“Pawsitive” Pet Feeder

Senior Design II

Senior Design II - Final Report

University of Central Florida

Department of Electrical Engineering and Computer Science

College of Optics and Photonics, CREOL

Dr. Wei Lei
Dr. Samuel Richie
Dr. Aravinda Kar

<u>Group 8</u>

Isabella Pardo	Photonic Science Engineering
Ervin Dupuis	Computer Engineering
Jocelyn Ignacia	Computer Engineering
Ayush Shashikant Pindoria	Computer Engineering

Table of Contents

Table of Figures

List of Tables

List of Equations

1.0 Executive Summary

In today's busy world, pets have become an integral part of families, owners are becoming increasingly concerned about their pets' health and well-being. With their owners busy with work, pets around the world are suffering due to the sudden change in everyday dietary routines. This growing health concern has resulted in a surge of products in the market, as a result, the demand for pet feeders has increased. There are many types of pet feeds on the market today that integrate the use of electronics and automation dispensing of food. This product helps save time, energy, and stress that is put on the owner's daily work routine when they are away from their pets and home. Devices that can automatically feed pets already exist in the world, however, most of the time these feeders rely on simple timers and fail to feed several pets at a time, or even different types of pets at the same time. The use of optics and electrons can be used to ensure that these pets are being fed routinely and with the correct dietary amount. This device will be able to pet three different pets, which is not common in the market. Most pest feeders have typically one or two bowls. Also, what makes this design different from buying multiple pet feeders is not only the comfort of having just one device, but the innovative features we will be including in this pet feeder by being able to connect to a phone application, this will be what differentiates our products from others that already exist in the market today.

The product that we will be creating will consist of three separate components, these parts will ensure the proper feeding of the user’s pets. The first part is the collar tag.

In order to create an intelligent pet feeder that could feed multiple pets, we created a collar tag that will be worn around the owner's pet. The Pet food is dispensed to that pet based on the identification tag collars that are read by the feed. To do this, an LED light embedded inside the tag will emit a specified color of light, and the pet feeder will recognize the color and dispense the correct food for the specified pet. This function is especially useful for households that have more than one pet because the second or even third pet could be using a similar collar tag that emits a different color and the pet feeder will be able to recognize that it is not the same pet.

The second main component is the automatic food dispenser. This device will consist of a camera that will be used to monitor and search the areas for the predetermined color of light that is illuminated from the tag collar on the pets. There will be a big bowl with 2 divisions, separating the bowl into three. Once the food has been dispensed to the pet’s bowl, the lid that covers the bowl will lift whenever the corresponding pet is within the area of the pet feeder. Throughout the bowl, infrared LEDs will provide a continuous signal to the photodiodes. When the pet starts to eat from the bowl, the photodiodes will be blocked, thus causing the lid to remain open. When the pet completely removes its head from the bowl the detectors regain the signal from the infrared LEDs and a countdown timer starts for the lid to start closing.

The third and last component is the application on the smartphone. Being able to remotely control the feeder is an important aspect of this device in case the owner forgets to put on the pet's collar tag before leaving to work. This ensures that the owner does not have to worry about leaving the pet home alone, as they can feed them automatically and manually. The application was created for the purpose to make owners' lives simpler now that there is an automatic feeder feeding their pets. In this application, there will be a live feed on what is shown on the camera connected to the pet feeder, this way owners can see what is surrounding the pet feeder. Through their microphone built into their smartphone they can talk to their pet using the speaker that is connected to the device. There will also be a button that allows the owner to refill the bowl dispense more food into the bowl, and receive a notification when the pet is eating or if the food storage is low in food, this way the user does not need to occasionally check the food storage.

2.0 Project Description

The "Pawsitive" Pet Feeder is a new type of pet feeder that uses both optics and electronics to automatically feed the user's pets whenever they are not around. Instead of worrying about getting home in time to fill a bowl, the owners can breathe easy knowing their pets are not being neglected.

2.1 Project Background

In households, pet owners are responsible to ensure that their animals are being taken care of every day, these animals need to be fed accordingly to ensure that they get their daily nutrients. The portions given to the pets depend on a lot of factors such as Breed, Age, Gender, Bodyweight & Composition, and many more. These details are easily neglected from unforeseen events such as rushing to work or getting distracted from tasks around the house. When this happens, the owners tend to overfeed the pets, and this leads to obesity that negatively impacts the animal's health.

Another common issue, within a household that contains multiple pets, is ensuring that all the pets are equally fed, or fed according to their size (for example big dog and small puppy). Often, whenever an owner is trying to feed multiple pets, all the pets start rushing toward the bowls and fight to dominate each other's food. This causes an imbalance of portions when trying to ensure they consume enough daily nutrients.

In situations where the pet is sick, the doctor gives medication or prescribes specific food for the pet to consume to ensure they regain health. This can be dangerous if another pet gets a hold of the sick pet’s food. Therefore, the owner needs a way of ensuring which pet is eating from which bowl.

Lastly, canned food is a great alternative to ensure a hydration boost as this food usually is portioned and contains a well-rounded diet that targets all kinds of needs. However, often if the pet is not quick enough to finish the food, the moisture evaporates causing the remaining food to harden. This makes the pet become a picky eater, and they will be less inclined to eat their meal. For that reason, the design of the pet feeder consists of a lid, which will ensure that no bacteria can grow within the food, meaning the food will stay fresh longer.

An automatic pet feeder can help manage the stress owners deal with by dispensing the proper amount of food at the same time each day, ensuring that their pets are taken care of. With this, the owner will only need to focus on refilling the dispenser whenever the food inside is low. And changing the default setting of the pet feeder to ensure it matches the owner's personal dog, these changes are, the amount of food to dispense, time set to dispense food, etc. The design we have planned out consists of the ability to feed multiple pets from one device, this also means that the feeder can feed up to three different pets of the same type (up to 3 cats or 3 dogs). The project will consist of motors, gears, belts, cameras, sensors, and WiFi functionality to ensure that these complex requirements are fulfilled.

The specialty of this project is taking the principle idea of a basic automatic feeder, and adding optical components that allow the system to identify the different pets (Pet A, Pet B, Pet C), allowing the system to dispense the correct amount of food to the specified pet. This system works due to the addition of the collar tag, by the addition of color LED lights, and the camera that will be attached to the pet feeder will be able to detect and label Pet A, Pet B, and Pet C. The owner will need to assign the pets different color tags to ensure no confusion within the system. Whenever the camera can pick up the color within a specified range distance, the predetermined amount of food will be dispensed, and the lid attached to the bowl will be lifted. Allowing the pet to eat from the bowl.

The addition of the lid will guarantee that the Pet does not eat each other's food. Therefore, the system will only open the lidded bowl of the assigned pet. In order for this lid to open and close, we will be using (IR) LEDs and photodiodes that will best detect IR light built within the pet’s bowl.

2.2 Motivation

In the modern world nowadays, 70% of U.S households (90.5 million homes) own a pet as of 2022. In which 69 million U.S. households have a pet dog, compared to 45.3 million with a cat. With families busy with their own career lifestyle, often these pets get neglected. Such neglect can result in pets illness, starvation and through extreme cases this can lead to the death of the pet. The pet owners then tend to either underfed these pets due to neglect, or overfed them due to trying to make up for the times they have forgotten to feed their pet. Both the cases as mentioned lead to pets misery and poor health. Therefore, this was the main key motivation of the production and design to this Senior Design project. This will not only be something that has already been created in the market, but the additional Optical features adds a unique feature that has not been implemented in the market nowadays. This can also improve the quality of life for these pets and solve the issue pet owners face in their day to day life.

Manual pet feeders generally only consist of two plastic/metallic bowls, one for water and the other for the food. These bowls are placed on the floor and require the owner to refill twice/thrice a day depending on how often the pet requires to be fed. Pet feeders are relatively a new concept that recently reached the market, and contains more room to grow. The pet feeder uses the automation concept, a device like this makes the difference between pets getting fed or left starving. Devices like this are not meant to take all the responsibility of taking care of a pet. This device is meant to assist pet owners whenever they are busy with work that can affect the health of these pets, due to working over hours, or late at night etc. One of the features that we wanted to make our device unique to other products that are available to the market is the application feature that allows users to stay connected to their pets even when they are away from their home.

This project allows us to demonstrate the knowledge each team member has learnt here in the University of Central Florida. This project also creates the challenges of showing that each member can communicate effectively in order to achieve a common goal. This is a great opportunity and experience before embarking on our future career path. Our mutual common interest of owning a pet has led us to the idea of this project, and the features that we have added were inspired by the lack of product as well as our love towards our pet. The hurdle of the challenges that we will be facing creating this project is a perfect simulated experience for the “real world” project that we will all encounter down the road of our career. And a device like this can change the future lives of pets worldwide

2.3 Goals

The overall goal in this project is to create an automatic system that requires minimum user input in order to ensure that the owner does not need to worry about feeding their pet around their busy schedule.

2.3.1 Core Goals

Goal Number	**Goal Description**
1	Power supply successfully charges the device and help simplify users' life
2	Use optics and photonics will be utilized to recognize pets with the correct tag to access each individual food
3	"Pawsitive" Pet Feeder will store approximately two days' worth of food equivalent for three pets. This can be either 3 dogs or 3 cats.
4	Maintain market price (~$250) while offering more features than the average feeder in the market
5	Allow user to set a profile that will determine the correct time of day on when the device should dispense the food according to the pet's needs
6	Food storage dispenses a desired amount of food per feeder
7	Self-opening and self-closing lid when the pet arrives or leaves to the pet feeder device

Table 1: List of Core Goals

2.3.2 Advanced Goals

Goal Number	**Goal Description**
1	Create water resistant LED collar tags to avoid damage while not altering the pets’ lifestyle
2	Create an application for the user to have the ability to control the system remotely for functions

Table 2: List of Advanced Goals

2.3.3 Stretch Goals

Goal Number	**Goal Description**
1	Allowing the user to remotely open the pet feeder lid. In instances where the collar tag is unresponsive or malfunctioning
2	User configurable setting to adjust a specific time of certain days for the lid to remain open.
3	Allow users to talk to pets remotely through the mobile application.
4	System will include a nocturnal mode.

Table 3: List of Stretch Goals

2.4 Objectives

The project consists of many different components and thus the following objective that are listed below defines the measured actions that will be taken in order to achieve the overall goals that was mentioned in section 2.3

2.4.1 Core Objectives

Objective Number	**Objective Description**
1	Utilize wall outlet adapter. Thus, users are not required to switch out batteries when power runs out.
2	Create a camera system that recognizes RGB colors from LED collar tags in order to open an assigned bowl and lid system.
3	Food storage for each feeder will be at least 4 Liters.
4	Utilize the use of 3D printers, and maximize components to drive cost of materials down
5	Mobile App will be used by the user to set certain times when the pet should be fed throughout the day.
6	Figure out the amount dispense within one rotation of the gear and calculate to the correct serving for the given pet (dog/cat)
7	Utilize an ultrasonic sensor that helps detect pet movements, as well as collar detection front camera module. Data will be used to control movement of the motor that is attached to the lid.

Table 4: List of Core Objectives

2.4.2 Advanced Objectives

Objective Number	**Objective Description**
1	Use IPX-4 LEDs. Aim to use IPX-7 LEDs
2	Use sensors within the system as received data
3	Create an android mobile application for the user to use to set certain times when the pets' food should be dispensed and to dispense the desired amount of food.

Table 5: List of Advanced Objectives

2.4.3 Stretch Objectives

Objective Number	**Objective Description**
1	Creating a function in the mobile application to override the lid system. Forcing the lid system to remain open or closed.
2	Creating a time-based feeding system. The user will have the ability to choose what time the lid will open and close for feeding.
3	If budget allows, utilize a camera with a built-in speaker for the user to interact with the pet through the "Pawsitive" Pet Feeder via the mobile application's microphone functionality.
4	Allow the system to have a nocturnal mode. By using a daylight sensor, we would allow the camera system to detect at night or in the dark.

Table 6: List of Stretch Objectives

2.5 Required Specification

The design project is for a "Pawsitive" Pet Feeder. In our original design the user would be able to go through the day without having to worry about not having to feed the pet. This “automation system” would have been controlled by three major components, the application and the collar tag that is worn by the pet and the Smart Pet Feeder. The application side is where the user can set a specific time of day when the Pet Feeder would dispense the food for the pet. The collars used by the pet would be used to identify which bowl’s lid to open that will be controlled by the detection color from the camera. The photodiodes that will be placed around the lid will be used to detect the light and determine whenever the lid should be kept open or closed. This entire process (with the exception of user input of time and day) would be completed without any required user interactions.

2.5.1 Hardware Requirements

The table below contains the requirements concerning the device's hardware. The requirements below have been established to guarantee that the system constraints adequate hardware functionality to achieve proper system behavior.

The "Pawsitive" Pet Feeder will be able to perform the following requirement specifications as shown in the following table:

1	Camera System Detection Colors	3
2	Detection Range	Minimum: 0.5 Feet Maximum: 3 Feet
3	Power Consumption	< 100 Watts
4	Response Time	< 5 Seconds
5	Dispensing Time	45 Seconds ± 15 Seconds
6	Water Resistant	IPX - 4: Protected against a splash of water from any direction < 10 Minutes
7	Automatic Lid Closing After Pet Leaves	60 Seconds ± 10 Seconds
8	Amount of Food Dispense	± 20% of User Inputted Amount
9	Cost Requirement	Between $ 200 - $ 400

Table 7: Hardware Requirements for Design

The roles assigned for Group 8 are assigned in the block diagram shown below, Figure 2.5.1-1.

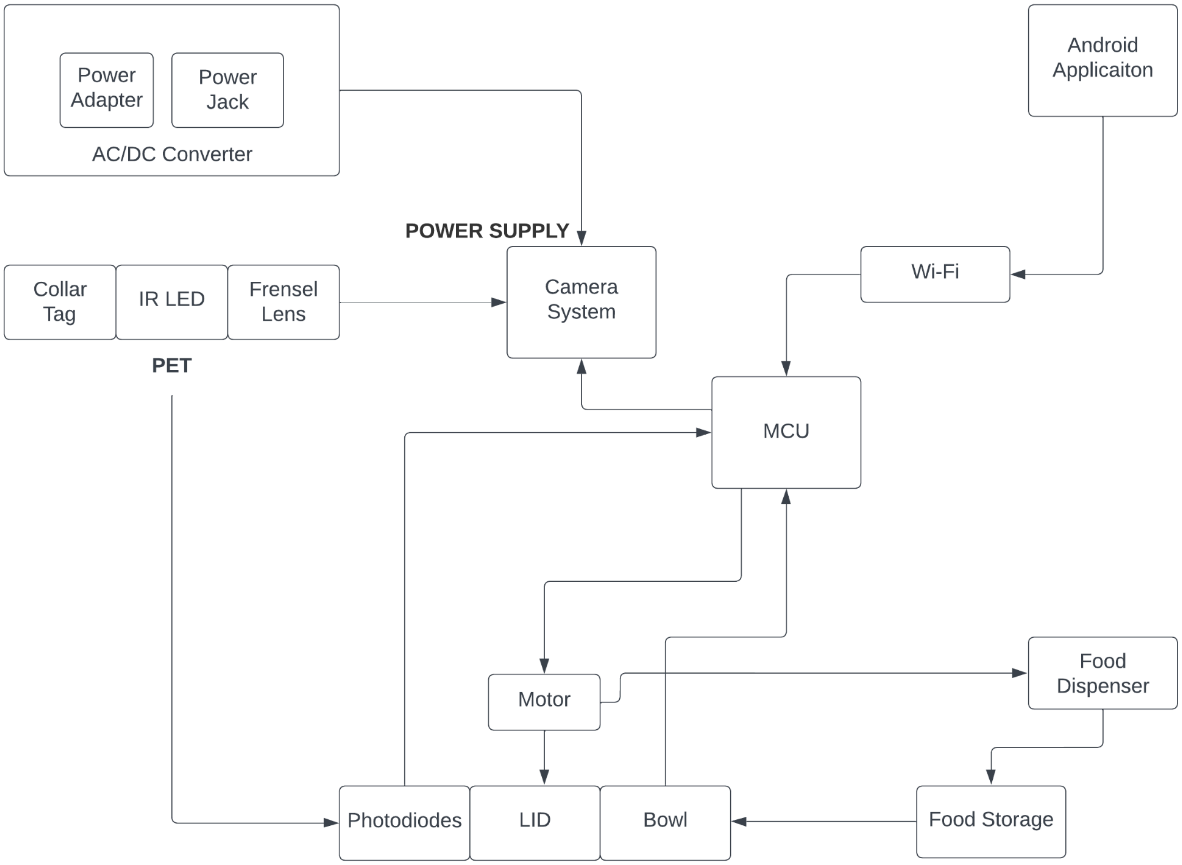


Figure 1: Hardware Block Diagram

2.5.2 Software Requirements

The system software requirements are outlined as shown in the following table below. A requirement list serves as the starting point for developing software functionality and components within a system that will be acceptable in terms of functionality and functionality.

1	The system will include a Wi-Fi module that enables connection with microcontroller to control lid motors
2	The System shall have a mobile application, that allows the user to set up an input of the time of day when the feeder will dispense food for the pets to be feed
3	The system software should utilize camera module to detect if the collar tag is within range to enable the system to lift the lid for the pets access the bowls
4	The system should have motion sensor that detects movement when the pet is near the device if pet wants to be fed
5	The system should notify the user through a mobile application if the food storage is low in food.
6	The system should allow the user to remotely dispense food to feed pet via mobile application

Table 8: Software Requirements for Design

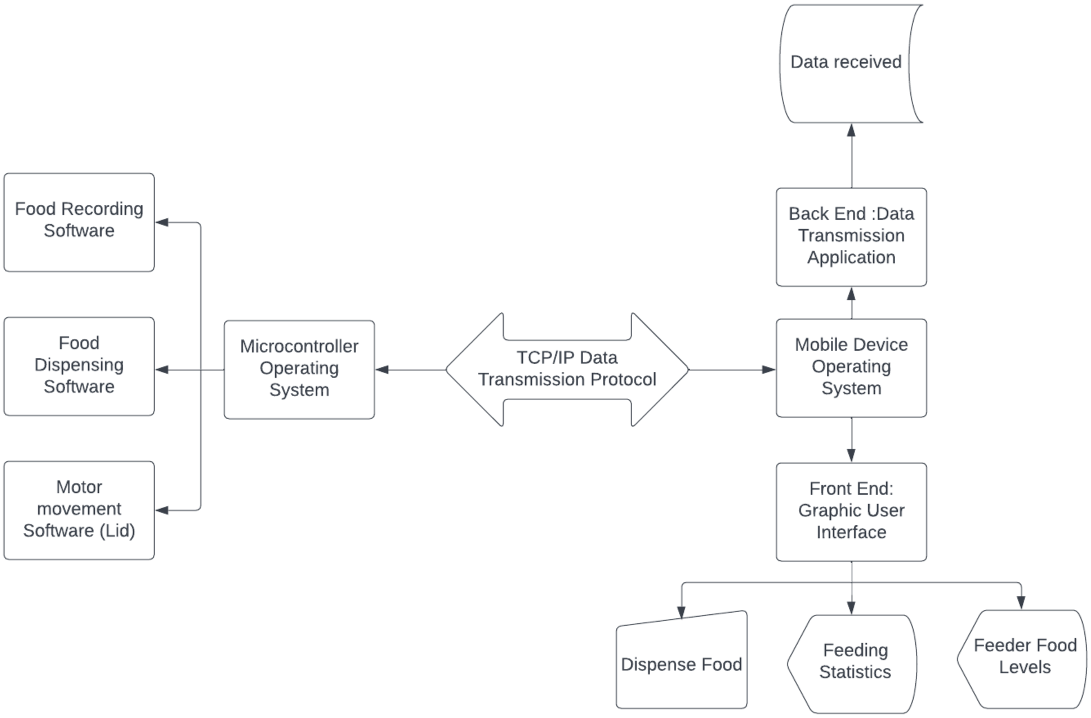

Data received
Back End :Data Transmission Application
Food Recording Software
Food Dispensing Software
Microcontroller Operating System
TCP/IP Data Transmission Protocol
Mobile Device Operating System
Motor movement Software (Lid)
Front End: Graphic User Interface
Dispense Food
Feeding Statistics
Feeder Food Levels

Table 2: Software Block Diagram

2.5.3 Enclosure Requirements

The system software requirements are outlined as shown in the following table below. A requirement list serves as the starting point for developing software functionality and components within a system that will be acceptable in terms of functionality and functionality.

1	The enclosure should possess at least 17” L x 10”W of space for device of food dispenser system
2	The collar should be tolerant to humidity, i.e waterproofing any tags, if the pets wears the collar and drinks water the device will not be damaged
3	The device should be portable, and simple, meaning no equipments or tools is required for the user to install the device

Table 9: Enclosure Requirement Specifications

2.6 Functionality of the “Pawsitive” Pet Feeder

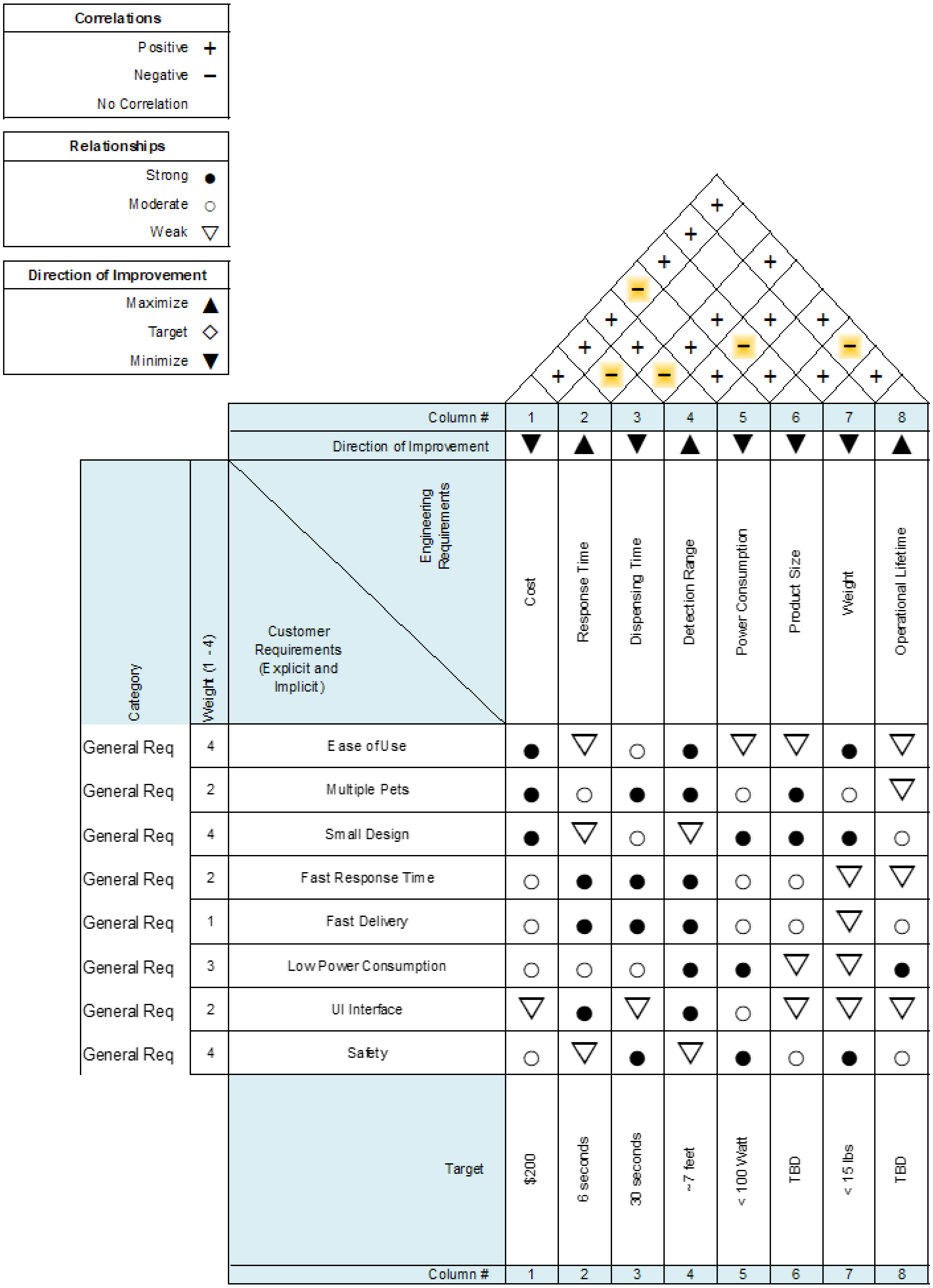
When there is no activity around the “Pawsitive” Pet Feeder, all the photodiodes will be activated by the IR LEDS and the lids will remain closed. When the pet enters the area of the "Pawsitive" Pet Feeder with the collar tag, which has an attached colored LED and Fresnel lens, the camera system will then be able to pick up which color LED is shining and recognize which pet is entering the area. Once the camera system detects the pet waiting to get food, the lid will open. While the pet is eating, its head will block the signals of the IR LEDs. This will cause a drop in power detected by the photodiodes. Once the pet is done eating, the photodiodes will gain an increase in power. This will start an internal countdown. If the photodiodes do not go through another drop in power before the timer is done, the lid will close. If the power does drop, the countdown process will repeat.

2.6.1 Website / Application

The pet feeder includes a webpage and application functionality that allows users to communicate with the device remotely via Bluetooth and Wi-Fi. This enables the user to use the device to its full potential while on the go and away from their pets. The Pet Feeder webpage first introduces the product to the user through its interactive homepage, presenting the product's features, explaining the product's goals and objectives, and further describing the creation team's overall motivation and drive, answering the question "why are we creating this product?" After navigating through the homepage, users can go to the sign-up page if they are a new user or the log-in page if they are a returning user. To access the Pet Feeder functionality safely on the website, new users must go through an online account registration process, providing basic personal information such as first name, last name, email, and password. The creation of a secure account will rightfully eliminate any hazardous and harmful security risks for all of our online users. After completing the online account registration process and having the user's email verified, new and existing users can now log into their newly created account, which will direct them straight into the webpage dashboard that neatly displays all of the Pet Feeder's functionality for the users to fully control at their leisure. The application functionality for users to use follows the same usage steps as the webpage; however, this application may be downloaded and used on your smartphone or tablet to make remote access to the device conveniently accessible to all users on your smartphone or tablet.

2.7 Marketing and Engineering Requirements

The house of quality as shown below in Table 2. shows the different relationship between engineering challenge and customers specification on the "Pawsitive" Pet Feeder project. Since currently there are many different similar products that are found in the market today, it can get difficult to keep track of the importance of highlighting the quality of the product for the user ends. The house of quality table is very important in the planning stage as it helps in the product planning matrix that is built to show how customers' requirements relate to the way and method we will be using to achieve the requirements in our design product. A strong house of quality can help reinforce the goals of the project and focus on what customers have asked for within the project.



