Electrify Atwater Kent, Final MQP Report

Juan Caraballo, Shannon McCormack, Jeffrey Wagner

April 24, 2019

0.1 Abstract

The Electrify Atwater Kent project aims to breathe new life into the dull, dreary building of Atwater Kent. LED stairs, bringing a modern look to the front stairwell, as well as the RasPiano, with customizable filters and visualizations to demonstrate the discipline of signal processing, will be installed in the building. When the project is complete, Atwater Kent will be not just a welcoming space that students feel comfortable studying in, but an inspiration for future engineers as well.

0.2 Acknowledgments

We would like to offer a special thanks to the individuals listed below for helping to make our MQP the greatest possible learning experience we could have achieved. Each person offered us their expertise and insight into numerous problem solutions and methods and truly helped drive our projects to a well designed state.

- Prof. John McNeill advised this project, meeting with us weekly and helping guide our success.
- Prof. Stephen Bitar aided us through technical problems.
- Bill Appleyard lent us equipment and taught us about cabling, acrylic gluing, and more.
- Mitra V. Anand assisted in fabricating prototype PCBs and laser cutting acrylic.
- Shadi Tasdighi Kalat attended two consultations regarding the 3D model of our Light Bars. Gave advice on how to strengthen the structure of the bars as well as some design tips and material recommendations.
- Devdip Sen and Ian Costanzo shared a lab space with our team and frequently offered their expertise.
- Analog Devices sponsored this project.
- Allegro Microsystems sponsored this project.

0.3 Executive Summary

In this executive summary we will cover the background of our MQP project which discusses the MQP's initial goals and what our team hoped to achieve, provide an overview of the two systems/devices we intended to develop, the design and implementation of each, and conclude with a discussion of what we were able to accomplish.

0.3.1 Background

When we were first introduced to the MQP it was titled "Pumpkin Lounge Power Panel" after the project's initial target project shown below.

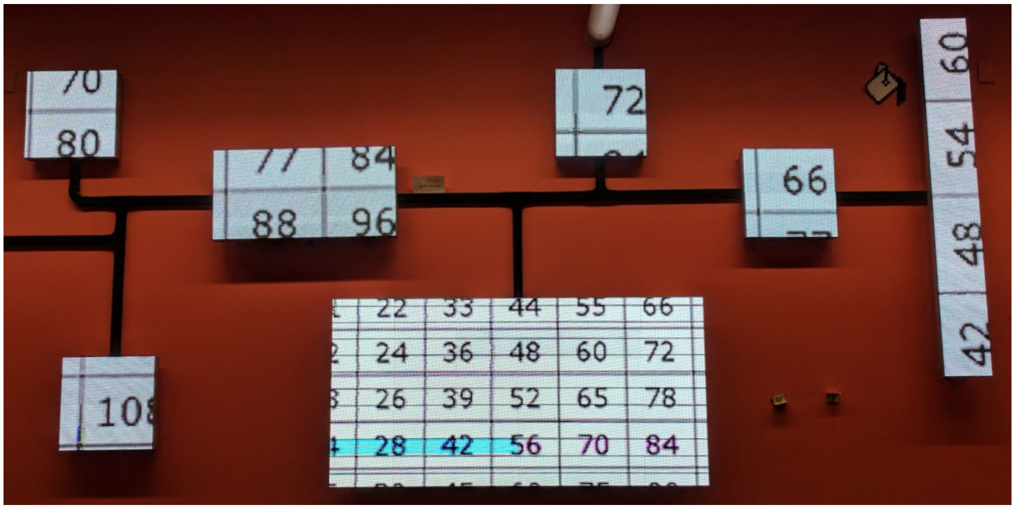


Figure 1: Pumpkin Lounge Power Panel with Test Image

The goal of the MQP was to initially repair and add features to this led matrix display as it was not fully operational at the time. As shown below in figure 2, the system was plagued with graphical bugs and would often crash or freeze within 30 minutes to 2 hours of being active.

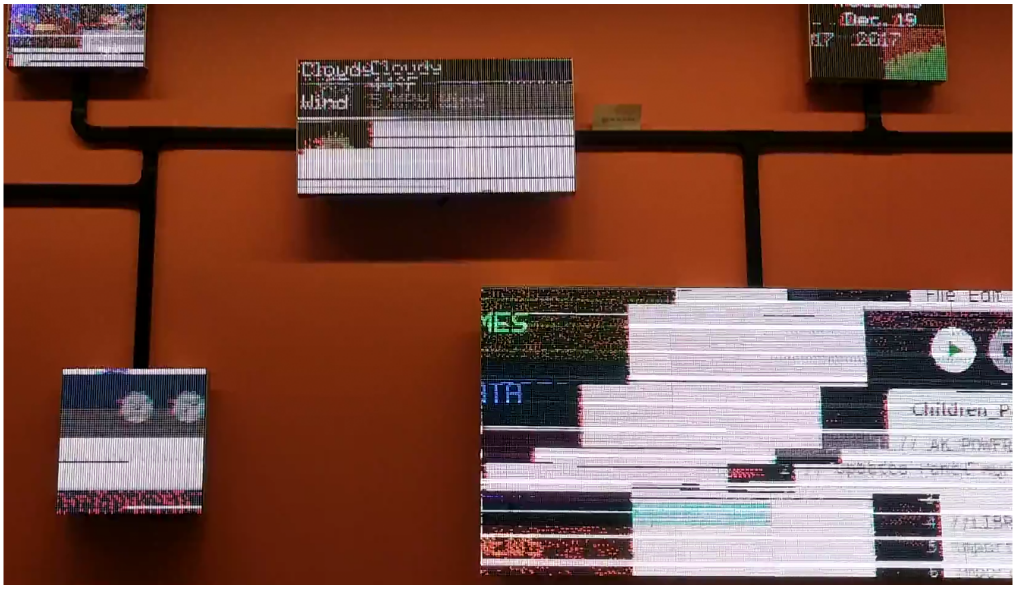


Figure 2: Pumpkin Lounge Power Panel Glitching

As our team was interested in adding only features to the display we took on the task of repairing the system the winter before our MQP started. This process involved debugging both the hardware and software of the system which yielded the following sources of problems:

- Host graphics processing unit
- Software bugs
- Insufficient power supply

Starting with the host graphics processing unit we found that micro-computer devices such as a Raspberry Pi and Odroid XU4 (used in the original design) were not capable of properly outputting video to the displays. It is still not clear why that is however we did find that full Windows machines were fully capable of driving the displays without any graphical issues.

Multiple bugs were found within the systems software which caused it to close its application or halt operation whenever data could not be retrieved from the internet. These bugs were bypassed by having the system continue with old data and ignore broken media streams.

The final bug found with the system is due to how it is powered. Currently the power panels share the same power breaker as all connected devices within AK113. When all lab computers are operating, powering and running of the Power Panels trips the rooms breaker indicating that too much current is being drawn from the single breaker. The resolution for this is to simply add a new breaker for AK113 and give the Power Panel exclusive access to it.

After resolving the software issues and driving the system with a desktop computer our team was able to bring the Power Panels to a stable and operational state.

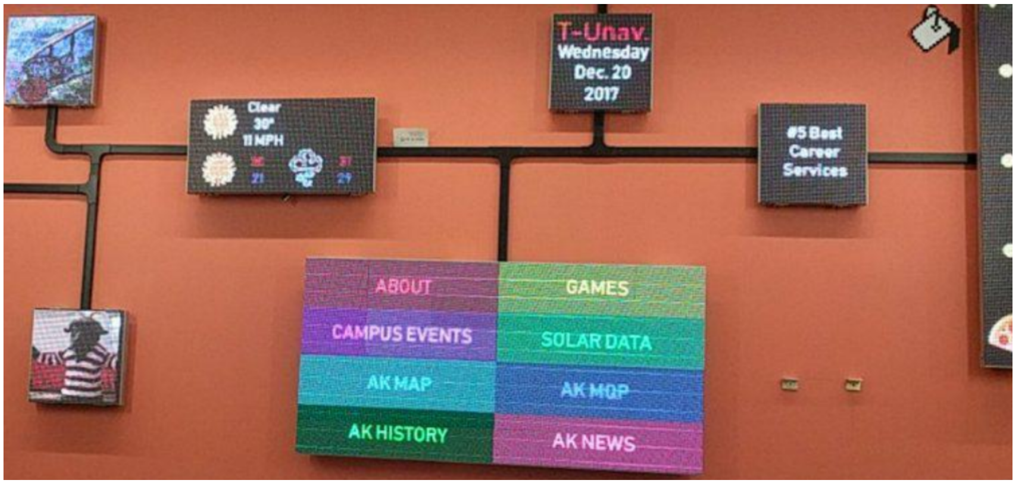


Figure 3: Operational Pumpkin Lounge Power Panels

After having brought the Power Panel's to a functional state, our team was interested in creating a new device to add more activity to Atwater Kent. Additionally it became clear that outside of software there was not enough electrical design work to be on the Power Panels to take them on as a full MQP project.

To begin our ideation we performed a survey which sought to gather as many project ideas as possible geared toward improving Atwater Kent and the persona it maintains. This survey asked the following three questions:

- If you had an infinite budget, what would your dream tech building include?
- What does Atwater Kent have that doesn't work?
- What is Atwater Kent missing?

As you may notice the scope of each questions takes a drastic shift from unlimited resources that could enable the wildest of ideas to identifying specific issues within Atwater Kent. Specifically the first questions asks what an individual's dream building would include assuming they had an infinite budget. This questions seeks to invoke a surveyor to consider the things that would make them happiest in a place of work even if unrealistic. We then ask what Atwater Kent has that doesn't work to shift the surveyor's state of mind from generating creative ideas to identifying specific things they wish could be improved. Finally we ask what the building is missing. This question is explicitly placed last as it intends to take the surveyor's response from the first question, consider what they believed could be improved about the building in the second question, and finally determine what may be more realistically achievable that may still satisfy their desires for their "dream tech building."

With over 100 responses the results were overwhelmingly positive. Numerous surveyors identified similar critical aspects of Atwater Kent that they believed needed to be improved. A summarized version of the frequent responses are below:

1. Atwater Kent needs more light - it is too dimly lit
2. The building feels dated and uninspiring
3. The building is too inactive

Using this feedback concerning the student's impressions of Atwater Kent and considering their responses for devices that could be provided within the building such as a Piano, we generated many project ideas for what could be done for Atwater Kent. After performing value analyses on our many ideas and meeting with our advisor for his input as the ECE department head at the time we decided to take simultaneously take on two projects under the name "Electrify Atwater Kent." a LED Stair System and RasPiano.

0.3.2 Project Overview

With the two projects decided on for our MQP we will now perform an overview of each. First we will cover the main idea for each and how they apply to our technical and project requirements. Then we will discuss the block diagrams for each and describe how the different functional blocks contribute in constructing their respective systems. Note that we reserve discussion concerning the specific hardware and software implementations of each block within the block diagrams below for parts 2 and 3 of the report.

0.3.2.1 RasPiano

The RasPiano is intended to serve as both an educational tool and interactive display. The primary goal is to demonstrate to visitors and prospective students the kinds of devices that can be created with an education in electrical and computer engineering from the Worcester Polytechnic Institute (WPI). Shown below is a 3D model and diagram of how the finished device is intended to appear.

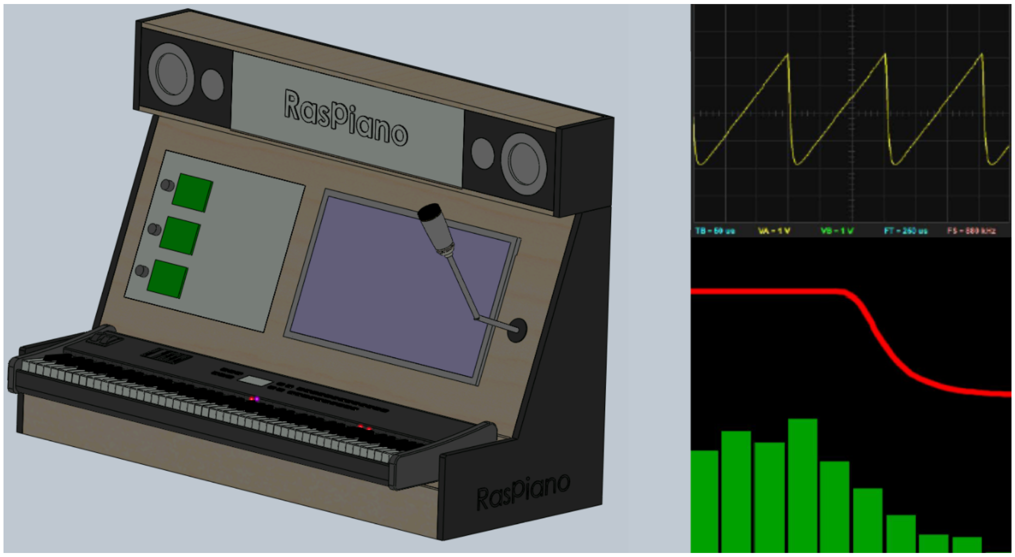


Figure 4: RasPiano 3D Model and Display Artist Rendition

Inspired by the housing of arcade cabinets, the RasPiano would feature a MIDI keyboard, microphone, and auxiliary port as the primary device inputs. On the left of the housing face would be a series of diagrams and descriptions of the different filters available to control on the device. On the main display is a split view of an oscilloscope and spectrum analyser with transfer function display. The purpose of the display is to demonstrate how the audio waveforms in the system appear and the effect of the built-in filter chain on them. On these two screens would be a control description detailing how the peripheral keys on the MIDI keyboard control the device.

This project is valuable to us as it would meet our project goals of adding to making AK a more welcoming environment that promotes innovation. Additionally it would allow us to practice signal processing, embedded development, and with the use of a analog filter chain, design and implement analog circuitry. Thus meeting the majority of the technical skills our team desired to sharpen.

Below is the block diagram for the RasPiano system:

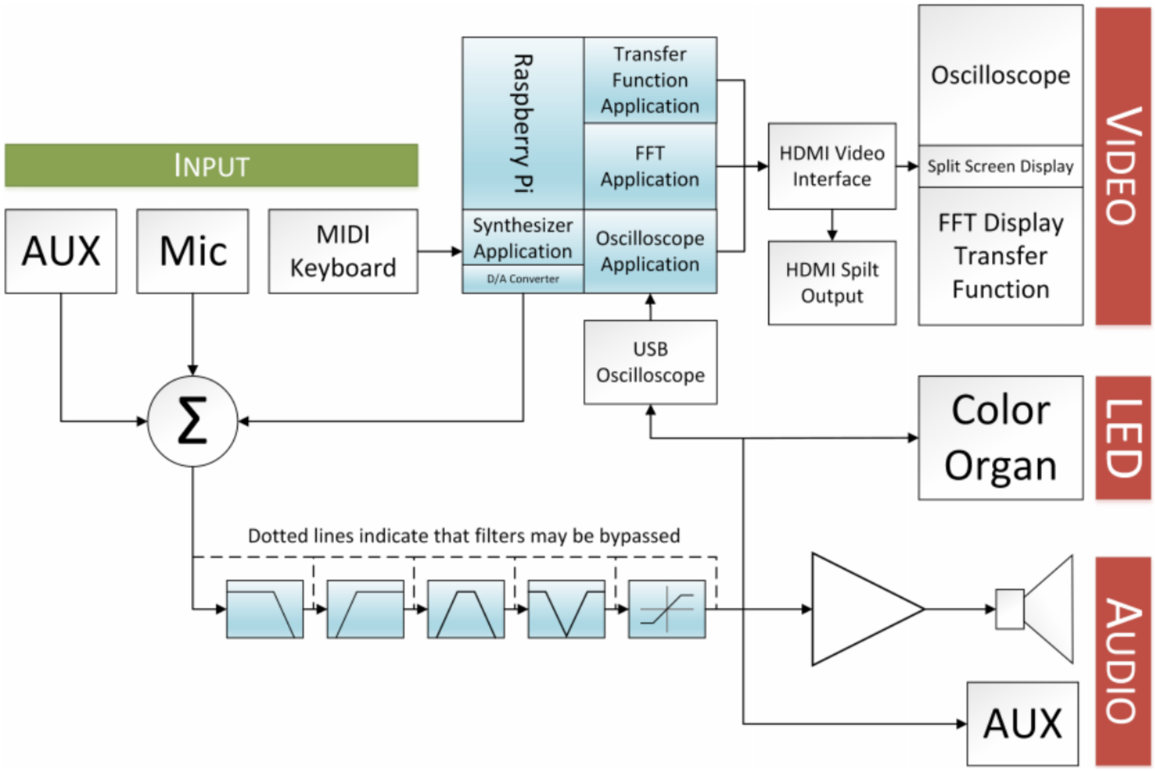


Figure 5: RasPiano Block Diagram

Starting from the device inputs there are three means of providing audio to the system. These are the auxiliary input for mobile media players and smartphones to stream music to. A microphone for users to speak or sing into. And finally a MIDI keyboard which controls a synthesizer that outputs sample audio from an analog auxiliary output on the Raspberry Pi. These three signals are summed and passed through a filter chain that consists of a lowpass, highpass, bandpass, bandstop, and clipping distortion filter.

After the filter chain, the modified audio signals pass to multiple devices and outputs. First it passes to a auxiliary headphone output and a amplified speaker to provide an audio output. In addition this signal is passed to a discrete color organ which analyses the audio and illuminates several bands of leds corresponding to the active frequencies on the audio spectrum. Finally the audio signal also passes to a USB oscilloscope. This oscilloscope reads the analog waveform, digitizes it, and reports the data to a oscilloscope and Fast-Fourier Transform (FFT) application. These applications then form respective diagrams to connect to a monitor. On the top half of the monitor is the oscilloscope display which traces the audio waveform, and at the bottom is a spectrum analyser. Within this spectrum analyser is also a trace of the filter chain's transfer function. Please refer to figure 4 for an artist rendition of this display.

0.3.2.2 LED Stairs

The LED Stair project aims to meet the project goals of further providing a welcoming environment to Atwater Kent and promoting modern technology and innovation whilst inspiring creativity. As a system that is more geared towards aesthetics than education the LED Stair project is a system intended to be installed on the front left staircase of Atwater Kent as shown below.

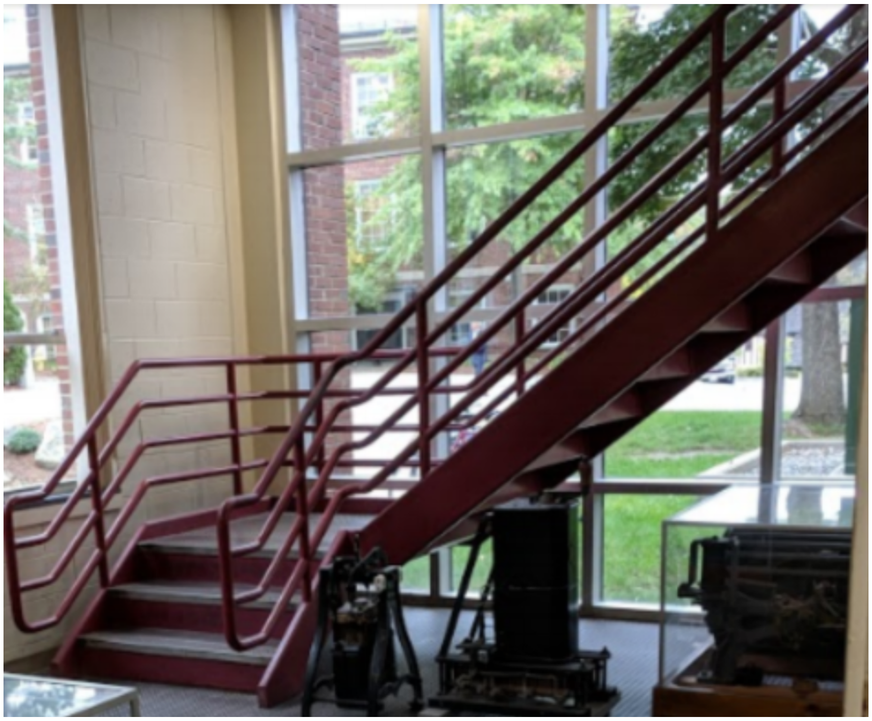


Figure 6: Atwater Kent Front Staircase

This location is perfect for installation of the system as it provides open visibility from both the outside and inside of the building. As demonstrated in figure 6 below light bars would be mounted vertically on each step and designed with the LED pattern.

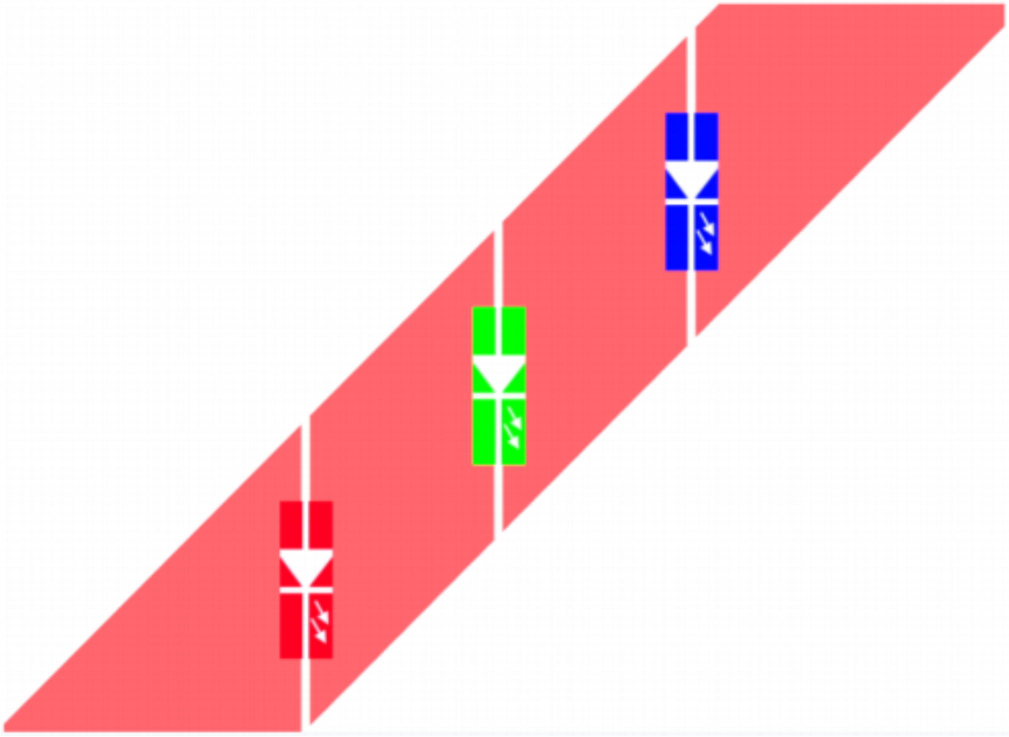


Figure 7: Concept Sketch for Vertical Light Bars

When the steps are not occupied colored animations will play across the entire staircase. As soon as a person steps on a step, animations will disable (turn off) on all steps and only the currently occupied steps will stay illuminated in white.

This project also presented a strong value to our team as it not only met the customer requirements associated with the project, but it provided an opportunity to practice much real-time embedded development and sensor design. With the need to manufacture many light bars for the system's 21 steps it also presented an opportunity to practice additional engineering skills such as CAD 3D modeling, acrylic cutting and 3D printing, and system power routing.

Below is the block diagram for the LED Stair system:

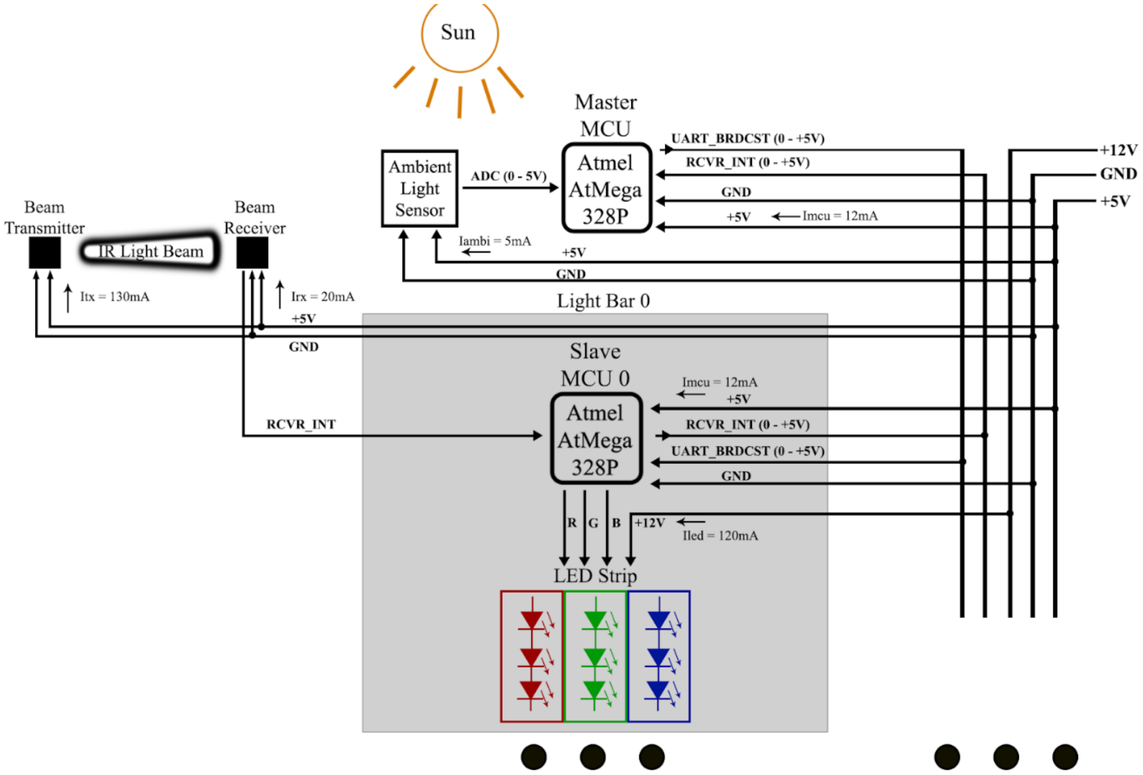


Figure 8: LED Stair Block Diagram

For the implementation of the microcontrollers notice that we utilize a master-slave architecture. Starting from the ambient light sensor data concerning the brightness of the surrounding environment is collected and reported to the master. From here the master uses this information in conjunction with its own animation procedures to transmit packets which specify which slave light bars should illuminate and how. This is done through a shared UART_BRDCST (UART broadcast) line. Note that only the master is allowed to communicate on this channel, slaves are only permitted to listen to it and will not respond unless a packet addresses them specifically.

Aside from ambient light sensing the system has only one other form of input: information concerning whether a person is occupying a given step. Mounted onto each step alongside a slave light bar is a infrared (IR) transmitter and receiver. The transmitter emits a light beam which is detected by the receiver and signals that a given step is unoccupied. Once a person steps into and thus trips the beam a signal is routed to both the step's respective slave light bar and to the master. The slave light bar uses this signal to illuminate its led strip white while the master utilizes the signal to disable animation on all unoccupied steps.

0.3.3 Conclusion

Reflecting over the term of our MQP, our team holds great pride in what we were able to achieve for both the LED Stair and RasPiano projects. Moreso do we hold pride in the things we learned in completing this project. We were capable of gaining additional experience in our specific fields of interest such as real-time embedded development, analog circuit design and signal processing. We gained practical engineering skills and gained experiences with specific tools such as Solidworks for CAD modeling parts, Eagle for pcb design, and practicing signal analysis techniques. And perhaps most important of all were the skills we gained in project management. Utilizing a Agile project workflow we learned to improve how we scheduled and prioritized tasks given resource and time constraints. Finally we learned to maintain a strong group dynamic and develop an environment where each team member can comfortably bring up issues, ideas, and solutions to each other and build on them constructively.

Before we conclude our executive summary we now cover the state of both the LED Stair and RasPiano projects. Specifically we will go over the features we were successful in implementing as well as the future work we recommend to improve each system.

0.3.3.1 RasPiano

At the moment the RasPiano system has multiple functional blocks but is not yet capable of operating as a singular unit/device. All color organ circuitry and hardware works properly when powered and given an audio signal. Each filter within the filter chain behaves as a operational filter but cannot be modified due to a SPI communication error between the Arduino and digital potentiometers currently in use. With respect to software the Raspberry Pi properly boots into Raspbian but does not have the custom-designed software implemented. As our team ran out of time to implement the software for the RasPiano we decided to provide third-party software to implement the oscilloscope and spectrum analyser applications. A figure of the applications provided as well as the synthesizer is shown below.

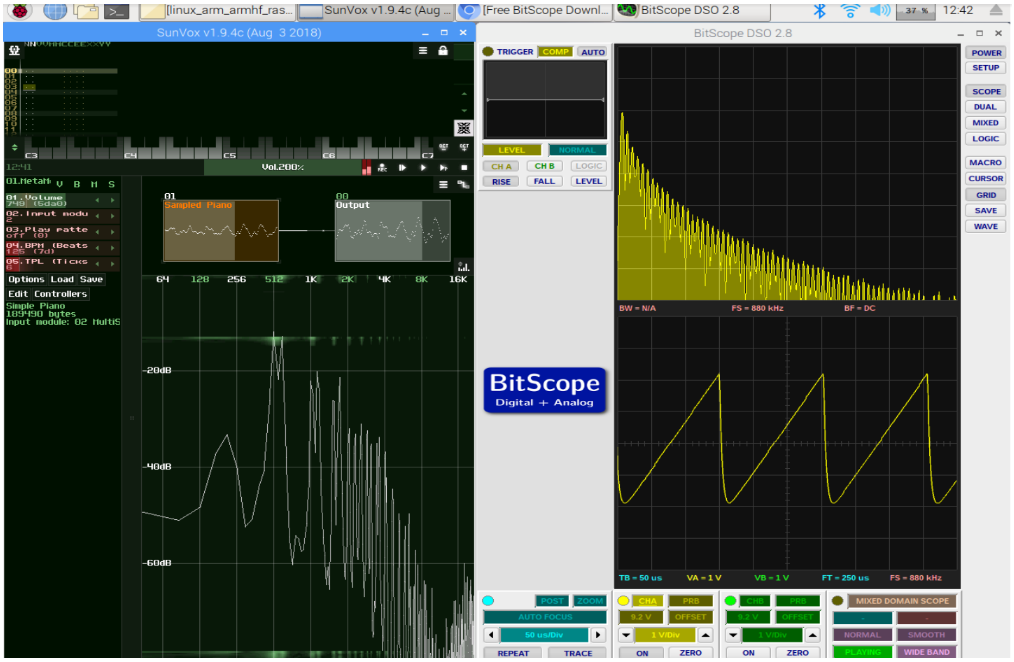


Figure 9: RasPiano Applications

On the left is the sunvox synthesizer which takes keystrokes from the USB-connected MIDI keyboard and outputs corresponding piano key samples. This analog signal is outputted from the Raspberry Pi's auxiliary 3.5mm analog output. On the right of the screen is the BitScope Micro DSO application. Also connected through USB, the BitScope Micro oscilloscope passes data to this application to trace an analog waveform and provide spectrum analysis. Unfortunately as we could not construct the corresponding application there is currently no way to view the filter chain's transfer function.

Despite the software not being complete for the RasPiano, the applications pictured above do provide a reasonable demonstration for how the system should be configured. Additionally they also meet our project requirements of providing the oscilloscope and transfer function functionality to the device. For future work there are three primary recommendations we have:

- Troubleshoot and repair digital potentiometer control within the filter chain
- Implement the custom oscilloscope, transfer function, and spectrum analyser applications
- Add custom application control from auxiliary MIDI keyboard buttons
  - Interface this control with the digital potentiometers and transfer function application

As mentioned previously, control over the digital potentiometers used within the different filters of the filter chain was unsuccessful due to an unknown SPI communication error. To ensure this component of the RasPiano is ready for interfacing with the Raspberry Pi it would be critical to troubleshoot and repair the error. With this done each filter would then be capable of being digitally modified and as such controllable by the Raspberry Pi and its applications.

The next recommendation for future work is developing the custom oscilloscope, transfer function, and spectrum analyser applications whose design is specified within part of this report. With these applications operational input from the auxiliary MIDI keyboard controls can be utilized to control the system. Once a user modifies a control for a filter these applications can then provide the appropriate configuration to the digital potentiometers and update the corresponding transfer function on screen.

0.3.3.2 LED Stairs

Currently, all appropriate PCB designs and software is complete for the LED Stair system. With the system powered and all master-slave, trip sensor, and led strip connections appropriately made, the system will begin to perform autonomous animation on all connected light bars. When a sensor is tripped its respective light bar will illuminate white. Simultaneously all other unoccupied sensors will have their respective light bars off. Five seconds after the last sensor is tripped the system will resume full animation.

To add or remove slaves simply disconnect the desired amount from the end of the slave light bar chain. To ensure appropriate animation across all connected devices one must update any changes in the amount of light bars within the master's program. Specifically, one must modify an internal value which specifies the number of connected slaves as well as verify the address and both the software and physical topographies of each slave. A figure of a team member actively occupying two "steps."is shown below. Notice how only the occupied steps are illuminated and none of the rest.

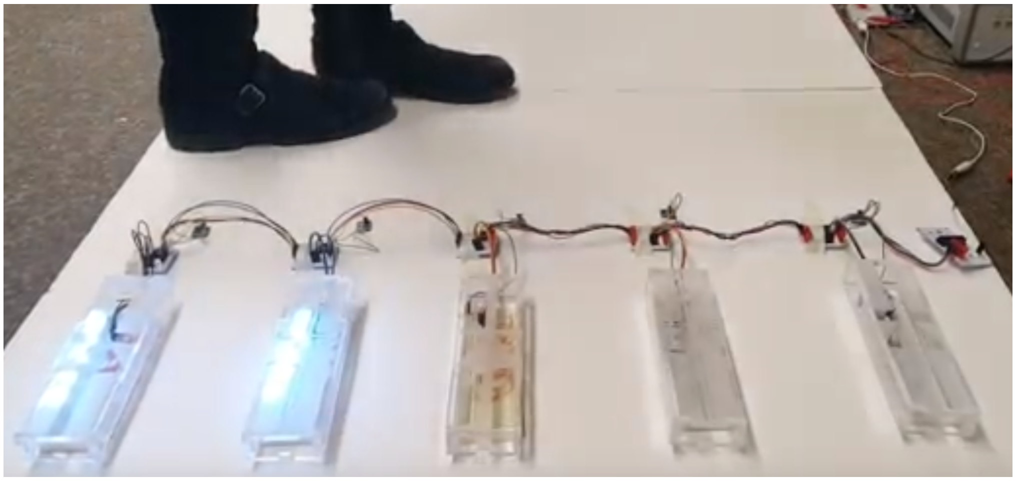


Figure 10: Snapshot from LED Stair Demo Video

Moving onto future work a list of improvements that can and should be made is below.

- Reprint slave PCBs
- Fully install LED Stairs at Atwater Kent front staircase
- Add automatic topography identification

Starting with the slave pcbs an error was made in their original design which caused the sensor trip signals to also be transmitted to other slaves in the system. As such when one step was occupied, the entire system would illuminate white instead of just the occupied step. For our demonstration pictured in figure 8 this was resolved with minor modifications to each PCB. However, for long-term stability and efficiency the modified slave PCBs should be printed and used instead.

After reprinting the slave PCBs the system should be fully installed at its intended location at the front Atwater Kent staircase. Due to resource and time constraints our team was not able to perform an installation. However at its current state the system is fully operation and ready for installation.

A final recommendation for future work is to add automatic topography identification. The purpose of this feature is to allow the system to be easily rescaled by simply adding or removing slave light bars without the need for modifying the master's software. For proprietary installations such as that on the front Atwater Kent staircase this feature is not necessary as the system should not be rescaled. However for more artistic or demonstrative purposes this feature could be a relevant addition. The inspiration for this feature primarily stems from interest in originally designing the system to allow modification for residential use. With the approach taken for UART broadcast communication from master to slaves, the LED Stair system provides a base for a potentially competitive design to systems such as the Nanoleaf Auroura/Canvas smart light panels. A sample of which is shown in the figure below.

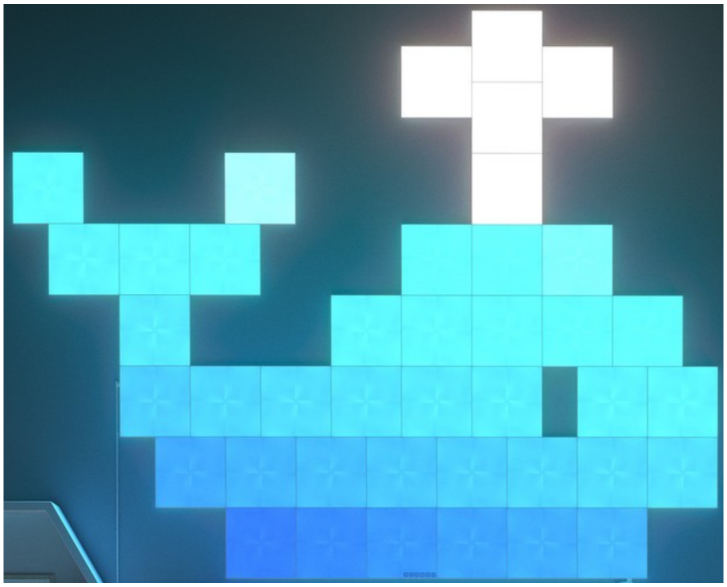


Figure 11: Nanoleaf Canvas Light Panels

This product utilizes a similar master-slave architecture with each light panel serving as an individual slave. Each is likely individually addressable with the overall topography identified and updated on power-up or runtime.

0.4 Report Organization

This report is broken into three main parts: Introduction, RasPiano, and LED Stairs. The first part will detail the idea process and transitions this project went through to end with our final project ideas. Next, we will discuss the RasPiano from the idea to final product. The third part will reflect similar organization as the second, except focus on the LED Stairs.

0.4.1 Organization of Part One

The introduction part is split into three chapters. The first chapter, Background, introduces our original project idea and what specific work was completed on the LED Displays within the Pumpkin Lounge within section 1.1 AK Matrix Display. It will then flow into section 1.2 Improving AK which explains how our project grew from working on the AK Matrix Displays to working on the building as a whole and why. The next section, 1.3 Brainstorming, shows our idea process and how we took our team goals and morphed these into project ideas. Lastly, 1.4 The Big Three includes our final project ideas we presented to Professor McNeill. The chapter was meant to reflect upon the process of innovation for our team, leading up to our final project ideas.

Chapter Two, Electrify Atwater Kent, gives a summary of the original project ideas and goals. Therefore, it is divided into two sections, 2.1 RasPiano and 2.2 LED Stairs. Each of these sections give a brief introduction to each project along with why they were important to us. This chapter is meant as an introduction to the finalized project ideas to explain why they were chosen in light of our goals.

The third chapter is Report Organization, this explains all sections within the report and gives reasoning to our organization. The chapter is broken into three sections to discuss each part within the report. Since this report includes two projects, the organization is a bit unique and this chapter can be used as a guide.

0.4.2 Organization of Part Two

There are four chapters within this section to flow through this project from beginning to end. We felt this would be best organized by first introducing the project, mainly a more detailed version of section 2.1 followed by a design chapter to explain in detail all decisions made throughout the project that led us to our final product. Next, continuing with the flow, we will discuss the final project and results of this project. Lastly, we reflect on our results, compare them to our original goals, and discuss future recommendations.

The first chapter of the part, Introduction, is fairly simple. It just meant to introduce the project before going into detail on the design. This chapter is purposely short as it is already discussed earlier in the report.

The second chapter in Part One is the “meat” of Part Two, so we split this into many sections. It starts with the design of the overall product in section 4.1 General Overview, this includes the design process and system specifications. Followed by this, we go in extreme detail of all components that make up this product. To organize this, we broke the product into three main systems. As with almost all technology, there will be input from the user, algorithms, and an output to the user. Therefore, we reflected this basic system and started with section 4.2 Inputs. For the RasPiano, there will be three inputs, and within this section there is four subsections. First, the three inputs, keyboard, microphone, and auxiliary. The last subsection is not an input, but was included in this section as this is the sub system that combines the inputs. This is our summing circuit. We then move to our "algorithm" section, 4.3 Filter Chain. The filter chain is the heart of this project, taking the input waves and adjusting them according to parameters by the user. This section will move through the research accomplished to chose the sections of our filter chain, the filters, clipping distortion, and lastly, the final filter chain design. The final section of this chapter is 4.4 Outputs and includes subsections discussing the Raspberry Pi implementation and software design, selections for visual output, selections for audio output, and our color organ. Chapter 4 is where to find every detail on all design decisions made in this project for our RasPiano.

Chapter 5 is to show the final design of the RasPiano, subsequently being named Final Design; this is broken into two sections. The first section, 5.1 Connectivity of Subsystems details how all sub-systems described in Chapter 4 connect together. This is followed up by section 5.2 PCB Design, to discuss the design of the PCBs for the RasPiano. The chapter can be used as an explanation of the final product and how it was implemented.

Chapter 6, Conclusion, is introduced by reflecting on our original goals for the RasPiano and what we hoped to get out of this project. It summarizes everything we were taught from this experience and how the end result compares with our original end goal. This chapter is followed by Chapter 7, Future Recommendations, that primarily discusses features of the RasPiano we were not able to include in our final design. Due to timing constraints, not all of the features from our original idea were implemented. However, research was still completed for these features, so any and all information gathered for these additions are included in this section.

0.4.3 Organization of Part Three

Similar to Part Two, this part contains the same four chapters. However, the sections within these chapters are much different as the LED Stairs is completely separate from the RasPiano. Part Three, LED Stairs, is divided into the introduction of the project, the design process, final results, and reflection.

Chapter 8 is meant to introduce the project LED Stairs to give an idea of the project's purpose for Atwater Kent as well as our team. This shows our original goals when beginning this project, much like section 2.2, but with a bit more detail.

Chapter 9 is largest chapter of Part Three, it details all parts selected for our final product as well as every decision made from project goals to final implementation. This was organized a bit differently than the RasPiano. The first two sections are the same, starting off with section 9.1 General Overview to give a larger picture of the LED Stairs, including the design process and system specifications. This then flows into 9.2 Inputs to discuss the two main inputs to the LED Stairs system, the step sensors and ambient light sensor. Next is section 9.3 System Configuration and Software - this section is rather large as it includes an immense amount of information on the "brains" of this system. Many different approaches in regards to software and communications were considered and are all discussed in this section. Lastly, this chapter ends with 9.4 Light Bar Housing and Circuit Hardware. This section goes more in depth on the hardware of the slave and driver PCBs as well as the design of the Light Bars. Chapter 9 is meant to give a detailed description of every part within the LED Stairs project, along with every decision made that led us to these chosen parts.

Chapters 10, 11, and 12 are structured identically to those in Part Two, except applied to the LED Stairs. First discussing the final outcome of the project and its implementation, then to reflect and mention future recommendations.

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Background

This project was first defined as simply fixing and improving upon the Pumpkin Lounge LED Matrix Displays. After working on this project, prior to beginning the MQP, we realized that this was not enough work. From here, we settled on our new project definition: Electrify Atwater Kent. To improve upon our beloved Electrical and Computer Engineering building, Atwater Kent (AK), we researched the state of the art and began to brainstorm ideas. Many decisions later, we came to our final project ideas:

- Fix the Pumpkin Lounge LED Matrix.
- Create LED stairs that respond to human interaction.
- Make an educational piano for the Pumpkin Lounge to invite visitors and students to the world of audio signals.

1.1 AK Matrix Display

In 2016, an MQP team of students designed the Atwater Kent Power Panel to bring an interactive, creative display to the Atwater Kent Pumpkin Lounge. This was intended to welcome visitors, but the project had a life span of less than one week once completed. Throughout the year of 2017, the matrix display was off during all hours of the day. When turning the displays on, there was a horrendous glitch; to make matters worse, the panel drew enough power to trip the circuit breaker and shut down the lab in AK113.

It was within this year that our soon-to-be MQP team had agreed to take on this project and bring life back to the Pumpkin Lounge. In the Winter of 2017, our team fixed the software glitches and all that was left was to fix was the power supply problem.

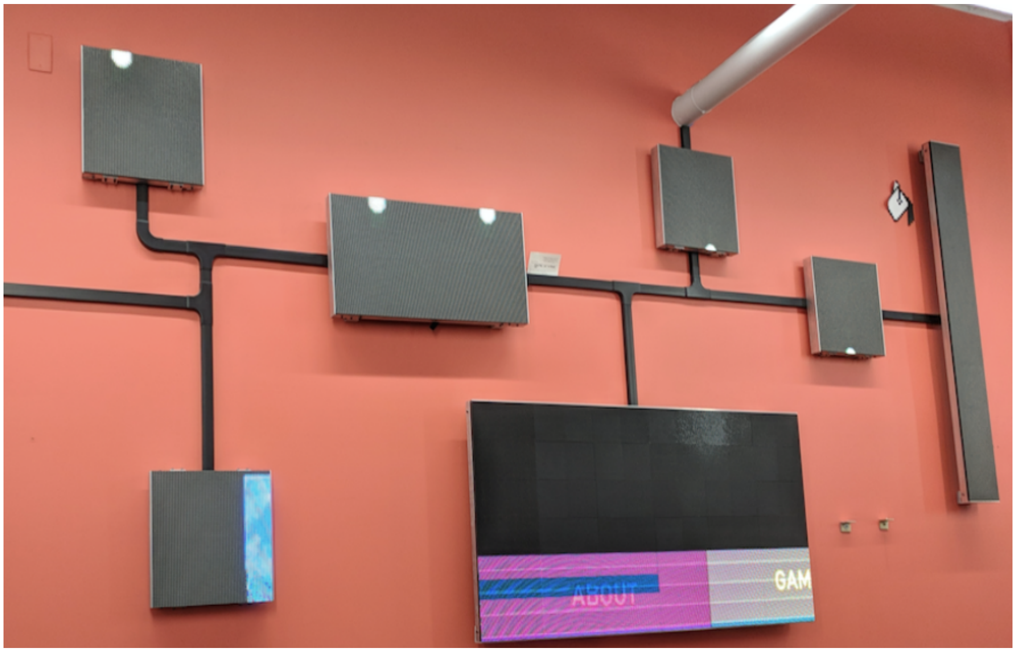


Figure 1.1: Pumpkin Lounge Matrix Display Glitch

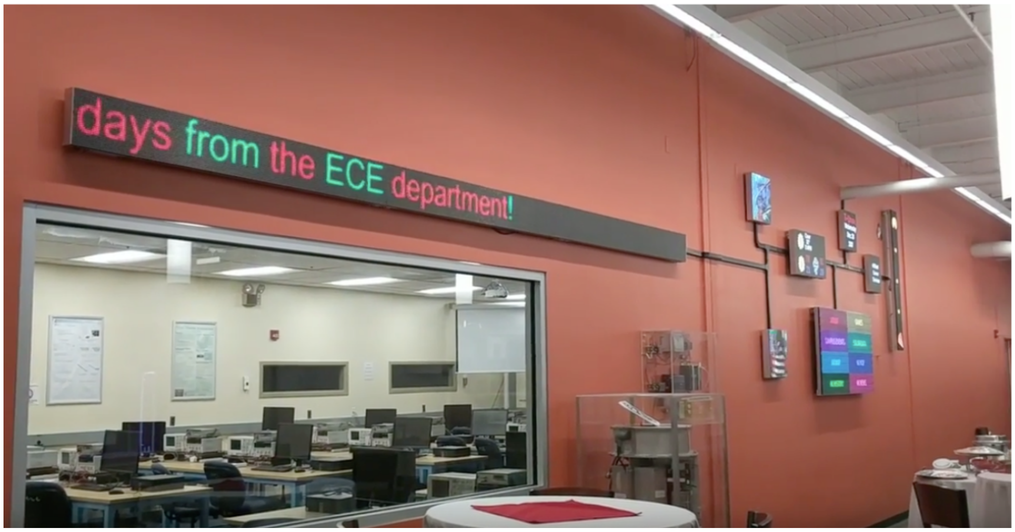


Figure 1.2: Pumpkin Lounge Matrix Display Fixed

Additional details on our contribution to this project can be found in Appendix A. Since we had not yet started our MQP, it became clear that there was not a large enough workload to fulfill our MQP requirement. From then until A Term of 2018, we began brainstorming ideas for peripherals of the Atwater Kent Power Panel. The list created can be found in Appendix B; some of our top ideas were: interactive map, touchscreen study-buddy table, juke box, and project illumination.

At the start of A Term, Professor McNeill gave the team books to help spark innovation, “A Kick in the Seat of the Pants”, which focused on the roles that one plays throughout the process and “A Whack on the Side of the Head”, which focused more on the dos and don’ts of the creative process. There were four roles:

1. Explorer: Researches and explores ideas.
2. Artist: Changes and plays around with ideas.
3. Judge: Analyzes value and narrows down ideas.
4. Warrior: Implements and fights for idea.

After the readings, each member wrote essays on the brainstorming activities and tricks taught in the book, as well the roles defined. One writing assignment helped show the strengths and weaknesses of each team member. This is important to know when working on the project, it gave us an idea of who to ‘lean on’ or who to help out and eases distribution of tasks. The other instructed us to write about two exercises we did while reading, leading us to ideas for our MQP, as well as gave us tools to use moving forward. More detail on the teachings from these assignments can be found in Appendix C.

In addition to the assignments, Professor McNeill prompted a question, “Why just fix up the Pumpkin Lounge?”. With the brand new Foisie Innovation Studio casting what felt like the greatest shadow over Atwater Kent, our academic building looked more worn down than ever before. Atwater Kent has been around for over a hundred years, so it goes without saying that it is in need of a tune up. The team agreed, and thus began our new and improved project to improve AK.

1.2 Improving AK

To begin our research, we walked around Atwater Kent to produce a list of things that we felt were good or bad about the building. Next, we headed over to Foisie Innovation Studios to walk through and do the same; afterwards, we came up with some conclusions. Details on these findings are listed in Appendix D. The main aspects that we felt AK lacked where Foisie excelled were lighting, open space, and windows. For a deeper look into innovation strategies used around the world, we also conducted some online research. We looked into some of the top technical and innovative universities in the world, as well as innovation strategies used by architects and designers for companies aiming to inspire innovation. All the information was gathered, recorded, and used within pictures and floor plans of Atwater Kent.

Continuing our research, we then conducted a student survey. In total we received over 100 responses with 86% of respondents being ECE students. Additionally, 42.2% of respondents were seniors, 25.3% juniors, 22.1% sophomores, and 7.4% freshman. The 3% other we assumed to a miscellaneous collection of graduate students and faculty. Overall, we were surprised to see how different many of the responses were. Ranging from people who felt Atwater Kent at its current state needed nothing more, and others going as far as to say the building should be removed and reconstructed entirely. Nonetheless, we were overjoyed that we were able to get our respondents creative juices flowing as with the many responses came numerous fantastic ideas for ways to improve Atwater Kent.

In creating and deploying our survey the goal was to view what objects and services students were interested to help generate more ideas for our project. To encourage respondents to imitate the roles of being the explorer and artist (see Appendix C) we formatted our three survey questions to:

- Generate as many ideas as possible regardless of how whacky or unrealistic.
- Generate ideas based off preexisting knowledge of what does not function well in Atwater Kent.
- Generate ideas on what could be added to the building to improve it.

As such, the three main questions we created, in order, were:

1. If you had an infinite budget, what would your dream tech building include?
2. What does Atwater Kent have that doesn't work?
3. What is Atwater Kent missing?

The questions were designed to be open ended in order to get a wide range of inspirational answers. We sent the survey via email to all ECE students and placed paper surveys in the Pumpkin Lounge next to a drop-in box. Many students expressed dissatisfaction about the lack of light and the general feeling of gloominess of AK, while some gave very specific replies about parts of the building in need of repair. We prepared a document listing every response organized by subject, this was given to both Professor McNeill as well as Professor O'Rourke and can be found in Appendix E.

1.3 Brainstorming

For this part of the project we all were focused on the role of the explorer. Our main objective was to come up with as many ideas as possible and document them. Alongside this list, we also wrote down a list of goals, see Appendix F. These goals were organized into 4 sections:

- Project Goals: We hoped to make plans for Atwater Kent to revitalize it. We want students to feel excited about working here. The building has a lot of history and character and we want to show that off while still looking ahead.
    - To make Atwater Kent:
        - Inspire creativity
        - Provide a welcoming environment
        - Embrace the history of the building
        - Promote modern technology and innovation
- Technical Goals: We looked forward to making use of the skills and techniques we've learned in our classes, in this building, and bringing it to life. We also wanted to hone some more practical skills that aren't explicitly taught in classes, such as soldering, PCB design, and coding.
    - Design and implementation of analog circuitry
    - Create an interesting application of signal processing techniques
    - Engage in IOT and embedded systems development
    - Hone practical electrical engineering skills (e.g. soldering)
- Academic Goals: We hoped to exceed all expectations set up for this project. The skills we apply in the project was a great addition to our resumes. We also planned to learn LaTeX to make a high-quality final report.
    - Impress Professor McNeill and earn an A
    - Meet all deadlines
    - Apply our specialties so we can document them on resumes
    - Develop a professional technical report
- Social Goals: This was to be a great experience to improve our teamwork skills, in addition to teaching us a lot about our ability to work independently.

  - Form a coherent team that performs
  - Improve our networking skills
  - Grow our work ethic
  - Learn to accept and admit when you need help to promote team progression

After completing our ideas and goals lists, the goals were ranked by importance and the top five ideas were selected, in addition to a lengthy list of small project ideas, to be compared with the goals. The original list, however, was still kept in the running as these five ideas were not fully decided on and still could use some artist on them. Multiple methods of value analysis was performed on these five ideas.

Our team then went back to the drawing board. We looked at our original list, discussed what we liked about our top five selections, grabbed ideas from the survey results, merged and edited ideas, and came up with new ideas.

1.4 The Big Three

The three project ideas used for final analysis were: "Bringing Jazz to AK," "Bringing Light to AK," and "Bringing Information to AK." With these defined projects, we could perform proper cost and value analyses. No longer were any of the ideas too small to perform or too similar to compare.

1.4.1 Bringing Jazz to AK

This project is centered around a piano located in the Pumpkin Lounge. This piano would output into a series of filters, each tunable and with a bypass switch. After the filters, the signal is fed into a computer which displays the signal in both the magnitude and frequency domains vs. time. Students and visitors will be able to turn on and off any number of filters and adjust them to see how they affect the signal. Headphone ports will be available for students who wish to listen in.

To avoid making the installation a distraction, the built-in speakers on the piano would need to be kept at a very low volume or muted altogether. The project may also include an auxiliary input in case a student wishes to use their own music. Furthermore, an individual volume knob for each headphone jack would make the installation more inviting for students with hearing impairments. The display should be adjustable for different timescales as well.

Possible filter types included:

- Low pass
- High pass
- Band pass
- Band stop

Other effects to consider:

- Amplifier/clipping distortion (OpAmp)
- Rectifier/other diode circuit

For this idea, we also came up with a list of possible components this project would require:

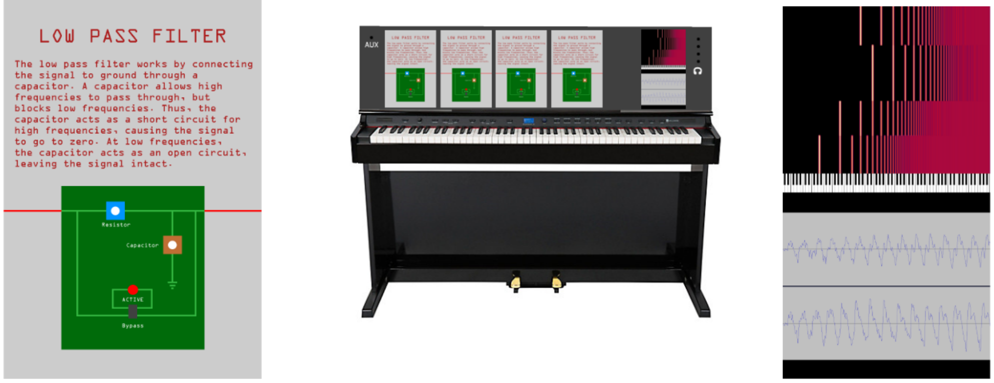
- Digital Piano ($500?)
  - Weighted keys
  - Line output
  - Multiple sounds
  - Auxiliary input (optional)
- Filters
  - Individual filter modules
  - Parallel filters (activate 1 at a time) or series (activate multiple)?
  - Research filter adjustability with analog components - cutoff freq, sharpness, etc.

