Senior Design 2 Final Project Document

Robo-dog: Smart Robot

Department of Electrical Engineering and Computer Science
Group 29

Esteban Arguello:	Electrical Engineering
Anthony Coniglio:	Computer Engineering
Jonathan Hernandez:	Computer Engineering
Adrian Lopez:	Electrical Engineering

Table of Contents

List of Figures

List of Tables

1. Executive Summary

In recent years, home automation, smart devices, and assistive technology have greatly increased in popularity. Once considered inaccessible or luxurious, smart assistive devices such as Amazon's Alexa, Apple's HomePod, or Google's Assistant are now common household devices. Now the inaccessible devices are mobile and multi-input solutions which can interact with the users in more ways than one. Not by just a simply reply to them to activate but by responding in full sentences back to the user and even started making its own language between devices that human couldn't understand. When it comes to devices that have music playback capabilities, controlling smart home networks connection, or just adding items to your grocery list through command. All these features lack the physical interactivity and personality aspect and that is the market we are trying to sell to. This shortcoming alone leaves a space for a new type of products which has fell off the market that needs to come back. Our project seeks to create a prototype of a smart robot dog with multiple levels of interactivity, responding to audio, visual, and kinesthetic cues. This has greater aim of this level of interaction and responsivity is to fill another role for which a need has arisen, that is the growing awareness of the problems caused by loneliness and cognitive stagnation.

Our robot is a synthesis of various technologies culminated into a compact and convenient package resembling a facsimile of a dog. For this prototype, the chassis and structure of the robot consist of a 3-D printed plastic body and servo motors at each joint, with a microcontroller at the center of a custom designed PCB serving as the motor controller, directing the movement of the robot body. It can walk forwards and backward. It can also sit and give its paw when commanded. A small computer, such as the Raspberry Pi, is used to program the more complex features of the robot (e.g., speech and image processing). Cameras, microphones, capacitive touch sensors, and speakers are integrated into the robot for realization of the desired behavior. A display can be included to convey certain expressions and information in reaction to input form the user. It utilizes rechargeable lithium-ion batteries and a custom designed power delivery system to provide adequate use time and convenience, avoiding excessive down time.

The robot runs a suite of various programs to perform its functions. Within that suite is included image recognition software, which (at a minimum for this prototype) should be able to recognize different solid colors on sheets of paper and trigger a corresponding action for each. This serves as a proof of concept for more complicated image and speech recognition and response in possible development stages beyond prototyping and with more powerful hardware. Similarly, another program is capable of processing simple speech, such as normal one- or two- word commands ("sit", "lay down", "paw", etc.". One desired outcome of this project is to limit cost and improve access to such interactive and smart robot devices, as current existing products are still relatively prohibitively expensive. The vision of a finalized version of this product is beyond the scope of this project, but the concepts that could be further explored and expanded can be easily integrated onto the prototype of the project. Our project seeks to provide amusement, enjoyment, and enrich the lives of its users. We want the people to see our robot dog as a member of their family. This is what makes our product different from the rest. We are trying to seek that human connection so people can feel comfortable knowing that a robot dog is walking around house.

2. Project Description

This project involves designing a robot that resembles and moves like a dog, while also having the capabilities of a virtual assistant like Amazon's Alexa. The project is made to create a unique companion for all ages while also providing utility. The robot dog is small and compact, designed do not be heavy or difficult to carry around. This section on the project description gives the reasoning, motivation, and goals of developing a robot dog with the idea of being a home companion for house chores, assistance, and general emotional wellbeing. In the final part of this section the projects requirement specifications are laid out in order to meet the goals and objectives.

For a successful project, it is important to highlight the motivations, goals, objectives, and requirements that are necessary. Having a clear idea of these before the project is started allows the project to perfectly fit its description, without difficult complications. Constantly being reminded of the motivation of the project allows creators to develop new ideas and improve on old concepts while still maintaining the vision of the initial project. Motivations are the base of the project. They are more important than the idea of the project itself. Ideas are only solutions to the problems presented by the motivations. So, for the project to be successful, it must follow the motivation. This means that the original idea of a project can change so long as the motivations and goals are accomplished.

Goals are also an important part to determine in a project. Goals represent the long-term guidelines of the project. Objectives are made to accomplish these goals. Without a clear set of goals, the designers have no real direction to pursue. Goals allow for clear objectives to be established and requirements to be set. Objectives are the means in which the designers and engineers reach the goals set by the motivations of the project. Objectives are broad requirements that help satisfy a specific goal. They answer all questions on how the goals were achieved. Objectives are generally measurable and specific steps in a design. Having these defined early on in a project make the progress of the project easy to track while also creating a clear path for the engineers to follow. Finally, requirement are the specifications of the project that must be accomplished for the project to work successfully. These requirements are developed by understanding the multiple objectives created to accomplish the goals and motivations of the project. Requirements give a minimum of what needs to be achieved for the project to accomplish its goal. When creating a project, requirements much be set in advance so the people developing the project have a list of clear objectives that must be accomplished. Failing to meet a requirement should mean failing to reach an objective, which the entails a failure of reaching the goal and motivation of the project.

2.1 Motivation and Goals

This project seeks to provide users with an alternative to live animal emotional support companionship. There is an increasing number of people, old and young, suffering from loneliness and anxiety at home, with not many alternatives to traditional cats or dogs. Emotional support animals require time and money to take care of. This can become overwhelming or unattainable for many people. Furthermore, some people in need of these animals have allergies or a family member with allergies that do not allow obtaining an emotional support animal. Therefore, an interactive robot that mimics the basic behavior of an emotional support animal, tied with the functionality of a virtual assistant such as Amazon's Alexa™, could provide a sense of companionship without the need for continuous expenses. Moreover, the integration of the virtual assistant allows this companion to only provide emotional support, but utility in a smart household as well.

When brainstorming multiple ideas for potential projects, we had one clear aspect that the team desired: utility. The team wanted to create a project that would be helpful to its user. Nowadays, the majority of electronic devices are connected to each other via the Internet of Things (IoT). This interconnection permits one potential device to control several devices. With this idea in mind, the team decided on creating a device that could control the IoT within a household. After some brainstorming, it became clear that the best and easiest solution is a virtual assistant. Virtual assistants are slowly taking over as the main control hubs for multiple IoT devices. If we could create a project with virtual assistant functionality, then our device would be capable of controlling multiple devices with ease. Moreover, companies like Amazon and Google have made integrating their own virtual assistant onto simple projects, like our own, possible. This allows our project to not only control these multiple devices but is also able to have all the advanced features created by these advanced technology companies. However, this would mean that we are not creating our own virtual assistant, rather integrating it to our project. This is where the next motivation for our project ties everything together.

As mentioned previously, another motivation for creating this project is to help people in need of companionship. The recent pandemic, mental illnesses, and the current society have created a lot of issues when it comes to people receiving their desired companionship. The team viewed this project as an opportunity to use our skills to create a robot that could provide the companionship desired by many. From experience, there are few better companions than a pet. Pets are extremely loyal and loving towards their owners. The team decided that they would then create a robot that would act like a pet. A robot that would be loyal to the user while also never showing negative feelings. From several research papers and studies, it is evident that infusing emotion onto a robot is very complex and beyond the knowledge of the team. As is explained later, the robot moves and shows basic emotions to act like a dog. The team hopes that our project can be used as a prototype for a much more complicated robot, capable of perfectly imitating emotion. It is when we combine both motivations that we obtain the overall idea of our project; a robot dog with the capability of connecting and controlling IoT devices using a virtual assistant.

The overall goal of the project is to have a portable, easy to use autonomous robot that can be interacted with to perform both the functions of a virtual assistant, and the functions of a dog. The project should be light weight (easy for the average adult to pick up and carry), not prohibitively expensive, interactive, reliable, and with a reasonable use time. The expected size of the robot is around the size of a small dog, like a Chihuahua. The size of the robot allowed us to be easily transport it while also making it cheaper to make. The vast majority of the robot costs was in creating or purchasing the overall product. Maintenance is cheap since it has a rechargeable and should last a decent amount of time. From this description, it is clear that one of the goals is to make the robot easy to use. The commands should not be so complicated that the user must remember specific random sentences. Rather, the actions of the robot should be tied with the definition of the phrases commanded by the user.

There are two main goals for this project. First, the robot must appear and act like a dog. The appearance should be obvious enough that it is not necessary to be explained. There are several objectives created to help accomplish this goal. These are explained in further detail in the next section. For the robot to accomplish this goal, it must be capable of reacting and acting like a dog. To do this, the robot must possess as many senses as possible that a real dog possesses. This are sight, hearing, touch, smell, and taste. The more obvious and important of these senses are sight, hearing, touch. This was a focus of the project. The design of the robot must also look like a dog. The robot should have four articulated legs, for example, which can move in a similar manner as that of a dog's legs. The robot should also have a head and ears shaped like a dog to help with the appearance. This goal is important since we want the robot to not feel like a robot. We are not capable of giving the robot emotion, but the robot should at least appear like a dog.

The second main goal of this project is to have the robot be more useful that a regular dog. There needs to be a distinction from obtaining a dog and obtaining a robot dog, since the robot dog is never going to be able to compete with the genuine feelings of a living being. There are a few objectives created to help achieve this goal. These goals helped the robot dog to be useful in more than one way.

2.2 Objectives

The objectives of the project are to integrate a virtual assistant into the robot and have the robot perform certain actions that make it mimic an emotional support dog. These include following the user, standing still on command, and "sitting" (by folding the legs). Moreover, because the dog is electronic, we can make it even more useful than a regular emotional support dog. The robot dog should be able to display information or data onto a screen located where the eyes would normally be. Furthermore, the dog is also linked to a smart assistant and can use all functions that are included with this capability. These include searching queries on the internet, controlling smart devices at home, and setting reminders. The other objective of the project is to make the robot act like a dog. Some planned functions include wiggling its tail, following the user, and sitting on command. The function of the project is to have a useful and understandable robot which gives the user a positive experience. As a reminder, the robot dog should give a general sense of companionship. For reference of the basic physical frame of the robot, see common children's toys such as:" Remote Control Robot Dog Toy" by Top Race. Unlike the plan of our own robot design, this robot uses wheels for its legs for easier mobility. The overall shape of the robot is also a good reference on how to make the shape of our robot dog. This reference is only an insight on how we plan to design our robot physically. Is a good reference to what is expected of the first goal of the project. Furthermore, the toy helped us understand what objectives are necessary for our own project.

The objectives of this project are created to satisfy the two main goals mentioned previously. The table below (Table 1) shows a list of some of the objectives used to accomplish these goals. Objectives are specific but not as specific as the requirements. These are what are used in the specify the necessary requirements for this robot. Because of this, the objectives have no requirements or specifications detailing how these tasks can be accomplished.

Table 1: List of some objectives

Objective	Description
Dog mobility	• The robot moves like a dog normally does. The robot has four legs with joints. The robot should be able to adopt dog-like poses.
Dog senses	• The robot should sense its surrounding like a normal dog would. These include sight, hearing, touch, smell, and taste.
Dog appearance	• The robot appears like a dog. For example, the robot should ideally have four legs, a head, and a tail. The ears should look like dog ears.
Obtain internet search queries	• The robot should be capable of looking for and showing search results on the internet.
Understand voice commands	• The robot is capable of listening and reacting to commands

The objectives shown in the table are only some examples of objectives used in this project. As can be seen, the first objectives are useful in accomplish the first goal while the last objectives presented are helpful for accomplishing the second goal. As a reminder, the first goal is to make the robot resemble a pet (specifically a dog). To do this, objectives are created. Completing these objectives ensures that the robot moves like a dog and react like a dog. These objectives help break down the goal into smaller pieces that can be tackled individually. The fourth objective in table 1 states that the robot must be capable of making internet search queries. This ties directly into the second goal of the project; to make a robot that is helpful and capable of connecting with IoT devices. Some objectives tie into both goals, as can be seen from the last objective. This object asks the robot to understand voice commands. This allows the dog to react to its name and act like a dog, while also providing a means for the dog to act as a communication device between humans and devices. These and many other objectives are used to make sure both goals of the robot are completed. Unlike requirements, objectives do not need to be specifically accomplished. Objectives help accomplish the overall goal, but they are not what is necessary to do. Requirements are what are necessary for the goal to be accomplished. Likewise, requirements are what help complete objectives. Objectives are not specific and not useful when designing specifics for the robot. For these specifics, requirements and specifications are necessary.

2.3 Requirements and Specifications

To successfully accomplish the goals and objectives established above, requirements and specifications are needed. These are the specific details of the project that define how the robot is designed. Unlike objectives, the description of these requirements must be very specific. If a range is used, it must be very small to allow for precision and reliability. For this project to work successfully, several requirements need to be accomplished. All these requirements are specified in Table 2. The table also shows the exact requirements desired for Robodog. Table 2 details the specific objectives that are accomplished by the requirements. Therefore, the table has some detail but not the full detail needed for requirements. This detail is given in the requirements table shown in Table 3. Continuing, the different objectives of Table 2 are described.

First, the movement capability. The robot needs to be capable of moving its joint fluidly while also having enough rigidness to support the weight and structure of the robot. The battery needs to be able to last a useful amount of time. This means that in standby it should be able to last at least a few hours and at least an hour active. Engineering requirements such as these are outlined in Table 2, along with several other measurable requirements for this project. This makes the robot useful throughout the day. Another aspect is audio output, or speakers. The robot should be capable of providing “doglike sounds” and other sounds to show that it is active or that it has understood a command. Another requirement is for the robot to know when it is being touched. This makes the robot feel more like a pet and act like one too. When it is touched, depending where, the robot dog reacts in a specific way which is programmed to. The touch sensor is sensitive enough to detect the touch on top of the robot dog, but not sensitive enough that it can cause constant false triggers.

Another important requirement for this project is the virtual assistant integration. This requirement forces the robot to be capable of internet access. Furthermore, the robot uses the virtual assistant for its mandated purpose. New commands can be coded into the virtual assistant to allow the robot to do more actions and task. As a reminder, the idea of this robot is to not only act as a virtual assistant, but to also act as an animal. The robot should react to specific stimuli and show emotion towards them. The virtual assistant integration makes this robot more than just a robotic animal. The virtual assistant capability makes this robot dog useful on a day-to-day basis. Of course, for this to work, the robot is also required to have some sort of voice recognition. The robot needs to be capable of listening to commands in order to execute them. The voice recognition needs to be accurate enough so that it can understand the user most of the time, without annoying said user.

Two more important requirements for this robot are the reset and wake up functions. The reset function is simply a way to reset the robot in case there is a small malfunction of error in its behavior. The button should turn off the robot and the robot should start up again when turned on. This helps in case there is any issue that does not allow the robot to be shut down normally. The other important requirement is the wake-up function. The robot should enter standby mode if it is not used in a certain period. During this mode, the robot consumes significantly less power. The robot is able to leave this mode when it is called by voice recognition or by touch. This feature helps prolong the battery life of the robot and help the robot run more efficiently from the natural restart.

These requirements and objectives ensure that the robot accomplish its desired goals. The in-depth description provided above with the summarized description in Table 2 help the team understand how the specifications of the robot must be created. As mentioned before, the specifications and requirements are the most important part of the robot, as they are what the team as engineers use as a main goal to help achieve the overall goal.

Table 2: Description of Engineering Requirements

Requirement	Description
Movement capability	• Motors need to be able to move joints fluidly and capable of rotating 180 degrees. • Robot should be capable of performing at least 2 positions related to dogs. Some examples are sitting, laying down, lifting a leg, giving a leg on command, etc.
8-hour battery life	• Battery should last at least 8 hours on average. • Battery should last 1 hour when using max power consumption.
Virtual Assistant integration	• Connects with a virtual assistance API to perform default tasks. • Capable of performing new commands. • Connects to the internet via Wi-Fi. • Has Bluetooth capability.
Voice recognition	• Microphone capable of recognizing specific commands up to 2m away. • Should respond to its name within first try. • Delay no longer than 500 milliseconds.
Audio response	• Speakers should emit "doglike sounds" to announce presence and audibly show understanding of commands • Speakers should be at least 50dB. • Volume level adjustable.
Detection of when it is petted	• Multiple touch sensors with an operating supply voltage no larger than 4V.
Display capability	• OLED or LCD screen. • At least 240*240 resolution.
Wake up function	• Goes into standby if it is not used for a certain period of time. • During standby, the robot must use at most 5% of the maximum consumption. • Can be reactivated by touch or voice command.
Reset button	• Button that fully resets the robot. • Should work regardless of robot condition (in the middle of a command or unresponsive by other means)

Table 3 depicts the engineering requirements that is used to satisfy all the requirements mentioned above (Table 2). It is not only important to understand what the robot needs, but also how we are able to produce this robot. Table 3 shows what is to be expected from each piece of technology required in the design. These are the important requirements that the robot must satisfy to accomplish its overall goals. The team has also decided to specifically choose three of these requirements to be the most important. These are voice recognition, virtual assistant integration, and the movement capability. Satisfying these three major requirements allows the project to successfully complete the goals detailed previously. This was the focus of the team and are essential for the project to work properly and as desired.

Table 3: Engineering Requirements

Requirements	Specifications
Weight	< 15 lbs.
Response Time	< 3 sec.
Articulation*	180-degree rotation
Touch Response	< 500 ms delay
Audio Quality	at least 50dB at 3ft.
Audio Recognition*	80% of words understood
Display Size	10cm x 8cm
Standby Mode	5% of full load energy consumption
Dimensions	30cm x 45cm x 25cm
Battery life	3 hours idle, 1 1/2 hours active
Video Quality	at least 240*240 resolution 8-bit full color
Color Detection*	< 500 ms delay in response
Camera Quality	• > 1 megapixel quality • Works with regular indoor and outdoor lighting

When it comes to the specifications, the robot must accomplish all of these. Should any of these fail, the robot's quality, reliability, and overall design this result in the robot not be satisfactory for the team. These are the specifications that are be used for determining what parts we used and how the robot was designed. Not only does the quality of each sensor and device in the robot matter, but their weight and design matter as well.

Table 4 shows some of the project constraints. These constraints are limitations that are imposed in our robot to make sure it functions safely and properly. This needs to be taken into consideration when designing the robot. Other constraints include economic and time constraints, as well as environmental, social, political, and ethical constraints, which is elaborated upon further in this document. Table 5 shows the standards the robot must follow. This robot requires using multiple technologies that have set standards. These standards must be followed in order for these technologies to function properly and reliably. Straying from these standards, or not using pre-existing standards, could result in development difficulties within this project, as well as severely limit expansion or scaling as the complexity of the project increases. Many of these standards are communication based, whether wired as in the current design, or even wireless should the design head in that direction. Like the constraints, the standards of this project are further explained and detailed further in the document. Both the constraints in Table 4 and the standards in Table 5 limit the manner in which the robot can be designed. It is crucial to follow both of them and keep them in mind at every step of the project design and definition process, as they are what allows the robot to work properly and safely. The proper use of standards and observance of constraints allows the final project result to be open and usable by as many people as possible.

Table 4: Project Constraints

Constraints	Description
Audio sounds must be separate	Audio sounds must be played one at a time to avoid confusion and interference.
Device cannot overheat	The device is continuously and in a relatively closed casing.
Robot needs to be light weight	The robot needs to be comfortably lifted by an average adult.

These are only a few examples of project constraints and standards the project has contain. The descriptions are simple and are meant to only provide a general idea. As mentioned previously, these constraints and standards are explained in much greater detail further on. The purpose of placing these tables here is to help create the House of Quality following this section. It is important to have at least a small idea of the constraints and standards of the project before the design of the project is created. Even simple tables like Tables 4 and 5 are useful when designing the initial prototypes of the robot dog desired for this project. Otherwise, many factors of the original designed had to be modified or removed so we didn't fail to meet specific standards and constraints that are affect the overall design of the project. By avoiding this, the team saves time and money to create a much better project.

Table 5: Related Standards

Standards	Description
USB	For communicating between the robot and a computer when plugged in, and potentially (pending further research) for charging the battery.
I2C Rev. 6	Protocol for communicating between the different sensor modules and microcontrollers, this protocol is also used for servo motor control alongside PWM.
Bluetooth	Wireless communication standard for potentially connecting a wireless controller to interface with the robot dog.
Wi-Fi	Wireless communication standard to connect the on-board computer.
Python	An interpreted and high-level general-purpose programming language with a focus on functional syntax and expandability.
C	A general-purpose programming language with low-level access to memory and a focus on portability.

Table 5 shows many of the possible parts that are used for the project. A description for where these parts are used is also provided. This table helped us determine the overall cost of the robot as an initial estimation pending further research, as well as help to understand what was needed to design and build this robot. It is important that there are enough parts to fulfill every requirement of the robot (specified in Table 1) and provide some avenues to expand the capability of the robot and provide some flexibility in the design process. From Table 6, it should be understandable how the robot looks like electronically, and where every part fit is in the overall design. An overview like this early in the design process is useful in planning for the rest of the project design process. This part list is tentative from an early stage of the project and might change as the project becomes closer to implementation. But as of right now this is our layout when it comes to parts for this project.

Table 6: Parts

Part name	Description	Possible numbers
Motors	Motors are used for the wheel legs, neck, tail, and joints.	8
Camera	The camera is integrated near the eyes for use of AI vision.	2
Screen	Screen is replaced eye sockets and used to display information	1 rectangular
Microphone	Used to receive audio input for commands	1
Speakers	Located where the ears would be. Gives output audio.	1 or 2
Wheels	Used instead of legs for easier mobility.	4
Controller	Goes inside and controls peripherals.	2
Battery	Inside powering all electronics.	1
Virtual assistant	Interactive component of the robot, providing standard features of a virtual assistant	1
Plastic Chassis	3D printed body	1
Touch sensor	In the back for “petting detection”	1

2.4 House of Quality Analysis

For the purposes of analyzing the engineering requirements of this project with respect to potential customer requirements and demands, a House of Quality (HOQ) analysis was performed (shown in Figure 1). This analysis focuses on understanding the relationship between the two types of requirements under consideration, as well as the correlation (i.e., trade-off) between each engineering requirement. It additionally observes the weight of each customer requirement, assigning a level of priority to each requirement. This HOQ (Figure 1) analysis also considered inputs regarding the strengths and weaknesses of potential competitors’ products.

This analysis in its entirety provides an overview of how various aspects of the project interact and challenge each other. It also provides a graphical glance at our potential final product against similar products on the market (albeit against a very broad price range). This analysis was done with very early-stage research against similar existing products, and more research into similar projects and products are outlined later in this documentation.

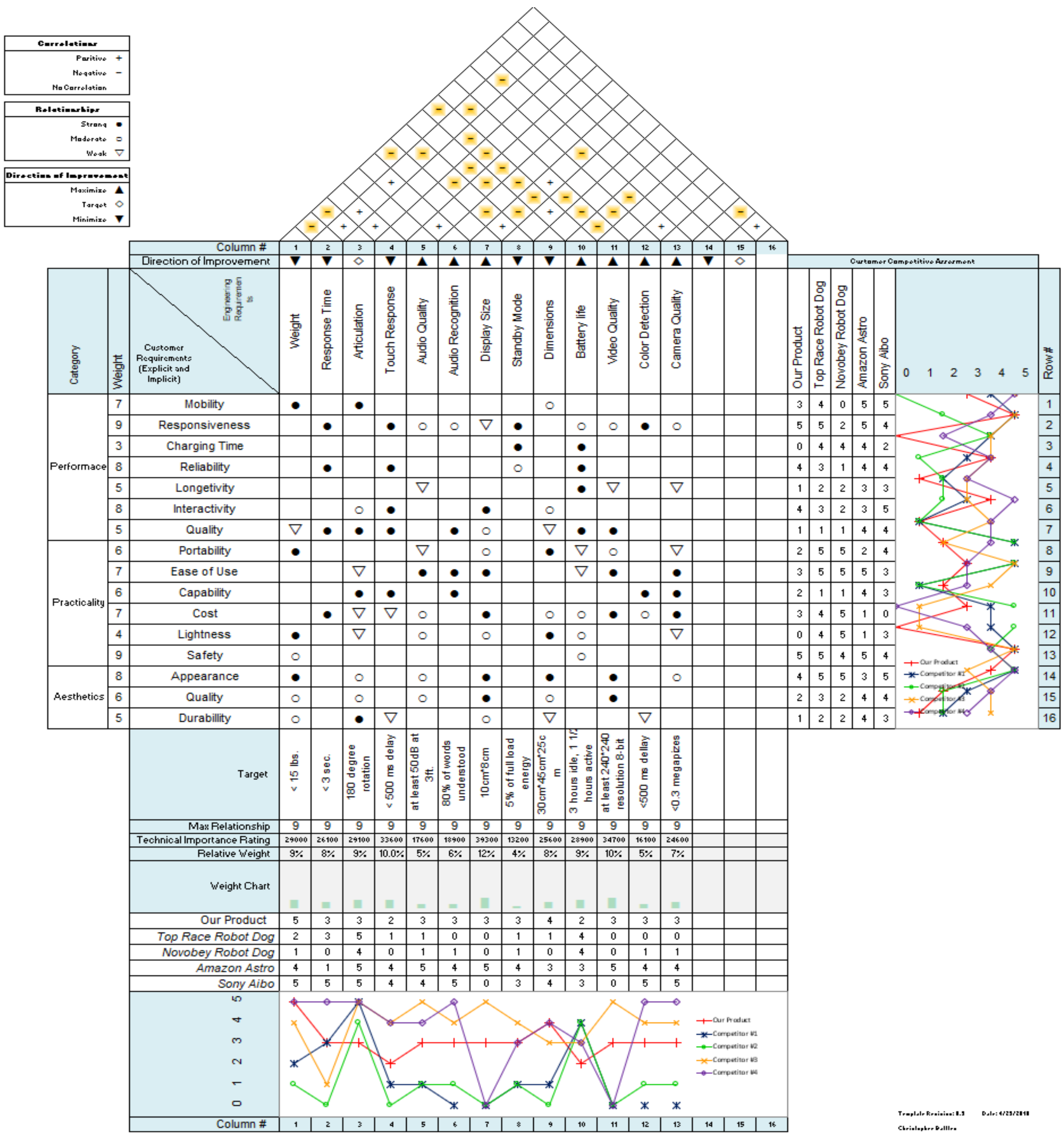

Figure 1: House of Quality analysis chart & associated competitor comparisons

Figure 1 shows the House of Quality for our product. Specifically, it shows how engineering requirements and customer requirements relate to each other, as well as how our design compares to competitive design in the market. There are several engineering and customer requirements for out project. Observing carefully, it is clear quality and capability have very strong relationships with both types of requirements. Customers want good quality and capability for this product. Equally, many of the engineering requirements are determine the quality and capability of the product. Knowing this relationship helps understand the importance of satisfying engineering requirements. When observing our scores with the competition, our design performs right in the middle. The robot dog has a better score than toy robot dogs, but a lesser score against high end robots. This is the goal range for this product making these results logical. However, it is noticeable that each robot design has its own good and low scores. Knowing this before creating the robot, we can think on how to improve the low scores while also understanding that some low scores should be expected.

The diamond in the upper part of the House of Quality shows the relationship each requirement has with each other. At first, it is confusing and hard to read. The main points to take away from is that the better the sensor and other components of the robot are, the more negatively battery life is affected. This means that the team needs to pay close attention on the amount of power consumed by each of the components in the robot. Another observation from this diamond is that weight also has a major involvement with the rest of the components. Better components tend to be heavier and affect the weight. This in turn affects the performance of other components. This tells the team that we must also make sure that the weight is carefully observed to not negatively affect the design of the robot dog. A final observation that can be made from this diamond is that better sensors lead to a lower response time, which leads to a robot of better quality. When viewing all three of these observations together, the team can notice that the robot wants to have good sensors to reduce response time. Having good sensors affect battery life, which also leads in a possible increase in weight. This tells the team that we must find a balance of quality sensors that are not very heavy and do not consume much power.

2.5 Overview Block Diagrams

To help with the design of the project, block diagrams are created and divided among the different team members. To simplify further, the bock diagrams are divided into hardware and software components of the project. Figures 2 and 3 provide the block diagrams that were used to help plan how the robot was designed. The blocks contain the major components of the design. Each component has a person assigned that oversees and makes sure the component functions properly and it available when it is needed. Everyone worked in each component. Figure 2 shows the hardware block diagram. The diagram divides each major section of the robot into two people. First, we can clearly see that the robot needs at least two different microcontrollers. One of them is controlling the AI and virtual assistance interfaces. As can be seen in the software diagram (Figure 3), this microcontroller must be powerful as it must manage several heavy tasks. The other microcontroller used to control all peripherals that are not directly controlled by the virtual assistant AI. This are mainly the motors for mobility. This microcontroller can be simple with little processing power. Second, diagram shows that several peripherals were used to imitate the different sense the robot must have. The robot dog includes microphones and speakers for the auditory sense, a touch sensor, for touch sense, a camera for visual sense, and motors for articulation and mobility.

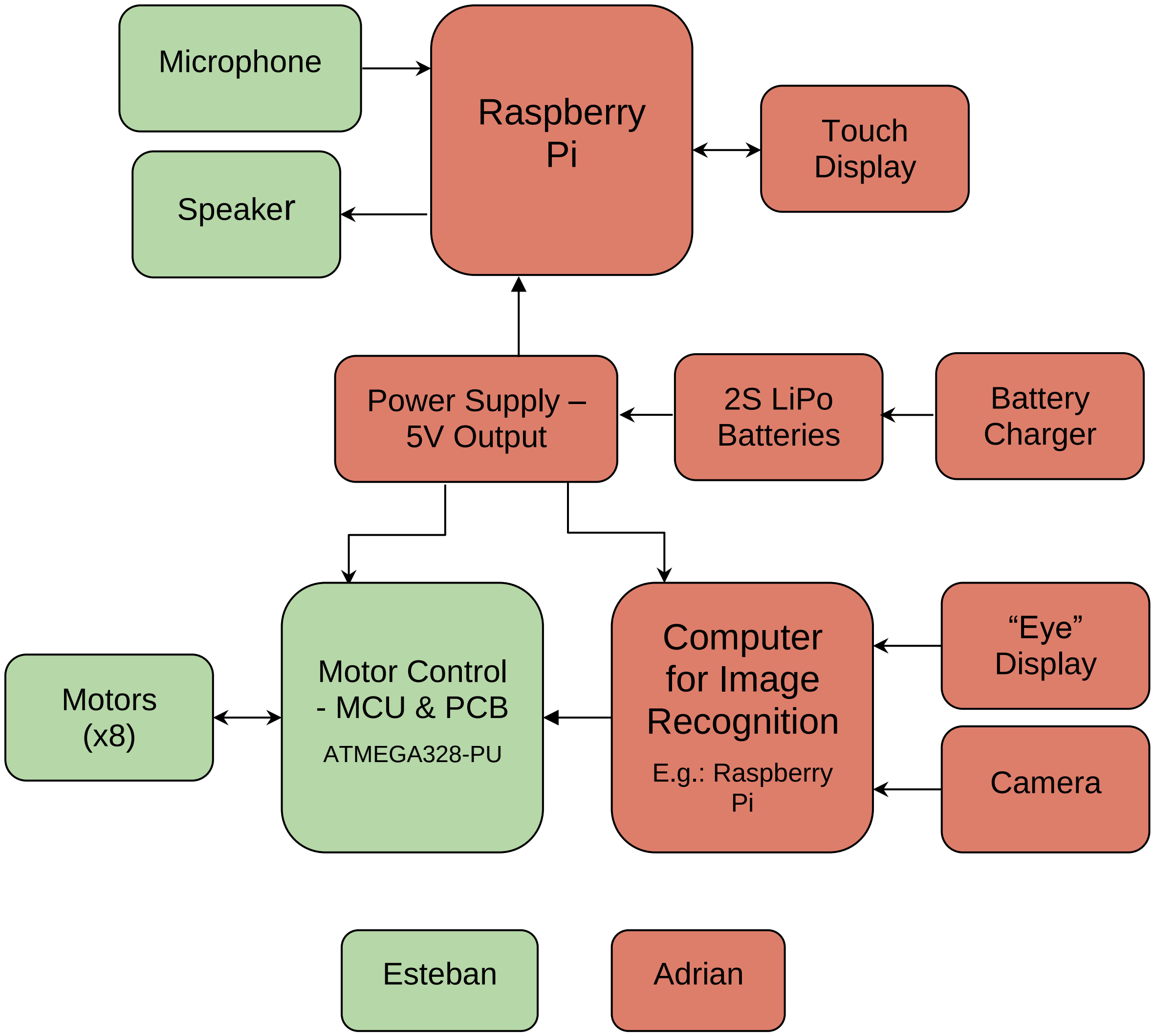

Figure 2: Hardware Block Diagram

Figure 3 shows the software block diagram. This project requires both hardware and software to work as desired. The software components are shown in Figure 3 and divided in blocks to help visualize each component better. Like with the hardware diagram, the software diagram contains all the main section of software that need to be implemented for the robot to work as desired. The software diagram is further divided into major sections named sensors, perception, navigation, vehicle control, and global services. From the software diagram, we can see the software needs to be capable of controlling all the peripherals. Furthermore, with the multiple sensors, the robot dog should be capable of having obstacle recognition, face recognition, detect when the touch sensors are activated, and listen to commands from the user. Additionally, the software implemented must allow the robot to move effectively without crashing into objects or falling.

By combining both Figure 2 and Figure 3, it is clear what all the components used in the robot are and how they are to be divided among the team members. It is important to reiterate that everyone worked on each part of the robot; the names are there to signify who is responsible for each section. Just by viewing the diagrams it is not possible to understand the design of the robot. For example, the diagrams never explain that the robot is shaped like a dog. The diagrams just explain the components that are used. There is an entire section that goes in detail on how the design was selected and the reasoning. Moreover, the design is heavily influenced on which components hardware components are used and how each of them interconnect with each other. Hence, both of these block diagrams are important for the overall design of the robot dog.

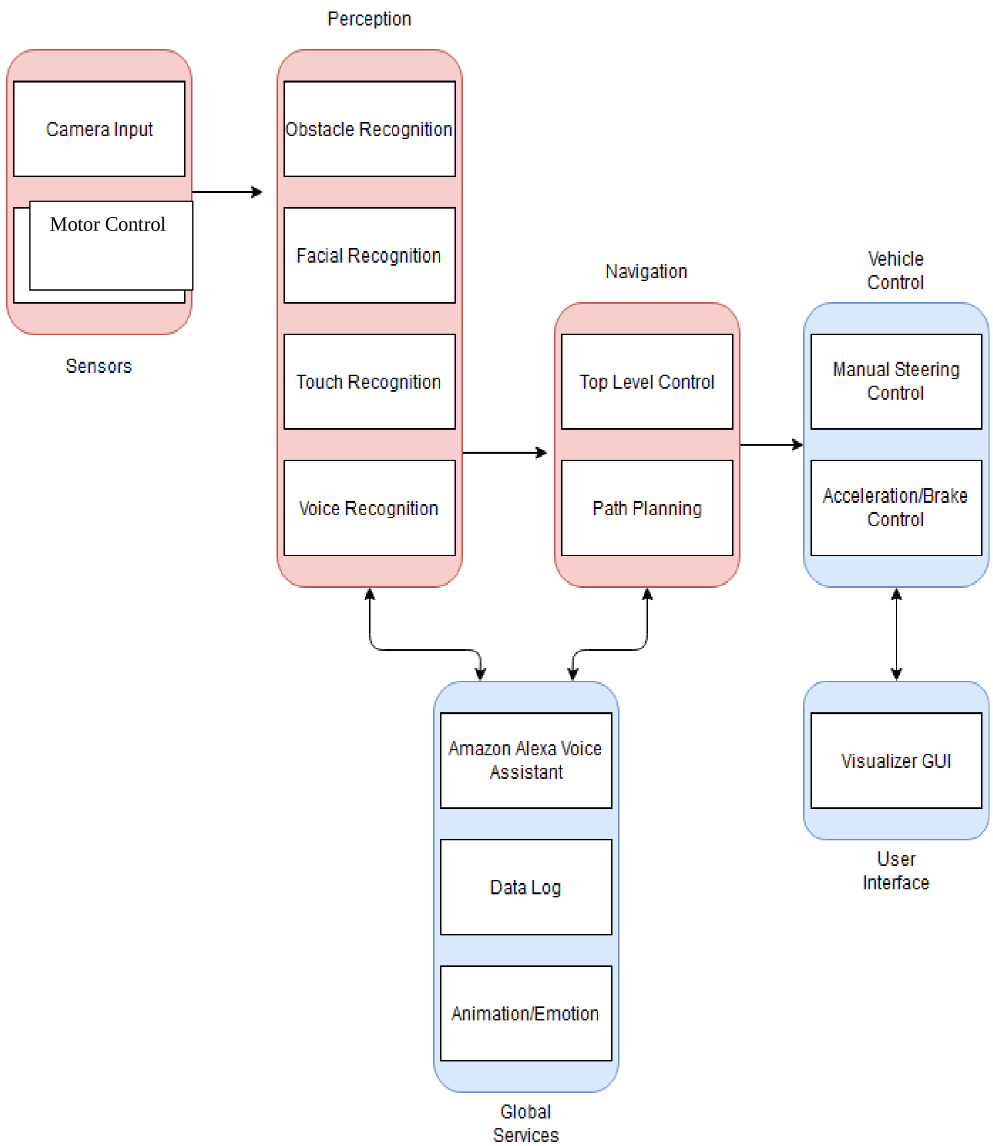

Figure 3: Software Block Diagram

The block diagram in Figure 3 is divided into two different colors. Each color represents a different team member in charge of overseeing those particular sections. The division was grouped with the intention of dividing the responsibility evenly. Previously, it was mentioned that the entire team is working on each individual part. The separations are necessary and helpful to ensure that there is one person specifically responsible for the completion and progress of each part.

3. Research

Globalization and the internet provide many advantages when creating a new project. Many designs, technologies, and products already exist that can prove extremely beneficial for this project. Several other people have already tried to create our project using their own ideas and method. This section has the purpose of explaining how every design, technology, idea, and product influenced our own design. Furthermore, the section provides knowledge of new technologies that can be potentially used in the product. Research is necessary to create the best project available using the least number of resources and the best technology available. In other words, proper research ensures each of the parts of the robot dog are the best in its price range and accomplish the requirements desired.

With proper research, the team can find the best technology and components that can meet all the standards and constraints presented on this project. Furthermore, proper research allows the team to observe multiple designs and analyze what worked and failed in each of them. This knowledge is then used to create a project that takes all the advantages of the previous designs and tries to improve of their weaknesses. Research is the most important part of this project as it heavily determines what the project looks like and how the project is designed.

3.1 Existing Similar Projects and Products

During the research process, several existing projects and products were discovered that are similar in concept to the ideal final version of this project beyond the prototype stage. They range from open-source community/university projects to highly polished commercially available products. This means some of the projects had a very small budget while others had millions to work with. Likewise, some of these projects had a lot of people, time, and resources dedicated to the project, while others had the opposite number of resources. Table 6 shows a summary depicting the similarities and differences among the different products. These similar products are listed below:

- Top Race Robot Dog
- Novobey Robot Dog
- Amazon’s Astro
- Sony’s Aibo.

3.1.1 Top Race Robot Dog

The Top Race Robot Dog is a remote-control robot dog for the ages 2 and up. Figure 4 shows how the robot looks. It comes with a remote control that has set functions for the dog and its movement. Functions include, imitating 10 animal forms like slipping, lie down, standing on 2 legs whereas 7 physical gestures function like forward, backward, sing, dance, walk, crawl, expression performance and can be controlled with the remote. The remote has 50-foot range remote. The Robot Dog is made from food grade plastic which is completely safe for kids. Built with a battery voltage of 7.4v and a battery capacity of 600mAh which is a rechargeable battery. The remote control needs 2xAA batteries which are not include. This robot dog has a Length of 3.94" inches, Width of 5.91" inches and Height of 13.74" inches. This product has a one-hour USB charging time. Which holds a charge up to 50- 70 minutes or play time. This product cost $55.99 in today market. This product is well made and has one key function that we implemented in our dog, and this is voice recognition. We need our dog to be able to recognize your voice so you can interact with your Alexa or Google home thru voice command.

