**Name :**

**Roll no :**

**Group B Lab Assignment:** 12

**Subject :**PSDL

**Title :** Interfacing PIC18FXXX to Temperature sensor

**Assignment No: 12**

**Title :** Write an Embedded C program for interfacing PIC18FXXX to Temperature sensor interfacing using ADC & display on LCD

**Aim :** To write a C program to interface PIC18F4550 to a temperature sensor (LM35) and display the temperature on LCD.

**Experimental Setup:** MicroPIC18F board, USB cable, Power supply adaptor, MPLABx IDE, PICLoader software.

**Objective:**

* To Perform interfacing of real-world input and output devices to PIC18FXXX microcontroller.
* To write and execute an Embedded C program to interface PIC18F4550 to a temperature sensor (LM35) and display the temperature on LCD.

**Theory:**

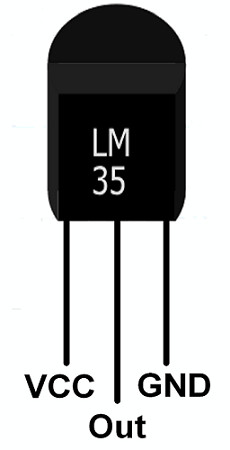
**1). LM35 Temperature Sensor Interfacing with PIC18F4550**

**1.1). LM35 Temperature Sensor**

* LM35 is a temperature measuring device having an analog output voltage proportional to the temperature.
* It provides output voltage in Centigrade (Celsius). It does not require any external calibration circuitry.
* The sensitivity of LM35 is 10 mV/degree Celsius. As temperature increases, output voltage also increases.

            E.g. 250 mV means 25°C.

* It is a 3-terminal sensor used to measure surrounding temperature ranging from -55 °C to 150 °C.
* LM35 gives temperature output which is more precise than thermistor output.