    - Name of filter printed on glass pane
    - On/off switch (bypass)
    - Knobs for each filter - cutoff, sharpness, resonance…
        - OR just knobs for resistance/capacitance to show how values change filter
    - ALSO - could use different effects, e.g. clipping distortion, reverb/delay, etc.
    - Research implementation of various effects

- Audio Output
    - Hardwired headphones
    - Optional speakers - figure out mute button system

- Screen ($100-200)
    - One screen, two plots
    - Waveform display with zoom controls (i.e. oscilloscope)
    - FFT display - frequency log display w/ piano key overlay
    - Touchscreen pinch controls? Zoom in/out - more elegant than buttons

- Fix up LED Matrix



LOW PASS FILTER

The low pass filter works by connecting the signal to ground through a capacitor. A capacitor allows high frequencies to pass through, but blocks low frequencies. Thus, the capacitor acts as a short circuit for high frequencies, causing the signal to go to zero. At low frequencies, the capacitor acts as an open circuit, leaving the signal intact.

Figure 1.3: Bringing Jazz to AK Visual

1.4.2 Bringing Light to AK

Bringing Light to AK was a project idea that was based off a smart building idea. The sub-projects of this project include:

- Building-wide sensors
- Billie Jean ceilings
- LED stairs
- LED raceways

Sensors would be building-wide, with some communicating to LEDs. These LEDs would be for the Billie Jean ceilings (one for each of the two hallways on the first floor), the stairway LEDs, and the LED raceways. We aimed to have a common purpose for the sub-projects and felt all these ideas had a good common ground: lighting up AK. The hub of this system would be a visual representation of the active sensors in the building. At the time, we saw this as a 3D map of AK to be displayed in the Pumpkin Lounge, but this addition was still fluid.

Components that this project would require are listed below:

- Billie Jean Ceiling
    - LED Strips
        - Represent hallway "resistance"

  - Fix up LED Matrix
    - Show information on the hallway resistance
  - Wood Panel
    - Need 2 (one per hallway)
  - Arduino
    - Need 2 (one per hallway)
  - Acrylic Cubes

- LED Stairway
  - LEDs
    - One bar per stair
    - Visible Spectrum

- LED Raceways
  - Locations:
    - Basement across wall under chalkboards
    - Wall between bathrooms on 3rd floors
    - Two hallways on 2nd floor

- Building-wide sensors

- 3D Model of Building that is representing IOT, set in pumpkin lounge.

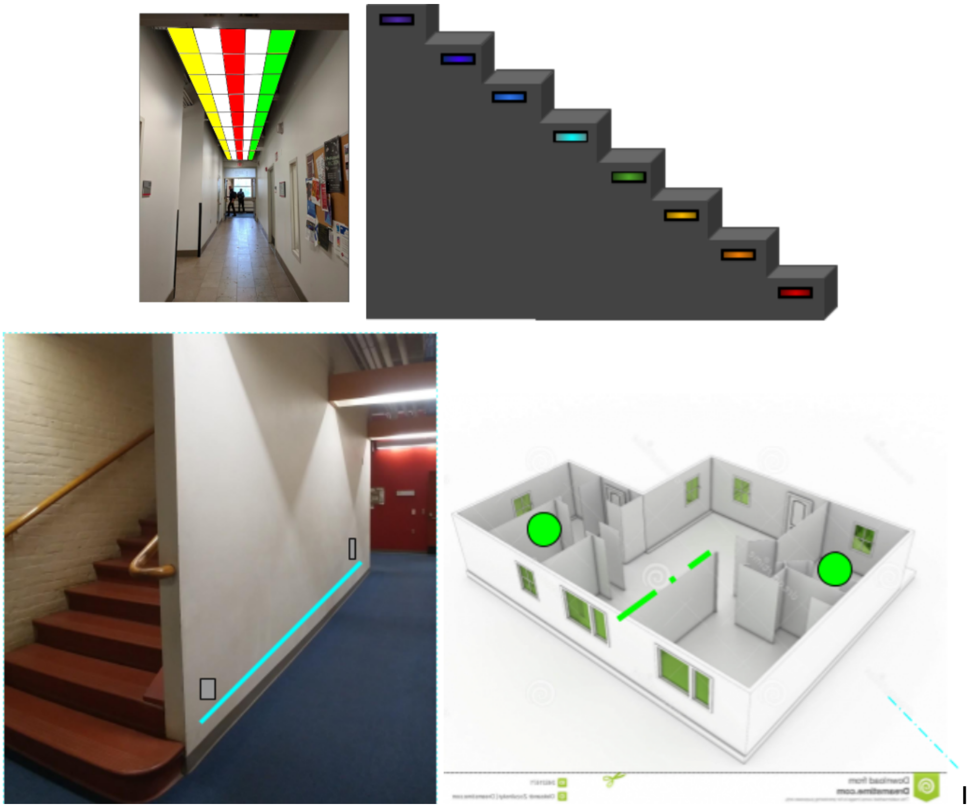


Figure 1.4: Bringing Light to AK Visual

1.4.3 Bringing Information to AK

What defined this project was the large array of display panels that would be placed around the building. There were four types of displays, each tasked with a specific purpose:

- Mailbox/Professor Information Display
- TV Matrix
- MQP Posters/Pseudo Windows
- LCD Dry Erase Boards

Components for this project included:

- LCD Dry Erase (Replace basement blackboards)
  - Ability to display a circuit or anything and take notes with marker
  - Corner email input to get pdf version sent (reach goal)
  - Touch screen
  - Engi-chip magnets, nfc chips (reach goal)
  - ECE digital help center
  - Sensor to save power
- TV Matrix (2nd floor on empty wall space)
  - Sensor to save power
  - Catch phrase recognition
  - Toggle button to switch between menus
    - Lab ideas/instructions menu
    - Definitions and equations menu
    - Historical AK(/ECE?) facts menu
- Mailbox/Professor Information Display (Replace paper mailbox list)
  - LCD Display
  - Sensor to save power
  - Button to toggle between professor office, projects, and course information or mailbox information.
- Pseudo Window/MQP Poster (3rd floor hallway)
  - LCD Display
  - Sensor to save power
  - Joystick
    - Up/Down : Pseudo Window/MQP Posters
    - Left/Right : Swiping through posters or live feeds
  - Maybe easter egg games in the display (like pacman)
- Fix LED Matrix

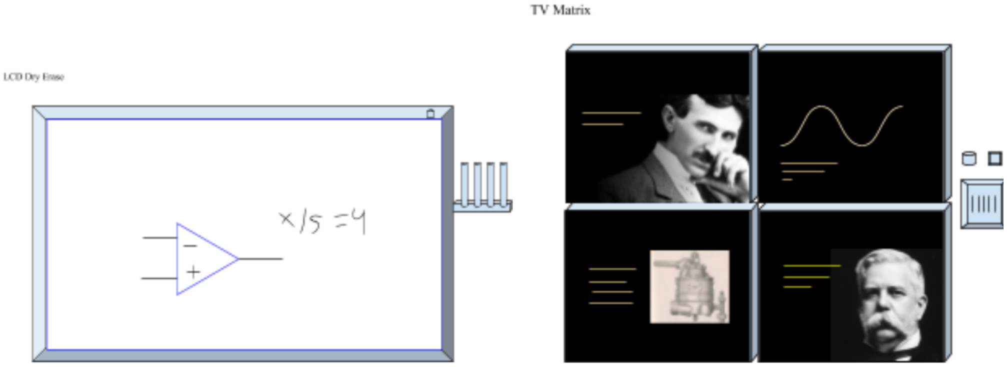


Figure 1.5: Bringing Information to AK Part 1

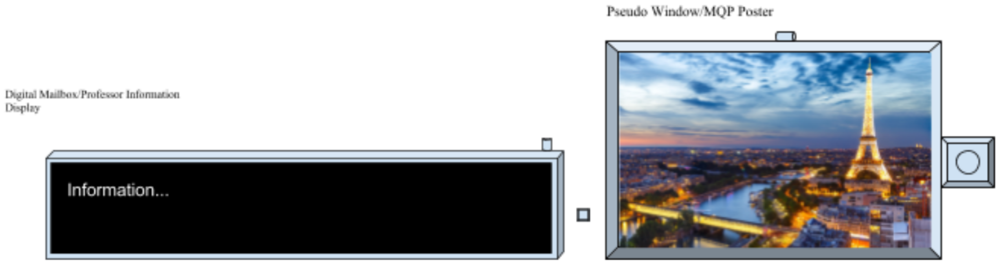


Figure 1.6: Bringing Information to AK Part 2

1.4.4 Value Analysis

Before determining the value of our top three ideas, we first needed to calculate the value of our goals. Each member performed their own value analysis on each goal, these three analyses can be found in Appendix G. These results were averaged and revised by the team to use as means of weighing each goal within our idea value analysis, as shown in Appendix G as well.

The scores for each project were as follows:

- Bringing Jazz to AK - 8184
- Bringing Light to AK - 8305
- Bringing Information to AK - 9192

However, these scores were not convincing anyone. Ultimately, it was going to be up to us on what direction this project goes in, and our team was pretty divided in opinion. Each idea brought about different challenges, much like how our team brings different interests and skill. Therefore, we prepared a presentation to argue each idea to the customer, Professor McNeill. He gave three final project definitions:

- LED Stairs (top priority): Make the AK stair way, visible from the outside, light up when being used.
- Piano (selected as second priority): A piano in Pumpkin Lounge that shows signals in real time.
- Billie Jean Ceilings (reach goal): LED ceilings that react to human presence in the first-floor hallway of AK.

The LED Stairs grabbed inspiration from the bringing light to AK idea mixed with a Michael Jackson music video. From first glance, this project offered experience with IOT, embedded, and analog circuitry. The Piano was taken straight from Bringing Jazz to AK, and related to Jeff's audio signal learning goals. The Billie Jean Ceilings was a reach goal that was later abandoned due to time constraints. This project primarily achieved our project goal of effecting more than one area within the building, but did offer additional learning experiences as most of the technical challenges would be very similar to the LED Stairs project.

Chapter 2

Electrify Atwater Kent

One of the top priority project goals the team agreed upon was the need for our project to affect more than one aspect of AK. This, along with our varied interests in electrical and computer engineering, led us to the desire to complete more than just one project. In addition, all three of these projects would have large wow factors and allow our department to stand out.

Electrify Atwater Kent will bring LED Stairs, the RasPiano, and a working LED Matrix Display to the Electrical and Computer Engineering building of Worcester Polytechnic Institute. The LED Stairs is composed of Light Bars the team has designed that are controlled by infrared sensors and MCU communication. Our RasPiano is a piano whose audio signals are manipulated by a filter chain and displayed with the help of a Raspberry Pi. Lastly, the LED Matrix Display is a past year's project that we have ensured can now work and amaze students for the following years.

Upon surveying the students to get feedback on the current academic building, a recurring answer was that the building felt dark and "depressing." Other students noted that they would like to see more electrical engineering projects on display, or displayed better. Some students even specifically wrote that they wanted the LED Matrix in the Pumpkin Lounge working. This project will bring life and light to the building to create a better environment for the students that spend most of their college career inside Atwater Kent.

Each member of this project had a different area they desired more experience with. These three projects offered practice in all of these areas. For examples, Shannon was able to design and build the circuits of the sensors within the LED Stairs, Juan worked with the MCU communication for the LED Stairs, and Jeff designed the filters for processing the audio signals of the RasPiano. These along with further developing our soldering, PCB design, and other skills made this project perfect for strengthening us as electrical and computer engineers.

2.1 RasPiano

Atwater Kent displays projects in the Pumpkin Lounge advertising Robotics and Aerospace, but the electrical components of the projects go overlooked. The goal of the RasPiano is to be an educational piece for students interested in audio, embedded, and analog electronics. In addition, the piano will bring more positive energy and encourage a social atmosphere for the Pumpkin Lounge.

2.2 LED Stairs

The LED stairs will be a monument to the first academic building in the country dedicated to electrical engineering, Worcester Polytechnic Institute's Atwater Kent. Light bars will light up the entrance staircase of Atwater Kent, visible from both the inside and outside of the building, showing off electrical engineering in a lively, creative fashion.

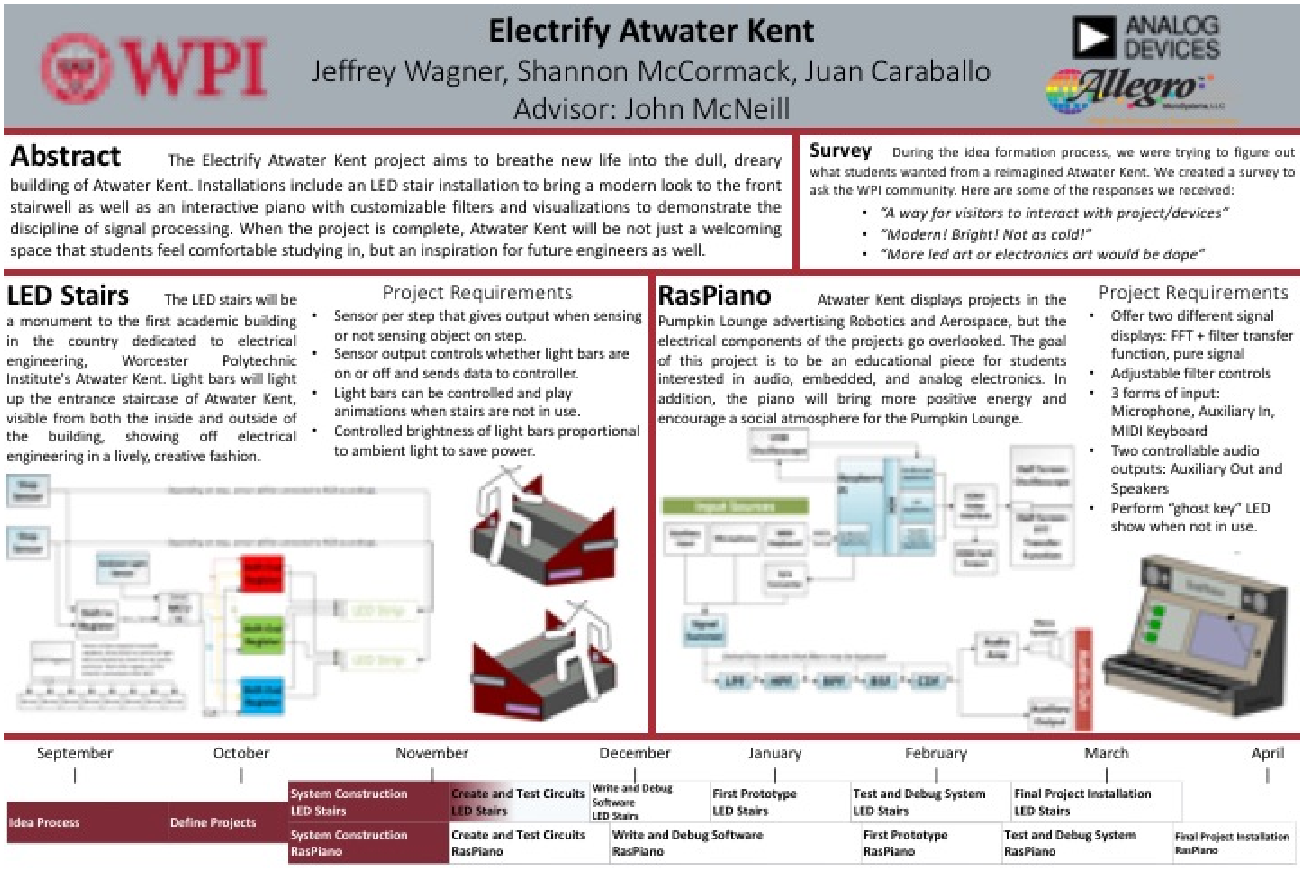
2.3 Project Management and Tasks

Our project started off a bit different, the team was not formed by signing up for the same term-long project, rather the team was formed first and the project definition revolved around our needs and goals. Originally, we did have an idea of where we were heading with the LED Matrix, see Part One, Chapter One, Section One, but after completing this before the start of our senior year, we needed to decide on a new end goal. At the start of A-Term, the first term of our MQP, the team knew they wanted to improve AK, but was unsure of what way. Therefore, this term was marked as our idea phase.

The team decided that for the idea phase, a flat team management structure would work well. This way, all group members are communicating the ideas can freely flow. More specifically, it would help keep us in the explorer role of innovation before heading in the direction of the judge. At this point, the team gave simple assignments such as learning LaTeX and creating, morphing and adding to ideas. All tasks were organized using Taiga.o. Taiga.o is a platform that helps organize and assign tasks, scrums, and sprints. Using this, we implemented sprint boards where every week new tasks would be assigned to a new board and members could update the progress of each task throughout the week. During the process of better organizing our sprint boards on this software, the team found a potentially powerful feature within the Taiga platform. This was the ability to create a wiki, commonly used within Agile development to organize important information that outline processes and provide reference information to use later in a project. The wiki included summaries and notes of readings, basics and syntax of LaTeX, research notes, and passages such as these. It was a good way to organize useful documents for other team members.

Nearing the end of A Term, the team knew we needed to revise our management structure and task organization. Taiga was useful to check on everyone's progress, but the more tasks that we needed to assign, the more unorganized and more of a hassle it proved to be. In addition, it was harder to see the larger picture as each week a new board was created. Therefore, when B Term started, the team began using a Gantt Chart to track progress, with weekly sprints to update tasks, and all scrum-like communications were done through Slack. In addition, a second more detailed schedule was also posted to use for weeks two and three. Since our first sponsor presentations were in week three, we had very specific deadlines and tasks that needed to be completed.

For our first sponsor presentation, we created the poster below:



Electrify Atwater Kent

Jeffrey Wagner, Shannon McCormack, Juan Caraballo
Advisor: John McNeill

Abstract The Electrify Atwater Kent project aims to breathe new life into the dull, dreary building of Atwater Kent. Installations include an LED stair installation to bring a modern look to the front stairwell as well as an interactive piano with customizable filters and visualizations to demonstrate the discipline of signal processing. When the project is complete, Atwater Kent will be not just a welcoming space that students feel comfortable studying in, but an inspiration for future engineers as well.

Survey During the idea formation process, we were trying to figure out what students wanted from a reimagined Atwater Kent. We created a survey to ask the WPI community. Here are some of the responses we received:

- *"A way for visitors to interact with project/devices"*
- *"Modern! Bright! Not as cold!"*
- *"More led art or electronics art would be dope"*

LED Stairs The LED stairs will be a monument to the first academic building in the country dedicated to electrical engineering, Worcester Polytechnic Institute's Atwater Kent. Light bars will light up the entrance staircase of Atwater Kent, visible from both the inside and outside of the building, showing off electrical engineering in a lively, creative fashion.

Project Requirements

- Sensor per step that gives output when sensing or not sensing object on step.
- Sensor output controls whether light bars are on or off and sends data to controller.
- Light bars can be controlled and play animations when stairs are not in use.
- Controlled brightness of light bars proportional to ambient light to save power.

RasPiano Atwater Kent displays projects in the Pumpkin Lounge advertising Robotics and Aerospace, but the electrical components of the projects go overlooked. The goal of this project is to be an educational piece for students interested in audio, embedded, and analog electronics. In addition, the piano will bring more positive energy and encourage a social atmosphere for the Pumpkin Lounge.

Project Requirements

- Offer two different signal displays: FFT + filter transfer function, pure signal
- Adjustable filter controls
- 3 forms of input: Microphone, Auxiliary In, MIDI Keyboard
- Two controllable audio outputs: Auxiliary Out and Speakers
- Perform "ghost key" LED show when not in use.

September | October | November | December | January | February | March | April

Idea Process | Define Projects

System Construction LED Stairs | Create and Test Circuits LED Stairs | Write and Debug Software LED Stairs | First Prototype LED Stairs | Test and Debug System LED Stairs | Final Project Installation LED Stairs

System Construction RasPiano | Create and Test Circuits RasPiano | Write and Debug Software RasPiano | First Prototype RasPiano | Test and Debug System RasPiano | Final Project Installation RasPiano

Figure 2.1: B Term Sponsor Presentation Poster

As shown in this poster, we had created a general project schedule for the rest of the year. It had been obvious for a while that this MQP would extend to a four-term project, so this was incorporated in our diagram. The term progressed and the team was working hard to prepare for a first prototype. This term involved various design changes, additions, and the removal of the reach project.

C Term focused hard on getting working prototypes for both projects since D Term was meant to be dedicated for testing, debugging, installation, and report writing. Therefore a detailed Gantt Chart was created for this term as well. During this term, some features to the RasPiano and LED Stairs were removed due to time constraints.

When entering D Term, we had a first prototype for the stairs completed, but with new large design changed that needed to be made. The RasPiano was still not at the prototype phase. Continuing into the term, it quickly became clear that we most likely would not be able to install both projects. Therefore, we held a team meeting to revise the term goals and tasks. We wanted to prioritize our goals under the assumption that we may not complete the third, or fourth, goal. At this moment in the project, Juan and Shannon still had a large amount of work ahead of them to reach the goal of making our ideal RasPiano first working prototype. That being said, we had planned to continue to dedicate our time to the GUI and communication tasks before starting on the installation of the stairs.

The problem with this was the possibility that unforeseen problems could arise down the road that continue to slow our progress and could lead to us not having enough time to install the stairs. With all the money and work that has been put into this installation, we decided unanimously that it was in our best interest to put the RasPiano on hold and primarily focus on the report and stairs installation. Therefore, alongside writing the report, we will be working towards the completion of the LED Stairs installment. Once these are entirely installed, we can then get back to the RasPiano with the time we have left.

As we were aware that the amount of time left over for the RasPiano might be minimal, we came up with a “plan-B” working prototype. This prototype would involve completely digital filters through our synthesizer. The filter circuits in this case would be eliminated, leaving only the color organ circuit being used within our system. We recognized that this was not ideal, but we felt that we would rather have this solution with the stairs installed rather than not being able to install the (extremely pricey) stairs and have the ideal RasPiano first prototype. In addition, the most valuable thing we received from this project is the experience and knowledge that came along with all the work we had completed. Therefore, even if a lot of our work will not be shown in our working prototype, all work that was accomplished within this project is fully detailed in this report and taken with us in our future journeys. Our new priority list was as follows:

1. Report
2. Light Bar Construction/Installation
3. RasPiano Working Prototype - implement filters digitally in RasPi
4. RasPiano with All Features - use PCBs for analog filters

As shown above, our report took top priority over the stairs because this is our way of showing everything we have been taught through this experience. We felt that since this report goes in depth on every decision throughout this project, that it is best at representing all of our achievements, and thus takes top priority.

A week later, we proposed the new goal priorities to Professor McNeill. We had assumed that the money spent on the installation of the stairs would weigh heavily on the decisions for goal priorities, but we wanted to confirm this. It goes without saying that continuing our progress on the RasPiano would have a great amount more of educational value than installing 32 Light Bars. Therefore, it would be more valuable to our team as engineering students to continue this rather than install the stairs, but with all the expenses that came along with the original plan to install, we did not think this was a valid option.

In this meeting, we proposed the new goal priorities and received approval from Professor McNeill, but before proceeding we brought up a second proposition.

1. Report
2. Light Bar Working Prototype - 5 steps
3. RasPiano Working Prototype - all features included

We were aware that this proposition would mean a good amount of our budget was spent on parts that will go unused. However, we needed to confirm that this cost what a higher priority than the educational value we could potentially have to gain from continuing work on the RasPiano. Professor McNeill said not to worry about the expenses, that the decision was in our hands.

Therefore, it was decided, our next project tasks were to implement an LED Stairs demonstration using five steps and build a working prototype of the RasPiano with its ideal features. Alongside these project tasks, we wrote sections of the report, this still being our top priority.

Part II

RasPiano

Chapter 3

Introduction

The initial idea for the RasPiano came from the student survey. One of the questions asked what students would put in their dream tech/engineering building, and one student simply responded “a piano.’ The team wasn’t very interested in this response at first, but after some brainstorming, we saw how we could transform a piano into an interesting teaching tool that would be applicable for an MQP.

The central idea the team came up with was to use a piano to teach visitors how analog electronics can be used for signal processing in an audio context. Hopefully, it will provide new information to users and expose them to a new perspective on the applications and disciplines of electrical engineering. As an interactive project, it will also help make students more interested in the Atwater Kent building. The overall effect for students and visitors will be to make a more welcoming experience in Atwater Kent. The team was also hoping to earn some valuable experience from working on this project. One of the reasons this project was chosen was that it would be a great learning experience in the realm of analog circuits as well as both analog and digital signal processing.

Chapter 4

Design Options

4.1 General Overview

This chapter covers the design of the RasPiano system. First, the overall system architecture is discussed, followed by the design of individual subsystems.

4.1.1 Design Process

Before beginning work on designing the system, the team had to consider the requirements and use cases. These all stem from the user experience. There were three main cases or perspectives to consider: when the system is not in use, someone is using the system, or someone is watching somebody else use the system. Based on these cases, the team put together a broad description of the user description.

When the piano is not in use, it should be inviting and welcoming to students and visitors in the room - there should not be anything that indicates to them that it is for display only, it should appear to be interactive as obviously as possible. Furthermore, despite the piano being the centerpiece of the project, any student should be able to use it and have a similar experience, regardless of their skill in playing the piano. Any non-instrument controls should be easy to reach for the person playing the piano, and ideally, for any spectators as well. Any screens should be easily viewable by both a player and all spectators.

These considerations helped the team develop a few more ideas for features for the project. One idea to make the piano more inviting was to create a “ghost piano’ feature, inspired by the player piano. The basic idea is to have an “idle mode’ when nobody is using the system that plays music automatically. Instead of the self-actuating keys in the player piano, an LED above each key would illuminate in time with the current song, determined by a stored MIDI sequence. The team also added two extra input sources to the design to improve accessibility - a microphone, so users can speak and use their voice as an instrument; and a 3.5mm auxiliary input, so users can plug in their phone and play music from that.

Aside from these new features, the team also organized the original set of features from the inception of the project. The primary input method would be a piano keyboard, although we had yet to decide on some specifics (e.g. MIDI input vs. digital piano, sampler vs. synthesizer). The team settled on using the four basic filter types of lowpass, highpass, bandpass, and band stop. An extra effect was added to the filter chain as well: a clipping distortion circuit to show how analog circuitry can create nonlinear effects. The signal would be output to an oscilloscope as well as an FPGA to handle FFT and transfer function calculation and display. In addition, the audio signal will be passed through a set of speakers set to a quiet level and a set of headphone jacks so users can listen on their own headphones.

After this process, the team began drawing up a simple block diagram to act as a guide for the design journey. The first draft is shown below.

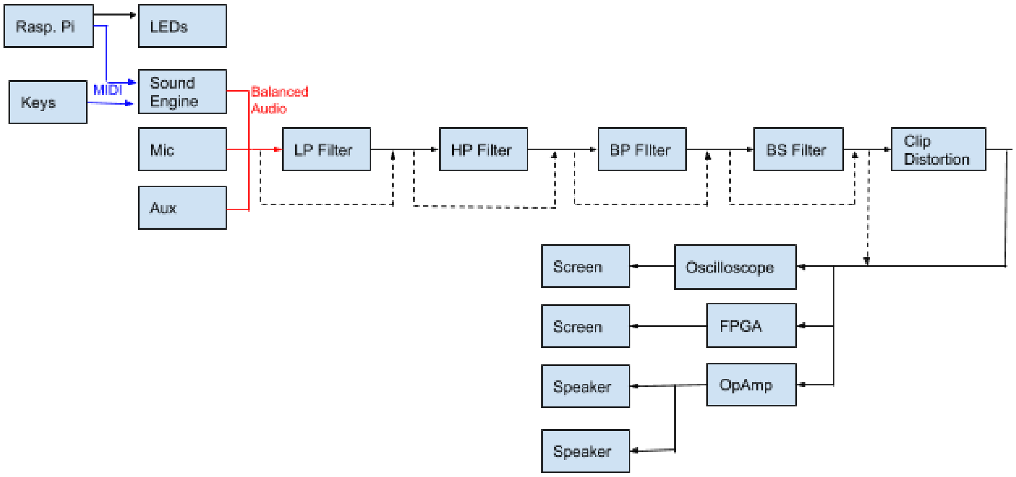


Figure 4.1: Initial RasPiano Block Diagram

Over the course of the project's development, the team made a handful of changes, additions, and removals to the system. Many of these were due to time and budget concerns. For instance, the FFT application was moved from an FPGA to the Raspberry Pi, as that was an environment the team was much more familiar with. The ghost piano feature and LED keyboard illumination were scrapped as that would be too much time invested for a feature with too little impact. Originally, the filter controls were planned to be physical potentiometers, but that was changed to a digital system for better integration with the transfer function calculation. For controlling these potentiometers, the team initially wanted a touchscreen, but then shifted to using the built-in controls on the piano due to budgetary limits. A color organ was also added for an extra bit of visual flair and a different application of filters and signal processing/conditioning. The final block diagram is shown below.

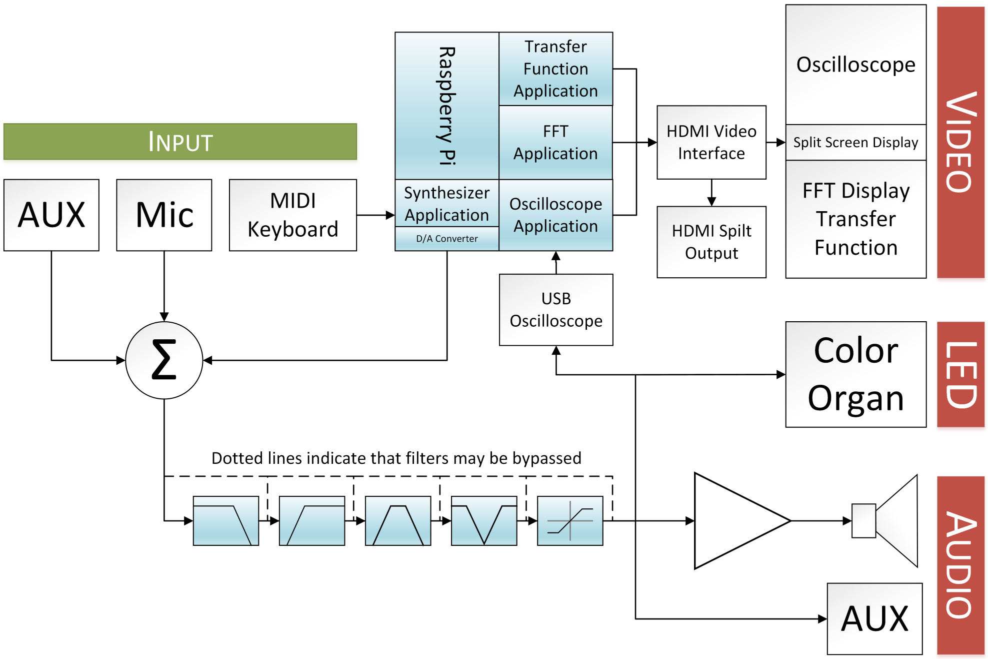


Figure 4.2: RasPiano Final Block Diagram

4.2 Inputs

An important theme of the RasPiano is interactivity. One of the first ways this is achieved is in the input. The team designed the RasPiano to allow the user to provide auditory input to the system in several ways. As per the system requirements, three input mechanisms were chosen - piano keyboard, microphone, and auxiliary input. In this section, we will take a closer look at the choices the team made for each input, how the solutions came to be, and the reasoning behind them.

4.2.1 Keyboard

The keyboard input was the original inspiration for the RasPiano project. On the original survey the team sent out, one of the responses mentioned a piano with little in the way of explanation. The entire RasPiano project was born from that survey response, and the piano input was one of the few constants throughout the project's design process.

The team's first investigations into piano keyboard input for the RasPiano skipped right past traditional acoustic pianos, which would be far too expensive, heavy, and difficult to interface with. One of the first realistic options that was considered was a digital piano. A digital piano would do the simple job of taking in power and outputting audio directly when a user plays the keys. However, of the models the team researched, many of them seemed to be out of the budget for the project.

The next option the team considered was a MIDI controller. Unlike the digital piano, a MIDI controller would not output audio directly, but just note data in MIDI format - i.e., which keys were being pressed and how hard. Using a MIDI controller opened up the possibilities of the system. The controller could be plugged in to the Raspberry Pi at the heart of the system and use a synthesizer or sampler application to load up any conceivable sound. This would put a bit of extra stress on the Pi's processor, but the cheaper price and additional customization were much more appealing. Furthermore, MIDI controllers often have a selection of extra buttons, which can be used for customizable inputs to the computer it is plugged into. The MIDI controller the team chose was the M-Audio Keystation 88 II, which provided a full 88-key range, an affordable price, and a handful of extra buttons to use as controls.

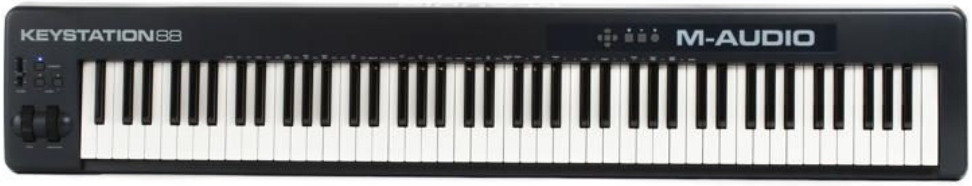


Figure 4.3: M-Audio Keystation 88 II MIDI Controller

The RasPiano needed a way to translate the MIDI information from the keyboard into audio to pass along to the filter chain. The team settled on SunVox, which is a popular, free, synthesizer program in Linux that runs excellently on the Raspberry Pi. SunVox provides many options with regards to either synthesizing our own instruments or using a sampler. The resulting audio is sent through the Raspberry Pi's DAC and out its 3.5mm port.

4.2.2 Microphone

The team wanted to include a microphone as an input device to take advantage of the musical instrument that (almost) every human is born with. The main problem in the microphone selection process was balancing price, performance, and ease of integration into the system. If the cheapest possible option was desired, it would be easy to find a mic with USB or 3.5mm phone jack output, but the audio quality would be underwhelming. As the microphones get more expensive, the quality increases, but connectivity is sacrificed. Higher price & quality microphones usually have XLR output, which would impose extra design challenges in converting the microphone signal into something that is usable in the system. After searching through dozens of microphones, the Samson Q2U was chosen. It was a bit on the expensive side, but the sound quality is excellent and it offers three outputs: XLR, USB, and 3.5mm phone jack. In the system, the microphone is powered via USB, but only taking audio output from the 3.5mm port to lessen the load on the Raspberry Pi.

Figure 4.4: Samson Q2U Microphone

4.2.3 Auxiliary

The auxiliary input offers a few options for users. It allows people who aren't comfortable with playing piano or speaking into a microphone a way to input audio, and it also provides a way to easily pass a full-spectrum audio signal into the system. The only part necessary is a 3.5mm audio jack, which is easily available for a few cents on sites like Digi-Key. Something important to note here is that the audio sent into the port will be in stereo, but the RasPiano only operates in mono, so one of the channels will not be connected.

Figure 4.5: 3.5mm Audio Jack

4.2.4 Summing Circuit

After deciding on the various inputs, the signals must be combined for use in the RasPiano. This is best accomplished with a summing circuit. It is closely related to the inverting amplifier, and uses one op amp and a handful of resistors. Each input signal is sent through a resistor to the inverting input of the op amp. The non-inverting input is connected to ground, creating a virtual ground at the inverting input. Then, a feedback resistor is placed between the output and the inverting input. The current through each input resistor is sent through the feedback resistor, which creates a voltage drop equal to the negative of the sum of the three input voltages. The weighting of the voltages can be changed by varying the resistance values, but the voltages are simply added directly for this application, so 10kΩ resistors are used everywhere in this circuit.

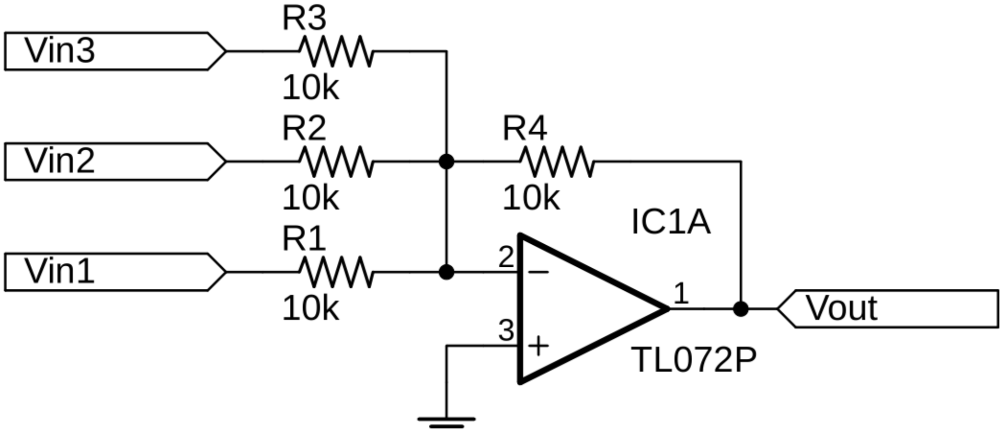


Figure 4.6: 3-Input Summing Circuit

4.3 Filter Chain

The filter chain is the heart of the RasPiano, and where the system evolves from “musical toy’ into ‘teaching tool.’ The basic concept is a series of filters that users can tweak to see how an audio signal is processed using analog electronics. During the brainstorming stage, the team was looking to include the four basic filter types of low-pass, high-pass, band-pass, and band-stop, as well as at least one additional effect. Complex effects like delay and reverb were lofty goals, so the team settled on a simple clipping distortion effect to round out the filter chain with a nonlinear circuit. Each filter can be independently toggled on or off and independently tuned - both by cutoff frequency and quality - so users can hear how the various circuits interact with each other in an audio context. Furthermore, the transfer function of the filter chain is displayed in real-time along with the FFT of the signal on the screen of the RasPiano.

4.3.1 Filters

The first challenge the team faced was figuring out which filter circuit to use. Being that filters have such a long history in the field of electronics, it wasn’t hard to find resources with extensive lists of filter circuits, their characteristics, and how to design them. The two resources that proved the most helpful were the Analog Devices Basic Linear Design handbook [1] and the Elliott Sound Products web page [2]. These two sources made it easy to see what advantages and disadvantages each filter circuit had, and trivialized the design process.

The starting point for this journey was the basic single-pole, passive RC low-pass filter. However, this circuit didn’t offer the tunability necessary, so the team moved on to the two-pole RLC filter. While the RLC filter, as a resonant circuit, has an adjustable quality (Q) factor, it can only be changed by changing the capacitor and/or inductor. Potentiometers are ubiquitous, but tunable capacitors and inductors are very scarce (and often expensive). For this reason, the team made the leap to active filters.

This was the team’s first exposure to active filters, so the previously mentioned resources started coming in handy here. Initial research went into investigating the more simple filter circuits, such as Sallen-Key and Multiple Feedback. What the team soon found was that these circuits were only tunable by cutoff frequency - the Q value was determined by the capacitors, so the problem with two-pole passive filter designs came up again. After this, the team looked into some of the larger and more complex circuits, and found a winner: the state variable filter.

The state variable filter hits all of the requirements for the filters. The cutoff frequency and Q value are independently adjustable by changing only resistance values, meaning there isn’t any search for tunable capacitors or inductors. The state variable filter also provides simultaneous low-pass, high-pass, and band-pass outputs, with an additional band-stop output available by summing the low-pass and high-pass outputs. All of the filter requirements were knocked out by a single circuit - aside from one limitation.

The last issue to address with the filters was the transfer function drawing. It was necessary to get the information of the potentiometer setting to the Raspberry Pi so it could calculate the transfer functions. Originally, the team was focused on using purely analog circuits, and the Sallen-Key needs two resistances to be tuned in unison to adjust the cutoff frequency, so dual-gang potentiometers were necessary for this to work correctly. For the extra information to send to the Pi, an extra gang was required, and 3+ gang pots are hard to find and expensive. Because of this, the team opted to switch over to digitally-controlled potentiometers.

The Elliott Sound Products page provides the necessary design equations to decide component values. Inside the state variable circuit are two integrators, whose components must be equal. The resistor and capacitor control the cutoff frequency of the filter, as in the passive single-pole RC circuit:

$$f_0 = \frac{1}{2\pi RC}$$

For a 100kΩ potentiometer, the maximum resistance corresponds to the lowest cutoff frequency. For the full audio spectrum of 20Hz–20kHz, this would mean an 80nF capacitor. Therefore, the maximum cutoff frequency for the same capacitor value occurs when the potentiomer is set to 100Ω. It is important that the potentiometer does not fall below this resistance as it can lead to instability in the filter. For an implementation of the circuit using hardware potentiometers, this would mean adding a 100Ω resistor in series, but for the digital potentiometers, this minimum value can be set in the code.

The Q value of the filter is set by the ratio of two resistors, labelled as R3 and R8 in Figure 4.7:

$$Q = \frac{\frac{R3}{R8} + 1}{3}$$

By using another 100kΩ potentiometer for R3 and a 10kΩ resistor for R8, a Q range of $\frac{1}{3}$ to almost 4 can be achieved. The other three resistors in the main State Variable layout are set equal to 10kΩ, and the three resistors in the summing amplifier for the band stop output are set to the same output.

4.3.2 Clipping Distortion

The idea for a clipping distortion effect was inspired by the classic distorted guitar tone. By including a nonlinear effect like this, the RasPiano can give visitors an idea of some of the creative ways engineers use analog circuits to achieve unique results in signal processing.

The initial design included an adjustable, non-inverting amplifier feeding into a pair of diodes facing opposite directions connecting to ground. The effect of this circuit was that as the gain of the amplifier was increased, more of the signal was lost to the rectification of the diodes. While this does introduce distortion to the signal by clipping it, the effect of the distortion is obfuscated by the amplification of the signal. The team decided to look for a more direct solution that only changes the signal by introducing clipping.

The circuit that the team settled on was inspired by a circuit used in Lab 1 of Microelectronics I. In this circuit, a voltage source separates the diode from ground. Changing this voltage source changes the clipping ceiling. To make the design more realistic for implementation, a voltage source is simulated by connecting the outer terminals of a potentiometer to a power rail and to ground. The center terminal, connected to the wiper of the potentiometer, is then connected to a voltage buffer which in turn connects to the other end of the diode. This circuit uses the 1N4148 diode, which has a voltage drop of about 0.7V. The phone-level audio signals in the RasPiano have a peak voltage of about 0.8V, meaning that without amplification, barely any of the audio signal will be clipped at the highest clipping amount. This is solved by adding an inverting amplifier to the input, with a complementary inverting amplifier at the output. The output amplifier has the same resistor values as the input amplifier, but they are switched, so the output amplifier acts as an attenuator to restore the signal to the same level as it was before the input amplifier.

4.3.3 Final Filter Chain Design

The final filter circuit uses the State Variable topography, with all non-tuning resistors equal to 10kΩ. Both capacitors are 80 nF, allowing the filter to be swept across the full 20Hz - 20kHz audio frequency range. TL074 op amps were chosen for their low noise performance and high input impedance. In the circuit schematic below, potentiometers are used to represent the 100kΩ MCP4151/MCP4251 digital potentiometers. The four outputs are sent to a small selection of jumpers, so that the output for each individual board can be selected. Bypassing is handled with the 74HC4053 analog switch IC.

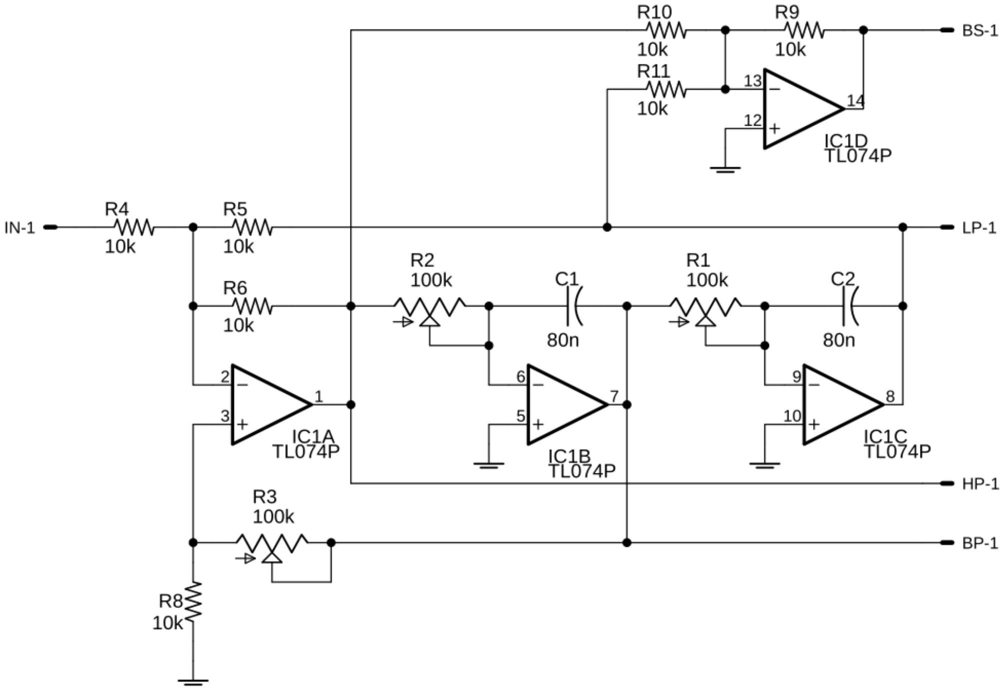


Figure 4.7: State Variable Filter Circuit

The clipping circuit below uses the same MCP4251 digital potentiometer as the filter circuit above. The input and output amplifiers are matched with equal, switched resistors. A voltage divider across the rails in the voltage source ensures that the voltage coming out of the pot wipers does not exceed the peak of the amplified audio signal. This lets us use the full range of the potentiometers. Again, bypass switching is handled with the 74HC4053.

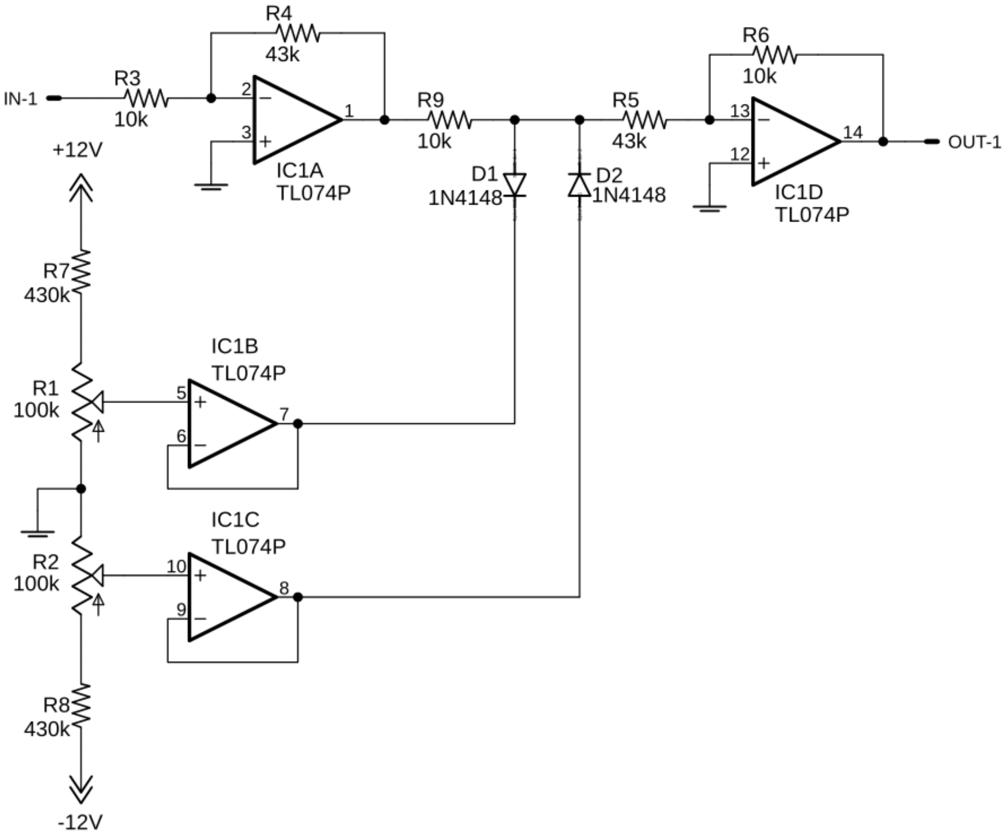


Figure 4.8: Clipping Distortion Circuit

4.4 Outputs

This section covers all of the various outputs of the RasPiano. First, the Raspberry Pi is explored, and all of its outputs are explained in detail. Following this are the visual and audio outputs.

4.4.1 Raspberry Pi

The Raspberry Pi is the core component of the Raspiano tasked with interfacing all i/o and performing any required processing for the system outputs. In this section we will cover the requirements needed to implement the features of a oscilloscope, spectrum analyzer, and transfer function display. We will then cover the final software design and the decisions made to complete our first functioning prototype.

4.4.1.1 Oscilloscope

Provided with the BitScope Micro DSO is a software suite that supplies varying forms of signal analysis tools such as oscilloscopes, spectrum analyzers and decoders. A requirement of the RasPiano is to have adjustable oscilloscope controls that alter time/division and voltage/division (to name a few). Unfortunately the BitScope Micro DSO software provided with the BitScope Micro does not provide the ability to use external controls. As such, how we display signal information and provide control must be created custom to the RasPiano.

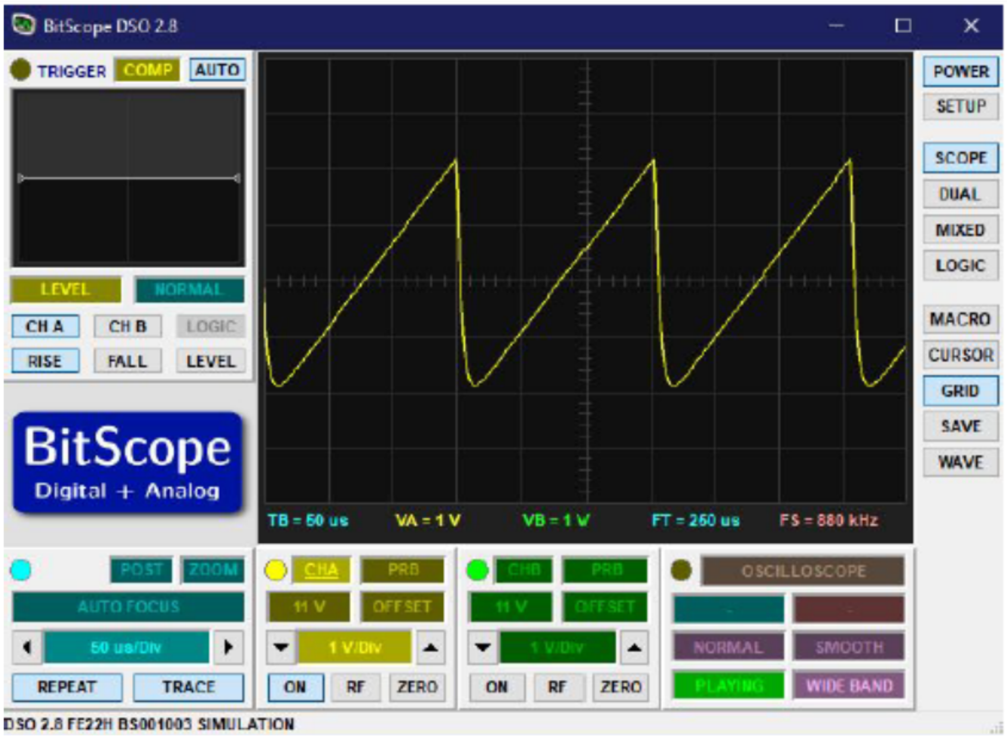


Figure 4.9: BitScope Micro Oscilloscope Application

To properly match our technical requirements of being able to adjust the oscilloscope display the application must have the following specifications:

- Provide reasonably accurate waveform traces
- The ability to change voltage scale
- The ability to change time scale
- The ability to change edge triggering
- Ability to interact with external controls such as buttons

Fortunately the BitScope Micro DSO also comes with a programming library named the BitScope Library API. This library provides numerous functions for interface configuration between the Raspberry Pi and the BitScope Micro, and functions to collect and process data.

One such function is BL_Acquire() which stores a new stream of signal data into a set buffer. Our application will need to process this data and appropriately form a trace between the values. It must also scale the trace to appropriately match the currently set voltage and timing scales. As will be discussed further below, this data will also need undergo a fast-fourier transform to construct a spectrum analysis display. As such, data collected from the BitScope will need to be passed to the spectrum analyzer software in addition to the oscilloscope software.

To provide a general understanding of how the oscilloscope application is designed a program flow chart is provided below. Note that the "DrawWave" process persists after initialization in the flowchart as the Bitscope data will need to be traced in real-time. Additionally as the oscilloscope application will be the main interface by which signal data is collected, whenever controls are pressed to modify the voltage or time scales the application will need to reconfigure how data is collected from the BitScope and began tracing using the new configuration.

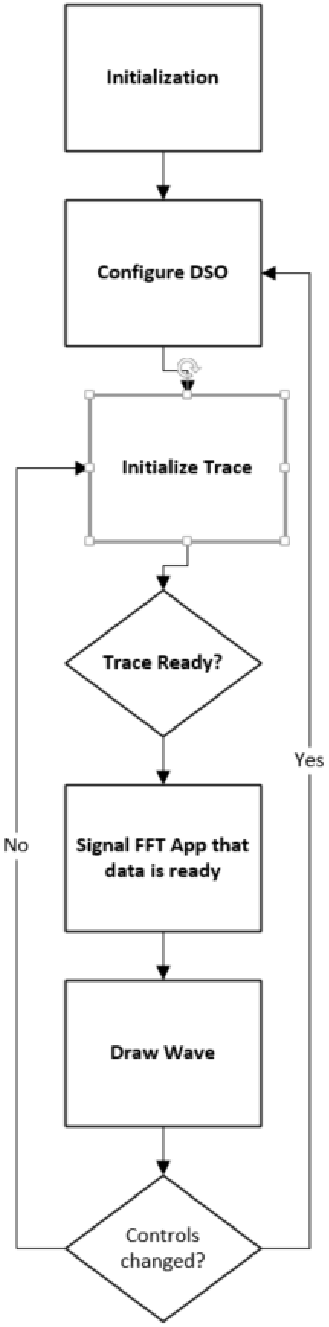


Figure 4.10: Oscilloscope Application Flow Chart

4.4.1.2 Spectrum Analyzer and Transfer Function

Similar to the oscilloscope this pair of displays must utilize the signal produced after the Raspiano's filter chain to properly visualize their effect on all audio passing through the system. As the only signals being processed are audio, these two displays should cover the frequency spectrum from 20Hz to 20kHz.

Roughly the display for these two components should appear similar to that below.

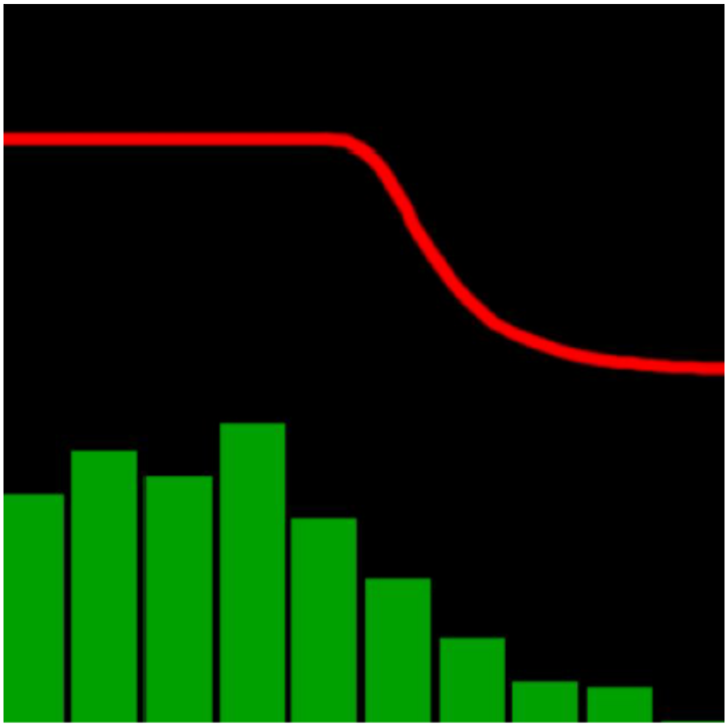


Figure 4.11: Rough Sketch of FFT/Transfer Function Application

We would display the spectrum activity of the audio and just above it display the filter's transfer function. This way the filtering effects on the audio are more apparent.