Figure 4: Top Race Robot Dog

3.1.2 Novobey Robot Dog

The Novobey Robot Dog is a robot dog without a remote for the ages 3 and up. The robot dog is shown in Figure 5. It is designed to introduce kids to the STEM field. It is controlled by sound, a sensor inside the plastic bone, and touch sensor. It has a programed movement system which is synchronized with musicmaking. It also has a sound bite of a realistic bark, eating sounds, yawning, and snoring sleep mode and has eyes feature. The user is able to assemble and disassemble different parts which is what give kids the introduction to the STEM field. All parts of the dog toy are sturdy and durable and in line with international safety certification standards. Using harmless and non-toxic ABS plastic, it does not cause physical harm to children when playing. The robot dog uses 2xAA 1.5v batteries which is not included. This product has a Length of 7.87" inches, Width of 7.08" inches and Height of 2.16" inches. This product cost $24.99 in today market. This product is great for kids has two key function that we implement in our dog, and this is the dog appearance and behavior. We need our dog to look and behave like a dog so people can be more comfortable when our dog is near them.

Figure 5: Novobey Robot Dog

3.1.3 Amazon's Astro

The Amazon Astro uses advanced navigation to get around your home without any assistants and avoid most obstacles in real time. With its navigation sensors and obstacle sensors, you are able to select a room in your house with the amazon app on your phone and it goes to the room you have selected. Also, with the sensors the Astro can follow you around for entertainment purposes or even find you in your house if you have a phone call, message or even an alert. It also has a cup holder so it can carry your drink. Thru API connection if you have any Ring or Alexa products it can connect to them to add all the features that these products have to your Astro. For example, it can schedule autonomous patrols, configure Astro to investigate detected events, save videos in Ring's cloud storage, call 911, have smart alerts to sounds, start barking when it recognizes motion outside, and sound a siren if it detects someone inside your house. It has a built in Lithium-Ion rechargeable battery and has a Length = 16.7" inches, Width = 9.8" inches, and the Height = 17.3" inches with 45-minute charging station in which the robot returns to the charging station when battery is low. This product cost $999 for now but increases to $1,449.99 after day 1 releases. The releases date has not been confirmed yet. This product is well made and every feature in this robot is what we want to implement into our dog. The Amazon Astro is exactly what we want are project to be, but it just has one problem it doesn't have the characteristic of a dog.

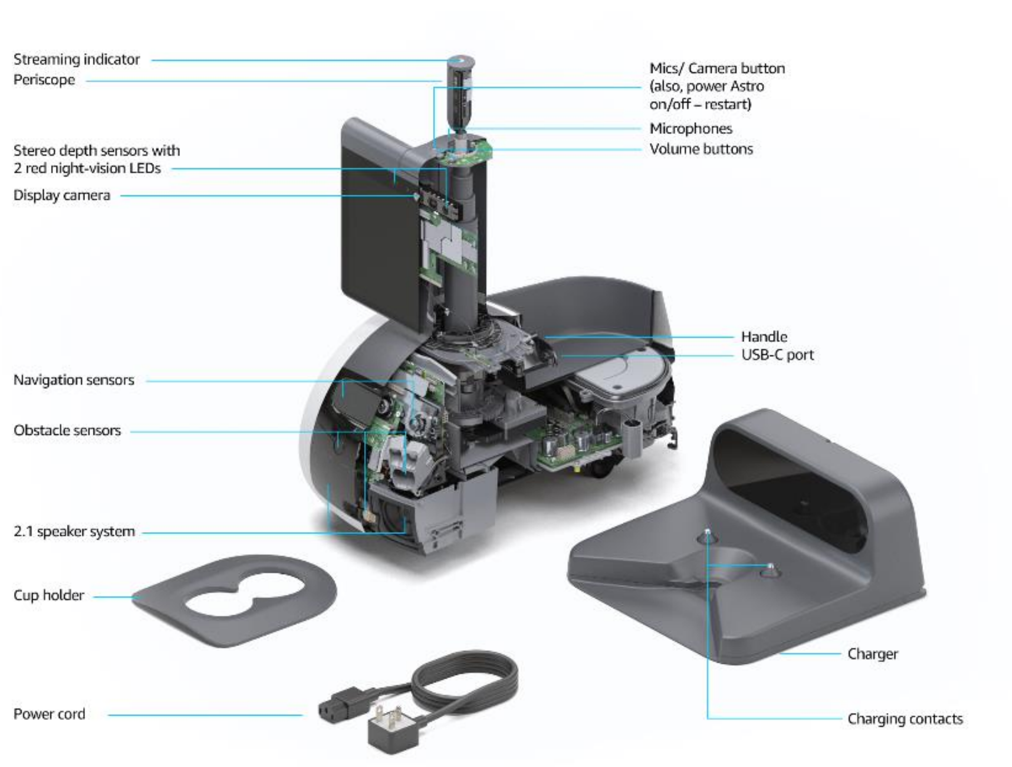

Figure 6: Amazon Astro

3.1.4 Sony's Aibo

The Sony's Aibo is Sony attempt to make a real-life dog with A.I. technology. Their goal is to make a robot dog that is exactly like a dog. With deep learning the Aibo start building its own personality from day one and auto update constantly. It learns from its surroundings and interact with people just like a dog would. It is also able to follow you around and learn new tricks. With 3-axes Head, 1-axis, 1-axis Neck, 1-axis Waist, 3-axes for each Legs so 12-axes in total, 1-axis for each ear so 2-axes in total, and 2 axes in the tail for a grand total of 22-axes in this robot. This is what was needed to give the look its realistic movements. With a 64-bit Quad-Core processor, Touch senor, front camera, 2 OLED Eye displays, 4 Microphones, and a 4 cell Lithium-Ion battery which has a 2-hour operating time and 3-hour charge time. All able to fit within the dimension of Length = 12.0" inches, Width = 7.08" and the Height = 11.5" inches. Now this self-learning dog can be yours today, but it costs you $2,899.99 in today's market. Which is not cheap give the size of the robot and purpose of this robot. But the AI abilities that this dog has is groundbreaking and helped the way we study AI and AI interactions. But when it comes to our project, we are not going to get anywhere close to the software capability and the movement that this robot possess that this robot possess. But we are going to try are best to make this dog not only move like a dog but also look and interact as one to.

Figure 7: Sony's Aibo

Below is a table of some the specifications of the 4-robo dog that is on sale in today market. If you notice the age group and material all share the same result. 3 and Up for the age group and the material is made of plastic. The reason we pick these 4-robo dog is because each one has features that we would like to implement into our robodog from a wide price range. For the Top Race robot dog, we need our dog to be able to recognize your voice so you can interact with your Alexa or Google home thru voice command. The Novobey dog has two key function that we want to implement in our dog. One is the dog appearance, and the other is the dog behavior. We need our dog to look and behave like a dog so people can be more comfortable when interacting with our product. When it comes to Amazon Astro it has every single feature, we would like to have in our robot dog and then some. The only problem is the Astro is not a dog. The Sony's Aibo is the perfect robot that acts like a real-life dog and the is what we want in our robot. We want to have our robot behave like a dog. From these 4 products we can see who competitors are and now we know what our standards need to be to beat are competitors.

Table 7: Similar Designs Specifications

Brand	Top Race	Novobey	Amazon's Astro	Sony's Aibo
Dimensions	3.94" L x 5.91" W x 13.78" H inches	7.87" L x 7.08" W x 2.16" H inches	16.7" L x 9.8" W x 17.3" H inches	12.0" L x 7.08" W x 11.5" H inches
Age Group	3 And Up	3 And Up	3 And Up	3 And Up
Material	Plastic	Plastic	Plastic	Plastic
Special Feature	Talking Robot Dog	Dog appearance and behavior	Alexa and Ring API connection	Realistic characteristics of a dog and self-learning capabilities

3.2 Market Analysis

The smart assistant market is still a relatively young industry. The first emerging modern assistant came with the release of Apple Siri in 2011. Following the eventual success of Apple's smart voice assistant, multiple companies began releasing their own versions with Amazon Alexa in 2014, and Google Assistant in 2016. Over the years, virtual assistants have become more common in average households. Each of these companies providing different models. In today's market, these smart assistants are found in cars, speakers, phone, laptops, and even refrigerators. These companies are trying to make as many devices use a virtual assistant as possible. This is where the robot dog comes in. The robot dog is like any of the other smart devices where these companies are trying to add virtual assistants. The advantage this product has over others in the market is that robotic pets and animals are still very expensive. These companies have the money and manpower to make these smart animal devices, but they make them complicated. Our product is simpler, and should, therefore be much cheaper than the rest of the competition. This allows the product to strive in the current market.

Below, analysis of the virtual assistant market is observed to determine how useful and expensive our project would need to be. Moreover, the analysis proves why our project is beneficial and necessary today. Virtual assistants are being integrated in almost every electronic device. Once every device has a virtual assistant, the market shift in a direction where they create their own devices with virtual assistants. This market is still new, but the data below shows it is growing. This is the market the robot dog enters. This proves how a successful virtual assistant robot is something large virtual assistant making companies are looking for and is the likely future of the technology. Our mission is not to create a product that can compete with the large companies, but rather show an idea of what can be achieved with a small number of resources.

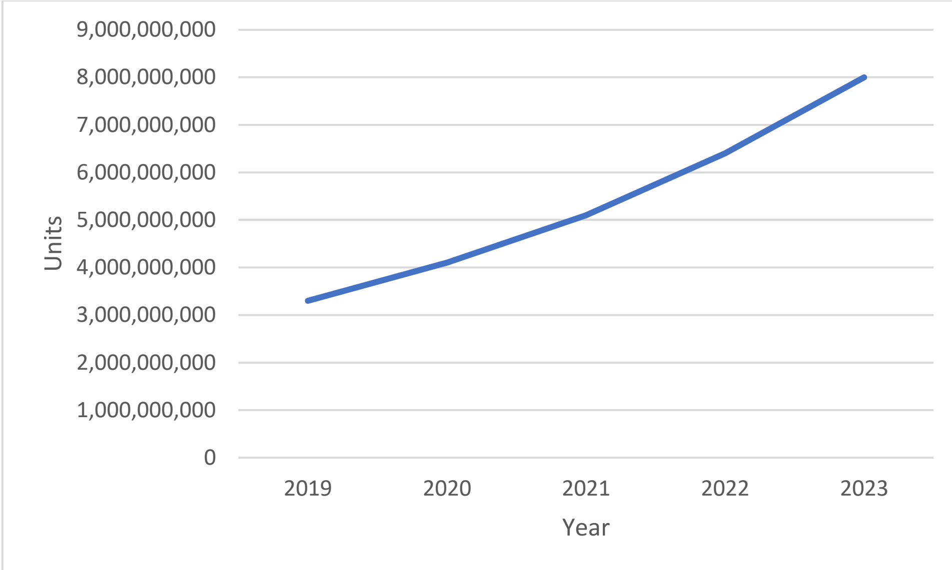

Figure 8: Number of Worldwide Voice Assistants in Use (Statista, 2020)

The graph above (Figure 8), shows how the virtual assistant market is steadily increasing. Since our product utilizes a virtual assistant, it would fit directly into this market. The market for robotic animals is very scarce and not common at all. Currently, it is dominated by Boston Dynamics, but they robots are too expensive. The robot dog in this project is not nearly as complicated but and could never compete in such an advanced market. The growth of the voice assistant market shows that the robot dog might has its use in an average household. The next graph shows the different companies and their share of virtual assistant in the market.

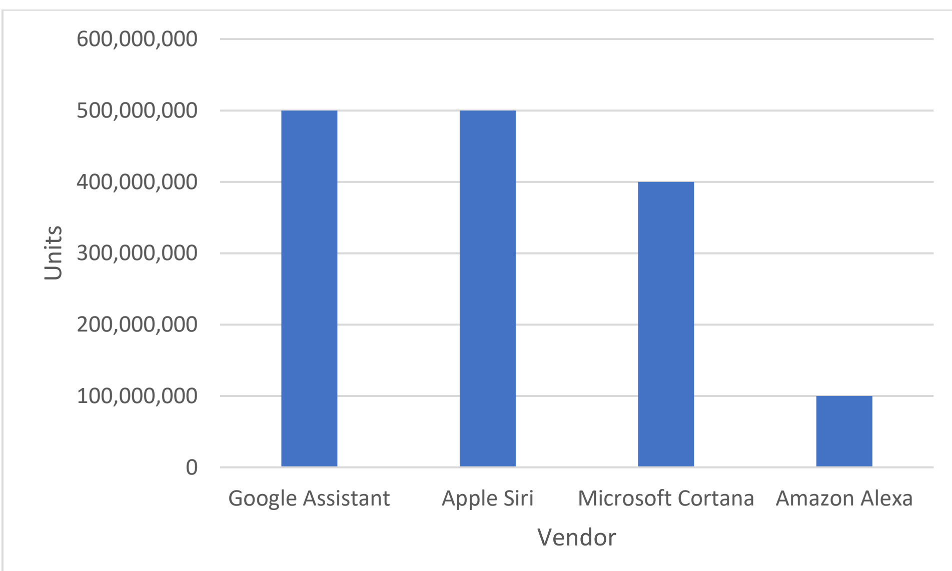

Figure 9: Voice Assistants Market Share by Vendor (t4, 2021)

The figure (Figure 9) demonstrates that there are four major companies participating in the market. The biggest ones are Google Assistant, and Apple Siri. This is likely due to these being the virtual assistants present in smartphones and tablets. This means these are as prevalent as the smartphone is today. Microsoft Cortana falls shortly behind as their virtual assistant is in a few phones and at least half the share of laptops and PC desktops. The most interesting value in this graph is the Amazon Alexa. Alexa appears to be in every household, yet they only share a quarter of the units the big companies possess. This is because, generally, people only have one or two Amazon Alexa's in their home. There is no need for more. Alexa is designed from controlling appliances in the home. Only until recently has Alexa started integrating their virtual assistant in homes.

The next graph (Figure 10) shows where each company has their focus of virtual assistant. For example, it is apparent Siri is most prevalent in smartphones, and devices like the Apple Home (a speaker with a virtual assistant) are not nearly as successful for them. A similar situation happens with Google Assistant. Alexa, in the other hand, has a small share of the virtual assistants used in smartphones, but has the biggest share in household speakers. Currently, most cars have Apple CarPlay, which uses the Siri virtual assistant provided by Apple. The robot dog designed in this project would fall on the other category of this graph. The combination of graphs demonstrates that these may be the major competitor in the virtual assistant market, but they have a very small share of assistant when it comes to devices that are not as prevalent in a household. It demonstrates that the robot dog does have a future in this market. Fortunately, Alexa and Google Assistant allow users to connect their own devices and use one of their own virtual assistants. This means that our robot dog is capable of being one of the top virtual assistants, while also participating in the market where the large companies have a small grasp.

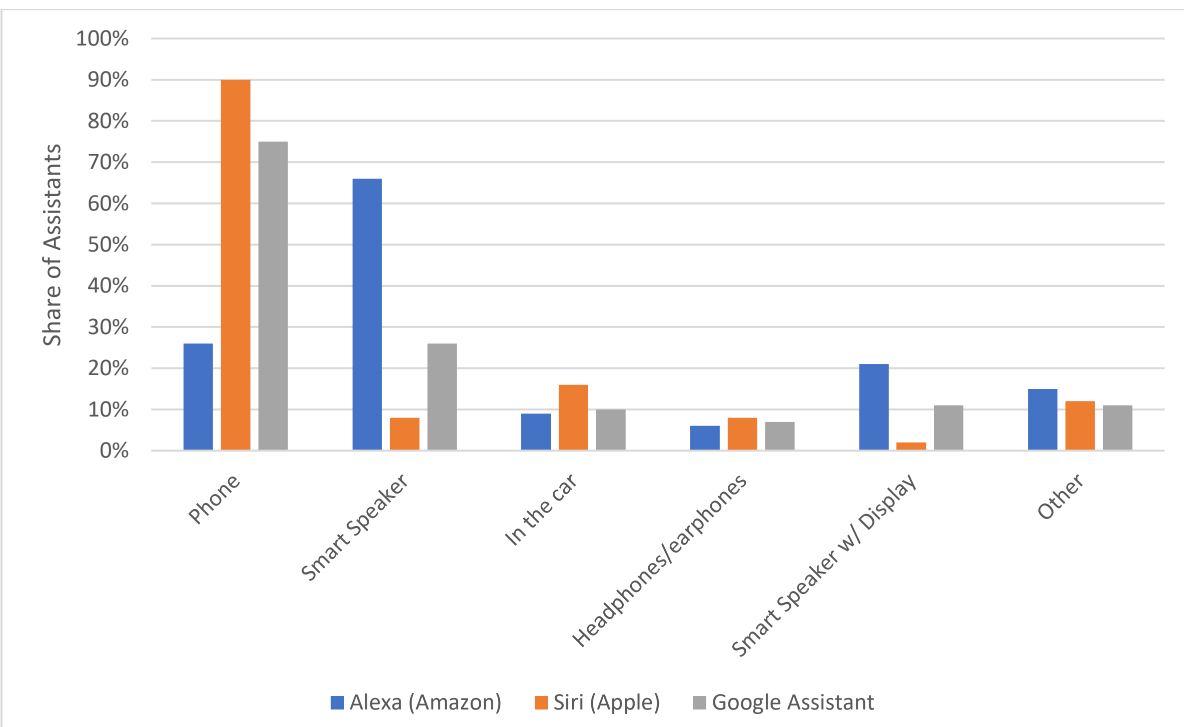

Figure 10: Voice Assistants Usage in the United States in 2021 (Statista, 2021)

3.3 Relevant Technologies

This project involves the integration of many different technologies, both on the hardware and software side of this project. These technologies help make the robot efficient and effective when performing its desired objectives, while also facilitating its creation. Over the years, many new technologies have been developed that can make the robot dog many functions that would have been impossible a few years back. This project tries to use the most advance, yet non-expensive technologies out there. The robot dog tries to combine some of the newest technologies and create a new product out of them. On the hardware side, servo motors, pulse width modulation, a battery system, and DC-to-DC convertors are all necessary technologies to include. Also, we are using a raspberry pi 4 and a ATmega328P microcontroller. For the software side we are using TensorFlow with python for the AI vision and just python for the LCD display, AutoCAD for the body design, and C++ for the robotic movement.

3.2.1 Servo motors

A servo motor is a small motor that is highly efficient and is very energy efficient. We used this motor to give our robot dog walking ability, ability of turning it head, and movement of the tail. A servo motor is just a DC motor with a potentiometer and a control circuit. With the motor attached by gears which control the movement. When the motor rotates the potentiometer changes the resistance so that the control circuit can control how much movement needs to happen and the direction. What makes the servo motor very energy efficient it all comes down to the shaft of the motor. When the shaft hit the correct position, the power supplied to the motor is stopped. Now if the motor is not at the desired position, it tries to correct itself. This makes sure that the motor uses the correct amount of power to complete it takes.

A servo motor is controlled with the pulse width modulation. So, there is a minimum and maximum pulse and a repetition rate. A servo motor normally turns 90° with a total movement of 180°. There is normally 3 servo position 0°,90°,180°. Depending on the duration of the pulse from the PWM it determines the position of the shaft on the motor. The servo expects to see a pulse ever 20ms. Let say a 5ms pulse turns the motor to 90° then anything less than 5ms turns the motor to 0° and anything more the 5ms turns the motor towards 180°. When it comes to the servo it moves to the correct position and holds that position until a new pulse comes in or you repeat the same pulse over and over. If the motor doesn’t get a pulse by 20 milliseconds, then it returns to its natural state.

There are two major types of servo motors. One of them uses AC while the other servo motor uses DC. AC servos motors can handle higher current surges and are normally used in industrial machinery. For this project, we are using DC servo motors since it is better to use DC servo motors on small robots. They are easier to control and easier to implement. They are also much more common in smaller sizes. Most of the smaller DC servo motors already come with their own feedback controller and their corresponding gearbox.

3.2.2 Pulse Width Modulation

Essential for the use of the servo motors. Pulse Width Modulation (PWM) allows for immediate control of the voltage supplied to the motor. This allows the motor to be either fully on or completely off. This provides a much better power usage and less heat consumption than using variable resistors. As the name suggests, PWM allows the user to control the width of the pulse signal, also known as the duty cycle. Instead of sending a continuous DC signal, a square wave is sent with a constant amplitude and a constant frequency. High voltage keeps the motor on while the low voltage turns the motor off. Varying the length of the time high voltage is set during one period (or pulse) changes the average DC signal the motor receives. This is also known as changing the duty cycle percentage. For example, a generic square wave has a 50% duty cycle. During one period, the square wave is in high voltage for half the time (50%). With PWM, the duty cycle is changed to 75%. Now, the wave is on high voltage for three fourths of the time and only low voltage for a quarter. Overall, the DC average power received by the motor increases. Additionally, by doing this we are not affecting the amplitude of the signal, so the motor always acts in full capacity, while also increasing its speed. Changing the pulse provides for much better speed, accuracy, and precision. A potentiometer or software can be used to change the duty cycle of the signal.

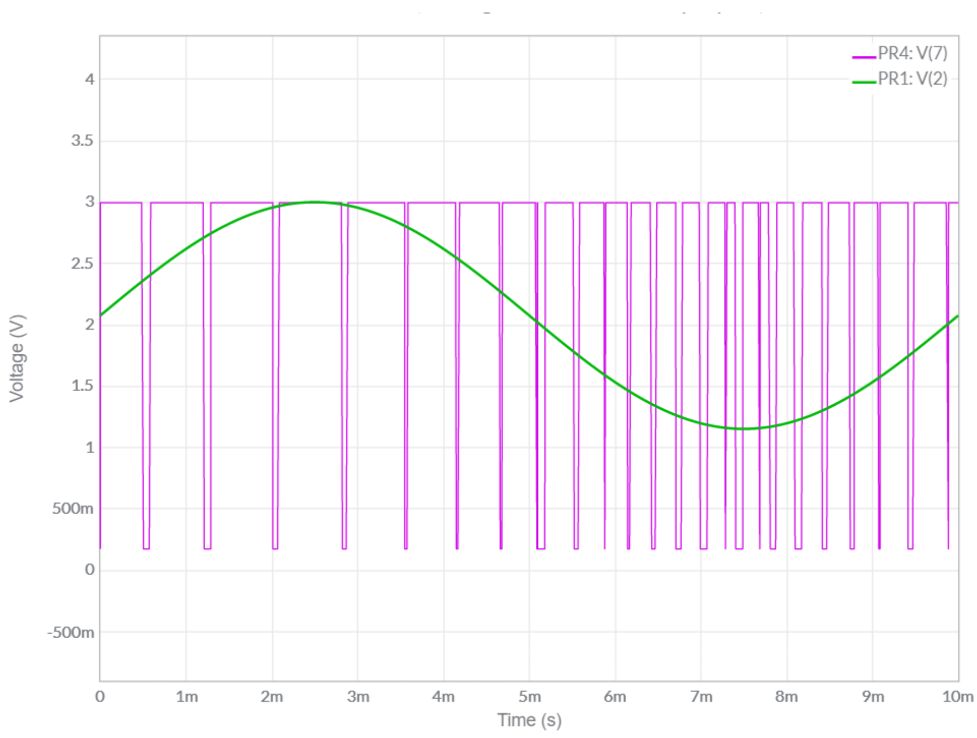

Figure 11: Example of PWM output

The figure above (Figure 11) shows how PWM can be used in complicated ways to change the overall energy given an output. In this example, the PWM is controlled by simple sine wave. As the voltage in the sine wave increases, the duty cycle increases. When the sine wave voltage decreases, the duty cycle of the output decreases as well. Changes like these are simple to do with a proper integrated circuit controller. This allows the power given to the output of a circuit to be precisely controlled. In the case of a servo motor, for example, PWM can be used in conjunction of the feedback the servo motor provides to control its location and speed, with impressive precision.

3.2.4 Voltage Regulator

Voltage regulators are a technology which maintains a stable desired voltage under a changing input voltage. Simple linear voltage regulators can be built using rectifiers but are very inefficient, and better systems have been developed. Linear voltage regulators are cheap and found in convenient packages, but the heat dissipation considerations alone make them difficult to account for in a compact design. Switching voltage regulators are more efficient system which dissipate less power than linear voltage regulators and are more flexible. They control output voltage by turning on and off at a high frequency, such as in the hundreds of kilohertz, and adjusting the duty cycle of said switching. Voltage regulators can be found in any power supply system and are the basis for DC-to-DC convertors that was used in this robot to utilize batteries. Figure 12 shows one possible configuration of a step-down, or buck, switching voltage regulator circuit.

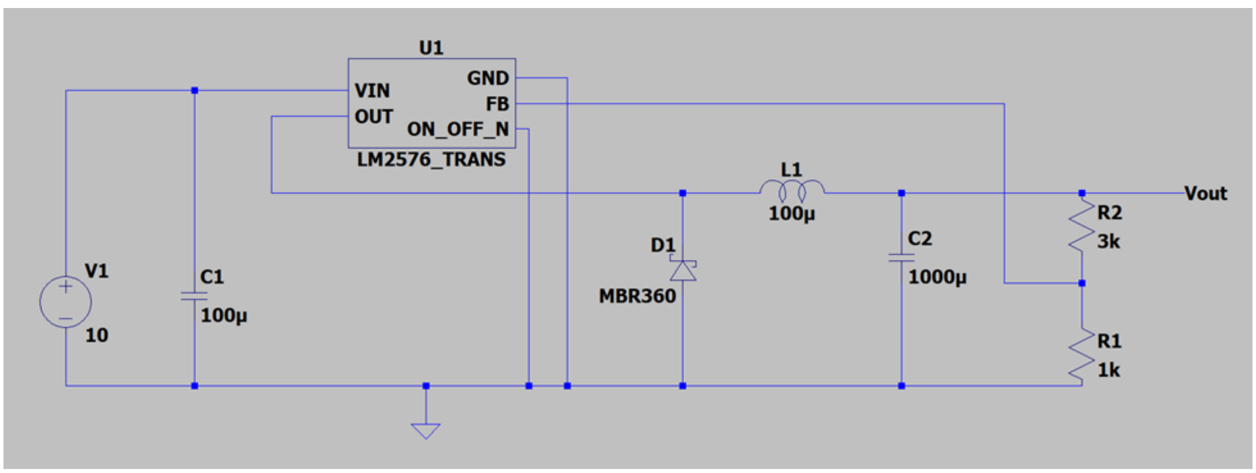

Figure 12: Example switching voltage regulator circuit

3.2.5 DC-to-DC Convertors

The power supply system for this robot requires the use of a DC-to-DC convertor. DC-to-DC convertors are a class of circuit which takes one DC input voltage and outputs another, or several DC output voltages. Voltage regulators and DC-to-DC convertors are closely related, with voltage regulators often making up the center IC in a DC-to-DC convertor. Specifically relevant to this project, DC-to-DC switching regulators are used. There are several different topologies for DC-to-DC convertors, such as boost, buck, boost-buck, and flyback convertors. An example of a boost, or step-up DC-to-DC convertor circuit is shown in Figure 13. DC-to-DC convertors can be used in low to high power settings. This project constitutes a low power setting, and the DC-to-DC convertor is used to convert the output voltage of a lithium polymer battery solution to one appropriate for our motor control PCB and Raspberry Pi computer to operate from.

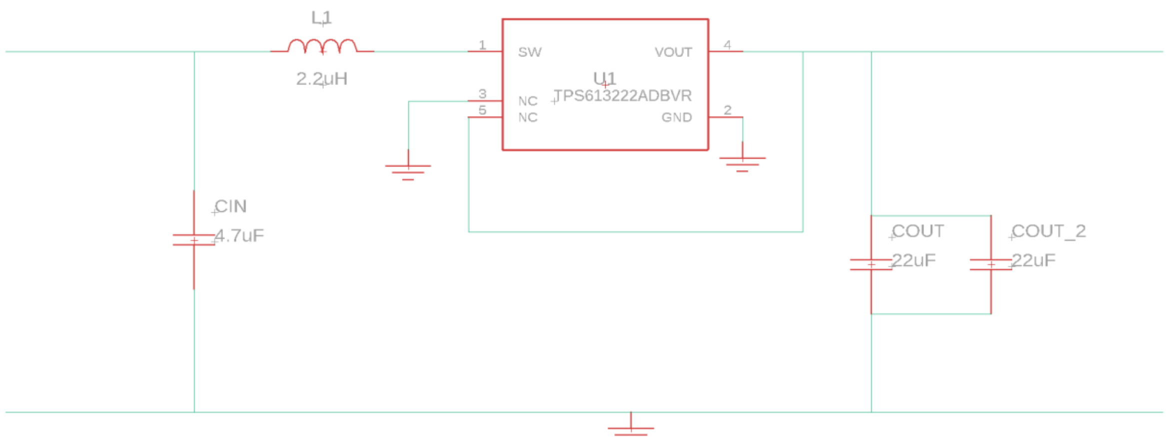

Figure 13: Example of a DC-to-DC convertor circuit using a boost topology for 5 V output

3.2.6 AI Assistants

Artificial Intelligence (AI) assistants combine a group of AI technologies to facilitate the use of devices. Some of these technologies include a Wake Word (WW) detector, Automatic Speech Recognition (ASR), Natural Language Understanding (NLU), Dialogue Manager (DM), and Text to Speech (TTS). Combining all of these, assistants can perform multiple tasks by simple giving voice commands. In the present, AI assistants are mostly used to perform common technological tasks. They are not good for maintaining a conversation and struggle to understand hidden conversational meanings or clues hidden within the context. Commands need to be direct and specific for the AI assistant to understand what they are being asked to do. Future research will eventually allow these assistants to be able to hold conversations and have a better understanding of the context of different commands.

Wake word detector involves a complicated engine that allows a voice assistant to “activate” upon hearing the wake word. For example, Amazon’s Alexa “wakes up” by saying “Alexa. Apple’s Siri uses “Hey Siri” as its default wake word. As the name suggests, ASR is an AI technology that allows virtual assistants to understand words and phrases spoken in a specific language. This is then improved with NLU. NLU is an AI technology that allows the device to combine the words and phrases it hears and understand the meaning behind them. It is the technology responsible for knowing what you mean when you say a specific sentence. The DM is the AI technology responsible for giving the assistant specific verbal responses to questions or phrases. Finally, TTS technology allows assistant to “read out loud” anything they read. This is how an assistant can search something on the web, and then tell you what the search results are.

As can be seen, combining all these technologies together can make for a useful AI. Incorporating an assistant on a smartphone allows the user to update their calendar, send messages, dial numbers, search on the web, and many other functions by simply talking to the device. Having an AI assistant on a hub on multiple IoTs like Amazon's Alexa, the assistant can control multiple devices with regular voice inputs. Our project is implementing this advanced technology to make our robot not only a robotic dog, but also have the multiple functions that AI assistants provide**.**

3.2.7 3D Printing

3D printing is the creation of a three-dimensional object from a digital three-dimensional object. It's a process by which one of various materials is ejected and joined by a computer. The most common process for 3D printing is using a thermoplastic material which is extremely moldable at certain increased temperatures and solidifies quickly upon cooling. Currently the most popular process is called fused deposition modeling (FDM) in which filament is fed from a large spool through a moving, heated extruder head, and is then deposited on the growing work. The first 3D printers which used the modern additive manufacturing technique printers used today, were first invented and used to some degree in the industry in the 1980's. However, the most common technique of FDM did not reach maturity until the 2010's and didn't emerge prominently in the consumer market until the 2020's.

The main casing of the robot is made using 3D printing technology. 3D printing allows engineers to create solid objects in three dimensions from a digital file. This allows engineers to easily create and print parts that are specific to their needs and not widely available. Depending on the material, printer and printing technology, the parts vary in durability and strength. In the case of the robot dog, the material used is a type of plastic (either PLA or ABS). This creates a casing strong enough for its purpose. To create a 3D-printed object the object must first be digitally created using any software like SolidWorks. Once the object is created, it needs to be "sliced" into multiple layers so the 3D printer can read and print the file properly. The 3D printer works by printing layer by layer until the full object is printed. There are different technologies and materials used to print the object. The 3D printer used for this project uses the Fused Deposition Modelling (FDM) technology.

FDM is the most common and recognizable process for 3D printing. This process works my melting plastic filaments that are then placed in a heated extruder which moves depending on the image it has for the layer it is currently in. The plastic solidifies once it is cooled. After the layer is completed, more plastic is melted, and the extruder machines moves the extruder to continue to the next layer. Once all layers are printed, the final object is complete. FDM also requires a support technology to remove excess plastic. Depending on the machine this can be a water-soluble material that simply washes away (Soluble Support Technology). Another support technology is Breakaway Support Technology (BST). This technology creates supports which can be easily snapped off from the final object. The advantage of this technology is that is that the materials is uses are quickly cooled and can be sanded, painted, drilled, etc.

There are two common materials used for FDM. These are Polylactic Acid (PLA) and Acrylonitrile Butadiene Styrene (ABS). PLA is a special plastic that is bio-degradable, making it environment friendly. It comes in a variety of colors and is even available transparent. ABS in another type of technology also used for 3D printing filaments. It is a strong plastic that also comes in a wide range of colors. ABS is more flexible and stronger than PLA. ABS is also easy to find since is it sold by several non-proprietary sources. For the robot dog, either PLA or ABS work well. The dog is designed to be as light weight as possible and the size of a small dog. This allows the motors to not need much torque and the 3D printing to be cheaper and easier to create.

3.3 Strategic Components and Parts Selection

The robot needs several sensors and small components to perform all its functions. With present technologies, there are several ways to create such sensors and parts. Some parts simply have different manufactures and manners of working, while other parts work with entirely different methods and technologies. This section tries to explain all the different options available while also explaining the reasoning for selecting each part. In some scenarios, there are multiple parts that could satisfy all the requirements necessary for the robot dog. In these cases, one option is chosen with reasoning provided. There are many reasons as to why specific parts are chosen. After research, it is clear that the biggest influences in part selection are price, constraints, requirements, and part availability. There are many instances where a better part is available but above the price range. Likewise, there are instances where a part I perfect for the design, but constraints do not allow the part to be used in our project.

3.3.1 Microphone Array with Speaker

A microphone array uses more than one microphone and connects all their signals together. Combining these signals with a microcontroller allows the robot to identify where the sounds are coming from. This can be used to make the robot turn towards a specific direction or tilt its head to show a sign that it is listening at that direction. Naturally, the more microphones the robot has, the better it is at listening and detecting where the sound is coming from. For this robot, the specifications require it to have at least 2 microphones; one in each ear. The robot moves its head and body until it faces the person it is talking to. The more microphones the robot has, the better it can track location, but the more expensive it is.

A microphone array can use any number of microphones. The more microphones in the array, the better the signal and the more accurate the array is. This comes with the obvious drawback of price. The robot dog is not using only auditory feedback to detect where a person is; there is also a camera. Because of this, it is not necessary for the array to be complicated, as it works in conjunction with the visual sensors. Due to this, a simple 2 mic array is used. The 2 microphones help differentiate left and right and give the robot a sense of having only two ears.

When comparing microphones, it is important to understand their signal-to-noise ratio (SNR). This ratio explains how much noise in the environment affects the input received from the device. The same can be said with the speakers of the device and how they are affected by the noise in the surroundings. This ratio takes the overall signal input or output and divides it by the noise input or output. This creates a ratio, which can be expressed in decibels, to explain how loud or clear the signal is when compared to the noise. A high SNR means a better-quality device, but as many have probably experienced, high gain boosts on audio are very noticeable and can detriment the overall quality. However, the team requires a microphone with enough power to comprehensively read the input from the microphones with regular ambient noise. Likewise, the speakers need to be loud enough to be heard from someone standing a meter or two from the device with ambient noise.

There are several products that satisfy the needs of this robot. Some of these are presented in the table shown below:

Table 8: Microphone and speaker device comparison

Adafruit Voice Bonnet	Seeedstudio Respeaker Mic Array V2.0	Seeedstudio Respeaker 2 mics Pi HAT
• **includes 2 microphones (L and R)** • **relatively cheap** • **2 1W speakers** • **3.5mm audio jack** • **I2C digital audio for both input and output** • **98dB SNR**	1. 4 high performance digital microphones 2. Includes speaker 3. Speech algorithms included on chip 4. 3.5mm audio jack 5. Hears voices up to 5m away. 6. Background noise cancellation 7. More expensive 8. 63dB SNR	• **Designed for AI and voice applications in mind** • **1W speaker** • **2 microphones (L and R)** • **Hears voices up to 3m away.** • **Cheapest** • **98dB SNR** • **3.5mm audio jack**

As can be seen from the table above (Table 8), each mic array board also contains a speaker, which is perfect for displaying output audio. This facilitates part selection, as individual speakers are no longer a necessity. Moreover, each of the boards contain an audio jack socket to connect another simple speaker should it be necessary. This allows improvement in the audio output quality if necessary. The audio jack is also potentially useful for connecting the device to the microcontroller that is used to control the inputs and outputs of the device. The Adafruit and Pi HAT use the same speaker codec. This gives them the same SNR which is higher that the expensive V2.0.

Observing the specifications and requirements for the robot, the team has decided on adopting the Pi HAT as the board of preference. This uses the same speaker codec as the cheap Adafruit Voice Bonnet and include a support for farther distance. The expensive 4 mic V2.0 is better in the sense that it includes noise cancellation and better voice detection. However, the price is too large for it to compare against the Pi HAT. The Pi HAT also contains all the components and specifications that make integrating it to the overall project simple. The audio jack is useful to easily connect it to the Raspberry Pi or to add better speakers if required. Furthermore, there are multiple options in purchasing the Pi HAT, which almost guarantees the team finding one of these components available for purchase.

3.3.2 Touch Sensor

Nowadays, there are several effective technologies for touch sensors. The objective of the touch sensor used in the robot dog is to detect when the robot dog is petted and send commands to the microcontroller. The sensor must have low power consumption, be responsive to strokes, and should penetrate the plastic casing. Furthermore, to keep simplicity in the design, the sensor should ideally work with 5 V input voltage. The different options can be divided in resistive sensors, capacitive sensors, Surface Acoustic Wave (SAW) sensors, and Infrared (IR) sensors. Each sensor provides its own advantages and disadvantages that are discussed below. Next, a table is shown summarizing the advantages and disadvantages of each sensor. Finally, the sensor is chosen and the explanation of why it was chosen is also be presented.

Resistive touch Sensor:

A resistive touch sensor is basically two metal plates separated by a narrow gap. When the user presses on the sensor, the two plates make contact resulting in current flow. To determine position, first one plate is charge and used to determine one axis, and then the other plate is charged to determine the other axis. This operation is done in a matter of milliseconds. For touch screens, the top layer is covered with a thin film and the bottom layer contains the glass. This technology does not provide the best touch quality and is dependent on pressure. Depending on the sensor quality, and sensitivity, a simple tap does not activate the sensor. This technology is cheap and has low power consumption. Furthermore, it is very durable and work with any type of touch; it does not need to be a bare finger.

Surface Capacitive Sensor:

Surface capacitive sensors work by placing a transparent electrode layer on the surface of the sensor. When a human body touches the surface, a small electrical charge transfers from the screen to the user. This creates a decrease in capacitance which is detected by multiple sensors located on the corners of the surface. These sensors can then pinpoint the distance and the location of where the user touched the device. Because the touch needs to draw a small charge, this technology only works with the bare finger or specialized capacitive styluses. The technology has very high scratch resistance and has outstanding resistance against contaminants (such as dirt, oil grease, etc.) and liquids.

Projected Capacitive Sensor:

Unlike the surface capacitive sensor, this sensor can be used thin cotton gloves or surgical gloves. Furthermore, this sensor also can detect multiple individual touches (fingers). These sensors are created by using sheet of glass that is embedded with electrode films. The sensor also contains an IC chip, which creates an electrostatic field covering all three dimensions. The chip is responsible for detecting the ratios of the multiple changes in capacitance and determine where the touch is located on the screen. A general schematic of the sensor can be seen in the image below (Figure 14). Like the surface capacitive sensor, the sensor detects the change in capacitance when charge is absorbed as the finger touched the electrode patten later. From right to left, the diagram shows for components of the sensor: the glass panel protecting the electronic components, the resistive coating to detect touch, the micro insulators to separate capacitive plates, and the conductive membrane detecting and locating where the touch occurred. The protective cover is designed to have little interference with the interaction while also protecting the fragile electrode pattern later.

