

University of Central Florida

## Frequency Based Instrumental Lights

### Group 25:

Leith Rabah - CPE  
Stefan Gonzalez - EE  
Carson Warner - CPE  
Anja Frohock - EE

### Sponsor:

Dr. Richard Leinecker

# Table of Contents

1. Executive Summary.....	1
2. Project Description.....	3
1. Motivation.....	3
2. Goal.....	3
3. Project Narrative Description.....	3
4. Requirements & Specifications.....	5
5. Block Diagrams.....	7
1. Hardware Diagram.....	8
2. Software Diagram.....	9
3. House of Quality.....	10
4. Illustrative Prototype.....	11
3. Research & Technology Comparisons.....	12
1. Microphones.....	12
1. History of Microphones.....	12
2. Shotgun Microphone.....	14
3. Contact Microphone.....	16
4. Clip-On Microphone.....	17
5. Shotgun vs Contact vs Clip-On.....	18
2. LED Strips.....	19
1. Addressable LED Strips.....	19
2. SMD LED Strips.....	20
3. COB LED Strips.....	20
4. SMD vs COB.....	21
3. Power Supply.....	22
1. Rechargeable Lithium-Ion Battery Pack.....	22
2. Lithium Polymer Battery with Powerboost.....	23
3. Raspberry Pi Battery Pack.....	24
4. Raspberry Pi Battery Hat.....	24
5. USB Wall Charger.....	25
6. Power Supply Comparisons.....	26
4. DC-DC Converter.....	26
1. Low-Power Mode.....	28
5. Single Board Computers.....	29
1. Raspberry Pi 4, Model B.....	29
2. Raspberry Pi 3, Model B+.....	32
3. Nvidia Jetson.....	33
4. Raspberry Pi 4 vs Pi 3 vs Nvidia Jetson.....	34
6. Display Technologies.....	35
1. Liquid Crystal Displays.....	36
2. Light Emitting Diode Displays.....	37
3. LCD vs LED Displays.....	38

7. Audio Transmission.....	39
1. Bluetooth Transmitter and Receiver Boards.....	39
2. Bluetooth Transmitter and Receiver Plug-Ins.....	39
3. Wired Audio Transmission.....	41
4. Bluetooth vs Wi-Fi vs Wired Audio Transmission....	42
8. Analog-to-Digital Converter.....	42
1. Dual Slope Architecture.....	42
2. Flash Architecture.....	43
3. Pipeline Architecture.....	44
4. Sigma-Delta Architecture.....	45
5. Successive-Approximation Architecture.....	46
6. Architecture Comparisons.....	47
9. Comparable Products.....	47
1. Lumi Keys.....	48
2. Razer Synapse.....	48
10. Software Technologies.....	50
1. Programming Languages.....	51
1. C++.....	51
2. Python.....	51
3. Scratch.....	52
4. C++ vs Scratch vs Python.....	52
11. Required Mathematics.....	53
1. Fourier Series.....	53
2. Fourier Transform.....	54
3. Discrete Fourier Transform.....	54
4. Mathematical Process Selection.....	55
12. Instruments.....	56
1. Piano.....	56
2. Transparent Instruments.....	57
3. Trumpet.....	58
4. Drums.....	58
13. Part Selection.....	59
1. Microphone.....	59
2. LED Strip.....	60
3. Power Supply.....	61
4. Single Board Computer.....	61
5. Display Screen.....	61
6. Audio Transmission.....	62
7. Analog-to-Digital Converter.....	62
8. Programming Language.....	63
9. Instruments.....	63

10. Box Design.....	64
11. Preamplifier.....	64
4. Design Constraints & Standards.....	68
1. Constraints.....	68
2. Standards.....	69
1. USB Type-A.....	69
2. TRS.....	70
3. XLR.....	70
4. USB Type-C.....	71
5. DSI Display Connector.....	72
6. Micro SD.....	73
7. 40 pin GPIO.....	73
8. Python .....	74
9. LCD.....	75
10. 5V USA power supply.....	76
5. Testing.....	78
1. Microphones.....	78
2. LED Strips.....	80
3. Power Supplies.....	83
4. Single Board Computers.....	84
5. Display Technologies.....	85
6. Printed Circuit Boards.....	85
7. Instruments.....	87
8. Vibration Testing.....	92
9. Thermal Testing.....	94
10. Electromagnetic Testing.....	96
11. Material Strength Testing.....	98
12. Breadboard Testing.....	100
13. Light Placement Testing.....	101
6. Prototyping.....	103
1. First Full Prototype.....	103
7. Design.....	104
1. Software.....	104
1. LED Strips.....	104
2. Microphones.....	104
3. Microphone and Light Cooperation.....	105
2. Hardware.....	106
1. Sigma-Delta ADC.....	107
2. DC-DC Converter.....	112
3. Pre-amp.....	
3. Overall Schematic.....	112



4. User Interface.....	114
5. Screen and Raspberry Pi Enclosure.....	117
8. Implementation.....	119
1. Software.....	119
2. Hardware.....	119
3. Designs.....	119
9. Administration.....	121
1. Budget.....	121
2. Project Milestones.....	122
10. Conclusion.....	125
11. Appendix.....	126
1. Bibliography.....	126
2. List of Tables.....	131
3. List of Figures.....	133

# 1. Executive Summary

It is time for instruments to take their next step of advancements. Music recitals and orchestras will no longer be something to listen to but also something to watch. Our team, along with our sponsor Dr. Richard Leinecker, planned and created a product that will take the music that comes from various instruments and turn it into a light show that will enhance the experience of those in the audience. This will be done through frequency based instrumental lights and our target audience will be musicians, schools, and performance venues.

Our team consists of two electrical engineers who will be handling the hardware for the project and two computer engineers who will be in charge of the software aspect. Our electrical engineers are Anja Frohock and Stefan Gonzalez whose areas of interest are circuit design and electrical design respectively. Our computer engineers are Leith Rabah and Carson Warner whose areas of interest are software engineering and computer hardware engineering respectively. Although, as university students, we all have limited experience in our fields, we all have the necessary skills and talents to complete this project and produce a satisfactory product.

One of the main problems and musicians face these days is capturing larger audiences. We are not referring to well known singers who put on concerts but to instrumental recitals and orchestras. Although the popularity of classical performances such as Hamilton is on the rise, many of the younger generations today see instrumental performances as boring or for the old and wealthy. We plan to change that by creating a light show that is controlled solely by the instruments being played. Every note will result in a different color and brightness being displayed on the instruments themselves. These light shows will evoke a sense of awe from the audience and attract new people to these shows. No two songs will result in the same light show which will keep the audience coming back to see the infinite possibilities. If the product proves successful, it will bring a massive return on investment for the customers as their audiences will not only increase in size, but also increase in individual retention and repetition. The product itself can be created and sold for less than the price of a single instrument which will be appealing to the consumers.

Our product currently has no competition on the market therefore, there will not be a large need for advertising. After successful testing, we plan to display our product to the music department at the University of Central Florida. If accepted, the University will begin to use our product for their performances and videos will begin to circulate online as it will be something the audience had never seen before; resulting in free advertisement for our product.

For our initial research and testing phases, we will be funded by our sponsor, Dr. Richard Leinecker. Although he has not given us a specific budget, he has given us a range of around \$1,000. The project's research phase is expected to be completed before December 6th, 2022. The testing phase is projected to be completed by February 2023 and the product is expected to be completed in May 2023.

## **2. Project Description**

Music is made up of various vibrations. Faster vibrations result in a higher tone of sound and slower vibrations create lower tones. For example, the A note on a guitar is a result of the guitar string vibrating at a rate of 440 vibrations each second. We plan to utilize these vibrations by reading them into a Raspberry Pi using a microphone and converting them into wavelengths. We will then set the wavelengths to a specific color and brightness which will be output from the Raspberry Pi and into the LED strips attached to the instruments. This will result in a colorful display of lights that are continuously changing color and brightness along with the music being emitted from the instruments.

### **2.1 Motivation**

Throughout history, many things have seen multiple technological advancements. We went from getting all our news from papers, to carrying computers in our pockets. Watching plays to watching television. Traveling on horseback to flying across the globe. However, one thing that has not advanced much in recent generations is instruments. Orchestras have remained largely the same for the past few centuries. This project aims to make musical instruments and shows more captivating by adding a colorful light feature that adapts and changes along with the music. Ranging from a bright high note to a darker low note, a musical display will now have an aurora of colors to visual aid the performance and captivate the audience.

### **2.2 Goal**

Our desired goal for this project is to be able to visualize sound emitted from an instrument with LEDs. The device must be able to analyze frequencies from an instrument through a recording device i.e., a microphone. The signal from the microphone must then be processed through an ADC (Analog to Digital Converter) so that the information can be processed by the microcontroller. The frequency input into the microcontroller will be compared with a lookup table of all frequencies and will correspond to a color scheme that will be directly output onto an LED strip attached to the instrument. There will be multiple different color scheme configurations that can be selected to output different moods or genres to match the music being played through the instrument i.e. Jazz, Rock, Funk, Classical.

### **2.3 Project Narrative Description**

What we desire from the project is an interactive system that turns on LEDs at the notion of an instrument playing, each instrument will generate its own distinct response based on the notes/keys played. The current instruments that will be desired are the flute, piano, acoustic guitar, and drum set. Each instrument will have its own distinct logistical challenges in which the input and output will vary. This product was

desired by our sponsor Dr. Leinecker for the need for interactive instruments with current day technology; furthermore, his main request is just for the flute instrument to be designed and made. The complexity of the project will increase with the involvement of more instruments, the portability of the interactive system, and the intensity of the LEDs responses based on the volume and pitch. The Sponsor wants the system to display light intensities based on the individual instrument's wave response when played at different frequencies. The project is estimated to be relatively low cost, portable, low power, and easy to use. Dr. Leinecker agreed to be flexible on the design and output of the product. On the market there is no public version of the product, hence the need for it. The use of Raspberry Pi, specifically Pi4, will be treated as the brain of the project based on the recommendation of the Sponsor. The project will have the potential to raise the quality of performances and experience of playing the instruments. Having the responsive lights to only be turned on or on a certain color when played correctly, acting as a visual aid for both the user and the observers. This visual aid can be beneficial to those that lack the skills to interpret perfect pitch or even to those that lack the ability to hear. This aspect of the design could also gain the attention of other sponsors but as of now our main focus is to complete the desires of Dr. Leinecker. Our group took interest in this project for its applications of embedded systems, coding, circuit designing, wave frequencies, and fabrication. In which this project will involve the use of existing hardware, matching devices together, manual construction and designing of devices, and possibly 3D-Printing for parts. In hopes of keeping the project relatively low-cost the project will be done with the integration of the UCF Innovation Lab, for their equipment mainly. The materials and components will be sponsored by Dr. Leinecker, though the budget limit has not been discussed in depth yet. Currently the group will be consulting with Dr. Leinecker once a week or every 2 weeks depending on his availability. The meetings with him will involve making the product for his desired design and seeing how flexible he can be as problems will be found when actually fabricating the project.

## **2.4 Requirements & Specifications**

The requested constraints and specifications of the project by the consumer include an LCD touch screen that would be able to select themes and alternate forms of power such as an attached battery pack/power supply power efficiency will be required to enable portability for extended periods of time. This would also have to include a Bluetooth module to control the LEDs. Another would be an LED light strip or laser projected light to illuminate the desired instrument which reacts to tone, pitch, and volume from the desired instrument. This has to be done via a microphone attached near the bell of the instrument, a floor standing microphone, or a direct connection through line level if the instrument is electronic and provides its own sound to avoid microphone interference. With a floor standing shotgun microphone, we would have to attach a line from it to the musician to power and control the LEDs. However, that would display more of a challenge when trying to pick up the sound of only the

instrument with the LEDs. The project and all software involved must run off of a Raspberry Pi 4 or an Nvidia Jetson and use a USB standard interface to be able to connect with a computer. The device must also be within the specified build dimensions, reason being to avoid being bulky and hard to handle. Also the device needs to be able to have a battery life of at least sixty minutes. While also being able to start up within a minute of giving the device 5VDC of power. Once on the device will have to be sending power to the LEDs, LCD screen, and Analog-to-Digital Converter. Where the LEDs have to reach a minimum brightness of two hundred lumens, and will have to vary in color and contrast based on the feedback from the microcontroller. If there is not enough power the device will need to enter an idle mode to avoid unnecessary power consumption and improve the device's energy efficiency. Meaning that once the device is not being used for at least five minutes the device should enter in the low-power mode. The table below shows more of the required specifications for the device to be operational and of standard for the demonstration. The specifications marked in green on the table will be the criteria for grading in the demonstration for the demo aspect of senior design.

Requirements	Units
The system will have fully customizable colors attached to each musical note.	12 customizable colors
The system shall have a minimum SNR value.	$\geq 25\text{dB}$
The Raspberry system shall be portable within the specified dimensions and weight	$\leq 7 \times 5 \times 5$ inches $\leq 3$ lbs
The system shall update and organize incoming signals from the microphone source multiple iterations per second	$\geq 5$ iterations per second
The system shall process incoming signals and send feedback from the microphone source within a time range	$\leq 100\text{mS}$
The system shall be active within a specified time	$\geq 1$ Minute
The system will enter Low-Power Mode within a time constraint.	$> 5$ minutes
The system shall be able to emit light at a high brightness	$\geq 200$ Lumens
The total cost of the system will not exceed over \$1000	$\leq 1000$ \$
The LCD screen will not exceed over these dimensions	$\leq 5 \times 4 \times 0.5$ inches
The system will display on the LCD screen the mode it is in	$\leq 4$ Modes
The Microphone system's dimensions	
The power supply of the Raspberry controller and microphone will not exceed over 5VDC.	$\leq 5$ VDC
The microcontroller must have a form to manually turn on and off the system.	$\leq 1$ Power Switch
The microcontroller must be compatible with a secondary battery supply that is capable of 5VDC.	$\geq 2$ Power Supplies

Table 1: Requirements and specifications

Although we had many options to choose from for the three requirements to present, we chose the three highlighted in blue for the following reasons:

- *The system shall update and organize incoming signals from the microphone source multiple iterations per second.*

This is the main feature of our project. Without it, there will be nothing to present. This is what causes the lights on the instruments to change color and brightness according to the sounds being played into the microphone. If this requirement is not met, the entire project fails.

- *The system will have fully customizable colors attached to each musical note.*

This will give the user the ability to customize the lights displayed by each instrument's music to their liking. It will also increase the variety and give the audience a new visual performance despite hearing the same music.

- *The system shall have a minimum SNR value of 25dB*

The goal of this requirement is to filter out as much background noise as possible. If we succeed in filtering out the noise coming from other instruments, we will give our product the ability to be used in bands and orchestras. This will create a dynamic light show where each instrument's lights can display a different color and brightness without interference from other instruments.

## **2.5 Block Diagrams**

In this section we can see the different block diagrams for Hardware and Software that help illustrate the inner workings of how the hardware components work with each other, and how the blocks of software interact with each other. This section also includes the House of Quality as well as our first illustrative prototype draft of what the project will look like when it is fully assembled. These block diagrams serve as an intuitive way for the consumer of this report to easily understand the inner workings of the multiple parts of the project and the project as a whole from a quick glance.

### **2.5.1 Hardware Block Diagram**

This diagram breaks down all of the hardware responsibilities for each hardware team member and labels each phase of the project for each individual member of the group. The block diagram shows the relationship between the hardware parts of the project and which parts are directly dependent on each other.



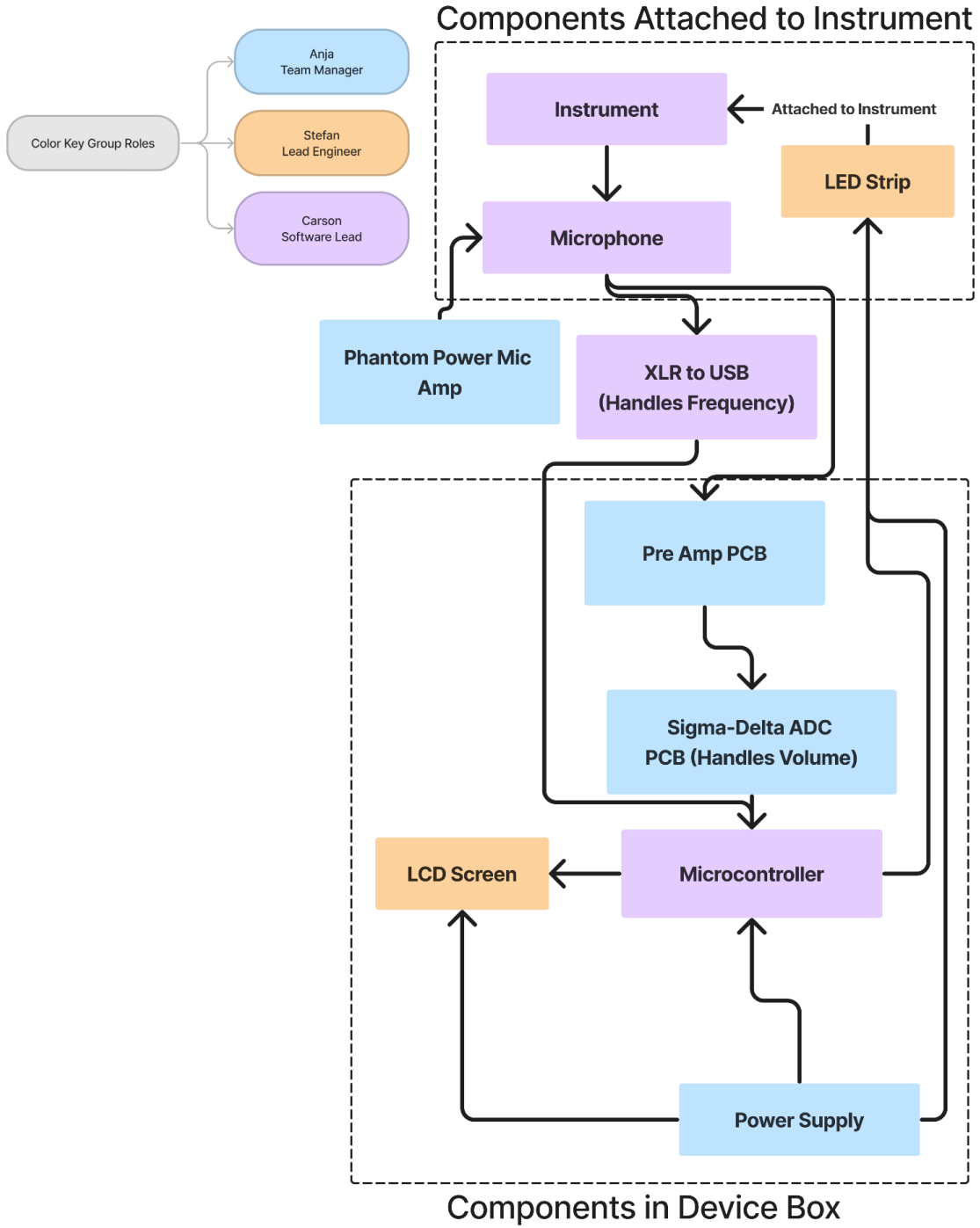


Fig. 1: Hardware Block Diagram

## 2.5.2 Software Block Diagram

This diagram breaks down all of the software responsibilities for each software team member and labels each phase of the project for each individual member of the group. The block diagram shows the relationship between the software parts of the project and which blocks of the program are directly dependent on each other and how they work if one condition is met or is not met with the subsequent actions of the program based on the conditions.

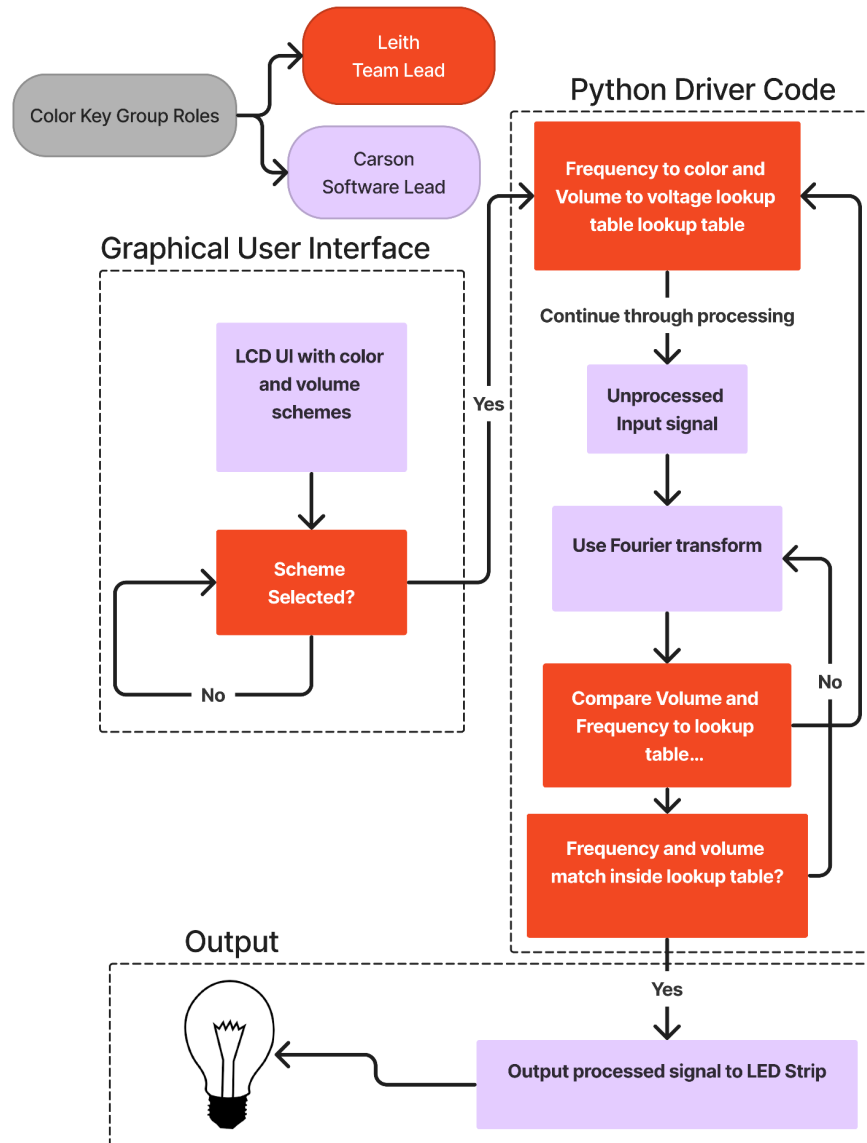


Fig. 2: Software Diagram

### 2.5.3 House Of Quality

The House Of Quality below represents the expected engineering requirements and customer requirements of our project. It displays how our team intends to move forward with the engineering requirements of the project by showing what we intend to maximize, minimize, and target. It also displays the correlation between the engineering requirements which lets us know what we will need to decide to compromise on and what to give more focus to. Lastly, it represents how the customer requirements will affect the engineering requirements and how we move forward with the project while taking into consideration what we want to create versus what the customers want to use.

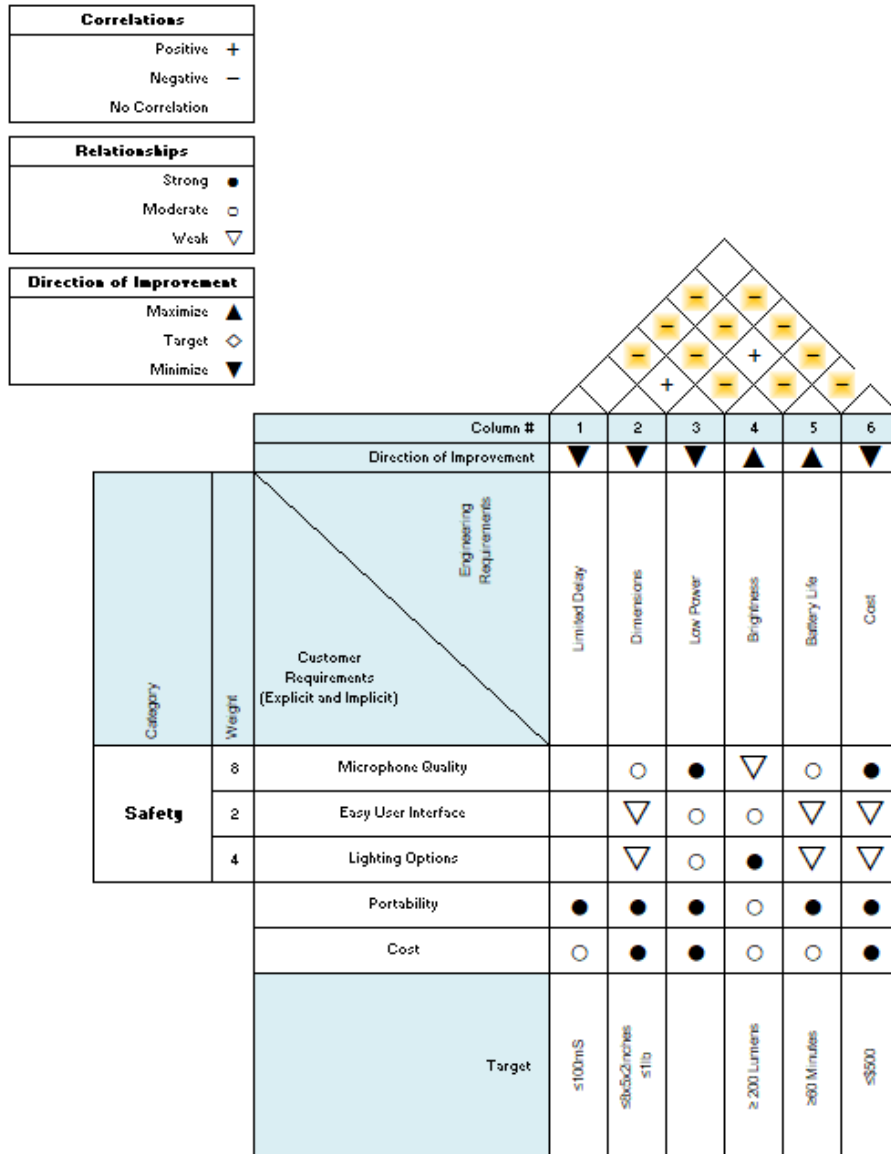


Fig 3. House Of Quality

### 2.5.4 Illustrative Prototype

The following image is an illustrative prototype of how we originally planned to create our project and the design of our first prototype.

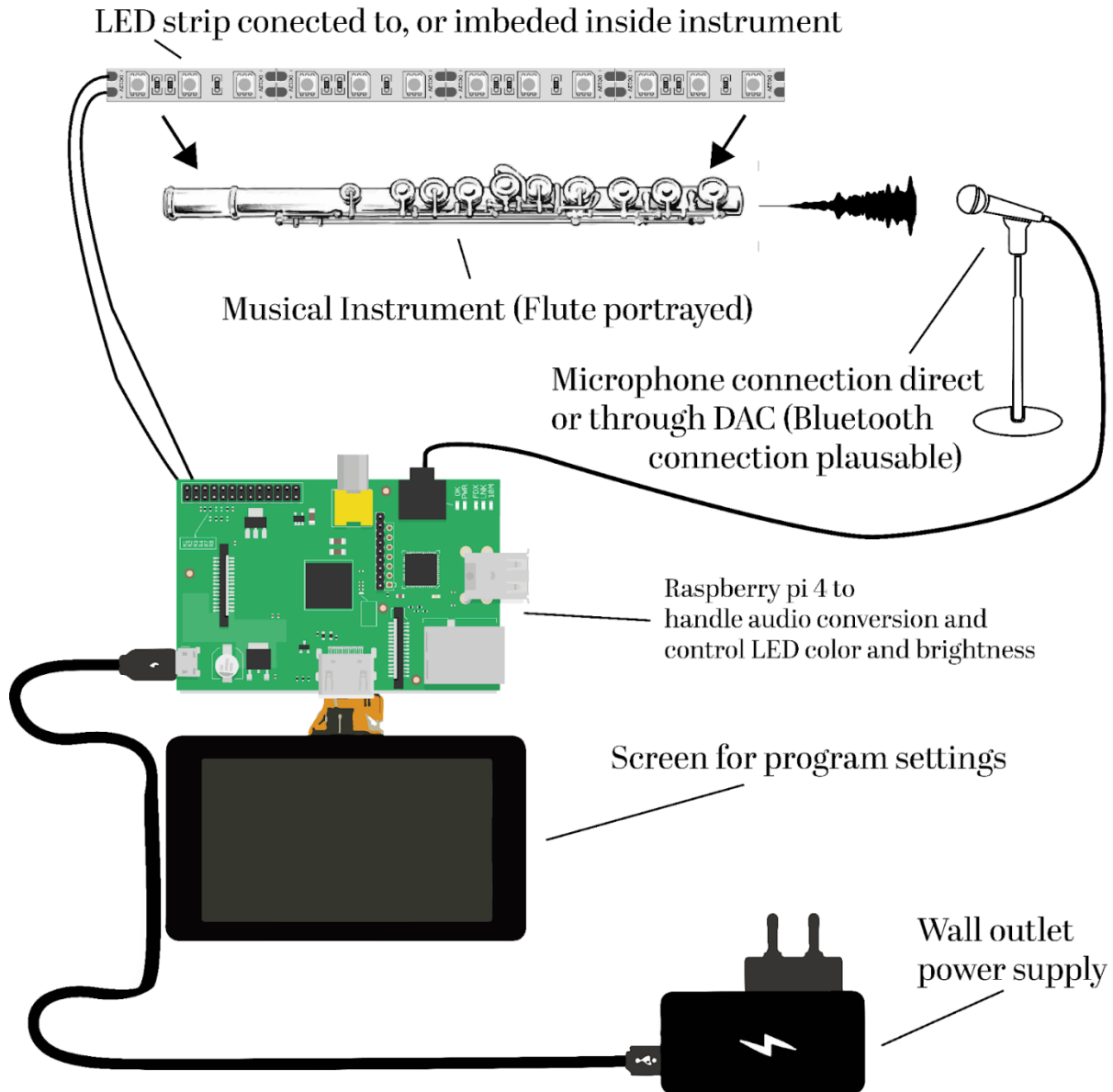


Fig. 4: Prototype

### 3. Research & Technology Comparison

Our first task was research. Once we determined the parts we were going to need in order to build the product, we split them amongst ourselves and assigned each member an equal number of parts to research and report back to the group with. Once each member completed their research, a discussion was held to determine which specific parts we would like to use for the testing phase of the project.

#### 3.1 Microphones

This project requires a microphone in order to read in the sounds emanating from the instruments and convert it to wavelengths. This can be done with a variety of different types of microphones. Upon researching the possible microphones, we came upon three possible types of microphones we can use that would return different results. Shotgun microphone, Contact microphone, and Clip-On Condensers.

##### 3.1.1 History of Microphones

The first modern microphone was reportedly invented in 1877 by Emile Berliner and Thomas Edison, and was also invented by David Edward Huges at the same time in England. A long dispute followed both patent requests. However, Emile Berliner and Thomas Edison were awarded the patent with their Carbon-Button microphone which utilized a drum like device which was used to help amplify input audio with the resonance of the drum. Carbon-Button microphones are comprised of carbon granulates whose relative resistance decreases when they are compressed together, the carbon granules are contained between a thin metal diaphragm and a fixed electrode which helps compress and decompress the carbon granules when sound waves strike the diaphragm. When a current is flowing through the assembly the output resistance can be measured and interpreted as sound.

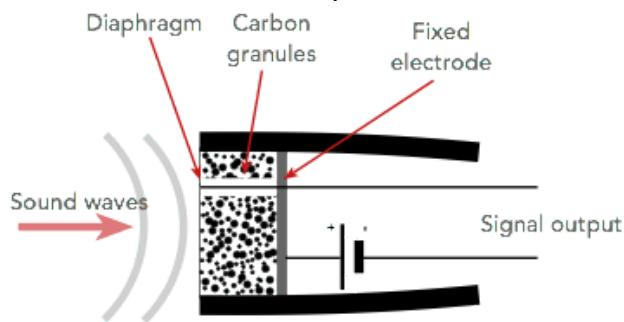


Fig 5. Carbon Microphone Schematic

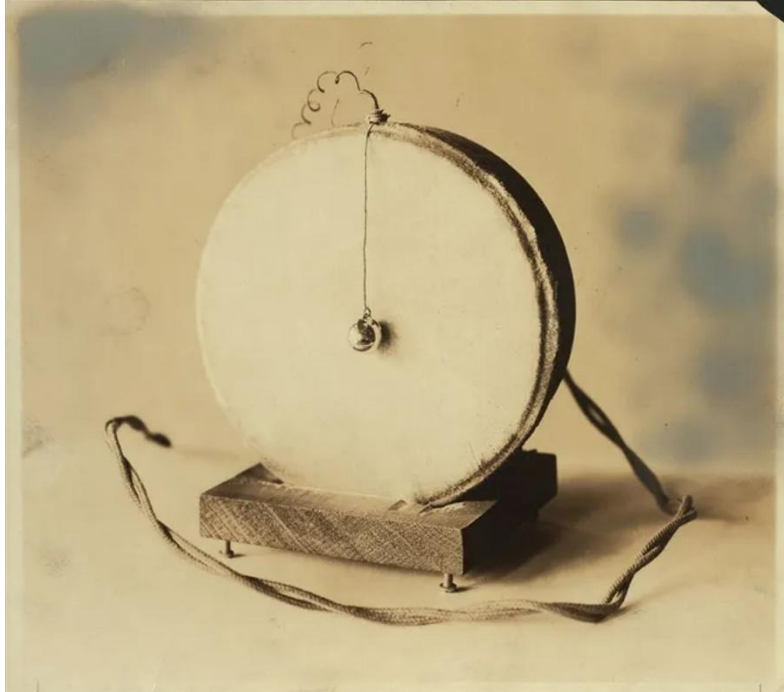


Fig 6. Carbon-Button Microphone Edison/Emile

After the invention of the Carbon Microphone followed the Condenser microphone, the Condenser Microphone was invented and patented in 1916 by E.C. Wenthe was originally used as a tool to measure sound intensity inside a lab and not to record audio for consumption. Later in the 1920's Condenser microphones started to be produced for the use of broadcast and recording, mainly radio broadcasting and music recording. Early Condenser microphones required an external piece of hardware to ensure functionality due to its impedance conversion requirement and the only practical solution to this was to use a thermionic valve. Condenser microphones, however, were not perfect at this point in time, reports from the BBC in 1926 claimed that Condenser microphones were susceptible to moisture inside of the assembly which ultimately ruined sound quality and caused background noise in recordings that members from the BBC claimed to be 'frying noises'. After the Condenser microphone came the Electromagnetic microphone in the 1920's, the Electromagnetic microphone consisted of a coil, iron, and ribbons that moved with aid of a magnet to capture audio. Due to the fact that magnets at the time were relatively weak the Electromagnetic microphones required the use of external power supplies to assist in the movement of the magnetic coil, as a result of this the first Electromagnetic microphones were much larger than the previously mentioned Condenser microphones and required massive amounts of power to be operational. Following the invention of the Electromagnetic microphone came the more modernized RF Condenser microphone which improved upon the original Condenser microphones' weaknesses to humidity and increased the microphone's sound quality. To make such an improvement GFHull and JJ. Zaalberg van Zelst in 1946 and 1947 respectively, replaced the outdated and bulky valve

impedance converters with more robust and smaller transistor circuits which helped eliminate humidity and sound quality issues associated with the first Condenser microphones. Since 1947 we can see the widespread use of the Condenser microphone by large microphone manufacturers such as Sennheiser, Shure, and Audio Technica and have seen widespread use in audio capture in the modern era.

