**Title: Design Process of Room Temperature Monitor**

**Aaron Lin –**

## 1. Summary

This report details the development and testing of a room temperature monitoring prototype. The system was constructed using an Arduino Uno microcontroller, a TMP36 temperature sensor, an LCD display, indicator LEDs (red and green), and a piezo buzzer, all housed within an ABS enclosure. The Arduino is programmed to continuously read the temperature via the TMP36 sensor, convert the reading to Celsius and Fahrenheit, and display these values on the LCD screen. The prototype alerts the user with a red LED and a buzzer when the temperature falls outside a predefined range (60°F - 75°F). Power is supplied by a 9V battery, providing an estimated operational time of approximately 9 hours based on a measured current draw of 76 mA, though external power options exist. Key design choices include using 22 AWG and jumper wires and specific resistor values to regulate LED current. The project resulted in a functional proof-of-concept device capable of real-time temperature display and threshold-based alarming, demonstrating an approach for automated environmental monitoring, while highlighting the battery life limitations inherent in the current portable design.

## 2. Introduction

This report details the design, build, and test of a room temperature monitor prototype that uses an LCD screen to monitor temperature and an alarm to notify the user if temperature goes out of the desired range. This is important as humans are only comfortable within a certain temperature range. By altering the user when the temperature is outside the range, it allows time for the user to adjust the temperature through other means. The report covers the materials used in the creation of the prototype and the effectiveness of the design. The purpose of the report is to detail the prototype and its effectiveness.

## 3. Design elements

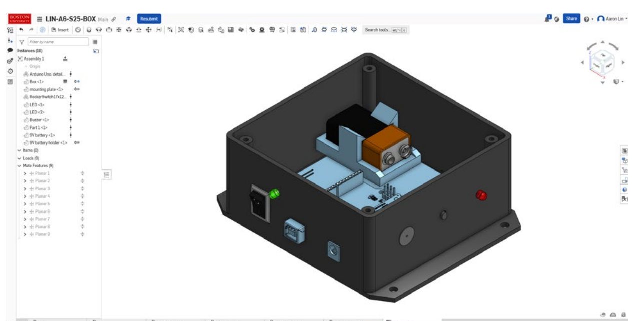
a. I used an Enclosure, Arduino Uno board, 9V battery, 9V battery snap, 9V battery holder, Solid core/stranded jumper wires, 22 AWG solid/stranded wires, a temperature sensor (TMP36), Alphanumeric I2C LCD (16x2 characters), red LED, green LED, piezo capsule (buzzer, PKM17EPP-4001-B0), resistors, and a 2-way switch in my design.

b. A table of relevant dimensions.

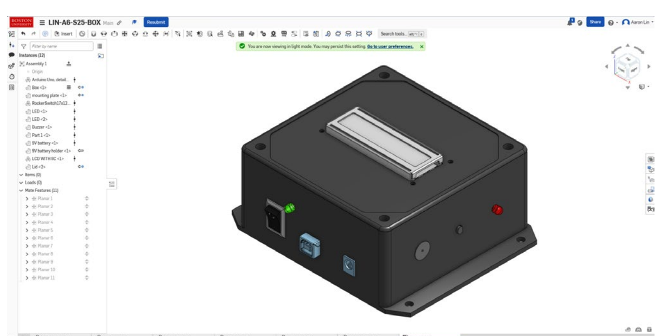
| Item | Sketch identification | W [mm] | L [mm] | H [mm] | Diameter [mm] |
|---|---|---|---|---|---|
| ABS enclosure | Top | 147 | 147 | N/A | N/A |
| ABS enclosure | Front | 147 | N/A | 57 | N/A |
| Arduino Board | Front | 53.3 | 72.8 | N/A | N/A |
| Arduino Board | Side | 53.3 | N/A | 12.6 (USB) | N/A |
| Switch | Bottom | 14.7 | 20.9 | N/A | N/A |
| Switch | Front | 14.7 | N/A | 23.3 (contacts) | N/A |
| LCD 2X16 | Top | 36 | 80 | N/A | N/A |
| LCD 2X16 | Front | N/A | 80 | 8.6 | N/A |
| Buzzer | Front | N/A | N/A | 6.1 | 12.5 |
| Temperature sensor | Front | N/A | N/A | 3.6 | 4.6 |
| LED | Front | N/A | N/A | 8.6 | 5 (inner) |

c. CAD drawings. Represented as screenshots of the final assembly.

