```
; filename: lm35_pwm_cc.asm

; This is fan controller with LM35 input and PWM output. Two controls provide threshold (TH)
; and slope (SL).
; PWM=(T-T(th))*SL with SL in 0.5 unit steps from 0.5-4.5 (slope multiplier).
; Difference is always >= 0, PWM>0x0F (15%)
; This is a version for common cathode displays

; 250524 v0.1   Start project
; 250531 v0.2   A/D
; 250531 v0.3   A/D auto switchover
; 250531 v0.4   PWM algorithm
; 250601 v0.5   Powerup max PWM and ignore adjustments for 10 s
; 250707 v0.6   Adjust with buttons (needs EEPROM save)
; 250711 v0.7   2x AVG on LM value
; 250712 v0.8   Show HEX PWM value when both buttons pressed (DEBUG mode)
; 250925 v0.9   Swap LED_L & R + switches because of PCB fuckup
; 251007 v1.0   Release
; 251128 v1.1   Sanity check of TH and SL values, silent restore from EEPROM in case of glitches,
                repeat TH button

; GPL Copyleft 2025 pic@polonai.se

        LIST    P=16F818, F=INHX8M
        #include <p16f818.inc>

        __CONFIG        _WDT_ON & _PWRTE_OFF & _INTRC_IO & _MCLR_ON & _BODEN_OFF & _LVP_OFF &
_CPD_OFF & _WRT_ENABLE_OFF & _DEBUG_OFF & _CCP1_RB3 & _CP_OFF
        ERRORLEVEL -302                 ; sod message about using proper bank

; Equates
RESET_V         EQU     0x00            ; Address of RESET Vector

; Registers
TEMP1           EQU     0x20            ; Temp register
TEMP2           EQU     0x21            ; Temp register
MSB             EQU     0x22            ; High nibble of display value
LSB             EQU     0x23            ; Low nibble of display value
CNTR            EQU     0x24            ; Lamp Test/Show Preset Counter
SKIP            EQU     0x25            ; Counter for skipping ADC to prevent jittery display
REPT            EQU     0x26            ; Counter for repeat action of threshold
LM              EQU     0x27            ; 2x AVG value from LM35 temperature sensor
LM_NEW          EQU     0x28            ; New value from LM35 temperature sensor
LM_PREV         EQU     0x29            ; Previous value from LM35 temperature sensor
TH              EQU     0x2A            ; Value for Threshold
SL              EQU     0x2B            ; Value for Slope
UPDATE          EQU     0x2C            ; Display Update Counter
EEBUF           EQU     0x2D            ; Byte to write into EEPROM
BIN             EQU     0x2E            ; For BIN2BCD routine
hundreds        EQU     0x2F            ; For BIN2BCD routine
SL_MULT         EQU     0x30            ; Slope multiplier
DIFF            EQU     0x31            ; Difference between actual and current temp (LM)
DIFF_MULT       EQU     0x32            ; DIFF Multiplicand
d1              EQU     0x33            ; delay register 1
d2              EQU     0x34            ; delay register 2
FLAGS           EQU     0x7F            ; General Purpose Flags

; PORTA,0 A/D input for LM35 sensor has no name defined
; PORTB,3 (RB3/CCP1) PWM output has no name defined

#define LED_L   PORTA,6                 ; Left LED display CC (tens)
#define LED_R   PORTA,7                 ; Right LED display CC (units)
#define DP      PORTA,1                 ; Decimal point of slope display
#define UP      PORTA,3                 ; Up button aka Slope button
#define DN      PORTA,4                 ; Down button aka Threshold button
#define LAMPTST FLAGS,0                 ; Lamp Test at poweron
#define SHOWTH  FLAGS,1                 ; Show threshold
#define SHOWSL  FLAGS,2                 ; Show slope
#define SHOWPCM FLAGS,3                 ; Debug PCM HEX value
#define HELDTH  FLAGS,4                 ; Threshold (DN) Button held, wait for release
#define HELDSL  FLAGS,5                 ; Slope (UP) Button held, wait for release
#define POWERUP FLAGS,6                 ; Startup delay to prevent showing SL or TH
```

```
; EEPROM Data
        org     0x2100
        DE      0x21                    ; TH 33
        DE      0x02                    ; SL 1.0


        org     0x00
        GOTO    START
        org     0x04

; Interrupts
        BTFSC   INTCON,TMR0IF           ; Every 4.1 ms
        GOTO    Multiplex               ; Update LED display
        CLRF    PIR1
        RETFIE                          ; If none of the above is true, then just leave the
                                        ; interrupt handler (shouldn't occur)

Multiplex                               ; Display MSB or LSB generated by BIN2BCD routine
        CLRWDT                          ; Clear WDT
        BCF     INTCON,TMR0IF           ; Clear Interrupt source
        BTFSC   LAMPTST                 ; Lamp Test ongoing?