### 3.1.2 Shotgun Microphone

A Shotgun microphone would ideally be used on a floor stand to capture sound from the desired instrument from a far distance. Shotgun microphones are commonly used in video and audio recording/production, some of the main advantages of Shotgun microphones are its polar pickup pattern which is extremely focused in a small degree field which in short means that the microphone can only pick up sounds emitted in that very focused region. The polar pattern of a shotgun microphone is called the lobar polar pattern, polar patterns show the area in a 360-degree field of view where the microphone can pick up sound and at what decibel gain the microphone can pick up that sound at. Due to this narrow pickup field shotgun microphones can be used to clearly pick up audio from a subject who is in an environment with high noise interference (i.e., a crowded mall) since the interfering audio is outside the lobar polar pattern.

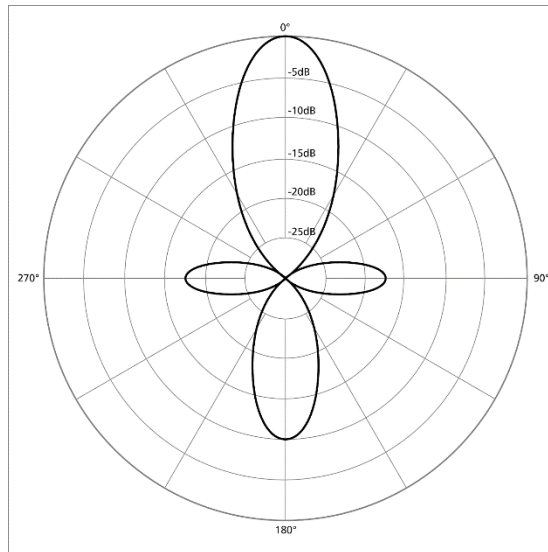


Fig. 7 Lobar/Shotgun Polar Pattern

In the potential use case for this project a Shotgun microphone would be ideal for accurately picking up audio from an instrument from a far distance, some advantages of a Shotgun microphone over a regular Cardioid microphone are its pickup pattern which allows the Shotgun microphone to pick up sounds that are only in the area that it is pointing in. Cardioid microphones will pick up sounds in roughly a 180-degree area from the center of the microphone, this could pose problems for pitch

recognition if the desired instrument to be visualized is in an environment with other instruments or loud background noise that could interfere with audio recognition. Some downsides that the Shotgun microphone poses is that while its focused polar pattern is a great asset for filtering out unwanted noise it is also a handicap if the desired subject were to move from their original position while playing, this is not an issue with larger instruments such as an upright bass, or a drum set, but most brass and woodwind instruments have the ability to be mobile and most performers generally do not stay in one place while they are performing unless in a concert or symphony situation. In an ideal situation the Shotgun microphone would be best suited to an environment where performers have designated areas or larger instruments that are not mobile. Another shortcoming of Shotgun microphones are their size and portability, generally Shotgun microphones are mounted to a floor stand which keeps the microphone in place, if we want the entire setup of the device to be somewhat portable then a floorstanding Shotgun microphone may hinder the ability to setup and teardown the device quickly and efficiently. This portability hindrance may not be an issue however if the device were to be used in a concert setting where there is plenty of time for set up and tear down that can be utilized.



Fig 8. Vidpro Xm-55 Shotgun Microphone

### **3.1.3 Contact Microphone**

A Contact microphone would be used in a portable scenario and can be directly attached to the instrument being recorded. The main advantages of utilizing a Contact microphone are that due to its intimate location in relation to the recording subject it can clearly pick up sound from the subject with virtually no noise interference. Contact microphones are in a relatively niche segment in terms of microphones and one of the most commercially used instances of Contact microphones are in clip on guitar tuners.



These tuners record sound directly from the head of the guitar and can distinguish the pitch of each string down to single Hz accuracy. In the use case of our project, we can utilize this frequency accuracy to help distinguish individual pitches for each note played by the desired instrument. This accuracy combined with the low level of noise interference makes the Contact microphone a prime candidate for the application that this project requires.



Fig 9. Korg Clip-On Contact Microphone

Contact microphones are not often seen in use in real world applications for musical instrument recordings due to the fact that they cannot capture high fidelity audio with the configuration of how they operate. Contact microphones can only pick up sounds that are emitted by direct vibrations meaning that sounds such as fingering on a guitar or high-pitched tones do not emerge as well in recordings as they would on a regular floor standing Cardioid microphone. The Contact microphone's intimate location also can add unwanted noise to a recording if the microphone is placed too close to an area of an instrument that moves. If a Contact microphone were to be placed upon a Saxophone near its valves it would pick up the sound from the instrument as well as the metallic clicking of each of the valves on the instrument; this could cause trouble in the majority of woodwind instruments if the performer plays with high intensity and vigor, however the interference does not pose a deficit upon the majority of brass instruments of those which have considerably less valves and moving parts that could cause noise interference.

### **3.1.4 Clip-On Condensers**

A Clip-On Condenser microphone can be used in the same portable scenario as the Contact microphone however, it improves upon high fidelity recording in relation to Contact microphones due to its traditional condenser design in combination with an

intimate relation to the recording subject. Clip-On microphones are widely used in the music industry for high quality recording of wind instruments in concert settings, their compact size and uncompromised sound quality lend them the most popular choice in instrument recording in the industry however all of these aforementioned advantages come at an abundant expense along with the fact that to be fully operational condenser microphones require the use of a pre-amplifier to amplify the microphones input volume to a level where the audio can be recorded effectively and clearly interpreted, this can add additional weight to the whole assembly however, most clip-on condenser microphones include a pre-amp that can clip on to the performers waist for improved portability.



Fig 10. Audio-Technica ATM350UcW Clip-On Condenser Microphone

Clip-On microphones can be seen in use as far back as 1983 when the audio giant company Sennheiser developed the first Clip-On microphone (The Sennheiser MKE 40) which originally was intended for lapel use for radio and tv show hosts as an alternate for boom microphones which improved background noise isolation due to its close proximity to the subject that it was recording. Since the invention of the Sennheiser MKE 40 small Clip-On microphones have seen a large increase in popularity for subject recording. The lapel or lavalier clip-on microphone can be seen in use now for speech recording for TV, Radio, and Movies due to their compact size and ability to clip on to the subject's shirt for discrete audio capture. The gooseneck clip-on microphone has seen widespread use in audio capture for musical instruments due to its mounting versatility and compact size. Nearly every musician or musical group can be seen using gooseneck clip-on microphones for audio capture for wind instruments such as trumpets, saxophones, and baritones and gooseneck clip-on microphones can

also be seen in use for percussive instruments such as snare drums and tom drums. The Clip-On microphones unparalleled sound quality combined with its small dimensions and versatility in mounting lends it to be one of the most widespread microphones used today for instrumental audio capture however, all these points come at a higher cost than most regular floor standing microphones and in a budget minded scenario may render the Clip-On microphone as an unlikely candidate for audio capture over more budget friendly microphones such as the Contact microphone.

### 3.1.5 Shotgun vs. Contact vs. Clip-On

Comparison	Shotgun	Contact	Clip-On
Pickup pattern	Lobar	Direct	Cardioid
Cost	Higher (\$90)	Lower (\$20)	Higher (\$150-180)
Size	Larger	Smaller	Smaller
Sound Quality	High Fidelity	Standard	High Fidelity
Application	Floor Standing	Direct Mount	Direct Mount

Table 2: Microphone Comparison

## 3.2 LED Strips

The initial idea was to attach a strip of RGB LEDs to the instrument which we would use to create the light effects similar to that of light up computer keyboards. We began by researching the lights used by a major company who uses WS2812B light strips which come in various sizes and densities. The first concern was brightness. We needed a light strip that would be bright enough to be visible to an audience, however not too bright as to negatively impact the musician's vision and performance. Possibly similar to that of the flashlight of a mobile phone which has a brightness of approximately 111.5 lumens or 1200 nits. We would also require lights with a high intensity in order to clearly display the various colors that the music would create in order to better captivate an audience. Upon further research, we came up with several options of LED strips to use for the instruments:

### 3.2.1 Addressable LED Strips

These are LED strips in which we have the capability of controlling each and every light on the strip individually, giving them their own colors and brightness without affecting the entire strip. This would lead to more various and unique ways to create customized preset lighting however it does not necessarily align with our goal of having the colors and brightness adjust based on the music being played.

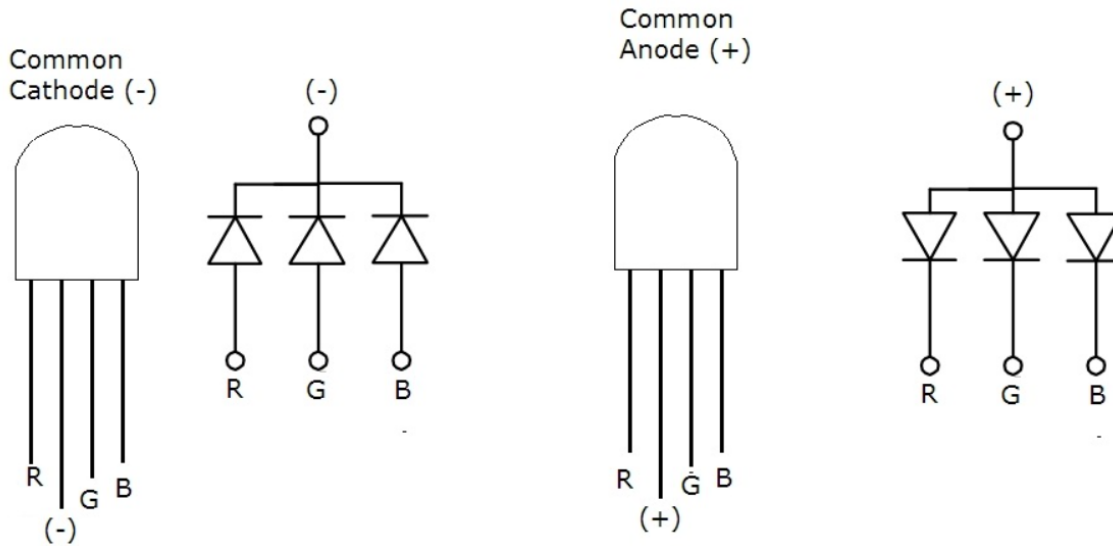


Fig 11. RGB LED Schematic

There are two types of RGB LEDs. Common cathode LED and common anode LEDs. With common cathode LEDs, each of the RGB LEDs share a negative connection and in a common anode LED, they share a positive connection. Regardless of whether it uses a common cathode or common anode, each RGB LED has four pins. One for each of the red, green, and blue LEDs, and one for the common cathode or anode.

### 3.2.2 SMD LED Strips

The first type of LED strip is SMD (Surface Mounted Diodes) strips. The LED chips in these strips are fused to a PCB (Printed Circuit Board) that is built on a flat rectangular object and uses 3 diodes. The most common SMD strips are the 3528 and 5050 LED strips. These types of lights are most commonly used on displays and signs. The strips consist of a collection of LED chips all connected to each other. The following image is of a single chip that is attached to the LED strip.

### WS2812B LED CHIP



RGB

Fig 12. SMD LED Chip

### 3.2.3 COB LED Strips

COB (Chip On Board) strips use 9 or more diodes fixed together to create a single light source. These are LED strips which have a much higher density of light units which gives the lights a smoother and more concise look by removing the dark areas between the LEDs that would appear in a strip with a lower density of light units such as with SMD strips. Some COB strips also have a ceramic material on the substrate which helps maintain the temperature of the lights. The following is an image of a COB LED chip that is what the strip consists of with typical measurements.

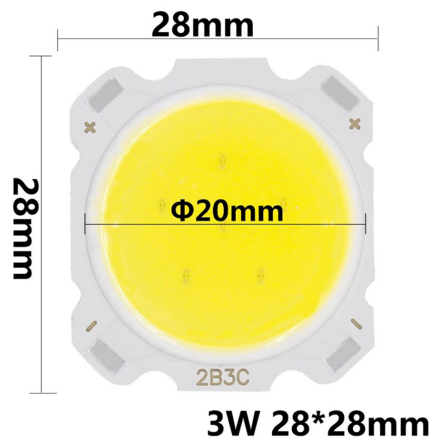


Fig 13. COB LED Chip

### 3.2.4 SMD vs COB

The following table represents the features taken into account when comparing SMD and COB light strips.

Comparison	SMD	COB
Quality of Light	-Features a glare -Best used as a point light	-No glare -Uniform light beam -Best used as a surface light
Production Efficiency	Lower	Higher
Cost	Higher (~5%)	Lower (~5%)
Color & Temperature	Adjustable with a wide variety of colors	Non-Adjustable color
Application	Signs & Displays	Security & Industrial
Energy Efficiency	Highly Efficient	Highly Efficient
Brightness	50-100 lumens/watt	80+ lumens/watt

Table 3: LED Comparisons



Fig 13. COB vs SMD LEDs

### 3.3 Power Supply

The project will have multiple devices all connected and powered through the Raspberry Pi. There were two options when discussing how to power the Raspberry Pi. The first was a battery to add portability to the system, the second was a mini-USB wall charger. There were 4 different battery options considered when making this decision, all of which had pros and cons to them that determined their best use case scenario. After weighing out these pros and cons we were able to come to a conclusion on the best power supply to utilize.

#### 3.3.1 Rechargeable Lithium-Ion Battery Pack

The first option was a rechargeable lithium-ion battery pack with an AC/DC converter built in. The built in AC/DC converter allows the battery to output multiple different voltages including 5V for the Raspberry Pi. This would add considerable size and weight to the design as it is 1.64 pounds. That amount of weight would severely limit the portability especially as the user is playing an instrument. This also would increase the cost substantially. However, it has the longest battery life of the three options at 8300mAh.



Fig14. Rechargeable Lithium Ion Battery Pack

#### 3.3.2 Lithium Polymer Battery with Powerboost

The second option would be a 3.7V lithium polymer battery with a powerboost module. The powerboost module would act as the DC/DC converter as well as preventing overcharging. This works as the power boost module would increase the 3.7V battery to 5V which is what is required to operate the Raspberry Pi. Buying the two parts separately rather than the one is lighter; however, the price is only slightly less. There would be added complications due to the charger module being an exposed board to prevent damage to the board, a case for the battery system would be necessary. The easy ways to do that would either be to buy a case or 3d print one.



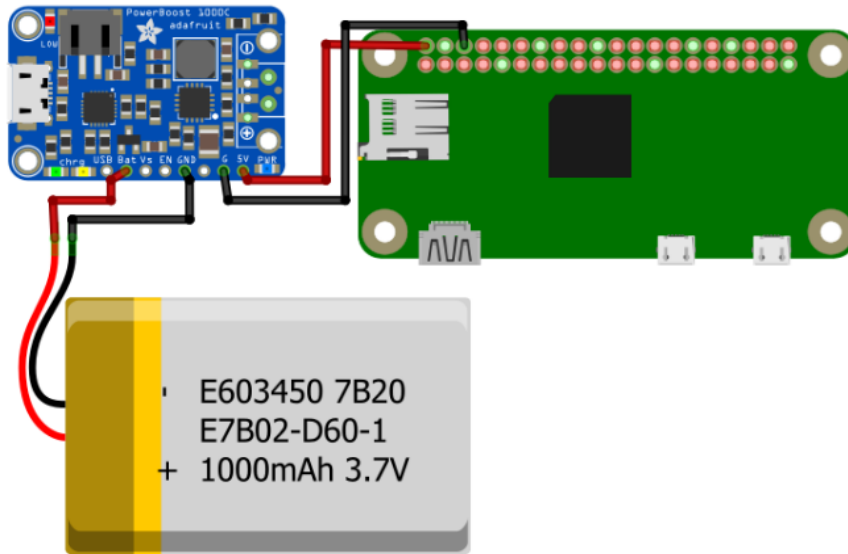


Fig 15. 3.7V Lithium Polymer Battery Pack with Powerboost Configuration

### 3.3.3 Raspberry Pi Battery Pack

The third option would be a specific Raspberry Pi battery pack. This would be the simplest as it has a case and would have a built-in cord to plug directly into the Raspberry Pi. This is the cheapest battery option. It is also relatively light therefore this would be the best option for the battery pack.



Fig 16. Raspberry Pi Battery Pack

### 3.3.4 Raspberry Pi Battery Hat

The fourth option would be a hat for the Raspberry Pi that acts as a battery as well as the ability to have an additional battery for longer battery life if necessary. This would have very little added bulk or weight as it is just added on top of the Raspberry



Pi. It also has the ability to have the system shutdown in case of low power preventing sudden shutdowns which may result in data loss. However, it would add a considerable amount of cost to the system and would most likely require an additional battery to be purchased as it has only 1820mAh.

Share



Fig 17. Raspberry Pi Battery Hat

### 3.3.5 USB Wall Charger

The other option was the mini-USB wall charger which would make the system stationary. This is greatly cheaper than the battery options as it would only require a cord. This is also the easiest option however this forces the system to either use a mic stand, require a tether between the instrument and the Raspberry Pi, or transmit the audio from the mic via Bluetooth. The Bluetooth option would add an exceeding amount of complexity to the project. Therefore, the system being stationary limits the uses for it.



Fig 18. Wall Charger for Raspberry Pi

### 3.3.6 Power Supply Comparisons

Comparison	Rechargeable Lithium Ion Battery Pack	3.7 V Lithium Polymer Battery with Powerboost board	Raspberry Pi Battery Pack	PiJuice Hat	USB Wall Charger
Size (inches)	1.9x3.2x6.3	2.6x1.42x0.39 1.14x0.91x0.08	4.17x3.82x0.75	4.33x4.92x1.38	N/A
Weight (lbs)	1.56	1.25	0.28	0.22	N/A
Cost	\$53	\$38	\$26	\$90	\$12
mAh	8300mAh	3000mAh	4000mAh	1820mAh	N/A

Table 4: Power Supply Comparisons

### 3.4 DC-DC Converter

DC-DC Converters act as a safe power supply for the device, meaning that it will always be outputting the desired DC voltage to the device. It is most commonly connected in series to a battery pack to ensure there is no surplus or loss of voltage. Use EagleCad to design the DC-DC converter which is compatible with the battery pack that is going to be used. The desired output voltage is going to be 5VDC, currently only needed for the microcontroller. The advantage of DC-DC converters is that it makes the device portable, and efficient with its power. Buying the DC-DC converter should be relatively cheap and easily accessible, however if designed through EagleCad it will have to be fabricated and the price can range.

However, adding a DC-DC converter would increase the needed space of the device. Depending on the size of the converter will be the main constraint for the inclusion of it in the final build. The other constraint would be the input voltage, meaning the power supply has to fall in a range that the DC-DC converter can operate with. If not the system will have a significantly bad efficiency rate and there would be unnecessary power loss, meaning that the system would lose power faster than it should. Meaning less operational time for the device until it's forced to shut down or go into low power mode.

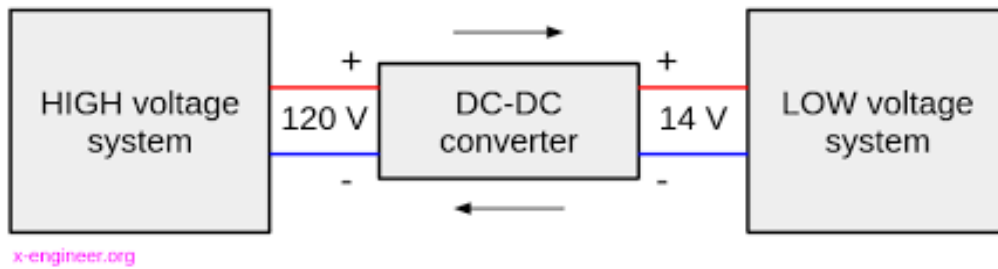


Fig 19. DC-DC Converter

The DC-DC converter that is designed already is made compatible with two double-A (AA) batteries. The batteries can either be that of Alkaline, most commonly found as the non-rechargeable variant of batteries, or Nickel-Metal Hydride (NiMH) batteries, commonly found as the rechargeable variant of batteries. The batteries would fall in the voltage range of 1.65VDC and 1.4VDC for the alkaline battery and 1.45VDC and 1.2VDC for the NiMH battery. In series these batteries would generate either 3.3VDC or 2.4VDC as the input voltage for the DC-DC converter and then set it to a 5VDC output voltage. The NiMH batteries are a good choice in having, however, they are more expensive and die much faster than the standard alkaline batteries. Though they are just another option as the power source for the design. The DC-DC converter also would be outputting a maximum of 150 milliAmps as the current.

The design for the DC-DC converter will be as shown from Fig 20. DC-DC Converter Schematic, showing the use of Integrated Circuits (ICs), capacitors, inductors, and power sources. With the use of the IC it can be set as an amplifier that increases the voltage output. The designed DC-DC converter listed price on Texas Instrument (TI) was set to \$12.89, including the fabrication of the device. Buying the components alone can also be cheaper based on how the fabrication will be done by the students. In the case of this project the best choice would be relying on Texas Instrument to provide the fabricated device.

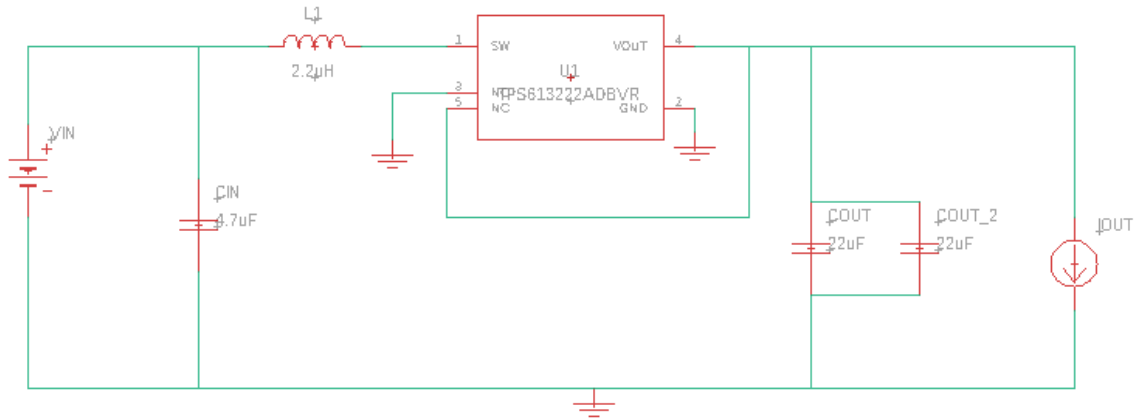


Fig 20. DC-DC Converter Schematic

### 3.4.1 Low-Power Mode

Having the microcontroller use Low-Power mode is optimal for the preservation of the device, and reduces the chance of damage due to low to zero power within the system. What low-power mode does is give the microcontroller to enter in a state where it prioritizes where power is being sent to the microcontroller. In common cases with other devices, the displays would be shut off while the device is still on to preserve power. Low-power mode has four modes within it in which each has their own relevance and can be implemented when they are needed. The microcontroller will be what switches these modes on and off depending on the power level within the device. Ideally the device should last for at least fifteen minutes after the microcontroller enters its Low-Power mode, giving the user time to find a charger to properly turn off the device to replace the battery. Keeping ACLK running as long as possible is the ideal usage for the microcontroller when entering Low-Power mode due to it being the timer that keeps the microcontroller going (usually a thirty-two Kilo-Hertz Quartz Crystal).

	CPU	MCLK	DCO	SMCLK	ACLK
<b>LPM0</b>	InActive	InActive	Available	Available	Available
<b>LPM1</b>	InActive	InActive	InActive	Available	Available
<b>LPM2</b>	InActive	InActive	InActive	InActive	Available
<b>LPM3</b>	InActive	InActive	InActive	InActive	Available
<b>LPM4</b>	InActive	InActive	InActive	InActive	InActive

SMCLK is not available in LPM1 if sourced from DCO

Fig 21. Low-Power Modes

The Low-Power mode will be directly linked to the battery supply of the microcontroller, meaning that once the total voltage of the battery reaches a lower limit of their charge it will activate the different Low-Power modes. In the case for this device the power supply will be set as 6.6VDC in which the device will be in active mode. The device will then swap from active mode to Low-Power mode once the total voltage in the power supply falls under 4.8VDC. This mode should then indicate the microcontroller to use less power like turning off the LCD screen, sensors, microphones, and any LEDs that are not being used. Also if possible for the future the microcontroller should have an LED that switches on and off to indicate to the user that the microcontroller is in Low-Power mode.

### 3.5 Single Board Computers

This project will require the use of a single board computer to have the capability to run the audio decoding and frequency addressing program with GUI capability for theme selection, provide power to the LED strip attached to desired instrument, record input from an analog microphone, and have touch screen LCD support to enable user interaction with the project. The ideal single board computer for the project should be powerful enough to handle the aforementioned responsibilities along with 5V power support onto the external devices and be affordable enough to purchase to minimize overall expenses of the device since the single board computer makes up a large portion of the project budget and is the single most expensive component that is required.

#### 3.5.1 Raspberry Pi 4, Model B

The Raspberry Pi 4, Model B is a single board small form factor all in one computer that features many I/O ports, direct display out ports, built in camera ports, and the suite of connectors that allow it to maintain the full functionality of a computer

(i.e., HDMI, USB 2.0/3.0, Ethernet (RJ45), 3.5mm audio, 5Ghz WiFi, microSD, and Bluetooth 5.0). The Raspberry Pi 4 Model B uses a Quad-core ARM processor that utilizes a 64-Bit memory bus specifically the Cortex-A72, this processor is clocked at 1.5Ghz under normal operation and is comparable in speed to the Apple A8 which was also an ARM based CPU that could be found inside of the Apple iPhone 6, iPad mini gen 4, and iPod touch gen 6. This CPU may be relatively underpowered in comparison to current gen desktop CPUs, but the speed of the CPU should be more than adequate to process digital audio in real time and apply the necessary signal outputs from the frequency lookup table. The Raspberry Pi 4 includes a Broadcom BCM2711 VC6 GPU to help deal with graphical load on the device and allows the computer to output up to 2 4K (3840x2160 pixels) displays at 30Hz simultaneously, the GPU will be an important component of the device because the project will include a GUI (Graphical User Interface) to help select modes for the lighting profiles.

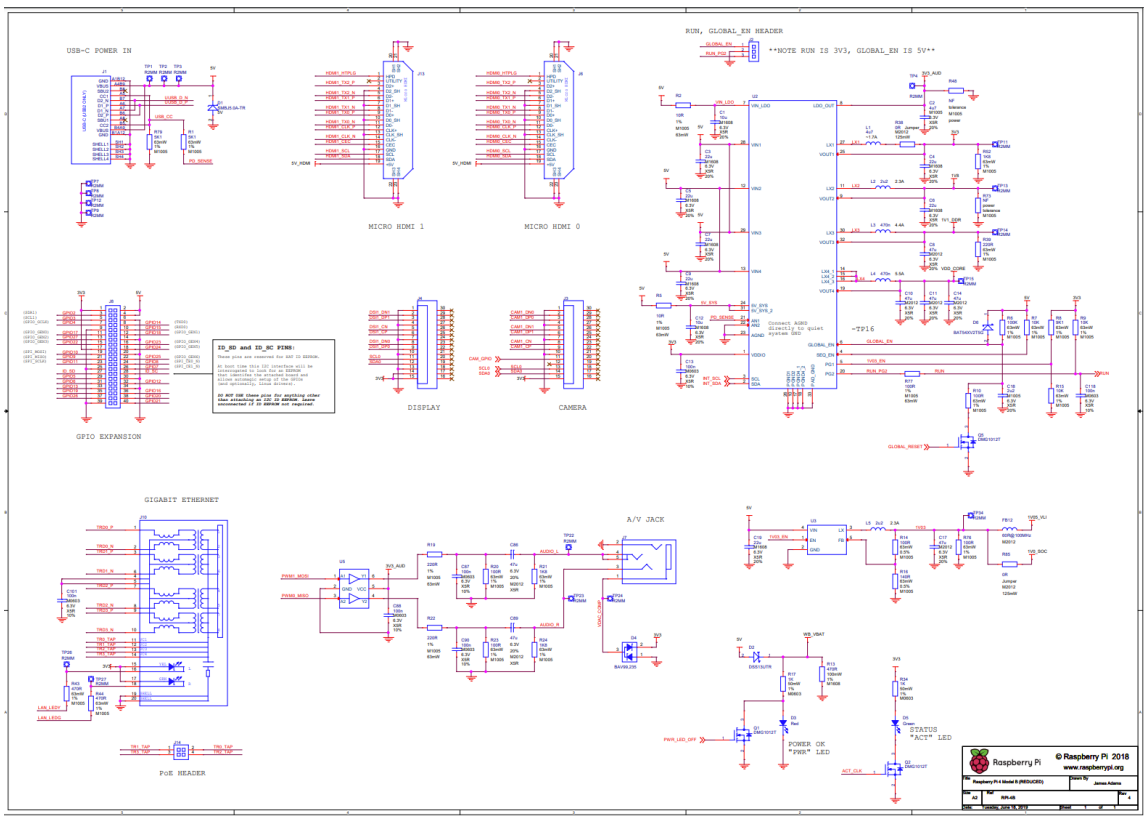


Fig 22. Raspberry Pi 4 Schematics

Having a dedicated GPU helps smooth out any graphical features that may be implemented and helps contribute to a stress-free user experience. The Raspberry Pi 4 can be found in many different configurations for the amount of onboard memory that it includes, these configurations include 2GB, 4GB, and 8GB of LPDDR-4 RAM. The amount of RAM included on the Raspberry Pi 4 does not entirely improve performance

for our specific application however, if we intended to program the Raspberry Pi 4 while utilizing multitasking and background processes then higher amounts of RAM may prove to be more effective in process speedup time. The Raspberry Pi 4 also requires the use of a microSD card for program and operating system storage, it can handle between 8-32GB of storage where 8GB is the minimum amount of storage to be able to install Raspberry Pi OS on to the Raspberry Pi, some users of the Raspberry Pi claim that the upper limit of its microSD storage support is 1TB of microSD storage space This combination of CPU, GPU, and RAM in the use case of the music visualizer should prove to be more than enough to effectively accomplish the desired GUI, audio conversion, LED power/signal, audio key/pitch lookup and visual processing required for the device. The Raspberry Pi 4 runs on Raspberry Pi OS which is a custom Unix-like operating system based on Linux that has been adapted to optimize performance for the ARM processors that are contained in the various versions of the Raspberry Pi. The Raspberry Pi 4 also has the ability to run other Linux based operating systems such as Ubuntu, Mate, and LibreElec and can even run Microsoft's Windows 10. This wide range of compatible operating systems allows the Raspberry Pi 4 to be highly configurable and adaptable to any application regardless of the required operating system. The relatively good performance of the Raspberry Pi 4 does come at a premium with starting prices being as low as \$158 (MSRP \$35)

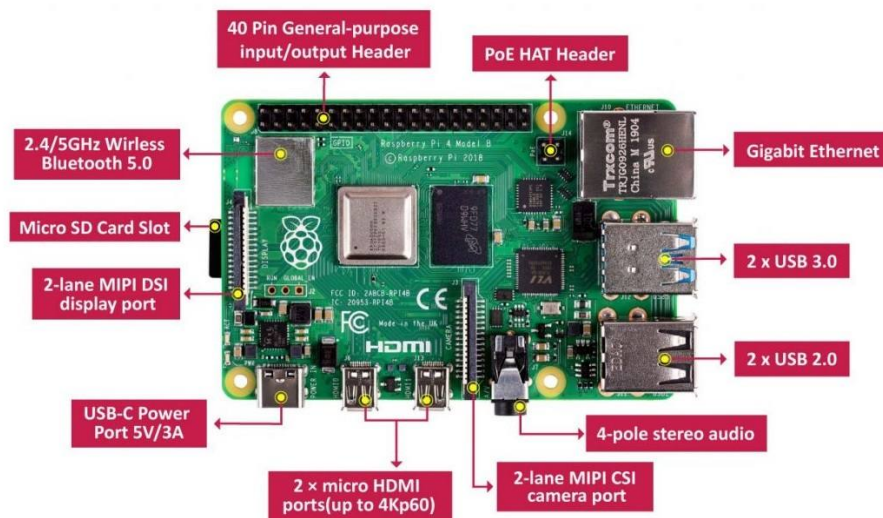


Fig 23. Raspberry Pi 4, Model B

### 3.5.2 Raspberry Pi 3, Model B+

The Raspberry Pi 3, Model B+ like the Raspberry Pi 4 is a single board small form factor all in one computer that features many I/O ports, direct display out ports, built in camera ports, and the suite of connectors that allow it to maintain the full functionality of a computer (i.e. HDMI, USB 2.0, Ethernet (RJ45), 3.5mm audio, 5Ghz Wi-Fi, microSD, and Bluetooth 4.2). The Raspberry Pi 3, Model B+ uses a Quad-core ARM processor that utilizes a 64-Bit memory bus specifically the Cortex-A52, this



processor is clocked at 1.4Ghz under normal operation and is comparable in speed to the AMD A4-1200 APU (Desktop CPU), Intel Core Duo T2400 (Laptop CPU), and Qualcomm APQ8016 (IoT application processor). In comparison to the Raspberry Pi 4s Cortex-A72 the Raspberry Pi 3, Model B+s Cortex-A53 can perform at roughly half of the speed in relation to the Raspberry Pi 4. With the large decrease in performance in relation to the Raspberry Pi 4 we can assume that the Raspberry Pi 3 is not going to be as capable for high performance applications requiring immense CPU power however, the performance of the Raspberry Pi 3's CPU may be adequate enough to process digital audio in real time and apply the necessary signal outputs from the frequency lookup table. The Raspberry Pi 3 also includes a Broadcom BCM2837B0 V8 GPU to handle graphical load on the device and is capable of outputting video at 1080p (1920 x 1080 pixels) at 60hz to a single display, the GPU of the Raspberry Pi 3 will be a useful component of the device because a GUI (Graphical User Interface) will be used for program selection and lighting profiles and it allows the user to experience smooth animations when interacting with the device via the touch screen. Unlike the Raspberry Pi 4, the Raspberry Pi 3 is only available in a singular memory configuration with 1GB of LPDDR2 SDRAM. The low amount of available RAM in the Raspberry Pi 3 may potentially pose issues in overall performance when running memory intensive applications or running python programs which require many additional packages for audio manipulation and GUI creation. The Raspberry Pi 3 also requires the use of a microSD card for program and operating system storage, it can handle between 8-32GB of storage where 8GB is the minimum amount of storage to be able to install Raspberry Pi OS on to the Raspberry Pi, some users of the Raspberry Pi claim that the upper limit of its microSD storage support is 1TB of microSD storage space. The Raspberry Pi 3, like its successor, can run on many different operating systems including Raspberry Pi OS, Ubuntu, Mate, LibreElec and Windows 10. The wide range of available operating systems lends to the Raspberry Pi 3's versatility in available software and programs that can be run and executed through it. The Raspberry Pi 3's combination of CPU, GPU, and RAM while not extremely impressive may be barely adequate enough for the intended application in this project, from a quick glance it seems the limiting factor of the Raspberry Pi 3 is its low amount of RAM that is available for the SOC to use. The low amount of RAM may not be a dealbreaker when selecting this single board computer however because the Raspberry Pi 3 can often be found at a much lower price than the Raspberry Pi 4 at as low as \$80 (MSRP \$35).



