

```

;
; LIST ON
; Configuration program for SD SYSTEMS I/O-8 BOARD
; This program initializes one or more of the Z8030 SCC's & allows the
; configuration of the MM58167 Clock chip.
;
; PORT ASSIGNMENTS
;
BCTL EQU 010H ; CHANNEL B CONTROL
ACTL EQU 011H ; CHANNEL A CONTROL
BDA EQU 012H ; CHANNEL B DATA
ADA EQU 013H ; CHANNEL A DATA
SCCSEL EQU 014H ; SCC SELECT
RTC_SEL EQU 015H ; RTC Reg Select (R/W), and SSC Status (Read only)
RTC_DATA EQU 016H ; RTC Data
SDS_STAT EQU 017H ; SD IO8 Board status port

; BDOS EQUATES (VERSION 2)
;
RDCON EQU 1 ; CP/M Read character
WRCON EQU 2 ; CP/M Write character
PRINT EQU 9 ; CP/M Print string
BDOS EQU 5
ESC EQU 1BH

ORG 100H
START:
LD SP, STACK
;
LD DE, SIGNON ; Signon/main menu
LD C, PRINT
CALL BDOS
;
LD C, RDCON ; get a char (menu 0 0r 1)
CALL BDOS
;
CP 30H
JP Z, BEGIN_SCC
CP 31H
JP Z, BEGIN_TIME
CP ESC
JP Z, DONE
;
BAD_CHAR:
LD DE, ABORT_MSG
LD C, PRINT
CALL BDOS
JP START
;
; ----- Configure the SCC's -----
; -----
;
BEGIN_SCC:
LD DE, SEL_SSC_MSG ; Which SCC, 0,1,2,3

```

```

LD    C, PRINT
CALL  BDOS
;
LD    C, RDCON      ;get a char
CALL  BDOS
;
CP    ESC
JP    Z, START
CP    '1'          ;check 1-4
JP    C, BAD_CHAR
CP    '4'+1
JP    NC, BAD_CHAR
;
SELECT:  SUB    A, 31H      ;Set range 0,1,2,3
        OUT    (SCCSEL), A ;Select SSC #

LD    DE, CRLF_MSG ;Finish Line
LD    C, PRINT
CALL  BDOS

CALL  INIT_SSC      ;Initilize the currently selected Zilog 8530's
                        ;On SD Systems IO8 Board

LD    DE, INIT_DONE      ;Say done
LD    C, PRINT
CALL  BDOS
;
LD    DE, SEL_BAUD_MSG  ;Which BAUD Rate
LD    C, PRINT
CALL  BDOS
;
LD    C, RDCON      ;Get a char
CALL  BDOS
PUSH  AF
LD    DE, CRLF_MSG ;Finish Line
LD    C, PRINT
CALL  BDOS
POP   AF
;
CP    30H          ;300baud
JP    NZ, NEXT_B
LD    B, 0FEH
LD    DE, DONE_B0_MSG ;Done BAUD Rate
JP    SET_BAUD
NEXT_B:  CP    31H          ;1200baud
        JP    NZ, NEXT_C
        LD    B, 03EH
        LD    DE, DONE_B1_MSG ;Done BAUD Rate
        JP    SET_BAUD
NEXT_C:  CP    32H          ;2400baud
        JP    NZ, NEXT_D
        LD    B, 01EH
        LD    DE, DONE_B2_MSG ;Done BAUD Rate
        JP    SET_BAUD

