```
; Test program for HT6542 PC Keyboard controller on S100Computers/N8VEM "MS_DOS Support Board (Load with CPM).
;
;     V0.1              ;Original version 2/23/2014
;
;     John Monahan      S100Computers.com
;
; This is a simple test program to work with the MS-DOS Support Board. It is written so
; the only other hardware use is the Consol I/O port.
; Note the data is displayed in crude (bulk) form. A proper scancode to ASCII translation
; routine must be written for practical use.  See the IBM PC BIOS or SKEY.Z80 docs



;     PORT ASSIGNMENTS

KEY_DATA    EQU   60H         ;Port used to access keyboard & Mouse (also sometimes Controller itself)
KEY_CTRL    EQU   64H         ;Port to block 8259A interrupts

KEYSTAT           EQU   0H          ;Propeller Console IO S-100 board or SD SYSTEMS VIDIO BOARD FOR CONSOLE
KEYIN       EQU   01H         ;Console input port. Normally the Propeller Driven S-100 Console-IO Board
KEYOUT            EQU   01H         ;Console output port. Normally the Propeller Driven S-100 Console-IO Board

ESC         EQU   1BH
CR          EQU   0DH
LF          EQU   0AH
TAB         EQU   09H
BELL        EQU   07H

      ORG   100H
START:
      LD    SP,STACK

      LD    HL,SIGNON         ; Signon
      CALL  PRINT_STRING

      LD    C,0AAH            ;Test PS/2 Controller
      CALL  KEY_OUT
CHK1: CALL  KEY_IN_STATUS     ;wait for feedback
      JR    Z,CHK1
      IN    A,(KEY_DATA)
      CP    A,55H             ;If not 55H then error
      JR    Z,DONE_INIT
      LD    HL,INIT_ERR       ;Say error
      CALL  PRINT_STRING
      HALT                    ;Just Halt!
```

```
DONE_INIT:
      LD    HL,INIT_OK         ;Say all OK
      CALL  PRINT_STRING

      LD    C,0AEH             ;Enable 1st PS/2 port
      CALL  KEY_OUT            ;Send it

LOOP: CALL  KEY_IN_STATUS      ;See if keyboard key available
      JR    Z,LOOP
      IN    A,(KEY_DATA)
      LD    C,A                ;Store in [C]
      LD    HL,SCAN_MSG
      CALL  PRINT_STRING       ;No registers changed

      CALL  A_HEXOUT           ;Display Hex value of typed character + two spaces

      CP    0F0H               ;Is it an UP key
      JR    NZ,DOWNKY          ;Must be a down key stroke
      LD    HL,UPKEY_MSG       ;Say Up Key
      CALL  PRINT_STRING
      CALL  ZCRLF
      JR    LOOP

DOWNKY:
      CP    58H                ;Is it CAPS Lock key
      JR    NZ,NOT_CAPSKEY
      LD    HL,CAPS_MSG        ;Say Caps lock key
      CALL  PRINT_STRING
      CALL  ZCRLF
      JR    LOOP

NOT_CAPSKEY:
      CP    12H                ;Is it a SHIFT key
      JR    Z,SHIFTKEY
      CP    59H                ;Is it the other SHIFT key
      JR    NZ,NOT_SHIFTKEY
SHIFTKEY:
      LD    HL,SHIFT_MSG       ;Say Shift key
      CALL  PRINT_STRING
      CALL  ZCRLF
      JR    LOOP

NOT_SHIFTKEY:
      CP    14H                ;Is it the CTRL key
      JR    NZ,NOT_CTRLKEY
      LD    HL,CTRL_MSG       ;Say CTRL key
      CALL  PRINT_STRING
      CALL  ZCRLF
      JR    LOOP

NOT_CTRLKEY:
      CP    77H               ;Is it the NUM LOCK key
      JR    NZ,NOT_NUMKEY
      LD    HL,NUM_MSG        ;Say Number key
      CALL  PRINT_STRING
      CALL  ZCRLF
      JR    LOOP

NOT_NUMKEY:
      PUSH  BC                ;Save Character
