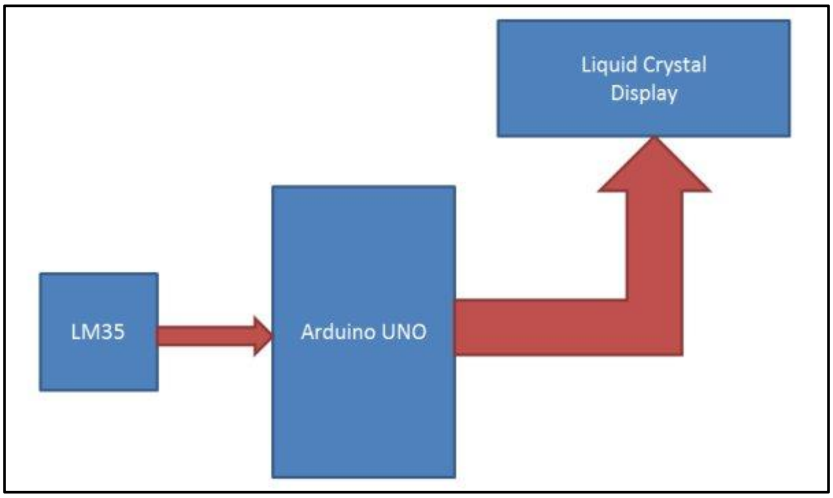
Abstract

Thermometers are useful apparatus that have been used since long time for temperature measurement. In this project we have made an Arduino based digital thermometer to display the current ambient temperature on a 16x2 LCD unit in real time. It can be deployed in houses, offices, industries etc. to measure the temperature. We can divide this Arduino based thermometer into three sections - The first section senses the temperature by using temperature sensor LM35, second section converts the temperature value into a suitable number in Celsius scale which is done by Arduino, and last part of system displays temperature on 16x2 LCD. The same is demonstrated in below block diagram.

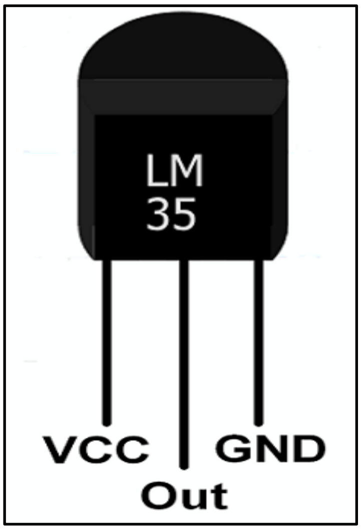


Introduction

In an era of technological advancements, the integration of microcontrollers like Arduino has revolutionized the landscape of DIY electronics projects. One such practical and widely applicable project is the creation of a digital thermometer using Arduino. This project harnesses the power of the LM35 Temperature Sensor, a versatile and accurate device capable of measuring temperature in the Centigrade scale. By interfacing the LM35 with Arduino and incorporating a 16*2 LCD display, we embark on a journey to design an efficient and user-friendly digital thermometer.

LM35

The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. 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. i.e.; 250 mV means 25°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

Specification of LM35 Temperature Sensor

- Operating Voltage: 4 V to 30 V
- Output Voltage: 10mV/°C
- Sensitivity: 10mV/°C
- Linearity Error: ±1°C (for 0°C to +100°C)
- Operating Temperature: -55°C to +150°C
- Output Impedance: 100 Ω
- Power Consumption: 60 µA (typical)
- Package Type: TO-92, TO-220, SOIC
- Output Type: Analog
- Accuracy: ±1°C (typical)

