CHAPTER 5

5.1 General

The RS-232C port performs communication by the start-stop synchronization method (refer to the description of serial communication in Chapter 4). Generation of the TXD binary signal and read of the RXD binary signal are performed by software. The master MCU transmits data (TXD) and the slave MCU receives data (RXD). The slave MCU receives 1 character of data which it sends to the master MCU via the SCI. The master MCU then uses an SCI interrupt to store this data in the receive buffer (Fig. 5-1).

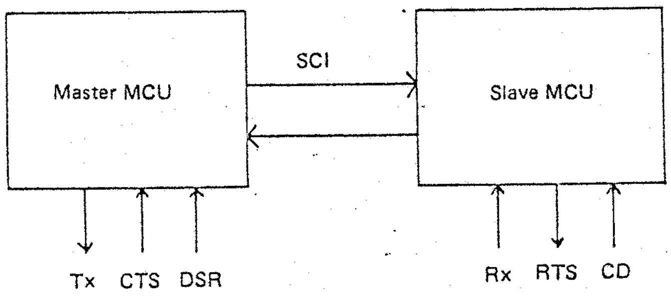


Fig. 5-1 Assignment of RS-232C Functions

5.2 Data Transmission Method

TXD is controlled by port P21 of the master MCU. When a value is set in the OCR and the OCF is set to 1, the value of the OLVL (bit 0 of TCSR) is output from P21 (Fig. 5-2).

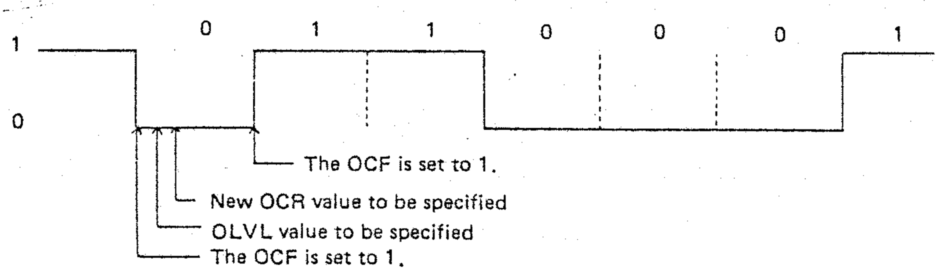


Fig. 5-2 Timing of TXD Transmission

5.3 Data Reception Method

Receive data is input to port P20 of the slave MCU. Input of a start bit in P20 is monitored.

The value of FRC when it takes the value specified by IEDG (bit 1 of TCSR) is set in ICR and this is used to measure the timing of the start bit. Based on this, the calculated center of each pulse is sampled to obtain one character of data (Fig. 5-3).

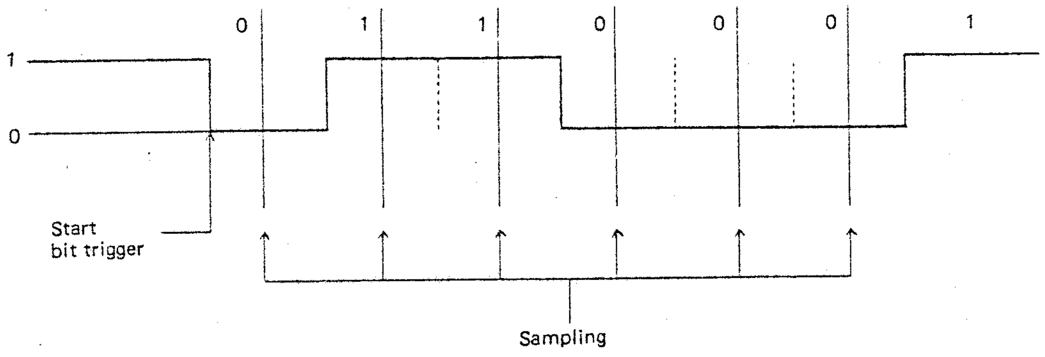


Fig. 5-3 Sampling of Receive Data

One character of data is then transmitted to the master MCU via the SCI (Fig. 5-4).
The master MCU enables receive interrupt by the SCI. The SCI receive interrupt routine stores the receive data in the receive buffer. When the buffer becomes full, an error flag is set and data received subsequent to this will be discarded. The slave MCU cancels input of data through the RS-232C port when a command is sent to it from the master MCU.

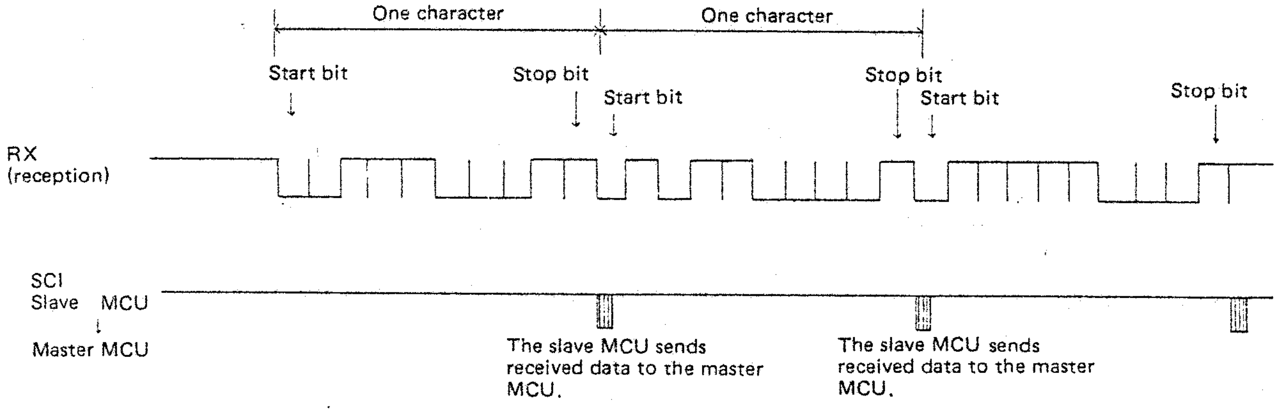


Fig. 5-4 Timing of Data Reception

5.4 Data Communication
Data communication via the RS-232C port is performed by the following procedures.

(1) Setting Parameters
Values for bit rate, word length, parity bit, stop bit length, CD, RTS, DSR detection, are specified by subroutine RSMST. This subroutine specifies the values for constants used in data communication in the I/O work area.

(2) Driver ON

Subroutine RSONOF turns the RS-232C driver ON. When the driver is turned ON, both RTS and TXD go low (RTS is turned OFF and TXD becomes logic 1). A 10-bit preamble (logic 1) is then output. DTR is directly connected to the driver power and therefore goes high (ON) when the driver is turned ON.

(3) Receive buffer open

The receive buffer in the master MCU is opened by subroutine RSOPEN. Once the receive buffer has been opened, the slave MCU begins sending data. The RTS value is set to the value specified procedure (1) above.

(4) Input of one character

Data is fetched from the receive buffer using subroutine RSGET. The data received by the slave MCU is stored in the receive buffer during SCI interrupt processing.

(5) Output of one character

Subroutine RSPUT outputs one character of data. Note that no buffer is used when outputting data.

(6) Termination of data reception

Subroutine RSCLOS terminates RS-232C data reception.

(7) Driver OFF

RSONOF is used to turn the RS-232C driver OFF.

5.5 Notes on I/O Open Condition

The main MCU enable SCI interrupt during RS-232C reception. When the SCI port is accessed directly, the SCI interrupt must be disabled. When the slave MCU receives new data from the SCI port, it cancels data reception from the RS-232C port. The master MCU uses subroutine SNSCOM to send a command to the slave MCU during RS-232C reception and calls subroutine CHKRS (resumption of the interrupted RS-232C data reception) upon completion of transmission of the command.

5.6 Bit Rate Setting

Subroutine RSMST is used to set bit rates for RS-232C transmission (110, 150, 300, 600, 1200, 2400, 4800 and 9600 BPS). To set a transmission speed other than one of those listed above, RSMST must be called and the desired bit rate set directly in variable RSBAUD (01AF, 01B0). This 2-byte variable indicates the number of MCU clock pulses and is set at 1000_{16} for a bit rate of 150 BPS. A bit rate of 75 BPS is therefore obtained by setting 2000_{16} in variable RSBAUD. Note that this value is used directly by the transmission subroutine so the bit rate will change as soon as the value of RSBAUD is altered.

5.7 RTS Operation and Carrier Detection

When using a half-duplex MODEM, the RTS output must be changed and the carrier ON/OFF must be detected. Both RTS and the carrier ports are connected to the slave MCU. RTS control and CD detection are performed by the procedures described below.

(1) RTS Control

Method 1: Subroutine RSOPEN

RTS is set when reception is opened by subroutine RSOPEN. Reception is temporarily closed (subroutine RSCLOS) and the appropriate parameters are set by subroutine RSMST (the previously set parameters remain effective if this is not performed). Reception is then reopened by subroutine RSOPEN. (Fig. 3-5).

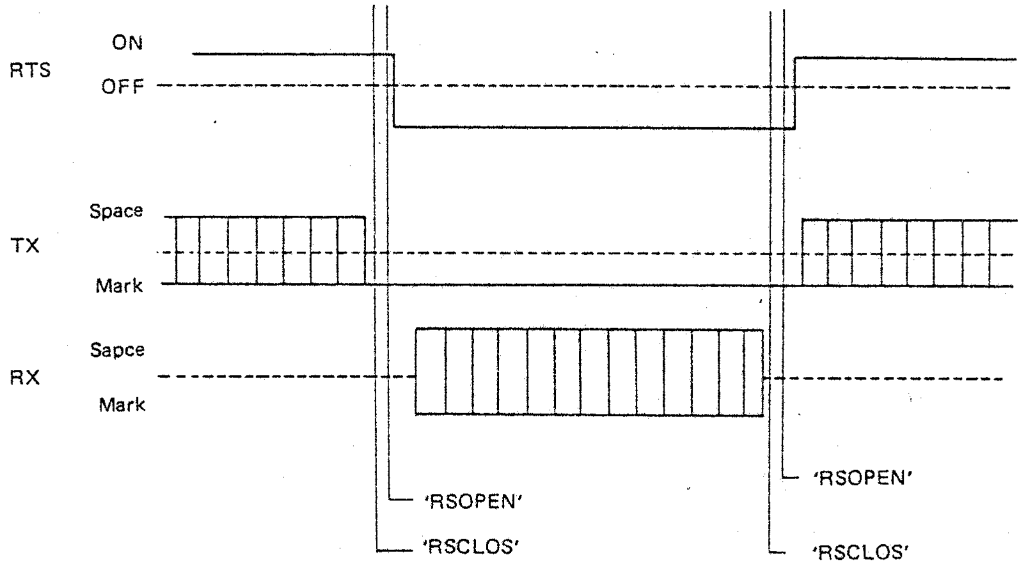


Fig. 5-5 RTS Control (1)

Method 2: Slave MCU command

When performing half-duplex communication, RTS is normally turned ON while data is being transmitted and turned OFF when data is being received. Command 4D, sent to the slave MCU, controls the RTS. This command should be used to turn RTS ON before the start of data transmission. RTS should be turned OFF to open reception (Fig. 5-5).

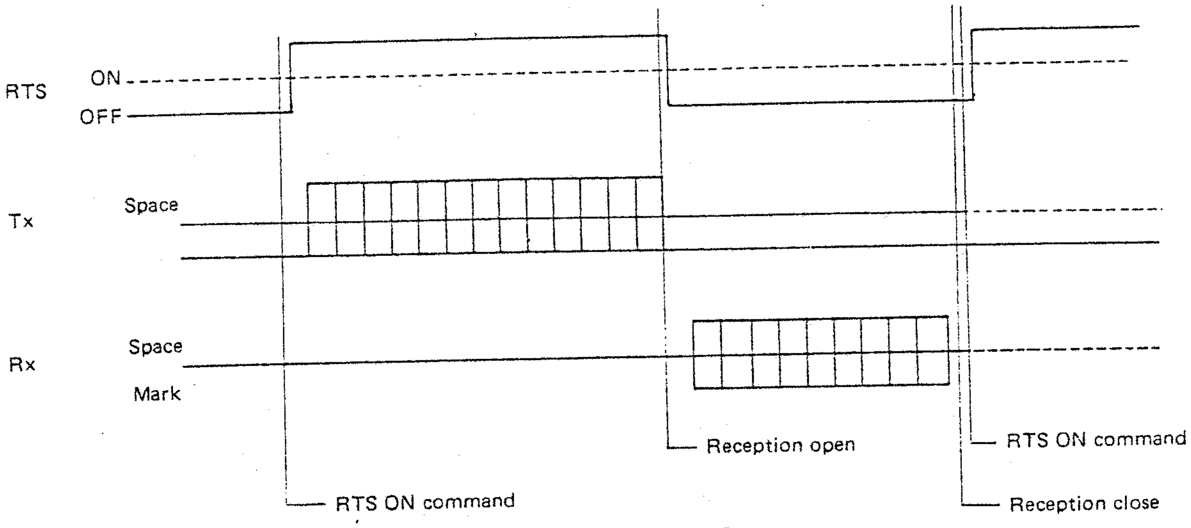


Fig. 5-6 RTS Control (2)

(2) Carrier detection

When the reception is opened, the carrier status is set in port P12 of the master MCU (port P47 of the slave MCU actually detects the carrier status but this data is set in port P12 of the master MCU by software). When the carrier is OFF, P12 of the master MCU is set to 1. When the carrier is ON, P12 is set to 0. Note that after reception has been opened, if carrier OFF status has been detected, carrier ON will cause data reception to start but P12 will not become 0.

