No data in the receive buffer
A: FFH A character in the receive buffer
Z flag = 0 The RS-232C port is not open.
A: 3
HL: Not changed



Refer to the parameter block list of the OPEN function.
BC: LOC (the number of characters in the receive buffer)
The contents of the BC register is undefined when error occurs (Z flag = 0).

4. Checking whether the RS-232C is ready to send (OUTST)

Entry parameters

B: 4XH
X = 0 RS-232C on the main board
1 Option 1, channel A
2 Option 1, channel B
3 Option 2, channel A
4 Option 2, channel B
HL: Address of the block which is used to store return information

Return parameters

Z flag = 1 Normal completion
A: 00H Not ready to send
A: FFH Ready to send
Z flag = 0 The RS-232C port is not open.
A: 3
For the HL register contents, refer to the OPEN function.

5. Getting data from the receive buffer (GET)

Entry parameter

B: 5XH

X = 0 RS-232C on the main board
1 Option 1, channel A
2 Option 1, channel B
3 Option 2, channel A
4 Option 2, channel B

HL: Address of the block which is used to store return information

Return parameters

Z flag = 1 Normal completion
A: Receive data
HL: Refer to the OPEN

Z flag = 0 The RS-232C is not open.
A: 3
HL: Return information block address

6. Sending 1 byte data (PUT)

Entry parameters

B: 6XH

X = 0 RS-232C on the main board
1 Option 1, channel A
2 Option 1, channel B
3 Option 2, channel A
4 Option 2, channel B

C: Send data

HL: Address of the block which is used to store return information

Return parameters

Z flag = 1 Normal completion
HL: Refer to the OPEN function.

Z flag = 0 The RS-232C is not open.
A: 3
HL: Not changed

7. Reading status of the control line (CTLIN)

Entry parameter

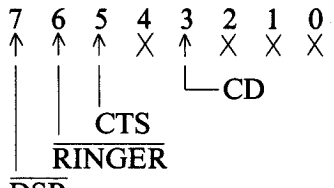
B: 7XH

X = 0 RS-232C on the main board
1 Option 1, channel A
2 Option 1, channel B
3 Option 2, channel A
4 Option 2, channel B

Return parameters

Z flag = 1 Normal completion

A: Conditions of the control line flags



Z flag = 0 Abnormal completion

A = 3

The RS-232C port is not open.

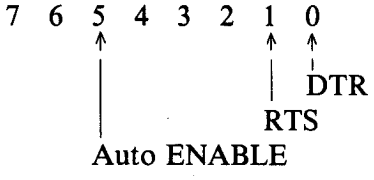
8. Setting data on the control line (SETCTL)

Entry parameters

B: 8XH

X = 0 RS-232C on the main board
1 Option 1, channel A
2 Option 1, channel B
3 Option 2, channel A
4 Option 2, channel B

C: Set data



Return parameters

Z flag = 1 Normal completion

Z flag = 0 Abnormal completion

A = 3

The RS-232C port is not open.

9. Checking error status and clearing error flag (ERSTS)

Entry parameter

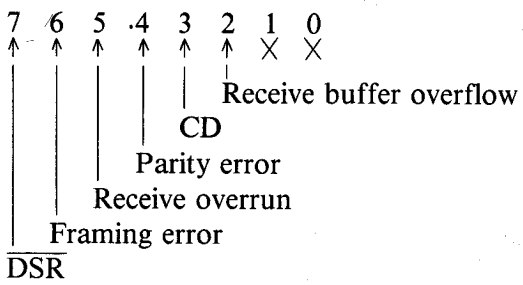
B: 9XH

X = 0 RS-232C on the main board
1 Option 1, channel A
2 Option 1, channel B
3 Option 2, channel A
4 Option 2, channel B

Return parameters

Z flag = 1 Normal completion

A: Error status



The error status flags are cleared after this.

Z flag = 0

A = 3

The RS-232C port is not open.

10. Checking the use of RS-232C port (SENS)

Entry parameters

B: 0FXH

X = 0 RS-232C on the main board
1 Option 1, channel A
2 Option 1, channel B
3 Option 2, channel A
4 Option 2, channel B

Return parameters

Z flag = 1 The RS-232C port can be opened.

Z flag = 0

A: 1 No option board
2 Busy (i.e., the port is used by another program.)

NOTE:
All the RSOPEN, RSCLOSE, RSINST, RSOUTST, RSIN and RSOUT routines call the RSIOX routine after setting parameters according to the data set by the CONFIG command. At this time, the top address of the parameter block is 0FE65H, the buffer address is 7000H and the buffer size is 200H.

LIGHTPEN F657

LIGHTPEN is the entry point of the subroutine which (1) checks whether the light pen is pressed to the screen and (2) reads the location on the screen where the light pen is touched.

