

```

; MYPIC.ASM Test Program to test a 8259A by John Monahan (monahan@vitasoft.org)
; V0.2 3/28/2010 ;First version
; V1.0 11/14/2010 ;Version to test final S-100 Board
;
;This is a simple test program to show the use of the PIC/RTC board
;using Z80 Mode0 Interrupts.

PICportA EQU 020H
PICportB EQU 021H
;
;SD Systems Video Board for all character I/O. Modify the port for other CRT's
;
CONOUT$PORT EQU 1H ;Console out port
CONIN$PORT EQU 1H ;Console In (keyboard) port
CONSTAT$PORT EQU 0H ;Consol status port

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

ORG 100H

begin: LXI SP,STACK
      LXI H,SIGN$ON ;Print a welcome message
      CALL PSTRING
      CALL ZCI ;Wait for start key
      LXI H,CRLF$MSG ;print a CR/LF
      CALL PSTRING

      DI ;No INT's until we get things setup
; DB 0EDH, 5EH ;IM2 mode for Z80 (Uses [I] register for high byte
address)
; DB 0EDH, 56H ;IM1 mode for Z80 (Jump to 38H in RAM always)
DB 0EDH, 46H ;IM0 8080 mode <-----<<<<<<<<<,
; MVI A,0H ;Set [I] register for 0H
; DB 0EDH,47H ;Mov I,A

;First we need to setup the RST vectors in low memory

;Setup Z80 Int. Mode 0 Int. Jump vectors (0 is used by
CP/M, skip)

MVI A,0C3H ;@08H JMP RST1 Vector Location
STA 0008H
LXI H,RST1
SHLD 009H

STA 0010H ;@10H JMP to RST2 vector location
LXI H,RST2
SHLD 11H

STA 0018H ;@18H JMP to RST3 vector location
LXI H,RST3
SHLD 19H

STA 0020H ;@20H JMP to RST4 vector location
LXI H,RST4
SHLD 21H

STA 0028H ;@28H JMP to RST5 vector location
LXI H,RST5
SHLD 29H

STA 0030H ;@30H JMP to RST6 vector location

```

```

LXI   H,RST6
SHLD  31H

STA   0038H           ;@38H JMP to RST7 vector location
LXI   H,RST7
SHLD  39H

;Setup the 8259A PIC Controller (For Poll mode)
MVI   A,00010111B    ;Set for A5-7 of service routine=0
;Edge triggered, Routine interval=4
;Single 8259, and IVW4 needed
OUT   PICportA       ;Send it (This is ICW1)

MVI   A,00H          ;Upper Byte of service routine table (00H) Does not
really matter
;since we are running in poll mode here anyway
OUT   PICportB       ;Send it (ICW2)

OUT   PICportB       ;No slaves ICW3

MVI   A,00000010B    ;Set not fully nested, non buffered mode, NON-auto EOI
;and not 8086 mode
OUT   PICportB       ;Send it (ICW4)

;Now send Output control bytes....

MVI   A,0H           ;Enable ALL bits
OUT   PICportB       ;Send it (OCW1)

MVI   A,10100000B    ;Set to auto rotating priority
OUT   PICportA       ;Send it (OCW2)

MVI   A,00001100B    ;Set 8259A to Poll mode
OUT   PICportA       ;Send it, (OCW3)

EI           ;Int's can now be handled in poll mode
IN     PICportA       ;It's not clear to me why but one input here is required
;to get the ints/8259A going

LOOP: EI           ;This is a do nothing loop. The program will reside here
NOP        ;unless it is interrupted.
NOP
NOP
DI
MVI   C,','         ;Show things are alive by putting a characater on CRT
CALL  ZCO
CALL  ZCSTS         ;Anything at Keyboard?
JZ    LOOP
CALL  ZCI
CPI   ESC           ;Abort back to CPM (0H, Reboot) if an ESC key
JNZ   LOOP
JMP   0H

;
;----- Interrupt routines -----
;
RST1: DI           ;Don't allow another Int until we finish this one
      PUSH  H
      LXI   H,SIGN$MODE0           ;Print a welcome message
      JMP   IntRoutine

RST2: DI           ;Don't allow another Int until we finish this one

      PUSH  H
      LXI   H,SIGN$MODE1