The system waits for carrier ON by the following two methods.

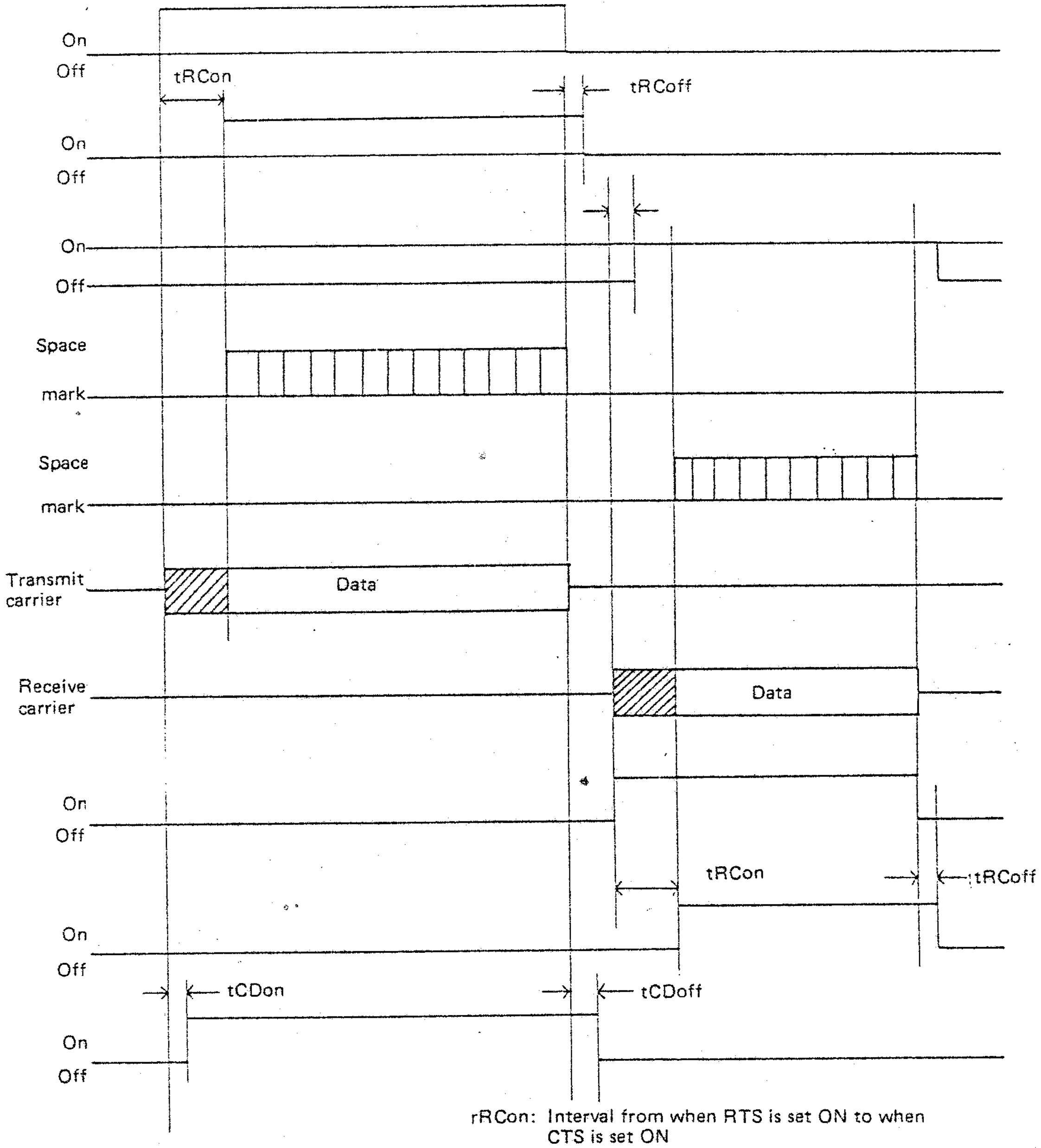
Method 1: If P12 is 1 when the reception is opened, reception is closed and then reopened. This is repeated until carrier ON is detected.

Method 2: Command 80, which sets the value of the slave MCU port in port P12 of the master MCU, is executed for the slave MCU until the carrier is set ON (P12 is set to 0). Reception is then opened.

5.8 Communications Using a MODEM

When using a MODEM, in addition to the data lines for transmission and reception, the control lines must be operated. Fig. 7 shows the timing for a 1200-BPS, half-duplex MODEM.

When data communication is performed as shown in this figure, RTS control as well as CTS and CD detection must be confirmed.



rRCon: Interval from when RTS is set ON to when CTS is set ON
tRCoff: Interval from when RTS is set OFF to when CTS is set OFF
COon: Interval from when the carrier is received to when the CD is set ON
COoff: Interval from when the carrier is turned OFF to when the CD is set OFF

Fig. 5-7 Timing of 1200-BPS, Half-duplex MODEM

The reception routine provides a mode in which data can be received even if no carrier has been detected. If the carrier OFF state is not of great importance, the reception can be opened in this mode and the carrier ignored.

1200-BPS reverse channels
A 1200-BPS MODEM may use a 75-BPS reverse channel. This is performed by the following two procedures.

(i) 1200-BPS transmission and 75-BPS reception. This is enabled by opening reception (RSOPEN) at 75 BPS and then setting the mode (RSMOD) at 1200 BPS.

(ii) 1200-BPS reception and 75-BPS transmission
Reception is opened at 1200 BPS and the bit rate is set to 75 BPS (2000_{16} in variable RSBAUD).

Since master MCU interrupt is disabled during data transmission, data received at this time will be lost as shown in Fig. 5-8.

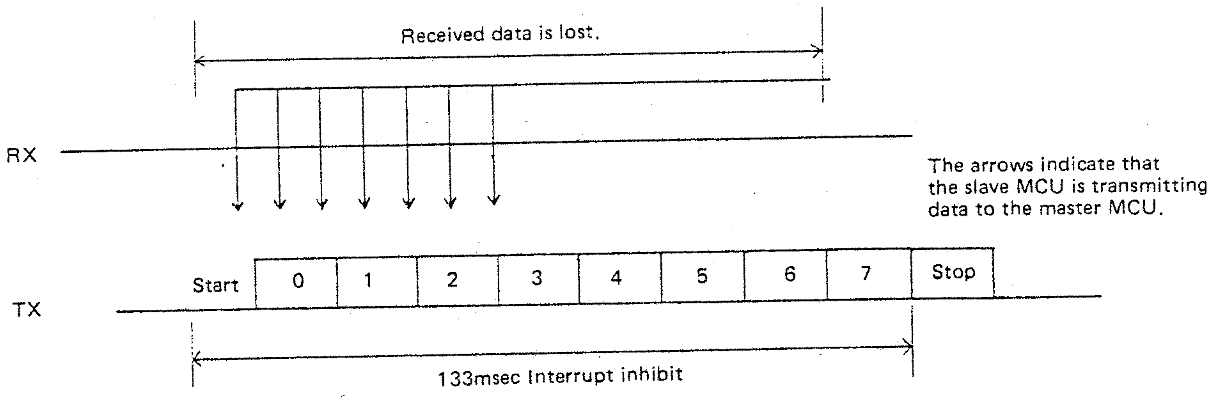

Received data is lost.
RX
The arrows indicate that the slave MCU is transmitting data to the master MCU.
TX
Start
0
1
2
3
4
5
6
7
Stop
133msec Interrupt inhibit

Fig. 5-8 Full-duplex Communication at 1200 and 75 BPS

To protect receive data, the data transmission routine in which interrupt inhibit instruction SEI is omitted must be used. (See end of this chapter.)

5.9 Cautions For Serial Driver ON/OFF

(1) When the Driver is Turned ON
Signal rise may be unstable when the driver is turned ON as shown in Fig. 5-9.

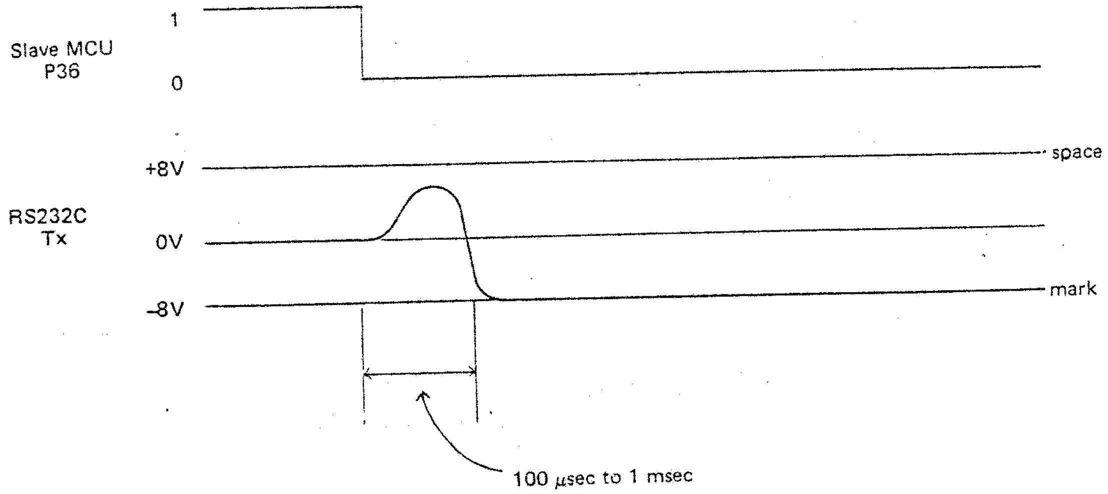


Fig. 5-9 Voltage Change when Driver is Turned ON

In this case, the receiving side may receive incorrect data because it interprets the space state when the driver is turned ON as the start bit.

(2) When the Driver is Turned OFF

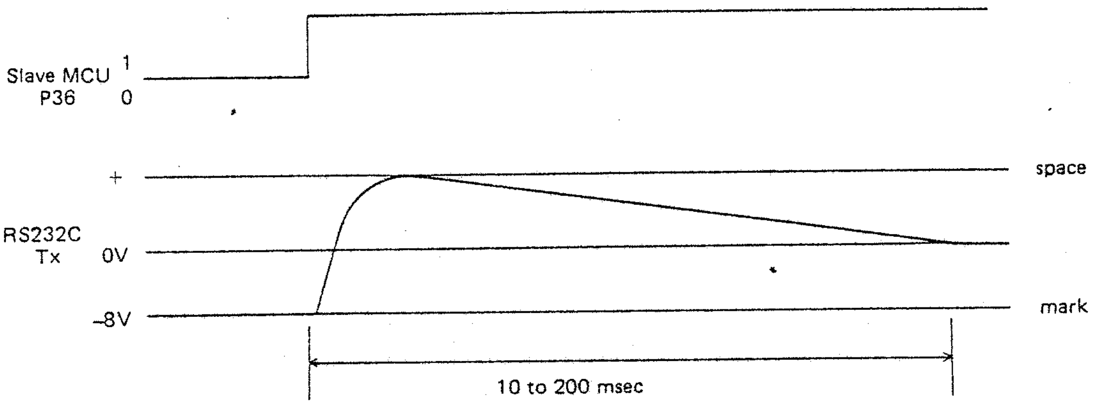


Fig. 5-10 Voltage Change when Driver is Turned OFF

The voltage may change as shown in Fig. 5-10 when the driver is turned OFF. Again, the receiving side may interpret the resulting several tens or hundreds of bits of space states as data, resulting in erroneous data reception.
The driver is turned OFF when the input through the RS-232C port is closed in BASIC. Turn the serial driver ON if you wish to leave the driver on after the RS-232C output is closed. (In terms of software, the serial and RS-232C driver are treated as separate elements. Therefore the driver will only be turned OFF when both drivers are set to OFF from software.)
Press the BREAK key and check the contents of bit 7 of address 7A. When bit 7 is 0, the driver is ON and when it is 1, the driver is OFF. The default value for bit 7 is 0.

5.1Ø Another Method of Managing Control Lines

Since the RTS and CD control lines are connected to the slave MCU, during RTS control and CD detection there is an idle time (time required for exchanging the master MCU commands) which may cause the user inconvenience.

To avoid this, serial POUT and PIN can be used instead of RTS and CTS as control lines (Fig. 5-11).

POUT corresponds to bit 5 of address 26, and is active low.

Subroutine WRTP26 is used to set data in address 26. PIN corresponds to bit 6 of port 1 and is also active low.

(Example)

Note:
As the floppy disk unit does not use PIN and POUT for serial communication, the RS-232C port can use them as control lines.

