

```

;
; Super Probe - Maximum Functions from Minimum Parts
;
; Current Functions
; -----
; Logic Probe
; Voltmeter
; Frequency Counter
; Event Counter
; Capacitor Test
; Signal Generator
; Diode Tester
; Logic Pulser
; Serial Generator (4 rates)
; Midi Generator
; r/c servo pulse
; Square Wave
; 20khz digital noise
; 38KHz burst
;
; 29 added r/c square noise burst baudrates
; 29 added memu forward/back
; 30 added pwm function
; 31 reworked square wave for interrupt operation
; 32 fix logic probe function
; 33 (compatibility with new assembler)
; 34 fix dpcap syntax for hex values - last 2
; 35 add inductance measure

list      p=16f870          ; list directive to define processor
#include <p16f870.inc>      ; processor specific variable definitions

errorlevel -302          ; Register not in bank 0 warning

radix dec

__CONFIG _CP_OFF & _WDT_OFF & _BODEN_OFF & _PWRTE_ON & _HS_OSC & _LVP_OFF
& _DEBUG_OFF & _CPD_OFF

#define VERS              035

#define LPTIME            42990          ;calibration time for counter

#define MAXMODE           17            ;total operational modes

#define CLAMP             PORTA,5        ;clamp input to T0
#define R20               PORTA,0        ;20 ohm direct feed
#define R100              PORTA,1        ;100 ohms
#define R470              PORTA,4        ;470 ohms
#define R150              PORTA,2        ;150
#define R10K              PORTC,7        ;10k
#define R10KA             PORTC,6        ;10k
#define R100K             PORTA,3        ;100k feed to input

#define BLACK             R150
#define WHITE             R100
#define SOUT              R20

#define BUT1              PORTC,4        ;button 1
#define BUT2              PORTC,5        ;button 2

org 20h

wsave equ 20H

```

```

ssave equ    21H
icount      equ    22H
flag equ    23H
count equ    24H
scount      equ    25H
csave equ    26H
ddata equ    27H          ; was data
sdata equ    28H
temp equ    29H
ontime      equ    2BH
oftime      equ    2CH
isrvec      equ    2DH
qtime equ    2EH          ;comm timer for serial
timer equ    30H
freq equ    33H          ;
pwmp equ    35H
mode equ    36H
rate equ    37H
baud equ    38H
midic equ    39H          ;midi channel
pmode equ    3AH
hi equ    3BH
lo equ    3CH          ; was low
rand equ    3DH
acc equ    40H          ;32 bit register
xacc equ    44H          ;aux 32 bit value
digits      equ    48H          ;segment data for LED
dignr equ    4CH
dp equ    4DH
ftimer      equ    4EH
segmask     equ    50H
dela equ    51H
bcd equ    54H          ;10 digit bcd buffer
pto equ    59H
pti equ    5AH
cnt equ    5BH
ii equ    5CH
frame equ    5DH

```

```

#define HOLD          flag,0
#define DIR           flag,1
#define SAVE          flag,2
#define RUN           flag,3
#define SWIT          flag,4

```

```

org 0

goto start

org 4

isr movwf wsave
    swapf STATUS,w
    movwf ssave
    clrf STATUS

    movf isrvec,w
    addwf PCL,f
    goto pwmi
    goto sqri

isrx swapf ssave,w

```

```
movwf STATUS
swapf wsave, f
swapf wsave, w
retfie
```

```
digseg      addwf PCL, f
            retlw 3fh
            retlw 6h
            retlw 5bh
            retlw 4fh
            retlw 66h
            retlw 6dh
            retlw 7dh
            retlw 7h
            retlw 7fh
            retlw 67h
```

```
getano      addwf PCL, f
            retlw 1
            retlw 2
            retlw 4
            retlw 8
```

```
; bit 7 decimal point
; bit 6 flash decimal
; bit 5,4 dp location
; bit 3-0 starting digit from bcd buffer
```

```
getdp addwf PCL, f
            retlw 03h
            retlw 03h
            retlw 03h
            retlw 03h
            retlw 94h
            retlw 0A5h
            retlw 0C6h
            retlw 0D7h
            retlw 0E8h
            retlw 0F9h
```

```
dpcap addwf PCL, f
            retlw 01h
            retlw 01h
            retlw 84h
            retlw 84h
            retlw 84h
            retlw 95h
            retlw 0A6h
            retlw 0B7h
            retlw 0C8h
            retlw 0D9h
```

```
dpuls addwf PCL, f
            retlw 170
            retlw 55
            retlw 17
            retlw 5
```

```
srq movf temp, w
      incf temp, f
      addwf PCL, f
      retlw low 1380 ;1200
      retlw high 1380 ;1200
```

```

    retlw low 680           ;2400
    retlw high 680        ;2400
    retlw low 330         ;4800
    retlw high 330        ;4800
    retlw low 160         ;9600
    retlw high 160        ;9600

modes movf temp,w
      incf temp,f
      addwf PCL,f
vbase retlw 73h           ;Prob
      retlw 50h           ;Prob
      retlw 5ch           ;Prob
      retlw 7ch           ;Prob
      retlw 73h           ;PULS
      retlw 3eh           ;PULS
      retlw 38h           ;PULS
      retlw 6dh           ;PULS
      retlw 3eh           ;VoLt
      retlw 5ch           ;VoLt
      retlw 38h           ;VoLt
      retlw 78h           ;VoLt
      retlw 05eh          ;diod
      retlw 4             ;diod
      retlw 5ch           ;diod
      retlw 05eh          ;diod
      retlw 71h           ;FrEq
      retlw 50h           ;FrEq
      retlw 79h           ;FrEq
      retlw 6fh           ;FrEq
      retlw 0             ;Cnt
      retlw 39h           ;Cnt
      retlw 54h           ;Cnt
      retlw 78h           ;Cnt
      retlw 0             ;CAP
      retlw 39h           ;CAP
      retlw 77h           ;CAP
      retlw 73h           ;CAP
      retlw 39h           ;Coil
      retlw 5ch           ;Coil
      retlw 4             ;Coil
      retlw 38h           ;Coil
      retlw 0             ;SIG
      retlw 6dh           ;SIG
      retlw 6             ;SIG
      retlw 3dh           ;SIG
      retlw 54h           ;ntSC
      retlw 78h           ;ntSC
      retlw 6dh           ;ntSC
      retlw 39h           ;ntSC
      retlw 00h          ;SEr
      retlw 6dh           ;SEr
      retlw 79h           ;SEr
      retlw 50h           ;SEr
      retlw 15h          ;Midi
      retlw 4             ;Midi
      retlw 5eh           ;Midi
      retlw 4             ;Midi
      retlw 21h          ;r/c
      retlw 52h          ;r/c
      retlw 58h          ;r/c
      retlw 0             ;r/c
      retlw 39h          ;[]
      retlw 0fh          ;[]

