**D.J.SANGHVI COLLEGE OF ENGINEERING**

INDUSTRIAL SUMMER TRAINING PROGRAMME

Project: To measure and display the temperature on the lcd screen using thermistor.

By: Vatsal Makani

**Executive Summary**

This project intends to measure the temperature and display the temperature on the lcd screen using PSOC3.

**Step 1: Project specification**

The project is to measure the temperature using thermistor and display on the lcd screen.

It makes the use of voltage divider rule to calculate the value of the resistance of the thermistor and find out the temperature.

Finally display the temperature on the lcd screen. Here a PSOC3 is used.

**Step 2: Component Selection**

1. AMux
2. DAC
3. ADC
4. LCD screen

**Step 3: Pin configuration**

1. Here an lcd screen has to be connected externally.

2. Design pins should be equal to the physical pins. There should be 1 to 1 mapping.

3. VRef is connected to p5[5]. And vtherm is connected to p5[6].

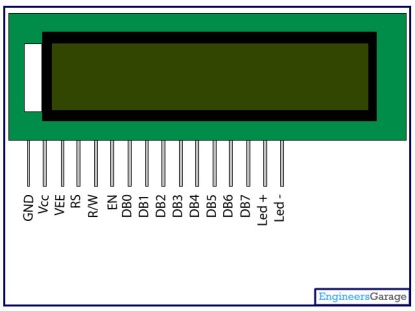
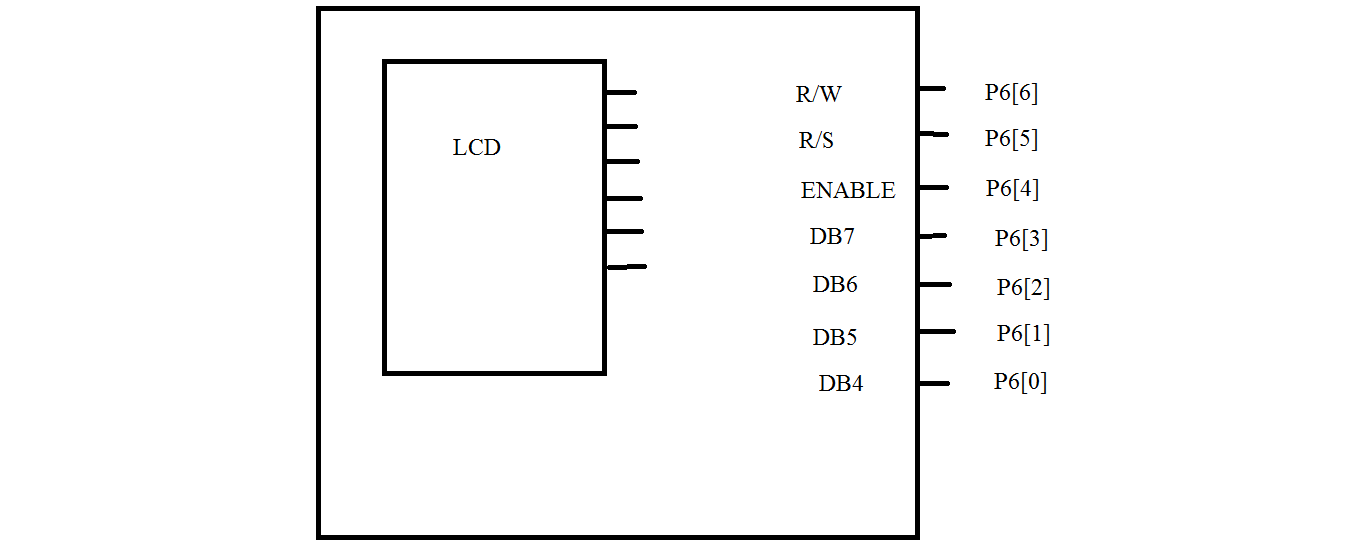
Pins of the lcd are connected to p6[6:0] ie to the port 6 of the psoc.