Isometric View (Lid off):



Isometric View (Lid On):

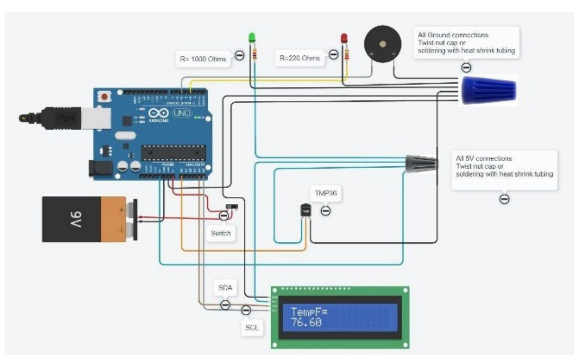


d. Top view picture of actual prototype with and without the lid while turned on

Left: Prototype with Lid on. Right: Prototype with Lid Off


e. The Arduino board in the circuit measures readings from the temperature sensor and executes code, converting analog signals to digital. Without the Arduino board, there would be no way to interpret the temperature measured by the TMP36, making the device useless.

f. I had the choice between soldering Arduino board connections to Right Angle Header Pins or using jumper wires and plugging the wires directly into the Arduino board. I ultimately chose to directly plug the wires into the Arduino to allow for future reuse of materials. The 5V and ground connections were connected using a twist nut cap to allow for common voltage to all components in the system. The alternative method would call for more soldering, which adds a point of failure in the circuit. The power switch was connected to the battery with spade connectors. Below is the wiring diagram.




g. For the most part, I used 22 AWG wires. Since the amperage within the circuit is so low (only 76 mA total), 22 AWG wires are able to carry current without any major loss due to resistance. However, for the LCD display, due to the transmission rate of data, jumper wires are needed to facilitate the flow of information. The resistors used in series with the Green LED was 1000 Ohms, while the resistor used in series with the Red LED was 220 Ohms. The operating current for the red LED (2 V drop) is 13.6 mA, while the green LED (2.2 V drop) draws 2.8 mA. More resistance is needed for the green LED to lower the overall current of the circuit to keep the LEDs in their operating ranges. The current was calculated using KVL, which states the net voltage across a loop must be zero.

h. The Arduino board accepts voltages ranging from 6-20 V. Since the 9-V batter falls in the range, it was used due to its availability and cost. As a low-cost solution, it is easily replaceable when it runs out of power. The 9-V has a nominal capacity of 690 mAh. Given that my box draws 76 mA of current, that means that the 9-V battery can run my box for around 9 hours straight.[1] We can also power the device externally with a DC jack that plugs into the Arduino. For extended use, the 9-V battery is not a good choice and should be replaced by a higher capacity version or external power should be used.

i. Arduino code used. (see appendix)

j. The 9-V battery supplies voltage at 9 Volts. The operating voltage of the circuit is 5 V. The box draws 76 mAh when running, leading to an operating time of around 9 hours on battery power alone.[2] The TMP temperature sensor can measure temperatures from -40°C to +125°C with an accuracy of ±1°C at +25°C and ±2°C over the −40°C to +125°C temperature range. As the device is meant to be used in a household, the temperature range is set from 60 degrees to 75 degrees Fahrenheit. Outside of this range, the prototype will alert the user to the out-of-range temperature. A 1kΩ resistor was used with the green LED to reduce the current drawn by the LED, to keep it within its operating range. A red LED requires more current, thus a 220 Ω resistor was used with the red LED.