        GOTO    LAMPTEST                ; Yes

; Get the data which we want displayed
; Test Digit
        BTFSS   LED_L                   ; Leftmost digit?
        GOTO    HANDLER_L               ; Yes

; We're at the right digit (LED_R)
        BSF     LED_R                   ; Switch off right digit (CC)
        MOVFW   MSB                     ; Get value to be displayed on left digit
        ANDLW   0x0F                    ; Make sure the lookup table doesn't overflow
        CALL    bin2seg
        MOVWF   PORTB
        BCF     DP
        BTFSC   SHOWSL                  ; In SL mode?
        BSF     DP                      ; Switch on decimal point (x.y)
        BCF     LED_L                   ; Switch on left digit (CC)
        RETFIE

HANDLER_L
        BSF     LED_L                   ; Switch off left digit (CC)
        MOVFW   LSB                     ; Get value to be displayed on right digit
        ANDLW   0x0F                    ; Make sure the lookup table doesn't overflow
        CALL    bin2seg
        MOVWF   PORTB
        BCF     DP
        BTFSC   SHOWTH                  ; In TH mode?
        BSF     DP                      ; Switch on decimal point (xy.)
        BCF     LED_R                   ; Switch on right digit (CC)
        RETFIE

; 7-segment lookup
bin2seg ADDWF   PCL,F                   ; Jump into the lookup table
;                 gfed-cba                LED segments (RB3 is PWM)
        RETLW   B'01111111'             ; Return segment code for 0
        RETLW   B'00001110'             ; Return segment code for 1
        RETLW   B'10111011'             ; Return segment code for 2
        RETLW   B'10011111'             ; Return segment code for 3
        RETLW   B'11001110'             ; Return segment code for 4
        RETLW   B'11011101'             ; Return segment code for 5
        RETLW   B'11111101'             ; Return segment code for 6
        RETLW   B'00001111'             ; Return segment code for 7
        RETLW   B'11111111'             ; Return segment code for 8
        RETLW   B'11011111'             ; Return segment code for 9
        RETLW   B'11101111'             ; Return segment code for A
        RETLW   B'11111100'             ; Return segment code for b
        RETLW   B'01111001'             ; Return segment code for C
        RETLW   B'10111110'             ; Return segment code for d
        RETLW   B'11111001'             ; Return segment code for E
        RETLW   B'11101001'             ; Return segment code for F

LAMPTEST
        BTFSS   LED_L
        GOTO    $+4
        BCF     LED_L
```

```
        BSF     LED_R
        RETFIE

        BSF     LED_L
        BCF     LED_R
        DECFSZ  CNTR,F
        RETFIE

CONTLT
        MOVLW   0x20                    ; Duration of lamp test
        MOVWF   CNTR
        BTFSC   PORTB,7                 ; Segment g lit?
        GOTO    OFFSEGG
        BTFSC   PORTB,6                 ; Segment f lit?
        GOTO    OFFSEGF
        BTFSC   PORTB,0                 ; Segment a lit?
        GOTO    OFFSEGA
        BTFSC   PORTB,1                 ; Segment b lit?
        GOTO    OFFSEGB
        BTFSC   PORTB,2                 ; Segment c lit?
        GOTO    OFFSEGC
        BTFSC   PORTB,4                 ; Segment d lit?
