Assignment no.	**03**		
Submitted by:	**Shahid Umar, Shehryar Aamer, Moeez Aslam**		
Registration no.	**170401021, 170401069, 170401036**		
Batch:	**16**	**Section:**	**EE-16-A**
Submitted to:	Sir. Saad		
Date:	**July 19th 2018**		

Introduction

An automatic temperature control system has the ability to monitor and control the temperature of the specified space without human intervention. The primary purpose is to manage the temperature of a given area based on settings by a user on settings by a user of the system.

Apparatus

PIC microcontroller 18f2520

LM35 temperature sensor

Keypad

And relays, resistors and transistors etc.

Parts

This project is divided into two parts

In first part temperature is recorded using the temperature sensor and in the second part the recorded temperature is compared with the reference temperature entered by the user.

Code

```
#include <xc.h>

#include <stdlib.h>

#include <stdio.h>

#include <plib/adc.h>

#include <plib/xlcd.h>

#include <plib/delays.h>

#include "config.c"     //Configuration bits


#define HEATER PORTAbits.RA1

#define FAN PORTAbits.RA2

#define ON 1
```

```
#define OFF 0


void init_ADC(void);       //Initialize ADC
void init_XLCD(void);      //Initialize LCD display
void DelayFor18TCY( void ); //18 cycles delay
void DelayPORXLCD (void);   // Delay of 15ms
void DelayXLCD (void);      // Delay of 5ms
int kbd_getc();


//#define set_port_kbd PORTB // Change if port is different
#define row1port LATCbits.LATC0
#define row2port LATCbits.LATC1
#define row3port LATCbits.LATC2
#define row4port LATCbits.LATC3
#define col1port PORTCbits.RC4
#define col2port PORTCbits.RC5
#define col3port PORTCbits.RC6
//#define col4port PORTBbits.RB7   //if a 4x4 keypad is used


unsigned char  const Seven_Segment_MAP[10] = {0,1,2,3,4,5,6,7,8,9};
char const keyPadMatrix[] =
{
   '1',  '2', '3',
   '4',  '5', '6',
   '7',  '8', '9',
   '*',  '0', '#',
```

```
    0xFF
};

char key,old_key;

//
// This function creates seconds delay. The argument specifies the delay time in seconds
//
void Delay_Seconds(unsigned char z)
{
    unsigned char x,y;
    for(y = 0; y < z; y++)
    {
        for(x = 0; x <  100; x++)__delay_ms(10);
    }
}
// This function clears the screen
//
void LCD_Clear()
{
    while(BusyXLCD());
    WriteCmdXLCD(0x01);
}

//
// This function moves the cursor to position row,column
```

```
//
void LCD_Move(unsigned char row, unsigned char column)
{
 char ddaddr = 0x40*(row-1) + column;
 while( BusyXLCD() );
 SetDDRamAddr( ddaddr );
}

//**************Declare Global Variables************************************
unsigned int ADCResult=0;
unsigned short Temp_Ref ;     //  Reference Temperature
float ActualTemp;
unsigned char stringKey[10],stringKeyActual[10];
unsigned long keypress;
/*
 *
 */
void main(void) {
    OSCCON=0x76;        //Configure to use 8MHz internal oscillator.
    TRISC   = 0xF0;          //Use PORTB for Keypad
    TRISB   = 0x00;          //Use PORTB to display
    PORTB   = 0x00;
    TRISAbits.RA0 = 1;
    TRISAbits.RA1 = 0;    //RA1 is output (Heater)
    TRISAbits.RA2 = 0;    //RA2 is output (Fan)
    LATB    = 0x00;
```

```
 init_XLCD();        //Call the Initialize LCD display function
 init_ADC();         //Call the Initialize ADC function

LCD_Move(1,3);              // Move to row=1,column=3
putrsXLCD("Automatic");         // Display heading
LCD_Move(2,1);              // Move to row=2,column=1
putrsXLCD("Temp Control...");   // Display heading
Delay_Seconds(2);           // 2 seconds delay
LCD_Clear();                //Clear display

HEATER = OFF;               //Switch OFF Heater on start up
FAN = OFF;                  //Switch OFF Fan on start up

  //ON startup, read the Reference Temperature from the Keypad
    START:
   LCD_Clear();                 //Clear display
   LCD_Move(1,1);               // Move to row=1,column=1
   putrsXLCD("Enter Temp Ref");
   Temp_Ref=0;
   LCD_Move(2,1);               // Move to row=2,column=1
   putrsXLCD("Temp Ref: ");

while (1){

   keypress = kbd_getc();
