

```

;
;Configuration program for the S-100 I/O board
;This program initilizes the Z8030 SCC's,the V-Stamp Chip & DLP-USB Controller chip
;
;   Author      John Monahan      (S100Computers.com)      6/20/2020
;
;
; PORT ASSIGNMENT FOR DLP-USB Controller chip
;
USB_DATA EQU 0ACH ;<--- Adjust as necessary, also update Signon MSG below
USB_STATUS EQU 0AAH ; Status port for USB port (Port C of 8255, bits 6,7)
;
USB_RXE EQU 80H ; If Bit 7 = 0, data available to recieve by S-100 Computer
USB_TXE EQU 40H ; If Bit 6 = 0 data CAN be written for transmission to PC
;
; PORT ASSIGNMENTS OF THE ZILOG SCC CHIP
;
BCTL EQU 0A0H ; CHANNEL B CONTROL ;<--- Adjust as necessary,
ACTL EQU 0A1H ; CHANNEL A CONTROL ; also update Signon MSG below
BDTA EQU 0A2H ; CHANNEL B DATA
ADTA EQU 0A3H ; CHANNEL A DATA

PortA_8255 EQU 0A8H ;A port of 8255 ;<--- Adjust as necessary
PortB_8255 EQU 0A9H ;B port of 8255
PortC_8255 EQU 0AAH ;C Port of 8255
PortCtrl_8255 EQU 0ABH ;8255 configuration port

AinBout8255cfg EQU 10011000b ;Set 8255 ports:- A input, B output,
;C(bits 0-3) output, (bits 4-7)input
;AoutBin8255cfg EQU 10001010b ;Set 8255 ports:- A output, B input,
;C(bits 0-3) output, (bits 4-7)input

;<<<< SD Systems Video board will be used for Consol I/O <----- Adjust if different
;
CONSOL_IN EQU 01H
CONSOL_OUT EQU 01H
CONSOL_STATEQU 00H

CHAR_IN_OK EQU 02H ;02H if there is a character at consol keyboard
CHAR_OUT_OK EQU 04H ;Mask for CRT output

; BDOS EQUATES (VERSION 2)
;
RDCONEQU 1 ;CP/M Read character
WRCON EQU 2 ;CP/M Write character
CONSTAT EQU 11 ;CP/M Get consol status
PRINTEQU 9 ;CP/M Print string
BDOS EQU 5

ESC EQU 1BH
CR EQU 0DH
LF EQU 0AH

ORG 100H

START: LD SP,STACK
LD DE,SIGNON ;Signon/main menu
LD C,PRINT
CALL BDOS

IDEinit: ;Initilze the 8255.
LD A,AinBout8255cfg ;A input, B output, C(bits 0-3) output, (bits 4-7)input
OUT (PortCtrl_8255),A ;Config 8255 chip, Mode 0

```

```

;Configure the SCC's
BEGINSCC:
  LD  C,RDCON          ;Get a char
  CALL BDOS
  PUSH AF
  LD  DE,CRLFMSG      ;Finish Line
  LD  C,PRINT
  CALL BDOS
  POP  AF
;
  CP  30H
  JP  Z,INITSCC      ;Initilize the SCC
  CP  31H
  JP  Z,SET_BAUD    ;Set its baude rate (default is 4800)
  CP  32H
  JP  Z,SCC_LOOP_TEST
  CP  33H
  JP  Z,USB_LOOP_TEST
  CP  34H
  JP  Z,SPEECH_TEST
  CP  ESC
  JP  Z,DONE
  JP  BADCHAR
;
SET_BAUD:
  LD  DE,SELBAUDMSG   ;Which BAUD Rate
  LD  C,PRINT
  CALL BDOS
;
  LD  C,RDCON          ;Get a char
  CALL BDOS
  PUSH AF
  LD  DE,CRLFMSG      ;Finish Line
  LD  C,PRINT
  CALL BDOS
  POP  AF
;
  CP  30H              ;600 baud
  JP  NZ,NEXTB
  LD  B,0FEH
  LD  DE,DONEB0MSG     ;Done BAUD Rate
  JP  SETBAUD
NEXTB:  CP  31H          ;1200 baud
  JP  NZ,NEXTC
  LD  B,07FH
  LD  DE,DONEB1MSG     ;Done BAUD Rate
  JP  SETBAUD
NEXTC:  CP  32H          ;2400 baud
  JP  NZ,NEXTD
  LD  B,040H
  LD  DE,DONEB2MSG     ;Done BAUD Rate
  JP  SETBAUD
NEXTD:  CP  33H          ;4800 baud
  JP  NZ,NEXTE
  LD  B,01EH
  LD  DE,DONEB3MSG     ;Done BAUD Rate
  JP  SETBAUD
NEXTE:  CP  34H          ;9600 baud
  JP  NZ,NEXTF
  LD  B,0EH
  LD  DE,DONEB4MSG     ;Done BAUD Rate
  JP  SETBAUD
NEXTF:  CP  35H          ;19200 baud

