Adam Yanai - U42259107
Professor Sebesta
EK131 A7
May 3, 2023

Design Process of Room Temperature Monitor

1. Summary

The following report details the entirety of my process in designing and preparing a working room temperature monitor. The first part of the process was comprised of creating engineering design drawings to capture the dimensions of each element that would be incorporated into the design of the monitor. Next, an OnShape design for the monitor was created by designing a digital twin for the provided ABS enclosure box and within it planning out the composition of the parts. Additionally, a battery holder was designed from scratch to provide secure housing for a 9V battery. Once the digital twin was created and it was ensured that the parts all fit seamlessly into the enclosure, I created a circuit design for the project through Tinkercad. I then wrote the code that would be uploaded onto the Arduino to read temperature values through the TMP and display them on an LCD. Additionally, the monitor was programmed to play a tune through a buzzer and light up red through an additional LED (to complement a green LED which indicates that the device is on). With the planning process complete, I began creating the circuit by soldering the materials to one another and shrink-wrapping all exposed wires. Once the circuit was created, I attached the Arduino and battery holder to a mounting plate that connects to the bottom of the enclosure and drilled additional holes into the enclosure which serve as spots for the LEDs, buzzer, and TMP. I then inserted all parts into the enclosure and fastened each part into its spot as depicted below. The temperature monitor is able to display the temperature in both degrees F and C, as well as alert the user when the room temperature is recorded to be above or below a certain threshold (above 76 F or below 70 F). The significance of this project is that it could be used in rooms where steady temperature conditions are crucial (such as labs), or as a means of detecting dangerously high or low temperatures in public spaces (by adjusting the range of ‘normal’ temperatures).

2. Introduction

Temperature monitors – though simple devices – can prove crucial in certain situations. They can aid in detecting fire outbreaks, breaches into private spaces, the malfunctioning of everyday electronics such as a refrigerator, and much more. It is for this reason that designing temperature monitors in effective and efficient ways is an incredibly important task. Therefore, the goal of this project was to demonstrate a way in which temperature monitors could be built so that they are accessible and easy to use. Since a central aspect of modern innovation is the development of products that are simple, reliable, and readily available, the temperature monitor is a device the use of which will continue to prevail. The goal of this report, in turn, is to illustrate the process of creating such a device. It is through this report that I aim to emphasize the accessibility of both building a temperature monitoring device and its use by a variety of people from differing economic backgrounds.

3. Design Elements

The components used in this design are an ABS Enclosure, Arduino Uno Board, Switch, LCD 2X16, Buzzer, TMP36, Red LED, Green LED, jumper wires, shrinkwrap, screw caps, and resistors (one 220 Ω, one 330 Ω, and two 47 kΩ). The dimensions of these components can all be found in Figure 1 below. As for the placement locations for each of the electrical components, I decided to insert the green LED by the switch because the LED's role is to indicate whether the device is on or off. I decided put the buzzer, red LED, and TMP36 on the adjacent side to the green LED, switch, and Arduino port openings to allow for shorter wires that wouldn't create clutter within the box (because many of the wires from each individual part had to be connected through a screw cap). The placement decisions I made can be found below in Figures 2 and 3.

Item	Sketch identification	W [mm]	L [mm]	H [mm]	Diameter [mm]
ABS enclosure	01	117.4	117.4	60.0	N/A
Arduino Board	02	57.6	80.8	16.2	N/A
Switch	03	15.1	20.9	24.9	N/A
LCD 2X16	04	36.3	80.1	20.3	N/A
Buzzer	05	N/A	N/A	10.8	22.0
Temperature sensor	06	N/A	N/A	18.4	4.4
LED	07	N/A	N/A	27.8	5.1