```

```

    retlw 00h          ;[]
    retlw 0            ;[]
    retlw 73h         ;Prn
    retlw 50h         ;Prn
    retlw 54h         ;Prn
    retlw 0           ;Prn
    retlw 04h         ;ir38
    retlw 50h         ;ir38
    retlw 4fh         ;ir38
    retlw 7fh         ;ir38
    retlw 73h         ;Pwm
    retlw 2ah         ;Pwm
    retlw 15h         ;Pwm
    retlw 0           ;Pwm

vpuls retlw 0         ; 5
      retlw 0         ; 5
      retlw 0         ; 5
      retlw 6dh       ; 5
      retlw 0         ; 50
      retlw 0         ; 50
      retlw 6dh       ; 50
      retlw 3fh       ; 50
      retlw 0         ; 500
      retlw 6dh       ; 500
      retlw 3fh       ; 500
      retlw 3fh       ; 500
      retlw 0         ; 5.0
      retlw 0         ; 5.0
      retlw 0edh     ; 5.0
      retlw 3fh       ; 5.0

vbaud retlw 06h       ;1200
      retlw 5bh       ;1200
      retlw 3fh       ;1200
      retlw 3fh       ;1200
      retlw 5bh       ;2400
      retlw 66h       ;2400
      retlw 3fh       ;2400
      retlw 3fh       ;2400
      retlw 66h       ;4800
      retlw 7fh       ;4800
      retlw 3fh       ;4800
      retlw 3fh       ;4800
      retlw 67h       ;9600
      retlw 7dh       ;9600
      retlw 3fh       ;9600
      retlw 3fh       ;9600

; execute routine as selected from menu

exec  movf mode,w
      addwf PCL,f
      goto lp1        ;logic probe
      goto pulse      ;logic pulser
      goto volt       ;voltmeter
      goto diode      ;doide voltage
      goto fcount     ;frequency count
      goto dcount     ;event count
      goto cap        ;cap meter
      goto induct     ;inducance meter
      goto sig        ;signal generator
      goto vgen       ;video generator
      goto s9600      ;serial out

```

```

        goto midi          ;midi out
        goto radcon       ;servo pulse
        goto square      ;square wave
        goto noise       ;digital noise
        goto ir38        ;38 khz test signal
        goto pwm         ;variable pulse width

start bsf    STATUS,5
        movlw b'00111111'
        movwf TRISA
        movlw 0
        movwf TRISB
        movlw b'10110000'
        movwf TRISC
        movlw b'10001000'
        movwf OPTION_REG
        bcf    STATUS,5
        movlw 4
        movwf T2CON
        clrf  segmask
        bsf   segmask,0
        clrf  dignr
        call  getset          ;restore current mode
        call  signon
        movlw low 1000
        movwf freq
        movlw high 1000
        movwf freq+1
        movlw 48
        movwf pwmp

; menu selection of operational modes

cycmode    call  putset          ;save all parameters

cycle bcf    INTCON,GIE
        movf  mode,w
        call  setmode          ;load segment menu data

c2    call  segout          ;run display
        movlw 10
        call  delay
        btfsc BUT1          ;still held down?
        goto  c4          ;no
        btfss BUT2
        goto  c2

c5    call  segout
        movlw 10
        call  delay
        btfsc BUT1
        goto  doexec
        btfsc BUT2
        goto  c5
        decf  mode,f
        movlw MAXMODE-1
        btfsc mode,7          ;underflow?
        movwf mode
        goto  cycle

c4    call  segout
        movlw 10
        call  delay

```

```

    btfsc BUT2
    goto doexec          ;both buttons up - execute
    btfsc BUT1
    goto c4
    incf mode,f
    movf mode,w
    xorlw MAXMODE
    btfsc STATUS,Z
    clrf mode
    goto cycle

doexec    call putset          ;first, save op mode
          call clreg
          goto exec

clreg    bsf STATUS,RP0      ;then clear all lines
          bsf R20
          bsf R100
          bsf R470
          bsf R150
          bsf R10K
          bsf R10KA
          bsf R100K
          bsf CLAMP
          movlw 6
          movwf ADCON1
          bcf STATUS,RP0
          return

; save current setup in eeprom

putopt    btfss SAVE          ;flagged for save?
          return              ;no.
          bcf SAVE            ;yes.
putset    clrf hi
          clrf lo
          movf mode,w
          call eewrite        ;save current mode in eeprom
          incf lo,f
          movf pmode,w
          call eewrite
          incf lo,f
          movf midic,w
          call eewrite
          incf lo,f
          movf baud,w
          call eewrite
          return

; get op mode from eeprom

getset    clrf hi
          clrf lo
          call eeread
          movwf mode
          movf mode,w
          addlw -MAXMODE
          btfsc STATUS,C
          clrf mode          ;restore
          incf lo,f          ;next location
          call eeread
          movwf pmode        ;pulse mode

