UCF Senior Design II

SynthSign – An American Sign Language Controlled Synthesizer with a Laser Projection Display

Department of Electrical Engineering and Computer Science
CREOL College of Optics and Photonics
University of Central Florida
Dr. Lei Wei, Dr. Chung Yong Chan, Dr. Aravinda Kar

Advisory Committee
Dr. Patrick LiKamWa: CREOL
Dr. Mingjie Lin: ECE

CREOL Group 8
Tristan Barber: Computer Engineering
Christopher Jean: Computer Engineering
Christian Paredes: Photonic Sciences and Engineering
Jacob Goc: Photonic Sciences and Engineering

Table of Contents

List of Figures

List of Tables

1. Executive Summary

Whether professionally or as a hobby, many people identify with musical instruments as a means of self-expression. However, some musical instruments like the synthesizer, are known for being big, bulky, and expensive. These kinds of issues inhibit the ability to make music, as the price of a typical synthesizer may prevent some from even trying to pick up music. Beyond that, the modern-day synthesizer is difficult to transport, which prevents musicians from making music wherever they please. These kinds of issues are what our team aims to address. The synthesizer is a constantly evolving piece of equipment, and many modified versions have been released over the years. A rarity, however, is a synthesizer that can be played without the press of a single key. This is the kind of innovation our team wishes to create. Additionally, our team wishes for our synthesizer to provide a user experience greater than devices currently on the market, by implementing a projector display that can output our user interface onto any flat surface. However, it is important to note that these are some broad and overarching goals, and we have constraints of which we must adhere to. More details will be provided later, but it is important to lay out the fact that our team has sixteen weeks to prototype, build, test, and present our gesture-controlled synthesizer.

The gesture-controlled synthesizer will be implemented with the desire to allow musicians to create music on-the-go, without large equipment. To realize this idea, our gesture-controlled synthesizer will implement a custom-trained machine learning model that will detect various user gestures. The machine learning model will run on its own development board and will be connected to various components including a camera, an audio output device, and a laser projection unit. The laser projection unit will consist of a photodiode, which will emit the light for the projector, a photoresistor, which will help with synchronization, and two motors with mirrors, which will be used to guide the laser. At it’s heart will be an MCU, which will provide all of the control logic required to produce a quality user interface.

The report following this executive summary documents our entire process for researching, designing, prototyping, and testing our gesture-controlled synthesizer. First, more details about the project will be revealed, followed by all of our goals and objectives being laid out. Afterwards, an in-depth explanation of the project requirements will be showcased. Chapter 3 highlights our research and development for each of the key aspects of the project. The chapter following discusses the above-mentioned constraints in a much higher level of detail, as well as laying out various design standards the team will follow. From there, the actual design will be broken down into hardware and software, where each piece of the puzzle can be described and explained. Next, prototyping and testing is shown. Some sections will detail prototyping and testing completed, while others will lay out our plans for future prototyping and testing. Finally, the paper will discuss administrative efforts followed by a final conclusion.

2. Project Description

For this section, we begin to investigate the project. The project revolves around the idea of branching the innovative ways of optics with programming. We will do this primarily by utilizing the power of gesture recognition and laser projection in collaboration with software programming helping transition between each process. We understand that musician's tool around altering their music daily and thought of an idea in which we could revolutionize the way musicians did this. This project itself will act as a step into modernizing the way music enthusiasts compose and appreciate music.

2.1 Project Introduction

The purpose of this project is to design an all-in-one solution for music hobbyists to create and enjoy music, without the issues brought about by traditional equipment. The project will be highly technical in nature, with design requirements in software, hardware, and optics. Primarily, the device will use computer vision to detect gestures from a user and play various tones based on the gesture provided. The project will also have a graphic user interface via a laser projector, which will display information to improve the user experience.

2.2 Project Motivation

The idea to create a gesture-controlled synthesizer came from the music background that two of the group members share. Large amounts of people, whether for leisure or as a source of income, find themselves formally introduced to playing a musical instrument at some point in their lives. Music is an excellent hobby that allows people to flex their creative muscles and is an artform that allows people to express themselves. However, the problem arises with the price of the instruments themselves. Whether it’s the cost of the material that goes into making quality musical instruments, or the delicate and complicated design of certain instruments that requires skilled hands, price can be one deterrence from acquiring an instrument. Traditional setups costing so much can often be a barrier of entry for the average music enthusiast.

Additionally, quality synthesizers are non-portable and can often be difficult to transport for on-the-go use. A person with a traditional synthesizer must transport the keyboard itself, a stand used to support said synthesizer, and more than likely an audio output device like a speaker or amplifier. This is one of the issues our team aims to solve with our device. Portable options that do currently exist are expensive, and still often require auxiliary equipment.

The current products available do not solve all these issues at once. Our group strongly believes that our unorthodox solution will provide an alternative that encompasses the portable and cost-effective qualities that would benefit the music industry.

2.3 Goals

Our core goals include the ability to synthesize any of the 12 notes within an octave. This collection of notes is called a chromatic scale. We also want to provide a three-octave range, from notes C3 to B5. As this synthesizer will be controlled using hand gestures, a pivotal goal is to develop a machine learning object detection model to process user gestures. We also want the device to be able to interface with Bluetooth speakers.

The advanced goals we wish to achieve exist to provide the team with some exciting challenges. We aim to extend our octave range from three to five, giving the ability to synthesize notes from C2 to B6. Finally, we want to include a secondary mode, where users can select any one of the twelve major keys, and play preprogrammed chords based on the key signature selected.

Finally, our stretch goals. The team hopes to design a singular PCB to house all our electronics, doing away with any kind of off-the-shelf development kit. Additionally, we would like to give users the option to select between the twelve minor keys, as opposed to limiting them to major keys. We would also like to add a sustain functionality, which would imitate the sustain pedal on a piano. Allowing the user to map their own gestures would also be a fantastic feature. Finally, to improve musical performance capabilities, the team wants to develop a mechanical swivel interface that tracks the user as they move around the room.

2.4 Objectives

To achieve our core goals, the team has come up with a few key objectives. First, we have determined that our machine learning object detection model will need to recognize 9 distinct gestures. One gesture for each of the 7 notes in a C major scale, a gesture to raise an accidental flag, and a gesture to change the octave range. Additionally, we will have to implement a sound generation library within the software.

For our advanced goals, we'll need to broaden our scope. We will need to implement an additional 8 gestures. We need a gesture to select the key signature, and a gesture for each of the seven chords within a key signature. We will also need to expand our sound library to allow for the increased octave range.

Finally, we have a few key objectives for our stretch goals. We'll need to implement facial tracking within our machine learning model, which we will use to track a user's movement. Additionally, we will have to research servo and stepper motors to build our rotating platform. Finally, we will have to add additional gestures to change the laser projection unit display color and to toggle the tracking mode.

2.5 Existing Products and Projects

Since our project is something that hasn't been developed in the consumer world, so to best compare our project to other technologies, we will use three different technologies and compare them to the respective subsystems. First are portable synthesizers, which will be compared to the size and available settings of our project. Next is a former senior project: an ASL motion to text converter, that will be used to compare our gesture detection and technology needed to run. Finally, the closest thing we could find to our project: a keyboard-less synthesizer that works off the positioning of the users hands called the UFO.

Compared to the portable synthesizers found on the market, the key differences are the inclusion of a GUI separate from the synthesizer, the non-physical keys, the portability of the synthesizer, and the power supply. Since the GUI is separated from the synthesizer and is not made of solid material, we can add more customization, such as a dynamic note indicator. The main draw of the project, the gesture-controlled note production, is different from the physical keys, reducing the chance that a key may become broken, and reduces the distance one can control the synthesizer from. One drawback of our project, however, is how portable some of the smaller synthesizers are compared to ours. Ours may be more portable than some more expensive and advanced models, but it requires a camera setup and a housing for the GUI display, which are less portable than most conventional, small synthesizers, which can fit into backpacks. Most of these models are also battery powered, which is a double-edged sword: on one hand, the battery has a limited lifespan, and requires charging every so often. On the other, a wall plugin is not required for operation, allowing for use in the most remote areas.

The ASL project, for one, used a glove to improve detection, and allow for movement detection, flex detection, and gyroscopic motion. Having to wear a glove whenever one wanted to operate the device would be cumbersome, however, and if the glove were to become damaged from use, then the device would fail to work. The project also required the use of a computer and screen, which reduced the portability, compared to our project, which has a built in SBC and displays everything onto any surface the LPU is pointed towards.

Our project is a unique approach to creating music, and as such, there are a limited number of similar projects or products out there. One such project is the UFO, a synthesizer that also does not require the use of a physical keyboard, but rather used ultrasonic sensors. These sensors would detect the distance of the user's hands and convert it to a signal that would control what note was played, along with the volume. Although the concept is the same, the use of ultrasonic versus image processing provides a marginal slowdown to the response time, as the speed of sound is orders of magnitude slower than the speed of light. The size of the apparatus is also concerning, as it takes up a majority of the desk in the picture provided on the website. There does not seem to be a GUI that is separate from the computer, meaning the only possible way to use the UFO is when there is a readily available PC, reducing the portability of the device even further. On the other hand, the UFO does provide interfacing with sound development programs, something that our project is not planning to include.

2.6 Requirements Specifications

Table 1 - Engineering Requirements

Requirement	Specification	Description
Gesture Detection Accuracy	Shall meet or exceed 80% of captured gestures.	The correct response should be produced from a given gesture for at least 80% of all gestures.
Dimensions	Maximum size of 12" x 12" x 6" with a weight not exceeding 20 pounds.	The system should be small enough for easy transportation.
Response Time	Less than or equal to 2 seconds	Device must detect and respond to a gesture within two seconds.
Recognition Distance	Less than or equal to 5 feet	User input should be detected within 5 feet directly in front of the camera.
Camera FOV	Horizontal: 80 – 120 degrees Vertical: 45 – 60 degrees	Under these conditions, the user may not have to be directly in front of the camera for gestures to be detected.

Requirement	Specification	Description
Power	120VAC 60Hz power, via a type B wall plug.	Power must be easy to obtain to promote portability.
Laser Projector Projection Region	Maximum resolution of 12” x 9” at 12” from any flat surface	Device data must be easily relayed to the user.
Camera Resolution	At least 720x1280	The images must be of high enough quality for the image processing algorithm to function properly.
Cost	All materials should not exceed $400	Cost should be limited to remain affordable.

2.7 House of Quality

The House of Quality (HOQ) is an important tool for product planning, used in the early development of many devices. This figure, shown below as Figure 1, is designed to establish how the marketing and engineering requirements relate to each other, as well as how the engineering requirements relate to themselves. Our marketing and engineering requirements are defined in the above Requirements Specifications section.

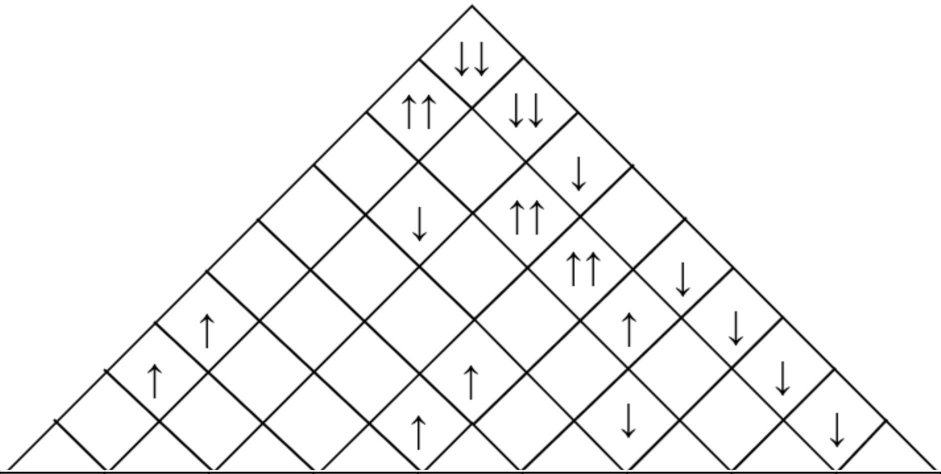
Table 2 - Marketing Requirements

Marketing Requirements
1. The device should have a low cost. 2. Images should be of high enough quality to successfully be detected. 3. Object detection should be reliably accurate. 4. The device should be low maintenance. 5. The device should be portable and lightweight. 6. The device should be easy to use.

Within the House of Quality, each marketing requirement is weighted against each engineering requirement, and if there is correlation, the proper correlation arrow is placed in the matrix box associated with the engineering-marketing intersection. The four possible correlation options are as follows: Positive Correlation, Strong

Positive Correlation, Negative Correlation, and Strong Negative Correlation. The associated symbols are shown in the legend below the HOQ in Figure 1. Additionally, each marketing or engineering requirement has a distinct plus or minus symbol. These symbols depict whether is imperative to maximize (+) or minimize (-) the output of the device with respect to that requirement. Finally, the triangle portion of the diagram weighs the correlation of each of the engineering requirements with the other engineering requirements. This is important, as increasing the performance of one parameter may have an effect on another. This is not done within the marketing requirements, because they do not represent a specific component of the device in question.

An example within our device is the Negative Correlation between the ‘Battery Life’ engineering requirement and the ‘Cost’ engineering requirement. While increasing the battery life (mAh) of the battery pack is important, it is likely that a more capable battery pack will also cost more. Another example, within the marketing-engineering comparison is the Strong Positive Correlation between the ‘Transportability’ marketing requirement and the ‘Dimensions’ engineering requirement. Decreasing the dimensions of the device will allow for easier transportability, as the device would be more likely to fit into a container or bag without difficulty.



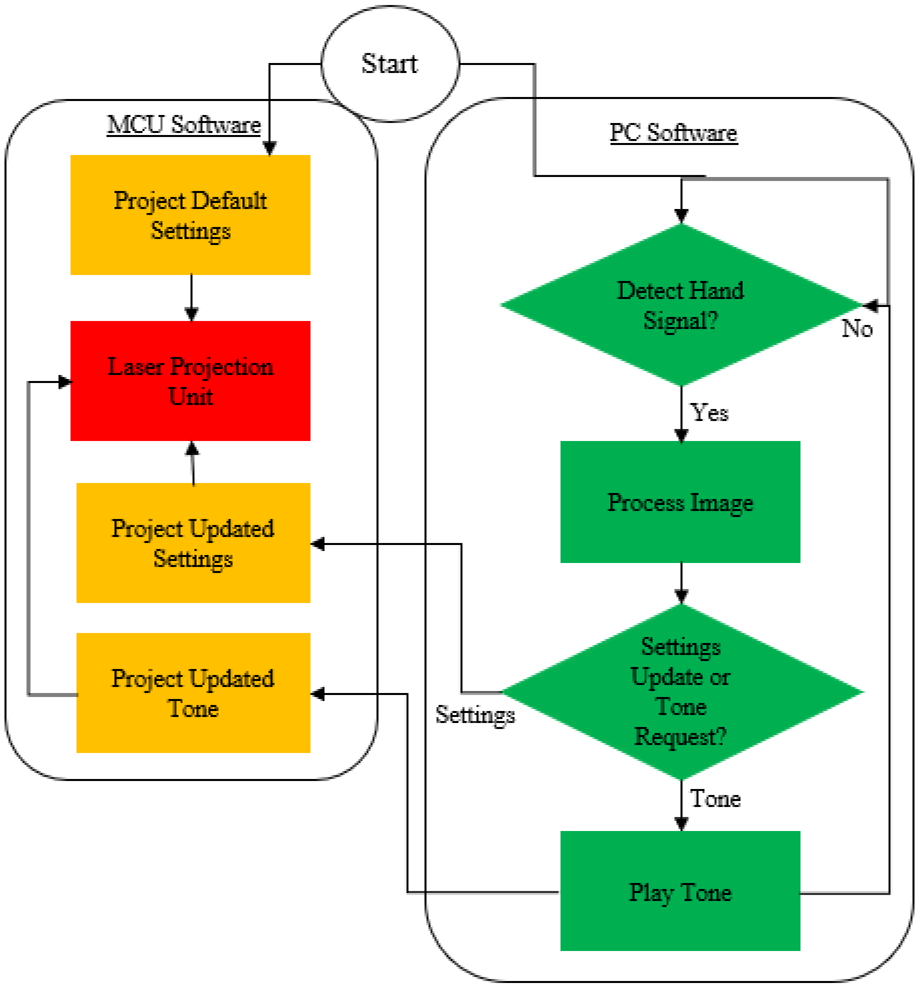
		Gesture Recognition Accuracy	Size and Weight	Response Time	Recognition Distance	FOV	Power	LPU Resolution	Camera Resolution	Cost
		+	-	+	+	+	+	+	+	-
Cost	-				↓↓	↓		↓↓	↓↓	↑↑
High Image Quality	+	↑↑		↑	↑↑	↑			↑↑	↓↓
High Performance	+	↑↑		↑	↓	↑			↑↑	↓↓
Low Maintenance	-		↑↑				↑↑			↑↑
Transportability	+		↑↑				↑↑			↓↓
Ease of Use	+	↑	↑	↑↑			↑	↑		↓
Target for Engineering Requirements		>80%	12"x12"x6" and ≤ 10 lbs	≤ 2 seconds	≤ 10 feet	90° - 120° H & 45° - 60°V	Type B Wall Plug	12"x9" at 12" from surface	≥720x1280	≤ $400

Legend	
+	Increasing this requirement increases the desirability of the device.
-	Decreasing this requirement increases the desirability of the device.
↑	Positive Correlation
↑↑	Strong Positive Correlation
↓	Negative Correlation
↓↓	Strong Negative Correlation

Figure 1 - House of Quality

2.8 Software Block Diagram

The following figure shows the software block diagram. The goal of this figure is to detail each component within the software, and how each component interfaces with one another to form the complete system. As is evident from the figure, the software breaks up into two key development areas. The image processing computation takes place on the PC and includes gesture recognition and sound generation. The other portion of the software, being implemented onto the microcontroller development board, controls the laser projection display, which acts as the graphic user interface for the device. Additionally, it will control the variable zoom lens based on data. The variety of demands between hardware, software, and optical design caused our group to split ourselves into areas of development that we feel confident we can complete, while still growing our skills in our personal areas of interest. It's important to note that the software diagram only has a status association within each primary area of development. This is unlike the following figure regarding hardware, which has a status association for each block. It should be noted that the Laser Projection Unit block within the MCU software bubble does not actually represent a piece of software. That block exists to show how data flows within the flow chart.



Legend

Member Block Assignment

- Tristan Barber
- Jacob Goc
- Christopher Jean
- Christian Parades

Status Codes

To Be Acquired (TBA) – block will be purchased or donated
Acquired (A) – block has been donated or purchased
Research (R) – block design approach is being investigated
Design (D) – block is currently being designed
Prototype (P) – block design is currently being prototyped
Completed (C) – block design is a finished prototype

Figure 2 - Software Block Diagram

2.9 Hardware Block Diagram

The following figure shows a block diagram created to provide a visual representation of the hardware components and how they work amongst each other. Additionally, each block provides detail about who is responsible for each component, and what phase of development the component is in. The two centralized blocks, PC and MCU, exist to show how and where data from individual hardware is routed to be processed by software. Arrowheads are used to show data flow between different areas of hardware. Like the software breakdown, members of the group assigned themselves tasks based on areas of personal interest and confidence.

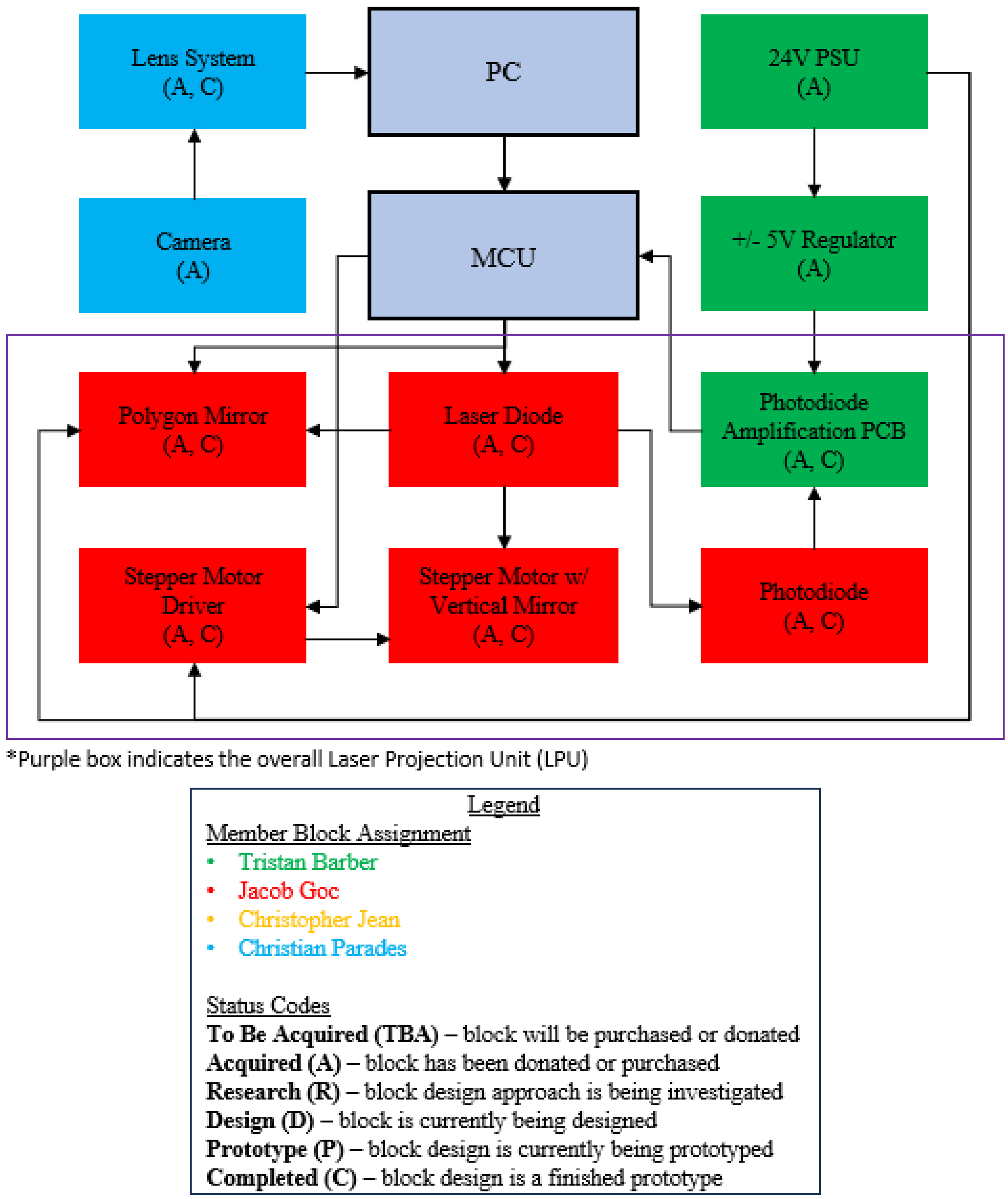


Figure 3 - Hardware Block Diagram

2.10 Optical Diagram

For this section, we will discuss the optical schematic diagram that will be further dissected in the following sections later in the report. The project itself will consist of two optical designs provided by our following photonics members. It will comprise the two-lens system and the laser projection unit's design. The lens system will comprise of the raspberry pi high quality camera, followed by a bi-convex and plano-convex lens. It contains a physically structured aperture to help further enhance our distance requirement. The laser projection unit itself will involve the utilization of an instapark DRM104-D003 laser diode, with a photodiode and a controlled mechanism. This mechanism will consist of an aperture and a polygonal mirror that is driven by a DC motor. Finally, the projection unit will output the resulting display. Below, the schematic diagram illustrates the two optical designs highlighted to help showcase where each lies in respect to the overall diagram.

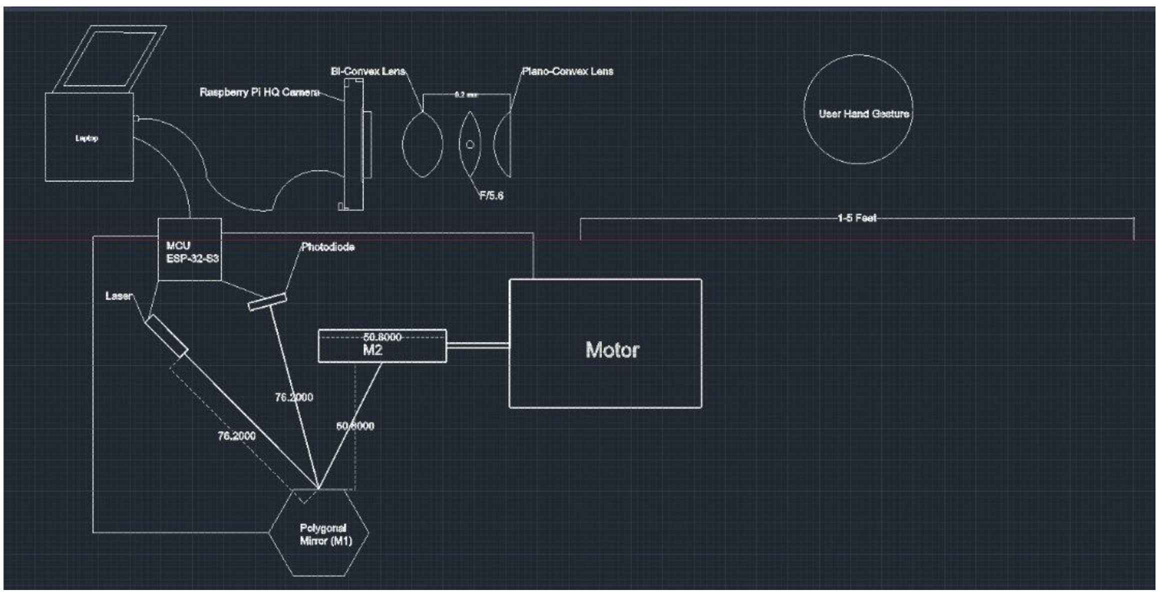


Figure 4 - Optical Schematic Diagram

3. Investigation and Part Selection

Research and investigation was important for this project in large part because of the many concepts that we were not familiar with. The fact that for some of us, this was their first interaction with real world designing and cooperating with others outside their majors was daunting. Understanding how each individual part works, what the properties they bring to the project are, and how they will work with the other systems in the project are all something the group learned throughout this semester. Therefore, selecting these key parts and making sure they will work with the other components was key in developing our project.

This section details every key part in the synthesizer and LPU. Each part has been laid out in the pages below, and within each section, there is a comparison between at least two different products. We will discuss the thought process behind our choices, what the upsides and downsides of each product are, and what we will be prioritizing.

The part research and investigation section is broken up into nine main subsections, starting with the single-board computer, the brains of the project, which will be controlling the image processing of the gestures, as well as communicating with the microcontroller to create the GUI for the synthesizer. Following this is the machine learning object detection model that will be on the SBC. Our MCU choice will be further discussed afterwards, and to nicely finish off our computer design, the serial communication technology, which is how our SBC and MCU will communicate, will also be decided upon here.

On the optical side, we first decide upon the camera we will be using, and then the lens imaging system that will be attached to it. Since the LPU will have large amounts of components, the section will be divided between the mirrors, the laser diode, the motor, and the photodiode. Two smaller sections follow these, with one being the power source, and the other being the audio output we will be allowing.

3.1 Single-Board Computer Versus PC

Our gesture-controlled synthesizer requires the use of a machine learning (ML) inferencing compatible SBC that will incorporate computer vision to detect and differentiate different hand signals from the user. This component, being the core of the device, would be required to meet a plethora of criteria. These criteria may include compatibility with a Camera Serial Interface (CSI), tensor processing compatibility, Bluetooth compatibility, Audio output compatibility, and minimal power requirements. These criteria require the use of an SBC with an onboard processor capable of handling and accelerating machine learning inferencing. Fortunately, artificial intelligence (AI) and ML projects incorporating computer vision have skyrocketed in popularity in the recent times, so there are plenty of SBCs to select from.

3.1.1 SBC Options

Below is a list of all SBC development kit options considered for selection by our group. Some of these selections were eliminated at an early stage of selection, and the reason for their elimination is included. The more promising options that survived the initial round of selection are compared further within a later sub-section of 2.1.

3.1.1.1 Coral Dev Board

The Coral Dev Board from Google was an initially very promising option. The board contains Google's Edge Tensor Processing Unit (TPU), which is a hardware accelerator designed specifically for AI and ML workloads. Additionally, the Coral

Dev Board supports TensorFlow Lite frameworks. Some other notable features are described below:

- Bluetooth 4.2 Compatible
- On-board 3.5mm audio jack
- 40 GPIO Pins
- PyCoral API for Python
- Libcoral and Libedgetpu API for C++
- Plethora of documentation and resources

However, this product is not without its drawbacks. The major issue with the Coral Dev Board is the price. There are only two models available for purchase on the coral.ai website. The first model is available with 1 GB of RAM, at an MSRP of $129.99. The other model ships with 4 GB of RAM, and retails at $169.99. Additionally, as previously mentioned, the only native framework support for the Coral Dev Board is TensorFlow Lite. As a result, any incompatible ML models would have to be converted, curbing the potential for our team to use a large majority of the currently available open-source models. Ultimately, despite the promising performance, the price of this product sits between 33 to 43 percent of our total budget. This factor, along with potential compatibility issues, caused our team to eliminate this option.

3.1.1.2 Raspberry Pi 4

Another single-board computer considered by our group is the tried-and-true Raspberry Pi 4. The Raspberry Pi 4 is significantly more affordable than the previous example, while still boasting significant performance. There are three product options within the scope of our device, with the 1 GB model pricing in at $34.99. Additionally, the 2 GB version MSRPs for $44.99 and the 4 GB version for $54.99. Relevant technical specifications are shown below:

- 2-lane MIPI CSI camera port
- 40 pin GPIO Header
- TensorFlow and PyTorch framework support
- Bluetooth 5.0

The Raspberry Pi 4's price, along with its availability, make it a viable option for further consideration. Without any additional hardware acceleration, however, the Pi 4 may struggle in some respects. With that being said, the group has determined that the 4 GB version, specifically, will be considered further.

3.1.1.3 Raspberry Pi 4 + USB Accelerator

The major drawback with the Raspberry Pi 4 is the lack of onboard dedicated AI acceleration hardware. A potential solution is to include an additional USB Accelerator to handle the AI and ML tasks. Our group compared two USB Accelerator options, and that information can be found below.

3.1.1.3.1 Coral USB Accelerator

A potential solution to the Raspberry Pi's lack of hardware acceleration is the Coral USB Accelerator. This product retails at $59.99 and offers dedicated AI hardware via the Edge TPU for use with the Raspberry Pi processor. However, many of the issues found with the Coral Dev Kit are prevalent with the Coral USB Accelerator. One problem is the lack of support for TensorFlow, which limits the pre-trained ML models available. Additionally, the Coral USB Accelerator is limited to inference models only, so our stretch goal of user-mapped gestures would become out of reach.

Despite the limitations of the Coral USB Accelerator, the lack of multiple better options available on the current market caused us to consider the Raspberry Pi 4 + Coral USB Accelerator for further research and comparison. The addition of the Coral USB Accelerator makes total memory less of an issue, so the 2 GB version of the Raspberry Pi will be considered in conjunction with the Coral USB Accelerator.

3.1.1.3.2 Intel Neural Compute Stick 2 (NCS2)

The other consideration for an external USB Accelerator was the Intel Neural Compute Stick 2. This product is equipped with the Intel Myriad X, with is a Visual Processing Unit (VPU) specifically designed for AI and ML workloads. The NCS2 also supports a wider range of frameworks, including TensorFlow. The NCS2 was available at an MSRP of $86.92.

Similarly, to the Coral USB Accelerator, it is limited to inference. The biggest problem with this product is that it has reached the end of its life cycle, with no successor being announced. The last order date for the Intel NCS2 was February 28th, 2022, and technical support for the product ended June 30th, 2023. This means that despite the promising performance and documentation, we may struggle to obtain one at an affordable price. Currently, Intel NCS2's can be found for sale on third-party resell websites used, for well above MSRP. The performance limitations in tandem with the lack of obtainability caused us to rule out the Intel Neural Compute Stick 2.

