

;***** VARIABLE DEFINITIONS

```

#define LED      GPIO,0      ;OUT:    GREEN LED
#define BTN      GPIO,1      ;IN:     3 button array
#define PWM      GPIO,2      ;OUT:    control current to coil
#define SDATA    GPIO,3      ;IN:     MAX6675 serial data (MCLR)
#define CS       GPIO,4      ;OUT:    MAX6675 chip select
#define SCLK     GPIO,5      ;OUT:    MAX6675 serial clock

#define CARRY    STATUS,C    ;carry bit
#define ZERO     STATUS,Z    ;zero bit

#define cADCset   b'00010010' ;Fosc/8 = b'-001----', AN1= b'-----1-'
#define cADCctrl  b'10000101' ;right justify, Vdd, AN1, Enabled

#define PWM_Active_HI b'00001100' ;110x PWM mode active hi

; 97.119 msec = 16984 = 0x4258
#define cHeartbeatLO 0x58 ; TMR1L
#define cHeartbeatHI 0x42 ;16984 TMR1H = 97.119 ms

;7.5 min = 4500 ticks = 0x1194
#define cRUNTIMElo 0x94
#define cRUNTIMEhi 0x11

;3.5 min = 2100 ticks = 0x0834
#define cEXTRATIMElo 0x34
#define cEXTRATIMEhi 0x08

; 1 min = 600 ticks = 0x0258
;30 sec = 300 ticks = 0x012C
;15 sec = 150 ticks = 0x0096
; 1 sec = 10 ticks = 0x000A
#define cTOKEITlo 0x0A
#define cTOKEITHi 0x00

#define cBUTTONinc 0x13 ;18.35 = 5°F * 3.67
#define cBUTTONdec 0x0F ;14.68 = 4°F * 3.67

#define cBTNdelay 0x0A ;10. = 1 second

;===== Temperature Control Parameters =====
;MAX6675 value = degrees F * 4.02 (M14V2U3)
;MAX6675 value = degrees F * 3.67 (M14V2U4)

#define cSPMINlo 0xDE ;SETPOINTmin
#define cSPMINhi 0x02 ;= 734. / 3.67 = 200°F

#define cSPMAXlo 0x2B ;SETPOINTmax
#define cSPMAXhi 0x07 ;= 1835. / 3.67 = 500°F

#define cCTMAXlo 0xE2 ;COILTEMPmax
#define cCTMAXhi 0x07 ;=2018. / 3.67 = 550°F

#define cCHANGEmax 0xC0 ;192/3.67 = 52°F (per sample)

```

```

;===== Duty Cycle Control Parameters =====
;duty cycle is 0 to 1023 (0% to 100%) b'11 1111 1111' = 0x03FF
#define CDCMAXlo      0xFF      ;maximum duty cycle value
#define CDCMAXhi      0x03

#define aERRORmax     0x40      ;17°F above - go DCzero
#define bERRORmax     0x40      ;17°F below - go DCmax

;vAbove:      gains - decrement if greater
#define aPgain       .6        ;(P)ID dec
#define aIgain       .2        ;P(I)D dec
#define aDgainInc     .8        ;PI(D) dec
#define aDgainDec     .4        ;PI(D) inc

;vBelow:      gains - increment if greater
#define bPgain       .3        ;(P)ID inc
#define bIgain       .2        ;P(I)D inc
#define bDgainInc     .8        ;PI(D) inc
#define bDgainDec     .4        ;PI(D) dec

#define CDCBIASminlo  0x20      ;minimum bias value
#define CDCBIASminhi  0x00

#define CDCBIASmaxlo  0x80      ;maximum bias value
#define CDCBIASmaxhi  0x00

;===== Minimum BUTTON Values =====
;Btn1:  +5  2/(1+2)    2/3 * 1024 = 682    = $02AA
;Btn2:  -4  2/(1+1+2)  1/2 * 1024 = 512    = $0200

#define cBTN1minlo    0x6C      ;0C =620. = 2/3 * 1023 - 10%
#define cBTN1minhi    0x02      ;0D
#define cBTN2minlo    0xD1      ;0E =465. = 1/2 * 1023 - 10%
#define cBTN2minhi    0x01      ;0F

;=====

#define cLEDONTicks    0x02
#define cLEDOFFticks   0x02
#define cLEDONerror    0x02      ;1/2 degree F (3.67 = 1°F)

#define cWDT5min_pre   b'00010111' ;1:65536 (WDTCON)
#define cWDT5min_post  b'00001111' ;1:128 (OPTION_REG)
; #define cWDT5min_post  b'00000011' ;1:32 (OPTION_REG)
#define cWDT256ms_post b'00001100'

;*****
;***** data registers:  $20 .. $2F
;*****

ORG 0x0020

;----- Coil Temperature buffer ----- 8 values
;store lo-byte, followed by hi-byte
COILTEMP      EQU 0x20      ;.. 0x2F
constant vNoSensor=2        ;bit set if sensor missing
constant cCOILTEMPstart=0x20
constant cCOILTEMPend=0x2F