```

```

JP      NZ,NEXTG
LD      B,6
LD      DE,DONEB5MSG      ;Done BAUD Rate
JP      SETBAUD
NEXTG:  CP      36H      ;38400 baud
JP      NZ,NEXTH
LD      B,2
LD      DE,DONEB6MSG      ;Done BAUD Rate
JP      SETBAUD
NEXTH:  CP      37H      ;76800 baud
JP      NZ,SKIP1
LD      B,0      ;<<<<<< 0H
LD      DE,DONEB7MSG      ;Done BAUD Rate
JP      SETBAUD
;
SETBAUD:      ;Note, later we could distinguish Channels A/B here.
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,SKIP1MSG ;Skip BAUD Rate
LD      C,PRINT
CALL    BDOS
JP      START
;
DONE:  LD      C,0      ;Back to CP/M
CALL    BDOS
;
BADCHAR:LD DE,ABORTMSG
LD      C,PRINT
CALL    BDOS
JP      START
;
;
;
SCC_LOOP_TEST:      ;Test if we can send & recieve characters from SCC serial
Port A
next:  IN      A,ACTL      ;Are we ready to recieve a character from Zilog SCC chip
AND    01H
JP      Z,CONT      ;Zero if nothing
IN      A,ADTA
OUT     CONSOL_OUT,A      ;Send to consol port (Skip status check, consol can keep
up!)

CONT:  IN      A,CONSOL_STAT      ;Anything at consol keyboard
AND    A,CHAR_IN_OK
JP      Z,next
IN      A,CONSOL_IN ;Get keyboard character
CP      A,ESC      ;If ESC abort loop
JP      Z,START
LD      C,A

next1:  IN      A,ACTL      ;Are we ready to send a character to SCC
AND    04H

```

```

    JP    NZ,SENDC
    LD    A, '.'
    OUT   CONSOL_OUT,A      ;Send a "." to CRT if SCC is busy/not working
    JP    next1
SENDC:   LD    A,C
    OUT   (ADTA),A        ;Send it to SCC port
    JP    next
;
;
USB_LOOP_TEST:      ;Test if we can send & recieve characters from USB Port
USB_next:
    IN    A,(USB_STATUS)  ;Is there a character to recieve from the PC via the USB
    Port
    AND   USB_RXE
    JP    NZ,USB_CONT ;Zero if nothing there
    IN    A,(USB_DATA)
    OUT   CONSOL_OUT,A    ;Send to consol port (Skip status check, consol can keep
    up!)
    JP    USB_next

USB_CONT:
    IN    A,CONSOL_STAT   ;Anything to send from S-100 consol, check consol status
    AND   A,CHAR_IN_OK
    JP    Z,USB_next ;Nothing at consol, back to start of loop
    IN    A,CONSOL_IN ;Get keyboard character
    CP    A,ESC
    JP    Z,START        ;Abort if ESC key
    LD    C,A            ;Store it in [C]
USB_next1:
    IN    A,(USB_STATUS)  ;Are we ready to send a character to PC via USB port
    AND   USB_TXE        ;Is chip ready
    JP    Z,USB_SENDC
    LD    A, '.'
    OUT   CONSOL_OUT,A    ;Send a "." to CRT while we wait
    JP    USB_next1
USB_SENDC:
    LD    A,C
    OUT   (USB_DATA),A    ;Send it to tp PC via USB port
    JP    USB_next
;
;
SPEECH_TEST:
    LD    A,(SSC_Init_Flag)
    OR    A,A            ;Was SSC initilzation done
;;    JP    Z,SP_TEST1
;;    LD    DE,INITNOTDONE ;Say SCC initilzation NOT done
;;    LD    C,PRINT
;;    CALL  BDOS
;;    JP    START

SP_TEST1:
    IN    A,CONSOL_STAT   ;Anything at consol keyboard
    AND   A,CHAR_IN_OK
    JP    Z,SP_TEST1
    IN    A,CONSOL_IN ;Get keyboard character
    CP    A,ESC          ;If ESC abort loop
    JP    Z,START
    LD    C,A

nextS:   IN    A,BCTL      ;Are we ready to send a character to SCC
    AND   04H
    JP    NZ,SENDS
    LD    A, '.'
    OUT   CONSOL_OUT,A    ;Send a "." to CRT if SCC is busy/not working

