Chromesthesia Simulation Device

Senior Design II Final Report

Department of Electrical Engineering and Computer Science

University of Central Florida

Dr. Lei Wei

Group 27

Angel Garcia	Computer Engineer
Wesley Ellery	Computer Engineer
Nicholas Alban	Electrical Engineer
Brooke Roeder	Electrical Engineer

1 Contents

Table Index

Figure Index

2 Executive Summary

This project was to design a device that would take audio and visual inputs and produce a visual output that simulates/recreates the visual experiences that a person with chromesthesia would experience. The way it works is by combining audio input devices (i.e., microphones) and an optical capture device (i.e., a camera) on a single platform that can be oriented and pointed in any direction. This platform receives audio and visual inputs from the direction it is pointing, and then perform a computation process to recreate the same imagery from the visual input edited with visual effects influenced by the audio input. The visual output gets displayed to a screen for the user to view. The platform containing the audio and visual input devices, the PCB/computational hardware, and the display are all integrated together into one device that the user can carry and thus, aim the microphones and camera in any direction and receive a corresponding visual output, as if looking around with their own eyes and experiencing chromesthesia.

Chromesthesia is an audio-visual variant of the phenomenon called "Synesthesia". Synesthesia is when the stimulation of one sensory or cognitive pathway leads to involuntary experiences in a second sensory or cognitive pathway. Chromesthesia is the involuntary association of sounds to visual effects such as shapes and colors.
The design for this project incorporates many different components including software and hardware which work closely together to perform the desired task. The software is extremely important in communicating to the hardware so that everything functions accordingly. The components picked for the project were chosen based on cost, efficiency, and availability. Due to COVID and certain shortages, we took availability of certain components into consideration because the pandemic caused components to take longer to ship or just be completely unavailable.

The goal was to make a user-friendly device that is efficient, compact, and inexpensive. It is comfortable to wear and should only be where the user has to look around and aim with their head to operate the device. The device is portable and easy to operate for the user. The design incorporates some sort of microphone array and a camera to collect audio and visual inputs. These inputs are fed into a microcontroller unit on a printed circuit board. The microcontrollers perform logic and other audio and image processing functions to then produce one single output in the form of an image which is identical to the input image that the camera captured, except that it is altered and edited with visual effects that were influenced by the audio inputs. Depending on the audio inputs, variances in frequency and location will determine the visual effects. This produces an experience that simulates the phenomenon of Chromesthesia. Although many people experience Chromesthesia with varying degrees of difference when questioning each individual person who has the Phenomenon with their brain, this device finds a general neutral ground that can portray the phenomenon in a way that is not necessarily accurate, but entertaining and accurate enough to convey the concept to the people who are using the device.

3 Project Description

For the project description section, we are going over the basics of our project and its description. The general idea of this chapter will be going over the details of our project. The Chromesthesia simulation device or CSD allows users to simulate the phenomena by using a camera, point it in any direction, and the simulation result will be displayed on a screen. We hope by creating a device like this it will help allow users to dive into the world of synesthetes at least for a moment. We are going through our motivation, goals, design, software, and the foundations of our project in the following parts.

3.1 Existing Similar Projects and Products

In our current times, there are a lot of advanced technologies out there that we as humans get to use with our daily lives, so our project of making a device to simulate a phenomenon is definitely not enough to change the world. We have smart computers in our pockets, portable computers, and tablets that we take with us anywhere we want. We have the ability to call and have video chats with friends, family, and even strangers all across the globe. There are gaming systems where you put on a set of virtual reality goggles, and you are completely immersed into countless worlds. As a group, we used these advanced features and ideas and built from them and continued to develop and potentially take things a step further.

3.1.1 Sound Color Project

The Sound Color Project is a web app that was created by Kevin Groat and Derek Torsani. The two had the project featured at SoundScene2020. The DC Listening Lounge, an audio collaborative of Washington, DC-based sound artists and lovers, hosts Sound Scene, an annual free and interactive audio arts festival. This all-ages public festival, now in its 14th year, honors the talent of local and international artists. Sound Scene explores acoustics, beat-making, sound production, live performances of dance, music, and spoken word, and small group workshops. Sound Scene has been presented as a two-day full museum takeover by the Smithsonian Hirschhorn Museum and Sculpture Garden since 2016.

The Sound Color Project began as an experiment to see whether there were any ways to make music more accessible by using visual elements like light, color, and texture. The project's objective of studying multimodal accessibility continues to evolve as the project progresses. The web app analyzes frequency and volume using an audio input source such as a device's microphone or a USB audio interface. These two variables are then transformed into light, color, and texture using a predetermined color pattern and mapping audible into visual qualities. The web app was built on JavaScript, and it utilizes a Web Audio API. In regard to keeping any of the information that it requires, it does not store or transmit audio, it just uses these values to drive the colors onto the screen. What is interesting about this web app is the developers figured out 5 different ways they could express chromesthesia and made them into color presets into the web application, as well as letting the user create their own custom settings. These five presets are as follows: Chromesthesia, Chakras, Emotion, Chromotherapy and Adolescence.

To explain one of the presets in more depth, the Chakra preset refers to the energy points in the body that are widely used to achieve spiritual or conscious balance through meditation. The colors of the rainbow represent each of the chakras: base, sacral, solar plexus, heart, throat, third eye, and crown, with red at the bottom (root) and violet at the top (crown) as shown in Figure 1. Every chakra, or energy center in the body, is tuned to a specific frequency or note, which is usually accessed using singing bowls. The rainbow's spectrum of colors represents the order to which we must return in the body via the practice of inner self attention. To add the cherry on top, they made it so any user that owns Philips Hue lights could connect to the web app and have the lights simulate the colors that correspond to each preset.

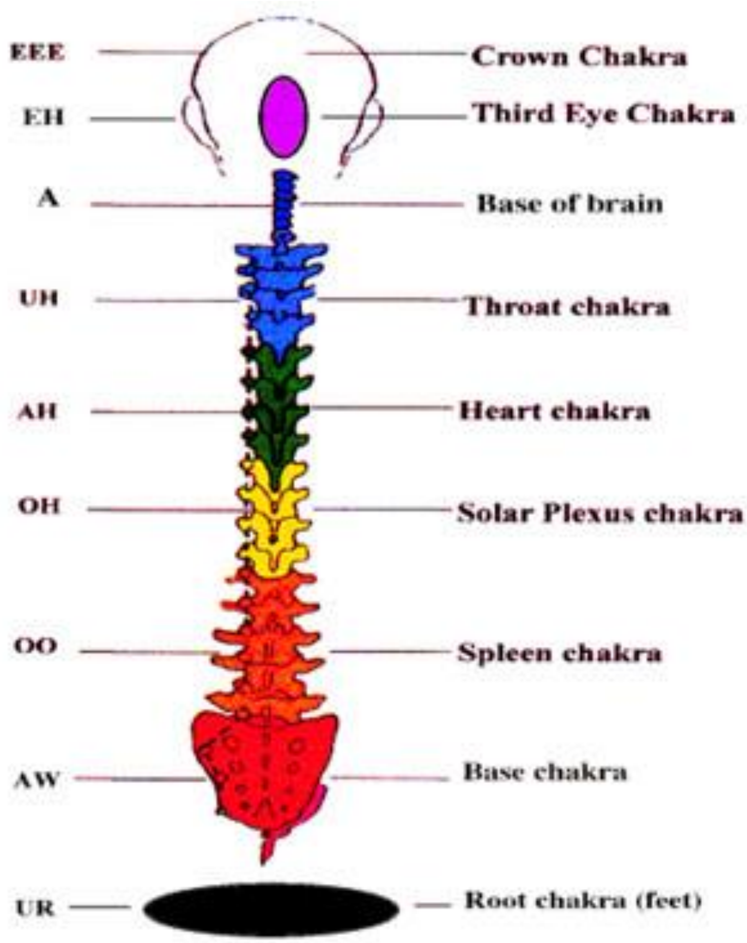


Figure 1: Vowel Sounds and Colors for Toning based on the Chakra System

3.1.2 E-MOCOMU (E-Motion, Color, and Music)

The E-MOCOMU is a chromesthesia prototype created by Ellena Partesotti, who is a music and technology PhD student from the University of Valladolid, which is a public university located in Spain. Her idea for this project is the same as ours. Can we create a device that can help people who are not synesthetes experience this phenomenon? One of the possibilities of her device is to leverage the therapeutic potential of the three dimensions of color, music, and movement to improve communication in autistic children, for example. By moving around in space freely (Figure 2), the E-MOCOMU device lets users change sounds, pictures, and colors. Essentially if someone moves their arm, there will be sounds, colors, and images that will be associated to the action that are displayed onto a screen. This technology can be used in the field of physical rehabilitation to help with normal activities, as well as in the field of psychology to help with the treatment of various diseases. Ellena's device is based on Kinect, which is a technology that allows players to operate and interact with the console without having to maintain physical touch with a standard video games joystick. Ellena points out that her device "has been designed as an accessible low-cost technology that anyone can have at home, as it can be used for play in a family context. We have created different programs to integrate the visual and sound elements."

Figure 2: A User Tests E-MOCOMU

3.2 Project Considerations

It is critical to identify and comprehend what a project entails before choosing on which project to pursue. A decision matrix can be used to gather needs by analyzing and ranking criteria, which can be important because undertaking a project of this size must be based on a number of factors. The two most important factors are money and time.

As soon as our group was confirmed, we took two days to meet with each other and discuss what would be the ideas moving forward. We each brought a different notion to the table when it came to deciding on our project. Each idea varied in difficulty, one was too simple, another was difficult, so as a group, we decided to brainstorm during our first meeting. When we began considering which ideas may work, we first considered whether they were practical and feasible. Putting a limit on our budget was something we had to keep in mind, so we had to make sure that the idea was chose was also in our budget range. Throughout the process of these discussions, we considered projects that contained a significant amount of design. Our primary goal as a team in Senior Design 1 was to construct a design-intensive project that could be created and completed over the span of two semesters. During our time spent in lectures, we came to learn of many other factors that must be taken into account when deciding an idea for the project such as sponsorship, social values, motivation, and educational roles.

3.3 Project Motivation

The motivation of this project was mainly led by 2 things. The first motivation was that this was simply a compelling technological device to create. Augmented reality is a up and coming field. It is very exciting to talk about, and this device technically could fall under the category of augmented vision. This could be considered augmented Vision because the device is assisting the user does operating it to change their own vision using the device. The device is not only conveying what is possible to be seen in the real world, but it is also applying to that image of the real world an unnatural effect.

This sounds like a very fun and enjoyable device to use and interact with from a user perspective. This motivation excites not only us, the developers, but should also excite the users. People like to get their hands on devices and experience what it is like to live in a completely different world other than their own. With this device, people will be able to experience their world in a whole new way.

The second main source of motivation is education. Chromesthesia is a real-life phenomenon that only a small percentage of people get to experience. There are other forms of it involving different senses and they are all extremely interesting, yet mostly unknown to the general population. When describing this project to colleagues and friends and family, many of them had never heard of the phenomenon before. Although this device can serve as entertainment, this device can also serve as an educational tool. Every person who interacts with this device and operates it will learn about the phenomenon of Chromesthesia. It is a very interesting phenomenon, and more people should know of its existence.

3.4 Goals and Objectives

This section outlines the main goals and objectives for this project. The main objective is to design an efficient consumer device that can simulate the chromesthesia phenomenon, so people can see how it is like for synesthetes. Secondary goals would include a low-cost design and good product longevity, allowing for a low-end pricing for the average user. The following will be the main ideas and concepts that will guide our design choices. They will constantly be referenced and used to determine important design choices such as what component to buy and what functions should be prioritized.

3.4.1 Accuracy

In order to simulate Chromesthesia, our device needed to be accurate in determining audio frequencies and be able to determine the exact location of the audio source. Any inaccuracies will cause incorrect visuals and disrupt the recreation of the phenomena. Processing of the audio inputs will be especially important.

3.4.2 Speed

The device needed to produce an image in real time. Ideally, the time between the user aiming the device and the device creating an output image is as short as possible to simulate "looking" with their own eyes. This means the processes are fast and efficient. This is heavily dependent on the software optimization and hardware design and quality.

3.4.3 Software

The software is able to communicate with the microprocessor and able to perform complicated algorithms including noise processing, 3D sound localization, and image processing. These algorithms must be as quick and efficient as possible to determine what color correlates to the audio frequency and how it should be displayed and produce a near real-time picture.

3.4.4 Hardware

The hardware consists of a microprocessor, multiple microphones, and a camera able to display images of quality 720p or higher. These are all key components to the Chromesthesia device. The hardware was chosen carefully to optimally perform the calculations needed to process audio and visual inputs so the device can operate at the desired speed.

3.4.5 Convenience

Ideally, we aimed for the final device to be lightweight and wearable. Our vision for the project was for it to be a pair of goggles or any other similar item that you can wear, and one would be able to see through them and experience Chromesthesia. Alternatively, the device is handheld and can be aimed with the user's hands like a camcorder. Both designs demanded ease of use with minimal inputs from the user (i.e., the user only needs to aim the device with their head or hands) and a light weight (ideally under 5 lbs.)

3.4.6 Experience

The device adequately represents the phenomenon of chromesthesia, and the final visual output has a semblance of the experience, as if "through the eyes" of someone with chromesthesia. Chromesthesia varies between people, and it has nuances in how it is experienced. Through our research, we got a better definition of how exactly we would represent chromesthesia visually on the display. Ultimately, we hope we have recreated the phenomenon in a way that has the perfect balance between accurate and fun. People should be excited to use the device and want to experiment aiming it at different sound sources with different pitches.

3.5 Features

In the beginning, we had plans for our device to have three defined features. These features were classified by their complexity. They were categorized as basic, extended, and advanced. The subsections will go into detail what each feature classification means and why it is important to implement.

3.5.1 Basic Features

This section defines the minimum functionality of the device. For this project, there were two basic features; one was the ability to process audio inputs with a microprocessor and perform a basic algorithm that can display colors correlating to the frequency of the soundwaves from the audio input. The second basic feature was the ability to capture visual images in real time, parallel with detecting and processing the audio inputs.

As can be seen in the following figures, there are two separate instances. These examples provide an outlook as to what the desirable results would be. The two systems, i.e. the audio processing and the visual processing units are both operating at the same time, but they are independently collecting data and Performing tasks. This is the minimum functionality of the device. Without this function, the device simply could not work or deliver on its more advanced features.

PURE SONOCHROMATIC SCALE		
(invisible)	Ultraviolet	Over 717.591 Hz
	Violet	607.542 Hz
	Blue	573.891 Hz
	Cyan	551.154 Hz
	Green	478.394 Hz
	Yellow	462.023 Hz
	Orange	440.195 Hz
	Red	363.797 Hz
(invisible)	Infrared	Below 363.797 Hz

SONOCHROMATIC MUSIC SCALE (basic 12/360)		
	Rose	**E**
	Magenta	**D#**
	Violet	**D**
	Blue	**C#**
	Azure	**C**
	Cyan	**B**
	Spring	**A#**
	Green	**A**
	Chartreuse	**G#**
	Yellow	**G**
	Orange	**F#**
	Red	**F**

Figure 3: Example of correlation between wavelength of sound and color

Figure 4:Example of captured image from camera

3.5.2 Extended Features

This section builds upon the basic features and adds some sophistication to the device's abilities. The next logical step from the aforementioned basic features was to create some sort of neural network or other algorithm to better process the audio inputs, allowing the device to perform 3D sound localization and combat noise. Additionally, a visual output was to be produced, which combines the raw visual input from the camera with the color determined by the algorithm that uses the frequency to determine the color. Figure 5 is a example of what the output would be if the audio input's frequency was calculated to correspond to the color blue.

The potential application of neural networks, though it would have been great to have as an extended feature, might have to turn into an advanced feature depending on time and budget constraints as the timeline of the project progresses. The implementation of advanced features allows this project to expand its scope further without taking away essential features that need to be included. The possible applications of a neural network and machine learning are further explored and discussed in the next section – advanced features. With the shortages of parts and components, extending the basic features of this device became extremely complicated.

Figure 5: Example of outputting color of detected sound on same output image as visual input

3.5.3 Future Development

Advanced features go beyond what generally is the fundamental functionality of the device, further sophisticating existing features, or adding additional advanced features. Here, we considered using 3D sound localization to determine the spatial location of the sound sources in the acquired image and overlaying the corresponding frequency color to that location. Figure 6 demonstrates what an advanced future development might look like. This might not be a realistic example, but it demonstrates how different sounds will be coming from various locations and the frequency of those different sound sources would display a different color. Due to the limited budget and short development time, the figure below is only a representation of the complete capabilities of this device that one day could be realized.

Figure 6: Example of outputting color of detected sound on same output image dependent on location in image

There was some speculation as to potentially implementing neural networks and or machine learning as an advanced feature. Machine learning could be used in two ways: one way would be to use machine learning for the audio processing to further help improve the sensitivity and the ability to detect specific frequencies given noise levels that are considered challenging. Using a neural network, we could train it separately on a computer from the system using a database of inputs with known goals for measurements. We could train the neural network to better identify sounds and differentiate between frequencies of sounds amongst a large volume of noise. The benefits this would be that we don't need to create a robust algorithm by hand, instead, we could use the power of machine learning and neural networks to train our device to become better at identifying frequencies and filtering noise out from the inputs. Neural networks are a great way to demonstrate the full potential of what this device can do, but it would require a budget and a team that is well outside the scope of the initial prototype phase of this project.

Secondly, we could use machine learning or neural network training to identify imagery in the captured camera images in frames. This would allow us to refine our ability to identify common sound sources. For example, we could use a database of common sound sources such as humans faces, stereo speakers, cars, Etc. Essentially, things that are known to make noise. This could help refine our 3D sound localization and when it comes to driving and editing are imagery for the output image, we could cross reference this data with the data determined by the 3D sound localization algorithm to get an even more accurate or better representation of where the sound is originating from on the image. Using equipment that is sensitive to the surrounding environment at a larger scale is the path we hope to accomplish in the future with a secondary prototype well after the semester has ended.

Another way is that we can use machine learning for the audio processing to further help improve the sensitivity and the ability to detect specific frequencies given noise levels that are considered challenging. Using a neural network, we could train it separately on a computer from the system using a database of inputs with known goals for measurements. We could train the neural network to better identify sounds and differentiate between frequencies of sounds amongst a large volume of noise. The benefits this would be that we don't need to create a robust algorithm by hand, instead, we could use the power of machine learning and neural networks to train our device to become better at identifying frequencies and filtering noise out from the inputs.

Furthermore, machine learning could be used to create even more advanced location-based visual effects based on sound inputs and frequencies. For example, instead of just creating a sphere that forms around the Sound Source that is colored in a way that relates to the specific frequency that is being detected in that spot, it could specifically highlight the exact Sound Source and change the color of it. For example, if someone is talking at the camera with a specific frequency, it could instead of creating a sphere around the person's face or color the face itself with the color that correlates to that frequency. With the success of this initial prototype, we feel that this design should be a case study in using machine learning algorithms.

3.6 Requirements and Specification

The major goal of this section is to break down the project and its design into various pieces and explain what each component's goal is. The actual hardware section and the software section are the two key components. We built the groundwork for the project by breaking down the needs and specifications. Table 3 contains all the constraints and requirements that need to be met and considered during the development of the project. These requirements were all taken into consideration for the details discussed in the sections for Designs A and B, and the goals and objectives of the device.

Regarding our time, as a team we met once or twice a week to discuss the progress of the project. All the team members were free in the evenings during the weekdays so we got together and had meetings that began after 6:00PM. What was discussed in the first meeting of the week were the goals to be set for individual group members and the second meeting was a recap of what was done and what still needed to be done, and from there we moved forward using this method. Throughout the week we communicated throughout the week so if anything happened it was understood between everyone and allowed us to account for time that was missed, while keeping in mind everything that needed to be done and continue moving forward with our schedule. In the first couple of weeks, we got together as a group and assigned individual tasks to be responsible for in the following weeks that way everyone was held accountable. Our first goal was to have a 60-page research paper ready by the beginning of November, and then afterwards have a 100-page research paper by the end of the semester for Senior Design 1. The following major due date after that was having a completed project and research paper, and successfully complete our demo by the end of the semester for Senior Design 2.

Our costs varied as the weeks went by due to prices increasing because of many shortages happening at the moment post-Covid. We agreed to keep the budget around $100 per person which was enough to make our chromesthesia simulation device. We understood that given our current situation and prices fluctuating, it may or may not have been a bit more expensive as we progressed into our project. As a group, we decided together what parts needed to be purchased, replaced, etc. and if the situation arose where we go over budget, that was also discussed as a group to figure out a plan moving forward.

In terms of our scope, we wanted to successfully create a chromesthesia simulation device which allowed users to simulate the phenomena by using a camera, point it in any direction, and the simulation result will be displayed on a screen. For the device's quality, we expected it to be cost effective, portable, durable, and easy to look at with no safety hazards to potentially harm the user. A risk we take is potentially having something hurt the user since they will be wearing the device, or potentially being shocked from a short circuit.

Components	Specifications
Wearable Devices	• This piece of equipment should be as lightweight as possible. • Industry VR headsets range from 0.84 lbs. to 2.5lbs. We can arbitrarily set a limit to double their standards: 5lbs. • Our aim in weight limitations will be 5lbs +/- 1lbs.
Microphones	• I2S Microphones • Humans can hear 20Hz-20kHz, so the equipment needs to be able to detect similar range
Camera	• Only one camera needed • Resolution should be at least 720p
Microcontroller	• Needs to be able to do necessary calculations for our algorithms for audio and image processing • Raspberry Pi 4 • MSP430FR6989 • STM32F3 • BPI-M2 Zero

Display	• Needs to be identical resolution to the camera • Small size if device is wearable, full sized if stationary
3D Printed Platform	• Will hold the microphones and camera

Table 1: Physical Hardware Requirements and Specifications

For the microphones, noise is a big factor. Audio processing is arguably be the most important factor in our signal processing ability of our device. Additionally, microphones come in a variety of “cones”. Since we have an aimable device, we needed to use microphones with a narrower cone. Ideally, the microphones will have a field of view identical to the camera. For the production of our project, after getting together once or twice, our group concluded that the estimated cost of building the device should fall under $600. The main components that took a bigger chunk of the budget would be the microcontrollers, due to most of them being in the range of $50 to $100 or more, and most likely the display and cameras. Unfortunately, we were not able to get to our advanced design, but if we could have, the additional cost of a VR headset is shown in Table 1, which would also be another considerable amount of money.

Components	Specifications
Audio Spectrum Analysis	• Needs to process audio input signals from microphone • Determine the frequency of audio received
Noise Cancellation	• Will take audio input and perform noise cancellation and DSP to reduce noise on the input signal
Frequency Calculation	• Simple Algorithm • Will take frequency of audio input and correlate frequency quantity to a specific RGB value
3D Sound Localization	• Using microphone inputs, this will determine the exact angle direction from center point of device • Should coincide with location of camera • Should deliver coordinates along the display screen • Should represent which pixels on the display are visually capturing the audio source

Video Processing	• Input from camera should be processed by MCU • Be able to translate input directly to an output pixel display
Image Editing	• Take the image and apply a transform of the RGB value of identified pixels received from frequency calculator, video image, and 3D sound localization.

Table 2: Software Requirements and Specifications

Table 3 concludes section 2.4, which consists of the "what's" and a rough idea of the "where's" of the project along with its design being broken down and we go in depth of what each component should be doing. The following section 2.6 will feature Figure 7, which is our House of Quality. To simulate Chromesthesia, our device needs to be accurate in determining audio frequencies and be able to determine the exact location of the audio source. Any inaccuracies will cause incorrect visuals and disrupt the recreation of the phenomena. Processing the audio inputs will be especially important. The device needs to produce an image in real time. Ideally, the time between the user aiming the device and the device creating an output image will be as short as possible to simulate "looking" with their own eyes. This means the processes need to be fast and efficient. This will be heavily dependent on the software optimization and hardware design and quality. The software must be able to communicate with the microprocessor and be able to perform complicated algorithms including noise processing, 3D sound localization, and image processing. These algorithms must be as quick and efficient as possible to determine what color correlates to the audio frequency and how it should be displayed and produce a near real-time picture. The hardware will consist of a microprocessor, multiple microphones, and a camera able to display images of quality 720p or higher. These will all be the key components to the Chromesthesia device. The hardware should be chosen carefully to optimally perform the calculations needed to process audio and visual inputs so the device can operate at the desired speed. Our long-term vision for the project is for it to be a pair of goggles or any other similar item that you can wear, and one would be able to see through them and experience Chromesthesia. Alternatively, the device will be handheld and aimed with the user's hands like a camcorder. Both designs demand ease of use with minimal inputs from the user (i.e., the user only needs to aim the device with their head or hands) and a light weight (ideally under 5 lbs.). While this is not part of the initial prototype requirements, the design consideration has helped make this device as compact as possible for the demonstrated prototype. The device should adequately represent the phenomenon of chromesthesia, and the final visual output should have a semblance of the experience, as if "through the eyes" of someone with chromesthesia. Chromesthesia varies between people, and it has nuances in how it is experienced. People should be excited to use the device and want to experiment aiming it at different sound sources with different pitches.

Component(s)	Parameter	Specification
Microphones	Audio input	Frequency Range 20Hz – 20kHz
Microphones and Microprocessor	Audio input and Processing	70% accuracy up to 5 meter range
Microprocessor	Processing Speed	Associate color with designated note for defined frequency range under 2 seconds (I.e Red is associated with the key of F, frequency range 363-440Hz
Camera	Power Consumption	6.7 to 10 watts power consumption
Screen Display	Resolution	At least 1280 x 720 pixels
Battery	Discharge Time	2 hours

Table 3: Core Engineering Specifications

3.7 House of Quality

The House of Quality, as shown in Figure 7, is an engineering development tool (HOQ). This methodology considered both engineering and customer tradeoffs, making it valuable for market considerations throughout the product life cycle. This tool can be utilized at many stages of the development process.

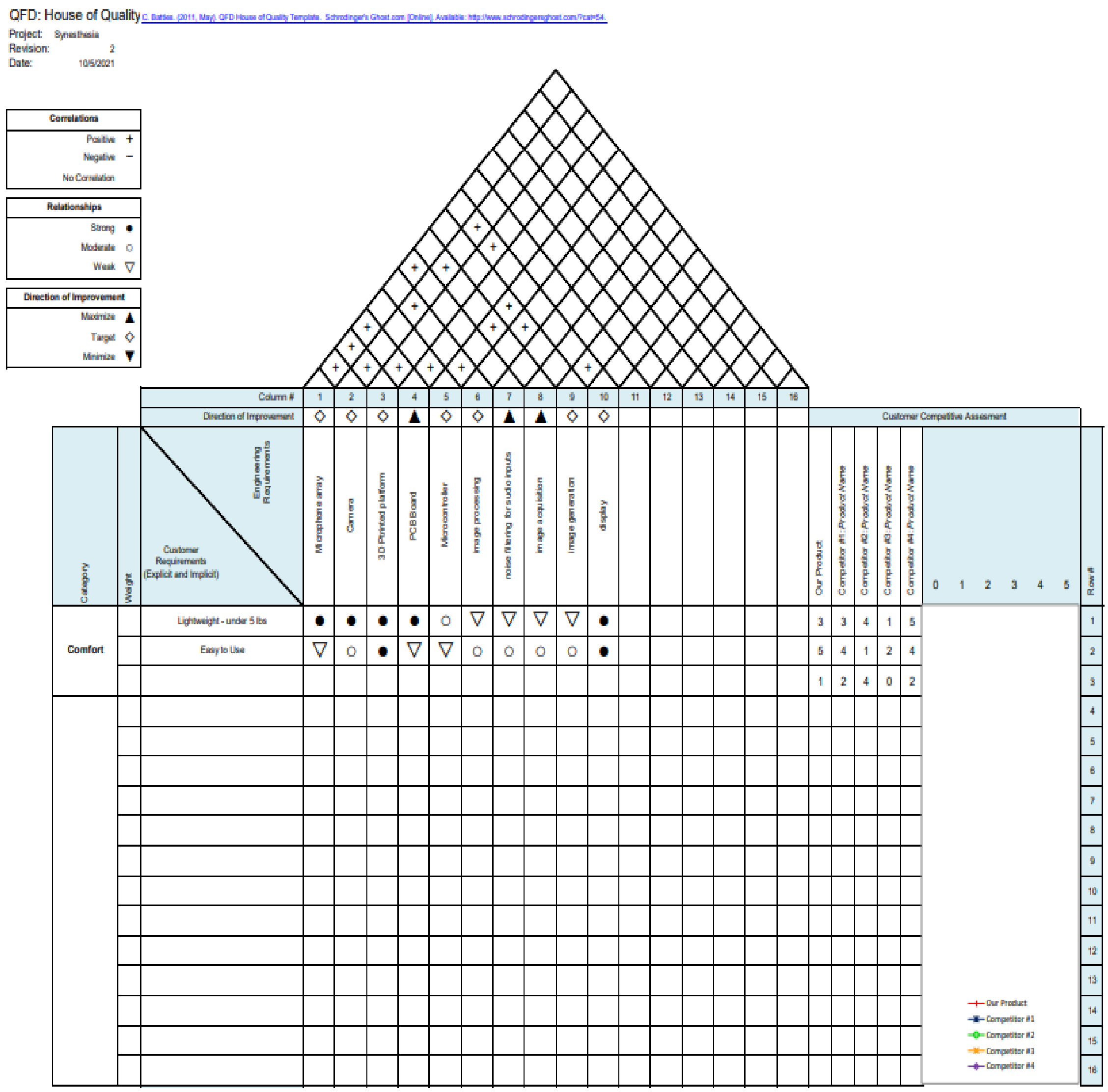


Figure 7: House of Quality

3.8 Design Overview

This section of the chapter focuses on the overview of our chromesthesia simulation device. We referenced this for multiple reasons, such as delegating work among group members, having a general roadmap of what needed to be done to achieve the final product, and for transparency in case someone fell behind, we could allocate our time and resources if an emergency appeared. Figure 8 shows our key for our block diagrams.

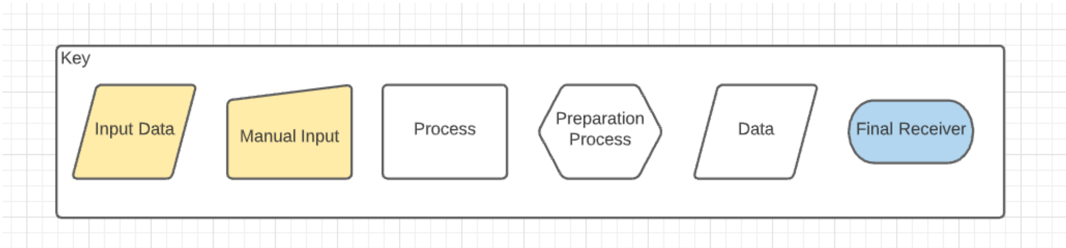


Figure 8: Block Diagram Key

3.8.1 Device Block Diagram Overview

Figure 9 below shows the basic overview of our chromesthesia simulation device. It takes three different kinds of inputs from visual/optical captures, audio sources, and directional aiming. The camera connected takes in the visual inputs while the microphone array receives inputs audio sources while also having the device aimed at specific directions. All of this information gets fed through the software algorithm that was developed and determines how far these frequencies that are being received are. It correlates where the audio is coming from the image, and then the algorithm sorts this information and applies the specified visual effect to the captured image. This result is displayed by a screen.

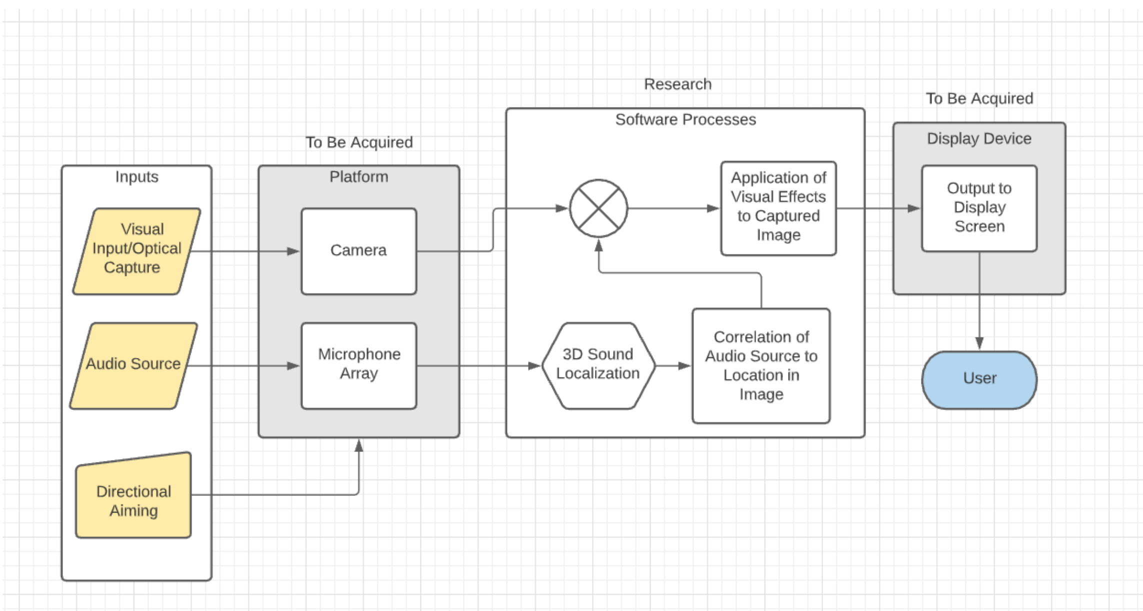


Figure 9: Device Block Diagram Overview

3.8.2 Software Block Diagram Overview

Figure 10 below shows an overview of our Frequency to Color software design. The program takes in a visual input and processes it and converts the data into a coordinate plane of the 2D image received. On the audio side of the software process, it takes the frequency values of the incoming audio signals, utilize 3D sound localization, and determines the location of the audio source. Once the program has those two pieces of information, it correlates the data and locations of the sources and displays the hue of the frequency matched onto a display screen.