3.1.1.4 Sipeed MAIX Go Suit

The Sipeed MAIX Go Suit is a versatile AI development kit engineered for edge AI and machine learning applications. At its core lies the Sipeed MAIX module, boasting a Kendryte K210 dual-core RISC-V processor complemented by a dedicated AI co-processor. This combination empowers developers to harness the power of AI acceleration for real-time image and video processing, voice recognition, object detection, and more. With its compact form factor and integrated components such as a camera, microphone, and speaker, the MAIX Go Suit offers a convenient and cost-effective solution for AI experimentation, prototyping, and deployment on the edge.

This board has some unique components that drew our team in. The idea of a dual core processor with an additional AI co-processor led us to believe that we may be able to successfully use this board for control of the speaker, as well as the image processing algorithm. Additionally, the Sipeed MAIX Go Suit ships with a built in CSI camera-module, which was a huge selling point. That fact alone has caused the group to reconsider camera integration, and whether or not we would be better suited connect the camera to the SBC or the separate MCU.

However, the major downside to the Sipeed MAIX Go Suit is its lack of native Bluetooth support. At this point in development, our team is unsure whether the speaker control will be handled by the SBC or the MCU. However, it is important that both products have native Bluetooth support, so we have the option to choose what's best suited for the design. Ultimately, the lack of native Bluetooth support was enough to eliminate the Sipeed MAIX Go Suit early on in our investigation.

3.1.1.5 NVIDIA Jetson Nano Developer Kit

The NVIDIA Jetson Nano was quickly brought to the team's attention as a popular option for an SBC built for AI and ML applications. The NVIDIA Jetson Nano is equipped with a quad-core ARM Cortex-A57 CPU and a 128-core Maxwell GPU with CUDA support, making it capable of performing AI and machine learning tasks with high performance and efficiency. Additionally, it supports popular deep learning frameworks like TensorFlow, PyTorch, and Caffe, meaning that the team would not be limited to designing our own ML model, should an open-source model already exist. Finally, The Maxwell GPU with CUDA support provides hardware acceleration for AI and computer vision tasks, significantly speeding up inference times.

There are two options from NVIDIA. The more affordable version of the Jetson Nano Developer Kit ships with 2 GB of LPDDR4 RAM, with an MSRP of $59.99. Additionally, there is a more powerful option that, a 4 GB version with an MSRP of

$99.99. Although there are minute differences between the models, the key shared technical specifications are shown below:

- 4Kp30 | 4x 1080p30 | 9x 720p30 (H.264/H.265) video encoder
- 4Kp60 | 2x 4Kp30 | 8x 1080p30 | 18x 720p30 (H.264/H.265) video decoder
- microSD expandable storage
- 40-pin GPIO header

The major component missing from the Jetson Nano is native Bluetooth support. There is native support for Wi-Fi, using the 802.11ac protocol, but that is not necessarily within the scope of our device. However, the team was willing to look past that fact due to the incredible price to performance ratio found within the Jetson Nano. Unfortunately, similarly to the Intel Neural Compute Stick 2, global supply chain issues have caused the Jetson Nano series to be discontinued. As a result, these devices are only available on third-party websites, and for prices significantly above MSRP. The revised prices are comparable to the Coral Dev Board discussed earlier, but with worse performance. This was a deciding factor in the elimination of the Jetson Nano developer kit from our consideration.

3.1.1.6 Arduino Portenta H7

The Arduino Portenta H7 is a high-performance development board designed for a wide array of applications, from IoT projects to advanced computing tasks. It boasts a dual-core Arm Cortex-M7 and Cortex-M4 processor, wireless connectivity, and Arduino IDE compatibility. Its modularity and industrial-grade features make it a versatile choice for both beginners and experienced developers. Additionally, the Portenta H7 is marketed to run processes created using TensorFlow Lite, so it’s ability to run ML tasks is on par with the Coral USB Accelerator. Some key features are shown below:

- Powered by an STM32H747XI microcontroller with Cortex-M7 and Cortex-M4 cores.
- Native Wi-Fi and Bluetooth support
- Additional storage can be added via an M.2 Expansion Slot
- Programmable using the Arduino IDE

While the Arduino Portenta H7 offers impressive capabilities, it does come with some considerations. Its advanced features may pose a challenge for beginners, who may find its complexity intimidating. Additionally, its price point may be higher than other Arduino boards, which could be a factor for users on a tight budget. Finally, the board's specific power requirements (5V to 28V) might necessitate specific power sources or adapters, which could be less accessible for all users.

Despite these concerns, our group has selected this product for final comparison due to its price to performance, and the lack of availability of some competitors.

3.1.2 Extensive SBC Comparison

A total of seven Single-Board Computer configurations were initially considered by the team, and a brief overview of each has now been listed. Out of those seven, the remaining three are going to be compared under further scrutiny. The following section compares these three in the area of CPU and GPU clock frequency, cost, memory size and speed, and the incorporation of other individual features that may prove useful during the software development process.

3.1.2.1 SBC CPU, Graphics and Memory Comparisons

In the table shown below, the performance of the CPU and GPU for each of the Single-Board Computers is compared, as well as the RAM and Flash memory. The performance of these components is critical, because without the ability to recognize gestures almost instantly, the gesture-controlled synthesizer cannot function. Additionally, the SBC has to communicate with the PCB, and may directly communicate with the camera and speaker.

Table 3 - SBC CPU, GPU, and Memory Comparison

	Arduino Portenta H7	Raspberry Pi 4	Raspberry Pi 4 + Coral USB Accelerator
CPU	Dual-core STM32H747	Quad-core ARM Cortex-A72	Quad-core ARM Cortex-A72
GPU	Chrom-ART Accelerator	VideoCore VI	VideoCore VI and Edge TPU
CPU Clock Frequency	Cortex M7: 480MHz Cortex M4: 240MHz	1.8 GHz	1.8 GHz
RAM	8MB SDRAM	4GB LPDDR4	2GB LPDDR4
Flash Memory	16MB NOR Flash	Expandable storage via microSD or USB	Expandable storage via microSD or USB

As the table shows, the Pi 4 has a significantly higher clock frequency than the Portenta H7. Additionally, the Pi 4 has two more cores than the Portenta H7, and they can all run at the 1.8 GHz clock rate. However, the GPU is going to play the biggest role in winner of this category. The Edge TPU coprocessor found on the Coral USB Accelerator is greatly more powerful than the onboard graphics found within the Portenta H7 and Pi 3 SBC's. Although the Edge TPU is not very flexible, as it can only process TensorFlow Lite models, its performance is still enough to rank highest in this category.

Additionally, the Portenta H7 and Raspberry Pi 4 use different memory configurations. The Portenta H7 uses SDRAM, which can typically have higher performance than LPDDR4, but with higher power consumption. Unfortunately, the performance increase does not trump the exponentially decreased memory capacity. With that being said, the Raspberry Pi 4 lacks a traditional Flash Memory setup, giving the Portenta a clear advantage in that regard. Still, the Raspberry Pi 4 4GB takes this category, as it has the greatest total memory, not considering any other components.

3.1.2.2 SBC Pricing

As this project is self-funded, the price of each product does play an important role in deciding what to go with. Given the fact that this SBC is only one component within this highly complex device, it needs to take up as little of the total budget as possible.

With that being said, the total budget is also flexible, as the team wants to build an end-product that will work and work well. We have all agreed that function is the most important aspect, followed closely by price. The least important factor in our eyes for the SBC is ease-of-use. The end-user won't have any direct interfacing with the SBC, so the SBC is not an ease-of-use critical component.

The Raspberry Pi 4GB model prices in at an MSRP of $54.99. The Coral USB Accelerator has an MSRP of $59.99, and the additional cost of the Raspberry Pi 2GB ($44.99) brings the total for the pair to $104.98. The most expensive of the three is the Arduino Portenta H7. Currently, the SBC is available on Arduino's website for $113.90. Ultimately, from a price-critical perspective, the Raspberry Pi 4 standalone has the greatest advantage.

3.1.2.3 SBC Development Software Suite Comparison

While the team is motivated to choose the best product on the primary basis of performance, we still find it important to consider the limitations of each device in regard to their individual programming interfaces and library support. For this comparison, the team compared the various options for programming the SBC's and what code libraries were supported, primarily libraries for ML inferencing.

The Portenta H7 has two relevant IDE options for programming a ML model. The Portenta H7 can be programmed using C++, through the Arduino IDE. However, unlike many typical Arduino devices, the Portenta H7 supports MicroPython development using the OpenMV IDE. The primary difference between MicroPython and Python is that MicroPython does not implement the entire Python 3 standard library. At this time, Python is the language of choice for development of AI and ML models, so MicroPython support from the Portenta H7 is a plus. The Portenta H7 presently supports the TensorFlow Lite framework for ML inferencing. The lack of full TensorFlow support is not necessarily a dealbreaker, but it does limit the development options, and may hinder our ability to use a pre-trained ML model if an open-source model exists.

The Raspberry Pi 4 is in a slightly different class than a Portenta H7, as it's a true single-board computer capable of booting with many different Linux operating systems. That fact gives it a distinct edge, as there is virtually no limit from a programming language standpoint. For ML development specifically, the Pi 4 can run Python 3, which gives an advantage over the Portenta H7. From a ML framework standpoint, the Pi 4 standalone is also significantly less limited. The Pi 4 can support the TensorFlow, TensorFlow Lite, Keras, and PyTorch frameworks. This fact brings about much less limitation for development, and further increases the possibility of using an open-source pre-trained ML model.

Including the Coral USB Accelerator will introduce a framework limitation. Although development with the Raspberry Pi 4won't drastically change, the Coral USB Accelerator only supports the TensorFlow Lite framework. This lessens the flexibility that gives the Pi 4 most of its edge. With that fact in mind, the Raspberry Pi 4 standalone is the best choice from a software development point of view.

3.1.2.4 SBC Individual Features Comparison

Within the products being considered, small differences can be found within the individual features they ship with. This section highlights these small differences and discusses how these specifications may help, or hinder, the development process.

A major area of importance is audio data transfer, as the SBC may be responsible for communicating directly with the speaker. As previously mentioned, the synthesizer will support wired speaker connection via a 3.5mm audio jack and Bluetooth compatibility for wireless Bluetooth speakers. The Raspberry Pi 4 has native support for Bluetooth 5.0 and has an on board 4-pole stereo audio 3.5mm jack. The Portenta H7 supports Bluetooth 5.0 via a Cordio stack and Bluetooth 4.2 via an Arduino stack. However, the Portenta H7 does not have an onboard 3.5mm audio jack. The Portenta H7 does have six Analog Pins, so a 3.5mm audio jack would be able to be added later. Nonetheless, the Raspberry Pi 4 has an advantage in this regard.

The USB I/O requirements are very different between the two boards, as the Raspberry Pi 4 will have an operating system and will be flashed via a microSD card. This means that during the development phase, the Pi 4 will be required to support mouse and keyboard connectivity. The Pi 4 has four onboard USB ports, with two being USB 3.0 and two USB 2.0. The keyboard and mouse will not benefit from a USB 3.0 connection. However, if the Coral USB Accelerator is implemented, it will utilize one of the two USB 3.0 ports, because USB 2.0 provides a major performance bottleneck for the Edge TPU. In contrast, the Portenta H7 only has one USB-C port, used for power delivery, DisplayPort out, and software flashing from host to device. As the software flashing process is different, a mouse and keyboard would not be required for development using the Portenta H7.

Another feature important for the SBC is camera interface support. The Raspberry Pi 4 has one 2-lane MIPI CSI camera port, which is a 12-bit protocol supporting data transfer rates up to 1Gbit/s. The Portenta H7 has an 8-bit CSI port, with a speed of up to 80MHz. Although the ML model likely will not require 1Gbit/s speeds, the major important difference between the two is the larger data bus found on the Pi 4, making it better suited to handle this task.

In regard to display capabilities, the Raspberry Pi 4 has two on-board Micro HDMI ports, one being H265 (4kp60 decode), and one H264 (1080p60 decode, 1080p30 encode). The Portenta H7 has a DisplayPort out through the USB-C power connector. Although the final synthesizer design will have a Laser Projection Unit to display a GUI, video out from the SBC will be useful for the development process. HDMI is a sufficient protocol for the display support we would need, so this portion of the comparison did not affect the groups decision.

Ultimately, the Raspberry Pi 4 has a greater overall advantage in terms of on-board individual features. However, as two of our proposed designs use the Pi 4, the team wanted to determine a concrete winner for this category. It was determined that because the Coral USB Accelerator will utilize one of the available USB 3.0 ports, the Pi 4 standalone option has technically more flexibility for future possible need. With that in mind, the Pi 4 standalone is highest ranked for individual features.

3.1.2.5 SBC Inferencing Benchmarking Comparison

The primary scope of the SBC is object inferencing, so it's important to compare the further selected SBCs specifically on object inferencing benchmarks. It's important to note that even though the Arduino Portenta H7 is considered a recommended model for ML object detection, the team was unable to find any benchmarking data for the three SBCs that all used the same model. With this in mind, the Raspberry Pi 4 (with and without the Coral USB Accelerator) benchmark results will be evaluated via a table, and the results can be compared to available findings for the Portenta H7. In the table below, object detection inference time for the Raspberry Pi 4 based devices are shown. These benchmarks were sourced from *Benchmarking TensorFlow Lite on the New Raspberry Pi 4, Model B,* by hackster.io. As expected, the Coral USB Accelerator provides insurmountable performance increases, thanks to its dedicated Edge TPU. It should be noted that DNR stands for "Did Not Run", as the Coral USB Accelerator does not support TensorFlow.

Table 4 - Raspberry Pi 4 time per inference

Model	Framework	Raspberry Pi 4	Coral USB Accelerator (USB 3)
SSD MobileNet v1 (300 x 300)	TensorFlow Lite	82.7 ms	14.9 ms
SSD MobileNet v2 (300 x 300)	TensorFlow Lite	122.6 ms	18.2 ms
SSD MobileNet v1 (300 x 300)	TensorFlow	263.9 ms	DNR
SSD MobileNet v2 (300 x 300)	TensorFlow	483.5 ms	DNR

The most in-depth benchmarking data available for the Portenta H7 comes from an article by EloquentArduino. This article compares the Portenta H7 against two other devices with an ARM Cortex M7 CPU. The devices are a Teensy 4.0 and an STM32 Nuclei H743ZI2. 54 runs were completed in total, with each run consisting of 6 classifiers in each of 9 datasets. Importantly, the ML models used for these benchmarks are simpler in comparison to the MobileNet models used for the Raspberry Pi 4 variants. As shown in the table below, the Portenta H7 had the least number of 1st place rankings among the three, but consistently better performance comparatively.

Table 5 - Arduino Portenta H7 benchmarks vs similar ARM Cortex M7 devices

Overall Performance	Arduino Portenta H7	Teensy 4.0	STM32 Nucleo H743ZI2
1st Place	16 run(s)	45 run(s)	24 run(s)
2nd Place	35 run(s)	1 run(s)	5 run(s)
3rd Place	3 run(s)	8 run(s)	25 run(s)

Although it's difficult to compare these performance metrics to the Raspberry Pi 4 performance metrics, these do tell an important story. The Portenta H7, being nearly five times the cost of the Teensy 4.0, is not capable of handling the ML object detection algorithms required for this project.

3.1.3 SBC Selection

In order to keep the SBC product choice equitable, our team decided to implement a trade study based on the areas of in-depth comparison. Each product was assigned a ranking for each section, and each section was assigned a weight. The products weighted score for each category was defined by multiplying the weight and the rank. The product with the lowest average weighted score is considered best suited.

Table 6 - SBC Trade Study

Category	Weight	Arduino Portenta H7	Raspberry Pi 4 4GB	Raspberry Pi 4 2GB with Coral USB Accelerator
CPU and GPU	20%	3	2	1
Pricing	15%	3	1	2
Memory	20%	3	1	2

Category	Weight	Arduino Portenta H7	Raspberry Pi 4 4GB	Raspberry Pi 4 2GB with Coral USB Accelerator
Software	10%	3	1	2
Individual Components	10%	3	1	2
Benchmarks	25%	3	2	1
Weighted Average		3	**1.45**	1.55

As shown in the table above, the group has determined that the Raspberry Pi 4 4GB is the best device for the job. Despite its secondary placement in some performance areas, the flexibility and price to performance is what ended up making it the winner. Plus, this selection allows the team to save some money, without hindering the ability to upgrade in the future if need be. The Coral USB Accelerator will still work with the Raspberry Pi 4 4GB, so if more performance is needed later, it can be added with relative ease.

3.1.4 SBC Shortcomings Versus PC

Following the SBC selection, testing began, and it was determined that the SBC did not have the processing capacity required to run the multiple machine learning models found within the design. As a result, the switch from an SBC to a PC was made. Below is a table highlighting the recorded accuracy of the software running on the Raspberry Pi 4 versus the PC. As shown, our specification of 80% was not able to be met using the Raspberry Pi 4.

Table 7 - Justification for Switching to Laptop

Raspberry Pi 4:	1	2	3	4	5	6	7	8	9	10	% Correct	Laptop:	1	2	3	4	5	6	7	8	9	10	% Correct
A:	A	A	A	A	A	A	A	A	A	A	100%	A:	A	A	A	A	A	A	A	A	A	A	100%
B:	B	B	F	B	D	B	F	B	B	B	70%	B:	B	B	B	B	B	B	B	B	B	B	100%
C:	C	C	O	C	E	C	C	O	C	O	60%	C:	C	C	C	C	C	C	C	C	E	C	90%
D:	D	D	D	D	D	F	D	D	F	D	80%	D:	D	D	D	D	D	D	D	D	D	D	100%
E:	E	E	E	A	C	E	E	E	C	E	70%	E:	E	E	E	E	A	E	E	E	E	E	90%
F:	F	F	F	F	F	B	F	F	F	F	90%	F:	F	F	F	F	F	F	F	F	F	F	100%
G:	G	G	G	G	G	G	G	G	G	G	100%	G:	G	G	G	G	G	G	G	G	G	G	100%
Octave (ASL O):	O	O	C	O	O	C	O	O	O	E	70%	Octave (ASL O):	O	O	O	O	O	O	O	O	O	O	100%
Accidental (ASL K)	K	D	D	G	D	K	G	D	K	D	30%	Accidental (ASL K)	K	D	K	K	K	K	K	D	K	K	80%

3.2 Machine Learning Object Detection Model

This section details how the team decided what machine learning object detection model to use in conjunction with our SBC. Comparisons include, but are not limited to, parameters such as resource use, documentation, ease of implementation and individual features and aspects. The four ML models we will be considering include TensorFlow, TensorFlow Lite, PyTorch, and Keras.

3.2.1 Individual Model Descriptions

The following sections give an overall description for each of the products. Each description encapsulates the pros and cons and highlights any features that allow a model to stand out. Following these descriptions, our selection will be highlighted, and explanations for our choice will be laid out.

3.2.1.1 TensorFlow

TensorFlow, a powerful framework from Google, provides access to a wide range of pre-trained models and tools for building custom neural networks. It has widely been used for image recognition and object detection. Furthermore, as TensorFlow is one of the leading models, there is a plethora of documentation and community forums to reference when attempting to mitigate hiccups. A huge plus is that TensorFlow's cross-platform functionality allows for custom models to be generated on a powerful machine, then deployed onto the PC.

3.2.1.2 TensorFlow Lite

A sub-development of TensorFlow, TensorFlow Lite was created specifically as a lightweight alternative more capable of generating good results on a resource limited device. There are some pros specific to TensorFlow Lite, including the conversion flexibility. TensorFlow Lite allows for conversion from other frameworks like TensorFlow or PyTorch. This is an advantage, as any pretrained models that may exist don't necessarily have to be created in TensorFlow Lite for our group to utilize them. Additionally, TensorFlow Lite has offline inference capability, so inferencing can be done without an internet connection. Finally, likely the largest advantage for TensorFlow Lite, is actually the limitation of the Coral USB Accelerator. As previously discussed, the team does not plan on using the Coral USB Accelerator. However, we have concluded that in the event that we need more capability later on, the Coral USB Accelerator will be used to upgrade the Pi 4. Furthermore, the Coral USB Accelerator is limited to processing TensorFlow Lite models, so if we choose to use it later, we'll have to adapt to a TensorFlow Lite model anyways.

Understandably, not all TensorFlow models can be converted to TensorFlow Lite, and the team can't rely on that being a functionality. Additionally, there are performance trade-offs to consider. While TensorFlow Lite is designed for speed and efficiency, there can still be performance trade-offs compared to running models on more powerful hardware. Finally, TensorFlow Lite models can be relatively large, which could become an issue if storage space on our Pi 4 becomes limited.

3.2.1.3 PyTorch

PyTorch is another machine learning object detection model that works a little bit different than TensorFlow and its subsidiary. PyTorch uses dynamic computation graphs, which allow for more flexibility in model design and debugging. Furthermore, PyTorch has a large and active community of users and contributors, which results in a wealth of resources, tutorials, and third-party libraries. Additionally, PyTorch's interface is very Pythonic and intuitive, making it easy for developers who are already familiar with Python to get started quickly. The team has already decided that Python will be the development language for the ML model, so this is a plus.

On the other hand, PyTorch also lacks in some key areas. While PyTorch is suitable for prototyping and research, it may not be as optimized for production-level deployment as some other deep learning frameworks like TensorFlow or TensorFlow Lite. Additionally, deploying PyTorch models in production can be more challenging than in TensorFlow, as it requires additional steps to convert the model into a production-ready format. Furthermore, although it's Pythonic and relatively easy to get started with, it may have a steeper learning curve for beginners in deep learning compared to more high-level tools like Keras. This is important because presently, nobody on the team has any experience developing or using a deep learning object detection model. Finally, PyTorch is primarily designed for traditional CPU and GPU architectures, and its support for mobile and embedded devices can be limited. This could end up limiting performance on an embedded device like the Raspberry Pi 4.

3.2.1.4 Keras

Keras is known for its user-friendly, high-level API, which makes it easy to build, train, and deploy neural networks, especially for those new to deep learning. As previously discussed, the team lacks experience in this development field, so this could be advantageous. Furthermore, Keras can be used with multiple backends, including TensorFlow or TensorFlow Lite. This flexibility allows you to leverage the power of TensorFlow as the backend while benefiting from Keras' simplified interface. Finally, Keras, particularly when using TensorFlow as a backend, is optimized to work efficiently on the PC, taking advantage of its CPU and GPU capabilities.

Unfortunately, ease of development typically sacrifices customization. Keras, being a high-level API, can be less flexible for building highly customized or unconventional neural network architectures. This likely wouldn't be an issue, as recognizing hand gestures should be relatively simple. Additionally, Keras abstracts many details of model construction, which is an advantage for simplicity but a disadvantage when you need fine-grained control over model components. Finally, though a TensorFlow Lite backend could prove advantageous, Keras models require a backend framework, which can further lead to less flexibility.

3.2.2 Machine Learning Object Detection Model Selection

Since we switched to a powerful PC architecture, we are not processor limited. As a result, we chose to opt for the most powerful Machine Learning model available. TensorFlow, with a Keras backend, will allow us to have extremely accurate gesture detection.

3.3 Microcontroller (MCU)

For our senior design project, we have identified that the core of our system will be a microcontroller, serving as the central processing unit. The selection of an appropriate microcontroller is crucial, as it must align with the unique requirements of our project. This microcontroller will operate a laser projection unit and interface with a computer to process images, generate sound, and provide user notifications about system settings. We aim to find a microcontroller that combines versatility and ease of use within our project's timeframe.

We are using a 24 VDC 10A power supply unit (PSU) to power our system, chosen for its ability to supply consistent and reliable power to multiple system components. This choice ensures we have sufficient power to operate the existing components and the flexibility to incorporate additional elements without power constraints. The microcontroller's task includes controlling the laser projection unit and managing the data to be displayed, such as system settings and graphical user interface (GUI) elements. It must have enough pins to connect the laser projection unit and communicate with the computer via a UART connection, underscoring the need for a microcontroller with extensive connectivity options.

As we progress with our project, making well-informed decisions about the microcontroller's technical specifications is imperative. Critical factors include the CPU speed and memory capacity, which are essential for meeting the project's real-time processing and control requirements. The CPU speed should be sufficient to handle the demands of real-time data processing and control tasks efficiently. Meanwhile, the memory capacity must be large enough to accommodate the program code, data storage, and buffer requirements, facilitating smooth and continuous system operation. We will finalize the exact CPU speed and memory needs through detailed assessment and exhaustive testing, ensuring the chosen microcontroller meets the sophisticated demands of our software and data processing activities.

3.3.1 MCU Options

In this section, we will lay out six initial microcontroller options. We will list the advantages and disadvantages of each, ultimately eliminating three. The remaining three microcontrollers will proceed to a more extensive comparison.

3.3.1.1 MSP430FR6989IPZ

Texas Instruments developed the MSP430FR6989IPZ microcontroller as a versatile and robust embedded system. It features a 16-bit RISC architecture and can reach speeds of up to 16MHz, including 128 KB of flash memory and 2 KB of RAM. The MSP430FR6989IPZ offers 83 I/O pins, facilitating easy integration with external devices, and supports multiple communication interfaces such as UART, SPI, I2C, and USB, providing our group with numerous connectivity options for external devices. Including eight analog-to-digital channels will enable precise measurements, while four pulse-width modulation channels will allow for the control of analog devices and motor speed regulation.

A significant advantage of this microcontroller is its FRAM technology, offering fast write speeds and low power consumption in non-volatile memory. This microcontroller exhibits extremely low power consumption, requiring only 0.9 μA in standby mode, and supports a real-time clock, making it ideal for power-efficient applications. It also ensures security through hardware encryption and decryption accelerators, memory protection units, and access control mechanisms. Its LQFP package type adds flexibility to the system design.

However, this microcontroller has limitations, such as its 2 KB RAM, which may restrict data processing capabilities, particularly with large data volumes from the camera. Moreover, its 16 MHz clock speed could affect the project's responsiveness to gestures, which is crucial for achieving accurate and swift responses in music-related applications. Considering the advantages and limitations, the MSP430FR6989IPZ remains viable for projects requiring flexibility and low power consumption. The team's familiarity with this product line has also influenced its selection for further detailed comparisons.

3.3.1.2 MSP430G2553

Texas Instruments developed the MSP430G2553 microcontroller as a compact and cost-effective embedded system. It features a 16-bit reduced instruction set computer architecture and operates at a maximum speed of 16 MHz. With 16 KB of flash memory and 512 bytes of RAM, it provides adequate storage for code and data for small-scale projects. However, its limited 16 I/O pins constrain connectivity options for interfacing with external devices. A significant advantage of the MSP430G2553 is its ultra-low power consumption, approximately 0.1 μA, and multiple low-power modes, which is crucial given the high power demand of our laser projection system and additional external devices from the 24 VDC 10A power supply unit. The MSP430G2553 supports various communication interfaces, including UART, SPI, and I2C, but lacks USB connectivity, which would have benefitted our project. Its integrated analog-to-digital converter with eight channels facilitates analog measurements for sensors or other analog inputs. The microcontroller's compact packaging, available in 28-TSSOP, suits the project's size and weight constraints.

However, the MSP430G2553's flash memory of only 16 KB limits the code's complexity, and the minimal RAM of 512 bytes poses challenges for handling large data sets. The constrained number of I/O pins, 20 in total, requires meticulous planning for connecting external devices like cameras and laser projection units. Despite its cost-effectiveness and power efficiency, the MSP430G2553's limited computing power and memory capacity may compromise the performance of our visually controllable synthesizer, which demands fast and accurate real-time computing. Therefore, due to its inferior performance compared to the MSP430FR6989IPZ, we have decided to exclude the MSP430G2553 from further comparison.

3.3.1.3 ATmega328P

Microchip Technology developed the ATmega328P microcontroller. It is a commonly used and versatile embedded system. It has an 8-bit AVR architecture and a max clock speed of 20 MHz. The microcontroller offers 32 KB of flash memory and 2 KB SRAM, providing ample space for our project. With only 23 I/O pins, it has a good balance of connections for interfacing with external devices via I/O pins. A popular reason that this microcontroller is chosen for projects is because it is so popular. This will mean ample support and guides to use it to its full potential while also allowing for help with troubleshooting. For example, it will have ample libraries, tutorials, and resources to make our time in development easier and offer a wide range of compatibility with development tools and software environments. As with the previous two microcontrollers, the ATmega3280 also excels in low power consumption, making it suitable for 24 VDC 10A PSU. We could consider using a power bank with how little power this microcontroller could use. While in its lowest power mode, we can significantly extend its battery life with only a few microamps. We have multiple communication interfaces such as UART, SPI, and I2C when looking at built-in peripheral features. It has no USB, which is a negative because it limits our options, but it has an integrateed analog-to-digital converter with multiple channels for sensor readings. It supports pulse-width modulation for the precise control of analog devices.

The most significant limitation of the ATmega328P microcontroller will be that it is an 8-bit microcontroller. This will be a problem regarding our project needing real-time motion capturing with fast and accurate responses. Another specification that doesn’t help its case for our project would be the 32 KB of flash memory. I believe that might be the bare minimum we would need for our project, but we would like to have more so we have some leeway. While considering all of this, the ATmega328P is a widely supported and reliable microcontroller that would make programming easier if we choose it. Still, unfortunately, the memory and 8-bit issues might hold back our project when it comes to handling large amounts of data. With all of that being said, the group has chosen to include this product in more extensive comparison.

3.3.1.4 ESP32-S3-S3

The ESP32-S3-S3 microcontroller from Espressif Systems, designed specifically for IoT applications, features a 32-bit dual-core processor capable of reaching speeds up to 240 MHz. This makes it an excellent choice for projects needing real-time gesture analysis. Its standout attributes include dual-mode Bluetooth and Bluetooth Low Energy (BLE) capabilities, facilitating seamless connectivity with smartphones, computers, and IoT devices. Moreover, it supports BLE 5.0, enhancing data transfer rates, security, and power efficiency.

The microcontroller's integrated Wi-Fi supports 802.11 b/g/n standards, allowing straightforward integration with Wi-Fi networks and cloud services, which enables remote device monitoring and control, thus enhancing system automation. With 384 KB of ROM and 512 KB of SRAM, the ESP32-S3-S3 offers ample memory for complex applications. Its comprehensive peripheral interfaces, including I2C, SPI, UART, and GPIO, provide various options for connecting external devices. Additional features include an analog-to-digital converter, pulse-width modulation, and a USB interface, which aids in programming and debugging and facilitates Over-the-Air updates for remote firmware improvements and troubleshooting.

The primary challenge with the ESP32-S3-S3 is its power consumption, especially when using Wi-Fi and BLE features, which could impact our project's power management. However, this can be addressed with a more robust power bank to ensure adequate energy supply. The ESP32-S3-S3's extensive capabilities, substantial memory, and advanced connectivity make it an up-and-coming candidate for our project, warranting further investigation.

3.3.1.5 CY8C6347BZI-BLD53

The CY8C6347BZI-BLD53, from Cypress Semiconductor's PSoC 6 BLE series, integrates a high-performance Arm Cortex-M4 core with an energy-efficient Arm Cortex-M0+ core. This dual-core configuration provides the requisite capabilities for robust processing and energy-conserving operations. With 1 MB of flash memory and 288 KB of SRAM, it offers substantial storage capacity for the extensive programming demands of our initiative. This microcontroller's array of communication interfaces, encompassing UART, SPI, I2C, and USB, ensures comprehensive connectivity with peripheral devices. It also boasts Bluetooth functionality, including an integrated radio with BLE 5.0 support, rendering it apt for wireless interactions with BLE-compatible devices, thus aligning with requirements for wearable technology and intelligent home automation systems.

The device's hallmark is its adaptability, facilitated by Cypress's proprietary programmable system-on-chip technology, allowing precise configuration of analog and digital peripherals. This adaptability not only augments design and programming versatility but also supports the integration of a diverse range of sensors and peripherals. The microcontroller's multiple low-power modes are conducive to battery-dependent systems, optimizing the longevity of portable devices' battery life. The microcontroller is fortified with protective measures, including a hardware-based cryptographic acceleration engine, a genuine random number generator, and mechanisms for secure boot and firmware updates, reinforcing our project's security framework.

