Matt Dorrycott

Dr. Dickerson

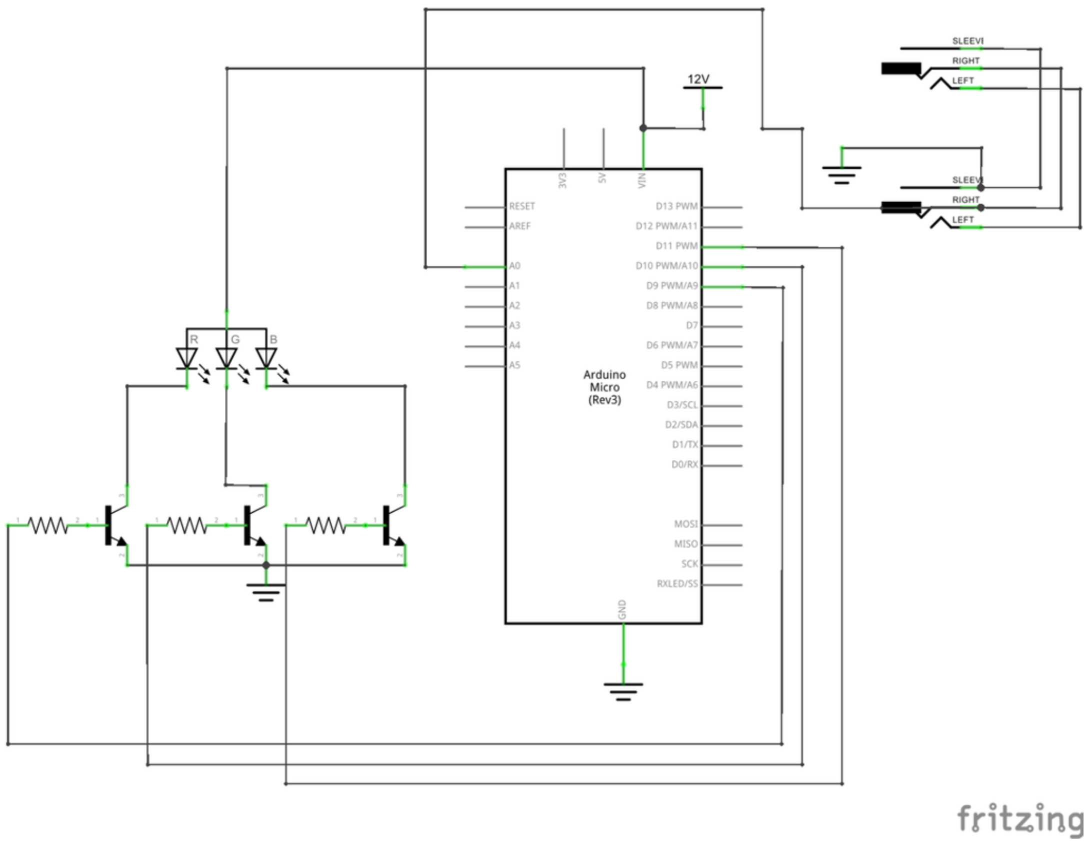
ECE 1895- Junior Design

18 December 2022

Design Project 3- Audio Reactive Lights

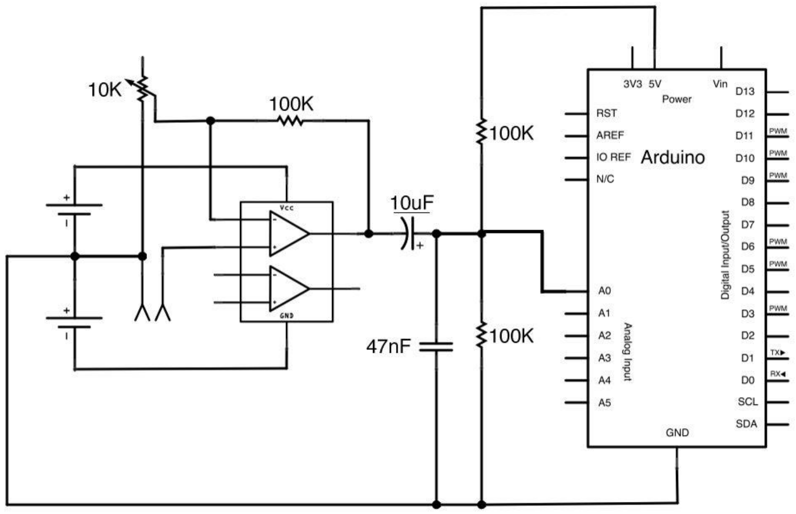
Design Overview

The goal of my project was to create a circuit for transforming an audio source into LED lighting corresponding to the sound waves. While my original design featured an enclosure, 2 3D-printed frames for LED strips (left and right channels), a PCB, and frequency-based lighting, my final design ended up featuring none of that due to time constraints and unfortunate planning. However, I was able to produce a fully functional circuit utilizing an ATMega328P for reading an audio signal and writing signals to the LED strip. The original design of my project was based on the schematic shown below.