Correlations

Positive +
Negative −
No Correlation

Relationships

Strong ●
Moderate ○
Weak ▽

Direction of Improvement

Maximize ▲
Target ◇
Minimize ▼

		Column #	1	2	3	4	5	6	7	8
		Direction of Improvement	▼	▲	▼	▲	▼	▼	▼	▲
Category	Weight (1 - 4)	Engineering Requirements Customer Requirements (Explicit and Implicit)	Cost	Response Time	Dispensing Time	Detection Range	Power Consumption	Product Size	Weight	Operational Lifetime
General Req	4	Ease of Use	●	▽	○	●	▽	▽	●	▽
General Req	2	Multiple Pets	●	○	●	●	○	●	○	▽
General Req	4	Small Design	●	▽	○	▽	●	●	●	○
General Req	2	Fast Response Time	○	●	●	●	○	○	▽	▽
General Req	1	Fast Delivery	○	●	●	●	○	○	▽	○
General Req	3	Low Power Consumption	○	○	○	●	●	▽	▽	●
General Req	2	UI Interface	▽	●	▽	●	○	▽	▽	▽
General Req	4	Safety	○	▽	●	▽	●	○	●	○
		Target	$200	6 seconds	30 seconds	~7 feet	< 100 Watt	TBD	< 15 lbs	TBD
		Column #	1	2	3	4	5	6	7	8

Figure 3: House of Quality Diagram

]]>

3.0 Research Related to Project

This section will focus on the research done based on the previous work that has been done. Along with the technologies and equipment they have used in order to complete their projects. This provides the opportunities to see an overview of the components they have selected and compare them with our research in order to provide an optimal selection for our project.
The following section contains the content that is listed below:

- A list of existing projects and projects related to the "Pawsitive" Pet Feeder
- An outline of features and accessories that we relates to the design of the project
- An outline of general components that we carefully selected for the project

3.1 Existing Project and Products

Pets are a big part of our families. Nonetheless, it can be time consuming to keep a pet fed correctly. With busy lifestyles, there is a need to find an easier way to feed the pets, watch them, and make sure the food is isolated from insects. Due to this big concern, many products and designs have been done in the past to fix this problem. In the following section we are going to be exploring the previous senior design projects made at the University of Central Florida that tried to automate the feeding process of pets at home. We will also be looking at market products and comparing them to our design and show what we are doing differently and how our design will be innovative and practical. Automated pet feeders are not a new idea and there are those that perform either similarly or the same functions. There are existing technologies out there that can dispense food at predetermined times with very different design ideas.

This research was significant to the design of our project as we wanted to take the best features from existing products and integrate into one. Also, we learned from constraints from previous groups. We use this information to avoid future problems or improve in the area they might have lacked more research and development.

3.1.1 Market Most Popular Pet Feeders

In this section, we researched the most popular pet feeder on the market to have a better understanding of the marketing we are entering. This section was helpful to understand what features customers value more.

3.1.1.1 PETLIBRO Automatic Pet Feeders

It is affordable and easy to manual-program to dispense food throughout the day. Nonetheless, it doesn't work well for multiple pets as the bowl is small. It doesn't work well for households with pets that have different diet restrictions as there is only one bowl. Also, it doesn't work well with WiFi.

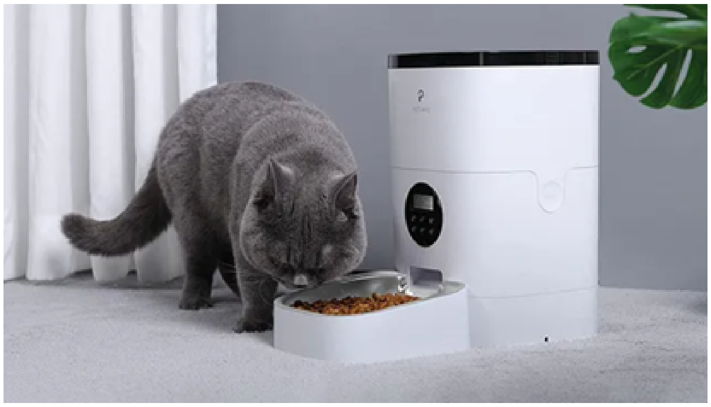


Figure 4: Existing Automated Food Dispenser In Market (Courtesy of PETLIBRO)

3.1.1.2 Petlibro Granary Automatic Dog & Cat Feeder

Although it is a more expensive option than the PETLIBRO Automatic Pet Feeders, it is better for households with multiple pets as it has an adjustable food splitter. It has a touchscreen for programming feedings. Nonetheless, it lacks remote control. If the user forgets to program the feeder, food will not be dispensed until pet's owners manually dispense it.

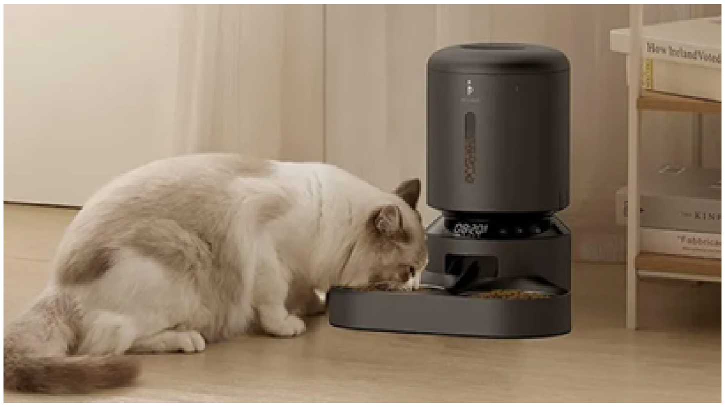


Figure 5: Petlibro Granary Automatic Dog & Cat Feeder (Courtesy of PETLIBRO)

3.1.1.3 SureFlap SureFeed Microchip Automatic Pet Feeder

This is a great option for households with multiple pets with different diet restrictions. It has a lid that only opens for one pet. The way it recognizes the pet is with microchip-based signals. Most of the time pets already have a chip which means it wouldn't add additional cost to the product. It doesn't work well for big breeds, nor does it dispense more food, which might be a problem for pet owners with busy schedules, who are not at home during the day.

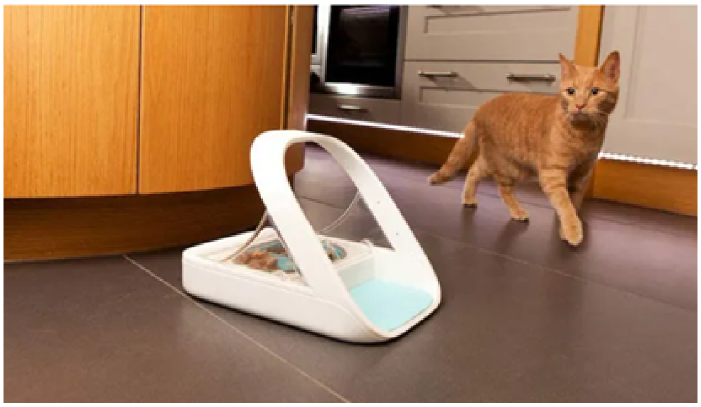


Figure 6. SureFlap SureFeed Microchip Automatic Pet Feeder (Courtesy of Sure Petcare)

3.1.2 Previous Automatic Pet Feeder Senior Design Projects

Automated pet feeders are not a new idea and there are those that perform either similarly or the same functions. Nonetheless, the objective of this section was to study the components, functionality and features of previous senior design projects to better understand the market we were entering and the products that are profitable. We made sure we chose features that set us closer to our goal of feeding, watching, and remote controlling the feeder to accommodate busy schedules of pet owners.

3.1.2.1 Opto-Smart Pet Feeder, Group 9, Spring 2022

This design is a pet feeder that integrates optics to identify targets to output predetermined user settings. Along with a computer-based program that allows the user to add, remove or edit various settings. From this project, we learned techniques to identify the different pets at their bowls.

3.1.2.2 The Smart Bird Feeder, Group 7 Fall 2021

This design uses various sensors and electrical components to build a bird feeder. An innovative smart bird feeder is developed and takes advantage of computer vision and machine learning to distinguish between birds and squirrels visiting the feeder. From this project, we learned different techniques to use moving lids to prevent pets from eating food that is not theirs.

3.1.2.3 Automatic Pet Food Dispenser, Group 25, Summer 2021

This design provides the end user with automatic/manual control of when, and how much, food is dispensed for their pet.It will be used remotely, or physically, at the dispenser through buttons and/or a menu. From this project, we use some features in the mobile app while improving some features they didn't include, such as notifying when the food is low.

4.0 Relevant Technologies

In this section, a brief overview of the relevant technologies used in the device is given along with an explanation on how these components operate.

4.1 Fresnel Lens

Fresnel lenses diffuse light by adjusting the beam light source to parallel light, this process significantly improves the brightness. This type of lens will be used inside the collar tags in front of the LEDs to improve its brightness. This is useful because the light from just the LED might not be enough for the camera system to detect the color from a far distance. With the fresnel lens, we can ensure the pet has its lid already open by the time it reaches the feeder. This lens is a low-cost alternative, as well as lighter weight compared to a regular lens. This is an advantage to our design because the pets will be always wearing this collar, and it will be easier for them if it's light. Some downsides to using this type of lens is that the image quality will decrease, but for our purposes of light diffusion, this is not an inconvenience. Another advantage of this lens is that it can be molded from plastic into a circular disk and then milled with a CNC machine to produce the ridges required to focus light.

4.2 Photodiodes

Photodiodes are optoelectronic devices we will use in our design to convert light into electrical current. As devices that detect light, we can choose the wavelength for which it will be receptive. We plan to operate at specific wavelengths and thus operate like a bandpass filter. We are planning to use infrared (IR) light around 950 nm. We will use them to detect either the presence or absence of light. When the pet approaches the bowl, the photodiode will receive signals and therefore produce photocurrent that will open the lid corresponding to the color the camera system identifies. To keep the lid open, the photodiode will keep receiving photons from the IR light. Once the photodiode doesn't receive any signal, it means the pet has left the bowl and it can now be closed. Another advantage of using photodiodes is that they are low cost and we will be able to purchase many to increase the sensibility.

4.3 LEDs

Light emitting diodes, also known as LEDs, are semiconductor devices that emit light when current flows through them. Electrons in the semiconductor recombine with electron holes and release energy in the form of photons. They are cheap, light and low power devices. They are great for our design because we are aiming to use the least heavy materials for the collar tags. Also, we plan to compensate for the low power by placing a Fresnel Lens in front of it to diffuse light and make it visible to the camera system from a farther range.

4.4 Voltage Regulators

Voltage regulator will be used to convert a range of input voltages into a desired DC output voltage to be used to provide power to components in a circuit. These are mostly used when the input voltage is large and needs to be reduced to be introduced to the circuit. When voltage regulators lower the input voltage, it dissipates the excess power as heat which would make the new input safe for the microelectronics. There are multiple different variations of voltage regulators, including ones built with the use of diodes and separate integrated circuits that can be applied on their own.

4.5 Microcontrollers

A microcontroller is a small computer on a semiconductor integrated circuit. It contains one or more processors, a RAM memory, and a programmable input/output peripheral. We will be looking at different options for our design including word lengths of 4 bit, 8 bit, 16 bit, and 128 bit. A microcontroller is manufactured to automatically control products and devices for them to function properly inside of an embedded system.

4.6 MOSFETS

MOSFET (metal-oxide-semiconductor field-effect transistor) is a type of field-effect (FET) and is most used to control conductivity, in other words, how much electricity can flow between its source and drain terminals based on the amount of voltage applied to its gate terminal. There are two types of MOSFETS, PNP and NPN, consisting of the enhancement and depletion type.

5.0 Research and Part Selection

The "Pawsitive" Pet Feeder has multiple hardware component elements. To select the best option possible for our goals and budget, we made extensive research in the selection of each component. This section includes the research and selection of components based on comparison between components.

The following section contains research and part selection for:

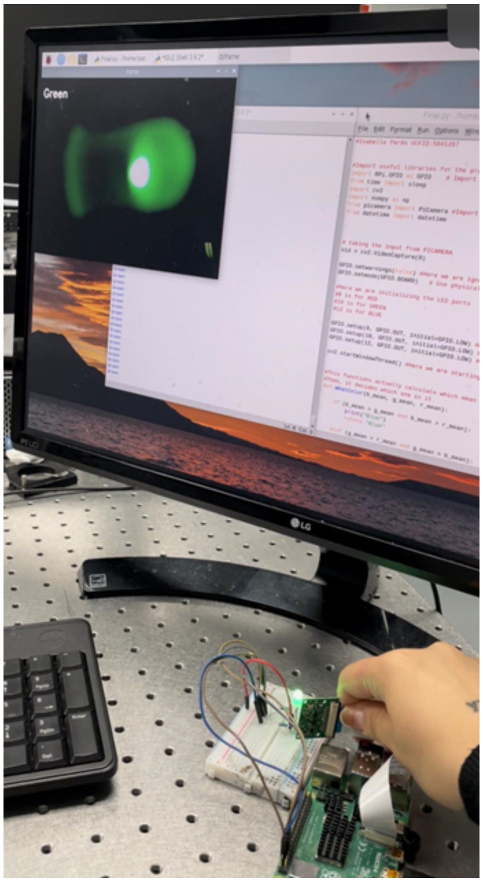
- Camera system
- Collar Tag
- Food Dispenser
- Bowl and Lid System
- Photodiodes
- Phone application

5.1 Camera System Research

The camera system will be able to identify three different colors corresponding to the LEDs inside the collar tags. As shown in Figure. 4. The camera will look for one of the specific three RGB values and identify it correctly. At the same time, it will be able to live stream to the smartphone application.

We did our Fall 2022 midterm demonstration on the camera system. For this we used a microprocessor, a camera module, an RGB LED, three 330 Ω resistors, a breadboard, an jumper wires. In this demonstration, we built the camera system we will be using for our device. The RGB LED is simulating the collar tag the pet will be wearing. We coded the microprocessor to print the color it is observing, to show the image the camera is observing, and to print the color on screen as well. By holding the camera module close to the LED, we achieve our desired outputs successfully. From this demonstration, we learned the importance of Fresnel lenses for our collar tags. As we moved the camera farther from the LED, it was harder for the camera to identify the color correctly and by having the Fresnel lens in front of the LED inside the collar tag, light diffusion is higher and easier for the camera to identify colors.

The camera system will constantly be monitoring the feeder to see the assigned color and activate the system to start the process of opening the lid. Below we will explore the different features of multiple cameras that were considered for our design and the process of choosing the one that fits our goals and budget better.



```
#Isabella Pardo UCFID:5041287

#Import useful libraries for the pr
import RPi.GPIO as GPIO    # Import
from time import sleep
import cv2
import numpy as np
from picamera import PiCamera #Import
from datetime import datetime


# taking the input from PICAMERA
vid = cv2.VideoCapture(0)

GPIO.setwarnings(False) #here we are ign
GPIO.setmode(GPIO.BOARD)    # Use physical
#here we are initializing the LED ports
#8 is for RED
#10 is for GREEN
#12 is for BLUE

GPIO.setup(8, GPIO.OUT, initial=GPIO.LOW) #
GPIO.setup(10, GPIO.OUT, initial=GPIO.LOW) #
GPIO.setup(12, GPIO.OUT, initial=GPIO.LOW) #
cv2.startWindowThread() #here we are starting
#This functions actually calculate which mean
#Then, it decides which one is it.
def WhatColor(b_mean, g_mean, r_mean):
    if (b_mean > g_mean and b_mean > r_mean):
        print("Blue")
        return "Blue"
    elif (g_mean > r_mean and g_mean > b_mean):
```

Figure 7: Camera System During Testing

5.1.1 Camera Selection

In the following section we research the different cameras we consider for our project.

5.1.1.1 OV5647 Mini Camera Module

A pack of two pieces OV5647 Mini Camera Module is sold on Amazon for $8.99. It is compatible with Raspberry Pi Model A, B, B+, Pi 2 and Raspberry 3, 3 B+, Pi 4. It records in 1080P with a focus lens of 5 megapixels. This camera has a 6-inch ribbon cable that can be used to easily integrate into the small pockets of The Raspberry Pi.

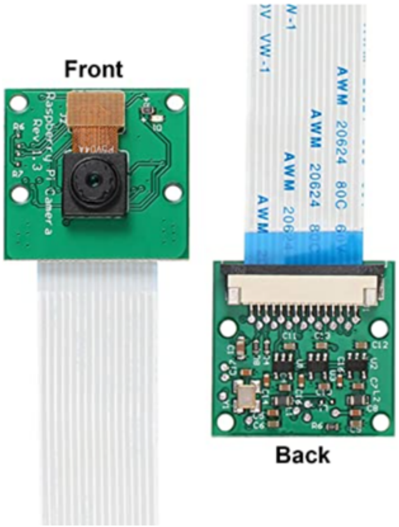


Figure 8: OV5647 Mini Camera Module (Courtesy of Amazon)

5.1.1.2 Ximimark USB Camera

The Ximimark USB camera model LY096 is sold on Amazon for $8.79. It is compatible with all Raspberry Pi versions. It has a 640x480P video recording and adjustable focal length. It can be easily integrated into The Raspberry Pi by plugging in its USB port. It is 5 x 1.8 x 0.7 inches and lightweight.

Figure 9: Ximimark USB Camera Courtesy of Amazon)

5.1.1.3 Arducam Mini Module Camera Shield (OV2640)

The Arducam Mini Module Camera Shield is sold on Amazon for $25.99. It is compatible with Arduino boards and Raspberry Pi. Despite being the most expensive option, it was considered since it is compatible with Arduino boards, and this is an option we are considering for the microprocessor. It has a focus lens of 2 megapixels, and it is 2 x 1.7 x 1.6 inches.

Figure 10: Arducam Mini Module Camera (Courtesy of Amazon)

5.1.1.4 Arducam 5MP Ultra Wide Angle Fisheye Lens Camera (OV5647)

The Arducam 5MP Ultra Wide Angle Fisheye lens camera is sold on Amazon for $28.99. It has a 220° horizontal field of view allowing to see large-scale areas at close range and fully compatible with Raspberry Pi. There is no need to install an additional driver for this camera to work properly. It has a 36mm x 36mm size, with a 5 MP focus lens.

Figure 11: Arducam 5MP Ultra Wide Angle Fisheye Lens Camera (Courtesy of Amazon)

5.1.1.5 Camera Options Comparison

	OV5647 Mini Camera Module	Ximimark USB Camera	Arducam Mini Module Camera Shield (OV2640)	Arducam 5MP Ultra Wide Angle Fisheye Lens Camera (OV5647)
Price	$8.99 for two	$8.79	$25.99	$28.99
Compatibility	Raspberry Pi	Raspberry Pi	Raspberry Pi and Arduino	Raspberry Pi
Resolution	1080 p	640 x 480 p	-	1080 p
Size	0.98 x 0.94 x 0.35 inches	5.0 x 1.8 x 0.7 inches	2.0 x 1.7 x 1.6 inches.	0.98 x 0.94 x 0.35 inches
Port Type	CSI	USB	Pins	USB

Table 10: Camera Options Comparison

5.2 Collar Tag System

The collar tag will have the ability to be integrated into any normal collar so that pets of all sizes can use it. Each pet will have a different color LED. In front of the LED, we will use a Fresnel lens to diffuse light in order to enable a larger detectable distance for the camera system. The following section includes the research done in the components of the collar tag: LEDs and Fresnel Lens.

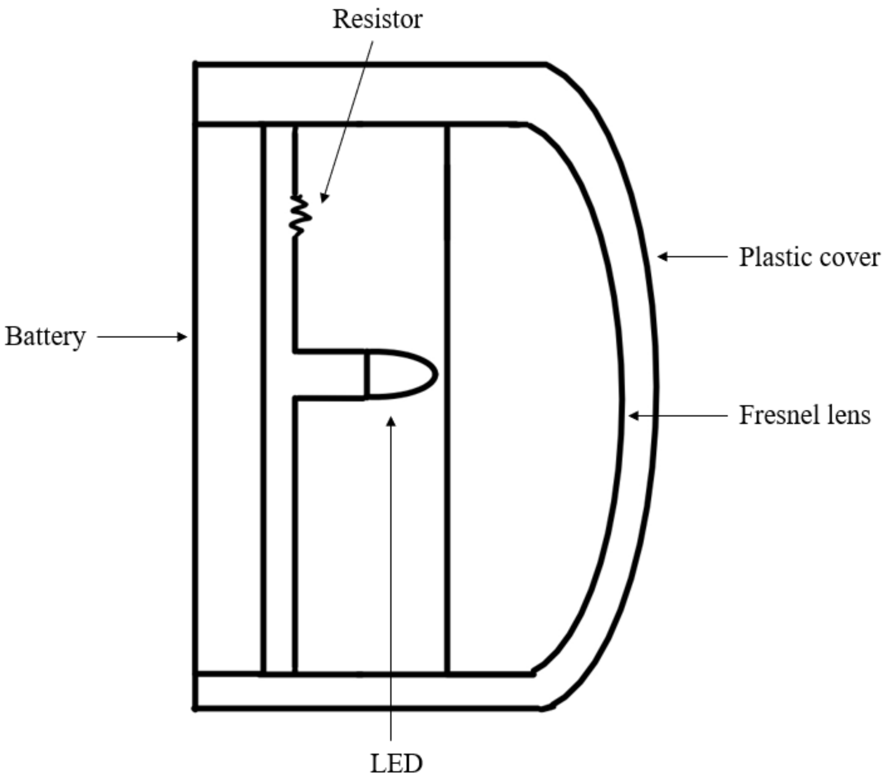


Figure 12: Collar Tag System

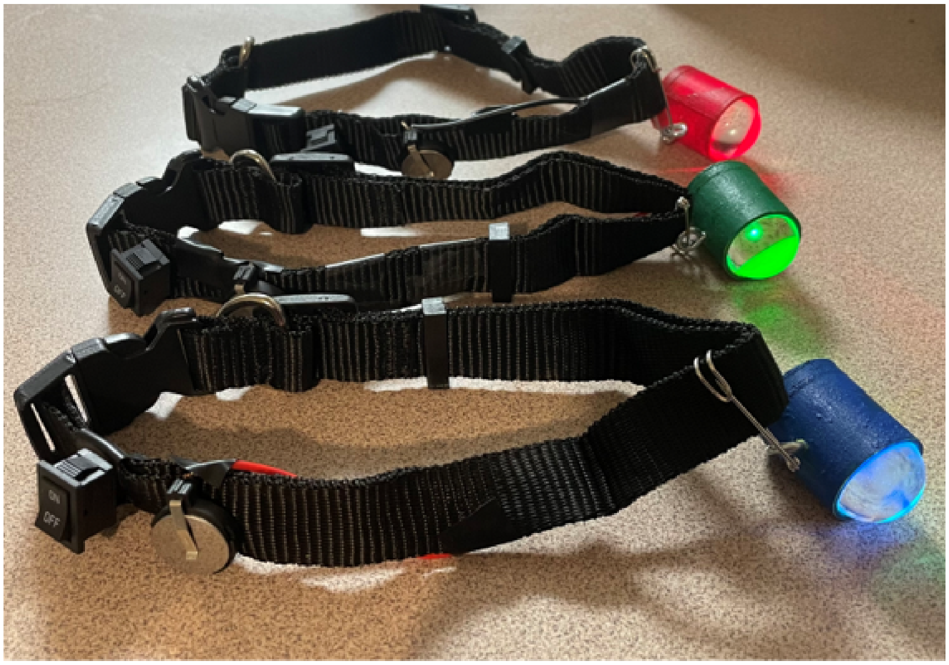


Figure 13: "Pawsitive" Pet Feeder Collar Tags

5.2.1 LED

LEDs are a vital part of our project as their size and shape allow for a small collar tag. Also, LEDs are typically low power since the device will run on a single battery. We will be using red, green and blue LEDs. In this section we will explore the different options studied for all three LEDs and the final choice made.

The below equation will be used to help us calculate key information about LEDs.

$$\frac{\left(V_{in} - V_{LED}\right)}{I_{LED}} = R_{s}$$

Equation 1: LED Voltage Equation

5.2.1.1 RGB LED Options

For our CREOL Fall 2022 Midterm Demonstration we used an RGB LED as we already had it available. It was a convenient option as we didn't have to wait for the shipment time and could start working immediately in the demonstration.

5.2.1.1.1 754-1615-ND (LED RGB CLEAR)

Digikey sells this component for $1.92. It operates Red, Green, Blue (RGB) 630nm Red, 525 nm Green, 465 nm Blue LED Indication. It has a forward voltage of 1.9V Red, 3.3V Green, 3.3V Blue Radial.

5.2.1.1.2 COM-00105 (LED - RGB Clear Common Cathode)

Sparkfun sells this component for $2.25. It has forward voltages of 2.0 V Red, 3.2 V Green, 3.2V Blue. It has an operating current of 20 mA and a viewing angle of 50°.

5.2.1.1.3 NTE30159 (LED 10MM RGB COMM CATHODE)

Sparkfun sells this component for $0.80. It has forward voltages of 2.1 V Red, 3.0 V Green, 3.0V Blue. It has an operating current of 20 mA and a viewing angle of 30°.

5.2.1.1.4 RGB LED Comparison

	754-1615-ND	**COM-00105**	**NTE30159**
Wavelength Red (nm)	630	623	625
Wavelength Green (nm)	525	517.5	525
Wavelength Blue (nm)	465	466	465
Intensity Red (mcd)	400	800	9000
Intensity Green (mcd)	1700	4000	7000
Intensity Blue (mcd)	900	900	6500
Forward Voltage Red (V)	1.9	2.0	2.1
Forward Voltage Green (V)	3.3	3.2	3

Forward Voltage Blue (V)	3.3	3.2	3
Forward Current (mA)	20	20	20
Viewing Angle (degrees)	50	50	30
Cost	$1.92	$2.25	$0.80 (pack of 3)

Table 11: RGB LED Comparison

For our design RGB LEDs will not be used as we decided that using LEDs with a single operating wavelength would be more efficient and simpler. Single color LED only uses 2 lead systems whereas the multicolor LED uses a 4-lead system. The multicolor LED was also found to put more strain on the battery since each color required a different voltage and more resistors would need to be included in the collar tags PCB.

5.2.1.2 Red LED options

In this section, we will discuss the best options for red LEDs for the "Pawsitive" Pet Feeder.

5.2.1.2.1 5 mm Round Top Red LED – Ultra Bright

Lighthouse sells the 5 mm Round Top Red LED – Ultra Bright green LED for $0.18. It operates at 630 nm. It operates at a current of 20 mA. It has a forward voltage of 2.1 V and a viewing angle of 30 degrees. Its mount style is through hole.

5.2.1.2.2 RL5-8030 (Red LED)

Super Bright LEDs sells RL5-8030 for $0.28. It operates at 635 nm. It operates at a current of 20 mA. It has a forward voltage of 2.2 V and a viewing angle of 30 degrees. Its mount style is through hole. This LED has a power dissipation of 80 mW.

5.2.1.2.3 SSL- LX5093ID-5V

Digikey sells the SSL- LX5093ID for $0.69. It operates at 635 nm. It has a forward voltage of 5 V and a viewing angle of 60 degrees. Its mount style is through hole.