```

```

movf  pmode,w
addlw -4
btfsc STATUS,C
clrf  pmode
incf  lo,f
call  eeread
movwf midic
addlw -15
btfsc STATUS,C
clrf  midic
incf  lo,f      ;next location
call  eeread
movwf baud      ;pulse mode
movf  baud,w
addlw -4
btfsc STATUS,C
clrf  baud
return

```

; put mode text on 4 char display

```

setmode    movwf temp
           bcf   STATUS,C
           rlf  temp,f
           rlf  temp,f
           call modes
           movwf digits
           call modes
           movwf digits+1
           call modes
           movwf digits+2
           call modes
           movwf digits+3
           return

```

; show version nr and op mode at startup

```

signon    movlw VERS
           call  wtod      ;put version nr
           movlw 0        ;blank.
           movwf digits
           call  sign
           movf  mode,w
           call  setmode
           call  sign
           return

```

; run display slow for flashing effect

```

sign      clrf  count
x1        call  segout
           movlw 15
           call  delay
           decfsz count,f
           goto  x1
           return

```

; PWM interrupt handler

```

pwmi      bcf   INTCON,T0IF
           btfss SWIT
           goto  on

```



```

off    bcf    SWIT
       bcf    R20
       movf  oftime,w
       movwf TMR0
on     goto  isrx
       bsf    SWIT
       bsf    R20
       movf  ontime,w
       movwf TMR0
       goto  isrx

; PWM control routine

pwm    bsf    STATUS,RP0
       bcf    R20
       movlw b'10000010' ;/32 TMR0
       movwf OPTION_REG
       bcf    STATUS,RP0

       movlw 0
       movwf isrvec           ;set interrupt re-vector
       bsf    INTCON,T0IE
       bsf    INTCON,GIE

p2     movf  pwmp,w
       addlw 2
       call  wtod
       movf  pwmp,w
       sublw 0
       movwf ontime
       movf  pwmp,w
       sublw 96
       sublw 0
       movwf oftime
p4     movlw 10
       call  delay
       call  segout
       movlw 10
       call  delay
       call  segout
       decfsz timer,f
       goto  p4
       movlw 25
       movwf timer
p3     movlw 5
       call  delay
       call  segout
       btfss BUT1
       goto  dn
       btfss BUT2
       goto  up
       clrf  timer
       goto  p3

dn     btfss BUT2
       goto  cycle
       decfsz pwmp,f
       decf  pwmp,f
       incf  pwmp,f
       goto  p2

up     btfss BUT1
       goto  cycle

```

```
incf pwmp, f
movf pwmp, w
xorlw 96
btfsc STATUS, Z
decf pwmp, f
goto p2
```

```
ir38 bsf STATUS, RP0
bcf R20
bcf STATUS, RP0
```

```
ir2 call brst
movlw 5
movwf count
```

```
ir4 call segout
movlw 9
call delay
decfsz count, f
goto ir4
call segoff
btfsc BUT1
goto ir2
btfsc BUT2
goto ir2
goto cycle
```

```
brst movlw 38
movwf count
```

```
b2 bsf R20
movlw 20
movwf dela
```

```
b3 decfsz dela, f
goto b3
nop
nop
nop
```

```
bcf R20
movlw 20
movwf dela
```

```
b5 decfsz dela, f
goto b5
nop
decfsz count, f
goto b2
return
```

;------

```
; 20 khz digital noise @20Mhz
```

```
noise bsf STATUS, RP0
bcf R20
bcf STATUS, RP0
movlw 55h
movwf rand
```

```
n2 call pseudo
btfsc rand, 0
bsf R20
btfss rand, 0
bcf R20
call segout
movlw 65
movwf dela
```

```

n4    decfsz    dela, f
      goto    n4
      btfsc   BUT2
      goto    n2
      btfsc   BUT1
      goto    n2
      goto    cycle

```

```

pseudo    movf   rand, w
          addwf  rand+1, w
          movwf  rand+1
          addwf  rand+2, w
          movwf  rand+2
          bcf   STATUS, C    ;do 8 bit rotate
          rrf   rand, f
          btfsc STATUS, C
          bsf   rand, 7
          addwf rand, f
          return

```

; pulse width generation @20Mhz

```

radcon    bsf   STATUS, RP0
          bcf   R20
          bcf   STATUS, RP0
          movlw low 1500
          movwf timer
          movlw high 1500
          movwf timer+1
rc1       movf  timer, w
          movwf acc
          movf  timer+1, w
          movwf acc+1
          clrf  acc+2
          clrf  acc+3
          call  b2bcd
          call  format
rc2       movlw 50
          movwf count
rc3       call  segout
          movlw 10
          call  delay
          decfsz count, f
          goto  rc3
          movf  timer, w
          movwf lo
          movf  timer+1, w
          movwf hi
          call  segoff
          call  rcpl5
          btfss BUT1
          goto  rcdn
          btfss BUT2
          goto  rcup
          goto  rc2
rcup      btfss BUT1
          goto  cycle
rcup2     movlw 10
          addwf timer, f
          btfsc STATUS, C
          incf  timer+1, f