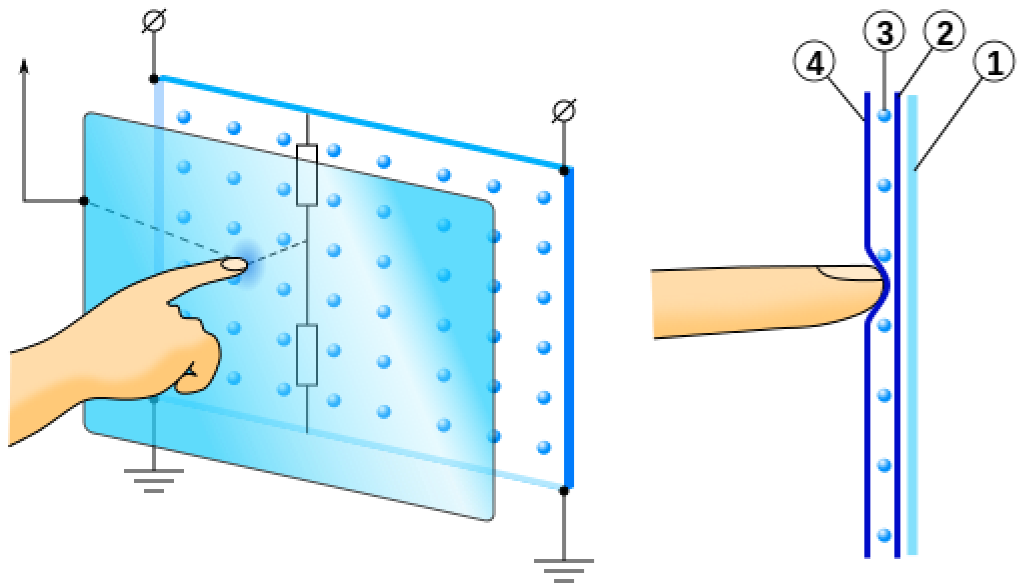

Figure 14: Projected Capacitive Sensor from Wikimedia Commons by CC BY 3.0

Since the chip is performing the calculations, it is possible to use two or more fingers on the sensor. The chip is capable and responsible of determining the locations and sources of the different signals. This technology has the same advantages and disadvantages of the surface capacitive sensor. The only downside to its cousin is that this sensor is slightly more expensive.

Surface Acoustic Wave (SAW) Sensor:

As the name suggests, the SAW sensor monitors several piezoelectric transducers and receivers located on the glass plate surface. This creates an invisible array of ultrasonic waves on the surface. When the screen is touched, the sensors detect the portion of the wave being absorbed by the touch. With this information, the transducers located on the glass can determine the X and Y coordinates of the touch location. This technology works with soft-tipped styluses, gloves, and fingers. It does not work with items such as pens or fingernails. The technology is not very common with small object and is hard to find. In fact, due to the recent pandemic and manufacturability constraints, these sensors have become fairly expensive and are very hard to come by.

Infrared (IR) sensor:

Unlike any of the other sensors, a special surface is not needed for this sensor to work. It only needs IR emitters and receivers to create an invisible grid of light beams on top of the surface. When an object interrupts the light stream, the sensors pick up the anomaly and pinpoint where it comes from. This tend to be more expensive that the other sensors. They can also trigger false inputs since the light is above the screen. Additionally, it can be sensitive to ambient lighting, water, and dirt. These are best used as motion sensors rather than distance sensors. They are considered distance sensers, however, because they can detect the distance an object is just like any of the other sensors IR sensors do have advantages as well. These sensors are long-lasting and insensitive to pressure. Like the SAW sensor, these sensors can be very expensive and recent constraints have made them difficult to find in small quantities. A summary of each of the sensors is shown in the table below.

Table 9: Main types of touch sensors

Resistive touch sensor (FSR Interlink 406)	Surface Capacitive touch sensor (AT42QT1012)	Projected Capacitive touch sensor (TPS65-201A-S)	SAW touch sensor	IR touch sensor (NNAMC1580PC01)
• **~$10.00 (40 x 80 mm)** • **~0.2-2 N of force** • **Durable (datasheets claim can support up to 2.5kg for 24 hours)** • **No multi-touch technology** • **Pressure dependent** • **Operating voltage (1.8 – 5V)**	• No Multi-touch technology • Bare finger required • ~$6.00 (20 x 30 mm) • I2C • (1.8 to 5V)	• Multi-touch technology • Finger, cotton, or surgical gloves • I2C • Operating voltage: 1.65 to 3.6 V) • ~$6.00 (40 x 40 mm)	• Works with any soft-tipped surface • Long-lasting • Better scratch resistance than capacitive or resistive	• Expensive: ~$100.00 (158 x 134 mm) • Weak to water, dirt, etc. • Can trigger false inputs • Long-lasting

After consideration, it would be ideal if the robot could respond to any touch, regardless of if it is a finger or not. The dog should identify when it is petted and simply "touched." The sensors must be located under the main body of the dog (hidden). Like a regular dog, the robot should detect anything that touches it. This discards any sensor that requires finger touch. These sensors would be the capacitive sensors and the SAW sensor. IR sensors are too expensive and not the best when placed under the surface. It is expected for the robot surface to bend or deform after being constantly touched. With the IR sensor, there is a potential of the bent body giving false reading in the long run.

Taking all this information, the team has decided on using resistive touch sensors. The sensors are easy to find and can be bought in bulk (useful for testing and expanding if needed), do not require much power, durable, and can be placed below the robot's surface. It was important to make sure the sensor can receive enough pressure to receive the touch. Moreover, these sensors are made to look like pads, which makes them easier to hide should they need to be placed on the surface of the dog. Although multi-touch would be useful and could add more functionality to the robot, they are not required for the basic design in mind. Moreover, the top layer of the sensor can be covered, if necessary, since the sensor does not need to be directly touched; the sensor only needs enough pressure to connect both plates and send a signal.

Comparing a few of the specifications presented in table 11, it is clear that when it comes to power consumption it does not matter which sensor is used. Both the capacitive and resistive sensors consume around the same amount of energy. Furthermore, the IR sensor is not very common and the smallest size currently available is too large for its purpose on this project, making it unusable. This is also the case for SAW touch sensors; these sensors are not very common and are hard to find. When comparing the price between the resistive and capacitive sensors, there is very little difference. However, the resistive sensor is available in all shapes and sizes and are easy to find. They are also simple to set up and not as sensitive as the capacitive touch sensors.

If the resistive touch sensor does not work under the body of the dog, then the capacitive sensor is the second-best option. The sensor is supposed to be placed as the surface of the dog. This is not ideal, but this option allows the dog to have other functions with the touch sensor, specifically the multi touch capability. The multi-touch allows for different commands to be given to the dog based on specific touch feedback.

3.3.3 Ultrasonic Sensor vs. IR Sensor

For the facial recognition to work properly, the camera also requires at least one additional sensor to give it the ability to perceive depth. There are many sensors that can help accomplish this. The two main sensors for detecting distance are the ultrasonic sensor and the infrared (IR) sensor. Each technology has its different way of perceiving distance and is discussed below. Moreover, each sensor has its own advantages and disadvantages that make them unique and useful for specific scenarios. There are other technologies such as radar and Lidar, but these are too advanced and complicated for the simple function that the sensor is used for.

Ultrasonic sensor:

Ultrasonic sensors, like the one shown in figure 15, are special sensors that work by emitting a burst of sound waves at high frequency. These sound waves travel at the known speed of sound and reach a target. When the target is reached, the waves bounce back, and the sensor uses the time to determine the distance. These sensors are relatively simple. The main drawback of these sensors is that they need a clear line of sight to the target. Any physical obstruction makes the sensor provide an incorrect reading. Moreover, the surface should ideally be flat, so the waves can bounce accurately to the sensor and the sensor can give correct results.

The image demonstrates that these sensors are small and do not occupy much space. As mentioned, the ultrasonic sensor needs to be clear of obstruction, so the emitter and receiver need to be visible from the outside. In most robots trying to imitate living creatures, the ultrasonic sensor is used as a pair of eyes. For our own project, the robot was planned to have animated eyes so the ultrasonic sensor cannot be used to appear as eyes. Hence, should the team use an ultrasonic sensor, it is ideal that the sensor is small and can be easily hidden. Fortunately, the burst of sounds waves produced by this sensor are invisible. This means they do not cause any hindrance in the appearance of the robot.

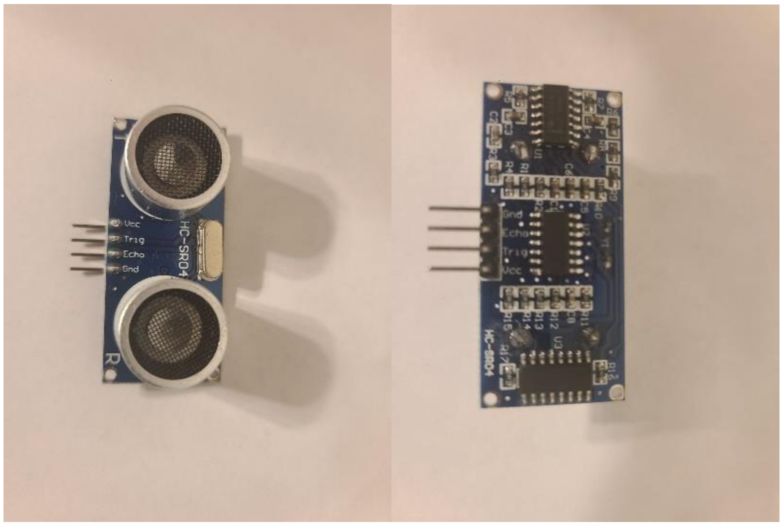

Figure 15: Ultrasonic Sensor

The figure above (Figure 15) shows a typical HCSR04 ultrasonic sensor. The important component to observe are the two round components ultrasonic sensor has. One of them is a speaker that emits the ultrasonic wave that is later detected by the other round component. Furthermore, we can see that a few IC chips and other small components are required to make the sensor function properly. One of these components is the crystal located in between the large round components. The crystal helps the sensor time the sound wave bursts. This is important to note since it means if this component is selected, the team would not have to worry in purchasing the correct frequency oscillator for the component. The team also avoids the fairly common problem of oscillator being too sensitive and causing issues.

IR Sensor:

Infrared sensors send a small infrared light beam. As with the ultrasonic sensor, the infrared sensor detects when the light path has been obstructed and uses time to determine the distance. One diode is set to emit the light while the other reads the output. The drawback of using this sensor is that they are heavily affected by the lighting in the environment. This gives it very little range and could only compete with the ultrasonic sensor is dark environment. Furthermore, the technology can be fairly expensive and not easily available.

These sensors come in various sizes and light intensity. The more powerful the light, the farther the sensor can detect distances, yet the more expensive and bigger the sensors tend to be. These sensors are more commonly used as motion detectors in doors or windows since they only require measuring up to length of a door. Even so, powerful enough sensors can reach very far and compete with the ultrasonic sensor. Moreover, these sensors are technically distance sensors, despite the short distance since they calculate distance. In fact, as mentioned previously, they use the same method of calculating distance as the ultrasonic sensor and other similar distance sensors.

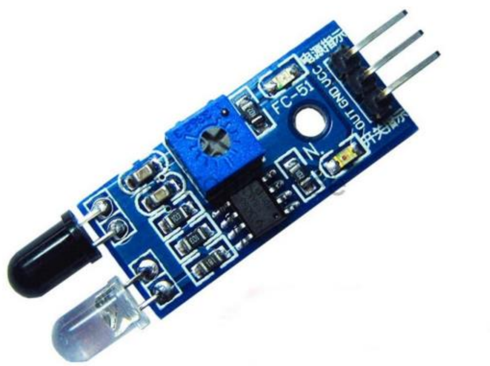

Figure 16: IR Sensor

The IR sensor demonstrated above (Figure 16) shows that these sensors can be made small. There are also larger versions that provide more accurate readings or can travel much farther distances. These sensors work like an ultrasonic sensor. They use one light to emit an infrared beam, the beam then bounces back and is receive by the other light source (receiver). This particular infrared sensor is small but can only detect up to 30cm, which is almost nothing compared to the long distance available by any ultrasonic sensor of the same side. This can be clearly seen in the table below.

Table 10: Comparing IR sensor to ultrasonic sensor

Ultrasonic Sensor (HCSR04)	IR Sensor (FC-51)
• **2-400cm non-contact measurement** • **15-degree measuring angle** • **Accuracy within 3mm** • **5V supply and consumes 15mA** • **~$5.00** • **2-420cm distance detection**	• 100-550 cm non-contact measurement • 4.5-5.5V and consumes ~30mA • ~$30.00 • 2-500cm distance detection

Viewing the comparison of both sensors, the team has decided on using ultrasonic sensors. The sensors need to work reliably with the camera. If the camera sees an image, we want the sensors to help the software determine the depth of each piece of the image. This would be very difficult with an IR sensor if lighting constantly changes the results. The robot should perform in any lighting; there is no constraint in the environment the robot functions. Hence, the IR sensor might prove unreliable. There is also the issue that the IR sensor can only detect really short distances (when comparing sensors of equal price). A more expensive IR sensor would have to be used to compete with the range of the ultrasonic sensor. An IR sensor like this has been found to cost (as seen in Table #) around $30.00. This price much is too much when the ultrasonic sensor can do the same for much less. The ultrasonic sensor does a much better job of identifying the depth of multiple components.

3.3.4 Servo Motors vs Stepper Motors

The robot dog requires eight motors for movement and mobility two in each leg one in each shoulder and one in each elbow. Furthermore, the robot is designed to not be heavy and easy to carry. This reduced the load in the motor required. Moreover, the robot used multiple motors and a battery as a power source. We need to make sure the motors are strong enough to move the dog or parts, but also not use a lot of power. For all the motors used in this robot, it is necessary to be able to control the speed and accuracy of the motor. Furthermore, the load is constantly changing for the limbs, and feedback is required to ensure the robot reaches the correct angle and position every time. With all these specifications in mind, there are several motors that can be helpful for our purpose. DC motors are perfect for low power, which makes them ideal for our project. Within DC motors there are different types such as stepper motors and servo motors.

After consideration of stepper vs servo motors, the team decided that servo motors were ideal for all the limbs and head movement. When considering what motor to choose, they are several qualities to consider. These are the load measurements, the speed, the variation of load, the need for torque limiting or other special functions, and budget. Although they are slightly more complicated, the feedback system allows the robot to achieve different positions even with shifting weights. Moreover, servo motors allow the robot to respond quickly and not have slow movements. The robot dog must act like a dog; and therefore, should have decently quick movement. This is something that is not possible with a stepper motor.

Stepper Motor:

Stepper motors are special motors that can be set to multiple locations or steps. They have several windings on the interior (pole counts) which allow them to magnetically move between each of the poles. Stepper motors are motors with high torque at low speeds, but low torque at high speeds. This is an effect of the motor containing generally pole counts of 50 or more. Pole counts are the number of magnetic poles on the router. These poles are powered in order, so the rotor turns in a series of steps. Since there are so many poles, it appears as if the movement is continuous. The disadvantage of having many poles is that the motor is not capable of going very fast, since the poles need to be quickly switched and powered every time the motor rotates from them. Stepper motors are also run without feedback, so there is no need for an encoder.

Because of the high pole count, these motors tend to be slightly bigger than a servo motor (with has much lower pole count). They are also heavier. Furthermore, stepper motors have the major flaw of having no inherent feedback. This means that if a strong weight or another force untunes the motor, the motor has no way of returning to its original position. For example, let us assume there is motor set at a 90-degree rotation that is then pushed by a strong force by 3 degrees. The motor can be asked to move to the 130-degree location, but it is permanently offset by theses 3-degrees until some other component tells the sensor to correct itself. This fault is not a big when the motor is doing a repetitive task with no significant changes (like a printer). However, this lack of feedback can lead to issues when precise movement is required and weight on the motor is changing constantly.

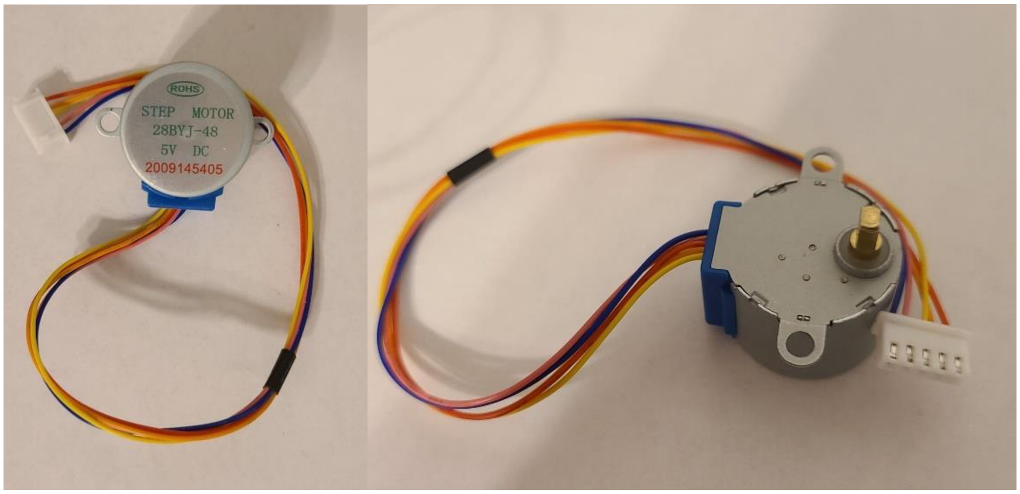

Figure 17: Stepper Motor

Figure 16 depicts a stepper motor. The servo motor has several coils on the inside. These coils the pole counts mentioned previously. A stepper motor has tens of these while a servo motor has much less. Having more pole counts allows the motor to move to precise locations but have less overall speed. Current is moved between each of the coils creating a magnetic field which force the rotor to move. The more coils allow for better precision but reduces the overall speed of rotation. The below image shows a set of servo and stepper motors which were tested to show how a servo vs stepper motor behaves between its torque capabilities, noise, weight, and generated heat.

Servo Motor:

There are many designs for servo motors. The main difference between these and stepper motors is they contain significantly less pole counts on the rotor. This allows the motor to rotate faster but have less starting torque. Even so, the low pole count also allows a feedback system to be implemented with the motor. Using an encoder, the motor can create high accuracy positioning and torque can be tuned accordingly.

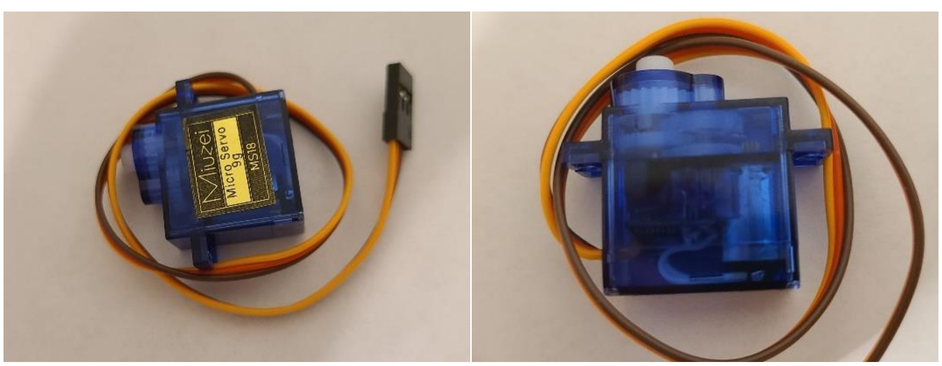

Figure 18: Servo Motor

Above is the classic SR90 9g servo motor. This is a potential contender to be used in the robot dog. From the image, it is visible that the servo motor contains gears at the top of the box. These gears are the gearbox of the servomotor used to control the speed and torque. The change of gears is done by software. Notice that the gears of the SG90 are made of plastic. This makes the motor cheap to make, but also unreliable under heavy loads, as the gears wear easily and can be damaged.

The inside of the blue box also contains the encoder. The encoder is what is used to read the information for the feedback system and determine how to set the gearbox. Most small drivers like these already include an encoder and gearbox so the user does not need to worry about these components.

Figure 19: Image of initial testing servo (left) vs stepper (right)

The following table (Table 10) shows the difference between servo motors and steppers motors. In this case, a simple stepper motor and servo motor are compared. The image of the motors compared is shown above (Figure 19). From the image, it is apparent that the motors are similar in size. The stepper motor on the right is slightly heavier and bulkier than the servo motor. It is also clear that both motors use the same number of cables to control the motors. As explained previously, the servo motor has a gearbox and an encoder located within the motor component, while the stepper motor is only the motor itself. Additionally, both motors have special holes so they can be easily screwed onto objects, signifying that the motors are means to be used in fixed structures should not be moving around, although it is a possibility. The important part to look at this table is the overall differences between these two motors.

The type of motor chosen for the robot dog limbs is the servo motor, for its speed and precision. There are some servo motors that already include the encodes and gearbox, making this not an issue for the overall assembly (as shown in previous images). Servo motors are ideal for robotic legs because of the constant shifts in weight and the advantage the feedback system provides. The weight shift can cause stepper motors to lose a step and they won't be able to recover on their own. Servo motors, alternatively, use a feedback system to assure they can accurately reach the required position, even with changes in speed and weight. Servo motors are slightly more expensive, but their size and weight make up for the small increase in price. Furthermore, this part is one of the most important for the robot, so a big part of the team's budget can be dedicated to choosing appropriate motors.

Table 11: Motor Comparison

Servo Motor	Stepper Motor
Operates on a closed loop (feedback)	Operates on an open loop
Has an internal feedback system which means less prone to error; can fix errors in positioning (improved accurate)	Has no feedback system which means more prone to error
About 1.2x more expensive than stepper motor of same size	0.8x cheaper than servo motor of same size and weight
Larger in size (due to encoder and gearbox)	Smaller in size (does not need encoder and gearbox)
Lower torque than stepper motor at low speeds	High torque at lower speeds than a servo motor of same torque values
Requires encoder and gearbox for accurate control	No encoder or gearbox required (less accurate)
Higher speeds achievable rather than a stepper motor of equal specifications	Lower speed possible than a servo motor with equal specifications
Tends to pulsate or vibrate in standstill position	No pulsation or vibration at standstill position (sturdier)

By now, it is obvious that servo motors are the parts selected to be used in the robot dog project. Knowing this, it is essential to compare the different types of servo motors available to determine which fit best for the robot dog. As mentioned in the table above, the biggest weakness of the servo motor is the possible pulsations and vibrations that occur when it is in standstill. This is due to the encoder and gearbox feedback system. Some servo motors include gearboxes and encoders while others require the user to buy them independently. For the robot dog, the servo motors do not need to be very big or hold too much weight. Therefore, the main comparison being used in the SG90 9g servo motor and the MG90S 9G servo motor. The MG90S is said to be an upgrade to the SG90. This does come with an increase of cost, however. There are very few differences between both servo motor models. The biggest difference being the metal gears used in the MG09S versus the plastic gears used on the regular SG90. The table below summarizes the main differences between both servo motors.

Table 12: Servo motor comparison

SG90 9g Servo Motor	MG90S 9g Servo motor
10 for ~$22	**8 for ~$26**
Plastic gears (more noise and less resistance)	**Metal gears (stronger than plastic and quieter)**
0.08 ± 0.01 (6V) seconds / 60 degrees	**0.11 seconds / 60 degrees (4.8V)**
Stall torque: 1.8 ± 0.2 kg*cm (4.8V)	**Stall torque: 2.2 kg*cm (4.8) V**
Stall current: 1300 ± 40 mA (4.8V)	**Stall current: 1300 ± 40 mA (4.8V**
Weight: 10g	**Weight 13.3g**

The MG90S is more expensive than the SG90, but the metal gears allow for better motor control. Plastic gears tend to oscillate more as they are not as rigid. Plastic geared motors are also designed to be as cheap as possible, so their performance is hindered. Moreover, the robot is expected to be heavier than 2kg, so MG90s servo motor would better. These are the reasons why the team has decided on using the MG90s as the servo motors for the robot dog movement. They are similar to their SG90 counterpart, but they are more durable, and the metal gears allow them to be even more responsive, without much vibration. If the robot proves to be too heavy for this type of servo motor, then a motor with more torque would've been used. This motor would follow the same description as the MG90s. In other words, the servo motor always has metal gears and runs best at around 5V. These two conditions are what make any motor better than the SG90; as well as the reasons why the SG90 was not chosen.

3.3.5 Battery Technology Selection

For any robot, some sort of power delivery subsystem is required. There are several different popular technologies to choose from. These include common primary cell alkaline batteries of various possible sizes such as AA or 9-volt, somewhat less common rechargeable formats made of materials such as nickel-cadmium or nickel-metal hydride (NiMH), and the almost ubiquitous rechargeable lithium-ion (li-ion) battery. The compact form factor of our robot eliminates lead-acid batteries and other large, heavy battery choices. The decision needed to be made between these three choices based on the requirements of the project. All three of these battery technologies have arguments for being easy to use, as they are all familiar and commonplace. For convenience as well as environmental concerns, rechargeability is desirable, eliminating primary cell alkaline batteries from the list of options. Given the requirements for stand-by and active battery life, battery size is an important factor to consider as well.

Table 13: Comparison matrix summary between battery technologies

	Lithium-ion	Lithium polymer	Nickel metal hydride
Cost (cheaper is better)	2	3	1
Weight (lighter is better)	2	1	3
Power (higher and faster output is better)	1	2	3
Durability (less volatile and more durable is better)	3	2	1

The most adaptable and configurable in different sizes out of all these options is lithium polymer (LiPo). The energy storage (battery) technology for the power delivery system of the robot are lithium polymer batteries. Lithium polymer batteries were chosen also because the Raspberry Pi, which is the computational center of our robot requires sufficient power input, easily supplied by a normal lithium polymer battery and accompanying circuit. There are several different standard sizes for lithium polymer batteries which are used in different applications. The sizes roughly correlate to different capacities but can vary within a range. Voltage and current output can also vary based on density or other characteristics. Size is an important consideration given the compact form we desired for the robot. Table 13 is a list of some of the smaller, more relevant to this experiment li-ion battery sizes and the physical dimensions they correspond to. The size name themselves have a format which gives the dimensions in the name, with the first two digits in the name being the diameter of the cylindrical battery cell and the second two digits being the length of the cell, both in millimeters.

Table 14: Li-ion battery cell sizes and capacities

Size Name	Physical Dimensions, diameter x length (mm)	Capacity (mAh)
10440	10 x 44	250 – 350
14500	14 x 50	700 – 1000
16340	16 x 34	400 – 900
18650	**18 x 65**	**1500 – 3500**
21700	21 x 70	3000 – 5000

These sizes refer to the size of individual cylindrical cells. However, like with any battery, they are typically bundled together to produce the actual battery pack for use of powering electronics. Lithium polymer battery packs also typically come shipped with a protection circuit due to the high current discharge which modern LiPo batteries are capable of outputting, especially without the proper protection limiting the discharge. These protection circuits necessarily effect the size of the battery cells, making them larger as the cell and circuit are all in the same packaging. There are also rectangular shaped lithium-ion batteries, such as those commonly found in portable electronics like cell phones.

To select the right type of battery, we need to consider the specifications of the main components. The Raspberry Pi 4 Model B specifications detail an operating voltage of 4.7 to 5.25 V or a target of 5 V, with a 5 A supply in, or 15 watts. A microcontroller such as a Texas Instruments (TI) MSP430FR6989 has a wide supply voltage range from 1.8 to 3.6 V as it is designed to be a low-power device, while an Atmega258P has a supply voltage of 5 V. The Atmega258P was used in this project, so one 2S battery was sufficient.

Lithium polymer (LiPo) batteries are popular for use in powering hobbyist electronic devices such as remote-controlled cars, toy planes, and drones. They output a nominal voltage of 3.7 V, which can be increased for the Raspberry Pi and Arduino 5 V supply inputs by combining them in series and using a step-down regulator or DC-to-DC convertor to output 5 V. They safely operate in a voltage range of 3.2 to 4.2 V. Some sources found suggested a minimum voltage of 3.0 V was operable, but we have chosen to assume 3.2 for an extra margin of safety. Going with LiPo batteries ensures a lot of flexibility, and they are lighter than comparable NiMH solutions.

Combining the lithium polymer batteries in series simply adds their voltage. So, two LiPo batteries in series results in a voltage range of 2 x 3.2 = 6.4 V minimum to 2 x 4.2 = 8.4 V maximum with a nominal voltage of 2 x 3.7 = 7.4 V nominal. This combination of individual units based on nominal voltage (labeled 1S) are used to denote LiPo voltages when selecting them. This labeling system is outlined further in table 14. Other specifications for LiPo batteries include capacity, measured in mAh and current discharge rating (shorthanded to "C") which determines the rate at which the battery can be safely discharged. The value of C is multiplied by the capacity to determine the maximum rate at which the battery can be discharged. A high C rating means that the battery can perform short bursts of a lot of power, which was occasionally necessary for this project. Therefore, we used a high capacity 2S battery with a high current discharge, to ensure ample voltage and current supply without having to recharge too often.

Table 15: Li-ion and LiPo battery configurations and associated voltage levels.

Cell Configuration	Minimum	Nominal	Maximum
1S	3.2 V	3.7 V	4.2 V
2S	**6.4 V**	**7.4 V**	**8.4 V**
3S	9.6 V	11.1 V	12.6 V

A 2-cell lithium polymer battery solution was chosen due to a convergence of different factors, including the availability of voltage regulators and battery chargers. While a 2-cell lithium-ion battery does make it largely incompatible with compact, plug-and-play, off-the-shelf battery chargers and protection boards, it does allow for the use of efficient buck convertors. Buck convertors were also the only kind of voltage regulator which were available (in-stock) during the design process of this robot. Voltage regulators and buck convertors were used in this project to reduce the ranging voltage of the 2-cell lithium polymer batteries to the design specified steady 5 V. Ultimately, a 2S 'shorty' LiPo battery with 5000 mAh capacity, typically used with RC cars, was chosen to be used in this project.

3.3.6 Battery Charger Selection

An off-the-shelf consumer battery charger was used in this project. The charger is capable of charging both 2S and 3S batteries. They connect with a standard JST-XH connector which is common in hobbyist LiPo and lithium-ion batteries. This battery charger was relatively cheap, well reviewed, and dependable, as well as easily purchasable in different models should that be necessary. This charger was decided on because of the importance of having a safe and reliable charging when using LiPo batteries, which can be dangerous when charged improperly. For this prototype stage, getting an RC car type battery charger was necessary because non-1S battery chargers are difficult to find in a breakout-board form. For this project, a custom battery charging circuit was originally designed, but after PCB integration, testing yielded undesirable results with a non-constant current output. As a constant and stable current output is the vital part of a charging circuit, it was unusable.

3.3.7 Voltage Regulator Selection

One of the most important selections for this project is the voltage regulator which was planned to be used at the center of the power supply circuit in this project. It is important that it is flexible, efficient, low-cost and has a small a footprint as possible while maintaining those other key factors. With a large assortment to choose from, with at least hundreds of relevant ICs available, the selection process must also be efficient to avoid unnecessary delay. The selection is narrowed greatly by the constraints placed on the design process due to current market and manufacturing conditions. Upon initial research, many convenient, preassembled solutions were found that only worked with a single cell lithium-ion battery but when it comes to custom designed power supply solutions, most high-quality, low temperature and high efficiency voltage regulators were unavailable. Table 16 shows some of the options which could be considered given a 2S li-ion battery. Only the TPS565208 was in stock at the time of the initial selection process.

Table 16: Select buck converter specification comparisons

Specifications	**TPS563240**	TPS565208	**TPS62903**
Type of Convertor	Buck	**Buck**	Buck
Maximum Output Current	3 A	**5 A**	3 A
Input Voltage Range	4.5 V to 17 V	**4.5 V to 17 V**	3 V to 17 V
Output Voltage Range	0.6 V to 7 V	**0.76 V to 7 V**	0.4 V to 5.5 V
Switching Frequency	1.4 MHz	**500 kHz**	1 MHz or 2.5 MHz
Junction temperature	-40 °C to 125 °C	**-40 °C to 125 °C**	-40 °C to 150 °C

One method of improving the selection process was using tools like Texas Instrument's web browser-based online power supply tool Webench®. This tool allows a user to input into an online form the minimum, maximum, and (optionally) nominal input voltage to the power supply and specify an output voltage and maximum output current. Using these user-specified parameters, the tool generates hundreds of possible circuits, which can then be sorted or eliminated based on criteria such as efficiency, bill of materials (BOM) cost, footprint, or the number of BOM parts. Previously found knowledge of Raspberry Pi and Atmega258P power specifications and a common configuration of LiPo batteries provides the necessary parameters for inputting into Webench®. Doing so and basing a final decision from prioritizing a combination of efficiency, cost, and footprint, yielded the design seen in Figure 21, centered around the TPS565208 4.5-V to 17-V input, 5-A synchronous step-down voltage regulator.

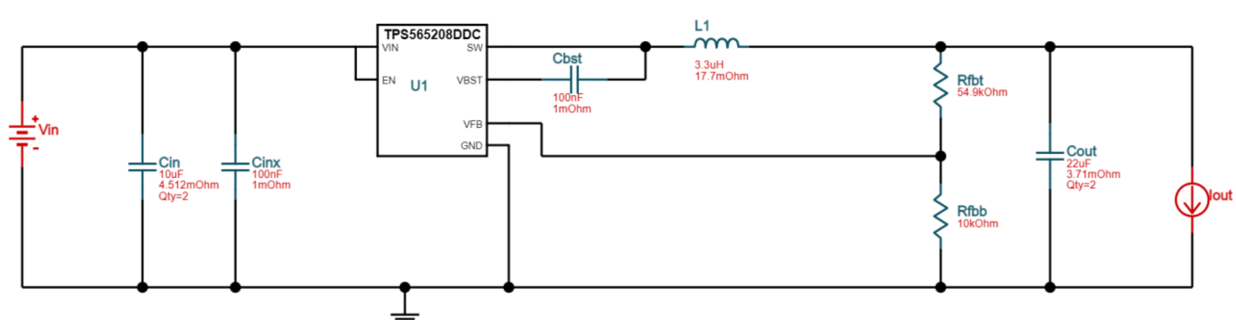

Figure 20: Proposed Raspberry Pi and Atmega328P power supply circuit

The TPS565208 voltage regulator had several benefits. It required a low number of BOM parts to operate which reduces the overall footprint and cost of the circuit, a top priority for a compact project. It was thought to be simple to implement. For a project on a short time scale such as this one, that is also very important given that it should theoretically be quickly deployable and one less system to troubleshoot once deployed. Another factor in the selection of this part which is not a technical parameter but is vital in the current environment is that the part is in stock and ready to order through consumer markets. Many of the voltage regulators and DC-to-DC convertor circuits considered were found to be out of stock or had parts within them be out of stock. Again, given the time frame of this project and current market plus manufacturing conditions, making sure that every part selected is ready-to-order is of the utmost priority. The lack of ready availability of voltage regulators is one area where these shortages are apparent and greatly influence possible selections.

While this initial design worked in the breadboard testing phase, it resulted in low voltage and PCB overheating issues. Due to a variety of time and monetary constraints, a totally new design had to be pivoted to. The STMicroelectronics LD1085V50 voltage regulator was chosen based on factors such as its low drop-off voltage as well as the low bill-of-materials cost along with a low amount of necessary additional components. This comparison table shows three different voltage regulators that were considered. As they all have fairly similar specifications, what it ultimately came down to was the maximum output current. The high output current of our chosen regulator allows it to have the low drop off voltage even at high current output which is vital for our design.

Table 17: Select voltage regulator specification comparisons

	LM78M05CH	**LD1085V50**	**L7805CV**
Max Input Voltage	35 V	**30 V**	35 V
Max Output Current	0.5 A	**3 A**	1.5 A
Voltage output	5 V	**5 V**	5 V

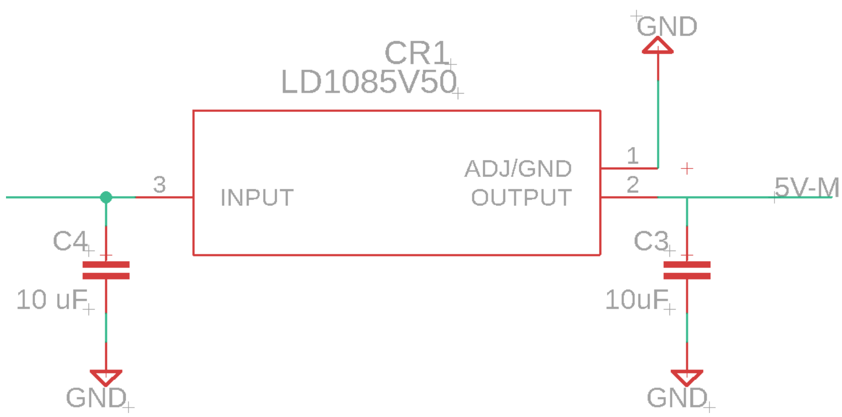

Figure 21: Final implemented motor and MCU power supply circuit module.

3.3.8 Mobility Selection

In the design of the robot, there was a point in which the robot would use wheels. The different wheels considered are shown below. The advantages and disadvantages are shown in this section as well is the most controversial among the group and the decision was made after careful consideration of all the information available. The method of mobility is crucial to any free moving robot and affects many other factors in the design process, so this level of consideration and discussion was necessary.

Wheels:

One of the main objectives of this robot is to act and resemble a dog. To do this, it would make sense that the robot be a quadruped robot with articulating legs, allowing for movement. However, the option of using wheels for movement is available and has its own benefits. Firstly, the legs do not need to be programmed to articulate in specific positions for movement. Instead, a motor needs to simply me activated to move the robot in a desired direction, like a four-wheeled vehicle. Should there be enough time, the robot could be designed to have the quadruped articulation walking movement, apart from the regular wheel movement. Using continuous servo motors on the wheels. The robot could then walk and move like a vehicle congruently. This, of course, would be very complicated as the robot would need to have near perfect control of its limbs and wheels performance and location. For now, wheels are going to be the main method of movement and transportation.

With today’s technology and advancements, there are several types of wheels that can be used for this robot. It is important to choose wheels that can support the robot, allow it to move on most surfaces, and are not too big so to distract or stain the “doglike” appearance expected to achieve. Furthermore, the wheels would not have any rotating mechanisms such as axles. In other words. The feet would just constitute of a motor and a wheel. Controlling the direction of each of the wheels separately with a microcontroller would determine in what direction the robot moves and its speed. Size of the wheel also matters. The team does not expect the robot to be used outside, but rather in a home setting. Thus, the majority of surfaces the robot would encounter would be flat. Furthermore, the robot would have working limbs that can lift the wheel should it be necessary. Because of this, the wheels just need around an inch to support the weight and have the mobility required. The options for wheels are fixed wheels, orientable wheels, ball wheels, and omni wheels. The two categories that could be beneficial for our robot are the omni wheels and the fixed wheels.

For orientable wheels, there exists centered and off-centered orientable wheels. These wheels are connected by a fork on top of the wheel. Most orientable wheels have a swivel joint for 360-degree movement. Although these wheels can freely move, they cannot be directly connected to a motor and are very loose when not touching the ground. As examples, these wheels are used in the front of shopping carts. Ball wheels are generally used for stability. These wheels contain a spherical ball positioned on a holder. They cannot be driven and, as mentioned, are just used for balancing in flat surfaces. These and orientable wheels are not useful for our robot since the robot requires each of the legs to have a wheel that can be driven by a motor, something none of these wheels can accomplish.

Fixed wheels are the most common and simple wheels used in robot. They only have two degrees of freedom; they can only move front and back. These wheels are normally used to drive the robot as they can be directly connected to the motor. Omni wheels are perfect for multi-directional movement. These wheels can move in any direction with low resistance. The wheels contain smaller wheels which are perpendicular to each other, allowing for the free movement. These wheels can be used for both steering and driving the robot. The main disadvantage of these wheels is they are expensive and cannot be used for controlling position, as the wheels tend to slip.

Figure 22: Mecanum wheel reprinted with permission from Adafruit Industries by CC BY 2.0

Another less common type of wheel is the mecanum wheels (Figure 22). Mecanum wheels are special wheels that unlike directional wheels, can also move sideways. As can be seen from Figure #, the wheels have multiple rollers placed diagonally on the wheel. The position of the rollers allows the wheels work as regular wheels, but also move sideways if given the correct force. If on an axle or with directional control, the wheels can allow a vehicle to move any direction by rotating the wheels. The greatest disadvantage of these wheels is the high price and complicated design.

Table 18: Types of wheels pros and cons

Fixed Wheels	Mecanum Wheels	Omni Wheels	Ball Wheels	Orientable Wheels
• Forward and backward directions • Can be used to drive the device • Cheapest	• Move diagonally, forward, and backward • Can be used to turn the device • Can be used to drive the device • Most complicated and expensive	• Move in every direction • Can be used to drive the device • Can be used to turn the device	• Excellent for balancing on flat surfaces • Cannot drive the device	• Good for balancing the device • Can be used to turn the device • Cannot drive the device.