## INTERFACE

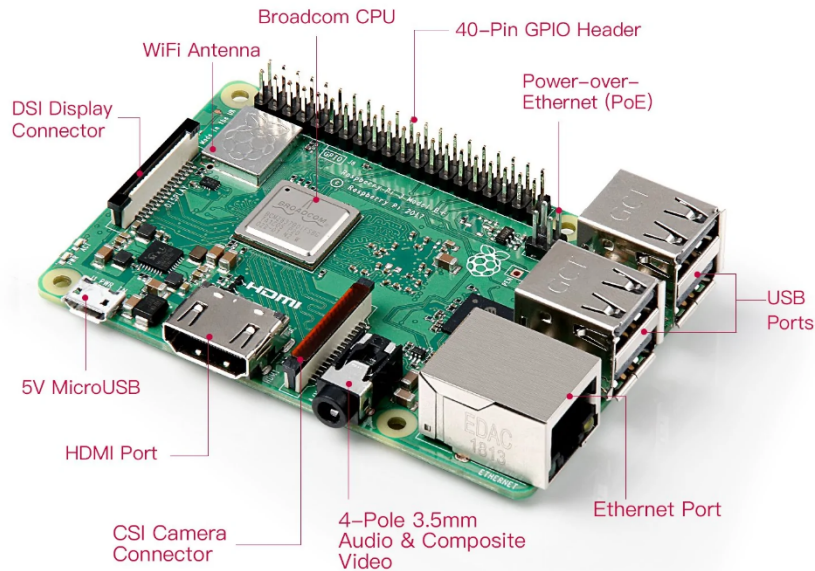


Fig 24. Raspberry Pi 3, Model B+

### 3.5.3 Nvidia Jetson Nano

The Nvidia Jetson, unlike the Raspberry Pi, is a small computer module that is intended to be used with Nvidia's Jetson Nano Developer kits which add a carrier board that the Jetson Nano module slots into to complete the small computer platform. The Jetson carrier board adds I/O ports that the Jetson Nano can use such as USB 3.0, Ethernet (Gigabit speed), MicroSD, HDMI, DisplayPort, Camera connectors, and a 40 pin I/O header. The Nvidia Jetson Nano uses a Quad-core ARM Cortex-A57 CPU at 1.43Ghz that utilizes a 64-Bit memory bus. The Cortex-A57 is similar in performance to the AMD Opteron 290 (Desktop CPU), Intel Core2 Duo T9300 (Laptop CPU), and Qualcomm Snapdragon 615 MSM8939 (ARM based CPU for Android Tablets and Smartphones). In relation to the Cortex-A72 of the Raspberry Pi 4, the Cortex-A57 of the Nvidia Jetson Nano performs roughly twice the speed of the former. This large jump in performance makes the Nvidia Jetson Nano the most capable computer out of the three that have been compared and seems to be the highest performing computer that can be utilized for the project. The Nvidia Jetson also includes a Nvidia 128 CUDA core GPU from Nvidia's Maxwell architecture (GeForce 700/900 series GPU architecture) which can provide immense amounts of graphical capability in comparison to the integrated Broadcom graphics processor of the Raspberry Pi 3 and 4. The Nvidia Jetson Nano features 4 GB of onboard LPDDR4 memory which utilizes a 64-bit memory address and should prove to be more than an adequate amount of RAM for the use case of the project. The Nvidia Jetson Nano also features 16 GB of built in

eMMC solid state storage for program and operating system storage which is a great advantage over the Raspberry Pi 3 and 4 which both do not include onboard storage and instead require the use of a microSD card to store programs and operating systems on the device. The Nvidia Jetson Nano can be purchased for \$298 which makes it nearly double the price of the Raspberry Pi 4 and triple the price of the Raspberry Pi 3. With the minimal performance increases over the Raspberry Pi 4, the Nvidia Jetson Nano does not seem like a viable option in terms of price to performance in relation to the Raspberry Pi 4.

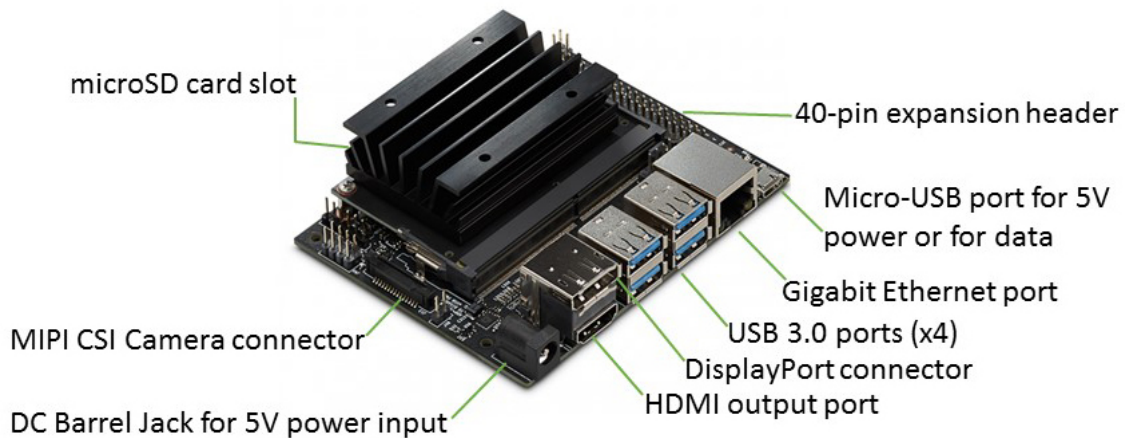


Fig 25. Nvidia Jetson Nano Dev Kit

### 3.5.4 Raspberry Pi 4 vs Raspberry Pi 3 vs Nvidia Jetson

Comparison	Raspberry Pi 4	Raspberry Pi 3	Nvidia Jetson Nano
Cost	\$158	\$80	\$298
Processor Speed	Faster	Slowest(1/2x Pi 4)	Fastest (2X Pi 4)
RAM	2-8GB	1GB	4GB
I/O	All	All	Missing 3.5mm

			Audio
Size	Smaller	Smaller	Larger
Power Requirement	5V DC	5V DC	5V DC

Table 5: Microcontroller Comparisons

### 3.6 Display Technologies

The project requires an electric display to indicate to the user which settings are being actively used. The display will need to pass various specifications in order to be used in the project, such as: Size, Light Intensity, Features, Cost, Availability, Performance, and Compatibility. Each of these specifications will be considered for the final design. The display will be placed within the small container that holds the other electronics and components and the screen will be fitted on the surface of the container. The two types of displays in consideration for the project are Light-Emitting Diodes Displays (LED) and Liquid Crystal Displays (LCD). Both types can perform the general goal of the project but only one will be used. The most crucial specification is the compatibility of the display to the designated microcontroller. The display would have to properly process information from the microcontroller and emit information like battery life, mode setting, and lighting's brightness level. For the project we have chosen to use an LCD display for our particular build of the project to be more practical in satisfying all the design constraints the project has. In the next two sections the reader will be given a more in-depth understanding of LED and LCD displays, so the reader can understand why the LCD display was chosen. Potential products for the LCD and LED Displays need to be compatible with the Raspberry Pi 4 Microcontroller that is going to be used. Meaning the Display has to have either a micro-HDMI connection port or be able to connect to the two lane MIPI DSI Display Port on the microcontroller board.

Touch Screen capabilities are on the rise amongst new displays, eliminating the need for external switches/buttons on the device. However, it brings a new task of finding a display that is compatible with the project's microcontroller, meaning that the display will need to have an interactive user's interface. The touch screen will need to install and update any required software, typically written in python, to generate a graphical user interface (GUI). The touch screen will then act as the switch for the various modes and settings the project will have all from the GUI. The main issue from touch screens is the rise of errors in the sensors for the touch screen, meaning that there are possibilities of reading the wrong condition. Example having two inputs on the screen at the same time, the microcontroller might not know how to respond unless it is properly taken into account. This is due to how touch screens operate with a conductive layer that reacts when touch occurs from the small electrical charge that flows from the finger to the conductive layer.

### 3.6.1 Liquid Crystal Displays

Liquid Crystal Displays (LCD) was officially discovered in 1964 by George H. Heilmeyer at RCA Laboratories, though the actual liquid crystal research can date back to 1888 by the researcher Friedrich Reinitzer. The importance of their findings is what pushed cathode-ray tube (CRT) displays into being obsolete. This is due to the efficiency, size reduction, and operational lifespan. LCD screens also do not use phosphors, so they do not experience Burn-Ins as often as CRT displays. The principle of how LCD works is that they block light from a source when current is being provided onto the liquid crystal layer. In older structures LCDs would have mirrors within them to reflect the outside light into the liquid crystal layer, going through the electrode layers, diffusers, polarizers, and films.

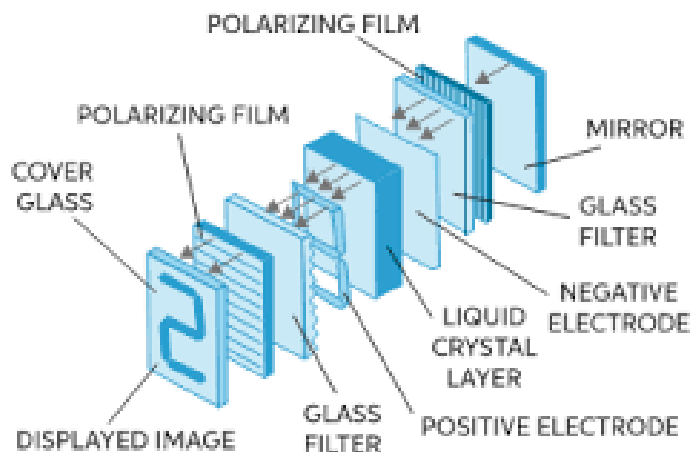


Fig 26. LCD Display Structure

This was before the introduction of LED backlighting, which is now in common use, also allowing us to improve the design of LCD screens. The integration of LEDs is what made the LCDs the most used displays, meaning that the cost for these displays is not as high for all the advantages they bring.

### 3.6.2 Light-Emitting Diodes Displays

The first practical Light-Emitting Diodes (LED) displays originated in 1968 by Hewlett-Packard (HP). LED displays are known for their higher contrast levels and utilized in various applications that demonstrate higher resolution than being projected. An LED display contains LEDs that illuminate various colors and light intensity to fabricate an image from a certain distance. In the early development of these displays the LEDs were not as advanced, meaning larger displays were needed and lower quality images were produced. However, with the introduction of smaller LEDs that have the capability of producing various color hues and light intensity, the display

quality has increased significantly, and the display size now can vary immensely depending on the application. Such an innovation has now made it possible to accurately depict desired images from the combinations of LEDs being flashed on/off that are signaled from a microcontroller. The most common usage for LED Displays is actually used as backlighting for LCD screens, improving the LCD performance with the contrast. As new techniques and technology develop comes the introduction of more efficient displays. Quantum dot LED (QLED) and Organic LED (OLED) displays are what the current displaying technologies utilize with LEDs. QLED Displays utilizes phosphorescent crystals between the LED backlighting and the LCD Layer, from this it enhances the images being played from the LCD layer with the color combination of Red, Blue, and Green.

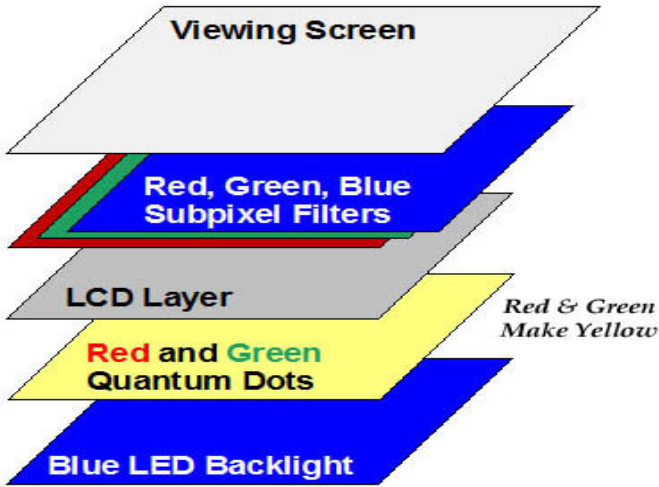


Fig 27. QLED Display Structure

OLED utilizes a thin carbon based semiconductor layer that acts as a solid-state device when powered the device emits light through a transparent electrode. The intensity of the light is determined by the total current being applied and the color is determined by the material being used. There are three common techniques for OLED lighting structures: Single-Stack, Stacked, and Striped OLED.

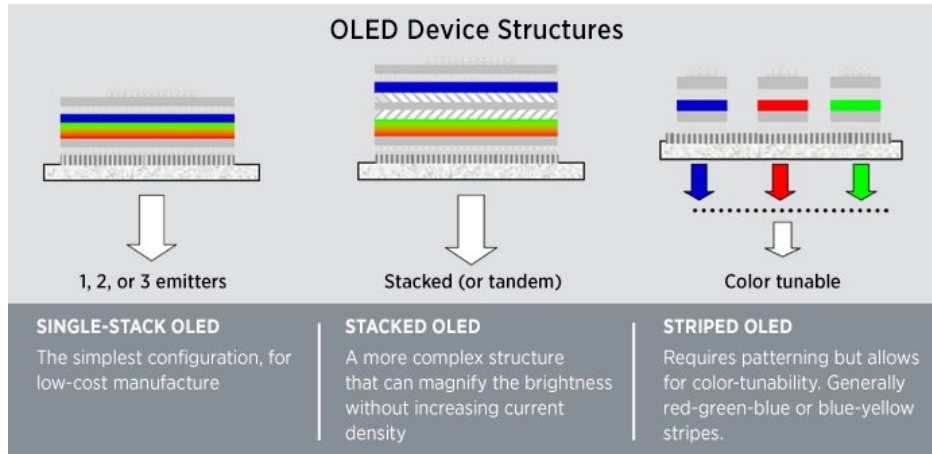


Fig 28. OLED Display Structure

OLED in comparison to QLED and LED displays, OLEDs are more efficient in saving energy due to their adjustable light intensity levels. The reason being is that QLED's LEDs are set to one brightness level and are not as adjustable as the OLED's LED. Meaning that OLED display can be made to be very thin in comparison to QLED and even molded in various forms due to its flexibility. An issue though that both types of displays experience is Burn-Ins, which is the result of an image that's been projected on the display for a relatively long time etches itself onto the display's screen. It is a common problem for the common display, though with new research into MicroLEDs, it is now possible to reduce and even eliminate the chances of Burn-Ins. However, MicroLEDs are not cheap and will not be cost efficient for the application this project requires.

### 3.6.3 LCD vs LED Displays

As a comparison between the discussed displays such comparisons being: Size, Resolution, Features, Cost, Availability, Performance, and Compatibility. The comparison is done on the potential products that would be chosen for the display of the device. For the LCD the product is the 5inch Capacitive Touch Display for Raspberry Pi, DSI Interface, 800×480. While for the OLED the product selected was the 5.5" HDMI OLED-Display with Capacitive Touchscreen (V2.0). Both would be viable however the one that the team is going to choose is the LCD display, mainly for the price and being able to do the same objective tasks, the only tradeback is the significantly lower quality in resolution.

Comparison	LCD	OLED
Cost	\$39.99	\$119.00
Screen Size	5 Inches	5.5 Inches
Features	Touch Screen	Touch Screen
Availability	1000+	2
Resolution	800x480	1080x1920
Compatibility	Raspberry Pi 3A/3B/4/4B	Raspberry Pi 3A/3B/4/4B
Power Consumption	5 VDC	5 VDC

Table 6: LCD vs OLED

### 3.7 Audio Transmission

There are many methods to transmit audio from a microphone. The options we researched were to transmit the audio over Bluetooth or wire it directly into an ADC converter into the Raspberry Pi. The added portability of using bluetooth would allow the musician to not have to worry about a cord while moving. However, it would add weight to the instrument as well as the musician as the use of a microphone stand wouldn't be possible. The first Bluetooth method would be a Bluetooth audio transmitter and receiver module. The second would be over Wi-Fi using wireless transmitters and receivers. The third would be a tether or cord. These are vastly different options in battery life, sound quality, cost, and customizability.

#### 3.7.1 Bluetooth Audio Transmitter and Receiver Boards

This method uses two boards, a transmitter board, and a receiver board. The transmitter board would be plugged into the microphone either through an EP or AUX connection through a converter and then would transmit the audio to the receiver. The receiver would then plug into the ADC converter using an AUX or EP cable. The boards are very cost effective, however since they don't have any casing, cases would have to be made in order to prevent damage. The boards don't include batteries therefore they'd have to be purchased separately and soldered on. This option allows a much more customizable approach as the battery life and the sizing of the case are not fixed. Overall, this would allow the product to be more portable with minimal weight added.



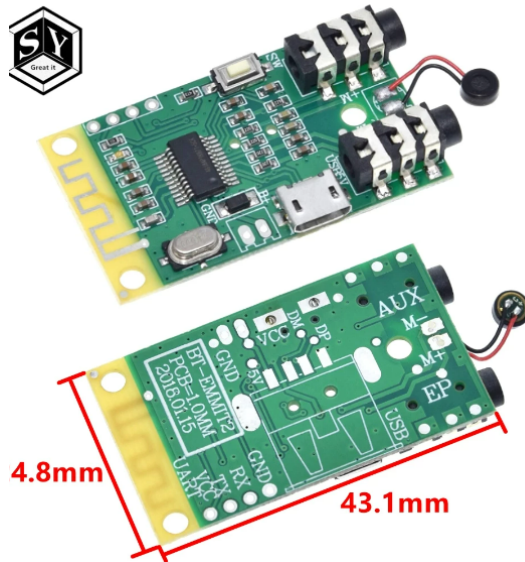


Fig 29. Bluetooth Transmitter and Receiver Boards

### 3.7.2 Bluetooth Audio Transmitter and Receiver Plug-ins

This method uses Bluetooth devices that plug directly into a microphone and transmit it to the receiver end. This has a built-in battery with a larger range as well. However, due to the plug-in nature of the device it would limit the microphone options or require an adapter. This transmitter wouldn't work with the condenser microphone or the contact microphone as it's meant to plug directly into a handheld microphone. This option is also much more expensive and would increase the cost of the project substantially.



Fig 30. Bluetooth Audio Transmission Plug-ins



### 3.7.3 Wired Audio Transmission

This method uses a wire to go in between the mic and the ADC converter. This prevents mobility and adds stress if the musician is moving around. However, this would provide the best sound quality as it wouldn't lose as much data during transmission. This does also limit the range as the system has to be close to the musician for the cable to be applicable. Using wired transmission would also reduce lag in the system. Due to the nature of this project the lag between the notes being played and the LEDs responding needs to be as little as possible so that is an important contributing factor. This would be the easiest and most reliable method to use.



Fig 31. Wired Audio Transmission

### 3.7.4 Bluetooth vs Wi-Fi vs Wired Audio Transmission

Comparison	Bluetooth Audio Transmitter and Receiver Boards	Wi-Fi Audio Transmitter and Receiver Plug-ins	Wired Audio Transmission
Size (inches)	1.7x1+case	3.9x1.2	N/A
Weight (lbs.)	N/A	0.0625	N/A
Cost	\$5-15	\$100-200	\$15-20
Range (feet)	32.8	90	25
Sound Quality	Average	High	High

Table 7. Audio Transmission Comparisons

## 3.8 Analog-to-Digital Converter

For this project we've decided to create an analog-to-digital converter (ADC) on a printed circuit board. Analog-to-Digital converters convert an analog signal which is continuous into a discrete digital signal which is a series of 1s and 0s. This is done because analog signals are more difficult to store. Also by converting them it is much easier to trim noise off of the signal and to process the signal than in the analog form. For example, for this project we are converting it to be able to process it using fast fourier transforms. They do this by approximating the signal using different methods. The main thing to be concerned about when dealing with ADCs is the Nyquist rate which states in order to prevent aliasing which is a measurement error in the signal the sampling rate must be twice that of the maximum frequency. There are five different setups for an ADC considered for this project each with their own pros and cons. The key components to compare are resolution, size, price, and latency.

### 3.8.1 Dual Slope Architecture

The dual slope ADC works by using a switch to go in between the input voltage and the negative reference voltage. The input then goes into an integrator and then a comparator and gets compared to 0. Then it goes into an AND gate with a clock signal as the other input. That gets sent into a counter. When the counter reaches all ones it switches the input from the  $V_{in}$  to the negative reference voltage. Then it restarts when the reference voltage's output reaches 0. The important thing to note about this ADC is the first stage when  $V_{in}$  is the input it is rising for a fixed amount of time until the timer reaches 1111. Then for the second stage when the reference voltage is the input it rises until it goes back down to 0 where the first stage started. This causes high accuracy or resolution of the original signal. However since it has to continuously go up and down it results in high latency in the system.

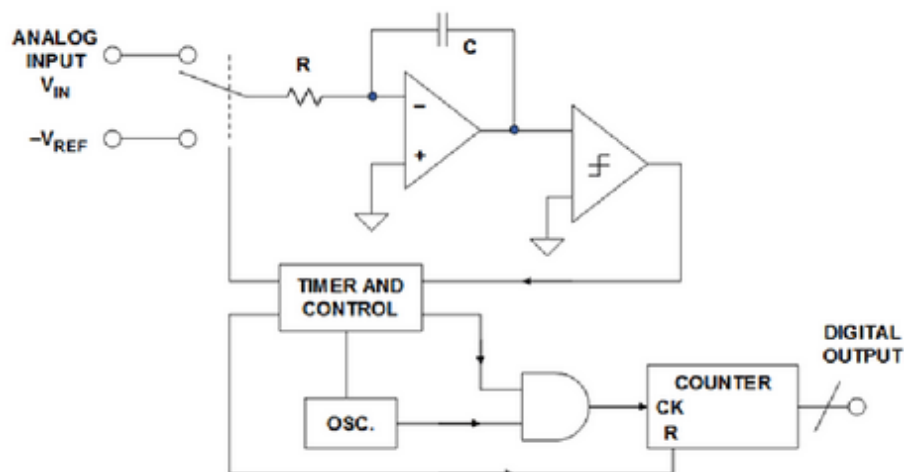


Fig 32. Dual Slope ADC Block Diagram

### 3.8.2 Flash Architecture

The flash ADC works by using comparators to find each bit in the digital output. It does this by having the reference voltage go through a line of resistors with each resistor branching off to the negative input on its own comparator. The positive input for each comparator being the analog input signal. It then compares which one is higher if the input is higher it outputs a 1 if the reference is higher it outputs a 0. This then goes into a priority encoder which translates the number into binary. If multiple comparators output 1 it takes the highest priority 1 as the real output and ignores the rest. This design is very fast as it does not have to loop through the system like the other types. However to use this for high resolution applications it would get very big and costly as you would need  $2^n$  comparators when  $n$  is the number of bits. Therefore, for 16 bit resolution you would need 65,535 comparators which isn't realistically feasible. This type does work with non-periodic signals as it can pick up anywhere in the cycle.

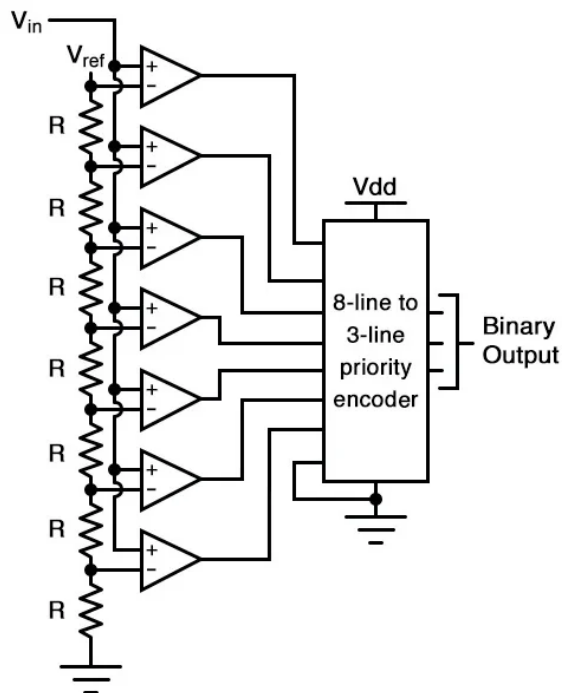


Fig 33. 3-bit Flash ADC Block Diagram

### 3.8.3 Pipelined Architecture

The pipelined ADC works by sending the input signal through a flash ADC then out to a DAC and subtracting that from the original signal. It repeats this cycle through multiple stages in series. This allows the previous stage to start on the next input signal while it's processing. Each stage works by narrowing the quantization levels. For example with a 3-bit flash ADC that will have 8 levels from which it determines which the input is at. Then it would break that level into 8 levels for the next stage and so

forth depending on the resolution. Then at the end it would go through a final flash converter to determine the remaining bits. This decreases the latency in the system by using the flash converters more efficiently by doing so it also allows for higher resolution. It still requires a high number of comparators but instead of 65,535 it would be  $n/2 * (2^{(n/2)}-1)$  which is 60.

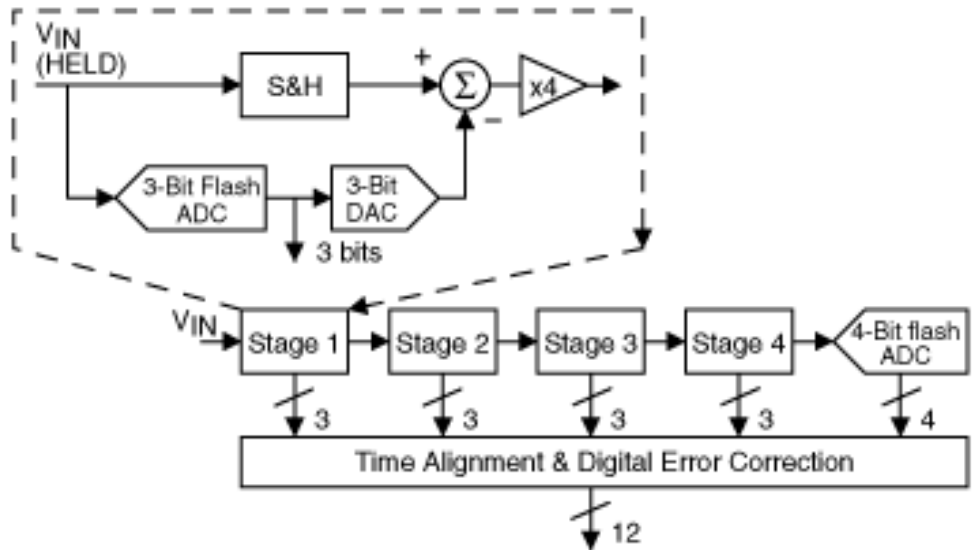


Fig 34. 16-bit pipelined ADC Block Diagram

### 3.8.4 Sigma-Delta Architecture

Sigma-Delta ADCs work by sending the analog input through a difference amplifier to add the feedback back into the signal. Then through an integrator to do noise shaping by removing noise from the lower frequencies and pushing it to higher frequencies. This works as we don't use the frequencies that the noise has been pushed to. That then goes into a comparator to compare it to the ground which determines if the input is positive or negative. Then through a D-latch and 1-bit digital-to-analog converter back into the differentiator at the beginning of the system. This allows the system to get many 1s and 0s whose density is proportional to the original analog input. For example, if there are mostly ones it's close to the peak of the system and with more points to average or more bits the higher the resolution. Quantization is the analog-to-digital conversion process as it is the process of taking a continuous signal and representing it as a discrete value. This process allows oversampling which makes the quantization noise more spread out which lowers the amplitude also, resulting in a high signal to noise ratio. This is helpful as the actual signal is a much higher peak than the noise and is at the beginning of the sample. This allows a digital filter to be used to cut off the high frequency noise to create a cleaner output. The output is then put through a decimator which reduces the number of values to that within the Nyquist rate to reduce aliasing. This makes the need for extra

filtering outside of the ADC virtually nonexistent. This allows the ADC to have very high resolution. Due to the noise shaping factors it is best used for low frequency applications. However it has high latency due to its process of oversampling. It is also unideal for non-periodic signals as it has to have proportions of 1s and 0s to approximate the signal.

## $\Delta\Sigma$ Modulator

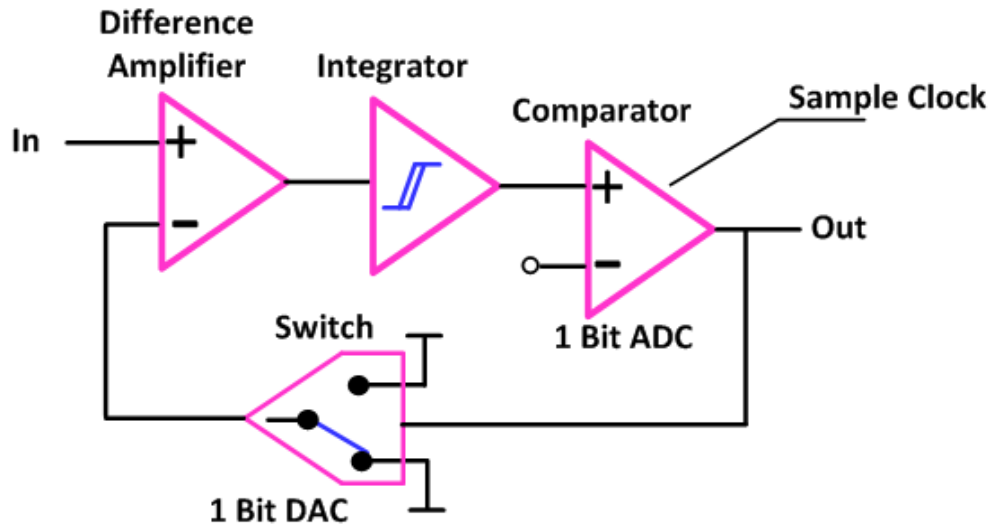


Fig 35. Delta-Sigma ADC Block Diagram

### 3.8.5 Successive Approximation Architecture

Successive approximation ADCs hold the analog data and send it through a comparator which compares it to the output of a digital-to-analog converter. It does this by starting with the most significant bit and if that is higher than the analog data it sets it to 0. However if it is less than the analog data it sets it to one and does that until it hits the least significant bit. Then each bit is stored in the successive approximation register until every bit is weighed. This means that with this set up it has to go through the whole system for however many bits there are. For example if it wanted 16-bit resolution it would have to go through the system 16 times once to approximate each digital bit. This can add latency to the system for applications that require higher resolution for example with audio it is recommended to have anywhere in between 16 and 24 bits. They do work for non-periodic signals as the conversion can start anywhere in the cycle. They also allow undersampling which can allow aliasing to be used to convert the signal to a lower frequency. They are also low-power which make them very useful in portable situations where batteries are used.

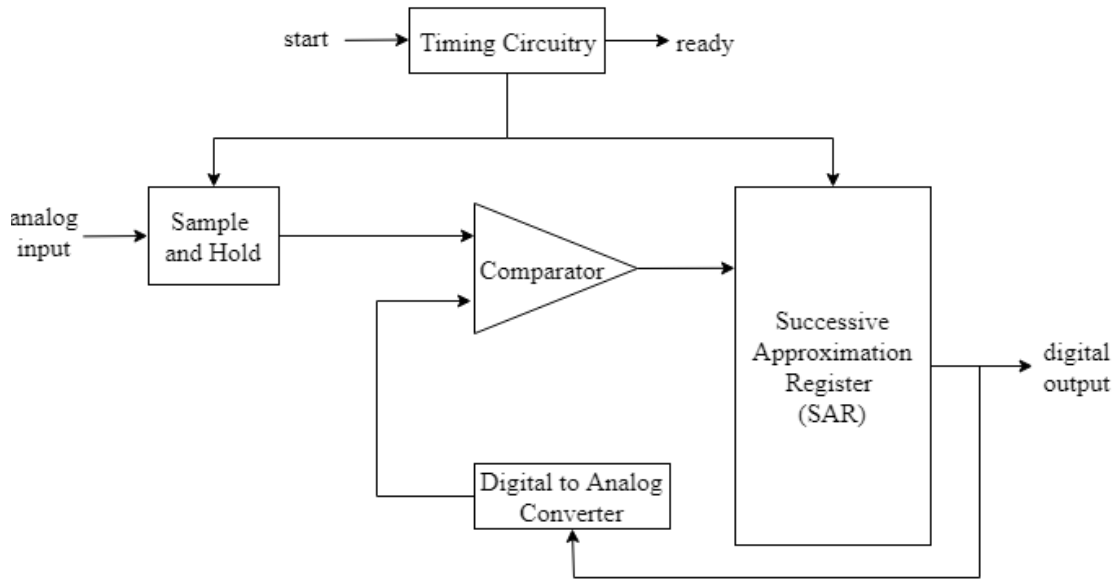


Fig 36. Successive Approximation ADC Block Diagram

### 3.8.6 Dual Slope vs Flash vs Pipeline vs Delta-Sigma vs Successive Approximation Architectures

Comparisons	Dual Slope	Flash	Pipelined	Delta-Sigma	Successive Approximations
Price	Low	Expensive	Low	Expensive	Low
Size	Small	Large	Medium	Small	Small
Latency	High	Low	Low	High	Medium
Resolution	High	Low	Medium	High	Medium
Throughput	Low	High	High	Low	Medium
Noise	Low	Medium	Medium	Ultra Low	Low

Table 8: ADC Comparisons

## 3.9 Comparable Products

Upon initial research, the most similar products to that of our project were Lumi keys (a light up piano keyboard) and children's toys. There is nothing on the market today that matches what we intend to create. This resulted in more opportunities for

research and innovation. However, it also meant we had nothing to use as a reference when it came to research and testing. There are products with similar features that contributed to the vision of our project. These products being the aforementioned piano and Razer's computer keyboards with their integrated synapse RGB features.

### **3.9.1 Lumi Keys**

Lumi Keys are piano keyboards in which the keys light up with various colors depending on the settings. The keys can light up depending on which key you need to push for the next note, or you can have pre-set lights that change color when pushed. Lumi Keys comes with a variety of interactive customization settings that make the colorful lights very captivating.