Nevertheless, the principal challenge posed by the CY8C6347BZI-BLD53 is its intricacy. The steep learning curve associated with its advanced features and the necessity for a comprehensive comprehension of the PSoC architecture may impede the rapid acquisition of proficiency, a critical factor given the time constraints of a semester-long project. Moreover, the relative obscurity of Cypress products translates to diminished community support and a need for more educational resources, such as tutorials and guides. Considering these aspects, especially the hurdles posed by its complexity and the lack of extensive community support, our team has resolved to exclude the CY8C6347BZI-BLD53 from further evaluation.

3.3.1.6 R5F565NCDDFC#30

The R5F565NCDDFC#30, part of Renesas Electronics' RL78/G1D series, is engineered for applications prioritizing energy efficiency and robust processing capabilities. It utilizes a 32-bit RXv2 CPU core capable of reaching speeds up to 120 MHz and comes equipped with 1.5 MB of flash memory and 640 KB of RAM, offering considerable storage capacity. Its outstanding power efficiency, achieved through various power-saving modes and techniques, is a primary benefit. This efficiency makes the microcontroller suitable for portable devices that demand strong processing power while conserving battery life. The microcontroller's comprehensive range of peripheral interfaces, such as UART, SPI, I2C, and GPIO, ensures seamless integration with external devices. Additionally, it includes analog-to-digital converters and pulse-width modulation for effective analog device management.

The microcontroller also boasts robust security features, including hardware-based encryption/decryption and a unique ID for enhanced data protection and secure communication. Renesas Electronics supports this with a well-established ecosystem, offering extensive development tools, software libraries, and technical assistance, easing development and troubleshooting processes. Nonetheless, the R5F565NCDDFC#30's fixed memory capacity is a notable drawback, as it cannot expand, potentially constraining future enhancements. Its complex architecture necessitates a thorough understanding to unlock its full potential, and its power demands could pose challenges to the system's overall energy management. In light of these considerations, despite its positive attributes, the fixed memory limitation and architectural complexity have led our team to decide against further consideration of the R5F565NCDDFC#30, especially when weighed against other microcontrollers with more flexible memory options and less demanding operational complexities.

3.3.2 Extensive MCU Comparison

After completing a high-level overview of our product options, three original six microcontroller options were selected for further comparison. Those products are the MSP430FR6989IPZ, ATmega328P, and ESP32-S3-S3. Below is a table laying out the relevant technical specifications for these three microcontrollers.

Table 8 - Extensive Microcontroller Comparison Specification Table

Microcontroller	MSP430FR6989IPZ	ATmega328P	ESP32-S3-S3
Cost	$10.77	$2.89	$1.85

Microcontroller	MSP430FR6989IPZ	ATmega328P	ESP32-S3-S3
Low-Power Mode	Yes	Yes	Yes
GPIO Pins	83	23	45
Clock Speed	16 MHz	20 MHz	240 MHz
Architecture	16-bit RISC	8-bit	32-bit
Operating Voltage	1.8V ~ 3.6V	2.7V ~ 5.5V	3V ~ 3.6V
Operating Temperature	-40 ~ 85ºC	-40 ~ 125ºC	-40 ~ 85ºC
ROM Size	128KB	32KB	384KB
RAM Size	2K x 8	2KB	512KB
SPI	4	3	4
UART	2	2	3
I2C	2	2	2
USB	No	No	1
Wi-Fi	No	No	Yes; 802.11 b/g/n Protocol
Bluetooth	No	No	Yes; Bluetooth 5.0

3.3.3 MCU Selection

Our team has thoroughly researched and evaluated various microcontrollers, including the R5F565NCDDFC#30, CY8C6347BZI-BLD53, ATmega328P, MSP430G2553, MSP430FR6989IPZ, and ESP32-S3-S3. After careful consideration, we unanimously decided that the ESP32-S3-S3 microcontroller best fits our project, given its superior performance, connectivity features, and other critical factors.

The ESP32-S3-S3's performance is noteworthy, thanks to its Xtensa LX7 CPU core, which operates at a high clock speed, outperforming its competitors in processing power. This capability ensures efficient handling of complex algorithms and tasks, which is essential for our project's success. Its integrated Wi-Fi and Bluetooth features offer advanced connectivity options, reducing the need for external components and cutting costs. The ESP32-S3-S3's memory capacity is a significant advantage, offering ample storage for code, data, and any advanced computations our project might require. This generous memory allocation guarantees we can fulfill our project's demands without compromising performance. Regarding peripherals, the ESP32-S3-S3 excels, providing a wide range of interfaces and features, allowing easy integration with various sensors, displays, and communication modules. This adaptability enables us to tailor our project to specific requirements, ensuring broad compatibility and enhanced functionality.

The ESP32-S3-S3 also benefits from strong community support, featuring many resources such as detailed documentation, tutorials, and example projects. This support network streamlines our development process, aids in quick issue resolution, and promotes innovation. Unique features of the ESP32-S3-S3, like camera integration, gesture recognition, and laser projection control, align closely with our project's needs, offering specialized capabilities that are particularly relevant to our objectives.

In conclusion, the ESP32-S3-S3 microcontroller is the clear choice for our project due to its exceptional processing capabilities, comprehensive connectivity options, substantial memory, versatile peripheral support, and robust community resources. The ESP32-S3-S3 will enable us to create a sophisticated, efficient, high-quality solution that meets and exceeds our project's requirements and expectations.

3.4 Serial Communication Technologies

To facilitate effective data exchange, we must establish a communication link between the ESP32-S3-S3 microcontroller and the computer. The ESP32-S3-S3 microcontroller will actively receive information from the computer and control the laser projection unit. Concurrently, the computer will process the camera's data and manage the audio output system. An appropriate communication protocol is essential for seamless interaction between the ESP32-S3-S3 board and the computer, ensuring smooth and efficient system operations.

3.4.1 I2C

The first protocol we will examine is the Inter-integrated Circuit (I2C), which features a two-wire interface comprising a data line (SDA) and a clock line (SCL). These lines enable bidirectional communication among devices connected to the I2C bus. The protocol supports multiple devices with a unique address, allowing for direct communication between specific devices. For our project, to facilitate communication between the ESP32-S3-S3 Microcontroller Unit (MCU) and a computer, we propose using an I2C interface, possibly through a USB to an I2C converter, to connect with the computer. This setup will permit communication via the I2C protocol. We will configure the ESP32-S3-S3 MCU as the master device to initiate and manage the communication process. At the same time, we will set up the computer as the slave device to respond to commands and transmit data upon request. We will assign a unique I2C address to each device on the bus to ensure streamlined communication.

Using I2C offers advantages due to its simplicity, user-friendliness, and ability to support multiple device configurations. It is instrumental when the number of available pins is limited, as it requires fewer pins than other protocols. However, a significant limitation of the I2C protocol is its relatively slow data transfer rates, prompting the need to consider alternative communication protocols for applications requiring fast data transmission.

3.4.2 UART

The Universal Asynchronous Receiver-Transmitter (UART) protocol facilitates the connection between the ESP32-S3-S3 microcontroller and the computer. This protocol operates through a wired interface, sequentially transmitting and receiving data bit by bit across two lines: transmit (TX) and receive (RX). Our configuration employs a single USB-to-UART cable, effectively merging the TX and RX lines and thus streamlining the connection between the microcontroller and the computer.

We will configure the microcontroller and the computer to function in UART mode and align their baud rates to ensure synchronized communication speeds. UART's primary advantages include its straightforwardness, widespread availability, support for asynchronous communication, sufficient data transfer speed, and adaptability, making it an excellent choice for interfacing with various external devices. Nonetheless, as we design and implement our project, we must account for its limitations, such as the restricted speed, point-to-point communication nature, lack of inherent error detection and correction, and constrained flow control.

3.4.3 SPI

The Serial Peripheral Interface (SPI) is a synchronous serial communication protocol specifically tailored for short-distance communication between the ESP32-S3-S3 microcontroller and the computer. The setup process entails identifying the SPI pins on both devices, verifying voltage compatibility, and connecting the SCK (Serial Clock), MOSI (Master Out, Slave In), MISO (Master In, Slave Out), and CS (Chip Select) pins in a precise manner. SPI's attributes, including its capability for high-speed data transfer, full duplex communication, and support for multiple slave devices, render it exceptionally apt for projects that demand rapid and dependable data transmission.

3.4.4 Serial Communication Protocol Conclusion

In summary, a reliable communication link between the ESP32-S3-S3 microcontroller and the computer is paramount for effective data and control signal exchange in our project. We assessed three serial communication protocols: I2C, UART, and SPI, noting their respective advantages and drawbacks. I2C stands out for its simplicity and support for multiple devices, yet its slower data transfer rate is a significant limitation. SPI excels in high-speed data transfer and full duplex communication but demands more complex implementation.

We selected UART as the communication protocol for this project due to its optimal balance between simplicity, reliability, and sufficient data transfer speed. The single USB-to-UART cable connection simplifies the setup, providing a straightforward and reliable data exchange pathway that aligns with the project's requirements for easy implementation and consistent performance. Despite UART's speed and point-to-point communication limitations, its broad availability, straightforward configuration, and adequate data transfer capacity render it the best choice for our project's needs and anticipated future expansion.

Table 9 - Serial Communication Comparison

Feature	SPI	UART	I2C
Mode of Operation	Full-duplex	Asynchronous	Half-duplex
Number of Devices	Multiple (1 master, multiple slaves)	Point-to-point (1-to-1)	Multiple (multi-master, multi-slave)

Feature	SPI	UART	I2C
Speed	High	Moderate	Moderate
Complexity	Moderate	Low (simple hardware)	Moderate
Connection	4 wires (MOSI, MISO, SCK, SS)	2 wires (RX, TX)	2 wires (SDA, SCL)
Data Transmission	Synchronous (clocked)	Asynchronous (no clock)	Synchronous (clocked)
Distance	Short (onboard communication)	Longer distances (serial ports, RS-232, RS-485)	Short (typically within a PCB)
Error Handling	Limited	Built-in	Limited
Use Case Examples	High-speed peripherals like SD cards, displays	General serial communication, debugging, low-speed peripherals	Sensors, EEPROMs, low-speed peripherals

3.5 Camera

The project has a crucial component needed in the camera. It will act as the transition component needed for image capturing, processing, and optical tuning. In this section, we will be discussing the various products we have researched to find the most suitable camera that will work seemingly easily when in between the SBC and variable focus lens. It will serve as the primary imaging component that we will use to capture any visual data from the user and transmit it to the processing unit. We split the research into two different types of cameras that each contain three different options. The first three will primarily focus upon the MIPI CSI camera, while the other three will focus upon the USB 2.0 cameras. The MIPI CSI camera stands for the Mobile Industry Processor Interface Camera Serial Interface, as it is a great option to help with our project. The MIPI CSI camera will allow for high-speed transfer of data between the capturing point and the processing point for real-time analysis. The USB 2.0 cameras, on the other hand, offer a more cost-effective camera that will allow for easier compatibility with the hardware configuration. As we discuss later on, though cost-efficiency is a crucial component within this project, keeping a high-quality camera will best suit our project scope in the end.

3.5.1 Camera Options

Here we have six different camera options that will be described throughout. We will begin by dissecting the strengths and weaknesses of each of the cameras listed below and then following up by a comparison table between the various types.

3.5.1.1 Raspberry Pi Camera Module v3

First on the list of viable products, are the three different types of MIPI CSI Cameras. These cameras offer quite different approaches to image capturing needed for this project. The Raspberry Pi Camera Board v3 is an image sensor that was specifically made for the use in collaboration with a Raspberry Pi. It is a compact camera device that is equipped with a high-quality sensor in the Sony IMX708. Its resolution offers an astounding 12 megapixels for capturing image, implying that it contains twelve million pixels within. It comes with a fixed focus lens whose field of view is a 66 x 41-degree view. It is a very versatile product as it will be used in image capturing in various lighting conditions. Its dimensions are 25mm by 24mm by 11.5mm and weighs 4 grams. Its image sensor is 7.4mm diagonally and has a focal length of 4.74mm. The downside to this option lies in its compatibility when it comes to the SBC. This camera module is heavily geared towards being used in cohesion with a raspberry pi. Unfortunately, it can create a headache of sorts if a raspberry pi is not selected as a complementary piece with this, as additional modifications would be necessary for the camera module to function with other types of SBCs. Another downside would be in its field of view and its optical zoom, however, these likely would be negated with the use of the variable focus lens.

3.5.1.2 Arducam IMX477 Camera Module

The Arducam IMX477 camera module is another great option from the MIPI CSI camera list as it provides an alternative to other versions of the camera. The camera module itself comes equipped with an image sensor Sony IMX477, which is an excellent choice for image quality. The resolution on this option manages to increase the megapixels to 12.3 megapixels, equivalent to 4056 x 3040 pixels. This would provide an even more enhanced option in image capturing. It comes with a CS-mount lens of 6mm (about 0.24 in) which can cause it to be very flexible in its customization options, however, as we detail it later, it can potentially cause problems. The camera module would connect with a SBC through the MIPI CSI-2 interface which will allow for a transition-eased transfer between the SBC and camera. Its compatibility is where it begins to thrive as it is more flexible in its interface compatibility. It would be able to connect with Raspberry Pi's, NVIDIA Jetson and other host devices. Unfortunately, it comes with its downsides as cost and its need for manual focus can cause this option to dwindle in its stature. Due to the fact that we will be building a housing property for the variable focus lens, it can overly complicate matters when calibration and testing would be done as we would essentially have to be calibrating two different zoom lens systems. Though this camera module would provide quite the enhanced image capturing qualities, its overall cost can cause a strain in the project when it comes down to budgeting necessities.

3.5.1.3 Waveshare IMX219 Camera Module

The last of the options from the MIPI CSI cameras is the Waveshare IMX219-160 camera module. This camera module, like the previous ones, uses the same image sensor in the Sony IMX219 sensor, seemingly, as they are a great sensor offering excellent opportunities in image capturing. Its resolution is 8 megapixels with a resolution equivalent to 3280 x 2464 pixels. Its interface would still connect to a SBC through the MIPI CSI-2 interface. The Waveshare camera module is primarily focused on being compatible with a raspberry pi board and its various models. It comes with the ability to be flexible in its mount lens, as it can allow for modifications on the m12 mount lens. It contains a field of view of 160 degrees diagonally, and has a focal length of 3.15mm, while its dimensions are 25mm x 24 mm. Its only real downside we would say comes in the fact that it has a fixed-focus lens, however again, this isn't necessarily a downside when it comes to the scope of our project. This is due to the variable focus lens housing.

3.5.1.4 Logitech C270 HD Webcam

We now transition to the three viable products of a USB 2.0 camera. First off is the Logitech C270 HD Webcam. The Logitech C270 HD Webcam is a USB 2.0 camera in which it connects to the SBC by the use of a USB-A port. It features plug-and-play type compatibility which would simplify the way the integration process would go. Considering how the MIPI CSI cameras work, this integration wouldn't require complex configurations. Due to the versatile ways this webcam works, it would provide a seamless easy connectivity to the SBC. It has an HD resolution of 720p and a frame rate of 30 frames per second. Its dimensions are 2.9 x 1.6 x 2.8 inches. It is a very cost-efficient webcam that is easy to set up and is very user-friendly on installation. The diagonal field of view for this webcam is 60 degrees, and it contains a microphone, however, it would not be necessary in our project's scope. Overall, it is a fairly simple and easy camera that allows for near complete compatibility with nearly every SBC at an affordable price.

3.5.1.5 Logitech C920s Pro HD Webcam

Next on the list of viable USB 2.0 cameras is the C920s Pro HD Webcam. It contains a resolution of 720p to 1080p, which would offer a very detailed image capturing necessary for gesture recognition. Its camera is equipped with a 3-megapixel resolution and has an autofocus feature for the focus type. The webcam is equipped with a high-quality sensor that would also contribute to the overall image quality. The field of view for this webcam is 78 degrees diagonally, allowing for great visibility. Its dimensions measure 43.3mm x 94mm x 71mm, while its weight is 162 grams. One great feature of this webcam is that it contains an auto light correction of sorts for it. It is very compatible with various SBCs which helps further enhance the versatility that this webcam offers. Like the previous option, this webcam connects to the SBC via a USB-A port. Overall, this is another very cost-efficient webcam that we can consider.

3.5.1.6 Logitech 1080p Pro Stream Webcam C922

One of the last options comes in the Logitech 1080p Pro Stream Webcam. Though it is considered a USB 2.0 camera, it offers quite impressive specifications. It has a resolution of 1080p, allowing for detailing image capturing, while keeping a frame rate of 30 frames per second. It also allows for resolution of 720p while holding a 60 frames per second if we were to choose to decide on the latter during the design process. The webcam offers automatic adjustments such as adjustments to low light which would help with capturing the hand gesture in certain conditions where lighting appears to be faint. The screen size for this webcam is 68.58 and contains a diagonal field of view of 78 degrees. The camera maintains a 3-megapixel resolution, with an autofocus on its focus type. Its dimensions are 44mm x 95mm x 71mm and weighs 162 grams. Arguably what sets this camera apart from the other USB 2.0 cameras is that one of our members is willing to donate his to the project, which would help in the overall budget allocation.

3.5.2 Camera Comparison

For this section, we will go into detail comparing the various options we have discussed so far, to then dwindle that amount to the final option we will ultimately select for our project. All these cameras have their own strengths and weaknesses, but we must decide on one that will best fit our project's scope at the best budgetary reasoning.

Table 10 - Camera Specifications

Camera:	Raspberry Pi Camera Module v3	Arducam IMX477 Camera Module	Logitech 1080p Pro Stream Webcam C922
Camera Type	MIPI	MIPI	USB 2.0
Resolution	4608 x 2592	4056 x 3040	1920 x 1080
Sensor	12Mpixels	12.3Mpixels	3Mpixels
FOV (diagonally)	75	78	78
Focal Length	4.74	6	Autofocus
Compatibility	MIPI CSI-2 Interface	MIPI CSI-2 Interface	All with USB A Port
Price	$25	$75.99	$99.99
Manufacturer	Adafruit	Uctronics	Logitech

3.5.3 Camera Selection

From the various types of cameras we have been discussing in this section, we have decided upon the Raspberry Pi Camera Module v3. The key factor in deciding upon this specific camera when weighing the options of the other cameras was primarily in its connectivity with the single board computer. We have already selected the Raspberry Pi 4 as the single board computer, so we found it best to keep compatibility as a key point in our selection. As well, this camera helps fit within our specified budget of $400, as finding components at cost-efficient pricing is a great determinant into our selections. This camera module also features quite impressive specifications regarding its resolution, sensor and field of view. When mentioning its resolution, this camera module has an impressive resolution of 4608 x 2592 pixels, while having a sensor of 12 megapixels. Like we said it is extremely compatible with the raspberry pi 4 in terms of connecting via the mipi csi-2 interface and tops it off with a field of view of 66 degrees by 41 degrees, equivalent to 78 degrees diagonally. Overall, this camera will serve a great purpose into which we can properly integrate our software side of our project with the optical design of the lens system later on.

3.5.4 Camera Revision Change

During the testing phase of our project, it came to light that the camera selection with the raspberry pi 4 was scrapped in favor of direct connection towards the laptop via usb connectivity. Therefore, we began to research possible remedies to fixing this newcoming problem, by researching usb connectivity cameras and selecting the best one. However, we understand that usb cameras have a small sensor unfortunately, which can prevent us from providing enough sufficient light to reach our sensor. To fix this, we found a solution in the raspberry pi high quality camera, that would need an adapter to function directly to the laptop.

Table 11 – Camera Revision Selection

Camera:	**Raspberry Pi HQ Camera**	**Arducam 1080p Usb Camera**
Camera Type:	CSI	USB 2.0
Resolution:	4056 x 3040	1945 x 1109
Sensor:	12.3Mpixels	2Mpixels
FOV:	Needs Lens	160
Focal Length:	Needs Lens	6
Optical Size:	1/2.3"	1/2.8"
Price:	$50.00	$50.00

3.6 Lens Imaging System

The following sections will act as a preliminary research section for our lens imaging system. Though many lenses will be thoroughly looked at, ultimately only a handful will be selected to help complement the selected camera option mentioned earlier. Since our main objectives are to enhance the depth of field and widen the field of view in both the horizontal and vertical direction. It is imperative that the research done on the lens system is structured in a way that helps obtain those objectives. To begin with, we will be discussing the lens system followed by breaking it down into the depth of field research and the wide-angle field of view research.

3.6.1 Lens System

As we progress on our project, we will begin to investigate the optical design of the lens imaging system. This will act as an enhancement to the overall performance of our project. The requirement we set upon ourselves is the precise capturing of hand gestures in a range up to 5 feet. Though our main priority is properly capturing and analyzing hand gestures at a rapid pace, issues can generally arise. These issues primarily stem from the notion that if the user were to move around whether it be closer or farther, that the camera could have issues capturing the gesture in a clear and concise way. Also, if the user were to move around not only in the sense of forward and back, but rather left or right, the camera system would need to readjust for this change. In this case, we will dwell into the notion of capturing hand gestures within a certain angle cone of view and within a certain range distance. The lens imaging system will be placed between the user and the camera, allowing it to be a middle ground of sorts. Seamlessly helping to integrate with the camera to allow for a more optimized image capture. In the following sections, we will begin to briefly research into the various ways we can introduce to help with improving upon image quality. This will lay the foundation in which later on, we will further discuss this in our design sections with a more thorough examination.

3.6.2 Depth of Field

For this section of the project, we will be focusing on depth of field and its optical impact upon the lens system. Throughout our research, we will be investigating ways into attempting to enhance image capturing. Through careful research and selection of lenses, we will be tailing our depth of field to implement improved image quality. As well, we hope that we have a seamless integration with other components that have been thoroughly researched and selected by other members of the team. Depth of field in this project will allow for a range in which captured images maintain their clarity seen within a far and close proximity of the camera without significant quality loss. Our requirement for this will be up to 5 feet, so employing a system in which our depth of field can enhance our quality of the image being captured within this 5-foot range is essential in gesture recognition at a qualitative fast rate. We established this as one of our requirements for our project's scope. As detailed earlier, our depth of field research will aid in the prevention of hand gesture blur when the user was to move closer or farther in distance. Blur in this case would hamper the ability for the software to analyze the hand gesture and produce an accurate output. In our lens imaging system, we must understand the important consideration we must have on depth of field and its contributions towards our hand gesture recognition. Depth of field is the range in which captured images would maintain their clarity without any major droppage in quality. We face a challenge in attempting to capture hand gestures at varying distances from our camera in the optical axis. We will explore the different methodology types known for maintaining sharpness throughout different depths. One of the first ways of affecting the depth of field within an optical system would be in the distance between the subject and the camera itself. In our case, we are attempting to expand the depth of field, essentially extending our range of focus. This would be done by increasing the distance between the subject and the camera. The second type of method in affecting the depth of field would be in introducing an aperture of sorts to act as a way to essentially modify the way in which light rays would travel within the optical axis in its path towards the sensor of the camera and the user itself. Depending on where we place the aperture in our optical system, it would play a major factor in whether the depth of field would be fuller rather than shallow, as well as the size of the aperture itself. We understand the importance of researching the proper balance in which we can fully expand our depth of field to encapsulate the subject and will ensure that the one we ultimately decide upon for our design portion will properly align with our depth of field objective.

3.6.3 Wide-Angle Field of View

Wide-angle field of view is the next area in which we will be discussing. Attempting to broaden the field of view of our project plays a key role in our capturing process. This section will help with further understanding how the lenses we will be designing will be designed with the intention of broadening our field of view. The two types of lenses that we will be researching for this will be the plano-convex and bi-convex lenses. Both of these lenses would play a fundamental role in helping control the in-coming light rays to provide a broader view. Bi-convex lenses work by having curved spherical surfaces on both sides of the lens, which would be tremendous in helping control the path of light and allowing a broader view. It will further allow for the efficient convergence of the light rays which will widen our range in which to capture the hand gestures. The plano-convex lens works in a similar fashion, however, instead of two curved surfaces, it only has one curved surface, while the other side surface is flat in its orientation. Further analysis of the optical characteristics of these lenses will allow us to determine which lens type is most suitable for our wide-angle field of view goal. The research we do into the wide-angle field of view will help be the ground basis in which we dive deeper in our design and prototyping phases. With advanced software like Zemax, we will be able to get a more detailed schematic of how the lens system will operate under certain lenses. Zemax will allow us to run tests and simulations in which we can better understand how these types of lenses work in certain conditions and how they improve upon the wider field of view. Since it is imperative that we attempt to enhance the hand gesture capturing in not only varying distances but at varying degrees. It is worth mentioning that the potential integration of compounded lenses can be a possibility. This would require the combination of multiple lens elements. It could potentially introduce a more complex lens system design, but it can be an option depending upon how design works. The design portion of this optical setup will provide a detailed understanding that there could be possible and potential challenges that we must address such as aberrations.

3.7 Laser Projection Unit

The GUI of our synthesizer is going to be one of the standout features of the project. Foregoing the use of a GUI built into a program run on a computer, we will instead use a laser system to project the settings of our synthesizer onto a surface near the user. The projection will have to be dynamic, as the user will be changing settings at their leisure and will therefore need to be quick with how it can change the display. The laser will also have to be powerful enough to be read in a room with high background lighting, as the gesture detection portion of the synthesizer would be rendered unusable since the camera would not be able to pick up signals in a low light environment.

Another consideration is the size of the Laser Projection Unit (LPU). One of our goals is to reduce the size and weight of the synthesizer so that it may be portable. This will influence the number of optical elements that can be included in the LPU, and how many lines of text that can be created from the display. The quality of the text, along with the font size of the text will be affected by this.

The cost also comes into question, as opting for cheaper components could lead to lower quality results, such as the laser: choosing cheap laser diodes not only affects the power of the beam, which results in lower intensity and dimmer output, but could also affect the beam shape and quality of the actual electronics involved in making the device lase. However, if we need to increase the amount of lasers in the LPU to fit more lines of text, then the more expensive lasers leave the discussion, as the cost to build, and the instability of the project itself could lead to higher expenses than expected.

Beam quality is important for our display as it affects many qualities of the final output; if the waist of the beam is too large, then the resolution takes a hit, and the brightness indirectly. Too dim of a beam, and as previously stated, the user would have a hard time reading the actual text. Improperly aligned components could drastically affect the output, as the beam could become misguided within the LPU.

3.7.1 Guiding the Laser

Laser guidance is done mainly by the rotation and translation of mirrors within a system. Research began with analyzing scanning systems that used a laser, and how they guide said laser. The guiding system were more than likely be made from several parts, namely, a laser diode, a lens system that would clean and focus the laser beam, and two mirrors that would rotate in two separate directions.

3.7.2 Galvo Laser System

Galvo lasers get their namesake from galvanometer, a device which was invented for use in measuring current. A coil is wound around a permanent magnet, with each end connected to a current source. In a conventional galvanometer used for measuring current, one end of the coil has a torsion spring with a pointer attached to the end of it, which is unwound when the coil rotates from the magnetic field generated from the provided current.

With the discovery of lasers and the idea of directing the beam using mirrors, devices have been created that utilize the rotating effect of the coil for guiding the beam path. Instead of a pointer being attached to the coil, a mirror set upon a stage is used, and the current is controlled to allow for rotation of the mirror. This allowed for variable angle control of the laser path. In conjunction with another galvo mirror rotated 90°, one can achieve control in both the x and y direction.

For use in creating a GUI with a laser, this type of system would be useful in allowing for a simple, programmable beam guidance system. Constantly changing the supplied current to the galvo mirrors creates changes the rotation of the mirrors rapidly, which can be used in making a rasterized laser display. Modulating the signal applied to the laser can be used to change when the laser is on and off, which can be programmed to create text and images with the laser.

In the following sections we will be listing out the three galvo mirrors we considered for the project, what their strengths and weaknesses are, and

3.7.2.1 Scanlab Galvo dynAXIS L

- 5A maximum RMS current
- 15A peak current
- 0.85Ω coil resistance
- 25° maximum scan angle
- Approximately 400g weight
- <1μrad repeatability

A lower coil resistance allows lower voltage requirements across the galvo, and the tight repeatability spec allows for cleaner details in the result. The drawbacks come from the high weight, which goes against our plan for a more compact and light design, and the low scan angle, which reduces the size of the display at desirable distances.

3.7.2.2 VantagePro SS30Y-AG

- 4A maximum RMS current
- 10A peak current
- 5.8Ω coil resistance
- 40° maximum scan angle
- 74g weight
- <10μrad repeatability

Along with the lower weight, this model also has one of the higher scan angles found on the market. The lower max current reduces the amount of precision available for guiding the beam, and the repeatability is a little high, but finer details are secondary to the weight and scan angle this model allows for.

3.7.2.3 Cambridge 6230 Galvo Mirror

- 7.1A maximum RMS current
- 25A peak current
- 1.07Ω coil resistance
- 40° maximum scan angle
- 267g weight
- 8μrad repeatability

A high max current, decent coil resistance, great scan angle, and an alright weight and repeatability spec make this model an all-around great pick for what we need in this project.

3.7.2.4 Extensive Galvo Mirror Comparison

Two of the three galvo mirrors listed previously will have an in-depth comparison in this section. The VantagePro model was promising, but with the only price we could find being $3500, this put the mirror out of the question. As for the other models, noted specs will be the maximum RMS current, peak current, coil resistance, maximum scan angle, weight, and the repeatability of each mirror.

3.7.2.4.1 Maximum RMS and Peak Current

The current matters for how much power the galvo mirror will take up, and how much variability we have for the actual coil movement. If the maximum current is higher, we can adjust the position of the mirror by smaller increments. This is a minor benefit, but something to think of, nonetheless.

3.7.2.4.2 Coil Resistance

Same thing as the current, this just makes the voltage required for the galvo mirror higher, since the voltage directly correlates with the current going across the mechanism. One thing to note, is that the Cambridge model has the highest peak current and has a comparable resistance to the Scanlab model.

3.7.2.4.3 Maximum Scan Angle

The culmination of the previous two specs appears in the scan angle. The scan angle is probably the most important part of the galvo mirror. Of note, the Scanlab model has a smaller scan angle than the other model, altogether removing this choice from consideration. If our goal of the LPU is to project out GUI at most a foot in front of the whole mechanism, then any reduction in the scan angle could amount to inches of lost display area. This leads into the reason why the Scanlab model has the lowest resistance, while also having a relatively small maximum current, since at the highest current, the galvo mirror will be at the maximum angle.

3.7.2.4.4 Weight

One portion of one of the requirements of the project is to weigh less than 10lbs, meaning any saving on weight would be nice. Choosing the Scanlab model would mean allocating almost a pound of our weight budget to the mirror.

3.7.2.4.5 Repeatability

A minor spec, as all it only applies to is the clearness of our display. The small variations will not be too noticeable to the naked eye, even for the finer details.

3.7.2.4.6 Price

The true decider for what mirror was picked for the project. Not only does the Cambridge model have a larger scan angle and lower weight, along with having a higher maximum current, it is also cheaper than the Scanlab models we could find, being over half the price. With a requirement of being around $400, taking the Scanlab model would mean using up almost two-thirds of our budget.

3.7.2.4.7 Galvo Mirror Conclusion

Table *12* - Table Galvo Mirror Comparison

	Scanlab Galvo dynAXIS L	Cambridge 6230 Galvo Mirror
Maximum RMS Current	5A	7.1A
Peak Current	15A	25A
Coil Resistance	0.85Ω	1.07Ω
Maximum Scan Angle	25°	40°
Weight	400g	267g
Repeatability	<1µrad	8µrad
Price	$249.00	$111.86

With the combination of the specs listed in the previous sections, the smartest choice of the two is the Cambridge model. The price, weight, and scan angle all make up for the repeatability difference and coil resistance that the Scanlab model has over it.

Although the use of the galvo mirror was a very promising idea, after further investigation, the pinout codes for the galvo could never be deciphered, and without these, we would have no true control over the mirror, leading us to brainstorm other ideas for the mirror.

