UCF ECE SENIOR DESIGN PROJECT
ION

Summer 2018

Senior Design II

Group 11:

Dennis Gebken, CpE

David Jaffie, CpE

Mohammad Rizeq, EE

1.0 Executive Summary

Working at a desk for long hours can get boring and monotonous. Whether it be writing a paper for a high-school history class or working a forty-hour week as a software engineer, many people are overworked and quickly become unproductive when sitting at a desk for hours. The group worked to combat this with the introduction of Ion - an interactive personal robot-assistant. Ion brings fun and entertainment to the desktop to provide relief to hours of coding or researching for a paper. By being able to connect to the internet and a multitude of application, Ion provides entertainment by playing music, telling jokes, or even provide support in your tasks by searching the internet for any questions you may have. By doing this, Ion aims to integrate all your needs into a cute, amusing little robot who can be both your aide and your companion.

Voice-recognition assistants have taken storm on the market with huge companies such as Apple, Amazon, and Google participating. With IoT (Internet of Things) becoming a big phenomenon, it is becoming a necessity to have a way to connect everything and control them through a single application. For many companies, voice-recognition assistants are the answer to connect all these products that they could potentially put out onto the market and create a new craze around. Internet searches, weather, traffic, phone control, TV control, and house control can all be controlled through a single voice-recognition application.

Other than a personal project called Peeqo, which is described later in section 3.1.4, many of the voice-recognition personal assistants are all software-based smart-speakers and don't have a physical embodiment which can be more interactive with users. With inspiration from Peeqo, the group built a project that has that physical embodiment and can use that to its advantage to better portray emotion and reaction to the user. Rather than being a faceless app, Ion is more innovative and a step closer to the user by having face-tracking so that it seems like it is fully engaged in conversation and aware of your presence, voice recognition and machine learning technologies to have intelligent conversations, and servos motors to be able to move fluidly and portray some more human-like characteristics.

2.0 Project Description

The section serves to provide an understanding of the group's motivation and goals in choosing the project and in heading into development of the project. These motivations and goals helped shape the design of the project by drawing out the requirements that will also be specified in this section.

2.1 Project Motivation, Objectives, and Goals

The main motivation is to deliver a product that both serves a realistic purpose and demonstrates our abilities and the knowledge the group have gained through their education in the College of Electrical Engineering and Computer Science at UCF. While the group is aiming to complete the course, they also want to deliver a product that they are proud of, meets all their expectations, and can match up to the products and projects that inspired theirs. Those products and projects that inspired this project are given more attention in section 3.1.

Many of the goals for the project relate to what the group wanted to learn while working on it and what they want to deliver with their project. Coming into the process, the group didn't want to be constrained by keeping to topics that they already knew - they wanted to learn new skills in this process. A few of these new skills include UX design for the menu system, PCB design, mechanical engineering for the servos motors, 3D printing for the body and miscellaneous pieces, 3D modeling to design the device, and character design to create the personality of Ion. As you can see by the length of this list, the group didn't want to focus in one area or only stick to what they already knew - the group wanted to branch out and push themselves to deliver a great project.

2.2 Requirements Specifications

The requirements specifications included below are important to note because they severely influenced and constrained how the project was designed. By noting and keeping to these specifications, it was much easier to shape the project from the beginning and know what the group could work with when delivering their project

2.2.1 Size, Weight, and Dimensions

Most of the requirements relating to the size, weight, and dimensions of the project stem from the user experience expectations. The group's goal for this product was for it to be a small, interactive robot that can sit on a desk or countertop to be used by the customer. The product should be mobile enough to move from room to room or from home to office. The product should also be small enough to carry in one hand and not take up too much table space. This limits the size, weight, and dimensions requirements specifications regarding size, weight, and dimensions, which are specified below.

- The device shall weigh less than five (5) lbs.
- The device shall have dimensions less than 7" by 7" by 15".

2.2.2 Software Speed and Efficiency

The speed at which the software runs greatly affects user experience. The goal is a product that has a short start-up time and, when being used, will not show any signs of lag or loading. The product is very responsive to give the impression that it is actively interacting with the user and portray more of a human-like personality. As far as software efficiency, it is very important to limit errors and time spent 'thinking' as, again, the goal is to make it as human as possible and give the impression that it is an intelligent, sentient being. When a user interacts with the device, the device should be able to realistically interact back with the user through speaking, expressing emotion, and through tracking the user's face with sight. As such, this restricts the software speed and efficiency requirements specifications to the ones stated below.

- The device shall have a boot-up time of less than thirty (30) seconds.
- The device shall have an initial setup time of less than fifteen (15) minutes.
- The device shall respond to voice commands within two (2) seconds of being spoken to by the user.
- The device shall understand 80% of what is spoken to it.
- The device shall be able to fully understand and process the English language.
- The device shall be able to recognize faces 80% of the time under normal lighting conditions.

2.2.3 Power Consumption

Power consumption is one of the more important aspects of the requirements specifications especially since the device works wirelessly when being moved around the house, office, or wherever used. The requirements specifications for power consumption are as follows:

- The device shall be powered by a rechargeable battery.
- The device shall have an average power consumption of 50 watts or less.
- The device shall have a power switch to turn the device on or off.
- The device shall have a battery life of at least an hour.

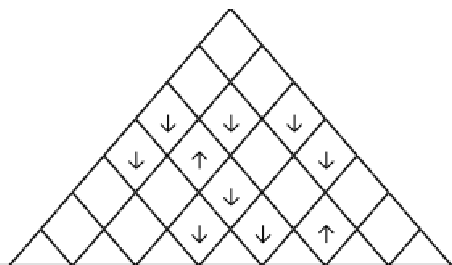
2.2.4 Costs

Users, competitor costs, and personal budget were the main concerns in relation to cost of the product. Through research, the group found that competitors had products available to purchase at an approximate cost of around $100-150. Considering the extra capabilities that the group's product has with using servos to move and having an interactive personality, the fact that the product was only a prototype, and the fact that the group was competing with major companies who have access to premium supply, they thought it was fair to aim for an approximate cost of $500. If they did move forward will full production of the device, they could hypothetically cut the production costs and, since they will know what components they will need and could streamline production, it should cut the cost significantly. With that in mind, the group aimed for a component cost of $120 as stated below. The group goes more in depth with the specific budget in section 9.2 of this document.

- The component cost of the device shall be less than $120.

2.3 House of Quality Analysis

In order to better understand the impacts of constraints on another, a "house of quality" analysis was conducted shown below in Figure 1. It shows that the constraints with the highest impacts were cost, power consumption, and display size. Display size is important because it directly affects user experience. A larger display not only makes it more enjoyable to view content, it also makes it easier to navigate the configuration menu because the device's setup relies on touch input. While the size of the display is somewhat restricted to the available space, the more impactful constraints are cost and power consumption. A larger display is more expensive and requires more power to run.



		Dimensions	GUI Functions	Power Consumption	Cost	Display Size	Setup Time	Voice recognition speed and accuracy
		-	+	-	-	+	-	+
Cost	-	↓		↓	↑↑	↓↓		↓
Weight	-	↓			↓			
Battery life	+	↓		↓↓	↓	↓↓		
Power consumption	-				↓	↓↓		↓
Sound quality recording and playback	+	↓		↓↓	↓			↑↑
CPU power	-		↓	↓↓	↓↓		↑	↑↑
Display size	+	↓↓	↑	↓↓	↓↓	↑↑	↑	
Targets for Engineering Requirements		< 7x7x15"	Change settings, user profiles, interaction with	< 50 watts average	< $120 component cost	maximize	less than 15 minutes	understand %80 with < 2 second reaction

Figure 1: House of Quality

Power consumption is such an impactful constraint because the device was intended to run on battery power. The more power is consumed, the shorter the battery life would be. To reduce power consumption, the most efficient components had to be chosen to perform their tasks according to the specifications. Cost correlates negatively with every specification because improved performance almost always relies on higher quality components which in turn are often more expensive. The impact of weight is not as high as other components because the device was intended to be stationary. If necessary, it was possible to sacrifice the weight constraint to buy a bigger battery to reduce the impact of higher power consumption, therefore allowing a less power-efficient processor to save cost to buy a larger display.

3.0 Project-Related Research

This section details the research that went into planning the development of the project. This includes similar projects and products that were on the market and helped influence the requirements and features of the project as well as research for the different features and technologies that went into the device.

3.1 Similar Projects and Products

The group drew a lot of inspiration from products already on the market or from personal projects found on the internet. Through research, knowledge was gained on what types of features to include in the product and how to innovate from those products already on the market. More in-depth analysis of the major products on the market and projects that influenced Ion are in the sections below.

3.1.1 Amazon Echo

Amazon’s Echo can be considered the leader among home control smart speakers with its voice assistant counterpart, Alexa. Amazon Echo, seen in Figure 2, is a smart speaker that also holds access to Amazon’s intelligent virtual assistant, Alexa, who can respond to voice commands, check the weather, play music through Bluetooth, and set alarms among a multitude of other things. This product is on the market for $100. With Amazon’s huge influence on the market and their strong marketing tactics, Amazon and Echo are a leader in their industry.

Figure 2: Amazon's Echo – Image Credits to WikiMedia

Housing an array of seven (7) microphones that give the Echo 360° omni-directional voice control, the Echo gives both a wide and long range for the user to interact through voice. It is connected to an outlet through an AC power adaptor, so it does have to stay put in one place and have an outlet available. Coming in at 3.3” by 3.3” by 9.3” and 2.3 lbs., it is a great size, but does not house an LCD with touch capability like Ion does. This had to be considered when planning for size and dimensions.

A very cool feature that the Amazon Echo holds is the ability to learn, in addition to its dozens of base features, skills from a library of tens of thousands of skills that are being added daily by both professionals and by the community. These skills can connect to almost any app or smart product, allowing the customer to order a pizza from Dominos or control your TV with dish. Amazon is very educating on how customers can develop their own skills, and this provides great comfort in knowing the support that Echo will keep receiving from both Amazon and from the community.

Amazon does have an Echo product, the Echo Show, that houses dual 2.0” speakers and a 7.0” screen but isn’t as popular and isn’t exactly what the group is going for in innovating their design. Their goal in providing a screen is to provide a way for the user to watch videos or have their skills be more visual-based, while the group wants to portray more emotion and interaction with the user in using the screen.

The group drew inspiration from the Echo, drawing its home and skill features such as checking weather, setting alarms, and responding to other voice commands with omni-directional voice control. The group looked to innovate this design by creating a more interactive visual interface in a robot that gives reactions and feedback instead of just being a hands-free smart speaker.

3.1.2 Apple HomePod

The HomePod, in Figure 3, is Apple’s attempt at producing a smart speaker like the Amazon Echo or Google Home. It’s a great concept but it came about two years late and it showed as they had poor sales and recently stopped production in April 2018, a mere three months after release in February 2018. Priced at almost $350, it is no wonder sales didn’t pick up. Housing seven tweeters and a four-inch subwoofer, as well as six microphones and a touchscreen on the top, it is a top-level smart speaker with Siri integration. The major downside is the price and the fact that it is two years late to doing exactly what Amazon and Google did. It has the basics you’d expect from an Apple smart speaker but didn’t bring anything to the table and was overall a disappointment for many consumers.

Figure 3: Apple HomePod – Image Credits to WikiMedia

3.1.3 Google Home

Google Home, seen in Figure 4, is Google's response to Amazon Echo. Released in late 2016, it aimed to compete with Amazon Echo, with an emphasis on home automation. Along with the normal features of music playback, weather, etc., Google Home has the ability to sync with home automation products including the Nest, SmartThings, and Philips Hue. Google comes with Google's version of Siri or Alexa, called Google Assistant. Google Assistant is a virtual personal assistant that uses natural language processing and machine learning technologies and carries many of the same features as the previous two competitors.

Figure 4: Google Home – Image Credits to WikiMedia

3.1.4 Peeqo

Peeqo, the GIF bot, was the main inspiration behind Ion and most-closely relates to the idea that was aimed for with this project. Peeqo, seen in Figure 5, is a personal project developed by Abhishek Singh for a thesis project at NYU. He had an idea of an interactive personal assistant which mainly communicated through gifs while helping the user complete tasks in a fun, cute way while providing some entertainment to brighten their day. Abhishek describes Peeqo as 'the love child of Amazon Echo and a Disney Character' which perfectly blends the roles of functionality with bringing amusement.

Figure 5: Peeqo – Image Credits to WikiMedia

Using a Raspberry Pi 3 as the brain and Arduino minis to control the 3D-printed plastic body, Peeqo can see the user using a built-in camera and play sounds and music through a built-in speaker. Also housing a crisp 5.6" 1080p LCD screen, Peeqo can draw gifs from a library on the internet to communicate its emotions and responses to the user.

On the software side of the device, the main program written to run on the Pi was written using Electron and uses HTML, CSS, and JS. To store user preferences, a server was written in Node. Peeqo uses Google Speech API for voice recognition and pulls from a library of GIFs to respond to the user. This feature of responding in GIFs and moving fluidly to emulate human emotion had a huge influence on the goals and designs of the project.

3.1.5 Product Comparison

Shown next page in Table 1 is the comparison done between all the four different devices in terms of price, size, weight, speakers, microphones, smart assistant, WiFi, and Bluetooth.

	Amazon Echo	Apple HomePod	Google Home	Peeqo
Price	$85	$350	$129	N/A
Size	9.3” x 3.3” by 3.3”	6.8” x 5.6” x 5.6”	5.6” x 3.8” x 3.8”	9” x 5” x 5”
Weight	2.35lb	5.5lb	1.05lb	
Speakers	One tweeter, one woofer	Seven tweeters, one 4-inch subwoofer	Two tweeters, two subwoofers	USB speaker
Microphones	Seven	Six	Two	
Smart Assistant	Amazon Alexa	Siri	Google Assistant	Google Speech
Wifi	Yes	Yes	Yes	Yes
Bluetooth	Yes	No	Yes	No

Table 1: Product Comparison

3.2 Relevant Technologies

This section aims to provide a description of the major technologies that went into the development of Ion and includes face detection and recognition, voice recognition, signal processing, and user-device interaction among others.

3.2.1 Face Detection and Recognition

One of the major features of Ion is the ability to detect and recognize faces.

3.2.1.1 Face Detection

Face detection is the ability to recognize whether there is a face or not. Ion is supposed to know when there is a human face within view. Because the only sensor that provides the necessary input to recognize a face is the camera, Ion relies on computer vision software for this task. A generic approach to detect faces is to recognize features (eyes, nose and mouth) and correlate their relative position with the average setup of a human face. This is done by taking a capture with a camera, optimizing the image by reducing noise and enhancing edges, and finally detecting the features based on templates from known examples shown below in Figure 6.

The math behind this process is complicated and would be a big undertaking for such a small part of the project but OpenCV, an open source program library, includes many image processing capabilities and provides a simple API to facilitate programming of this function. OpenCV was used not only because it facilitates programming, but also because the functions contained in the library have been optimized over the years. Ion relies on facial detection to turn towards the user, so any lag in the program makes Ion's movement more sluggish.

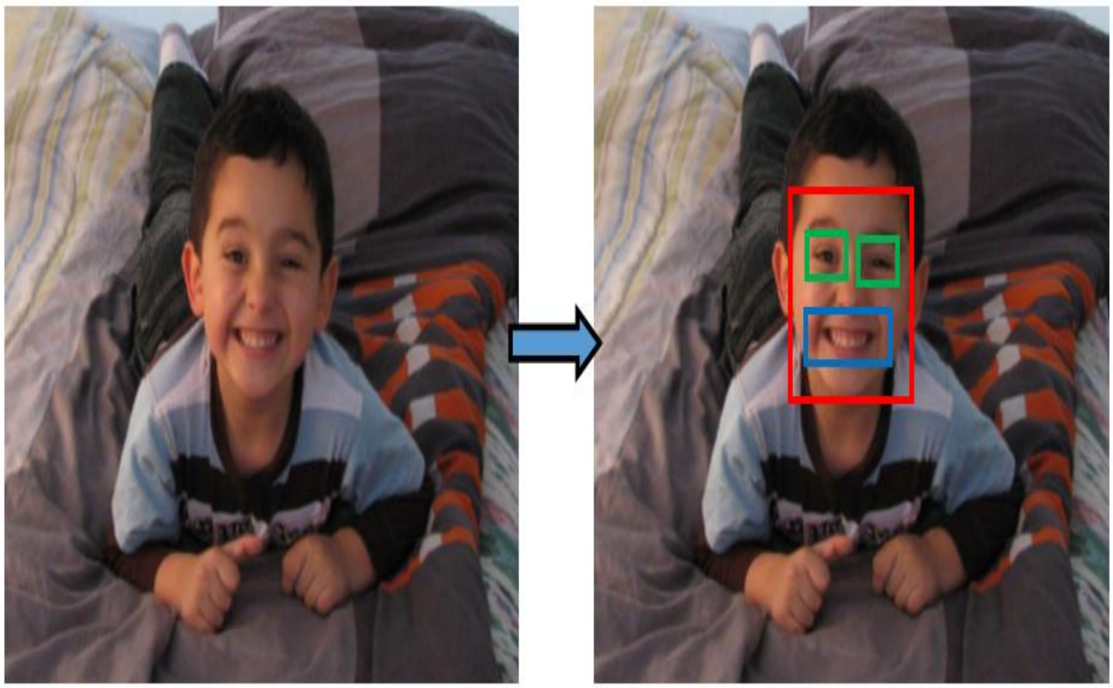


Figure 6: Facial detection. The modified photo on the right shows that the eyes (green boxes) and mouth (blue box) were detected as features and as a result, the face (red box) was detected. This picture is a mockup, simulating facial detection software.

3.2.1.2 Facial Recognition

Facial recognition is one step further from facial detection. Once a face has been detected by the facial detection software, a secondary process is planned to recognize the face. This is done by comparing the features and geometry of the recognized face to a user database of familiar faces. There are several techniques to do this including: Eigenfaces, Fisherfaces, and histograms. These techniques will be further discussed in the software design portion of this document. OpenCV the library that was used for this task. The scope of this project was too broad to develop a proprietary facial recognition algorithm.

3.2.2 Voice Recognition

Voice recognition technologies are starting to be included in a lot more products with the emergence of hands-free devices, virtual assistants, and home automation. Transforming analog sound waves into digital commands. Ion relies on the microphone to take in the sound waves which are translated to a digital form which can be understood by speech recognition software, Google Speech API in this case. A standard approach to voice recognition is to listen for keywords in a sentence and then guess the sentence or command with a certain amount of confidence. Higher-level systems utilize artificial intelligence and machine learning to better guess the command and to raise the confidence that it is correct. With this sentence, the software can then decide which command to start or task to complete.

3.2.3 Signal Processing

Another critical relevant technology related to the project would be signal processing, since the device is receiving and sending lots of signals. Those signals are processed accurately and fast enough in order to be understandable by both the user and the device while at the same time preventing wait time.

3.2.3.1 User-Device-User Interaction

One of the interactions requiring signal processing related to the device is the user-device-user interaction. When the user speaks, analog signals are produced, while the device, being a computer, can only understand and produce digital signals. To get around that, the analog signals produced by the user are processed and converted into digital signals for the device to understand them and act accordingly. After that, the device produces new digital signals, which sometimes are sent back to the user. So, the digital signals are then processed and converted into analog signals for the user to understand them.

There are six components involved in the process; a microphone to receive the analog signals from the user, an analog to digital converter (ADC) to convert the analog signals, a digital to analog converter (DAC) to convert the digital signals, an audio amplifier to amplify the analog signal to be more accurate, speakers to send out the analog signals to the user, and the CPU to process the signals. All those components are further explained in section 3.3 of the document. The user-device-user process is better illustrated in a figure as shown next page in Figure 7.

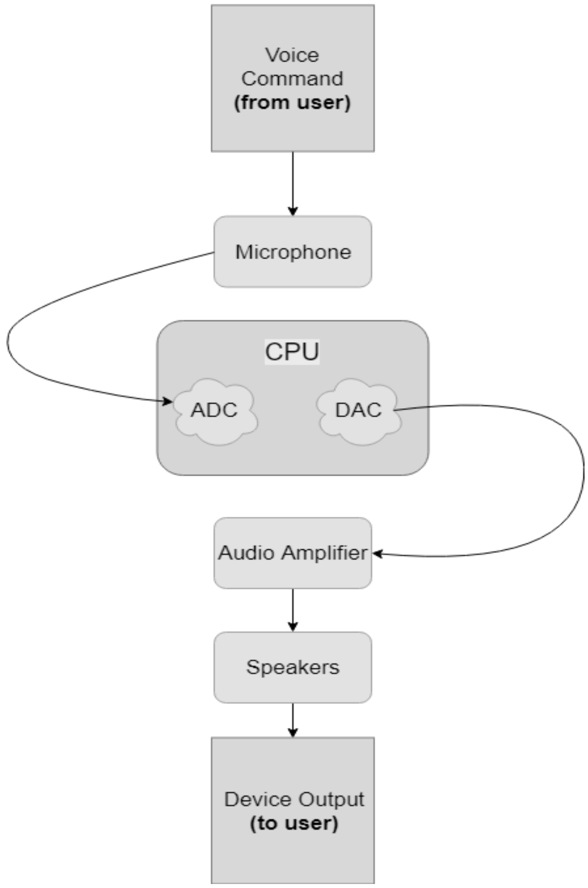


Figure 7: User-Device-User Interaction Flowchart. Example: User asks a question and the device answers back.

3.2.3.2 User-Device Interaction

The user-device interaction is a simpler process than the user-device-user interaction. The difference between the user-device and the user-device-user is that the device no longer has to convert its own digital signals to analog signals for the user. So, the device still has to convert the analog signals from the user into digital signals, process them, and then send the output digital signal to wherever appropriate based on the command. An example of that would be when the user issues the device a command but is not expecting an answer back.

Components involved in the user-device interaction include the microphone, an analog-to-digital converter, the CPU, and whatever other component that performs the task issued by the CPU output as a digital signal. This process is illustrated next page in Figure 8.

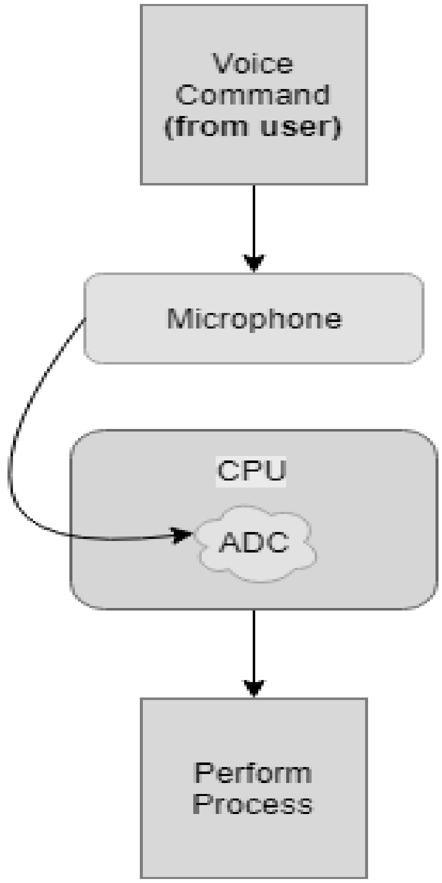


Figure 8: User-Device Interaction

3.2.4 Circuit Protection Device (CPD)

A circuit protection device was thought to be necessary to protect the battery and the device components from an electrical surge. Since the individual circuits are protected by voltage regulators, the only thing left is the battery. There are multiple types of CPDs, including circuit breakers, electronic fuses, diode arrays, and so on. Since the battery is the component that should be protected, the area of concern was having too much current drawn from it or device failure, and the best candidate for a CPD was an electronic fuse.

An electronic fuse is basically a resistor with a low resistance made of metal that melts in high temperature caused by a high amount of current. So, in a case of too much current draw from the battery, the fuse would absorb that overdrawn current and melt; therefore, breaking the circuit and shutting the device down before frying the device components and the battery altogether. The right electronic fuse would've been picked based on the desired maximum working voltage rating and current rating.

In the final design, a CPD was not used since the components did not require too much current and a battery was not used. So, a CPD was not required to protect our power source.

3.3 Strategic Components

3.3.1 Microcontrollers

3.3.1.1 PWM

Pulse width modulation (PWM), for the purpose of this project, means to create an electrical signal in the form of a square pulse with a fixed amplitude and frequency, but with a varying length of that pulse (duty cycle). It is used to provide the servos with a signal based on which they alter their position. In theory, a typical servo expects a 5 volt signal every 20 milliseconds and will adjust its position according to the length of that signal. The duration of the pulse can be anywhere between 1 and 2 milliseconds, where 1 ms sets the servo to 0 degrees, 2 ms set the servo to 180 degrees and anything in between those values adjust the servo continuously in a linear matter as shown in Figure 9.

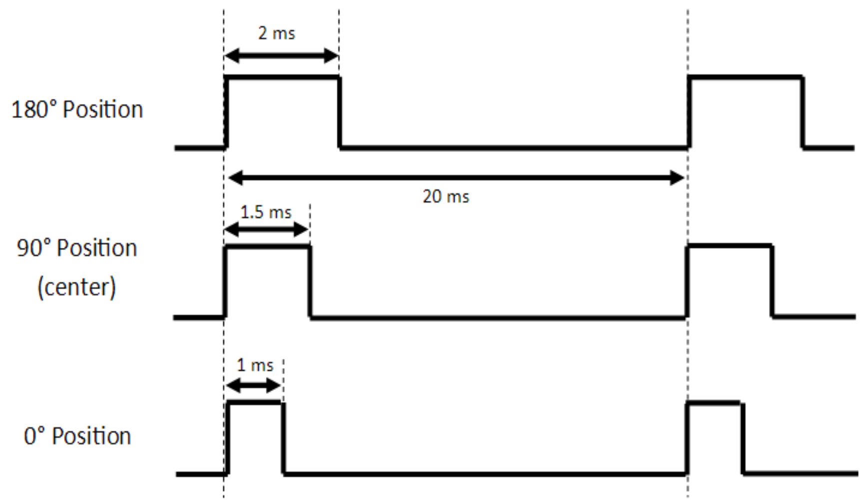


Figure 9: PWM signal for varying positions. The y-axis is the voltage of the signal and the x-axis is time.

While this describes a typical servo, there are other models that use a different timing scheme; and there are models that are able to rotate 360 degrees or more (continuous servos). Furthermore, in Ion's design the width of the pulse is not set to a specific position, but rather increased or decreased incrementally to mimic slow movement towards a direction. Figure 10 shows how PWM is used for this design. Servo selection for Ion is discussed in more detail in a later section.

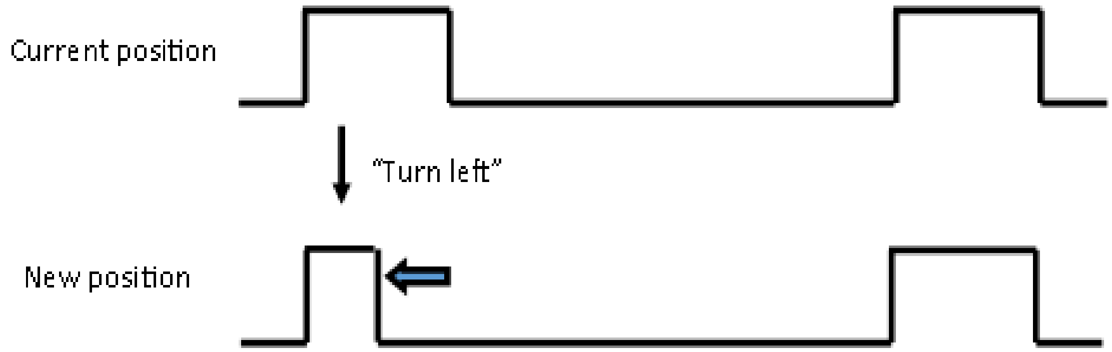


Figure 10: Adapted use of PWM signal to create a "movement" effect.

It is essential that the signal is as accurate or more accurate as the integrated controller of the servo motor. Accuracy refers to the length of the pulse that is generated. If continuous pulse of 1.5 ms is required to hold the servo in center position, but the pulse generated varies by a slight threshold that is picked up by the servo control, the motor will continuously adjust instead of remaining stationary. This unwanted motion not only looks bad in a finished design, it also uses up unnecessary energy. The generated pulse needs to be accurate. However, it does not have to be able to provide enough current to move the servo. This can be done by a separate power connection.

3.3.2 Processors

The design of Ion calls for a central processing unit (CPU) with sufficient processing power to handle facial detection, facial recognition, feedback for motor control, multimedia playback (audio and video), graphical user interface (GUI) with touchscreen input, and other general-purpose inputs and outputs. Preferably, this CPU would come with pulse width accurate modulation (PWM) for two separate outputs to drive two servos. The SBC should also support an operating system like Linux where all necessary coding libraries can be installed successfully. During the prototyping phase, the SBC will be used as generally available, but a final version of Ion would require it to be incorporated into the custom PCB. Therefore, the design of the SBC should also be open source, or it should be somehow possible to integrate it into a PCB design.

3.3.2.1 BeagleBone

The BeagleBone Black is an SBC based on the AM3358 ARM Cortex-A8 with a 1Ghz frequency. It has all necessary Peripherals and an ARM Cortex-A8 can be integrated into the final PCB design. However, with only one processing core running at a maximum frequency of 1Ghz, it is substantially less powerful than the Raspberry Pi 3 and might not suffice to run the required software without noticeable latency.

3.3.2.2 Asus Tinkerboard

The Asus Tinker Board SBC is based on an ARM RK3288 (Cortex A17) with 4 processing cores running at 1.8Ghz. All required peripherals are available, and it has more than enough processing power to handle the more intensive parts of Ion's processes such as facial recognition. The drawback of this design is the price at almost double the cost of the Raspberry Pi and the availability of the SoC it is based on. Initial internet searches did not provide a source for the necessary hardware to reconstruct this SBC as part of the final PCB.

3.3.2.3 PINE A64

The Pine 64 is an SBC based on the ARM Cortex A53 64-Bit processor with four cores running at 1152Mhz. It is comparable to the Raspberry Pi in almost every way. It's main difference is the availability of a low cost prototyping board for $35 that would include the chip used in the final PCB design ("Sopine A64", connected to PCB via SODIMM).

3.3.2.4 Raspberry Pi

The Raspberry Pi 3B is a popular Single Board Computer (SBC) with a low price of $35 and good functionality. It has a 1.2Ghz quad core Broadcom CPU that is able to handle the required workload, and most other necessary components required for prototyping. Another benefit of the Pi is a large support community and many helpful tutorials for implementations similar to its purpose for Ion. Its only drawback is that it only has one PWM output which, according to several user forums, lacks in precision, creating unwanted jitter in the servo. The Raspberry Pi 3B was used for prototyping and a final PCB implementation used the Raspberry Pi Compute Module 3 in Figure 11 below, which was attached to the PCB via a fixed SODIMM connector.

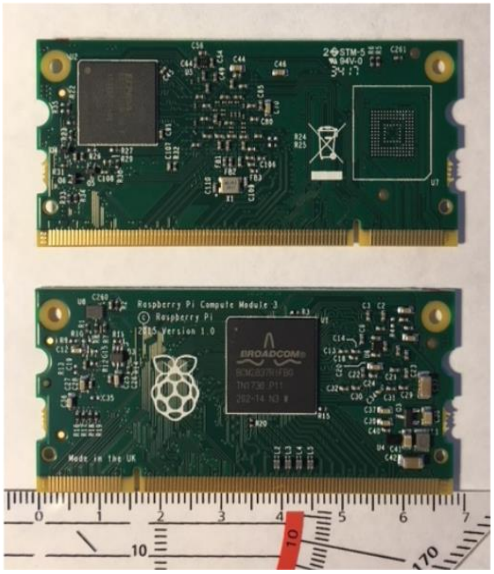


Figure 11: Raspberry Pi Compute Module 3L back and front with cm measure for scale

3.3.2.5 Overall Comparison

The table shown next page, Table 2, shows an overview of the features that were of importance in selecting the prototyping board. The two most plausible contenders were the

Raspberry Pi because of its abundance of tutorials and the Sopine A64 because of its similarities to the Raspberry Pi. Although the Sopine and Pi cost the same amount of money, the Sopine chip could've been used in the final design, reducing the overall cost of the project.

	Raspberry Pi	BeagleBoard	Asus Tinkerboard	Sopine A64
CPU/SOC	4-core 1.2Ghz	1-core 1.0Ghz	4-core 1.8Ghz	4-core 1.152Ghz
Price	$ 35.00	$ 49.00	$ 55.44	$ 35.00
Component Availability	Readily available	Readily available. CPU can be bought directly for $20.	Prototype board available on Amazon. Components for final design not found yet.	Readily available
Peripherals	Everything necessary, except 2nd PWM output. Also, PWM does not seem to be accurate enough.	All necessary peripherals available.	All necessary peripherals available.	Everything necessary, except 2nd PWM output.
Final Implementation	Connect Raspberry Compute Module 3 via fixed SODIMM connector to PCB.	SBC design to be used as reference to add CPU directly to final PCB.	Currently not possible because components have not yet been found.	Connect Sopine A64 via fixed SODIMM connector to PCB.
Notes	Abundant information and tutorials available on community forums and other websites.	Too expensive and the CPU is not powerful enough.	Too expensive but powerful. This might only become an option if it turns out that more processing power is needed.	Relatively new product but based on a known processor design. The processor used for prototyping can be used for the final product, thus reducing cost.

Table 2: Processor comparison

3.3.3 Audio Components

3.3.3.1 Mono-Audio Amplifier

The mono audio amplifier's main goal was to basically amplify the signals being output by the device in case they were too weak or not clear enough to be understood. The mono audio amplifier receives the output signals from the device, which should have already been processed and converted into analog signals then amplify them to be better understood by the user, then send the amplified signals through the speakers.

