EE3043D EMBEDDED SYSTEMS COURSE PROJECT

DIGITAL THERMOMETER

Submitted by:

Madhav Menon
(B210821EE)

DEPARTMENT OF ELECTRICAL ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT

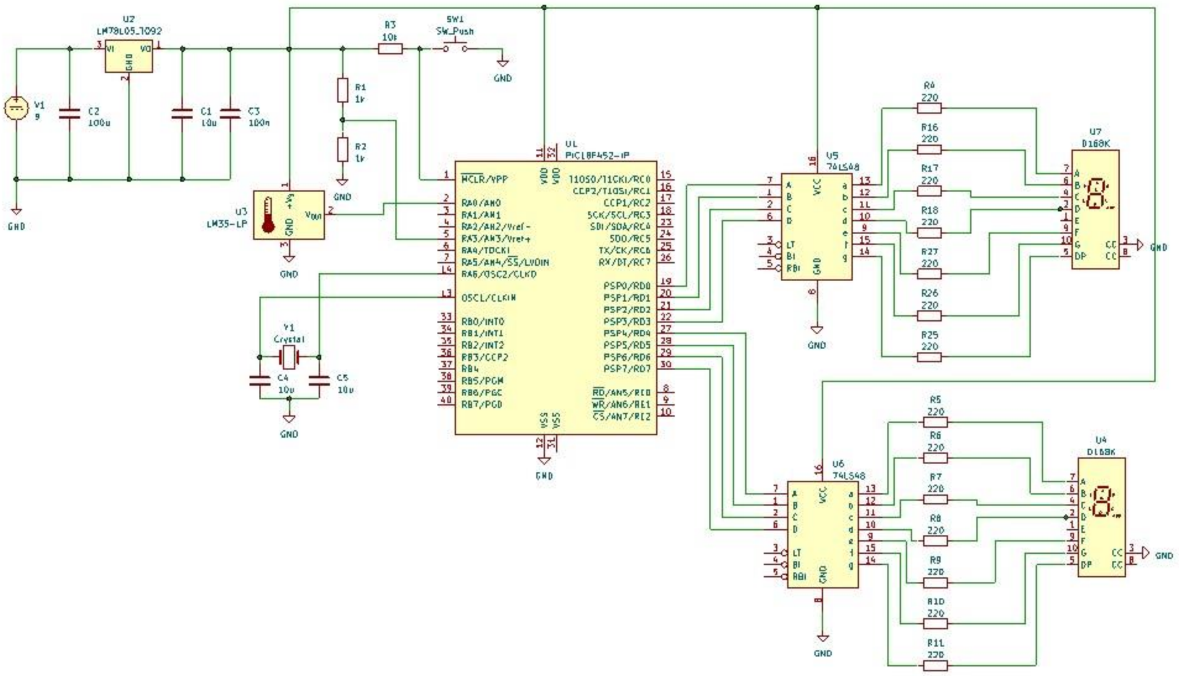
AIM

To display temperature digitally in two 7 segment LEDs using an LM35 temperature sensor and PIC18F452 Microcontroller

COMPONENTS USED

- PIC18F452
- LM35 Temperature Sensor
- 7 Segment LEDs
- Capacitors – 100uF, 10uF, 22pF
- Resistors – 1k ohm, 10k ohm, 220 ohm
- IC 7448 BCD to 7 Segment Decoder
- LM7805 Voltage Regulator
- Crystal Oscillator – 16 MHz

CIRCUIT DIAGRAM



WORKING PRINCIPLE

- Interrupt method is used for measuring the temperature.
- The Analog Input from LM35 is sent to RA0 pin (Channel 0 of ADC) of PIC18f452.
- A frequency equal to 1/64 times the oscillator frequency is used as clock source for the ADC.
- Reference Voltage is set to 2.56V so that step size of the 10 bit ADC becomes 2.5mV .
- LM35 has a sensor gain of 10mV/°C which means the binary output of the ADC is 4 times the actual temperature.
- Features of LM35 Sensor :
  - Calibrated Directly in Celsius
  - Linear +10mV/°C Scale Factor
  - Operates from 4V to 30V
  - Less than 60uA Current Drain
  - Low Self Heating, 0.08°C in Still Air
- So the output needs to be divided by 4 to get the correct temperature.
- The corresponding binary output is converted to BCD and is fed to the BCD to 7 Segment Decoder IC 7448.
- Common Cathode 7 Segment LEDs are connected to the IC which displays the temperature.

