EK 131 Semester Project

Spring 2021

Rachel Gardiner

Summary

The final prototype of this project is a temperature monitor that displays the temperature of a room and alerts users with a flashing LED and a buzzer when the temperature is outside of a specified range. I first measured the dimensions of all components and of the enclosure. I designed the layout of my components so they could be wired neatly and so necessary components are accessible from outside the enclosure. I constructed the enclosure out of cardboard because of lack of machinery and made precision cutouts with an x-acto knife. I coded an Arduino Uno board to read the temperature, display it, and flash the red LED and sound the buzzer when the temperature is outside the set range. The final prototype effectively meets all feature requirements and is extremely useful for any household.

Introduction

This report details the process of designing and prototyping a temperature monitor to display the temperature of a room and alert users if the temperature is below or above the desired range. It is useful for anyone because it clearly displays the temperature of a room, allowing someone to adjust their thermostat as needed. It is also helpful in the event that an air conditioning or heating unit malfunctions; users will notice the blinking light and beeping buzzer and be able to fix the problem. I plan to keep the monitor myself and use it in my own bedroom. I like my room to be cold when I go to sleep, but I often forget to crack the windows in the evening to begin to cool it down. This monitor will be a helpful reminder to open the windows when my room gets too hot and will also be useful if the heating or AC stop working.

The purpose of this semester-long project is to demonstrate the skills I learned throughout the EK131 course, including circuit wiring, mechanical design, and Arduino coding. The monitor required many features that I previously did not know how to install. For instance, a red LED light blinks when the temperature is too low or too high. Before this class, I did not know anything about Arduino boards or coding and would not have been able to program the light to blink. Throughout the process of making the monitor, I also learned mechanical design skills including engineering sketching by hand and using 3D design software. In this report I will describe the process of designing and building the temperature monitor and will evaluate my final result.

Project Components and Features

The components needed for the prototype:

1. Enclosure
2. Arduino Uno board
3. 9V battery
4. 9V battery snap
5. USB cable
6. Breadboard 400 points
7. Jumper wires (22 AWG)

8. Temperature sensor [TMP36]
9. Alphanumeric I2C LCD (16x2 characters)
10. LEDs (red and green)
11. Piezo capsule (buzzer) [PKM17EPP-4001-B0]
12. Resistors
13. 2-way switch

Features:

- Temperature sensor able to measure the temperature of the room.
- LCD screen visible from the outside of the box.
- Two-way switch to turn on/off the monitor.
- Green LED to indicate on/off function.
- Red LED to indicate out-of-range temperature.
- Buzzer to indicate out-of-range temperature.
- Arduino board accessible from outside the enclosure.
- 9V battery mounted inside to power the Arduino board.

Dimensions

Table 1. Dimensions of Components.

Item	**W [mm]**	**L [mm]**	**H [mm]**	**Diameter [mm]**
ABS enclosure	117	117	57	N/A
Arduino board	53	69	4	N/A
Switch	12	20	15	N/A
LCD 2x16	10	71	27	N/A
Buzzer	N/A	N/A	8	22
Temperature Sensor	N/A	N/A	6	7
LED	N/A	N/A	8	5