        GOTO    OFFSEGD
        BCF     PORTB,5                 ; Switch off segment e
        BSF     PORTB,7                 ; Switch on segment g
        BCF     LAMPTST
        RETFIE

OFFSEGG                                 ; Switch off segment g, switch on segment f
        BCF     PORTB,7
        BSF     PORTB,6
        RETFIE

OFFSEGF                                 ; Switch off segment f, switch on segment a
        BCF     PORTB,6
        BSF     PORTB,0
        RETFIE

OFFSEGA                                 ; Switch off segment a, switch on segment b
        BCF     PORTB,0
        BSF     PORTB,1
        RETFIE

OFFSEGB                                 ; Switch off segment b, switch on segment c
        BCF     PORTB,1
        BSF     PORTB,2
        RETFIE

OFFSEGC                                 ; Switch off segment c, switch on segment d
        BCF     PORTB,2
        BSF     PORTB,4
        RETFIE

OFFSEGD                                 ; Switch off segment d, switch on segment e
        BCF     PORTB,4
        BSF     PORTB,5
        RETFIE

; End interrupt routines

; Subroutines
BIN2BCD
; Decodes binary value in BIN to BCD and shows the result on display (MSB & LSB files)
; http://www.piclist.com/techref/microchip/math/radix/b2bhp-8b3d.htm
;*******************************
;binary_to_bcd - 8-bits
;
;Input
;       bin     - 8-bit binary number
;        A1*16+A0
;Outputs
; hundreds - the hundreds digit of the BCD conversion
; BCD - the tens and ones digits of the BCD conversion
        CLRF    hundreds
        SWAPF   BIN,W                   ; swap the nibbles
        ADDWF   BIN,W                   ; so we can add the upper to the lower
```

```
        ANDLW   B'00001111'             ; lose the upper nibble (W is in BCD from now on)
        BTFSC   STATUS,DC               ; if we carried a one (upper + lower > 16)
        ADDLW   0x16                    ; add 16 (the place value) (1s + 16 * 10s)
        BTFSC   STATUS,DC               ; did that cause a carry from the 1's place?
        ADDLW   0x06                    ; if so, add the missing 6 (carry is only worth 10)
        ADDLW   0x06                    ; fix max digit value by adding 6
        BTFSS   STATUS,DC               ; if was greater than 9, DC will be set
        ADDLW   -0x06                   ; if if it wasn't, get rid of that extra 6

        BTFSC   BIN,4                   ; 16's place
        ADDLW   0x16 - 1 + 0x6          ; add 16 - 1 and check for digit carry
        BTFSS   STATUS,DC
        ADDLW   -0x06                   ; if nothing carried, get rid of that 6

        BTFSC   BIN,5                   ; 32nd's place
        ADDLW   0x30                    ; add 32 - 2

        BTFSC   BIN,6                   ; 64th's place
        ADDLW   0x60                    ; add 64 - 4

        BTFSC   BIN,7                   ; 128th's place
        ADDLW   0x20                    ; add 128 - 8 % 100

        ADDLW   0x60                    ; has the 10's place overflowed?
        RLF     hundreds,F              ; pop carry in hundreds' LSB
        BTFSS   hundreds,0              ; if it hasn't
        ADDLW   -0x60                   ; get rid of that extra 60

        BTFSC   BIN,7                   ; remember adding 28 - 8 for 128?
        INCF    hundreds,F              ; add the missing 100 if bit 7 is set
        BTFSS   hundreds,0              ; BCD>99?