5.2.1.2.4 Red LED Comparison

	5 mm Round Top Red LED – Ultra Bright	**RL5-8030 (Red LED)**	**SSL- LX5093ID-5V**
Wavelength (nm)	**630**	**635**	**635**
Forward voltage (V)	**2.1**	**2.2**	**5**
Operating Current (mA)	**20**	**20**	**-**
Viewing angle (degrees)	**30**	**30**	**60**
Typical Intensity (mcd)	**8000**	**18000**	**2000**
Cost	**$0.18**	**$1.47**	**$0.69**

Table 12: Red LED Comparison

Shown in purple, RL5-8030 was chosen because it gives the best price for intensity and viewing angle.

5.2.1.3 Green LED options

In this section, we will discuss the best options for green LEDs for the "Pawsitive" Pet Feeder.

5.2.1.3.1 SLR-56MG3F

Digikey sells the SLR-56MG3F green LED for $0.59. It operates at 572 nm. It operates at a current of 10 mA. It has a forward voltage of 2.1 V and a viewing angle of 40 degrees. It has a typical luminosity of 16 mcd.

5.2.1.3.2 Super Bright Green 5mm LED

Adafruit sells a pack of 25 super bright LEDs for $8.00. It operates at 520 nm. It has an operating current 20 mA and has a forward voltage 3.2 – 3.8 V. It has a viewing angle of ±10 degrees. It has a typical intensity of 8000 mcd which is significantly higher than the other LED option.

5.2.1.3.3 ALMD-CM3E-Y1002

The ALMD-CM3E-Y1002 Green LED is sold by Mouser for $1.47. It operates at 525 nm. It has an operating current of 30 mA. It has a forward voltage of 3.2 V. It has a viewing angle of 30 degrees.

5.2.1.2.4 Green LED Comparison

	SLR-56MG3F	ALMD-CM3E-Y1002	Super Bright Green 5mm LED
Wavelength (nm)	572	525	520
Forward voltage (V)	2.1	3.2	3.8
Operating Current (mA)	10	20	20
Viewing angle (degrees)	40	30	20
Typical Intensity (mcd)	16	9300	8000
Cost	$0.59	$1.47	$8.00 for 25

Table 13: Green LED Comparison

Shown in purple, ALMD-CM3E-Y1002 was chosen because it gives the best price for intensity and viewing angle.

5.2.1.4 Blue LED options

In this section, we will discuss the best options for blue LEDs for the "Pawsitive" Pet Feeder.

5.2.1.4.1 APTD1608QBC/D

Mouser sells the APTD1608QBC/D for $0.50. It operates on a wavelength of 470. It has an operating current of 20 mA and a forward voltage of 3.3 V. It has an intensity of 250 mcd with a viewing angle of 40 degrees. Its intensity is 250 mcd.

5.2.1.4.2 C503B-BCS-CV0Z0461

Digi key sells the C503B-BCS-CV0Z0461 Blue LED for $0.21. It operates at a wavelength of 470 nm. It has a forward voltage of 3.2 V and an operating current of 20 mA. It has a viewing angle of 30 degrees and has an average intensity of 4800 mcd.

5.2.1.4.3 Super Bright Blue 5mm LED

Adafruit sells a pack of 25 bright blue LEDs for $8.00. It operates at a 465 nm wavelength. It has a forward voltage between 3.2 and 3.8 V. It has an operating current of 30 mA. It has a 20-degree viewing angle, with an intensity of 7000 mcd.

5.2.1.2.4 Blue LED Comparison

	APTD1608QBC/D	C503B-BCS-CV0Z 0461	Super Bright Blue 5mm LED
Wavelength (nm)	**470**	**470**	**465**
Forward voltage (V)	**3.3**	**3.2**	**3.8**
Operating Current (mA)	**20**	**20**	**30**
Viewing angle (degrees)	**40**	**30**	**20**
Typical Intensity (mcd)	**250**	**4800**	**7000**
Cost	**$0.59**	**$1.47**	**$8.00 for 25**

Table 14: Blue LED Comparison

Shown in purple, C503B – BCS – CV0Z0461 was chosen because it gives the best price for intensity and viewing angle.

5.2.2 Fresnel Lens

Fresnel lenses are made up of concentric circles which can take light and turn them into a narrow beam. This concentrates the light and allows it to be focused in a certain direction. Fresnel Lenses have a short focal length so it can be relatively close to the LED in the collar which would help with keeping the collar more compact.

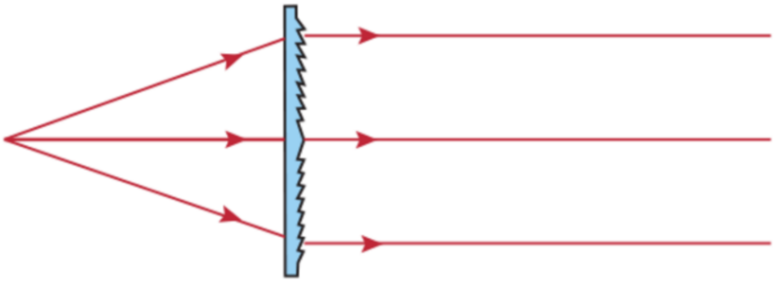


Figure 14: Representation of Fresnel Lens Diffusing Light

To choose the right fresnel lens for our design, we run a test on the intensity of the LED alone along with its detectable distance using the camera system built for the Fall 2022 CREOL Midterm Demonstration.

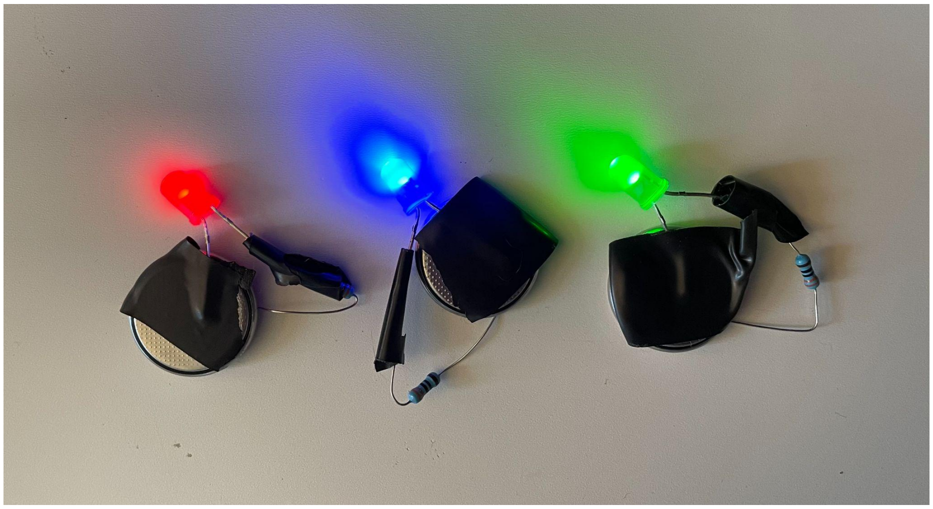


Figure 15: LED Setup

Shown in Figure 14. the three LEDs chosen for the design were taped together with a 3 V battery and a 330 Ω resistor to supply power to the LED. Once the LEDs were set up, we measured the detectable distance from the camera and Table 15. shows the results. Each trial was made at different backgrounds with different room lights.

LED	**Detetable distance (1st trial)**	**Detetable distance (2nd trial)**	**Detetable distance (3rd trial)**
Blue	12 cm	8 cm	10
Red	24 cm	13 cm	18 cm
Green	Needs more testing	Needs more testing	Needs more testing

Table 15: Results From Detectable Range Distances In Different Backgrounds

From this test, we observed the importance of incorporating a fresnel lens in the collar tag as the LED alone doesn't diffuse light in a long range. Also, the connections need to be weld instead of taped as the tape is not a good conductor and therefore absorbs some power.

Also, the green LED needs more testing as the default of the code for the camera system was green and therefore the dta we obtained when using the green LED was not meaningful.

To calculate the focal length desired for our design we used the following equations:

$$P = \frac{n_{lens} - n_o}{n_o}\left(\frac{1}{R_1} - \frac{1}{R_2}\right)$$

Equation 2: Lens maker equation

$$P = \frac{1}{f}$$

Equation 3: Power and focal length relationship

$$\frac{1}{f} = \frac{1}{Object\ distance} + \frac{1}{Image\ distance}$$

Equation 4: Imaging equation

$$M = \frac{-\,Image\ distance}{Object\ distance}$$

Equation 5: Fresnel magnification

$$\frac{I_R}{I_L} = R = \frac{\left(n_2 - n_1\right)}{\left(n_2 + n_1\right)}$$

Equation 6: Fresnel loss

$$n_1 sin\theta_1 = n_2 sin\theta_2$$

Equation 7: Snell's law

From these calculations we found the focal length to be 25 mm and to keep consistency inside the collar tag, the diameter will be 1 inch as the diameter of the battery is 1 inch as well. Another important factor we considered to choose the lens was uniformity of the light passing through the system. Since we are using three different wavelengths, uniformity among wavelengths is necessary. Below, we will look at the options considered for the fresnel lens.

5.2.2.1 LA1951 – N-BK7 Plano-Convex Lens

The LA1951 is sold by Thor labs for $25.14. It has a focal length of 25 mm and its diameter is 1 inch. Also, it works for different wavelengths.

5.2.2.2 FRP125 Lens

The FRP125 Lens is sold by Thorlabs for $23.27. It has a focal length of 25 mm and its diameter is 1 inch. It is 1.5 mm thick and is lightweight. The designed wavelength is 588 nm, which becomes an issue when we use wavelengths further away from this one.

Lens	**LA1951**	**FRP125**
Focal length (mm)	1	25
Diameter (Inch)	1	25
Efficient with multiple wavelengths	Yes	No
Cost	$25.14	$23.27

Table 16: Comparing LA1951 & FRP125 Lenses

LA1951 was chosen because although it is more expensive than the FRP125, it works best for the wavelengths we are using for our design.

5.3 Food Dispenser

The food dispenser should be able to dispense the correct amount of food. The dispensing mechanism should also be dispensing food within 60 seconds. In order to accomplish this, the dispensing system will consist of a screw and a DC motor in order to power the spinning of the screw. This screw will be attached to a food storage that contains the user's pet food (cat food or dog food). These motors will be connected to the dispensing screws, enabling the release from the stored containment when the screw is spun.

5.3.1 Food Storage

The "Pawsitive" Pet Feeder will be able to store the pet's dry food supply as well as allocating the supply to the respective bowl. In the following sections, we will talk about the Food Storage situation as it pertains to the supply itself, and then the bowl where the pet will actively eat out of. With how the project is used, a configuration of either 3 cats or 3 dogs will be considered. Therefore, for the food supply sections it will be different to food supply concerned for cats, and a food supply concerned for dogs. However, it is taken into consideration that the bowl can be used by even the largest animal.

Food storage for the dry pet food must also meet critical safety standards due to direct contact. Without the proper food storage, the food will have a possibility of contamination. With the possibility of contamination, it may lead to health concerns and death. The project's number one priority is the health, safety, and wellbeing of the animals. Therefore, extra caution and instructions are necessary for the loading of dry food into the "Pawsitive" Pet Feeder.

Instructions will also be given on how to properly clean the food storage device to ensure cleanliness of the soon-to-be-consumed food as well as the product.

5.3.1.1 Food Supply for Cats

According to Section 2.4.1, one of the main core features of the "Pawsitive" Pet Feeder is that there will be Food Storage Availability for up to 4 liters of food. For cat food, we have decided to use Meow Mix's serving suggestions for cats.

Weight of Cat (Ibs)	Dry Food Feeding Amount Per Day (Cups)
5 to 9	½ to 1
10 to 14	1 to 1 ½

Table 17: Serving Size for Cats

* Kittens (Under 1 year) should be fed 2 times the adult requirement.

* Pregnant and Nursing Cats should be fed 2-4 times the adult requirement.

The amount listed will provide the cat for their size and average activity level. However, the volume of food a cat needs to eat depends on how many calories are in its food and the energy demands of the cat. As each cat may have different conditions to their health, different formulas may be beneficial to optimize their caloric intake and health benefits. Dr. Leslie, an expert employed by Meow Mix, states:

> *"It is well documented that an individual cat may need up to 50 percent more or less than what is considered average. It is crucial to tailor the amount of food to help your cat achieve or maintain an ideal, lean body weight" – Dr. Leslie*

In addition to how much should each cat be fed daily, Dr. Leslie stated that the dry food can be nutritionally sound as long as there is fresh water available for the pet. Cat owners should consult with a veterinarian to accurately determine how much food should be fed to their cat.

5.3.1.2 Food Supply for Dogs

According to Section 2.4.1, one of the main core features of the "Pawsitive" Pet Feeder is that there will be Food Storage Availability for up to 4 liters of food. For dog food, we have decided to use Purina's serving suggestions for dogs.

Weight of Dog (Ibs)	Dry Food Feeding Amount Per Day (Cups)
3 to 12	1/3 to 1
13 to 20	1 to 1-1/3
21 to 35	1-1/3 to 2
26 to 50	2 to 2-2/3
51 to 75	2-2/3 to 3-1/3
76 to 100	3-1/3 to 4-1/4
100+	4-1/4 plus ¼ cup for each 10 lbs. of body weight over 100 lbs.

Table 18: Serving Size for Dogs

There are 3 factors that should be considered when deciding how much food to feed your dog: Weight, Activity Levels, and Age.

> ***Weight****: Maintaining an ideal body condition is important for your dog's overall health. "If your dog is not in his ideal body condition, you may need to make adjustments in what and how much you are feeding him," says Dr. Callie Harris, DVM.*
>
> ***Activity Levels****: The feeding recommendations on dog food packages are based on average adult dogs with normal activity levels. If your dog gets a lot of daily exercise by going on daily runs or hikes with you, he may need more calories to meet his*

> *energy needs. "Dogs who are less active and overweight may need a more managed daily caloric intake," according to Dr. Harris.*
>
> ***Age**: As your dog grows, his nutritional needs change. If your dog is still a puppy, he needs puppy food for growth and development. Senior [Dogs] may benefit from senior formulas that help keep older dogs active and their minds sharp.*

Similar to cat food, companies provide different formulas (from puppy to senior formulas) to optimize caloric intake health benefits. Dog owners should consult with a veterinarian to accurately determine how much food should be fed to their dog.

5.3.2 NEMA 17 Stepper Motor

The Nema 17 stepper motor provides the amount of torque needed for dispensing system to operate properly. This motor has a rated voltage of 12V DC, a step angle of 1.8 degrees, 200 steps per revolution, and a pull-in torque of 3.2 kg-cm. The NEMA 17 is able to push the amount of food from the storage area into the bowl with an assist from gravity.

5.4 Bowl and Lid System

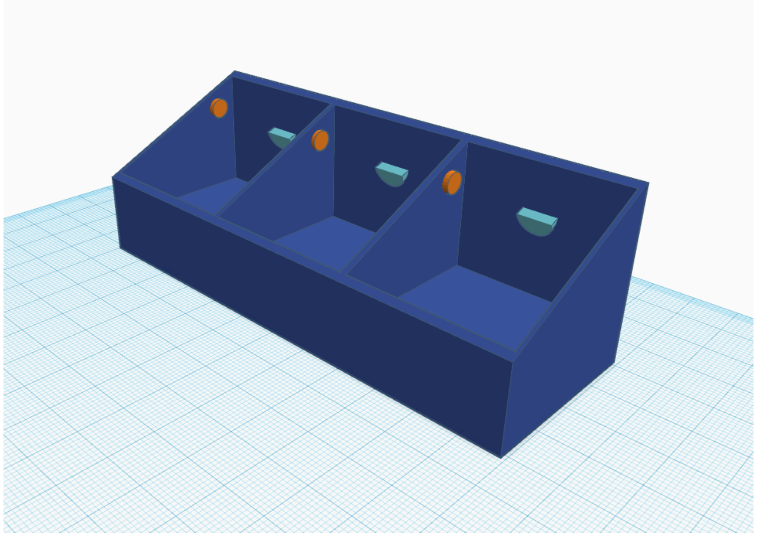


Figure 16: Bowl and Lid System Concept

The bowl and lid system are crucial to the system's design as this will be accessible to the pets. It is the bowl and lid system's job to store the desired amount of food for the pet to eat while, using a motor, raise and lower the lid for the pet to access the food. In the following sections, we will contribute research about parts we have decided to use and how this system can contribute to our goals and objectives.

5.4.1 Microcontroller

A microcontroller is a tiny integrated circuit that is found in an embedded system and has the goal of performing a certain function. We required a core microcontroller that could do calculations, store memory, and have a sufficient number of input/output pins for the device to be able to function properly so that we could integrate and operate the many peripherals that were included in the project. Controlling when the motor comes on, sending and receiving signals from the ultrasonic sensor, and sending output signals to the speaker will be the primary responsibilities of the microcontroller. It will also be responsible for receiving and transferring data from the microprocessor. It will be possible for the computer to connect with a microprocessor (section 4.9) via either the UART, SPI, or I2C protocols. The capability of the microcontroller to transmit and receive signals in a synchronous fashion is an essential characteristic of this kind of device. This indicates that a clock signal is used to keep the signals in sync with one another. The transmission of PWM, also known as pulse-width-modulation, signals require this particular form of signal. PWM signals achieve their effects by modulating the duty cycle, which refers to the proportion of time spent in the 'high' state by the signal. When it comes to regulating the motor, this kind of signal assumes a very crucial role.

Before we can choose a microcontroller for this project, we need to have a solid understanding of the precise requirements that will be imposed on the device. This comprises characteristics such as the mode of communication, the number of pins, the operating voltage, and the size of the component. It will be necessary for the microcontroller to provide support for a total of four peripherals. These peripherals include the ultrasonic sensor, the external speaker, the door motor, and the connection between the MCU and microprocessor. This indicates that at least 6 of the I/O pins that are available on the MCU will be used. This contains two pins for sending and receiving data to and from the microprocessor, at least one pin for controlling the door motor, two pins for the ECHO and TRIG pins on the ultrasonic sensor, and one pin for providing input to an external speaker. Since an increase in the number of pins on a chip would result in an increase in price, we are not seeking for the module that has the most pins when we are searching for a microcontroller; rather, we are searching for the module that will only fulfill our needs.

We need a module that has adequate speed to quickly communicate back and forth with the development kit and be able to swiftly manage the peripherals so that the speed of the microcontroller may be optimized. However, it is not required to have a CPU speed that is really fast since doing so would just result in an increase in the cost of the gadget.

A significant need for the project is to locate a microcontroller that can operate in a low-power mode. Because our marketing and engineering criteria describe the qualities that a device must have, we need to discover a microcontroller that has the capability of switching into a low-power mode when it is not actively doing computation or transmitting or receiving signals from the peripherals.

When thinking about memory and RAM, because the MCU will not be keeping data but instead will only be transmitting data to the microprocessor, where it will be saved on external memory, the memory requirements of the MCU does not need to be particularly big because it is not necessary for the MCU to store data. Random Access Memory is what should be prioritized. The RAM size is 2KB on average for a microcontroller. This indicates that they have 2048 distinct units that may be addressed separately and that they can write or read 8 bits at a time. In addition, this indicates that there are 11 address lines and 8 data lines in the system.

5.4.1.1 Arduino Uno Rev 3

The Arduino Uno Rev 3 will have the capability to provide the required hardware level operations for the pet feeder, as well as turning on and off the motor when the proper color data is obtained and activating it when it is received and the mechanism for shutting the lid. This microcontroller will be incapable of managing the image processing responsibilities based on the information that is obtained straight from the camera, therefore an extra CPU is required in order to process the image's color and then transmit that processed color data to the Arduino. When carrying out this project, the use of an Arduino will make it possible to apply a variety of color detection and imaging libraries that are still open to the public (due to the open-source nature of Arduino programming). The the Arduino Uno Rev 3's inbuilt features include the following:

- A single 16-bit timer that supports comparison and capture modes respectively
- Maximum clock speeds of 16 MHz
- Two 8-bit timers, each of which have a compare mode
- Interfaces via the SPI, I2C, and UART protocols
- Operating Voltage between 1.8V and 5.5V
- 10-bit Analog to Digital converter

5.4.1.2 MSP430FR6989

The MSP430FR6989 is a microcontroller that was developed by Texas Instruments and is part of a series of boards. This specific circuit board is a 16-MHz module that has a RAM capacity of 2 KB. In active mode, this microprocessor has a current consumption that can go up to 100 uA/MHz, but in sleep mode, it can use as little as 0.4 uA. The LCD display that is included in the MSP430FR6989 Launchpad Development kit is one of the many reasons why this kit is so beneficial to the process of developing electrical products. If we wanted to be able to examine the data captured on the microcontroller before it was transferred to the development kit, this may come in very helpful for us. This particular microcontroller has support for all three of the aforementioned kinds of communication protocols—I2C, UART, and SPI. Because this launchpad has been required in previous classes, all CpE members of our group already have one. This would be beneficial to the budget of our project because it would allow us to prototype with this microcontroller rather than having to buy a new unit that would be used solely for prototyping.

One other thing to keep in mind is that all CpE members of the group already have a book that covers the fundamentals of the MSP430 microcontroller. When it comes to the process of designing the functionality of the microcontroller, this information book would be of great use. In general, the MSP430FR6989 is an excellent option for our microcontroller since it has a vast array of features and has shown to be very popular for use in senior design projects.

5.4.1.3 MSP430G2553

Similar to the MSP430FR6989, the MSP430G2553 is a microcontroller that consumes very little power. The central processing unit (CPU) and registers of this device both have a 16-bit width. In addition to that, this microcontroller has a digitally regulated oscillator, which makes it possible for the system to be roused from its low-power state in a matter of milliseconds rather than seconds. Similar to the MSP430FR6989, the CpE members of our group have expertise with this microprocessor from prior courses; hence, this would be a significant advantage to picking this module as opposed to modules that we have not had experience with in the past. However, in comparison to the Arduino Uno Rev 3 microcontroller, this module has far less community support. When it comes to the research and development of the programs for the project, this would be a disadvantage. When operating in active mode, the CPU requires 2.2 V and 230 uA to function at 1 MHz. This falls to 0.5 uA while the device is in standby mode. Its active mode is marginally less efficient than the MSP430FR6989's active mode, with a current consumption that is 130 uA more than that of the MSP430FR6989. This microcontroller might be a fantastic choice for the one we're going to use in our design since it offers five distinct ways to save power. In a manner similar to that of the MSP430FR6989, the CpE members of the group already own the TI Launchpad kit that contains this microcontroller; hence, it would not be necessary to spend money on a new unit in order to prototype.

5.3.1 Microprocessor

A computer processor known as a microprocessor is one in which the data processing logic and control are contained on a single integrated circuit or a limited number of integrated circuits.

Microprocessors are typically very compact and inexpensive. A computer's microprocessor is the component that houses the arithmetic, logic, and control circuitry needed to carry out the duties of the central processing unit (CPU) of the computer. In addition to being able to do arithmetic operations and understand and execute program instructions, the integrated circuit also has this capability. The microprocessor is a digital integrated circuit that can perform multiple tasks, is driven by a clock, is based on registers, and can accept binary data as input. It then processes this data in accordance with the instructions that are stored in its memory and outputs the results (also in binary form). Combinational logic and sequential digital logic are both included in microprocessors, and they perform their operations on numbers and symbols that are represented using the binary number system.

5.1.1.1 Raspberry Pi 4

The Raspberry Pi 4 is the most recent iteration of the inexpensive Raspberry Pi microprocessor. It may be connected to a display with a straight plug-and-play connection and can make use of standard computer accessories like a keyboard and mouse. The Raspberry Pi 4 is capable of doing all of the fundamental tasks of a computer, such as navigating the internet, playing high-definition video, making spreadsheets, typing documents, and playing games, among other things. In order to operate and make use of all of the functionalities that a Raspberry Pi 4 is capable of, the programming language Python is used the majority of the time, and the Raspberry Pi 4 runs its own operating system (OS) called Raspbian. The Raspberry Pi 4 is equipped with two, four, or eight gigabytes of random access memory (RAM), as well as USB 2.0 and 3.0 connectors, Micro HDMI ports that are capable of supporting 4K resolution, a USB-C power supply, and a Gigabit Ethernet port.

5.4.2 Food Bowl

As the Pet Feeder will feed both dog and cat breeds, the project will focus on feeding the largest possible animal.

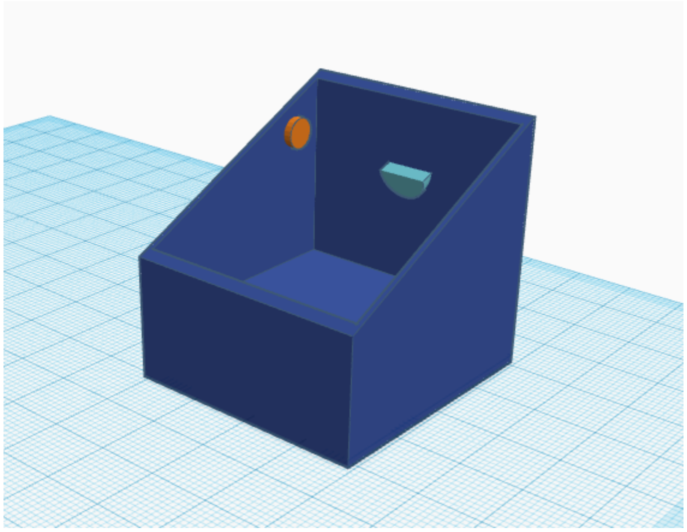


Figure 17: 3D Printed Bowl Concept

	Color	Component
1	Blue	Bowl
2	Orange	Lid Motor Placement
3	Cyan	Food Dispense Output

Table 19: Figure 15's Color Key for 3D Printed Bowl Concept

Shown in Figure 13, this is a concept model for one of the bowls. The food bowl will be 3D printed as this will be a custom bowl made for the project. According to Section 5.1.1.2, the largest animal may reach above 4 ¼ cups of dry food. Therefore, all bowls must be able to withstand a large amount of dry food in the case of having 3 large animals at a time. Although a bowl with a large food capacity will be used, the bowl will be accessible for all pets on the project.

5.4.3 Lid

The lid of the product will be designed to ensure that interference will not occur with the food dispensers and will open and close with ease. The lid will be 3D printed to warrant product compatibility with the bowl and the rest of the system. Furthermore, the lid must meet the health and safety standards for storing dry food.

The lid also has the responsibility to not open if the pet attempts to. It is important that the lid is snug and flush with the top of the bowl system whereas the pet cannot force entry to the bowl and destroy the lid system. The use of the lid motor is paramount to the lid system as it will have the ability to open and close the lid. Therefore, the motor should not activate to open or close if it risked injuring the animal that needs to be fed. As we are designing 3 lid systems, we must also ensure that the right lid will open as each Pet may have food that is tailored to a specific pet. Failure for the designated lid to open, may result in the wrong feed being fed to the wrong pet or no feed to appear in the designated bowl. This can lead to numerous effects which may include starving a pet or feeding a pet that is custom-made for another pet and having negative effects.

