CHAPTER 4 SERIAL COMMUNICATION

4.1 General

Serial communication is performed by the start-stop synchronous transmission system. In start-stop transmission, the signal is logic 1 while no data is being sent and becomes 0 to show the start of data. (See Fig. 4-1.) This first bit is called the start bit. Bits are then transmitted sequentially from the lowest order bit. A stop bit (logic 1) is always included to end transmission of the character. Stop bits may be 1 or 2 bits long. The word length (data bits) must be 5 to 8 bits. Bit time varies according to the bit rate. At 300 BPS (bits per second), the duration of a single bit is 3.3 msec.

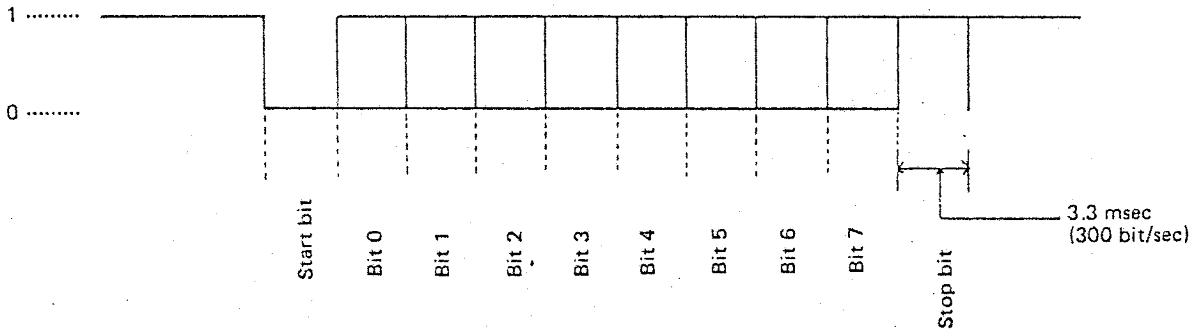


Fig. 4-1 Start-stop Data Transmission Format

The figure below shows an example of signal status when data 3A (00111010$_2$) is transmitted in start-stop format with a single stop bit.

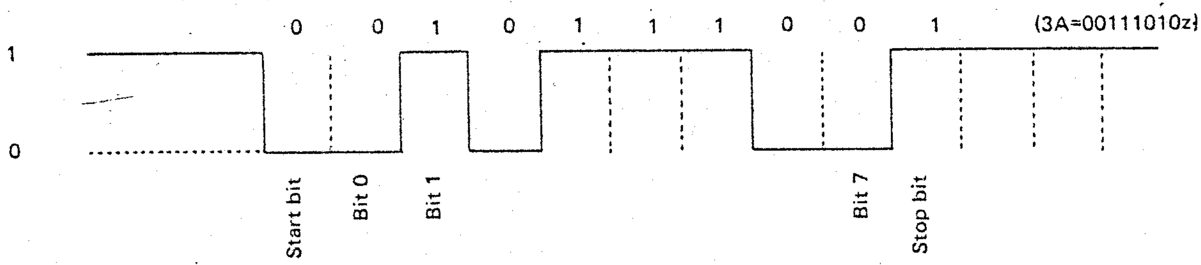


Fig. 4-2 Start-stop DATA 3A

Data 1 is represented by a low signal (-3 to -8V) and data 0 by a high signal (+3 to +8V) as shown in Fig 4-3 below.

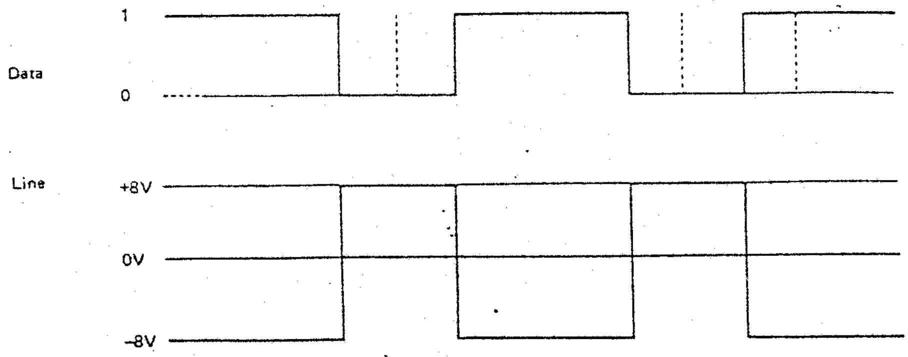


Fig. 4-3 Relationship Between Data and Signals States

The status signal lines are RTS (output), CTS (input), DTR (output), DSR (input), and CD (input). These signals are ON when high and OFF when low (Fig. 4-4).

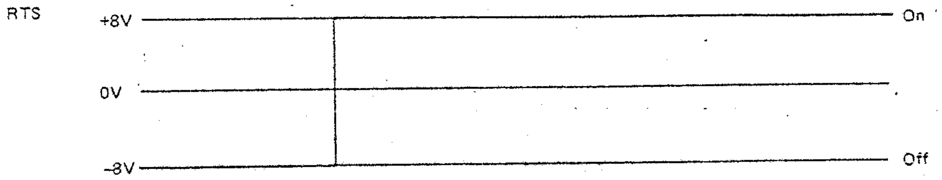


Fig. 4-4 Signal Line Output Status (RTS)

The HX-20 is provided with two types of interfaces. These are serial and RS-232C. The serial interface uses the serial port of the MCU and the bit rates and word length are fixed. The RS232C interface, however, performs handshaking by software. It can therefore support bit rates up to 4800 bps and both the bit rate and word length can be set by the user. Fig. 4-5 shows the respective range of functions for the serial and RS-232C interfaces.

	Transmission speed	Word length (bits)	Stop bit (bits)	Control lines (input)	Control lines (output)
Serial	38.4K BPS 4.8K BPS 600 BPS 150 BPS	8		1	1
RS232C	4.8K BPS max.	5, 6, 7 or 8	1 or 2	3 (CTS, DSR, and CD)	2 (RTS and DTR)

Fig. 4-5 Functions of Serial and RS-232C Interfaces

The serial interface is used for communication between the master and slave MCUs and for the floppy disk units.

4.2 I/O Ports

Table 4-1 I/O Ports for Serial Communication

MCU	Port (address)	Input/ output	Signal name or function	Signal state	Port bit state
Master	P10	Input	DSR (RS-232C)	High	0
				Low	1
	P11	Input	CTS (RS-232C)	High	0
				Low	1
	P16	Input	PIN (serial control line)	High	0
				Low	1
	P21	Output	TXD (RS-232C)	High	0
				Low	1
	P22	Output	Selection of slave or serial for CPU serial communication		0: Slave 1: Serial
	RMCR (0010)	Output	Serial bit rate control		

MCU	Port (address)	Input/ output	Signal name or function	Signal state	Port bit state
	TRCSR (0011)	Input	Serial control and status		
	SRDR (0012)	Input	Serial receive data		
	STDR (0013)	Output	Serial transmit data		
	$26 Bit 5	Output	POUT (serial control line)	High	0
				Low	1
Slave	P20	Input	RXD (RS-232C)	High	0
				Low	1
	P31	Output	RTS (RS-232C)	High	0
				Low	1
	P36	Output	Serial and RS-232C interface driver ON/OFF		0: On 1: OFF
	P45	Output	P20 signal selection		0: RS-232C 1: Micro-cassette
	P47	Input	CD (RS-232C)	High	0
				Low	1

Note: DSR: Data set ready
CTS: Clear to send
TXD: Transmit data
RTS: Request to send
DTR: Data terminal ready
RXD: Receive data
CD: Carrier detect

4.3 Serial Communication Procedure

The SCI (serial communication interface) in the MCU performs serial communication in the following procedure.

(1) Driver ON

The communication driver is turned ON. The port for driver ON is connected to the slave MCU. Subroutine SERONF turns the drivers ON/OFF.

(2) Serial switching

The serial communication lines of the MCU can be used either for external data communication or for communication with the slave MCU. Normally, the slave MCU is selected. To select external communication, port P22 of the main MCU is set to 1.

(3) Bit rate setting

RMCR (address 10) sets the bit rate. The bit rate is normally set to 38.4K BPS. Table 4-2 shows selection of bit rates by RMCR.

Table 4-2 SCI Bit Time and Bit Rates

Lower 4 bits	Hexadecimal	Bit time/bit rate
0100	4	26 μsec/38.4 BPS
0101	5	208 μsec/4.8K BPS
0110	6	1.67msec/600 BPS
0111	7	6.67msec/150 BPS

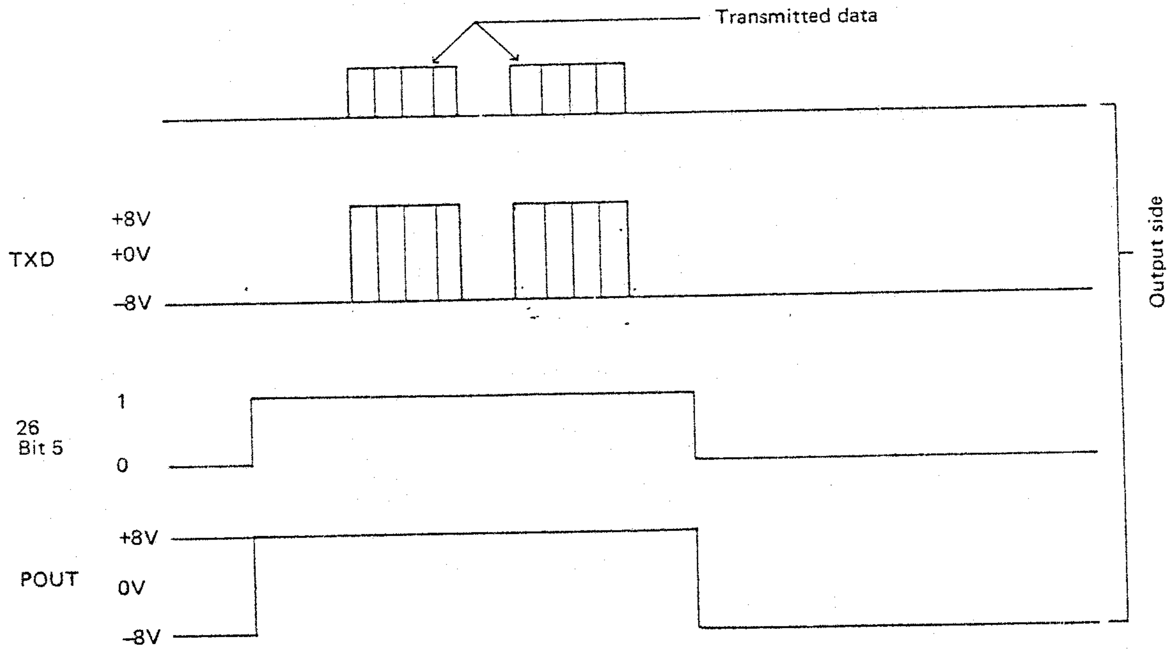
(4) Data transmission (one byte)
TRCSR (address 0011) is input and when it is confirmed that TDRE (bit 5 of TRCSR) is 1, one byte of data is transmitted by writing it to TDR(address 0013).

(5) Data reception (one byte)
TRCSR is input and if RDRF (bit 7 of TRCSR) is 1, serial data can be received by SRDR (address 12). One byte of Serial data is then received. Note that if the received data is not fetched before the next data is received, an overrun error occurs (ORFE is set at 1).

(6) Termination procedure
The bit rate is set to 38.4K bps (procedure 3) and the driver is turned OFF. This procedure is followed because transmission of commands to the slave MCU is always performed at 38.4K PBS.