The pin configuration of the lcd is as follows:

Pin1: Ground

Pin2:Vcc

Pin3:contrast

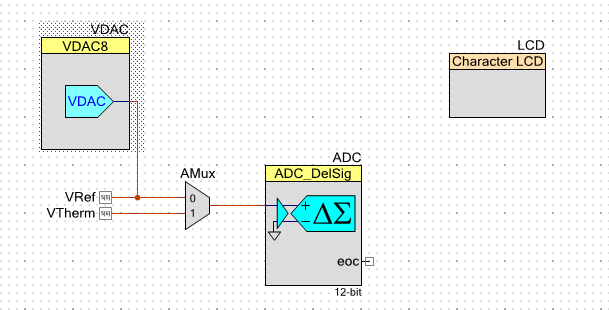
Pin4: R/s register select.

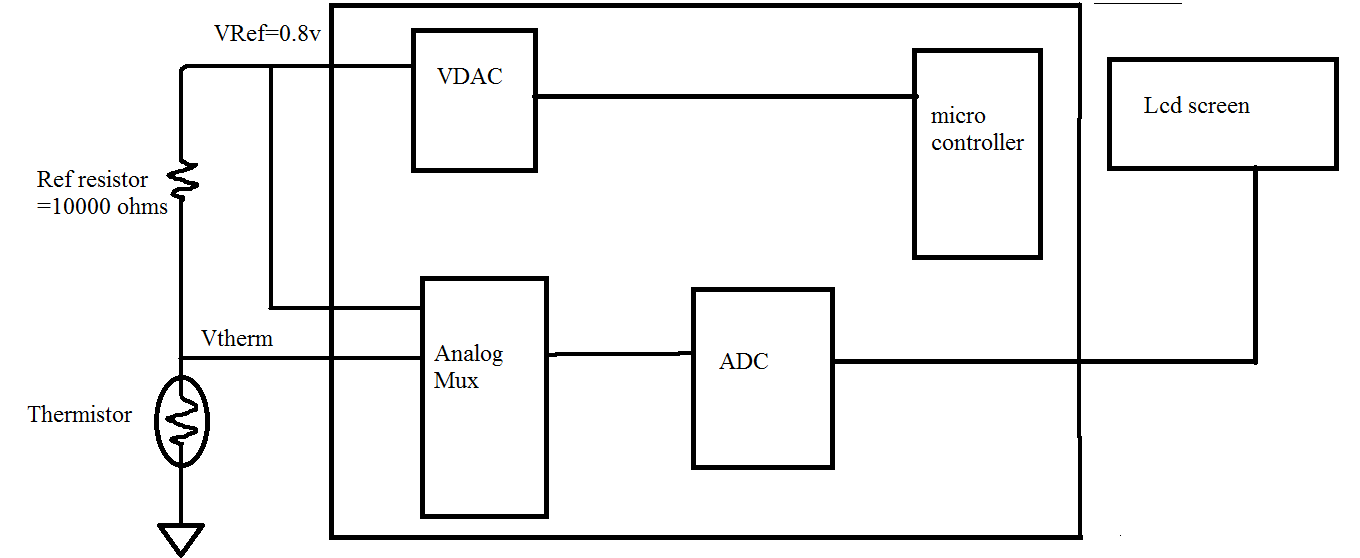
Pin5: R/W read/ Write.

Pin6: Enable.

Pin7-pin14: data bits DB0-DB7. Pin15,pin16:back led.

**Step 4: Component configuration**

****



1. Here a voltage DAC is being selected.

Output range of the dac is kept to 0 to 1.020v.

In the project high speed is not required hence slow speed is configured.

Data source of DAC is CPU or DMA.

Strobe mode is kept Register write. Here the register refers to the internal register of the dac.

If the data source is ‘DAC bus’ then the strobe mode is ‘external’.

If the outsider wants to give information to the DAC it has to send a strobe.

2. Here an analog mux is been selected. Since the voltage is analog.

In analog mux the number of the channels selected is two. One for vref and another for vtherm. The isolation is kept medium.

3. The output of the analog mux is given to the ADC.

Adc is configured as follows:

It is being used in multichannel.

Resolution is 12 bits.

Conversion rate is 10000.

Input range is 0 to 2vref.