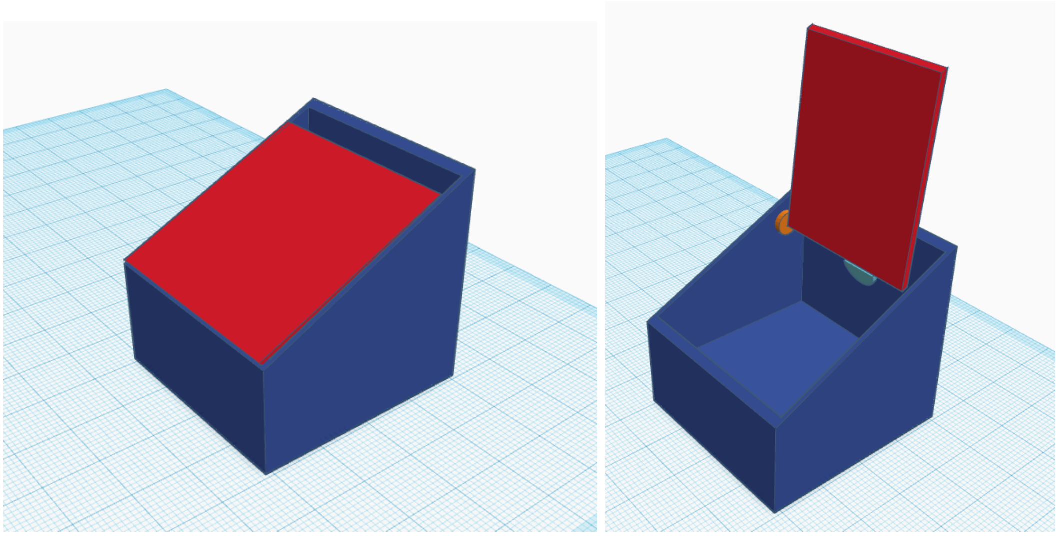


Figure 18 & 19: 3D Printed Lid Concept with Motor Closed and Open

	Color	Component
1	Blue	Bowl
2	Cyan	Food Dispenser Output

3	Orange	Lid Motor
4	Red	Lid
5	Orange	Lid

Table 20: Figure 16 & 17's Color Key for 3D Printed Lid Concept with Motor

5.4.3.1 Lid Motor

The purpose of the lid motor is to rotate the lid open and closed. When choosing the lid motor for the project we will be looking at 4 areas: cost, power, speed, and compatibility. With our budget allotted, the lid motor has to be cost effective. With a total budget of $400, and considering components might be replaced, a cost-effective option will be necessary. We do not want to compromise the quality of the motor for a lower price point. Additionally, as we are using 3 bowls, there will have to be a motor for each bowl's lid.

The amount of power is important for the lid motor. The overall power consumption of the "Pawsitive" Pet Feeder will be designed to be around 100 W. With a set power budget, we need to ensure that the motors does not consume most of the power budget and still be able to do it's required functions.

As stated in Section 2.3, a required specification is that the automatic lid should close about 60 seconds after the pet leaves. Therefore, speed and responsiveness is important. If the lid is too slow during the closing procedure, the pet may get hurt as they would like to get more food through the dispenser. If the lid is too fast, it has a possibility to injure the pet with the lid's force created by its speed.

As we will be 3D printing components and using off-the-shelf components, compatibility of the motor to the rest of the components will be required. Compatibility is important because these components must work in harmony for the project to work. The motor's connectivity to the board must also be manageable in terms of power and programmability for the desired specifications of the project. Compatibility is also important in the event a component needs to be switched out due to unforeseen circumstances or normal wear and tear.

5.4.3.1.1 HiLetgo ULN3003 4-Phase Stepper Motor with 5V Drive Board

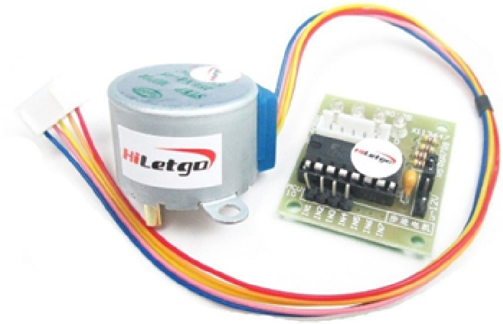


Figure 20: HiLetgo ULN 3003 4-Phase Stepper Motor with 5V Drive Board (Courtesy of Amazon)

The HiLetgo ULN2003 4-Phase Stepper Motor with 5V Drive Board is sold on Amazon for $14.59. It is a speed down stepping motor that can choose external voltage for power supply via pins. The motor has a diameter of 28mm and a voltage of 5V. The stepping angle is 5.625 x 1/64, with a speed reduction ratio of 1/64. When the item is shipped to the group, it will come with the motor driving board that is welded and tested, and a 5V four phase five lines stepping motor.

5.4.3.2 Texas Instruments Innovation Lab at UCF

The Innovation Lab at UCF allows students to bring their ideas to build prototypes with 3D printers, laser cutters, Texas Instrument components and equipment and other high-tech machines. The lab also provides lab technicians that can show us how to properly use lab equipment to maintain our safety. The Innovation Lab is open Monday through Friday from 10am to 6pm. We will need to provide the lab technicians with a .DXF file on a flash drive to aid in any projects in the lab.

5.4.4 Photodiode

For this project, photodiodes will be operated with a reverse bias voltage to increase both the depletion width and the electric field to improve the quantum efficiency and the speed of the response. As we will be using IR LEDs around the bowl we chose 950 nm as the working wavelength for the photodiodes

5.4.4.1 S2386-5K

This photodiode is sold by HAMAMATSU for $12.09. It has two pins, a wavelength peak sensitivity of 960 nm, dark current of 5 pA.

5.4.4.2 BPV23FL

This photodiode is sold by Digikey for $1.05. It has an spectral range of 870 - 1050 nm, dark current of 2nA, and viewing angle of 120°.

5.4.4.3 BPW82

This photodiode is sold by Vishay for $1.04. It has a peak wavelength of 950 nm, dark current of 2nA, and rise time of 100 ns.

5.5 Coding Languages

Choosing the right programming language(s) for our project is important if we want to make a system that works well as a whole. This part shows all the different ways to code that can be used. So, the team has a way to design, develop, or prepare our project to get to the end result we want, as well as a detailed comparison table at the end to list, show, and compare all the unique differences between the coding languages that could be used.

5.5.1 Python

Python is a beginner friendly object-oriented programming language which has a variety of libraries. It also has great statistical capabilities, which efficiently solve many math problems. It also offers a variety of libraries for visualization of various plots, such as histogram, bar plots, scatter plots, and much more. These libraries include, but are not limited to, matplotlib, and seaborn. Python has many applications, such as web development, scripting, data science, and artificial intelligence.

It also offers libraries in machine learning, deep learning, and reinforcement learning. Such libraries include tensorflow, Keras, and Theano. This will be helpful for detecting the pet in our app. We will train a convolutional neural network (CNN) with many images of pets. When the model is accurate enough, we can connect it to a camera. The input from the camera will be used as input to detect the pet and perform the desired result.

5.5.2 C#

C# is a .NET based programming language developed by Microsoft. It is object-oriented, and component based. It offers many features such as garbage collection, nullable types, exception handling, lambda expression, LINQ, and much more. It uses concepts of types, namespaces, members, and assemblies to build its program. It is used for website development, windows development, game development, and much more.

5.5.3 C

C is a general-purpose computer programming language created in the 1970s by Dennis Ritche. It has huge applications in the field of operating systems, device drivers, and protocol stacks. It supports structured programming, lexical variable scope, and recursion. It also provides low-level access to memory, which is useful for memory management in the computer, using pointers and dynamic memory allocation. Many languages such as python, java, JavaScript, etc. are influenced by or built upon the concepts of C. Some limitations of C include string handling, exception handling, type checking, etc.

5.5.4 C++

C++ is a middle-level programming language developed by Bjarne Stroustrup in 1979 at Bell Labs. It is cross-platform, meaning it can run on Windows, Mac, or Linux. It offers many features similar to C, such as memory management, recursion, etc. The thing which stands out is that C++ is object-oriented. Being object-oriented means, it has support for concepts such as classes, inheritance, and polymorphism. It has applications in many sectors of the digital world, such as app development, computation programming, game development, and embedded systems.

5.5.5 Java

Java is an easy-to-learn, object-oriented, platform-independent programming language. It offers a wide range of applications in software development, such as:

- Mobile applications: IDEs, such as Android Studio, and Kotlin, are useful to create mobile applications using Java. With its OOP principles, it offers improved security and the ability to convert to application packages (APKs).
- Desktop GUI applications: Software such as AWT, Swing, and JavaFX are useful for creating desktop applications using Java. They offer pre-built tools such as menu, list, button, scroll panes, tables, and trees.

Other applications include web-based, enterprise, scientific, gaming, big data, business, distributed, and cloud-based.

5.5.6 Swift

Swift is a new programming language that may be used to construct applications for iOS, macOS, watchOS, and tvOS. There are many parts of Swift that are familiar with the programming expertise you have with C and Objective-C. The creators of Swift have taken elements from a variety of other commonly used programming languages, including Objective-C, Rust, Haskell, Ruby, PYTHON, and CLU, amongst others. The most cutting-edge ideas in current language design have been included into the development of Apple's Swift programming language. The performance of the compiler has been optimized, and there have been no concessions made in the development of the language.

Swift also has optional sorts that may be used to handle situations in which there is no value. Either "there is value" or "there is no value at all" might be used as the optional phrase in this sentence. The use of options is analogous to the usage of the value nil with regard to Objective-C pointers; however, they are not just functional for classes but for any type. Optional features are not just safer and more expressive than none in Goal-C, but they are also key to many of the most powerful aspects of Swift. Goal-C does not have any optional characteristics.

Swift is a strongly typed programming language that gives you the ability to comprehend the many sorts of values with which your code interacts. If you need to use a String in any portion of your code, using type safety will prevent you from using an integer instead. In a similar vein, type safety prevents you from accidentally sending an optional string to a section of code that expects a string that is not optional. Using type safety throughout the design process gives you the ability to identify and correct errors as quickly as feasible.

5.5.7 HTML

HTML is a markup language which is useful in creating a base model for web pages. It is used by nearly all websites on the internet. In terms of text, it offers a variety of headings, text formats, font colors, sizes, and more. It is also helpful to create forms, which can take in user input and send requests to servers. It is a major language used in the frontend section of application development. It also offers the capability to link pages, and traverse through web pages, based on button click or user input.

5.5.8 CSS

CSS is abbreviated for Cascading Style Sheets. They are used to style web pages and give them a layout. It offers all styling tools provided by HTML. The reason CSS is preferred over HTML for styling is due to information distribution. Inline HTML styling creates a cluttered code which is hard to understand for a future developer who wants to work on the project. This is harmful for long term maintenance of the code.

Using keywords such as id and class offers us the opportunity to select specific elements in the HTML code. These keywords can be used as a reference to style elements based on their id or class. This also helps to style multiple elements with the same class name. CSS offers layout tools such as flex, grid, and much more.

5.5.9 JavaScript

JavaScript is a scripting language with an amazing developer community. It allows use of complex functions and programming concepts within web pages. JS is used for creating the frontend and the API of a webpage. It offers functionality to a page. It uses various libraries to call functions to connect to pre-built or developer-created databases. It can be used to perform complex mathematical questions or return a search query.

The applications of JS for web related software are immense. It offers the ability to display interactive maps, display animated 2D/3D graphics, create chatbots, buy/sell stuff on the internet, provide an immersive experience of knowledge, and much more. "The latest survey in 2022 states that JavaScript is the most used programming language, with a 78% usage rate among developers. This survey is reported by the State of JS Survey."

5.5.10 MATLAB

MATLAB is an interactive system using an array as its fundamental data element. Arrays in MATLAB do not need to have their dimensions specified. Because of this, you are able to solve a large number of technical computing issues, particularly those that involve the formation of matrices and vectors, in a far shorter amount of time than it would take you to develop a program in a scalar noninteractive language such as C or Fortran. The acronym for matrix laboratory is more often known as MATLAB. The LINPACK and EISPACK projects, which together constitute the state-of-the-art in software for matrix calculation, were responsible for the initial development of MATLAB, which was first intended to enable simple access to matrix software produced by both projects. MATLAB has developed over the course of several years thanks to the contributions of its numerous users. It is the educational tool of choice in university settings for beginning as well as more advanced levels of science, mathematics, and engineering classes. When it comes to high-productivity research, development, and analysis in the commercial sector, MATLAB is the tool of choice.

5.5.11 Haskell

Many coding language developers regard Haskell as a general-purpose, statically typed, purely functional programming language with type inference and lazy evaluation. In simpler terms, it is a very simple, advanced functional language that allows you to write code in a very simple manner, a sort of you get what you type coding language.

In contrast to other programming languages that focus on "how to solve," Haskell focuses solely on "what to solve." As the programming language is built on the foundation of combinatory logic, Haskell allows functions to be solved in a very mathematical sense.

Because the code compiler evaluates the code before the runtime, the Haskell code is evaluated in real compilation time, allowing the finished code to be very correct, clean, and concise. Shorter development time, tight control of side effects, and scalability, as well as Haskell being a cleaner and more reliable coding language than most, all entice coding developers to use this code to manage, structure, and scale large amounts of data.

5.5.12 R

R is a free programming language designed specifically for statistical computing environments and graphical use in GUI work. R is highly extensible and provides a wide range of statistical (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, and so on) and graphical techniques.

R is also used to create and further implement machine learning algorithms for the programming developer to investigate, refine, and analyze the given and imputed data. To make it user friendly for all developers using the language and environment, data analysis in R is done in very understandable and simple to understand steps.

These steps are programming, transforming, discovering, modeling, and communicating the results, which are all done and completed in various forms of procedures.
As the developers use R to program their language written in code, they transform the code into a collection of libraries designed specifically for data science tailored to their own use, resulting in the discovery of a deeper dive into their own data with additional analysis to better model their refined data to a greater capture of the essence of what is desired to be displayed. Finally, communicate the results by generating a report from the R environment's data.

5.5.13 PHP

PHP, also known as Hypertext Preprocessor, is a free and open-source scripting language that is used by many programming developers for web development. It is very similar to other web development programming languages such as HTML, CSS, and JavaScript. In fact, PHP pages contain HTML with embedded code that can be filled with functions that can produce methods by the code running on the server and generating the HTML sent to the client.

When discussing the PHP coding language, it is also worth noting that they generate PHP files that contain text, HTML, CSS, JavaScript, and of course PHP code itself, which is then executed on the server and returned to the browser as a plain HTML page.

5.5.14 Coding Language Comparison

The differences between the coding languages that could be used will be compared in the tables below. One more thing to note is that, except for MATLAB, all of the languages mentioned are free to use. The cost comparison was left out of all of the tables, even the one that included MATLAB, because there was no cost difference in any of the other tables. Students have to pay $49.00 for MATLAB. Also, the next tables will be set up so that they flow into the next one, and the comparisons for each coding language will be the same. In the title of the table, it will say if it is a continuation of the last table or if it is the first table in a comparison of three different programming languages. The chosen programming language(s) that will be used to design this project will not be written out in the tables. Instead, they will be explained in more detail in the section of this report called "Design."

	C	**C++**	**C#**
Paradigm	Imperative	Imperative, Object Oriented, Generic	Imperative, Object Oriented, Generic, Reflective, Functional, Event-Driven
Standardized	Yes	Yes	Yes
Type Checking	Static	Static, Dynamic	Static, Dynamic (for interop)
Parameter Passing Methods	By value, by reference (through pointers)	By value, by reference (through reference types)	By value, by reference (through managed pointers [explicitly in, out or in-out])

Intent For Use	System, Embedded	System, Embedded	System, Embedded
Design Goal	Low Level Access, Minimal Constraint	Abstraction, Efficiency, Compatibility	Rapid Application Development

Table 21: Coding Language Comparison (C , C++ and C#)

	Java	**JavaScript**	**Python**
Paradigm	Imperative, Object Oriented, Generic, Reflective	Imperative, Object Oriented, Functional, Reflective	Imperative, Functional, Reflective, Array Programming
Standardized	Yes	Yes	No
Type Checking	Static	Dynamic	Dynamic
Parameter Passing Methods	By value	By value	By value (Call by object reference)
Intent For Use	System, Embedded	Client-side web scripting	System, Embedded
Design Goal	Write once, run anywhere	N/A	Simplicity, Readability, Expressiveness, Modularity

Table 22: Coding Language Comparison (Java, JavaScript and Python)

	MATLAB	**HTML**	**CSS**
Paradigm	Procedural, Imperative, Array Programming	Markup, Declarative, Domain-Specific Programming Language	Markup, Declarative, Domain-Specific Programming Language
Standardized	No	Yes	Yes
Type Checking	Dynamic	Static	Static
Parameter Passing Methods	By value	Tags	Tags
Intent For Use	Numeric computation and visualization	Web development	Web development
Design Goal	At the beginning, it was designed as an interpreter for easy use of Fortran libraries. Nowadays, it is used for high-performance numerical analysis and visualization	Lingua Franca for the Web, Simplicity, Scalability, Platform Independent, Content is not for presentation markup, Support of Cascaded Style Sheets, Support of the Visually Impaired	Allows a browser engine to paint elements of the page with specific features (colors, positions or decorations)

Table 23: Coding Language Comparison (MATLAB, HTML and CSS)

	Haskell	**R**	**PHP**
Paradigm	Functional, Generic, Lazy Evaluation	Imperative, Functional, Reflective, Array Programming	Imperative, Object Oriented, Reflective
Standardized	Yes	Yes	No
Type Checking	Static	Dynamic	Dynamic
Parameter Passing Methods	By value, By reference	Value by need, by name (programmer chosen)	By value, By reference
Intent For Use	System, Embedded	Statistics, Numerical Computation, Visualization, Education	Web Application
Design Goal	Lazy Evaluation, Teaching and Research, completely formally described as Report Preface	Expressiveness, Interactive, Manipulative, and Analysis of Datasets	Robustness and Simplicity

Table 23: Coding Language Comparison (Haskell, R and PHP)

5.6 Programming Language Editing Software

It is critical to correctly select which programming language editing software to use in order to properly create, edit, and debug the necessary code for the digital aspect of our feeding system to function efficiently.

This section showcases all of the various programming language editing software that can be used to design, develop, or prepare our final product. In addition, at the end, there is a detailed comparison table that lists, displays, and, of course, compares all of the unique and distinct differences between the programming language editing software that could ultimately be used for our feeder.

5.6.1 Visual Studio Code

Visual Studio Code is an editor for source code that supports a wide range of programming languages. These languages include Java, JavaScript, Go, Node.js, Python, C++, C, Rust, and Fortran. The Electron framework, which is used for the development of Node.js web applications that are powered by the Blink layout engine, is the foundation on which it is founded. Visual Studio Code is pre-configured with support for the fundamentals of the majority of programming languages used today. This fundamental assistance encompasses features such as syntax highlighting, matching brackets, code folding, and customizable snippets. In addition, the default installation of Visual Studio Code includes support for debugging in Node.js, as well as IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML. Extensions that are freely downloadable from the Visual Studio Code Marketplace may be used to extend support for new languages.

5.6.2 Repl.it

Repl.it is an online integrated development environment (IDE). It is a Software as a Service (SaaS), which offers the ability to create programming projects online. It supports many languages, such as C, C++, Python, Java, and much more. It also offers features included in other offline IDE's such as version control, collaborative coding, syntax highlighting, debugging, hosting, unit testing, package management, etc. It is highly user-friendly and has a huge community of developers using it around the globe. It is mainly used by school/college education institutions to teach students how to code in a collaborative manner.

5.6.3 Atom

Atom is a cross-platform editing IDE which works across all operating systems. It contains a built-in package manager which is applicable to install new packages/libraries from Atom or create your own. It also provides many other features such as smart autocompletion, file system browser, multiple panes, etc. It supports nearly all languages such as C, C++, C#, COBOL, Java, JavaScript, and much more.

5.6.4 Notepad++

Notepad++ is a source code editor and Notepad replacement that works with many programming languages, such as Java, C, Python, etc. Notepad++ runs in a Microsoft Windows environment. It was written in C++ when it was first made, and it uses the Win32 API and STL. This gives it a guaranteed fast execution runtime and a small program size so it can run on the platform. Notepad++ is free, yes, free, and it also supports tabbed editing, which lets you work on multiple open files in a single window.

5.6.5 Sublime Text

Sublime Text is a complex text editor that many coding developers are familiar with. This editor is frequently used by code developers to write code in various programming languages such as Java, C, and Python. Sublime Text also supports front-end web development languages such as HTML, CSS, and JavaScript.

Sublime Text includes many features that aid in the arduous task of coding many lines of text, such as Syntax Highlight for the code that is implemented in a function, Auto Indentation for the structural organization of the key elements in the code, File Type Recognition for the creative and organizational orientation of the code being used, Sidebar for easy access to all the code involved in the project, and so on. Unfortunately, users can download and evaluate the Sublime Text programming language editing software for free; however, in order to use the program on a regular and consistent basis, the product must be licensed once at a cost of $99.

5.6.6 Programming Language Editing Software Comparison

The following comparison tables will help us decide which software we will use to change the code for our Pet Feeder project design. The cost comparison was taken out of the tables, just like the coding language comparisons, because Sublime Text was the only editor that wasn't free (it costs about $99).

After looking at Tables x and x+1, it is clear that Atom and Visual Studio Code are the best code editors for us to use to write and fix our code. Both options are free, easy to install (because it does not take up much space), and have features that will speed up the programming process. Atom's cross-platform editing was the feature that stood out the most. This would be useful if more than one person in a group wanted to help develop software on different devices. In the case of Visual Studio Code, its auto completion feature and ability to debug with breakpoints, along with the fact that it is a lightweight program, made it another top choice for developing software for our project.

	Atom	**VSCode**	**Repl.it**	**Sublime Text**	**Notepad++**
Programming Languages	Supports Many Languages	Supports Many Languages	Supports Many Languages	Supports Many Languages	PHP, JavaScript, HTML, CSS
OS	Windows, Linux, Mac OS	Windows, Linux, Mac OS	Online IDE Editor, Compiler, Interpreter	Windows, Linux, Mac OS	Windows, Linux, Mac OS (By using a 3rd party tool), UNIX
Best Features	Cross platform editing. Built-in package manager	Auto-comp letion debugging with breakpoints	Can code, compile, run, and host all online	Provides instant switching between projects. Cross platform support	Syntax Highlighting, Auto Indentation, Auto Completion
Written In	Built using web technologies	TypeScript, JavaScript, CSS	Built using web technologies	C++ and Python	C++ and uses Win32 API & STL

Table 25: Programming Language Editing Software Comparison

5.7 OS Distribution Software

Choosing the right OS distribution software is critical for properly compiling the written code to run and function in order to create a fully functional fully digital system for our feeder. This section displays all of the available OS distribution software that can be used to design, develop, or prepare our final product. In addition, there is a detailed comparison table at the end.

5.7.1 Ubuntu

Ubuntu is a Linux based terminal application which is used by many experienced developers. It offers high open-source security for full stack. It contains optimized kernels for AWS, Azure, Google, Oracle, and IBM. It also has cloud workload migration services, and enterprise support maintenance.

5.7.2 Debian

Debian, also known as Debian GNU/Linux, is a Linux OS distribution software that is made up of a free source and one of the first of its kind open-source software. It was made in 1993, making it one of the first of its kind to be made for the developers to use for their own project. This paved the way for other OS distribution software to be made in the future.

5.7.3 OS Distribution Comparison

The table below will fully list, display, and, of course, compare all of the unique and distinct differences between the OS distribution software that could be used for our final product.

	Ubuntu	Debian
Installation	Stable and Streamlined	Iterative and Interactive
Target Audience	General, Server, Desktop, Supercomputer, IBM Mainframe	General, Server, Desktop

Supported Architectures	armhf, I686, PowerPC, ppc64el, s390x, x86-64	x86, x86-64, arm, ppc64, loongson, mips, sh, s390x, risev
Cost	$0	$0

Table 26: OS Distribution Comparison

5.8 Web Application Development Software

Choosing the right web application development software is important if you want to make a digital system that everyone can use on the internet.They have a lot of benefits, one of which is that they can do all the necessary tasks using a web browser instead of software that needs to be installed. This makes them easy to use and set up for the user. Plus, cloud-based functionality is making web applications an important part of using products in today's growing world.

Functional systems are making it possible for developers to create unique web applications that can be published online. This helps developers meet the needs, wants, and desires of their applications. Web applications also get rid of the worry about whether the mobile app for the product works well and efficiently on different platforms. Because it works on multiple platforms, web applications are no longer a luxury, but a necessity, especially for smart devices. The ability to update themselves without the user having to do anything is also a good thing for future improvements to the product interface. This section shows all the different kinds of web application development software tools that can be used to design, build, or prepare the final product of our project.

5.8.1 ReactJS

ReactJS is a JavaScript library for building UI interfaces. It is free, open-source, and component-based. It is used for creating single/multi page web, mobile, or server-rendered applications. It has many features such as react hooks, class-based components, virtual DOM, and much more. We can install many libraries/components from the internet into ReactJS for our use.

5.8.2 React Native

React Native is an open-source UI software framework that many people use to build unique web apps that are structurally real and natively considered as a render. Web application programmers can use it as a sort of structure guide for their projects to make the best use of the JavaScript platform's features.

5.8.3 Node.js

Node.js is open-source software that can run code on a number of different platforms. In particular, JavaScript code that is not in the browser. Quite often, we use Node to make API back-end services, which are services that power client-based applications like web apps. The surfaces of these apps need back-end services to run a service that works for all users. Node.js is great for agile development because it is quick and can be used for a wide range of large code applications.

5.8.4 Express

Express is a framework for building web apps with Node.js. It is free and open-source software because it is a part of Node.js. Its purpose is to help developers build web applications and APIs for their own projects. Most web developers know it as the most popular Node web framework. It gives developers a unique way to write handlers for requests that figure out the right reference to the web application to post on the internet.

5.8.5 Bootstrap

Bootstrap offers a variety of styling effects using the concept of class names. It can create buttons using the ‘btn’ class name. It offers grid layout capabilities using ‘row’ and ‘ col’ dividing the web page in 12 columns. Bootstrap is used by 21.7% of all websites. It is reportedly used by 42949 tech companies including Spotify, Udemy, and Twitter. It takes away the inconvenience of styling everything using CSS.

5.8.6 SwaggerHub

SwaggerHub is an integrated API development platform which is capable of building, documenting, managing, and deploying APIs. It is highly useful for testing API calls to the database. We can run an API on a terminal and try to make a change to the database using an API function call. Without the presence of the front end, it is hard to make such calls. SwaggerHub gives us the opportunity to check the functionality of these calls without creating the front end.

5.9 Mobile Application Development Software

Choosing the right mobile application development software is important if you want to make a digital system that works well and can be used by anyone on a mobile device. With more and more people using smartphones and tablets to connect to the Internet, mobile app development is a great way to reach many people who might be interested in our smart product.

