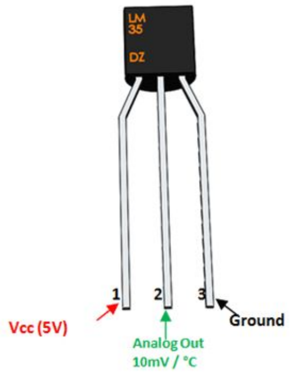
# EMBEDDED SYSTEMS ASSIGNMENT: INTERFACING LM35 TO 8051 MICRO-CONTROLLER

SUBMITTED BY:
DEEPANKAR ACHARYYA
CSB17017

**Question:**
With the help of a neat diagram, show how LM35 is interfaced with the 8051 micro-controller. WAP in Assembly/C to read in temperature data from LM35 by the 8051 via the ADC and then display the data on 7-segment LEDs or an LCD. Explain each line of code.

**Solution:**

**LM35 sensor :**
LM35 is a temperature sensor that outputs an analog signal which is proportional to the instantaneous temperature. The output voltage can easily be interpreted to obtain a temperature reading in Celsius. The advantage of LM35 over thermistor is it does not require any external calibration. It can be used to measure temperature anywhere between -55°C to 150°C.
Power the IC by applying a regulated voltage like +5V (VS) to the input pin and connected the ground pin to the ground of the circuit. If the temperature is 0°C, then the output voltage will also be 0V. There will be a rise of 0.01V (10mV) for every degree Celsius rise in temperature. The voltage can be converted into temperature using the below formulae.



$$V_{OUT} = 10\ mv/°C \times T$$

where

- $V_{OUT}$ is the LM35 output voltage
- T is the temperature in °C

## 16x2 LCD module :

It consists of 16 rows and 2 columns of 5×7 or 5×8 LCD dot matrices. It is available in a 16 pin package with backlight, contrast adjustment function, and each dot matrix has 5×8 dot resolution. The JHD162A has two built-in registers namely data register and command register. The data register is for placing the data to be displayed, and the command register is to place the commands.