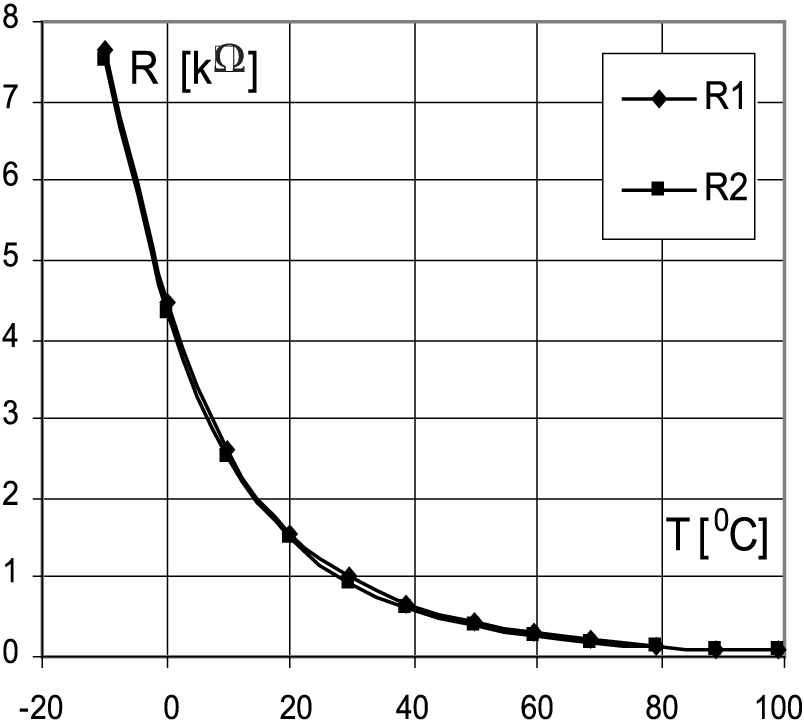
Input mode is single ended.

No of configuration is 1.

4. A normal character LCD is selected to display the characters.

**Step 5: Algorithm**

In the project NTC thermistor is being used. In this type of the thermistor as the temperature goes on increasing the resistance goes on decreasing.

