```
.PAGE   85
.TITLE  "TURBODOS OPERATING SYSTEM SERIAL/PARALLEL DRIVER (MuSYS NET/82)"
.SBTTL  "COPYRIGHT (C) 1982, SOFTWARE 2000, INC."
;
; COPYRIGHT (C) 1982, SOFTWARE 2000, INC.
;
; AUTHORS: RONALD E. RAIKES
;          MICHAEL D. BUSCH
;
; VERSION: 06/21/82
;
.IDENT  SPDN82          ;MODULE ID
;
.INSERT DREQUATE        ;DRIVER SYMBOLIC EQUIVALENCES
;
SIOVEC  = 20H           ;SIO INTERRUPT VECTOR ADDRESS
;
SIOADR  = 00H           ;SIO PORT A DATA REGISTER
SIOACR  = 01H           ;SIO PORT A CONTROL REGISTER
SIOBDR  = 02H           ;SIO PORT B DATA REGISTER
SIOBCR  = 03H           ;SIO PORT B CONTROL REGISTER
;
RDA     = 0             ;RECEIVED DATA AVAILABLE BIT
TBE     = 2             ;TRANSMIT BUFFER EMPTY BIT
DCD     = 3             ;DATA CARRIER DETECT BIT
CTS     = 5             ;CLEAR TO SEND BIT
;
TIM0    = 10H           ;TIMER 0 DATA REGISTER
TIM1    = 11H           ;TIMER 1 DATA REGISTER
TIM2    = 12H           ;TIMER 2 DATA REGISTER
TIMCTL  = 13H           ;TIMER CONTROL REGISTER
;
T0CMD   = 36H           ;TIMER 0 COMMAND
T1CMD   = 76H           ;TIMER 1 COMMAND
T2CMD   = 0B6H          ;TIMER 2 COMMAND
;
        .LOC    .DATA.# ;LOCATE IN DATA AREA
;
S0IBSZ::.WORD   64      ;SERIAL 0 INPUT BUFFER SIZE
S0IBUF: .WORD   0       ;SERIAL 0 INPUT BUFFER ADDRESS
S0IPTR: .WORD   0       ;SERIAL 0 INPUT POINTER
S0OPTR: .WORD   0       ;SERIAL 0 OUTPUT POINTER
S0ICNT: .WORD   0       ;SERIAL 0 INPUT COUNT
S0IWCT: .BYTE   0       ;SERIAL 0 INPUT WAIT COUNT
S0OCHR: .BYTE   0       ;SERIAL 0 OUTPUT CHARACTER
S0BR:   .BYTE   0       ;SERIAL 0 BAUD RATE CODE
;
S0ISPH:                 ;SERIAL 0 INPUT SEMAPHORE
        .WORD   0       ;SEMAPHORE COUNT
..S0IH: .WORD   ..S0IH  ;SEMAPHORE P/D HEAD
        .WORD   ..S0IH
;
                        ;SERIAL 0 OUTPUT SEMAPHORE
S0OSPH: .WORD   0       ;SEMAPHORE COUNT
..S0OH: .WORD   ..S0OH  ;SEMAPHORE P/D HEAD
        .WORD   ..S0OH
;
                        ;SERIAL 0 OUTPUT SEMAPHORE
S0XSPH: .WORD   1       ;SEMAPHORE COUNT
..S0XH: .WORD   ..S0XH  ;SEMAPHORE P/D HEAD
        .WORD   ..S0XH
;
S1IBSZ::.WORD   16      ;SERIAL 1 INPUT BUFFER SIZE
S1IBUF: .WORD   0       ;SERIAL 1 INPUT BUFFER ADDRESS
```

```
S1IPTR: .WORD   0       ;SERIAL 1 INPUT POINTER
S1OPTR: .WORD   0       ;SERIAL 1 OUTPUT POINTER
S1ICNT: .WORD   0       ;SERIAL 1 INPUT COUNT
S1IWCT: .BYTE   0       ;SERIAL 1 INPUT WAIT COUNT
S1OCHR: .BYTE   0       ;SERIAL 1 OUTPUT CHARACTER
S1BR:   .BYTE   0       ;SERIAL 1 BAUD RATE CODE