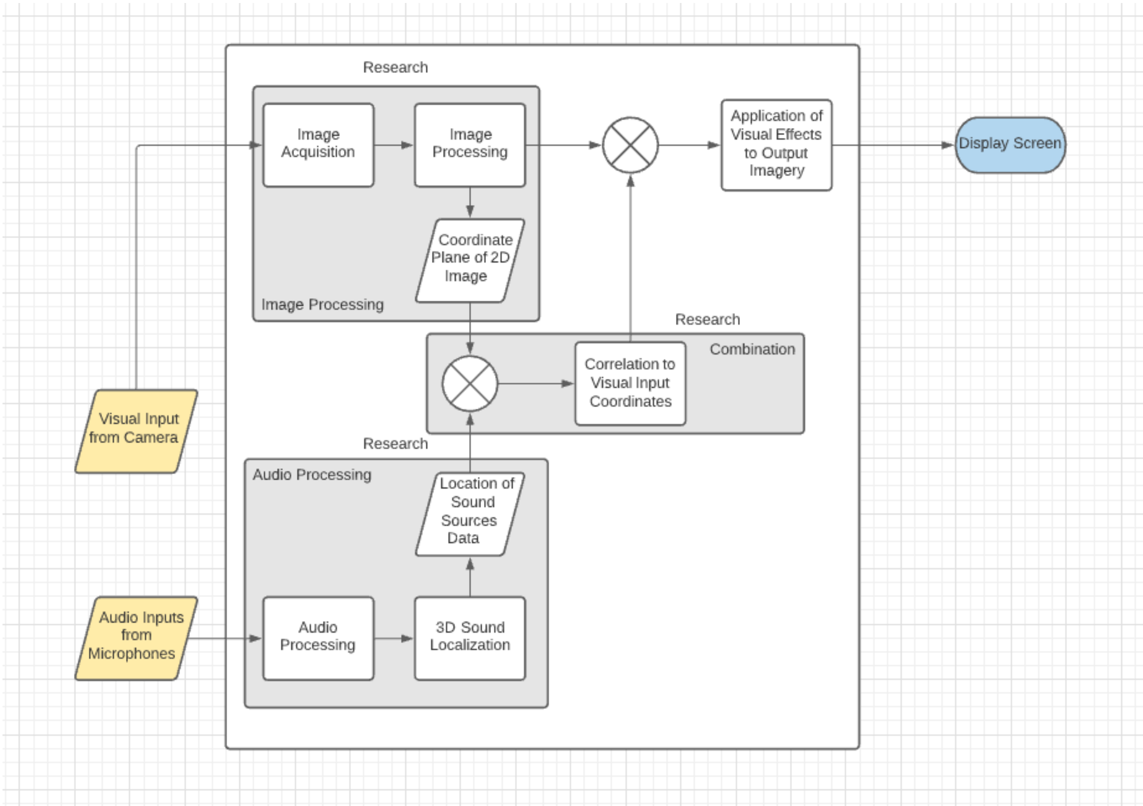


Figure 10: Software Block Diagram Overview

3.9 Designs

There are two designs that were considered for this project. Both designs are functionally the same; the device is aimed to collect audio and visual inputs, the inputs are processed, and then a visual output is generated and displayed in real time to allow the user to experience chromesthesia. Both designs satisfied the project goals and objectives, and have the same basic, extended, and advanced features. The main difference between the two designs is how the user interacts with the device.

3.9.1 Design A

A custom platform was made to hold the microprocessor, microphones, and camera. The platform was shaped for handheld use so the user can use their hands to aim it. Whichever direction the user aims the device, the microphones and camera would obtain inputs dependent on the direction of aiming. The microphones are positioned on the platform in a way that 3D sound localization can be performed. Once it processes the audio, the audio and visual inputs are sent to the microprocessor, which determines what color will correlate to the frequency of the audio input and generate an augmented visual output of the captured scene. The display screen is not fixed to the device. This was dependent on the weight and cost of the screen itself. A smaller screen would have been more expensive but potentially enable our device to fit onto one handheld platform. A larger, more traditionally sized screen would be less expensive, but heavier. A heavy screen will most likely need to be placed on a table. In the case of a stationary screen, the device will still function as intended, however it will diminish the experience of "looking" around with the device. Figure 11 is a rough sketch of our idea behind Design A.

Pros of Design A	Cons of Design A
Platform design will be more appropriate for our skill levels	Less immersive
Simpler platform design	Potentially heavier screen
Easier to pass between users and troubleshoot	

Table 4: Pros and Cons Table of Design A

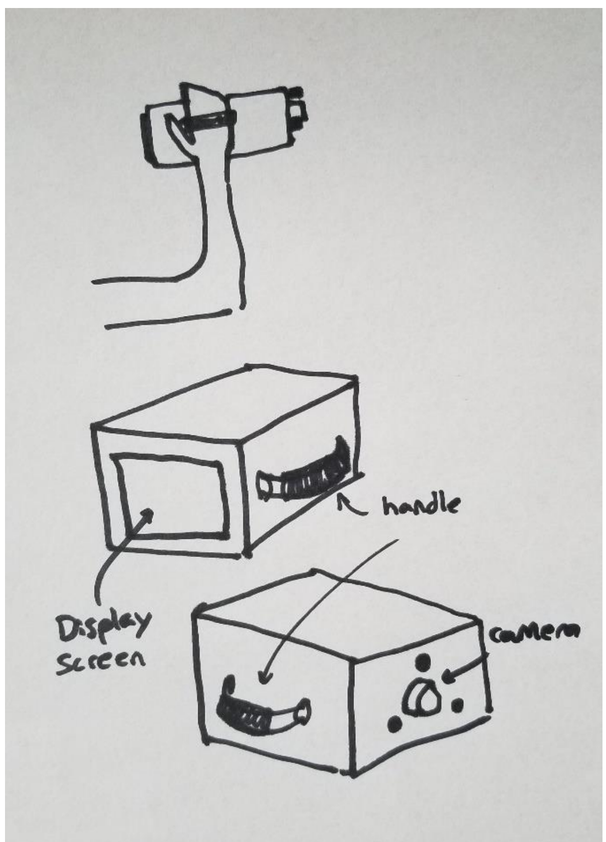


Figure 11: Design A Sketch

3.9.2 Design B

Design B was our aim, and it would have been pursued if everything was on time as expected. Everything from Design A would have been translated to the context of a pair of VR goggles or a sort of wearable vision device. Ideally, the device would have used a third-party wearable screen display and instead of the microphone/camera/PCB platform being designed to be handheld, it would have been attached to the goggles. This way, wherever the user physically aims their head, the device would collect audio and visual inputs from that direction. This would have been the ideal design for the device because it would have maximized the experience by making users feel like they are looking and experiencing chromesthesia with their own eyes and senses. Figure 12 is a rough sketch of our idea behind Design B which was more advanced.

Pros of Design B	Cons of Design B
Much more immersive	Need to interface with third party VR headset
No need to create ergonomic platform unlike Design A	More expensive (VR headset) and concerns with sanitization when users switch
	Potential issues connecting platform to headset

Table 5: Pros and Cons Table of Design B

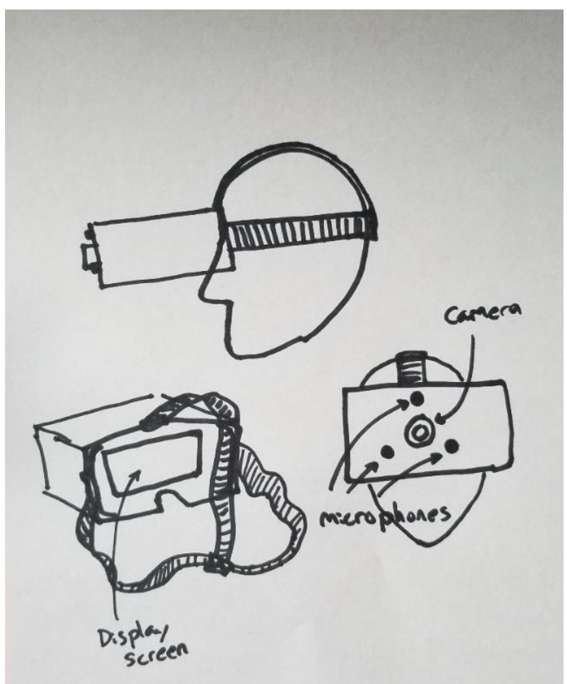


Figure 12: Design B Sketch

4 Design Constraints and Standards

Design limitations are critical to the success of any project, and the following section details those limits and standards. We can deal with those limits appropriately by using an engineering design method. In order to be successful for the creation of this project, let's go over the financial, technological, and legal limits in the chapter below.

4.1 Related Standards

The standards that were used for our chromesthesia device were acquired from the Institute of Electrical and Electronics Engineers Standards Association, or IEEE-SA. This institute is a well-known functioning arm of IEEE that creates and distributes global standards for a wide range of global technology businesses. Some of these businesses include: the National Electrical Safety Code, power, and energy, etc. Our Chromesthesia device is a consumer technology so the standards that weree used will be ones that refer to those.

4.2 IEEE and Relevant Standards

This section will be discussing the effects on using these standards for the development of the chromesthesia device. This device's purpose is to generate interest in the world of synesthetes and see how they live their lives with this special phenomenon that happens to them.

Customer/User Satisfaction
The user will be able to experience a very rare phenomena that only a small percentage of humans in the world can naturally experience. The target audience is anyone that wants to see the world how synesthetes experience it and they should be able to gain educational and enjoyment value. The project would be managed in a dynamic manner with effective risk management, and we would be able to achieve our goals. We should be asking questions and referring to our project's aims and objectives at every stage of the design process.

Risk Management
This is a process that is ongoing throughout the development of a device, system, or service. It can be used to assess risks associated with system development, maintenance, and operation. Risk analysis must be addressed within the team to identify any potential risks and hazards, in order to establish the risks and contingencies that could possibly hurt the users and/or developers. For example, the Chromesthesia device could break if the user mishandles it and could potentially short circuit or cause some other kind of injury. If mishandled, the device could also produce delayed outputs which would change the way the user would experience chromesthesia.

Durability
The device needs to be durable; it should be able to be used dozens of times, be handled by different users, and maintain the same performance throughout all simulations. The device should ideally be able to be used more than 50 times by 50 distinct people. The microphones should be free of interference, but due to the conditions expected in the

engineering atrium, the microphone array will require a particularly powerful noise processing system to cope with the environment. By choosing a reliable power source, along with having a reliable build, while withstanding potential exterior forces, being durable is obtainable. Since this is an electronic device, the possibility of dust accumulating inside of the device is possible if not being used over a period of time, usually three months. With this in mind, any user needs to be mindful of dust buildup which could possibly cause malfunctions. Furthermore, the device will be susceptible to other hazards for electronics such as moisture and temperature.

4.3 Design Constraints

Design limitations are critical to the success of any project, and the following section details those limits and standards. We dealt with those limits appropriately by using an engineering design method. To be successful in the creation of this project, we are going to go over the financial, technological, and legal limits in the chapter below.

4.3.1 Economic and Time Constraints

Based on what was necessary for the project and the date (April 2022), some roadblocks could be identified, giving our crew a total of 8 months to finish it. It was the group's responsibility to overcome these obstacles and deliver a proper design and demonstration.

Economic Constraints
Some of the economic constraints the group took notice of were problems like not knowing the exact price of all the parts and pieces we needed for the project. There was no sponsor for this group, so finding funds was more difficult. There was no proper instruction in the beginning of how our budget for the device would be spent, although after conversing with group members it was determined that everything would be split equally. The members of this group wanted to demonstrate and showcase an amazing project, but we found out some pieces that would be the best parts that could possibly be purchased come out to be expensive and out of our budget. This resulted in the team members needing to take a step back and re-evaluate the situation and consider alternatives, which could cause a big delay in production. With all this being said, all potential problems and solutions, such as manufacturing of materials, software, equipment costs, etc., were being reviewed constantly.

Time Constraints
Some of the time constraints the group took notice of were potential issues such as being a team project if one person falls behind, everyone falls behind. This turns into a matter of checks and balances and making sure everyone stays on track or needing to pick up the slack for other group members due to external circumstances. This project is of a large magnitude and needs to have certain time periods for project milestones to be completed. With our combined group’s experience being low, there were many uncertainties in regard to knowing when certain things need to be completed on specific dates. Another constraint for our group was the availability of the individual group members, which made having meetings more difficult than average. Other constraints that have been discussed were late delivery times, incorrect parts being delivered, or parts being the wrong size, not exactly

what is needed, or possibly just not being what the project will need. Time management is key. If there is no sense of good time management throughout the team, this will inevitably cause a downward spiral in team morale and more importantly, project production.

4.3.2 Social, Environmental, and Political Constraints

Social Constraints
These consist of factors that can come up as a result of growing interest in a project or potential opposition. If a project uses public money, public concern and potentially pressure from media outlets can create tight constraints on a project which can result in changes to the product's original development roadmap.

Environmental Constraints
Any circumstance concerning the wellbeing of an individual person, or the environment are real potential problems that need to be considered when determining constraints. Since this project is not the development of this project mainly concerns:

- Use of hazardous materials.
- Energy consumption
- Proper disposal and waste management
- Safety and wellbeing of users

Political Constraints
The political constraint factors for our project primarily had to do with our use of microphones. While the design of this project did not utilize an internet connection, some questions might be brought forward about our algorithm, and the possibility of voice being recorded, and the data being stored locally.

4.3.3 Ethical, Health, and Safety Constraints

Ethical
Some ethical limitations that needed to be considered relate to our handling of audio and visual recording that were required for our project. By utilizing a camera, we have a visual archive of any individual who may be present during a demonstration. It is expected that footage not useful to a demonstration will not be saved, and promptly removed.

Health
As with any visual based process, consideration needs to be made for the health of a user. The design of this project incorporates visual based imagery, so it is important to notify a potential user that imagery on the screen can cause issues for conditions such as epilepsy.

Safety
Safety considerations are important in both design, and demonstration of our device. While the focus of this device was mostly be software related, we needed to consider the fact that faulty software could lead to unintended consequences, especially with a design based around heavy computations. One such issue could have been the overheating of our

Microprocessor. Depending on the severity, this can cause a fire hazard or injury if touched. Another safety consideration is making sure we keep to IEEE standards, as we used microprocessors and various other electronics with this device. Failure to follow these standards can result in dangerous exposure to electricity.

4.3.4 Marketability and Interoperability Constraints

Marketability Constraints

Marketing also has constraints, such as the capacity to get a product into the hands of potential customers. Consumers must be attracted to promotions that appropriately reflect the items and services offered. Your sales could be severely hampered if you make a mistake with your demographic or price approach. The size of your market, the demand for your product, your ability to supply those goods, and the nature of your competitors are all marketing restrictions. Your marketing efforts may be hampered by the quality of your employees, management's direction, and your general ability to fund your approach.

It is impossible to create a product in a vacuum. They must represent the marketplace's requirements and wants. Products that do not match market needs are a marketing limitation that will have an impact on sales in the long run. Negative word-of-mouth can drive future sales down, even if initial sales are great. As a result, it's critical for marketers to take steps to genuinely understand what customers value and to create products and services that meet those demands.

Marketers' promotional materials, both in terms of content and style, must clearly represent the characteristics of the items and services being marketed to consumers. Promotions that aren't tailored to the right audience or distributed through communication channels that reach the right people are a major bottleneck. Furthermore, consumer dissatisfaction can be exacerbated by advertising that over-promise or exaggerate brand or product features.

Marketing's fundamental purpose is to influence societal change and to educate consumers about their options and the consequences of their decisions. Consumers can look to marketing and advertising for explanations of what components certain brands contain, rather than just buying any cereal. The vitamin content, the amount of fat and sugar in each serving, and the number of calories in each serving can all be displayed on the box. Further marketing can be done on a company's website, in printed materials, and through spoken advertising and sponsorship to explain why those health components are vital to consumers.

Interoperability Constraints

There are various methods to define interoperability, and the goal is to come up with one that can be used uniformly across the organization or group. It is preferable to utilize the same definitions for the enterprise. Many organizations find it helpful to divide interoperability into the following categories: Corporate interoperability, also known as operational interoperability, describes how business processes are to be shared. The term "information interoperability" refers to how information is shared. Technical interoperability specifies how technical services should be exchanged or at the very least connected. Interoperability should be improved such that it unambiguously fits the needs of the project.

4.4 Language Standards

One of the first things a computer programmer learns while learning a new language is that different languages have different standardized techniques. A programming language standard is a document that defines the syntax, instructions, and semantics of that language in computer science. Consider the programmer as adhering to a contract with the language implementation. When a programmer adheres to a standard, he or she can expect the implementation to produce the anticipated results. The implementer, on the other hand, can presume that the programmer employed a language with no further features than those required.

Another way of looking at it is that a standard is a collection of guidelines that a project must follow. Not all languages are standardized; the following are some instances of standardized and non-standardized languages.

Standardized languages:

- C++ (ISO/IEC)
- C# (ECMA-334)
- C (ISO/IEC)
- Ruby (ISO/IEC)
- JavaScript (ECMA-262)
- Common Lisp (ISO/IEC)

Non-standardized languages:

- Python
- PHP
- Perl

4.4.1 HTML Language Standards

The rationale behind "Hypertext," or HTML, is that it allows you to click on links in online sites. "Markup" is a term that refers to something that would ordinarily be marked up in English to denote something. During the Cold War, the internet grew in popularity. It was a mechanism that was used for bunkers to communicate with each other. The public was allowed access to the network in 1969. Meanwhile, in the year 1999, Tim Burner Lee, a British inventor and computer scientist, proposed the world wide web through a document he had written. This stated that it would connect a large number of computers all around the world and to control this network, he wished to use "HTML." He aimed to build a human-readable and understandable language and imagined a world where anybody could alter documents. They would also include a built-in code editor with the computers, which the customer could use immediately. HTML, at the time, was not yet fully developed when it was originally introduced.

Standard Generalized Markup Language, or SGML, was a fairly similar language at the time. The Internet Engineering Task Force, or IETF, was in charge of assigning the first real version of HTML. The Internet Engineering Task Force was merely standardizing what had already been created. The W3C, or the World Wide Web Consortium, was eventually formed as a new standard task group. They did the same thing as the IETF although there was an element called XHTML that had no standard, so they standardized it with HTML. The language was built with a rigorous syntax thanks to XHTML. It was known as the first-time people had a clear understanding of what they needed to do, because there were hundreds of ways to build HTML. People had to identify whether they were writing a strict or a transitional document and XHTML established a clear standard.

Despite being a clear guideline for developers, XHTML was doomed to fail due to the debut of Internet Explorer, a well-known browser. Over 80% of Internet Explorer users were unable to use their code because it was not compatible with XHTML. As a result of this industry's success, the WHATWG (Web Hypertext Application Technology Working Group) was a new group that was formed. They had a leader who made the ultimate decisions, which allowed them to complete more jobs faster and they also came up with two of the world's most famous concepts: There being two types of web applications, web forms and web applications. In modern times, humans now buy things on the internet using web forms. People can only give a service on the internet through web applications. W3C and WHATWG agreed to collaborate in 2009, putting their differences aside and HTML5 was developed around this time.

The Doctype declaration, <!DOCTYPE html>, tells the browser the version of HTML a user is working with, in this case HTML5. This is simply a note at the top of the HTML file to indicate which version we're working with. The web page is then followed by <html>, which states that anything between these two tags is the web page. All other HTML tags will be contained within this container. The next tag is the head tag, <head>, which simply contains metadata about the data in the document. Within the page, there are a variety of tags that can be used between the head and the body, such as <title> and <body>. The title tag determines how the web page's title will appear in the browser. The body tag is also present. The actual content of the document will be contained within the body tag along with any text and/or photos. Tags can go inside other tags, which is a key feature of HTML, and thus most of the HTML tags used will be within the body tag. To prevent the code from getting cluttered and making it easier to see what tags are within others, the common convention is to use a standard indent in order to space out the tags.

Standard Generalized Markup Language, was a fairly similar language at the time. The Internet Engineering Task Force, or IETF, was in charge of assigning the first real version of HTML. The Internet Engineering Task Force was merely standardizing what had already been created. The W3C, or the World Wide Web Consortium, was eventually formed as a new standard task group. They did the same thing as the IETF although there was an element called XHTML that had no standard, so they standardized it with HTML. The language was built with a rigorous syntax thanks to XHTML. It was known as the first-time people had a clear understanding of what they needed to do, because there were hundreds of ways to build HTML. People had to identify whether they were writing a strict or a transitional document and XHTML established a clear standard.

4.4.2 JavaScript Language Standards

One may say that JavaScript is one of the most important languages utilized nowadays when surfing the web. JavaScript was introduced by Netscape in 1995 as an "easy-to-use object programming language designed for constructing live web applications that link objects and resources on both clients and servers." JS follows the ECMAScript language standard, just like other languages. The basic JavaScript language is defined as a lightweight, interpreted programming language that is complementary to and integrated with Java and HTML, as well as being open and cross-platform.

JavaScript allows a user to perform and implement complex features on web pages. Some functions a developer can implement into a web page are animating images, displaying interactive maps, and update content dynamically. Another interesting functionality that is built in JavaScript are APIs, or Application Programming Interfaces. Essentially, they provide the user with an insane amount of power and functionality that they can implement in their code where in other languages or situations, it would be impossible to do so.

APIs are blocks of code that are ready to go, like furniture kits at IKEA, where a consumer can purchase a kit that contains panels that are pre-made and simply just screw them together instead of going through the arduous process of purchasing all the raw materials, cutting and shaping them, and putting them together from scratch. APIs fall into two categories: third party and browser. Browser APIs are already built inside web browsers and use the data from a user's computer to do complex tasks. An example of this would be Google, with their app Google Maps. It utilizes the geolocation API in the browsers which receive geographical information, and it finds your location, plots it on a map, and can even take it a step further and show you businesses and areas near you. Third party APIs by default are not built into browsers. Usually, a developer would have to acquire the API from a company's website or somewhere on the Internet. One interesting third party API that a user could utilize is the Twitter API, which grants users the ability to display their tweets on a website of their choice.

4.4.3 C Language Standards

C is a high-level and general-purpose programming language. Within this programming language, there are paradigms in which statements are used to affect the state of a program in computer science. Object-oriented, functional, logical, and imperative are only a few of the different paradigms that C belongs to.

Between the years 1972 and 1973, Dennis Ritchie, a computer scientist at Bell Labs, developed C. C was a successor to the programming language B. It gained popularity because of its supporting structure, which included lexical variable scope, recursion, and a static type of system. The language is classified as a high-level language, or HLL, which is a programming language that allows programmers to create self-contained applications. C is classified as HHL because it mimics human languages and differs from machine languages like MIPS. As time progressed, companies began to move toward using different programming languages, and although C is not as popular as it was in the beginning, it is still commonly used. ISO/IEC 9899:2018 is the most recent version of the C standard. Many referred to it as C17, which is an abbreviation for ISO/IEC 9899:2018.

C may be used to generate lists of commands that the computer will execute. Following the completion of the lists of instructions, the code will be compiled, making C a compiled language. A compiled language means that your code will be run by a compiler after you've written it. After that, the compiler will take the code and turn it into an executable, which will allow the computer to run or execute. The C program is readable by people, whereas the executable form is readable by computers. To be able to write and run C code, the user must have access to a C compiler. C compilers are a free service that anybody can use; if the user is using a UNIX machine, the compiler is called "cc" or "gcc" and may be accessed via the command line via a terminal. They do have other services that can compile C. An example is Microsoft's Visual C++ environment which is widely utilized in businesses. This commercial usage program costs several hundred dollars and is utilized to assist students in learning by large corporations or colleges.

4.4.4 Python Language Standards

Python is also classified as a general-purpose programming language, similar to C, although it is also an interpreted language and a high-level language. Python was developed considerably later than C and has now risen in prominence. Guido van Rossum developed Python, which was initially released in 1991.

What sets Python apart from other general-purpose programming languages is its emphasis on code readability, as evidenced by its extensive usage of white space. This leads into Python being a very basic and easy-to learn programming language for beginners. Python helps get into the area of object-oriented programming, like languages such as Java/JavaScript. Since it is basic and easy to use and/or learn, Python is popular among programmers because of the level of productivity they can achieve while using it. There is no need for compilation because Python has a cycle where you can edit, test, and debug. A common problem in C and Java is that an error or bug can cause a segmentation fault and not allow the program to run, but in Python this is not the case. Python just raises a flag and labels it an “exception” to inform the programmer.

Unlike C or Java, Python is not standardized, like many other modern programming languages. What these languages use instead, is called de-facto standards, which basically implies whatever the original working implication is. There are two major advantages of having a language standardized. One being that it will not vary randomly through time, unlike standardized languages. The user will have access to the documentation, which will make developing an original compiler or interpreter for the language much easier.

4.5 RoHS Compliance

RoHS stands for the Restriction of Hazardous Substances Directive, also known as the "lead-free directive". If they use "any of the restricted 10 substances, any business that sells applicable electrical or electronic products, equipment, sub-assemblies, cables, components, or spare parts directly to RoHS-directed countries, or sells to resellers, distributors, or integrators" who in turn sell products to these countries, is affected. It requires product compliance at the product level and restricts the use of some hazardous substances that are in electronic equipment. It specifically contains ten chemicals that are prohibited. It is specified that there cannot be more than 100 ppm of Cadmium (Cd) or Lead (Pb) and there cannot be more than 1000 ppm of the following substances: Mercury (Hg), Hexavalent Chromium (Cr VI), Polybrominated Biphenyls (PBB), Polybrominated Diphenyl Ethers (PBDE), Bis(2-Ethylhexyl) phthalate (BBP), Dibutyl phthalate (DBP) and Diisobutyl phthalate (DIBP).

The RoHS standard must be met by the products in the following categories. Large and small domestic appliances, computing and communications equipment, consumer electronics, lighting, power tools, toys and sports equipment, medical devices, and equipment, monitoring and control equipment, automatic dispensers, and other electronics not mentioned in the list are all included. With RoHS 3.0, the "other electronics not covered in this list" become law in 2019. Of course, there are goods that are prohibited, just like any other compliance, such as equipment designs for launch into space, big scale stationary industrial instruments, large size fixed installations, transportation vehicles, and so on.

Exemption periods range from five to seven years and are determined on a case-by-case basis. As new technology becomes available, new regulations and exceptions are created. For example, applications related to electrical quotations have recently come into scope, and new applications with aspects of innovation are on the list of exempted applications.
You can also ask for the certification of any standard to be changed or renewed. The item must be removed if the change or renewal is not accepted. If you need to be exempt for whatever reason, the process can take up to 18 months. The ROHS can be demonstrated in a variety of ways. For instance, displaying a declaration of conformance with an industry standard such as IPC-1752. CE is a mark that can be applied to any product to indicate that it complies with industry standards. ED needs official company communication after an employee makes a declaration, and that letter must state specific references to the parts or products that are covered by the declaration, and it must be signed by the person's name, contact title, and dated.

4.6 Soldering Standards

Soldering is one of the constraints that we kept in mind while developing our project. Soldering electrical and electronic assemblies has evolved to a particular level around the world. The next subsections will be going into more detail about how important soldering will be for our project.

4.6.1 Soldering Iron

In terms of our prototype's construction, soldering was critical to the assembly process. Soldering was required to allow all the components to be correctly integrated, from PCB board installation of parts to different sensors, cables, and necessary controllers.

However, when it comes to soldering, there are several national regulations that we as engineers must follow. According to the Department of Health and Safety of the University of Cambridge, they have provided us with the necessary information and instructions to guarantee that soldering standards and safety precautions are followed correctly.

To begin, the first safety criteria to consider is the use of the iron tip to initiate soldering. Both organizations state unequivocally that the iron point should never be physically touched since it surpasses 400 degrees Celsius and may cause bodily injury if brief or long contact is made. To guarantee even more safety, these colleges recommend holding any possible soldering parts with extension tools such as tweezers or clamps to keep safe distances during soldering. A moist sponge will be needed to remove excess solder from the tip iron in order to protect it from damage and maintain its efficiency. Finally, when not in use, to avoid surface damage and potential fire dangers to the work area, invest in adequate solder iron storage, such as stands or holders, and always unplug the iron while not in use.

Additionally, one can consider putting gloves on to protect their hands from any potential residue coming out of the solder, but if the gloves are too big or tight, they can actually restrict movement and can actually be a hazard when using a solder. Eye protection is also a good safety measure since solder spitting is common and there is a very high chance of some residue landing in the user’s eye or someone around them. Figure 13 is an example of what could happen when soldering. The spit could very easily land in someone’s eye and cause some very serious damage.

Figure 13: Example of eyeglass protection from solder spit

A good practice if someone is not experienced in soldering, is they could take a junk device that they can sacrifice and practice soldering on it until they are sure they can do what they had originally planned on doing.

4.6.2 Solder, Flux, Cleaner

At all times, safety gear should be worn when soldering mainly because solder excess might be a physical safety issue. Leaded solder and other kinds of residue can very easily rub off on everything so users should take note and be careful what they do in and out of the workplace. We can ensure that any solder packs are lead-free to avoid respiratory sickness from fume inhalation. While soldering and after soldering, keep the area free of solder particles and wash your hands, arms, and other potential solder contact sites on a frequent basis to keep your health. Finally, any hardware, solder packs, or equipment that has special solder safety rules, whether for user or product protection, should be avoided.

4.6.3 Rosin Exposure

A quick word of caution to be mindful of when using Rosin. If a person is soldering in an area where rosin is present, it is best to take note of the visible vapors that are being created and follow proper safety practices, such as wearing protective eyewear. The main reason being if exposed, the fumes could cause severe damage to a person's body, throat, lungs, and brain over time. The best precaution to take to avoid this would be to read product labeling correctly and make sure there are no substances that could harm the user.

4.6.4 Lead Exposure

Traditionally, solder was a lead-tin alloy with around 40% lead content. Lead is a highly toxic metal, and the use of lead solders was regulated by the RoHS. When soldering, the temperature is usually at about 380 degrees Celsius, or 716 degrees Fahrenheit, and lead fumes typically gets bad around 450 degrees Celsius, or 842 degrees Fahrenheit. If the person using the solder is experienced in "soft soldering" method, the possibility of breathing in lead fumes is insignificant.

Since we are not experts at soldering, it was imperative to use lead-free solder since inhaling fumes can cause serious reproductive difficulties. Lead exposure can also cause neurological damage along with health problems consisting of muscular, respiratory, and more. Lead exposure is dangerous because serious health problems can occur with short- or long-term exposure to it.

4.6.5 First Aid

Knowing how to administer first aid in the case of any situation that could harm the user is important to know. When most people get burned, a natural reaction would be to stick their burned skin into their mouth. This is something that would certainly need to be avoided because there is the possibility of the skin being contaminated with lead or flux. One should keep in mind where the solder is and should avoid having it hovering over any exposed skin or body parts since the residue from the solder could potentially land on clothes, which makes it hard to remove and difficult to prevent burning.

Burning can also be avoided by being aware of the metals that are being used. Copper is a very common metal that is used when soldering and it conducts heat very well so the user must be cautious when handling the copper and be aware of the copper pins that are being soldered. There is also copper in the PCB, so it is important to take note of that as well. If there was a fire or a burn, the proper steps to handle the situation would be to place the solder tip in a safe area away from other substances that could escalate the problem and then go through the process of doing first aid on a blister. The steps for that would be to go to a sink or grab some cold water over the wound for a period of time, since this is an example of a solder burn, we would pour the cold water over the wound for at least 10 to 15 minutes.

What happens next depends on if there is a professional, or a first responder in the area that can help attend to the wound. If there is, let them handle the first aid; but in the case that there is no medical professional around, then the next step would be to wrap the wounded area in a bandage or gauze depending on how severe the burn is. In the event that it is just a first-degree burn, then it would be easy to resolve, and no additional medical attention would be required. In the case that the burn is worse, like over 3 inches in diameter, then professional medical help would need to be contacted to treat the burn properly.

4.6.6 Waste

The proper way to handle waste would start before actually using any of the materials. The user must take note beforehand whether or not the material they will be using is hazardous or not and whether it will be recyclable. In most cases, any material that is recyclable will go in its designated container for recycling and any material that is hazardous must be deposited in its appropriate waste bin for those specific materials, and any solder that has not been used but should be discarded needs to go in a container such as this and be labeled properly. Whether it be a specific bag or container, these areas for waste must follow state and municipal standards, and this includes anything that was used to clean solder, such as rags.

4.6.7 Fumes

Before a person begins to solder, they must take note of their location and make sure that they are going to do the work in a well-ventilated location. An item that a user can use would be a fume extractor which consists of a fan and an air filter. Usually, the smoke when soldering tends to follow, so having a kind of forced airflow in the work area is helpful in extracting the harmful fumes. An open window, blowing fumes away with a fan, blowing fumes away with your breath, and having some kind of an air filter are all different things a person can do that can he with reducing the risks of breathing in harmful fumes. Sometimes when soldering some smoke and fumes will be emitted, and even though they are not harmful or lethal compared to lead or rosin, these are emissions that need to be monitored. Any kind of fume exposure is best kept to a minimal amount.

4.6.8 Fire/Electrical Safety

Keeping Fire and Electrical safety in mind, some ways to reduce the risk of a fire happening, a person can work on fireproof surfaces and wear fire resistant clothing that covers and protects any body part that is close to the soldering iron. There should always be a fire extinguisher nearby in the event of a fire happening. One should also keep in mind the outlets that will be connecting to the soldering iron. Having knowledge on whether or not the outlet is following regulations and working correctly can help prevent the chance of short circuits and any kind of electrical damage.

5 Research and Part Selection

Our project's research was multi-leveled. The first step in the research was to look for similar projects that have been completed by other students or businesses (as shown in section 2.1). We needed to study how these projects had previously been approached in order to learn how to improve and avoid common pitfalls. Once we researched how we were going to go about our project, the process of finding out how to design our device began. The first topic to research was what the synesthesia phenomenon was, how it would correlate with our device, and then we would compare and contrast various goods that are comparable to ours. Other components that needed to be researched were the microcontrollers, because we needed to find one that could compute the calculations of all the audio frequencies and sort them through the algorithm we would create. Another component to be researched would be the microphones. There are so many different kinds of microphones out there so we had to see which ones could handle the specific frequencies we needed along with how far they could pick up audio inputs. By looking at existing goods, we gained a better understanding of the industry and used that information to help us construct and design our Chromesthesia simulation device. We'll look at several electrical component markets to see which ones will work together. Below, we'll go through several devices and their advantages, as well as a table comparison to show which component we'll choose.

5.1 Synesthesia Research

This section will contain research related to chromesthesia itself, along with different case studies found.

Synesthesia is a strange merging of senses in which activation of one modality causes sensations in another, which offers a unique chance to explore how multimodal information is encoded into the human brain. The number of people who experience this phenomenon is less than 1 in every 2,000 people. Those who have chromesthesia are more likely to have perfect pitch, which means they can hear a note and know precisely what it is. This is most likely due to the fact that they are aware of which hue corresponds to each note.

Musical pitch is bound by the qualitative measure of sounds that reflects the concept of octaves, not just by tone frequency. There are two aspects that must be kept in mind in relation to musical pitch, pitch height and pitch class. The pitch class can be looked at as a one-dimensional circle containing the different keys (Figure 14) and the height of the pitch can be considered an infinite one-dimensional line spiraling up from low and high frequencies. A good way to represent musical pitch would be as a helical structure, as seen in Figure 15.

