VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF Electronics And Communication Engineering (SENSE)

Submitted for the course:

Microcontrollers and its Applications (ECE 3003)

"DIGITAL THERMOMETER USING 8051 MICRO CONTROLLER"

PROJECT REPORT

By

19BEC0864 Sankha patra
19BEC0254 Padmini Reethi
18BEC0910 M Guru Maheswar Reddy

Name of the faculty: Dr. Padmini T N
Slot:L19+20

TABLE OF CONTENTS:

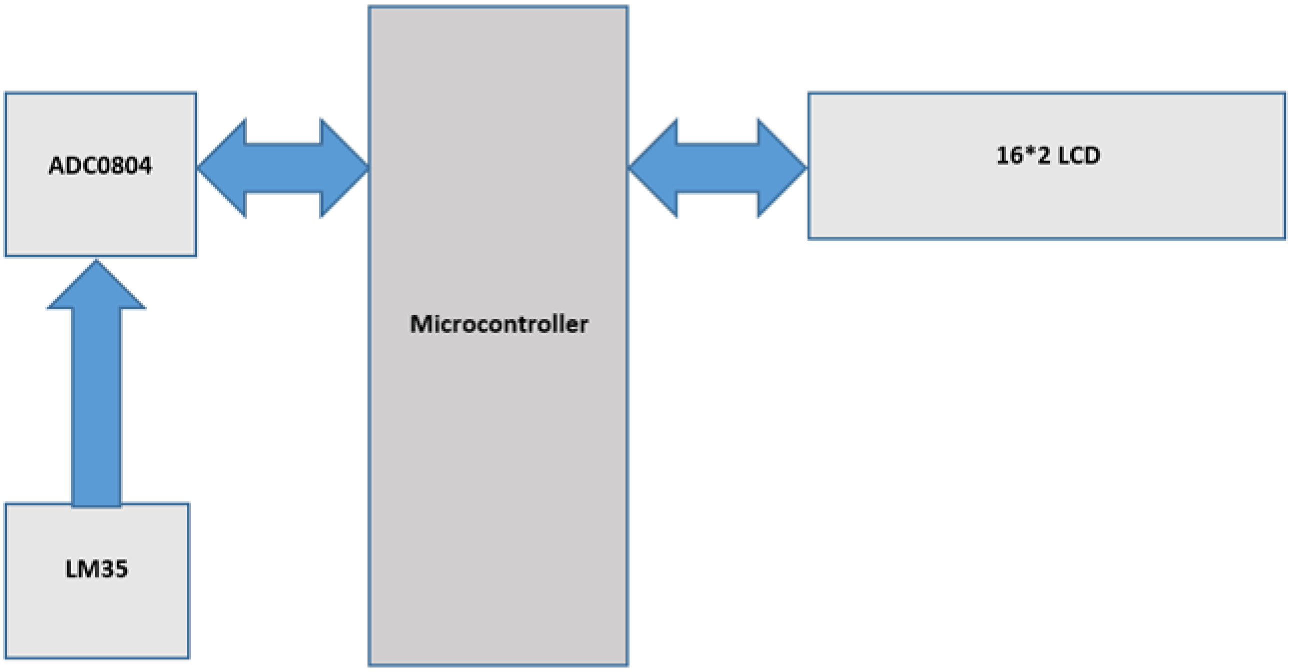
ABSTRACT

- The objective of this project is to design a digital thermometer using 8051 microcontroller.
- Digital thermometer displays the ambient temperature through a LCD display. It consists of two sections.
- One is that which senses the temperature. This is a temperature sensor LM 35.
- The other section converts the temperature value into a suitable number in Celsius scale which is done by the ADC0808.

