

LED Color Organ

Final Project Report



Mark Vrablic

6.101

Table of Contents

1. Abstract
2. Introduction
 - 2.1 Project Inspiration
 - 2.2 Project Overview
 - 2.3 Goals
3. Design
 - 3.1 Block Diagram
 - 3.2 PCB Modules
 - 3.2.1 L+R Adder
 - 3.2.2 Band-Pass Filter Array
 - 3.2.3 Rectification and Smoothing
 - 3.2.4 Ramp Generator
 - 3.2.5 Voltage to PWM
 - 3.2.6 TLC5940 and Arduino
 - 3.2.7 Multiplexer
 - 3.2.8 FET Array
 - 3.2.9 Charge Pump
 - 3.3 Breadboard Modules
 - 3.3.1 Rectification and smoothing
 - 3.3.2 Cooling the LM7812 Array
 - 3.4 PCB Overall Schematic
 - 3.5 PCB Layout
4. Implementation
 - 4.1 Challenges
 - 4.2 Further Work
5. Conclusion

1. Abstract

This project recreated and updated the design of the analog “color organs” popular in the 1970s. The final product changes the location of brightness and color across 5 strips of RGB LEDs based on an audio input. This is accomplished with an array of band-pass filters, PWM generators, and MOSFETs. It is built on a PCB with the option to also multiplex digital effects using the audio, enabling ambient lighting when there is nothing playing and other complex digital effects when there is music. The organ is currently used to control LED strips mounted on the ceiling of an East Campus bathroom. A video of it in operation is available at https://youtu.be/HUw1-Kxq9_U.



2. Introduction

2.1 Project Inspiration

This project was inspired by the “Color Organs” popular in the 1970s. These contained a microphone or aux input and would create visualizations of audio in real time. There is no current equivalent for these products, as any effects they were capable of displaying can be easily recreated on a TV or computer screen through software.

From a 1970s Radio Shack catalog¹:

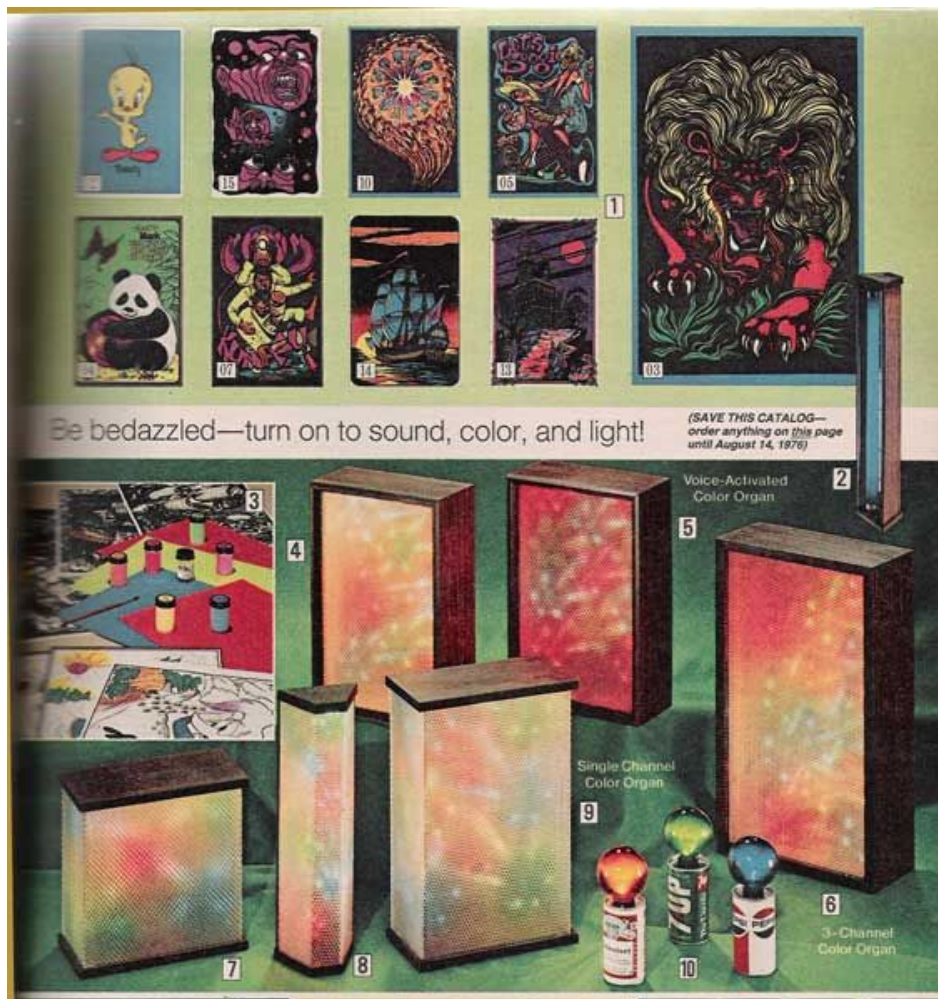


Figure 1: Advertisement for color organs from the 1970s

¹ Scanned by Plaid Stallions (<http://www.plaidstallions.com/images/color.jpg>)

2.2 Project Overview

The goal of this project is to update the hardware based color organ. Using arrays of band-pass filters, PWM generators, and MOSFETs, this organ converts audio input to a light show capable of controlling five external variable length strips of RGB LEDs. An Arduino and PWM expansion chip are also attached to the final PCB to enable ambient, digitally controlled effects when no music is present, or to enable more advanced digitally controlled music effects.

In addition to the control circuitry described above, a 12v power supply was also built for the LEDs, which can theoretically draw a combined peak current of almost 10A according to the datasheet², but normally operate at a much lower current of around 2-3A.

2.3 Goals

1. Control each of the 5 LED strip's brightness and color with bands of frequency
2. Multiplex music brightness with digitally controlled PWM effects from an Arduino
3. Build a power supply capable of sustaining ~12V output at 0-10A
4. Design and manufacture a PCB to permanently install the project

