

```
;-----title:      voltage test-----
;-----contain:    voltage measuring,digital display,alarm
;-----oscillator: about 4MHz-----
;-----TIMEO:      used for display
;-----TIME1:      used for alarm
        list p=16f870
        #include <p16f870.inc>
msxxr   EQU 20h ;--FOR interrupt interval time
datah   EQU 21h ;\\\\\\
data1   EQU 22h ;///// For result of ADC (average)
adrh    EQU 23h ;\\
adrl    EQU 24h ;//data in hex of adc after calculating
data3   EQU 25h ;\\
data2   EQU 26h ;\\
data1   EQU 27h ;//four digit in decimal of ADC
data0   EQU 28h ;//
displ3  EQU 29h ;\\
displ2  EQU 2ah ;\\
displ1  EQU 2bh ;//four digit of display
displ0  EQU 2ch ;//
dispaddr EQU 2dh ;\\
dispbset EQU 2eh ;//for display
ctoph   EQU 2fh ;\\
ctopl   EQU 30h ;\\
clowh   EQU 31h ;//for comparation utmost
clowl   EQU 32h ;//
cpregh  EQU 33h ;\\
cpregl  EQU 34h ;//for comparation subroutine
temp    EQU 35h
tempi   EQU 36h
sbuf    EQU 37h
fbuf    EQU 38h
pchbuf  EQU 39h
mystatus EQU 3ah
mute    EQU 3bh
bcdreg1 EQU 3ch
bcdreg2 EQU 3dh
bcdreg3 EQU 3eh
h_byte  EQU 3fh
l_byte  EQU 40h
count   EQU 41h
dreg    EQU 42h
delayk  EQU 43h
reg7    EQU 44h
adcount EQU 45h
cblock  0x50
delay0
delay1
delay6
endc

;-----mystatus bit define-----
        keyin   EQU 0
        overflow EQU 1
        alarm    EQU 2

;-----note:50~6fh have used for adc result buffer
;////////////////////////////////////
        wbuf    EQU 70h
```

```

        org 0x00
        goto main
        org 0x04
        goto int
        org 0x0a
;-----interrupt subroutine-----
int     movwf wbuf           ;;\
        movf  STATUS,w      ;;\
        bcf  STATUS,RP0     ;;\
        bcf  STATUS,RP1     ;;\
        movwf sbuf         ;;==save specail register
        movf  FSR,w         ;;//
        movwf fbuf         ;;//
        movf  PCLATH,w      ;;//
        movwf pchbuf       ;;//
        btfss INTCON,T0IF
        goto intalarm
        bcf  INTCON,T0IF
        movlw 0xe3          ;;==timer0 about 50ms interval interrupt
        movwf TMR0         ;;//256*193=49408us=49ms
;-----display below-----
disp    call  output
        bsf  STATUS,RP0
        bcf  STATUS,RP1
        bcf  TRISB,4        ;;\
        bcf  TRISB,5        ;;==PORTB.4~PORTB.7 are used for digit selection
        bcf  TRISB,6        ;;//
        bcf  TRISB,7        ;;//
        movlw 0x00
        movwf TRISC
        bcf  STATUS,RP0
        btfss dispbsel,7
        goto dispb7
        btfss dispbsel,6
        goto dispb6
        btfss dispbsel,5
        goto dispb5
dispb4  bsf  PORTB,5
        bsf  PORTB,6
        bsf  PORTB,7
        bcf  PORTB,4
        goto com
dispb5  bsf  PORTB,6
        bsf  PORTB,7
        bsf  PORTB,4
        bcf  PORTB,5
        goto com
dispb6  bsf  PORTB,7
        bsf  PORTB,4
        bsf  PORTB,5
        bcf  PORTB,6
        goto com
dispb7  bsf  PORTB,4
        bsf  PORTB,5
        bsf  PORTB,6
        bcf  PORTB,7
com     bsf  STATUS,C
        rrf  dispbsel