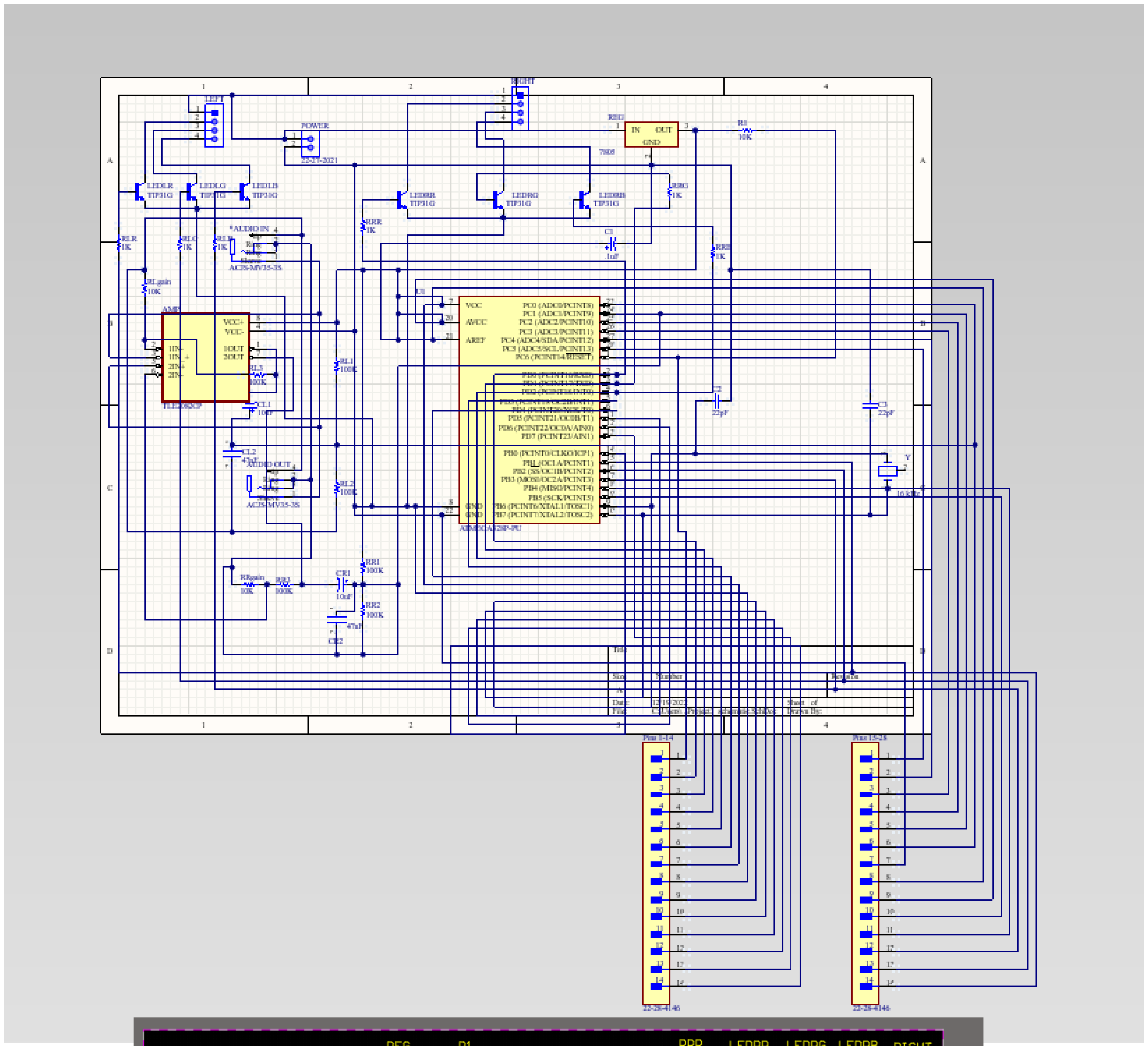
https://www.instructables.com/Music-Reactive-Multicolor-LED-Lights/

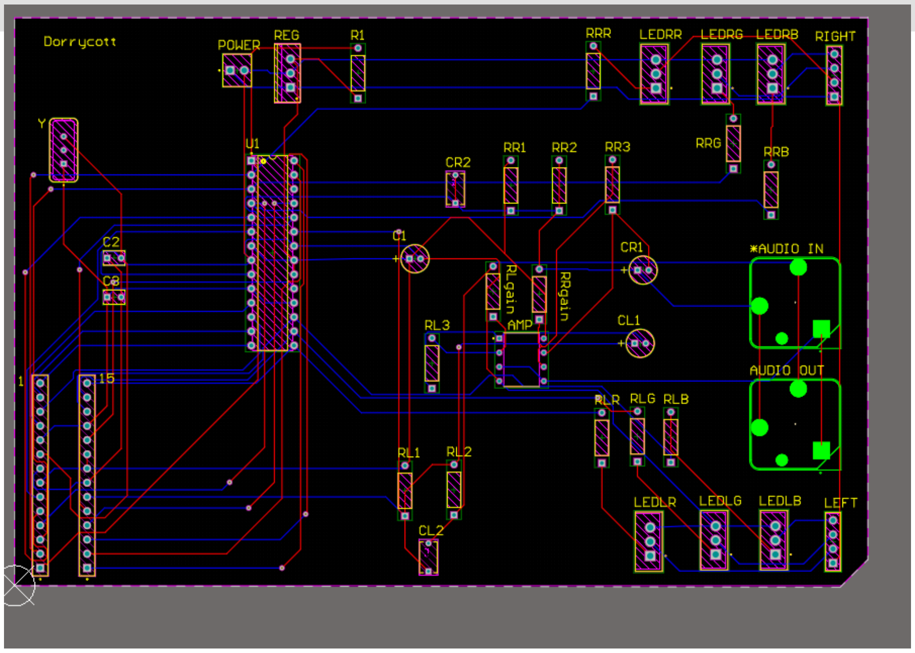
One oversight in the original drafting of my design was the fact that ATMega328Ps are unable to read analog input less than 0V. To compensate for this, I had to utilize a TLE2082CP which was given to me by Dr. Jacobs who helped me significantly with the design verification process. This op-amp was used to offset the audio signal from -2.5V -2 .5V to a more acceptable 0V -5V range. I also applied a small gain using a 10 KOhm resistor to ensure that the signal would never leave those bounds. The signal that was read for this circuit was only the left audio channel from a DC audio jack. The figure below shows the circuit used to offset the voltage into the ATMega328P.



https://www.instructables.com/Arduino-Audio-Input/

I had also originally planned to use both audio channels, but unfortunately there was a miscommunication when ordering parts and there was not enough time to correct this. The PCB created for my project did allow for the right audio channel to be used provided another RGB LED strip could be plugged into the 4 pin connector on the board. Pictured below is the full circuit schematic that was used to design my PCB, as well as the original PCB design itself.


