ALEXA AUTOMATED PET FEEDER

Final Paper

Group No. 18:

Carlos Lairet (CpE)

Liam Tsoi (CpE)

Cameron Nero (EE)

Jacob Paul (EE)

Customer:

Pet Owners

Table of Contents

List of Figures

List of Tables

1.0 Executive Summary

In today's world, Internet of Things is an ever-expanding realm. As more and more devices get integrated into the smart world, naturally, a Pet Feeder should arise. The product will allow the user to have the desired setting like when the food bowl will get refilled. The pet feeder will be voice enabled with Amazon cloud services since Amazon provided a lot of documentation on how to set up a device. Amazon also makes it so that you can have custom skills so that the device is not limited to a certain routine. We want to use a motion sensor so that when the animal walks up then it wakes up the device to see if there is a need for food to dispersed. We want to make sure that there is not just any movement that will wake up the pet feeder, so we want to add computer vision. The Jetson Nano will be responsible for the computer vision so it can learn and recognize when the pet is the one walking up to the pet feeder and not just somebody or another pet walking past it. There will also be weight detection which will check if there is enough food and to make sure that over feeding is not a problem. This will ensure that your pet has adequate food. Although this device is more expensive than the traditional way of manually refilling the food bowl, this product makes it much easier on the pet owner. The pet owner will not have to wake up in the middle of the night so they can feed the animal when they are hungry since some pets will try to get the attention of the owner. The goal of this project is to make the daily life of a pet owner easier and more convenient.

2.0 Project Description

The main goal of this project is to create an Alexa enabled pet feeder. The objectives are to allow users to easily feed their pets when they are at home, or on the go. We are going to utilize a scale to know how much food to give at one time. Additionally, we would like to incorporate a camera that will allow the user to see if their dog is eating, or if it has already eaten. The camera will also act as a safety feature incase that the scale that we are going to implements fails. We want this device to be easy to use, low cost, and run-on low power.

Easy to use is crucial for our project. We want the user to be able to easily integrate our device with their Alexa account. Moreover, we want it to be easy for the user to set the desired food quantity and dispense said amount. On ease of use another important factor is cleanliness. We want our feeder to incorporate a bowl that can be easily interchangeable and be removed for cleaning. Preferably the bowl will be waterproof, so that it can easily be cleaned in the sink. The main food container should also be easily removed and cleaned, to make sure that the device does not smell bad or is significantly dirty after weeks of use. We will include a bowl to make sure that the scale reading is accurate to how much food is going to be dispensed.

Low cost is also an important factor. There are solutions in the market for automatic pet feeders, but they are very expensive at over $80. Not only that, but they are not Alexa enabled. We want our device to offer great value to our customer and offer more functionality than the devices currently in the market.

Another important aspect of this project is RFID identification. The plan for this is to attach an RFID tag to the pet and once they are hungry and go over to the pet feeder. The RFID will be close enough to the reader and if enough time has passed, it will dispense food for the pet. This is another fail safe plan for the project incase nobody is home, and the Alexa does not have the automated feeder schedule turned on.

An aspect of this project is to have some sort of ultrasonic sensor that is inside the housing unit to alert the user when the unit is running low on food. This will be very important because the user would want to know when the feeder is running low on food in case, they are leaving the house for a period of time.

This project should also have a weight sensor that the bowl will sit on and once the user inputs a certain weight, the feeder will dispense food until that weight is reached on the sensor and will then shut off the motor. We choose to implement this idea into our project, because we wanted something to control the amount of food that is being given to the pet. Since we have multiple ways to dispense food in the project, we want it where no matter which method is chosen to dispense food, it will always dispense the same amount, so we are not over/under feeding the animal.

The idea is to make our pet feeder extremely low power. Some of the devices on the market must be plugged in, which is very inconvenient. By making our device battery power, it will make it easier to install, since the user can install it wherever they want without worrying about power. Since the device will be operated by batteries, there will be an indicator that will tell the battery level. The battery will not be some special component that you need to buy but instead will be AA batteries that you can find around the house.

There will be simple to use app that will allow the user to be able to dispense whenever they want. If the user dispenses food manually then the timer for the next automatic dispensing will refreshing to their desired time so the user does not have to worry about having too much food out. The app can give notification to the user like when to clean the bowl, battery level, and when the food level is running low.

2.1 Motivation

The Alexa Enabled Pet Feeder is a device that dispenses food for your pet. The feeder will be set up to dispense food whenever the user wants it to through the Alexa application or through an Alexa enabled speaker (for example an echo dot). The motivation for this project is that it can be very frustrating to leave your home only to realize that you forgot to feed your pet. This will allow you to keep your pets happy easily and conveniently. Since COVID started, users are often times staying at home and can feed their pets whenever it is time for them to eat. As the government begins lifting restrictions and people start to return to their workplace, their pets will once again be left at home alone.

Along with this, we as students will be graduating soon and entering the work force, leaving our pets, and wanting a solution to provide optimum care for our furry friends. This project could help save time and make it stress free.

2.2 Stretch Goals

Some stretch goals of this project include:

- Machine learning to recognize a specified pet (dog or cat) and dispense only when that pet approaches the unit. This would serve as an alternative for RFID sensing to operate the dispenser.
    - Constraints on this would-be budget. It will cost more to buy a processor capable of preforming machine learning. A quality camera will also be more expensive.
- Accommodating food storage to match the battery life goal of 7 days.
    - Constraints include having to increase the size of the storage container and possibly the whole unit. May need to provide an external food hopper to accomplish this.
- Solar powered battery charging. This would be helpful so that we do not have to worry about batteries dying and therefore we can stay away from home for a longer period of time if needed.
    - Constraints include budget and light source in the user's home.

2.3 Requirements and Specifications

In this section of the report, we will talk about the requirements and specifications of our project and what we expect to get out of our project. These requirements and specifications should be testable and be able to be verified or else they are not good requirements. An example of a requirement that would not be good is "this project should be lightweight." The reason that is not a good requirement is because you cannot determine what is lightweight and it is up for interpretation.

#	Requirement	Specification
1	Battery Life	The battery shall last a minimum of 1 week.
2	Weight Sensor Status	Battery should be able to be charged within 12 hours of being plugged in.
3	On Demand Feeding	Pet Feeder shall start dispensing food within 20 seconds after acknowledging command from Alexa.
4	Reservoir Sensor Status	Pet Feeder shall alert the user, via Alexa, that the reservoir is low once the sensor sees a distance greater than 15cm.
5	Power	The microcontroller shall be able to accept at least +5V
6	Wi-Fi Range	The range of the Wi-Fi module shall cover all the home or at least 100ft.
7	Cost	The cost of this project shall not exceed $400 total or equivalent to $100 per person.
8	Reservoir Capacity	Pet Feeder shall be able to contain 3 days' worth of food in the reservoir, or approximately 5 cups.

Table 1: Requirements and Specification

2.3.1 Housing Unit/Hardware Specifications

- The enclosure shall be built so that the electronics of it will not be able to be seen from the outside.
- The electronics will be safe from pet interference, tampering and slobber from eating.
- The removable bowl and dispenser shall allow for easy cleaning.
- The feeding bowl shall be interchangeable to accommodate pet preferred bowl types.
- The food hopper will be easily accessible for cleaning and refilling.
- The food dispensing control will be done using an auger type driver to release the food.

2.3.2 Electrical Specifications

- The microcontroller shall have enough pins to be able to support all the features of our project.
- The microcontroller shall be able to be controlled through an app, Alexa, and RFID tags.
- The pet feeder shall have Bluetooth/Wi-Fi communication so that the motor can be controlled through an Amazon Echo (Alexa).

- The bowl shall incorporate a scale to enable users to easily set a desired amount of food.
- The stepper motor shall allow for accurate food dispensing and enough torque to hold food that is in the dispenser.
- The battery will be large enough to power the unit for 1 week.
- The combined electrical hardware will be efficient enough to operate the unit for 1 week from the selected internal battery.

2.3.4 Software Requirements

- The stepper motor shall only turn on till the desired weight is reached and the stop and not dispense anymore food.
- The application shall be able to turn on the feeder as well.
- The stepper motor shall have a voice activated feature which will be hooked up through Amazon Echo.
- The software shall be able to extract the weight from the pressure plate to ensure correct amount of food is dispensed.

2.4 System Block Diagrams

In the next two sections of the report, we will insert the hardware and software block diagrams and how each of aspects of this project flow between one another. Each of these particular block diagrams will be color coated with a key on the right-hand side telling who will be doing research on which aspect of the project.

2.4.1 Hardware Block Diagram

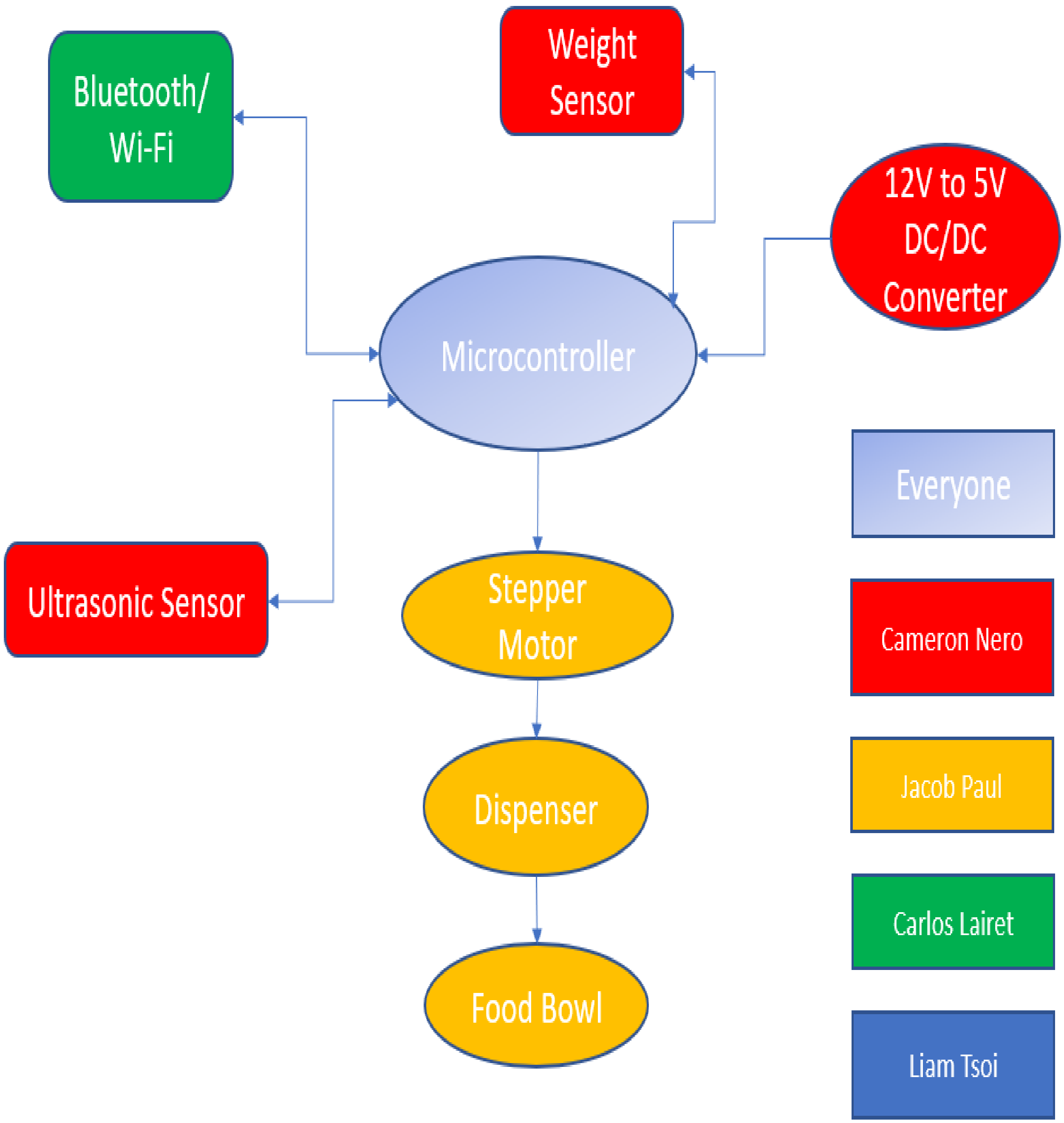


Figure 1: Hardware Block Diagram

2.4.2 Software Block Diagram

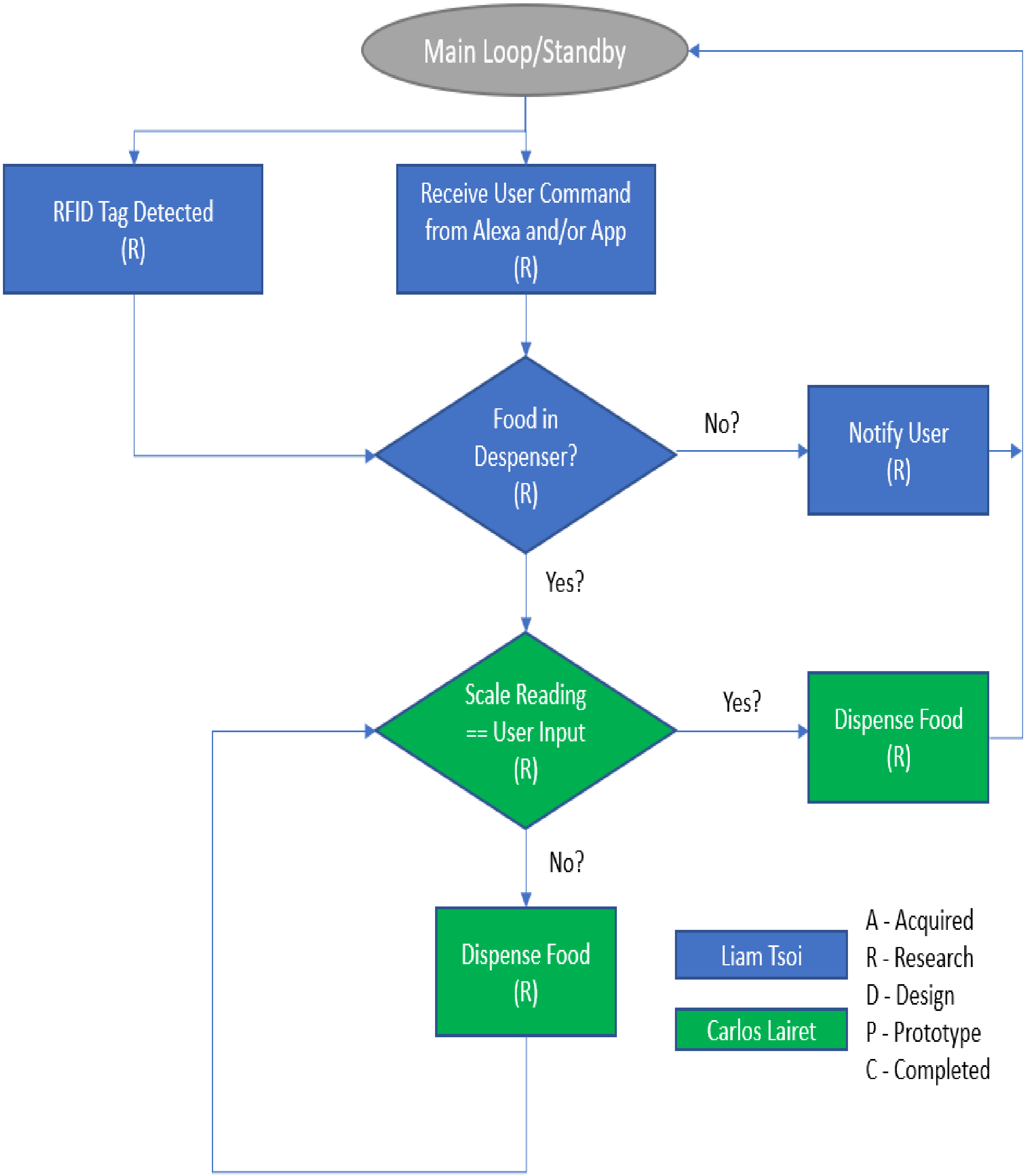


Figure 2: Software Block Diagram

2.5 House of Quality

The house of quality shown below (Figure 3) shows the customer and engineering requirements and specifications and their correlation to one another whether that be a positive or a negative correlation.

		Store Multiple Days of Food	Scale to determine when bowl is full	Battery Life	Weight Sensor	Motor	Bluetooth/WiFi	Food Level Warning	++++++++		
		+	+	+	+	+	+	+			
Design	+	↑	↑	↑	↓			↓			
Cost	-	↑	↑	↑	↓	↑	↑	↓			
Weight Sensor	+	↓	↓	↓	↑	↑	↓	↓			
Reliable	+	↑	↑	↑							
Monitor Food Levels	+			↑	↑						
Storage	+	↑									
Alexa Enabled	+			↑			↑				
		Hold up to 3 days worth of Food	Dispense only up to 10kg of food	Battery should last up to 1 week	RFID Should only dispense 2 times a day	Weight Sensor shall read accurately up to 9oz	The dispenser shall be able to be controlled thru Alexa	Sensor to tell the user when food levels are low			

Correlations:

↑ = Positive Correlation

↓ = Negative Correlation

Figure 3: House of Quality

2.6 Finances

This section will talk about the parts that will be required to complete this project and their estimated cost. Since this project has not been completely finalized with the exact parts list, this table is very tentative and is subject change throughout the next two semesters. The funds for this project will be provided by all four team members and will be split evenly.

Materials	**Cost**
MCU	$37.87
Housing	$70
Amazon Alexa	N/A (owned)
PCB	$85.63
Stepper motor	$19.69
Motor Driver	$8
Ultrasonic Sensor	$9
Pressure sensor	$12.99
3D printed parts	$5
Wi-Fi module	$8
Breadboard, wiring, LEDs	$39.20
Battery	$15.60
Battery Charger	$17.98
Miscellaneous	$30
Total	$358.96

Table 2: Materials and Cost

2.7 Milestones

In this section of the report, the milestones, and timelines of when our group would like to have things finished by will be mentioned. This will go for both senior design 1 in the spring and senior design 2 in the summer.

Spring 2021			
Week(s)	Milestone	Start	End
1 & 2	Come up with project ideas	1/11/2021	1/21/2021
3	Finish Divide and Conquer 1.0	1/22/2021	1/29/2021
4 & 5	Start Research and Update Divide & Conquer 2.0	1/30/2021	2/12/2021
6-8	Research Power Consumption and Constraints	2/13/2021	2/27/21
8-11	Research on Programming Microcontrollers	2/28/21	3/26/2021
12	Work and Finish 60 Page Draft	2/13/2021	4/2/2021
13 & 14	Continue Research and Finish 100 Page Report	4/2/2021	4/16/2021
15	Start to Order Materials/Components	4/16/2021	4/23/2021
16	Finish Final Document and Submit	4/16/2021	4/27/2021

Table 3: Spring 2021 Milestones

Summer 2021			
Week(s)	Milestone	Start	End
1 & 2	Start to build Housing Unit/ Prototype	5/17/2021	5/28/2021
3	Begin to test Components/Hardware/Software	5/28/2021	6/4/2021
4	Fix/Fine Tune Documents & Continue to Test	6/4/2021	6/11/2021
5	Finish Prototype	6/11/2021	6/18/2021
6	Make Adjustments/Continue to Test	6/18/2021	6/25/2021
7	Final Testing	6/25/2021	7/2/2021
8	Final Adjustments to Document	7/2/2021	7/9/2021
9	Finalize Project & Documentation	7/9/2021	7/16/2021
10, 11 & 12	Final Demonstration/Documentation	7/16/2021	8/6/2021

Table 4: Summer 2021 Milestones

3.0 Research

Before even beginning to build and test this project, we need to first do plenty of research all of the possible components that can be used for this project. This stage of the project is probably the most critical one because we are beginning to make decisions on which parts, we want to build this project with, so selecting a part that is not going to work is going to delay the progress on the project. The main research that is being done is for the hardware side and the most important component, the MCU. There will also be plenty of research on the software side as we need to make sure we all able to code in any particular IDE, understand communication protocol between devices/components.

3.1 Similar Products on the Market

Currently there are few Alexa Automated Pet Feeders on the market and this section is going to investigate their features, pricing and what they offer vs. what our project is going to offer.

3.1.1 Petnet SmartFeeder

Figure 4: Petnet SmartFeeder

This pet feeder is very similar to our idea, in that it is compatible with Alexa. This product also has its own app and sends notifications when the pet has been feed, the amount of which the pet was feed, when the food level is low, etc. This pet feeder also has a built-in camera that can be opened through the app to see your pet/make sure the pet has eaten its food. The cost of this pet feeder is about $180-$200 in used/like new condition according to Amazon. While this product is more than likely going to be a little cheaper than our pet feeder, ours will offer some machine learning that will allow for pet detection and if one is detected, food will be dispensed until a certain weight is reached.

3.1.2 PETKIT Automatic Pet Feeder

Figure 5: PETKIT Automatic Pet Feeder

This PETKIT Automatic Pet Feeder is also very similar to the previous pet feeder that we looked at that is already on the market. This pet feeder offers the ability to set a feeding schedule and the ability to feed the pet anytime through an Android and iOS application. Also, this pet feeder is Amazon Echo friendly (Alexa). The pet feeder will also have a sensor that can detect food levels while alerting the user when those levels are low. Lastly, it offers a weight sensor that will allow the user to accurately dispense as little as 1/5 cups to a maximum of 5 cups or 400g. The one thing this feeder is not going to offer which we plan to implement is machine learning that will allow for pet detection and if one is detected, food will be dispensed until a certain weight is reached. The cost of this pet feeder is around $140 according to Amazon.

These pet feeders in general offer a lot of what we are going to be offering besides none of these offer a pet feeder that has a camera and performs machine learning with the ability to detect the specific pet and being able to dispense food that way. The ability to identify the user's specific pet and be able to tell the difference from other pets/humans will make this product better. That is the main reason as to why our pet feeder will more than likely be more expensive that these other pet feeders on the market.

3.2 Food Level Detection Sensor

In our project there will be a sensor mounted on the inside of the housing unit to detect whether or not food level is low and that it needs to be refilled or not. The sensor would essentially look to see if anything is in front of it (in this case, food) and if it does not detect any food, we will have it alert the user. Obviously, we want the sensor to communicate with the microcontroller and if a flag is raised, it will then communicate with the Alexa software and either give an alert or send a text message saying that the "Food Level is Low". Now let us take a look at a few different kinds of sensors that would be compatible with this project.

3.2.1 HC-SR04 Ultrasonic Sensor

Figure 6: HC-SR04 Ultrasonic Sensor

This Ultrasonic ranging module can have a ranging distant from 2cm to 400cm. The basic principle of how this work is that it uses an IO trigger for 10us, and the module will send eight 40 kHz and detect whether or not a pulse signal is sent back. The working parameters for the HC-SR04 Ultrasonic Sensor can be found in the table below and will be used to compare to different sensor and we will be able to determine which is best. The data in table was found from the datasheet, given by Sparkfun.

Working Voltage	5V DC
Working Current	15mA
Working Frequency	40Hz
Max Range	400cm
Minimum Range	2cm
Measuring Angle	15 degrees
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and range in proportion
Dimensions	45x20x15 mm
Cost	$3.95

Table 5: HC-SR04 Specifications

3.2.2 Infrared Proximity Sensor -Sharp GP2Y0A21YK

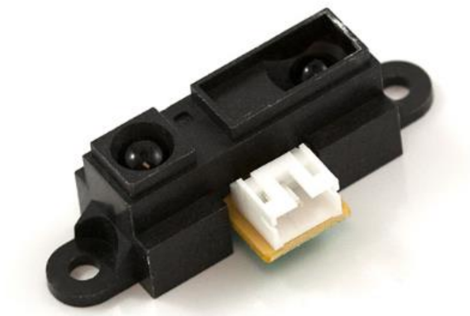


Figure 7: Infrared Proximity Sensor

This sensor uses infrared for detection. It has a minimum detection measurement of 10cm to a maximum of 80cm. This sensor could be very useful to us, with maybe an only constraint being that its maximum detection range is much less than that of the Ultrasonic sensor that was talked about above. The specs of this sensor can be found in the table below. The cost for this infrared proximity sensor is more than three times than that of the Ultrasonic sensor which could lead to some budget issues.

Operating Supply Voltage	+4.5 to +5.5V
Output Terminal Voltage	-0.3 to +0.3V
Operating Temperature	-10 to +60 degrees C
Detection Distance	10 to 80cm
Dimensions	28x14x2 mm
Cost	$13.95

Table 6: Infrared Proximity Sensor Specifications

3.2.3 HRLV-MaxSonar-EZ0 Sensor

Figure 8: HRLV-MaxSonar Sensor

Lastly, we have another Ultrasonic sensor which has a way greater detection distance compared to the infrared sensor. It can also detect from a further distance that HC-SR04 by a little bit but it also able to detect from a distance of 1mm, which is a lot more accurate than the previous two sensors that were looked at. The small dimensions of this sensor appeal to this project since it will not take up a lot of space when mounted on the inside of the housing unit. This sensor has a similar constraint to that of infrared sensor in that it is way more expensive than both of the previous sensors. The specifications of this sensor can be found in the table below, which will provide information from the datasheet.

Operating Voltage	2.5 to 5.5V
Current Draw	2.5mA at 3.3V, 3.1mA at 5V
Detection Range	1mm to 5m
Operating Temperature Range	-15 to +65 degrees C
Cost	$34.95

Table 7: HRLV-MaxSonar Sensor Specifications

3.2.4 Sensor Selection

For this project in particular, we choose to use the HC-SR04 Ultrasonic sensor due to the group's knowledge of this sensor from previous classes. Also, the very low cost compared to other sensors was another reason we choose this sensor since we want this project to be as cheap as possible since our group is supplying all of the funds. The compatibility of this Ultrasonic sensor with multiple different microcontrollers was very appealing as our group in case we need to switch microcontrollers at some point this semester. Lastly, the need for only two GPIO pins on the microcontroller was a big factor as we did not want the Ultrasonic sensor to take up a lot of the pins and therefore, we will have more freedom to include other features for our project.

3.3 RFID Module

For this project, we want to have some sort of RFID module as a sort of backup plan in case the pet owner is not home at the time or the Alexa feature is not working properly. Ideally, the pet feeder would have some sort of RFID reading module mounted to the front of the feeder and when the pet approaches the feeder, it is able to detect the RFID tag that will be hanging from the pets' collar and then will be able to dispense food.

We want to be careful not to over feed the pet, therefore, if the RFID is read during any point of the day, it will not try to read the tag for the next 24 hours. Being that we already have a primary way in feeding the pet in Alexa, we do not want this RFID module to act as the primary source of dispensing the pets' food.

RFID stands for radio-frequency identifications and uses EM fields to identify a specific tag on an object. For example, high level poker tournaments that are streamed on television use playing cards that have an RFID tag and once placed over the reader on the table, it is able to accurately tell the people behind the scenes which cards every player has and then they can show the whole cards on TV while the hands are still being played. The range of frequency required for this project is considered "High Frequency" meaning anywhere from 3MHz to 30MHz. There are also two different types of tags, one being passive and the other being active A passive RFID transponder means it is a transponder with no power source and cannot broadcast a signal. It must harvest energy emitted be an antenna. An active RFID tags have a transmitter and usually a battery as their own power source, these types of tags are very commonly used in cell phones. For our project specifically, passive tags will be used to them being cheaper and not requiring their own power source. Instead of an antenna, a RFID reader will be mounted on the pet feeder so that it is going to be in range of the tag. Next, we discuss some different tags that can be placed on the pets' collar.