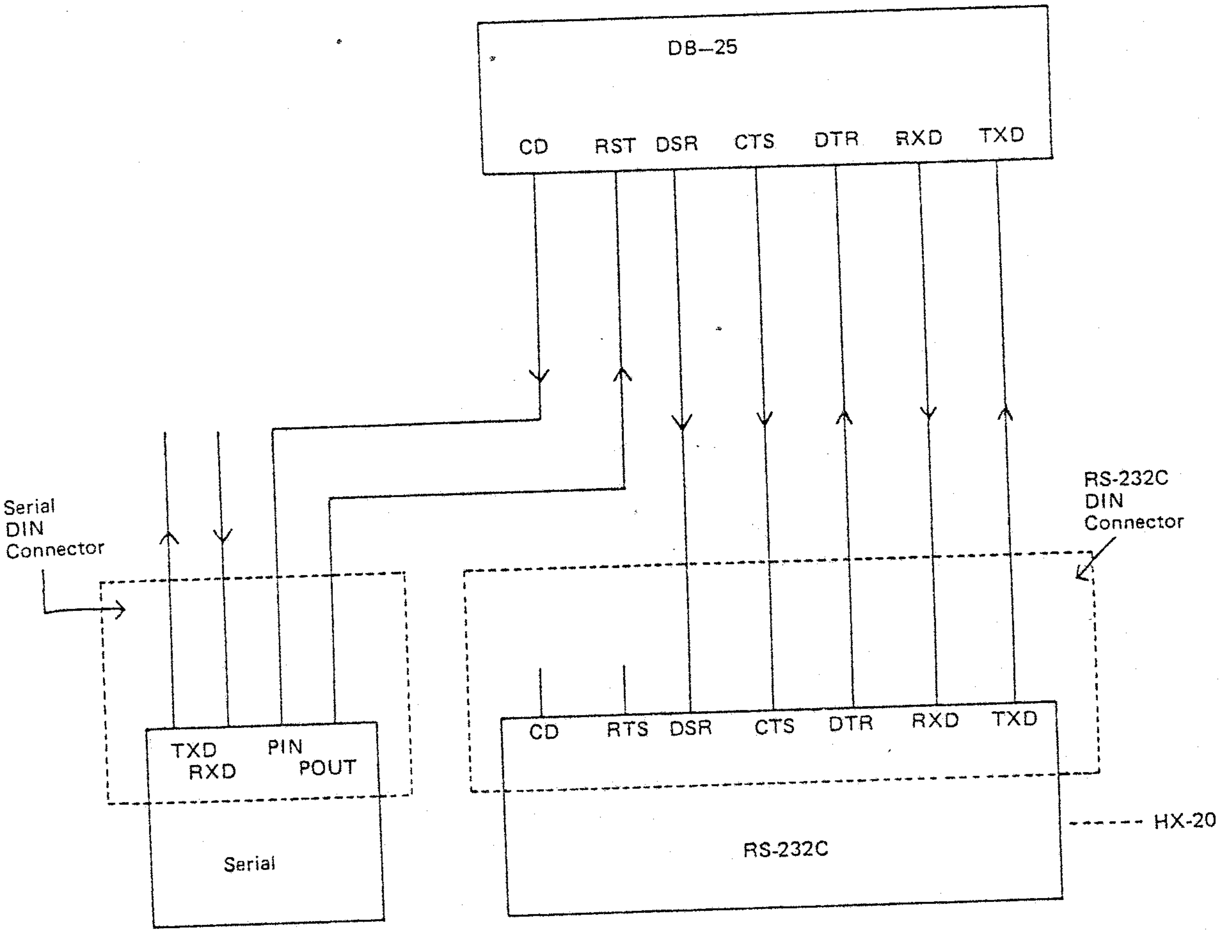


Fig. 5-11 Modification of RS-232C Control Lines

5.11 RS-232C Subroutines

Subroutine name	Entry point	Description
RSMST 7 0 A 11110Ø01 B 0000Ø000	FF8A	Specifies the RS-232C mode. Sets values in variables RSBITL, RSMODS, and RSBAUD. Communications with the slave MCU are not performed.
		Parameters: At Entry (A): Mode Bit Ø and 1: Stop bit length (1, 2 or 3) Bit 2: Specifies whether or not carrier detection will be performed. Ø: Carrier detection 1: No carrier detection Bit 3: RTS (Ø: OFF 1: ON) Bit 4: DRS Ø: Checks DSR 1: Does not check DSR Bit 5: CTS Ø: Checks CTS 1: Does not check CTS Bits 6 and 7: Parity Ø: Even 1: Odd 2 or 3: None (B): Bit rate and word length Bits Ø through 3: Word length (5, 6, 7, and 8) Bits 4 through 7: Bit rate Ø: 11Ø BPS 1: 15Ø BPS 2: 3ØØ BPS 3: 6ØØ BPS 4: 12ØØ BPS 5: 24ØØ BPS 6: 48ØØ BPS 7: 96ØØ BPS (transmission only) At Return None Registers retained (A), (B), and (X) Subroutines referenced None Variables used None
RSONOF	FF85	Turns ON/OFF the RS-232C driver. When bits 3 and 4 of SRSTS are off, this subroutine turns the driver ON and transmits a 1Ø-bit preamble (data logic 1). If the driver is already ON, the ON procedure will be ignored but no error will occur.

Subroutine name	Entry point	Description
		Parameters: At Entry (A) 0: Turns OFF the driver power. 1: Turns ON the driver power. At Return (A): Error code (C): Abnormal I/O flag (Z): According to the value of (A). Registers retained (B) and (X) Subroutines referenced SNSCOM Variables used None
RSOPEN	FF82	Opens the RS-232C input, initiates fetching data into a buffer, and exchanges commands between the master and slave MCUs. Receive data is stored in the receive buffer via the SCI (interrupt processing). When the RS-232C input is opened, RTS is set at the value specified in subroutine RSMST.
		Parameters: At Entry (A, B): Receive buffer size (X): Starting address of the receive buffer At Return (C): Abnormal I/O flag (A): Return codes 00: RS-232C input has been correctly opened. 01: The driver is OFF. Registers retained None Subroutines referenced SNSCOM, SNSCOW and SNSDAT Variables used None (Example) In this example, a 260-byte monitor buffer is opened as the receive buffer. LDA A #$OD Even parity, CTS/DSR check, RTS high CD check, 1 stop bit LDA B #$27 300 BPS 7-bit word length JSR RSMST LDA A #1 Driver ON JSR RSONOF LDD #260 Buffer size = 260 bytes LDX #CASBUF JSR RSOPEN
RSCLOS	FF7F	Closes input to the RS-232C port and sends a command to the slave MCU to terminate reception. This subroutine does not turn the driver OFF.

Subroutine name	Entry point	Description
		Parameters: At Entry None At Return (C): Abnormal I/O flag (A): Return codes ØØ: RS-232C has been correctly closed. (Only this code is currently available.) (Z): According to the value of (A) Registers retained (B) and (X) Variables used None Subroutines referenced None
RSGSTS	FF7C	Inputs the value of the status register. When a receive error occurs, this subroutine fetches the error status from the slave MCU and inputs this value to the master MCU. Then, the error status of the slave MCU is cleared. Logic 1 in any bit indicates an error.
		Parameters: At Entry None At Return (C): Abnormal I/O flag (A): Status managed by master MCU (RS-232 transmitting side) Bit 7: 1: Receive buffer overflow (B): Status managed by the slave MCU (RS-232C receiving side) Bit Ø: Carrier disconnection (OFF) Bit 1: Parity error Bit 2: Overrun error Bit 5: Receive error Registers retained (X) Subroutines referenced SNSCOM and CHKRS Variables used None
RSGET	FF79	Fetches one character from the receive buffer. The data in the receive buffer is stored in word length + parity bit format. Once a character is fetched, the parity bit is set to Ø. This parity bit is not stored in the receive buffer if the format is 8 bits + 1 parity bit.
		Parameters: At Entry None At Return (C): Abnormal I/O flag (A): Received character (B): Return codes

Subroutine name	Entry point	Description
		ØØ: Normal Ø1: Receive buffer full CØ: Parity error C1: Carrier disconnection (OFF) Note: Carrier disconnection (OFF) error occurs not when the carrier falls but when the buffer becomes empty. (Z): According to the value of (B). Registers retained (X) Subroutines referenced None Variables used RØH
RSPUT	FF76	Transmits one character through the the RS-232C port. Note that no transmit buffer is provided.
		Parameters: At Entry (A): Output characters If the number of bits to be transmitted is less than 8 bits, data is right-justified. The remaining bits (including the parity bit) can be any value. At Return (C): Abnormal I/O flag (B): Return codes ØØ: Normal Ø1: No data transmitted when DSR is OFF. Ø2: No data transmitted when CTS is OFF. Ø3: No data transmitted when both DSR and CTS are OFF. (Z): According to the value of (B) Registers retained (A) and (X) Subroutines referenced None Variables used RØ, R1, and R2H
CHKRS	FF16	Sends a command to the slave MCU to resume the interrupted RS-232C input.
		Parameters: At Entry None At Return None Registers retained (A), (B), (X), and condition code (CC) Subroutines referenced RSRSRT Variables used None

5.12 RS-232C Work Areas

Address		Variable	Byte	Description
(from)	(to)	name	count	
1AF	1B0	RSBAUD	2	RS-232C bit rates (clock cycles) 150 BPS : 1000_{16} 300 BPS : 800_{16}
1B1	1B2	RSCRC	2	Polynominal expressions generated for CRC Polynominal expression CRC-CCITT $(1+x^5+x^{12}+x^{16})$ equals 8408_{16} (default value) CRC-16 $(1+x^2+x^{15}+x^{16})$ equals $A001_{16}$ x^{16} is always 1, x^{15} is bit 0, and x^0 is bit 15.
1B3	1B4	RSBCC	2	BCC register for CRC check
1B5	1B5	RSBITL	1	RS-232C word length (stop bit excluded).Word length must be 5, 6, 7, or 8.
1B6	1B6	RSMODS	1	RS-232C mode Bits 0 and 1: Stop bit length (bit 1, bit 0): (0,1) = 1 (1,0) = 2 Bit 2: Carrier (CD) detection 0: Carrier detection 1: No carrier detection Bit 3: RTS 0: RTS OFF (low level) 1: RTS (high level) Bit 4: DSR check 0: Checks if DSR is OFF. 1: Does not check if DSR is OFF. Bit 5: CTS check 0: Checks if CTS is OFF. 1: Does not check if CTS is OFF. Bits 6 and 7: Parity (bit 7, bit 6) = (0,0) : Even parity (0,1) : Odd parity (1,x) : No parity x: don't care
1B7	1B7	RSSTSR	1	RS-232C error status register For all bits of this variable, logic 0 indicates normal operation and logic 1 indicates error. Bit 0: Carrier disconnection (OFF) Bit 1: Parity Bit 2: Overrun Bit 3: Undefined Bit 4: Undefined Bit 5: Receive error Bit 6: Transmit error Bit 7: Receive buffer overflow