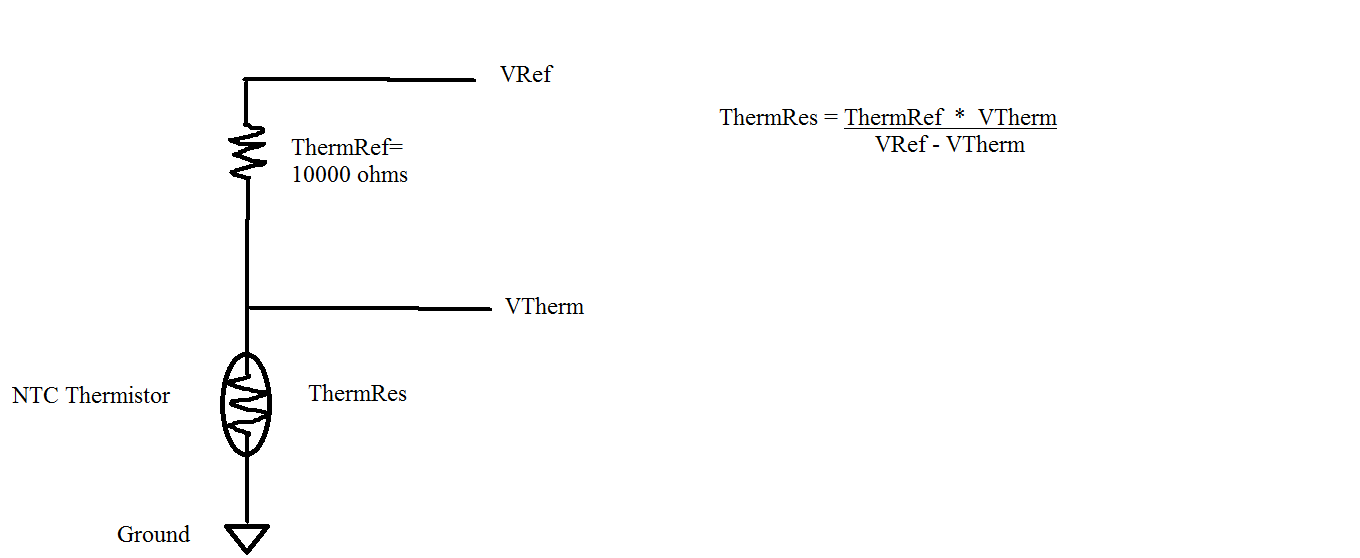


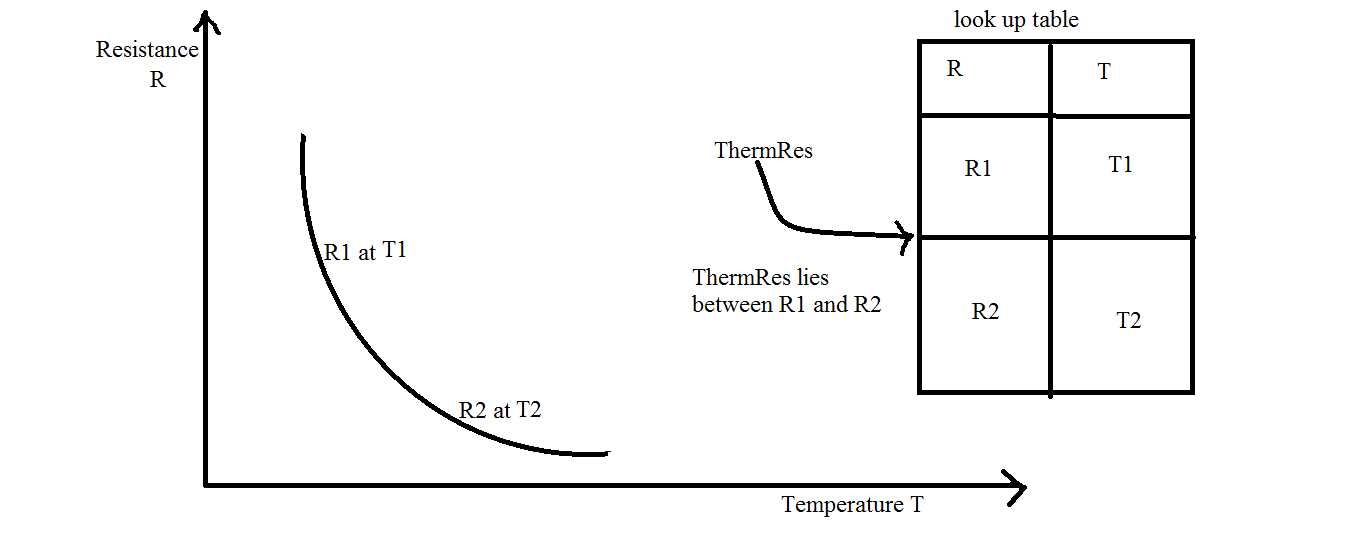
A look up table is a table which gives the values of the resistance at different temperatures. It can be noted from the look up table that as temp goes on increasing the resistance goes on decreasing.

In the project look up table gives the values of the resistance from -40 to +125 degree celcius.

The voltage divider rule is also been used to determine the resistance of the thermistor and then by looking at the look up table provided by the manufacturer, the temperature is being calculated.

The voltage divider rule is given as



****

Get R1,T1 and R2,T2.

Interpolate TherRes to get T.

Slope = (R2-R1) / (T2-T1)

T= ((TherRes - R1)/slope) + T1.

It follows the following steps:

Step 1: Decide the reference resistor value. It is set as 10kohms.

Step 2: Set the output range of the VDAC. Here voltage dac is been used. It is set

to 0 to 1.020v.

Set the voltage reference value. The reference voltage value is 0.8volts.

Step 3: Start the lcd, amux, adc.

Step 4: select the appropriate channel of the amux and get the value of the vref

and VTherm. Vref is taken again in order to minimize the error introduced

because of DAC. If there is an error in vref then error will propogate in further

calculations. In order to prevent this Vref is taken.

Step 5: Calculate the value of the Thermistor resistance as shown above and then

look in the look up table and get the value of upper and lower resistance. The

ThermRes will be present between the upper and the lower resistance.

Step 6: Calculate the slope and the final temperature.

Step 7: Print the temperature on the lcd screen.