;
                        ;SERIAL 1 INPUT SEMAPHORE
S1ISPH: .WORD   0       ;SEMAPHORE COUNT
..S1IH: .WORD   ..S1IH  ;SEMAPHORE P/D HEAD
        .WORD   ..S1IH
;
                        ;SERIAL 1 OUTPUT SEMAPHORE
S1OSPH: .WORD   0       ;SEMAPHORE COUNT
..S1OH: .WORD   ..S1OH  ;SEMAPHORE P/D HEAD
        .WORD   ..S1OH
;
                        ;SERIAL 1 OUTPUT SEMAPHORE
S1XSPH: .WORD   1       ;SEMAPHORE COUNT
..S1XH: .WORD   ..S1XH  ;SEMAPHORE P/D HEAD
        .WORD   ..S1XH
;
        .LOC    .INIT.# ;LOCATE IN INITIALIZATION AREA
;
SPINIT::LXI     H,SIOISR  ;GET SIO INTERRUPT SERVICE ADDR
        SHLD    SIOVEC  ;SET SIO INTERRUPT VECTOR ADDRESS
        LXI     H,SIOPGM  ;GET SIO PROGRAM LIST
        LXI     B,SIOAPL<8!SIOACR  ;B=LENGTH/C=CONTROL REG
        OUTIR           ;PROGRAM SIO PORT A
        LXI     H,SIOPGM  ;GET SIO PROGRAM LIST
        LXI     B,SIOBPL<8!SIOBCR  ;B=LENGTH/C=CONTROL REG
        OUTIR           ;PROGRAM SIO PORT B
        LHLD    S0IBSZ  ;GET SERIAL 0 INPUT BUFFER SIZE
        CALL    ALLOC#  ;ALLOCATE PACKET FOR SERIAL BUFFER
        SHLD    S0IBUF  ;SAVE SERIAL 0 INPUT BUFFER ADDRESS
        SHLD    S0IPTR  ;SET SERIAL 0 INPUT POINTER
        SHLD    S0OPTR  ;SET SERIAL 0 OUTPUT POINTER
        LHLD    S1IBSZ  ;GET SERIAL 1 INPUT BUFFER SIZE
        CALL    ALLOC#  ;ALLOCATE PACKET FOR SERIAL BUFFER
        SHLD    S1IBUF  ;SAVE SERIAL 1 INPUT BUFFER ADDRESS
        SHLD    S1IPTR  ;SET SERIAL 1 INPUT POINTER
        SHLD    S1OPTR  ;SET SERIAL 1 OUTPUT POINTER
        RET             ;DONE
;
SIOPGM: .BYTE   18H     ;RESET CHANNEL
        .BYTE   4       ;SELECT WR4
        .BYTE   44H     ;WRITE REGISTER 4 CONTROL WORD
        .BYTE   5       ;SELECT WR5
        .BYTE   0EAH    ;WRITE REGISTER 5 CONTROL WORD
        .BYTE   3       ;SELECT WR3
        .BYTE   0C1H    ;WRITE REGISTER 3 CONTROL WORD
        .BYTE   1       ;SELECT WR1
        .BYTE   10H     ;WRITE REGISTER 1 CONTROL WORD
;
SIOAPL  = .-SIOPGM      ;SIO PORT A PROGRAM LENGTH
;
        .BYTE   2       ;SELECT WR2
        .BYTE   SIOVEC  ;WRITE REGISTER 1 CONTROL WORD
;
SIOBPL  = .-SIOPGM      ;SIO PORT B PROGRAM LENGTH
;
        .LOC    .PROG.# ;LOCATE IN PROGRAM AREA
;
SERIAL::
COMDRV::MOV     A,E     ;GET FUNCTION NUMBER
        ORA     A       ;FUNCTION NUMBER=0?
```

```
        DCR     A       ;FUNCTION NUMBER=1?
        JRZ     SERIN   ;IF SO, CONTINUE
        DCR     A       ;FUNCTION NUMBER=2?