² <https://cdn.sparkfun.com/datasheets/Components/LED/5060BRG4.pdf>

3. Design

3.1 Block Diagram

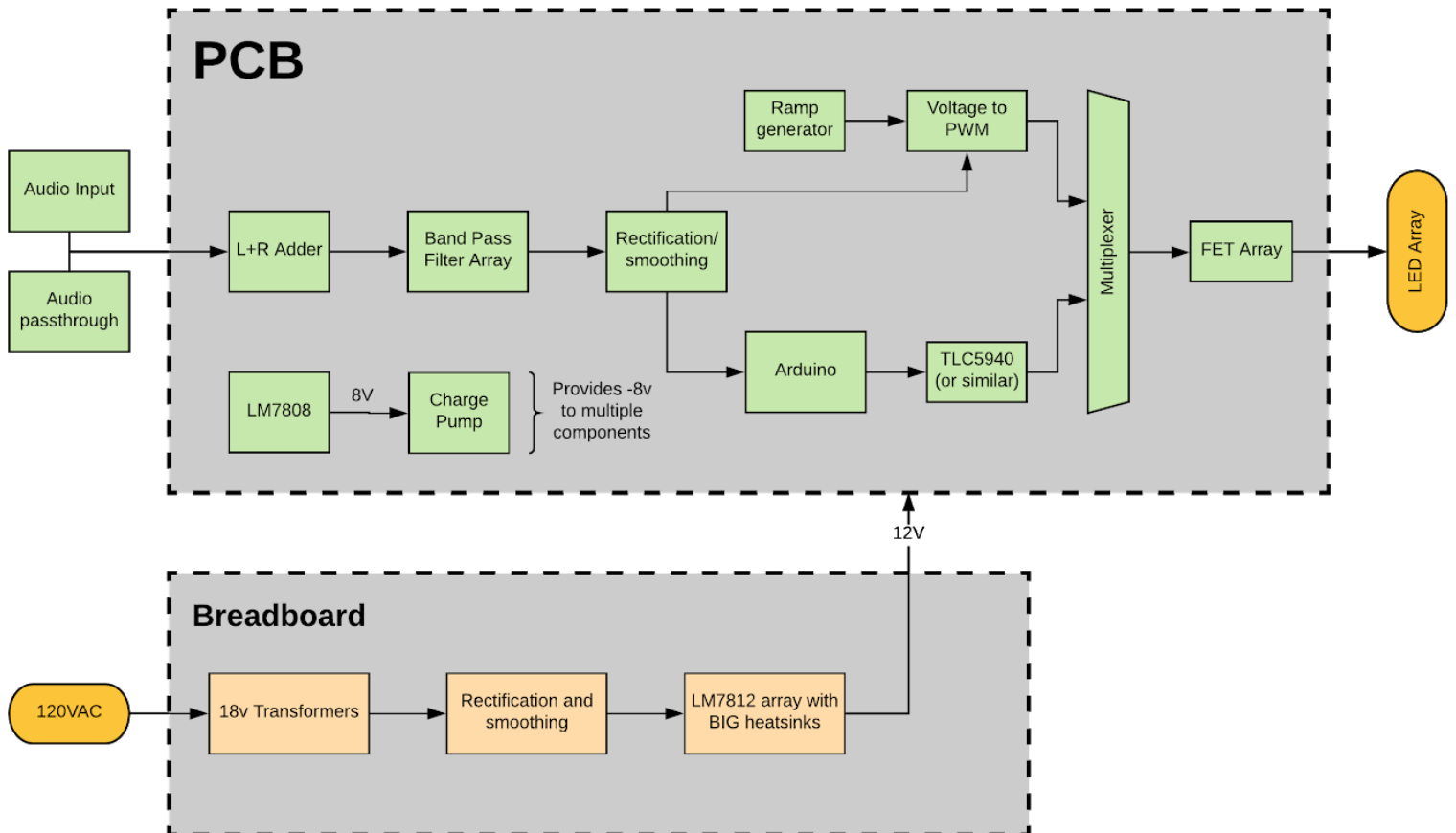


Fig. 2: Overall block diagram

3.2 Modules on PCB

In order from audio input to light output

3.2.1 L+R Adder

This module combines the left and right channels of audio into one, mono output. This is accomplished with an op-amp adder as shown below. The adder can also apply gain to the L+R signal by using a higher value of R3. To get a clearer output from the band-pass filters which it feeds, this was adjusted to a factor of $6.8 \cdot (V_{\text{Left}} + V_{\text{Right}})$, yielding a comfortable listening volume output in the range of $6v_{\text{pk-pk}}$.

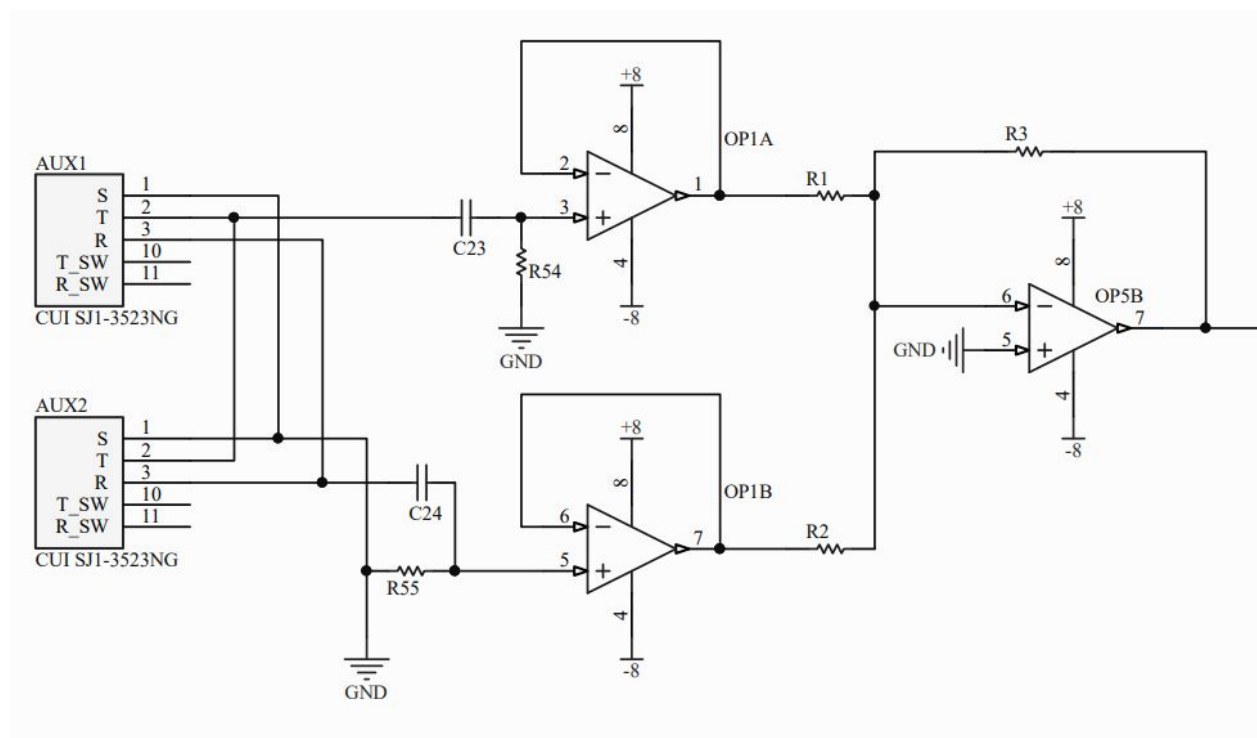


Fig. 3: Buffer and adder schematic for left and right channels

This module also contains the input coupling capacitors in case a connected source has DC offset. This behaves as a high-pass filter, so it was important to choose a high enough resistor value to bias the op-amp's input to ground. Here, a value of 360 k Ω and 0.22 μF gave a cutoff frequency of 2.0Hz, significantly lower than anything humans can hear in music.

$$F_{\text{cutoff}} = 1/(2\pi RC) = 1/(2\pi \cdot 360\text{k}\Omega \cdot .22\mu) = 2.0\text{Hz}$$

3.2.2 Band-Pass Filter Array

For audio visualization, a hard cutoff frequency on a band-pass filter isn't necessarily desirable. By allowing the filter to let some frequencies around where it was tuned through, a cool effect formed where a frequency sweep slowly passed amplitude from filter to filter, fading the corresponding channel. This allowed for a very simple design using only two capacitors and two resistors per band, cascading a high-pass filter and a low-pass filter. Using a second order filter such as the Sallen-Key topology would require two more passive components plus an op-amp for each channel while taking extra board space and providing a similar effect.

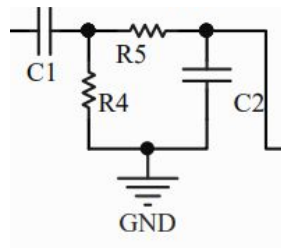


Fig. 4: Band-pass filter

For each filter, the high-pass cutoff frequency and low-pass cutoff frequency are shown in the table below. The cutoff frequencies were chosen based on the ranges of different parts in music.

Table I

| High-pass cutoff frequency $F_c = 1/(2\pi RC)$ | Low-pass cutoff frequency $F_c = 1/(2\pi RC)$ | R4 | R5 | C1 | C2 |
|---|--|-----------------|----------------|--------|------|
| 20 Hz | 100 Hz | 79 k Ω | 1.6 M Ω | 100 nF | 1 nF |
| 150 Hz | 300 Hz | 10.6 k Ω | 530 k Ω | 100 nF | 1 nF |
| 500 Hz | 1 kHz | 3.2 k Ω | 159 k Ω | 100 nF | 1 nF |
| 2 kHz | 5 kHz | 800 Ω | 32 k Ω | 100 nF | 1 nF |
| 13 kHz | 15 kHz | 1.2 k Ω | 8.2 k Ω | 10 nF | 1 nF |

3.2.3 Rectification and Smoothing

The AC signal from the band-pass filter array needs to be amplified by some amount, rectified, and smoothed in this module. The ideal amount of amplification to make the band roughly $6V_{\text{pk-pk}}$ isn't the same value for each filter. Therefore, potentiometers were used here. This also allows each band to be tuned in real time, preventing one loud range of frequencies in a song from being so bright that it makes the other bands hard to see.

