Design process of room temperature monitor

Vincent Brunn

1. Summary

This report covers the design elements of and process behind the creation of a room temperature monitor. The process began with a preliminary CAD drawing of the device, followed by a full assembly of its interior. From there, the ABS enclosure was modified to match the CAD drawing, and the circuitry for the components was designed. Once assembled, the circuitry and components were installed into the enclosure. Lastly, code was written to operate the Arduino within the device. The finalized and functional device represents the outcome of a semester of learning, demonstrating a capability to apply key engineering principles, concepts, and techniques.

2. Introduction

The purpose of this project was to give students the opportunity to apply the skills and concepts learnt throughout the semester in a hands-on way that required the use of tools and components also introduced throughout the semester. The project introduced an opportunity for students to exercise creative freedom within a structured environment by allowing individual control over the design and wiring layout of the device, while still standardizing its components, end functions, and capabilities in order to ensure a cohesive and functional product. The project itself was to create a room temperature monitor. This monitor would have to be capable of measuring temperature (through a TMP36 sensor), converting said measurement into a comprehensible form (using an Arduino and LCD screen), and setting off a pulsing buzzer-LED alarm if the ambient temperature exited a preset temperature range. This report will break down the design elements (components, specifications, functions, and design choices) behind each part of the room temperature monitor—from the physical enclosure to the underlying code—while also analyzing areas for improvement in the resultant product as compared to competing devices.

3. Design elements

a. The following components were used in my design:

1. ABS Enclosure **2.** Arduino Uno board **3.** 9V battery **4.** 9V battery snap **5.** USB cable **6.** Solid core jumper wires **7.** Stranded jumper wires **8.** 22 AWG solid/stranded wires **9.** TMP36 temperature sensor **10.** Alphanumeric I2C LCD (16x2 characters) **11.** LEDs (red and green) **12.** Piezo capsule (buzzer) [PKM17EPP-4001-B0] **13.** Resistors **14.** 2-way switch **15.** 9V battery holder **16.** Twist nut caps **17.** Protoboard. **18.** Mounting Plate **19.** Screws **20.** Lock nuts

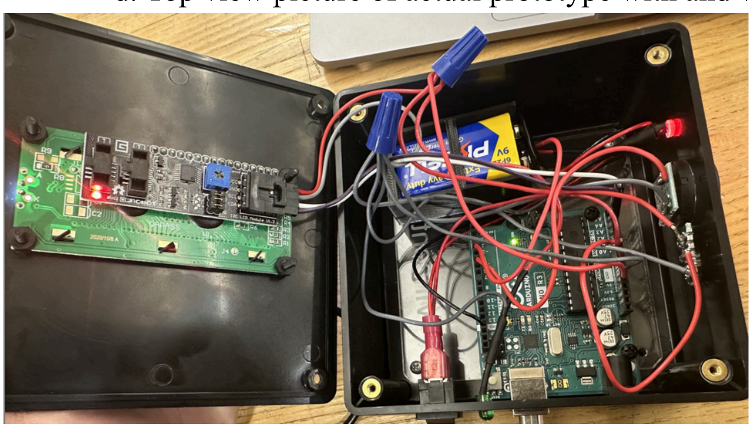
b. Precision measurements:

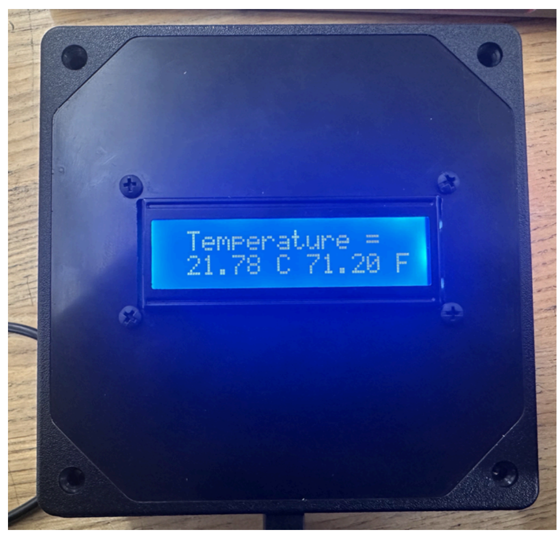
Precision measurements were taken of every major component's relevant measurements using digital calipers; and a table documenting these relevant dimensions can be found in the Supporting Materials section under subsection: a.

c. CAD drawings:

CAD drawings of the final layout's assembly were completed in the Onshape software and can be found in the Supporting Materials section under subsection: b.