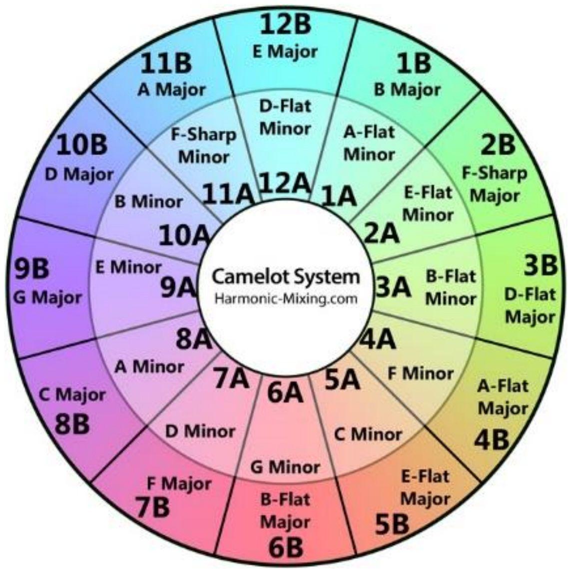


Figure 14: Camelot Wheel

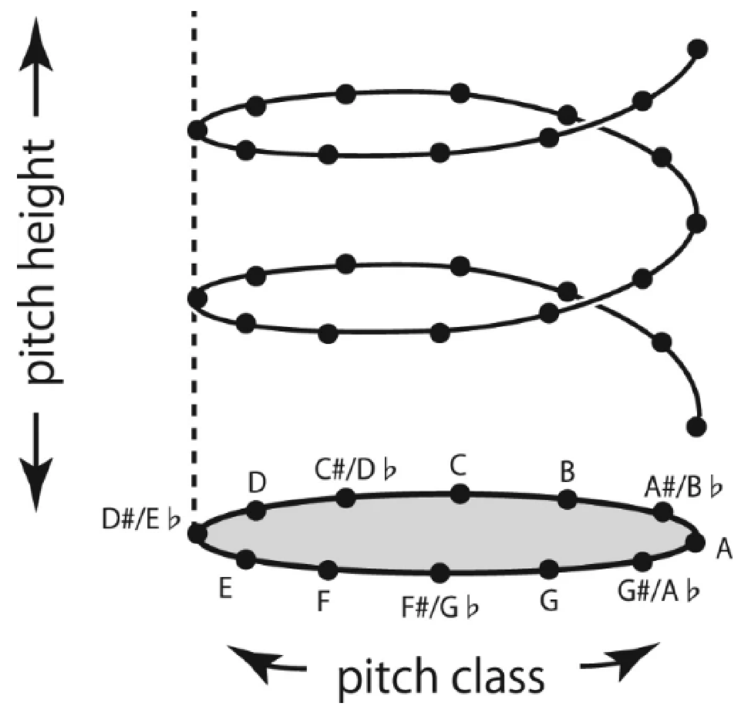


Figure 15: Music Pitch Represented in Helical Model

It is well known when conducting research related to chromesthesia or sound-color synesthesia, that low frequencies are associated with dark colors while high frequencies are paired with bright colors as seen in Figure 16. That being said, the answer to how pitch class correlates to colors with synesthetes is still being investigated, due to the fact that results with different pitches varies from person to person.

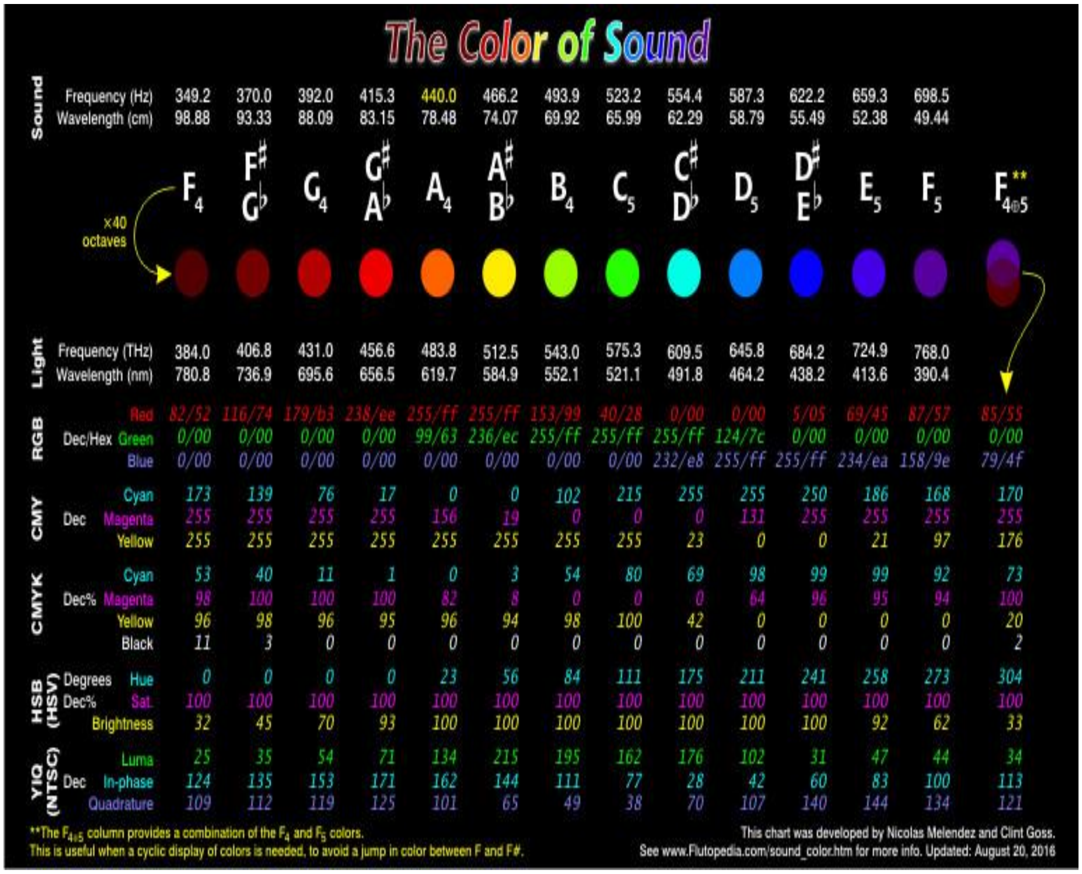


Figure 16: The Color of Sound

A 2017 study published in Nature.com titled “Musical pitch classes have rainbow hues in pitch class-color synesthesia” was the basis for our research and algorithm development. The study was done on 15 subjects who possessed “pitch class-color synesthesia” and had moderate to high levels of absolute pitch. These 15 subjects indicated that they had experienced sensations of colors while hearing different pitch classes and to verify these claims, a consistency test was conducted.

This test consisted of the subjects choosing a color for each of the 12 pitch classes and having these classes specified verbally and recording the results. It was administered twice a day on two different days that were roughly half a year apart, the order of the classes was randomized, and the results pertaining to the RGB values of the colors chosen were recorded and averaged. The researchers who conducted the survey “investigated the pitch classes (*do, re, mi, etc.)* and how they were associated with the three dimensions of color (hue, saturation, and value/brightness.”

As they conducted these studies, results started to reveal that these pitch classes had rainbow hues, which started with red correlating to *do* and yellow correlating to *re*. Another

interesting find was that when there were any sharp or flat pitches, they would have a similar color to their base color (i.e., *do-sharp* correlating to a reddish color).
Table 5 below shows the data collected in the study.

After the subject profiles, the individuals went through a test and then a series of experiments where they had to respond as fast as they possibly could vocally. The test was the color selection test where the subjects were to pick a color for each of the pitch classes, which were specified vocally. After this test, the method used to determine accuracy was that whatever answer the subject would give, it would be compared to whatever they chose in the color selection test. If they matched, it would be deemed correct.

The first was named syllable-to-color, consisting of each person being presented the names of the 7 pitch classes of white key notes on a screen. The second, named syllable-to-syllable, consisted of the test subjects reading the pitch classes that were presented on the computer screen and this was to transform the visual representation of a syllable to its auditory representation before responding. The third experiment, pitch-to-color, was the subjects were presented a series of white key notes at random and they had to report the color they associated the sound with vocally. The fourth experiment, pitch-to-syllable, was similar except they had to report the pitch class, or key, of the notes that were presented randomly.

Figure 17 with the case numbers from Table 6, shows the results of the color selection test along with the averaged colors from the subjects represented with squares. Overall, our observations are that there are many similarities between subjects, especially those who had colored keys on their keyboard when learning music. Table 6 is a rather large table, so it will be on the following page.

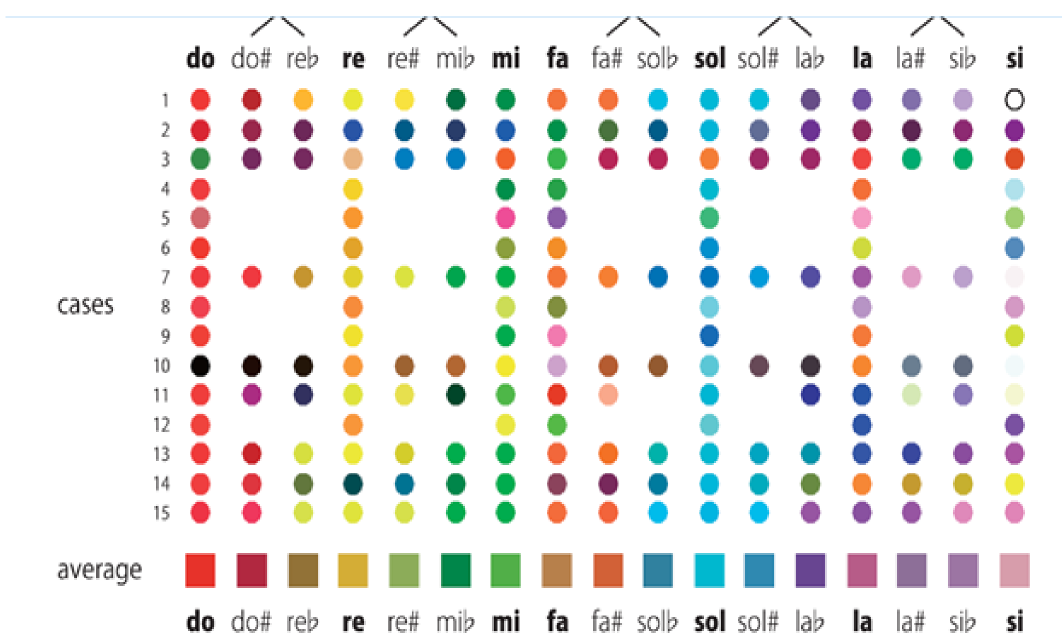


Figure 17: Color-Selection Test Results

Case number	Age	Sex	Music Training (ages)	AP Score	Colored Hearing	Trigger	Acquired
1	22	f	5-18	97	yes	Syllable, pitch	Painted colors on notes
2	22	f	2-22	100	yes	Syllable, pitch	No idea
3	22	m	7-22	100	yes	pitch	No idea
4	21	f	5-21	86	yes	syllable	No idea
5	18	f	5-15	100	yes	Syllable, pitch	No idea
6	21	f	6-20	100	yes	syllable,	No idea
7	19	f	4-19	86	yes	Syllable, pitch (inconsistent)	Color stickers on keyboard
8	21	f	5-18	100	yes	syllable	Color stickers on keyboard
9	20	f	4-20	100	yes	syllable	Color stickers on keyboard
10	19	f	3-18	86	yes	pitch	No idea
11	19	f	11-19	100	yes	syllable	Color stickers on keyboard
12	20	f	3-20	91	yes	Syllable, pitch keyboard key	Color stickers on keyboard
13	20	m	4-17	91	no	syllable	No idea
14	18	f	4-18	94	no	syllable	No idea
15	20	f	6-14	80	no	syllable	No idea

Table 6: Subject Profiles

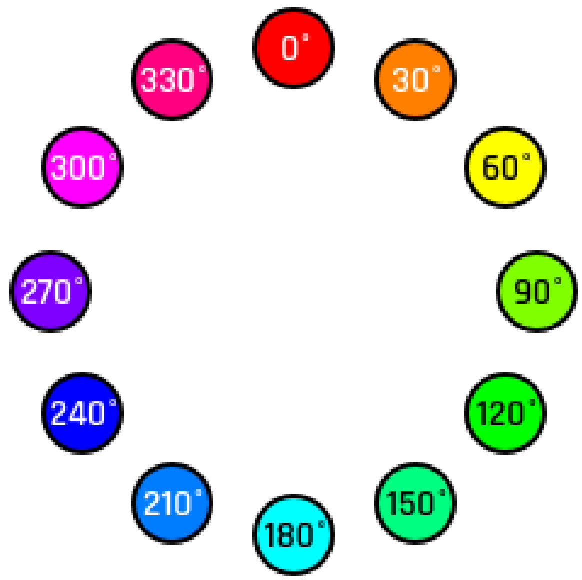


Figure 18: The Hue Wheel

Hue is part of the term HSB, which stands for Hue-Saturation-Brightness. Hue is a number from 0 to 360 that is measured in degrees (Figure 18). Saturation is a number from 0 to 100 which correlates to the “richness” of a color, or in other words, how colorful the color is. Brightness is similar to saturation, the easiest way to think of it would be thinking of a regular light bulb. If it is off, the brightness value is 0 and if it is on, the brightness value would be 100.

When playing the keys on the piano from left to right, it starts with a very low note on the far left and continuing down the sound of the keys gets higher in tone. The way some people would get trained, was to color code the piano keys, starting from the color associated with low frequencies (red) and climb all the way up to the higher frequencies (violet/purple), as shown in Figure 36 in Section 5.2.3.1.

5.2 Microcontroller Considerations

A microcontroller is an integrated circuit that controls some or all of the functions of an electronic device or system. It includes a microprocessor, memory, and associated circuitry. It’s a compact microprocessor designed to operate the functions of embedded systems, as well as any product that can calculate or display data. A microcontroller consists of a processor, non-volatile memory, a clock, an input/output control unit, and more. Due to these many factors, microcontrollers can be divided into different categories in relation to their specifications.

In the next subsections, the various types of microcontrollers are discussed in detail, as well as which ones were investigated for use in the effort to construct our chromesthesia simulation device. The original goal was to create a simple board that would allow us to work quickly and worry less about potential problems. Each component was chosen with care, and a table will be used to compare them. The most important things that we needed to consider when choosing a board include Processor Speed, available RAM, and sufficient peripherals. If our algorithm did not have proper resources to conduct the calculations, we would have run into issues throughout usage. Therefore, CPU and RAM was carefully analyzed and compared between the various Microprocessors. We also needed to consider more than one may be used, in order to facilitate integration of the various amount of processing that are occurring, from the image processing to the interpretation and locating of the sound. To add on to this, other factors that must be considered when doing research on microcontrollers is whether they are single board chips, or microcontrollers. The difference between them would be their processing power and seeing if it aligns to our needs for our device.

5.2.1 BPI-M2 Zero

This is a very robust option as it seems to have all the components desirable for the project. This microprocessor comes with a quad core CPU, a GPU, DDR3 RAM, an SSD slot, and HDMI for visual output. The combination of all these components made this processor a convincing contender since it seems to be made almost exactly for the complex and potentially hardware demanding audio and video processing. Additionally, this processor is advertised to come ready for programming and is able to interface with Linux and Windows systems.

Features	Values	Overview
Architecture	16 bit Architecture with 16 MHz Clock	Processor Clock Speed
Processor	Allwinner H2+ Quad Core Cortex-A7	Processor
RAM	512MB DDR3 SDRAM	Driver Libraries Included
Communication Module	GPIO, UART, SPI, I2C Compatible	Enhanced Serial Comms
Power	5V	Low-power optimized
Peripheral Features	Supported OS: Android, Linux	Supported Operating Systems
Input/Output	40 I/O Lines	Programmable I/O
Operating Range	-40C to +125C	Automotive Temperature Range

Dimensions	65mm x 30mm	Standard
Weight	15g	Standard
Wifi Module	802.11 b/g/n	Wireless Capable
Bluetooth Module	Bluetooth 4.0	Bluetooth Capable
Storage	1 Storage Slot	MicroSD

Table 7: BPI-M2 Zero Table of Features

5.2.2 STM32F3 Series

This processor uses the ARM cortex, which is a well-documented MCU which most engineers in the UCF EE/ECE program have interacted with. This seemed like a solid MCU option with some variety in configurations. A deeper investigation could show which configuration might best suit the needs of this device.

Features	Values	Overview
ARM Cortex M4F	32 bit	Architecture
RAM	8K x 8	Driver Libraries Included
Communication Module	USART/UART, SPI, I2C Compatible	Enhanced Serial Comms
Power	1.65V to 1.95V	Low-power optimized
Peripheral Features	DMA, I2S, POR, PWM, WDT	Prescaler/Compare Mode
Input/Output	36-85 I/O Lines	Programmable I/O
Mounting Type	Surface Mount	Standard
Operating Range	-40C to +85C	Automotive Temperature Range
Dimensions	65mm x 30mm	Standard

Weight	15g	Standard
Wifi Module	802.11 b/g/n	Wireless Capable
Bluetooth Module	Bluetooth 4.0	Bluetooth Capable
Storage	1 Storage Slot	MicroSD

Table 8: STM32F3 Series Table of Features

5.2.3 Arduino Uno Rev3

This board is a versatile, and very popular Microcontroller. It can be used for a variety of purposes. Due to its popularity, there is ample documentation and community-made resources, and the board is open source meaning any information needed would be available online. Developing with this board would be very convenient, additionally, there is the Arduino IDE that can be used, which could streamline development. The project could be developed with this board, which uses an ATMEGA328P MCU. This seems like a good general-purpose device, and we could translate what we developed on the Arduino board onto our own PCB if we use this same MCU.

Features	**Values**	**Overview**
16-bit RISC Architecture	16 MHz Clock	Processor Clock Speed
RAM	32KB Programmable Flash Memory	Driver Libraries Included
Communication Module	USART, SPI, I2C Compatible	Enhanced Serial Comms
Power	2.7V to 5.5V	Low-power optimized

Peripheral Features	Two 8 Bit Timers, One 16 Bit timer. Six PWM Channels.	Prescaler/Compare Mode
Input/Output	23 I/O Lines	Programmable I/O
Operating Range	-40C to +125C	Automotive Temperature Range
Dimensions	65mm x 30mm	Standard
Weight	15g	Standard
Wifi Module	802.11 b/g/n	Wireless Capable
Bluetooth Module	Bluetooth 4.0	Bluetooth Capable
Storage	1 Storage Slot	MicroSD

Table 9: Arduino Uno Rev3 Table of Features

5.2.4 Texas Instruments MSP430-FR6989

The MSP430 is a Microprocessor that supports many functions. This can be programmed using C and C++ and is designed to be integrated with outside controllers. The MSP430 contains several optimizations related to power consumption and is ideal if power in our system becomes a concern. The CPU contained on this particular board is a 16-bit CPU. This microcontroller has multiple registers that can take data for input for high volume applications and still give a high calculation performance. The MSP430 family of architecture is well documented, and the datasheet provides enough information that we can consider this as a candidate for our device. In order to meet the requirements, we have specified, the MSP430 must meet several computation and protocol standards in regard to integration of a camera and microphone.

Features	Values	Overview
16-bit RISC Architecture	16 MHz Clock	Processor Clock Speed
RAM (FRAM)	128KB Nonvolatile Memory at 125ns per Word	Driver Libraries Included

Communication Module	UART, SPI, and I²C	Enhanced Serial Comms
Power	1.8V to 3.6V	Low-power optimized
Low Power Mode	100 µA/MHz optimized low power current draw	Low current draw depending on state
Dimensions	65mm x 30mm	Standard
Weight	15g	Standard
Wifi Module	802.11 b/g/n	Wireless Capable
Bluetooth Module	Bluetooth 4.0	Bluetooth Capable
Storage	1 Storage Slot	MicroSD

Table 10: MSP430 FR6989 Table of Features

5.2.5 Raspberry Pi 4

The Raspberry Pi 4 is a small, compact microprocessor that has the functionality of a personal computer. This microcontroller can run several different compilers, from Linux, C++, Java and Python. The Raspberry is a well-known, and well utilized microprocessor that has extensive documentation available. The Raspberry Pi 4 contains an HDMI port that we can output any process through, making it ideal for displaying what we process through it. In addition, it contains four USB ports, and GPIO pins for attaching components. The 64-bit architecture of this system is ideal, since we require processing and RAM usage of a 64-bit system. The Operating System will use Linux, and we integrated that as part of our design.

The Raspberry Pi is also Wi-Fi and Bluetooth capable. This feature makes it exceptionally useful for wireless applications and can reduce the use of cables.

Features	Values	Overview
Broadcom BCM2711, Cortex-A72	64-bit SoC @ 1.5GHz	Processor
Memory	2GB LPDDR4-3200 SDRAM	Driver Library Included
Communication	Ethernet, 2.4 GHz and 5.0 GHz Wireless	Internet Capable Connection
Wifi Module	802.11ac Wireless	WiFi Capable
Bluetooth	BLE 5.0	Bluetooth Capable
Input/Output	40 Pin GPIO	40 Pins
Power Standard	5V DC USB/GPIO	Device Power
Operation Window	0-50 Degrees Celsius	Ambient Temperature
Dimensions	65mm x 30mm	Standard
Weight	15g	Standard
Wifi Module	802.11 b/g/n	Wireless Capable
Bluetooth Module	Bluetooth 4.0	Bluetooth Capable
Storage	1 Storage Slot	MicroSD

Table 11: Raspberry Pi 4 Table of Features

5.2.6 Banana Pi M5

The Banana Pi is a board that uses an Amlogic S905X3 quad-core Cortex-A55 and a Mali-G31 GPU. Similar to the BPI-M2 Zero, this board consists of multiple components that could be beneficial given the desired functionality. One of the most desirable features of this board is it contains 4GB of RAM, easily enough to process our algorithm. The GPIO protocols utilize several types also.

Features	Values	Overview
Processor	Amlogic S905X3 Quad Core Cortex-A55	Processor Type
Operating Voltage	3.3V	Operating Voltage
Protocol	UART, I2C, SPI, PWM	Multiple Protocols
I/O pins	28 GPIO, 21PWM, all interrupt cable	Input and Output Pins
Bluetooth	Built-in BLE 5.0	Bluetooth Standard
Power	5V	Power Consumption
Memory	4GB LPDDR4	DDR4
Dimensions	65mm x 30mm	Standard
Weight	15g	Standard
Wifi Module	802.11 b/g/n	Wireless Capable
Bluetooth Module	Bluetooth 4.0	Bluetooth Capable
Storage	1 Storage Slot	MicroSD

Table 12: Banana Pi M5 Table of Features

5.2.7 RedBoard Artemis

The RedBoard Artemis is a board that is comparable to the Arduino Uno. When exploring options for functionality and operating specifications, this board became an important board to compare against the Arduino Uno. In fact, this board is compatible with the IDE provided with Arduino products, making it functionally simple to prototype and utilize. This board has the functionality to boost its processor speed up to around 96 MHz, an important consideration when we consider the algorithm that will be required for this device. This board also contains 384 KB of RAM, which is significantly more than the

Arduino Uno. As we are prototyping the algorithm, this extra RAM coupled with the boost of the Processor speed will complement each other and provide sufficient speed for our prototype. In addition, there are 10 ADC channels with 14-bit precision, allowing for a range of 1024 of operation for our pins if we use their full capabilities. Finally, in addition to the Bluetooth capabilities described with the Arduino Uno, we have UARTs, 6 I2Cs, and 4 SP buses, a PDM interface, and an I2S interface with a Digital MEMS mic for voice command features, proving that our MCU is the entire package in terms of hardware and programmability.

Features	Values	Overview
Processor	32-bit Ambiq Apollo Cortex M4F	Processor Type
Operating Voltage	3.3V	Standard Voltage
IDE SW	Arduino or Ambiq Apollo SDK	Software Type
I/O pins	24 GPIO, 21PWM, all interrupt cable	Input/Output Pins
Bluetooth	Built-in BLE 5.0	Bluetooth
Power	5V	Power Consumption
Memory	384kB RAM, 1MB Flash	DDR2
Dimensions	65mm x 30mm	Standard
Weight	15g	Standard
Wifi Module	802.11 b/g/n	Wifi Capable
Bluetooth Module	Bluetooth 4.0	Bluetooth Capable
Storage	1 Storage Slot	MicroSD

Table 13: RedBoard Artemis Table of Features

5.2.8 Microcontroller Selection

In order to choose a Microcontroller, several factors had to be carefully analyzed. The primary factor is since this project relied on an algorithm, it was important we included a board that has a well utilized library of tools for our code. The Raspberry Pi is Linux based, and supports a wide variety of programming languages, like C++ and Java. The integrated HDMI port and GPIO pins allowed easy integration to any sort of device, we may use.

With this in mind, we have decided to choose the Texas Instruments MSP430-FR6989 and Raspberry Pi as our Microprocessors of choice for our device Processor speed and RAM were other considerations in making this decision, with the Arduino Uno and RedBoard Artemis providing excellent speed that was comparable with the MSP430 but didn't contain enough additional features.

Microcontrollers	MSP430FR698	Raspberry Pi 4	RedBoard Artemis
Price	$24	$60	$20
Processor Speed	16MHz	1.5Ghz	96 MHz
Protocols	N/A	Bluetooth 5.0, Bluetooth low energy	Bluetooth
RAM	2-KB	2 GB	1-MB
Non-volatile memory	128-KB	4-GB	384-KB
SPI	4	2	6
I/O Pins	83	40	21
Data bandwidth	16-bit	32-bit	32-bit
Operating Voltage	1.8 to 3.6V	5.1V	3.3V

Table 14: Microcontroller Comparison Table

5.3 Camera Considerations

This section is composed of multiple sections, each providing an overview of a potential camera module we decided to investigate for use in our device.

5.3.1 Arducam Mini OV2640

The Arducam Mini camera is a 2 Megapixel, high-definition camera module. This particular camera utilizes the I2C protocol, which is beneficial to connecting it to our Microprocessor. Since we needed a clear, precise image, the Aducam Mini is an excellent choice.

Features	Values	Overview
Processor	32-bit Ambiq Apollo Cortex M4F	Processor Type
Sensor	2 MP OV2640 Sensor	2 Megapixel Sensor
Mount	M12 Mounting	Changeable Lens Option
Protocol	I2C	Sensor Configuration
Software Protocol	SPI	Camera Commanding
Array Size	1600 x 1200 (UXGA)	Total Size
Power Supply	Core: 1.3V DC ± 5% Analog: 2.5~3.0V DC I/O: 1.7V to 3.3V	Device Power
Power Consumption	Free running: 125mW Standby:600µA	Standard Power Consumption
Image Sensor Format	Type 1/4″	Industry Standard
Maximum Image Transfer Rate	1600×1200@15fps, SVGA@30fps, CIF@60fps	Transfer Rate
Sensitivity	0.6V/Lux-sec	Image Sensitivity
S/N ratio	40dB	S/N Ratio Range
Dynamic Range	50dB	Total Range
Pixel Size	2.2 x 2.2 µm	Pixel Density

Output Format	YUV/RGB/Raw RGB Data/MJPEG	Video Output
Shutter Type	-	Rolling Shutter
Low Power Mode	5V at 20ma	Device Low Power
Dimensions	34 x 24mm	Standard
Weight	20g	Standard
Operating Temperature	-10C to +55C	Operating Range

Table 15: Arducam Mini OV2640 Table of Features

5.3.2 Arducam OV7670

The Arducam OV7670 is an older camera module than the OV2640 that has a 3.6 MP sensor. It uses a VGA output, so conversion of the signal will need to be implemented to make use of this camera. The camera output video is not great, only being 640 x 480 which will be tricky to properly display imagery for our process. This camera has an advantage when it comes to power, however. The consumption is slightly less than the OV2640 model, and therefore should be considered.

Features	Values	Overview
Processor	32-bit Ambiq Apollo Cortex M4F	Processor Type
Sensor	3.6 MP OV7670 Sensor	2 Megapixel Sensor
Mount	M12 Mounting	Changeable Lens Option
Protocol	I2C	Sensor Configuration
Software Protocol	SPI	Camera Commanding
Array Size	640 x 480 (VGA)	Video Output
Power Supply	Core: 1.8V DC ± 10% Analog: 2.45~3.0V DC I/O: 1.7V to 3.0V	Device Power
Power Consumption	Free running: 60mW Standby: < 20µA	Device Power Consumption
Image Sensor Format	Type 1/6″	Image Sensor Size

Maximum Image Transfer Rate	VGA@30fps	Transfer Rate
Sensitivity	1.3V/Lux-sec	Image Sensitivity
S/N ratio	46dB	S/N Ratio Range
Dynamic Range	52dB	Total Range
Pixel Size	3.6 x 3.6 µm	Total Pixel Dimension
Output Format	YUV/RGB/GRB/Raw RGB Data	Output Type
Shutter Type	-	Rolling Shutter
Low Power Mode	5V at 20ma	Low Power Consumption
Dimensions	34 x 24mm	Standard
Weight	20g	Standard
Operating Temperature	-30C to +70C	Total Operating Range

Table 16: Arducam OV7670 Table of Features

5.3.3 Camera Selection

When comparing the choices for the camera, utilizing the important specifications is important when selecting our camera. This disadvantage for the OV7670 includes the poor resolution in VGA output of 640 x 480, which is not sufficient for our design. While the OV7670 camera does have a lower power consumption, the OV2640 camera will be selected for its capability to display high resolution video, which is an important requirement for this design.

Features	OV2640	OV7670
Array Size	1600 x 1200 (UXGA)	640 x 480 (VGA)
Power Supply	Core: 1.3V DC ± 5% Analog: 2.5~3.0V DC I/O: 1.7V to 3.3V	Core: 1.8V DC ± 10% Analog: 2.45~3.0V DC I/O: 1.7V to 3.0V
Power Consumption	Free running: 125mW Standby:600µA	Free running: 60mW Standby: < 20µA
Image Sensor Format	Type 1/4″	Type 1/6″

Maximum Image Transfer Rate	1600×1200@15fps, SVGA@30fps, CIF@60fps	VGA@30fps
Sensitivity	0.6V/Lux-sec	1.3V/Lux-sec
S/N ratio	40dB	46dB
Dynamic Range	50dB	52dB
Pixel Size	2.2 x 2.2 μm	3.6 x 3.6 μm
Output Format	YUV/RGB/Raw RGB Data/MJPEG	YUV/RGB/GRB/Raw RGB Data
Shutter Type	Rolling Shutter	Rolling Shutter

Table 17: Camera Comparison

5.4 Microphone Considerations

This section is composed of multiple sections, each providing an overview of a potential microphone we decided to investigate for use in our device. When choosing a microphone, we needed one that could detect audio signals with a narrow cone. The reason for this is because we wanted the device to work in a directional manner. We did not want a microphone that could detect audio signals with a large cone. This would have resulted in detecting signals beyond the scope of the camera's aperture. Ideally, we would have like to use a microphone that has a cone similar to that of the cameras visual collection, however it is understandable that this might be hard to achieve, especially considering the low-budget nature of our microphones that will be choosing from.

We decided it was preferable to have a microphone whose cone is larger than the camera rather than narrower. It would be easier and more convenient, and cater towards a more enjoyable user experience, if we had to limit the size of the microphones reading ability rather than that of the camera. if we wanted to cut off the cameras, this would involve reducing the resolution and pixel output. The ramifications of this decision were not explored. However, we can foresee this being a complicated issue to handle. Therefore, it was more desirable to have microphones that had a cone larger than that of the camera rather than a camera with a column larger than that of the microphones.

We identified several microphones that we could have used as potential components for our device. For each of them we briefly explore them and share some information about them.

5.4.1 Adafruit I2S Microphone

When considering microphone usage for this device, several factors need to be analyzed when selecting a proper device that can properly localize the sound and meet our requirements. Our microphone should be able to adequately interpret and pick up the voices of our test subject, while minimizing the impact from outside sound sources nearby. In addition, the selected microphone should be taken and after extensive research, it was determined that the Adafruit I2S Microphone is a suitable choice. This microphone has the ability to. The I2S standard will complement our Micro Processor, as this microphone uses digital audio, instead of analog. The Raspberry Pi and MSP430 both use the I2S standard, so integration of this microphone is the proper route to take.

Features	Value
Manufacturer	Adafruit
Dimensions	16.7mm x 12.7mm x 1.8mm
Unit Weight	0.4g
Operating Supply Voltage	1.6 V to 3.6 V
Interface Type	I2S
Microphone	SPH0645LM4H
Type	MEMS (Silicon)
Direction	Omnidirectional
Frequency Range	20 Hz ~ 10 kHz
Output Type	Digital, I2S

Table 18: Adafruit I2S Microphone Table of Features

5.4.2 DAOKI High Sensitivity Sound Microphone

These microphones are PCB ready and compatible with standard pre-made PC B input pins. These microphones are specifically made in mind for the application of an Arduino Uno or some other pre-made printed circuit board. For this reason, these microphones are not the highest quality, but they are already completed and finished and functioning. The benefit of these is that they are inexpensive, robust, already made, and incredibly easy to integrate into his system and to use for testing.

Features	Value
Manufacturer	DAOKI
Dimensions	34mm x 16mm x 15mm
Unit Weight	4.5g
Operating Supply Voltage	3.3 V to 5 V
Microphone Type	Electret Condenser
Direction	Omnidirectional
Frequency Range	50 Hz ~ 20 kHz
Output	Digital

Table 19: DAOKI Microphone Table of Features

5.4.3 dB Unlimited Unidirectional Electret Condenser Microphone

The dB Unlimited microphone is a unidirectional electret condenser microphone. With a unidirectional microphone, we would be able to capture a more directionally targeted sound. Omnidirectional microphones will detect sound coming from any direction, while unidirectional microphones only detect sounds that come from the front and sides of its face. This would allow for a reduction of noise coming from behind the microphones and would allow 3D sound localization to be easier and more accurate.

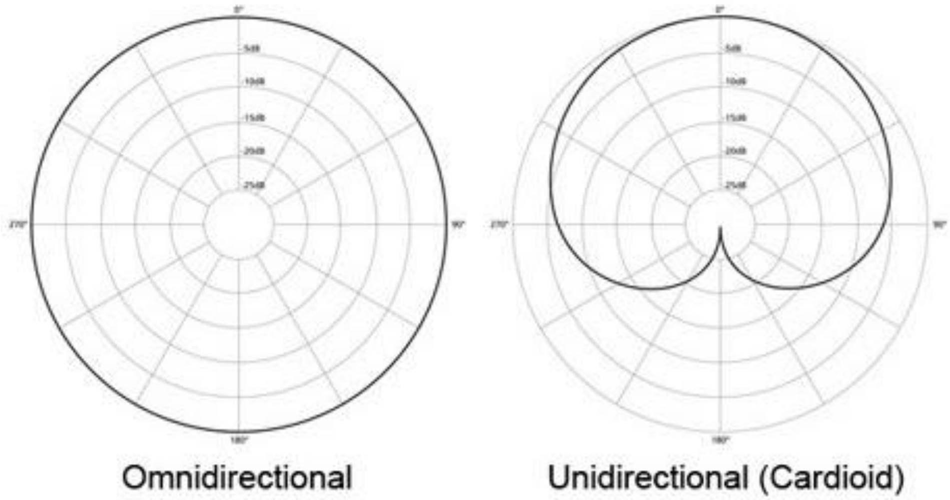


Figure 19 Polar Graphs of Microphone Directionality

Since this product was only the transducer, a preamplifier circuit needed to be added to the design to boost the analog signal from the electret microphone. This was not difficult to design, as there were multiple example circuits that can be found online.