LEFT
POWER
RIGHT
REG
7805
R1
10K
LEDLR
LEDLG
LEDLB
TIP31G
LEDRR
LEDRG
LEDRB
AUDIO IN
AUDIO OUT
AMP
TLE2082CP
U1
ATMEGA328P-PU
Pins 1-14
Pins 15-28
22-28-4146
Y
16.0MHz
C1
C2
C3
22pF
CR1
CR2
CL1
CL2
RRgain
RLgain


Dorrycott
POWER
REG
R1
RRR
LEDRR
LEDRG
LEDRB
RIGHT
Y
U1
CR2
RR1
RR2
RR3
RRG
RRB
C2
C8
C1
RLgain
RRgain
CR1
AUDIO IN
RL3
AMP
CL1
AUDIO OUT
1
15
RLR
RLG
RLB
RL1
RL2
CL2
LEDLR
LEDLG
LEDLB
LEFT

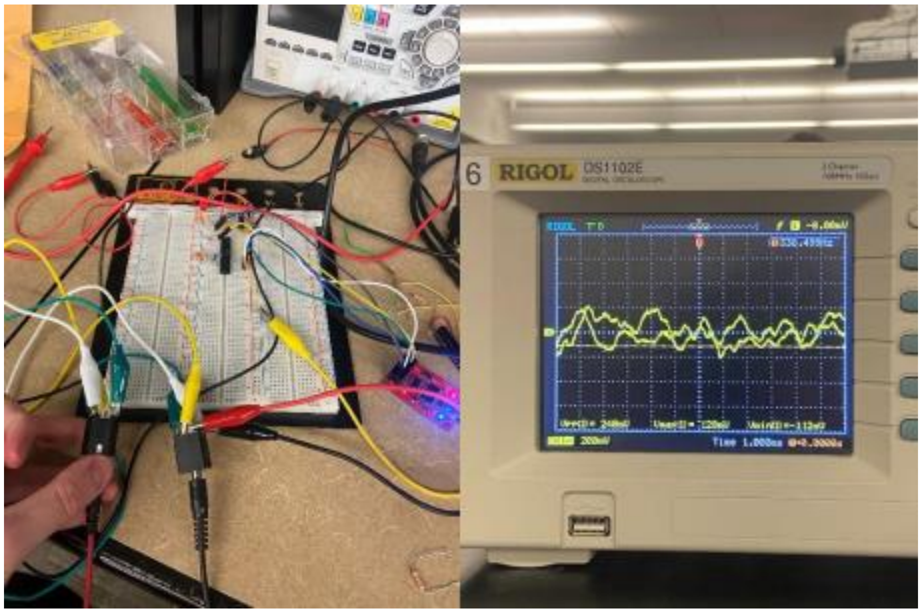
There are 3 main inputs to the device as noted by “Right” and “Left” for the LED strips and the “AUDIO IN” ACJS-MV35-3S DC audio jack which uses stereo audio with pins known as “tip-ring-sleeve.” The “AUDIO OUT” audio jack is the same. The “tip” pin is known as the left audio channel in a stereo configuration, and the “ring” is the right channel. After the signals are sent through the OP-AMP circuit, they are measured on the analog input pins of the ATMega328P.

The 5 meter, RGB LED strips used require 12V and about .5 amps to work properly. Unfortunately, the power adapter I had ordered was 12V, 2A which was more power than necessary and caused some overheating issues on the LEDs. To fix this issue, I instead opted to use the DC power supply in the ECE labs for a safer circuit. The LED strip also required 3 NPN transistors for the red, blue, and green inputs to the strip. These transistors took the 5V maximum output of the ATMega328P’s output pins as a gate voltage to ground the 12V that were inputted to each LED color on the strip. By writing different values to the gate on a scale from 0-255, I was able to control the color intensity of the strip and manage the color output on the strip.

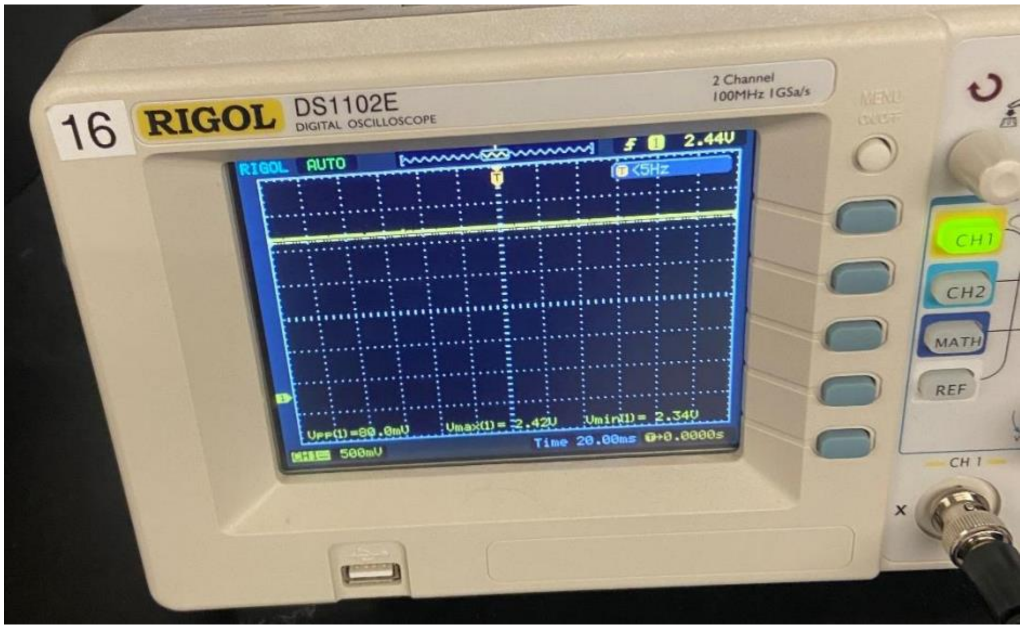
By combining all of these elements, I was able to create a unique design to be implemented on a PCB. Along with this PCB was its enclosure which was meant to be a laser cut box design from https://www.festi.info/boxes.py/index.html with holes specifically for the audio jacks and LED strip pins. I also included 2 3D printed frames in my design for the LED strips which could be hung onto a wall or even a speaker.