The audio amplifier the group decided to go with is a class-D amplifier called SparkFun Mono Audio Amp Breakout - TPA2005D1 for prototyping. This specific mono audio amplifier's features include a volume control potentiometer, where the user can control the volume output by reducing the input signal from 100% to 0%, and a shutdown pin, where the amplifier shuts down to save power when it is not being used. The shutdown pin shuts off the amplifier if the input voltage were either less than 2V or connected to ground or does nothing otherwise. The mono audio amplifier also has two gain resistors with a resistance of 150kΩ that provide a gain of 2 V/V. The gain resistors can then be changed to different values to further increase the gain of the amplifier. The gain is defined by the equation G=2*150kΩ/Ri, where G is the gain and Ri is the through-hole resistance that can be added to increase the gain. The smallest value for Ri that should be used is 15kΩ to generate a gain of 20. Those features were very convenient for the group's device since it saves power by shutting off the amplifier given that the device could run wirelessly by a battery. The amplifier can also be powered by an input voltage that ranges from 2.5V to 5.5V DC. [1]

Throughout the designing process, the group built their own amplifier with a low pass filter. But, in the final design no amplifier was used since a USB microphone was used.

3.3.3.2 Speakers

The speakers were mainly used to output analog signals to the user, whether they were replies from the device or music being played by it. The speaker used for this device was an 8-ohm speaker with a power rating of 0.5 watts. [2] So far only one speaker was enough for the device since it was already loud enough to be heard properly in an average-size room. The speaker is directly connected to the mono audio amplifier to receive the final, processed analog signals to be output to the user. In case the group decided that one speaker was not enough for the final completed device, another speaker could've also been added in parallel or in series with the first speaker.

If the two speakers would've been put in parallel with each other, they would've both shared the same amount of voltage from the source. The only downfall of that is that now the source supplying the voltage must supply greater current to be divided across the two parallel speakers which leads to more heat in the wires. But, in this case, the speakers each can take a maximum voltage of 2V and current of 0.25A. So, in total the source must supply a current of 0.5A which is not a problem at all temperature-wise.

If they would've been connected in series, which is possible as well, the voltage source must supply a greater voltage, meaning a stronger amplifier, but less current, meaning less heat. In total, the voltage source would be supplying a voltage of 4V and a current of 0.25A.

In conclusion, the better option was to go with speakers in parallel in case more speakers were to be needed, since the voltage being supplied stays the same and the current is too small to worry about heat.

In the final design, the 8-ohm speakers were not used to reduce PCB space by getting rid of the digital to analog converter and mini USB speakers were used. The mini USB speakers used include two speakers, 0.25W each and an impedance of 8-ohms.

3.3.3.3 Microphones

The microphone is mainly used to pick up analog signals from the user and then send them to the analog to digital converter, so the device can process the voice signals. The microphone used by the group for the device was a simple Electret Microphone, shown below in Figure 12, which was used for prototyping and the final design.

Figure 12: Electret Microphone

The microphone operates with a voltage range of 1VDC to 10VDC, a frequency range around 100Hz to 10KHz, and a maximum current of 0.5mA. [3] The issue with the microphone the group used is that it is cheap and poorly filtered, so it was susceptive to noise. To fix that, the microphone needed to be filtered to reduce noise. Once the noise was filtered out, the analog to digital converter could then work with a better signal. A low-pass filter was the best fit to filter out the noise since human voice is transmitted at low frequencies while noise is considered high frequency. The cutoff frequency for the filter was around 2KHz to include all human voice frequencies. After that, using an RC circuit to build the filter, the values for the resistor and capacitor was calculated using the cutoff frequency equation $f_c=1/2\pi RC=2KHz$ where there could be infinite combinations of resistor and capacitor values to be chosen from. The best combination was to pick low capacitor and resistor values for less size and cost.

In the final design, the electret microphone was not used for the same reason as the speakers. It reduced PCB size immensely by getting rid of the amplifier and low pass filter circuit, and reduced PCB signal noise. Also, no analog-to-digital conversion was required anymore since the mini USB microphone used already does that.

3.3.4 Camera

Ion has one camera that faces the user. The purpose of the camera is mainly to provide images used in facial detection and recognition. Because of this, the image resolution does not need to exceed 640x480 pixels. There are several options that were considered for the design of Ion: Pixy, generic webcams, Pi-compatible cameras.

3.3.4.1 Pixy Camera

CMUcam5 (aka. Pixy) is a camera produced by Charmed Labs with an embedded computer vision processor. It can detect objects and relay just basic information to a CPU while processing images on-board. This greatly reduces the strain on the CPU, making it possible to reduce the power specification of the processor, therefore possibly reducing its cost and power consumption. Because of the camera's specific purpose, it is also very fast and easy to program. However, after some basic research about the pixy cam, it was determined that it would not be a viable option because it couldn't detect or identify faces. Furthermore, at a cost of $69, the Pixy camera was more than double the cost of its alternatives.

3.3.4.2 Open MV M7 Camera

Like the Pixy camera, the Open MV camera has an embedded system that is specifically designed to provide the camera with computer vision. The main difference to the Pixy cam is that it supports facial detection. That, along with available I/O pins for servo control makes the camera a great match for the need that it had to fulfill. However, at $65 it was considerably more expensive than a "plain" camera without embedded features. Furthermore, part of the reason to include computer vision in this project was interest in the subject. While it is likely that this camera may be used in a refined version of the design, a self-designed algorithm was used to further the understanding gained of computer vision from EEL 4660 "Robotic Systems".

3.3.4.3 Webcams

The selection of available webcams was large. Cheaper models with sufficient capabilities start at $7. The added benefit of a webcam was that most models include a microphone, making the need for an external microphone obsolete. The drawback was that a USB port had to be added to the final PCB design in order to connect it to the SoC.

3.3.4.4 Pi-Compatible Cameras

A pi-compatible camera facilitated design greatly. Instead of the USB connection necessary for webcams, the Camera Serial Interface (CSI) bus was used. A basic model, capable of capturing 640x480 video at 90 frames per second was available for $10.99.

These cameras are also not limited specifically to the Raspberry Pi. Any SoC with a CSI bus had the ability to connect to it.

3.3.5 WiFi Components

3.3.5.1 Sparkfun ESP8266 Thing Dev Board

The Sparkfun ESP8266 Thing Dev Board is a WiFi microcontroller that can enable a device to connect to the internet wirelessly. It can be used for prototyping how the device would react once connected to the internet and what functions it can perform. The ESP8266 Thing is very convenient for testing since it is breadboard compatible, can be powered and programmed through a micro-B USB cable, has a built-in voltage regulator, a built-in USB-to-serial converter, an ON/OFF switch to conserve power, and the list goes on. The ESP8266 Thing is shown below in Figure 13. [4]

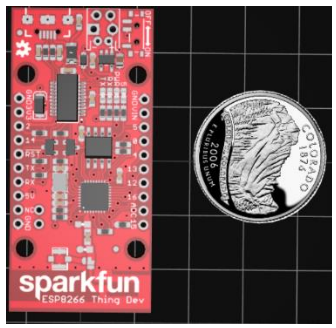


Figure 13: Sparkfun ESP8266 Thing Dev Board – Image credits to SparkfunTM ElectronicsSparkfun ESP8266 WiFi Module

The ESP8266 WiFi Module is a system-on-chip (SoC) that can give a microcontroller the ability to connect to the internet wirelessly. Since the ESP8266 Thing can only be used for prototyping and testing the device with internet, the ESP8266 could've been replaced by the ESP8266 SMT module, the ES-12S chip, and used in the final design then connected to the device microcontroller on the PCB. The ESP8266 WiFi Module was convenient for use since it has a built-in 1MB flash memory, power management units, and an integrated TCP/IP protocol stack. The module also works in almost all operating conditions. The only issue with the ESP8266 WiFi Module was that it is not breadboard friendly but would've had to have been converted if needed. The ESP8266 WiFi module is shown below in Figure 14. [5]

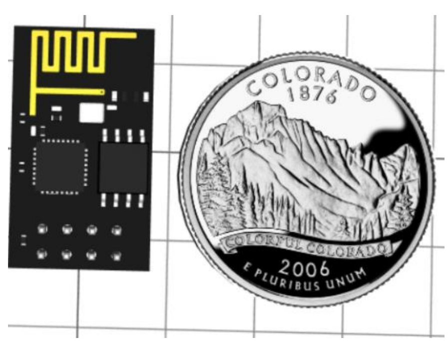


Figure 14: ESP8266 WiFi Module – Image credits to SparkfunTM Electronics

3.3.6 LCD

There was a long list of attributes that had to be kept in mind when looking for a LCD. The LCD was a huge chunk of the budget, being up to $100. The LCD's main purpose is to display visuals including the GUI, YouTube videos, and GIFS, so a decent size screen was needed - between 5" and 7" and having 480p resolution at a minimum. Those qualities were decent enough to display what the group needs to. The touch screen is the secondary interaction with the device, behind voice control, and had to have touchscreen capabilities so that the user could interact with it and navigate the menu system.

Of the three options below, the Pi Foundation was ultimately chosen. The group went with that choice because of its large size and resolution, multi-touch capabilities, and for its attached driver board. The large size and resolution were great to display what the group needed to, including the GUI interface and multimedia such as GIFs and videos. Having a decent size and resolution really sells the product as top notch. Having multi-touch capabilities was great for interaction with the GUI and it was very responsive and calibrated well. Most importantly, having the attached driver board saved the group a ton of time as they didn't have to put resources into designing one and it was virtually plug-and-play while developing with a Raspberry Pi. The Pi Foundation worked wonderfully.

3.3.6.1 Option 1: Pi Foundation 7" Display w/ Touchscreen

This screen, shown next page in Figures 15 and 16, is more than what was needed for the scope of this project. Being 7", it is a bit larger than what was needed and may have been hard to integrate into the design since the aim is to keep the body small. It is 800x480, the same resolution as the other options. It has touchscreen capability - 10-finger touch capability, in fact, which may be too much for the scope of this project. It also comes with an adaptor board that handled all power and conversions. It includes two connections, a power connector and ribbon cable. The cost of this LCD is $79.95. This option was viable but may have been too much for the scope of the project and budget may have been wasted on the extra features of this LCD which weren't necessary.

Figure 15: 7" LCD – Image Credits to Adafruit

Figure 16: 7" LCD – Image Credits to Adafruit

3.3.6.2 Option 2: 5" HDMI Display - 800x480 w/ Touchscreen and Mini Driver

This option, shown next page in Figures 17 and 18, is a technical downgrade, being only 480p and 5" in size. The price of this LCD is $79.95. It does include resistive touchscreen capabilities as well as a USB-powered driver board which would saveed a lot of time since one wouldn't need to be developed. It has a 40-pin connector with 24-bit color capability but also an HDMI connection which would be the best option to use, especially with the Raspberry Pi which supports HDMI. For power, this requires 5V and 500mA. This was a great option if the goal was to have a decent screen without putting resources into developing a driver.

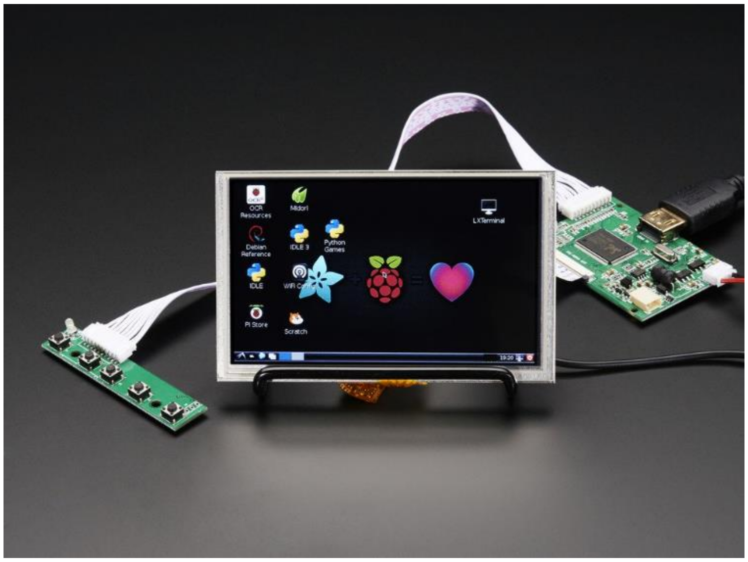


Figure 17: 5" LCD – Image Credits to Adafruit

Figure 18: 5" LCD – Image Credits to Adafruit

3.3.6.3 Option 3: 5.0" 40-pin TFT Display - 800x480 w/ Touchscreen

This option, shown next page in Figures 19 and 20, seemed like a good fit, having a 800x480 5.0" TFT display and touch capabilities. It is priced slightly higher at $39.95, which made it much more appealing, as the project budget was only a couple hundred dollars. Coming with an LED backlight and a resistive touchscreen overlay, this can perform all the functions that would be asked of it, including having the GUI interface and playing videos and GIFs. The connectors include a 40-pin which has 24-bit color capability. Some issues for this display includes having a high-refresh rate which requires a powerful processor which don't come with many microcontrollers. The backlight also requires a constant-current mode boost converter which go as high as 24V and would've been difficult to integrate into the design of the project. All these issues would've probably called for buying their recommended driver board that goes hand-in-hand with this LCD. If this option would've been chosen, the group would've most likely had to design their own driver board which would've been a huge undertaking and take a lot of resources and time. The final LCD comparison is shown in Table 3 next page.

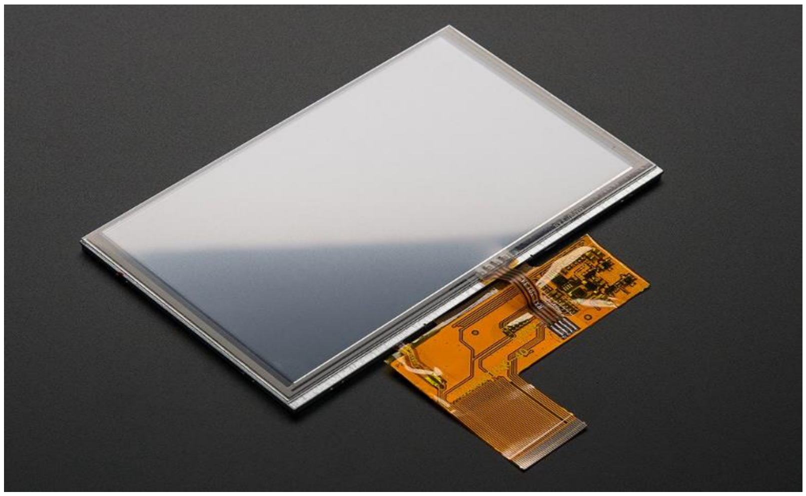


Figure 19: 5" TFT – Image Credits to Adafruit

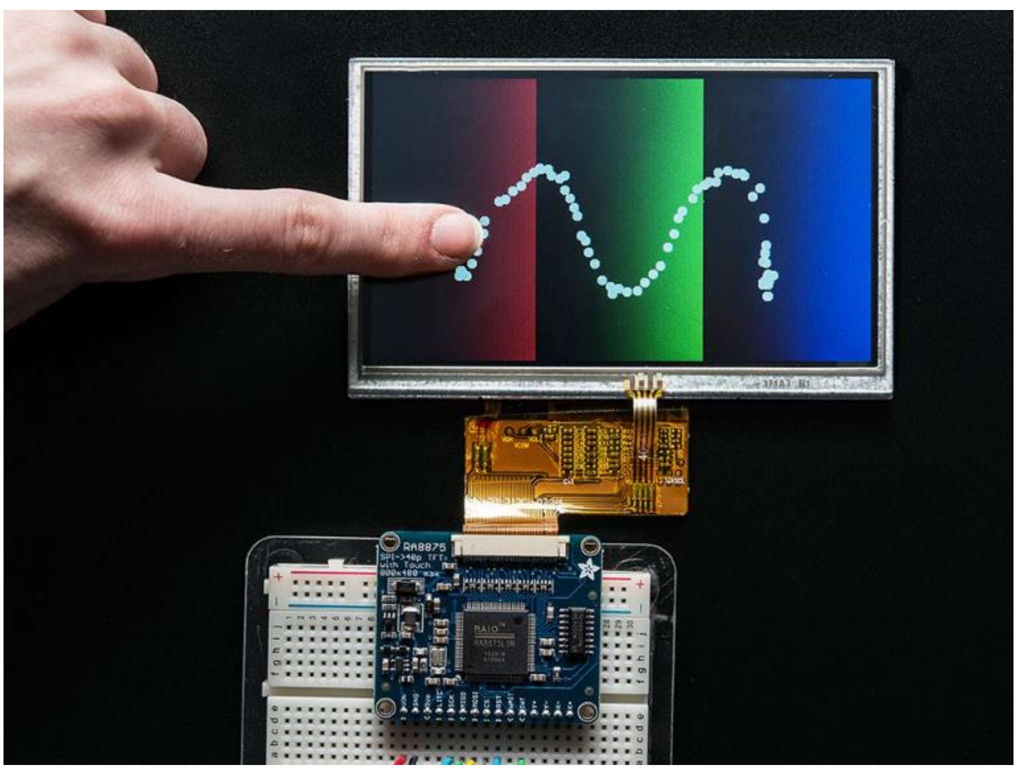


Figure 20: 5" TFT – Image Credits to Adafruit

	7” Pi Foundation LCD	5” LCD	5” TFT
Size	7”	5”	5”
Resolution	800x480	800x480	800x480
Display Type	LCD	LCD	TFT
Driver board?	Yes	Yes	No
Cost	$79.95	$79.95	$39.95
Connectors	Ribbon (DSI)	HDMI	Ribbon (DSI)
Touch capability	Capacitive	Resistive	Resistive

Table 3: LCD Screen Comparison

3.3.7 Motors

Ion was supposed to have four degrees of freedom, or two axes of movement. that required at least two motors that needed to be controlled by the CPU and powered by the battery. The required torque generated by the motors depended heavily on the design of Ion’s casing, what kind of bearings, gears, and weight they need to move. Since the idea is to rotate Ion to face the user, it is necessary when, how fast, in which direction, and how far the motor was supposed to turn. Initial research showed two types of motors that would be able to provide the necessary torque and precision: DC motors and servos.

3.3.7.1 DC Motors

A standard DC motor generates torque from DC power. There are two main types: Brushed and Brushless. A brushed DC motor is cheaper to construct and simpler to control. In the case of Ion, some sort of feedback was required to position the motor according to the signal from the CPU (contributors, 2018). One of the options was to use a rotary encoder to translate the amount of movement to a position. This type of closed loop system is also referred to as a servomotor (contributors, 2018).

3.3.7.2 Servomotors

As described in the previous paragraph, servo motors are electrical motors that are capable of moving to a specific position via the help of a control system. There are two types of servos that were useful for Ion’s design: Positional rotation, and continuous rotation servos. Positional rotation servos can rotate on one axis by about 180 degrees. The position is dictated by a PWM signal as described in detail in another section of this document. Continuous servos can rotate indefinitely in either direction at a varying speed.

3.3.7.3 Conclusion

Servo motors were used in the final implementation. Both types of servo motors described here would have worked in the design. A positional servo was used because it was not deemed necessary to give Ion more than a 180-degree field of motion. Movement was also restricted to left-right to make the body design less complicated.

3.3.8 Light Emitting Diodes (LEDs)

The main purpose for the LEDs were to make the device express itself even further with different lighting. There were two options for the RGB LEDs; clear and diffused LEDs. They were both similar in price even though usually diffused is cheaper, but clear provides a much better lighting than diffused. One LED was not enough, so multiple LEDs were used but programmed to work in unison. The group used RGB clear LEDs to reduce the number of LEDs, compared to using multiple LEDs that produce one color each, while still achieving the same result. The LEDs used have an internal frequency of 800KHz and require a power supply of 4.5V-6V DC. Shown below in Table 4, the different LED colors are illustrated along with what they are supposed to mean.

Color	Meaning
Yellow	The device is listening to the user.
Green	The device understood the command.
Red	The device did not understand the command.
Orange	The device is currently powering on.

Table 4: LED Colors and Functions

In the final design, nonprogrammable RGB LEDs were used instead of the programmable ones since they were much easier to use. The color functions changed as well; Red means the device is searching for a face, Green means the device detected a face.

3.3.9 Batteries

Picking the right battery was the tricky part, since batteries can be found in a very large range of options; chargeable or nonchargeable, different nominal voltage ratings, different capacities, and the list goes on. The one certain thing the group decided to go with is that the battery must be a rechargeable battery for reasons including economic and environmental. The economical side of it would be that the user does not have to worry about replacing the batteries as often by having to buy new ones; so it is cheaper for the user. The environmental side of it is that the batteries will be reused many times rather than discarding the batteries after every use; better for the environment.

Next, the nominal voltage rating was determined by looking at the components being used for the device and determining what is the maximum voltage required for each component, and the highest determined the nominal voltage for the battery. Then, a voltage bus was created to supply the DC voltage to all the circuits in the device. But, a voltage regulator was needed to regulate the amount of voltage being supplied to each circuit at a constant rate. Since the nominal voltage rating was low based on the components used for the device, cost wasn't too much of an issue.

After that, it was important to determine the energy capacity of the battery; too much current being drawn from the battery could've caused damage to it or to the circuits. To prevent damage to the battery, a circuit protection device had to be implemented, just in case. So, in conclusion, it was important to make sure the battery provided the right amount or excess amounts of current.

Finally, another parameter that had to be considered is charge cycles; the amount of times the battery can be recharged and discharged before being discarded. Having an extremely low amount of recharge cycles would not be too smart since it defeats the original purpose of picking a rechargeable battery for the device. That means, a decent range of recharge cycles was around 500-1,000 cycles.

Now that the parameters were defined, the next thing to look at was the different types of rechargeable batteries, illustrated below in Table 5, along with their advantages and disadvantages to decide which was the perfect fit for the project.

Battery Type	Advantages	Disadvantages
Nickel-Metal Hydride (NiMH)	• Recharge cycles between 150-500. • Delivers current at a more constant rate. • Has a self-discharge rate of only 1% per day. • It is best when it is used frequently.	• If idle, self-discharge rate can increase up to 5% per day. • Current rate supply can drop by 15% after 100+ charges. • Batteries should be handled properly, or performance might
Nickel Cadmium (NiCd)	• Up to 1,500 recharge cycles. • Good for high temperatures and rugged. • Steady voltage throughout a charge cycle.	• Cadmium is a highly toxic component. • Low energy capacity compared to NiMH.
Lithium-Ion (Li-ion)	• Recharge cycles between 500-1,000+. • Lowest self-discharge rate between rechargeable batteries.	• Highly impacted by aging. • Can be expensive.

Table 5: Different Battery Types Comparison

In the final design no battery was used. Since the device would be sitting on a desk, it could be directly connected to a power source. At first, the power source was changed to an AC wall adapter that gets connected to a DC power jack on the PCB. In the final design, the power source used was a micro-USB port on the PCB. The reason behind changing the power source was also related to simplicity and not needing an extra voltage regulator to keep the battery output voltage constant.

3.3.10 Power Switch

Since almost all electronic devices usually have a manual power switch, the group figured that one was needed for the device to save power when the device was not in use or for an emergency shutdown. Given that the battery would supply a voltage around 3.3V-5V and a current around 5000+mAh, it was a bit tricky to pick the right power switch that can handle that much current without getting fried. The power switch's main goal was to break the circuit connecting the battery to the device when needed. The best fit the group found was an ON/OFF SPST rocker switch that has a blue LED rated at 12V and 16A, shown below in Figure 21. It is 20.2mm in diameter and could also be run at a voltage rating of 5V. The blue LED turns on to signal that the device is on or it is off otherwise. [6]

Figure 21: Rocker Switch w/ Blue LED

In the final design, no power switch was used. Since it was unnecessary and complicated PCB connections even further, the group decided to use a digital power switch that can be done through the device instead of a physical switch.

3.3.11 MicroSD Card Reader

The microSD card reader was mounted on top of the PCB to hold a microSD card to hold the device's operating system along with on-board storage. However, it was complicated to find the right microSD card reader, so a breakout board was used for prototyping, then using its schematics, the group created their own microSD card reader mounted on their PCB. The microSD card reader schematic was created by following a microSD card breakout board schematic found on *Adafruit*. Some of the components, like the internal voltage regulator, were removed since they are not needed, and only required components were kept. The microSD card reader requires a 5V DC input supply along with 100mA of current. [7]

3.3.12 Digital-to-Analog Converter (DAC)

The digital-to-analog converter is required to convert the digital signals being output by the microprocessor to analog signals for the mono audio amplifier and speaker. There were lots of different options for DACs, many of which included through-hole components. The through-hole components were decent but looked unprofessional on a PCB. So, the group researched DAC's that could be mounted on top of the PCB without any through-holes. The best fit that the group found was the MCP4725 12-bit DAC provided by *Adafruit*. Once again, a breakout board could've been used for prototyping then following its schematics, the group was able to come up with their own DAC using the MCP4725 but without the board. [8]

As stated previously, no DAC was used in the final design since the microphone was switched out for a mini USB microphone that does not require a DAC anymore.

3.4 Possible Architectures

3.4.1 Overall System Design

3.4.1.1 Powerful vs. Simple CPU

The selection of the CPU depends on the overall design of the architecture. The two extremes considered were:

1. A powerful CPU that handles everything to control passive components.
2. A simple CPU that ties together a system of smart components.

3.4.1.1.1 Powerful CPU

In this type of architecture, the CPU must calculate, control and time everything including:

- Activate, dim and power LEDs
- Generate PWMs for servos
- Calculate positioning of servos based on feedback
- Calculate feedback for servos based on processed images from camera
- Process images from camera (facial detection and recognition)
- Power and control camera
- Process audio received from microphone
- Generate audio signal for amplifier/speakers
- Provide power and data to LCD
- Receive and process touch input from LCD
- Run OS and main program (GUI)

See Figure 22 next page for a visual representation of this system. Arrows show the flow of data and power.

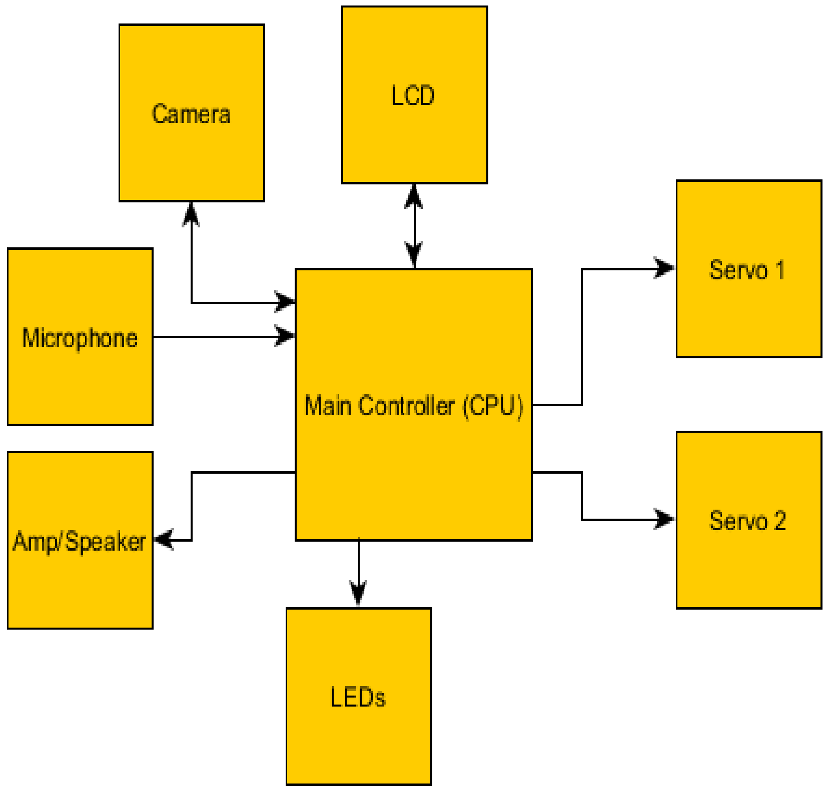


Figure 22: Architecture based on a powerful CPU

The benefit of this architecture was the low cost of components because "smart" components come at a premium price. The drawbacks would be the price of the CPU and the complexity of the main program. The main routine of the program would have to account for all components on top of the image processing. This may have lead to lag problems such as a delayed response of the servos because the processor is busy processing images.

3.4.1.1.2 Simple CPU

The opposite of the architecture described above would be a simple CPU that only needs to run the OS and GUI (from the list of functions outlined in the powerful CPU section above). All other processing would be done by the components themselves. For example, the camera would have an embedded system that automatically focuses and adjusts its gain. Images would be processed and only the necessary data, such as position of users face relative to image center, would be transmitted to the CPU. See Figure 23 next page for a visual representation of this system. Arrows show the flow of data and power. The obvious benefit of this architecture is the relief on the CPU, which would only have to act as a scheduler apart from running the OS and GUI. The main drawbacks would be the price of "smart" components and their availability. Currently, a cost effective (project budget is $500 for the whole system) face-recognition camera for small systems does not exist on the market.

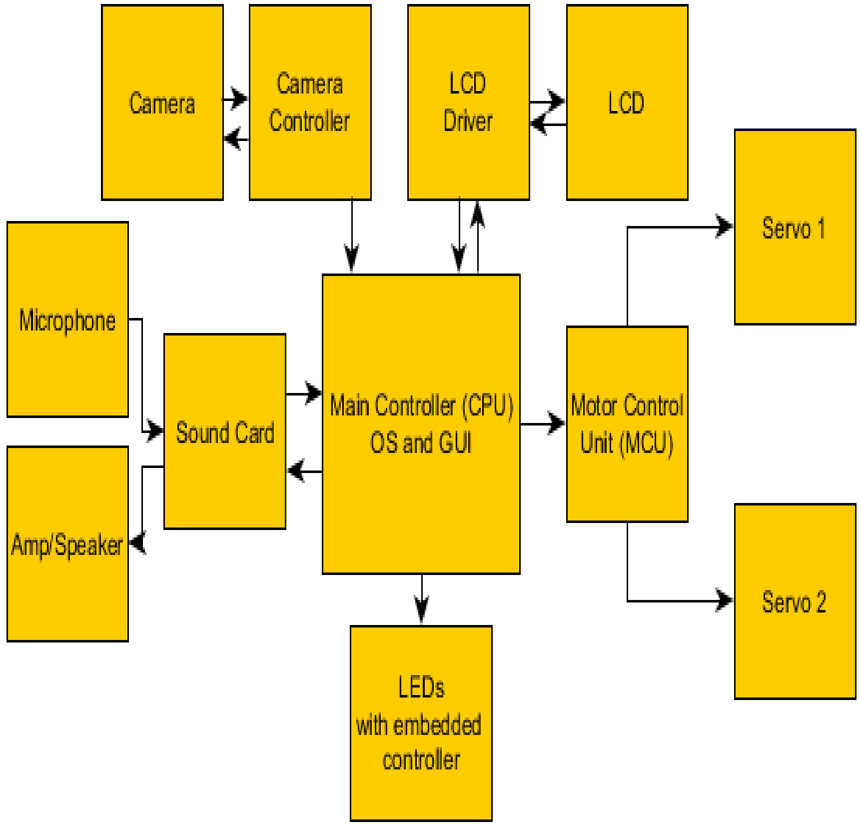


Figure 23: Architecture based on simple CPU and "smart" components

3.4.1.1.3 Conclusion

Both architectures provide considerable benefits over one another and also have considerable drawbacks. Since these architectures are extremes, the solution was to implement a hybrid architecture that focuses on taking advantage of readily available and inexpensive "smart" components, such as a dedicated MCU, and compensating for lack of available components by having the CPU to process images. See Figure 24 next page for the planned hybrid architecture.