        JZ      SEROUT  ;IF SO, CONTINUE
        DCR     A       ;FUNCTION NUMBER=3?
        JZ      SERSBR  ;IF SO, CONTINUE
        DCR     A       ;FUNCTION NUMBER=4?
        JZ      SERRBR  ;IF SO, CONTINUE
        DCR     A       ;FUNCTION NUMBER=5?
        JZ      SERSMC  ;IF SO, CONTINUE
        DCR     A       ;FUNCTION NUMBER=6?
        JZ      SERRMC  ;IF SO, CONTINUE
        RET             ;ELSE, DONE
;
SERST:  LDED    SOICNT  ;GET SERIAL 0 INPUT BUFFER COUNT
        LHLD    SOOPTR  ;GET SERIAL 0 OUTPUT POINTER
        MOV     A,B     ;GET CHANNEL NUMBER
        ORA     A       ;CHANNEL NUMBER=0
        JRZ     ..COM   ;IF SO, CONTINUE
        LDED    S1ICNT  ;GET SERIAL 1 INPUT BUFFER COUNT
        LHLD    S1OPTR  ;GET SERIAL 1 OUTPUT POINTER
..COM:  MOV     A,D
        ORA     E       ;SERIAL INPUT BUFFER COUNT=0?
        RZ              ;IF SO, DONE
        MVI     A,0FFH  ;ELSE, SET RETURN CODE=0FFH
        MOV     C,M     ;GET SERIAL INPUT CHARACTER
        RET             ;DONE
;
SERIN:  MOV     A,B     ;GET CHANNEL NUMBER
        ORA     A       ;CHANNEL NUMBER=0?
        JRNZ    ..S1I   ;IF NOT, CONTINUE
..SOI:  DI              ;ELSE, DISABLE INTERRUPTS
        LHLD    SOICNT  ;GET SERIAL 0 INPUT COUNT
        MOV     A,H
        ORA     L       ;SERIAL 0 INPUT COUNT=0?
        JRZ     ..WTO   ;IF SO, CONTINUE
        DCX     H       ;DECREMENT SERIAL 0 INPUT COUNT
        SHLD    SOICNT  ;UPDATE SERIAL 0 INPUT COUNT
        LHLD    SOOPTR  ;GET SERIAL 0 OUTPUT POINTER
        MOV     A,M     ;GET CHARACTER FROM BUFFER
        INX     H       ;INCREMENT SERIAL 0 OUTPUT POINTER
        XCHG            ;SERIAL 0 OUTPUT POINTER TO DE-REG
        LHLD    SOIBSZ  ;GET SERIAL 0 INPUT BUFFER SIZE
        DCX     H       ;DECREMENT INPUT BUFFER SIZE
        LBCD    SOIBUF  ;GET SERIAL 0 INPUT BUFFER ADDRESS
        DAD     B       ;CALC LAST INPUT BUFFER ADDRESS
        DSBC    D       ;BUFFER WRAP-AROUND?
        JRNC    ..NWA0  ;IF NOT, CONTINUE
        MOV     E,C     ;GET SERIAL 0 INPUT BUFFER ADDRESS
        MOV     D,B
..NWA0: SDED    SOOPTR  ;UPDATE SERIAL 0 OUTPUT POINTER
        EI              ;ENABLE INTERRUPTS
        RET             ;DONE
..WTO:  LXI     H,SOIWCT  ;GET SERIAL 0 INPUT WAIT COUNT
        INR     M       ;INCREMENT INPUT WAIT COUNT
        LXI     H,SOISPH  ;GET SERIAL 0 INPUT SEMAPHORE
        CALL    WAIT#   ;WAIT FOR CONSOLE INPUT
        JMPR    ..SOI   ;CONTINUE
..S1I:  DI              ;DISABLE INTERRUPTS
        LHLD    S1ICNT  ;GET SERIAL 1 INPUT COUNT
        MOV     A,H
        ORA     L       ;SERIAL 1 INPUT COUNT=0?
