

# Chapter 5

## DATA COMMUNICATIONS SUPPORT

### 5.1 PX-4 Communications Support

The PX-4 computers come as standard with an RS-232C interface, a serial (SIO) interface, and a cartridge serial interface. These interfaces are treated as communications devices through which the PX-4 communicate with external peripheral equipment. The devices allow serial input/output and are assigned the following drive names:

- COM0: For RS-232C input/output
- COM1: For SIO input/output
- COM2: For RS-232C input or SIO output
- COM3: For Cartridge serial input/output

**NOTE:**  
*Only one serial device can be opened at a time.*

The cartridge serial and RS-232C interfaces can be handled in the same way using the same commands except for the communications procedures. The differences in communications protocol are due to the presense or absence of control signals for the respective interfaces. The control signals available for the above interfaces are listed below.

□ Control signals □

RS-232C	SIO	Cartridge serial
DSR	SIN	None
DTR	SOUT	None
CTS	None	None
RTS	None	None
DCD	None	None

**NOTE:**  
*For descriptions of the control signals for the individual interfaces, refer to Chapter 3, "STANDARD INPUT/OUTPUT INTERFACE" of the Operating Manual.*

When connecting an external device other than those available for the PX-4 to the SIO or cartridge serial interface, it is necessary to select the proper cable considering the pin assignments and communications procedures.

## 5.2 RS-232C Interface

The RS-232C interface is an interface whose electrical characteristics or specifications are stipulated by standard known as the RS-232C standards. Presently, the use of each signal line and communications protocol for this interface differ from equipment to equipment. When connecting an acoustic coupler or another computer to the PX-4, therefore, the user must carefully check the electrical specifications of the external device and match the conditions for connecting these devices. The PX-4 permits you to set up these conditions from BASIC. You can set up the similar conditions for the SIO and cartridge serial interfaces except for some control signals.

### *NOTE:*

*Refer to the Operating Manual for the cables for connecting between the PX-4 and external devices.*

### 5.2.1 Opening the RS-232C interface

The RS-232C communications interface is opened for data transmission and reception by executing the OPEN statement.

#### (1) Format

**OPEN < mode >, < file number >, "COM0:[(< options >)]"**

#### **mode:**

Specify "I" when the file is to be opened for input (receive) and "O" when it is to be opened for output (transmit). When using the file for both input and output, open it using two OPEN statements with different file numbers as if it were two separate files. In this case, BASIC ignores any options specified in the latter OPEN statement and takes the ones specified in the first OPEN statement.

#### **file number:**

You can specify any number from 1 to 15 smaller than the number specified in the /F: parameter of the BASIC command. Thereafter the file is identified by this number in the program.

#### **drive name:**

Specify one of the communications device names from COM0: to COM3:.  
Specify COM0: when using the RS-232C interface.

**options:**

Specifies the data communication protocol and control options. Specify options with one to seven characters in the (blpscxh) format.

b..... Specifies the baud rate. Specify one of the following letters representing the available baud rates:

Letter	Baud rate (bps)
0	(Send = 1200/Receive = 75)
1	(Send = 75/Receive = 1200)
2	110
3	.....
4	150
5	200
6	300
7	.....
8	600
9	.....
A	1200
B	.....
C	2400
D	4800
E	9600
F	19200
G	38400

l ..... Specifies the number of bits comprising character.

7: 7 bits/character

8: 8 bits/character

p..... Specifies the type of parity check to be made.

N: No parity check

E: Even parity

O: Odd parity

s ..... Specifies the number of stop bits.

1: 1 bit

3: 2 bits

c ..... Specifies which of the four control lines are to be checked. Specify a hexadecimal number 0 to F. Bit 3 has no meaning and may be either 1 or 0. Bits 2 to 0 has the following meanings:

bit 2 ... Specifies whether the DSR level is to be checked in the send mode (DSR corresponds to SIN in the SIO interface).

1: Off (no check).

0: On (DSR checked). If DSR is off, the interface suspends the transmission (output) of a character until DSR goes on.

bit 1 ... Specifies whether the DSR level is to be checked in the receive mode.

1: Off (no check).

0: On (DSR checked). If DSR is off, the interface suspends the reception (input) of a character until DSR goes on.

bit 0 ... Specifies whether the DCD (Carrier Detect) level is to be checked in the receive mode.