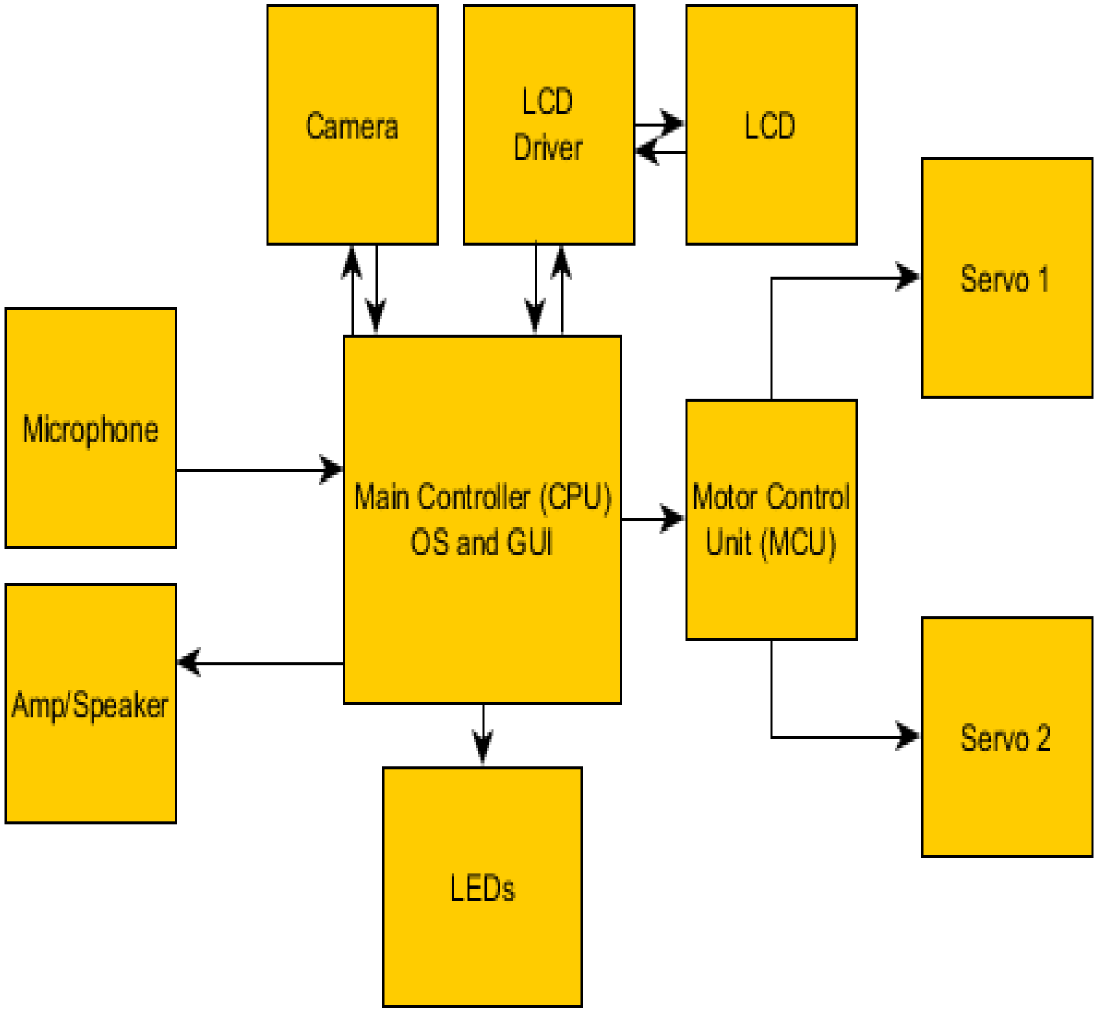


Figure 24: Hybrid Architecture

3.4.1.2 CPU Choice

Because of the high processing requirements of facial recognition/detection and OpenCV's capability to leverage multiple CPU cores, only 4-core CPUs were considered. The main driver in CPU selection past the 4-core requirement was the cost and availability of prototyping boards. As discussed in section 3.3.2.5, the Raspberry Pi 3B had been chosen as the reference SBC, and therefore the CPU choice was an ARM cortex A53 based microprocessor.

3.4.2 MCU

3.4.2.1 Controller vs. Dedicated MCU

Most SoC and therefore SBCs (used for prototyping) are able to put out two or more PWM signals as is required in this project. However, some SoC are not able to provide an accurate signal in this frequency because PWM is also used to dim LEDs. An inaccurate signal would cause too much variation in a duty cycle that is supposed to be of fixed length. This would lead the motor to move (jitter) when it is supposed to remain at its current position. Furthermore, the SoCs should be dedicated to run the main program, so the burden of generating the pulse should be delegated to a different piece of hardware. To solve these problems, an appropriate solution seemed to be a dedicated motor control unit (MCU) that would generate two accurate PWM signals that can be varied by simple signals from the main controller.

3.4.2.2 MCU Considerations

The requirement of the MCU was to provide two independent 4.8V PWM signals that can be varied by the main system controller.

A 556 timer was the most basic approach to generating a PWM and by far the cheapest option. However, it required a separate control unit such as a digital potentiometer to vary the width of the pulse. Instead of using several components to achieve the goal of a variable PWM source, it made more sense to use a chip that has these capabilities integrated. Two cost effective chips had been identified for review because of their available prototyping boards: Texas Instruments M430 (Prototyping board: LaunchPad) and Atmel MEGA328 (prototyping board: Arduino Uno).

The M430 is a little easier to program through the Texas Instruments LaunchPad. However, the native output voltage of the M430 is 3.3V vs. 4.8V for the Atmel chip. Since servos usually require a PWM signal of 4.8V, it was more convenient to use the MEGA328 because there was no need to amplify the control signal, resulting in a simpler circuit. Programming of the MEGA328 was done through either in system programming (ISP) or in circuit serial programming (ICSP).

3.4.3 Mechanical Components

3.4.3.1 Body

Ion’s body houses all of the mechanical and electrical components that make him work while presenting him as a clean, human-like device that can smoothly interact with the user. Many of the main pieces including the base and head were planned to be modeled in Autodesk Fusion 360, a 3D CAD program, and printed using a 3D printer. The head would house the LCD screen, speakers, and microphones. The base would house the power supply. Inside the rest of the body would be the other electronics including the PCB and Raspberry Pi Compute module as well as the servos components which would control Ion’s movement. The connecting ‘tissue’ between the head and base of Ion would be made of a flexible cloth or spandex which would allow him to tilt and move fluidly.

Ultimately, the group didn’t have enough time and resources to put into 3D-printing a decent body for Ion. Being CpE and EE majors, the group wanted to factor on those components that applied to their majors to make a great product rather than focusing on other mechanical factors. The group ended up going with a basic tubing to hold the components while still allowing Ion to rotate. This worked wonders as it was a huge time save while also looked presentable and did its job beautifully.

3.4.3.2 Moving Parts

3.4.3.2.1 Cabling

For the choice of cabling, maximum current draw, available space, and flexibility had been taken into consideration. The most common gage for breadboards is 22AWG, which is rated for up to 7 A. While flexibility will at last depend on the coating, 22AWG tends to be flexible enough to be used in limited motion range applications such as Ion's. With the likely exception of PWM cables, all cables are unshielded. Cables for prototyping were different than that of the final build in terms of termination, see Figure 25 below, for ease of use and cost saving.

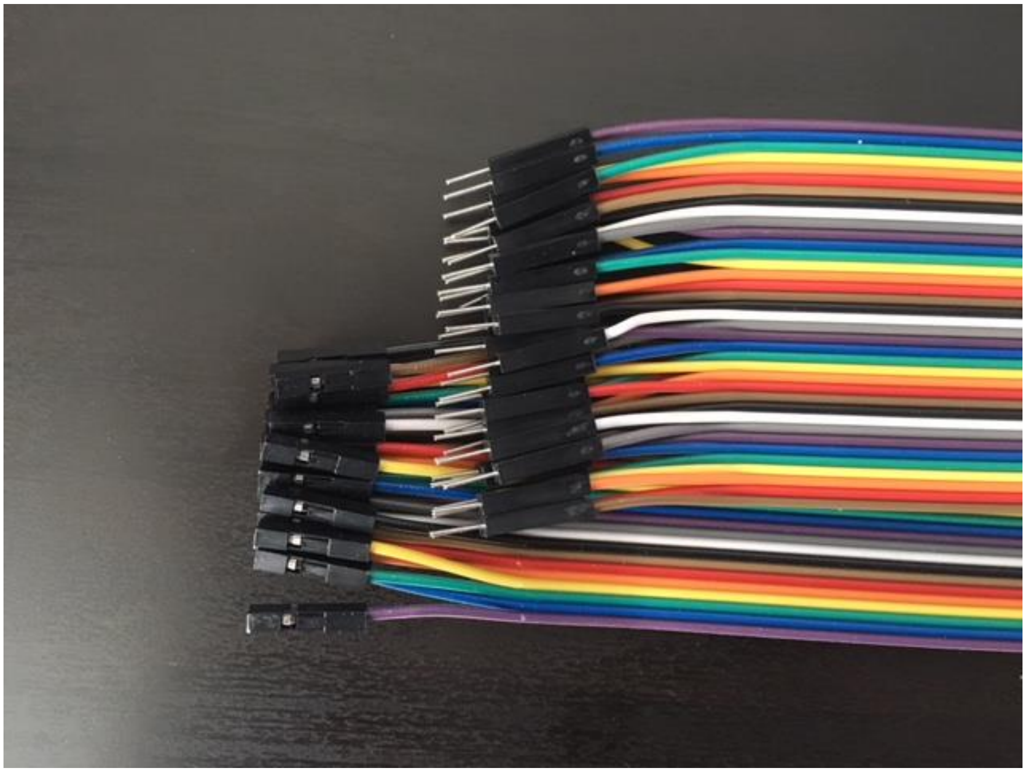


Figure 25: Pre-terminated cabling used for prototyping

3.4.3.2.2 Gears and Bearings

Ion only has one servo. It was planned to have one rotate it on a horizontal plane within 180 degrees and the other one to either bend it to point the camera up and down or just move the camera; see Figure 26 next page. Only horizontal turning was implemented in the final design. To facilitate the rotation on the horizontal plane, a "lazy Susan" style turntable bearing, shown in Figure 27 next page, was used. This reduced the torque requirement, and therefore the physical size and power draw of the servo.

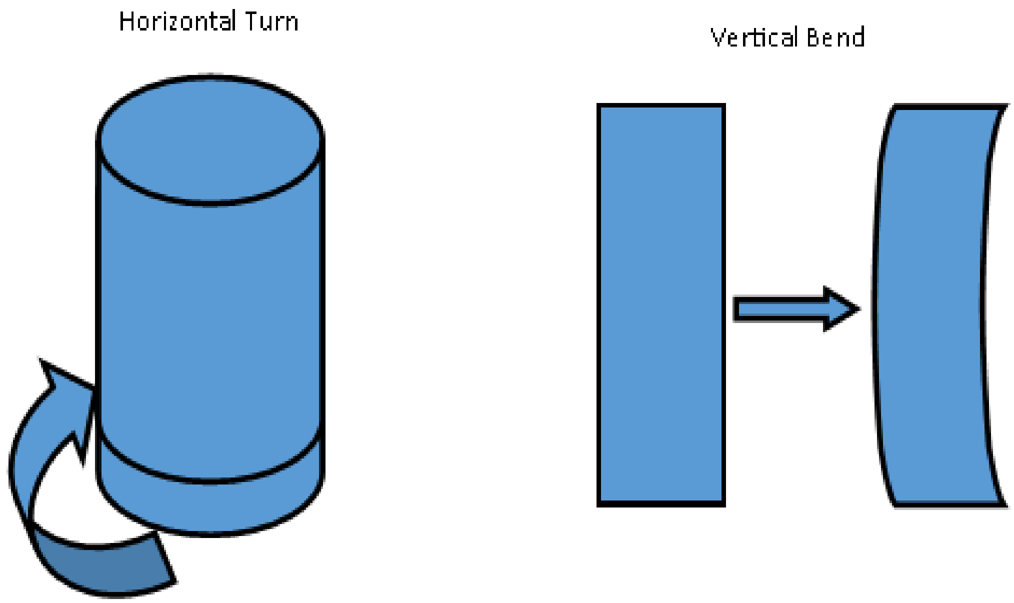


Figure 26: Ion's motion

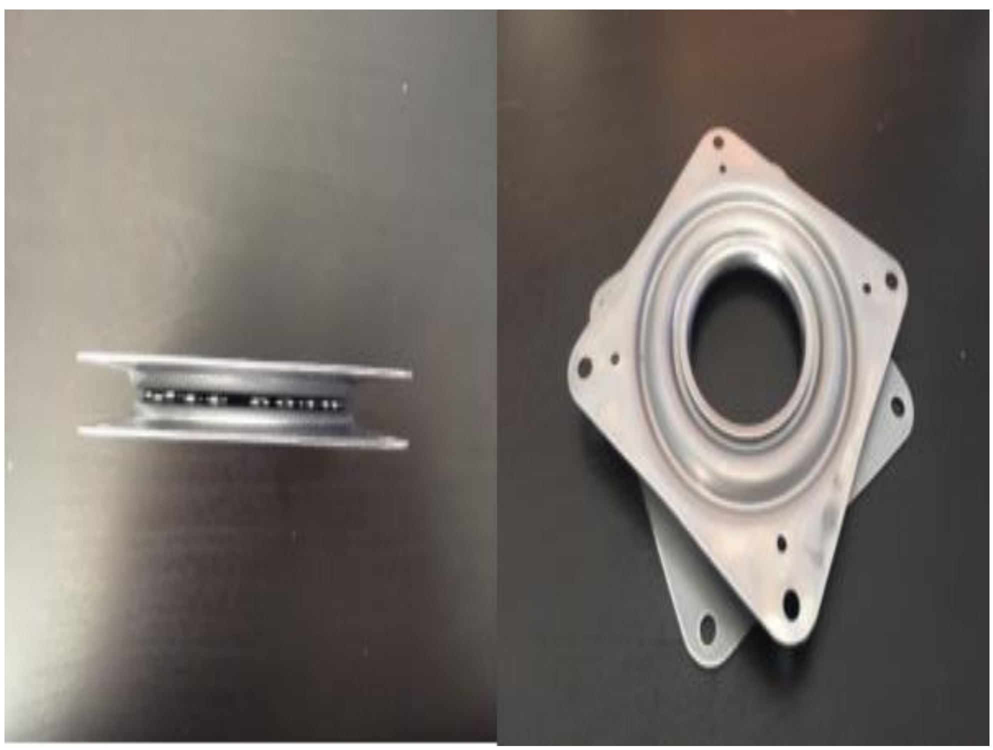


Figure 27: Turntable Bearing. Side view (left) and top view.

4.0 Applicable Standards

4.1 Standards and specification research

Since Ion is meant to be a prototype and not a finished consumer product, no consumer safety standards such as IEC 60950 apply to this design. Because of the wi-fi module, the IEEE 802.11 specification applies but the wi-fi module used already complies with this standard. Other pre-manufactured modules and peripherals are either certified or tested according to, but not limited to the standards listed in Table 6.

Standard	Description
ANSI C63.4-2014 (Revision of ANSI C63.4-2009)	American National Standard for Methods of Measurement of Radio-Noise Emissions from Low-Voltage Electrical and Electronic Equipment in the Range of 9kHz to 40GHz
FCC 47 CFR Part 15, Subpart B	Subpart B deals with unintentional radiators – devices for which the purpose is not to product radio waves, but which do anyway, such as computers

Table 6: Applicable Standards

4.2 Design Impact

The impact of these standards on this design were minimal since most standards applied to peripherals and pre-manufactured modules as mentioned above. Unless the design changed or was developed further for sale and consumer use, no standard affects the design in its current stage.

5.0 Design Constraints

Mentioned below are all the Accreditation Board for Engineering and Technology, Inc. (ABET) design constraints that influenced the design of Ion. These include ethical, environmental, economic, time, health and safety, manufacturability and sustainability, as well as social and political. All these constraints put a limit on our designs in ways which will be described in this section.

5.1 Ethical Constraints

Because the product is being used directly by the user, there are many ethical constraints that come along with it. With use of a camera and microphone, there is an ethical standard that must be followed in relation to privacy and securing data. It is understood that those components won't be used without the user's permission or be used in ways that aren't already specified or make sense for the device. Also, with an 'intelligent' two-way conversation system like the one at hand, it is important to make it appropriate for the audience. The project's audience includes anyone who can make use of the device - from a young toddler to an older person. Because small toddlers may use it, it is necessary to make sure that the device doesn't say profane words or access inappropriate content.

5.2 Environmental Constraints

As far as environmental constraints, the product is meant to be used in a dry, indoor room in which it won't be subject to abnormally high heat or humidity. High heat or humidity could cause the device to overheat or malfunction. The microprocessor and CPU need a relatively low heat to keep cool and function properly. Humidity and precipitation can also cause the system to overheat or malfunction. The device is also meant to sit on a flat surface on which it can move around and tilt. Using the servos motors, the device can tilt and turn to convey its emotions and responses and needs a large enough space to be able to do this.

5.3 Economic Constraints

The group didn't have many economic constraints besides what individual members could afford. Being college students with limited budgets, the group's members could only put in around $200 each, for a total project budget of $600. This amount was more than enough to do research, order parts, and develop a fully-working prototype. Other than the personal budgets, the only other constraint that was considered was the price of similar products on the market. The group wanted to provide a decent product for a competitive price, so they had to take that into account. Similar products on the market go for $100-150, but the companies that produce and manufacture them are major powerhouses in the industry and have supply accesses and more resources than the group does.

5.4 Time Constraints

There were multiple time constraints for the design that had to be dealt with. The first time constraint was that the group had only two semesters to finish the design; one to research and the other to build it. Since the group had decided to finish in the Summer semester, the time constraint became an even bigger factor, given that the Summer semester is a few weeks shorter than other semesters. To deal with that, the group came up with milestones, further illustrated in section 9.1 of this document, that should be achieved to keep a better track of the group's progress, while also making sure the design is being completed in a sound manner, not all crammed in one week.

Another time constraint was scheduling. It was known that the group members each had their own busy schedules of school and work together, so each member had limited time to meet up and discuss the design with the rest of the group.

5.5 Health and Safety Constraints

There were several health safety constraints for the device. One of them was that the device should not produce electrical surges, especially not while the user is near and ends up getting shocked or zapped. Another was that the device should not heat up to a point where components would melt or break. Also, the device should not behave unexpectedly and blow up if anything in the circuits goes wrong. The device should be accessible by children as well, so, it should not consist of any small or loose parts that children can swallow.

5.6 Manufacturability and Sustainability Constraints

The main constraint for manufacturability is the available equipment for 3D printing. The PCB and other internal components could've become too big to fit into the maximum sized housing that could be produced with the available equipment. Another constraint was moving mechanical parts. The design team consisted of only electrical and computer engineers, so custom designed moving parts such as gears and bearings were outside the scope of this project. This constraint limited the design to components readily available on the market.

The sustainability of the device depended mostly on the life and support for the electrical components. Most components, such as the MCU chip have been available for a long time and are mostly guaranteed to be available for the next five years by their manufacturer. Other components, such as the servos are interchangeable and can be replaced with equivalent parts if necessary. Should any of the electrical components be discontinued, a new design may be necessary if there is no available alternative.

As far as the body, we planned to use a 3D printer, specifically the MakerBot Replicator+. This 3D printer can print up to 11.6" by 7.6" by 6.3", which met the requirements of the product. It can use the Tough PLA Filament Bundle, which is more durable and high-impact, which helps our product in case of damage or dropping. This fit the product requirements in terms of printing size, printing material, and printing cost. The group is using an acquaintance's 3D printer as opposed to a professional 3D printer as it is cheaper and easier to work with. If the time did come to mass-produce, the group would have to think about going elsewhere for manufacturing. As for the prototype, this 3D printer was a great option for manufacturability and sustainability.

5.7 Social and Political Constraints

A large constraint created by society includes the negative outlook on data privacy and AI. With the recent data breaches hitting huge companies such as Facebook and Equifax, it is no wonder that consumers are becoming very skeptical of giving their personal information to companies. Many are beginning to realize how much information is given to big companies and is on the internet in general and it is becoming a huge issue when these same big companies who throw millions into security are being hacked or misusing data that gets it in the hands of someone who can call you as a tele-marketer or use it to gain political edge. The group could curb this worry by only collecting the necessary information and informing the user of what their data will be used for and sticking to that promise. Another large social stigma, at least in technology, is the use of artificial intelligence, or AI. Many people are becoming skeptical of the use of AI for many reasons. From the old mindset that AI could take jobs, to the newer big-news stories like Tesla's AI auto-pilot having some technical issues and crashing – AI is a huge social speedbump that will have to be overcome in the coming years. While using AI, the group could calm that worry by making sure the user knows what functions the AI can do and the fact that it is a fun, interactive form of AI will certainly make it easier.

6.0 Project Hardware and Software Design

6.1 Hardware Design

6.1.1 Wireless System

6.1.1.1 WiFi Chip

For the groups final design, a WiFi chip had to be used rather than an entire WiFi microcontroller board to reduce the number of PCBs used. The group decided to go with the ESP8266 SMT Module, the ESP-12S, as a WiFi chip that can be mounted on their own PCB. Advantages the ESP8266 module provides are that it has a 4MB flash chip, an integrated analog-to-digital converter, and an antenna on-board that are already pre-programmed. It also has an integrated TCP/IP protocol stack and supports 2.4GHz WiFi with WPA/WPA2 security. The ESP8266 SMT module also features three different power saving modes: deep sleep mode, sleep mode, and active mode. In the sleep modes, the operating current can be reduced to around 12µA or 0.5mA. The ESP8266 SMT module is shown below in Figure 28, it is physically small and fits well on a PCB. [9]

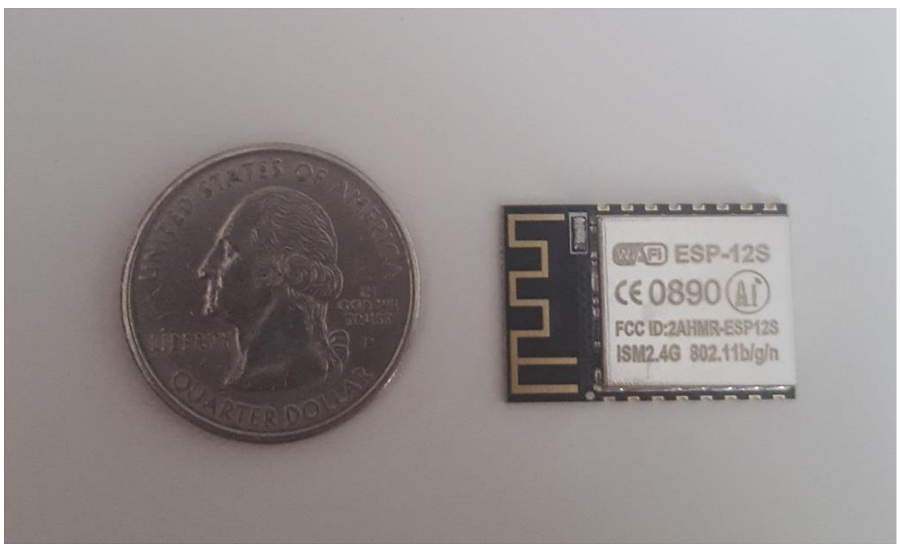


Figure 28: ESP 8266 SMT Module/ESP-12S

In the final design, an ESP-12E chip was used instead, which is very similar to the ESP-12S but it has more pins on the bottom that can be utilized.

6.1.1.2 Circuit Design

Circuit design for the ESP 8266 SMT module was simple since Adafruit provides the schematic and board layout that you need to make it work. The ESP8266 SMT module board layout is shown next page in Figure 29 that illustrates all the inputs and outputs for the module along with its pins. This schematic was very important for building and testing the WiFi chip for the final design and will be further discussed in the PCB schematics section.

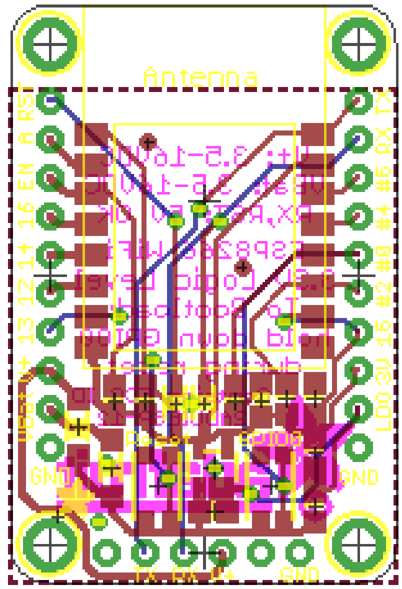


Figure 29: ESP8266 WiFi SMT Module Board – Board credits to Adafruit

6.1.2 Motor Control Unit

The motor control unit (MCU) is responsible for taking instructions from the CPU and positioning the servos accordingly.

6.1.2.1 Design Considerations

Ion planned to have two servos that need to be controlled by the MCU, so the minimum number of PWM outputs of the MCU chip had to be 2. It must also have been able to receive commands via I2C.

6.1.2.1.1 PWM Source

As outlined in section 3.4.2.2 "MCU Considerations", it was determined that the source of the PWM signals should be an external component. Among the two options researched, the Atmel MEGA328 stood out as the better option because of its native output voltage of about 5V, which reduces the voltage control components required on the PCB.

6.1.2.1.2 Power Source

The PWM source itself was powered through the main controller because the current of the signals is very low. The Servos was powered directly from Ion's power source because of the motor's required higher and varying current. The high current could not only damage the PWM source, variations in the high current could generate noise that interferes with more sensitive components on the PCB.

6.1.2.1.3 Servo Specifications

The required specifications for the two servos are listed in Table 7 next page.

Specification	Up/Down Servo	Rotational Servo
Operating Voltage	5V +- 10%	
Torque (kg/cm)	At least 1.5	At least 3
Size (cm, w l h)	Less than 2.5 x 1.5 x 2.5	Less than 3 x 2 x 3
Speed (s)	Less than 1	
Weight (g)	9	
Gear Material	Plastic	

Table 7: Servo Specifications

6.1.2.1.4 Feedback System

The purpose of the feedback system is to create a closed loop control system for the camera movement. The servo can move to a designated position based on the given control signal but there had to be a way to determine the relation of the motor position to the focal point of the camera to ensure that the camera is properly centered on the user's face.

6.1.2.1.4.1 Position Based Adjustment

The facial detection software provides the position of the detected face relative to the center of the captured frame. The focal point of the camera is the center of the frame, so the distance of movement necessary to position the face in the center can be calculated. Based on that calculation, the position for the servos can be updated (see Figure 30).

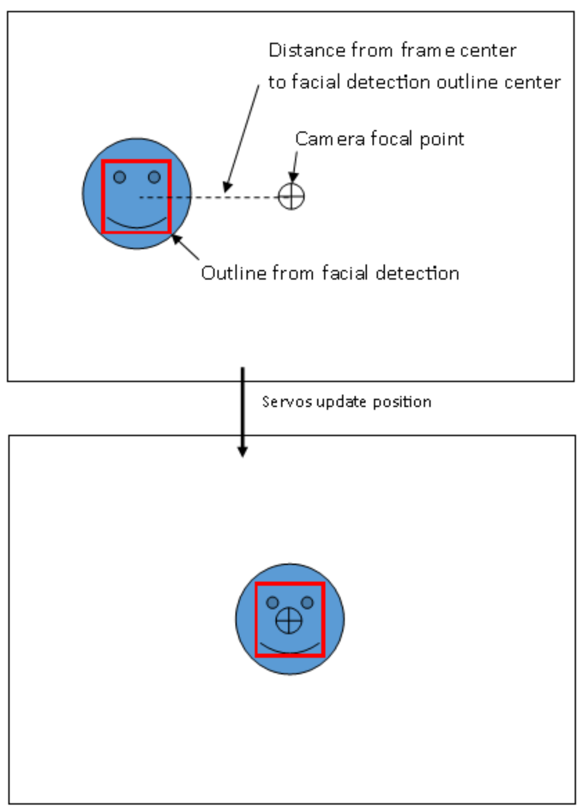


Figure 30: Camera position update based on calculated distance from center.

However, because of possible inaccuracies in facial detection, wrong calculations can lead to unwanted over- or under-correction in position. The position of the face is based on the center of the rectangle drawn around the detected face. The perceived center of the face can be wrong and an overestimated (or underestimated) distance from frame center to face center could be calculated. As a result of an overestimated distance, the position of the camera can overshoot at first and then correct itself to the opposite direction. This could be perceived as jitter. Furthermore, small movements by the user, or turning of the face could lead to many minor corrections of the camera position. This is not only annoying but also draw more power from the battery than necessary.

This feedback design also relies on proper estimation of distance from face to center. For this, the distance on the frame needs to be translated into rotational distance. To accurately do this translation, the distance from the face to the camera must be known. Figure 31 shows a bird's eye view of the camera and two heads. Note that the two heads are the same person but at different distances from the camera.

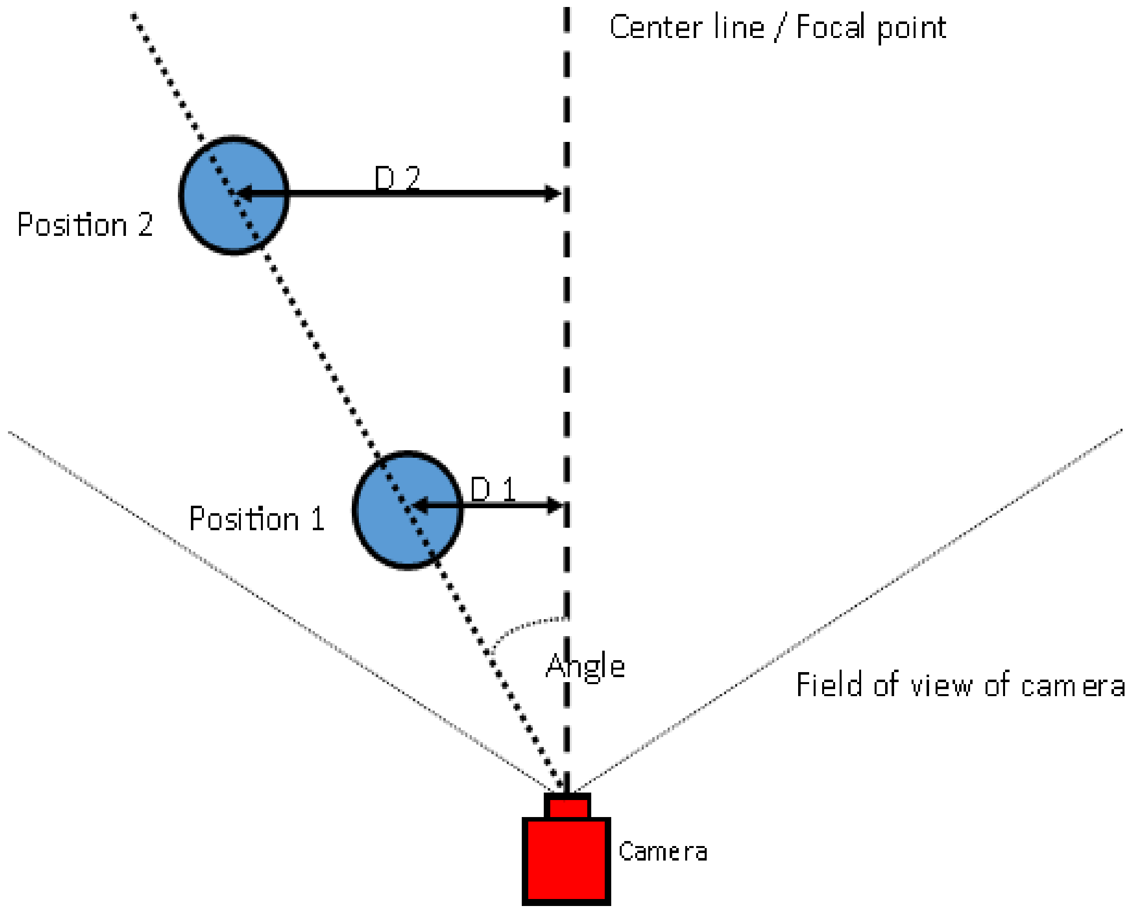


Figure 31: Bird's eye view

With the current design of Ion, the only way to estimate this distance is by measuring the size of the rectangle that is drawn around the face and relate that to the size of an average face. Figure 32 shows the same face in position 1 and 2, based on Figure 31, as perceived by the camera.

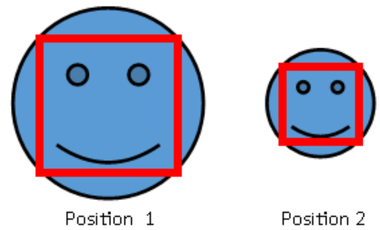


Figure 32: Same face viewed by camera at different positions.

If the size of the viewed face is known, the distance can be somewhat accurately calculated. The lack of accuracy stems from the possible lack of accuracy in the facial detection software. Only a small error in drawing the rectangle around the face can result in a miscalculated distance of the face and therefore a miscalculated target position for the servos. Another source for this type of error could be the difference in size of user's face. A person with a face that's larger than average will always be perceived to be closer to the camera, leading to a constantly overestimated angular adjustment of the servo.

A fix for overcorrection is a larger focal point for the camera. By reducing the required accuracy of the corrective movement, small errors result in less continuous, small adjustments in position. There may still be some adjustments if the center of the face is close to the border of the focal point though. To properly avoid jitter and overcorrection, a more dynamic approach to a feedback system was necessary.

6.1.2.1.4.2 Direction Based Adjustment

Instead of providing a fixed position the servos should move to, a vector was drawn from the center of the face to the focal point of the camera. That vector was converted into two vectors (for movement along x, and y axis) to provide a direction for each servo to turn to. This was done by correcting the position of the servo by a small amount towards the wanted position for every analyzed frame as shown in Figures 33 to 35.