Introduction

- This is about 0-100°C digital thermometer with 1°C resolution using 8051. The circuit is based on LM35 analog temperature sensor, ADC0808 and AT89c51 microcontroller. LM35 is an analogue temperature sensor IC which can measure a temperature range of -55 to 150°C. Its output voltage varies 10mV per °C change in temperature.
- For example, if the temperature is 32°C, the output voltage will be 32 x 10mV = 320mV. ADC 0804 is used to convert the analogue output voltage of the LM35 to a proportional 8 bit digital value suitable for the microcontroller. The microcontroller accepts the output of ADC, performs necessary manipulations on it and displays it numerically on a 2 digit seven segment LED display.

Block Diagram



Oveview view of software components:

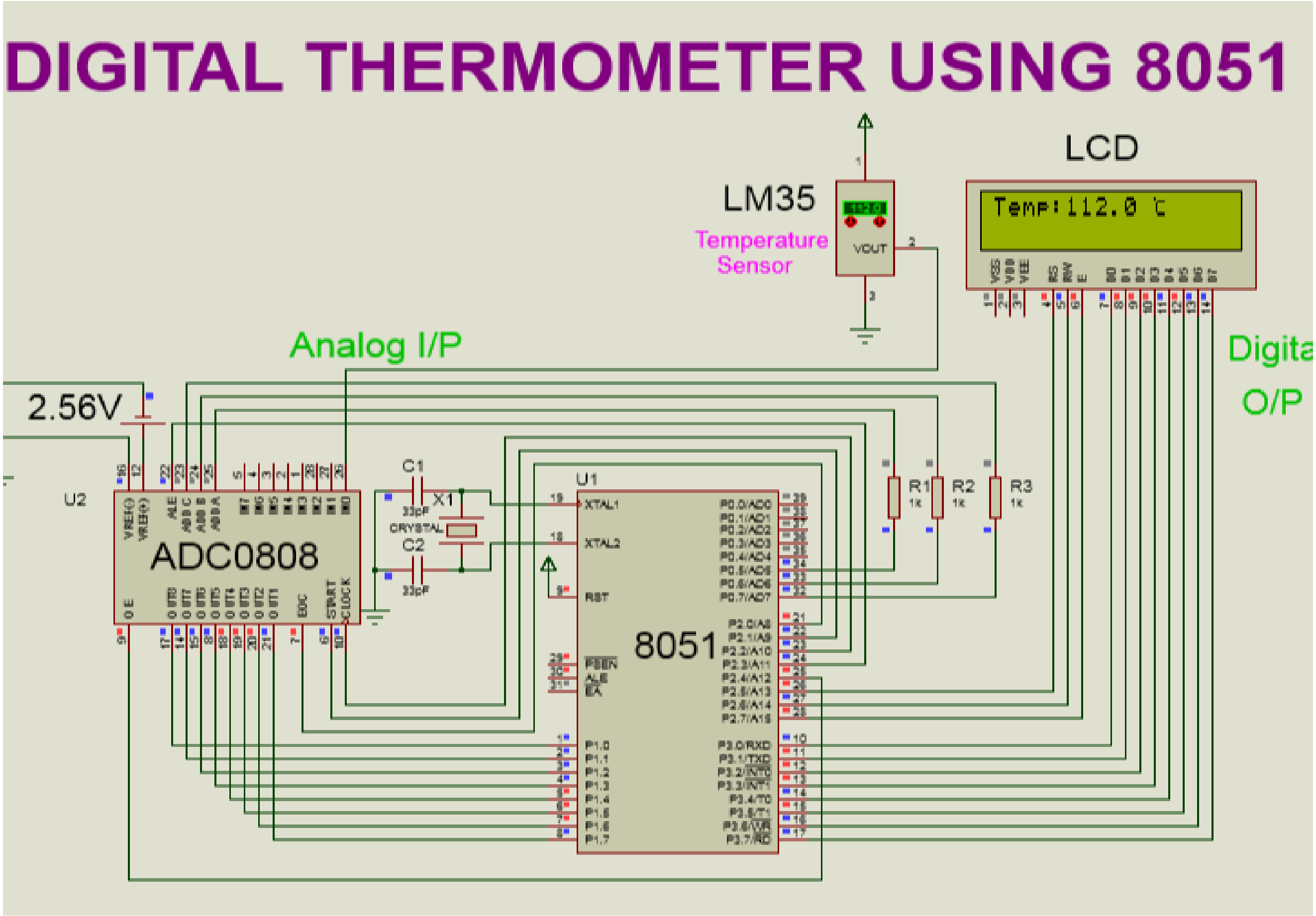
1. ADC0808 8Bit ADC:
   ADC0808 is an Analog-to-Digital Converter (ADC) with 8-Channel Multiplexer.
2. LM35 Temperature Sensor:
   LM35 is a sensor which is used to measure temperature. It provides electrical output proportional to the temperature (in Celsius).
3. LCD 16x2 Display Module:
   LCD16x2 has two lines with 16 character in each line. LCD16x2 is generally used for printing values and string in embedded application.
4. 8051 micro-controller:
   It is used to process all the input given by the temperature sensor