The robot has four legs, and each has a driving motor. This makes it unnecessary to have balancing wheels. Therefore, the wheels on Table 23 are not ideal for this robot. Although they give great mobility, they cannot be driven by a motor and tend to be bulkier (bigger) than the other types of wheels. Mecanum wheels are also not ideal because they are too expensive, and they multi-direction capability is not necessary. Having four driving wheels makes turning the robot easy. This is also the reason why omni wheels were not used. They are more expensive than the fixed wheels and do not provide any useful advantage. Fixed wheels seem to be the best option for this robot.

Quadrupedal locomotion and its advantages to wheels:

Another option for movement which is what we ended up doing is using all four legs for movement. There exist several theorems and designs on how to make this possible. The robot is relatively lightweight and small, making quadrupedal movement not too complicated. Since the robot is made to resemble a dog, it must include four articulable legs. These legs have at least two servo motors each with at least 180-degree articulation. Having balanced feet and correct movement allows the robot dog to walk. There are many advantages that come from using quadrupedal motion instead of wheels. Firstly, the robot is much more "dog-like" if it is capable to displace itself like a dog. Moreover, by not using wheels, the robot would look a lot more like a dog, which was one of the major objectives of this project. Second, it is much easier to balance a robot on four legs than on four wheels. If using wheels, the motors would have to be very precise and powerful enough to keep the robot static. There could also be issues if the robot were to run out of power and the motor could no longer control the wheel. There is a possibility of the robot not being capable of remaining static if the motors are not rigid enough. If instead of wheels legs are used, then the robot cannot move unless the legs are moved. This makes is easy to maintain it stationary.

Another advantage of quadrupedal motion is that the different articulation points necessary for the motion can also be used to make the robot pose in different positions. By incorporating these articulations, we can make the robot sit, lay down, and walk moving these articulations. Wheels make sitting down and laying down difficult because the robot would be constantly slipping. Consequently, if the robot needed to sit, stand up, and then walk, it would need some time to balance itself after standing before it would start walking. Using quadruped motion and the correct articulation movements, this sequence of actions can be done fluently. It is because of these and many more advantages that the team has decided to use quadruped motion over wheel mobility. Wheels provide easier movement and less motors than quadrupled motion. However, one of the most important objectives of this project is make the robot feel like a real dog. This task would become very difficult if the robot were moving like a vehicle instead of a pet.

3.3.9 Camera Part Selection

The camera is connected to the IC used in the Raspberry Pi. There are many options for cameras that with such a device. As a reminder, the camera was used for color detection and for the robot to understand where it is in relation to its environment. The camera does not need to be high quality or have an enormous amount of zoom. The camera just needs enough resolution to identify different objects in the environment. Also, the camera is used in conjunction with the software to help the AI identify its owner. Furthermore, the camera is planned to be located on the face of the dog. The camera cannot be blocked so it cannot be hidden. Therefore, a small camera is ideal as it helps hide it. The camera needs to be as invisible as possible so the robot can keep its doglike appearance. The different options of cameras researched are summarized in the table below.

Table 19: Different camera options available

V2-8 Megapixel (RPI-CAM-V2)	8 Megapixels USB Camera	Arducam 5MP Camera V1
• **8-megapixel native resolution** • **Capture video at 1080p30, 720p60, and 640x480p90 resolutions** • **No infrared filter** • **Field of view: 62.2x48.8 degrees** • **Capable of 3280x2464 static images** • **~$30.00**	• 8-megapixel resolution • Capture video at 640x480 resolution • Infrared filer: 650 +- 10 nm • Field of View: 70 degrees • ~10.00	• 5-megapixel resolution • Capture video at 1080p30, 720p60, and 480p90 resolutions • Contains Infrared filter • Field of view: 54x51 degrees • Capable of 2592x1944 static images • ~$15.00

As can be seen from the table above (Table 19), there is a wide variety of cameras available and useful for this project. The best of the selection above is the V2-8 Megapixel from Adafruit. This camera module is specifically designed for the Raspberry Pi. It is the one used in many official tutorials in their website and the one Raspberry Pi recommend. This camera is fairly expensive but have excellent resolution and works in both bright and dark lights. If the budget allows it, this is definitely the best camera for our project.

The second-best option is the Arducam with 5-megapixel resolution. This camera can reach the same video resolution as its expensive counterpart. Where is lacks is in the resolution of the static images, which is lower than the expensive counterpart as shown in the table. Furthermore, the Arducam does use an infrared filter, so it wouldn't work well in low light. This filter is common in low quality cameras or outdated cameras. Present technology has made this filter unnecessary, and the filtering is done with software or other less interfering technologies. The manufacturers also state that the camera wouldn't work in direct sunlight. When compared to the best option, the biggest drawback is the infrared filter. However, the camera is available at half the price which is a big plus. Another disadvantage this camera has is the manufacturer. Looking at images of the camera, it is clear that the manufacturer intended to make a very low-quality product. The overall quality of the camera is lackluster, and the reviews are not great.

A third potential option for a camera is the 8 Megapixels camera. This camera apparently works well for image processing and AI vision. This camera costs only a third of what the V2-8 costs but does not provide nearly as many specifications. The camera can only capture video up to 640. Moreover, 8MP is extremely poor for today's standards. This camera does have the greatest field of view, but most likely the depth is not great. Also, the increase in field of view is minimal when compared to the other two options presented. This camera is the worst out of the three. Despite being the cheapest, the difference in price is small enough to go for the better cameras.

From this comparison, the team has decided on using the 8 MP RPI-CAM-V2. This camera is the most expensive out of all of them, but the AI integration is an important part of the design, and it is important that it works properly. Furthermore, the price difference is also very minor compared to the larger components used for the robot dog, so price is a very small limiter in this scenario. The camera's field of vision is like all the others, so this makes little difference. The biggest advantage this camera has is that it avoids using an infrared sensor. This alone allows the camera to perform better in atmospheres with both bright lighting and dark lighting. Additionally, the camera was found at a cheaper price (~$20.00). The camera has enough capability to provide clear images to the microcontroller. Furthermore, combining the camera with the ultrasonic sensor, the software should be capable of identifying people and objects in its surroundings.

3.3.10 OLED vs. LCD Display

There are many different technologies when it comes to display resolution. For the small displays required for this project, there are two main competitors: LCD and OLED. There are other technologies, but they are either two old, new, or very hard to find in the size required for the robot dog. Both technologies are useful for this product, so some analysis was required to determine which one was best. Continuing, both technologies are explained, and their advantages and disadvantages are presented. By the end, an explanation is given on why one technology was chosen over the other.

LCD Display:

A liquid crystal display (LCD) works by having a liquid crystal material in between two sheets of glass with transparent electrodes. When no voltage is applied. The liquid molecules lay flat and parallel to the glass. When voltage is applied, the particles change direction and are now perpendicular to the glass surface. By varying the voltage, we can vary the orientation of the liquid crystal. The figure below (Figure 18) gives a general idea on how each of the parts are related. Impermeable state is the state in which the liquid is parallel to the glass. Permeable state is when the liquid crystal is perpendicular to the glass. This only works because of the predictive behavior of liquid crystals.

To explain the image, the liquid crystal is made from nematic molecules. Nematic molecules are molecules that have no positional order. This means they can be changed with sources such as voltage. Depending on how the electromagnetic forces affect the liquid crystal, the molecules position themselves in different positions. As mentioned previously, for LCD displays the most important of these are the permeable state position and the impermeable state position. Not shown in the image, but a backlight is required so that can pass through the liquid crystal and onto the filters that create the different pixels. In most display technologies, light is divided into three different colors using filters to create a pixel. Depending on the intensity of these colors (which is varied by changing the state of the liquid crystal) a display can display practically every color. Pixels are also used in the OLED technology presented below, although the filtering light adjustments with a different method and technology.

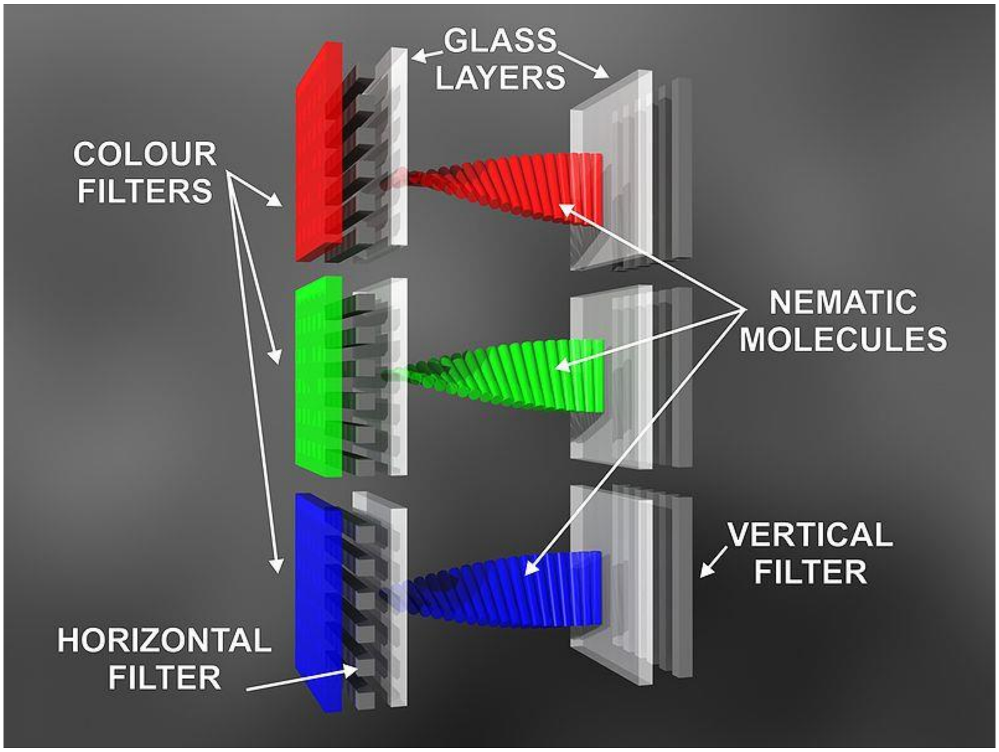

Figure 23: LCD display from Wikimedia Common by CC BY 2.5

The glass sheets (filters) in front of the backlight have horizontal slits while the other glass sheet has vertical slits. A white backlight passes through both glass and the liquid crystal, before arriving at a color filter. If there is no voltage app lied along the electrodes, then the light passes through the horizontal slits and through the liquid crystal, since the molecules are also aligned horizontal. This provides the maximum brightness. If voltage is applied, then the liquid crystal becomes perpendicular to the glass. The light passes through the horizontal slits, but the liquid crystal does not let it pass through the vertical slits, thus showing the least brightness. The color filters are used for creating the pixel.

This technology is the most common present technology when it comes to displays. It is not the newest, but it is available in multiple sizes and sold by several manufacturers. In small displays, the backlight brightness can be controlled affecting the overall brightness of the display. Since the technology is so common, it is very easy to find a screen with the correct resolution required for the robot dog. The disadvantage of using an LCD screen is that there are better technologies available and the backlighting much cause unnecessary power consumption.

OLED Screen:

Organic light emitting diode (OLED) displays are much newer than the LCD display. Eventually, these displays are expected to take over the current dominance of LCD displays. Unlike LCD displays, OLED technology does not require a backlight making them thinner, and more energy efficient. OELD displays use organic films that are placed between two conductors. When a voltage difference is passed through these organic films, light is produced. This light has much higher lumens that LCD displays. Additionally, OLED displays use much less energy since they do not have a backlight. The lack of a backlight also makes the black color of these screener darker.

Briefly, OLED is clearly the superior technology. Despite this, it is still somewhat not as widely available as LCD screens. If we find the perfect OLED screen for the robot, then there is no real reason to not pick OLED. IT is energy efficient, brighter, and flexible. However, when researching for multiple products available in the market, there is much bigger selection of LCD displays than OLED displays. Some LCD displays even claim to have the same performance as an OLED display. The abundance of options is what made the team decide on using an LCD display for the eyes of the robot.

3.3.11 Microcontrollers

The role of the microcontrollers in this project are split into two microcontrollers depending on processing power requirements. The Arduino being the lesser power controller handled much of the systems related to movement/servo motor control for the robot-dog’s unique animations and emotions as well as control of the ultrasonic sensor and its data. We decided to go with the latest Raspberry Pi being the Raspberry Pi 4 as it accomplished much of the advanced and computationally expensive features of this project. The Raspberry Pi was responsible for the robot-dog’s AI, facial recognition, pattern recognition, color recognition, object detection/avoidance, LCD display to the robot-dog’s eyes, and API integration with Amazon Alexa and Google Assistant.

Selecting the best microcontroller is important since they are the components that cost the most individually. These are also the components which affect the software design of the robot. They determined the language used by the software and the IDE used to program. Many sensors and other parts were selected so that they specially work with the chosen microcontroller. Hence, choosing the microcontroller also plays a role in the parts used for the overall robot dog design.

MSP430 Series

Despite this microcontroller not being included in the overall comparison table of MCU’s in Table 18, it is still an important microcontroller to consider. The microcontroller is not included since it was very quickly discarded. Simply put, the team has much better experience using the Arduino Uno microcontroller and there are several advantages that this microcontroller has over the MSP430 series.

Comparing the Arduino Uno to the MSP430, there are many similarities among both microcontrollers, especially when observing the multiple options of the MSP architecture. One big drawback of the MSP is that it runs from 1.8 to 3.6 V. The MSP320 IC can still be used by integrating a DC-to-DC convertor, but by using an Arduino Uno, with levels of 5V, we can avoid the use of convertors for many of the sensors. Furthermore, there is a lot more content and design ideas with the Arduino Uno community. The several advantages the Arduino Uno has are also reasons for choosing it over the MSP430.

ATMEL ATmega328P - Arduino Uno

The first choice for what microcontroller to use was the Arduino Uno. Overall, this seemed like the obvious choice as other microcontrollers with similar specifications such as the MSP430 would not have been as well suited for our design vision. The main reason for choosing the ATmega328P is its ease-of-use. The processor does not have a heavy setup process and its code base allows for much easier inputs to be programmed properly than most other boards. The ATmega328P additionally has much easier syntax and thus, allows for the coding/interfacing with many components to be facilitated in a much simpler fashion. Additionally, it runs at a 16 MHz processing speed with an operating voltage of 5V which is good enough for its use case in the project as well as having a low power consumption would keep it cool when within the chamber of other electronics. One negative to the Arduino is that it uses a language which is much less familiar to the programmer's aka C++ while the MSP430 uses C, a language that is taught in many classes in the curriculum. Arduino Uno also has a big community backing it. Solutions to potential problems that can arise in the robot dog design can be potentially found amongst the community.

The Arduino Uno is also open source. This means that several components and copies of the microcontroller have been made. This allows for great variety in prices and quality available. MSP, on the other hand, is only sold by Texas Instruments. This assures quality but heavily limits flexibility. This flexibility is another reason why the Arduino Uno is preferred to the MSP430. There is plenty of information on how the Arduino Uno works that it is very difficult to be lost with this device. There are also several tutorials to multiple sensors that use the Arduino as a microcontroller since it is probably the most common in the electronics DIY community.

Table 16 also shows how the Arduino Uno consumes very little current. This means that the Arduino was capable of handling all that it needs without providing a strain to the power circuits. The Arduino Uno also contains its own low power mode function. Understanding how this function works on the Arduino Uno helped the team implement onto the robot dog. Low power mode allows the robot to last longer, especially when it is mostly on standby.

A drawback presented on the table is the Arduino Uno does not have much memory. Evens so, for the functions required in this project, memory is not necessary. The code should be short enough to fit in the memory given in the Arduino Uno. The integrated circuit was only responsible for controlling the servo motors and the touch sensor. Additionally, the servo motors have their own driver circuit giving less strain to the microcontroller. This means that the low memory is not a real worry for the team and not a disadvantage to worry about.

Raspberry Pi 4

The second microcontroller we considered was the Raspberry Pi specifically, the Raspberry Pi 4. The consideration for this microcontroller was its familiarity with the programmers as well as the general DIY project community. With many projects and tutorials available the Raspberry Pi was a no brainer for its ease of use, its widespread popularity, its power, and its availability. The Raspberry Pi offers an MCU sufficiently fast enough to carry out the functions necessary such as AI, face detection, LCD control, pattern recognition, color recognition, object detection, and many more features we may wish to expand this project with into the future. Most important to our decision was its availability and its price. Currently, many microcontrollers and other silicon products are in low supply with high demand causing prices to be astronomical or nonexistent. The team managed to find an 8Gb model of the Pi available for MSRP and took the opportunity. The scarcity of devices is explained the constraint section of this document.

Looking at table 18, the Raspberry Pi already comes with a video decoder. This was useful for obtaining information from the camera that was used in the robot dog final design. The table also shows that this microcontroller is capable of wireless connectivity to the internet. This means that the device can be used wirelessly and can connect to the other devices the same way. This is helpful for the virtual assistant function that is desired for the robot dog.

There is a reason why the Raspberry Pi is so prevalent among the DIY project community. The microcontroller is designed to be powerful yet small. It is made to be easy to interface and program with, while also having enough complexity for complicated programs. Overall, the Raspberry Pi is one of the best and most powerful microcontrollers with its price range.

Jetson Nano

The third microcontroller we considered was the NVIDIA Jetson Nano. Many of the add-ons designed with the Raspberry Pi in mind also work well with the Jetson Nano. The Jetson Nano has many of its use in cases with GPU intensive tasks. Looking at the specifications in table 25, both the Jetson Nano and the Raspberry Pi are fairly similar. The biggest difference among both computers are the graphical processing units (GPU). Research has demonstrated that machine learning is better performed on a GPU. The Jetson Nano is also slightly more expensive than the Raspberry Pi. Another disadvantage of this computer is that the community around it is much smaller than the Raspberry Pi so finding helping projects and design ideas is more complicated.

From table 16, we can see many of the differences that are prevalent when comparing the Arduino Uno with a Raspberry Pi 4 and a Jetson Nano. When it comes to the operating voltage, all microcontrollers are capable of working with the 5V standard. Arduino does have a slight advantage that it can work with much lower voltages. This is due to the Arduino requiring less power overall to function. Next, temperature range is relatively the same for all boards and perfect for the function of this robot. There is an extremely low chance that the robot will ever reach sub-zero temperatures or above 80 degrees Celsius.

Continuing, the Arduino Uno has 6 PWM pins which were perfect for controlling the servo motors that were used for moving the robot. There are more motors than PWM pins. Luckily the Arduino Uno is also capable of simulating PWM on its digital pins. This is done with simple programming and is not very different from using pins designed for PWM control. To keep things homogenous, motors controlling the same functions are connected to the adapted digital pins to avoid any discrepancies that might cause an issue. Meanwhile, the Raspberry Pi 4 has a powerful CPU and large memory to perform the complicated calculations required for AI software. All boards are also fairly small and ideal for the small design planned for this robot. Notice that the Raspberry Pi consumes up to 4.0 watts idle. This is something that needs to be taking into heavy consideration when powering the robot.

Next, the table comparing the Arduino UNO, Raspberry Pi 4, and the Jetson Nano is displayed. As mentioned in the MSP 430 Series section, the microcontrollers are not mentioned since the Arduino UNO is more comfortable for the team and never have much of a consideration for the MSP 430. It had to be mentioned, however, since it is an important competitor of the Arduino UNO. There are multiple similarities between both microcontrollers and the MSP 430 would have worked just as well as the Arduino. This decision came mostly out of conformity and previous knowledge and experience the team had.

Table 20: MCU Comparison

Features	Arduino UNO	Raspberry Pi 4	Jetson Nano
Operating Voltage	1.8 – 5.5V	4.7 V - 5.25 V	4.75V - 5V
Operating Temperature Range	-40°C to 85°C	0°C to 80°C	-25°C to 80°C
GPU	N/A	Broadcom VideoCore VI	128-core NVIDIA Maxwell
CPU	ATmega328P	Broadcom BCM2711, Quad core Cortex-A72	Quad-core ARM A57
Max Clock Frequency	16 MHz	1.5GHz	1.43 GHz
Memory	32 KB	8GB LPDDR4-3200 SDRAM	4 Gb LPDDR4 25.6GB/s
Storage	N/A	Expandable microSD	Expandable microSD
Video Decoder	N/A	H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode) OpenGL ES 3.1, Vulkan 1.0	4Kp60 | 2x 4Kp30 | 8x 1080p30 | 18x 720p30 (H.264/H.265)
Connectivity	N/A	2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE Gigabit Ethernet	Gigabit Ethernet, M.2 Key E
USB	N/A	2 USB 3.0 ports; 2 USB 2.0 ports.	4x USB 3.0, 1x USB 2.0 Micro-B
Others	6 PWM Digital I/O Pins, 6 DC Current per I/O Pin 40 mA DC Current for 2.2V Pin 50mA Analog Input Pins	2-lane MIPI DSI display port 2-lane MIPI CSI camera port 4-pole stereo audio and composite video port	40-pin header (GPIO, I2C, I2S, SPI, UART) 12-pin header (Power and related signals, UART) 4-pin fan header
Bit-Count	8-bit	64-bit	64-bit
Power Usage	Active Mode: 200µA@1MHz Off Mode: 0.1µA	Idle: 3.8W to 4.0W Active Cores: 3.8W to 5.5W	Active Mode: 5W minimum
Dimensions	68.6 mm x 53.4 mm x	88 mm × 58 mm × 19.5 mm, 46 g	100 mm x 80 mm x 29 mm
Price	$22	$75	$99.00

Going through this table (Table 18), this project has complex needs and because of that we knew that we were going to go with two microcontrollers in order to service every feature we wish to implement properly and easily. The first MCU we decided on was the ATmega328P or the Arduino UNO. The main reason for this choice is its simplicity, cost, and ease-of-use make it a perfect candidate for this project. Other choices such as an MSP-based system would have too many features for what we need simpler microcontroller to handle. As far as the choice for the second MCU we knew we needed something more powerful and capable of handling intense graphical and computational workloads. Our requirements were 64-bit in order to use many modern software programs and algorithms. A large memory pool for fast, snappy response to input stimulus. A fast and cool operating processor that would be reliable while operating 24/7. Something easily expandable, easily worked on, easy to configure, and provides a smooth workflow with minimal unplanned issues. Most importantly however, was its price to performance ratio as well as its availability. As far as how the Nano and the Pi stack up the Raspberry Pi offers a later generation, faster, cooler CPU processor as well as a larger memory capacity. The Nano however offers a much faster GPU, but we feel that our project would not have a need for a GPU more powerful than what is in the Pi. Finally, the price to performance as well as the availability of the Raspberry Pi is greatly superior. Overall, both the Raspberry Pi 4 and the Jetson Nano could both complete the task, but in the end, we chose the ATmega328P and the Raspberry Pi.

3.4 Possible Architectures and Related Diagrams

In this section, the possible architectures are discussed. Architectures are the plans used for the structures of the project. Designs are the plans used to create the overall project. Hence, this section focuses on the architectures that defined the skeleton of the project. The design section of this document goes into more specifics on how each of the sensors were connected to each other. The first possible architecture quickly discarded involved the robot using wheels for mobility. The wheels would allow mobility to be easy to control and fast. This architecture was discarded because the wheels would make the dog lose its appearance. Moreover, the required design specifies that the robot dog moves like a dog and has articulating legs. If these are already to be included, it would be redundant to add extra motors just for easy movement with wheels. Hence, the architecture change and the robot used quadruped motion. This means that the robot was designed with the intent of having four legs that can bend around the half-way point. All four legs were connected to a main body which is then be connected to the head of the robot. This architecture follows the architecture of a dog and helped keep the dog-like appearance.

Another possible architecture involves the location of the microphones and speakers. The architecture planned involves the sensors to be located where the senses would normally be located in a dog. For example, the microphones should be located on the sides of the head to symbolize hearing from the ears. Likewise, the camera should be near the location where the eyes normally are on a dog. This architecture does lead to some problems, especially with the mic array selected. The main disadvantage of using this array is that the mics are fixed onto the PCB board they come in. To solve this, the PCB was adapted to separate the mics from the PC boards and allow them to be placed in the “ears” of the robot dog. This made the robot dog have a better appearance and ability to detect sound similar to the way a regular dog would. An architecture like this helped the robot keep the appearance desired.

An influence in the architecture of the dog is cable management. The robot dog has several cables that connect to the different components. Most components were connected form the limbs of the robot to its body. This means that the dog must be designed with the intention of hiding as many of these cables as possible. Furthermore, the cables must also be chosen to be thin and safe to use. Where each of the hardware components are stored is also a major influence in the overall design of the project. A third influence is the weight distribution. To simplify matters, it is ideal if the weight of the robot is equal throughout the entire body. This leads for an architecture where the head and the heaviest parts of the robot are separated. This is because the head tilts the center of gravity of the robot towards the front and weight is needed in the back to maintain the robot balanced.

To help design the project, the project is divided into multiple subsystems. These are the power subsystem, the back sensors, and the front sensors (sensors located in the head). Each of these subsystems are better explained in a later section of this document. These subsystems require a specific architecture to help accomplish their goals. The power subsystem has a very simple architecture, designed to hold very large and heavy objects that were used for the power. Most of the architecture of this subsystem is located in the body of the dog. This means that the body needs to be sturdy while also being large enough to hold all the components. To solve this, a large rectangular box is the type of architecture that was utilized. Previewing the design of the project, this box eventually became the body of the dog.

The next subsystem that requires an architecture is all the sensors that are located in the head of the robot. These sensors include the camera, ultrasonic sensor, microphones, and speakers. As can be seen, there are several sensors located in the front. Moreover, the PCB of the microcontroller utilized to control these sensors also needs to be placed nearby to reduce the use of long cables. To solve this, a large head was required. This does cause issues with the weight distribution mentioned previously. It would be preferable it the PCB and any of the sensors that can be placed here are placed on the back of the head. This would help with the weight distribution of the robot. To summarize, the architecture of this subsystem involves a decent sized box which holds several sensors and has most of its weight on its back. In the design section, the box is described as having the shape of the head of a dog or something similar. There is also a possibility for the head to be part of the body, like many other robot dogs in the market. Deciding the best design is explained further on.

Finally, the third subsystem also needed an architecture. The third subsystem is mostly the motors of the robot. In other words, the architecture being describe here are the legs of the robot. As mentioned, it is ideal is the robot has four legs with no wheels and potential paws. Since the paws are not intended to have motors, the robot does not need to have paws; the robot only needs to be capable of balancing when the motors are shut off. Moving on, the legs needed to be created in a way that they hide the cables that run from the motors to the PCB controller located probably in the body of the dog. This means that the legs were essentially hollow. With hollow legs, the cables can be placed on the inside and be easily hidden in a cable channel. This does cause a major issue; the legs had to hold most of the weight of the robot. The legs needed to be sturdy enough to hold the entire robot while also being slightly hollow to hold all the cables. Furthermore, the legs were around the size of those of a dog. They are not very bulky or thin. The specifics of this are later explained in the design portion of this document.

3.5 Parts Selection Summary

The table (Table 19) summarizes all decision made during the research and part selection sections. It considers the research, possible designs, constraints, goals, and objectives of the project. It also considers the standards and constraints given in this project. These are better explained in future parts of this documents. The parts mentioned in the table below are the ones that were used for the final prototype. The table also includes a summarized description of why each part was chosen.

The Pi HAT mic array was chosen for several reasons. One reason is that it is designed to be used with the Raspberry Pi. Another reason is that it includes two microphones. These microphones were planned to be placed on the ears of the dog; therefore, two were planned to be needed (one for each ear). The mic array also includes speakers so that part it no longer required separately. Ultimately, only one microphone was used. The FSR sensor was chosen because it is simple to use and fits perfectly with the use it is intended to have with the robot dog. The sensor is small, but many can be bought to increase the size. The sensor is also easy to find and overall better than the other sensors. Ultrasonic sensor was chosen because it is easy to use and many in the team already have experience with this sensor. Furthermore, it is much cheaper than the IR sensor counterpart. Li-ion batteries are used because they fit the constraints and standards of the robot. They are compact, energy dense, and the type selected has a high capacity. Furthermore, they are highly available and some of the most common batteries. The voltage regulator selected was chosen because it is easy to use, has low BOM, and provides good efficiency in the design of the robot dog.

Quadrupedal motion was chosen instead of wheels because it allows the robot to use less motors, proving for lower power consumption. Moreover, the leg articulation motors allowed the robot to achieve different positions which help with the goal of making the robot look like a dog. It was not the easiest objective to satisfy, but the team was able to reach a satisfactory goal with the number of resources available. The LCD display is used instead of OLED because they are more widely available. LCD displays are cheaper, available in multiple sizes, and are almost the current standard when it comes to choosing a display. OLED is a relatively new technology and is still not available in many shapes and sizes. There is also a lot more software drivers and programs already designed with LCD displays in mind. Finally, IC chips of the Arduino Uno and the Raspberry Pi are chosen. The Arduino Uno is used instead of the MSP430 series because of its ease of use and the experience the team has with it. Moreover, the Arduino Uno has a bigger community which leads to finding solutions to many common issues a simple task. The Raspberry Pi is chosen over the Jetson Nano because of the multiple devices that are created to work with the Raspberry Pi. The wide adoption of the Raspberry Pi in hobbyist as well as professional applications has made it a very accessible and well documented computer, facilitating ease of use, and improving the quality of projects that use it as long as the appropriate resources are utilized. It is also slightly cheaper than the Nvidia counterpart which is important for the monetary constraints we have in place for this project. More detailed explanations are provided throughout this entire section.

Table 21: Part Selection Summary

Part	Description
Seeedstudio Respeaker 2 mics Pi HAT Microphone Array	• Designed to work with Raspberry PI • Includes 2 microphones which is ideal for the design of this robot • Includes speakers
Square Force- Sensitive Resistor (FSR) – Interlink 406	• Can be used under the plastic casing • Can be activated without the need of bare finger • Widely available in all sorts of sizes
Ultrasonic Sensor	• Capability of detecting large objects and a reasonable distance • Good companion to the camera used on the robot
Servo motors	• Selected for their accuracy, speed, and control. • The feedback system allows for precise joint movements
LiPo Batteries	• Widely used and compact • Can easily provide enough power for the entire robot
Voltage regulator	• Easy-to-use and implement with low number of BOM parts.
Quadrupedal locomotion	• Made the robot look very doglike in appearance and movement. • The robot is also capable of setting itself in different positions with articulations in the legs
8MP camera by Adafruit	• Provides the best resolution.
LCD Display	• Preferred due to its price and wide availability • Has good quality for its required purpose
Atmega328P with Raspberry Pi	• The Atmega328P controls all the servo motors and other sensors not used for AI software. • A stronger CPU is needed for this and so the Raspberry Pi was also be used for this purpose.

4. Related Standards and Realistic Design Constraints

This section goes into detail of all the related standards and constraints that affect the robot dog. This has major effects on how multiple parts are selected and how the overall design of the robot dog is created. Following these is crucial for the robot dog to work successfully and also makes expanding and improving the design easier. Standards and constraints are essential to follow for any good engineering project.

4.1 Standards

Standards are essential for any engineering design. Standards are definitions and guidelines on how to design and manufacture different technical aspects. Standards help with safety, efficiency, productivity, and reliability. Some standards are very simple and can be simply explained in a few paragraphs while others might require several pages. It is important to note most standards are not forced by law. Standards are completely optional. However, standards are what allow people to use other designs and technology in their own projects. Without standards, everyone would have their own way of doing simple functions, leading to confusion and division in the field. Furthermore, costs can be heavily reduced if something is standardized since it is easy to reproduce.

There are many types of standards, and they can be divided in multiple ways. The American National Standard Institute (ANSI) divides standards into 5 major components: voluntary, de facto, consortia, regulatory, and corporate standards. As the name suggests, voluntary standards are standards that are never forced by law. These standards include product-based standards, management system standards, certification standards, and performance-based standards. Another type of standard is the de facto standard. These standards exist because they are simply widely used. They are not necessarily approved by a standards organization or government. An example of this standard is the QWERTY keyboard. There is no standard on how the keyboard letters should be placed, however many companies and manufactures determined that the QWERTY design was best, and it became a de facto standard.

Another type of standard is the consortia standard. Consortia standards encapsulate many different types of standards. Overall, these standards are the ones that limit participation. ANSI gives the example of a few companies requiring a membership so they can agree and work together. Regulatory standards are standard written and used by government agencies. Like any standard, these standard helps make government run efficiently and everyone using the same technical designs. Finally, there exist corporate or company standards. Like regulatory standards, these are used by specific companies to help organize themselves. Especially if a company is divided in several departments, standards help departments easily transfer information and resources to one another without having to deal with translating or simplify documents.

Standards are essential for safe, productive engineering. The robot dog design had to include many standards for every component of the robot to work together properly. There are standards for how the sensors communicate with the microcontrollers, how the batteries power the system, how the microcontrollers can be programmed to control the multiple peripherals, etc. These standards ensure that our robot can be easily replicated and can be easily created. Moreover, these standards help adhere to the safety and other constraints this robot is subject to.

4.1.1 Search of Standards

A cursory internet search yielded a lot of important information regarding relevant standards. As this project is a thorough integration of software and hardware, relevant standards cover many different areas. Among the standards found were standards regarding battery safety and the Universal Serial Bus (USB) standard, for example. These standards are useful for determining which technologies are used and how they are integrated onto different components. Some of these standards are not presented fully as the standards cover a wide range of users, including companies and manufacturers. There are several sites for searching for standards. Generally, these are the sites where the standard issuers are located. Some of the bigger ones are IEEE, ISO, and EIC, although many other sites exist. There are also some websites which try to encapsulate as many standards as possible.

Soldering Standard IPC J-STD-001H and J-STD-001ES addendum:

This standard is issued by the Institute of Printed Circuits (IPC) and defines the proper and safe way to solder electric and electronic components. This standard makes sure that all soldering in products has the best quality and reliability. This standard affects what materials are used, how the components are made, and how the PCB is created. The standard also specifies the space need in between solder and the amount of solder needed per pad. The standard also includes other industry related specifications such as how to perform visual inspections of the products and how clean the surface must be during mass production.

The addendum of the standard explains several requirements for the process of soldering. The standard explains corrosion, materials, flux, thermal protection, part mounting, exposed metal and damage, wire extensions, proper through-hole soldering, SMT lead forming, chip components, surface mount array packages, etc. All of these processes affect how to solder, what components are viable for soldering, and how the PCB board must be created to allow for proper soldering.

The first part of the standard specifies the materials that can be used for solder alloys and for the flux. It also details how to test the effectiveness of each material. The standard then goes into detail on how chemical strippers should not damage the components and how the components should have heat sinks to prevent damage from soldering. The section on thermal protection also goes into detail on suitable methods for mounting parts onto a PCB board in a way the components can be soldered without being damaged. Continuing, the standard then goes to specify the amount of exposed metal allowed and how this metal can be properly soldered. Finally, the standard ends by describing different soldering methods and how to properly clean the workspace for soldering.

Voice Commands Construction and Testing ISO/IEC 30122-1-2:2017:

This standard specifies how voice commands are to be made and tested. The standard is created by the International Organization of Standardization (ISO) in conjunction with the International Electrotechnical Commission (IEC). The standard establishes which are the default voice commands that are used for every language. It also briefly explains how voice recognition must have language tolerance (notice different accents or impairments when using a language). This standard is divided into several parts. The only parts that apply to the robot are the first two, which explain the general commands used in voice input, and how to create other commands. The standard also includes a few commands that could be easy to memorize and facilitate the use of information/communication technology (ICT) devices. Examples of these devices are tablets, mobile devices, navigation systems, etc.

Wi-Fi Standards IEEE 802.11:

Wi-Fi standards are set by the Institute of Electrical and Electronics Engineers (IEEE). These standards are made to ensure that all devices can connect to Wi-Fi properly and safely. The standard overlay the maximum speeds allowed for each frequency of Wi-Fi. Furthermore, the standard also define which frequency bands can be used for such technology. The standards also specify range, modulation technique, bandwidth, and many other descriptors necessary to utilize the Wi-Fi networks. The fastest commercial Wi-Fi available is the Wi-Fi 6 (802.11ax). this Wi-Fi standard uses 2.4, 5, and 6 GHz bands with speeds up to 9.6 Gbps. This Wi-Fi is still not very common and most use Wi-Fi 4 (802.11n) or Wi-Fi 5 (802.11ac). Specifications for these are also describe in the 802.11 standards.

Measuring methods and technologies of touch displays IEC 62908:

This standard defines the measuring conditions used for testing a touch display. This includes both displays and sensors. The International Electrotechnical Commission (IEC) created the standard IEC 62908 so specify what is considered a touch sensor. The standard begins by defining all possible technology permissible to be used as touch sensors and how they can be used. Section 12 of the standard proceeds to explain in detail how touch sensor sensitivity is measured and the ideal conditions to do. It goes on explaining who the touch sensor should be tested by providing different methods. The standard also gives a test bar touch sensors should pass to be considered useful. The standard includes many figures specifying how the measurements should be done. Overall, the standard not only

Universal Serial Bus (USB) Standard:

The USB standard covers the use and implementation of many different physical connections, communication protocols, and power delivery. The USB standard is maintained by the USB Implementers Forum (USB-IF). The USB standard defines socket design and pinouts on those sockets. It determines the compatibility of these different connectors and between different USB generations. The standard also specifies the speeds a USB port can handle and the different technologies available. USB is a very popular connector, and the standard covers the recommended usages for different USB socket types. The standard gives guidelines which allow for USB to be used for data transfer as well as power delivery, as a goal of USB is to eliminate the need for additional cables and connections. Currently, there are four major USB technologies: USB1, USB2, USB3, and USB4. USB4 is the newest technology, and it is now even a year old. The technology and standard created are so new that very few companies have had the opportunity to follow them. Most components, especially the ones used in this project use UCB1 or USB2. These are slower than the other technologies, but they are the most common, abundant, and easiest to manufacture.

Table 22: Summary of standards

Soldering standard	Voice Commands Construction and Testing Standard	Wi-Fi Standard	Touch Display Standard	USB Standard
• Issued by IPC. • Defines permissible soldering materials and quality of such materials. • Defines amount of solder and spacing needed per pad. • Constrains the design of the PCB board. • Explains soldering methods.	• Created by ISO in conjunction with IEC. • Specifies methods of creating new voice commands. • Specifies methods of testing new voice commands. • Requires language tolerance for proper voice recognition. • Provides easy to memorize commands.	• Set by IEEE. • Allow all devices to connect to the internet wirelessly properly and safely. • Sets standard frequency for Wi-Fi use. • Specifies modulator technique, bandwidth, and other descriptors. • Shows specifications for each commercially available Wi-Fi system.	• Created by IEC. • Defines what is considered a touch sensor. • Explains the technologies that are capable of producing a touch sensor • Explains methods of testing a touch sensor • Shows examples of how a touch sensor should function.	• Controlled and issued by the USB-IF. • Defines the electronic specification of a USB connection. • Assures that every USB connection released in the market works. • Provides testing specifications and procedures. • Provides guidelines on how to efficiently transfer data through a USB socket.

The summary of standards table (Table 20) shows all the key points in each standard. It only shows all the points that affect specifically our project. The previous explanations included detailed descriptions of each of the standards. This table should help visualize which part of the standards affect our project. Their specific impact is shown in greater detail in the next section. By observing the table, we can see that the soldering standard appears to be the most impactful out of all of them. The soldering standard affected every single component in the project, including our own custom components. These standards also help ensure that several constraints are satisfied especially when regarding safety and sustainability. The other standards still have an impact on the project, but the impact was not on all the components like it is with the soldering standard.

4.1.2 Design Impact of Relevant Standards

Each of the standards presented above had their own impacts on how parts were selected and the overall design of the robot. These standards helped create many systems of the robot and gave specifications on how these systems needed to be built. The standards also allow our design to interact with other smart devices, since they are following the same standards. Moreover, these relevant standards are also helpful in solving and satisfy many of the constraints that are also present in this robot dog project.