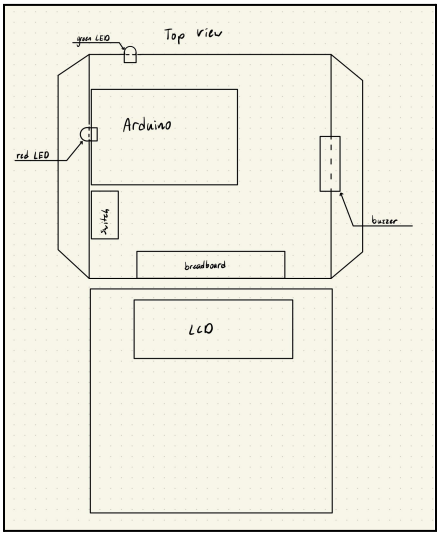
Design Alternatives

I. Design Process

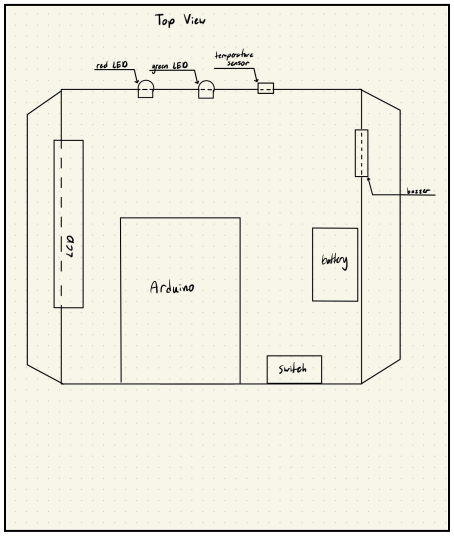
Using the dimensions above, I planned my design for the temperature sensor. In my first sketch, I decided to keep the temperature sensor inside the box to reduce the clutter of wires. I cut slits in the walls of the box to allow air to flow through the box so the room temperature was accurately measured. In this design, I also cut a hole for the LCD screen in the lid of the box for easy viewing if the box were mounted on the wall. I used Onshape to make a 3D model of this design. After completing my 3D model, I realized the difficulties that cutting holes in the walls would present, including that the box would not look neat and clean. Additionally, I decided that having the LCD screen visible through the lid would be inefficient. If I mounted it to the lid I would have to add extra wire length so that the lid could be removed. If I mounted it at the base of the box with a hole to look into the box, I would have to make sure no wires crossed in front of it.

After recognizing these challenges, I sketched two new designs. In both, I mounted the LCD on the side of the box. In one of the sketches, I did not include a breadboard. In the other, I moved some components around so that there was a wall without cutouts where I could mount a breadboard. Ultimately, I chose to use a breadboard to make the circuit wiring easier to troubleshoot. With the breadboard, I can adjust the placement of wires to make it neater inside the box. I used Onshape to create a 3D model of my chosen design. In Onshape, I also designed a battery holder.[1] The holder needed to hold the battery in but allow users to easily take it out to replace it. Based on these requirements, I made a holder with extrusions on the front that are big enough to hold the battery in but angled so that the battery could be snapped in.

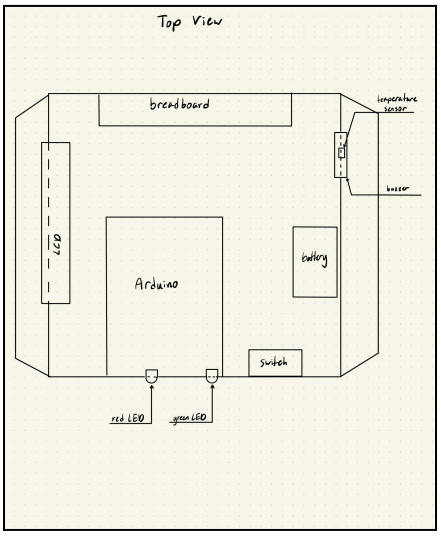
II. Sketches



First technical drawing.



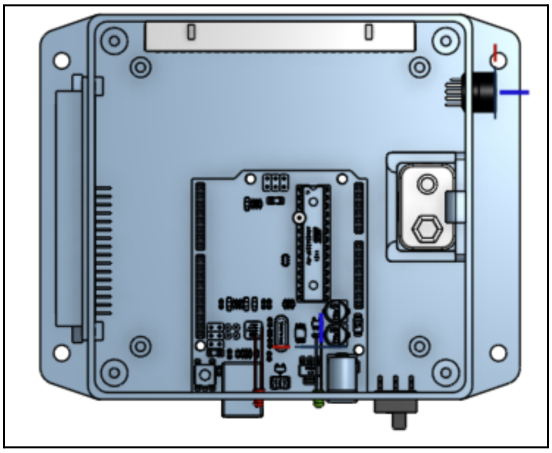
Second draft.



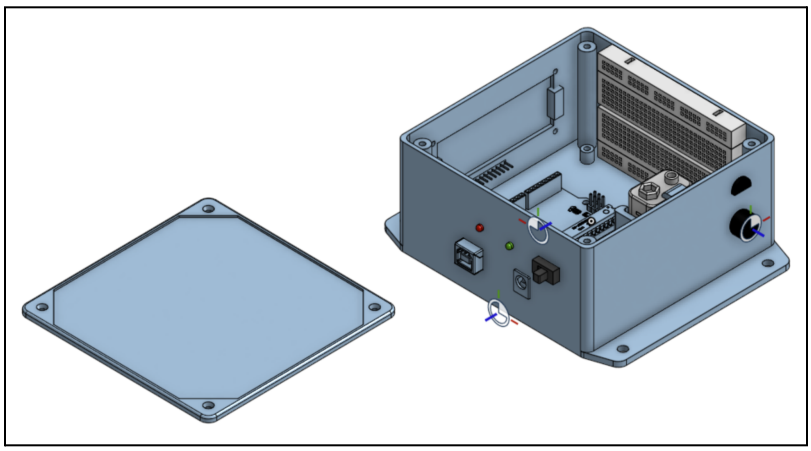
Final draft.

