Title: Design process of room temperature monitor

Paul Nguyen, U80024817

1. Summary

This report details the design and prototype of a room temperature monitor using an Arduino Uno board and various electronic components. The project involves designing the mechanical layout within an enclosure, with components such as the temperature sensor, LCD, and LEDs placed strategically for visibility and accessibility. A custom 3D-printed battery holder was developed to securely mount the 9V battery inside the enclosure. The circuit was first prototyped on a breadboard, followed by careful wiring and soldering of the components to the Arduino board. Code was written to handle individual components, including blinking LEDs, reading temperature data, displaying it on the LCD, and activating the buzzer and red LED when the temperature exceeds the threshold. The project resulted in a fully functional room temperature monitor capable of monitoring and alerting users to temperature deviations. The key outcomes include the integration of various hardware components with the Arduino platform, demonstrating the practical application of sensors, displays, and alarms in embedded systems. This monitor has applications in places where maintaining specific temperature ranges is critical.

2. Introduction

Temperature monitoring is critical in many applications, ranging from environmental control in industrial settings to home automation systems. Maintaining a stable temperature is essential for ensuring the safety, comfort, and proper functioning of various systems and environments. In such contexts, real-time temperature data is valuable, especially when it is necessary to quickly detect any fluctuations that might fall outside the desired range.

This project focuses on the design and development of a room temperature monitor using an Arduino Uno. The monitor is equipped with an LCD display to show the current temperature, an alarm system (comprising a red LED and a piezo buzzer) to alert the user when the temperature exceeds a predefined threshold, and a user-friendly interface with a power switch for easy operation. The goal is to create an easy-to-use temperature monitoring system that provides both visual and audible alerts for temperature deviations, allowing users to take corrective actions as needed.

The purpose of this report is to outline the design process, component selection, circuit wiring, and coding involved in developing the room temperature monitor. The report aims to provide a comprehensive overview of how the various elements of hardware and software were integrated to achieve the desired outcome.

3. Design elements

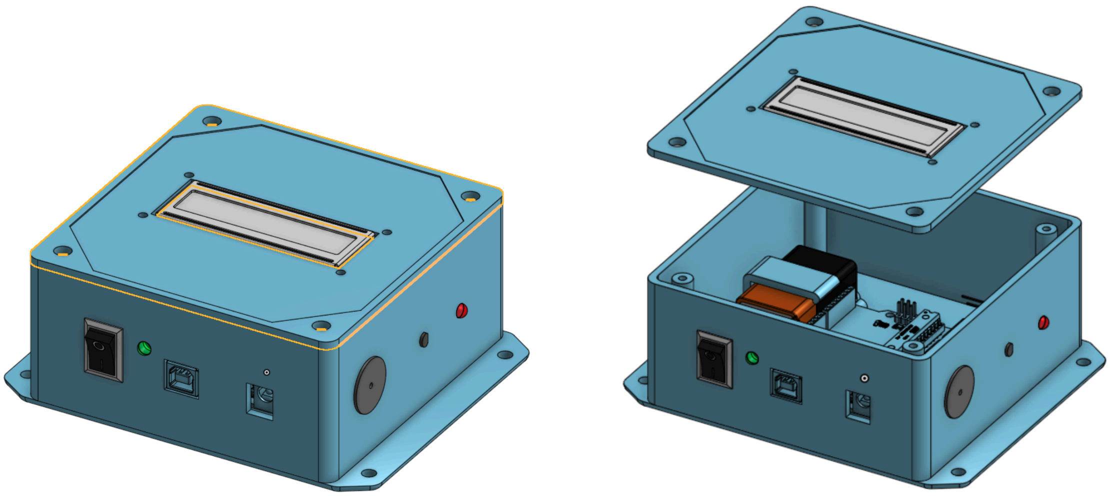
a. Components:

   i. (1x) Arduino Uno, (1x) Piezo, (1x) Temperature Sensor TMP36, (1x) Red LED, (1x) Green LED, (1x) Switch, (1x) Arduino LCD 2X16, (1x) 9 Volt Battery, (1x) 3D-printed battery holder, (1x) 220Ω Resistor, (1x) 1000Ω Resistor, (1x) ABS Enclosure, (1x) ABS Lid, (4x) Metal Screws, (1x) Base Plate, (8x) Plastic Screws, (8x) Plastic Nuts, (2x) Twist Nut Caps, (2x) Crimp Connectors, Solid Colored Wires (Black, Red, Purple, Gray), Head Shrink Tubing