4.4 Control Lines

Two control lines are available: PIN (input) and POUT (output). PIN is connected to P16 (bit 6 of port 1) and POUT is connected to bit 5 of address 26. Both of these signals are set at 1 when the signal goes high and at 0 when the signal goes low. Subroutine WRTP26 is used to set data in address 26. Fig. 4-6 shows the relationship of the signals and ports.



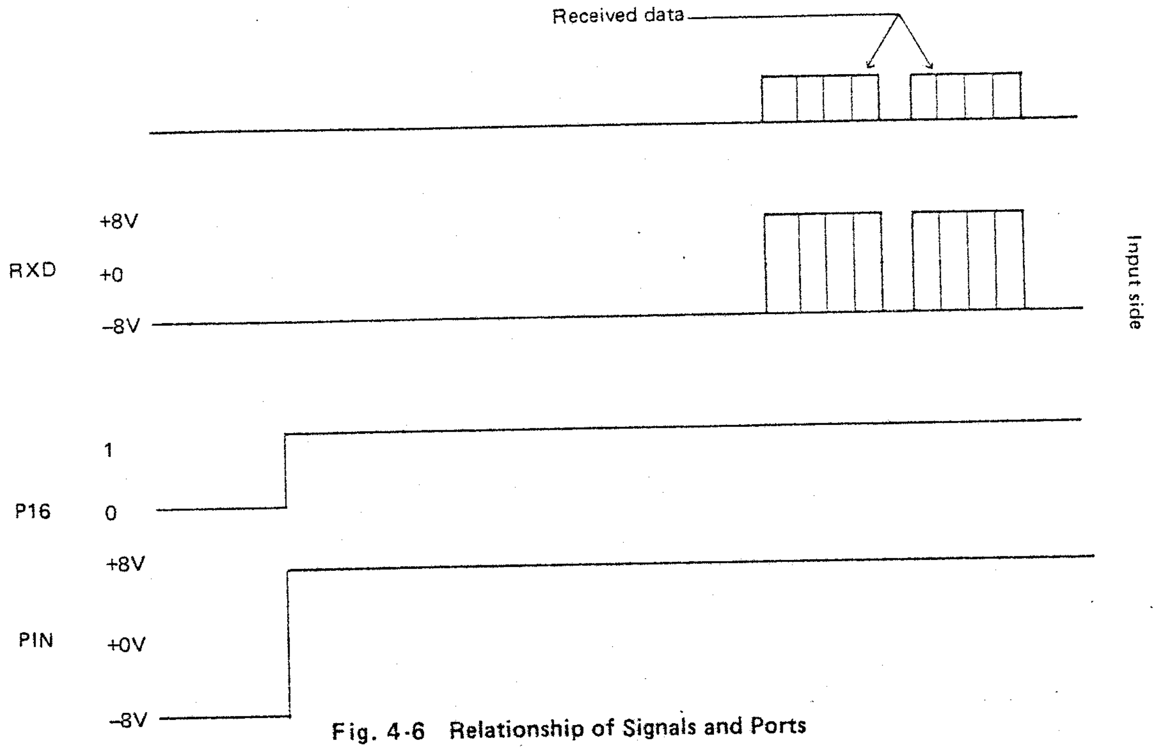


Fig. 4-6 Relationship of Signals and Ports

4.5 High-Speed Serial Communication
EPSP (EPSON Serial Communication Protocol) is provided to enable serial communication between the HX-20 and a floppy disk unit (TF-20) or between two HX-20s.

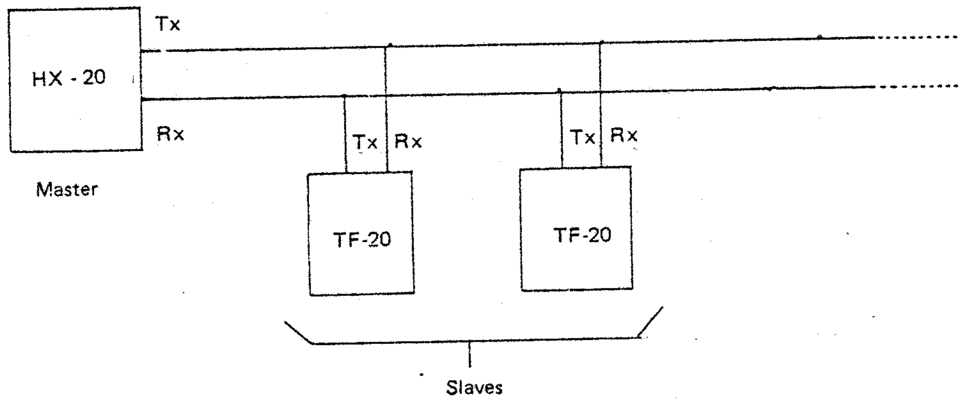


Fig. 4-7 Connection of Slave Devices to HX-20 for Serial Communication

Fig. 4-7 shows how slave devices can be connected by data lines to the HX-20. Up to two slave devices can be connected to a single master device. Each slave is assigned a device number by the master. The master then uses the device number to select which of the slaves to perform communication with. The master can only communicate with one slave at a time. Communication between slave devices cannot be performed.
Fig. 4-8 shows the format for messages sent from the HX-20 to a slave device.

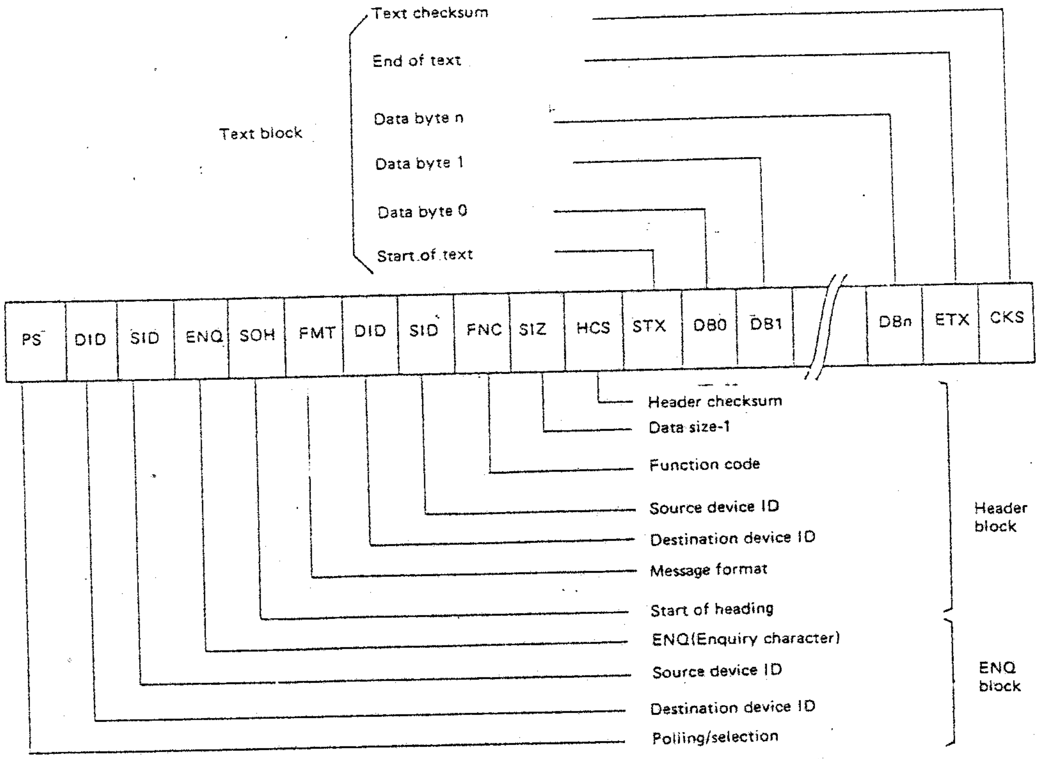


Fig. 4-8 Message Format

Messages sent from the master can be divided into three blocks described below.

(1) ENQ block
PS to ENQ in Fig. 4-8. The master sends this block to request connection with a slave.

(2) Header block
SOH to HCS in the above figure.
This block specifies the data format etc.

(3) Text block
STX to CKS in the above figure. The text block contains the actual data transmitted.

Details of each block are as follows.
ENQ block
The contents of the ENQ block are shown below

PS	DID	SID	ENQ

The master device selects one of the slave devices and issues a connection request to it, using this block. When connection with the slave has been established, the header and text blocks are sent. Once a slave device has been connected, this procedure is not repeated until a new slave device is selected for communication. The selected slave device issues an ACK signal in response to the connection request from the master (Fig. 4-9).

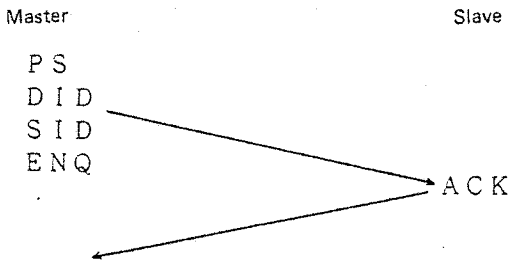


Fig. 4-9 ENQ Block Procedure

PS specifies polling/selection. At present, however, only selection is supported. The code for PS is 31_{16} = 1.
DID indicates the destination device ID. The following destination device IDs are available.

31_{16}	: Floppy disk drive A
32_{16}	: Floppy disk drive B
33_{16}	: Floppy disk drive C
34_{16}	: Floppy disk drive D
20_{16}	: Master HX-20

The code for NEQ is 05_{16}.

Header block
The master transmits the header block to specify the message format and the function codes as well as text size to be sent to the floppy disk unit in the text block that follows.
The contents of the header block are shown below.

SOH	FMT	DID	SID	FNC	SIZ	HCS

SOH: Indicates the start of the header. The value is 01.
FMT: Indicates the header block format.
 00 indicates that the master device is transmitting a message to a slave device.
 01 indicates that a slave device is transmitting a message to the master device.
DID: Indicates the destination device ID. The codes for DID are the same as in the ENQ block.
SID: Indicates the source device ID.
FNC: Specifies the function of the disk unit. Must be 00 to FF. For details of each function, refer to the descriptions in the corresponding sections.
SIZ: Indicates the text block size. This value is the number of bytes in the text block (excluding STX and CKS) minus 1. The value of SIZ must be in the range 0 to 255_{10}.
HCS: Indicates the checksum of the header block. The value is such that the lower 8 bits of the sum of the values of the header block (SOH to HCS) will all be 0.

When the slave device receives a correct header block, it responds by sending 'ACK' to the corresponding source device. If the slave device receives an incorrect header block, it responds by sending 'NCK' to the source device.

Text block
The text block contains the actual data to be sent to the selected device. The text block follows the header block.
The contents of the text block are shown at right.

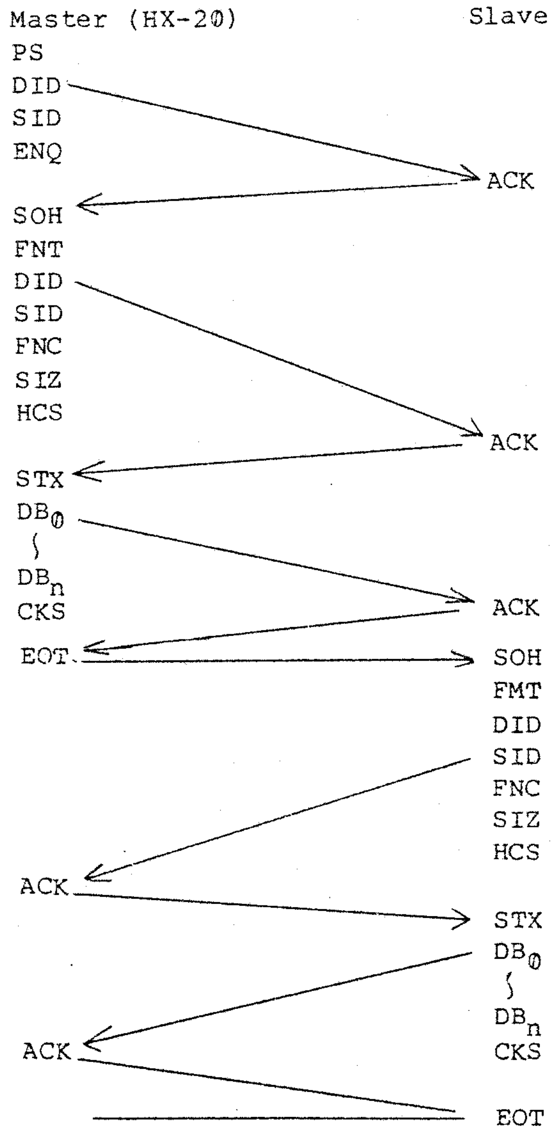
STX	DB_0	DB_1	~	DBn	CKS

STX: Indicates the start of text. The value is Ø2.
$\mathrm{DB}_{\emptyset}$: Data Ø
DB_n: Data n (n ⩽ 255)
EXT: Indicates the end of text.
CKS: Indicates the checksum of the text block. The value is such that the lower bits of the sum of the values of the text block (STX to CKS) will be Ø.