Again, to implement the FFT display we will need to collect the audio waveform data passed through the filters. Once the data is collected from the oscilloscope application it will need to be processed into "frequency buckets" for visualizing the FFT. As such the FFT application must meet the following requirements:

- Collect waveform data from the oscilloscope application
- Perform an FFT over the collected data
- Organize the FFT results into multiple frequency buckets
- Draw bars whose magnitude is reflective of the amount of signals falling into each frequency bucket

After the bars are drawn the FFT portion of this application will be complete. Simultaneously the transfer function will need to be constructed as described in the transfer function discussion below.

The FFT and transfer function application will need to have two distinct functionalities. Firstly, after appropriately configuring and establishing a connection to the BitScope Micro one process will process the oscilloscope's data to generate the fft plot. Simultaneously, after configuring GPIO outputs on the Raspberry Pi another process will need to check the currently assigned levels for the transfer functions of each filter, update the potentiometers if necessary, and just the resulting transfer function on the display. A flow chart of the process is shown below.

Many open source programing APIs are widely available to perform functions such as the FFTs quickly, and accurately. While we could certainly attempt creating our own FFT implementation it will be most time effective for us to use tested software. In addition, decisions must be made as to what language will be used to run the FFT/Transfer Function application.

To handle processed waveform data and display the graphical user interface (gui), we will utilize an object-oriented programming language known as Python as they provide the appropriate abstractions to quickly prototype the required guis.

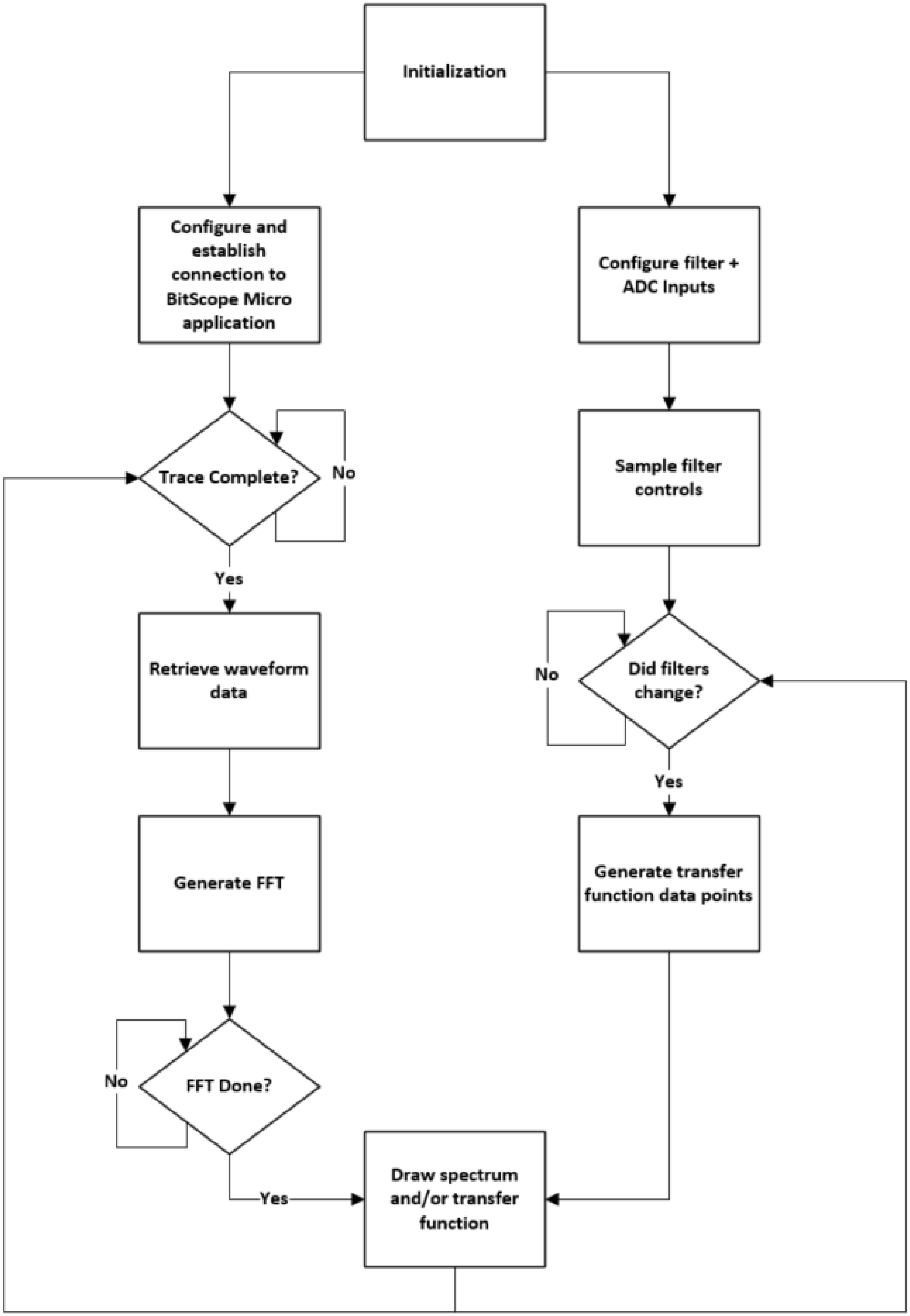

Initialization
Configure and establish connection to BitScope Micro application
Configure filter + ADC Inputs
Trace Complete?
No
Yes
Sample filter controls
Did filters change?
No
Yes
Retrieve waveform data
Generate FFT
Generate transfer function data points
FFT Done?
No
Yes
Draw spectrum and/or transfer function

Figure 4.12: FFT/Transfer Function Application Flowchart

To reduce the amount of simultaneously running processes and optimize data flow from the BitScope to the oscilloscope, fft, and transfer function application we will create a joint back-end application for the three displays. That way waveform data can be simultaneously available to both the oscilloscope and fft.

For the transfer function display we shouldn't need any particular libraries as we can create a set of data points with the filter transfer function. However as mentioned previously it would be effective to use an API for performing the FFT itself. As such we will be utilizing a combination of the Scipy and matplotlib Python libraries. A summarized description for these two libraries is below:

- Scipy - A scientific data processing library that provides several FFT implementations
- Matplotlib - A mathematical graphing library which can be used to visualize data vectors similar to matlab

4.4.2 Monitor and Splitter

Two critical output devices are the monitor panel and hdmi splitter. Per our technical requirements we would like for the Raspiano display to be capable of being replicated onto other panels.

Figure 4.13: Powered HDMI-Splitter

This splitter is a generic 1x2 HDMI splitter which accepts a single video input and provides two duplicate video outputs. With the various iterations on the HDMI video standard we specifically chose to use a powered HDMI splitter for the following reasons:

- Better device compatibility for displays which may consume more power from HDMI
- Improve signal distance for displays connected at a distance from the Raspiano

In choosing a monitor for the Raspiano we wished to meet the following requirements:

- 24'-27' display size - for improved visibility of Raspiano graphics
- VESA compatible - for secure mounting to a Raspiano housing

Initially chose a Sceptre E248W-19203R. A 24 inch HDMI monitor available on Amazon.com for 98.19 dollars.

Figure 4.14: Sceptre E248W-19203R

The monitor supports VESA mounting, has a 24' screen size, and displays with a resolution of 1920x1080. Due to the few monitor requirements we have, any hdmi monitor should service well for the Raspiano. This Sceptre monitor was chosen due to it meeting our requirements and being the cheapest of all monitors considered. Please see appendix xHfor a table of the monitors considered.

Due to the remaining budget at the time of writing this section (approximately 100 dollars) we decided to use a 24 inch Dell monitor available in the ECE Analog lab.

Figure 4.15: Dell 24 Inch HDMI Monitor

4.4.3 Speaker

One of the sound outputs for the RasPiano is a pair of speakers. These will be set to a low volume, adjustable only by opening the Raspiano, so that the sound does not become a distraction for students studying in the lounge. There were a few directions we could have taken when choosing how to publicly output the sound. Originally, we considered purchasing bare speaker drivers and integrating them into the piano enclosure itself. However, we decided that this would waste time better spent elsewhere and would not save enough money over purchasing a pair of bookshelf speakers. After searching the web, we found a few sets of speakers for $20-30. Of these speakers, the best choice was a pair of Monoprice MP-65TW 6.5' speakers shown in the figure below. This pair offered the widest frequency response at a competitive price.

Figure 4.16: Monoprice MP-65TW Speakers

An amplifier will connect the output signal of the filter chain to the speakers. Our reach goal was to build this amplifier ourselves. To begin, we documented the specifications requires:

- Total Harmonic Distortion + Noise (THD + N) is a measure of just how much effect the amplifier has on the sound output.
  - Probably want this as low as possible so that amplifier does not interfere with the desired soundwave from the filters.
  - Low as possible.
- Bandwidth: The frequency range at which the amplifier can operate.
  - 20Hz - 20kHz
- Slew Rate: The maximum rate of change of output.
  - Microphones can be 5-50 mV, and normal audio is 0.3-2 V, so max rate would be 0.005 V to 2V with a frequency of 20 kHz
  - 20 kHz, 1.995 V
- Gain: Ratio between the magnitudes of input and output signals.
  - Speaker: 30 Wrms, 60 W, 6 Ohms
  - MAX Output: 13.4 V to 18.9 V
  - MAX Input: 2 V
  - Voltage Gain: 6.7, 16.5 dB to 9.45, 19.5 dB
- Stability: The ability to provide constant and reliable output.
  - Decently important
- Linearity: The degree of proportionality between input and output signals.
  - Assume we want max linearity to provide the user with output audio closest to the input.
  - Highest linearity possible
- Efficiency: Another very important characteristic, it is the ratio between the output power and power consumed.
  - Yes, would like to be as efficient as possible.

After obtaining a better idea of the specifications required for our system, we performed research on the different types of amplifiers. The class A amplifier had characteristics closest to own specification excluding power consumption. With efficiency as a concern, we decided class AB would be the best for our design due to its linearity and efficiency. This class closely aligns with class B, however the crossover distortion given from having to wait till 0.7 V for the BJTs to turn on is avoided using bias voltages. This is most commonly done with resistors or diodes.

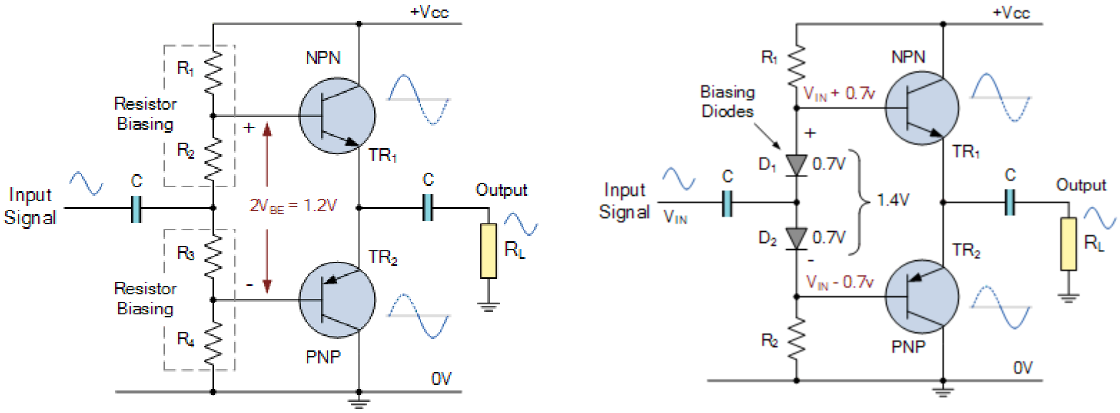


Figure 4.17: Common AB Amplifier Circuit Design

After selecting the type of amplifier to use our design, the next step was to begin designing the circuit and calculating values for the parts within the circuit.

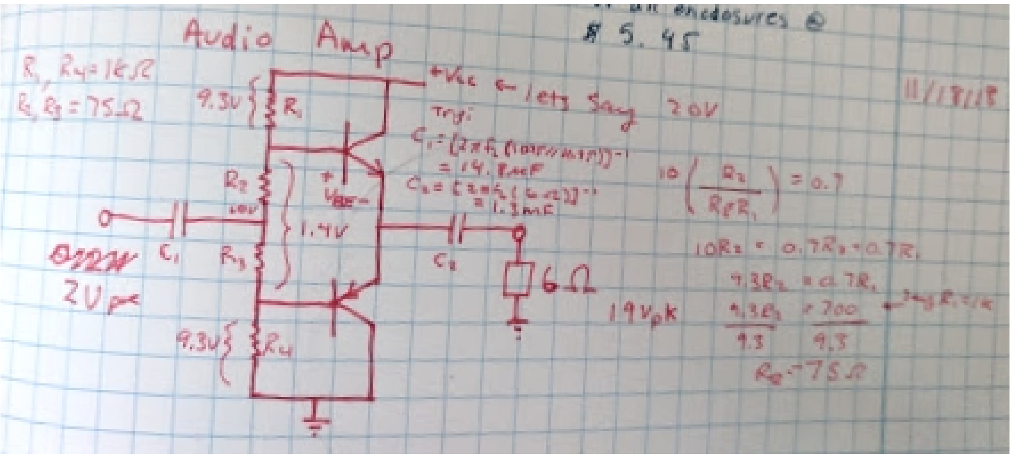
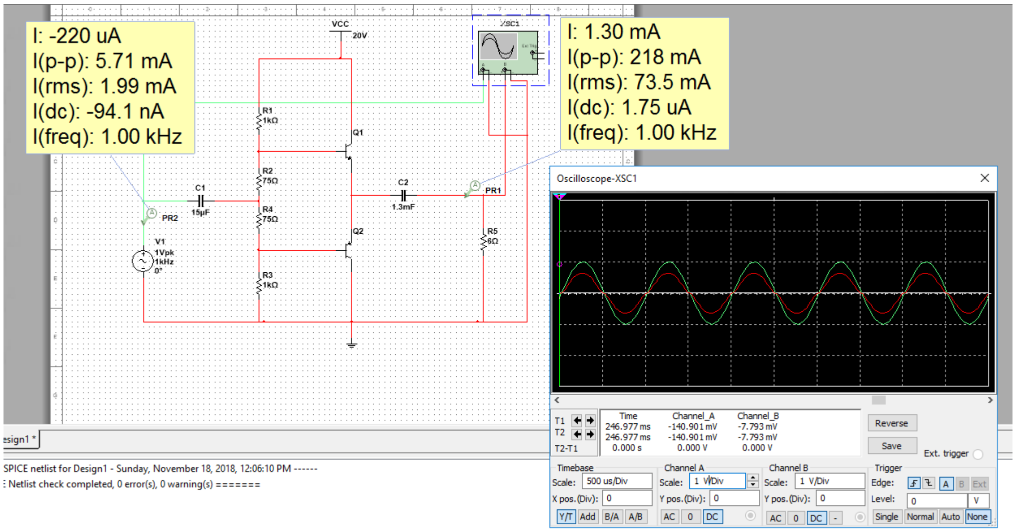


Figure 4.18: Audio Amp Circuit Design Notes

Values were chosen in 4.18, and used for the simulation shown in 4.19.



I: -220 uA
I(p-p): 5.71 mA
I(rms): 1.99 mA
I(dc): -94.1 nA
I(freq): 1.00 kHz

I: 1.30 mA
I(p-p): 218 mA
I(rms): 73.5 mA
I(dc): 1.75 uA
I(freq): 1.00 kHz

```
SPICE netlist for Design1 - Sunday, November 18, 2018, 12:06:10 PM ------
: Netlist check completed, 0 error(s), 0 warning(s) =======
```

Oscilloscope-XSC1

	Time	Channel_A	Channel_B
T1	246.977 ms	-140.901 mV	-7.793 mV
T2	246.977 ms	-140.901 mV	-7.793 mV
T2-T1	0.000 s	0.000 V	0.000 V

Timebase Scale: 500 us/Div; X pos.(Div): 0
Channel A Scale: 1 V/Div; Y pos.(Div): 0
Channel B Scale: 1 V/Div; Y pos.(Div): 0
Trigger Level: 0 V

Figure 4.19: Audio Amp Circuit Simulation

The Class AB Amplifier amplifies the current at the output, however the voltage gain will be less than one. With the speaker specifications, the ideal max output current should be approximately 3.15 A. This issue was solved in a later rendition of the circuit, the improved design, adding an operational amplifier, had no voltage loss in the circuit. The operational amplifier forces the output with a feedback loop to equal the input voltage.

Moving forward with the project, the design of this self-built amplifier was put on hold. However, before putting this on hold, a pre-made audio amplifier was selected in case the project's deadline restricts progress on building this amplifier. This was because the goal of creating the audio amplifier for this system was low on the priority list on comparison to our various other tasks that required time and effort.

The part chosen for the back-up audio amplifier is a class T amplifier since sound quality is less of a concern with professionally made amplifiers, class T is more efficient, and there is a better selection of class T amplifiers matching our speaker impedance and power. It is a Pyle Home Mini Audio Amplifier, 60W Portable Dual Channel Surround Sound HiFi Stereo Receiver with 12V AC Adapter, AUX, MIC IN, Supports Smart Phone, iPhone, iPod, MP3 for 2-8 Ohm Speakers.

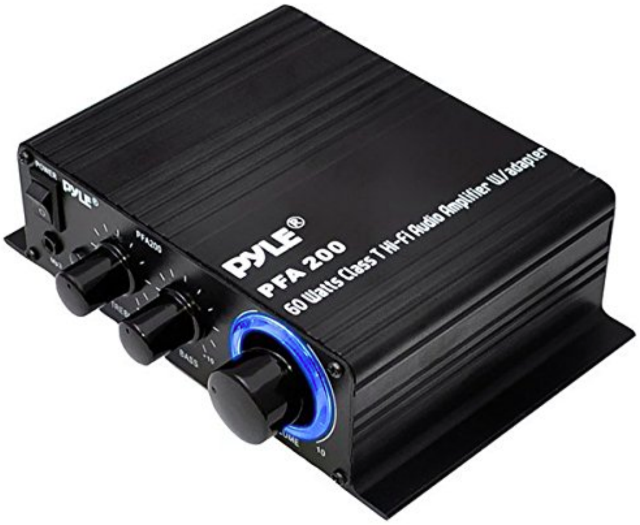


Figure 4.20: Audio Amplifier Pre-made Part

Upon reaching our fourth term of the MQP, it was decided that the pre-made audio amplifier would be implemented rather than continuing our progress on our amplifier design. Although the research completed for designing the audio amplifier was not used within our final project, there was still value in learning about amplifier circuit design. This can be reflected back to one of our original goals, to learn more about application of the knowledge obtained from our past courses. Thus making this experience still relevant and valuable to our project.

4.4.4 Color Organ

The color organ adds an extra layer of fun and charm to the RasPiano. The basic idea is to take the output audio signal from the system and split it into frequency bands, then have an array of LEDs that lights up according to the activity in each frequency band.

Initial designs for the color organ included a multiple feedback bandpass filter for each frequency band, followed by a full-bridge rectifier, an RC element, ending with an LED that is driven directly by the voltage from the RC element. This RC element is a capacitor and resistor in parallel connecting to ground. It operates such that when a peak in the signal occurs, the capacitor is charged directly and slowly discharges through the resistor, generating a decaying exponential voltage. The effect is that a rhythmic audio signal, such as a series of kick drums, would not just have very short flashes, but a decay as well. This aspect was inspired by Jameco Electronics' Color Organ [3], although most of our design was invented. Some elements of our system worked well, but there was a lot of improvement needed.

One of the first changes to the circuit was to replace the full bridge rectifier with just a single diode. Audio signals are mostly symmetrical, and most audio frequencies would be undetectable to the human eye, so not much is lost if only half of the waveform is used. The second issue was driving the LEDs. The voltage from the RC element was very low, and could barely light up a single LED. Instead, we looked for a way to amplify the voltage such that we could use the LED strips employed in the LED Stairs project. Our solution was to pass the output of the RC element into the analog input of an Arduino Nano, which would then drive the LED strips with BJTs, similar to in the LED Stairs. This also allows us to control the red, green, and blue channels of the LEDs independently, letting us implement animations and color transitions.

Finally, we wanted to address the problem of dynamic range. Small audio signals passed into the color organ would barely light up the LEDs, so we wanted a way to normalize the signal level. To do this, we used an automatic gain control (AGC) circuit from Microelectronics II, lab 6. We made a couple of adjustments - first, a voltage divider at the input to ensure the signal was not too hot going into the circuit, and increasing the capacitor to 1μF to increase the attack and release time. The output is normalized to a level of approximately 5V, which works perfectly, as the Arduino's maximum voltage for analog inputs is 5V as well.

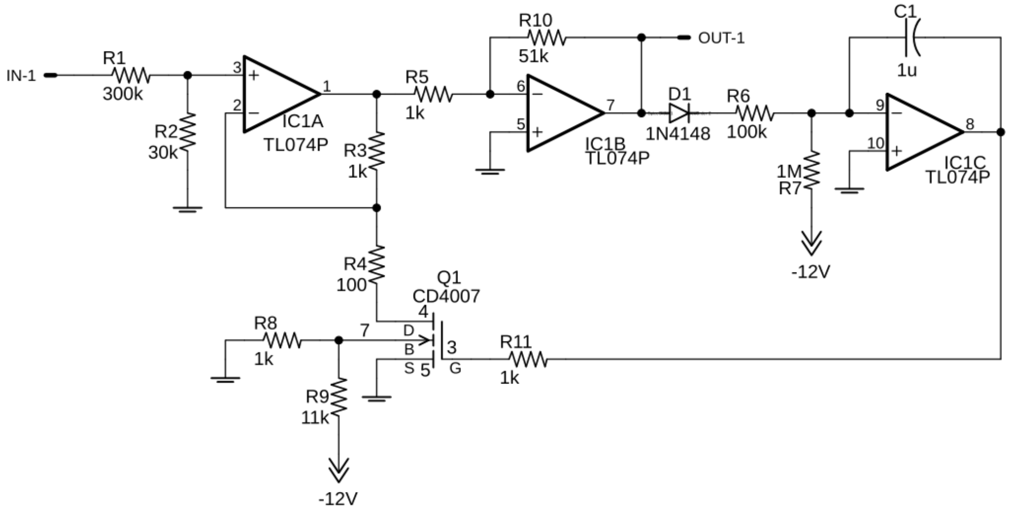


Figure 4.21: Color Organ Automatic Gain Control Circuit

R1 and R2 keep the input voltage from exceeding approximately .1V, which during testing, caused distortion in the output. C1 was bumped up from 0.1 to 1 μF to increase the time it takes for the gain to change. Q1 in the schematic is a discrete MOSFET, but in the real circuit we used a CD4007 NMOS array. The output leads to the five frequency bands.

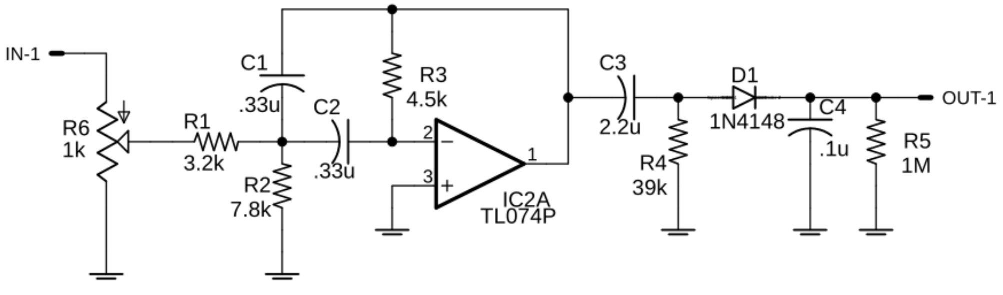


Figure 4.22: Color Organ Single Frequency Band

Each color band uses a 1kΩ potentiometer at the input to adjust the output brightness. Initial testing without this feature indicated that the color bands may have different brightness levels relative to each other, so with the potentiometer, they can be adjusted independently so they all have a similar brightness. R1-3 and C1-2 control the center frequency of the multiple feedback bandpass filter, and differ between each frequency band. C3 and R4 form a high-pass filter with a very low cutoff frequency to remove any stray DC bias. D1 rectifies the signal, allowing C4 and R5 to generate the control voltage for the LEDs as discussed earlier.

Chapter 5

Final Design

In this chapter, we will take a look at how the various subsystems of the RasPiano were brought together. This includes the various connections, power rails, and signal paths, as well as PCB design.

5.1 Connectivity of Subsystems

The most significant design challenge in regards to communication in the RasPiano was tackling the digital potentiometers in the filters. The MCP4151 and MCP4251 that are used in the filters use the SPI interface, which makes some things easy and some things a bit difficult.

SPI protocol is designed to be scalable across multiple slave devices, which is exactly what is going on in the RasPiano. Each filter has two digital potentiometers (plus one for the clipping distortion), and there are four filters, so there are a total of nine slave devices to control. SPI uses three common lines - a clock signal, master in slave out (MISO), and master out slave in (MOSI). In addition, a number of chip select (CS) lines are necessary - one for each slave device. In total, twelve lines are used purely for communication between the Raspberry Pi and all of the digital potentiometers.

SPI communication begins with the chip select line. To select a slave device to communicate with, its CS is pulled low. Then, the clock and MOSI lines come into play, with the latter handling the data itself and the former ensuring synchronization. The data transferred is relatively simple - first, an address is sent to indicate which wiper is being moved, followed by a number to indicate the value to set the wiper to. Additional commands exist to increment or decrement the wiper, as well as to read the wiper's current position, but they are not necessary for this application. The data transmission is handled by built-in functions in SPI libraries in both Python (used on the Raspberry Pi) as well as Arduino, making for easy prototyping.

However, one problem remains. The digital potentiometers do not allow any signals on any of their pins outside of the voltage range set by the power rails. The digital signals used to control these chips operate on 3.3V to 5V levels, referenced to ground. However, the audio signals flowing through the potentiometer terminals are centered at ground and vary between approximately +1V and -1V. Furthermore, the digital potentiometers have a relatively narrow supply range, so sharing the 12V rails provided for the op amps is out of the question. The chosen solution was to power the digital pots with +2.5V and -2.5V rails, and bias the digital signals to those levels as well.

The rails are created using another DCH01 DC/DC converter, with the same +/-12V output as the one used to power the op amps. However, these outputs are then passed to the LM317 and LM337 voltage regulators to drop the voltage down to the correct levels. The LM317 handles the positive voltage and the LM337 handles the negative. The regulators are adjustable, so a pair of resistors are used for each to set the output level. Some experimentation is necessary to generate the correct voltage due to component tolerances, but each resistor should be close to 1 k Omega . Capacitors are applied to filter both the input and output voltages, and a diode is also added to the network for protection against capacitor discharge.

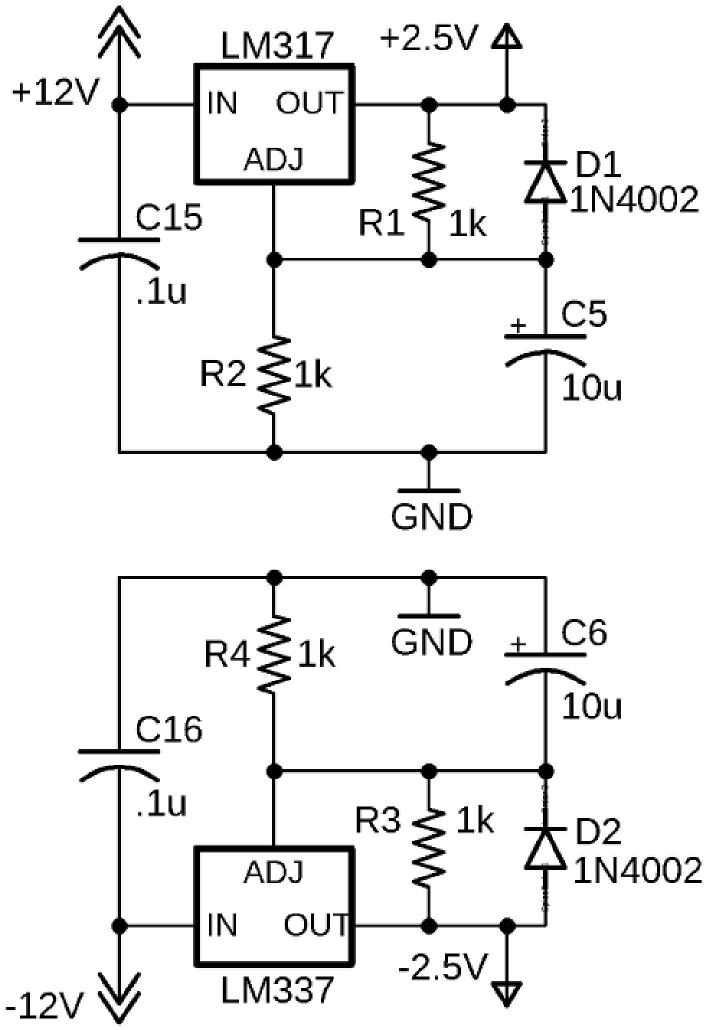


Figure 5.1: LM317 and LM337 Voltage Regulator Circuits

Finally, the digital signal biasing is accomplished via the 74HC4053 analog multiplexer/demultiplexer. This same chip was used as an analog switch in the filter circuits, but it finds a new use here. By attaching the +2.5V and -2.5V rails to the X and Y inputs for a channel and the digital signal to the channel select, the output for the channel is the digital signal adjusted to the new power rails. Each IC has three channels, so four chips are necessary to convert all 12 signals.

Finally, the signals and power rails are sent to the filter boards with 8-conductor cables. The only signal that doesn't need to be converted is the bypass switch signal for each filter. This signal is sent to each filter and controls another 74HC4053 that swaps the output between the input signal and the filtered signal. On every cable, the signals are sent through each position as such:

1. Chip select for digital potentiometer controlling cutoff frequency
2. Chip select for digital potentiometer controlling Q
   - Note that this pin is tied to (1) for the clipping distortion board, as there is only one potentiometer to control.
3. Serial clock
4. Master Out Slave In
5. Master In Slave Out
6. Bypass Switch
7. -2.5V Rail
8. +2.5V Rail

5.2 PCB Design

PCBs were a necessity for the RasPiano as there weren't enough breadboards for all of the circuits, and soldering everything on perfboards would be a bit too tedious. For schematic capture and board layout, EAGLE was the software of choice, as it is somewhat of an industry standard and is free for students. This was the first exposure to PCB design for everyone in the team, so a lot of trial and error was involved. These were a few of the goals for all of the PCB layouts in the RasPiano:

- Small board area
- Consideration of parasitics
  - For example, keep high-impedance nodes such as op-amp inputs short
- Logical and aesthetically pleasing layout
- Bypass capacitors for power rails and IC power inputs

The final PCB designs are shown below:

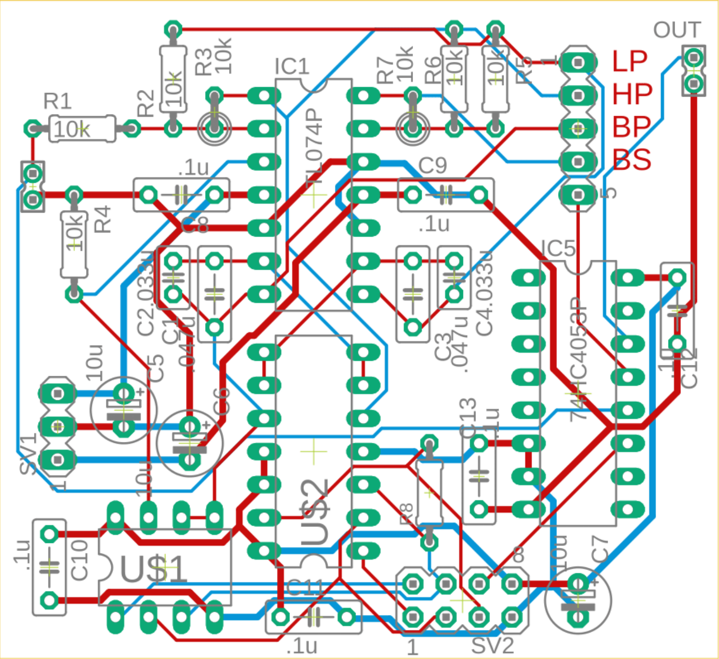


Figure 5.2: State Variable Filter PCB

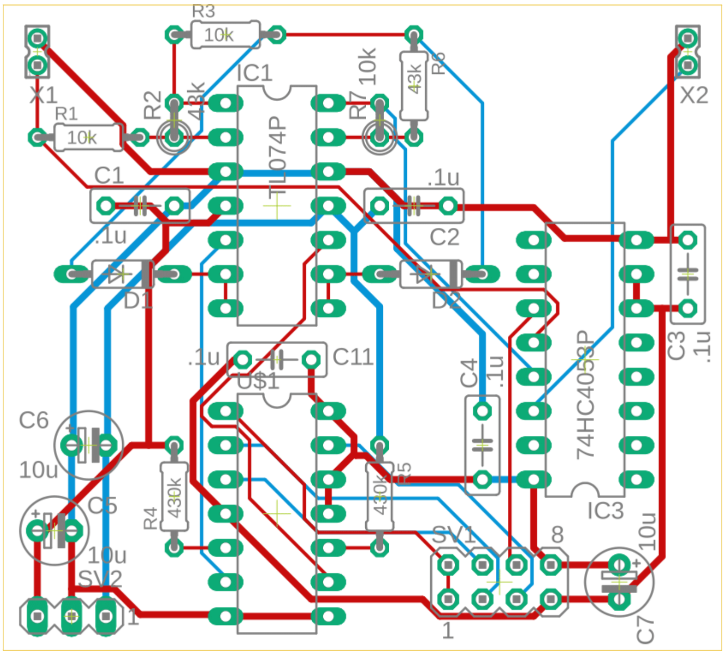


Figure 5.3: Clipping Distortion Effect PCB

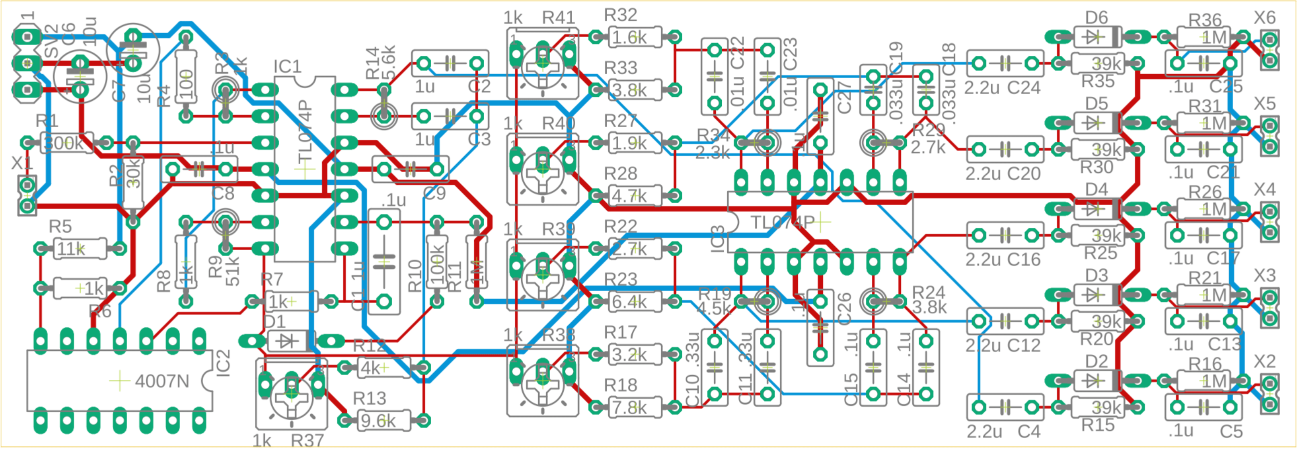


Figure 5.4: Color Organ PCB

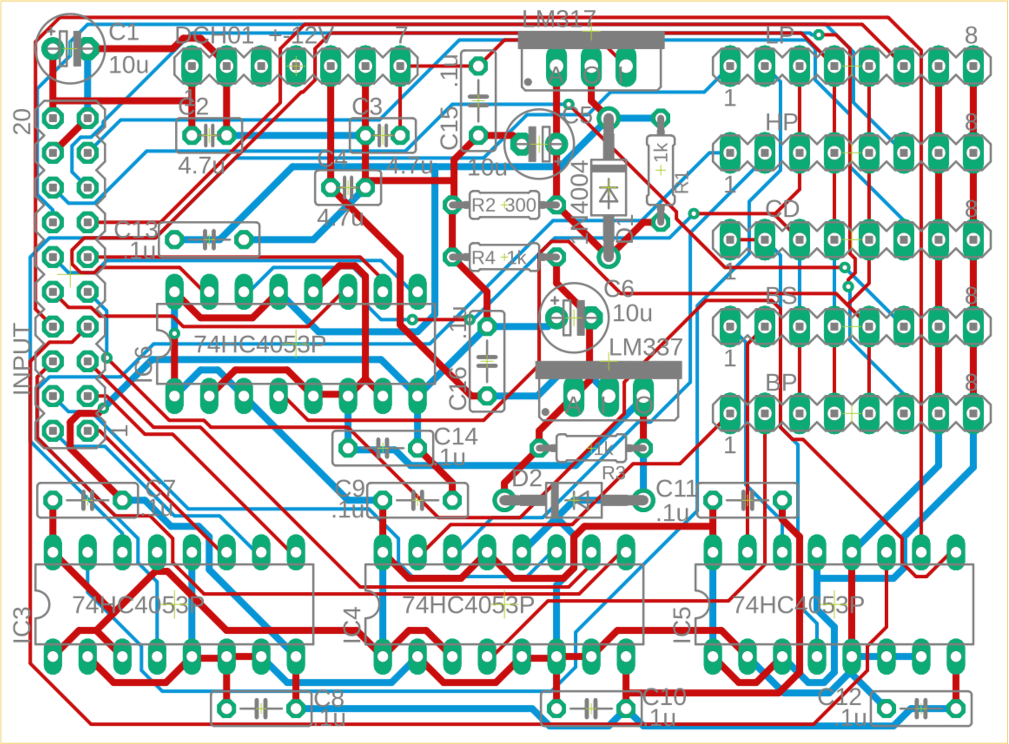


Figure 5.5: Potentiometer Level Translation PCB

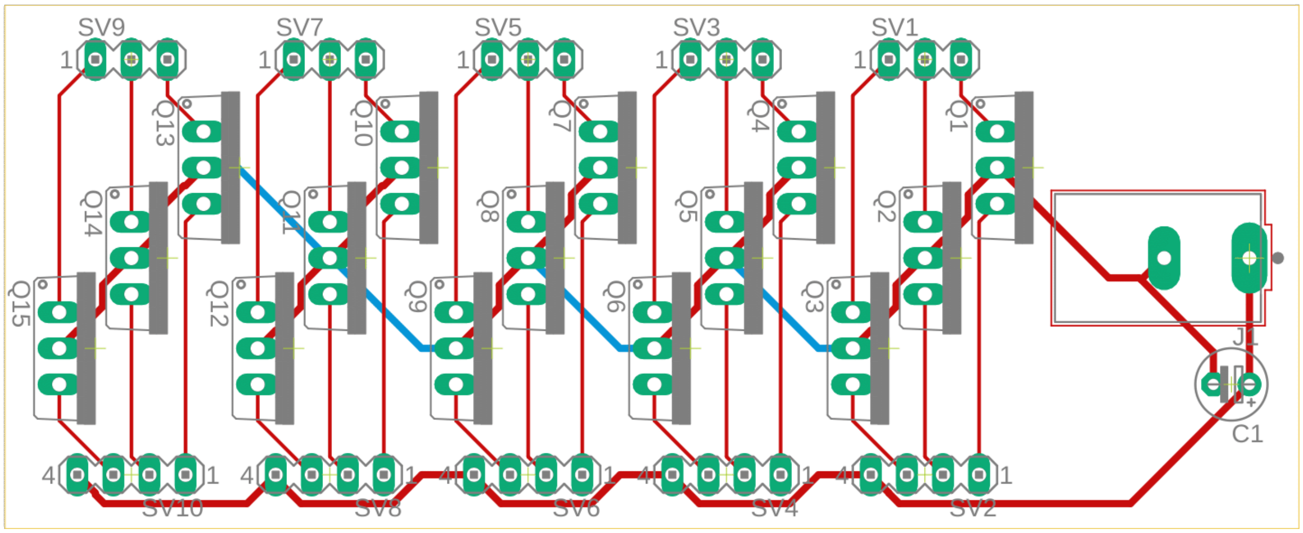


Figure 5.6: Color Organ LED Driver PCB

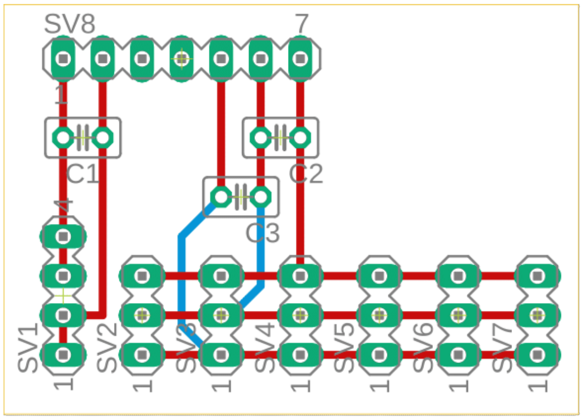


Figure 5.7: 5V to +/-12V DC-DC Conversion PCB using DCH01

After several revisions, the PCBs were ordered. By the time the team was ready to order, it was a few weeks into D term with not much budget left. It was imperative that the fabrication service chosen had a quick turnaround and shipping time with low cost. The team did some research on various manufacturers and settled on JLCPCB after some recommendations by another MQP team. JLC offers extremely competitive pricing, quick turnaround times, and relatively cheap and quick shipping from China. At the time of ordering, JLC did not charge extra for different color solder masks, so the team was able to order a variety of colors. Shipping only took about one week.

5.3 Acquiring RasPiano Software

In this guide we will cover how to acquire and install the software and Raspbian operating system required to create a software prototype for the RasPiano. Note that instructions concerning downloading and acquiring Raspbian OS, the Sunvox synthesizer, the BitScope Micro DSO application and running the above applications are performed on Windows 10. However, all steps are equivalent on the Raspberry Pi. Any steps that must be performed exclusively on the Raspberry Pi will be indicated explicitly.

5.3.1 Installing Raspbian OS on the Raspberry Pi

For the RasPiano we currently use a graphical user-interface (gui) version of Raspbian OS. A Linux distribution made specifically for operation on any/all Raspberry Pi hardware. In general the steps for installing Raspbian OS onto a Raspberry Pi are as follows:

1. Download the latest disc image available of Raspbian (at the time of this writing Raspbian Stretch)
2. Insert a microSD card into a microSD to full size SD card adapter
3. Insert or connect the microSD card and adapter onto a computer
4. Write the Raspbian disc image to the microSD card
5. Insert the microSD card into a Raspberry Pi and power the device

As the specific steps and applications vary from Windows, Mac OS, and Linux, we reserve detailed instructions on how to install Raspbian for instructions found on the official Raspbian website: `https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up` Once the installed onto a microSD card, simply insert the card into a Raspberry Pi and power-on the device. Raspbian will automatically boot as shown below.

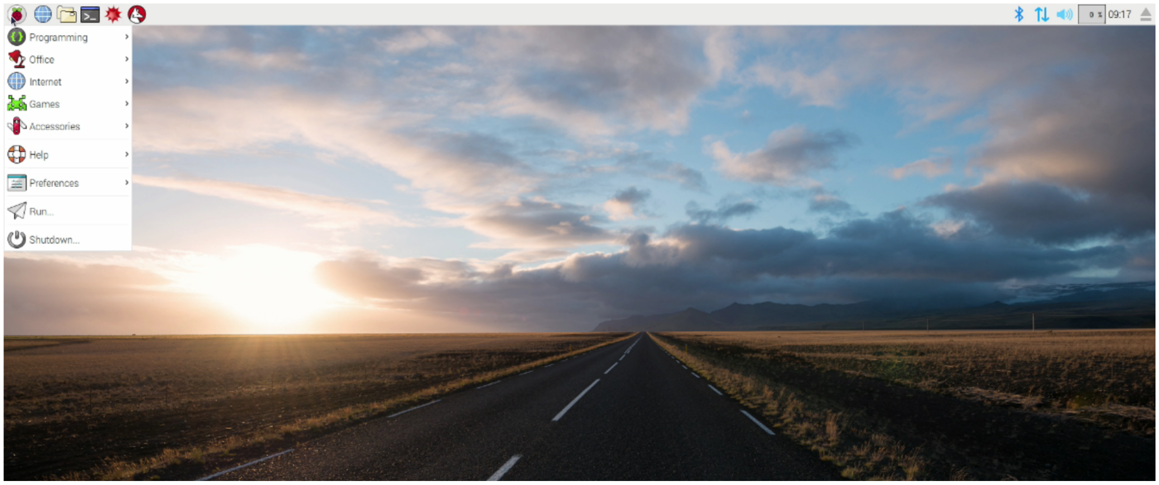
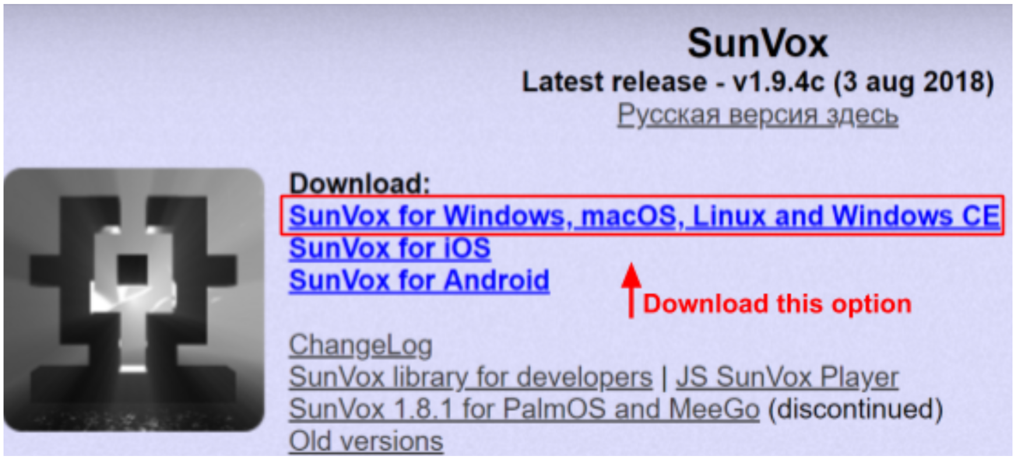


Figure 5.8: Raspbian OS Desktop

5.3.2 Acquiring and Configuring the Sunvox Synthesizer

To obtain Sunvox visit its official website at WarmPlace.ru here: `http://www.warmplace.ru/soft/sunvox/` On this page you will find download links for Android, IOS, and a zip which contains copies of the application for Windows, Mac, Linux, and Windows CE devices as shown below. Select and download "SunVox for Windows, macOS, Linux and Windows CE"



SunVox

Latest release - v1.9.4c (3 aug 2018)

Русская версия здесь

Download:

SunVox for Windows, macOS, Linux and Windows CE

SunVox for iOS

SunVox for Android

Download this option

ChangeLog

SunVox library for developers | JS SunVox Player

SunVox 1.8.1 for PalmOS and MeeGo (discontinued)

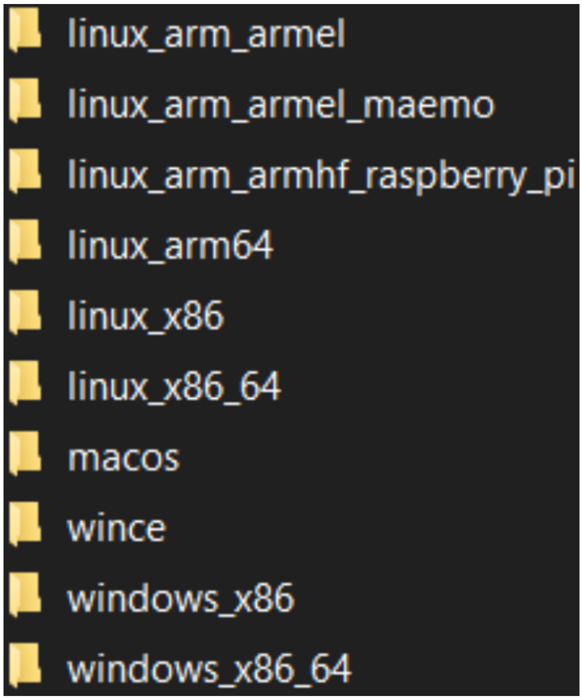
Old versions

Figure 5.9: Sunvox Download Link

Once downloaded extract the zip file and open it. Within the folder should be another labeled ?sunvox?, open this. Within this sunvox folder will be an array of folders and yet another labeled ?sunvox?, open this folder. Note that the directory reached so far will be similar to:

```
.../sunvox-x.x.xx/sunvox/sunvox
```

Within this directory should be a folder structure similar to that below.



linux_arm_armel
linux_arm_armel_maemo
linux_arm_armhf_raspberry_pi
linux_arm64
linux_x86
linux_x86_64
macos
wince
windows_x86
windows_x86_64

Figure 5.10: Sunvox Application Folder Structure

Listed here are multiple directories containing Sunvox executables for different kinds of systems. For the Raspberry Pi open "`linux_arm_armhf_raspberry_pi`". This directory will contain two executables. One named `sunvox` and another named `sunvox_lofi`. Either is fine for now. A discussion concerning their difference and the importance of that can be found in the section "Additional Notes and Troubleshooting" further below.

Once the program is executed it should look similar to that below. Note that the application theme was changed for visibility and will likely open to its dark theme by default.

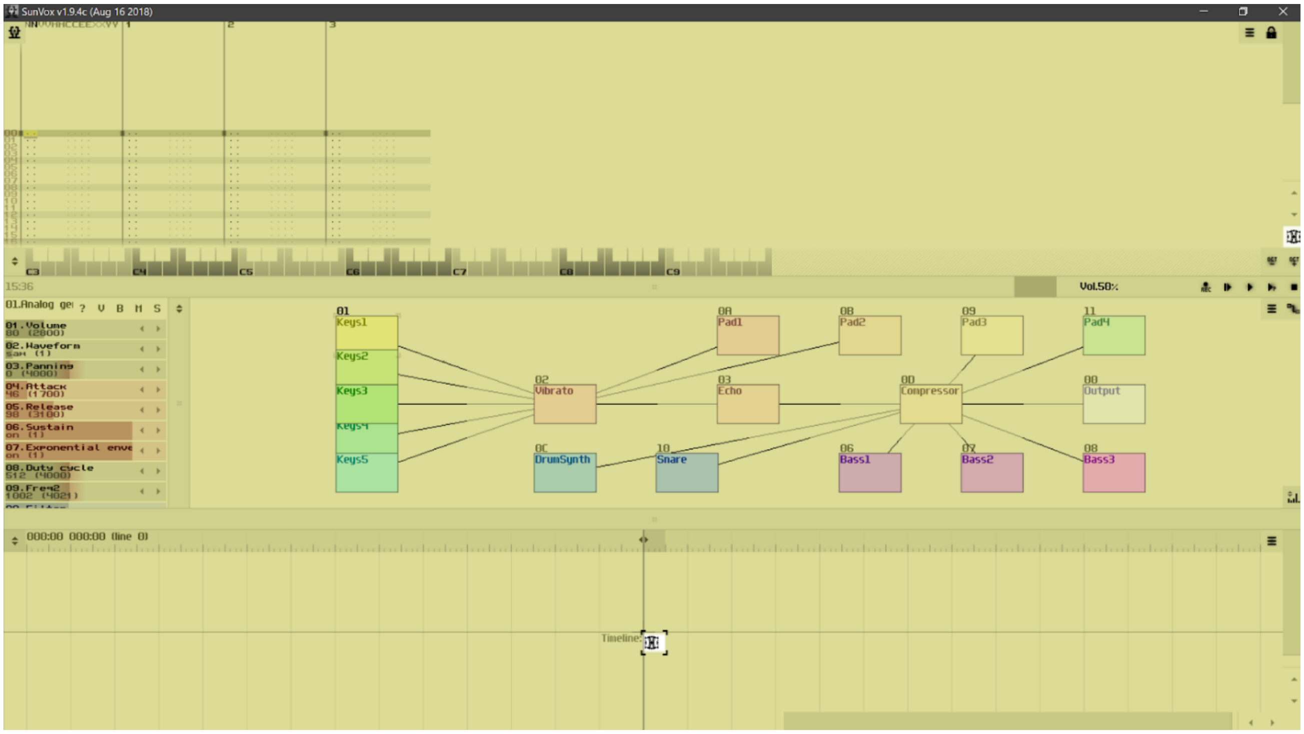
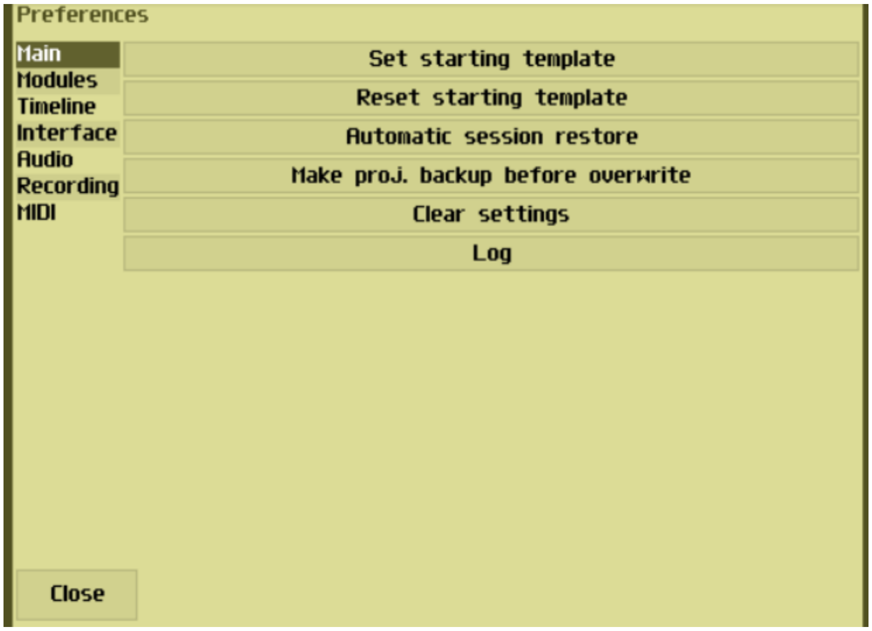


Figure 5.11: Sunvox Application

At this point the synthesizer instruments may be configured or modified as desired. To allow input from the MIDI keyboard select the crosshair button located at the top-left of the application window and select the "Preferences" option in the resulting drop-down menu. A window similar to that below should appear. Along the left column of it select "MIDI."



Preferences

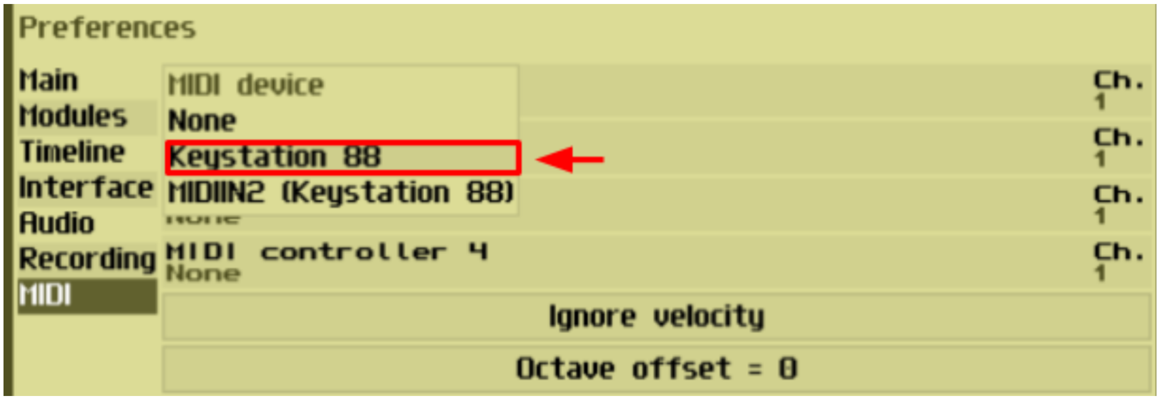
Main
Modules
Timeline
Interface
Audio
Recording
MIDI

Set starting template
Reset starting template
Automatic session restore
Make proj. backup before overwrite
Clear settings
Log

Close

Figure 5.12: Sunvox Preferences Menu

Making sure the MIDI keyboard is connecting to the Raspberry Pi through USB, select "MIDI controller 1." In the following dropdown select "Keystation 88." Note that two options may appear as shown in the figure below. The second option corresponds to the extra keys located at the top bezel of the Piano. To control the synthesizer with the piano keys the first options must be chosen.