Soldering Standard IPC-J-STD-001H:

The standard has no impact on the exterior design of the robot, but it has a major impact on the PCB design and layout. The standard not only defines what materials should be used for the PCB solders pads, but also the materials allowed for soldering. Furthermore, the standard also describes what proper soldering looks like. When it comes to design, this standard mainly affected the size and quality of any PCB used in the project. It also affected how protection included with the components. The standard also impacted how much solder was to be used on the PCB and the overall its overall quality. Without this standard, there was a chance of receiving a PCB or a microcontroller that is not soldered properly. Some PCBs would also be smaller since they would not give much importance in the distance between soldering pads.

The standard affects almost every component used in this project. The components already built (such as sensors and the servo motors) followed this standard to ensure they were selling a reliable and sage product. These standards also affected the custom PCBs made by the team. The team had to solder their own parts and following this standard was entirely necessary to ensure that our custom PCBs have a similar quality, safety, and reliability compared to those manufactured by larger companies.

Voice Recognition ISO/IEC 30122-1-2:2017:

The standard has an effect on how new commands are to be designed. The robot dog uses a virtual assistant that already exists such as Amazon Alexa or Google Assistant. However, the robot has the capability to also respond to custom commands, as permitted by these standard and the API of the virtual assistant. Should these new commands be implemented, this standard helps determine the ideal phrase or word for each command considering different accents and dialects.

Wi-Fi Standards IEEE 802.11:

Determining which Wi-Fi network to use came down to how the standard defines each network and which is best for our robot. Our robot uses AI assistance. This means, the AI software had to be capable of connecting to the internet. The robot is also a mobile, so external cables are not preferred. Thus, the best way to connect the robot to the internet is via Wi-Fi. The 802.11 standards help determine which Wi-Fi standard is best for our robot and the working range the AI assistant has. The standard must comply to that it can connect with other devices on the internet using the same standards. The routers and modems used to connect to the internet are using this standard. Therefore, the robot dog software also had to use this standard because it needed to connect to the internet easily and securely.

Measuring methods of touch displays IEC 62908-12-20:2019:

This standard is useful when it comes to understanding which touch sensor to use in our design. By explaining the different technologies, the team is clear in knowing which touch sensor would be ideal for the project. Furthermore, the standard also explains how the touch sensor should be tested. It contains several methods and some of them are used to test the touch sensor of our design. These methods are later explained in the testing section of this document. This standard assures that the touch component used in our design is of high quality and tested properly. This led to a safer environment and one for less problems to occur. It also provided the team the ideal manner of testing the sensor ourselves to ensure that it works properly and to the ideals set by previous related standards.

Universal Serial Bus (USB) Standard

The USB standard in this project influenced data communication between different systems in the robot and power delivery. The battery charger input was planned to use the micro-USB connection, which in a future non-prototype revision contributes to the design goals of convenience and sustainability. The ubiquity of the USB standard, and micro-USB in particular means that in a final product, any common micro-USB ended wall charger could work to recharge the robot's batteries. USB can also be used to communicate between the more complex parts of the robot, such as the camera and Raspberry Pi, or the microphone and Raspberry Pi. Proper implementation of this standard is necessary to ensure a high-quality product which works as expected to the customer. It also helped the team by avoiding the need of creating custom ports. Most sensors and devices connecting to a PC use the USB standard to do so. By following the standard, we can guarantee that the robot dog is able to use these devices easily without much complication.

Battery Standards

The use of lithium-ion or lithium polymer batteries comes with it a wide range of standards, particularly to address safety concerns. These many standards need to be enumerated and considered during the design process. While the many standards should be kept in mind, many are guidelines for manufacturers and transporters of li-ion or LiPo batteries, which do not directly impact the design of this project. Two important standards which need to be verified for any batteries used in this project are laid out in table 18. Great care must be taken when designing any device that uses li-ion or LiPo batteries, as an unsafe battery or improperly designed circuit can cause bodily injury to a user.

Table 23: Battery standards

Standard Number	Description
ANSI C18.2M, Part 1	General specifications for rechargeable batteries. Written to achieve physical and electrical interchangeability and testing standards.
ANSI C18.2M, Part 2	Safety standards for rechargeable batteries. Written to ensure safety under normal use and potentially expected misuse.

4.2 Realistic Design Constraints

Design constraints conditions set by the customer, standard, or the engineer that need to occur so the project can be successful and work properly. Also, these constraints can be imposed by the environment and time period in which the project is constructed. They limit the different capabilities of the design and are strong determinants on how the overall design end up. They define what parts can be used, what parts are available, and how the design and part interconnect. The House of Quality shown in Figure 1 depicted many of the constraints set by the engineering and marketing requirements. The House of Quality gives a good idea on how these constraints can affect other aspects of the engineering process such as manipulating requirements and forcing adherence to particular standards. Most of these constraints cannot be avoided and must be dealt with when designing the project. As is explained further on, most of these constraints are related and affect each other. This generally means that adhering to one constraint also helps adhering to other constraints that are tied. A summary of the impact each of the general constraints has on the project is shown on Table # below.

The table shows the multiple constraints in the project and their impact on the design and creating of the robot dog. To better understand all these constraints, they have been divided in economical, time, ethical, health and safety, environmental, social, and political constraints. Economical constraints are constraints focused on the budget of the team and the price of resources at the time in which they are being purchased. Time constraints involve the time given to complete the project as well as the current time period of the project. Time constraints affect the technology available and the amount of complexity that can be put on the robot project. Ethical constraints limit the actions and data that can be given and read by the robot. There are several ethical constraints when working with robots and AI. Health and safety constraints ensure that the project is not harmful and is safe to use at any moment. Environmental constraints are complicated and can encompass several different factors. One of the biggest environmental constraints affecting this project is the Covid-19 pandemic. Social constraints are imposed by knowledge of certain vendors and engineers that could potential improve or detriment the project. Finally, political constraints limited the number of specific products available on the market and affect other constraints like the economical constraints.

As can be seen, there are many different sources that affect the overall design of the robot. Some constraints limit how the product function, some of them are imposed and had to be considered when creating the robot dog. These constraints are largely unavoidable, and the design of the robot dog needs to be implemented in a way that works around these limitations. Realizing what these constraints are before the robot dog is designed and created helped in making the first design of the project successful and with as few problems as possible. As is later explained, this is an important feature of the design process of the robot dog because of the limitations set the time constraints given. Constraints are not all negative. They allow the robot to be safely used by people. Furthermore, they allowed the robot dog project to also conform to specific standards. They also made it easier for the robot to be compatible with other components since these components also are following similar guidelines.

Table 24: Constraint impact on the robot dog design

Constraint	Impact
Economic	Budget only allows for specific parts to be used in the project even if better technologies are available.
Time	There is one semester to research the desired design of the project and another to create the design. This allows the team only enough time to test one design rather than testing and evaluating multiple designs. This constraint forces the team to work with the issues of the first design while also tackling only the larger problems.
Environmental	The time and location in which the project is made influences the technology and amount of time spent on the project.
Social	Knowledge of vendors, specialists, and reliable people help reduce the impact of the other constraints. Worldwide communication systems like the internet have made this constraint have a minimal effect on the robot.
Political	The country and its relations to other nations control the number of parts and resources available from other countries, affecting which parts are used on the robot dog.
Ethical	Concerns on how the robot listens for commands and the functionality of the AI forces the team to limit how well the robot perceives the outside world. The design is also affected to demonstrate when the robot is collecting information.
Health and Safety	Assure the robot is safe to use and will not cause any harm to its user. Affect how parts are housed and the shape and design of the robot.
Manufacturability	Limits the number of parts available and the time in which they are available. Huge factor in determining which parts are used in the final design. To make our project manufacturable, it must also adhere to several manufacturability-related standards.
Sustainability	Assures the materials used in the design are long-lasting. It also affects the technology used in the robot dog.

4.2.1 Economic and Time Constraints

Economic constraints hinder our ability to make a high-quality project. To prove this, it is better to observe the scenario in which economic constraints are not present. In a scenario like this, the robot dog would use the best sensors available and would overperform in all aspects. Economic constraints are what limit the robot from simply using better technology. The economic constraints in this project are a big influence in the parts used in the project. Since the project is small, there are several technologies available that could easily improve the efficiency and accuracy of our sensor. This, however, would lead to an increase in price above the economical budget constraint. Another economical constraint is that all the money used in this project comes from the team. Without any sponsors and the project not generating any monetary profit, it is unwise for the team to create a large budget for this project. This means that the team tried to create the best product possible with the lowest budget possible. In other words, the team wants to increase the economic constraint of the project, as long as it does not heavily affect performance, accuracy, and reliability of the robot dog.

The greatest factor in the economic constraint is the budget. Our upper limit budget goal for this project is around $500. As this is a prototype, the price to complete the project is significantly higher than what we could expect for a finished and mass-produced product. Using pre-existing and readily available parts would hopefully allow for an accessible and relatively low price in a product beyond the prototyping stage, but realistically for this product under our current economic and time constraints, that goal is not be fully realizable. Recently, several other constraints have been directly tied to the economic constraint of this project. For example, political and manufacturability constraints have led to increase in prices from some components, which affects which components are available due to the economic constraints. To summarize, economic constraints limit the quality of the design as well as the parts available.

Time constraints also play a big factor in the quality of our project. Having unlimited time, the team could constantly improve every little bug and problem that presents itself in the product. Nothing starts of perfect, so we expect some errors and issues in the design and creation of the robot dog. Time constraints only allow the team to tackle the larger problems. Furthermore, time constraints do not allow the team to experiment on different designs. The design decided on by the end of this document is likely the one to be used by the team. Even if the design has flaws, the best course of the team, given the time constraint, is to try to fix or circumvent the flaws present, rather than developing a new design. Overall, the quality and way problems are approached are affected by time constraints. Time constraints also affect which parts are used in the project. Because of other constraints, some parts are not currently available. The time constraints imposed in this project do not allows the team to wait until these parts are back in stock. Because of this, the project must use other parts that are planned to be used by the team. Showing another reason how time constraints affect the project of the robot dog.

We have a little over four months to build our robot. So, our final product is just be considered a prototype by the industry standards given the fact that there not enough time for excessive testing. The given time constraints made our team reframe the project as a prototype to showcase the possible functions and features of smart robot which could be further developed and built with a richer feature set given more time and economic resources. The constraints result not in simplifying the robot for the sake of it but in working around existing constraints so that the robot can be an effective demonstration of our broader vision. A longer time frame would allow for the implementation of many more features, and, in particular, making the features built into the robot more complex and intelligent. Things like facial recognition, recognizing people or facial expressions, gesture control, more intricate movements, broad object recognition, learning capabilities, and more were all discussed but ultimately shelved due to awareness of time constraints. Found a few weeks into project research, Stanford University's "Pupper" project is similar to the scope and vision of our project except for the time constraints, and that project took several years to reach the stage it is currently in.

4.2.2 Environmental, Social, and Political Constraints

Environmental constraints for this project include ensuring adequate battery longevity in terms of its lifetime, because lithium-ion batteries are major pollutants as e-waste. Designing the system around limiting waste restricts the number of possible designs to implement. In general, as with any system, we want to avoid redundancy which manifests as waste while still building a reliable and long-lasting system. Many toys or devices built for entertainment purposes are thrown away after a short amount of time and repurchased. Environmental constraints for the design of the robot dog to include parts that are not very harmful to the environment. This leads to a limitation on the available parts.

Environmental constraints can also be considered as the constraint imposed by the current environment in which the product is being developed. One example of this is the time period in which this model is created. Current technology allows our project to perform many of the functions and requirements needed with fairly simple devices and implementations. However, in some scenarios the team needs to adapt to the current technology and create other ways of performing certain tasks. It is possible that in the future there are machines or new sensors that that allow for even simpler implementations or more accurate readings. The project is limited to use the current technology available, which could be tied as an environmental constraint. Another environmental constraint of this manner is the location in which the project in constructed. The project is constructed with the resources given from the university by undergraduate students. The team did not use a large facility or complex laboratory to build the parts. The university does have several tools which make the design easier, but there are better tools available in other locations which were not used by the team. Hence, the location the team is using is also an environmental constraint.

A smaller environmental constraint is the weather. Being in Florida, should the device be used outdoors, it is expected for the components to heat up from the heat of the sun. The team needs to accommodate from this and consider this heat as a minor environmental constraint. Likewise, if the device would be used in a colder environment, there is a possibility that the device does not function as well as how it functions under the temperature it is tested at. These are constraints that must be considered by the team if the project is to be reliable in multiple environments. Another minor environmental constraint is the location of each team member during the day. Each person has their own life and has their own duties elsewhere from the university. This has an impact on the communication and time available for the project.

Social constraints are the constraints imposed by the connections the team has to outside resources and people. For example, if the team knew people that produced a specific part or where capable of creating custom parts, this would allow the team to create a much better design. Because the team has little knowledge of people in the field, it is difficult to find people that can help lessen the impact other constraints have on the design. Social constraints force the team to work with the resources they know and the people they know. These constraints are now so impactful nowadays thanks to the internet and other forms of worldwide communication. Even so, the project could be vastly improved if the team could talk to multiple experts in the field of robotic animals and anything else tied to this project. These experts could provide useful and insightful input on the design and components used in the project. These people would not only facilitate research, but they would also give potentially more valuable information than what can be obtained online.

Political constraints are heavily tied to manufacturability and economical constraints. These constraints are also tied to the location in which the product is being made. In this case, the political constraints are tied to the political tensions created by the United States government. Political tensions, especially with China, has led to many Chinese products to be either unavailable or out of stock. The United States' government has made cuts in the trade among the United States and China, which has caused a light shortage of Chinese products. Many components are made in China, as it is much cheaper to do so, but the political restraints have for the team to potentially look at other manufactures for the components. Moreover, the pandemic originating in China has led to many countries limiting their trade with the country, affecting the availability of many products through the entire world. This constraint does not heavily affect the design itself, but rather other constraints that have a bigger impact on the design.

4.2.3 Ethical, Health and Safety Constraints

There are a wide range of ethical, health and safety constraints to consider for this project. Ethically, we need to be aware of possible constraints regarding how personal and emotive the robot is. Regarding health and safety constraints, we need to be aware of any possible hazards that the robot can cause to its user. These constraints assure than any person using the robot fells safe and not be harmed in any manner by the device. The robot is already designed to be used by anyone, so these constraints are already integrated into the goals of the project. This means that these constraints are still present, but the team already considers these constraints when the main objectives are created. This section explains them to understand their impact and how they change the design of the project. These constraints are essential for the robot dog to be happily used by its users without them having a fear of being harmed, fear of being spied on by the camera and microphone, and many other fears.

While we seek to provide a full experience, we do not want to create a false or misleading sense of liveliness or intimacy. This constraint applies more to a more developed version of this product, and less so to the prototype which was produced by the end of the project period. Another way to think about this constraint is being sure to avoid the "uncanny valley" which some people experience when interacting with robots and cause discomfort. This constraint is a combination of ethical and social. Another constraint which is a combination of factors is privacy and intrusion. This is an ethical, social, and political constraint. The latter two have already been discussed. The ethical constraint revolves more around the expectation people have around in-home devices. If we were to gather data without disclosing doing so, then that would be strongly unethical behavior and constitutes another reason why we are not wade into that in the prototype stage.

The recent near ubiquity of smart devices and assistants has given rise to new criticism and concern in social and political spheres. Much conversation and controversy has surrounded the debated invasiveness and lack of transparency about the technology. As it necessarily resides in the home, often interacts with members of a family, and is constantly in the background, smart devices can ignite worries about companies being too integrated into people’s lives, often without letting people know what they're doing with their data, if anything. It is unlikely that our product would be exempt from this same scrutiny and criticism. Accordingly, certain constraints come into view. For this project, the only programming the prototype running is what is necessary to meet the requirements specifications. The robot collects no data and is not self-learning or -training. This is done to avoid any potential legal issues which could arise from including data collection as one of the robot’s abilities. Certain American states as well as political bodies like the European Union (EU), have strict laws to follow regarding data collection and user privacy, and must be navigated accordingly. An example of this is the EU’s General Data Protection Regulation (GDPR) which mandates a sweeping number of obligations and procedures that an organization must follow in order to collect or serve people with data within the EU. It purposefully avoids specifics to facilitate broad coverage and edge cases, and was designed to apply regardless of scale, covering anything from large multinational corporations, small to medium businesses and others. The GDPR law also applies to organizations outside of the European Union, so long as that organization processes any data from any EU citizen. As a preventative measure, the fines for violating the GDPR can be very high. The GDPR describes a set of 7 data protection principles which every organization must follow in their consideration of how they use and collect user data. This constraint is significant, considering the current popularity of using large data sets to create and improve artificial intelligence models. It limits the adaptability of our robot but keeps monetary and time costs down since our team does not have to wade through regulatory information. Furthermore, this constraint forces the robot to not record using the microphone unless it is explicitly clear it is doing so. The robot is designed so it can tell the user when it is recording audio and when it is not.

When it comes to the health and safety constraints, the team wants a wide audience of possible users, especially regarding age. This means the team must take child safety restrictions and constraints into account. The product must avoid potential safety hazards, to the greatest extent reasonably possible, such as shock and bodily injury. For this prototype, the safety constraints and considerations are not fully applicable, such as extensive testing of possible product misuse, but basic constraints must be observed. The lithium-ion battery and accompanying circuit fits under existing safety standards to ensure the safety of users, and care and possibly extra cost is paid to ensure that the power delivery system is hidden away in the body of the robot. The robot is not designed to be handled by small children, but as a precaution, most of the small parts of the robot are hard to access to avoid unnecessary injury. Furthermore, many components of the robot dog have failsafe features that prevents the components from causing serious harm should there be any problems in the circuitry or power usage. Additionally, the robot was designed to avoid any sharp edges. Every electric component in the design was also correctly grounded and isolated from the body of the robot. When working with electricity, it is important that there is no exposed cable or way in which the user can affect the circuitry. It is the health and safety constraints that force cables to have insulation around them. This demonstrates how these constraints not only affected the design of our device but also the components used within the device.

Safety and health constraints are in place to make sure that the product does not injure anyone. These constraints force the design to not be dangerous and not use dangerous amount of current or power. These constraints are important is the project is to be successful in a larger scale. After all, the robot dog is designed so people want to use it, rather than fear it. People need to feel safe around the robot dog. These constraints are not only something the team wants to adhere to, but they are also imposed and forced by multiple agencies and laws. Again, this is to ensure that the product does not cause any harm to people and only produces satisfaction.

4.2.4 Manufacturability and Sustainability Constraints

For the prototype to be produced in this project, manufacturability and sustainability constraints are different than if they were to be considered for a more developed product. For the prototype, manufacturability at any sort of scale or expedient time scale is virtually impossible. This constraint is particularly acute in the current moment while supply chain issues (such as shipping delays and resource bottlenecks) are still affecting manufacturing on a global scale because of the COVID-19 pandemic. Throughout the entire design process, there were many instances were the first components selected were “out of stock” or had lead times of months. The time constraint manufacturability constraint forced the team to look for other options. It is clear the Covid-19 pandemic is heavily influencing many of the constraints on this project and is, therefore, a major impact on the design and part selection of his project.

Our prototype is a very small-scale project and relies on external as well as somewhat internal manufacturing. We ordered many parts from various suppliers, which are very time-sensitive due to the current environment. If certain parts are unavailable, (i.e., out of stock or backordered to a time too far out for this project), then other parts were substituted to ensure manufacturability. This constraint forces our team to be more flexible but could also result in increased costs or manufacturing difficulties. Manufacturing constraints also affect the size of our robot’s chassis. To meet cost requirements as well as ensure manufacturability while minimizing delays and removing outside dependencies, the robot chassis was 3-D printed. 3D printing the robot ourselves allows the team to avoid manufacturing constraints provided by outside companies. Furthermore, the project being printed by the team influences the design of the robot. The robot needs to be small and transportable. The team would not be able to print a large body or something very hard to transport around, due to lack of resources. This lines up perfectly with our objective of keeping the size of our robot down, but it is a constraint in that it removes the flexibility of making the robot bigger if we found that necessary or simply wanted to. Not used by our team, but an option available was to print the chassis at the Texas Instruments Innovation Lab under the University of Central Florida’s Maker Space Lab Complex. Using this local lab avoids having to pay expensive fees or shipping but is a constraint on possible materials that could be used, leaving only plastic 3-D printing filaments such as PLA, PETG, or polycarbonate. Use of metals like aluminum, recycled materials, or sustainable materials such as wood to build the chassis are therefore also out of the question due to these manufacturing constraints at the prototype stage.

There are also sustainability constraints which affect the design of the robot. Sustainability constraints are constraints set that limit the resources used in specific parts. They assure that the product lasts a decent amount of time and will not be a constant harm to the environment. Creating a sustainable product helps future generations have the resources to create improve versions of our project. For example, sustainability constraints forced the team to use plastics that last a very long time, not use many resources, or be biodegradable. Another example was the technology used in the sensors. Sensors are made to be sturdy and last for multiple readings. This means that they rarely need to be replaced and are accurate most of the time. This allows the team to design the robot in a way where these sensors do not need to be easily reached since they should not be replaced often. This constraint has a minor impact on the design of the project since only one is being made and it is not used for an extensive period of time. This constraint does have an impact on the parts used. The parts used in this project last a lot longer than the robot is expected to last and many were created in a way that they impact the environment as little as possible.

4.2.5 COVID-19 Effect on Constraints

There are several constraints that affect the design of the robot dog. The biggest constraint affecting our dog is the manufacturing constraint given by the recent COVID-19 epidemic. At the start of the pandemic, several Chinese factories were forced to shut down. China is one of the leading countries in producing electronics globally. During the pandemic, several workers were laid off and many businesses were required to shut down. Furthermore, physical sales decreased substantially. Another restraint created from this constraint is the health protocols and resources that now need to be allocated by distributors and manufactures. This leads to increased prices, delays, or less of a product being distributed to a specific country. Currently, it is hopefully near the end of the pandemic and many businesses are starting to revamp to their previous capabilities. This has caused an increased in price in several components, and many of the less used components to stop production all together. This is what causes a heavy constraint in the design of the robot dog.

The robot dog depends on several small resistors and other components that are generally manufactured in bulk. Factories had to close, but not business. Businesses were still capable of performing with an online setting. This caused a massive supply bottleneck that has led to many of the components required for the robot dog to become scarce or more expensive. This ties into the budget and economic constraint of the robot. With COVID-19, the price of some materials has increased, while others have stopped appearing. This has led to several changes in the original plan of the design. The constraints with pandemic have created, force the team to create backup designs to the original. This document details many of the original parts, but the team has secured several sources of where to obtain more of these parts should this be required. Furthermore, the design of the robot is now more focused on the team doing most of the design. The pandemic has forced the team to not rely on prebuilt boards of specific PCB houses. Instead, the team is forced to adapt and use nearby sources, such as the resources provided by the university, to create the majority of components. When this is not possible, the team acknowledges that the overall price of the robot dog might increase as the scarcity of parts forces the team to buy parts at a higher cost.

There are some positives involved in the constraints provided by COVID-19. One of these strengths of this constraint is the opportunity to explore new technologies. With the supply deficit of many common parts, the competition was able to prosper. There are now new sellers on the market trying to gain a step over their competition before they climb back. This has led to the prices of many products similar to the common ones, to be cheaper. This had great impact in the part selection of our design. Many parts were advertised as cheaper than the competition, but the pandemic has led this to no longer be true. Part selection now has a larger variety and the design on the robot was heavily affected by this.

Another issue caused by the COVID-19 pandemic is the shortage of semiconductor manufacturing. Many semiconductors are manufactured in Asia which was shut down for a large part of the pandemic. Furthermore, transportation and budget constraints limited the number of semiconductors that can be supplied. This constraint has led to microcontroller and IC prices to increase dramatically as well as lead times. Furthermore, it has also led to scarcity of these products. This constraint heavily affected which ICs are used in the design. Many ICs had to be chosen quickly because the number of them available have become limited. Other IC's keep increasing in price as time passes on. This led to the purchase of IC's to be a priority for the team. This constraint also limited the amount of IC's available and the microcontrollers available. Like with the small components, several ICs are "out of stock" or sold at double or even triple the price. Several components needed to be replaced by similar products or by products that were slightly worse or better. Several adaptions were made into the design.

Adding to the troubles in manufacturing and transportation, the pandemic has also forced many companies and distributors to change their health protocols. Resources need to be accommodated to ensure the product being delivered satisfy the health requirements of a country. Furthermore, quarantining might be required which leads to delays in the arrival of shipments. This are other reasons on how the COVID-19 pandemic is a strong constraint for this project. Parts are not arriving on time and the time constraint of the robot dog does not allow the team to use these parts. Moreover, some companies might not want to invest the resources into the health protocols and requirements and simply not ship the product to Florida, leading to a lack of parts needed for the robot dog. As new coronavirus variants emerge globally such as the delta and omicron variants, new lockdowns and/or workplace requirements are put in place in different countries. For example, China, a very important country for electronics manufacturing, has a zero-tolerance policy when it comes to new COVID-19 outbreaks, affecting global manufacturing and trade of silicon and components that are needed in this project.

Overall, the pandemic leading to budget and manufacturability constraints led the design to be made in a flexible manner. The robot is not designed to work with a particular component, but rather designed to work with several similar components. This design approach allows the robot to be created without having to worry too much that a specific part is not available or outside the budget range. Dealing with this constraint has not been particularly difficult, but it is very impactful in the design, As mentioned, many desired components and features of the robot had to be removed or changed because this constraint tighten the effect of other constraints. For 2022, if everything continues moving in a positive direction, it is expected for the impact of this constraint to be reduced. However, this constraint is very unknown to the current world and the team must be ready to adapt in case more problems and limitations arise from the Covid-19 constraint. There is a possibility that this constraint might cause greater problems and the team must be prepared to confront them accordingly.

5. Project Hardware and Software Design Details

The design of the robot dog needs to be separated into multiple subsystems so that it can be easily tested and created. The first subsystem involves the power control of the entire robot. This subsystem explains how the robot is powered and the circuits used to do so. The second subsystem deals with all the components connected to the ATmega328p. The third subsystem deals with all the components related to the Raspberry Pi and the AI. These subsystems are designed in an independent manner; they expect the subsystem they depend on to work. For example, the second and third subsystems are created with little regard to their need in power. The power subsystem is designed knowing the requirements the second and third subsystems need, yet not knowing how the subsystem distributes the power. This division in subsystems is necessary to allow for individual problems to be fixed efficiently.

The potential problems of each subsystem are different and different methods are required to solve them. Problems in the first subsystem include incorrect voltage readings of unusual draws of current. Problems in the second subsystem include robot balancing and coordination between the motors. Problems in the third subsystem are mostly software related. To reiterate, this division can help fix the multiple problems. Issues with current and voltage originated from the first subsystem, while issues with movement are blamed on the second subsystem. There were scenarios where this problem division was not necessarily true, and the problem was in another subsystem, but these were minor and not very significant.

5.1 Initial Design Architectures and Related Diagrams

In section 3.4, many potential architectures where discussed that could fit the ideal of this project. From that section, it is evident that the team wants an architecture that has an outside that looks like a dog and an inside wide and big enough to house all the electric components. The architecture also needs to hide as many cables as possible and sensors are possible to make sure the robot looks realistic. There are many potential design architectures and by the end of this section, it is clear how our initial design of our project looks like and why it was chosen.

The very first stages of the design process involved conceptualizing the physical layout of the robot and the structure required for the project. Figure 23 is an initial sketch which was drawn up to illustrate the rough outline of the robot to be constructed. The sketch shows an architecture that fits all possible architectures discussed in section 3.4. For now, the head part is discarded, wheels are not considered, and no tail is placed. This design was created to understand, visually, where all the sensors and components would be placed and how the team would organize the design of the project. The architecture in figure 23 outlines the placement of various parts on the 3-D printed robot main body and 2-part legs. At each joint of the legs are identical servo motors (M1) used for controlling the robot's movements. For example, standing up, sitting down, and walking. The main body of the robot houses the electronics used to power the robot. On the exterior of the main body some sensors are placed such as the camera (S1) which acts as the input for image recognition and processing. The microphone (S2) for input to speech processing is also on the exterior of the body, along with the capacitive touch sensor (S3). On the front of the body which should typically face the user for interaction, is the main display (D1) approximating the location of eyes. While these were planned to be two separate screens, they were controlled by the same computer and be designed to operate and display in tandem.

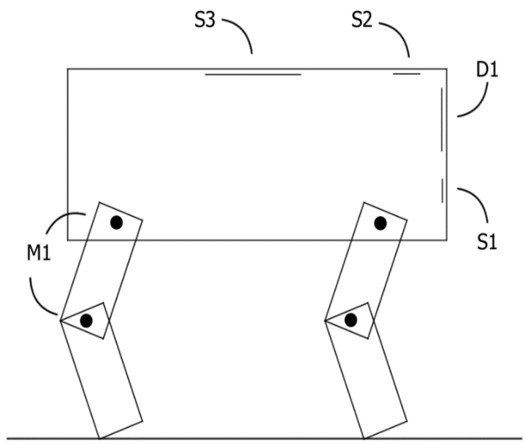

Figure 24: Initial sketch diagram of robot body

After considering this design, the team considered whether to use wheels on the design of the robot dog. By adding wheels, it would be easy to move the robot without having to worry about balancing and proper articulation and pacing between the motors. The wheels would be easy to control and would make the robot capable of moving quickly throughout any flat surface Furthermore, the motors for the wheels would have more torque allowing them to better control the speed of the robot dog. Even so, there was a major problem with this design. By adding wheels, the team would be failing to accomplish the first goal of the project; to make the robot appear like a dog. The wheels make the robot appear more like a vehicle that an animal, which is not what the team or the plans of this project desire. The use of wheels for this design was quickly discarded due to the fact that the wheels would have affected the overall appearance of the robot. Dogs do not have wheels and by placing wheels, the robot felt more like an automobile and less like a robot. Furthermore, the robot would already have motors in the joints of the legs. This means that the wheels are not necessary for movement, since the robot already has motors that can allow movement if they have enough torque, and the robot is balance appropriately. Hence, by removing the wheels, we would be removing redundant motors. A larger comparison between using wheels and no wheels is given in a previous section of this document.

Another design architecture decision came on whether the dog would have a head, or simply a face attached to its body. For the robot to look like a dog, it is ideal that is does have a head. This would potentially mean adding an additional motor and worrying about an extra articulation. There are advantages of the dog having no head. First, it would be easy to connect all the part and house the PCB. Instead of having a head, the robot would use the front of the body as its face. This would mean making the body wider, which gives an advantage to the design, as there would be more space to place all the electronics. The drawback from this design is that the robot loses its feeling of looking like a dog. For the robot to look like a dog, a head is preferred. More on this, by adding a head, the robot is capable of moving its sensors toward an objective. Previously, the team concurred that the robot would use a quadruped motion for mobility. This means that to tilt the body of the robot, it would require precise movement from several servo motors on the legs. It would be simpler to have one motor that rotates the head and mount all vision and auditory parts on the head. Then, only the head would need to be tilted to allow the robot to have a better idea of its surroundings.

Table 25: Comparison table depicting whether the robot should have a head part

Head	No Head
• **The robot would resemble a dog** • **Sensors can be placed in locations where they would be in a dog (i.e., microphone where ears should be)** • **Would require a motor for the neck** • **Time would need to be dedicated to design the head part, so it looks like a dog** • **Head tilting allows for vision of surrounding by only controlling one motor.**	• Wider front of the body gives more room for electronic components • Less cables exposed • One less motor and articulation to power and program. • No complicated design required, simply increase the size of the rectangular body. • Entire robot must be rotated to view surroundings (quadruped motion makes this complicated).

Looking at the table above (Table 25), most of the decision making comes from simplicity and functionality versus appearance of the dog. This decision can therefore be solved by looking at the requirements and specifications given in detail about this project. One of the requirements and goals of this project, is to create a robot that looks and feels like a dog. By choosing the design without a head, the robot would not look like a dog and would only be helping the engineers working on the robot. Therefore, the design is planned to have a head so the robot can look like a dog. To simplify things, the design does not specify that the head should move. Depending on how the rest of the design goes, the design leaves a possibility of having a fixed neck on the robot. This solution is still better than having a robot with no head at all. The initial design still has no clear idea on how the head looks, but it is likely be simple and only one degree of freedom.

After consideration, even if the idea was initially discarded, the team decided to use a fixed head for the robot. Creating a new head would be too complicated for the amount for time and so the team decided against it. The new head should also look like a dog, but it loses some of the doglike appearance desired. This solution is not the one desired but it does not affect the performance of the robot dog in any way, and it was necessary so the dog could be completed on time.

Now that the team has a clear idea on the appearance of the design, the team must decide on how the components are organized and where they are placed on the object. As explained on the explanation of figure 23, most of the sensors are located on the front of the robot, most of the servo motors on the legs of the robot, and the power and PCB boards are located on the body of the robot. To help organize this, the robot dog is divided into three subsystems (as can be seen in table 23) that are designed independently yet are still dependent from one another. The first subsystem is the power control subsystem. This subsystem includes are the PCB boards, batteries, and regulators that are used to control the current and voltage that is supplied to the other subsystems of the robot. Although designed independently, this subsystem must consider the requirements of other subsystems. The second subsystem is described as every component controlled by the

Atmga328p microcontroller. This subsystem includes the servo motors, the touch sensor, and all PCBs required to make them connect and be controlled by the Atmega328p chip. Again, it is designed independently from the other subsystems, but still needs to connect to them. This subsystem receives power from the first subsystem and also potentially gives some of the sensors input data to the other microcontroller. This other microcontroller is the Raspberry Pi, and it controls the third subsystem. This subsystem includes all the sensors that are located on the face of the robot as well as handle the AI and virtual assistant software. To better organize this subsystem, it is further subdivided into the hardware and software components. This subsystem oversees the camera, microphones, ultrasonic sensor, speakers, LCD screens, and most of the software. For the software design, there are two microcontrollers that help manage all the data being transmitted by the multiple sensors. A summary of all of this is explained in the table below.

Table 26: Division of overall design

Subsystems	Description
Power Management	Controls the voltage and current being supplied to the entire robot. Includes batteries, regulators, and any other PCB regulating voltage and current.
Motors and Touch Sensor	Includes all servo motors and the touch sensor. Everything in this subsystem is controlled by the Atmeg328p and its corresponding PCB.
Head Sensors	Includes the camera, microphones, speakers, LCD displays, and the ultrasonic sensor. All controlled by the Raspberry Pi microcontroller.
AI and Virtual Assistance Software	Software used to control the virtual assistance and AI vision of the robot. All managed by the Raspberry Pi microcontroller.

5.2 First Subsystem: Power Management

The first subsystem is assigned to all the power controls of the robot. The system needs to be capable of powering all the devices and components used in the robot dog. As mentioned, it was created independently, but data of how much power each subsystem is drawing is essential for this subsystem to be created. This subsystem must be tested by connecting it to the other subsystems and making sure they all work properly. This subsystem is expected to involve many circuits and be a big factor in the functionality of the robot. It is crucial that this subsystem works since it is the one that allows for the other subsystem to be implemented and tested.

To accomplish powering the robot, voltage regulators were required to provide a safe power source to all the power supplies. Furthermore, a DC-to-DC converter was necessary to convert the batteries voltage into 5 V DC. The power supply also needs to supply different components separately. One of the most power consuming components is the motor subsystem. It is expected for each motor to draw 100 mA each. There was a total of 8 motors in the entire robot. This is 0.8 A of current being drawn every time the robot plans to move all its movement motors. Since it is such a large amount of current, it is better to have this part of the system powered separately from the rest. The next section to power is the ATmega328p integrated circuit and the touch sensor attached to it. All these components are in the same subsystem, so it is crucial that they share the same ground node. Powering this part of the robot is simple and the voltage needs to simply be stepped down and regulated to 5V, as both components work well with this value. The entire power system was designed to share the same ground between both the power and the electronics (MCU, motors, and peripherals) for the most effective grounding possible.

The next part this subsystem needs to power is the Raspberry Pi, and all the sensors includes (this is the third subsystem). This subsystem is also expected to take a fair amount of current (around 500 mA). Fortunately, this system also uses 5 V to function so the same batteries with a buck converter could be used. All components used were selected to make sure powering and using them would be straightforward. This explains why everything works when the voltage is regulated to 5 V, and why this is necessary. This system was also designed with rechargeable batteries in mind. The biggest issue with this is the current consumption. If all motors are powered at max capacity, a typical 1300 milliamp-hour battery would only last around 2 hours. This system is designed to power everything and be as light as possible. The low weight allows for less power consumption from the motors.

The batteries used are rechargeable batteries. This means less waste and allows the robot dog to be recharge at any moment to ensure it can last the desired amount of time for a specific period. In a previous design, the team considered having the robot connect itself to charge, but the team realized this would be an unnecessary complication. Therefore, if there is extra time available, the intention is to add this feature to the robot. By using rechargeable batteries since the beginning, switching to the robot automatically charging should be easier. The change would then involve simple coding and maneuverability with the motors. In fact, if the robot works as the final product is supposed to, adding these features is not complicated at all, since all the parts needed are already built in the robot. Ultimately however, integrated battery charging had to be scrapped late in the project building process, resulting in using an external battery charger that requires the battery to be temporarily disconnected from the robot. However, even after hours of testing, the 2S LiPo battery used never had to be charged, starting from a full charge.

To regulate and step down the voltage, both simple 5V voltage regulators, as well as a breakout buck convertor board were used. The batteries that were used in this design have larger voltages than 5 V, which is what all the components on the robot dog required. This means a buck converter is needed to lower their voltage to this setting. Basically, this subsystem only includes several convertors and regulators, as well as the battery to make sure the entire robot is powered. The input and output of these circuits are the most important parts and can be sensitive. It is extremely important that these circuits provide accurate results that are also stable. Varying voltages could lead to damages in components which can cause delays in the integration and building of the project. Therefore, it is crucial that this subsystem works almost perfectly.

Details of all the different regulators, convertors, and batteries used are explained throughout this document. These have all been chosen to appropriately power the entire robot in the most efficient way possible. The team expected this subsystem to have some issues arise while building the project, so much of the time designing was designated to this subsystem. This turned out to be the case. This subsystem was important to the entire design since it is the system that determined whether the design would turn on. Although the system is not working directly with the features and most requirements of the robot, it is still important for the robot dog to function. Issues on this subsystem and the following subsystems are discussed in the section regarding testing and integration.

5.3 Second Subsystem: Atmega328p and Components

The focus of this subsystem is the motors and the touch sensor. This subsystem is the “body” of the dog. Every component not located near the front of the robot is part of this subsystem. Connections in this subsystem are simple most of the design comes from fitting all the parts into the plastic casting. The subsystem is later connected to the first subsystem for power and the third subsystem to send different inputs received by the servo motors and the touch sensor. Most of the power consumption comes from the motors. Because of this, the motors use a different PCB for power supply than the rest of this subsystem. However, it is possible to power all of it without such a separate power supply if the need arises due to parts or space constraints. Further details are given in section 5.2.

The Atmega328p microcontroller oversees handling all the motor controls and the touch sensor. All sensors not related to the AI vision of the robot are controlled by the Arduino Uno board; specifically, the Atmega328p microcontroller at the center of the Arduino is what processes it, and the board was used to interface with the relevant project hardware. The Arduino Uno uses the same integrated circuit and was used to prototype the design and test the software. This section aims to specify which sensors are controlled by the Arduino, how they are connected and how they influence the design. The components that are connected to the Atmega328p are the touch sensors as well as all the motors. The mic array and the audio sensor are located within the AI subsystem since they are heavily connected and need a more powerful computer to adequately process and output. The AI subsystem uses the mic array to communicate. The LCD screen is also heavily tied to the commands given by the AI aspect of the robot. There is the possibility of connecting the Atmega328p to the Raspberry Pi, but it is much simpler to keep both IC’s separate. However, this is something that is available and can be easily implement onto the PCBs of both subsystems. All other sensors are also tied to the face of the robot, so the team has decided to place them in the third subsystem. To summarize, the Atmega328p in charge of mostly the motors and the touch sensor, while all the other sensors are in the third subsystem.

The Atmega328p has 13 digital pins which can be used for PWM. Hence, these are the pins that are used to control all the motors. Some of these pins can be clearly seen in figure 25 below. The Atmega328p is the same integrated circuit (IC) microcontroller that is used in the Arduino. The Arduino Uno is precisely the device that is used to prototype this robot. The Arduino is a microcontroller board that everyone in the team has previous knowledge working on. It is simple to use. Furthermore, creating a PCB for this integrated circuit is easy and a good choice because the integrated circuit is smaller than the integrated circuit used by the MSP430 series of Texas Instruments microcontrollers, for example. It can be easily integrated into any custom PCB design we use for this project. At least 8 servo motors are required to move all the limbs. Also, using

PWM means that a clock needs to be functional. Therefore, a resonator or crystal oscillator must be used to power this clock. The schematic below contains all the components that are used specifically for the motor subsystem. It is important to note that it does not include anything related to the power supply, sensors, and other devices that pertain this robot. The schematic only contains half of the pins (since these are the PWM pins). The other have is shown in a later figure.