The diode voltage drop has an unintended side effect of not letting noise through, as it never exceeds the $0.7V$ required to overcome the diode. When audio is playing, this drop is easily overcome by the amplified signal.

C11-C15 and R14-R18 were chosen based off of testing to look best at $33\mu F$ and $1k\Omega$. Increasing R14-R18 makes the effect of a loud beat last longer for a given frequency band, which can lead to the lights appearing always on in busy songs. The slower fadeouts can look really nice in less busy slow songs, but for most music this is not the case.

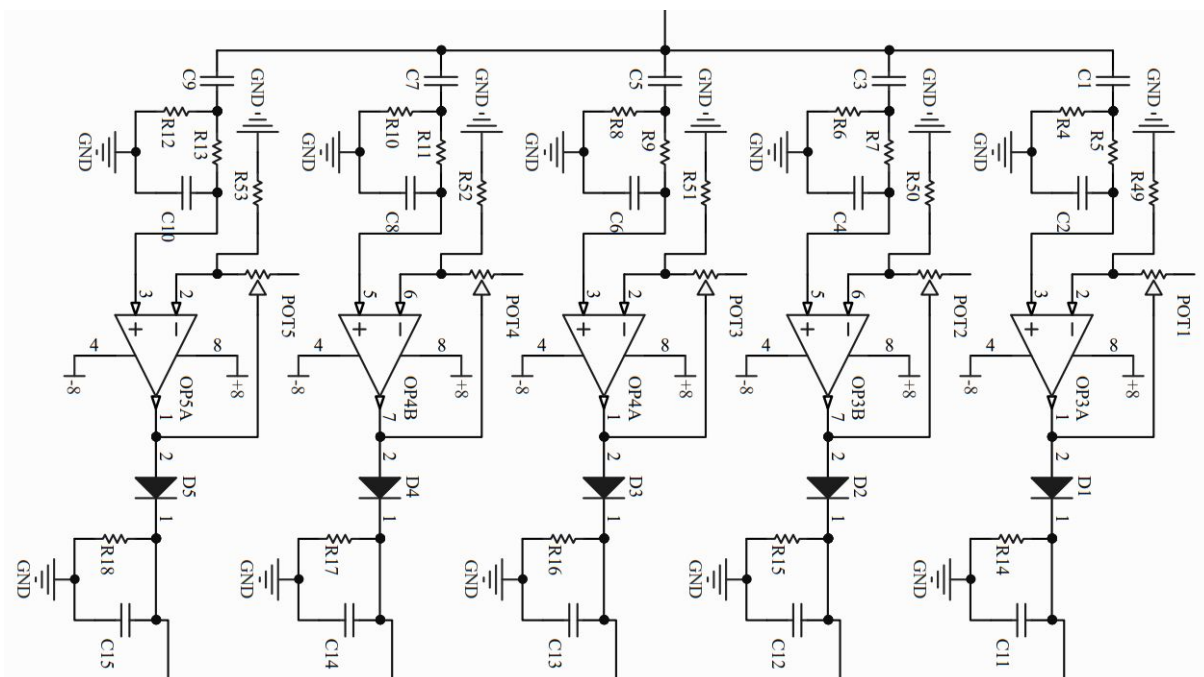


Fig. 5: Band-pass filters cascaded with rectification and smoothing schematic

3.2.4 Ramp Generator

To generate a PWM signal from an analog voltage, the voltage can be compared to a ramp to generate the necessary duty cycle. A JFET current source charges a capacitor connected to a 555 timer. The 555 timer triggers the capacitor to discharge to $V_{cc}/3$, generating a ramp. Since the rectified frequency bin can go down to 0V, it cannot directly be compared to this ramp. To fix this, the ramp has a constant value subtracted from it using an op-amp. It runs at 1.4 KHz. Adjusting R43 allows the ramp to be shifted up or down, increasing or decreasing the threshold at which the lights begin to respond to sound. On the final PCB, this was replaced with a 10K Ω rheostat to allow fine tuning of the threshold at which the lights turn on.

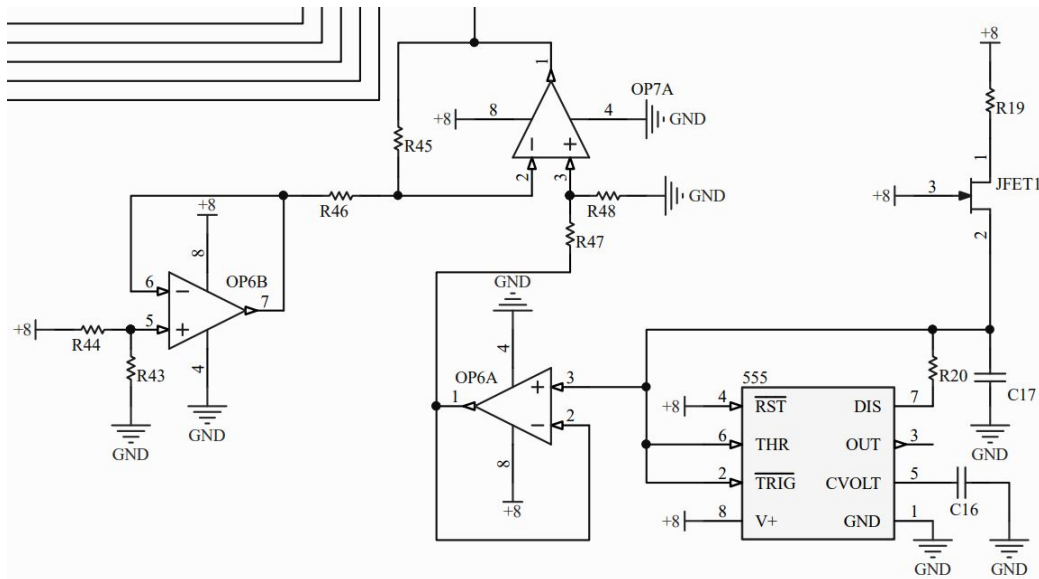


Fig. 6: Ramp generator schematic

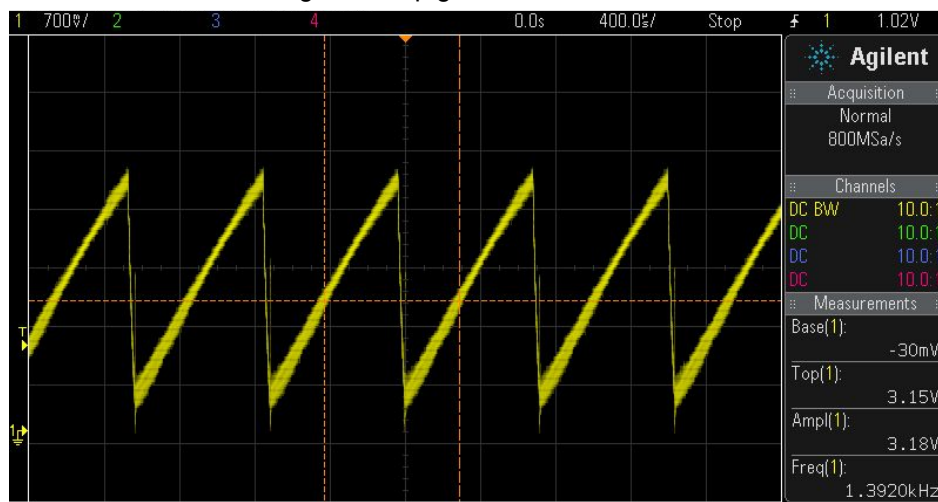


Fig. 7: Ramp output. Note that it starts just above ground with the current tuning of R43.