1: Off (no check).

0: On (DCD checked). If DCD is off, the interface generates an error.

x ..... Specifies whether the XON/XOFF protocol is to be used for communications control.

X: XON/XOFF protocol enabled.

N: XON/XOFF protocol disabled.

h ..... Specifies whether the shift-in/shift-out (SI/SO) control sequences are to be used.

S: Shift-in/shift-out control enabled in the 7-bit/character communications mode. S is disallowed in the other communications mode.

N: Shift-in/shift-out control disabled.

The (blpscxh) options can be omitted entirely or partially. Spaces must be specified, however, for options which are omitted if there are any following options. When BASIC is started, these options are initialized to (D8N3FNN). These settings may be changed using the optional CONFIG command.

## (2) Controls lines

Note that the controls lines to be used differ from device to device.

### **DTR:**

Set to ON whether the interface is opened in the "I" or "O" open mode. DTR is set to OFF when the interface is closed (valid for COM0:, COM1:, and COM2:).

### **RTS:**

Set to ON when the interface is opened in the "O" open mode. RTS is set to OFF when the interface is closed (valid for COM0:).

### **DSR:**

This line is valid when DSR check is enabled in the receive mode. When the interface is opened in the "I" mode with the DSR receive check bit set to 0, the execution of the OPEN statement is not completed until DSR is set to ON. (valid for COM0:, COM1:, and COM2:).

### **DCD:**

This line is valid when DCD check is enabled in the receive mode. When the interface is opened in the "I" mode with the DCD check bit set to 0, the execution of the OPEN statement is not completed until DCD is set to ON. (valid for COM0: and COM2:).

## (3) Errors that may occur during open processing

- **FC Error (Illegal Function Call)**

An attempt was made to open two communications devices at a time.

- **DU Error (Device Unavailable)**

The specified communications device is not ready.

- **DT Error (Device Time Out)**

The DSR or DCD line did not go on within a certain period of time after an OPEN "I" statement was executed with the DSR receive check or DCD check bit set to ON.

## 5.2.2 Output to the RS-232C interface

You can send data to the RS-232C interface in the same way as writing data to a file on an auxiliary storage device such as the RAM disk using the PRINT #, PRINT # USING, and WRITE # statements.

### (1) Control lines

#### **CTS:**

When set to ON, enables data transmission to the external device.

#### **DSR:**

When the DSR send check bit is OFF, data is sent to the RS-232C port regardless of the state of the DSR line. When the DSR send check bit is set to ON, data transmission to the external device is deferred until the DSR line goes on (valid for COM0:, COM1:, and COM2:; COM1: uses the DSR line on the input port).

### (2) Errors that may occur during send operations

- **DT Error (Device Time Out)**

- The CTS line did not go on within a specified time.
- The DSR line did not go on within a certain period of time after an OPEN "I" statement was executed with the DSR receive check bit set to ON.
- The STOP key was pressed for some reason when the interface was waiting for receive data.

## 5.2.3 Input from the RS-232C interface

You can receive data from the RS-232C interface in the same way as reading data from a file on an auxiliary storage device such as the RAM disk. The statements and functions that are used to receive data from the RS-232C are INPUT # and LINE INPUT # (statements) and EOF, LOC, LOF, and INPUT\$ (functions).

### (1) Functions

- **EOF(<file number>)**

The EOF function returns -1 (true) if executed when the receive buffer is empty and 0 (false) when the buffer is not empty.

- **LOC(<file number>)**

The LOC function returns the number of data bytes remaining in the receive buffer.

- **LOF(<file number>)**

The LOC function returns the number of free bytes remaining in the receive buffer.

- **INPUT\$(<no. of characters>, <file number>)**

The INPUT\$ function receives the number of data bytes specified in <no. of characters> from the RS-232C interface and returns them as a character string.

**NOTE:**

*BASIC has a receive buffer of 240 bytes.*

## (2) Control lines

**DSR:**

When the DSR check bit is set to ON, the DSR line is monitored during a receive operation and an error is generated if it is not ON (valid for COM0:, COM1:, and COM2:).

**DCD:**

When the DCD check bit is set to ON, the DCD line is monitored during a receive operation and an error is generated if it is not ON (valid for COM0: and COM2:).