When the slave device receives a correct text block, it responds by sending ACK to the source device. If the slave device receives an incorrect text block, it responds by sending 'NCK' to the source device.

Switching Transmit State

There are cases when the master (HX-2Ø) will request data transmission from a slave device (e.g., floppy disk unit). In this case, the sending and receiving sides (master and slave) are reversed. Switching over from master-to-slave to slave-to-master data transmission is accomplished by the following procedure.
The master sends EOT (code Ø4) to the slave after it has received an ACK from the slave indicating the slave has correctly received the text block. The slave device, after receiving EOT, sends the header and text blocks to the master device. It then sends EOT to the master and the transmit state returns to master-to-slave. (Fig. 4-11).



Details of protocol are shown in the EPSP standard at the end of this chapter.

4.6 Subroutines for Serial Communication

The following four subroutines support serial communication using EPSP procedures:

1. SERONF: Turns ON/OFF the serial communication drivers.
2. SEROUT: Transmits the ENQ, header, and text blocks.
3. SERIN: Receives the header and text blocks.
4. SRIWIT: Sets constants and performs initialization.

4.7 High-Speed Serial Subroutines

Subroutine name	Entry point	Description
SERONF	FF73	Turns ON/OFF the high-speed serial driver. This subroutine checks bit 4 of 'SRSTS' and turns ON the driver only when both are off.
		The contents of the SERONF parameters are the same as those of RSONOF.
SEROUT	FF7Ø	High-speed serial data output (EPSP-based data transmission). This subroutine transmits the ENQ, header, and text blocks to the specified device according to the ENQ... SOH... STX procedure.
		Parameters: At Entry (X): Head address of a data packet (A): Indicates whether to proceed to the receive procedure after completion of the transmit procedure. ØØ: Transmit procedure only. Ø1: (LSB=1) proceeds to the receive procedure after completion of the transmit procedure. Packets 1. EMT (1 byte) 2. DID (1 byte) 3. SID (1 byte) 4. FNC (1 byte) 5. SIZ (1 byte) 6. (data string) (1 byte) ∫ n At Return (C): Abnormal I/O flag (A): Return codes ØØ: Normal end BØ: Time out B1: Not linked (device error) B2: Communication error B3: Driver OFF. (Z): according to the value of (A) Registers retained None Subroutines referenced CHKRS Variables used RØ, R1, R2, R3, R4, and R5H
SERIN	FF6D	Receives the header and text blocks according to the SOH ... STX ... procedure (high-speed serial data block reception.)
		Parameters: At Entry (X): Head address of receive data block.

Subroutine name	Entry point	Description
		Parameters: At Return (C): Abnormal I/O flag (A): Return codes 00: Normal B0: Time out B2: Error during receive procedure (B): Indicates the receive block status when (A) is 00. 00: Data with a header string (SOH...) received. 01: Data without a header string received. (Z): According to the value of (A). Note: The format of a data block received is the same as that of the data block transmitted. 1. FMT 2. DID 3. SID 4. FNC 5. SIZ 6. Data ⸾ n Registers retained None Subroutines referenced CHKRS Variables used R0, R1, R2, R3, R4, and R5H

SRINIT	FFIC	Sets constants and performs high-speed serial initialization.
		Values of constants on initialization.
		SRTCN ← 3
		SRTMO ← 10_{10}
		SREMO ← 100_{10}
		SRAMO ← 10_{10}
		SRTDL ← 1
		Others ← 0

		Parameters: At Entry (A): Value of SRMODE 00 or 01 (00: master) At Return None Registers retained None Subroutines referenced None Variables used None

4.8 High-Speed Serial Communication Work Areas

Address	Variable name	Byte count	Description
1C4 1C4	SRFMT	1	FMT (format) data
1C5 1C5	SRDDEV	1	DID (Destination Device ID) data
1C6 1C6	SRSDEV	1	SID (Source Device ID) data
1C7 1C7	SRFNC	1	FNC (function) data
1C8 1C8	SRSIZ	1	SIZ (Size) data
1C9 1C9	SRACKC	1	ACK character (Sent from destination device on completion of block transmission)
1CA 1CA	SRTRCN	1	Number of time same block has been sent.
1CB 1CB	SRTIMO	1	Time out for received characters (unit: msec)
1CC 1CC	SRETMO	1	Time out for received block reception (unit: msec)
1CD 1CD	SRATMO	1	Time out for received ACK characters (unit: msec)
1CE 1CE	SRMODE	1	Relationship between devices (0: Master, any other value: Slave)
1CF 1CF	SRETDL	1	Idle time after EOT transmission (unit: msec)
1DO 1DO	SRBLCN	1	Number of received data (block reception)
1D1 1D1	SRERMD	1	Error (block reception)
1D2 1D2	SRRVFL	1	Not used
1D3 1D4	SREIX	2	Address where received data is stored (block reception)

APPENDIX SERIAL COMMUNICATION PROTOCOL (EPSP)

1. BASIC LINE SPECIFICATION

1.1 Transmission Speed
1.2 Synchronization
1.3 Communication
1.4 Transmission
1.5 Response System
1.6 Error Control
1.7 Transmission Codes
1.8 Bit Transmission Sequence bit Ø, bit 1, ... bit 7

2. TRANSMISSION CHARACTERS AND SEQUENCE

PS DID SID ENQ	Request receiving side to prepare to receive data
SOH	Indicates start of header block.
STX	Indicates start of text block.
ETX	Indicates end of text block.
ACK	Acknowledge
NAK	Negative acknowledge
DLE ;	Waits for WAK, acknowledge or transmission.
ENQ	Prompt for block response.
EOT	Releases data lines.

PS must be '1'=31_{16}. Control characters, DID and SID must be 8 bits (MSB=Ø).

3. MESSAGE FORMAT

3.1 Header Format

SOH	Start of header
FMT	Text format ØØ: Indicates that the master is transmitting a block. Ø1: Indicates that a slave is transmitting a block.
DID	Destination ID
SID	Source ID
FNC	Text function
SIZ	Text size (in bytes) This value is the length of the text block (excluding STX, ETX and CSK) minus 1.
HCS	Checksum of header block This is a value such that the lower 8 bits of the sum of SOH to ACS are Ø.

3.2 Text Format

STX: Start of text
DB0: Data 0
DB1: Data 1
≀
DBn: Data n
ETX: End of text
CKS Checksum of a text block

The value of CKS is such that the lower 8 bits of the sum of STX to CKS are 0.

Text length excluding STX, ETX and CKS must be within 256 bytes.

4. RESPONSE TO SLAVE SELECTION SEQUENCE

(a) ACK (Acknowledgement)

Indicates that that slave can receive a block. The master then initiates data transmission.

(b) NAK (Negative acknowledgement)

Indicates that the corresponding I/O device is not connected or that an error has occurred and the slave cannot receive data. The master then issues EOT and terminates the data link. The master will also send EOT to terminate the data link by transmitting if no response is received within a fixed period of time or an invalid response other than ACK and NAK is received after a selection sequence has been sent.

5. HEADER BLOCK TRANSMISSION

5.1 Response to a Header Block

(a) ACK (Acknowledge)

Indicates that the slave has received a correct header block. The master proceeds to the next phase.

(b) NAK (Negative acknowledge)

Indicates that the slave has received an incorrect header block. In this case, the master repeats transmission of the same block. If the master still receives NAK after the block has been transmitted a specified number of times, it assumes a line error and terminates the data link (by send EOT).

(c) WAK (Acknowledge and temporary wait)

Indicates that the slave has received a correct block but that it cannot yet receive the next block. The master will wait and then issue ENQ to prompt a response from the slave.

(d) No response or invalid response
If no response is made within a given time or a response other than ACK, NAK or WAK is received, the master will issue ENQ to prompt a response from the slave.
If no correct response is received even after ENQ has been transmitted a specified number of times, the master assumes an error and terminates the data link.

6. TERMINATION

(a) When, after sending EXT to the slave, the master receives ACK, it sends EOT to the slave and terminates the data link.

(b) When a transmission error occurs after the data link has been established, or during data transmission, the master will terminate the data link by transmitting EOT.

7. TIME SUPERVISION

(1) Number of selection sequences transmitted
The master will repeat the selection sequence after receiving a response other than ACK from the slave for the number of times listed in the table below.

	Mode Ø	Mode 1
NAK	One time	One time
No response or invalid response	Three times (at 1-sec intervals)	Three times (at 3-sec intervals)

(2) Number of transmitted ENQs (response retransmit request)

	Mode Ø	Mode 1
No response or invalid response	Three times (at 1-sec intervals)	Three times (at 3-sec intervals)

(3) Timers

	Mode Ø	Mode 1
Response wait timer	1 sec	3 sec
Interblock supervision	32 sec	96 sec
Character supervision	1 sec	3 sec

8. TERMINAL NUMBERS

31_{16} : Floppy disk drive A
32_{16} : Floppy disk drive B
33_{16} : Floppy disk drive C
34_{16} : Floppy disk drive D

Terminal numbers (slave): 30_{16} through $3F_{16}$ for mode 0
40_{16} through $5F_{16}$ for mode 1
Center number (master) : 20_{16}

9. OMISSION OF A HEADER BLOCK

If the terminal previously transmitted to is still selected and the header to be transmitted is the same as the last transmission, the header may be omitted. In this case, the master need only transmit the data block following STX. The slave treats this data block without header as if it included the header of the previously received data block.

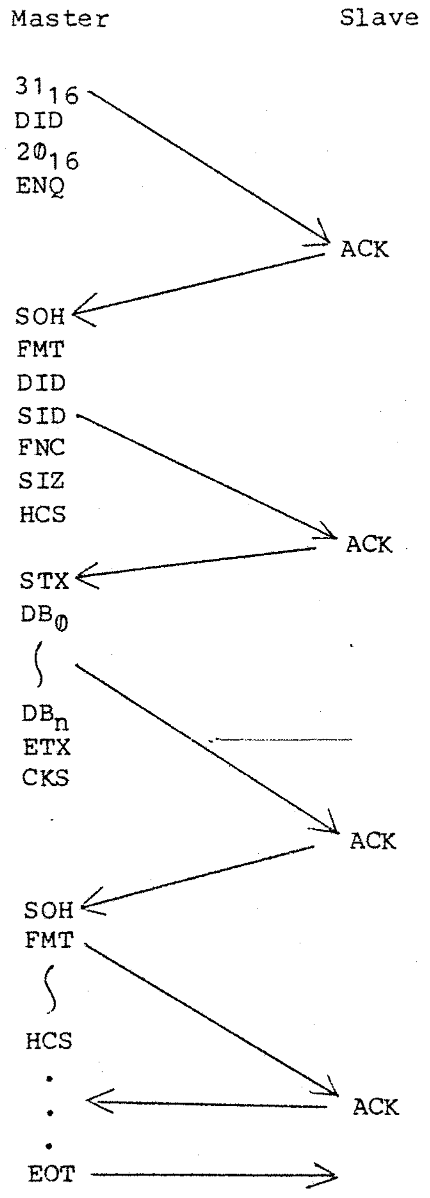
10. TRANSMISSION CONDITIONS SUPPORTED BY VER-1

Transmission speed is 38.4K BPS.
Mode 0 is used for a time supervision.
Header block cannot be omitted.