Code:

#include <project.h>

#include "stdio.h"

const uint32 ThermTableRes[]=

{

328996,307906,288311,270096,253153,

237386,222670,208964,196194,184288,

173184,162822,153148,144112,135667,

127773,120404,111506,107048,100997**, // look up table**

95326,89988,84983,80288,75882,

71745,67874,64235,60812,57593,

54563,51698,49000,46460,44068,

41813,39690,37687,35798,34014,

32330,30737,29232,27810,26465,

25193,23990,22851,21773,20752,

19785,18868,17998,17174,16392,

15650,14946,14278,13644,13041,

12468,11923,11406,10913,10445,

10000,9575,9172,8787,8421,

8072,7739,7422,7119,6831,

6555,6293,6042,5803,5574,

5356,5147,4948,4757,4575,

4150,4233,4074,3921,3775,

3635,1501,3372,3249,3131,

3018,2910,2807,2707,2612,

2520,2432,2348,2267,2189,

2114,2042,1973,1906,1842,

1781,1722,1665,1611,1559,

1509,1460,1413,1368,1325,

1283,1243,1204,1167,1131,

1096,1063,1030,999,969,

940,912,885,859,834,

910,756,764,742,720,

700,680,661,643,625,

607,591,575,559,544,

529,515,501,488,475,

463,451,439,427,416,

406,395,385,376,366,

357,

};

**//note that the look up table starts from temperature -40**

**to 125 degree celcius. But the array starts from 0.**

int32 ThermRef=10000; /**/Reference Resister 10k**

void main()

{

int count; **//Array Index**

float T;

int32 ThermRes; **//Res. of Thermistor**

uint16 ADC\_VRef,ADC\_VTherm,ADC\_VThermistor;

float slope;

long TempUL, TempLL**; //Upper & lower limit resistors**.

char buff[15]={'\0'};

CyGlobalIntEnable; **//Needed for ADC**

VDAC\_Start();

VDAC\_SetRange(VDAC\_RANGE\_1V);  **//Output Range for DAC**

VDAC\_SetValue(200);

**// (200/255) \* Max Range (which is 1V in our case) = 200/255 \* 1.020 =0.8V**

LCD\_Start();

AMux\_Start();

ADC\_Start();

LCD\_PrintString("TEMP\_Vt\_Temperature"); **//Display on LCD**

while(1)

{

AMux\_Select(0);

ADC\_Stop(); **//To clear the garbage value of a previous channel**

ADC\_Start();

ADC\_StartConvert();

ADC\_IsEndConversion(ADC\_WAIT\_FOR\_RESULT);

ADC\_VRef=ADC\_GetResult16();

AMux\_Select(1);

ADC\_Stop();

ADC\_Start();

ADC\_StartConvert();

ADC\_IsEndConversion(ADC\_WAIT\_FOR\_RESULT);

ADC\_VTherm=ADC\_GetResult16();

ThermRes=(((int32)(ADC\_VTherm)\*ThermRef)/((int32)(ADC\_VRef-ADC\_VTherm)));

for(count=0;ThermTableRes[count]>=ThermRes;count++);

{

TempUL=ThermTableRes[count-1];

TempLL=ThermTableRes[count];

}

slope=TempLL - TempUL; **//1 Degree change and hence no x axis for denominator**

count = count -40-1; // **as the array starts from 0 and lup starts from -40.**

T = count + ((ThermRes-TempLL)/slope);

sprintf(buff,"%.1f",(float)T);

**//Conert the numerical temp. value of type float into character and then display it on**

**LCD.**

LCD\_Position(1,2); **// display at 2nd row and 3rd column**

LCD\_PrintString(buff);

}

}

**Step 6. Test and debug:**

The project designed was tested and was successfully implemented.

**Conclusion:**

Thermistors are widely used as [inrush current limiters](http://en.wikipedia.org/wiki/Inrush_current_limiter), temperature [sensors](http://en.wikipedia.org/wiki/Sensors), [self-resetting overcurrent protectors](http://en.wikipedia.org/wiki/Resettable_fuse), and self-regulating [heating elements](http://en.wikipedia.org/wiki/Heating_element). This project find application in non critical applications.