CODE

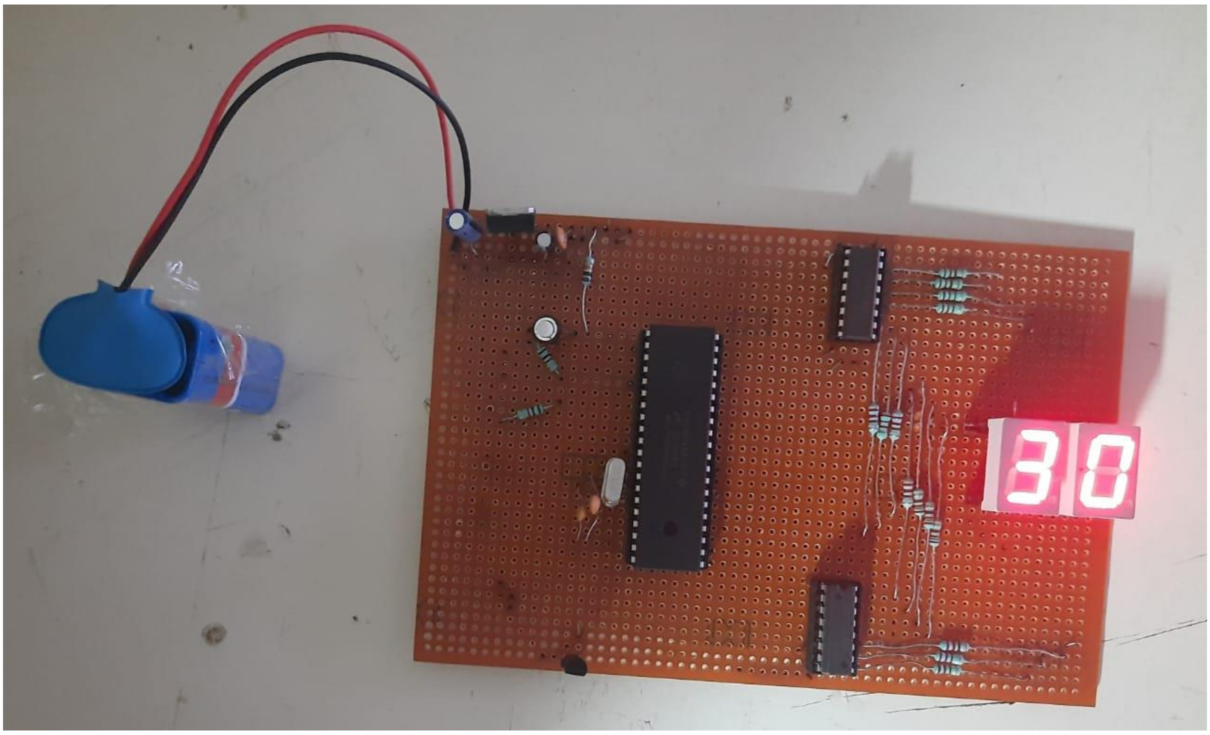
```
; PIC18F452 Configuration Bit Settings

; Assembly source line config statements

; CONFIG1H
  CONFIG  OSC = HS              ; Oscillator Selection bits (HS oscillator)
  CONFIG  OSCS = OFF            ; Oscillator System Clock Switch Enable bit
(Oscillator system clock switch option is disabled (main oscillator is source))

; CONFIG2L
  CONFIG  PWRT = OFF            ; Power-up Timer Enable bit (PWRT disabled)
  CONFIG  BOR = OFF             ; Brown-out Reset Enable bit (Brown-out Reset
enabled)
  CONFIG  BORV = 20             ; Brown-out Reset Voltage bits (VBOR set to 2.0V)

; CONFIG2H
```