3.7.3 Motor for Mirror Attachment

To counteract the above problem, we decided to create our own improvision of a galvo mirror. The important part was for the mirror to rotate in along one axis, so a DC motor was the perfect base for this idea. In addition, we already had a controller for the motor, an L298N motor controller, meaning we did not need to purchase one and wait for shipping. The motor controller would allow for precise speed and position control and would enable the motor to be connected to the MCU where it can be programmatically controlled.

There were many aspects to consider when choosing the right motor for the project. As with the previous components, voltage and current would control what power supplies we would realistically have access to. The length of the shaft upon which the mirror would be attached would control how long the housing for the whole LPU would need to be to fit every component in and still be able to guide the laser. Weight was of another concern, and with the motor being on one side of the housing, the other side would need to be balanced out so the entirety of the LPU would not tip over from its own weight. Speed would determine just how quickly the mirror could spin, thus providing faster response when it comes to updating the GUI.

Four motors were considered for the project: The PPN7PA12C1 from NMB Technologies Corporation, the ROB-11696 from SparkFun Electronics, the 711 model from Adafruit Industries, and the NEMA 17 stepper motor from Sorand.

3.7.3.1 PPN7PA12C1

- 5VDC rated voltage
- 1-7VDC voltage range
- 1.3A starting current
- 270mA rated load current at 5VDC
- 15.6mm casing diameter
- 31.8mm total length
- 9.5mm shaft length
- 10g mass
- 11600rpm rated load speed

With the highest speed among the three, this motor would allow for flexibility in how fast the mirror can change orientation. This model also has the lowest weight, meaning less of a balancing act when it comes to the final housing of the LPU. One drawback, however, is the higher amperage and voltage required by the motor, which restricts what power supplies we have to choose from slightly. Compared to the other motors, however, there was more documentation and specifications provided from the manufacturer, which would allow us to plan ahead for what we need rather than having to receive the motor and test it, which the others models lack.

3.7.3.2 ROB-11696

- 12VDC max operating voltage
- 1-3VDC normal operating voltage
- 110mA operating current at 1VDC

- 26g mass
- 6600rpm rated load speed

This model has the lowest operating voltage of the three, and also had a lower rated current than the first model listed here. The upsides end there though, as this model is also heaviest of the three, and has the lowest speed. This model also had a gear attached to the shaft, which may be removable, but if not, then purchasing it could turn into a headache of removing the gear. There was also limited documentation on the model, as the datasheet was a wasteland when it came to providing any information that DigiKey did not already provide, which was unwell for planning our design of the LPU.

3.7.3.3 711

- 6VDC rated voltage
- 4.5-9VDC voltage range
- 20mm casing diameter
- 8mm shaft and bearing length
- 17.7g mass
- 9100rpm rated load speed

This model is a compromise between the two previous models. Beyond this, the only thing that stands out is the price, being the lowest between the three. Same as the second model, however, the documentation was nonexistent, and somehow even worse than the already lacking documentation of the second model.

3.7.3.4 NEMA 17 Stepper Motor

- 24VDC rated voltage
- 4.2cm x 4.2cm x 4cm casing dimensions
- 2.2cm shaft length
- 1.5A rated current per phase
- 1.8° step angle
- 600 rpm

Although this may look like the worst of the motors, the fact that it is a stepper motor and can be controlled using a motor controller, and has libraries within the coding languages we will be using, this makes it much easier to control than the other motors, and allows for more feedback to our MCU.

3.7.3.5 Motor Conclusion

Table 13 - LPU Motor Comparison Table

	PPN7PA12C1	ROB-11696	711	NEMA 17
Rated Voltage	5VDC	12VDC	6VDC	24VDC
Voltage Range	1-7VDC	1-3VDC	4.5-9VDC	-
Load Current	270mA @5VDC	110mA @1VDC	-	1.5A per phase
Dimensions	D = 15.6mm	-	D = 20mm	4.2cm x 4.2cm x 4cm
Total Length	31.8mm	-	28mm	6.4cm
Shaft Length	9.5mm	-	<8mm	2.2cm
Mass	10g	26g	17.7g	-
Load Speed	11600rpm	6600rpm	9100rpm	600rpm
Price	$3.34	$2.10	$1.95	$16.71

Although the fourth mirror is the most expensive the models, it is the only one that can be easily controlled without making the coding portion of the LPU a headache. The power it requires is not an issue given that our power supply is rated at 24 V and 10A output. For these reasons, it trumps any of the motors choices that were considered and tested.

3.7.4 Rotating Polygonal Mirror

Next up is the mirror that will be scanning in the horizontal, or x-axis, of the GUI. The idea we found most useful was to use a spinning polygonal mirror commonly found in laser printers. When a voltage is applied, the mirror starts spinning. Combined with the galvo mirror from the previous section, this would cover the two axes of the display we need for the LPU.

Only one model was found that would be available for use at a decent price: the mirror found in a Brother MFC-8220 printer. The only spec we could find was that the voltage required for the motor is 24V.

3.7.5 Laser Diodes

A galvo laser system would not be complete without a laser, and for this project, we would like to use something that is compact and safe for human eyes when not in direct contact. For this, laser diodes offer the perfect solution.

3.7.5.1 HL6312G Laser Diode

- 5mW optical power
- 40mA operating current
- 2V operating voltage
- 625-640nm wavelength, 635nm typical
- 8° parallel beam divergence
- 31° perpendicular beam divergence
- Price of $24.45

This model holds the middle ground between the other two laser diodes when it comes to optical power, but has the highest operating current.

3.7.5.2 L635P5 Laser Diode

- 6mW optical power
- 30mA operating current
- 2V operating voltage
- 635-643nm wavelength
- 8° parallel beam divergence
- 32° perpendicular beam divergence
- Price of $27.13

Almost similar to the previous model, with a slightly lower operating current and higher optical power, which could prove to be more dangerous when it comes to accidental exposure to the eye. Given all of this, the higher price is not really justifiable for the minimal increase in efficiency.

3.7.5.3 Instapark DRM104-D003

- 1.8-2.8mW optical power
- 20-40mA operating current
- 3V operating voltage

- 640-660nm wavelength, 650nm typical
- 10mm beam spot size at 15m
- Price of $4.79

Although this diode has a higher operating current, and operates at a wavelength higher than the normal red wavelength, the price is 20% than the other two diodes.

3.7.5.4 Extensive Laser Diode Comparison

Table *14* - Laser Diode Comparison

	HL6312G Laser Diode	L635P5 Laser Diode	Instapark DRM104-D003	Tegg Mini Laser Diode
Optical Power	5mW	6mW	1.8-2.8mW	5mW
Operating Voltage	2V	2V	3V	3.3V
Operating Current	40mA	30mA	20-40mA	20-40mA
Wavelength	625-640nm, 635nm typical	635-643nm	640-660nm, 650 typical	650nm
Beam Divergence	8° parallel, 31° perpendicular	8° parallel, 32° perpendicular	N/A	N/A
Spot Size	N/A	N/A	10mm at 15m	N/A
Price	$24.45	$27.13	$4.79	$6.99 for 10

The lower power of this laser diode works in our favor, as it lowers the risk of eye damage if the laser were to be exposed to the naked eye. The beam spot size is something that can be cleaned up later with lenses if need be. For our project, due to safety concerns, and the fact that the laser diodes from Thorlabs requires the use of a TEC controller, which adds at least another $100 in price, we will be using the Tegg model.

3.7.6 Photodiode

To synchronize the two mirrors and generate the interrupts that will be used to create the signal for the laser diode, which is used to create the GUI, we will be using a photodiode that will sit in the path of laser beam. Characteristics that we prioritized were as follows: photosensitivity, how reactive the semiconductor material is to light hitting it; short circuit current, how much current runs through the photodiode at maximum intensity; dark current, the amount of current that runs through it when there is no light incident on the detector; photosensitive area size, how large in diameter or the dimensions of the detector are; operating temperature, what temperatures the photodiode can be used at; and price.

3.7.6.1 Hamamatsu S2387-33R

- 0.37A/W @ 633nm photosensitivity
- 5.8μA short-circuit current
- 5pA dark current
- 2.4mm by 2.4mm photosensitive area
- -20°C to 60°C operating temperature
- $49.95 price

3.7.6.2 Perkin Elmer C30902E

- 77A/W @ 830nm photosensitivity
- 5mA short-circuit current
- 15nA dark current
- 0.5mm diameter photosensitive area
- -40°C to 70°C operating temperature
- $58.00 price

Having the highest photosensitivity and tied for the highest short circuit current, this model would pick up any amount of light that is entering the housing of the LPU and create a current. This is good for picking up a signal easier, but also means any background light would also trigger the photodiode, which would not be good for creating the GUI we need.

3.7.6.3 Hamamatsu S5973

- 0.44A/W @ 660nm photosensitivity
- 90nA short-circuit current
- 1pA dark current
- 0.4mm diameter photosensitive area

- -40°C to 100°C operating temperature
- $24.95 price

The smaller photosensitive area means the beam will most likely cover the all of it, which leads to pulling out as much current as possible. The lower short circuit current and photosensitivity enables the LPU to be more selective with the light it detects, while also being subject to a lower signal to noise ratio, which could be a problem.

3.7.6.4 Excelitas C30817

- 40A/W @ 633nm photosensitivity
- 5mA short-circuit current
- 50nA dark current
- 0.8mm diameter photosensitive area
- -40°C to 70°C operating temperature
- $50.00 price

As with the Perkin Elmer model from earlier, the high photosensitivity is nice for ease of creating a signal, but background light could affect the synchronization of the mirrors.

3.7.6.5 OSI Optoelectronics UV-015

- 0.4A/W @ 633nm photosensitivity
- 100µA short-circuit current
- 3.05mm by 3.81mm photosensitive area
- -20°C to 60°C operating temperature
- $49.95 price

This model has a pretty large photosensitive area, which could mean some lost signal when the laser hits it.

3.7.6.6 Extensive Photodiode Comparison

Going forward, we will only compare the two Hamamatsu models and the OSI Optoelectronics model.

Table *15* – Photodiode Comparison

	Hamamatsu S2387-33R	Hamamatsu S5973	OSI Optoelectronics UV-015
Photosensitivity	0.37A/W @ 633nm	0.44A/W @ 660nm	0.4A/W @ 633nm
Short-circuit Current	5.8µA	90nA	100µA
Dark Current	5pA	1pA	-
Photosensitive Area	2.4mm by 2.4mm	0.4mm diameter	3.05mm by 3.81mm
Operating Temperature	-20°C to 60°C	-40°C to 100°C	-20°C to 60°C
Price	$49.95	$24.95	$49.95

Going forward, we will be using the Hamamatsu S5973 model, as it has the smallest size, a decent photosensitivity, and is cheaper than the other two models.

3.8 Power

In our project to develop a gesture-controlled synthesizer with an integrated laser projection unit, selecting a 24 VDC 10A power supply unit (PSU) plays a crucial role. This PSU supplies energy to each component, from the motors driving the mirrors to the microcontrollers managing the system. However, this choice of PSU brings a unique set of pros and cons that we must carefully consider and address through our system design and embedded programming.

The primary advantage of a 24 VDC 10A PSU is its high power capacity. This capacity ensures that our system receives sufficient power for optimal performance, particularly the power-intensive components like the motors controlling the polygon mirror and the laser unit. This is crucial in a system where precise movements and timing are integral to functionality. Another benefit is the efficiency that comes with such a PSU. High efficiency translates to more reliable power delivery to our system components, enhancing overall performance and reducing the risk of power-related failures. Moreover, the scalability offered by this PSU cannot be understated. As our project evolves, additional components or functionalities might be added. A PSU with a higher power capacity than is currently necessary provides the headroom for these future enhancements, making our system future-proof.

Despite these advantages, the 24 VDC 10A PSU presents specific challenges. The most significant of these is the need for voltage regulation. Many components in our system, including the ESP32-S3 microcontrollers and various sensors, operate at much lower voltages, typically 3.3V or 5V. To address this, our design includes a series of voltage regulators and step-down converters. These components will be crucial in transforming the 24V output of the PSU to the required lower voltages. In selecting these regulators and converters, we focus on efficiency, precision, and the ability to handle the necessary current load without overheating or introducing noise into the system. Managing heat is another challenge. Stepping down from a higher voltage to lower voltages like 3.3V or 5V can generate significant heat, especially when the current requirements are high. For our system design, we will have to keep increases in temperature in mind and try to find an efficient way to incorporate them. These cooling solutions are essential to dissipate heat effectively, maintaining the optimal operating temperature for the system components. The larger size and higher cost of the 24 VDC 10A PSU compared to lower-capacity alternatives are also factors we have considered. While the larger size impacts the overall form factor of our synthesizer, the trade-off is justified given the PSU's power capacity and efficiency. The PSU's scalability and reliability mitigate the cost factor, reducing the need for future upgrades or replacements.

Addressing the varied voltage requirements of our system's components is crucial to our project. For components that operate at 3.3V, such as the ESP32-S3 microcontrollers, we are using low-dropout (LDO) voltage regulators. These regulators are ideal for their stability and efficiency, especially in applications where the input voltage is close to the output voltage. For components requiring 5V, we are implementing buck converters. These converters are efficient for stepping down from higher voltages to 5V, and their switching nature makes them better suited for applications with a significant voltage drop. Our power distribution strategy involves placing these voltage regulators and converters close to their respective components. This minimizes voltage drop and electromagnetic interference, ensuring stable operation. We are also incorporating decoupling capacitors at the input and output of these regulators to smooth out transient voltage spikes and dips. Safety is paramount in our power management design. We include protective features such as overcurrent protection, short-circuit protection, and thermal protection circuits. These features are crucial for preventing electrical anomalies that could damage the system or pose a safety risk.

In conclusion, while presenting particular challenges, our choice to use a 24 VDC 10A PSU in our synthesizer project offers significant power capacity and system efficiency advantages. Our approach to overcoming the challenges involves carefully planning and implementing a sophisticated power distribution system. This system will ensure all components receive the correct voltage levels, integrating safety mechanisms and heat management solutions. Our goal is to create a synthesizer that meets our current needs and is prepared for future expansions and enhancements. This endeavor underscores our commitment to innovation, technical excellence, and creating a user-friendly, interactive musical instrument.

4. Design Constraints and Standards

This section describes the various constraints the team faces in the development of the gesture-controlled synthesizer, as well as the applicable standards for different areas of development. The purpose of the constraints subsections is to shed light on the factors and limitations that influence our scheduling and component selection processes. As well as helping to ensure that the final product is a product that is accessible to all members of the community and functions within the established parameters that each project must follow. Following this, we delve into the subsections dedicated to establishing the framework for the array of standards and regulations essential for our team's informed decision-making in the development and selection of various elements for the gesture-controlled synthesizer.

4.1 Constraints

Within this project, our team is expected to be aware of the constraints that must be acknowledged so that we can properly design our project and incorporate a plan involving said constraints. Some of the constraints that we have considered throughout our initial research will be further explored in this section. This section will heavily revolve around the notion of explaining the numerous concerns that will ultimately play a divisive part in our overall component selection and design implementation for the user. Establishing our constraints will help with optimizing the user experience of our product once it is ready for consumption.

4.1.1 Economic Constraints

The primary goal of this project was to create a design in which music enjoyers would have the opportunity to modify their musician settings at an affordable rate. Our budget was set to be under $400. To complete this project, we will have done extensive research into finding the pieces and components necessary to fit within our criteria but ones that would not hinder the overall production of the end product.

Ideally, most projects within the Senior Design course tend to be funded by some sponsor, whether it be a company or even a professor or advisor themselves. These projects receive financial assistance and even insights into what components or pieces are necessary for certain objects. Unfortunately, this does not appear to be the case for our team, so the project will ideally face some limitation on resources. Also, we can face an increase in difficulty in the overall process due to this. However, we will not waver in the fact that collecting extensive research into this project will produce a great enough effort into finding the best supplies at the most reasonable price. Economic constraint does not only affect our supply budget, but we must also keep in mind creating an overall product that is as low-cost production as it is low relative cost to potential consumers. If we were to create a product that was too simply expensive for the public, then most consumers would shy away from even considering our project. There ideally are components that would be of even higher quality offered in the public, but due to our budget allocation, we simply cannot afford to go outside our set prices. An example of this would be in the lens system itself. Though there are many different ways to achieve our outlined objectives, utilizing the combination of various lenses in a series is a more cost-efficient method. Lens mounts are offered on sites such as Thorlabs and Edmund Optics, but these range at an extreme price that would simply put us way over our specified budget. Our budget must account for the fact that there are many major areas of which we carefully research into and we would have to buy components for these areas. Also, there is the potential unforeseen expenses that could play a factor down the line in the project that if not tabulated for, could result in our budget being exceeded. A major cost-effective alternative we have found so far in our project is researching viable components on websites that don't have such a high price set on their products. This greatly demonstrates our commitment to exploring ways in which we could save the cost of our components without degrading our overall product. An example of this is within our projection unit. Our group members were able to find ways to piece apart certain devices to integrate into our project without having to spend a significant price on the components. Since our group does not contain an electrical engineering member, we don't have to burden ourselves with the added complexity of a designed pcb board which could as well result in money being spent on extra components that wouldn't align with our general knowledge.

4.1.2 Time Constraints

A constraint that plays a big factor in this would have to be the time constraint. This is due primarily to the fact that we are all college students attempting to juggle between classes, work, and our own personal lives. We chose a design that could be completed in a timely manner. Beginning our project in the Fall 2023 semester implies we would finish our project in the Spring 2023 semester. We are unfortunately reduced in our time goal, and we have to factor in time allowances with limited time into full designing and prototypes. In an ideal scenario, we would have more time in which we could extend our scope of what our project could ultimately be, but due to deadlines within each semester, we must decide what we can get done within the time we are given. It may appear as if the eight months in which we must get our project working in full capacity is a lot, time does not give us the luxury we would ideally like. There will be times in which one of the members or even more will be required to attend to other matters simply because we are college students that do not have the luxury of primarily focusing on just this project alone. Potentially one of the team members may have to worry about working on certain days, internships requiring a team member to be away at certain hours, and even other classes taking time away in which could be valuably used on the project itself. Another time constraint that we must take into consideration is the time it would ideally take for the image processing to correctly recognize gestures. For our project to be practical in its use, the duration in which the gesture takes to be properly recognized should be minimized. This would help ultimately enhance the appeal of our project in consumers' eyes. We must also factor in the fact that not all portions of the project would ideally be done within a certain window. Due to the fact that we have two optics members and two programming members working on different sections of the project, sometimes one may require the other. There can possibly be times in which one member may have to wait for the other member to finish their portion of whether it be research, design or testing before they can properly get what they need done. Ultimately, it is up to the responsibility of the group to finish working on their sections of the project in a reasonable time to help mitigate the potential loss of time prior to submitting the report.

4.1.3 Environmental Constraints

Due to our product involving pieces of technological hardware, we understand the importance of attempting to minimize the environmental impact it may have throughout all stages of the product's lifecycle. We want to ensure that the product does not cause any harm to any living beings and to the environment surrounding it. Our product is mainly meant for indoor usage, so there is not a necessary requirement in terms of waste in outdoor environments, but we will at least keep it in mind. We will emphasize the importance of careful resource management to help reduce the environmental footprint as much as possible. Though it is expected that our product will have a lengthy lifecycle, we will research ways to possibly reduce the waste generation that could occur. As we are preparing to commence the prototype and testing phases of our project, we will keep in mind the significance of material usage and are committed to minimizing the overall consumption of materials that we will use throughout our next phases.

4.1.4 Social Constraints

In our attempts to create a product that is as user-friendly as possible, our project will incorporate some sense of a lookup table of sorts. This lookup table will help facilitate the way hand signals will be recognized and then accurately implemented. We understand that designing this lookup table will help sort of bridge the way. It will also help promote digital literacy among any consumer. Our product will put a heavy emphasis on allowing musicians of any ability to use this product without any difficult issues arising. The eased accessibility of this product will help make the whole overall user experience rather inclusive to help prevent potential consumers from shying away. If we were to promote inclusivity for all users, this would help not only with word of mouth on our product, but even improve general musician awareness. The lookup table will essentially allow for users to see how certain hand gestures lead to certain changes in the resulting gesture. Our lookup table will utilize many similar gestures that are commonly used in asl. This will help our product cater to audiences that are from many different backgrounds and be very inclusive to those with different communication styles as well. Thus, the asl-like gestures will allow for a more seamless and enjoyable experience for all users that attempt to use the product.

4.1.5 Political Constraints

At the current moment, there are no political constraints that are limiting the design of this project. It is effectively our responsibility to ensure that any change within regulations and policies within governments are subject to be corrected in any instance. It is also up to our group to be kept up to date with any engineering standard that could potentially change while we are working on our project.

4.1.6 Ethical Constraints

When investigating the ethical constraints of our project, we must note the various types. Initially, we notice that creating a secure product is essentially a must for consumers in today's market. Implementing ways to ensure that the user's data is protected from outside threats is not considered a priority on our list of constraints, but rather a requirement. We can provide this by implementing safety measures in which data is properly stored and secured. It would then be properly processed with the user's consent and used in a way that is adhering to the purpose of the product. Being transparent would also fall under the ethical constraint section, due to it being a foundation in which we build trust and maintain it with the users. It is done by being as clear as possible in informing users of the data that is being collected and eventually utilized. In addition to this, we are incorporating an interface that would be used by the users. It is crucial to create an interface that is as user-friendly as possible to help with minimizing any difficult interactions. We can improve upon this by ensuring that the interface is simple for a user to interact regardless of their technological knowledge or musical ability. We understand that our user community varies in different types of abilities, thus we are fixated on designing the interface to be accessible to all users. Testing our product constantly through the use of design and prototyping will be essential in gathering feedback to help ease the user experience so that it will be an eased and enjoyable interaction. By viewing these ethical constraints, we can commit ourselves into creating an overall product that would satisfy our users in overall enjoyment with full trust and security.

4.1.7 Health and Safety Constraints

The health and safety of every user utilizing this product is an important factor not only for this project, but for every project in general. We, as engineers, are required to follow every guideline that is appropriate when designing a project. To guarantee user safety, our design will involve the incorporation of minimizing the potential risk of injury. Our project will be involving the use of a laser diode in the laser projection unit. To help adhere with laser safety standards, we have properly conducted extensive research in determining what type of selected laser would not be potentially harmful to the user. There are many different laser diodes offered throughout the market, but it is up to us as engineers to ensure that no potential harm could happen due to the utilization of which laser diode, we end up utilizing for our projection unit. It is important as well to address our full commitment to ensuring that no physical harm can be done to any structure of our product. We will completely ensure that the user experience when using this product is safe and comfortable no matter their physical condition. It will be up to us as well to give a user guidance manual to help with facilitating what the guidelines are for safe usage. This will help with clarifying the correct usage of the product to help avoid potential risks. By adhering to the health and safety of our potential consumers of the product, we commit ourselves to meeting these safety regulations but also monitoring user well-being and overall enjoyment of the product.

4.1.8 Manufacturability Constraints

Manufacturability is an interesting topic when thinking about this project. We must primarily focus on the idea of finding pieces and components that would be viable for our project at an accessible and reasonable price. In our time frame in which we have two semesters to research, design and fully build the product for final testing, we aren't afforded the luxury of being able to. There is a lot that has to be done to fully integrate each major portion of the project. For instance, our software team is currently conducting research into which microcontroller and single board computer would fit best in the parameters we have set upon ourselves for this project. Thus, we are researching on which pieces we want to buy, but we must factor in whether the pieces that we want are even available in the window in which we must purchase and would ideally fit within the preset budget to allow allocation to other areas on the project. For our optics side of the project, we must research which types of lenses, sensors and laser diodes would be best suited for our project and if there are even in stock. Though it may not sound like there is much to purchase for this project at times, the minor components will eventually add up and there will be times where an additional piece will be necessary to help successfully integrate the major components together. Sometimes certain components will not be available for purchase in moments where we would like them to be, so it is imperative that we accelerate our processes in determining the suitable components and buying them ahead of time to prevent any delay. It is our understanding that if our product was deemed suitable for mass production once the terms have finished, the supplier would be able to ship out a high number of units to supply customer demand. As we have detailed in previous sections, there are many factors to consider when looking into the manufacturing process. Finding the best quality components which are available and fit within the budget parameter is essential for our project to succeed in all cycles of its life.

4.1.9 Sustainability Constraints

Sustainability plays a crucial role when it comes down to longevity of the product. We want this project to not only function properly, but for it to also have a lasting impact on any consumer that would use it. In this day and age, technology is very prevalent in being outdated constantly due to technological advances. We are attempting to design this product with the notion that this product has to not only survive in terms of initial but as well into the future. Durability is an important consideration into sustainability as designing this project will require using certain components that we will either purchase or design based on thorough research. Ultimately tasked with making sure that the product will not break down or even fail at all times. This would go hand in hand with our earlier discussions of other constraints as they all tie up into the longevity of our product based on the components of our budget to quality ratio. Our project will be integrating a 120v ac 60hz power, type B wall plug. It is in our best interest that we understand the importance of energy efficiency. As well it is important to understand that power must be easy to obtain for our product to help with portability to other areas. Knowing that it is essential for our project to run smoothly without any major hindrances. Having our project have efficient energy would further enhance the way we capture gestures and output everything in a seemingly eased way.

4.2 Standards

This section discusses the various development standards and regulations our team is being held to in the development of the gesture-controlled synthesizer. Standards are important within many different aspects of life. Within the scope of this project, adhering to the following standards will ensure that the technology within our device will work as designed. In our case, an example would be wireless communication between our synthesizer and a speaker. Standardization of communication protocols allows for the two technologies to interact successfully, and without interference from other nearby devices. Thus, adherence to that standard is crucial, as is the same for the other standards discussed below.

4.2.1 Bluetooth 5.0 Standards

Bluetooth 5.0 is the fifth major version of the Bluetooth wireless communication standard, which was officially released in late 2016. It introduced several significant improvements and new features over its predecessor. Originally, Bluetooth specifications were created in a joint effort to unify short-range radio technology by Intel, Ericsson, and Nokia.

The Bluetooth connection process consists of three primary states. First, inquiry, where two Bluetooth devices without previous pairing history must try to discover each other. By default, unpaired devices sit in a waiting state, so when one device sends out an inquiry for discovery, any unpaired device can immediately respond with information. Once the devices have discovered each other, Paging can begin. Paging is the formal connection process to connect two Bluetooth devices. Finally, after the devices have been successfully connected, they can individually be placed into one of four modes; active mode, sniff mode, hold mode, or park mode.

Bluetooth 5.0 operates in the 2.4 GHz ISM (Industrial, Scientific, and Medical) band and offers a maximum data rate of 2 Mbps. Additionally, Bluetooth 5.0 provides an extended range, with up to four times the range of its predecessor. The range can vary depending on environmental factors, but it's typically around 200 meters. Bluetooth 5.0 uses 40 channels, and it employs a modulation scheme, known as Gaussian Frequency-Shift Keying (GFSK). Bluetooth 5.0 also maintains the low energy consumption characteristics of Bluetooth Low Energy (BLE). It's designed to be highly power-efficient, making it suitable for battery-powered and energy-conscious devices. Furthermore, Bluetooth 5.0 is designed to be backward compatible with previous versions of Bluetooth, ensuring that it can work with older Bluetooth devices. Although there are other features available within the Bluetooth 5.0 standard, they are not necessarily relevant to the development of the gesture-controlled synthesizer.

4.2.2 C++ Language Standards

As part of our senior design project, we are developing a gesture-controlled synthesizer by integrating a microcontroller development board and a computer. The development board is the project's core, and we are programming it using C++. We chose C++ for its advanced features and compatibility with embedded systems. The ISO/IEC 14882:2020 standard, also known as C++20, is the governing standard for C++, created through the collaboration of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC).

The C++20 standard ensures the language remains current and effective by incorporating contributions from various organizations worldwide. It defines the syntax, structure, and constraints of C++ programming and establishes the semantic rules for interpreting the code. It also specifies the representation of input and output data in C++ programs and defines the boundaries of a conforming implementation. While the C++20 standard provides comprehensive guidelines for C++ programming, it leaves unspecified aspects, such as the transformation and invocation mechanisms of C++ programs on data-processing systems and the conversion processes for input and output data. These areas, which are particularly relevant to the compilation of C++ code, vary across compilers but must conform to the ISO/IEC 14882 standard.

ISO/IEC 14882's primary goal is to improve the portability of C++ code, allowing developers to create consistent and reliable software across different platforms. This standardization is critical to our project. Unlike a conventional computer setup where code is directly executed, our system requires compiling the code on a computer and then uploading it to the microcontroller development board. The C++20 standard guides this process, ensuring our code is robust, efficient, and compatible with the microcontroller despite the compilation occurring in a different environment.

Therefore, adhering to the ISO/IEC 14882 standard is crucial to successfully developing our gesture-controlled synthesizer. It simplifies the computer programming process and ensures that the microcontroller development board will accurately execute the code, aligning with our project's technical and operational objectives.

4.2.3 Python Language Standards

All of the code written for execution on the PC within the gesture-controlled synthesizer will be written using Python. With that being said, it’s important to delve into the Python Enhancement Proposals (PEPs) created by the Python development team as guidelines for implementing Python code and proposing changes to the language. Although there are various PEP categories and they focus on different areas of the language, our group's focus is on PEP 8 – Style Guide for Python Code. The PEP 8 documentation lays out specific conventions for Python code, with a focus on the Python standard library. This is a living document, as it changes with updates to the Python language.

Excluding introductory and appendix sections, PEP 8 contains seven sections 90discussing conventions for various areas of programming. These sections include "Code Lay-out", "String Quotes", "Whitespace in Expressions and Statements", "When to Use Trailing Commas", "Comments", "Naming Conventions", and "Programming Recommendations" [3]. Within each section, there are one to many subsections detailing specific guidelines. For example, the Code Lay-out section has "Indentation", "Tabs or Spaces?", "Maximum Line Length", "Should a Line Break Before or After a Binary Operator?", "Blank Lines", "Source File Encoding", "Imports", and "Module Level Dunder Names" [3] as sub-sections.

To summarize, PEP 8 provides important guidelines for both code portability and readability. Nobody within this team has previous experience with Python, and multiple members will have to share development of a single codebase. Referencing this standard will allow individual members to write code that everyone can understand, and provide confidence that code written on a host machine will run without fault on the PC.

4.2.4 IPC PCB Standards

As part of the Senior Design development process, our group is required to create a printed circuit board (PCB) to house a portion of our components, including the MCU. With that, it's important to research and understand any relevant standards relating to the creation of PCBs. The Institute for Printed Circuits (IPC) is a non-profit organization that developed PCB standards for a wide variety of implementations.

At this time, IPC has twenty-two listed standards describing conventions for developing printed circuit boards. Within the scope of this project, there are a few that apply. IPC-2221B details generic standards relating to design of PCBs. IPC-2222B details specific design characteristics of rigid PCBs. IPC-2251 discusses designing for the packaging of high-speed electronics. IPC-2611 goes over required product documentation for electronic products. Furthermore, IPC-2614 lays out requirements for documentation for board fabrication. In regard to integrating the MCU, IPC-7092 discusses design and assembly processes for embedded PCB components. Finally, IPC-7351B discusses requirements for designing in conjunction with surface-mount components.

These standards are important because the board we design must be manufacturable. PCB design involves using a computer-aided-design (CAD) software to import components, connect them, and place them on a PCB. From there, a file can be sent to a printing company where they can print the physical PCB. If our PCB is not designed to manufacturable standards, the printing company may charge more, or outright refuse to print our design.

4.2.5 Laser Safety Standards

A large part of this project involves the use of a low power laser around the 633 wavelength. Although this is not a particularly dangerous laser, there are still risks associated with it. One thing we do not have to worry about is the skin damage, since our laser is in the visible range and is only a few mW in power. However, any amount of eye exposure is dangerous and can cause permanent damage, no matter the power of the laser. Even if the lasers expressed intent is to be pointed towards a surface to create a GUI, accidents can still occur, so knowing the safety standards behind lasers is important in designing the LPU.