        GOTO    $+2                     ; No
        MOVLW   0x99                    ; Limit value to 99

HEXDISPLAY
; Decodes the BCD (HEX) value in WREG and stores its HEX value in MSB and LSB
        MOVWF   TEMP1
        MOVWF   TEMP2
        RRF     TEMP1,F
        RRF     TEMP1,F
        RRF     TEMP1,F
        RRF     TEMP1,F
        MOVLW   0x0F
        ANDWF   TEMP1,W                 ; Blank high nibble and store in W
        MOVWF   MSB
        MOVLW   0x0F
        ANDWF   TEMP2,W                 ; Blank low nibble and store in W
        MOVWF   LSB
        RETURN

DELAY1                                  ; Ugly delay loop 0.10 s
        DECFSZ  d1,F
        GOTO    $-1
        DECFSZ  d2,F
        GOTO    $-3
        RETURN

DELAY4                                  ; Ugly delay loop 45 ms
        MOVLW   0x72
        MOVWF   d2
        DECFSZ  d1,F
        GOTO    $-1
        CLRWDT
        DECFSZ  d2,F
        GOTO    $-4
        RETURN

DO_ADC
        MOVLW   B'10000001'             ; AD conv ON, AN0 Selected (LM), 64T(osc)
        MOVWF   ADCON0
        NOP
        BSF     ADCON0,GO               ; Start A/D

CONV0                                   ; Dummy A/D
        NOP
        BTFSC   ADCON0,GO               ; Test if A/D conversion done
```

```
        GOTO    CONV0
        NOP
        BSF     ADCON0,GO               ; Start A/D

CONV0A                                  ; Used A/D
        NOP
        BTFSC   ADCON0,GO               ; Test if A/D conversion done
        GOTO    CONV0A
        BSF     STATUS,RP0              ; Select Bank 1
        BCF     STATUS,C                ; Make sure carry is clear before rotate
        RRF     ADRESL,W                ; Rotate A/D result into W (limit value to 0x7F)
        BCF     STATUS,RP0              ; Select Bank 0
        MOVWF   TEMP1
        ADDWF   LM_PREV,F
        BCF     STATUS,C                ; Clear carry before rotate
        RRF     LM_PREV,W               ; Divide by two
        MOVWF   LM                      ; Store into LM
        MOVFW   TEMP1                   ; Move LM (new)
        MOVWF   LM_PREV                 ; to LM_PREV
        RETURN

DO_SANITY                               ; Sanity check if TH and SL are within bounds in case of
                                        ; glitches
        BTFSS   UP                      ; Up button pressed?
        RETURN                          ; Yes
        MOVLW   0x3C
        SUBWF   TH,W                    ; TH<=60?
        BTFSC   STATUS,C
        GOTO    $+5                     ; No
        MOVLW   0x09
        SUBWF   SL,W                    ; SL<=4.5?
        BTFSS   STATUS,C
        RETURN                          ; Yes

; Restore last saved values from EEPROM
        CLRW
        CALL    EEPROMREAD
        MOVWF   TH                      ; Restored TH
        MOVLW   0x01
        CALL    EEPROMREAD
        MOVWF   SL                      ; Restored SL
        RETURN

DO_PWM                                  ; Calculate PWM based on TH and SL
        MOVFW   TH
        SUBWF   LM,W
        MOVWF   DIFF
        BTFSS   STATUS,C                ; DIFF negative?
        CLRF    DIFF                    ; Yes
        CLRF    DIFF_MULT               ; Clear DIFF Multiplicand
        MOVFW   SL
        MOVWF   SL_MULT                 ; Slope Multiplier
        BCF     STATUS,C

DO_MULT
        MOVFW   DIFF
        ADDWF   DIFF_MULT,F             ; Multiply single step
        BTFSC   STATUS,C                ; DIFF_MULT overflow?
        GOTO    OVF_PWM                 ; Yes
        DECFSZ  SL_MULT,F               ; Done multiply?
        GOTO    DO_MULT                 ; No
        GOTO    DONE_PWM

OVF_PWM
        MOVLW   0xE0
        MOVWF   DIFF_MULT

DONE_PWM
        BCF     STATUS,C                ; Make sure carry is zero
        RRF     DIFF_MULT,W             ; Divide by two into WREG
        ADDLW   0x0F                    ; Minimum duty cycle 12.5%
        MOVWF   CCPR1L                  ; and store in PWM register
        RETURN

DO_BUTTONS                              ; Button handler
        BTFSS   UP                      ; Slope/Up button pressed?
```

```
        GOTO    DO_SL
        BTFSS   DN                      ; Threshold/Down button pressed?
        GOTO    DO_TH
        MOVLW   0x0A
        MOVWF   REPT                    ; Reset button one second repeat counter (threshold)
        BTFSC   HELDSL                  ; Button is not held anymore?
        GOTO    SAVE_SL
        BTFSC   HELDTH
        GOTO    SAVE_TH
        RETURN

DO_TH
        BTFSC   SHOWTH                  ; In set TH mode?