3.2.5 Voltage to PWM

This compares the ramp generator output to the rectified and smoothed band voltage using the LM311, creating a PWM signal. This is repeated once per band. LEDs are attached to the comparators in order to have visualization of the effect on board even without any output LED strips attached.

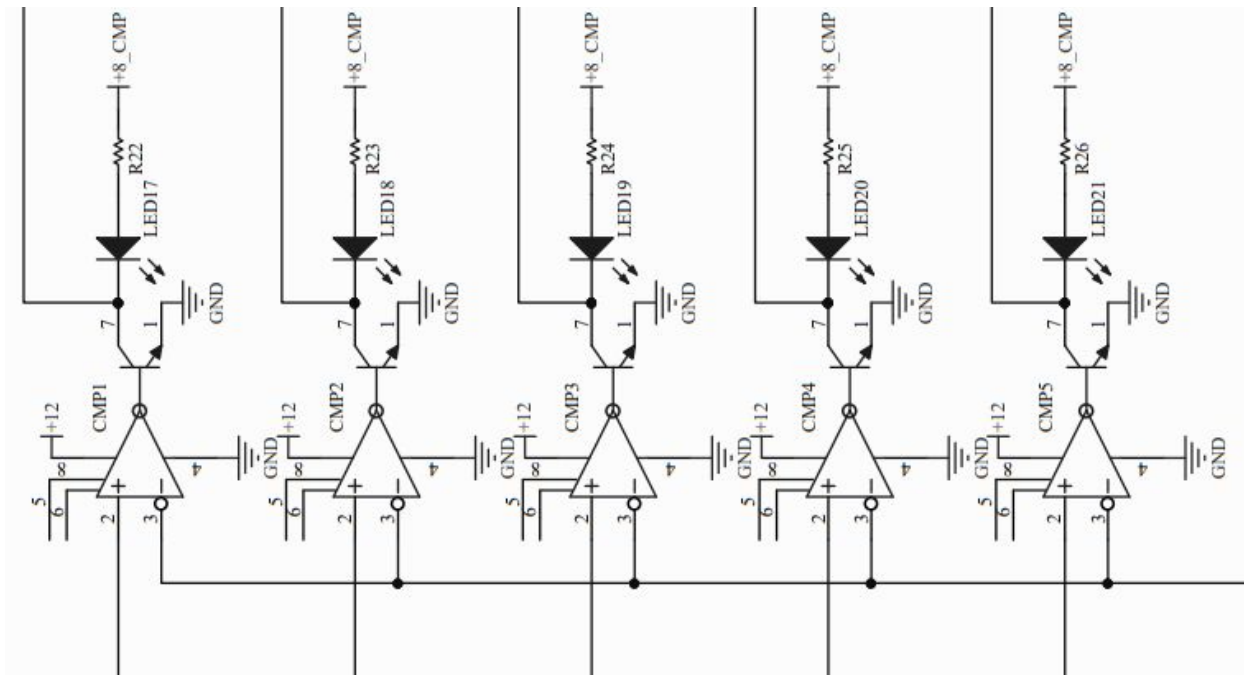


Fig. 8: Voltage to PWM schematic

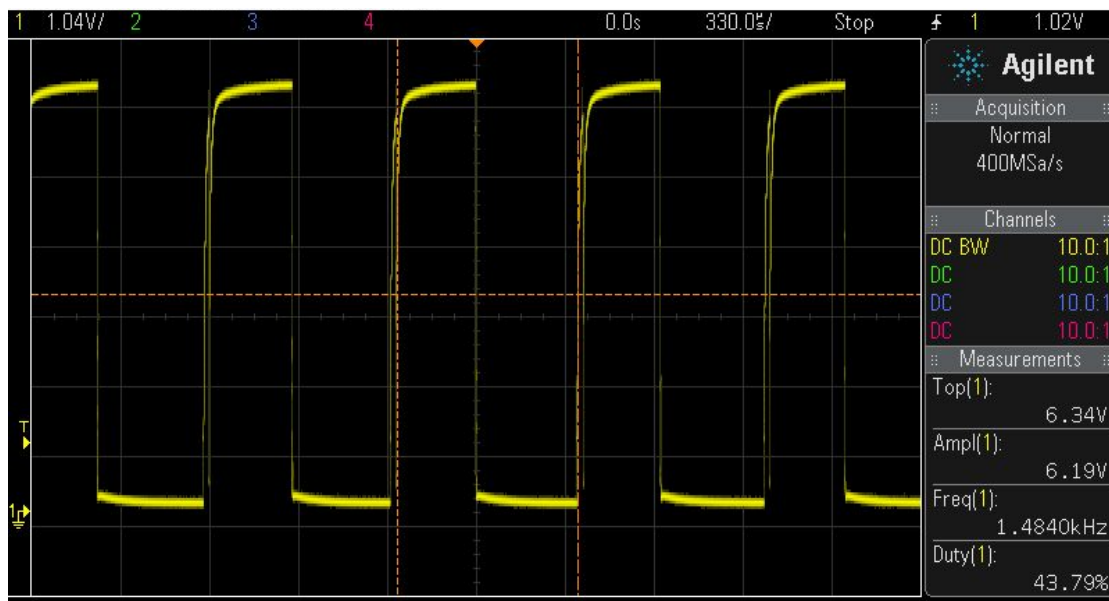


Fig. 9: Output PWM signal

3.2.6 TLC5940 and Arduino

This combination allows the Arduino to perform digital effects across 16 PWM outputs. The Arduino can perform processing using the rectified bands, or it can use its program memory to operate independent of the audio input. The Texas Instruments TLC5940 is a 16 channel PWM current sink designed to attach directly to current sensitive loads like LEDs without a resistor in series. By using a resistor, the voltage can be dropped down to zero when the channel is on since the programmed current will never be achieved from the supply voltage. Although the TLC5940 is capable of sinking a somewhat large amount of current without external MOSFETs, this is not being taken advantage of for a few reasons. First, multiple meter-long LED strips consume more current than the chip's channels can handle. Second, the analog side already requires the MOSFETs for its PWM output. PWM Shift registers that without an integrated driver exist and would be ideal for this application, but all lack the large amount of documentation and support surrounding the TLC5940.