```

```

        movf timer+1,w
        xorlw 9
        btfsc STATUS,Z
        goto rcdn2
        goto rc1

rcdn  btfss BUT2
        goto cycle
rcdn2 movlw -10
        addwf timer,f
        btfss STATUS,C
        decf timer+1,f
        movf timer+1,w
        xorlw 2
        btfsc STATUS,Z
        goto rcup2
        goto rc1

rcpls incf hi,f
        incf lo,f
        bsf R20
rc4   nop
        nop
rc5   decfsz    lo,f
        goto rc4
        decfsz    hi,f
        goto rc5
        bcf R20
        return

; Square wave interrupt handler

sqri  bcf  INTCON,T0IF
        movf freq,w
        addwf timer,f
        movf freq+1,w
        btfsc STATUS,C
        incfsz    freq+1,w
        addwf timer+1,f
        btfss timer+1,7
        bcf R20
        btfsc timer+1,7
        bsf R20
        movlw -73
        addwf TMR0,f
        goto isr

; linear squarewave generation @20Mhz

square    bsf  STATUS,RP0
        bcf  R20
        movlw b'10001000'
        movwf OPTION_REG ;/32 clock
        bcf  STATUS,RP0
        movlw 1
        movwf isrvec
        bsf  INTCON,GIE
        bsf  INTCON,T0IE ;timer-0 interrupt on

sqh1  movf  freq,w
        movwf acc
        movf  freq+1,w

```

```

        movwf acc+1
        clrf  acc+2
        clrf  acc+3
        call  b2bcd
        call  format

sq2    movf  rate,w
        movwf count
sqs1   call  segout
        movlw 10
        call  delay
        decfsz    count,f
        goto  sqs1
        movlw -5
        addwf rate,f
        movf  rate,w
        addlw -5
        btfsc STATUS,C
        goto  sq4
        movlw 6
        movwf rate
sq4    btfss BUT1
        goto  sqdn
        btfss BUT2
        goto  squp
        clrf  rate
        goto  sq2

squp   btfss BUT1
        goto  cycle
        incf  freq,f
        btfsc STATUS,Z
        incf  freq+1,f
        goto  sqh1

sqdn   btfss BUT2
        goto  cycle
        movlw -1
        addwf freq,f
        btfss STATUS,C
        decf  freq+1,f
        btfss freq+1,7
        goto  sqh1
        clrf  freq
        clrf  freq+1
        goto  sqh1

; logic pulser

pulse  bsf   STATUS,RP0
        movlw 6
        movwf ADCON1           ;all digital on ra
        bcf   STATUS,RP0

ppp2   movf  pmode,w           ;get pulse mode
        addlw (vpuls-vbase)/4 ;NOTE: = (vpuls-vbase)/4 in this assembler
        call  setmode
        movf  pmode,w           ;now, get delay factor
        call  dpuls
        movwf timer
        goto  ppp3

ppp5   bsf   STATUS,RP0
        bcf   R20               ;activate pulse

```

```

nop
bsf R20 ;end of pulse
bcf STATUS,RP0
bsf digits,6 ;show pulse is active
goto ppp8

ppp3 bsf R20 ;set latch to low.
      btfsc R470 ;which way to pulse?
      bcf R20 ;then set latch to high.
      movlw 8 ;low level detected?
      btfsc R20
      movlw 1 ;no - show high
      movwf digits ;in 1st display
ppp8 movf timer,w
      call delsq ;delay w/display
      movlw 2
      call delsq ;adjust for hi end error
      btfsc BUT1
      goto ppp4 ;no button 1.
      btfsc BUT2
      goto ppp5 ;but 1 only - do pulses
      goto cycmode ;exit to menu.

ppp4 btfsc BUT2 ;only button 2 ?
      goto ppp3 ;no - set pulse direction
ppp4x movlw 10
      call delay
      call blank
      call segout
      btfss BUT1 ;exit?
      goto cycmode ;yes - and save params
      btfss BUT2
      goto ppp4x
      incf pmode,f
      movf pmode,w
      sublw 3
      btfss STATUS,C
      clrf pmode
      goto ppp2

```

; signal generator (0.5 v square wave)

```

sig bsf STATUS,RP0
     bcf R10K ;engage 10k resistor
     bcf R100K ;engage 100k resistor
     movlw 6
     movwf ADCON1 ;all digital on ra
     bcf STATUS,RP0
     bcf R10K ;10k pull to ground
     bcf R100K ;100k for signal gen
     clrf digits
     movlw 5
     call digseg
     movwf digits+1
     movlw 0
     call digseg
     movwf digits+2
     movwf digits+3

```

```

sis3 btfss timer,6
      bsf digits,6
      bsf R100K

```

```

    call  sis8
sis4  incf  timer,f
      btfsc timer,6
      bcf  digits,6
      bcf  R100K
      call sis8
      btfsc BUT1
      goto sis4
      btfsc BUT2
      goto sis3
      goto cycle

sis8  movlw 32
      movwf count
sis9  call  segout          ;run diplay
      movlw 14
      call short
      decfsz    count,f
      goto  sis9
      return

; do short delays

short movwf dela
shs2  nop
      nop
      nop
      nop
      nop
      decfsz    dela,f
      goto  shs2
      return

; measure capacitor from tip to gound lead

cap   bsf   STATUS,RP0
      bcf   R100K          ;enable 100k resistor
      bcf   STATUS,RP0
      bsf   R100K         ;pull up resitor
      bcf   R20           ;pull down for clamp
      bsf   HOLD          ;default hold condition

cac2  call  segoff
      clrf  acc           ;clear accum
      clrf  acc+1
      clrf  acc+2
      clrf  acc+3
      bsf   STATUS,RP0
      bsf   R20           ;release clamp
      bcf   STATUS,RP0
      nop
      nop                ;compensate
      nop                ;for stray
      nop                ;capacitance
      nop
      nop

cac7  nop                ;tune loop with nop's
      nop
      nop
      nop