        GOTO    DEC_TH                  ; Yes
        BTFSC   SHOWSL                  ; In set SL mode?
        GOTO    DEC_SL                  ; Yes
        BSF     SHOWTH                  ; Enter set TH mode
        MOVLW   0x0E
        MOVWF   CNTR                    ; 1.5 s counter
        BSF     HELDTH
        RETURN

DEC_TH
        MOVLW   0x0E
        MOVWF   CNTR                    ; 1.5 s counter
        DECFSZ  REPT,F
        GOTO    $+2
        GOTO    REP_DEC_TH
        BTFSC   HELDTH
        RETURN
        BSF     HELDTH
        DECFSZ  TH,F
        RETURN
        INCF    TH,F                    ; TH can't become zero
        RETURN

REP_DEC_TH
        MOVLW   0x02
        MOVWF   REPT                    ; 0.3 s repeat button
        DECFSZ  TH,F
        RETURN
        INCF    TH,F                    ; TH can't become zero
        RETURN

INC_TH
        MOVLW   0x0E
        MOVWF   CNTR                    ; 1.5 s counter
        DECFSZ  REPT,F
        GOTO    $+2
        GOTO    REP_INC_TH
        BTFSC   HELDTH
        RETURN
        BSF     HELDTH
        INCF    TH,F
        MOVLW   0x3C
        SUBWF   TH,W                    ; TH<=60?
        BTFSS   STATUS,C
        RETURN                          ; Yes
        MOVLW   0x3C
        MOVWF   TH                      ; Set TH to 60
        RETURN

REP_INC_TH
        MOVLW   0x02
        MOVWF   REPT                    ; 0.3 s repeat button
        INCF    TH,F
        MOVLW   0x3C
        SUBWF   TH,W                    ; TH<=60?
        BTFSS   STATUS,C
        RETURN                          ; Yes
        MOVLW   0x3C
        MOVWF   TH                      ; Set TH to 60
        RETURN

SAVE_TH
        MOVFW   TH
```

```
        MOVWF   EEBUF
        CLRW                            ; EEPROM address for TH
        CALL    EEPROMWRITE
        BCF     HELDTH
        RETURN

DO_SL
        BTFSS   DN                      ; TH button also pressed?
        GOTO    DO_SHOWPWM
        BTFSC   SHOWTH                  ; In set TH mode?
        GOTO    INC_TH                  ; Yes
        BTFSC   SHOWSL                  ; In set SL mode?
        GOTO    INC_SL                  ; Yes
        BSF     SHOWSL                  ; Enter set SL mode
        MOVLW   0x0E
        MOVWF   CNTR                    ; 1.5 s counter
        BSF     HELDSL
        RETURN

DEC_SL
        MOVLW   0x0E
        MOVWF   CNTR                    ; 1.5 s counter
        BTFSC   HELDSL
        RETURN
        BSF     HELDSL
        DECFSZ  SL,F
        RETURN
        INCF    SL,F                    ; SL can't become zero
        RETURN

INC_SL
        MOVLW   0x0E
        MOVWF   CNTR                    ; 1.5 s counter
        BTFSC   HELDSL
        RETURN
        BSF     HELDSL
        INCF    SL,F
        MOVLW   0x09
        SUBWF   SL,W                    ; SL<=4.5?
        BTFSS   STATUS,C
        RETURN                          ; Yes
        MOVLW   0x09
        MOVWF   SL                      ; Set SL to 4.5
        RETURN

SAVE_SL
        MOVFW   SL
        MOVWF   EEBUF
        MOVLW   0x01                    ; EEPROM address for SL
        CALL    EEPROMWRITE
        BCF     HELDSL
        RETURN

DO_SHOWPWM
        BSF     SHOWPCM
        BCF     HELDSL
        BCF     HELDTH
        BCF     SHOWSL
        BCF     SHOWTH
        RETURN

DO_POWERUP                              ; Ramp up and down to max in 10 s
        CLRF    PORTA                   ; Switch on both displays
        INCF    CCPR1L,F
        CALL    DELAY4                  ; 45 ms delay
        MOVLW   0x80
        SUBWF   CCPR1L,W
        BTFSS   STATUS,Z                ; Duty cycle max?