        JRZ     ..WT1   ;IF SO, CONTINUE
        DCX     H       ;DECREMENT SERIAL 1 INPUT COUNT
        SHLD    S1ICNT  ;UPDATE SERIAL 1 INPUT COUNT
```

```
        LHLD    S1OPTR  ;GET SERIAL 1 OUTPUT POINTER
        MOV     A,M     ;GET CHARACTER FROM BUFFER
        INX     H       ;INCREMENT SERIAL 1 OUTPUT POINTER
        XCHG            ;SERIAL 1 OUTPUT POINTER TO DE-REG
        LHLD    S1IBSZ  ;GET SERIAL 1 INPUT BUFFER SIZE
        DCX     H       ;DECREMENT INPUT BUFFER SIZE
        LBCD    S1IBUF  ;GET SERIAL 1 INPUT BUFFER ADDRESS
        DAD     B       ;CALC LAST INPUT BUFFER ADDRESS
        DSBC    D       ;BUFFER WRAP-AROUND?
        JRNC    ..NWA1  ;IF NOT, CONTINUE
        MOV     E,C     ;GET SERIAL 1 INPUT BUFFER ADDRESS
        MOV     D,B
..NWA1: SDED    S1OPTR  ;UPDATE SERIAL 1 OUTPUT POINTER
        EI              ;ENABLE INTERRUPTS
        RET             ;DONE
..WT1:  LXI     H,S1IWCT  ;GET SERIAL 1 INPUT WAIT COUNT
        INR     M       ;INCREMENT INPUT WAIT COUNT
        LXI     H,S1ISPH  ;GET SERIAL 1 INPUT SEMAPHORE
        CALL    WAIT#   ;WAIT FOR CONSOLE INPUT
        JMPR    ..S1I   ;CONTINUE
;
SEROUT: MOV     A,B     ;GET CHANNEL NUMBER
        ORA     A       ;CHANNEL NUMBER=1?
        JRNZ    ..S1O   ;IF SO, CONTINUE
        LXI     H,SOXSPH  ;GET SERIAL 0 OUT SEMAPHORE
        PUSH    H       ;SAVE SERIAL 0 OUT SEMAPHORE
        CALL    WAIT#   ;WAIT ON MUTUAL EXCLUSION
        LXI     H,SOOCHR  ;GET SERIAL 0 OUTPUT CHARACTER
        MOV     M,C     ;SAVE OUTPUT CHARACTER
        LXI     D,SOOPOL  ;GET SERIAL 0 OUT POLL ROUTINE
        CALL    LNKPOL# ;CREATE POLL ROUTINE
        CALL    SOOPR   ;EXECUTE POLL ROUTINE
        LXI     H,SOOSPH  ;GET SERIAL 0 OUT SEMAPHORE
        CALL    WAIT#   ;DISPATCH IF NECESSARY
        POP     H       ;GET MUTUTAL EXCLUSION SEMAPHORE
        JMP     SIGNAL# ;SIGNAL PROCESS AS READY
..S1O:  LXI     H,S1XSPH  ;GET MUTUAL EXCLUSION SEMAPHORE
        PUSH    H       ;SAVE MUTUAL EXCLUSION SEMAPHORE
        CALL    WAIT#   ;WAIT ON MUTUAL EXCLUSION
        LXI     H,S1OCHR  ;GET SERIAL 1 OUTPUT CHARACTER
        MOV     M,C     ;SAVE OUTPUT CHARACTER
        LXI     D,S1OPOL  ;GET SERIAL 1 OUT POLL ROUTINE
        CALL    LNKPOL# ;CREATE POLL ROUTINE
        CALL    S1OPR   ;EXECUTE POLL ROUTINE
        LXI     H,S1OSPH  ;GET SERIAL 1 OUT SEMAPHORE
        CALL    WAIT#   ;DISPATCH IF NECESSARY
        POP     H       ;GET MUTUTAL EXCLUSION SEMAPHORE
        JMP     SIGNAL# ;SIGNAL PROCESS AS READY
;
SOOPOL:                 ;SERIAL 0 OUTPUT POLL ROUTINE
        .WORD   0       ;SUCCESSOR LINK POINTER
        .WORD   0       ;PREDECESSOR LINK POINTER
;
SOOPR:  MVI     A,10H   ;GET RESET EXTERNAL STATUS COMMAND
        OUT     SIOACR  ;RESET EXTERNAL STATUS
        IN      SIOACR  ;GET SIO PORT A STATUS
        BIT     TBE,A   ;TRANSMIT BUFFER EMPTY?