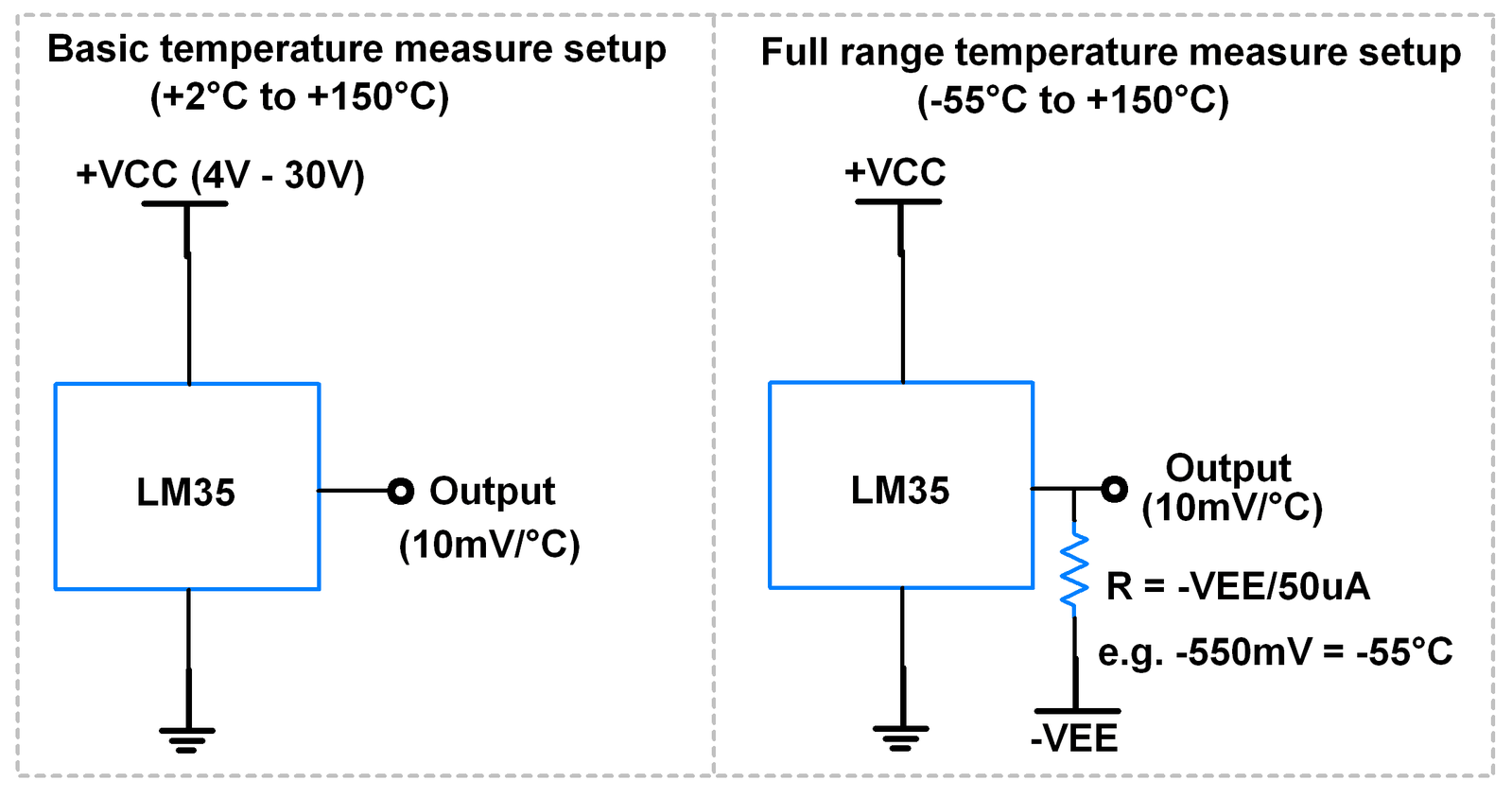


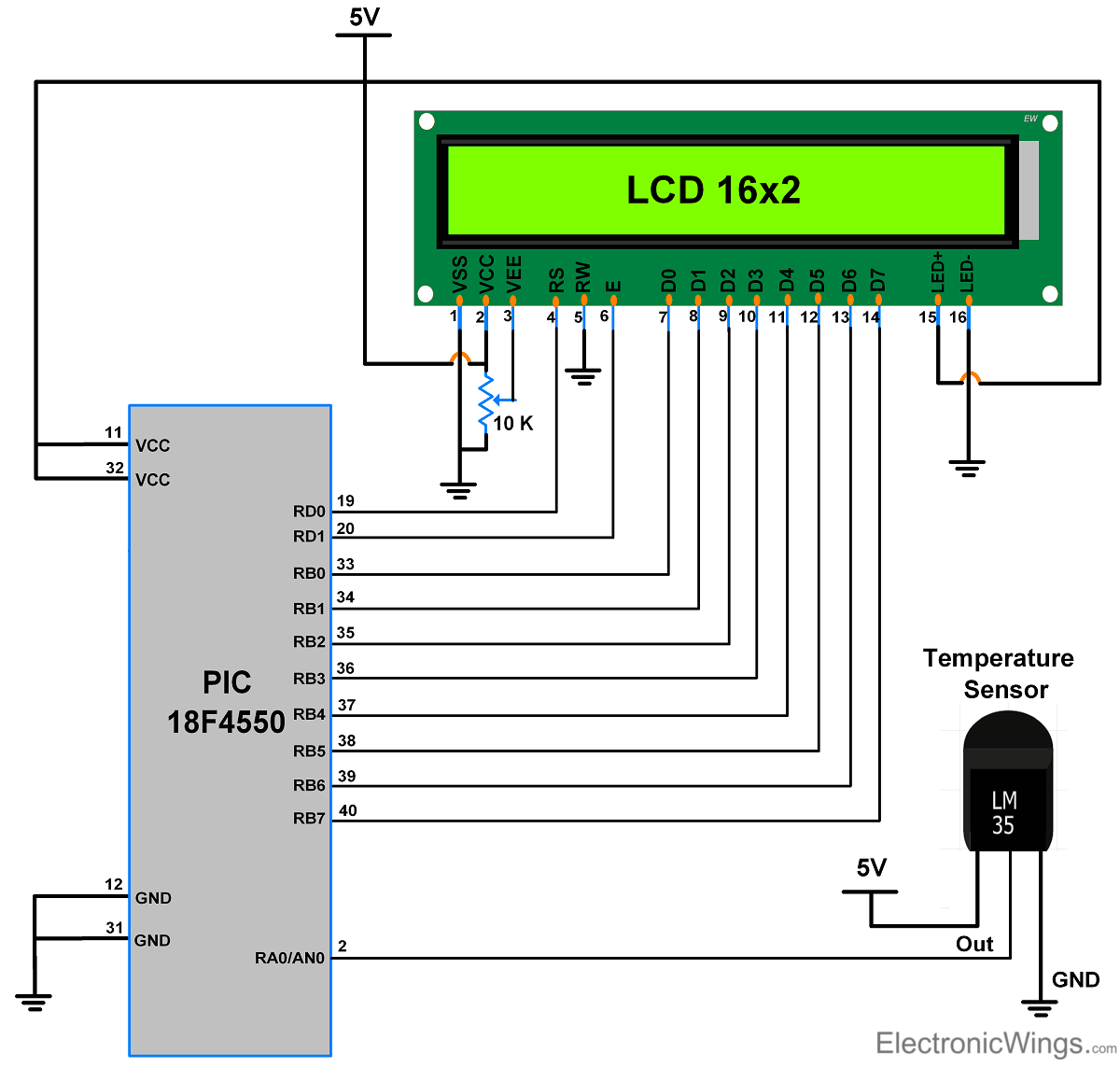
**VCC:**Supply Voltage (4V – 30V)