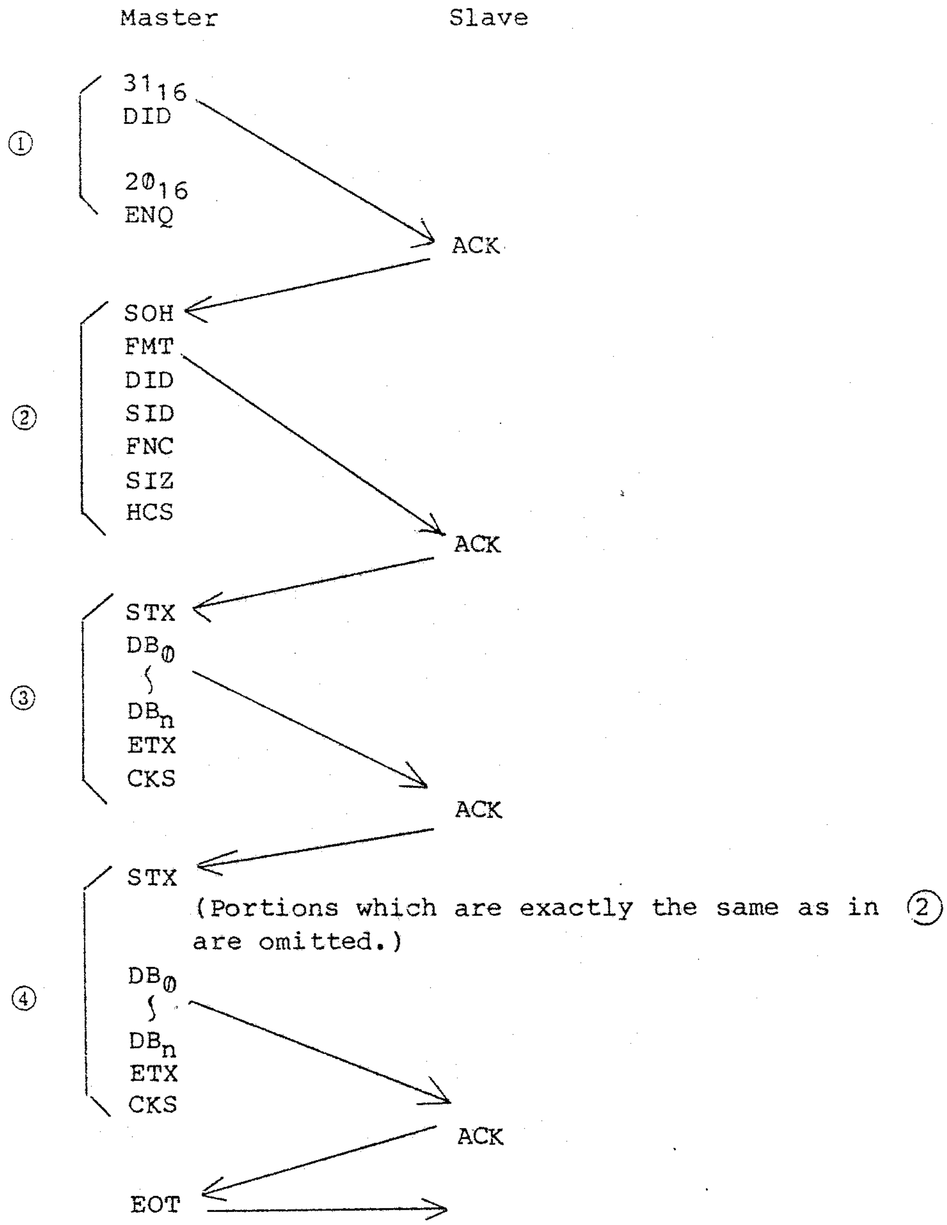
TRANSMISSION PROCEDURE DIAGRAMS

1. WITHOUT ERRORS

(1) When the slave does not send a data block to the master in response to the master's transmission and a header is not omitted.

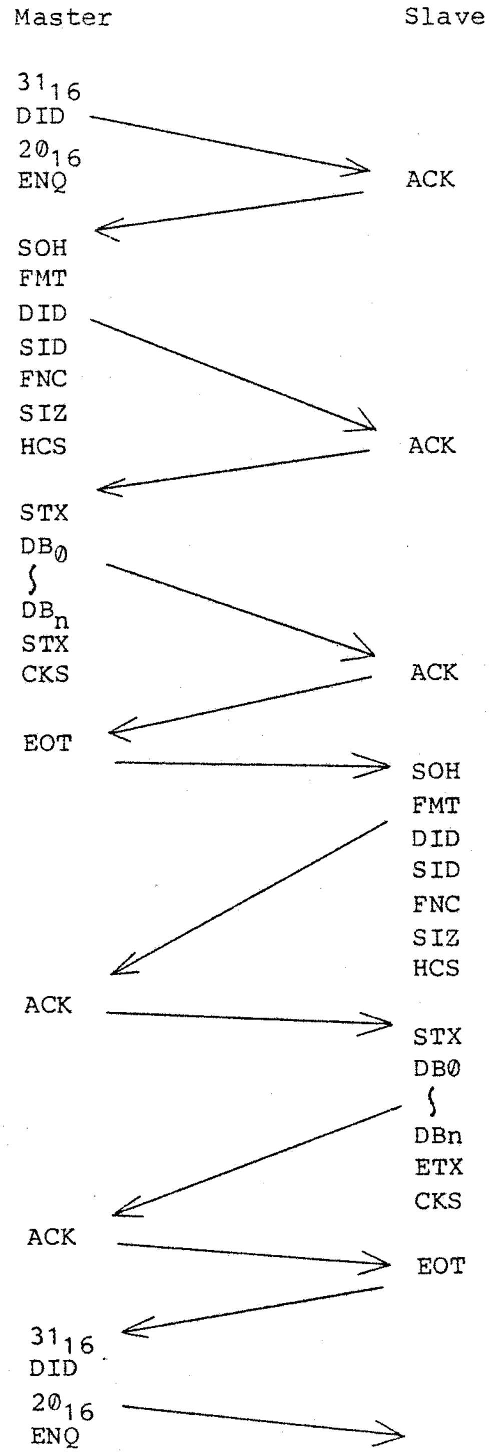


(2) When the terminal does not transmit a data block to the master in response to the master's transmission but the header is omitted.

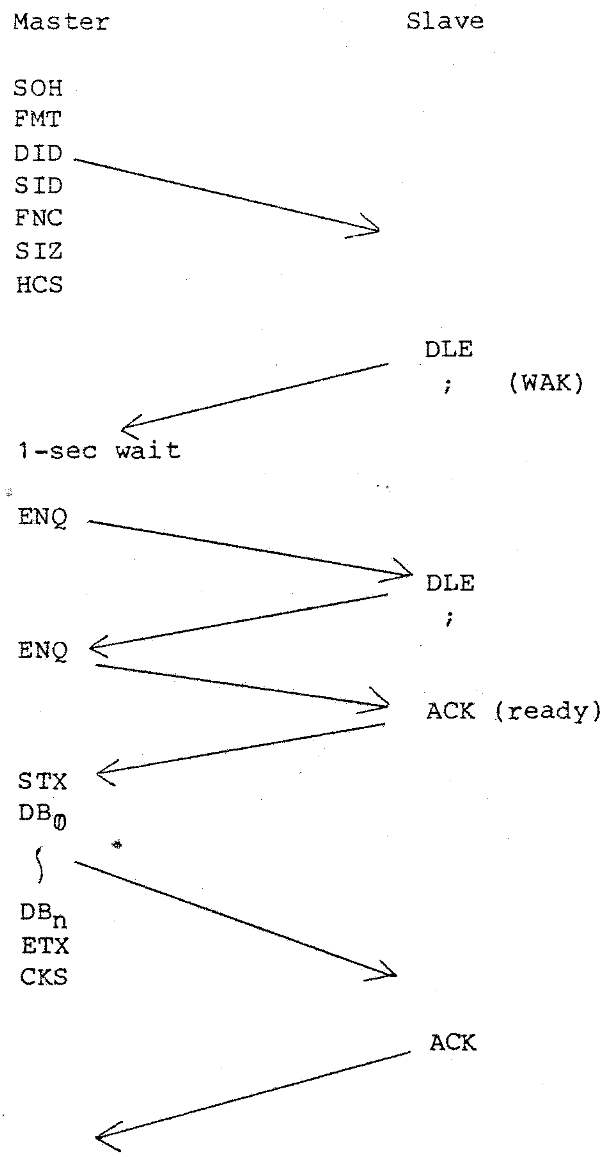


(Portions which are exactly the same as in ② are omitted.)

(3) When the terminal transmits a data block to the master in response to EOT from the master and the header is not omitted.

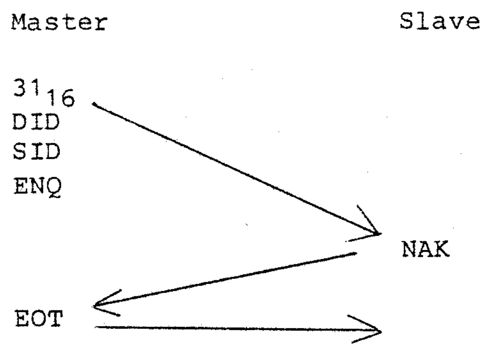


(4) When the slave responds with WAK to a block transmission with header from the master.

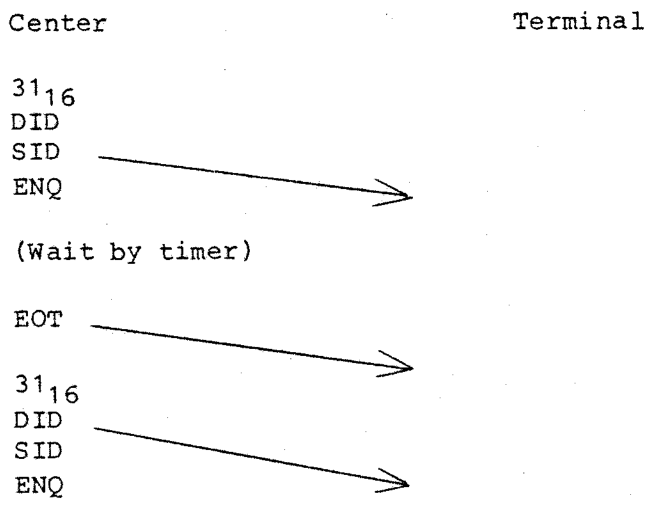


2. WITH ERRORS

(1) When the slave responds by sending NAK in response to ENQ from the master.

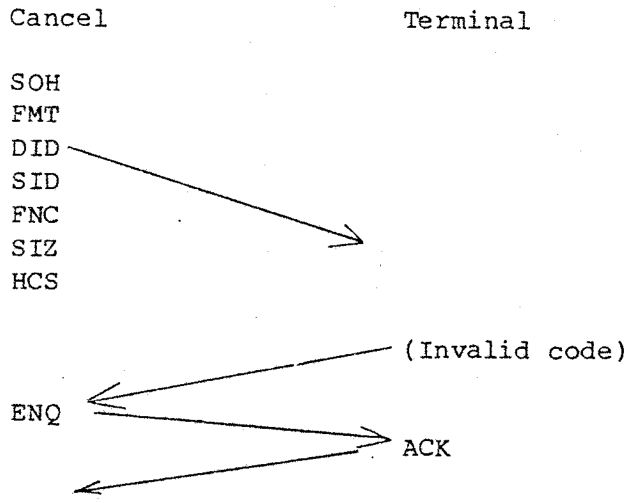


(2) When no response or an invalid response is received from the slave in response to ENQ from the master.