```

```

nop
nop
nop
nop
nop
nop
btfsc R20          ;check direct (a/d) input
goto  cahit
incfsz            acc,f
goto  cac7
incfsz            acc+1,f
goto  cac7
incfsz            acc+2,f
goto  cac7
incfsz            acc+3,f
goto  cac7

cahit bsf  STATUS,RP0
      bcf  R20          ;discharge cap
      bcf  STATUS,RP0
      call b2bcd
      call cform        ;format for caps
cac6  call show2
      btfsc BUT1
      goto cac6        ;no button - hold reading
      btfsc BUT2
      goto cac2        ;but1 only - do reading
      goto cycle

; measure inductor from tip to ground lead

induct      bsf  STATUS,RP0
            bcf  R100          ;enable 470 ohm resistor
            bcf  STATUS,RP0
            bcf  R100          ;pull up resistor
            bsf  HOLD          ;default hold condition

inc2  call  segoff
      movlw 80
      call  delay
      clrf  acc          ;clear accum
      clrf  acc+1
      clrf  acc+2
      clrf  acc+3
      bsf  R100          ;start voltage

inc7  btfss R20          ;check direct (a/d) input
      goto  inhit
      incfsz            acc,f
      goto  inc7
      incfsz            acc+1,f
      goto  inc7
      incfsz            acc+2,f
      goto  inc7
      incfsz            acc+3,f
      goto  inc7

inhit bcf  R100
      call b2bcd
      movlw 3
      call xform
      bsf  digits+2,7
inc6  call show2
      btfsc BUT1

```



```

        goto inc6          ;no button - hold reading
        btfsc BUT2
        goto inc2          ;but1 only - do reading
        goto cycle

; turn off all digits
segoff      bcf  PORTC,0
           bcf  PORTC,1
           bcf  PORTC,2
           bcf  PORTC,3
           return

; Logic probe function
; show Low, High, and Float w/ Pulse detection

lp1  bsf  STATUS,RP0
      bcf  R10K          ;turn on voltage feed
      bcf  R10KA         ;turn on other feed
      movlw b'10101000' ;no prescale on TMR0
      movwf OPTION_REG
      movlw 0Eh          ;ana0 on, left justify
      movwf ADCON1
      bcf  STATUS,RP0
      bsf  R10K          ;10k pull up
      bcf  R10KA         ;10k pull down
      movlw 81h          ;read a/d on RA.0
      movwf ADCON0
      clrf  dp           ;no decimal
      movlw 0ffh
      movwf TMR0        ;set for single count overflow

lp1x1 btfsc BUT1
      goto lp1x2
      btfsc BUT2
      goto lp1x2
      goto cycle
lp1x2 call  atod          ;read the input
      movf  hi,w         ;8 bit value
      sublw 40           ;low enough for low?
      btfsc STATUS,C
      goto lp1low
      movf  hi,w
      sublw 180
      btfss STATUS,C
      goto lp1hi
lp1flo movlw 40h          ;'-'
      goto lp1hl
lp1hi  movlw 76h          ;'H'
      goto lp1hl
lp1low movlw 38h          ;'L'
lp1hl  call  letter
      btfss INTCON,T0IF   ;counter overflow?
      goto lp1no
      bcf  INTCON,T0IF
      movlw 0ffh
      movwf TMR0          ;set for next time
      clrf  timer
      movlw 73h          ;'P'
      movwf digits+1     ;show pulse detected
lp1no call  show1
      incf  timer,f
      movf  timer,f

```

```

        btfsc STATUS,Z
        clrf  digits+1    ;always flash off
        goto lpx1

; run display for a short time

show2  clrf  count
        goto  shos2
show1  clrf  count
        bsf   count,7
        goto  shos2
show   clrf  count
        bsf   count,6
shos2  call  segout          ;show digits
        call  flash         ;add dp
        movlw 10
        call  delay
        decfsz count,f
        goto  shos2
        return

; show one character and blank other 3

letter  movwf digits
        goto  blnk2
blank  clrf  digits
blnk2  clrf  digits+1
        clrf  digits+2
        clrf  digits+3
        return

; show 8 bit value on display

wtod  movwf acc
        clrf  acc+1
        clrf  acc+2
        clrf  acc+3
        call  b2bcd
        call  format
        return

; measure diode drops
; same as voltmeter but with 5v feed to probe

diode  bsf   STATUS,RP0
        bcf   R10K          ;enable feed resistor
        goto  vdx          ;otherwise, like voltmeter

; voltmeter function, 0-5 volts

volt  bsf   STATUS,RP0
vdx   movlw 8Eh            ;enable ra.0 analog right justify
        movwf ADCON1
        bcf   STATUS,RP0
        bsf   R10K          ;only used for 'diode' mode.
        movlw 81h          ;select ana0 and enable a/d
        movwf ADCON0

vov2  call  atod           ;get 10 bits
        bcf   STATUS,C
        rrf   hi,f
        rrf   lo,f         ;scale to 9 bits

```

```

    movf  lo,w
    movwf acc
    movf  hi,w
    movwf acc+1
    clrf  acc+2
    clrf  acc+3
    call  b2bcd
    call  format
    bsf   digits+1,7
    movlw 200
    movwf count
vov4   call  segout
    movlw 10
    call  delay
    decfsz    count,f
    goto  vov4
    btfsc BUT1
    goto  vov2
    btfsc BUT2
    goto  vov2
    goto  cycle

; read 16 bit a/d value in hi/low

atod   bsf   ADCON0,2
        nop
        nop
adchk  btfsc ADCON0,2 ;check for conversion complete
        goto adchk
        bsf   STATUS,RP0
        movf  ADRESL,w
        bcf   STATUS,RP0
        movwf lo
        movf  ADRESH,w
        movwf hi
        return

; count events using TMR0 directly

dcount   bsf   STATUS,RP0
        bsf   CLAMP ;disable clamp
        bcf   R10K ;use 10k only
        bsf   R150 ;disconnect other 10k
        bsf   R100K ;no 100k feed
        bsf   R20 ;float direct resistor
        movlw b'10101000' ;no prescale on TMR0
        movwf OPTION_REG
        movlw 6
        movwf ADCON1 ;all ra is digital
        bcf   STATUS,RP0
        bsf   R10K ;use 10k pullup

dcic2   clrf  TMR0 ;clear hardware timer
        clrf  xacc ;clear 32bit count
        clrf  xacc+1
        clrf  xacc+2
        clrf  xacc+3
dcic3   call  xfer
        call  b2bcd
        movlw 3
dcic5   call  xform ;show low 4 digits
        call  segout ;run display