        RZ              ;IF NOT, DONE
        LXI     H,SOBR  ;ELSE, GET SERIAL 0 BAUD RATE CODE
        BIT     6,M     ;CTS HANDSHAKING REQUESTED?
        JRZ     ..NCTS  ;IF NOT, CONTINUE
        BIT     CTS,A   ;ELSE, CHECK CLEAR TO SEND STATUS
        RZ              ;IF CLEAR TO SEND FALSE, DONE
..NCTS: LDA     SOOCHR  ;GET SERIAL 0 OUTPUT CHARACTER
        OUT     SIOADR  ;OUTPUT CHARACTER
        LXI     H,SOOPOL  ;GET SERIAL 0 OUT POLL ROUTINE
```

```
        CALL    UNLINK# ;UNLINK POLL ROUTINE
        LXI     H,S0OSPH  ;GET SERIAL 0 OUT SEMAPHORE
        JMP     SIGNAL# ;SIGNAL PROCESS AS READY
;
S1OPOL: .WORD   0       ;SUCCESSOR LINK POINTER
        .WORD   0       ;PREDECESSOR LINK POINTER
;
S1OPR:  MVI     A,10H   ;GET RESET EXTERNAL STATUS COMMAND
        OUT     SIOBCR  ;RESET EXTERNAL STATUS
        IN      SIOBCR  ;GET SIO PORT B STATUS
        BIT     TBE,A   ;TRANSMIT BUFFER EMPTY?
        RZ              ;IF NOT, DONE
        LXI     H,S1BR  ;ELSE, GET SERIAL 1 BAUD RATE CODE
        BIT     6,M     ;CTS HANDSHAKING REQUESTED?
        JRZ     ..NCTS  ;IF NOT, CONTINUE
        BIT     CTS,A   ;ELSE, CHECK CLEAR TO SEND STATUS
        RZ              ;IF CLEAR TO SEND FALSE, DONE
..NCTS: LDA     S1OCHR  ;GET SERIAL 1 OUTPUT CHARACTER
        OUT     SIOBDR  ;OUTPUT CHARACTER
        LXI     H,S1OPOL  ;GET SERIAL 1 OUT POLL ROUTINE
        CALL    UNLINK# ;UNLINK POLL ROUTINE
        LXI     H,S1OSPH  ;GET SERIAL 1 OUT SEMAPHORE
        JMP     SIGNAL# ;SIGNAL PROCESS AS READY
;
SIOISR: SSPD    INTSP#  ;SAVE STACK POINTER
        LXI     SP,INTSTK#  ;SET UP AUX STACK POINTER
        PUSH    PSW     ;SAVE REGISTERS
        PUSH    B
        PUSH    D
        PUSH    H
        CALL    ..SOI   ;CHECK FOR SERIAL 0 INPUT
        CALL    ..S1I   ;CHECK FOR SERIAL 1 INPUT
        POP     H       ;RESTORE REGISTERS
        POP     D
        POP     B
        POP     PSW
        LSPD    INTSP#  ;RESTORE STACK POINTER
        EI              ;ENABLE INTERRUPTS
        RETI            ;DONE
..SOI:  IN      SIOACR  ;GET SIO PORT A STATUS
        BIT     RDA,A   ;CHARACTER AVAILABLE
        RZ              ;IF NOT, DONE
        IN      SIOADR  ;GET SIO PORT A DATA CHARACTER
        LXI     H,S0BR  ;GET SERIAL 0 BAUD RATE CODE
        BIT     5,M     ;INHIBIT INPUT FLAG SET?