Features	Value
Manufacturer	dB Electronics
Manufacturer #	MU064402
Dimensions	6mm x 6mm x 2.7mm
Voltage Range	1 V to 10 V
Direction	Unidirectional

Frequency Range	40 Hz ~ 20 kHz
Output Signal	Analog

Table 20: dB Unlimited Unidirectional Microphone Table of Features

5.5 Battery and Power Technology

What is a battery? A battery is made up of three basic components: an anode, which is the negative side, and a cathode, which is the positive side, with some form of electrical liquid in between. When electrons flow up into the electrical liquid and then enter the cathode, a chemical reaction occurs. Chemical reactions are converted into electrical energy by batteries. In order to start or utilize a battery, it must be connected from the negative side to the positive side in a closed circuit as shown in Figure 20. When an equilibrium is reached, it means the battery has died and you need to replace it. As a group we needed to decide on a battery that can power our microcontrollers along with potentially the display and camera.

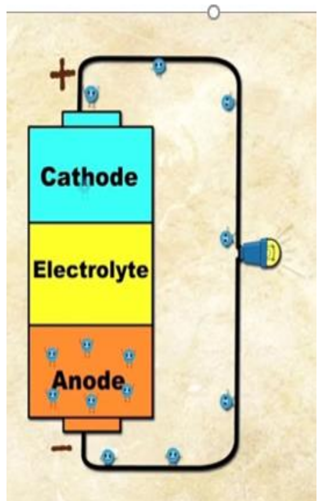


Figure 20: Battery

5.5.1 Alkaline Batteries

Alkaline batteries are typically used to power small gadgets that do not require a lot of power. They're known as the batteries that last longer and are found in calculators and other small gadgets like remote controls. They are commonly used with devices that have low drainage. They are an electrochemical cell that transform chemical energy to electrical energy instantly. Although these batteries are able to be recharged completely, it was not in our project's best interest to use them, simply because naturally they give out 1.5V and we needed at least voltages from 1.8 to 3.6V to power our microcontroller.

5.5.2 Ni-MH (Nickle Metal Hydride)

Nickel Metal Hydride (NiMH) batteries are newer rechargeable batteries that come in common sizes such as AA or AAA and are significantly more popular than NiCad batteries. They are twice more expensive than Ni-Cad which results in having a larger impact on our budget if we decide to go with this option. These batteries have a cell voltage of 1.25V. This means we would need to use at least three of them to achieve the needed voltage. Nickel Metal Hydride has a good life cycle, with most of them lasting for 500 cycles or more. Ni-MH batteries are also simple to recharge, but they suffer from self-discharge, which means that even while not in use, they will lose charge. Ni-MH batteries do not have the same memory charge problems as Ni-Cad batteries; thus, we don't have to be as careful when charging them. We prefer these rechargeable batteries to Ni-Cad since they have all of the benefits and none of the drawbacks. As long as we remember to charge the batteries before we use them, self-discharge should not hinder our project presentation.

5.5.3 Lithium Ion

Lithium-ions are rechargeable batteries that are commonly used to power gadgets that require a large amount of electricity. These batteries were developed in the 1970s and first commercialized in Japan in 1991. They are now used to power cellphones, computers, electric vehicles, and other electronic devices. Furthermore, as these batteries' use has grown in popularity, their safety has improved, and they now contain fewer contaminated metallic elements than other batteries. Lithium-Ion batteries that are 1kg can generate up to 150 watt-hours of electricity. We can see how a lithium-ion battery can be useful because it can be compact while still storing a considerable quantity of energy. When completely charged, a typical lithium-ion battery has a voltage range of 3.0 to 4.2V and a lifespan of two to three years, or 300 to 500 control cycles.

5.5.4 Power source Comparison and Selection

Table 21 below is a comparison table made for the purpose of displaying information found of rechargeable batteries that we researched. As a group we decided we wanted to go a battery route, but also considered using a charging bank. In the table, there will be key factors the batteries were compared to that we believed were important for the design of our device. We investigated voltage, life cycles, charge capacity, price and more. Based on these features and specifications, we chose which rechargeable battery to utilize.

Different Batteries/Power Options	Energizer AA Batteries (24) count	Lithium Ion	Ni-MH	Portable Anker PowerCore 20100mAh
Voltage	1.5V	3V to 4.2V	1.25V	5V
Cost	$16.73(24)	$10+ each	$2+ each	$49.99
Life Cycles	1,000+	300+	300+	1,200+
Charge Capacity	3000mAh	750mAh for phones, otherwise varies	25000mAh	20100mAh
Power Density	297 Wh/Kg	126 Wh/Kg	100 Wh/Kg	72.36 Wh/Kg

Table 21: Power Source Comparison

The main concern for battery power supply is the large amount of high-power demand components. As the project developed, the group decided to explore the potential of using two microcontrollers - one for handling image processing, and another for handling audio processing. In addition to this, this device would also have a high-definition display, and for sensory components (three microphones, and one camera). paste on power consumption of the components that we have identified; it seems possible to base our power supply on Lithium Ion batteries

Additionally, using a portable battery as the power supply enabled us to let the user operate the device in any location without any restrictions to their location or orientation. Avoiding cables was a main goal, especially since the user would be unable to operate the device wherever and however, they want to if they were forced to remain plugged in to an external power supply that was not connected or attached to the device itself.

Below is Table 21 which highlights the power consumption of all the devices that are under consideration.

As it can be seen each component requires around 2 volts or more. this was a concern especially when the fact that multiple components of the same type will be used at the same time. The microphones especially are going to be a huge power draw because there will be three or four of them and the same time processing audio inputs real-time continuously until the device is turned off. likewise, the power consumption of the MSP430 was likely to be on the higher end especially since they would be performing very computationally rigorous calculations constantly as long as the microphones and Camera are reading audio and visual inputs.

Component Type	**Component Name**	**Power Consumption**
MCU	MSP430	1.8-3.6 V
Microphones	DB Condenser Microphone	6 V
Power Supply	9V Battery pack	N/A
Camera Module	OV7670	2.5-3.0V
Display Screen	--	2.6V

Table 22: Itemized List of Power Demand by Component

5.6 Housing Unit and Build Materials

Our device consists of a camera, a display screen, 4 microphones, and 2 microcontrollers, so the housing unit and materials needed to be able to withstand the requirements for all of these components. Some things we considered include air circulation and ensuring that components do not overheat. The weight of the item would have been heavily influenced by the materials used. We also thought of potential cosmetics. How nice will this device look? What would be a good color for its presentation? If sold to consumers, should we allow some sort of customization?

We took a couple materials into consideration because we wanted our device to be solid and durable. At first we considered wood, it’s readily accessible and unfortunately very pricey in today’s market. We would just need a sheet of wood and it would be enough for the whole project. The downside to this was wood can be very heavy. None of us had power tools so there comes the problem of how were we going to cut it and exactly what sizes will we need. So we scrapped the idea and moved towards metal. Sheet metal could have also been purchased locally, it’s also relatively the same price as wood, and it is very fragile and light. Ideally we decided it could cause some conductivity problems in the long run so we moved on from this idea.

Materials for constructing the housing for the device were explored, with the main final idea being a 3D printed plastic housing to contain the PCB and other electronics. 3D printing allowed us to make a high-quality custom-made housing with short manufacturing time. Alternatively, the University of Central Florida provided a laser cutter which could be used to cut wood boards to fit the PCB and electronics inside.

The housing unit was to be designed to have the camera and microphones at the front face and in the center. The camera's location relative to the microphones were noted and documented to enable accurate 3D sound localization as the project develops.

Material	Size	Cost
Sande Plywood	3/4in x 4ft x 8ft	$64.88
Oriented Strand Board	1/2in x 4ft x 8ft	$20.35
Acrylic Sheet	24 in x 48 in x .093n	$39.44
Material	**Size**	**Cost**
PVC Pipe	1 in x 10 ft	$8.12
Galvanized Steel Pipe	3/4 in x 10 ft	$30.96

Table 23: Cost Comparison of Materials

5.7 Serial Communication Technology

With the use of a PCB design board, embedded electronics will be used in the development of this project. The device would have microchip(s) that could communicate with each other. This is done by incorporating serial communications into the circuitry. This kind of communication is available in a variety of formats, including I2C, UART, and SPI. These forms of communication protocols are used to join or establish a system that has interlinking circuits (processors or other integrated circuits). This is accomplished by using a standard communication protocol, which could be defined in two types: parallel or serial, in which each circuit exchanges information. The distinction between parallel and serial interfaces is that parallel interfaces send numerous bits of data at once, whereas serial interfaces send one bit at a time.

5.7.1 I2C Protocol

I2C is a serial communication technology that is widely used due to its ease of implementation. It allows one or more master chips to connect with many slave digital integrated circuits or chips. The simplicity of implementation stems from requiring only two cables (SCL and SDA) to allow communication from 128 devices using 7-bit addressing to 1024 devices using 10-bit addressing. Allowing the master to select what device it will be communicating with makes this implementation feasible by each device's unique ID. An example of an I2C bus, Courtesy of Texas Instruments, is shown in Figure 20.

We can start with the SCL wire, or the serial clock. The master device generates the SCL line, which is the clock signal that synchronizes data transfer between devices on the I2C bus. The SDA wire, or in other words the serial data wire, is the data line. The two lines are open-drain, which means that the output can "either pull the bus down to a voltage (ground, in most cases) or 'release' the bus and let it be pulled up by a pull-up resistor." Normally, active lines on an I2C bus are usually transmitting lows, so these pull-up resistors are attached to the lines and are in charge of pulling the voltage up. This results in the bus not having any communication problems because usually a device will not be sending a high, so the situation of a device sending a high and another sending a low (which causes a short) is avoided.

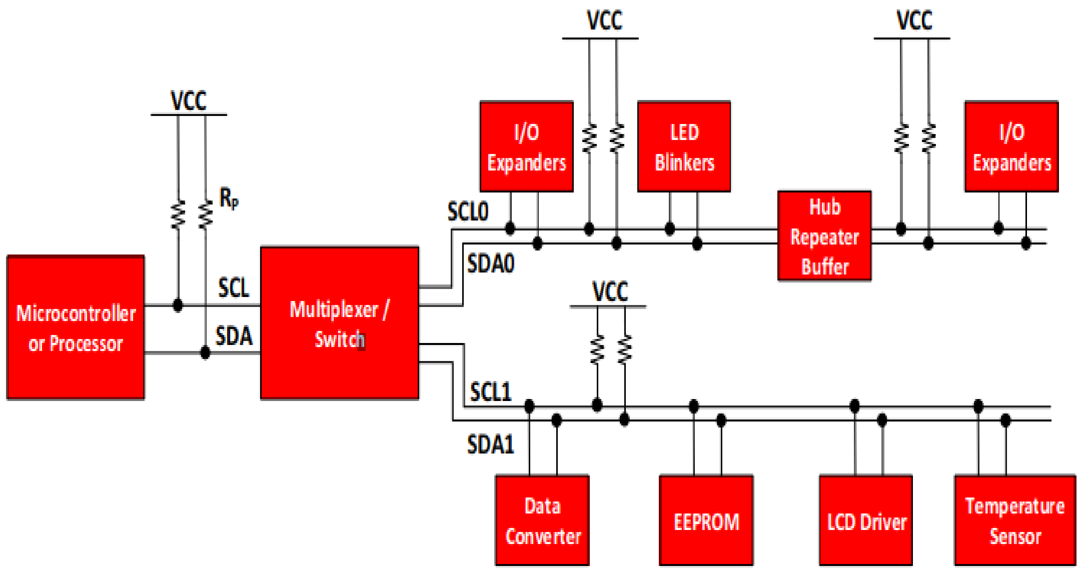


Figure 21: Example of an I2C Bus

Figure 20 depicts an example of an embedded system, with the master being represented by the microcontroller. The entire system is controlled by the two pins on the master, and it is shown that this system has many slave devices, such as the EEPROM, LCD Driver, and temperature sensor. Figure 21 shows a simple wiring with multiple devices on one bus.

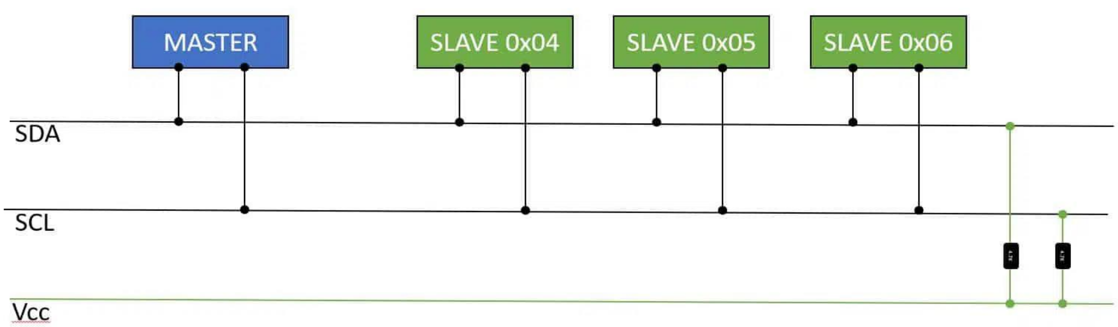


Figure 22: Multiple Devices on One Bus

The data is transmitted in 8-bit sequences, and the sequence is initiated by a particular start condition. Once this sequence is completed, a bit called ACK, or acknowledge, is triggered, which indicates the address of the slave to which the data is being transmitted. Afterwards, another cycle of this sequence begins, but this time it is for the internal registers of the slaves. Following the second cycle, a third acknowledgement is raised, and data begins to transmit through the repetition of the address sequences until it is completely finished. Once everything is done, an end condition is activated. As an example, Figure 22 below depicts a representation of this process with the data sequence of a TMP102 temperature sensor that utilizes I2C.

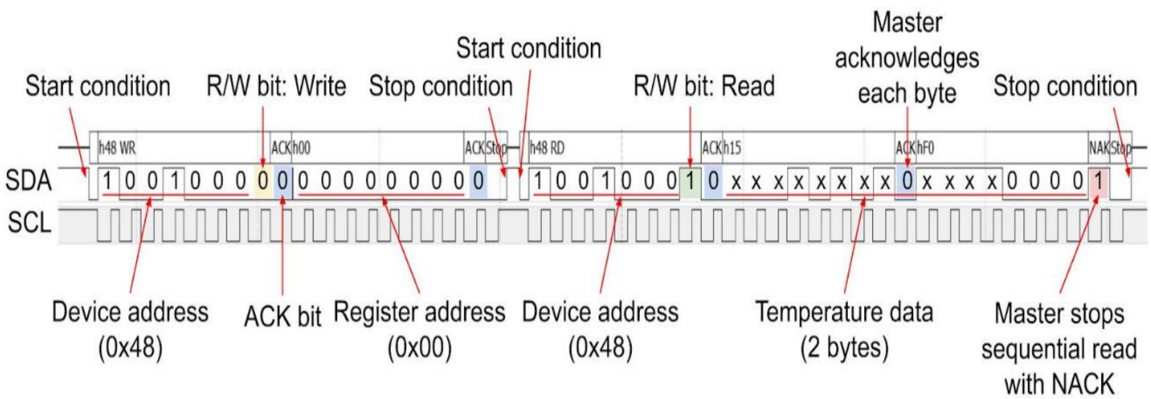


Figure 23: Data Sequence of TMP102 Temperature Sensor using I2C

Looking at Figure 22, it is shown that when the data line dips low and the clock line is high, the starting sequence begins. Once the clock starts, each bit of data is sent at every clock pulse. The 8th bit is utilized to indicate to the master if it should write to the slave with logic low or read from the slave with logic high. The starting sequence begins the most significant bit and ends with the least, which results in a total of 7 bits. Afterwards the slave uses the next bit as an acknowledgement bit to check if the set of bits were transmitted successfully. The slave device is then given control of the SDA line over from the master if the transmission was successful and it is brought down, otherwise this action will not happen. A common reason this happens is not being able to receive more data. The internal register addressing mechanism happens next, which addresses information or data stored in the slave's memory. Once completed, the transfer transmits sequences from either the master or slave, which depends on the read/write bit. When the line goes from SDA low to high and the SCL line is high, the transfer will come to a halt with a stop condition.

5.7.2 UART Protocol

The Universal Asynchronous Receiver-Transmitter protocol is a universal, and well known communication protocol that allows devices to communicate with each other. This particular protocol was developed in the 1970’s, allowing for over 50 years of solid robustness and usage. UART is a unique protocol, because unlike other communication protocols, it does not use a clock. Since no clock is used, it relies on syncing the baud rate between devices. When two devices are communicating with each other, only those 2 devices can use that same UART serial port. UART can send data serially in one of three modes: full duplex, half duplex, or simplex. Full duplex means that data is communicated to and from each device master and slave at the same time. Half duplex communication occurs when just one data packet is sent in one direction at a time, whereas simplex communication is only one-way. Figure 23 below shows the layout of a UART data packet.

Data is conveyed in the form of data packets, which start with a start bit, which pulls the high line to ground. The actual data being sent is contained in the data frame. If a parity bit is used, it can be five to eight bits long. The data frame can be nine bits long if no parity bit is used. The data is usually delivered with the least significant bit first. The even or oddness of a number is described by parity. By using the parity bit, the receiving UART determines if data was changed during the transmission. Once the data frame is read, the process of determining whether the parity bit matches the data begins. The data transmitted has its bits counted with values of ones and it is determined whether or not it is even or odd. For example, if the parity bit is 1, which results in an odd parity, the data's bits will add up to an odd number. Finally, the data's transmission line's voltage changed from low to high for one or two bits to signify the end of the data packet.

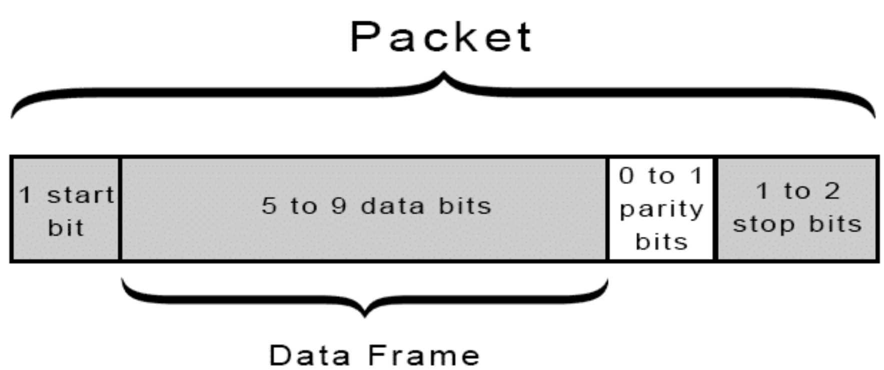


Figure 24: UART Data Packet

There are five steps in the UART transmission process. First, data from the data bus is received in parallel by the sending UART. Second, the transmitting UART inserts the data frame's start bit, parity bit, and stop bit(s). Third, the full packet is transferred serially from the sending UART to the receiving UART, starting with the start bit and ending with the stop bit. The data line is then sampled by the receiving UART at the specified baud rate. Fourth, the receiving UART discards the data frame's start, parity, and stop bits. Fifth, on the receiving end, the receiving UART translates the serial data to parallel and passes it to the data bus.

5.7.3 SPI Protocol

The Serial Peripheral Bus (SPI) protocol is closely related with the I2C protocol; however, it does have some slight differences. This protocol in particular requires three wires, all shared by the bus, in order to work properly. In turn, however, the SPI communication protocol can transfer data at a faster rate, which can be beneficial when the data is needed closer to real time without delay, and it supports duplex capabilities. In a continuous stream, any number of bits can be sent or received. Data is transferred in packets across

I2C and UART, and each packet is limited to a certain number of bits. The beginning and end of each packet are defined by start and stop conditions, which makes it so the data is interrupted while being transmitted. With SPI communication, there is a serial clock, master in slave out, master out slave in, and slave select/chip select which is the line where the master selects what slave will be receiving the data. A basic configuration of the SPI system is shown below in Figure 24. The slave's sampling of bits is then synchronized with the master's output of data bits through the clock signal. Because each clock cycle transfers one bit of data, the frequency of the clock signal determines how fast data is sent.

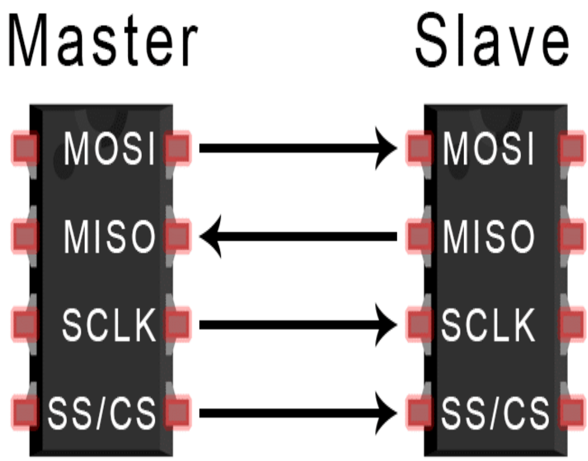


Figure 25: Simple SPI Configuration

There are four steps in the SPI transmission process. First, the clock signal gets output from the master and second, the slave gets activated because the master switches the slave select/chip select pin to low voltage. The third step is the master sending data to the chosen slave bit by bit along the master output line and if a response is needed, the fourth step involves the slave returning the data received bit by bit back to the master.

5.7.4 Protocol Comparison Table

Down below is Table 24 which compares key features about each protocol SPI, UART, and I2C. It shows advantages and disadvantages and functions of each protocol. It was important for us to make a side-by-side comparison table in order to make the best decision for our device.

Protocol	I2C	SPI	UART
Speed	Faster than UART	Fastest	Slowest
Number of devices	Up to 127, the more devices, the more complex	Many, also gets complex	Up to 2 devices

Complexity	Easy to chain devices	Complex as device number increases	Simple
Number of Wires	2	4	1
Duplex	Half	Full	Full
# of masters and slaves	Multiple slaves and masters	1 master, multiple slaves	Single to single

Table 24: Protocol Comparison Table

5.8 Software Selection

In the next subsections, various softwares are discussed in detail, discussing which ones were investigated for use in the effort to construct our chromesthesia simulation device. The main goals were to select software languages that are compatible with our microprocessors and familiar to us. Additional software was also explored and considered in terms of assistance for this project.

5.8.1 C/C++

Immediately, the C language was strongly considered for its familiarity, community support, and low-level operation that is well suited for software development close to the Machine language level. With our Microprocessor selection, C/C++ are the primary languages utilized for these boards, therefore this selection is only based on these similar languages. Furthermore, the main board that the device would be using has a practice development board, which we can use to implement code rapidly and see how it works on a functional level.

Code Composer studio is an official IDE based on C that can be used to program MSP430 based boards such as the one we would have in our final design and on the MSP430FR6968 boards that will be used for early-stage implementation and testing.

5.8.2 Python

The python language is a higher-level interpreter focused on readability. The main benefit of python is its intuitive design and its support for machine learning. Using the python language, we could implement the machine learning suggested for the more advanced features.

5.9 Hardware Part Selection

This section will go over the final Hardware components chosen to be used in the final design of our project. determining the final Hardware was a very important decision. this effectively blocks in the choices for the rest of the project and for a senior design 2. once the hardware components are chosen, the group would begin to procure and develop the design using these components.

To start, the microcontroller unit that was chosen was the MSP430 and Raspberry Pi. The reason we chose the MSP430 was because of its extensive support from the community and its easily accessible documentation provided by Texas Instruments. These two factors alone are incredibly beneficial to developing our project, especially once we start to add more advanced features to the device as development continues to move forward. The MSP430 is an extremely popular board, and it is likely that on an individual basis, the different functionalities, or at least, problems and projects that resemble our desired functionalities have been encountered and pursued in the past by other individuals developing on the same microcontroller unit. These individuals developing on the same MSP430 reached out to communities online and got help from them. Likewise, we reached out for help for roadblocks we ran into and additionally, we referenced the questions asked by people who came before us. Additionally, the MSP430 is a microcontroller unit that most people in our group have developed with before. This familiarity sped up Designing and implementing code into our project. We decided to mainly use the MSP430 for the audio processing and using the very high processing capabilities of the raspberry pi to process our visual inputs.

For the camera module, our group decided to use the OV7670 as the final camera module. the specifications of this camera are capable of the two key features that were sought after in a camera. These two features are resolution and low power consumption. The OV7670 achieves both and outperforms the other cameras that were under consideration. This was a very easy decision and we felt as though it was the best choice for our design.

For the microphones, this decision was a bit hard to make, especially since most of the microphones had similar properties. As mentioned before, the desirable properties the microphone will be a narrow cone with a long physical range of detection. Also, the microphone would be able to detect frequencies within the range of human hearing. Of course, the microphone should be affordable and lightweight. but at the same time, it must be power efficient. The microphone that we decided to use was the dB unlimited unidirectional Electret condenser microphone. this microphone seems like it had the best combination of capturing the correct frequency range and having a desirable cone of sound detection. This microphone also had decent power consumption, in the sense that it was efficient and low power consumption and was on par with the other microphones we are considering.

In addition to the condenser microphone, we also purchased some of the high sensitivity sound microphones that were premade and ready to connect to a breadboard. The reason we purchased some of these was for prototyping and testing in the initial stages of development. We did not intend to use this for the final device.

Finally, after evaluating the power demands of all of the systems whether it be using the exact component that we used or, referencing similar components that's surround the ideal component that we wish to implement, here are able to determine that we would probably need to lithium ion battery power supplies to power the system.

6 Project Hardware and Software Design

This section of the chapter focuses on the final process of our Chromesthesia simulation device. Discussion will include our approach to the design of both hardware and software.

6.1 Hardware Design Summary

For this section, it will go over the final determined parts and designs that were settled on for the final device design.

Component Type	Component Name	Power Consumption
MCU	MSP430	1.8-3.6 V
Microphones	dB Condenser Microphone	5 V
Power Supply	9V Battery pack	N/A
Camera Module	OV7670	2.5-3.0V
Display Screen	GeekPi 7 inch 1024 x 600 HDMI Screen LCD Display	2.6V

Table 25: Summary of Part Comparison

6.1.1 Hardware Design (block diagram included)

The hardware block diagram (Figure 25) shows all the components that would be needed and involved in the functionality of the device we are making. The device and all the components would be housed and integrated into a single chassis. The chassis would house the PCB and wiring, but there would be a couple of exceptions to what is housed externally on the outer surfaces of the device. The two exceptions would be the Microphone and camera plate, and the display screen

The display screen obviously needed to be on the outside, facing the user. It would be fixed into a cutout of the back wall of the container, and it will have the screen facing outside. This component would be sturdily attached to this opening so that is can handle being manipulated and swung around by the user. It would be fastened tightly to the device and should not disconnect when being used.

The same applied to the microphone and camera platform. As the name implies, the microphone and Camera platform is a single plat which houses the microphones and camera. The microphone array is positioned in a triangular orientation around the camera with the aperture of the camera right in the center of the triangle of microphones. The reason for this would be so that once we implemented sound triangulation, we would be able to know that the location of the image capture is right at the center of the sound triangulation, or at least as close as we can achieve.

With these components attached in the aforementioned orientation, they would be fastened to a plate which would then be attached to the front of the device, which is the opposite end of the device as the screen. This plate would be a solid, sturdy plate that can hold the weight of the components at a structural level. Since these components should not weigh too much, the material of 3D printed resin or the laser cut boards that we are considering would undoubtedly work fine for this purpose. This plate once fastened to the front of the device would have all the sensors (camera and microphones) facing outwards towards the environment ready to detect and gather data. The wiring will all be coming out of the rear end of the plate, leading directly to the PCBs (Printed Computer Board) and the MSP430s for the algorithms and processing.

Like the screen, the plate needs to be fastened firmly to the chassis of the device so that it can withstand being aimed and pointed by the user in a potentially rapid manner.

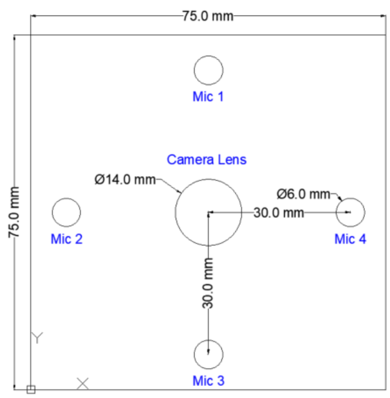


Figure 26: Front of Device

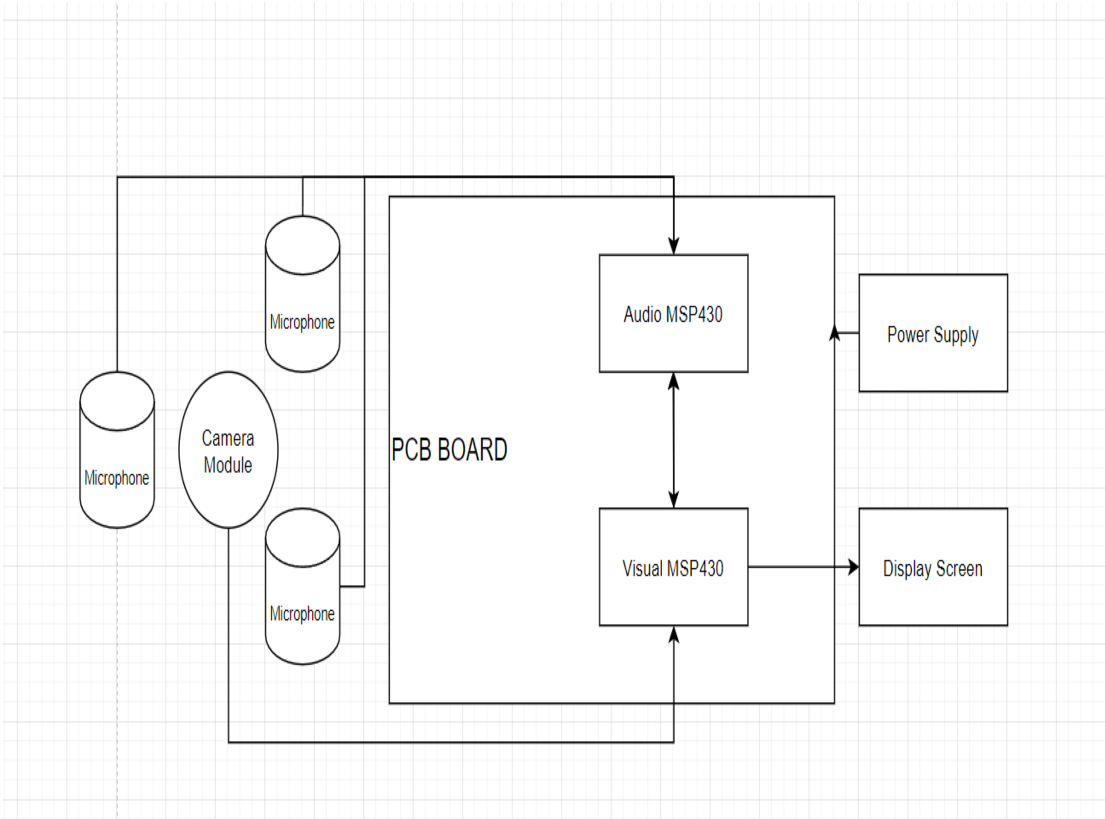


Figure 27: Components Needed for Device

6.1.2 Hardware Component Testing

Testing is an important part of the design process. Testing our components gave us valuable insight and understanding of our components and give realistic expectations and understandings of how to practically use these components. We have determined that testing has two main goals. These two goals are:

1. Understanding the component
2. Evaluating the component

Understanding the component means learning about the component and how it functions from a theoretical and a practical standpoint. Studying schematics and documentation of the device's design would be considered theoretical testing, and physically using the component, taking measurements, and applying the use of it would be considered practical testing.

Practical testing and theoretical testing are both equally important, but practical testing is marginally more desirable since direct connections can be made between the tests and applying what has been learned to the final design of our Chromesthesia Simulation Device. This brings us to a better understanding of the component, what it can do, and helps us determine whether this component is usable for the role we have in mind. Theoretical testing and studying can only tell us so much about the component; hands-on experience can bridge the gap between what has been learned on papers and applying it.

This second part is evaluating the component portion of testing. Once we have our hands on the component and get the chance to practically test it, we can evaluate the component and determine if it will truly be able to perform the way we have in mind.

6.1.2.1 Screen Testing

Screen testing was simple. Understanding the screen was the biggest part of the testing and procurement process. We wanted to make sure the screen was robust and provided the functionality we desired. The functionality in question was the ability to display the output imagery and with the desirable fidelity. The fidelity was dependent on the display's resolution, and the ability to display was dependent on simply getting the correct connection method. The only additional testing we would do is to potentially use this display as a real-time display for testing visual outputs. A scenario that this would be applicable would be to test it when testing the PCB board's ability to process and display images from the camera module.

6.1.2.2 Microphone Testing

The capabilities of the simulation device are heavily dependent on the fine-tuned nature of the microphone system and the underlying audio processing capabilities. Therefore, testing the microphones and obtaining a better understanding of their capabilities and what is needed was one of the first and most important parts of testing that was pursued.
For testing the microphones, we needed to do the two goals of testing: understanding and evaluating. To understand the microphones, we read the documentation and gained a deeper understanding of the design of the microphones. Then, we tested the microphones and determined how they worked.

The practical testing involved using a breadboard setup to connect the microphone to various other components to test it. The first test that was done was to see if the microphone worked. Figure 26 below is a block diagram of the setup we used for testing the microphone's functionality.

This test involved connecting the microphone to the breadboard and to an LED to visually determine if the microphone can pick up audio signals and to see if its output voltage responds in a way that makes sense. The microphone has three pins: Vcc - or Vin - ground, and V out. We conducted this test by connecting the Vcc and GND pins to the power supply, and then connecting the output voltage pin to the node of a LED light and connecting the other end of the LED light to ground. The results of this test were that we were able to confirm that the microphone not only worked, but it also responded to varying audio sources. We could also confirm the microphones worked to a degree of range.