```

```

constant cCOILTEMPbytes=0x10

;*****
;***** data registers:  $30 .. $6F
;*****

;----- temperature vars -----
ORG 0x0030
SETPOINT      EQU 0x30      ;0x30..0x31 temperature set point
;SETPOINT+1    EQU 0x31
constant cSETPOINTadr=0x30
SPERROR       EQU 0x32      ;lo(setpoint error)
constant cSPERRORadr=0x32
;SPERROR+1     EQU 0x33      ;hi(setpoint error)
SPERRORsum    EQU 0x34      ;total SPERROR since last SETPOINT
;SPERRORsum+1  EQU 0x35
SPERROR2      EQU 0x36      ;previous SPERROR
;SPERROR2+1    EQU 0x37      ;previous SPERROR
SPERRORchg    EQU 0x38      ;SPERROR - <previous SPERROR>
SPERRORchg2   EQU 0x39      ;previous SPERRORchg
CHANGEmax     EQU 0x3A      ;max SPERRORchg with DCMAX
SPERRORtrip   EQU 0x3B
;SPERRORtrip+1 EQU 0x3C
constant cSPERRORtripadr=0x3B

;monitor SPERROR swing for 5 seconds
SPERRORcnt    EQU 0x3D      ;count down to reset SPERRORabove, SPERRORbelow
constant cSPERRORcnt=0x0F    ;=15 readings =5 seconds (3 per sec)
SPERRORabove  EQU 0x3E      ;largest SPERROR above SETPOINT
SPERRORbelow  EQU 0x3F      ;largest SPERROR below SETPOINT

;----- duty cycle vars -----
ORG 0x0040
DUTYCYCLEsp   EQU 0x40
;DUTYCYCLEsp+1 EQU 0x41
constant cDUTYCYCLEsp=0x40
DUTYCYCLEnew  EQU 0x42
;DUTYCYCLEnew+1 EQU 0x43
constant cDUTYCYCLEnew=0x42
DCBIAS        EQU 0x44      ;neutral duty cycle (lo)
;DCBIAS+1      EQU 0x45      ;neutral duty cycle (hi)
constant cDCBIASadr=0x44
BTNVALUE      EQU 0x46      ;lo value from AtoD reading
;BTNVALUE+1    EQU 0x47      ;hi value from AtoD reading
constant cBTNVALUEadr=0x46
;free         EQU 0x48
;free         EQU 0x49
;free         EQU 0x4A
;free         EQU 0x4B
;free         EQU 0x4C
;free         EQU 0x4D
;free         EQU 0x4E
;free         EQU 0x4F

;----- counters -----
ORG 0x0050