Preliminary Design Verification

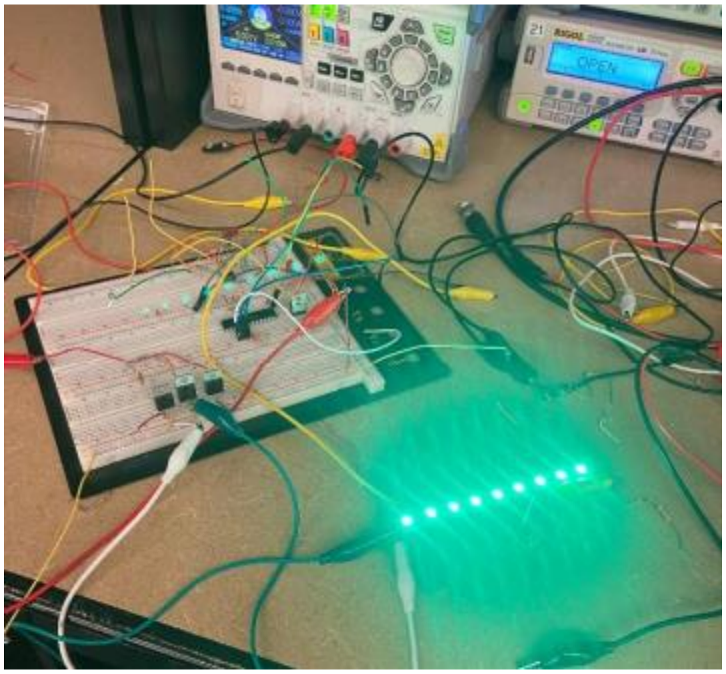
To verify my design, I tested the Op-Amp circuit, audio jacks and LED strip circuits individually. For testing the audio jacks, I connected the two audio jacks to make sure that the audio quality would be listenable if the audio was transferred from one jack to another. I also viewed the waveform of the audio signals on an oscilloscope, as shown below.



I then proceeded to verify the 2.5V offset circuit by connecting the output of the circuit to the oscilloscope.



Finally, I verified the LED circuit in my design by outputting 5V from a digital pin of the microcontroller to the gate of a BJT which was connected to an LED strip and proved that the microcontroller would work with the LED strip. I demonstrated this below with green on the LED strip.