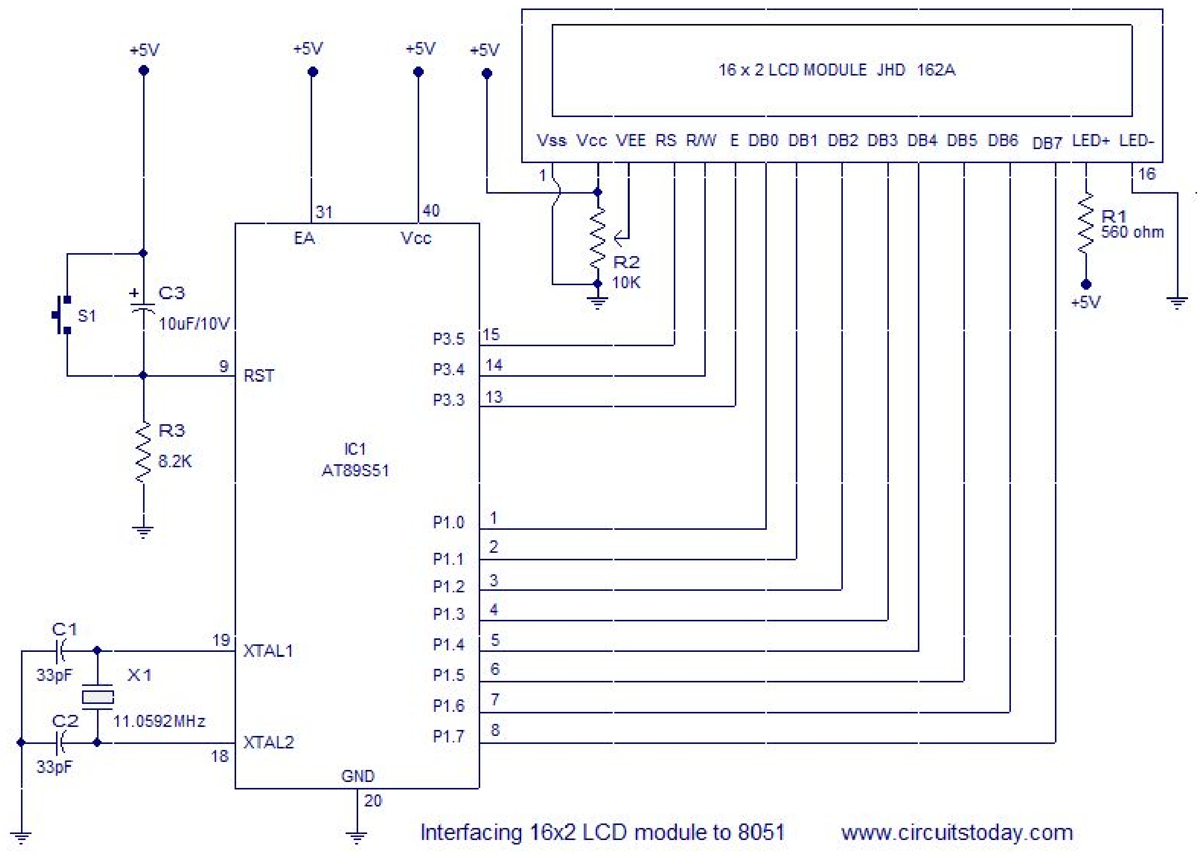