3.3.1 RFID Tags

This section in particular will talk about two different RFID tags that could be possible for this type of project. The two RFID tags that are talked about in the next section are very similar and both of them operate at a frequency of 13.56MHz. The two are very different in their applications, one of them is similar to a dog tag and can hand from key chain or a dog's collar while the other is very similar to a credit card and must be placed in a wallet. Therefore, if we had a project design that required RFID to open a doggy door, the key chain might be the best option.

3.3.1.1 13.56MHz ISO14443-A Standard RFID Tag

Figure 9: 13.56MHz RFID Tag

This RFID tag acts as a key fob and should be able to be easily placed on the pet's collar without bothering pet since the weight of the tag is only 5kg. The communication between the tag and RFID reader module is used by a radio link with an operating frequency of 13.56MHz and through a proprietary protocol. When the tag is scanned by the RFID reader, a 16-bit CRC protection code is used and is specific to the one tag and ensures the integrity of the information. Meanwhile, the speed of data being transmitted can reach up to 100 kbits/second.

3.3.1.2 13.56MHz ISO14443-A Standard RFID Card

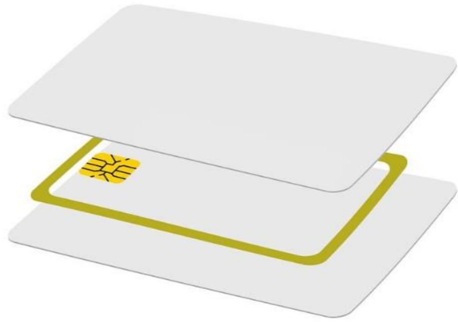


Figure 10: 13.56MHz RFID Card

Another type of tag that could be used is a card which operates exactly like tag that was previously discussed. The problem with this reader is that it would not work for this project specifically. Since we want to have something attached the pets' collar, a card would not be able to achieve that. These types of tags are mainly used to get into secured buildings. For this project, we will implement the 13.56MHz key tag because of its low weight, low cost, and its' ability to easily secured to the pets' collar without an issue. Next, we will discuss specific RFID readers that can be mounted to the front of the housing unit for the ability to scan the tag.

3.3.2 RFID Reader Modules

This section in particular will talk about two different RFID readers that could be possible for this type of project. The two RFID readers that are talked about in the next section are very similar except one of them operates at 13.56MHz and the other operates at 125kHz. The two are very different in their applications except for the fact of the operating voltage and operating frequency. Another main difference is the number of GPIO pins required by the microcontroller as the RFID-RC522 requires 8 GPIO pins while the 28140 Parallax reader only requires 4.

3.3.2.1 RFID-RC522

Figure 11: RC522 RFID Reader

This is a possible reading module that is very compatible with microcontrollers that are made by Arduino or Texas Instruments. This specific reader goes along with the two possible RFID key tags that were talked about prior because of the fact that the RC522 operated at 13.56MHz. Other specs of this part can be found in the table below. The one main constraint of these types of RFID readers are lack of long-range sensing, hence why implementing RFID is a stretch goal for this project. This reader, specifically, has a maximum detection range of 5cm, which might not be feasible for this project. Being that the key fob would have to hanging from the pets' collar, it would be hard to mount this reader in a place where it would be close enough to the pet to actually scan the tag.

Operating Frequency	13.56 MHz
Operating Voltage	2.5V to 3.3V
Communication	SPI, I2C and UART
Max Data Rate	10Mbps
Read Range	0-5cm
Current Draw	13-26mA
Power Down Mode Consumption	10uA (min)

Table 8: RC522 Specifications

3.3.2.2 28140 Parallax 125kHz RFID Reader

While this reader would not be compatible with the tags talked about in the previous section, it would almost be identical with the same key/cards at 125KHz. A reason to use this as possible solution is its low cost and rather simplicity to set up with a microcontroller. We see the same constraints as the RFID reader above and that is its low read range. Due to the lack of a high budget because the team members themselves are supplying all of the funds for this project, it is not feasible to find a low-cost long-range RFID reader that can also communicate with specific microcontrollers. The specifications for this RFID reader can be found in the table below.

Operating Frequency	125 kHz
Operating Voltage	4.5 to 5.5V
Communicating	USB or Serial
Current Consumption (Active)	100mA to 200mA (max)
Current Consumption (Idle)	10mA
Read Range	~4in

Table 9: 28140 Parallax RFID Reader Specifications

Figure 12: 28140 Parallax RFID Reader

3.3.3 RFID Reader/Tag Selection

Ideally if we can figure out a way to make sure that the read distance is long enough that a tag on the pets' collar will be able to read, we would implement the RFID. The reader that we were going try to work with will be the RFID-RC522 13.56MHz. This reader was chosen because of its compatibility with multiple different microcontrollers such as Texas Instruments, Atmega, etc. The ability to operate at a higher frequency compared to other ones that operate at 125kHz was also appealing because we are able to send information at a much quicker rate. Ultimately, RFID was not used in this project due to the fact that there were already plenty of commands and sensors that controlled the way the pet feeder worked, and we didn’t want ot add an RFID module. Additionally, the RFID module would have put us over budget so that was another reason as to why we did not choose to use the RFID module.

3.4 Microcontrollers

In this section, we will take a look at a few different microcontrollers and try to decide which is best for this project. This will objectively be the most important electrical decision because the microcontroller is what controls all of the different electrical components. The important criteria needed for the MCU is whether we can perform all of our main goals of the project without a problem. By that, we need plenty of GPIO pins to support different sensors, motors, motor drivers, Wi-Fi/Bluetooth module, etc. It also needs to compatible with the hardware and software so that no issues arrive when we go to build this project. Now, let's take a look at a few microcontroller options and decide which is best for this project.

3.4.1 Atmega328P

Figure 13: Atmega328P Microcontroller

This microcontroller is made by Microchip Pico Power and is an 8-bit AVR RISC-based microcontroller that has 32KB of flash memory with read-while-write options. This MCU in particular could prove to be very helpful since it has 23 GPIO pins and 32 general purpose working registers. Since this MCU is easily found on the Arduino UNO, which is a development board, it would be rather easy for us to be able to test/debug this specific microcontroller. The specs of this microcontroller can be found in the table below.

Program Memory Type	Flash
Memory Size	32 KB
RAM	2KB
Data Bus Width	8-bit
Operating Voltage	1.8-5.5V
GPIO Pins	23
Cost	$2.29

Table 10: Atmega328P Specifications

3.4.2 MSP430G2553

Figure 14: MSP430G2553 Microcontroller

The MSP430G2553 controller is a part of the MSP430X family of microcontrollers made by Texas Instruments. This microcontroller could be a really good choice for this project because of its simplicity and ability to use for a multitude of different projects. A reason this microcontroller is appealing to our group is our familiarity with it, as we used this back in our embedded systems class. The package and PDIP design of this controller allow for it to be easily programmed, since we could easily test on a dev board and then place onto our prototype circuit and test to see if it works. Since most of our group already has the launchpad, we would not need to pay the ~$10 for it and we would also not need to pay for the chip either since it can be easily removed from the launchpad board and placed onto a different circuit. The specifications of this microcontroller can be found in the table below.

Program Memory Type	Flash
Memory Size	16 KB
RAM	0.5 KB
Data Bus Width	16-bit
Operating Voltage	1.8 to 3.6V
GPIO Pins	24
Cost	$2.70

Table 11: MSP430G2553 Specifications

3.4.3 MSP432P401R

Figure 15: MSP432P401R Microcontroller

This is another microcontroller that is manufactured by Texas Instruments and is very similar to the MSP430X series of MCU's. It also very similar to microcontroller that was previously discussed but would be a more expensive option. We would need to purchase the development board to do some testing. The processor clock is very high compared to the two previous microcontrollers discussed. The clock speed along with a bunch of the specs can be found in the table below.

Clock Speed	48 MHz
Operating Voltage	1.62 to 3.7V
Storage	256 KB
# of GPIO Pins	48, 64, or 84
Cost	$7.87
Dev Board Cost	$19.99

Table 12: MSP432P401R Specifications

3.4.4 C8051F98x

This specific microcontroller is known for its use of low power and is considered the "industry's lowest power microcontroller." This 8-bit microcontroller created by Silicon Labs consumes only 150 uA/MHz while in active mode ad 10nA while in sleep mode. This is a clear advantage of the other microcontrollers we have looked at so far. The main constraints of this microcontroller compared to other ones mentioned in this project are its memory storage size and the RAM size. Those two being very low would make it hard to perform multiple large tasks at the same time. The table below will list out more of the specs for the C8051F98x microcontroller.

Memory Size	4 KB
RAM	512 Bytes
Active Current	150 uA/MHz
Sleep Mode Current	10nA
Operating Voltage	1.8 to 3.6V
# of GPIO Pins	17
Cost	$1.08

Table 13: C8051F98x Specifications

3.4.5 PIC16F877A

This microcontroller is very powerful, as it has 200ns instruction execution and it fairly easy to program. This microcontroller has plenty of pins which could be very useful for our project. Another positive is it is also a low power MCU and only needs 2 volts to sustain good clock speed. This chip comes with an already pre-enabled bootloader unlike all other microcontrollers we have looked at so far. The one constraint of this microcontroller compared other one is its very low memory size, which would make like difficult if we wanted to more complex programming which is what we are doing with a Jetson Nano/ Raspberry Pi. The table below gives the specs of this MCU.

Program Memory Type	Flash
Memory Size	14 KB
DRAM	368 B
Operating Voltage	2 to 5.5V
# of GPIO Pins	33
Cost	$5.43
Dev Board Cost	$26.29

Table 14: PIC16F877A Specifications

3.4.6 Microcontroller Selection

In the past sections, we have looked at a variety of different microcontrollers that were all great options to be the brains behind this project but only one can be chosen. The table below will compare all of the most important features of all the different microcontrollers and can help us make a more informed decision. The microcontroller will control all of the hardware aspects of this project so this decision cannot be taken likely. Taken the information that is in the table below, we will now do a more in-depth comparison between the MSP430G2553 and the Atmega328P. These two microcontrollers were our two final options because of their simplicity and previous exposure to them and their development boards.

Microcontroller	Cost	Operating Voltage	# of GPIO Pins	Clock Speed	Amount of Memory
Atmega328P	$2.29	1.8-5.5V	23	20 MHz	32 KB
MSP430G2553	$2.70	1.8-3.6V	24	16 MHz	16 KB
MSP430P401R	$7.87	1.62-3.7V	48/64/84	48 MHz	256 KB
C8051F98x	$1.08	1.8-3.6V	17	7 kHz	4 KB
PIC16F877A	$5.43	2-5.5V	33	20 MHz	14 KB

Table 15: Microcontroller Summary

3.4.6.1 Operating Voltage and GPIO Pins

One of the most important things when looking at different aspects of a microcontroller is the number of GPIO pins. We need to make sure that the chip we decide to use has enough of those pins to support all of the sensors/Bluetooth module/motor/etc. that is going to be in the project. When looking at the Atmega328P compared to the MSP430G2553, we see hardly any difference in the number of GPIO pins. The Atmega328P has virtually the same amount of GPIO pins so there is no difference when trying to compare these two different microcontrollers based on the number of pins.

Now, taking a look at operating voltage, we notice that for the Atmega328P, the operating voltage is much higher compared to the MSP430G2553. This is a benefit to the MSP430G2553 because when it is in its active mode, it is requiring less power. While the Atmega328P might require more voltage, it also provides more in other aspects. The table below will summarize when was just talked about.

Microcontroller	Operating Voltage	# of GPIO Pins
Atmega328P	1.8-5.5V	23
MSP430G2553	1.8-3.6V	24

Table 16: Atmega328P vs MSP430G2553 Operating Voltage and GPIO Pins

3.4.6.2 Cost of Chip and Development Board

One of the more important constraints in senior design is cost. Since our group is providing all of the funds for this project, we need to make sure we are being cost effective. The cost of these chips themselves are relatively inexpensive while the cost of the development boards to test all of our components can more expensive.

We also need to make sure that whatever microcontroller we decide to use is able to be placed onto our PCB, since that is a main requirement in this course. Because of this we have decided to not use any of the microcontrollers that do not offer a DIP package. We want to be able to easily program the MCU on its development board and then place onto our PCB. That is not possible with the non-DIP package because we would not be able to flash the program onto the MCU and then place onto our PCB because of the size of the pins.

MCU	Chip Cost	Development Board Cost
Atmega328P	$2.29	$23.00
MSP430G2553	$1.84	$9.99

Table 17: Atmega328P vs MSP430G2553 Cost of Chip/Board

As you can see from the table above, the MSP430G2553 is cheaper in terms of just the chip as well as the cost of the development board. While budget is an issue with this project, the MSP430G2553 does not offer the same number of features as the

Atmega328P for this small price difference to be the leading factor as to why we would choose the MSP430G2553 over the Atmega328P.

3.4.6.3 Memory Size

Memory size of the microcontroller is another huge aspect to this project. The size of the memory can allow us to do many things such as complex calculations or complex coding when communicating between multiple things at once. One of the main types of memory is RAM or better known as random-access memory which allows the MCU to retain information while running through the code or performing calculations. RAM is volatile which means that power must be on for it to store memory and once power is turned off, the memory is erased.

Another type of memory is Flash memory. Flash memory means memory that can store on board after specific code has been written onto it. Acting as the opposite of RAM, Flash memory is non-volatile which means the memory will be saved even when the power to the MCU is turned off. Ideally, we want a MCU with a large flash size so we can write more code onto the MCU. The table below will compare the RAM and the Flash Memory for both microcontrollers.

MCU	RAM	Flash Memory Size
Atmega328P	2KB	32KB
MSP430G2553	0.5KB	16KB

Table 18: Atmega328P vs MSP430G2553 Memory Size

3.4.6.4 Final Microcontroller Decision

We have chosen the Atmega328P as the microcontroller for this project. Even though this microcontroller was very close in a variety of areas to the MSP430G2553, the flash memory size and RAM size were a big factor in this decision. This microcontroller has many different hardware capabilities and should allow us to perform a variety of different things in this project. Given that everybody in our group is familiar with Arduino and this microcontroller in particular has full Arduino support, there should plenty of information on the internet to help implement everything that our project has to offer. Lastly, the development board, Arduino UNO, uses this microcontroller and will come with the MCU already having a bootloader, so our group should not have to deal with that. The development board is a little more expensive than the MSP430G2553 development board, but it has plenty to offer in terms of GPIO pins and plenty of memory. Being that this development board also has the DIP style microcontroller already installed, we should easily be able to remove the MCU from the development board to the already manufactured PCB and vice versa. This will be very beneficial in allowing us to easily test our code and make sure all the peripherals of this project are working properly.

3.5 Weight Sensing

One feature of the automated pet feeder is to allow the feeder to dispense a user defined amount of food with a minimum of 80% accuracy in terms of weight. The objective for this portion of the paper is to perform research on various types of weight sensing devices and determine which will fit our application the best. The determining factors to keep in mind when choosing the appropriate sensor include cost, accuracy, size, and power consumption. The various sensors that have been researched will first be discussed individually, then compared side by side. Following a side-by-side comparison will be the selection of the sensor with supporting data and reasoning for the decision.

A strain gauge load cell is a weight sensing device that produces a measurable voltage differential. These devices come in various shapes and sizes, yet their operation principles are similar. Resistors are positioned in the metal in a half bridge or Wheatstone bridge configuration that is highly sensitive to any sort of flexion in the material, essentially creating variable resistors. The resulting change in voltage is extremely small, order of millivolts. In order to get an accurate reading, the voltage must be passed through an amplifier and calibrated.

Load cells can be used in a variety of applications, which means there are many different styles of them. Through studying low weight appliances such as human body weight scales and kitchen food scales, it is clear there are a couple of types and variations of load cells to choose from for the food measurement on the pet feeder.

3.5.1 Button-Style Sensor

First, we have the button style sensor. This sensor is a small, flat cylinder with a small “button” on top where the load is to be applied. These sensors have the advantage of being highly accurate, and able to withstand heavy loads relative to their size. A key attribute to this style of sensor is there is little to no movement when sensing a load. This is useful for precision applications there is no unwanted movement when sensing a force. These sensors come with the cost of being more expensive due to the preciseness needed in production. Figure … shows an example of the compact nature of a button sensor.

Figure 16: Button-Style Sensor

3.5.2 Half-Bridge Strain Gauge Sensor

The next sensor to consider is the half bridge strain gauge sensor. This style of sensor uses a half bridge style circuit embedded in a flat, square body that allows a pressure point to be placed in the middle and will sense any pressure applied. These sensors can be used individually, in a pair, or with 4 devices to give weight readings. A benefit of this style of device is its low profile and the ability to pair multiple devices for higher weight distribution and reading. These sensors are relatively inexpensive and can be bought as kits with an amplifier included. Figure … is an example image of this style of sensor.

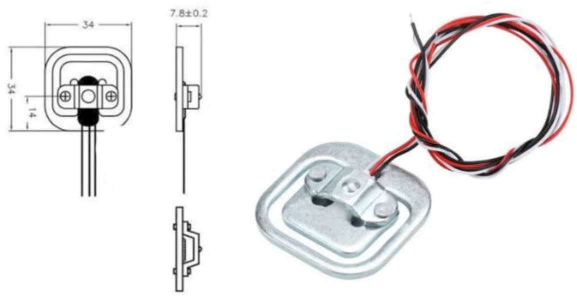


Figure 17: Half-Bridge Strain Gauge Sensor

3.5.3 Bar-Style Load Sensor

The last sensor under consideration is the bar-style load cell. This sensor mounts on one end using the predrilled mounting holes and suspends the other end, where the load is placed on top of. Using a full bridge circuit, also known as a Wheatstone bridge, this sensor accurately displays minute changes in resistance caused by flexion in the material, in the form of voltage changes. The voltage is then amplified into a readable level. The advantage of this sensor is its low cost, accompanies with its ease of use as it can be purchased in a kit with an included amplifier and pre-built mounting plates. It also comes in the desired range for food measurement (0-5kg).

3.5.4 Weight Sensor Comparison & Decision

Device	Operation Supply Voltage Range	Current Consumption	Size Specifications	Price
Button Style	<= 6VDC	N/A	9mm tall 19mm diameter	N/A, min 10 units
Flat Square Style	2.6 – 5.5V	1.5mA	34x34x7.8mm	$8 for 4 w/ amp
Bar Style	3.3 – 5V	1.5mA	80x12.7x12.7mm	$12.99 for kit w/ amp

Table 19: Weight Sensor Comparison

Viewing table above, it can be seen that the current and voltage specs are similar for the three devices, aside from the button style current consumption not being provided from the manufacturer. The size specifications vary with each sensor, with the button style sensor being the smallest and the bar style being the largest. The price point can really only be applied to the flat square style and the bar style. The distributor for the button style sensor requires a minimum order of 10 to even be able to provide a price per unit, so for that reason the button style is ruled out for our one-off pet feeder but will still be included in this report in case of mass production further down the line.

We have chosen to use the bar style load cell for our project. First, the bar style sensor falls within our desired operating voltage range. It also has a low current consumption, making it efficient. The biggest feature that set this sensor apart from the other two in consideration is the kit that is included with it. The bar style sensor we have decided on comes with an amp and mounting plates. These included features make this module easily integrated into our system and eliminates the hassle of mounting the sensor, which if not done correctly can affect the operation and accuracy of the sensor. All of these components included for a total price point of $12.99 makes this weight sensor a bargain.

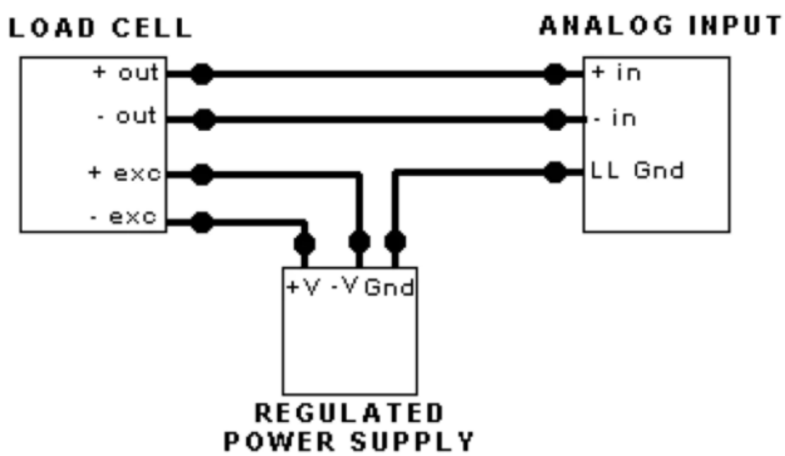


Figure 18: Bar-Style Wiring Diagram

3.6 Food Dispensing Motor

In the process of deciding what type of motor to use to drive the food dispensing auger, there are a few criteria that is to be considered. First, the motor needs to be able to provide a sufficient amount of torque immediately from a dead stop. Second, the motor should be efficient, as to not draw an excessive amount of power that cannot be supplied by a built-in battery. Third, the motor should be easily mounted and have an auger attached to the drive spindle.

There are two dominant types of electric motors on the market: stepper motors and DC motors. Addressing DC motors first, these are the most basic type of motor. A DC motor has two wire leads that are supplied with a positive and negative DC voltage specified by the manufacturer. The applied voltage induces a magnetic field on the inner rotating portion of the motor that opposes the permanent field, created by magnetic metal, in the outer portion of the motor, thus causing the motor to rotate. This simplified process can be seen in figure …. The more voltage that is applied to a DC motor causes the motor to spin faster, due to the nature of the induced magnetic field becoming stronger with more current flow.

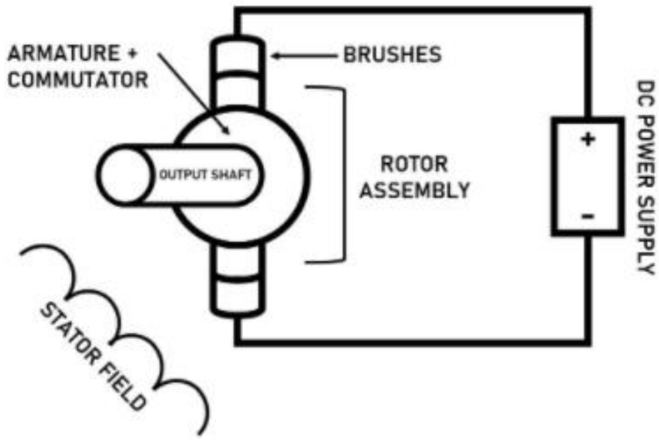


Figure 19: DC Motor

Moving on to the stepper motor, this motor is also powered by DC current but operates differently from the DC motor. The stepper motor utilizes about 40 poles (varies depending on the intended purpose of the motor) positioned evenly around the motor to total a 360-degree ring around the motor. Each pole is controlled successively from one another and pairs to the magnetic rotor to rotate the motor around. Figure … shows a simple, scaled down version of a stepper motor with only 4 poles instead of the usual 40+. The stepper motor requires external control from a microcontroller to energize the poles in the user desired pattern.

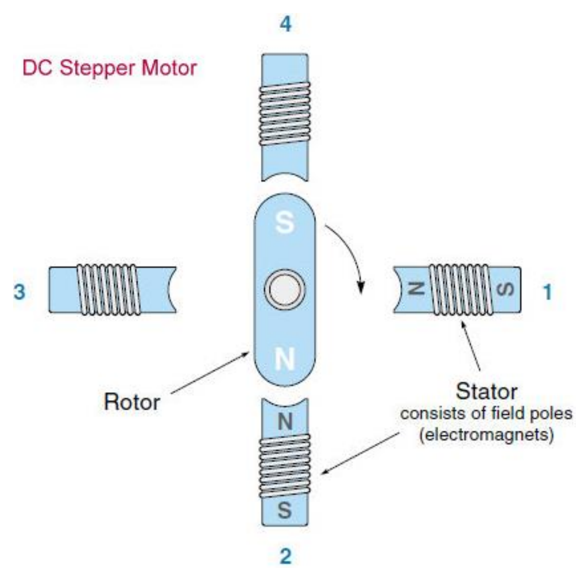


Figure 20: DC Stepper Motor

Now that we have a general understanding of each style of motor, we can compare each of their functions and features to the ones we defined as requirements. The first and most important requirement, to have immediate torque from a stopped position, is crucial because we want to eliminate the possibility of pet food pieces getting stuck and jamming the auger, preventing the motor from turning and dispensing food. DC motors come in various speed and torque ratios. Generally, the lower the RPM, the higher the torque. There is a drawback, however, the DC motor performs best when in motion. If the magnetic fields in the DC motor are not lined up just right at a stop, the motor will be at a mechanical disadvantage starting from a stationary position, therefore giving room for error. In comparison with the stepper motor, which will always have its maximum torque from a stopped position due to the high number or poles and ability to provide positive torque even when stationary, it can be confidently concluded that the stepper motor has the advantage for our first requirement. Moving to the second requirement, it can be simply put that the DC motor is more efficient than the stepper motor. This is due to the fact the DC motor is much more simplistic by nature. The stepper motor's need for an external microcontroller to operate the poles and the large amount of switching needed to rotate all the way around causes the stepper motor to be less efficient than the DC motor. Therefore, the second requirement is awarded to the DC motor. The third and last requirement of mounting convenience can be equally met by either style of motor. Both types of motors can be bought with pre-drilled mounting holes and/or mounting brackets. Even if there is an absence in mounting equipment included with the motor, a simple 3D print design can resolve mounting issues.

Each style of motor has an even number of advantages and disadvantages through our three categories of requirements. As a group, we have decided it is more important to reduce the likelihood of the motor stalling or getting stuck, and possibly preventing food from being dispensed to a user's furry friend. Due to the first requirement being of upmost importance, we have chosen to use a stepper motor. Although the stepper motor is not as efficient and will draw more power than the simple DC motor, we are willing to trade some efficiency for the benefit of reliability.

3.6.1 Motor Selection

The variety and selection of stepper motors in our project budget and power restrictions is limited. Outside of spending hundreds of dollars on one stepper motor, along with attempting to keep the voltage requirement for the motor under 12V, we are left with two feasible options: the 28BYJ-48 stepper motor and the Nema 17 stepper motor. Both models will be discussed and compared below.

3.6.1.1 Stepper Motor 28BYJ-48 5V

The 28BYJ-48 is a 5V, unipolar stepper motor. Unipolar means that current is sent to one pole pair at a time during operation, as discussed in the previous section when reviewing what a stepper motor is. The 28BYJ-48 is advertised as an inexpensive, versatile stepper motor meant for use on a variety of small electronics projects that require some degree of controlled rotation. This product is being considered for our pet feeder food dispenser because of its low price point and low energy consumption. This motor requires 5VDC and draws a max of 0.5A during full load operation. The low power consumption of the 28BYJ-48 comes with the cost of only producing a max of 3.4Ncm.

3.6.1.2 Nema 17

The Nema 17 series of bipolar stepper motors is well known in the 3D printing world as being a powerful yet reliable option for operating 3D printers. The Nema 17 comes in various sizes, listed from least to most powerful: 17HS2408, 17HS4401, and 17HS8401. There are also a few more models that are more powerful, but they are out of our budget range and also unnecessary to consider for our application due to being too large and powerful. The 17HS2408 has a rated current of 0.6A and a holding torque of 12Ncm. The 17HS4401 has a rated current of 1.5A and a holding torque of 42Ncm. The 17HS8401 has a rated current of 1.7A and a holding torque of 52Ncm. The Nema 17 motors come at a moderate price point, but as you can see have the drawback of being relatively power hungry.