---

[1] The actual time of operation would be lower than 9 hours since the battery's nominal capacity is calculated using a minimum voltage of 4.6 volts, below the 6 volts needed to power the Arduino.

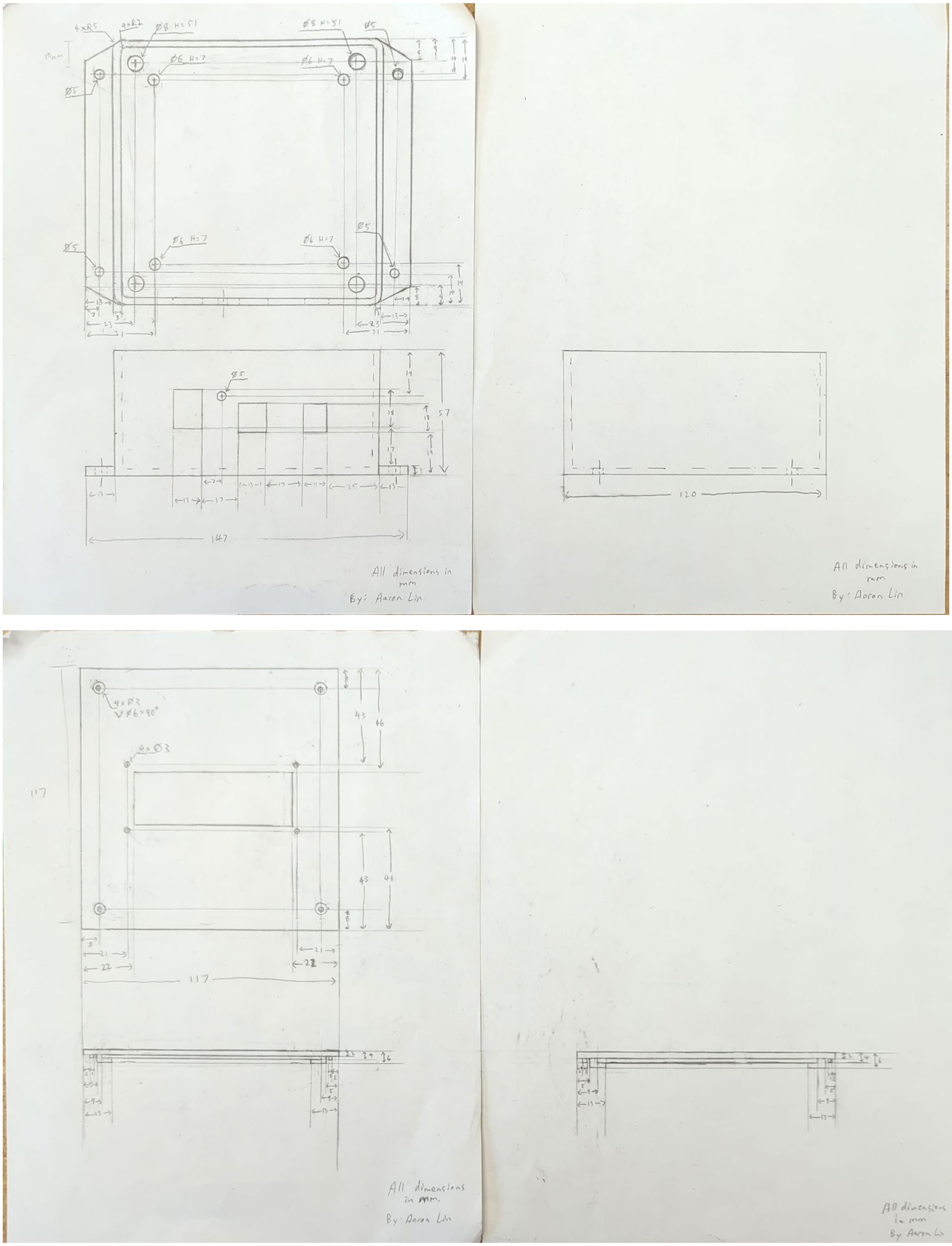
[2] Same as above.