```

```

        movf  dispaddr,w
        movwf FSR
        movf  INDF,w
        movwf PORTC
        incf  dispaddr
        movf  dispaddr,w
        xorlw 0x2d
        btfss STATUS,Z
        goto  intret
        movlw 0x29 ;;;;;;;;;;display start address
        movwf dispaddr
        movlw 0x7f
        movwf dispbsel
        goto  intret
intalarm bcf  PIR1,TMR1IF
        movlw 0xdf
        movwf TMR1L
        movlw 0xb1
        movwf TMR1H
        incf  mute
        bsf   STATUS,RP0
        bcf   TRISB,1
        bcf   STATUS,RP0
        btfss mute,0
        goto  outhi
        bcf   PORTB,1
        goto  intret
outhi   bsf   PORTB,1
intret  nop
        movf  pchbuf,w
        movwf PCLATH
        movf  sbuf,w
        movwf STATUS
        movf  fbuf,w
        movwf FSR
        movf  wbuf,w
        retfie
;-----output subroutine-----
output  movf  data0,w
        call  table
        addlw 0x80
        movwf displ0
        movf  data1,w
        call  table
        movwf displ1
        movf  data2,w
        call  table
        movwf displ2
        movf  data3,w
        call  table
        movwf displ3
        return
table   addwf  PCL
        retlw 0x40 ;0
        retlw 0x79 ;1
        retlw 0x24 ;2
        retlw 0x30 ;3
        retlw 0x19 ;4
        retlw 0x12 ;5

```

```

        retlw  0x02 ;6
        retlw  0x78 ;7
        retlw  0x00 ;8
        retlw  0x10 ;9
;-----get average in 16 bytes subroutine-----
average  movlw  0x10
        movwf  temp
        movlw  0x6f ;;;  adc result endless address
        movwf  FSR
        movlw  0x00
        movwf  datah
        movwf  datal
lpc      clrwdt
        movf   INDF,w
        addwf  datal
        btfsc  STATUS,C
        incf   datah
        decf   FSR
        movf   INDF,W
        addwf  datah
        decf   FSR
        decfsz temp
        goto  lpc
        movlw  0x04
lpd      movwf  temp
        bcf   STATUS,C
        rrf   datah
        rrf   datal
        decfsz temp
        goto  lpd
        btfsc STATUS,C
        incf  datal
        return

;-----multi subroutine---k=2mv
multiply bcf   STATUS,RPO
        bcf   STATUS,RPL
        bcf   STATUS,C
        rlf   datal
        rlf   datah
        return

;-----for coparation result subroutine-----
comp     clrwdt
compha  bcf   mystatus,overflow
        movf  clowh,w
        subwf cpregh,w
        btfsc STATUS,Z
        goto  compl1
        btfss STATUS,C
        goto  over
        goto  compb
compl1  movf  clowl,w
        subwf cpregl,w
        btfss STATUS,C
        goto  over ;=====cpreg>clow? not jump to over;yes,continue
compb   movf  cpregh,w
        subwf ctoph,w
        btfsc STATUS,Z

```

```

        goto    compl2
        btfss  STATUS,C
        goto    over
        goto    result
compl2  movf    cpregl,w
        subwf  ctopl,w
        btfsc  STATUS,C
        goto    result ;=====cpreg<ctop?
over    bsf    mystatus,overflow
result  return
;-----for hex to bcd subroutine-----
bcd     BCF    STATUS,RP0
        BCF    STATUS,RP1
        MOVF   adrh,w
        MOVWF  h_byte
        MOVF   adr1,w
        MOVWF  l_byte
        MOVLW  .16
        MOVWF  count
        CLRF   bcdreg1
        CLRF   bcdreg2
        CLRF   bcdreg3
        BCF    STATUS,C

SHIFT   RLF    l_byte
        RLF    h_byte
        RLF    bcdreg3
        RLF    bcdreg2
        RLF    bcdreg1
        clrwdt
        DECFSZ count
        GOTO   ADJDEC
        return
;-----adjdec-----
ADJDEC  MOVLW   bcdreg3
        MOVWF  FSR
        CALL  ADJBCD
        MOVLW  bcdreg2
        MOVWF  FSR
        CALL  ADJBCD
        MOVLW  bcdreg1
        MOVWF  FSR
        CALL  ADJBCD
        GOTO   SHIFT

ADJBCD  MOVLW   0X03
        ADDWF  INDF,W
        MOVWF  temp
        BTFSC  temp,3
        MOVWF  INDF
        MOVLW  0X30
        ADDWF  INDF,W
        MOVWF  temp
        BTFSC  temp,7
        MOVWF  INDF
        RETLW  0X00
;-----delay subroutine about 0.5/3*delaykms time delay
delay   nop
deloopa movlw  0x7b