The Laser Institute (LIA), in collaboration with the Occupational and Safety Health Administration (OSHA), created the ANSI Z136 series of laser safety standards. There are multiple subsections of the series that are pertinent to our project. Z136.1 – Safe Use of Lasers, being the parent document, lays the foundation for the standards in industry and academia. Since we are doing this project in an educational setting, and utilizing senior design labs found on campus, Z136.5 – Safe Use of Lasers in Educational Institutions is pertinent when following building codes, especially those laid out for the CREOL building. This goes hand-in-hand with Z136.8 – Safe Use of Lasers in Research, Development, or Testing, as the main purpose of the design labs is to test and develop the means for which the laser will be used.

With these standards in mind, testing can go forward on the LPU, and the risk of damage being caused in the future can be mitigated. Properly protecting against uncontrolled laser guidance with beam blocks and PPE in the form of laser safety glasses are some of the methods that will contribute to this reduction.

5. ChatGPT / LLM Discussion

This section will provide us with a look at how ChatGPT and other language models could help us during our project. For example, with coding or brainstorming, extra features users would want. We will look at five overall models, with two being from

ChatGPT, and see how we did use them or could have used them to help further our project scope.

5.1 LLM Comparison Introduction

As a group of senior engineering students working on our project—a gesture-controlled synthesizer with laser projection using a PC and ESP-32 boards—we have dived into the realm of AI platforms to evaluate how they can aid our endeavor. We are comparing five key AI models: ChatGPT 3.5, ChatGPT 4, IBM Watson, Google AI, and OpenAI Codex. Each offers unique benefits and limitations; understanding these is pivotal for effectively leveraging them in our project.

Starting with ChatGPT 3.5, we found it impressive for its language processing capabilities and rapid responses. It is a boon for brainstorming and understanding broad concepts, particularly helpful in the early design stages of our project. However, its knowledge is capped at 2021, which means it might miss out on the latest advancements in our field. Its inability to interact in real-time and potentially limited depth in specialized tech areas could be a drawback for the more hands-on aspects of our project, like fine-tuning the interaction between the PC and ESP-32.

Moving to ChatGPT 4, this model is a step up with its enhanced understanding of complex queries and a broader knowledge base, making it more suitable for tackling sophisticated aspects of our project. It is convenient when dealing with complex algorithms or integration challenges. Nevertheless, like 3.5, it is constrained by its training cut-off and cannot provide real-time assistance, which could be a hiccup when dealing with the most current tech solutions or hands-on debugging.

When we explored IBM Watson, its standout feature was advanced analytics. Watson can be a game-changer for our project, especially optimizing performance and processing input from the gesture interface. The downside? It is complex to set up and might be cost-prohibitive, which could be better given our student budget constraints and limited time frame.

With its wide range of tools and integration with Google services, Google AI is like a treasure trove. AI could be invaluable for a tech-intensive project like ours, especially for machine-learning aspects related to gesture recognition. However, its complexity and the resources it demands might be challenging for some of our team members, who still need to get up to speed with advanced AI concepts.

Finally, OpenAI Codex caught our attention for its code generation capabilities and integration with coding environments. It is an excellent asset for rapidly developing and testing code for our PC and ESP-32 setup. The catch here is that the code it generates often requires tweaking, and its primary focus is on coding, so its utility is limited for other aspects of our project.

In conclusion, as we navigate our project, each AI model offers distinct advantages. ChatGPT versions are great for theoretical grounding and initial brainstorming, IBM Watson and Google AI are remarkable for their advanced analytics and technical depth, and OpenAI Codex is excellent for coding efficiency. Our task is to strategically use each of these platforms where they fit best, enhancing our project's development and learning experience as budding computer engineers.

5.2 ChatGPT 3.5

In our senior design project, which involves creating a gesture-controlled synthesizer using PC and ESP-32 boards, ChatGPT 3.5 has played a multifaceted role. As a senior Computer Engineering student, we have found its capabilities and limitations intriguing and instructive.

ChatGPT 3.5's key strength lies in its ability to assist with conceptualization and creativity. For instance, when brainstorming how to integrate gesture control with sound synthesis, ChatGPT 3.5 provided us with a foundational understanding of similar technologies. It suggested several relevant algorithms and design patterns, such as machine learning for gesture recognition, which we had yet to consider initially. ChatGPT expanded our project scope and provided a solid theoretical base to build. Another area where ChatGPT 3.5 proved invaluable is in coding assistance. For example, when we struggled to write Python scripts for the PC, it helped debug a persistent error in our code that involved integrating the ESP-32 board. It also offered suggestions on optimizing our code for better performance. Although it could not write the entire codebase, its insights helped us overcome significant hurdles in the development process.

However, ChatGPT 3.5 has limitations, particularly in its knowledge being current only up to 2021. ChatGPT became evident when we explored using a newer version of a gesture recognition library. ChatGPT 3.5 provided outdated information, and we had to cross-verify it with the latest documentation. ChatGPT highlighted the need to stay updated with the latest developments in our field beyond what ChatGPT 3.5 could offer. Moreover, ChatGPT 3.5's inability to interact in real-time posed a challenge during our testing phase. For instance, when we were calibrating the laser projection system, we needed immediate feedback to adjust parameters, something ChatGPT 3.5 could not provide. We had to rely on other tools and our trial-and-error methods for this real-time calibration.

In conclusion, ChatGPT 3.5 has been a significant asset in our project, particularly in ideation, theoretical understanding, and coding assistance. Its limitations, notably in providing up-to-date information and real-time interaction, have taught us to combine AI-driven guidance with our research and hands-on problem-solving skills. This blend of AI assistance and proactive learning has been instrumental in advancing our project.

5.3 ChatGPT 4

In our ambitious endeavor to create a gesture-controlled synthesizer using PC and ESP-32 boards, ChatGPT 4 has emerged as a cornerstone of our project's development. Our experience as a senior Computer Engineering student has allowed us to deeply appreciate the nuanced enhancements that ChatGPT 4 brings over its predecessor, especially in its application to our project.

A key instance where ChatGPT 4's capabilities shone was during our project's initial ideation phase. When brainstorming the user interface design, ChatGPT 4 provided us with advanced insights into user experience best practices and innovative interface options. For example, it suggested integrating haptic feedback for a more immersive user experience, a concept we had yet to consider initially. ChatGPT enriched our design and expanded our understanding of interactive technologies. ChatGPT 4's enhanced language processing skills also proved invaluable in coding. Its ability to understand and explain intricate concepts was a game-changer when working with complex algorithms for sound synthesis and gesture recognition. It helped us optimize our machine-learning model for gesture recognition by suggesting more efficient data processing techniques. This advice significantly improved our model's accuracy and responsiveness, which is crucial for the real-time aspect of our project.

However, like any tool, ChatGPT 4 has its limitations. Despite its extensive knowledge base, we needed more in its understanding of the latest technological advancements. ChatGPT became evident when we were exploring cutting-edge libraries for laser projection. ChatGPT 4 provided foundational information, but we had to rely on current technical papers and developer forums for the most up-to-date methods and practices. Another limitation emerged in real-time testing scenarios. For instance, we needed immediate feedback on system performance when integrating the ESP-32 board with the PC for real-time sound processing. ChatGPT 4's lack of real-time interaction meant we had to use other debugging tools and rely on hands-on experimentation for immediate system feedback and calibration.

In retrospect, ChatGPT 4 has been an invaluable asset, particularly in enhancing our theoretical knowledge and assisting in complex coding challenges. Its limitations, especially in providing the latest technological updates and real-time interaction, served as a reminder of the importance of complementing AI assistance with continual learning and practical experimentation. This combination of cutting-edge AI insights and hands-on engineering has propelled our project forward.

5.4 IBM Watson

In developing our gesture-controlled synthesizer project, which incorporates PC and ESP-32 boards, we have considered the potential role of IBM Watson. Its analytics and AI-driven insights strengths suggest that it could be a powerful addition to our project, especially given its capabilities in handling complex data sets. As a group of senior Engineering students, we see several ways in which IBM Watson could enhance our project despite the specific challenges it may present.

IBM Watson could significantly contribute to data analysis and optimization within our project. For instance, in calibrating the gesture recognition system, Watson's AI analytics could analyze sensor data more efficiently. Its ability to provide deep insights into gesture detection algorithms could enhance the accuracy and responsiveness of our system, which is crucial for ensuring the synthesizer responds accurately to user gestures. Additionally, Watson's capabilities in predictive maintenance could foresee potential system failures or inefficiencies. Based on operational data analysis, it could alert us to issues like overheating in our PC, allowing us to adjust cooling mechanisms and proactively prevent malfunctions.

However, integrating IBM Watson into our project would not be without challenges. Watson's initial setup and configuration to suit our specific needs could be complex, requiring substantial time investment. This factor is a considerable constraint, considering our tight project timeline. Another potential hurdle is the cost associated with IBM Watson's advanced features. Operating within a student budget, we must judiciously select the services we utilize to balance affordability and the need for advanced AI capabilities. Moreover, Watson's primary focus on specialized, enterprise-level solutions might limit its versatility for general queries or broader aspects of our project. We might rely on additional resources for more general or theoretical information outside of Watson's specialized scope.

In conclusion, while we have not utilized IBM Watson in our current project phase, its potential to process and analyze large volumes of data could be immensely beneficial in optimizing our synthesizer's performance. The key to leveraging IBM Watson effectively would be to balance its AI-driven capabilities with the complexities, costs, and specific needs of our project. It represents an intriguing possibility for enhancing our project with advanced AI analytics and predictive abilities.

5.5 Google AI

In the conceptual phase of our gesture-controlled synthesizer project, integrating PC and ESP-32 boards, we have explored the potential benefits of incorporating Google AI. Known for its wide range of tools and advanced capabilities, Google AI could offer significant enhancements to our project. As a senior Computer Engineering student, we recognize the potential of Google AI to expand and refine our project's scope, albeit with certain complexities to navigate.

One area where Google AI could contribute is improving the machine learning aspect of gesture recognition. Utilizing TensorFlow, part of Google's AI suite could enable us to develop a more accurate model for interpreting hand gestures. TensorFlow's advanced libraries allow for fine-tuning neural network parameters, potentially enhancing the synthesizer's responsiveness to user gestures. Such precision in gesture recognition is vital to the seamless functionality of our synthesizer. Google AI could also be instrumental in cloud computing and data storage, mainly through the use of Google Cloud Platform. This would be advantageous for storing and analyzing the large datasets generated during our testing phases. The ability to efficiently manage and process data collected from multiple users would be crucial in refining our gesture recognition algorithms, and the scalability of Google Cloud could support this aspect effectively.

However, integrating Google AI into our project would come with its challenges. The suite's tools, while powerful, have a steep learning curve, particularly for those less acquainted with advanced AI and machine learning concepts. This complexity means that significant time would need to be allocated for training and familiarization, which could impact our project timeline. Additionally, the resource intensity of some Google AI tools could pose a limitation. Training a sophisticated machine learning model, for example, requires substantial computational power. Access to high-performance computing resources, only sometimes readily available, would be a prerequisite for effectively utilizing these tools, potentially impacting our ability to iterate and test quickly.

In conclusion, while Google AI has yet to be utilized in our current project phase, its potential to enhance machine learning capabilities and manage large data sets could significantly advance our synthesizer's development. The key to effectively leveraging Google AI lies in balancing its advanced tools and functionalities with the complexities and resources they demand. This exploration underscores the potential of integrating cutting-edge AI technologies to push the boundaries of our project, presenting a compelling avenue for future development.

5.6 OpenAI Codex

As we navigate through our project to develop a gesture-controlled synthesizer using PC and ESP-32 boards, we recognize that incorporating OpenAI Codex could significantly enhance our coding and rapid prototyping efforts. Our perspective as a group of Engineering students allows us to see how this tool could be effectively leveraged while also being mindful of its limitations.

OpenAI Codex, with its capability to generate code, could be particularly beneficial in streamlining the development of firmware for the ESP-32 and PC. For example, Codex could suggest efficient coding structures and offer solutions for managing Bluetooth connectivity, potentially saving us considerable time. This aspect of Codex would be akin to having an additional team member capable of quickly producing code snippets and offering novel coding approaches that we might not have previously considered. Another area where OpenAI Codex could prove valuable is in troubleshooting and debugging. If we encountered issues, such as a bug in the PC audio processing script, Codex's ability to analyze and suggest fixes could be highly advantageous. This feature would be handy when working under tight project deadlines, providing us with quick insights and potential solutions.

However, we also anticipate challenges with integrating OpenAI Codex. One concern is the accuracy of the code it generates. For instance, in developing algorithms for laser projection, we might find that the initial code provided by Codex requires considerable adjustments to align with our project's specific requirements. This realization underscores the need to review and refine any Codex-generated code thoroughly. Additionally, while OpenAI Codex excels in software-related aspects, it does not offer direct feedback on hardware interaction. In phases of our project where seamless hardware integration is crucial, such as ensuring effective communication between the PC and ESP-32, Codex's utility might be limited. We must complement its use with other tools and hands-on testing to ensure successful hardware-software integration.

In conclusion, while OpenAI Codex has not been a part of our project, its potential contributions to coding and debugging could be immensely beneficial. Its strengths in these areas could accelerate our development process. However, its limitations in code accuracy and lack of hardware interaction feedback highlight the importance of combining AI-assisted coding with manual expertise and practical engineering. This integration could be a critical factor in efficiently advancing our project.

5.7 Language Model Comparison Conclusion

In our analysis of various AI platforms, namely ChatGPT 3.5 and ChatGPT 4, and considering the potential use of IBM Watson, Google AI, and OpenAI Codex for our gesture-controlled synthesizer project, it is clear that each platform offers distinct advantages and disadvantages. This research has provided invaluable insights into the utilization of these AI tools for our project.

ChatGPT 3.5 and ChatGPT 4 were integral to our project. They provided robust theoretical support and coding assistance and fostered creativity and brainstorming. Their advanced natural language processing capabilities facilitated a deeper understanding of complex concepts and enhanced our project documentation. However, their limitations in providing the latest technical knowledge and lack of real-time interaction necessitated supplementing their input with current research and hands-on experimentation.

While we did not use IBM Watson, its potential in analyzing and interpreting large data sets and offering predictive insights could have been valuable in fine-tuning our system's performance. Its ability to optimize algorithms and ensure efficient system operations is noteworthy, though we recognize the challenges it poses in terms of complex setup and cost considerations.

Similarly, with its extensive suite of tools, Google AI could have been a significant resource in our project, particularly in machine learning and cloud computing. It could enable efficient handling and processing of large datasets and offer advanced capabilities for developing gesture recognition algorithms. However, its complexity and high resource demands imply a steep learning curve and the need for high-performance computing resources.

OpenAI Codex, which we also did not use, presents itself as a tool capable of aiding in coding and rapid prototyping, potentially accelerating the development process. While impressive, its code generation and debugging capabilities would require manual review and adjustment, as well as integration with hardware efforts.

Each AI platform brought unique perspectives to our project, whether used or considered. ChatGPT versions directly enhanced our theoretical understanding and coding efforts; IBM Watson, Google AI, and OpenAI Codex, though not utilized, illuminated possibilities for data analytics, machine learning, and software development. This comparison advanced our project and enriched our learning experience, highlighting the importance of selecting appropriate tools for specific tasks and combining AI capabilities with practical engineering and research skills. This project has been a significant step in our journey as budding engineers, demonstrating the profound impact of AI in real-world applications.

Table 14 - ChatGPT Comparison Table

AI platform	Pros	Cons	Impact on Senior Design Project
ChatGPT 3.5	-Strong language processing -Quick responses	-Outdated knowledge -No real-time interaction	-Aids in conceptualization -Useful for coding basics
ChatGPT 4	-Advanced language skills -Broader knowledge base	-Limited current knowledge -No real-time interaction	-Enhanced support for complex concepts -Improved code assistance
IBM Watson	-Advanced analytics -Predictive insights	-Complex setup -Potentially high cost	-Optimizes performance -Useful for data analysis
Google AI	-Wide tool range -Strong in machine learning	-Complex for beginners -Resource-intensive	-Facilitates advanced ML Applications -Effective data management
OpenAI codex	-Efficient code generation -Good for debugging	-Code accuracy Varies -Software-focused	-Accelerates coding process -Aids in rapid prototyping

6. Hardware Design

This section will detail all of our hardware design choices, including electrical, optical, and mechanical. Throughout each individual section topics such as power distribution, lens design, and LPU circuit design and implementation will be discussed.

6.1 Electrical Design

The following subsections will describe the electrical design process our team went through when designing our gesture-controlled synthesizer. At this current moment in our development process, our primary focus has been on identifying and designing the DC/DC converters that will be required.

6.1.1 Introduction

For our gesture-controlled synthesizer, complemented by a laser projection system, we realized we required a DC/DC converter, which will be vital for overall project powering. This converter was explicitly made to step down the 24 VDC 10A PSU to the essential 3.3V and 5V required by our system. Our needs may evolve as the project matures, prompting us to refine or completely revamp this initial converter design to meet the emerging demands of our system.

Thanks to our meetings, Dr. Chan noticed that we would not have to make a complex PCB or even have one since our group has no electrical student; we have streamlined our design process by avoiding the complexities of a printed circuit board (PCB) at this stage. He has confirmed that for the scope of our work, we can proceed without a PCB or, at most, implement a very basic one. This advice aligns perfectly with our team's composition since we are a team of two computer engineering and two optics students, allowing us to not have to get involved with advance of electrical circuits.

The decision to forgo a complex PCB at this phase brings several benefits. It allows us to prototype and iterate rapidly and adapt our design in real time as we test and learn. This approach is invaluable in an educational setting where learning through hands-on experience is often more impactful than theoretical study. Our current DC/DC converter setup, built with standard electronic components on a breadboard or prototype board, reflects our commitment to an agile and experimental development process.

Looking ahead, we fully expect to refine our power system. Whether we transition to a custom PCB or enhance our standalone converter, the knowledge we gain during this initial phase will inform our decisions and designs. Dr. Chan's approach of starting simple has been instrumental, focusing our efforts on the synthesizer's core functionalities while allowing room for growth and improvement. We recognize that our design journey is iterative, and this initial converter is a launchpad for future innovation. The project's exciting phase underscores the importance of adaptability and learning in engineering design.

6.1.2 Power Distribution System

Our power distribution system consists of a 24V 10A DC Power Supply, which interfaces with 120VAC Power using a Type B wall plug. The 24V Power Supply will directly power both mirrors, as well as the step-down circuits. To provide power to our Photodiode Amplification Circuit, we have included a -/+ 5V regulator.

6.1.3 Photodiode Comparator Circuit

The photodiode comparator consists of an LM393 voltage comparator used to send digital data to the MCU when the laser hits the photodiode. A trim potentiometer is used to set the reference voltage on the positive input of the LM393. The negative voltage is determined using a voltage divider, with a 100K resistor and the photodiode in a reversed biased configuration.

Additionally, our PCB also contains a footprint for our stepper motor controller, which was added in an effort to clean up wiring within the device. Below is a 3D model of our PCB Design:

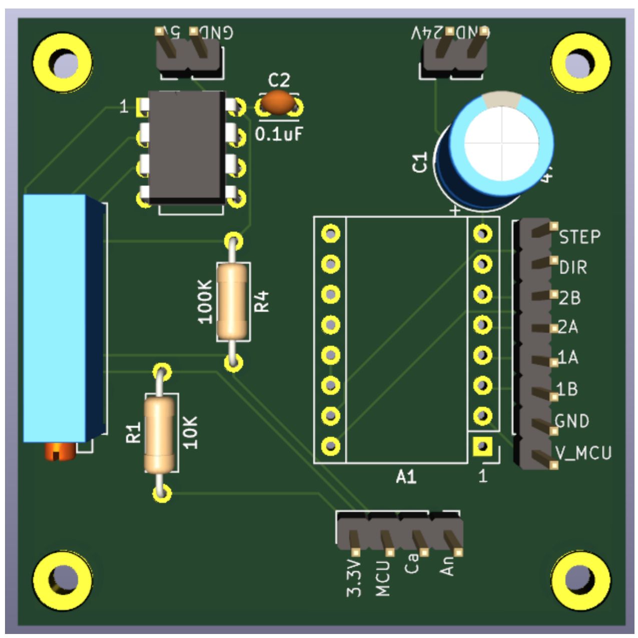


Figure 5 - Photodiode Amplifier PCB

6.2 Optical Design

There are two main optical subsystems to this project: The LPU and the camera with lens system attached. Each system will have its own problems, with the LPU dealing with proper beam guidance and designing the correct electrical system to successfully create the rasterization of the GUI, while the lens system will have to deal with finding the right lens for optimizing the input for the camera.

6.2.1 Laser Projection Unit

The LPU section will start off with discussing how to control the beam path with reflective surfaces, along with the process of figuring out the orientations and distances needed to create our specified display size. Getting the timing right for the polygonal mirror and the mirror-mounted motor will also be covered, as the timing determines how often the GUI is updated, and when the laser should be pulsed. Our final point for the LPU is the idea that the beam will be reflected, so making sure the image we project does not become flipped from the mirrors, while also preserving beam quality.

6.2.1.1 Controlling the Beam Path

Learning how to manipulate a laser beam to be moved in two dimensions was important for the LPU. Two mirrors are needed: one in the horizontal direction, and the other in the vertical direction. The first mirror will rotate upon one axis, while the second mirror will rotate on one of the other axes perpendicular to the first. For our purposes, the LPU will be designed with the polygonal mirror rotating upon the axis that is perpendicular to plane generated from the surface the GUI will be displayed, i.e., a desk, table, or any other flat surface. The second mirror will rotate upon the axis that is perpendicular to the first mirror, and the axis that passes through the user.

6.2.1.2 Orientation and Dimensions

The laser, photodiode, polygonal mirror, and mirror-mounted motor will all be located on the same plane to reduce the amount of angular noise within the system. The second mirror will need to point downwards to enable the laser light to contact the surface upon which the LPU is sitting on. A rough schematic is shown in Figure 4.

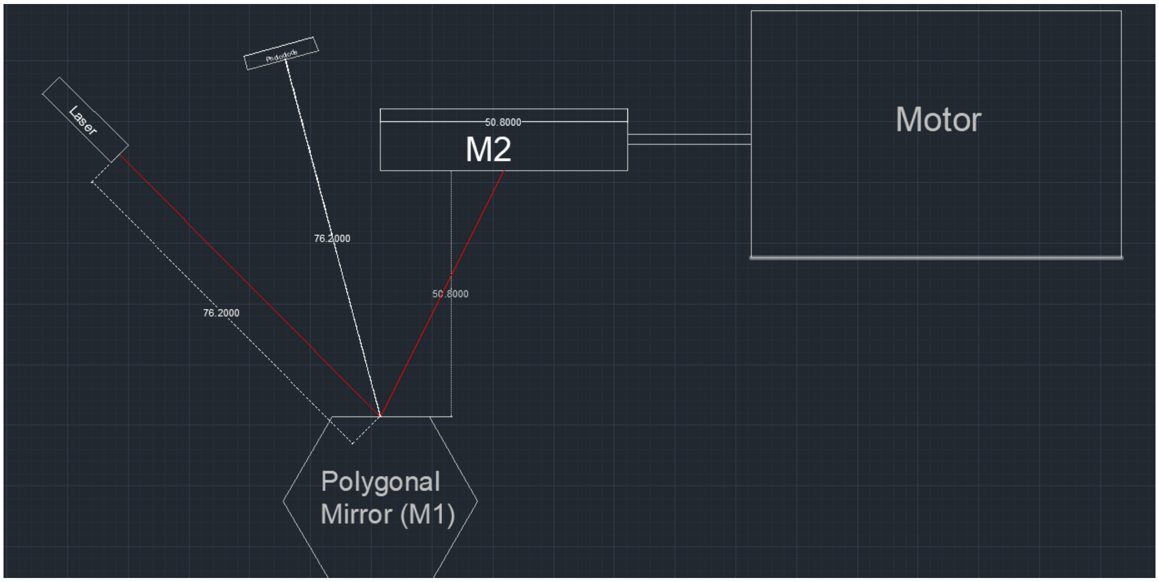


Figure 6 - LPU Schematic. Distances in mm.

6.2.1.3 Laser to Mirror 1 Distance

The only thing that the distance between the laser and the polygonal mirror affects is the divergence of the laser. Minimizing this distance as much as possible reduces the amount of spread the laser light will experience, which affects how clear the GUI will be. This also affects the amount of light the photodiode receives from the laser, as the photodiode has a small active region. Any light outside of the active region will not be detected, and the intensity experienced by the photodiode will be reduced. This affects the amount of current generated by the photodiode that synchronizes the mirrors to effectively generate the text, and therefore could cause disturbances in the GUI generation.

6.2.1.4 Mirror 2 Angular Travel

Our specification calls for the GUI to be created at a distance of 12” from the device, with dimensions 12” long and 9” wide. The polygonal mirror will control the length of the GUI, while the second mirror controls the width. To find the angle the of the transmitted laser beam with respect to the normal of the mirror, I used $\arctan\left(\frac{x}{y}\right)$, where x is the desired width of the GUI and y is the total optical path length the laser travels. This means that, with $x = 9''$ and $y = 12''$, the laser is at an angle of 36.87°. To find the how much of an angle the motor must sweep, we divide this angle by 2, to get an angular travel of 18.44°.

6.2.1.5 Laser Beam Incident Angle

Using the same equation when we calculated the angle of the second mirror, we can find that, to reach the full length of 12” at an OPL of 14”, we would need the laser to be incident on the surface of the polygonal mirror at 49.4° relative to the mirror. One thing to consider is that while the polygonal mirror is rotating, the distance between the first and second mirror will grow smaller as it approaches the edge of the hexagon the mirror is shaped out of. This should have minor effects on the size of the GUI and would be hard to correct for. As such, we will think of the specification as an average size of the GUI and not an absolute.

6.2.1.6 Mirror 2 Size

Since the above equation uses the OPL, the distance between the two mirrors must be considered when calculating the length provided from the first mirror. Assuming this distance is 2”, we can find not only the angle at which the laser will need to hit the first mirror, but also the determine how large the second mirror needs to be to capture the full sweep of the laser beam. This will help optimize the amount of room we have, as reducing the size of the second mirror to a little over the length we need can help reduce the width of the device.

6.2.1.7 Mirror Speeds and Timing

To synchronize the mirror speeds, we will be using a photodiode that will be in the beam path as the polygonal mirror is turning. Placed right before the second mirror, when the photodiode receives a light from the laser diode, it will allow current to pass, which will send a signal to the MCU and generate interrupts in specific timespans. Another purpose of the photodiode is to calculate the speed of mirror 1.

These interrupts will serve as moments in time where the laser diode will pulse, based on information provided from the MCU. The pulses will be what creates the text for our GUI.

Since we will be using a hexagonal mirror, we can use a reduced RPM for the mirror. The laser will be sweeping the distance it needs to 6 times faster than the RPM of the mirror.

6.2.1.8 Laser Quality

As mentioned in the previous sections, the laser beam quality affects the clarity of the output GUI. If the beam is too wide, then the edges will appear fuzzy and mesh together too much and may even make the text unreadable. For this reason, the distance the laser will have to travel is going to be kept at a minimum, as the farther the laser travels, the more the beam will diverge. Another point of consideration is the optical power and intensity of the beam. Since we will be using a lower powered laser, we must consider the fact that the laser will appear dim, especially when reflecting off of two mirrors and constantly shifting direction. For this reason, if the laser we are using does not appear well in our testing, a higher power laser could be used. Warning labels will be in place to warn against pointing the laser into the eyes.

6.2.2 Lens System Design

For this next phase of our project, our optics and photonics team member will oversee designing a lens system that is focused on improving upon our image capturing capabilities. The main focus will be on improving upon both the depth of field and the field of view. To begin this section, our priority is to gain a better understanding of the Raspberry Pi High-Quality Camera. This camera serves as a base foundation for our project. Therefore, it is imperative that we are able to gather all the necessary information regarding its specifications and its sensor details. Gathering all the information regarding this camera will allow us to establish critical parameters and sensor format information that will help in our following sections on lens designing. Once the camera’s specifications are laid out, we will begin to understand how the image capturing process is done without the use of any lens designing. This will help create a base in which we could measure just how effective our designs are at enhancing the camera’s capabilities. Afterwards, we will begin to dwell into just how effective each objective is on our image quality. The first objective that we are attempting to complete in our lens system is the enhancement of the depth of field. We will be using a calculations to help figure out a way to not only replicate what the camera can produce but attempt to surpass this as well as an initial starter into our design process. The main focus of this will be utilizing knowledge of optics into analyzing the ways we could focus our camera at varying distances, specifically within the 5-foot range that is our gesture recognition process. The second objective that we will be attempting to complete will be in the widening of the field of view. This section as well will likely utilize heavy calculations to help construct the correct lens necessary when considering the factors involved. These factors will include lens types and curvatures that will be heavily designed. It is important that using the knowledge and simulation tools, that we can construct the appropriate lens that will help in improving upon the field of view without drastically compromising the image quality of the camera. The final portion of this section will involve combining all the knowledge that we have formulated from earlier in the depth of field and wide-angle field of view into combining these lenses into a singular system. This series lens system will ultimately be the design that we will integrate into our project. This will allow us to slowly inform our methods in which we design our lens system and for our gesture recognition system. As we begin this design process, we will continue to investigate ways to design our lens system that will meet our project's objectives and help create an efficient and effective gesture recognition system.

6.2.2.1 Camera Specifications

For this section, we will begin to further understand the raspberry pi hq camera to fully establish everything in relation to the lens system. This will ultimately help in the sequential sections for the lens system. The Raspberry Pi High-Quality Camera comes with a Sony IMX477 sensor. This sensor has a resolution of 4056 x 3040 pixels, that covers an image area of 6.287mm x 4.712mm (7.9mm diagonally). Each pixel comes in at 1.55 micrometers x 1.55 micrometers, and the optical size is (1/2.3"). The focus on this camera is adjustable, which can be programmed and adjusted via the threading of the lens through the m12 lens adapter, providing a depth of field of approximately 314mm to infinity. The focal length of this camera is dependent upon the lens system. Since this raspberry camera is essentially just a CMOS sensor, it is imperative that in the construction of our lens system, it is sufficient in not only providing an image, but one that can meet the specifications regarding the distance and field of view.

Table 16 – Raspberry Camera Sensor Information

Camera Specifications:	Information
Camera Model:	Raspberry Pi HQ Camera
Sensor:	Sony IMX477
Sensor Resolution:	4056 x 3040 pixels
Sensor Image Area (Width x Height):	6.287 x 4.712mm (7.9 mm diagonally)
Pixel Size:	1.55 µm x 1.55 µm
Optical Size:	1/2.3"
Focus:	Adjustable
Depth of Field:	Based on Lens System
Focal Length (F):	Dependent on Lens System
Horizontal FOV:	Dependent on Lens System

Vertical FOV:	Dependent on Lens System
Focal Ratio (f-stop):	Dependent on Lens System

6.2.2.3 Effective Focal Length of Camera

To establish the ideal types of lenses we would be using, we now need to understand the effective focal length of the camera itself. This is an important tool that will be needed for ultimately deciding upon the lenses we would be designing. It will be so that we are consistently maintaining a consistent focal length throughout the lens system once the lenses themselves are implemented. The effective focal length is the distance from which the optic is to the focal point. The following compounded lens equation will help in finding the solution:

$$F = \frac{Horizontal\ Sensor\ Dimension}{2 * \tan(\frac{AFOV}{2})}$$

From using this equation, we were able to calculate the necessary focal length the lens system must obtain to meet the specifications regarding the horizontal field of view.