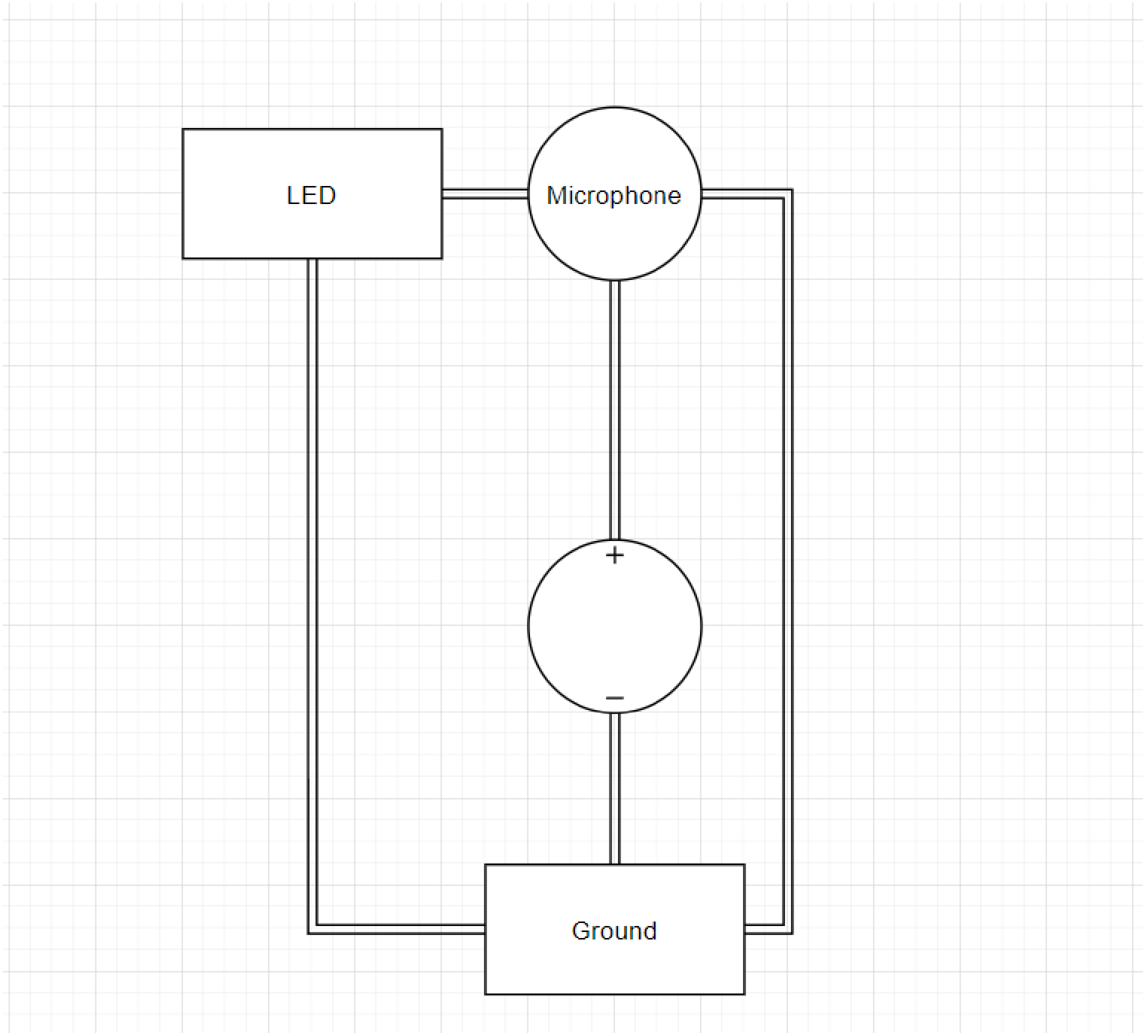


Figure 28: Block Diagram of Microphone Testing Setup

The next test was done virtually on Code Composer Studio. Because we are using the MSP430 and we still have the MSP430FR6968 boards from past classes, we decided to use these boards as a testing environment. This allowed us to use code composer studio to quickly create and implement code, thanks to familiarity and user interface.

We set up a test where we connected the three-pin microphone to a breadboard and using wires connected the breadboard nodes to the MSP430 board. We wrote code to read the input signal from the microphone onto the input pin and wrote a function to continuously write the values of the output from the microphone onto the command line. Although we could not explicitly determine what the values correlated to (i.e., whether the higher voltages meant higher decibels or different frequencies), we were able to read the output from the microphones onto the MSP430 and through an input pin.

Because our project was going to require reading the audio inputs and detecting their frequency, we needed to create an audio frequency spectrum analyzer or something similar to prove that it is possible to detect the audio inputs and the frequency of them.

Using the MSP430FR6968 was extremely helpful with the process of testing and evaluating components. It allowed us to test new code and software quickly and efficiently and rapidly make changes. Additionally, there are many resources and community discussions available online.

6.1.2.3 Camera Testing

Camera testing was the same as the microphone testing. Prior to ordering the part, we needed to evaluate it and make sure it had what we needed. We studied the schematics and details of the camera module on paper and gained theoretical knowledge and understanding of this component. The most important aspect of this was to understand how to process the image the camera would capture, and how to translate that into manipulatable data that our device can then use when it comes to applying the visual effects generated by the audio inputs of the microphone.

The same approach as the microphones would be used - starting with the most basic level, we tried to create the features. First, we would like to see if the component works. Like how the microphone was connected to an LED to visually show that there is a signal it is outputting, we did a similar test.

The next step would be to test the ability to read these inputs with the MSP430. The MSP430FR6968 would be our developmental testing platform in which we would be able to rapidly create and implement code to determine how to set up the camera module connected to the PCB. This way, once we create our own PCB, we would know exactly how to connect the camera module component to it in order to get the exact same output that we expected. The goal is repeatability of what we accomplish in testing and what we are building once we have the final components and PCB.

6.1.2.4 Chassis Testing

Testing the Chassis and the physical soundness of the device is just as important as testing the functionality of the components. Our device is by nature and interactable device that is expected to be interacted with in non-delicate ways. The Chassis needed to be tested if it is sturdy enough to handle being picked up, held, and moved around - all of these with a potential degree of violence. It is possible one user swings the device around with ample force and the device must be able to withstand these forces on its structural integrity.

Although the device will not be meant to be handled in a violent fashion, it is ultimately out of our control once it is in the hands of the user. Therefore, the device must be capable of handling anything within reason that might be thrown at it.
There are two types of effects that this device must be prepared to handle:

1. User intended forces
2. Non-user intended forces

User intended forces account for effects and forces applied to the structural integrity of the device that are intentional by the user. Despite the name, these forces are not necessarily malicious by nature. This accounts for when a user uses the device within its intended means, however, relative to their considerations, the way they use it is notably harsher than what a developer would prefer someone to delicately handle their device.

The second type of effect accounts for the moments that either the user uses or misuses the device in a way that it was not intended for it to be used. The simplest example of this would be if the user drops the device on the ground (this drop would be from anywhere between chest height and head height.

Both are potential ways in which unexpected forces can be applied to the device, and they must both be accounted for. Obviously, to overcome these, precautions must be made to construct a device that can avoid issues with these forces by doing two things:

1. Preventing and avoiding these effects
2. Withstanding these effects

What would be considered preventing and avoiding these effects are where the design comes into play. This means our team has taken measures and integrated it into our designs to make sure these potential forces are not encountered. This would include features like a sturdy handle fastened to the chassis so that the user has a stable and reliable point of contact with the device. Or, if the device ends up being design A - the wearable design - it should have a snug fit onto the user's head. These details are preventative measures that will be integrated into the design in a way that is natural and robust.

Withstanding the effects comes into play when the forces are applied. This is also dependent on the soundness of the engineering of the device. The most important aspect is a sturdy design. Although our device is not meant to be whipped around by the user or dropped onto the ground, the design is constructed to handle these traumas and result in zero or minimal negative effects on the functionality of the device.

To test these, the chassis needed to be rigorously tested - especially with the materials that were used to construct and bind the components and the chassis together. The materials used to construct the chassis are important and will thus be tested extensively. Depending on the material that will be used for the chassis, we conducted tests of the same wall thickness. We conducted drop tests at varying heights of smaller boxes constructed with the same materials (construction materials and binding materials) and determine which ones will be able to withstand the forces. The chassis ended up being made with a 3D printed resin, so we experimented with the thickness of the walls of the chassis to see which one has minimal cracking and breaking at varying heights. Ideally, we found the minimum thickness that can withstand the force of falling, while also being minimally thick so that we can minimize the weight of the device. Since the device will be handheld or worn on the user's head, it must be lightweight for comfortable extended use.

Table 23 below is a table of proposed thicknesses and drop heights for testing. We needed to take note to what degree the structure failed if it did – i.e., did it shatter, crack etc. We also considered what is a passable amount of failure. For this, we have created a table that quantifies the degree of failure (Table 24).

Height (inches)	**Resin Thickness A**	**Resin Thickness B**	**Resin Thickness C**	**Resin Thickness D**
48	pass/fail	pass/fail	pass/fail	pass/fail
52	pass/fail	pass/fail	pass/fail	pass/fail
56	pass/fail	pass/fail	pass/fail	pass/fail
60	pass/fail	pass/fail	pass/fail	pass/fail
64	pass/fail	pass/fail	pass/fail	pass/fail
68	pass/fail	pass/fail	pass/fail	pass/fail
72	pass/fail	pass/fail	pass/fail	pass/fail
76	pass/fail	pass/fail	pass/fail	pass/fail
80	pass/fail	pass/fail	pass/fail	pass/fail
84	pass/fail	pass/fail	pass/fail	pass/fail

Table 26: Proposed Thicknesses and Drop Heights for Testing

It is worth noting that this does not take into consideration the functionality of the device, just the physical well-being and structural soundness of the device. This is the most important concern with manual physical testing.

These drop tests will be conducted with mock chassis that represent the final form to the best ability, while still staying cost effective and cheap to produce. A weight will likely be added to represent the additional weight of the electronic components within and outside the chassis. The weights will also be positioned in a way that represents the weight distribution of all the electrical components to the best of our capabilities. This will provide an accurate expectation for what we need to achieve in terms of structural integrity and user-capable design.

Failure Level	Description
0	The chassis did not sustain any notable damage that can be perceived with the naked eye.
1	The Chassis sustained moderate damage such as notable scratches, dents, or deformations. Despite this, the chassis is still structurally sound and the damage does not impact the chassis' shape or structural integrity in a notable way.
2	The chassis sustained damage that is notable and might negatively affect the structural integrity of the chassis but in a minor way. This can include cracks going through the structure and deformation of the shape.
3	The chassis sustained notable damage that does negatively affect the structural integrity, such as major cracks and notable loosening of structural bindings.
4	The chassis was effectively destroyed and lost all structural integrity.

Table 27: Degree of Failure

6.1.3 Hardware Schematic

This section goes over multiple schematics used in the hardware systems of our final design. the schematics are of the system as a whole, and the individual components within the system. It is important to understand each individual component and document their schematics and how they work so that they can be referenced in the future at any point during the project. It is also helpful for any users who seek to read about and wish to understand this project.

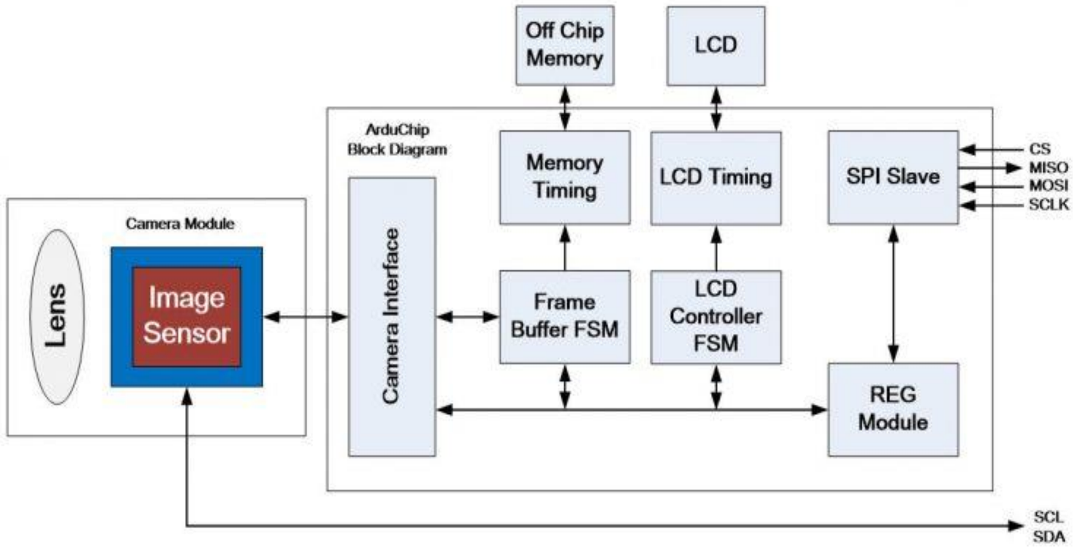


Figure 29: Camera Module Block Diagram

The following diagram depicts the microphone input array into the MSP430 to then perform audio processing. The microphones are attached to a preamplifier circuit to amplify the analog signal from the microphone. The resulting signal is then connected to the analog ports of the MSP430 to then undergo analog to digital conversion.

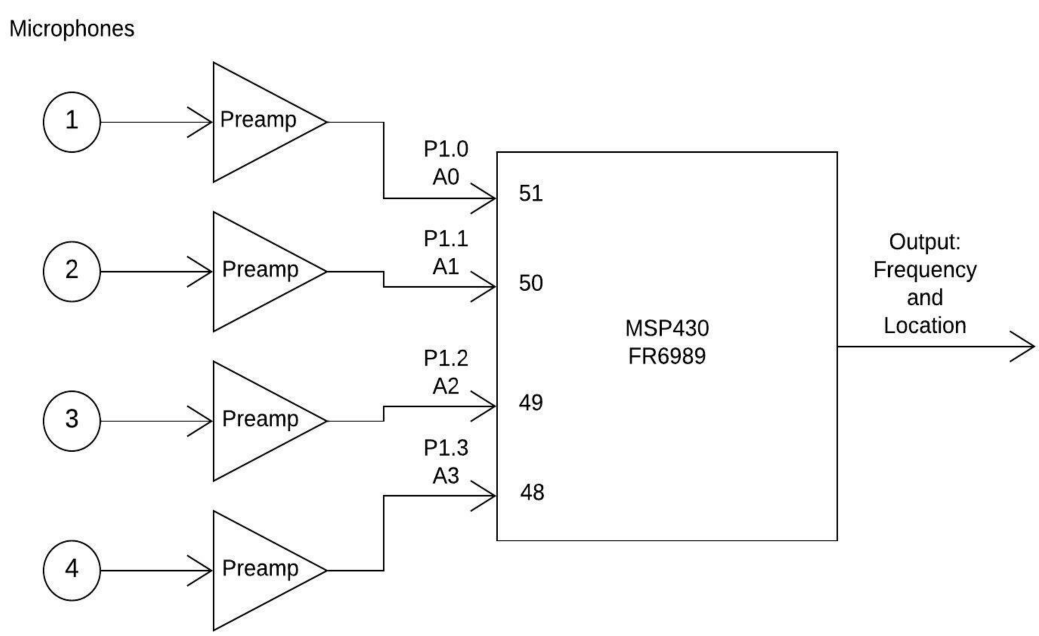


Figure 30: Diagram of Microphone Array to MSP430

The preamplifier circuit is shown below, using a MAX4466 microphone preamplifier. The circuit was obtained from the MAX4466 data sheet as a common implementation of the preamplifier.

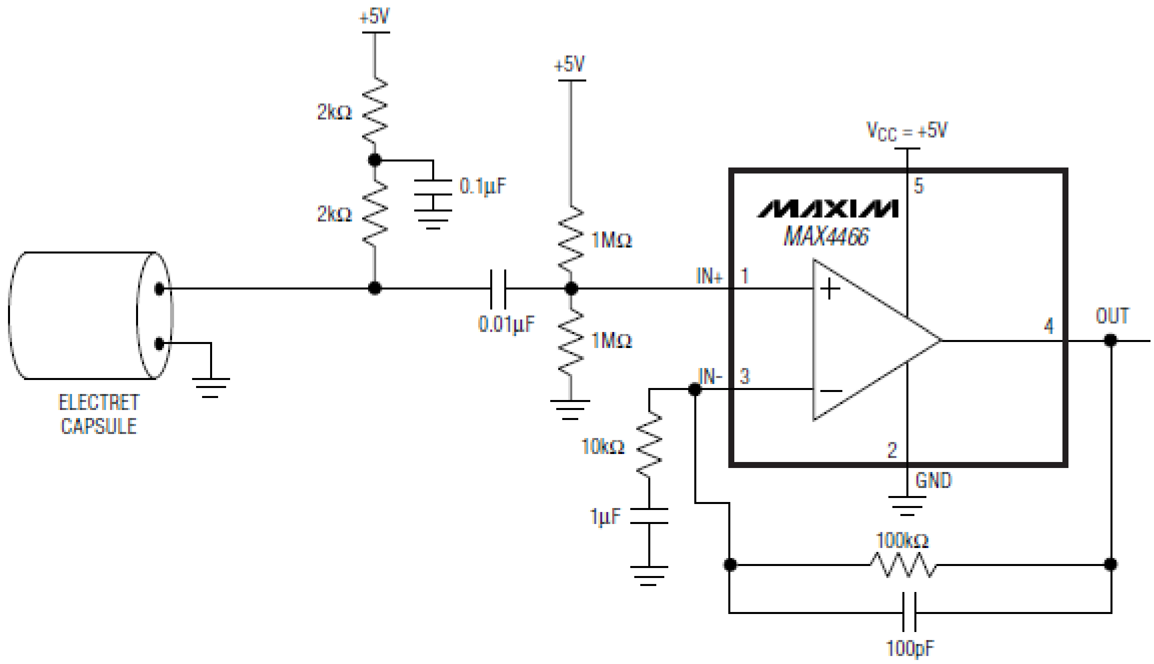


Figure 31: Microphone Preamplifier Schematic

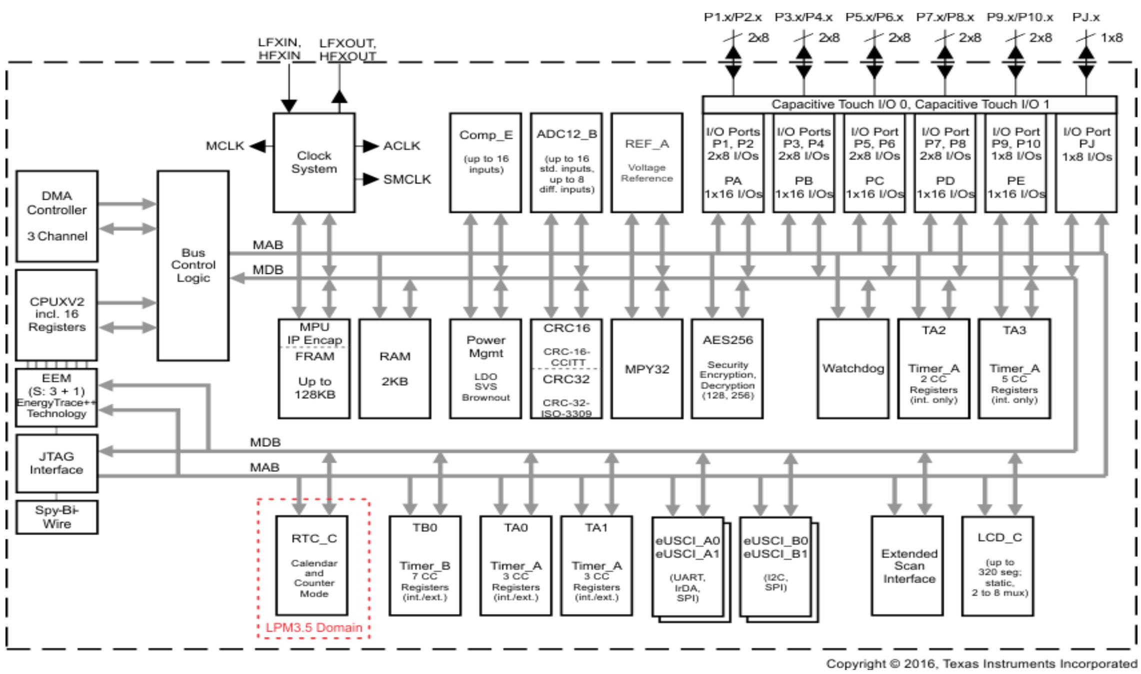


Figure 32: Block Diagram of MSP430

6.1.4 Power System Schematic

The power system is arguably one of the most important components of this device. due to the high computational demand of the audio processing and visual processing that will be performed by two microcontroller units both of which are MSP430s. Additionally, the printed circuit board will incorporate multiple microphones, a camera module, and a display screen - all of which will have a power supply demand that needs to be met.

The power system schematic attached in the section is simple, however, it is necessary to map out the power system for future reference and so that there are no inconsistencies or questions regarding what needs power and where the power is coming from. It is important to establish what needs power and how much it needs in order to determine the kind of power supply that is needed to support the system and all the functionalities that we have in mind for it.

Furthermore, this schematic will provide and easily digested representation of the power demands of all the components, devices, and features of the system.

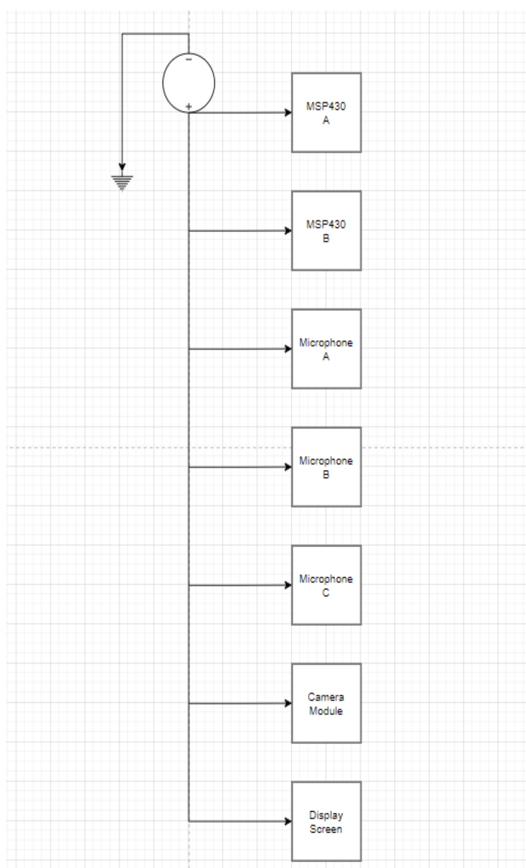


Figure 33: Power Supply Block Diagram

6.2 Software Design Summary

This Section will provide a comprehensive overview of all the software requirements and functionality of the device. Not only will this section explore the functionality of the software but will also explore the software that we used and also explore the programming languages and programming software that we used.

6.2.1 Software Language Selection

The C/C++ language was chosen as the main language for the software development of this device. C languages are robust and enable the development of complex code. Additionally, the MCUs (Microprocessor Units) that are being considered all support development in C and some even have IDEs (Integrated Development Environment) that can be used. C is a basic language, and most members of the project are well familiar with it or have worked with it in the past. Furthermore, for testing and developing the MSP430 using the MSP430FR6968 development board, we can use code composer studio, which uses the C language to write programs that can be quickly implemented onto our board. This will make development run smoothly and enable us to make design choices and changes rapidly

Additionally, python will also be considered for the development of any machine learning or neural network-based features that we might add depending on the timeline and progress that the group makes on the project. 1 potential idea for an advanced feature that we had was to include machine learning to identify objects that are commonly known as noise sources. one major example of this would be a face of a person. If our device could use the images captured by the camera to perform some neural network-based machine learning to identify objects in the image that are faces that could help us create some additional visual effects that could be considered Advanced features.

The training would be performed from the MSP430, and therefore we could use a more abstract programming language like python to develop this feature. The reason python is being considered is because python is a higher-level programming language that is commonly used for machine learning. There are many resources online concerning creating machine learning or AI (Artificial Intelligence) using python development. Additionally, there are many databases and reference libraries of things that create noise and sounds Which we could use for training the neural network. Once the neural network has been trained from the device, we could then Implement that code after trading into the msp430 and use it to identify sources of the sound. this in combination with the 3D sound localization performed by the microphones would enable us to get not only an even more accurate estimation as to where the sound is coming from, but also enable us to do more advanced features.

One example of a more advanced feature that would use machine learning to his Advantage is if we created some sort of visualization, such as making the cone of the Sound Source appears smaller or larger correlating to how far our neural network has determined The Sound Source to be away from the camera. For example, with faces, the neural network would be able to tell how far away from the aperture of the camera to face is and therefore be able to illustrate a larger or smaller representing the Sound Source.

6.2.2 Software Design

There are multiple parts of the software design process that are included in here (Figures 31 and 32).

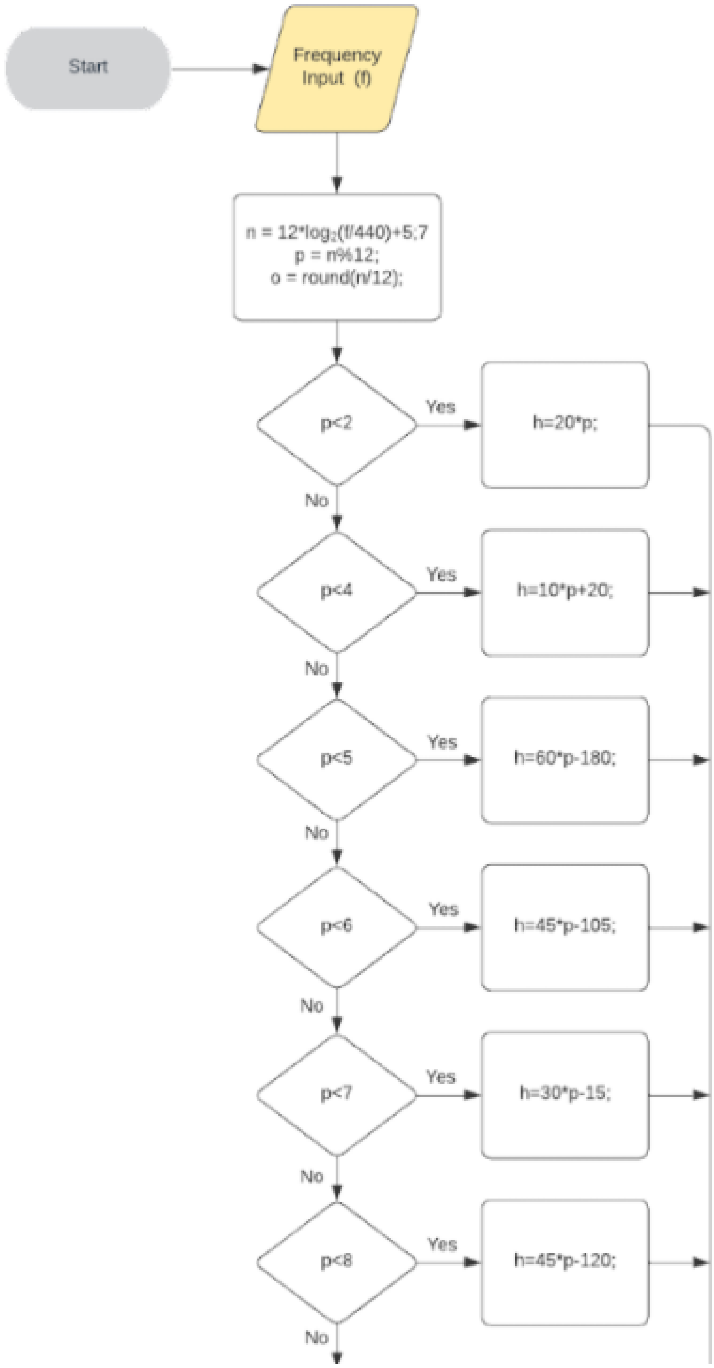


Figure 34: Part 1 Frequency to Color Process

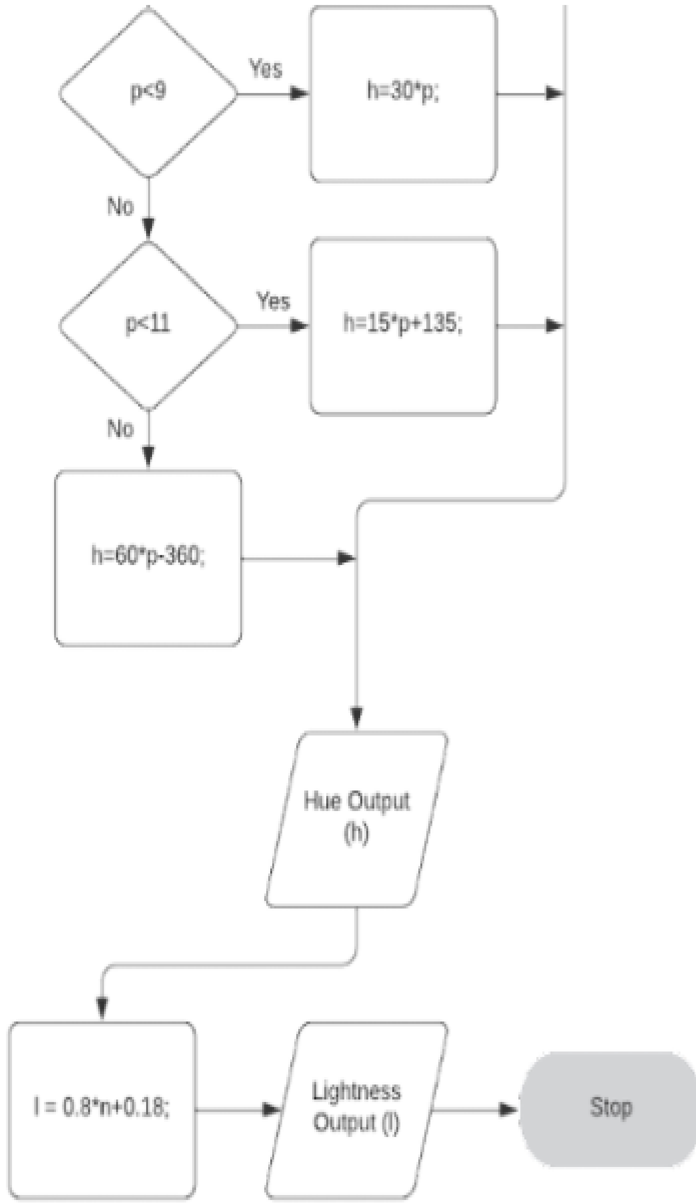


Figure 35: Part 2 Frequency to Color Process

6.2.3 Frequency-to-Color Algorithm Development

The frequency-to-color algorithm developed is based off of Case 12 from the study "Musical pitch classes have rainbow hues in pitch class-color synesthesia" and the color scale created by D.D. Jameson in the book "Colour-Music", where colored keys are used to create a sound-music color scheme.

The subject in Case 12 had an AP score of 91, was triggered by both syllable and pitch, and acquired synesthesia from playing with colored keys. This case seemed to best follow an increasing hue, as shown in the table below, while the other pitch-triggered subjects who acquired synesthesia in a similar way had a sudden change in hue around the pitch class “fa”.

Key #	Pitch Class	Color	R	G	B	H
1	C		239	67	61	2
3	D		248	150	58	29
5	E		233	231	65	59
6	F		85	185	71	113
8	G		96	199	207	184
10	A		47	86	166	220
12	B		122	81	161	271

Table 28: Case 12 Results

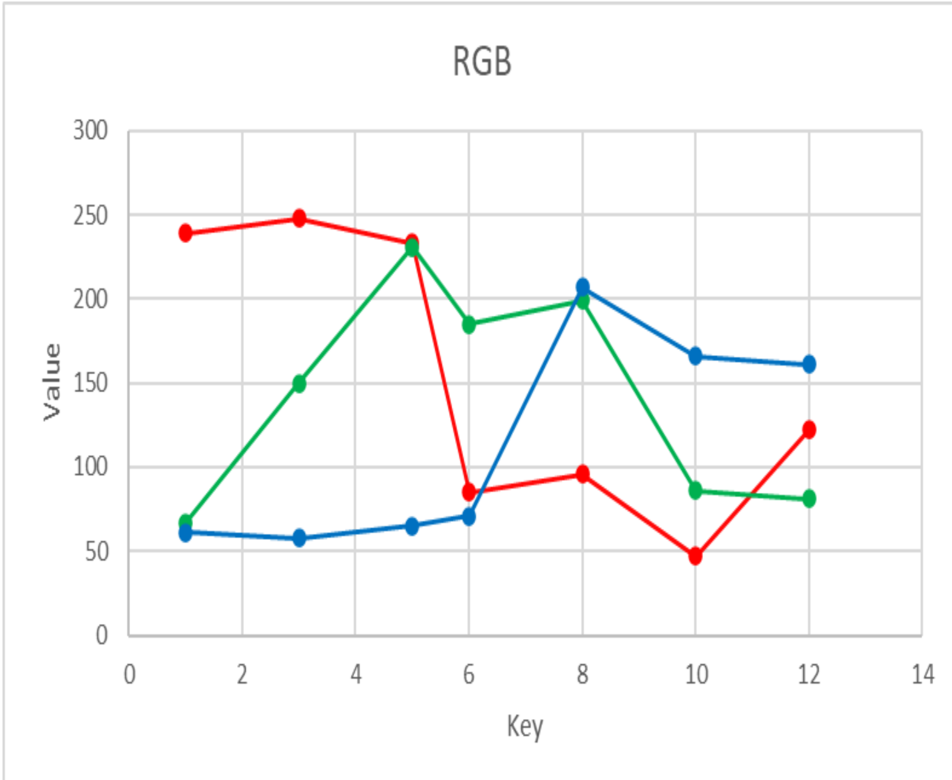


Figure 36: Color to Key Trend

Figure 37: Hue to Key Trend

The pitch-to-color scheme from the case study is similar to the colored key scheme created by Jameson, seen below, but omits the flat and sharp keys.

APPLICATION TO SOUND-MUSIC.

PREPARATION OF INSTRUMENTS.

PIANO-FORTE. — Papers of the several colours, of the different sizes, and in the order specified in the following Table, should be pasted on the keys of each octave of the common piano-forte.* *See Illustration.*

TABLE.

Colours and Semi-Colours, in their order.		Heights.	Widths.
Red.		Octave 1—2 Inches.	Whatever the keys will permit.
	Red-orange.	2—1½	
Orange.		3—1¼	
	Orange-yellow.	4—1	
Yellow.		5—¾	
Green.		6—½	
	Green-blue.	7—¼	
Blue.			
	Blue-purple.		
Purple.			
	Purple-violet.		
Violet.			

* A more durable mode is to dye the ivory; but then ivory should be substituted for the ebony now used for the semitonic keys.