        GOTO    $-5                     ; No
        DECF    CCPR1L,F
        CALL    DELAY4                  ; 45 ms delay
        MOVLW   0x0F
        SUBWF   CCPR1L,W
        BTFSS   STATUS,Z                ; Duty cycle back to 12%?
        GOTO    $-5                     ; No
        BCF     POWERUP
```

```
        RETURN

EEPROMREAD
        ; W contains address to read, at return contains read data
        BANKSEL EEADR                   ; Select Bank of EEADR
        MOVWF   EEADR                   ; Data Memory Address to read from W
        BANKSEL EECON1                  ; Select Bank of EECON1
        BCF     EECON1,EEPGD            ; Point to Data memory
        BSF     EECON1,RD               ; EE Read
        BANKSEL EEDATA                  ; Select Bank of EEDATA
        MOVF    EEDATA,W                ; W = EEDATA
        BCF     STATUS,RP0
        BCF     STATUS,RP1
        RETURN

EEPROMWRITE
        ; W contains address to write, data is in EEBUF
        BANKSEL EECON1                  ; Select Bank of EECON1
        BTFSC   EECON1,WR               ; Wait for write
        GOTO    $-1                     ; to complete
        BANKSEL EEADR                   ; Select Bank of EEADR
        MOVWF   EEADR                   ; Data Memory Address to write
        BANKSEL EEBUF
        MOVFW   EEBUF                   ; Data to write from buffer
        BANKSEL EEDATA
        MOVWF   EEDATA                  ; Data Memory Value to write
        BANKSEL EECON1                  ; Select Bank of EECON1
        BCF     EECON1,EEPGD            ; Point to DATA memory
        BSF     EECON1,WREN             ; Enable writes
        MOVLW   0x55
        MOVWF   EECON2                  ; Write 55h
        MOVLW   0xAA
        MOVWF   EECON2                  ; Write AAh
        BSF     EECON1,WR               ; Set WR bit to begin write
        BCF     EECON1,WREN             ; Disable writes
        BCF     STATUS,RP0
        BCF     STATUS,RP1
        RETURN

; end subroutines

START
; Init stuff
        CLRF    STATUS                  ; Do initialization, Select bank 0
        CLRF    INTCON
        CLRF    PCLATH                  ; Keep in lower 2KByte
        CLRF    CCP1CON
        BSF     STATUS,RP0              ; Select bank 1
        MOVLW   B'00111101'
        MOVWF   TRISA                   ; RA7,6,1 Outputs, 5-2,0 Inputs
        CLRF    TRISB                   ; RB7-0 Outputs
        MOVLW   B'00000011'             ; Timer0, prescaler 1:16, WDT 16 ms (DELAY4!)
        MOVWF   OPTION_REG
        MOVLW   B'01110000'             ; 8 MHz clock
        MOVWF   OSCCON
        MOVLW   B'11001110'             ; AN0 Analog input, ADRESL only (ADRESH discarded, right
                                        ; justified)
        MOVWF   ADCON1
        MOVLW   0x63
        MOVWF   PR2                     ; 50 µs PWM
        BCF     STATUS,RP0              ; Select bank 0
        MOVWF   TMR2
        MOVLW   0x04                    ; Timer2 ON, prescale 1:1
        MOVWF   T2CON
        MOVLW   0x0F
        MOVWF   CCP1CON                 ; PWM mode
        MOVWF   CCPR1L                  ; PWM 12.5%
        MOVLW   B'01000001'             ; AD conv ON, AN0 Selected, 16T(osc)
        MOVWF   ADCON0
        CLRF    FLAGS
        CLRF    PORTA                   ; Make all PORT A outputs low
        BSF     LED_L                   ; Switch on left digit
        BSF     LAMPTST
        MOVLW   0x20
        MOVWF   CNTR                    ; Lamp Test Counter
        MOVLW   0x0A
```

```
        MOVWF   REPT                    ; Threshold button repeat
        BSF     POWERUP
        MOVLW   0x80
        MOVWF   PORTB                   ; Switch on segment g to start lamp test
        CLRF    MSB
        CLRF    LSB
        MOVLW   0x01
        MOVWF   SKIP

; Restore data from EEPROM
        CLRW
        CALL    EEPROMREAD
        MOVWF   TH                      ; Restored TH
        MOVLW   0x01
        CALL    EEPROMREAD
        MOVWF   SL                      ; Restored SL

        CLRF    INTCON
        BSF     INTCON,TMR0IE           ; Enable Timer 0 interrupt
        BSF     INTCON,GIE              ; Enable global interrupts
        CALL    DO_ADC                  ; Populate LM and LM_PREV for the first time to prevent
                                        ; glitch in PWM
        MOVFW   LM
        MOVWF   BIN
        CALL    BIN2BCD                 ; Move binary value from BIN to MSB and LSB (BCD display
                                        ; 00-99)

MAIN
        BTFSC   LAMPTST                 ; Lamp Test ongoing?