```

```

NEXT_D:    CP    33H            ;4800baud
           JP    NZ,NEXT_E
           LD    B,0EH
           LD    DE,DONE_B3_MSG ;Done BAUD Rate
           JP    SET_BAUD
NEXT_E:    CP    34H            ;9600baud
           JP    NZ,NEXT_F
           LD    B,06H
           LD    DE,DONE_B4_MSG ;Done BAUD Rate
           JP    SET_BAUD
NEXT_F:    CP    35H            ;19200baud
           JP    NZ,SKIP1
           LD    B,02H
           LD    DE,DONE_B5_MSG ;Done BAUD Rate
           JP    SET_BAUD
;
SET_BAUD:  ;If needed later we could distinguish Channels A/B
           LD    A,0CH          ;Point to WR12 (Low Byte)
           OUT   (ACTL),A
           LD    A,B            ;get selected new baud rate
           OUT   (ACTL),A
;
           LD    A,0CH          ;Point to WR12 (Low Byte)
           OUT   (BCTL),A
           LD    A,B            ;get selected new baud rate
           OUT   (BCTL),A
;
           LD    C,PRINT        ;Print new Baud rate (DE)
           CALL  BDOS
           JP    START
;
SKIP1:    LD    DE,SKIP1_MSG    ;Skip BAUD Rate
           LD    C,PRINT
           CALL  BDOS
           JP    DONE
;
DONE:    LD    C,0              ;Back to CP/M
           CALL  BDOS
;
;
;----- Configure the MM58167 Clock chip -----
;
BEGIN_TIME:
           LD    DE,TIME_MENU    ;Time menu
           LD    C,PRINT
           CALL  BDOS
;
           LD    C,RDCON          ;Get a char
           CALL  BDOS
           PUSH  AF
           LD    DE,CRLF_MSG ;Finish Line
           LD    C,PRINT
           CALL  BDOS
           POP   AF

```

```

;
CP    ESC
JP    Z,START
CP    30H          ;Get Time
JP    Z,GTIME
CP    31H
JP    Z,STIME      ;Set Time
CP    32H
JP    Z,ZERO_TIME
;
LD    DE,ABORT_MSG
LD    C,PRINT
CALL  BDOS
JP    DONE
;
GTIME:                                ;Get Time and Print at current cursor position
LD    DE,DATE_MSG
LD    C,PRINT
CALL  BDOS

LD    A,07H
OUT   (RTC_SEL),A ;Point to MONTH
IN    A,(RTC_DATA)
CALL  PRINT_REG
LD    E,'/'
LD    C,WRCON
CALL  BDOS
;
LD    A,06H
OUT   (RTC_SEL),A ;Point to DAY
IN    A,(RTC_DATA)
CALL  PRINT_REG
LD    DE,YEAR_2000
LD    C,PRINT
CALL  BDOS
;
LD    A,09H
OUT   (RTC_SEL),A ;Point to RAM store of YEAR (0-99)
IN    A,(RTC_DATA)      ;I use the .01 Sec store area
CALL  PRINT_REG
LD    DE,TIME_MSG
LD    C,PRINT
CALL  BDOS
;
LD    A,04H
OUT   (RTC_SEL),A ;Point to HOURS
IN    A,(RTC_DATA)
CALL  PRINT_REG
LD    E,':'
LD    C,WRCON
CALL  BDOS
;
LD    A,03H
OUT   (RTC_SEL),A ;Point to MINS

```

```

IN    A,(RTC_DATA)
CALL  PRINT_REG
LD    E,':'
LD    C,WRCON
CALL  BDOS
;
LD    A,02H
OUT   (RTC_SEL),A;Point to SEC
IN    A,(RTC_DATA)
CALL  PRINT_REG
LD    DE,CRLF_MSG
LD    C,PRINT
CALL  BDOS
JP    BEGIN_TIME
;
STIME:                                ;Set the time in the chip
;    RESET ALL TIMERS
;
LD    A,12H        ;point to counter reg 12
OUT   (RTC_SEL),A
LD    A,0FFH
OUT   (RTC_DATA),A
;
;    RESET ALL RAM LOCATIONS
;
LD    A,13H        ;Point RAM register
OUT   (RTC_SEL),A
LD    A,0FFH
OUT   (RTC_DATA),A
;
LD    DE,GET_MONTH
LD    C,PRINT
CALL  BDOS
CALL  GET_HEX
JP    C,BEGIN_TIME
PUSH  AF
LD    A,07H        ;Months register
OUT   (RTC_SEL),A
POP   AF
OUT   (RTC_DATA),A
;
LD    DE,GET_DAY
LD    C,PRINT
CALL  BDOS
CALL  GET_HEX
JP    C,BEGIN_TIME
PUSH  AF
LD    A,06H        ;Days register
OUT   (RTC_SEL),A
POP   AF
OUT   (RTC_DATA),A
;
LD    DE,GET_YEAR
LD    C,PRINT