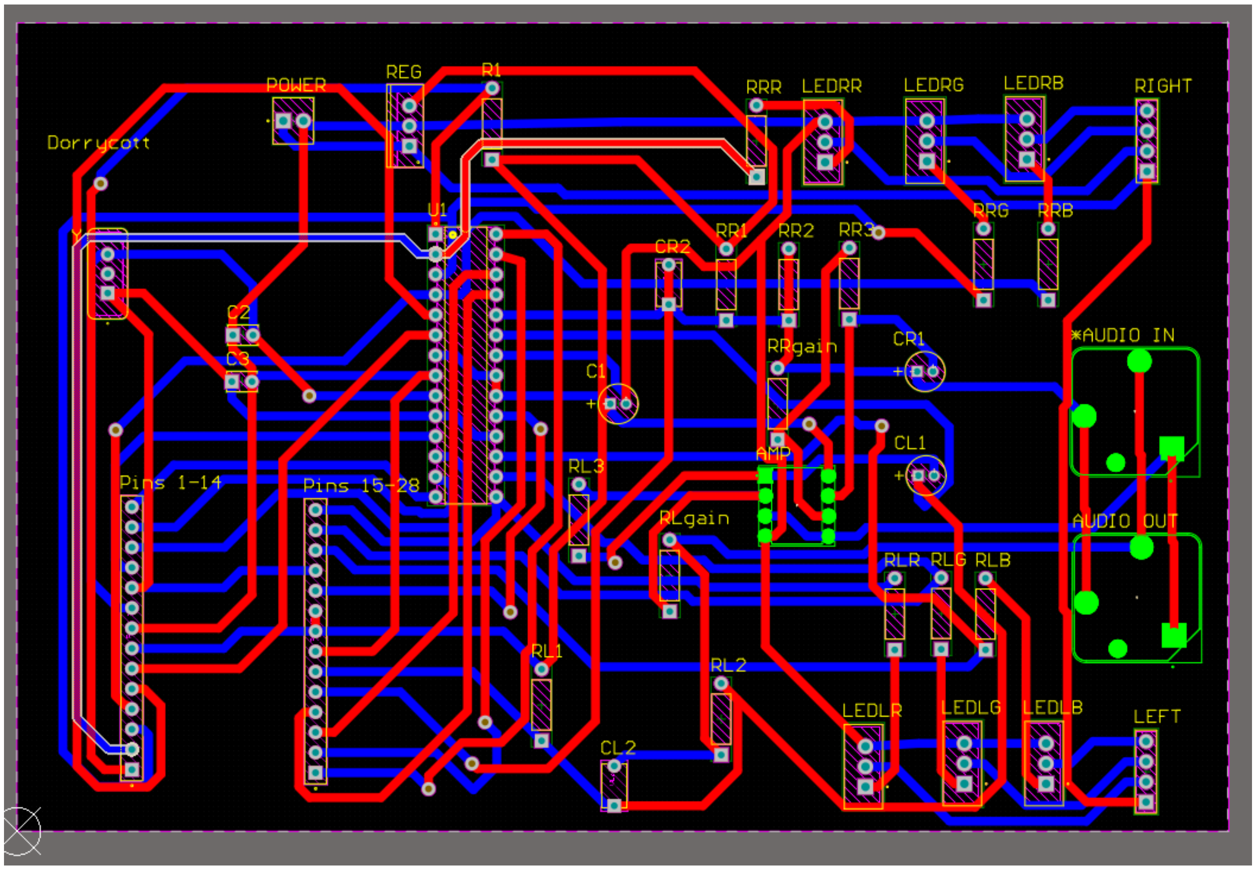


Design Implementation

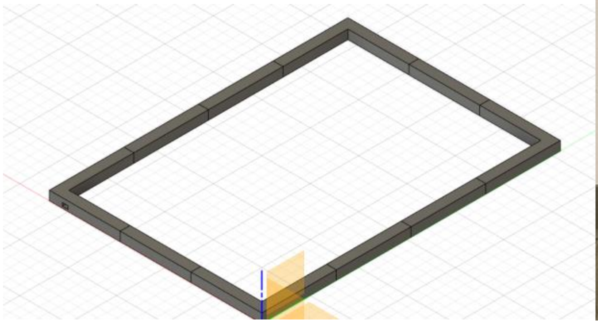
Going into the final week of the project, I had received all the parts for my design except for the PCB, which I was notified would not be received in time for the due date of the project. The parts for my design included:

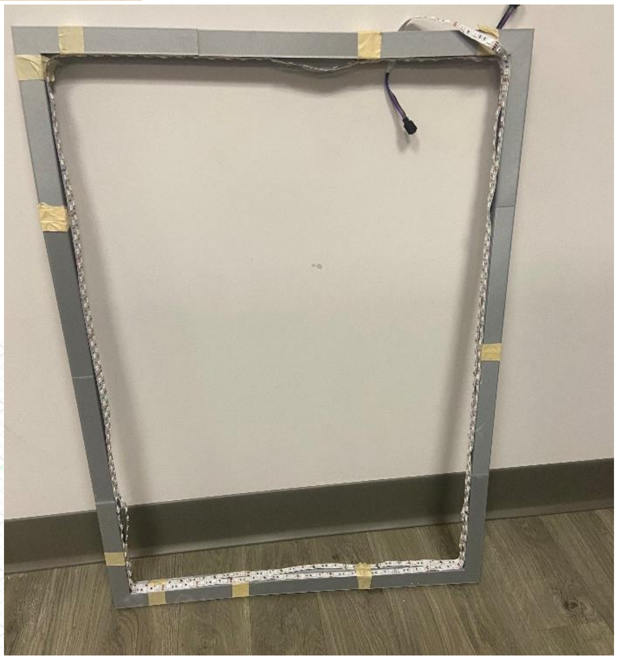
- TLE2082CP Op-Amp
- ATMega328P microcontroller
- 2 ACJS-MV35-3S DC audio jacks
- 6 NPN Transistors
- 2 LED strips
- 12V 2A Power Adapter
- 7805 Voltage regulator
- 16 MHz Crystal
- Various Pins and connectors for outputs and inputs
- Various capacitors and resistors for microcontroller and amplifying circuits.

After being notified about my PCB not arriving, I proceeded to redesign my PCB to be milled in SERC immediately. Shown below is the resubmitted design.

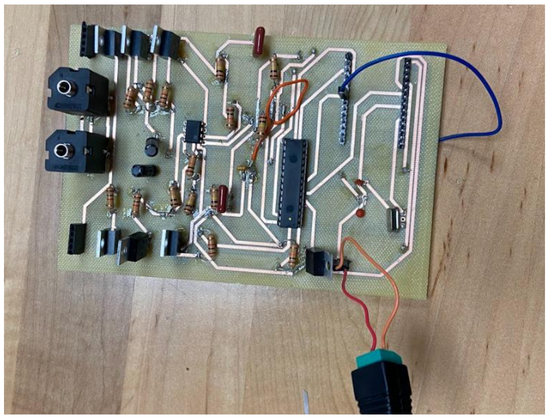
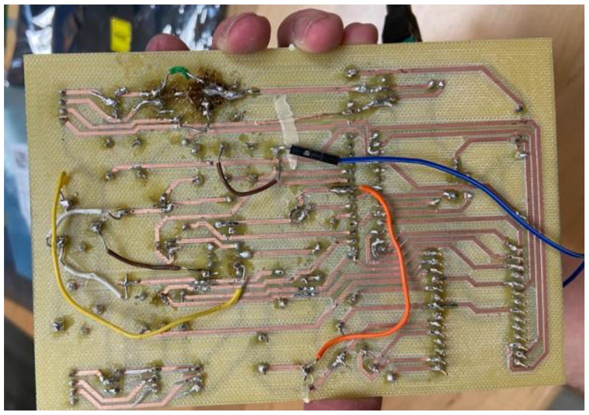


During the time it took for my PCB to be milled, I designed the frame for the LED lights. This frame was printed into 10 parts across 2 prints to fit in the 3D printer's dimensions. It took 32 hours total for the 2 prints. The outer dimensions of the frame were 30.5in x 21in, with 1 inch thickness on the parts, which fits one of my home speakers specifically. There is also a hole for running the LED strip's cable through. The parts were glued and taped together. Another frame would have been made if I more time and had received the second LED strip for the right-side audio channel. The finished frame, CAD drawing, and separate parts are shown below.