Address		Variable	Byte	Description
(from)	(to)	name	count	
1B8	1B9	RSBFAD	2	Starting address of RS-232C receive buffer
1BA	1BB	RSBFBT	2	Last address of RS-232C receive buffer plus 1
1BC	1BD	RSBFSZ	2	Size of RS-232C receive buffer (in bytes)
1BE	1BE	RSINP	2	Pointer indicating the last data stored in the RS-232C receive buffer (Indicates the next address the buffer in which received data will be stored.)
1C0	1C1	RSOUP	2	Pointer indicating the last data fetched from the RS-232C receive buffer (Indicates the next address to be fetched when data is fetched from the receive buffer.)
1C2	1C3	RSDCNT	2	Number of data in the RS-232C receive buffer (in bytes)

```
ERR   SEQ    LOC  OBJECT       PROGRAM  RS232C     -- RS232C SEND/RECEIVE DATA ROUTINE ---

     00001                                     NAM    RS232C
     00002                                     TTL    -- RS232C SEND/RECEIVE DATA ROUTINE ---
     00003                              *
     00004                              * RS232C SUBROUTINE.
     00005                              *  2 SUBROUTINES
     00006                              *  1. GET RECEIVED CHARACTER FROM RS232C RECEIVED DATA BUFFER (RSGET).
     00007                              *  2. TRANSMITTE ONE CHARACTER TO TXD LINE (RSPUT).
     00008                              *
     00009                              * FILE NAME  'EX5A'    BY K AKAHANE
     00010                                     OPT    PAGE=55
     00011                                     OPT    LOAD
     00012                              *
     00013                              * MCU 6301 I/O PORT
     00014              0002  A         PORT1  EQU    $02         * I/O PORT 1
     00015              0003  A         PORT2  EQU    $03         * I/O PORT 2
     00016              0006  A         PORT3  EQU    $06         * I/O PORT 3
     00017                              * OTHER REGISTERS
     00018              0009  A         FRC    EQU    $09         * FREE RUNNING COUNTER
     00019              000B  A         OCR    EQU    $0B         * OUTPUT COMPARE REGISTER
     00020              0008  A         TCSR   EQU    $08         * TIME CONTROL AND STATUS REGISTER
     00021                              * GENERAL REGISTERS USED BY I/O ROUTINE
     00022A 0050                               ORG    $50
     00023              0050  A         R0     EQU    *           * 2 BYTES REGISTER  (R0H,R0L)
     00024A 0050        0001  A         R0H    RMB    1
     00025A 0051        0001  A         R0L    RMB    1
     00026              0052  A         R1     EQU    *           * 2 BYTES REGISTER  (R1H,R1L)
     00027A 0052        0001  A         R1H    RMB    1
     00028A 0053        0001  A         R1L    RMB    1
     00029              0054  A         R2     EQU    *           * 2 BYTES REGISTER  (R2H,R2L)
     00030A 0054        0001  A         R2H    RMB    1
     00031A 0055        0001  A         R2L    RMB    1
     00032              0056  A         R3     EQU    *           * 2 BYTES REGISTER  (R3H,R3L)
     00033A 0056        0001  A         R3H    RMB    1
     00034A 0057        0001  A         R3L    RMB    1
     00035A 007A                               ORG    $7A
     00036A 007A        0001  A         SRSTS  RMB    1           * SERIAL STATUS
     00037                              *                         * BIT 0,1: RS232 MODE(00:STOP 01:INTERRUPT READ
     00038                              *                                                   02:READ ONE CHARACTER)
     00039                              *                         * BIT 2: EXECUTE/PAUSE (0:ON EXECUTE  1:PAUSE)
     00040                              *                         * BIT 3: RS232 DRIVER (0:OFF  1:DRIVER ON)
     00041                              *                         * BIT 4: SERIAL DRIVER (0:OFF  1:DRIVER ON)
     00042                              *                         * BIT 5,6,7: CPU SERIAL RECEIVE INTERRUPT MODE
     00043                              *                         *        0:EXTERNAL CASSETTE READ
     00044                              *                         *        1:MICRO CASSETTE READ
     00045                              *                         *        2:RS232C READ
     00046                              *                         *        3:READ FROM SERIAL COMMUNICATION
     00047                              *                         *        4:EXTERNAL CASSETTE WRITE
     00048                              *                         *        5:MICRO CASSETTE WRITE
     00049                              *                         *        6,7:UNDEFINED FOR WRITE
     00050A 007B        0001  A         RUNMOD RMB    1           * RUN MODE   ($80:BASIC   $00:SYSTEM)
     00051A 007C        0001  A         SIOSTS RMB    1           * SLAVE I/O STATUS   (EACH BIT  0:OFF,   1:ON)
     00052                              *                         * BIT 0: PRINTER
     00053                              *                         * BIT 1: EXTERNAL CASSETTE
     00054                              *                         * BIT 2: INTERNAL MICRO CASSETTE
     00055                              *                         * BIT 3: RS232C ON (READ)
```

```
ERR   SEQ    LOC  OBJECT       PROGRAM  RS2320C     -- RS232C SEND/RECEIVE DATA ROUTINE ---

     00056                            *                               * BIT 4: SPEAKER ON
     00057                            *                               * BIT 5: ROM CASSETTE
     00058                            *                               * BIT 6: BAR CODE READER
     00059                            *                               * BIT 7: BREAK SLAVE CPU (0:ON EXECUTE
     00060                            *                               *                            1:BROKEN BY INTERRUPT
     00061A 007D     0001  A          MIOSTS RMB     1                * MAIN I/O STATUS  EACH BIT (0:OFF 1:ON)
     00062                            *                               * BIT 0: LCD ON READ/WRITE CHARACTERS
     00063                            *                               * BIT 1: NOW SENDING COMMAND TO SLAVE CPU
     00064                            *                               * BIT 2: NOW SENDING DATA TO SERIAL LINE (1:ON
     00065                            *                               * BIT 3: ON CLOCK INTERRUPT  (1:ON)
     00066                            *                               * BIT 4: (POWER FAIL)
     00067                            *                               * BIT 5: (OFF POWER SWITCH)
     00068                            *                               * BIT 6: ON PAUSE KEY
     00069                            *                               * BIT 7: ON BREAK KEY
     00070                            * WORK AREA
     00071A 01AF                            ORG     $1AF
     00072                            *
     00073                            *
     00074                            * RS232C WORK AREA
     00075           01AF  A          RSWKTP EQU     *                * RS232C WORK TOP ADDRESS
     00076A 01AF     0002  A          RSBAUD RMB     2                * RS232C BIT RATE (NUMBER OF CLOCK CYCLE)
     00077A 01B1     0002  A          RSCRC  RMB     2                * RS232C GENELATING POLYNOMIAL
     00078A 01B3     0002  A          RSBCC  RMB     2                * RS232C BCC REGISTER
     00079A 01B5     0001  A          RSBITL RMB     1                * RS232C BIT LENGTH (5 6 7 8)
     00080A 01B6     0001  A          RSMODS RMB     1                * RS232C MODE
     00081                            *                               * (0,1:NUMBER OF STOP BITS)
     00082                            *                               * (2: CARRIER DETECT MASK   0:CHECK 1:MASK)
     00083                            *                               * (3: CLEAR TO SEND  0:LOW  1:HIGH)
     00084                            *                               * (4: DSR 0:CHECK 1:NO CHECK)
     00085                            *                               * (5: CTS 0:CHECK 1:NO CHECK)
     00086                            *                               * (6,7:PARITY 00:EVEN 01:ODD 10,11:NONE PARITY
     00087                            * RS232C BUFFER POINTER
     00088A 01B7     0001  A          RSSTSR RMB     1                * RS232C STATUS REGISTER
     00089                            *                               * (0: CARRIER DETECT  0:NORMAL 1:ERROR)
     00090                            *                               * (1: PARITY          0:NORMAL 1:ERROR)
     00091                            *                               * (2: OVERRUN         0:NORAML 1:ERROR)
     00092                            *                               * (5: READ ERROR      0:NORMAL 1:ERROR)
     00093                            *                               * (6: WRITE ERROR     0:NORMAL 1:ERROR)
     00094                            *                               * (7: BUFFER OVER     0:NORMAL 1:OVERFLOW)
     00095                            *
     00096A 01B8     0002  A          RSBFAD RMB     2                * RS232C READ BUFFER ADDRESS
     00097A 01BA     0002  A          RSBFBT RMB     2                * RS232C READ BUFFER BOTTOM ADDRESS + 1
     00098A 01BC     0002  A          RSBFSZ RMB     2                * RS232 READ BUFFER SIZE (0001 - FFFF)
     00099A 01BE     0002  A          RSINP  RMB     2                * POINTER WHERE NEXT RECEIVED CHARACTER IS STO
     00100A 01C0     0002  A          RSOUP  RMB     2                * POINTER WHERE NEXT CHARACTER IS LOADED
     00101A 01C2     0002  A          RSDCNT RMB     2                * NUMBER OF DATA IN THE BUFFER
     00102                            *
     00103                            * RS232C: GET ONE CHARACTER FROM RECEIVE BUFFER
     00104                            *   1: GET ONE CHARACTER FROM RS232 RECEIVE BUFFER
     00105                            *   2: IF BIT LENGTH < 8, AND 'PARITY CHECK' MODE, DO PARITY CHECK AND
     00106                            *       SET RETURN CODE
     00107                            * PARAMETER
     00108                            * ON ENTRY
     00109                            *   NONE
     00110                            * ON EXIT
```

```
  00111                         *   (A): CHARACTER (WITHOUT PARITY BIT)
  00112                         *   (B): STATUS    $01:RECEIVED BUFFER IS EMPTY
  00113                         *                  $00:NORMAL
  00114                         *                  MSB= 1:ERROR  0:NORMAL
  00115                         *                  $C0:PARITY ERROR    $C1:CD ERROR (CARRIER DOWN)
  00116                         *   (C): SLAVE STATUS   0:NORMAL  1:ERROR
  00117                         *  SET Z N FLAG DEPEND ON VALUE OF (B) REGISTER
  00118                         * REGISTER  PRESERVE  X
  00119                         *  WORK USE AS REGISTER
  00120                         *    ROH:EFFECTIVE BITS AS DATA (BIT LENGTH=7 THEN $7F,
  00121                         *                                BIT LENGTH=8 THEN $FF)
  00122           01C4  A       RSGET  EQU    *
  00123A 01C4 C6 01     A              LDA B  #$01       * PRESET 'BUFFER EMPTY' CODE
  00124A 01C6 0D                       SEC               * PRESET ERROR I/O FLAG
  00125A 01C7 7B B07D   A              TIM    #$80,MIOSTS * ERROR I/O ?
  00126A 01CA 26 42 020E               BNE    RSIN23
  00127A 01CC 3C                       PSHX
  00128                         *
  00129A 01CD FE 01C2   A              LDX    RSDCNT     * ARE THERE DATA IN THE BUFFER ?
  00130A 01D0 27 3D 020F               BEQ    RSIN25     *  (B):1
  00131                         * SET EFFECTIVE BITS TO ROH
  00132A 01D2 FC 01B5   A              LDD    RSBITL     * (B):RSMODS
  00133A 01D5 7F 0050   A              CLR    ROH        * (A):BIT LENGTH
  00134A 01D8 0D                RSIN1A SEC
  00135A 01D9 79 0050   A              ROL    ROH        * ROH <-- ($7F IF B=7), <--- ($FF IF B=8)
  00136A 01DC 4A                       DEC A
  00137A 01DD 26 F9 01D8               BNE    RSIN1A
  00138                         *
  00139A 01DF 0F                       SEI               * IF RS232C RECEIVED INTERRUPT IS CAUSED, THE
  00140A 01E0 FE 01C0   A              LDX    RSOUP      *  POINTER MAY BE DESTROYED.
  00141A 01E3 A6 00     A              LDA A  0,X        * (A): DATA
  00142A 01E5 08                       INX
  00143A 01E6 BC 018A   A              CPX    RSBFBT     * IF THE POINTER SHOWS BOTTOM ADDRESS + 1 OF TH
  00144A 01E9 26 03 01EE               BNE    RSIN10     * BUFFER, POINTER MUST BE SET TO TOP ADDRESS.
  00145A 01EB FE 0188   A              LDX    RSBFAD
  00146A 01EE FF 01C0   A       RSIN10 STX    RSOUP
  00147A 01F1 FE 01C2   A              LDX    RSDCNT
  00148A 01F4 09                       DEX
  00149A 01F5 FF 01C2   A              STX    RSDCNT     * DATA COUNTER <---  CURRENT VALUE - 1
  00150A 01F8 0E                       CLI
  00151                         * PARITY ERROR CHECK
  00152A 01F9 58                       ASL B             * PARITY CHECK MODE ?
  00153A 01FA 25 0F 020B               BCS    RSIN15     * MODE = 'CHECK PARITY' ?
  00154A 01FC 58                       ASL B
  00155A 01FD 16                       TAB               * (B) <--- DATA,  (C)<--- PARITY MODE (0:EVEN)
  00156A 01FE 94 50     A              AND A  ROH        *  TAKE DATA BITS(IGNORE PARITY BIT)
  00157A 0200 24 02 0204        RSIN11 BCC    RSIN12