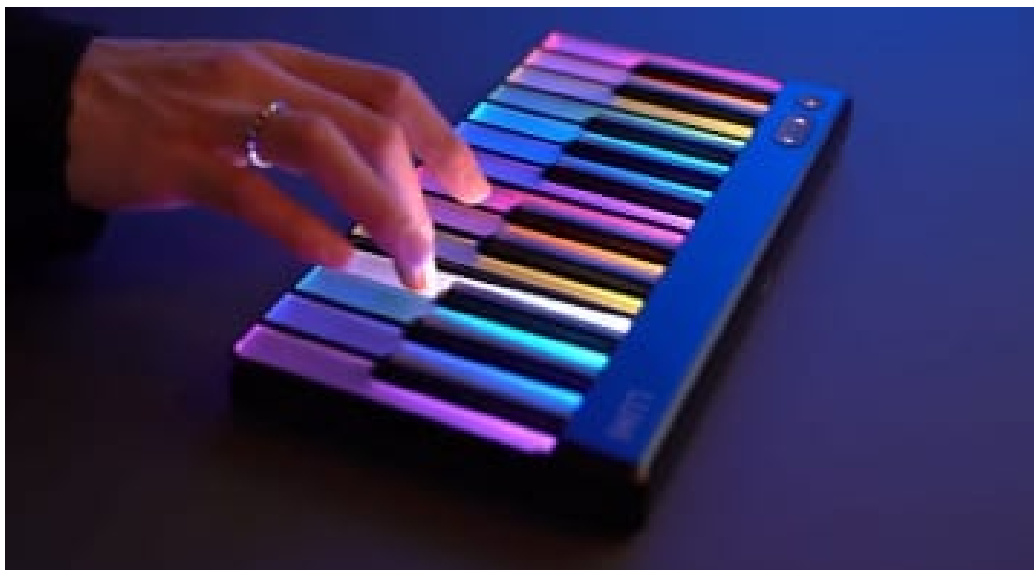


Fig 37. Lumi Keys

However, unlike our project, Lumi Keys do not change color and brightness in response to the sound being emitted from the instrument. We can look to Lumi Keys for inspiration on changing the colors of the RGB lights with changes in inputs. In our case, the input would be the different wavelengths that result from the sounds coming from the instruments.

### **3.9.2 Razer Synapse**

Razer has many different products that feature their coveted Chroma RGB lights. One of the most popular of these products are their keyboards that feature Synapse integration. Synapse is the software owned by Razer that allows their consumers to fully customize the lights on their products in any way they can think of with thousands of possibilities.

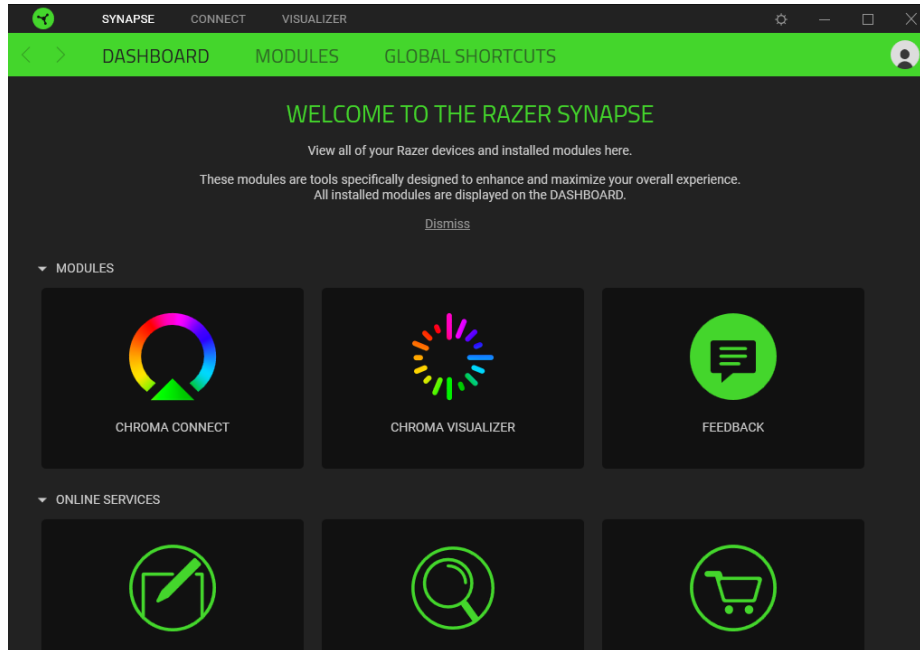


Fig 38. Razer Synapse Software

One of the predefined settings on Razer’s Synapse software is the “Audio Meter” which changes the lights on the product in response to the audio being outputted by the computer.

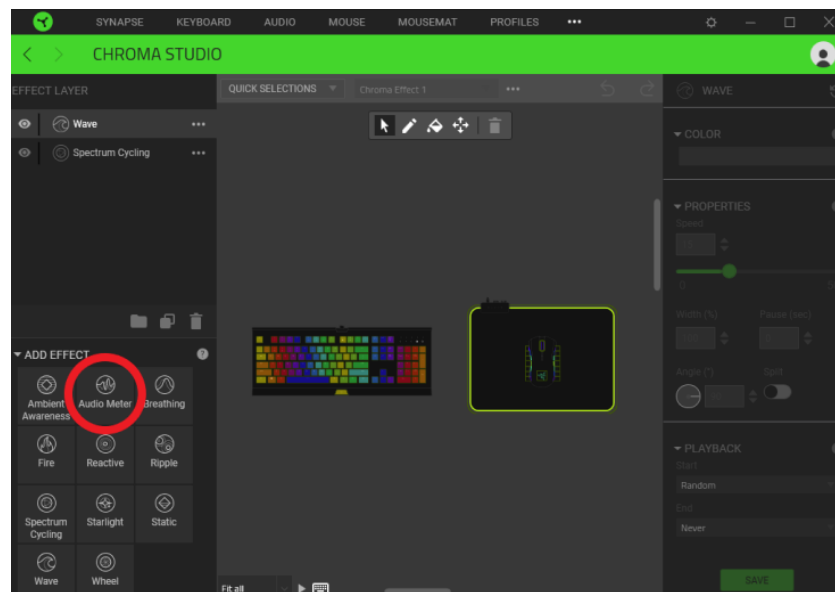


Fig 39. Razer Synapse Customization

This feature can be comparable to reading in an audio input and converting it to a code that will output a message to the LED lights in order to turn the lights on or off. However, a key difference between this and our project is that Razer’s Audio Meter



does not change the colors of each light. It assigns a specific color to each light and turns them on or off depending on the audio output in order to make it look like an audiometer similar to the below image.

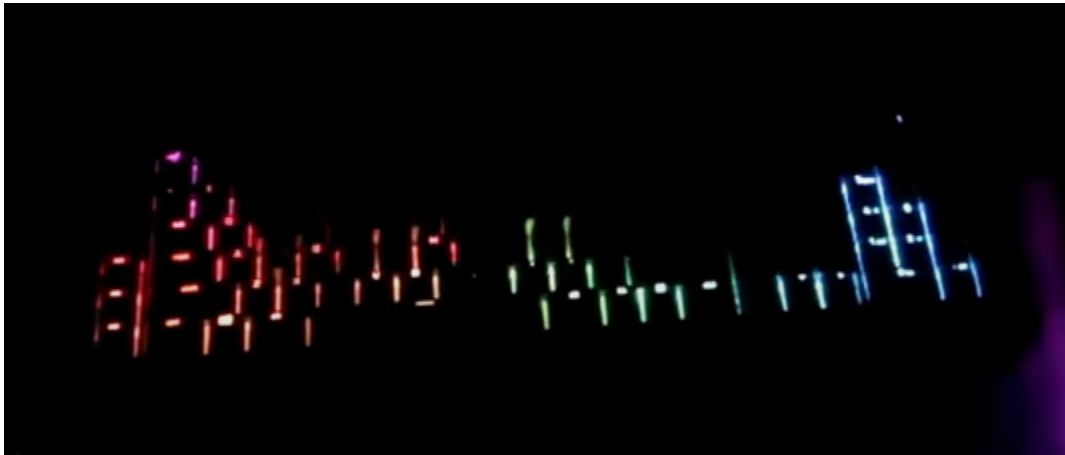


Fig 40. Razer Audio Meter Setting

If we were to implement a similar design, we would be required to change the core programming to output a change in color for each light with respect to the sound being created instead of simply turning the lights on or off.

### **3.10 Software Technologies**

In terms of programming, we initially had multiple routes we could have chosen to go down. Our initial project idea was to create an app that allowed for a fully customizable light display however, due to our time constraint, we decided to simplify it to just converting the sound being emitted from the instruments into wavelength and turning those readings into specific colors and brightness to return to the LED strips. Because of this decision, the project is not very heavy on software and will not require very complicated code. We were also told by our sponsor that he has another team working on an app that he plans to combine with our project which would make any grand ideas we had either an inefficient use of our time or the other team's time. Our first hurdle was deciding which programming language we will be using to program the Raspberry Pi that will be the heart of our project. We took the following languages into consideration.

#### **3.10.1 Programming Languages**

The Raspberry Pi can be programmed with one of three programming languages: C++, Python, and Scratch. Here we will be taking all three of these languages into consideration in order to choose which of the three we will be using for the programming. Outside of these languages, we have the ability to run any other

language by installing the compiler or interpreter onto the operating system of the Raspberry Pi 4 however, this would be too time consuming considering the given deadline for our project so we will only take the three pre-given languages into consideration.

### **3.10.1.1 C++**

C++ is a high-level programming language that was created as an extension of the C programming language. Although C is typically a functional programming language, C++ has evolved into more of an object-oriented programming language but also has features of generic and functional programming languages. The language is also statically typed which means it performs type checking at compile time. This makes it faster than dynamically typed languages.

### **3.10.1.2 Python**

Python is an object-oriented high-level programming language that supports a large volume of libraries to help aid programmers write code in a clear and logical manner. The result of Python's large library support equates to a considerable decrease in the amount of code needed to perform a similar program on a comparable programming language such as C++. Although the library support on Python lends it high versatility it does some at a cost to performance since Python is a Dynamically typed programming language meaning the languages interpreter assigns any variables at runtime based on the values contained in them in contrast to checking them during compilation which is what a Statically typed programming language like C++ does which helps improve its performance. Even with this slight deficit on speed Python seems to be the best solution for our specific application since Raspberry Pi OS has Python natively built in to help streamline the programming process. Another point advantage of Python in our use case is its library support which will be extensively used with the combination of audio processing and graphic displaying that will be required for the device's operation.

### **3.10.1.3 Scratch**

Scratch is the last of the three high-level programming languages that we can choose to code the Raspberry Pi with. Scratch is a programming language targeted at children. It is more of a way for children to learn programming skills than it is a way to create actual projects. For this reason, we quickly decided not to use this programming language for our project.

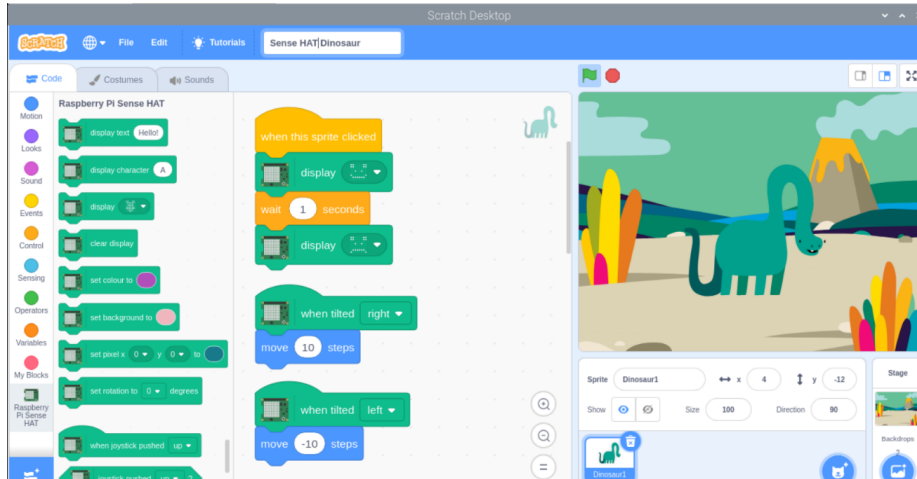


Fig 41. Scratch Programming

### 3.10.1.4 C++ vs Scratch vs Python

The following table comprises the points we used to decide between C++ and Python for our preferred language to use to program the Raspberry Pi 4.

Comparison	C++	Scratch	Python
Density	More code	Less Code	Less code
Garbage Collector	Unsupported	Unsupported	Supported
Compilation	Precompiled	N/A	Interpreted
Speed	Faster	Slower	Slower
Nature	Statically Typed	N/A	Dynamically Typed
Difficulty	Steep learning curve		Easy to learn

Table 9: Software Comparisons

After considering the various points in the table along with our familiarity with the two languages, we decided to use python for our programming needs. We came to this decision because not all of our programmers have experience in both languages and Python is the easier to learn between the two. Although C++ is the fastest language, the time it would take for half our team to learn it would be too great given our time constraints.

## 3.11 Required Mathematics

This project required calculations to take the audio input and convert it into the frequency domain. This required a Fourier analysis, however there were multiple different methods depending on the input.

### 3.11.1 Fourier Series

Fourier series is used for periodic continuous signals. It represents the function as a sum of sine and cosine functions. As more sine and cosine waves are summed over more time the more the resultant waveform is equivalent to the original input. This is useful for simplifying an input into sine and cosine but as we are trying to convert the input into the frequency domain it isn't a useful method.

$$f(x) = \frac{1}{2} a_0 + \sum_{n=1}^{\infty} a_n \cos(n x) + \sum_{n=1}^{\infty} b_n \sin(n x),$$

where

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) dx$$

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(n x) dx$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(n x) dx$$

Fig 42. Fourier Series Equation

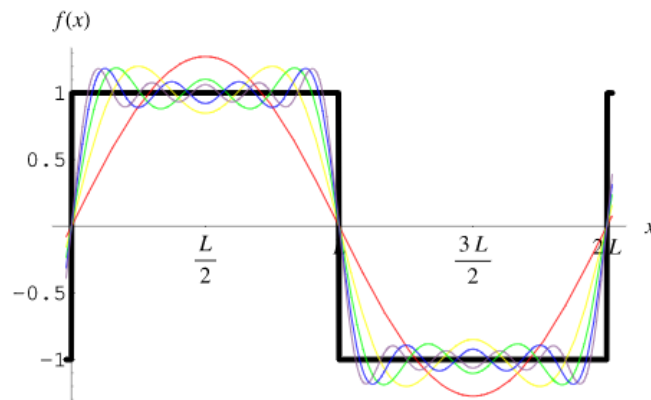


Fig 43. Fourier Series Approximation

### 3.11.2 Fourier Transform

Fourier transforms are used to convert continuous and non-periodic signals into the frequency domain. They can also be converted back into the time domain if needed. This allows the frequency or time domains to be adjusted to affect the other. In this project's case the input would be the analog input coming from the microphone as a function over time. This would then break down into a sum of sine and cosine waves and transferred into the frequency domain. This also allows the microcontroller to see the frequencies of each note being played from the input audio signal.

$$\mathcal{F}\{g(t)\} = G(f) = \int_{-\infty}^{\infty} g(t)e^{-i2\pi ft} dt$$

Fig 44. Fourier Transform Equation

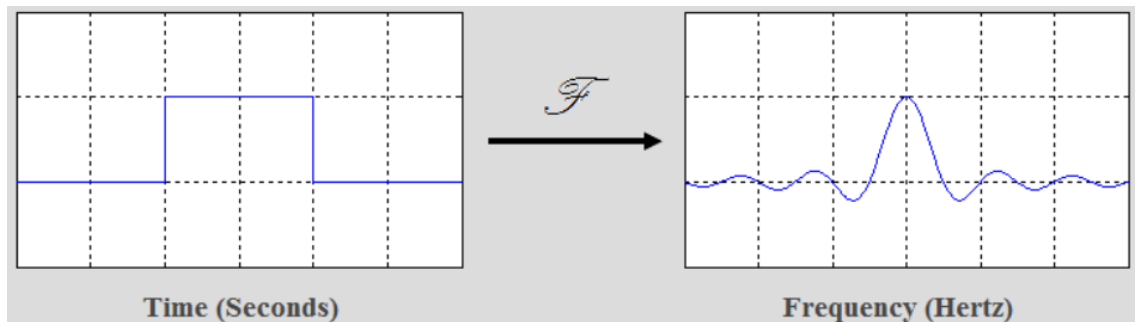


Fig 45. Fourier Transform

### 3.11.3 Discrete Fourier Transform

Discrete Fourier transforms or DFTs convert signals which are discrete into the frequency domain. Discrete signals are signals that have values at only specific points in time unlike continuous signals which have values for any point in time. An example of a discrete signal is a digital signal. DFTs are necessary to use fast Fourier transforms or FFTs. FFTs are much faster and more efficient to compute. This is due to a computational algorithm. The most popular algorithm is the Cooley-Tukey algorithm which allows the DFT to be broken down into smaller DFTs that can be more easily calculated. This allows FFTs to be used for many applications of digital signal processing. One of those examples is filtering noise out of a signal. This is done by doing the FFT removing the unwanted frequencies and then calculating the inverse FFT to get the original signal without the noise interruption. This project will use the frequency received from the Fourier transform and based on that determine the color the LEDs display.

$$x(n) = \sum_{k=0}^{N-1} X'(k) e^{j\frac{2\pi}{N}kn} \quad n = 0, 1, \dots$$

Fig. Discrete Fourier Transform Equation

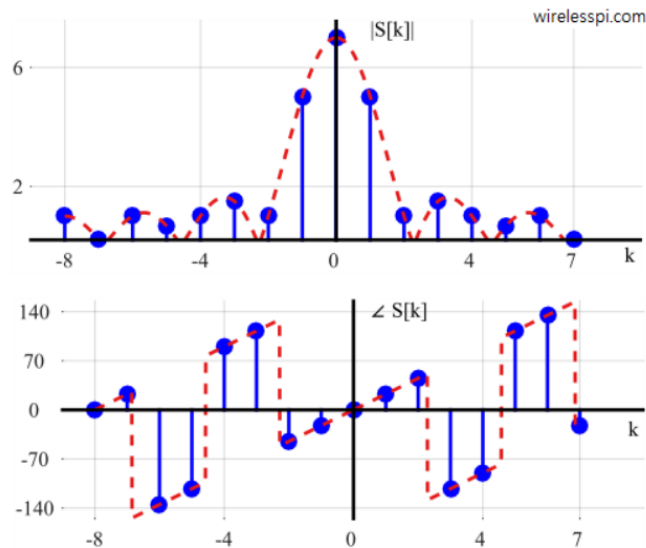


Fig. Discrete Fourier Transform

### 3.11.4 Mathematical Process Selection

The mathematical process selected is the FFT. However, to make this possible the input from the microphone needs to be put through an ADC converter. Fourier transforms were the best method to use to find the frequency of the audio input. However, due to the nature of this project the lag between not being played and the LEDs reacting need to be minimized as much as possible. Therefore, the best method would be to convert the signal from analog to digital and perform an FFT. This was also the best method as it has a function in a library for most common programming languages. Therefore, the microcontroller can do the calculation internally using a preprogrammed algorithm. However, one thing to consider is the sample rate and the FFT size when calculating. The sample rate, which is the amount of time between each value from the input, must be twice that of the frequencies measured. The FFT size which is the number of divisions in the measured window multiplied by 2. Thus, the frequency resolution is the sampling rate divided by the FFT size. However, as long as the two parameters are met this allows the process to be made as quickly and efficiently as possible.

## 3.12 Instruments

One of the first decisions we had to make for this project was to decide what instrument we will be using for testing our product. The final goal is to have a product that can be integrated into any instrument so that it can be used for everything from solo recitals to complete orchestras. We began with four options: piano, flute, trumpet, and drums. Most instruments will simply have a microphone standing in front of them unless stated otherwise.

### 3.12.1 Piano

Our first and most desired idea was to use a piano or keyboard. The semi-translucent white keys seemed like an optimal way to display the lights without simply placing them on top of the instrument but by placing them inside the keys themselves. The microphone would have had to have been placed inside the piano where the strings are being struck in order to pick up the best quality sound for our program. However, due to the highly expensive costs of pianos, our only hope was to be given permission to use a piano from the music department at the University of Central Florida and we deemed it highly unlikely they would allow us to test their instruments. Please note, in the following figures, the microphone placement is indicated by the blue box.

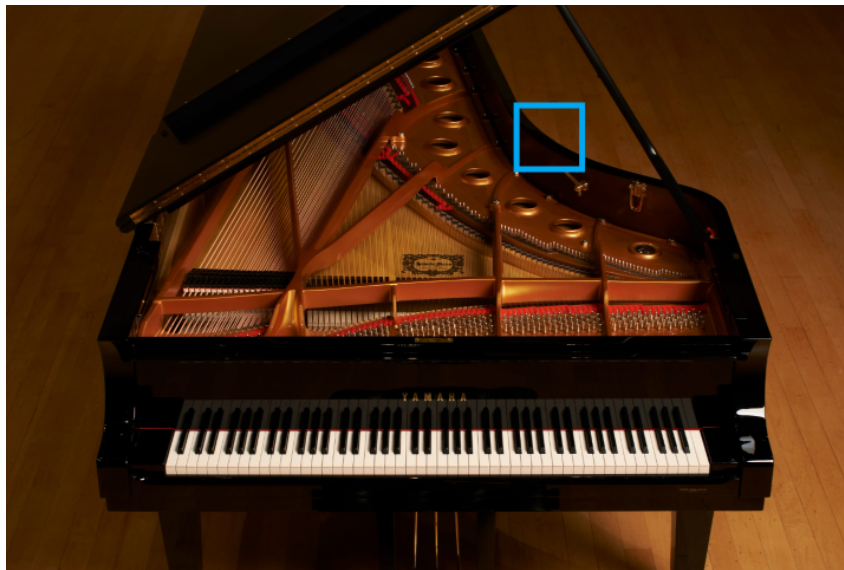


Fig 50 Microphone Placement for pianos

### 3.12.2 Transparent Instruments

Continuing the same thought process as the pianos, our next idea was to use transparent instruments. Possibly made of glass or plastic. The ability to have the lights be visible from inside the instrument was very appealing to us as a group as we felt it would better display the lights and colors throughout the entire instrument. One

possible example was a glass guitar, however, due to the extreme pricing of these instruments, we quickly decided against it. One possible example was a glass guitar. However, due to the expensive nature and rarity of these instruments, we quickly decided against it as these instruments can cost upwards of tens of thousands of dollars and are more fragile than their wooden counterparts.



Fig 51. Transparent Electric Guitar

### 3.12.3 Trumpet

Upon discussions with our sponsor Dr. Leinecker, we were told he had a trumpet on hand and that he was willing to loan us in order to conduct our testing for the project. This completely distinguished any concerns we had in regard to economic constraints and the general shape of a trumpet was ideal to place the lights onto the instrument without damaging it. Although, as a brass instrument, this meant giving up on our hopes of having the lights travel throughout the entire instrument.



Fig 52. Bach TR300H2 Student Bb Trumpet



### 3.12.4 Drums

Our last idea was to use a drum. Not an entire drum set but a marching band drum. With these, we could simply line the edge of the drum with LED lights facing inwards and they would shine on the surface of the instrument or line the inside of the instrument to face upwards through the head of the drum if the drumhead is transparent. The cost of a single marching snare drum can vary depending on its size and which percussion brand it is from but generally a full size marching snare drum can cost upwards of \$659 which means it is not the most ideal if we were to purchase one for the use of the project however, a marching snare drum may be able to be sourced from the UCF Marching Knights or another local percussion group that would be willing to let us use a snare drum for the purpose of this project and in exchange they may show interest in our project with its visualization parameters and may want to pick it up after the project is finalized which could help mass produce the device so it can be available to the public. One of our group members also owns a drum set with multiple individual drums that could be adapted for use with the project which could be a backup option if a marching snare drum cannot be sourced for use with the project.



Fig 53. Tama MS1412TSBK Marching Snare Drum

### 3.13 Part Selection

After conducting the necessary research and discussing each component's options with the entirety of the team and our sponsor, we have come to the decision to use the following parts in our project. Note that this is prior to physically testing the components and is liable to change in the future.

### 3.13.1 Microphone

Following extensive research and weighing out the pros and cons of each microphone with our sponsor and team we came to the conclusion that the Clip-On Condenser microphone will best suit our needs for the project, It may be on the more expensive side in terms of microphones that were considered however, its portability, versatility, and uncompromised sound quality pushed us to consider its use. In comparison to the contact microphone which can only record instruments that produce high amplitude vibrations, Clip-On microphones can effectively record any instrument due to its traditional microphone structure. We ruled out the Shotgun microphone first due to its large size and footprint needed for its operation, as well as its inability to move with the performer if they so choose to move during their performance.



Fig 54. Clip-On Microphone

### 3.13.2 LED Strip

After researching both the Chip on Board (COB) and Surface Mounted Device (SMD) light strips, it was an easy decision to use the SMD strips. Although the COB light strips emit a smoother light than the SMD strips, the SMD strips are far more customizable with different colors than the COB strips are. This feature is pivotal to our project as our main goal is to emit lights that change colors according to the audio being emitted from the instruments.

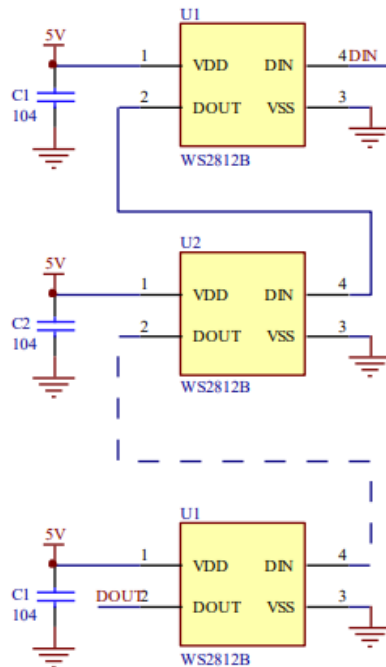


Fig 55. Circuit Schematic for 3-section-LEDs

### 3.13.3 Power Supply

After comparing the battery options, it is clear each battery option had its own downfalls. They would allow more mobility in the system with added complications. The rechargeable lithium ion battery pack had substantial battery life but would be too bulky and limit the mobility. The Lithium polymer battery with power boost would be more complicated for less battery life than the Raspberry Pi battery pack premade. The Raspberry Pi prebuilt battery pack was the easiest option with average battery life and would just need to be plugged in and adhered to the case. The Raspberry Pi hat would end up needing a secondary battery to reach the battery life necessary which would be costly as well as add complications. However, the Raspberry Pi hat adds many features to the Raspberry Pi as well as battery life such as a low power mode and programmed shut down so as not to lose data. The biggest complication however would be that they would need to be recharged and that the system could run out of power during a song. This led us to decide the wall charger would be the safest, least complicated, and cheapest option.

### 3.13.4 Single Board Computer

Following extensive research and discussions amongst team members and the project sponsor we ultimately decided to choose the Raspberry Pi 4 for our single-board computer. In terms of cost and performance it was our middle of the road option between the Raspberry Pi 3, and Nvidia Jetson Nano. The reason we chose the Raspberry Pi over the Nvidia Jetson Nano frankly was cost to purchase and overall community support, the Nvidia Jetson Nano is not often used for consumer homebrew

projects therefore there is little documentation for it and less information of how to effectively use it with the other parts that we selected for the project. The reason why we chose the Raspberry Pi 4 over the Raspberry Pi 3 is overall performance, availability, and the Pi 4's improved feature set. The Raspberry Pi line was a certain choice for the project due to its immense community support and documentation since it is the golden staple for small board computers.

### 3.13.5 Display Screen

The display chosen was the 5" DSI Capacitive Touch Display for Raspberry Pi (800×480). The reasoning is that the Touch Display is very adaptable in terms of software, its size fits within the constraints, the cost is relatively low, uses a 5VDC (450mA~500mA) for power source, and is capable of touch screen. The hardware's resolution is 800x480 with a 60Hz refresh rate, which is optimal for the graphical challenges that are going to be used for. In Fig 56. LCD Display is the part that will be bought.



Fig 56. LCD Display

### 3.13.6 Audio Transmission

After researching the different methods to transmit audio over distances it became obvious that in order to get high quality audio transmitted over a distance it would either have to be wired or it would add considerable cost to the project. The receiver and transmitter boards while battery life and size was mostly customizable wouldn't allow the best sound quality and would take longer to transmit than a wired connection. The transmitter and receiver plug-ins were very expensive and would limit the type of mic used due to the nature of the plug and size. The wired connection

would be a reliable, fast, and quality connection though not as portable. Therefore, in order to preserve the audio quality and still maintain some mobility we decided to use a long wired transmission.

### **3.13.7 Analog-to-Digital Converter**

After researching the different architectures possible for an ADC, there were four popular types: dual slope, flash, pipelined, successive approximation, and sigma delta. They each had their own drawbacks as if the latency was decreased so was the resolution. The main factor considered was that for audio display conversion 16-bit would be the best resolution option. Dual slope architecture had high resolution but also high latency and would be unusable for continuous input applications. Flash architecture was ultra fast but would get unrealistic and expensive for 16-bit resolution. pipelined architecture partially fixed the resolution problem with the flash architecture however there still wasn't high enough resolution for our application. Successive approximations architecture is average across the many different categories; it's a multi-purpose design however there was still a better option. Delta-Sigma architecture has the high resolution while not having the lowest latency; it still works with continuous inputs. The most contributing factor though is the noise shaping which is especially important when dealing with audio as we need as little noise as possible to get accurate readings from the fast fourier transforms. Therefore, the delta-sigma architecture was the most relevant for the application of this project.

### **3.13.8 Programming Language**

After researching the required parts for the project, we were given 3 main options to use as the programming language for the Raspberry Pi. C++, Python, and Scratch. Scratch was never considered because it is a tool meant to teach children how to program therefore making it more realistic to say C++ and python were our only two options. Upon comparing the two programming languages, we would have liked to use C++ due to its higher speed compared to Python however, the learning curve for C++ was too high compared to Python and we would have wasted too much time learning the language before we could even work on it. We were also informed by our sponsor that he is sponsoring another group specifically for the app portion of the project that he is planning to merge with our group, therefore using Python would translate easier between the two groups due to its easier readability. For these reasons, we decided to do all our programming in Python.

### **3.13.9 Instruments**

Upon careful consideration, our instrument options realistically fell to two. The marching snare drum and the trumpet from our sponsor. For the initial stages of our project, we will be using the trumpet. We thought it would be best to save the extra funds for any miscalculations that may occur and require us to get new parts. The fact

that we would not have to spend any of our budget greatly outweighed our desire to have a perfect display when considering our economic and time constraints. Although, one of our members later discovered that he has the ability to source a drum for a limited time. This makes it possible for us to use both the trumpet and a drum for testing. Thanks to these, it has opened more doors for testing. Having a second instrument to test with gives us the ability to conduct our research and testing while playing two instruments simultaneously in order to see how they will affect the incoming data.

### **3.13.10 Box Design**

The box design is going to be what houses the electronics and components needed for the device to work. Meaning that the device needs to be large enough to have the display shown, have the microcontroller set without too much contact to other components, and have some of the ports exposed such as the microphone connector port, and also have the battery pack inside as well. The biggest decision will be simplicity vs aesthetics for the design, meaning choosing a design that is easy to build while being efficient versus being too complex but very aesthetically pleasing. Ideally, the design should be a mix between the two with the consideration of varying materials. Meaning if the design will be made by hand, milled, or even 3D-Printed. The simplest way would be by making it by hand, signifying that there will be a need for material that can be worked by hand. Such materials include wood, metal, and plastic in respect to the construction types. It is most likely going to be built by hand or done by a 3D-Printer. The issue with 3D-Printing is the availability due to the printer being of UCF and there is a queue for jobs being done. Also doing it with the 3D-Printer the use of softwares like autoCAD, autoDesk Maya/Mudbox, and Houdini. However, the team does not have as much experience with these softwares but it would be in the team's best interest to learn because of the advantages it brings. Advantages can include exact scaling for components and parts, also having less connection points making it have less pieces. The Design the team is deciding is the rectangular box structure with a dimensions of seven by five by five inches with a hollow cavity to hold the electronics in.

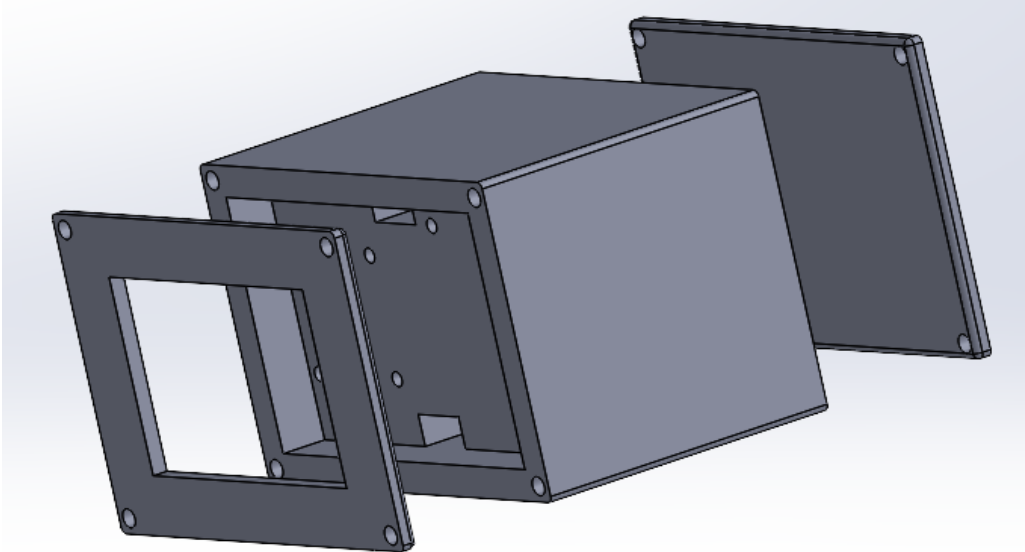


Fig 57. Iteration One of the Box Design

The dimensions might vary with the construction period, however, that is the current desired dimensions. The design will also need to have an open side to display the LCD screen that is going to be fitted inside the box. Once more is also desired to have a gap between the electronics inside to avoid contact with the PCB boards. A way to have gaps between the electronics is to have spacers placed, the size of the spacers will vary when the device will be built. The platform was updated in the build and was no longer going to be 3D-printed but rather be made of balsa wood for better adaptability on moving components. The platform within the Box frame will also be holding the microcontroller, the LCD screen, and space to set the ADC and pre-amp on. The frame will have to be tested for durability, electromagnetic interference, thermal venting, and comfortability. Each test will be specified by their own criteria and will be under the quality aspect of the device. The designed frame will also be under a stress test on the material to ensure the safety of the electronics inside, accidents can occur in the field of usage, meaning that it is highly desired to test the endurance of the device to ensure a capable product.