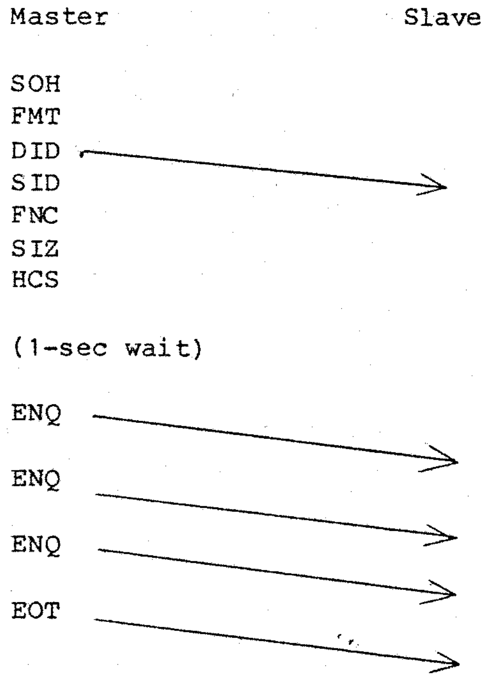


ENQ is transmitted three times.

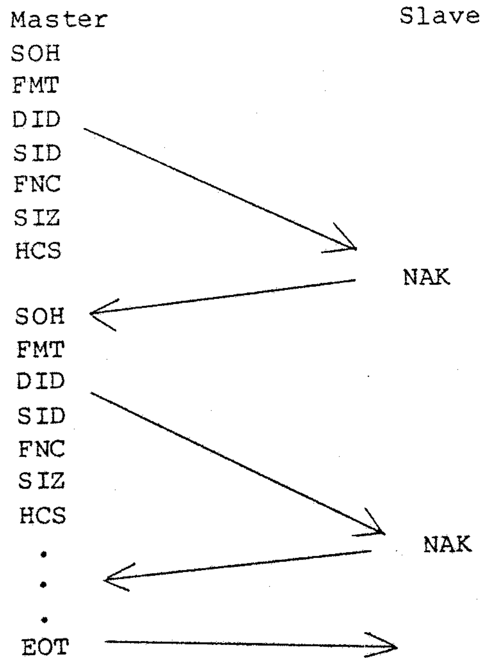
(3) When the master receives an invalid code from the slave in response to transmitting a header to the terminal.



(4) When no response is received from the slave in response transmission of a header from the master.



(5) When NAK is sent from the slave in response to transmission of a header from the master.



If the master receives NAK three or more times, it terminates the data link by transmitting EOT.
If the slave transmits NAK three times in succession, the master will not send the header but will send EOT to terminate the data link.

(6) When the master receives an invalid response code from the slave in response to text transmission, the master handles this as in (3) above.

(7) If no response is received from the slave, the master handles this as in (4) above.

(8) If the slave responds with NAK when the master transmits text, the master handles this as in (5) above (text retransmission).

(9) When the master does not receive a correct response after sending EOT to the slave.
   a. If there is no response, the master waits for 1 second and terminates the data link by transmitting EOT.
   b. If the master receives a code other than EOT from the slave, it terminates the data link by transmitting EOT.

(10) When the slave has not correctly received and responded to EOT sent from the master (when response from the slave is necessary)

a. If the slave has not received EOT and does not respond to the master, the master will wait (3 seconds in mode Ø), terminate the data link and restart the link procedure from the beginning.

b. If the slave receives a code other than EOT, the master assumes that the terminal has made no response. If the slave does not receive EOT after the master has sent EOT the specified number of times (3 times), the center returns to the link start procedure.

(11) If the slave does not transmit a header after the master transmits EOT, the master requests the slave to transmit the header by retransmitting EOT after waiting for a given time. If the slave does not transmit the header even after EOT has been transmitted the specified number of times, the master assumes that an error occurred and terminates the data link.