  00158A 0202 C8 80     A              EOR B  #$80
  00159A 0204 58                RSIN12 ASL B
  00160A 0205 26 F9 0200               BNE    RSIN11
  00161A 0207 56                       ROR B             * BIT7,BIT6 <--- (C)
  00162A 0208 57                       ASR B
  00163A 0209 20 01 020C               BRA    RSIN20     * PARITY ERROR = $C0
  00164                         *
  00165A 020B 5F                RSIN15 CLR B             * NORMAL RETURN
```

```
ERR   SEQ    LOC  OBJECT        PROGRAM  RS232C      -- RS232C SEND/RECEIVE DATA ROUTINE ---

     00166                              * BUFFER IS EMPTY
     00167             020C  A          RSIN20 EQU    *
     00168A 020C 5D                            TST B           * CLEAR (C), SET (Z)
     00169A 020D 38                            PULX
     00170A 020E 39                     RSIN23 RTS
     00171                              * BUFFER IS EMPTY, IS CARRIER DOWN ?
     00172             020F  A          RSIN25 EQU    *
     00173A 020F 7B 047A  A                    TIM    #$4,SRSTS * ON PAUSE ?
     00174A 0212 26 F8 020C                    BNE    RSIN20
     00175A 0214 7B 0402  A                    TIM    #$4,PORT1 * SFLAG = ON ?
     00176A 0217 27 F3 020C                    BEQ    RSIN20
     00177A 0219 C6 C1     A                   LDA B  #$C1       * CD ERROR
     00178A 021B 20 EF 020C                    BRA    RSIN20
     00179                              *
     00180                              *
     00181                              *
     00182                              *
     00183                              * SEND ONE TRANSMITTED CHARACTER SUBROUTINE
     00184                              * PARAMETER
     00185                              * ON ENTRY
     00186                              *  TRANSMITTED CHARACTER
     00187                              * ON EXIT
     00188                              *  (B): BIT 0 (1:DSR LOW) CHARACTER IS NOT SENT
     00189                              *       BIT 1 (1:CTS LOW) CHARACTER IS NOT SENT
     00190                              *       BIT 2 - 7 (ALWAYS 0)
     00191                              *  (Z)  DEPEND ON VALUE OF (B)
     00192                              *  (C)  0:NORMAL  1:I/O ERROR
     00193                              * REGISTER PRSERVE  A,X
     00194                              *
     00195                              *  WORK USE AS REGISTER
     00196                              *  R0H:PARITY BIT (LSB)
     00197                              *  R0L:FLAG OF 'WITH PARITY BIT'  (0:YES   1:NO)
     00198                              *  R1H:SAVE DATA
     00199                              *  R1L:BIT LENGTH
     00200                              *
     00201                              * NOTE. OCR IS USED. AND OCR IS USED BY KEY ROUTINE EITHER.
     00202                              *
     00203             021D  A          RSPUT  EQU    *
     00204A 021D 0D                            SEC              * PRESET I/O ERROR FLAG
     00205A 021E 7B B07D  A                    TIM    #$80,MIOSTS * I/O ERROR ?
     00206A 0221 26 0F 0232                    BNE    SNDR04
     00207                              *
     00208                              * CHECK DSR, CTS
     00209A 0223 F6 0186  A                    LDA B  RSMODS     * TAKE MODE (DSR CTS BITS)
     00210A 0226 57                            ASR B             * RSMODS (DSR:BIT 4,CTS:BIT 5) MASK=1
     00211A 0227 57                            ASR B
     00212A 0228 57                            ASR B
     00213A 0229 57                            ASR B             * PORT1 (DSR:BIT 0, CTS:BIT 1) NORMAL='LOW'
     00214A 022A 53                            COM B
     00215A 022B D4 02     A                   AND B  PORT1      * CHECK DSR, CTS
     00216A 022D C4 03     A                   AND B  #$3
     00217A 022F 27 02 0233                    BEQ    SNDR05
     00218A 0231 0C                            CLC               * CTS, DSR LOW (ERROR)
     00219A 0232 39                     SNDR04 RTS
     00220                              *
     00221A 0233 36                    SNDR05 PSH A
     00222A 0234 97 52      A                 STA A   R1H
     00223A 0236 3C                           PSHX
     00224A 0237 CE 01AF    A                 LDX     #RSWKTP   * (X): TOP RAM ADDRESS OF WORK AREA FOR RS232C
     00225A 023A 0F                           SEI               * DISABLE INTERRUPT
     00226A 023B A6 06      A                 LDA A   RSBITL-RSWKTP,X
     00227A 023D 97 53      A                 STA A   R1L
     00228A 023F 4F                           CLR A
     00229A 0240 E6 07      A                 LDA B   RSMODS-RSWKTP,X * RSMODS  (BIT7:WITH PARITY FLAG
     00230A 0242 05                           ASLD                BIT6:EVEN OR ODD)
     00231A 0243 97 51      A                 STA A   ROL       * ROL:NUMBER OF PARITY BITS (ROL: 0 OR 1)
     00232A 0245 4F                           CLR A
     00233A 0246 05                           ASLD
     00234A 0247 97 50      A                 STA A   ROH       * LSB <--- PARITY
     00235                             *
     00236            0249  A          SNDR20 EQU     *
     00237A 0249 7B 4008    A                 TIM     #$40,TCSR * OCR OVERFLOW ?
     00238A 024C 26 0B 0259                   BNE     SNDR30
     00239                             * NOT OVERFLOW
     00240A 024E DC 08      A                 LDD     OCR       * 'TIME TILL NEXT EDGE' < 1.6*$20 MICRO SEC ?
     00241A 0250 93 09      A                 SUBD    FRC       * YES, THEN WAIT OCR OVERFLOW, NOW START 'START
     00242A 0252 83 0020    A                 SUBD    #$20      *  BIT'.
     00243A 0255 28 F2 0249                   BMI     SNDR20    * NO, THEN WAIT TIME OF START BIT
     00244A 0257 20 07 0260                   BRA     SNDR40
     00245                             * OCR OVER, SET NEXT TIME
     00246A 0259 DC 09      A          SNDR30 LDD     FRC       * SET TIME OF START BIT
     00247A 025B C3 0020    A                 ADDD    #$20
     00248A 025E DD 0B      A                 STD     OCR
     00249            0260  A          SNDR40 EQU     *
     00250A 0260 71 FE08    A                 AIM     #$FF-$01,TCSR * SET 'LOW'
     00251                             *
     00252            0263  A          SNDR45 EQU     *
     00253A 0263 7B 4008    A                 TIM     #$40,TCSR * WAIT UNTIL OVERFLOW
     00254A 0266 27 FB 0263                   BEQ     SNDR45
     00255                             * SET NEXT DATA BIT
     00256A 0268 5F                           CLR B
     00257A 0269 77 0052    A                 ASR     R1H
     00258A 026C 59                           ROL B             * (B)  0 OR 1
     00259A 026D 26 05 0274                   BNE     SNDR50
     00260                             * SET  0
     00261A 026F 71 FE08    A                 AIM     #$FF-$1,TCSR
     00262A 0272 20 06 027A                   BRA     SNDR53    * (PARITY IS NOT CHENGED)
     00263                             * SET 1
     00264            0274  A          SNDR50 EQU     *
     00265A 0274 72 0108    A                 OIM     #$1,TCSR
     00266A 0277 75 0150    A                 EIM     #$1,ROH   * COMPULE PARITY
     00267                             *
     00268A 027A E8 05      A          SNDR53 EOR B   RSBCC+1-RSWKTP,X * COMPUTE CRC
     00269A 027C A6 04      A                 LDA A   RSBCC-RSWKTP,X
     00270A 027E 04                           LSRD
     00271A 027F 24 04 0285                   BCC     SNDR54
     00272A 0281 A8 02      A                 EOR A   RSCRC-RSWKTP,X
     00273A 0283 E8 03      A                 EOR B   RSCRC+1-RSWKTP,X
     00274A 0285 ED 04      A          SNDR54 STD     RSBCC-RSWKTP,X
     00275                             * SET NEXT TIME
```

```
    00276A 0287 DC 08      A              LDD    OCR
    00277A 0289 E3 00      A              ADDD   RSBAUD-RSWKTP,X
    00278A 028B DD 08      A              STD    OCR
    00279A 028D 7A 0053    A              DEC    R1L        * FINISHED ?
    00280A 0290 26 D1 0263                BNE    SNDR45
    00281                          * ADD PARITY ?
    00282A 0292 96 51      A              LDA A  R0L
    00283A 0294 26 0A 02A0                BNE    SNDR60
    00284A 0296 D6 50      A              LDA B  R0H        * SET PARITY  (R1H <--- R0H)
    00285A 0298 4C                        INC A             * 'ADD PARITY' FLAG <--- 'NONE' (R0L <--- 0)
    00286A 0299 DD 51      A              STD    R0L
    00287A 029B 7C 0053    A              INC    R1L        * BIT COUNT <--- 1
    00288A 029E 20 C3 0263                BRA    SNDR45
    00289                          *
    00290                          * ADD STOP BITS
    00291             02A0  A      SNDR60 EQU    *          * WAIT UNTIL START OF LAST BIT
    00292A 02A0 7B 4008    A              TIM    #$40,TCSR
    00293A 02A3 27 FB 02A0                BEQ    SNDR60
    00294                          *
    00295A 02A5 DC 08      A              LDD    OCR
    00296A 02A7 E3 00      A              ADDD   RSBAUD-RSWKTP,X
    00297A 02A9 DD 08      A              STD    OCR
    00298                          *
    00299A 02AB 72 0108    A              OIM    #$1,TCSR   * STOP BIT
    00300             02AE  A      SNDR70 EQU    *
    00301A 02AE 7B 4008    A              TIM    #$40,TCSR * WAIT UNTIL START TIME OF STOP BIT
    00302A 02B1 27 FB 02AE                BEQ    SNDR70
    00303                          *
    00304A 02B3 EE 06      A              LDX    RSMODS-RSWKTP-1,X
    00305A 02B5 18                        XGDX              * (X):OCR LAST TIME,
    00306A 02B6 C4 03      A              AND B  #$3        * (B):MSMODS (LS 3BITS:NUMBER OF STOP BITS)
    00307A 02B8 26 01 02BB                BNE    SNDR80
    00308A 02BA 5C                        INC B             * IF 0, 1 STOP BIT
    00309A 02BB 4F                 SNDR80 CLR A             * (X): NUMBER OF STOP BITS
    00310A 02BC 18                        XGDX
    00311A 02BD F3 01AF    A       SNDR90 ADDD   RSBAUD     * (A,B):HIGH BIT TIME
    00312A 02C0 09                        DEX
    00313A 02C1 26 FA 02BD                BNE    SNDR90
    00314                          *
    00315A 02C3 DD 08      A              STD    OCR
    00316                          *
    00317A 02C5 38                        PULX
    00318A 02C6 32                        PUL A
    00319A 02C7 0E                        CLI               * IF RECEIVED KEY INTERRUPT, KEY SAMPLING TIME I
    00320                          *                       * NOT PUNCTUAL.
    00321A 02C8 5F                        CLR B
    00322A 02C9 39                        RTS
    00323                          *
    00324             0000  A             END
***** TOTAL ERRORS      0
```

```
ERR   SEQ    LOC  OBJECT       PROGRAM  TERM       --- TERMINAL MODE WITHOUT HARD COPY ---

     00001                           *
     00002                                  NAM    TERM
     00003                           * TSS TERMINAL MODE
     00004                           * 300 BPS, FULL DUPLEX, WITHOUT HARD COPY
     00005                           * FILE NAME    'EX$5'  BY  K.A
     00006                                  TTL    --- TERMINAL MODE WITHOUT HARD COPY ---
     00007                                  OPT    LOAD
     00008                                  OPT    PAGE=55
     00009                           *
     00010A 1000                            ORG    $1000
     00011                           *
     00012                           * EXEMPLE OF TERMINAL MODE
     00013                           *
     00014          FF4F  A          DSPSCR EQU    $FF4F
     00015          FF5E  A          SCRFNC EQU    $FF5E
     00016          FF85  A          RSONOF EQU    $FF85
     00017          FF88  A          RSMST  EQU    $FF88
     00018          FF82  A          RSOPEN EQU    $FF82
     00019          FF7F  A          RSCLOS EQU    $FF7F
     00020          FF79  A          RSGET  EQU    $FF79
     00021          FF76  A          RSPUT  EQU    $FF76
     00022          FF9A  A          KEYIN  EQU    $FF9A
     00023          FF9D  A          KEYSTS EQU    $FF9D
     00024                           *
     00025                           * INITIALIZE
     00026A 1000 CC 8422  A                 LDD    #$8422    * CONSTRUCT SCREEN PACKET
     00027A 1003 FD 105B  A                 STD    SCRPK1
     00028A 1006 86 87    A                 LDA A  #$87
     00029A 1008 B7 105D  A                 STA A  SCRPK2
     00030A 100B CC 1303  A                 LDD    #$1303
     00031A 100E FD 105E  A                 STD    SCRPK2+1
     00032A 1011 CC 1400  A                 LDD    #$1400
     00033A 1014 FD 1060  A                 STD    SCRPK2+3
     00034A 1017 CE 105B  A                 LDX    #SCRPK1   * INITIALIZE SCREEN
     00035A 101A BD FF5E  A                 JSR    SCRFNC
     00036A 101D CE 105D  A                 LDX    #SCRPK2
     00037A 1020 BD FF5E  A                 JSR    SCRFNC
     00038A 1023 CC 3D27  A                 LDD    #$3D27    * SET MODE(STOP:1 CD:NO-CHECK, RTS:ON, PARITY:E
     00039                           *                     * 7 BITS LENGTH, 300 BPS
     00040A 1026 BD FF88  A                 JSR    RSMST
     00041A 1029 86 01    A                 LDA A  #1        * RS232C DRIVER ON
     00042A 102B BD FF85  A                 JSR    RSONOF
     00043A 102E FE FFDC  A                 LDX    $FFDC     * (X):BUFFER ADDRESS (SYSTEM BUFFER)