CODE WITH COMMENTS:

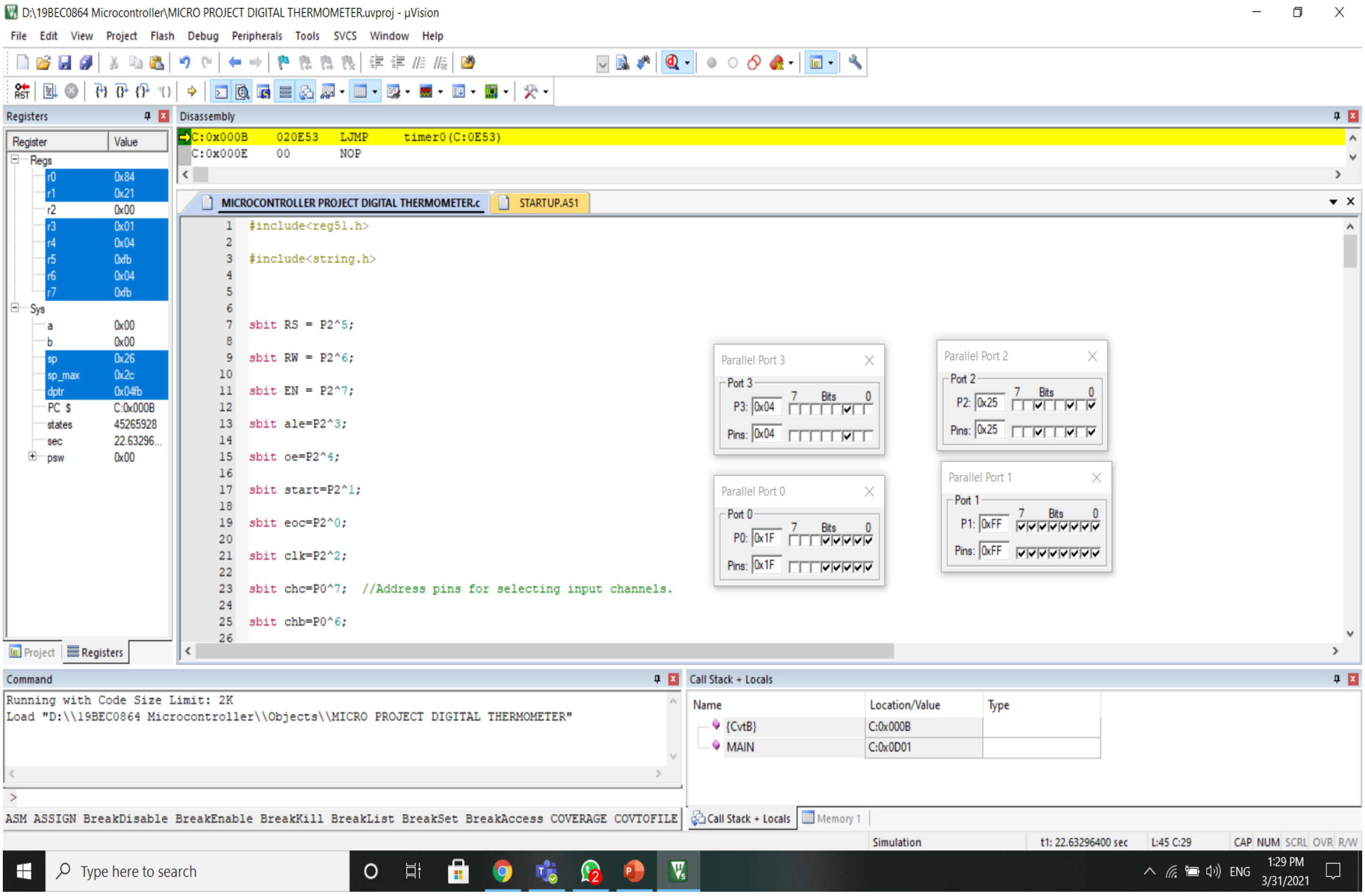
```
#include<reg51.h>
#include<string.h>
 sbit RS = P2^5;
sbit RW = P2^6;
sbit EN = P2^7;
sbit ale=P2^3;
sbit oe=P2^4;
sbit start=P2^1;
sbit eoc=P2^0;
sbit clk=P2^2;
sbit chc=P0^7;  //Address pins for selecting input channels.
sbit chb=P0^6;
sbit cha=P0^5;


void delay(int t);
void lcd_init(void);
void lcd_command(char c);
void lcd_data(char d);
void str(char a[]);
void print( long float p);
long float k;
unsigned long int q,r,x,y,z;


void timer0() interrupt 1   // TIMER 0 interrupt ISR
{
   clk=~clk;
}


void main()            // MAIN PROGRAM
{
lcd_init();   // lcd initialization
str("MICRO PROJECT");
lcd_command(0x01);  // clear display
str("G2 SLOT");
lcd_command(0x01);  // clear display
str("Temp:");
lcd_command(96);  //custom character (°c) display
lcd_data(0x10);
lcd_data(0x07);
lcd_data(0x08);
lcd_data(0x08);
lcd_data(0x08);
lcd_data(0x08);
lcd_data(0x07);
lcd_command(0x8b);
lcd_data(4);
eoc=1;            // make eoc an input
ale=0;
oe=0;
start=0;
TMOD=0x02;    // timer 0 in mode 2
TH0=0xc2; // 15khz
IE=0x82; //  set timer 0 interrupt
TR0=1;   // start timer 0
while(1)
{
chc=0;  // select channel 0
chb=0;
cha=0;
ale=1;  // send high to low pulse on start and ale pin
start=1;
delay(1);
ale=0;
start=0;
while(eoc==1);     // wait for conversion