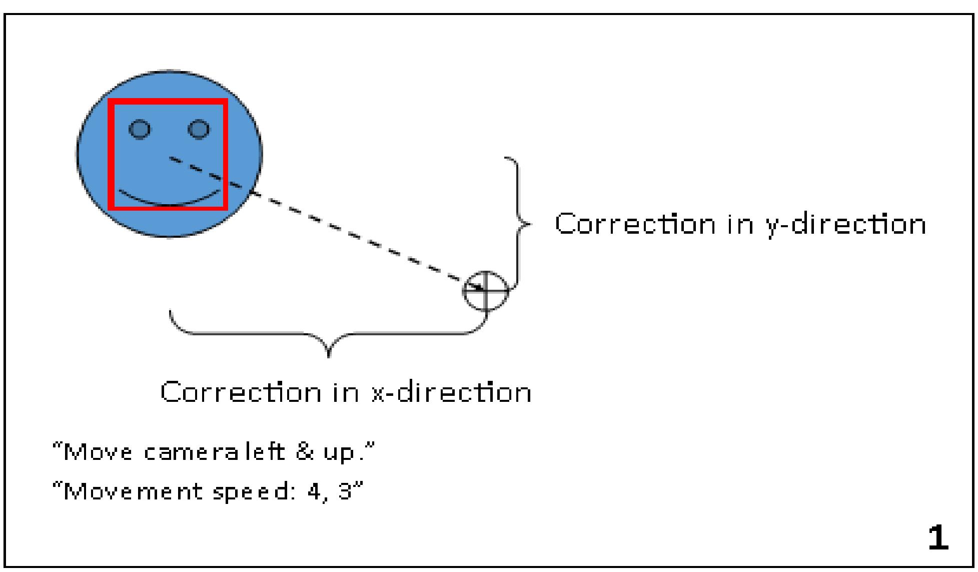


Figure 33: Dynamic feedback system step 1

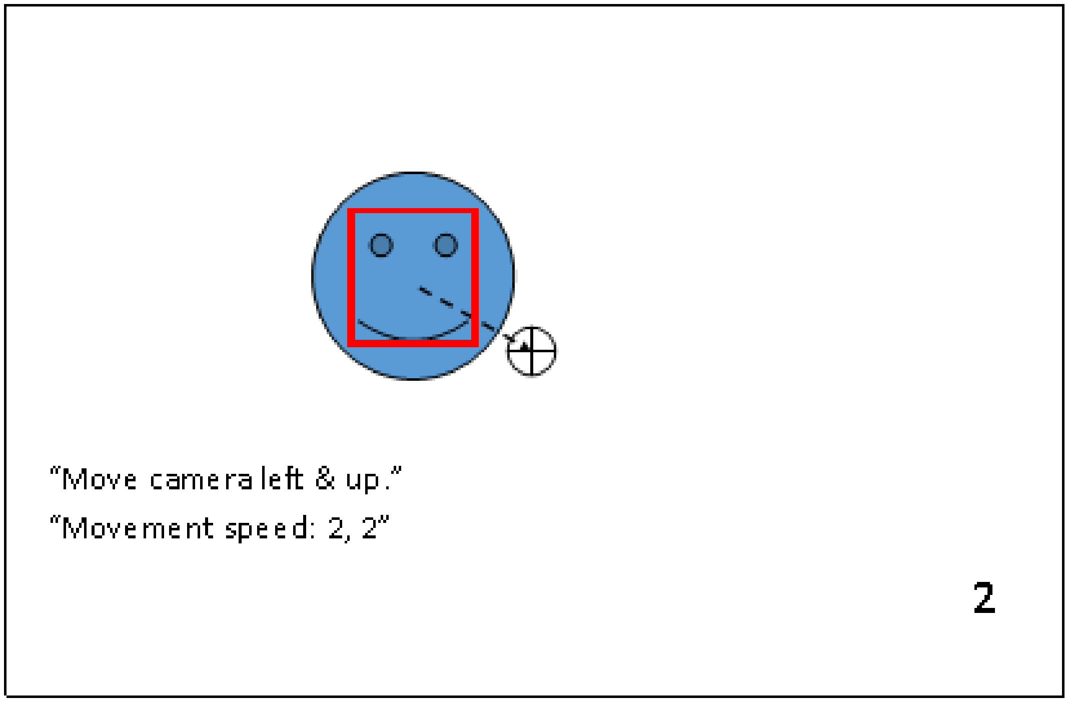


Figure 34: Dynamic feedback system step 2

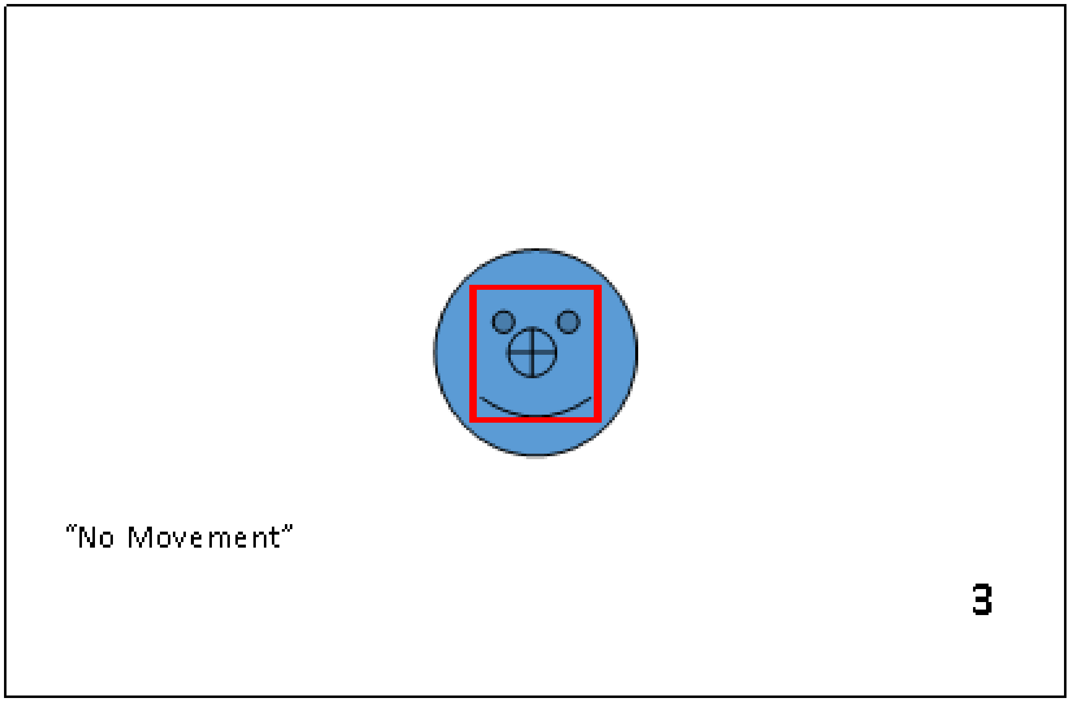


Figure 35: Dynamic feedback system step 3

Since the position is updated several times per second, the movement is stopped once the distance to the camera focal point is within a certain range. To avoid overshoot, the movement speed is adjusted by the magnitude of the vector. The closer the face is to the focal point, the slower the adjustment in the servo position.

The problem with this approach is that the movement will stop as soon as the face is within the closest border of the determined range from the focal point. This leads to corrections as soon as the face leaves that range. To reduce these types of corrections, as soon as the face is in range, initiation of movement depends on leaving a larger area as shown in Figure 36.

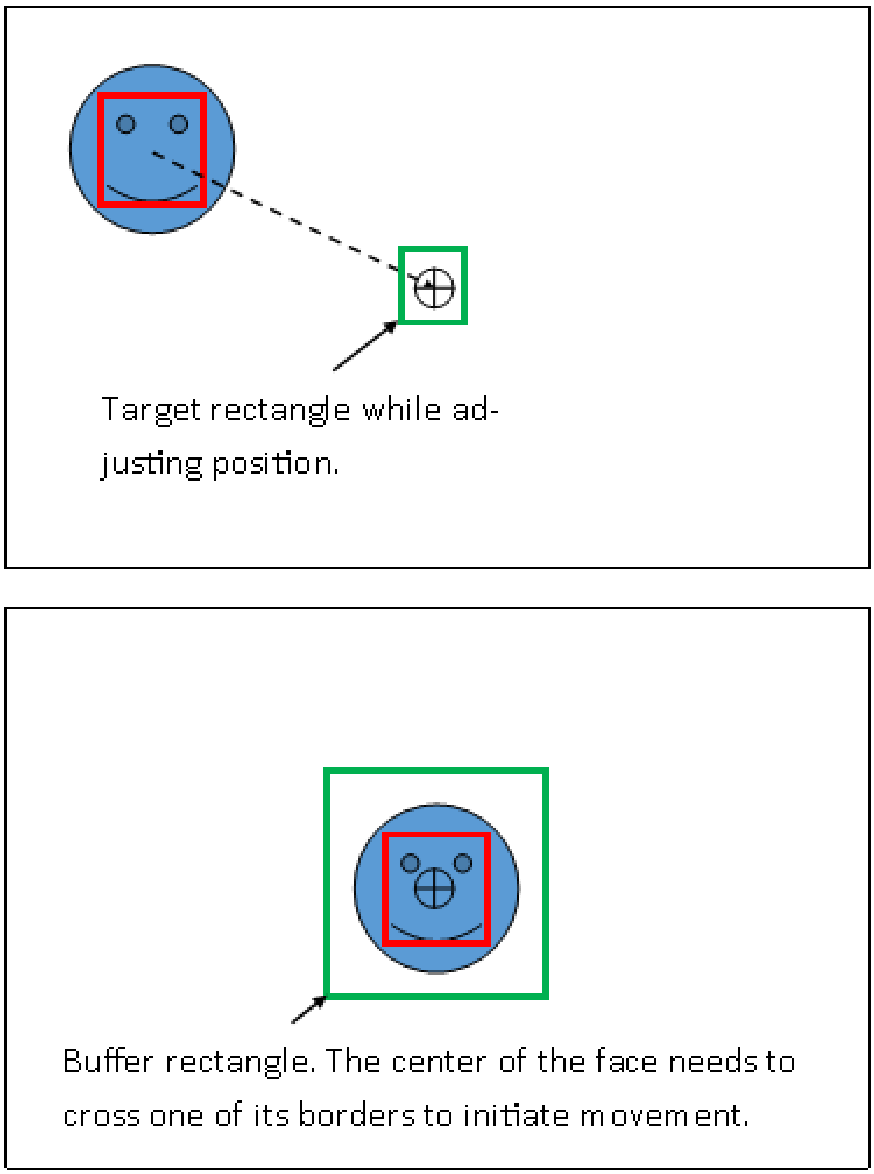


Figure 36: Varying buffer zone based on movement

Another variable to consider when determining the magnitude of the adjustments is the distance from the user to the camera. As discussed in the Position Based Adjustment, the distance from the face to the camera can be estimated by the size of the face. Since a smaller bounding rectangle means that the face is further away, the linear distance measured on the camera frame can lead to an overestimation of the necessary correction to the position. Therefore, the smaller the bounding rectangle of the face, the smaller the corrections should be and vice versa. Different sized faces were not a problem since the amount of correction would still decrease the closer the face gets to the center of the frame.

In summary, a vector was drawn from the center of the face to the center of the captured frame. It will then be broken down into two vectors along the x-, and y-axis. These vectors give the direction of the correction for each servo and the magnitude of these vectors, along with the size of the face's bounding rectangle determine the amount of the correction. Several corrections per second result in movement of the camera until the user's face is centered in the frame.

6.1.2.1.4.3 Implementation

The size of the varying buffer zones, as well as the necessary amount of position adjustment based on the face's position could currently not be easily calculated. Both of these variables depend on the speed and accuracy of the facial detection software, which in turn depends on the power of the CPU and software implementation. Because of this, instead of trying to calculate the appropriate variables before the rest of the system was built, the code for the feedback system was written in such a manner that it was easy to calibrate these parameters in the finished prototype. The feedback for the MCU relies entirely on software.

6.1.2.1.4.4 Algorithm Flowchart

Figure 37 shows a high-level flowchart of the algorithm. Note that the function does not repeat itself. The loop of the main program calls this function when needed. This adds the option to pause movement and retain more control over the feedback system.

6.1.2.1.5 Communications

All calculations for the feedback happen in the CPU, hence the only signal that is transmitted to the MCU are the variables that need to change and the amount of that change. Since the MCU and CPU reside on the same PCB with close proximity, an I2C bus was used. The MCU does not transmit any data to the CPU, so the CPU can be set as master and the MCU as slave. The data sent to the MCU is saved by the CPU and MCU separately to the respective variables. This eliminates the need for retrieving data from the MCU.

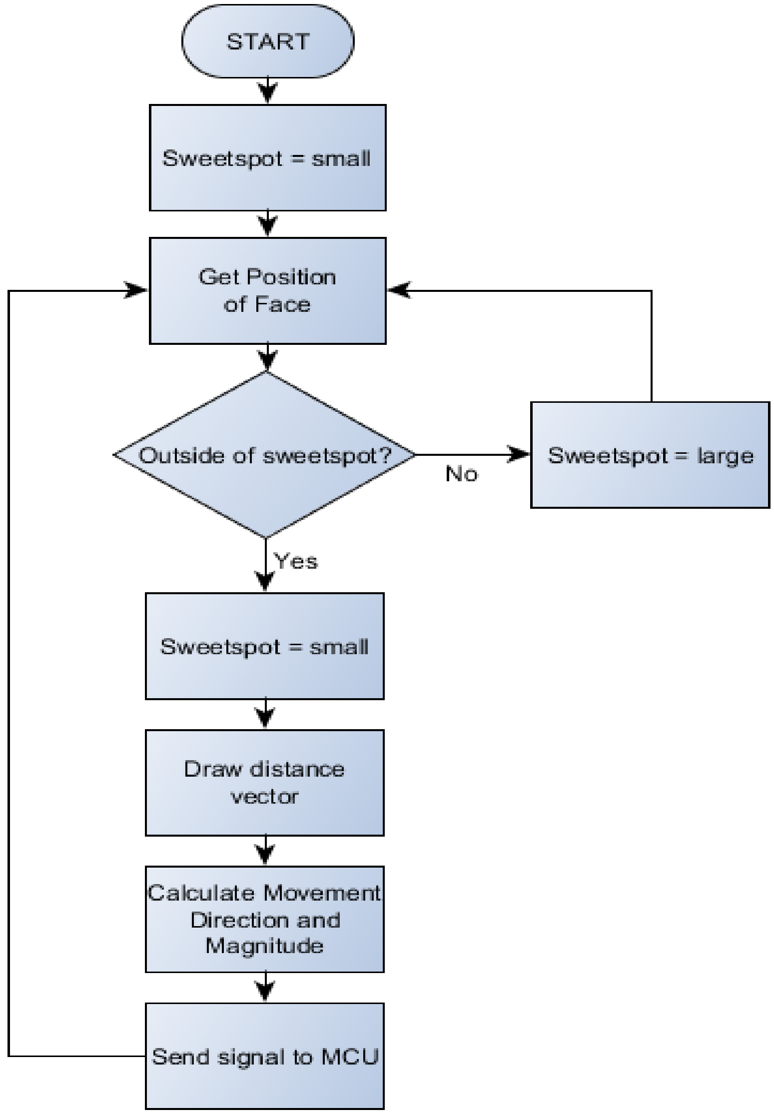


Figure 37: Feedback system algorithm flowchart.

6.1.3 Power System

6.1.3.1 Electrical Power Consumption

Electrical power consumption helps determine the total power consumed by the device. It was important to keep track of it to make sure that the maximum total power consumption of the device being 50W requirement met.

To keep track of electrical power consumption, although not 100% accurate and still an estimate since power consumption is not constant, each member of the group listed the nominal DC input voltage for each individual circuit along with the nominal current draw. Then a table, illustrated next page in Table 8, was constructed with all the nominal input

DC voltages and nominal current draws along with the total of each. After that, using the power equation P=VI, the total power consumption was calculated by adding up the individual power consumption of each circuit. Note that some of the components' current draw was unknown, so they were approximated. In conclusion, 1A current was enough to supply the device and 2A was more than enough. Also, the device meets the requirement of consuming less than 50W.

Component	Nominal Input Voltage (V)	Nominal Current Draw (mA)	Power Consumed (mW)
Audio Amplifier	5	2.8	14
Processor	5	16	80
LCD	5	~100	2400
Servo Motors	5	500	2500
Motor Control Unit	5	0.5	2.5
WiFi Chip	3.3	80	264
Camera	5	250	1250
LEDs	5	~20	100
microSD Card	3.3	100	330
Total:		~1.07A	~6.94W

Table 8: Electrical Power Consumption

In the final design, due to changes to the components and the device in general, a new table, Table 9, was constructed shown below.

Component	Nominal Input Voltage (V)	Nominal Current Draw (mA)	Power Consumed (mW)
USB Peripheral	5	~500	~2500
Processor	5	700	3500
LCD	5	~200	~1000
Servo Motors	5	200	1000
Motor Control Unit	5	0.2	1
WiFi Chip	3.3	80	264
Camera	5	250	1250
LEDs	3.3	~20	~66
microSD Card	3.3	100	330
Total:		~2.05A	~10W

Table 9 Final Design Electrical Power Consumption

6.1.3.2 Power Connections

The power connections were an important part of the design, especially since the power connections show a basic overview of how all the circuits will be connected to each other. The power all begins with the battery supplying a certain voltage and current, and then every circuit will require a certain constant DC input voltage along with a current draw. Since it is expected that all the circuits will be integrated into the PCB, a voltage bus can be created on the PCB along with a common ground bus. There were multiple proposals to design the power connections.

The first proposal was to place a voltage regulator at the input of each circuit to keep the input voltage at a constant DC value if different from the voltage bus value. The only advantage of that power system would be that it is guaranteed that the current draw for each voltage regulator is accurate, and that the DC voltage being supplied to each circuit is exact. There are many disadvantages to that, if an LDO regulator were to be used, there will be a lot of power wasted, or if any voltage regulator were used, there will be a lot of space being occupied for no reason. The resulting system looks something like Figure 38 below, which is very wasteful.

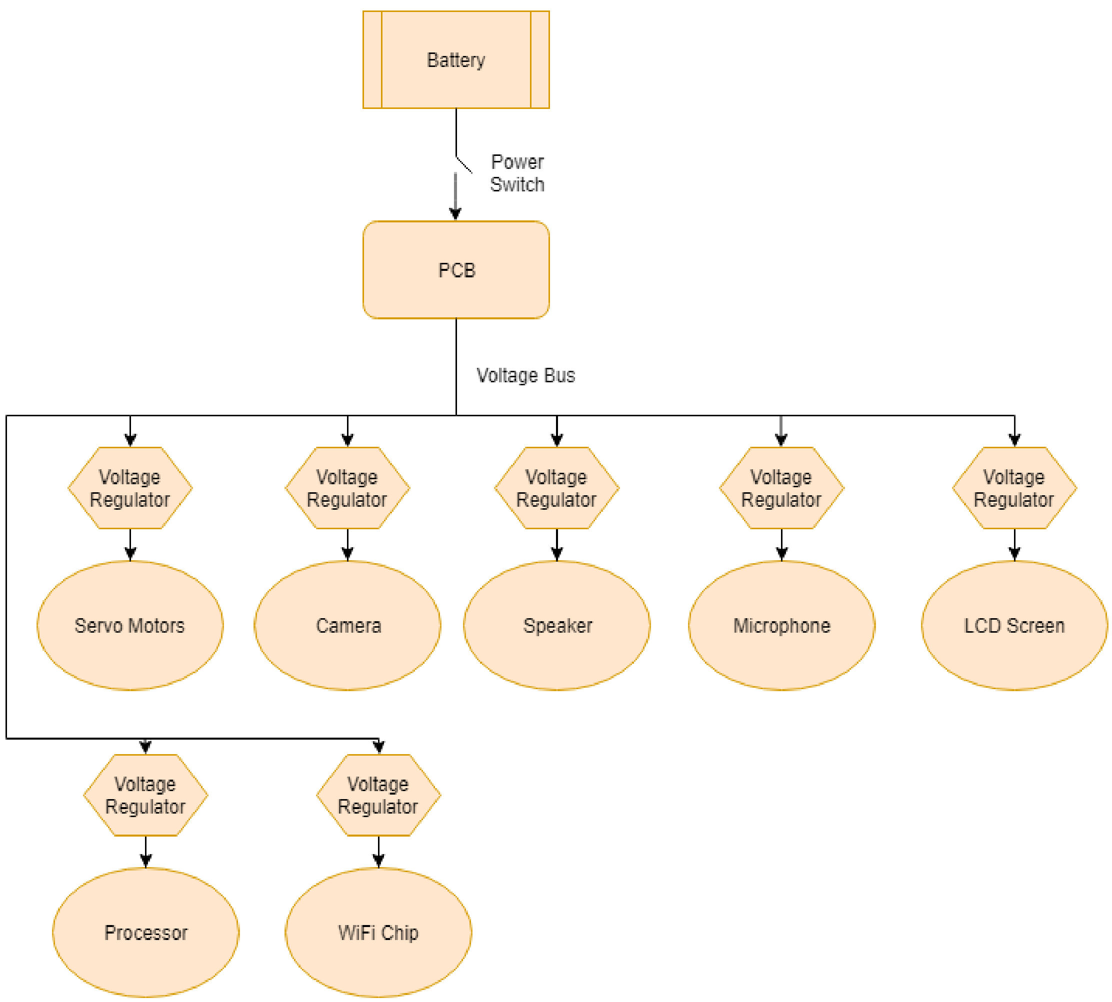


Figure 38: Initial Power Connections Proposal

The second proposal was to have only two voltage regulators, one that boosts the battery from 3.3V to 5V, then steps down the voltage from 5V to 3.3V for the WiFi chip. This system ends up being simple and easy to implement, already much better than the previous system. But, it turns out the group's processor required multiple voltage inputs, and then using its GPIOs, it can power some of the components which would lead to a much better and simpler design so long as the current draw is not too high for the processor. The resulting connections are shown below in Figure 39.

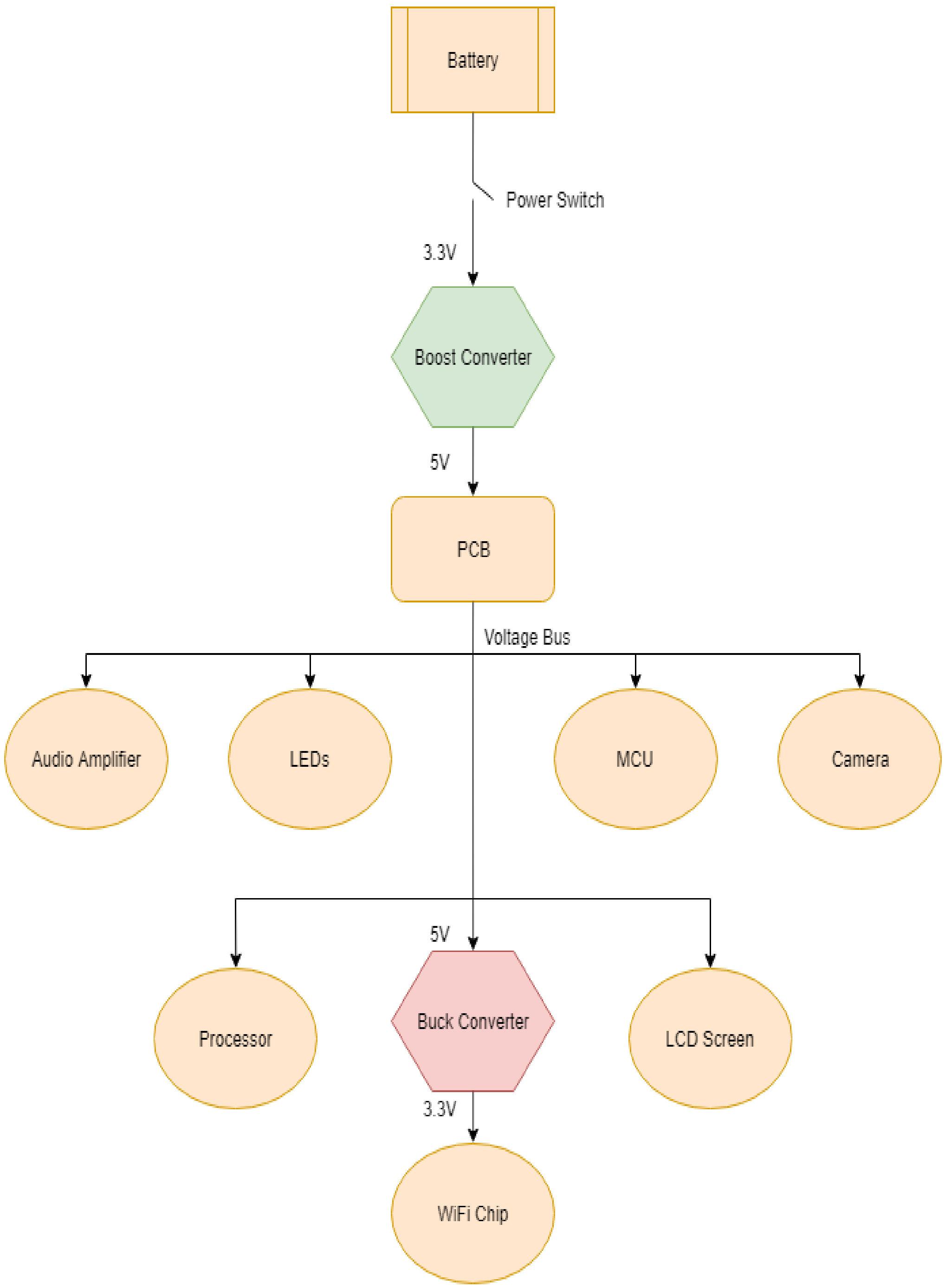


Figure 39: Second Power Connections Proposal

The final proposal was to create two voltage buses that use DC-to-DC voltage regulators keeping the voltage at a constant 3.3V and 5V since the battery voltage might drop down after multiple uses. Then, two step-down voltage regulators can be used to drop the voltages to 1.8V and 2.5V to power the processor. After that, some of the components were powered the processor itself using its GPIO pins. The final proposal for power connections is shown below in Figure 40.

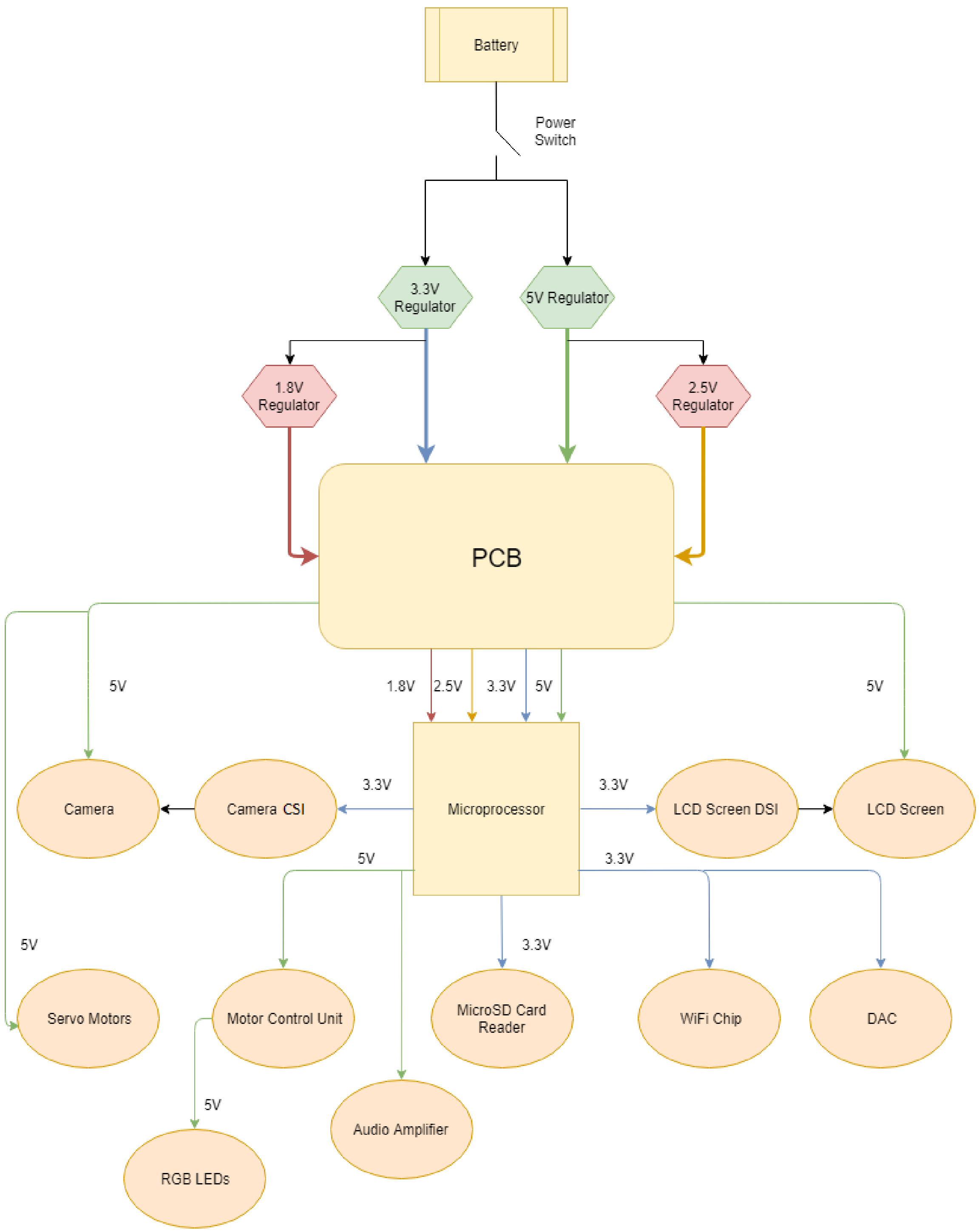


Figure 40: Final Power Connections Proposal

In the final design, since also the previous diagrams were misleading since the PCB is not a component, but instead holds the components, the final power connections diagram was constructed, shown below in Figure 41, that shows all the connections to fit the changes done.

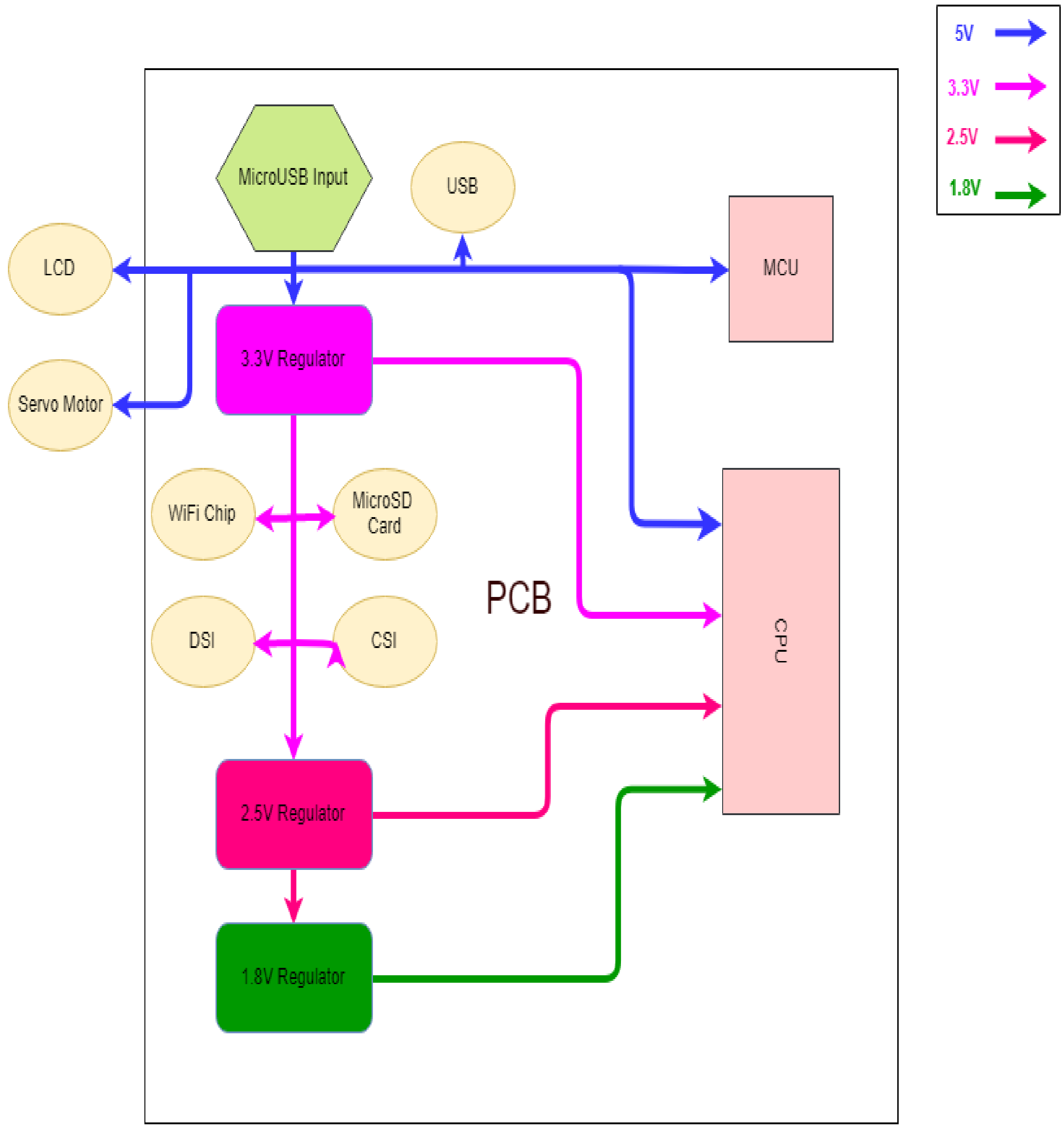


Figure 41 Final Design Power Connections

6.1.3.3 Voltage Regulators

A voltage regulator is necessary when there are multiple circuits in a system with different input voltages, and especially when there is a battery that does not provide a constant voltage over a long time. The goal of the voltage regulator was to regulate the voltage coming in from the battery through a voltage bus and outputting a specific constant voltage required depending on the circuit or electronic component.