     00044A 1031 CC 0104  A                 LDD    #260      * (A,B): BUFFER SIZE
     00045A 1034 BD FF82  A                 JSR    RSOPEN    * RECEIVE OPEN
     00046                           *
     00047A 1037 BD FF9D  A          REDKEY JSR    KEYSTS    * ACCEPT FROM KEY BOARD ?
     00048A 103A 25 1E 105A                 BCS    BRKRTN    * IF BREAK KEY IS PRESSED, RETURN (IN BASIC MOD
     00049A 103C 27 09 1047                 BEQ    RCVRS
     00050                           * ACCEPTED CHARACTER FROM KB.
     00051A 103E BD FF9A  A                 JSR    KEYIN
     00052A 1041 BD FF76  A                 JSR    RSPUT     * TRANSMIT ACCEPTED CHARACTER.
     00053A 1044 BD FF4F  A                 JSR    DSPSCR    * DISPLAY ACCEPTES CHRACTER TO VIRTUAL SCREEN.
     00054A 1047 FE FFD8  A          RCVRS  LDX    $FFD8     * ARE THERE RECEIVED CHARACTER IN THE BUFFER ?
     00055A 104A EC 00    A                 LDD    0,X
     *** 6301 CROSS MACROASSEMBLER   VER1.0 ***                                   10/21/82    09:42:16

ERR   SEQ   LOC  OBJECT       PROGRAM  TERM        --- TERMINAL MODE WITHOUT HARD COPY ---

     00056A 104C 27 E9 1037               BEQ    REDKEY
     00057A 104E BD FF79  A               JSR    RSGET
     00058A 1051 81 7F    A               CMP A  #$7F
     00059A 1053 24 E2 1037               BCC    REDKEY     * IGNORE 7F - FF CHARACTERS
     00060A 1055 BD FF4F  A               JSR    DSPSCR     * DISPLAY RECEIVED CHRACTER TO VIRTUAL SCREEN.
     00061A 1058 20 DD 1037               BRA    REDKEY
     00062                         *
     00063A 105A 39                BRKRTN RTS
     00064                         * VIRTUAL SCREEN PACKET
     00065A 105B    84    A        SCRPK1 FCB    $84        * SELECT SCREEN DEVICE (LCD)
     00066A 105C    22    A               FCB    $22
     00067A 105D    87    A        SCRPK2 FCB    $87        * SET SCREEN SIZE AND BUFFER ADDRESS
     J0068A 105E    13    A               FCB    19,3
          A 105F    03    A
     00069A 1060    1400  A               FDB    $1400
     00070                         *      :
     00071                         *
     00072          0000  A               END
***** TOTAL ERRORS      0
```

```
ERR   SEQ    LOC  OBJECT        PROGRAM  TERM        --- TSS TERMINAL MODE WITH HARDCOPY ---

     00001                                  NAM    TERM
     00002                                  TTL    --- TSS TERMINAL MODE WITH HARDCOPY ---
     00003                              *
     00004                              *  FILE NAME  'EX$2'        BY K.A
     00005                                  OPT    LOAD
     00006                                  OPT    PAGE=55
     00007                              *
     00008                              * EXEMPLE OF TERMINAL MODE
     00009                              *  300 BPS FULL DUPLEX TERMINAL MODE  (1200 BPS)
     00010                              *  VIRTUAL SCREEN SIZE = 20*4
     00011                              *  RECEIVED AND TRNSMITTED CHARACTERS ARE ABLE TO PRINT TO SERIAL
     00012                              *   PRINTER (MP-80, ...). THE CONNECTOR FOR HARD COPY IS 'SERIAL'.
     00013                              *  HARD COPY ROUTINE IS INCLUDED IN INTERRUPT PROCEDURE.
     00014                              *
     00015                              * CABLE
     00016                              *  1. FOR CONNECT TO MODEM (CP-20)
     00017                              *    OPTINAL CABLE
     00018                              *  2. FOR HARD COPY
     00019                              *    HC-20 SERIAL (DIN 5 PINS)     MP-80 SERIAL (DB-25)
     00020                              *     1 (GROUND)     ------------     7 (GROUND)
     00021                              *     2 (PTX)        ------------     3 (RXD)
     00022                              *     3 (PRX)        ------------     2 (TXD)
     00023                              *     4 (POUT)       ------------     6 (DSR)
     00024                              *     5 (PIN)        ------------     20 (DTR)
     00025                              *     FG             ------------     1 (PROTECTIVE GROUND)
     00026                              *
     00027                              *
     00028                              * OPERATION
     00029                              *  PF1 KEY: START HARD COPY
     00030                              *  PF2 KEY: STOP HARD COPY
     00031                              *  PF3 KEY: 1200 BPS (DISPLAY MONITOR (RECEIVED CHARACTER) = OFF)
     00032                              *  PF4 KEY: 300 BPS
     00033                              *  PF5 KEY: QUIT
     00034                              *  PF6 KEY: MONITOR DISPLAY ON
     00035                              *  PF7 KEY: MONITOR DISPLAY OFF
     00036                              *  PF8 KEY: ESC 'I'+$20 '0'
     00037                              *
     00038                              * 1200 BPS FULL DUPLEX TERMINAL PROCEDURE
     00039                              *   1: PF3 (1200 BPS)
     00040                              *   2: PF6 (MONITOR DISPLAY OFF, HARD COPY ON)
     00041                              *   3: (PF8 ?????)
     00042                              *
     00043                              * SUBROUTINE ENTRY POINT
     00044          FF4F   A            DSPSCR EQU    $FF4F      * DISPLAY ONE CHARACTER TO VIRTUAL SCREEN
     00045          FF5E   A            SCRFNC EQU    $FF5E      * VIRTUAL SCREEN FUNCTION
     00046          FF85   A            RSONOF EQU    $FF85      * RS232C DRIVER ON/OFF
     00047          FF88   A            RSMST  EQU    $FF88      * SET RS232C PARAMETERS
     00048          FF73   A            SERONF EQU    $FF73      * SERIAL DRIVER ON/OFF
     00049          FF82   A            RSOPEN EQU    $FF82      * OPEN RS232C RECEIVE
     00050          FF7F   A            RSCLOS EQU    $FF7F      * CLOSE RS232C RECEIVE
     00051          FF79   A            RSGET  EQU    $FF79      * GET RS232C ONE CHARACTER
     00052          FF76   A            RSPUT  EQU    $FF76      * SEND RS232C ONE CHARACTER
     00053          FF9A   A            KEYIN  EQU    $FF9A      * GET ONE CHARACTER FROM KEYBOARD BUFFER
     00054          FF9D   A            KEYSTS EQU    $FF9D      * GET NUMBER OF CHARACTERS IN THE KEY BUFFER
     00055          FF25   A            MENU   EQU    $FF25      * MENU
```

```
      00056                           * CONSTANTS OR REGISTERS
      00057             0011  A       TRCSR   EQU    $11        * TRANSMIT/RECEIVE CONTROL REGISTER
      00058             0013  A       STDR    EQU    $13        * SERIAL TRANSMIT DATA REGISTER
      00059             0012  A       SRDR    EQU    $12        * SERIAL RECEIVE DATA REGISTER
      00060             0008  A       TCSR    EQU    $08        * TIMER CONTROL AND STATUS REGISTER
      00061             000B  A       OCR     EQU    $0B        * OUTPUT COMPARE REGISTER
      00062             0009  A       FRC     EQU    $09        * FREE RUNNING COUNTER
      00063             0010  A       RMCR    EQU    $10        * RATE AND MODE CONTROL REGISTER
      00064                           *                         * 04:38.4 KBPS,  05:4.4 KBPS
      00065             0002  A       PORT1   EQU    $02        * I/O PORT1
      00066             0003  A       PORT2   EQU    $03        * I/O PORT2
      00067             1000  A       BUFSIZ  EQU    4096       * BUFFER SIZE FOR PRINTER
      00068             00C8  A       SCBSIZ  EQU    200        * BUFFER SIZE FOR SCREEN
      00069             1000  A       RSBSIZ  EQU    4096       * BUFFER SIZE FOR RS232C
      00070             0001  A       ECHODT  EQU    1          * TERMINAL MODE = 'ECHO CHARACTER' ?
      00071                           *                         * 0:YES,  1:NO
      00072             0109  A       SERVCT  EQU    $109       * SCI RECEIVE INTERRUPT ADDRESS
      00073                           *
      00074A 1000                             ORG    $1000
      00075                           *
      00076                           * INITIALIZE
      00077A 1000 86 01      A                LDA A  #ECHODT
      00078A 1002 B7 11E3    A                STA A  ECHO
      00079A 1005 CE 11C7    A                LDX    #SCRPKD    * SET SCREEN PACKET  X:DATA ADDRESS
      00080A 1008 C6 0E      A                LDA B  #SCRPKE-SCRPKD * (B):NUBER OF DATA
      00081A 100A A6 00      A        INIT10  LDA A  0,X
      00082A 100C A7 0E      A                STA A  SCRPK1-SCRPKD,X
      00083A 100E 08                          INX
      00084A 100F 5A                          DEC B
      00085A 1010 26 F8 100A                  BNE    INIT10
      00086                           *
      00087A 1012 CE 11D5    A                LDX    #SCRPK1    * INITIALIZE SCREEN
      00088A 1015 BD FF5E    A                JSR    SCRFNC     * SELECT SCREEN DEVICE
      00089A 1018 CE 11D7    A                LDX    #SCRPK2
      00090A 101B BD FF5E    A                JSR    SCRFNC     * SET SCREEN SIZE AND BUFFER ADDRESS
      00091A 101E CE 11DC    A                LDX    #SCRPK3    *
      00092A 1021 BD FF5E    A                JSR    SCRFNC     * SET CURSOR MARGIN
      00093A 1024 CE 11DE    A                LDX    #SCRPK4    *
      00094A 1027 BD FF5E    A                JSR    SCRFNC     * SET SCROLL STEP
      00095A 102A CE 11E1    A                LDX    #SCRPK5    *
      00096A 102D BD FF5E    A                JSR    SCRFNC     * SET SCROLL SPEED
      00097                           *
      00098A 1030 86 01      A                LDA A  #1         * MONITOR ON
      00099A 1032 B7 11EA    A                STA A  MONFLG
      00100                           *
      00101A 1035 CC 11F1    A                LDD    #BUF       * SET BUFFER POINTER FOR HARD COPY
      00102A 1038 FD 11EB    A                STD    BPIN
      00103A 103B FD 11ED    A                STD    BPOUT
      00104A 103E CC 0000    A                LDD    #0         * CHARACTER COUNTER = 0
      00105A 1041 FD 11EF    A                STD    BUFCNT
      00106A 1044 B7 11E4    A                STA A  PRTFLG     * HARD COPY = 'NO'
      00107                           * REWRITE SERIAL RECEIVE INTERRUPT VECTOR
      00108                           * NOTE. IF WE WANT TO SEND A CHARACTER TO THE PRINTER, WE MAY DETATCH
      00109                           *       SLAVE MPU WHILE 20 MILI SECOND AFTER WE GOT THE CHARACTER FROM
      00110                           *       SLAVE MCU.
```

```
     00111A 1047 FC 010A  A             LDD     SERVCT+1   * SAVE VECTOR ADDRESS
     00112A 104A FD 11E5  A             STD     SERADR
     00113A 104D CC 1146  A             LDD     #SERINT    * WRITE NEW INTERRUPT ADDRESS
     00114A 1050 FD 010A  A             STD     SERVCT+1
     00115                       *
     00116A 1053 CC 3D27  A             LDD     #$3D27     * SET MODE(STOP:1 CD:NO-CHECK, RTS:ON, PARITY:E
     00117                       *                         * 7 BITS LENGTH, 300 BPS)
     00118A 1056 FD 11E8  A             STD     RSPARM     * SAVE PARAMETERS
     00119A 1059 BD FF88  A             JSR     RSMST
     00120A 105C 86 01    A             LDA A   #1         * RS232C DRIVER ON
     00121A 105E BD FF85  A             JSR     RSONOF
     00122A 1061 86 01    A             LDA A   #1         * SERIAL DRIVER ON
     00123A 1063 BD FF73  A             JSR     SERONF
     00124                       *
     00125A 1066 CE 22B9  A      INIT30 LDX     #RSBUFF    * (X):BUFFER ADDRESS (SYSTEM BUFFER)
     00126A 1069 CC 1000  A             LDD     #RSBSIZ    * (A,B): BUFFER SIZE
     00127A 106C BD FF82  A             JSR     RSOPEN     * OPEN TO RECEIVE RS232C
     00128A 106F BD FF9D  A      REDKEY JSR     KEYSTS     * ACCEPT FROM KEY BOARD ?
     00129A 1072 25 7E 10F2*            BCS     BRKRTN     * IF BREAK KEY IS PRESSED, RETURN (IN BASIC MOD
     00130A 1074 27 27 109D             BEQ     RCVRS
     00131                       * ACCEPTED CHARACTER FROM KB.
     00132A 1076 BD FF9A  A             JSR     KEYIN
     00133A 1079 81 FE    A             CMP A   #$FE       * FUNCTION CODES ?
     00134A 107B 26 13 1090             BNE     GETKEY
     00135                       *
     00136                       * FUNCTION KEYS
     00137A 107D C0 F1    A             SUB B   #$F1       * F1 - F10 ?
     00138A 107F 25 1C 109D             BCS     RCVRS      *  NO, IGNORE
     00139A 1081 C1 0A    A             CMP B   #$A
     00140A 1083 24 18 109D             BCC     RCVRS
     00141A 1085 58                     ASL B
     00142A 1086 CE 10D6  A             LDX     #FNCTBL    * GET FUCTION ADDRESS
     00143A 1089 3A                     ABX
     00144A 108A EE 00    A             LDX     0,X        * (X) <-- ENTRY POINT OF EACH SUBROUTINE
     00145A 108C AD 00    A             JSR     0,X
     00146A 108E 20 0D 109D             BRA     RCVRS
     00147A 1090 BD FF76  A      GETKEY JSR     RSPUT      * TRANSMITTE CHARCTER TO RS232C.
     00148A 1093 F6 11E3  A             LDA B   ECHO       * ECHO ?
     00149A 1096 27 02 109A             BEQ     GETK10
     00150A 1098 8D 1C 10B6             BSR     PSHCHR     * PUSH RECEIVED CHARACTER TO STACK
     00151A 109A BD FF4F  A      GETK10 JSR     DSPSCR     * DISPLAY CHRACTER TO VIRTUAL SCREEN.
     00152                       *
     00153A 109D FE FFD8  A      RCVRS  LDX     $FFD8      * ARE THERE CHARACTERS IN THE RS232C BUFFER ?
     00154A 10A0 EC 00    A             LDD     0,X