        GOTO    MAIN                    ; Yes
        BTFSS   POWERUP                 ; Powerup ongoing?
        GOTO    $+4                     ; No
        BCF     INTCON,TMR0IE           ; Disable display (multiplex) during processing
        CALL    DO_POWERUP              ; 10 s ramp up/down fan
        GOTO    MAIN
        BSF     INTCON,TMR0IE           ; Re-enable display (multiplex)
        CALL    DELAY1                  ; Ugly 0.1 s delay (debounce)
        BCF     INTCON,TMR0IE           ; Disable display (multiplex) during processing
        CLRF    PORTB                   ; Blank display (common cathode)

        CALL    DO_BUTTONS

        DECFSZ  SKIP,F                  ; Skip ADC?
        GOTO    SKIP_ADC
        MOVLW   0x08
        MOVWF   SKIP                    ; 0.8 seconds of skipping ADC
        CALL    DO_ADC                  ; Get LM value
        CALL    DO_SANITY               ; Check if SL and TH values are valid, silent restore if
                                        ; not
        CALL    DO_PWM                  ; Calculate PWM based on LM

SKIP_ADC
        BTFSC   SHOWPCM                 ; Debug PCM value?
        GOTO    DEBUGPCM                ; Yes
        BTFSS   SHOWSL                  ; Show slope?
        GOTO    TESTTH                  ; No
        MOVFW   SL                      ; Prepare SL for display (SL/2) (x.y)
        MOVWF   TEMP1                   ; X
        MOVLW   0x05
        MOVWF   TEMP2                   ; Y

; http://www.piclist.com/techref/microchip/math/mul/4x4.htm
        CLRW                            ; Clear Result
        BCF     STATUS,C                ; Clear Carry for RLF later                     ;
        BTFSC   TEMP2,0                 ; ?*1
        ADDWF   TEMP1,W                 ; W = ?X
        RLF     TEMP1,F                 ; X = 2X
        BTFSC   TEMP2,1                 ; ?*2
        ADDWF   TEMP1,W                 ; W = ?X + ?2X
        RLF     TEMP1,F                 ; W = 4X
        BTFSC   TEMP2,2                 ; ?*4
        ADDWF   TEMP1,W                 ; W = ?X + ?2X + ?4X
        RLF     TEMP1,F                 ; W = 8X
        BTFSC   TEMP2,3                 ; ?*8
        ADDWF   TEMP1,W                 ; ?X + ?2X + ?4X + ?8X
        MOVWF   BIN
```

```
        CALL    BIN2BCD
        DECFSZ  CNTR,F
        GOTO    MAIN
        BCF     SHOWSL
        GOTO    MAIN

TESTTH
        BTFSS   SHOWTH                  ; Show threshold?
        GOTO    SHOWLM                  ; No
        MOVFW   TH
        MOVWF   BIN
        CALL    BIN2BCD
        DECFSZ  CNTR,F
        GOTO    MAIN
        BCF     SHOWTH
        GOTO    MAIN

DEBUGPCM                                ; Show PCM HEX value
        MOVFW   CCPR1L
        CALL    HEXDISPLAY
        BCF     SHOWPCM
        GOTO    MAIN

SHOWLM                                  ; Show temperature
        MOVFW   LM
        MOVWF   BIN
        CALL    BIN2BCD                 ; Move binary value from BIN to MSB and LSB (BCD display
                                        ; 00-99)
        GOTO    MAIN

        END
```