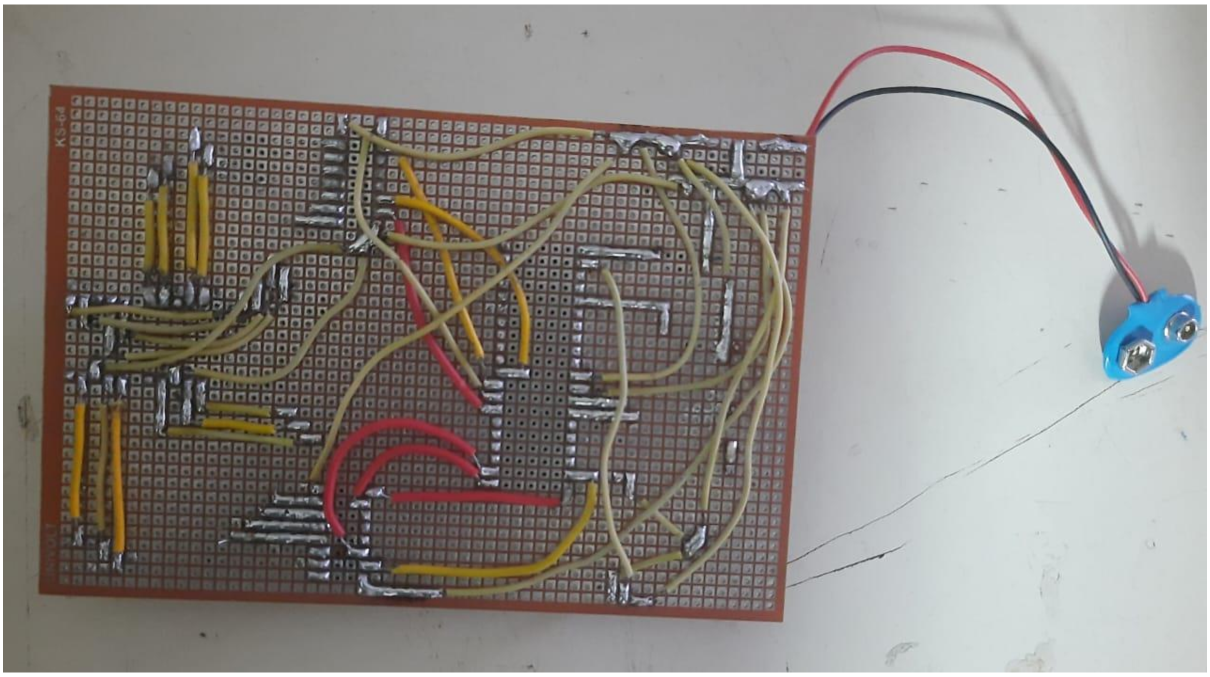
```
  CONFIG  WDT = OFF              ; Watchdog Timer Enable bit (WDT enabled)
  CONFIG  WDTPS = 128           ; Watchdog Timer Postscale Select bits (1:128)

; CONFIG3H
  CONFIG  CCP2MUX = OFF         ; CCP2 Mux bit (CCP2 input/output is multiplexed
with RB3)

; CONFIG4L
  CONFIG  STVR = OFF            ; Stack Full/Underflow Reset Enable bit (Stack
Full/Underflow will not cause RESET)
  CONFIG  LVP = OFF             ; Low Voltage ICSP Enable bit (Low Voltage ICSP
disabled)

; CONFIG5L
  CONFIG  CP0 = OFF             ; Code Protection bit (Block 0 (000200-001FFFh)
not code protected)
  CONFIG  CP1 = OFF             ; Code Protection bit (Block 1 (002000-003FFFh)
not code protected)
  CONFIG  CP2 = OFF             ; Code Protection bit (Block 2 (004000-005FFFh)
not code protected)
  CONFIG  CP3 = OFF             ; Code Protection bit (Block 3 (006000-007FFFh)
not code protected)

; CONFIG5H
  CONFIG  CPB = OFF             ; Boot Block Code Protection bit (Boot Block
(000000-0001FFh) not code protected)
  CONFIG  CPD = OFF             ; Data EEPROM Code Protection bit (Data EEPROM not
code protected)

; CONFIG6L
  CONFIG  WRT0 = OFF            ; Write Protection bit (Block 0 (000200-001FFFh)
not write protected)
  CONFIG  WRT1 = OFF            ; Write Protection bit (Block 1 (002000-003FFFh)
not write protected)
  CONFIG  WRT2 = OFF            ; Write Protection bit (Block 2 (004000-005FFFh)
not write protected)
  CONFIG  WRT3 = OFF            ; Write Protection bit (Block 3 (006000-007FFFh)
not write protected)

; CONFIG6H
  CONFIG  WRTC = OFF            ; Configuration Register Write Protection bit
(Configuration registers (300000-3000FFh) not write protected)
  CONFIG  WRTB = OFF            ; Boot Block Write Protection bit (Boot Block
(000000-0001FFh) not write protected)
  CONFIG  WRTD = OFF            ; Data EEPROM Write Protection bit (Data EEPROM
not write protected)

; CONFIG7L
  CONFIG  EBTR0 = OFF           ; Table Read Protection bit (Block 0 (000200-
001FFFh) not protected from Table Reads executed in other blocks)
```

```
  CONFIG  EBTR1 = OFF           ; Table Read Protection bit (Block 1 (002000-003FFFh) not protected from Table Reads executed in other blocks)
  CONFIG  EBTR2 = OFF           ; Table Read Protection bit (Block 2 (004000-005FFFh) not protected from Table Reads executed in other blocks)
  CONFIG  EBTR3 = OFF           ; Table Read Protection bit (Block 3 (006000-007FFFh) not protected from Table Reads executed in other blocks)
; CONFIG7H
  CONFIG  EBTRB = OFF           ; Boot Block Table Read Protection bit (Boot Block (000000-0001FFh) not protected from Table Reads executed in other blocks)

#include <p18f452.inc>

L_Byte    EQU 0x20