[1] See Figure 2 in Appendix.

III. CAD Drawing of Final Design



Top view.



Isometric View.

IV. Electrical Components

After selecting the layout of my electronic components in the box, I began the circuit wiring. The components are wired as shown in Figure 3 in the appendix. Because I did not have access to a machine room, I could not solder wires together. I connected wires by either putting them in the breadboard or using male and female terminals of jumper wires. I used electrical tape to make sure all components were safely connected. The jumper wires I used are 25 AWG. 25 AWG allows 0.457 Amps of current, which is higher than the amount of current flowing through the circuit. The power supply for the Arduino Uno is a 9V battery, which is what is recommended by Arduino. The purpose of the Arduino board is to code functions for the temperature sensor, LCD screen, red LED, and buzzer. I used analogRead to get the reading from the TMP36, then converted it to voltage, then to degrees Celsius, then to degrees Fahrenheit. I displayed the temperature in degrees Fahrenheit on the LCD screen by using lcd.print after initializing the LCD. I wrote an ‘if’ statement with a temperature range. When the temperature reading is outside the set range, the red LED will flash because I utilized digitalWrite HIGH, added a delay, then digitalWrite LOW, and added another delay. A tone of 1kH with a delay in between is sent to the buzzer when the temperature is out of range. View the code with comments in Figure 4 of the appendix.

V. Specification of Prototype

Requirement	Specifications
Operating voltage	5.8 V
Maximum current	354 mA
Power Rating	2.05 Watts
TMP 36 temperature range	-50 °C (0V) to 125 °C (1.75V)
LCD screen	16x2; Operates for 4.7V - 5.3V

Evaluation of Final Prototype

My final prototype accurately displays the temperature of the room. When the temperature is outside of the specified range, the red LED blinks and the buzzer sounds. Therefore, it effectively meets all feature requirements. Because of Covid-19, I was unable to access the machine room to cut holes in the plastic enclosure, so I reconstructed the enclosure out of cardboard. It is not as sturdy as plastic would be, but the dimensions and layout are still accurate, it is safe to operate, and necessary components are accessible from the outside. This year I also could not solder wires, but my circuit stays together well because I used male/female jumper wire connectors and electrical tape. I could not 3D print my battery holder, either, but in the future I hope to work with a 3D printer after learning the basics in this course. During this course I learned mechanical drawing techniques, how to use an Arduino board, circuit wiring, and how to use Onshape to make 3D models. Working on this project also required implementing many critical thinking and problem solving skills. I enjoyed the challenge and am proud of the final result.