## 4. Evaluation of Results

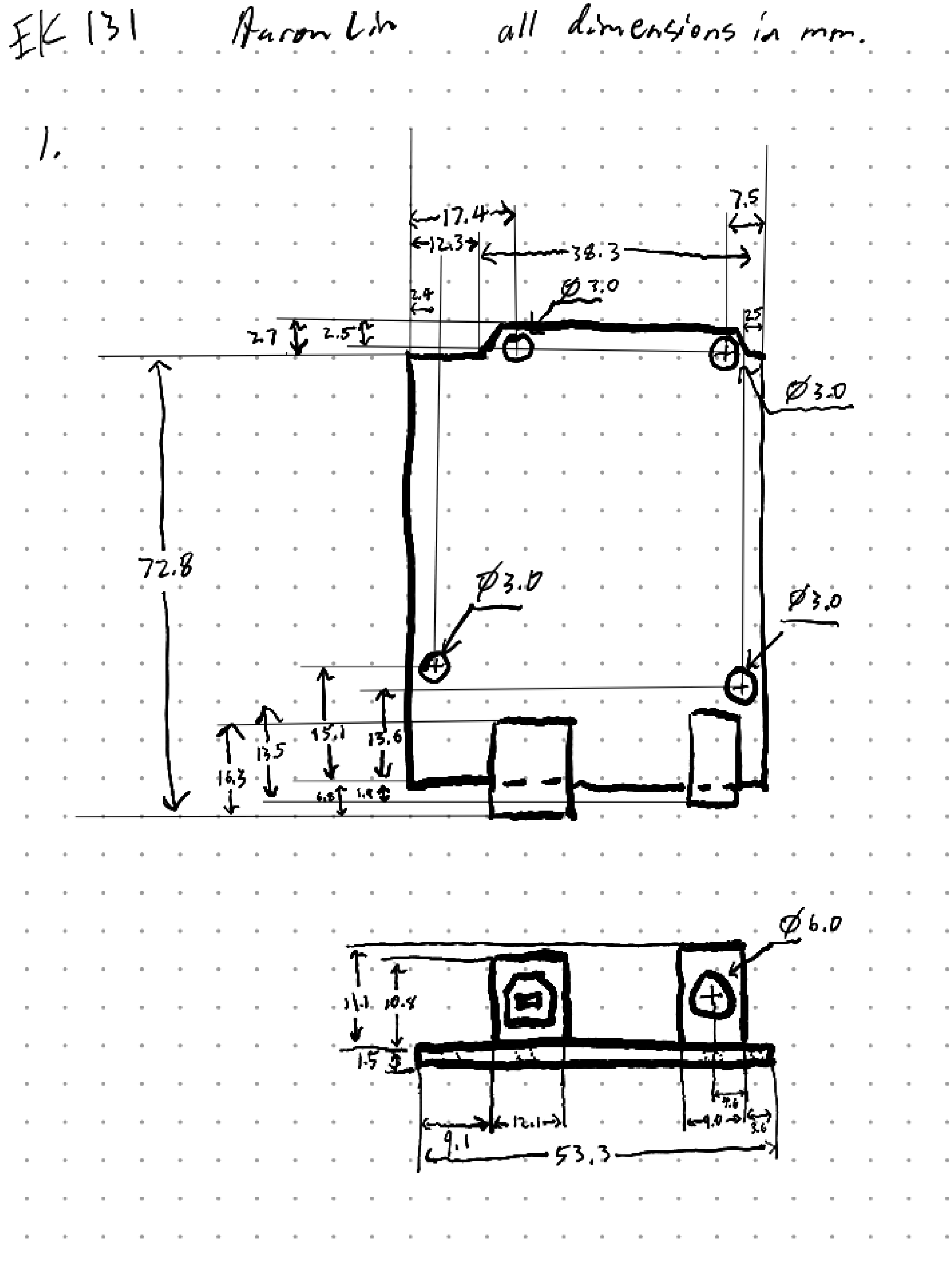
The prototype is successful in creating a room temperature monitor that uses an LCD screen to monitor temperature and an alarm to notify the user if temperature goes out of the desired range. Compared to standard thermometers, this device is able to alert the user when the room temperature leaves the desired range. This protype was constrained by the materials on hand. For a commercial product, consumers usually expect a much smaller sized device (see Nest Temperature Sensor), with the ability to be hardwired into the wall (to avoid the need for backup power). Future revisions could use more premium materials and reduce the weight of the device further. Optimizing production for economies of scale, different construction methods would need to be implemented to get rid of manual sauntering of wires.

