BY: Apala Bhatt (190911047), Ahana Agrawal (190911048), Mohammed Ghouse (190911037)

INERFACING LPC1768 WITH LM35 TEMPERATURE SENSOR USING LCD

PROJECT ABSTRACT

In this project we are going to interface an LM35 temperature sensor with LPC1768 ARM Microcontroller to accurately estimate the temperature and display that result on an 16x2 alphanumeric LCD with the help of a 12 bit ADC.

Aims of this project:

To understand software development for ARM microcontroller using embedded C language.

To design real world systems using ARM cortex-M embedded systems.

To understand interfacing circuits necessary for various applications and programming using ARM.

COMPONENT DETAILS

LPC1768

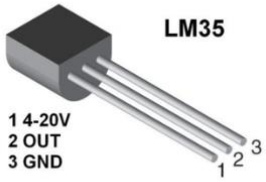
The NXP (founded by Philips) LPC1768 is an ARM 32-bit Cortex M3 Microcontroller. The code has been written in Embedded C and tested out on an ALS evaluation board.

A 12-bit internal Analog-to-Digital Converter has been used to convert the voltage value to its corresponding value in degree Celsius.

LM35 Temperature Sensor

The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly proportional to the Centigrade temperature.

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.

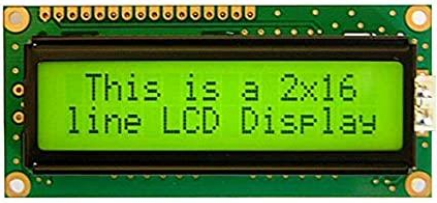


16x2 Alphanumeric LCD

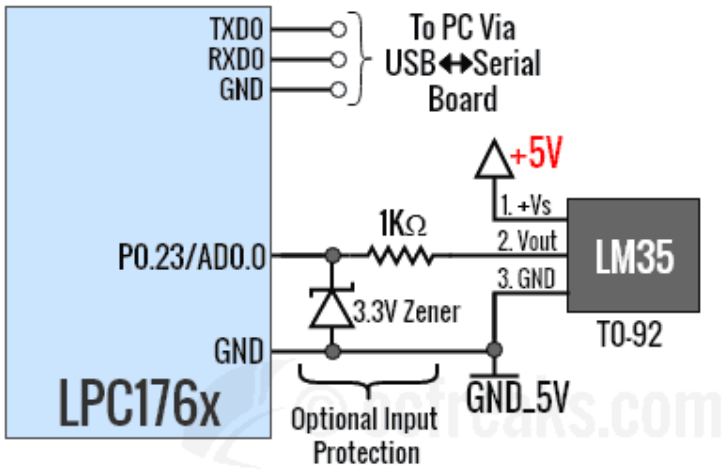
An LCD (Liquid Crystal Display) screen is an electronic display module and has a wide range of applications.

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. The 16x2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols.

This LCD has two registers, namely, Command and Data.



CIRCUIT DIAGRAM



Assuming VREFP is connected to +3.3V & VREFN is connected to 0V(GND)

CODE ALGORITHM

- Step 1:

Configure all the pins: P0.1-P0.7 as the LCD 4 bit mode, P0.8 pin as register select pin and P0.9 as the enable pin and P0.23 to function-01 for ADC.

- Step 2:

Configure the ADCR register: Ch-0 Select, Power ON, software mode and start conversion now

- Step 3:

Wait till the done bit is 1, and after that load the digital result of the conversion of the analog voltage of temperature sensor into temp variable

- Step 4:

Extract this result from temp by masking the value and left shifting the data by 4 bits

- Step 5:

In the final variable store this result of temp by dividing it by 12.41 because,

Resolution of ADC = 3.3V/(2^{12}) = 0.805mV

For every degree Celsius LM35 provides 10mV of charge

Hence ADC value is divided by 12.41 to get temperature

- Step 6:

Convert this reading into string to display it on LCD and store it on the variable temper

- Step 7:

Display string on LCD serially

EMBEDDED C CODE

```
#include<LPC17xx.h>
#include <stdio.h>
unsigned int i,j,temp;
float final;
char temper[20];
unsigned long LED =0x00000010;
#define rsctrl 0x00000100 //P0.8
#define enctrl 0x00000200 //P0.9 enable LCD
#define dtctrl 0x000000f0 //P0.4-7 4 bit mode
void lcd_init();
void wr_cn();
void clr_disp();
```

```
void delay();
void lcd_com();
void wr_dn();
void lcd_data();
void clear_ports();
void lcd_puts(unsigned char *);
unsigned int i,temp1=0,temp2=0;
int main(void)
{
    SystemInit();
    SystemCoreClockUpdate();
    LPC_PINCON->PINSEL1|=(1<<14); //configuring pin P0.23 to FUNCTION-01
    while(1)
    {
        LPC_ADC->ADCR=(1<<0)|(1<<21)|(1<<24); //ch-0 SEL, Power ON, and Start conversion NOW
        while(((temp=LPC_ADC->ADDR0)&(1<<31))==0); //wait till done bit is 1
        temp=LPC_ADC->ADDR0;
        temp&=0xFFF0; //extracting the digital result
        temp>>=4;  //12 bit data
        final=(float)(temp/12.41);
        sprintf(temper,"%3.2fC",final); //convert reading into string to display on LCD
        lcd_init();
        temp1=0x80;
        lcd_com();
        delay(800);
        lcd_puts(&temper[0]);
    }
}

void lcd_init()
{
    LPC_PINCON->PINSEL0=0;
    LPC_GPIO0->FIODIR|=dtctrl;
    LPC_GPIO0->FIODIR|=rsctrl;
    LPC_GPIO0->FIODIR|=enctrl; //register select, enable and data bits made output
    clear_ports();
    delay(3200);
    for(i=0;i<3;i++)
    {
        temp2=(0x30);
        wr_cn();
```

```
        delay(30000);
    }
    temp2=(0x20);
    wr_cn();
    delay(30000);
    temp1=0x28;
    lcd_com();
    delay(30000);
    temp1=0x0c;
    lcd_com();
    delay(800);
    temp1=0x06;
    lcd_com();
    delay(800);
    temp1=0x01;
    lcd_com();
    delay(10000);
    temp1=0x80;
    lcd_com();
    delay(800); //going through all the commands
}

void lcd_com()
{
    temp2=temp1&0xf0;
    temp2=temp2;
    wr_cn();
    temp2=temp1&0x0f;
    temp2=temp2<<4;
    wr_cn();
    delay(1000);
}

void wr_cn()
{
    clear_ports();
    LPC_GPIO0->FIOPIN=temp2;
    LPC_GPIO0->FIOCLR=rsctrl;
    LPC_GPIO0->FIOSET=enctrl;
    delay(25);
    LPC_GPIO0->FIOCLR=enctrl;
```

```
}

void lcd_data()
{
   temp2=temp1&0xf0;
   temp2=temp2;
   wr_dn();
   temp2=temp1&0x0f;
   temp2=temp2<<4;
   wr_dn();
   delay(1000);
}

void wr_dn()
{
   clear_ports();
   LPC_GPIO0->FIOPIN=temp2;
   LPC_GPIO0->FIOSET=rsctrl;
   LPC_GPIO0->FIOSET=enctrl;
   delay(25);
   LPC_GPIO0->FIOCLR=enctrl;
}

void delay(unsigned int r1)
{
   unsigned int r;
   for(r=0;r<r1;r++);
}

void clr_disp()
{
   temp1=0x01; lcd_com();
   delay(10000);
}

void clear_ports()
{
   LPC_GPIO0->FIOCLR|=rsctrl;
   LPC_GPIO0->FIOCLR|=enctrl;
   LPC_GPIO0->FIOCLR|=dtctrl;
}
```

```
void lcd_puts(unsigned char *buff)
{
   unsigned int i=0;
   while(buff[i]!='\0')
   {
      temp1=buff[i];
      i++;
      lcd_data();
   }
}
```

EXECUTION STEPS

- Connect leftmost side of the flat side of the LM35 to the +5V of the interfacing board
- Make a ground connection to the other side of the breadboard with the rightmost side and connect the 1KΩ resistor to the V_{out} middle pin of LM35
- Connect zener diode serially to the resistor and ground on the other side
- Take the serial output of the zener and resistor line to the FRC cable
- Connect this FRC to CNA3
- Connect 2 more FRCs to the LCD display
- Make a new Keil µVision project with the startup files and code file under Source Group
- Build the target, this will create a Hex file
- Open Flash Magic and browse the Hex file just created
- Click on start button to get the code running