## (3) Errors that may occur during receive operations

- **DF Error (Device Fault)**

The DCD or DSR line was found OFF during a data receive operation. This error can occur when the RS-232C port is opened for input with the DSR receive or DCD check bit set to ON.

- **IO Error (Device I/O Error)**

A parity, overrun, or framing error occurred during a receive operation. The computer can continue processing ignoring the error, but the validity of the data in the receive buffer at that time cannot be guaranteed.

- **IE Error (Input Past End)**

The **STOP** key was pressed while the PX-4 was waiting for receive data during the execution of an INPUT #, LINE INPUT #, or INPUT\$.

## Sample data communications programs:

### <Computer A>

```
5 'COMPUTER A(SEND)
10 OPEN "0",#1,"COM0:(DBN3FNN)"
20 PRINT "ENTER MESSAGE TO SEND...";
30 LINE INPUT A$
40 IF A$="*" THEN 70
50 PRINT #1,A$
60 GOTO 20
70 PRINT #1,A$
80 CLOSE
90 END
```

```
^U
ENTER MESSAGE TO SEND...TEST! TEST! THIS
IS COMPUTER A
ENTER MESSAGE TO SEND...*
OK
```

### <Computer B>

```
5 'COMPUTER A(SEND)
10 OPEN "I",#1,"COM0:(DBN3FNN)"
20 LINE INPUT A$
30 IF A$="*" THEN 60
40 PRINT A$
50 GOTO 20
60 PRINT "MESSAGE COMPLETE"
70 CLOSE
80 END
```

```
TEST! TEST! THIS IS COMPUTER A
MESSAGE COMPLETE
```

## 5.2.4 Closing the RS-232C interface

You can close the RS-232C interface in the same way as a file on an auxiliary storage device such as the RAM disk using the CLOSE statement.

### 5.2.5 Sending a BASIC program

You can send and receive BASIC programs to and from external peripheral devices or other computers via an RS-232C interface using the LOAD, SAVE, and LIST commands. When using these commands for the RS-232C interface, you may specify communications options in the same way as in the OPEN statement. The option character 1 (bits/character) must always be set to 8 or 7 bits and shift-in/shift-out be enabled. Some examples follow.

**SAVE"COM0:"**  
**LIST"COM0:"**

The above two commands send the program in the active program area to the standard RS-232C interface. These commands send the program in the ASCII format and appends a CTRL + Z (&H1A) code to the end of the file. The P option has no meaning for the SAVE"COM0:" command; it sends the program in the ASCII format whether the Option is specified or not.

**LOAD"COM0:"**

This command receives an ASCII format program via the standard RS-232C interface and loads it into the active program area. The command continues receiving program data until it receives a CTRL + Z (&H1A) code. Loading from the RS-232C interface can be stopped by pressing the **STOP** key.

**NOTE:**

*You can also specify COM1: to COM3: interfaces to direct program data to other external devices if an optional RS-232C board is installed.*

**Examples:**

```
SAVE "COM0:"
SAVE "COM0: (88N1) "
LIST "COM0: (67N S) "
```

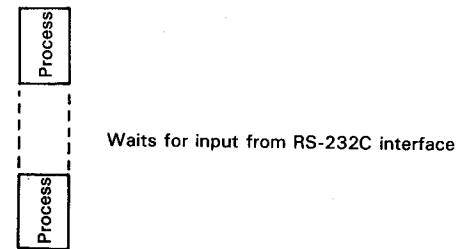
```
LOAD "COM0:"
LOAD "COM0: (E8E1) "
```

### 5.2.6 Handling interrupts from the RS-232C interface

When waiting for data from the RS-232C interface, BASIC normally can do nothing but wait. By using interrupt handling commands, however, BASIC instructs BASIC to do other processing while it is waiting for input from the RS-232C interface.

**When interrupt is disabled**

BASIC can do nothing while it is waiting for input from the RS-232C interface as illustrated below.



• **When interrupt is enabled**

As shown below, BASIC can do some tasks while it is waiting for input from the RS-232C interface. When input data is received at the RS-232C interface, BASIC executes the interrupt handling routine specified in the ON COM GOSUB statement. Control is returned from the interrupt handling routine via a RETURN statement in the interrupt handling routine.