Function character code table

	Ø	1
Ø		DLE
1	SOH	
2	STX	
3	ETX	
4	EOT	
5	ENQ	NAK
6	ACK	
7		
8		
9		
A		
B		
C		
D		
E		
F		

```
      00001                                    NAM     EPSP
      00002                               *
      00003                               *  READ CHARACTER FROM KEYBOARD AND SEND CHARACTERS TO ANOTHER HC-20
      00004                               *    BY EPSP.
      00005                               *  AND AT ANOTHER HC-20.
      00006                               *    RECEIVED CHARACTER FROM EPSP, DISPLAY CHARACTERS ON THE VIRTUAL
      00007                               *    SCREEN.
      00008                               *
      00009                               *  FILE NAME   'EXS1' BY  K.A
      00010                                    TTL     --- SERIAL (EPSP) EXAMPLE ---
      00011                               *
      00012                                    OPT     LOAD
      00013                                    OPT     PAGE=55
      00014                               *
      00015                               *
      00016                               * SERIAL COMMUNICATION
      00017                               *
      00018                               * CONDITION SWITCH
      00019             0000   A          SRSL   EQU     0           * SERIAL SELECT PROCEDURE
      00020                               *
      00021                               * COMMON DEFINITION
      00022                               *
      00023                               * MPU 6301 I/O PORT
      00024             0002   A          PORT1  EQU     $02         * I/O PORT 1
      00025             0003   A          PORT2  EQU     $03         * I/O PORT 2
      00026             0006   A          PORT3  EQU     $06         * I/O PORT 3
      00027                               *
      00028                               * OTHER REGISTERS
      00029             0011   A          TRCSR  EQU     $11         * TRANSMIT/RECEIVE CONTROL & STATUS REGISTER
      00030             0010   A          RMCR   EQU     $10         * RATE & MODE CONTROL REGISTER (RMCR)
      00031             0013   A          STDR   EQU     $13         * SERIAL TRANSMIT DATA REGISTER
      00032             0012   A          SRDR   EQU     $12         * SERIAL DATA RECEIVE DATA REGISTER
      00033                               *
      00034                               * SUBROUTINE ENTRY POINT
      00035             FF4F   A          DSPSCR EQU     $FF4F       * DISPLAY ONE CHARACTER TO VIRTUAL SCREEN
      00036             FF5E   A          SCRFNC EQU     $FF5E       * VIRTUAL SCREEN FUNCTION
      00037             FF6D   A          SERIN  EQU     $FF6D       * SERIAL RECEIVE
      00038             FF70   A          SEROUT EQU     $FF70       * SERIAL TRANSMITTE
      00039             FF73   A          SERONF EQU     $FF73       * SERIAL DRIVER ON/OFF
      00040             FF9A   A          KEYIN  EQU     $FF9A       * GET ONE CHARACTER FROM KEYBOARD BUFFER
      00041             FF9D   A          KEYSTS EQU     $FF9D       * GET NUMBER OF CHARACTERS IN THE KEY BUFFER
      00042                               *
      00043A 0050                                ORG     $50
      00044                               * GENERAL REGISTERS USED BY I/O ROUTINE
      00045             0050   A          R0     EQU     *           * 2 BYTES REGISTER  (R0H,R0L)
      00046A 0050       0001   A          R0H    RMB     1
      00047A 0051       0001   A          R0L    RMB     1
      00048             0052   A          R1     EQU     *           * 2 BYTES REGISTER  (R1H,R1L)
      00049A 0052       0001   A          R1H    RMB     1
      00050A 0053       0001   A          R1L    RMB     1
      00051             0054   A          R2     EQU     *           * 2 BYTES REGISTER  (R2H,R2L)
      00052A 0054       0001   A          R2H    RMB     1
      00053A 0055       0001   A          R2L    RMB     1
      00054             0056   A          R3     EQU     *           * 2 BYTES REGISTER  (R3H,R3L)
      00055A 0056       0001   A          R3H    RMB     1
```

```
     00056A 0057     0001  A         R3L     RMB     1
     00057           0058  A         R4      EQU     *          * 2 BYTES REGISTER  (R4H,R4L)
     00058A 0058     0001  A         R4H     RMB     1
     00059A 0059     0001  A         R4L     RMB     1
     00060           005A  A         R5      EQU     *          * 2 BYTES REGISTER  (R5H,R5L)
     00061A 005A     0001  A         R5H     RMB     1
     00062A 005B     0001  A         R5L     RMB     1
     00063A 007A                             ORG     $7A
     00064A 007A     0001  A         SRSTS   RMB     1          * SERIAL STATUS
     00065                           *                          * BIT 0,1: RS232 MODE(00:STOP 01:INTERRUPT REA
     00066                           *                                            02:READ ONE CHARACTER)
     00067                           *                          * BIT 2: EXECUTE/PAUSE (0:ON EXECUTE  1:PAUSE)
     00068                           *                          * BIT 3: RS232 DRIVER (0:OFF  1:DRIVER ON)
     00069                           *                          * BIT 4: SERIAL DRIVER (0:OFF  1:DRIVER ON)
     00070                           *                          * BIT 5,6,7: CPU SERIAL RECEIVE INTERRUPT MODE
     00071A 007B     0001  A         RUNMOD RMB      1          * RUN MODE   ($80:BASIC   $00:SYSTEM)
     00072A 007C     0001  A         SIOSTS RMB      1          * SLAVE I/O STATUS   (EACH BIT  0:OFF,  1:ON)
     00073                           *                          * BIT 0: PRINTER
     00074                           *                          * BIT 1: EXTERNAL CASSETTE
     00075                           *                          * BIT 2: INTERNAL CASSETTE
     00076                           *                          * BIT 3: RS232C ON (READ)
     00077                           *                          * BIT 4: SPEAKER ON
     00078                           *                          * BIT 5: PROM CASSETTE
     00079                           *                          * BIT 6: BAR CODE READER
     00080                           *                          * BIT 7: BREAK SLAVE CPU (0:ON EXECUTE
     00081                           *                          *                          1:BROKEN BY INTERRU
     00082A 007D     0001  A         MIOSTS RMB      1          * MAIN I/O STATUS  EACH BIT (0:OFF 1:ON)
     00083                           *                          * BIT 0: LCD ON READ/WRITE CHARACTERS
     00084                           *                          * BIT 1: ON CONTINUE SEND COMMAND TO SLAVE CP
     00085                           *                          * BIT 2: ON CONTINUE TO SEND SERIAL LINE (1:O
     00086                           *                          * BIT 3: ON CLOCK INTERRUPT (1:ON)
     00087                           *                          * BIT 4: (POWER FAIL)
     00088                           *                          * BIT 5: (OFF POWER SWITCH)
     00089                           *                          * BIT 6: ON PAUSE KEY
     00090                           *                          * BIT 7: ON BREAK KEY
     00091                           *
     00092                           *
     00093                           * RAM COMMON WORK AREA
     00094A 01C4                             ORG     $1C4
     00095                           *
     00096                           * WORK FOR SERIAL COMMUNICATION
     00097           01C4  A         SRWKTP EQU      *          * SERIAL WORK TOP ADDRESS
     00098A 01C4     0001  A         SRFMT  RMB      1          * FORMAT                                      (0)
     00099A 01C5     0001  A         SRDDEV RMB      1          * DESTINATION DEVICE                          (1)
     00100A 01C6     0001  A         SRSDEV RMB      1          * SOURCE DEVICE                               (2)
     00101A 01C7     0001  A         SRFNC  RMB      1          * FUNCTION                                    (3)
     00102A 01C8     0001  A         SRSIZ  RMB      1          * TEXT SIZE                                   (4)
     00103A 01C9     0001  A         SRACKC RMB      1          * RECEIVED ACK CHARACTER                      (5)
     00104A 01CA     0001  A         SRTRCN RMB      1          * SEND TRY COUNT                              (6)
     00105A 01CB     0001  A         SRTIMO RMB      1          * FOR RECEIVE CHARACTER TIME OVER LIMIT       (7)
     00106A 01CC     0001  A         SRETMO RMB      1          * FOR RECEIVE BLOCK TIME OVER LIMIT           (8)
     00107A 01CD     0001  A         SRATMO RMB      1          * FOR RECEIVE ACK TIME OVER LIMIT             (9)
     00108A 01CE     0001  A         SRMODE RMB      1          * SERIAL MASTER/SLAVE MODE (0:MASTER)         (10)
     00109                           *                          *                          (NOT:SLAVE)
     00110A 01CF     0001  A         SRETDL RMB      1          * AFTER SEND 'EOT', IDLING TIME (1 = 1 MILI S
    00111A 0100     0001  A        SRBLCN RMB    1          * FOR RECEIVE BLOCK, BLOCK COUNTER
    00112A 0101     0001  A        SRERMD RMB    1          * FOR RECEIVE BLOCK, ERROR MODE
    00113A 0102     0001  A        SRRVFL RMB    1          * FOR RECEIVE BLOCK, RECEIVED CHARACTER FLAG
    00114A 0103     0002  A        SREIX  RMB    2          * FOR RECEIVE BLOCK, DATA STORED ADDRESS
    00115           0105  A        SRWKBT EQU    *          * WORK BOTTOM
    00116                          *
    00117                          *
    00118                          * SERIAL COMMUNICATION ROUTINE
    00119                          *
    00120           0001  A        SOH    EQU    $01        * SOH
    00121           0002  A        STX    EQU    $02        * STX
    00122           0003  A        ETX    EQU    $03        * ETX
    00123           0004  A        EOT    EQU    $04        * EOT
    00124           0005  A        ENQ    EQU    $05        * ENQ
    00125           0006  A        ACK    EQU    $06        * ACK
    00126           0015  A        NAK    EQU    $15        * NAK
    00127           0010  A        DLE    EQU    $10        * DLE
    00128           0038  A        WAK    EQU    $38        * WAK (;)
    00129           0031  A        DEVCRT EQU    $31        * DEVICE NO. (CRT)
    00130                          *
    00131A 1000                           ORG    $1000
    00132                          *
    00133                          *
    00134                          * OUT TO SERIAL ROUTINE
    00135                          *
    00136                          *  INITIALIZATION OF SERIAL
    00137                          *  1. CLEAR FMT, DID, SID, FNC, SIZ WORK
    00138                          *  2. SET 'NAK' CODE TO ACK CHARACTER AREA
    00139                          *  3. SET RETRY COUNT (INITIAL 5)
    00140                          *  4. SET TIME OVER COUNT (INITIAL 0.5 SEC)
    00141                          *  5. SET START BLOCK TIME OVER COUNT (INITIAL 10 SEC)
    00142                          *
    00143                          *
    00144                          * RECEIVE ONE CHARACTER FROM SERIAL PORT
    00145                          * ON ENTRY
    00146                          *  PARAMETER NONE
    00147                          * ON EXIT
    00148                          *  (C): I/O ERROR FLAG  0:OK  1:ERROR
    00149                          *  (V): TIME OVER FLAG  0:OK  1:TIME OVER   (TIME OVER = 0.1 SEC)
    00150                          *  (A): RECEIVED CHARACTER (IF (C)=0 AND (Z)=1)
    00151                          * REGISTER PRESERVE  B,X
    00152                          *
    00153A 1000 B6 01CB  A        SRVSGL LDA A  SRTIMO     * SET TIME OVER COUNTER
    00154                          * ENTRY POINT (PARAMETER  (A):TIME OVER LIMIT)
    00155A 1003 37                 SRVSXX PSH B
    00156A 1004 16                        TAB
    00157A 1005 8D 0D 1014         SRVS40 BSR    SRVBYT     * RECEIVE ONE CHARACTER
    00158A 1007 25 09 1012                BCS    SRVS50     * I/O ERROR ?
    00159A 1009 28 07 1012                BVC    SRVS50     * OK ?
    00160A 100B 5D                        TST B             * TIME OVER LIMIT CHECK
    00161A 100C 27 F7 1005                BEQ    SRVS40
    00162A 100E 5A                        DEC B
    00163A 100F 26 F4 1005                BNE    SRVS40
    00164A 1011 03                        SEV               * TIME OUT
    00165A 1012 33                 SRVS50 PUL B
```

```
00166A 1013 39                        RTS
00167                          *
00168                          *
00169                          * RECEIVE ONE BYTE
00170                          * REGISTER PRESERVE  B,X
00171                          *
00172A 1014 3C                 SRVBYT PSHX
00173A 1015 CE 0BB8   A               LDX    #3000     * 21*1.6*3000 = 100,000
00174A 1018 4F                 SRVS10 CLR A            * (C)<---0, (V)<--- 0            (1 C/S)
00175A 1019 09                        DEX              * NOT RECEIVED, CHECK TIME OVER  (1 C/S)
00176A 101A 0B                        SEV              * PRESET (V)                     (1 C/S)
00177A 101B 27 12 102F                BEQ    SRVS30    *                                (2 C/S)
00178A 101D 0D                        SEC              * PRESET I/O ERROR FLAG          (1 C/S)
00179A 101E 7B B07D   A               TIM    #$B0,MIOSTS *                              (3 C/S)