```

```

RUNTIME            EQU 0x50      ;ticks until shutdown
;RUNTIME+1         EQU 0x51
FLASHdelay0        EQU 0x52      ;delay between flash patterns
FLASHcount0        EQU 0x53      ;number of flashes in pattern
LEDcnt             EQU 0x54      ;ticks remaining for this LED state
FLASHcnt           EQU 0x55      ;number of LED on/off sequences
TOKEITdelay        EQU 0x56      ;count 30 seconds of ~vBelow
;TOKEITdelay+1     EQU 0x57
BTNcount           EQU 0x58      ;count ticks same button held
;free              EQU 0x59
;free              EQU 0x5A
;free              EQU 0x5B
;free              EQU 0x5C
;free              EQU 0x5D
TEMPVAR            EQU 0x5E
;TEMPVAR+1         EQU 0x5F

;----- eePROM read/write -----
    ORG 0x0060
eeCOILTEMPptr      EQU 0x60      ;pointer to buffer of MAX6675 reading
eeDUTYCYCLEptr     EQU 0x61      ;pointer to buffer of DUTYCYCLEsp values
eeFrom             EQU 0x62
eeTo               EQU 0x63
eeCnt              EQU 0x64
eeByte             EQU 0x65
;free              EQU 0x66..0x6F

;*****
;0x70..0x7F (0xF0..0xFF) mapped in both banks
;*****
    ORG 0x0070
vSTATE             EQU 0x70      ;vaporizer state
    constant      cvSTATEadr=0x70
    constant      vTick=0        ;set on Heartbeat interrupt
    constant      vWait=1        ;wait for BTN idle
    constant      vRun=2
    constant      vLEDact=3      ;LED is ACTIVE (1)
    constant      vLEDon=4      ;LED is ON (set) / OFF (clear)
;    constant      *****=5    ;free
;    constant      *****=6    ;free
    constant      vError=7

vSENSOR            EQU 0x71      ;sensor state
    constant      vBtnON=0       ;Button press active
    constant      vBtn1=1        ;Button1 = +5, RESET
    constant      vBtn2=2        ;Button2 = -4, START
;    constant      *****=3    ;free
;    constant      *****=4    ;free
;    constant      *****=5    ;free
    constant      vBtn=6         ;Button (AtoD) reading ready
    constant      vTypeK=7       ;K-Thermocouple reading ready

vCOILTEMP          EQU 0x72      ;coil temperature state
    constant      vAbove=0       ;COILTEMP is above SETPOINT
    constant      vBelow=1       ;COILTEMP is below SETPOINT
    constant      vInc=2         ;positive slope = temp increasing

```

```

constant vDec=3           ;negative slope = temp decreasing
constant vAccel=4         ;temp change increasing
constant vDecel=5         ;temp change decreasing
constant vChgMax=6        ;COILTEMPchg at max value
; constant *****=7      ;free

vPWM EQU 0x73             ;pulse width modulator state
constant vPWMon=0         ;PWM is active
constant vDCmax=1         ;Duty Cycle is max allowed value
constant vDCzero=2        ;Duty Cycle at zero
constant vDCsp=3          ;Duty Cycle at setpoint value
; constant *****=4      ;free
; constant *****=5      ;free
constant vDCsign=6        ;Duty Cycle adjustment flag
constant vDCnew=7         ;new Duty Cycle value ready

vERROR EQU 0x74
constant cvERRORadr=0x74
constant vWDT=0           ;WatchDog timer expired
constant vZeroR=1         ;zero reading from MAX
constant vTooHot=2        ;maximum temperature exceeded
constant vRuntime=3       ;maximum runtime exceeded
constant vBattery=4       ;battery dead
; constant *****=5      ;free
; constant *****=6      ;free
; constant vTypeK=7       ;K-Thermocouple missing

vCOILTEMPold EQU 0x75     ;previous COILTEMP status
vSENSORold EQU 0x76       ;previous tick sensor reading
COILTEMPptr EQU 0x77      ;pointer to most recent coil temperature value
;free EQU 0x78
;free EQU 0x79
LOOPcnt EQU 0x7A          ;loop counter (ReadCOILTEMP)
BTNdelay EQU 0x7B         ;20 ticks (2 seconds)
FLASHdelay EQU 0x7C       ;ticks until LED flashes start
COILTEMPdelay EQU 0x7D    ;ticks until next temp reading (0..3)

STATUS_save EQU 0x7E      ;push STATUS
W_save EQU 0x7F           ;push W

;*****
;***** data registers: $A0 .. $BF
;*****
; ORG 0x00A0
; ORG 0x00B0

;*****
;***** EEPROM memory: $00 .. $FF
;*****
ORG 0x2100
;reserve first 16 bytes

;m14v2unit3:
;0x060C = 1548. == 385F == 4.02 (measured)
;temp(F) * 4.02 = MAXunits
;400F = 0x0648