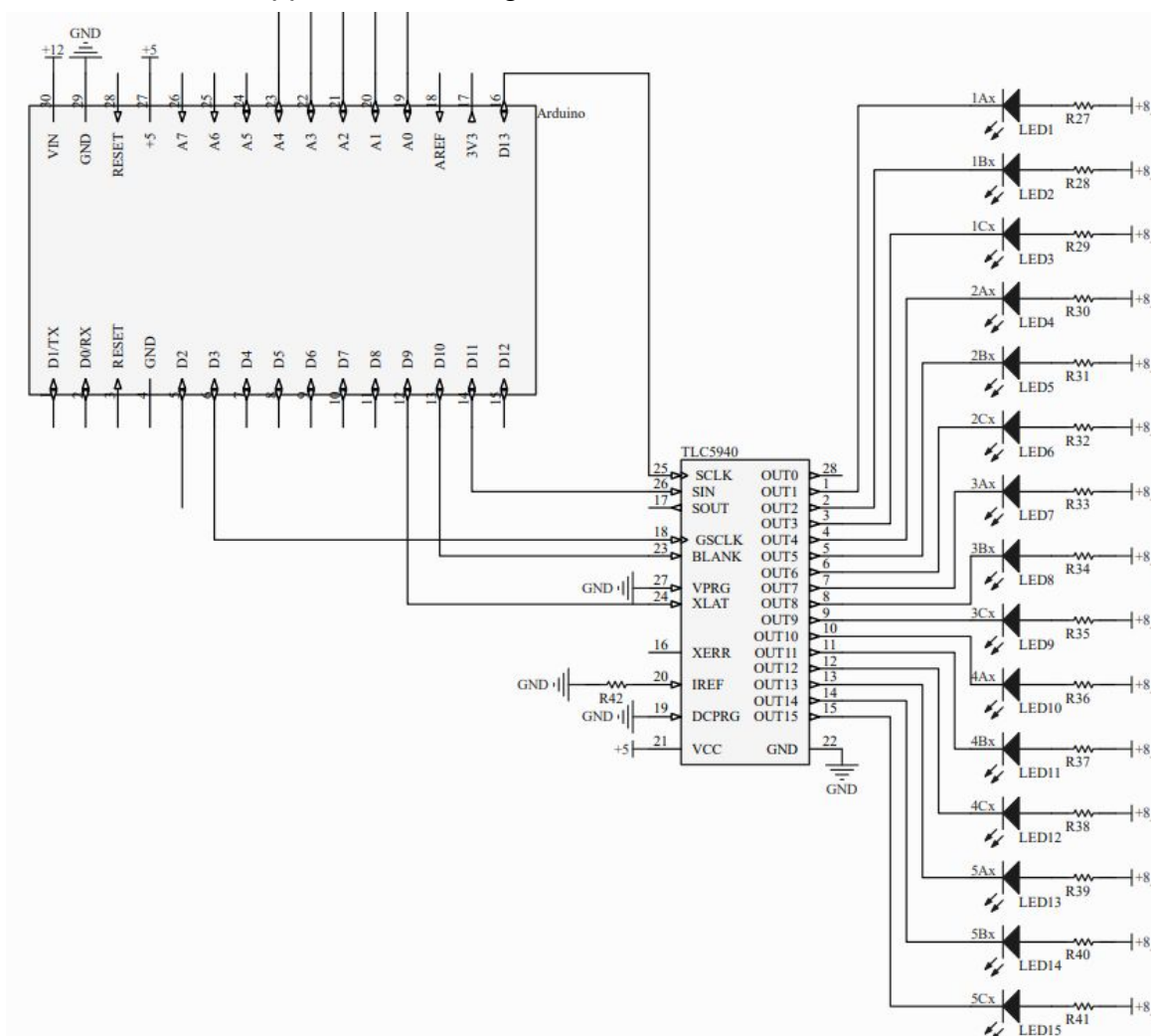


Fig. 10: Schematic for TLC5940 and Arduino. The wires 1Ax through 5Cx are fed to the multiplexer, shown here by label only to simplify the schematic

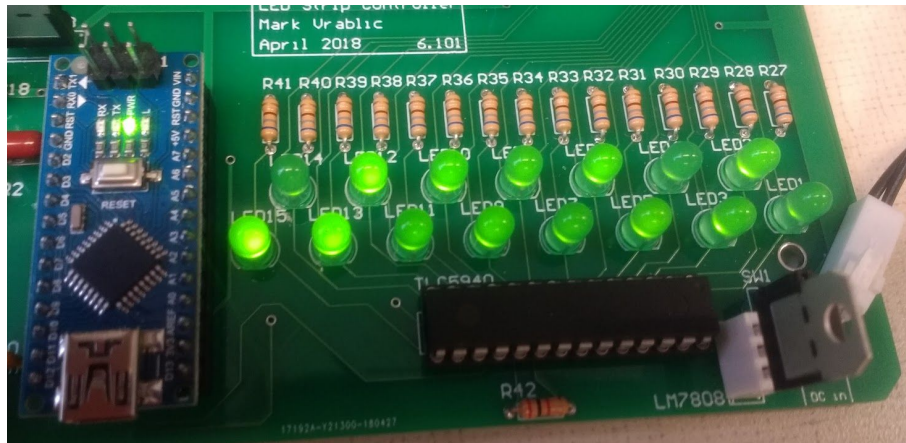


Fig. 11: Arduino and TLC5940 on PCB. Note output traces for 1Ax-5Cx on the top.

3.2.7 Multiplexer

This allows switching between the two 15 channel inputs, outputting one 15 channel PWM signal. Ideally it would simply be a 15PDT switch, but that doesn't exist. Luckily there are ICs which perform this task, but they are only available in SMD. In order to verify the functionality of every component on the PCB before ordering it, a DIP solution was needed. The IC ultimately chosen was Ti's CD4053BE, a two channel multiplexer with three multiplexers per package. This meant that five ICs were needed to switch the 15 channels, taking up lots of valuable board space (see fig. 12).

Since only five frequency bands are fed to the multiplexers, each band is fed to three channels of LED strip to make up the 15 total channels. Rather than simply using one band per three channel LED strip which would produce 5 independently controlled strips of white light, the wires were mixed between the strips in a pattern such that lower frequencies were generally mapped to the left strips, mid frequencies to the middle, and high frequencies to the right. This gives the effect of separate parts corresponding to different LED strips while creating a variety of colors.

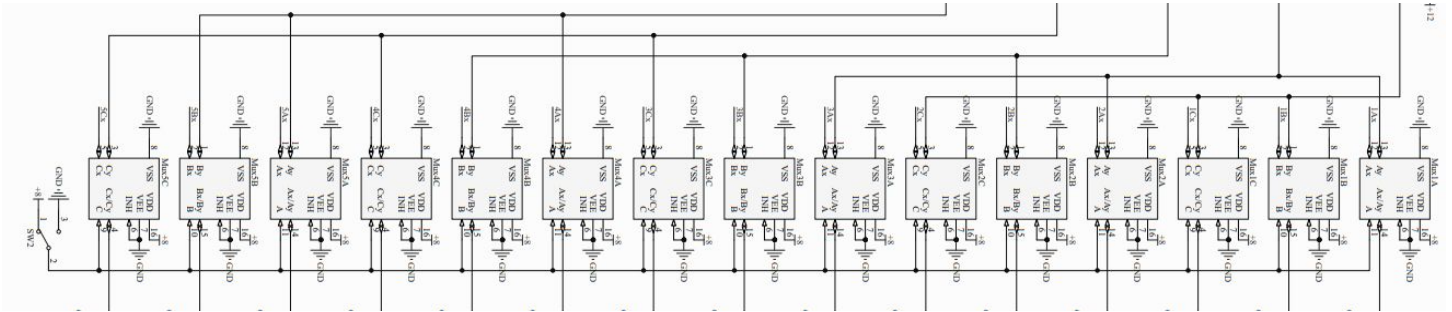


Fig. 12: Multiplexer schematic. PWM from the 5 bands enters on the top, the TLC5940 output connects to the labelled wires on each of the 15 multiplexers.

3.2.8 FET Array

In order to accommodate varying current draw for different lengths of LED strip, N-channel MOSFETs were used for switching. The surface mounted Si2302ds is cheap and fits the requirements perfectly. One important consideration when laying out the board in this region was the width of the traces carrying current from the DC input.