Entry parameter:

C: 2 for function (1)
C: 3 for function (2)

Result

Z flag: 1 Normal completion
Function (1) A: 0 Light pen is not pressed to screen.
A: OFFH Light pen is pressed to screen and the light pen interface contains location data.
Function (2) A: 0 Light pen interface contains no data.
A: Other than 0
Light pen interface contains data.
BC: Horizontal position
• Green CRT
0 to 79 non-MFBASIC normal mode
0 to 39 non-MFBASIC MF mode
MFBASIC width 80 mode
MFBASIC width 40 mode
• Color CRT
0 to 39
DE: Vertical position
• Green CRT
0 to 24 non-MFBASIC normal mode
0 to 399 non-MFBASIC MF mode
MFBASIC width 80 mode
MFBASIC width 40 mode
• Color CRT
0 to 399
Z flag: 0 Error
A: 1 Parameter error

This subroutine changes the contents of other registers.

NOTE:

The light pen interface issues successive interrupts if it is continuously pressed onto the screen or against another object, such as a desk or finger. This prevents the CPU from doing other processing. Remember that the interrupt level of the disk drives is lower than that of the light pen.

MASKI F65A

MASKI is the entry point for the subroutine which sets and resets interrupt masks. Masks can be set and reset regardless of the slot in which each option card is installed.

Entry parameters

B: 0	Unmask interrupts from the option interface specified in register C (that, is enables interrupts from the option interface).
1	Masks option level 1. (Disables interrupt)
2	Masks option level 2.
3	Masks option level 3.
4	Masks option level 4.
5	Masks option level 5.
6	Masks option level 6.
7	Masks option level 7.
9	Masks software timer 1.
0DH	Masks software timer 1.
20H	Masks software timer 2.
0FFH	Masks all option levels (except software timers 1 and 2).
C: 0AH	Software timer 1
0BH	Software timer 2
3BH	Light pen
91H	GPIB (IEEE488 bus)
98H	Optical fiber
0A3H	A/D, D/A converter
0ACH	RS-232C
0CCH	RS-232C
0FDH	Multiple font ROM

Result

Z flag: 1	Normal completion
A: 1 to 7	Option interrupt level This value is used as the mask setting parameter.
9	Software timer 1
0DH	Software timer 2
20H	Light pen
Z flag: 0	
A: 1	Option board is not installed.
4	Parameter error Option ID or interrupt level is incorrect.

This subroutine changes the contents of other registers.

NOTE:
All interrupts except those from the keyboard, standard RS-232C, Multiple font ROM, flexible disk controller and clock are masked when a cold start is made. Therefore, interrupts other than the above cannot be processed unless the corresponding mask is reset with this subroutine. The interrupt levels of option interfaces vary according to the slots in which they are installed. Therefore, it is necessary to determine the slot location before resetting a mask.

LOADX F65D

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

Entry parameters

HL: Data address
C: Memory bank number
0: Main bank
1: User bank #1
2: User bank #2 (option)
-1: System bank

Result

A:
0: Normal completion
Other than 0: RAM chips are not installed.
C: Data read
This routine does not change the contents of the other registers.

STORX F660

STORX is the entry point for the subroutine which writes 1 byte of data in a specified memory bank.

Entry parameters

HL: Address in which data is to be written
C: Memory bank number
0: Main bank
1: User bank #1
2: User bank #2 (option)
-1: System bank
A: Data

Result

A:
0: Normal completion
Other than 0: RAM chips are not installed.
This routine will not change the contents of the other registers.

LDIRX F663

LDIRX is the entry point for the subroutine which transfers data between two memory banks.

Entry parameters

Registers

HL: Starting address of the memory area from which data is transferred
DE: Starting address of the memory area to which data is transferred
BC: Number of data bytes to be transferred

Memory

0FEF1H: Source memory bank number (-1 to 2)
0FEF2H: Destination memory bank number (-1 to 2)

Result

Registers

A:
0: Normal completion
Other than 0: RAM chips are not installed.

This routine destroys the contents of all the other registers.

JUMPX F666

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

Entry parameters

IX: Destination address
0FEF2H: Destination bank number (-1 to 2)

This routine does not affect the contents of any registers.

CALLX F669

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

Entry parameters

IX: Call address
0FEF2H: Destination bank number (-1 to 2)

Result

IY: Contents destroyed; the contents of the other registers are maintained, and the contents of address 0FEF2H are destroyed.

NOTE:
The CALLX subroutine uses 1 level of stack. Therefore, the stack pointer area must be settled in the common RAM area (0E000H - 0FFFFH) and its size must consider 1 level for this routine.

GETPFK F66C

GETPFK is the entry point for the subroutine which sets the character string assigned to a specified programmable function key in a specified memory area.

Entry parameters

B: 0: Function key for non-MFBASIC
1: Function key for MFBASIC
C: Programmable function key number (0 to 9)
HL: Starting address of the memory area in which the character string is set.

Result

HL: Contents maintained; the contents of all the other registers are destroyed.

PUTPFK F66F

PUTPFK is the entry point for the subroutine which assigns the character string stored in a specified memory area to a specified programmable function key. If the specified memory area contains more than 16 characters, only the first 16 are valid.

Entry parameters

B: 0: Function key for non-MFBASIC
1: Function key for MFBASIC
C: Programmable function key number (0 to 9)
HL: Starting address of the memory area which contains the character string to be assigned to the programmable function key. The first byte of the area represents the number of characters contained in the area and the following bytes contain the character string.

Result

HL: Contents maintained; the contents of all the other registers are destroyed.

For examples of BIOS call, see Appendix F.