Figure 38: Jameson Colored Key Scheme

6.2.3.1 *Key-to-Color*

From these two sources, the resulting pitch/note-to-color scheme was created.

Key #	Pitch Class	Color	R	G	B	H
1	C		255	0	0	0
2	C#/Db		255	85	0	20
3	D		255	165	0	39
4	D#/Eb		255	215	0	51
5	E		255	255	0	60
6	F		0	255	0	120
7	F#/Gb		0	255	191	165

8	G		0	191	255	195
9	G#/Ab		0	0	255	240
10	A		127	0	255	270
11	A#/Bb		191	0	255	285
12	B		255	0	255	300

Table 29: Pitch/Note-to-Color

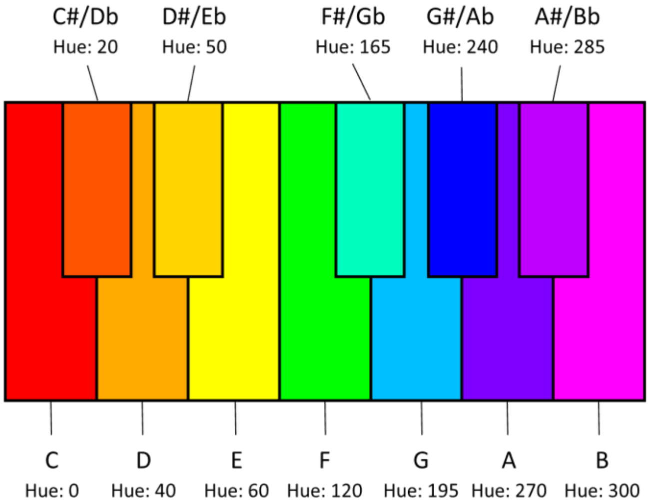


Figure 39:Hues Correlating to Piano Keys

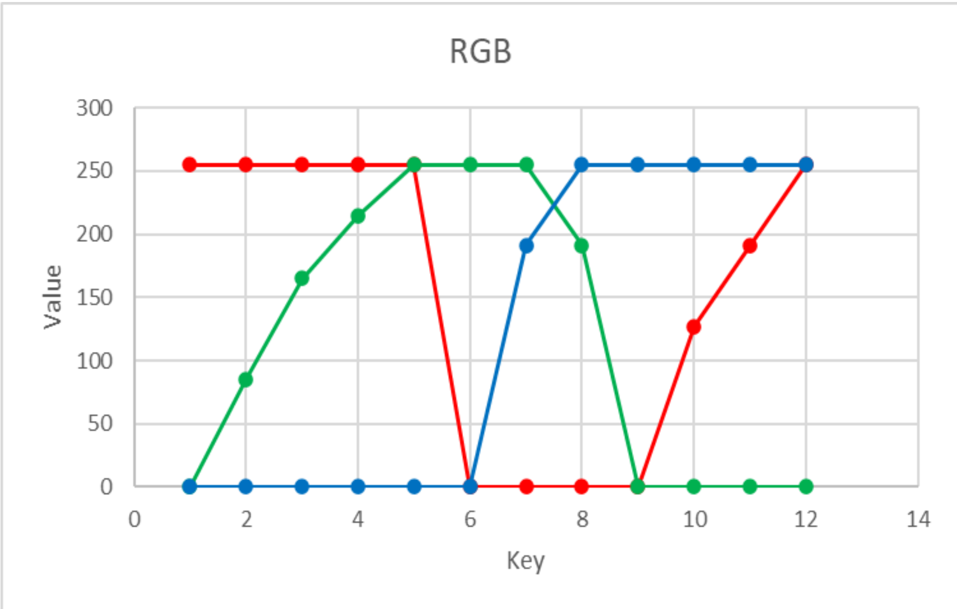


Figure 40: RGB to Key Trend

Figure 41: Hue to Key Trend 2nd

Luminance/lightness of the color is then chosen based off of the octave of the key. The "middle" octave of a piano is the 4th octave, so the lightness for that value is set to 0.5 (range is between 0 and 1). The lightness increases by 0.08 as the octave increases and decreases by 0.08 as the octave decreases from the "middle" octave. Lower octaves have a lower pitch and frequency, while higher octaves have a higher pitch and frequency, so darker and lighter colors are then correlated to these octaves.
The resulting color scheme can be seen below in Table 27.

Octave #	Luminance												
1	0.18												
2	0.26												
3	0.34												
4	0.42												
5	0.50												
6	0.58												
7	0.66												
8	0.74												
9	0.82												
10	0.90												

Table 30: Correlated Octaves

6.2.3.2 Note Identification

The following equations convert frequency to the numbered key on a piano.

Key number (on 88 key piano) to frequency equation:
$f = 2^{\frac{k-49}{12}} \times 440\ Hz$

Inverting the equation:
$k = 12 \log_2\left(\frac{f}{440}\right) + 49$

Offset by 9 so C at octave 0 is at n=0
$n = k + 8 = 12 \log_2\left(\frac{f}{440}\right) + 57$

6.2.3.3 Algorithm Equations

The following equations are used to identify pitch class (note in an octave) and octave and then find the correlating hue and luminance/lightness.

Frequency to key number:
$n = 12 \log_2\left(\frac{f}{440}\right) + 57$

Key number to pitch class:
$p = n\%12$

Key number to octave:
$o = Round\left(\frac{n}{12}\right)$

Pitch-class to hue:

$$h = \begin{cases} 20p, & 0 \le p < 2 \\ 10p + 20, & 2 \le p < 4 \\ 60p - 180, & 4 \le p < 5 \\ 45p - 105, & 5 \le p < 6 \\ 30p - 15, & 6 \le p < 7 \\ 45p - 120, & 7 \le p < 8 \\ 30p, & 8 \le p < 9 \\ 15p + 135, & 9 \le p < 11 \\ 60p - 360, & 11 \le p < 12 \end{cases}$$

Octave to lightness:
$l = 0.8n + 0.18$

6.2.4 Software Design Flowchart

This section will cover the flow and movement of data from point of entry via input from microphones and or the camera module all the way to the output of the display screen. This section will attempt to go over all of the features and all of the functionality that will be happening within the system between these two points.

The first flow chart (Figure 39) covers the acquisition of audio data and how it is processed. The audio processing procedure involves noise cancellation, Fourier transforms, 3D sound localization, application of frequency identification, receiving an input from the camera module inputs and using this to determine a correlation between 3D sound localization and the pixels on the display captured by the camera, and finally an association between the detected frequency and the respective color paste on our Chromesthesia algorithm.

The flowchart (Figure 40) covers the visual side of the software. Beginning with the capture of an image, a single frame is collected with the camera, and it is sent to the MSP430. some function will determine the size of that image communicating the width and height of every frame a retrieve and sends this to the audio handling MSP430. Once the audio processing MSP430 completes this process, it will send its data back to the visual processing MSP430. the visual processing MSP430 well then use the data to redraw the image with the desired visual effects and apply it. Then, it will send this new image to the display and output each frame.

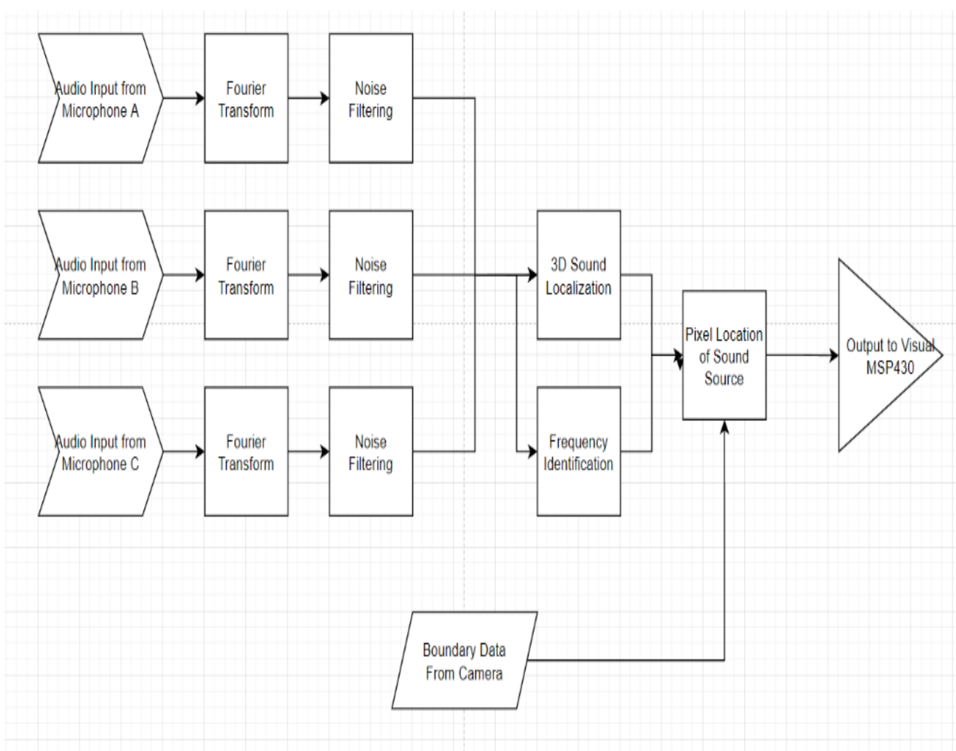


Figure 42: Audio Software Flowchart

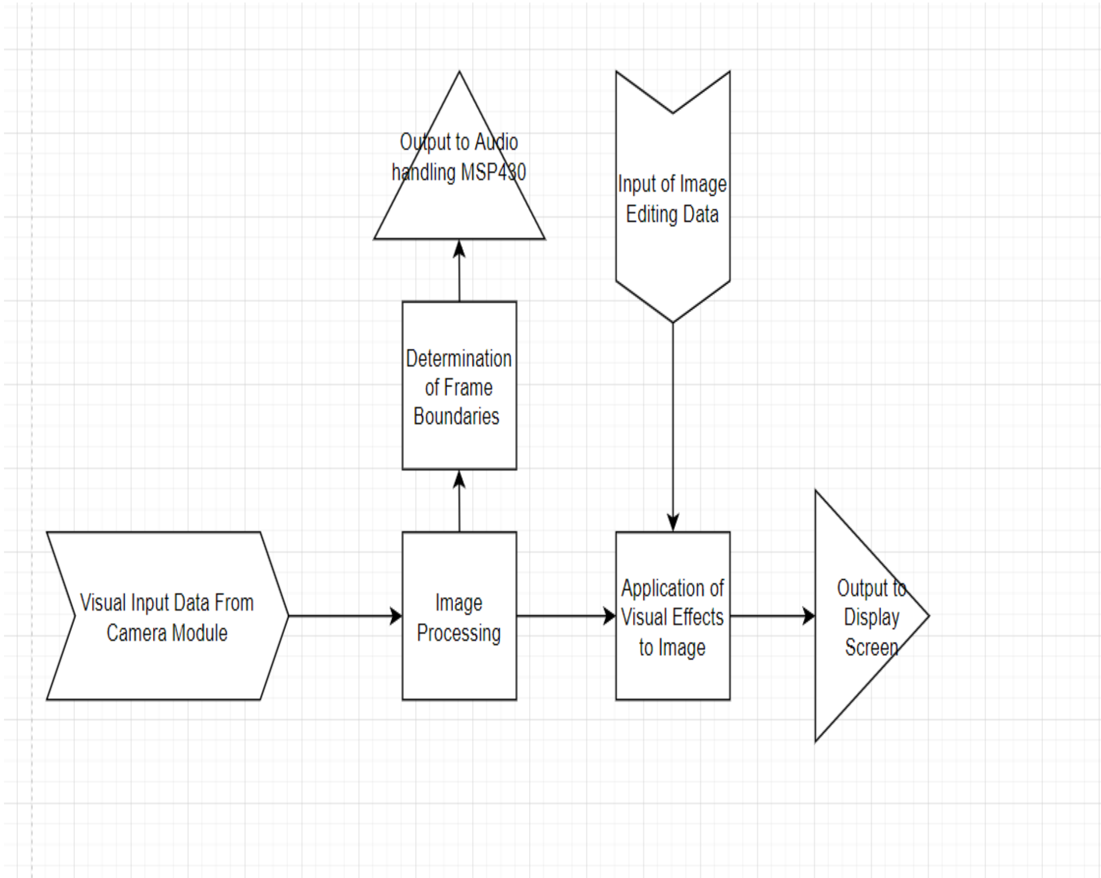


Figure 43: Visual Software Flowchart

6.2.5 Software Block Diagram

This section will include a basic block diagram of the software functionality at a high-level of understanding and abstraction. This block diagram highlights all the basic functions and Communications between the software components. This was one of the first block diagrams related to the project that was made. It has proven to be crucial to the roadmap of this project as we often reference and utilize it as a form of guidance. This has been an especially useful reference when it came to keeping direction for software development.

As can be seen, right from the get-go it was understood that there would be two major processing functions. These include audio and image processing from both input sources which are the microphones and the camera. It is diagram also accounted for how the 2 channels of data flow would eventually intersect and at what point they would need to share data between each other in order to produce the final outcome of the device.

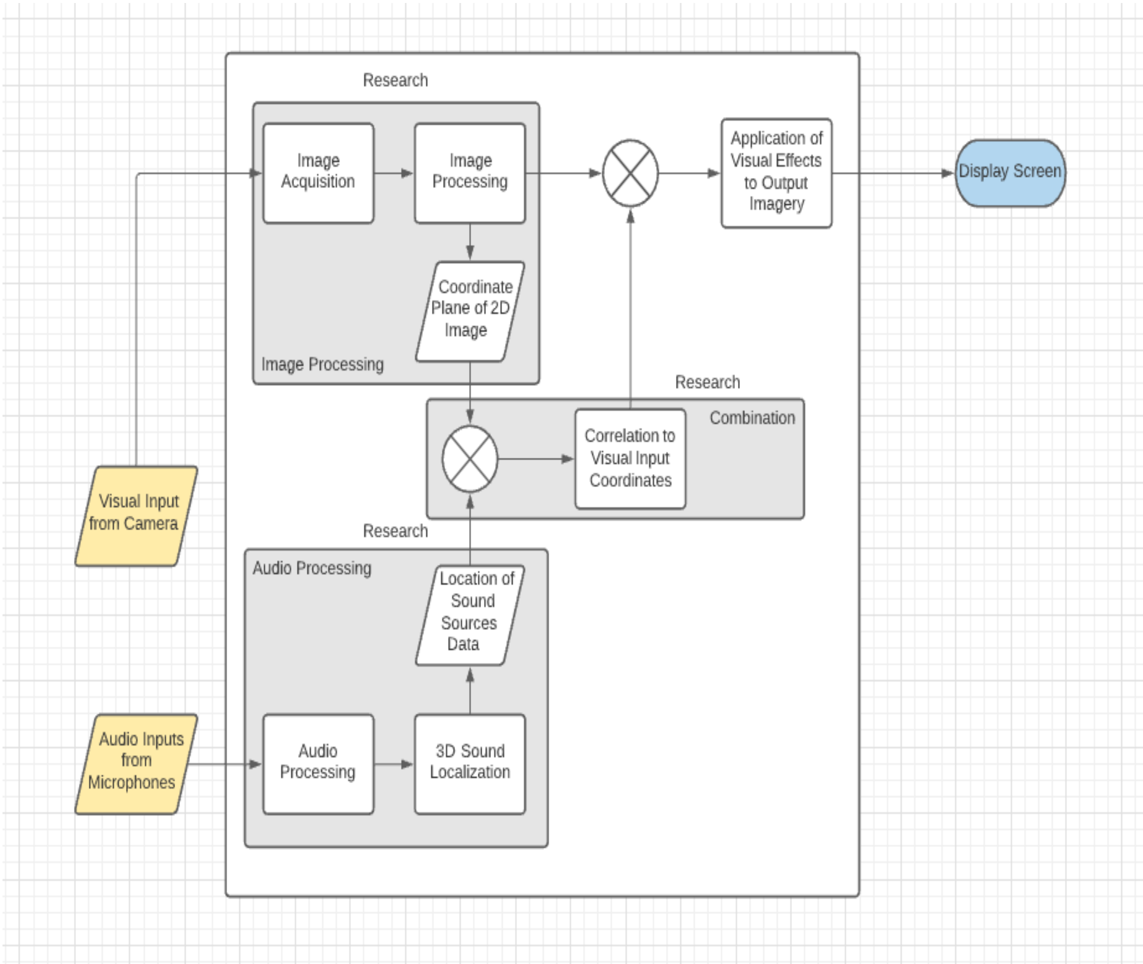


Figure 44: Block Diagram of Software Overview

6.2.6 Software Interaction with Microcontroller

This design will have two microcontrollers, the msp430 and the raspberry pi, integrated into the printed circuit board. One microcontroller will be responsible for handling audio processing and other audio related functions, and the other microcontroller will be responsible for handling image processing and other image related functions. The reason behind this is because audio processing and image processing are both known to be capitation Ali rigorous processes. Therefore, our group thought it would be beneficial to split up the workload between two microcontrollers. the microcontrollers will be interacting with these different inputs by running code and Logic on them. one of the main hurdles that will be investigated further is the timing between both microcontrollers. For this to work effectively, both microcontrollers must be working in tandem with each other and be on the same steps of the process to ensure that they produce the correct and desired output.

6.3 Build Design Summary

To summarize, the printed circuit board will incorporate one MSP430 microcontroller and one Raspberry Pi. The MSP430 will handle audio processing and the Raspberry Pi will handle image processing. In each of these microcontrollers will be programmed to perform logic relevant to both forms of input that they are handling. Once they both have been processed in their own respected microcontroller, the microcontrollers will share data and combine it to create one single output on the display screen. The aim is to perform this quickly, efficiently, and accurately.

7 Project Prototype Construction

We'll break down the main notion of how we'll build our circuit board for initial and final testing in this chapter. In this section, we'll go over the specifics of our build, the reasoning and logic behind our decision, and any of our worries or roadblocks that we'll be looking out for during final testing. KiCAD will be addressed in further detail as design software, as it was sufficient in our operation. The following sections will cover: an in-depth look at how our design was created and how various components worked together, how our chromesthesia simulation device should behave, the installation of the PCB, and a progression of the PCB designs that were created by the software we chose.

We look to accumulate every single aspect of our design, from our microcontrollers, microphones, camera

7.1 PCB

The PCB is one of the main features of the device. The printed circuit board holds a critical role in the functionality of our device. Our printed circuit board will involve MSP430 MCU and a Raspberry. In addition to these two microcontrollers are all the connections between them and the other components on the PCB. These components include but are not limited to the microphones, the camera module, and the screen. In addition to these components there will undoubtedly be many wires, resistors, capacitors, transducers, and other digital logic components that are required to perform the desired functionality.

The PCB is a small size because we designed it to be able to sit on top of our raspberry pi. This is an important design detail, especially since it saved us plenty of space and resourcesa.

For the Prototype, we are prioritizing functionality over Hardware accuracy. We were able to use the pre-built developmental boards in a way that would be true and accurate to how the final PCB design will look. However, if the time frame permits, we attempted to try to print a PCB for our prototype. Obviously, a PCB specifically printed for our device would be a much more accurate representation of how it would work. This would be ideal; however, functionality is the main feature that we want to demonstrate in the Prototype.

Because we developed the functionality primarily on the MSP430FR6968, we could easily use it and integrate it into the rest of the device in a semi complete form. Although this wouldn't be technically accurate, we would say that it would be accurate in the form of functionality. Furthermore, thanks to the extensive documentation and community use of the MSP430FR6968 board, we would easily be able to translate this pre-made board's functionality to our custom-made printed circuit board's functionality.

7.1.1 PCB Design

The PCB is one of the most important components for our chromesthesia simulation device. Despite our devices' interactions with the physical world, the majority of the heavy lifting will be done on the PC be with interpreting inputs from electrical components and Performing algorithms and other software functions to achieve the desired experience that we want the users to have. The way the process is going to be for our project is we are going to design the PCB in a PCB software environment, then send over the schematics to a company that will assist us in making the PCB. As long as we get the order done in time, the PCB will be constructed to exactly what our device will need, and will avoid any unnecessary components.

Center device is heavily based on collecting visual input data and audio input data, and both of these processes are notorious for being very demanding computationally, we decided to use two microcontroller units, which we've decided will be msp430s. 1 msp430 will handle audio processing and other functions related to it oh, and the other one will handle all visual processing and other auxiliary functions related to it. The reason for this is to ease the load of each microcontroller by splitting up the work between the two. This way, not 1 microcontroller will be responsible for all of the work. They will both be able to work together to provide the desired functionality. Although there are certainly benefits to this, there're also drawbacks and potential issues that might arise, these are discussed in a later section.

To design our board, we needed to pick a PCB design program that we would use. There are numerous PCB design software out there, ranging from free in price to a very high cost. The first factor we considered when choosing was to make sure this software would remain in our budget. Afterwards, we would consider the benefits of the software, such as if it cost a little more, what benefits would we gain out of it? From there we would assess the options. After much consideration, our group decided to go with Eagle mainly because half of our group members were already familiar with the software due to their experience in Junior Design. They learned how to design board layouts, schematics, importing schematics, and generally use many of the features within the software. Another big plus, was that we had access to these labs on campus, so we could reference any old material learned to help guide us along the way. Eagle also consists of many features that we used for the construction of our PCB design, but ultimately went with KiCad.

In addition to the two MSP430 s, the printed circuit board will also include connections to the microphone array and the camera module. Wiring for the power supply will also need to run through the PCB. Power delivery is going to be extremely important especially since there will be two microcontroller units doing very demanding copy stations constantly for prolonged periods of time. The power delivery system should be robust and reliable. Therefore, we strongly looked into how to properly implement the power supply into the PCV and determine how to maximize the efficiency with the PCB design. Depending on the microphone design that we decide to use, we might have to create our own transducer circuit as a microphone. Depending on constraints such as budget, we might decide to include the circuitry for the microphones on the main printed circuit board. The reason for this is that it could reduce costs by reducing the amount of printed circuit boards we would have to order. If we just need to order one circuit board that has everything on it, that would be much less expensive than ordering one large circuit board and three smaller ones that would just be used for the microphones. Or, it might be the other way around. This is definitely something that we researched and investigated as we got closer to printing our own circuit boards and ordering the components and designs online.

7.1.2 PCB Fabrication

Once all the components are finalized and all of the circuitry logic is determined, our group will use the Eagle printed circuit board development software that we used in junior design one. Using the KiCad development software, we were able to select components from the parts list, design our circuit board and order all of these components and the board itself. This step should be fairly straightforward and ideally, we knew exactly what components we needed and where they will go on the PCB by the time our design was finished, and we were ready to order.

It is important to note that designing and ordering a PCB takes time, and unfortunately, we did not have the exact PCB that we want to use by the time our prototype needs to be used for demonstration. Because of this, we relied on the MSP430FR6968 developmental board that several team members have ownership of.

7.2 Prototype Expectations

Throughout chapters 4 and 5, the construction process is described. Various additions, board designs and considerations, and theoretical build configuration on how we expect the operation to be accomplished in practice have been covered in these chapter subsections. In this section, however, we must also consider the prototype's potential hurdles and failure methods. This section will provide in-depth information on the types of failures we expect to experience, as well as how to limit the likelihood of failure and the impact it may have on device performance.

The prototype is an important milestone in this project. The prototype needs to be able to demonstrate in real time the functionality of our device and provide some sort of semblance of what our device will do and invite people to speculate on what else it could do. As previously stated, the Prototype will not be exactly like the final version. However, we would like to achieve a prototype that comes as close to the final version as possible and attempt to represent it as best we can.

To do this the main goal is to achieve an accurate functionality that properly represents what the final product will not only include but build upon. The main functionality that the device promises to deliver is the ability to detect sound. Audio inputs are going to be detected by the microphones and should be able to deliver a signal to the circuitry inside the printed circuit board. The printed circuit board should then be able to perform audio processing on the signals and inputs from the microphones, then be able to use the signals to modify them and use them to modify other variables. There will be a variety of functions within the software that will contribute to processing these which will be explored in the next section, but for this section let's only discuss it on a surface level matter.

In tandem with the audio processing with the microphones, the prototype needs to be able to collect images from the camera module and interpret the input data from the camera itself. In addition to the microphones, another main feature that the Prototype should include is the ability for the camera to intake images and send them to its own microcontroller unit. As mentioned before, this design will involve two microcontrollers which are both msp430s.

We would like to demonstrate how the audio signals will go to one MSP430, and the image signals will go to the other. Although this isn't a feature that is necessarily demonstrable to the common user which we would expect to use this device, this is an important feature that aren't we hope our device will use. The reason why is because audio processing and image processing are both very taxing processes on computational power. This would be an important milestone in the Prototype because we can show that the primary technological functions that are happening within the device are indeed happening and working as intended.

The prototype development should be able to demonstrate in real time the functionality of our device and provide some sort of semblance of what our device will do and invite people to speculate on what else it could do. As previously stated, the Prototype will not be exactly like the final version. However, we would like to achieve a prototype that comes as close to the final version as possible and attempt to represent it as best we can.

Within the MSP430s, we would like to perform the relevant functions and software for each respective processing i.e., image processing or audio processing. This can be demonstrated through the output which will take data from both the image input and the audio inputs and use it for functions that can then produce a final output of our desired function. This output would be in the form of an image displayed onto a screen. The image is edited. It takes the visual inputs collected by the camera and edited using functions performed using the audio input data. These functions include but are not limited to determining the frequency of the collected audio, noise detection end noise canceling, auditory triangulation, correlation of frequency to a specific RGB color output, identifying the pixels of the image of which the microphones were able to triangulate the location of the supposed audio source, and more.

7.2.1 Potential Software Issues

This section will explore the potential software issues that we could experience during the demo and in the prototype. The prototype will most likely not feature a complete and final version of the software the final version will include. However, the aim of the Prototype will be to include as much as we can, especially concerning the basic functions of the device.

All software-based projects encounter a variety of potential challenges that must be addressed; the most important for this project are reliability, accuracy, and performance speed. Because of the restricted resources and debugging capabilities, enabling real-time computing on embedded systems might cause problems. These few critical characteristics must be tested upon when developing and implementing software into hardware design. The first issue to check for when beginning project development is microcontroller connectivity in order to import our software program, and the list that has to be verified includes power supply, right serial port, drivers, and any pull ups. The first step in programming a microcontroller is to connect it to a computer via a USB port; if the microcontroller does not receive adequate power, an error will occur. After we've taken care of the power, we can move on to picking the correct serial port, which is crucial. An error will occur if the program's communication port is not specified. The third step is normally handled by the application, but if a driver fails, a manual installation may be necessary. Pull-ups are required for many microcontrollers before they can be programmed.

We can concentrate on microcontrollers talking with one another once all of the basic connectivity and Wi-Fi difficulties have been addressed. When working with many microcontrollers, the issue of proper communication between the devices emerges. For example, suppose that the same device or address connects to or accesses the same memory. Assume one device operation accesses the memory address where it writes/reads; the problem occurs when one reads while the other writes. In this case, the read command would be dependent on the execution of the write command. If this happens, the calling function will be reliant on the return data, and the behavior will be inappropriate.

Once all of the basic connectivity and Wi-Fi issues have been resolved, we can focus on microcontrollers communicating with one another. Observing this situation, it is important that we factor in this situation as a possibility, especially since this project is going to rely on software quickly calculating and deciphering live inputs. The read command would be dependent on the write command's execution in this situation. If this happens, the caller function will become dependent on the return data, resulting in undesirable behavior. Preventing this situation can happen by debugging code regularly and having a sufficient software test plan in place.

There are plenty other mishaps that can happen, thus using flowcharts and uML diagrams can help avoid some of them. Finally, in embedded processes, it is important to avoid stack overflow. A stack is a temporary working memory buffer that is tracked in memory. The stack is crucial because it maintains track of what needs to be processed next when the processor is available. An illustration of how stack works can be found below in Figure 42.

As an example, it shows how buffer overflow can influence any type of software, when a transaction overwrites executable code, the software program can behave in an unpredictable manner.

Figure 45: Overflow of Data

As a result, overflow data safeguards are required to maintain memory protection, and the stack can expand its allotted area. A few safeguards, such as stack pointers with test and measurement methods, will be used to do this. Stack guard zones, timer interruptions, and test cases are all used. We may avoid stack overflow by following these guidelines, ensuring that the application is constructed without errors.

One of the main concerns on a software level for this project is the fact that it will utilize MSP430s. There are clock cycle and timing constraints that we took a look at, and worked around and ensure that they are accurate and on time. There will be moments where both msp430 s will need to communicate with each other in order to achieve their functionality. For example, when the visual msp430 receives data from the audio MSP 40, the visual MSP 4030 will need to use this data to apply visual effects on to its output imagery. The concern here is that the audio and visual signals both correspond to the same time. It is ideal to make sure that every frame captured by the camera module will also correlate to a specific instance in time of audio input. That audio input and frame capture must have never gone out of sync. If it does, this will lead to many issues of timing. With timing issues, this might cause the machine to apply incorrect audio data to incorrect visual data. Both of which would not match up.

An example scenario in which this might be an issue is if in one frame there is a picture of a person speaking towards the camera and microphones. The audio and visuals of this specific instance and time will be collected oh, but what if they get out of sync? What if the next frame immediately after this one is of something completely different, for example a car?

The sound from the car comes from its exhaust pipe, meanwhile the sound from a human comes from their mouth. Assuming that the camera and the device are aimed at both objects in a similar fashion, the sound sources will come from different points on the screen. However, if it gets out of sync for the frame that captured the car, it might highlight an area in the top center of the screen - somewhere a person's head would be located on a normal frame. However, since the frames are out of sync it would be displaying the frame of the car. Assuming the car is facing towards the left side of the image, the sound of the car should be emerging from the right side where the exhaust pipe is. If the color highlighting for the person is accidentally applied to the image of the car it would simply create an incorrect output. Although this is a very far-fetched example, it is a relevant concern. There will undoubtedly be moments where one frame is drastically different than the next.

Another software concern is the accuracy of the audio processing. Audio processing and accuracy of it in terms of representing the frequency measured is one of the most important baseline components of this device. Without proper audio processing, or accurate audio processing, the entire device might be considered ineffective. For this reason, the 3D sound localization and the audio processing functions need to be bullet proof in terms of their functionality and execution. Because this part of the design is extremely important and sensitive, it is highly unlikely that we got it perfect the first time. Because of this we can expect there to be issues at this step of the processing. Hopefully, this will be more of a matter of accuracy rather than a matter of functionality or execution. If it is a problem with execution or functionality, then this means that we would have to rework the design of the software. On the other hand, if this is an issue of accuracy, we would only have to fine-tune our algorithms and software.

7.2.2 Potential Hardware Issues

Hardware components are an extremely important factor for this project. Issues concerning shipping or manufacture of hardware parts can be essential to the overall development of the device in a timely way, especially as time is our biggest enemy in this endeavor. If parts are delayed, sold out, or damaged throughout the manufacturing process, not only will production be slowed, but it will also affect the flow of other portions of the software development process, as one sector cannot continue without the other. Fortunately, most product suppliers have a big supply of the parts we require, and shipping and accessibility resources have permitted a regular flow of goods at cheaper rates to avoid disruption. All of the parts are widely available, and there are a number of options for alternatives.

Other hardware considerations include the danger of laboratory errors resulting in product harm. We must account for this issue, whether these items were dropped, soldered in an incorrect manner, circuit damage, or a manufacturing defect, these errors are to be expected. The easiest way to combat this is to purchase multiple items of whatever we are using in large quantities if possible, or at the very least have enough stock to account for the possibility of damage. Because most of these components are easily available, they will help us save money and time in the long run. Having an emergency spare part stockpile reduces costs in the second phase of Senior Design, and it will allow the production process to be efficient, because time is our most precious resource.

The prototype's structural integrity may be jeopardized if the board design fails. The board's specs must be produced to our exact specifications because it is the backbone of the operation that powers and conducts the components connected. Given that these components are our most expensive and valuable, the time delays and financial expenses associated with catastrophes or miscalculations could significantly stifle our progress. Using software such as Eagle, we can check every single aspect of the board and determine exactly where things need to be so when the time comes to have our PCBs made, the manufacturer or company has a good idea of what we need and what we are asking for. It is our job to thoroughly review every detail of our board before having it produced. As a result, the fewer foreseen errors we can avoid, the better off the group will be, both financially and in terms of time.

Sensor failure is the second most common cause of PCB failure. It's an issue when our devices don't do what they're supposed to do. Since we are working with a sensor-sensitive device, timing and precision are critical for the data we produce to maintain structure and consistency in the prototype. The cornerstone of how the device will work is miscommunication in voltage differences between the numerous sensors interacting or transitory layering of sensors where one action affects the next. If this happens, we have no data and consequently no control over the device until the sensors are working. The sensor arrangement must be divided into levels of implementation. Setting up stages or hierarchy levels in which sensors are configured to work in a building block stage. Once one sensor has met our expectations, we add another level to function at the same level if they are linked, and we test both sensors gradually again. This approach definitely takes a little longer in the beginning, but it will take less time in the end if each layer has been thoroughly tested and is known to work without errors. This method will also reduce the risk of financial losses since were able to improve each component layer by layer to avoid any unanticipated problems.

Some other minor but often undiscovered faults include mistreatment of electrical equipment and components. This can result in damage that can go unseen towards the PCB and any electrical devices that are connected internally or externally. All of the wiring, connections, and maintenance of the board should be consistent and be kept free of dust and dirt at all times. Dust and dirt can cause trips, shorts, and could potentially damage other components of the device over time. There are proper procedures for cleaning components and maintaining them such as using compressed air to blow dust off. One could also use disinfecting wipes that are safe for any electronic devices, components, etc.

There must also be considerations for spatial concerns. The layout of our electrical components and non-electrical parts must work together seamlessly. Internal damage to parts like our microphones burning out owing to being stuck or not correctly housed could be caused by improper sizing and measurements of various housing sections of the gadget. As the project continues, we must make sure we know where each component will be and how it will be placed with the required specifications. The more organized the spaces are, the more efficient our board can run without being hindered. To add on to this, during the duration of the project and development of the prototype, it will be constantly tested, unplugged, and reconnected. One must take notice and be careful of any wires or plugs that are being handled incorrectly to avoid any chance of electrical or physical damage to the board or device. Taking all these precautions are critical in preventing any kind of hardware issue, to avoid any kind of heavy burden for the group down the line in Senior Design 2, and we can focus more on the progress of our device.

Our device relies on strict timing, and these are critical for the data we produce to maintain structure and consistency in the prototype. The cornerstone of how the device will work is miscommunication in voltage differences between the numerous sensors interacting or transitory layering of sensors where one action affects the next. If this happens, we have no data and consequently no control over the device until the sensors are working. The sensor arrangement must be divided into levels of implementation. Setting up stages or hierarchy levels in which sensors are configured to work in a building block stage. Once one sensor has met our expectations, we add another level to function at the same level if they are linked, and we test both sensors gradually again. This approach definitely takes a little longer in the beginning, but it will take less time in the end if each layer has been thoroughly tested and is known to work without issue. It will be important that we keep this in consideration.