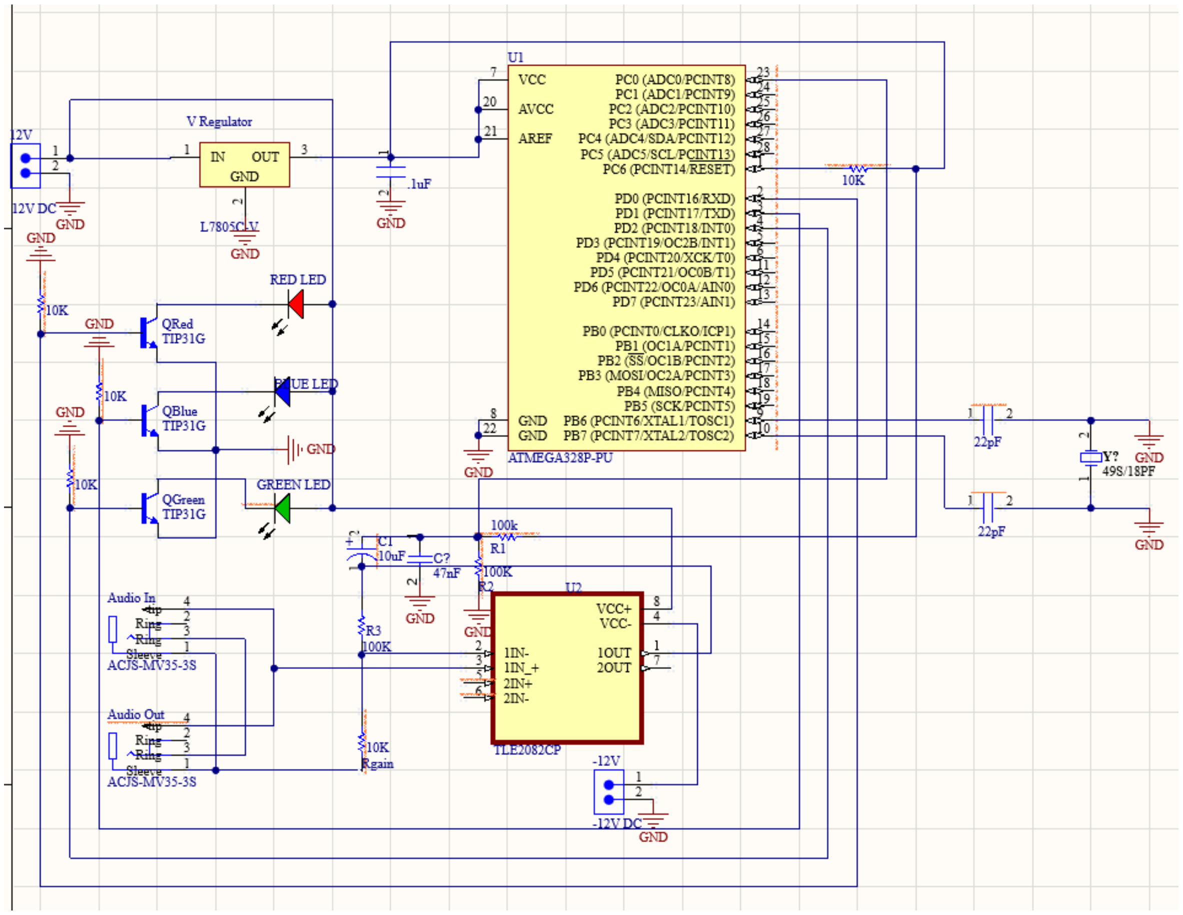




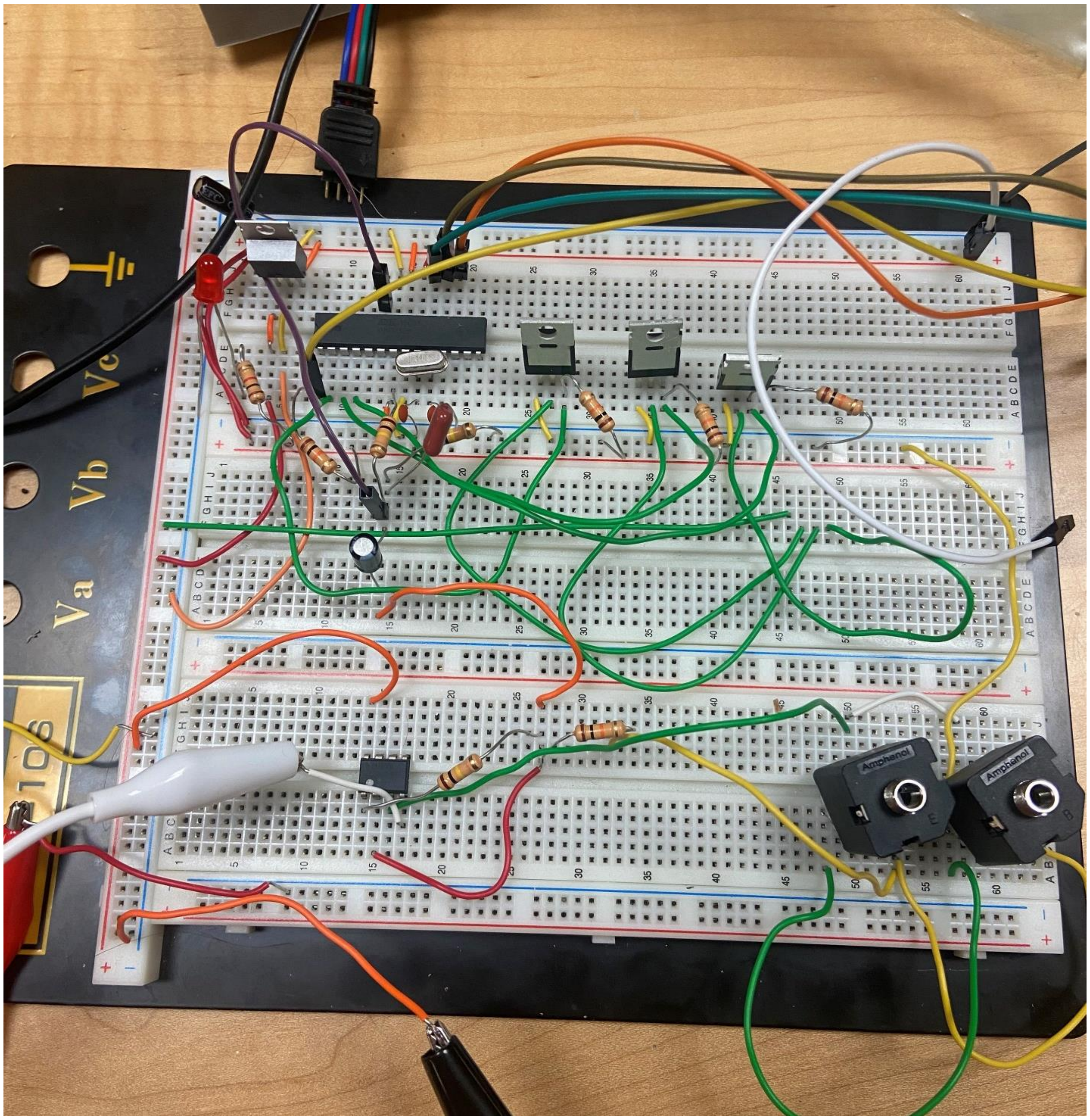
Unfortunately, due to errors with the mill in SERC, they were able to get the board to me a couple of days after I submitted my PCB design, on Friday 12/17. This delayed my implementation process until that day when I began soldering my parts onto the board. I worked all day on soldering my board as I noticed some issues with my original circuit. In implementing my PCB, I had accidentally swapped the + and – inputs to the Op-Amp which created complications when soldering my PCB. My soldering skills are also not the best so there were many torn traces that I had to rewire and as the process continued. At the end of the day however, my PCB was nonfunctional.