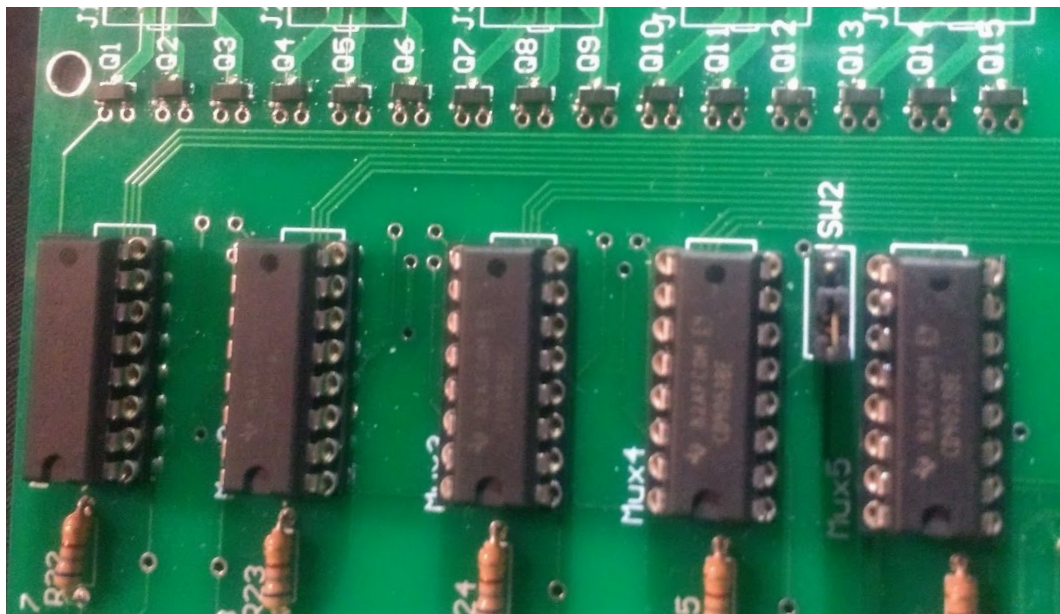


Fig. 13: Multiplexers and FET array

3.2.9 Charge Pump

To supply -8V to the many components which require one without an uncommon power supply with both positive and negative voltages, a charge pump was used. To maintain a reliable negative voltage, an IC was ordered to perform this function. The Intersil ICL7660A charge pump was chosen for its ability to provide the necessary 20mA current and small DIP footprint. 10 μ F capacitors were used as suggested in the common configuration schematic on page six of the datasheet³. To mitigate the noise the pump creates on the positive rail, a 330nF capacitor was placed on the positive rail near the IC.

³ <https://www.intersil.com/content/dam/Intersil/documents/icl7/icl7660.pdf>

3.3 Modules on breadboard

3.3.1 Rectification and Smoothing

Converts the 18VAC into a DC voltage from which the linear regulators are attached. In order to stay within the voltage limit of the 3300 μ F electrolytic capacitors used and minimize the heat produced by the LM7812 array, only a half wave rectifier was built.

18V RMS AC with a half wave rectifier will output a peak voltage of $18V * \sqrt{2} - V_{diode} = 25.5V - 0.7 = 24.8V$, allowing the use of 25V rated capacitors. LM7812s are not rated below 14.5V input⁴, so to calculate the capacitance required to sustain the 10A until the next period 1/60 seconds later, a $14.5V/10A = 1.4\Omega$ resistor is used to calculate a value which must be below time constant's actual value (This makes the assumption that at 24V, 17A of current is drawn). Then, using $14.5V = 24.8V * (e^{-(0.0167/(1.4 * C))})$, an overestimate of C is 22000 μ F. Therefore, three 3300 μ F capacitors, totaling 9900 μ F was tested and proved to be sufficient to prevent dropout.

3.3.2 Cooling the LM7812 Array

Linear regulators allow a consistent voltage with highly varying load current, but generate a lot of heat when the current being drawn increases. With the LED strips in use today, current can theoretically swing from 0 to 10A; too large for a buck converter and likely to require a harder to build switching power supply if efficiency is a concern. Since any place the final board is installed will have a proper switching 12VDC power supply, this power supply was built using linear regulators. They are mounted on to a server heatsink using thermal grease on the back and glue around the sides, which works surprisingly well to keep them cool. Two 12V server fans cool the heatsink and are also powered off of the linear regulators. These heatsinks typically can dissipate 100W of heat while remaining a reasonable temperature (depending on efficiency of heat transfer), so the 5 regulators running at peak of 2A each will dissipate at peak $(24.8V - 12V) * 10 = 128W$ of heat. This is at peak and an overestimate. Typical power consumption is much less, and the heatsink doesn't even get warm. Additionally, the LM7812s enter thermal shutdown when they overheat so even if peak is sustained,

⁴ <https://www.jameco.com/Jameco/Products/ProdDS/889305.pdf>

the worst that can happen is one of the strips shutting off until its corresponding regulator cools back down.

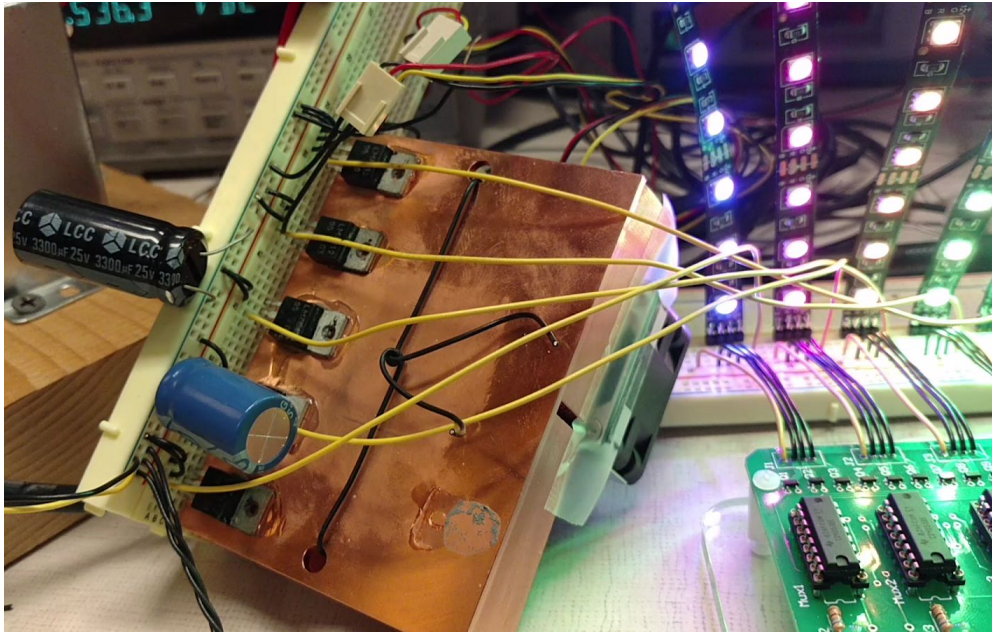


Fig. 14: Power supply with heatsink and LM7812s running (very loud)

