SOROTI UNIVERSITY

School of Engineering and Technology

Department of Electronics and Computer Engineering

BACHELOR OF ENGINEERING IN ELECTRONICS AND COMPUTER ENGINEERING

EEE 3108: WIRELESS SENSORS NETWORKS

LAB 1 ASSIGNMENT

BY

ANKUNDA PATRICIA 2301600103

KAWALA DESIRE 2301600079

MUCUNGUZI BENJAMIN 2301600084

ARIKOD CHARLES 2301600098

1. INTRODUCTION

Temperature monitoring is a crucial concept in several fields. Traditional temperatures sensors don't provide real-time alerts making them unreliable. This project shows the use of IoT devices in designing and implementing a SUN temperature monitoring and alert system.

2. PROBLEM STATEMENT

Extreme temperatures can pose serious risks to human safety and equipment reliability. A simple temperature monitoring system that only displays numerical values may not be sufficient for quick decision making as users must interpret the data before acting. Hence the need for a low-cost, easy to use system that measures and displays temperature in real time with intuitive alerts for temperature changes. This project addresses that need by extending a basic Arduino based monitoring setup with a visual alert mechanism using an RGB LED.

3. DESIGN RATIONALE

The system is designed with an Arduino Uno for its simplicity and reliability. A TMP35 sensor for its linear analog output and I2C LCD for clear real time display with minimal pin usage. An RGB LED is added to provide intuitive visual alerts where green indicates safe conditions, yellow signal elevated temperatures and red warns of dangerous levels. These choices ensure the node is accurate, user friendly and easily scalable into larger wireless sensor networks.

4. IMPLEMENTATION STEPS

4.1.WIRING

The implementation combined both hardware configuration and software. The hardware platform consisted of the TMP35 sensor whose output pin is connected to the Arduino's analog pin A0, its power pin to 5V and ground to GND. The I2C LCD is wired through the SDA and SCL lines to pins A4 and A5 of the Arduino board respectively. The RGB LED is connected through resistors to digital pins 9, 8 and 7 with the cathode grounded.

On the software side the Arduino is programmed to continuously read the analog signal from the TMP35, convert it to a voltage and apply calibration formula to compute temperature in Celsius. The LCD displays the values in real time. The program also contains logic for controlling the RGB LED.

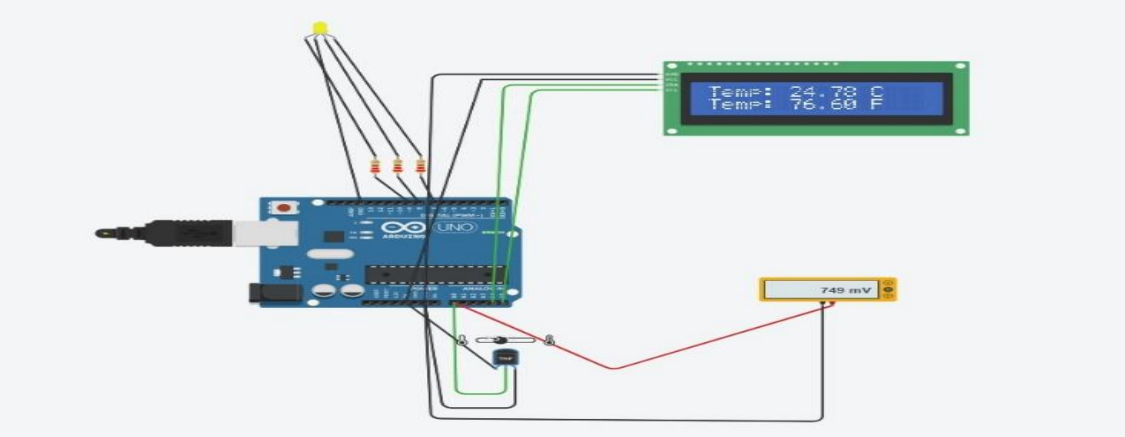
4.2.CODE

```
// Adafruit library for controlling the LCD
#include <Adafruit_LiquidCrystal.h>
Adafruit_LiquidCrystal lcd_1(0);

const int sensorPin = A0;     // TMP36 output pin

// intializing LED pins using separate Arduino digital pins
const int redPin = 9;
const int greenPin = 8;
const int bluePin = 7;

void setup() {
  // Initialize LCD with 16 columns and 2 rows
  lcd_1.begin(16, 2);
  lcd_1.print("Temp Monitor");
  delay(2000);  //Wait fot 2 seconds for the message to stay visible
  lcd_1.clear();

  // Set the RGB pins as outputs
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}

void loop() {
  // Read the analog value from the temperature sensor
  int sensorValue = analogRead(sensorPin);
  float voltage = sensorValue * (5.0 / 1023.0); // Convert ADC value to voltage

  // TMP36 outputs 0.5V at 0°C and increases 10mV/°C
  float temperatureC = (voltage - 0.5) * 100.0;  //Formula
  float temperatureF = (temperatureC * 9.0 / 5.0) + 32.0; // Convert °C to Fahrenheit

  // Display Celsius reading on the first line of the LCD
  lcd_1.setCursor(0, 0);
  lcd_1.print("Temp: ");
  lcd_1.print(temperatureC);
  lcd_1.print(" C   ");

  // Display Fahrenheit reading on the second line of the LCD
  lcd_1.setCursor(0, 1);
  lcd_1.print("Temp: ");
  lcd_1.print(temperatureF);
  lcd_1.print(" F   ");

  if (temperatureC < 20) {
    setColor(0, 255, 0);   // Green indicating normal temperature
  } else if (temperatureC > 20 && temperatureC < 42) {
    setColor(255, 255, 0); // Yellow indicating a warning
  } else if (temperatureC > 41){
    setColor(255, 0, 0);   // Red indicating harsh and unsafe conditions
  }

  delay(1000);
}

// Function to control RGB LED color
void setColor(int red, int green, int blue) {
  analogWrite(redPin, red);
  analogWrite(greenPin, green);
  analogWrite(bluePin, blue);
}
```