     00155A 10A2 27 0F 10B3             BEQ     RCVR80
     00156A 10A4 BD FF79  A             JSR     RSGET
     00157A 10A7 81 7F    A             CMP A   #$7F
     00158A 10A9 24 08 10B3             BCC     RCVR80     * IGNORE 7F - FF CHARACTERS
     00159A 10AB F6 11EA  A             LDA B   MONFLG     * DISPLAY ON ?
     00160A 10AE 27 03 10B3             BEQ     RCVR80
     00161                       *
     00162              10B0  A      RCVR10 EQU     *
     00163A 10B0 BD FF4F  A             JSR     DSPSCR     * DISPLAY CHRACTER TO VIRTUAL SCREEN.
     00164                       *
     00165A 10B3 7E 106F  A      RCVR80 JMP     REDKEY
```

```
00166                            *
00167                            *
00168                            *
00169                            * PSH RECEIVED CHARACTER TO PRINT STACK
00170                            * ON ENTRY
00171                            *  (A): CHARACTER
00172                            * ON EXIT
00173                            * REGISTER PRESERVE
00174                            *  (A), (B)
00175                            *
00176A 10B6 7D 11E4   A          PSHCHR TST    PRTFLG     * HARD COPY = YES ?
00177A 10B9 27 1A 10D5                  BEQ    PSHC80
00178A 10BB 0F                          SEI
00179A 10BC FE 11EB   A                 LDX    BPIN       * PSH A CHARACTER TO THE STACK
00180A 10BF A7 00     A                 STA A  0,X
00181A 10C1 08                          INX
00182A 10C2 8C 21F1   A                 CPX    #BUF+BUFSIZ
00183A 10C5 26 03 10CA                  BNE    PSHC10
00184A 10C7 CE 11F1   A                 LDX    #BUF
00185A 10CA FF 11EB   A          PSHC10 STX    BPIN
00186A 10CD FE 11EF   A                 LDX    BUFCNT
00187A 10D0 08                          INX
00188A 10D1 FF 11EF   A                 STX    BUFCNT
00189A 10D4 0E                          CLI
00190A 10D5 39                   PSHC80 RTS
00191                            *
00192                            *
00193                            * FUNCTION KEY PROCEDURE TABLE
00194                            *
00195A 10D6    10EA   A          FNCTBL FDB    PFKY10     * PF1   (HARD COPY ON)
00196A 10D8    10EE   A                 FDB    PFKY20     * PF2   (HARD COPY OFF)
00197A 10DA    10F3   A                 FDB    PFKY30     * PF3   (1200 BPS)
00198A 10DC    1108   A                 FDB    PFKY40     * PF4   (300 BPS)
00199A 10DE    1130   A                 FDB    PFKY50     * PF5   (QUIT)
00200A 10E0    1115   A                 FDB    PFKY60     * PF6   (MONITOR ON)
00201A 10E2    1119   A                 FDB    PFKY70     * PF7   (MONITOR OFF)
00202A 10E4    1123   A                 FDB    PFKY80     * PF8   (ESC 'I'+$20 '1')
00203A 10E6    1145   A                 FDB    INVLKY     * PF9   (UNDEFINED)
00204A 10E8    1145   A                 FDB    INVLKY     * PF10  (UNDEFINED)
00205                            *
00206                            * PF1  PRINT(HARD COPY) ON
00207A 10EA 86 01     A          PFKY10 LDA A  #$1        * ON PRINT FLAG
00208A 10EC 20 01 10EF                  BRA    PFKY25
00209                            * PF2  PRINT (HARD COPY) OFF
00210A 10EE 4F                   PFKY20 CLR A             * OFF PRINT FLAG
00211A 10EF B7 11E4   A          PFKY25 STA A  PRTFLG
00212          10F2   A          BRKRTN EQU    *
00213A 10F2 39                          RTS
00214                            *
00215                            * PF3 1200 BPS
00216A 10F3 CC 3D47   A          PFKY30 LDD    #$3D47     * SET MODE(STOP:1 CD:NO-CHECK, RTS:ON, PARITY:E
00217                            *                        * 7 BITS LENGTH, 1200 BPS)
00218A 10F6 FD 11E8   A                 STD    RSPARM     * SAVE PARAMETERS
00219A 10F9 BD FF7F   A          PFKY35 JSR    RSCLOS     * CLOSE RS232 FOR OPEN AGAIN.
00220A 10FC FC 11E8   A                 LDD    RSPARM     * CHANGE BIT RATE
```

```
00221A 10FF BD FF88  A             JSR    RSMST
00222A 1102 38                     PULX
00223A 1103 CE 1066  A             LDX    #INIT30   * REWRITE RETURN ADDRESS
00224A 1106 3C                     PSHX
00225A 1107 39                     RTS
00226                       * PF4 300 BPS
00227A 1108 CC 3D27  A      PFKY40 LDD    #$3D27    * SET MODE(STOP:1 CD:NO-CHECK, RTS:ON, PARITY:E
00228                       *                     * 7 BITS LENGTH, 300 BPS)
00229A 110B FD 11E8  A             STD    RSPARM    * SAVE PARAMETERS
00230A 110E 86 01    A             LDA A  #1        * DISPLAY MONITOR = ON
00231A 1110 B7 11EA  A             STA A  MONFLG
00232A 1113 20 E4 10F9             BRA    PFKY35
00233                       * PF6  MONITOR ON
00234A 1115 86 01    A      PFKY60 LDA A  #1
00235A 1117 20 06 111F             BRA    PFKY75
00236                       * PF7  MONITOR OFF
00237A 1119 86 01    A      PFKY70 LDA A  #1        * HARD COPY = ON
00238A 111B B7 11E4  A             STA A  PRTFLG
00239A 111E 4F                     CLR A
00240A 111F B7 11EA  A      PFKY75 STA A  MONFLG
00241A 1122 39                     RTS
00242                       * PF8  ESC  'I'+$20 '1'
00243A 1123 86 1B    A      PFKY80 LDA A  #$1B      * ESC
00244A 1125 8D BF 1086*            BSR    PSHCHR
00245A 1127 86 69    A             LDA A  #'I+$20   * 'I'+$20
00246A 1129 8D BB 1086*            BSR    PSHCHR
00247A 112B 86 31    A             LDA A  #'1       * '1'
00248A 112D 8D B7 1086*            BSR    PSHCHR
00249A 112F 39                     RTS
00250                       * PF5  QUIT
00251A 1130 BD FF7F  A      PFKY50 JSR    RSCLOS    * CLOSE RS232
00252A 1133 4F                     CLR A            * DRIVER OFF
00253A 1134 BD FF85  A             JSR    RSONOF
00254A 1137 4F                     CLR A
00255A 1138 BD FF73  A             JSR    SERONF
00256A 113B FC 11E5  A             LDD    SERADR    * RECOVER INTERRUPT VECTOR
00257A 113E FD 010A  A             STD    SERVCT+1
00258A 1141 38                     PULX
00259A 1142 7E FF25  A             JMP    MENU
00260                       *
00261             1145  A   INVLKY EQU    *
00262A 1145 39                     RTS
00263                       *
00264                       *
00265                       * SERIAL RECEIVE INTERRUPT (RECEIVE RS232C) ROUTINE
00266                       *  PUSH RECEIVED DATA TO PRINTER STACK AND SEND THE CHARACTER WHICH IS
00267                       *   IN THE PRINTER STACK
00268                       *
00269             1146  A   SERINT EQU    *
00270A 1146 B6 11E4  A             LDA A  PRTFLG    * HARD COPY = 'YES' ?
00271A 1149 27 0F 115A             BEQ    SERI30    * NO, JUMP TO INTERRUPT ROUTINE
00272A 114B 96 11    A             LDA A  TRCSR     * GET DATA
00273A 114D 96 12    A             LDA A  SRDR
00274A 114F 84 7F    A             AND A  #$7F      * SUPRESS BIT 7
00275A 1151 81 7F    A             CMP A  #$7F
```

```
00276A 1153 24 03 1158           BCC    SERI30    * IGNORE 7F - FF
00277A 1155 BD 10B6   A          JSR    PSHCHR
00278                     * HARD COPY ON
00279A 1158 8D 05 115F    SERI30 BSR    HRDCPY    * SEND 3 CHARACTERS (9 MILI SEC)
00280                     *
00281A 115A FE 11E5   A   SERI80 LDX    SERADR
00282A 115D 6E 00     A          JMP    0,X
00283                     *
00284                     * PRINT TO SERIAL PRINTER
00285                     *  THIS ROUTINE CALLED ONLY IN INTERRUPT
00286                     *  REGISTER PRESERVE
00287                     *    (A)
00288                     *
00289A 115F 36            HRDCPY PSH A
00290A 1160 B6 11E4   A          LDA A  PRTFLG    * HARD COPY = 'YES' ?
00291A 1163 27 60 11C5           BEQ    HARD80
00292                     * YES, PRINTING
00293A 1165 86 03     A          LDA A  #3        * COPY COUNT = 3 (PRINT 3 CHARACTERS)
00294A 1167 B7 11E7   A          STA A  CPYCNT
00295A 116A 7B 4002   A          TIM    #$40,PORT1 * PRINTER READY ?
00296A 116D 26 56 11C5           BNE    HARD80
00297                     * ARE THERE DATA IN THE BUFFER ?
00298A 116F FC 11EF   A          LDD    BUFCNT
00299A 1172 27 51 11C5           BEQ    HARD80
00300                     *
00301A 1174 71 FB03   A          AIM    #$FF-4,PORT2 * DETACH SLAVE MCU, (SELECT SERIAL)
00302A 1177 96 11     A          LDA A  TRCSR
00303A 1179 36                   PSH A            * SAVE TRCSR
00304A 117A 86 05     A          LDA A  #$05      * 4800 BPS
00305A 117C 97 10     A          STA A  RMCR
00306A 117E 86 0A     A          LDA A  #$0A
00307A 1180 97 11     A          STA A  TRCSR
00308                     *
00309A 1182 FE 11ED   A   HARD10 LDX    BPOUT     * LOAD DATA FROM THE STACK
00310A 1185 A6 00     A          LDA A  0,X
00311A 1187 08                   INX
00312A 1188 8C 21F1   A          CPX    #BUF+BUFSIZ
00313A 118B 26 03 1190           BNE    HARD20
00314A 118D CE 11F1   A          LDX    #BUF
00315A 1190 FF 11ED   A   HARD20 STX    BPOUT     * INCREMENT DATA POINTER AT THE BUFFER
00316A 1193 FE 11EF   A          LDX    BUFCNT
00317A 1196 09                   DEX
00318A 1197 FF 11EF   A          STX    BUFCNT
00319                     *
00320A 119A 7B 2011   A   HARD30 TIM    #$20,TRCSR * WAIT READY.
00321A 119D 27 FB 119A           BEQ    HARD30
00322A 119F 97 13     A          STA A  STDR      * STORE DATA TO THE TRANSMIT REGISTER.
00323                     *
00324A 11A1 7A 11E7   A          DEC    CPYCNT    * WERE 3 CHARACTERS SENDED ?
00325A 11A4 27 0A 11B0           BEQ    HARD40
00326A 11A6 7B 4002   A          TIM    #$40,PORT1 * PRINTER READY ?
00327A 11A9 26 05 11B0           BNE    HARD40
00328A 11AB FC 11EF   A          LDD    BUFCNT    * IS BUFFER EMPTY ?
00329A 11AE 26 D2 1182           BNE    HARD10
00330                     *
```

```
      00331                                 *  WAIT 2 MILI SEC (TIME OF SENDING ONE CHARACTER)
      00332A 11B0 7B 2011  A                HARD40 TIM    #$20,TRCSR
      00333A 11B3 27 FB 11B0                       BEQ    HARD40
      00334A 11B5 CE 0190  A                       LDX    #400
      00335A 11B8 09                        HARD50 DEX
      00336A 11B9 26 FD 11B8                       BNE    HARD50
      00337                                 * RECOVER SERIAL COMMUNICATION
      00338A 11BB 86 04    A                       LDA A  #$04        * SELECT SLAVE MCU
      00339A 11BD 97 10    A                       STA A  RMCR
      00340A 11BF 32                               PUL A              * RECOVER TRCSR
      00341A 11C0 97 11    A                       STA A  TRCSR
      00342A 11C2 72 0403  A                       OIM    #$4,PORT2
      00343A 11C5 32                        HARD80 PUL A
      00344A 11C6 39                               RTS
      00345                                 *
      00346                                 *
      00347A 11C7     84   A                SCRPK0 FCB    $84         * SCREEN DEVICE SELECT (LCD)
      00348A 11C8     22   A                       FCB    $22
      00349                                 *
      00350A 11C9     87   A                       FCB    $87         * SET SCREEN SIZE AND BUFFER ADDRESS
      00351A 11CA     13   A                       FCB    19,3
           A 11CB     03   A
      00352A 11CC     21F1 A                       FDB    SCRBUF
      00353                                 *
      00354A 11CE     C3   A                       FCB    $C3         * SET CURSOR MARGIN
      00355A 11CF     04   A                       FCB    4
      00356                                 *
      00357A 11D0     C4   A                       FCB    $C4         * SET SCROLL STEP
      00358A 11D1     0A   A                       FCB    10          * X
      00359A 11D2     03   A                       FCB    3           * Y
      00360                                 *
      00361A 11D3     CB   A                       FCB    $CB         * SET SCROLL SPEED
      00362A 11D4     09   A                       FCB    9
      00363                                 *
      00364           11D5 A                SCRPKE EQU    *
      00365                                 *
      00366                                 *
      00367                                 * WORK AREA
      00368A 11D5     84   A                SCRPK1 FCB    $84         * SCREEN DEVICE SELECT (LCD)
      00369A 11D6     22   A                       FCB    $22
      00370A 11D7     87   A                SCRPK2 FCB    $87         * SET SCREEN SIZE AND BUFFER ADDRESS
      00371A 11D8     13   A                       FCB    19,3
           A 11D9     03   A
      00372A 11DA     21F1 A                       FDB    SCRBUF
      00373                                 *
      00374A 11DC     C3   A                SCRPK3 FCB    $C3         * SET CURSOR MARGIN
      00375A 11DD     04   A                       FCB    4
      00376                                 *
      00377A 11DE     C4   A                SCRPK4 FCB    $C4         * SET SCROLL STEP
      00378A 11DF     0A   A                       FCB    10          * X
      00379A 11E0     03   A                       FCB    3           * Y
      00380                                 *
      00381A 11E1     CB   A                SCRPK5 FCB    $CB         * SET SCROLL SPEED
      00382A 11E2     09   A                       FCB    9
      00383                                 *
```

```
     00384A 11E3    01     A      ECHO    FCB     1         * TERMINAL MODE ECHO.
     00385                        *
     00386A 11E4    0001   A      PRTFLG  RMB     1         * HARD COPY (MP-80 PRINTER) ON/OFF FLAG
     00387                        *                       * 0:OFF  1:ON
     00388                        *