Instead of continuing with the failed PCB. I decided to instead recreate my circuit on a breadboard. Luckily, I had ordered some spare parts in case of an emergency, so I was able to put together the breadboarded circuit using the schematic shown below.

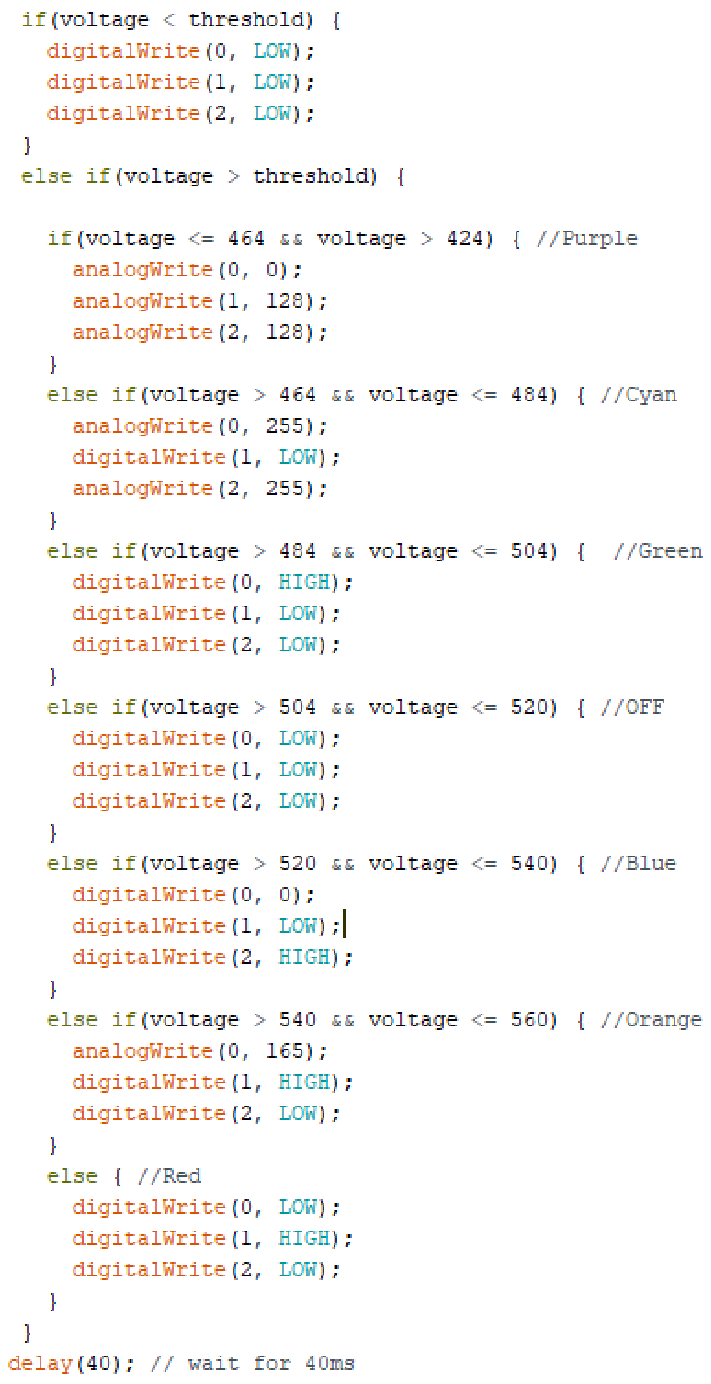


The fully implemented breadboard circuit is shown below. Note that the 12V voltage was accidentally disconnected in this image and should have been connected to the left power rail rather than appearing to be connected to the bottommost ground rail. The LED strip was also disconnected by accident as well.



Because I resorted to a breadboard circuit and because of time constraints, I was unable to build an enclosure. If the PCB was properly completed, I would have been able to obtain measurements for the holes to be placed on the box enclosure.

The final part of the implementation for my design was the code itself. The code I created was rather simple and was based around reading the analog amplitude of the signal provided from the Op-Amp circuit. Because Arduino uses binary encoding for the analogRead() command, I based the input voltage around 512, which equates to 2.5V, or the offset voltage of the amplifying cicrcuit. From there, I set thresholds for each color. The delay period used for changing colors of the LEDs was set to 40ms. This means that the code would take the voltage from the amplifier about every 40ms, and because sound waves are sinusoidal, this would be either a positive or negative voltage. I designed my code from colder colors to warmer colors as the signal got louder. So, the quietest signals would output blue and green, then orange and cyan, then finally red and purple. In hindsight, the orange and purple would be switched, but I am an engineer, not an artist. Shown below is an excerpt from the code I used which shows the color switching technique.