00180A 1021 26 0C 102F                BNE    SRVS30    *                                (2 C/S)
00181A 1023 7B 0403   A               TIM    #$4,PORT2 *  CONNECTED EXTERNAL SERIAL     (3 C/S)
00182A 1026 26 07 102F                BNE    SRVS30    *                                (2 C/S)
00183A 1028 7D 0011   A               TST    TRCSR     * RECEIVED ?                     (3 C/S)
00184A 102B 2A EB 1018                BPL    SRVS10    *                                (2 C/S)  21 C
00185                          * RECEIVED                * (A) <--- RECEIVED CHARACTER
00186A 102D 96 12     A               LDA A  SRDR      * (C),(V) <--- 0 BY 'TST' INSTRUCTION
00187A 102F 38                 SRVS30 PULX
00188A 1030 39                        RTS
00189                          *
00190                          * WAIT TO BE SELECTED
00191                          *   RECEIVE SEQUENCE
00192                          *   1. WAIT SERIAL IDLING
00193                          *   2. CHECK 'EOT'
00194                          * PARAMETER
00195                          * ON ENTRY
00196                          *  (A):DESTINATION DEVICE  (FOR SENDING SIDE)
00197                          *  (B):SOURCE DEVICE       (FOR SENDING SIDE)
00198                          *  (X):TIME OVER LIMIT (1=0.1 SEC,  0:NO LIMIT)
00199                          * ON EXIT
00200                          *  (C):I/O ERROR FLAG  (0:NORMAL  1:ERROR)
00201                          *  (A):RETURN CODE  (0:NORMAL) ($B3:TIME OUT ERROR)
00202                          *  (Z):DEPEND ON VALUE OF (A)
00203                          *
00204                          * WORK USE AS REGISTER
00205                          *  R0H:31
00206                          *  R0L:DID
00207                          *  R1H:SID
00208                          *  R1L:ENQ
00209                          *
00210                          *  ENTRY POINT: RECEVED 'EOT', CHECK ENQ PATTERN.
00211A 1031 DD 51     A        SRSLET STD    R0L        * SET ENQ PATTERN
00212A 1033 86 05     A               LDA A  #ENQ
00213A 1035 97 53     A               STA A  R1L
00214A 1037 86 31     A               LDA A  #$31
00215A 1039 97 50     A               STA A  R0H        * SET P1
00216A 103B 20 2E 106B                BRA    SRSL18
00217                          *
00218                          *  ENTRY POINT: WAIT 'EOT' 'P1', ... 'ENQ' PATTERN
00219A 103D DD 51     A        SRSLCT STD    R0L        * SET ENQ PATTERN
00220A 103F 86 05     A               LDA A  #ENQ
```

```
ERR   SEQ   LOC  OBJECT      PROGRAM  EPSP       --- SERIAL (EPSP) EXAMPLE ---

    00221A 1041 97 53     A              STA A  R1L
    00222A 1043 86 31     A              LDA A  #$31
    00223A 1045 97 50     A              STA A  R0H        * SET P1
    00224                         *
    00225         1047    A       SRSL10 EQU    *
    00226A 1047 71 FB03   A              AIM    #$FF-$4,PORT2 * SELECT EXTERNAL SERIAL
    00227         104A    A       SRSL11 EQU    *
    00228A 104A 72 0111   A              OIM    #$1,TRCSR * SET WAKE UP FLAG
    00229         104D    A       SRSL13 EQU    *
    00230A 104D 7B 0111   A              TIM    #$1,TRCSR * IDLE ?
    00231A 1050 27 0E 1060               BEQ    SRSL15
    00232A 1052 0D                       SEC
    00233A 1053 7B 0403   A              TIM    #$4,PORT2
    00234A 1056 26 30 1088               BNE    SRSL30     * BROKEN SERIAL ?
    00235A 1058 96 11     A              LDA A  TRCSR      * IGNORE RECEIVED CHARACTER
    00236A 105A 2A F1 104D               BPL    SRSL13
    00237A 105C 96 12     A              LDA A  SRDR
    00238A 105E 20 EA 104A               BRA    SRSL11
    00239A 1060 BD 1014   A       SRSL15 JSR    SRVBYT     * READ CHARACTER
    00240A 1063 25 23 1088               BCS    SRSL30     * I/O ERROR ?
    00241A 1065 29 22 1089               BVS    SRSL40
    00242                         * RECIEVD  'EOT' ?
    00243A 1067 81 04     A              CMP A  #EOT
    00244A 1069 26 1E 1089               BNE    SRSL40
    00245A 106B 5F                SRSL18 CLR B             * (B):RECEIVED COUNTER  DID:0  SID:1
    00246A 106C BD 1014   A       SRSL20 JSR    SRVBYT
    00247A 106F 29 18 1089               BVS    SRSL40     * TIME OVER ?
    00248A 1071 25 15 1088               BCS    SRSL30     * I/O ERROR ?
    00249A 1073 3C                       PSHX
    00250A 1074 CE 0050   A              LDX    #R0
    00251A 1077 3A                       ABX
    00252A 1078 A1 00     A              CMP A  0,X
    00253A 107A 38                       PULX
    00254A 107B 26 0C 1089               BNE    SRSL40
    00255A 107D 5C                       INC B
    00256A 107E C1 04     A              CMP B  #4         * RECEIVED  00 'DID' 'SID' 'ENQ' ?
    00257A 1080 26 EA 106C               BNE    SRSL20
    00258                         * RECEIVED 'ACK' SEQUENCE'
    00259A 1082 86 06     A              LDA A  #ACK
    00260A 1084 BD 1098   A              JSR    SSRSGL
    00261A 1087 4F                       CLR A
    00262A 1088 39                SRSL30 RTS
    00263                         * TIME OUT ?
    00264A 1089 8C 0000   A       SRSL40 CPX    #0         * CHECK TIME OUT ?
    00265A 108C 27 B9 1047               BEQ    SRSL10
    00266A 108E 09                       DEX
    00267A 108F 26 B6 1047               BNE    SRSL10
    00268A 1091 86 33     A              LDA A  #$33       * TIME OUT ERROR RETURN
    00269A 1093 7F 01C5   A              CLR    SRDDEV
    00270A 1096 4D                       TST A
    00271A 1097 39                       RTS
    00272                         *
    00273                         * SEND ONE CHARACTER TO SERIAL PORT
    00274                         * ON ENTRY
    00275                         *         (A): SEND CHARACTER
```

```
00276                            * ON EXIT
00277                            *       (C): I/O BREAK FLAG   0:OK   1:ERROR
00278                            * REGISTER PRESERVE  ALL
00279                            *
00280             1098   A       SSRSGL EQU     *
00281A 1098 0D                  SSRS10 SEC              * PRESET I/O ERROR FLAG
00282A 1099 7B 807D   A                TIM     #$80,MIOSTS * BREAK ?
00283A 109C 26 0D 10AB                 BNE     SSRS20
00284A 109E 7B 0403   A                TIM     #$4,PORT2 * CONNECT TO EXTERNAL SERIAL ?
00285A 10A1 26 08 10AB                 BNE     SSRS20
00286A 10A3 7B 2011   A                TIM     #$20,TRCSR * SEND READY ?
00287A 10A6 27 F0 1098                 BEQ     SSRS10
00288A 10A8 97 13     A                STA A   STDR
00289A 10AA 0C                         CLC              * OK RETURN
00290A 10AB 39                  SSRS20 RTS
00291                            * SERIAL INITIALIZE
00292                            *  ON ENTRY
00293                            *  (A):MODE  (MASTER:0  SLAVE:NONZERO)
00294                            *
00295A 10AC 36                  SRINIT PSH A
00296A 10AD CC 0011   A                LDD     #SRWKBT-SRWKTP * CLEAR WORK (A):PATTERN, (B):COUNT
00297A 10B0 CE 01C4   A                LDX     #SRWKTP
00298A 10B3 3C                         PSHX
00299A 10B4 A7 00     A         CLRB   STA A   0,X
00300A 10B6 08                         INX
00301A 10B7 5A                         DEC B
00302A 10B8 26 FA 10B4                 BNE     CLRB
00303                            *
00304A 10BA 38                         PULX
00305A 10BB 32                         PUL A
00306A 10BC A7 0A     A                STA A   SRMODE-SRWKTP,X * SET MASTER/SLAVE MODE
00307A 10BE 62 0306   A                OIM     #3,SRTRCN-SRWKTP,X * SEND RETRY COUNT = 3
00308A 10C1 CC 0A64   A                LDD     #10*256+100
00309A 10C4 ED 07     A                STD     SRTIMO-SRWKTP,X * TIME OVER LIMIT = 1 SEC
00310                            *                         * RECEIVE BLOCK TIME OVER LIMIT=1
00311A 10C6 A7 09     A                STA A   SRATMO-SRWKTP,X * RECEIVE ACK TIME OVER = 1 SEC
00312A 10C8 6C 08     A                INC     SRETDL-SRWKTP,X * AFTER 'EOT', IDLING TIME
00313                            *
00314A 10CA 39                         RTS
00315                            *
00316                            *
00317                            * RECEIVE FROM SERIAL (FOR SLAVE DEVICE)
00318                            *  ON ENTRY
00319                            *   (X):RECEIVED DATA STORED ADDRESS
00320                            *  ON EXIT
00321                            *   (A):RETURN CODE   0:OK  $80:TIME OVER   $82:RECEIVE ERROR
00322                            *                   $88: RECEIVED 'EOT'
00323                            *   (B): 0:RECEIVED WITH HEADER   1:RECEIVED WITHOUT HEADER
00324                            *        (EFFECTIVE (A)=0)
00325                            *   (Z):DEPEND ON VALUE OF (A)
00326                            *
00327                            *  WORK USE AS REGISTER
00328                            *   R0H:TOP CHARACTER OF BLOCK ($1 OR $2)
00329                            *   R1:ADDRESS OF STORED DATA
00330                            *   R2L:BLOCK LENGTH
```

```
    00331                                *   R3H:RETRY COUNT
    00332                                *   R3L:TIME OVER COUNTER
    00333                                *   R4H:TOP CHARACTER OF BLOCK
    00334                                *   R4L:OMITTED HEADER FLAG (0:NOT  1:OMITTED)
    00335                                *
    00336                                *
    00337                                *
    00338                                *
    00339                                * ERROR RETURN ROUTINE
    00340                                *   SET ERROR CODE TO (A), CLEAR (C)
     0341                                *
    00342A 10CB 86 80     A              SRER80 LDA A  #$80       * ERROR $80 (TIME OVER)
    00343A 10CD 20 0A 10D9                      BRA    SRER10
    00344                                *
    00345A 10CF 86 31     A              SRER81 LDA A  #$81       * ERROR $81  (
    00346A 10D1 20 06 10D9                      BRA    SRER10
    00347A 10D3 86 88     A              SRER88 LDA A  #$88       * ERROR $31  (RECEIVED 'EOT')
    00348A 10D5 20 02 10D9                      BRA    SRER10
    00349                                *
    00350A 10D7 86 83     A              SRER83 LDA A  #$83       * ERROR 83 (DRIVER OFF)
    00351A 10D9 7F 01C5   A              SRER10 CLR    SRDDEV     * CLEAR DID (FOR START FROM ENQ PROCESS)  (C):
    00352             100C   A           SRER20 EQU    *
    00353A 10DC 7E 117F   A                     JMP    SRR390
    00354                                *
    00355                                *
    00356                                * EPSP  RECEIVE (SLAVE DEVICE) SUBROUTINE
    00357                                * RECEIVE FROM SERIAL
    00358                                *  ON ENTRY
    00359                                *   (X):RECEIVED DATA STORED ADDRESS
    00360                                *
    00361A 10DF DF 52     A              SERRCV STX    R1
    00362A 10E1 7B 107A   A                     TIM    #$10,SRSTS * DRIVER ON ?
    00363A 10E4 27 F1 10D7                      BEQ    SRER83
    00364                                *
    00365                                *  SELECT SERIAL (DETATCH SLAVE)
    00366A 10E6 96 11     A                     LDA A  TRCSR      * SAVE TRCSR FOR RECOVER RS232
    00367A 10E8 97 5B     A                     STA A  R5L
    00368                                *
    00369             10EA   A           SERINS EQU    *
    00370A 10EA 0F                              SEI
    00371A 10EB 71 EF11   A                     AIM    #$FF-$10,TRCSR * SERIAL INTERRUPT DISABLE
    00372A 10EE 71 FB03   A                     AIM    #$FF-$4,PORT2
    00373             10F1   A           INSR05 EQU    *
    00374A 10F1 4F                              CLR A
    00375A 10F2 4C                              INC A
    00376A 10F3 97 59     A              SRR810 STA A  R4L        * OMITTED HEADER BLOCK (INITIAL)
    00377A 10F5 86 01CA   A              SRR820 LDA A  SRTRCN     * SET RETRY COUNT
    00378A 10F8 97 56     A                     STA A  R3H
    00379                                * RECEIVE FIRST CHARACTER
    00380A 10FA DE 52     A              SRR830 LDX    R1         * (X): STORED DATA ADDRESS
    00381A 10FC 86 01CC   A                     LDA A  SRETMO     * SET TIME OVER FOR WAITING BLOCK
    00382A 10FF BD 1003   A                     JSR    SRVSXX
    00383A 1102 25 D8 100C                      BCS    SRER20
    00384A 1104 29 C5 10C8                      BVS    SRER80     * TIME OVER ERROR ?
    00385A 1106 C6 04     A                     LDA B  #4         * (B): BLOCK SIZE (PRESET FOR HEADER BLOCK)
    00386A 1108 81 01      A              CMP A  #SOH
    00387A 110A 27 2E 113A                BEQ    SRRB50
    00388A 110C 5C                        INC B             * (B): 5
    00389A 110D 3A                        ABX               * (X): DATA STORED ADDRESS
    00390A 110E F6 01C8    A              LDA B  SRSIZ      * (B): BLOCK SIZE (FOR DATA BLOCK)
    00391A 1111 81 02      A              CMP A  #STX
    00392A 1113 27 25 113A                BEQ    SRRB50
    00393A 1115 81 05      A              CMP A  #ENQ
    00394A 1117 27 19 1132                BEQ    SRCE10
    00395A 1119 81 04      A              CMP A  #EOT       * EOT ?
    00396A 111B 27 B6 10D3                BEQ    SRERB8
    00397                          *