3.6.1.3 Comparison and Decision

While the 28BYJ-48 comes with a very affordable price point and low per demands, we do not think it will provide our design with enough torque to consistently drive the food dispensing mechanism. This is a critical point in our design that must have zero failures or issues with operation. The Nema 17 motors are very reliable and have been used, tested, and reviewed on numerous occasions, used on 3D printers and by hobbyists. The unknown reliability and low torque specifications eliminates the 28BYJ-48 from our consideration.

Moving on to the Nema 17, we have three options to choose from. If power supply was not an issue, the choice would be simple: go with the most powerful model (17HS8401). This extra power would eliminate concern for food jamming the mechanism and stopping rotation. It would be able to push through and move the jammed food out of the way and continue dispensing. Although this would be simple, we do not have unlimited power and need to be conscious of power draw when operating off of a battery. Upon discussing as a group, we concluded that if we design the dispensing system correctly, we would be able to ideally eliminate the possibility of a jam. If no jams occur, there would be a minimal amount of resistance, therefore requiring a low amount of torque. The only resistance would be from the mechanism actually moving the food, and if designed correctly that can be minimized as well. If everything was designed perfect, we could easily choose the lower rated model of the Nema 17 series (17HS2408). As they say, expect the unexpected, the unexpected being a jam. We deliberated over a work around that doesn’t require brute force to clear a jam, and decided we can use the flexible nature of the stepper motor to our advantage. With some creative coding, any jams will be able to be cleared by momentarily reversing the auger, then driving it back forward. This is a feature that would be much more difficult to accomplish using a basic DC motor, hence why the stepper motor again proves to be superior.

Taking into consideration all of the points discussed above, the 17HS2408 stepper motor will be used for this project. We believe it will provide us with the power, mounting, and programming flexibility needed for the pet food dispenser. The Nema 17 draws more power than would be preferred, but this problem is manageable, and we are willing to work around it in order to maintain reliability.

Stepper Motor	**Rated Voltage**	**Rated Current**	**Torque**	**Length**
28BYJ-48	5V	0.5A	3.4Ncm	~25mm
17HS2408	12V	0.6A	12Ncm	28mm
17HS4401	12V	1.5A	42Ncm	40mm
17HS8401	12V	1.7A	52Ncm	48mm

Table 20: Stepper Motor Comparison

3.6.2 Motor Driver

Now that the food dispensing motor has been decided upon, a stepper motor driver must be chosen. The reason a stepper motor driver is necessary is because the poles inside of the stepper motor must be controlled in a sequence in order to make the motor function properly. The driver has a built in IC that delivers current to poles in an alternating fashion, called phases. When researching drivers, some key features we are looking for is the ability to output at least 0.6A and have two-phase-on driving capability. The reason we need two-phase-on driving is because our motor is a two-phase motor, along with the fact that having two phases on at a time produces more torque compared to just one.

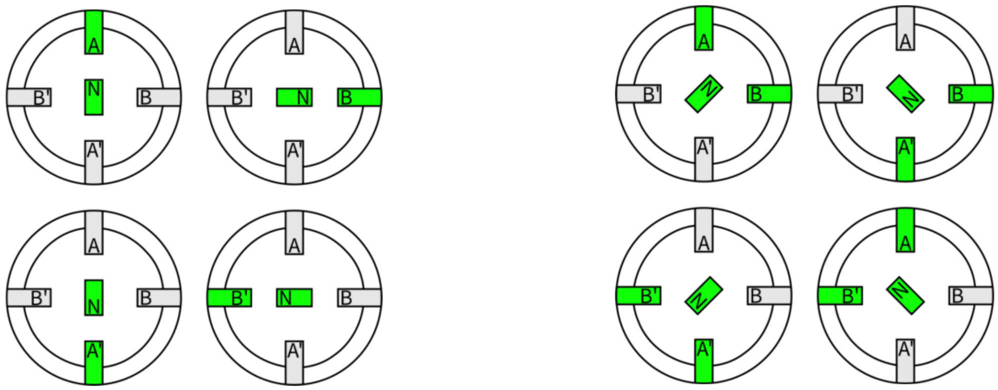


Figure 21: One-phase-on vs Two-phase-on

In this section, a few different models of motor drivers will be discussed and compared. The parameters of the motor driver need to accommodate the requirements of the Nema 17HS2408 stepper motor that will be used. The stepper motor must also be able to take input from the microcontroller to turn it on, off, and reverse.

3.6.2.1 TB6600

The TB660 stepper motor driver utilizes a PWM driven, chopper type bipolar stepper motor driver IC. It features built in temperature control and over current protection. It is often paired and sold with the Nema 17HS4401 stepper motor, which is more powerful than our chosen stepper motor. This driver has multiple output current options, with the lowest being 0.5A and the next increment being 1A. This output current is acceptable, but more flexibility may be desired. This motor has the advantage of being powerful, therefore it would be unlikely to see any heat issues when running on the low end of its capability. This powerful driver comes with the drawback of being power hungry, which is key to keep in mind for our project, given a limited power supply.

3.6.2.2 A4988

The A4988 is a DMOS micro-stepping driver that has a built-in translator. Having a built-in translator means that a minimum number of pins can be used from the microcontroller to control the operation of the driver and motor. This is ideal as one of our goals is to minimize the amount of GPIO pins that are used on the microcontroller for simplicity. This driver also features overcurrent protection. The driver accepts a logic supply range of 3.3V-5V from the microcontroller, which is acceptable for our standards. The A4988 also has an output drive capacity of 8V-35V and a 2A current supply. The output current is adjustable using a potentiometer, therefore the current out will be able to be adjusted to accommodate our motor.

3.6.2.3 A3967

The A3967 stepper motor driver is the most inexpensive, yet simplistic driver of the group. It is claimed to be compatible with anything that can output a 3.3-5V control signal. This diver also has a built-in translator, for ease of use and to minimize the amount of control pins needed. The operating voltage range for this driver is 7V-30V and has the ability to output 0.75A. This driver has the benefit of being inexpensive and easy to use, but although it designed for a 2-phase motor, there are no more built-in adjustments for stepping. There is also some concern that this driver may not be powerful enough. It does fall within the desired parameters for our motor but may not be able to overdrive the motor if needed and may get overheated if running at its max output consistently.

3.6.2.4 Motor Driver Comparison and Decision

Although all three of the motor drivers discussed would be capable of driving our stepper, we are looking for the driver that is best suited for our pet feeder. The balance between power delivery, power consumption, and reliability needs to find. Upon analyzing these three products, the A3967 can be immediately removed from the options due to its low output ability and potential reliability issues that could follow. This leaves us to choose between the TB6600 and the A4988. The TB6600 is definitely the most powerful and feature-packed option of the two but comes with the cost of being power hungry and lacks small increment adjustability. The A4988 on the other hand is not quite as powerful but is capable of outputting over twice the motors rated current. The A4988 also doesn't have as many built in features as the TB6600 but it does have all of the necessary and desired features. Based on the simplicity and lower power demands, the A4988 is the stepper motor driver that will be used to drive our Nema 17HS2408.

Stepper Motor Driver	**Output Voltage**	**Output Current**	**Multi-phase control**	**Price**
TB6600	9V-42V	4.0A max	Yes	$12
A4988	8V-35V	2A max	Yes	$10
A3967	7V-30V	0.75A max	No	$7

Table 21: Stepper Motor Driver Comparison

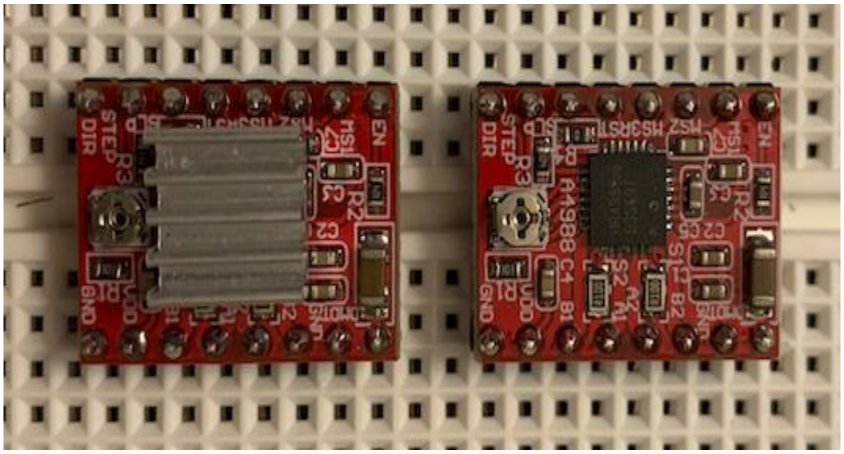


Figure 22: A4988 Driver with and without Heat Sink

3.7 DC-DC Converters

For our project, we are going to need a DC-DC convertor. Since we are going to have an external power supply that is 12V, we have to step down that voltage to 5V in order to be able to operate our microcontroller. This section will talk about different solutions to this problem.

3.7.1 Voltage Regulators

The power supply is once again one of the most important aspects to this project. We need to be able to supply enough voltage and current to both the microcontroller and the Jetson Nano so that both of them can run efficiently. One to do this is have a large enough power supply such as a DC voltage of 6V, 9V, or ever 12V and then use a voltage regulator to step down the voltage to either 3.3V or 5V depending on what the specific part is demanding. A voltage regulator is basically responsible for taking a specific input voltage and stepping it down to a different certain voltage and keeping it there to maintain the ideal voltage needed for that specific device. It ensures a steady constant voltage supply, and it even regulates voltage during powers fluctuations and when changing a load. Obviously, the main reason to have one is to take a higher voltage and supply a more stable out voltage but another reason to have a voltage regulator as part of a circuit is to have to so it will protect a circuit from voltage spikes that can damage and/or fry the board in worse case scenarios.

3.7.2 Linear vs Switching Regulators

This section will talk about the difference between linear and switching regulators.

3.7.2.1 Linear Regulators

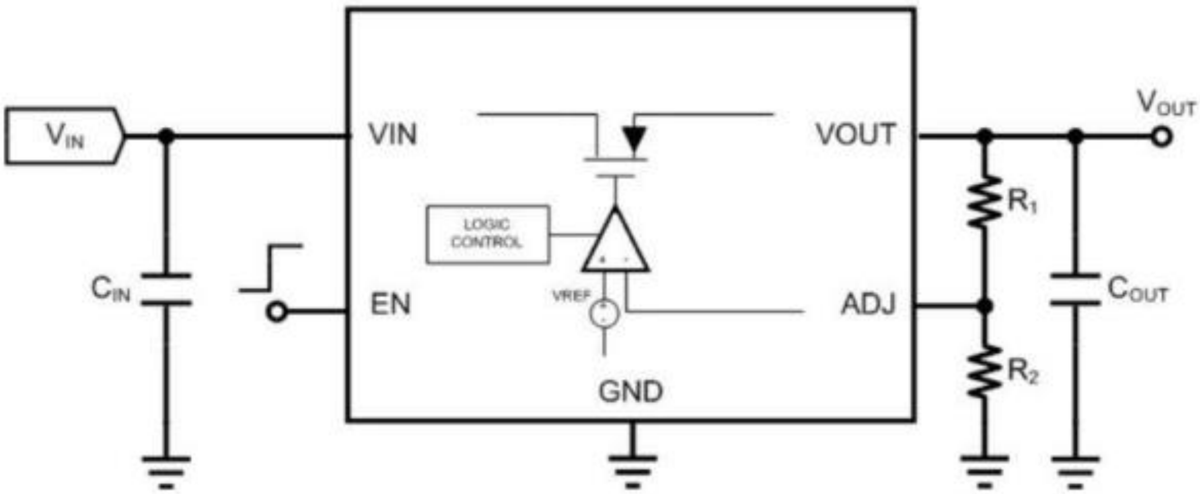


Figure 23: Linear Regulator

Linear regulators are mainly thought of as variable resistance devices because in order to maintain a constant output voltage, you have to vary the internal resistance. The variable resistance that we have to vary can be thought of as a transistor that is controlled by an amplifier feedback loop. All linear regulators require an input voltage at least some minimum amount that is higher than what the output voltage is going to be. That minimum voltage is called a dropout voltage. A low-dropout or LDO is a linear regulator which can regulate output voltage even when the input voltage is very close to the input voltage. One thing that a linear regulator cannot do is that it does not have the ability to boost a voltage. Therefore, if you have certain voltage such as 5V, you cannot use a linear regulator to boost the voltage up to let us say 12V. Linear regulators are a very simple and cheap to design but a usually very inefficient because the difference between the input voltage and output voltage is dissipated as heat.

Low dropout regulators can regulate all the down to 100mV but their ability to reject ripple and noise will reduce the input supply voltage by 500mV. This is not great for a design because you have to take in all of these different factors before you go a build one because problems might arise later on when testing a circuit. The main thing to look out for is heat dissipation and ripple that could cause the design to be no good if you do not take these factors into account before designing.

Design Flexibility	Buck Only
Efficiency	Low to Medium depending on difference between Vin and Vout
Complexity	Low
Size	Small
Cost	Low
Ripple/Noise	Medium
Vin Range	Narrow due to power dissipation

Table 22: Linear Regulator Summary

3.7.2.2 Switching Regulators

Switching regulators temporarily store energy and then release that stored energy as a different output voltage. Devices such as a DC-to-DC convertor, SMPS or switched mode power supply, switching regulator are all the same and operating by controlling an SSD that acts like a switch. Switching regulators rapidly switch a series element like a transistor or diode on and off. The operating with both synchronous and non-synchronous switches or better known as FETs. These switches have a duty cycle that can set the amount of charge that is going to be transferred to the load. The main advantages to using the switching regulator is that they are very efficiency and can operate at high power while the main disadvantages consist of them being more complex circuit and noisy and those are both something to consider when designing the circuit. Lastly, another main disadvantage is the higher cost compared to the linear regulator and that could play a factor in the economic constraints of our project. But, if the switching regulator is much better than the linear regulator, we might have to try to add more to our budget in order to support the design of the switching regulator. Two example circuits of how the switching circuit is setup can be found below.

3.7.2.3 Step-Down Switching Regulator

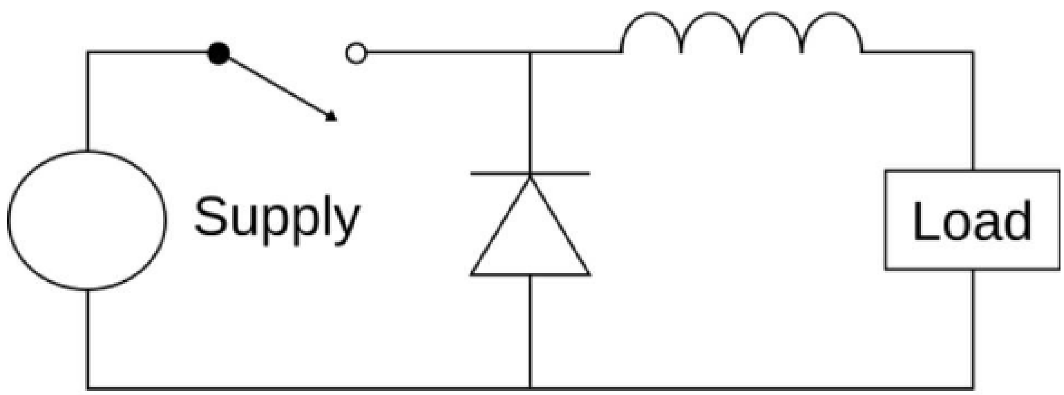


Figure 24: Step-Down Switching Regulator

A buck convertor or step-down regulator takes an input voltage and converts it to a lower output voltage. This is a similarity between the switching regulator and the linear regulator except for the fact that a switching regulator will consume a lot less power and therefore will be a lot more efficient than that of a linear regulator.

3.7.2.4 Step-Up Switching Regulator

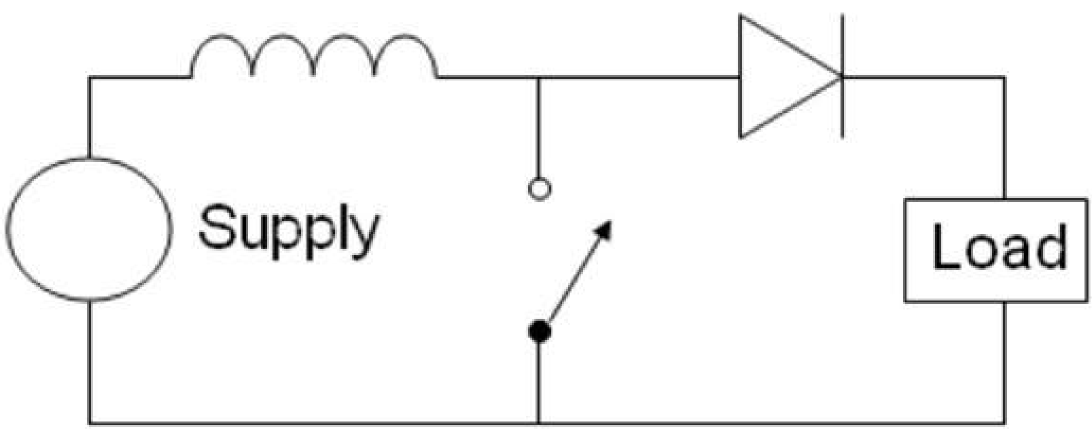


Figure 25: Step-Up Switching Regulator

A boost convertor or a step-up regulator takes a certain input voltage and converts it to a higher input voltage. This is one of the main differences between the switching regulator and the linear regulator because the linear is not capable of boosting a voltage. So, if you want to boost a voltage up, a switching voltage regulator is the only way to go. Considering that we are not really going to be using any boost regulators in this project, this is not going to be that big of a deal but due to the fact that the switching regulators give more options, it might be the way to go. The table below gives a good summary of switching regulators and highlights what is good and what is not so good about the switching regulators.

Design Flexibility	Buck/Boost/Buck-Boost
Efficiency	High
Complexity	Medium to High
Size	Depends on the switching frequency
Cost	Medium to High due to more external components
Ripple/Noise	Medium to High
Vin Range	Very Wide because of the lesser power consumption than the Linear Regulator

Table 23: Switching Regulator Summary

3.7.3 Power/Efficiency Comparisons & Conclusions

Now let us take a look at the power savings compared between a low drop-out linear voltage regulator and a switching regulator. The more we want to drop the voltage between these two regulators, the more power we are going to be saving if we decide to use the switching regulator compared to the linear regulator. The difference continues to increase as our output current goes up as well. So, for our project in particular, we are going to need a voltage regulator that can take 12V input and give an output of 5V. Because of the big difference between the input and output voltage, we are going to want to use a switching regulator even though the cost and complexity is going to higher. The graph below gives a good comparison in power savings between the two. For our project in general, we are going to go with a DC-DC convertor/Switching Regulator and the reason for this is because we want to be able to drop the voltage a considerable amount and using a switching regulator is going to give us the best option to do that. We also do not want to burn the microcontroller or board in anyway and if we were to choose to use the linear regulator, we run the risk of the output voltage not being as efficient as we like and that could harm the board if the output voltage ends up being higher than what we are designing the circuit to do. At the end of this section, we will take a look at several different switching regulator that step the voltage down from 12V to 5V and compare them. We will then in turn make a final decision and start the design process, testing of the regulator and start building the PCB with the regulator added to the circuit to support the input voltage of the microcontroller.

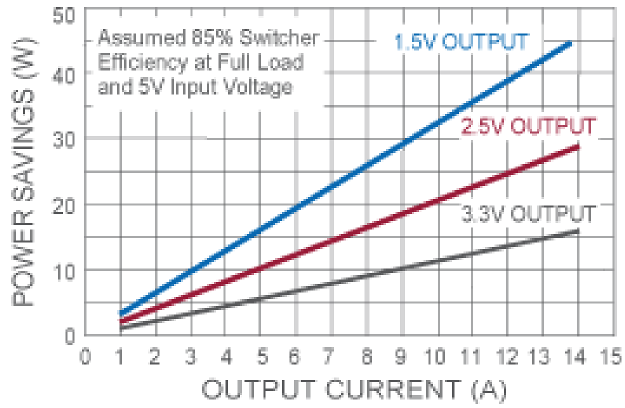


Figure 26: Power Savings vs Output Current

Now switching the focus to efficiency between the switching and linear regulators. As we can see from the graph below that we are going to get a constant efficiency for a LDO no matter which output current we desire, which is a good thing. The only downside to this is that the efficiency is still so low compared to that of a switching regulator that it still does not make much sense to use the linear regulator over the switching regulator. For the switching regulator, we see a parabolic shape for the efficiency with the highest point of efficiency being a somewhat medium output current and then continue to slowly decrease as the output current is decreased. Since we are going to need an output current of approximately 3A for our project, that is going to give us an efficiency of around 85% compared to around 65-70% to that of a linear regulator. So, with this given information, it would be best to choose the switching regulator over the linear regulator.

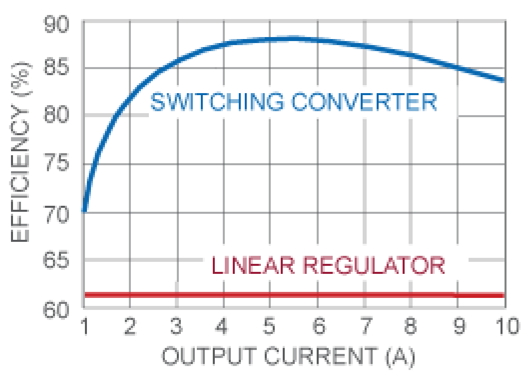


Figure 27: Efficiency vs Output Current

3.7.4 WEBENCH Tool Designs

One of the most important features of a lot of senior design projects is the ability to deliver power to all of the components of the projects, whether that be voltage or current. Another important aspect to keep in mind is keeping the parts as efficient as possible. For our project in particular, we need to have a battery that can supply enough voltage and current to the microcontroller. The Atmega328 microcontroller that we are using takes an input voltage of 5V. We plan on using a 12V battery to power everything, but we need to make sure that we step down the voltage to 5V to be able to power the microcontroller therefore we are going to need a DC/DC converter in order to do that.

Using the WEBENCH tool that is designed by Texas Instruments, we are able to find a good converter that will be able to take 12Vin and supply a voltage of 5V with enough current to be able to support all of other supporting features connected to the microcontroller. Below are the design parameters that were inputted to WEBENCH for where it will provide multiple circuits that can perform these parameters:

Input Voltage	Output Voltage	Max Load Current
12V	5V	3A

Table 24: WEBENCH Input #1 Specifications

After inputting these parameters into WEBENCH we are able to select from a variety of different DC/DC converter circuits. Due to economic constraints of our project, we need to pick a design that is relatively inexpensive and that does not have a lot of components since the more components the circuit has, the more expensive the converter will be. The design we choose, and the schematic can be found in the image below. This design uses an LM2576HV Step-Down Regulator.

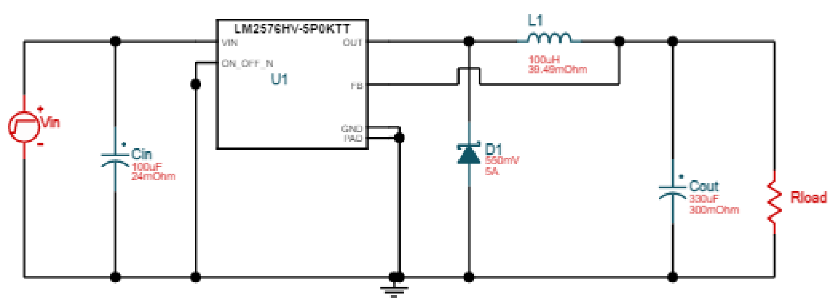


Figure 28: WEBENCH Design

In case we need more current such as 5A, we can also use WEBENCH to design a step-down switching regulator that can step down 12V to 5V with a maximum load current of 5A. Since we plan on using the Jetson Nano for this project for computer vision, it might be running at full power when preforming this and the maximum current draw for the Jetson is 3A so we might need a larger max load current that can support the Jetson, the MCU and all of the other sensors that the project requires. The input parameters into WEBENCH can be found in the table below.

Input Voltage	Output Voltage	Max Load Current
12V	5V	5A

Table 25: WEBENCH Input #2 Specifications

After inputting these parameters into WEBENCH we are able to select from a variety of different DC/DC converter circuits. Due to economic constraints of our project, we need to pick a design that is relatively inexpensive and that does not have a lot of components since the more components the circuit has, the more expensive the converter will be. The design we choose, and the schematic can be found in the image below. This design uses an TPS56637RPA Synchronous Buck Converter.

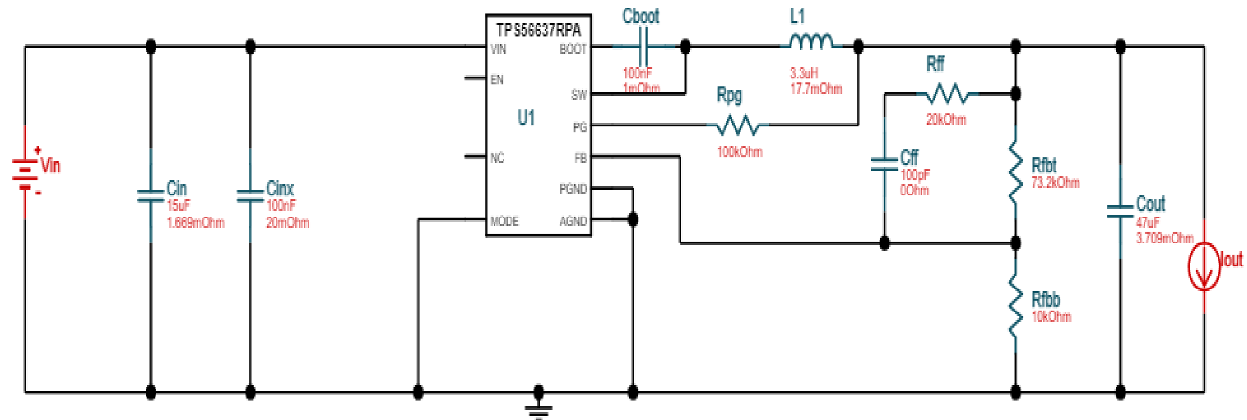


Figure 29: WEBENCH Design #2

3.8 Alexa Integration

In this section we will cover the tools and devices needed to setup Amazon Alexa on a device. Amazon offers a lot of different avenues and capabilities for Alexa integration. This is a critical decision as the implementation of this software will provide the core functionality to the Pet Feeder. The device responsible to handle Alexa requests must be able to communicate with all the other mechanical components of the Feeder and relate this information to the user. This research is critical for the software component of the Peet Feeder.

3.8.1 Amazon Developer Account

When making an Alexa enabled device you are going to have to make a developer account through Amazon. There is a section that you have to register a product for the

Alexa integration, this page will as for a name and the product ID. When register for your product to Amazon, you are submitting your product to be certified.

After making the account there is a section where you can start making skills from the Alexa Skills Kit page or manage the Alexa enabled devices through Alexa Voice Service.