while(eoc==0);
oe=1;
k=P1;
lcd_command(0x85);
print(k);   // send the digital data to lcd
oe=0;
}

}

void str(char a[])  // lcd
function to display string

{

int j;

for(j=0;a[j]!='\0';j++)
{

lcd_data(a[j]);

}
}

void lcd_init(void)  // lcd
initialization

{

lcd_command(0x38);
//8 bit,2 line,5x8 dots

lcd_command(0x01);
// clear display

lcd_command(0x0f);      //
display on, cursor blinking

lcd_command(0x06);
//Entry mode

lcd_command(0x0c);
//cursor off
lcd_command(0x80); ////
force cursor to beginning
of first row

}

void lcd_command(char
c) // lcd command
function
{

P3=c;

RS=0;   // select command
registeer

RW=0;

EN=1;

delay(5);

EN=0;

delay(5);

}

void lcd_data(char d) //
lcd data function

{

P3=d;

RS=1;   //select data
register

RW=0;

EN=1;

delay(5);

EN=0;

delay(5);
}
void delay(int t)      //
delay function

{

int j;

for(j=0;j<t*1275;j++);
}
void print( long float p)  //
number display function
{
x=p*10;

if(x>=1000)

{

q=x/1000;

q=q+48;

y=(x%1000)/100;

y=y+48;

z=((x%1000)%100)/10;

z=z+48;

r=x%10;

r=r+48;

lcd_data(q);

lcd_data(y);

lcd_data(z);

lcd_data(46);      //ascii
value of point

lcd_data(r);

}

else

{

q=x/100;

q=q+48;

y=(x%100)/10;

y=y+48;

z=x%10;

z=z+48;

lcd_data(q);

lcd_data(y);

lcd_data(46); //ascii value
of point

lcd_data(z);

r=0;

lcd_data(r);

}
```