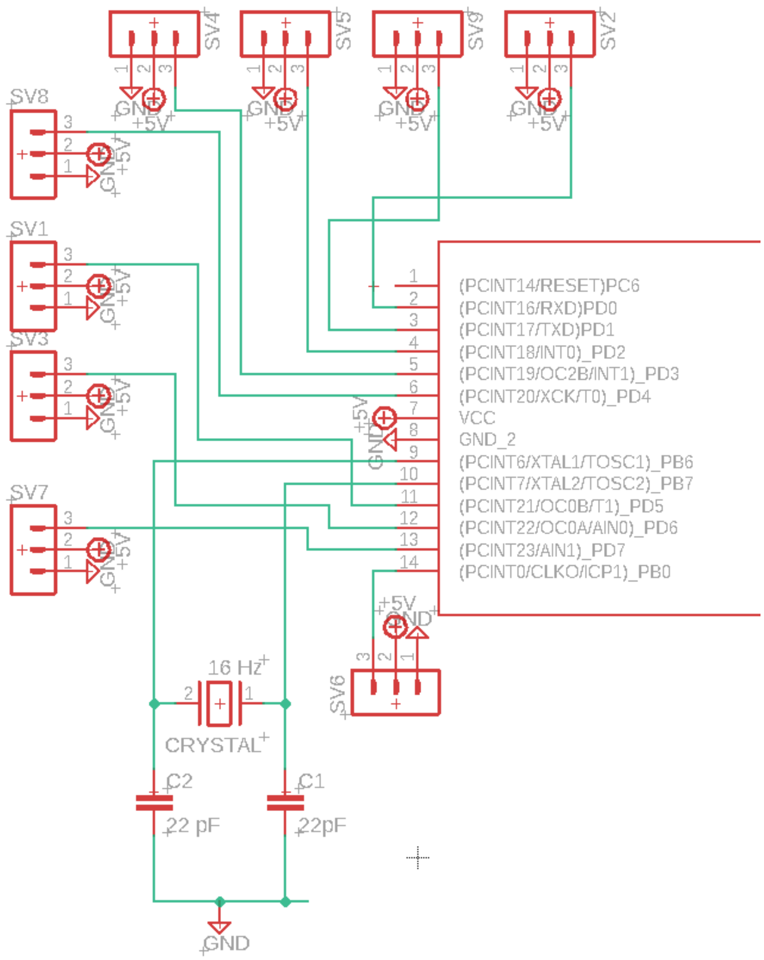

Figure 25: Rough Schematic of servo motor part of Atmega328p Subsystem

All servo motors are connected to the same power source. Servo motors are perfect for this function because there are constant shifts in weight as the robot moves. The power source needs to be capable of several hundreds of milliamps for all the motors to work properly. In most scenarios, not all motors on simultaneously, but the circuit must be designed to account for this. From Figure 25, it is depicted that the crystal oscillator has two small capacitors. These capacitors are there to help stabilize the crystal. This subsystem is simple as the servo motors and the Atmega328p are design to work together. The most complicated part of this subsystem is the power supply circuit. It is apparent from the schematic that several 3 pin headers are used to connect the servo motors to the PCB board. These headers were chosen because each one can hold one servo motor and allow for easy cable management. The motors are connected to the integrated circuit on the pins that are capable for PWM control. The design of this subsystem matters greatly, especially when it comes to power consumption and efficiency. Each of the motors need to receive enough current for them to work properly. They should be connected in series to the same power source. This allows them to all receive the same amount of current. This causes the circuit to pull a good amount of current. This is where the design of the power subsystem helps.

The motors are situated in the joints of the robot leg. The robot is series articulated. Series articulation means that it is contain two servo motors per leg, as can be seen in the figure below (Figure 26). Series articulation allows for simple movement without redundancy. It also provides simple balancing techniques and rotation movements. This design was chosen because it is similar to the articulation used by dogs. As can be seen by the image, one of the servo motors actuates the knee join, while the other connect the leg to the body. This type of articulation has some difficulties balancing, but by placing the stable feet, or inverting the joints of the back legs, the design can be made more stable. The legs are designed to be wide enough, so they hide the cables inside the leg (more on this in the design and architecture sections of this document). Should this not be possible for any reason, then the cables are placed towards the inside of the robot, to try to hide their existence. The robot must not look like a machine, so hiding as much cables as possible is important.

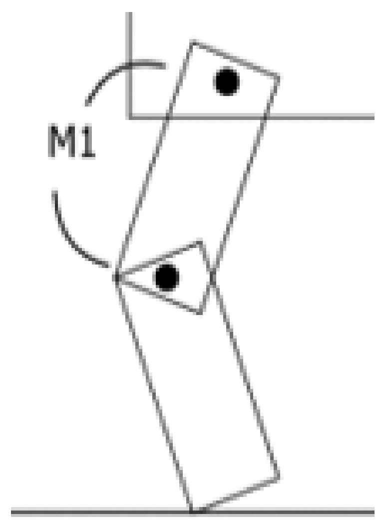

Figure 26: Series articulation design

As mentioned, all these motors are connected to the integrated circuit used by the Arduino Uno microcontroller. With precise software, this subsystem controls how the robot moves. By varying the angles of the servo motors' positions, we can modify the stances the robot dog takes. The team can make the robot sit or lay down with correct motor control. It is essential that weight is considered and that the motors have enough torque to move the body. Furthermore, the motors are not required to move quickly, but they must move precisely. Ideally, the faster the motors can move to certain position the better.

The design of the body is more straightforward and is technically not part of this subsystem. However, the body handles all the PCB motor drivers and the ATmega328p controlling the motors and the touch sensor. Hence, the body is designed to be large enough to fit all these components. The design of the body is a large rectangle and is made of PLA plastic. This plastic should be strong enough to hold all the materials and weight is it required to. This means that all PCBs and components that can be placed on the body of the robot are placed there. This allows for a reduction of cables and control of the weight of the robot. The next and final part this subsystem is the touch sensor.

The resistive touch sensor acts on pressure and is best used on an analog pin. This sensor acts like a resistor and is therefore very easy to implement. All that is required is a pull-down resistor and a power supply. For simplicity, a push button symbol is used to symbolize the resistive touch sensor. This circuit is simple and does not require much thought on its design. Viewing the subsystem, it is extremely important to manage and test the power consumption of this subsystem. The sensor is expected to be placed on the back of the dog body (at the top of the robot design). As mentioned previously, the body is a flat rectangular surface. The cover of the body is holding much weight, so it is thinner than the bottom. This allows for the touch sensor to work under the plastic casting. Should the team find that this configuration is not possible after testing, the sensor is still thin enough that should not affect the overall appearance of the dog if it needs to be placed on top of the body. Depending on the dog size, it might be necessary to use more than one touch sensor. As can be seen from the schematic (Figure 27), there are plenty extra analog pins for additional sensors. The team is trying to avoid this, however, since it increases the cost of the design. From the image, pins 24-28 can be used for additional touch sensors if required.

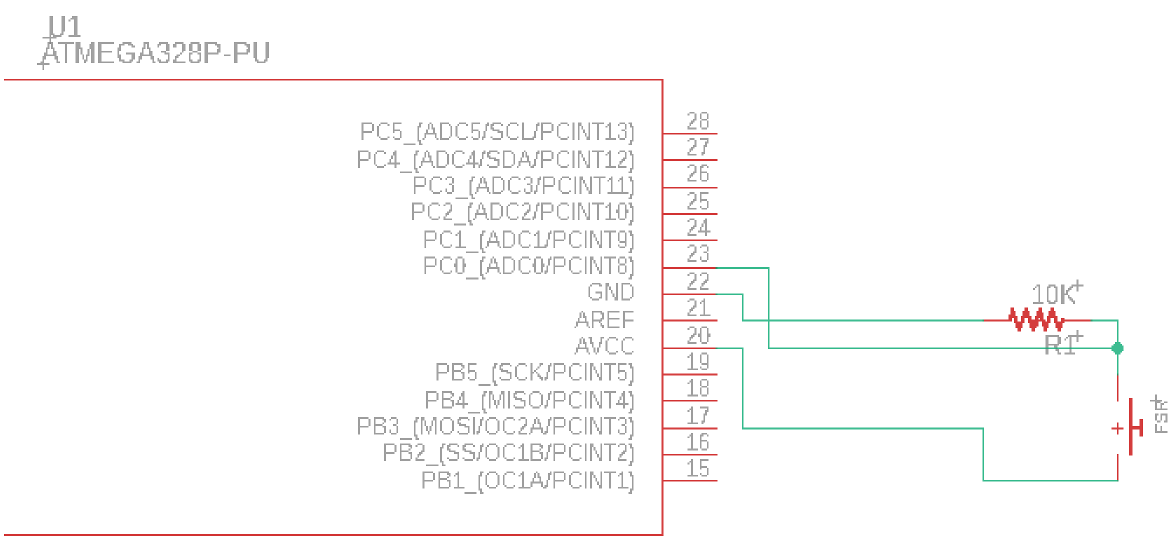

Figure 27: Schematic of Touch Sensor connected to Atmega382p Subsystem

One of the major issues in this subsystem is weight management. The motors need to be capable of handling the weight of the entire robot dog. This has been taken into consideration when choosing motors in the part selection section. Furthermore, hiding cables is also important. The legs have been designed to fit the cables inside, so they do not need to be visible. A final consideration of this subsystem is the power consumption. Servo motors take a decent amount of current when compared to the sensors and microcontrollers used in the project. It is important that the power subsystem can handle the current draw that resulting from all the motors. Motors need to be selected so they can be energy efficient but also lightweight. The choice of motor is better explained in the part selection section of this project.

5.4 Third Subsystem: Raspberry Pi Sensors and AI

This subsystem controls every sensor located in the “face” of the robot dog. This subsystem is also heavily tied to the software system. This subsystem contains all the sensors that are connected to the powerful Broadcom BCM2711. This is a powerful IC that can handle all these sensors. This IC is also responsible for the AI integration of the robot. This IC is specifically designed to be used with the Raspberry 4. Because of this, a custom PCB design cannot be made. To simplify our problems, a breakout board was made for each of the components. The mic array already contains a breakout board, and its schematic is shown on a different section. Likewise with the LCD screen. The schematic of the Raspberry Pi is also shown in another section. This makes the design of this subsystem straightforward. Most of the complexity of this subsystem relies on the software design. There is some complexity in this design, however. The sensor can be accommodated on the head of the robot, but they are tightly packed. It is important that the sensors that need visibility are not obstructed by other sensors, cables, or objects that might alter their readings. Furthermore, all these sensors have cables that have to be long enough to reach the microcontroller, which is not situated in the head, but in the body of the dog. This is done so the sensors have enough room to fit properly in the head of the robot dog.

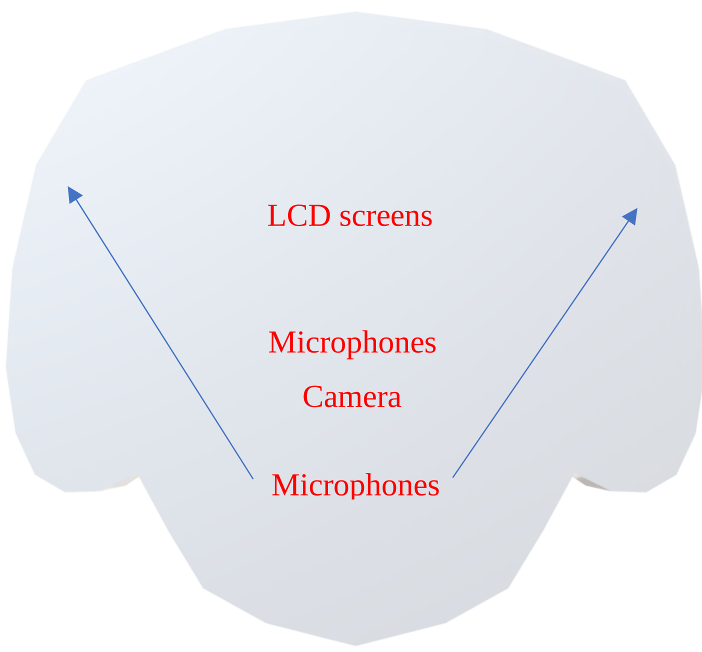

Figure 28: Representation of where sensors are in respect to face of robot

As can be seen from the figure (Figure 28), the sensors are layered and not obstructing each other. The purpose of this design to make sure the camera and ultrasonic sensors are not obstructed. The microphones should also help make the robot feel more realistic and dog-like. Connecting the multiple sensors is simple because none of the sensors are complicated or require special voltages. The difficulty in the physical design of this subsystem is dependent of how all the sensors are placed on the head of the robot. As mentioned, there are several sensors that are directly connected to the face of the robot. All these sensors need to fit without obstructing each other. To solve this issue, the subsystem is created to resemble the physical aspects of a dog's face. The microphones are placed where the ears would normally go, the LCD screens are supposed to go where eyes normally go. As can be seen, already these two components are far apart and should not obtrude each other. The camera is placed in the nose of the dog. This way it does not obstruct the other component and has clear vision of everything. Likewise, the ultrasonic sensor is placed in between the nose of the dog and the eyes. With proper aesthetic design, this can be made to look and act fairly realistic when compared to a real dog. Separating the components this way also allowed the components to have ample space for cable management, and the camera has enough vision of its surroundings.

The biggest issue in this subsystem is making everything fit in the front of the robot. Space has been made in the middle body of the robot to fit additional parts should it be necessary. The team realized that not everything can fit in the front of the dog. This means that the microcontroller controlling all these sensors are located farther back to allow the configuration of sensors at the front. The design architecture shown in figure 28, only shows the front of the design, it does not show where all the PCB boards and microcontrollers are placed. As mentioned, these were placed in the body of the robot near the front.

The most complex part of this subsystem is the software. This subsystem needs to have software that not only controls al the sensors mentioned previously, but it also manages the functions of the virtual assistant. There are many sensors involved in this subsystem. This subsystem determines if the robot dog senses like a dog and is of big importance on whether the robot dog can perform many of the requirements set by the engineering team. Specific details of the software design are given in the software design section of this document.

To summarize all three subsystems. The first is the power subsystem that makes sure every component in the robot has enough power. The second subsystem is the motor and touch sensor subsystem. This subsystem oversees making sure the robot can move to different locations. It also makes sure the robot moves likes a dog and can make dog-like poses. Finally, the third subsystem contains most of the sensors and the true "brain" of the robot dog. This subsystem makes sure the robot can sense like a dog and can view its surroundings. Together, these three subsystems make the entirety of the robot dog.

5.5 Raspberry Pi Camera Module V2

This camera is what gives our robot the vision it needs to recognize objects. With it 1080p30 resolution this camera should be more than powerful enough for are robot to recognize objects. When it comes to the camera placement we try and place the camera on its nose. The reason for the nose placement is because the eyes were supposed to be LCD's and the camera won't be able to fit properly next to the LCD's. Thus, making the next logical placement the nose. This camera dimensions are as followed with the Length = 120mm, Width = 75mm, and Height = 23mm and a weight of 32g. This camera is powered with the Raspberry Pi and uses TensorFlow with python to give are robot the vision it needs. When it comes to an end result, we want are camera to at least be able to tell the different between colors but overall, we want to have been camera to be able to detect it owners and recognize unfamiliar members and categories them.

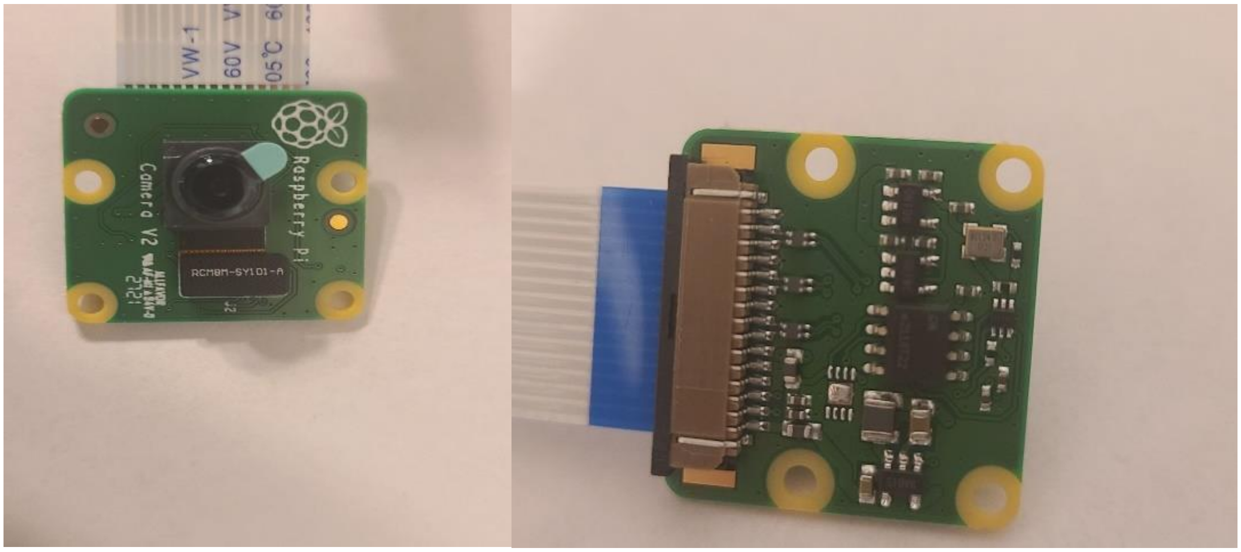

Figure 29: Raspberry Pi Camera Module V

5.6 Hardware Design

Having observed the larger scope of the project, it is important to look at the hardware design of the individual parts. These are the parts that are used for individual components of the design. Previously, the functions, technologies and specifications of the parts have been explained. In this section, the hardware details of the important larger parts are explained.

This section includes the details of the parts that are used in the robot dog. Unlike the previous section, this section aims to look at the individual design of each of the hardware parts. Some of these parts come in already built PCB controllers. These are explained in further detail. Both microcontrollers used for programming the robot dog are also explained in this section. By the end of this section, it should be clear how all the hardware components used in this project are designed**.**

5.6.1 ATmega168/328P

The first figure in this section (Figure 30) shows the ATmega328p on the Arduino UNO Rev 3. The Arduino uses this integrated circuit as its main component. The Arduino is used to explain some of the features of the integrated circuit. The Arduino UNO is designed to work specifically with this integrated circuit. Hence, most of the features available in the integrated circuit and shown and exemplified in the Arduino UNO.

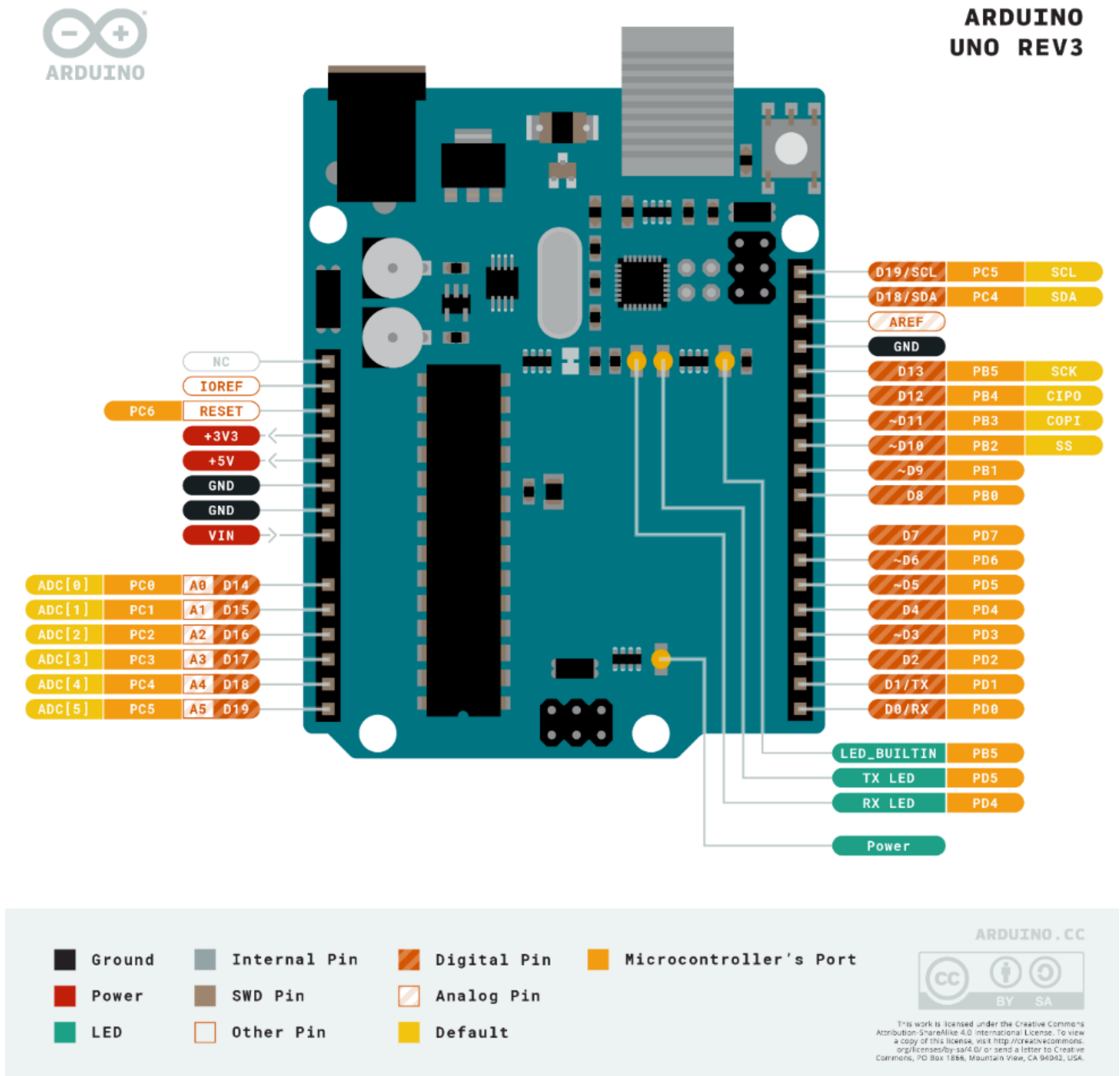

Figure 30: Arduino UNO Rev 3 Board Layout

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions each operate at 5 volts. The pins are seen in figure 30. Each pin can provide or receive up to 20 mA as recommended operating condition. The pins also have internal pull-up resistors, which are disconnected by default, with a resistance value of 20-50k ohm. A maximum of 40mA cannot be exceeded on any I/O pin to avoid irreversible damage to the microcontroller.

Additionally, the Arduino Uno not only has digital pins, but it also has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e., 1024 different values). By default, they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. There are a couple of other pins on the board such as AREF which is the reference voltage for the analog inputs this pin can be called for use and referenced with analogReference(). As well as reset, which brings this line low to reset the microcontroller. This pin is typically used to add a reset button or is shielded to block the one on the board. As a reminder, these pins and their explanations are the same ones in the Atmega328p, which is the integrated circuit used for the robot dog.

All sensors used with the ATmega328p also work with a 5 V supply. The design of this integrated circuit makes it perfect for controlling all the sensors that it we are required to control. As mentioned, the ATmega328p used in our design controls the touch sensor and the motors. The feedback system of the servo motors is heavily used to ensure the robot can move precisely and correctly. The rest of the Arduino hardware is helpful for interfacing with the integrated circuit chip and programming it. The figure below gives a better explanation on how each of the pins are configured for its use in the robot dog.

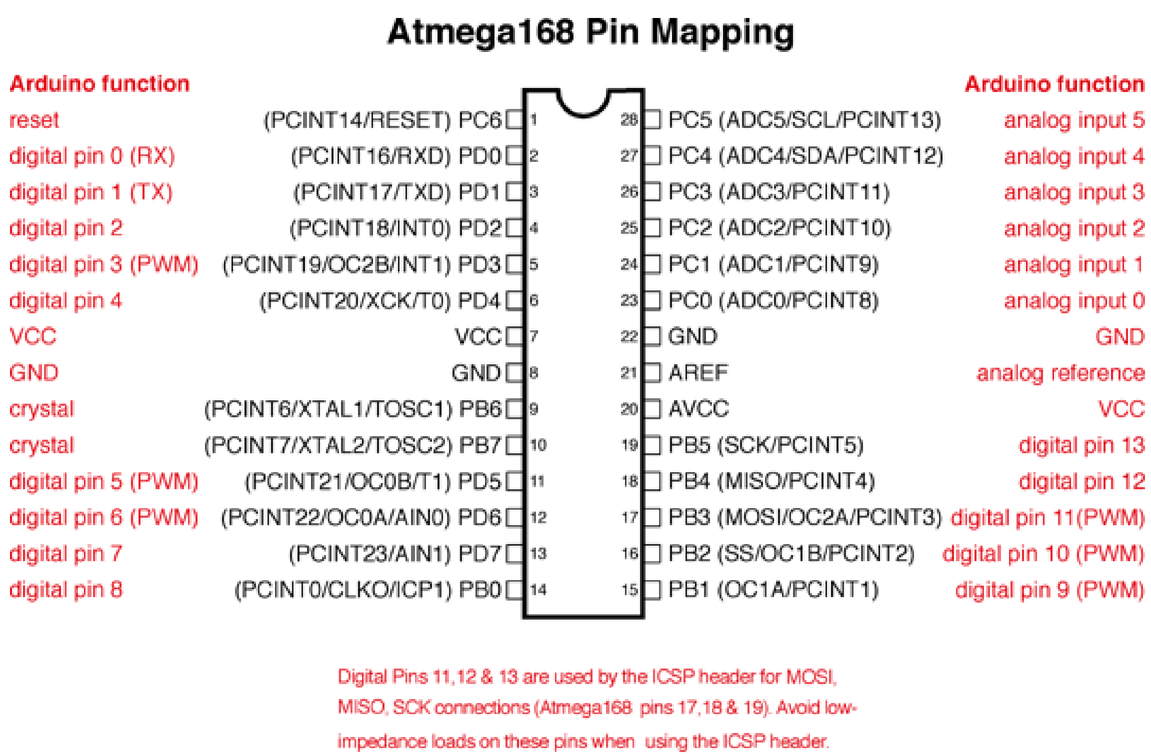

Figure 31: ATmega168/328P-Arduino Pin Mapping

Previous sections have already showed that nine digital pins 2-6 and 11-15 that are used to control the servo motors. Pins 9 and 10 are used to control the crystal that determines the PWM cycle. Finally, pin 23 is used to control the touch sensor. The integrated circuit is designed to have all analog pins on one side grouped and have the other pins be digital. The hardware design of this chip makes it easy to use without having to constantly check the datasheet. Working knowledge of the pin mapping is necessary for this project to advance without those types of delays. It is a versatile and feature-rich microcontroller.

5.6.2 Broadcom BCM2711

The Raspberry Pi 4 has a very powerful integrated circuit that is capable of handling multiple sensors as well as AI and virtual integration. The integrated circuit is known as the Broadcom BCM2711. Broadcom has made an effort to make the Raspberry Pi as powerful as a regular desktop computer, while also keeping a very small size. Each Raspberry Pi version has their own integrated circuit. The integrated circuit in this specific version is the most powerful they have done. Other version would also work for this specific project, but this particular one was chosen because of constraints making it the only one available currently on the market. Due to its powerful processing power, the Raspberry Pi is used for mining cryptocurrency making it extremely hard to find in the present market. This integrated circuit is specifically designed to work only with the Raspberry Pi 4. Unlike other microcontrollers, the Raspberry Pi has a uniquely designed integrated circuit so it can provide the best performance possible. Below, an image of the Raspberry Pi used with label components is shown. It is crucial to understand the capabilities of this integrated circuit and microcontroller to understand why it is useful for the integration of the virtual assistant in the robot dog.

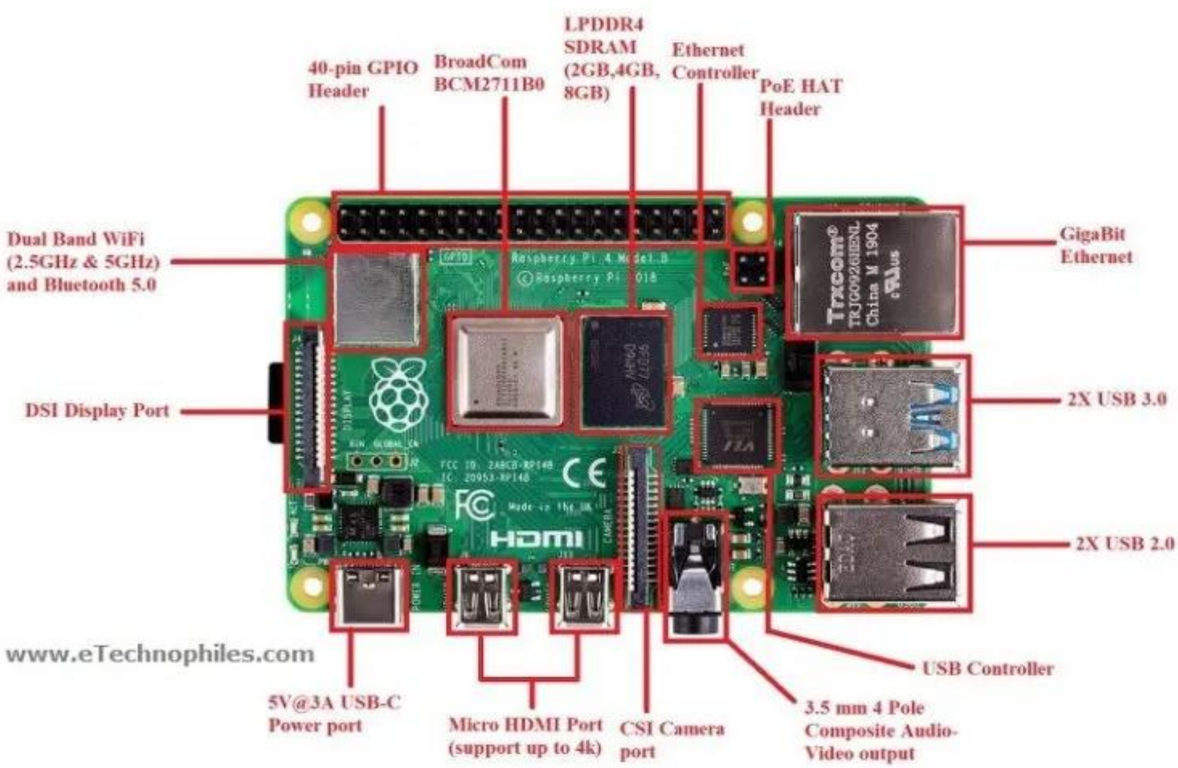

Figure 32: Raspberry Pi 4 Board Layout

The Raspberry Pi 4 (Figure 14) is the hardware design that is used to accompany the Broadcom BCM2711 chip that is performing all the heavy computation. The Raspberry Pi 4 is capable of Dual Band Wi-Fi and Bluetooth 5.0. Furthermore, it contains 2 micro-HDMI ports, a 3.5mm 4 pole Composite Audio-Video output, Gigabit Ethernet, a DSI Display port, a CSI camera port, 2 USB ports, and one USB-C port. Moreover, the Raspberry Pi 4 also contains 40 GPIO headers pins. All these ports are more than enough to connect all the sensors that are required to work in conjunction with the CPU. The sensors are connected on the GPIO pins, while the camera is connected on the CSI camera port. The microcontroller was also connected to a PC for programming. The Raspberry Pi is powerful enough to have its own OS. This means that a high-level programming language can be used. Furthermore, it also means that the Raspberry Pi has enough processing power to manage the AI vision and the virtual assistant. The Raspberry Pi small design also makes it ideal for fitting inside the small chassis of the robot. Figure 33 goes into more detail of the CPU itself.

The design of this piece of hardware is made to be easy to interface. As mentioned, it has several ports so the user can access the many functions this microcontroller provides. These ports are easily accessible. As mentioned previously, the robot dog has multiple cables coming out of the head of the design and connect to this microcontroller. It is important for the microcontroller to have easily accessible ports so the sensors can be connected easily and without causing much conflict.

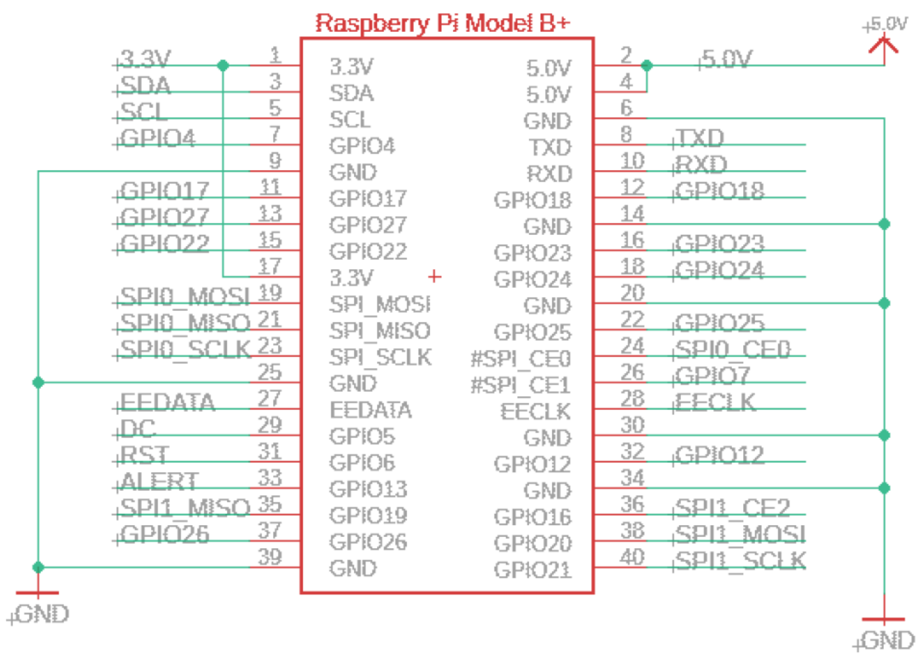

Figure 33: Broadcom BCM2711 GPIO Connector Pinout

By itself, this model does not provide much insight on how it is useful for the overall design of this project. One thing to note, is the 5V and 3.3V levels this integrated circuit is capable of using. This is very useful as the sensors used in this robot all work perfectly with a 5 V supply. This means that the same supply source can be used for all components. Furthermore, from the schematic it is apparent that the chip contains several GPIO pins which are required for connecting the multiple sensors. A special camera port was added to connect the camera easily to the integrated circuit.

5.6.3 Adafruit 1.54" 240x240 Wide Angle TFT LCD Display with MicroSD - ST7789

In order to connect the LCD display to the microcontrollers, an LCD display board is required. The boards allow the LCD display to be configured through the microcontroller. In this case, the team has decided on adopting the Adafruit Wide Angle TFT LCD Display. For hardware design, one of the most important constraints is display size. We need the display to be big enough so they eyes can be visible, yet not big enough that the display is intrusive to the overall design. This display is big enough to animate the eyes while also being simple enough to program. The rectangular shape of the display also helps with the aesthetics of the dog; it looks better if the dog has symmetrical square eyes rather and large rectangular eyes. The technology used for the LCD display is explained in further detail in a previous section.

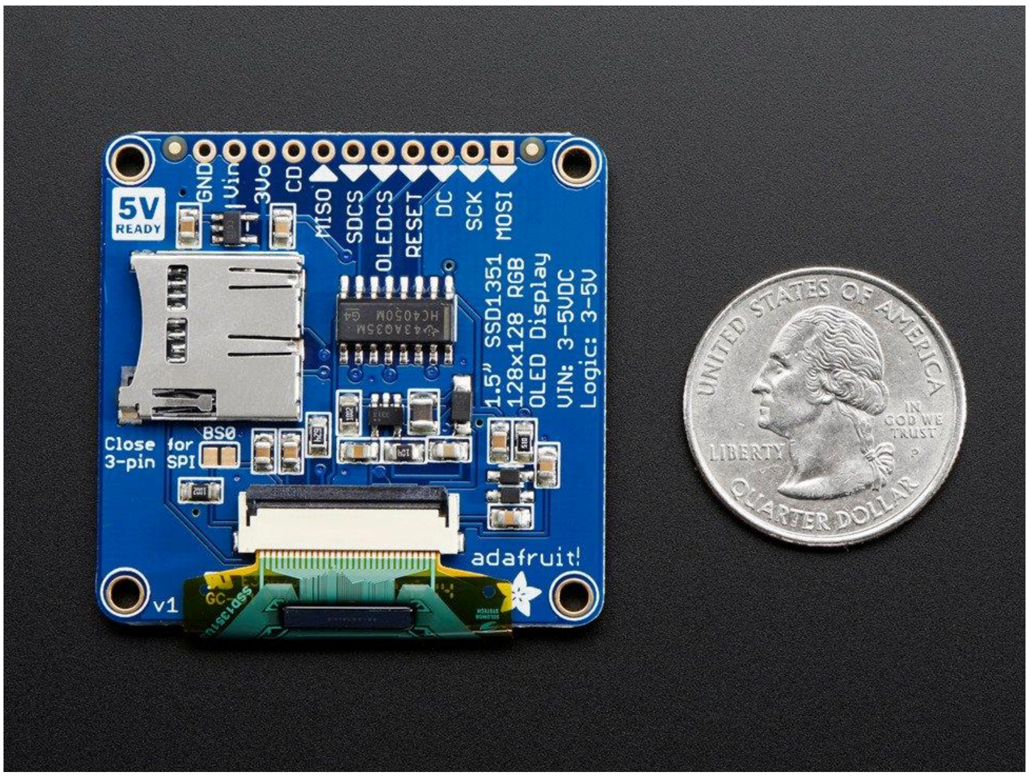

Figure 34: Adafruit LCD Board Layout

Figure 16 shows how the board looks and compares it to a quarter. As can be seen, the controller is very small and perfect for our design. It is not very bulky and can be easily hidden. It is also light weight and thin. All the components are hidden on the back of the PCB so the LCD screen can be main center of attention. Looking at the possible connections, it is apparent that the LCD display uses SPI connection interface. This is clearly indicated with the MOSI and MISO pins. It is also clearly visible as a label on the PCB board. There is also a reset pin on the PCB board. This is interesting since the reset is normally done with software and there is no need of using this pin.

Besides this pin, there are the regular voltage in and ground pins, and other common pins found in sensors. Next, the schematic of the controller is shown below. The Raspberry Pi integrate circuit is also shown to demonstrate how the two interconnects with each other.