6.1.3.3.1 3.3V-5V DC-to-DC Voltage Regulator

The group decided it was smart to start off with regulating the battery voltage since its voltage will drop after a while. Simulations could be done on WEBENCH® Design Center. Since most of the components used for the device were expected to be powered by around 5V_{DC}, a boost converter was expected to be used for the battery to regulate the output at 5V and maximum output current of around 3A. Shown below, in Figure 42, is the boost converter TPS61088RHLR schematic placed at the battery output along with its important parameters in Table 10 and its bill of materials in Table 11. [10]

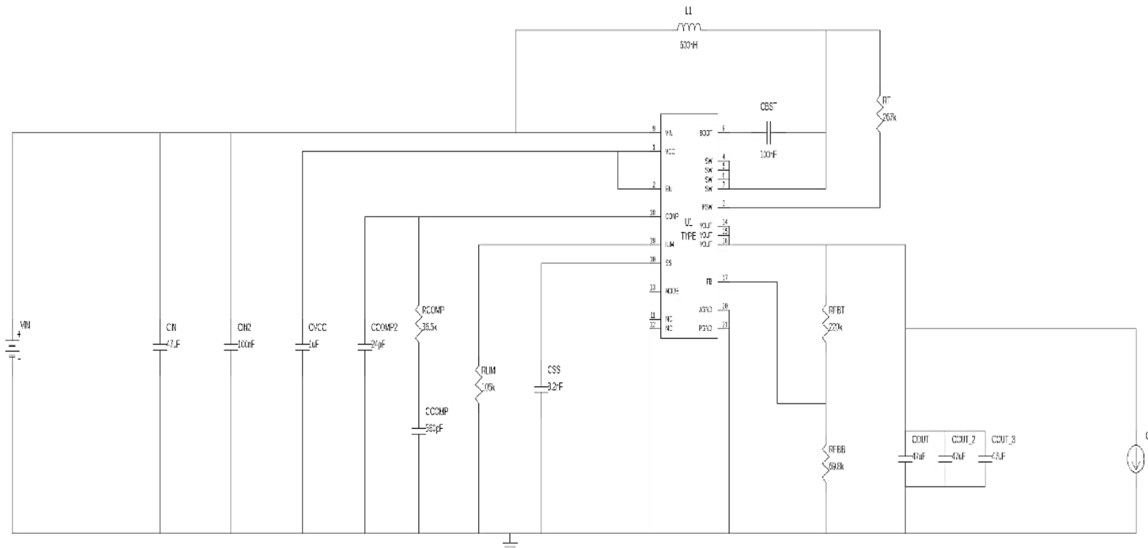


Figure 42: 3.3V to 5V Boost Converter Schematic – TPS61088RHLR – Schematic Credits to WEBENCH® Design Center

Design	BOM Footprint (mm^2)	Efficiency	BOM Count	I_{out} Max (A)	IC Cost	BOM Cost
Fully-Integrated Synchronous Boost converter	151	93%	17	10.6	$1.60	$3.58

Table 10: 3.3V to 5V Boost Converter Important Parameters

Part	Manufacturer	Part Number	Quantity	Price ($)
Rfb1	Vishay-Dale	CRCW0402976RFKED	1	0.01
Rfb2	Vishay-Dale	CRCW04022K94FKED	1	0.01
Cbyp	MuRata	GRM155R60J104KA01D	1	0.01
Ccomp	TDK	CGA4J2C0G1H333J125AA	1	0.09
Ccomp2	TDK	C2012C0G1H182K060AA	1	0.03
Cin	Panasonic	16SVPF1000M	1	0.77
Cout	Panasonic	16SVPG270M	4	0.68
Csense	MuRata	GCM2195C1H103JA16D	1	0.05
Cvcc	Panasonic	ECPU1C684MA5	1	0.22
D1	Vishay-Semiconductor	M3035S-E3/4W	1	0.63
L1	Bourns	SRP1250-1R0M	1	0.69
M1	Texas Instruments	CSD16323Q3	1	0.36
Rcomp	Vishay-Dale	CRCW04022K67FKED	1	0.01
Rfadj	Vishay-Dale	CRCW040276K8FKED	1	0.01
Rivp1	Yageo America	RC0201FR-0744K2L	1	0.01
Rivp2	Vishay-Dale	CRCW040248K7FKED	1	0.01
Rs1	Vishay-Dale	CRCW040297R6FKED	1	0.01
Rsense	CUSTOM	CUSTOM	1	0
U1	Texas Instruments	LM3481MM/NOPB	1	0.80

Table 11: 3.3V to 5V Boost Converter Bill of Materials

6.1.3.3.2 3.3V-3.3V DC-to-DC Voltage Regulator

The 3.3V to 3.3V boost converter was required to create a 3.3V bus within the PCB for multiple uses including powering the microprocessor and other components. The current draw was assumed to be around 3A again. The design shown below in Figure 43 was created using WEBENCH® Design Center. The important parameters and bill of materials are shown in the tables Table 12 and Table 13 respectively. [10]

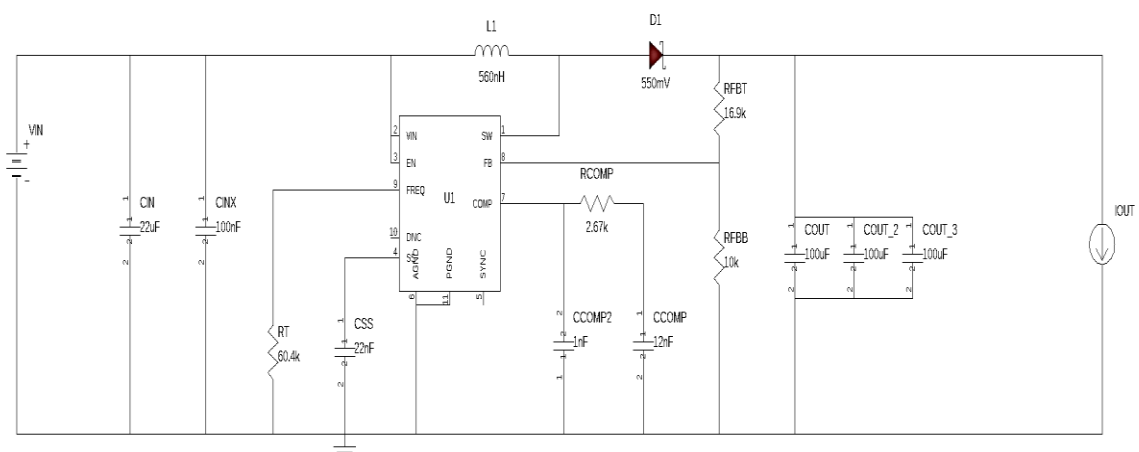


Figure 43: 3.3V to 3.3V Boost Converter Schematic - Schematic Credits to WEBENCH® Design Center

Design	BOM Footprint (mm^2)	Efficiency	BOM Count	I_{out} Max (A)	IC Cost	BOM Cost
Wide Input Range Boost DC/DC converter	208	87%	16	5	$1.30	$2.95

Table 12: 3.3V to 3.3V Boost Converter Important Parameters

Part	Manufacturer	Part Number	Quantity	Price ($)
Coutx	Kemet	C0603C105Z8VACTU	1	0.01
Ccomp	Kemet	C0603C123J3GACTU	1	0.1
Ccomp2	MuRata	GRM1555C1H102JA01J	1	0.01
Cin	MuRata	GRM31CR61A226ME19L	1	0.09
Cinx	Kemet	C0805C104K3RACTU	1	0.01
Cout	MuRata	GRM32ER60J107ME20L	3	0.22
Css	MuRata	GRM155R60J223KA01D	1	0.01
D1	Diodes Inc.	B540C-13-F	1	0.19
L1	Vishay-Dale	IHLP1212BZERR56M11	1	0.53
Rcomp	Vishay-Dale	CRCW04022K67FKED	1	0.01
Rfbb	Yageo America	RC0201FR-0710KL	1	0.01
Rfbt	Vishay-Dale	CRCW040216K9FKED	1	0.01
Rt	Vishay-Dale	CRCW040260K4FKED	1	0.01
U1	Texas Instruments	TPS55330RTER	1	1.30

Table 13: 3.3V to 3.3V Boost Converter Bill of Materials

6.1.3.3.3 3.3V-1.8V DC-to-DC Voltage Regulator

The 3.3V-1.8V voltage regulator was only used to power the microprocessor which also needs a 1.8V input. The current draw this time did not have to be set to a high value since the 1.8V was not expected to be used for anything else. The following design shown below in Figure 44 was created in WEBENCH® Design Center using a current draw of 1A. The important parameters and bill of materials are both shown below in tables Table 14 and Table 15 respectively. [10]

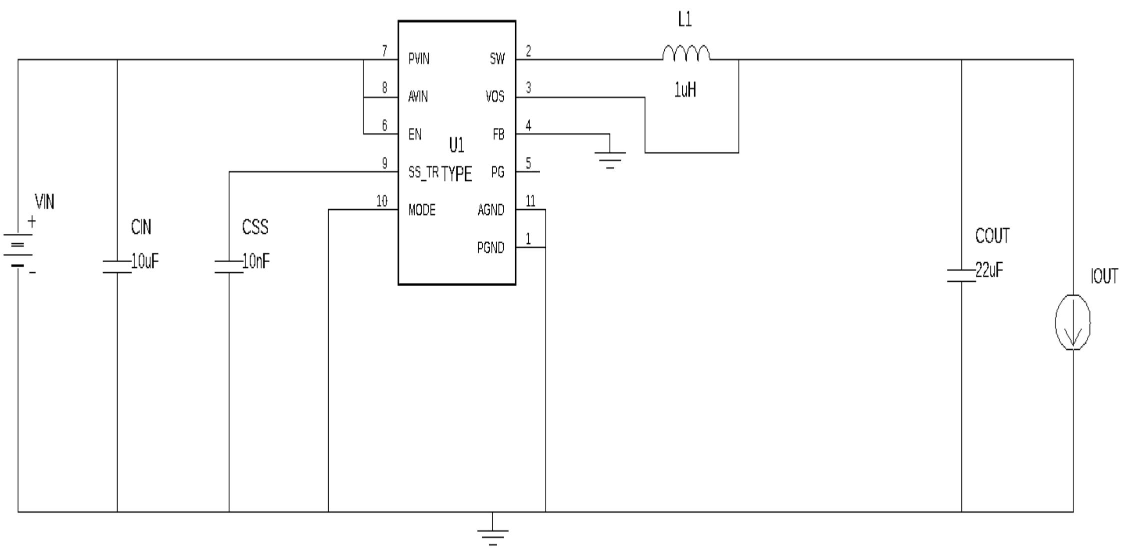


Figure 44: 3.3V to 1.8V Buck Converter Schematic - Schematic Credits to WEBENCH® Design Center

Design	BOM Footprint (mm^2)	Efficiency	BOM Count	I_{out} Max (A)	IC Cost	BOM Cost
2-A High Efficiency Step Down Converter with DCS Control	87	91%	5	2	$0.74	$1.00

Table 14: 3.3V to 1.8V Buck Converter Important Parameters

Part	Manufacturer	Part Number	Quantity	Price
Cin	MuRata	GRM188R60G106ME47D	1	0.02
Cout	MuRata	GRM21BC80G226ME39L	1	0.05
Css	MuRata	GRM033R61A103KA01D	1	0.01
L1	Bourns	SRN6045-1R0Y	1	0.18
U1	Texas Instruments	TPS6209718RWKR	1	0.74

Table 15: 3.3V to 1.8V Buck Converter Bill of Materials

6.1.3.3.4 5V-2.5V DC-to-DC Voltage Regulator

The final voltage regulator would be used to drop down the voltage from 5V to 2.5V DC for the microprocessor input power. The reason behind dropping 5V to 2.5V and 3.3V to 1.8V was to have less of a voltage drop in the regulators. Again, WEBENCH® Design Center was used to create the following design shown in Figure 45 with a 1A output current since 2.5V was only used by the microprocessor. The important parameters and bill of materials are both shown in tables Table 16 and Table 17 respectively. [10]

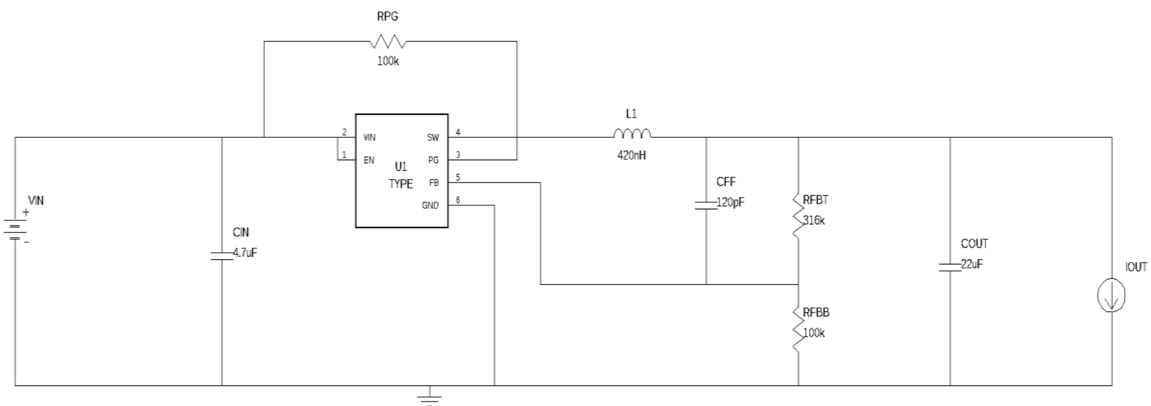


Figure 45: 5V to 2.5V Buck Converter Schematic - Schematic Credits to WEBENCH® Design Center

Design	***BOM Footprint (mm^2)***	***Efficiency***	***BOM Count***	***I_{out} Max (A)***	***IC Cost***	***BOM Cost***
High Efficiency 3-A Step Down Converter	*35*	*92%*	*8*	*3*	*$0.60*	*$0.96*

Table 16: 5V to 2.5V Buck Converter Important Parameters

Part	Manufacturer	Part Number	Quantity	Price
Rfbb	Vishay-Dale	CRCW0402100KFKED	1	0.01
Rfbt	Vishay-Dale	CRCW0402316KFKED	1	0.01
Cff	MuRata	GRM033R71E121KA01D	1	0.01
Cin	MuRata	GRM188R60J475KE19D	1	0.02
Cout	Taiyo Yuden	JMK212BJ226MG-T	1	0.06
L1	Coilcraft	EPL2014-421MLB	1	0.24
Rpg	Vishay-Dale	CRCW0402100KFKED	1	0.01
U1	Texas Instruments	TPS62088YFPR	1	0.6

Table 17: 5V to 2.5V Buck Converter Bill of Materials

In the final design, none of the above voltage regulators were used. Since a battery was no longer being used, power efficiency is no longer an important factor for the design. So, since it is much simpler to use a linear dropout voltage regulator, the group used three of them in the final design as shown in the power connections section.

6.1.4 Sound System

For the sound system, there are two main components that would be processing all audio operations; the soundcard and the main processor. The soundcard would mainly be consisting of an analog-to-digital converter (ADC) to convert the analog signals coming in from the microphone, a digital-to-analog converter (DAC) to convert the digital signals coming out from the processor to analog to be output by the speaker, and finally, a digital signal processor (DSP) that processes all the signals in digital domain. The group's Motor Control Unit already has an ADC that is not being used for much, so that can be used for the microphone-processor communications. While the group's processor has a built-in DSP, it could end up slowing the main processor. An external DSP can be added instead if that occurs. Finally, the DAC ends up being the only component that should be added externally.

6.1.4.1 Recording

For recording, all the communications were done in between the microphone and the main processor. The entire process is illustrated in the flowchart shown below in Figure 46. The microphone transformed the audio signal into an analog signal that was converted into a digital signal using the group's motor control unit ADC. After that, the digital signal was sent to the processor's DSP to be further understood by the processor and stored in the processor's RAM.

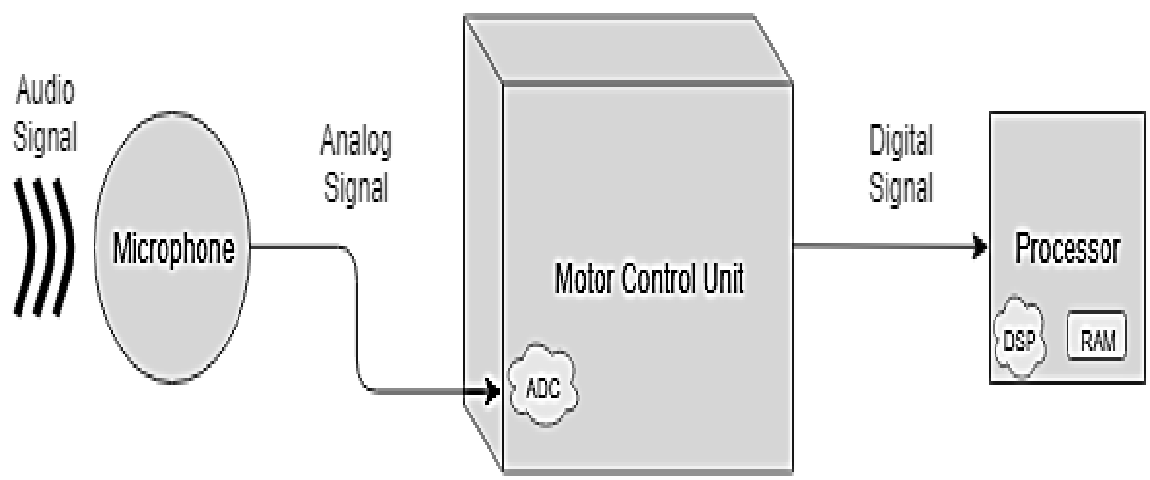


Figure 46: Audio Recording Process

6.1.4.2 Playback

For playback, all the communications were done between the processor, the audio amplifier, and the speaker. The entire process is illustrated in the flowchart shown below in Figure 47. Whether the user wanted to play music through the device, or the device wanted to respond to the user, this is the process it used. The processor sends out the command to an external DAC to produce an analog signal, then amplifies that signal using the group's own mono audio amplifier. Finally, the signal is good to go to be sent out through the speakers for the user to understand them.

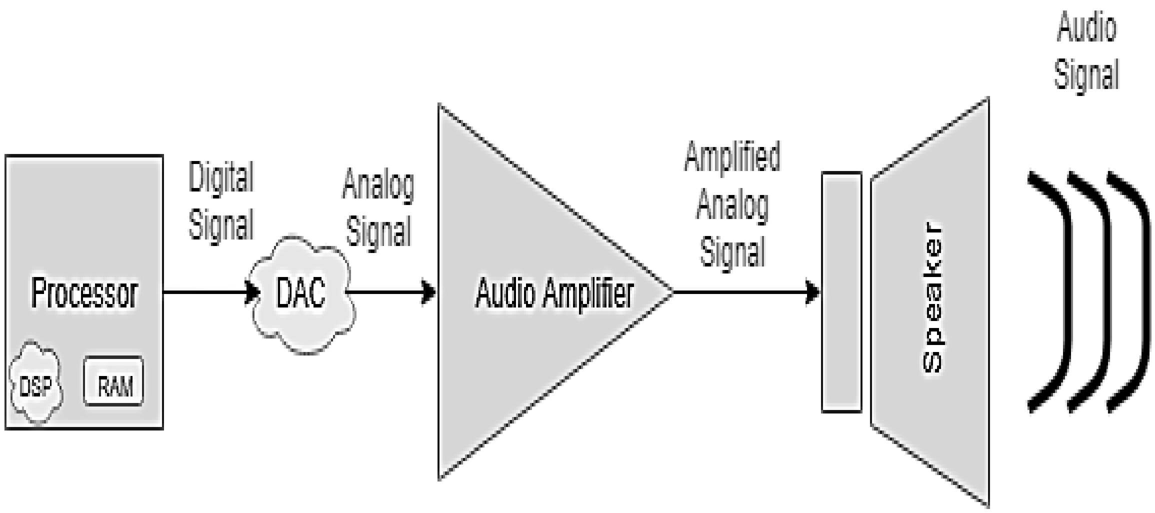


Figure 47: Audio Playback Process

As stated previously, in the final design the above section has changed. The operations were made much simpler by using a mini USB microphone and speakers. They directly communicate with the CPU without the use of DACs and ADCs.

6.2 Software Design

6.2.1 Programming Environment

6.2.1.1 IDE

The IDE of choice for this project was Atom. Atom is, at base, a text-editor that provides a clean, smart interface for the user to easily configure it to their liking. It is extremely lightweight while having a full, comprehensive package manager so it fits the needs of the software that will need to be developed for Ion. Most of the software was developed in Python, and Atom supports that wonderfully. As far as collaboration and version control, which was of huge importance, Atom has built-in Git and GitHub integration to make collaboration more smooth and easy.

6.2.1.2 Revision Control

Revision control was used to track changes to code, merge sections that have been written by different coders, marking stable versions, facilitating collaboration through cloud storage, and creating backup copies. This was done with the decentralized revision control system "Git", using Github.com for cloud storage specifically with a Git GUI called

GitKraken, which provides a graphical interface along with git command capabilities. Providing a GUI for using Git helped to track revisions with more ease.

6.2.2 GUI

The GUI functions as a secondary means of communication between the user and the device, behind voice control. It is the main point of integration between all that Ion can do including voice recognition, face detection and recognition, user data and profiling, applications such as Spotify, YouTube, and internet searches, and editing user preferences and settings. From boot-up to shut-down the GUI handles everything that Ion hears, sees, and says and provides a way for the user to interact with that. Everything in the GUI can be accessed through voice command by an authorized user but having the GUI as a backup allows the user a secondary means of interaction. One last major functionality of the GUI is to display Ion's emotional states as GIFs including smiling, yawning, etc. This provides Ion a fun, interactive way to visually interact with the user.

There were many factors that went into the group having to change the GUI. Firstly, Kivy had a lot of trouble working with the Raspberry Pi, including embedding browsers and videos into the GUI as well as launching windows or browsers from the GUI. The library that the group initially wanted to use to embed windows, browsers, and videos into the GUI was found to be incompatible with ARM processors (including the Pi). The backup plan was to launch a Chromium browser window on top of the GUI, but the window provider for the GUI was very glitchy and would always keep Kivy on top of every other window. The group attempted every workaround including changing the window provider or launching the GUI in windowed mode, but all attempts were either incompatible or too glitchy to be a solid option. After attempting every workaround, the group decided to ditch the effort of trying to embed a browser or video, which was unfortunate.

6.2.2.1 Language and Library Selection

Python is the best language for the functions Ion's GUI handles, as it is event-driven programming rather than the procedural programming that includes languages like C or Java. Using event-driven programming is great for a product such as a GUI because that's all a GUI is - buttons and menus linked together that are driven by touch or mouse input. In the case of Ion, it is driven by touch and allows the user to navigate the GUI. Python also allows for very smooth integration. With the many technologies integrated into Ion, mostly through the GUI, it was important to choose a language that would make integration easy. Integrating the voice recognition, face detection and recognition, as well as the multitude of planned features including a face interface to show emotions and playing YouTube videos or Spotify music, Python provides a way to integrate all of those through the GUI with ease.

The library chosen to create the GUI is Kivy, an open-source Python library that allows for development of touch applications with a natural user interface. It includes a library of touch-aware widgets and hardware accelerated OpenGL drawing which made a great development tool for the GUI. It runs on Raspberry Pi, which was great for both prototyping and incorporating into the actual product which has a Raspberry Pi Compute module.

6.2.2.2 Menu Layout

The design of the menu is meant to be clean, simple, and easy to use. It is a secondary interface, the main being voice control, so it only houses the main settings and interfaces, while the main interaction is through voice on the device. A main functionality of the GUI is to display the human-like emotions of Ion to seem fun and interactive. The main method of doing this is choosing from a library of GIFs to display Ion's reaction and emotion in response to the user's input. A few main facial GIFs were designed to display Ion's face throughout his interaction with the user - examples include waking up, yawning, etc. and are displayed at specified times during interaction with the user. This goes hand-in-hand with the user using the GUI and interacting with Ion.

On bootup, the system initializes and automatically boots up the GUI program. This includes Ion's welcome with a GIF. If the user cannot immediately be initialized and logged in through face detection, the login screen will pop up to log in or register. There is also the option to recover their account or give another shot at Ion detecting their face. The login menu is shown below in Figure 48.

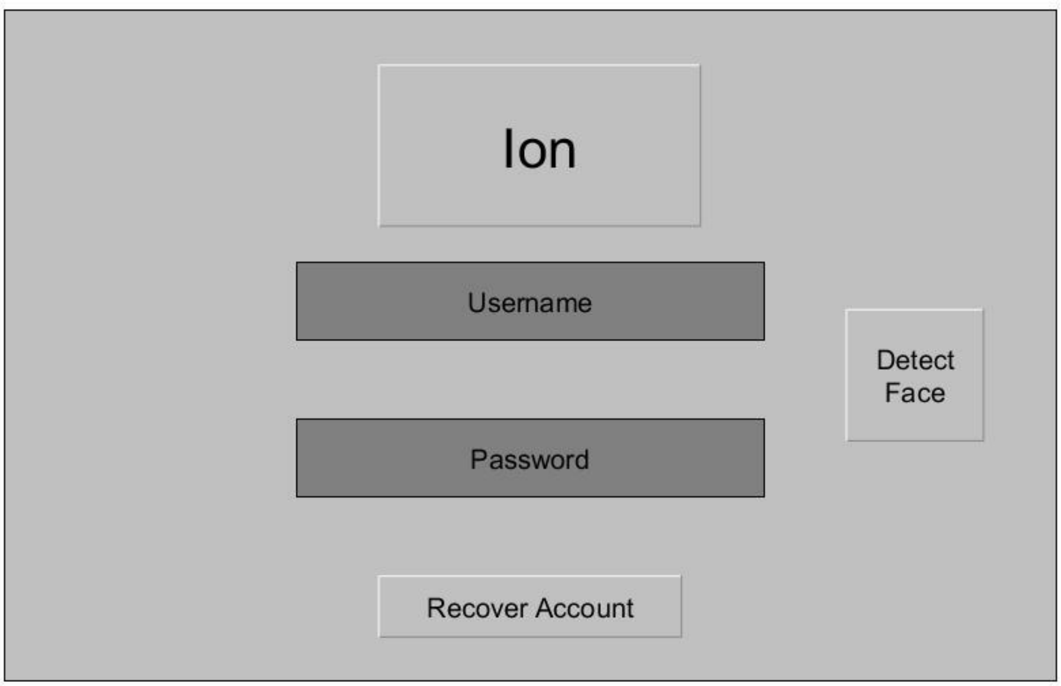


Figure 48: Initial Login Menu Design

After login and initialization, Ion automatically goes to an authorized rest state which waits for commands from the user through voice. There is also an option to enter the GUI manually by tapping the screen. The rest state is shown below in Figure 49.

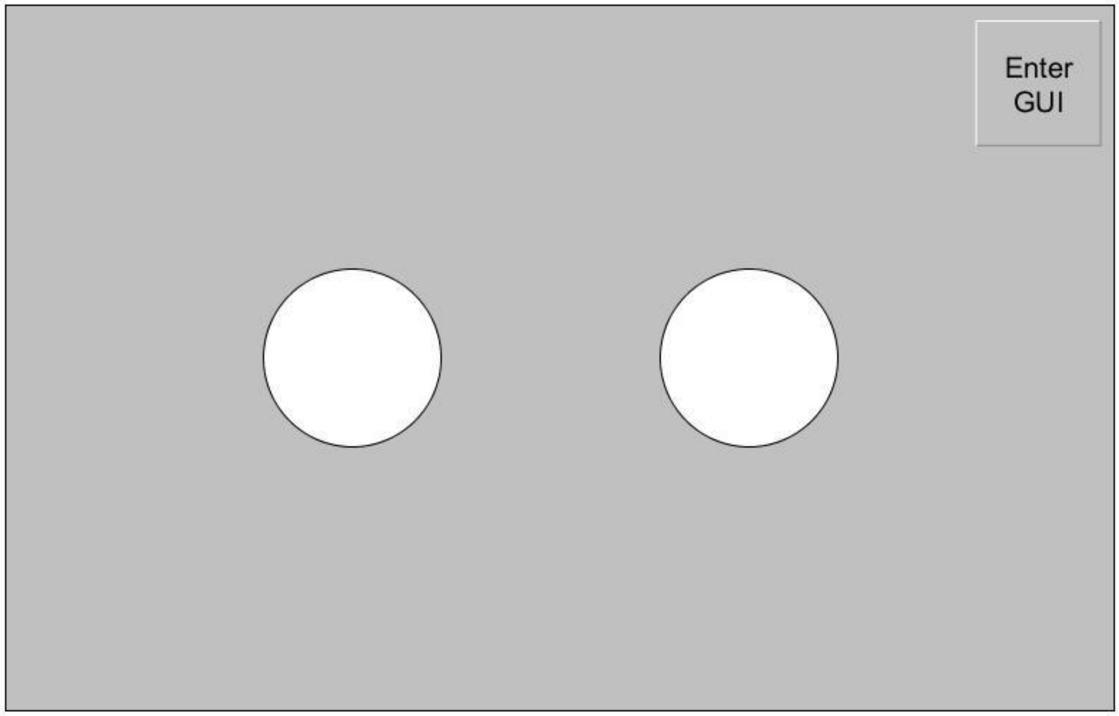


Figure 49: Initial Rest State Design

If entering the GUI manually, the user is able to enter different menus including preferences, account settings, and display settings. The preferences menu can be used to edit personal preferences including Ion's personality choice and external account settings like Spotify or YouTube. Account settings give the user access to editing settings such as email, password, phone number, etc. Display settings let the user change display settings including brightness and contrast. All of this can also be accessed and edited through voice commands. The preferences and settings menu are shown below in Figure 50.

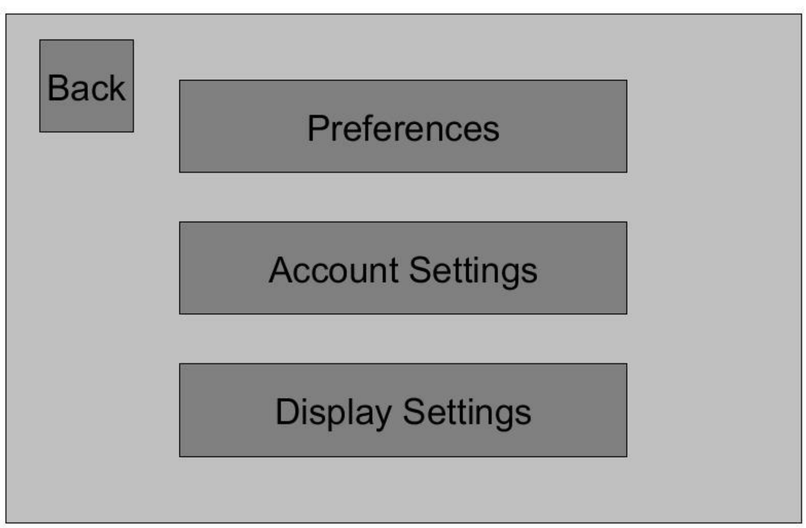


Figure 50: Initial Preferences and Settings Menu Design

The login menu was quickly discarded as the group didn't want to implement any security practices into the project. Much of the time and resources put into the GUI design were spent on trying to implement a web browser or video player into the GUI, to no avail as mentioned above. Once that possibility was discarded, the group focused on implementing a simple GUI system.

Ultimately, a GUI was designed with a welcoming front page which the user can tap to navigate to the main menu page. The main menu page features a clock, a back button to navigate back to the front page, a button to navigate to the commands page to view available voice commands the user can give, and a button used to navigate to a general settings page. These features are all illustrated in the GUI screenshots shown in Figures 51 to 54.