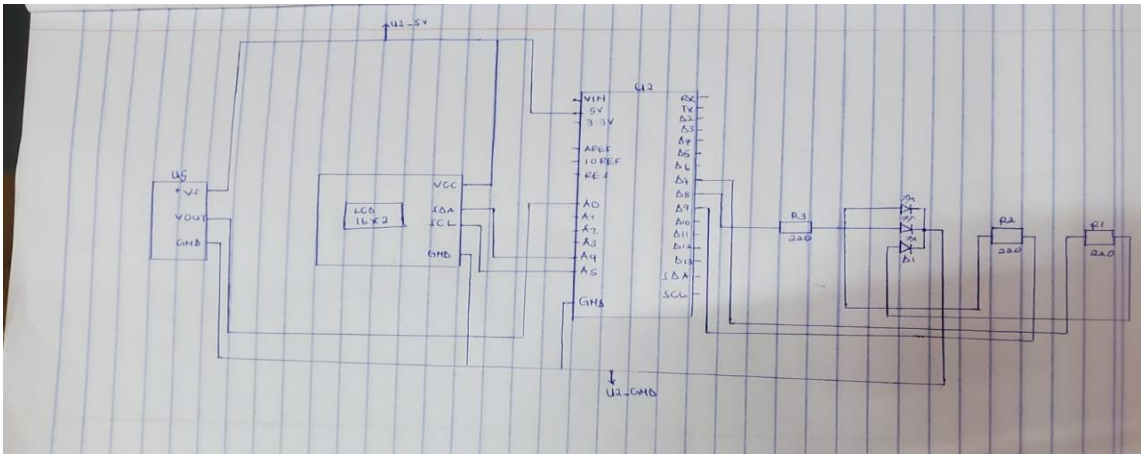
5. CIRCUIT DIAGRAMS

Link to the simulation: https://www.tinkercad.com/things/cwWrUThU756-group-4-wsn?sharecode=Pfmnv9D_HxmVOrF9VqV7ciFhFsvySRKU23dZzwTJJKM

TINKERCARD SCREENSHOT



HAND-DRAWN SCHEMATIC



6. OBSERVATION

After simulation in Tinker card, the LCD displayed the temperature values both in Celsius and Fahrenheit for every temperature change. For extreme temperatures that exceeded 42^0C, the RGB LED blinked continuously with red color. It turned yellow when the temperature was raised above room temperature(20^oC) but below 42^0C. For normal temperature below room temperature(20^oC), the LED turned green.

7. REFLECTION

This project demonstrates how a simple sensing node can provide real-time environmental monitoring and alerts. Within a larger Wireless Sensor Network, such a node can serve as one of the many distributed units, each collecting data in its local environment. With addition of a wireless module such as WIFI, each node would transmit its temperature readings to a central gateway where the data could be aggregated and analyzed.

8. CONCLUSION

The system effectively provides real time temperature monitoring with both display and alert functions. Its reliable, accurate and easy to use for detecting safe, warning and dangerous levels.