      LD    HL,IBM1_MSG       ;Say Table 1 lookup
      CALL  PRINT_STRING
      LD    HL,IBM1TBL        ;Point to lookup table for upper case
      CALL  SHOW_CHAR

      POP   BC                ;Get back character
      LD    HL,IBM2_MSG       ;Say Table 2 lookup
      CALL  PRINT_STRING
      LD    HL,IBM2TBL        ;Point to lookup table for upper case
      CALL  SHOW_CHAR

      CALL  ZCRLF
      JR    LOOP

SHOW_CHAR:
      LD    D,0
      LD    E,C
      ADD   HL,DE             ;Add in offset
      LD    C,(HL)
      LD    A,C
      CP    A,ESC
      RET   Z                 ;ESC messes up the screen display
      CP    A,CR
      RET   Z                 ;CR messes up the screen display
      CP    A,LF
      RET   Z                 ;LF messes up the screen display
      CP    A,TAB
      RET   Z                 ;TAB messes up the screen display
      CALL  ZCO               ;Display on Screen
      RET


KEY_IN_STATUS:                ;Ret NZ if character is available
      IN    A,(KEY_CTRL)
      AND   A,1
      RET                     ;Ret NZ if character available

KEY_OUT:                      ;Send a byte (in [C]) to Control port
      IN    A,(KEY_CTRL)
      AND   A,2
      JR    NZ,KEY_OUT        ;Chip is not ready yet to recieve character
      LD    A,C
      OUT   (KEY_CTRL),A
      RET


;     A_HEXOUT                ;output the 2 hex digits in [A]
A_HEXOUT:                     ;No registers altered
      push  AF
      push  BC
      push  AF
      srl   a
      srl   a
      srl   a
      srl   a
      call  hexdigout
      pop   AF
      call  hexdigout         ;get upper nibble
      LD    C,' '
      call  ZCO               ;Space for easy reading
      call  ZCO
      pop   BC
      pop   AF
      ret

hexdigout:
      and   a,0fh             ;convert nibble to ascii
      add   a,90h
      daa
      adc   a,40h
      daa
      LD    c,a
      call  ZCO
      ret

; Main consol I/O routines
;
ZCO:  IN    A,(KEYSTAT)
      AND   04H
      JP    Z,ZCO
      LD    A,C
      OUT   (KEYOUT),A
      RET

ZCI:  IN    A,(KEYSTAT)
      AND   02H
      JP    Z,ZCI
      IN    A,(KEYIN)
      RET

;
; Send CR/LF to Consol
;
ZCRLF:      PUSH  AF
      PUSH  BC
      LD    C,CR
      CALL  ZCO
      LD    C,LF
      CALL  ZCO
      POP   BC
      POP   AF
      RET


PRINT_STRING:
      PUSH  AF
      push  BC
print1:     LD    a,(HL)      ;Point to start of string
      inc   HL                ;By using the CS over-ride we will always have
      cp    A,'$'             ;a valid pointer to messages at the end of this monitor
      JP    z,print2
      cp    A,0               ;Also terminate with 0's
      JP    Z,print2
      LD    C,A
      call  ZCO
      jp    print1
print2:     pop   BC
      POP   AF
      ret

;-----------------------------------------------------------------------------------------------------

SIGNON:            DB    CR,LF,LF
            DB    'Test HT6542B PC Keyboard & Mouse controller chip on MSDOS Support Board.'
            DB    CR,LF,'$'
INIT_ERR:   DB    CR,LF,BELL
            DB    'Error:  The 0xAA Test of Controller did nor return 0x55. Program Halted.'
            DB    CR,LF,'$'
INIT_OK:    DB    CR,LF
            DB    'The 0xAA Test of Controller returned 0x55. Now enter keyboard keys.'