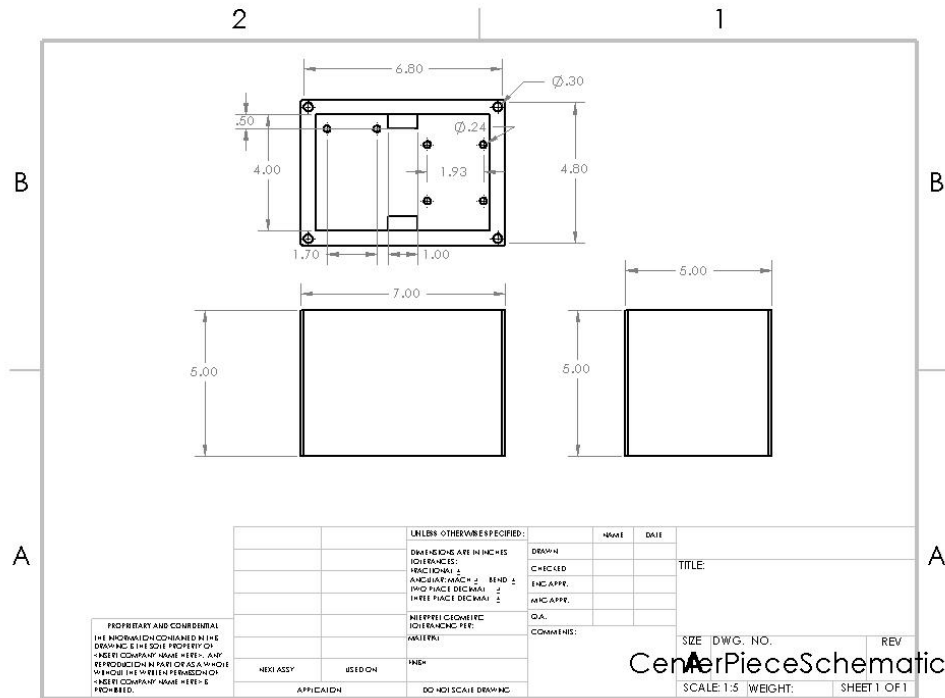


Fig 58. SolidWorks Design for the Box Design Centerpiece

In designing the frame, considerations had to be made to increase the accessibility of design for low cost and time. In that regard the frame will be made with a thickness of 0.5 inches for the walls of the frame. Reason being is that for the first iteration of this device it is meant to act as a prototype to test with. The approach in the design is to have three sections of the box: front section, center section, and rear section. Each section will be responsible for their own task of holding components and they must be compatible with each other. For the Center piece section of the frame, it will be where the microcontroller, DC-DC converter, Analog-to-Digital Converter, battery pack, and LCD screen will be fitted into. Within the cavity of the box there will be a platform that holds the microcontroller and separates the LCD screen and the other electronics, with holes along the sides of the platform to connect any necessary cables from the display to the microcontroller. The Front piece section is tasked to hold the LCD display in place on the front of the Center piece and will be attached by the four screws on the corners. While the Rear piece will be on the back side and have the battery pack attached to it, with velcro to avoid any chances of voltage leakage, like grounding the batteries. The Rear piece will also be connected to the centerpiece with screws. The thread and diameter of the screws will vary based on the quality of the print in the fabrication of the design in senior design 2. Though in Fig 59. Iteration One of the Box Design is what the first iteration of the design will look like.



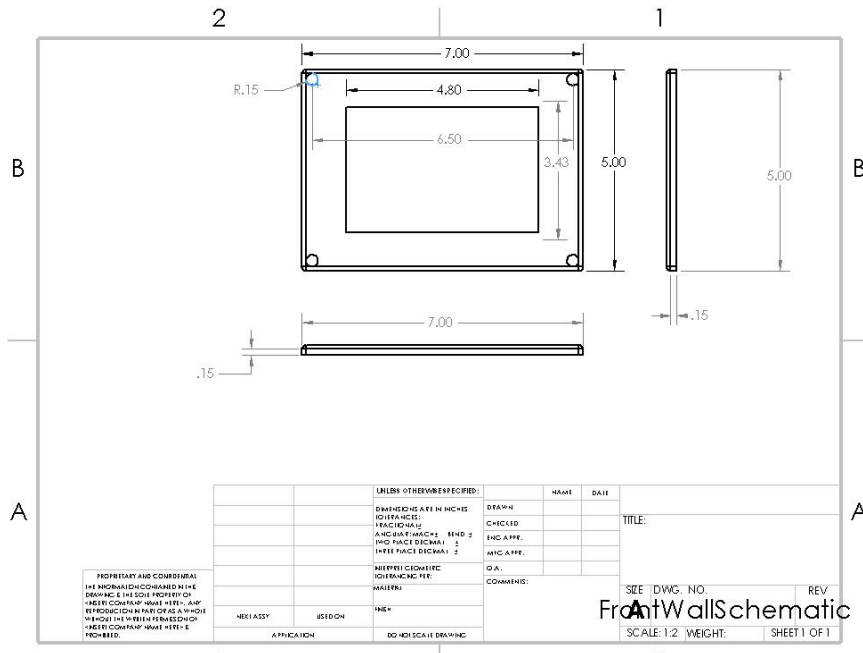


Fig 59. SolidWorks Design for the Box Design Front piece

The current iteration was modified in accordance to all the needs of the electrical components within the device. Meaning the box design had to be drilled through in certain areas to enable the connections of various ports on the circuit boards. The first design was not accounting the various connections needed to make the device operational, such as the inputs and outputs for the device. For time and efficiency reasons the project required the need for modifications on the existing design print, rather than making a new design and print from scratch. The modifications included the need for holes for xlr cables for input and outputs and the need for power cables.

While the next iterations will be more focused on quality and resource reduction. Meaning the consideration of reducing the thickness of the frame, like the walls, inner platforms, and covers to get a slimmer version. The software used to make the first iteration was done with SolidWorks 2021. A consideration that was also made was to have the edges rounded all around the design to avoid the chance of the edge getting hooked/stuck on another object. Since the device will be on hand it is ideal that the design is smooth and easy to grab, however, the current iteration is rather large; reason being to test the space of the box's cavity and make it easier to construct in the build phase of senior design. The next iteration will be done after the trial print to observe and note the quality of the three-dimensional (3D) printer that is going to be used. The next iteration will have more precise dimensions for the screws holes/threads, holes for the microphones on the side of the design's wall, better placement within the box for the components, thinning the thickness of the design, and any fixes for unexpected failures. Some unexpected failures can include cracks on the print, clumps on the

threaded holes, and loose components. The main objective of the frame is to be able to have all the device's input be able to connect to the microcontroller and have the microcontroller placed in a stable and secure location. The Rear piece will also have velcro on the exterior side to be the way to attach the device onto a user.

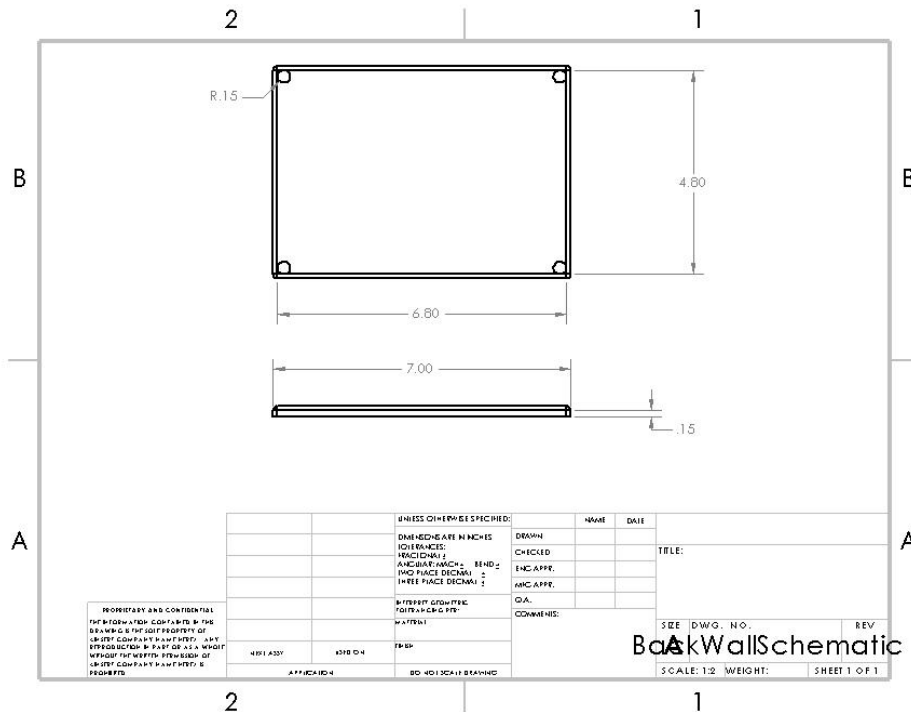


Fig 60. SolidWorks Design for the Box Design Rear piece

### 3.14 Pre-Amplifier

Through the need of raising a signal to match the Analog-to-Digital converter to the minimal signal threshold. The design of the pre-amp is to raise the input signal from a lower signal frequency to a much higher frequency through the gain of the Operational Amplifier. The project needed to have the microphone signal output to enhance the signal. Meaning that the gain of the operational amplifier had to be at least 15. In the design schematic reference derived from Texas Instrument, for non-inverting preamps, the gain was set to be in the range of 45 to 50. This depends on the resistive load being used.

## **4. Design Constraints & Standards**

In terms of constraints, the only major factors are time and money. Our sponsor is very easy-going about the project and has given us full creative freedom as long as the budget is within reason. Our main concern is time as we have hard deadlines to complete our research throughout the semester.

As far as standards are concerned, some of our components don't necessarily have industry standards, such as the LED strips, which required more research from our parts in order to find the most efficient parts to use for our project.

### **4.1 Constraints**

In regard to our sponsor, he has not put any constraints on our project. As long as we have a working product by the end of our Senior Design 2 term, he will be happy with the result. However, in regard to product and time constraints, we have observed a few hurdles.

First of all, the debate on how to attach the LEDs to the instruments was our main product constraint. We could drill holes into the instruments and attach the lights inside of the holes, but that would put us at risk of damaging the instruments. There was also the option of soldering the lights onto the instruments, however that also has a risk of affecting the vibrations emanating from them. For the first stage of project testing, we will simply attach the lights with glue or tape and after further testing, may decide to solder them on.

Next, we had the time constraint. Obviously, everyone has grand ideas for the project. However, with the constraint of only two semesters to create a working project, we had to be a little more realistic and scale down our goals for the final product. We initially wanted to include a touch screen display with various features programmed into it in order to allow maximum customization of the lights colors and brightness but we may not have the time to include them all so we will either need to minimize the customizable options for the lighting color and brightness or scale down the overall feature set to a simple LCD screen with only buttons for user input rather than a LCD touchscreen which may be more expensive to purchase overall and can add complexity in implementation and programming for the user interface so the LCD screen without touch inputs and with additional buttons would be more cost-effective overall than a touchscreen LCD.

Finally, the economic constraint. Although we were not given a strict budget for our project, our sponsor did not want to spend too much on parts as the price for everything in the current market is inflated. For example, the difference in price for the various microphones we considered. With our options we were either going to spend

\$20 or up to \$180, which is a very wide range. Although we would prefer to test each and every option, we will likely be required to settle for two options at most. Along the same path of the inflated prices, a Raspberry Pi 4 has a Manufacturer Suggested Retail Pricing (MSRP) of \$35. However, the cheapest one we could find on the market was priced at \$150, which is a price increase of over 400% from MSRP. This large inflation of pricing for the Raspberry Pi 4 served as a severe hindrance when trying to maximize the quality of the project while staying within a reasonable budget.

## **4.2 Standards**

Standards are established guidelines, characteristics, or processes that are used consistently between companies and countries. This allows products to be used together and allows consistency between products. It also helps design engineers when creating a component because it is known that there are certain criteria that have to be met for the product to be reliable. Standards are also essential to ensure a product is safe and simplify quality control as they already know certain ideas will work. Standards are set by accredited companies such as the Institute of Electrical and Electronics Engineers (IEEE) and then presented to the American National Standards Institute (ANSI) to be passed as an official standard.

### **4.2.1 USB Type-A**

USB, otherwise known as Universal Serial Bus, is an industry standard that certifies and establishes standards and specifications for protocols, connectors, and cables. The original purpose of the USB standard was to improve the manner in which computers are connected to external devices such as printers, accessories, monitors, and peripherals. The USB standard has been widely accepted since its introduction and can be found in nearly every consumer electronic device. The USB Type A connector was introduced in January 1996 and was considered to complement IEEE 1394 otherwise known as FireWire however, USB Type A quickly surpassed FireWire in popularity and industry wide acceptance and since then USB Type A has seen the most use in electronic devices.

#### Impact of USB Type-A on Design

The biggest impact of using USB Type-A with the project is sourcing a single board computer that supports the use of USB Type-A communications with data and power, not just USB Type-A that only supports device powering.

## 4.2.2 TRS

TRS (Tip, Ring, Sleeve) is an audio connection standard that enables analog audio signal connection from one device to another, usually input and output/processing. The TRS connector specifically is a ¼ inch connector with 2-3 contact points for either mono or stereo audio transmission. The TRS standard was first seen in 1877 on the first telephone switchboard, and since then has seen revisions for stereo audio support with 3 separate contact points and widespread acceptance in audio connection standards. Today the ¼ inch audio connector can be seen in use by guitarists, music producers, synthesizer designers, and even home audio consumers.

### Impact of TRS on design

One of the main impacts of using the TRS connector with the project is sourcing, or creating an Analog to Digital converter that converts analog audio to digital audio that supports the use of the TRS connector since many ADCs in production currently only have support for the smaller version of TRS known as the 3.5mm audio jack which ultimately can be utilized with TRS however, the TRS connector will need to be ran through an adapter connector to be compatible with the 3.5mm audio jack. This aspect of ADCs means that creating an ADC in house from scratch with our own defined specifications may be necessary to enable use with the TRS connector with a microphone to convert the signal from analog to digital so that the Raspberry Pi can interpret the audio signal effectively.

## 4.2.3 XLR

XLR is an audio connection standard primarily found in professional audio deployments that enables analog audio signal connection from one device to another with the ability to carry a balanced audio signal. The XLR connector is a circular 3 prong connector that features a locking clip to ensure a stable connection between the male and female connectors. The XLR connector that is commonly used today was first seen in 1955 and was invented by James H. Cannon and was originally called the Cannon connector and later named XLR after the polychloroprene rubber that was used in the female connectors. The XLR connection standard has seen many different configurations since its inception including variations with 4, 5, 6, or even 7 connection pins, not all of which are limited to audio use. The 3 and 4 pin XLR configurations are the most popular for audio use with the 3-pin connector which is found in almost every microphone on the market being able to transmit balanced mono audio and the 4 pin connector being able to transmit balanced stereo audio. Balanced audio signal includes a ground connection, positive terminal, and negative terminal; this additional connection can help mitigate signal loss and in a balanced audio setup can reduce noise interference from nearby electronics with high electromagnetic fields.

### **Impact of XLR on design**

The main impact of using the XLR connector with the project like the TRS connector is being able to source or create an Analog to Digital converter that natively supports the XLR connection standard. Many ADCs in production only support either the 3.5mm audio jack or RCA audio connectors. The XLR connector can be adapted to the RCA audio connector to be used with an ADC. However, it is not as easy to accomplish since we are taking a balanced audio signal and converting it to an unbalanced audio signal. If the XLR connector was utilized for an ADC made from scratch we would need to find a way to either interpret the balanced audio signal through the ADC to convert to digital or only utilize 2 of the 3 pins in the XLR connector to utilize a more standardized analog audio input signal.

### **4.2.4 USB Type-C (Power)**

USB, otherwise known as Universal Serial Bus, is an industry standard that certifies and establishes standards and specifications for protocols, connectors, and cables. The original purpose of the USB standard was to improve the manner in which computers are connected to external devices such as printers, accessories, monitors, and peripherals. The USB standard has been widely accepted since its introduction and can be found in nearly every consumer electronic device. The USB Type-C connector was introduced in August 11th, 2014 and was the first new full size USB connector since USB Type B which was introduced in 1996 along with USB Type A, other USB types had been introduced since then including USB Micro-A, Micro-B, Micro-AB, Mini-A, Mini-B, and Mini-AB all of which were cut down versions of the full size USB Type-A connector. The USB Type-C connector can now be found in nearly every consumer electronic device from smartphones to computers and even computer peripherals and accessories. With the USB Type-C connector being a newer USB standard, it is more difficult to obtain cables and adapters that meet the minimum specifications for data transfer through the connector however when only power is needed to be transferred over the connector the data specifications do not need to be up to spec to be operational.

### **Impact of USB Type-C on design**

One impact of utilizing the USB Type-C connector with the project is being able to source a single board computer that supports power input over USB Type-C since the connector is relatively new; many inexpensive computers still utilize outdated USB connection standards such as USB Micro B. Another impact of utilizing the USB Type-C connector with the project is sourcing a cable and power adapter that fully support 5V power delivery and are correctly certified for the USB Type-C cable specification. A recent problem with USB Type-C is underperforming cables which do not fully support the full capabilities of the connection and can ultimately hinder performance or power delivery if they are not certified correctly to the USB standards.

## **4.2.5 DSI Display Connector**

DSI, otherwise known as Display Serial Interface, is a connection specification by Mobile Industry Processor Interface (MIPI) that is used to connect displays or display controllers to mobile devices. The goal of the DSI connector is to minimize the overall cost of display connection while maximizing overall data bandwidth from the host to its destination. The DSI connector can be found on modern single board computers such as the Raspberry Pi and can be used to connect external displays to the computer without having to utilize the computer's other video output connectors. With the increased bandwidth of the DSI connector over other display connector technologies the DSI connector can carry input signals from the display such as touch input without having the need for an additional USB cable or data cable coming from the display back to the device.

### **Impact of DSI Display Connector on design**

The main impact of utilizing the DSI Display Connector with the project is the limited availability of external displays that support the connection standard. There are many small external screens currently on the market that support the DSI connector, however they are often small in size roughly 4-7" inches so if a larger display was desired the HDMI connector would have to be utilized. The upside to utilizing the DSI display connector is that due to its increased bandwidth over HDMI it can utilize a screen that has touch input functionality which will be crucial for user input in the project since the device itself is meant to be compact without any use for an input controller such as a mouse or keyboard.

## **4.2.6 MicroSD**

MicroSD, otherwise known as micro-Secure Digital, is a proprietary flash memory card standard that was developed by the SD Association otherwise known as SDA. The microSD card, since its inception on July 13, 2005 by the SDA, has been the industry standard in small form-factor storage for phones, cameras, laptops, and even game consoles such as the Nintendo Switch. The microSD card was predated by the SD card and miniSD card, the original SD card was initially intended to compete with Sony's Memory Stick which was originally intended to be used by music suppliers to combat music piracy. The miniSD card was intended to be used with early mobile phones and often came with an adapter to be utilized as a regular full size SD card. The microSD card in its highest storage rating of microSDXC (SD eXtended Capacity) is a whopping 2TB of storage which is comparable to full size desktop SSDs and can allow massive storage in small devices such as a Nintendo Switch.

### **Impact of the microSD card on design**

The main impact of utilizing a microSD card with the project is the inability to use onboard storage of the single board computer that is selected for use. The selected single board computer for the project, however, does not include onboard storage which means the use of a microSD card is required. The overall speed of the microSD card is not as fast as that of a full-fledged SSD however it is still fast enough to be responsive in use with an operating system and programs. The microSD card also has immense capacity for its overall size with up to 2TB of storage in a single microSD card which, while impressive, will not be necessary for the project to function effectively.

### **4.2.7 40 pin GPIO**

The 40 pin GPIO connector is a Raspberry Pi Specific connector that enables I/O connection between the Raspberry Pi and a multitude of peripheral input and output devices. The 40 pin GPIO connector was first used in the Raspberry Pi 1 Model A and enabled flexible communications to devices that could not communicate over standard USB data connections. The 40 pin GPIO connector enables the user to be able to actively prototype writing for their desired project without the need to solder wires directly to the Raspberry Pi and therefore increases efficiency in prototyping. In some cases the 40 pin GPIO connector can be used in conjunction with daughter boards such as audio cards and sensor modules which can simplify wiring due to the female 40 pin GPIO connector on the aforementioned daughter boards.

### **Impact of the 40 pin GPIO connector on design**

The main impact of using the 40 pin GPIO connector with the project is not being able to solder connections directly to the Raspberry Pi 4. This can be seen in some cases as a drawback but with the abundant prototyping that will be taking place in preparation for the final product connections that do not require soldering can be seen as an advantage due to the increased efficiency of debugging and being able to add, remove, or relocate connections to the Raspberry Pi with ease. The 40 pin GPIO connector is essential in the operation of the Raspberry Pi due to the fact that it is the only other way to communicate with external peripherals and devices other than USB which does not always have the flexibility to be able to communicate with such devices as sensor boards, or even TFT displays.

### **4.2.8 Programming Languages - Python**

Python is an object-oriented high-level programming language that supports a large volume of libraries to help aid programmers write code in a clear and logical manner. Python was first deployed on February 20th, 1991 and was designed by Guido van Rossum at Centrum Wiskunde & Informatica in the Netherlands and was intended to be a successor to the ABC programming language. Python is dynamically-typed



meaning the language interpreter assigns any variables at runtime based on the values contained in them in contrast to checking them during compilation which is what a Statically typed programming language like C++ does which helps improve its performance. Python is also garbage collected which means that the program that is running can do automatic memory management where any memory which is allocated to the program that is no longer referenced can be reclaimed which helps keep overall memory usage to a minimum. Python can be utilized on many different operating systems including MAC OS, Windows, Linux, and other Linux/UNIX based operating systems such as Raspberry Pi OS. The most recent release of Python is Python version 3.11.0 which was released on October 24, 2022. (MOre here)

### **Impact of Python on design**

The impact of using Python on the project is not being able to utilize proprietary functions of other programming languages inside python. If we were to want to implement a user interface with javascript in the program we would have to utilize java as our main programming language which other than user interface support would greatly limit how we would be able to interact with the Raspberry Pi. With the use of Python we can fully utilize every functional feature of the Raspberry Pi to its fullest potential since Python is directly built into Raspberry Pi OS. The use of Python on Raspberry Pi OS also opens the door to the massive community support of the Raspberry Pi, there are many forums, how to guides, and documentations that can help beginners to the Raspberry Pi to confidently and fully access all of the Raspberry Pis features. Due to Python's immense library support the process for audio processing integration should become more streamlined and more efficient due to the custom libraries that can help with audio interpretation, manipulation, and analysis. Python also has libraries for LEDs and addressable LEDs that makes the overall process of programming a lighting sequence to the LEDs much easier due to the custom libraries that are available. Raspberry Pi OS supports many Python based programs, one of which being eric, which is a Python based program that helps users design their own user interface for custom python programs, in the case of this project that feature will be utilized for the LCD touch screen display to help the end user select any theme, color space, mood, or mode that they want the device to utilize. With all of the different libraries, community support, and ease of use of the python programming language we can determine that the use of Python in the project will be integral to the overall operation and successful execution of the device.

### **4.2.9 LCD (Liquid Crystal Display)**

A LCD or liquid-crystal display is a flat panel display that utilizes properties of liquid crystals as well as polarizers to produce an image in color or monochrome. Liquid crystals by themselves do not produce light so to effectively view images displayed by a liquid crystal display a backlight is generally needed to provide luminance to the image to increase visibility. The LCD was first invented in 1964 by

George Heilmeyer when a high electrical field was applied across transparent planes that housed negative liquid crystals and in turn scattered light across the plane which made it appear white in color<sup>[40]</sup>. Since its inception the LCD has been an industry standard in display technologies for TVs, Smartphones, Tablets, and even handheld consoles. The LCD rose in popularity after the CRT era due to its high power efficiency over the CRT. Over the years the LCD has seen many iterations and improvements including IPS panels which helped to immensely improve color accuracy and viewing angles when using an LCD on a device. LCDs have recently fallen in popularity over the recent improvements in LED displays with technologies such as OLED displays which have immense contrast ratios due to the fact that OLED pixels are individually lit and can turn off when displaying shades of black giving it almost lifelike color accuracy. LCD displays are still useful due to their inexpensive cost and relatively good performance and will most likely still see use in consumer applications for years to come.

### **Impact of LCD on design**

The impact of utilizing a LCD display with the project is not being able to utilize more modern technologies such as OLED and QLED. These panel technologies have been immensely improved over the LCD and can produce brighter and more color accurate images than the LCD can. One of the advantages of using an LCD display in general with the project is having the ability to monitor and observe what the Raspberry Pi is doing at all times and to be able to pick and choose what the program does depending on the end user's preferences. The main advantage of using an LCD display however is the overall buy in cost, in contrast to OLED displays , LCD displays have been around for over 20 years and therefore can still be purchased at a fraction of the cost of an OLED equivalent and for the specific application that the LCD display will be used for in the project the advantages of OLED would be negligible towards the end user. Along with the aforementioned advantages the LCD display also has the ability to accept touch input along with display output while this technology is technically available in OLED displays there do not seem to be many if at all OLED touchscreen displays that are compatible with the Raspberry Pi 4, so naturally in the use case of the project the LCD display seems to be the best and most affordable option in displaying information to our end user and improving overall usability and interactiveness with the end user.

### **4.2.10 5V USA power supply**

The 5V 2 prong unpolarized power supply used for the project utilizes many industry and national standards. The 5V power supply complies with the USA Non-polarized NEMA 1-15 2 pin 10 Amp standard for two prong power connectors and complies with the NEMA and IEC standards along with certifications from UL. NEMA also known as National Electrical Manufacturers Association is an ANSI-accredited organization that develops standards for electrical components in the USA, NEMA is

made up of business leaders, electrical experts, engineers, scientists, and technicians and is responsible for the majority of standards for electrical components in the USA. IEC also known as the International Electrotechnical Commission is the organization that is responsible for publishing international standards for all electrical, electronic, and related technologies. The 5V power supply has been certified by UL, UL also known as Underwriters' Laboratories is a global safety science company that specializes in product certification for safety and leaves their certification seal on the majority of consumer electronic devices. With these certifications behind it the 5V power supply seems to be up to the task of powering the Raspberry Pi 4 safely and efficiently without risk of electrical or safety issues.

### **Impacts of the 5V USA Power Supply on design**

One of the main impacts of using the 5V Power Supply with the project is having the peace of mind that we will have safe power delivery to our Raspberry Pi 4. Many inexpensive power supplies that do not feature any industry or national certifications can often be a hazard when used. In one example a laptop power supply by the company T power was sold through national retailers and did not feature any electrical certification, in many reported cases from users this power supply would often catch fire and melt when being used. This massive oversight by T power could have cost the lives of many consumers if they had not discovered the melting power supply before it burst into flames. Often these power supplies can be outed from certified power supplies by looking on the information label and seeing if it shows the UL solutions logo, often times if this is not present the power supply itself will not be up to the standards set by UL solutions and in extreme cases can end in situations similar to that of the T power power supply. With choosing a power supply that meets many different certifications and standards we can be 100% sure that we will not have any safety concerns with power delivery for the Raspberry Pi 4 and we can be confident in the efficiency of the power supply when it is used in conjunction with the Raspberry Pi 4 and the many add on peripherals like the LED strip, Microphone, LCD display, and ADC that need to be powered through this power supply.

## 5. Testing

Once the research for the various parts and programs required for the product was complete, our group decided on 1-2 of each component to move on to the testing phase with. We will test these various components and compare the results of each one with its “competitor” in order to decide on which components we will be using in our final product.

In regards to ordering the parts necessary to begin our testing phase, we had to begin communications with our sponsor. This proved difficult as our sponsor had a very busy schedule at the time and was unable to immediately order the parts we needed. However, through our own connections, we were able to obtain most of the core components of our project through friends and family that allowed us to borrow what we needed until our sponsor gave us the desired parts we ordered. This delayed some aspects of our testing by a couple weeks but we were at least able to begin testing with some lower quality products to begin with.

Our first goal with this phase was to test the lights and ensure that they were working properly and compatible with the Raspberry Pi, as well as being able to change the color and brightness of the lights through python. Then we will connect the microphone to the Raspberry Pi to ensure that we can analyze audio levels and pitch through python, after that we will begin to work on a program that will change the colors depending on the pitch/note. Once we have a working program that satisfies all of these requirements, we will begin testing with instruments and LCD displays which will also require testing the power supplies and printed

### 5.1 Microphones

When choosing a microphone to use we primarily chose to use a clip-on cardioid microphone like the YPA M613-XLR however we chose to test it with another microphone to ensure its capabilities the two different microphones that we tested were the YPA M613-XLR Clip-on condenser microphone from YPA microphones, and the Behringer XM1800 handheld cardioid microphone. We did not test two Clip-on microphones due to the immense cost to purchase one microphone by itself which limited our options for testing since our sponsor preferred to minimize overall expenses in the purchase of all of the products. The Behringer XM1800 was provided by a relative of one of our team members to assist with testing to have a comparison point to ensure that the YPA M613-XLR would be up to the task of recording instruments from a close distance. Initially when we started testing the microphones we ran into issues with the pyaudio and audioop libraries that hindered our ability to run the program to check the microphones audio level input however, after some research we were able to find a solution to our problem which involved installing a different version of python on the Raspberry Pi to ensure compatibility with pyaudio and audioop

To start off testing we utilized the Behringer XM1800 microphone and positioned it 4 inches away from the desired recording instrument perpendicular to the instrument's surface, in this case we utilized a snare drum to record audio from the microphone. To store the recorded microphone data we utilized the widely utilized audio recording program known as Audacity and analyzed the sound that was recorded from the microphone to check for peaking and audio artifacts. After we made sure that the microphone functioned as intended we plugged it directly into the Raspberry Pi via USB and ran our sample audio level reading program to ensure that the microphone would be compatible over USB with the audio python libraries that we intended to use. When the microphone was plugged in through USB it was immediately recognized by Raspberry Pi OS and worked flawlessly with our python audio level program. This result gave us much needed confidence in our approach to microphone implementation for this project and proved that this concept would be able to perfectly function with our intended program plan and project goals.

After we finished testing with the Behringer XM1800 we moved on to testing with the YPA M613-XLR to compare and contrast the audio quality in relation to the XM1800. We used the same format of testing with the YPA M613-XLR with the microphone being positioned 4 inches away from the instrument and placed perpendicular to the instrument's surface and in this case the microphone itself attached to the instrument which took out some guess work with exact placement of the microphone. Like the Behringer XM1800, the YPA M613-XLR functioned as intended after it was plugged in directly into the Raspberry Pi via USB and was run through our sample audio level reading program to ensure that it would be compatible over USB with the audio python libraries that we intended to use. When we plugged in the YPA M613-XLR through USB it was immediately recognized by Raspberry Pi OS and worked flawlessly with our python audio level program. This result gave us much needed confidence in our approach to microphone implementation for this project and proved the Clip-On Cardioid microphone that we chose to use was a viable option out of the 3 different microphones that we initially considered to use in the project.

Our first order of operation was simply to test audio amplitude with a simple python program that would retrieve the current audio level on a scale from 0-32768 in 0.1 second intervals for fast response. After we discovered that this program responded predictably to changes in audio volume we were confident in the ability to utilize volume level with the brightness of the LEDs and were able to tie in this functionality to a sample program that would brighten and dim the LEDs in response to audio levels albeit only in one color and at a slow refresh rate.

## 5.2 LED Strips

In terms of light strips, it will be very likely that we will be using a SMD light strip for the simple reason of it having the ability to change each light's color individually. For the testing phase, we will be testing the various programming methods to apply to the lights and have them display the correct colors. We will also be testing different levels of brightness and if it would be best to have all the LEDs light up at once or section by section.



Fig 61. BTF-Lighting 5050SMD

Our first product to test was the WS2812B RGB 5050SMD light strip from BTF-Lighting. For strictly testing purposes, we decided to order a shorter strip of length 3.3 feet before committing to a longer 16.4 foot strip. Once the strip arrived, we connected it to the Raspberry Pi 4 and began writing the code to control the lights. We utilized the Neopixel LED Library in Python to ensure compatibility with the lights.

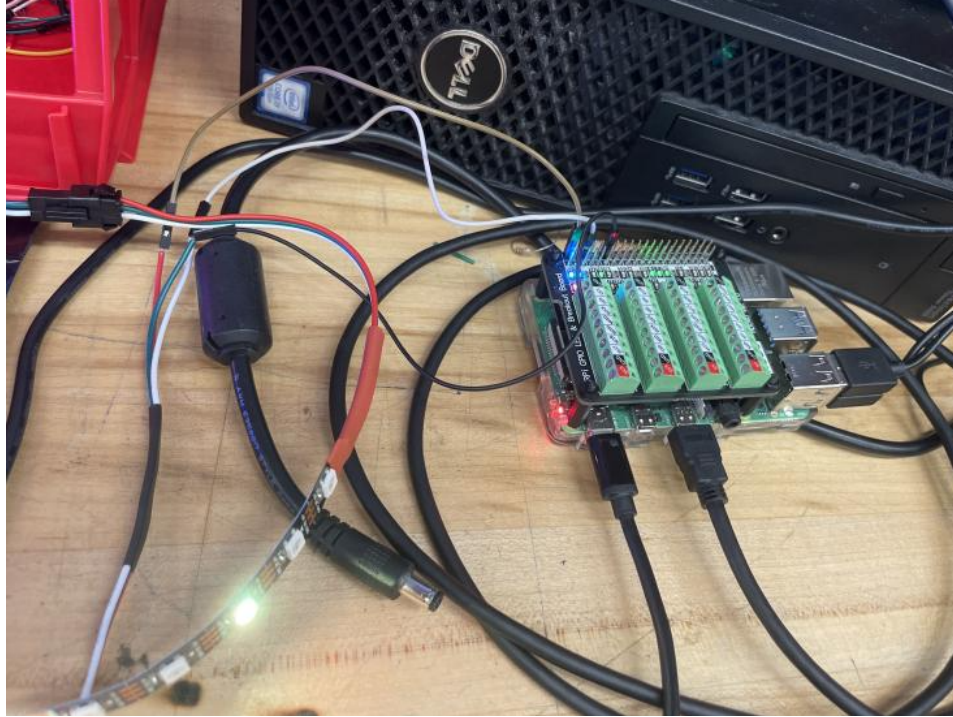


Fig 62: Lights Connected to the Raspberry Pi 4

Once the code was written, we ran our first test and were met with multiple syntax errors relating to package permissions, mainly the Neopixel python library that we had to install onto the initial installation of python on the Raspberry Pi; however, after a little research into the very specific problem that we were experiencing we were able to find a solution to our problem on Stack Overflow, we were able to resolve the package issues by running the python program with the sudo command inside the command window, the sudo command allows any command following it to use the highest level of administrative privileges when executed which is exactly what the Neopixel library needed to be functional in the program. Once the problem was resolved we were able to manipulate the lights however we pleased. We began by setting the lights to a single color in this case white to ensure that all LEDs were working congruently. If one individual red, green, or blue LED was broken or not functioning properly it would be easily spottable when displaying white since white is all colors combined and any imperfect LED would be discolored and have either a orange, purple, or cyan hue to it depending on the color of the LED that was malfunctioning. We also ran this test to determine if the brightness of the LEDs were adequate for our application and to make sure the adjustable brightness feature in the code was working properly.





Fig 63: Light testing

Once we were confident that the code was working, we began to experiment with the different color and addressable LED options to gain some confidence in the ability to change the colors of the lights. The first thing we tried was having the lights continuously change colors and brightness to check the color accuracy and overall luminance of the LED strip. Then we set each individual LED on the light strip to a different color to ensure that the addressable functionality of the LED light strip was usable and straightforward to implement in python without much hassle.



Fig 64: Light colors



The results of these tests gave us confidence in the brightness and color abilities of LED light strips. After testing multiple different brightness levels ranging from 0-255 we determined that the brightness of the LED light strip was more than sufficient for the application of the project and ensured confidence in our decision to utilize the specific LED light strip that we chose.

Although the quality of the individual LEDs was satisfactory, we felt the space between each LED was too large so we decided to test a light strip where the lights were more densely populated and compared it with the original strip.