Preferences

Main
Modules
Timeline
Interface
Audio
Recording
MIDI

MIDI device
None
Keystation 88
MIDIIN2 (Keystation 88)

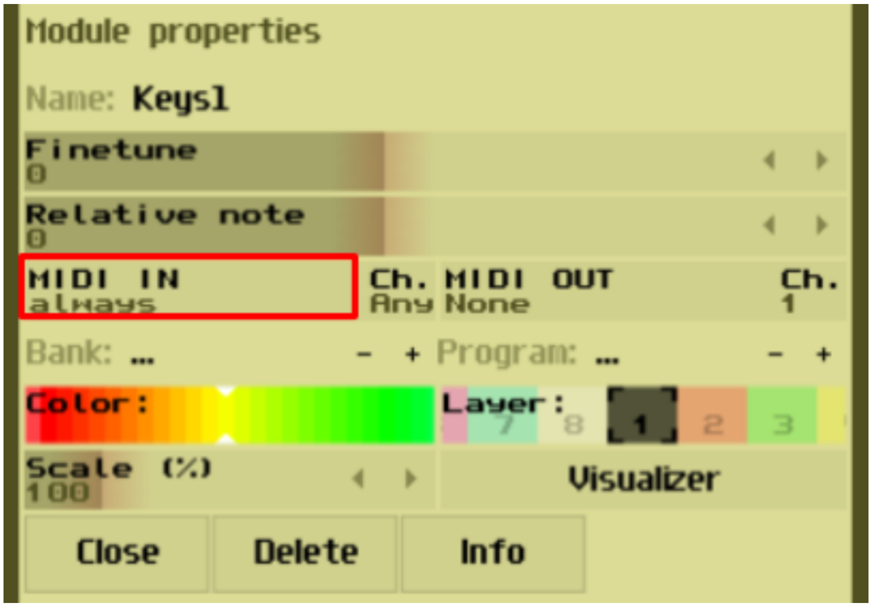
MIDI controller 4
None

Ch. 1
Ch. 1
Ch. 1
Ch. 1

Ignore velocity
Octave offset = 0

Figure 5.13: Sunvox MIDI Select

To modify which synthesiser elements are controlled by the MIDI keyboard right-click on the desired element and select "Module Properties" in the resulting dropdown. A menu similar to that below should appear. Left-click the "MIDI IN" button until it has a value of "always."



Module properties

Name: Keys1

Finetune
0

Relative note
0

MIDI IN
always

Ch.
Any

MIDI OUT
None

Ch.
1

Bank: ...

Program: ...

Scale (%)
100

Figure 5.14: Sunvox Module Properties

At this point configuration of Sunvox is complete. Pressing a key on the MIDI keyboard should play the corresponding key within the synthesizer.

5.3.3 Acquiring and Configuring the BitScope Micro DSO Application

To download the BitScope Micro DSO application visit the official BitScope webpage at: `https://bitscope.com/software/dso/` A similar page similar to that below should appear. At the top right find and click the "download" button. A dropdown will appear, select the option "Raspbian."

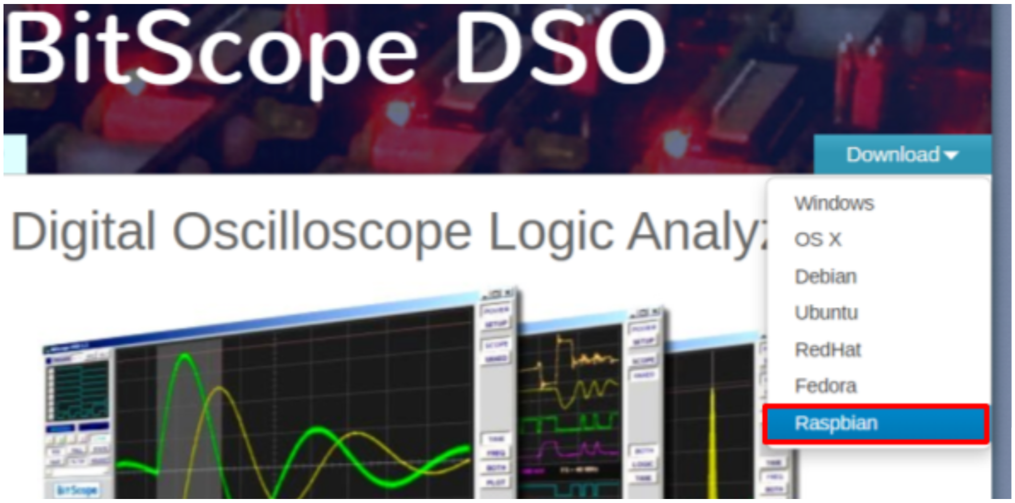


Figure 5.15: BitScope DSO Download

A new page will load with checkboxes for other applications that may be downloaded as well which support the Raspberry Pi. Check the option for BitScope DSO and select "Download."

Once downloaded, extract and install the application. It should now be added to the "programming" section of the Raspberry Pi start menu.

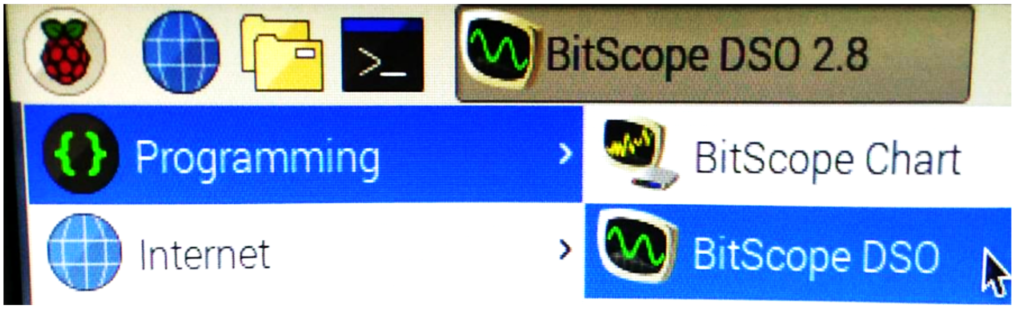


Figure 5.16: Launching the BitScope DSO Application

Once open the application window will show a spinning animation as shown below. Before proceeding, ensure that the BitScope Micro oscilloscope is connected through USB to the Raspberry Pi.

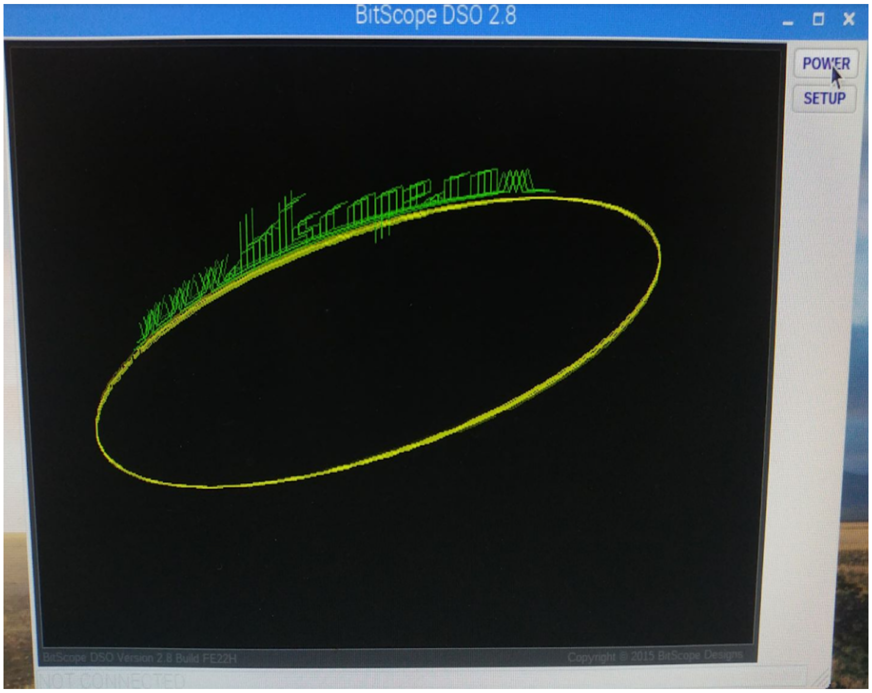
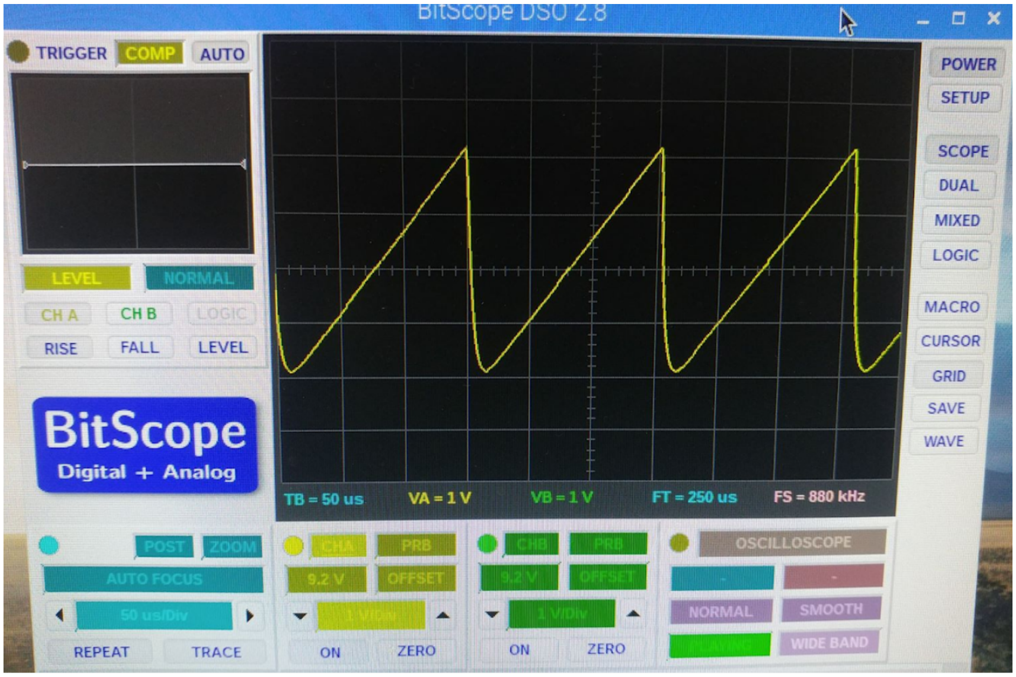

BitScope DSO 2.8
POWER
SETUP
bitscope.com
BitScope DSO Version 2.8 Build FE22H
Copyright © 2015 BitScope Designs
NOT CONNECTED

Figure 5.17: BitScope DSO main screen

The application will finally open to the screen below. From here setup is complete for the BitScope Micro oscilloscope software. Note that along the right column of the application are multiple different functions. One of which will enable a split-screen view of the oscilloscope and spectrum analyser.



TB = 50 us VA = 1 V VB = 1 V FT = 250 us FS = 880 kHz

Figure 5.18: BitScope DSO Waveform Display

Chapter 6

Conclusion

Due to time constraints, the full RasPiano system was unable to be assembled, but various parts of it saw success individually. In this section, we will examine the results of the three most significant subsystems in the RasPiano: the color organ, the filter chain, and the software.

6.1 Color Organ Results

After the team received the PCBs, we got to soldering. The color organ was the most thoroughly tested subsystem, so we had the most confidence that this circuit would work. All of the PCBs were tested before being fully implemented into the system. The only hiccup we found was in the layout of the LED driver board. The barrel jack was added to the layout reversed, which was remedied by adding a short jumper wire. Once performance was verified, we started putting the full system together.

The input to the color organ is a 3.5mm headphone jack. The output of the main color organ PCB is the five envelope signals, which are each sent to an Arduino Nano. For this implementation, the Arduino boards are simply plugged into breadboards for ease of replacement during testing. The PWM signals are then sent to the LED driver board, where the LED strips are connected and powered. The system is powered by a 12V, 1A switching power supply. This power supply provides a barrel plug which is paired with a barrel jack on the LED driver board. A set of pin headers on this board can also be used to provide power to the Arduinos. Then, the 5V output from the Arduinos is used to power the main color organ board via the DCH01 DC-DC converter. The Arduino code can be found in Appendix H.

Once everything was put together, the color organ worked very well. The only real issue was some flickering when the signals output from the main color organ board were very low. This may be due to issues of digital and analog ground separation, an issue that was discovered in the testing phase. Although this was largely eliminated by keeping the analog circuitry on its own PCB, it is possible that some digital noise is finding its way to the analog circuits through the DCH01. Overall, the color organ was very successful.

6.2 Filter Chain Results

The filter was the most interesting and complex part of the system in terms of signals. It was also one of the more important parts of the RasPiano, in that it achieved the system goal of being a teaching tool.

To test the filter, one of the filter PCBs was soldered first. Then, a smaller version of the level translation circuit was constructed on a breadboard. Then, using a spare Arduino Nano and a pair of potentiometers, a simple control system was constructed. The potentiometers were hooked up to the +5V and ground pins of the Arduino, with the wiper pin of each going to an analog input. The Arduino then takes the 10-bit number from the analog input, divides it by 4 to match the resolution of the digital potentiometers, then sends it out via SPI to set the digital potentiometer levels. All of the digital signals are passed through the level translation system.

In the end, the filter circuit was working properly as a filter, and all of the different filter outputs were correct. However, the digital potentiometer was not responding to the control signals sent by the Arduino. We checked the pins on the digital potentiometer to make sure that they were receiving the signals and that there wasn't a short along the path somewhere. Unfortunately, we were unable to locate the issue. Examining the output from the level translation circuit with an oscilloscope, we found some oscillations, characteristic of an underdamped response, shown below. It is possible that the oscillation at the low level is triggering the "high' threshold in the digital potentiometer, but we don't have any way to verify this.

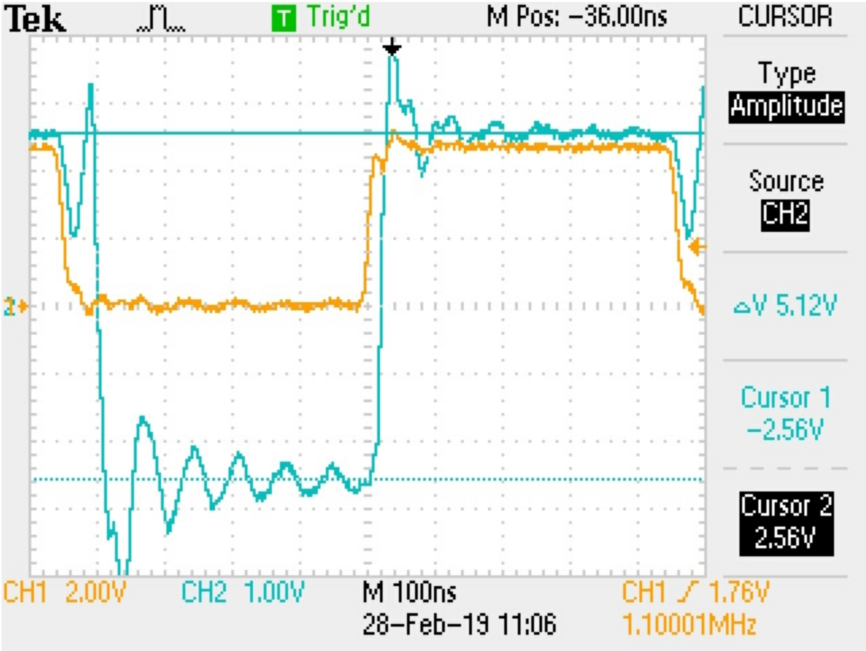


Figure 6.1: Level Translation Underdamped Response

Despite this issue, we were able to measure the transfer function of the filter in the low-pass configuration. It shows a cutoff frequency of about 900Hz and a very high resonant peak, indicating a Q value of at least 5. The frequency response was verified by recreating the circuit in Multisim and comparing the simulated and measured transfer functions, shown below.

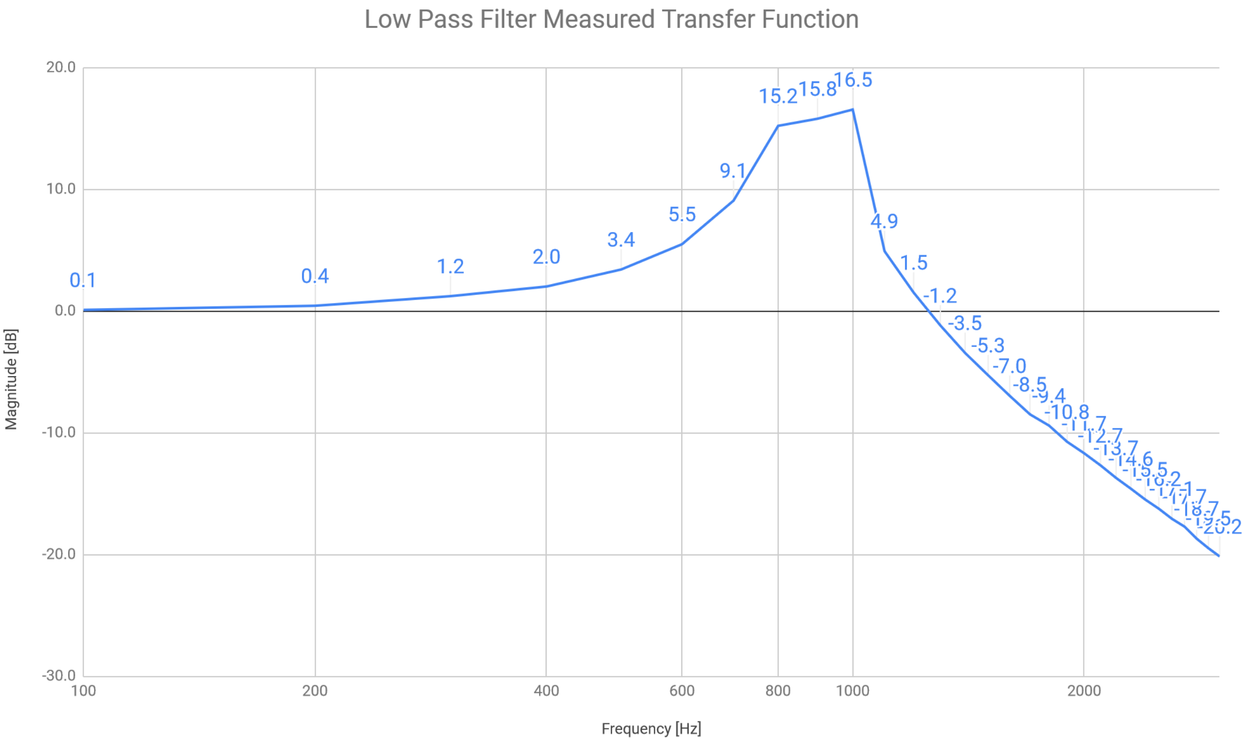


Figure 6.2: RasPiano Filter Measured Frequency Response

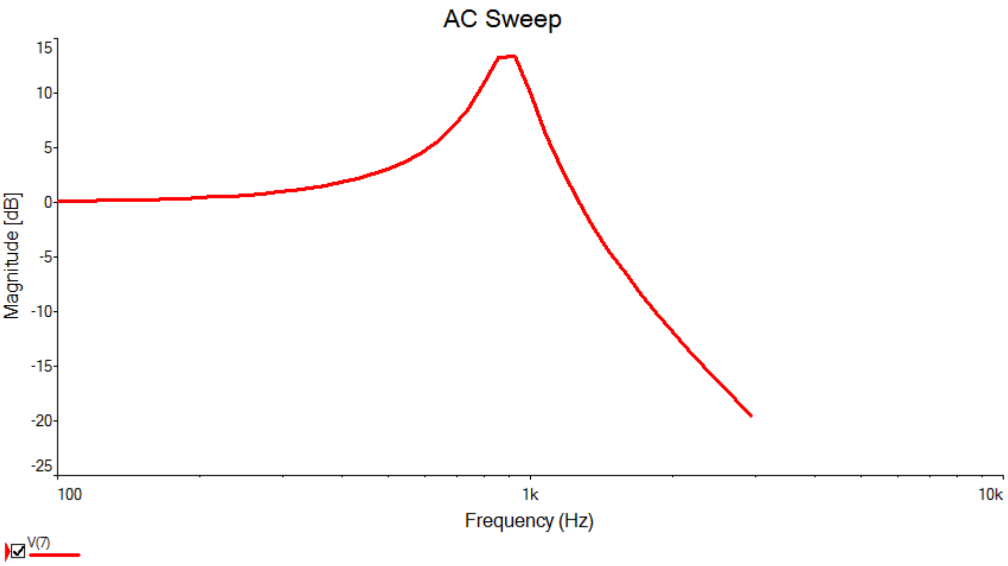


Figure 6.3: RasPiano Filter Simulated Frequency Response

6.3 Software Results

At the moment the RasPiano system has multiple functional blocks but is not yet capable of operating as a singular unit/device. All color organ circuitry and hardware works properly when powered and given an audio signal. Each filter within the filter chain behaves as a operational filter but cannot be modified due to a SPI communication error between the Arduino and digital potentiometers currently in use. With respect to software the Raspberry Pi properly boots into Raspbian but does not have the custom-designed software implemented. As our team ran out of time to implement the software for the RasPiano we decided to provide third-party software to implement the oscilloscope and spectrum analyser applications. A figure of the applications provided as well as the synthesizer is shown below.

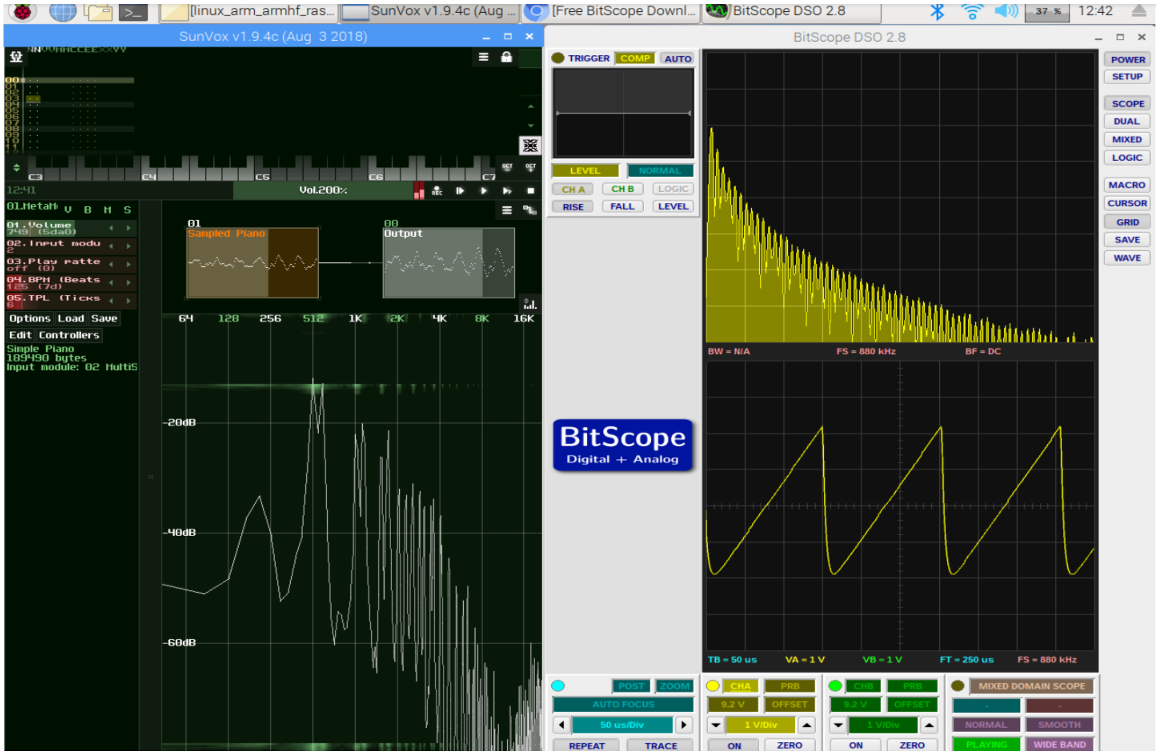


Figure 6.4: RasPiano Applications

On the left is the sunvox synthesizer which takes keystrokes from the USB-connected MIDI keyboard and outputs corresponding piano key samples. This analog signal is outputted from the Raspberry Pi's auxiliary 3.5mm analog output. On the right of the screen is the BitScope Micro DSO application. Also connected through USB, the BitScope Micro oscilloscope passes data to this application to trace an analog waveform and provide spectrum analysis. Unfortunately as we could not construct the corresponding application there is currently no way to view the filter chain's transfer function.

Despite the software not being complete for the RasPiano, the applications pictured above do provide a reasonable demonstration for how the system should be configured. Additionally they also meet our project requirements of providing the oscilloscope and transfer function functionality to the device. For future work there are three primary recommendations we have:

- Troubleshoot and repair digital potentiometer control within the filter chain
- Implement the custom oscilloscope, transfer function, and spectrum analyser applications
- Add custom application control from auxiliary MIDI keyboard buttons
  - Interface this control with the digital potentiometers and transfer function application

As mentioned previously, control over the digital potentiometers used within the different filters of the filter chain was unsuccessful due to an unknown SPI communication error. To ensure this component of the RasPiano is ready for interfacing with the Raspberry Pi it would be critical to troubleshoot and repair the error. With this done each filter would then be capable of being digitally modified and as such controllable by the Raspberry Pi and its applications.

The next recommendation for future work is developing the custom oscilloscope, transfer function, and spectrum analyser applications whose design is specified within part of this report. With these applications operational input from the auxiliary MIDI keyboard controls can be utilized to control the system. Once a user modifies a control for a filter these applications can then provide the appropriate configuration to the digital potentiometers and update the corresponding transfer function on screen.

Chapter 7

Future Recommendations

This chapter serves to fill in the gaps for which systems were not implemented in the RasPiano and how the existing ones can be improved.

7.1 Color Organ

The color organ's performance was excellent, so only minor tweaks to the design are needed. For instance, better separation of digital and analog ground nodes. Instead of pulling the 5V from the Arduinos to use with the DCH01, it may be better to use a voltage regulator such as the LM317 right at the 12V supply to generate the 5V rail. By remaining separate from the digital ground connections and only rejoining the two grounds at the supply, digital noise on the analog ground rail can be avoided.

Aside from this, the other improvement would be to include the Arduinos in the circuit board design and keep the system more discrete overall. The tested implementation used three custom PCBs and two breadboards connected with jumper wires. By combining the circuit board designs and adding headers to plug in the Arduino boards, the system can be made to fit on a single board. This will also make it easier to set up - no messing with dozens of jumper wires.

7.2 Filter Chain

Because the current system is already quite complex, it may be better to start from the beginning rather than trying to fix the existing system. For example, look for digital potentiometers that allow a wider range of voltages on their pins, outside of their supply range. This would eliminate the need for the level translation system and greatly simplify the system.

Another issue to consider is the digital potentiometer values. This stems from the function for the cutoff frequency:

$$f_0 = \frac{1}{2\pi RC}$$

Audio frequencies are heard logarithmically by humans, i.e. one musical octave is equivalent to a doubling in frequency. For this reason, it is desirable to have an exponential increase in cutoff frequency as the input device is moved linearly. However, using a linear potentiometer makes this very difficult. The ideal resistance curve for the potentiometer would be exponential, such that for each increment, the resistance is doubled, causing the cutoff frequency to be halved. This gives us the desired exponential frequency output in reverse, which can be accounted for in code.

The linear digital potentiometer has its increments spaced in such a way that would cause a frequency curve that is too extreme. For the 100kΩ potentiometer with 256 increments, one increment is approximately 400Ω. At the lowest setting of 1 increment. the cutoff frequency would be around 5kHz, cutting off a significant portion of the high-frequency selection. Moving up to 2 increments would mean a cutoff frequency of 2.5 kHz. The cutoff frequency crosses below 1kHz at only 5 increments, and below 100Hz at around 50 increments. This means that around 80% of the range of the potentiometer is dedicated to frequencies below 100Hz.

Logarithmic scale digital potentiometers are difficult to find, so the best way to work around this issue is by introducing a second digital potentiometer in series. The second potentiometer should have a much smaller resistance value with a similar number of increments. This solution offers much more resolution, meaning better control for higher cutoff frequencies. Of course, introducing another chip means introducing another set of control lines, creating additional complexity in board layout. Furthermore, the coding becomes a bit more difficult, as the transfer function calculations will have another variable to work with. However, this extra work is worth it for a much finer control over the cutoff frequency.

7.3 Software

For software future work of the RasPiano there are two primary recommendations we have:

- Implement the custom oscilloscope, transfer function, and spectrum analyser applications
- Add custom application control from auxiliary MIDI keyboard buttons
    - Interface this control with the digital potentiometers and transfer function application

Developing the custom oscilloscope, transfer function, and spectrum analyser applications whose design is specified within part of this report is critical. With these applications operational input from the auxiliary MIDI keyboard controls can be utilized to control the system. Once a user modifies a control for a filter these applications can then provide the appropriate configuration to the digital potentiometers and update the corresponding transfer function on screen.

Part III

LED Stairs

Chapter 8

Introduction

The LED Stairs idea mainly came from a mixture of two innovation sources. The largest driving source of this idea was need, students at Atwater Kent have described the environment of AK as creepy, depressing, old, and dark or dull. As one of WPI's top Engineering buildings, these descriptive words are extremely contradicting of the desired creative, lively, innovative environment all high-achieving technical schools strive for. This project aimed to bring light and positive energy to the building in hopes for a better learning environment for the students.

Another source of innovation was demographics, this tied in with the reasonings above, but in a different view, the students at AK are most often one of the follow: an electrical and/or computer engineer or a robotics engineer. However, most commonly, the students are electrical and computer engineers and for a large portion of these people, Atwater Kent is a second home. It is a shame that most projects on display at the building involve electrical and computer engineering, but none revolve around it. Therefore, we wanted to pursue this project to speak for the students that have love for digital communications, sensors, and lights. We even went as far as to incorporate an LED symbol within the design of the Light Bars.

To summarize, this project aimed to add to the environment at AK as well as showcase something that is purely electrical and computer engineering.

Chapter 9

Design Options

9.1 General Overview

In this section we will provide an overview of the elements that define the LED stair project. Here we will cover where the final installation will be placed and its overall layout, a block diagram discussing how the different components of the system are expected to behave and interact, and end with the final system specifications and technical requirements.

9.1.1 System Specifications and Technical Requirements

Before we begin discussing the specific design and hardware choices in implementing the led stair project, we now cover the technical requirements and specifications for developing the project. Here we will utilize the project objectives and customer requirements detailed in part 3 chapter 1, and part 1 chapter 1 respectively to answer the following questions:

- What inputs must be available for users to interact with the device?
- What should happen in response?
- What kind of hardware is required to accomplish this user-device interaction?
- How will it all be powered?
- What software is required to coordinate device operation?
- How should the system be housed?
- Since the system will be distributed along a full staircase, how should it be organized?

Answering the questions above provides an architecture to follow in designing the LED stair system.

9.1.1.1 Input, Controls, and Ease of Use

The LED stair system is intended to be a primarily autonomous system. It should interact and respond to users without requiring any form of setting or configuration. Therefore no controls (if any) should be provided anywhere aside from a primary control panel if needed.

In general the system should be capable of controlling its brightness depending on how bright the available ambient light is. During the day there will be plenty of natural sunlight entering the stairway area, therefore the lights should be brighter than they will be at night. To accomplish this, some form of light sensing circuitry should be provided.

Additionally, the system should be aware of human presence and interaction. When at least one person is traversing the steps, individual lights corresponding to their position should light up. If a given step is unoccupied, it should be off. Therefore, a sensor should be present on each step that detects when an individual is on a given step. Animations should occur when no one has activated any given step for a short period of time, or when there is simply no interaction at all. A system should then be in place that detects when none of the steps are triggered and potentially for how long.

Finally the LED stairs should be relatively easy to scale. If an owner of the steps wishes to add additional lighting elements for additional steps, it should not be difficult to accomplish or maintain.

9.1.1.2 Software

Any software driving the system should accomplish the following:

- Based on the amount of ambient light available, configure the brightness of the steps
- If no interaction with the steps is found, begin playing animations
- If any step is activated, cease animation and light only interacting steps

9.1.1.3 Power, Safety, and Housing

The system should be sufficiently capable of powering itself even under extreme scenarios such as all steps being activated simultaneously at full brightness. In addition to capably powering itself, the system should be able to mechanically handle a maximum use scenario. No power lines should overheat due to maximum use and potentially cause wire burns. If any potentially dangerous technologies are used such as lasers for presence detection, appropriate labeling should be used following any appropriate safety standards. Finally, nothing should provide any tripping hazards by being placed on the steps in the way of their traversal.

Electronic circuitry should be housed and not exposed. Housings should also protect any lighting elements, computers, and sensors from damage or misuse.

9.1.1.4 System Specifications and Technical Requirements Summary

To summarize all the technical requirements laid out above, we provide the bullet list below which organizes by category the design goals for each facet of the system.

- Input and Controls
  - Controls (if any) should be positioned in a central location
  - Ambient light sensing should be present
  - Lighting elements should be placed on every individual step
  - Brightness control should be available for the entire LED stair system
  - Occupation sensors should be present on each step
  - The system should be relatively easy to scale
- Software
  - System-wide brightness should be controlled from available ambient light data
  - Animations should be available during inactivity
  - Animations should cease during stair traversal and replay after a "cool down" period
- Power and Safety
  - The system should properly utilize available power
  - All circuitry and cabling should be capable of handling any and all expected electrical load
  - Any potential hazards such as lasers should be explicitly labeled
  - The system should meet electrical and safety building standards
- Housing
  - Electrical circuitry and lighting elements should be housed
  - Housings should protect circuitry and lighting elements from damage or misuse

9.1.2 Design Process

9.1.2.1 Installation Location

Beginning with the LED stairs installation location and layout, the system will be placed on the stairwell located at the front left of Atwater Kent pictured below.

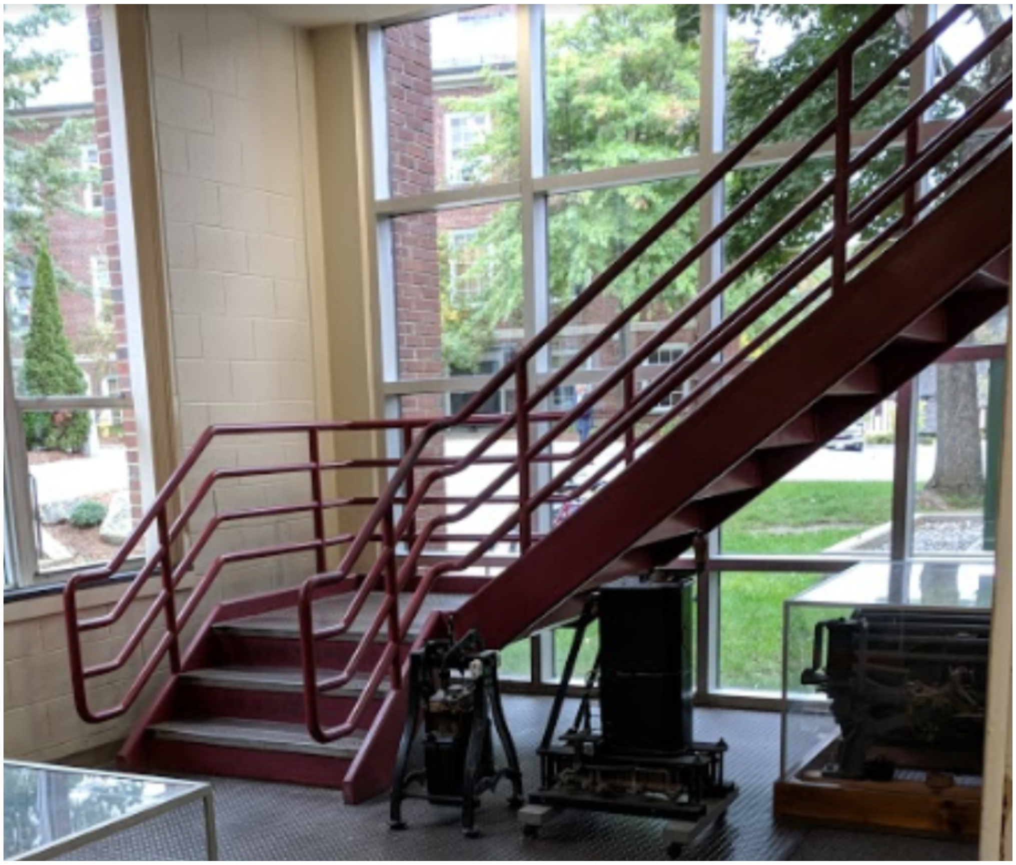


Figure 9.1: Atwater Kent Front Staircase

The purpose of this is to provide a display that can be seen from both inside and outside the building. This area provides the perfect location as it is surrounded by glass windows. For a better reference of how the system will be laid out, please refer to the 3D model below.

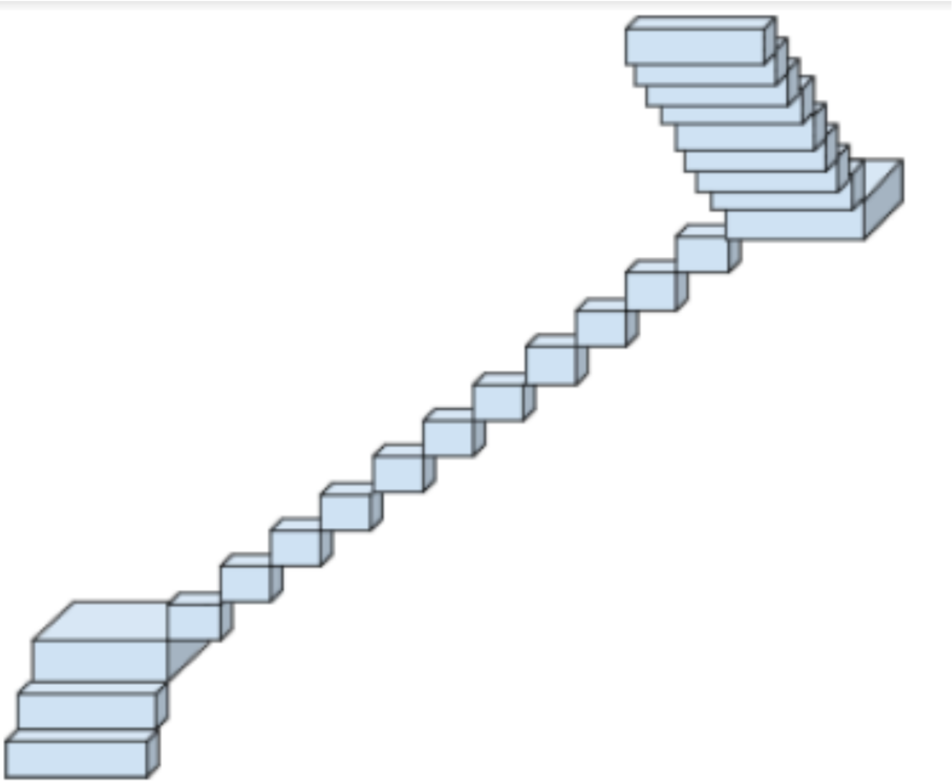


Figure 9.2: Atwater Kent Front Staircase 3D Model

Notice that the staircase consists of three sections: two bottom steps, eleven center steps, and eight top steps. On each step we place a vertical light bar whose housing is designed to replicate the symbol for an led circuit component as shown below.

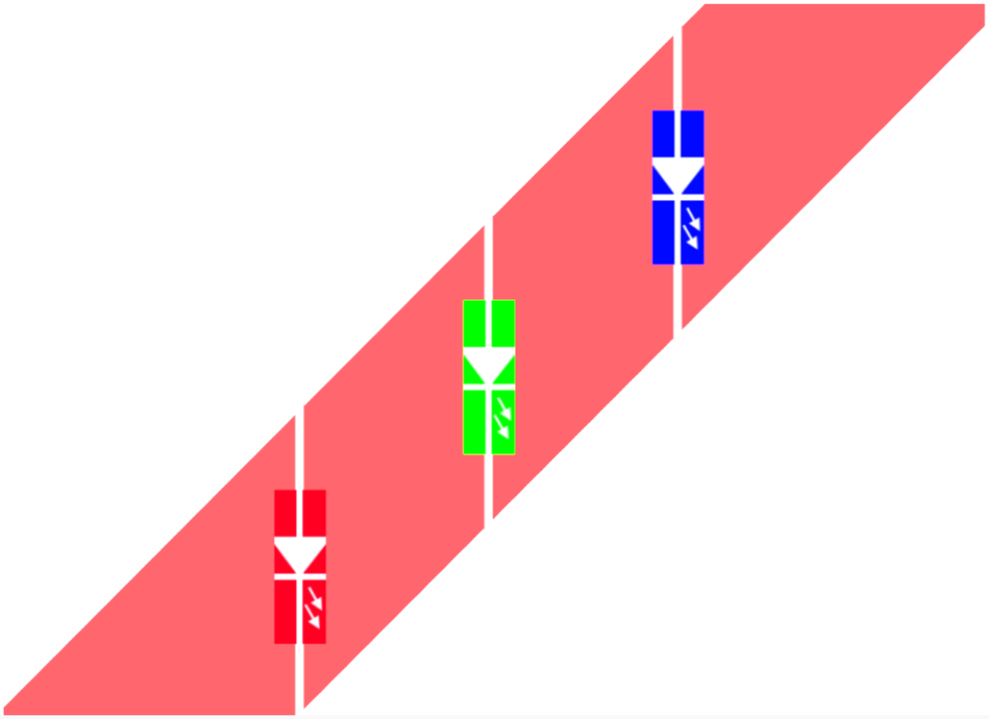


Figure 9.3: Concept sketch for vertical light bars

Because the opposite side of the staircase center portion is visible from the outside, we will place here a series of “dummy” light bars which simply replicate the lighting of their complement light bar. In total this means that we will have 32 light bars. 21 with individual lighting effects and 11 duplicates.

9.1.2.2 User Experience

When no one is present on the staircase, the system will autonomously play animations over time. As soon as a single individual steps onto the staircase, the animations will disable and begin illumination by occupancy. When a person walks alongs the steps, a light bar will illuminate corresponding to the step the individual currently occupies. This functions as well when multiple individuals occupy multiple steps as well.

To preserve energy and optimize visibility during different times of the day the system will record the ambient light brightness and update the brightness of the light bars accordingly.

9.1.2.3 System Block Diagram

To accomplish the behavior above the system is designed to follow the block diagram pictured below.

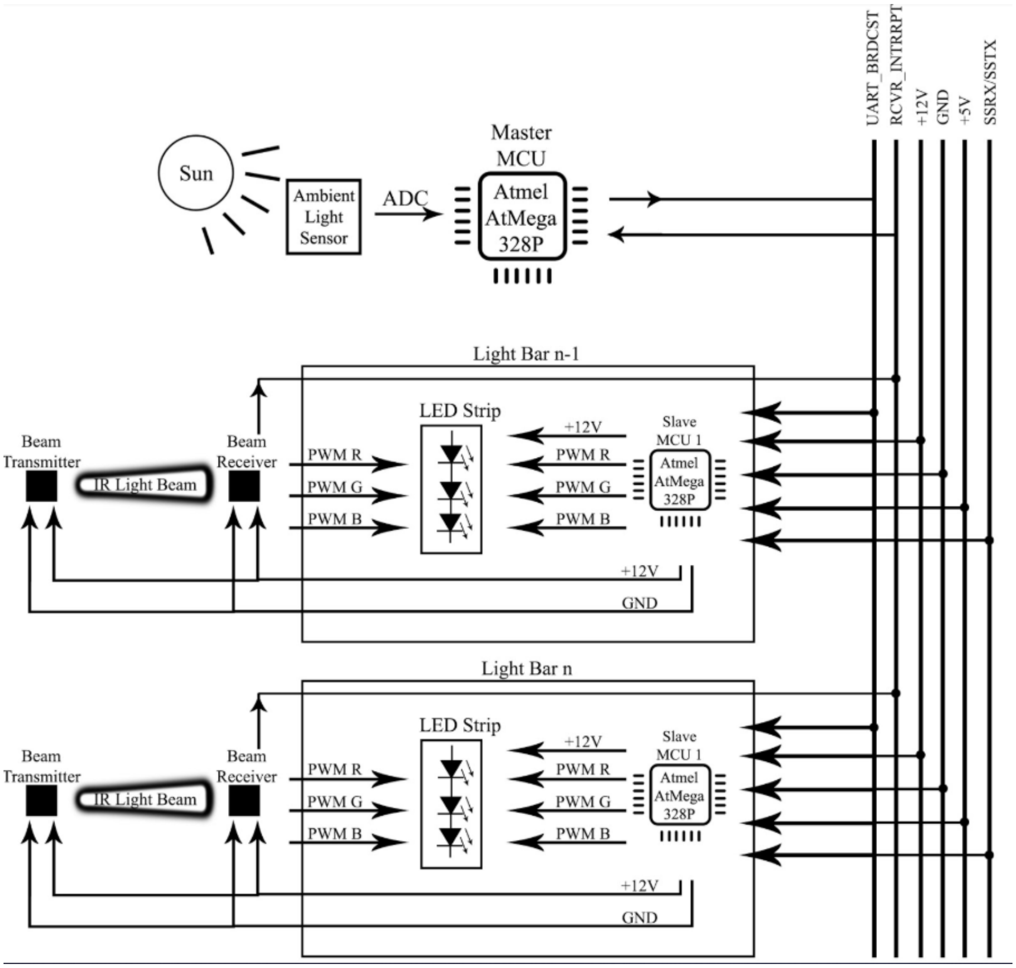


Figure 9.4: LED Stair System Block Diagram

As of its current design all animation is controlled by the master microcontroller (MCU). As shown in the block the master MCU communicates through UART to all slave light bars. When the slave MCU embedded in each light bar receives a message from the master that addresses it, it responds by illuminating the specified color.

Each light bar's lighting element can also be controlled by two sources: the slave MCU and the corresponding trip beam receiver. This was implemented to allow trip sensors to illuminate light bars while the master MCU disables animations within the system. Furthermore the master MCU utilizes a receiver interrupt signal that is shared common by all receivers to initiate animation disabling.

To control the overall brightness of the system, the master MCU uses its ADC to read the value of the ambient light sensor and modify the overall brightness levels.

Note that every device within the block diagram is powered by the 5V rail except for the led strips within each led bar which require 12V. The remaining SSRX/SSTX line within the block diagram is currently unused as we were not able to complete this feature. Specifically, this line is tasked with communicating messages from light bar to light bar upwards to the master during power-up. Ideally the master MCU would use this to determine the number of light bars in the system as well as the topography and address of each light bar that is connected. Unfortunately however, since this is not yet an implemented feature, the addresses of each light bar must be hard-coded into the master MCU.

9.2 Inputs

The LED Stairs system only takes in two inputs: a step sensor and ambient light sensor. The step sensor sent data to the slave for that step to alert the master of an interrupt and tell the coordinating light bar to turn on. The ambient light sensor is located on the same PCB as the master and sends data directly to the master to adjust the brightness of the system for efficiency.

9.2.1 Step Sensor

For the LED Stairs system to work, there must be some way to communicate when a user is traversing the steps. This can be done using a sensor to tell the LEDs to turn on and let the microcontroller know when to stop animations. To decide what sensor to use, specifications of the component were decided.

1. Accuracy: the sensor should only signal when a user is on one particular step. There should be no occurrences of a user being detected on an adjacent step and the LED being turned on incorrectly.

2. Sensing Distance: if someone steps on a step, their foot could be anywhere on the 113 cm by 29 cm step. The sensor would need to be able to detect a user in that area.

3. Awareness of Presence: the component needs to be able to detect presence. It should signal the moment a user is on the step and signal once that user steps off. If the component only detects motion, it would lead to a flaw in the system.

4. Cost: while considering options, it was important to maintain an eye on the budget. The three specifications above are non-negotiable, but all decisions following are primarily based on having the component be low-cost.

5. Power Consumption: ideally, the component would have low power consumption. This specification did not make or break the decision, but was kept in mind while considering options for the sensor.

Keeping these aspects of our ideal sensor in mind, we began designing. The design process had multiple turns and changed quite a bit over the course of this project. In the following sections, we will discuss this process along with details on our final design. The final design was an IR Sensor, this included an emitter circuit and receiver circuit. The emitter was pulsed at 38 KHz that emitted to the receiver that outputted a digital signal dependant on the frequency received.

9.2.1.1 Options, Design Changes

Originally, it was assumed that Pyroelectric (Passive) Infrared Sensors would be our best route, but before beginning our search for a part we wanted to be sure of this. In this research, we found an alternative solution, to instead use a laser trip-beam sensor.

Unlike a PIR sensor, the only thing the receiver would be concerned with is one laser beam. Once a user steps on a step, the receiver will know due to a break in the beam, whereas a PIR sensor may go off whenever something is sensed within its 120 degree cone. The laser not only has great accuracy, but the sensing distance is not a concern with the appropriate laser diode. This specification (sensing distance), along with cost, was the main reasoning behind building the sensor ourselves. Break beam sensors found online did not meet the sensing distance requirement unless they were out of price range. Lastly, the laser sensor is perfect for giving exact times on when the user steps on and off the step. Since the laser beam at the receiver would instantaneously be off or on with the user, we do not have to worry about using delays or communication between the steps to show when a user is no long present.

The schematic for the laser sensor is shown below:

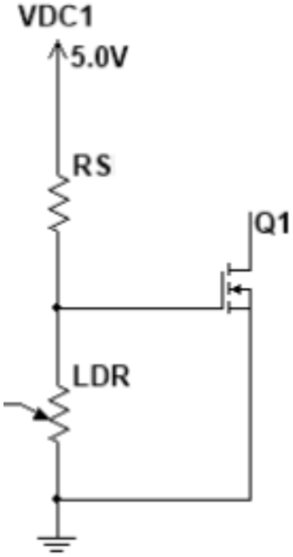


Figure 9.5: Trip-Beam Sensor Circuit Schematic

The circuit is made of a resistor, MOSFET, and photo resistor. The photo resistor chosen was a basic CdS photo resistor from Adafruit for $0.95.

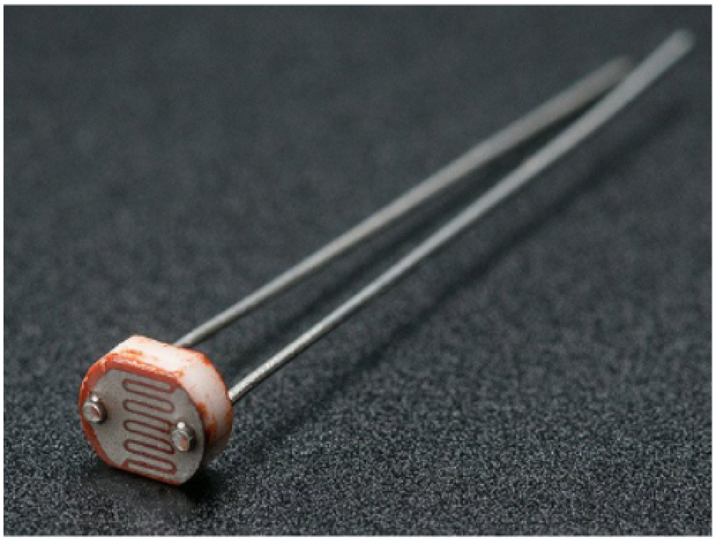


Figure 9.6: Photo Resistor Part Selection for Trip-Beam Sensor

To determine the value required for the resistor, we examined the specifications of the photo resistor and determined its maximum and minimum resistances. These values were obtained by varying the light exposure of the cell and measuring the device's resistance using an ohmmeter. The maximum resistance was measured at 80k Ohms with a minimum of 90 Ohms.

The node that was connected to the MOSFET is Vg. From the diagram, we pulled two equations:

$$V_{GMAX} = 5V \cdot \frac{R_{LDRMAX}}{R_{LDRMAX} + R_S}$$

$$V_{GMIN} = 5V \cdot \frac{R_{LDRMIN}}{R_{LDRMIN} + R_S}$$

When plotting Equation x.a and x.b over Rs, it results in the graph below. Blue represents Vgmax and red, Vgmin

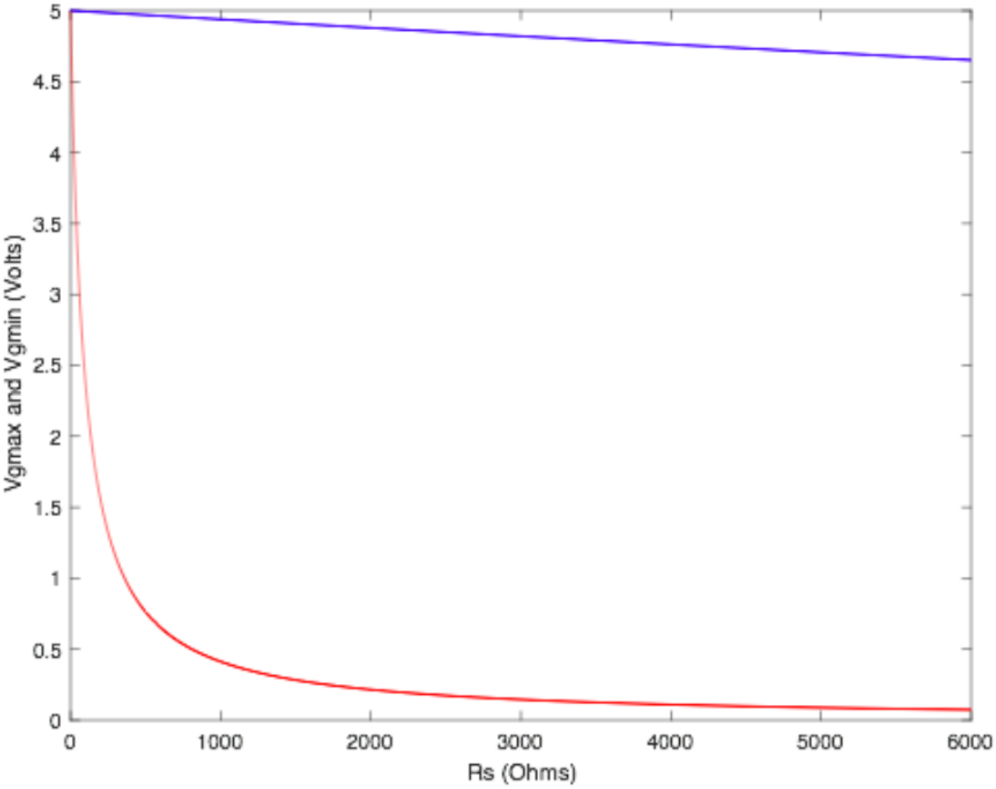


Figure 9.7: Max Vg and Min Vg vs Rs

Since the desired outcome is to build a circuit that can output changes in light, Rs should be chosen to achieve the largest difference in value of Vgmax and Vgmin. To better visualize this, we took the difference of Vgmax and Vgmin:

$$V_{GMAX} - V_{GMIN} = 5V \cdot \frac{R_{LDRMAX}}{R_{LDRMAX} + R_S} - 5V \cdot \frac{R_{LDRMIN}}{R_{LDRMIN} + R_S}$$

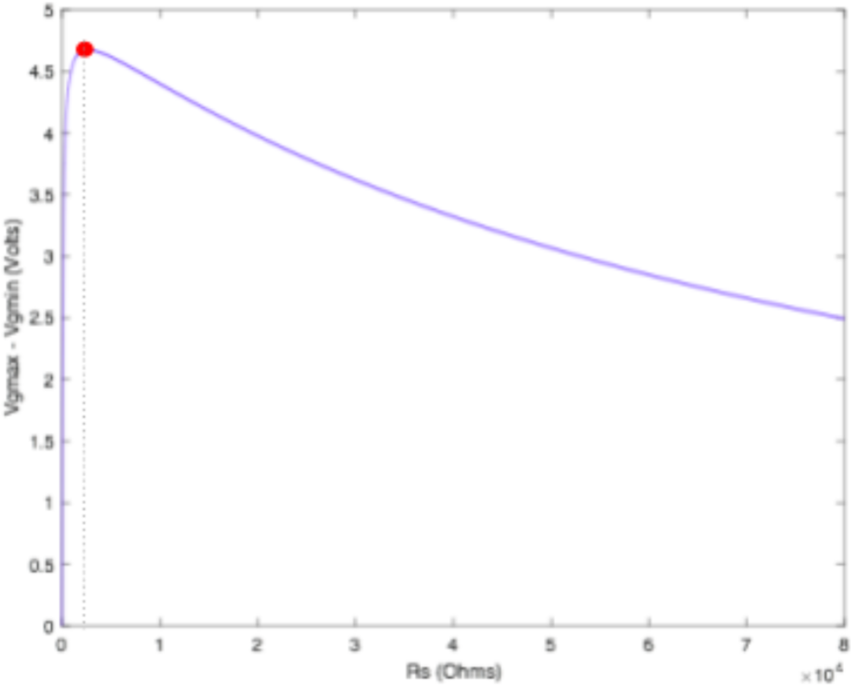


Figure 9.8: Vgmax-Vgmin vs Rs

In the figure above, the peak point is emphasized to show the desired Rs value, when the slope of the difference of Vgmax and Vgmin is equal to zero. Rs was solved for by taking the derivative of (Vgmax-Vgmin) equal to zero.

$$0 = \frac{d}{dR_S}[\frac{80000}{80000 + R_S}] = \frac{90}{(90 + R_S)^2} - \frac{80000}{(80000 + R_S)^2} = 90(80000 + R_S)^2 - 80000(90 + R_S)^2$$

$$R_S = \sqrt{\frac{80000(90^2) - 90(80000^2)}{90 - 80000}} = 2683.281573$$

This value was rounded to 2.7k Ohms for the series resistor.