            DB    CR,LF,LF,'$'

SCAN_MSG:   DB    'Scancode = $'
UPKEY_MSG:  DB    '(Up Keystroke)$'
CAPS_MSG:   DB    '(Caps Lock)$'
SHIFT_MSG:  DB    '(Shift Key)$'
CTRL_MSG:   DB    '(CTRL Key)$'
NUM_MSG:    DB    '(NUM Key)$'
IBM1_MSG:   DB    'Table 1 lookup -> $'
IBM2_MSG:   DB    '    Table 2 lookup -> $'


IBM1TBL:                 ;The "Normal" table
                  ;00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0a, 0b, 0c, 0d, 0e, 0f
            DB        0,'*',  0,'*','*','*','*','*',  0,'*','*','*','*',09H,'`',00H

                  ;10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1a, 1b, 1c, 1d, 1e, 1f
            DB        0,  0,  0,  0,  0,'q','1',  0,  0,  0,'z','s','a','w','2',0

                  ;20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2a, 2b, 2c, 2d, 2e, 2f
            DB        0,'c','x','d','e','4','3',  0,  0,' ','v','f','t','r','5',0

                  ;30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3a, 3b, 3c, 3d, 3e, 3f
            DB        0,'n','b','h','g','y','6',  0,  0,  0,'m','j','u','7','8',0

                  ;40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 4a, 4b, 4c, 4d, 4e, 4f
            DB        0,',','k','i','o','0','9',  0,  0,'.','/','l',';','p', '-',0

                  ;50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 5a, 5b, 5c, 5d, 5e, 5f
            DB        0,  0,27H,  0,'[','=',  0,  0,  0,  0,0DH,']',  0,'\',   0,0

                  ;60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 6a, 6b, 6c, 6d, 6e, 6f
            DB        0,  0,  0,  0,  0,  0,08H,  0,  0,11H,  0,13H,10H,  0,  0,  0
```

```
         ;70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 7a, 7b, 7c, 7d, 7e, 7f
      DB  0BH,7FH,03H,15H,04H,05H,1BH,00H,'*',02H,18H,16H,0CH,17H,'*',0

         ;80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 8a, 8b, 8c, 8d, 8e, 8f
      DB     0,  0,  0,'*',  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0


IBM2TBL:    ;If the SHIFT key or CAPS lock key is on
         ;00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0a, 0b, 0c, 0d, 0e, 0f
      DB  0, '*', 0, '*','*','*','*','*',  0,'*','*','*','*',09H,'~',00H

         ;10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1a, 1b, 1c, 1d, 1e, 1f
      DB    0,  0,  0,  0,  0,'Q','!',  0,  0,  0,'Z','S','A','W','@',0

         ;20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2a, 2b, 2c, 2d, 2e, 2f
      DB    0,'C','X','D','E','$','#',  0,  0,' ','V','F','T','R','%',0

         ;30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3a, 3b, 3c, 3d, 3e, 3f
      DB    0,'N','B','H','G','Y','^',  0,  0,  0,'M','J','U','&','*',0

         ;40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 4a, 4b, 4c, 4d, 4e, 4f
      DB    0,'<','K','I','O',29H,'(',  0,  0,'>','?','L',':','P', '_',0

         ;50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 5a, 5b, 5c, 5d, 5e, 5f
      DB    0,  0,22H,  0,'{','+',  0,  0,  0,  0,0DH,'}',  0,'|',  0,0

         ;60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 6a, 6b, 6c, 6d, 6e, 6f
      DB    0,  0,  0,  0,  0,  0,08H,  0,  0,11H,  0,13H,10H,  0,  0,  0

         ;70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 7a, 7b, 7c, 7d, 7e, 7f
      DB  0BH,7FH,03H,15H,04H,05H,1BH,00H,'*',02H,18H,16H,0CH,17H,'*',0

         ;80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 8a, 8b, 8c, 8d, 8e, 8f
      DB    0,  0,  0,'*',  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0


      DS   40H
STACK:     DB   0H
;
; END
```