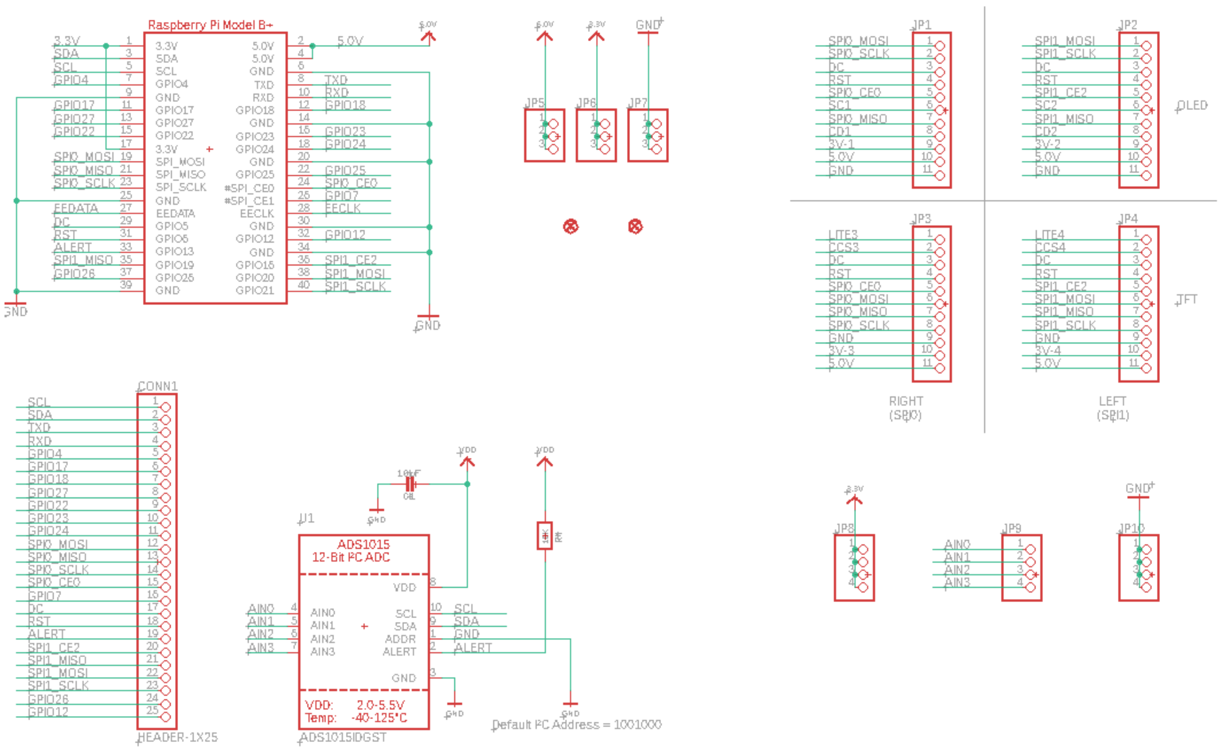

Figure 35: Schematic of LCD Eyes Bonnet

From the schematic shown above (Figure 35), we can notice that the board contains an ADS1115 circuit. This is a programmable analog to digital controller that is used to control the LCD display. This allows the display to vary the brightness of the different pixel colors. The schematic also includes the layout of the Raspberry Pi Broadcom BCM2711. This is the integrated circuit in charge of many of the components and not just the LCD. It is placed on the schematic to compare sizes and observe the different pin configurations available. Other components are mostly headers to connect the display and the integrated circuit together. Without these, it would be complicated to connect both components together.

This board is a perfect representation of the board used in our design. These are the components the team requires to be able to successfully handle the signals given by the LCD display driver present below. The main difference between both these schematics is that one contains the driver that is used to connect the LCD display to the Raspberry Pi controller while the other is the schematic of the LCD display. The schematic of figure 35 is slightly modified on the final design but it is ideally what the team used in the robot dog project. The schematic below (Figure 36) is made by the manufacturer and the team did not modify it since it is not necessary. It is only there to provide information about the LCD display.

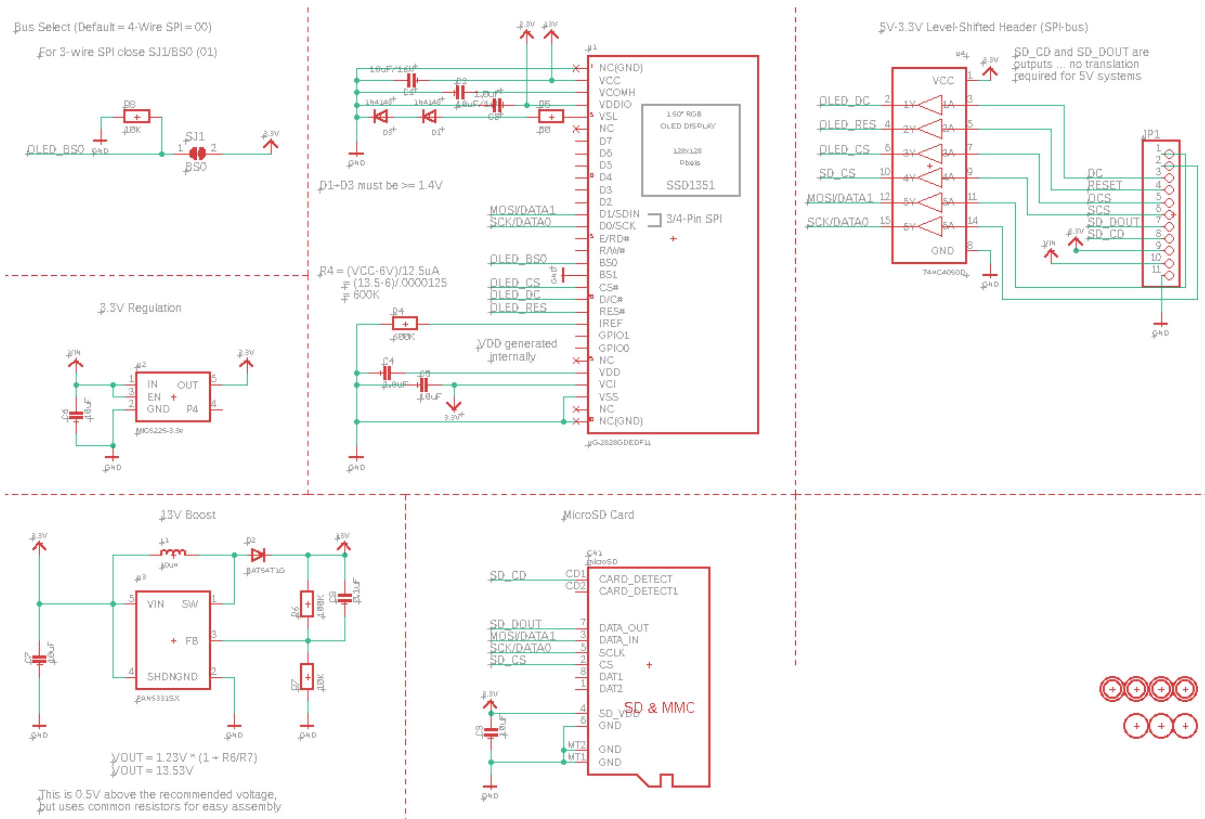

Figure 36: Adafruit LCD Board Layout

The schematic in the figure above (Figure 36) shows the LCD board layout. Comparing it to the previous schematic, this schematic holds all the components that are required to make the LCD display work, rather than the component used to connect the LCD display to the integrated controller. This is already built to make the LCD display function properly. This is not something the team is building since this is the schematic of the component itself. This layout explains how the manufacturers were able to create a powerful display in such a small design. The schematic includes a voltage booster, a regulator, an SPI switch, the headers, and an SD card. The SD card is not necessary for the LCD display to work, it simple stores information provided by the display. The header is used to connect the display to a microcontroller of other PCB board. Like with the LCD display and the integrated circuit, the header is necessary to make it easy to connect these components together. SPI is used to control the LCD display. It is a special type of interface used so the microcontroller can communicate with the sensor effectively, without mixing the input and the output signals. The voltage booster allows the display to have great resolution and quality with a low voltage power source. It also ensures the display receives the proper voltage it requires to work as intended.

The LCD display was planned but not included in the final project due to the unforeseen complexity surrounding including and intertwining an LCD with the other systems. Showcasing emotions which tie to certain actions taken by the dog was seen as an undertaking we would like to keep as a possible future upgrade requiring more time and modification to the robot head STL file.

5.7 Summary Design

The overall robot dog design has three major subsystems, and a software subsystem. Each subsystem is designed to help facilitate the overall design of the project. The second and third subsystems depend on the first subsystem, making the first the most important. The first subsystem deals with power all the components in the circuit. It is essential for any of the components to work. It determines whether the robot turns on and work correctly. The second subsystem deals with the motors and the touch sensor. It is the subsystem that handles the movement of the robot. It makes sure the robot moves like a dog and can be set in positions a dog would normally use. The third subsystem controls all the sensors located int the face of the robot: camera, ultrasonic sensor, microphone, speakers, etc. It ensures the robot can sense its surroundings and can do so in a way similar to how a dog would recognize its surroundings. This subsystem is also heavily tied with the AI integration of the software design. Furthermore, it is also responsible of the virtual assistant integration. It is the main “brain” of the project and was used to satisfy most of the robot dog’s requirements. Some of the subsystems also contain hardware that have their own unique designs created by their respective manufacturers. These designs help make the overall robot design light weight and easy to create.

All subsystems are joined together to create the hardware system. This system is what determines how the robot reacts and moves in its environment. For the hardware system to work, it needs a working software system. The software is divided among to microcontroller for prototyping. The software is designed to use as many components as possible to make the robot act like a dog. Furthermore, the software system design is created to make an easy-to-use virtual assistant. Combining all subsystems should lead to having all the components necessary for the dog to function properly. Diving the robot this way allows for easy troubleshooting and organizing.

The first subsystem controls the batteries and voltage provided to the rest of the design. The second involves all the components connected on the PCB that house the ATmega328 chip. This would include motor and movement control, as well as the touch sensor. The final subsystem involves the PCB controlling the LCD Display, Mic Array, Camera, and the ultrasonic Sensor. This is also the microcontroller responsible for handling the AI. The image below (Figure 37) gives a visual representation of each subsystem. This image should make the overall project understandable from a bird’s-eye view and lays out the priorities outlined earlier. For any robot, various subsystems typically overlap and interact with one another in a constant state, unlike other systems which can pass data or intermediate products onto another stage in a sequential manner. Our project system integrates custom components and designs with aftermarket solutions to reduce costs and improve interoperability.

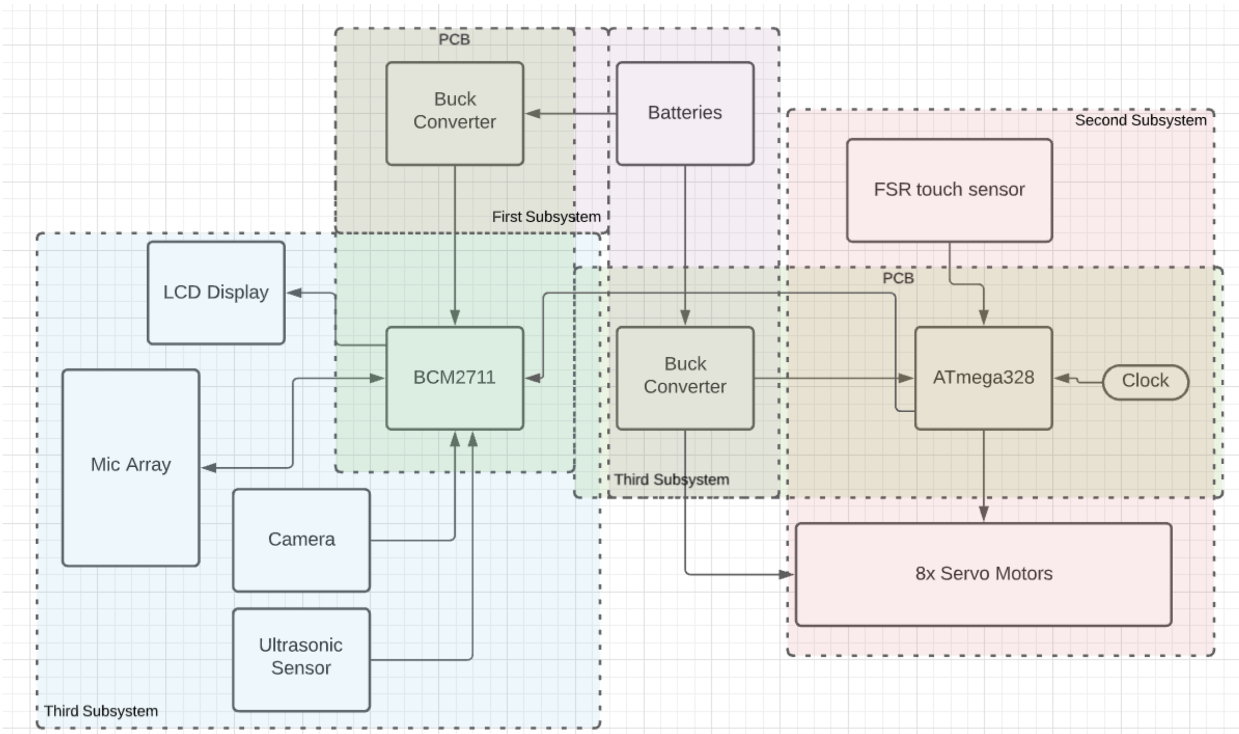

Figure 37: Hardware design and subsystem division

The flowchart above shows which components are inputs and which are output to different sections. It shows that the first subsystem powers both of the other subsystems and does not care about the process occurring withing them; it does not receive any input from either subsystem. The first subsystem is located near the back of the robot and contains a small PCB and the components mentioned previously. The third subsystem contains many of the sensors and it located in the front of the robot. As can be seen, both microcontrollers are connected but in separate PCBs. In the design, the microcontrollers are designed to work with each other as little as possible. The PCBs have a USB connection so the ATmega328 can connect to the other microcontroller if necessary. As mentioned, this division allows specific potential issues to be contained in an enclosed environment. As an example, if there is an issue power the devices, the power subsystem can be the focus of the next few weeks. If there is an issue with movement, then the second subsystem is the focus. More on this, my assuring the subsystems work independently before they are connected to the other subsystems, we can make sure there is now major issue when integrating and constructing the entire robot.

Most of the cables, connections and the PCBs were hidden inside the plastic model of the robot dog. The model is designed to be simple to print while also showing resemblance to a dog. The legs built to be divided in two major limbs. These limbs are connected and actuated by servo motors. The body of the dog is hollow and rectangular. Within it, the PCBs, batteries, and all other electronics are be hidden. Finally, the head of the robot contains many sensors that have cables extended to bring their connections to the body of the dog, where the PCBs are housed. This outlines how the robot was built and its many components and features.

6. Project Software Design Details

The software portion of our robot-dog was implemented on an Arduino Uno and Raspberry Pi. Each takes in inputs from its surroundings in order to process them into outputs. The code had many tasks to carry out. The main tasks include processing color and voice inputs, generating pulse width modulation signals to control the pulses to the servos, display the robot-dog's eyes/emotions on the LCD panel, receive and interpret touch data from the touchpad sensors, etc. The software needs to be designed in a way where it can interface with all the components, while also having each component separated for easy debugging.

6.1 Software Functionality

The objective of the software boils down to a few main purposes: reading/displaying information to the user and controlling the motor. This is done through many functions that are implemented into the Arduino Uno and custom code for recognizing and processing input from the 8 Megapixel camera to the Raspberry Pi. The software environment that was be used to compile and transmit the code was the Arduino Integrated Development Environment and Geany for the Raspberry Pi. The Raspberry Pi uses the inputs it receives and integrate them into the virtual assistant AI software. Overall, the software helps control the sensors, motors of the robot and control/create the virtual assistance used within the robot.

6.1.1 Camera Input

For the sensor input we are going to be using the Raspberry Pi camera. Our robot dog is able to perceive distance from itself to an object in order to avoid hitting any object. The Raspberry Pi camera has an object distance of TensorFlow's maximum calculatable distance based on the machine learning. The camera senor is 7.9mm diagonal 8MP Image Sensor with an active array of 3280x2464. The way this senor gets it distance is by using TensorFlow and machine learning algorithms. Then we can measure the distance by setting:

Distance = (Time x Speed of Light) / 2.

Speed of sound 340 meters per second = 0.034 centimeters per microsecond

Once these variables are set, we would need to acquire the value of time in microseconds when the sensor reads the object ahead of it. Once distance is obtained, this information is used in conjunction with the information given from the camera to help the AI visualize where the dog is located.

6.1.2 LCD Display

For the LCD we are going to use Adafruit 1.54" 240x240 Wide Angle TFT LCD Display with MicroSD - ST7789. The LCD has a screen length of 1.3" inches and a screen height of 1.2" inches with a 240x240 pixel display. The display is 16-bit pixels with a full color and can be powered with either 3.3V or 5V. It also has a built in microSD slot. The overall dimension has a length of 1.7" inches, width of 0.2" inches, and a height of 1.6" inches. With mounting holes with a length of 1.5" inches and a height of 1.4" inches. We were planning on using this LCD to display dog eyes and try to make it so the eyes follow you.

Table 27: LCD Display

Brand	Adafruit
Screen Dimensions	1.3" L x 1.2" H inches
Dimensions	1.7" L x 0.2" W x 1.6" H inches
Mounting Holes	1.5" L x 1.4" H inches
Special Feature	Built in microSD slot

As mentioned in the previous section 5.6.3, the LCD display was planned but not included in the final project due to the unforeseen complexity surrounding including and intertwining an LCD with the other systems. Showcasing emotions which tie to certain actions taken by the dog was seen as an undertaking we would like to keep as a possible future upgrade requiring more time and modification to the robot head STL file.

6.1.3 Pulse Width Modulation Motor Control

The motor we are going to use is a servo motor which is a small motor that is highly efficient and is very energy efficient. We are using this motor to give our robot dog walking ability, ability of turning it head, and movement of the tail. A servo motor is just a DC motor with a potentiometer and a control circuit. With the motor attached by gears which control the movement. When the motor rotates the potentiometer changes the resistance so that the control circuit can control how much movement needs to happen and the direction. What makes the servo motor very energy efficient it all comes down to the shaft of the motor. When the shaft hit the correct position, the power supplied to the motor is stopped. Now if the motor is not at the desired position, it tries to correct itself. This makes sure that the motor uses the correct amount of power to complete it takes.

A servo motor is controlled with the pulse width modulation. So, there is a minimum and maximum pulse and a repetition rate. A servo motor normally turns 90° with a total movement of 180°. There is normally 3 servo position 0°,90°,180°. Depending on the duration of the pulse from the PWM it determines the position of the shaft on the motor. The servo expects to see a pulse every 20ms. Let say a 5ms pulse turns the motor to 90° then anything less than 5ms turns the motor to 0° and anything more the 5ms turns the motor towards 180°. When it comes to the servo it moves to the correct position and hold that position until a new pulse comes in or you repeat the same pulse over and over. If the motor doesn’t get a pulse by 20 milliseconds, then it returns to its natural state.

Table 28: setPos Servo Motor Positions

Pos	Default	Standstill	Paw Trick	Left Front Forward	Right Front Forward	Left Rear Forward	Right Rear Forward
1	65	65	65	58	58	65	65
2	80	80	80	66	66	80	80
3	120	120	120	120	127	120	120
4	80	80	80	80	94	80	80
5	45	42	42	42	42	42	42
6	80	95	80	95	95	95	95
7	125	128	125	128	128	135	135
8	90	75	90	75	75	66	66

The table above (Table 27) provides the angles required for the different servo motors so the robot can be set in different positions. Arduino does not support multi-processing or multi-threading, meaning that the Arduino cannot do two or more things at once. Thus, the only method to properly actuate the legs of our robot would be to incrementally move the legs, alternating with tiny increments of each motor so that an appearance of simultaneous movement is created.

6.2 Code Algorithms Description

For this project we are going to be using AutoCAD to design the body of the dog and use python for the dog functionality. Python can handle all the dog movement, dog AI vision, and the dog's speaking abilities. We would like for our dog to be able to freely go around a house with no assistants just like a dog but due to time restraints we are going to be using a remote control for the movement. We also want the dog to have realistic head and tail movement when the owner interacts with it. When it comes to the dog Ai vision, we want are dog to be able to recognize it owners only be able to work for its owners. We also want are dog to able to avoid any obstacles in its path, so it won't collide into anything. For it speaking abilities we want the dog to have a realist bark and have all speaking functions that an Alexa has. All these features are coded in python and run off a Raspberry pi 4 and an Arduino uno.

6.2.1 Servo Motors

Use: To modulate the servos of the robot-dogs joints using Arduino's library <servo.h>

- START
- Read in the input stimulus from user or remote source
- Process signal using duration of pulse in ms
    - Obtain angle of servo based on inputted command
    - Write servo angle position to motor through Arduino servo.write()

- END

6.2.2 LCD Display

- START
- Clear what is on the display
- Input the value from external sources
    - If dog is being issued a command = Happy eye emotion
    - If touchpad senses robot-dog is being pet = Satisfied eye emotion
    - If dog is asked to idle/sleep = Sad or tired eye emotion
    - If dog recognizes owner/pattern/color shown = Ecstatic eye emotion
- Print the robot-dog's current emotion
- END

6.2.3 Speech Assistance Recognition

- START
- Read in input stimulus from user's voice signal
- Send to Alexa/Assistant API for voice processing
- Return to user response generated from Alexa/Assistant API
- END

6.2.4 Camera Object Detection

- START
- Connect to TensorFlow python file
- Loop to gather input signals from the sensor
- Send data to calculate distance
    - Distance = (Time x Speed of Sound) / 2. => duration*0.034/2
- END

6.3 Code Flow Chart

A code flow chart allows for an organized approach to programming and outlines the direction the software takes in its path to the outcome. In figure 37 we define the code diagram we followed in this project. The ovular shapes signifying the start and end of the program flow chart, the square shapes signifying an action the code takes, finally the diamond shapes showing reactions to previous events.

Code begins like most other programs by initializing initial conditions, setting up the timer, defining variables then setting I/O's for the various GPIO pin connected devices such as the sensor, servo motors, motor driver, camera, and the LCD. Once this preliminary setup is complete, we read the LCD and display the robot-dog's eyes. Once the sensor is read, we reach our first divergence; if closed we calculate the distance then reset the timer if open, we have a decision, if time is less than 255, we set the distance equal to zero else, we increment the timer. Next, read the distance if the distance equals zero stop else we set the motors and move in the given direction once location is reached, stop. Finally, we check the reset and if true we move to the second camera centric portion of the code flow.

In the second portion once check reset is true if image1 does not equal image2 and it is not a match redetect until the image matches in which case we branch to the else condition. If there is a match, then then we move towards the object. This concludes the scope of this code flow chart encompassing the initialization or various starting parameters, the LCD, sensors, manipulation of the internal timers, setting the motors, and camera image recognition.

This code flow chart concludes is the final coding plan as it encompasses much of what we considered from the programming perspective. In summation, our objective of this software comes to reading/displaying information to the user and controlling the motor which is done through many functions that are implemented into the Arduino Uno and custom code for recognizing and processing input from the 8 Megapixel camera to the Raspberry Pi.

The software environment that is used to compile and transmit the code is the Arduino Integrated Development Environment and Geany for the Raspberry Pi however multiple other software environments were used and considered when making this project including LTSpice, Lucidchart, and Autodesk Eagle. The Raspberry Pi uses the inputs it receives and integrates them into the virtual assistant AI software. Overall, the software helps control the sensors, motors and control/create the virtual assistance used within the robot.

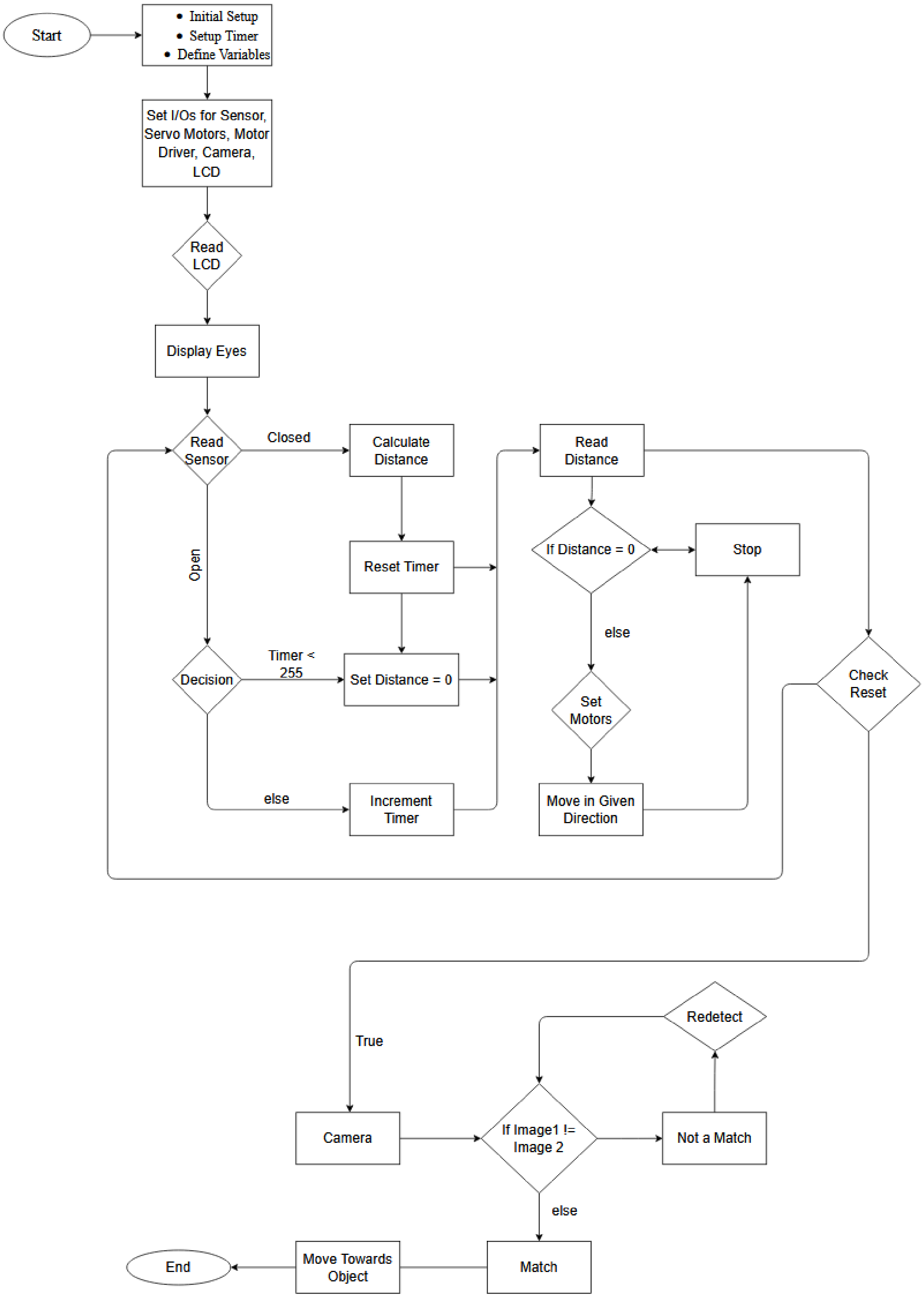

Figure 38: Movement flow chart

6.4 Signal Testing

Testing the software functionality is required in order to make sure that the tasks the programmer has outlined for the microcontroller and other parts of the system are being executed correctly. This testing was conducted on the boards we intend to use for this project. The ATmega328P was used to test the capabilities of the servo motors and touchpad sensors. The Broadcom BCM2711 was used to test the functionality of the LCD, microphone, and voice assistant. Testing the software of the sensor is already partly done with the hardware testing. In hardware testing, a simple software program is run to test all the sensors. The program is made to not strain the controller or the sensor. It is created in a manner that shows different scenarios that might be present in the sensor. For the software section, more complicated software can be made and tested to see how it reacts with the sensors. For example, with the ultrasonic sensor, the sensor sends a reading every second. To test the software, the timing of this pulse can be varied within the code and simple computations can be made with the values obtained. This assures that the software is picking up the readings from the sensor and that the program can also use these reading to perform other tasks. There are certain scenarios where this method of testing would not be idea. Some scenarios include PWM testing, LCD screen testing and testing the Mic Array. This happens because these signals are either more complex or require additional testing.

6.4.1 PWM Servo Control

Testing the PWM servo control is done by changing the PWM duty cycle and observing its effect on the different servo motors. Moreover, the servo motors can be connected to an oscilloscope and the respective duty cycle wave can be visualized. This way, it is apparent whether the servo motor is the issue of the PWM control being sent to the servo motor.

6.4.2 LCD Testing

To test the LCD, the LCD was connected to a microcontroller using a breadboard. Once connected, the test code made the LCD go through all the functions in its software library. Furthermore, the LCD program also makes the LCD go from darkest setting to lightest setting. This helped test the LCD hardware as well. This should be enough of a test to prove that the software of the LCD works.

6.4.3 Microphone Array Testing

To assure the software of the microphone array works, we need to test both the input and output of the array. Once the array is properly connected to the microcontroller, the inputs and outputs are tested. To test the inputs a digitally controlled sound was used. The sound was placed close to one microphone and the volume was varied. The reading of the microphone was then observed.

The same is done with the second microphone. Later, the digital sound is set at a medium volume and moved around the room. The reading of the sensors is observed to see if the sensor can pinpoint where the sound is strongest (left or right microphone). The reading given was used to test the software to see if it can accurately pinpoint where the sound if coming from. Furthermore, the sound was replayed using the speakers from the mic array. This also tested that outputting sound work effectively.

6.5 Additional Software

This section demonstrates all the different software used to help design this project. None of the software presented here is actually used on the robot itself; rather, it is software that has helped in the design of the robot. Some of this software helped with creating the model of the robot, while other parts of the software deal with testing the circuits virtually. This software heavily impacted the design and construction of this project. Some of the software, like Lucidchart, had no impact on the hardware design of the robot. Instead, it impacted the writing of this document or the software design. Without these technologies, the design would have looked very different

6.5.1 LTSpice

LTSpice is an excellent tool for testing circuits before construction. Simulating circuits is a good idea if the engineer is unsure whether the circuit works as they believe and need to see if the mathematical expectations hold when all the components are combined. It is also useful as a sort of sanity-check, if testing a physical circuit is not working as expected and seeing if it was an error in construction of the circuit or something more fundamentally wrong. Once the engineer understands that the theoretical circuit works, then the circuit components can be bought and tested on a breadboard. Simulations are useful in testing the theoretical function of a circuit. However, the circuit still needs to be tested physically as many approximations are generally done with circuits. LTSpice allows for the theoretical of the circuits to be done virtually. The program features many common components and its own virtual oscilloscope. It is perfect for simulating common circuits. This simulation can then be used when testing the physical circuit as a contrast. Simulations use theoretical or exact values in their results; they show how the circuit works in a perfect world. Physical circuits never act exactly like simulations. Normally, there are multiple deficiencies or inaccuracies in components and measuring tools that do not allow the theoretical measurements to occur in the real world.

6.5.2 Lucidchart

Lucidchart is an excellent program for creating diagrams and flowcharts. The program allows the user to use several of its features for free and with this version a user can store up to five different projects. The workspace available makes it easy to create long and complicated programs. The advantage Lucidchart has to Microsoft Word or Excel is that the program makes it easy to connect different components of the chart. The program even has some templates of how a specific chart should look. There are many options available as well. This program did not help with the design of the project. Instead, it helped with the planning, research, and creation of this document. The program allows for software diagrams to be easily created and readable. Flowcharts make organizing software programs a simple task. Furthermore, the flowcharts made with Lucidchart allow for several structures in the document to be better explained.

6.5.3 Autodesk Eagle

Eagle is an excellent software for printed circuit board (PCB) design. An engineer can easily create a circuit schematic in the program in a manner that is familiar to any electrical engineer, even on a student level as it is similar in practice to any SPICE circuit design program. It has an extensive built-in library of components, both common passive components and more complex active components. The popularity of the Autodesk Eagle software also means that many components on the market, such as those of Texas Instruments or Analog Devices, is available from the manufacturer as a package compatible with Eagle, making it convenient to use with whatever component is available at the time. After creating the schematic, Autodesk Eagle can then automatically generate a PCB with all the components used in the schematic. Once the board design is generated, the user is then able to easily move parts around and add mounting holes and other requirements needed for a PCB as they see fit, or if the automatically generated design had any issues. At any point after moving components around, the software can also generate PCB traces using user-inputted parameters and conditions. Additionally, the program can verify that the PCB works correctly and that there are no major issues in the wiring, either physically or electrically. Eagle is used extensively in designing the robot dog. All custom PCBs for the project are made using Eagle. The program provides a very simple and convenient way of creating PCBs and the entire team is comfortable and familiar with the program. Moreover, Eagle also provides a Bill of Materials (BOM) which can be linked to current availability from an online reseller and even has an option to buy all the parts directly in the program if necessary. This makes keeping track of inventory and organizing the purchasing of parts simple. Furthermore, if Fusion 360 (another Autodesk program) is installed, the used can observe the board in 3D. This is because Fusion 360 is an all-in-one CAD and circuit design software suite. This allows the team to observe the bulkiness of the board and confirm that the parts are the size we think they are. Eagle was chosen for use over Fusion 360 for PCB design because of lack of familiarity with the software among our team. Fusion 360 is a newer piece of software. Overall, our team had very little prior experience with PCB design, yet the ease of using this program makes PCB design a straightforward concept and facilities an efficient design process.

Eagle not only helped in creating our own PCBs, but it also helps analyze and edit the PCBs created by other people. There are several PCBs already created that people make available to the public to be edited. These PCBS were used by the team as bases and templates for creating our own PCBs. Eagle allows for simple editing and even identifying were different parts come from. Some companies keep their libraries public so that other users can copy their PCB design and use their parts. This was helpful when designing PCBs because it allowed the team to find the sizes required for certain PCB designs.

7. Integration and Testing

This section depicts how all the parts were tested and integrated onto the design. For a design to be successful, it is crucial that each part is tested so it works properly. Furthermore, each function of the robot must also be tested to assure that they work properly. Testing is done for the parts first, and then for the function of each. For example, the camera was tested on whether it can produce a clear image. Then it was tested for how well it works with the ultrasonic sensors. Finally, the entire system was tested to observe if it works well with the software. Speaking of which, software tests are also required to ensure there are no bugs or extraordinary events. This must be done for every part, system, and function of the robot. Generally, it is expected that a part works when coming out of the factory, but it is our job to ensure that the part work for its desired needs

Integration defines when and how the testing procedures were implemented during the design process of the robot. Furthermore, the integration section depicts when each part was added to the final prototype and how they were added. Integration is heavily tied to testing. The different components must first be tested individually before they can be integrated onto the robot. After individual testing, the components can be integrated into their respective subsystem and must be tested again. This is an example on how integration and testing work hand in hand to ensure the team can find any issues quickly and fix them before they affect any other parts of the project. Testing procedures, as the name suggests, describes all the procedures that are performed during testing. These sections also show the results of early testing.

7.1 Integration

It is crucial that each part of the robot is integrated after it has been fully tested. Therefore, integration plays an important role on what parts get tested and when they should be tested. For example, servo motors need to be tested individually, but it is also important that they are tested after they are integrated into the legs of the robot, and once again after they are integrated into the main body structure of the robot. The way the robot is integrated also play a role. Each part is built upon each other. The most important parts to be integrated first are anything to do with power all the circuits, especially the microcontrollers. If the brain of the robot does not work, then there is no point in continuing building the robot. Of course, to do this, the "skeleton" or plastic casing holding all the components must be built first. Once the plastic casing is created, the power supplies are mounted and tested to see if they work with the microcontrollers.

Before mounting anything onto the plastic casting, the plastic body must be tested first for rigidness and how it can handle weight. Moreover, the body was printed as multiple parts. These parts must be designed so that they can integrate perfectly with one another and not create a mechanical issue. Creating and integrating the different parts of the plastic architecture of the project is important as it is the structure that holds all the components.

The first parts integrated into the robot are the legs. Both articulations are joined together, and the servo motors are placed in their respective locations. Each of the limbs are tested and made sure they work properly. The legs are now ready to be integrated onto the main body of the dog.

Currently, the body is simply a plastic rectangle housing the power subsystem and all the PCBs. The legs are attached one by one until all four are integrated. Once again, the motors need to be tested to make sure they still have enough power and can support the weight of the parts that are already constructed from the robot. By this point, the first and second subsystems can be integrated onto the body and tested. Robot movement and power consumption can be tested now. Before integrating the first and second subsystem onto the constructed body, they must first be tested and constructed individually.

The next part planned to be integrated to the constructed body was the head. After consideration, implementing a head to the robot would make the robot hard to build so this was not included. Instead, a small head piece is used instead. Before the head is connected to the rest of the robot, the head is first built by adding all the sensors in their respective locations. The third subsystem must be built and integrated onto the head part of the robot. As mentioned, each sensor must be tested individually as they are being added to ensure they work properly. These sensors can now be tested inside the head to ensure they give accurate data and there is no issue with the design. Once all sensors are fitted and tested to work properly, the head is ready to be integrated onto the rest of the robot. Now with the head connected, the motors are once again tested with the new weight. Minor adjustments were necessary for the motors to do the same tasks they were performing before the head was integrated.

If everything works correctly up to this point, then all the components can be connected. By this point all the necessary components are functional and the robot basically works as intended. All that is needed is to place the remaining parts of the robot to ensure that all the electric components are secured and safe from damage. Additional tweaking of values and components was required and by this point the team is mostly troubleshooting the entire robot. If testing and integration is done correctly, the problems should not be major and should only be a result of the last integration done. All components are now successfully integrated, and the robot is complete and ready.

7.2 Testing Procedures

For testing, each individual part is tested separately to make sure they work properly. It is responsibility of the engineer to ensure that all parts work correctly. Once each part is tested, the parts are then be tested in their specific systems. If the systems work properly, then they can be combined into bigger systems and so on. Each subsystem is then tested individually before being joined together. The approach for testing is starting specific and then broadening out. The purpose of creating testing procedures this way is to make sure all the individual parts work properly, as well as make sure that they are capable of working together. The testing procedures for each individual part and system is described below.

Not only does each hardware component need to be tested, but software must also be tested and examined to ensure it all works correctly. Simple programs are used to test the software and hardware capabilities of the multiple sensors and components. Once these have been proven to work, the software can be modified to be more complex and test the different functions the components and software are required to satisfy. There are some parts of this project that are mostly software, such as the AI vision and virtual assistant integration. Responses from these systems are transmitted to the hardware and inputs are given by the hardware but results of these systems are heavily reliant of how well the software is designed. This explains why it is crucial that the software of these specific subsystems is tested to work as desired.

7.2.1 Hardware Testing

Leg Articulation:

The legs of the robot are very important to its overall function. Not only do they need to hold the entire weight of the robot, but they also need to do so while moving. Servo motors were individually tested to make sure they can hold the weight of the robot and have enough torque to move. The legs were tested to ensure full articulation and stability when moving. As a reminder, the requirement of the project is that the servo motors allow for a 180-degree movement for each motor. When the robot is designed, the weight each articulation used is considered, but it must be tested. To test, the robot legs can be articulated into different positions of varying angles, and then applying weight to them. If the robot is designed correctly, then the legs should be capable of holding the robot regardless of the angles.

Besides weight on the motors, the legs themselves must be tested to allow for smooth bending. The points of articulation must allow for smooth articulating and generate little friction. Without the motors, the legs should be able to easily bend and stretch. Furthermore, the legs must have some force to support some of the weight when they are completely straight. Like the motors this is tested by varying the weight in different parts of the robot and observing how the plastic and components react.

PCB and Circuit Testing:

Several PCBs were constructed to support the drivers and helping components of the sensors. These had to be tested before, during, and after they are connected to a power source. This is to assure they are functioning correctly. Before creating the PCB, it was necessary to first to test the individual circuits on a breadboard. Testing whether the circuit worked helped avoid unnecessary waste of materials if the circuit turned out to be useless. To test the circuits, a power supply is used instead of connecting the circuit to the power subsystem designed. This helps protect the subsystem from possible malfunctioning circuits. The image below (Figure 38) shows an example of how the power supply is connected to one of the breadboard circuits. With it, we can control the voltage or the current passing through the circuit and helped immensely with testing. It was very important that the voltage is kept within the datasheet specifications to avoid damaging the PCB.

Figure 39: Power Supply setup for Circuit Testing

The PCBs needed to be tested to ensure they work properly and to avoid damage to sensor or other devices that are connected to them. So as mentioned, before any PCB is created, the circuit itself must be tested. The connections on the breadboard weren't perfect, but the circuit was easy to change or replace when a problem arose. To test whether enough current or voltage is reaching a specific part, a multimeter can be used. If more detail is required, then an oscilloscope is better. The multimeter was very useful for testing current, voltage, and resistance readings. It is also useful in determining continuity of the circuit. Sometimes, it is better to observe how the voltage wave looks and observe if there are any peaks that are causing damages to the circuit. This was observable with an oscilloscope. The oscilloscope allowed the team to observe the input and output voltages of the circuit in a much more detailed manner. If the circuits are working correctly, there should not be big spikes in the graph and the voltage, or current should be staying at the levels expected from the team. If not, the readings from the oscilloscope can be used to detect where the problems are, and the team could fix it.

To prepare for circuit testing, surface mount components need to be soldered onto adapter boards for breadboard testing before ordering PCBs from a custom PCB manufacturer. An example of this is the TPS565208 buck voltage regulator is used in the power supply, shown soldered and ready for testing in Figure 39. While common components such as resistors, capacitors, and inductors are typically always available in a through hole format, certain diodes, voltage regulator, charging, and other integrated circuits are not always available in a through hole or dual in-line package for instant compatibility with a breadboard. This can be done by either hand soldering the components using a soldering iron or utilizing a stencil and hot air rework gun. The right tool for the job changed depending on the component package specifications, such as the number of pins or the physical dimensions of the package.