Figure 1: Table of Relevant Dimensions

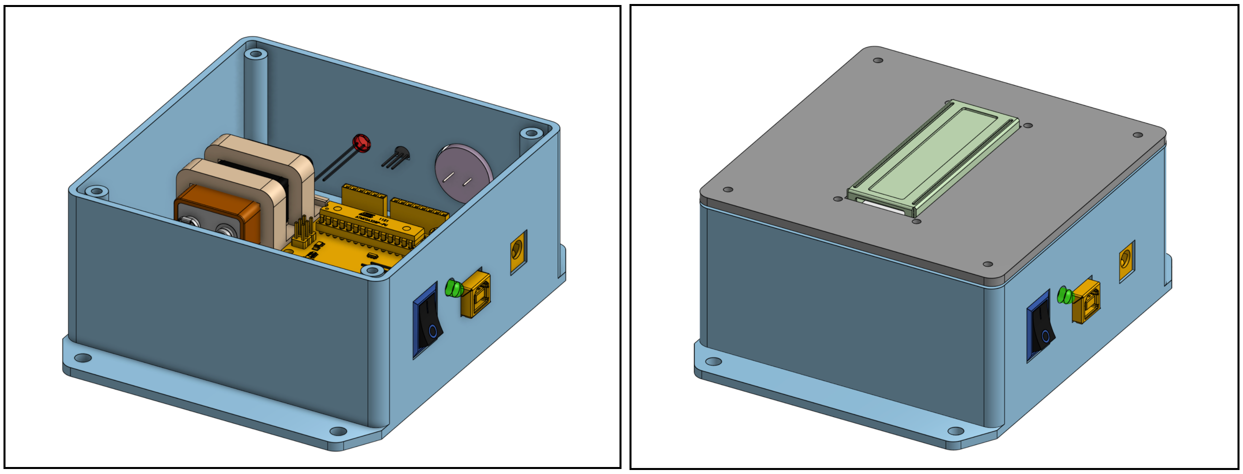

Figure 2: CAD Drawings

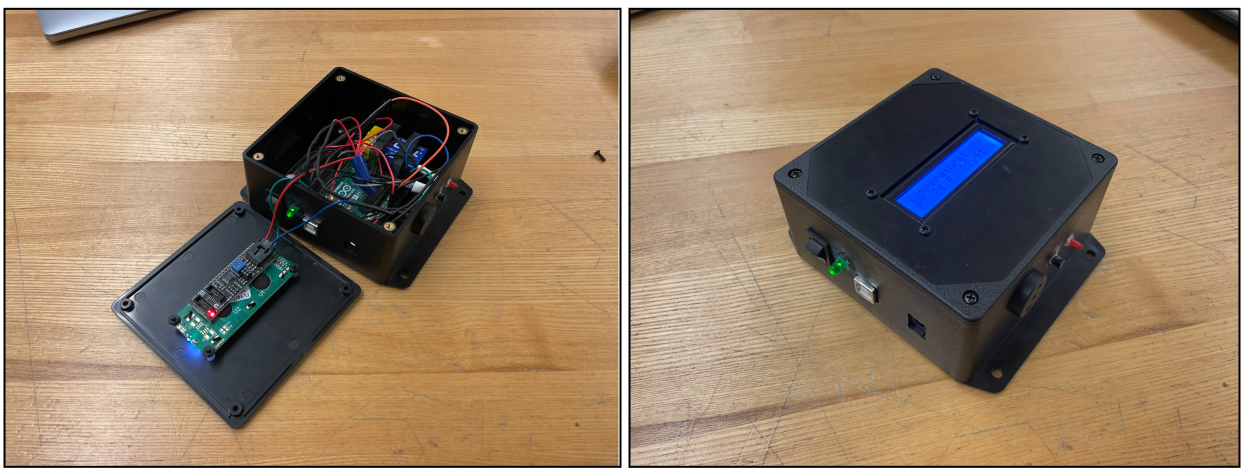

Figure 3: Pictures of Final Product

The purpose of the Arduino board in this temperature monitor design is to control and convert the analog signals of the circuit into digital signals. The Arduino is essentially the brain of the system, as it controls the entirety of function through the code that is downloaded onto it. My wiring plan for this project can be found in Figure 4 below and is also partly demonstrated in Figure 3 above, which demonstrates my use of shrink wrap to prevent exposed wires and solder, as well as screw caps to create intersections (or nodes) of several wires. Figure 3 also shows my use of 22 AWG jumper wires when creating connections that involve power (connect to the 5V port of the Arduino) and 28 AWG jumper wires when creating connections that involve data (such as the wire connected to the center rod of the TMP36, which extracts the data from the sensor). I decided to use 28 AWG instead of 26 AWG because I found it slightly easier to work with and solder onto. Since the maximum current of the circuit is less than 100 mA, I knew that the 28 AWG wires would work properly because they permit a maximum current of 920 mA, which significantly outweighs the max current of the circuit. As demonstrated in Figure 4 below, I decided to use a 330 Ω