```

```

CALL  BDOS
CALL  GET_HEX
JP    C,BEGIN_TIME
PUSH  AF
LD    A,09H      ;Years (Post 2000) register
OUT   (RTC_SEL),A
POP   AF
OUT   (RTC_DATA),A      ;Store in chip RAM
;
LD    DE,GET_HOUR
LD    C,PRINT
CALL  BDOS
CALL  GET_HEX
JP    C,BEGIN_TIME
PUSH  AF
LD    A,04H      ;Hours register
OUT   (RTC_SEL),A
POP   AF
OUT   (RTC_DATA),A
;
LD    DE,GET_MINS
LD    C,PRINT
CALL  BDOS
CALL  GET_HEX
JP    C,BEGIN_TIME
PUSH  AF
LD    A,03H      ;Mins register
OUT   (RTC_SEL),A
POP   AF
OUT   (RTC_DATA),A      ;all done
GO_CMD:                                ;This command simply zeros seconds and lower
fractions
LD    A,15H      ;GO register
OUT   (RTC_SEL),A
LD    A,0H
OUT   (RTC_DATA),A
;
LD    DE,CRLF_MSG
LD    C,PRINT
CALL  BDOS
JP    GTIME      ;Display what we have
;
ZERO_TIME:
LD    DE,ZERO_MSG
LD    C,PRINT
CALL  BDOS
JP    GO_CMD
;
----- SUPPORT ROUTINES -----
;
PRINT_REG:                                ;Display Clock register as High/Low byte on CRT
PUSH  AF      ;High & low minutes
RRA

```

```

RRA
RRA
RRA
AND    0FH
ADD    A,30H
LD     E,A
LD     C,WRCON           ;Write high byte mins
CALL   BDOS
POP    AF
AND    0FH
ADD    A,30H
LD     E,A
LD     C,WRCON           ;Write low byte mins
CALL   BDOS
RET

;
; SD Systems IO8 Serial SCC board initialization (assumes current SCC is already
; selected)
;
INIT_SSC:
    LD     A,ACTL           ;Program Channel A
    LD     C,A
    LD     B,0EH           ;Byte count for OTIR below
    LD     HL,SCCINIT
    OTIR

;

    LD     A,BCTL           ;Program Channel B
    LD     C,A
    LD     B,0EH           ;Byte count for OTIR below
    LD     HL,SCCINIT
    OTIR
    RET

;
;
;Return with 2 digits in [A]. If abort, Carry flag set + ESC in [A]
GET_HEX:
    PUSH  BC
    LD     C,RDCON
    CALL   BDOS           ;Get a character from keyboard & ECHO
    CP     A,ESC
    JR     Z,HEX_ABORT
    CP     '/'           ;check 0-9, A-F
    JR     C,HEX_ABORT
    CP     '9'+1
    JR     NC,HEX_ABORT
    CALL   ASBIN           ;Convert to binary
    SLA   A
    SLA   A
    SLA   A
    SLA   A           ;Shift to high nibble
    LD     B,A           ;Store it
    PUSH  BC           ;Because CP/M destroys BC
    LD     C,RDCON
    CALL   BDOS           ;Get a character from keyboard & ECHO

```

```

    POP    BC
    CP     A,ESC
    JR     Z,HEX_ABORT
    CP     '/'      ;check 0-9, A-F
    JR     C,HEX_ABORT
    CP     '9'+1
    JR     NC,HEX_ABORT
    CALL   ASBIN    ;Convert to binary
    OR     A,B      ;add in the first digit
    OR     A,A      ;To return NC
    POP    BC
    RET

;
HEX_ABORT:
    SCF                    ;Set Carry flag
    LD     A,ESC
    POP    BC
    RET

;
; ASCII TO BINARY CONVERSION ROUTINE
ASBIN:   SUB    30H
        CP     0AH
        RET    M
        SUB    07H
        RET

;
; ALL SSC's are set for 4800 BAUD
SCCINIT:
    DB     04H    ;Point to WR4
    DB     44H    ;X16 clock,1Stop,NP
;
    DB     03H    ;Point to WR3
    DB     0C1H   ;Enable, Recieve 8 bits
;
    DB     05H    ;Point to WR5
    DB     0EAH   ;Enable, Transmit 8 bits
;
;                          ;Set RTS,DTR, Enable
;
    DB     0BH    ;Point to WR11
    DB     56H    ;Recieve/transmit clock = BRG
;
    DB     0CH    ;Point to WR12
;
    DB     1EH    ;Low Byte 2400 Baud
    DB     0EH    ;Low Byte 4800 Baud <<<<<<<<<<<<<<<<<<<<<<<<<<<<
;
    DB     06H    ;Low byte 9600 Baud
;
    DB     0DH    ;Point to WR13
    DB     00H    ;High byte for Baud
;
    DB     0EH    ;Point to WR14
    DB     01H    ;Use 2.4576 MHz clock, enable BRG
;
;
SIGNON:   DB     13,10,10

