

LIST ON

```

; TESTS.Z80 This is the simple debugging program for bringing
; new boards. It is relocatable anywhere in RAM. It is ugly as
; hell code because the total program contains only jump
; relative code and so will run anywhere in RAM.
; It shows top of RAM (for stack).
; It then inputs all ports 0-ffH showing any that do not
; return 0ffh.
; It then shows a memory map of the total 64K address space.
; It then repeats the whole process continuously.
;
; Author John Monahan.
;
; Recent History...
; 11/4/09 Initial program.
;
BELL EQU 07H
SPACE EQU 20H
TAB EQU 09H ;TAB ACROSS (8 SPACES FOR SD-BOARD)
CR EQU 0DH
LF EQU 0AH
FF EQU 0CH
ESC EQU 1BH
DELETE EQU 7FH
CLEAR EQU 1AH ;TO CLEAR SCREEN
;
KEYSTAT EQU 0H ;keyboard status on SD Systems Video board (change as
required)
KEYOUT EQU 1H ;keyboard output port
;
ORG 04000H

START LD A,'>' ;Note All code is position independent
OUT (KEYOUT),A ;To test if we even get here.
;
CO1a: IN A,(KEYSTAT) ;Send ' Stack=' first
AND 4H
JR Z,CO1a
LD A,CR
OUT (KEYOUT),A ;To test if we can write again
CO1b: IN A,(KEYSTAT) ;Send ' Stack=' first
AND 4H
JR Z,CO1b
LD A,LF
OUT (KEYOUT),A ;To test if we can write again

CO2: IN A,(KEYSTAT) ;Send ' Stack=' first
AND 4H
JR Z,CO2
LD A,' '
OUT (KEYOUT),A ;To test if we can write again

```

```

CO2a: IN    A,(KEYSTAT) ;Send ' Stack=' first
      AND  4H
      JR    Z,CO2a
      LD    A,'S'
      OUT  (KEYOUT),A ;To test if we can write again

CO2b: IN    A,(KEYSTAT) ;Send ' Stack=' first
      AND  4H
      JR    Z,CO2b
      LD    A,'T'
      OUT  (KEYOUT),A ;To test if we can write again

CO2c: IN    A,(KEYSTAT) ;Send ' Stack=' first
      AND  4H
      JR    Z,CO2c
      LD    A,'A'
      OUT  (KEYOUT),A ;To test if we can write again

CO2d: IN    A,(KEYSTAT) ;Send ' Stack=' first
      AND  4H
      JR    Z,CO2d
      LD    A,'C'
      OUT  (KEYOUT),A ;To test if we can write again

CO2e: IN    A,(KEYSTAT) ;Send ' Stack=' first
      AND  4H
      JR    Z,CO2e
      LD    A,'K'
      OUT  (KEYOUT),A ;To test if we can write again

CO2f: IN    A,(KEYSTAT) ;Send ' Stack=' first
      AND  4H
      JR    Z,CO2f
      LD    A,'='
      OUT  (KEYOUT),A ;To test if we can write again

```

```

                                ;CALCULATE TOP OF RAM.
      LD    HL,0FFFFH ;START FROM THE TOP DOWN
MEMSZ2: LD    A,(HL)
      CPL
      LD    (HL),A
      CP    (HL)
      CPL                                ;PUT BACK WHAT WAS THERE
      LD    (HL),A
      JR    Z,GOTTOP
      DEC  H                                ;TRY 100H BYTES LOWER
      LD    A,H
      CP    0FFH                            ;See if wrap around
      JR    NZ,MEMSZ2                        ;KEEP LOOKING FOR RAM
      LD    HL,0                            ;Set to 0 if no RAM
      JR    GOTTOP