Mobile applications are different from web apps made for desktop computers because they do not use integrated software systems. Instead, each mobile app has a single, limited function that works with the product. Mobile apps are usually 1.5 times faster than web apps, and developers can code different features that use the built-in hardware of the device being used, such as the camera, GPS, accelerometer, gyro meter, fingerprint reader, facial recognition, etc.

The ability to update and push notifications to the application user whenever the product needs or wants it, as well as the ability to provide an easy-to-use interface for more complicated features, can often make it easier for product and application users to feel less scared and more confident about being able to control their specific features remotely. This section shows the mobile application development software tool that can be used to design, build, or prepare our project.

5.9.1 Flutter

Flutter is a UI Toolkit that makes it easy for mobile developers to make beautiful interfaces for either IOS or Android. It also makes it possible to design for different screen sizes and devices. Flutter lets mobile developers make creative and unique apps that work well on all kinds of devices with different screen sizes and access ratios. Flutter first asks for a blank window, which it uses as a blank canvas. It then draws on the blank canvas, adding beautiful widgets that make a mobile app look like it was made in a native way.

5.10 Database Management System

Choosing the right database management system software is important for making a digital system that works well and can be used by everyone over the internet.This step is very important if you want to store and organize a database that all server-based web and app systems can use. This section shows all the different database management system software tools that can be used to design, build, or prepare our project. As well as a comparison table with lots of details at the end.

5.10.1 Heroku

Heroku is a container-based cloud Platform as a Service (PaaS). It provides developers the ability to run various programming languages, such as python, java, and php, on its cloud server. It helps to build, manage, deploy, and scale their applications on the internet. Heroku also offers many other services such as Heroku Postgres, Heroku Redis, Heroku Teams, Heroku Enterprise, Heroku Connect, etc.

5.10.2 MongoDB

MongoDB is a SQL based database. It creates collections within databases to store specific kinds of data. For example, in a database named pets, I can have a collection of dogs, cats, rabbits, and more, with each collection containing images of the specified animal. This database can be used by the AI model to train to detect a pet in an image. We can also use MongoDB to store user information such as their email address, password, name, etc.

5.10.3 Firebase

Firebase is a development platform made by Google that lets developers build unique, high-quality apps for their web-based platform. Most people agree that this online platform is best used as a well-organized and safe document database that both new and experienced developers can use for their projects. Firebase Compass is a graphical user interface which offers an intuitive visualization of the databases. It also has the functionality to import database files using json format.

5.10.4 MySQL

MySQL is a kind of document database management system that is based on the client-server model of the product that is being used. MySQL lets individual developers build fast, efficient, secure document database storage unit systems quickly and accurately within the OS that runs the database system. This is because the program syntax stays organized, and query values in the database are written in a way that the coding languages in the structured virtual data storages can understand.

5.10.5 Database Management System Comparison

The differences between the database management software that could be used for our final product are all listed, shown, and compared in the table below.

	Firebase	Heroku	MongoDB	MySQL
Data Representation	JOSN Objects	PostgreSQL	JSON Documents	Tables and Rows
Query Language	NoSQL	SQL and NoSQL	JavaScript	SQL
Ideal Use	Managed backend infrastructure, production ready environment, NoSQL easy to use interface	Higher level of flexibility, real-time logs, and NoSQL and SQL Databases	Unstructured non-relational rapid growth database	Traditionally structured relational database
Data Security	Strong	Strong	Weak – Normal	Strong

Cost	$0	$7 / $25 / $50 / $250 Monthly Payment Plans	$0	$0

Table 27: Database Management Software Comparison

5.11 Artificial Intelligence Software

Our project requires image recognition in order to detect and recognize pets. Using the right model is very crucial in this step. Many machine learning, and deep learning frameworks provide ways to create such models. However, each one has its own intricacies, such as loss functions, activation functions, convolutional layers, pooling layers, kernels, and much more. Hence we need to decide which framework would be adequate for our application, and help analyze the data received in the most generalized manner. We would also need to look into software applications which would make coding easier and efficient.

5.11.1 Keras

Keras is a Python-based deep learning framework that makes it easy to construct and train almost any form of deep learning model. Keras is an application programming interface (API) for high-level neural networks that is written in Python. It is compatible with Tensorflow, Theano, and CNTK and may operate on top of any of them. It was designed with the purpose of facilitating rapid experimentation. It enables the creation of prototypes that are simple and quick. It can perform identically on both the CPU and the GPU. Convolutional networks, which are used for computer vision, and recurrent networks, which are used for sequence and time-series data, are both supported, as well as the combination of the two. It is compatible with any network design you can think of, including models with multiple inputs and outputs, shared layers and models, and so on. This indicates that Keras is suitable for use in the construction of deep learning models, including anything from generative adversarial networks to a neural Turing machine. Approximately 200,000 people use Keras, ranging from academic researchers and engineers working at both startups and major firms to graduate students and hobbyists. Keras is also utilized by companies that are just starting out. Keras is used by numerous companies in addition to Google, Netflix, Uber, Microsoft, and Square with the purpose of addressing a broad range of machine learning issues.

5.11.2 Tensorflow

TensorFlow is an open-source framework that may be used from beginning to finish to develop applications for machine learning. It is a symbolic math library that focuses on training and inference of deep neural networks and employs dataflow and differentiable programming to carry out the many tasks that need to be completed. It gives programmers the ability to design machine learning applications using a wide variety of tools, frameworks, and community resources. At this time, the most well-known deep learning library in the whole globe is TensorFlow, which was developed by Google. Machine learning is used throughout all of Google's products, including the search engine, translation, picture captioning, and recommendation systems, among other things. TensorFlow is the finest library there is since it is designed to be user-friendly for people of all skill levels. The Tensorflow library combines a variety of APIs, allowing for the construction of deep learning architectures such as CNN and RNN at scale. The calculation of TensorFlow is built on graphs, and the software enables programmers to see the building of neural networks using a tool called Tensorboard. The software can be debugged more easily with the assistance of this tool. Finally, Tensorflow has been optimized for use in large-scale deployments. It utilizes both the CPU and the GPU. When compared to the other deep learning frameworks, Tensorflow enjoys the most popularity on GitHub.

5.11.3 Theano

Python's Theano module gives us the ability to analyze mathematical operations, particularly those using multi-dimensional arrays, in a very fast manner. Its primary use is in the construction of deep learning projects. It runs much more quickly on the central processing unit (CPU) than it does on the graphics processing unit (GPU). Theano is capable of achieving high speeds, which puts it in direct rivalry with C implementations for solving tasks that involve a significant quantity of data. It is able to make use of graphics processing units (GPUs), which allows it to perform much better than C on a central processing unit (CPU) in certain specific situations. It is able to take structures and turn them into code that is quite efficient and makes use of numpy as well as certain native libraries. Its primary purpose is to manage the kinds of processing necessary for huge neural network algorithms used in Deep Learning, and its architecture reflects this focus. Because of this, it has become an extremely well-known library in the area of Deep Learning.

5.11.4 Anaconda Navigator

Anaconda is a desktop graphical user interface which allows developers to run programs and manage packages, environments, and channels without using command-line commands. It is compatible with Windows, MacOS, and Linux. The open-source software package contains Jupyter, Spyder, and other tools for scientific computing, data analytics, and large-scale data processing.

Anaconda supports both R and Python. In scientific computing fields including data science, machine learning applications, large-scale data processing, and predictive analytics, package management and deployment are made simpler with the help of the Python and R programming language distribution. The Anaconda distribution includes more than 250 packages by default, and more than 7,500 more open-source packages, the conda package manager, and the virtual environment manager may all be obtained from PyPI.

5.11.5 Jupyter Notebook

To create and share documents with live code, equations, visualizations, and text, we can use the free and open-source Jupyter Notebook online tool. The staff of Project Jupyter oversees maintaining the Jupyter Notebook. The IPython project, which once had an IPython Notebook project of its own, gave rise to Jupyter Notebooks. The primary programming languages it supports are Julia, Python, and R, thus the name Jupyter. Jupyter comes with the IPython kernel which enables Python programming, however there are presently more than 100 additional kernels available.

5.11.6 Spyder

There are many different integrated development environments (IDEs) available on the market, and some examples include Spyder, Atom, Pycharm, and Pydev. Spyder is the most popular integrated development environment (IDE) among data scientists, and the primary reason for this preference is because the software was developed with the express purpose of being used in data science. The user may browse through a variety of data variables using the interface, and there is also an option to get ready-made online assistance. On the same screen as the code itself, the output of the program may be seen in the Python console. You are able to work on many scripts at the same time, and when you are ready to test them, you may do so in either the same console or a new one, and in either case, the variables that were used will be saved in the variable explorer tab. In addition to this, there is the possibility to see graphs and other visualizations in the plot window.

5.12 Version Control

Version Control is one of the most important discoveries in the world of programming. It helps developers trace back updates to their projects. It also helps in splitting project tasks between individuals in a team. Each member can create a branch off of the main, make changes to the code, and merge it, while version control makes sure that the code does not create any conflicts with any other team member's change. This would be very helpful for our team to keep track of each person's contribution.

5.12.1 GitHub

GitHub offers cloud-based service for developers to store and manage their code. It also helps keep track of changes to their code. It offers version control functionality. It keeps track of changes using branching and merging concepts. It stores a duplicate version of the source code inside a repository. It helps in making changes to the code without breaking the whole code apart.

5.12.2 Git

Git is by far the most popular modern version control system in use today. Linus Torvalds, the famous creator of the Linux operating system kernel, founded Git in 2005 as a mature, actively maintained open source project. Git is used for version control in a staggering number of software projects, both commercial and open source. Git is a distributed version control system that functions efficiently on a broad variety of computer operating systems and integrated development environments (IDEs). Additionally, there is a sizable population of experienced software developers who have worked with Git.

5.13 3D Modeling Software

For the pet feeder's design, we are going to use a 3D modeling software because it gives designs a level of depth that isn't possible with rough sketches or 2D designs, such as better control over details. It will also test how a design works in the real world without having to give up because of physical limits.

5.13.1 SOLIDWORKS

SolidWorks is one of the products that are offered by SolidWorks Corporation, which is driven by the 3D experience platform developed by Dassault Systems. The mechanical design automation program known as SolidWorks is a feature-based, parametric solid modeling design tool that gives designers the ability to rapidly sketch out concepts, experiment with features and dimensions, and create models as well as detailed drawings. It makes use of the straightforward Windows graphical user interface, making it possible for designers to drag and drop files, copy and paste text in a manner that is identical to that of Windows. A good number of the icons, including print, open, cut, and save, are also a part of the SolidWorks program, which is acquainted with the capabilities of Windows. When you are creating a model using SolidWorks, you have the ability to envision it in three dimensions, just as the model will appear after it has been made. SolidWorks is organized into three primary categories, which are the part mode, the assembly mode, and the drawing mode. This software's fundamental structural component is called "part mode." As an example, you are required to first construct a component before you can make an assembly. The term "sub assembly" refers to the pieces or other assemblies that are contained inside assembly mode.

6.0 Design

In the following sections we will be discussing the design of the different components of the "Pawsitive" Pet Feeder. Each part has been decided based on the best fit for our project.

This section gives an explanation on why the team chose the part and how each system will work to make the feeder work.

The outline of the section includes:

- Camera system design
- Microprocessor design
- Pet Collar System Design
- LED Design
- Food Dispenser Design
- Bowl and Lid System Design
- Photodiode Design
- Microcontroller Design
- Application Design

6.1 Camera System Design

After the completion of the research done for the camera, it was found that the OV5647 Mini camera module is the best fit for our design. It is cost-effective as it comes in packs of two for only $8.99. It has a CSI port, so it can integrate into the Raspberry Pi at the port that was designed for cameras. It is tied as the highest resolution from our comparison. It is compact and will be able to be easily integrated into our design. For these reasons OV5647 Mini camera module will be integrated into our feeder.

6.3 Pet Collar System Design

The pet collar tag will be an independent device from the feeder, but fundamental to accomplishing the goal of feeding the pet.

The tag itself will be a 3D printed plastic shell that will be small enough to not bother the pets. The plastic shell will also protect the LEDs and the onboard circuit from liquids and dirt. Another feature that will be included with the collar tag is an easy way to change the battery so the device can continue to function. This device consists of a plastic housing, a red, green, or blue LED and a Fresnel lens.

6.4 LED Design

For the pet collar tag, we have decided to use red, green, and blue. These LEDs will be inside the collar tag and used to help identify the pet in tandem with the camera system. The LED will be powered with a CR1632 button cell battery.

Each LED has a different forward voltage and the voltage going to the LEDs must be adjusted for the LED being considered, this can be accomplished by adding resistors in series to the LED to vary the voltage for each of the chosen LEDs. The LEDs chosen for this project all have a minimum viewing angle of 30 degrees. The housing for the LED a must also has an air gap between 15 and 20 mm for the light being emitted from the LED to disperse properly

The lens also adds a small amount of weather resistance to the electrical components including the LED.

6.5 Food Dispenser Design

The food dispenser design will take advantage of gravity to transport dry food from the storage container to the bowl. A dispensing screw will be used to either hold the food in place or to allow the transfer of food from the container to the bowl. When the camera system detects an animal, the system assigned for the pet will activate. This will rotate the screw, allowing food from the food storage to drop into the pipe with the screw, to be transferred to the chute, which will slide the food towards the bowl. For this design to dispense the desired amount of dry food, testing must be done to determine how much food is dispensed per rotation and the number of rotations needed to achieve the desired amount of food, and the RPM required. As this design requires 3 separate food dispensing bays, the software will only activate the bay that is assigned to its pet.

The food dispenser also must not get caught by the food during its rotation as well. Therefore, when the allotted food is dispensed, the dispensing motor will rotate backwards. This way, food will not cluster between the screw and the pipe, and whole bits of the pets' food will remain intact. The motor for the dispensing screw must be strong enough to withstand some resistance when the feeding bay is activated. This is due to the food setting into the pipe between the food storage funnel and the dispensing screw.

Part of the "Pawsitive" Pet Feeder includes the ability to allow additional food to be dispensed into the bowl through the application. If this is the case, additional rotations of the dispensing screw are needed. Therefore the motor system that is attached to the dispensing screw is critical to the food dispenser's ability to disperse food into the chute that goes down towards the bowl.

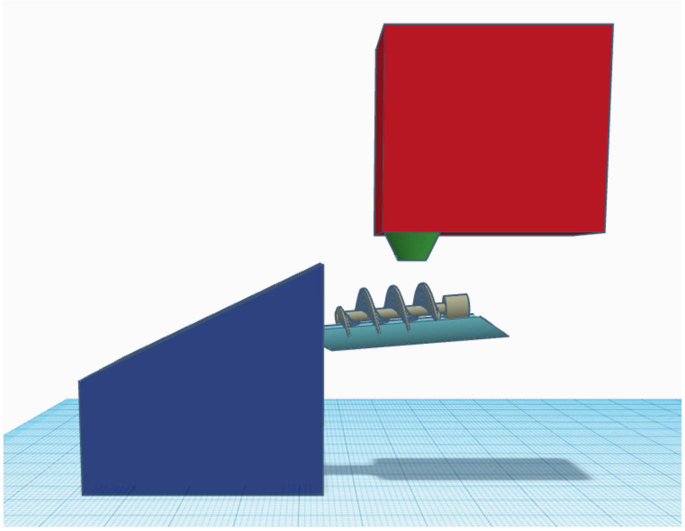


Figure 21: Visualization of Food Dispenser Design

	Color	Component
1	Blue / Cyan	Bowl & Food Dispenser Pipe
2	Grey	Dispensing Screw
3	Green	Food Storage Funnel
4	Red	Food Storage

Table 28: Figure 19's Color Key for Visualization of Food Dispenser Design

6.6 Bowl and Lid System Design

The bowl and lid system will play an integral role in the overall design of the pet feeder. This system consists of a semicircular plastic lid that sits over the top of the pet's food bowl and the food dispenser, the lid motor, the pet detection system IR LEDs, and the bowl. All these components are discussed in the section below. Photodiode design is in the next section.

6.6.1 IR LED Design

As a part of the pet detection system for the pet feeder the Infrared LED will be located above the bowl and the lid to the bowl. These will emit light while the system detects a specific-colored pet tag and will continue to stay on until the feeding time is over. These LEDs will function alongside the Photodiodes to detect the pet's head and to tell the system when to open and close the lid. The LEDs that were chosen will be invisible to the pet and the user but are close enough to the visible spectrum that they can be seen with a conventional cell phone camera, which will add ease to the customer giving them a sense of comfort that the device is functioning properly. The LEDs will be placed far enough away from the food that they will never intentionally encounter excessive foreign debris such as pet food crumbs or shed fur but will be close enough that there will be no problem detecting the pet's head no matter the breed or size. The LEDs will also be protected by a thin plastic screen to further reduce any chance of being blocked out by foreign material.

6.6.2 Lid Design

The lid for the pet feeder must reliably open and close. To achieve this, a high-quality motor was chosen so that this product can perform its task correctly every time. The motor will be place in an inconspicuous location on the device that will be out of sight of the pet and the end user. The motor will be placed above the dispenser spout and will use a metal worm gear to move the lid up and down and will do this in an inconspicuous manner. Through research, the chosen motor was found to work with both the raspberry pi that the pet feeder team is using along with the Arduino used that integrates the motor system with the logic board (microcontroller).

6.7 Photodiode Design

The photodiodes will be placed above the pet bowl and the lid system, so they absorb the maximum amount of light from the IR LED. The photodiodes will be placed in a horizontal line opposite the LEDs, this way as much of the light will make it through the system while the pet is not present and will receive the lowest input of light while the pet is present to ensure a clear signal. The current that is generated from the photodiode will be sent to the microcontroller that is located inside of the housing of the feeder. A predetermined on average and off average will determine whether the lid would remain open or should be shut.

6.2 Microprocessor Design

During the research portion of the potential microprocessors to use for our project, we chose to go with the Raspberry Pi 4. With the Raspberry Pi 4 microprocessor being a microcomputer, it will replace a traditional desktop within the system as the microprocessor takes in all inputs related to a traditional desktop such as a power supply, HDMI, keyboard, and mouse, with proper WIFI, and ethernet control making the possibilities of the Raspberry Pi 4 practically endless. The Raspberry Pi 4 will act as the brains of our device operation allowing the system to dissect logic with the implemented Python code, especially when we plug in a compatible camera input into the Raspberry Pi 4 and allow for the camera to detect the LED light attached to the collar of the pet to further determine whether or not the color of the LED light is correct (true), and ultimately open the lid properly for the animal to eat the food. Although The Raspberry Pi can be bought for $124.94 on Amazon. We will be able to borrow it from the UCF library during the duration of our project.

6.8 Microcontroller Design

The microcontroller that was selected to be used within our project after analyzing the information from the research section was the Arduino uno. Since our team will make use of a microprocessor, the Arduino architecture will be able to interface more simply with a microprocessor in order to transmit crucial image data. Additionally, Python can be used to program code for the Arduino as well as the Raspberry Pi 4, which makes synergy between the two devices much more seamless. The Arduino also comes equipped with a 9 V to 3.3 V voltage regulator, which will be able to be utilized when connecting to the Raspberry Pi 4 (Raspberry Pi 4 works off of a 3.3 V basis). Additionally, the Arduino can be powered by a 9 V adapter which can also allow the Arduino to power the Raspberry Pi 4 as well through a USB connection. The main use of this microcontroller in the pet feeder will be to activate motors in response to image data that will be processed by the Raspberry Pi 4. The Arduino will be programmed to activate the motors that open/close the lid and dispense the food whenever the Raspberry Pi 4 sends the data corresponding to the collar LED being recognized by the camera module. The images below show the two different ways of interfacing the Arduino with the Raspberry PI 4 that may be used in our design.

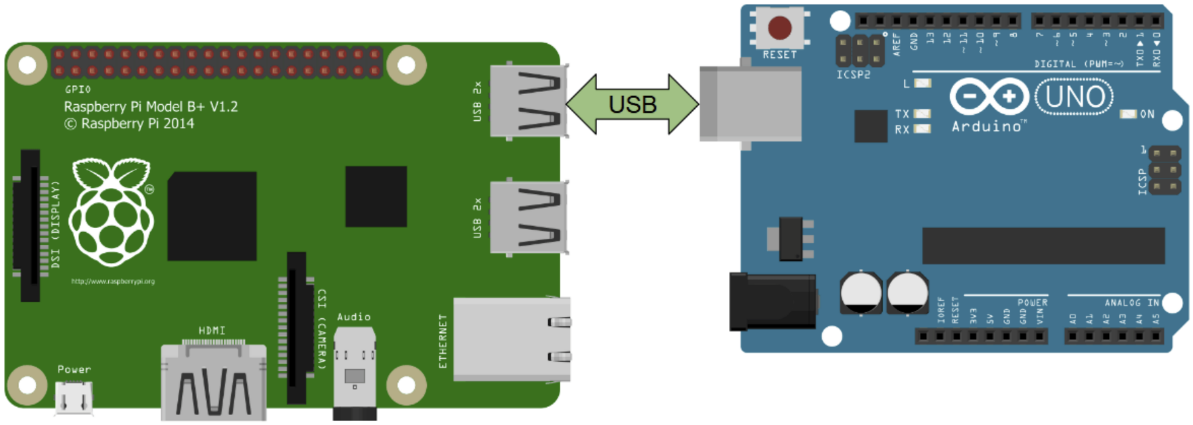


Figure 22: Arduino Uno and Raspberry Pi 4 Connection through USB (Courtesy of Robotics Back-End)

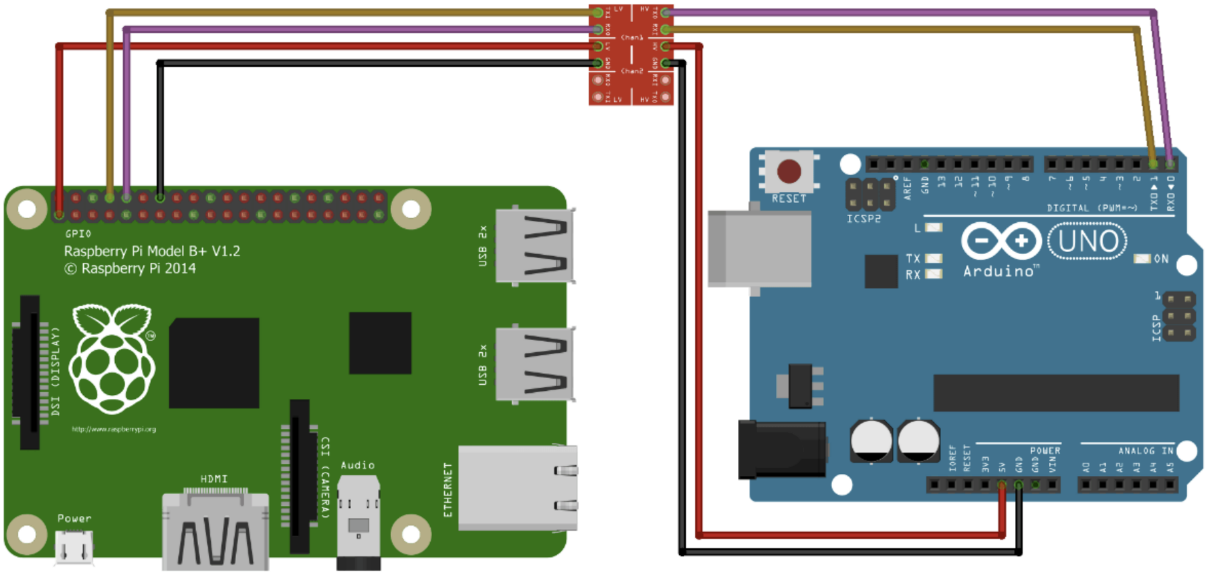


Figure 23: Arduino Uno and Raspberry Pi 4 Connection through GPIO Pins (Courtesy of Robotics Back-End)

6.9 PCB Design

There will be a PCB developed for the "Pawsitive" Pet Feeder that will be embedded in a collar that the pet will wear as a part of the Smart Pet Feeder with Optical Recognition. The PCB created will have powered batteries LED along with a switch that enables the user to turn on and off the collar tag. In order for the PCB to fit within the collar housing that needs to be constructed, we will be using KiCAD application to develop the schematic board layout with all the necessary components.

In order to create the board layout, a simulated version of the circuit schematic which will be used on the PCB will be formed and tested first, before the board layout is created. The section below will indicate detailed steps in completing the simulation and development for the PCB of the pet collar PCB.

6.9.1 Pet Collar Circuit Simulation

The colored LEDs that have been chosen in section 5.2.1 respectively can be used to simulate and design a circuit that will be used to power these LEDs within the pet collar system. Based on the design of this circuit, when the switch is set in the closed position, the power from a battery will be used to drive an LED on whenever the switch is in the closed position (the battery is physically connected to the circuit).

There are three separate figures provided below that have been created to accommodate the three different colors of LEDs that will be implemented into the collar of the PCB design (red, blue, and green). We will be using Multisim Live simulation software in order to simulate and test these circuits before being able to accurately transfer to the PCB.