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





e. Purpose of using an Arduino board.

An Arduino board was integrated into the circuit to act as a central processing unit, controlling critical functions–including the conversion of analog signals to digital signals. To start, the Arduino board is responsible for receiving and processing the raw output from the TMP36 sensor: it converts this raw data into interfaceable temperature values which then determines its decision-making. From there, if the recorded temperature is outside of a preset range–70-75 degrees Fahrenheit in this instance–the Arduino board is further responsible for initiating the alarm mechanism: consisting of a pulsing buzzer tone and illuminated red LED.

f. Wiring diagram and suggested wiring methods:

A diagram illustrating the internal wiring layout of the room temperature monitor can be found in the Supporting Materials section under subsection: c. In the case of the Arduino board, wires were tinned and slotted directly into preexisting ports. In most other cases, wires were soldered directly to their relevant components, with some including a resistor. The buzzer and TMP36 sensor were an exception, being first soldered to a protoboard before their relevant wires. Further, jumper wires were used in lieu of 22 AWG wires for all LCD screen connections, as it needed to accept data from the Arduino board. Lastly, spade connectors were used for the interface between the power source and Arduino board, as they provide a strong and reliable electrical connection. To organize and simplify wire groupings, two twist nut caps were used: one for all of the 5V connections, and another for all of the Ground connections.

g. Wire Gauge:

Two wire gauges were used in the wiring of this monitor: 22 AWG and 26 AWG. 22 AWG wires were used to supply power within the monitor, because a thicker wire experiences less voltage drop, leading to a more consistent connection; conversely, the 26 AWG jumper wires were used for data connections because thinner wires offer less resistance to passing signals, maintaining reliable data transmission. The maximum current these wires can handle is around 920mA (360 mA for some 26 AWG), while the maximum current drawn in their circuit is less than 100 mA. A 1K Ohms resistor was used in series with the Green LED, while a 220 Ohms resistor was used in series with the Red LED. Based on KVL (specifically I=V/R), the LEDs have the following operating currents: Red = 5V/220 Ohms = 22.7 mA, Green = 5V/1000 Ohms = 5 mA.

h. Internal power supply:

A 9V battery was used to power the device because it is a compact, affordable, reliable, and compatible power source (as the Arduino accepts voltages 6-20V). The zinc-carbon 9V battery used in this design has an approximate charge capacity of 500 mAh: at a device draw of 74.6 mAh, the battery should last about 500/74.6 hours, which is just under 7 hours. These factors make the 9V battery a solid option for this scale project, but an alternative power source would likely be recommended for any long-term use or expansion of the device. Because the Arduino's ports are within the device, any power sources must fit within the enclosure—unless the lid is removed, in which case the device can be powered "externally."