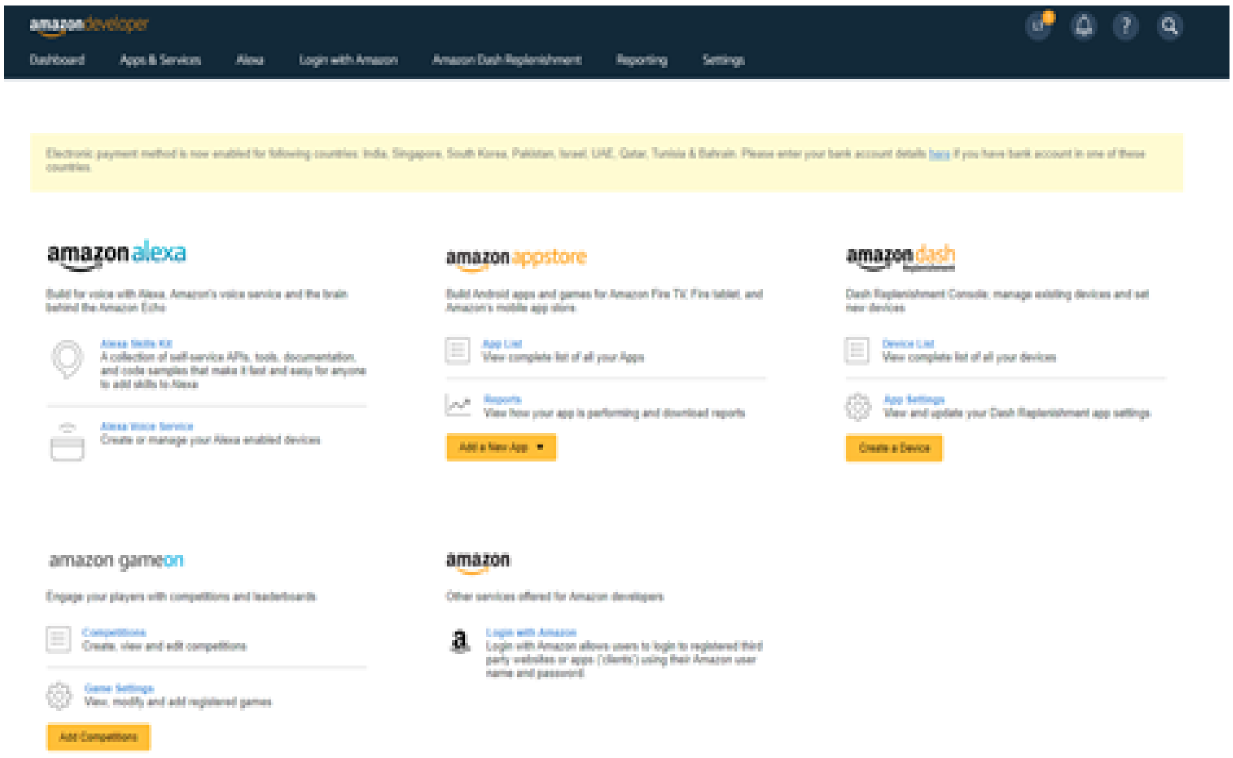


Figure 30: Amazon Developer Account

3.8.2 Alexa Skill

One of the options for integrating Alexa into a device is to develop an Alexa Skill. The Alexa Skill allows the user to interact with the device through the Alexa App or through any Alexa Smart Speaker (for example an Amazon Echo or Echo Dot). A skill works along with an Alexa Connected Device (in this case the Pet Feeder) to bring the user the desired functionality. It is referred to as a Skill because it gives “teaches” Alexa to communicate with a device. This device can be a Raspberry Pi, Jetson Nano, or type of Micro Controller (these alternatives will be discussed later).

There are multiple ways in which An Alexa Skill can be developed. The following are the possible methods:

Tools for Skill Development	Description
Alexa Developer Console	Using the Alexa Developer Console is extremely useful. It provides an all-in-one solution to skill development. Within this environment the skill's code can be written. Moreover, this tool allows you to simulate Alexa's responses and commands. Finally, this tool is very easy to use as it can run on a Web Browser and requires no Hardware to begin development.
Alexa Hosted Skill	Alexa hosted skills allows you to host the skill on AWS servers. This is not an ideal option, as it has limitations and requires more overhead. Nonetheless it is an option because, like the Alexa Developer Console, requires no initial hardware to deploy.
Alexa Skills Toolkit for Visual Studio Code	This Visual Studio Code extension allows you to have access to all of Alexa's prebuilt functions. These functions unlock all the possible interactions a user might have with Alexa. With these, the Alexa Skill can be developed.
ASK SDK	Alternatively, to the Alexa Skills Toolkit for Visual Studio Code, the toolkit can also be downloaded through Git Hub to have access to all of these libraries.
Alexa Skills Kit Command Line Interface (ASK CLI)	ASK CLI is a command line client that allows you to create new skills and deploy/simulate them. This tool is easy to deploy and offers a similar All-in-One Functionality like the Alexa Developer Console.
Skill Management API	Finally, the Skill Management API is also available from amazon, this is yet another way that Amazon allows you to implement Alexa Skills.

Table 26: Alexa Skill Development Tools

3.8.3 Alexa Built-In

Amazon has made it so that anyone can make devices that are Alexa enabled through the Alexa voice service or AVS, which is a cloud-based service. AVS has made it so that devices are able to have automatic speech recognition, natural language understanding, and text-to-speech. We could do this with our automated pet feeder so that we do not need an Alexa device present to be able to use voice commands on the feeder. AVS has development tools and documentation so that it is easier for people to start developing what they want. AVS allows the device that you are trying to develop to be able to process audio inputs which will then establish a connection to an Alexa device. As you can see in the figure below, the AVS will send the event to the Alexa which allows the Alexa to send back to the Alexa enabled device, the device will then respond with executed action.

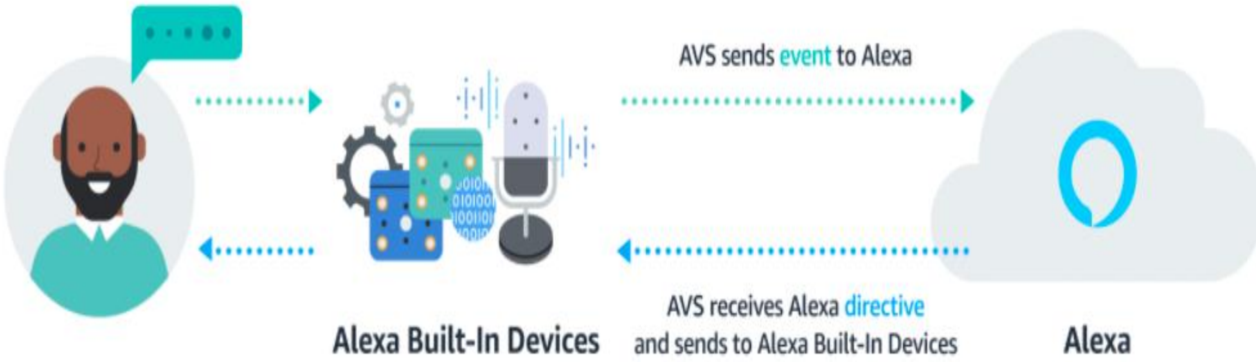


Figure 31: Alexa Built-In

AVS Device SDK is filled with libraries in C++ which will help build an Alexa Built-in product. The library allows any device that you want to be able to properly respond to a voice command that you have. If you have a smartwatch with you now, then you can use it to activate the pet feeder.

The figure below shows that when you say "Alexa" the device will start listening to what is said after that word. The data then goes to the Shared Data Stream (SDS) and from there, the information is split up and sent to two different locations. The two places that it goes to is the Wake Word Detection and Audio Input Process. The Wake Word Detection is where words are stored so that the Alexa device knows when to start listen to a command. That section of the commands is then sent to the Audio Input Processor. The next section is the cloud doing the job of communicating back the device, this is known as the AVS Protocol.

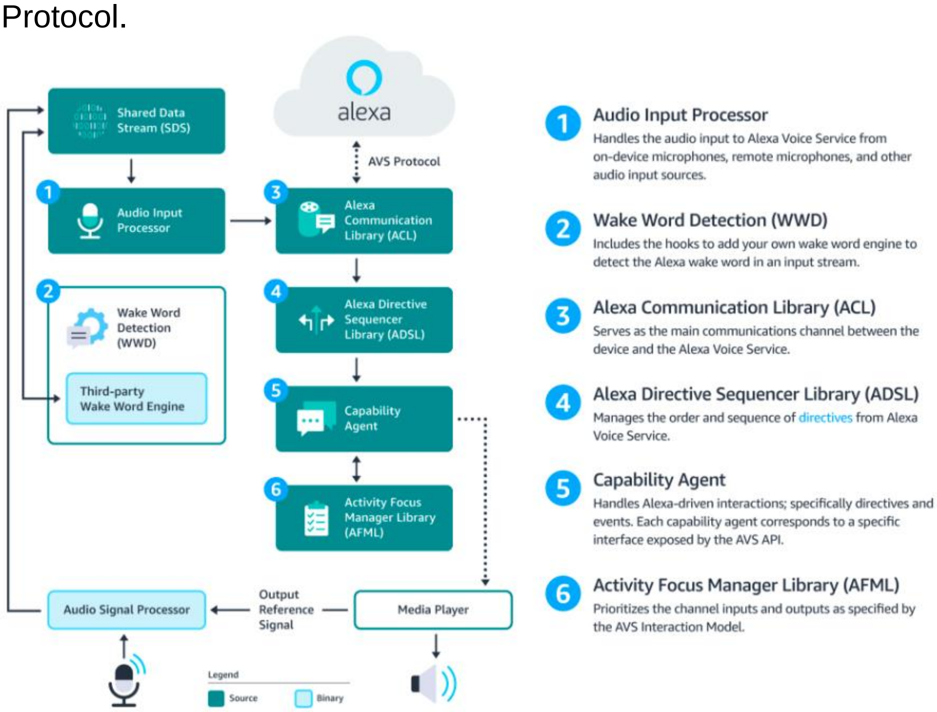


Figure 32: Alexa Chain of Command

3.8.4 Alexa Skill Development

The development of the Skill would require the use of one or more of the tools that were discussed above in the Alexa Skill (Section 3.51). The tools chosen by the team are the Alexa Developer Console as well as the Alexa Skills Toolkit for Visual Studio Code. Here the development process will be discussed.

First and foremost an Amazon Developer Account is required. After that, the Alexa Skills Kit SDK for Visual Studio can be downloaded. Then, the Amazon Alexa website discusses some ways to get started with a Hello World Skill. This is a perfect starting point to understanding Alexa Skill Development. In a nutshell, the SDK allows you to import request handlers for when the Skill is invoked by Alexa. Then the function of the code can be executed by the device.

Figure 33: Amazon Skill Development

As can be seen above, the user asks an Alexa Built-In device (this could be an Amazon Alexa Echo Dot or Echo Show) to perform a certain action. This action, in our case, could be to check if the dog has food or to dispense food if there isn’t any. After hearing the request, the Alexa service on the Cloud will recognize that the skill must be called upon. Here is where the request handler in the code comes into play. This piece of code is always waiting for Alexa to engage the skill and “handles” the request, hence the name request handler. Then it executes whatever the Skill does.

3.8.5 Alexa Skill Hardware Implementation

Since the Alexa implementation will be done through a skill, the correct hardware to run the skill must be chosen. There are multiple considerations when picking the correct device to operate the Alexa Skill requests. For example, this device must interface with the other microcontroller as well as have connectivity to the internet. These are just some of the factors that must be considered. In this section, different pieces of hardware will be considered for driving the Alexa Skill.

3.8.5.1 Node MCU

The model of the NodeMCU that is a potential option to use implements the ESP8266, which is manufactured by Espressif Systems, because it has Wi-Fi capabilities, this will help with the Alexa integration. The ESP8266 also has a full stack TCP/IP stack, has I2C software implementation. There is UART dedicated pins and there are options between the Arduino IDE and ESPlorer, which is meant for ESP8266 developers. The difference between the two options is that the ESPlorer uses a different language that most people will not know about if the user is used to working on ESP or embedded programming. The language that will be used for ESPlorer is Lua, while Arduino uses C++ which might make it a better choice since the different between C and C++ are not that much different. The table below will show the different types of modules.

	ESP-01	ESP-05	ESP-12	ESP-201
GPIO Pins	2		11	11
Form factor	Small	Small	Medium	Big
Price	$3	$3	$3	$3

Table 27: NodeMCU Model Comparison

Depending on if we are going to be doing a standalone module, then the ESP-01 is a good choice depending on size is a main concern because the ESP-01 is small and has GPIO pins too. This is the only option in the small form factor while including the GPIO pins. The other option that is available is the ESP-12 but is listed as a medium sized board with 11 GPIO pins. One of the big disadvantages of both these modules is that they are hard to test on breadboards because of the placements of the pins makes it hard to plug it directly into the breadboard. The solution for this is to use a breadboard adapter or use a female to male Dupont wires.

The NodeMCU V1.0 module comes with a built in Micro-USB plug to help with powering and programming the device. This model incorporates the ESP-12E module and has 4MB of flash memory. The advantage of this over its older version is that it is easier to prototype on a breadboard since it is not as big as the V0.9 model.

	NodeMCU V0.9	NodeMCU V1.0	Wemos D1 Mini	Wemos D1 R2
GPIO Pins	11	11	11	11
ADC	1	1	1	1
Price	$3 - 4	$3 - 4	$3 - 4	$6

Table 28: NodeMCU Price Comparison

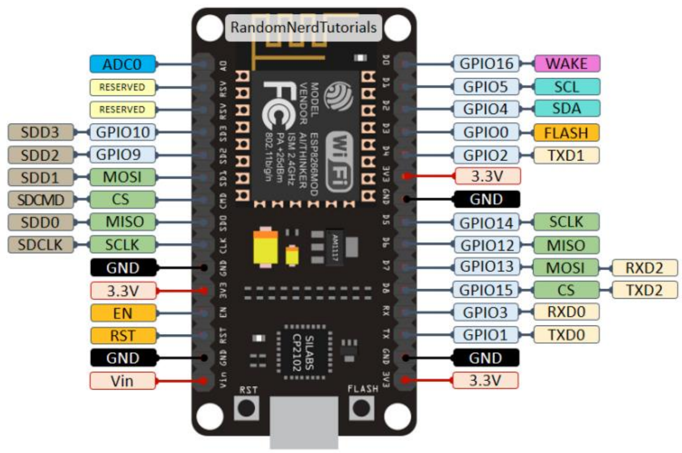


Figure 34: NodeMCU Pinout

The NodeMCU Development board has up to 16 GPIO pins but since the ESP is a System of a chip (SOC), this means that the chip is an entire system that is inside a chip, then there are some lines that are used internally. This leave about 11 GPIO pins but 2 of the pins are not available to use because the GPIO3 and GPIO1 pins are the Tx and RX pins for UART communication. The NodeMCU has pin names on the development board that are linked to the actual internal pins of the ESP8266 MCU. The table below shows the mapping of pins names to their respected GPIO pins. The remaining pins that are available for use are D0 to D8. Since the NodeMCU uses the ESP8266 then UART 0 is the only one that can receive data and UART 1 is the only one that can transmit data. There are two libraries that is available online that might be important for the implementation of connection multiple devices to the NodeMCU. The library ESPAsyncTCP helps with the functionality of the Wi-Fi capabilities. The ESPAsyncTCP is a fuolly asynchronous TCP library that can help with multi-connection network environment. The other library is called FauxmosESP, which is a library that helps with instantiating objects, adding the capability to add more devices to the NodeMCU, and making connecting to the Wi-Fi easier. The FauxmosESP is the primary thing that is going to enable the pet feeder to connect to Alexa and listen for the commands. When the NodeMCU is connected to an Alexa enabled device like the Echo and if you want the voice enabling functionality then there must be a method that is written in the code for this feature to be present.

3.8.5.2 Raspberry Pi

In this section the implementation of the Raspberry Pi for the Alexa Skill will be covered. Moreover, some of the specifications of the Raspberry Pi will be covered. The Raspberry Pi is a very popular alternative for driving an Alexa Skill. Moreover, it can also be used as an Alexa Built-In device. Setting up a Raspberry Pi to work with Alexa is extremely easy. First the Raspberry Pi needs to be registered as an AVS device with the Amazon Developer Portal. Next the AVS Device SDK dependencies must be installed and configured on the Pi. Finally, Amazon offers a sample Alexa application to test that the Raspberry Pi is working correctly.

There are different Raspberry Pi's that can be used to run the Amazon Alexa SDK's. Amazon requires that a Raspberry Pi 3 or 4 be used. Here are some of the features between the different Raspberry Pi Models:

Component	Raspberry Pi 3 A+	Raspberry Pi 3 B	Raspberry Pi 3 B+	Raspberry Pi 4
CPU	Broadcom BCM2837B0 Cortex-A53, Quad Core 1.4 GHz	Broadcom BCM2837 Quad Core 1.2 GHz	Broadcom BCM2837B0 Cortex-A53, Quad-Core 1.4 GHz	Broadcom BCM2711, Quad Core Cortex-A72 1.5Ghz
RAM	512 MB LPDDR2	1GB Ram	1 GB LPDDR2	2GB, 4GB, 8GB LPDDR4
Wi-Fi	2.4 GHz and 5 GHz IEEE 802.11.b/g/n/ac	BCM43438 Wireless LAN	2.4 GHz and 5 GHz IEEE 802.11.b/g/n/ac	2.4 GHz and 5GHz IEEE 802.11ac
Bluetooth	Bluetooth 4.2, Bluetooth Low Energy	Bluetooth Low Energy	Bluetooth 4.2, Bluetooth Low Energy	Bluetooth 5.0
GPIO	40 Pin	40 Pin	40 Pin	40 Pin
Power	5 V DC @ 2.5 A (Micro USB)	5 V DC @ 2.1-2.5 A (Micro-USB or GPIO)	5 V DC @ 2.5 A (Micro USB)	5 V DC @ 3A (USB Type -C or GPIO Power)
Cost	$25	$35	$35	$35, $55, $75

Table 29: Raspberry Pi Model Comparison

All of these options are viable to be used with the Amazon SDK's. The main constraint that must be evaluated is speed of Wi-Fi, power consumption, and CPU speed. Wi-Fi speed is crucial since the device must communicate with Alexa services that are available through the cloud. To ensure that a request from the user is handled quickly, the Wi-Fi needs to be reasonably fast. Moreover, it would be ideal to have 5 GHz Wi-Fi band since this band is better for apartments and other places were there could be interference from other nearby Wi-Fi networks. Next Power is also a big deciding factor as this must be accounted for along with any other device in the pet feeder. Finally, the CPU should be reasonably fast, although the only CPU among the options that is significantly faster is the Raspberry Pi 4's. One of the upsides is that among these, the GPIO configuration among these remains the same:

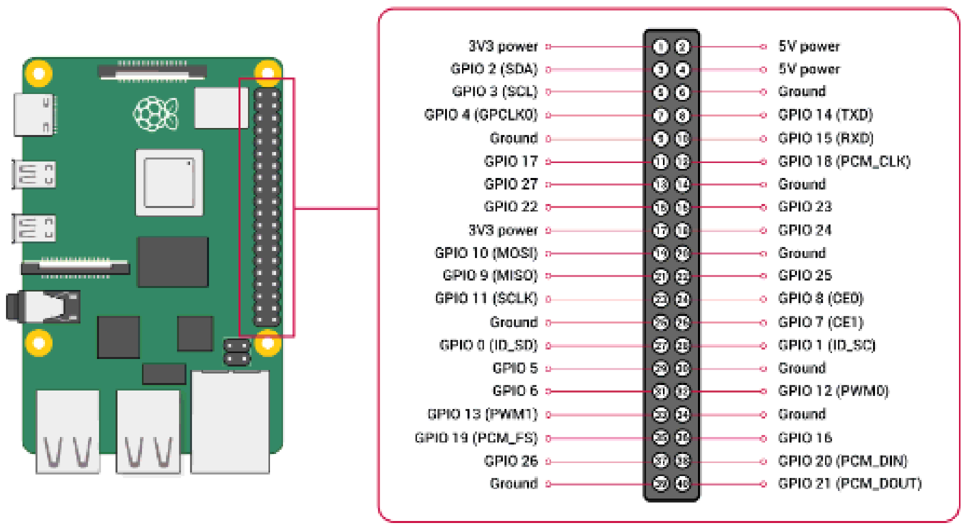


Figure 35: Raspberry Pi Pinout

3.8.5.3 Nvidia Jetson Nano

In this section, the implementation of an Alexa Skill with the Jetson Nano will be covered. Moreover, the different hardware options will be discussed. The deployment of Alexa onto a Jetson nano is very similar to the deployment of Alexa onto the Raspberry Pi. This is because at the end, they are both single board computers that operate very similarly. However, a different implementation can be followed. With the Jetson, Alexa will be integrated through the Visual Studio extension available from Amazon. This can be downloaded directly through the Visual Studio Application on the Jetson nano. It is called the Alexa Skills Toolkit. After installing the toolkit, all the Alexa service functions are available to create the Skill.

Next there are a couple of possible options to explore with the Jetson Nano. The Options are:

Component	Jetson Nano 2GB Developer Kit	Jetson Nano Developer Kit	Jetson Xavier NX Developer Kit
CPU	Quad-Core ARM A57 1.43 GHz	Quad-Core ARM A57 1.43 GHz	6-Core NVIDIA Carmel ARM v8.2 64-bit 6 MB L2 + 4 MB L3
GPU	128 Core NVIDIA Maxwell	128 Core NVIDIA Maxwell	NVIDIA Volta 384 Cuda Core and 48 Tensor Cores
RAM	2GB LPDDR4 RAM	4GB 64 Bit LPDDR4 RAM	8GB 128-bit LPDDR4x
Wi-Fi	802.11ac Wireless Adapter with Extension Cable	No Wi-Fi Out of the Box, can be added with M.2 Key E Wi-Fi card.	M.2 Key-E 802.11ac RTL8211 Chip
Bluetooth	No Bluetooth out of the box, can be added with Bluetooth USB Adapter	No Bluetooth out of the box, can be added with M.2 Key E Wi-Fi/Bluetooth Card	M.2 Key-E Bluetooth 5.0 RTL8211 Chip
GPIO	40 Pin	40 Pin	40 Pin
Power	5 V DC @ 3A (USB Type C)	5 V DC @ 2.0A (Micro-USB)	19 V DC
Cost	$60	$90	$400

Table 30: Jetson Nano Model Comparison

As can be seen above there are just a few options for the Jetson Nano. Compared to the Raspberry Pi, the options are much more expensive. Considering the budget, the options would most likely be limited to either the Jetson Nano 2GB or 4GB as the Xavier model is out of the project price range. The main advantage that the Jetson brings over the Raspberry Pi is the GPU. The Jetson boasts a very powerful (for its size) maxwell architecture GPU that can be great for some of the Computer Vision features that the pet feeder will include (these will be discussed later). Here the 2GB model is the most attractive for the project since it has enough ram for the pet feeder application. Even though it does consume more power, a 5W mode can be activated to consume much less power. Finally, the 2GB model also includes a Wi-Fi adapter, which is crucial for Alexa Skill deployment. Below is the GPIO configuration, which will be needed to communicate with the other microcontroller via UART:

SoC GPIO	Linux GPIO #	Alternate Function	Default Function			Default Function	Alternate Function	Linux GPIO #	SoC GPIO
			3.3 VDC	1	2	5 VDC			
PJ.03	75	GPIO	I2C1_SDA	3	4	5 VDC			
PJ.02	74	GPIO	I2C1_SCL	5	6	GND			
PBB.00	216	AUD_CLK	GPIO	7	8	UART1_TXD	GPIO	48	PG.00
			GND	9	10	UART1_RXD	GPIO	49	PG.01
PG.02	50	UART1_RTS	GPIO	11	12	GPIO	I2S0_SCLK	79	PJ.07
PB.06	14	SPI1_SCK	GPIO	13	14	GND			
PY.02	194		GPIO	15	16	GPIO	SPI1_CS1	232	PDD.00
			3.3 VDC	17	18	GPIO	SPI1_CS0	15	PB.07
PC.00	16	SPI0_MOSI	GPIO	19	20	GND			
PC.01	17	SPI0_MISO	GPIO	21	22	GPIO	SPI1_MISO	13	PB.05
PC.02	18	SPI0_SCK	GPIO	23	24	GPIO	SPI0_CS0	19	PC.03
			GND	25	26	GPIO	SPI0_CS1	20	PC.04
PB.05	13	GPIO	I2C0_SDA	27	28	I2C0_CLK	GPIO	18	PC.02
PS.05	149	CAM_MCLK	GPIO	29	30	GND			
PZ.00	200	CAM_MCLK	GPIO	31	32	GPIO	PWM	168	PV.00
PE.06	38	PWM	GPIO	33	34	GND			
PJ.04	76	I2S0_FS	GPIO	35	36	GPIO	UART1_CTS	51	PG.03
PB.04	12	SPI1_MOSI	GPIO	37	38	GPIO	I2S0_DIN	77	PJ.05
			GND	39	40	GPIO	I2S0_DOUT	78	PJ.06

Figure 36: Jetson Nano Pinout

3.8.5.4 Alexa Skill Hardware Decision

Among the options for operating the Alexa Skill, the group has decided to use the NodeMCU. This is due to a couple of reasons. Firstly, the NodeMCU operates at very low power. This will allow the Pet Feeder to remain mostly battery operated. Using a device like the Jetson Nano or the Raspberry Pi would have resulted in a pretty large spike in power consumption. Moreover, the NodeMCU is the cheapest among the three. A single device is under $5, which is prefect for the budget allocated for the project. Finally, the NodeMCU is easier to deploy since it has built-in Wi-Fi and is designed for IoT applications.

3.9 Computer Vision

In this section, the Computer Vision implementation to the Pet Feeder will be discussed. The idea here is to allow the user to interact on different levels with their pet. The Computer Vision capabilities could allow the user to set the pet feeder to automatically dispense food when the pet is detected. Alternatively, a window can be selected when the pet feeder will dispense food if the dog is detected. Moreover, the pet feeder could also stream through the Alexa skill the camera's feed if the user decides to see what the pet is up to when they are not home.

The hardware choice, when it comes to computer vision, is arguably one of the most important choices to execute computer vision smoothly and correctly. Next, the hardware that will be involved in the computer vision aspect of the pet feeder will be discussed. There are two main hardware components that will be covered, the Camera and the Jetson Nano (which will be the processor behind everything computer vision related in the Pet Feeder).

3.9.1 Jetson Nano

Choosing which device would drive the computer vision portion of the Pet Feeder was an easy decision. In the market, there are currently very few products that offer a low-cost entry into computer vision. One of the best and cost-effective products is the jetson nano. As discussed previously the Jetson nano can also serve as an Alexa connected device too. Therefore, the Jetson Nano can provide both Alexa and computer functionality decreasing energy consumption and reducing cost.

Figure 37: Jetson Nano

As discussed earlier, the main reason that the Jetson Nano is such a great device for Computer Vision applications is because it boasts a very powerful GPU. An NVIDIA Maxwell 128 CUDA Core GPU (as detailed in 3.7.5.3) brings a lot of graphical power to the table. Not only that, but it also includes the Hello AI package preinstalled. This package has a lot of pretrained models from cars to airplanes as well as dogs. This pre-trained model can be very easily deployed and put to use. With a well-trained model like the Jetson's any dog can be easily identified and provide the Computer Vision functionality that will be discussed in more detailed later. These models for computer vision work by scanning through each frame of the video pixel by pixel and attempting to guess the probability that a certain object or thing exists in the frame. These comparisons and scanning are aided by the powerful GPU because the backbone of these calculations consists of matrixes. Graphical processing units are specifically designed with matrixes in mind since frames are stored and processed in matrixes. This allows the processing to go smoothly and provides accurate computer vision detection. The Hello AI package is available on Nvidia's Git Hub and it uses the detect net interface for object detection. The docker container can be cloned from the git hub and then executed by using a simple python script. To conclude, the Jetson Nano is hands down the best alternative for Computer Vision and will provide a low cost and low power entry into adding computer vision to the pet feeder. We didn't end up using the Jetson Nano because our stretch goal of computer vision was not implemented into our project.

