# *Digital Thermometer using MCP9700 and Arduino*

*Pîrjol Adelina-Nicoleta*

*Pleş Andrei-Valentin*

*Table of Contents:*

# 1. Introduction

In the realm of electronics and circuits, the concern about heat generated by electronic components is a common one. Excessive heat can lead to damage, making it crucial to monitor and control temperatures. In this documentation, we present a project to address this concern by creating a digital thermometer using an MCP9700 temperature sensor and an Arduino board. This project not only allows temperature monitoring but also features a power cutoff mechanism when the temperature exceeds predefined limits.

# 2. Requirements

## 2.1. Hardware components required:

- Arduino UNO: The central processing unit for the project.
- MP9700 temperature sensor: A reliable and cost-effective temperature sensor.
- LCD Display – 1602A with I2C module: Provides a visual interface for temperature readings.
- Green LED light: Indicates normal temperature conditions.
- Red LED light: Signals high-temperature conditions.
- 2 X 220 ohm resistors: Resistors for the LED lights.
- Connecting wires: Establishes connections between components.
- Breadboard: Facilitates temporary and solderless connections, making prototyping and testing easy

## 2.2. Software requirements:

- Arduino IDE: The integrated development environment for programming the Arduino board.

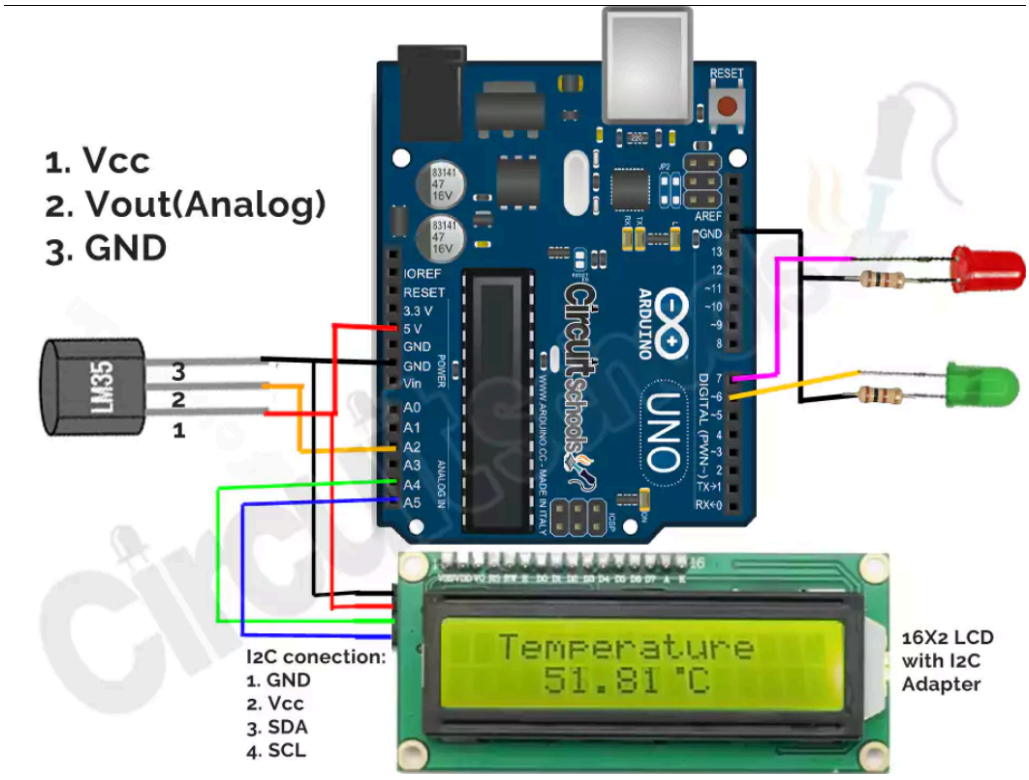
## 3. MCP9700 Temperature Sensor

The MCP9700 is a temperature sensor known for its accuracy and ease of use. Its key specifications include:

- Operating Voltage: 2.3V – 5.5V
- Temperature Range: -40 ℃ to +125 ℃
- Output Voltage at 0 °C: 500mV
- Slope: 10mV / ℃
- No need for additional components
- Pins: VDD, Vout, GND

## 4. Circuit Diagram Interfacing MCP9700 with Arduino

In the circuit diagram, the MCP9700 temperature sensor is connected to the Arduino. The VDD pin is connected to 5V, the Vout pin is connected to A2 analogue input port and the GND pin is connected to GND.