```

```

    JP     nextS
SENDS:  LD     A,C
        OUT    BDTA,A           ;Send it to SCC port
        OUT    CONSOL_OUT,A     ;Send to CRT also
        JP     SP_TEST1

;Zilog Serial SCC board initialization
;
INITSCC:
    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

;
    LD     DE,INITDONE ;Say SCC initialization done
    LD     C,PRINT
    CALL  BDOS
    XOR   A,A           ;Set flag for initialization done
    LD    (SSC_Init_Flag),A
    JP    START

;
;
;----- Support Routines -----
;
;CONVERT HEX TO ASCII
CONV: AND    0FH
      ADD    A,90H
      DAA
      ADC    A,40H
      DAA
      LD    C,A
      RET

;
;
;Return with 2 digits in [A]. If abort, Carry flag set + ESC in [A]
;
GETHEX:  PUSH  BC
        LD    C,RDCON
        CALL  BDOS           ;Get a character from keyboard & ECHO
        CP    A,ESC
        JR    Z,HEXABORT
        CP    '/'           ;check 0-9, A-F
        JR    C,HEXABORT
        CP    '9'+1
        JR    NC,HEXABORT
        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,HEXABORT

```



```

        DB      '3 = DLP-USB Controller/Port Loop test (Port = A4H, 8255 Status =
AAH))',13,10
        DB      '4 = Send test speech string to V_Stamp chip via SCC chip (Port =
A0H,A2H)',13,10
        DB      'ESC = Return to CP/M',13,10,10
        DB      'Menu choice ---->$'

CRLFMSG:   DB      13,10,10,'$'
INITDONE:  DB      'The SCC is now initilized to 8 Bits, 1 Stop bit, NP, RTS on, 19,200
Baud.',13,10,10,'$'
INITNOTDONE: DB      'The SCC must first be initilized to 8 Bits, 1 Stop bit, NP, RTS
on, 19,200 Baud.',13,10,10,'$'

SELBAUDMSG:DB      'Select a Baud rate for this SCC (Both A & B Channels)',13,10
        DB      '0 = 600 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      '6 = 38400 baud',13,10,10
        DB      '7 = 76800 baud',13,10,10
        DB      'Baud Rate number selected --->$'
DONEB0MSG: DB      'Baud rate RESET to 600 baud (Both A & B Channels)',13,10,'$'
DONEB1MSG: DB      'Baud rate RESET to 1200 baud (Both A & B Channels)',13,10,'$'
DONEB2MSG: DB      'Baud rate RESET to 2400 baud (Both A & B Channels)',13,10,'$'
DONEB3MSG: DB      'Baud rate RESET to 4800 baud (Both A & B Channels)',13,10,'$'
DONEB4MSG: DB      'Baud rate RESET to 9600 baud (Both A & B Channels)',13,10,'$'
DONEB5MSG: DB      'Baud rate RESET to 19200 baud (Both A & B Channels)',13,10,'$'
DONEB6MSG: DB      'Baud rate RESET to 38400 baud (Both A & B Channels)',13,10,'$'
DONEB7MSG: DB      'Baud rate RESET to 76800 baud (Both A & B Channels)',13,10,'$'
SKIP1MSG:  DB      13,10
        DB      'Skipped selecting a new BAUD rate for the board current
        SCC.',13,10,'$';
ABORTMSG:  DB      13,10
        DB      'Invalid menu item',13,10,'$'
;
;
        DS      40H
STACK:     DB      0H
SSC_Init_Flag: DB      0FFH
;
; END

```