3.9.2 Camera

To implement Computer Vision, a camera is a crucial component. There are many things to be considered when picking a camera. On one hand it needs to have a good resolution, but it must also have low power consumption. In this section these will be discussed.
The Jetson Nano allows the developers to use USB cameras or ribbon connector camera. These ribbon connectors are the same found in the Raspberry Pi, so the same camera modules can be used. To provide accurate computer vision detection of objects a high-resolution camera is crucial. If the feed of the camera is blurry then inaccurate detection can occur. Therefore, an HD camera is required, either 720p or 1080p should be ideal. If a camera with higher resolution is chosen, this would result in higher processing times (resulting in slower FPS) and higher power consumption. Below are some of the options that are available.

	Resolution	Frames/Sec	MegaPixels	Power	Cost
Logitech C270	720p -> 1280 x 720	30 fps	2.1 MP	5V @ 500 mA	$27.47

Table 31: Logitech C270 Specifications

Figure 38: Logitech C270

	Resolution	Frames/Sec	MegaPixels	Power	Cost
Arducam 5mp	720p or 1080p	30 fps 60fps / 480fps	5.0 MP	5V @ 200 mA	$9.99

Table 32: Arducam 5mp Specifications

Figure 39: Arducam 5mp

	Resolution	Frames/Sec	MegaPixels	Power	Cost
Aukey FHD Webcam	1080p	30 fps	2.0 MP	5V @ 500 mA	$29.99

Table 33: Aukey FHD Webcam Specifications

Figure 40: Aukey FHD Webcam

Among the options listed above, the Arducam appears to be the best option. The Arducam has the highest number of megapixels (5.0 MP) and consumes the least power. Moreover, it also offers different operating modes (1080p or 720p). The low power consumption of this camera is also a great feature as many users have claimed over-current and under-current problems when running high power devices from the Jetson nano. We didn't end up using the camera because our stretch goal of computer vision was not implemented into our project.

3.10 Serial Communication

So far there have been multiple components/processors discussed. However, something that has not been discussed is how these devices will communicate with each other. It is important that all these devices work in unison to provide a good experience to the user. These devices include the microcontroller driving the pet feeder, the Jetson Nano controlling computer vision, and the Amazon Alexa device. For these to communicate a communication interface must be used. In this section, the available interfaces will be discussed.

3.10.1 UART

Two UART compatible devices have the ability to communicate with each other with the use of the Rx and Tx ports. The Tx and Rx ports can be used to interface to a UART compatible microcontroller. The Tx port would connect to the Rx port of the microcontroller and the Tx port of the microcontroller would connect to the Rx port of the other device. Both devices must be set to the same baud rate so that they can communicate correctly. The baud rate is the speed at which data is transmitted. UART transmits data asynchronously, so there is no clock synchronization. When the two devices communicate, the sender creates a packet with a start bit, frame, parity bits, and stop bits. The data is contained within the data frame. So far, only the functionality of UART has been discussed. But next the actual programming portion must be discussed. Transmitting data through the UART interface of any device is fairly straight forward. After enabling UART communication on the respective device, a simple code can be written in C or Python to send data through UART. Now that the UART connection is established, talking to the microcontroller can be accomplished. When the user asks Alexa, a script can be activated to run that will tell the microcontroller that food must be dispensed for the dog.

3.10.2 I2C

Inter-integrated-circuits or I2C is a serial communication protocol that is similar to UART but are used for modules and sensors. It is a bidirectional two-wire synchronous serial bus that is used to transmit information between devices that are connected to the bus. I2C has two lines that it uses to transmit information, the serial clock line (SCL) and serial data line acceptance port (SDA). The serial clock line is in charge of synchronizing the transmission, while the SDA is a data line which contains the data being sent and received. For the transmission to work properly, the master has to address the slave before sending or receive any data. The master will have their own clock for synchronization so that it can transmit messages on the I2C bus. The data is transmitted when the clock is in the high point. If there are multiple clocks with low points, then the clock with the longest low will dictate when the clock goes high again and the first clock that finishes their high point will dictate when the SCL line goes low. One of the advantages in using I2C since it does not require that many pins for multiple devices connections. The disadvantage is that it’s not as quick as other method since it takes long for the pull-up resistors to enable the highs and lows.

3.10.3 SPI

Serial Peripheral Interface or SPI is a full duplex data transmission for serial communication protocol. SPI uses four ports for communication, MOSI, MISO, SCLK, NSS. The multi-slave requires each slave to have a separate enable signal which makes it more complicated to implement compared to the I2C protocol. The master will set the first data bit through the MOSI which the slave will set their bit in the MISO. The master will then drive the SCLK which makes the edge rise which means that both the master and the slave receive the bit from the MOSI and MISO lines. The master will then set the edge to low and then the master and slave will start to set up their next bit. One of the many advantages that this protocol has over I2C is that the master does not have to address the slave before trying to receive or send data over. It is also the fastest protocol when compared to UART and I2C. The one that generates the clock is called the controller and the other is called the peripheral. The microcontroller is usually the controller. Since UART uses TX and RX line, that means that they are asynchronous and that means that the data that is sent is not guaranteed. A lot of microcontroller has the ability to use SPI and can write your own program that can handle I/O lines for sending and receiving data. There is a SPI library from the microcontroller, but you have to use the SCK, COPI, and CIPO pins because those pins are hardwired. A disadvantage is that there might be more errors since there is no error checking unlike UART using a parity bit. Unlike I2C, SPI can only have one master but is allowed to have multiple slaves. This can cause more wires to be used since if there are multiple slaves then they will all be linked to one slave.

3.10.4 Logic Level Converter

If there are any problems in terms of compatibility in terms of voltage levels need for each component, then a Logic Level Converter will be needed. The group has decided that a 3.3V-5V Logic Level Converter would solve this problem. The Logic Level Converter allows for the communication for device that have different logic levels. The converter will safely step the voltage level from 3.3V to 5V and vice versa. This also works for other serial communications like SPI and I2C. The bi-directional logic converter is a one level shifting circuit on the PCB board that you can purchase. This process is then repeated four times so that it creates a four-level shifting channel. The bi-directional level shifting composes of a N-channel MOSFET and a couple pull-up resistors. The figure below is the schematic of the bi-directional level shifting circuit.

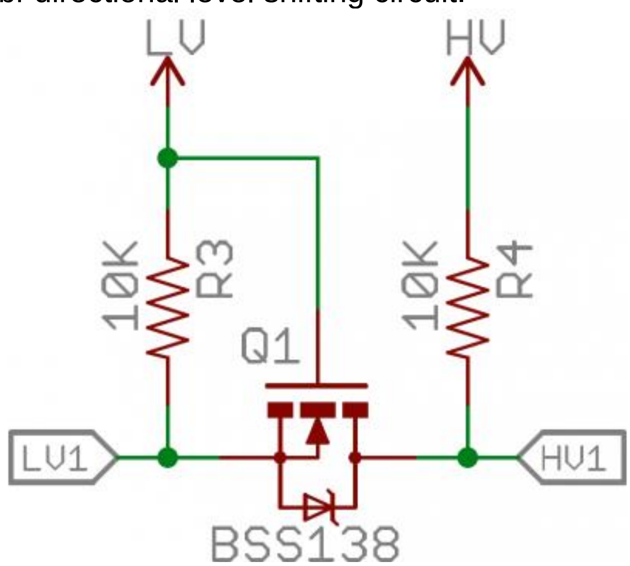


Figure 41: Logic Level Converter Circuit

As show in the figure below, the board consist of 12 pins that have 6 pins parallel to each other. One of the rows is for high voltage and the other row of 6 pins is for low voltage. The high voltage would be 5V and the low voltage would be 3.3V.

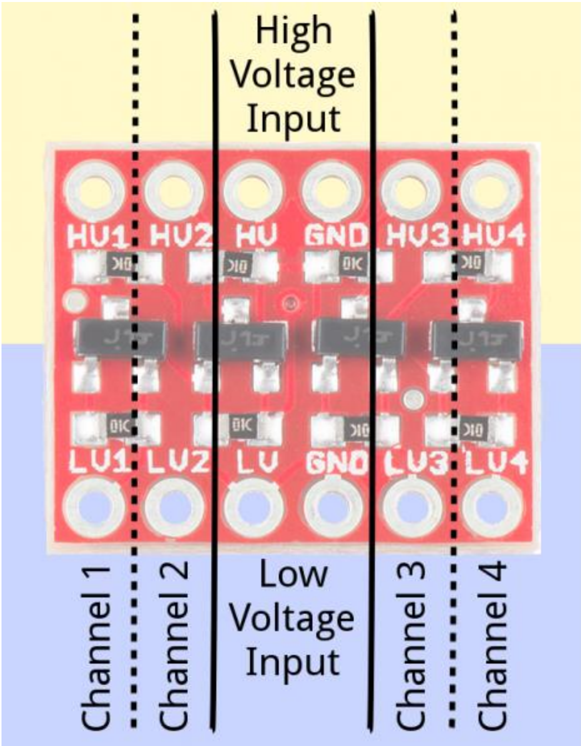


Figure 42: Logic Level Converter

HV stands for high voltage and LV means low voltage. If you are interfacing from 5V to 3.3V, the HV would be 5V and the 3.3V for LV. There are numbers that come after each HV and LV on the board and this represents that channel that you would be using. Make sure that you are consist with the channel you are using, if you are using HV1 then make sure that you are connecting to LV1 too.

3.10.4.1 Logic Level Converter for UART

Using a Logical Level Converter for UART communication does not use the Logical Level Convert to its fully potential but this does not mean that you should not use it. Make sure that the 3.3V is connected to the LV and the 5V is connected to the HV. Below is a figure that shows the wiring between an Electric Imp Breakout Board and an Arduino Uno.

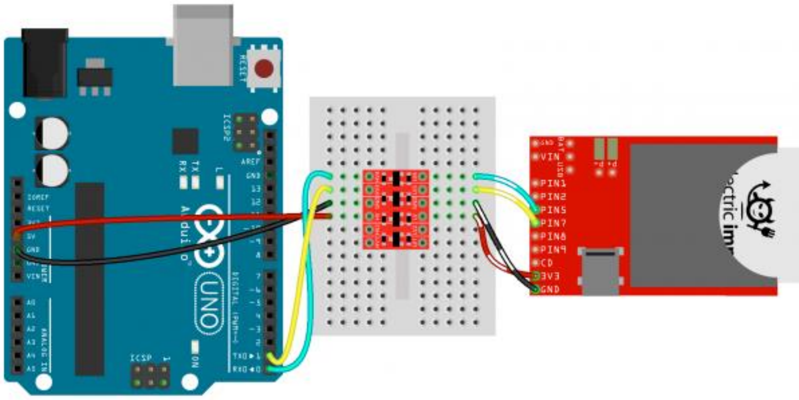


Figure 43:Logic Level Converter for UART Wiring Diagram

3.10.4.2 Logic Level Converter for SPI

Using a Logical Level Converter is perfect for SPI because there four required wires for this kind of communication. The figure below shows the required wire connection for SPI. The 5V from the Arduino board is connected to the high voltage (HV) pin from the logic level converter and the 3.3V is connected to the other column of the logic level converter, at the low voltage pin (LV). The table below will show what the pins are for the ADXL345 in the example below. The digital pins on the right side of the Arduino board are the SPI pins that are needed for the communication. D10 is the Slave Select (SS), D11 is Master Output Slave Input (MOSI), D12 is Master Input Slave Output (MISO), and D13 is the serial clock (SCK). In the example, the SDO is the Serial Data Output which is why is in the same channel as the MOSI pin, the SDA is the Serial Data Input for SPI and is connected to the MISO channel.

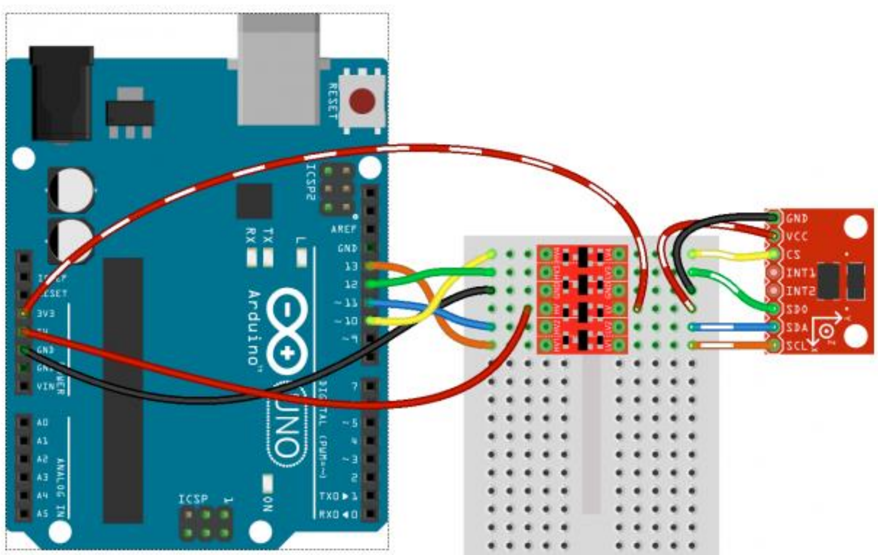


Figure 44:Logic Level Converter for SPI Wiring Diagram

GND	Pin to connect to ground
VCC	Supply voltage
CS	Chip Select
INT1	Interrupt 1 Output
INT2	Interrupt 2 Output
SDO	Serial Data Output (4 Wire SPI)
SDA/SDI/SDIO	Serial Data Input (4 Wire SPI) Serial Data Input/Output (3 Wire SPI)
SCL/SCLK	Serial Communications Clock

Table 34: SPI Connections

3.10.4.3 Logic Level Converter for I2C

When using a Logical Level Converter for I2C, make sure that the both the data and clock signals are bi-directional. You can add a second I2C interface if needed and they will be sharing the same level shifted SDA and SCL lines. The figure below shows an example of two I2C interfaces.

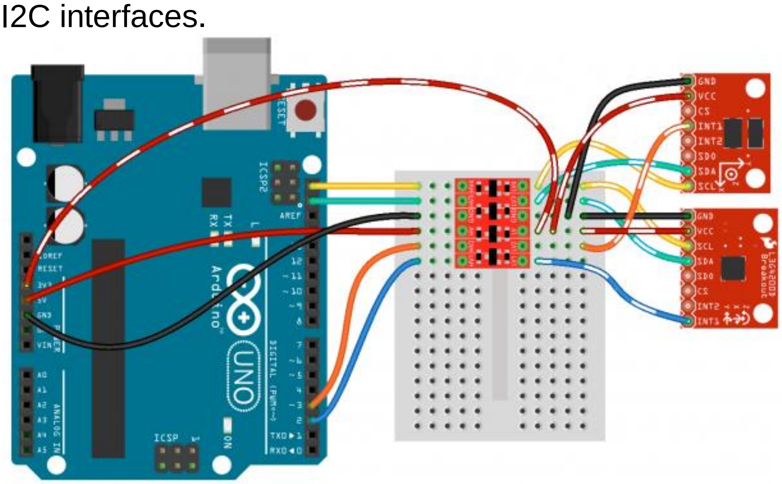


Figure 45:Logic Level Converter for I2C Wiring Diagram

3.11 Network

In this section the state of the Networking will be covered. Topics such as the OSI Model, Wi-Fi Standards, current technology, and future technology will be discussed. This topic is help understand how data will transferred over the network and how different Wi-Fi technology may affect the outcome of the design.

3.11.1 OSI Model

The Open System Interconnection Model or OSI Model for short is a framework for the different types of layers of networking. The model shows how different components of each layer work together to get everything to run together. This model was introduced in 1984 and is still used today. The OSI Model is comprised of 7 Layers, physical, data link, network, transport, session, presentation, and application layer.

The lowest layer is the physical layer, and this is where the connection between devices is established. The information here contain bits and are transferred from node to node. The bits are then converted into 0s and 1s and sent to the layer above called the data link. The data link layer is for transferring the data between nodes and there is error checking so that there isn’t information not being sent. There are two sub layers in the data link layer, the Logical Link Control (LLC) and Media Access Control (MAC). The next layer is the Network Layer also helps with the transmission of data and contains the IP address of the sender and receiver. This layer helps the data link layer find where the information is going and makes sure that the right connection is being made and that there is not a connection being made by accident. The Transport Layer helps with the application layer and uses the network layer. The layer makes sure that all the information is being sent correctly and that the message is not incomplete. The Session Layer is responsible for the making the actual connection between two endpoints. This layer makes sure that the right connection is being made and that there is not a connection being made by an unwanted visitor or someone that did not intend to make the connection. The layer also makes it so that once the connection is made then there should be no interruption in the connection. The Presentation Layer is where the raw data is converted so that the application can know what the data contains. This layer formats the data so that it can sent across the network and is also responsible for encrypting and decrypting the messages. The last layer is the Application Layer and is where the information is being displayed to the user. The Application Layer supports the interaction for the end users. The figure below shows all the different layers of the OSI Model and the roles of each layers.

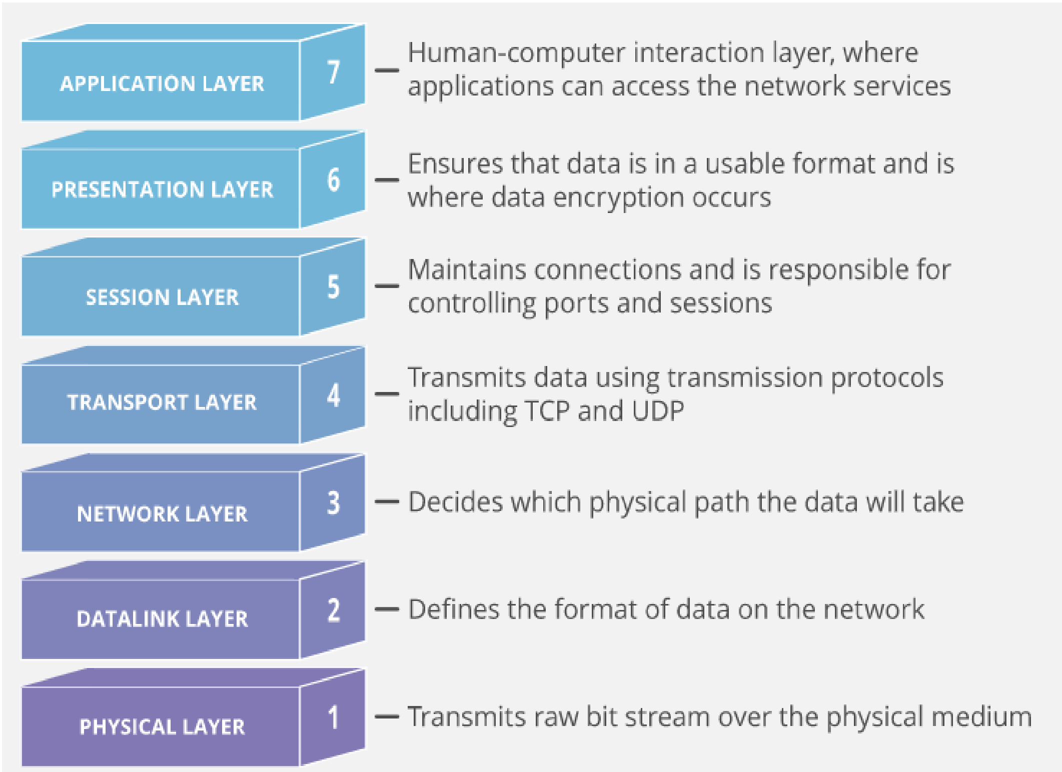


Figure 46: OSI Model Layer Diagram

3.11.2 Wi-Fi

In this section the state of the Art in Wi-Fi will be covered. Topics such as Wi-Fi Standards, current technology, and future technology will be discussed. Understanding this technology is crucial for the project as Amazon Alexa integration requires communication with Amazon servers. These Amazon servers provide the backbone support for Alexa.

3.11.2.1 Wi-Fi Standards

IEEE 802.11 is the standard of wireless LAN communications. The name behind this technology is what is known to us as Wi-Fi. This standard is overseen by the IEEE 802 LAN/MAN Standards Committee. Their main goal is the maintain the standards for networking and decide what is the best practices for local, metropolitan, and other area networks. There are several specifications in terms of the standards that are featured in the table below. The table below compares the different between the 802.11b and 801.11a/g.

Features	Wi-Fi (802.11b)	Wi-Fi (801.11a/g)
Primary Application	Wireless LAN	Wireless LAN
Frequency Band	2.5 GHz ISM	2.4 GHz ISM (g) 5 GHz U-NII (a)
Channel Bandwidth	25 MHz	20 MHz
Half/Full Duplex	Half	Half
Radio Technology	Direct Sequence Spread Spectrum	OFDM (64 channels)
Bandwidth	<= 0.44 bps/Hz	<= 2.7 bps/Hz
Modulation	QPSK	BSK, QQPSK, 16-QAM, 64-QAM
FEC		Convolutional Code
Encryption	Optional – RC4m (AES in 802.11i)	Optional – RC4 (AES in 802.11i)
Mesh	Vendor Proprietary	Vendor Proprietary
Access Protocol	CSMA/CA	CSMA/CA

Table 35: Wi-FI Standards Comparisons

3.11.2.2 Current Technology

The two main technology that is being use in today's devices are 2.4 GHz and 5 GHz. These two frequencies have different use cases because there are advantages and disadvantages from both. The 2.4 GHz has a larger effective range because of the lower frequencies which means it can penetrate obstacles like walls. All W-Fi enables devices will be able to use this lower frequency. This can also be a problem since that multiple devices will be on the same frequencies, then that means that there is going to be more interferences. While 5 GHz is faster in speed, the fact that it runs at a higher frequency means that it can go through wall as easily as the 2.4 GHz. This means that the access point has to be in a better location that the lower frequency one. If the user is not sure if the device is compatible with the higher frequency Wi-Fi, then the safer option is just to use the lower frequency.

Wi-Fi is an important part of the project since the product is going to be a smart device. The Wi-Fi will help with the communication between the device and Alexa, which is running off from the cloud. The pet feeder should be able to connect through the Alexa app through our phone. Since there is Wi-Fi on the pet feeder device, there might not be a need for a Bluetooth module since we can connect and use it directly from the Alexa app.

3.11.2.3 Future Technology

Wi-Fi Alliance announced IEEE 802.11ax on October 3, 2018, the common name for this technology is called Wi-Fi 6. On September 2020, IEEE includes Wi-Fi 6 as part of their specifications. Wi-Fi Alliance is also responsible for Wi-Fi 4 and 5. The pros of Wi-Fi 6 is that it offers four times the amount of throughput for each user in a crowded environment and is more efficient with power consumption, which can lead to devices lasting longer. Wi-Fi 6 uses technology like MU-MIMO and OFDMA that can help with multi-user and allows more connections to be able to connect simultaneously. The table below compares Wi-Fi 6 to the two previous Wi-Fis.

	802.11n (Wi-Fi 4)	**802.11ac (Wi-Fi 5)**	**802.11ax (Wi-Fi 6)**
Release Year	2009	2013	2019
Bands	2.4GHz / 5 GHz	5 GHz	2.4GHz / 5 GHz
Channel Bandwidth	20MHz, 40MHz (40MHz optional)	20MHz, 40MHz, 80MHz, 80+80MHz & 160MHz (40MHz support made mandatory)	20MHz/40MHz @ 2.4GHz, 80MHz, 80+80MHz & 160MHz @ 5GHz
Subcarrier Spacing	312.5kHz	312.5kHz	78.125 kHz
Highest Modulation	64-QAM	256-QAM	1024-QAM
Data Rates	Ranging from 54Mb/s to 600Mb/s (max of 4 spatial streams)	433Mb/s (80MHz, 1 spatial stream) 6933Mb/s (160MHz, 8 spatial stream)	600Mb/s (80MHz, 1 spatial stream) 9607.8Mb/s (160MHz, 8 spatial stream)
Channel Configuration	Single User MIMO & OFDM	Single User MIMO & OFDM Wave 1, Multi User MIMO & OFDM Wave 2	Multi User MIMO & OFDMA

Table 36: Future Technology Comparisons

3.12 PCB Creation Tool

This section of the document will go over different PCB creating tools that are available for students to have the ability to create schematics and PCB boards for the project. This will be very important for this project as we need to have an affordable, ideally free, software that we can use to help build our printed circuit board as that is one of the main requirements of the electrical engineers for this project. We will discuss different features of these different tools and compare price and what each of them offer and be able to decide on which one will be best for our group. At this end of this section, we will compare the price as well.

3.12.1 KiCAD

KiCAD is a software that is used to design printed circuit boards and therefore requires the user to download the software in order to use the program. KiCAD was originally developed by Jean-Pierre Charras and was initially released in 1992. It features an integrated environment for the building of the schematic along with the design of the printed circuit board.

With this software, the user is easily able to create a bill of materials or a BOM, Gerber files that can be sent to PCB manufactures so that they are easily able to create the user requested printed circuit board. KiCAD can also present 3D views of the PCB along with the components that go along with it. A very important aspect to this software is that it is free, and you are allowed to have an unlimited number of designs. One downside to this software over others is that it does not have a feature to simulate the circuit that you are building.

3.12.2 EagleCAD

Easily Applicable Graphical Layout Editor, or EAGLE, is an electronic design automation application and allows for schematic capture, printed circuit board layout, computer-aided manufacturing (CAM) and has an auto-router feature which allows the user to not have to route a PCB themselves. The auto-router feature is helpful for simple boards but not once you get into more complex designs.

Some pros of using Eagle are that provides a very large library that contains a large number of components and their footprints for the user to use for their printed circuit board. You must download the software, but it is a free software application. Another positive for Eagle is that our entire group has already used this software in Junior Design and thus are familiar with the software. However, using the free version of Eagle, the board being created can be no larger than 100mm x 80mm unless the team decides to buy the full version of Eagle.

A negative of Eagle is that it does not provide any simulation support into their software. This is a negative because we are not really able to test our circuit beforehand to make sure our design is correct. Verifying that our board would be working properly and as intended is very important before we go out and get it manufactured.

3.12.3 EasyEDA

EasyEDA is a web-based electronic design automation tool that allows all kinds of engineers to have the ability to design and simulate schematics, simulations, and printed circuit boards. EasyEDA also allows engineers to share their designs and boards, both publicly and privately with other engineers that use EasyEDA. Similar to Eagle, this tool includes features that allow for the creation of a bill of materials Gerber files and allows for the creation of files in multiple different file formats.

EasyEDA was originally released in August of 2013 as a web-based tool. Since it is a web-based tool, there is no need to download the software. A downside to this compared to other printed circuit board software's is that it requires an internet connection to be able to work on it. Another downside of EasyEDA is that nobody in our group has experience with using this tool but since there is many open-source projects and designs, our team would be able to have many examples at their side to help them develop their printed circuit board.

Now let us list some features of EasyEDA. First, it can import LTSpice schematics and symbols which provide a very easy way to port schematics to PCB layout and not having to redraw them. Next, once the Gerber files are generated from a complete PCB, they can submit their Gerber files directly to EasyEDA to have them manufactured. EasyEDA also has an auto-router feature, similar to Eagle, which the user must import the track size and where not to trace.