RESULTS:

CIRCUIT DIAGRAM:



1. SIMULATION IN KEIL:



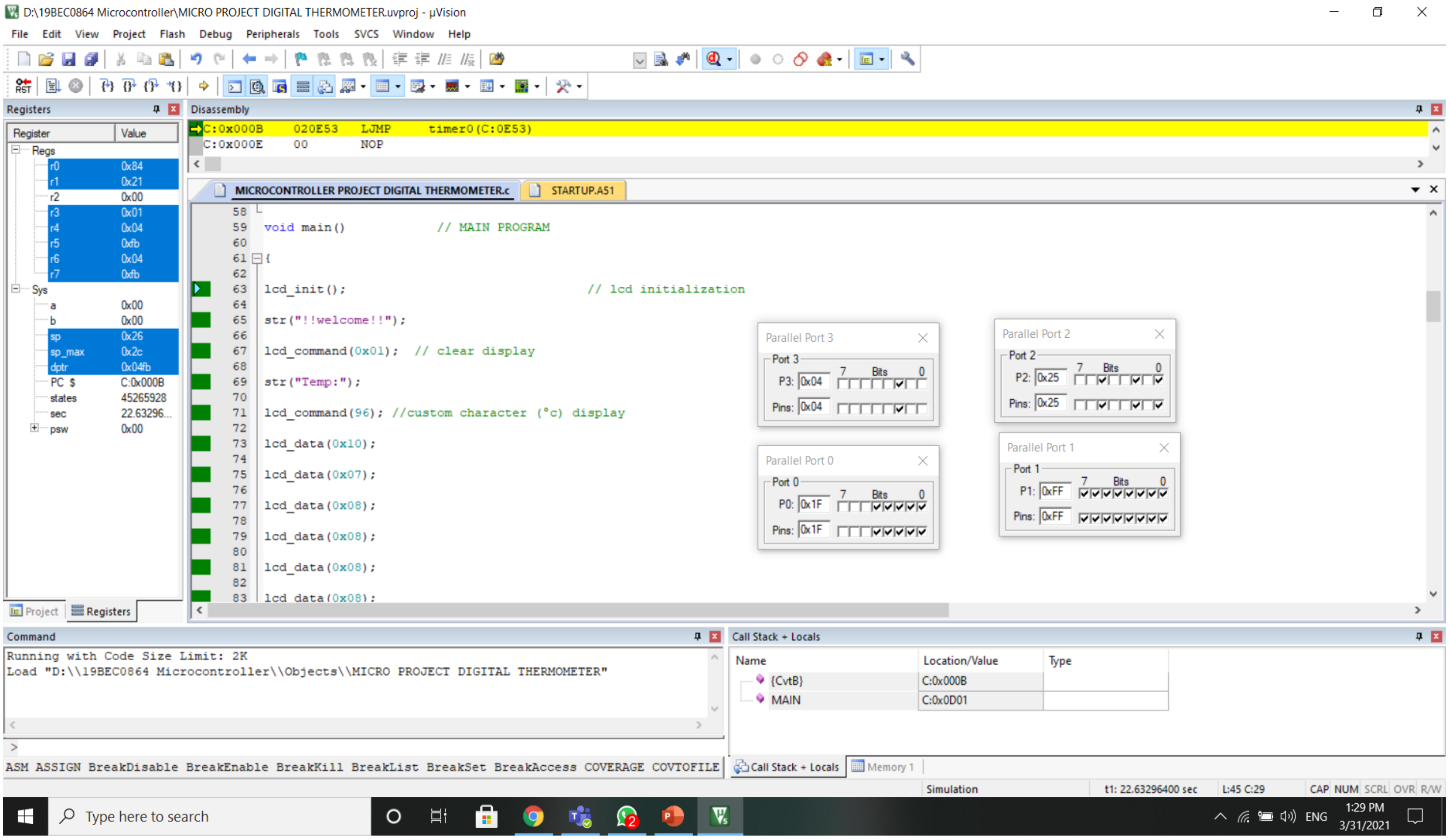
Register	Value
Regs	
r0	0x84
r1	0x21
r2	0x00
r3	0x01
r4	0x04
r5	0xfb
r6	0x04
r7	0xfb
Sys	
a	0x00
b	0x00
sp	0x26
sp_max	0x2c
dptr	0x04fb
PC $	C:0x000B
states	45265928
sec	22.63296...
psw	0x00

```
C:0x000B    020E53    LJMP     timer0(C:0E53)
C:0x000E    00        NOP
```

```
#include<reg51.h>

#include<string.h>


sbit RS = P2^5;

sbit RW = P2^6;

sbit EN = P2^7;

sbit ale=P2^3;

sbit oe=P2^4;

sbit start=P2^1;

sbit eoc=P2^0;

sbit clk=P2^2;

sbit chc=P0^7;   //Address pins for selecting input channels.

sbit chb=P0^6;