Figure 51: Final Front Page GUI Design

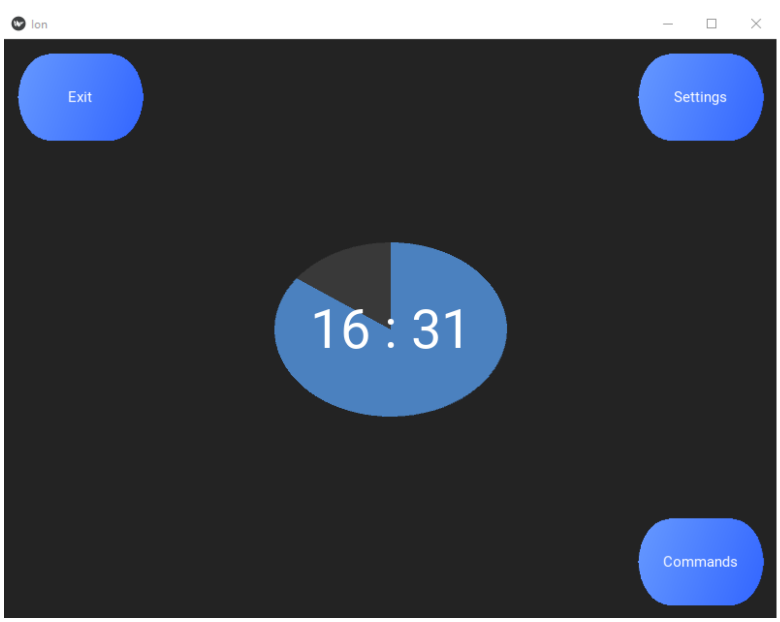


Figure 52: Final Main Menu Page GUI Design

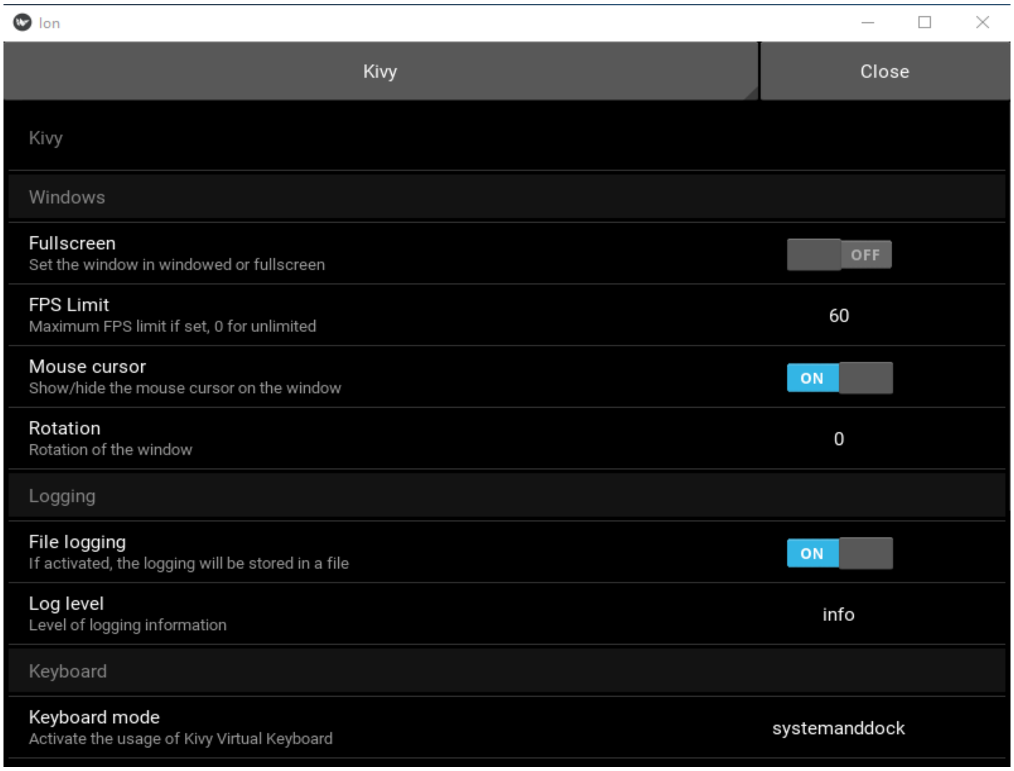


Figure 53: Final Settings Page GUI Design

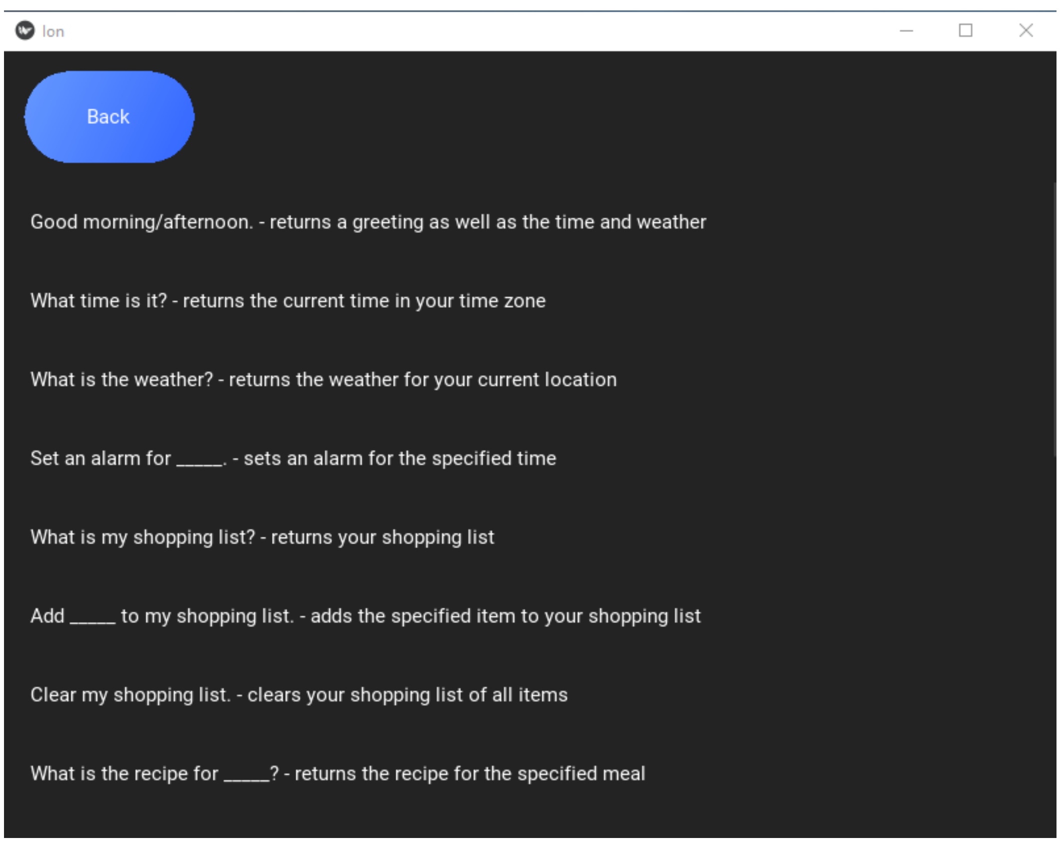


Figure 54: Final Commands Page GUI Design

All of those aspects of the GUI were running on one thread while, on another, the face detection algorithm was running. Incorporated into the GUI were popups that notified the user when a recognized user was detected as well as when an unrecognized user was detected. When a recognized face is detected, the system welcomes the user back. When an unrecognized face is detected, the system notifies the user and displays the photo of the unrecognized face. These are illustrated in the screenshots shown in Figures 55 and 56.

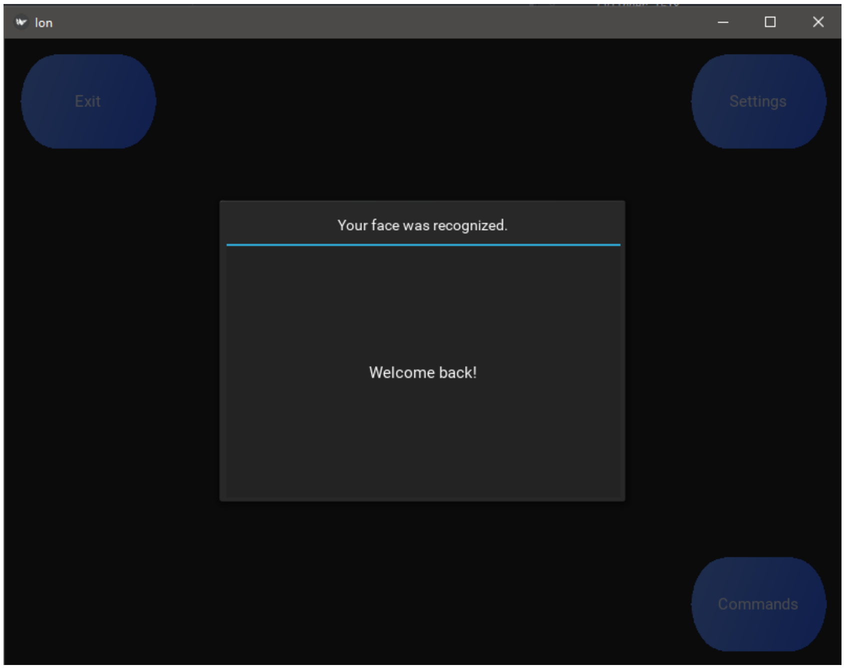


Figure 55: Recognized Face Popup

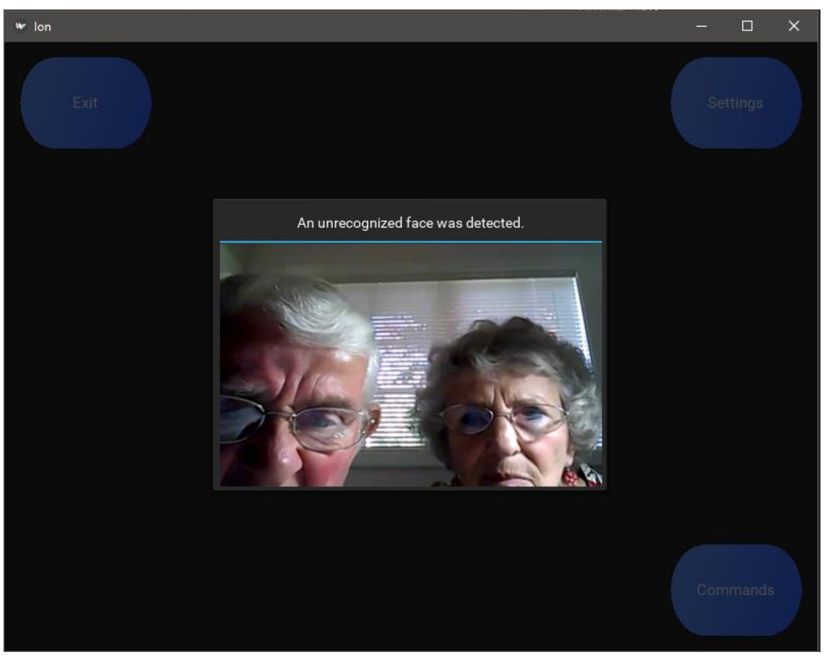


Figure 56: Unrecognized Face Popup

6.2.2.3 Algorithm Flowchart

The Graphical User Interface algorithm is better illustrated in a flowchart that is shown below in Figure 57.

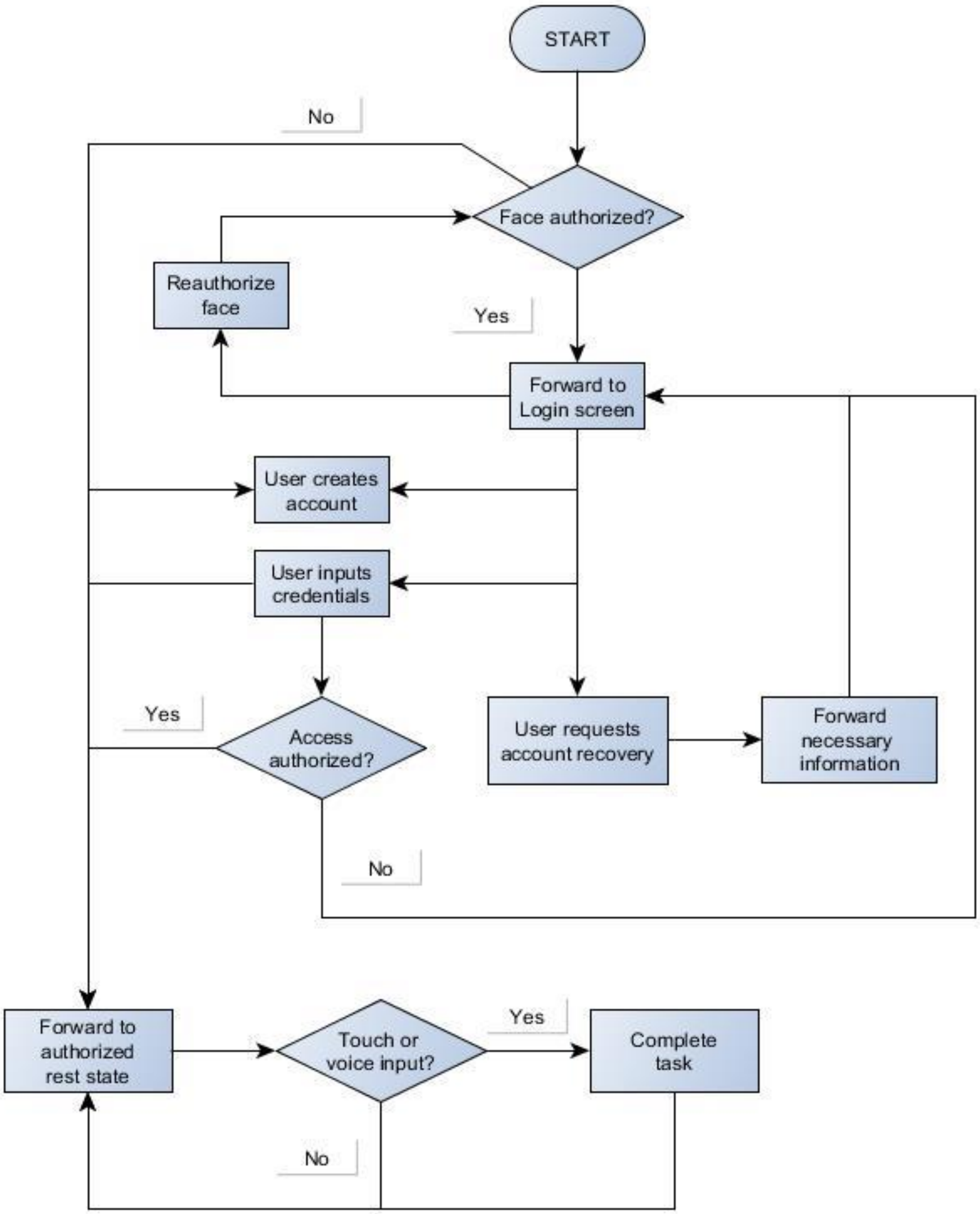


Figure 57: Initial GUI Flowchart

The GUI flowchart had to be altered to account for the changes mentioned above. The updated flowchart is illustrated in the Figures 58 and 59 next page.

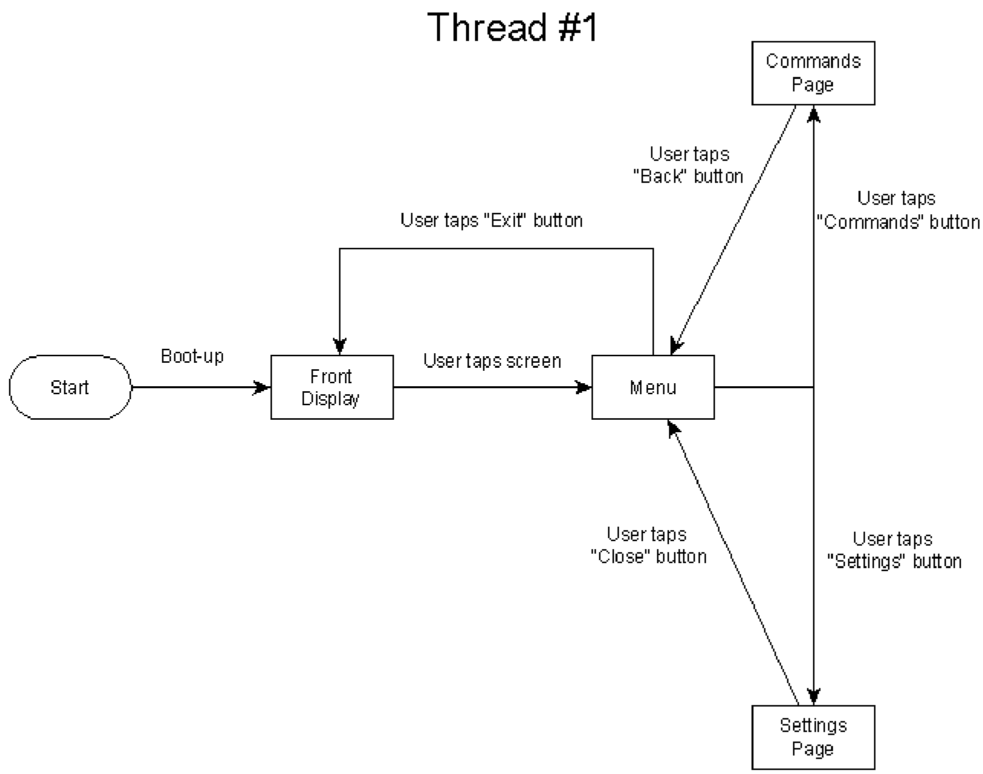


Figure 58: Final GUI Flowchart

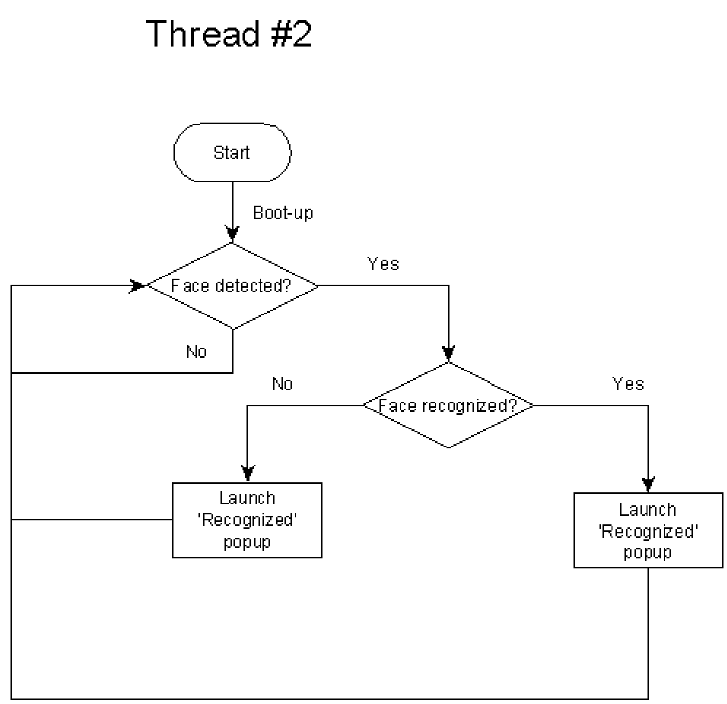


Figure 59: Final Face Detection GUI Popup Flowchart

6.2.3 Voice Recognition

6.2.3.1 Framework

The voice recognition program is implemented using Google Speech API. Bash or Python scripts could've been used to create an API request in the form or a JSON file. That JSON file can be processed by Google Speech API and send a JSON file back to Ion to be processed into a command that Ion can fulfill.

6.2.3.2 Key Hardware Constraints

The quality of the microphone posed a concern if it can't clearly make out speech in different environments and volumes. However, most microphones are usually decent enough to handle voice recognition and speech recognition software has come very far in the past couple years, so much so that a decent microphone didn't pose a problem. Processing power may have been an issue, as the voice recognition software needs to run at all times as well as process the information very quickly to reduce latency and provide a quick interaction.

6.2.3.3 Algorithm Flowchart

The voice recognition algorithm flowchart is illustrated in a flowchart shown below in Figure 60.

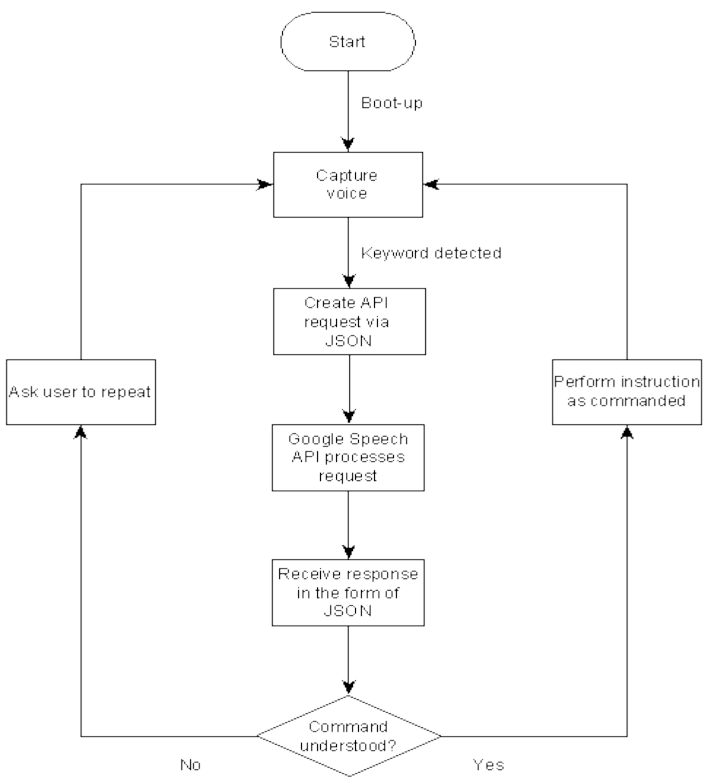


Figure 60: Voice Recognition Algorithm Flowchart

6.2.4 Computer Vision

Computer vision is one of Ion’s key components. Its components are necessary to detect faces, provide feedback to the MCU, and recognize the user.

6.2.4.1 Overview

Facial recognition and tracking are used for several different things:

- Identify the user
- Detect the user’s location

To identify the user, a face needs to be detected on a video feed and compared to a database. If a match is found, a prompt will confirm with the user whether the identification was correct or not. Once this has been done, the face tracking feature will take over until the face has been out of sight for an amount of time that can be adjusted by the user. The face tracking feature determines the position of the face relative to the picture frame and stream its location. On a high level, both the algorithms are combined as follows in Figure 61 next page.

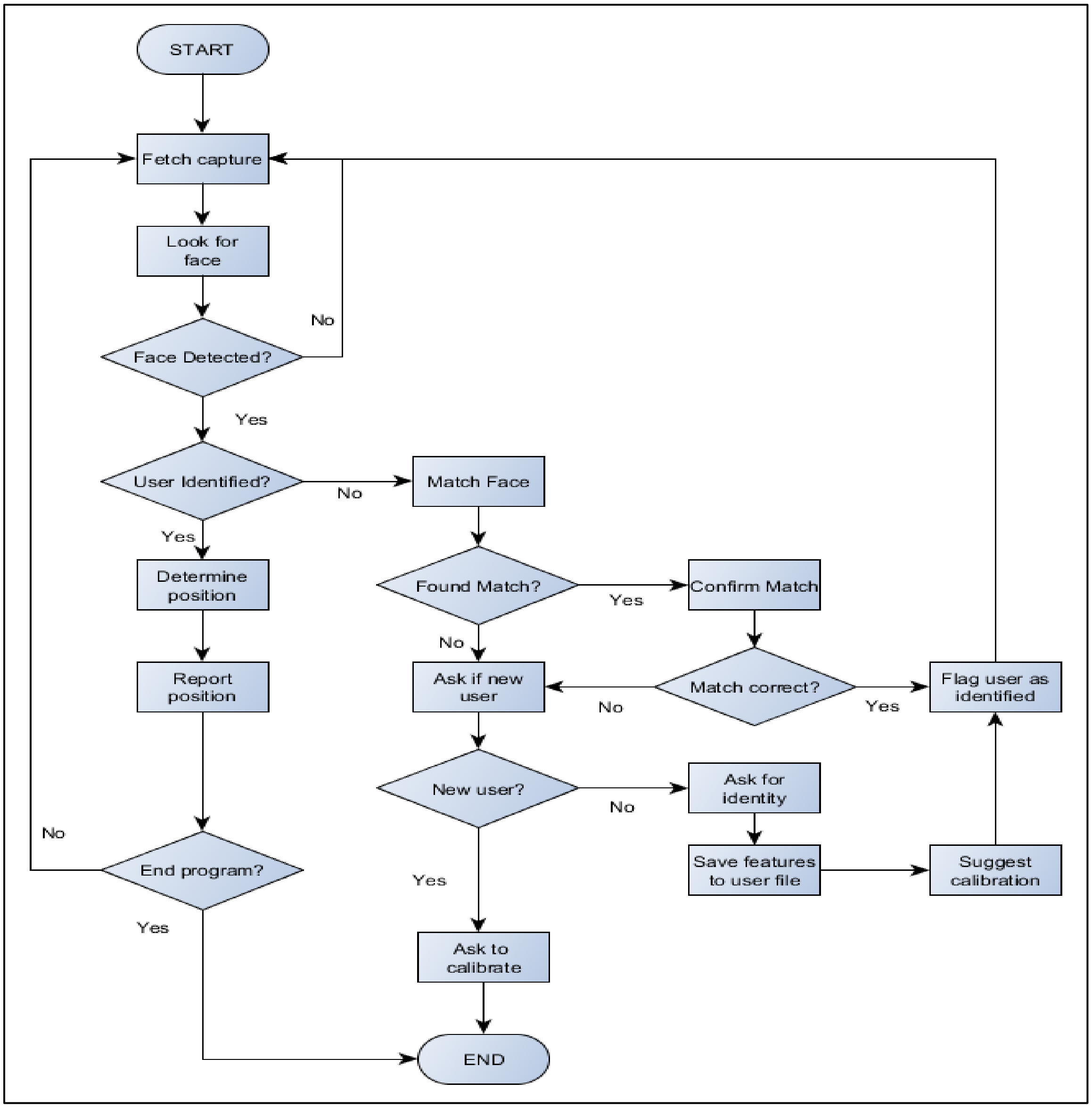


Figure 61: CV flow diagram

6.2.4.2 Facial Detection

The facial detection software's purpose is to detect faces in frames captured by the camera. It is not supposed to identify the user. For user identification, see section 6.2.4.3 - Facial Recognition.

6.2.4.2.1 Framework

The facial recognition program is written in Python and makes use of the OpenCV library. Python was chosen because of its readability and portability. It is pre-installed in the Linux operating system and has a broad spectrum of useful libraries that will be used for Ion.

Most programming was done in the Atom editor and prototyped on a PC with the help of a webcam.

6.2.4.2.2 Key Hardware Constraints

The input from the camera posed some difficulties in low-lighting conditions, but a 640x480 video feed with a framerate of 30Hz is more than enough. The main hardware constraint for facial detection is processing power. Since the facial detection software is a key component of the servo feedback system, lag had to be reduced as much as possible, preferably under 100ms. In order to free up processing power for other programs, facial detection does not run continuously.

6.2.4.2.3 Reference Designs

Two facial detection methods were considered: Haar classifiers and linear binary patterns (LBP) classifier. Both of these methods have key advantages and disadvantages. Haar tends to be more accurate and have a low false positive rate. This would reduce the chance for Ion to mistake an object for a person. However, Haar is slower and has problems in difficult lighting conditions. LBP on the other hand is fast and has less problems with changes in lighting conditions. The drawback is a loss in accuracy.

6.2.4.2.4 Possible Implementation

The two most important factors are speed and accuracy, so it is difficult to select one method over the other. One possibility was to create code for both methods and switch depending on processor load or lighting conditions. Another possibility was to test performance of both methods in conjunction with a mockup system and choose the method that provided a better user experience. No matter which facial detection algorithm was chosen at the end, the high level flowchart in Figure 62 below was going to be the same.

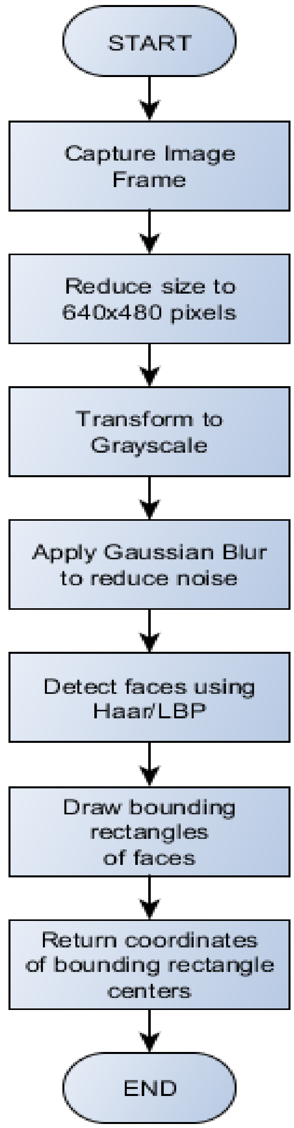


Figure 62: Facial detection algorithm flowchart.

6.2.4.3 Facial Recognition

The facial recognition software takes the output from the facial detection software (bounding rectangle of the detected face and the image in which it was detected) and tries to match the detected face to a known user.

6.2.4.3.1 Framework

As with the facial detection program, the facial recognition program is written in python and makes use of the OpenCV library. Python was chosen because of its readability and portability. It is pre-installed in the Linux operating system and has a broad spectrum of useful libraries that will be used for Ion. Most programming was done in the Atom editor and prototyped on a PC with the help of a webcam.

6.2.4.3.2 Key Hardware Constraints

Facial recognition relies heavily on matrix calculations, so the amount, speed, and availability of multiple CPU cores will determine how fast and accurate this program will be. The minimum required resolution from the camera is 640x480 pixels.

6.2.4.3.3 Design Considerations

Several designs were taken into consideration and their pros and cons are weighed below.

6.2.4.3.3.1 Eigenfaces

Principal Component Analysis (Eigenfaces) uses a training set of pictures taken in the same lighting conditions. Eigenvectors and eigenvalues will then be computed from a covariance matrix that represents the set of training pictures. These so-called eigenfaces can then be used to compute the difference of a face to these eigenfaces. Comparing these deltas of various pictures will generate a match (within determined range) if the face in the pictures is the same. Because of the reliance in even lighting conditions, this method requires a controlled environment. All compared pictures must be of the same size and normalized [11].

The advantage of using eigenfaces is that it is very fast, does not need a lot of storage space, and is very accurate when the lighting conditions are ideal and unchanged. It also does not need a very high-resolution image which can speed up processing time and reduce storage need even further. The drawbacks are that it handles varying lighting conditions, change in facial expression, varying background, and faces of different size poorly. The recognition depends on lighting rather than facial features.

Because of its drawbacks, the image is cropped around the face-binding rectangle produced by the facial detection algorithm. This reduces the effect of the background. The cropped image is then resized to a predetermined size, which should normalize it (face centered in the image). If that was not enough, it was considered to apply a radial mask around the face to reduce interference of background even further. To reduce the errors when the user's face is tilted or rotated, several different sets of eigenfaces could've been generated by asking the user to take several pictures from varying positions when calibrating the system.

6.2.4.3.3.2 Fisherfaces

Linear Discriminant Analysis, also known as fisherfaces, is an additional step beyond eigenfaces that takes into account different classes (specific features) found in the image and the variance in those features from one image to another. This reduces the eigenfaces' heavy reliance on a controlled environment. [12] The strategy to reduce errors with this method is the same as eigenfaces. It is especially important to reduce image size to reduce processing power and storage requirements.

6.2.4.3.3.3 Histograms

This method involves creating a histogram from the image containing the face and calculating the delta to histograms from images in the database. The closest match will have the lowest delta will be considered a match if the delta is within a determined range. This method will not be implemented as pure facial recognition. It was considered to be added as an additional step after a face has been detected, isolated from an image, and a decision needs to be made between two highly likely possible matches.

6.2.4.4 Comparison

Table 18 below shows a comparison of the described methods' pros and cons.

Method	Pro	Con
Eigenfaces	- Less space in the database - Speed (Real time facial recognition) - Can use reduced image sizes	- Sensitive to lighting, scale and translation - Difficulty with expression changes - Differences in background greatly affect accuracy - Depends on illumination, rather than facial features
Fisherfaces	- More accurate than eigenfaces (Less susceptible to lighting, orientation, and size)	- Still susceptible to lighting conditions and varying facial expressions - Requires large storage - Requires more processing power
Histogram	- Easy to implement - Fast - Low computational requirement	- Shapes are not taken into consideration - Inaccurate under different lighting conditions

Table 18 Comparison of facial recognition methods

Based on this comparison, and the fact that the environment's lighting wasn't ideal, fisherfaces was used for facial recognition. Once the software was written, prototyping showed how accurate it is and there was no need for a second verification step. The size of the image that was processed was tweaked to balance between accuracy and processing speed.

6.2.4.5 Algorithm Flowchart

Figure 63 below shows the flowchart of the facial recognition algorithm.

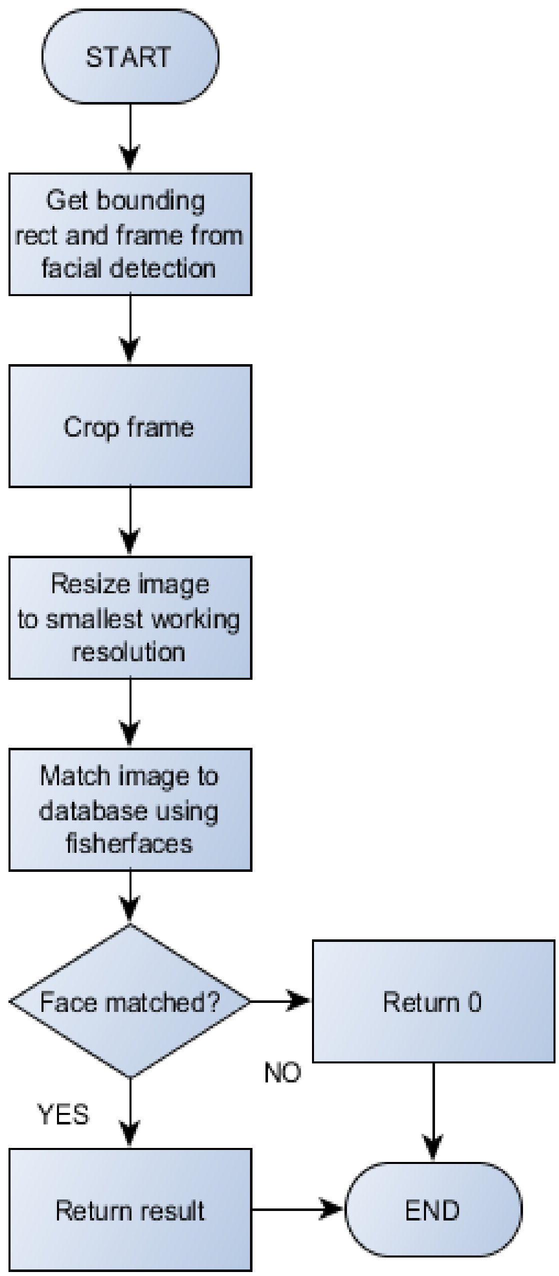


Figure 63: Facial recognition algorithm flowchart

7.0 Project Prototyping and Coding

7.1 PCB

7.1.1 Prototyping Hardware

7.1.1.1 Controller

The Controller (Raspberry Pi Compute Module 3L) was attached to the PCB via a SODIMM connector. Before the PCB was created, prototyping was done with the help of a Waveshare Compute Module IO Board Plus (see Figure 64 below).

