Project Report: Temperature Monitoring System

Introduction

Monitoring ambient temperature is essential in many applications, such as weather stations, greenhouses, industrial processes, and home automation systems. This project focuses on building a **Temperature Monitoring System** using an Arduino and a temperature sensor. The measured temperature is displayed on an **LCD** or printed to the **Serial Monitor**, depending on the available interface.

Objectives

- Read temperature data using a temperature sensor (e.g., LM35 or DHT11).
- Display the temperature readings on a 16x2 LCD screen or the Serial Monitor.
- Learn to interface sensors and output devices with Arduino.

Components Required

Component
Arduino Uno/Nano
LM35 / DHT11 Temperature Sensor
16x2 LCD Display (optional)
I2C Module (for LCD, optional)
Breadboard & Jumper Wires
USB Cable (for Arduino)
Power Supply (5V or USB)

System Overview

The temperature sensor senses the ambient temperature and sends analog or digital data to the Arduino. The Arduino processes the data and displays the temperature either on a 16x2 LCD screen or through the Serial Monitor in the Arduino IDE.

Circuit Design

A. Using LM35 Sensor

- **LM35 Connections:**
  - VCC → 5V on Arduino
  - GND → GND
  - OUT → A0 (Analog pin)
- **LCD (with I2C Module):**
  - VCC → 5V
  - GND → GND
  - SDA → A4 (on Uno)
  - SCL → A5 (on Uno)

Arduino Code

Option A: LM35 with Serial Monitor

```
const int sensorPin = A0;

void setup() {
```

```
  Serial.begin(9600);
}

void loop() {
  int reading = analogRead(sensorPin);
  float voltage = reading * (5.0 / 1023.0);
  float temperatureC = voltage * 100.0;

  Serial.print("Temperature: ");
  Serial.print(temperatureC);
  Serial.println(" °C");

  delay(1000);
}
```

Option B: LM35 with 16x2 LCD via I2C

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
const int sensorPin = A0;

void setup() {
  lcd.begin();
  lcd.backlight();
}

void loop() {
  int reading = analogRead(sensorPin);
  float voltage = reading * (5.0 / 1023.0);
  float temperatureC = voltage * 100.0;

  lcd.setCursor(0, 0);
  lcd.print("Temp: ");
  lcd.print(temperatureC);
  lcd.print(" C");

  delay(1000);
}
```

Output Demonstration

Serial Monitor Output (Example):

```
Temperature: 27.36 °C
Temperature: 27.41 °C
Temperature: 27.45 °C
```

LCD Output (Example):

```
Temp: 27.4 C
```

Applications

- Room temperature monitoring
- Industrial safety systems
- Data logging and weather stations
- Greenhouse and HVAC control systems

Advantages

- Simple and low-cost system
- Real-time monitoring

- Expandable for IoT/cloud integration

Future Enhancements

- Log data to SD card or send to cloud
- Add humidity sensor (DHT11/DHT22)
- Use an OLED display for better visuals
- Wireless transmission (e.g., Bluetooth/Wi-Fi)

Conclusion

This project successfully demonstrates how temperature sensors can be integrated with Arduino to build a simple and effective temperature monitoring system. The use of either Serial Monitor or LCD makes it versatile for various applications, from home automation to industrial settings.