```

```

;m14v2unit4:
;0x0648 = 1608. == 438°F == 3.67 (measured)
;temp(°F) * 3.67 = MAXunits
;400°F = 1468. = 0x05BC
;380°F = 1468. = 0x05BC 3.86

;m14v3unit1:

;0x05A0 / 0x0048
;0x05C0 / 0x004D

eeSETPOINT      de  0xEA      ;00: (lo) desired 12-bit value from sensor
                 de  0x05      ;01: (hi) 395°F
                 constant ceeSETPOINT=0x00

eeDCBIAS         de  0x48      ;02: (lo) DCBIAS
                 de  0x00      ;03: (hi)
                 constant ceeDCBIASadr=0x02

eeSPERRORtrip    de  0x24      ;04: SPERROR < 80°F
                 de  0x01      ;05
                 constant ceeSPERRORtripadr=0x04

                 de  0x00, 0x00 ;06, 07
                 de  0x00, 0x00 ;08, 09
                 de  0x00, 0x00 ;0A, 0B
                 de  0x00, 0x00 ;0C, 0D
                 de  0x00, 0xFF ;0E, 0F
                 constant ceevERRORadr=0x0F

;DEBUG***** block of loop counters
;eeCounter1      de  0x00      ;0C loop counter
;  constant ceeCounter1=0x0C
;eeCounter2      de  0x00      ;0D loop counter
;  constant ceeCounter2=0x0D
;eeCounter3      de  0x00      ;0E loop counter
;  constant ceeCounter3=0x0E
;eeCounter4      de  0x00      ;0F loop counter
;  constant ceeCounter4=0x0F

                ORG 0x2110
;112 bytes = 56 temperature readings
;@ 3 readings per second = 18 seconds + 2 readings
eeCOILTEMP      ;0x10..0x7F (112 bytes)
                 de  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
                 de  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
                 de  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
                 de  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
                 de  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
                 de  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
                 de  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
                 constant eeCOILTEMPfirst=0x10
                 constant eeCOILTEMPwrap=b'10000000'      ;wrap if bit non-zero
                 constant eeCOILTEMPlast=0x7F
                 constant eeCTBUFbytes=0x6F

```

```

    ORG 0x2180
;sync with eeCOILTEMP buffer

;factory settings - reset via button sequence:
;   BTN1 (2 seconds) only before start

    constant ceeFACTORYbytes=0x10           ;= 16.

eeFACTORYsp    de  0xC0      ;80
               de  0x05      ;81      395°F
    constant ceeFACTORYsp=0x80

eeFACTORYbias  de  0x50      ;82
               de  0x00      ;83
    constant ceeFACTORYbias=0x82

eeFACTORYtrip  de  0x24      ;84
               de  0x01      ;85
    constant ceeFACTORYtrip=0x84

               de  0x00, 0x00 ;06, 07
               de  0x00, 0x00 ;08, 09
               de  0x00, 0x00 ;0A, 0B
               de  0x00, 0x00 ;0C, 0D
               de  0x00, 0x00 ;0E, 0F

    ORG 0x2190
;112 bytes = 56 DUTYCYCLE values
eeDUTYCYCLE ;0x90..0xFF (112 bytes)
    de  0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
    de  0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
    de  0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
    de  0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
    de  0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
    de  0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
    de  0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
    constant eeDUTYCYCLEfirst=0x90
    constant eeDUTYCYCLEwrap=b'10000000' ;wrap if bit zero
    constant eeDUTYCYCLElast=0xFF
    constant eeDCBUFbytes=0x6F

;end      *****

```