i. The following is the Arduino code used in the design:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
// TMP36 Pin Variables
int sensorPin = A0; //the analog pin the TMP36's Vout
(sense) pin is connected to
// Buzzer Pin
int buzzerPin = 5;
// LED Pin
int ledPin = 3;
// Initialize the LCD
LiquidCrystal_I2C lcd(0x20, 16, 2); // Set the LCD
address to 0x20 for a 16x2 display
int backlightLevel = 255; // Set the backlight level
(adjust as needed)
void setup() {
// Initialize the LCD screen
 lcd.init();
 lcd.backlight(); // Set the backlight level
 // Set buzzer, LED, and LED pins as OUTPUT
 pinMode(buzzerPin, OUTPUT);
 pinMode(ledPin, OUTPUT);
}
void loop() {
 // Getting the voltage reading from the temperature
sensor
 int reading = analogRead(sensorPin);
 // Convert the reading to voltage
 float voltage = reading * 5.0;
 voltage /= 1024.0;
 // Convert to temperature in Celsius
 float temperatureC = (voltage - 0.5) * 100;
 // Display temperature on the LCD
 lcd.setCursor(1, 0);
 lcd.print("Temperature = ");
 lcd.setCursor(1, 1);
 lcd.print(temperatureC);
 lcd.print(" C ");
 lcd.print((temperatureC*9/5)+32);
 lcd.print(" F");
 // Check if temperature is outside the range of 21-24
degrees Celsius (equivalent to 70-75 degrees Fahrenheit)
 if (temperatureC < 21.0 || temperatureC > 24.0) {
   // If temperature is outside the range, pulse the
buzzer and blink the LED
   for (int i = 0; i < 3; i++) {
     tone(buzzerPin, 1000); // Sound the buzzer at 1kHz
     digitalWrite(ledPin, HIGH); // Turn on the LED
     delay(500); // Pulse duration
     noTone(buzzerPin); // Silence the buzzer
     digitalWrite(ledPin, LOW); // Turn off the LED
     delay(500); // Silence duration
   }
 } else {
   // Otherwise, turn off the buzzer and LED
   noTone(buzzerPin); // Stop the buzzer sound
   digitalWrite(ledPin, LOW);
 }
 // Output temperature data to the serial monitor
 Serial.print("Temperature C: ");
 Serial.print(temperatureC);
 Serial.print(" | Temperature F: ");
 Serial.println((temperatureC * 9.0 / 5.0) + 32.0);