### 5. Supporting Materials (appendix)

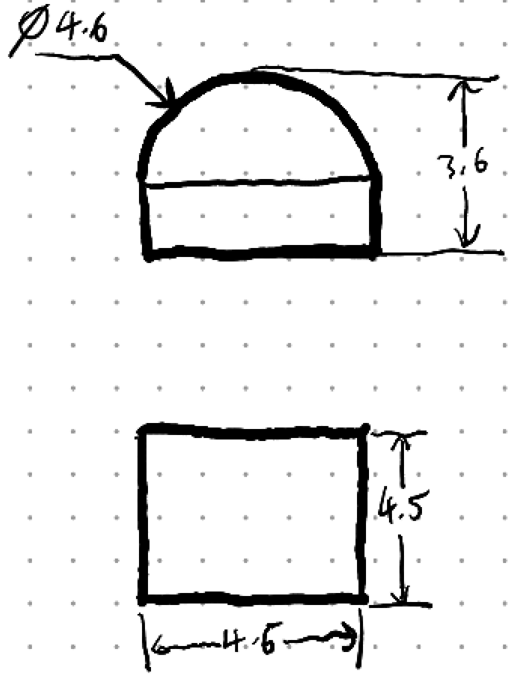
Detailed sketches of the enclosure:



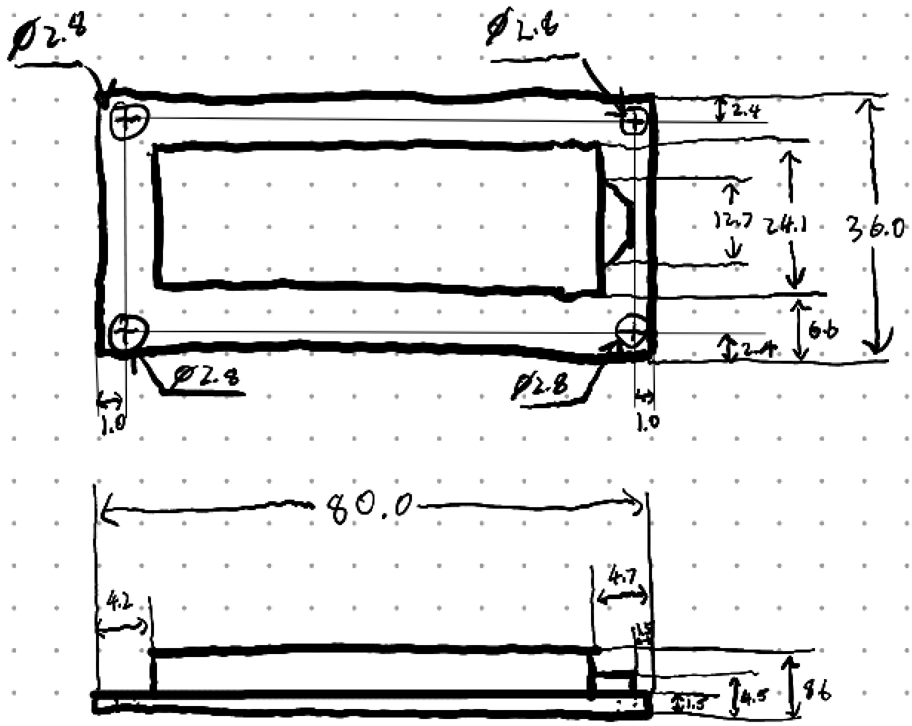
Below are other sketches of components included in the project:



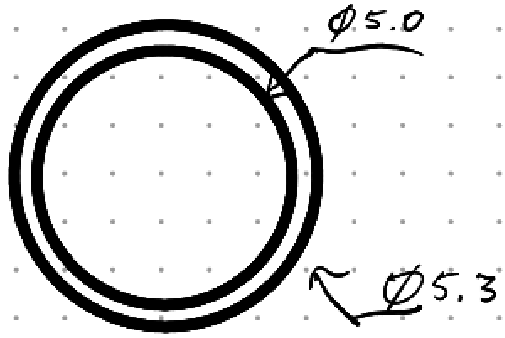
2 Temperature sensor  all dimensions in mm

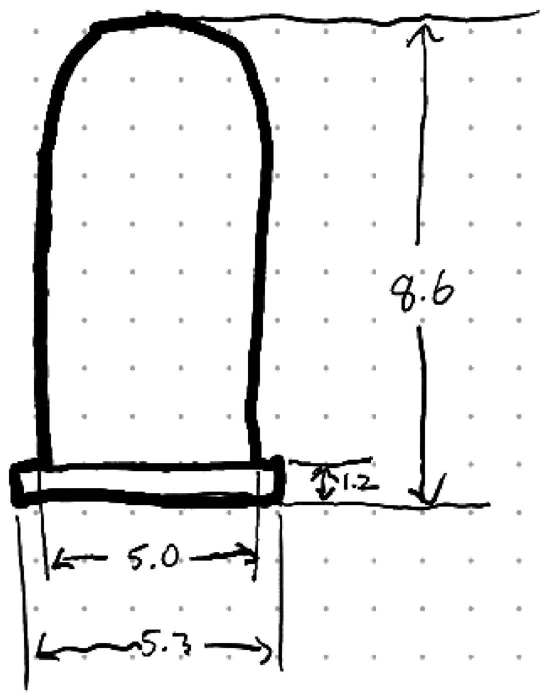



3, Alphanumeric I2C LCD  all dimensions in mm



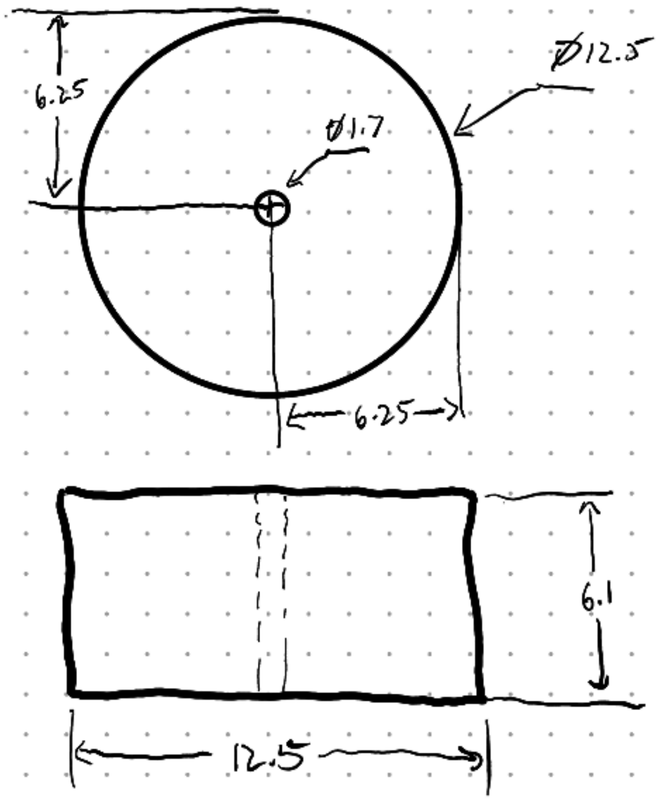
4 LED all dimensions in mm.