        RNZ             ;IF SO, DONE
        MOV     C,A     ;SERIAL 0 DATA CHARACTER TO C-REG
        BIT     7,M     ;SIGN BIT ON BAUD RATE CODE?
        JRZ     ..NADO  ;IF NOT, CONTINUE
        RES     7,C     ;ELSE, STRIP SIGN BIT ON CHARACTER
        CALL    SLVRES# ;CHECK FOR SLAVE RESET
        LDA     ATNCHR# ;GET ATTENTION CHARACTER
        CMP     C       ;CHARACTER=ATTENTION CHARACTER?
        JRNZ    ..NADO  ;IF NOT, CONTINUE
        LHLD    SOIPTR  ;ELSE, GET SERIAL 0 INPUT POINTER
        SHLD    SOOPTR  ;RESET SERIAL 0 OUTPUT POINTER
        LXI     H,0
        SHLD    SOICNT  ;SET SERIAL 0 INPUT COUNT=0
..NADO: LHLD    SOIBSZ  ;GET SERIAL 0 INPUT BUFFER SIZE
        LDED    SOICNT  ;GET SERIAL 0 INPUT COUNT
        INX     D       ;INCREMENT SERIAL 0 INPUT COUNT
        ORA     A       ;CLEAR CARRY FLAG
        DSBC    D       ;SERIAL 0 INPUT BUFFER FULL?
        RC              ;IF SO, DONE
        SDED    SOICNT  ;ELSE, UPDATE SERIAL 0 INPUT COUNT
```

```
        LHLD    SOIPTR  ;GET SERIAL 0 INPUT POINTER
        MOV     M,C     ;STORE INPUT CHARACTER IN BUFFER
        INX     H       ;INCREMENT INPUT POINTER
        XCHG            ;DE=INPUT POINTER/HL=BUFFER SIZE
        LHLD    SOIBSZ  ;GET SERIAL 0 INPUT BUFFER SIZE
        DCX     H       ;DECREMENT INPUT BUFFER SIZE
        LBCD    SOIBUF  ;GET SERIAL 0 INPUT BUFFER ADDRESS
        DAD     B       ;CALC LAST INPUT BUFFER ADDRESS
        DSBC    D       ;BUFFER WRAP-AROUND?
        JRNC    ..NWA0  ;IF NOT, CONTINUE
        MOV     E,C     ;GET SERIAL 0 INPUT BUFFER ADDRESS
        MOV     D,B
..NWA0: SDED    SOIPTR  ;UPDATE SERIAL 0 INPUT POINTER
        LXI     D,SOIWCT  ;GET SERIAL 0 INPUT WAIT COUNT
        LXI     H,SOISPH  ;GET SERIAL 0 INPUT SEMAPHORE
        CALL    ..SIGC  ;SIGNAL IF NECESSARY
        JMPR    ..SOI   ;CONTINUE
..S1I:  IN      SIOBCR  ;GET SIO PORT B STATUS
        BIT     RDA,A   ;CHARACTER AVAILABLE
        RZ              ;IF NOT, DONE
        IN      SIOBDR  ;GET SIO PORT B DATA CHARACTER
        LXI     H,S1BR  ;GET SERIAL 1 BAUD RATE CODE
        BIT     5,M     ;INHIBIT INPUT FLAG SET?
        RNZ             ;IF SO, DONE
        MOV     C,A     ;SERIAL 1 DATA CHARACTER TO C-REG
        BIT     7,M     ;ATTENTION DETECTION FLAG SET?
        JRZ     ..NAD1  ;IF NOT, CONTINUE
        RES     7,C     ;ELSE, STRIP SIGN BIT ON CHARACTER
        CALL    SLVRES# ;CHECK FOR SLAVE RESET
        LDA     ATNCHR# ;GET ATTENTION CHARACTER
        CMP     C       ;CHARACTER=ATTENTION CHARACTER?