 delay(1000); // Wait for a second
}
```

j. Specifications of the prototype:

Voltage of Power Supply:	9V
Operating voltage of the circuit:	5V
Current drawn:	74.6mAh
Operating time of battery:	~7 hours
Temperature range (TMP36):	-40 degrees Celsius to 125 degrees Celsius
Comfortable temperature range:	70 degrees Fahrenheit to 75 degrees Fahrenheit

The decision to use a 1K Ohms resistor with the green LED, while only using a 220 Ohms resistor with the red LED was a deliberate one, and can be explained using KVL. As stated by KVL: the sum of the potential differences across any closed loop in a circuit must be zero. In effect, this means that each resistor-LED pairing must have a total drop of 5V, 2V of which is resultant from the LED, with the other 3V having to be a result of the resistor. Based on I = V/R; with the same numerator, but a larger denominator, the 1K ohms resistor is going to have less current, meaning that the Green LED connected in series with it will be dimmer, as LED brightness is directly proportional to the current flowing through it. This is the desired result, as the Green LED is only a power indicator, and does not need to be very bright, while the Red LED, being an alarm, needs to be bright, and as such has a weaker resistor connected to it.

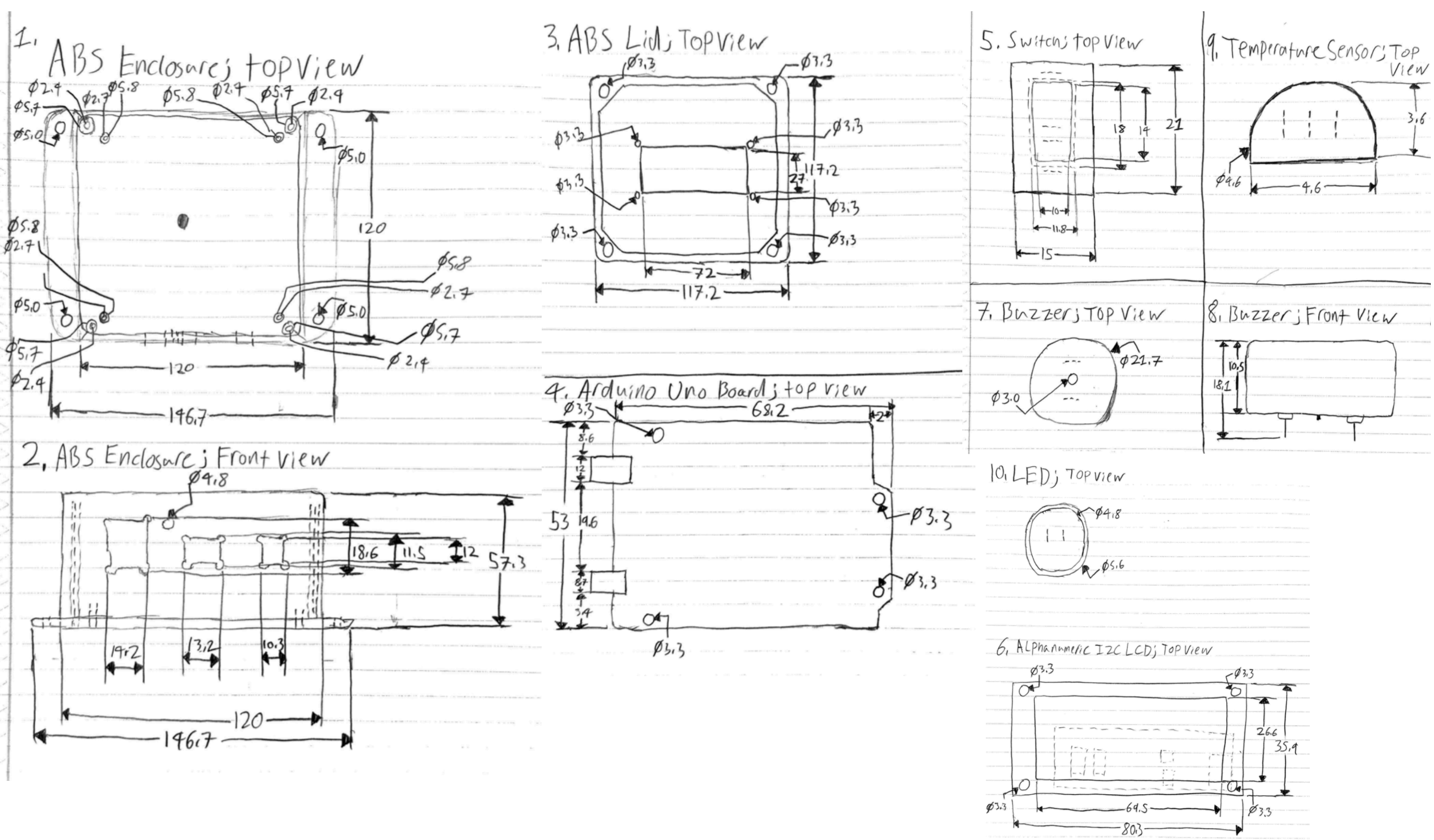
4. Evaluation of Results

Each of the design objectives have been achieved: I have created a functional room temperature monitor that can measure temperature and convert it into comprehensible values to be displayed on an LCD screen; further, it can determine whether said temperature is within a predetermined range, and can act accordingly if said range has been breached by activating an alarm comprised of a buzzer and Red LED until the temperature has returned within the range. This report has outlined the components within and methods by which a room temperature monitor was constructed. Aside from the form factor, a notable difference between this design and temperature monitors currently on the market is the functional temperature