Table 17 – Horizontal Field of View Requirements

Focal Length Requirement on Angular HFOV	
Angular HFOV (in Degrees)	**Focal Length (mm)**
60	5.445
65	4.934
70	4.489
75	4.097
80	**3.746**
85	**3.431**
90	**3.144**
95	**2.88**
100	**2.638**
105	**2.412**
110	**2.201**
115	**2.003**
120	**1.815**

Seeing this, we were able to highlight that to meet the horizontal field of view requirement, we must build a lens system that is at least 3.746milimeters. Next, we have to focus on establishing the focal length requirement for the vertical field of view. To do this, we must use the previously mentioned formula but this time utilizing the vertical sensor dimension.

$$F = \frac{Vertical\ Sensor\ Dimension}{2 *\ \tan(\frac{AFOV}{2})}$$

Table 18 – Vertical Field of View Requirements

Focal Length Requirement on Angular VFOV	
Angular VFOV (in Degrees)	**Focal Length (mm)**
30	8.793
35	7.472
40	6.473
45	**5.688**
50	**5.052**
55	**4.526**
60	**4.081**
65	3.698
70	3.365
75	3.07
80	2.808
85	2.571
90	2.356
95	2.159
100	1.977

Here we can see that the previously established 3.746mm total focal length will exceed our vertical field of view, which is perfectly fine when designing our lens system. With the effective focal length of the camera being at minimum 3.746mm, we now understand that we must design lenses that help complement this focal length. In the following section, we will be discussing the various lenses available and their relation to our project. However, we understand fully that these lenses will have to be designed to fit this focal length range to help with completing our enhancement of the depth of field and field of view. As we progress forward in the design section of the project and eventually dwell into the prototyping of our designs, we understand that adjustments to the effective focal length can become necessary once our lenses are fully integrated alongside the camera.

6.2.2.4 Types of Various Lenses

This section discusses the different lenses considered within our optical design, and why the team chose the lens options that we did.

6.2.2.4.1 Plano-convex lens

A plano-convex lens is a lens that is often utilized in configurations of lens systems. It has a flat side which we call plano and another side which is curved outwardly. The main purpose of the plano-convex is to focus lights and is helpful in converging light beams. It plays a crucial role in our project by helping bend light toward a central point.

6.2.2.4.2 Bi-convex lens

A bi-convex lens is a lens that shares a similar configuration to the plano-convex lens. It is a lens that is curved outward on both sides, as opposed to the plano-convex. It has a similar configuration to a plano-convex lens in that it helps with converging light beams. The main difference is that it can help converge light beams from either side. It is a very versatile lens where it allows for the light to converge from different angles as well.

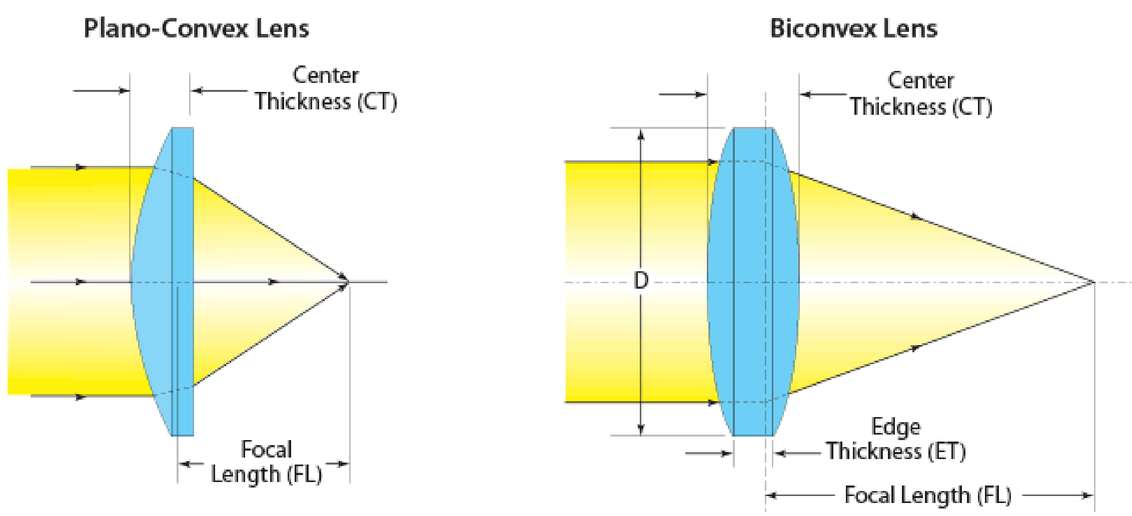


Figure 7 - Plano-Convex and Biconvex Lens Diagram

6.2.2.4.3 Plano-concave lens

A plano-concave lens is a lens that is flat on one side (plano) and contains a curved inward side on the other (concave). The main purpose of a plano-concave lens is that it helps with diverging light beams. Its ability to spread out light would be advantageous in situations where diverging light beams are necessary.

6.2.2.4.4 Bi-concave lens

The bi-concave lens is a lens that is curved inward on both sides. It has a similar purpose to a plano-concave lens in the fact that it diverges light. The main difference is that it helps with diverging light beams from either side of the lens. It is very effective at helping control dispersion and the divergence of light.

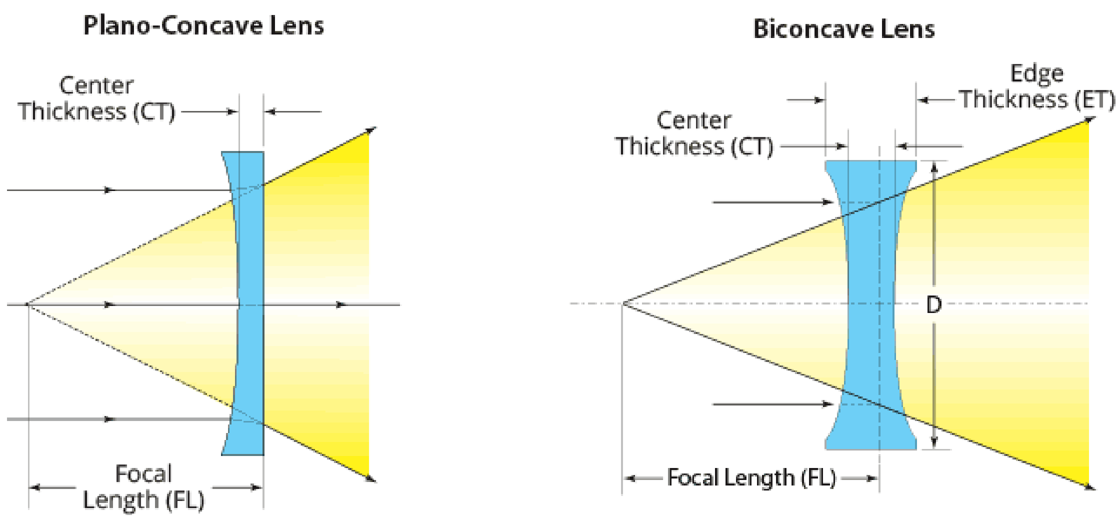


Figure 8 - Plano-Concave and Biconcave Lens Diagram

6.2.2.4.5 Spherical Lens

Spherical lenses are a type of lens that tend to be shaped in a variety of ways. Though they have similar configurations to the previously mentioned lenses, they aren't necessarily meant for similar use. These types of lenses are known for their ability to be very cost-efficient which tends to lead to optical aberrations. Due to their cost-efficiency, these lenses are used in products where lens budgeting isn't necessarily high. As for the optical aberrations, due to their properties, they tend to cause image degradation and comas which can be an off point for most complex optical designs.

6.2.2.4.6 Aspherical Lens

It is a lens that has a non-spherical surface that is designed to correct optical aberrations. It is very useful in helping spherical aberrations. They are helpful at helping with image quality by reducing distortions.

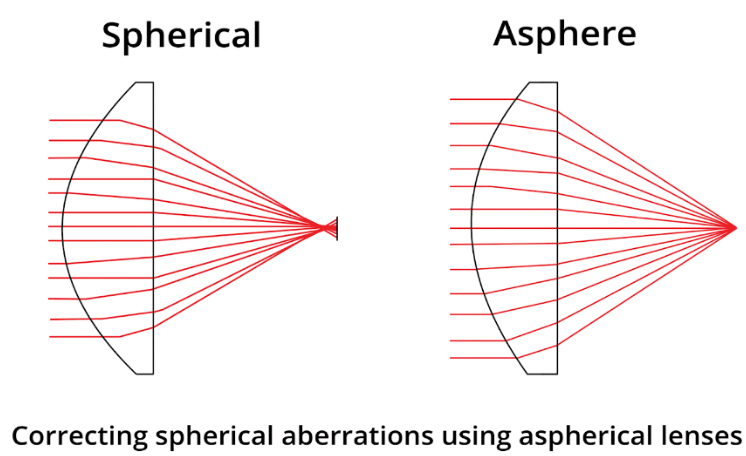


Figure 9 - Spherical and Aspherical Lens Digaram

6.2.2.4.7 Achromatic Lens

It is a lens that uses two or more lens elements from different materials. It is useful in correcting chromatic aberrations. The utilization of achromatic lenses helps different colors of light focus at different points.

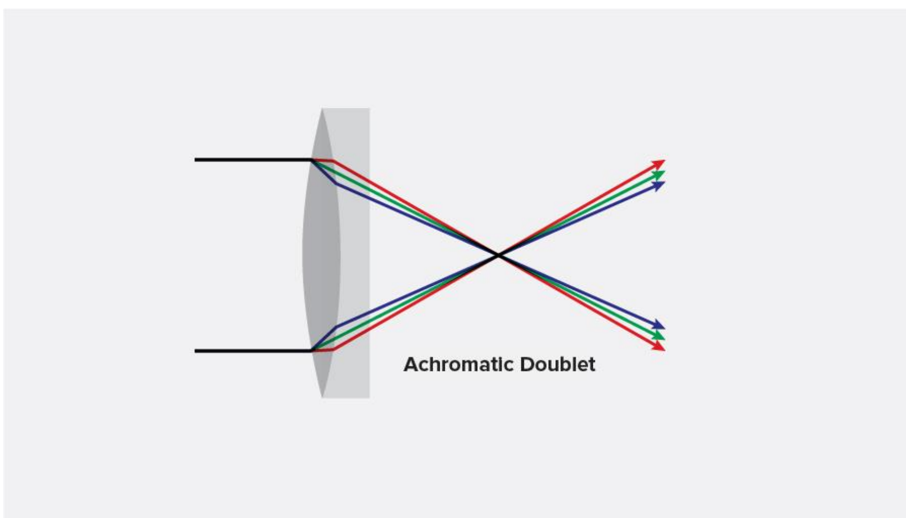


Figure 10 - Achromatic Lens Diagram

6.2.2.5 Depth of Field System

For this section we will be tasked with attempting to figure out the various ways in which we can enhance the depth of field. Since the camera we have chosen is very dependent on having a lens system to achieve an image, it is imperative that our lens system is first focused on attempting to gather a depth of field to then meet our distance goal. To get this, we need to first calculate the effective focal length of the camera, which we have done in the previous section as at least 3.746mm, and later calculated as 3.56mm then begin to figure out the hyperfocal distance and near focus limit calculations.

The hyperfocal distance and near focus limit formulas are as follows:

$$H = \frac{f^2}{N \cdot c} + f\,, \quad N_L = \frac{S \cdot (H - f)}{H + S - 2f}$$

where N is the f-number, c is the circle of confusion and f is the focal length. The hyperfocal distance equation. For this calculation, we will be using the circle of confusion of 0.025mm as this is the standard circle of confusion used in the raspberry camera module and we will be using the focal length and f-number from the raspberry camera as well. For the near limit focus formula, S represents the subject distance, and, in this case, it is 5 feet, which equivalates to 1524mm.

Hyperfocal distance of Lens System:

$$H = \frac{(3.56)^2}{(5.6)(0.025mm)} + 3.56mm = 98.821mm$$

Near Focus Limit of Lens System:

$$N_L = \frac{S \cdot (H - f)}{H + S - 2f} = \frac{1524(98.821 - 3.56)}{98.821 + 1524 - \big(2(3.56)\big)} = 89.779mm$$

Now that we have acquired the values for the hyperfocal distance and near limit of the raspberry camera, we can determine that the depth of field of the camera alone has a depth of field that begins at 304.8mm and extends all the way theoretically to infinity. The near focus limit of 89.779mm helps explain that that is where the lowest possible distance of which the sharpness is acceptable. We also understand that to increase depth of field, we must maximize the way we can do this by increasing the subject distance to camera. Therefore, we will implement our machine learning model to collect hand gestures at the furthest distance it will let us before it cannot acquire anymore.

6.2.2.6 Aperture Sizing

When it comes to understanding apertures, we must understand the dilemma of f/#, known as f-number. Since our requirement specification is heavily dependent on distance, we have to properly implement an aperture into our lens system. Here we must understand that aperture size plays a crucial role because of how much light we can possibly let enter towards the sensor. If we have a high f-number, this correlates to a small aperture size, while having a low f-number, corresponds to a large aperture size. Since our project is a requirement of having a specification of distance of 5 feet, we would ideally want a high f-number. Since it corresponds to a deeper depth of field. Now, there is a trade-off we must acknowledge when it comes to depth of field. Lighting is the issue we must address. Yes, it is imperative that we want a deep depth of field, but as we increase the depth of field, the light we let through the aperture hole becomes less bright as the hole decreases in size. Therefore, if we want to balance this phenomenon, we must settle upon a mid-length aperture size.

6.2.2.7 Wide-Angle Field of View System

In this section, we will be briefly talking about the construction and design of the lens system itself. We know that the two-lens system will comprise of the bi-convex lens and the plano-convex lens. Knowing this, we can utilize the compounded lens equation to figure out what focal length is needed on each lens to reach out total effective focal length. Since we know that the total effective focal length must be at least 3.746mm, we can vary between several different ones, while adjusting the distance between the two lenses to achieve the result.

Compounded Lens Equation:

$$\frac{1}{EFL} = \frac{1}{f1} + \frac{1}{f2} - \frac{d}{f1(f2)}$$

$$\frac{1}{EFL} = \frac{1}{4.5} + \frac{1}{15} - \frac{0.3}{4.5(15)}$$

$$\frac{1}{EFL} = 0.2844$$

$$EFL = \frac{1}{0.2844} = 3.56mm$$

New Horizontal Field of View Equation:

$$HFOV = 2 \cdot \tan^{-1}\left(\frac{Sensor\ Width}{2 \cdot f_{eff}}\right)$$

$$HFOV = 2 \cdot \tan^{-1}\left(\frac{6.287mm}{2 \cdot 3.56mm}\right) = 81.4722$$

New Vertical Field of View Equation:

$$VFOV = 2 \cdot \tan^{-1}\left(\frac{Sensor\ Height}{2 \cdot f_{eff}}\right)$$

$$VFOV = 2 \cdot \tan^{-1}\left(\frac{4.712mm}{2 \cdot 3.56mm}\right) = 65.6828$$

Table 19 - Focal Length Parameter Table

Focal Length	3.56mm
HFOV (in °)	81.4722°
VFOV (in °)	65.6828°

As we can see from our calculations and in the table provided above, using the two lenses that we have determined to be sufficient in our project, gave us the focal length of 3.56 millimeters. This will result in a horizontal field of view of at least 81.4722 degrees, surpassing those specifications. Also, our vertical field of view is at least 65.6828 degrees, which will greatly surpass our specification regarding the vertical field.

From the calculations we have done into the depth of field and the field of view, we have determined the two lenses from the design portion that we will be implementing into our prototyping of the project. The first lens will be a plano convex lens of a diameter of 12.7 millimeters, with a focal length of 15 millimeters. It will contain an aperture of F/1.81, with a radius of curvature of 7.75mm, and with a back focal length of 11.53 mm. In terms of its thickness, it will have a center thickness of 5.25 mm and an edge thickness of 1.94 mm. The second lens will be a bi-convex lens of a diameter of 3 millimeters with a focal length of 4.5 millimeters. This lens will have an aperture of F/1.5, with a radius of curvature of 4.29 mm and a back focal length of 3.78 mm. Its thickness will be a center of 2 mm, while maintaining an edge thickness of 1.47 mm.

Table 20 – Series Lens for Lens System

Type:	Plano Convex	Bi Convex
Diameter	12.7 mm	3 mm
Focal Length	15 mm	4.5 mm
Aperture	F/1.81	F/1.5
Radius of Curvature	7.75 mm	4.29 mm
Center Thickness	5.25 mm	2 mm
Edge Thickness	1.94 mm	1.47 mm
Back Focal Length	11.53 mm	3.78 mm
Coating	Uncoated	Uncoated
Substrate	N-BK7	N-BK7

6.3 Mechanical Design

For our project, there will be very minimal mechanical design needed, mainly the development of the housing for the LPU and for the lens system. The LPU will need proper support so that it can stand far enough from a surface without needing to be permanently attached, i.e., with glue or bolts. We will also need to provide good support for the interior mechanisms of the LPU, like the mirrors, laser, and photodiode. The consolidation of the wiring for the LPU will also need to be considered, as a messy display could lead to an accident or some kind of short waiting to happen. Our design will be featuring the creation of a box of sorts that will be slightly elevated in the air to serve a crucial purpose. The purpose will be to cut out a triangle shape underneath to help give the laser projection unit an output area after it has gone through countless components. The enclosure house will also contain a cut out circle on the front to help act as an opening for the camera. This will lead further into the housing for the lens system with the camera. The enclosure will contain inside all the circuitry, tucked away in a safe manner to prevent any harm to the user and to prevent a poor looking product. It will be up to the group to manage a way to contain all the circuitry in a way to prevent it from getting in the way of the pathway of the laser diode. This mechanical design shown below will help with facilitating a more visual representation of how we expect our enclosure housing to look at this time.

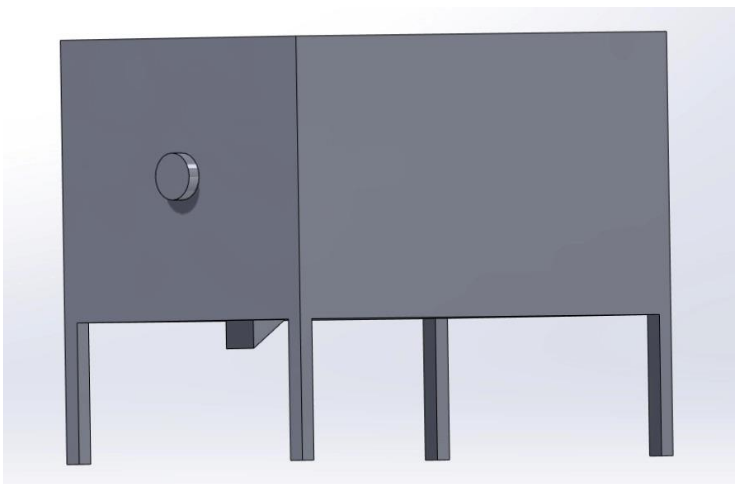


Figure 11 – Enclosure Housing Drawing

6.3.1 Lens Mounting

When discussing the mounting necessary for the lenses incorporated within our lens system, we would imagine that it would not involve a complex facilitation of components. At this time, we believe that it is only necessary to provide a lens mount for the lenses to be held in. The functional mounts which we are using are meant to securely hold the lenses within the optical setup. These mounts will play a crucial role in helping maintain the integrity of the lenses and to help with stability for when we align the lenses for our capturing process. It is crucial for us to at least secure these lens mounts for our various lenses as a safeguarding measurement, as without it, we can face issues where these lenses could potentially be marked or scratched due to mishandling of the fragile lenses. This in turn would delay the progress of our project as these lenses would likely lead to poor image processing and new lenses would need to be ordered. Since our lenses will eventually be housed in some capacity, the emphasis at this current moment is simply on securing functional lens mounts. We will continue to discuss the methods in which we build our housing around the lens system later in our project. Since our lenses are fairly small in size, we can essentially 3D print a housing of sorts to help with securing these lenses in place with the aperture as well, to then thread into our adapter onto the sensor itself. Below we have images regarding the 3D print itself.

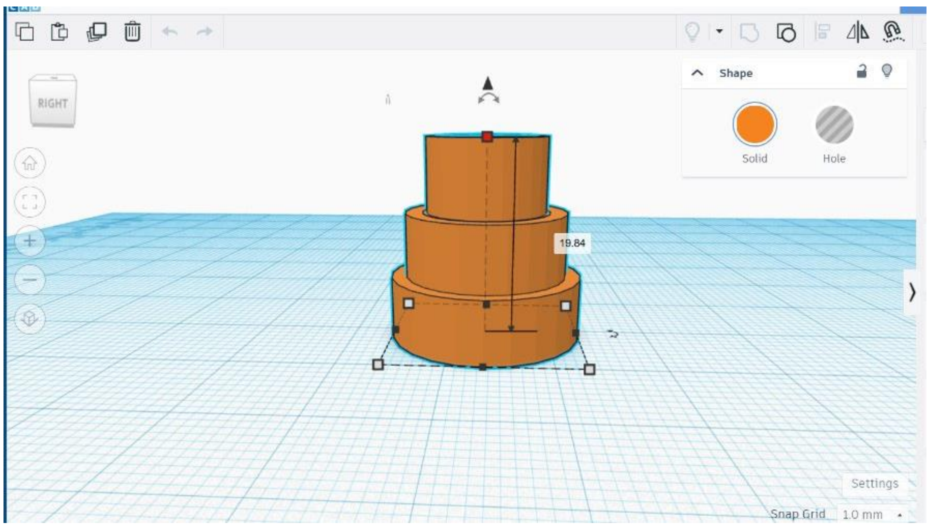


Figure 12 - Housing Distance

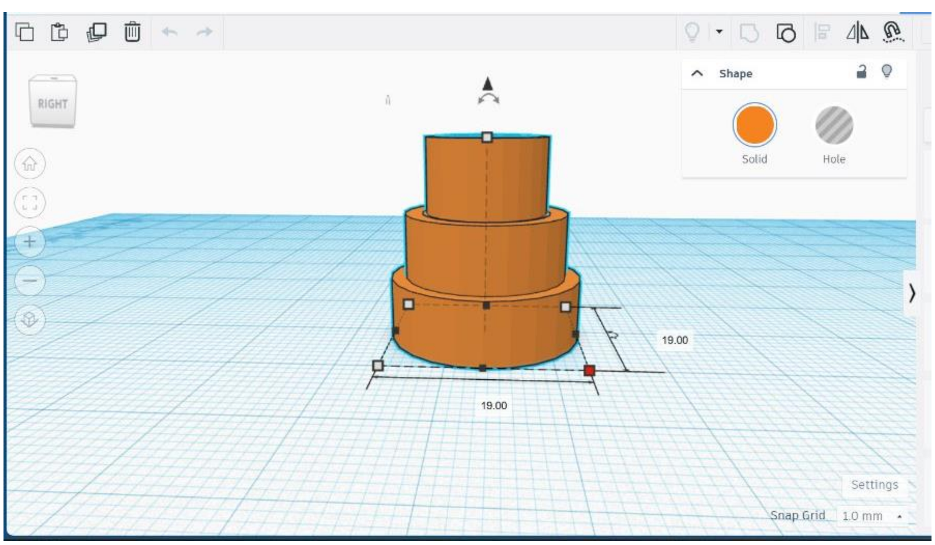


Figure 13 - Housing Specs

As we can see from the two figures placed above, this 3D printed mount will comprise of our lenses. We had to carefully plan out how we would place each lens inside this mount. It is essentially divided into three different layers. The first layer is the smallest, with it having a size of 12mm x 12mm. Since the C/CS mount with the M12 adapter is 12mm x 12mm, we had to size our housing's first layer to meet this parameter. However, the bi-convex lens does not exactly meet this size, so therefore, to help with this, we had to tinker with certain small 3D printed holders. We started this by using the tinker cad application available to size our lens holders. The bi-convex lens as previously mentioned, has a diameter of 3milimeters, so therefore, the hole had to be 3milimeters wide, and overall be 12milimeters. Once we 3D printed this, we essentially placed it within the first layer. When it came to the second layer itself, we had to 3D print the aperture to be fixated in this layer. Since in later sections we utilized Zemax to understand that aperture size is relative towards the sensor and lens system itself, an f/5.6 aperture is dependent on the simulation tools. When we used zemax itself, we noted that the hole of the aperture size was about 1.4milimeters in diameter. Once we 3D printed this and placed it within the lens holder, it perfectly secured itself in place.

The final layer would hold the second lens of our two-lens system, in the plano-convex lens. Since we understood that the plano-convex lens has a diameter of 12.7milimeters, we had to tinker with our drawing. If we were to print a 12.7mm layer, it would not fit exactly, so therefore we had to up the size of the final layer to 19 millimeters so that the

6.3.2 Camera Mechanical Drawing

The following mechanical drawing of the raspberry camera module will help serve as a guide into visually understanding its physical dimensions. This will serve as a helpful tool later in the construction of our enclosure housing. This is due to obtaining the physical dimensions that will allow for a more precise measurement into creating the various holes necessary for the camera to be placed in. Once we begin the building phase of our project, it is imperative that we have all the components laid out with all their specifications and dimensions to allow for a transition less period for where we can piece all the various components together. Since we do not have a member in our group that is necessarily mechanical engineering heavy, it will be up to us as a group to work together to properly construct everything in a neat fashion.

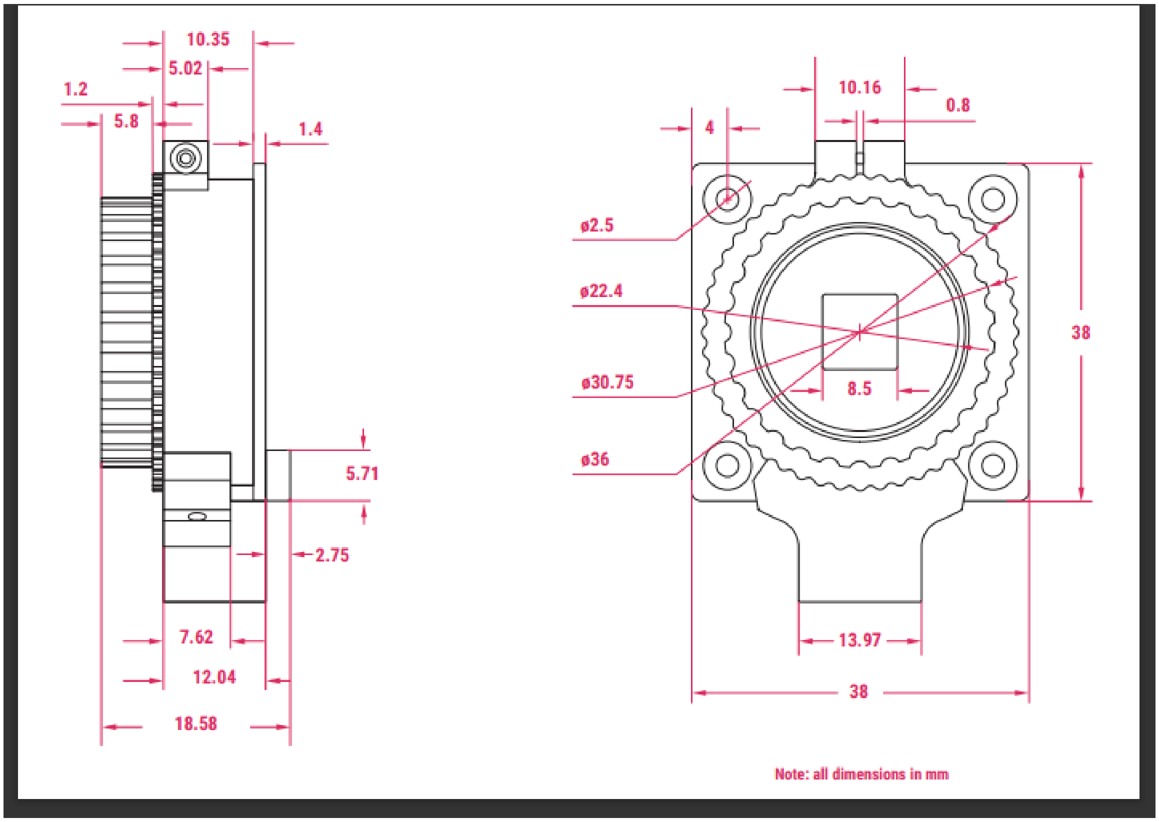


Figure 14 - Mechanical Drawing of Raspberry Pi HQ Camera

7. Software Design

This section details the software considerations our team has made, and plans to make, throughout the design of our gesture-controlled synthesizer. Software design is broken into two primary components, PC development and embedded development.

7.1 PC Development

This section gives explanation of our design processes for development on the PC. In this section we start off by discussing Python Development practices we're following. Afterwards, we'll discuss data collection, training, and implementation of our machine learning model. Finally, we'll go over odds and ends of development, such as sound generation and operating system installation.

7.1.1 Python Development

All our Python development was completed using the PyCharm IDE by JetBrains. This is a subscription-based IDE, but our team was able to secure a student license through UCF. PyCharm is a feature-rich IDE for Python that offers intelligent code assistance, advanced code refactoring, a built-in debugger and profiler, integrated testing support, version control integration, and web development support. PyCharm's code analysis and inspections help catch potential errors, and its customization options and plugin support make it adaptable to various development workflows. For us, PyCharm shines in the fact that it has built in support for importing Python libraries and dependencies, without the necessity of the command line.

To follow good Python development practices, a Python virtual environment was set up on any device that was used to import a Python library. This includes the PC. A Python virtual environment is a self-contained directory that contains its own Python interpreter and a set of libraries. The purpose of using a virtual environment is to create an isolated environment for a Python project, separate from the system-wide Python installation. This helps manage dependencies, avoid conflicts between different projects, and ensure that a project can run in a consistent and reproducible environment.

7.1.2 Audio Interfacing and Sound Generation

This section details the steps we'll go through to generate sound in our Python program. Initially we'll discuss requirements for generating an audio output in the code, and then we'll delve into how our program will interface wired and wireless audio devices.

7.1.2.1 Generating Sound with Python

To generate sound using Python, we'll have to create a couple of different Python scripts. Initially, we'll need to define a function that outputs a sine wave based on a given amplitude, frequency, and time. We'll use the pygame library for sound generation. Pygame is a powerful Python library used for making games in Python.

Once a program is written to output a sound wave, we'll generate a look-up table containing the frequencies of each of the notes we want to implement. For our purposes, frequency is the only information we'll need. To generate multiple notes at once, we'll have to make use of threads.

7.1.3 Machine Learning Model Development

This section provides an overview of our machine learning model development process. First, data collection for model training is discussed, followed by training the model. Finally, we'll discuss exporting the model into a TensorFlow platform.

7.1.3.1 ML Model Data Collection

The first step toward creating and training our machine learning object detection model is to collect data to give the model. In this case, our trainer requires images of our individual hand gestures. Our team chose to write a data collection script with Python.

Beyond typical Python libraries, our data collection script takes advantage of CV2 from OpenCV and MediaPipe. CV2 is a popular computer vision library for image and video processing. The cv2 module provides an interface to access the functions and tools offered by OpenCV in Python. MediaPipe is a library developed by Google that provides a framework for building multimodal applied machine learning pipelines. It is specifically designed for media perception tasks, allowing developers to build applications for various tasks, including face detection, hand tracking, pose estimation, and more. It is particularly useful for projects involving real-time analysis of video or image data, where tasks like tracking and recognizing specific features are required.

Our detection script takes advantage of the pre-trained HandDetector object detection class from cvzone. We can integrate HandDetector with MediaPipe, allowing us to open a live video feed from the webcam, and detect a human hand within the feed. From there, the script extracts the hand and outputs it over a white background as a separate window. Finally, holding the 's' key on the keyboard while a hand is in frame automatically saves a jpeg image of the hand into a directory of our choosing. We can now semi-automate the data collection process. For our purpose, we create a directory with a name corresponding to the gesture and capture approximately three hundred images of said gesture, attempting to do so using multiple different backgrounds.

7.1.3.2 Training the Model

Our team will be using Google's Teachable Machine website to build and train our machine learning model. Teachable Machine is a powerful and intuitive tool that makes it easy to create and export a model.

When getting started on Teachable Machine, we first select that we are designing an image project. Specifically, we are opting for an embedded image model to maximize performance. For each of our gestures, we define a class on the website, and upload all our gesture data to each class. After Teachable Machine trains the model, we can easily export it to TensorFlow Lite. As of this time, we can choose a 'TensorFlow Lite floating point' model or a 'TensorFlow Lite for Microcontrollers' model. Our team has opted to implement both and determine which has better performance. After benchmarking each model, we will make a finalized selection. The following figure shows the Teachable Machine UI.

