Max Karambelas
April 29, 2021
mkaramb@bu.edu

Engineering Technical Report

Design Process of Room Temperature Monitor

Summary: Report Topic, Approach Outline, Outcomes/Significance

The topic of my report was to create a room temperature monitor and place it into a CAD. The approach to the task was to learn different methods of design to determine the best layout for the components in the CAD. Then to determine the correct AWG to use and the best wiring methods to utilize in the project. The outcome of this project was that a circuit was created that did fully function as intended. With an LCD that displayed accurate temperatures and a buzzer/red LED system that sounded an alarm when temperature left a certain range. The circuit also was controlled by an arduino and a switch. But, due to restrictions some design components like the CAD and wiring methods had to be altered to be more manageable without the proper equipment being available. These results show that while it is possible to create a functioning circuit under less than ideal circumstances oftentimes the circuit will have to be less streamlined and professional.

Introduction: Explanation of Circuit, Motivation for Project, Aims of the Project, Purpose of Report

The topic of our report was to design a circuit that could measure the temperature of its surroundings using an Arduino Uno and other components. The system would be turned on and off by a switch which led to a 9V battery and would have a green LED that would indicate whether the system was on or off. If the temperature was recorded out of the safe range programmed into the Arduino a buzzer would sound an alarm and a red LED would flash until the temperature returned to the specified “safe” range.

This system could be utilized to check if someones heating or cooling systems are working or not. It could monitor the temperature and notify the user if the temperature doesn’t match up with the temperature their system is displaying.

The aims of this project were first to design a system that accurately read the temperature of its environment and notified the user when the temperature left a specified range. Another aim was to fit our entire system into an enclosure with openings on the side of the enclosure to see the temperature sensor, LEDs, LCD, and buzzer. We also wanted to provide openings to the two ports of the Arduino Uno so it could be reprogrammed without taking the lid off of the CAD.

The purpose of this report is to outline the design alternatives considered throughout the design process. To outline the choices made and provide reasoning as to why those choices were best. This report will also provide an evaluation of the results to determine how well the goals of the project were accomplished. As well as an appendix at the end that will provide relevant information to by design processes that is too long to include in the main part of this report.

Design Alternatives Considered:

In **Table 1** is listed all of the components used in the CAD enclosure and provides the precision measurements of each part. Freehand identification sketches of each component can be found in the appendix in **Figure 5**.

Table 1. Names of All Components in Design and Precision Measurements

Part Name	ID Sketches	Length(mm)	Width(mm)	Diameter(mm	Height(mm)
Buzzer	**Figure 5**	N/A	N/A	21.62	10.73
Red (LED)	**Figure 5**	34.42	N/A	5.71	N/A
Green (LED)	**Figure 5**	34.42	N/A	5.71	N/A
Arduino Uno	**Figure 5**	68.43	53.21	N/A	N/A
Switch	**Figure 5**	21.13	15.39	N/A	N/A
16x2 LCD	**Figure 5**	70.97	24.97	N/A	N/A
9V Battery	**Figure 5**	25.31	15.16	N/A	N/A
TMP36	**Figure 5**	18.71	4.63	N/A	N/A
Breadboard	**Figure 5**	55.32	85.16	N/A	N/A

The design process starts by drawing three freehand sketches of possible layouts that could be utilized in the project. These three sketches are shown in **Figure 1.** After learning more about the project the design shown by the right picture in **Figure 1** was chosen to be the final product. And an isometric view of that chosen CAD layout can be found in **Figure 2** where you can also see the layout of the components in the final design. In this final design the arduino was placed on the floor of the enclosure so it could be close to all of the components it needed to connect too. The other components were put on the walls so they could be utilized from the outside. So the switch could be flipped without taking the top off, so the LEDs and LCD could be visible from the outside as well, and so the TMP36 could read accurate temperatures and the buzzer could be heard.