3.12.4 DipTrace

DipTrace is another software tool used to create PCB designs. This tool allows for a schematic capture, PCB layout, and the ability to create a library. Some features of DipTrace include high-speed and differential signal routing high-speed auto-router, and real-time 3D PCB preview. A big downside to this software is that it is not free like the other tools we have looked at throughout this section. DipTrace has 4 modules: schematic capture editor, PCB layout editor with built-in shape-based auto-router and 3D-preview, component editor, and pattern editor. This tool has the ability to convert the schematic into a PCB design and thus create a blank PCB board with the components so that we are easily able to place and route them. Lastly, another big downside to DipTrace compared to a tool like EasyEDA, is that it is lacking the ability to simulate our schematic design. Once again, it is one of the most important aspects when creating a PCB. Knowing if our design is going to work properly when we send it to board design and then a manufactured is very important because since we have a time constraint. If we were to create a board and then send to manufacture and receive back and then notice that our PCB is not working properly, we would be wasting a lot of time which is not ideal for any senior design project. The only way to avoid this if we were to choose DipTrace, would be that we would have to simulate using a different tool before creating a schematic in DipTrace.

To have the ability to compare these different tools, a table has been provided below more easily to that will compare price and advantages/disadvantages and then we will make a final call as to which tool our group is going to use.

PCB Tool	Advantage	Disadvantage	Price
KiCAD	Unlimited number of designs	No ability to auto-route	Free
EagleCAD	Familiarity with software	No ability to simulate schematic	Free for Students
EasyEDA	Web-Based Software	Requires internet connection	Free
DipTrace	Real-Time 3D PCB Preview	No ability to simulate schematic	$145

Table 37: PCB Tool Comparisons

3.12.5 PCB Tool Selection

After comparing the features, price, advantages & disadvantages we are now able to make a final decision to on which tool we are going to use. Our group is going to choose EasyEDA for the PCB schematic and board design. The free to use and ease of use for first time PCB designers, allowed for EasyEDA to be an easy choose. Maybe the most important and deciding feature was all of the open-source projects that are available and make it so there are plenty of examples available in case our group has questions and concerns about a design feature. The reason we choose EasyEDA over DipTrace was the big discrepancy in the price and compared to Eagle, we are not limited on a board size. Lastly, a big difference compared to KiCAD, the lack of a reliable auto-router allowed EasyEDA to be an easy chose over the software. In conclusion, our group will be using the EasyEDA web-based software for our PCB Design.

3.13 Food Dispensing

While the dispensing style does not seem very technical or in depth, it is crucial to the function of the system as a whole. This section will compare and contrast a few different dispensing variations that could be used to deliver food to a pet. The goal for the dispensing protocol is to be simple, reliable, and relatively easy to construct.

3.13.1 Auger Orientation

Through researching food dispensing products on the market as well as custom built, it has been determined there are two distinct ways the auger can operate. The first way is having the auger and motor in a horizontal position below the hopper, where the food is pushed sideways through a sleeve and to the pet's bowl. The Second way is to have the auger and motor assembly in a vertical position below the hopper, with the auger shaft passing through the hopper and mounting the motor at the top.

3.13.1.1 Horizontal Auger Orientation

Analyzing the horizontal auger style, it seems this is a common method of conveying a product, whether it is a dry food on an assembly line, feed for livestock, concrete/gravel in an industrial application, or in our case, pet food to our furry friends. This method has the advantage of transporting a product horizontally for however long the auger and sleeve is. Most feeders use a very short auger and sleeve, usually less than one foot long. The food drops into the open top of the tube on one end and is driven through the tube by the force of the auger being turned and dispensed out of an opening on the other end. Another advantage of this method is the ability to make it low profile underneath the hopper, in turn reducing the overall size of the product. This style of auger is also easy to build and set up. A major disadvantage that has come to light is the chance of food kernels getting stuck and jamming the auger where the intake hole ends, and the auger passes by. This is not a problem in industrial applications where powerful motors and crushable material is used because the machine will simply crush whatever jam happens and move on. This is, however, noteworthy for our application as motor strength is limited due limited power availability when using a battery.

3.13.1.2 Vertical Auger Orientation

The second style of auger positioning is to use a vertical sleeved auger that drives food straight down out of the hopper and into the pet's bowl. This auger style uses gravity to its advantage, which helps eliminate potential jams, and in the case of a jam, makes for easier clearing of said obstruction. This favors a less powerful motor because a well-designed vertical auger, along with some creative programming will require minimal driving power. A downside of this design is having to mount the motor at the top of the hopper. This could be inconvenient when designing a system that can be taken apart and cleaned easily. Another aspect to keep in mind with this style is height. By placing the auger in a vertical position, this adds height to the overall system.

3.13.1.3 Auger Orientation Comparison

Now that we understand the pros and cons of both auger styles, the ideal type needs to be determined. The horizontal style comes with the advantage of conserving height and the ease of application but has the drawback of potentially causing jams and making it difficult to design the system perfectly to avoid jams this way. The vertical style, in theory, should be more resistant to jams and easier to correct if obstructions occur. This style does however will add overall height to the unit. Upon deliberating as a group, we have determined we are willing to add height to our unit to reduce the risk of an obstruction occurring, which would be detrimental for a pet and should be avoided at all costs. The addition of a few inches of height to the unit is not a concern because this will give us more freedom for electronics placement in the lower portion of the unit. One factor that will need to be addressed when implementing the vertical auger style will be the stepper motor mounting. The solution for this will be discussed in the auger selection section and hopper design section.

3.13.2 Hopper Design

The hopper on the pet feeder is the space where food is stored and where the auger draws the food from to dispense. In order to meet our requirements and specifications for the unit, the hopper shall hold 3 days' worth of food or the equivalent of 6 total meals. We are using a measurement of one cup per meal, so the hopper shall hold a total of 6 cups of dry pet food. This is based on the upper end of pet meal sizes for large pets, therefore users with smaller pets will further benefit from an extended amount of food capacity.
With the hopper being the largest component of the system, it is the main factor to consider while designing the overall housing size. The volume of 6 cups of dry food is less than 1.5L, precisely 1,419.5 cm^3, less than a two-liter bottle of soda for comparison. This size leaves us with the ability to be flexible with hopper design and not have to worry about being pressed for space attempting to build a very large hopper.

Two general designs are under consideration for the hopper: cylindrical with a cone bottom or cubic with a pyramid bottom. Sketches of both designs can be seen below. The design of the hopper will mainly be determined based upon size, cost, and ease of construction. The appearance of the hopper is insignificant as it is an internal component. The hopper must have a lid to access and fill with food, as well as to clean the unit.

3.13.2.1 Cylindrical Hopper

The cylindrical hopper would be designed as a large vertical cylinder with a cone shape bottom that tapers into the auger sleeve. The body of the hopper could be made out of PVC pipe, a bottle, 3D printed, or a hybrid of each. PVC pipe would be simplistic but could be costly for the size needed. Below is a sketch of the cylindrical hopper design.

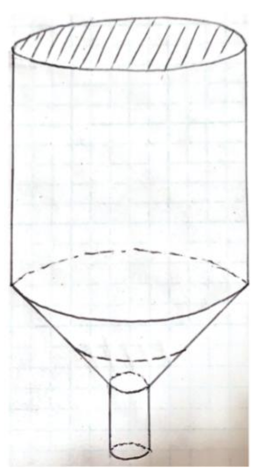


Figure 47: Cylindrical Hopper Design

3.13.2.2 Cubic Hopper

A cubic style hopper would occupy the majority of the upper portion of the unit, using four flat, vertical walls and four triangular slats at the bottom to form a pyramid shape that tapers into the auger tube. The materials used for this would be either plywood or plastic due to the need for ease of cutting and securing. This style of hopper could be easily modified and fine-tuned to fit precisely as needed. Below is a sketch of the cubic hopper design.

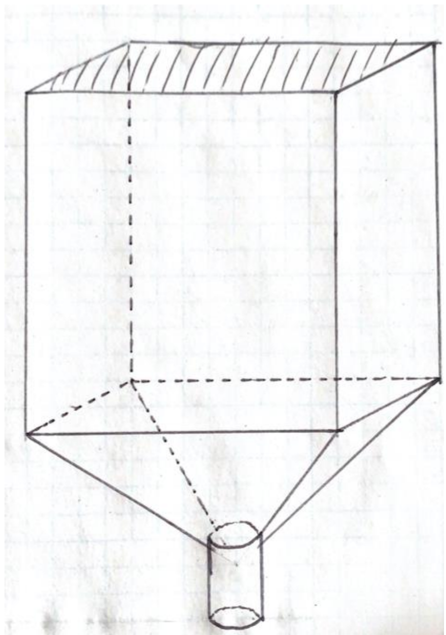


Figure 48: Cubic Hopper Design

3.13.2.3 Hopper Selection

The choice between the two hopper styles discussed above come down to material price and availability. Both designs are capable of meeting our requirements and will perform to our expectations. From doing some online research along with looking at the hardware store, PVC pipe in the desired size is more expensive than the cost of plywood needed. The cubic hopper will also be easier to mount because it can be attached directly to the side walls of the unit. For those reasons, the cubic hopper will be used unless a cheaper alternative presents itself for the cylindrical hopper option. Regardless of which design is used in the final design, it will not affect the layout of the other components such as the motor mounting, auger, and auger sleeve. These dimensions will remain the same, only the shape of the hopper around them and the lid would change.

3.13.2.4 Final Hopper Design

When it came time to start constructing the body of the pet feeder, it became apparent that it was going to be a significant mechanical undertaking to properly build this feeder. While the designed hopper and structure would most likely have worked well, our group simply did not have the time nor available resources to build the mechanical portion of the pet feeder. The solution for our final product was a pre-constructed pet feeder that was ordered online. This pet feeder, when stripped of all its existing electronics, worked well for us to retrofit with our exclusive electronics design. The hopper and “auger” assembly in this prefabricated pet feeder worked well together and allowed us to focus on our electronics and not have to worry about mechanical difficulties.

3.13.3 Auger Selection

Since the vertical auger orientation will be used, a long auger shaft is needed. The desired dimensions for the auger sleeve are 2” diameter by 4-6” tall. The diameter needs to be no larger than 2” but can be slightly smaller. The height of the auger blade itself is not as crucial but should have enough height to properly drive the food downward. An auger with a long shaft is desired and should be purchased with intentions to cut the shaft to size for an exact fit in the unit. When looking for options on the market, it became apparent that a garden tool used for auguring holes in the ground to plant flowers would fit our application well. These tools are sold in a variety of sizes and most often times come with a hexagonal shaft made to fit in a drill. This would work great, and a simple 3D printed adapter could be made to attach the shaft to the driver motor. The auger chosen is from Stageya, it’s dimensions are 1.97” (D) x 17.7” (L). The diameter will work perfect inside of a 2” tube, leaving a small margin to allow the auger to rotate freely but not too much where food will slide passed. Below is an image of the variety of augers that were chosen from.

Figure 49: Example of a few Augers

3.13.3.1 Final Food Dispensing Mechanism

While some prototyping was completed with the auger discussed in the previous section, our group did not end up using that design. As aforementioned in section 3.13.2.4, we opted to use a prefabricated pet feeder housing, retrofitted with our electronics. The Nema 17 stepper motor adapted well to our new pet feeder housing and was able to spin the dispensing mechanism. We were still able to fully control the output of food effectively using the stepper motor, very similar as originally planned with the auger. The final housing design that was used can be seen in the figure below.

Figure 50: Final Housing Design

3.14 Battery

To have a reliable, working food dispenser, the system needs sufficient power. The automated pet feeder is designed around convenience, therefore we decided to make it wireless and free of a tether to a wall outlet. This brings the challenge of supplying enough power to operate off of a battery for our required time of one week or 14 feedings on one battery cycle. This component is researched and discussed last because we need to account for all power draw from every component to select a proper battery. The battery should be able to supply the proper current to all components simultaneously running at maximum operation, while maintaining proper voltage to avoid overheating and damage of components.

The first step in the battery selection process is to calculate the power demands of the system as a whole. This means each maximum current demand needs to be summed together to get a current total. Then the duration of each dispensing cycle needs to be timed and multiplied by 14 to determine the amount of amp hours the battery must supply. Definition of amp hours: "Amp hour is the rating used to tell consumers how much amperage a battery can provide for exactly one hour."

The next step when selecting a battery is to find a reasonable sized battery that will fit properly in the unit. The battery should fit into one of the bottom sides of the unit to leave ample room on the other side for electronics and wiring. Once the power and size requirements of the battery have been met, the final constrains are price and the ability for safe indoor use.

3.14.1 Battery Power Requirements

As mentioned above, the standard measurement of battery capacity is amp hours (Ah). We know a 12-volt battery will be the most ideal for our use case in this project, therefore the amp hour rating on a 12-volt battery will tell how many amps can be produced for one hour at 12 volts. While not all components will operate at 12 V, this is a good overall voltage that has a wide variety of products available, and it can be stepped down to accommodate other electronics that operate at lower voltages.

The first step in determining battery power requirements is to get a total of the max current draw of all the components in the pet feeder. The reason the max voltage will be used is to ensure there will be no failures if all components happen to draw max current at the same time. Below, a table has been constructed by listing all components along with their operating voltage and maximum current consumption.

Component	Operating Voltage	Max Current
Ultrasonic Sensor	5V	15mA
Motion Sensor	5V	65uA
Weight Sensor	3.3-5V	1.5mA
Microcontroller	3.3-5V	1mA
Stepper Motor	12V	0.6A
Jetson Nano	5V	3000mA
Camera	5V	200mA
Node MCU (pending)	3.3V	200mA
Total	-	4017mA ≈ 4A

Table 38: Battery Power Requirements

By adding all of the maximum currents together in the figure above, it can be seen that the absolute maximum amount of current that could be drawn from the battery is about 4A. While this seems like a lot of draw, this is not the realistic amount that will be seen on an average food dispensing routine, it is only the theoretical maximum. A more realistic current draw is most likely well under 3 amps at the systems full operation. This can be confidently assumed because the Jetson Nano has a low power feature that allows it to run at 2 amps instead of 3, along with the fact that the other components will most likely not run at their max current draw. Regardless, the 4-amp maximum draw will be considered to allow for an ample power barrier to avoid any issues.

Now that we understand how much the maximum current consumption of the system will be, the run time duration can be calculated. As a group, we estimate that each feeding procedure will take approximately 2 minutes to complete. This feeding procedure will go as follows: external motion sensor senses movement and wakes up the Jetson Nano from low power sleep state, Jetson Nano will use computer vision to determine if it is a pet or other, if it is pet, then the feed command will be sent to the microcontroller which will initiate the stepper motor to deliver food. The stepper motor will continue to turn until the weight sensor determines the preset food weight has been met. When the food has been dispensed and the process is complete, the system as a whole will return to low power rest state, leaving only what is necessary on the Jetson Nano running to wait for any Alexa communication or sensor input. The process of waking up the Jetson Nano and performing the computer vision check is expected to take less than 30 seconds. The rest of the duration is allotted for the stepper motor to deliver the food, which we estimate will take less than 1 minute but will allow for up to 90 seconds. The quicker the better for the dispensing process because the faster the food is dispensed; the less power is used. Also, we would not want our furry friends to eat too much food while it is still being dispensed!

With an estimated 2-minute run time at full power, we can estimate how much battery life is needed. As stated previously, the unit's battery life is expected to last for one week, or 14 feeds before charging. To calculate amp hours, we must first understand what exactly an amp hour is in a mathematical sense. An amp hour is a function of current (I) multiplied by time (t), which just so happens to equal charge (Q). The equation below summarizes this.

Amp hours

$$Ah = Q = It$$

Using the equation above, we can now estimate the size of battery, in Ah, needed. The calculations go as follows:

$$Q = 4A * 2mins * 14cycles$$

$$Q = (112\ A*mins)/60mins$$

$$Q = Ah = 1.867Ah$$

These calculations conclude that to get 14 feed cycles with a duration of 2 minutes each, drawing 4 amps for the entire time of each cycle, we would need a 1.876 Ah rated battery. Now that we have this rating, we have some other factors to take into consideration. Although small, the resting power consumption between cycles cannot be ignored. The system will still draw power because a few functionalities need to remain active to search for Alexa communication and sensor input. It is safe to say these resting power consumptions will be minute compared to the active draw. Considering our overestimation of maximum current draw, along with extremely small resting consumption, we have decided that a 3Ah battery will provide a more than plentiful amount of power to run our system for at least one week.

3.14.2 Battery Types

While most batteries serve the same purpose, to store and deliver energy, there are multiple types and styles of batteries. Various styles of batteries are used for differing applications, from small electronics to large scale power storage for entire buildings. Output voltage levels change across the vast variety of batteries to suit specific power needs for electronics and the amount of stored energy plays a major role. There are two main categories of batteries: primary and secondary. Primary batteries are made for one time use cases such as AA batteries and are most commonly alkaline batteries. The chemical reaction that converts stored energy into useable energy inside alkaline batteries is non-reversable, therefore making them a onetime use, or primary battery. The other category of battery is the secondary battery. A secondary battery is the type that can be discharged, then recharged again. Specifically, secondary batteries have a chemical make-up that can be reversed from an external charging voltage to recharge and reuse the battery. The pet feeder will be using a secondary battery that can be recharged and used multiple times.

While we know we must use a secondary battery (i.e. rechargeable), there is still a large variety of secondary battery types to choose from. Secondary batteries come in all shapes and sizes; this is due to the chemical make-up of the batteries. Some battery use cases, such as cell phones or other mobile electronics, require compact, lightweight batteries. Other scenarios require large, heavy duty energy storage. These two example cases will use different batteries with varying chemical make-up. Energy density plays a key role in battery selection. Energy density is the ratio of energy per unit of mass. In small, portable electronics, energy density is extremely important because it is desirable to have as much energy stored as possible in the smallest size and weight that can be achieved. For our pet feeder, energy density is not extremely important because the pet feeder will for the most part be stationary, however, we do not have unlimited space, so it is important to select a battery that can perform to our specifications while not being extremely heavy or taking up too much space. Below, different styles of secondary batteries will be explored and then compared, to further narrow down the type of battery that will best suit the pet feeder.

3.14.2.1 Lithium-Ion

The first rechargeable battery we will discuss is the lithium-ion battery. This is one of the most commonly used rechargeable batteries. These batteries are used anywhere from cell phones to electric cars. Lithium-ion batteries are desirable because of their high energy density, meaning they are lightweight and store a large amount of power. Variants of lithium batteries are also offered that trade some energy density for safety depending on the use case. If a battery is being used somewhere where it may be dropped or damaged, it is critical to have a battery that will not combust or release harmful chemicals if damaged. Lithium-ion batteries come in a variety of shapes, sizes, and voltage levels. This is due to the ease of flexibility in manufacturing and leads to a great variety of selection for projects and applications. A downside to lithium-ion batteries is they tend to be on the more expensive side of the price point, comparatively. The figure below depicts how a lithium-ion battery is discharged and charged, showing how the chemical reaction is reversed to store energy after it has been depleted.

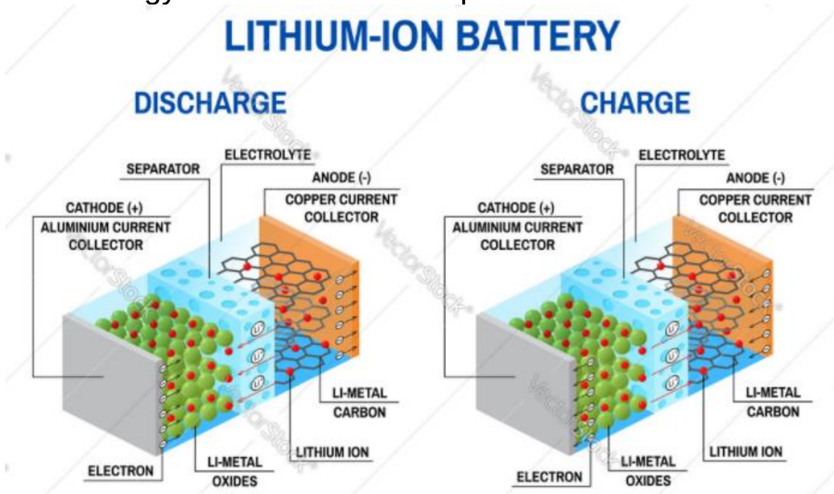


Figure 51: Lithium-Ion Battery

3.14.2.2 Nickel Cadmium

Nickel-Cadmium batteries are another style of rechargeable batteries. These batteries have the benefit of having the ability to maintain a constant voltage throughout their discharge along with holding a charge when not in use. This is beneficial for high discharge scenarios such as flashlights. A downside to these batteries is they do not always recharge back to 100% of their original potential, which makes them less effective over multiple discharges. They are also limited to smaller sizes, up to D battery size. This means they must be packaged together to be used for larger cases.

3.14.2.3 Nickel Metal Hydride

Nickel metal hybrid batteries are another nickel-based battery, but instead of using cadmium, a different alloy is used. These batteries, like the nickel cadmium batteries, benefit from high discharge properties but have the additional benefit of a higher energy density. Unlike the nickel cadmium batteries, these batteries do not suffer from the memory effect of not recharging back to full potential. These improved characteristics do come with the cost of a higher price point due to the materials used.

3.14.2.4 Lead-Acid Batteries

If you pay attention to your day-to-day activities, such as starting your car or using a trolling motor on your boat, you will quickly realize that lead-acid batteries are to thank for that. Lead-acid battery technology is among the older types of batteries. They are used mostly in workhorse environments, requiring large amounts of energy storage and high-power supply. These batteries are generally not used in portable applications due to their large size, weight, and low energy density. They are often used for powering electronics in combustion engine vehicles as well as for storage in solar power grids and back-up energy supplies. The generic lead-acid battery is relatively low in cost and reliable, making it ideal for creating large energy storage banks. Although new battery technologies have been developed and mainstreamed, lead-acid batteries remain as the go-to for most motor vehicles and continue to have an increasing demand.

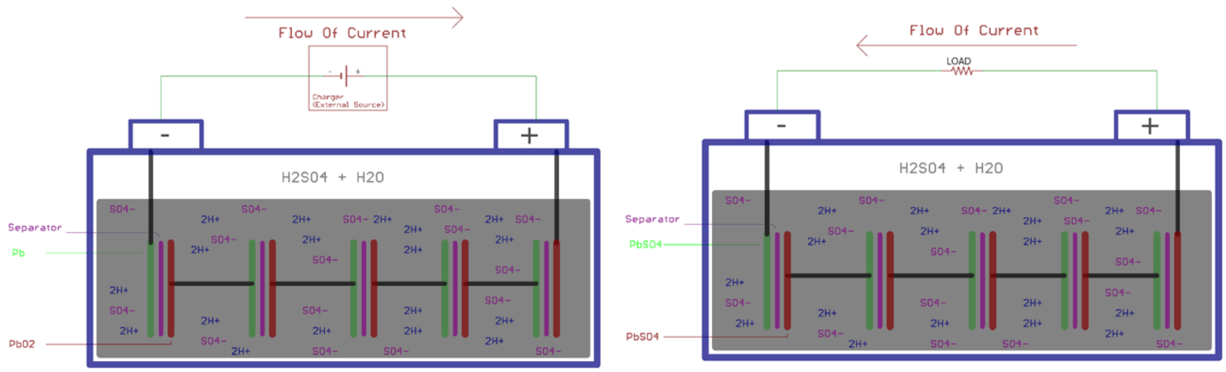


Figure 52: Lead-Acid Battery Charging and Discharging

3.14.2.5 Battery Type Comparison and Selection

All of the batteries discussed above serve to accomplish the same end goal of storing and producing energy, but we want the most suitable battery type for our unit. The nickel cadmium and nickel metal hydride batteries are a great choice to consider, but their cons outweigh their pros for our use case. The pet feeder needs sustained power for short intervals, repeating multiple times over the span of a week. The nickel-based batteries offer the discharge characteristics that are needed and have the advantage of holding a steady discharge voltage, but that is not of the upmost importance to our unit because all voltage going to components will be filtered through voltage regulators. The amount of energy that is able to be stored is more important, and it would be too costly to buy enough of these smaller style batteries and pack them together to get a large enough bank of energy.

Now to compare the lithium-ion battery to the lead-acid style battery. The lithium-ion batteries have a far greater energy density compared to lead-acid. This key for many portable applications, but it is not at the top of the list of important features for our unit, due to the fact that the unit will remain stationary for the majority of its use with the exception of the human user being able to relocate it around the pet's environment. The final decision comes down to cost and availability. It is apparent, through performing some market research, that a reliable lead-acid battery is significantly cheaper than a lithium-ion battery with similar ratings. The lead-acid battery is much larger in size and heavier in weight, but again, this is not a make-or-break factor in our consideration for the optimal power source.

3.14.3 Battery Safety

Now that the distinction has been made for a lead-acid battery to be used in the unit, the safety of using this type of battery must be thoroughly explored to ensure the safety of the users: both humans and pets. As the name suggests, lead-acid batteries are made of just that, lead and sulfuric acid. While these chemicals sound intimidating and harmful, a user should experience no issues when handling a lead-acid battery correctly. The main precaution that needs to be taken with this tyle of battery is during recharging. The discharging routine the battery goes through produces no external fumes or gases, but during the charging process, there is potential for gases to be produced. If a lead acid battery is charged too quickly with too many amperes pushed into it, or at too high of a voltage, the battery can become hot and produce hydrogen gas. From luminousindia.com, “Hydrogen is flammable and can become explosive if concentration of the gas in the air is equal or above 4%, which can be mostly achieved if batteries are charged in a sealed room or an area with poor ventilation.” Upon further investigation, the faster and hotter a battery is charged, the more hydrogen is produced. Larger batteries or battery banks will produce more hydrogen due to there being more chemical interaction as a whole.

Obviously preventing hydrogen buildup and avoiding explosions in our pet feeder is crucial for the health of the pets and human users. There is, however, good news. Variants of lead-acid batteries are made to prevent issues like acid being spilled and hydrogen gas being produced. There are batteries called sealed lead-acid (SLA) batteries and absorbent glass mat (AGM). SLA batteries stop or limit the venting that usually occurs with standard lead-acid batteries along with eliminating the risk or spilling acid if the battery is tipped over. AGM batteries are usually a type of SLA battery. These AGM style batteries have glass mats that absorb the electrolyte solution and hold it in place. The reason for this is to prevent the acidic solution from moving and spilling out of where it should be in the battery along with eliminating the need for “watering” the battery. “Watering” a battery is a task that is often necessary for standard lead acid batteries that are discharged and charged many times. The hydrogen and oxygen gasses produced lessens the amount of solution in the battery, therefore water must be added back into the battery to keep it at a safe, effective operating level.

Now that some available lead-acid battery technologies have been explored, we have enough information to make an educated selection on the battery that will be used in the pet feeder unit. This selection will be discussed in the following section.

3.14.4 Battery Selection

In this section, 3 brands of batteries will be discussed then compared. There are numerous brands and types of SLA/AGM batteries on the market, so it is worthy to discuss a few candidates in order to select the ideal battery for our product. There are some constraints to keep in mind when choosing the battery. One constraint that was discussed in section 3.14.1 is the battery needs to have the output characteristics of 12V and no less than 3Ah. The battery can be rated t more than 3Ah, but our calculations show the battery must be at least 3Ah. Another constraint is the battery must fall inside of the dimensions of 5 inches tall, 4 inches wide, and 3 inches deep. This will ensure that there will be enough room in the battery compartment to connect and run wires to the rest of the unit. The following sections will each present and discuss a type of battery, then in section 3.14.4 the 3 batteries will be compared, and a selection will be made based upon the balance of cost and performance.