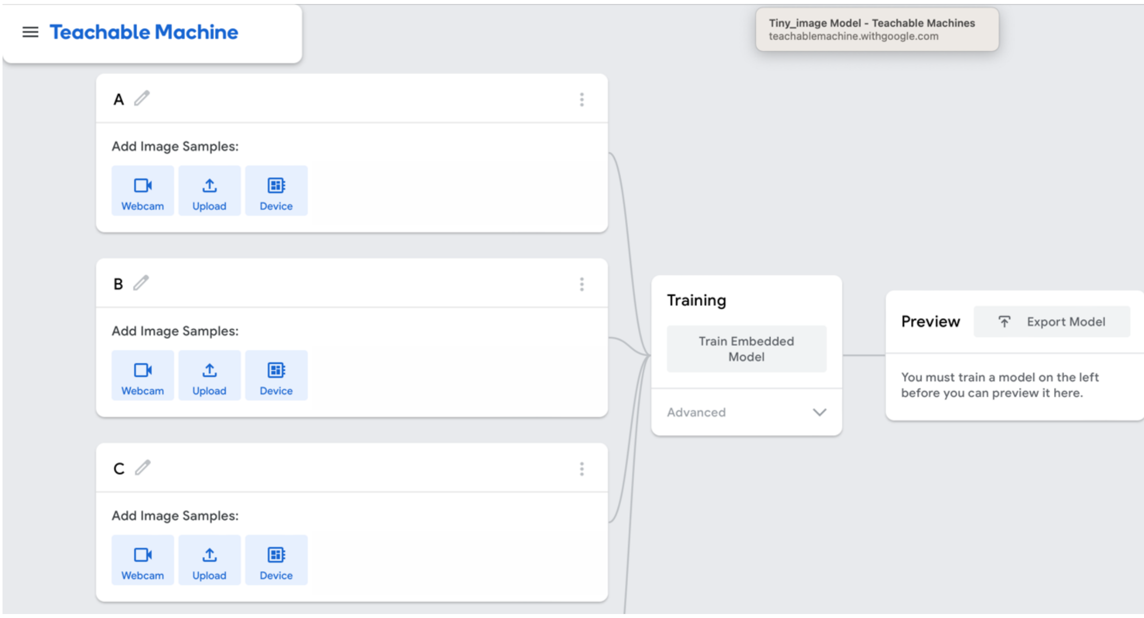


Figure 15 - Teachable Machine UI

7.1.3.3 Implementing the Model on the PC

Fortunately for us, Google has detailed guides for model implementation. The Tensorflow GitHub repository has implementation examples for many kinds of TensorFlow Lite models on various devices. This guide details how to create a virtual environment on the PC, how to import the different Python dependencies, and how to run the provided example model. Following this guide will allow us to get a feel for development on the PC and will give us the background information we need to run our custom machine learning model.

7.2 User Interface

The user interface for our synthesizer will be provided by the laser projection unit. Due to the fundamentally restrictive nature of these devices, the team was challenged to create a mock-up that would display any required information that may help enhance user experience. The synthesizer will have two modes, so two GUI mock-up images are provided below.

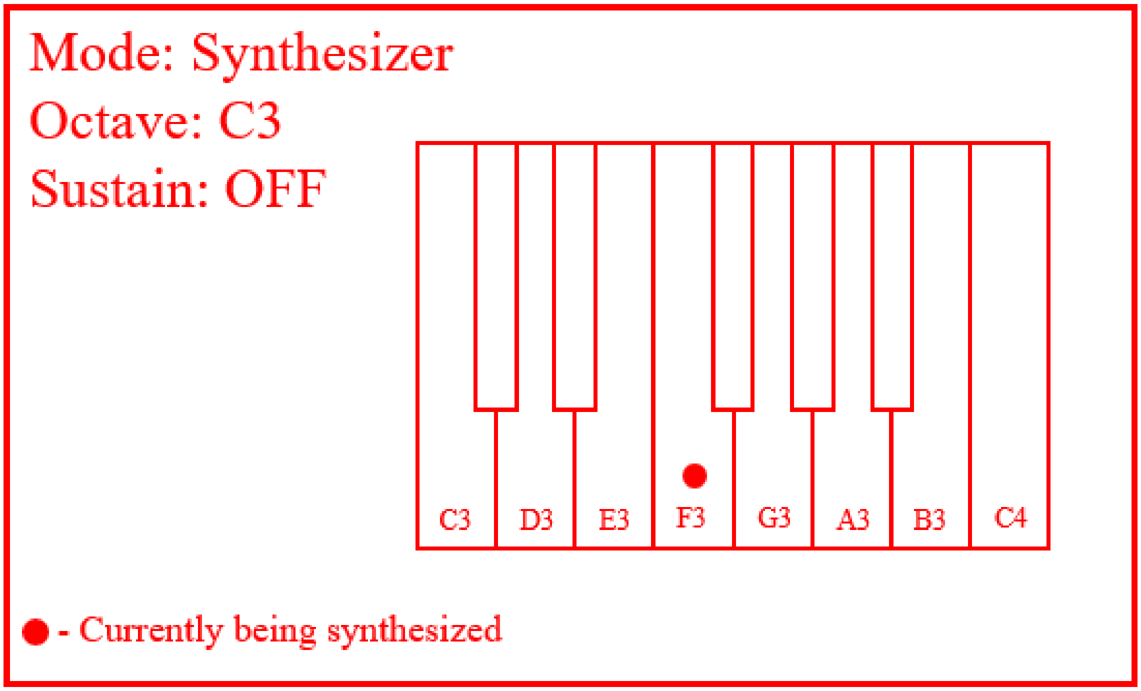


Figure 16 - Proposed User Interface; Synth Mode

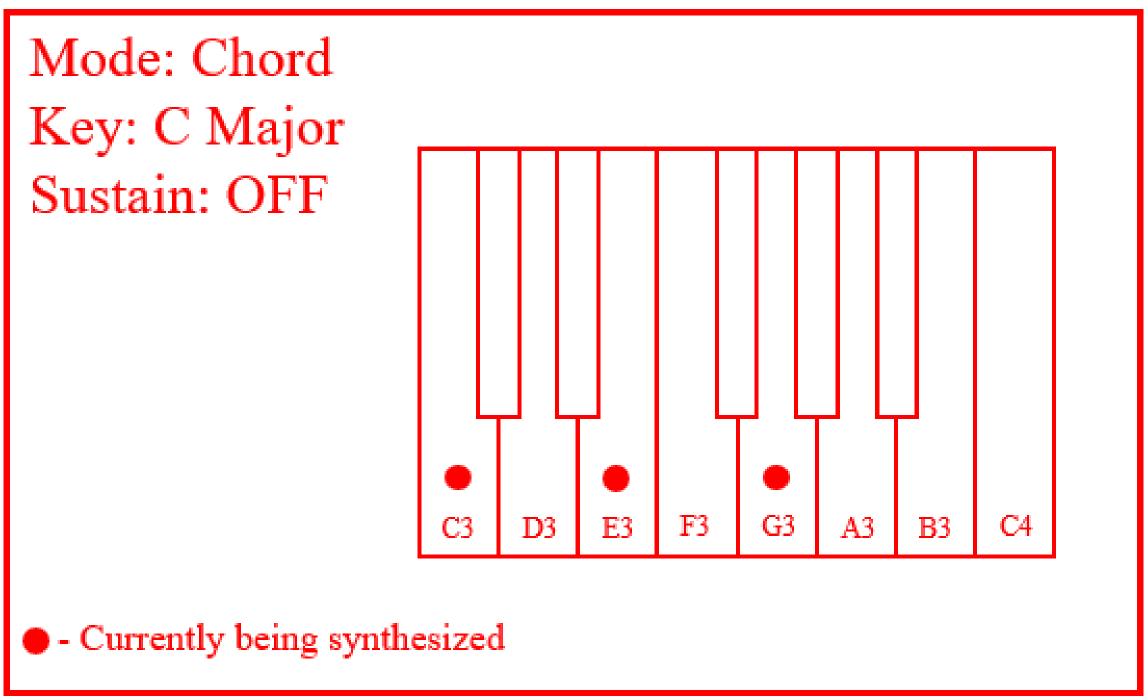


Figure 11 - Proposed User Interface; Chord Mode

7.3 Embedded Programming

In this section, we will discuss how we plan to program each component in our system—everything from computer, MCU, and the laser projection unit parts. We will use Python for the computer, and for the MCU, we will use C++ to control all external devices connected to it.

7.3.1 ESP32-S3-DevKitC-1-N8R2 Software Setup

To initialize the setup of the ESP32-S3-DevKitC-1-N8R2, we select the Arduino Integrated Development Environment (IDE) because it provides comprehensive support for a range of microcontrollers, including the ESP32-S3. Our initial step involves installing the Arduino IDE and enhancing its capability to accommodate the ESP32-S3 board. We achieve this by adding the ESP32-S3 boards manager URL to the Arduino IDE settings, a URL we can obtain from the Espressif official website or GitHub repository. This inclusion enables the installation of the ESP32-S3 package through the Boards Manager, equipping the Arduino IDE with the essential tools and libraries for programming the ESP32-S3.

As we advance with the development, we focus on programming the ESP32-S3. In the Arduino IDE, we initiate a new sketch specifically for our project, emphasizing UART for communication. Rather than initializing UART using GPIO pins, we configure the communication channel through a UART to USB connection. This approach involves preparing our Python code to detect and utilize a specific COM port for transmitting and receiving data to and from the ESP32-S3.

Our programming efforts are dedicated to establishing and refining the UART to USB communication process, pivotal for the data interchange between the gesture-controlled synthesizer and the computer. Within the Arduino environment, we meticulously develop functions to facilitate the transmission of gesture data to the computer and the reception of incoming data through this connection. This strategy ensures that a robust and reliable communication link is integral to the functionality of our project.

To complete the software configuration of the ESP32-S3, we upload our code onto the module. This step requires connecting the ESP32-S3 to the computer using a USB cable, emphasizing the importance of careful hardware management. In the Arduino IDE, we diligently ensure the appropriate board and port selection to eliminate any potential connectivity issues. Once we initiate the upload, the process translates our meticulously crafted code into actionable commands on the ESP32-S3, effectively establishing the UART to USB communication link crucial for our project's success.

7.3.2 Computer Software Setup

We are currently configuring the ESP32-S3 while preparing the computer for UART communication. This involves customizing the system settings to enable UART functionality and installing specific drivers or modifying system parameters to support serial communication.

We have decided to use Python for our programming language due to its versatility and extensive library support, particularly for serial communication through the PySerial library. Its straightforward syntax and legibility make it ideal for our project. We installed Python version 3 and the PySerial library, which provide a solid foundation for our UART communication script. The PySerial library is essential since it provides the necessary functions and methods for serial communication in Python, seamlessly integrating Python's capabilities with the computer's UART hardware functions.

Developing the Python script for UART communication is crucial to our computer setup. The script manages the UART port at a consistent baud rate with the ESP32-S3's settings, ensuring synchronized communication between the computer and the ESP32-S3. It includes a loop that continually checks for incoming data from the ESP32-S3. Once data arrives, the script processes it and converts the inputs into actionable gesture control commands for the synthesizer. Additionally, the script sends data back to the ESP32-S3, establishing a bidirectional communication pathway essential for our project's interactive nature. This capability allows the computer to interpret incoming data and dispatch control commands or responses to the ESP32-S3, facilitating a dynamic exchange of information.

The completion of the Python script is a significant milestone in our project. It activates the computer's role in the UART communication pathway with the ESP32-S3, embodying the software component of the UART communication framework between the computer and the ESP32-S3. This step is critical to the functionality of our gesture-controlled synthesizer. The meticulous preparation and programming of the computer for UART communication are essential to the success of our gesture-controlled synthesizer project. Through this process, we ensure a reliable and efficient communication link between the computer and the ESP32-S3, leveraging the power of Python and the PySerial library to facilitate this complex interaction. Our commitment to developing a sophisticated, interactive musical experience underpinned by rigorous academic principles and a focus on creating a seamless and user-responsive system is evident in this endeavor.

7.3.3 Photodiode

In our gesture-controlled synthesizer project, we have programmed the ESP32-S3 microcontroller to manage interactions between a photodiode, a stepper motor, and a laser. To accommodate the ESP32's architecture, we assign specific pins like PHOTOCELL_SYNC. We configure setting such as MOTOR_DRIVER_FREQUENCY and PHOTOCELL_ACTIVE_POLARITY to ensure the microcontroller interprets the photodiode signals correctly, whether active, high or low. We use global variables like `PC_COUNT` and `PC_TOTAL` to track the frequency of the photodiode's detected events. The `photocell_isr()` function, an interrupt service routine, is pivotal in managing photodiode trigger events. This function captures time intervals between successive light interruptions, filters out insignificant intervals to avoid errors caused by unintended reflections, updates global variables, calculates the average period and frequency of events, resets the line timer, and initiates a new line drawing cycle when the previous one completes.

During the system's setup phase, we initiate serial communication and configure Pulse Width Modulation (PWM) to control the motor and laser precisely. By attaching the `photocell_isr` to the `PHOTOCELL_SYNC` pin, we enable it to respond to edge-triggered interrupts based on the photodiode's active polarity, thus readying the system for accurate frequency measurement of detected events. During the system's operational phase, the `loop()` function frequently invokes `print_data(),` displaying the average period and frequency of events detected by the photodiode on the serial monitor. This continuous data output is vital for monitoring the system's interactive performance and ensuring its responsiveness to user gestures. The photodiode is crucial to integrating our synthesizer's laser projection unit. It detects laser light interruptions and facilitates interactive visual feedback, essential for the synthesizer to respond dynamically to user inputs, creating an immersive musical experience. We chose the ESP32-S3 for processing the photodiode's signals due to its efficient handling of analog signals and its capacity for precision in real-time signal processing. Setting up an analog-to-digital converter (ADC) within the ESP32-S3 allows for the accurate digitalization of the photodiode's analog outputs, forming the basis for responsive system behavior.

We developed algorithms that interpret the photodiode's signals and translate them into specific synthesizer responses. These responses correspond with different patterns of laser beam interruptions, ensuring that the synthesizer reacts with appropriate audio and visual outputs. We employ filtering techniques to distinguish genuine interactions from environmental noise and interferences, thus maintaining the system's accuracy and reliability. To achieve synchrony between the photodiode's input and the synthesizer's audio-visual outputs, we emphasize real-time, interactive feedback, where user gestures directly influence the synthesizer's output, enhancing the dynamic interplay between sound and visuals. We optimize the embedded programming for performance to ensure the system responds instantaneously to the photodiode's inputs, reflecting the immediacy of physical interactions in the synthesizer's output. We integrate failsafe mechanisms to address unexpected behaviors or signal disruptions, ensuring the system remains stable and reliable under various conditions. Regular testing and refinement of our algorithms are integral to our development process, allowing us to continuously verify the system's performance and safety.

In conclusion, the photodiode is indispensable to our synthesizer project, serving as the linchpin for interactive and responsive musical experiences. Through detailed programming and continuous optimization, we aim to leverage the photodiode's capabilities fully, ensuring it becomes a vital component of our synthesizer. This endeavor underscores our commitment to innovative musical expression, blending technology and creativity to forge new dimensions of interactive performance.

7.3.4 Polygon mirror

Incorporating a laser projection unit is a standout feature in our project to develop a gesture-controlled synthesizer. This unit relies on a polygonal mirror, controlled by a DC motor, for horizontal GUI projection. We selected the ESP32-S3 microcontroller for this task due to its efficient management of complex operations. Our embedded programming strategy with the ESP32-S3 centers on Pulse Width Modulation (PWM) to control the motor's speed and, thus, the mirror's rotation. This precision is crucial for accurately rendering the GUI, elevating the synthesizer to a visually interactive experience.

Our strategy's core involves using the ESP32-S3 to implement PWM. This modulation technique controls the pulse width in a pulse train, directly influencing the power delivered to the motor. Adjusting the PWM signals enables fine control over the mirror's rotational speed. We must program the ESP32-S3's GPIO pins to output PWM signals with frequencies and duty cycles that match the motor's specifications. The duty cycle's length, determining how long a signal stays high in a cycle, will directly affect the mirror's speed, which is essential for the GUI's effective horizontal projection. Another critical aspect is synchronizing the motor's rotation with the laser's operation. This synchronization ensures that the laser beam, deflected by the mirror at precise angles, accurately projects the GUI. We will develop an algorithm that coordinates the laser's on-off pattern with the mirror's angular position. This algorithm will calculate the angular positions required for different GUI elements, translating these into PWM signal adjustments. Safety and precision are paramount in our software design. Our code will include safety protocols to ensure the motor operates within safe limits, protecting the mirror and motor from potential damage. Precision in the mirror's positioning is vital; minor deviations can cause significant errors in the GUI projection. We will incorporate fine-tuning mechanisms in our programming for minor motor operation adjustments. If we include feedback mechanisms like rotary encoders, we will integrate these into our control system. The feedback will help adjust the mirror's position in real-time, enhancing projection accuracy.

Integrating the polygonal mirror into our synthesizer project depends heavily on the innovative and meticulous programming of the ESP32-S3. We aim to create a laser projection system that accurately displays a dynamic GUI by using PWM for motor control and synchronizing the mirror and laser operations. Our focus on safety, precision, and feedback mechanisms underlines our commitment to a high-quality, reliable system. This feature is a functional addition and a step toward redefining user interaction in musical devices. It represents our endeavor to seamlessly blend technology and creativity, creating a visually engaging and sonically rich instrument. Our iterative testing and refinement approach will ensure that each system component, from motor control to GUI projection, aligns perfectly with our vision of an innovative, interactive musical experience as we move forward.

7.3.5 Vertical adjustment DC motor

In our project, we are creating a gesture-controlled synthesizer with a unique feature: a laser projection unit that uses a vertically oriented polygonal mirror. This mirror is essential for accurately scanning the graphical user interface (GUI) vertically, and its precise control is crucial for displaying the GUI correctly. We chose the ESP32-S3 microcontroller because of its robust processing capabilities and suitability for complex control systems. Our approach focuses on embedded programming, translating electronic signals into finely tuned mechanical movements. This precision ensures that the mirror's motion aligns accurately with the GUI's requirements.

Our strategy for controlling the polygonal mirror involves using the ESP32-S3 to generate Pulse Width Modulation (PWM) signals. These signals control the mirror's motor, dictating its rotational speed. Precise PWM control is essential for accurate projection of the GUI, as it allows us to modulate the motor speed and ensure smooth movement of the mirror. This control must be exact, as the mirror's speed directly affects the GUI's alignment and readability during projection. We are programming the ESP32-S3's GPIO pins to output PWM signals with specific frequencies and duty cycles tailored to the motor's requirements. By adjusting the duty cycle's length, we can finely control the speed at which the mirror rotates, which is crucial for the effective vertical scanning of the GUI.

A significant part of our software development is creating algorithms that translate the GUI layout into specific motor movements. These algorithms will map GUI coordinates to precise motor positions. This involves calculating the angular positions needed for different GUI elements and adjusting the PWM signals accordingly. We also focus on synchronizing the motor's rotation with the laser's operation. This synchronization ensures that the GUI elements are projected at the correct vertical locations. We are developing sophisticated algorithms within the ESP32-S3 to manage this synchronization, coordinating the timing of the laser's on-off pattern with the mirror's angular position. In our programming approach, safety and precision are priorities. Our code implements safety protocols to ensure the motor operates within safe limits, protecting the hardware from damage. Accuracy in the mirror's movement is crucial for the clarity of the projected GUI.
Our code includes calibration routines and error-checking mechanisms for real-time adjustments, maintaining the integrity of the projection system. If our design includes real-time feedback mechanisms, such as positional sensors on the motor or mirror, we will integrate this feedback into the control loop. This integration will allow automatic corrections and adjustments, aligning the mirror's position with the intended GUI layout. Our code also ensures that the mirror's movement responds rapidly to user interactions. If the GUI is interactive, the mirror's position might need to change quickly in response to user inputs. We are developing routines within the ESP32-S3 that can adjust the mirror's position swiftly, enhancing the dynamism of the GUI.

In conclusion, the embedded programming of the ESP32-S3 to control the vertical movement of the mirror in our synthesizer's laser projection unit is a complex and exciting task. It involves a blend of precise PWM control, intricate algorithmic mapping, sophisticated synchronization, and the integration of real-time feedback. Our dedication to this detailed approach underscores our goal to create a visually interactive synthesizer that captivates users with its blend of sound and visual artistry. As we progress, our iterative testing and development process will be vital in refining and perfecting the system, ensuring a functionally impressive product and a visual spectacle, offering a unique and engaging musical experience.

7.3.6 Conclusion

In our project, we have successfully connected two microcontrollers using UART, setting a solid foundation for the advanced work we plan to undertake. This initial achievement marks a crucial step in building a system that includes the control of a photodiode, a polygon mirror, and a vertical adjustment mirror. The focus now shifts to the embedded programming required to integrate and operate these components effectively, showcasing the potential and sophistication of modern embedded systems in creating immersive experiences.

The connection of the two microcontrollers via UART ensures robust and reliable data transfer, which is essential for our project's complex tasks. This foundational work lays the groundwork for efficient collaboration between the microcontrollers, managing the intricate functionalities we will implement. Our next focus is on programming the photodiode in the laser projection unit. We aim to fine-tune the photodiode's sensitivity and response to accurately detect laser light and convert these signals into meaningful digital inputs. This task will involve detailed and innovative embedded programming to calibrate the photodiode precisely. Additionally, we are gearing up to tackle the control of the polygon mirror. We plan to program the motor driving this mirror to achieve specific rotational speeds and accurate positioning. This precise control is critical for creating visually striking patterns synchronizing with the synthesizer's sound, demonstrating our ability to integrate advanced software techniques with mechanical control.

Another key element we will focus on is the vertical adjustment mirror, which is crucial for accurately projecting the GUI. We will ensure that our programming facilitates smooth movement and precise positioning of this mirror, which is vital for the correct alignment and display of the GUI. This task will involve meticulous coding and rigorous testing to achieve precision. Our approach will emphasize real-time processing, synchronization, and effective system integration as we develop the embedded software for these components. Once we integrate and activate these components, we will prioritize modular programming, efficient signal processing, and extensive testing to ensure system reliability and performance.

In conclusion, having established the communication framework between the microcontrollers, we are excited to move into the more complex aspects of our project. We are committed to detailed and sophisticated programming for controlling the photodiode, polygon mirror, and vertical adjustment mirror. We aim to create an immersive synthesizer that functions effectively and engages users with an interactive audio-visual experience. This project is not just about technical achievements; it is about pushing the boundaries of interactive music technology and creating an instrument that blends precision with artistic expression.

8. System Fabrication and Prototyping

This section will detail the steps that will be taken when developing the prototypes for the project. Going forward, we will be using these prototypes to test the integration of the systems and find what works for the project as a whole. Things like the machine learning prototype to test for training and what gestures work, what we will do for host device implementation, and what the LPU prototype will be used for.

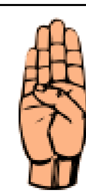
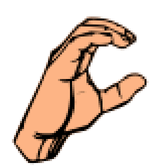
8.1 Machine Learning Prototype

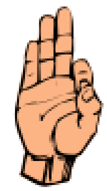
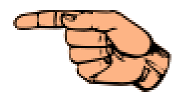
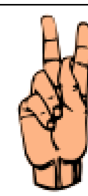
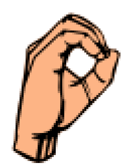
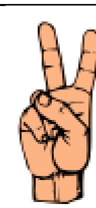
This section provides details about our machine learning prototyping process. In this section, the teams choice for hand gestures will be discussed, as well as how we implemented the machine learning model onto the host device as an initial proof of concept.

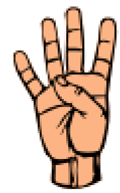
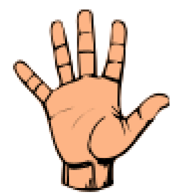
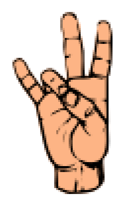
8.1.1 Hand Gestures

For each of our desired functions, the team had to assign an ASL hand gesture. After deliberation, the team decided on the gestures described in the table below for our initial prototype. After the machine learning model prototype is completed, we will initiate testing to determine if any of these gestures are too similar. In that case, changes can be made.

Table 21 - ASL Gesture Mapping

Desired Function	ASL Gesture	Gesture Example
Note A	Letter A	
Note B	Letter B	
Note C	Letter C	
Note D	Letter D	

Desired Function	ASL Gesture	Gesture Example
Note E	Letter E	
Note F	Letter F	
Note G	Letter G	
Accidental (Move up one half-step)	Letter K	
Octave	Letter O	
Key Signature	Letter S	
Major I Chord	Number 1	
Minor ii Chord	Number 2	

Desired Function	ASL Gesture	Gesture Example
Minor iii Chord	Number 3	
Major IV Chord	Number 4	
Major V Chord	Number 5	
Minor vi Chord	Number 6	
Dim. vii Chord	Number 7	

8.2 LPU Breadboard Prototype

Most of the testing will be done in the undergraduate lab of the CREOL building, where everything will be tested outside of the casing. We will be using a function generator and DC power supply for the prototype. Implementing the power supply and computers will come near the middle of the semester to test for any problems with integration. Getting the polygonal mirror to spin and finding the optimal placement of everything will help when developing and testing the housing for the LPU.

9. System Testing

This section breaks down our testing plan for the different pieces of our device. First, we'll discuss necessary testing for the PC. Next, we delve into similar testing procedures for the ESP32 MCU. Afterwards we'll go into detail about testing communication between our SBC and MCU. Finally, we'll discuss the various optical testing that will be required.

9.1 Machine Learning Testing

This section details our testing processes for the PC, and the parts and pieces that it is responsible for controlling. This consists of three sections. First, we will detail how we tested that our Pi 4 camera was functioning. Second, we'll delve into the process for testing our machine learning prototype. Finally, we discuss the steps we plan to take to test our audio output software, to ensure the correct tones are being played.

9.1.1 Raspberry Pi HQ Camera Test

The first thing the team did when we received our Raspberry Pi 4 and Pi Cam was write a basic Python script to confirm that the Pi and Pi cam worked properly. This script simply used cv2 to connect to the Pi cam and output the camera data onto the screen for the user. The screenshot below shows the output from the testing script.

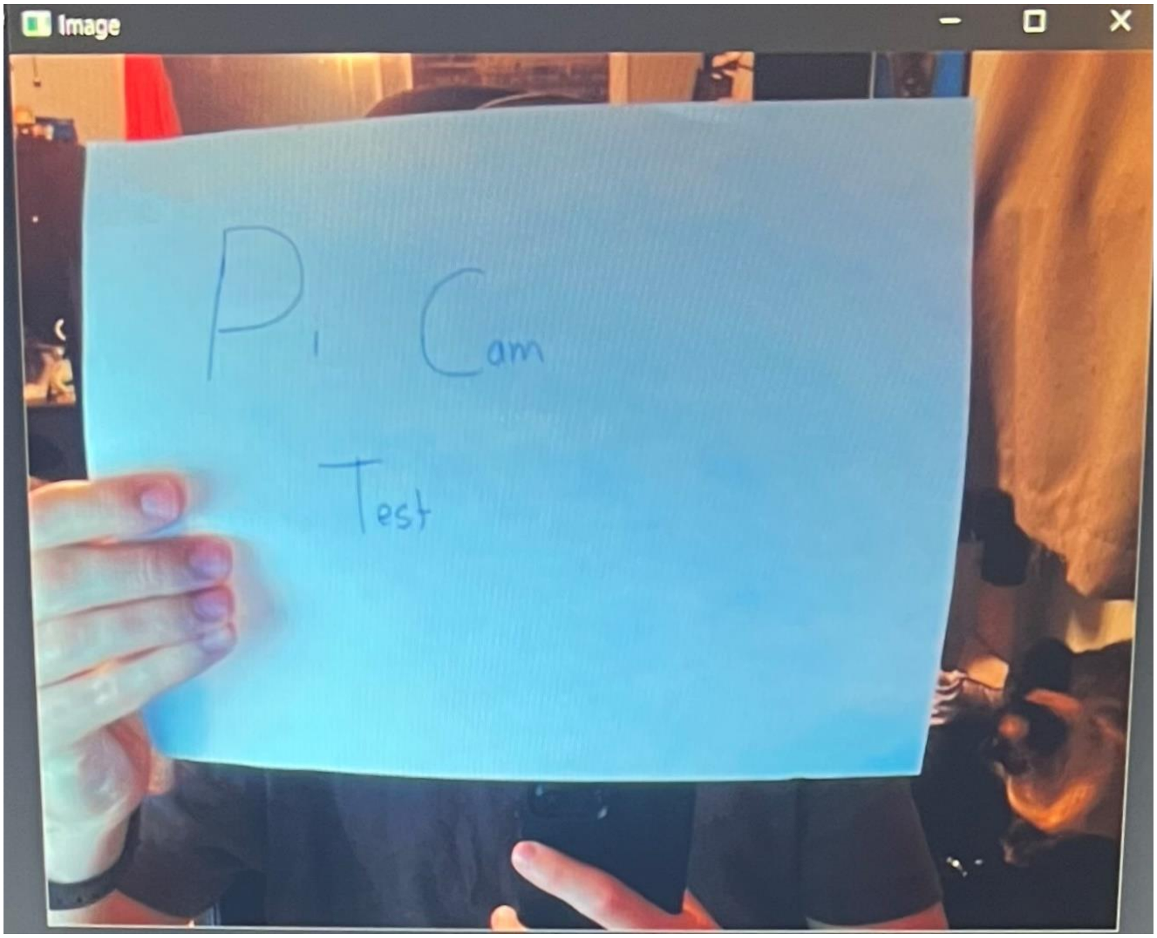


Figure 12 - Pi Cam Testing Output

9.1.2 Machine Learning Model Prototype Testing

After verifying that our Pi Cam worked, we were able to collect data for our machine learning model, as described in the design portion of this report. We used this initial round of data to create a prototype model. Again, the gestures we chose for our prototype are described in the section above. After training our prototype model using Teachable Machine and implementing it onto the PC, we were able to test the detection accuracy.

As expected, the first prototype is not perfect. We've discovered that a few of the gestures may be too similar to each other, and we've also decided we may need a more diverse database of images for the model to train with. Nonetheless, the model worked for most of the gestures, and provided us with a baseline. The figure below shows the output window of the model running and detecting the ASL letter 'A' hand gesture.

Figure 13 - Machine Learning Model Prototype Output Window

9.1.3 Audio Output Testing

Once the software for audio output is developed, we plan to write a testing script that will allow us to play any desired note without the requirement for a hand gesture. Additionally, we will use a portable tuner app on our phones to verify that the note we are attempting to synthesize is the correct note, and that it is in tune. For reference, these portable tuner apps are used by professional musicians world-wide to verify that their instruments are in tune, so this is a sophisticated enough testing process for our purposes.

9.2 ESP32-S3 Testing

Regarding testing the ESP32-S3 microcontroller, we have started the embedded testing process focusing on basic communication functionality. At this stage, our testing primarily involves sending commands from a computer through a command prompt and verifying that the ESP32-S3 accurately receives and processes these commands. This form of testing is foundational for more complex integration and functionality tests in the future stages of our project.

Our current testing method is simple but very effective for what we need. We connect the ESP32-S3 to a computer and use a serial communication interface of UART while using a USB cord. Through this setup, we input text commands into a command prompt or terminal window on the computer. These commands are then sent to the ESP32-S3. The core of our test involves the ESP32-S3 receiving these commands, processing them, and then sending a response back to the computer, essentially echoing the input. This process tests the ESP32-S3's ability to handle serial communication – a critical function, as serial communication is a fundamental method through which the ESP32-S3 will interact with our PC and the laser projection unit. This simple echo test serves the purpose of simulating our two boards communicating with each other. Firstly, it confirms that the ESP32-S3's serial communication channels function correctly. Secondly, it validates the basic firmware programming of the ESP32-S3, ensuring that it can receive, process, and respond to instructions. Finally, it provides a baseline assurance of the microcontroller's operational stability – a necessary step before we move on to more complex tasks.