```

```

;
SKIP10: JR    START

```

```

GOTTOP:
    LD    SP,HL        ;Setup Stack Pointer
                    ;NEXT PRINT TOP OF RAM [HL]
    LD    A,H         ;First [H] upper nibble
    RRCA
    RRCA
    RRCA
    RRCA
    AND   0FH         ;From CALL CONV (Note position independent)
    ADD   A,90H
    DAA
    ADC   A,40H
    DAA
    LD    C,A
CO2g: IN  A,(KEYSTAT) ;Send to consol
    AND   4H
    JR    Z,CO2g
    LD    A,C
    OUT   (KEYOUT),A ;To test if we can write again

    LD    A,H         ;H lower nibble
    AND   0FH         ;From CALL CONV (Note position independent)
    ADD   A,90H
    DAA
    ADC   A,40H
    DAA
    LD    C,A
CO2h: IN  A,(KEYSTAT) ;Send to consol
    AND   4H
    JR    Z,CO2h
    LD    A,C
    OUT   (KEYOUT),A ;To test if we can write again
;
;
    LD    A,L         ;Now [L] upper nibble
    RRCA
    RRCA
    RRCA
    RRCA
    AND   0FH         ;From CALL CONV (Note position independent)
    ADD   A,90H
    DAA
    ADC   A,40H
    DAA
    LD    C,A
CO2j: IN  A,(KEYSTAT) ;Send to consol
    AND   4H
    JR    Z,CO2j
    LD    A,C
    OUT   (KEYOUT),A ;To test if we can write again

    LD    A,L         ;L lower nibble
    AND   0FH         ;From CALL CONV (Note position independent)
    ADD   A,90H

```

```

DAA
ADC   A, 40H
DAA
LD    C, A
CO2k: IN  A, (KEYSTAT) ;Send to consol
AND   4H
JR    Z, CO2k
LD    A, C
OUT   (KEYOUT), A ;To test if we can write again

```

;-----

```

JR    CO3
SKIP9: JR    SKIP10
CO3:  IN  A, (KEYSTAT) ;Send 'CR,LF,Ports'
AND   4H
JR    Z, CO3
LD    A, CR
OUT   (KEYOUT), A

```

```

CO3a: IN  A, (KEYSTAT)
AND   4H
JR    Z, CO3a
LD    A, LF
OUT   (KEYOUT), A

```

```

CO3b: IN  A, (KEYSTAT)
AND   4H
JR    Z, CO3b
LD    A, 'P' ;P
OUT   (KEYOUT), A

```

```

CO3c: IN  A, (KEYSTAT)
AND   4H
JR    Z, CO3c
LD    A, 'O' ;O
OUT   (KEYOUT), A

```

```

CO3d: IN  A, (KEYSTAT)
AND   4H
JR    Z, CO3d
LD    A, 'R' ;R
OUT   (KEYOUT), A

```

```

CO3e: IN  A, (KEYSTAT)
AND   4H
JR    Z, CO3e
LD    A, 'T' ;T
OUT   (KEYOUT), A

```

```

CO3f: IN  A, (KEYSTAT)
AND   4H
JR    Z, CO3f
LD    A, 'S' ;S
OUT   (KEYOUT), A

```

```

CO3g: IN    A, (KEYSTAT) ;Send CR/LF
      AND   4H
      JR    Z, CO3g
      LD    A, CR
      OUT   (KEYOUT), A
CO3h: IN    A, (KEYSTAT)
      AND   4H
      JR    Z, CO3h
      LD    A, LF
      OUT   (KEYOUT), A
;
;---
;
      LD    B, 0           ;Now loop through all ports (0-FF)
      LD    D, 6           ;Display 6 ports across
      LD    E, 0           ;Will contain port number
      JR    LOOPIO
SKIP8  JR    SKIP9
LOOPIO: LD    C, E
      DB    0EDH, 78H      ;IN A, (C) [ZASM does not work with this opcode]
      CP    A, 0FFH        ;No need for 0FF's
      JR    Z, SKIP1
      LD    H, A           ;store port data in H for below
      LD    A, E           ;Need to print port # first
;--
      RRCA                ;First A high nibble
      RRCA
      RRCA
      RRCA
      AND   0FH           ;From CALL CONV (Note position independent)
      ADD   A, 90H
      DAA
      ADC   A, 40H
      DAA
      LD    C, A
CO4g: IN    A, (KEYSTAT)
      AND   4H
      JR    Z, CO4g
      LD    A, C
      OUT   (KEYOUT), A

      LD    A, E           ;Now A lower nibble
;
      AND   0FH           ;From CALL CONV (Note position independent)
      ADD   A, 90H
      DAA
      ADC   A, 40H
      DAA
      LD    C, A
CO4h: IN    A, (KEYSTAT)
      AND   4H
      JR    Z, CO4h
      LD    A, C