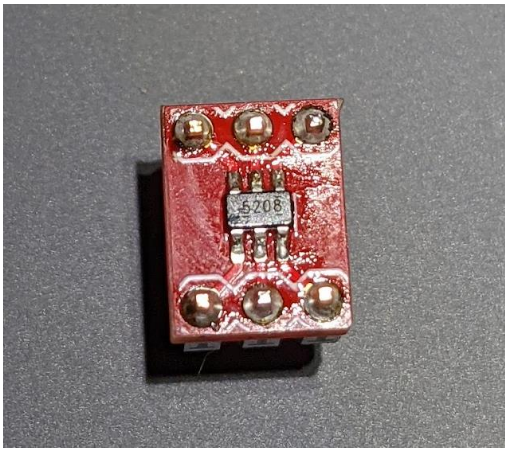

Figure 40: TPS565208 soldered onto SOT-23 breadboard adapter

Once the surface mount component has been mounted to an adapter board, it can be added to a breadboard along with the other through hole components of its accompanying circuit for testing. Figure 40 shows an example of such a testing setup, specifically the power supply which takes a voltage input from two batteries in series and regulates it for the Raspberry Pi and Atmega238P MCU. This breadboard testing allows for fast, cheap prototyping with a high level of flexibility. Breadboard prototyping means rapid replacement or reworking of circuit layout and design is possible and even convenient, before investing the time and energy of ordering a PCB with a potentially non-ideal design. Prioritizing breadboard prototyping in the very early stages of constructing the project streamlined the final PCB design process and reduce errors closer to the project deadline when time was valuable and scarce resource. Avoiding possible errors in anticipation of that late development stage is vital to the success of our project.

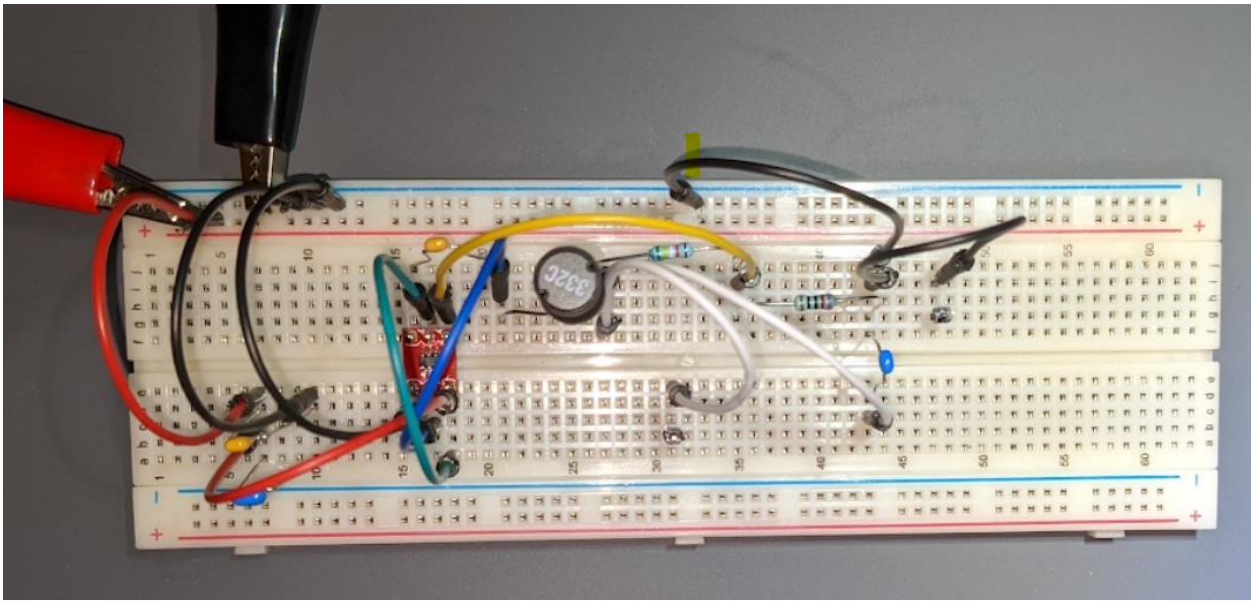

Figure 41: Power supply circuit assembled on breadboard

There were several issues after receiving final PCBs. In the power supply circuit, testing the buck converter-based PCB resulted in an overheated PCB and components which outputted much lower voltages than what was expected and needed. These voltage measurements were made using a multimeter. Further testing of these PCBs resulted in effectively zero voltage output, likely due to overheated ICs or “fried” components. This resulted in a need to pivot to new designs which were tested on a breadboard and then verified to work on a PCB in the final design, shown in the following figure. With the benefit of hindsight, a voltage regulator configuration with a higher voltage output would be more ideal, as voltage output was measured to be on the low end of a desired input for the servo motors. However, thought was put into keeping the circuit as physically and electrically simple as possible to avoid any possible PCB manufacturing issues and to reduce PCB versions, as time crunch became a factor with the increasing manufacturing, delivery, and supply issues that were only getting worse as the project timeline went on. PCB manufacturing issues also affected our team, with one implemented PCB stripping its own copper layer after plugging the battery into it. This was not a design issue, as the exact same PCB design was used again, and it worked without issue.

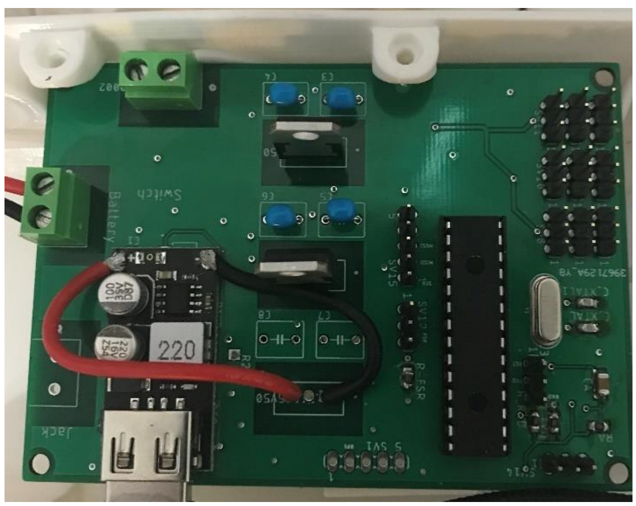

Figure 42: Final PCB with all subsystems integrated into the robot after testing.

The Arduino Uno is used to test the different sensors. The Arduino is programmed with several simple programs designed to test the circuit. Furthermore, it contains an integrated circuit that can easily connect to all the components used in the circuit. From the image, it is clear the Arduino Uno has a few LEDs that can be used to signal successful testing and a reset button in case the circuit does not work as expected. This microcontroller has been tested beforehand and is confirmed to not be the cause of any problems found during testing. To reiterate, it is crucial that the voltage and current levels are controlled do not damage any of the electronics, as this would involve increasing the cost of the project and potentially delaying integration of subsystems onto the project.

Figure 42 currently shows the cables used to connect a power source onto the microcontroller with a regulator circuit in between. The circuit is regulating 9 V to 5 V output. As can be seen on the image, the microcontroller appears to be working normally. To better prove this, the LED on the microcontroller is set to light green if the microcontroller is working properly and red if there is any issue. Beforehand, the regulator circuit was tested to ensure there was little risk in damaging the microcontroller when they were connected.

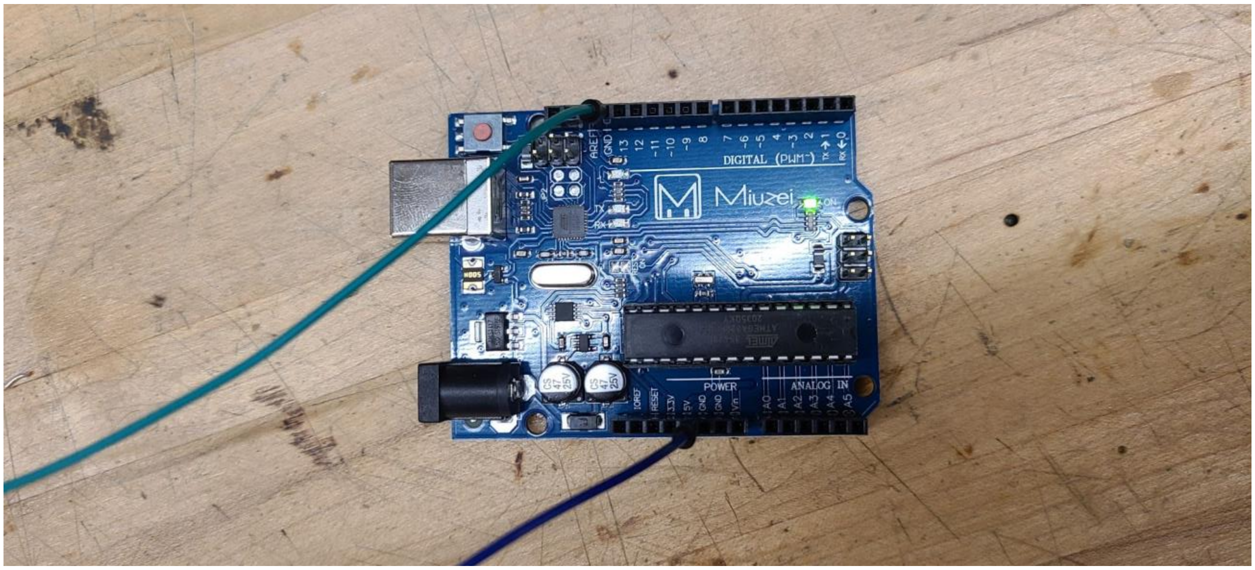

Figure 43: Connection of Arduino Board

Knowing that the circuit works well on the breadboard, we can know print the circuit on a PCB. The newly acquired PCB is also tested to ensure it works properly. If everything is soldered correctly, the PCB is created like the breadboard design, and the parts work as expected, the PCB should work like the breadboard circuit. The only way knowing this is to test it. To test the PCB, the voltages and currents of the PCB are compared to those obtained from the breadboard testing. If there are any spare parts, these can also be connected to the PCB to test whether it functions, without risking damaging any useful parts. This procedure ensures that all PCBs work properly, and that mistakes and errors are found as early as possible.

Servo Motors:

Each motor needs to be individually tested to ensure they can work efficiently and with maximum power. Unlike the leg articulation testing, this testing is exclusively the motors, without connecting them to the rest of the robot. To test the servo motors, they are connected to the Arduino Uno and programmed to perform certain tasks. The motors are programmed to move to different positions and their accuracy is recorded. It is expected that the motors work properly when they are new, but it is important to test them to avoid headaches in the future. Moving on, the motors need to be tested for how far they can rotate. This is easily tested by making the motor rotate to the maximum amount possible and then back. This also demonstrates if the motors can rotate both directions. Next, the motors are tested inside the legs of the robot. The test determined whether the motors have enough strength to bend the legs. Moving on, the motors were then be tested on how well they can handle weight and their torque capability. To test this, small weights are added to the legs and the motors are tasked to bend the legs once again. It is crucial to make sure that the weights are not excessive and do not damage the motors.

Touch sensor:

For the touch sensor, pressure and accuracy are the two most important components to test. First, the touch sensor needs to be tested to see if it works as intended. Like testing a button, the touch sensor is pressed multiple times and the time to notice the presses is recorded. Once the touch sensor is confirmed as functional, it needs to be tested to see if it can work under the shell of the dog. To test this, we take a piece of the shell and place it on top of the touch sensor; then we try to activate it. The sensor can then be calibrated with software to determine the right amount of pressure and sensitivity required for the sensor to work properly under the plastic shell. The sensor is also like a resistor, so the resistance of the sensor changes depending on the pressure applied to the sensor. The output voltage increases as the pressure increase. A multimeter can be used to measure the output voltage. Pressure on the sensor should change the voltage measured by the multimeter. If the sensor works correctly, then the multimeter should read similar values to those obtained from the datasheet.

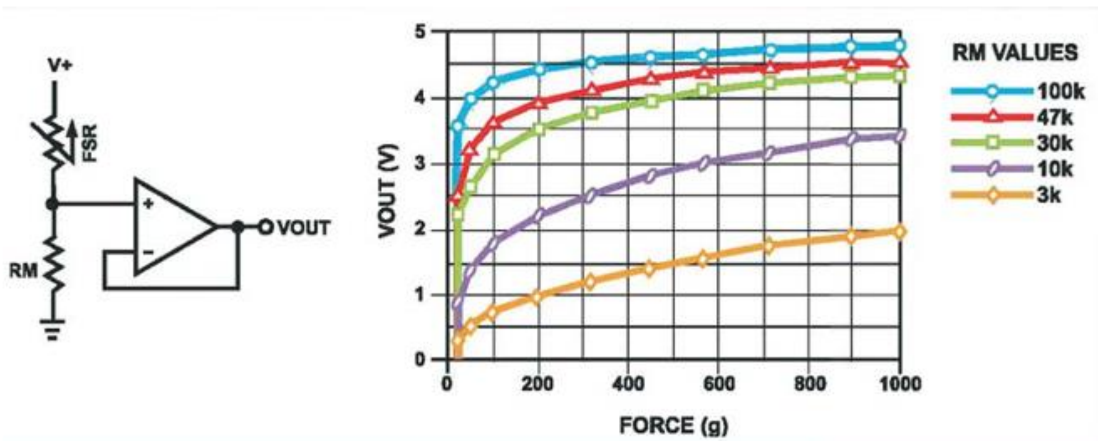

Figure 44: Touch Sensor Force vs. Output Voltage obtained from FSR sensor datasheet.

Figure 43 depict the expected force to output voltage conversion of the touch sensor according to the datasheet. The datasheet uses a theoretical 5 V input and shows the results for various voltage divider resistors. Testing is done with the 10k – 47k resistances and the same voltage input. This is done by changing the RM resistor and measuring the output voltage by varying the pressure. As can be seen, if the pressure increases, the voltage increases as well. This should also be seen with the testing preformed. This testing is first done with a power supply and then with the PCB board that is be connected to the sensor. This is to make sure there is no issue when using the PCB. If all testing is successful, the multimeter should measure around the same voltage outputs as those present in the datasheet (as shown in Figure 40).

Ultrasonic Sensor:

To test the ultrasonic sensor, it was connected to a microcontroller and using the IDEs provided console, its reading was observed. By this point, the microcontroller has already been tested and should work correctly. This makes the only dependent variable the sensor itself. Several readings need to be taken from different distances. The farthest distance it can measure and the closest distance it can measure are important to test as they are the limits of the sensor. Lighting can also be changed to find in which lighting the sensor works best. After the ultrasonic sensor is tested, it needs to be tested to see if it works with the software. This testing is best explained in the software testing section.

The following test was made at ambient lighting. The ultrasonic sensor is connected to the Arduino Uno. The software only tells the hardware to write the reading it receives. This is looped and repeated every second. The numbers on the far left are measuring an object approaching the sensor. The number in the middle is the sensor measuring an object going farther away. The numbers in the third row are describing normal readings from moving the sensor around the room. The image in figure 44 only depicts the test in which the object is approaching the sensor.

Figure 45: Testing Ultrasonic Sensor

The first column in figure 45 shows the measurements when making an object move closer to the sensor. As can be seen, the sensor can only read object until they are 2cm near it. After this, the sensor cannot accurately tell the distance. This goes in line with what the datasheet provides. It is important to note that a measuring tape is also used to make sure the reading is accurate. As the flat object is moved towards the sensor, the tester is observing the actual value as given by the rule. This test was run multiple times and the sensor was accurate every time.

The next test involved testing how far the sensor could read. Using measuring tape, the team assured that the sensor was providing accurate results as the sensor was moved away. From the second column in the figure 45, we can see that the sensor starts failing to give constant accurate results after passing 418cm. After performing this experiment multiple times, the average distance this occurred for this sensor was calculated as 420cm; in line with what the datasheet specifics. The sensor appears to read even farther but the results are no longer precise and should be considered unreliable.

Finally, the sensor was tested on whether on normal conditions. The sensor was moved around and observed on how it reacted to different elements. Precision was mostly measured in this test. For example, the sensor was pointing at an object measured 41cm away. The sensor is then quickly moved to another object measured 58cm away. The sensor is then moved back to the original object and measures again 41cm. Should the sensor have provided a different result, then we know the sensor is not entirely precise. Like the other test, this test was repeated multiple times for consistency.

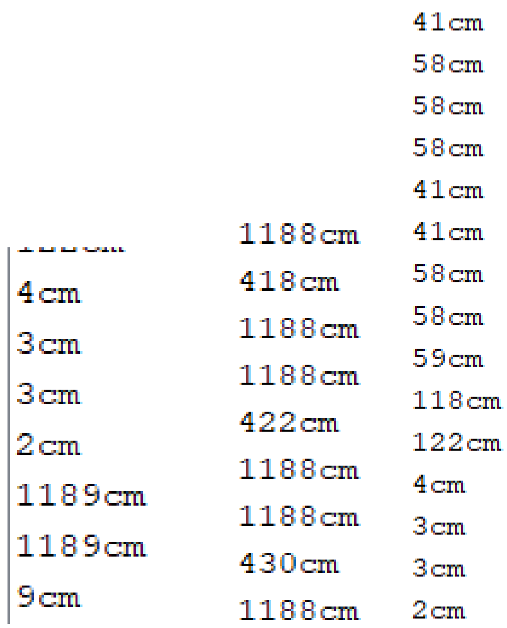

Figure 46: Reading from ultrasonic sensor test

After viewing the results of the multiple tests performed, it is clear that the sensor works perfectly and are indeed useful for its function in the robot dog. The sensor is capable of detecting objects within the datasheet specified range of 2cm -420cm. The sensor also works in all lighting which is perfect for aiding the camera. Although only one of the sensors was planned to be used on this project, they are cheap so a couple more were bought. These were also tested the same way and provided similar results; hence their results are not depicted on this document.

Camera:

Testing the camera is simple, all that is required is to connect it to any microcontroller and observe its output. The output must be like the one expected from the datasheet. If the camera works properly, it is then tested with its respected software and seen how it responds. The camera must also be tested with its corresponding PCB to make sure it receives enough power, and it works properly. The PCB should also be tested independently before doing this.

LCD Display:

The LCD display was be tested by connecting it to one of the microcontrollers and displaying different object on the screen. The test objects should vary in size, and some should occupy the entire screen. The objects should also test all the different possible colors. The screen brightness should also be tested by observing how well the screen appears in different lighting settings. If the LCD screen works properly, all the images should be shown perfectly.

As mentioned in the previous sections, the LCD display was planned but not included in the final project due to the unforeseen complexity surrounding including and intertwining an LCD with the other systems. Showcasing emotions which tie to certain actions taken by the dog was seen as an undertaking we would like to keep as a possible future upgrade requiring more time and modification to the robot head STL file.

7.2.2 Software Testing

There are multiple ways and programs to test software. The simplest way is to brute force multiple scenarios into the program and see how the program reacts. This method works well for simple programs but became very difficult for more advanced software. When using software that is already created, such as the AI assistant API, there are generally example commands and codes that can be run to test whether the code is functioning properly. This concept can be also applied to new software programs. The idea is to have certain preset scenarios and test whether these scenarios are working properly as the software gets updated.

The image below shows the connections used for testing the Voice Assistant technology. As can be seen in figure 45, only two cables are required, which connect to the microphone of the device. The Raspberry Pi also contains a USB port that is used to connect it to the PC for testing. Later testing used a 3.5mm AUX jack speaker and the final product used a USB-A microphone. With the finalized SDK testing was done with the aforementioned 3.5mm speaker and USB microphone by repeatedly activating Alexa's listening state and asking it questions of different type to analyze its different functions.

The question: "What is the weather like today?" Allowed us to see if the Alexa SDK had proper access to our location services as well as it's connection to weather services. The next testing statement allowed us to test the Alexa SDK follow-up question multi-turn interaction: "Set an alarm for X time" without specifying AM or PM made Alexa ask up the follow-up question: "Is that AM or PM?" after choosing an option waiting to see if the alarm was set and if it would activate the speaker correctly. Other basic trivia questions we asked to make sure it was properly searching the web for correct answers such as: "What was the first president of the United States?" and "What year was Abraham Lincoln inaugurated as president?" Further details and official Amazon developer documentation for verifying functional requirements are detailed in a section below.

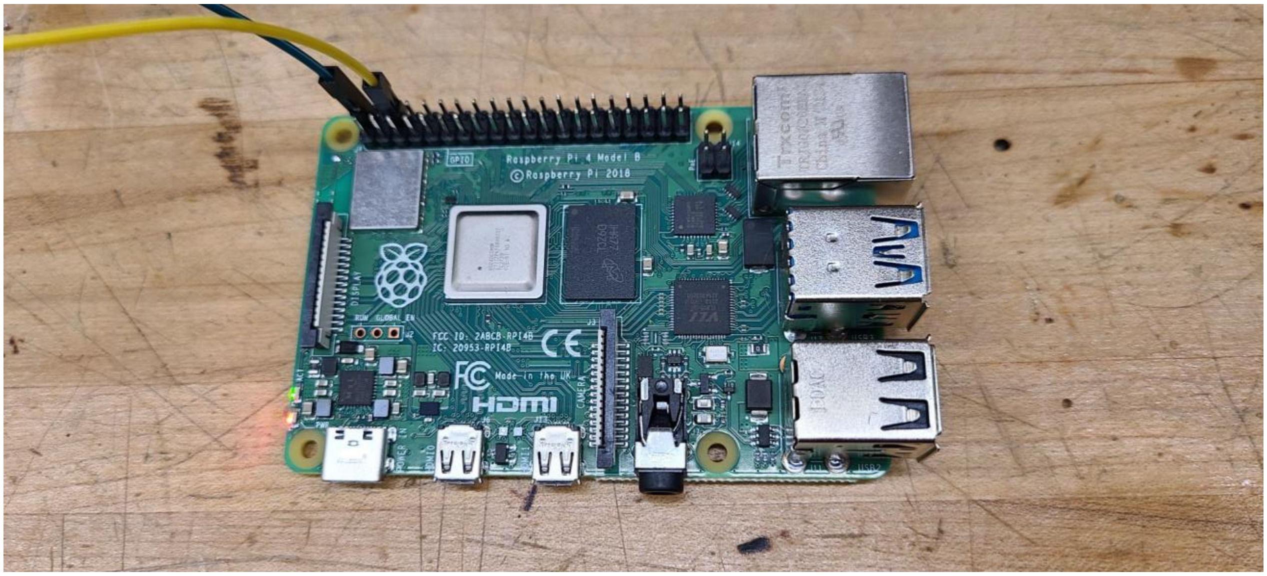

Figure 47: Setup for Voice Assistant Software Testing

Objective: The objective of this test is to ensure that the software is completing and processing the coded task successfully.

Environment: Most of the testing for the software portion of the prototyping was at UCF's Senior Design Laboratory. Personal computers (Windows and/or Linux) running the Arduino IDE and the Raspberry Pi was used to monitor and edit the program code implemented on the microcontroller.

Procedure: To test the software portion of the robot-dog, procedural steps we as followed:

1. Power on the processors and wait for ready state confirmation. The display should show the dog's eyes and the speaker should play a bark noise.
2. Select the correct port on the Arduino software IDE.
3. Send a sweep command to each individual servo motor.
4. Observe motors going from 180 degrees to 0 degrees.
5. Send second sweep command to each individual servo motor.
6. Observe motors going from 360 degrees to 0 degree.

PWM Testing

1. Using a test PWM function, connect the oscilloscope to the desired output terminal were the PWM signal is generated.
2. Set the frequency of the PWM (within the code) to 8 kHz and set the positive duty cycle to 50%.
3. Observe the waveform on the oscilloscope. This waveform should be a PWM with a HI 5V amplitude for roughly 62.5 microseconds and a LO 0V amplitude for 62.5 microseconds.
4. Make sure to measure the frequency of the waveform to match closely to 8 kHz.

Voice Assistant Testing

We followed Amazon Alexa's developer documentation for verifying functional requirements of the onboard voice assistants.

1. Test that the voice assistant API can receive audio input (i.e., gathering user speech via the microphone(s)).
2. Confirm that the voice assistant properly recognized voice input and outputted proper response. Ask the assistant to repeat its last command to ensure proper sentence recognition.
3. Test the voice assistant is capable of audio output (i.e., interfacing with the onboard speakers).
4. Voice assistant must have action buttons capable of interfacing with the voice assistants' current actions. Press action button to go previous, pause, next or other actions related to voice action.
5. Showing prominent visual cues when assistant is in attention states (i.e., Listening, Thinking, and Speaking).
6. Test multi-turn interaction with conversation including subsequent question (i.e., "Remind me to take out the trash", desired response: "At what time would you like to be reminded."

7.2.3 Voice Assistant Certification Requirements

For a product to be voice assistant certified it must satisfy 2 main requirement categories. The categories proceed as follows:

Functional Requirements

The functional requirements to be voice assistant certified are summarized with the following:

1.1. Must be capable of audio input

1.2. Must be capable of audio output
1.2.1. Provide device controls for adjusting volume

1.3. Robo-dog must have action buttons
1.3.1. Button must initiate voice interaction
1.3.2. Button must be able to interrupt output
1.3.3. Button must be easily accessible

1.4. Robo-dog must clearly convey attention states to the user
1.4.1. Prominent visual cores and/or prominent audio cues
1.4.2. If using both audio/visual cues must be synchronized

1.5. Support multi-turn interactions

1.6. Support silencing alerts, adjusting volume, stopping media with no internet connectivity.

Security Requirements

1.1. Robo-dog must use a secure software update distribution that uses cryptographic signing so that only authentic and authorized updates are applied to the Robo-dog.

1.2. Robo-dog must implement industry standard device hardening methods. For example, prohibiting default passwords, removing unnecessary network services and software, validating inputs before processing it in services on the Robo-dog, and applying all security patches to vulnerable open-source software.

1.3. Robo-dog must use TLS 1.2 or greater for all communications to endpoints outside of initial setup. The Robo-dog must implement certificate validation for all such TLS connections and must validate those connections to the Robo-dog are signed using the correct certificate.

1.4. Robo-dog must support Secure Connections when using Bluetooth BR/EDR or Bluetooth Low Energy (BLE).

1.5. Robo-dog must support Security Mode 4 Level 4 when using Bluetooth Low Energy (BLE) or Bluetooth BR/EDR protocols and services.

1.6. Robo-dog must support Security Mode 1 Level 4 when using Bluetooth Low Energy (BLE) protocol and services.

1.7. Robo-dog must use the Privacy feature when using the Bluetooth Low Energy (BLE) protocol.

1.8. Robo-dog must protect local Amazon software from unauthorized access. For example, on-Robo-dog MiTM attack or display hijacking.

1.9. Robo-dog must implement a hardware based on/off control mechanism for any microphones and cameras. This control must remove power from the microphones/camera and include a dedicated microphone/camera status indicator to inform users of the on/off status.

1.10. Robo-dog must use a chipset that relies on hardware-based security capabilities and meets PSA certified Level 1 or similar.

1.11. Robo-dog software components and use of Amazon SDKs must not violate the license terms of the SDKs.

8. Administrative Content

This section of the report discusses two main aspects of our project that helped manage our two main resources, time, and money. Completion of Senior Design 1 completed the first set of milestones, split into two main sections. The first being the prototype document of the design. This document was made to help understand what was required to build the final project. Many hours of research and designing were put into this initial document, in order to create a successful prototype in the second section. The second section involved building the actual prototype after following the requirements, standards and constraints presented in the initial document. This section of our final document discusses milestones and the budget analysis of the project. The monetary decisions are written in this section as we actually encountered and made them throughout the process of building and are properly enumerated here after completion of Senior Design 2.

8.1 Milestone Discussion

The initial project milestone for the first semester is primarily putting together all the necessary paperwork that is needed when building a project. For us to pitch the project idea to someone who is interested, we need to not only have a working project, but the paperwork to back it up. This paper takes the form of a 120-page final project documentation. Additionally, these documents help guide us as engineers to have working knowledge of all the hardware and software components in use for the project. To create this paperwork, we needed to do extensive research on the technology available, parts that could potentially be used, and other projects similar to this one. This research helps create the project that we desire while also keeping cost, efficiency, and reliability in mind. By the end of the first semester, we should have a clear idea on how to build the prototype which was the milestone of this semester.

The initial project milestone for the second semester is to use all the components listed in the report and the research gathered in the first semester to start building a project and have a working prototype to demonstrate to people. To fully accomplish this milestone, we want the project to be actively responsive to the user and autonomous in its avoidance of objects and recognition of people. The robot should act like a dog while also using its capability as a smart device to be a virtual assistant. Thus, making a smart robodog. These milestones are outlined in Table 28.

In any project, milestones are important to keep in mind and follow through on in order to ensure that no parts of the project slip through the cracks while in the midst of planning or building. Coming up against deadlines and rushing just before them can create issues in any project at any scale. The last thing any engineering team wants is to rush and risk creating lower quality work than would otherwise be done. So, milestones in both time and substance are vital to the engineering process and allow engineers to keep the final outcome of a project in the forefront while simultaneously paying attention to the intermediate steps in between. This project is no different. While many specific dates aren’t currently knowable, being aware of known unknowns is also important and should be accounted for in the milestone discussion.

Table 29: Milestones

Milestone Number	**Task**	**Date for Completion**
Senior Design 1		
1	Idea	
2	Project Selection and Role Assignment	
3	Report	
4	Divide and Conquer (Initial Document)	9/17/2021
5	Divide and conquer v2	10/01/2021
6	Table of Contents	10/08/2021
7	60-page Draft	11/05/2021
8	**Research, Documentation, and Design**	
9	Servos	10/01/2021
10	Micro Controllers	10/01/2021
11	Code Structure	10/01/2021
12	Power Supply	10/01/2021
13	LCD	10/01/2021
14	Touch Sensitive Surface	10/01/2021
15	General Structural Design	10/01/2021
16	Document Review Meeting (w/ Dr. Wei)	11/08/2021
17	Part Order	11/08/2021
18	100-page draft	11/19/2021
19	Final Document	12/07/2021
Senior Design 2		
20	Assemble Prototype	3/22/2022
21	Testing and Redesign	4/05/2022
22	Finalize Prototype	4/12/2022
23	Final Presentation	4/19/2022
24	Final Documentation	4/27/2022
25	Peer Report	4/27/2022

8.2 Budget Analysis

When conducting research on similar products we discovered that many systems with comparable functionality are in the cost range of $1000-$3000. These projects tend to have best of their technology with a lot of resources spent on the software and design aspects of the project. Out design is meant to be much simpler while using some of the ideas already created by these large companies. Thus, by choosing parts that can perform the desire job, while also being cheap enough, we can create a robot dog that fits in our hypothesized budget of $500.00. When compiling a list of initial necessary parts for this project we calculated an estimated cost of materials to be $380.07.

This cost of this project changed due to the constraints mentioned previously. The scarcity of parts, and the potential change of part selection after further consideration are some of the reasons while the cost change occurred. However, the original estimate leaves to a large enough margin for the budget requirement to still be met.

From table 29, it is clearly shown that the most expensive components of the robot dog are the motors. Each motor individually it not outrageously expensive but combining all the necessary motors adds up in cost. The next expensive part are the microcontrollers which are crucial for designing the software and prototyping the robot. These are not cheap components, but they provide so much utility that they are worth it. Moreover, as was seen in the part selection section of this document, these components are also some of the cheapest among their competitors.

The cost of the plastic casting around the design relies heavily on the cost of raw materials and the size of the robot. All these aspects are heavily determined by the constraints of what is available and the amount available. This is something very volatile and cannot be easily calculated or predicted. However, the margin in our budget should be more than enough to cover this expense.

As mentioned, the budget analysis is not entirely accurate, and some parts changed. Despite this, the budget analysis given provides a good estimate of what the total cost was. It’s beneficial to understand the room available for changes in pricing. As mentioned in the constraints, several factors are changing prices and availability of products. Although most components are bought early one, some may prove to be useless or better options might be necessary for the robot to function properly. Therefore, this budget analysis is not concrete and is set to change.

Table 30: Cost Estimation

Part name	Description	Quantity	Cost
MG90S mini servo motors	2 motors used for each leg; one for the knee and the other for the shoulder.	18	(6 x $20.99) x 3 = $62.97
Raspberry Pi Camera v2	The camera was integrated near the eyes for use of AI vision.	1	$27.00
Repeaker Pi HAT 2 mic-array	Used to receive audio input for commands. Also includes a simple speaker for audio output	1	$9.95
Gravity Digital Speaker Module	Located where the ears would be. Gives output audio. These are optional and used if the 2 mic-array speaker is not good enough.	1 or 2	2 x $6.00 - $12.00
PCB components and parts	Used to replace microcontrollers in final build	several	$39.62
Arduino Uno R3 Microcontroller Board.	Used to test the MCU that is controlling the peripherals such as motors and touch sensor.	1	$44.90
Atmega328p MCU	Goes inside and controls peripherals such as motors, touch sensors, etc.	5	5 x 2.87 = $14.35
Arduino Breadboard Basic Kit	Kit include an Atmega328p MCU, a 16Mhz clock, an LED, and several resistors and capacitors required. Used for testing the circuit on a breadboard.	1	$19.99
Sparkfun FTDI FT232RL Breakout Board	Breakout board used to program the Atmega328p while is outside the robot.	2	2 x 12.95 = $25.90
5000 mA 7.4 V Battery	5000 mA 7.4 V Battery	1	$67.00
2S Li-ion battery charger BOM	Connected to battery.	1	$20.00
Raspberry Pi	“Brain” of the robot	1	$79.99
Plastic Body	3D printed body		Cost of raw materials
Gravity Gesture/Touch Sensor	In the back for “petting detection”	1	$16.99
90 Degree USB a to USB C	Power from PCB to Raspberry Pi	1	$9.04
Gorilla Super Glue Gel	Adhering motor horns to body components	2	$180
VELCRO Tape roll	Adhering battery to underside of body	1	$7.47
Rubber Feet	Attaching to bottom of feet for traction	1	$6.99
Total Amount			$474.46

8.3 Project Design Problems

Multiple designs were considered when creating the design of the robot dog. At first, the main design was concentrated in creating a robot that resembled a dog. The size of all the components was neglected as well as their weight. This created an initial design that was simple and not very useful. Furthermore, this design left unclear which motors would be useful and how everything would fit together. The team soon realized that in order to design everything properly, every system and subsystem needs to be taken into consideration. This led to the first design problem: movement. When it came to moving the robot, it was decided that the robot won't use wheels to move. This was not the case in the beginning of the project design. At first, wheel was preferred because they allowed the robot to move easily as well as have no difficulties when it came to programming. The main issue of this design is that it did not meet the appearance requirement set early on. The wheels made the robot look more like a car that a dog, so they were discarded. Instead, eight servo motors are used to move all four legs of the dog, allowing it to resemble the walk of a dog. This design also led to many problems, but more software related. It is complicated to select the correct angles and steps to make the robot move forward and backward in a dog-like manner. These problems only require more time to solve and no changes in the overall design.

Another problem in the design was weight consideration. At first, the robot was expected to weigh very little, but the batteries, microcontrollers, and several motors made the robot too heavy to control with the cheap motors selected at first. After more research and knowledge, it became apparent that metal-geared servo motors were better for the job. The issue was not necessarily the weight of the robot but the robustness of the servo motors. The original motors used plastic gears in their gearbox which created a lack of rigidness. The motors would not stay firm and would constantly vibrate. To solve this, metal-geared motors with even better torque were used. These motors solved most of the problems and made the robot dog more durable, a key requirement of the project.

A third design problem involved cable management. There was little consideration to how the cables would be managed. When the project was first drafted the team expected much less cables. As the design unfolded, the team soon realizes that many components not only required several cables, but also several cable expansions. Furthermore, the number of PCBs was also much greater than what the team expected. Our initial design was focused so much on weight and size that it did not comprehend the amount of space that would be required to hide all the connections between the various components and PCBs. Since then, the design is updated to have a large body to store all the extra componentry and make the robot appear like a dog, an important requirement of the project. This type of flexibility in design changes came up again and it was necessary to adapt as these sorts of issues and considerations come up later on.

9. Project Summary and Conclusion

This project seeks to cater to users with an alternative to live animal emotional support companionship. The number of people regardless of age, old and young, are suffering from anxiety, depression, loneliness, and stress in the home, with not many alternatives to traditional cats or dogs. Besides the emotional benefactor of our project a robot-dog is also a monetary proposition. Emotional support animals require time and money to take care of. This financial burden can become overwhelming or render the possibility of attaining or caring for the dog unattainable for many people. Additionally, some people in need of these animals have allergies or a family member with allergies that do not allow obtaining an emotional support animal. Therefore, an interactive robot that mimics the basic behavior of an emotional support animal, tied with the functionality of a virtual assistant such as Amazon's Alexa™, could provide a sense of companionship without the need for continuous expenses.

This robot can bring, not only companionship, but several other qualities that make it unique. The robot was designed with the idea of being more than just a robotic dog. The robot has many features like touch sensing, hearing, speaking, and many other features than make the robot-dog feel real. The dog listens to custom commands, interact when it is petted, etc. It also has different poses and expresses different emotions. The robot is made with the purpose of being a realistic dog, while also providing new features. With a virtual assistant, the robot can communicate and interacting with several smart devices. It can also do anything any other smart device can do.

As we developed and designed the robot-dog we approached and overcame many obstacles in its hardware and in its software. The creation of a robot-dog is not unique and is a project that has been executed many times to varying degrees. We wished to create not just something emotional but also useful bringing the robot assistant to the modern era with onboard speech recognition and Alexa/Assistance support. To create a companion a family can, find utility and joy with.

10. Appendices

This section is used to show the bibliography containing all the material that was referenced throughout this paper.

10.1 Bibliography

An introduction to OLED displays. (2021, October 01). Retrieved from oled-info: https://www.oled-info.com/oled-introduction

Arduino Uno R3. (2021, June 12). Retrieved from Arduino: https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf

Functional Requirements for AVS Products. (n.d.). Retrieved from Alexa Developer: https://developer.amazon.com/en-US/docs/alexa/alexa-voice-service/functional-requirements.html

International Standard. (2017, February). Retrieved from ITEH: https://cdn.standards.iteh.ai/samples/63161/f8613383f0324aec903f47cf35af5319/ISO-IEC-30122-2-2017.pdf

IPC J-STD-001 Standard Soldering Requirements. (2021, September 23). Retrieved from Sierra Circuits: https://www.protoexpress.com/blog/ipc-j-std-001-standard-soldering-requirements/#The_joint_industry_standard_for_space_applications

LCD Basics. (n.d.). Retrieved from J Display: https://www.j-display.com/english/technology/lcdbasic.html

NNAMC1580PC01. (n.d.). Retrieved from Digikey: https://www.digikey.com/en/products/detail/neonode-inc/NNAMC1580PC01/7776715

Raspberry Pi 4 Model B. (2019, June). Retrieved from Raspberry Pi: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf

ROBOMECH Journal. (2020, June 25). Retrieved from Springer Open: https://robomechjournal.springeropen.com/articles/10.1186/s40648-020-00174-1

Square Force-Sensitive Resistor. (n.d.). Retrieved from Adafruit: https://www.adafruit.com/product/1075#technical-details

Standalone Toggle Capacitive Touch. (n.d.). Retrieved from Adafruit: https://www.adafruit.com/product/1375

Standards & Certification FAQ. (n.d.). Retrieved from ASME: https://www.asme.org/codes-standards/publications-information/faq

Statista. (2020, April). Retrieved from Statista: https://www.statista.com/statistics/973815/worldwide-digital-voice-assistant-in-use/

Statista. (2021, June). Retrieved from Statista: https://www.statista.com/statistics/1274398/voice-assistant-use-by-device-united-states/

t4. (2021, January 23). Retrieved from Voice Assistant Market Share: https://www.t4.ai/industry/voice-assistant-market-share

TPS65-201A-S . (n.d.). Retrieved from Mouser: https://www.mouser.com/ProductDetail/Azoteq/TPS65-201A-S?qs=pfd5qewlna5Lh8O0E8DcUQ%3D%3D&mgh=1

Types of Wheels Used in Robotics. (n.d.). Retrieved from Robotics Shop: https://roboticsshop.net/types-of-wheels-used-in-robotics/

Wheeled Robots. (n.d.). Retrieved from Robot Platform: https://www.robotplatform.com/knowledge/Classification_of_Robots/Types_of_robot_wheels.html