The following form of testing we plan on doing will be more complex, so it simulates closer to the final product. We plan to test its wireless communication capabilities, which are essential for remote control and data transmission within our synthesizer system. The subsequent testing involving the board will be its communication test with the PC and being able to connect and communicate using UART. Additionally, we will test the ESP32-S3's interactions with other hardware components, such as the motors in the laser projection unit and various sensors. This will involve integration tests where the ESP32-S3 will be commanded to perform specific actions, and we will observe and verify the responses of the connected hardware. We will also perform a stress test to see our limitations with the ESP32-S3 board. For example, how much data can it receive from PC and process it precisely enough to perform the required actions?

Our current focus on basic communication testing through a command prompt lays a solid foundation for the following comprehensive testing. This step-by-step approach ensures we validate each aspect of the ESP32-S3's functionality, leading to a reliable synthesizer and laser projection system. As our project progresses, the data gained from these initial tests will guide us in making necessary adjustments and improvements, aligning with our goal of developing an interactive musical instrument.

9.3 PC and MCU Communication Demo

For our project, establishing communication between the PC and the ESP32-S3-S3 DevKit is critical to our system's functionality. To demonstrate and test this essential interaction, we conducted a communication demo between these critical components, which play pivotal roles in our system.

The PC is the command center of our system. It handles camera control, image processing, and audio output. The ESP32-S3-S3 DevKit, our chosen Microcontroller Unit (MCU), is responsible for managing real-time control tasks, such as operating sensors and actuators and managing the mechanics of the laser projection. The communication demo aimed to test the data exchange between the PC and the ESP32-S3-S3, ensuring they can effectively work together. For the demo, we will establish a serial communication link between the PC and the ESP32-S3-S3. This setup involved using the standard communication protocol UART, which is well-suited for this type of inter-device communication. The demo involved sending the ASCII character from the PCto the ESP32-S3-S3 and having the ESP32-S3-S3 execute these commands, which could include simple tasks like lighting up an LED or repeating it back in the console.

This communication demo is vital for multiple reasons. Firstly, it verifies the PC's capability to send accurate characters to the ESP32-S3-S3. Secondly, the ESP32-S3-S3 must reliably be able to receive the characters it is sent and execute specific commands to control the laser projection unit based on that. Another critical aspect assessed during the demo will be the latency in the communication between the PC and the ESP32-S3-S3. For our system, responsiveness is vital because we will be playing notes, and we want the user to feel like as soon as they make a movement, they get a note and not have it lag. Therefore, measuring and optimizing the latency was a significant part of the demo to ensure real-time responsiveness.

The communication demo between the PC and the ESP32-S3-S3 DevKit is the most crucial part of our system testing. If we can get this to work and to be precise and fast, our whole system will succeed. This demo has provided us with essential insights into the PC and the MCU, bringing us closer to achieving a cohesive and responsive synthesizer system. As we continue to develop our project, the data gained from this demo will be instrumental in adjusting our communication strategies and enhancing the system's overall functionality.

9.4 Optical Testing

Within each system are a numerous number of things to test for. The LPU requires testing of the electrical components, the distances and angles of the mirrors, laser diode, and photodiode, and the testing for the laser quality to see if the GUI will be clear on most surfaces.

9.4.1 LPU Testing

In our research for the Laser Projection Unit (LPU), we initially focused on verifying the microcontroller unit's (MCU) ability to accurately detect a high digital input signal from the photodiode when the laser beam intersects it. This step was crucial to ensure the precise detection of the laser's presence by the photodiode.

We conducted extensive testing by iterating through various circuit designs to minimize noise and latency. We aimed to guarantee that the MCU registered a high digital signal solely when the laser directly hit the photodiode. This process involved refining the circuitry to protect against electromagnetic interference and optimize the signal path, achieving a clear and immediate response in digital signal detection. Further, we evaluated the photocell frequency under different horizontal motor speeds. The testing ranges from 256 to 2048, and these experiments led us to determine that motor frequencies of 512 or 1024 were most effective for generating the graphical user interface (GUI). These frequencies maintained a stable and precise projection while not overstressing the mechanical components.

We also tested the vertical mirror's functionality to ensure it could produce multiple horizontal lines, enhancing the GUI's complexity. During this testing phase, we methodically examined the mirror's motion range and integration with the laser and photodiode systems to ensure synchronized operation across vertical and horizontal axes. Testing the mirrors was a priority due to their critical role and sourcing challenges. We meticulously verified the motor and polygonal mirror pinouts and those for the laser diode and photodiode. We also assessed reflectivity to guarantee optimal light path and image clarity. We addressed the LPU's weight distribution, which is vital for a device designed to be stable on a flat surface. We aimed to achieve a balanced design by centering the substantial weight of the polygonal and secondary mirrors to prevent tipping over. Moreover, we evaluated the system's responsiveness to signals from the MCU, ensuring that both the LPU and MCU systems operated flawlessly. This step helped isolate issues with integrating the systems rather than individual components.

During prototype development and final construction, we prioritized precision alignment. We frequently adjusted and reprinted 3D parts to accurately align the laser with the mirrors. Ensuring a tight fit for components like the mirror, laser diode, and photodiode was crucial to avoid misalignment or instability. Through this detailed testing and refinement, we aimed to establish a responsive and stable LPU crucial for the successful functioning of our gesture-controlled synthesizer. This effort underscores our commitment to enhancing the interactive musical experience through meticulous research and development.

9.4.2 Lens System Testing

The first thing we would ideally be testing with these two lenses we are using for the project, is to determine that these lenses aren't scuffed or marked in any sense. After that, we would ideally test the lens system out in a physical sense. This would mean that we would secure the lenses in the lens mounts and try to properly align the lenses in front of the raspberry camera. We would then place the lenses are varying distances to see that the effects that the lenses have on the camera match with the calculations we managed to figure out during the design section. If these testing's match with the calculations we already determined, then we should be set into integrating this with the other components of the project into the housing for all the parts. If our findings during testing don't seem to support our calculations, then we would ideally go back into determining what went wrong in the calculations and repeat the steps over again over the break to help with ensuring everything is set for final building and testing. To help with with verifying our work in regards to the lens system itself, we will be utilizing ZEMAX to further validate our findings. Since ZEMAX is a methodology in which we can use to run simulation software to orchestrate how lens system work, this would be a great option to use.

9.4.3 ZEMAX Lens System Simulation

To first understand this, we had to recreate our set up in regards to the lens system itself. We did this by first opening up Zemax and familiarizing ourselves with how the application works. We begin by looking at the lens data editor and seeing how we could layout our lenses in the system. We begin by establishing our layout for where the lenses would be fixated, understanding that we are using a plano-convex lens and bi-convex lens. The aperture is very crucial on where it is placed between in terms of the overall lens system itself. Since from earlier sections, we know that distancing is an important factor when considering total focal length, we know that f1 represents our bi-convex lens and f2 represents our plano convex lens, we know that to reach our focal length of 3.56mm we had to space the lenses as described previously. We know that in zemax, the object represents outwards towards where the user will primarily be, and the image surface represents where the light rays will converge into a singular point. From this, we can establish that the image surface in zemax represents the sensor of the raspberry camera, as we need the light rays to converge onto the sensor itself to formulate the image. The

STOP surface represents our physical aperture itself, since we understand this, we can play with the STOP surface to test potential ways it effects the image quality and sharpness throughout. As established before, our lens system works by having the user with his hand facing towards the system itself, and working by having the plano convex lens first, with its curved side facing inwards, followed by a distance of sorts, then the physical aperture itself, and finally the bi-convex lens before it reaches the CMOS sensor. Below we can were able to do this and display a layout of how the system will be organized and displayed.

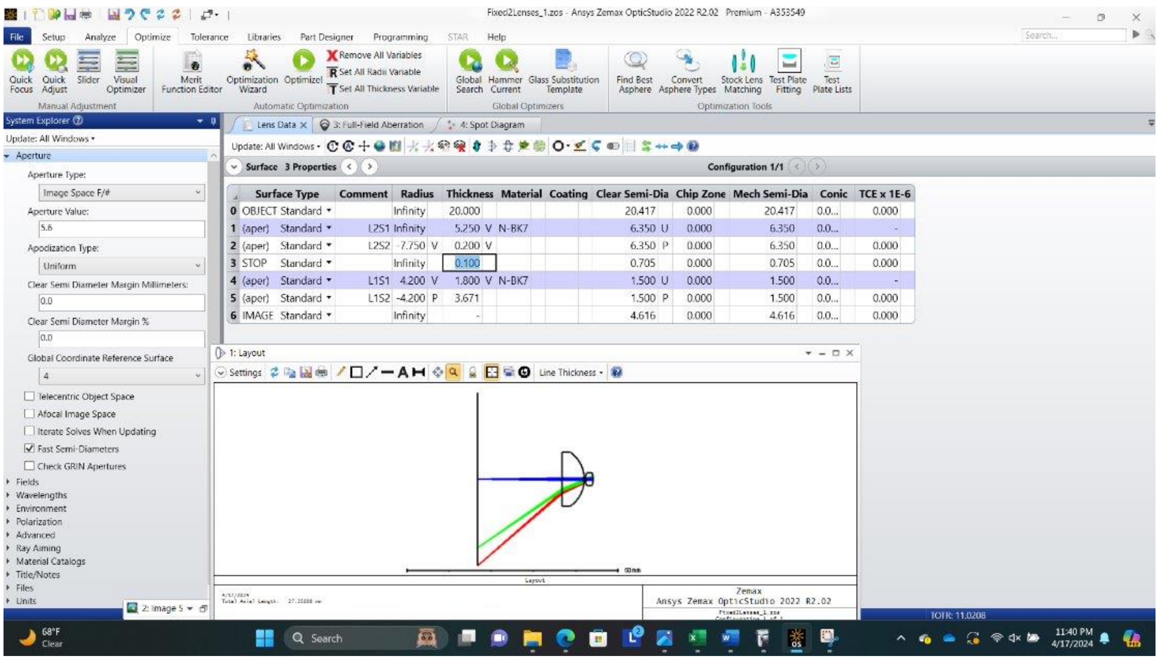


Figure 14 - ZEMAX Lens Layout

This further helps showcasing the lens system testing as we can validate the way Zemax shows that our initial calculations in regards to distancing between the lenses and how we can definitively place the aperture in between the lenses without fail.

9.4.4 ZEMAX Simulation for Lens System

In order to further understand the potential ways in which our lens system interacts with the CMOS sensor, we were able to use simulation tools in Zemax to help our testing. In zemax, there are tools that showcase how the image would look like using the image simulation tool, and a standard spot diagram to show how the light rays interact at different distances.

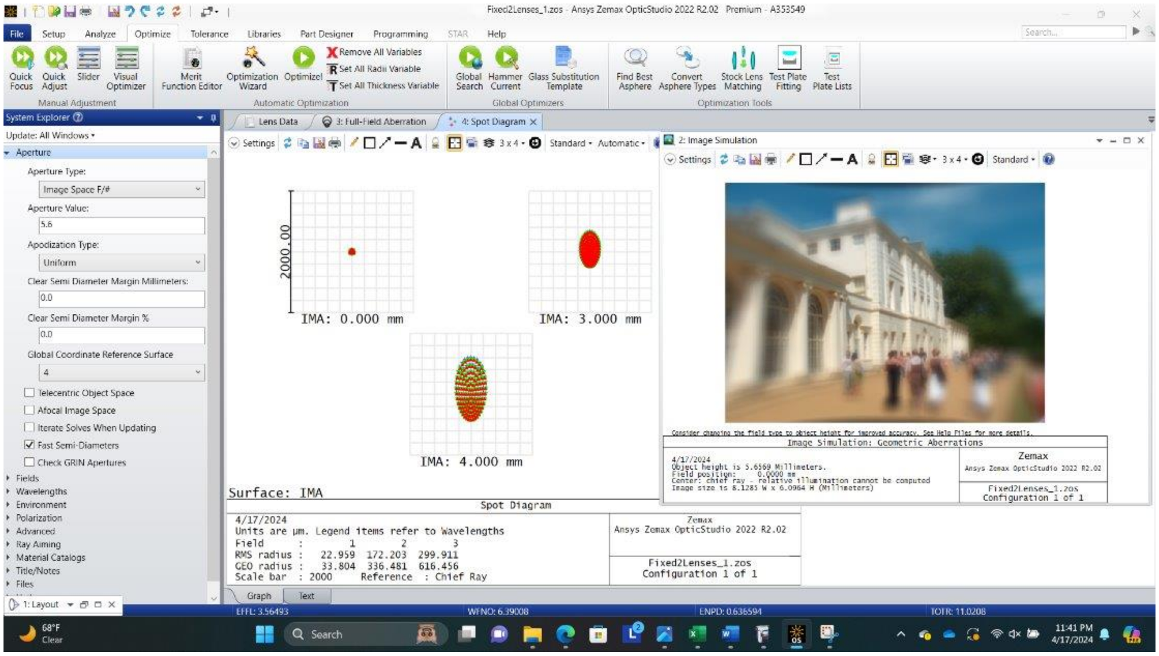


Figure 15 - Simulation of Lens System

As we can see from the image simulation, we notice that there will be in fact some vignetting and distortion occurring in the corners of the image itself. However, since we are focused on image clarity in the center of the image itself, this in fact will work with our goal of the project itself. As well, the standard spot diagram and image simulation both help show that when we increased the distance to 3 and 4 feet, that our simulation still holds and that the image will in fact still be clear in regards to when we increase the distances between the project and the hand of the user.

10. Administration

Our administration section discusses general aspects of our project, not directly related to research or development. The first section is project budgeting and financing, where we discuss how our project is being paid for, and our estimated costs for components. Afterwards, we discuss our milestones for SD1 and SD2. From there we delve into the various team member roles we have defined, both discussing what they are and who's assigned each. Finally, we'll go over the various project management tools we used for organization, including Discord, GitHub, and Trello.

10.1 Project Budget and Financing

Our budget will be self-financed, with the cost evenly split between the group members. Currently, our agreed-upon budget for this project is $400. This value was set up to be our engineering requirement. Currently, our estimated cost stands between $382 and $414, this averages out to $398. This value currently aligns within our preset budget, and we are hoping to not exceed this as the semester goes on. The items needed for the project are listed in the table below. For awareness, ‘various components,’ includes things such as mirrors and electronic components needed for the PCB. We understand that the costs for certain materials may change throughout the semester, so we will update our bill of materials accordingly.

Table 16 - Estimated Project Budgeting

Item:	Quantity:	Per Unit Price:	Estimated Cost:
Esp32 Dev Kit	1	$16.05	$16.05
PPN7PA12C1	1	$3.34	$3.34
Instapark DRM104-D003	1	$4.79	$4.79
Brother MFC-8220	1	$9.99	$9.99
Hamamatsu S5973	3	$24.95	$74.85
Alitove 24V 10A PSU	1	$19.59	$19.59
Raspberry Pi HQ Camera	1	$50.00	$50.00
Plano-Convex Lens	1	$29.99	$29.99
Bi-Convex Lens	1	$50.00	$50.00
24 to 5 DC/DC Converter	1	$12.50	$12.50

Item:	Quantity:	Per Unit Price:	Estimated Cost:
PCB Version 2	1	$10.44	$10.44
Digikey PCB Components	1	$25.61	$25.61
HiLetgo 5pc A4988	1	$10.19	$10.19
Sorand NEMA 17 Stepper Motor	1	$16.71	$16.71
Jetec 25pc Mini Size Acrylic Square Mirror	1	$9.39	$9.39
2x12 Lumber	1	$20.00	$20.00
Fasteners	1	$7.00	$7.00
Laser Diodes 10 pack	1	$7.44	$7.44
Standoff Set	1	$10.64	$10.64
Laser Diodes 10 pack	2	$6.99	$13.98
Arducam CSI to USB Adapter	1	$39.99	$39.99
Waveshare Portable Mini Tripod	1	$17.99	$17.99
Total:			$455.69

10.2 Project Milestones

This section details our estimated project milestones for senior design I and II. As shown in the tables below, the milestones are separated between SD1 and SD2, and are split into action items, timeframes, and descriptions.

10.2.1 Senior Design I

Table 17 - Senior Design I Milestones

Action Items	Timeframe	Description
Initial project idea	8/22/2023	Produce project idea for presentation to class.
Connecting with team	8/24/2023	Create a team based on what is needed and exchange information.
Brainstorming project ideas	8/25/2023-9/13/2023	Work with team to identify issues and develop engineering solutions and come to a final idea that is suitable for two semesters of work.
Responsibility assignment	9/11/2023	Discuss what is required of the project and determine who will be responsible for each required subsystem.
Divide and Conquer documents.	9/15/2023	Ten-page document due, including project description, requirements and specifications, block diagrams, budgeting, and milestones for the upcoming two semesters.
Research PCB design, lens design, camera sensors, lasers, light sources	9/15/2023	After deciding upon a final project, begin research of components that are readily available and figure out what will need to be designed.
Dr. Kar and Dr. Chan meeting	9/21/2023	Meet with Dr. Kar and Dr. Chan to present our idea, receiving feedback in the process.

Action Items	Timeframe	Description
Revising divide and conquer document	9/21/2023-9/29/2023	Utilize feedback from meeting and turn in revised ten-page document.
Mid-term demo of optical components	10/12/2023	Show working portion of the optical design from the project.
Sixty-page document	11/3/2023	Begin adding material to ten-page document, including research conducted, the design constraints, and a comparison of ChatGPT with other platforms.
Start testing hardware and software	11/3/2023	Begin assembly and testing of hardware along with the software provided by the CPEs in the group.
Sixty-page meeting	11/7/2023	Meet with Kar and Chan, discuss our research decisions and ask questions on design choices.
Hundred-page document	11/24/2023	Add material to sixty-page document, including the hardware and software design, breadboard testing, and overall schematic.
Final demo of optical components	11/23/2023	Bring in a more developed version of the project for the final demo in the discussion section.

Action Items	Timeframe	Description
120-page document	12/5/2023	Final version of the SD1 required document, with overall integration, PCB design, a further fleshed out and more detailed administration section, and the conclusion being the final additions.

10..2.2 Senior Design II

Table 18 - Senior Design II Milestones

Action Items	Timeframe	Description
PCB Send-Off	Week 5	Send our final PCB design off to a manufacturer to be printed
Final integration	Weeks 1-5	Construction of the final product.
Stretch goals	Weeks 6-8	Dedicated time to figure and add in stretch goals.
End testing	Weeks 8-14	Completion of testing to capture any deficiencies that arise during build that did not come up in research.
Final presentation	Week 15	Present and demo project to faculty and staff.

10.3 Project Member Contributions

This section describes how the team chose to lay out the project from an individual contribution standpoint, in order to effectively complete tasks in an organized fashion. We decided that this effort was best handled two-fold. First, a Work Distribution Table was developed to assign primary and secondary responsibilities for each of the major component blocks within the system block diagrams. Additionally, the team came up with a general set of roles for each of the major areas of development. Though these roles are not tightly fastened, it is important for each team member to know their priorities. Without these roles, facilitation of research and distribution of tasks was significantly more likely to become unbalanced, potentially creating a sub-par product and a hostile group environment. The roles, as well as a brief description of each, are laid out in a sub-section below.

10.3.1 Work Distribution Table

Table 19 - Work Distribution Table

Task	Primary Responsibility	Secondary Responsibility
Embedded Software Design	Christopher Jean	Tristan Barber
AI Image Detection Model Design	Tristan Barber	Christopher Jean
Laser Projection Unit Design	Jacob Goc	Christopher Jean
Camera and Lens Design	Christian Paredes	Jacob Goc
PCB Design	Tristan Barber	Christopher Jean
Misc Mechanical Design (Housing, Mounting Platform, Etc.)	Jacob Goc	Christian Paredes

10.3.2 Project Roles

- **Project Manager –** Responsible for the organization of team meetings, as well as the goals, objectives, and the delegation of tasks. Additionally, the project manager handles a large portion of non-technical project tasks, such as administrative tasks or management of the webpage. Finally, the project manager is responsible for bringing a defined agenda to each team meeting and taking meeting notes or otherwise assigning another group member to take notes.

- Role assignee(s): Tristan Barber

- **Software Engineer –** This role is assigned to members whose primary or secondary responsibilities include the development of software for the project. Members with this role are likely to be familiar with principles of effective software development.
  - Role assignee(s): Tristan Barber, Christopher Jean

- **Hardware Engineer (Electrical) –** This role is assigned to members who will have to research and/or design electrical hardware as a part of their individual components. Knowledge of circuit design and electronic components is required.
  - Role assignee(s): Christopher Jean, Jacob Goc, Tristan Barber

- **Hardware Engineer (Mechanical) –** This role is assigned to members who may have to design or construct mechanical components required for project completion. This is a minor role, as none of the members have a mechanical engineering background. Tasks may include researching or designing solutions to mounting components together onto a plate during final integration.
  - Role assignee(s): Christian Paredes, Jacob Goc

- **Optics Engineer –** As this is a CREOL group, consisting of two PSE students, our project is required to have a minimum of two major optical components. Thus, an Optics Engineer role was created to categorize our PSE members such that their optical design tasks are explicitly defined. For our project, the major optical components consist of a Laser Projection Unit, and an automatic Variable Focus Lens.
  - Role assignee(s): Jacob Goc, Christian Paredes

10.4 Project Management Tools

The development process of the gesture-controlled synthesizer was lengthy and required multiple forms of communication between group members. Additionally, ideas and progress had to be organized such that key elements didn't get lost in communication. This section discusses the various tools used by the team to promote effective communication and keep progress from coming to a halt.

10.4.1 Discord

When determining the optimal communication platform for our group's needs, we carefully evaluated a range of messaging applications compatible with various devices, offering the capability for well-structured discussions. Members of our team have previous experience with Discord, and we thought it was best fit for a communication hub. Discord is preferable as it goes beyond what a simple group-chat can do. Discord is set apart in the fact that users can create advanced chat rooms called “servers.” Within these servers, we can establish topic-specific channels, which facilitate focused conversations, collaborations, and content sharing without inundating a single group chat. At the beginning of Senior Design, the team shared stories about previous group projects, where information was lost, and instructions were forgotten. Our group aimed to solve this problem by choosing a platform that allowed us to create separate channels for storing important information.

While some may argue that this compartmentalization of discussions might create a sense of disconnection among group members, Discord unequivocally emerged as our top choice for team communication. Our decision to adopt Discord as our primary communication tool was influenced by various other factors aside from its versatility and channel segregation. Discord offers features like media uploads, post-message text editing, message pinning (to highlight important content and prevent it from being lost in the chatter), interactive bots for text-based interactions, and user-friendly voice channels for meetings and discussions.

Furthermore, Discord provides the option to assign special permissions to server members, empowering them to set up and moderate channels, thus enhancing our group's administrative capabilities. Notably, Discord's well-marketed and reliable software technology ensures low-latency voice and video communication at any time of day, guaranteeing that our team can convene and engage in clear and seamless communication through voice, text, or video as per our requirements.

10.4.2 Trello

With all of the pros of Discord, we still had difficulty with a key component of project management. Around the eighth week of the semester, the team realized that we needed a dedicated tool to help create, assign, track, and close tasks. Ultimately, the team implemented a Trello project management board to fit this need.

Trello is a popular project management and collaboration tool that uses a card-based system to help individuals and teams organize their tasks and projects. It provides a visual and intuitive way to create, prioritize, and track tasks on digital boards, which can represent different projects or workflows. Each task is represented as a "card" that can contain descriptions, due dates, checklists, attachments, and more. Users can move cards across customizable columns to reflect the progress of tasks. Trello is known for its simplicity and flexibility, making it a valuable tool for various use cases, from personal to professional project management. It fosters transparency and teamwork by enabling real-time collaboration and communication within cards and boards.

It should be noted that, although our task creation scheme is similar to the Agile methodology that is being adopted by many industry leaders, our team did not see fit to adopt the Agile workflow. We typically met one to two times a week and went over tasks we were working on as individuals and assigned members to new tasks created since the last meeting. With the various schedules between team members, including work, school, and extracurricular activities, the daily scrums that come with the Agile workflow would not have been effective.

10.4.3 Text Messaging and Cellular Phone Calls

Despite primarily using Discord for chat communication, the team also had a text message group-chat used for secondary communication. Beyond the text group-chat, individuals also used texting and cellular phone communication to address matters that did not necessarily apply to the whole group, or to attempt to receive a quick response from someone who wasn't in a place where they could communicate via Discord. For example, a few weekends during the development process, one of our team members was travelling and were not able to be present on Discord due to cellular service limitations. In this instance, that group member could communicate with others via the aforementioned text group-chat and take phone calls when available as required.

However, the significant limitations posed by a cellular communication did cause some difficulties, even when being used as secondary communication. For one, some members of the group has iPhone devices capable of using the built in iMessage feature. In a typical setting, this would allow for text messages to be sent over Wi-Fi. However, because some members of the team had Android phones, the primary text group-chat did not get to take advantage of the iMessage feature. This occasionally led to messages not being sent or delivered, even with Wi-Fi access. Luckily, this did not end up being a major issue, as Wi-Fi access allowed for communication via Discord anyways.

10.4.4 GitHub

GitHub is a web-based platform and service for version control and collaborative software development. It enables developers to host, manage, and share their code repositories, making it easier to work on projects with teams or individually. GitHub offers a range of features, including issue tracking, code review, and continuous integration, making it a central hub for software development, and fostering open-source collaboration. It has become an integral tool for developers and organizations worldwide to streamline the development process and maintain codebase integrity.

GitHub was primarily used by the two Computer Engineering group members, as a means to store code for the PC and the ESP32-S3. Though each Computer Engineering student had an individual primary responsibility, there were times when it was required to work outside of their module. In those situations, it was extremely helpful to pick up where the other left off. There are separate repositories set up for PC code and ESP32-S3 code, and an entire repository can be easily cloned as a local instance on a developer's working device. Once changes are made, they can be uploaded to the repository. The key aspect is that changes made do not override the history, so any different version of the software was accessible, even as features were continually implemented or improved.

11. Conclusion

Despite significant challenges, our team firmly believes that we have successfully met the requirements laid out for Senior Design I and II and successfully implemented our project. Additionally, the team has paid careful attention to design scope and individual workloads, to ensure that no single member will be overloaded throughout the course of Senior Design II. We have met all of the various deadlines for senior design throughout both Senior Design I and Senior Design II.

We had some struggles with part design and did suffer through an extensive redesign process during Senior Design II. Despite this, the team persevered, and did what needed to be done to get the project off the ground.

All in all, the team is immensely proud of the work we put in throughout the Senior Design pipeline and are thrilled to use the skills we've learned as we enter industry. We feel that this product could be scaled or modified to fit into many different markets, and that is an exciting idea on its own. Our ultimate goal throughout this process was to create a synthesizer that overcomes the issues of its predecessor, and we believe we made strides toward that end. In conclusion, our device combines various different features in software, hardware, and optics in order to create a successful user experience and optimize the current state of synthesizers on the market.

Appendix

A1 References

[1] *Bluetooth basics*. Bluetooth Basics - SparkFun Learn. (n.d.). https://learn.sparkfun.com/tutorials/bluetooth-basics/all#:~:text=century%20Danish%20monarch.-,How%20Bluetooth%20Works,differentiates%20it%20from%20other%20protocols.

[2] *ISO/IEC 9899:2018*. ISO. (2020, October 13). https://www.iso.org/standard/74528.html

[3] *Python enhancement proposals*. PEP 8 – Style Guide for Python Code. (n.d.). https://peps.python.org/pep-0008/

[4] *IPC Board Design Standards*. IPC International, Inc. (2021, October 5). https://www.ipc.org/ipc-board-design-standards

[5] *Dev Board*. Coral. (n.d.). https://coral.ai/products/dev-board/

[6] *USB accelerator*. Coral. (n.d.-b). https://coral.ai/products/accelerator/

[7] *Intel® neural compute stick 2*. Intel. https://www.intel.com/content/www/us/en/developer/articles/tool/neural-compute-stick.html

[8] *Sipeed Maix go suit (MAIX go + 2.8 inch LCD + OV2640 with M12 lens)*. Sipeed MAix GO Suit for RISC-V AI+IoT - Seeed Studio. (n.d.). https://www.seeedstudio.com/Sipeed-MAix-GO-Suit-for-RISC-V-AI-IoT-p-2874.html

[9] Raspberry Pi. (n.d.). *Buy A raspberry pi 4 model B*. https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

[10] *Nvidia Jetson Nano for edge AI applications and Education*. NVIDIA. (n.d.). https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/

[11] *Portenta H7*. Arduino Online Shop. (n.d.). https://store-usa.arduino.cc/products/portenta-h7?selectedStore=us

[12] Team, T. A. (n.d.). *Getting started with OpenMV and MicroPython*. Arduino Documentation. https://docs.arduino.cc/tutorials/portenta-h7/getting-started-openmv-micropython

[13] Team, T. A. (n.d.-b). *Setting up Portenta H7 for Arduino*. Arduino Documentation. https://docs.arduino.cc/tutorials/portenta-h7/setting-up-portenta

[14] Allan, A. (n.d.). *Benchmarking tensorflow lite on the new Raspberry pi 4, Model B*. Hackster.io. https://www.hackster.io/news/benchmarking-tensorflow-lite-on-the-new-raspberry-pi-4-model-b-3fd859d05b98

[15] *TinyML benchmark: Arduino Portenta H7 vs teensy 4.0 vs STM32 nucleo H743ZI2*. Eloquent Arduino Blog. (2021, January 23). https://eloquentauino.github.io/2021/01/tinyml-benchmark-arduino-

portenta-h7-vs-teensy-4-0-vs-stm32-nucleo-h743zi2/#tocbenchmark-2-rankings-per-classifier

[16] kumar, M. (2019, September 9). *Sound generation-python*. Medium. https://medium.com/py-bits/sound-generation-python-904e54f5398d

[17] YouTube. (2020, December 22). *Understanding audio jack switches and schematics*. YouTube. https://www.youtube.com/watch?v=GypQx7ZjBp4&ab_channel=CUIDevices

[18] Cutler, K. (2012, December 7). *Control Music Synthesizers With Gestures Through This Arduino-Based Saucer Called 'The UFO.'* TechCrunch. https://techcrunch.com/2012/12/07/ufo-midi-controller/

[19] *ANSI Z136 Standards.* LIA. (n.d.) https://www.lia.org/resources/laser-safety-information/laser-safety-standards/ansi-z136-standards

[20] Iliasam. (2019, February 4). *Making a DIY text laser projector*. Habr. https://habr.com/en/articles/438618/

[21] AI-Thinker team. (n.d.). ESP-32S Datasheet. Shenzhen Anxinke Technology.

[22] *The internet of things with ESP32*. The Internet of Things with ESP32. (n.d.). http://esp32.net/

[23] *S3 - the new flagship MCU from espressif*. ESP32. (n.d.). https://esp32s3.com/getting-started.html

[24] Teel, J. (2023, September 21). *Introduction to the ESP32 WIFI / bluetooth wireless microcontroller*. PREDICTABLE DESIGNS. https://predictabledesigns.com/introduction-to-the-esp32-wifi-bluetooth-wireless-microcontroller/

[25] I2C vs SPI VS UART – introduction and comparison of their similarities and differences. Total Phase Blog. (2023, May 17). https://www.totalphase.com/blog/2021/12/i2c-vs-spi-vs-uart-introduction-and-comparison-similarities-differences/

[26] Gardner, R. (n.d.). *Rags Int., Inc.. Digital Photography*. Circle of Confusion. http://www.rags-int-inc.com/phototechstuff/dof/

[27] Hollows, G., & James, N. (n.d.-a). *Imaging fundamentals*. Edmund Optics. https://www.edmundoptics.com/knowledge-center/application-notes/imaging/6-fundamental-parameters-of-an-imaging-system/

[28] Hollows, G., & James, N. (n.d.-b). *Understanding focal length and field of View*. Edmund Optics. https://www.edmundoptics.com/knowledge-center/application-notes/imaging/understanding-focal-length-and-field-of-view/

ChatGPT Declaration

We hereby declare that we have not copied more than 7 pages from the Large Language Model (LLM). We have utilized LLM for drafting and summarizing purposes.