Our project is heavily reliant on the accuracy and robustness of our sensors. There are two main sensors in our project - audio sensors in the form of microphones, and a visual sensor in the form of a camera module. Our project involves having both of these sensors collect data and combine this data to reduce a new visual output. Accuracy is a very important factor for all of this, and therefore we need high quality audio and visual capture available for our device.

One of the main concerns and potential issues regarding hardware would be the microphone's ability to detect sound at a desirable accuracy. It is very hard to judge whether the microphone will meet our desired amount of accuracy before purchasing and testing it. Thankfully, microphones are relatively cheap in terms of electrical components and our team would probably not be financially affected to an extreme degree if we did have to purchase and test different microphones.

Inherent accuracy is going to be a potential hardware issue for the microphones and for the camera module. Thankfully, these inaccuracies can be determined and accounted for. Once they have been determined, using software, we can compensate for the inaccuracies. Unfortunately, there aren't many ways to improve physical inaccuracy of the components themselves, however, we can make sure to reduce the windows for these inaccuracies to affect the device significantly. Accuracy on the team’s accord is very important. For example, one way to ensure that we have accurate readings of 3D sound localization is to accurately place the microphones in a triangulated manner that is known. There might be some small variations and inaccuracies in the placement of these devices, but we tried to place them as accurately as we can. The same goes for the camera - it must lie right in the middle of the microphone triangulation, and we must know the exact location of the camera relative to the microphones in order to ensure an accurate 3D Silo causation and correlation to a specific pixel location on the captured image.

With the prototype, there might be some hardware issues regarding wire connections. Assuming that we might use the pre-built MSP430FR6968 development boards for practice and potentially the Prototype version of our project, we might have to use basic pin connections with standard wire adapters. However, due to minor inaccuracies and discontinuities of wire connections, there might be some loose wires in which inaccurate voltages might be set.

Furthermore, in regard to the chassis, there might be some issues regarding the completeness of it for the prototype version. For this prototype version, we might settle for a non-3d printed chassis which will just be a simple platform that holds the circuit boards in place along with all the other electrical components. The reason for this is that our design might change up to the point of the prototype, and therefore we might not know the exact dimensions that we would need to house inside the chassis. If we use 3D printers to create the chassis, we might be able to rapidly change the design of it, however this is not guaranteed, and we might still use laser cut wood. Either way, the materials that might be used for the chassis are going to be scarce and it will be in our best interest to create a prototype that uses a Bare Bones incomplete version of the chassis. This prototype version will likely have exposed circuitry and PCB boards. However, this will not represent the final design intended. The final design will have a more user-friendly and protected design; however, this will only come after we know the exact dimensions and requirements of the chassis.

7.2.3 Prototype Constraints

Constraints of the prototype are factors that might not be able to be implemented into the prototype which we would otherwise hope to include in the final product. These prototype constraints are usually factors or features that we wish to include in the final product however due to the nature of the Prototype we are unable to implement it correctly or in a way that is desirable in the final form. Usually, these features that are not included in the Prototype are features that could be considered factors of quality and robustness. Some examples of these features would include a complete chassis, finalized end complete PCB design, accuracy in the software and Hardware domains, etc.

The Prototype will use the bare minimum components and quality to achieve function. Functionality is the primary objective When developing a prototype. The purpose of the prototype is to demonstrate functionality and usability. Although a main feature of our device is user interaction, the most important aspects of the device are functionality and accuracy. Ultimately, the Prototype will aim to represent the final forum in terms of functionality first and foremost.

Another major constraint the prototype faces is timing. The Prototype is not meant to be an exact representation of the final device. Therefore, it will likely use components and other features that are not exactly what the final device will use. For example, the Prototype will likely not use a printed circuit board (PCB) in its design. The reason for this is that the Prototype needs to demonstrate functionality first and it doesn't necessarily need to perfectly represent the physical Hardware of the final device. The final device will use a custom-made printed circuit board. However functionally speaking it should be more or less the same as the MSP430FR6968 boards that several group members already own. Therefore, the Prototype will most likely utilize these boards when testing and demoing the functionality of the device. The boards that the group members already own will provide the functionality and also allow us to avoid the time constraints of creating a prototype before the final device is made. Using these boards, we can skip the process of

Designing and ordering a printed circuit board and focus on developing the functionality and demonstrating it in a demo.

Similarly, the device will likely use pre-made microphones that are ready to use and capable of conducting and being integrated with a prebuilt general-purpose board.

7.3 PCB Schematics

The following schematics were all made on KiCad. These are the overall schematic designs between the battery management, microphones, MSP430 and the Raspberry Pi with all related connections.

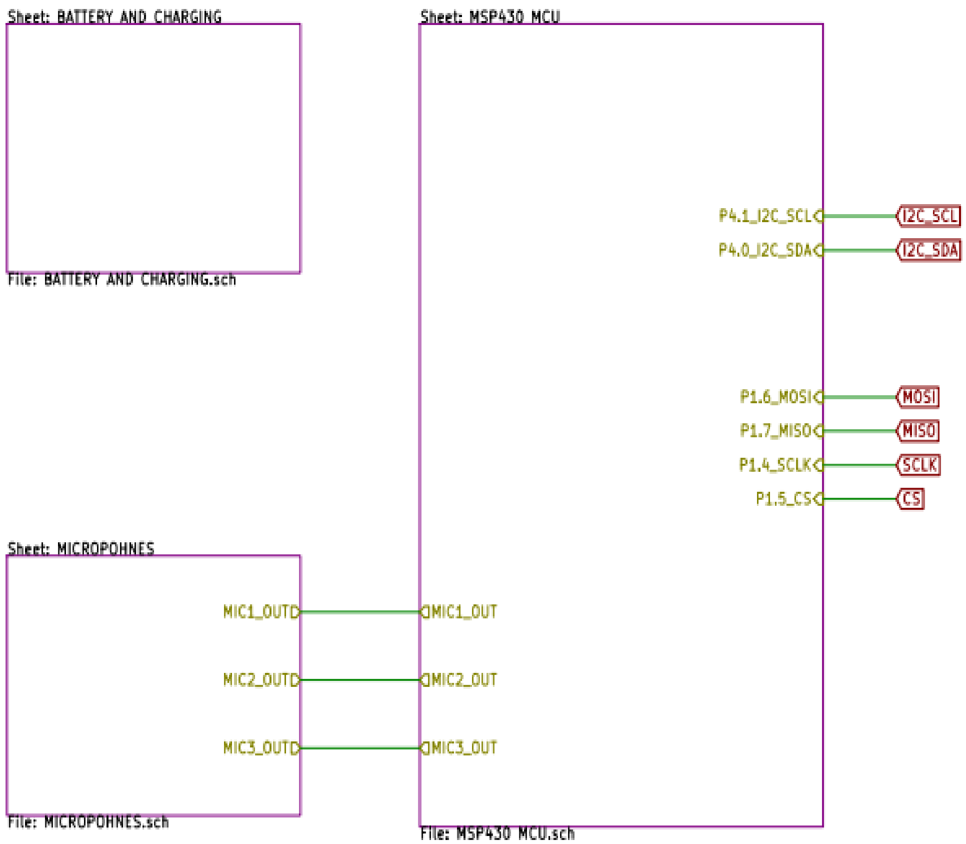


Figure 46: Overall Schematic part 1

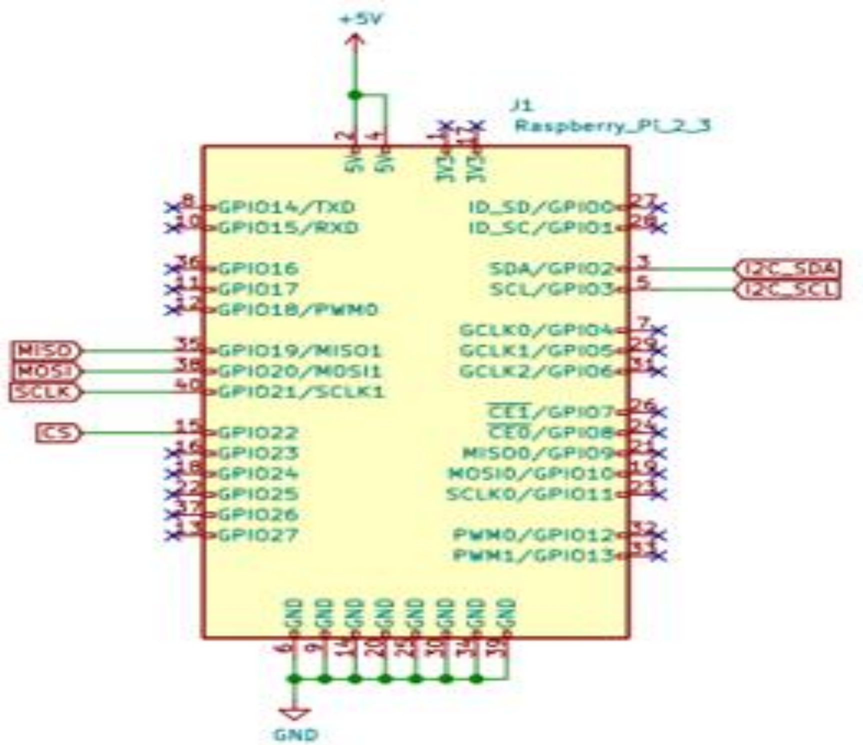


Figure 47:overall schematic part 2

The following schematics will be related to the battery and charging management for the device.

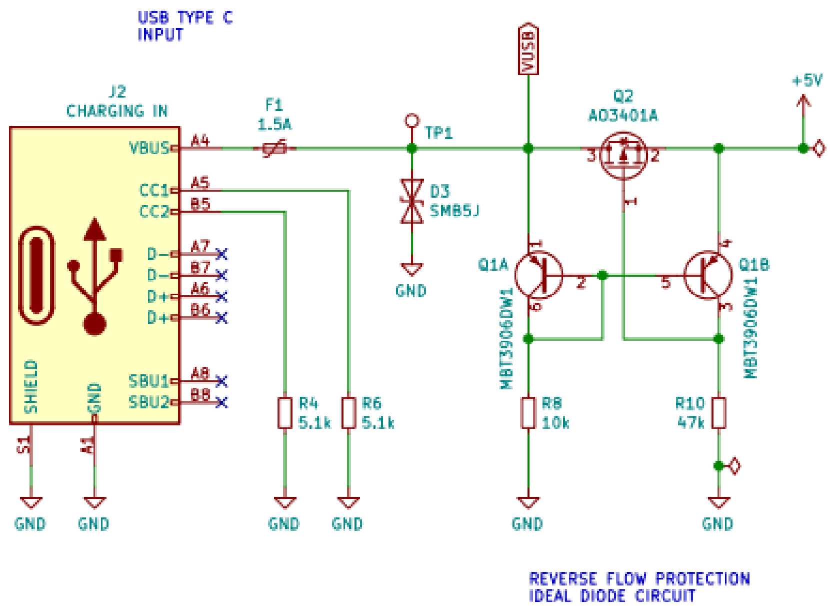


Figure 48: Reverse flow protection ideal diode circuit

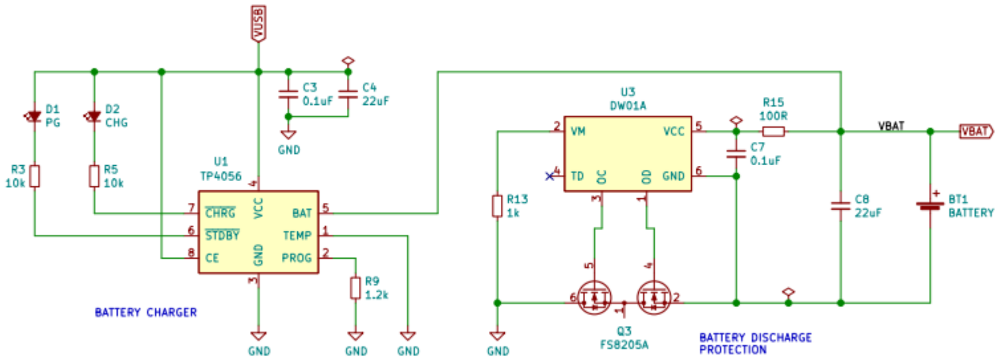


Figure 49: Battery discharge protection and battery charger

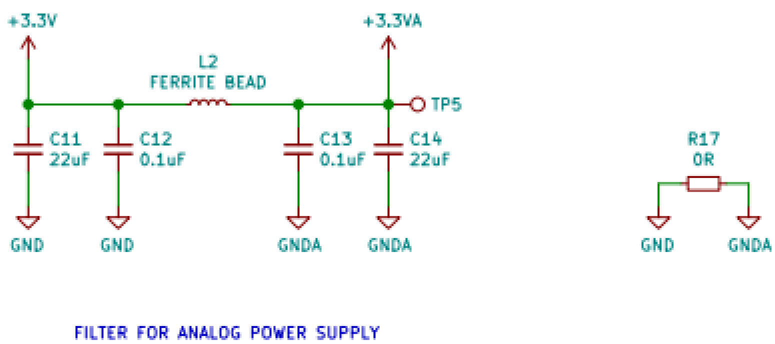


Figure 50: filter for analog power supply

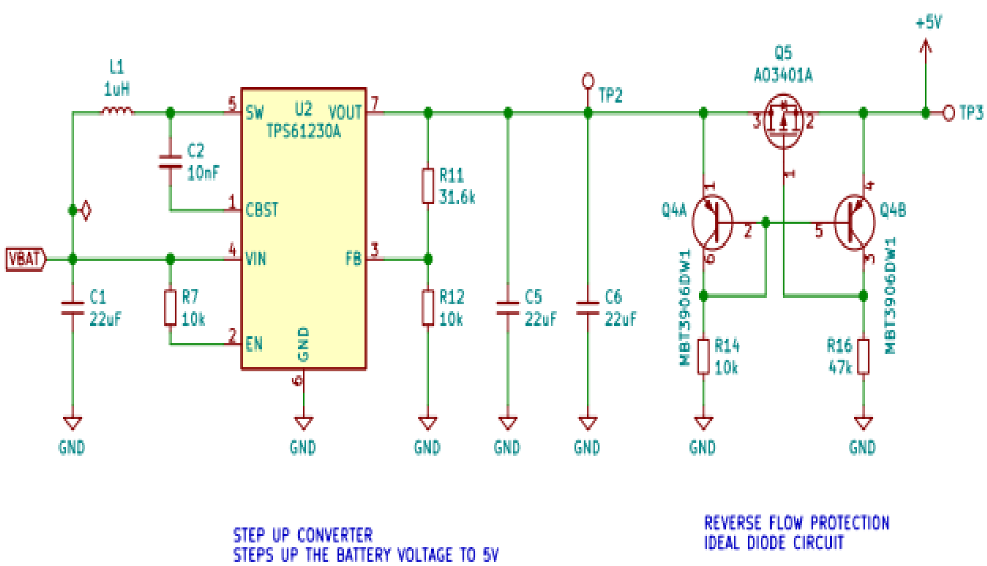


Figure 51: step up converter and reverse flow protection ideal diode circuit

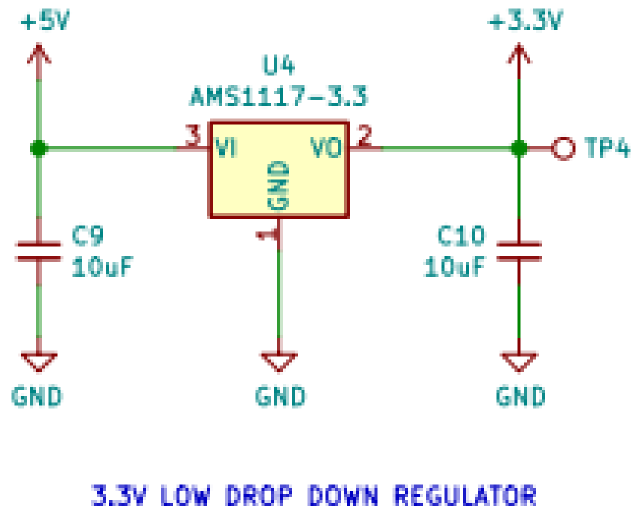


Figure 52: low drop down regulator

The following schematics are related to the MSP430 microcontroller.

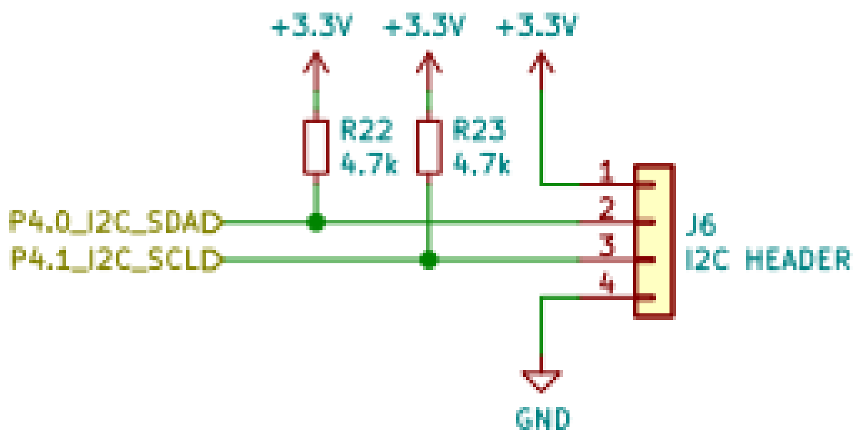


Figure 53: I2C headers

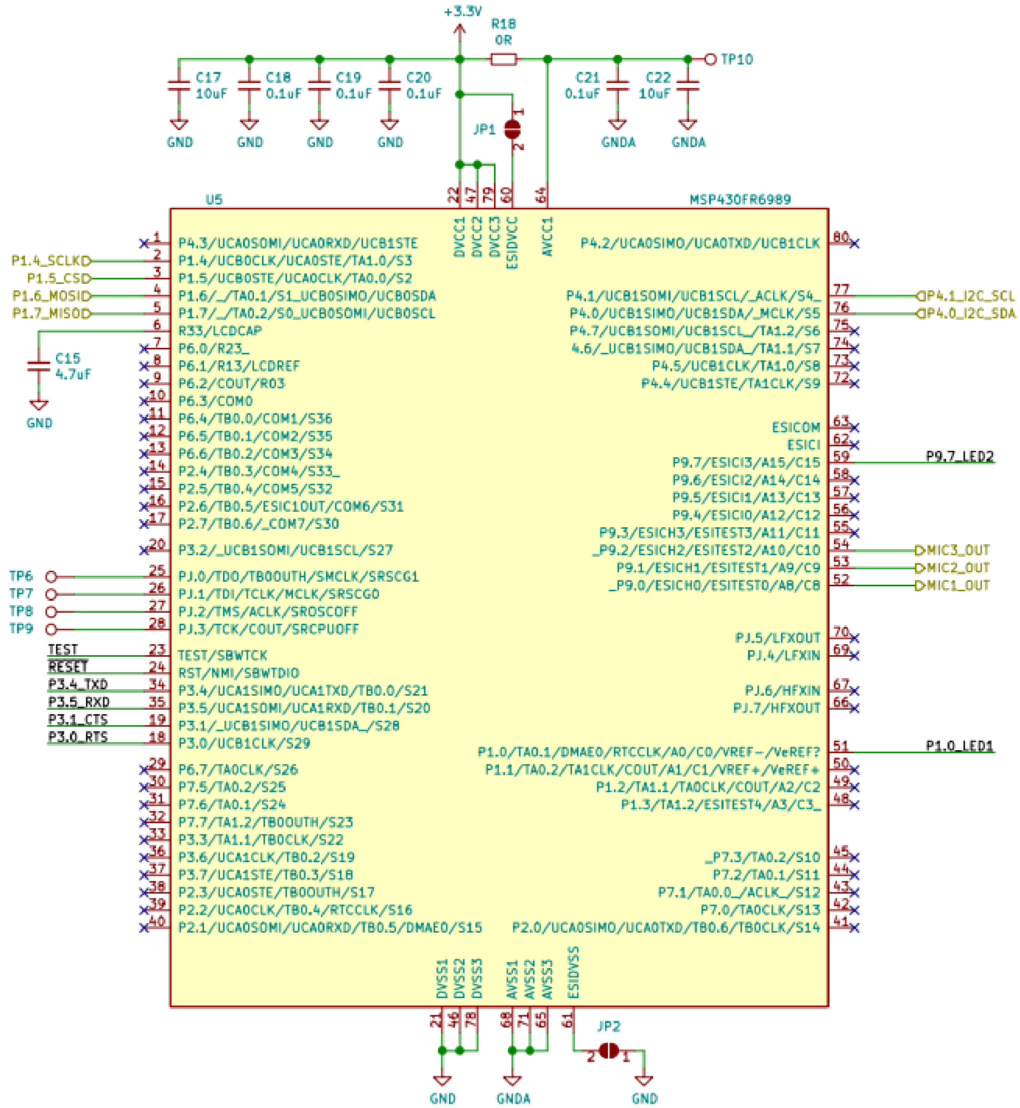


Figure 54: MSP430

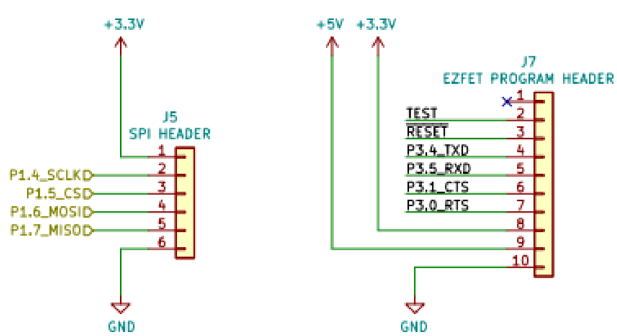


Figure 55: SPI and EZFET Headers

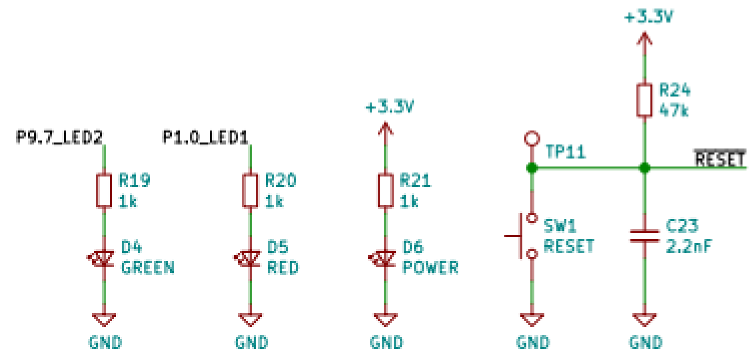


Figure 56: LEDs

The following schematics are related to our microphones. There are three identical circuits that are each connected to pins 52, 53, and 54 for Mic1_out, Mic2_out and Mic3_out respectively.

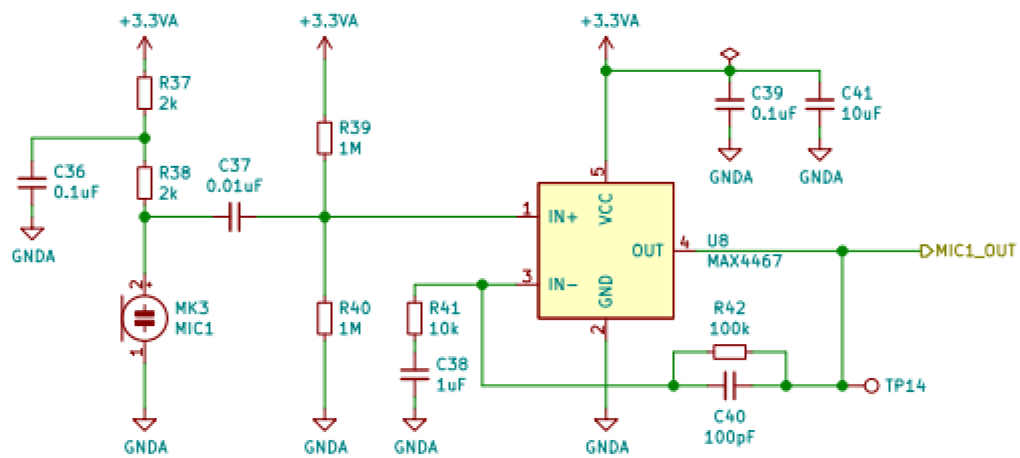


Figure 57:Microphones

8 Prototype and Demonstration

We'll go over the many sorts of testing methodologies we'll use in this part. When developing our Chromesthesia simulation device, this ensured a successful development. The test procedure will be described in full below, as well as the test's expectations. The key forms of testing that we attempted to implement throughout our testing process are as follows: functional testing, usability testing, performance

8.1 Functional Testing

The practice of determining if a piece of software is operating in accordance with predetermined requirements is known as functional testing. Here are just a few of the many steps that can be included in functional testing: the process of determining which functions software is expected to fulfill, test case execution, and comparing output. There are many different kinds of unit testing such as: unit testing, sanity testing, and smoke testing. Unit testing is where developers construct scripts to test if individual components/units of an application meet the requirements. Sanity testing, which verifies every essential functionality of an app works with no issues. Smoke testing, which is typically done before a sanity test, consists of making sure the stability of the software is fine and there are not any abnormalities.

A common misconception is that system testing is the same as function testing, but the difference between the two is that functional testing compares the results to the original design document while system testing compares results against the user/system requirements. In section 7.1.2, we go into more detail for system testing.

8.1.1 Manual Testing

This section will go over and cover all of the manual testing that will need to be done for the development of this device. Manual test cases usually serve as a guide for software testers who follow a step-by-step approach for testing. By clicking links, typing text, and hitting buttons, they can complete the tasks. While checking the output, the functionality of a product can be checked by a tester by comparing the actual results to the projected outcomes to see how effective the software is. Defects and faults in the code will be detected by the tester, so testing by hand will always be required. Manual testing is beneficial since it relies on human judgment and intuition to ensure that the intended results are achieved meanwhile automation occurs when a machine validates some objects but misses what a human would notice. If the program is continually changing, manual testing can be done; however, if the application is static and not changing, automation is preferable. Manual testing allows you to test quickly and easily without having to write a test script. Manual testing is usually less expensive in the near term than purchasing an automated test subscription, which is usually more expensive. Manual testing has a number of drawbacks, including the fact that it takes a long time and is inconsistent because each person sees an application differently and from their own perspective.

For this project, Manual testing is defined as testing that needs to physically be done by people because the process of testing or the features that are being tested are inherently physically limited and cannot be automated in any form. Therefore, any input or variable changes for the testing must be performed, measured, and set by individuals. Examples of this include the physical use of the device - aiming and positioning it in a manner similar to how it's intended to be used.

To demonstrate manual testing, we use the device in the way it's exactly meant to be used, which is to be held and aimed in directions determined by the user towards audio sources and being able to detect frequencies and sounds. It would also be convenient to create some sort of consistent Sound Source. Although our end goal is to create something that can detect sounds in any environment with ambient noises, it might be in our interest to create some sort of sound sources that are reliable and consistent and can act as a constant variable when it comes to testing and demonstrating the manual features of the device and the Prototype.

For example, we could create speaker platforms connected to someone's phone which plays a consistent noise at a specific frequency. This would allow us to be able to aim the device towards and away from The Sound Source with a frequency that is known and we can determine and prove whether or not the device and prototype are accurately detecting and performing the desired functionality based on the input of the sound sources.

8.1.2 System Testing

System testing is a type of test that verifies a software product's completeness and integration. The goal is to test end-to-end system specs. Unit testing and integration testing are required for all applications. When integration testing begins, the application is complete. When a system reaches the system testing phase, it indicates that the application is nearly finished. Despite the fact that minor defects may still exist, system testing indicates that the application is nearly complete. In this scenario, we are ensuring that the product is exactly what the user requested and intends to use.

Black box testing is one sort of testing that we used in development. This implies that offered an input to the program, which is referred to as a "black box." We must ensure that the data input was correct and logical. We must also ensure that the data is properly assimilated and that the necessary results are produced. System testing relies heavily on black box testing. Functional testing is one of the stages of system testing, and further information can be found above. However, performance testing and confirming that the application operates smoothly is another important aspect of system testing.

Audio Processing Prototyping:
Prototyping the audio processing unit and ensuring the capabilities of audio and frequency detection was one of the first and primary stress points that we needed to test and ensure that we had a desirable reading. To test this, a test was set up in a similar fashion to the black box nature that was described above. This demo feature was to essentially make a spectrum analyzer that can display and read out the audio inputs of the microphone using the MCU we decided on.

The audio detection and processing of the audio inputs from the microphone is one of the most important baseline tests that can be accomplished with this stage of development. The reason is because all of the technical complexity of the system and the device as a whole depends strongly on the audio processing aspects of this device. The audio processing steps involve everything from the level of the audio inputs from the microphone itself to the utilization of that collected data in the software residing in the MCU. The first and most basic test was to ensure that the microphones are able to detect varying frequencies of sound, that they are able to convey this data to the MCU via pin connections, and that the MCU is able to use this data in real time and output values based on those inputs.

To ensure that these three levels of the audio processing work, we would need to conduct a black box test to determine if a system that is set up for this specific function using the exact hardware we intend to use would indeed be able to perform these processes from end-to-end. To do this, we started by creating a simple demo setup using basic pieces of the hardware we intended to use. This includes an MSP430 launchpad board and one of the DAOKI microphones. Using a breadboard and wires to connect the pins between the microphone and the MSP430FR6968 developmental board.

The first objective for this experiment was to do pin analysis. Referencing the MSP430FR6968 board's documentation, we were able to identify the proper pins that can intake the output of the microphones. Based on the MSP430FR6968 documentation, we identified the proper pins and were able to establish pin connections between the proper pins using the breadboard and wires. The DAOKI microphones and the MSP430FR6968 developmental board were connected to the power supply – a 9V battery connection, and the pins were connected respectively. The Ground and power pins of the MSP430 and the microphone were connected to the power lines of the battery, and the output pin of the microphone was connected to the pin of the MSP430FR6968 board. However, between this connection, we connected a LED to output some form of visual feedback to confirm there was indeed voltage coming out of the microphone through its output pin. This served as a intermittent black box, allowing us to determine in the case of no signal being read on the MSP430 and its software, whether this is an error of the MSP430 or the software we have developed for it, or whether there was some issue with the microphone reading audio input signals. This specific step is very similar to the setup we used to test the raw functionality of the microphone earlier in this document. This acts as a quick and easy test for functionality on the hardware level.

The output pin of the microphone is connected to the P6.0/R23 pin. Using Code Composer studio, we developed code in C to run on the MSP430. This code would take the voltage values from the output pin of the microphone. This code would take the measured voltage value from the microphone and quantize it.

Based on a frequency look-up table, which we based on the frequency to hue generator that we described in the above sections.

Frequency	**Output Value**
1 kHz	19.913
2 kHz	31.913
3 kHz	38.913
4 kHz	43.913
5 kHz	47.776
6 kHz	50.933
7 kHz	53.601
8 kHz	55.913
9 kHz	57.952
10 kHz	59.776

Table 31: Frequency to Float Conversion Table

This uses the same formulas from the hue calculation formula, however, instead of displaying a hue, this code will output the appropriate value that will be used by the hue calculator algorithm. The most prevalent frequency detected will get its hue value detected and this numerical float value will be output onto the command line. This will be output as a continuous stream of numbers being output. With this setup, we can carefully watch the output as the microphone is detecting inputs real time.

The hardware setup of this Blackbox demo testing is displayed below in Figure XX:

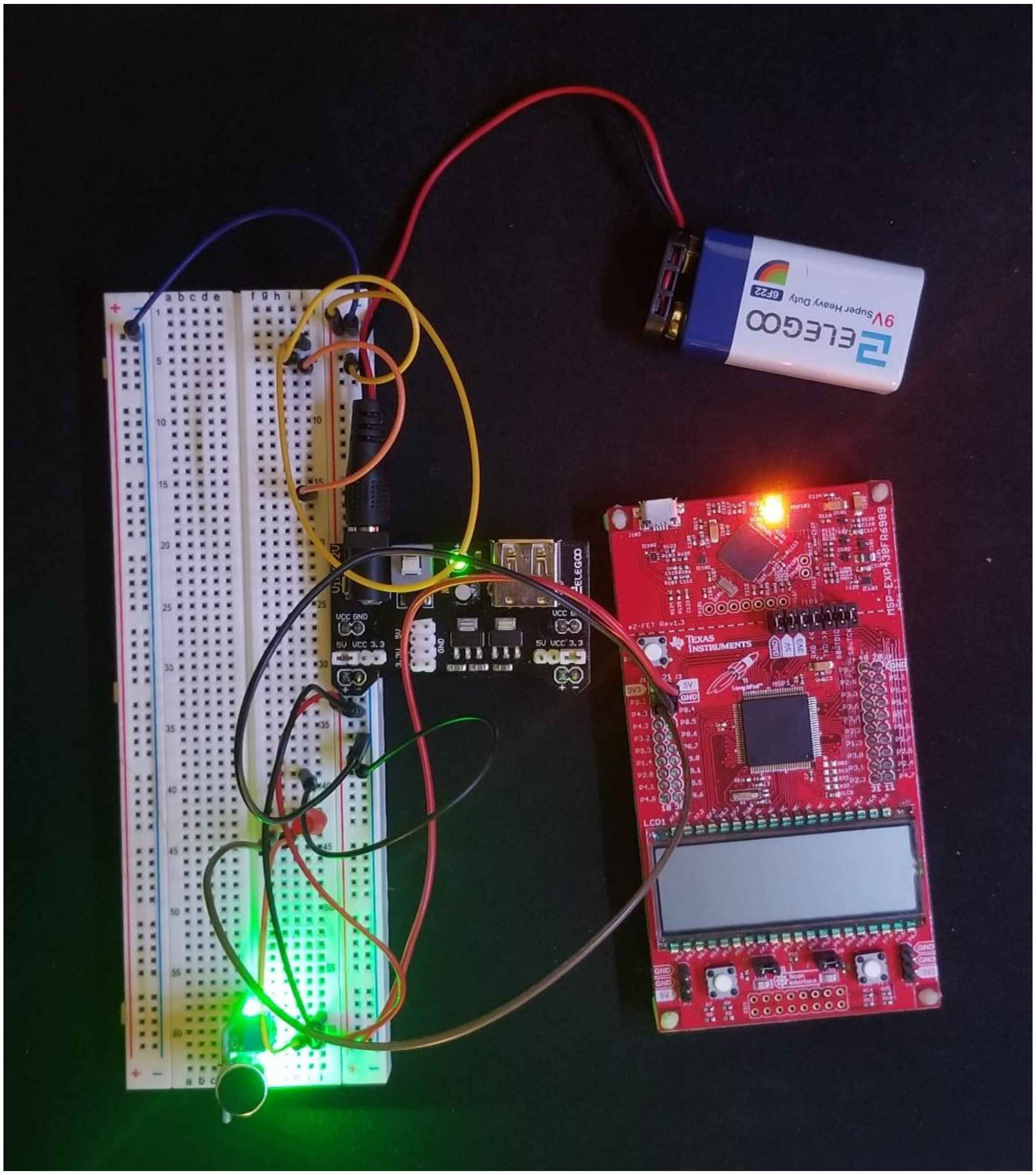
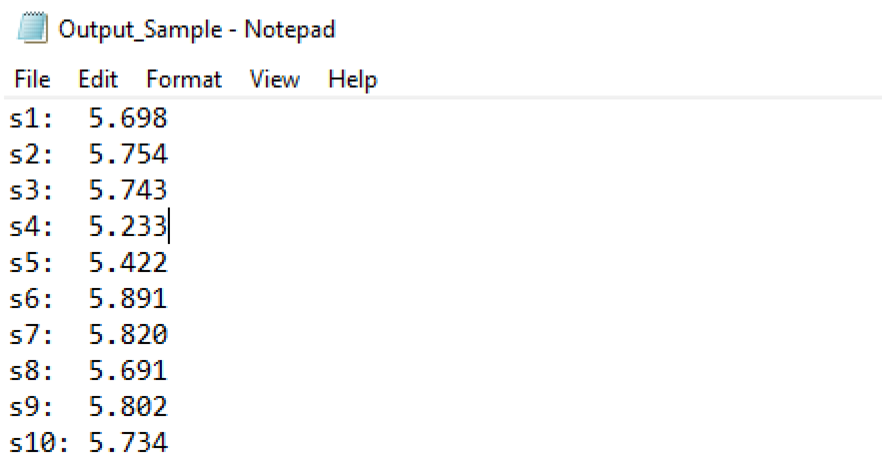


Figure 58: Hardware Setup of Audio Detection Demo

The output data collected from this was checked and measured in real time, in direct response to audio input signals collected from the microphone. The data was observed by the operator and the data was compared to a predetermined target value of the frequency measurement. To ensure this was accurate, a sound was played into the microphone of a predetermined frequency. There are many resources online that can generate frequency noises that are open and available to the public. The one we used was called “online Tone Generator”, which allowed us to generate a frequency pitch based on a input value. We could range the frequency to 20,154 Hz to 1 Hz. For the sake of the test, we would select a specific frequency, i.e. 440 Hz and would play the audio output through speakers placed nearby the microphone.