resistor for the green LED and a 220 Ω resistor for the red LED. I made this decision based on other similar devices, which have lights to indicate that the device is on but that are not too bright so that they become distracting. However, I wanted to make sure that it is clear and attention-grabbing when the red LED turns on because it indicates that the room temperature is no longer within the wanted range. I used Ohm's Law (V = IR) to determine that the lower resistance would allow for a greater current (since I = V/R) which in turn would cause the LED to light up brighter. The operating current of both LEDs is around 20-30 mA.

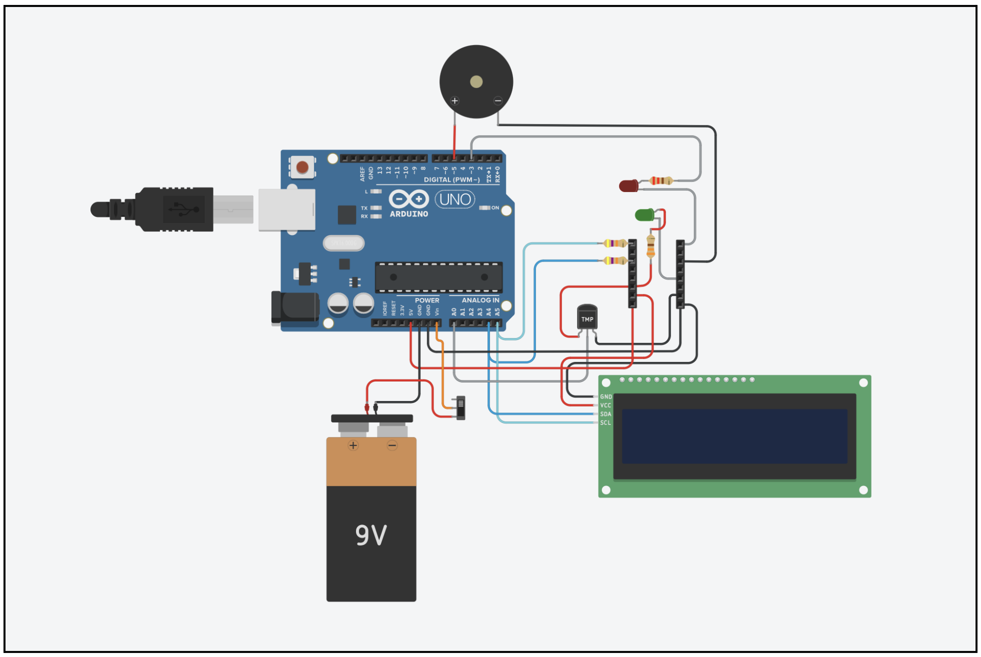

Figure 4: Wiring Diagram

The 9V battery is used as a power supply because it is reliable and efficient. Also, its voltage falls within the accepted voltage range of the Arduino, which is 6V to 20V. Regardless of where on the range the used power supply is, it would be converted to 5V through the built-in Arduino voltage regulators, so the 9V battery works perfectly fine. The charge capacity of a 9V battery is between 400 and 600 mAh, meaning that for the goal of this project the 9V is a great option (since the circuit does not consume an excessive amount of power). The capacity means that the battery could provide around 500 mA for one hour before dying, and power could also be provided externally by connecting the Arduino to a computer.