```

```

        call icup            ;update counter
        btfss BUT1         ;button 1 ?
        goto dcic9         ;no - continue
        btfss BUT2
        goto dcic2         ;reset counting.
        goto dcic3         ;continue count.
dcic9  btfss BUT2         ;both buttons?
        goto cycle         ;yes - exit.
        call xfer
        call b2bcd
        movlw 7
        goto dcic5

; update 32bit count from hardware timer

icup  movf TMR0,w          ;hardware counter to accum
      movwf xacc
      btfss INTCON,T0IF ;overflow
      return              ;not yet.
      bcf INTCON,T0IF
      incfsz xacc+1,f
      return
      incfsz xacc+2,f
      return
      incf xacc+3,f
      return

; move aux count to accum

xfer  movf xacc,w
      movwf acc
      movf xacc+1,w
      movwf acc+1
      movf xacc+2,w
      movwf acc+2
      movf xacc+3,w
      movwf acc+3
      return

; measure frequency using TMR0 with 8-bit prescale

fcount  bsf STATUS,RP0
        bsf CLAMP
        bsf R10K          ;no extra voltage
        bsf R150         ;disconnect other 10k
        bcf R100K        ;use 100k pulldown
        bsf R20          ;float direct resistor
        movlw b'10100111' ;256 count prescaler
        movwf OPTION_REG
        movlw 6
        movwf ADCON1     ;all ra is digital
        bcf STATUS,RP0
        bcf R150         ;pull down lightly
        clrf acc
        clrf acc+1
        clrf acc+2
        clrf acc+3
fcf2  call b2bcd
      call format
      btfsc BUT1
      goto fcf4

```

```

fcf7  btfss BUT2          ;both buttons?
      goto  cycle
fcfx  movlw 3
      call xform
fcf4  call cnton
      call onesec
      call cntoff
      goto fcf2

; clear 32bit count and turn on TMR0

cnton bcf  CLAMP
      clrf TMR0          ;clear timer and prescale
      clrf acc
      clrf acc+1
      clrf acc+2
      clrf acc+3
      bsf  STATUS,RP0
      bsf  CLAMP        ;tristate the clamp
      bcf  STATUS,RP0
      return

; turn off counter and pulse input to read out prescale

cntoff bcf  CLAMP
      bsf  STATUS,RP0
      bcf  CLAMP
      bcf  STATUS,RP0
      movf TMR0,w
      movwf acc+1
      clrf acc
cnc2  decf  acc,f
      bsf  CLAMP
      bcf  CLAMP
      movf TMR0,w
      xorwf acc+1,w
      btfsc STATUS,Z
      goto cnc2
      return

;-----
; add decimal point to display

flash swapf dp,w
      andlw 3           ;bits 5,4 = digit nr.
      addlw digits
      movwf FSR
      btfsc dp,7       ;decimal used?
      bsf  0,7         ;light decimal point
      incf ftimer,f
      btfsc STATUS,Z
      incf ftimer+1,f
      btfss ftimer+1,5
      goto fno
      btfsc dp,6       ;flash mode?
      bcf  0,7         ;flash back off
      return
fno   nop
      return

```

```
; open count gate for exactly 1.000000 second
```

```
onesec      movlw low LPTIME
            movwf timer
            movlw high LPTIME
            movwf timer+1
osx1        nop                ;100 cycle loop count
            nop
osx2        call ffill          ;time filler
            call segout         ;show last data
            call flash          ;include decimal
            call dotime         ;check for TMR0 overflow
            decfsz timer,f
            goto osx1
            decfsz timer+1,f
            goto osx2
            movlw 8
            movwf timer
osx5        call dotime
            decfsz timer,f
            goto osx5
            return
```

```
; just kill some time
```

```
ffill      movlw 16
            movwf temp
fif2       decfsz temp,f
            goto fif2
            nop
            nop
            return
```

```
; check for TMR0 rollover
; (constant execution times)
```

```
dotime     btfss INTCON,T0IF ;overflow?
            goto dno
            bcf INTCON,T0IF ;clear the flag
            incf acc+2,f      ;increment next byte
            btfsc STATUS,Z
            incf acc+3,f      ;and next on rollover
            return
dno        nop
            nop
            nop
            return
```

```
; squared delay function
```

```
delsq      movwf dela+1
            movwf dela+2
dlsd1      movf dela+2,w
            movwf dela
dlsd0      call segout
            decfsz dela,f
            goto dlsd0
            decfsz dela+1,f
            goto dlsd1
            return
```

```

; cubic delay function

delay movwf dela+2
dld2 movwf dela+1
dld1 movwf dela
dld0 decfsz    dela, f
      goto dld0
      decfsz    dela+1, f
      goto dld1
      decfsz    dela+2, f
      goto dld2
      return

; format used for capacitor measurements
; each count = 100pf

cform call first
      movf count, w          ;1st non zero
      call dpcap
      movwf dp
      andlw 0fh             ;where to start
      movwf count
      goto ff5

; ***** video pattern generation *****

vgen  bsf  STATUS, RP0
      bcf  BLACK
      bcf  WHITE
      bcf  R100
      bcf  STATUS, RP0
      bcf  R100
      bcf  PORTC, 2
      bcf  PORTC, 3
      movlw 0BFh
      movwf PORTB

vgv2  call vsync
      movlw 40
      movwf count

vc1   nop
      call bline
      decfsz    count, f
      goto vc1
      movlw 15
      movwf count

vc2   call dline
      call dline4
      call dline4
      call bline4
      call bline4
      call bline4
      call bline4
      call bline4
      call bline4
      call bline4
      call bline4
      call bline4
      call bline4
      call bline4
      call bline4
      nop
      decfsz    count, f
      goto vc2
      movlw 17