3.14.4.1 Mighty Max 12V 3Ah Model YTX4L-BS

This 12V 3Ah battery from Mighty Max is the first battery under consideration to power the Automated Pet Feeder. This battery is a sealed lead acid, absorbent glass mat style battery, which meets our pre-discussed safety requirements. The terminals on this battery are square style with mounting holes with hardware included. Having multiple holes for mounting on the terminals provides more versatility for mounting multiple connections to the positive and negative terminal if needed. The model YTX4L-BS is marketed as a replacement battery for motorcycles, scooters, and all-terrain vehicles to be used as a power source for a small, motorized vehicle's electronic ignition system. This style of battery is designed to operate under harsh conditions, deliver power quickly as well as efficiently, and to last for many starting cycles without having to be charged. These features would lend well to the pet feeder and would ensure the user that they are using a safe product to feed their pet.

3.14.4.2 Neptune 12V 5Ah Model NT-1250

Analyzing the next viable option for the pet feeder, we have a 12V, 5Ah rechargeable battery from Neptune Power Products. This battery offers a sealed AGM design, which makes it a safe, maintenance free source of power. The model NT-1250 comes with tab-style "F1 Terminal" connectors, that have a pre-drilled mounting hole that will make it easy to secure wiring to. Surrounding the battery is a high strength, sealed outer casing to protect from being exposed to any of the chemicals inside of the shell.

3.14.4.3 Mighty Max 12V 5Ah Model ML5-12

This battery from mighty max is similar to the one discussed in section 3.14.4.1 but boasts a 5Ah rating instead of 3Ah. The model ML5-12 also has a smaller base footprint with less than an inch added to the overall height, giving it a lot more power inside a similar sized packaging. Following our safety requirements, this battery is an SLA/AGM construction that protects the user from fumed produced during the charging cycle along with exposure to the chemicals inside the casing. Advertisement for this product claims "Max Power for your scooter, alarm, lighting and other battery-operated needs".[49]

Figure 53: Might Max 12V 5Ah Battery

3.14.4.4 Battery Product Decision

All three of the battery candidates meet or exceed our pre-defined parameters needed for reliable and successful operation of the automated pet feeder. The decision for which battery to use in the final product comes down to how much power we can get for the most affordable price. The mighty max model YTX4L-BS has the least to offer out of this group in terms of stored energy and price point, being the least powerful and the most expensive. Choosing between the Neptune NT-1250 and the Mighty Max ML5-12 is a toss-up because of the almost identical features the two have to offer. The final decision comes down to price point, with the Mighty Max winning with a slightly lower price and more convenient shipping availability. This battery exceeds our calculated estimate of power consumption, to further eliminate any concern for the unit being able to maintain functionality for one week of operation.

Brand/Model	**Dimensions LxWxH**	**Amp hours**	**Price**
Mighty Max YTX4L-BS	4.40x2.75x3.42 in	3Ah	$19.99
Neptune NT-1250	3.54x2.76x4.02 in	5Ah	$16.99
Mighty Max ML5-12	3.54x2.76x4.21	5Ah	$15.88

Table 39: Battery Comparisons

3.14.5 Battery Charging

When designing a wireless, battery operated product, it is important to consider the interface between the battery and a wall outlet for recharging purposes. This section will discuss general charging protocols along with choosing the charger that will be of best fit for the automated pet feeder.

Charging a battery is not an overly complex process, but it is important to keep in mind a few key concepts, because if done incorrectly, serious consequences can occur. One important concept is to not charge a battery too quickly. Charging a battery too quickly or overcharging a battery can cause excess heat and produce large amounts of hydrogen, which as discussed earlier, can become explosive. A general rule of thumb for charging lead-acid batteries is to charge with a supplied current of less or equal to 20% of the amp hour rating of the battery. Since we are using a 5Ah battery, 20% of that is 1 amp. This means the charger we use should supply no more than 1 amp to the battery during charging. Another thing to avoid when charging a battery is to not overcharge. The battery charger used should have a current cut off that stops charging when the battery reaches 100% of its capacity. The current cut off also acts as a safety mechanism in case of a short or other unforeseen occurrence. This safety feature shuts off the charger and stops the flow of electricity from the wall to the device connected to the charger.

Another important concept to understand when charging a battery is the charging electrical potential needs to be higher than that of the battery. This means the voltage produced from the charger needs to be higher than the voltage of the battery. In our case, we need to supply a charging voltage higher than 12V to our battery to ensure proper and safe charging. The following section will explore a few different types of battery chargers and keepers. These chargers will then be compared to determine which one will best suit our pet feeder.

3.14.5.1 Mroinge 6V / 12V 1A Automatic Charger

This battery charger/maintainer from Mroinge provides the user with claimed “fully automatic” charging and maintaining for SLA and AGM batteries. A feature of this charger is its ability to switch between 6V and 12V charging outputs, all at 1 amp. This charger has spark proof connections, reverse polarity protection, and overcharging as well as over temperature protection. The charger will automatically switch from the regular charging mode to float mode when the battery reaches a full charge, protecting the battery from overcharging, limiting the amount of gasses/pressure produced, and extending the lifespan of the battery.

3.14.5.2 PeleusTech 12V 14.4V 1A Smart Charger

The smart charger/maintainer produced by PeleusTech is designed for SLA and AGM style lead-acid, 12V batteries. This charger has an allowable voltage of up to 14.8V, which is automatically adjusted to deliver the proper amount of voltage to the battery. The advantage this feature brings to the table is that it solves the electric potential problem of needing to apply a higher potential than the battery rating in order to charge it properly. The wall plug has LED charging indicators to let the user know if the unit is charging or if the charging cycle has been completed. The connectors at the cable ends are clip-style with rubber finger insulation for ease of use and safety. A key feature of this product is its “accurate positive voltage detection function” that prevents the device from charging when the battery is full.

3.14.5.3 Foval 12V 750mA Smart Battery Charger

The last battery charger under consideration is the automatic trickle battery charger from Foval. This charger claims to be a user-friendly option for charging various styles of lead-acid batteries, including SLA and AGM. Some features of this battery charger include reverse polarity protection, quick disconnect harness, clip connectors and ring connectors, infinite sequential monitoring, and spark proof battery charger sensing. This 750mA battery charger falls within the safe charging capacity of under 20%-amp rating, but still has the ability to provide sufficient current as to not charge too slowly. This charger will charge a 12V battery to its 12V holding potential and no more, reducing the risk of overcharging and overheating.

3.14.5.4 Battery Charger Comparison and Selection

Upon analyzing the three chargers discussed above, it is apparent that these chargers all share very similar overall characteristics and will all accomplish our desired goal of safely charging the battery. It is crucial that the most ideal products are used in this project to avoid any chance of malfunction or harm to the user, so the safest, best fit charger should be chosen. The chargers from Mroinge and PeleusTech are both self-adjusting, automatic chargers, but they lie right on the upper limit of calculated charging current. Charging speed is not a concern nor a goal to be met with a charger, so a slower, safer battery charger is more desired. For this reason, the Foval 750mA battery charger will be used. Along with the slightly lower charging current, the safety and smart charging features make this a desirable choice for our product. The quick disconnect is a nice feature to have also to make the unit less cumbersome to not have the entire charging unit attached at all times.

Figure 54: Automatic Battery Charger

3.15 Displays

Attaching a display to the Pet Feeder is one of the stretch goals for the project. A simple display can allow for better usability for the user and easier testing/debugging for the team Members. For these reasons, it would be ideal to include a small display in the final version of the Pet Feeder. In this section, possible displays will be covered as well as how these devices interface with the micro controller.

3.15.1 2-Line Display

The first option for the display is a simple 2-line display. This type of display is straight forward, it is a liquid crystal display (LCD) with 2 lines of characters. Each line contains 16-character spaces. Below is an image of the described LCD display:

Figure 55: Front of LCD Display

Figure 56: Back of LCD Display

Overall, this display is a good option as it offers the desired functionality and is inexpensive. It also offers a backlight, so that it can easily be viewed in dark places. One of the draw backs, however, is the number of pins used. This will be discussed in more detailed in section 3.15.3 below.

3.15.2 4-Line Display

The second and final option is a 4-line display. This display is very similar to the 2-line display (described in section 3.15.1). The only difference is that it is larger, with a total of 4 lines each capable of displaying 20 characters. Below is the LCD display:

Figure 57: 4-Line LCD Display

This display would bring a bit more functionality to the table, compared to the 2-line, since more information can be displayed at a time. The only drawback here is that it would consume a little more power and it is also more expensive. Below the way in which these displays interface to the host will be discussed.

3.15.3 Display Connection to Host

Interfacing the display to the host/microcontroller is just as important as actually picking the display. As can be seen in both figures of the two display options, these devices use a lot of pins to operate. This is the main constraint faced when adding these displays to the Pet Feeder. This is because, there is a limited number of GPI/O pins that can be used. This is one of the reasons that this is a stretch goal, since there is other, more important, components that must be considered before adding the display. Alternatively, there is a daughter board that can be bought along with the display that allows the display to run just off 2 pins. This would be the avenue that would be taken if this functionality is added. This daughter board (shown in Figure … below) takes advantage of I2C protocol instead of using Digital pins. With I2C, there are only 4 pins to be used, the 5 V power line, ground line, I2C clock signal (SCL), and I2C data line (SDA).

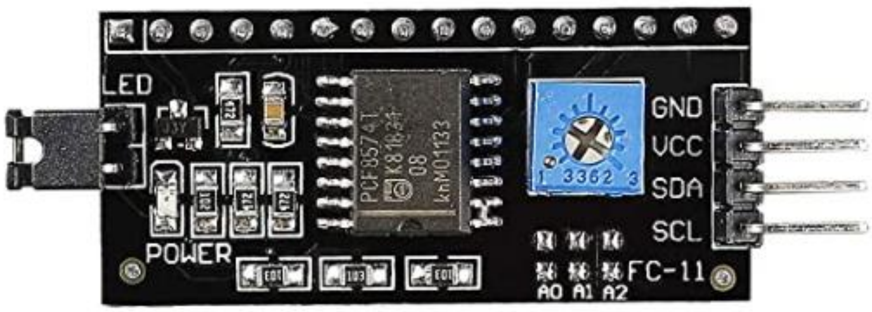


Figure 58: Display I2C Development Board

This daughter board also comes equipped with a potentiometer to easily adjust the brightness of the display. This board is compatible with both display options, since they both use the exact same pin layout, the libraries for operating them are just different. With these software libraries and I2C, deploying the board should be straight forward.

3.15.4 Display Decision

If this component is added to the Pet Feeder, a decision needs to be made regarding which of these two options will be chosen. Below is a comparison of both options:

	2-Line Display	**4 Line Display**
Dimensions	2 x Lines with 16 Characters Each	4 x Lines with 20 Characters Each
Power	5 V	5V
Backlight	Yes	Yes
Pins	4 (SCL, SDA, Power, GND)	4 (SCL, SDA, Power, GND)
Comms Interface	I2C	I2C
Cost	$8.99	$12.99

Table 40: 2-Line vs 4-Line Comparison

As can be seen in the table above, both of these devices are very similar. They essentially are identical except for the price and the size. For the project, the display that will be used is the smaller, 2 line one. This is due to the lower cost and presumed lower power consumption. Due to the smaller size, it is expected that this display will draw much less power than the larger, 4-line display. Moreover, the lower cost is ideal since the goal is to maintain costs as low as possible. All in all, the smaller display will provide the necessary/desired functionality at a lower cost and lower power consumption.

4.0 Standards and Constraints

To move forward with the project, Standards and Constraints that apply to the project must be discussed. Standards have the ability to facilitate development and manufacturing of the project, so it is important to have a clear understanding of them. Moreover, constraints are crucial, since they set the boundaries and limitations of the project, so that it is executed correctly. In this section these will be covered as to gain a better understanding of the scope of the project.

4.1 Coding Standards

Coding standards are imperative to create well developed applications and pieces of code. The main idea of coding standards to is to develop code that is well organized and properly layout. This allows all group members to easily edit, refine, and improve code along the way. In this project the main languages that will be used are Java and C. Some Python might also be used, but the standards for C and Java can apply to Python as well. In general, the idea is to enforce proper coding policies to allow for readable code (i.e., proper commenting, line breaking, and brace placement) this allows for maximum readability among group members.

4.2 USB Standards

In this section, the USB standards will be covered. The main standard to be covered is that of USB Type-C. It is imperative to understand how these components work to be able to integrate them into the pet feeder.

4.2.1 Type C USB

USB Type-C has recently become the standard for USB connected devices. The connector offers a lot of features and it is also used by the Jetson Nano for power-delivery. Here the USB Type-C connector will be discussed.

The Connector is made up by a 24-Pin receptacle, said connector is reversible/flip-able. Moreover, it has Standard USB4, USB 3.2, USB 2.0 legacy functionality, since adapters can be used to connect to these legacy USB devices through USB C. Moreover, with USB power delivery it can deliver up to 100 Watts of power to devices. The USB Type C connector takes advantage of a dual lane ping set-up. This dual lane setup allows for extremely fast speeds of up to 20 Gbps and even 40 Gbps. These are characterized below in Figure … as Lane 0 and Lane 1. Lane 1 is made up by the RX2+, RX2-, TX2+, and TX2-. While lane 0 is made up by TX1-, TX1+, RX1-, and RX1+.

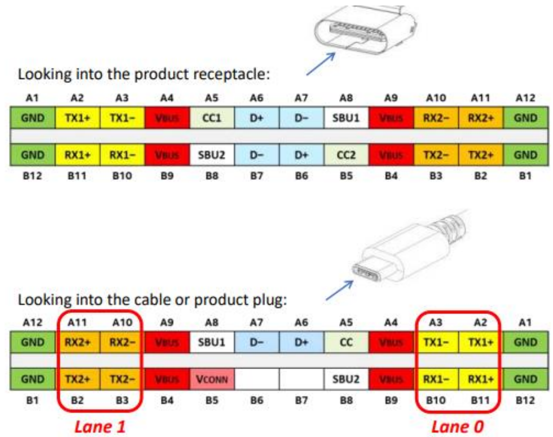


Figure 59: Type C USB Pinout

The connector has 2 sets of pins top and bottom. These two sets are identical but flipped. The reason for this is because type c has a reversible connector, so it does not matter which direction the connector is plugged in. This used to be an issue with older standards like Micro USB. Next the USB Type C Configuration Channel is used to detect when there is a connection, establish roles between two ports, configuration of Vbus, configuration of Vconn, establish the direction the connector was plugged in as, enter USB4 operation, and discover any other modes of operation. Essentially, this configuration channel establishes initial connectivity between the device and the host. Below, the connector system implementation model can be seen:

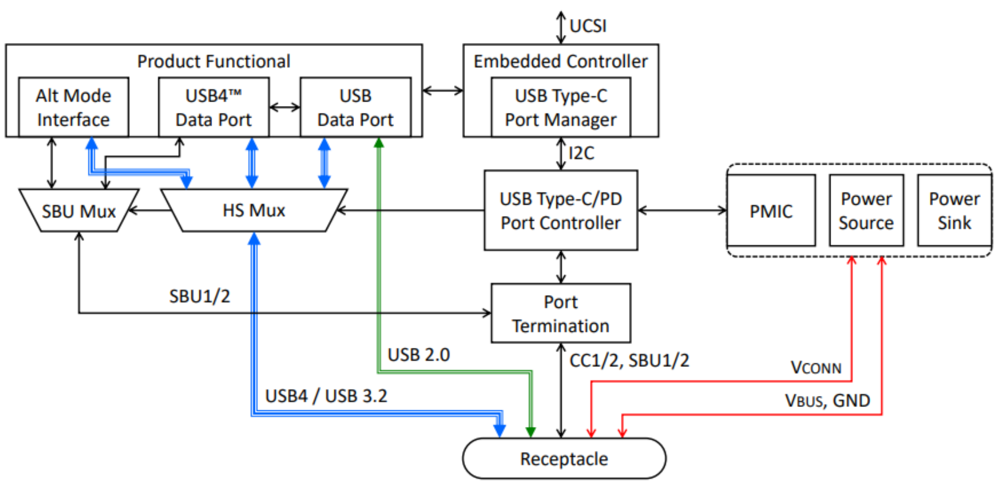


Figure 60: Type C USB Configuration

Understanding the Type C connector and its mode of operation is crucial since the idea is to power the jetson nano through USB Type C with our own power supply. Understanding this technology and its mode of operation allows for integrating the power supply for the jetson in the Pet Feeder.

4.3 Constraints

As like in any project, there are constraints placed on the team for numerous outside factors. It is important to understand these limitations in order to develop and complete the desired project. These constraints include buy or not limited to economic constraints, time constraints, and health constraints. These will be covered below.

4.3.1 Economic Constraints

One of the major economic constraints is that the funding will coming out of the pockets of the students since this is not a sponsored senior design project. These constraints may lead to items not being bought if the price is too high. The budget of this project would ideally be below $400, which would be split among 4 members. There are many items that would be amazing to have and would make the design much simpler. Certain members may have more expenses that they have to contribute to outside of this project than other members, so it is not fair for other members demanding more expensive parts if the other members do not want to pay for it. Since there is a price that we want to stay below, this has led to more research being done to see which components are needed. The group members would not like it if a component is not compatible with another certain part, which could result in ordering other parts and not being able to return the item that does not work in the project. One of the most important parts of the project is the PCB design so we have to make sure that the design will work with the components that we have. If the PCB does not work initially, we are going to have to redesign it and reorder which will cost us more money and limit the amount of time we have to do the necessary testing. Since the lab at UCF is limited to a certain capacity, readily going to the lab for quick testing is a major setback. We do not have the budget to invest in some of the equipment that would be helpful for testing the electronics.

4.3.2 Time Constraints

All the members right now have other classes they have to worry about besides Senior Design. This can lead to people falling behind in doing research and contributing to the team. It is important to plan out how you want to divide your time up and communicate with your team members on where you are in terms of finishing your assigned work. People who are falling behind should communicate this to their other team members that they are falling behind and should not tell them late minute because that could cause other problems. Scheduling a certain time each week is a must because other people might have plans so making a last-minute meeting time would not be ideal. If things change then let the people know immediately. We do not intend the initial PCB design to work as intended so we have to make sure we have enough time to keep testing and reordering. Since we will be taking Senior Design II in the summer, this may shorten our time that we have to complete this project since the summer term is usually shorter than Fall and Spring.

4.3.3 Health Constraints

One of the main constraints this year is COVID-19. With the widespread state of the COVID-19 pandemic, meeting in person has become extremely difficult among the group members of the project. Moreover, all meetings must be conducted remotely, this decreases the potential exposure to COVID-19 among the team. However, we hope that as COVID-19 vaccinations keep being distributed, the team members will not have to be concerned with this issue. With senior design 2 just around the corner, the in-person meetings will be a requirement since all the team members will need to work together on the implementation of the pet feeder. Here, the members will make sure to follow CDC guidelines for meeting with others. Moreover, mask wearing will be enforced in the meetings to reduce COVID-19 exposure. The team members will try their best to maintain social distancing during in person meetings as well. However, this will be more difficult as many components require assembling with two sets of hands, so team members must be extra careful.

5.0 Design and Development

The Alexa Pet Feeder will include a myriad of features. These being computer vision, motion sensing, precise food dispensing, and accurate readings for remaining food. To implements these features and put together a successful feeder, the design and development must be broken up, this process will be discussed here.

5.1 Software Design

The software in the pet feeder will bring all of the desirable functionality forward. For this reason, it encompasses a crucial part of the project. Here the overall software implementation of the project will be discussed, later some of the specifics involved in the project will be covered.

Altogether, the pet feeder will make use of the Jetson Nano (Stretch Goal), Node MCU, and Micro Controller (here other auxiliary devices are not included). Each of these playing their own role at bringing the Pet Feeder to function as desired. Below, in Figure …, the overall structure of how these devices will interact can be seen:

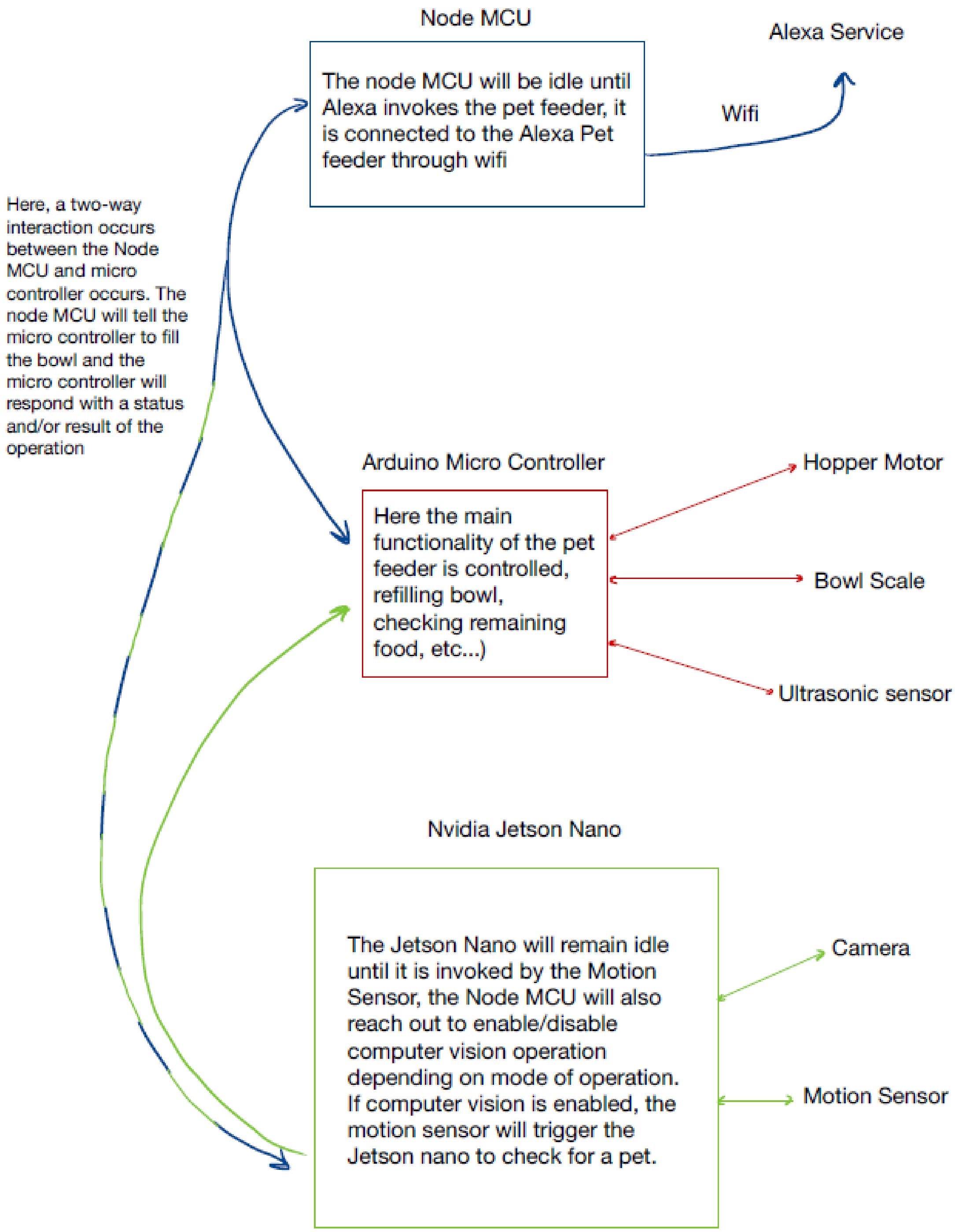


Figure 61: Software Flow Diagram

As can be seen above, there is a general overview as to how these devices will work in unison to bring forth the smart pet feeder functionality. Next the specifics of the operation of these devices will be discussed.

5.1.1 Micro Controller

The micro controller is truly the backbone of the core functionality in the Pet Feeder. This device, which will drive the design of the PCB controls the food dispensing mechanism of the pet feeder. For this reason, it must interact with both the Jetson Nano and the Node MCU. There a couple of different ways this device will interact with the other components, these will be discussed in the table below:

Input	Output
User requests Pet Feeder to Dispense Food	Node MCU communicates to the Micro Controller that Food must be Dispensed. First the Micro controller will ensure that there is food in the hopper, if there is the requested amount of food (depending on the dog chosen) will be dispensed. Amount of food left in the hopper will be checked again after dispensing is over. If food reserves are low, the micro controller will inform the node MCU so that it can inform Alexa.
Nvidia Jetson Detects Dog in front of Pet Feeder.	Jetson nano communicates to the Micro Controller that Food must be Dispensed. Similar procedure as the one outlined above will be followed.
User asks to see if there is any food in the bowl.	Here, the Node MCU will request the micro controller to report if there is food left in the pet bowl. The Micro Controller can check the built-in scale for food in the bowl and report back.
User requests food reserve status	Node MCU will request the micro controller to report if there is any food left in the dispenser. The micro controller will use the Ultra Sonic Sensor to check if the food in the dispenser is low.