A 16 X 2 LCD display is connected using an I2C adapter. Two small Red and Green 5mm LED lights are connected to digital pins 6 and 7, and 220-ohm resistors are connected along the ground of LED lights.

## 5. Code

```
#include <Wire.h>
#include <LCD_I2C.h>
#define  MCP A2
#define RED 7
#define GREEN 6
float MCP_value;
float voltage;
float tempc;
// Set the LCD address to 0x27 for a 16 chars and 2 line display
LCD_I2C lcd(0x27, 16, 2);
void setup() {
  // initialize the LCD
  lcd.begin();
  // Turn on the blacklight and print a message.
  lcd.backlight();
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  lcd.setCursor(0, 0);
  lcd.print(" Digital ");
  lcd.setCursor(0, 1);
  lcd.print(" Thermometer ");
  delay(3000);
  lcd.clear();
}
 void loop() {
```

```
  MCP_value = analogRead(MCP);
  voltage = (MCP_value * 5) / 1024.0;
  tempc = voltage - 0.54;
  tempc = tempc / 0.01;
  /*------Display Result------*/
  lcd.setCursor(2, 0);
  lcd.print("Temperature");
  lcd.setCursor(4, 1);
  lcd.print(tempc);
  lcd.print("C");
  //Condition
  if (tempc > 26) {  // here you can change the temperature threshold value of led changes
    digitalWrite(RED, HIGH);
    digitalWrite(GREEN, LOW);
  } else {
    digitalWrite(GREEN, HIGH);
    digitalWrite(RED, LOW);
  }
  delay(1000);
}
```

## 6. Working of Digital thermometer using Arduino

After uploading the code, the live temperature for the sensor is displayed on the 16 X 2 LCD display. If the temperature rises above 26, the red LED will light up, and if the temperature is below 26, the green LED will light up. You can adjust the threshold temperature in the above code to make the lights change at the desired temperature.

## 7. Advantages and Applications

- Monitor the components' temperature: By placing the sensor beside electronic components, users can monitor temperature changes.

- Versatile application: The small size of the sensor allows for a wide range of applications.

- Accuracy:  Provides nearly accurate results with only ±0.5 C error.

- Skin temperature measurement:  Can be used to measure human skin temperature for various applications.

- Liquid temperature measurement: With proper precautions, the MCP9700 can measure the temperature of liquids, provided the pins are covered with waterproof material.

## 8. Conclusion

This project demonstrates a practical solution for monitoring and controlling temperatures in electronic circuits using an MCP9700 temperature sensor and Arduino. The integration of an LCD display and LED indicators enhances the user interface, making it suitable for various applications where temperature regulation is critical. The flexibility of the code allows users to customize temperature thresholds, providing adaptability for diverse use cases.

## 9. Bibliography


- **https://www.farnell.com/datasheets/2243473.pdf**
- **https://www.circuitschools.com/build-digital-thermometer-using-lm35-and-arduino/**
- **http://handsontec.com/dataspecs/module/I2C_1602_LCD.pdf**
- **https://www.optimusdigital.ro/ro/optoelectronice-lcd-uri/62-lcd-1602-cu-interfata-i2c-si-backlight-galben-verde.html**
- **https://www.instructables.com/How-to-calibrate-a-cheap-temperature-sensor/**
- **https://www.hackster.io/akshayjoseph666/interface-i2c-16x2-lcd-with-arduino-uno-just-4-wires-273b24**
- **https://steve.fi/hardware/4-pin-lcd/**
- **https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library**
- **https://www.arduinolibraries.info/libraries/liquid-crystal-i2-c**