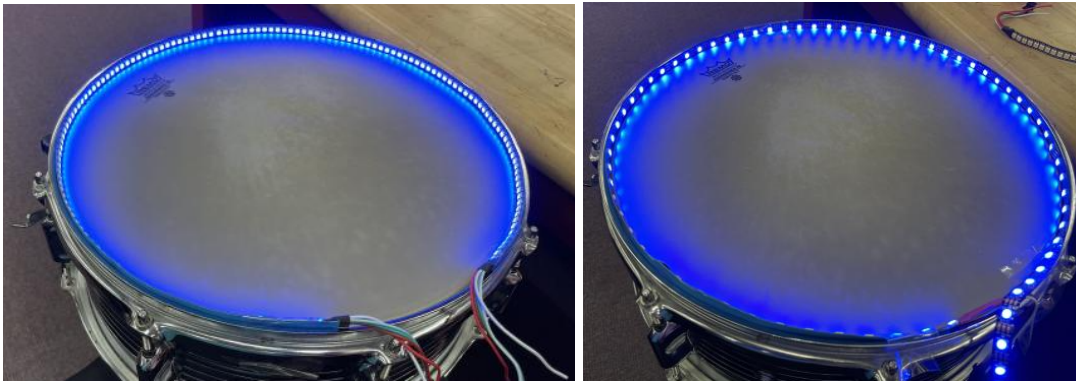


Fig 65. Dense vs. Regular spaced LEDs

Seeing these two light strips side-by-side, it was clear that the more densely populated strip would give off a much cleaner visual than our original strip. With this, we decided to use it for our first prototypes and no further testing was required.

### **5.3 Power Supplies**

When choosing a power supply to increase stability and to limit complications and set up. We decided to go with the 5V wall charger. The wall charger can be plugged directly into the wall and into the Raspberry Pi which would make it fast, reliable, and easy to set up. The wall charger included a switch to be able to turn on and off the Raspberry Pi 4's power to protect it. This is the major component to test with using the wall charger.

When testing the power supply we ensured that the power supply, when plugged in, would not power on the Raspberry Pi until the switch was flipped. We also tested to make sure that after the switch was turned off, the Raspberry Pi would not remain powered. This ensured the component was working as intended. Once we ensured that the power supply would be sufficient for our project, no further testing was required.

## 5.4 Single Board Computers

When choosing a single board computer to use we primarily chose the Raspberry Pi 4 for use with the project however, we chose to test the Raspberry Pi 4 along with the Raspberry Pi 3 to ensure that the added capabilities of the Raspberry Pi 4 were worth the price difference over the Raspberry Pi 3. Like with the microphone testing we did not purchase both the Raspberry Pi 4 and 3, our sponsor purchased the Raspberry Pi 4 and we were able to borrow a Raspberry Pi 3 from a relative of our teammate for comparison with the Raspberry Pi 4.



Fig 66. Raspberry Pi 4

To start testing our single board computers we first used the Raspberry Pi 4. The first step was the setup. Once the Raspberry Pi OS was programmed into it, we had to select the appropriate settings before we could begin using programs such as the internet browser to download whatever other programs we needed such as Python 3. Initial setup of the Raspberry Pi 4 introduced some unlikely challenges due to the fact that the time in Raspberry Pi OS was not syncing correctly when the Raspberry Pi 4 was connected to the internet, this differentiation in time disabled our ability to properly browse the web to install additional programs like Python 3. After we figured out how to configure the time and date through the Raspberry Pi OS's console, we were successfully able to browse the internet and install the latest version of Python 3 through the console. Setup proved difficult as we faced an onslaught of syntax errors when first testing the Neopixel libraries. The installation of the Neopixel libraries were straightforward with only 3 commands being needed to install all of the proper libraries and datapaths. However, when we went to test our first program we ran into some syntax errors with the python 3 IDLE as many function calls in python were flagged as not having the proper permissions to execute; so in order to work around this issue we resorted to executing the python programs inside Raspberry Pi OS's console by

preceding the command with the sudo tag to ensure proper high level access to any resources needed for the program. After the program permissions were sorted the python program ran as intended and only needed minor adjustments to code segments in order to achieve the desired test lighting effects.

## 5.5 Display Technologies

Our product will use the LCD screen with touchscreen capabilities that is also compatible with the Raspberry Pi Board. What will be tested on the LCD is the brightness and contrast to ensure that the User Interface (UI) is properly displayed and can be seen in bright and dark environments. The LCD screen should also be capable of changing the contrast based on the user's demand. Since the LCD screen will be touch screen the screen's sensors have to be checked and ensured that it is capable of reading the user's inputs. Ideally having the LCD screen go through a stress test would help understand the material strength, however, the project's budget is not limitless and it is in higher priority to keep the LCD screen in perfect condition. Meaning the avoidance of the material strength testing and vibration testing is needed to ensure the safety of the LCD screen. However, the LCD screen can be tested in regards to thermal testing and electromagnetic fields testing, ensuring that the LCD screen is not an outlier that will cause complications to the microcontroller and microphone.

## 5.6 Printed Circuit Boards

The printed circuit boards were designed using EasyEDA as that was what was used for JLCPCBs, the website that the PCBs were ordered from. The ADC PCB worked just as well as the tested breadboarded system. However, the Pre-amp didn't work at all. This was probably due to an overlooked error when setting the traces or grounds.

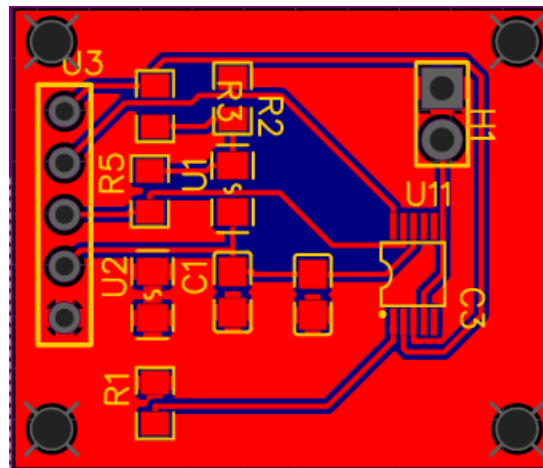


Fig 67. ADS1115 PCB

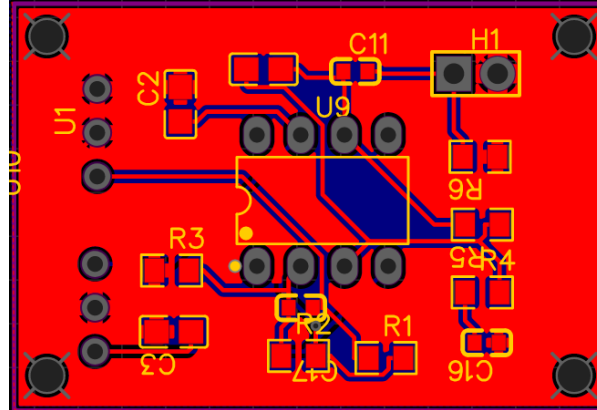


Fig 68. Pre-amp PCB

## 5.7 Instruments

Our product will be used in anything from solo recitals to orchestras. For this reason, we need to test it with both a single instrument and multiple instruments being played at the same time. The instruments we will be testing with are marching band drums, a guitar, and a trumpet.

We began by testing solely with the marching band drums, specifically a snare drum. The snare drum is only capable of producing a singular pitch due to the fact that it only has one striking surface. The marching snare drum has the capability to produce very soft and extremely loud sounds depending on the type of down stroke that the player is using to strike the head of the drum. The immense dynamic range capabilities of the snare drum may cause complications in audio recognition with the microphone due to clipping with the immense volume of a snare drum if the user plays a Glock Shot which is the loudest stroke that can be used on a snare drum. In a situation like this we may need to utilize a compressor or filter to try to control the overall sensitivity of the microphone so that the audio signal does not peak and potentially cause issues in the volume recognition part of the program.

The utilization of a marching snare drum with the device is relatively simple due to the instrument's rim that holds on the batter head; this rim can be used to clip the microphone on to for easy securing. With an unpitched instrument such as this we can fully explore volume based reactions and create a color sequence that corresponds with the type of music being played. The LED light strip on the marching snare drum was placed along the perimeter of the drum shell right below the top rim, with another loop above the bottom rim. This layout gave the snare drum two rings of light around it to produce the most amount of light possible. If we were to utilize a marching bass drum with translucent batter heads the LED light strips could be placed inside the inner

shell of the bass drum so that the light could shine outwards and would light up the batter heads large surface by the LED light strip.



Fig 70. Temporary Sourced Snare Drum

After we tested the device with a marching snare drum we then shifted our focus to an instrument that can emit different pitches, specifically a trumpet. Unlike the marching snare drum the trumpet can produce multiple pitches at varying volumes. The trumpet is a brass wind instrument and can typically be found in 3 valve configurations and the approximate range of a Bb trumpet is from F# in octave 3 to E in octave 6 (F#3-E6) so when we utilize this instrument with the device there are roughly 34 notes that can be interpreted and visualized accordingly.

Trumpets feature a bell that emits the sound emitted from the player buzzing into the mouthpiece, this bell on the end of the trumpet is a perfect surface to be able to mount our clip-on microphone due to the spring loaded clip. In relation to marching snare drums, trumpets are also capable of high dynamic range however on the high end they can not emit as intense of a volume as a marching snare drum. The trumpets decreased volume ceiling in relation to a marching snare drum means that no compressors or filters will be needed in conjunction with the microphone. Another difference of the trumpet over the marching snare drum is its mobility, the marching



snare drum can be mobilized with the use of a drum harness however in a concert scenario it will most likely be stationary on a basket stand.

The trumpet is a small brass instrument and is held in the hands of the performer which enables the performer increased mobility. This mobility may cause some implications with the portability of the device due to its need for an external power supply, one solution to clutter with an instrument such as this may be clips to route the microphones cable to a more stationary area of the performer such as their waist, this relocation will reduce frustration for the performer because of reduced interference of mobility with long unsecured cables. The LED light strip was attached around the bell of the instrument and emitted light in a ring round the bell, the excess LED light strip was placed in a spiral pattern down the bell of the instrument up to the valves to have more surface area to produce light.



Fig 71. Temporary Sourced Trumpet

The last instrument that we tested was an acoustic guitar. In contrast to the trumpet the acoustic guitar does not require air from the player for it to emit sound and it only requires the player to pluck the strings in combination with different fingerings on the frets of the guitar to produce different pitches. The acoustic guitar produces sound by vibrations of strings that are tuned precisely to a certain pitch, the acoustic guitar also has a sound hole in the middle of its body to serve as passive amplification to increase the volume of the sound produced by the strings.

A standard 6-String acoustic guitar can produce a large range of pitches from E in the second octave to Bb in the 5th octave (E2-Bb5). With this range of notes we can expect roughly 42 different notes that can be interpreted and visualized by the device. In terms of volume; acoustic guitars can not produce a high dynamic range similar to a trumpet or snare drum and this is partially due to the limitations of passive amplification inside the guitar. With this reduced dynamic range we can be confident that the microphone will have little to no problem picking up the volume and pitch of every note emitted from the acoustic guitar without clipping.

To mount the microphone we will have to create or purchase an additional mounting adapter for our clip on microphone since there is no small surface near the sound hole of the acoustic guitar for the clip to hold on to. Many guitar clips for microphones feature a vice-like system that clamps over the whole body of the guitar, something similar to this could be utilized to mount the microphone to the guitar. In terms of mounting the LED light strip to the guitar, the light strips can be mounted around the perimeter of the body of the guitar to shine outwards away from the performer to produce a bright and intense glow to the surrounding environment of the acoustic guitar.

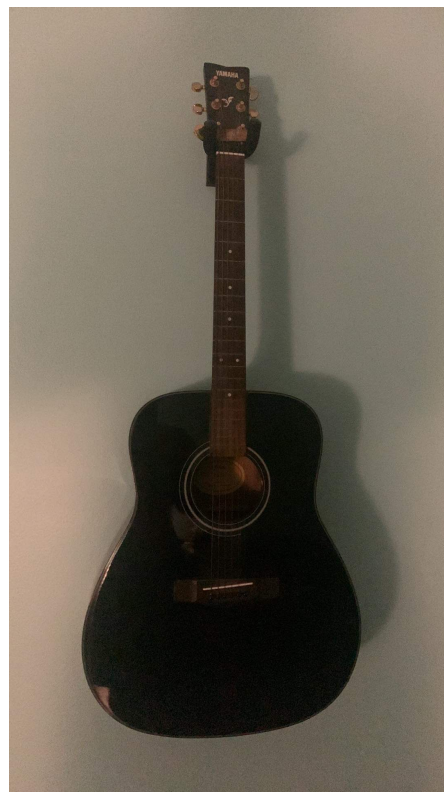


Fig 72. Temporary Sourced Acoustic Guitar

## 5.8 Vibration Testing

Vibration testing is putting the device through kinetic motion to identify any components that lose its functionality within the device. This form of testing was sought after to ensure the quality of products through rough conditions. Vibration testing is recommended for this project due to the device being portable and used on hand, meaning that the device will consistently be under stress of motion. If not tested,

the device may experience unexpected errors during the demonstration. To avoid errors the device will be undergoing custom vibration tests due to the budget not supporting a third-party source to do their professional tests on the device. A common test that is done is attaching the device on a platform that vibrates from a motor or servo, and then record all the errors that occur on the device on or off. The project will use that form of vibration test as well but not as complex. The device will undergo three tests, one being a basic motion test, second being the vibrating platform test, and the third is a drop test. What these tests will be doing is to ensure that the device can withstand the various forces from the low end of the spectrum to the extremities that can be experienced in the field of use. The forces will have an impact on the internal integrity of the device along with the structural design, like whether or not the plastic will crack or if a screw will fall out causing a chain reaction that could cause critical failure of the device. It is not ideal for the internal electrical components to touch one another due to the risk of shorts and physical damage to the pcb boards.

The motion test will consist of the device being attached to a system at motion, such as a student carrying the device around and moving or shaking the device around in the tester's hands. This is not a professional test, however, it does act as a first step to see if anything is loose, not properly installed, and or damaged. Through careful inspection it can be said it is functional and proceed to the next test, which is the vibrating-platform test. The vibrating-platform is more in line with the standardized vibration test that is done on devices. The way it will be achieved for this project is by attaching the device to a platform like a wooden table, creating one system, and having the table attached with an electric motor with a rotor that hits the platform causing the platform to oscillate. This oscillation will be more intense than the previous test due to an increase of kinetic energy that is placed on the device, which can stress the device and uncover any form of loose connections on the device. In Fig 73. Vibrating Table shows the ideal testing equipment for this form of testing, it generates vertical kinetic energy onto the device.



**MP-3000E2**

**Series vibration tester**

**operating instruction**



Fig 73. Vibrating Table

The third test consists of a drop test which is a strong kinetic force that is being applied on the device at the moment of contact with the ground. For the safety of the device this test will either be skipped or done with safety measures at the ground. The safety measure can be some sort of cushion or net that catches the device prior to making contact with the ground. Even though the test is not as risky it still can still be used to observe and identify any problems as the device is set in free fall. Also applying a spin while in free fall will cause the device to undergo centrifugal forces, and this could also help identify any issues within the device. Once all vibration testing is done an updated design iteration would be made to improve the previous design. A possible design flaw that can be predicted is the use of velcro strips to hold the battery pack inside the device and the velcro strip that will be used to hold the device on to the user. The concern being the unhooking of the velcro strip and having the battery pack loose inside and the risk of having the entire device fall off of the user. A solution for future iterations would be having an internal compartment that will house the battery pack, along with having a clip attached to the exterior wall of the device. Another solution is to have another external compartment attached to the user that can house the entire device. These solutions should reduce total vibration the system might experience but the risk of components getting loose is still a major concern. Through continuous testing on every iteration of the design will be the proof needed to improve the manufacturing quality of the device, meaning that this test will be directly corresponding to the physical design of the device.

The device did undergo a Vibration Test, it was dropped at a height of 3 feet with all the components inside attached securely. The test was successful and no noticeable damages were noted and the components remained connected and

secured inside, hence the passing of the test. This was a concern of ours and only one test was done due to not wanting the risk of permanently damaging our only frame for the device. This test was conducted within the senior design lab with carpet flooring.

## **5.9 Thermal Testing**

This form of testing is desired to test the thermal capabilities of the device and understand why the device can fail with the shift of temperature. Two forms of testing can be done to accomplish this, one is the adjustment of temperature within the board to see if functionality of each component is still viable, and the second is the assessment of the components while the device is active to see where the device is generating most thermal energy. The generation of heat on the board is a sign of bad efficiency meaning that the component is not properly compatible for the application being done. The heat from one of these spots could also affect other parts on the board which can cause failure for the device. Reason being that every component has different thermal thresholds before a shift in desired response or even failure. In respect to most components the ideal temperature for them would lie in the range between 313.15 Kelvin and 323.15 Kelvin.

The device can undergo critical failure if it reaches temperatures that are too high or too low. In the case of too high, components like capacitors have to avoid temperatures that reach 398.15 Kelvin to 423.15 Kelvin while active; if not they would become damaged. Also having temperatures higher than 463.15 Kelvin will have catastrophic failure on the device due to the solder material on the microcontroller board cannot withstand those temperatures without the solder reaching melting point which will cause components to slide off the board. While in the case of low temperature the device will behave quite differently, it affects mainly LCD screen displays, making their performance relatively slower. This is due to the liquid-crystal in the display becoming more viscous as the temperature decreases, the threshold temperature is 273.15 Kelvin. The LCD functionality gets worse as the temperature approaches the freezing point of the crystal used. However, this is not that common to occur especially for the location and application this specific device will be used. In the case of higher temperature, there is a possibility of having the device reaching very high temperatures, either by leaving the device outside in the sun on a really hot day or by placing the device near an object that is radiating heat, such as a generator.



Fig 74. Heat Damage on LCD Displays

For this project it would not be possible to send to a company that can test the thermal capabilities of the device in their workshop, this is due to time and budget for the project. In replace to that the device will be undergoing custom tests to ensure quality of the product. The tests that will be done are putting the device in a low and high temperature and checking the functionality of the device while active to ensure no hot spots on the device. The test for the low temperature is by putting the device in a freezer for a set amount of time like thirty minutes and checking the functionality of the device after being set to 255.15 Kelvin. There is a risk for permanent damage for the LCD screen, so testing the LCD screen will not be needed due to this risk. For the high temperature test, setting the device on a platform and heating the device with a heat gun would be a functional test; though for similar reasons with freezing test the device will run the risk of permanent damage and to avoid unnecessary damages this type of test will be avoided.

The other thermal test would be having the device on and checking for hot spots. The way to check is by having an Infrared thermometer, which can check the temperature from a distance. Another way is to have a heat sensor connected to the spots that the designer wants to have tested. In the case for this project it is more practical to have a hand held Infrared thermometer, as displayed in Fig 75. Infrared Thermometer, to measure the microcontroller board and identify the hottest spots.



Fig 75. Infrared Thermometer

The device is not at any high risk of temperature issues due to the environment prior and during the demonstration will be controlled. Meaning the temperature will consistently be that of room temperature which is 293.15 Kelvin and will be mainly used indoors. The device will also be attached to the student on the demonstration so it will be relatively on hand always. Though it does not mean that the device is safe in relation to itself, meaning that the device could possibly damage itself based on the device's structural design. Currently in the box design first iteration there are no vents that enable air flow through the device which can pose issues to the device's ideal conditions of operation. In the future iterations of the device's frame, vents will have to be implemented to avoid overheating within the cavity of the device. After testing it will also determine if additional cooling methods will be needed like having heat sinks, fans, or liquid cooling will need to be implemented. As it stands now there is no need for additional cooling methods and the implementation of vents will be the only needed addition to the design.

No noticeable changes occurred during the operational time of the device hence, this test was not conducted due to the lack of time. There were no signs of heat spots on the device, especially during long periods of use.

## 5.10 Electromagnetic Testing

Electromagnetic testing is needed in this project because this device utilizes electromagnetic frequencies that have to be filtered into the microcontroller and then processed. Having internal noise can cause a distortion in the electromagnetic reading and if not properly filtered out will cause the microcontroller to pick up wrong information. This practice is critical for very sensitive devices that can not adjust with

filters. In this project's case the microphone is sending the frequencies that are being picked up from the instruments being played, the point of interest is when the frequency is being translated to analog within the device. If the delta sigma analog to digital converter (ADC) has too much frequency noise around it, it will cause distortion and produce a non-desired response from the microcontroller. This brings the need for the testing in which insist the use of a spectrum analyzer with a near-field probe and at the front end a low noise preamplifier installed. In Fig 76. Spectrum Analyzer is the testing equipment that can be used to identify any phase noise that is being generated by the device.



Fig 76. Spectrum Analyzer

What can produce phase noise on a board is oscillators, waveform generators, and other signal sources. Types of oscillators can vary but typically are associated with crystals that are embedded in the circuits (like a 32 KHz quartz crystal) that are used as timers. Identifying the spots in which there is noise will help in designing a circuit board that does not have as much phase noise to it. In design there are five ways to reduce a board's noise. Starting by adding ground and power planes to the board, where one layer is reserved specifically for ground and another layer is for the power planes. Also having a well defined path for the areas with high-frequency signals without the use of interruptions or vias. The second is to optimize trace sizes, making sure the traces are as short and as thin as possible within the board. The third is to separate zones of the PCB so that the analog, digital, and high-frequency areas are all separate from one another. This helps simplify the board and helps the noise interference between the sections, as demonstrated in Fig 77. PCB Zoning.

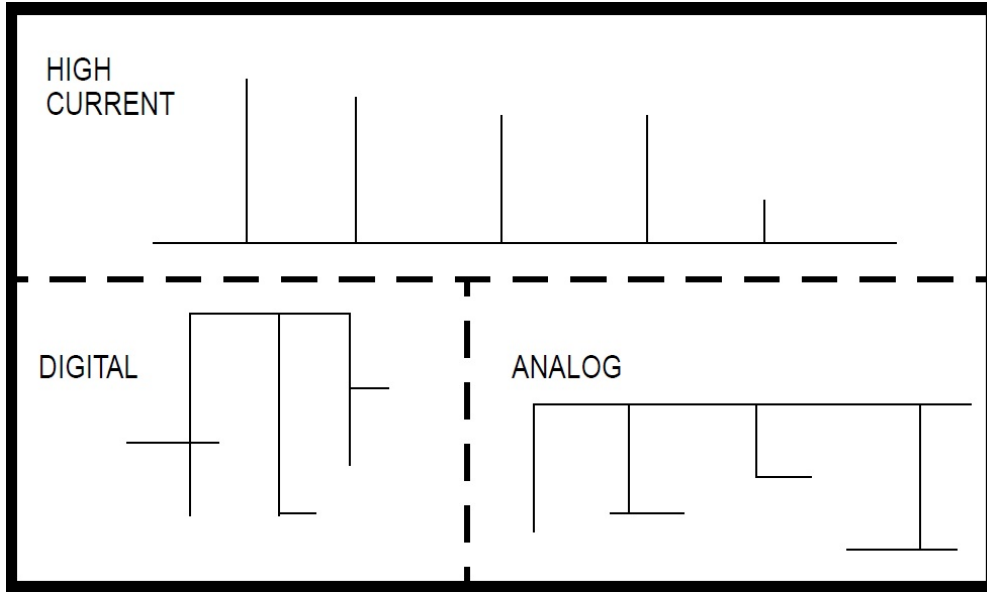


Fig 77. PCB Zoning

The fourth is decoupling capacitors, which is the placement of these decoupling capacitors near the power pin of active components, to reduce signal switching and signal bouncing back to ground. The fifth is trace routing to avoid the risk of coupling, to do so having two layers one being for horizontal traces and the other being for vertical traces will help. Though another consideration is making sure that the oscillators (crystals) in the circuit have their own layer to help avoid coupling and to stay away from sensitive signal carrying traces. In the current iteration of the device there should not be much interference due to the way the electrical components are structured. However, that does not stop outside frequency from entering the device, a possible solution can be having a frame with an internal or external shielding that can prevent outside frequencies from entering. The shielding can be done with any ferromagnetic metals containing traces of: iron, nickel, or cobalt. The cost effective metals would be like aluminum or steel, the thickness does not need to be excessive for the principles of the skin depth effects. This solution will be taken more in consideration after the electromagnetic testing that will be done on the device, but currently the device does not need this additional protection.

No noticeable changes occurred during the operational time of the device hence, this test was not conducted due to the lack of time.

## 5.11 Material Strength Testing

Material Strength Testing is one of the more vital practices in the control and quality aspect of research due to this test being the most physically demanding on the device. Meaning that this practice tests the device's integrity and stress threshold before reaching a point of failure. In the case of this device what will be mainly tested is

the frame of the device to ensure that the electronics are housed in a secure and structured frame. If the frame's material does not pass the test another material must be taken into consideration for the quality and safety of the device. The standardized test is for the device to undergo a constant force that causes the device to undergo structural stress to uncover any failures within the structure. A common case is having a hydraulic press that puts pressure on the device until it crushes the tested material. In this project's case a hydraulic press would be hard to acquire and use to test the frame, so as a replacement test would be putting weights on top of the device to see how much weight the frame can hold before cracking and disfigurement occurs. As shown in Fig 78. Hydraulic Press, is the equipment that is used to test the material strength of objects.



Fig 78. Hydraulic Press

Through careful assessment on the frame it can be deducted how much pressure the frame of the device can withstand before failure. The frame's material that is going to be tested can vary due to there being various options in the design portion of the project. The materials to consider are steel, acrylonitrile butadiene styrene (ABS), or polylactic acid (PLA). The strongest on paper of the three options is steel, then PLA, and finally ABS based on their material makeup. The design of the frame will not be changing, only the material will, making the test of the material viable. On paper the structural strength of steel is 100,000 pounds per square inch, for ABS it is 209 pounds per square inch, and PLA is 119 per square inch.

For the application the most viable to use is the ABS since the likelihood of the device to experience high forces is low. The main risk is setting the device on the ground and an object gets on top of the device and exert an excessive amount of pressure and force onto the device. For the current design iteration it will be made from ABS and from testing the device will show points of fracture and failure that might need to be

changed or reinforced depending on the severity of the failure. Example is having the covers crack under the pressure of the weights, so a solution would either be changing the material, design, or thickness. All in attempts to improve the structural strength of the device. Currently there is no testing done on the device so points of interest have not been made but assumptions can be made on where points of failures might occur. The one of most concern is the front piece of the device, since there is a window for the LCD screen, which can be seen as the point of least material strength since pressure will be placed on a smaller surface area in comparison to the other sides and pieces of the device's frame. The tension on the front piece would most likely be the first piece to experience fractures on the plastic.

The material was durable during the construction of the device and behaved similar to wood. Reason being that the frame had to be drilled through and the plastic fibers did not crack as bad and acted fairly well when drilled into.

## 5.12 Breadboard Testing

Testing was required to ensure that the circuits were working as simulated. When it came to this we ran into a lot of issues. Especially, with the ADC design as there were a lot of components. Sigma Delta ADCs had more complexity than originally theorized.

Figure 79 shows the output of the original ADC when bypassing the op-amps used for the difference amplifiers and integrators. The input signal of a triangle wave is in yellow, the comparator is shown in purple, the switch in green, and the D flip-flop in orange. These were all working as expected from the simulation. The op amps however, were not this is most likely due to the simulation using ideal op amps. When testing this one we also were not using bypass capacitors on any of the power supplies on the op amps which may have added noise and contributed to them not working. Another reason that was learned when testing the next revision was that we needed to use different op amps suited to each function which was not done for this attempt. We just used the THS4222DGK however throughout testing we tried 4 different op amps and none of them worked as simulated. We also had to use a different D flip-flop than originally bought as it was a 16 pin D-latch because when originally buying parts we hadn't used digikey and were not able to find where the smaller D flip-flops needed for this application were on digikey. We also could have not been using a high enough clock signal as the second circuit used a frequency 8 times greater than the 5 MHz tested with this design. The original comparator of the MAX907EPA+ also did not work as expected however when testing with 4 other comparators the TLC3702CP worked much better, Overall we were not able to find the conclusive problem with this circuit as there were too many factors and it could have been the design itself. We were recommended to find a design that had been used and tested before.





Fig 79. Original ADC Design Testing

For the second circuit when testing there were many issues with the comparators only outputting 1V. This was most likely due to the voltage buffers not working properly. The design used the OPA633 as voltage buffers however, they have since been discontinued. Therefore, they were replaced with the BUF634AIDR which was recommended by digikey and other sources. When testing the buffers did not output anything. This would explain why the comparator was only outputting 1V as it wasn't getting any input from the sample and hold circuit. This was especially hard to test as the buffer required a minimum bandwidth of 35MHz which was too high for the function generator to output. This would also explain why the rest of the circuit was incorrect as the other voltage buffers would not receive the minimum bandwidth and therefore not work as well as the feedback loop inputting the wrong values back into the circuit.

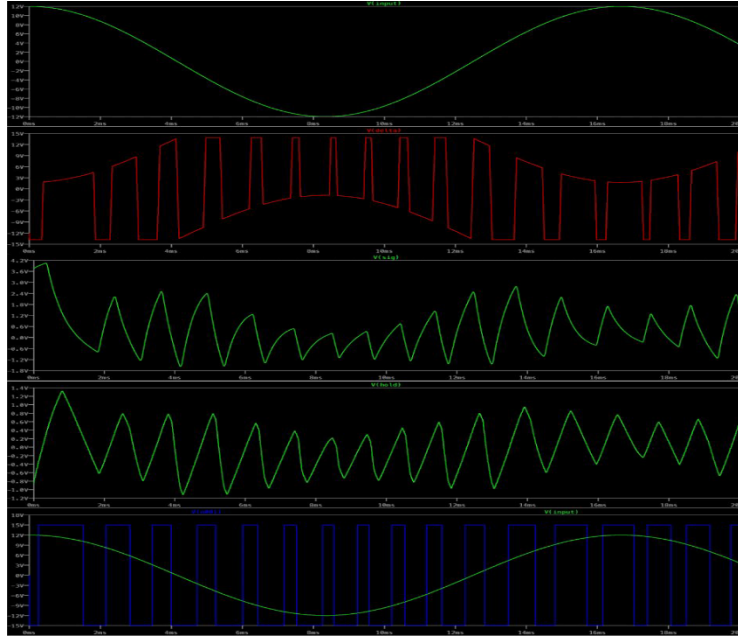


Fig 80. Second ADC Design Testing

Finally due to time constraints we decided to use a sigma-delta ADC IC the ADS1115. As shown in figure 81 the ADS1115 did work for the final project and was used to display the volume of the music being played on the LCD.



Fig 81. ADS1115 Testing

The pre-amplifier circuit was also breadboarded and tested. We used a gain of 15 in order to have a max voltage of 3.3 Vpp due to the limitation of the Raspberry Pi 4. Originally we were using a gain of 50 to bring the output of the microphone up to 5Vpp. However, we later learned the GPIO pins on the Raspberry Pi only allow up to 3.3V inputs. We tried using a level shifter to then bring down the voltage to 3.3 but found it easier just to adjust the gain. We wanted to bring the input up as much as possible so the ADC would have a higher accuracy. Then we tested multiple gains using different

resistor values in the feedback loop and came to the conclusion that anything above 3.3k Ohms would cause clipping too frequently.

### 5.13 Light Placement Testing

Finally, we began testing various placements of the light strips on various instruments. The first instrument we began testing with was the marching snare drum. We began by attaching the light strip along the inside of the upper rim as shown in the following figure:



Fig 82. Light Placement-Drum 1

Although it emitted a satisfying glow around the rim of the drum, the light strip was a millimeter thicker than the rim of the drum. This caused a problem as the drumsticks would end up hitting against the light strip and possibly damaging it. This led us to our next test, placing the lights inside the drum:



Fig 83. Lighting placement - drums 2

This resulted in a much more encompassing and satisfying glow along the head of the drum. Having the lights inside the drum also allowed us to hide the shine from the individual LEDs and made the glow much smoother. With this, we had found the optimal position for the lights to be placed on a drum.

## 6. Prototyping

Once all of the components had arrived after and the testing phase was completed, we were ready to begin prototyping our project. Our first prototype was with the marching snare drum. We placed the LED strip inside the drum, attached the clip-on microphone, and hooked up the Raspberry Pi.

### 6.1 First Full Prototype

Once everything was ready, we began by hitting the drum to make sure the color and brightness would change. The results can be seen in the video at the following link: <https://www.youtube.com/watch?v=hNbfKDOtJHo>. Although the drum was the only instrument on hand at the time, we wanted to further test the program and see how it would work with a wider variety of notes that are meant to result in a variety of different colors so we placed a mobile phone on the drum underneath the microphone and played a song for the microphone to pick up and react to. The results can be seen in the video at the following link: <https://www.youtube.com/watch?v=ueMOcoi3uU4>.



Fig 84. Prototype 1

The changing of colors was an uplifting success for the team. However, we noticed a delay when compared to the notes being played by the song. This was a result of a latency issue in the code. It is an issue that will be fixed as we continue to polish the software for our project and move on to further prototypes.

## **7.Design**

Once the testing phase was complete, we had all the necessary components to begin building our product. Initially, we split up into a software team and a hardware team. The software team began working on the programming for the Raspberry Pi 4, microphone, and LED lights. The hardware team began working on the DC-DC and AC-DC convertors, display, and mathematics for converting sound into numbers.

This chapter is dedicated to the design process of the project. It serves as a record of the building process of each aspect as we encounter the various struggles and successes. It mentions every design step until we reach our final product.

### **7.1 Software**

The software team, consisting of Carson and Leith, initially began with setting up the operating system on the Raspberry Pi 4. Although some difficulties were met with setup, specifically with time on Raspberry Pi OS refusing to sync up to allow us to use the internet, following some trial and error with Raspberry Pi configuration through the console we were able to quickly get the device working with all the necessary programs running on the Raspberry Pi itself. Once the Raspberry Pi was running properly, we began programming the project to read in the sounds being played into the microphone and converting them into numerical values of frequency and volume then sending those values as an output into the LED strips to determine the colors and brightness levels they displayed.

#### **7.1.1 LED Strips**

Once the pi was set up, our first goal was to program the main body of the code to control the brightness and color of the LEDs on the light strip. This was done quickly and easily through Python's "Neopixel" library. Using values of 0-255 for each of the three RGB values, we were able to change the color and brightness of each light as we saw fit. We later implemented predetermined color themes and a custom color theme to control the RGB values of the lights. Although the lights seemed very bright from a close distance, the intention for the project was for viewers to spectate the show from their audience seats as opposed to three feet away from the performer.

### **7.1.2 Microphones**

Our next goal was to write a program to convert the sound that was being inputted into a microphone into integer values. This program also had a convenient Python library called “pyaudio”. Using this library, we were able to take the sound being read by the microphone, and turn it into an integer value depending on its volume. The integer range for the volume was 0-32,768 with 0 being no sound and 32,768 being the max volume input. We then wrote a section of code that would take the value of the volume and divide it by 129. This resulted in every 129 units of volume to be converted into 1 RGB value. For example, if we had a volume of 16,125, it would return a value of 125 which would be the brightness of the LEDs.

### **7.1.3 Microphone and Light Cooperation**

The next step was to combine the program that controls the LEDs with the input from the microphone. For this, we had to temporarily halt the programming of LEDs until the microphone arrived the next day. Once the microphone arrived we were able to continue programming the LEDs along with the microphone. We were able to utilize the “pyaudio” and “audioop” libraries to retrieve information from the microphone into a simple python program. The program that we started to develop initially only was able to detect the overall volume from the audio input of the microphone. With this program we were able to see the volume from the microphone in text form from the console in a range of 0-255, these values could then be assigned to a variable and then that variable could be utilized in conjunction with the Neopixel library to adjust the brightness of the LEDs.