**Out:**It gives analog output voltage which is proportional to the temperature (in degree Celsius).

**GND:**Ground

**Application Setup**





**LM35 Temperature Sensor Interfacing with PIC18F4550**

**Procedure:**

**Step1:** Open MPLABX IDE on the PC for program development and create a new project and save it in a new folder.

**Step2:** Write the program in C language for interfacing temperature sensor (LM35) to PIC18F4550 and display result on LCD. (in program properties make sure to add the 0x800 offset)

**Step3:** Build the program and create hex file. In case of errors correct program and rebuild to create hex file.

**Step4:** Prepare the experimental setup by connecting the MicroPIC18F board to the PC using USB cable. Power ON the Board. Check for the USBtoSerial COMx allocated by the PC**.**

**Step5:** Using the PICLoader Software flash the hex file in the PIC18F4550.

**Step6:** Press reset button and execute the program.

**.**

**Source code:**

/\*

\* File: main.c

\* Author: MicroEmbedded

\*

\* Created on October 15, 2016, 4:35 PM

\*/

#include <pic18f4550.h>

#include <stdio.h>

#define LCD\_EN LATAbits.LA1

#define LCD\_RS LATAbits.LA0

#define LCDPORT LATB

unsigned char str[16];

void lcd\_delay(unsigned int time)

{

unsigned int i , j ;

for(i = 0; i < time; i++)

{

for(j=0;j<100;j++);

}

}

void SendInstruction(unsigned char command)

{

LCD\_RS = 0; // RS low : Instruction

LCDPORT = command;

LCD\_EN = 1; // EN High

lcd\_delay(10);

LCD\_EN = 0; // EN Low; command sampled at EN falling edge

lcd\_delay(10);

}

void SendData(unsigned char lcddata)

{

LCD\_RS = 1; // RS HIGH : DATA

LCDPORT = lcddata;

LCD\_EN = 1; // EN High

lcd\_delay(10);

LCD\_EN = 0; // EN Low; data sampled at EN falling edge

lcd\_delay(10);

}

void InitLCD(void)

{

ADCON1 = 0x0F;

TRISB = 0x00; //set data port as output

TRISAbits.RA0 = 0; //RS pin

TRISAbits.RA1 = 0; // EN pin

SendInstruction(0x38); //8 bit mode, 2 line,5x7 dots

SendInstruction(0x06); //entry mode

SendInstruction(0x0C); //Display ON cursor OFF

SendInstruction(0x01); //Clear display

SendInstruction(0x80); //set address to 0

}

void LCD\_display(unsigned int row, unsigned int pos, unsigned char \*ch)

{

if(row==1)

SendInstruction(0x80 | (pos-1));

else

SendInstruction(0xC0 | (pos-1));

while(\*ch)

SendData(\*ch++);

}

void ADCInit(void)

{

TRISEbits.RE2 = 1; //ADC channel 7 input

ADCON1 = 0b00000111; //Ref voltages Vdd & Vss; AN0 - AN7 channels Analog

ADCON2 = 0b10101110; //Right justified; Acquisition time 4T; Conversion clock Fosc/64

}

unsigned short Read\_Temp(void)

{

ADCON0 = 0b00011101; //ADC on; Select channel;

GODONE = 1; //Start Conversion

while(GO\_DONE == 1 ); //Wait till A/D conversion is complete

return ADRES; //Return ADC result

}

int main(void)

{

unsigned int temp;

InitLCD();

ADCInit();

LCD\_display(1,1,"Temperature:");

while(1)

{

temp = Read\_Temp();

temp = ((temp \* 500) / 1023);

sprintf(str,"%d'C ",temp);

LCD\_display(2,1,str);

lcd\_delay(9000);

}

return 0;

}

**Result:** Check if the temperature values are displayed on the LCD

**Conclusion:** Thus, we have studied C program interfacing PIC18FXXX to Temperature sensor interfacing using ADC & display on LCD.