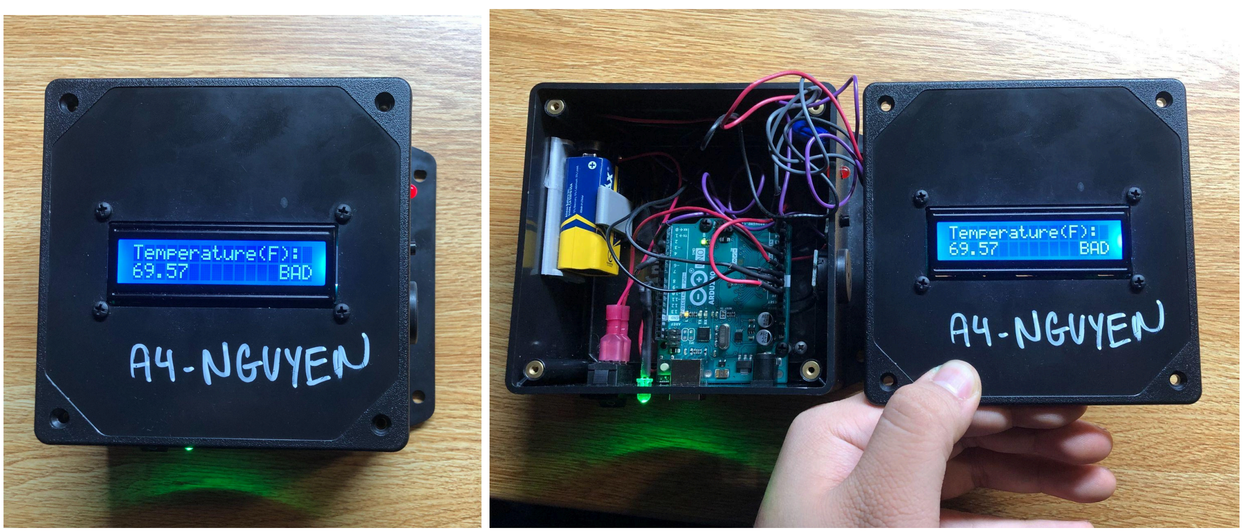
b. Precision measurements:

Item	W [mm]	L [mm]	H [mm]	Diameter [mm]
ABS Enclosure	146.8	120.0	57.5	n/a
Arduino Board	65.8	52.8	12.6	n/a
Switch	20.8	14.8	12.7	n/a
LCD 2X16	80.0	35.8	8.6	n/a
Buzzer	n/a	n/a	10.5	21.5
Temperature Sensor	n/a	n/a	4.4	4.5
LED	n/a	n/a	8.5	5.5

c. CAD Drawings:



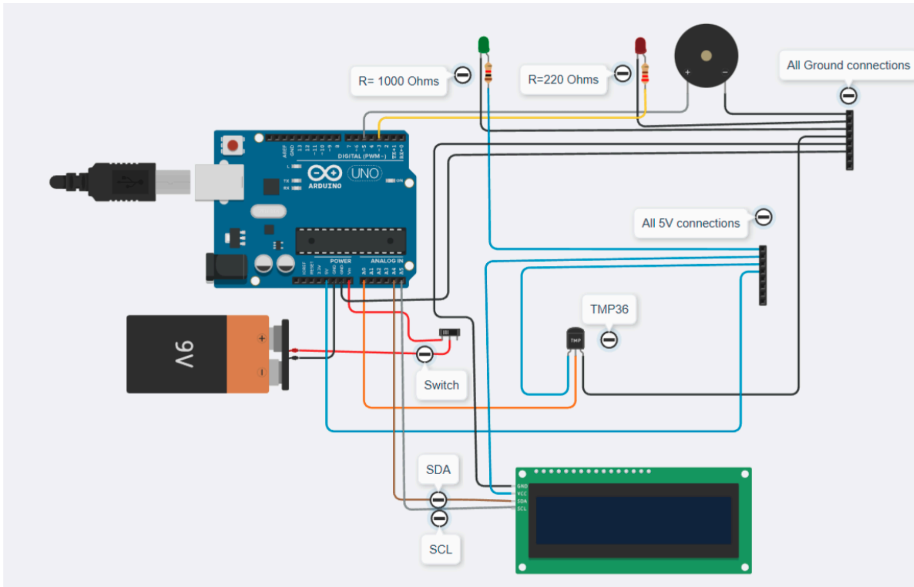
d. Physical View:



e. Arduino:

i. The Arduino microcontroller functions as the processor of the device. It receives analog temperature data from the TMP36 sensor, converts it into a digital Fahrenheit value, and then processes this information according to the programmed algorithm. The microcontroller then sends electrical signals to control the output devices, including the buzzer, LEDs, and LCD, to produce the desired behavior.

f. Wiring diagram



i. All ground connections connected by twist nut cap, all 5V connections connected by twist nut cap, 22AWG wire soldered to header pins for arduino pins, all other components directly soldered to wire.

g. For optimal reliability and robustness, 26 AWG and 22 AWG wires were selected over jumper wires for the final product. 22 AWG wires, capable of handling 920 mA, were used for power lines, while 26 AWG wires, rated for 361 mA, were employed for data lines connecting the Arduino, sensors, and actuators. The sensors' consistent current draw of less than 100 mA ensures the 26 AWG wires are more than adequate.
To limit current flow through the LEDs to a safe 20 mA, a 1000Ω resistor was paired with the green LED, and a 220Ω resistor was used for the red LED. These resistor values guarantee that the LEDs operate within their specified current range.