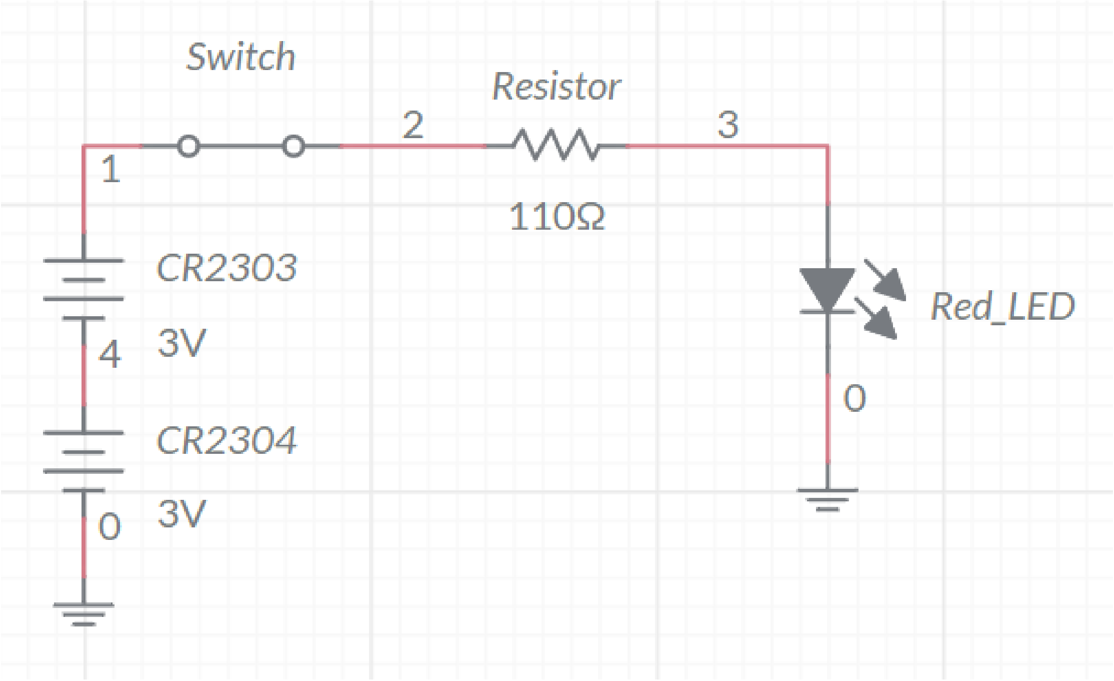


Figure 24: Circuit Schematic for Red LED

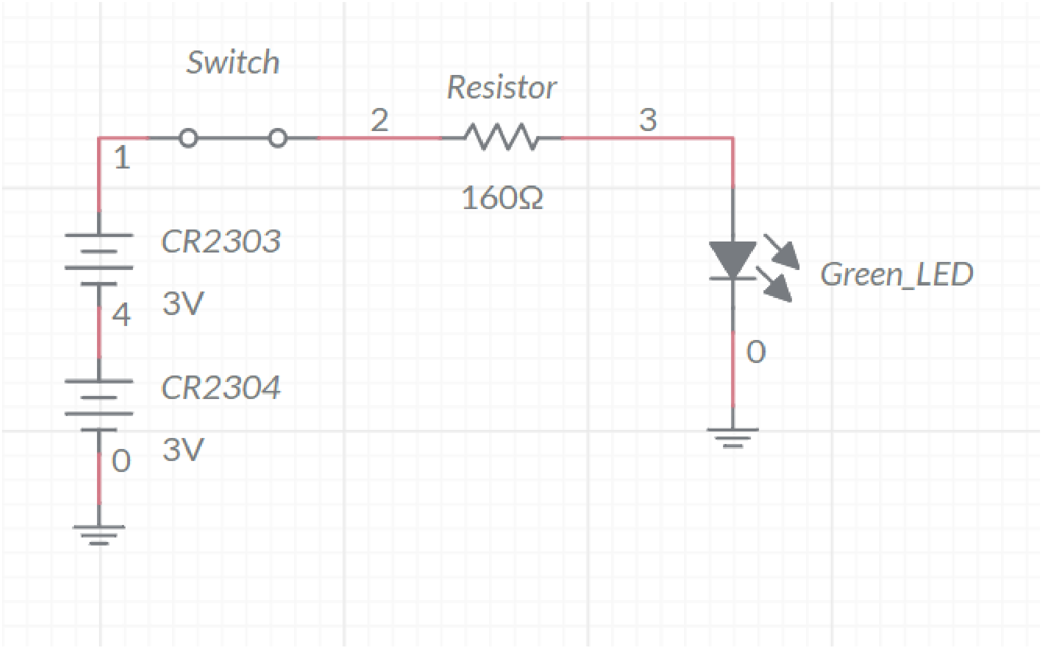


Figure 25: Circuit Schematic for Green LED

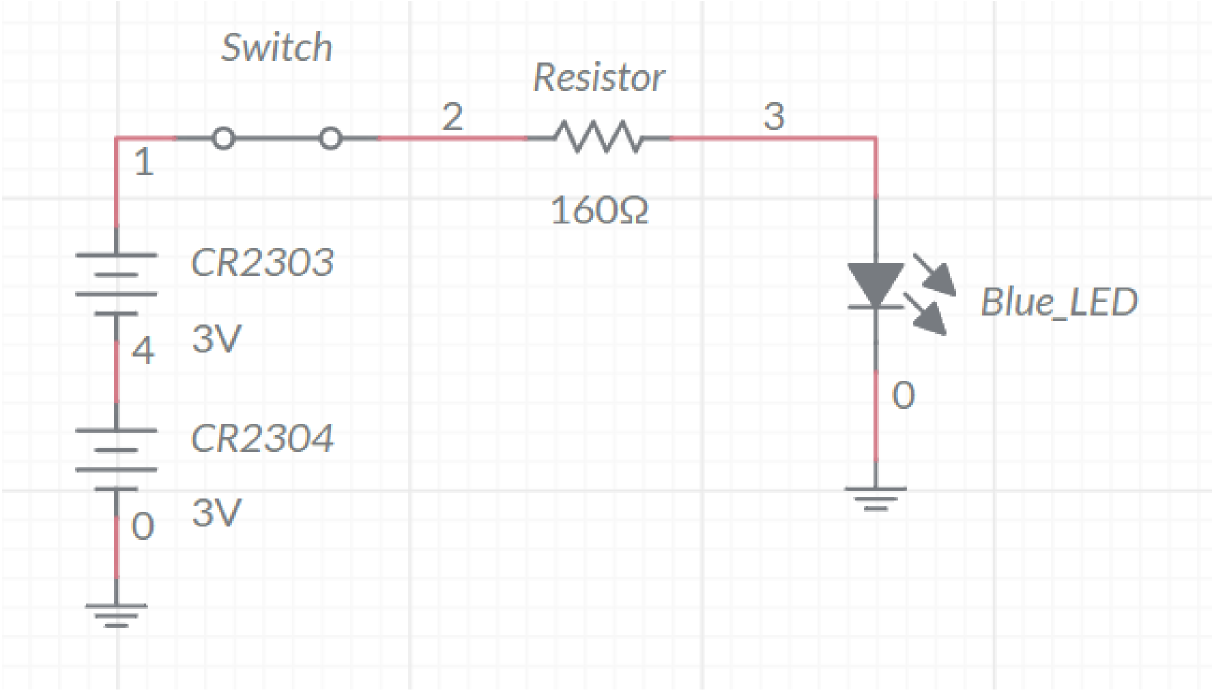


Figure 25: Circuit Schematic for Blue LED

Circuit	Red LED	Green LED	Blue LED
Voltage across limiting resistor (V)	3.2	3.2	3.2
Operating Current (mA)	20	20	30
Calculated limiting resistance (Ω)	110	160	160

Table 29: Limiting Resistor Calculations

The red, blue and green LEDs that are being simulated in the circuit above correspond closely to the LED's that have been selected within section 5.1.2. The SPLICE models that are being used within the LEDs will not be to specifically match the specifications to each LED that is being selected, however it gives a close proximity to how the LED will be with the rest of the component in the circuit.

Our teammates have agreed to add a second battery (CR2302) in series with the initial battery as demonstrated within the figures to provide a greater voltage to the circuit. Due to this increase of voltage, it will help limit resistors to be larger enough and prevent too much current from flowing through the circuit. This addition of a second battery also helps with extra capacity, this will result in a longer lifespan of the collar tag.

In order to determine the value of the limiting resistance, we will be using Ohm's Law (V = I R). The values of the operating current, denoted as I, and the voltage across the limiting resistor, denoted as V is obtained shown in the table above. By using these two parameters, we can solve for R and use the result in the minimum resistor value needed to maintain the LED in an ON state with the proper operating current.

Since this simulation is being done in Multisim Live, the components that are being used do not contain any tolerance or internal resistance that would be encountered if the simulation was being constructed within a physical circuit. Therefore, the values that are obtained through this Multisim Live simulation are only approximations, minor adjustments may be needed in the future when putting the PCB together during the design staging process. The table above highlights the +++s for the values of the limiting resistor to allow for the proper voltage drop across the LED.

Now that the minimum limited resistors have been obtained for all the three LED circuits, the circuit can be simulated. The value of voltage and current can be verified to solidify the final circuit design that will be implemented within the PCB.

By utilizing the tools that are provided within Multisim Live, the voltmeter/ammeter tool, the voltage as well as current data can be obtained from the input and at the ode in between the resistor and the LED. The three figures below demonstrated the simulated circuit with their appropriate voltages and current that matched the profile on the table shown above. The color LED indicates that the LED is on and working properly.

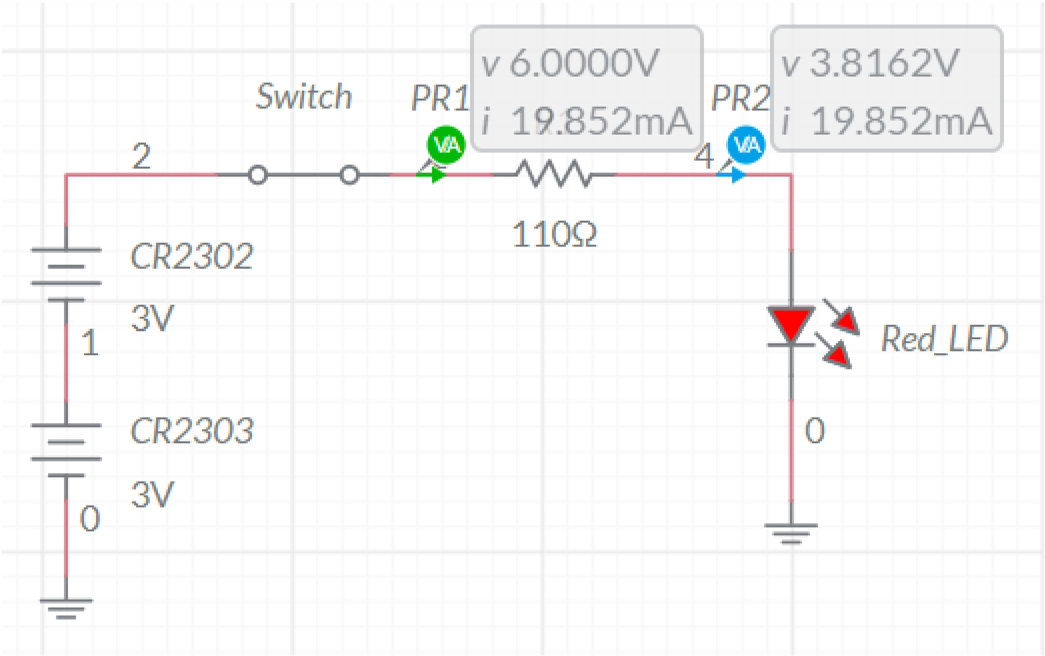


Figure 27: Circuit Simulation for Red LED

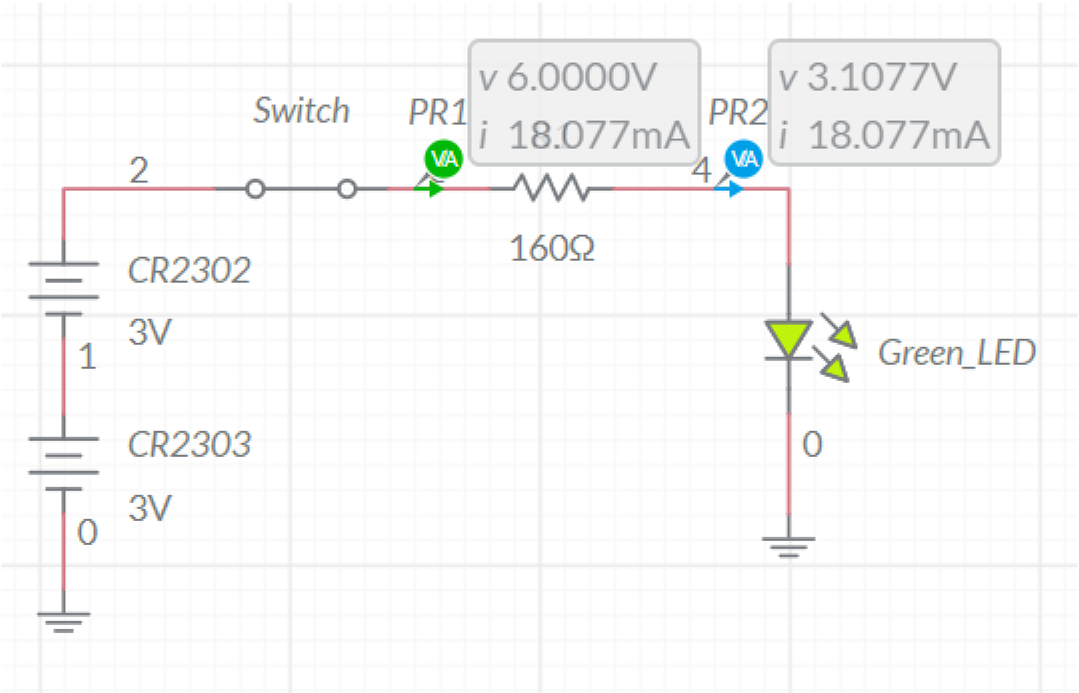


Figure 28: Circuit Simulation for Green LED

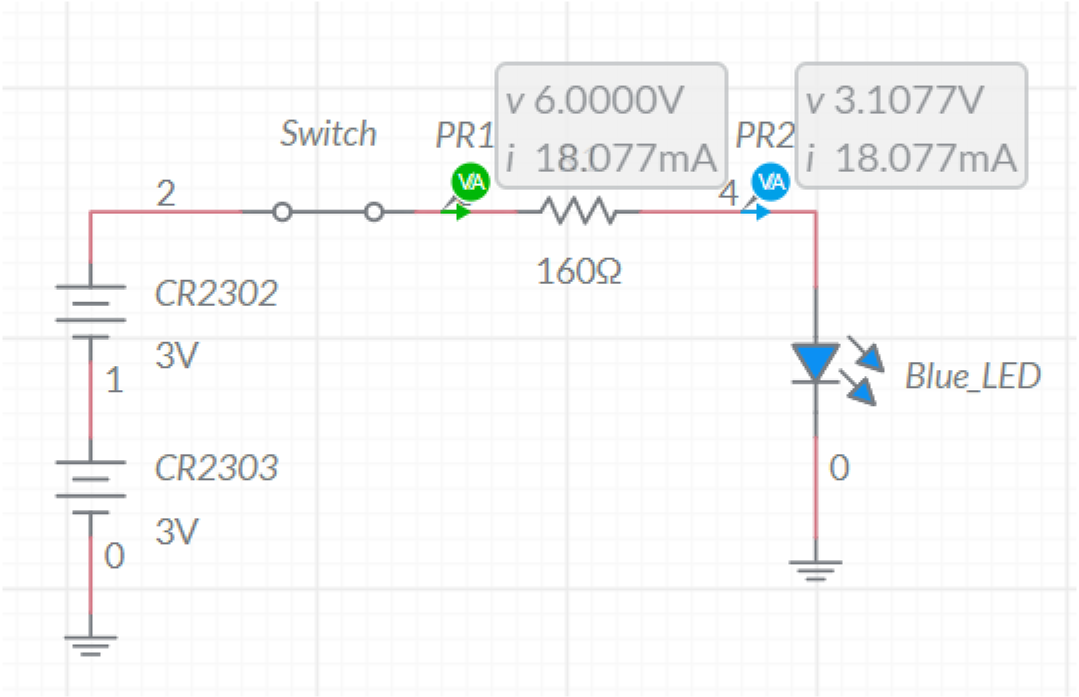


Figure 29: Circuit Simulation for Blue LED

From the observation that can be obtained prior to the circuit diagram, we can construct the following table of information values filled with the voltage and current values to the expected (theoretical) voltage and current values to ensure a proper simulation was conducted.

Circuit	Red LED	Green LED	Blue LED
Simulated Forward Voltage (V)	3.8	3.1	3.1
Theoretical Forward Voltage (V)	3.2	3.2	3.2
Operating Current (mA)	19.8	18	18
Theoretical Operating Current	20	20	20

Table 30: Simulated Circuit Values

The table shown above indicated the accuracy of the voltage across and current through the LED components within the simulated circuit. This shows the comparison of the values found within the respective LED datasheet that has been collected.

From the table shown, we can observe that the simulated values are close to the expected values; these slight errors in margin can be affected by the simulation, as it does not consider all non-ideal parameters that are affected if the circuit was built within a physical environment.

Moreover, we ensure that all the simulated operated current is less than the theoretical operated current. This is to ensure that the LED would receive enough current to ensure that the illumination brightness is enough, but not receive an amount of current that would drain more power from the battery without providing any visible increase in brightness.

6.9.2 KiCAD PCB Developer

The next step is to design the PCB, since the circuit simulation has already been verified and completed. The PCB board layout is designed by corresponding the physical components' footprints and measurements. The figure below is an example of a PCB schematic of the Red LED circuit, the other two PCBs (blue and green) will look similar however the value of resistor will be different respectively.

The PCB layout will not be affected based on which color of LED is being chosen, this is due to the collar being the same in shape and size. The schematic of the PCB collar is shown in the figure below. KiCAD is utilized to develop the PCB schematics.

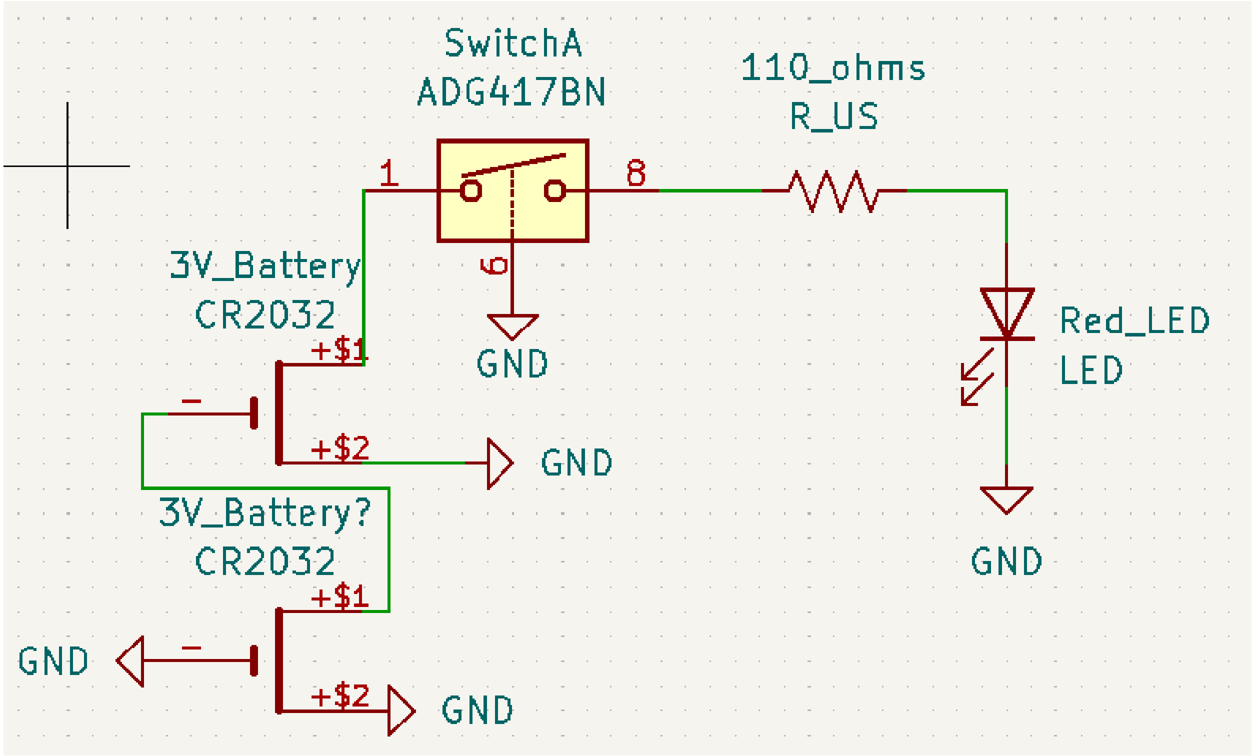


Figure 30: Collar PCB Schematic (Red)

The figure above shows the schematic that is being used for the PCB in the pet collar. In the figure, it consists of two batteries that are connected in series, this is used to provide enough voltage to the LED to ensure that it will light up under specified operating conditions.

The switch that is connected to the batteries is used to either connect or disconnect the batteries from the rest of the circuit, this is responsible for the LED light to turn on and off. The addition of the switch is to extend the lifespan of the collar tag. This allows the device to conserve more energy. The limiting resistor (the resistor value of 110Ω was calculated in the previous section) is connected in series to both the batteries and LED. The following table given below shows symbol, part number and types of components that are used within the schematic figure shown above.

Symbol	Value	Part Number	Type
SwitchA	-	ADG417BN	SPDT Switch
R_US	110Ω	-	0805 Resistor
Red_LED	-	RL5-8030	LED
CR2032	-	CR2032	Coin Cell Battery

Table 31: Component Selection for Collar PCB

6.9.3 Additional PCB Insight

Furthermore, there may be additional modules that may have to be developed using a PCB in addition to the PCB that has been developed for the pet collar of the ‘"Pawsitive" Pet Feeder’ so that they can be included in the entire design of the rest of the physical system for the pet feeder.

Since the testing has not been conducted, it is still uncertain if these modules will need to be implemented into the design in order to get the project to function. This additional module that might be added is the 3.3V to 5V circuit that will be able to take an input voltage within a range of 2.3 V and 3.3 V, this addition provides consistency in output voltage, that being 5 V.

There may be a need for a circuit of this type if we are planning to use batteries to power a section of the device that is separate from the battery powered PCBs on the collars that are powered by batteries. The figure below is a circuit design of a 3.3 V to 5 V converter that utilized the TI component referred to as TP5613221ADBYR.

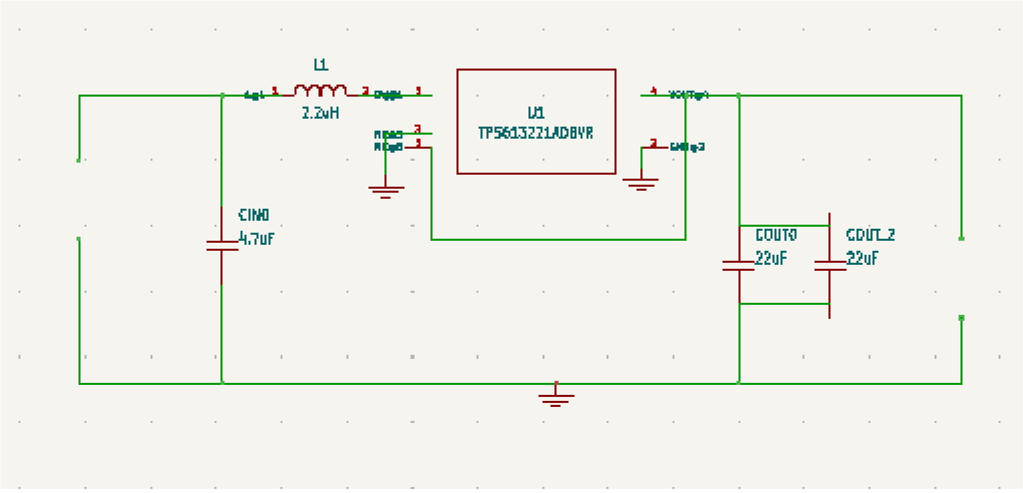


Figure 31: 3.3V to 5V Converter Schematic

6.10 Overall Design Schematics

Following the selection of a pet collar PCB design, along with the microcontroller and microprocessor, the overall schematics of the project can be formulated to indicate which ports and connections need to be made in order to fulfill the requirement and utilize the "Pawsitive" Pet Feeder.

Within the project, we will be making use of both the Arduino Uno along with the Raspberry Pi 4 in order to control our completed device. Both the schematic of the device will be respectively shown in the following figures below. The schematics for our project will be made mostly in software KiCAD (except for the Raspberry Pi 4, since it will be mostly its own device alone).

The parts are all added to the software KiCAD will need a proper footprint and symbol along with all the required footprint of the Arduino Uno microcontroller. The figure below shows the KiCAD schematic design for the overall design schematic that will be used within the project ‘"Pawsitive" Pet Feeder’. This schematic design shows all the pins and connections output for the PCB of Arduino UNO along with a separated schematic on the top left corner of the schematic for the pet collar PCB section (labeled in the figure).

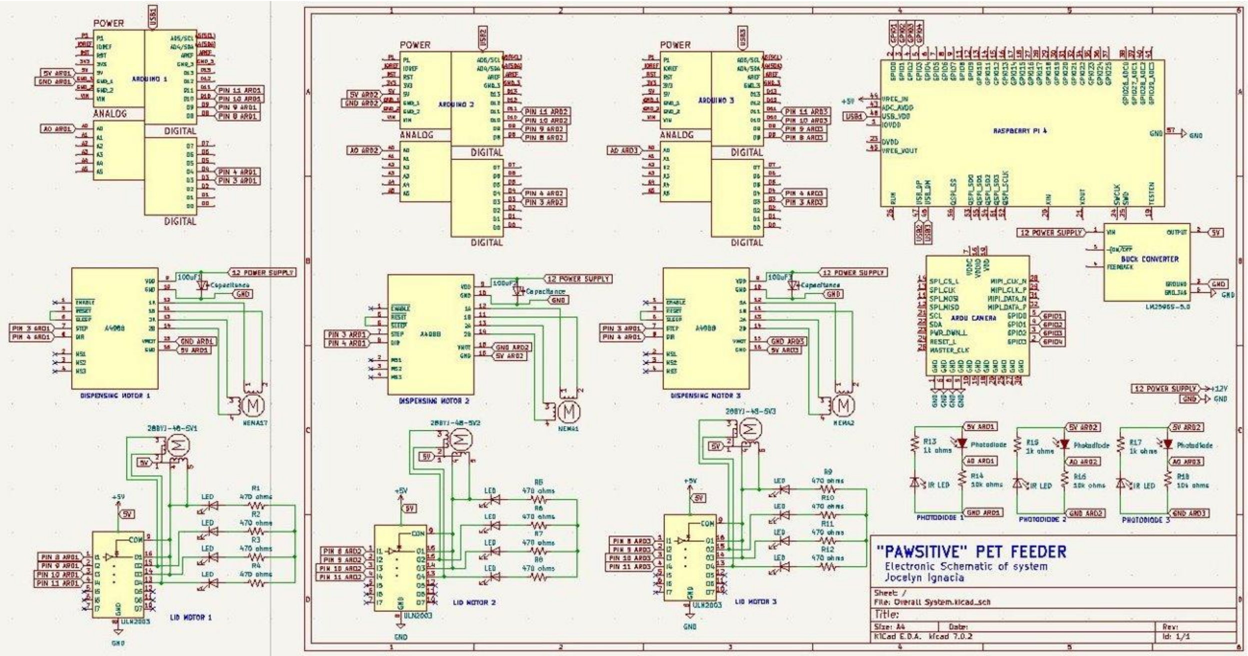
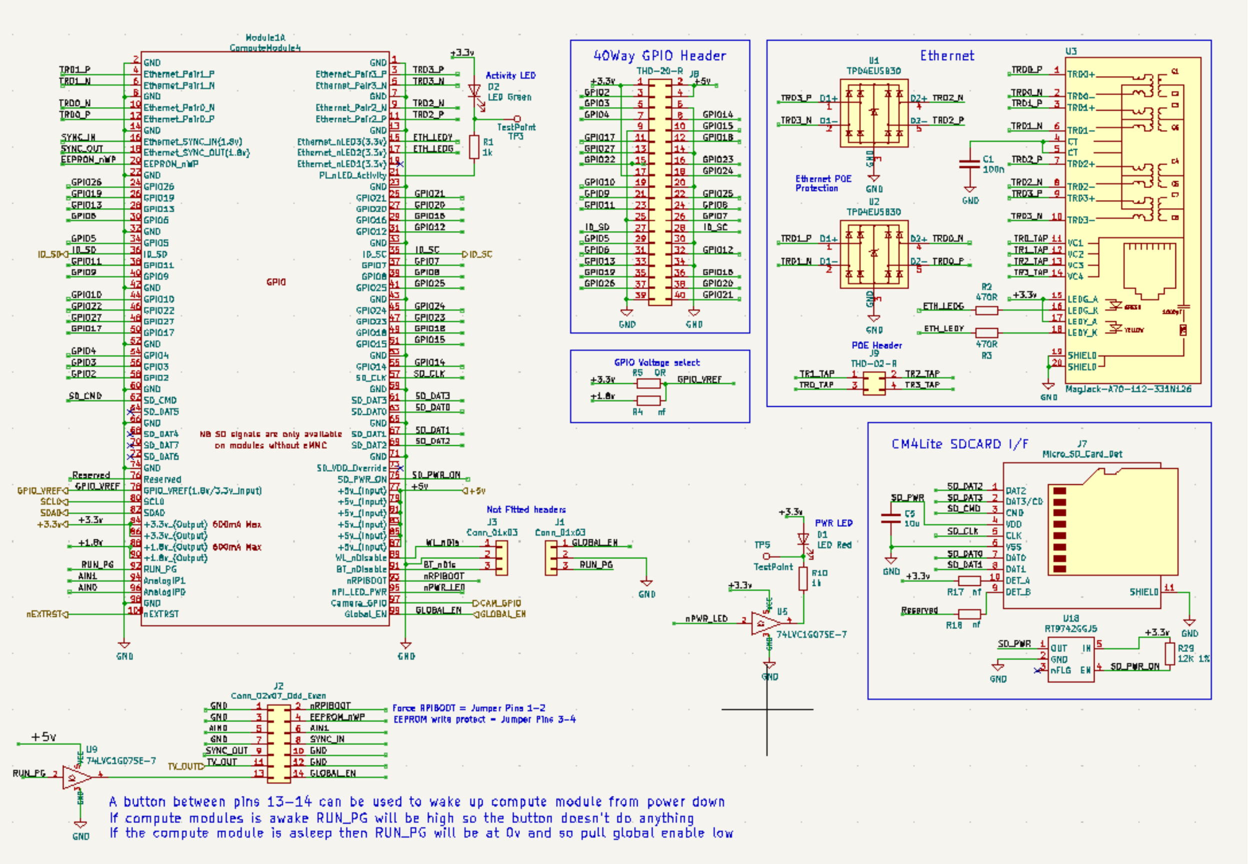


Figure 32 : "Pawsitive" Pet Feeder Overall Microcontroller/Pet Collar Schematic

Furthermore, an addition of the schematic design for the Raspberry Pi 4 is shown in the figure below. Since there is no alteration to the original PCB of a Raspberry Pi 4 that we will use, the schematic documentation can already be found within the software KiCAD and therefore the schematic file is not required.