```

```

        OUT    (KEYOUT),A
;---
;
CO4a: IN     A,(KEYSTAT) ;Send '->' first
        AND   4H
        JR    Z,CO4a
        LD    A,'-'      ;-
        OUT   (KEYOUT),A
CO4b: IN     A,(KEYSTAT)
        AND   4H
        JR    Z,CO4b
        LD    A,'>'     ;>
        OUT   (KEYOUT),A
        LD    C,'>'
        JR    OVER1
SKIP1  JR    SKIP
SKIP7  JR    SKIP8
OVER1:
        LD    A,H        ;get back port data
;--
        RRCA          ;print it
        RRCA          ;First E high nibble
        RRCA
        RRCA
        AND   0FH        ;From CALL CONV (Note position independent)
        ADD   A,90H
        DAA
        ADC   A,40H
        DAA
        LD    C,A
CO4j: IN     A,(KEYSTAT)
        AND   4H
        JR    Z,CO4j
        LD    A,C
        OUT   (KEYOUT),A

        LD    A,E        ;Now E lower nibble
        AND   0FH        ;From CALL CONV (Note position independent)
        ADD   A,90H
        DAA
        ADC   A,40H
        DAA
        LD    C,A
CO4k: IN     A,(KEYSTAT)
        AND   4H
        JR    Z,CO4k
        LD    A,C
        OUT   (KEYOUT),A
;---
;
        JR    OVER2
SKIP3  JR    LOOPIO
SKIP6  JR    SKIP7
OVER2:

```

```

LD      C, ' '
CO4c: IN  A, (KEYSTAT) ;Send ' ' first
AND     4H
JR      Z, CO4c
LD      A, ' '
OUT     (KEYOUT), A
CO4d: IN  A, (KEYSTAT)
AND     4H
JR      Z, CO4d
LD      A, ' '
OUT     (KEYOUT), A

DEC     D           ;6 ports per line
JR      NZ, SKIP
LD      D, 6

CO4e: IN  A, (KEYSTAT) ;Send CR/LF
AND     4H
JR      Z, CO4e
LD      A, CR
OUT     (KEYOUT), A
CO4f: IN  A, (KEYSTAT)
AND     4H
JR      Z, CO4f
LD      A, LF
OUT     (KEYOUT), A

SKIP: INC E           ;Next Port
DJNZ   SKIP3
;
JR     CO6a
SKIP5a: JR     SKIP6
;
CO6a: IN  A, (KEYSTAT) ;Send CR/LF
AND     4H
JR      Z, CO6a
LD      A, CR
OUT     (KEYOUT), A
CO6b: IN  A, (KEYSTAT)
AND     4H
JR      Z, CO6b
LD      A, LF
OUT     (KEYOUT), A
CO7a: IN  A, (KEYSTAT) ;Send CR/LF
AND     4H
JR      Z, CO7a
LD      A, CR
OUT     (KEYOUT), A
CO7b: IN  A, (KEYSTAT)
AND     4H
JR      Z, CO7b
LD      A, LF
OUT     (KEYOUT), A
;

```

```

CO5c: IN    A, (KEYSTAT) ;Send 'MemMap'
      AND  4H
      JR    Z, CO5c
      LD    A, 'M'      ;M
      OUT  (KEYOUT), A
CO5d: IN    A, (KEYSTAT)
      AND  4H
      JR    Z, CO5d
      LD    A, 'e'      ;e
      OUT  (KEYOUT), A
CO5e: IN    A, (KEYSTAT)
      AND  4H
      JR    Z, CO5e
      LD    A, 'm'      ;m
      OUT  (KEYOUT), A
CO5f: IN    A, (KEYSTAT)
      AND  4H
      JR    Z, CO5f
      LD    A, 'M'      ;M
      OUT  (KEYOUT), A
CO5g: IN    A, (KEYSTAT)
      AND  4H
      JR    Z, CO5g
      LD    A, 'a'      ;a
      OUT  (KEYOUT), A
CO5h: IN    A, (KEYSTAT)
      AND  4H
      JR    Z, CO5h
      LD    A, 'p'      ;p
      OUT  (KEYOUT), A