To communicate when the sensor is tripped by switching the current on and off, the gate of the MOSFET (Vg) is connected to the voltage divider. An n-Channel MOSFET will carry electrons from the source to the drain, flowing current from the drain to the source. However, in order for this current to flow, a high enough positive voltage must be applied to the gate. In order to turn on the LED Strips, they must connect to a 12 V source and ground. Therefore, by connecting the ground pin of the LEDs to the drain of the MOSFET (with the source of the MOSFET connected to ground), the switch of the LEDs can be controlled by Vg. Once Vg is high enough, the MOSFET will allow current to flow, connecting the LED Strip to ground, thus turning the strip on.

Once deciding on the type of MOSFET, we were able to choose what specific part would be best for this system. Since the lab offers n-Channel MOSFETs, we used an IRF520IR-ND to test the circuit.

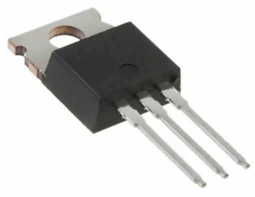


Figure 9.9: MOSFET Part Selection for Trip-Beam Sensor

The emitter was simply a 5mA red laser diode purchased from Adafruit:

Figure 9.10: Laser Diode Selection for Trip-Beam Sensor

This sensor worked as expected implemented within our first prototype. The prototype circuits for the receivers were built using FR4 and copper tape. These circuits were built to fit into 1.38" by 1.38" black boxes ordered from DigiKey for $2.30 each. Each receiver circuit had soldered on wires that stuck through drill holes to the outside of the enclosures. These same enclosures were also used to house the lasers.

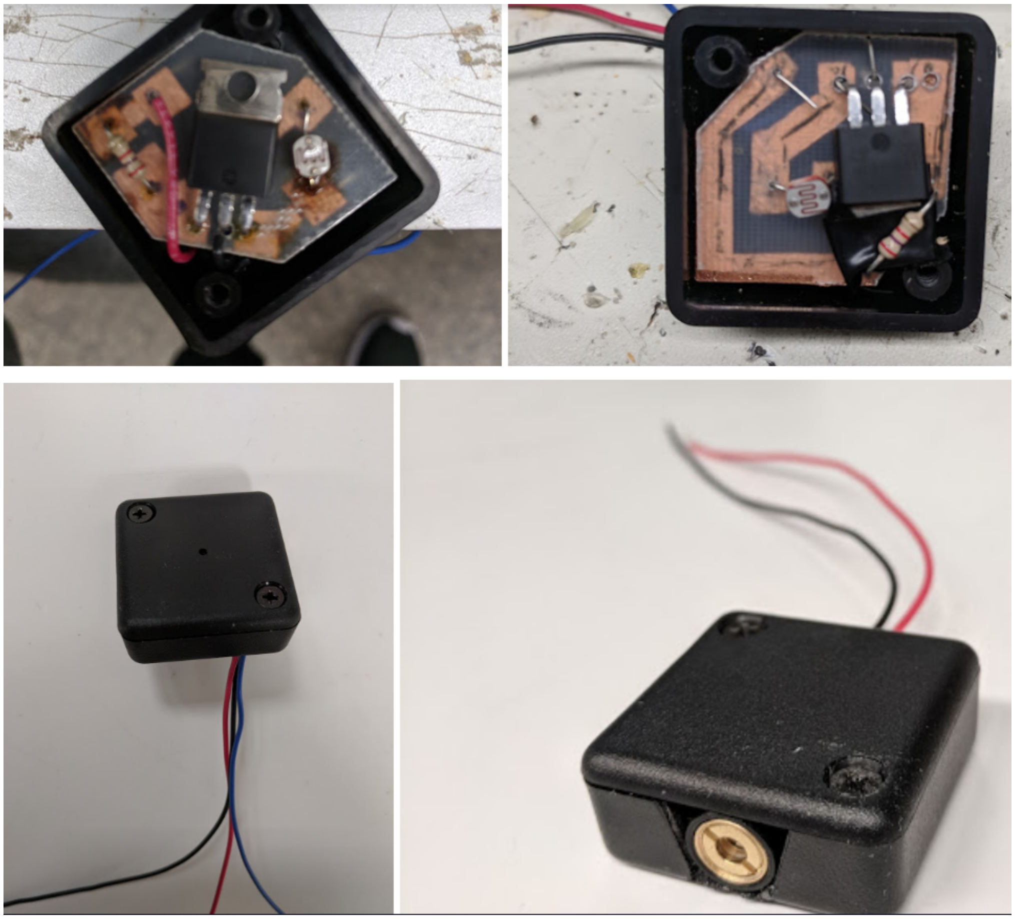


Figure 9.11: First Prototype of Step Sensor

However, we still had two major concerns with this circuit, safety and precision. Safety was handled by doing some research on laws and regulations with lasers. Although this would still be considered a hazard, legally, as long as we notified users of this hazard, we would be covered.

“Laser projectors and laser light shows are “demonstration laser products” as defined by 21 CFR 1040.10(b)(13):

“Demonstration laser product means a laser product manufactured, designed, intended, or promoted for purposes of demonstration, entertainment, advertising display, or artistic composition.”

Lasers promoted for entertainment purposes or amusement also meet FDA’s definition for “demonstration laser products.”

Laser products promoted for demonstration purposes are limited to hazard Class IIIa by FDA regulation 21 CFR 1040.11(c).

This means that projectors are limited to 5 milliwatts output power in the visible wavelength range from 400 to 710 nanometers. There are also limits for any invisible wavelengths and for short pulses. Laser light show projectors therefore may not exceed the accessible emission limits of CDRH Class IIIa. Laser light show manufacturers must submit a variance request for FDA approval in order to sell and operate higher class (Class IIIb and IV) laser light show equipment.”

According to the FDA, our product met the safety requirements for laser use within any advertising display or artistic composition. Since our laser fell under class IIIA, OSHA asks that we display a warning sign so that bypassers are aware where the lasers are located and the potential dangers.

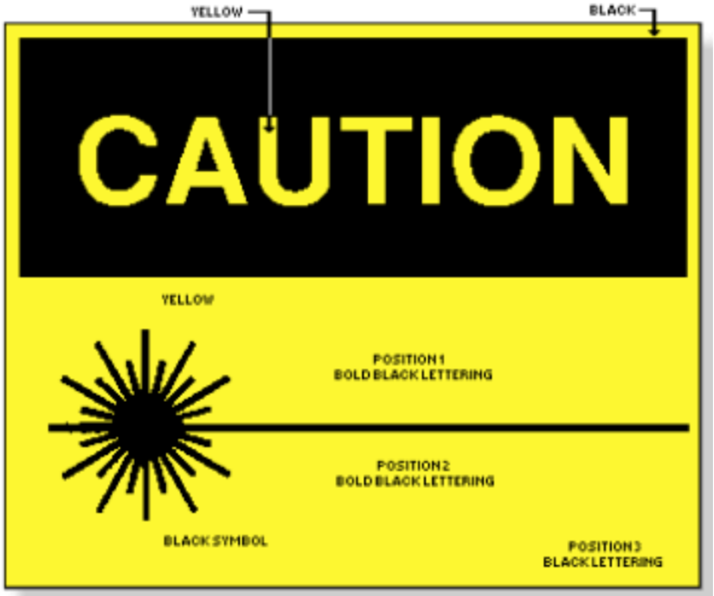


Figure 9.12: Caution Sign

After dealing with the first concern to the best of our ability while sticking with the current design, our next goal was to ensure precision of the laser beam onto the photocell. Various possible solutions were considered for this:

- Have laser shine through hole in enclosure and hit photocell placed inside enclosure on opposite side.
- Diffuse the laser into the enclosure with the photocell
  - CONS: May be too small of a difference for photocell since the light will be diffused.
- Point the laser at the photocell with use of a biconvex lens:

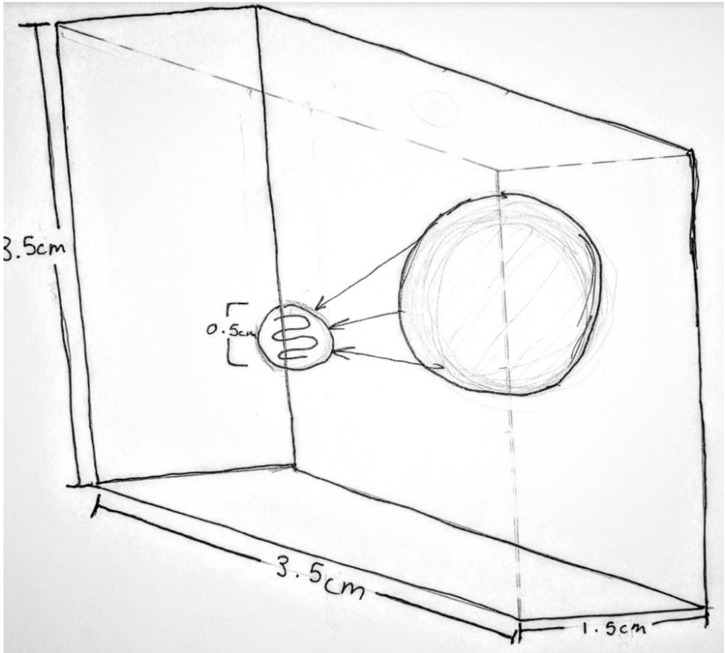


Figure 9.13: Biconvex Lens Idea Setup

  -
  - Here, the lens and the photocell are placed in the center of the faces of the box, making it so that no matter how the laser hits the lense, the light will focus at the one point (the LDR).
- Change the design, and have logic output of IC sensor, QSE159.

Figure 9.14: IR Receiver - QSE159

At first, the top choice was the biconvex lens, but upon taking deeper look into the IC sensor, we ended up changing the circuit completely. By using the QSE159 and IR LED, we would not have to worry about the placement of the laser or students tampering with the laser beam.

In the datasheet for the QSE159, it suggests that the device is matched with a QEE113, using these two devices, the following circuit was made and tested.

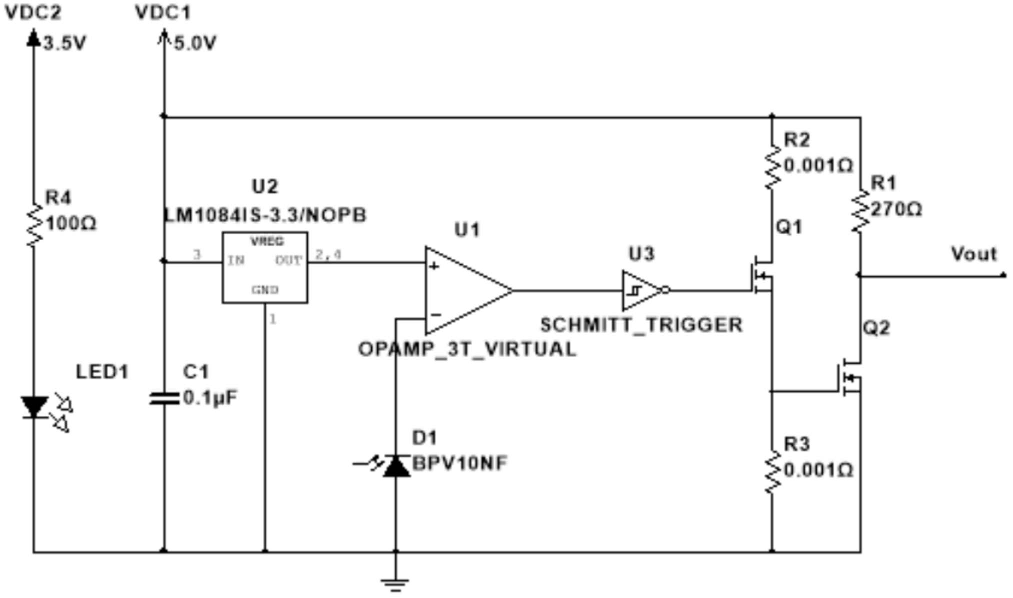


Figure 9.15: First IR Sensor Circuit Schematic

When this was first tested, an error would occur with the 5V output not falling back to a 0V output. This was fixed by placing a 0.1uF capacitor between the ground and source inputs to the QSE159. The circuit then worked properly for a distance of up to approximately 30 centimeters. As one could guess, this was a problem. The sensor for our application must work a minimum of 113 cm.

To increase the power of the emitter, we pulsed the LED with a 555 timer. Since the emitter was now being pulsed, we also switched out the component for the receiver with a TSOP38238. The TSOP38238 detects a 38 KHz frequency and outputs an active high signal when the signal is interrupted. An additional method used for increased the distance was putting two IR LEDs (TSAL6200) in series. These were put in series rather than parallel to maximize the current going through them.

9.2.1.2 Final Design

The schematics and layouts for the final design of the step sensors were made using Eagle CAD.

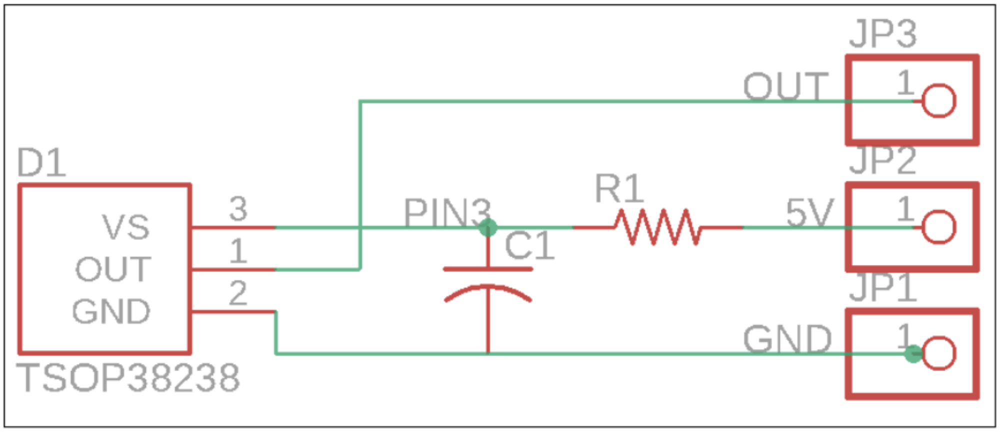


Figure 9.16: Final Step Sensor Receiver Circuit Schematic

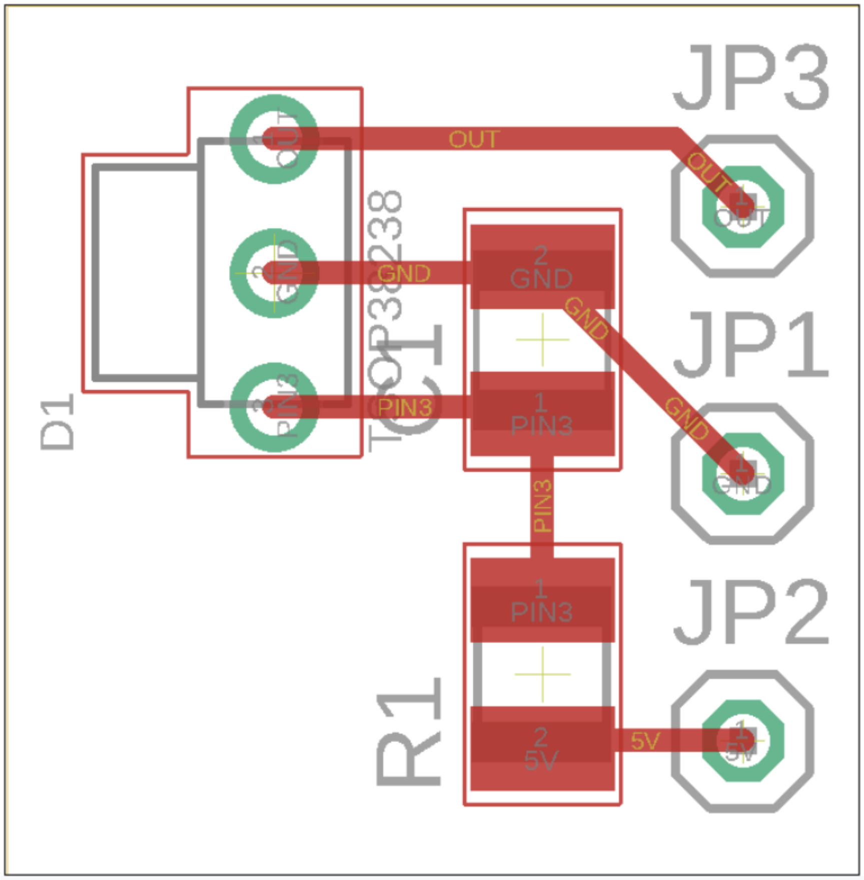


Figure 9.17: Final Step Sensor Receiver Circuit PCB

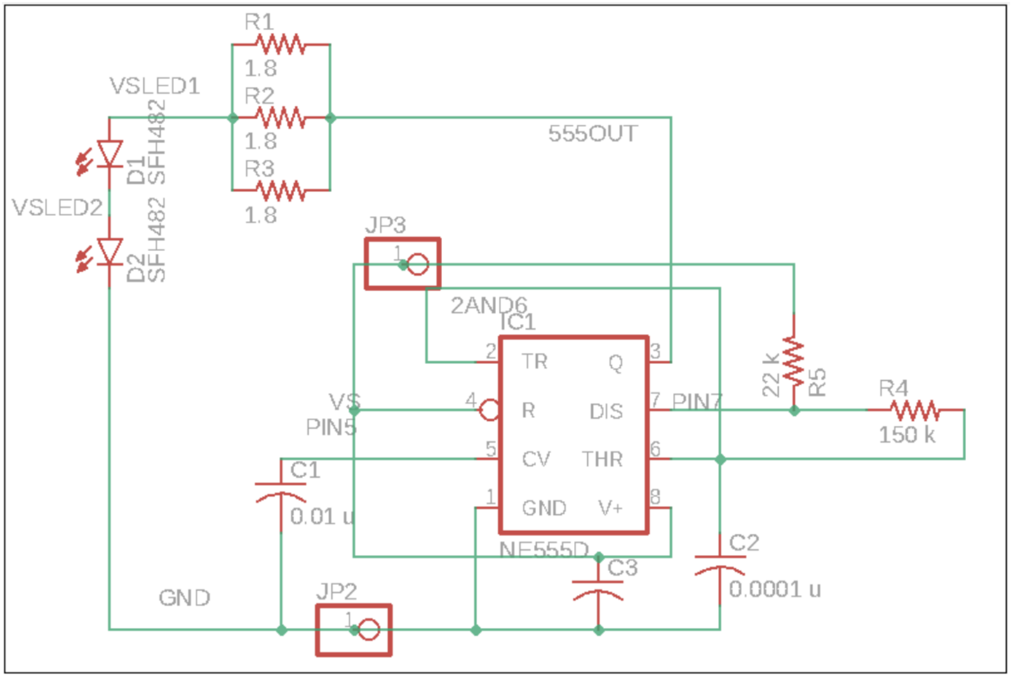


Figure 9.18: Final Step Sensor Emitter Circuit Schematic

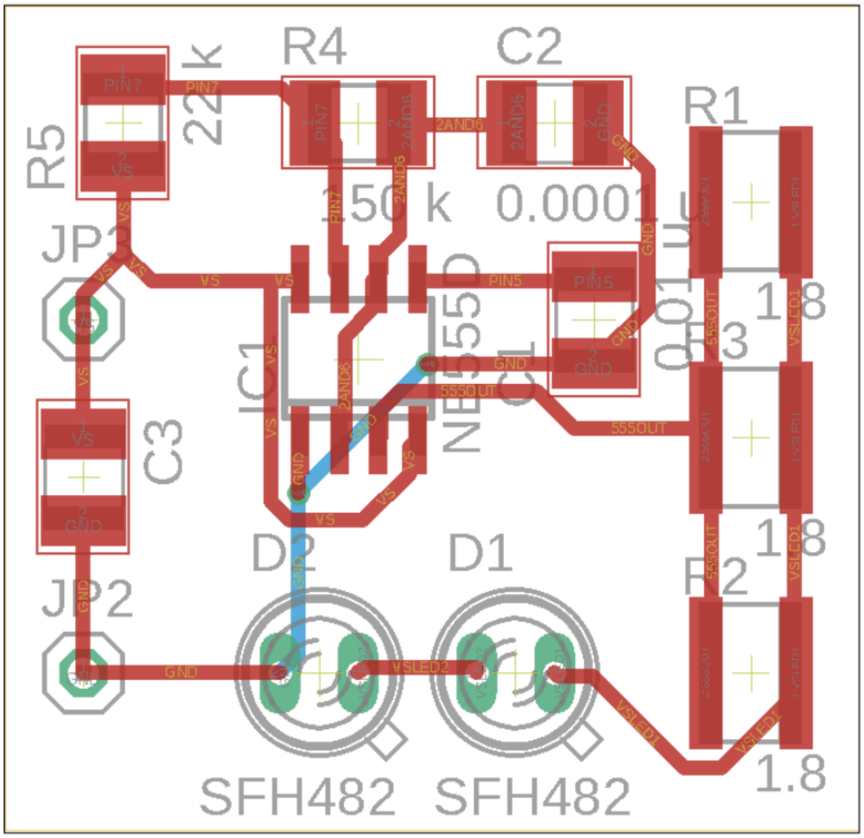


Figure 9.19: Final Step Sensor Emitter Circuit PCB

The output of the 555 timer needed to give off a 38 kHz signal with 5V peak for the IR LED to emit at the correct frequency for the TSOP38238. Therefore, our first task was to get an output at 38 KHz from the 555. The frequency output was dependent on the component values of pins 6, 2, and 7. For a 38 KHz frequency, R5 had a value of 22 kOhms, R4 had a value of 150 kOhms, and C2 needed to be equal to 0.0001 uF (see Figure N to reference component labels). Below is a photo of the oscilloscope screen for pin 3 (output) of the 555 timer in the emitter circuit.

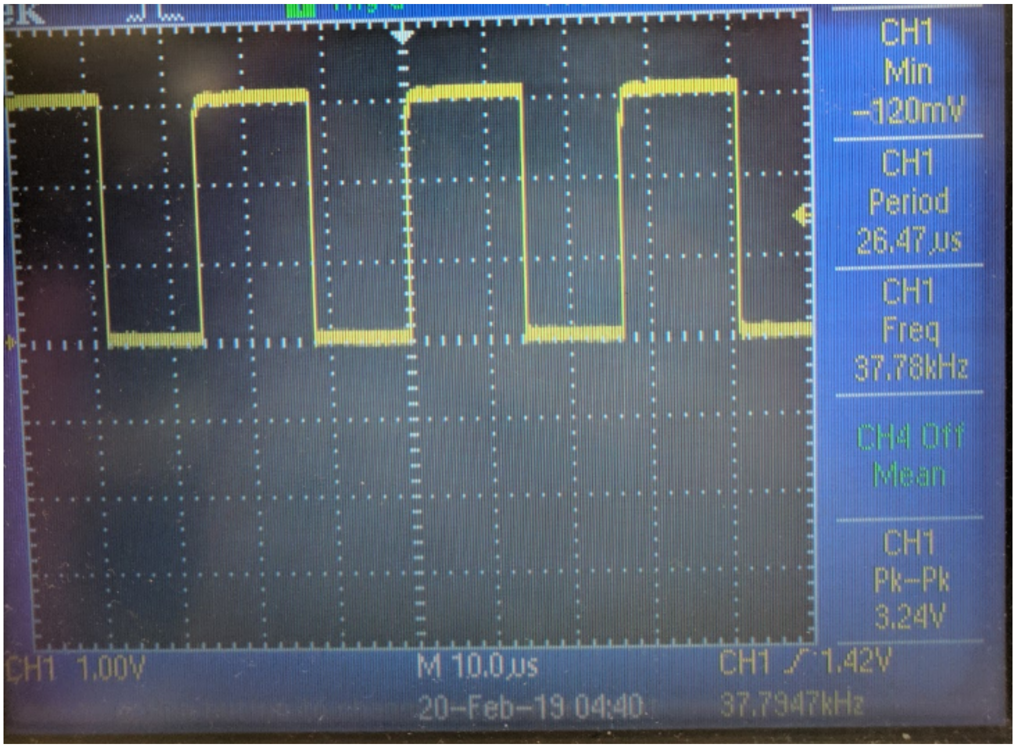


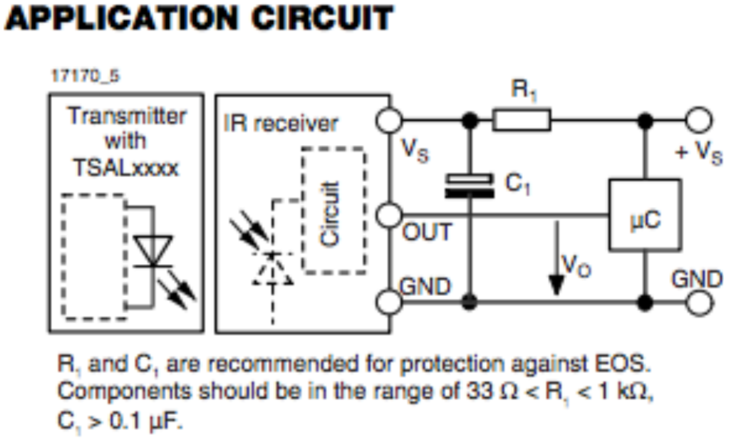
Figure 9.20: 38 KHz Emitter Signal - 555 Output

Once we obtained this frequency, we needed to work on powering the LEDs. Specified in the TSAL6200 data sheet, the max current for the device was 100mA, however, using the 555 timer, we were able to increase this. So the next step was to get from a pulsed 5V output to the input for the cathode of the first LED in series and supply it the right amount of power. Using a current value of 1 A pulsed, the data sheet for the TSAL6200 specifies a voltage drop of approximately 2.2V. Therefore, the voltage drop across two LEDs would be around 4.4V. This leaves us with 0.6V from our 5V supply. This voltage was used to determine resistance value that will give a desired current of 1 A.

$$R = \frac{5V - 4.4V}{1A} = 0.6\Omega$$

To ensure that the resistor would not burn out, we put three high power 1.8 Ohm resistors in parallel. Therefore, there is a 0.6V drop across each resistor with a current of 0.33 Amps flowing across each. This gives a current of 1 Amp to the LEDs in series.

The circuit used for the receiver was identical to that in the data sheet of the TSOP38238:



R_1 and C_1 are recommended for protection against EOS. Components should be in the range of 33 Ω < R_1 < 1 kΩ, C_1 > 0.1 µF.

Figure 9.21: Application Schematic for TSOP38238

For our circuit, we used a value of 100 Ohms for R1 and 1 uF for C1 per recommendation from the data sheet.

The PCBs were ordered through JLCPCB, and soldered by our team, the final products can be seen in the figures below.

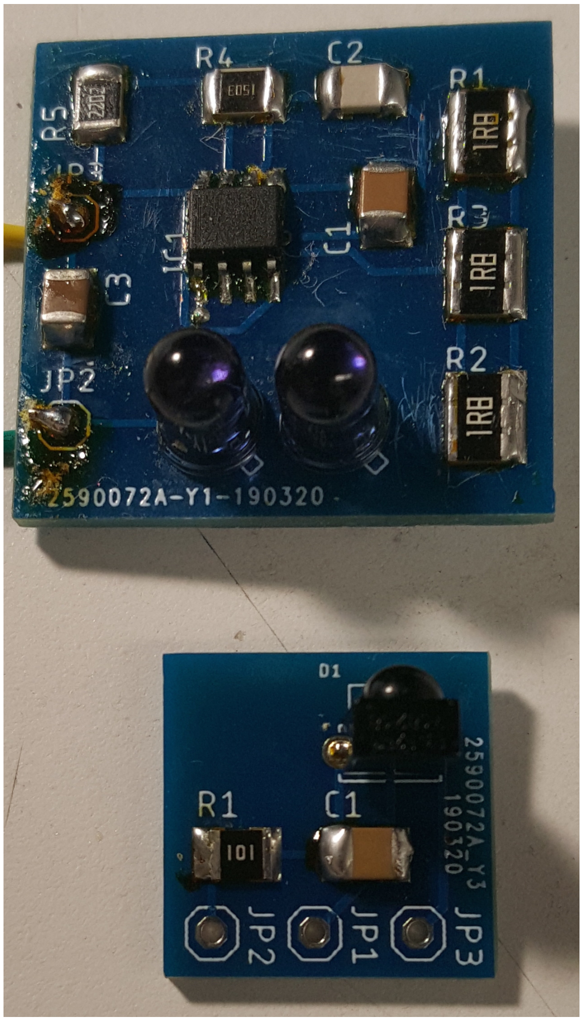


Figure 9.22: Emitter and Receiver Soldered PCBs

Each PCB was tested and verified after soldering, working perfectly when supplied 5 Volts. For our final prototype, we did not housing these as planned for installation. Our installation plan was to put them within raceways and cut small holes in the raceways for the emitter and receiver to "look' through. However, for our final prototype we just simply had the PCBs facing each other without any housing involved. The sensors still functioned as planned and the demonstration was successful.

9.2.2 Ambient Light Sensor

The brightness of the LEDs is directly related to the power consumed by each. Therefore, to maximize power efficiency, we have added an ambient light sensor to our system to output data to the microcontroller. For this, we used a similar circuit as our original step sensor circuit. Both circuits use the same photoresistor part, but the resistor value and placement was modified and the MOSFET is no longer being implemented.

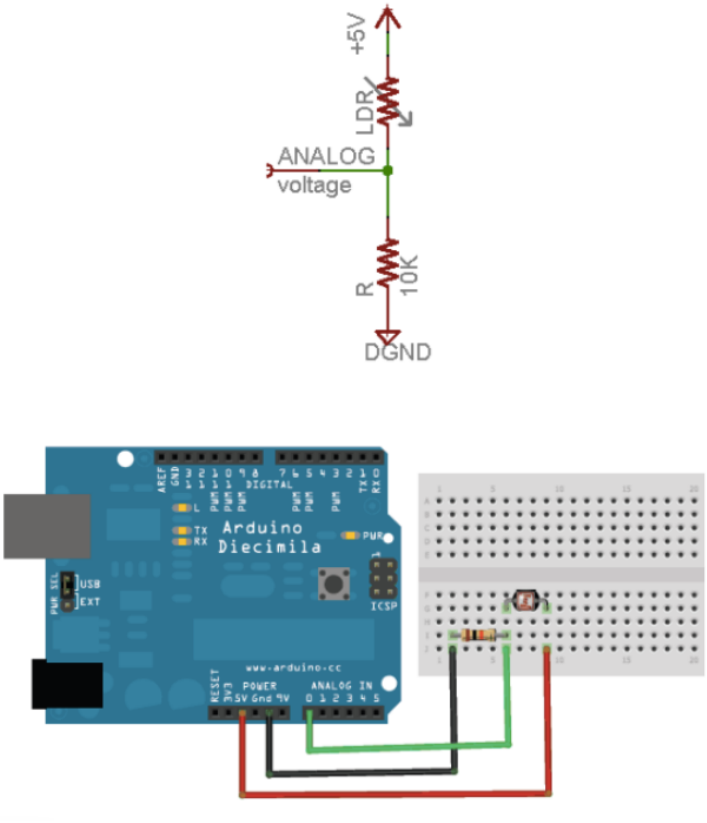


Figure 9.23: Ambient Light Circuit Schematic

The photoresistor was purchased on Adafruit, where application instructions are given. These application notes were used to decide on the value of the pull-down resistor as well as give an outline for Arduino communication. More information on the code used for this communication can be found in Section __.

9.3 System Configuration and Software

In this section we will now describe the implementation of the system microcontroller and cabling which effectively develop an architecture for the system to follow. First we start with a description of how we implemented the software. Following this is a discussion on the three proposed implementations by each team member and which was ultimately chosen. Finally we cover how communication is performed to execute animations and end with a discussion on the cabling layout of the system.

9.3.1 Software

Per our technical requirements, when an individual steps on any given step in the LED stair system, the light bar on the side of the step will illuminate, irrespective of whether an animation is active or not. Additionally, if any animations are playing, the moment an individual steps on a step all animations will immediately cease. Once no step interaction is detected, a cool-down of five seconds will occur before animations begin playing. This is to avoid any potential undefined behavior during edge scenarios such as someone kneeling down and waving their hand to repeatedly trip a sensor. After this cooldown period, if no interaction is detected, then the system will immediately begin playing animations. To preserve energy during moments of time where little to no interaction with the stairs will occur (such as during the night) we will design the system to keep track of how long no interaction has occurred. Similar to our five second cool down, if no sensors are tripped for 30 minutes then the LED stairs will enter a low power state. Here animations will only play for two minutes within 10 minute intervals. This way the steps can still be active and offer some form of animation at night while not being active 100% of the time. Below is a flow chart of the behavior described above:

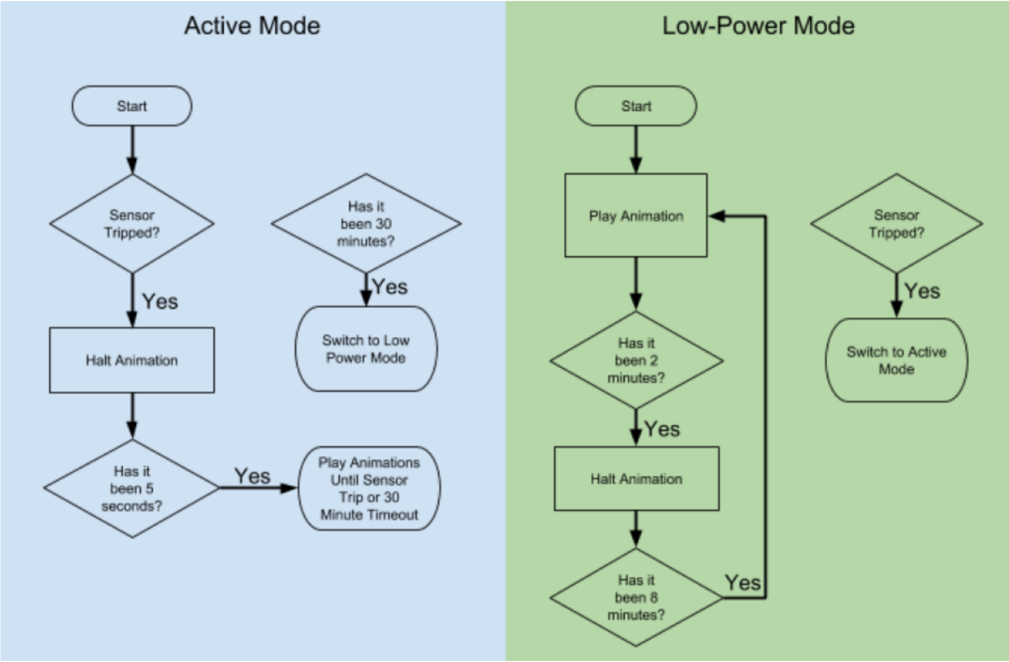


Figure 9.24: LED Stair System Behavior Flowchart

Initially we designed the sensors to communicate with the master MCU through a system of shift registers and serial communication. However, because of the realization that the system has no real use for knowing the state of each individual sensor, the design was reduced to a single communication line shared by all sensors and connected to a single GPIO on the MCU.

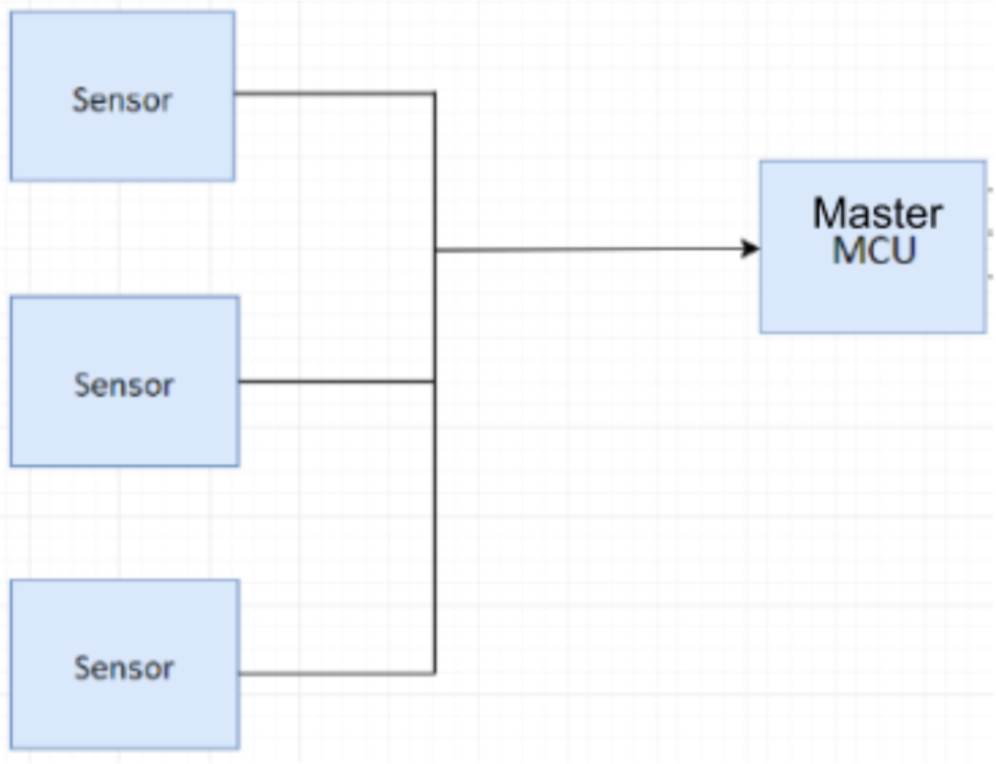


Figure 9.25: Sensor Interrupt Line Approach

This approach, visualized above, requires only one single communication line as opposed to the 3-5+ from the first design. Here, when a user trips the sensor on any given step, the sensor will activate the interrupt line with 5V DC. The corresponding GPIO on the MCU would then be configured as an interrupt line. When triggered, the MCU can switch to an interrupt service routine which serves to halt all animation. To immediately light the respective step, we can then connect the same digital output from each sensor directly to the light bar for simultaneous illumination. This modification is shown below in Figure 5. All latency that would have been introduced from using the information of which sensor was triggered and correspondingly which bar to light (if only the MCU was allowed to illuminate light bars) is then eliminated.

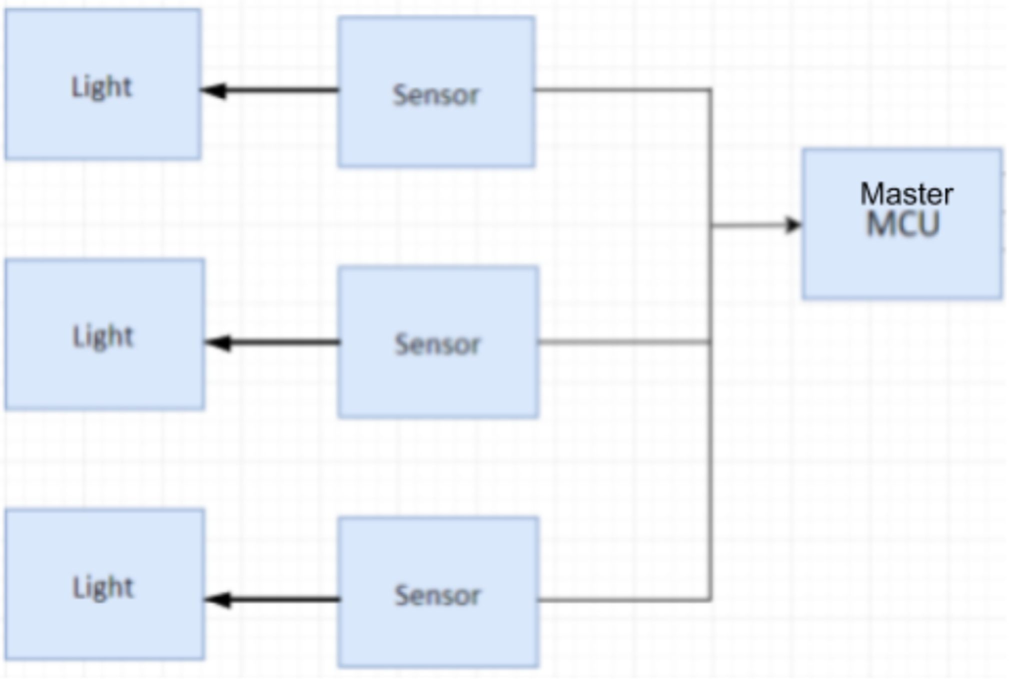


Figure 9.26: Sensor Interrupt Directly Triggering Steps to Light

This common interrupt line approach provides a scalability benefit as well. By sharing a communication line between all receivers to trigger animation halting we now only require 1 single shared connection as opposed to the 21 required with the previous shift-in register design. This approach thus satisfactorily meets our requirement of having system-wide interaction detection and contributes to an ease of scalability.

9.3.1.1 Animations

A this point we have a description that specifies how the software in the led stair system behaves to accomplish step lighting and autonomous animation. In this section we now cover the specific animations implemented.

9.3.2 Proposed System Architectures and What we Chose

When designing the microcontroller portion of the led stair system alternate designs were proposed by each team member. One which utilized a series of shift registers to control lighting on individual light bars, another which intended to utilize three cooperating microcontrollers to handle animation of 7 individual steps per microcontroller and effectively lighting all 21 steps, and the last which utilized master-slave layout to perform animation on individually addressable light bars.

We will start by describing the final chosen design: the individually addressable light bars and the cabling associated with it. Finally we will describe the other two implementations and why we ultimately individually addressable light bars were needed.

9.3.2.1 Microcontroller Configuration | Individually Addressable Light Bars

The final individually addressable configuration follows the general architecture below.

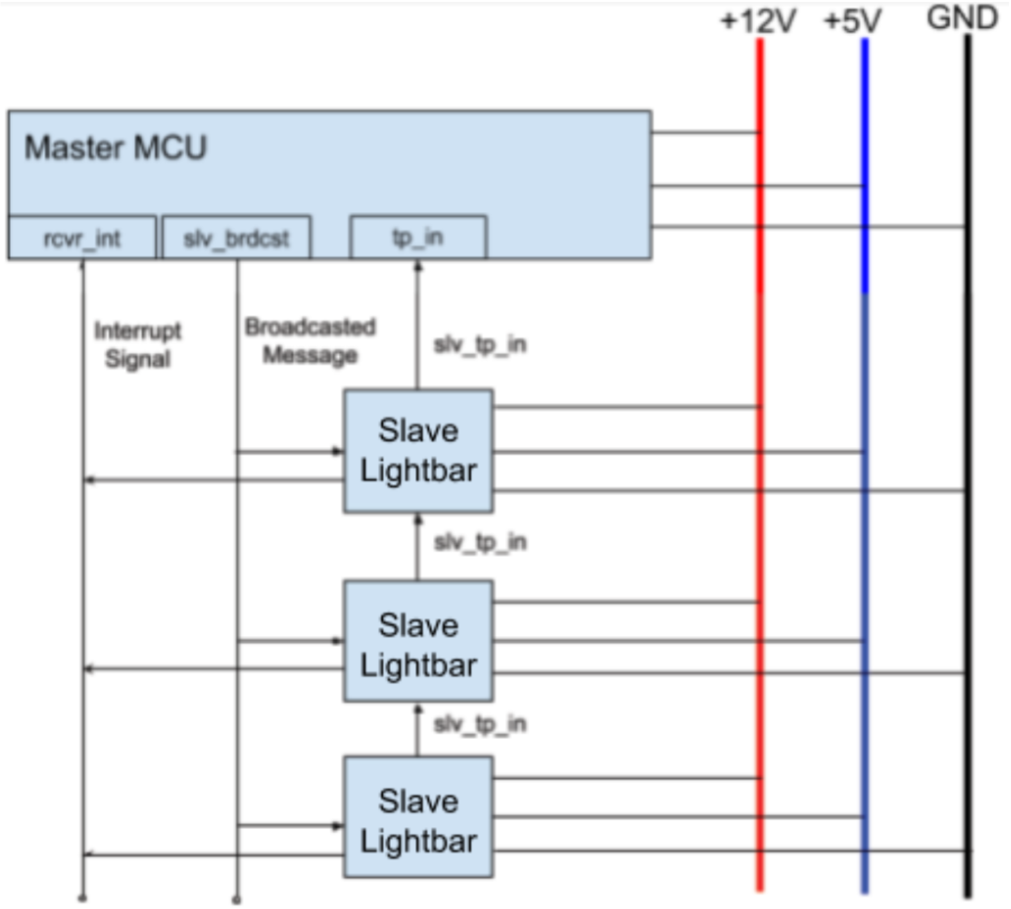


Figure 9.27: Individually Addressable Light Bars

Here we only require 5 total lines: 12V power, 5V power, ground, slave broadcast, and slave-topography-in. In this approach each light bar will have a slave microcontroller inside with data such as an ID that individually addresses it (similar to an ip address). When a specific light bar is to illuminate or perform a specific animation, the master MCU broadcasts a message onto the slave broadcast (slv_brdcst) line with information containing:

- The intended receiver of the message
- The type of illumination to be done

Because all light bars are connected to this broadcast line, they all see the same message. However, only the light bar whose ID matches the intended receiver of the message will respond accordingly.

But when performing location dependent animations how would the master MCU know the topography of the steps or the light bars? That's the goal of the slv_tp_in input line. Starting from the bottom-most slave: on first power-up the bottom-most slave sends a message with its ID to the second slave, simultaneously the second slave has also finished sending a similar message to the first and the first to the master MCU. The second slave holds onto the message for a short amount of time, then sends the message up to the first. As the master receives these messages, it creates an internal array of light bars with a counter that holds the number of slaves it is aware of. This process repeats until the master MCU ceases to receive identification messages.

This sequence of steps is shown again below:

1. System power up
2. Slaves send messages upward to the master MCU.
   (a) If a slave receives a message from a slave connected after it, hold the message for a short amount of time then transfer it upwards.
3. For every new message received, the master increments a counter and stores the counter value alongside the ID contained within the message.
4. When the master stops receiving messages, it stops awaiting for more and begins sending animation messages to individual light bars via the broadcast line.

Pros and Cons

This solution provides a few benefits over the initial two mentioned. These are:

- Reduction of the number of wires from 70+ to only 5
- Scalability is improved as all that is needed to scale the system is to simply add additional light bars to the chain. No programming or additional wiring is required
- PWM control is now done by each individual light bar itself

As nice as the reduction of data cables and improved scalability is, one negative factor comes to play with this system: the cost of each light bar requiring its own microcontroller. Fortunately, and as will be seen later in the project costs, the cost of the extra microcontrollers actually makes the system cheaper than if it had only one microcontroller and numerous communication wires.

9.3.2.2 Microcontroller Configuration | Alternate Approaches

With the LED stair's first and second designs numerous wires were needed to traverse from the system's microcontroller through several shift registers (or microcontrollers) and ultimately to 21 light bars. Two figures below represent the two other implementations for the led stair system. The first utilizing a system of shift registers, and the second, utilizing three microcontrollers.

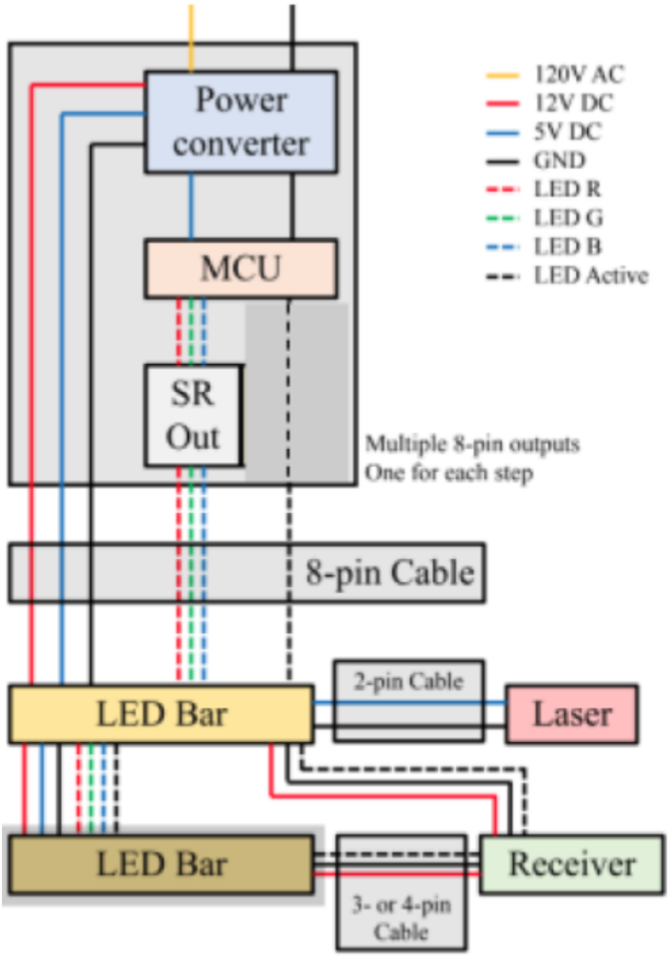


Figure 9.28: Simplified Shift Register Approach to LED Stair Cabling and Communication

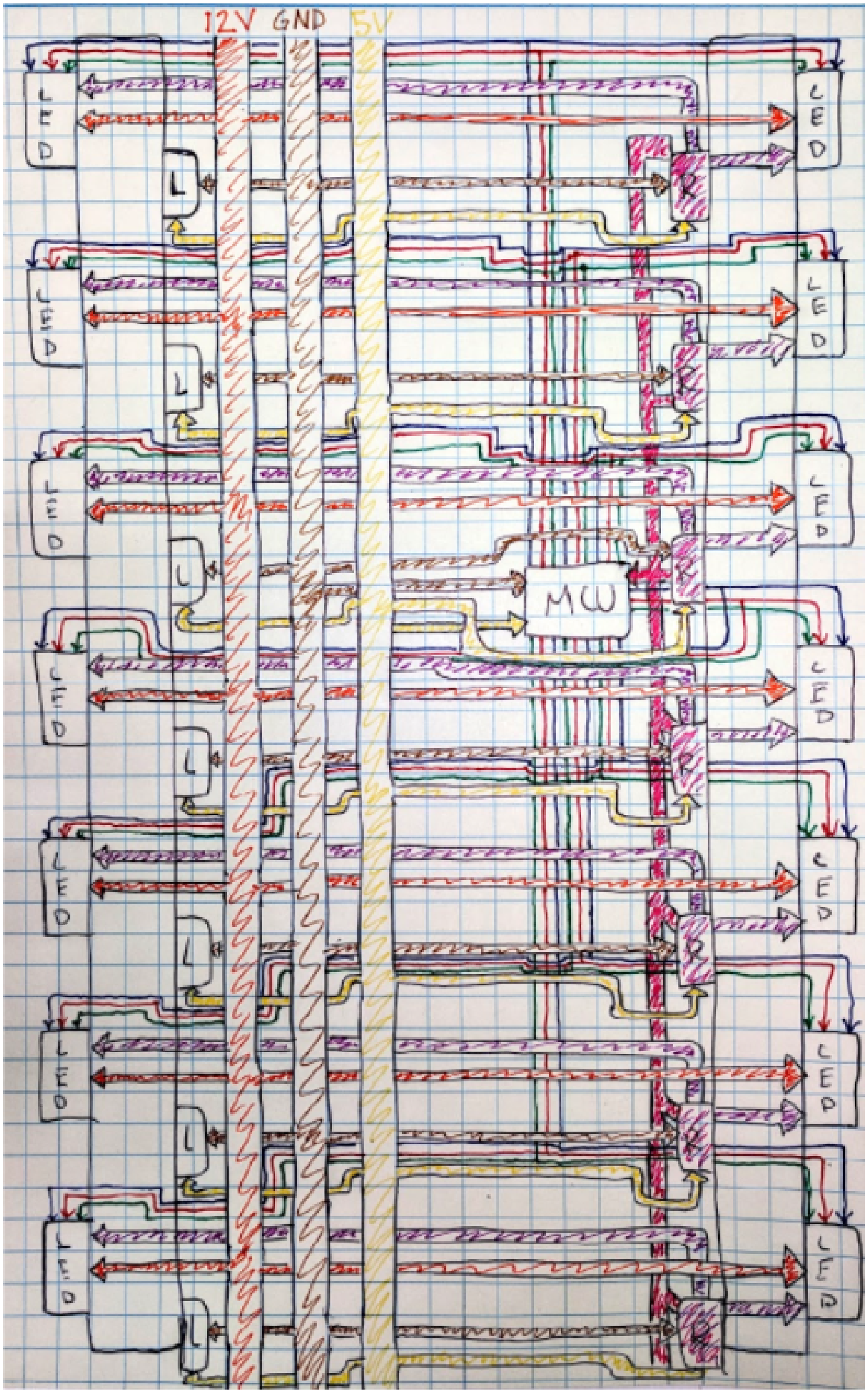


Figure 9.29: Three MCU Approach to LED Stair Cabling and Communication (one MCU shown)

If we assume only 21 light bars would be present in either system above (therefore no parallel light bars on the opposite side of a step), the first approach would require at least 75 wires. 63 total from the shift registers to individual LED strip color channels, 9 to connect shift registers to the MCU or to chain them together (if connected for in series serial communication), and 3 additional wires for 12V, 5V, and ground power lines. The second design would only marginally reduce this as we would require only 70 wires. 63 total rgb lines connecting microcontrollers to individual LED strip color channels, 4 serial communication lines for coordination between MCUs, and finally the 12V, 5V, and ground power lines.

Proposed Benefits of the Three MCU Design

The attempted improvement by this system is to avoid our initial design's requirement for having numerous long expanses of cabling going from the master MCU to several shift registers and ultimately individual color channels for every step. Additionally the goal of this implementation was to make the system scalable. To do so, simply add another microcontroller with connections for however many steps are required, and program the addition to the master microcontroller.

A few arguments can be made for why this system does not help resolve our initial issue of having numerous expanses of wire traveling down the steps:

1. As shown above we still require 63 RGB data lines. The addition of communication and power lines reduces the original 75 wire requirement to only 70.
2. Instead of reducing the amount of cabling needed, this approach primarily helps by shortening the length of the wires to be used by placing microcontrollers at specific locations along the steps.
3. By essentially replacing shift registers with microcontrollers, we eliminate a central microcontroller at the expense of code complexity (due to requiring microcontrollers to coordinate animation and behavior together).
4. In order to scale the system one would have to acquire additional microcontrollers and wire all additional light bars and their color channels to them.