    00398                          * OTHER CODES (SKIP CURRENT BLOCK AND SEND 'NAK')
    00399A 111D BD 1014    A       SRRB40 JSR    SRVBYT     * RECEIVED ONE CHARACTER ?
    00400A 1120 25 5D 117F                BCS    SRRB90
    00401A 1122 28 F9 111D                BVC    SRRB40
    00402                          * TIME OVER (NOT RECEIVED DATA 0.1 SEC)
    00403                          *
    00404                          * ERROR 'NAK' SEND
    00405A 1124 86 15      A       SRCSER LDA A  #NAK
    00406A 1126 B7 01C9    A              STA A  SRACKC     * SET NAK CHARACTER FOR 'ENQ'
    00407A 1129 D6 56      A              LDA B  R3H        * RETRY COUNT CHECK
    00408A 112B 27 A2 10CF                BEQ    SRERB1
    00409A 112D 7A 0056    A              DEC    R3H
    00410A 1130 27 9D 10CF                BEQ    SRERB1
    00411                          * ENTRY FROM 'ENQ'
    00412A 1132 B6 01C9    A       SRCE10 LDA A  SRACKC
    00413A 1135 BD 1098    A              JSR    SSRSGL     * SEND NAK
    00414A 1138 20 C0 10FA                BRA    SRRB30
    00415                          *
    00416                          *
    00417                          * RECEIVE DATA BLOCK (SOH....  OR  STX....)
    00418                          *
    00419A 113A 5C                 SRRB50 INC B
    00420A 113B D7 55      A              STA B  R2L
    00421A 113D 97 58      A              STA A  R4H        * R4H:RECIEVD FIRST CHARACTER
    00422A 113F 97 50      A              STA A  R0H        * R0H:$1 (SOH) OR $2 (STX)
    00423A 1141 16                        TAB               * (B):CHECKSUM
    00424                          * RECEIVE DATA STRING LOOP
    00425A 1142 BD 1000    A       SRRB70 JSR    SRVSGL
    00426A 1145 25 38 117F                BCS    SRRB90
    00427A 1147 29 DB 1124                BVS    SRCSER     * TIME OVER ?
    00428A 1149 A7 00      A              STA A  0,X
    00429A 114B 08                        INX
    00430A 114C 1B                        ABA
    00431A 114D 16                        TAB
    00432A 114E 7A 0055    A              DEC    R2L
    00433A 1151 26 EF 1142                BNE    SRRB70
    00434                          *
    00435A 1153 BD 1000    A       SRRB75 JSR    SRVSGL     * RECEIVE CHECKSUM
    00436A 1156 25 27 117F                BCS    SRRB90
    00437A 1158 29 CA 1124                BVS    SRCSER
    00438A 115A 1B                        ABA
    00439A 115B 16                        TAB
    00440A 115C 7A 0050    A              DEC    R0H        * IF STX..., RECEIVE 'ETX>
```

```
 00441A 115F 26 F2 1153          BNE    SRRB75
 00442A 1161 5D                  TST B          * CHECKSUM OK ?
 00443A 1162 26 C0 1124          BNE    SRCSER
 00444                     *
 00445A 1164 86 06     A         LDA A  #ACK
 00446A 1166 B7 01C9   A         STA A  SRACKC    * SAVE SEND ACK CODE FOR 'ENQ'
 00447A 1169 BD 1098   A         JSR    SSRSGL
 00448A 116C 25 11 117F          BCS    SRRB90
 00449A 116E DC 58     A         LDD    R4        * R4H<--- FIRST CHARACTER OF BLOCK, (B):MODE
 00450A 1170 88 01     A         EOR A  #SOH
 00451A 1172 26 0A 117E          BNE    SRRB80    * IF 'SOH', RECEIVED HEADER BLOCK  (A=0)
 00452                     *
 00453A 1174 DE 52     A         LDX    R1        * SET COUNTER
 00454A 1176 E6 04     A         LDA B  SRSIZ-SRFMT,X
 00455A 1178 F7 01C8   A         STA B  SRSIZ
 00456A 117B 7E 10F3   A         JMP    SRRB10
 00457                     * COMPLETED TO RECEIVE DATA BLOCK
 00458                     *
 00459A 117E 4F            SRRB80 CLR A
 00460A 117F 0E            SRRB90 CLI
 00461A 1180 DE 52     A         LDX    R1
 00462A 1182 71 FB7D   A         AIM    #$FF-$4,MIOSTS * STATUS, STOP SERIAL COMMUNICATION
 00463                     * RECOVER RS232  (NOT CHANGE C)
 00464A 1185 7B B07D   A         TIM    #$B0,MIOSTS * BROKEN ?
 00465A 1188 27 01 118B          BEQ    SRRB9A    * NOTE. AFTER 'CLI' INSTRUCTION, 'BREAK'
 00466A 118A 0D                  SEC              *  MAY BE CAUSED.
 00467                     *
 00468             118B  A SRRB9A EQU   *
 00469A 118B 7B 037A   A         TIM    #$3,SRSTS * ON RS232 READ RUNNING ?
 00470A 118E 27 0E 119E          BEQ    SRRB92
 00471                     * WAIT 250 MICRO SEC (FOR SERIAL TERMINAL TO RECEIVE CHARACTER)
 00472A 1190 36                  PSH A
 00473A 1191 86 32     A         LDA A  #50
 00474A 1193 4A            SRRB91 DEC A
 00475A 1194 26 FD 1193          BNE    SRRB91
 00476A 1196 72 0403   A         OIM    #$4,PORT2 * SELECT SERIAL SLAVE CPU
 00477A 1199 96 5B     A         LDA A  R5L       * RECOVER TRCSR
 00478A 119B 97 11     A         STA A  TRCSR
 00479A 119D 32                  PUL A
 00480                     *
 00481A 119E 8A 00     A   SRRB92 ORA A #$0       * FOR RECOVER (Z) (UNCHANGE C)
 00482A 11A0 39                  RTS
 00483                     *
 00484                     *


 00486                     *
 00487                     * PROGRAM OF SENDING SIDE (MAIN DEVICE)
 00488                     *  GET CHARACTERS FROM KEYBOARD AND SEND BY EPSP.
 00489                     *
 00490                     *
 00491                     *
 00492A 11A1 86 01     A   OPNBIS LDA A #1        * DRIVER ON
 00493A 11A3 BD FF73   A         JSR    SERONF
```

```
  00494A 11A6 CE 12B8  A               LDX     #SCRPSD    * SET SCREEN PACKET  X:DATA ADDRESS
  00495A 11A9 C6 02    A               LDA B   #SCRPSE-SCRPSD * (B):NUBER OF DATA
  00496A 11AB A6 00    A        INIT10 LDA A   0,X
  00497A 11AD A7 08    A               STA A   SCRPK1-SCRPSD,X
  00498A 11AF 08                       INX
  00499A 11B0 5A                       DEC B
  00500A 11B1 26 F8 11AB               BNE     INIT10
  00501                         *
  00502A 11B3 CE 12C0  A               LDX     #SCRPK1    * INITIALIZE SCREEN
  00503A 11B6 BD FF5E  A               JSR     SCRFNC     * SELECT SCREEN DEVICE (DISPLAY CONTROLER)
  00504                         *
  )0505A 11B9 BD FF9A  A        REPEAT JSR     KEYIN
  00506A 11BC 25 0C 11CA               BCS     BRKRTN
  00507A 11BE B7 1231  A               STA A   BUF
  00508A 11C1 4F                       CLR A
  00509A 11C2 CE 122C  A               LDX     #SNDPKT    * SERIAL TRANSMITTE
  00510A 11C5 BD FF70  A               JSR     SEROUT
  00511                         *
  00512A 11C8 20 EF 11B9               BRA     REPEAT
  00513A 11CA 39                BRKRTN RTS
  00514                         *
  00515                         *
  00516                         *


  00518                         *
  00519                         *
  00520                         * PROGRAM OF RECEIVING SIDE (SLAVE DEVICE)
  00521                         *  GET CHARACTERS FROM EPSP AND DISPLAY ON THE VIRTUAL SCREEN.
  00522                         *
  00523                         *
  00524                         *
  00525A 11CB 86 01    A        RECSID LDA A   #1         * DRIVER ON
  00526A 11CD BD FF73  A               JSR     SERONF
  )0527A 11D0 86 01    A               LDA A   #1         * SERIAL MASTER/SLAVE MODE = SLAVE
  00528A 11D2 B7 01CE  A               STA A   SRMODE
  00529A 11D5 CE 132E  A               LDX     #SCRPRD    * SET SCREEN PACKET  X:DATA ADDRESS
  00530A 11D8 C6 0E    A               LDA B   #SCRPRE-SCRPRD * (B):NUBER OF DATA
  00531A 11DA A6 00    A        RECS10 LDA A   0,X
  00532A 11DC A7 92    A               STA A   SCRPK1-SCRPRD,X
  00533A 11DE 08                       INX
  00534A 11DF 5A                       DEC B
  00535A 11E0 26 F8 11DA               BNE     RECS10
  00536                         *
  00537A 11E2 CE 12C0  A               LDX     #SCRPK1    * INITIALIZE SCREEN
  00538A 11E5 BD FF5E  A               JSR     SCRFNC     * SELECT SCREEN DEVICE
  00539A 11E8 CE 12C8  A               LDX     #SCRPK2
  00540A 11EB BD FF5E  A               JSR     SCRFNC     * SET SCREEN SIZE AND BUFFER ADDRESS
  00541A 11EE CE 12CD  A               LDX     #SCRPK3    *
  00542A 11F1 BD FF5E  A               JSR     SCRFNC     * SET CURSOR MARGIN
  00543A 11F4 CE 12CF  A               LDX     #SCRPK4    *
  00544A 11F7 BD FF5E  A               JSR     SCRFNC     * SET SCROLL STEP
  00545A 11FA CE 12D2  A               LDX     #SCRPK5    *
  00546A 11FD BD FF5E  A               JSR     SCRFNC     * SET SCROLL SPEED
```

```
00547                        * DEVICE : TREAT AS DISPLAY CONTROLLER.
00548A 1200 CC 3020   A             LDD     #$3020     * WAIT TO BE EPSP SELECTED
00549A 1203 CE 0000   A             LDX     #0
00550A 1206 BD 103D   A             JSR     SRSLCT
00551A 1209 25 BF 11CA       RCVR10 BCS     BRKRTN
00552                        *
00553A 120B CE 1233   A      RCVRPT LDX     #RCVPKT    * RECEIVE DATA
00554A 120E BD 10DF   A             JSR     SERRCV
00555A 1211 25 B7 11CA              BCS     BRKRTN
00556A 1213 27 0F 1224              BEQ     RCVR20     * ERROR ?
00557A 1215 81 B8     A             CMP A   #$B8       * RECEIVED 'EOT' ?
00558A 1217 26 F2 120B              BNE     RCVRPT
00559A 1219 CC 3020   A             LDD     #$3020     * WAIT TO BE EPSP SELECTED
00560A 121C CE 0000   A             LDX     #0
00561A 121F BD 1031   A             JSR     SRSLET
00562A 1222 20 E5 1209              BRA     RCVR10
00563                        *
00564A 1224 B6 1238   A      RCVR20 LDA A   RCVPKT+5   * DISPLAY RECEIVED CHARACTERS ON THE VIRTUAL
00565A 1227 BD FF4F   A             JSR     DSPSCR     *  SCREEN (LCD)
00566A 122A 20 DF 120B              BRA     RCVRPT
00567                        *
00568                        * PACKET OF SEND DATA STRING
00569                        *
00570A 122C    00     A      SNDPKT FCB     $0         * FORMAT
00571A 122D    30     A             FCB     $30        * SID (DISPLAY CONTROLLER)
00572A 122E    20     A             FCB     $20        * DID (HC-20)
00573A 122F    92     A             FCB     $92        * FUNCTION
00574A 1230    00     A             FCB     0          * DATA LENGTH
00575A 1231    00     A      BUF    FCB     0          * DATA
00576A 1232    00     A             FCB     0
00577                        *
00578                        * PACKET OF RECEIVE DATA STRING
00579                        *
00580A 1233    00     A      RCVPKT FCB     $0         * FORMAT
00581A 1234    30     A             FCB     $30        * SID (DISPLAY CONTROLER)
00582A 1235    20     A             FCB     $20        * DID (HC-20)
00583A 1236    92     A             FCB     $92        * FUNCTION
00584A 1237    00     A             FCB     0          * DATA LENGTH
00585A 1238    0080   A             RMB     128        * DATA
00586                        *
00587                        * SCREEN PACKET FOR SENDING SIDE
00588A 12B8    84     A      SCRPSD FCB     $84        * SCREEN DEVICE SELECT (DISPLAY CONTROLER)
00589A 12B9    30     A             FCB     $30
00590                        *
00591          12BA   A      SCRPSE EQU     *
00592                        *
00593                        *
00594                        * WORK AREA
00595A 12BA    0006   A             RMB     6
00596A 12C0    84     A      SCRPK1 FCB     $84        * SELECT SCREEN DEVICE
00597A 12C1    22     A             FCB     $22
00598A 12C2    0006   A             RMB     6
00599A 12C8    87     A      SCRPK2 FCB     $87        * SET SCREEN SIZE AND BUFFER ADDRESS
00600A 12C9    13     A             FCB     19,3
     A 12CA    03     A
```

```
ERR   SEQ    LOC  OBJECT       PROGRAM  EPSP        --- SERIAL (EPSP) EXAMPLE ---

    00601A 12CB     12D4   A                FDB     SCRBUF
    00602                            *
    00603A 12CD     C3     A         SCRPK3 FCB     $C3        * SET CURSOR MARGIN
    00604A 12CE     04     A                FCB     4
    00605                            *
    00606A 12CF     C4     A         SCRPK4 FCB     $C4        * SET SCROLL STEP
    00607A 12D0     0A     A                FCB     10         * X
    00608A 12D1     03     A                FCB     3          * Y
    00609                            *
    00610A 12D2     CB     A         SCRPK5 FCB     $CB        * SET SCROLL SPEED
    00611A 12D3     09     A                FCB     9
    00612                            *
    00613A 12D4     005A   A         SCRBUF RMB     90
    00614                            * SCREEN PACKET FOR RECEIVING SIDE
    00615A 132E     84     A         SCRPRD FCB     $84        * SCREEN DEVICE SELECT (LCD)
    00616A 132F     22     A                FCB     $22
    00617                            *
    00618A 1330     87     A                FCB     $87        * SET SCREEN SIZE AND BUFFER ADDRESS
    00619A 1331     13     A                FCB     19,3
         A 1332     03     A
    00620A 1333     12D4   A                FDB     SCRBUF
    00621                            *
    00622A 1335     C3     A                FCB     $C3        * SET CURSOR MARGIN
    00623A 1336     04     A                FCB     4
    00624                            *
    00625A 1337     C4     A                FCB     $C4        * SET SCROLL STEP
    00626A 1338     0A     A                FCB     10         * X
    00627A 1339     03     A                FCB     3          * Y
    00628                            *
    00629A 133A     CB     A                FCB     $CB        * SET SCROLL SPEED
    00630A 133B     09     A                FCB     9
    00631                            *
    00632           133C   A         SCRPRE EQU     *
    00633                            *
    00634                            *
    00635           0000   A                END
***** TOTAL ERRORS      0
```