Documentation

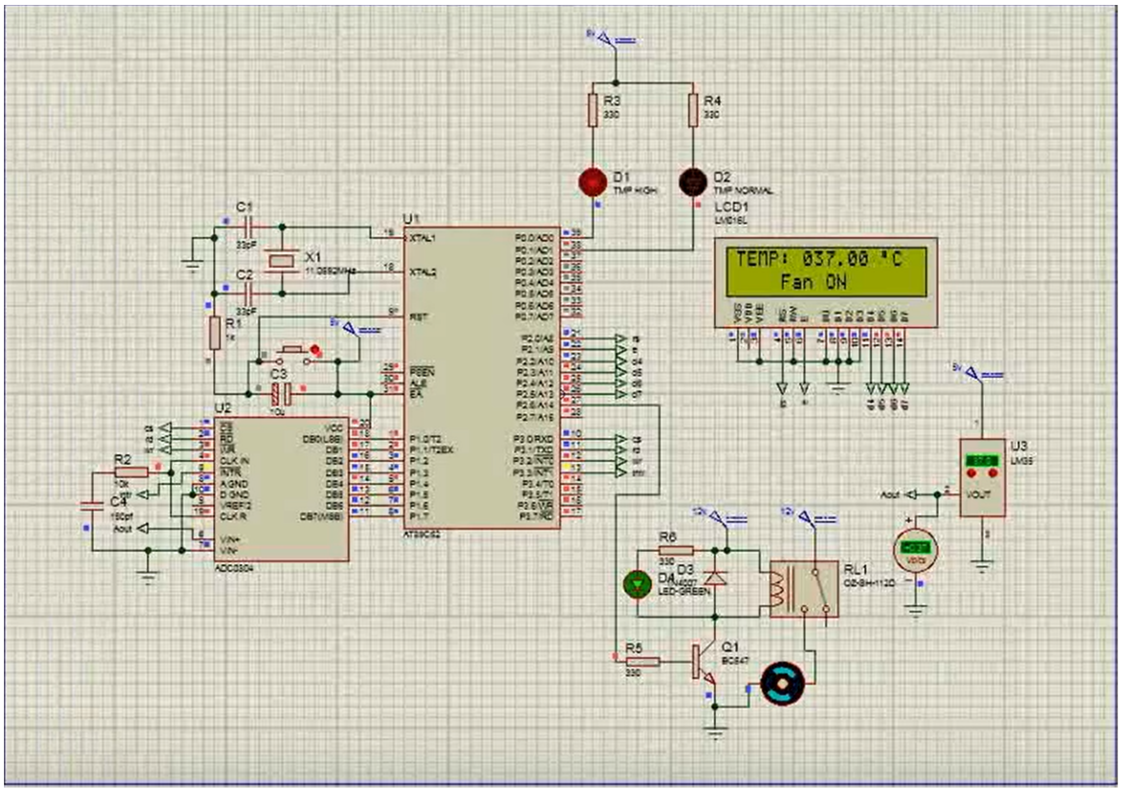
Aim

To design and implement a temperature monitoring and control system.

Components Required:

8051 Development Board
ADC8084 Board
16*2 LCD Display
LM35 Sensor
Jumper Wire
Fan

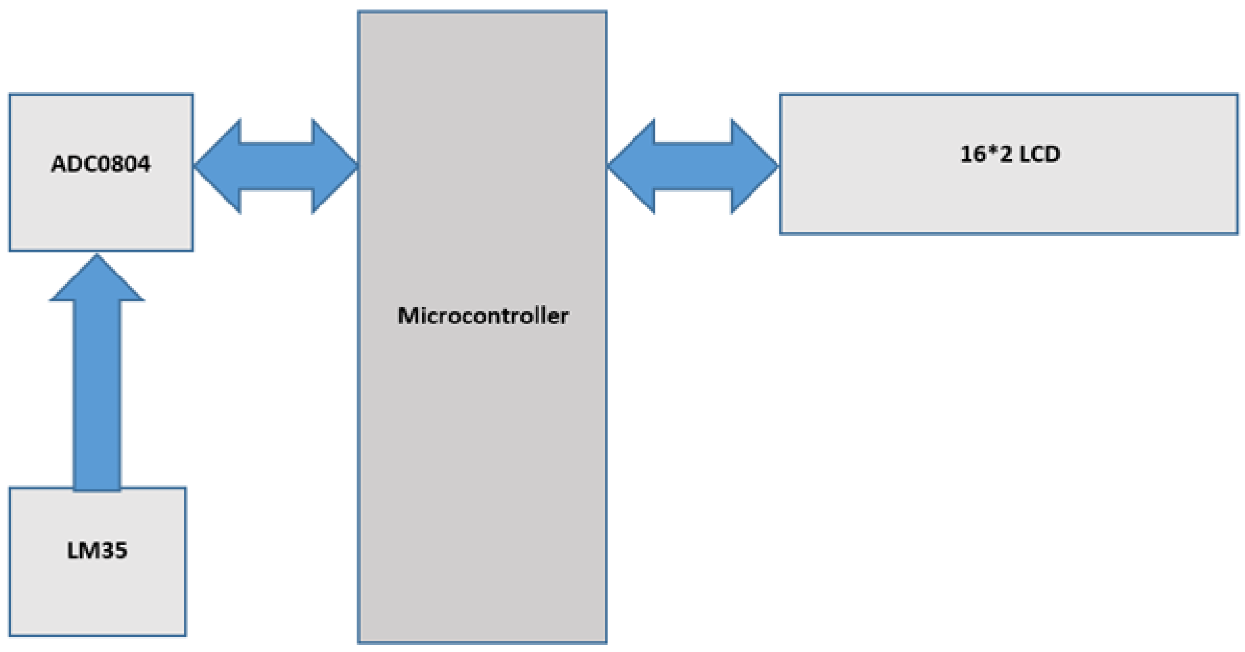
Circuit Diagram:



Measuring Temperature with LM35 using 8051:

8051 microcontroller is a 8 bit microcontroller which has 128 bytes of on chip RAM , 4K bytes of on chip ROM, two timers, one serial port and four 8bit ports. 8052 microcontroller is an extension of microcontroller. The table below shows the comparison of 8051 family members.

Feature	8051	8052
ROM (in bytes)	4K	8K
RAM (bytes)	128	256
Timers	2	3
I/O pins	32	32
Serial port	1	1
Interrupt sources	6	8



16x2 LCD:

16*2 LCD is a widely used display for embedded applications. Here is the brief explanation about pins and working of 16*2 LCD display. There are two very important registers inside the LCD. They are data register and command register. Command register is used to send commands such as clear display, cursor at home etc., data register is used to send data which is to be displayed on 16*2 LCD. Below table shows the pin description of 16*2 lcd.

Pin	Symbol	I/O	Description
1	Vss	-	Ground
2	Vdd	-	+5V power supply

3	Vee	-	Power supply to control contrast

4	RS	I	RS=0 for command register ,
			RS=1 for data register

5	RW	I	R/W=0 for write , R/W=1 for read
6	E	I/O	Enable
7	D0	I/O	8-bit data bus(LSB)
8	D1	I/O	8-bit data bus
9	D2	I/O	8-bit data bus
10	D3	I/O	8-bit data bus
11	D4	I/O	8-bit data bus
12	D5	I/O	8-bit data bus
13	D6	I/O	8-bit data bus

14	D7	I/O	8-bit data bus(MSB)
15	A	-	+5V for backlight
16	K	-	Ground

The below table shows frequently used LCD command codes.

Code(hex)	Description
01	Clear display screen
06	Increment cursor (right shift)
0A	Display off , cursor on
0C	Display on , cursor off
0F	Display on , cursor blinking
80	Force the cursor to beginning of 1st line
C0	Force the cursor to beginning of 2nd line

38	2 lines and 5*7 matrix

ADC0804 IC:

The ADC0804 IC is an 8-bit parallel ADC in the family of the ADC0800 series from National Semiconductor. It works with +5 volts and has a resolution of 8bits. The step size and Vin range varies for different values of Vref/2. The table below shows the relation between Vref/2 and Vin range.

Vref/2 (V)	Vin (V)	Step size (mV)
open	0 to 5	19.53
2.0	0 to 4	15.62
1.5	0 to 3	11.71
1.28	0 to 2.56	10

In our case Vref/2 is connected to 1.28 volts, so step size is 10mV. For ADC0804 step size is calculated as (2 * Vref / 2) / 256.

Following formula is used to calculate output voltage:

Dout = Vin / step size

Where Dout is digital data output in decimal, Vin = analog input voltage and step size (resolution) is the smallest change. Learn more about ADC0804 here, also check interfacing of ADC0808 with 8051.

LM35 Temperature Sensor:

The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. This device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55°C to 150°C temperature range. Lower cost is assured by trimming and calibration at the wafer level.