Alternatively, the Raspberry Pi 4 contains its own schematic documentation that will be followed when we start developing our project. All the external ports and pins have been highlighted within the schematic overview, the two main parts of the schematic relevant to the "Pawsitive" Pet Feeder, the USB connection module and the Raspberry GPIO Expansion. Both the schematics are shown below in the respective figures.



A button between pins 13-14 can be used to wake up compute module from power down
If compute modules is awake RUN_PG will be high so the button doesn't do anything
If the compute module is asleep then RUN_PG will be at 0v and so pull global enable low

Figure 33 : Raspberry 4 Pi GPIO Expansion

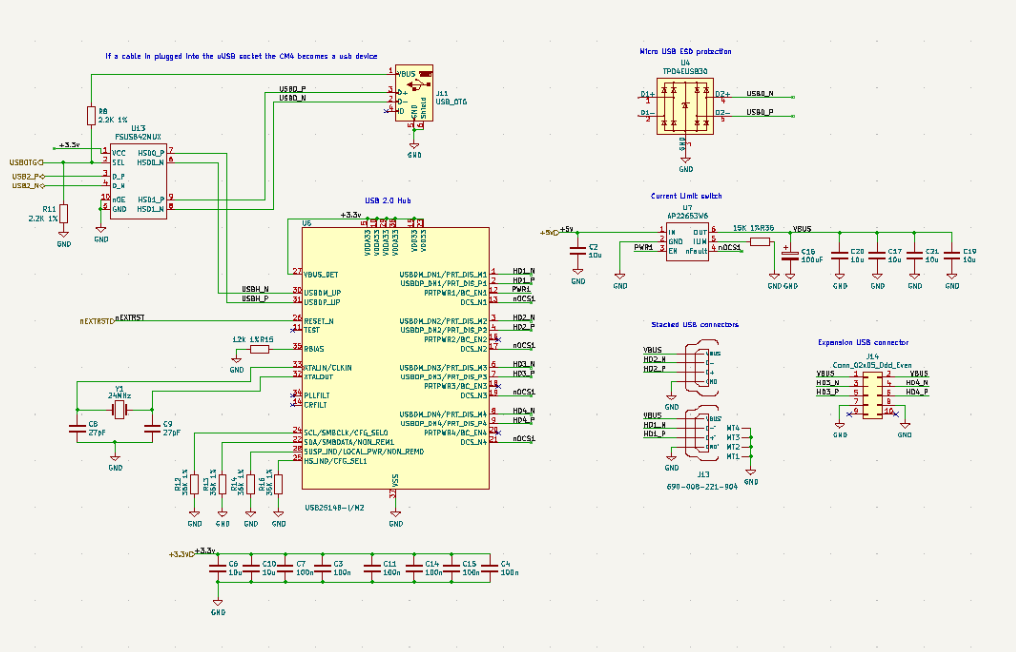


Figure 34: Raspberry Pi 4 USB Connection Module

6.11 Mobile Application Design

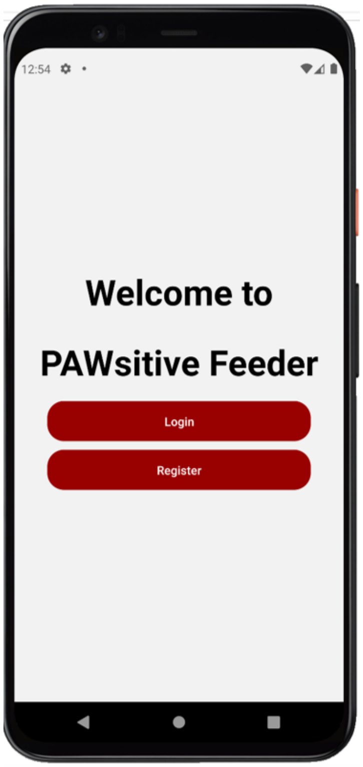
The team will use this section to highlight the functionality and design of the feeder's software components. We begin this section with a series of subsections that dive deep into the plan for the user interface to be simple and functional, then we discuss login/signup, which is then followed by a deep dive into the design and goals for the mobile application's landing page.

6.11.1 User Interface

The application must be designed to be simple and functional. The user interface must be approachable so that users are not frustrated by the various mechanisms provided by the application. To achieve this, the UI elements must be consistent and common. A bell symbol, for example, is synonymous with notifications, and using that symbol consistently in the pet feeder application will ensure that our notifications function is consistent with how users understand mobile applications. This design philosophy will be applied to all UI components. The page layout must be purposeful, with each section of the UI serving a specific purpose. The login page, home page, and live streaming page all serve a purpose in the application, and the transition between them must be seamless. For example, the streaming page should only be clearly accessible during a feeding or deterring event to avoid confusing the user when nothing is happening on the camera system. The color scheme should be used strategically to convey different levels of urgency to different events while also providing comfort to the user's needs. Colors such as red are used for important things like notifications and live streaming, so a consistent set of colors is important to the overall feel of the application.

6.11.2 Login/Signup

The application will start with a login/sign up page. Figure 33 shows a prototype of the welcome page that a given user will be met with upon opening the application. The user may either sign up for the app by creating a new account or log in with an existing account if they have used the app before.



Welcome to

PAWsitive Feeder

Figure 35: Application Log In & Sign Up Screen

If the user chooses to create a new account, they will select the button labeled "Register". Figure 34 shows the sign-up page of the application, which the user is redirected to by selecting the "Register" button. Here, the user is first given the option of choosing a profile picture for their account. After making this optional choice, the user must enter their name, email address, and a username, then choose the "Register" button on the sign-up page. Any user's information will then be held in a temporary table within a database, waiting for confirmation from an email verification system.

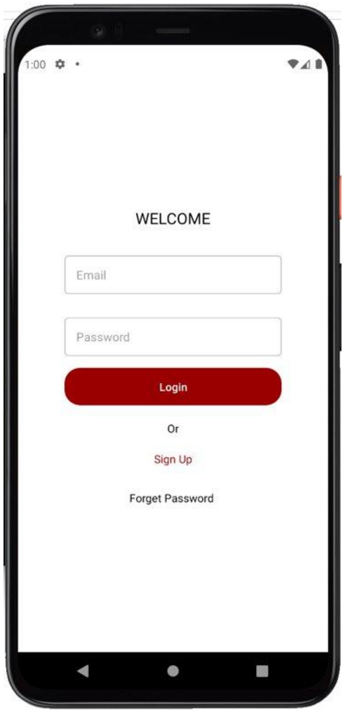


Figure 36: Application Log In Screen

The application will then display a message asking the user to confirm their email address in order to sign into their new account. After acknowledging this request, the user will be returned to the application's welcome screen, where they can sign into their new account (after confirming their own email address). When an existing user signs in, they can choose to have the app remember their login credentials on that device for future use. The user should then be able to locate and connect to their pet feeder device as long as both their smartphone and pet feeder are within range of the same WiFi network signal. To increase security, make sure the user goes through the verification process. This method also saves space by avoiding the creation of fake accounts. A user will not be able to sign into the application until they have confirmed their email address. A mongoose database will be used to store user information. The Mongodb Compass program allows us to visualize and modify the database structure. Using this compass, a temporary database will store the user's email address and username for a set period of time, unless the user confirms their account. When the confirmation process is finished, the user's data is moved from the temporary table to a permanent data storage table. The MERN stack, which will house the application's functionality, is made up of Node.js and React. The Node.js API will collect a user's information and store it in the appropriate database. The confirmation email will also be sent to the user's email address by the Node.js API. The login UI will be built with React and linked to the API code.

6.11.3 Landing Page

Figure 35 shows a prototype of the application landing page, which will be home to access most of the functionality of the app. The settings button will lead you to the settings page, which will contain choices regarding notifications. The notification button will open up a popup menu where the user will have the option to view or clear recent notifications. During the event of a notification, a bar will appear across the top of the application for a limited amount of time, showing off the most recent notification. For example, if a feeding event occurs, the user will see a box appear on the top detailing what pet is feeding.

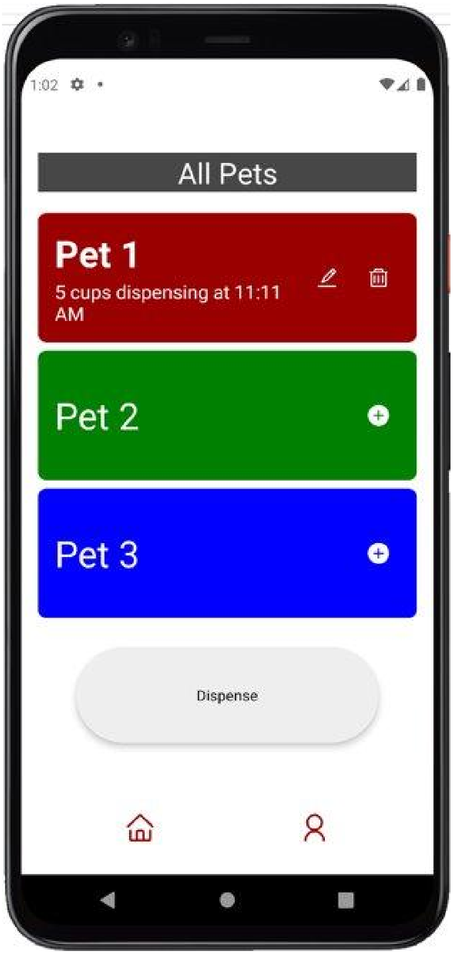


Figure 37: Landing Page

The center of the landing page will have three pet profiles/cards. When the user clicks on the title of this card, they will be taken to a dedicated pet page to edit/enter/delete the pets name, the time to dispense food and the amount of cups to be dispensed. Finally, the landing page's "Dispense" button will be used to send the data of all pets to the feeder and the feeder will dispense the food when it is time. This page's UI frontend design will be done in React Native, with subcategories such as notifications coded in Node.js. In order to detail that a feeding event is taking place and what species is currently feeding, the species detection software will interact with the Node.js code for notifications.

6.12 Web Application Design

For our project, we need a web application that can store data on a remote server and send it over the Internet through a browser. In building our web application, we will use a web application MERN stack structural design. These application UI components are a useful tool to improve the step-by-step application of the project. They also work well together to make a fully functional web application that can be hosted on a server and put on the internet for everyone to use. In a MERN stack (Mongoose Express React Node), the traditional front-end framework has capabilities like React.js and MongoDB, respectively. There are also FERN and MEAN stacks, which can be used in any case depending on the front-end JavaScript framework and document database that is preferred.

React.js, the declarative JavaScript library framework for making dynamic client-side user interface applications in HTML, is at the top of the MERN stack. React lets developers code their specific UI components in JavaScript to make an overall framework for a web application that looks good. Which basically means that React lets developers build complex interfaces with simple components, connect them to data on your backend server, and render them as HTML. The best thing about React is that it can handle stateful, data-driven interfaces with little code and little pain. It also has all the bells and whistles that developers expect from a modern web framework, such as great support for forms, error handling, events, lists, etc., which makes it easy for everyone to use. In terms of hosting the server side of a web application, the next level down is Express, which is inside of Node.js. Express has powerful models for URL routing (matching an incoming URL with a server function) and handling HTTP requests and responses efficiently. These functions, in turn, use the Node.js drivers for the document database of the developer's choice (in our case, Firebase) to access and change data in your document database.

Lastly, mongoDB can be very helpful if the app needs to store any kind of information. As mongoDB is a Google platform that gives you access to a lot of developer tools, most notably their real-time, NoSQL database, which you can change through their website, created and referenced in the developer's React.js front end that can be sent to the Express server, where they can be processed and stored directly into the Firebase database for later retrieval. The MERN Stack application will come full circle and come together to make a fully functional application that will be designed to be fully controlled remotely via the internet web page application as it connects and communicates with all the important key parts of the system's functionality. Some of the most important functions that can be controlled through the website include feeding the pet from afar, controlling how much food is given to the pet, etc.

6.13 System Software Design

After completing the research phase of potential system software programs to use for our project, we collectively decided to use Python as the primary coding language for our microprocessor because the Raspberry Pi 4 heavily supports the Python programming language for operating any type of custom functionality.

Python offers a much simpler syntax than older programming languages such as C, and C++, where in languages such as C, and C++, the formatting syntax can be very tricky and confusing due to the fact that most users must iteratively define every method and function used while maintaining the structure of the program. Debugging can be challenging in languages such as C and C++ because error messages are sometimes ambiguous, and segmentation faults occur when memory is not allocated. Python, on the other hand, makes debugging as simple as pressing a button, thanks to its clear error messages, single variable definitions, and automatic dynamic memory allocation, which prevents programs from running out of memory. Python also provides a plethora of libraries that can be referenced in the source code, enhancing the potential to accomplish any difficult logic/algorithms that may be even more challenging to program in object-oriented languages such as Java.

This Python code will typically be written, edited, compiled, debugged, and executed in Visual Studio Code for our system software development. Visual Studio Code is regarded as one of the best code editors available for Python due to its redefined and optimized building structure for editing, and a state-of-the-art debugging system especially shaped for the implementation of modern web, cloud, and mobile applications if the system is later required to communicate with such functions. Also, the fact that the program is free for all users is a benefit. The highlighting functionality that can be optionally added outside into the software is quite helpful with its colorful code organization hints and error hint messages that are offered before even compiling the code.

Other code editors such as Atom and Repl.it can get the job done, but Visual Studio Code's coding efficiency is unsurpassed. Atom makes it difficult to share code with others, and Repl.it makes it tough to compile numerous files, but Visual Studio Code can and will achieve the best of both worlds. Visual Studio Code also includes a compiler terminal, which may be accessed while editing in the same program. Visual Studio Code's ability to link directly to GitHub provides the ideal segue for presenting the correct implementation and use of the GitHub software in our group project. In the modern era of collaborative programming, the organizational standard and structural definition that GitHub provides, which enables the author to easily post online to the group's GitHub account and allow secure access to every member of the group who may need to push or pull the code to their project's code from online to their machine and vice versa, is extremely useful. As indicated earlier, not only is access to the collaborative work efficient, but it also gives structure and organization to the developed programming code, which is always more effective and professional than having all the code written jumbled up on a file kept on your local storage drive.

7.0 Testing Design

In order to validate the effectiveness of our design, we will compare the functionality of our working design against the objectives and specifications of the project discussed in section 2.0 and section 6.1, respectively. The optical components will be demonstrated in this section 7.1 in order to demonstrate the functional operation of the components as intended. In addition to this, we will discuss how the rest of our system will be tested to ensure that it functions correctly in the future.

During the testing of our system, we will not be using any real animals in order to ensure the safety of the system. The University of Central Florida does not allow non-service animals on campus, so we will be testing our design in an alternative way as a result. To demonstrate a working product, we will be using stuffed animals as props and using our hands to bring the collar that will be around the stuffed animals to move closer to the camera system.

7.1 Camera System

To test the camera system, we plan to first find the detectable object distance. We will keep the camera still and place the LED collar a far distance from the camera and start to slowly move towards it. We will start approximately 20 feet away from the camera and record the color the camera identifies as we approach. From this we will know at what distance it starts to correctly identify the LED. We will also investigate the minimum detectable object distance by bringing the collar tag as close as possible to the camera.

These procedures will be repeated 15 times to assure we find the correct detectable object distances. Once we have the data, we will average it to get the best result. The success of this device relies on the camera system identifying the right color, and we plan to present all three colors used from our measured detectable object distances to make sure the camera activates correctly.

7.2 Collar Tag System

It is important to have the collar tag system working at all times since the pet can approach the feeder at any time of the day. We need to make sure the LED constantly is receiving power within the collar housing. If any of the electronics inside the housing fails, the display may not turn, making the camera system fail to detect any color.

For the Midterm Fall 2022 CREOL Senior Design Demonstration, we built a circuit to simulate the inside of the collar tag. For our case, we powered it with the microcontroller. Since we were using an RGB LED (red, green, and blue), we used three 330 Ω resistors, but for a single-color LED, we will use only one resistor.

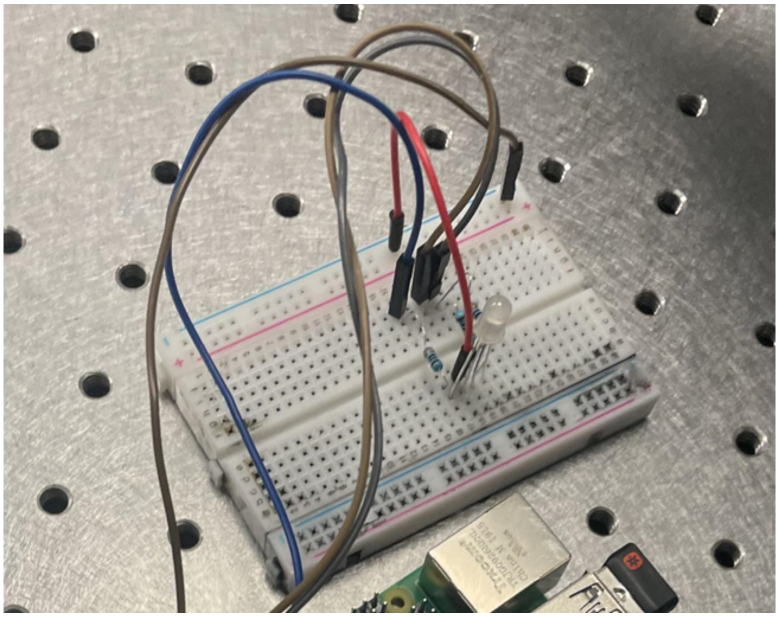


Figure 38: Pet Collar Circuit Sample

For the collar tag, the same circuit will be built, but it will be powered by a 9V battery that will be connected to a voltage regulator module that will convert 9V voltage to usable 5V voltage. The 5V will be sent across the resistor in series with the LED.

Although for the demonstration 330 Ω resistors we used, we plan to lower the resistance to 110 Ω resistors to make sure our LEDs turn one with this voltage. We expect to see all LEDs turn on every time we connect the circuit to the battery. We will turn on and off each LED 20 times, and to pass the test, we expect the LEDs to turn on every time.

7.3 Food Dispenser

It is important that the Food Dispenser is tested because it directly affects how much food the pet will eat. A failure with the food dispenser may result in numerous outcomes. If the food dispenser does not function at all, the pet may starve since there is no food being dispensed. If the food dispenser does function but does not stop, we have a chance to overfeed the pet. To ensure that the system does dispense, we will aim for a goal of the desired feed amount. was decided because the plus-minus range does not hunger the pet due to the small amount, but also does not overfeed the pet by a large amount. Once the food has been dispensed, two points of measurements will occur. As we are dealing with a device that animals will depend on, it is important that two measurements of testing should occur. Measurement one will be to use a measuring cup to calculate how much food is dispensed. Measurement two will be comparing the mass of the dispensed feed versus a desired food mass.

Another element that will be tested is the amount of time the dispenser needs to dispense the desired amount of food. Noted in Section 2.5.1, one of our hardware requirements is the dispensing time of the system will average 45 seconds, with a 1 second confidence interval. We do not want the dispensing to be too fast, or the dispensing screw will not gather enough food per rotation. Too slow, and the pet will be waiting for a significant amount of time for food.

7.4 Bowl and Lid System

The bowl and lid system is recognized to be a critical component of the system. If this section of the design fails, we risk not feeding the pet which can lead to health complications. To test the bowl and lid system, we will first ensure that the lid cannot be opened due to light use of force. This way, the pet does not destroy the design in search of food. Testing the bowl should determine if it can handle above 4 ½ cups of dry food to feed the largest animal based on the provided serving sizes. In tandem with the Collar Tag System, the lid will be tested to close within 60 seconds (with a ±10 second variation) after the pet leaves.

7.5 Software Testing

The system software is the other part of our pet feeder device system that will be checked. System software includes any part of the pet feeder device that is needed for it to work properly. It does not depend on any hardware that is unique to the pet feeder device or is calibrated to work with it. Below is a list of the system software that will be tested during this procedure.

Testing Procedures:

- Sign Up - User data storage and email verification
- Forgot Password - Password recovery in case the user forgets it
- UI Buttons - Test the buttons and make sure they work with the pet feeder device

7.5.1 Sign Up Testing

A user must sign up for the application in order to use the smart features of the pet feeder. Many of the features depend on syncing the pet feeder so that it can store information about a specific user. This is a very important feature. For a user to sign up for the app, their email address and password must be stored in the Firebase database, and they must also get a user ID that will be used to identify them. During this process, data is stored in two collections. Before a user has finished their verification process, their information will be saved in a temporary collection. After the user has been verified, his or her information will be moved to a permanent database. The sign-up process will be tested, and the database will be checked to make sure it works. The requirements for passing will be different for each of the tests below.

The temporary database collection will be at the center of the first test. On the page where you sign up, you will put in a test email address and password. After that, this information should be put into the temporary database. This test will be considered successful if the information was saved correctly in the temporary database but was not sent to the permanent database. Firebase compass will look in the database for the temporary and permanent collections to check the results.

The verification process will be at the center of the second test. As soon as a user signs up for the app, they should get a verification email to confirm their account. The information about how they signed in is then sent from the temporary database to the permanent database. To pass this test, the users should get the email within an hour, and their information should be added to the permanent database right away. Also, the data should be taken out of the temporary database as a separate check. From the time the test sign-up confirmation button is clicked until the email link arrives, a timer will be used. The link will then be clicked, and the Firebase database will be used to check the entry.

The last test will make sure that if someone tries to make multiple accounts with the same email address, the system will not let them. For this test, an account that already has an email address will be used to try to sign up and sign in. To pass this test, the sign-up page should show an error message saying that the email address has already been used to create an account.

7.5.2 Forgot Password Testing

If a user forgets their password and can no longer get into their account, they should be able to change their password in a safe way. These tests will be about sending the user an email with a link to reset their password after they enter their email.

The first test will start with the tester clicking "Reset Password" and entering an incorrect email. To pass the test, the system needs to check the permanent database and make sure that the email is not there. It then needs to tell the user that an account with that email address has not been created.

In the second test, the tester will put in an email address that is already linked to an account. To pass the test, the user's password should be changed in the permanent database and an email must be sent within an hour.

7.5.4 UI Buttons Testing

These tests will be less planned than the ones above, and they will focus more on how well the application works as a whole. It's important that all of the buttons in the app work as they should. In order to do this test, each button that should be able to be clicked or do something will be written down and tested on its own. So, a checklist will be made, and the tester will go through the app one button or UI element at a time to make sure it works as expected.

Some examples of this would be the different pages listed at the bottom of the main screen. If the user clicks on pet profiles, they should be taken to the dedicated pet page. When you press the notification bell, you should see a list of recent notifications that you can also click on. Some buttons will also only work at certain times, depending on what is going on with the pet feeder. Setting up the above tests will be needed to test these buttons, so tests will be done more than once.

At the end of the development process, these functionality tests must be run to make sure that adding new features to the app didn't break any features that had already been tested. For this set of tests to be passed, all buttons and user interface (UI) elements must work as they should. If the test fails and the code is changed to fix it, it's important to group everything that might have changed because of the new code and then retest. This will be an ongoing process that will make sure the application works perfectly in the end.

8.0 Design Constraints and Standards

This section details the standards and constraints that are related to the Pawsitive Pet Feeder. It will encompass the realistic design constraints we will have with the project's prototype and other directly impacted areas. By observing these standards and constraints, the product will be successful in the consumer market.

This section includes:

- Product Standards
- Standards for working with animals
- LED Safety Standards
- Electrical Power Safety Standards
- Financial Constraints
- Ethical Constraints
- Political Constraints
- Social Constraints
- Environmental Constraints
- Sustainability Constraints
- Manufacturability Constraints
- Presentation Constraints
- Product Fabrication

8.1 Product Standards

As we are dealing with the health and well-being of animals, the project must meet the standards placed by the United States Food and Drug Administration:

> *Store dry pet food [] in a cool and dry place. The temperature should be less than 80F, Excess heat or moisture may cause the nutrients to break down. – FDA.gov*
>
> *To pour the dry pet food into another storage container, make sure it's clean and dry, and has a lid that fits snuggly. A lid helps to maintain the food's freshness and to prevent your pet from getting into it. – FDA.gov*

To extend our focus on health and safety, a user's manual will be provided for the end user to operate the product safely and correctly. The manual will also include the proper use and environments to use the "Pawsitive" Pet Feeder. As well as include valuable information in regards to the components and products used within the project. This will contain: individual components used, a guide on how to use the product, and all of the design's standards and constraints. This is to mitigate user error that can cause injury to either the user or the pet.