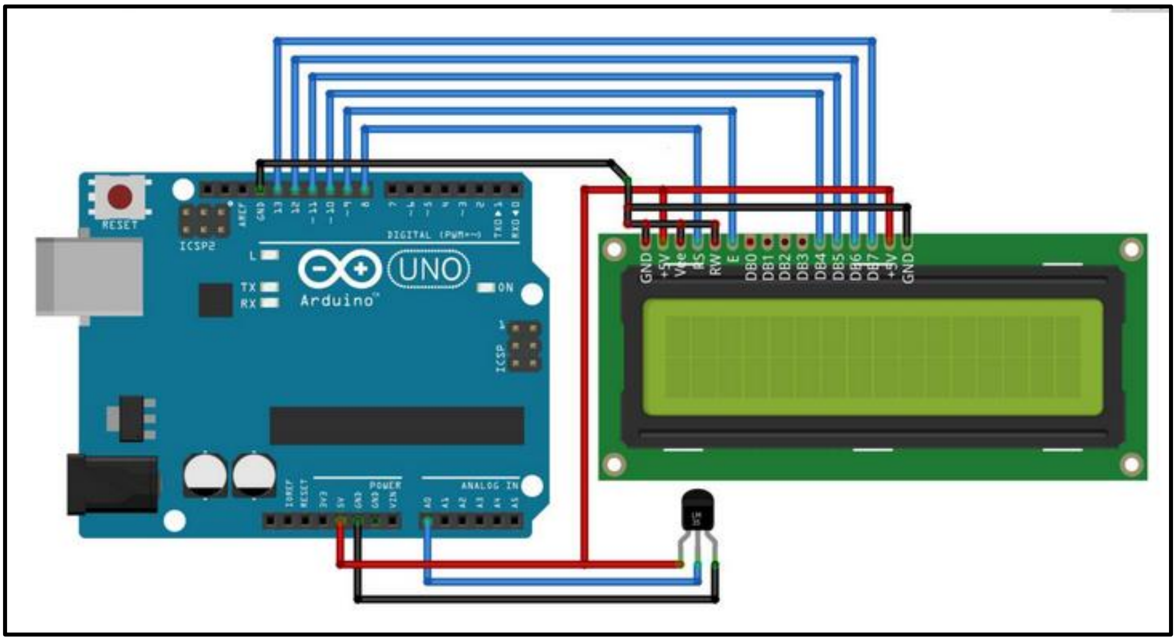
Alternate Options for LM35 Sensor

- TMP36
- DHT11
- DS18B20
- LM34
- RTD PT100

Components required

S.N.	Component Name	Description	Quantity
1	Arduino Board	Arduino Uno R3 Board Micro Controller Board	1
2	Temperature Sensor	LM35 Temperature	1
3	LCD Display	JHD162A 16x2 LCD Display	1
4	Potentiometer	10kOhm	1
5	Breadboard	GL-12 840 Points Solderless Breadboard	2
6	Connecting Wires	Jumper Cables	20

Circuit Diagram:



Working

In order to understand the working principle of the lm35 temperature sensor, we have to understand the linear scale factor. In the features of lm35, it is given to be +10 mills volt per degree centigrade. It means that with an increase in output of 10 mills volt by the sensor out pin the temperature value increases by one. For example, if the sensor is outputting 100 mills volt at vout pin the temperature in centigrade will be 10-degree centigrade. The same goes for the negative temperature reading. If the sensor is outputting -100 mills volt the temperature will be -10 degrees Celsius.

Now to building the project. Connect the 1st pin of LM35 to 5V of Arduino UNO and 3rd Pin to GND. Similarly, connect the 2nd pin to Analog input pin A0 of Arduino UNO

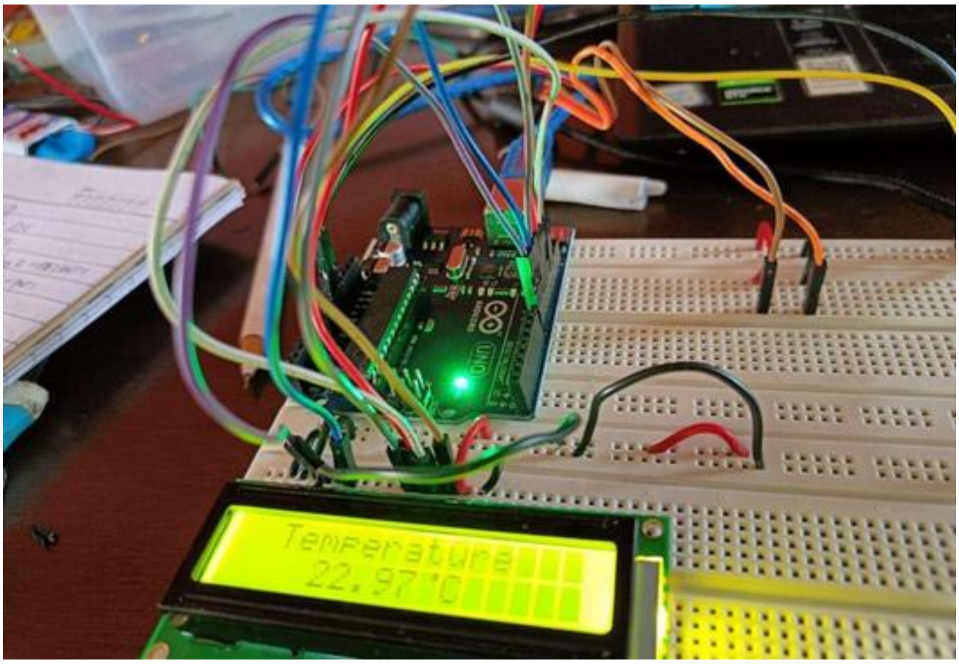
Now we will connect the 16×2 LCD to the Arduino.