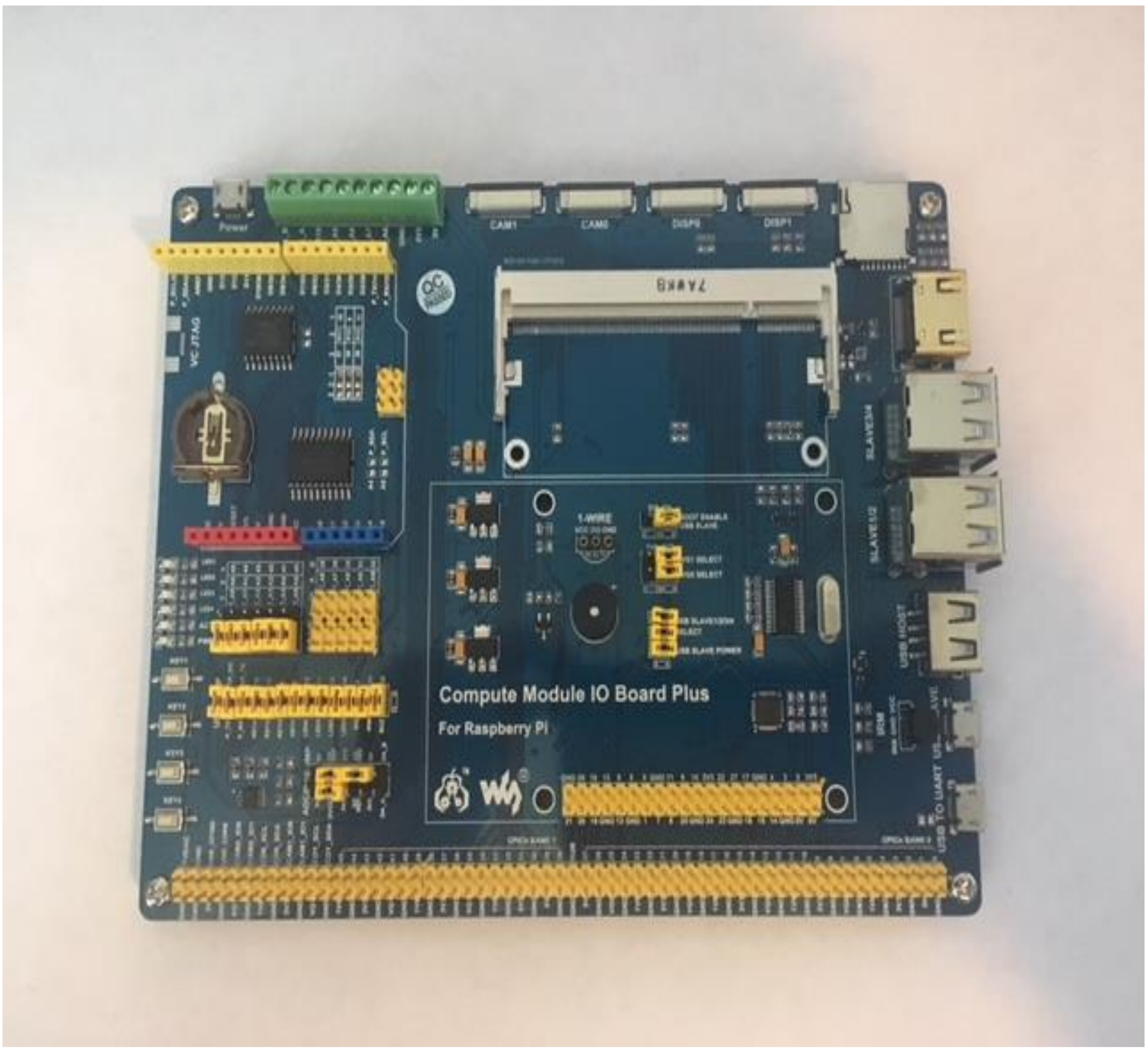


Figure 64: Waveshare Compute Module IO Board Plus

The IO Board has several built-in components that were be necessary for testing: An SD-card slot, ADC conversion, DAC conversion, Arduino headers, CSI interface, DSI interface, Compute Module GPIO header, an RTC chip for I2C, LEDs, Keys, USB connections, and a Buzzer.

The SD card slot was used to load the OS and other software. ADC and DAC conversion were used for sound processing. The CSI and DSI connectors were used to connect the display and camera. An Arduino was connected via I2C, using the Arduino headers to prototype the MCU. All other components mentioned were used for software testing, code prototyping and troubleshooting.

Once the prototyping was completed, the Compute Module and SD card was transferred to ION's PCB.

7.1.1.2 MCU

The MCU was prototyped with the help of the Waveshare Compute Module IO Board, shown previously in Figure 64, an Arduino UNO R3, a small Hextronik HXT900 servo and a battery adapter for AA batteries (see Figure 65 below).

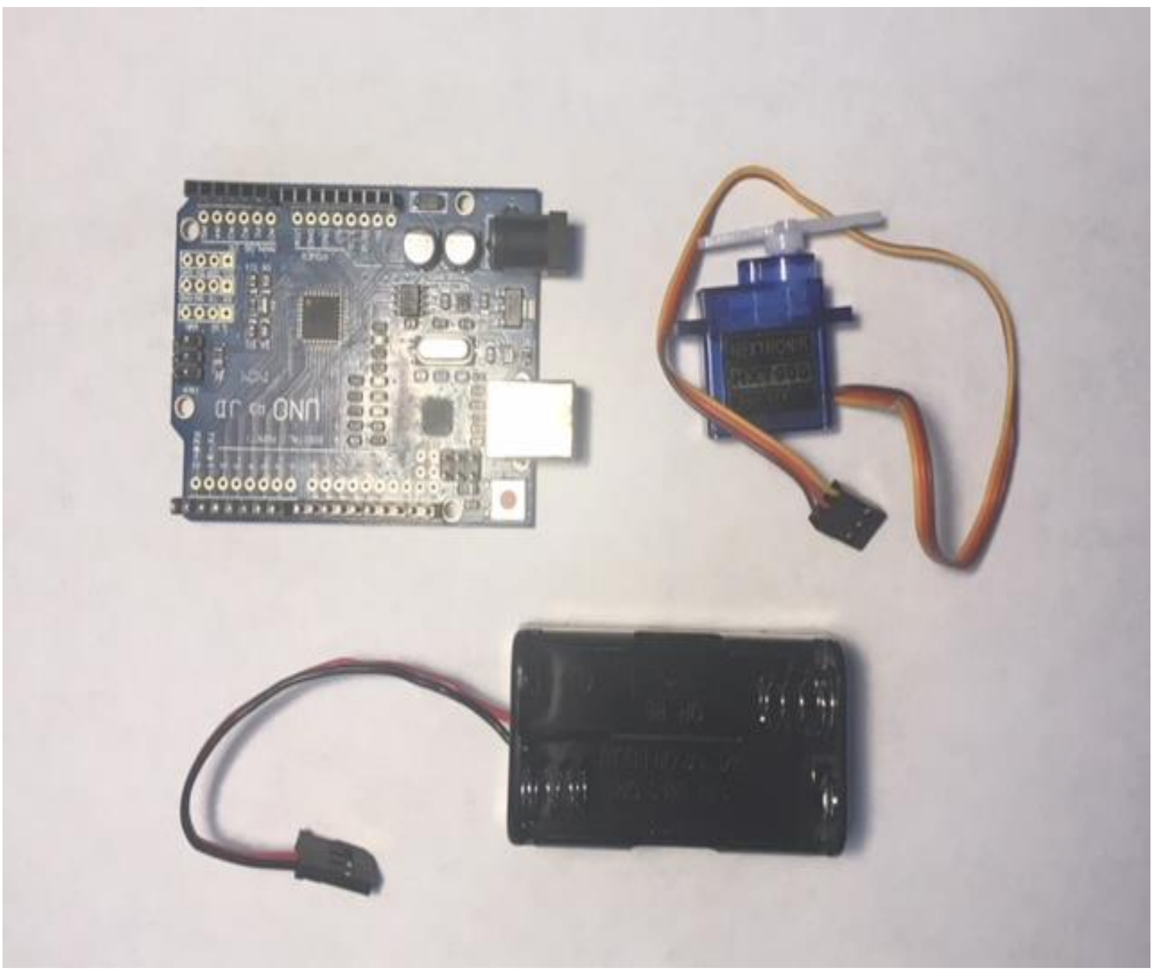


Figure 65: Arduino UNO R3 (top left), Servo (top right) and battery adapter for MCU prototyping

The Arduino Uno was connected to the Waveshare IO board via I2C and will also receive its power from the IO board. The servo receives a PWM signal from the Arduino and power from the battery adapter. The battery adapter contains three AA batteries to provide the servo with 4.5V. A Camera could've also been connected to the IO board to test the feedback system but was not necessary for initial MCU functionality testing.

Once the setup worked satisfactory, the working code was loaded onto an ATMega328 chip to be used in the final PCB design.

7.1.2 Vendor and Assembly

The PCB is the core of the project, if it fails the entire device altogether fails. So, it was necessary to pick the right vendor along with whether the group wanted the PCB printed and components mounted on top by the vendor, or whether the group wanted just the PCB delivered and then solder the components by hand. To compare vendors, the key things to look at were the quality of the PCB, the quantity of PCBs compared to their cost, shipping duration, and vendor reputation.

7.1.2.1 PCBWay

PCBWay is a PCB manufacturer from China. Its reputation is pretty good amongst other senior design groups that have already finished printing and dealing with PCBs. Even though they ship the PCBs from China, it only takes 3-4 days to deliver along with a high shipping fee. There are also express options to speed up the delivery if necessary. Compared to other PCB vendors, they provide good quality PCBs that go for around $39 per ten boards with the dimensions of 100mm x 100mm and two layers. With the regular shipping option, the total ends up being around $60 for ten PCBs.

7.1.2.2 4PCB

4PCB is stationed in the United States, which would be a great option for fast and low-cost shipping. They have multiple PCB options, but compared to PCBWay, they seem to be very expensive. Some of their options range from PCBs costing $33 each or $66 each. 4PCB also provides the option to create a custom shape for the PCB. As far as quality, their PCBs are of high quality, but that comes at a high cost.

7.1.2.3 Gold Phoenix Printed Circuit Board Co., Ltd

Gold Phoenix is a well-known PCB manufacturer also stationed in China. They are known for providing cheap PCBs to SparkFun and Microsoft. However, compared to PCBWay, they are not so cheap. Based on reputation, their PCBs are sometimes flawed, so they must be tested carefully. As far as customization, they provide a wide range of options, but the total cost of one PCB ranges from $75-$100 without any customization, which is still too much. Gold Phoenix also provides free shipping to North America.

7.1.2.4 OurPCB

OurPCB is another Chinese PCB manufacturer that provides cheap PCB services. The cost of ten PCBs with two layers and the dimensions of 100cm^2 go for around $36 and take around 4 days to manufacture. The cost is great compared to the other PCB vendors. However, based on reputation, their PCBs sometimes tend to be flawed as well. That might not be too much of a problem given that there could be an order of ten PCBs and only two of them might be flawed, so there will still be plenty of PCBs to work with. Their PCBs are either hand-made or machine-made. As far as shipping goes, it is definitely not free. They use the same shipping methods as PCBWay, so it is expected that shipping costs end up being high.

7.2 Schematics

7.2.1 Controller I/O

To better plan out the schematics for the PCB, a table was first created listing all the inputs and outputs. After knowing all the inputs and outputs, it was simple to create the schematics. The controller I/O are shown below in Table 19.

Controller: Raspberry Pi Compute Module 3	
Inputs	Outputs
Analog-to-Digital Converter (Microphone)	Digital-to-Analog Converter (Speaker)
LCD Screen Feedback	LCD Screen
SD Card	SD Card
WiFi Chip	WiFi Chip
Camera Feedback	Camera
Power	Power
	LEDs
	Motor Control Unit

Table 19: Controller Inputs/Outputs

7.2.2 Reference Schematics

After finishing the controller I/O, there were some devices such as the WiFi SMT module ESP-12S, the digital-to-analog converter MCP4725, and the microSD card reader that were complicated to design entirely from scratch. So, some reference schematics were used with all credits to Adafruit for providing them. Those schematics were mimicked and modified to fit the group's own PCB schematic. For example, there were some components that were not required for the group's PCB, such as voltage regulators, that could be removed since the controller GPIO itself can regulate the amount of voltage being input into other components. The ESP-12S, MCP4725 breakout board, and the microSD Card reader breakout board schematics are shown below and next page in Figures 66, 67, and 68 respectively.

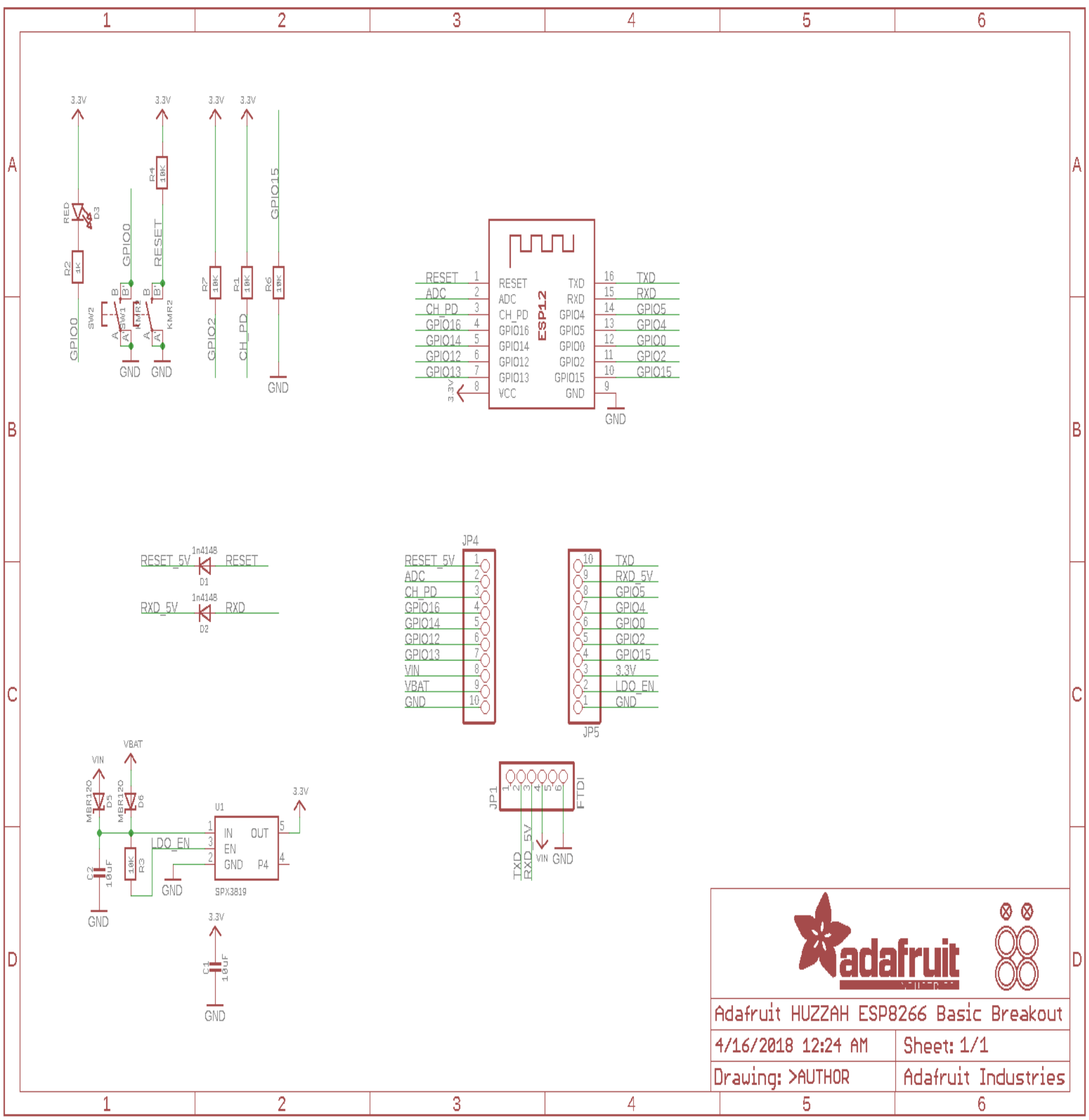


Figure 66: ESP8266 SMT Module Schematic - Schematic Credits to Adafruit

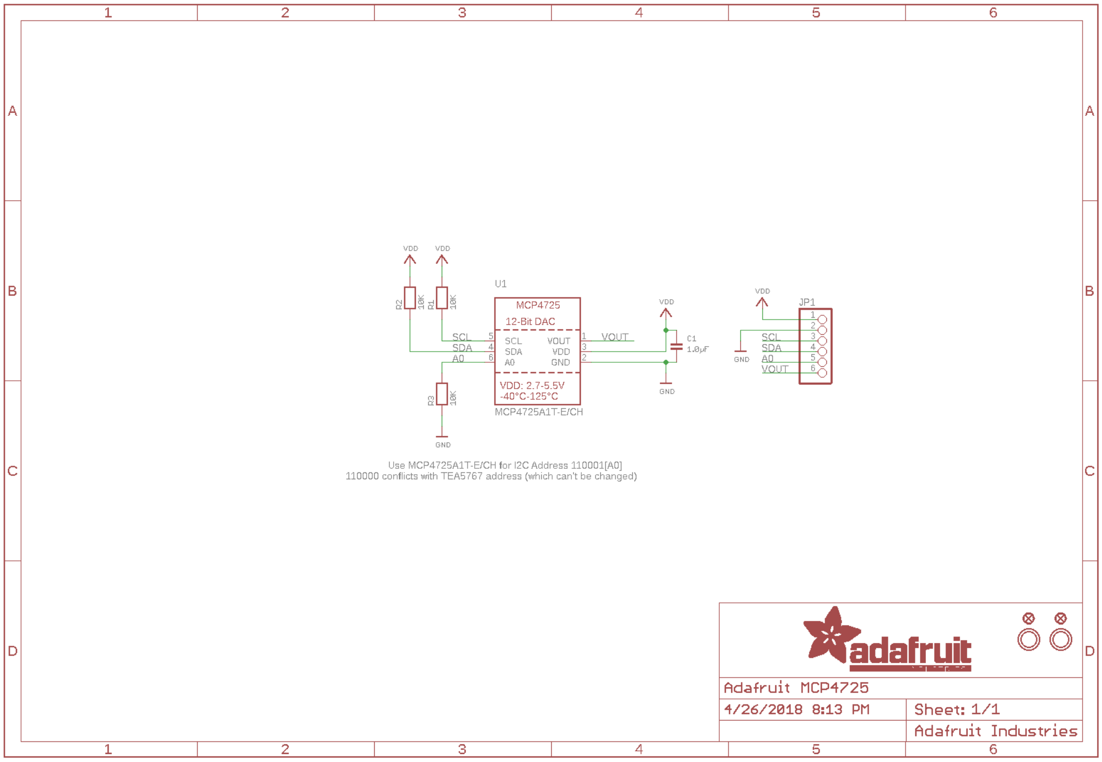


Figure 67: DAC MCP4725 Breakout Board Schematic - Schematic Credits to Adafruit

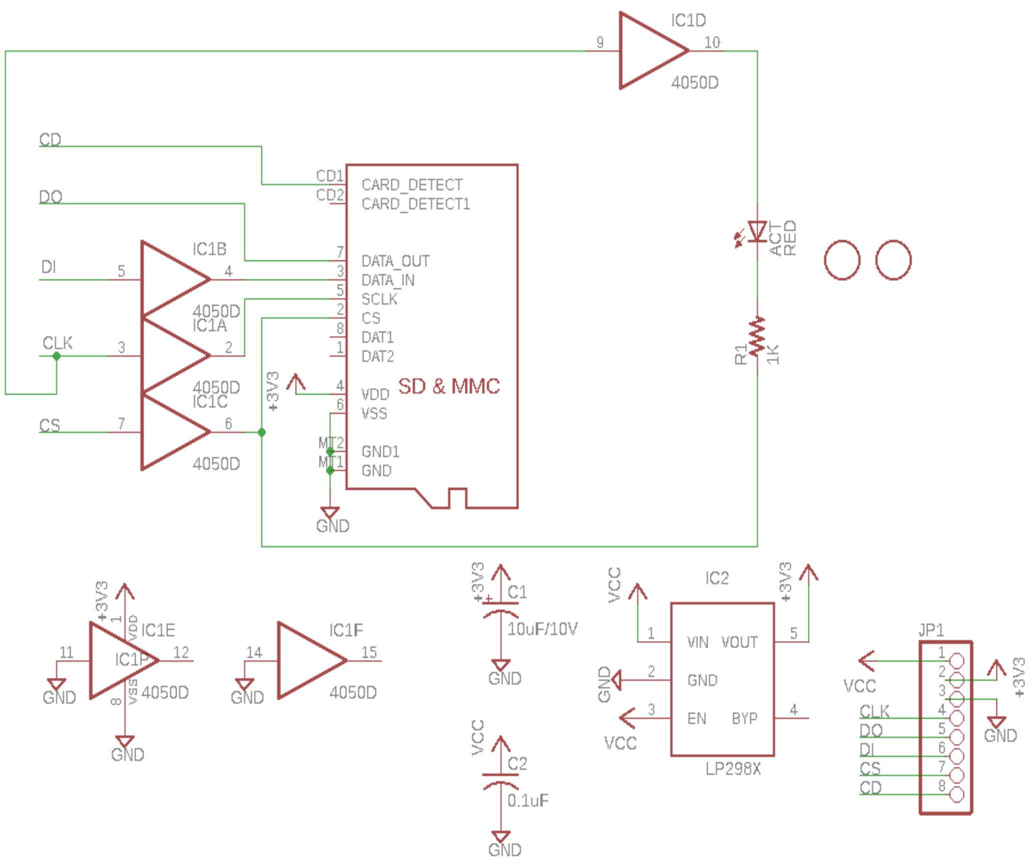


Figure 68: MicroSD Card Reader Breakout Board - Schematic Credits to Adafruit

7.2.3 Overview Schematic

The overview schematic was created using Autodesk® EAGLE by following the controller inputs and outputs and the power connections. The battery was input into the voltage regulators shown in section 6.1.3.3 of this document, that then created four voltage sources; 1.8V, 2.5V, 3.3V and 5V. Those all went into the processor and whatever another component requires 5V or 3.3V could be directly powered. Things to note are that the microprocessor was powering the WiFi chip using pin 36 (GPIO31 – 3.3V), the camera CSI using pins 76 and 78 (GPIO42 & GPIO43 – 3.3V), the LCD screen DSI using pins 75 and 77 (GPIO22 & GPIO 23 – 3.3V), the micro SD card reader using pin 45 (GPIO12 – 3.3V), and finally the DAC using pin 27 (GPIO8 – 5V). The resulting PCB overview schematic is shown below in Figure 69.

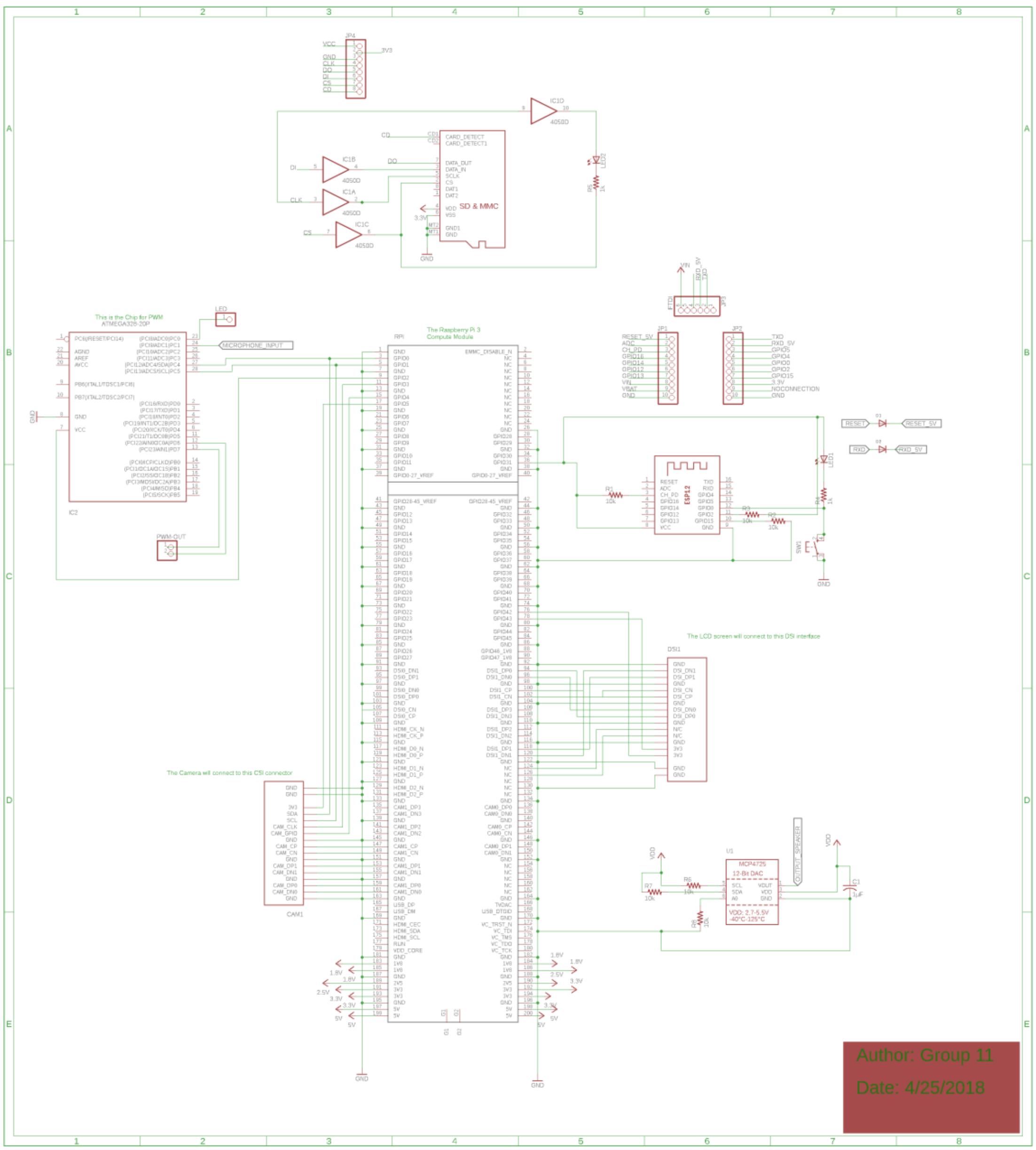


Figure 69: PCB Overview Schematic

For the final design, since a lot of changes were made, a new PCB schematic was used, shown below in Figure 70.

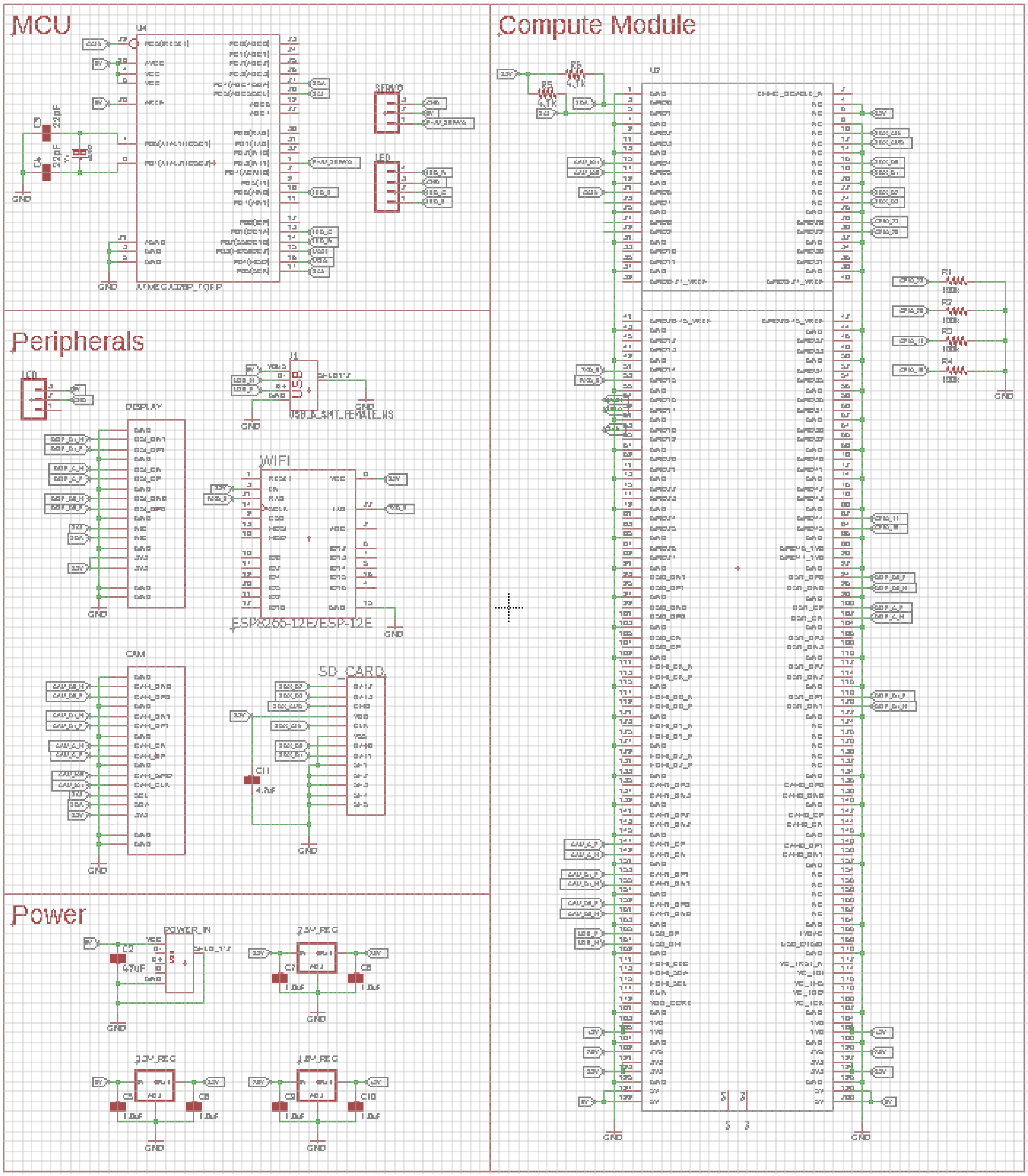


Figure 70 Final Design PCB Schematic

7.2.4 Bill of Materials

After completing the overview schematic, it is important to have a bill of materials to keep track of the required components, quantities, and costs. The bill of materials for the overview schematic is shown next page in Table 20.

Part	Manufacturer	Part Number	Quantity	Price ($)
10k Resistors	Vishay / BC Components	MCS0402PD1002DE500	6	0.63
1k Resistors	Vishay / BC Components	MCS0402PD1001DE500	2	0.63
Capacitor	AVX	12066C105KAT2A	1	0.30
Chip LEDs	OSRAM Opto Semiconductors Inc.	LG R971-KN-1	2	0.05
Diodes	ON Semiconductor	NSVR0170HT1G	2	0.20
Buffers	Nexperia	HEF4050BT,653	4	0.41
Switch	C&K	KMR221GLFS	1	0.49
1x1 Header	AVX	009296001603806	1	0.38
1x2 Header	Molex	505567-0271	1	0.48
1x6 Header	Molex	505567-0671	1	0.57
1x8 Header	Molex	505567-0871	1	0.61
1x10 Header	Molex	505567-1071	2	1.21
DAC	Microchip Technology	MCP4725A0T-E/CH	1	1.02
MicroSD Socket	4UCON Technology Inc.	15882	1	1.95
WiFi Chip	Espressif Systems	ESP-12S	1	6.95
CPU	Raspberry Pi	CM3	1	53.85
MCU	Microchip Technology / Atmel	ATMEGA328-20P	1	2.50
Total			29	78.70

Table 20: Initial PCB Bill Of Materials

8.0 Prototype Testing

8.1 Hardware Testing

8.1.1 Testing Environment

To test the hardware components, the testing had to be done in a safe environment in case anything went wrong including electrical surges, circuits blowing up or components catching on fire. The appropriate environment the group decided to test the hardware components was the Senior Design lab at the University of Central Florida. The lab had plenty of safety tools, along with a lot of equipment available for use. Equipment included 3-output DC supplies, digital multimeters, oscilloscopes, function generators, computers, resistors, capacitors, and plenty of space to work with.

8.1.2 Speakers

Testing the speaker was key to whether the speaker quality was good enough for the design, whether the number of speakers was enough, and whether there were defects in the speaker. Before testing the speaker, there had to be some soldering done, two wires should be soldered to the speaker terminals; one at each terminal. Then a continuity check had to be done to test whether the wires were soldered properly, and that the speaker still works. For the speaker to pass the continuity check, the Digital Multimeter had to beep once connected to both terminals of the speaker. Equipment used to test the speaker are shown below in Table 21.

Component	Equipment & Tools
Speaker	• Function Generator • Soldering Tools • Digital Multimeter

Table 21: Speaker Testing Equipment & Tools

To test the speakers, first, a very simple circuit had to be drawn. Since the speaker being used by the group has a maximum power intake of 0.5W and has an impedance of 8 ohms, it was simple to find the maximum voltage and current that could be supplied to the speaker. The general power equation P=VI was used, then using Ohm's law to substitute the voltage of a resistance into the equation to find the current. The equation ends up being $P=I^2R$, which results in a maximum current of 0.25A and a maximum voltage of 2V across the speaker. The resulting circuit is shown next page in Figure 71.