The famous LM35 linear temperature sensor is very useful to sense ambient/environment temperature. It is produced by National Semiconductor Corporation and offers a functional range from -40 degree Celsius to 150 degree Celsius. Sensitivity is 10mV per degree Celsius and the output voltage is proportional to the temperature. The output is linear to the temperature in degree Celsius and has high stability. The power supply required is 5V.

The LM35 is a temperature sensor whose output voltage is linearly proportional to Celsius temperature. The LM35 comes already calibrated hence requires no external calibration. It outputs 10mV for each degree of Celsius temperature.

LM35 sensor produces voltage corresponding to temperature. This voltage is converted to digital (0 to 256) by ADC0804 and it is fed to 8051 microcontroller. 8051 microcontroller converts this digital value into temperature in degree Celsius. Then this temperature is converted into ascii form which is suitable for displaying. This ascii values are fed to 16*2 lcd which displays the temperature on its screen. This process is repeated after specified interval.

Calculation

The output of the LM 35 is 10 mV (0.01 volts) per degree Celsius. This means at 20°C we will get 20 x 0.01 = 200 mV, or 0.2 volts.

At 100°C we will get 100 x 0.01 = 1.0 volts out.

At 50°C we will get   50 x 0.01 = 0.5v

Code For Temperature Control

```
###

#include <reg51.h> //Header file for 8051 microcontroller

void delay (void);        //delay function prototype
//Defining P3.0 bit as an ADDR_C of ADC
sbit ADDR_C = P3^0;      //Defining P3.1 bit as an EN which is Enable pin to
                       //DC motor
sbit ADDR_B = P3^1; //Defining P3.2 bit as an IN2 which is input2 pin to DC
motor
sbit ADDR_A = P3^3;
sbit ALE = P3^2; //Defining P3.1 bit as an EN which is Enable pin to DC motor
sbit OE = P3^5; //Defining P3.2 bit as an IN2 which is input2 pin to DC motor
sbit START = P3^4;
sbit EOC = P3^7;
sbit RELAY = P3^6;

void main (void)
{       //main function starts here
unsigned char mybyte;
        P1=0xff; //P1 acts as an input port, reads the value of ADC
        EOC=1;
        ALE=0;
```

```
    START=0;
    OE=0;

    while (1)    //Do forever
  {

ADDR_C=0;
ADDR_B=1;
ADDR_A=1;

delay ();
                  ALE=1;
delay();
START=1;
                  delay();
ALE=0;
                 START=0;
while (EOC !=1)
{
 OE=1;
                  mybyte=P1;
 mybyte=mybyte-35;
 if (mybyte > 40)
 {
  delay();
  RELAY=1;
        delay();delay();
  }
  else
  {
   delay();
    RELAY=0;
        delay();delay();
   }
}
}//while end
}       //End of main program


void delay (void)
{       //delay function starts here
unsigned int i;       //variable defined
for (i=0;i<=600;i++);       //This for loop generates delay
}       //End of program



###
```