```

```

vc3    movwf count
      call bline
      nop
      decfsz     count, f
      goto vc3
      call zline4
      btfsc BUT2
      goto vgv2
      goto cycle

vsync  movlw 6
      movwf count
vv2    bcf  BLACK
      call dla10
      bsf  BLACK
      movlw 46
      call vdel1a
      decfsz     count, f
      goto vv2

      movlw 6
      movwf count
vv3    bcf  BLACK
      movlw 46
      call vdel1a
      bsf  BLACK
      call dla8
      decfsz     count, f
      goto vv3

      movlw 6
      movwf count
vv4    bcf  BLACK
      call dla10
      bsf  BLACK
      movlw 46
      call vdel1a
      decfsz     count, f
      goto vv4
      return

zline4    call dla4
zline    bcf  BLACK      ;start h-sync
      call dla16
      call dla7
      bsf  BLACK
      call vseg
      movlw 89
      call vdel1a
      nop
      return

bline4    call dla4
bline    bcf  BLACK      ;start h-sync
      call dla16
      call dla7
      bsf  BLACK
      movlw 93
      call vdel1a
      nop
      return

```



```

dline4    call   dla4
dline bcf   BLACK           ;start h-sync
        call   dla16
        call   dla7
        bsf   BLACK
        call   dla18
        call   dla18
        movlw 19
        movwf temp
bd2      nop
        nop
        nop
        nop
        nop
        nop
        bsf   WHITE
        nop
        bcf   WHITE
        decfsz temp,f
        goto  bd2
        movlw 5
        call  vdel1
        return

; delay in processor cycles (includes call/return)

```

```

dla18 nop
dla17 nop
dla16 nop
dla15 nop
dla14 nop
dla13 nop
dla12 nop
dla11 nop
dla10 nop
dla9  nop
dla8  nop
dla7  nop
dla6  nop
dla5  nop
dla4  return

```

```

vdel1 movwf temp
vd2   decfsz temp,f
      goto  vd2
      return

```

```

; light alternate digits (12 ct)

```

```

vseg  incf  frame,f           ;count frames
      bcf  PORTC,0
      bcf  PORTC,1
      btfss frame,0
      bsf  PORTC,0
      btfsc frame,0
      bsf  PORTC,1
      nop
      return

```

```

; ****      end video gen *****

```

```
; turn off displays
```

```
clear movlw 0F0h  
andwf PORTC,f  
return
```

```
; cycle thru individual segment drives on 4 digits  
; (constant execution time: 30 instructions)  
; ***** Do Not Modify *****
```

```
segout    movlw 0F0h  
          andwf PORTC,f           ;clear low 4 bits  
          movf  dignr,w           ;get digit nr  
          addlw digits           ;base of segment list  
          movwf FSR  
          movf  0,w              ;get the data  
          andwf segmask,w        ;mask the bit  
          xorlw 0ffh             ;invert  
          movwf PORTB           ;one possible segment line low  
          movf  dignr,w  
          call  getano  
          iorwf PORTC,f          ; set one high.  
          bcf  STATUS,C  
          rlf  segmask,f  
          btfsc STATUS,C  
          goto nseg  
          nop  
          nop  
          nop  
          nop  
          nop  
          return  
nseg     bsf  segmask,0          ;rotate  
          incf dignr,f  
          btfsc dignr,2          ;overflow?  
          clrf dignr  
bak      return
```

```
; display digits from specified position
```

```
xform movwf count  
      goto ff5
```

```
; format first 4 signifcant digits on display
```

```
format    call  first           ;find 1st non-zero digit  
ffx      movf  count,w  
          call  getdp           ;get formating info  
          movwf dp  
          andlw 0fh  
          movwf count  
ff5      call  ff9              ;get segment pattern  
          movwf digits  
          call  ff9  
          movwf digits+1  
          call  ff9  
          movwf digits+2  
          call  ff9  
          movwf digits+3  
          return
```

```

ff9  movf  count,w
     decf  count,f
     call  getbd      ;get bcd value
     call  digseg
     return

```

```

; set count to 1st non zero digit

```

```

first movlw 9          ;start w/last digit
      movwf count
ff2   movf  count,w      ;get next digit
      call  getbd
      xorlw 0           ;set z flag
      btfss STATUS,Z
      return           ;hit non zero
      decfsz count,f
      goto  ff2
      return

```

```

; get bcd digit specified by 'w'

```

```

getbd movwf temp      ;save digit nr
      clrw
      btfsc temp,7     ;negative value?
      return          ;zero if negative.
      bcf  STATUS,C
      rrf  temp,f      ;find buffer offset
      movf temp,w
      rlf  temp,f      ;restore all bits
      addlw bcd        ;add start of bcd buff
      movwf FSR        ;set pointer
      movf 0,w         ;get the byte
      movwf ddata      ;and store it.
      btfsc temp,0     ;low or hi nibble?
      swapf ddata,f    ;hi.
      movf ddata,w
      andlw 0fh
      return

```

```

; Convert 32-bit binary number at <acc> into a bcd number
; at <bcd>. Uses Mike Keitz's procedure for handling bcd
; adjust; Modified Microchip AN526 for 32-bits.