```

```

        JMP     INTRoutine

RST3:  DI                      ;Don't allow another Int until we finish this one
        PUSH  H
        LXI   H, SIGN$MODE2
        JMP   IntRoutine

RST4:  DI                      ;Don't allow another Int until we finish this one
        PUSH  H
        LXI   H, SIGN$MODE3
        JMP   IntRoutine

RST5:  DI                      ;Don't allow another Int until we finish this one
        PUSH  H
        LXI   H, SIGN$MODE4
        JMP   IntRoutine

RST6:  DI                      ;Don't allow another Int until we finish this one
        PUSH  H
        LXI   H, SIGN$MODE5
        JMP   IntRoutine

RST7:  DI                      ;Don't allow another Int until we finish this one
        PUSH  H
        LXI   H, SIGN$MODE6
        JMP   IntRoutine

IntRoutine:
        PUSH  PSW                ;Good practice not to alter any registers
        PUSH  B                  ;Note in a real world system we would save and setup a
local stack as well.

        MVI   A, 00001111B      ;Need to get Int status right away
PicPortA
        OUT   PICportA          ;Send 8259A a Poll command, Reads IS Register on next IN
        IN    PICportA          ;Send it, (OCW3)
        STA   IntStatus          ;Get and show Bit pattern returned.
        CALL  PSTRING            ;Store it for later display below
        ;print what HL point to (any one of teh above RST
vectors)
        LDA   IntStatus          ;What were the status bits
        CALL  ZBITS              ;Display the 8259A poll status

        MVI   A, 20H             ;Send EOI now
        OUT   PICportA          ;Send it (OCW2)
        POP   B
        POP   PSW
        POP   H                  ;Be sure to balance up the stack
        RET

;----- SUPPORT ROUTINES -----

ZCO:   PUSH  PSW                ;Write character that is in [C]
ZCO1:  IN    0H                  ;Show Character
        ANI   04H
        JZ    ZCO1
        MOV   A, C
        OUT  CONOUT$PORT
        POP   PSW
        RET

ZCSTS: IN    0H                  ;Get Keyboard Status
        ANI   02H
        RET

```

```
ZCI:  IN    CONSTAT$PORT      ;Return keyboard character in [A]
      ANI   02H
      JZ    ZCI
      IN    CONIN$PORT
      RET
```

```
PSTRING:MOV A,M              ;Print string @ [HL] up to 0
        ORA  A
        RZ
        MOV  C,A
        CALL ZCO
        INX  H
        JMP  PSTRING
```

```
PHEX:  PUSH  PSW              ;Print HEX [A]
        PUSH  B
        PUSH  PSW
        RRC
        RRC
        RRC
        RRC
        CALL  ZCONV
        POP   PSW
        CALL  ZCONV
        MVI  C,' '
        CALL  ZCO
        POP  B
        POP  PSW
        RET
```

```
;
ZCONV:  ANI   0FH              ;HEX to ASCII and print it
        ADI  90H
        DAA
        ACI  40H
        DAA
        MOV  C,A
        CALL ZCO
        RET
```

```
;
ZCRLF:  PUSH  PSW
        MVI  C,CR
        CALL ZCO
        MVI  C,LF
        CALL ZCO
        POP  PSW
        RET
```

```
;
;DISPLAY BIT PATTERN IN [A]
;
```

```
ZBITS:  PUSH  PSW
        PUSH  B
        PUSH  D
        MOV  E,A
        MVI  B,8
BQ2:    DB    0CBH,23H        ;Z80 Op code for SLA A,E
        MVI  A,18H
        ADC  A
        MOV  C,A
        CALL ZCO
        DCR  B
        JNZ  BQ2
        MVI  C,' '
```

```

CALL ZCO
POP D
POP B
POP PSW
CALL ZCRLF
RET
;
SIGN$ON: DB 'PIC/RTC board Z80 Interrupts Test Program (1.0) ',CR,LF
DB 'Will use Z80 interrupt mode IM2 [I=0].',CR,LF
DB 'Simple test of S-100 bus INT pins V0-V7.',CR,LF
DB 'Connect SMB system tick to any one of the S-100 Vector Interrupt
lines.',CR,LF
DB 'Press any key to start. Esc to Abort... ',0
CRLF$MSG: DB CR,LF,0
SIGN$MODE0: DB CR,LF,'Jump to Int Routine at 08H,'
DB ' 8259A Poll status bits = ',0
SIGN$MODE1: DB CR,LF,'Jump to Int Routine at 10H,'
DB ' 8259A Poll status bits = ',0
SIGN$MODE2: DB CR,LF,'Jump to Int Routine at 18H,'
DB ' 8259A Poll status bits = ',0
SIGN$MODE3: DB CR,LF,'Jump to Int Routine at 20H,'
DB ' 8259A Poll status bits = ',0
SIGN$MODE4: DB CR,LF,'Jump to Int Routine at 28H,'
DB ' 8259A Poll status bits = ',0
SIGN$MODE5: DB CR,LF,'Jump to Int Routine at 30H,'
DB ' 8259A Poll status bits = ',0
SIGN$MODE6: DB CR,LF,'Jump to Int Routine at 38H,'
DB ' 8259A Poll status bits = ',0

IntStatus: DB 0H ;Store 8259A poll status bits here

DS 200H ;Make a large stack since we have non stop ints
STACK: DW 0H
STACKSTORE DW 0H

;END

```