In effect the second design offers little benefit to our original design in the first figure above. Even if microcontrollers can be purchased at a cost comparable to a shift register (the cheapest on Digi-Key for example being an 8-bit MCU IC costing 20c: Digi-Key part no. STM8S001J3M3TR-ND), unnecessary complexity is added in requiring multiple pre-programmed MCUs to utilize an interface to jointly decide what animations to play and coordinate their playback as a group.

Given the discussion above we restate problems found in the first and second designs:

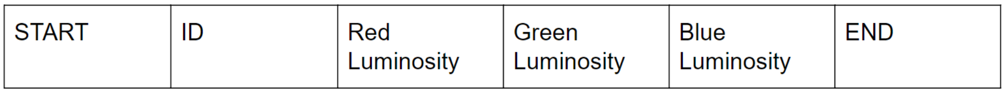
- Numerous amounts of cabling are required in order to control only 21 light bars
- This cabling must be expansive and reach as far from the 2nd floor down along the stairs until the first floor (approximately 8 meters in length)
- The system must be relatively easy to scale and require little to no programming or reconfiguration to do so

As the individually addressable configuration reduces the amount of wires down to 5 and allows for quick removal and addition of light bars, it satisfied our requirements and became a part of the final design.

9.3.3 Master-Slave Communication

As previously discussed communication between master and slave functions off message sending on a shared broadcast line that each slave light bar listens to actively. This communication utilizes the UART serial protocol. This was chosen instead of SPI and I2C as it did not require a slave-select line nor does it require a shared clock signal for process synchronization.

Each UART packet we send in the led stair system follows the format below.



START	ID	Red Luminosity	Green Luminosity	Blue Luminosity	END

Figure 9.30: LED Stair UART Serial Packet

Note that each field above is 8 bits wide meaning with its current design the led stairs can occupy a maximum of 256 slaves. When a packet is sent slaves begin listening as soon as they see the start byte. If the next byte read is the slave's ID/address, then it will record the next three bytes as the red, green, and blue luminosities respectively. If after the first 5 bytes are sent the slave sees the END byte, then it will know to store and acknowledge the recorded luminosities. Otherwise it will ignore the packet just as it would have if it did not address the slave. This is to avoid any potential issues with respect to incorrect animation on a a damaged or noisy broadcast line.

9.4 Light Bar Housing and Circuit Hardware

Having covered the design of the led stair system architecture, how step sensing is performed, and the software required to accomplish our specified functionality tasks, we now cover the design of the housing and pcb design of the slave light bars as well as the design of the master MCU pcb.

9.4.1 Master PCB

Below is a figure of the master PCB design for the led stairs.

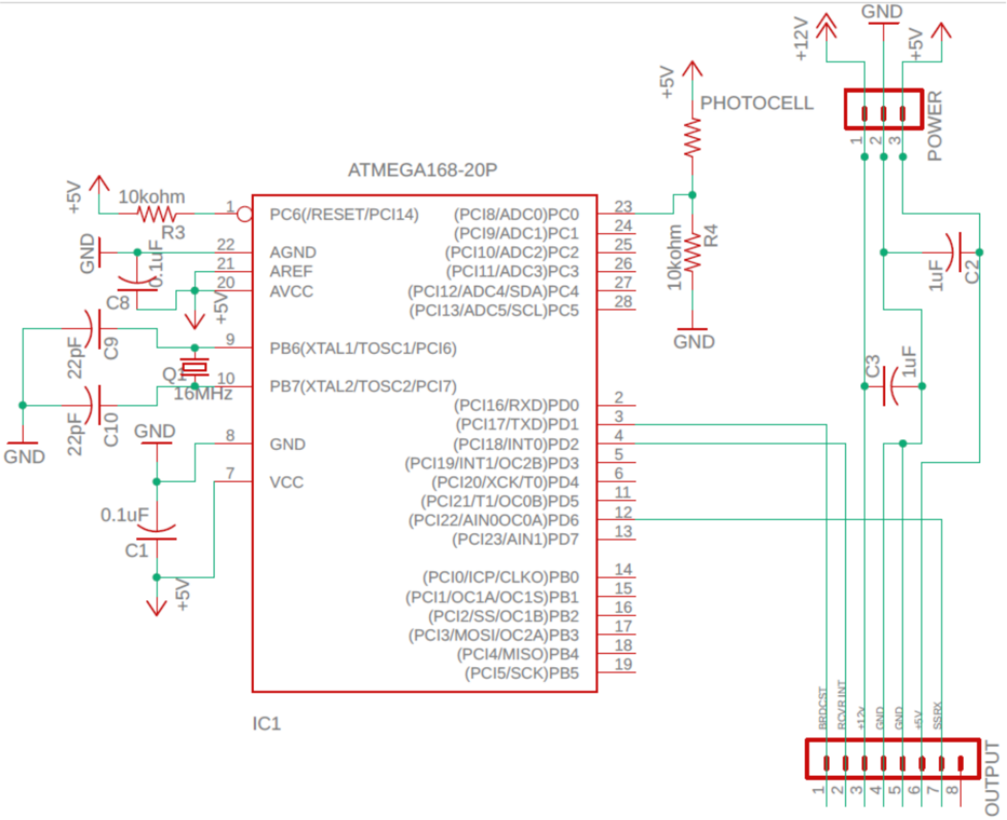


Figure 9.31: LED Stair Master PCB Schematic

Starting with the 10kohm resistor, 0.1uF and 22pF capacitors, and the 16MHz crystal on the left of the schematic, these components are critical for the microcontroller to run. The 16MHz crystal is used to provide the clock source for the MCU. Note that 2 22pF capacitors are connected from the crystal's terminals to ground to prevent any oscillating signal transients from affecting the rest of the system. The 10kohm resistor pulls the MCU's reset pin to a constant 5V to prevent random resets. The remaining 0.1uF bypass capacitors are used to provide power to the MCU from the 5V and GND rails. On the 3-pin power supply connector as well we also provide 2 1uF capacitors. The purpose of all these bypass capacitors is to dampen any potential noise that may propagate from the AC power lines to the power supply's DC outputs.

We also have the photocell which connects to the ADC0 pin on the MCU. This provides us with the ability to perform ambient light sensing for system brightness control.

We intended for the master microcontroller to serve as the "central" node of the led stair system. All communication and power lines needed should be provided by the master to the rest of the system. As such we see the broadcast, receiver interrupt, and soft serial communication lines being provided to an 8 pin output alongside the 12V, 5V, and GND power lines.

9.4.2 Slave PCB and LED Drivers

Due to the interest of centralizing the led strip within each light bar as discussed in the section below, the pcb for each light bars slave component was required to be split into two separate boards: one which held the slave microcontroller and provided any needed connections to drive the led strip and pass data to the next slave, and one to take the pwm signals from the slave MCU and drive the 12V led strips. Additionally each would need to be less than or equal to 5cm in length and have a maximum width of 5cm. A representation of the internal light bar layout is shown below.

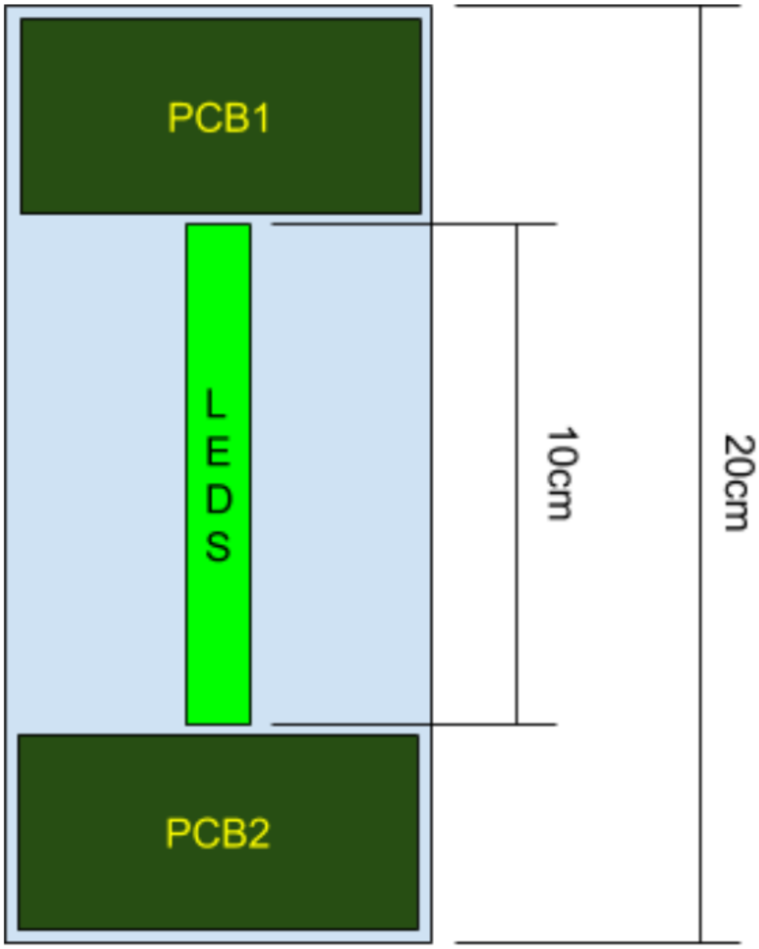


Figure 9.32: Internal Light Bar layout

Starting with the main slave PCB, its schematic is shown below.

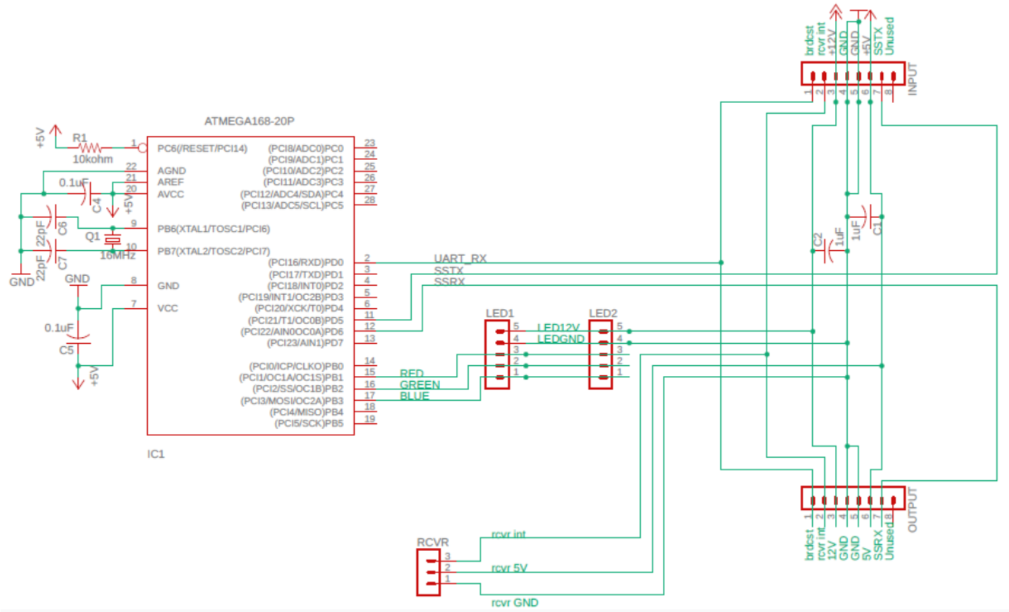


Figure 9.33: Slave PCB Schematic

Notice that on the left-most side of the schematic are the same exact components used for the master pcb. Again, these are required for the MCU to operate properly. The 1uF power line bypass capacitors are here as well. Notice that the all connections coming from the master or the top/previous light bar simply pass through the schematic from one 8-pin connector to the next. The slave microcontroller and the appropriate connections for the led strips and receiver simply latch onto the needed connection. For example, the master MCUs UART receiver line is attached to the broadcast line, the soft-serial TX and RX lines are also connected to the appropriate top and bottom connectors for power-up configuration.

The three pwm signals controlling the light bar's red, green, and blue color channels are passed to two duplicate led strip connectors. There are two here so that one may be connected to the led driver of the current light bar and the other to the led driver of a dummy light bar for the steps that require them.

Finally the trip sensor receiver connects directly to the receiver interrupt line so as to provide the interrupt signal for the master MCU.

Below is the schematic for each light bar's led driver.

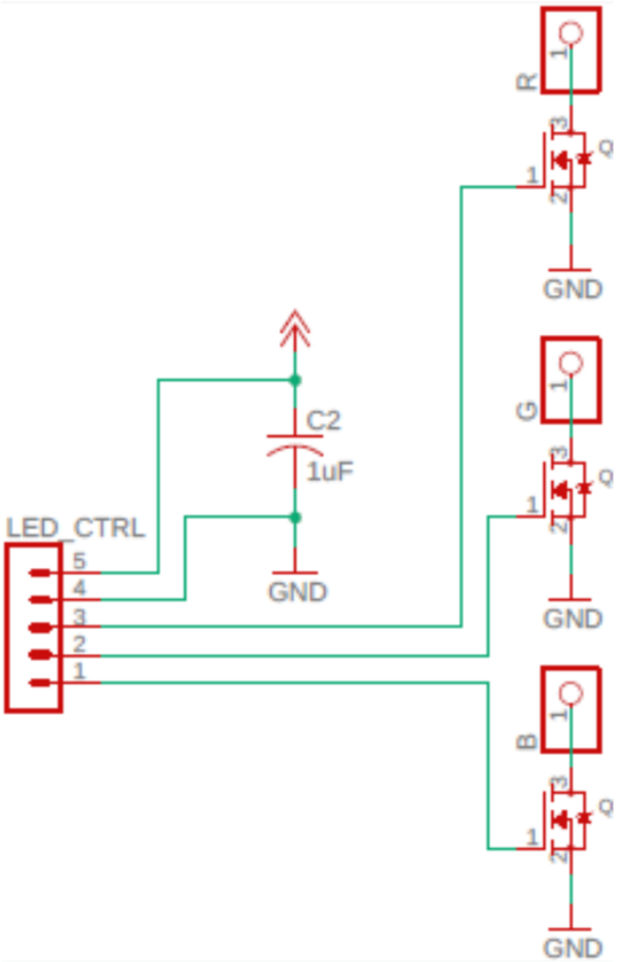


Figure 9.34: LED Driver PCB Schematic

Here the 3 rgb pwm signals are passed from the slave PCB to individual NMOS transistors which connect the led strip's color channels to ground. As the pwm wave controls the gate voltage of each transistor, the corresponding color channel on the led strip turns on and off accordingly. Finally we also have a 1uF bypass capacitor for the 12V power line.

9.4.3 Light Bar Housing Design

The staircase on which the project was installed can be split into three sections:

- Section one starts from the first floor then proceeds with two stair steps followed by a platform.
  - This section is not visible by the outside.
- Section two is 11 stair steps that are sandwiched between 2 platforms.
  - This section has one side visible to the outside.
- Top section starts with a platform, then 8 stair steps leading to the 2nd floor.

– This section is not visible by the outside.

The side views of each step can be seen on sections 1, 2, and 3 for people viewing from the inside, but as mentioned above, only section 2 can be viewed from the outside. This means that we would need to implement 32 separate displays, one per viewable side of step. This section will go through all the designs and decisions made that led to the final product shown in Figure A.

9.4.3.1 Design Process

When the project was first established, the design of these bars was very minimal. Not much thought had been put into them other than the shape. A triangular shape was briefly considered, but the team decided to instead move forward with horizontal bars. The first design was intended to mimic the dimensions of the steps, so the steps were studied before moving forward with our design. Measurements and visuals can be seen in the figures below:

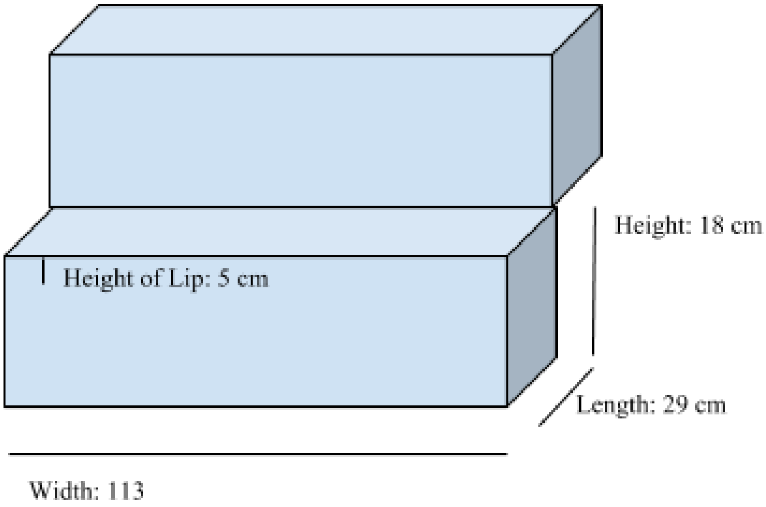


Figure 9.35: Step Measurements

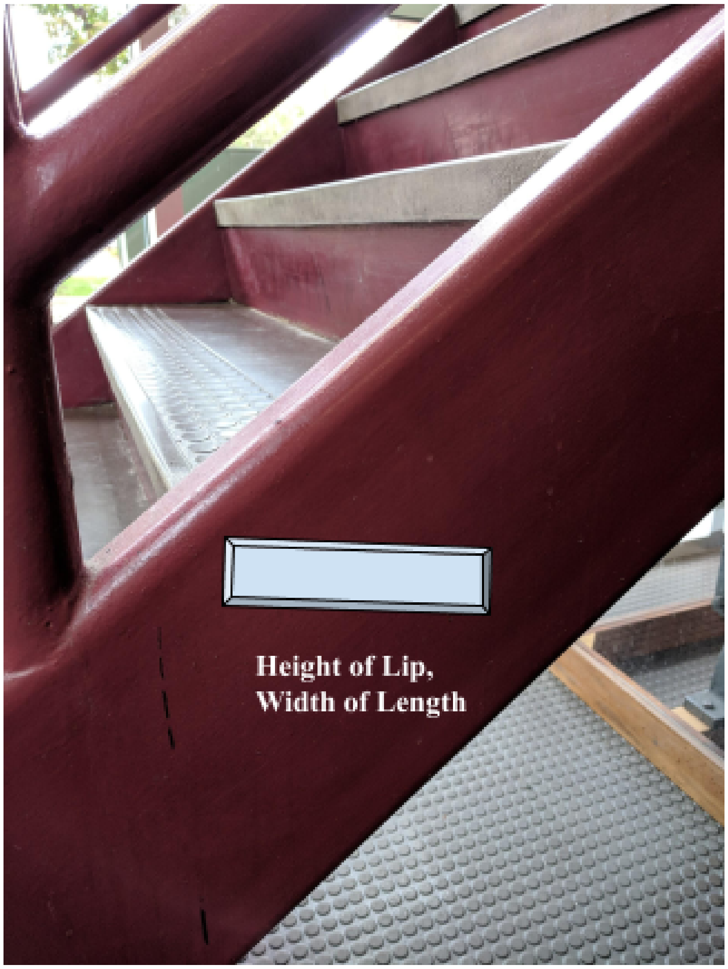


Figure 9.36: Visual of First Light Bar Design

The first idea was to house each display in a wooden box with a sanded acrylic side to diffuse the light emitting from the LEDs. Given that each display would have a maximum width and height of 29 cm and 7 cm, it was decided that the displays should hold around 6 LEDs (6-8 depending on cut intervals of the strips). Specifications for the LEDs are listed in the table below.

RGB	Easier to buy as whole, and cheaper (less would be wasted and wouldn't have to pay for shipping of various colors). Easier to customize.
Long Life Expectancy	This will weigh into our decision, as maintenance of these stairs is one of our top priorities.
5050 Chip Size	The 5050 is one of the brightest chips available, so we decided to use this and adjust brightness as needed.
Low Cost	This is not our top concern, but will still weigh into our decision. If two parts equally match with our requirements, this will be a deciding factor.

Table 9.1: Original LED Specifications

Keeping these specifications in mind, several components were looked at for implementation. All components considered can be found in the table below.

Site	Color(s)	Total Cost	Meters	Width	Cut Intervals	Lumens	Power Rating	Voltage	Current	Chip Type
Digikey	RGB	$120.40	5	1cm	10cm	18 lm/W Red, 69 lm/W Green, 6 lm/W Blue	72 W	24V	3 A	N/A
Adafruit	RGB	$80.00	5	1.05cm	10cm	N/A	N/A	12V	60mA	5050
Amazon	RGB	$16.98	5	N/A	N/A	N/A	N/A	12V	N/A	3528
Ebay	RGB	$9.99	5	N/A	N/A	N/A	N/A	12V	N/A	5050
Ebay	RGB	$6.79	5	N/A	N/A	10-12lm/led	0.24w/led; 72w/300leds	12V	5~6A	5050
LED-Lights	RGBW	$22.79	5	1.24cm	~8cm (3 LEDs)	N/A	14.4W/Meter	24V	N/A	5050
Amazon	RGB	$14.65	5	1.02cm	3 LEDs	630 lumen/meter	72W	12V	N/A	5050

Table 9.2: Component Research

Since the LED strip selection was limited in variation, the primary concern was cost. For multiple electronic component distributors, the strips seemed overpriced. Therefore, we looked to Amazon and Ebay to see what they had to offer. The cost of LEDs on these sites were almost 10% of the prices advertised on the electronic distributor's sites. With most products, this raises a quality concern, so a cheap roll of RGB LEDs was ordered to test the quality of a less-trusted product from Amazon. The row highlighted in Table B shows the ordered component, RGB LED Strip from Zollan.

LED Strips were chosen over individual LED chips for maintenance reasons, if a strip becomes defective one would simply just replace the strip. We planned to keep maintenance of these stair as a high priority to ensure a long lifespan for this project. Maintenance was also a heavy factor when determining housing and wiring for the LEDs. Below is a simple drawing of our housing design.

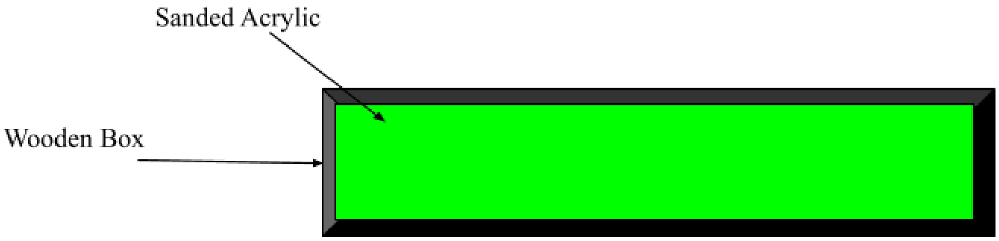


Figure 9.37: Minimal First Design of Light Bars

With this housing design, we would require about 192 LEDs for a 6 LED per display design and a maximum of 256 for a design of 8 LEDs per display.

Next, we wanted to explore deeper into what materials would be best for the design of the housing. As mentioned previously and shown in Figure D, the first materials considered were acrylic and wood, but this was just a placeholder until research was completed.

To better perform cost analysis, we had to define the dimensions of the Light Bars. This was dependent on the dimensions of the stairs as well as the cut intervals of the the chosen LED strips. The RGB LED Strip from Zollan had cut intervals of 3 LEDs, so we chose to use 6 total LEDs per Light Bar (approximately 10 cm in length and 1 cm in width).

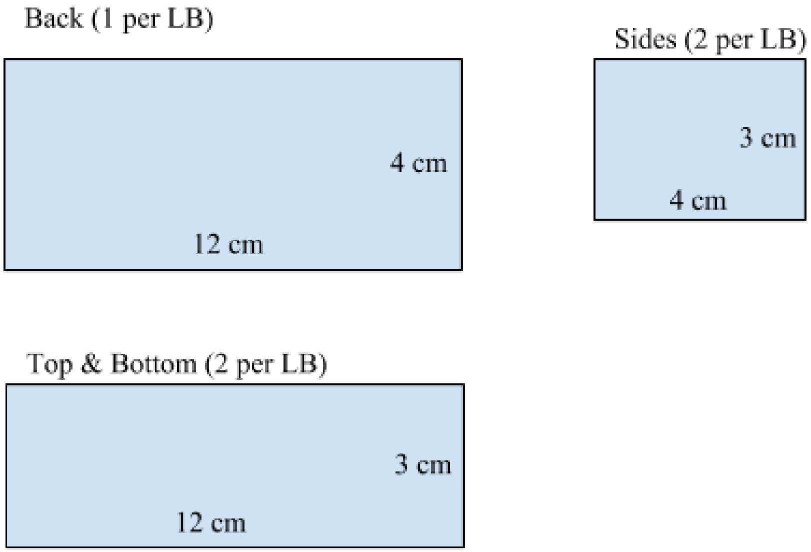


Figure 9.38: Original Light Bar Dimensions

3D Printing

For the examples below, we used a thickness of 0.2 cm, this gives an approximate volume of 28.8cm3needed. Foisie Innovation Studio, where we planned to perform the prints, offers Black PLA and Black Polycarbonate as printing materials.

- PLA has a density of 1.25g/cm3, and is priced at $0.03/g. Therefore, it would cost $1.08 per enclosure, totalling at approximately at $34.56.
- Polycarbonate has a density of approximately 1.22g/cm3, and is priced at $0.05/g. Therefore, it would cost $1.76 per enclosure, totalling at $56.23.

The 3D Printing option was first thought to be off the table due to cost, but stayed in the running since we planned to use injection molding for the final builds. The prototypes would still be 3D printed, but to lower cost, we planned to send the final design to an injection molding company.

Plywood (Home Depot)

Plywood of 1/8' thickness and 1/4' thickness were looked into. For all five segments of our housing, we would require 144cm2 per light bar, and 0.4608m2 for our full system, or to be safe, 0.5m2.

- Eucalyptus White Hardboard gives over 4 times the amount needed, however is the most aesthetically pleasing and still manages to be almost 10% the cost of our 3D printing options. For 4ft by 8ft of this plywood it would cost us $10.98. This board is 1/8' thick, and does not come pre-cut.

Figure 9.39: Eucalyptus White Hardboard

- Hardboard Tempered Wood gives an option at half the size of the Eucalyptus, 2ft by 4ft for $5.45. Just like the eucalyptus board, this board is 1/8' thick, and does not come pre-cut.

Figure 9.40: Hardboard Tempered Wood

- The best option found here was the PureBond Cherry Plywood Project Panel 2ft by 2ft, having the least amount wasted, priced at $8.00. This board is 1/4' thick and includes free custom cutting by Home Depot.

Figure 9.41: PureBond Cherry Plywood Project Panel

Metal

Another option to consider was metal, with help from the shop in the ECE department, we could build our light bar housing using metal sheets.

- 24 in. x 36 in. 26-Gauge Zinc Metal Sheet would be perfect for shaping and cutting, priced at $19.97 from Home Depot. One sheet of this would give us enough for the project.

Figure 9.42: 26-Gauge Zinc Metal Sheet

- 24 in. x 24 in. Zinc-Plated 26-Gauge Sheet Metal would be perfect for shaping and cutting, priced at $10.58 from Home Depot. One of these steel sheets would give us just enough for the project.

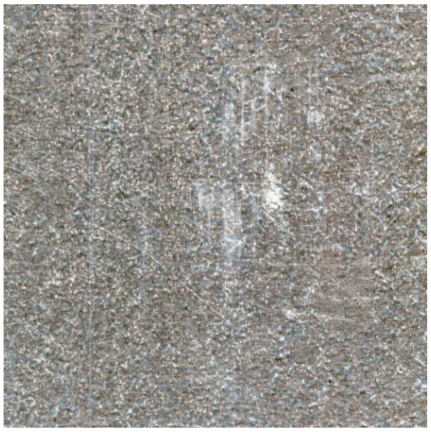


Figure 9.43: 26-Gauge Steel Metal Sheet

- IMPERIAL 24-in x 3-ft Galvanized Steel Sheet Metal is 30-Gauge, therefore it is thinner than our other options. Online reviews says easy to bend (or dent, might be a concern) and the size would give enough for the full project. The sheet costs $10.48.

Figure 9.44: 30-Gauge Steel Metal Sheet

Premade Housing

- EMachineShop offers a customizable metal box, but at very high prices. Therefore, if deciding to use metal as the material for the Light Bars, it would be best to build ourselves.
- Digikey had boxes that would fit well for our design, however the housing would be around the same cost as 3D printing without our exact measurements.
- Mouser had cheaper options, but no box came close to the ideal measurements.

We decided to veer away from ordering from these sites because would would have no flexibility on the design of the box. In addition, if we cut the thickness in half for 3D printing, it would be cheaper than any of these options.

For a summary of the options stated above, please refer to the table below.

Housing Type	Details	Total Cost
3D Printed Black PLA	0.5cm, precise cuts and full enclosure build	$34.56
3D Printed Black Polycarbonate	0.5cm, precise cuts and full enclosure build	$56.23
Eucalyptus White Hardboard	1/8’, 4ft by 8ft board (cut and assembly needed)	*$10.98
Hardboard Tempered Wood	1/8’, 2ft by 4ft board (assembly needed)	*$5.45
PureBond Cherry Plywood Project Panel	1/4’, free custom cut of 2ft by 2ft board (assembly needed)	*$8.00
Zinc Metal Sheet	26 Gauge, 24 by 36 in sheet (cut and assembly needed)	$19.97
Steel Metal Sheet	26 Gauge, 24 by 24 in sheet (cut and assembly needed)	$10.58

Table 9.3: Light Bar Materials Considered. An asterisk in Total Cost means there will be additional costs, e.g. nails, screws, hot glue, etc.

The cost of 3D printing could be lowered by thinning the thickness, however even at half the cost it would not compare to the costs of the wood. Premade options were not being considered due to the reasons given above. Considering the prices of our options, wood was thought to be the best option for this project. Specifically, PureBond Plywood, largely because it would be professionally cut, but also, when adding the diffusion side to the build, a thicker cut may be easier the secure the acrylic slide in the box. However, this was not our final decision due to durability concerns.

After presenting these options to the customer, Professor McNeill, it was decided that, despite the cost, plastic would be the best material for this design. The lifespan of this product is far more important than the initial costs of material, therefore the team moved forward with the new decision. For a better visual of the dimensions of the bars, we built a first prototype of the Light Bar using duct tape, wax paper, and cardboard.

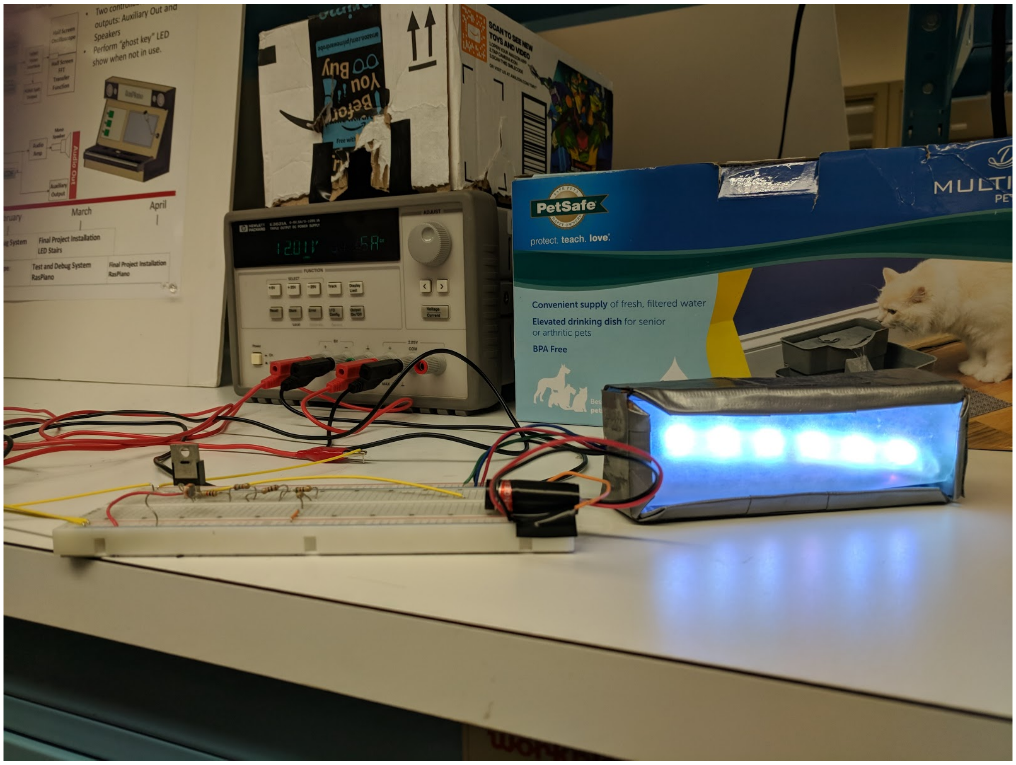

PetSafe
protect. teach. love.
Convenient supply of fresh, filtered water
Elevated drinking dish for senior or arthritic pets
BPA Free
MULTI

Figure 9.45: Original Light Bar Model

The first detailed Light Bar digital model is in the figures below show different angles. The materials had not been included in the SolidWorks design yet, so note that the smooth rectangle is clear as it represents the acrylic sheet. Also, since this was only the first design, many other ideas on how the acrylic and box should be constructed were still being considered.

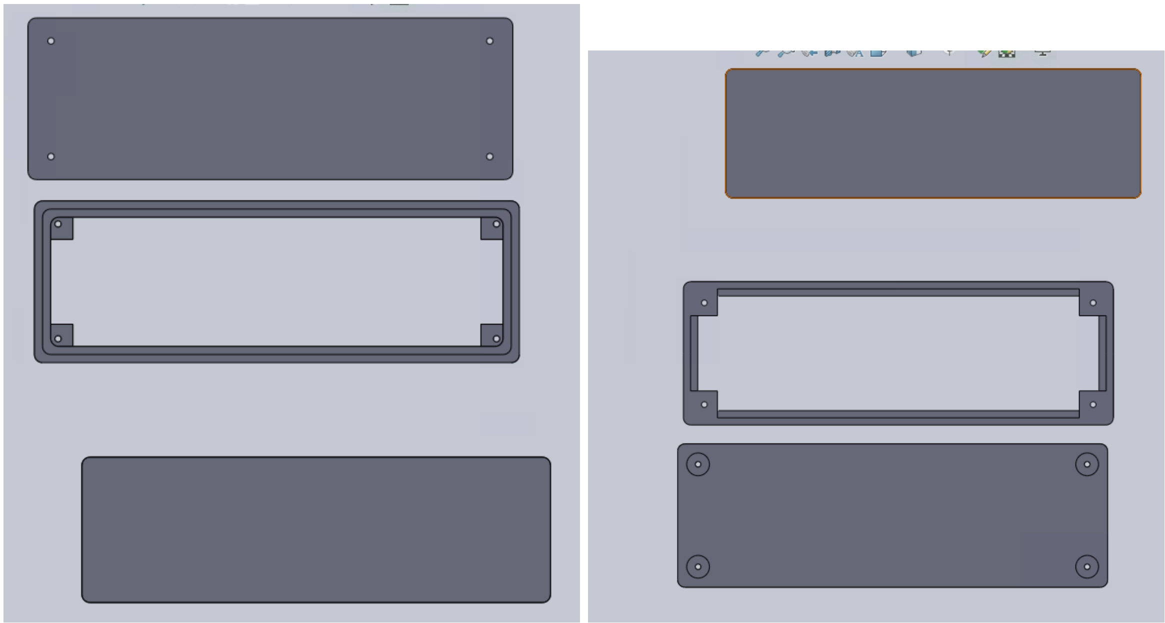


Figure 9.46: Top and bottom View of First Digital Light Bar Design

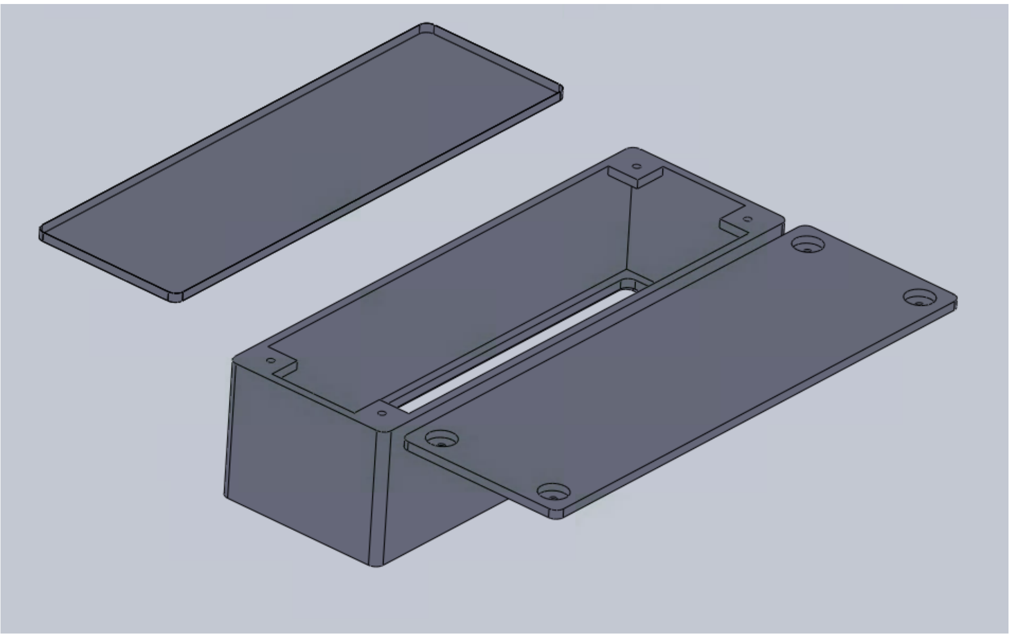
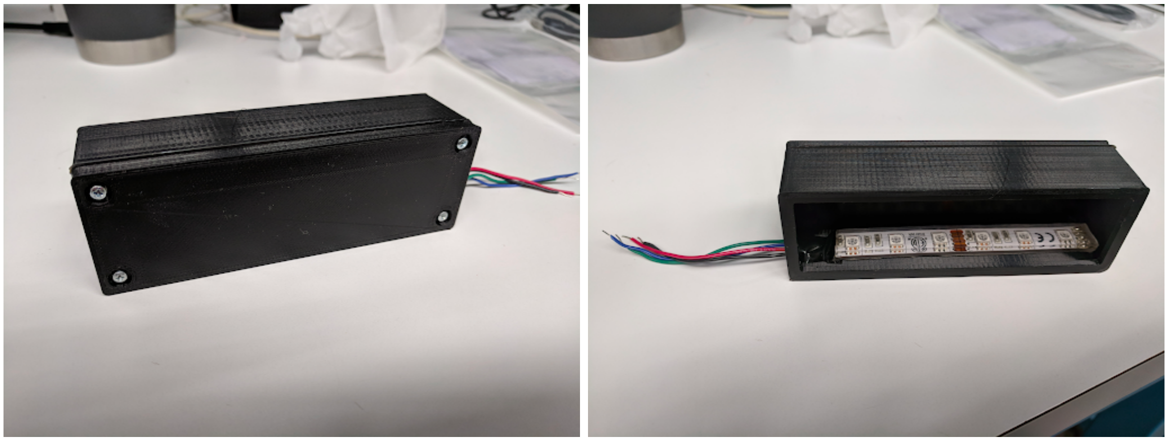


Figure 9.47: Isometric View of First Digital Light Bar Design

The light bar design was finalized in Solidworks and examined by Shadi Kalat. After the design of the box was critiqued, the final edits were made and submitted to Foisie Makerspace to be 3D printed. Three lids and three boxes were sent at a total of $4.23, this was cheaper than expected due to the 3D printer's hexagonal layering method.



(a) Light Bar Prototype 2 (Back)

(b) Light Bar Prototype 1 (Front)

Figure 9.48: Light Bar Prototypes

These boxes were used in our first system prototype, but one concern that Shadi Kalat had mentioned stayed with us. This concern was about the screw holes of the housing, with the materials offered for 3D printing, these will wear over time. Therefore, years later, the screws may start falling out due to worn-down screw holes. To fix this, the preferred material would be acrylic. This meant using the laser cutter instead of a 3D printer, so a new SolidWorks design was made. The Light Bar was designed so that 1/4' thick pieces could fit together and binded using acrylic glue. The back would be screwed in and elevated on the inside of the box, this elevated part would have a section fit for the PCB of the slave. The new design is shown in the figure below.

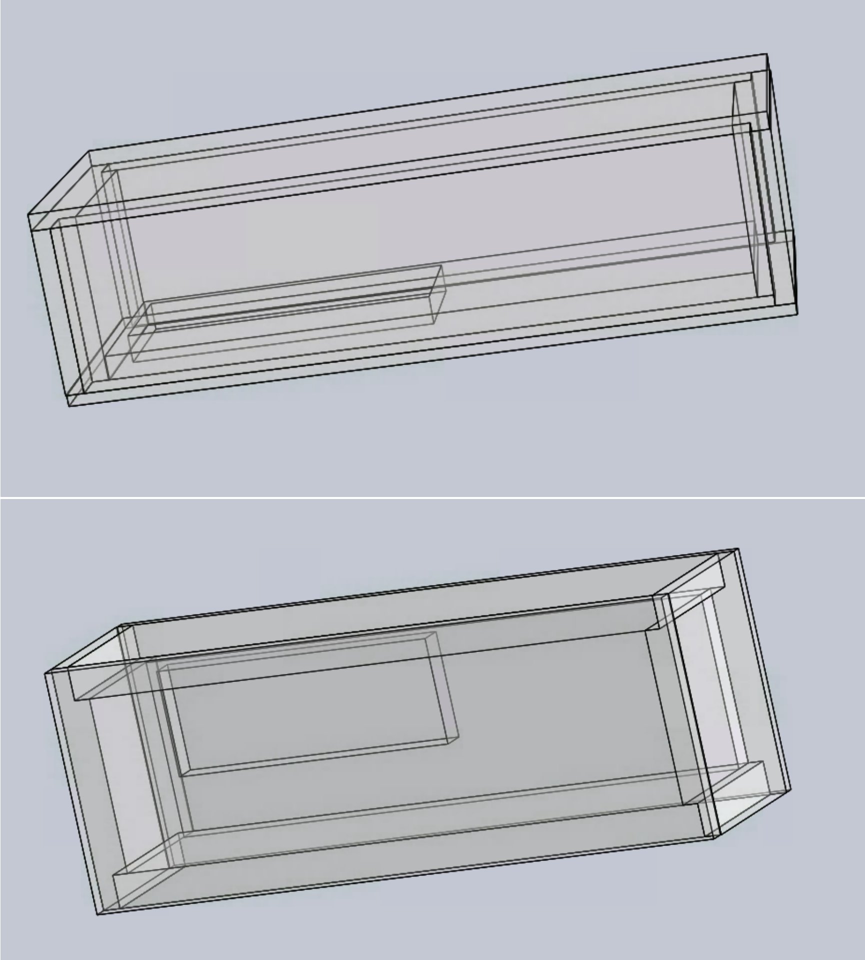


Figure 9.49: First All-Acrylic Light Bar Design

We planned to add triangular slits into the backing of the Light Bar once the height of the PCBs were determined to hold them in. Additionally, holes would need to be added to the design for the wires. Another change in design was to the color of the Light Bars, instead of them being black with one clear side to emit the light, the whole box would be made of clear sanded acrylic. This would shine light through every side of the box. A 12' by 12' clear acrylic sheet 1/4' thick was ordered to build a first prototype of the new design.

Although approved by Professor O'Rourke, the team was reluctant to drill holes into the stairs for installment. However, if holes were not going to be drilled, then we would need to consider the wiring system within our design. The horizontal bars would need vertical wires coming from above and below them for best installation. The thought of having vertical wiring with horizontal Light Bars seemed extremely unattractive. To avoid this, we began to reconsider the shape of the bars. If we had the bars horizontal, it would no longer clash with the wiring, but it was uncertain if this would look awkward on the stairs. Then the idea was born to make them look like components in a schematic, and since these were light bars, to etch the LED symbol on the bars. A concept sketch for this new idea was shown in Figure 10.3 of Design Process Section.

9.4.3.2 Final Build

Within the first acrylic design model, we had a section to insert the PCB. We planned to use triangle inserts to lock the PCB in place; however, this design was changed. Instead, we wanted to screw the PCB in place for ease of maintenance and maximum security. The PCB design was rearrange to reflect this new decision, it now had to include screw holes, just as the model. The final Light Bar model is shown in the figure below.

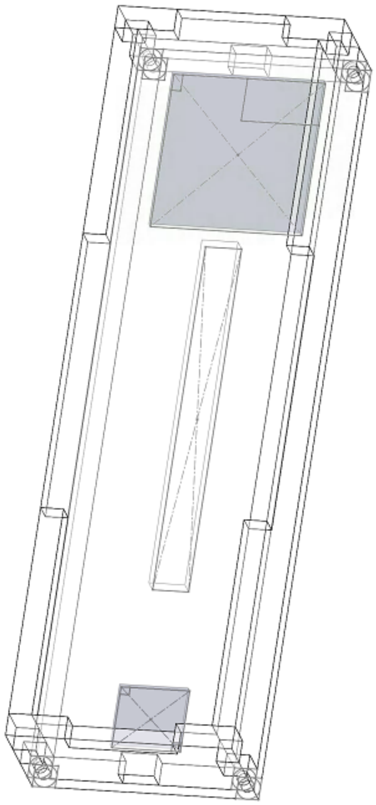


Figure 9.50: Final Acrylic Light Bar 3D Model

To include our LED symbol design, we plan to sand blast the Light Bars, but for demonstration, we manually sanded the bars. This was done with masking tape and sandpaper, completing three different design ideas on five separate bars.

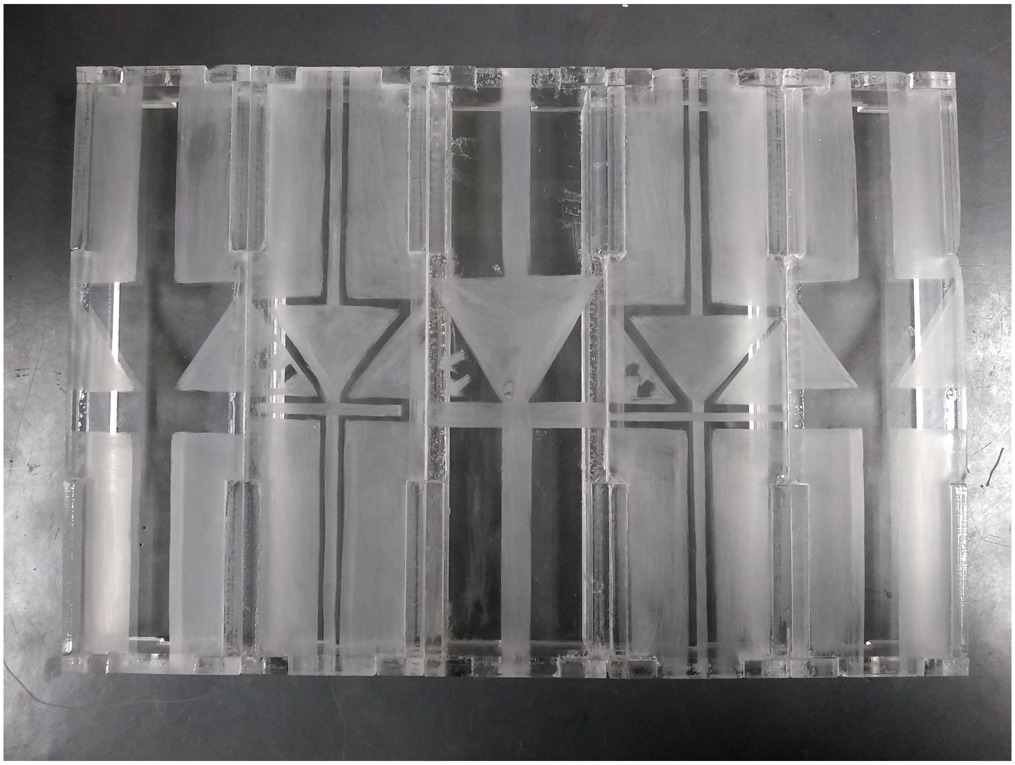


Figure 9.51: Manually Sanded Light Bars

Chapter 10

Final Design

Before ending the chapter we will now cover the changes made to implement the final prototype, review the power and wiring the system, and detail the processes of installing the led stair system and modifying its code base.

10.1 Connectivity of Subsystems

With the end of D-term approaching our team did not have the time available to perform a final installation of the led stair system on the intended staircase of Atwater Kent. Simultaneously we were also concerned with further completion of the Raspiano project which still had features which required implementation. As we approached an increasing time constraint it was important for us to decide whether we would perform the final led stair installation and risk having a partially completed Raspiano or to complete the Raspiano and risk not installing the led stairs. Ultimately we decided to not install the led stair system and instead construct a project demo that would allow us to showcase the functional components of the project and the intended final appearance.

The demo utilizes the same exact components that were to be used in the final installation with the exception that only three to five steps would be implemented. Below is a figure of this project demo.

FIGURE LED Stair Project Demo

As shown above there is one master microcontroller and five slave devices forming five "steps.' These five steps each have a infra-red (ir) receiver, transmitter, slave microcontroller, and LED strip driver. Additionally each step still has the same light bar designed for installation. Powering the display is also the same power supply that would have been used. In essence there is no difference between the final installation and the demo except for the demo having only 5 of the intended 21 steps in addition to being constructed on a panel as opposed to a physical staircase.

As we were not able to add the topography identification feature to the master and slave devices the only change necessary to this system was the software. Here we modified the topography to have only 5 steps and removed the addresses for the other 16 slave devices. Fortunately none of the changes discussed above affected the functionality of the demo. Here we implemented a successful replication of how the final led stair system would have looked and functioned.

10.2 Power and Signals

10.2.1 Power Consumption

Below are two tables of the power-consumption per component for the master device and each step within the led stair system.

Per the RD-125A's (the chosen power supply for our led stair build) 7.7A maximum can be drawn from the 12V and 5V rails individually and supports a total maximum power consumption of 130.9W. The master device consumes only 13.792mA from the 5V rail leaving 7.686A available for use. With respect to power consumption by each step, our limiting factor is the

Components (Max Values)	**Voltage**	**Current**	**Power**
MCU	5 V	12 mA	60 mW
Ambient Light Sensor	5 V	1.792 mA	8.96 mW
Total 5V Rail Consumption	5 V	13.792 mA	68.96 mW

Table 10.1: Power-Consumption of Master Device

Components (Max Values)	**Voltage**	**Current**	**Power**
Light Bar (LED Strip)	12 V	120 mA	1.44 W
IR Transmitter	5 V	130 mA	650 mW
IR Receiver	5 V	20 mA	100 mW
MCU	5 V	12 mA	60 mW
Total 12 V Consumption	12 V	120 mA	1.44 W
Total 5 V Consumption	5 V	282 mA	810 mW

Table 10.2: Power-Consumption Per Step (No Dummy Light Bar)

maximum potential power consumption by each led strip as it draws more power than all other step hardware. With 7.7A available on the 12V rail our chosen power supply can support a theoretical maximum of 42 light bars/steps. This yields a current draw of 7.686A from the 12V power rail and 937.792mA from the 5V power rail or 96.92W of power total. Note that these power calculations assume no step has a dummy light bar attached to it. If so we must account for the the additional 183mA maximum additional current draw from each.

10.2.2 Wiring and Device Cabling

For the system wiring we review figure x1. Each cable connecting either the master-device to a slave or slave to another slave will be the same. Below is a figure demonstrating the mapping of each signal to each pin on a female (cable-wise) connector.

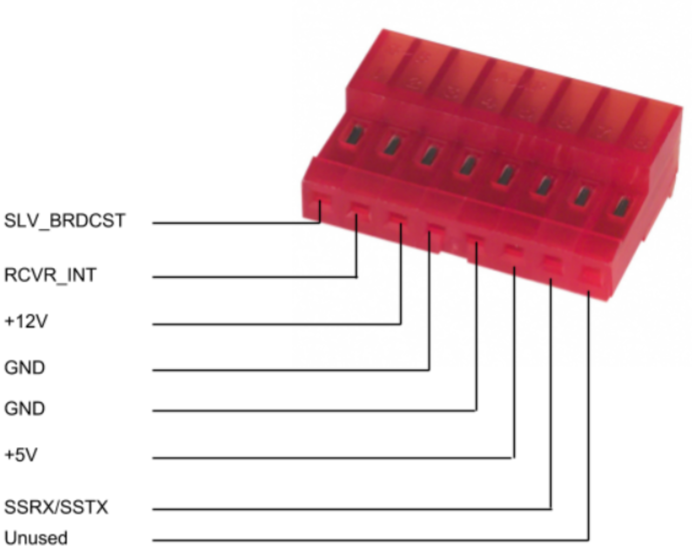


Figure 10.1: Female Connector Signal Mapping

Similarly below we have the pin mapping for each male (pcb-wise) connector.

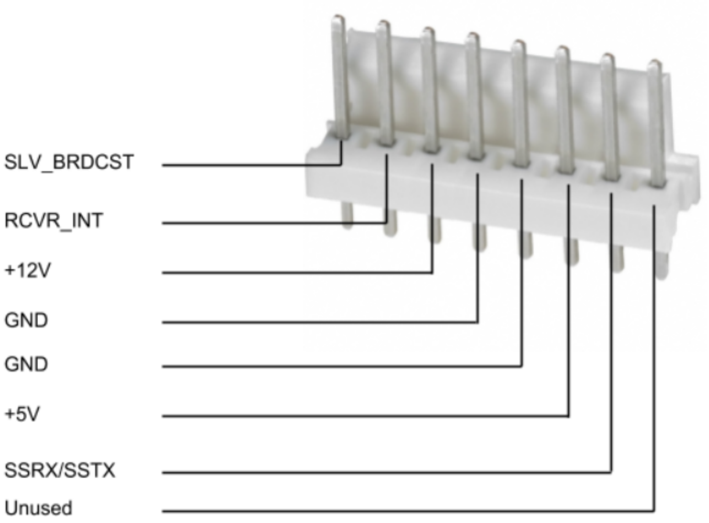


Figure 10.2: Male Connector Signal Mapping

10.3 Acquiring and Modifying LED Stair Code

This section aims to specify how to acquire the code that runs the led stair system. We start with obtaining any/all prerequisite software. Following this are the steps to compile and flash the code onto Arduino devices. The document ends with suggestions and best practices for users interested in modifying the code base to alter the led stair system's behavior or replicate it onto other devices.

10.3.1 Prerequisites

Before the led stair code base can modified or uploaded to an Arduino device, the Arduino IDE must first be installed, followed by a download of the code.

10.3.2 Obtaining the Arduino IDE

To compile and flash any code onto an Arduino one must first obtain the Arduino Integrated Development Environment (IDE). Below are steps to acquire this.

First visit www.arduino.cc A webpage similar to that below should result.

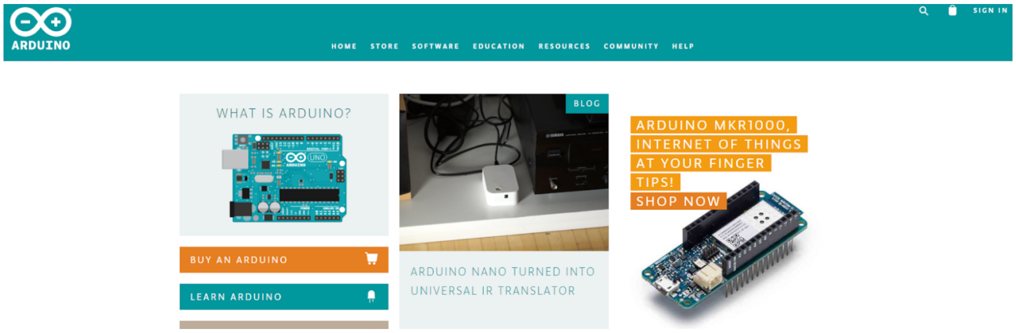


Figure 10.3: Arduino Main Webpage

At the top of the page are multiple tabs "Home," "Store," "Software," etc. Select Software then click Downloads in the drop down that appears.

A new webpage will load displaying available software downloads and resources. On this page find the Arduino IDE download section which may appear as below.