        JRNZ    ..NAD1  ;IF NOT, CONTINUE
        LHLD    S1IPTR  ;ELSE, GET SERIAL 1 INPUT POINTER
        SHLD    S1OPTR  ;RESET SERIAL 1 OUTPUT POINTER
        LXI     H,0
        SHLD    S1ICNT  ;SET SERIAL 1 INPUT COUNT=1
..NAD1: LHLD    S1IBSZ  ;GET SERIAL 1 INPUT BUFFER SIZE
        LDED    S1ICNT  ;GET SERIAL 1 INPUT COUNT
        INX     D       ;INCREMENT SERIAL 1 INPUT COUNT
        ORA     A       ;CLEAR CARRY FLAG
        DSBC    D       ;SERIAL 1 INPUT BUFFER FULL?
        RC              ;IF SO, DONE
        SDED    S1ICNT  ;ELSE, UPDATE SERIAL 1 INPUT COUNT
        LHLD    S1IPTR  ;GET SERIAL 1 INPUT POINTER
        MOV     M,C     ;STORE INPUT CHARACTER IN BUFFER
        INX     H       ;INCREMENT INPUT POINTER
        XCHG            ;DE=INPUT POINTER/HL=BUFFER SIZE
        LHLD    S1IBSZ  ;GET SERIAL 1 INPUT BUFFER SIZE
        DCX     H       ;DECREMENT INPUT BUFFER SIZE
        LBCD    S1IBUF  ;GET SERIAL 1 INPUT BUFFER ADDRESS
        DAD     B       ;CALC LAST INPUT BUFFER ADDRESS
        DSBC    D       ;BUFFER WRAP-AROUND?
        JRNC    ..NWA1  ;IF NOT, CONTINUE
        MOV     E,C     ;GET SERIAL 1 INPUT BUFFER ADDRESS
        MOV     D,B
..NWA1: SDED    S1IPTR  ;UPDATE SERIAL 1 INPUT POINTER
        LXI     D,S1IWCT  ;GET SERIAL 1 INPUT WAIT COUNT
        LXI     H,S1ISPH  ;GET SERIAL 1 INPUT SEMAPHORE
        CALL    ..SIGC  ;SIGNAL IF NECESSARY
        JMPR    ..S1I   ;CONTINUE
..SIGC: LDAX    D       ;GET SERIAL INPUT WAIT COUNT
        ORA     A       ;SERIAL INPUT WAIT COUNT=0?
        RZ              ;IF SO, DONE
        DCR     A       ;DECREMENT SERIAL INPUT WAIT COUNT
        STAX    D       ;UPDATE SERIAL INPUT WAIT COUNT
        JMP     SIGNAL# ;SIGNAL PROCESS AS READY
```

```
;
SERSBR: MOV     A,B     ;GET CHANNEL NUMBER
        LXI     H,SOBR  ;GET SERIAL 0 BAUD RATE CODE
        ORA     A       ;CHANNEL NUMBER=0?
        JRZ     ..COM1  ;IF SO, CONTINUE
        LXI     H,S1BR  ;ELSE, GET SERIAL 1 BAUD RATE CODE
..COM1: MOV     M,C     ;SAVE BAUD RATE CODE
        CALL    GETBTV  ;GET BAUD RATE TIMER VALUE
        MOV     A,B     ;GET CHANNEL NUMBER
        ORA     A       ;CHANNEL NUMBER=0?