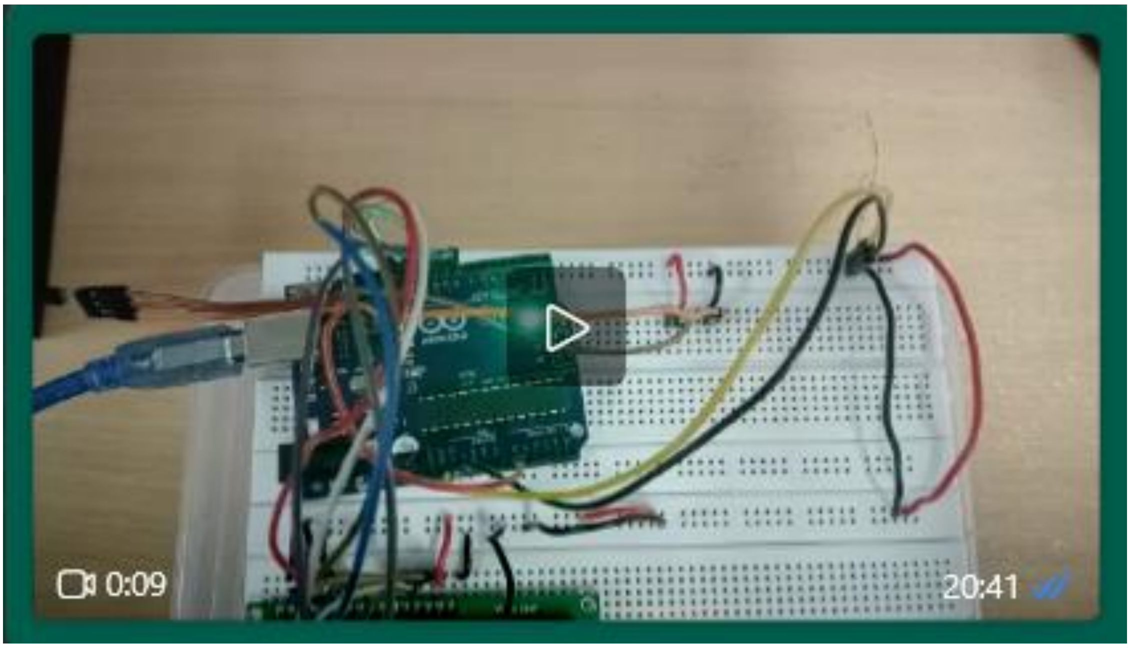
- Connect pin 1,3,5,16 of LCD to the GND.
- Connect pin 2,15 of LCD to the VCC (5V).
- Connect pin 4 of LCD to pin D8 of Arduino.
- Connect pin 6 of LCD to pin D9 of Arduino.
- Connect pin 11 of LCD to pin D10 of Arduino.
- Connect pin 12 of LCD to pin D11 of Arduino.
- Connect pin 13 of LCD to pin D12 of Arduino.
- Connect pin 14 of LCD to pin D13 of Arduino.

Code/Program:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8,9,10,11,12,13);  // Initializing LCD with pin connections
#define sensor A0  // Defining the analog input pin for the temperature sensor
byte degree[8] = {  // Creating a custom character for the degree symbol
  0b00011,
  0b00011,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00000
};



void setup() {
  lcd.begin(16,2);  // Initializing the LCD with 16 columns and 2 rows
  lcd.createChar(1, degree);  // Loading the custom degree symbol to
LCD
  lcd.setCursor(0,0);
  lcd.print(" Digital ");  // Displaying "Digital" on the first line of
LCD
  lcd.setCursor(0,1);
  lcd.print(" Thermometer ");  // Displaying "Thermometer" on the
second line of LCD
  delay(2000);  // Delay for 2000 milliseconds (2 seconds)
  lcd.clear();  // Clearing the LCD screen
}

void loop() {
  /*---------Temperature-------*/
  float reading = analogRead(sensor);  // Reading analog value from the
temperature sensor
  float temperature = reading * (5.0 / 1023.0) * 100;  // Converting
analog value to temperature in Celsius
  delay(10);  // Delay for 10 milliseconds

  /*------Display Result------*/
  lcd.clear();  // Clearing the LCD screen
  lcd.setCursor(2,0);
  lcd.print("Temperature");  // Displaying "Temperature" on the first
line of LCD
  lcd.setCursor(4,1);
  lcd.print(temperature);  // Displaying the temperature value on the
second line of LCD
  lcd.write(1);  // Displaying the custom degree symbol
  lcd.print("C");  // Displaying "C" for Celsius
  delay(1000);  // Delay for 1000 milliseconds (1 second)
}
```

Project





(Video Attached)

References

[1] https://duino4projects.com
[2] https://www.electronicwings.com/sensors-modules/lm35-temperature-sensor
[3] https://how2electronics.com