```

```
Running with Code Size Limit: 2K
Load "D:\\19BEC0864 Microcontroller\\Objects\\MICRO PROJECT DIGITAL THERMOMETER"
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
```

Name	Location/Value	Type
{CvtB}	C:0x000B	
MAIN	C:0x0D01	



Register	Value
Regs	
r0	0x84
r1	0x21
r2	0x00
r3	0x01
r4	0x04
r5	0xfb
r6	0x04
r7	0xfb
Sys	
a	0x00
b	0x00
sp	0x26
sp_max	0x2c
dptr	0x04fb
PC $	C:0x000B
states	45265928
sec	22.63296...
psw	0x00

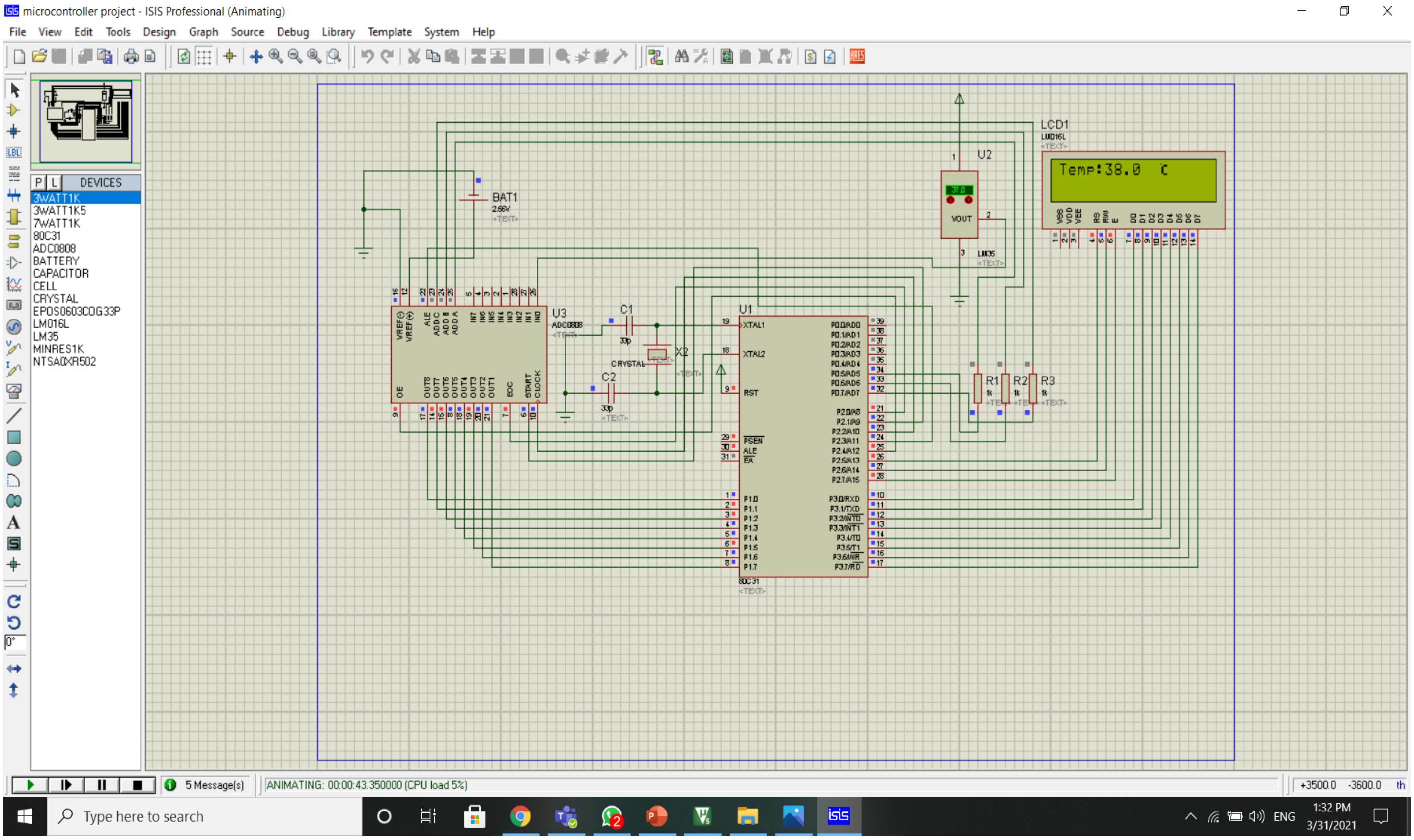
```
C:0x000B    020E53    LJMP     timer0(C:0E53)
C:0x000E    00        NOP
```

```

void main()            // MAIN PROGRAM

{

lcd_init();                          // lcd initialization

str("!!welcome!!");

lcd_command(0x01);  // clear display

str("Temp:");

lcd_command(96); //custom character (°c) display

lcd_data(0x10);

lcd_data(0x07);

lcd_data(0x08);

lcd_data(0x08);

lcd_data(0x08);

lcd_data(0x08);
```

```
Running with Code Size Limit: 2K
Load "D:\\19BEC0864 Microcontroller\\Objects\\MICRO PROJECT DIGITAL THERMOMETER"
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
```