3.4 PCB Overall Schematic

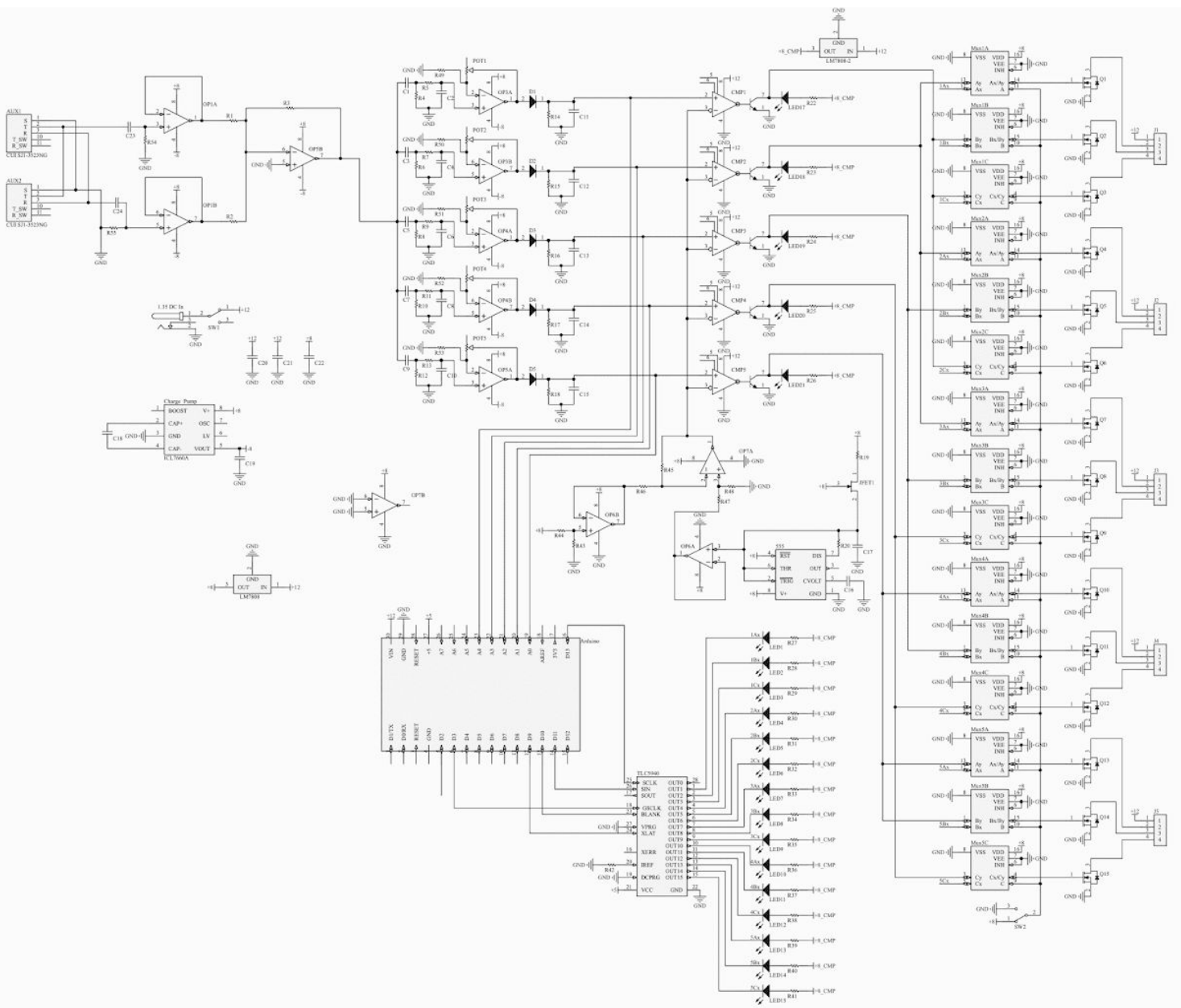


Fig. 15: Overall Schematic. For more details visit section 3.3

3.5 PCB Layout

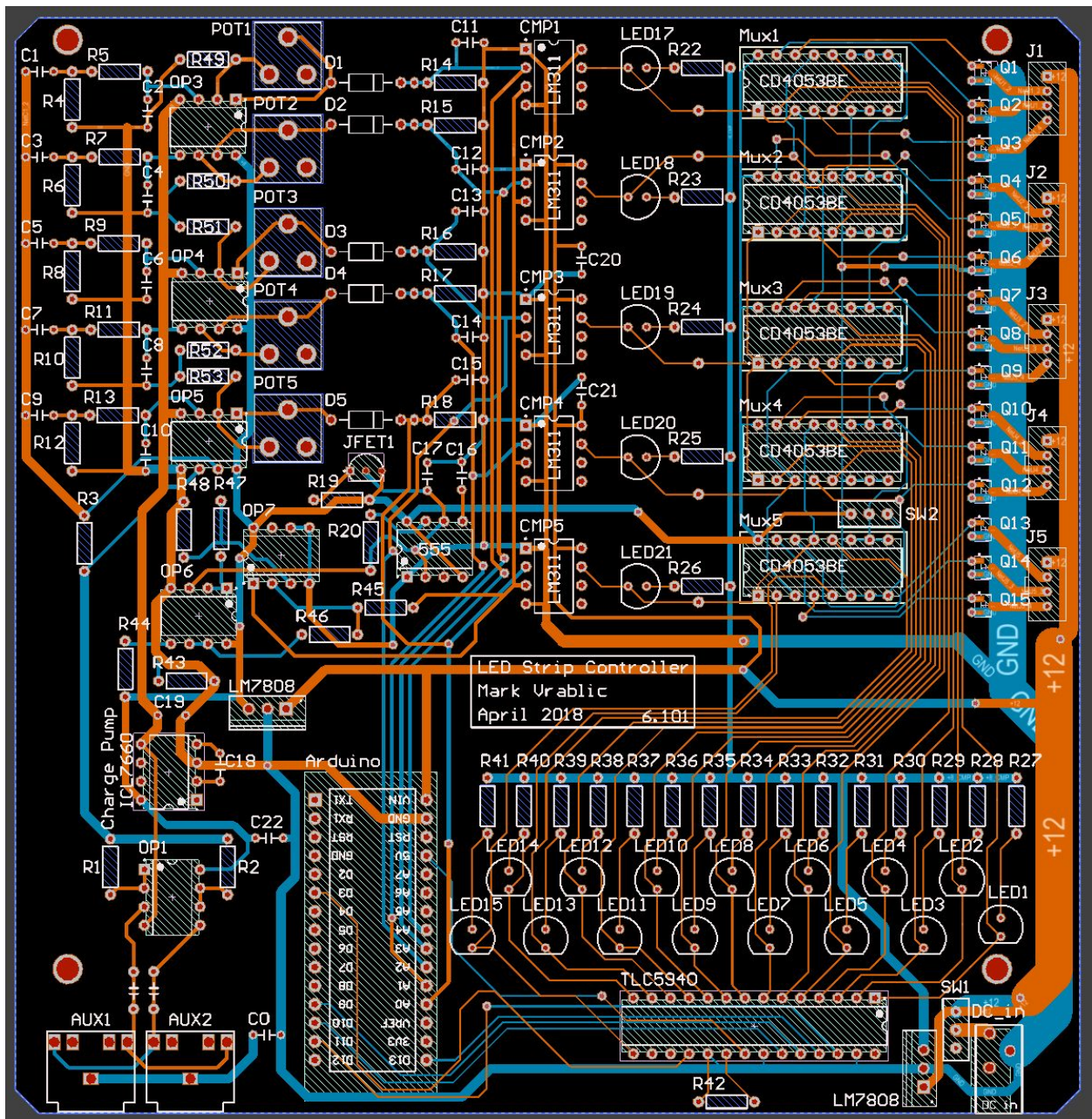
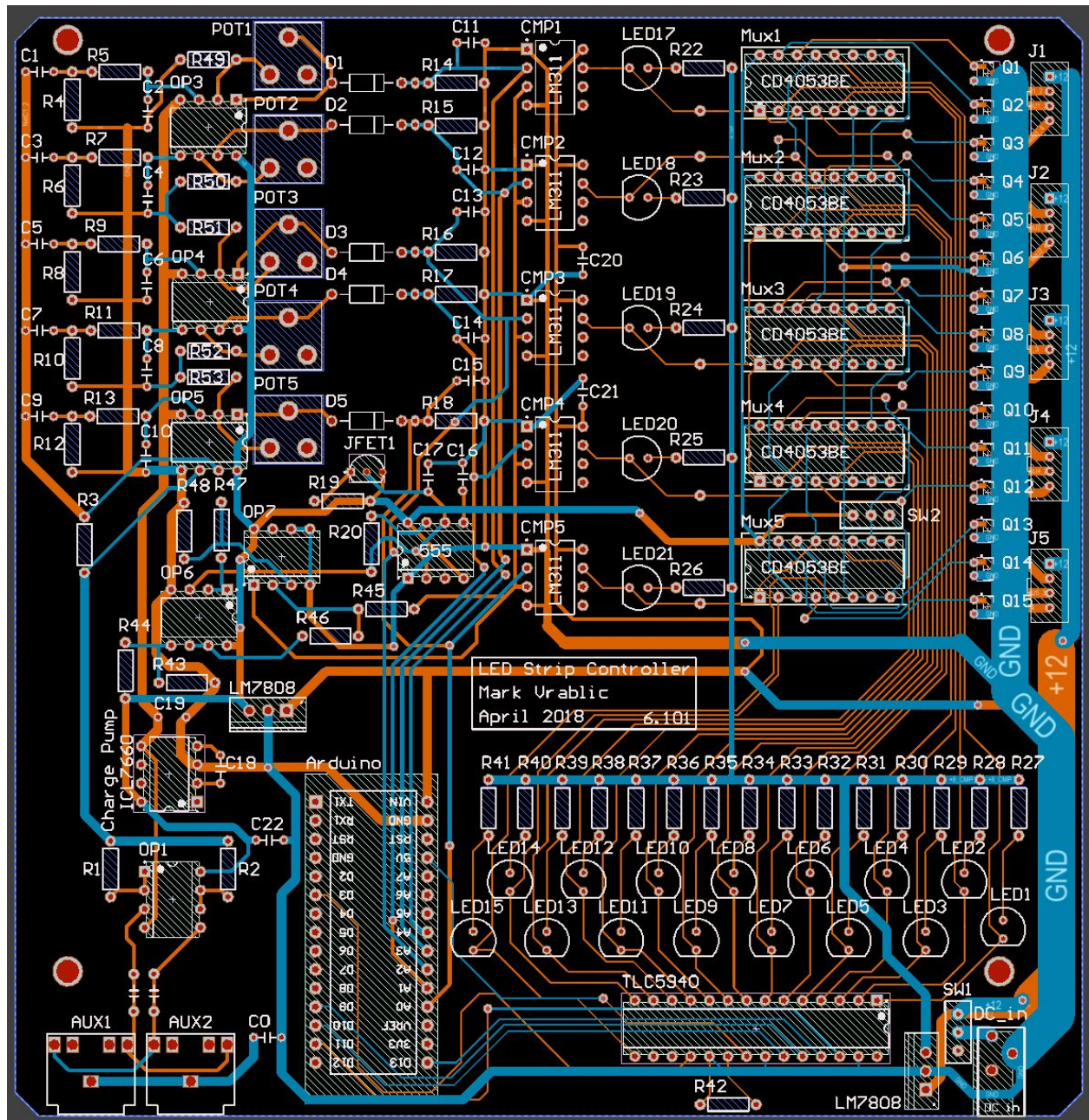