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

#define redLED 5
#define greenLED 3
#define buzzerPin 5
#define switchPin 2
#define tempPin A0

LiquidCrystal_I2C lcd(0x20, 16, 2);

float tempF;
float tempC;

void setup() {
  pinMode(redLED, OUTPUT);
  pinMode(greenLED, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(switchPin, INPUT_PULLUP);
  lcd.init();
  Serial.begin(9600);
}

void loop()
{
    // Define the notes array
    const int notes[] = {262, 294, 330, 349, 392, 440, 494, 523};

    lcd.backlight();
    digitalWrite(greenLED, HIGH);
    lcd.clear();
    lcd.setCursor(1, 0);
    lcd.print("Temperature:");
    lcd.setCursor(1, 1);
    lcd.print("Loading...");
    delay(1000);
```

```
    lcd.clear();
    while (true)
    {
      tempF = analogRead(tempPin) * 0.48828125;
      tempC = (tempF - 32) * (5.0/9.0);
      lcd.setCursor(0, 0);
      lcd.print("Temperature:");
      lcd.setCursor(0, 1);
      lcd.print(tempF);
      lcd.print(" F  ");
      lcd.print(tempC);
      lcd.print(" C");
      if (tempF >= 76 || tempF <= 70) {
        digitalWrite(redLED, HIGH);
        // Play "Hot and Cold" melody
        for (int i = 0; i < 16; i++) {
          tone(buzzerPin, notes[i % 8], 250 * (i / 8 + 1));
          delay(250 * (i / 8 + 1));
        }
        digitalWrite(redLED, LOW);
        delay(500);
      }
      else {
        digitalWrite(redLED, LOW);
        noTone(buzzerPin);
      }
      delay(500);
    }
}
```

Figure 5: Arduino Code

For this device, the voltage of the power supply (9V battery) was around 9.44V (as measured with the DMM), and the operating voltage of the circuit was around 5V because of the Arduino's built-in voltage regulators. The total current drawn in the circuit is around 250 mA, and the operating time of one 9V battery for this device is about 4 hours and 45 minutes. Based on the sensor specifications, the temperature range is from -40 °C to 125 °C, but the comfortable temperature range I set for this device is 70 °F to 76 °F. I decided to use a 330 Ω resistor for the green LED and a 220 Ω resistor for the red LED after using Kirchoff's Voltage Law (KVL) to determine that the lower the resistance, the greater the current and therefore the brighter the LED would light up (and I wanted the red LED to be brighter than the green).

4. **Evaluation of Results**

With the temperature monitor completed, I managed to create a fully functioning temperature sensor which alerts a user when the temperature exceeds a certain threshold/range. Upon being turned on, a green LED lights up to indicate that the device is running and the LCD screen turns on and begins displaying the measured temperature in both degrees F and C. When the measured temperature (by the TMP36) exceeds a range of 70 - 76 degrees F, a red LED turns on and a buzzer plays a tune, which would alert the user of the situation. When the measured temperature returns to the "safe" range, the red LED turns off and the tune stops playing. This kind of device has a variety of uses and applications, so I was happy that my rendition of it was a success. Of course, this device does not work to the same capacity as some other high level thermometers. However, it is still capable of functioning as a reliable temperature indicator and would be easy to repair if it ever malfuctioned. If I were to create this project again on my own, I would focus on making the device far smaller and more compact, so that it could be used in a greater variety of locations. I could do this by using a smaller enclosure, as well as a smaller LCD and perhaps one less LED which would allow me to use a smaller battery with less voltage. This would also help in reducing the weight of the box, since it is quite heavy when compared to other devices that complete the same function. Nevertheless, the usefulness of a temperature