Name	Location/Value	Type
{CvtB}	C:0x000B	
MAIN	C:0x0D01	

2. SIMULATION IN PROTEUS:



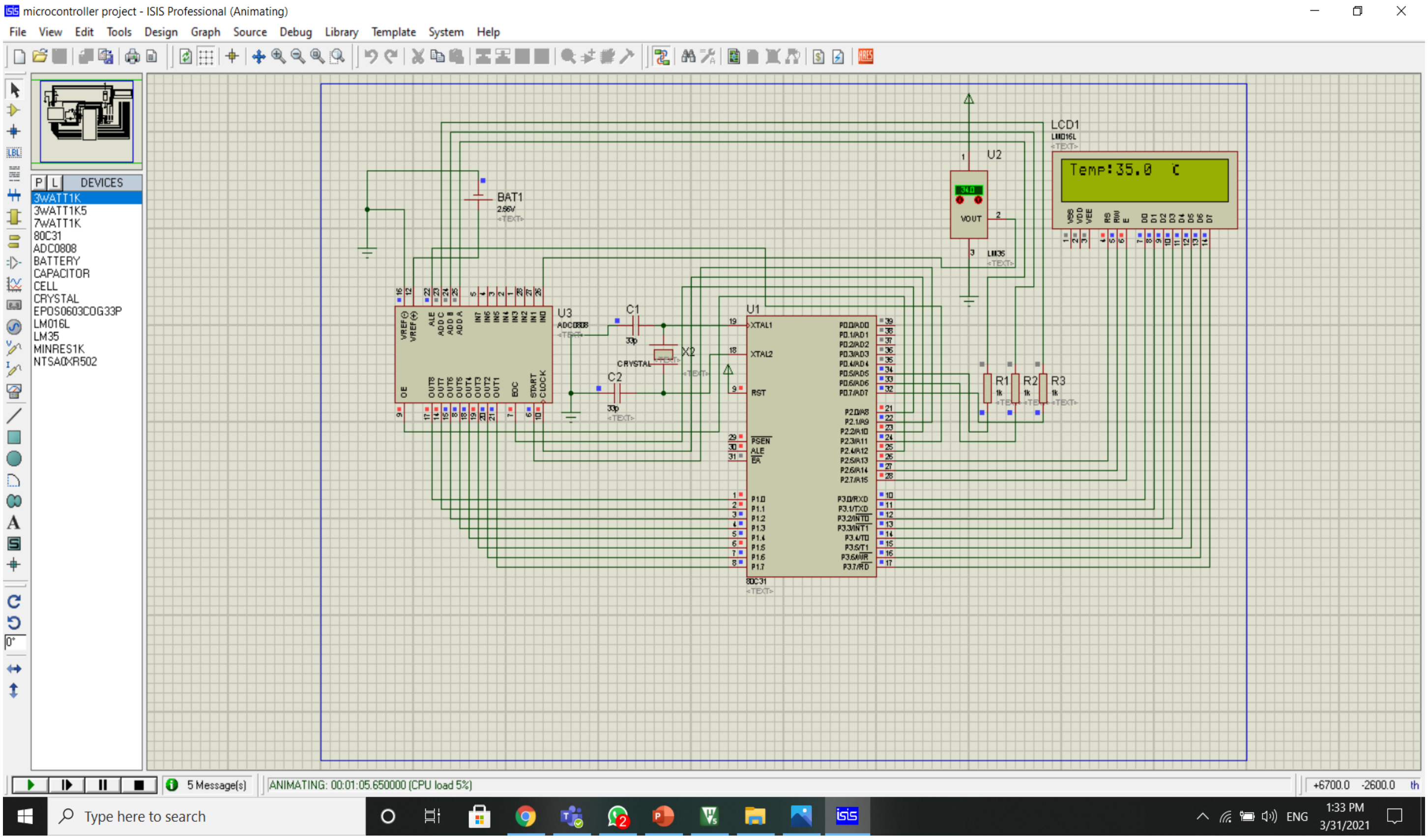
microcontroller project - ISIS Professional (Animating)

Temp:38.0 C

5 Message(s)

ANIMATING: 00:00:43.350000 (CPU load 5%)

+3500.0 -3600.0



microcontroller project - ISIS Professional (Animating)

Temp:35.0 C

5 Message(s)

ANIMATING: 00:01:05.650000 (CPU load 5%)

+6700.0 -2600.0

CONCLUSION:

In the end of the project, By using the lm35(temparature sensor), ADC0808(analog to digital convertor) and 8051 micro controller we are able to get the temperature reading in the lcd display.

REFERENCES:

- https://ieeexplore.ieee.org/abstract/document/6703317
- https://ieeexplore.ieee.org/abstract/document/5629906
- https://circuitdigest.com/microcontroller-projects/digital-thermometer-using-lm35-8051
- https://www.circuitstoday.com/thermometer-using-8051
- https://www.researchgate.net/publication/344417353_DIGITAL_THERMOMETER_BY_USING_AT89C51_MICROCONTROLLER