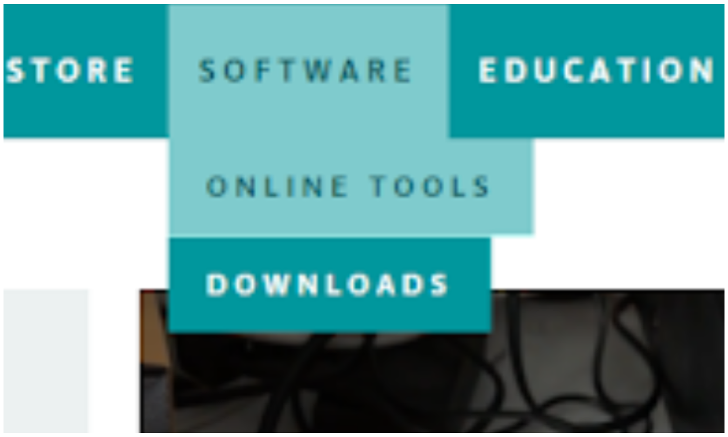
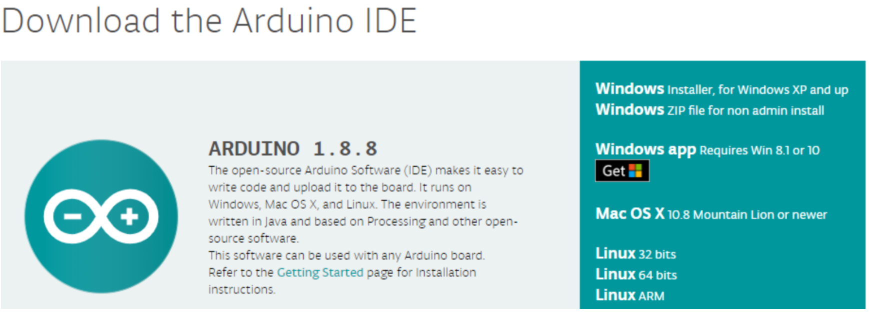


Figure 10.4: Software-Downloads Tab



Download the Arduino IDE

ARDUINO 1.8.8

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board.
Refer to the Getting Started page for Installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

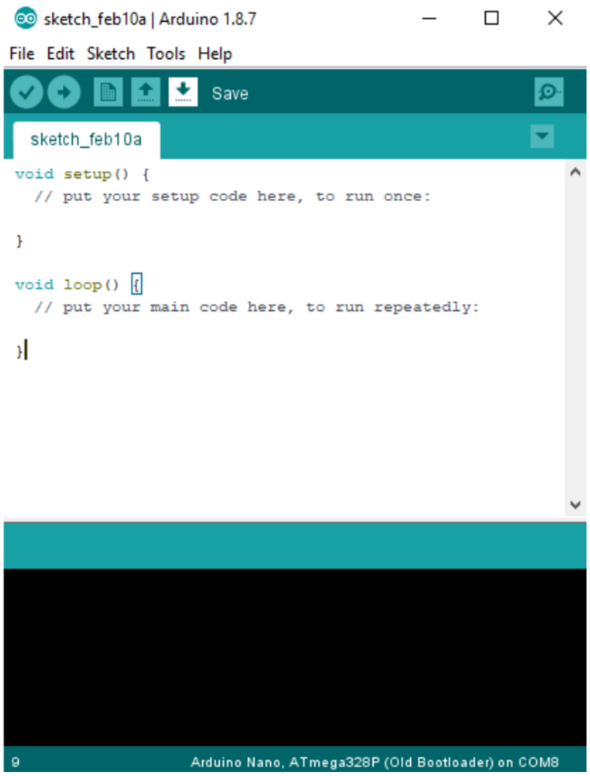
Windows app Requires Win 8.1 or 10
Get

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

Figure 10.5: Arduino IDE Download Links

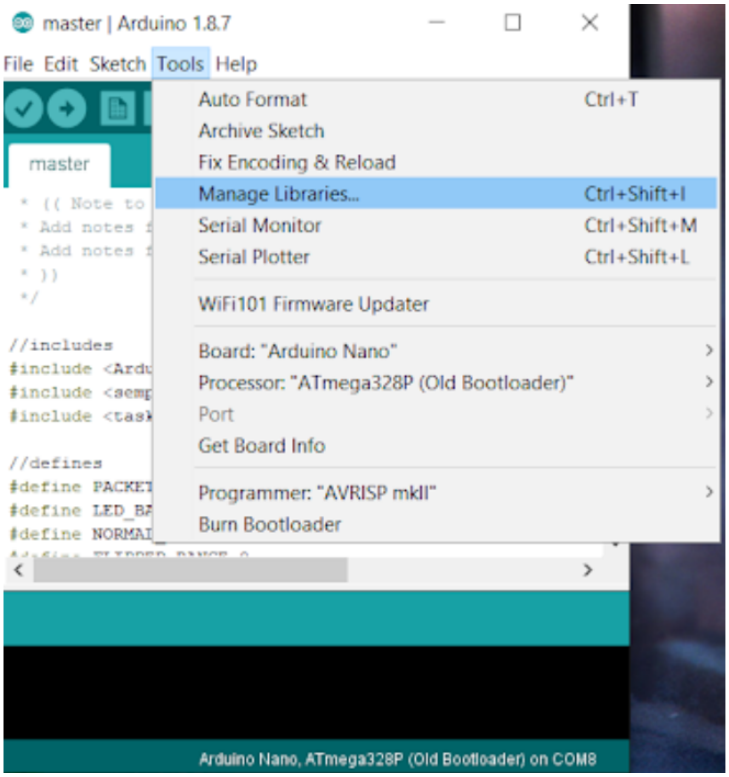
Download the installer that corresponds to your computer. Once downloaded, run the installer and follow the steps presented. After installation you should now have the Arduino IDE available for use. Running it will open a new editing window where you can begin development.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

9 Arduino Nano, ATmega328P (Old Bootloader) on COM8

Figure 10.6: Arduino IDE


master | Arduino 1.8.7
File Edit Sketch Tools Help
master
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino Nano"
Processor: "ATmega328P (Old Bootloader)"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Arduino Nano, ATmega328P (Old Bootloader) on COM8

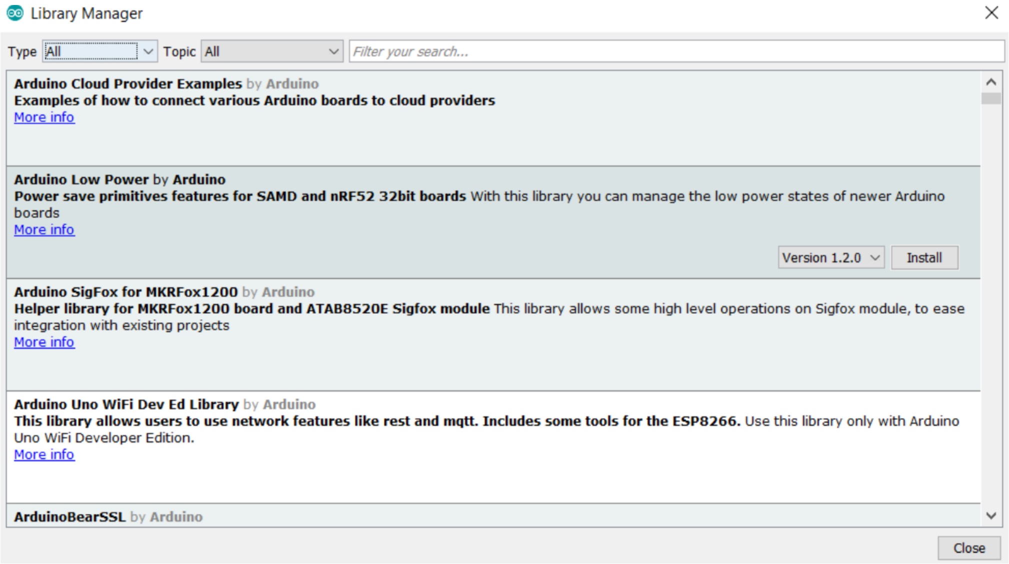
```
 * (( Note to
 * Add notes f
 * Add notes f
 * ))
 */

//includes
#include <Ardu
#include <semp
#include <task

//defines
#define PACKET
#define LED_BA
#define NORMAL
```

Figure 10.7: Manage Libraries Section

10.3.3 Obtaining Required Code Dependencies

In order to compile the software for the LED master devices you will need to install the FreeRTOS library for Arduino AVR devices. This can be done simply within the Arduino IDE itself.

Within the Arduino IDE select the Tools tab. A list of options should display. Select Manage Libraries.



Arduino Cloud Provider Examples by Arduino
Examples of how to connect various Arduino boards to cloud providers
More info

Arduino Low Power by **Arduino**
Power save primitives features for SAMD and nRF52 32bit boards With this library you can manage the low power states of newer Arduino boards
More info

Arduino SigFox for MKRFox1200 by Arduino
Helper library for MKRFox1200 board and ATAB8520E Sigfox module This library allows some high level operations on Sigfox module, to ease integration with existing projects
More info

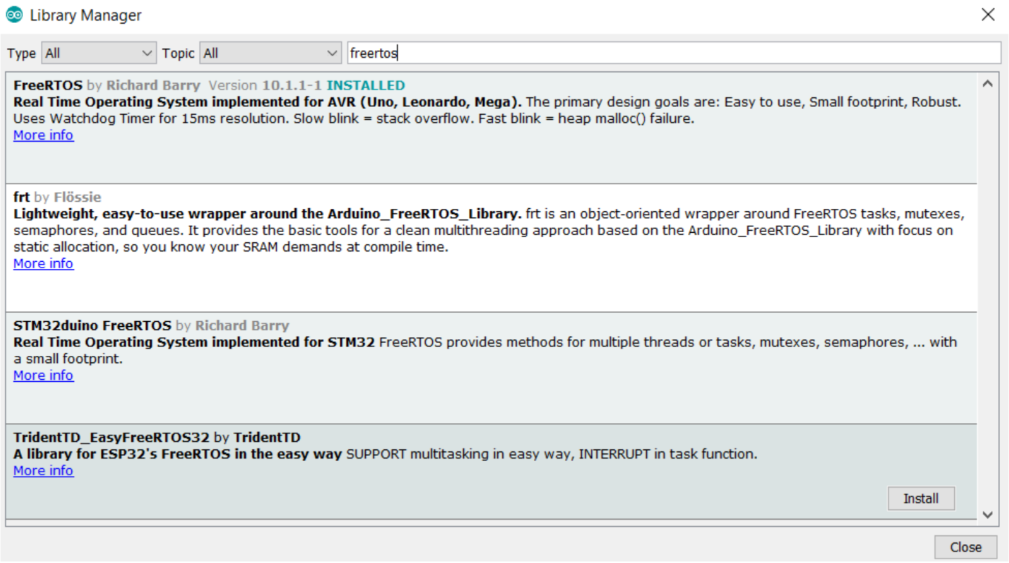
Arduino Uno WiFi Dev Ed Library by Arduino
This library allows users to use network features like rest and mqtt. Includes some tools for the ESP8266. Use this library only with Arduino Uno WiFi Developer Edition.
More info

ArduinoBearSSL by Arduino

Figure 10.8: Arduino IDE Library Manager

The Arduino Library Manager should open. Here is where you can find numerous libraries to enhance the functionality of your Arduino software. Each library has many functions/methods to perform tasks like connecting Arduinos to the internet or, like we will be doing, enable multi-tasking.

In the search box enter FreeRTOS. Several options will load. The one required for the led stair software is FreeRTOS by Richard Barry - Real Time Operating System Implemented for AVR? Install this one.



FreeRTOS by Richard Barry Version 10.1.1-1 INSTALLED
Real Time Operating System implemented for AVR (Uno, Leonardo, Mega). The primary design goals are: Easy to use, Small footprint, Robust. Uses Watchdog Timer for 15ms resolution. Slow blink = stack overflow. Fast blink = heap malloc() failure.
More info

frt by Flössie
Lightweight, easy-to-use wrapper around the Arduino_FreeRTOS_Library. frt is an object-oriented wrapper around FreeRTOS tasks, mutexes, semaphores, and queues. It provides the basic tools for a clean multithreading approach based on the Arduino_FreeRTOS_Library with focus on static allocation, so you know your SRAM demands at compile time.
More info

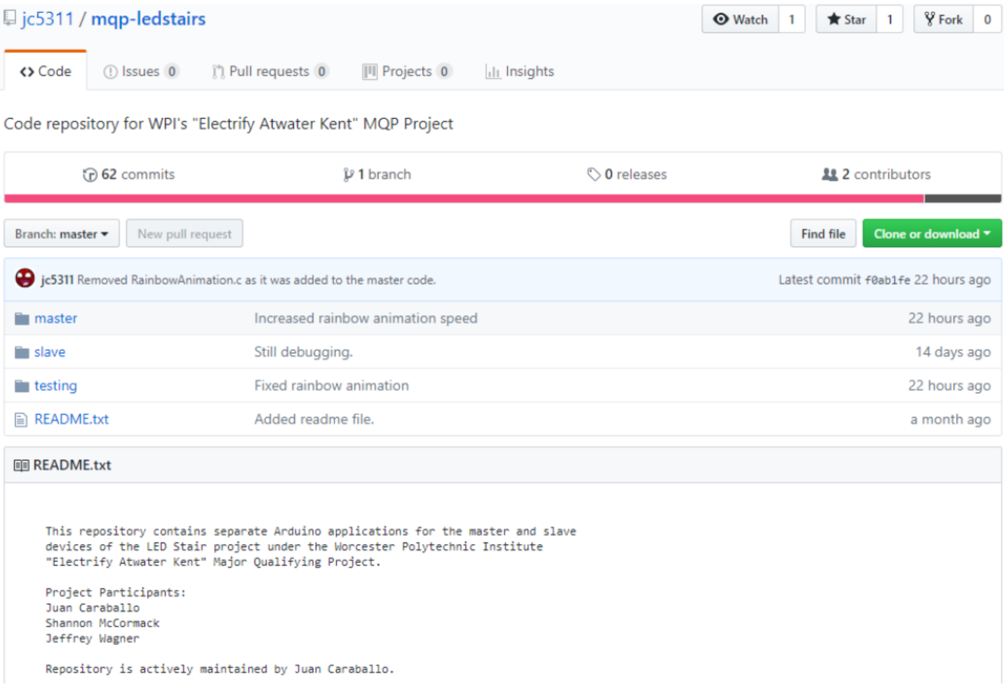
STM32duino FreeRTOS by Richard Barry
Real Time Operating System implemented for STM32 FreeRTOS provides methods for multiple threads or tasks, mutexes, semaphores, ... with a small footprint.
More info

TridentTD_EasyFreeRTOS32 by TridentTD
A library for ESP32's FreeRTOS in the easy way SUPPORT multitasking in easy way, INTERRUPT in task function.
More info

Figure 10.9: Library Manager FreeRTOS Search

At this time this is the only dependency required to compile the led stair software. The next section will now guide you on how to obtain the led stair code. Downloading the LED Stair code repository Now that the Arduino IDE has been acquired you will need to download the led stair code repository which holds all software for both master and slave devices.

Visit www.github.com/jc5311/mqp-ledstairs. This will bring you to a webpage similar to that below,



Code repository for WPI's "Electrify Atwater Kent" MQP Project

jc5311 Removed RainbowAnimation.c as it was added to the master code. Latest commit f0ab1fe 22 hours ago

master	Increased rainbow animation speed	22 hours ago
slave	Still debugging.	14 days ago
testing	Fixed rainbow animation	22 hours ago
README.txt	Added readme file.	a month ago

README.txt

```
This repository contains separate Arduino applications for the master and slave
devices of the LED Stair project under the Worcester Polytechnic Institute
"Electrify Atwater Kent" Major Qualifying Project.

Project Participants:
Juan Caraballo
Shannon McCormack
Jeffrey Wagner

Repository is actively maintained by Juan Caraballo.
```

Figure 10.10: MQP-LedStairs Github Repository Page

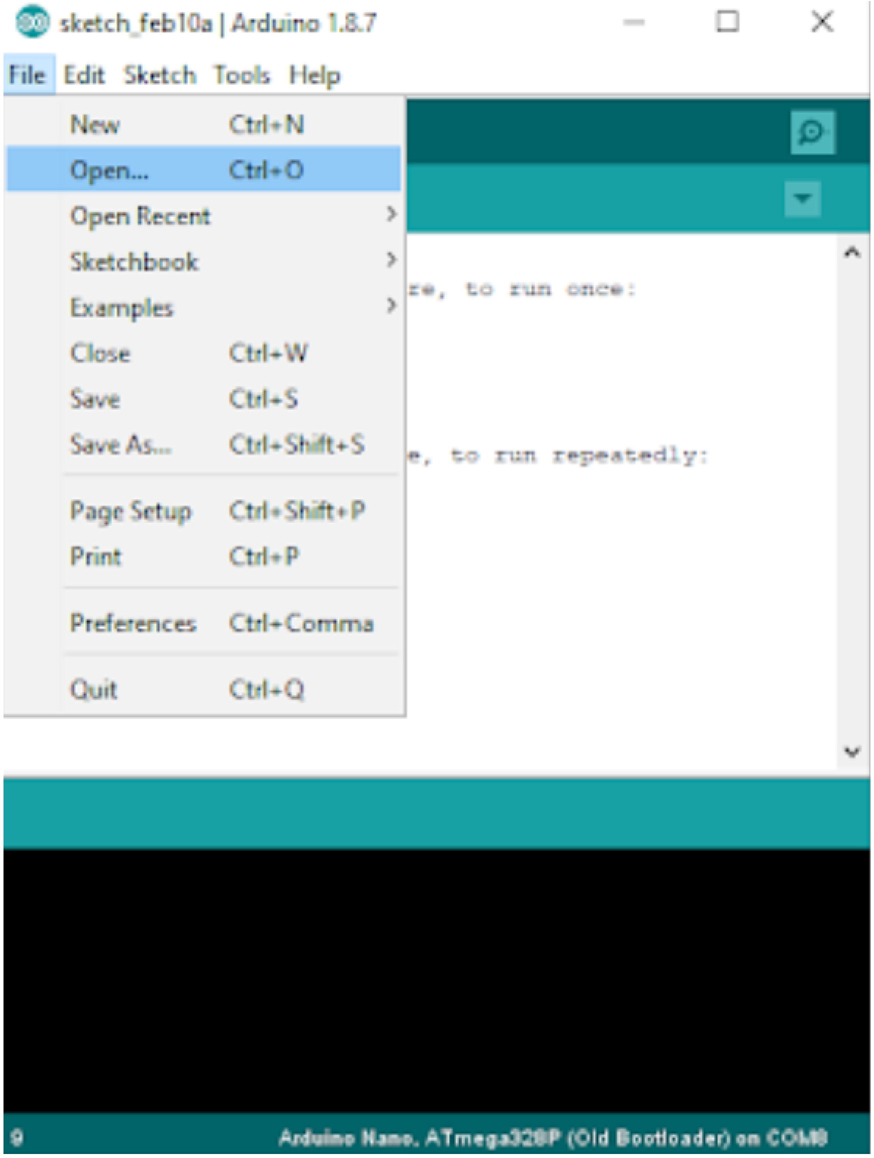
This page is where the codebase for the led stair project is hosted. Here you will find multiple folders containing software for the master and slave devices as well as a README with information concerning the project participants, the state of the software, and how to run the code once it has been downloaded.

On this page find the "Clone or Download" button. A dropdown will appear presenting a link to "Clone" the repository (more information on this can be found in the suggestions section) and a button to download the repository as a .ZIP file. Select "Download ZIP.". Once downloaded extract the contents of the file, this should yield a mqp-ledstairs-master folder containing all of the data from the repository page. Note that git uses the "-master" label on this folder to signify the git branch from which the repository was obtained and has no relation to the master device in the led stair system.

10.3.4 Running and Flashing the Code

With the Arduino IDE installed and the LED Stair Code repository downloaded we can now review the code and flash it to an Arduino device. Note that we will present the instructions as if flashing the master device. The steps are mostly the same for flashing the slave devices with the only caveat that the device's id must be updated before flashing. More details on this can be found in the suggestions section.

Open the Arduino IDE application and select File → Open.



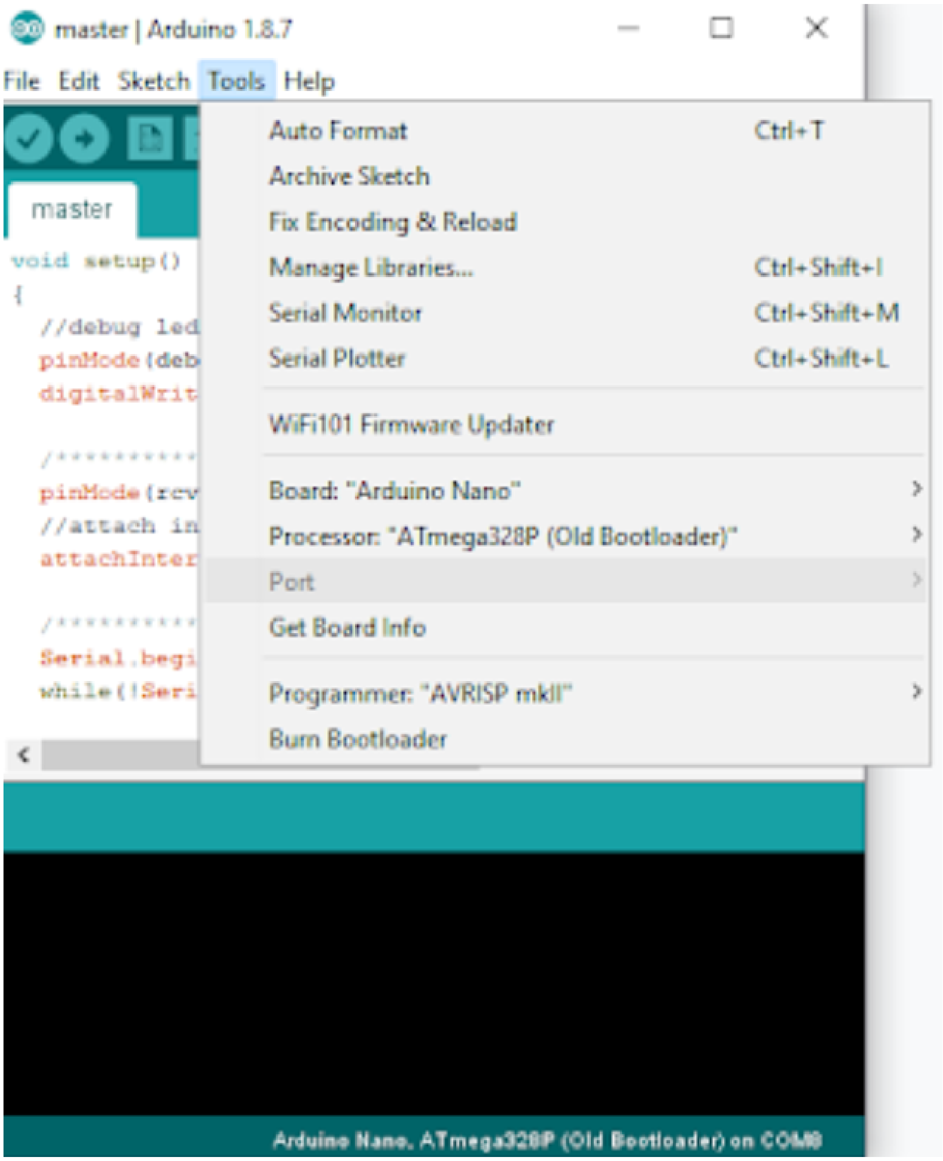
```
re, to run once:

e, to run repeatedly:
```

Figure 10.11: Arduino IDE, Opening .ino files

A file browser should open. Navigate to the mqp-ledstairs-master folder previously downloaded and open the master.ino file located within the master subfolder.

The Arduino IDE should now load all of the code for the led stair master device. Review it as you please. Once you are ready to install it onto a Arduino device, connect the Arduino to your computer.

With the Arduino connected to your computer select Tools → Port and select the COM port corresponding to the Arduino just connected. If only one is connected there should only be one listed. Note that in the figure below no Arduinos were connected to the computer and as a result no COM ports were listed.


master | Arduino 1.8.7
File Edit Sketch Tools Help
master
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino Nano"
Processor: "ATmega328P (Old Bootloader)"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader

```
void setup()
{
  //debug led
  pinMode(deb
  digitalWrit

  /**********
  pinMode(rcv
  //attach in
  attachInter

  /**********
  Serial.begi
  while(!Seri
```

Arduino Nano, ATmega328P (Old Bootloader) on COM8

Figure 10.12: Arduino IDE Port Selection

Finally, select the arrow button to compile and upload the code to the Arduino. If successful the black box at the bottom of the IDE should show status updates that compilation was successful and that the code was uploaded to the Arduino. After a few seconds, the Arduinos will begin attempting to send messages to slave devices.

10.3.5 Suggestions and Tips

10.3.5.1 Mac Laptop Crashes when connecting Arduino through USB

One of our team members, Shannon, had a problem where her Macintosh laptop would crash when trying to connect our test Arduinos to her machine for flashing. This is a known driver bug when using Macs to program clone Arduino boards with older versions of the Arduino AVR drivers. If you are experiencing a similar issue visit the following forum post which includes a couple of potential solutions to resolve this problem. `https://forum.arduino.cc/index.php?topic=428325.0}`

10.3.5.2 A note about FreeRTOS

At the top of the led stair master.ino file you will notice a few lines similar to those shown below.

FIGURE FreeRTOS Include Snippet

When using the FreeRTOS with Arduino devices, simply including the `#include<Arduino_FreeRTOS.h>` line will provide some of the macro definitions and functions required to utilize the library. However to reserve memory on the Arduino many features are placed in separate header files which are not included with the line `#include <Arduino_FreeRTOS.h>`. For example, if the use of semaphores is desired from the FreeRTOS you will need to include semphr.h in your .ino file. To find other parts of the library you may use, find the directory where Arduino libraries are installed on your machine. On my Windows 10 machine this location is `D:\jcpsp\Documents\Arduino\libraries`. All FreeRTOS header files are located in the FreeRTOS/src directory. For more information on how to utilize the library and to learn about its different features visit `https://www.freertos.org/`.

10.3.5.3 Utilizing Git and Github to Modify and Update LED Stair Software

It is heavily encouraged that you make use of the version control software known as Git when deciding to modify and update the code for the led stair system. Git provides a means of incrementally tracking changes and updates you make to software and provides tools to merge your work with other developers. It also provides a means of easily undoing changes you make in the event that an addition breaks your most recent version of the software. This is incredibly useful as it encourages the maintenance of an always-functional code base that gets updated with new features and bug fixes only when they have been developed and tested. You will also need a storage server to maintain your Git repositories. Github is a popular choice that provides an easy to understand interface and community for Git utilization. For more information on how to acquire and use Git visit `https://git-scm.com/`. For information on Github and how it can be used with your copy of Git, visit `https://github.com/`.

10.3.5.4 Cloning Git Repositories

As mentioned in the section Downloading the LED Stair code repository Github provides the ability to “clone” code repositories. Cloning is the action of downloading a code repository with a dedicated .git file that holds connections to the repository that was cloned. This is especially important when using git source control to manage changes you make to the code base. After you have made changes to the code, if you are interested in updating the repository you may “Push” hese changes to it. When attempting to do so git will use the local .git file stored within your copy of the repository to determine where the changes should be sent. Please visit the links provided in the previous tip Utilizing Git and Github to Modify and Update LED Stair Software for more information concerning actions such as clone, pull, commit, and push.

10.3.5.5 Changing Slave Device IDs during flashing stage

As stated in this section the feature of topography identification was not added by the end of this MQP project. As such the ID of each slave must be kept track of with the intended topography specified in the master.ino file. When updating the software on slave devices keep this topography in mind and make sure to modify the id variable for each to match where the slave is intended to be located in the slave-chain. Failure to abide by the address-location pair may result in slave devices never responding to the master and light bars illuminating out of order.

Chapter 11

Conclusion

Currently, all appropriate PCB designs and software is complete for the LED Stair system. With the system powered and all master-slave, trip sensor, and led strip connections appropriately made, the system will begin to perform autonomous animation on all connected light bars. When a sensor is tripped its respective light bar will illuminate white. Simultaneously all other unoccupied sensors will have their respective light bars off. Five seconds after the last sensor is tripped the system will resume full animation.

To add or remove slaves simply disconnect the desired amount from the end of the slave light bar chain. To ensure appropriate animation across all connected devices one must update any changes in the amount of light bars within the master's program. Specifically, one must modify an internal value which specifies the number of connected slaves as well as verify the address and both the software and physical topographies of each slave. A figure of a team member actively occupying two "steps" is shown below. Notice how only the occupied steps are illuminated and none of the rest.

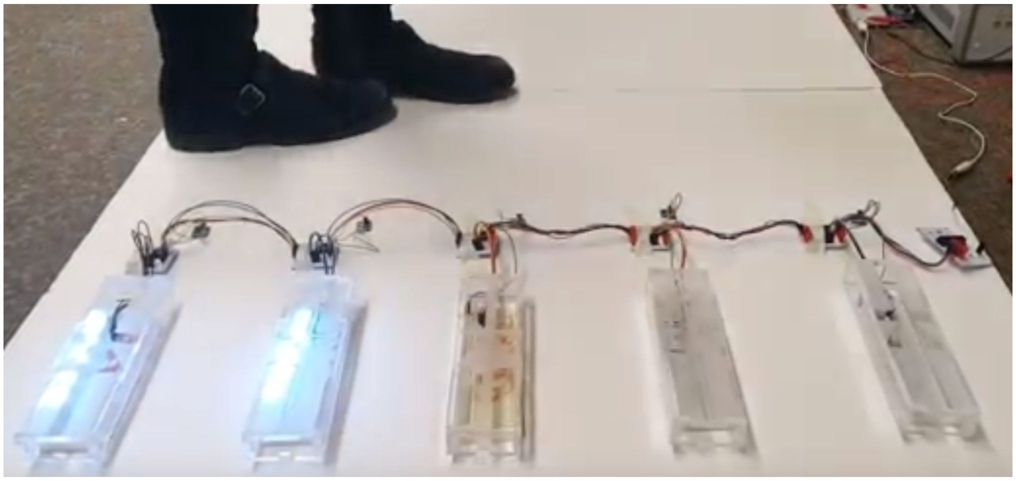


Figure 11.1: Snapshot from LED Stair Demo Video

Chapter 12

Future Recommendations

With the LED Stair hardware and software complete, our recommendations for this system are primarily for feature and installation additions. A list of our recommended improvements below.

- Reprint slave PCBs
- Fully install LED Stairs at Atwater Kent front staircase
- Add automatic topography identification

Starting with the slave pcbs an error was made in their original design which caused the sensor trip signals to also be transmitted to other slaves in the system. As such when one step was occupied, the entire system would illuminate white instead of just the occupied step. For our demonstration pictured in figure 8 this was resolved with minor modifications to each PCB. However, for long-term stability and efficiency the modified slave PCBs should be printed and used instead.

After reprinting the slave PCBs the system should be fully installed at its intended location at the front Atwater Kent staircase. Due to resource and time constraints our team was not able to perform an installation. However at its current state the system is fully operation and ready for installation.

A final recommendation for future work is to add automatic topography identification. The purpose of this feature is to allow the system to be easily rescaled by simply adding or removing slave light bars without the need for modifying the master's software. For proprietary installations such as that on the front Atwater Kent staircase this feature is not necessary as the system should not be rescaled. However for more artistic or demonstrative purposes this feature could be a relevant addition. The inspiration for this feature primarily stems from interest in originally designing the system to allow modification for residential use. With the approach taken for UART broadcast communication from master to slaves, the LED Stair system provides a base for a potentially competitive design to systems such as the Nanoleaf Auroura/Canvas smart light panels. A sample of which is shown in the figure below.

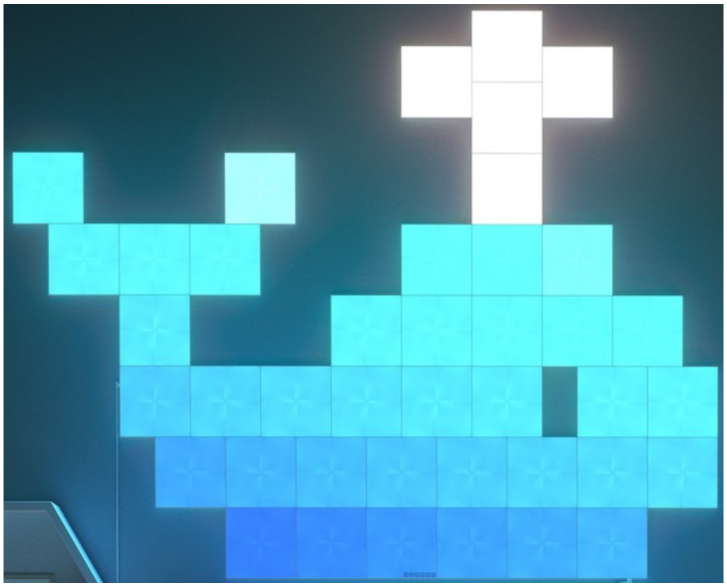


Figure 12.1: Nanoleaf Canvas Light Panels

This product utilizes a similar master-slave architecture with each light panel serving as an individual slave. Each is likely individually addressable with the overall topography identified and updated on power-up or runtime.

Part IV

References

Bibliography

[1] H. Zumbahlen, *Basic Linear Design.* Analog Devices, 2007. [Online]. Available: https://www.analog.com/media/en/training-seminars/design-handbooks/Basic-Linear-Design/Chapter8.pdf

[2] R. Elliott, “Active filters - characteristics, topologies and examples,” Jan 2014. [Online]. Available: http://sound.whsites.net/articles/active-filters.htm#s5

[3] C. Cunningham, “Circuit skills led color organ.” [Online]. Available: https://www.jameco.com/Jameco/workshop/video/ledcolororgan.html

[4] “attachinterrupt().” [Online]. Available: https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/

[5] S. Fox, “What is mdf? - mdf 101,” Dec 2015. [Online]. Available: https://www.bobvila.com/articles/what-is-mdf/

[6] “interrupts().” [Online]. Available: https://www.arduino.cc/reference/en/language/functions/interrupts/interrupts/

[7] “Language reference.” [Online]. Available: https://www.arduino.cc/reference/en/

[8] S. Legros, “Medium density fiberboard (mdf) vs particle board,” May 2018. [Online]. Available: https://forestplywood.com/blog/mdf-vs-particle-board/

[9] ——, “How to waterproof mdf (medium density fiberboard),” Jun 2018. [Online]. Available: https://forestplywood.com/blog/how-to-waterproof-mdf/

[10] “Mdf vs. plywood - differences, pros and cons, and when to use what,” Feb 2019. [Online]. Available: https://www.addicted2decorating.com/mdf-vs-plywood-differences-pros-and-cons-and-when-to-use-what.html

[11] “Plywood,” Mar 2019. [Online]. Available: https://en.wikipedia.org/wiki/Plywood

[12] “Queues, mutexes, semaphores... [inter-task communication and synchronisation].” [Online]. Available: https://www.freertos.org/Embedded-RTOS-Binary-Semaphores.html

[13] “Tasks [more about tasks...].” [Online]. Available: https://www.freertos.org/RTOS-task-priority.html

[14] “What the heck is mdf anyway?” [Online]. Available: http://www.modgsi.com/What-The-Heck-Is-MDF-Anyway_b_24.html

Part V

Appendices

Appendix A

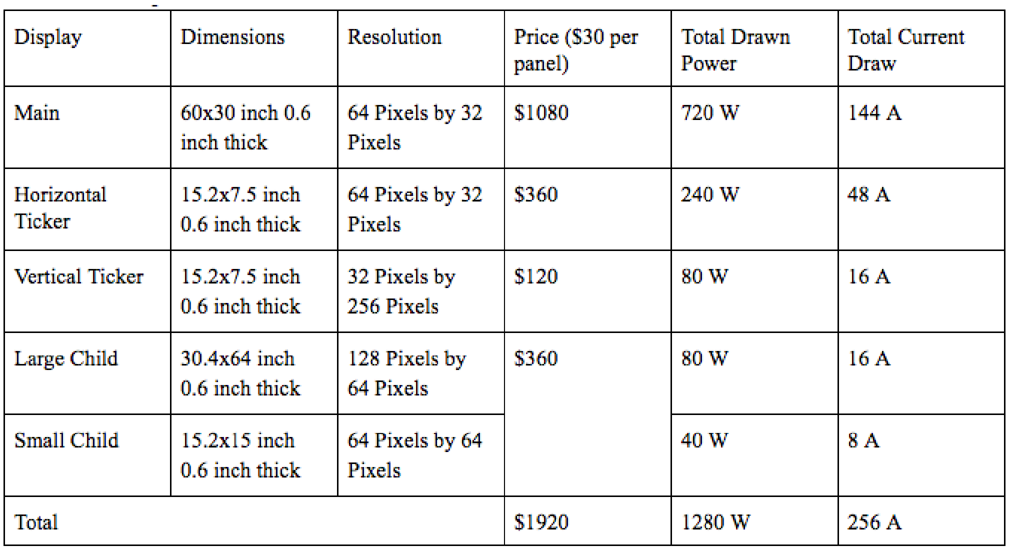
Pumpkin Lounge LED Matrix Display

A.1 Research from Past MQP Reports

A.1.1 Features

- AK Solar Panels
  - Measurements of power generation
  - Historical trend data on building energy consumption
- General Info
  - Campus events
  - Simple attraction
  - Display building map
  - Games
  - Fun facts

A.1.2 Information on Panels



Display	Dimensions	Resolution	Price ($30 per panel)	Total Drawn Power	Total Current Draw
Main	60x30 inch 0.6 inch thick	64 Pixels by 32 Pixels	$1080	720 W	144 A
Horizontal Ticker	15.2x7.5 inch 0.6 inch thick	64 Pixels by 32 Pixels	$360	240 W	48 A
Vertical Ticker	15.2x7.5 inch 0.6 inch thick	32 Pixels by 256 Pixels	$120	80 W	16 A
Large Child	30.4x64 inch 0.6 inch thick	128 Pixels by 64 Pixels	$360	80 W	16 A
Small Child	15.2x15 inch 0.6 inch thick	64 Pixels by 64 Pixels		40 W	8 A
Total			$1920	1280 W	256 A

Figure A.1: Panel Information

A.1.3 Information on Capacitive Touch

- PCB Final Size of 8.25x7.25 inches

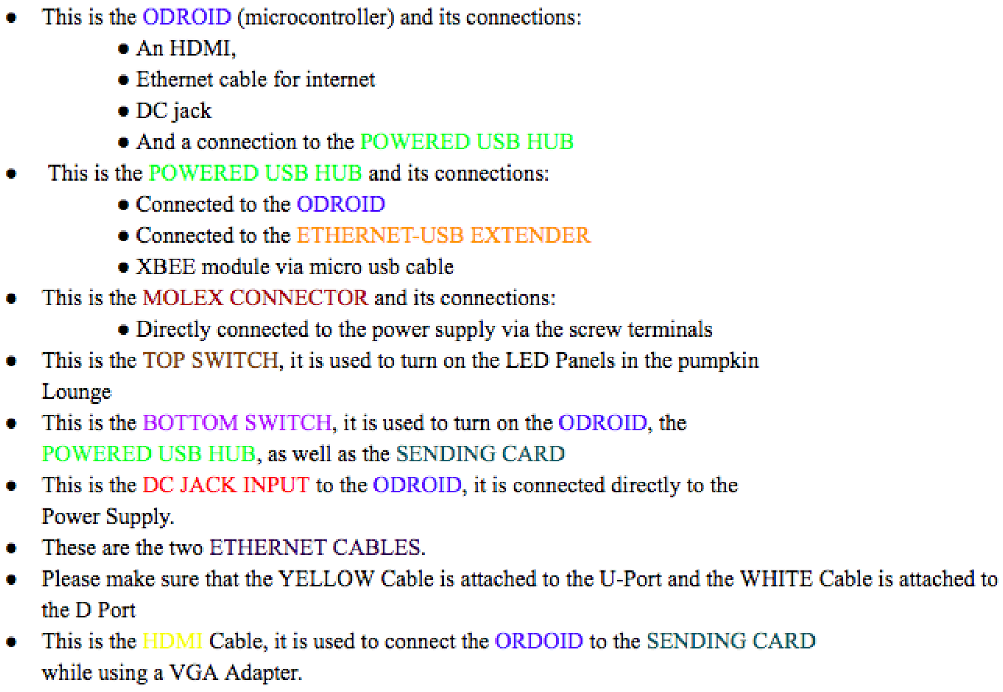
- Arduino Micro is soldered directly to PCB
- Only one cable (micro USB cable for power and data) connecting capacitive touch to ODRIOD XU4 Display
- Four directional arrows and select pad for easy navigation through menu
- Menu pad and back pad for jumping between levels in user program
- Unassigned buttons for user flexibility

A.1.4 Other important values

- 120 VDC from Building
- 5 VDC is operation voltage for panels
- 36 VDC is maximum input of XL4015 Converter
- 12.9 A is max current for power panel
- 14 AWG wire from distribution block to main display (16 AWG for others)
- 18 AWG used for remainder of power panel

A.1.5 The Box

The ECE box holds the micro controller and other components crucial to the power panel. The major components are color coded and their connections:



- This is the ODROID (microcontroller) and its connections:
  - An HDMI,
  - Ethernet cable for internet
  - DC jack
  - And a connection to the POWERED USB HUB
- This is the POWERED USB HUB and its connections:
  - Connected to the ODROID
  - Connected to the ETHERNET-USB EXTENDER
  - XBEE module via micro usb cable
- This is the MOLEX CONNECTOR and its connections:
  - Directly connected to the power supply via the screw terminals
- This is the TOP SWITCH, it is used to turn on the LED Panels in the pumpkin Lounge
- This is the BOTTOM SWITCH, it is used to turn on the ODROID, the POWERED USB HUB, as well as the SENDING CARD
- This is the DC JACK INPUT to the ODROID, it is connected directly to the Power Supply.
- These are the two ETHERNET CABLES.
- Please make sure that the YELLOW Cable is attached to the U-Port and the WHITE Cable is attached to the D Port
- This is the HDMI Cable, it is used to connect the ORDOID to the SENDING CARD while using a VGA Adapter.

A.1.6 Results and Installation

- Behind the main display there is a “distribution block’ where all energy lines seem to be going from the displays. This must be where the main power is.
- Correction, yes power lines are fed through here. However the main power source is in ak 113. There is a random “gray’ box that has the power supply as well as a few other devices.
- Distribution blocks I guess are just hubs where certain cords travel through to separate displays.

- They claim the power panel (the displays) are operational
- The ODRIOD they mention very often is the actual computer running the displays and hardware
- The ODRIOD and sending card (video controller?) are in the gray box in AK113
- The box also contains an XBEE for communicating with the XBEE in the AK317 lab that transmits data from the solar panels on the roof
- The box has two switches that control the hardware in the box and the displays (I think this may be the power switch for the displays).
- Not able to use wind turbine because not enough power
- Campus uses E-Mon and D-Mon sub-meters (meter info through Honeywell Software & AK meters from Automated Logic)
- Stranded, insulated wire to connect power to panels
- If raspberry pi loses WiFi, take to Network Operations in Morgan Hall
- They used their student account to register the ODROID as a device for WiFi so repeat the same process if the ODROID is not getting recognized due to the account being closed
- Solar Panels are not in optimal positioning and are not working well-this display cannot run off of these

A.2 Day One

4:30 PM to 6:30 PM Sunday, December 17, 2017

A.2.1 Inspection Notes

- No HDMI VGA Converter, assumed stolen. Therefore we are unable to display on monitor.
- Fix with a converter or monitor with HDMI. This converter can be used with monitor or video control depending on where you would like the screen to be displayed.
- The ODROID turns on (but the fan in this device stays off, unsure if this works properly and just has not turned on because the device has not reached a concerning temperature), just no way to display how the device is working. Everything in box seems to be powered on upon switching the bottom switch.
- Displays do not turn on due to no converter.
- Fan is disconnected, no found place for connection so far (saving this for a later concern).
- Connections seem fine.

A.2.2 General Notes

- NEED CONVERTER
- Juan will bring this for testing purposes.
- Upon attaining this, investigate functionality of device through computer monitor, and assuming everything goes well, we plan to try panels.
- Are the panels being powered properly? Will they be able to turn on?
- Need to also look into powering this device, what needs to be off? How many computers being on does it take to short the circuit breaker? Is there another way to power the panels?
- How long can this run for without a different power supply? Would the panels be able to be charged from a source in the lounge?

A.3 Day Two

9:37 AM to 11:57 AM Monday, December 18, 2017

A.3.1 General Notes

- Fan turned on, must have needed the display output to force turn on. This assumption was incorrect. Upon testing this theory, we unplugged this from the display and the fan still turned on.
- Tried connecting ODROID to monitor with HDMI to VGA. No luck getting this to display to monitor.
- Still in need of DVI HDMI converter. DVI Dual link
- On ODROID, getting a signal (green LED is flashing) and red LED is on, assume this means power.
- Xbee is on and powered.
- Power supply signals that it is being fed.

A.3.2 ODROID Notes

Datasheet: `https://magazine.odroid.com/wp-content/uploads/odroid-xu4-user-manual.pdf` LED Status:

- The red LED: Is on when power is available
- The blue LED: Is on (solid light) when the bootloader is running
- The blue LED: Blinks slowly when the kernel is running, like a heartbeat
- The blue LED: Blinks quickly when the kernel is in panic mode

As of now, it seems power is available and the kernel is running, everything seems to work properly. Just still no display.

Next test, letting it run for 5 whole minutes and wait for an image to appear on monitor. After a minute and a half about, the fan turns off. LEDs work accordingly whole time. After five minutes, still nothing on display.

For the next test, we will be using a HDMI to DVI converter, with an HDMI cable to connect the ODROID to the monitor. We hope this will display as was expected today, but after the complications that arose today, we suspect we will have to look further into the output of the ODROID. Worst case, last test we will do before assuming something is wrong with this device, would be to connect it to a monitor with no converter thus only using HDMI connections. For this we will need a monitor with an HDMI port.

Upon changing wires and using an HDMI cable with DVI adapter, we were able to get the connection working and now display is working properly. The monitor starts on Ubuntu desktop.

Typed this: "sudo ethtool eth0"

Into terminal to check internet connection, there was no internet connection found.

Moving on, we did the steps specified in report two, running the programs and then unplugging the monitor and connecting the ODROID to the video control the switching top switch. Although the panels were lit up, the connection seems to be faulty. The monitor screen that was showed previous appeared scattered on the displays and were flashing and distorted. We believe this is due to a faulty wire or error in code. The display changed when hitting the back button on the capacitive touch pads.

This error could be connection related, software related, or a hardware fault somewhere within video controller.

A.3.3 Things to be done

- Find origin of video fault
  - It seems that this is a common issue from other people who own an ODROID and a solution has not been know yet. However, a solution was discovered through the LED Studio Software that is further explained later in this section. So check software first!!!
  - Check hardware
  - Check wiring
  - Check distribution box

- Connect to internet
- Check how many computers can be powered while powering panels or find a different location for our power source

Attempt relocating “the box” to the ECE office or have it connect wirelessly to her computer (so that Christine can control this).

A.4 Day Three

6:00 PM to 8:30 PM Tuesday, December 19, 2017

A.4.1 General Notes

- Plan to ask IT to set up ethernet down the road, however for troubleshooting purposes we have obtained an internet connection for the device from a phone. We were successful in getting a connection.
- Connected video control to a laptop, everything works perfectly when doing so. May have to do this for Christmas party, or want to do this in the future.
- One option would be to use a computer in Christine’s office specifically for the display, getting rid of the ODROID entirely. In that case, our tasks would be to relocate the box to the office and map out on the laptop screen where each display starts and ends.

A.4.2 Mapping Notes

Resolution (1280x768) Initial location of power panel (-1, 308) size (384, 192) Initial location of children panels and horizontal ticker (-1, 21) size (768, 224)

A.4.3 Power Panel Code

Power Panel Navigation:

- X down (X and W is also to navigate through a section)
- W up
- D right
- A left
- S select
- E back

A.4.4 Power Panel Abilities

- AK map can be displayed
- About and all tabs in it display correct information
- In games
  - Pong is working
  - All other options lead to black screen
- AK history upon clicking through has two working slides (rest are black screen)
- AK MQP menu opens but all options lead to a black screen
- AK News breaks code and shows error message

Upon connecting this to internet, the error was gone and program fully works. Section on main display where LEDs cannot shine any shade of red.

A.4.5 Children Code

The children panels when run come up with errors saying the files could not be accessed or found for pacman and other games.

To fix this, we located the pizza image and re-typed the address for it, this ran successfully without the error. Upon fixing the pacman, more errors arose.

Our next task is to relocate all images into one easy folder and re-type all addresses for them in the code. We have made a desktop folder “ChildrenImages” and within it is all image file locations. When implementing these new address in the code, all image errors were erased. Now we face: java.lang.NullPointerException With multiple pointer errors

The problem here was that there were two Children Panel files and the one being used was missing font information. Therefore I copied the “data” file from the other Children Panel folder and ran it. The error was gone.

A.5 Day Four

10:40 AM to 6:00 PM Wednesday, December 20, 2017

A.5.1 General Notes

In process of making video to display for Christmas party.

All code on ODROID is running smoothly with working capacitive touch, need to get ethernet working (for testing purposes we used a USB connection with our phone). The only problem is the distorted display when projected to panels. This is not a problem with the video controller or the panels, because when connecting them to a computer instead of ODROID the image is clear and perfect. Therefore why we are creating a video for the christmas party.

We plan to try and get rid of ODROID all together and get a computer upstairs for Christine to control for display panels and use same video controller and most of the other equipment, just replacing ODROID.

For video placement edits GOMPEI 2 pixels on left 2 pixels on bottom BASKETBALL 3 Right 1 down Clock and fun facts need to be moved one up thus moving horizontal display one up

A.5.2 VIDEO MANUAL

1. Change computer resolution to 1280 x 768
2. If possible change the screen sleep mode to never instead of 15 minutes so that the display does not enter sleep mode while the video is playing
3. Play video in full screen mode
4. Click option buttons and click Zoom to fill
5. In options, also click repeat so that the video is on a continuous loop
6. Unplug DVI from monitor and instead connect it to video control
7. Flip bottom and then top switch to power everything on
8. All finished!

A.5.3 McNeill Notes

Change the word facility to faculty Horizontal ticker make text bigger instead of two lines Say “Welcome Joseph and Phyllis Satin” (white words) “Happy Holidays from the ECE Department” (alternating red and green words)

A.5.4 Final Presentation

Everything fitting screen perfectly, up and running through computer in lab AK 113.

A.6 Day Five

12:00 PM to 3:00 PM Wednesday December 27, 2017

A.6.1 General Notes

As of now, the ODROID to video controller connection does not work at all. Before, the display was distorted and flashing, now nothing shows at all. Although, ODROID still has the ability to display to a monitor and a computer can still be displayed by the video controller.

An attempt was made to run the java code on eclipse on a PC in the lab and was unsuccessful.

We now are downloading the processing application to a personal computer and going to try to run the code on that. If this is successful, our next step would be to get the code running on a computer that can be used by Christine and relocate the video controller (and possibly "the box") to the second floor.

Once processing was downloaded, we moved all necessary files onto Juan's personal laptop, edited the locations of files within the code, and ran the code on the computer. This ran successfully. Next we plan to make this organized folder into a zip folder; allowing the ability to access this on a different computer. The computer we plan to use for Christine is undecided, this will be chosen after resolving the issue of connecting to the video controller.

A.6.2 Box Options

1. Extend cable to connect an upstairs computer to video controller, leaving box untouched.
2. Get ODROID running properly and allow access for Christine via remote desktop.
   (a) Maybe try backing up necessary data and completely clean operating storage and system (full wipe out), then reinstall saved data. (BUT KEEP FULL BACKUP/CLONE INCASE THIS DOES NOT WORK)
   (b) Look through settings, in search of any disturbances/restrictions in relation to display.
3. Replace ODROID with new hardware and allow access for Christine via remote desktop.

A.6.3 Things To Do Next

Our next step is to attempt option 2a, we will first be backing up a complete clone of the device and then use the follow commands within the terminal.

We will be updating this using the SD card, but plan to do this update tomorrow given the expected amount of time this will take. Today is being wrapped up after cloning the device and saving everything on the personal computer.

A.7 Day Six

2:34 PM to 4:30 PM Friday, December 29, 2017

A.7.1 General Notes

Starting from where we left off, we will now clone the ODROID and update/upgrade the device.

After several attempts with the currently used device, we came to the conclusion that the ODROID may not be our best option. We have decided to go forward with option 1, and find a suitable computer for our fully functioning software. This will be selected with the following specifications in mind:

- Ability to connect to a network and be controlled via remote desktop
- Ethernet port
- Run full OS (similar to raspberry pi)
- Low power
- Small, yet fully functioning computer
- Preferably HDMI friendly

We still have to research to find exactly what capacitive touch is connected to. We know that it works when using ODROID (with the glitch, but still works) and it does not work when using computer. So our guess would be that it's connected to the ODROID somehow and not the video control.

Looking at the specifications, the ODROID is seemingly perfect, therefore we are going round two. Updating now.

Upon updating, we then displayed it on the panels. This showed a new and improved glitch.

Our next task is to upload a new version of LED studio software onto the video card via Juan's PC. This is with the hope to change the resolution of the video card itself, possibly fixing the distorted image that currently appears on the displays. We have ran out of time on this due to the upload coming up with an error.

Ending today with the feels that the ODROID is a lost cause. Next time we meet we will have done individual research on components to use instead as well as Juan will bring in a raspberry pi to test the software with in the meantime. In addition to, communication with each other and reaching out to past professionals that have helped with the installation process for this project. Therefore, the next time we meet we will have a plan of attack as well as a device for testing purposes.

A.7.2 Replacements for ODROID

1. PC
2. Raspberry Pi 3

Appendix B

Improvements for Pumpkin Lounge Display

B.1 Tasks for Functionality

- Trading out the ODROID to fix current visual artifacts.
  - Thinking of computer or raspberry pi.
- Having the computer be displayed behind a glass display case in the pumpkin lounge with the ability for ECE Department Office to access using remote desktop.
- Possibly (depending on level of difficulty) fix solar panel data output.
- Additional circuit breaker installment for powering displays.
  - Talk with O'Rourke
- Matrix power panel analysis
- Clean, restore, and update software.
  - Make available to department office
  - Mazumder is thinking of a possible timer to make the displays automatic

B.2 Additions

- Develop application for a simple way of modifying the displays (for department office).
  - E.g. runtime, display content
  - Scheduling/timing feature
  - Add manual text entries for ticker
- The Jukebox
  - Represented by the glass box that holds the computer, with touch LCD screen and speakers added to outside of box.
  - Play songs from a previously selected and approved list.
  - Using a state machine, will switch GIF displays to audio visualizer and name of song.
  - Possible 50 cent Goat Bucks charge (similar to printers) for adding a song in queue.
  - Wireless headphones, swipe ID and use headphones
- Illuminate Projects
  - Upon selecting a MQP on the main display for more information on it, the project in the lounge will light up wirelessly.
  - This will also include updating the current MQPs that show on main display to include MQPs that are shown in Lounge.

- LEDs on distribution channels/cable raceways
  - Programmable LEDs that light up simulating data transmission between panels from the main display.
- Periodic Program Reset
  - Since the software works properly from startup for an unpredictable amount of time, have periodic resets of the software in case any extraneous bugs or glitches occur.
- IP Camera
  - Live stream the displays to an online web page, could be used to monitor them occasionally for any bugs that may occur that require attention
- Bug report button
  - Large physical button students can press to report that there is a bug (may be simply timestamped and the IP camera and save a screen shot at the time of button press)
- Button Professors can press in their offices to signal whether they are occupied or available (updates a section on the main panel if someone is looking for professor availability)
- Jokes

Appendix C

Innovation Readings

C.1 Readings

The intention behind these reading assignments was to improve our innovative thought processes. A Whack on the Side of the Head and A Kick in the Seat of the Pants both discuss four main roles of the creative process.

- The Explorer: this is where the innovator must explore all options. Gather ANY information that has to do with the subject at hand and pool all resources.
- The Artist: the artist takes the ideas gathered from the explorer and plays around with them. Looks at them in a new light or from a different perspective, merge two ideas and make a new idea.
- The Judge: now is where decisions are made. The judge weighs out all the ideas from the artist and explorer and decides which ones are good and which are bad.
- The Warrior: this is where the action takes place. After the judge has decided on the idea and has weighed all his/her options, the warrior is left with a final decision. His/her job is to take that decision and make it a reality, fight for that idea and do not take no for an answer.