        MVI     A,T0CMD ;GET TIMER 0 COMMAND
        MVI     C,TIM0  ;GET TIMER 0 DATA REGISTER
        JRZ     ..COM2  ;IF CHANNEL NUMBER=0, CONTINUE
        MVI     A,T1CMD ;ELSE, GET TIMER 1 COMMAND
        MVI     C,TIM1  ;GET TIMER 1 DATA REGISTER
..COM2: OUT     TIMCTL  ;SELECT TIMER
        OUTP    E       ;OUTPUT LSB OF TIMER VALUE
        OUTP    D       ;OUTPUT MSB OF TIMER VALUE
        RET             ;DONE
;
GETBTV::MOV     A,C     ;GET REQUESTED BAUD RATE CODE
        ANI     0FH     ;EXTRACT RELEVANT BITS
        ADD     A       ;X2
        MOV     E,A     ;TO E-REG
        MVI     D,0     ;MAKE IT DOUBLE LENGTH
        LXI     H,BRTBL ;GET BAUD RATE TABLE
        DAD     D       ;INDEX INTO TABLE
        MOV     E,M     ;GET TIMER VALUE
        INX     H
        MOV     D,M
        RET             ;DONE
;
BRTBL:  .WORD   3072    ;50 BAUD TIMER VALUE
        .WORD   2048    ;75 BAUD TIMER VALUE
        .WORD   1396    ;110 BAUD TIMER VALUE
        .WORD   1142    ;134.5 BAUD TIMER VALUE
        .WORD   1024    ;150 BAUD TIMER VALUE
        .WORD   512     ;300 BAUD TIMER VALUE
        .WORD   256     ;600 BAUD TIMER VALUE
        .WORD   128     ;1200 BAUD TIMER VALUE
        .WORD   85      ;1800 BAUD TIMER VALUE
        .WORD   77      ;2000 BAUD TIMER VALUE
        .WORD   64      ;2400 BAUD TIMER VALUE
        .WORD   43      ;3600 BAUD TIMER VALUE
        .WORD   32      ;4800 BAUD TIMER VALUE
        .WORD   21      ;7200 BAUD TIMER VALUE
        .WORD   16      ;9600 BAUD TIMER VALUE
        .WORD   8       ;19200 BAUD TIMER VALUE
;
SERRBR: LXI     H,SOBR  ;GET SERIAL 0 BAUD RATE
        MOV     A,B     ;GET CHANNEL NUMBER
        ORA     A       ;CHANNEL NUMBER=0?
        JRZ     ..COM   ;IF SO, CONTINUE
        LXI     H,S1BR  ;ELSE, GET SERIAL 1 BAUD RATE
..COM:  MOV     A,M     ;GET CURRENT BAUD RATE CODE
        RET             ;DONE
;
SERSMC: MVI     A,0EAH  ;GET WRITE REGISTER 5 CONTROL WORD
        ANI     #82H    ;STRIP RTS/CTS CONTROL BITS
        BIT     7,C     ;RTS REQUESTED?
        JRZ     ..NRTS
        SET     1,A     ;IF SO, SET RTS BIT
..NRTS: BIT     6,C     ;DTR REQUESTED?
        JRZ     ..NDTR
        SET     7,A     ;IF SO, SET DTR BIT
..NDTR: MOV     D,A     ;REQUESTED MODEM CONTROLS TO D-REG
```

```
        MOV     A,B     ;GET CHANNEL NUMBER
        ORA     A       ;CHANNEL NUMBER=0?
        JRZ     ..COM   ;IF SO, CONTINUE
        MVI     C,SIOBCR  ;GET SIO PORT B CONTROL REGISTER
..COM:  MVI     A,5     ;SELECT WRITE REGISTER 5
        OUTP    A
        OUTP    D       ;OUTPUT CONTROL WORD
        RET             ;DONE
;
SERRMC: MVI     C,SIOACR  ;GET SIO PORT A CONTROL REGISTER
        MOV     A,B     ;GET CHANNEL NUMBER
        ORA     A       ;CHANNEL NUMBER=0?
        JRZ     ..COM   ;IF SO, CONTINUE
        MVI     C,SIOBCR  ;GET SIO PORT B CONTROL REGISTER
..COM:  MVI     A,10H   ;GET RESET EXTERNAL STATUS COMMAND
        OUTP    A       ;RESET EXTERNAL STATUS
        INP     D       ;GET SIO MODEM STATUS
        XRA     A       ;CLEAR RETURN VECTOR
        BIT     CTS,D   ;CTS SET?
        JRZ     ..NCTS  ;IF NOT, CONTINUE
        SET     7,A     ;ELSE, SET CTS BIT
..NCTS: BIT     DCD,D   ;DCD SET?
        RZ              ;IF NOT, DONE
        SET     5,A     ;ELSE, SET DCD BIT
        RET             ;DONE
;
.XSYM
.END
```