Code For Temperature Monitoring

```
/*this program is for displaying the temperature on 16*2 lcd display using 8051
microcontroller , LM35 sensor and ADC0804*/


#include<reg51.h>

sbit rs=P2^7; //Register Select(RS) pin of 16*2 lcd

sbit rw=P2^6; //Read/Write(RW) pin of 16*2 lcd

sbit en=P2^5; //Enable(E) pin of 16*2 lcd

sbit rd_adc=P3^0; //Read(RD) pin of ADC0804

sbit wr_adc=P3^1; //Write(WR) pin of ADC0804

sbit intr_adc=P3^2; //Interrupt(INTR) pin of ADC0804

void delay(unsigned int)  ; //function for creating delay

void cmdwrt(unsigned char); //function for sending commands to 16*2 lcd display

void datawrt(unsigned char); //function for sending data to 16*2 lcd display
void convert_display(unsigned char); //function for converting ADC value to
temperature and display it on 16*2 lcd display

void main(void) //main function

{

   unsigned char i;

   unsigned char cmd[]={0x38,0x01,0x06,0x0c,0x82};//16*2 lcd initialization
commands

   unsigned char data1[]="Temperature:";

   unsigned char value;

   P1=0xFF; //make Port 1 as input port

   P0=0x00; //make Port 0 as output port

   for(i=0;i<5;i++) //send commands to 16*2 lcd display one command at a time

   {

      cmdwrt(cmd[i]); //function call to send commands to 16*2 lcd display

  delay(1);

   }

   for(i=0;i<12;i++) //send data to 16*2 lcd display one character at a time

   {

      datawrt(data1[i]);  //function call to send data to 16*2 lcd display

  delay(1);

   }

   intr_adc=1; //make INTR pin as input

   rd_adc=1;    //set RD pin HIGH

   wr_adc=1; //set WR pin LOW

   while(1)    //repeat forever
   {

  wr_adc=0; //send LOW to HIGH pulse on WR pin

  delay(1);

  wr_adc=1;

  while(intr_adc==1); //wait for End of Conversion

  rd_adc=0; //make RD = 0 to read the data from ADC0804

  value=P1; //copy ADC data

  convert_display(value); //function call to convert ADC data into temperature and
display it on     16*2 lcd display

  delay(1000);  //interval between every cycles

  rd_adc=1;   //make RD = 1 for the next cycle

   }

}

void cmdwrt (unsigned char x)

{

   P0=x;  //send the command to Port 0 on which 16*2 lcd is connected

   rs=0;  //make RS = 0 for command

   rw=0;  //make RW = 0 for write operation

   en=1;  //send a HIGH to LOW pulse on Enable(E) pin to start commandwrite
operation

   delay(1);

   en=0;

}

void datawrt (unsigned char y)

{
   P0=y; //send the data to Port 0 on which 16*2 lcd is connected

   rs=1; //make RS = 1 for command

   rw=0; //make RW = 0 for write operation

   en=1; //send a HIGH to LOW pulse on Enable(E) pin to start datawrite
operation

   delay(1);

   en=0;

}

void convert_display(unsigned char value)

{

  unsigned char x1,x2,x3;

  cmdwrt(0xc6);  //command to set the cursor to 6th position of 2nd line on 16*2
lcd

  x1=(value/10); //divide the value by 10 and store quotient in variable x1

  x1=x1+(0x30); //convert variable x1 to ascii by adding 0x30

  x2=value%10; //divide the value by 10 and store remainder in variable x2

  x2=x2+(0x30); //convert variable x2 to ascii by adding 0x30

  x3=0xDF; //ascii value of degree(°) symbol

  datawrt(x1);  //display temperature on 16*2 lcd display

  datawrt(x2);

  datawrt(x3);

  datawrt('C');

}

void delay(unsigned int z)

{
  unsigned int p,q;
  for(p=0;p<z;p++)    //repeat for 'z' times
  {
    for(q=0;q<1375;q++);   //repeat for 1375 times
  }
}
```

Code For Proteus Simulation

```
//LCD Module connection
sbit LCD_RS at P2_0_bit;
sbit LCD_EN at P2_1_bit;
sbit LCD_D4 at P2_2_bit;
sbit LCD_D5 at P2_3_bit;
sbit LCD_D6 at P2_4_bit;
sbit LCD_D7 at P2_5_bit;
//End of LCD Module connection

// ADC pin specification
sbit cs at P3_0_bit;
sbit rd at P3_1_bit;
sbit wr at P3_2_bit;
sbit intr at P3_3_bit;
sbit heat at P0_0_bit;
sbit cool at P0_1_bit;
sbit fan at P2_6_bit;
// end of ADC pin specification

char count=0;
char display[]="000.00";
char txt1[]="Fan OFF";
char txt12[]="Fan ON";
char adc_read=0,value=0;

void conversion()
```

```
{
cs=0;
wr=0;
delay_ms(2);
cs=1;
wr=1;
while(intr==1);
cs=1;
rd=1;
delay_ms(2);
cs=0;
rd=0;
}
void print()
{
display[2]=count%10+48;
display[1]=(count/10)%10+48;
display[0]=(count/100)%10+48;
display[4]=(value/1000)%10+48;
display[5]=(value/10000)%10+48;
lcd_out(1,7,display);
lcd_chr(1,14,223);
lcd_chr(1,15,'C');
}

void main()
{
Lcd_Init();                    //Initializes Lcd
Lcd_Cmd(_LCD_CLEAR);           // Clear LCD Screen
Lcd_Cmd(_LCD_CURSOR_OFF);     // LCD Cursor OFF
Lcd_out(1,1,"Temp");
fan=0
delay_ms(5000);
Lcd_Cmd(_LCD_CLEAR);
Lcd_out(1,1,"Temp:  ");
fan=0
while(1)
{
conversion();
count=p1;
count=count*1.95;
print();

if(count>0&&count<35)
{
cool=fan=0;
Lcd_out(2,5,txt1);
}else
{
cool=1;
}

if(count>35&&count<151)
{
heat=0;
fan=0;
Lcd_out(2,5,txt2);
}else
{
heat=1;
}


}
}
```