```

```

b2bcd movlw 32        ; 32-bits
      movwf ii        ; make cycle counter
      clrfsz bcd      ; clear result area
      clrf  bcd+1
      clrf  bcd+2
      clrf  bcd+3
      clrf  bcd+4

```

```

b2bcd2 movlw bcd      ; make pointer
      movwf FSR
      movlw 5
      movwf cnt

```

```

; Mike's routine:

```

```

b2bcd3 movlw 33h
      addwf 0,f      ; add to both nybbles

```

```

    btfsc 0,3          ; test if low result > 7
    andlw 0f0h        ; low result >7 so take the 3 out
    btfsc 0,7          ; test if high result > 7
    andlw 0fh         ; high result > 7 so ok
    subwf 0,f         ; any results <= 7, subtract back
    incf FSR,f        ; point to next
    decfsz          cnt,f
    goto b2bcd3

```

```

    rlf  acc+0,f      ; get another bit
    rlf  acc+1,f
    rlf  acc+2,f
    rlf  acc+3,f
    rlf  bcd+0,f      ; put it into bcd
    rlf  bcd+1,f
    rlf  bcd+2,f
    rlf  bcd+3,f
    rlf  bcd+4,f
    decfsz          ii,f ; all done?
    goto b2bcd2      ; no, loop
    return

```

; read data 'w' at address hi/low

```

eeeread    bcf  STATUS,RP0
           bsf  STATUS,RP1      ;bank-2
           movf lo,w
           movwf EEADR
           movf hi,w
           movwf EEADRH
           bsf  STATUS,RP0      ;bank-3
           bcf  EECON1,EEPGD    ;access data memory
           bsf  EECON1,RD       ;start the read
           bcf  STATUS,RP0      ;bank-2
           movf EEDATA,w
           bcf  STATUS,RP1
           bcf  STATUS,RP0
           return

```

; write data 'w' at address hi/low

```

eeewrite   bcf  STATUS,RP0
           bsf  STATUS,RP1      ;bank-2
           movwf EEDATA        ;set data
           movf lo,w
           movwf EEADR         ;set address
           movf hi,w
           movwf EEADRH
           bsf  STATUS,RP0      ;bank-3
           bcf  EECON1,EEPGD    ;access data memory
           bsf  EECON1,WREN     ;start write operation
           movlw 55h
           movwf EECON2
           movlw 0AAh
           movwf EECON2
           bsf  EECON1,WR
           nop
           nop
ee2        btfsc EECON1,WR      ;wait for complete
           goto ee2
           bcf  STATUS,RP0
           bcf  STATUS,RP1
           return

```

```

s9600 call clear
      bsf STATUS,RP0
      bsf SOUT ;set to input
      bcf STATUS,RP0
      movf baud,w
      movwf temp
      bcf STATUS,C
      rlf temp,f
      call srq ;get low byte
      movwf qtime
      call srq
      movwf qtime+1
      movlw (vbaud-vbase)/4
      addwf baud,w
      call setmode

ss5 call segout
    btfss BUT2
    goto ss8
    btfsc BUT1
    goto ss5
    btfss BUT2
    goto cycle
    call setdir
    movlw 'A'
    movwf sdata
    movlw 26
    movwf scout

ss2 movf sdata,w
    call serout
    incf sdata,f
    movlw 43
    call delsq
    decfsz scout,f
    goto ss2

    movlw 13
    call serout
    movlw 100
    call delsq
    movlw 10
    call serout
    movlw 220
    call delsq
    btfss BUT1 ;still held down?
    goto ss5 ;repeat operation.
    goto s9600

ss8 call clear ;blank display
    movlw 20
    call delay
    btfss BUT1
    goto cycle
    btfss BUT2
    goto ss8
    incf baud,f
    bcf baud,2 ;limit 0-3
    bsf SAVE
    goto s9600

midi movf midic,w

```

```

        addlw 1          ;convert to midi number
        call wtod
        movlw 39h       ;'C'
        movwf digits
        movlw 76h       ;'H'
        movwf digits+1
        bsf STATUS,RP0
        bsf SOUT        ;set to input
        bcf STATUS,RP0
ms5    call segout
        btfss BUT2
        goto madj
        btfsc BUT1
        goto ms5
        call setdir
        movlw 45        ;31250 baud
        movwf qtime
        clrf qtime+1
        movf midic,w    ;get midi chan nr.
        iorlw 90h       ;add note on command.
        call serout     ;note on
        movlw 60
        call serout     ;middle C
        movlw 40
        call serout     ;velocity
ms6    movlw 20
        call delsq
        btfss BUT1     ;wait for button release
        goto ms6
        movf midic,w
        iorlw 90h
        call serout
        movlw 60
        call serout
        movlw 0
        call serout     ;send note off
        call putopt    ;maybe save channel
        movlw 250
        call delsq
        goto midi
madj   movlw 100
        call delsq
        btfss BUT1
        goto cycle
        btfss BUT2
        goto madj
        bsf SAVE       ;flag for saving new channel
        incf midic,f   ;next midi channel
        movlw 0fh
        andwf midic,f  ;roll over at 15
        goto midi

; check for current state

setdir  bcf DIR        ;check for resting state
        btfss SOUT
        bsf DIR
        bsf STATUS,RP0
        bcf SOUT      ;set to output
        bcf STATUS,RP0
        return

```

```

; send serial data from outdat

serout      clrfs count      ;bit count
            bsf   count,3    ;count=8
            movwf ddata
            call  zero        ;start bit
sers2 call  full
            btfsc ddata,0
            call  one
            btfss ddata,0
            call  zero
            rrf   ddata,f
            decfsz count,f
            goto  sers2
            call  full
            call  one
            call  full
            call  full
            call  full
            return

zero btfss DIR      ;which way is up?
     bcf  SOUT
     btfsc DIR
     bsf  SOUT
     return

one  btfss DIR      ;which way is up?
     bsf  SOUT
     btfsc DIR
     bcf  SOUT
     return

full movf  qtime+1,w
     movwf temp+1
     movf  qtime,w
     movwf temp
     incf  temp+1,f
     incf  temp,f
ff3  decfsz temp,f
     goto  ff3
     decfsz temp+1,f
     goto  ff3
     return

      end

```