```

```

        movwf dreg
deloopb  clrwdt
        decfsz dreg
        goto deloopb
        decfsz delayk
        goto deloopa
        return
;-----main-----
main     nop
        clrwdt
        bcf STATUS,RP0
        bcf STATUS,RP1
        movlw 0xfa
        movwf delayk
        call delay
        movlw 0x00
        movwf mystatus
        movlw 0x70
        movwf dispbsel
        movlw 0x29
        movwf dispaddr
;=====for test=====
;=====the end of test=====
        clrfs data0
        clrfs data1
        clrfs data2
        clrfs data3
;-----enable timer interrupt-----
        movlw 0xe3 ;\
        movwf TMR0 ;\
        bsf STATUS,RP0
        movlw 0x07 ;==Enable INT T0,about ??ms
        movwf OPTION_REG ;//
        bsf INTCON,T0IE ;//
        bcf STATUS,RP0
        movlw 0xf0 ;\
        movwf TMR1H ;\
        movlw 0x5f ;\
        movwf TMR1L ;\
        bsf STATUS,RP0 ;==Enable timer1 interrupt ,about 10ms
        bsf PIE1,TMR1IE ;//
        bcf STATUS,RP0 ;//
        bsf INTCON,PEIE ;//
        bcf T1CON,TMR1ON;//
        bsf INTCON,GIE ;//

;-----set adc mode-----
        bsf STATUS,RP0
        movlw 0x8f
        movwf ADCON1
        bcf PIE1,ADIE
        bcf STATUS,RP0
        movlw 0x80
        movwf ADCON0
;-----adc loop-----
once     movlw 0x10
        movwf adcount
        movlw 0x50 ;adc start address

```

```

movwf FSR
adcloop  clrwdt
        bsf   STATUS,RP0
        bcf   STATUS,RP1
        bsf   TRISA,0
        bcf   STATUS,RP0
        bsf   ADCON0,ADON
        movlw 0x28
        movwf reg7
delay2   clrwdt
        decfsz reg7
        goto  delay2  ;;;;;;;;;;about 60us
        bsf   ADCON0,GO;;;;;;;;;start AD
wait     clrwdt
        btfss PIR1,ADIF
        goto  wait
        bcf   PIR1,ADIF
        nop
        movf  ADRESH,w
        movwf INDF
        incf  FSR
        bsf   STATUS,RP0
        movf  ADRESL,w
        bcf   STATUS,RP0
        movwf INDF
        incf  FSR
        movlw 0x1f;;;;;;;;;????????
        movwf delayk
        call  delay
        decfsz adcount
        goto  adcloop
;-----
        call  average
        call  multiply
        movf  data1,w ;\\
        movwf adrl  ;\\
        movf  datah,w ;== ready for bcd
        movwf adrh  ;//
        call  bcd
;-----given 4 digit for display-----
        movf  bcdreg3,w
        movwf cpregl  ;;;;;;;;;;for comparation high byte
        andlw 0x0f
        movwf data0
        swapf bcdreg3,w
        andlw 0x0f
        movwf data1
        movf  bcdreg2,w
        movwf cpregh  ;;;;;;;;;;for comparation low byte
        andlw 0x0f
        movwf data2
        swapf bcdreg2,w
        andlw 0x0f
        movwf data3
        movlw 0x15 ;\\
        movwf ctoph;\\
        movlw 0x06 ;\\
        movwf ctopl;\\
        movlw 0x14 ;//=====utmost for comparation

```

```
movwf clowh; //
movlw 0x94 ; //
movwf clowl; //
call comp
btfss mystatus, overflow
goto alarmoff
bsf T1CON, TMR1ON
goto once
alarmoff bcf T1CON, TMR1ON
goto once
end
```

<http://gongkong.gkbk.com>

pdfMachine - is a pdf writer that produces quality PDF files with ease!
Get yours now!

"Thank you very much! I can use Acrobat Distiller or the Acrobat PDFWriter but I consider your product a lot easier to use and much preferable to Adobe's" A.Sarras - USA