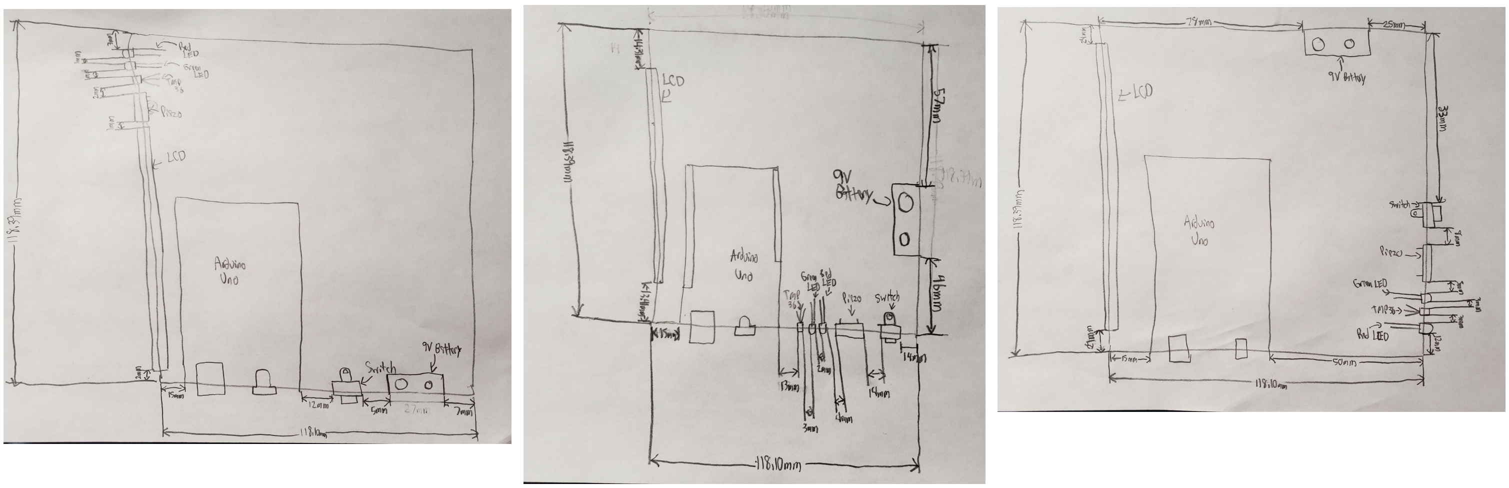

Figure 1: The three preliminary freehand layout sketches above

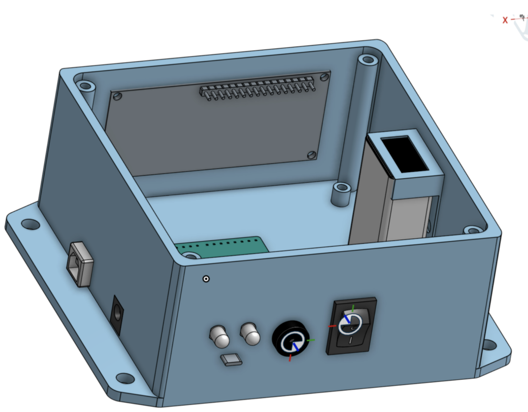

Figure 2: Shows the placement locations of the components in an isometric view of my CAD enclosure

An Arduino Uno was used in this product for multiple reasons. For one the arduino uses an open source platform and a free integrated development environment(IDE), which makes it easier for someone with less experience to program their circuit. The arduino also has prebuilt female pins which make all of the wiring in the circuit more manageable. The arduino is also beneficial for inexperienced creators because there are numerous online examples and applications readily available. Finally, the Arduino Uno is relatively inexpensive, perfect for a school project.

When wiring it is important to understand why the decisions were made for this circuit specifically. First, spade connectors were used to connect the switch to the battery, breadboard, and arduino. This is because due to covid restrictions the materials necessary to perform soldering were not available. So instead, two spade connectors were used because it was the easiest method to do from my room. The final wiring diagram can be seen below in **Figure 3**. Jumper wires were utilized to connect the components in this project because they had an American Wire Gauge(AWG) of 22. This AWG value was chosen because the measured maximum current of the circuit was 44 milliamps. Taking into account that the wire should be able to handle 125% and 175% of measured maximum resistive and inductive loads then an AWG of 22 matched that through calculations using ohm's law and the inclass table. Also, 22 AWG is ideal for use on breadboards and arduinos so it was the natural choice for my project since an arduino uno was being used to control the circuit and the wires fit securely into the ports without any soldering. Since this method was unavailable.

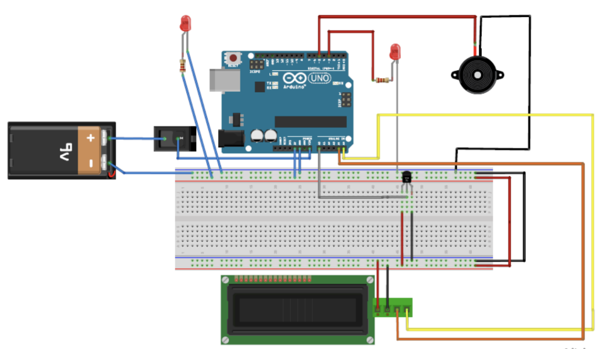

Figure 3: Wiring Diagram for Temperature System

A 9V battery was used as the main power supply for multiple reasons. For one, the arduino uno is recommended to take anywhere from 7V-12V, and the 9V battery falls within this range. Also, the 9V battery is an internal power supply. Due to the goals of this project it wouldn't work to use an external power supply because then the CAD assembly wouldn't be fully portable. Finally, the battery is cheap and small and therefore also perfect for the project's goals.