As the product will need a Wi-Fi connection for communication to the application, a secure connection will be needed. This is considered to make sure that the user and the product's software will not be hacked and used for malicious intent. The application is also important due to an override feature that the end user will have to feed additional food to their pets.

Additional constraint is failure of function of the product. If the product does fail, the pet may not be fed, which can cause starvation and the possibility of death. Through testing, the design time will ensure the product functions properly and measures take place to prevent starvation.

8.2 Standards for working with Animals

The *Animal Welfare Act (AWA)* requires that the minimum standards of care and treatment be provided for certain animals bred for research or testing. This current statute was passed by the U.S. Congress and incorporated into the United States Code beginning at 7 U.S.C. 2131. The statute gives USDA authority to make regulations to implement and enforce the law by issuing regulations. Animals will not be needed for the research phase of the "Pawsitive" Pet Feeder. In the event any animal does get involved during the process, our group will comply with all statutes and standards mentioned in the Animal Welfare Act. The animal will be treated humanely and will be given any care necessary for their health and wellbeing. To emulate animals without working with live animals during the testing phase, stuffed animals may be used to test the project's features like the LEDs on the collar that will be worn by an animal. By using stuffed animals, the project's requirement specifications and features are able to be measured.

8.3 LED Safety Standards

LEDs are more commonly used in products and replacing other types of lighting. This is critical to the "Pawsitive" Pet Feeder as it will be used to determine which pet is approaching for food. However, there are safety precautions when it comes to the use of LEDs within the product.

When it comes to LED emissions and the health effects on the eye, we are using traditional LEDs for the design. These types of LEDs are considered safe, with no need for separate LED safety standards. Based on current exposure limits, most visible LEDs and infrared LEDs pose no acute hazard to the eye.

Risk Group	Philosophical Basis
Group 0 (Exempt)	No photobiological hazard
Group 1 (Low Risk)	No photobiological hazard under normal behavioral limitations
Group 2 (Moderate Risk)	Does not pose a hazard due to aversion response to bright light or thermal discomfort
Group 3 (High Risk)	Hazardous even for momentary exposure

Table 32: Exposure Assessment and Limits For LEDs

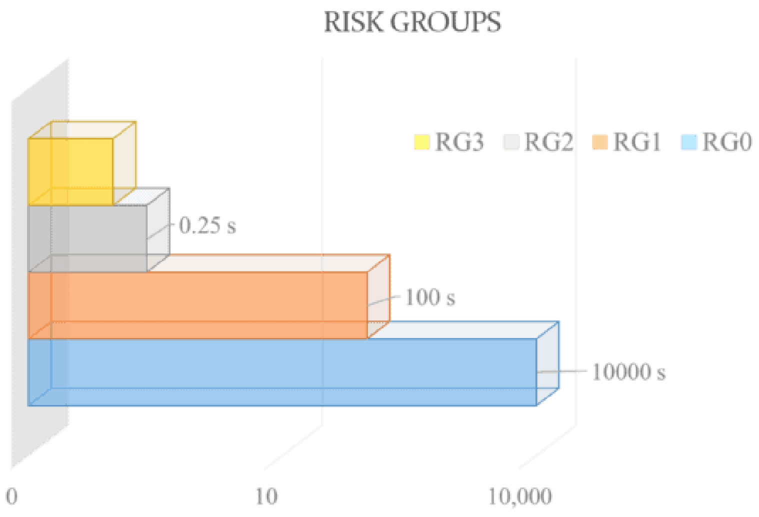


Figure 39: Risk Groups and Exposure Time (Courtesy of FireFlier)

As we are working with Animals, we will be working with LEDs within Risk Group 0. This risk group poses no photobiological hazard to individuals according to EN 62471: 2008 Paragraph 6.1. Exposure time for risk group 0 is approximately 10,000 seconds. It may pose a risk only when looking at the fixture continuously and directly longer for that. 10,000 seconds equates to 3 hours of continuous exposure. Therefore, with the activeness of a pet, it poses little threat to the animal. However, the design is aware of this and will ensure that the photobiological safety of the LEDs and the safety of the animals remains a priority.

8.4 Electrical Power Safety Standards

Power sources that will be used must be compatible with the power adapter that will be used. Power extension cords or power strips that will be used must meet the standards for consistent connected power and should not be connected simultaneously. Failure to meet these standards will impose a fire risk to the product, environment, and individuals within the vicinity.

To safely power the "Pawsitive" Pet Feeder, the power adapter must connect to a direct source that will provide the voltage required for the device while remaining grounded. This will include a wall outlet as it will provide consistent power to the device with little safety risk. As this is a core objective of the product, we will ensure that the wall outlet adapter is up to standards with current power regulations as it will be designed to power electrical components inside of the "Pawsitive" Pet Feeder.

8.5 Ethical Constraints

As mentioned with the standards for working with animals, animals are required to have standards of care and treatment during research and testing. When a project that directly affects the health and wellbeing of an animal by using electrical and computer components, the need to uphold and exceed these standards are needed. Safety is a priority in this project, whether it comes to the animals that will be needed for the project, end users and the project group members. All machinery and electrical components must be insulated at a designated section of the prototype where no individual or pet can be harmed. Additionally, components like the lid for the food storage and the device that will open the food stored into the designated bowl must be met with extreme care to not injure the pet.

As the project also involves optical components that are worn by the pets and the passive scanning by the camera, extra constraints will be considered. Optical components of the project must not interfere or damage the user or pet’s vision. Additionally, the collar needed to locate the pet through the camera must function to feed the pet and survive the normal wear and tear of an animal.

8.6 Financial Constraints

As the market stands, automatic pet feeders can range from $49.99 to above $300 via Amazon.com. With a target cost of $304 to build a prototype, our project will be in the upper mid-range of pet feeders on the market. However, the project will be feeding three different animals at the same time with one feeder. This allows our project in the upper-mid range to be of value as multiple pet feeders within that range only allow the feeding service of one animal. The target audience for our project will be owners who are very active, with multiple pets, and need a product that can feed those multiple pets while they are not home.

Furthermore, the group will be funding the project out-of-pocket. This will make budget allocation tighter, with the focus on finding alternative solutions to their expensive counterparts. However, as we find components that will be off the shelf and 3D printed, if the product requires maintenance after the initial build, it will be possible. The majority of the parts and components that are used for the design of the project are accessible to the public, and will reduce the need of purchasing another pet feeder. We will also ensure each group member provides consent and is comfortable spending budget on any purchase as the budget will be a shared responsibility.

8.7 Political Constraints

After investigating the project and its respective market, Group 8 determined that political constraints do not apply towards the Pet Feeder for Multiple Pets with Optical Recognition.

8.8 Social Constraints

There are social constraints when it pertains to the Pet Feeder for Multiple Pets with Optical Recognition. The target audience for our project will be owners who are active, with multiple pets, and need a product that can feed those multiple pets while they are not home. As the world opens back up due to the COVID-19 pandemic, individuals are looking towards a feeding solution for their pets as time will be spent on other activities and tasks. The product has a social responsibility to feed the pets for their health and well-being, as well as protecting the food from contamination from outside sources.

An additional social constraint that is considered is how much food each is provided per pet. The project does not want to malnourished the pets by giving less feed allotted. The opposite is also considered as too much food can be dangerous for the pet which can lead to health problems. In the application of the Pet Feeder, the end user may have the ability to adjust the amount of feed that will be given in the bowl.

As it relates to food, it is still the responsibility of the end user to check food for spoilage before feeding the pets. The "Pawsitive" Pet Feeder's design does not determine if the food within the storage system is spoiled. Without checking the food before going into the storage system, it runs a risk that the food is immediately spoiled and fed to the pets for an extended period of time. We caution pet owners to check each batch of food before refilling the Pet Feeder.

It is also the responsibility of the end user to ensure that the pet food purchased for the "Pawsitive" Pet Feeder is nutritiously sound for each animal. There are multiple formulas of food available on the market that cater to the age and activity of each pet. There are formulas that help senior dogs with healthy aging, and there are other formulas that help puppies with growth. To add, there are dogs with underlying health conditions that need special care with their food, different from the other pets in the household. Therefore, the end user has a social responsibility to make sure that each pet is getting the correct food for their own specific diets and health needs.

The last social constraint to be noted is the ability to refill the Pet Feeder in a timely manner to ensure continuous service for the pet. If the end user is unable to refill the pet feeder, it may lead to the pets not being able to be fed. Therefore, it is up to the project and the owner to determine when to refill the feeder. Whether when the storage is empty of all food, or if the user is alerted when the food has reached a certain low level.

8.9 Environmental Constraints

Pet Feeders in the market can be purchased, delivered, and activated within a day. The "Pawsitive" Pet Feeder needs multiple days to be set up due to using off the shelf components and 3D printed materials. However, as most of the components of the "Pawsitive" Pet Feeder are off the shelf or 3D printed, it can easily be replaced or recreated at a low cost. With regular maintenance, the "Pawsitive" Pet Feeder may have the ability to last for years at a time.

Additionally, the "Pawsitive" Pet Feeder feeds up to 3 pets at a time. This satisfies a need in the market for using only one device to feed multiple animals instead of an owner needing to purchase multiple devices to feed multiple pets. The ability for maintenance with quality components and eliminating the need for multiple devices can reduce environmental waste that comes from the frequent replacement of pet feeders.

8.10 Sustainability Constraints

Items that are similar to the "Pawsitive" Pet Feeder in the market are constructed for consistent use over long periods of time. Effectively, with the mindset of the product being used for years at a time. The "Pawsitive" Pet Feeder will be using parts that will be used constantly that will require it to be in motion once the device has been activated.

The sustainability of the "Pawsitive" Pet Feeder cannot be accurately determined due to the restricted amount of time allotted for the project. This includes the device's true lifespan in comparison to other products, and how the "Pawsitive" Pet Feeder's housing and parts would perform under normal wear and tear conditions.

During the design of the "Pawsitive" Pet Feeder, sustainability was considered when choosing motors and choice of materials. The quality of the motors and materials are selected to support the concept of the product and will positively affect a longer lifespan for the product. In the same context, using quality motors and materials will produce less waste and repairs. Ensuring that the consumer will only pay for the final product and will not expect surprise costs when using the "Pawsitive" Pet Feeder.

8.11 Manufacturability Constraints

During the design of the Pet Feeder, the group must consider the manufacturability constraints. For instance, we are choosing to 3D print components for the Pet Feeder to substitute buying parts from 3rd party sources. The 3D printed components must be up to standard for the longevity of the project for the end user. The components must not be seen as a cost-cutting measure but as an alternative solution to production materials. Lastly in the case of 3D components, it shall not be infringing on patented concepts as that may result in a violation.

When manufacturing the product, components that are critical to the "Pawsitive" Pet Feeder must be shielded. This includes the circuit boards, gears, motors, and electrical contacts.The components must be covered, insulated, and grounded for normal use. If a pet or individual comes into contact with this equipment, it can cause injury for the pet or individual due to the equipment themselves, or a malfunction to occur during the process of feeding.

The University of Central Florida has provided a laboratory to manufacture designs that will be beneficial to the project. Space in the laboratory is limited as other design groups will be using the space. However, equipment will be provided to ensure a safe and properly built product for the consumer.

Although the design group does have University of Central Florida’s resources at our disposal, time is a factor when it comes to manufacturability. Most larger companies and design groups will have more months dedicated to research and development. Along with the more months dedicated to research and development, the design group are still students at the University. Each member of the group would need to attend their daily classes, or work their job, etc. Research and Development groups only focus on the research and development with little factors contributing to the time invested. Therefore, time management for the manufacturing is wise to ensure a properly working Pet Feeder for the Senior Design course.

8.12 Presentation Constraints

The University of Central Florida maintains a no-pets policy on campus and in its on-campus residence halls. Therefore, for the testing and presentation portions of the project, we will be unable to show a live demonstration. However, this can be mitigated by using stuffed animals to act as the subject when testing the optical components of the Pet Feeder. Using stuffed animals will receive similar results as if they were used by their live animal counterparts.

8.13 Product Fabrication Constraints

To fabricate the "Pawsitive" Pet Feeder, we will be using off-the-shelf components. The project will be built based on the availability of components in the market. Therefore, there will be components that will be unattainable due to availability. We will be attaining off-the-shelf components through websites like Amazon, Wish, and Alibaba. We will also go to stores like Walmart and Hobby Lobby to receive components on the same day. Lastly, if components cannot be bought and delivered in a timely manner, and the component would be easier to replicate, we will 3D print the component to our desired specifications.

The overall design of the "Pawsitive" Pet Feeder can be replicated due to the use of off the shelf components. However, when it comes to the maintenance of the product, it will be beneficial as these components are publicly available.

9.0 Administrative Content

The administrative content provides an overview of the budget and financing of the project, as well as a milestone schedule. This is to ensure that that project does not go over budget as well as to keep group members accountable for finishing certain tasks by a designated time. Items within these sections are time based and will fluctuate between reports. The administrative content was written with current information as of November 20th, 2022.

9.1 Budget & Financing

Component	**Quantity**	**Total Cost Estimate**
Camera	1	$15
Red LED	8	$8.40
Green LED	1	$12.81
Blue LED	1	$7.13
Fresnel Lens	3	$97.68
Photodiodes / IR LEDs / Batteries	-	$50.63
Battery Holders	1	$8.01
Switches	1	$7.48
Stepper Motor (5 piece)	2	$29.18
Belker AC/DC Adapter Power Supply	1	$14.90
3D Printing (From Seminole County Public Library)	-	$33.50
A4988 Stepper Motor (5 piece)	1	$14.99

LM2596S Power Module (5 piece)	1	$9.19
Nema 17 Stepper Motor (3 piece)	1	$31.02
PVC Sheet	-	$70.00
Various Housing Materials	-	$30.00
USB Extender Cable	3	$21.00
Food Storage	3	$30.00
Lid Supplies	-	$10.00
	Total Cost:	$590.54

Table 33: Budget & Financing of Design

The total budget initially allotted for the "Pawsitive" Pet Feeder is $400. As of the completion of the project, we have spent a total of $590.54. We have went over budget for the project, however it was due to supply issues as well as components that were needed to be replaced. The final cost of the project will be split evenly across all group members.

9.2 Schedule

Senior Design I	Task Assigned To:	Start Date	Due Date	Status
Understand the Project	Group 8	8/23/22	8/30/22	Completed

Assign Roles	Group 8	8/30/22	8/30/22	Completed
Identify Parts	Group 8	9/6/22	12/1/22	Completed
Project Reports				
Initial Document	Group 8	9/6/22	9/13/22	Completed
Divide & Conquer:				
First Draft	Group 8	9/6/22	9/16/22	Completed
Second Draft	Group 8	9/16/22	10/7/22	Completed
Standards	Group 8	10/7/22	10/14/22	Completed
60 Page Draft	Group 8	10/7/22	11/4/22	Completed
100 Page Draft	Group 8	10/7/22	11/18/22	Completed
Final Senior Design I Documentation	Group 8	10/7/22	12/6/22	Completed
Research, Documentation & Design				
Microcontroller	Jocelyn	9/13/22	12/2/22	Completed
Camera System	Ervin	9/13/22	12/2/22	Completed
Fresnel Lenses	Isabella	9/13/22	12/2/22	Completed

IR LED’s & Photodiodes	Isabella	9/13/22	12/2/22	Completed
Dispenser Motors	Jocelyn	9/13/22	12/2/22	Completed
Wi-Fi	Ervin	9/13/22	12/2/22	Completed
Ultrasound Sensor	Ervin	9/13/22	12/2/22	Completed
Batter & Power Supply	Ayush	9/13/22	12/2/22	Completed
PCB & Schematics	Ayush	9/13/22	12/2/22	Completed
Lid, Bowl, Collar, Food Storage Design (3D Print)	Group 8	9/13/22	12/2/22	Completed
Application Design	Group 8	9/13/22	12/2/22	Completed
Optical Design Simulation	Group 8	9/13/22	12/6/22	Completed
Proof-of-Concept Test	Group 8	9/13/22	12/6/22	Completed

Table 34: Project Milestones SD1

Senior Design II	Task Assigned To:	Start Date	Due Date	Status
Order & Test Parts	Group 8	12/6/22	3/15/23	Completed
3D Print Parts	Group 8	12/6/22	4/7/23	Completed

Build Prototype	Group 8	1/1/23	4/1/23	Completed
Testing & Redesign	Group 8	1/1/23	4/10/23	Completed
Final Prototype	Group 8	4/10/23	4/17/23	Completed
Final Senior Design II Documentation	Group 8	1/1/23	4/25/23	Completed

Table 35: Project Milestones SD2

9.3 Conclusion

The "Pawsitive" Pet Feeder is a new type of pet feeder that uses both optics and electronics to automatically feed the user's pets when they are not around. Instead of worrying about getting home in time to feed a bowl, the owner can breathe easy knowing that their pets are not being neglected. We have set goals to relieve the stress of feeding your pet, recognize the pet is eating at the bowl, develop an application to remotely control the feeder and minimize materials to meet the price goal. Along with the objective, we have set objectives that we intend to meet to achieve these goals.

Although we recognize that there are automatic pet feeders out there, our design would like to take the best features from existing products and integrate them into one. Along with taking the best features, we also took into consideration their constraints to avoid or improve on these problems during our development. The "Pawsitive" Pet Feeder will be able to recognize a certain pet with LEDs and dispense the optimal amount of food based on their information. Once the pet has walked away, the lid will close and wait for the next pet to arrive.

10.0 Appendix

10.1 Camera System Code for Midterm Demonstration

The below code was used in the optical demonstration to show a working concept of our colored LED collar.

```
#Author:Isabella Pardo, UCFID:5041267

#Import useful libraries for the project
import RP1.GPIO as GPIO
from time import sleep
import cv2
import numpy as np
from picamera import Picamera #Import Raspberry Pi GPIO Library
from datetime import datetime

#taking the input from PICAMERA

vid = cv2.VideoCapture(0)

GPIO. setwarnings (False) #Here we are ignoring the warnings that are
possible detected in the pinout of the raspberry
GPIO. setmode (GPIO.BOARD)# Use physical pin numbering

#Here we are initializing the LED ports
#8 is for RED
#10 is for GREEN
# #12 1s for BLUE

GPIO. setup(8, GPIO_OUT, Initial=GPIO.LOW) #RED LED
GPIO. setup(10, GPIO OUT, initial-GPIO.LOW) #GREEN LED
GPIO. setup(12, GPIO. OUT.initial=GPIO.LOW) #BLUE LED

cv2. startwindowThread() #Here we are starting a thread to create a
window display of the camera

#This functions actually calculate which mean is greater for each
different color
#Then, it decides which one is it.

def WhatColor (b_mean, g_mean, r_mean):

 if (b_mean > g_mean and b_mean > r_mean) :
    print ("BLue"')
    return "Blue
```

```
elif (g_mean > r_mean and g_mean > b_mean) :
    print ("Green"')
    return "Green"

else:
    print ("Red"')
    return "Red"

#Main code

while True: # Run forever

#GREEN LED
    t1 = datetime.now()
    while (datetime.now()-t1).seconds <= 10: #Loop for 10 seconds
        GPIO.output (8, GPIO.HIGH) #Display GREEN LED
        GPIO.output (10, GPIO. LOW) #0ff Red
        GPIO. output (12, GPIO.LOW) #Off Blue

             #capturing the current frame

       _,frame = vid. read()

    #setting values for base colors
       b = frame[:,:,1]
       g = frame[:,:,1:2]
       r= frame[:,:,2:]

    #computing the mean
      b_mean = np.mean(b)
      g_mean = np. mean (g)
      r_mean = np. mean(r)

     #Call function to compare mean and evaluate the color
     color = WhatColor (b mean, g_ mean, r_mean)

     #Set text in the window frame
     cv2.putText (frame, color, (10, 50), cv2. FONT_ HERSHEY_SIMPLEX, 1,
(255, 255, 255), 4, 2)
     cv2.namedwindow ("frame")

     # displaying the current frame
     cv2.imshow ("frame" frame)



#RED LED

t1 = datetime.now ( )
while (datetime.now()-t1). seconds <= 10: #Loop for 10 seconds
    GPIO. output (8, GPIO. LOW) #GREEN OFF
    GPIO. output (10, GPIO. HIGH) #Display RED LED
    GPIO.output (12, GPIO. LOW) #BLUE OFF

    #capturing the current frame
       frame = vid.read()

    #setting values for base colors
       b = frame[:,:,1]
       g = frame[:,:,1:2]
       r= frame[:,:,2:]

    #computing the mean
      b_mean = np.mean(b)
      g_mean = np. mean (g)
      r_mean = np. mean(r)

     #Call function to compare mean and evaluate the color
     color = WhatColor (b mean, g_ mean, r_mean)

     #Set text in the window frame
     cv2.putText (frame, color, (10, 50), cv2. FONT_ HERSHEY_SIMPLEX, 1,
(255, 255, 255), 4, 2)
     cv2.namedwindow ("frame")

     # displaying the current frame
     cv2.imshow ("frame" frame)
```

```
#BLUE

t1 = datetime.now ( )
while (datetime.now()-t1). seconds <= 10: #Loop for 10 seconds
    GPIO. output (8, GPIO.LOW) #GREEN OFF
    GPIO. output (10, GPIO.LOW) #RED OFF
    GPIO.output (12, GPIO.HIGH) #Display BLUE LED

    #capturing the current frame
       frame = vid.read()

    #setting values for base colors
       b = frame[:,:,1]
       g = frame[:,:,1:2]
       r= frame[:,:,2:]

    #computing the mean
      b_mean = np.mean(b)
      g_mean = np. mean (g)
      r_mean = np. mean(r)

     #Call function to compare mean and evaluate the color
     color = WhatColor (b mean, g_ mean, r_mean)

     #Set text in the window frame
     cv2.putText (frame, color, (10, 50), cv2. FONT_ HERSHEY_SIMPLEX, 1,
(255, 255, 255), 4, 2)
     cv2.namedwindow ("frame")

     # displaying the current frame
     cv2.imshow ("frame" frame)
```

10.1 Acronyms

DC	Direct Current
IC	Integrated Circuit
IR	Infrared
LED	Light Emitting Diode
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistors
PCB	Printed Circuit Board
RAM	Random Access Memory
RGB	Red Green Blue
TBD	To Be Determined

Table 36: Acronym List

10.2 Work Cited

1. Lemonade, Team. “These Pet Ownership Statistics From Around the World Shed Light on the Most Common Pets, Costs of Owning a Pet, and the Rise of Pet Health Insurance.” *Lemonade Pet*, 4 Sept. 2022, www.lemonade.com/pet/explained/pet-ownership-statistics.
2. “Top 10 JavaScript Usage Statistics to Prove Its Awesomeness in 2022.” *Radixweb*, 20 Sept. 2022, radixweb.com/blog/top-javascript-usage-statistics.
3. *Structuring the Web With HTML - Learn Web Development | MDN*. 21 Sept. 2022, developer.mozilla.org/en-US/docs/Learn/HTML.
4. “Why Developers Like Bootstrap.” *StackShare*, stackshare.io/bootstrap. Accessed 3 Nov. 2022.
5. *Usage Statistics and Market Share of Bootstrap for Websites, November 2022*. w3techs.com/technologies/details/js-bootstrap. Accessed 3 Nov. 2022.
6. *Advantages of Java*. www.ibm.com/docs/en/aix/7.1?topic=monitoring-advantages-java. Accessed 3 Nov. 2022.
7. Chand, Swatee. “Top 10 Applications of Java Programming Language - Edureka.” *Medium*, 11 Dec. 2021, medium.com/edureka/applications-of-java-11e64f9588b0.
8. *Swift - Apple Developer*. developer.apple.com/swift. Accessed 3 Nov. 2022.
9. “Enterprise Open Source and Linux.” *Ubuntu*, ubuntu.com. Accessed 3 Nov. 2022.
10. *Visual Studio Code - Code Editing. Redefined*. 3 Nov. 2021, code.visualstudio.com.
11. “Original Choice | Dry Cat Food | Meow Mix.” *Meow Mix*®, www.meowmix.com/cat-food/dry/original-choice. Accessed 3 Nov. 2022.
12. “Cat Feeding Tips | Meow Mix.” *Meow Mix*®, www.meowmix.com/cat-care/cat-feeding-tips. Accessed 3 Nov. 2022.
13. “Frequently Asked Questions (FAQ).” *Meow Mix*®, www.meowmix.com/cat-food/faq. Accessed 3 Nov. 2022.
14. Center for Veterinary Medicine. “Proper Storage of Pet Food and Treats.” *U.S. Food And Drug Administration*, 14 Apr. 2020, www.fda.gov/animal-veterinary/animal-health-literacy/proper-storage-pet-food-treats.
15. *Animal Welfare Act | National Agricultural Library*. www.nal.usda.gov/animal-health-and-welfare/animal-welfare-act. Accessed 3 Nov. 2022.
16. Center for Food Safety and Applied Nutrition. “Bisphenol a (BPA): Use in Food Contact Application.” *U.S. Food And Drug Administration*, 27 June 2018, www.fda.gov/food/food-additives-petitions/bisphenol-bpa-use-food-contact-application.
17. *Equipment – TI Innovation Lab*. www.cecs.ucf.edu/innovationlab/equipment. Accessed 3 Nov. 2022.
18. www.mouser.com/ProductDetail/Vishay-Semiconductors/BPW82?qs=/jqivxn91ccMymVfeKLCiQ==. Accessed 3 Nov. 2022.
19. www.digikey.com/en/products/detail/vishay-semiconductor-opto-division/BPV23FL/4071207. Accessed 3 Nov. 2022.

20. www.newark.com/hamamatsu/s2386-5k/photo-diode-960nm-5pa-to-5-2/dp/21AH4487?CMP=AFF-CJ-100357191-Newark+Redirect+Link. Accessed 3 Nov. 2022.
21. www.digikey.com/en/products/detail/nte-electronics,-inc/NTE30159/11646909?utm_adgroup=NTE+Electonics,+INC. Accessed 3 Nov. 2022.
22. www.digikey.com/en/products/detail/kingbright/WP154A4SUREQBFZGC/3084119?utm_adgroup=Optoelectronics. Accessed 3 Nov. 2022.
23. "LED - RGB Clear Common Cathode." *COM-00105 - SparkFun Electronics*, 5 Feb. 2015, www.sparkfun.com/products/105.
24. *What Is Photobiological Safety Standard? – Fireflier Lighting Limited*. 1 Apr. 2021, fireflier.com/what-is-photobiological-safety-standard/. Accessed 6 Dec. 2022.
25. "Light-Emitting Diodes (LEDs)." *Ehs.lbl.gov*, ehs.lbl.gov/resource/light-emitting-diodes-leds/#:~:text=The%20focus%20of%20a%20safety%20assessment%20is%20RG3%2C. Accessed 6 Dec. 2022.