   if ( keypress == '#' )break;       //If Enter '#' pressed
```

```
      if ( keypress == '*' )goto START;  //If Clear '*' pressed

      if(keypress!=0xFF){
        putcXLCD(keypress ) ;
        Temp_Ref = (10*Temp_Ref) + Seven_Segment_MAP[keypress-'0'];
      }
}
      LCD_Clear();                  //Clear display
      LCD_Move(1,1);                // Move to row=1,column=1
      putrsXLCD("Temp Ref: ");
      itoa (stringKey,Temp_Ref,10);   //Convert to string in stringKey
      putsXLCD(stringKey);          //Display the temp ref
      LCD_Move(2,1);                // Move to row=2,column=1
       putrsXLCD("Press # to Cont.");

     keypress =0;
     while(keypress!='#')
     {
       do
        keypress = kbd_getc();  // read keypad
       while(!keypress);
     }
      LCD_Clear();             //Clear display
      LCD_Move(1,1);           // Move to row=1,column=1
      putrsXLCD("Temp Ref: ");
      putsXLCD(stringKey);     //Display the temp ref
```

```
    putcXLCD(223);          // Different LCD displays have different char code for degree
    putcXLCD('C');


    //Program loop
    while(1){
     //Display Referance Temperature and Actual Temperature
     ADCResult =0;
     ConvertADC(); //Start conversion
     while(BusyADC()); //Wait here until conversion is finished
     ADCResult = ReadADC(); //Read the converted data
     ActualTemp = (ADCResult*5.0)/10.24;      //convert data into temperature (LM35 produces 10mV per degree celcius)
     sprintf(stringKeyActual, "%.3g", ActualTemp ); // Convert voltage to string
     LCD_Move(2,1);                 // Move to row=2,column=1
     putrsXLCD("Temp is: ");
     putsXLCD(stringKeyActual);     //Display the temp ref
     putcXLCD(223);             // Different LCD displays have different char code for degree
     putcXLCD('C');
     putrsXLCD("  ");            // Clear after comma


     //Compare reference Temp with actual Temp
    if (Temp_Ref >  ActualTemp)  //If Temp Ref is greater than actual Temp, Switch ON Heater
    {
    HEATER = ON,
    FAN = OFF;
```

```
    }
     if (Temp_Ref <  ActualTemp)  //If Temp Ref is less than actual Temp, Switch ON Fan
    {
    HEATER = OFF,
    FAN = ON;
    }
    if (Temp_Ref ==  ActualTemp)  //If Temp Ref is equal to actual Temp, Switch OFF Fan and Heater
    {
    HEATER = OFF,
    FAN = OFF;
    }
      Delay_Seconds(10);   //Wait 10 s then repeat
   }

}

int kbd_getc(){
   // This routine returns the first key found to be pressed during the scan.
   char key = 0, row;

   for( row = 0b00000001; row < 0b00010000; row <<= 1 )
   {
     {  // turn on row output
        row1port = (row & 0x0001)>>0;
        row2port = (row & 0x0002)>>1;
        row3port = (row & 0x0004)>>2;
```

```
        row4port = (row & 0x0008)>>3;
        __delay_ms(1);
      }
      // read colums - break when key press detected
      if( col1port )break;  key++;
      if( col2port )break;  key++;
      if( col3port )break;  key++;
      //if( col4port )break;  key++;
    }
   row1port = 0;
   row2port = 0;
   row3port = 0;
   row4port = 0;
   if (key!=old_key){
    old_key=key;
    return keyPadMatrix[ key ];
   }
   else
   return keyPadMatrix[ 0x0C ];
}

void init_XLCD(void)            //Initialize LCD display
{
OpenXLCD(FOUR_BIT&LINES_5X7);     //configure LCD in 4-bit Data Interface mode
                        //and 5x7 characters, multiple line display
while(BusyXLCD());              //Check if the LCD controller is not busy
```

```
                        //before writing some commands
WriteCmdXLCD(0x06);             //Move cursor right, don't shift display
WriteCmdXLCD(0x0C);             //Turn display on without cursor
}
void init_ADC(void)             //Initialize ADC
{
OpenADC(ADC_FOSC_2 & ADC_RIGHT_JUST & ADC_2_TAD,
ADC_CH0 & ADC_INT_OFF & ADC_REF_VDD_VSS,
ADC_1ANA);
}
void DelayFor18TCY( void )      //18 cycles delay
{
Delay10TCYx(20);
}
void DelayPORXLCD (void)        //Delay of 15ms
{
Delay1KTCYx(30);
}
void DelayXLCD (void)           //Delay of 5ms
{
Delay1KTCYx(10);
}
```

Simulations