In the appendix in **Figure 4** is the code used for my project. The code basically converts the analog readings from the TMP36 and converts that to degrees fahrenheit and displays that temperature on the LCD every second. Then it takes that recorded temperature through an if-else statement which notifies the user with a buzzer and red LED if the temperature falls outside 70F-75F.

The specification of the prototype is that the operating voltage was 9V simply because the power source was a 9V battery. And, from the measurements taken from a DMM it was found that the maximum current was 44 milliamps.

Evaluation of Results: Indicate Level of Success, Summarize Report, Acknowledge Limitations and Recommend Solutions, Highlight Significance of Work

The design objectives for this project were mostly achieved. The system did output an accurate temperature which was confirmed by the readings on a DMM, which were all very close to the readings of the system. The system also displayed its temperatures and was able to notify the user when the temperature went outside of the specified range. But, due to restrictions some planned goals of the project were not achieved. For one, a battery holder was not able to be created and a CAD enclosure couldn't be used due to these restrictions. So cardboard was a substitute for the enclosure and battery holder. Also, since the necessary materials for soldering weren't available the wiring ended up being messy and therefore a breadboard had to be utilized in the end to make the circuit more manageable. So while there were restrictions due to covid which limited some aspects of the project all of its functional components worked as intended. So in the end any recommendations for future work form this report wouldn't be well informed since the project was completed under extraordinary circumstances.

The significance of this report was to show the thought process throughout the construction of the circuit. To explain why the decisions were made through the design process and to outline the alternatives considered. This would allow someone to not only recreate this project but to build off of it, because they would have the knowledge necessary to make possible improvements and modifications to the project.

Supporting Materials/Appendix:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
int LED = 3;
int buzzer = 5;
LiquidCrystal_I2C lcd(0x20,16,2); // set the LCD address to 0x20 for a 16 chars and 2 line display
int sensorPin = 0;
void setup()
{
lcd.init(); // initialize the lcd
Serial.begin(9600);
pinMode(LED, OUTPUT);
pinMode(buzzer, OUTPUT);
}
void loop()
{
int reading = analogRead(sensorPin);
float voltage = (reading * 5.0)/1024;
float temperatureC = (voltage - 0.5)*100;
float temperatureF = (temperatureC*9.0/5.0)+32.0;
lcd.backlight();
lcd.setCursor(1,0);
lcd.print("Temperature(F): ");
lcd.setCursor(1,1);
lcd.print(temperatureF);
delay(1000);
lcd.clear();
if (temperatureF < 70 || temperatureF > 75)
  {
    digitalWrite(LED, HIGH);
    delay(100);//wait for 100ms
    digitalWrite(LED, LOW);
    delay(100);//wait for 100ms
    tone(buzzer, 1000); // Send 1KHz sound signal...
    delay(1000);        // ...for 1 sec
    noTone(buzzer);     // Stop sound...
    delay(1000);        // ...for 1sec
  }
  else
    {
      digitalWrite(LED, LOW);
      digitalWrite(buzzer, LOW);
    }
}
```

Figure 4: Arduino Code

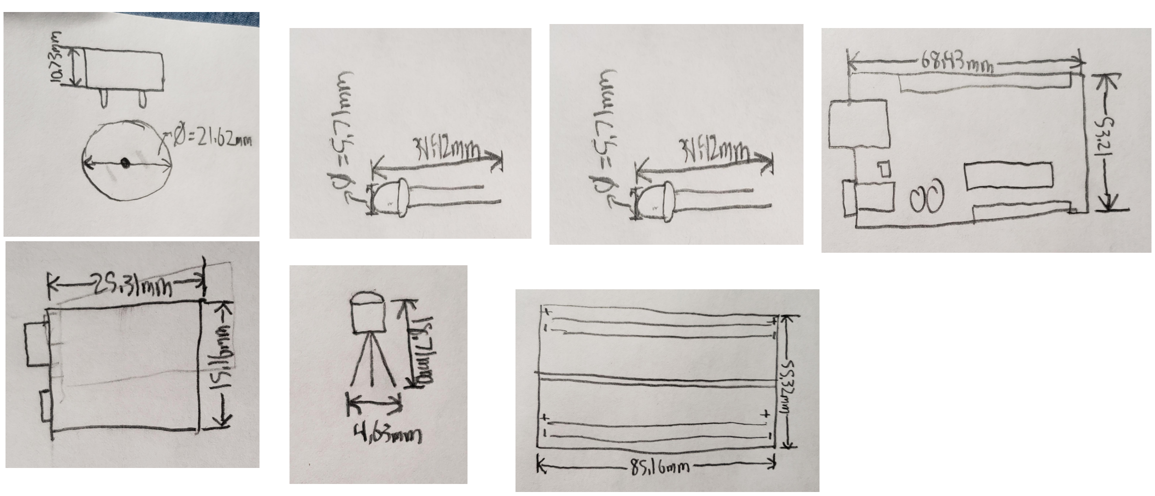

Figure 5: Component Identification Sketches