After the audio amplitude functionality of the microphone was determined we then set our sights on frequency analyzation through python. To test this we first tested the microphone through the audio recording program known as Audacity to ensure that our microphone was recording correctly without imperfections and after that we had the confidence to continue with implementing frequency analyzation in real time from the microphone. At this point in the design process we were able to implement frequency detection in the program via “pyaudio” by utilizing a fourier transform for the frequency of the audio input to obtain the pitch of the audio in Hz from 20-20000 Hz which is also the frequency range that the average human ear can detect.

Once we were able to extract the frequency of the input audio from the microphone in Hz we were then able to utilize a lookup table and assign each note in the chromatic scale (A, Bb, B, C, Db, D, Eb, E, F, Gb, G, and Ab) to a frequency range in Hz. The note A in the fourth octave is defined as 440Hz so when we wanted to detect the note A in the program we checked for the value of 440 Hz however there is a problem with this implementation. In the real world with wind, string, and even percussive mallet instruments there are minor imperfections in the instruments themselves so no instruments besides fully synthesized ones can accurately hit exactly

440 Hz consistently during a performance. Many wind instruments can become flat or sharp depending on the temperature outside which can affect the instruments overall ability to accurately produce correct frequencies. To combat this real world problem we instead assigned every note in the chromatic scale to a range of frequencies in Hz. For the note A instead of assigning it to only 440 hz we instead assigned it to the frequency directly in between Ab and A on the low range and the frequency directly in between A and Bb, this gives us for example a range of 427 Hz - 453 Hz which recognizes any frequency in that specified range as the note A regardless of if it is sharp or flat.

Following the discovery of utilizing ranges of frequencies for our lookup table rather than exact frequencies we then implemented the known frequency ranges for all 12 notes on the chromatic scale from octave 1 to octave 5 (A1-A5). Defining notes for multiple octaves gives the program the information to identify not just notes in one octave but notes over multiple octaves that are commonly used in musical pieces. We intend to add octaves 6-8 later in the programming process for the project however, as a starting point to ensure program functionality we decided that 4 octaves was more than sufficient (48 notes total).

After we implemented the program functionality for pitch recognition we then tested the program with our microphone and LED light strip to ensure that the program functions as intended. In the Senior Design lab we were able to recreate different colors for each pitch that we had played into the microphone however outside background noise did interfere with the programs ability to decipher pitches effectively so oftentimes we would see jumps in color when anyone in the lab had made a loud noise. This issue will eventually be remedied by a low pass filter in conjunction with a directly mounted microphone to the instrument. These two factors working together should help to minimize unwanted noise and relieve strain from the program to decipher actual notes from background noise.

## **7.2 Hardware**

The hardware team, consisting of Anja and Stefan, designed and tested the DC-DC converter, pre-amplifier, and analog-to-digital converter. The preamp and analog-to-digital converter were designed, simulated, tested, and then printed onto a printed circuit board. The first and hardest step though was the design process.

### **7.2.1 Sigma-Delta ADC**

When beginning to design the ADC the first specification to decide was what order the converter should be. As the order increases due to noise shaping the noise gets lowered in the beginning but leaves a lower frequency that can be used without noise as it gets pushed to higher frequencies. However that doesn't affect our design as much as humans can only hear to 20kHz therefore as long as the usable range is up



to 100kHz we can still use it and get the desired resolution. However, with each added order the complexity as well as size of the circuit increases because you have to add on another difference amplifier and integrator for second order. It gets more complex, as more than that has to be added increasing past second order. This also adds complexity when calculating the component values. Due to those factors we decided to go with a second order ADC.

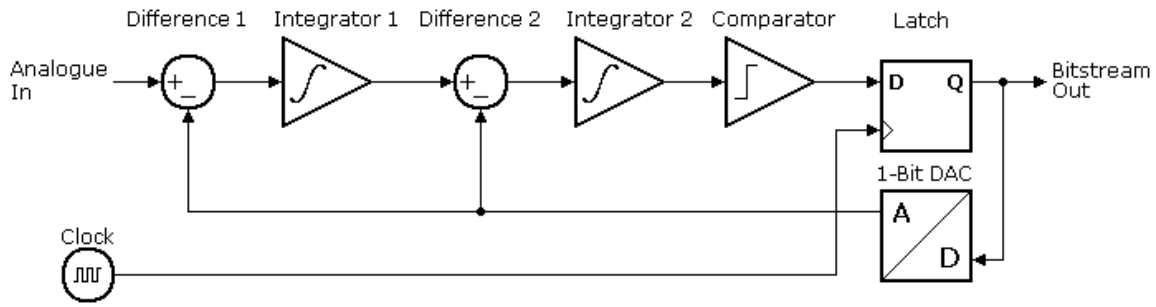


Fig 85. Second Order Delta-Sigma ADC

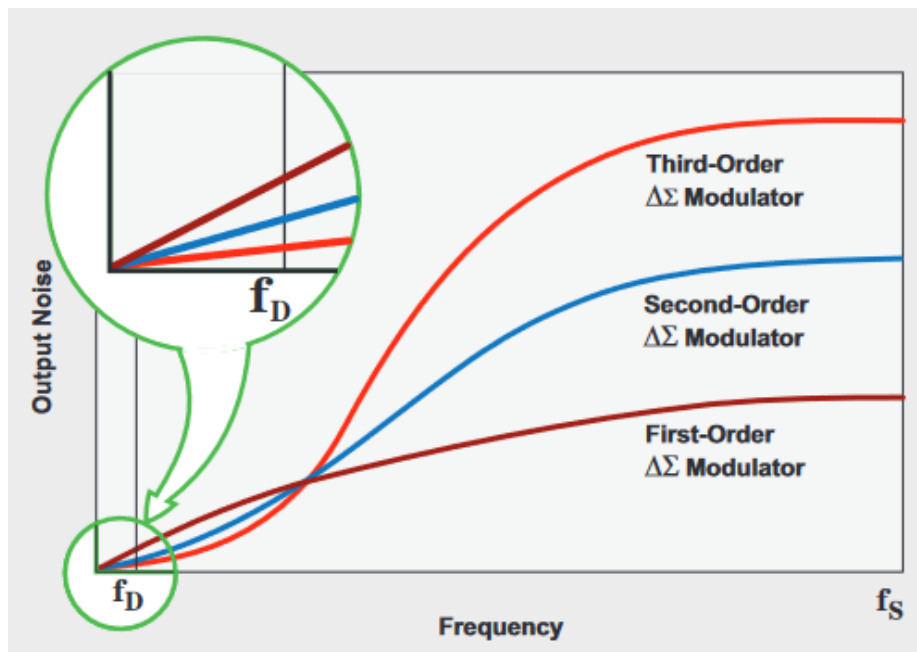


Fig 86. Noise Shaping Differences Between Orders for Sigma-Delta ADC

The first thing considered when designing the ADC was the op amp. It had to be high speed and low power preferably. If the op-amp was 5V it could be powered by the pi instead of using a boost converter to change the voltage for the op-amp supply. However that does limit the range of voltages that the op-amp can operate at. The op amp was needed for 2 tasks for the difference amplifier and the integrator. We originally decided to use the same type of op-amp for both. The high speed factor was the most important specification as it needed to process a continuous signal with speed and



accuracy. There needs to be as little latency in the system as possible so that the LEDs are in sync with the music. However, the Sigma-Delta ADC isn't the fastest due to the oversampling to reduce noise but since this project deals with an audio input the resolution was more important. If the resolution was too low the fourier transforms the Raspberry Pi is calculating would be off. This would cause the whole system to be unusable. Therefore, we originally chose the THS4222DGK due to its speed as well as its 5V power supply option.

The second component considered was the comparator. The comparator acts as a 1-bit ADC; it takes whatever analog signal is inputted and decides if it is higher or lower than the ground depending on whether it gives a 5V or 0. This like the op-amp had to be a high speed design to keep the system running efficiently. For this we chose a MAX907EPA+. This does use a 5V supply so we could power it with the 5V from the Raspberry Pi.

The third component was the 1-bit DAC. This was made through a R-2R switching ladder circuit with an inverting op-amp. The switch at the bottom is controlled by the comparator whenever the comparator is high it moves the switch to give a  $V_{ref-}$ . When the comparator is low it stays at  $V_{ref+}$ . The output then goes through the inverting op-amp to correct the positives and negatives.

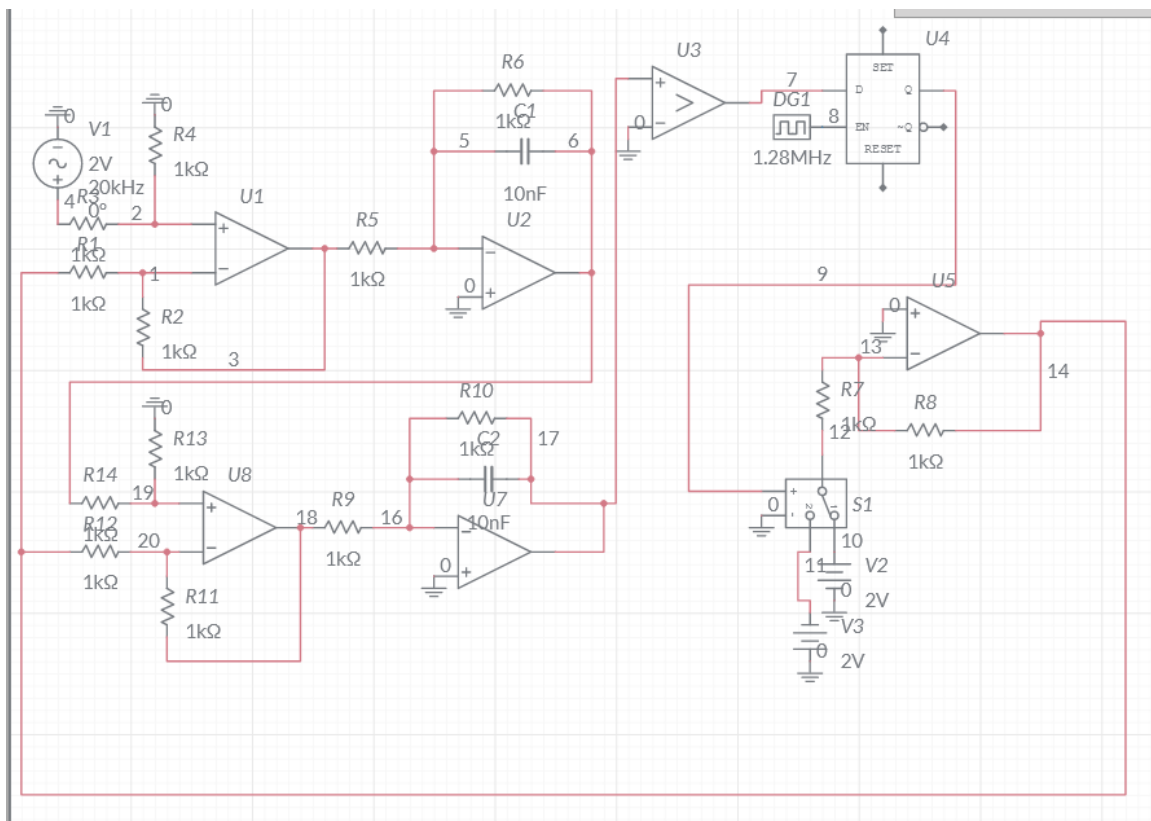
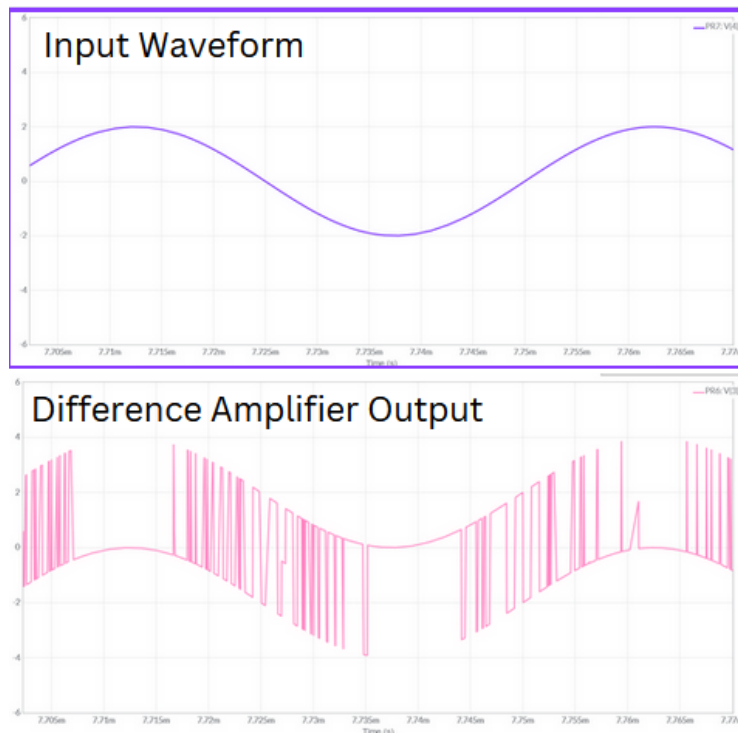


Fig 87. Second Order Sigma-Delta ADC Schematic

The next step was to start designing the circuit. This schematic shows three sets of difference amplifiers and integrators feeding into a comparator and then into a D flip-flop. The resistor and capacitor values were chosen due to the equation  $1/\omega_p = RC$  so for this circuit it would be  $10\mu s$ . The supply voltages were chosen per specification from their data sheets. The input for this test case was 20kHz because that is the maximum frequency that humans can hear therefore if there were any frequencies higher we wouldn't care about them for this project. The clock frequency was chosen to be 1.28MHz. This was due to the equation mentioned above in the first paragraph due to the oversampling the equation for resolution is  $(N)^*(L+.5)$  where L is the order of the function and N is the amount of times the oversampling rate doubles the original frequency. Therefore, for this circuit it would be  $(6)*(2.5)$  which is 15-bit resolution. That should be sufficient for this project.



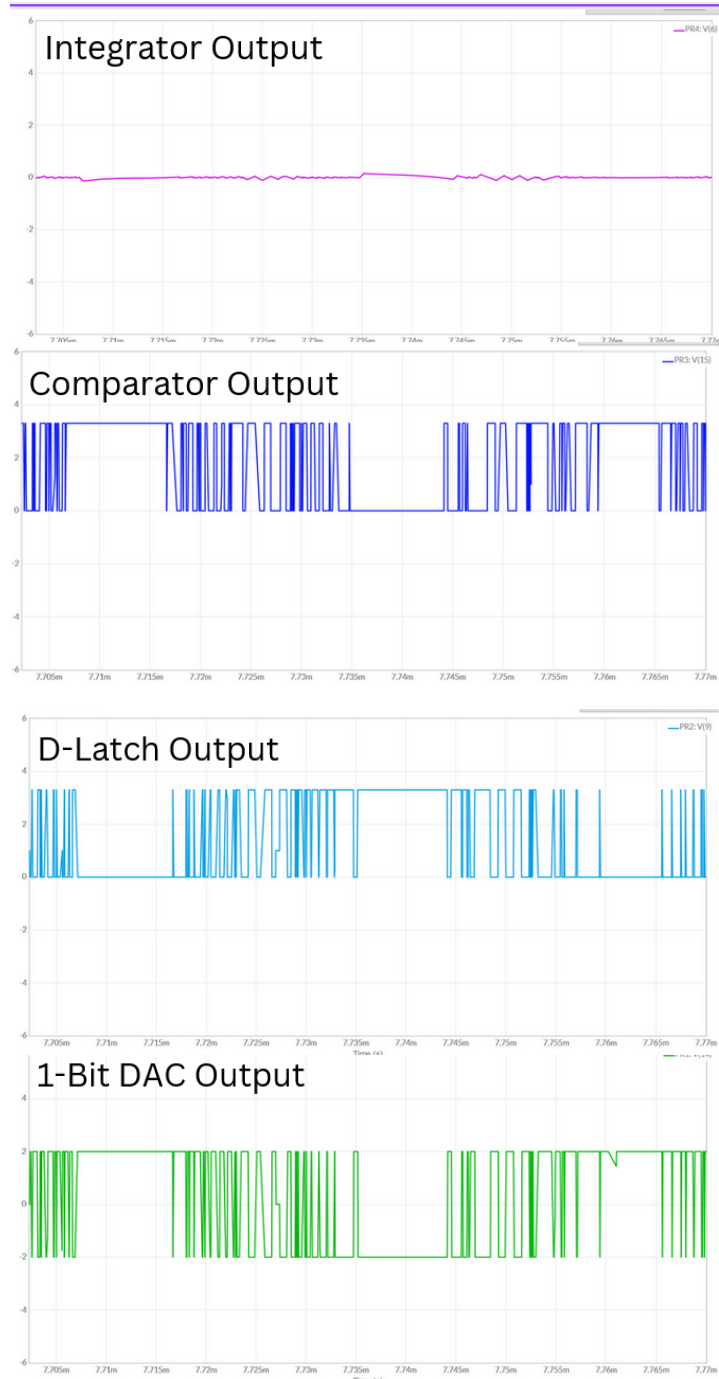


Fig 88. Simulation Results for Sigma-Delta Modulator

These simulations show what is happening in each step. The second shows the output of the difference amplifier which is subtracting the output of the 1-bit DAC from the input. The third shows the integrator's output which resembles a triangle wave since the input has a square wave merged into it. The fourth is the output from the comparator which is a PWM signal which means the amount of highs and lows directly translates into where it is in the signal. Then it goes through the latch and 1-bit DAC to

change back into an analog signal to be fed back into the system. This is done so the modulator can compare the new data to the last point calculated which is much faster.

To test the circuit was working as desired we tested the noise shaping of the simulation data by finding the FFT of the output. Each time the oversampling rate is times by 2 the frequencies are expanded over 2 times the frequencies. It also reduces the noise by 3dB each time the oversampling rate is doubled. This causes the signal to be a large peak in the beginning with low noise which reduces and eliminates the quantization noise increasing the SNR. Under ideal circumstances the SNR should be  $6.02N+1.76\text{dB}$  so for this circuit which has an oversampling rate of  $2^6$  it would be 37.88dB. The noise shaping allows the frequency distribution to skew towards higher frequencies. It does this by using the feedback loop to minimize error between the input and the output by comparing them with the difference amplifier.

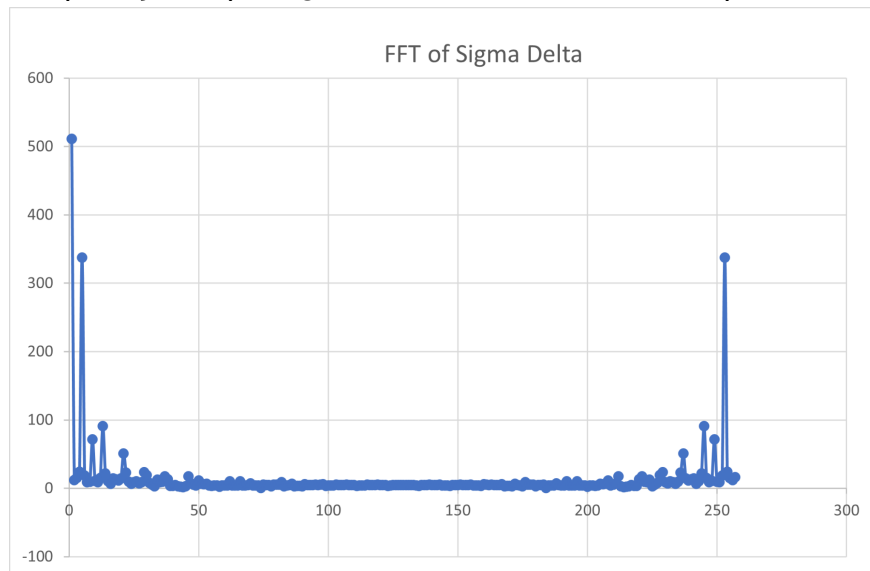


Fig 89. Noise Shaping of Sigma-Delta

The second design attempted was a design done by MIT students. They built a sigma delta ADC PCB for their project. Their design was similar to the one designed for our project but they included a sample and hold circuit as shown in figure 88. Their output for the simulation is very similar to what the original design was outputting for the simulation as shown in figure \_\_. They also included voltage buffers. The parts used for this were an LM741 for the difference amplifier, LF356s for the integrator and subsequent op amps, Lm311 for the comparator, and BUF634AIDR for the voltage buffers. Also 2N7002 MosFETs. They used a 60MHz digital clock however this project is using an 8MHz oscillator adapted to a square wave and then pushed through an 8 times multiplier to create a 64MHz clock.

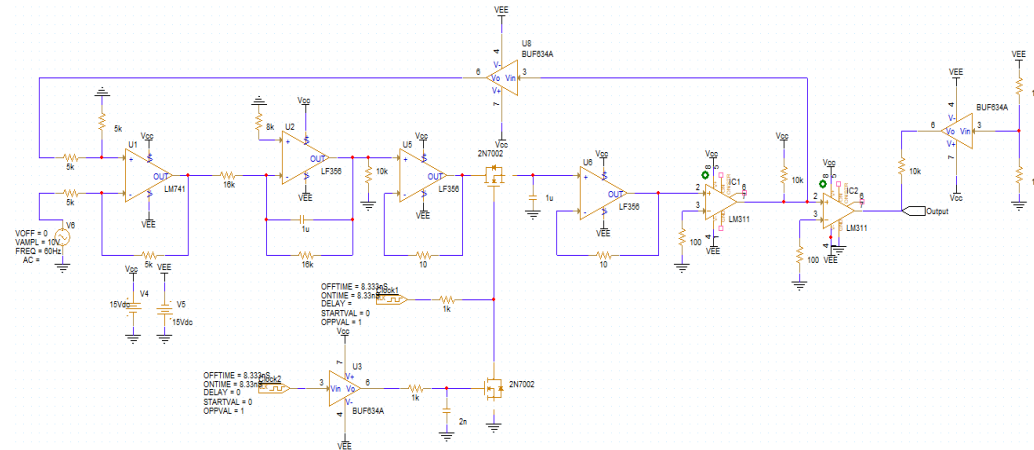


Fig 90. Second Design Sigma-Delta ADC Schematic

The third and final design was an ADC IC, the ADS1115. For that the specifications were that it needed to have a high resolution of at least 15-bits and have a high SNR value as both of those are very important when converting audio signals. The ADS1115BQDGSRQ1 was chosen as it had libraries for the Raspberry Pi as well as a module by ADAFRUIT. The ADS1115 has I2C pins to communicate the output directly to the Raspberry Pi. The design was built as specified in the datasheet using 10k pull-up resistors on the alert, SCL, and SDA pins. It also had a pull down resistor on the alert pin with 1uF bypass capacitors on each voltage supply. The design also had fuses to prevent the circuit from overheating due to too much current.

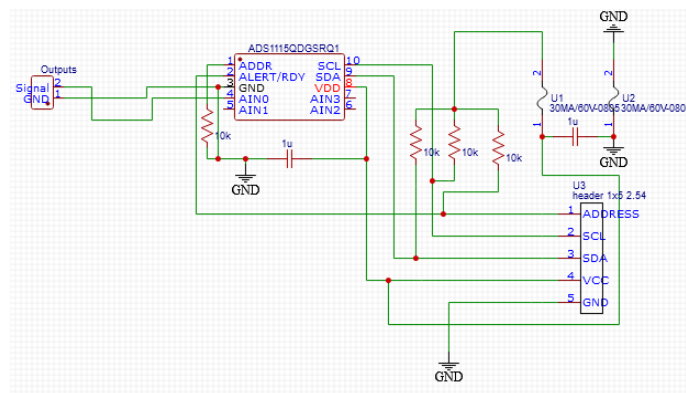


Fig. 91. ADS1115 Schematic

## 7.2.2 DC-DC Converter

This is the design that was used for the DC-DC converter and reserved in the attempt to make the device use a portable power supply, like a battery. The converter was meant to be in series with the power supply to ensure the output voltage of 5 VDC, with an overall maximum efficiency of 88.1%. Meaning that the DC-DC converter is able to accurately send the desired 5V DC voltage with not as much margin of error.

Although, due to time constraints and the focus on the ADC and pre-amplifier, despite having the pcb on hand, we did not have enough time remaining to implement it into our project.

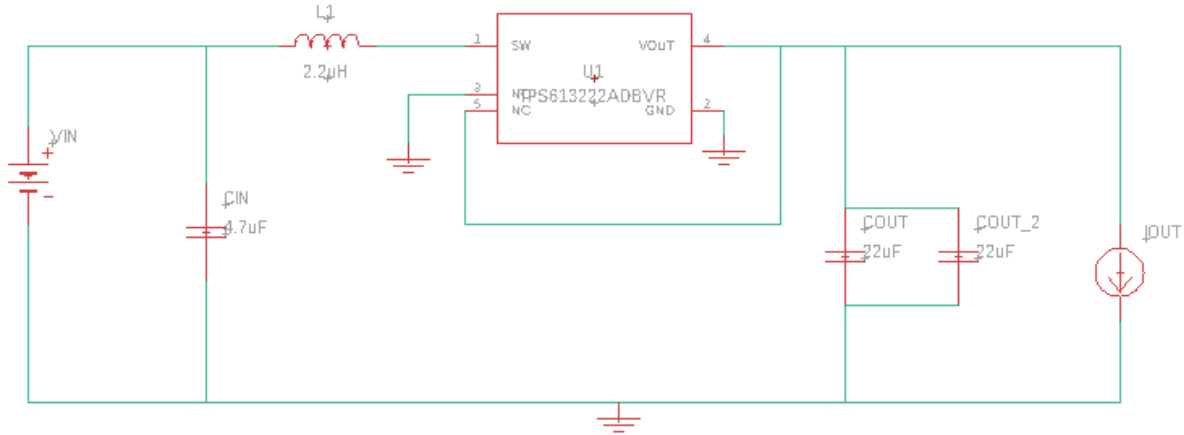


Fig 92. DC-DC Converter

### 7.2.3 Pre-Amplifier

The design was made to be in conjunction with the clip-on microphone that was used in the testing of the project, and is set to output the signal directly into the ADC. Specifically designed to take the output of the mic after the phantom power supply and amplify it from mic level to a voltage that would work with the analog-to-digital converter. Originally the mic level was measured to peak at 50mV. However, the input used wasn't loud enough so when it was measured again it peaked at 200mV. Originally as well the gain was designed to be 50 to multiply the original peak of 50mV to 5Vpp. The Raspberry Pi GPIO pins only take a maximum voltage of 3.3V. That with the remeasured peak of 200mV led us to design a pre-amplifier with a gain of 15. The original design amplified the signal at a gain of 50x which made the signal too large and caused clipping which obscured the output enough for the ADC to not process without the risk of clipping. Hence the use of the gain of 15 to avoid the risk of clipping within the ADC.

The schematic was based on the reference design for a non-inverting microphone preamp from Texas Instruments. With the microphone and gain specifications calculated, we used the following schematic:

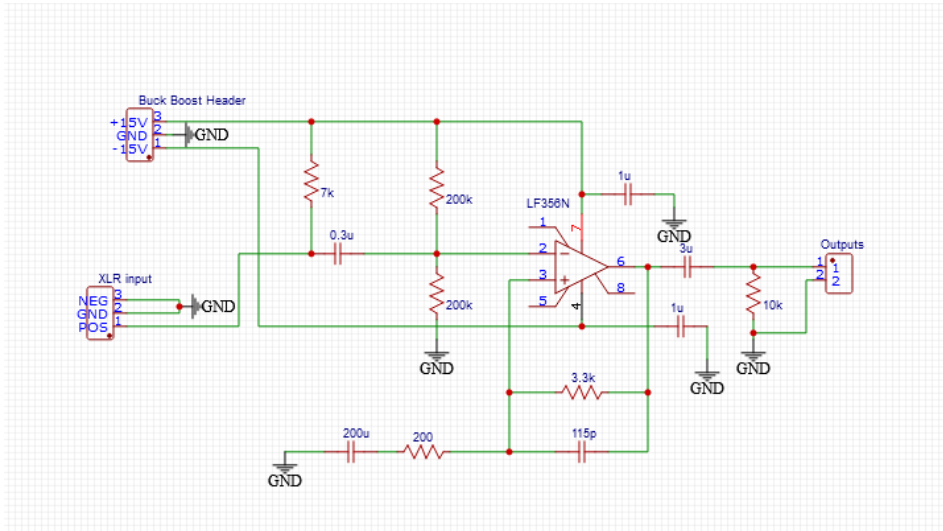


Fig.93. Pre-Amplifier Schematic

The Pre-amp only needed one iteration in the making of the PCB, and can even be adjustable from the actual PCB, meaning gain can be altered without changing the PCB footprint. This was desired to ensure the most optimal output based on the components available.

### 7.3 Overall Schematic

The overall schematic is the implementation of all the known electronic components interconnected. Meaning that the schematic contains all the components from previous schematics onto one. The figure below shows the eagleCad schematic of all the components and also shows how they are arranged. This was important for the fabrication of the device due to the various ports being used on the Raspberry Pi 4 board and understanding where the appropriate placements of these components needed to be. The source for these components come from the eagleCad libraries exported from Snapeda and UltraLibrary. Once exported eagleCad had all the compatible components needed to make the desired circuit for the device. In the case for the overall schematic, mashing all the circuits together would not prove to be efficient due to the wire placement and arrangement ground. This shows both of the PCBs used for this project and how they are connected via headers.



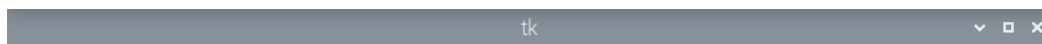


# Welcome to FRGB Instruments!

Select Theme... ▾

Continue...

Fig 95. Program Landing Screen



# Welcome to FRGB Instruments!

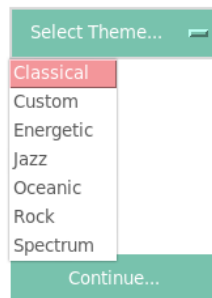


Fig 96. Theme Selection Menu

The next screen in the program, if the user selects the custom theme, is the color selection screen. On this screen the user can individually select the color that is applied to each musical note. Once the button that corresponds to a musical note is pressed, a popup appears in front of the screen. This popup allows the user to drag three sliders that correspond to the RGB values of the color spectrum. The result of the selected values is displayed to the user on the right of the popup along with the colors hex value which the user can also manually select. After the user has selected a color, they press “ok” on the popup and the color they chose is displayed under the button for the musical note they chose to edit.

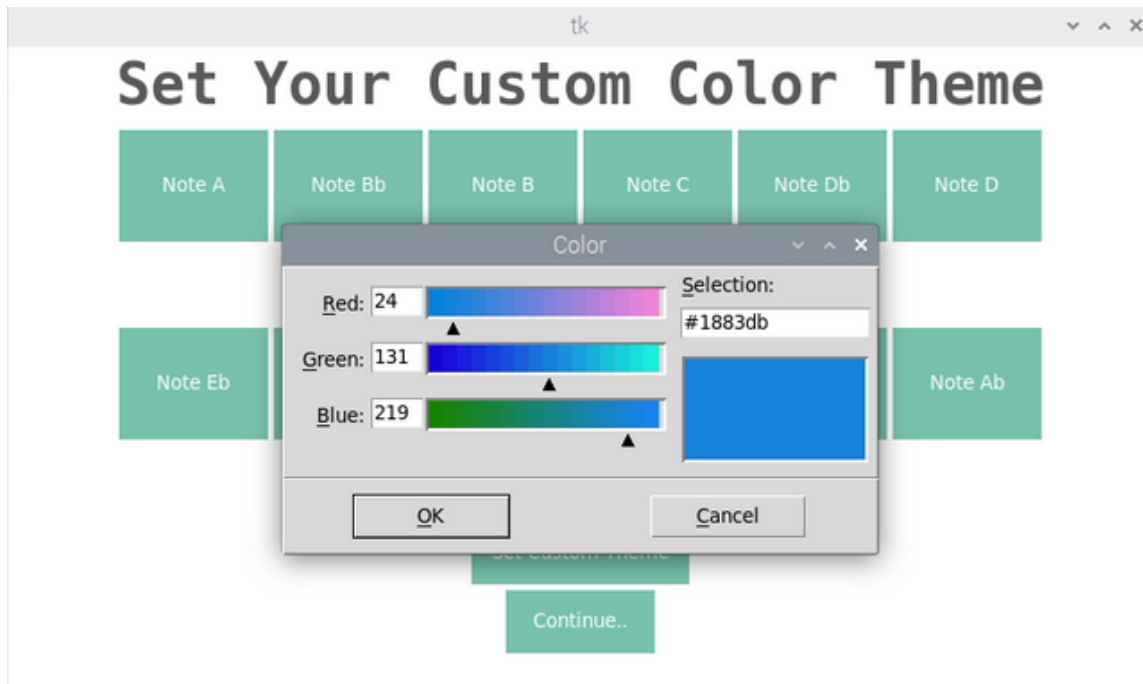


Fig 97. Color Selection Screen

The next screen of the program is the main active screen of the program that shows a real time readout of the intensity of the microphone input along with the real-time frequency to note conversion value based on the input picked up by the microphone. On the top of the screen, a notification is displayed telling the user that the visualization is currently in progress. In the middle of the screen, we have real time data readouts with the right-most readout being the current musical note being played in the 12 note chromatic scale after the note has been processed through the program, to the left of the musical note we have a volume value and a meter which shows the real time volume and brightness of the lights while the note is being played. This can also be used to calibrate the placement of the microphone. If the user sees the volume meter peaking constantly they can move the microphone further away from the instrument to get a clearer readout so as to not distort the audio input. Along with these useful readouts the user also has access to another button whose functionality allows the user to end the program all together and return to the applications home screen with the restart button.