```

```
;
```

```
    JR    MMAP
```

```

SKIP5:  JR    SKIP5a
MMAP: IN  A, (KEYSTAT) ;Send CR/LF
      AND  4H
      JR    Z, MMAP
      LD    A, CR
      OUT  (KEYOUT), A
CO5k: IN  A, (KEYSTAT)
      AND  4H
      JR    Z, CO5k
      LD    A, LF
      OUT  (KEYOUT), A

```

```
;
```

```
;------MEMMAP-----;
```

```
;
```

```

      LD    HL, 0
      LD    B, 1
MAP1: LD    E, 'R'      ;PRINT R FOR RAM
      LD    A, (HL)
      CPL
      LD    (HL), A
      CP   (HL)
      CPL

```



```

LD      (HL),A
JR      NZ,MAP2
CP      (HL)
JR      Z,PRINT
MAP2: LD  E,'p'
MAP3: LD  A,0FFH
CP      (HL)
JR      NZ,PRINT
INC     L
XOR     A
CP      L
JR      NZ,MAP3
LD      E, '.'
PRINT: LD  L,0
DEC     B
JR      NZ,NLINE
LD      B,16

CO8a IN  A,(KEYSTAT) ;Send CR/LF
AND    4H
JR      Z,CO8a
LD      A,CR          ;CR
OUT    (KEYOUT),A
CO8b: IN  A,(KEYSTAT)
AND    4H
JR      Z,CO8b
LD      A,LF          ;LF
OUT    (KEYOUT),A

JR      HXOT4          ;Print HL on Consol

SKIP4: JR      SKIP5

NLINE: IN  A,(KEYSTAT) ;Send a space
AND    4H
JR      Z,NLINE
LD      A,' '          ;' '
OUT    (KEYOUT),A
CO9a: IN  A,(KEYSTAT)
AND    4H
JR      Z,CO9a
LD      A,E            ;send R,P or .
OUT    (KEYOUT),A

INC     H
JR      NZ,MAP1

CO8c IN  A,(KEYSTAT) ;Send CR/LF
AND    4H
JR      Z,CO8c
LD      A,CR          ;CR
OUT    (KEYOUT),A
CO8d: IN  A,(KEYSTAT)
AND    4H

```

```

        JR    Z,CO8d
        LD    A,LF          ;LF
        OUT   (KEYOUT),A
CO8e: IN    A,(KEYSTAT)
        AND   4H
        JR    Z,CO8e
        LD    A,LF          ;LF
        OUT   (KEYOUT),A
CO8f: IN    A,(KEYSTAT)
        AND   4H
        JR    Z,CO8f
        LD    A,LF          ;LF
        OUT   (KEYOUT),A

        JR    SKIP4
;
HXOT4:                                ;Print HL
        LD    A,H          ;First [H] upper nibble
        RRCA
        RRCA
        RRCA
        RRCA
        AND   0FH          ;From CALL CONV (Note position independent)
        ADD   A,90H
        DAA
        ADC   A,40H
        DAA
        LD    C,A
COAg: IN    A,(KEYSTAT) ;Send to consol
        AND   4H
        JR    Z,COAg
        LD    A,C
        OUT   (KEYOUT),A ;To test if we can write again

        LD    A,H          ;H lower nibble
        AND   0FH          ;From CALL CONV (Note position independent)
        ADD   A,90H
        DAA
        ADC   A,40H
        DAA
        LD    C,A
COAh: IN    A,(KEYSTAT) ;Send to consol
        AND   4H
        JR    Z,COAh
        LD    A,C
        OUT   (KEYOUT),A ;To test if we can write again
;
        JR    SKIPA
NLINE1: JR    NLINE
;
SKIPA:
        LD    A,L          ;Now [L] upper nibble
        RRCA
        RRCA

```

```

RRCA
RRCA
AND 0FH ;From CALL CONV (Note position independent)
ADD A,90H
DAA
ADC A,40H
DAA
LD C,A
COAj: IN A,(KEYSTAT) ;Send to consol
AND 4H
JR Z,COAj
LD A,C
OUT (KEYOUT),A ;To test if we can write again

LD A,L ;L lower nibble
AND 0FH ;From CALL CONV (Note position independent)
ADD A,90H
DAA
ADC A,40H
DAA
LD C,A
COAk: IN A,(KEYSTAT) ;Send to consol
AND 4H
JR Z,COAk
LD A,C
OUT (KEYOUT),A ;To test if we can write again
JR NLINE1
;END

```