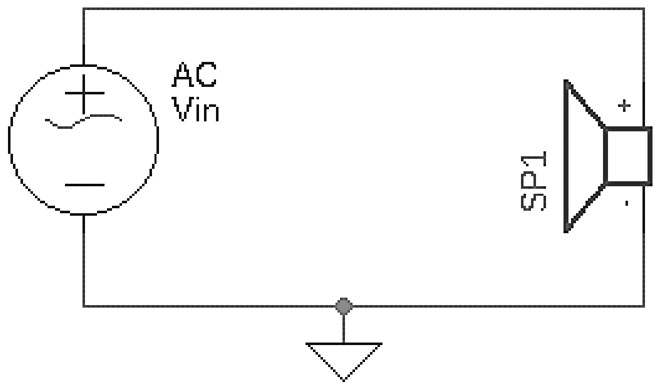


Figure 71: Speaker Testing Circuit

The speaker could've been tested in many ways, including building a circuit on breadboard with a sound amplifier for example. But, the fastest and easiest way to test the speaker was to connect the speaker directly to a function generator and generate random AC signals. The function generator was set up to produce a sine wave with 4V peak-to-peak, and a 1Hz frequency signal. After that, the group kept on varying the frequencies to 200Hz, then 500Hz, and finally 1KHz to test whether the speaker produces sound or not, and whether the sound quality is loud and clear enough for the design. The table shown below, Table 22, illustrates the testing plan and expectations. If the expected results were not met, that means it is a failed test.

Waveform	Input Voltage (V_{pp})	Frequency (Hz)	Expected Results
Sine Wave	4	1	Beeps every 1 second
Sine Wave	4	100-500	Continuous noise
Pulse Wave	4	1	A beep every 1 second
Pulse Wave	4	100-500	Loud pulses
Sine Wave	2	100-500	Lower sound compared to $4V_{pp}$

Table 22: Speaker Test Expectations

8.1.3 Microphone

Testing the microphone required multiple steps. The first step was to test the microphone by itself to check if it worked properly. Then, the microphone had to be tested for noise. If noise was present in disruptive amounts, then as stated previously, a low-pass filter was required. The last step was to test the microphone with a low-pass filter. Testing equipment and tools to be used for the microphone are shown below in Table 23.

Component	Equipment & Tools
• **Electret Microphone**	• Function Generator • Oscilloscope • Breadboard • Resistors • Capacitors

Table 23: Microphone Testing Equipment & Tools

To check if the microphone works fine, it is given that the Electret Microphone has a power supply input of 5V max, with a maximum load resistance of 2.2kΩ. First, a function generator could've been used to supply $10V_{pp}$, connected to a 2.2kΩ resistor, then connected to the positive terminal of the microphone, while the negative terminal is connected to ground. The circuitry is illustrated below in Figure 72. The oscilloscope then was connected between the positive and negative terminals of the microphone, and it was expected for the group to see waves on the oscilloscope when talking, otherwise the microphone was not working.

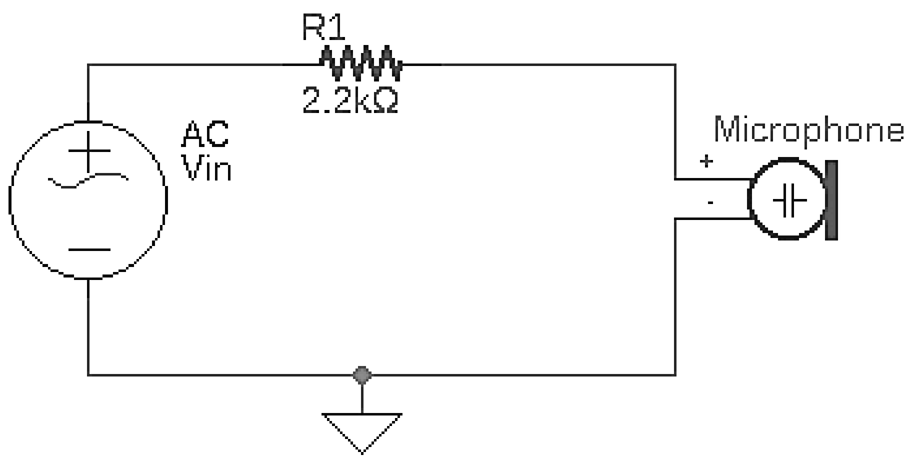


Figure 72: Unfiltered Microphone Testing Circuit

After that, if a lot of noise was present in the signals, a low-pass filter should've been added to the circuit. There are multiple options for having a low-pass filter, including a first order RC circuit, fourth order operational amplifier circuit, and many more. For testing, a simple first order RC circuit could've been built to filter the noise out. Instead of adding another resistor in series to the previous resistor, both could've been combined into one resistor "R0", and then a capacitor added to the circuit. The circuitry is shown below in Figure 73. The values for R0 and C0 were found using the equation previously derived in section 3.3.3.3 of this document $f_c = \frac{1}{2\pi RC} = 2KHz$, where R=R0 and C=C0. After that, it was expected to view clearer waves on the oscilloscope.

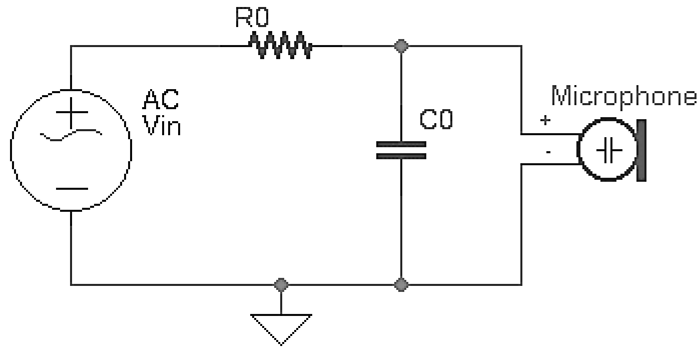


Figure 73: First-Order Filtered Microphone Testing Circuit

8.1.4 Audio Amplifier

Testing the mono audio amplifier breakout required the use of the speakers once again along with their specifications. The testing equipment and tools used for the audio amplifier are shown below in Table 24. First, the mono audio amplifier breakout was powered by a 2.5V to 5.5V DC input, and when that voltage was supplied, a red LED should turn on to indicate that the amplifier was on. The connections were done by connecting the "+" of the PWR pin to the input DC source, and the "–" of the PWR pin to ground. Next, once the mono audio amplifier LED lights on to confirm that it worked, the whole board was tested using the speaker, and an input AC signal. The gain should've been left at 2 V/V without changing the gain resistors at first, then the gain can be changed to test how high up the amplifier can go. The input AC signal can be generated using a function generator to generate a random sine wave with a certain frequency and amplitude. The function generator can be set to generate a sine wave with an amplitude of $2V_{pp}$ and a frequency of 500Hz. To confirm that the amplifier works fine, with a gain of 2 V/V, the results should've been the same as connecting the speaker directly to a $4V_{pp}$, 500Hz signal.

Component	Equipment & Tools
Mono Audio Amplifier	• Function Generator • 3-Output DC Supply • Speaker • Breadboard

Table 24: Mono Audio Amplifer Testing Equipment & Tools

For the connections, the "+" lead of the speaker was connected to the "+" of the OUT pin and the "–" lead of the speaker was connected to the "–" OUT pin. Finally, the function generator red (positive) lead was connected to the "+" of the IN pin and the function generator black (ground) lead was connected to the "–" of the IN pin. The resulting circuit looked like the following as shown in Figure 74 below.

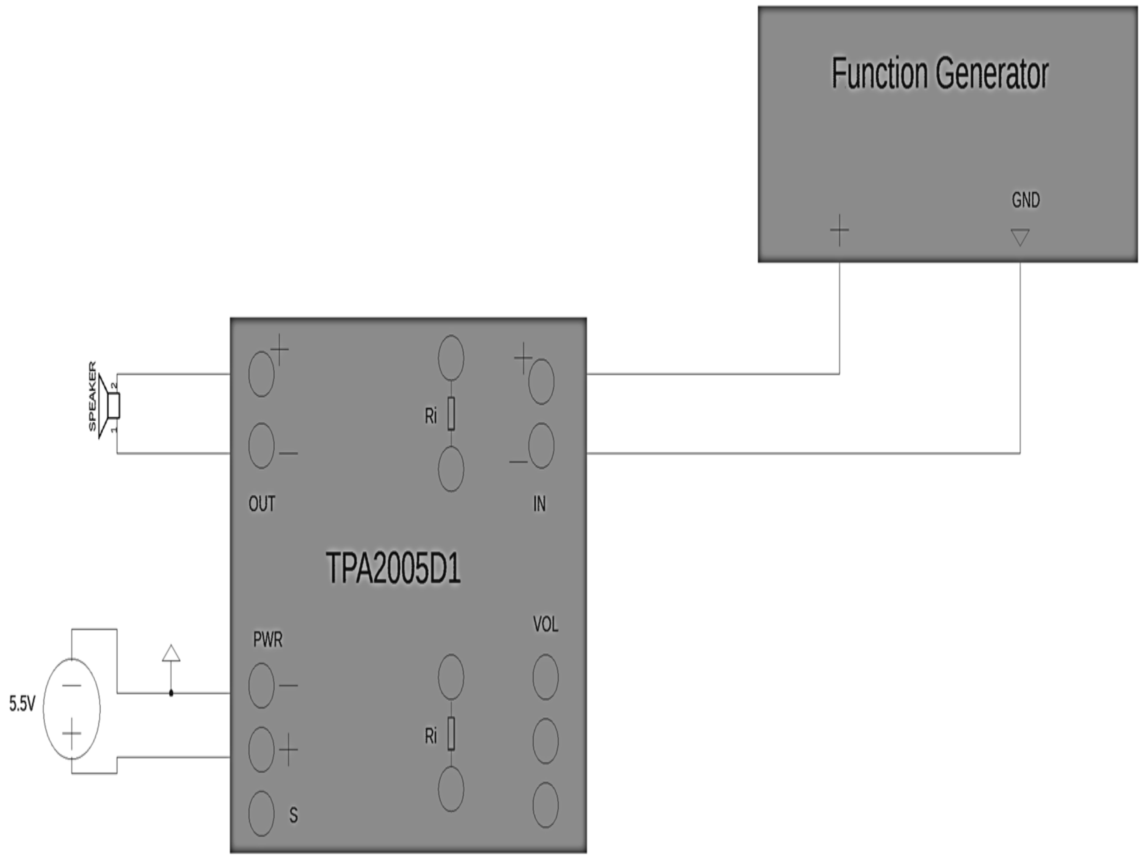


Figure 74: Mono Audio Amplifier Initial Test

After that, the different values for gain were tested by changing the gain resistors. But, if the gain became too high, the speaker might've blown up since it can take a maximum of 2V. To fix that, a resistor was put in series with the speaker in between the "+" leads in order to have a voltage drop across it and achieve 2V at the speaker again. The results were that the further the gain increases, the input signal amplitude should be reduced while the results should stay the same.

Finally, the shutdown pin was tested by simply connecting it to ground to see whether the amplifier shut off or not, and after that change the input DC source to 1V and check again whether it stays off or not. Once all the tests were finished, the group's own designed mono audio amplifier was expected to meet all the test expectations and results of the TPA2005D1 audio amplifier.

8.1.5 WiFi Module

Testing the WiFi Module was a little more complicated than the other hardware components. Since the group was using the ESP8266 WiFi Module for prototyping, it was used to observe the multiple inputs and outputs for the WiFi chip and how it worked. Once the ESP8266 WiFi module was finished for testing, it was mimicked and then it was expected that when the group used the ESP8266 SMT Module the same results should be observed. Testing equipment to be used for the ESP8266 WiFi Module and the ESP8266 SMT Module are shown below in Table 25.

Component	Equipment & Tools
ESP8266 WiFi Module	3-Output DC Supply Microcontroller or PC Wireless Network Soldering Tools
ESP8266 SMT Module – ESP-12S	3-Output DC Supply Microcontroller Wireless Network Breadboard

Table 25: WiFi Modules Testing Equipment & Tools

To test the ESP8266 SMT Module, it was required to use the module schematic provided earlier in this report as a reference, since it illustrates all the inputs and outputs required for it. Once all the connections were made correctly and the module was powered on, the module could then be connected to the group's microcontroller for further testing. Once connected to the microcontroller, the module could then be programmed through a personal computer to work in the same manner as the ESP8266 WiFi module. After that, the SMT module could be tested by checking if it could provide internet connection for the group's microcontroller. Expectations were that the group's microcontroller could connect or disconnect wirelessly to the internet, perform multiple internet functions like accessing a website, and then download a very small file around 500kB or less.

8.1.6 LCD

Testing the LCD had multiple functionalities that must be accounted for - this includes power, using the touchscreen interface, testing the GUI on the LCD, and displaying GIFs and videos on the LCD. To test power, it had to be connected to a similar power supply and test that it turned on and displayed properly. Ion has a functionality that includes a touchscreen interface to allow the user to interact with the GUI, so that had to be tested. To test the touchscreen, the necessary connections had to be made to power and allow these functionalities. To test the GUI on the LCD, the GUI had to be written and loaded onto the device and the LCD connected. The GUI resolution had to match that of the LCD so that it displays appropriately. Also related to the functionality of the GUI is the touchscreen, so that had to be tested. The last major functionality of the LCD was displaying GIFs and videos that could be played with the different applications that could be integrated into Ion. After integrating the different applications with the GUI and Ion, videos and GIFs played at the correct size and scaling as well as with minimal lag.

8.1.7 Camera

Testing the camera ensured that it was in working condition, was connected properly and displayed an image of reasonable quality.

8.1.7.1 Needed Equipment and tools

To test the camera, a prototype of Ion was needed with the camera connected as intended in the final design. Additionally, a specialized testing script was needed that will try to capture a frame from the camera and display it on the screen. The script was written in python specifically for this purpose.

8.1.7.2 Procedure

To test the functionality and image quality of the camera, follow the procedure in Table 26 below.

Step	Desired Result
Boot System	System boots without any error messages
Run camera testing script	An image is displayed that shows the camera's point of view
Visually inspect image	The image appears focused, with natural colors and lighting.
Check for error messages in the console	No error messages are displayed.

Table 26: Camera testing procedure

8.1.8 Servo Motors

Testing the servos verified that they were in working condition and connected properly to the designated outputs.

8.1.8.1 Needed Equipment and tools

To test the servos, a prototype of Ion was needed with the servos connected as intended in the final design. Additionally, a specialized testing script was needed that will rotate the specified servo to a desired orientation.

8.1.8.2 Functionality and connection

To test the functionality and calibration of the servos, follow the procedure in Table 27.

Step	Desired Outcome
Boot up system	System starts with no error messages
Run test script	No errors reported, script asks which servo to turn
Enter “1”	Script asks which orientation (0-180)
Enter “0”	Verify servo 1 turns to 0 degree position
Enter “90”	Verify servo 1 turns to 90 degree position
Enter “180”	Verify servo 1 turns to 180 degree position
Enter “X”	Script asks which servo to turn
Enter “2”	Script asks which orientation
Enter “0”	Verify servo 2 turns to 0 degree position
Enter “90”	Verify servo 2 turns to 90 degree position
Enter “180”	Verify servo 2 turns to 180 degree position

Table 27: Servo test procedure

The test was deemed successful when each step had completed with the desired outcome.

8.1.9 PCB Testing

Testing the PCB was very important since there were a lot of rumors about PCBs not working once they are received from their manufacturers. The equipment and tools required to test the PCB include all the previous hardware equipment and tools along with the hardware components. There were multiple steps for the group’s test plan to test the PCB, first, the multiple PCBs ordered should all be tested and checked for any defects. After that, since the group decided to assemble their own PCB, some of the components of the subsystems could be connected to the PCB to be tested each separately; first, to make sure the PCB is still working fine and second, to make sure the PCB is performing its expected task. After that, once it passed the test of powering one subsystem, multiple subsystems could be connected to it and tested each along the way. If any subsystem ends up failing, it could be tested on its own for troubleshooting, and then if it appears that the failure is from the PCB itself, then a new PCB is required. If not, then the problem should be resolved, and the rest of the subsystems should be connected to the PCB. Finally, once all tests pass, all the components must be soldered onto the PCB and then it should be tested again to make sure that neither the PCB nor the components were damaged due to soldering, otherwise all the PCB testing process should be restarted from the beginning. Once the PCB passed all the tests, it was good to go and placed in the group’s final design project. The process is better illustrated in the flowchart below in Figure 75.

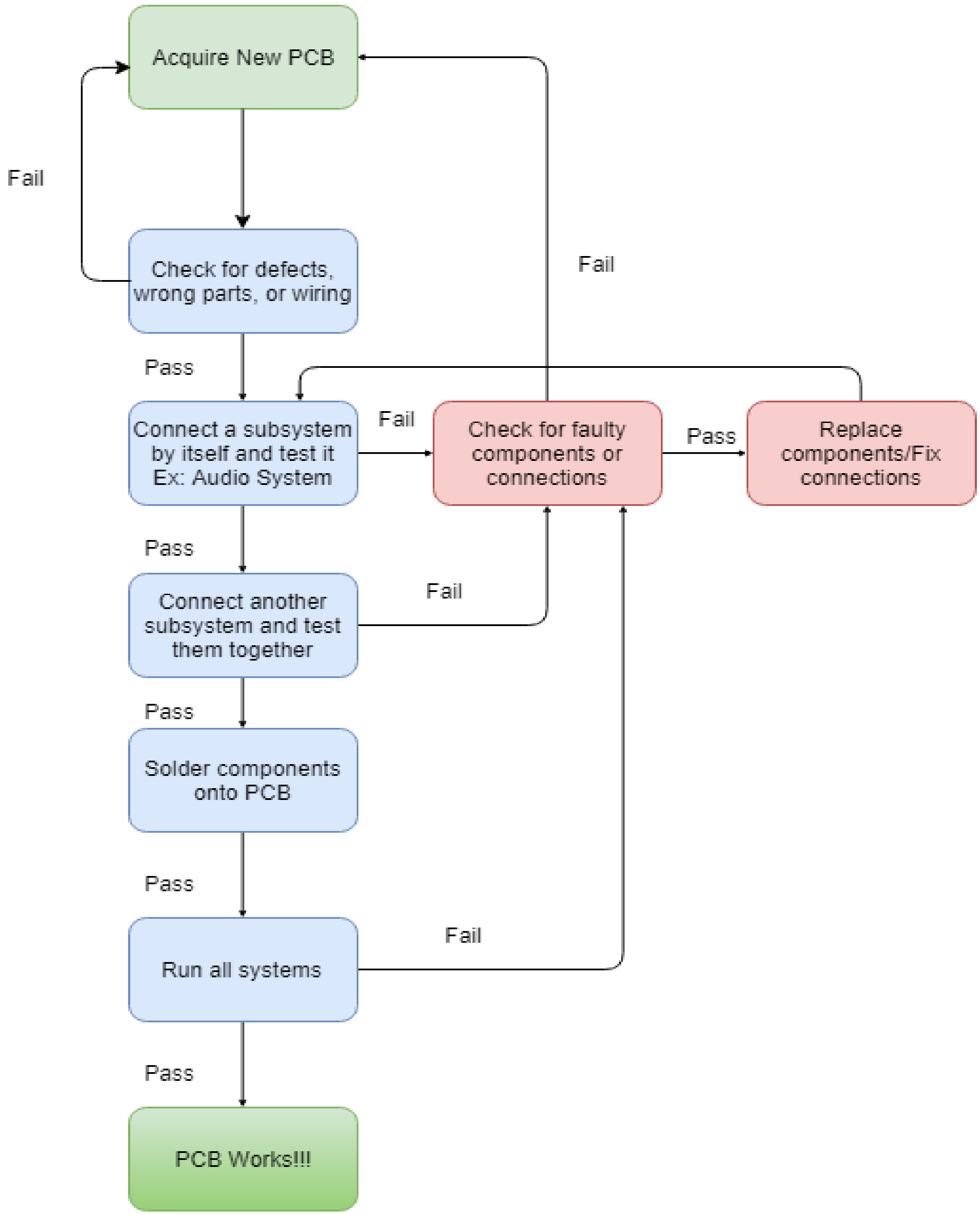


Figure 75: PCB Testing Procedure

8.2 Software Testing

8.2.1 Controller to MCU Interface

Testing the controller to MCU interface ensured that the I2C I/O points had been connected properly and correct protocol was used to establish a connection. The MCU is passive, so only commands from the CPU to the MCU were tested.

8.2.1.1 Needed Equipment and tools

To test the Controller to MCU interface (connection via I2C), a prototype of Ion was needed with the MCU connected as intended in the final design. An LED needed to be connected to pin 4 of the MCU and grounded. Additionally, a specialized testing script was needed that tried to establish a connection to the MCU and cause the LED to blink.

8.2.1.2 Procedure

To test the functionality of the Controller to MCU interface, follow the procedure outlined in Table 28.

Step	Desired Outcome
Boot up system	System starts with no error messages
Run test script	No errors reported.
Look at LED	Verify LED blinks at the frequency of 1 Hz.

Table 28: Controller to MCU interface testing procedure

The test is completed successfully, if all steps result in the desired outcome.

8.2.2 GUI

Testing the GUI relied on many of the other different technologies being already incorporated into it. A true prototype test of the GUI required it to be fully loaded onto Ion, meaning on the Raspberry Pi Compute module and all other hardware including the microphone, speakers, and LCD correctly connected. The main strategy for testing the GUI was to test every requirement and feature from start-up to shut-down. At start-up, Ion should automatically boot into the login screen if a face or profile cannot be recognized. At the login screen, all the associated features and options had to be tested including login, account creation, account recovery, and face detection. On boot-up if a face was recognized, then it would log into the profile and load up the welcome screen and rest state to wait for a command. From there, menu navigation had to be tested by tapping the screen to bring up the menu and tapping the different buttons to navigate to the respective pages. When the settings and preferences are changed, they should be tested to see if they change the associated values in the system and fully complete the task which they advertise. For example, changing Ion's customizable profile to the female version should change the voice and personality of Ion. Also, for all these settings and preferences, the voice commands that can also control these changes had to be tested to see if they appropriately updated the GUI. One last big part of testing the GUI is making sure that Ion had a fluid, realistic emotion feedback appropriately represented through GIFs that display on the screen. That could be test by seeing if Ion displayed a welcome GIF on boot-up or by otherwise conversing with Ion.

8.2.3 Voice Recognition

Testing voice recognition required a microphone and the voice recognition software complete. To test it, keywords signifying commands had to be spoken into the microphone and it had to be verified that the Google Speech API was correctly translating and interpreting the voice commands to be utilized. The main keywords that signified commands and had to recognize them with high confidence and correctly interpret them 80% of the time, per the software efficiency requirements. The commands were then processed, and Ion would respond or complete the task within two (2) seconds per the software efficiency requirements.

8.2.3.1 Procedure

To test the functionality and accuracy of the voice recognition, follow the procedure in Table 29 below.

Step	Desired Result
Boot System	System boots without any error messages
Say "Open Youtube"	System confirms that it correctly understood the command within 2 seconds
Repeat above step for a total of ten times	System responds with the desired result at least 8 out of 10 times.
Say "Play study playlist"	System confirms that it correctly understood the command within 2 seconds
Repeat above step for a total of ten times	System responds with the desired result at least 8 out of 10 times.
Say "This Banana plays nothing" (does not make sense on purpose to test against false positives)	Systems asks for clarification
Repeat above step for a total of ten times	System responds with the desired result at least 8 out of 10 times.

Table 29: Voice Recognition Test Procedure

8.2.4 Facial Detection

8.2.4.1 Needed Equipment and tools

To test the facial detection software, a prototype of Ion was needed with the camera connected as intended in the final design. Additionally, a specialized testing script was needed that would run in the console environment to provide feedback in text form. The script was written in python specifically for this purpose.

8.2.4.2 Procedure

To test the functionality and accuracy of the facial detection, follow the procedure in Table 30 below.

Step	Desired Result
Boot System	System boots without any error messages
Run facial detection test script	A live video feed from the camera is shown and the console does not print any error messages.
Turn the camera so that it does not capture any faces. Enter “T”.	The live video feed is unchanged and the console prints “0”.
Turn the camera so that it captures only one face. Enter “T”.	The face in the live video feed is surrounded by a green rectangle and the console prints “1”
Turn the camera so that it captures two faces. Enter “T”.	Only one face in the live video feed is surrounded by a green rectangle and the console prints “2”
Turn the camera so that it captures three faces. Enter “T”.	Only one face in the live video feed is surrounded by a green rectangle and the console prints “3”
Enter “S” and sweep the camera across the room, preferably with several objects of different size, shape and color in it. Monitor the screen while doing so.	During the sweep, only faces are bound by a green rectangle, not objects.
Check for error messages in the console	No error messages are displayed.

Table 30: Facial detection testing procedure

The test was completed successfully when all steps have been completed and produced the desired results.

8.2.5 Facial Recognition

8.2.5.1 Needed Equipment and tools

To test the facial recognition software, a prototype of Ion was needed with the camera connected as intended in the final design. Two users that have calibrated their profile had to be present. Additionally, a specialized testing script was needed that will run in the console environment to provide feedback in text form. The script was written in python specifically for this purpose.

8.2.5.2 Procedure

To test the functionality and accuracy of the facial detection, follow the procedure in Table 31 below. The test is completed successfully when all steps have been completed and produced the desired results.

Step	Desired Result
Boot System	System boots without any error messages
Run facial recognition test script	A live video feed from the camera is shown and the console does not print any error messages.
Turn the camera so that it does not capture any faces. Enter "T".	The live video feed is unchanged and the console prints "No Face Detected".
Turn the camera so that it captures only the face of a known user. Enter "T".	The user's face in the live video feed is surrounded by a green rectangle and the console prints the user's name.
Turn the camera so that it captures two faces. The face of the previously recognized user and an unknown person Enter "T".	Only the previously recognized user's face in the live video feed is surrounded by a green rectangle and the console prints the user's name.
Turn the camera so that it captures two faces. The face of the previously recognized user and another known user Enter "T".	Only the previously recognized user's face in the live video feed is surrounded by a green rectangle and the console prints the user's name.
Check for error messages in the console	No error messages are displayed.

Table 31: Facial recognition testing procedure

9.0 Device Operations

Operating the device begins with powering it on by plugging it in. From there, the device automatically boots up and the software, including the GUI, voice recognition, and face detection and recognition, will all begin running on bootup. The GUI will launch into the front page. The user can navigate to the different pages to change settings or see commands. With the face detection software running in the background, every so often, when a face is detected, the face recognition software also running in the background will analyze the detected face and display a popup accordingly, as discussed in the sections above. When the voice recognition software starts on bootup, an audible melody will be played along with a welcoming message from Google. From there, the user can give commands to the device by saying ‘Hey Google’ followed by the command. The available commands can be found on the Commands page. Once the user is done using the device, it can be shut off by simply being unplugged.

10.0 Administrative Content

This section serves to provide some details on some administrative content related to the project including a schedule of milestones and a layout of spending and finances.

10.1 Project Milestones

The project milestones and tentative schedule are shown in Table 32 below. This table completely outlines all the major milestones through the Spring 2018 and Summer 2018 semesters - from initial idea brainstorms to the final presentation of our project.

Task Number	Task Name	Start Date	Due Date
Senior Design 1			
1	Idea Brainstorms	1/16/18	1/23/18
2	Project Selection & Role Assignments	1/23/18	1/25/18
3	Initial Document - Divide & Conquer	1/23/18	1/28/18
4	Bootcamp	1/28/18	1/28/18
5	Research	1/30/18	1/30/18
7	Divide & Conquer V1	1/23/18	1/28/18
8	Divide & Conquer V2	2/25/18	3/11/18
9	Table of Contents	3/13/18	3/20/18
10	First Half	3/21/18	4/9/18
11	First Half Review w/ Dr. Richie		4/12/18
12	First Draft	3/21/18	4/20/18
13	Report Due		4/27/18
Senior Design 2			
14	Assemble Prototype	5/01/18	5/25/18
15	Testing and Redesign	5/26/18	6/14/18
16	Finalize Prototype	6/14/18	7/16/18
17	Final Presentation		7/23/18
18	Peer Review Report		7/30/18
19	Final Documentation		7/30/18

Table 32: Project Milestones

10.2 Project Budget and Finances

As this project was self-funded by a 3-person group, the group was on a budget. The aim was to limit the individual budget to about $200, adding up to $600 in total. All the components that were purchased are listed in Table 33 below, along with the projected cost at the beginning of the project and the actual cost on the date of purchase. As the project wasn't funded by an outside company and there were no plans to sell the product on the market, budget wasn't a huge deal outside of the personal budgets and what each individual group member was willing to spend.

Through research the group figured that the average cost of a similar product, like the Amazon Echo or Google Home, would be around $100-150. Companies like Amazon and Google are absolute powerhouses in the market today, with access to deals with shipping and supply that the group don't have, so naturally their cost was a lot higher. The project was also only a prototype, so the cost was a lot higher than a final product whose design would be known and streamlined to be cheaper.

Component Costs and Budgeting		
Component	**Projected Cost ($)**	**Actual Cost ($)**
Microphones	3.00	7.78
Speakers	6.00	15.90
LCD Touchscreen	60.00	80.00
CPU	35.00	26.00
Camera	15.00	10.00
PCB(s)	50.00	60.00
3D-printed body	50.00	0.00
Buttons, switches	10.00	1.50
Resistors, capacitors, etc.	10.00	50.00
LEDs	10.00	5.90
MCU	20.00	7.52
Servo(s)	30.00	6.99
WiFi Modules	32.00	11.95
Battery	20.00	0.00
Voltage Regulators	15.00	12.58
Total	366.00	296.12

Table 33: Project Budget

Appendix A – List of Tables

Appendix B – List of Figures

Appendix C – References

[1] Mono Audio Amp Breakout: https://www.sparkfun.com/products/11044

[2] 0.5W 8-ohm Speaker: https://www.sparkfun.com/products/9151

[3] Electret Microphone: https://www.sparkfun.com/products/8635

[4] ESP8266 Thing Dev Board: https://www.sparkfun.com/products/13711

[5] ESP8266 WiFi Module: https://www.sparkfun.com/products/13678

[6] Round Rocker Switch w/ Blue LED: https://www.sparkfun.com/products/11155

[7] microSD reader: https://learn.adafruit.com/adafruit-micro-sd-breakout-board-card-tutorial/introduction

[8] DAC Breakout: https://www.adafruit.com/product/935

[9] ESP-12S WiFi Chip: https://www.adafruit.com/product/2491

[10] Voltage Regulators: http://www.ti.com/design-tools/webench-power-design/power-designer.html

[11] https://en.wikipedia.org/wiki/Eigenface

[12] https://github.com/bytefish/bytefish.de/blob/master/blog/fisherfaces.md

Appendix D – Permissions

1. SparkFun™ Electronics:

Images Permission Inbox x

Mohammad Rizeq
Dear Sparkfun, My group partners and I are currently researching for Senior D...

SparkFun Customer Service <cservice@sparkfun.com>
to me

Type your response ABOVE THIS LINE to reply

Mohammad Rizeq
Subject: Images Permission

APR 18, 2018 | 10:06AM MDT
Annabel B. replied:

SparkFun Electronics is an open source company, so you're welcome to use images from our website as long as you give us credit for them!

If you need anything else, let us know!

Have a great day!

Annabel Bonner
Customer Service
SparkFun Electronics
303.284.0979

2. Adafruit:

Image Usage Permissions Inbox x

Mohammad Rizeq
Dear Adafruit Support, My group and I are Undergraduate students at Universit...

Adafruit Industries
HI Mohammad, Thanks for the note. Please provide a link to the exact photos y...

Mohammad Rizeq
Hello Jessie, Here are the images: WiFi Module: https://cdn-shop.adafruit.com...

Adafruit Industries
to me

this is all OK. use either or both.

thanks,
adafruit support, phil

Phillip Torrone <pt@adafruit.com>
Mon 4/9, 8:46 PM
David Jaffie; legal <legal@adafruit.com>

Reply all

Inbox

thanks for the note, totally OK!

David Jaffie
Mon 4/9, 6:55 PM

Hi,

I am researching for a project for a Computer Engineering degree program and want to ask for your permission to reference your components and use their information and photos in a project report.

Thanks,

David

3. Webench® License: http://www.ti.com/design-tools/disclaimer.html
4. WikiMedia:

You are free:

- **to share** – to copy, distribute and transmit the work
- **to remix** – to adapt the work

Under the following conditions:

- **attribution** – You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **share alike** – If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

https://commons.wikimedia.org/wiki/File:Amazon_Echo.jpg
https://commons.wikimedia.org/wiki/File:Google_Home_sitting_on_table.jpg
https://commons.wikimedia.org/wiki/File:An_Apple_HomePod_speaker_.png

5. Peeqo:

Abhishek Singh <abhishek@shek.it>
Fri 4/6, 5:42 PM
David Jaffie

Reply all

Inbox

Hey David, thanks for reaching out and I'm happy the project was able to help you in some way. Feel free to reference it in your paper. I took a cursory look at your paper and it looks great but will dig in deeper as soon as I have some time on my hands. Do you have a working prototype built?

David Jaffie
Thu 4/5, 5:27 PM

DeskBot_DivideAndCon...
416 KB

Download Save to OneDrive - Knights - University of Central Florida

Hi Abhishek,

I am a senior Computer Engineering student at the University of Central Florida located in Orlando, FL. For our degree program, we take a final course called Senior Design where we design a project that incorporates electric and software components from scratch to working prototype. Through research into personal assistant and robots, we came across Peeqo and were blown away. We love the idea of making a desktop, interactive little assistant who can show emotion and communicate with the user while also being a helpful assistant.

We want to show our appreciation for your project and ask for permission to reference it in our research, paper, and presentation. This includes both the research and information you have presented as well as images and videos of Peeqo.

I have some documentation of our project if you're interested and we'd love some feedback if you have time.

Thanks,

David Jaffie
davidjaffie@knights.ucf.edu