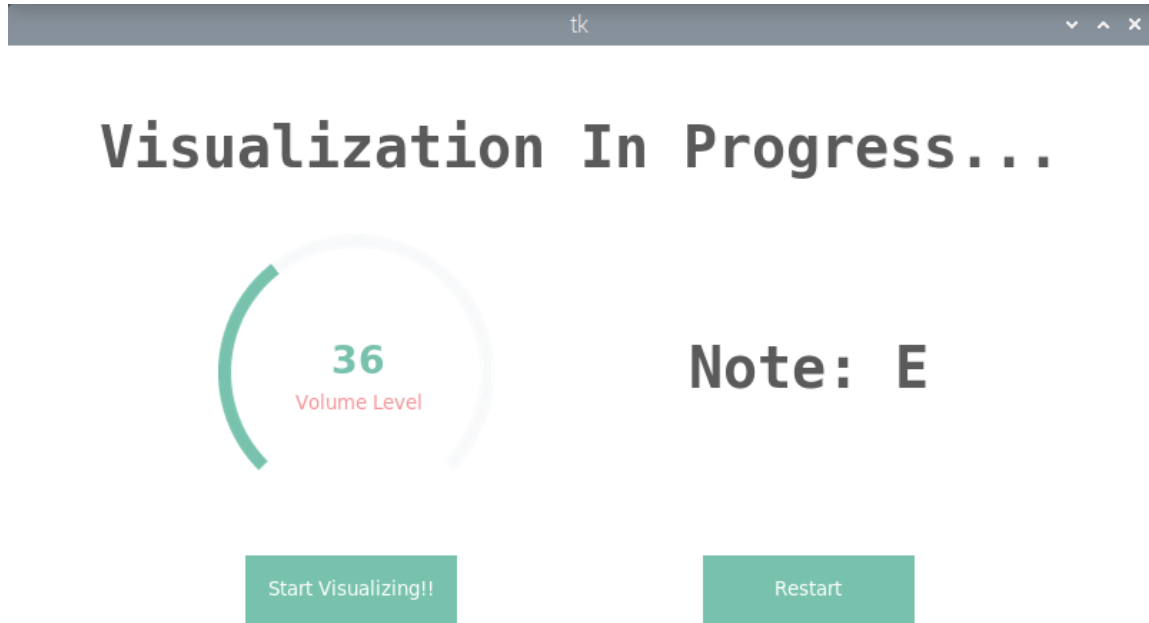


Fig 98. Program Active

## 7.5 Screen and Raspberry Pi Enclosure

The enclosure consists of three pieces: Front, Center shelf, and Rear. Each piece is responsible for holding their respective components along with their main purpose of protecting the electronics and to keep the system compact. The Front pieces are responsible for holding the LCD screen within the Center Piece front cavity. The Front piece is also mounted with screws on the four corners of the design and has its edges rounded slightly to reduce the rough edges of the design. While the Center Piece is designed to hold the majority of the components like the Analog-to-Digital Converter, and the preamp. The components and printed circuit boards are mounted on the internal shelf with screws, washers, and hex nuts. The washers and hex nuts alleviate pressure on the actual platform and increase the stability of the printed circuit boards. The Rear piece is responsible for protecting the printed circuit boards, wiring, and other components that are attached to the center shelf.

## **8.Implementation**

The implementation chapter is dedicated to the final prototypes and construction of our product. The final updates to the hardware components and software programs in order to maximize the efficiency and quality of the product.

### **8.1 Software**

The software of our first prototype consisted of a python program implemented in the Raspberry Pi 4. Using various libraries, such as PyAudio, NeroPixel, and ADS1115, we created a program that reads an input from a microphone as the music is being played and determines the frequency and volume of the music. We then used various ranges of frequencies to determine the musical notes. These ranges then decided the color that was displayed by the LED strips. The numerical value of the volume of the sound being played into the microphone was also split into multiple ranges and turned into the level of brightness that the lights displayed the colors in. We then used the TKinter and TTKbootstrap in order to create the GUI for the program which made it much more user friendly. The program first sets the gain to a value of 2, sets the sample rate and chunk size, then conducts the necessary calculations in order to convert the frequency into integer values and volume into a value of 0-100. Next it uses the frequency value to determine the musical note being displayed and displays the color attached to that note at a brightness corresponding to the volume level at a value of 0-1.

### **8.2 Hardware**

The Hardware used in the design are the Raspberry Pi 4 microcontroller, the 5” DSI touch screen LCD, Analog-to-Digital Converter, preamplifier, buck booster, and the LED strip. These components have been carefully selected to be the most compatible and accessible to achieve the design specifications of the project.

### **8.3 Designs**

The design was finalized to be an external device on the user to be able to process and send feedback information. The input information is produced by the instrument being played, then the input frequency is processed by the ADC and sent to the Raspberry Pi 4 microcontroller. Once processed the information is sent to the LCD screen and to the LEDs strip to then have a desired color and brightness to ensure the right mode of the system. The device also uses a touch screen to minimize the use of buttons as well, meaning that the microcontroller also has to process information being sent from the display and send accurate feedback to the corresponding systems being used. The latency for both the frequency input and display input should be at 100ms. The input is received from an external microphone that is placed in the best location to

read the instrument's input frequency and be the trigger for the device. The microphone sends the input frequency to the ADC which sends it to the microcontroller for processing. This signal is used to determine the volume, there is a secondary cord attached to the microphone which sends the frequency reading to the microcontroller. The microcontroller sends a feedback signal to the LEDs strip and to the LCD screen in correspondence to the mode it is in.

## **9. Administration**

This section discusses the milestones and budget of the project. Our initial budget of \$1000 was well over the required funds for the project. The milestones in this section were to help stay on track with due dates as well as improve our own time management. This was very helpful throughout this project.

### **9.1 Budget**

The overall budget of the project depends on how many parts are ordered and at what price each of those parts are purchased at. Overall there are 8 parts that contribute to this project that must be purchased with many parts having multiple options with different price ranges depending on quality and quantity of the product. The overall goal of the project budget is two fold, to stay under a total budget of \$1000, and find the best parts to use at the least expensive price point. Essentially in this portion of the project we want to find the best products suited for the project while having a reasonable purchase cost, basically we want the best price to performance ratio for all of our parts so that nothing is wasted on overpriced parts or excessive parts. Not all overpriced components can be avoided however, the price of the MCU component due to supply shortages has been highly inflated over the MSRP (Manufacturer Suggested Retail Price) which gives us little options in finding an affordable option for that specific part.

<b>Part</b>	<b>Cost Estimate</b>	<b>Description</b>
ST-SOT23-5	\$9.95	D-FF SMD to DIP adapter
MAX4494AKA+T	\$24.92	Op-Amp
AD8566WARMZ-REEL	\$21.63	Op-Amp
MCP6282-E/P	\$8.54	Op-Amp
LM258AP	\$2.25	Op-Amp
TSV792IDT	\$21.20	Op-Amp
TC7W74FK,LF	\$1.26	D Flip Flop
MC10EL52DG	\$12.62	D Flip Flop
SN74ABT16373ADLR	\$3.96	D Latch
MAX907EPA+	\$28.62	Comparator
NJM2903D	\$5.44	Comparator
TLC3702CP	\$11.20	Comparator
LM393DT	\$2.10	Comparator
ACPL-0873-500E	\$8.14	Digital Filter with Decimator
LTC6930IMS8-5.00#PBF	\$20.15	Oscillator
MAX4544CPA+	\$5.56	Switch
<b>Total</b>	<b>\$187.54</b>	

Table 10: Original ADC Budget

<b>Part</b>	<b>Cost Estimate</b>	<b>Description</b>
LM741CN	\$2.73	Op-Amp
LF356N	\$9.20	High density RGB LED strip with 144 LEDs per meter to light up the flute
BUF634AIDR	\$33.48	Buffer
LM311D	\$3.35	Comparator
SIT9121AI-2C2-33E60	\$10.90	60MHz Clock
2N7002	\$1.85	MOSFETS
ECS-2200B-666	\$9.45	66.666MHz Clock
501MLFT	\$11.25	Clock Multiplier
LTC6930MPMS8-8.00	\$16.48	8MHz Clock
<b>Total</b>	<b>\$98.69</b>	

Table 11: Second ADC Budget

<b>Part</b>	<b>Cost Estimate</b>	<b>Description</b>
Microphone	\$79.99	Floor Clip-on Microphone to pick up music being played.
High density LED Strip (144 LED/m)	\$21.99	High density RGB LED strip with 144 LEDs per meter to light up the flute
5in Touchscreen LCD IPS/TFT display	\$48.49	Touchscreen LCD display to apply different light modes and settings can be IPS or TFT
XLR splitter	\$12.88	XLR splitter used to split the microphone between the Raspberry Pi and the Pre-Amp
Power Supply	\$13.98	Power supply used for the LEDs
Level Shifter	\$8.99	Used to attempt to bring the voltage of the Pre-amp to 3.3V for the GPIO pins of the Raspberry Pi
Terminal Blocks	\$9.89	Used to Break out the power supply to the LEDs, Buck booster, and ADC
ADS1115 Module	\$15.99	Used to test the ADS115 PCB designed for accuracy
SMD to DIP adapter set	\$11.99	Used to test various SMD components
Raspberry Pi 4 Microcontroller	\$209.90	Raspberry Pi 4b 8GB
6-pin SMD to DIP adapter	\$8.38	6-pin adapters
3-pin SMD to DIP adapter	\$7.99	3-pin adapters used for the MOSFETs
Boost Converter	\$9.29	Used to convert the 5V power supply to 15V for the Pre-amp
Power Supply	\$11.89	Power Cable with switch to turn off power to the Raspberry Pi



ADC 1	\$187.54	All components for the first ADC design testing
ADC 2	\$98.69	All components for the Second ADC design testing
ADS1115BQDGSRQ1	\$46.62	ADS IC used in PCB
ADS1256IDBR	\$16.83	ADS IC back up test
PCBs	\$20.50	ADC and Pre-amp PCBs
Misc Components	\$43.19	Components used for DC-DC Converter and ADC such as resistors, capacitors, and inductor
<b>Total</b>	<b>\$875.12</b>	

Table 12: Overall Budget

## 9.2 Project Milestones

Initially, for the first semester, the primary milestone is to have a finalized design for the project. We plan to have all parts and design specifications finalized by the end of the semester along with our final report so that we do not need to spend time in the next semester ironing out any design flaws; allowing us to focus on building the actual project. We have a secondary goal of meeting specific deadlines for the paper that we placed upon ourselves in an attempt to finish the report ahead of schedule in order to begin building the project before the first semester comes to a close.

Milestone	Deadline
General Idea	09/16/2022
Part & Design Research	12/24/2022
Parts Testing	01/01/2023
30 Page Draft Complete	10/15/2022
60 Page Draft Complete	11/04/2022
90 Page Draft Complete	11/15/2022
100 Page Draft Complete	11/18/2022
120 Page Draft Complete	11/31/2022
Final Report	12/06/2022
Begin Building	12/07/2022

HARD DEADLINE – SOFT DEADLINE  
 Table 13: Milestones Fall 2022

The primary milestone for next semester is to have the two main features of the project functioning properly. These features are RGB lights that adjust their color according to the frequency of the sound being read in and brightness that adjusts according to the volume. There is a secondary milestone of creating an interactive app to create prebuilt light settings for the instruments and making the entire project more innovative than the initial idea (such as the implementation of robot vision to project the light on the instruments) through constant communications as we work on the project.

Milestone	Deadline
First Feature Functional	01/15/2023
PCB Built	1/15/2023
Second Feature Functional	02/01/2023
Hardware Testing Complete	02/01/2023
Both Features Polished	02/10/2023
App Rough Draft	01/01/2023
App Functional	01/10/2023
Project Complete	04/16/2023

HARD DEADLINE – SOFT DEADLINE  
 Table 14: Milestones Spring 2023

## 10. Conclusion

Frequency Based Instrumental Lights (FBIL) started as a small idea from a University Professor in the computer engineering department, credited to Dr. Richard Leinecker. The idea was to have a device that can distinguish sounds produced by an instrument, Dr. Leinecker's desired instrument was the flute, and had the device illuminate the instrument with varying colors based on the frequency the instrument is playing. The idea came to be with the desire to incorporate more technology into modern day instruments. The use of lights and noise detection is common within daily life, however, being able to distinguish the frequency by filtering out the noise and having an accurate feedback that corresponds to the input is relatively new. Meaning that there was a need that had to be filled, and this is what inspired the team to formulate a design that is capable of improving the relations between a person and an instrument.

FBIL processes information from a microphone that is stationed in front of the instrument and then sends that information through the ADC and then proceeds to the Raspberry Pi 4 microcontroller, which collects and processes the information. The feedback information is then sent to the LEDs and the LCD to decide the modes that are being played. All the electronics in the device were carefully selected for the insurance of having the FBIL perform its desired tasks with the best efficiency to cost ratio.

Throughout brainstorming the features and capabilities that FBIL can handle, parts were being ordered for the purpose of getting ahead in the fabrication of the device. Reason being that the device must undergo various testing practices before being considered a viable product that can be evaluated by other engineers. The team has already begun the construction of the device in terms of the physical model of the device and the software within the microcontroller. The other electronics are being purchased and were built ahead of the schedule to avoid complications.

In terms of personal milestones for the team, some were easier to meet than others. As the project is relatively simple, we struggled to meet the deadlines for the various drafts, such as the 60 page draft and the 120 page draft, throughout the project. However, once we were able to order the parts, we were able to begin testing and prototyping much earlier than originally anticipated. As we began to see successful tests, this increased the team's enthusiasm and morale for the project. For example, building was not set to start until December however, we were able to begin in early November, a full month before our projected start time. The team looks forward to completing the product and seeing what we can accomplish with our own abilities.

# 11. Appendix:

## 11.1 Bibliography

[1] “COB vs SMD: Which Led Is Better & Brighter.” *Gindestarled*, 15 Dec. 2020, [www.gindestarled.com/cob-vs-smd-which-led-is-better-and-brighter/](http://www.gindestarled.com/cob-vs-smd-which-led-is-better-and-brighter/). Accessed 30 Oct. 2022.

[2] Shechmeister, Mathew. “Birth of the Microphone: How Sound Became Signal.” *WIRED*, WIRED, 11 Jan. 2011, [www.wired.com/2011/01/birth-of-the-microphone/](http://www.wired.com/2011/01/birth-of-the-microphone/). Accessed 26 Oct. 2022.

[3] Wikipedia Contributors. “Microphone.” *Wikipedia*, Wikimedia Foundation, 27 June 2019, [en.wikipedia.org/wiki/Microphone](https://en.wikipedia.org/wiki/Microphone). Accessed 26 Oct. 2022.

[4] Bai, Mingsian R., and Yi-Yang Lo. “Refined Acoustic Modeling and Analysis of Shotgun Microphones.” *The Journal of the Acoustical Society of America*, vol. 133, no. 4, Apr. 2013, pp. 2036–2045, 10.1121/1.4792147. Accessed 26 Oct. 2022.

[5] electronics notes. “Carbon Microphone | Electronics Notes.” *Electronics-Notes.com*, 2019, [www.electronics-notes.com/articles/audio-video/microphones/carbon-microphone.php](http://www.electronics-notes.com/articles/audio-video/microphones/carbon-microphone.php). Accessed 26 Oct. 2022.

[6] Robjohns, Hugh. *A Brief History of Microphones*. 2006. Rycote, 2001, [aaron.headwai.com/344/Reading/Part%203%20%E2%80%93%20Microphone%20Theory%20and%20Techniques/Required/Part%203A/A%20brief%20history%20of%20microphones.pdf](http://aaron.headwai.com/344/Reading/Part%203%20%E2%80%93%20Microphone%20Theory%20and%20Techniques/Required/Part%203A/A%20brief%20history%20of%20microphones.pdf). Accessed 26 Oct. 2022.

[7] Bellis, Mary. “The History of Microphones.” *ThoughtCo*, 10 Aug. 2019, [www.thoughtco.com/history-of-microphones-1992144#:~:text=1983%3A%20Sennheiser%20developed%20the%20first](http://www.thoughtco.com/history-of-microphones-1992144#:~:text=1983%3A%20Sennheiser%20developed%20the%20first). Accessed 26 Oct. 2022.

[8] Singh, Abhishek. “Raspberry Pi 4 Specifications pin Diagram and Description».” *Hackatronic*, 12 Nov. 2021, [www.hackatronic.com/Raspberry-pi-4-specifications-pin-diagram-and-description/](http://www.hackatronic.com/Raspberry-pi-4-specifications-pin-diagram-and-description/). Accessed 29 Oct. 2022.

[9] Raspberry Pi Staff. “Raspberry Pi Documentation.” *Www.Raspberrypi.com*, 2012, [www.Raspberrypi.com/documentation/](http://www.Raspberrypi.com/documentation/). Accessed 29 Oct. 2022.

[10] Ltd, Raspberry Pi (Trading). “Raspberry Pi OS.” *Raspberry Pi*, [www.Raspberrypi.com/software/](http://www.Raspberrypi.com/software/). Accessed 29 Oct. 2022.

[11] Ltd, Raspberry Pi (Trading). “Buy a Raspberry Pi 3 Model B+.” *Raspberry Pi*, [www.Raspberrypi.com/products/Raspberry-pi-3-model-b-plus/](http://www.Raspberrypi.com/products/Raspberry-pi-3-model-b-plus/). Accessed 30 Oct. 2022.

[12] Nvidia Staff. “Jetson Nano Developer Kit.” *NVIDIA Developer*, 6 Mar. 2019, developer.nvidia.com/embedded/jetson-nano-developer-kit. Accessed 30 Oct. 2022.

[13] Ramirez, Ruzell. “How to Power Your Raspberry Pi with a Battery.” *Circuit Basics*, 29 July 2021, www.circuitbasics.com/how-to-power-your-Raspberry-pi-with-a-lithium-battery/#:~:text=Connect%20a%20TP4056%20charge%20controller. Accessed 1 Nov. 2022.

[14] Cawley, Christian. “3 Raspberry Pi Battery Packs for Portable Projects.” *MUO*, 14 Dec. 2019, www.makeuseof.com/tag/pi-go-x-ways-powering-Raspberry-pi-portable-projects/. Accessed 1 Nov. 2022.

[15] pratiksha6294. “Difference between Python and C++.” *GeeksforGeeks*, 30 Mar. 2020, www.geeksforgeeks.org/difference-between-python-and-c/. Accessed 1 Nov. 2022.

[16] Cheever, Erik. “Linear Physical Systems - Erik Cheever.” *Swarthmore.edu*, 2019, lpsa.swarthmore.edu/Fourier/Xforms/FXformIntro.html. Accessed 2 Nov. 2022.

[17] Cheever, Erik. “Introduction to the Fourier Series.” *Lpsa.swarthmore.edu*, 2005, lpsa.swarthmore.edu/Fourier/Series/WhyFS.html#Audio\_Signals. Accessed 2 Nov. 2022.

[18] Chaudhari, Qasim. “DFT Examples.” *Wireless pi*, 12 Aug. 2016, wirelesspi.com/dft-examples/. Accessed 2 Nov. 2022.

[19] Arar, Steve. “An Introduction to the Discrete Fourier Transform.” *Allaboutcircuits.com*, 20 July 2017, www.allaboutcircuits.com/technical-articles/an-introduction-to-the-discrete-fourier-transform/. Accessed 2 Nov. 2022.

[20] “Fourier Transform.” *Www.thefouriertransform.com*, 2010, www.thefouriertransform.com/. Accessed 2 Nov. 2022.

[21] DiCola, Tony. “FFT: Fun with Fourier Transforms.” *Adafruit Learning System*, 3 Oct. 2013, learn.adafruit.com/fft-fun-with-fourier-transforms/background. Accessed 2 Nov. 2022.

[22] Weisstein, Eric W. “Fourier Transform.” *Mathworld.wolfram.com*, mathworld.wolfram.com/FourierTransform.html. Accessed 2 Nov. 2022.

[23] Weisstein, Eric W. “Fourier Series.” *Mathworld.wolfram.com*, 2022, mathworld.wolfram.com/FourierSeries.html. Accessed 2 Nov. 2022.

[24] Chaudhary, Kartik. “Understanding Audio Data, Fourier Transform, FFT, Spectrogram and Speech Recognition.” *Medium*, 10 Mar. 2020,

towardsdatascience.com/understanding-audio-data-fourier-transform-fft-spectrogram-and-speech-recognition-a4072d228520. Accessed 2 Nov. 2022.

[25] Khillar, Sagar. “Difference between FFT and DFT | Difference Between.” *Http://Www.differencebetween.net/*, 29 Sept. 2021, [www.differencebetween.net/technology/difference-between-fft-and-dft](http://www.differencebetween.net/technology/difference-between-fft-and-dft). Accessed 4 Nov. 2022.

[26] ---. “USB.” *Wikipedia*, Wikimedia Foundation, 2 May 2019, [en.wikipedia.org/wiki/USB](https://en.wikipedia.org/wiki/USB). Accessed 1 Nov. 2022.

[27] ---. “Phone Connector (Audio).” *Wikipedia*, 28 Dec. 2020, [en.wikipedia.org/wiki/Phone\\_connector\\_\(audio\)](https://en.wikipedia.org/wiki/Phone_connector_(audio)). Accessed 1 Nov. 2022.

[28] ---. “XLR Connector.” *Wikipedia*, 14 Apr. 2021, [en.wikipedia.org/wiki/XLR\\_connector](https://en.wikipedia.org/wiki/XLR_connector). Accessed 1 Nov. 2022.

[29] MIPI Staff. “MIPI Display Serial Interface (MIPI DSI).” *MIPI*, MIPI Alliance, 9 Dec. 2016, [www.mipi.org/specifications/dsi](http://www.mipi.org/specifications/dsi). Accessed 1 Nov. 2022.

[30] Hardaway, Grant. “LibGuides: Engineering Subject Guide: Standards.” *Libguides.trinity.edu*, 11 July 2022, [libguides.trinity.edu/engr/standards](http://libguides.trinity.edu/engr/standards). Accessed 1 Nov. 2022.

[31] Woodgate, John. “The Basics of Electrical Engineering Standards - Technical Articles.” *Www.allaboutcircuits.com*, 24 June 2022, [www.allaboutcircuits.com/technical-articles/the-basics-of-ee-standards/](http://www.allaboutcircuits.com/technical-articles/the-basics-of-ee-standards/). Accessed 1 Nov. 2022.

[32] ---. “SD Card.” *Wikipedia*, 14 Sept. 2020, [en.wikipedia.org/wiki/SD\\_card](https://en.wikipedia.org/wiki/SD_card).

[33] ElProCus Staff. “LCD - What Is LCD: Construction and Working Principles of LCD Display.” *ElProCus - Electronic Projects for Engineering Students*, 29 Nov. 2013, [www.elprocus.com/ever-wondered-lcd-works/](http://www.elprocus.com/ever-wondered-lcd-works/). Accessed 2 Nov. 2022.

[34] Wikipedia Staff. “Liquid-Crystal Display.” *Wikipedia*, 2 Nov. 2022, [en.wikipedia.org/wiki/Liquid-crystal\\_display#:~:text=A%20liquid%2Dcrystal%20display%20](https://en.wikipedia.org/wiki/Liquid-crystal_display#:~:text=A%20liquid%2Dcrystal%20display%20). Accessed 2 Nov. 2022.

[35] Kawamoto, Hirohisa. “The History of Liquid-Crystal Display and Its Industry.” *IEEE Xplore*, 1 Sept. 2012, [ieeexplore.ieee.org/document/6487587](http://ieeexplore.ieee.org/document/6487587). Accessed 2 Nov. 2022.

[36] Kayne, R. “What Is an Analog Monitor? (with pictures).” *EasyTechJunkie*, 20 Oct. 2022, [www.easytechjunkie.com/what-is-an-analog-monitor.htm](http://www.easytechjunkie.com/what-is-an-analog-monitor.htm). Accessed 2 Nov. 2022.

[37] Katzmaier, David. “QLED vs. OLED TVs: What’s the Difference Anyway?” *CNET*, 17 Oct. 2022,

[www.cnet.com/tech/home-entertainment/qled-vs-oled-samsung-and-lg-tv-technologies-explained/](http://www.cnet.com/tech/home-entertainment/qled-vs-oled-samsung-and-lg-tv-technologies-explained/). Accessed 2 Nov. 2022.

[38] U.S. Department of Energy Staff. “OLED Basics.” *Energy.gov*, [www.energy.gov/eere/ssl/oled-basics#:~:text=An%20OLED%20is%20a%20solid](http://www.energy.gov/eere/ssl/oled-basics#:~:text=An%20OLED%20is%20a%20solid). Accessed 3 Nov. 2022.

[39] Waveshare Staff. “5inch DSI LCD - Waveshare Wiki.” *Wwww.waveshare.com*, [www.waveshare.com/wiki/5inch\\_DSI\\_LCD](http://www.waveshare.com/wiki/5inch_DSI_LCD). Accessed 3 Nov. 2022.

[40] ---. “The History of Liquid-Crystal Display and Its Industry.” *IEEE Xplore*, 1 Sept. 2012, [ieeexplore.ieee.org/document/6487587](http://ieeexplore.ieee.org/document/6487587). Accessed 11 Nov. 2022.

[41] Baker, Bonnie. “How Delta-Sigma ADCs Work, Part 1.” *How Delta-Sigma ADCs Work, Part I*, vol. 1, no. 1, 1 July 2011, pp. 13–16, [www.ti.com/lit/an/slyt423a/slyt423a.pdf?ts=1668469784755&ref\\_url=https%253A%252F%252Fwww.google.com%252F](http://www.ti.com/lit/an/slyt423a/slyt423a.pdf?ts=1668469784755&ref_url=https%253A%252F%252Fwww.google.com%252F). Accessed 17 Nov. 2022.

[42] Black, Brian. “Analog-To-Digital Converter Architectures and Choices for System Design | Analog Devices.” *Wwww.analog.com*, [www.analog.com/en/analog-dialogue/articles/analog-to-digital-converter-architectures-and-choices.html](http://www.analog.com/en/analog-dialogue/articles/analog-to-digital-converter-architectures-and-choices.html). Accessed 17 Nov. 2022.

[43] Koichi Hamashita Asahi-Kasei Microdevices Corp. *Delta Sigma ADC and DAC for Hi-Fi Audio, and next Trend Koichi Hamashita Asahi-Kasei Microdevices Corp.*

[44] Herd, Bil. “Tearing into Delta Sigma ADC’s.” *Hackaday*, 7 July 2016, [hackaday.com/2016/07/07/tearing-into-delta-sigma-adcs-part-1/](http://hackaday.com/2016/07/07/tearing-into-delta-sigma-adcs-part-1/). Accessed 16 Nov. 2022.

[45] Herd, Bil. “Tearing into Delta Sigma ADC’s.” *Hackaday*, 11 August 2016, [hackaday.com/2016/08/11/tearing-into-delta-sigma-adcs-part-2/](http://hackaday.com/2016/08/11/tearing-into-delta-sigma-adcs-part-2/). Accessed 16 Nov. 2022.



- [46] Shahramian, Shahriar. “TSP #32 – Tutorial on the Theory, Design and Measurement of Delta-Sigma Analog to Digital Converters | the Signal Path.” *Thesignalpath.com*, 3 Aug. 2014, [thesignalpath.com/blogs/videofeed/tsp-32-tutorial-on-the-theory-design-and-measurement-of-delta-sigma-analog-to-digital-converters/](https://thesignalpath.com/blogs/videofeed/tsp-32-tutorial-on-the-theory-design-and-measurement-of-delta-sigma-analog-to-digital-converters/). Accessed 16 Nov. 2022.
- [47] All About Circuits. “Delta-Sigma ADC | Digital-Analog Conversion | Electronics Textbook.” *Www.allaboutcircuits.com*, [www.allaboutcircuits.com/textbook/digital/chpt-13/delta-sigma-adc/](http://www.allaboutcircuits.com/textbook/digital/chpt-13/delta-sigma-adc/). Accessed 16 Nov. 2022.
- [48] L, Alberto. “Delta-Sigma ( $\Delta\Sigma$ ) and SAR ADC Converter in Microcontrollers.” *Mis Circuitos*, 16 Dec. 2017, [miscircuitos.com/delta-sigma-sar-adc-converter/](http://miscircuitos.com/delta-sigma-sar-adc-converter/). Accessed 16 Nov. 2022.
- [49] microchipdeveloper. “Sigma-Delta ADC Compared to SAR ADC - Developer Help.” *Microchipdeveloper.com*, [microchipdeveloper.com/32arm:samc21-sdadc-vs-saradc](http://microchipdeveloper.com/32arm:samc21-sdadc-vs-saradc). Accessed 16 Nov. 2022.
- [50] Smith, Elliott. “Understanding the Successive Approximation Register ADC.” *Allaboutcircuits.com*, All About Circuits, 28 Dec. 2015, [www.allaboutcircuits.com/technical-articles/understanding-analog-to-digital-converters-the-successive-approximation-reg/](http://www.allaboutcircuits.com/technical-articles/understanding-analog-to-digital-converters-the-successive-approximation-reg/). Accessed 16 Nov. 2022.
- [51] mathworks. “N-Bit Successive Approximation Register (SAR) Based ADC - Simulink.” *Www.mathworks.com*, [www.mathworks.com/help/msblks/ref/saradc.html](http://www.mathworks.com/help/msblks/ref/saradc.html). Accessed 16 Nov. 2022.
- [52] Kuphaldt, Tony R. “Flash ADC.” *Allaboutcircuits.com*, All About Circuits, 17 Feb. 2015, [www.allaboutcircuits.com/textbook/digital/chpt-13/flash-adc/](http://www.allaboutcircuits.com/textbook/digital/chpt-13/flash-adc/). Accessed 16 Nov. 2022.
- [53] Beis, Uwe. “An Introduction to Delta Sigma Converters.” *Www.beis.de*, 27 Feb. 2020, [www.beis.de/Elektronik/DeltaSigma/DeltaSigma.html](http://www.beis.de/Elektronik/DeltaSigma/DeltaSigma.html). Accessed 16 Nov. 2022.
- [54] sciencedirect. “Dual Slope - an Overview | ScienceDirect Topics.” *Www.sciencedirect.com*, [www.sciencedirect.com/topics/engineering/dual-slope](http://www.sciencedirect.com/topics/engineering/dual-slope). Accessed 16 Nov. 2022.
- [55] Random Nerd Tutorials. “How Do RGB LEDs Work? | Random Nerd Tutorials.” *Random Nerd Tutorials*, 2 Feb. 2019, [randomnerdtutorials.com/electronics-basics-how-do-rgb-leds-work/](http://randomnerdtutorials.com/electronics-basics-how-do-rgb-leds-work/). Accessed 3 Dec. 2022.

## 11.2 List of Figures

Figure 1: Hardware Block Diagram.....	8
Figure 2 :Software Block Diagram.....	9
Figure 3: House of Quality.....	10
Figure 4: Prototype.....	11
Figure 5: Carbon Microphone Schematic.....	12
Figure 6: Carbon-Button Microphone Edison/Emile.....	13
Figure 7: Lobar/Shotgun Polar Pattern.....	14
Figure 8: Vidpro Xm-55 Shotgun Microphone.....	15
Figure 9: Korg Clip-on Contact Microphone.....	16
Figure 10: Audio-Technica ATM350UcW Clip-On Condenser Microphone.....	17
Figure 11: RGB LED Schematic.....	19
Figure 12: SMD LED Chip.....	20
Figure 13: COB LED Chip.....	21
Figure 13: COB vs SMD LEDs.....	22
Figure 14: Rechargeable Lithium Ion Battery Pack.....	23
Figure 15: 3.7V Lithium Battery Pack with Powerboost Cofiguration.....	23
Figure 16: Raspberry Pi Battery Pack.....	24
Figure 17: Raspberry Pi Battery Hat.....	25
Figure 18: Wall Charger for Raspberry Pi.....	25
Figure 19: DC-DC Converter.....	27
Figure 20: DC-DC Converter Schematic.....	28
Figure 21: Low Power Modes.....	29
Figure 22: Raspberry Pi 4 Schematics.....	30
Figure 23: Raspberry Pi Model B.....	31
Figure 24: Raspberry Pi Model B+.....	32
Figure 25: Nvidia Jetson Nano Dev Kit.....	33
Figure 26: LCD Display Structure.....	37
Figure 27: QLED Display Structure.....	38
Figure 28: OLED Display Structure.....	38
Figure 29: Bluetooth Transmitter and Receiver Boards.....	40
Figure 30: Bluetooth Audio Transmission Plug-ins.....	41
Figure 31: Wired Audio Transmission.....	42
Figure 32: Dual Slope ADC Block Diagram.....	43
Figure 33: 3-bit Flash ADC Block Diagram.....	44
Figure 34: 16-bit Pipelined ADC Block Diagram.....	45
Figure 35: Delta-Sigma ADC Block Diagram.....	46
Figure 36: Successive Approximation ADC Block Diagram.....	47
Figure 37: Lumi Keys.....	48
Figure 38: Razer Synapse Software.....	49
Figure 39: Razer Synapse Customization.....	49

Figure 40: Razer Audio Meter Setting.....	50
Figure 41: Scratch Programming.....	52
Figure 42: Fourier Series Equation.....	53
Figure 43: Fourier Series Approximation.....	54
Figure 44: Fourier Transform Equation.....	54
Figure 45: Fourier Transform.....	54
Figure 46: Discrete Fourier Transform Equation.....	55
Figure 49: Discrete Fourier Transform.....	55
Figure 50: Microphone Placement for Pianos.....	56
Figure 51: Transparent Electrical Guitar.....	57
Figure 52: Bach TR300H2 Student Bb Trumpet.....	58
Figure 53: Tama MS1412TSBK Marching Snare Drum.....	59
Figure 54: Clip-On Microphone.....	60
Figure 55. Circuit Schematic for 3-Section LEDs.....	60
Figure 56: LCD Display.....	62
Figure 57: Iteration One of the Box Design.....	64
Figure 58: SolidWorks Design for the Box Design Centerpiece.....	65
Figure 59: SolidWorks Design for the Box Design Frontpieces.....	66
Figure 60: SolidWorks Design fo the Box Design Rearpiece.....	67
Figure 61: BTF-Lighting 5050SMD.....	80
Figure 62: Lights Connected to the Raspberry Pi 4.....	81
Figure 63: Light Testing.....	82
Figure 64: Light Colors.....	82
Figure 65: Dense vs Regular Spaced LEDs.....	83
Figure 66: Raspberry Pi 4.....	84
Figure 67: ADS1115 PCB.....	87
Figure 68: Pre-amp PCB.....	88
Figure 70: Temporary Sourced Snare Drum.....	89
Figure 71: Temporary Sourced Trumpet.....	90
Figure 72: Temporary Sourced Acoustic Guitar.....	91
Figure 73: Vibrating Table.....	93
Figure 74: Heat Damage on LCD Displays.....	95
Figure 75: Infrared Thermometer.....	96
Figure 76: Spectrum Analyzer.....	97
Figure 77: PCB Zoning.....	98
Figure 78: Hydraulic Press.....	99
Figure 79: Original ADC Design Testing.....	102
Figure 80: Second ADC Design Testing.....	103
Figure 81: ADS1115 Testing.....	103
Figure 82: Light Placement-Drum 1.....	104
Figure 83: Lighting Placement-Drum 2.....	104
Figure 84: Prototype 1.....	105

Figure 85:Second Order Delta-Sigma ADC.....	109
Figure 86: Noise Shaping Difference between Orders for Sigma-Delta	
ADC.....	109
Figure 87: Second Order Sigma-Delta ADC Schematic.....	110
Figure 88: Simulation Results for Sigma-Delta Modulator.....	111
Figure 89: Noise Shaping of Sigma Delta.....	113
Figure 90: Second Design Sigma-Delta ADC Schematic.....	114
Figure 91: ADS1115 Schematic.....	114
Figure 92: DC-DC Converter.....	115
Figure 93: Pre-Amplifier Schematic.....	116
Figure 94: Overall Schematic.....	116
Figure 95: Program Landing Screen.....	117
Figure 96: Theme Selection Screen.....	118
Figure 97: Color Selection Menu.....	119
Figure 98:Program Active.....	120

### 11.3 List of Tables

Table 1: Requirements and Specifications.....	6
Table 2: Microphone Comparisons.....	18
Table 3: LED Comparisons.....	21
Table 4: Power Supply Comparisons.....	26
Table 5: Microcontroller Comparisons.....	27
Table 6: LCD vs OLED.....	39
Table 7: Audio Transmission Comparisons.....	42
Table 8: ADC Comparisons.....	43
Table 9: Software Comparisons.....	52
Table 10: First ADC Budget.....	122
Table 11: Second ADC Budget.....	
Table 12:Overall Budget.....	
Table 13: Milestones Fall 2022.....	123
Table 14: Milestones Spring 2023.....	124