```



```

DB      'Program To Configure SD Systems IO/8 Serial Board.',13,10
DB      'MAIN MENU',13,10
DB      '0 = Initilize one of the four SCCs (Z8530s)',13,10
DB      '1 = Configure Clock/Time chip',13,10
DB      'ESC = Return to CP/M',13,10,10
DB      'Menu choice ---->$'

SEL_SSC_MSG:  DB      13,10,10
DB      'SCC MENU',13,10
DB      'Select the SCC# you wish to modify [1,2,3,4, (ESC to Skip)]---
>$'
NAME_SEL:    DB      13,10,10
DB      'You selected SCC#$',
CRLF_MSG:    DB      13,10,10,'$'
INIT_DONE:   DB      'The selected SCC is now initilized to 8 Bits, RTS
on.',13,10,10,'$'
SEL_BAUD_MSG: DB      'Select a Baud rate for this SCC (Both A & B
Channels)',13,10
DB      '0 = 300 baud',13,10
DB      '1 = 1200 baud',13,10
DB      '2 = 2400 baud',13,10
DB      '3 = 4800 baud',13,10
DB      '4 = 9600 baud',13,10
DB      '5 = 19200 baud',13,10,10
DB      'Baud Rate number selected --->$'
DONE_B0_MSG:  DB      'Baud rate RESET to 300 baud (Both A & B
Channels)',13,10,'$'
DONE_B1_MSG:  DB      'Baud rate RESET to 1200 baud (Both A & B
Channels)',13,10,'$'
DONE_B2_MSG:  DB      'Baud rate RESET to 2400 baud (Both A & B
Channels)',13,10,'$'
DONE_B3_MSG:  DB      'Baud rate RESET to 4800 baud (Both A & B
Channels)',13,10,'$'
DONE_B4_MSG:  DB      'Baud rate RESET to 9600 baud (Both A & B
Channels)',13,10,'$'
DONE_B5_MSG:  DB      'Baud rate RESET to 19200 baud (Both A & B
Channels)',13,10,'$'
SKIP1_MSG:    DB      13,10
DB      'Skipped selecting a new BAUD rate for the board current
SCC.',13,10,'$';
ABORT_MSG:    DB      13,10
DB      'Invalid menu item',13,10,'$'
TIME_MENU:    DB      13,10,10
DB      'Configure MM58167A Clock Chip.',13,10
DB      'CLOCK MENU',13,10
DB      '0 = Get Time',13,10
DB      '1 = Set Clock/Time',13,10
DB      '2 = Zero seconds',13,10
DB      'ESC to Skip',13,10,10
DB      'Menu choice ---->$'
DATE_MSG:     DB      'Date: $'
TIME_MSG:     DB      ' Time: $'
YEAR_2000:    DB      '/20$'
GET_MONTH:    DB      13,10,10

```

```
GET_DAY:    DB      'Enter Month, two characters, (01-12)--->$'
            DB      13,10,10
            DB      'Enter Day, two characters, (01-31)--->$'
GET_YEAR:   DB      13,10,10
            DB      'Enter Year after 2000, two characters, (01-99)--->$'
GET_HOUR:   DB      13,10,10
            DB      'Enter Hour, two characters, (00-23)--->$'
GET_MINS:   DB      13,10,10
            DB      'Enter Minutes, two characters, (00-59)--->$'
ZERO_MSG    DB      13,10
            DB      'Reset clock seconds.',13,10,'$'
;
            DS      40H
STACK:      DB      0H
;
; END
```