Red LED:

KLV: $V_R + V_{Red}$

$5V = I * 220\Omega + 1.8\ V$

$I = (5V - 1.8V)/220\Omega = 14.5mA$

Green LED:

KVL: $V_R + V_{Green}$

$5V = I*1000\Omega + 2.2V$

$I = (5V - 2.2V)/1000\Omega = 2.8mA$

In summary, the red and green LEDs, being paired with 220Ω and 1000Ω resistors encounter a usual current of 14.5 mA and 2.8 mA respectively, well under their 20mA operating limit, ensuring safe operation.

h. Power supply:

i. The Arduino, compatible with input voltages between 6V and 20V, is conveniently powered by a 9V alkaline battery. With a typical charge capacity of 550 mAh and a 50 mA current draw from the Arduino, the battery offers approximately 11 hours of operation. Additional components drawing power from the Arduino may reduce this lifespan. For alternative power sources, the Arduino can be connected to an external power supply via the barrel jack or USB port, bypassing the V_{in} pin typically used for battery power. The 9V battery strikes a balance between consistent power, longevity, and device size, making it a suitable choice for the ABS enclosure.

i. Arduino C++ Code:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
int pot_pin = A0;
int pot_value;
float Voltage;
float tempc;
float tempf;
int ledPin = 3;  // Pin for the red LED
int buzzerPin = 5; // Pin for the piezo buzzer
int tempPin = A0;
float highTemp = 75.0;
float lowTemp = 70.0;
float sensorValue;

void setup()
{
  lcd.init();
  pinMode(ledPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(900);
}

void loop()
{
  pot_value = analogRead(pot_pin);
  Voltage = (pot_value/1023.0)*5;
  tempc = (Voltage-0.5)*100;
  tempf = (tempc*1.8)+32;
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Temperature(F):");
  lcd.setCursor(0,1);
  lcd.print(tempf);
  lcd.setCursor(12,1);
  lcd.print("GOOD");
    if ((tempf > highTemp) || (tempf < lowTemp)) {
      while ((tempf > highTemp) || (tempf < lowTemp)) {
        digitalWrite(ledPin, HIGH);
        tone(buzzerPin, 1000);

        delay(350);

        digitalWrite(ledPin, LOW);
        noTone(buzzerPin);

        delay(350);
        pot_value = analogRead(pot_pin);
        Voltage = (pot_value/1023.0)*5;
        tempc = (Voltage-0.5)*100;
        tempf = (tempc*1.8)+32;
        lcd.backlight();
        lcd.setCursor(0,0);
        lcd.print("Temperature(F):");
        lcd.setCursor(0,1);
        lcd.print(tempf);
        lcd.setCursor(12,1);
        lcd.print(" BAD");
      }
    }
  delay(350);
}
```

j. Device Specifications:

i. The prototype can be powered either internally with a 9V battery or externally with a 6-20V power source. Drawing 45-55 mA, the 9V battery (approximately 550 mAh) provides 10.5-11.5 hours of operation. While the TMP36 temperature sensor can function between -40°C and 125°C, the practical operating range is limited to -20°C to 60°C due to component constraints. According to Kirchhoff's Voltage Law (KVL), the 1000Ω and 220Ω resistors for the green and red LEDs, respectively, limit current to 2.8 mA and 14.5 mA, ensuring safe operation below their 20 mA maximum rating.

4. Evaluation of Results

The primary goal of designing a reliable, real-time temperature sensor capable of reporting temperature and issuing alerts was successfully met. The project involved the design and assembly of both the ABS enclosure and the electronic circuit. The circuit, powered by a 9V battery with an 11-hour lifespan, utilizes a TMP36 temperature sensor to input data, which is then processed by an Arduino. The processed temperature is displayed on an LCD screen, while a Piezo buzzer and red LED provide alerts for temperatures outside the 70-75°F range. The compact ABS enclosure, measuring 146.8 mm x 120.0 mm x 57.6 mm, houses the components. While the device offers similar temperature measurement capabilities to a thermostat, it provides additional functionality through automated alerts and digital temperature display.