Knowing the output frequency, we could calculate by hand what the value should be, and compare this to the value output by the MSP430. In this situation of 440 Hz, we expected to observe an output value of 5.7, which was observed within a reasonable range.

The following output was recorded at 440 Hz audio input:



Output_Sample - Notepad

File Edit Format View Help

```
s1:  5.698
s2:  5.754
s3:  5.743
s4:  5.233
s5:  5.422
s6:  5.891
s7:  5.820
s8:  5.691
s9:  5.802
s10: 5.734
```

Figure 59: Output Sample of Audio Inputs at 440 Hz

Measuring the sound interval over the course of 10 seconds resulted in a relatively stable readout, detecting each audio input measurement over the interval of 1 second. The measured outputs were written onto a text file and the results were viewed with notepad. The expected output at 440 Hz was 5.7, therefore, the frequency readout was quite consistent. There were other notable details about the test which were not necessarily measured due to the fact that these details were not the objective of the test, however, they will be described and noted for future reference.

The main one was the fact that the speaker playing the 440 Hz frequency was placed less than one foot away from the microphone sensor. The speaker was an Edifier R1280T, which has a power reading of 42 Watts RMS. Unfortunately, we did not have access to a decibel reader, therefore we were unable to properly determine the exact volume of the speaker, however, we can assume that it was around the middle range of the speaker’s maximum volume range. This is an important detail to note, because the sensitivity of the microphones would be the next step in testing. We have determined that we gain a better understanding of the microphones in this regard and further testing of the same test, but with varying volume and distance from the microphones will be conducted. Below is an image of the testing setup.

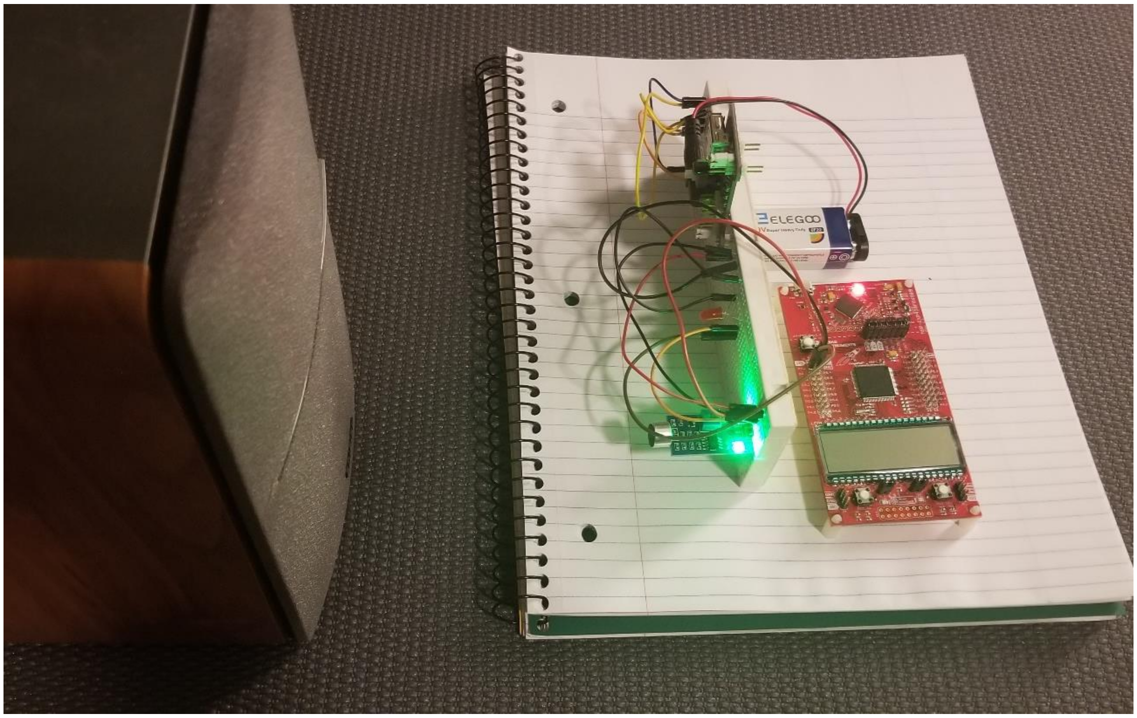


Figure 60:Testing of Blackbox Microphone Setup

AS can be observed in the image of the testing setup, the speaker is less than 1 foot away from the microphone. The purpose of this test was to ensure the microphone could intake audio and reflect that it is reading the correct frequency based on a known frequency input. Furthermore, the factor of additional noise is of concern, and the noise filtering and 3D sound localization process will be refined and explored in future tests. Much more research of how sound works in the electronics domain is necessary; the understanding of how sound works and what influences measurements is very important to further develop this model.

8.1.3 Automated Testing

Automated testing is a technique in which a software tool generates a script that is used to test various aspects of an application. The validation is determined by the automation script using a previous run that has already been created, and it only validates that run. Automation has the advantage of being quick because it is executed by a computer and there is a test suite that contains all of the scripts and runs them all at once. When compared to a human running it one by one, the overall time for testing is cut in half. Another benefit of automating is that you can write a single script that validates every component of the application because the scripts are predetermined. Testing is more precise with automation, which decreases the chance of human error. To put it another way, a testing tool or computer will not become bored if it makes a mistake.

Automation does have a few drawbacks, one of which is the high cost of the tools. Thousands of dollars can be spent on a single license. Automation only works for software that is in a stable state. Continuously changing software is not a suitable choice for automation. Automation has some drawbacks, such as the necessity to update scripts when the program changes. Because the tester must comprehend the automation technology, it necessitates a higher level of ability than manual testing. The tester will always be required to be fluent in the tool's language. Manual testing will never completely replace automated testing. It is best for the tester to employ both automated and manual testing to conduct a comprehensive test. Since the computer is not doing all the mechanics, this demonstrates efficiency and flexibility while also improving quality. The computer's components are being verified by a user. As a result, the goal is to minimize risk while facilitating software quality improvement. Many others, however, question if automation is even cost-effective. If you can afford to automate, it is quite beneficial. You can perform regression testing after any given push. Even if it did not do it right away, if any issues are discovered, they will be fixed right away, as opposed to a human who takes hours to test and report their findings.

There were no guaranteed opportunities for automated testing in this system, however, as mentioned in section 2.4.2 - Advanced Features, there were discussions of potential machine learning to be used. Neural networks and machine learning have become a very hot topic in the modern field of computer technology. Computer vision is extremely useful, and it can be used to recognize patterns such as facial features, alphanumeric characters, etc. There was discussion of a potential advanced feature which would allow us to use machine learning to identify common sound sources such as faces on an image (i.e., each individual frame captured by the camera) and to be able to use this pattern recognition ability to draw around the sound source or highlight certain things that are recognized by the pattern recognition and then add visual effects to them.

However, as it is, there are currently no official plans to implement this into our device. However, there is still potential depending on how quickly the project progresses along during the next leg. This will be explored and considered, especially since several group members have/are taking computer vision courses. The knowledge gained in those courses will be easily applied to this feature, but to reiterate the point, there is uncertainty whether we would be able to implement such a feature since the more fundamental features are more imperative to our project’s functionality.

However, if we do implement it, we would use machine learning and automated testing to train the neural network to become more accurate. Training a neural network is an integral step in implementing machine learning, and would be extremely important to getting a better, and more accurate neural network.

On the topic of training the neural network for recognizing faces - a common source of sound created by people - this would be achieved by using a multi-layered neural network, which uses a sigmoid function for the output. The sigmoid function is displayed in Figure 43 below:

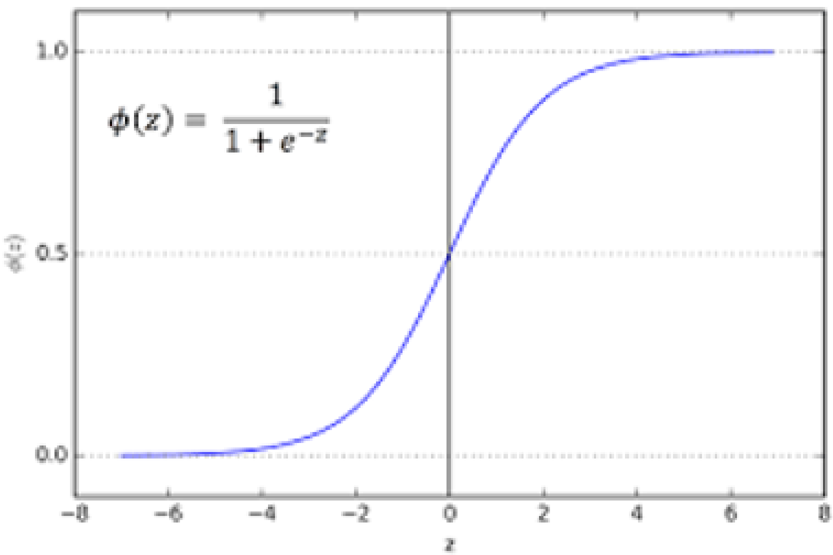


Figure 61: Sigmoid Function

The purpose of the sigmoid function is to center in onto the value of 0.5, where the closer the output gets to this, the more accurate the neural network is. This takes many epochs, or generations of the neural network being trained to achieve a more accurate measurement.

Additionally, we would need to utilize some sort of face database for the neural network to reference and use to train. There are countless databases online for free, which we investigated and determined which to use. But again, this has not been seriously explored yet since the majority of effort has been placed on the basic features of our chromesthesia simulation device.

This is the extent of the automated testing. The only situation that automated testing would be applicable would be for neural network training. However, since this is a highly Advanced feature that might not be implemented Depending on time and budget constraints, there is no guarantee that this will happen.

For testing the other functions of the device, there are still opportunities to use automated testing. The device will be functioning in real time constantly collecting data from audio and visual inputs to create one visual output. One way that testing could be used to evaluate the effectiveness of this process is by creating test case scripts that will feed inputs to the device and its software continuously and at a rate faster than any ordinary person could input.

While automated testing might have a narrow scope, it is important to realize that with a sophisticated system in place, throwing various inputs is a possibility, but more time and a sufficient budget will have to be considerations. This might involve additional research that will not be within scope of the resources we are using.

Some data that we might automate inputting would be frequency inputs correlating to coordinates on the xy-plane of the image frame captured from the camera. This would allow us to automate the testing for this exact feature and potentially create a script that generates a vast array of randomized input data that could potentially test our system's ability to process inputs that we might have not thought of or anticipated. This way once we run into a bug or an error, we might see an input issue that we might have never expected to encounter but thanks to this automated testing we would be able to identify it and apply some sort of changes to handle this specific test case. Test cases will be very important for testing this feature, especially considering how frequently and rigorously our system will be expected to perform this specific function.

9 Administrative Content

The administrative section will aid in the development of the chromesthesia simulation equipment as well as budgeting and budgetary planning.

9.1 Division of Labor

This project's workload was decided upon at the start of the semester. This team comprises four members, two of whom are electrical engineers and two of whom are computer engineers. This worked out well because this project is divided into two key components, one of which is hardware and the other of which is software. The task will be distributed as shown in Table 28 below.

Member	Leading	Co-Leading/Assisting
Angel Garcia	• PCB • Microcontroller	• Product Hunting and Identification
Brooke Roeder	• Pitch-color algorithm method • Image processing	• Testing and review processes
Nicholas Alban	• Audio Processing Software • Constructing Platform	• PCB
Wesley Ellery	• 3D Sound Localization • Microcontroller	• Integration Testing and Review Processes

Table 32: Division of Labor

This distribution of work is ideal because we assigned work based on interest and strengths. Our group comes from two majors: Computer Engineering and electrical engineering.

There was a lot of overlap between us, however, there are many subcategories of both of these fields where they overlap and where they don't. we decided to assign work and responsibilities not based on a traditional leadership structure. Normally, you would see a leadership structured similar to that of a business or military hierarchy. however, with our design we are able to dynamically shift and help people. everyone is responsible for their own specific roles, and then there are people who are specifically designated to support these people who are leading these roles. Leading roles were assigned based on skills prior to joining the senior Design Group, and interest. for example, if someone was interested in image processing, they could opt in to contribute to it. But also, if someone had experience in image processing, they could also join it too.

9.2 Estimated Finances and Budgeting

This section will be about the bills of materials for our chromesthesia simulation device. These components contain all of our key selections, but they are subject to change during Senior Design 2.

The PCB will be designed and ordered during Senior Design 2 and is not included in this bill of materials. The materials, pricing, quantity, and store from which we have purchased the items will all be listed in the table below. Originally, we believed the project was not going to go over $400, essentially it being split $100 for each member of the group. At the moment it seems that fortunately we have spare components that will help reduce the cost of the overall project and we definitely were able to meet these criteria. Currently, we are in a period of time after the initial spike of the Covid pandemic.

Although some parts of the world are back to normal with the exception of some new policies, there are many kinds of shortages happening around the world: Chip shortages, wood shortages, even chicken shortages. All of this has caused prices for many different items to soar up so some of the components we need could be more expensive than they were two years ago. We believe that the price listed below will be our estimated cost to complete our project. It is our projected cost because we are still waiting on some other unknowns would arise once we implemented our concept. One potential unknown could be one of our pieces failing and needing to be replaced.

Furthermore, currently we are under budget, so in order to prevent any future expenses checking all of our components and materials and testing them to verify quality as previously described in Chapter 5. The benefit of being under budget, allows us the freedom to purchase additional components to improve our simulation device if there is enough time to do so. Another benefit of being under budget allows us to potentially purchase high-quality parts which could result in a higher quality finished product. At the moment, we have a budget of $400 and the costs given below are estimates based on online research and are subject to change at any time.

Item	Price
Adafruit I2S MEMS Microphones	$6.95
Arducam Mini Module Camera Shield with OV2640 2 Megapixels Lens	$25.99
PCB	$10-$50
Raspberry Pi 4	$100
Display	$30
DAOKI High Sensitivity Sound Microphone	$1.14 each (5)
DB Unlimited Unidirectional Electret Microphones	$1.840 (6)

Table 33: Itemized List of Materials

The materials listed above, and totals are subject to change. A finalized sheet will be implemented during Senior Design 2.

9.3 Suppliers

It was critical for us to discover the correct vendors when it came to acquiring our components for our project design. When researching a supplier, we looked at a few things, one of which was their trustworthiness as a legitimate site and business to guarantee we weren't buying a fake or inaccurate product. We wanted to make sure the supplier we ordered from wasn't a scam website or a firm with a negative reputation because the internet is so large and accessible to essentially anyone with a computer. As a result, we chose well-known websites such as Amazon, DigiKey, and Mouser to ensure this.

Another factor we evaluated while choosing our vendors was their ability to deliver on schedule. We must factor in the time it takes to place an order, process it, ship the item, and receive it. Amazon was chosen as the best alternative for each component that needed to arrive as soon as possible. If we have an Amazon Prime membership, we may receive most things within two business days, and shipping is "free" because it is included in the membership price. We kept this in mind in cases where Amazon sold the component for the same or very similar pricing. Unfortunately, there was a downside to Amazon, which was the fact that they did not have what we needed in stock, or if it was, there would have been only one item in stock, very limited. To add on to that, sometimes the item would be available through a third party on Amazon, but at a higher price. We ended up ordering a lot of our components through Mouser, because this supplier had everything we needed in stock, and the shipping was pretty quick. It did not compare to Amazon's delivery times, and shipping was not paid for, but it was enough for us since we had time on our side and ordered most of our components early. One benefit from ordering from Mouser was that they did have incentives for ordering multiple things on their site to qualify for the free shipping, which we fortunately were able to do. Along with that, we ordered multiple components to make sure we had backups in case things were broken, fried, etc. We ended up sticking with Mouser mainly, and Amazon whenever we needed something as soon as possible and if it was available. None of us needed to order our launchpads from Texas Instruments because we had purchased our own in a previous semester.

With several external factors including the widespread chip shortage, we must understand there is a possibility that shipment of our hardware and components could take several months. This is why we chose several different types of hardware, since we need to take this as a factor toward our development.

9.4 Project Tools

In order to successfully complete a project of this size, our group needs to be efficient, productive, and have good communication amongst one another. Fortunately, we are in the twenty first century and the internet has provided us with many useful tools, along with the University of Central Florida. The sections below will go over the tools used among the team, how we communicated and what we did to plan meetings, and how we discussed our schedules to make sure we did not fall behind.

Team failure could be cause by a lack of communication. Our team has worked hard to improve our communication skills over the design process of our project. During our meetings we would converse about specs, what needed to be done, what components needed to be ordered, and how we would test all of the parts ordered. As group members, and our graduation on the line, we all initially agreed to be completely transparent with each other about our individual tasks progress throughout the course of the semester. Listed below are some mediums we used in order to communicate with each other.

9.4.1 Discord

Discord is a free platform dedicated to voice, video, and text chats. It is used by millions of people around the world, professionally and recreationally. For example, some groups can use discord for art projects, esports, streaming, business meetings, and even simply homework. The University of Central Florida even utilizes Discord, like the UCF CS, ECE, and IT server. A lot of students create a GroupMe when a semester begins, but we quickly learned as a group that we all would just have those chats muted. Instead, we opted to create our own server where we could create text channels dedicated to research, part components, and even have an announcement channel to make sure everyone gets vital updates. Whenever someone needed help or had a question, we could respond quickly to each other or even go into one of the voice channels and communicate through there. Having this kind of power was extremely beneficial to the organization and improvement of communication in the group.

Discord also has the features of having bots added into our servers. What we did was we add a bot that would keep track of our schedules and make an announcement whenever our planned day was approaching. If someone needed to make a change or couldn't make it, we would simply reference the bot and message each other whether or not we would change the meeting time.

9.4.2 Email

Electronic mail (e-mail) is a method of sending and receiving messages using computers and other electronic devices. It is the most professional method of communicating. This served as a forum for us to communicate with our lecturers. We utilized it to ask questions and request one-on-one sessions with the professor whenever we ran into problems and needed some direction to get back on track. If required, we contemplated sending files and links via email, but eventually decided against it due to some emails having a restriction on file sizes. Email is an excellent tool for work and general questions, but it is considered incredibly slow as a regular back-and-forth communication mechanism. We did, however, exchange university email addresses so that we would have a means of communication if the need arose.

9.4.3 In Person Meetings

Our team initially agreed to meeting up two times a week, every Tuesday and Thursday in the evening, mainly due to the fact that our work schedules were not aligning well. The constant meetings made it so we connected with each other more often and improve our team dynamic, along with constant progress on the designing process of our project. Week by week, we would discuss and keep moving forward. Discord was our biggest tool because we did not need to be at our computers to communicate. There is an app for it so a lot of the communication were done from our phones, and the meetings would be when everyone was home and able to access their PCs.

Throughout the semester, we had to meet with our professors, and during those meetings we would utilize Zoom to communicate. After each meeting we would either go into the voice channel of our server, or discuss through the one of the text channels to discuss how we thought the meetings went and what we needed to do to progress. There were plenty of moments where we all were in the voice channel, cranking things out and making sure everything was in good shape. We are aware that we are very fortunate as a group because everyone holds themselves accountable and make sure that each individual's attendance for our team meetings stayed consistent.

9.5 File Preservation/Collaboration

In today's world, technology always has the possibility to fail. There must be precautions made in order to make sure we do not get caught off guard. This project is extremely important to us, as it is our last obstacle towards graduating, and we cannot take any unnecessary risks. During the research phase, we made a live word document through SharePoint, which was provided to us by the University of Central Florida, and that was shared between our four members. We would all make edits on this and keep track of the changes. To add to this, we also backed this file up in Google Drive, and another in Drop Box just to be completely safe.

9.6 Facilities/Assembly

A very common meeting spot on campus at UCF is the Atrium that is located in Engineering I. Due to COVID restrictions, in the beginning of the semester it was pretty empty, but later on it grew into how it was before the pandemic, and was filled with students discussing projects, having study sessions, or simply just socializing. Eventually it became too noisy for our group, so we decided to think of other potential areas to meet. Two of our team members are Electrical Engineers, so we were able to access some of the labs provided on campus, and more often than not they were empty at the times that we would go as a group. Additionally, we would also meet at each other's houses, until we found which one was the most comfortable for all of us. We did this because we needed a reliable spot where we could have outlets for all of our devices, and we did not want to constantly transport all of our components, especially when the building of our device begins. We did not want to risk any of our precious work breaking.

A lab that we made use of once the testing of our components begins is the Senior Design Lab. It is has many resources that we can utilize for our prototyping. The lab has components like resistors, capacitors along with tools like multimeters. Inside of the labs are computers that students can use in case they need to access the Internet and figure things out, and they are also loaded with plenty of software that can be utilized. The lab has completely open hours so we can gather with our group members and work on the prototype whenever it is convenient for all of us.

9.7 Information of Group Members

The section below will describe brief descriptions of our group members, including what their majors are, their goals and aspirations, and their skills that they brought to the table when working on the chromesthesia simulation device.

9.7.1 Angel Garcia

Angel is a Computer Engineering student. He currently works as a Field Technician for AdventHealth Hospitals. He hopes to work his way up the ladder and learn more about hospital networking and logistics. During his time in one of the Embedded Systems classes, he realized he enjoyed working with these kinds of devices so he was put in charge of the construction of the PCB. Angel possesses a diverse skill set and is proficient in a variety of programming languages for both software and hardware development. He is also knowledgeable about embedded systems and assists in ensuring that the electrical and software development aspects of the project are in sync and can produce a seamless result.

9.7.2 Nicholas Alban

Nicholas is an Electrical Engineering student. He is currently interning at General Dynamic Mission systems, supporting live service and development of government contracted military simulation and training software and frameworks. He's taken a few of classes related to microcontrollers and circuit boards, and many courses related to programming, machine learning, and computer vision. These skills are anticipated to be useful and strongly applicable when it comes to applying algorithms and complex function to the board itself.

9.7.3 Wesley Ellery

Wesley is a Computer Engineering student. He currently interns as a Systems Engineer for Collins Aerospace, supporting requirements development and testing efforts for the Comac C919 airplane. He has also worked on various projects for the Boeing 777X program, as well as innovation projects regarding aircraft fuel savings. Wesley hopes to gain more experience in the field of Systems Engineering, and involve himself more with field testing, as well as flight testing.

9.7.4 Brooke Roeder

Brooke Roeder is an Electrical Engineering student. She currently works as an intern at Lockheed Martin in systems engineering, supporting the testing and implementation of sensor technology. In addition to supporting the conductions of tests, she has also automated and developed scripts used for image processing. She is currently transferring to the electrical engineering department at Lockheed Martin and will begin to work in hardware design.

9.8 Milestone Discussion

We'll define major and secondary objectives in this section. Each week, it is hoped that a fresh contribution to the project will be completed on time. Basic technology research and the formulation of goals and priorities divided among the individuals assigned would be the primary milestones. This time period should, on average, last 3-4 weeks. At this step, teams are created, and general brainstorming for concept selection begins.

Once this is accomplished, the project may move forward in a more detailed manner, since it will take about a week to assign all participants to sub-roles and determine which criteria need to be explored after the idea has been chosen. Following the assignment of duties, the 5th-7th week (3 weeks) should be devoted entirely to research, covering all areas of the project from parts to budget and everything in between. As all members make significant headway filling out data to produce the paper, this midway point will focus mostly on detail and analysis. By the halfway point, the 60-page goal should be nearly complete, and we hope that further inspection by our professors will confirm that the criteria has been met. We may then spend the final 3-4 weeks finalizing numbers and perfecting the additional details, features, and possibly non-essential functions we would like to include in our document to make the overall product that much more appealing on paper. The overall goal by the conclusion of the semester, after a 12-week research process, would be to have a 100-page document written that thoroughly shows our concept in vivid detail while staying within budget. The most important quality we hope to achieve by this deadline is to emphasize the feasibility and practicality of the design and its components in order to make the concept tangible to the average consumer while also assisting the team in starting the initial prototype development, which we hope to begin by early to mid-January. The prototype construction may assume full priority once the document submission is accepted, and the project paperwork is completed.

Design building should take place within the first two weeks. The focus should be figuring out how we might simulate and lay out the basic concept's skeleton work. The goal is to achieve practicality and usefulness, therefore getting the product to perform in the simplest and most efficient way possibly comes first. Following the completion of the skeleton design, part development and ordering will take another 2-3 weeks. The best method to accomplish this is to divide the prototype into portions that will be assembled in stages in our multistage design and have extra parts in the case that we need them. When we are halfway through our development, at least one major stage of the prototype should be functioning to our team’s basic standards of operation. Within 3-4 weeks, it should be expected to have that done and begin to work on secondary stages of the prototype. With only a few weeks until final submission, the product should be fully created and perform basic duties. On a basic level, the equipment is supposed to work and do what is expected.

9.9 Conclusion

Our project aims to create a wearable interactable device that simulates the experience of what it is like to live with Chromesthesia, which is a rare neurological phenomenon in which sound frequencies correlate to unintentional perception of colors. Our device accomplishes this by capturing imagery using a camera and audio inputs using a microphone array. It then processes and combines these signals to produce a singular output which is the same image that was captured by the camera, but with additional visual effects that are influenced and determined by the audio inputs. The user is effectively able to point and aim at the device in any direction they want to. Depending on the direction they are aiming it, the device will capture images and audio from that direction and the output will be displayed on a small screen on the same device. This will effectively simulate the sense of what it's like to look around in a world well experiencing a simulated example of chromesthesia.

The device will incorporate several microphones, a camera, a display, and a complicated circuit board with multiple microcontroller units. This specific microcontroller that will be used is the MSP430, and this device will utilize two of them. One of the microcontrollers will handle audio processing, and the other will handle image processing. Both my controllers will perform demanding computational algorithms and functions to process their respective inputs and produce and output. their outputs will be combined to create the singular output to the display screen.

As far as functionality goes, this device aims to create a modified real-time image that is displayed on the screen. The modifications of this image will be determined by the audio inputs and the functions and algorithms that these audio inputs will undergo to determine the location and the corresponding frequency. basic features of this device will include the ability to aim it and see exactly what the device is aiming at and see some sort of altered image influenced by the audio inputs. Advanced features include modifications of this same recipe. ideas such as machine learning applications, localized visual effects, and other advanced effects are on the table for application oh, but it depends on time frame and feasibility.

Ultimately, this device will provide its users with a fun experience and an educational outlook on what it is like to experience this unique natural phenomenon that only a small population of people get to experience.

10 References

1. ISO. (2021, 10 14). *ISO/IEC 14882:2017*. Retrieved from iso.org: https://www.iso.org/standard/68564.html

2. "ECMA-334." *Ecma International*, 27 Nov. 2020, https://www.ecma-international.org/publications-and-standards/standards/ecma-334/.
3. "Getting Started with STM32 - I2C Example." *DigiKey Electronics België*, https://www.digikey.be/en/maker/projects/getting-started-with-stm32-i2c-example/ba8c2bfef2024654b5dd10012425fa23.
4. Goss, Clint. "The Color of Sound." *To*, https://www.flutopedia.com/sound_color.htm.
5. "How I2C Works (I2C Explained Simply)." *The Geek Pub*, 19 Aug. 2021, https://www.thegeekpub.com/18351/how-i2c-works-i2c-explained-simply/.
6. "ISO/IEC 13816:1997." *ISO*, 1 Oct. 2007, https://www.iso.org/standard/22987.html.
7. "ISO/IEC 30170:2012." *ISO*, 29 Apr. 2021, https://www.iso.org/standard/59579.html.
8. "ISO/IEC 9899:2018." *ISO*, 13 Oct. 2020, https://www.iso.org/standard/74528.html.
9. Itoh, Kosuke, et al. "Musical Pitch Classes Have Rainbow Hues in Pitch Class-Color Synesthesia." *Nature News*, Nature Publishing Group, 19 Dec. 2017, https://www.nature.com/articles/s41598-017-18150-y.
10. Jensen, Simon Holm, et al. "Type Analysis for JavaScript - Aarhus Universitet." *Paper.pdf*, Danish Research Council for Technology and Production, https://cs.au.dk/~amoeller/papers/tajs/paper.pdf.
11. Kennedy, Erik D. "The HSB Color System: A Practitioner's Primer." – *Learn UI Design*, https://learnui.design/blog/the-hsb-color-system-practicioners-primer.html.
12. *MSP430FR698x(1), msp430fr598x(1) Mixed-Signal ... - Ti.com*. https://www.ti.com/lit/ds/symlink/msp430fr6989.pdf.
13. Person. "UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter." *UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter | Analog Devices*, https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html.
14. Marks, L. E. (1975). On colored-hearing synesthesia: Cross-modal translations of sensory dimensions. *Psychological Bulletin*, 82(3), 303-331, https://doi.org/10.1037/0033-2909.82.3.303

15. Polyclick. "Who's Afraid of Creative Code?" *Medium*, DENKWERK STORIES, 11 Jan. 2018, https://medium.com/denkwerk-stories/whos-afraid-of-creative-code-2b774eb20023.
16. Published by Lee Jackson on August 1, and Lee Jackson. "OV2640 Vs OV7670: Spec Comparison, Sample Images, Datasheets, Lens Options, Pinouts,

Compatibility, and Tutorials.” *Arducam*, 18 Aug. 2021, https://www.arducam.com/ov2640-vs-ov7670-detailed-comparisons-and-resources/.
17. Raspberry Pi. “Raspberry Pi 4 Model B Specifications.” *Raspberry Pi*, https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/.
18. “ RoHS Guide.” *2021 RoHS Compliance Guide: Regulations, 10 Substances, Exemptions*, https://www.rohsguide.com/.
19. “Serial Communication Basic Knowledge -RS-232C/RS-422/RS-485.” *CONTEC*, https://www.contec.com/support/basic-knowledge/daq-control/serial-communicatin/.
20. “SPI Explained.” *SPI Explained | Dev Center*, developer.electricimp.com/resources/spi.
21. “Sound Color Project.” *Sound Color Project*, https://soundcolorproject.com/.
22. “Sparkfun Redboard Artemis.” *DEV-15444 - SparkFun Electronics*, https://www.sparkfun.com/products/15444.
23. Valdez, Jonathan, and Jared Becker. *Understanding the I2C Bus*, Texas Instruments, June 2015, https://www.ti.com/lit/an/slva704/slva704.pdf?ts=1635606612631&ref_url=https%253A%252F%252Fwww.google.com%252F.

Permissions

Permission to use information from Nature.com "Musical pitch classes have rainbow hues in pitch class-color synesthesia"

Rights and permissions

SPRINGER NATURE

Musical pitch classes have rainbow hues in pitch class-color synesthesia

Author: Kosuke Itoh et al

Publication: Scientific Reports

Publisher: Springer Nature

Date: Dec 19, 2017

Copyright © 2017, The Author(s)

Creative Commons

Copyrights

Please limit you response to 3000 characters

*Please write your questions or comments here

Hello, my name is Angel Garcia. I am a student at the university of central florida. I am currently writing a documentation for my senior design project and the document includes references to pictures and information on one of your articles named "UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter".
https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html

May I please have permission to use these references for my research paper?

Thank you.

SI Support io <Support.io@digikey.com>
Sun 10/31/2021 6:43 PM
To: You; support@maker.io

Angel,

With our absolute pleasure. There is no issue at all. We hope your project goes well and thanks for thinking of Digi-Key!!!

David and the Maker.io Team

CG Clint Goss <clint@goss.com>
Mon 11/1/2021 12:49 PM
To: You

Angel -

You have my permission!

Anything else, just let me know ...

-- Clint Goss
FluteCast.com ... direct links to Native American flute resources
Goss.com ... index of all our web sites

KB Kellen Beaver <KBeaver@soundproofcow.com>
Tue 11/2/2021 8:06 AM
To: You

Angel,

Good morning and thank you for reaching out. Being that you are using it as a reference in a research paper we have no issue with you using our article. Please let me know if you have any other questions at the moment.

Best,
Kellen

Great, thank you so much! | Thank you for the clarification. | Okay, thank you!

Are the suggestions above helpful? Yes No

Cristina González - DiCYT <agencia@dicyt.com>
Tue 11/2/2021 1:42 PM
To: You

Hola, buenas tardes. Sí, puedes utilizarla sin ningún problema, tan solo es necesario citar como fuente a la Agencia DiCYT.

Un saludo cordial y suerte con su investigación,

Cristina G. Pedraz, editora de DiCYT