range. The TMP36 sensor used in this design has a temperature range of -40C to 125C, while many temperature monitors currently on the market only operate between 0 and 50 degrees Celsius, per Acurite's specifications. Also, some monitors on the market must be permanently mounted on a wall, while this device is fully portable. While boasting a larger temperature range, this device does, admittedly, suffers from a few limitations: the most noteworthy of which being battery life, size, and weight. In specific, the device is powered entirely by a 9V battery, meaning that it does not have the capacity to be left operational for long periods of time; further, it is quite bulky, especially compared to some modern thermometers, meaning it may not fit in certain places and would be very conspicuous; and lastly, it is quite heavy, resulting in a non-sleek profile that hinders it from being used in unstable or unsupported environments. Future iterations should seek to slim down this frame by optimizing the ABS enclosure (as there is plenty of unused space remaining), prolong battery life by coding run-cycles or switching to an alternative power source, and reduce weight by changing the material of the ABS enclosure. Overall, however, the device is still relatively portable and consistent, meaning its efficacy is entirely in the consumer's hands: its placement and operating times can be adjusted to either purposefully monitor or avoid abnormal temperature zones (as to not skew data), per the user's needs.

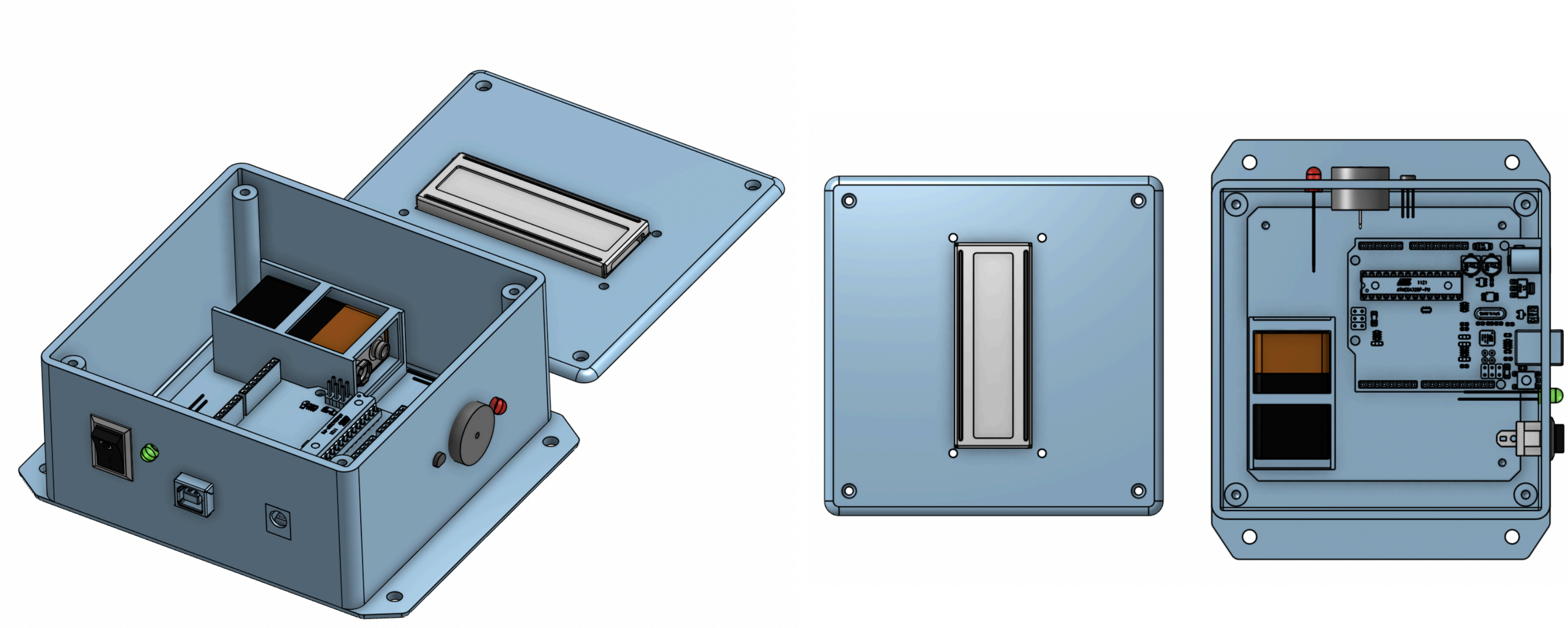
5. Supporting Materials (appendix)

a. (Table describing precision measurements taken of major components)

Item	Sketch identification	W [mm]	L [mm]	H [mm]	Diameter [mm]
ABS enclosure	1, 2	120	146.7	57.3	N/A
Arduino Board	4	53	68.2	12.5	N/A
Switch	5	15	21	24.9	N/A
LCD 2X16	6	35.9	80.3	7.8 (Screen)	N/A
Buzzer	7, 8	N/A	N/A	10.5	21.7
Temperature sensor	9	N/A	N/A	4.7	4.6
LED	10	N/A	N/A	8.5	4.8
ABS Lid	3	117.2	117.2	4	N/A



b. (CAD drawings and assembly of the device)



c. (Final wiring diagram)