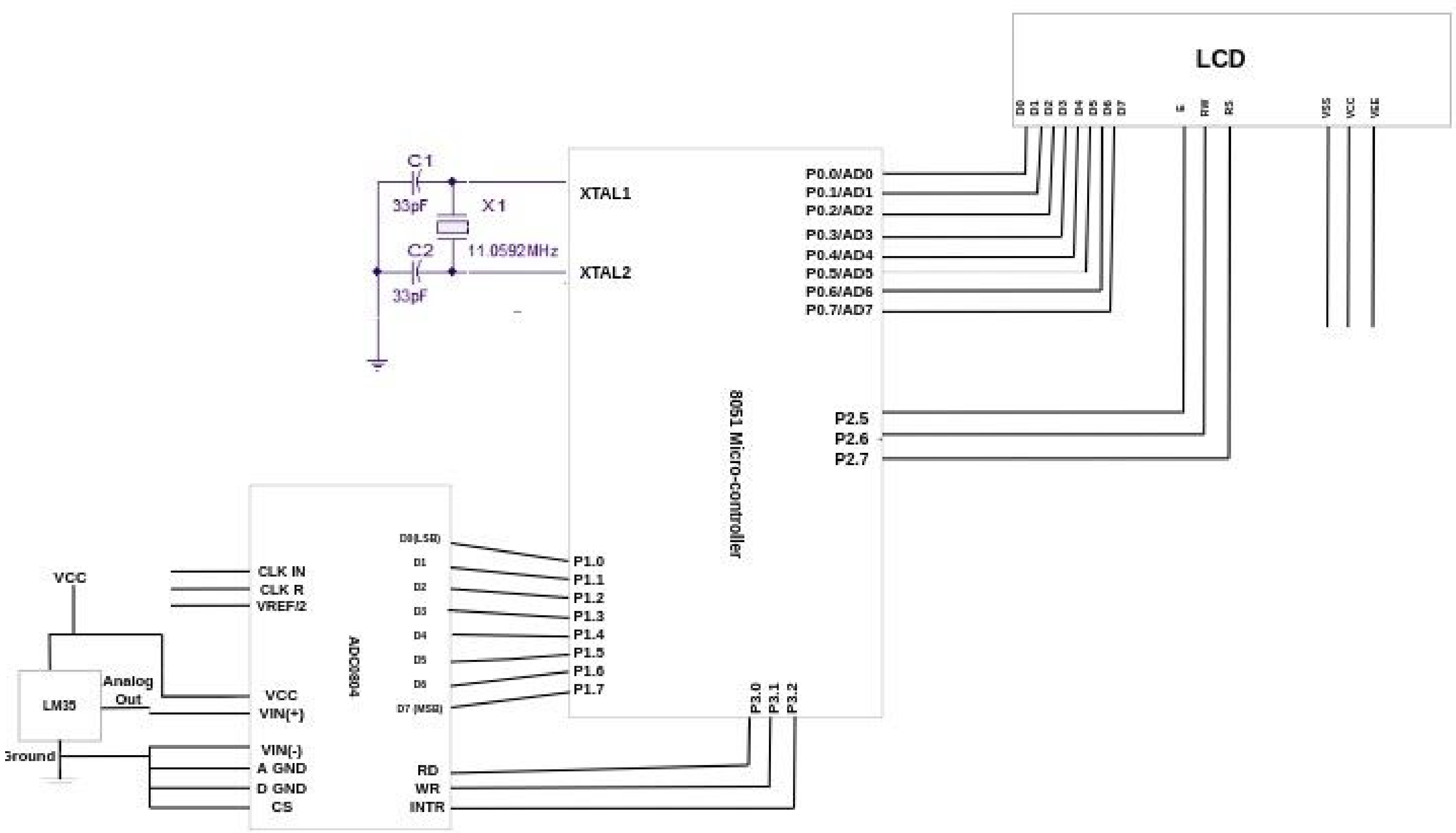