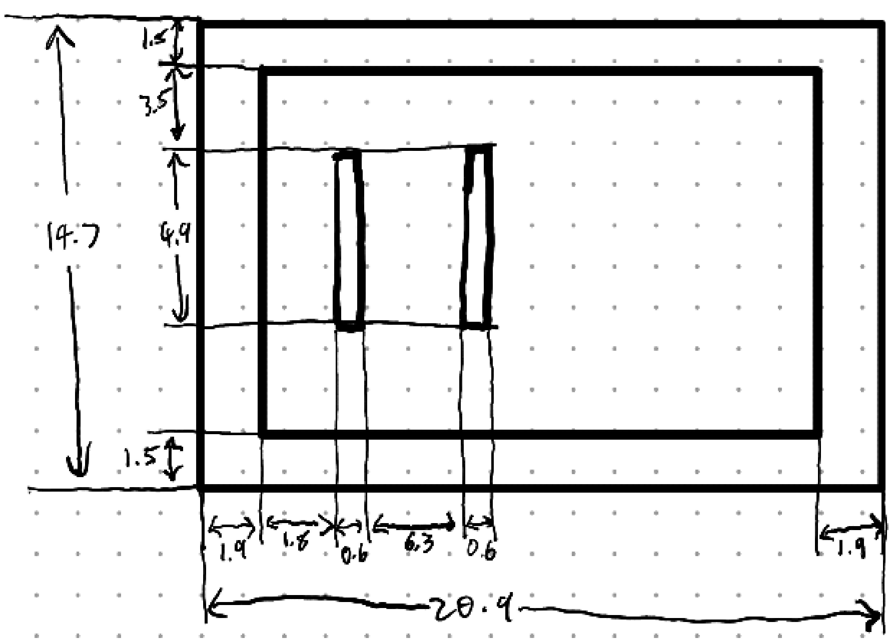
5. Piezo buzzer/capsule

all dimensions in mm.




6. 2-way switch bottom view

All dimensions in mm.



```
// C++ code
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
int LEDpin = 3;
int buzzer = 5;
int sensorPin = A0; //assigning the pin
int TADC; //ADC value 0-1023
float vTemp; // Actual voltage
float tempC; // temp in deg C
float tempF; // temp in deg F
LiquidCrystal_I2C lcd(0x27,16,2);  // set the LCD address to 0x27 for 16 char and 2 line display

void setup() {
  lcd.init(); // initialize the LCD
  lcd.backlight(); // Turns on LCD backlight
  pinMode(LEDpin, OUTPUT);
  pinMode(buzzer, OUTPUT);
  Serial.begin(9600); // Transfer speed
}

void loop() {
  delay(250);
  TADC = analogRead(sensorPin); //read pin value
  vTemp = 5*(TADC/1023.0); // convert voltage to vTemp
  tempC = 100*(vTemp-0.5); // convert vTemp to C
  tempF = (9*tempC/5)+32; // convert TempC to F
  Serial.print(tempC); Serial.println(" in degrees C.");
  Serial.print(tempF); Serial.println(" in degrees F");
  lcd.setCursor(0,0); lcd.print("Temp is: "); lcd.print(tempC); lcd.print(" C");
  lcd.setCursor(0,1); lcd.print("         ");lcd.print(tempF); lcd.print(" F");
  if (tempF <= 60.0 || tempF >= 75.0) {
```

```
    digitalWrite(LEDpin, HIGH);
    tone(buzzer, 500); // # is frequency
    delay(250);
    noTone(buzzer);
    digitalWrite(LEDpin, LOW);
  }
}
```