sensor is always clear. It can be used to regulate temperatures in – among many other things – homes, schools, appliances (refrigerators, ovens, etc.), and public spaces that involve extreme high or low temperatrures such as saunas and ski resorts. The temperature monitor is a universal device that will always aid humans in a variety of ways.

5. **Supporting Materials**

Included below are sketches of the individual components of the temperature monitor. More information about these components and their dimensions is found above in Figure 1.

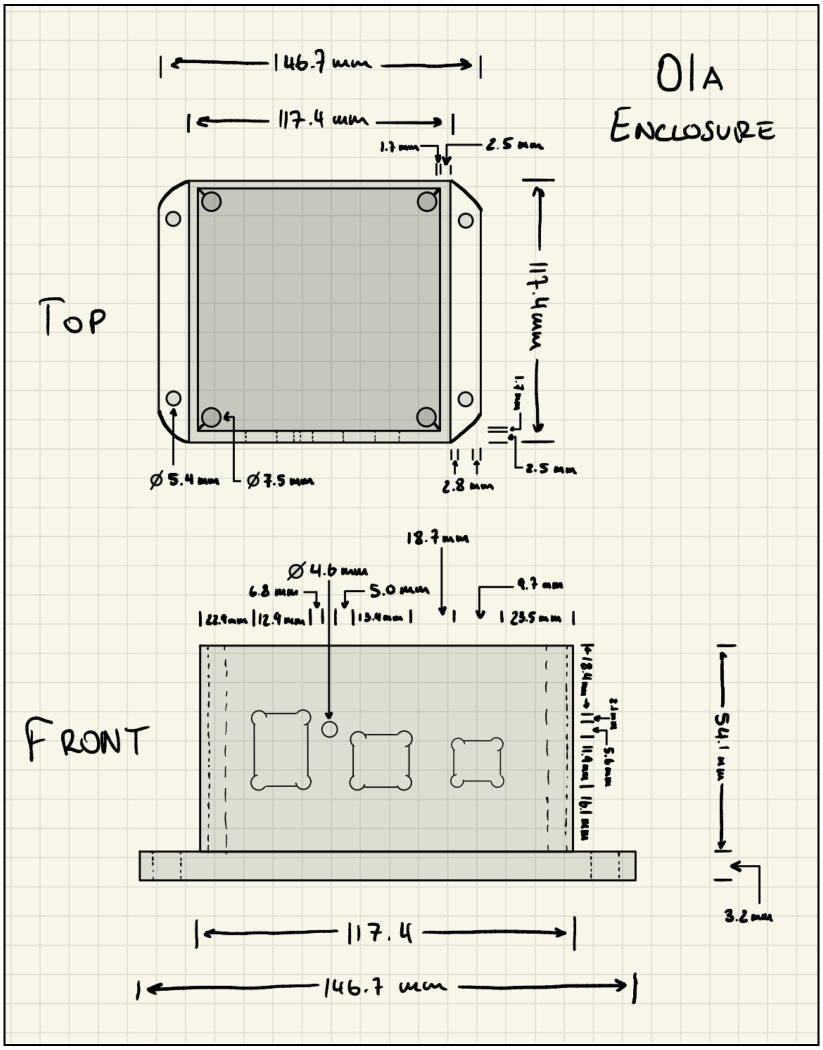

Figure 6.1: Sketches of Enclosure

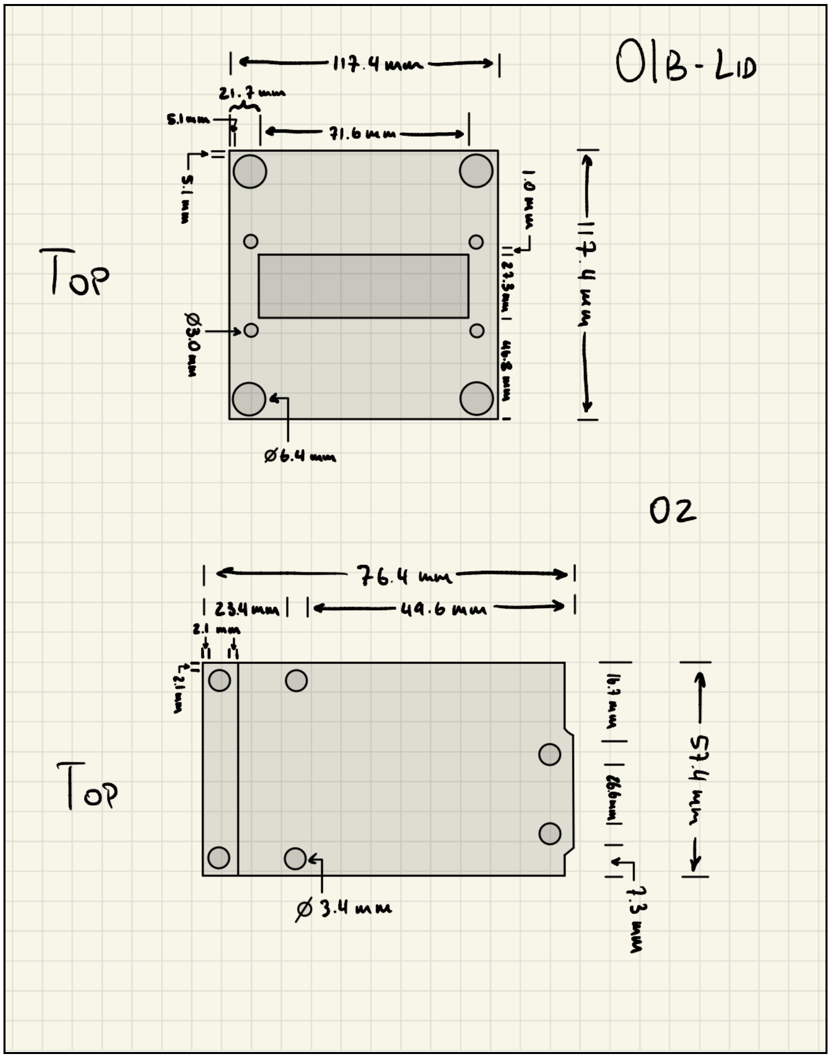

Figure 6.2: Sketches of Lid and Arduino Board

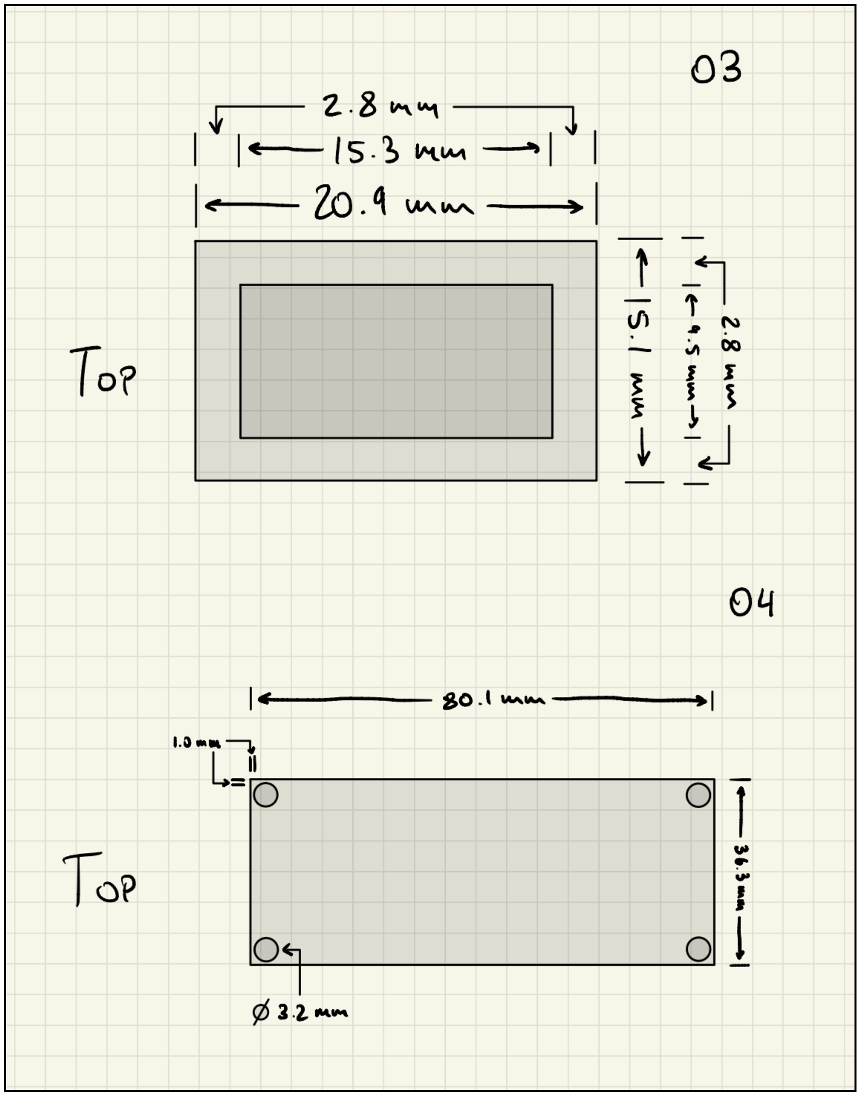

Figure 6.3: Sketches of Switch and LCD

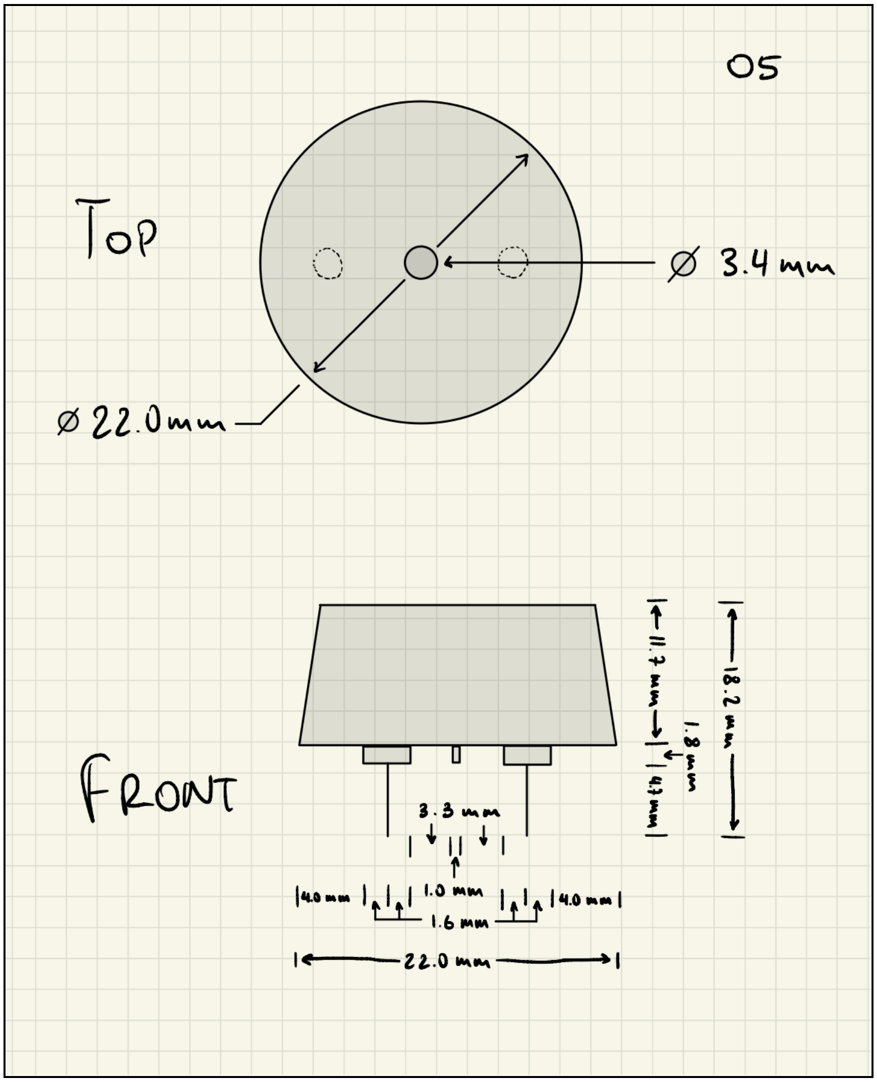

Figure 6.4: Sketches of Buzzer

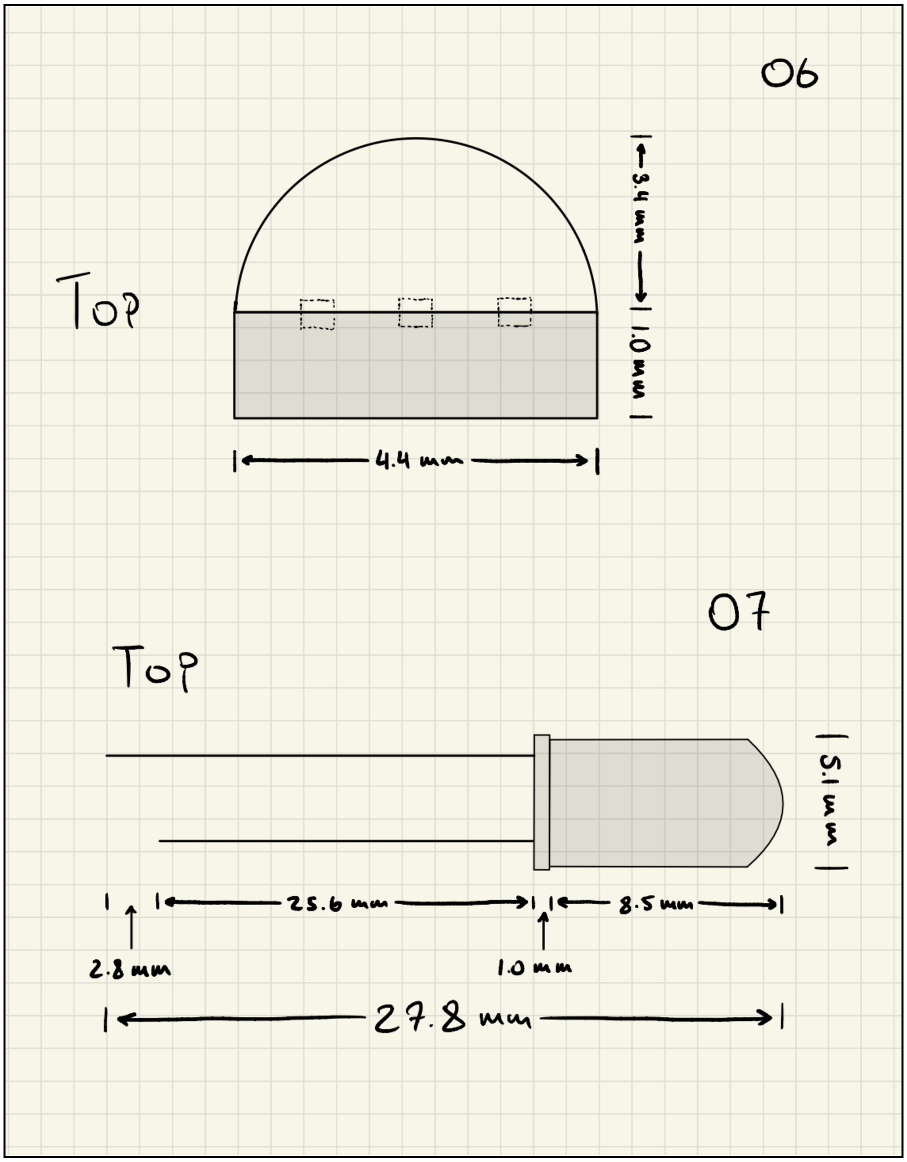

Figure 6.5: Sketches of Temperature Sensor and LED