     00389A 11E5    0002   A      SERADR  RMB     2         * SAVE SERIAL RECEIVE INTERRUPT VECTOR.
     00390A 11E7    0001   A      CPYCNT  RMB     1         * WORK FOR HARD COPY
     00391A 11E8    0002   A      RSPARM  RMB     2         * RS232C OPEN PARAMETER
     00392A 11EA    0001   A      MONFLG  RMB     1         * DISPLAY RECEIVED CHARACTER = YES ?
     00393                        *                       *  1:DISPLAY  0:DISPLAY OFF
     00394                        *
      0395                        * SERIAL SEND BUFFER
     00396A 11EB    0002   A      BPIN    RMB     2         * POINTER WHERE NEXT CHARACTER IS STORED
     00397A 11ED    0002   A      BPOUT   RMB     2         * POINTER WHERE NEXT CHARACTER IS LOADED
     00398A 11EF    0002   A      BUFCNT  RMB     2         * NUMBER OF CHARACTERS IN THE BUFFER
     00399A 11F1    1000   A      BUF     RMB     BUFSIZ    * BUFFER
     00400                        *
     00401A 21F1    00C8   A      SCRBUF  RMB     SCBSIZ    * SCREEN BUFFER
     00402A 2289    1000   A      RSBUFF  RMB     RSBSIZ    * RS232C RECEIVE BUFFER
     00403          32B9   A      WRKEND  EQU     *
     00404                        *
     00405                        *
     00406          0000   A              END
***** TOTAL ERRORS      0
```

```
      *** 6301 CROSS MACROASSEMBLER   VER1.0 ***                                    10/22/82     14:42:57
ERR    SEQ    LOC  OBJECT        PROGRAM  MODEM        --- CONTROL HALF DUPLEX MODEM ---

     00001                                *
     00002                                       NAM     MODEM
     00003                                       TTL     --- CONTROL HALF DUPLEX MODEM ---
     00004                                *
     00005                                * TSS TERMINAL OF HALF DUPLEX MODEM
     00006                                *    WITHOUIT HARD COPY
     00007                                * FILE NAME     'EX38'    BY  K.A
     00008                                       OPT     LOAD
     00009                                       OPT     PAGE=55
     00010                                *
     00011                                *  CONTROL HALF DUPLEX MODEM
     00012                                *
     00013             FF19   A           SNSCOM EQU     $FF19
     00014             FF16   A           CHKRS  EQU     $FF16
     00015                                *
     00016                                *
     00017                                * CONSTANT VALUE
     00018             00FD   A           RSPRM1 EQU     $FD          * STOP BITS = 1, CARRIER DETECT:CHECK
     00019                                *                           * RTS:LOW,   CTS:CHECK  DSR:CHECK
     00020                                *                           * PARITY:NONE
     00021             0048   A           RSPRM2 EQU     $48          * BIT LENGTH = 8, BIT RATE = 1200 BIT/SEC
     00022                                *                           *
     00023                                *
     00024A 1000                                 ORG     $1000
     00025                                * ENTRY POINT OF 'START RS232C COMMUNICATION'
     00026.                               * PROCEDURE
     00027                                *  1:RTS LOW, SET BIT RATE, DRIVER ON
     00028                                *  2:START TO RECEIVE
     00029                                *
     00030                                *  PARAMETER
     00031                                *   ON ENTRY.   NONE
     00032                                *   ON EXIT.    NONE
     00033                                *
     00034                                *
     00035                                * SUBROUTINE ENTRY POINT
     00036             FF4F   A           DSPSCR EQU     $FF4F        = DISPLAY ONE CHARACTER TO VIRTUAL SCREEN
     00037             FF5E   A           SCRFNC EQU     $FF5E        * VIRTUAL SCREEN FUNCTION
     00038             FF85   A           RSONOF EQU     $FF85        * RS232C DRIVER ON/OFF
     00039             FF88   A           RSMST  EQU     $FF88        * SET RS232C PARAMETERS
     00040             FF82   A           RSOPEN EQU     $FF82        * OPEN RS232C RECEIVE
     00041             FF7F   A           RSCLOS EQU     $FF7F        * CLOSE RS232C RECEIVE
     00042             FF79   A           RSGET  EQU     $FF79        * GET RS232C ONE CHARACTER
     00043             FF76   A           RSPUT  EQU     $FF76        * SEND RS232C ONE CHARACTER
     00044             FF9A   A           KEYIN  EQU     $FF9A        * GET ONE CHARACTER FROM KEYBOARD BUFFER
     00045             FF9D   A           KEYSTS EQU     $FF9D        * GET NUMBER OF CHARACTERS IN THE KEY BUFFER
     00046                                * CONSTANTS OR REGISTERS
     00047             0002   A           PORT1  EQU     $02          * I/O PORT1
     00048             0003   A           PORT2  EQU     $03          * I/O PORT2
     00049             1000   A           RSBSIZ EQU     4096         * BUFFER SIZE FOR RS232C RECEIVE
     00050             0055   A           SCBSIZ EQU     85           * BUFFER SIZE FOR SCREEN
     00051             0001   A           ECHOOT EQU     1            * TERMINAL MODE = 'ECHO CHARACTER' ?
     00052                                *                          * 0:YES,  1:NO
     00053                                *
     00054A 1000                                 ORG     $1000
     00055                                *
```

```
    00056                          * INITIALIZE
    00057A 1000 CE 1002  A                LDX     #SCRPKD    * SET SCREEN PACKET  X:DATA ADDRESS
    00058A 1003 C6 0E    A                LDA B   #SCRPKE-SCRPKD * (B):NUBER OF DATA
    00059A 1005 A6 00    A         INIT10 LDA A   0,X
    00060A 1007 A7 0E    A                STA A   SCRPK1-SCRPKD,X
    00061A 1009 08                        INX
    00062A 100A 5A                        DEC B
    00063A 100B 26 F8 1005                BNE     INIT10
    00064                          *
    00065A 100D CE 10E0  A                LDX     #SCRPK1    * INITIALIZE SCREEN
    00066A 1010 BD FF5E  A                JSR     SCRFNC     * SELECT SCREEN DEVICE
    00067A 1013 CE 10E2  A                LDX     #SCRPK2
    00068A 1016 BD FF5E  A                JSR     SCRFNC     * SET SCREEN SIZE AND BUFFER ADDRESS
    00069A 1019 CE 10E7  A                LDX     #SCRPK3    *
    00070A 101C BD FF5E  A                JSR     SCRFNC     * SET CURSOR MARGIN
    00071A 101F CE 10E9  A                LDX     #SCRPK4    *
    00072A 1022 BD FF5E  A                JSR     SCRFNC     * SET SCROLL STEP
    00073A 1025 CE 10EC  A                LDX     #SCRPK5    *
    00074A 1028 BD FF5E  A                JSR     SCRFNC     * SET SCROLL SPEED
    00075                          *
    00076A 102B CC 3547  A                LDD     #$3547     * SET MODE(STOP:1 CD:NO-CHECK, RTS:OFF, PARITY:E
    00077                          *                       * 7 BITS LENGTH, 1200 BPS)
    00078A 102E BD FF88  A                JSR     RSMST
    00079A 1031 86 01    A                LDA A   #1         * RS232C DRIVER ON
    00080A 1033 BD FF85  A                JSR     RSONOF
    00081A 1036 CE 1151  A                LDX     #RSBUF     * (X):BUFFER ADDRESS
    00082A 1039 CC 1000  A                LDD     #RSBSIZ    * (A,B): BUFFER SIZE
    00083A 103C BD FF82  A                JSR     RSOPEN     * OPEN RS232C RECEIVE
    00084                          *
    00085A 103F BD FF9D  A         REDKEY JSR     KEYSTS     * ACCEPT FROM KEY BOARD ?
    00086A 1042 25 29 106D                BCS     BRKRTN     * IF BREAK KEY IS PRESSED, RETURN (IN BASIC MODE
    00087A 1044 27 14 105A                BEQ     RCVRS
    00088                          * ACCEPTED CHARACTER FROM KB.
    00089A 1046 BD FF9A  A                JSR     KEYIN
    00090             1049  A      GETKEY EQU     *
    00091A 1049 36                        PSH A
    00092A 104A BD FF4F  A                JSR     DSPSCR     * DISPLAY CHRACTER TO VIRTUAL SCREEN.
    00093A 104D 32                        PUL A
    00094A 104E 81 0D    A                CMP A   #$0D       * CR (SEND DATA) CODE ?
    00095A 1050 26 08 105A                BNE     RCVRS
    00096A 1052 BD 106E  A                JSR     TXD        * TRANSMITTE DATA STRING TO RS232C
    00097A 1055 86 0A    A                LDA A   #$0A       * DISPLAY 'LF'
    00098A 1057 BD FF4F  A                JSR     DSPSCR
    00099                          *
    00100A 105A FE FFD8  A         RCVRS  LDX     $FFD8      * RECEIVED CHARACTER FROM RS232C ?
    00101A 105D EC 00    A                LDD     0,X
    00102A 105F 27 DE 103F                BEQ     REDKEY
    00103A 1061 BD FF79  A                JSR     RSGET
    00104A 1064 81 7F    A                CMP A   #$7F
    00105A 1066 24 D7 103F                BCC     REDKEY     * IGNORE 7F - FF CHARACTERS
    00106A 1068 BD FF4F  A                JSR     DSPSCR
    00107A 106B 20 D2 103F                BRA     REDKEY
    00108                          *
    00109A 106D 39                 BRKRTN RTS
    00110                          *
```

```
     00166                              *
     00167A 10C4 86 4D     A                   LDA A   #$4D       * RTS:LOW
     00168A 10C6 BD FF19   A                   JSR     SNSCOM
     00169A 10C9 86 00     A                   LDA A   #$00
     00170A 10CB BD FF19   A                   JSR     SNSCOM
     00171                              *
     00172A 10CE BD FF16   A                   JSR     CHKRS      * RESTART RECEIVING
     00173                              *
     00174A 10D1 39                            RTS
     00175                              *
     00176                              *
     00177                              *
     00178A 10D2      84   A            SCRPKD FCB     $84        * SCREEN DEVICE SELECT (LCD)
     00179A 10D3      22   A                   FCB     $22
     00180                              *
     00181A 10D4      87   A                   FCB     $87        * SET SCREEN SIZE AND BUFFER ADDRESS
     00182A 10D5      13   A                   FCB     19,3
          A 10D6      03   A
     00183A 10D7      2151 A                   FDB     SCRBUF
     00184                              *
     00185A 10D9      C3   A                   FCB     $C3        * SET CURSOR MARGIN
     00186A 10DA      04   A                   FCB     4
     00187                              *
     00188A 10DB      C4   A                   FCB     $C4        * SET SCROLL STEP
     00189A 10DC      0A   A                   FCB     10         * X
     00190A 10DD      03   A                   FCB     3          * Y
     00191                              *
     00192A 10DE      C8   A                   FCB     $C8        * SET SCROLL SPEED
     00193A 10DF      09   A                   FCB     9
     00194                              *
     00195            10E0 A            SCRPKE EQU     *
     00196                              *
     00197                              *
     00198                              * WORK AREA
     00199A 10E0      84   A            SCRPK1 FCB     $84        * SCREEN DEVICE SELECT (LCD)
     00200A 10E1      22   A                   FCB     $22
     00201A 10E2      87   A            SCRPK2 FCB     $87        * SET SCREEN SIZE AND BUFFER ADDRESS
     00202A 10E3      13   A                   FCB     19,3
          A 10E4      03   A
     00203A 10E5      2151 A                   FDB     SCRBUF
     00204                              *
     00205A 10E7      C3   A            SCRPK3 FCB     $C3        * SET CURSOR MARGIN
     00206A 10E8      04   A                   FCB     4
     00207                              *
     00208A 10E9      C4   A            SCRPK4 FCB     $C4        * SET SCROLL STEP
     00209A 10EA      0A   A                   FCB     10         * X
     00210A 10EB      03   A                   FCB     3          * Y
     00211                              *
     00212A 10EC      C8   A            SCRPK5 FCB     $C8        * SET SCROLL SPEED
     00213A 10ED      09   A                   FCB     9
     00214                              *
     00215A 10EE      91   A            SCRPK7 FCB     $91        * GET EXTENT OF VIRTUAL SCREEN.
     00216A 10EF      0004 A                   RMB     4
     00217                              *
     00218A 10F3      97   A            SCRPK8 FCB     $97
```

```
ERR   SEQ    LOC  OBJECT      PROGRAM  MODEM      --- CONTROL HALF DUPLEX MODEM ---

    00219A 10F4     005A   A                RMB    90
    00220                         *
    00221A 114E     01     A          ECHO    FCB    1         * TERMINAL MODE ECHO.
    00222                         *
    00223A 114F     0001   A          PRTFLG  RMB    1         * HARD COPY (MP-80 PRINTER) ON/OFF FLAG
    00224                         *                          * 0:OFF   1:ON
    00225A 1150     0001   A          TXCNT   RMB    1         * NUMBER OF CHARACTERS WHICH ARE SENT TO HOST
    00226                         *                          * COMPUTER
    00227                         *
    00228A 1151     1000   A          RSBUF   RMB    RSBSIZ    * RS232C RECEIVE BUFFER
    00229                         *
    00230A 2151     0055   A          SCRBUF  RMB    SCBSIZ    * SCREEN BUFFER
    00231                         *
    00232                         *
    00233           0000   A                END
***** TOTAL ERRORS      0
```