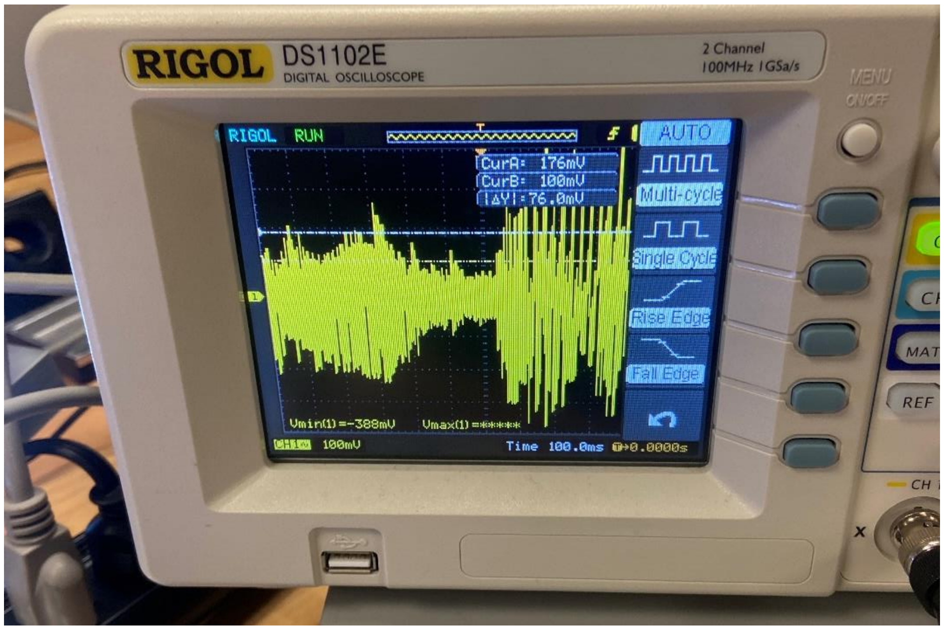
```
 if(voltage < threshold) {
   digitalWrite(0, LOW);
   digitalWrite(1, LOW);
   digitalWrite(2, LOW);
 }
 else if(voltage > threshold) {

   if(voltage <= 464 && voltage > 424) { //Purple
     analogWrite(0, 0);
     analogWrite(1, 128);
     analogWrite(2, 128);
   }
   else if(voltage > 464 && voltage <= 484) { //Cyan
     analogWrite(0, 255);
     digitalWrite(1, LOW);
     analogWrite(2, 255);
   }
   else if(voltage > 484 && voltage <= 504) {  //Green
     digitalWrite(0, HIGH);
     digitalWrite(1, LOW);
     digitalWrite(2, LOW);
   }
   else if(voltage > 504 && voltage <= 520) { //OFF
     digitalWrite(0, LOW);
     digitalWrite(1, LOW);
     digitalWrite(2, LOW);
   }
   else if(voltage > 520 && voltage <= 540) { //Blue
     digitalWrite(0, 0);
     digitalWrite(1, LOW);
     digitalWrite(2, HIGH);
   }
   else if(voltage > 540 && voltage <= 560) { //Orange
     analogWrite(0, 165);
     digitalWrite(1, HIGH);
     digitalWrite(2, LOW);
   }
   else { //Red
     digitalWrite(0, LOW);
     digitalWrite(1, HIGH);
     digitalWrite(2, LOW);
   }
 }
delay(40); // wait for 40ms
```

Design Testing

Testing my design happened as I continued work on my implementation rather than all at the end. Once I had completed the circuit, I modified my code multiple times to tune it to max volume output of my computer's audio. The typical sound waves of my computer actually ranged from 250mV to -250mV, so I tuned the voltage thresholds repeatedly for the color coding.

The best method for testing my design was simply playing a song and changing the volume of my computer. At the lowest detectable volume for my computer, it was clear that only the quietest colors were being output. As I raised my computer's volume, the louder colors started to appear more frequently. Another good test was to mute the audio from my computer which would effectively set the voltage to 2.5V (with offset) and cause the LED strip to stop blinking entirely.

As I changed the volume of my computer to monitor the LED colors, I also viewed the peak-to-peak voltages of the sound waves on an oscilloscope, to make sure that as the waveform increased amplitude, the colors were changing. You can see the waveform of the music being played in my demonstration video. An example waveform using a song is shown below.



Summary

With more time and better planning, I believe the full potential of my project could be realized, But I am proud of the result of the breadboarded prototype. In the end, my final goal was achieved in creating a design which would allow audio waves to be transformed into various colors on and RGB LED strip. I also managed to 3D print a frame for the LED strip as well, that makes for a nice lighting effect.

In the future, I would like to continue working on this design with a proper PCB and power source rather than a DC power supply. I would also like to see if I could transform the code to be based around the input waveform’s frequency, rather than the amplitude, this would make the LEDs react to pitch. The code could definitely be perfected in some other ways to be more applicable to a wide range of audio sources, rather than just my computer’s built in sound card at max volume. I would also like to implement the right audio channel with another frame and a duplicated amplifying circuit for the ring pin of the audio jack. Along with all of this would be a proper enclosure for the PCB to house the device and make it more applicable to a real-world scenario.

Although it became stressful towards the end of its implementation, I felt a great deal of relief once I achieved a functional prototype. I hope to one day be able to achieve an improved version of this design which could be implemented on my own speaker system. I would like to thank Dr. Dickerson for being very accommodating for turning in this project, all of the TAs for their help, and the people in SERC who helped me thoroughly with printing the PCB (although it went unused) and 3D printing the frame.

Presentation: https://youtu.be/z3eDlfCXLJQ

Secondary Demonstration (flashing lights warning): https://youtu.be/N1wE4fE3E68