Appendix

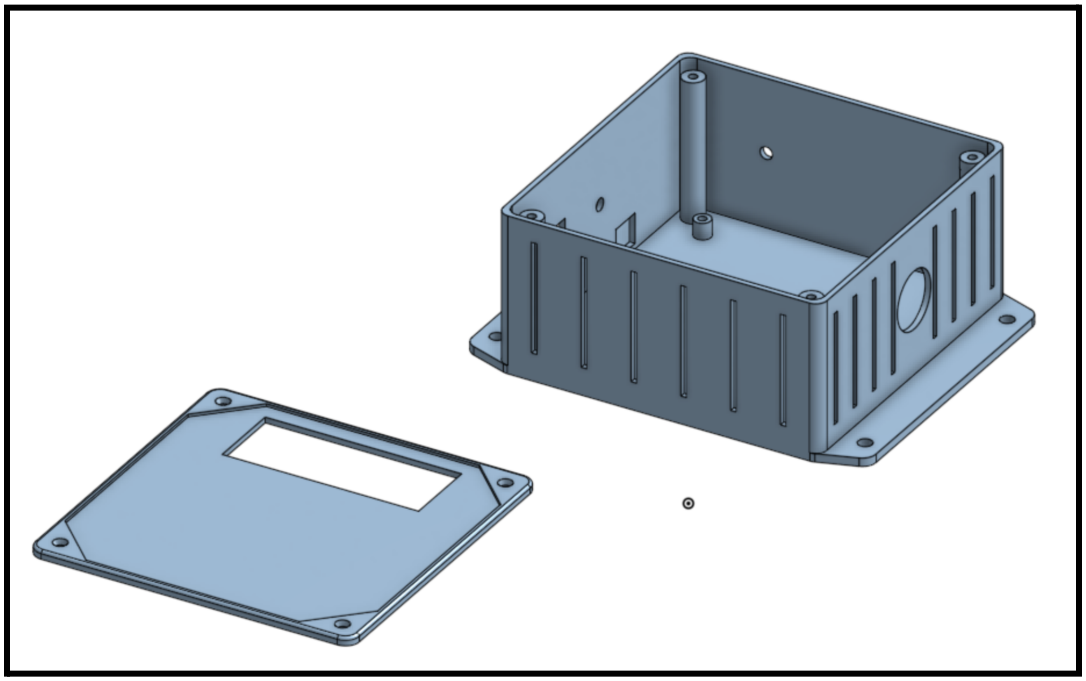


Figure 1. CAD draft of first hand-drawn sketch. Holes cut manually using measured dimensions.

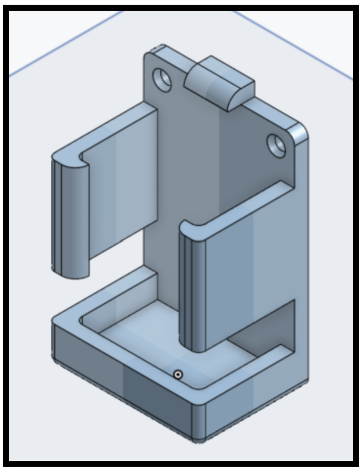


Figure 2. CAD 3D model of battery holder.

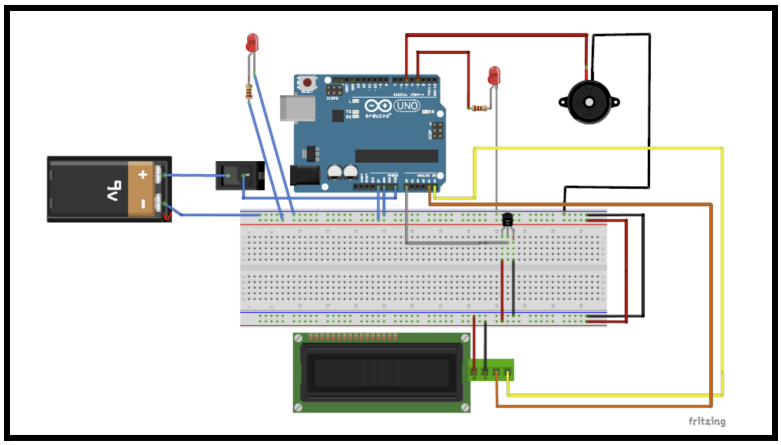


Figure 3. Wiring of electrical components.

```
int ADCV=A0;
int buzzer = 8;
int LED = 7;
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x20,16,2);  // set the LCD address to 0x20 for a 16 chars and 2 line display
void setup()
{
  pinMode(buzzer,OUTPUT); /*set buzzer -pin 8 as an output*/
  pinMode(LED,OUTPUT); /*set LED -pin 7 as an output*/
  lcd.init();                      /* initialize the lcd */
}
void loop()
{
 int VoltageRead=analogRead(ADCV);
  float Vo=(VoltageRead/1023.0)*5;
  float TempC=(Vo-0.5)*100;

  float TempF = (TempC * 9.0 / 5.0) + 32.0;

  delay(1000);

 lcd.backlight();
  lcd.setCursor(1,0);
  lcd.print("Temperature [F]"); // Print a message to the LCD.
  lcd.setCursor(1,1);
  lcd.print(TempF);
if (TempF > 80 || TempF < 75)
{
  tone(buzzer, 1000); /*send 1KH sound signal ...*/
  digitalWrite(LED, HIGH); /*...turn on LED...*/
  delay(500); /*...for 1 sec*/
  digitalWrite(LED, LOW);
  tone(buzzer, 1000);
  delay(500);
  noTone(buzzer); /*stop sound...*/
  digitalWrite(LED, HIGH); /*...turn off LED...*/
  delay(500); /*...for half a sec*/
  noTone(buzzer);
  digitalWrite(LED,LOW);
  delay(500);
}
}
```

Figure 4. Code for Arduino board. Reads temperature of the room with TMP36, displays temperature on LCD screen, sets temperature range with an “if” statement, and makes LED blink and buzzer beep when temperature is outside of range.