These roles are fluid in the creative process, often one will jump between roles. However, it is important to take note of the role that needs to be in place give every role a chance to shine. If you are the artist too early, then there may still be ideas yet to discover. If the judge comes out before his/her time, then an idea could be thrown away before allowing it to be fully developed into something brilliant. Many things can go wrong in the process, therefore these books helped by giving tips on how to proceed when faced with a creative block and exercises to open up your mind to new ideas.

C.2 Writings

All team members wrote two assignments:

1. Write a paragraph for each role explaining a time you had to play this role as well as a concluding paragraph on your strengths and weaknesses in relation to these roles.
2. Write a page for two exercises you completed and your thoughts on them.

At the end of the week, we shared about our writing assignments and learned from each other. Something we noted were the roles each member felt confident or weak in. With this in mind, we would know when one team member may need more help, as well as who to go to when you are stuck. It helped the team open up to each other a bit more and see who fit which roles best moving forward. Another thing we gained from these readings was all the exercises and tips given. Some were used to generate ideas for this project, while other things were kept in the back of our heads to help us through the creative process.

Appendix D

Foisie Innovation Versus Atwater Kent

D.1 Atwater Kent and Foisie Observations

Atwater Kent is home to Electrical, Computer, Robotics, and Systems Engineers, along with many other students and faculty. Built in 1907 and named for a radio pioneer and member of WPI's class of 1900, Atwater Kent Laboratories was the first building in the country dedicated to education in electrical engineering. It has since been updated, but held at very different standard than today's technical/academic buildings world-wide. The general consensus of the building was:

- Broken equipment and displays
- Poorly displays student work
- Not enough windows
- Amazing history within building
- Cubicle-like work areas
- Isolated
- Much of the building sporadically dark
- Inconsistent paint/color schemes
- Random furniture
- Inconsistent aesthetic - disorganized mix of old and modern design

Most of these problems could be easily fixed, but in order to make a worthy project out of this, we needed something more. Therefore, we moved to Foisie to inspire creativity. Foisie's main focus was openness, which is a key factor for any innovative work space. We walked through every floor of this building and took down notes:

- Fluid architecture
- Creative and well-thought-out wall designs
- New furniture
- LED lighting
- Open space
- Large use of windows and glass
- Light, no dark corners
- Consistent color schemes
- Clean
- Artistic vibes - a lot of thought was put into the aesthetics

D.2 Conclusions

After observing both buildings, we listed some general takeaways we had from these experiences.

- AK is MUCH darker than Foisie; there are many spots that are randomly dark and areas that should have more light. Some examples are:
  - Basement study area
  - Basement grad study room
  - Office hallway in front of the Pumpkin Lounge
  - 3rd floor hallways
- Related to light, there are much more Windows in Foisie than AK, adds to natural light and potentially affects individual moods
- AK has many walls and hallways making it feel enclosed compared to Foisie which has an open breathable feeling
- AK has inconsistent color schemes between all floors and offices
- 3rd floor is kind of depressing, very grey and just plainly feels...dead
  - The newly renovated labs space looks much cleaner and open, like a new canvas
- Random furniture at the back of the first floor
- MQP project posters are not obvious or presented eye-catchingly; easily ignored
- AK although regularly cleaned appears very dirty; rugs, ceilings, some walls; either stained or permanently dirty from many years of interaction
- Rugs often have different patterns; placed in areas that likely don't need them despite medium traffic
- Some areas of the building are very old in design while random splotches are modern due to recent renovations (Robotics office space, MITRE Lab, 3rd floor renovated lab)
- Overall, the building has an inconsistent aesthetic (e.g. mismatched furniture, incoherent color scheme, etc.)
- Inspiration can be taken from the Dean of Engineering's office

This was along with a list of ideas on how we could possibly improve AK:

- Display panels that digitally display MQP project posters both old and new
  - Also considering moving the displays to another area of the building (or taking out) to allow for a TV matrix (similar to one in Foise) so that more information can be displayed.
- Acrylic water flow pipes around the main entrance foyer with pumps, reservoirs, etc. with a panel describing how electronic circuitry can be compared to hydro mechanical systems when explaining electron flow.
- Displays in the entrance foyer surrounding the inner doors to perform hydro mechanical system animations, weather, and other miscellaneous content with some panels geared to presenting information
- Sensors and devices to welcome people walking into the buildings
- Sensored devices that react to human present (turn on lights, enable a display, etc.)
- Deciding on a building color scheme that would allow for areas to have different accent colors similar to the different floors of Foisie
- A robot in the entrance foyer that welcomes or waves to visitors
- Basement chalkboards replaced with magic sketch boards (search for Magic Sketch Board toy online)
- Add more lighting to the building and/or new lamp and light architectures
- Fix poster facing elevator on second floor, or replace with "digital poster" touch LCD that allows one to flip through MQP posters by touch screen or left and right arrow buttons.

- Make ECE mailbox list on second floor digital
- “Pseudo windows,” display of a live feed video of another area in the building or outside for rooms without or with limited windows to feel less like a dungeon
- Circuit designs for past projects painted on the walls, we thought this would be like an ECE history book
- Moving Tesla Coils around and painting the wood beneath them to better display them, maybe adding lights as well. We felt this was a very cool addition to pumpkin lounge, however it goes under-appreciated due to its presentation.
- Changing location of the old power panel thing or adding lights so that it is not hidden away in a dark corner. Brings some fascinating history to AK, but most people do not notice it.

Appendix E

Survey Questions & Results

I. Question One: What is your graduation year?

2018:	1	0.94%
2019:	42	39.6 %
2020:	28	26.4%
2021:	22	20.8%
2022:	7	6.6%
Other:	6	5.7%

II. Question Two: What is your major?

ECE:	68	64.2%
ECE/RBE:	12	11.3%
RBE:	2	1.89%
RBE Double Major:	1	0.94%
ECE Double Major:	10	9.43%
Other:	13	12.3%

III. Question Three: If you had an infinite budget, what would your dream tech building include?

Food (17)

Healthy food (2)
Snacks (2)
Food court
A student usable kitchen
Vending machines
A place to get coffee
Dunks/starbucks
Wide diversity of food choices
FOOD THAT IS FREE (unlike Foisie)
Coffeee shop
24/7 cafe
Food vendor
Food court
Cafe
Maybe a starbucks I dunno

Seating (15)

Bean bags (3)
Couches
100 couches
Sleeping pods
Hammock
Beds/cots for students to take naps in
Comfy chairs and hammocks
Amazing chairs!
Nap pods
New chairs on the first floor outside AK113
Comfortable chairs
Comfortable furniture
Comfortable and cool furniture

Boards (8)

Smart boards (2)
More whiteboards
Rooms with walls that can be written onto.
Big white boards
White boards
Lots of whiteboards everywhere
Digital whiteboard

Equipment/Parts (70)

-PCB Answers (10):
PCB printers (6)
PCB etching
A PCB Fab machine that does vias and through hole components.
A way to fabricate printed circuit boards in house
Pcb milling machines
-3D Printer Answers (9):
3D printers (7)
Metal and other luxury material 3d printers
Metal printers, most materials available for purchase
-Soldering Answers (8):
Ventilated soldering station (2)
Better soldering iron (2)
Full soldering setups (hot air guns, desolder tools, ect.)
A ton of soldering irons
More updated equipment in soldering area
More soldering irons
-New Lab Equipment Answers (8):
Updated test equipment
Brand new equipment
Equipment that actually works
Up-to-date electrical lab equipment so the already hard labs become do-able.
better labs? update equipment, stuff like that.
New equipment
Nicer ece equipment (i.e. oscilloscopes, DMMs etc.)
More high tech oscilloscopes and lab equipment
New oscilloscopes/multimeters/etc.
Latest lab equipments on the market.
-More Equipment/Access Answers (35):
Laser cutters (4)
Actually being able to get parts easily at the shop. That's the annoying thing, besides the work area near the shop being inadequate, it's discouraged to get parts for personal projects from the shop. Even when needed for a project, it's a pain.
Different level of equipments to be available to every student in the building.
EQUIPMENT
Every type of tool
Easily accessible fabrication (cnc, laser cutter, 3d printer, ect.)
Every single power tool known to man , as well as CNC machines, etc.
Access to machines, tools, and lab-bench equipment.
Easily accessible parts (easier and cheaper than the ece shop)
Table microscopes
Things for basic proof of concept (Arduinos, teensys, Ras Pis - not things people could take, but ones they could test a basic idea on to see if they could use that board before buying one)
Keysight oscilloscopes
Wood engraver
Power tools
Oscilloscope probes, parts bins/ accessible prototyping area
Alligator clips at every lab computer
Free parts office
Better scales for weighing out chemicals (UO lab).
HP/Tectronix everything.
Install surveillance cameras to monitor equipment.
Nice oscilloscopes/general bench top equipment
Drawers with components.
Lots of lab equipment
Lots of fabrication machines and materials. Also, nice tools.
Nicer scopes and open supplies(caps, resistors, wires) to students
auto transformers
cnc router
The aesthetics of foise with an unlimited amount of equipment
Stocked components shop that will actually give you the parts you want, without a bad attitude and unhelpful policies
Easily accessible equipment (like foisie allowing people to check out equipment) maybe the pick and place machine that's in Washburn (I have no idea if it works but no one knows how to use it and it is, or was, just sitting around. It'd be cool to learn how to use that)
ECE kits provided for students instead of being paid for individually to prevent mis-orders and stuff like that.
kit of various resistors, capacitors , and other circuit components
Large inventory of basic components

Work Space (46)

-Team Space Answers (7):
More advanced meeting rooms
More updated collaborative work areas.
Labs that are set up for team projects
More conference rooms
Group work stations
Large tables for homework/group work
Rooms that have big monitors to allow team projects to be more efficient.
-Lab Space Answers (15):
Lots of labs
Labs for open use with electronics components.
Computer labs
Well equipped labs
Maker spaces equipped with supplies (electrical components, breadboards, soldering irons, arduinos, LEDs, batteries, etc) that anyone can use to work on projects for class or just for fun. We could also host workshops there where we could do fun projects. Since classes are so condensed in our 7 week terms we often don?t get to do some of the fun applications of the material because there isn?t time. With a maker space that is specifically for experimentation we can learn new skills like soldering and using developer technology like arduino which is not necessarily taught in classes (especially the lower level ones), but are essential practical skills that will be valuable for MQP and in the workplace. Additionally, having this space along with workshops that will teach us how to use it could help younger students find a passion in ECE.
Labs with all new equipment
A project space with tons of materials available to students for free that has professors and grad students to help
A usable makerspace that is designed for students who actually do stuff, rather than being a form-over-function tour show piece. Even if it's just a room with rows of benches, storage, and scopes and power supplies that actually work, along with tools, soldering stuff, etc.
Lots of space to test robots
More usable maker space with some parts supplied.

Awesome lab space. With big windows.
Actual Maker Spaces
Easily available project space
A lab station complete with the most up-to-date equipment (soldering station, multimeters, etc.)
Computer lab station with essential ECE programs installed, such as Multisim, MATLAB, and some of the basic coding languages.
-Study Space Answers (10):
Tech suites (3)
Multiple areas to ask for guidance
Space for work
Large places to study.
More open area to study with comfy chairs
More study places
Enough workspace / tables to work at
Well lit work benches
-Variety of Spaces Answers (3):
Big tables to spread out on (in large spaces) as well as some smaller conference rooms
Many different work areas (some lounge style with more couch/ beanbag seating, some more formal Workstations with lab equipment, some cubical-type spaces for individual work, some areas with larger tables and a handful of chairs for group work)
-Lounge Space Answers (3):
A nice atmosphere and places to rest and move around after hours of sitting down.
gym and lounges to sleep so I never have to leave
Hangout spots
-Other Answers (8):
Open areas
Nap room!
Placed right across from the park, AK has one of the most privileged geographical locations on campus. If only it had more seating space with windows overlooking the park, AK would be the most popular building on campus.
The basement has so much potential but it smells and the carpet seats are kind of gross
Places to rest
Honestly just space to work with windows...
Quiet areas
Big rooms
A complete green screen room.

Other (71)

-Outlet Answers (8):
Lots of wall outlets (3)
Plugs EVERYWHERE
More outlets
More outlets
LOTS of power strips
Outlets in classrooms
-Printing Answers (8):
Free printing (3)
30 printers
Printers
More regular and functional printers
Free PRINTING!!!!
Free color printing
-Lighting Answers (6):
Better lighting
Functioning lights
Lots of natural lighting
Modern! Bright! Not as cold!
Bright lighting
-Tech Answers (13):
Tv's
MacBook desktops with parallel windows
Class capture database but better than Echo 360.
Fast computers
Computers with lots of GPU for machine learning.
Lots of computers and tv screens
Chromecasts
Smart devices everywhere!!
Indoor gps mapping (floor plan location) in real time Which lecture halls are being used for what classes/info sessions
Automated doors
New computers that actually we're fast
-Fun Addition Answers (12):
A slide (2)
Roombas
Firepole
Mainly, holographic displays
LOTS of secret tunnels
A piano
Solar Panels
Hallway security cameras.
LEDS EVERYWHERE
Projectors/ kiosks
Laser tag
-Maintenance Answers (12):
Working (efficient) elevators (2)
Clean (2)
Wifi that isn't crap.
Better carpet
Air fresheners because sometimes the labs can get a lil stinky due to kids staying in there for days at a time
Air exchange good enough that the central rooms don't smell like ass
Functional water fountains
Better desk setup in AK116
Better conditions for bathrooms, including a gender neutral bathroom would be great.
Non-squeaky desks and chairs
-Other Answers (12):
Graduate RBE labs moved from 8SP to AK
Everything AK already has
L desk, outlet, magnifying class, lots of space, tony stark style
Tables
Open access
Working tesla coil display, something to show off the dope ECE program
All the answers to everything, and tutors.
Think of Washburn, Foisie and collablab all together
Walls that can be moved/stored to transform the space by the push of a button
More tables.
AK just scaled up.
Infinity pool on the roof

Not Real Answers (3)

Piles of free money for student loans
1
Marlene

IV. Question Four: What does Atwater Kent have that doesn't work?

Broken Tech/Equipment (41)

-LED Display Answers (9):
The LED display
That weird display thing above this table. (Matrix in Pumpkin Lounge)
The circuit board in the pumpkin lounge
Half of the LED panels
The displays in the front. It’s a shame they’re not capable of operating all of the time.
The really cool LED panels
The LED matrices.
The circuit board display in the pumpkin lounge
The screen thingy
-Oscilloscope Answers (13):
Broken oscilloscopes
Some of the oscilloscopes in 227 & 317
Some oscilloscopes, function generators, etc. throughout the labs that aren’t working (e.g one oscilloscope in AK 113).
90% of oscilloscopes
Equipment oscilloscopes etc
A few scopes
A few lab scopes aren’t working properly in AK113.
Oscilloscopes are always questionable whether or not the readings are accurate, and finding probes is a pain.
Scopes.
Half of the oscilloscopes in the labs
A lot of the oscilloscopes
Oscilloscopes, DMM’s
Older oscilloscopes with lower resolution
-Computer Answers (3):
The computer with the constantly red screen in AK 113
Computers can be slow
One computer in AK113 is broken (constantly red screen).
-Other Answers (16):
Some lab devices are broken
Soldering stations are sub-optimal
Not a lot of work space tools
Blackboards are obsolete.
Most of the Lab Equipment
Soldering machines, printers, lab equipment
Printers, monitors
The lab dc power supplies are garbage.
A lot of the ece equipment is really old and sometimes doesn?t work
As mentioned, the printers and a lot of oscilloscopes are missing parts of their traces, small stuff in the labs like that is really common. Also, the students :P
Easy and cheap access to parts
Annoying alligator clips on the power supplies
Soldering iron, desoldering stations
The printer in 113 prints black streaks on papers
The shop.
The work area outside the shop.

Maintenance (36)

-AC Answers (6):
Better AC!!!!!
AC
The AC in room 232
232 is still somehow unairconditioned
Although they work, the air conditioning units are extremely noisy and annoying.
Too much heat in the winter
-Carpet Answers (2):
Musty carpets
Gross smells and ugly carpet
-Side Door Answers (3):
The small entrance on the right side of the building does not unlock for students after-hours. There is no difference between entering through the main door and this one so why not enable access through this door as well?
The side door doesn’t let you in at night
The side door alarm
-Water Fountain Answers (6):
Water fountains
Some water fountains i think? haven?t tried them since last semester though
Water fountains (besides the one)
Some water fountains
Water fountain 2nd floor by bathroom
Water fountains
-Chair/Desk Answers (8):
The damn chairs!
I haven’t been in AK in almost a year. My card access at night doesn’t work, but I don’t think it should. I think there are a few broken chairs?
Poor desks in second floor rooms
The rolling chairs with the wide bases that you can’t move without making a whole row of people slide over
A few lab chairs are dirty or don’t work.
More chairs and tables than we know what to do with
The chairs in lecture halls 219 and 233 suck.
-Other Answers (11):
AK also has this huge, magnificent door leading to the park that was closed off and turned into an awkward "working/seating space." Bring back the door.
Orange walls are terrible
Windows along back hallway don’t work
Mens bathroom door first floor has been kinda messed up (closes way too violently)
The vestibule in AK does not seem to serve its purpose. The second set of doors is always open anyways. Could that space be remodeled to remove that left side door for easier access to the stair-case?
Doors
Bathrooms are kinda jank
The custodians don’t seem to clean the tables on the second floor
Elevator
The elevator is extremely slow
A tilted stairwell going up to the third floor

Labs (12)

The ECE labs

Some lab equipment

Most of the labs are missing a lot of wires and parts

Many of the labs are missing key test equipment

Lab equipment is iffy sometimes

Having taken 5 RBE courses there along with other ECEs, I wouldn’t say things don’t work but the building could overall be redone a lot nicer. The labs don’t seem to have good air circulation which is very noticeable when 50 RBE students are in the lab sweating and in general i think its just an ugly building compared to the newer ones. If I had money to donate to the school Id have them tear down AK and build a much nicer lab for the RBE and ECE students

The soldering lab equipment is dated

Horrible lab equipment

Old oscilloscopes in AK 113

The organization of the oscilloscope, multimeter, and function generator wires

The way the labs are designed and it has so many inaccessible resources like boards etc

Last year the labs never had enough probe leads and cables

Some of the lab equipment is broken

RBE lab (pre foise) non existent, ECE labs can get VERY busy

Other (26)

-Work Area Answers (6):

Space to study. There’s only one good group study room and 1 person always takes it up. More tables and seating

Most engineering school actually have ECE based makerspaces. The Foisie one basically has nothing to do with ECE and hobbyist electronics. The current shop setup doesn’t work, and discourages people from working on independent projects.

Seating/layout of pumpkin lounge doesn?t work that great, ex: is there a reason we have barstool chairs ??? it makes it harder to plug my laptop in on the outlet at the bottom of the wall and harder to grab things out of my backpack without physically having to get out of the chair to get it... plus the space is not flexible bc i can?t move any of the chairs to the shorter tables on the side wall or into the lab bc they are so tall.

A few more areas for group work

Cramped spaces with small tables that are too close together (in narrow hallways), big screens that are not easily visible to everyone who walks in the building, weird lighting (at least the way it looks off the orange walls is weird)

NOT ENOUGH WORK SPACE

-Old Answers (7):

It’s old, cramped, and doesn’t have enough classroom space.

It?s visually very dated

Everything works, it’s just a little out of date

It’s old

Just feels old

Basement is scary

It just looks older and less inviting

-Architecture Answers (4):

It’s hard to find everything

Weird layout, too many stairs, rooms inside of rooms, hallways

Hallway seating is uncomfortable and lacks enough outlets

There is so much traffic when class gets out it can take 10 minutes to get into AK 116 if there is a large lecture in there. The work space that exists does not have enough outlets. Also the seating in the pumpkin lounge is SO UNCOMFORTABLE and the orange is just abrasive. TBH the best place to work is in the basement and it is kinda weird and creepy but at least it has whiteboards and meeting tables

-Other Answers (9):

Not sure because it hasn’t been a place I thought I would see myself relaxing in.

The floor plan

The shop. Everything about the shop.

Tour groups, the RBE department, memes

If there is only one thing that you can do, find out how to reduce the feeling of claustrophobia in the building. Once you’re inside it feels like time doesn’t pass. It’s uncomfortable.

It need more water cooler. And I think the color red is repeated too much, we should try some other colors.

Non-existent tables do not work.

Projects, tables, scopes

Paint the building in bright colors, it’s very depressing after spending 4 years in there. White is a good suggestion.

No input (18)

IDK, don?t go there too often
ECE’s
1
I don’t know, never had any classes there
N/A - Have not been there long enough to know
Nothing this year
Nothing I am aware of
Professor Hakim
Nothing
Nothing that I am aware of... I mean..
No idea
Students
Idk
Seems everything that i have used works
Robot labs don?t belong in ece ..... Arduino 101 = rbe degree
No Response
Other
Nothing, really

V. Question Five: What is Atwater Kent missing?

Better Equipment (14)

More cool old electric devices
Faster computers
Logic analyzers
Decent soldering irons.
I’m not sure if we have any hot air solder tools
Equipment
Properly setup soldering stations.
The aforementioned PCB printers and regular printers
Good alligator clips on the power supplies, open access to a lab with fancy test equipment (nice oscilloscopes)
A lot of soldering irons that students can use
The building lacks many soldering irons and prototyping resources.
Better lab equipment
Modern scopes.
3D printers

Living Essentials (19)

-Food Answers (9):
Food (2)
Maybe a little kitchenette would be good for them to have (or just an electric tea kettle for ramen)
24/7 pizza, FREE pizza
Kitchen
Dining Place
Food besides pop tarts for late-night lab work
A hot water maker.
The water boiler (hot water maker) is gone :((IEEE Lounge)
-Sleep/Living Answers (4):
Showers and beds for the poor rbe’s who live there.
Sleeping pods
Showers
Nap areas
-Other Answers (6):
A makeover!! more food, more water fountains, nicer tables/more tables and chairs. Sofas also would be great.
A cafe and beanbags
Gender neutral bathroom
Good clean bathrooms
More filtered water stations
A full water station in the third floor. (The water fountain there cannot refill water bottles)

Building Updates (41)

-AC Answers (6):
Better AC control
Windows that stay open
Proper air circulation
Ac - we raise the age salary by so much
Better AC!!!!
More airflow
-Door Answers (4):
Access to the back of the building (to institute road)
Quality Doors
IF you’re asking about aesthetics, replacing the front doors to something cleaner, new doors would make the building look better.
Card access to the side door for students.
-Outlet Answers (6):
More outlets, screens for student use
Outlets in classrooms when I’m trying to participate in databases (or doing mqp instead of taking notes)
Enough outlets on the second floor table
Outlets
More outlets in lecture halls
Electrical outlets
-Tech/Equipment Answers (7):
You guys should make the control thing near the bottom of the stairs work . Would be fun to play around with that.
AK could probably use some more circuit components, such as MOSFETs and BJTs for the Microelectronic classes. It has most of the listed items from my dream tech building, but some of these items are either broken or out of date.
Printers that work consistently
Enough lab computers for everyone in the RBE lab.
New equipment
Cool Labs and working Equipment
A new age tech feel, everything is very old school

-Furniture Answers (6):

New Chairs/Furniture

Standing desks

Tables to sit at on the 3rd floor.

Supportive chairs!! Please fix them!!

More tables to do work

More tables for working, I hate pumpkin lounge.

-Other Answers (12):

Whiteboards (or transparent wall pieces for working out problems)

Little cramped/ more space

A renovation

A more updated look, it looks sad and depressing a bit when going in there

A FUNCTIONAL ECE SHOP FOR PEOPLE INTERESTED IN GETTING PARTS AND WORKSPACES FOR INDEPENDENT PROJECTS.

Easy access to stuff

Natural light

Better decoration

Clear sign for each classroom .

A serious update in aesthetic quality

Few more little spots to do group work/hang out and study/do homework in groups

A larger ECE shop

Building Feature Ideas (8)

A way for visitors to interact with project/ devices (similar to the door lock thing)

More soldering events and learning to design a PCB

A giant fish tank with octopus and jellyfish to inspire soft robotics.

A vending machine for common electronic parts

A giant screen like Foisie's?

Machines/tools to make pcbs and information about how to use the ece shop

Instructions/rules/lab monitors for the soldering lab

More led art or electronics art would be dope

Work Space (37)

-Study Space Answers (11):

Comfortable work space

Study lounge

Open concept space where students can stay in a quiet environment and be productive.

More areas to do work

More spaces to do hw. tables are limited and people don?t like sharing

More places that students would want to work, more accessible tools

We have a lot of tables to study at which is cool, but there are no private areas where people can go if they don?t want to study in the hallway.

Good study spaces.

Better study area (idk I?m in Goddard)

More study spaces, smart boards

More study places

-Group Space Answers (5):

More places for group meetings

No collaboration spaces that people can reserve to work in with big tables and whiteboards

More collaborative areas for people to meet in groups, also more whiteboards

Collaboration rooms

A collaborative space would be nice.

-Lounge Answers (4):

A more open/spacious, fun, and social area for friends to come together and study or perhaps just chat while having a coffee.

Something like the Physics lounge but for ECE, free printing, pcb printer

My charming personality, study lounge near windows on a top floor.

It's missing a cooler aesthetic to some of its lounge areas... I mean the main lounge is pretty cool looking but non-ECE majors might not even think of going upstairs

-Lab Answers (7):

An area to work on larger projects

Somewhere that you can work on projects not class related.

An actually usable makerspace /project workspace. Storage for projects. A way to get parts easily for projects. MQP space that has better than 2 channel 2002-era scopes.

Big windows in labs.

Comfortable labs

A daily lab maintenance team

An open (metaphorically) Lab working space that people can work on projects in and get electronics related help or ask questions etc. Boiled down: an ece focused makerspace that doesn't supply pipe solder (@foisie, seriously go ask for a soldering kit and it's acid rosin core pipe solder, both corrosive and too big for small electronics)

-Other Answers (10):

Classroom space

A common library space specific to ECE

Areas to showcase personal student projects in ECE

Current and updated spaces

Dedicated HKN space.

Space

Faster workstations.

Large open space

You should change that little area on the first floor back hall where the windows are and make it into a sleeping room. We are always so tired at night finishing labs, that would be great

A modern atmosphere. It feels very musty and old (which it is so)

Other (6)

Easier way to access research for students. It’s hard to know if any professors are ever looking.
Free printing (2)
Not enough staircases
New feel
It’s an old building, a modern upgrade would be nice

Not Useful (8)

1
Best lecture halls are here, can’t think of anything
I’m not sure
Other
Honestly though, in terms of format of the rest of the building, AK is pretty great. I love it here. I hope they don’t ruin it with the weird and funky furniture like they’ve been adding in the library and Foisie.
Nothing
Custodians
New Building

Appendix F

Original Goals List

F.1 Project Goals

- Something that lasts for years to come.
- Aesthetically pleasing.
- Changes people's views on AK.
- Change more than one aspect of AK.
- Make something easy to use.
- Not too much power consumption.
- Create something innovative, something WE thought of.
- Get an A.
- Finish before deadline.
- Impress Mcneill.
- Have something good to put on resume.
- Create plans for renovation for Atwater Kent
- Restore the Pumpkin Lounge matrix displays to full functionality, or replace them if something better comes along
- Better demonstrate the capabilities and opportunities of studying ECE or robotics at WPI to visitors
- Our project means something to all students
- Inspire creativity
- Provide a welcoming environment
- Embrace the history of the building
- Promote modern technology and innovation
- Offer a medium for AK community members to interact through interesting things to look at around the building

F.2 Technical/Learning Goals

- Apply analog/semiconductors to project for practice.
- Final paper is easy to follow and read for future projects that wish to use our paper.
- Design and implementation of analog circuitry
- Create an interesting application of signal processing techniques

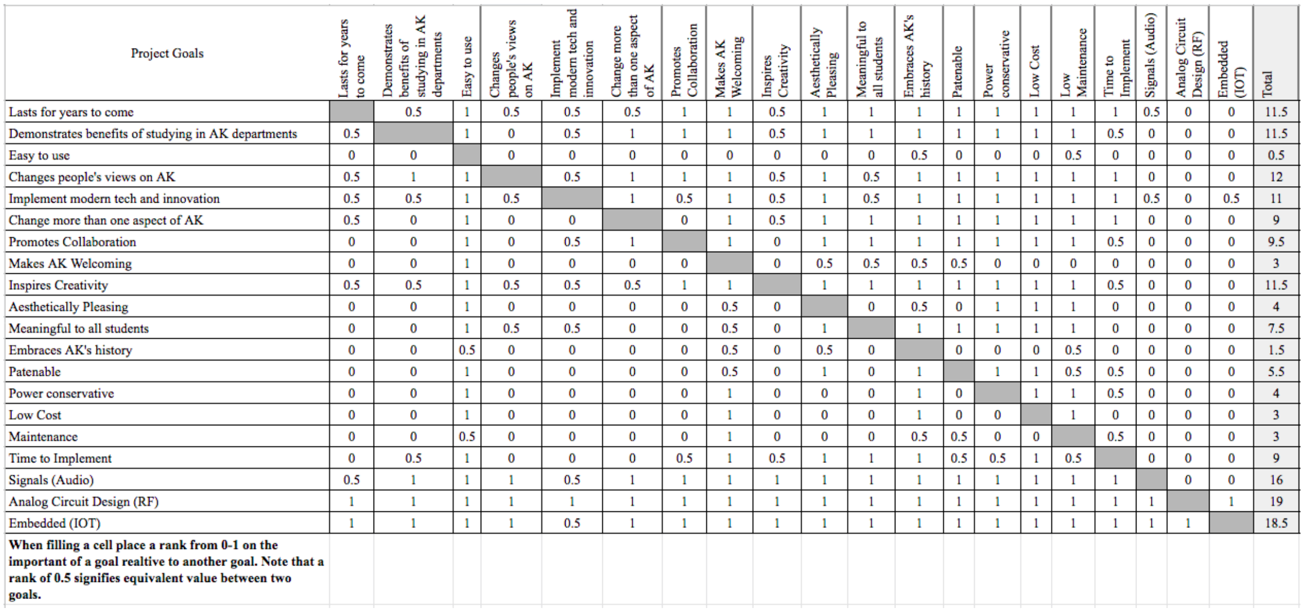
- Engage in IOT and embedded systems development
- Hone practical electrical engineering skills (e.g. soldering, PCB design)
- Meet all deadlines
- Develop a professional technical report
- Learn about wireless communication
- Apply knowledge of C and learn/ apply studied languages such as Python, Java, and C++ protocols
- Apply computer engineering skills to make more efficient real-time code
- Gain experience with the interconnect between hardware and software and their application
- More experience in programming
- Design and implement a filter.
- Micro controller experience
- Digital signal processing techniques for more complex challenges.
- RF Circuit
- Voice Recognition
- Create interactive “smart’ device
- Implement Display LEDs or LCDs
- Touch Screens

F.3 Social Goals

- Everyone equally listens and shares, no ideas go unheard.
- Successful use of team structure/scrum.
- Improved team skills.
- Improved networking skills (especially if we get a sponsor).
- Form a coherent team that performs
- Improve our networking skills
- Grow our work ethic
- Learn to accept and admit failure to promote team progression

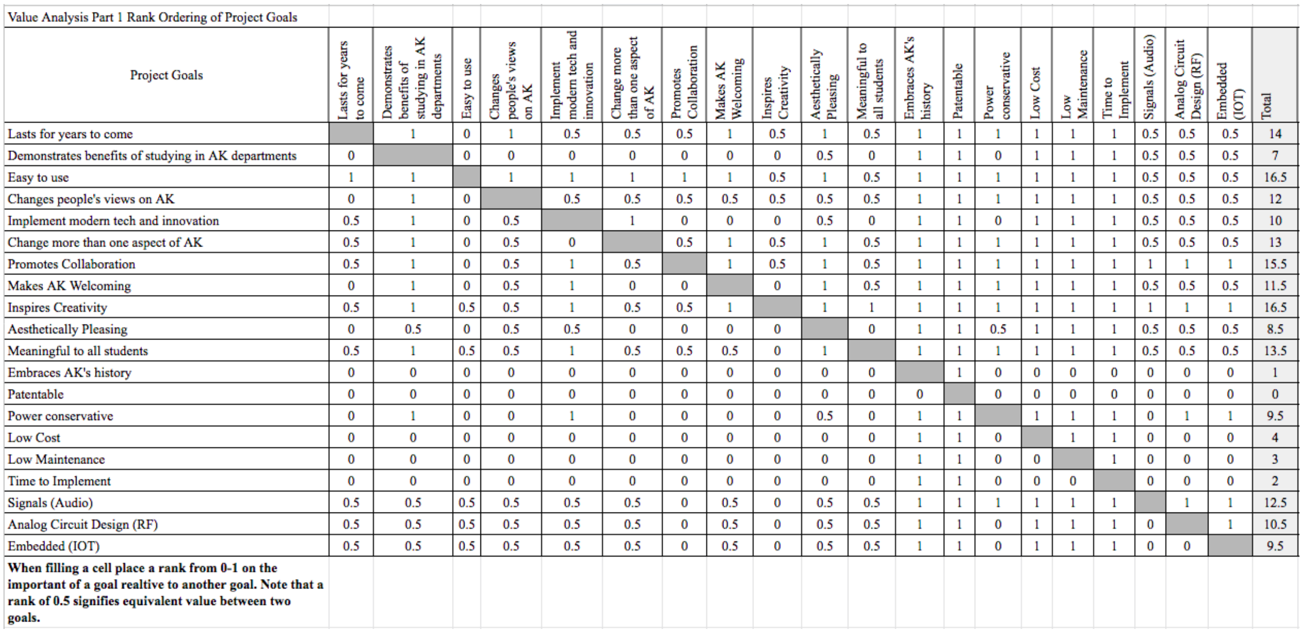
Appendix G

The Big Three Decision Matrix



Project Goals	Lasts for years to come	Demonstrates benefits of studying in AK departments	Easy to use	Changes people's views on AK	Implement modern tech and innovation	Change more than one aspect of AK	Promotes Collaboration	Makes AK Welcoming	Inspires Creativity	Aesthetically Pleasing	Meaningful to all students	Embraces AK's history	Patenable	Power conservative	Low Cost	Low Maintenance	Time to Implement	Signals (Audio)	Analog Circuit Design (RF)	Embedded (IOT)	Total
Lasts for years to come		0.5	1	0.5	0.5	0.5	1	1	0.5	1	1	1	1	1	1	1	1	0.5	0	0	11.5
Demonstrates benefits of studying in AK departments	0.5		1	0	0.5	1	1	1	0.5	1	1	1	1	1	1	1	0.5	0	0	0	11.5
Easy to use	0	0		0	0	0	0	0	0	0	0	0.5	0	0	0	0.5	0	0	0	0	0.5
Changes people's views on AK	0.5	1	1		0.5	1	1	1	0.5	1	0.5	1	1	1	1	1	1	0	0	0	12
Implement modern tech and innovation	0.5	0.5	1	0.5		1	0.5	1	0.5	1	0.5	1	1	1	1	1	1	0.5	0	0.5	11
Change more than one aspect of AK	0.5	0	1	0	0		0	1	0.5	1	1	1	1	1	1	1	1	0	0	0	9
Promotes Collaboration	0	0	1	0	0.5	1		1	0	1	1	1	1	1	1	1	0.5	0	0	0	9.5
Makes AK Welcoming	0	0	1	0	0	0	0		0	0.5	0.5	0.5	0.5	0	0	0	0	0	0	0	3
Inspires Creativity	0.5	0.5	1	0.5	0.5	0.5	1	1		1	1	1	1	1	1	1	0.5	0	0	0	11.5
Aesthetically Pleasing	0	0	1	0	0	0	0	0.5	0		0	0.5	0	1	1	1	0	0	0	0	4
Meaningful to all students	0	0	1	0.5	0.5	0	0	0.5	0	1		1	1	1	1	1	0	0	0	0	7.5
Embraces AK's history	0	0	0.5	0	0	0	0	0.5	0	0.5	0		0	0	0	0.5	0	0	0	0	1.5
Patenable	0	0	1	0	0	0	0	0.5	0	1	0	1		1	1	0.5	0.5	0	0	0	5.5
Power conservative	0	0	1	0	0	0	0	1	0	0	0	1	0		1	1	0.5	0	0	0	4
Low Cost	0	0	1	0	0	0	0	1	0	0	0	1	0	0		1	0	0	0	0	3
Maintenance	0	0	0.5	0	0	0	0	1	0	0	0	0.5	0.5	0	0		0.5	0	0	0	3
Time to Implement	0	0.5	1	0	0	0	0.5	1	0.5	1	1	1	0.5	0.5	1	0.5		0	0	0	9
Signals (Audio)	0.5	1	1	1	0.5	1	1	1	1	1	1	1	1	1	1	1	1		0	0	16
Analog Circuit Design (RF)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	19
Embedded (IOT)	1	1	1	1	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1		18.5
When filling a cell place a rank from 0-1 on the important of a goal realtive to another goal. Note that a rank of 0.5 signifies equivalent value between two goals.																					

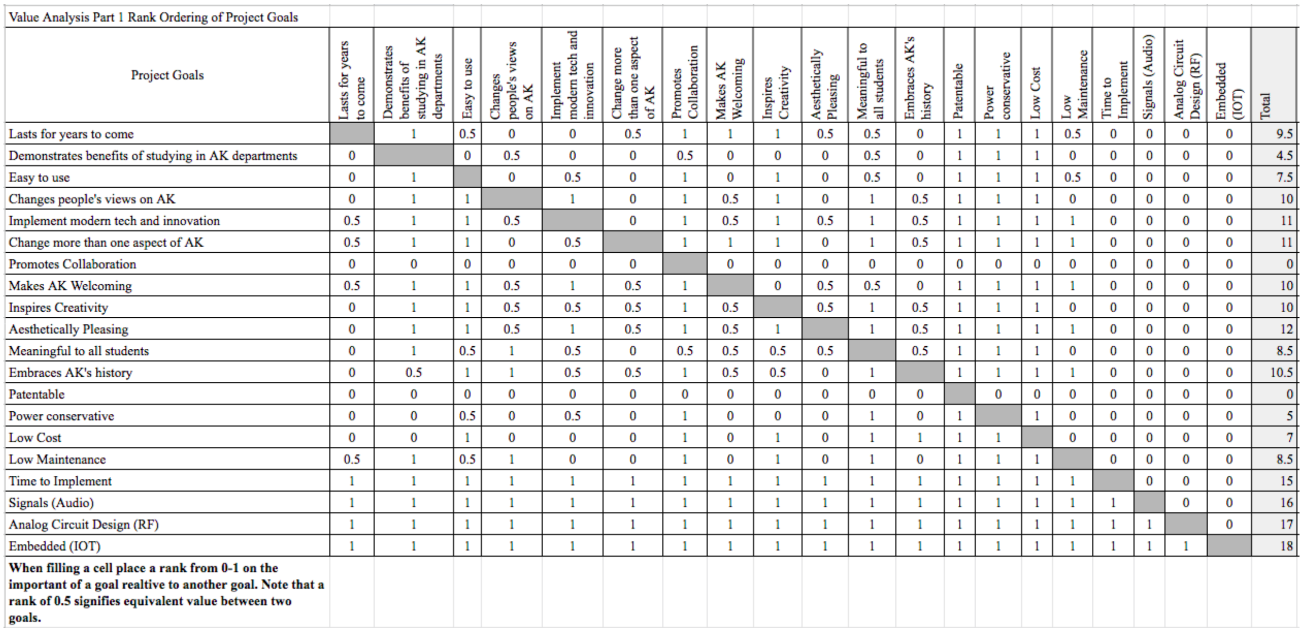
Table G.1: Shannon's Goals



Value Analysis Part 1 Rank Ordering of Project Goals

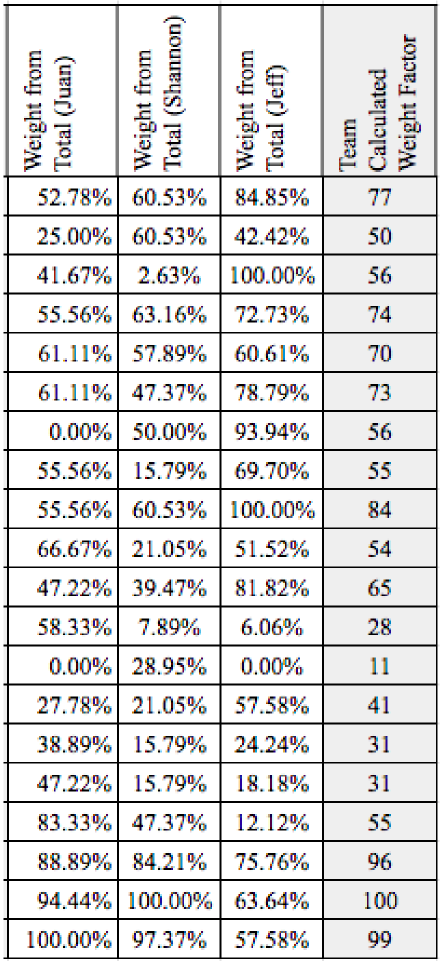
Project Goals	Lasts for years to come	Demonstrates benefits of studying in AK departments	Easy to use	Changes people's views on AK	Implement modern tech and innovation	Change more than one aspect of AK	Promotes Collaboration	Makes AK Welcoming	Inspires Creativity	Aesthetically Pleasing	Meaningful to all students	Embraces AK's history	Patentable	Power conservative	Low Cost	Low Maintenance	Time to Implement	Signals (Audio)	Analog Circuit Design (RF)	Embedded (IOT)	Total
Lasts for years to come		1	0	1	0.5	0.5	0.5	1	0.5	1	0.5	1	1	1	1	1	1	0.5	0.5	0.5	14
Demonstrates benefits of studying in AK departments	0		0	0	0	0	0	0	0	0.5	0	1	1	0	1	1	1	0.5	0.5	0.5	7
Easy to use	1	1		1	1	1	1	1	0.5	1	0.5	1	1	1	1	1	1	0.5	0.5	0.5	16.5
Changes people's views on AK	0	1	0		0.5	0.5	0.5	0.5	0.5	0.5	0.5	1	1	1	1	1	1	0.5	0.5	0.5	12
Implement modern tech and innovation	0.5	1	0	0.5		1	0	0	0	0.5	0	1	1	0	1	1	1	0.5	0.5	0.5	10
Change more than one aspect of AK	0.5	1	0	0.5	0		0.5	1	0.5	1	0.5	1	1	1	1	1	1	0.5	0.5	0.5	13
Promotes Collaboration	0.5	1	0	0.5	1	0.5		1	0.5	1	0.5	1	1	1	1	1	1	1	1	1	15.5
Makes AK Welcoming	0	1	0	0.5	1	0	0		0	1	0.5	1	1	1	1	1	1	0.5	0.5	0.5	11.5
Inspires Creativity	0.5	1	0.5	0.5	1	0.5	0.5	1		1	1	1	1	1	1	1	1	1	1	1	16.5
Aesthetically Pleasing	0	0.5	0	0.5	0.5	0	0	0	0		0	1	1	0.5	1	1	1	0.5	0.5	0.5	8.5
Meaningful to all students	0.5	1	0.5	0.5	1	0.5	0.5	0.5	0	1		1	1	1	1	1	1	0.5	0.5	0.5	13.5
Embraces AK's history	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	0	0	0	0	1
Patentable	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
Power conservative	0	1	0	0	1	0	0	0	0	0.5	0	1	1		1	1	1	0	1	1	9.5
Low Cost	0	0	0	0	0	0	0	0	0	0	0	1	1	0		1	1	0	0	0	4
Low Maintenance	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0		1	0	0	0	3
Time to Implement	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0		0	0	0	2
Signals (Audio)	0.5	0.5	0.5	0.5	0.5	0.5	0	0.5	0	0.5	0.5	1	1	1	1	1	1		1	1	12.5
Analog Circuit Design (RF)	0.5	0.5	0.5	0.5	0.5	0.5	0	0.5	0	0.5	0.5	1	1	0	1	1	1	0		1	10.5
Embedded (IOT)	0.5	0.5	0.5	0.5	0.5	0.5	0	0.5	0	0.5	0.5	1	1	0	1	1	1	0	0		9.5
When filling a cell place a rank from 0-1 on the important of a goal realtive to another goal. Note that a rank of 0.5 signifies equivalent value between two goals.																					

Table G.2: Jeff's Goals



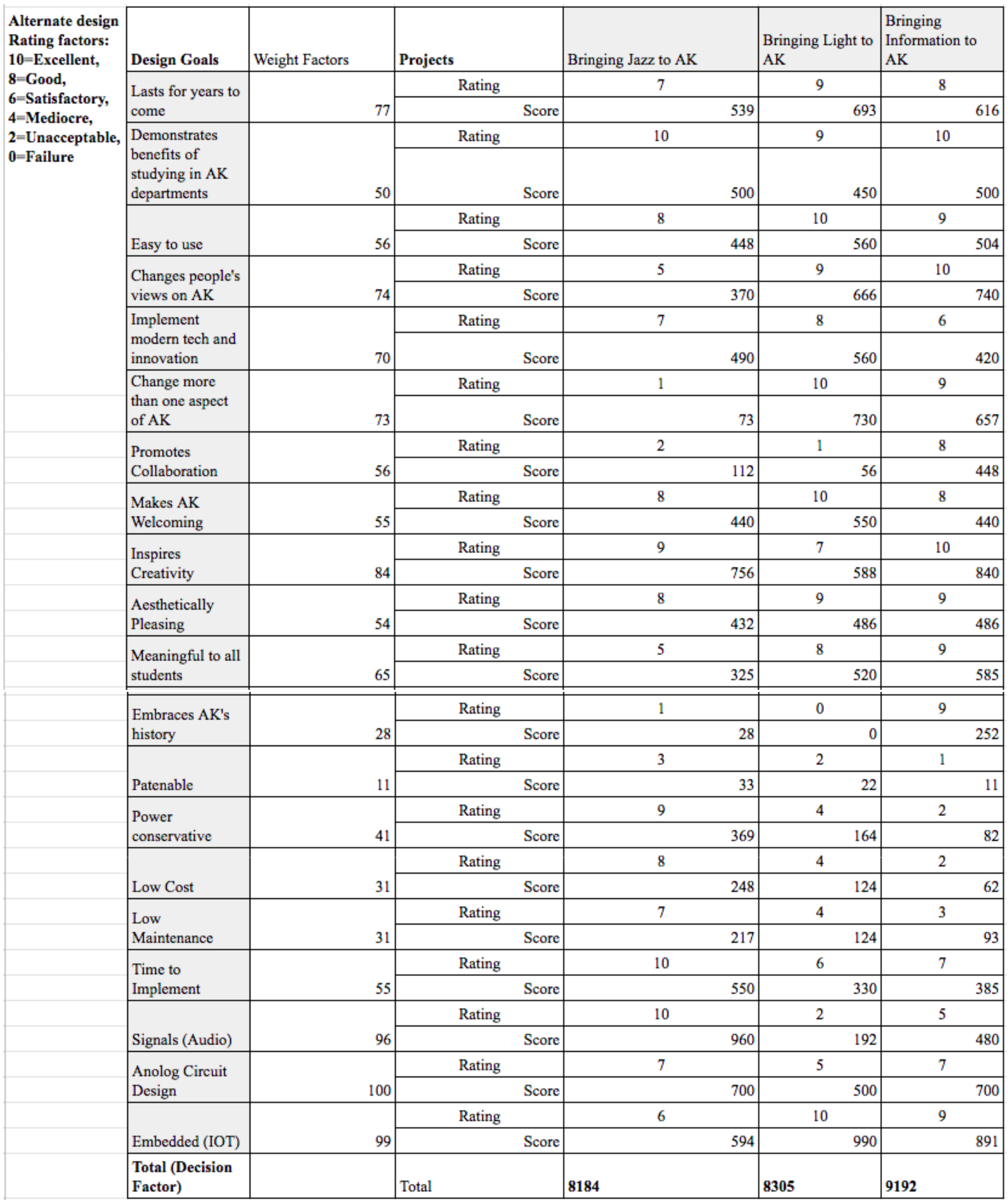
Value Analysis Part 1 Rank Ordering of Project Goals																					
Project Goals	Lasts for years to come	Demonstrates benefits of studying in AK departments	Easy to use	Changes people's views on AK	Implement modern tech and innovation	Change more than one aspect of AK	Promotes Collaboration	Makes AK Welcoming	Inspires Creativity	Aesthetically Pleasing	Meaningful to all students	Embraces AK's history	Patentable	Power conservative	Low Cost	Low Maintenance	Time to Implement	Signals (Audio)	Analog Circuit Design (RF)	Embedded (IOT)	Total
Lasts for years to come		1	0.5	0	0	0.5	1	1	1	0.5	0.5	0	1	1	1	0.5	0	0	0	0	9.5
Demonstrates benefits of studying in AK departments	0		0	0.5	0	0	0.5	0	0	0	0.5	0	1	1	1	0	0	0	0	0	4.5
Easy to use	0	1		0	0.5	0	1	0	1	0	0.5	0	1	1	1	0.5	0	0	0	0	7.5
Changes people's views on AK	0	1	1		1	0	1	0.5	1	0	1	0.5	1	1	1	0	0	0	0	0	10
Implement modern tech and innovation	0.5	1	1	0.5		0	1	0.5	1	0.5	1	0.5	1	1	1	1	0	0	0	0	11
Change more than one aspect of AK	0.5	1	1	0	0.5		1	1	1	0	1	0.5	1	1	1	1	0	0	0	0	11
Promotes Collaboration	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0
Makes AK Welcoming	0.5	1	1	0.5	1	0.5	1		0	0.5	0.5	0	1	1	1	1	0	0	0	0	10
Inspires Creativity	0	1	1	0.5	0.5	0.5	1	0.5		0.5	1	0.5	1	1	1	0	0	0	0	0	10
Aesthetically Pleasing	0	1	1	0.5	1	0.5	1	0.5	1		1	0.5	1	1	1	1	0	0	0	0	12
Meaningful to all students	0	1	0.5	1	0.5	0	0.5	0.5	0.5	0.5		0.5	1	1	1	0	0	0	0	0	8.5
Embraces AK's history	0	0.5	1	1	0.5	0.5	1	0.5	0.5	0	1		1	1	1	1	0	0	0	0	10.5
Patentable	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
Power conservative	0	0	0.5	0	0.5	0	1	0	0	0	1	0	1		1	0	0	0	0	0	5
Low Cost	0	0	1	0	0	0	1	0	1	0	1	1	1	1		0	0	0	0	0	7
Low Maintenance	0.5	1	0.5	1	0	0	1	0	1	0	1	0	1	1	1		0	0	0	0	8.5
Time to Implement	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		0	0	0	15
Signals (Audio)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		0	0	16
Analog Circuit Design (RF)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		0	17
Embedded (IOT)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		18
When filling a cell place a rank from 0-1 on the important of a goal realtive to another goal. Note that a rank of 0.5 signifies equivalent value between two goals.																					

Table G.3: Juan's Goals



Weight from Total (Juan)	Weight from Total (Shannon)	Weight from Total (Jeff)	Team Calculated Weight Factor
52.78%	60.53%	84.85%	77
25.00%	60.53%	42.42%	50
41.67%	2.63%	100.00%	56
55.56%	63.16%	72.73%	74
61.11%	57.89%	60.61%	70
61.11%	47.37%	78.79%	73
0.00%	50.00%	93.94%	56
55.56%	15.79%	69.70%	55
55.56%	60.53%	100.00%	84
66.67%	21.05%	51.52%	54
47.22%	39.47%	81.82%	65
58.33%	7.89%	6.06%	28
0.00%	28.95%	0.00%	11
27.78%	21.05%	57.58%	41
38.89%	15.79%	24.24%	31
47.22%	15.79%	18.18%	31
83.33%	47.37%	12.12%	55
88.89%	84.21%	75.76%	96
94.44%	100.00%	63.64%	100
100.00%	97.37%	57.58%	99

Table G.4: Averaged Goals



Alternate design Rating factors: 10=Excellent, 8=Good, 6=Satisfactory, 4=Mediocre, 2=Unacceptable, 0=Failure	**Design Goals**	Weight Factors	**Projects**	Bringing Jazz to AK	Bringing Light to AK	Bringing Information to AK
	Lasts for years to come	77	Rating	7	9	8
			Score	539	693	616
	Demonstrates benefits of studying in AK departments	50	Rating	10	9	10
			Score	500	450	500
	Easy to use	56	Rating	8	10	9
			Score	448	560	504
	Changes people's views on AK	74	Rating	5	9	10
			Score	370	666	740
	Implement modern tech and innovation	70	Rating	7	8	6
			Score	490	560	420
	Change more than one aspect of AK	73	Rating	1	10	9
			Score	73	730	657
	Promotes Collaboration	56	Rating	2	1	8
			Score	112	56	448
	Makes AK Welcoming	55	Rating	8	10	8
			Score	440	550	440
	Inspires Creativity	84	Rating	9	7	10
			Score	756	588	840
	Aesthetically Pleasing	54	Rating	8	9	9
			Score	432	486	486
	Meaningful to all students	65	Rating	5	8	9
			Score	325	520	585
	Embraces AK's history	28	Rating	1	0	9
			Score	28	0	252
	Patenable	11	Rating	3	2	1
			Score	33	22	11
	Power conservative	41	Rating	9	4	2
			Score	369	164	82
	Low Cost	31	Rating	8	4	2
			Score	248	124	62
	Low Maintenance	31	Rating	7	4	3
			Score	217	124	93
	Time to Implement	55	Rating	10	6	7
			Score	550	330	385
	Signals (Audio)	96	Rating	10	2	5
			Score	960	192	480
	Anolog Circuit Design	100	Rating	7	5	7
			Score	700	500	700
	Embedded (IOT)	99	Rating	6	10	9
			Score	594	990	891
	Total (Decision Factor)		Total	**8184**	**8305**	**9192**

Table G.5: Decision Matrix

Appendix H

Color Organ Arduino Code

```
#define COLOR_CYCLE 15000   // length of color cycle in [units]
// #define MASTER             // determine whether this is a master or slave board
#define SLAVE

float color_base[3] = {1, 0, 1};                                               //starting color
float color_target[3] = {1, 0, 0};                                             //color to cycle to
volatile float color_array[3] = {color_base[0], color_base[1], color_base[2]}; //current display color
int color_output[3] = {0, 0, 0};                                               //PWM output
volatile int c = 0;                                                            //counter for color change
int vol = 0;

void setup() {
//configure analog input
pinMode(A1, INPUT);

#ifdef MASTER
pinMode(2, OUTPUT);
#endif

#ifdef SLAVE
pinMode(2, INPUT);
attachInterrupt(digitalPinToInterrupt(2), UpdateColor, RISING);
#endif

//configure and initialize gpio 6,7, and 8
pinMode(9, OUTPUT);
analogWrite(9, color_output[0]);
pinMode(10, OUTPUT);
analogWrite(10, color_output[1]);
pinMode(11, OUTPUT);
analogWrite(11, color_output[2]);

//begin serial device
Serial.begin(115200);
}

void loop() {
//take a reading
vol = analogRead(A1)/4;

//assign brightness based on analog value * color array
color_output[0] = vol*color_array[0];
color_output[1] = vol*color_array[1];
color_output[2] = vol*color_array[2];
//transmit PWM signal
analogWrite(9, color_output[0]);
analogWrite(10, color_output[1]);
analogWrite(11, color_output[2]);

//change color
#ifdef MASTER
UpdateColor();
digitalWrite(2, HIGH);
digitalWrite(2, LOW);
#endif
}

void UpdateColor() {
if (c<COLOR_CYCLE){
color_array[0] += (color_target[0]-color_base[0])/COLOR_CYCLE;
color_array[1] += (color_target[1]-color_base[1])/COLOR_CYCLE;
color_array[2] += (color_target[2]-color_base[2])/COLOR_CYCLE;
c++;
} else if (c<COLOR_CYCLE*2){
color_array[0] += -(color_target[0]-color_base[0])/COLOR_CYCLE;
color_array[1] += -(color_target[1]-color_base[1])/COLOR_CYCLE;
color_array[2] += -(color_target[2]-color_base[2])/COLOR_CYCLE;
c++;
} else {
color_array[0] = color_base[0];
color_array[1] = color_base[1];
color_array[2] = color_base[2];
c=0;
}
}
```