Fig: Interfacing the LCD module with 8051

NOTE: In this diagram Port 1 is connected to the LCD module, but for our purpose, we have connected Port 0 with the LCD module as shown in the next diagram. Also, the pins E, RW, RS are connected to pins 5,6,7 from Port 2.

**Fig: Interfacing LM35 with 8051 via ADC0804: (The setup for the code)**



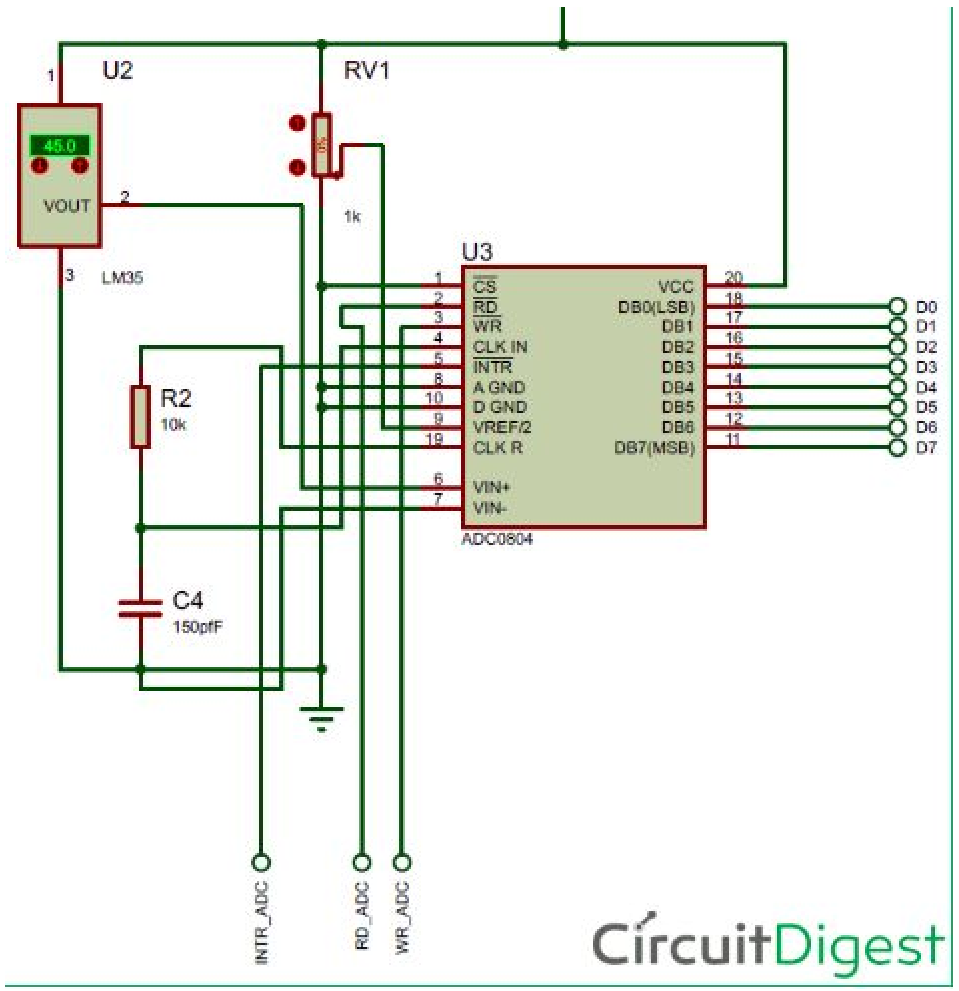


**Fig: Interfacing LM35 with ADC0804**

## C CODE PART:

```
#include<reg51.h>

//Defining the pins

//Pins for the ADC module
sbit wr=P3^1;                      // define wr pin of ADC use for writing purposes
sbit intr=P3^2;                    //defines intr pin use for sending interrupts from microcontroller
sbit rd=P3^0;                      //defines rd pin of ADC use for reading purposes

//Pins for the LCD module
sbit rs=P2^7;                                      //Register Select(RS) pin of the lcd module
sbit en=P2^5;                                      //Enable(E) pin of the lcd module
sbit rw=P2^6;                                      //Read/Write(RW) pin of the lcd module


//function for creating delay in msec
void delay(unsigned int t){
  unsigned int i,j;
  for(i=0;i<t;i++)                                //to be repeated for t times
    for(j=0;j<1275;j++);                          //to be repeated for 1275 times for each msec
}


//function to send the commands to the lcd display module
/*This function sets enable pin high; RS pin and RW pin 0*/
void send_cmd_lcd(unsigned char x){
   P0=x;             //sending the command to Port 0 on which the lcd module is connected
   rs=0;             //making RS = 0 for command
   rw=0;             //making RW = 0 for write operation
   en=1;             //send a HIGH to LOW pulse on Enable pin to start commandwrite operation
   delay(1);
   en=0;
}
```

```
//function for sending data to the lcd display module
/*This function sets enable and RS pin high;and RW pin 0*/
void write_data_lcd (unsigned char x){
   P0=x;            //sending the data to Port 0 on which the lcd module is connected
   rs=1;            //making RS = 1 for command
   rw=0;            //making RW = 0 for write operation
   en=1;            //send a HIGH to LOW pulse on Enable pin to start datawrite operation
   delay(1);
   en=0;
}


//function for converting ADC value to temperature and display it on the lcd display module
void convert_and_display(unsigned char value){

  unsigned char x1,x2,x3;
  send_cmd_lcd(0xc6);  //command to set the cursor to 6th position of 2nd line on lcd display

  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

  write_data_lcd(x1);  //display the temperature on the lcd display module
  write_data_lcd(x2);
  write_data_lcd(x3);
  write_data_lcd('C');
}



void main(){

   unsigned char i;
   unsigned char cmd[]={0x38,0x01,0x06,0x0c,0x82};            //lcd module initialization commands
/*
0x38: The command 0x38 means we are setting 8-bit mode lcd having two lines and character shape between 5×7 matrix.
0x01 : Command to clear the lcd
0x06 : Entry mode/ Increment cursor
0x0c : Display on cursor off
0x82 : address of the cg ram
*/
   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 the lcd display one command at a time
      send_cmd_lcd(cmd[i]);         //function call to send commands to 16*2 lcd display
      delay(1);                     //delay of 1msec
   }

   intr=1; //make INTR pin as input
   rd=1;    //set RD pin HIGH
   wr=1; //set WR pin LOW

   while(1){             //keep repeating
            wr=0;        //send LOW to HIGH pulse on WR pin
            delay(1);    //delay of 1ms
            wr=1;

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

            rd=0;              //make RD = 0 to read the data from ADC
            value=P1;          //read in the ADC data from port 1
            convert_and_display(value);
                               //function call to convert ADC data into temperature and display it on the lcd display
            delay(1000);       //creating a delay of 1 sec=1000msec:interval between every cycles
            rd=1;              //make RD = 1 for the next cycle
   }
 }
```