H_Byte    EQU 0x21
BIN_TEMP  EQU 0x22
MYREG     EQU 0x08
BCD_OUT   EQU 0x15
BIT_CTR   EQU 0x17

      ORG 0000H
      GOTO MAIN                ; bypass interrupt vector table

      ORG 0008H
      BTFSS PIR1, ADIF         ; Did we get here due to A/D int?
      RETFIE                   ; NO. Then return to main
      GOTO AD_ISR              ; Yes, Then go to AD_ISR

      ORG 00100H
MAIN  CLRF TRISD               ; make PORTD an output
      BSF TRISA, 0             ; make RA0 an input pin for analog input
      BSF TRISA, 3             ; make RA03 an input pin for Vref
      MOVLW 0x81               ; Fosc/64, channel 0, A/D is on
      MOVWF ADCON0
      MOVLW 0xC5               ; right justified, Fosc/64
      MOVWF ADCON1
      BCF PIR1, ADIF           ; clear ADIFfor the first round
      BSF PIE1, ADIE           ; enable A/D interrupt
      BSF INTCON, PEIE         ; enable peripheral interrupts
      BSF INTCON, GIE          ; enable interrupts globally
OVER  CALL DELAY1              ; wait for Tacq (sample and hold time)
      BSF ADCON0, GO           ; start conversion
      BRA OVER

AD_ISR
      ORG 200H
      MOVFF ADRESL, L_Byte ; save the low byte
      MOVFF ADRESH, H_Byte ; save the high byte
      RRNCF L_Byte, F          ; rotate right twice
      RRNCF L_Byte, W
      ANDLW 0x3F               ; mash the upper 2 bits
      MOVWF L_Byte
      RRNCF H_Byte, F      ; rotate right twice
      RRNCF H_Byte, W
      ANDLW 0xC0           ; mask the lower 6 bits
      IORWF L_Byte, W      ; combine low and high
      MOVWF BIN_TEMP
      CLRF BCD_OUT
      MOVLW D'8'
      MOVWF BIT_CTR        ; counter of number of bits to convert

BIN_TO_BCD
      RLCF BIN_TEMP        ; rotate left through carry
      MOVF BCD_OUT, W
      ADDWFC BCD_OUT,W     ; add to itself (double) and carry from rotate of
BIN_TEMP
      DAW                  ; Decimal adjust BCD word
      MOVWF BCD_OUT
      DECFSZ BIT_CTR       ; decrement the bit counter
      BRA BIN_TO_BCD

      MOVFF BCD_OUT, PORTD

      CALL DELAY2
      BCF PIR1, ADIF       ; clear ADIF interrupt flag bit
      RETFIE

DELAY1                     ; delay to get sample and hold time
      MOVLW D'5'
      MOVWF MYREG
AGAIN NOP
      NOP
      DECF MYREG, F
      BNZ  AGAIN
      RETURN

DELAY2
      MOVLW D'200'
      MOVWF 55H
B3    MOVLW D'100'
      MOVWF 56H
B2    MOVLW D'25'
      MOVWF 57H
B1    NOP
      NOP
      DECF 57H, F
      BNZ B1
      DECF 56H, F
      BNZ B2
      DECF 55H, F
      BNZ B3
      RETURN
END
```

RESULT

The digital thermometer showed accurate readings of the temperature to be measured.





REFERENCES

- PIC Microcontroller and Embedded Systems Using Assembly and C for PIC18, by Muhammad Ali Mazidi