Fig. 16: PCB top layer



4. Implementation

4.1 Challenges

In order to verify the design before ordering the PCB, two channels were built on breadboards; one for the highest frequency band and one for the lowest. By doing this, all bugs were worked out prior to ordering the PCB.

Many of the issues encountered in this process had to do with current spikes on power rails. The CD4053BE multiplexers are very sensitive to supply voltage ripple, and were attached to the same 8V rail as the charge pump. This was solved by using a 10 μ F capacitor in parallel with a .1 μ F capacitor next to the charge pump. Additionally, the 8v rail had a somewhat high current demand with the 20 indicator LEDs on the board, charge pump, and op-amps all using it, so ultimately another LM7808 was used to distribute the load. With the entire load on one regulator there would be 300mA peak current, ~200mA in normal operation; $P_{\text{dissipated}} = V \cdot I = .3A \cdot (12-8)V = 1.2W$, enough to make a single LM7808 run hot without a heatsink.

In breadboarding the band-pass filters, the highest frequency band worked great, but the lowest was not activating. After trying to debug it using a tone generator program and oscilloscope, it was discovered that the circuit itself was functioning fine, but the laptop's headphone output amplitude dropped off quickly at frequencies below 70Hz. This was not a problem when using the function generator or a phone.

In order to install the organ permanently in multiple locations, designing a PCB was highly desirable for this project. This brought along its own set of constraints. Long manufacture shipping time due to International Labor Day meant that the order needed to be placed only a few days after the final components to verify operation on a breadboard arrived. Additionally, the copper trace thickness imposed a limit on the current which the strips can draw, elaborated below.

With ideal wires and MOSFETS, the LED strips can consume any amount of current with this design. In reality, there are a couple of limitations. First, the MOSFETS used only support 3A per channel, leading to a limit of 9A per strip. This isn't a big issue though, as $P_d = 12V \cdot 9A \cdot 3 \text{ channels} = 324 \text{ watts}$, with 5 strips that's over 1.5 kW of LEDs and impractical to power from a single 12V power supply. The second and bigger limitation is with trace width on the PCB, which was made as big as possible to accommodate larger currents, but with a 1oz/ft² copper thickness on the PCB current is theoretically limited to 8A combined between all channels for a 35°C temperature rise

over ambient based on trace width calculators online⁵. This could be increased by soldering extra wires to the pads to increase capacity or ordering a thicker board. To date the largest combined load tested is 5A, and with it the traces don't get warm at all, likely having plenty of headroom for more current than the math would suggest based off of generic IPC-2221 PCB curves⁶.

When the board arrived and was assembled, the lights wouldn't turn all the way off, but would get very close. This was caused by the comparators receiving 0v from both the rectified band output and the bottom edge of the ramp. Luckily, this was easy to fix by altering R43 to decrease the voltage being subtracted from the 555's ramp output as described in 3.2.4.

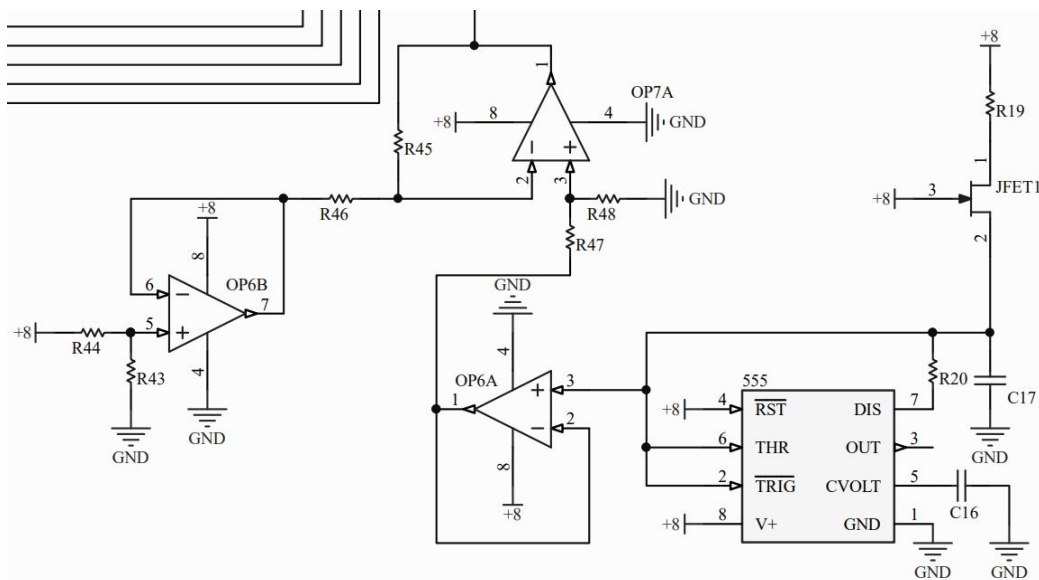


Fig. 18: Ramp generator schematic

The final challenge encountered when installing this in its current location was with the connector footprint chosen. It was based off of a 5050 RGB LED strip, and is smaller than the standard .1" spacing on breadboards. Unfortunately, in the final installation location, a different brand of LED strips were used with a .1" header. This was fixed by soldering 0.1" female headers to the 5 strip outputs.

⁵ www.4pcb.com/trace-width-calculator.html

⁶ <http://www.ipc.org/TOC/IPC-2221.pdf>

4.2 Further work

There is space on the board to add .1" headers in addition to the current smaller ones. If the 3 boards left all get used, they will be added before reordering. Additionally, the CD4053BE multiplexers are expensive considering that 5 are needed for the board. There exist single chips which should be able to replace them but they are only available in SMD (Texas Instruments SN74CBT162292 for example)

The effect looks nice now, but narrower frequency bands may look nicer for some songs, specifically the 150Hz though 300Hz band. Luckily, this doesn't require any board modification, only replacement of the resistors and capacitors in the band-pass filters.

Finally, more complex software needs to be made for the Arduino, as it currently only runs an ambient color scroll effect when no music is playing rather than a mix of music correlated effects on its own.

5. Conclusion

The project functions as expected, producing a satisfying light output synchronized to music. It is currently installed in an East Campus bathroom, where it is connected to the hall-wide music and speaker system. It runs 24 hours a day and was used during the most recent party on hall without issue.