Table 41: NodeMCU and MCU Interaction Comparison

The inputs listed above are the main ways that the micro controller will bring forth functionality to the pet feeder. These main scenarios provide a preliminary design to the code that the micro controller will be executing. The idea here is to have the micro controller on standby for most of the time. Here it will be waiting for a request to dispense food from either the Jetson Nano or the Node MCU. This interrupt will go off if information is received from the SPI ports. Once a signal is received the micro controller will wake up and decode the message. After understanding what the user requested, the micro controller will execute/reply to the requests. After completing the desired function, it will go back to sleep and wait for another interrupt. Below the code flow diagrams will be covered:

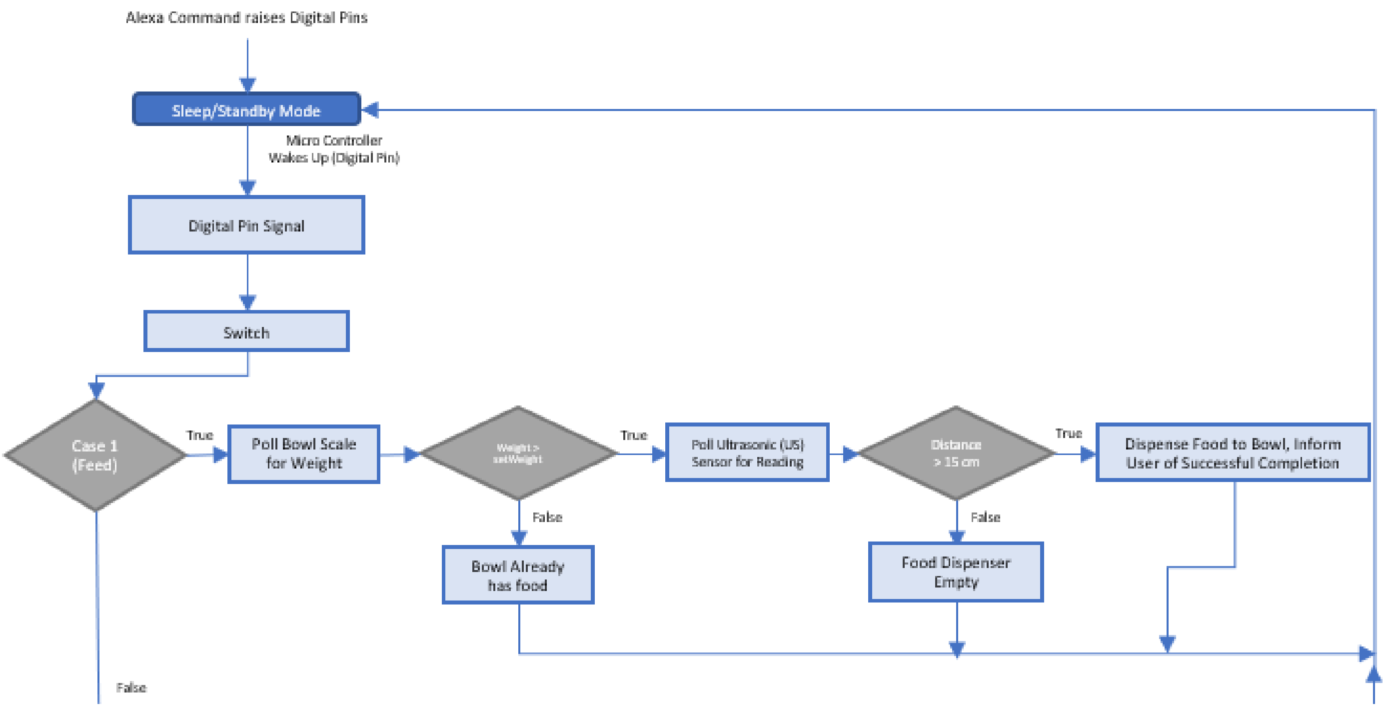


Figure 62: Bowl Scale Software Flow Diagram - Case 1

In Figure 61 above, the first portion of the flow diagram for the Micro Controller can be seen. As discussed, the microcontroller will be in sleep mode waiting for the wake signal from the Node MCU. The digital pin will signal which case will be called to the microcontroller. The first case is included in the Figure above. This mode is when the user requests to feed the pet. First the microcontroller will poll the scale in the bowl to check if there is food already present. The exact value for the weight of food threshold will be set by the user's desired value, but for now assume <1 is an empty bowl. If the bowl is full or contains some food, the micro controller will inform the node MCU that there is food in the bowl. Then the node MCU can inform the user through Alexa. If the bowl is empty, though, the ultrasonic sensor will be polled to check if there is food in the dispenser. If there is, the food will be dispensed. Once the food is dispensed, the micro controller will check if there is food left in the dispenser to inform the user if reserves are low. If there is not, then the micro controller will message back through digital pins to inform the user. Next comes Case 2, in which the user asks if there is food in the bowl:

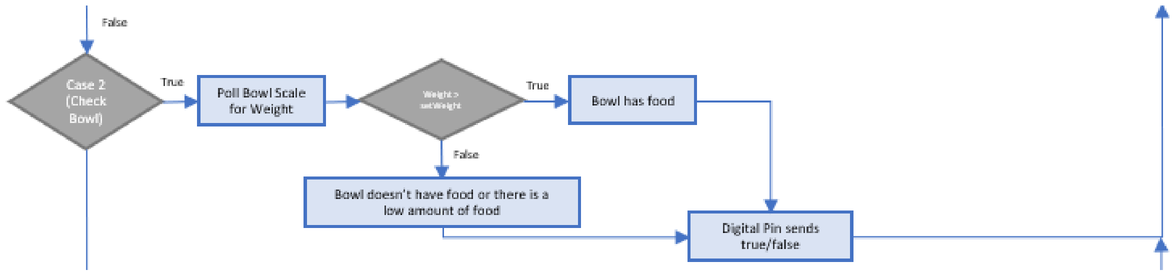


Figure 63: Bowl Scale Software Flow Diagram - Case 2

The case above shows what will happen when the user asks if there is food in the bowl. First the micro controller will poll the Bowl Scale for weight. If the weight is >1 (as discussed above the weight threshold will be user inputted) the bowl has food, otherwise the micro controller will inform the node MCU so that Alexa can tell the user. Next case 3 will be discussed, in which the user asks if there is food left in the dispenser:

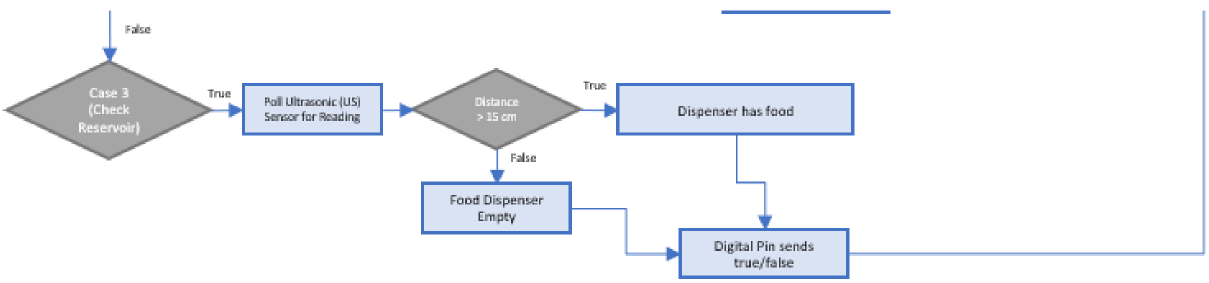


Figure 64: Bowl Scale Software Flow Diagram - Case 3

The case shown above shoes what will happen when the user asks to check if there is food in the dispenser. The micro controller will poll the Ultrasonic sensor for a reading. If the reading is less than 15cm then there is food, if not there is not. These findings of the micro controller will be reported back to the user through the node MCU. Finally, there is the default case, this occurs if there was a transmission error or for some reason, the mode of operation was sent incorrectly by the Node MCU. This will basically serve as a safety net for these kinds of errors. Below the default case can be seen:

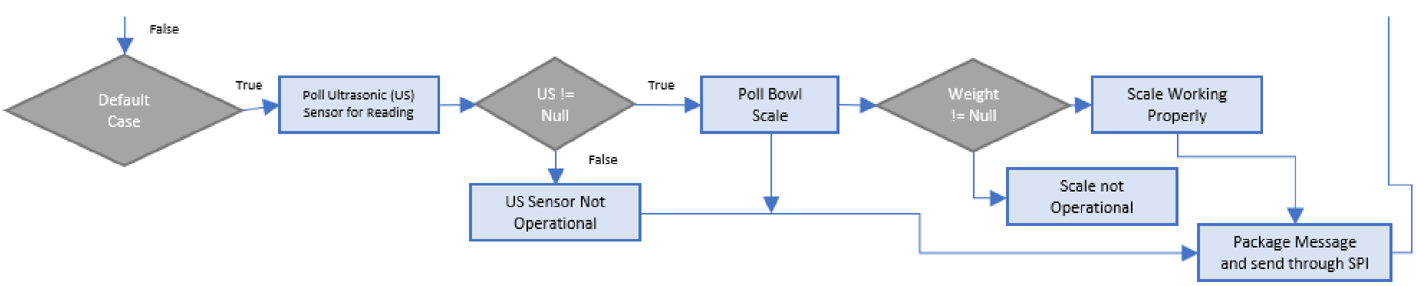


Figure 65:Bowl Scale Software Flow Diagram - Default Case

This case will, essentially, poll the devices to ensure that systems (on the micro controller side) are operationally and working correctly. The idea is that the micro controller will poll both the ultrasonic sensor and the scale to make sure it is able to acquire a reading. If it is great, then systems are working properly, if not then something is malfunctioning. The micro controller will inform the Node MCU of both cases after it is done. The pet feeder will most likely go long periods of time (couple of hours) without any interaction from the user. During these times, the Node MCU will be reaching out to the micro controller to execute this default case to ensure that systems are operational. This way if something is not working the user can know. Next the way that the sensors in the pet feeder will operate will be discussed. When designing the final product, we decided not to implement the default case.

5.1.1.1 Ultrasonic Sensor

The Ultrasonic sensor will reside towards the bottom portion of the hopper. This sensor will provide the micro controller with information on how much food is left in the hopper. The ultra-sonic sensor has the ability to detect the distance from it to the object it is pointing at. Moreover, it can do so precisely for small distances, as the dimensions of the hopper. Here the sensor will be polled to check the distance it has to other objects. If there is a food in the hopper the ultrasonic sensor will automatically read a number <1. This will occur because the sensor will be obstructed by the pet's food. However, if the distance is relatively equal to the width of the hopper, it can be inferred that the food is low. Below, in Figure … the desired placement of the ultrasonic sensor is shown:

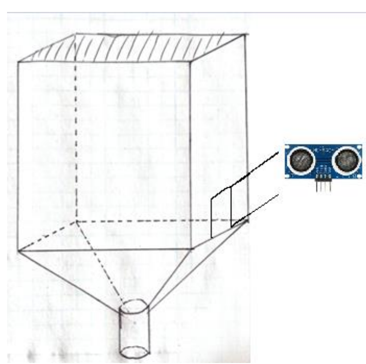


Figure 66: Ultrasonic Sensor Placement on Hopper

5.1.1.2 Scale

The Scale will reside under the Food Bowl. As describe in the Research section, the scale used utilizes a load cell. This scale will be used to measure that precise amount of food is dispensed for the dogs. Moreover, it also stops the bowl from being over filled in the case that the user accidentally requests the bowl to be filled when it already has food. The scale will be implemented into the Food Bowl as such:

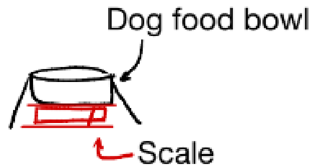


Figure 67: Pet Bowl with Scale Underneath

5.1.1.2 Hopper Motor

The hopper model will reside inside the hopper and will provide the dispensing functionality in conjunction with the Auger. This motor will be operated by the micro controller until the desired weight/feedback from the scale is received.

5.1.1.3 Arduino Uno and ATmega328p Development

We are planning on programming the ATmega328P using the Arduino Uno which is an open-source microcontroller board. The specific Arduino Uno board we are getting is the Arduino Uno R3 which already comes with the ATmega328P, you can remove and test it on a breadboard if it is necessary. The table below will display the specifications of the Arduino Uno board and how much power is necessary to power the board. The power consumption of this development is not very important to us since we are only using this method to program the microcontroller, so we are not using the whole development board as part of our project design.

Operating Voltage	5 V
Recommended Input Voltage	7-12 V
Limit of Input Voltage	6-20 V
Digital I/O Pins	14 (6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32KB
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Table 42: Arduino UNO Specifications

What makes the Arduino Uno so nice to program with is that Arduino have they own IDE that makes it easy to start. The Arduino IDE is a very easy beginner level environment so that potentially all the members in the group can help out with giving their inputs on what can done better. Uploading the code from the Arduino IDE to the microcontroller is as simple as plugging the USB cable to the Arduino UNO and then running the code. There are two microcontrollers on the Arduino board. The two are the ATmega328P-PU and the ATmega16U2-MU, these two different microcontrollers are different types of microcontroller and have a different purpose. The ATmega16U2 is a surface mount device (SMD), this cannot be removed unlike the ATmega328P which is a dual-in line package (DIP). The purpose of the ATmega16U2 is that it acts as the bridge between the development board and the PC that you are going to be uploading code from. While the ATmega328P does everything else on the board. The ATmega328P does not natively support a USB so that is why the other microcontroller was add so that the sole purpose is to provide the connectivity of the USB port.

If you are purchasing the Arduino Uno, then the bootloader also comes with the controller that is on the development board. If somehow the microcontroller is messed up during the process of programming, then you can just purchase the microcontroller for around three dollars. If you have the microcontroller as a standalone purchase, then the first thing you have to do is to burn the bootloader onto the microcontroller. A tool that you will need is a USB ISP AVR which helps flash the bootloader onto a microcontroller. The bootloader is written by the manufacturer that is in the memory and this is important because it allows the microcontroller to be connect to a PC so that it can be programmed via a serial port connection. The bootloader is like the BIOS or Basic Input/Output System for a PC. The figure below shows the processes of what the bootloader does when the device is being powered on or being reset. The bootloader will check if there is anything coming from the serial communication line which then checks for new software that is available to be downloaded. If there is something there waiting, then it will download it and store it in a certain location so that nothing is overwritten. If there isn't anything waiting at the serial communication line, then the microcontroller will run the code that was most recently loaded onto the chip. The bootloader will instruct the chip to keep running the code as long as there is power.

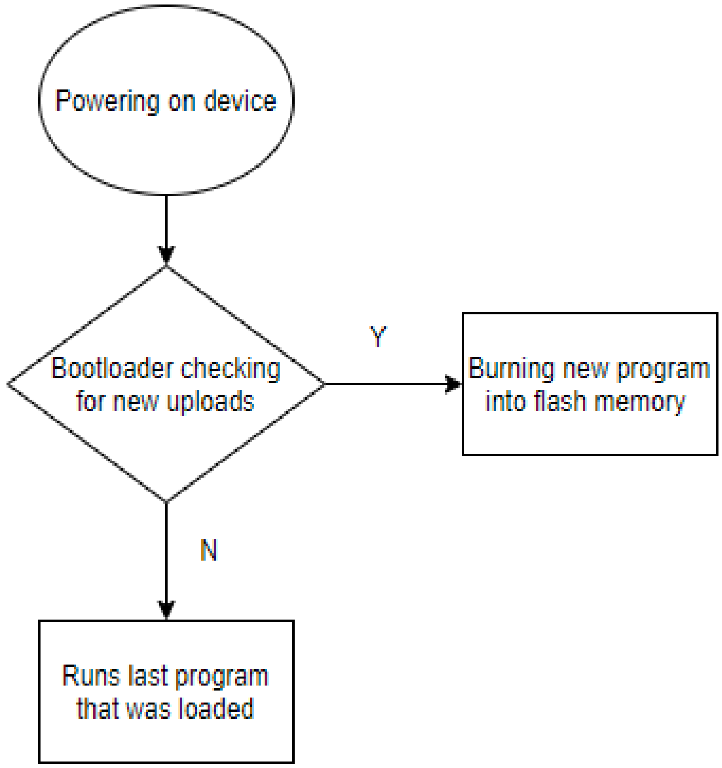


Figure 68: Programming MCU Flow Diagram

The ability to be able to burn the bootloader on the actual PCB that we are design would be a plus. This process is called In-System Programming or ISP, the PCB has to be designed with certain header pins so that the microcontroller can access it when it is on the PCB. The figure below is the schematic of the Arduino Uno that has the ability to program the ATmega328p before putting it onto our own PCB. Using the schematic of the Arduino UNO that was provided from the Arduino website, the headers that enable the board to be able to program the ATmega32p is the three SPI pins (MOSI(D11), MISO(D12), SCK(D13), VCC, GND, and the reset pin). We can use this schematic as a reference to ensure that the board works when attaching the microcontroller to our own custom PCB.

The figure below shows the combination of the USB ISP AVR and 10 pins to 6 pin adapters and how it can be used to connect the adapter to the header that was mentioned earlier so that the microcontroller can be programmed while being attached to the PCB. While the pins are not required to be on the header, the figure below show that there is an item that you can attach to the adapter called pogo pins so that you can just connected the adapter to the board.

Figure 69: USB Adapter

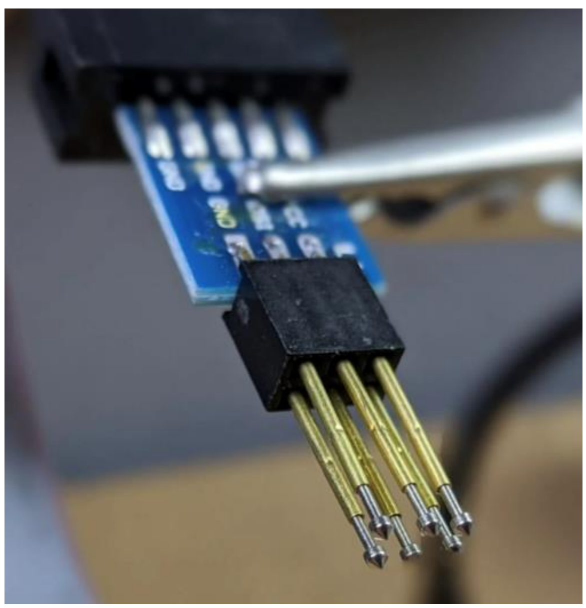


Figure 70: USB 6 pin adapter

5.1.2 Computer Vision

As has been discussed in the overview of the Software Design in Figure 60, the Jetson Nano will bring forth the computer vision functionality (Stretch goal). This feature will allow for the Pet Feeder to dispense food if a dog is detected in front of the Pet Feeder. This can be implemented in two different ways. One way is to have the pet feeder dispense food anytime that the Jetson Nano detects a dog in front of it, granted that there is no food in the bowl already. However, this method could prove detrimental to the dog's health, since it could be easily overfed. Alternatively, the user could set up a certain time frame in which the pet feeder will dispense food if the dog is detected, again granted that the food bowl is empty. Now that the functionality of the computer vision was discussed, the way the code will work can be covered.

In order to achieve object detection, the Jetson needs a trained model for the specific object that must be detected. These models are trained by showing sets of images with the desired object as well as images with other objects that are not the desired one. It can take a long time to develop accurate models, however many highly trained models can be found online and are included with the jetson (note that these models are open source). Therefore, to achieve object detection, the code will work by scanning through each frame of the video feed. Every frame is then processed by the Detect Net framework built into the Jetson Nano to return the objects found. With this powerful framework, the Jetson can easily identify objects in front of the Pet Feeder in real time. Then, if the dog is detected, depending on the protocol chosen, the Jetson Can signal the Micro controller over SPI to dispense food. An example of the Jetson Detecting objects in real time is shown below:

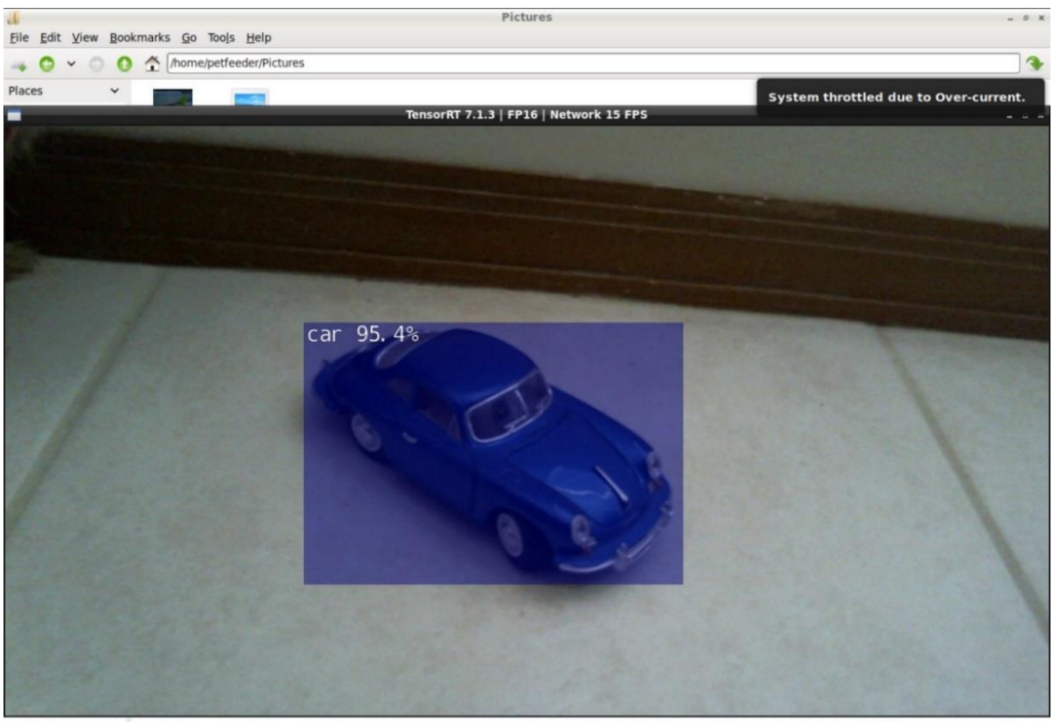


Figure 71: Jetson Nano Computer Vision

As can be seen above, the Jetson Nano can accurately detect a car with 95.4% confidence level. These are stellar results, especially considering that here the Jetson is operating in 5W mode. Moreover, it is detecting the objects at a steady 15 fps, which, for its size, is impressive. One thing to note is that the System Throttled during operation, as can be seen on the top right of Figure 70. The Jetson will throttle if the voltage input or ampere input is too low or high, so it is important to have a stable power supply for the Jetson Nano. As mentioned above, the Computer Vision stretch goal wasn't reached, therefore the Pet feeder will not feature Computer Vision.

5.1.3 Node MCU

The Node MCU will incorporate Alexa into the Pet Feeder. This small, yet powerful device will be responsible for most of the interactions between the user and the Pet Feeder. The Node MCU will interact with AWS services to store the state of the pet feeder. The NodeMCU will ping the AWS Cloud every couple of seconds to update the status of the pet feeder. This way, Alexa will know the status of the bowl and the reservoir of the pet feeder. Once the User pushes a request through Alexa, the Node MCU will communicate said request through a digital status line between the NodeMCU and the ATMega328p. The Node MCU will achieve connection to the Alexa Service through Wi-Fi. Below is a code flow diagram for the general operation of the Node MCU:

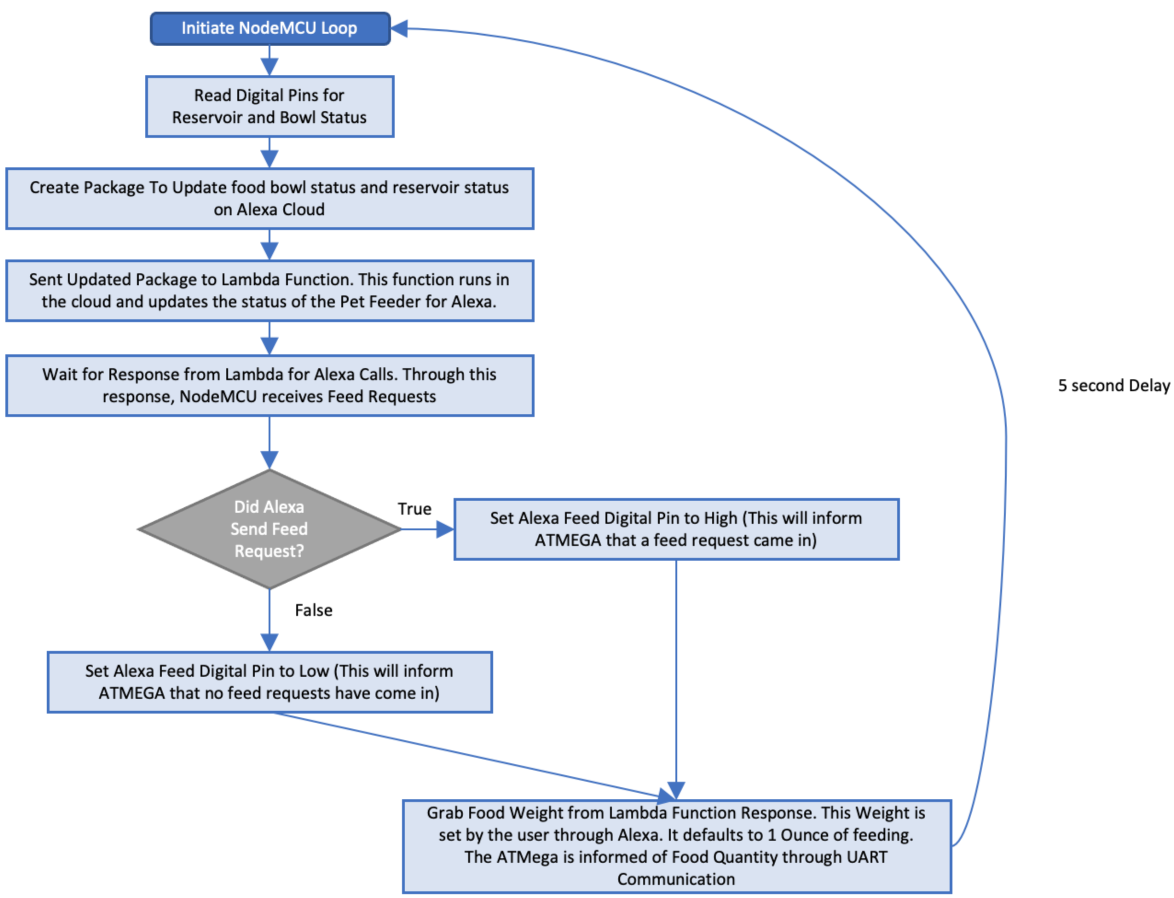


Figure 72: NodeMCU Flow Diagram

As can be seen above, the Node MCU will be in constant contact with the AWS Lambda Function. To update the Lambda Function, the NodeMCU acquires the status of the Pet Feeder, by performing a digital read on two lines connected between the ATMega and the NodeMCU. One of these digital connections between the NodeMCU and the ATMega is used for Bowl status and the other is used for Reservoir status. The ATMega will write a HIGH or LOW signal depending on the status of these respectively. The Lambda Function is run in the cloud and updates the status of the Pet Feeder (More about how Alexa Integration Operates will be explained in section 5.1.3). Every 5 Seconds, the NodeMCU updates the function and also receives a package in return from the Lambda Function. This package contains information on the amount of food the user set to feed the pet and if an Alexa feeding command was sent by the user. The NodeMCU takes this information and updates the Alexa Digital Pin to inform the ATMega, if a feed request came through. If a feeding request did in fact come through, the Pin will be set to HIGH, otherwise it will be set to LOW. The ATMega will then read this pin to figure out if a feeding request came through.

5.1.3 Alexa Integration Design

In this section, the design of the Alexa Skill and other Alexa Components will be covered. Alexa Integration uses a couple of AWS (Amazon Web Services) solutions to operate the Pet Feeder, as well as the Alexa Developer Console. The Alexa Developer Console was used to develop the skill that interacts with the AWS services. The Pet feeder makes use of the AWS Simple Storage Solution (S3) to store the status of the Pet Feeder. Essentially, AWS S3 allows you to store files on the AWS Cloud. In the Pet Feeders Case, the S3 Service holds files that contain the Status of the pet feeder. The status includes the reservoir status, bowl status, and Alexa command status. Next, the Pet Feeder also uses AWS Lambda Functions to operate. AWS Functions are simply functions written in Python and Java Script that can operate when called upon. In the case of the Pet Feeder, there are two AWS Lambda Functions used, one is used by the NodeMCU and the other is used by the Alexa Skill. When the NodeMCU updates the pet feeder, it calls a Lambda Function and sends it the latest status of the Pet Feeder. Then this function updates the S3 Files with the latest status and also reads the S3 file to check if the user requested to feed the pet. The Function also reads this file for the quantity the user set to feed the dog. Once the function acquires this information, the function sends it back to the NodeMCU. On the Alexa Skill side, whenever the Skill is called upon, the Skill calls a Lambda Function with what the user asked. The Lambda Function receives the request and executes the instructions for said request. There are a total of 4 requests: Feed Request, Reservoir Status Request, Bowl Status Request, and Set Food Bowl Weight (Food Quantity) Request. As mentioned above, if there is a feed request, the lambda function will update the Feed Intent on the S3 File to a 1 (representing Feed). For the reservoir request, the status is updated on the S3 File by the NodeMCU every 5 seconds. Therefore, the Lambda Function for the skill will simply check the value of the Reservoir Status Bit on the S3 File. For the Bowl Request, the process will be the same, the Lambda Function checks the food bowl status bit on the S3 File. Finally, for the set food bowl weight intent, the Lambda Function updates the S3 File on the cloud with the new food bowl weight requested by the user. Then this value will be sent to the NodeMCU whenever the NodeMCU calls the Lambda Function again. Below is a general overview of how Alexa Integration Operates for the Pet Feeder:

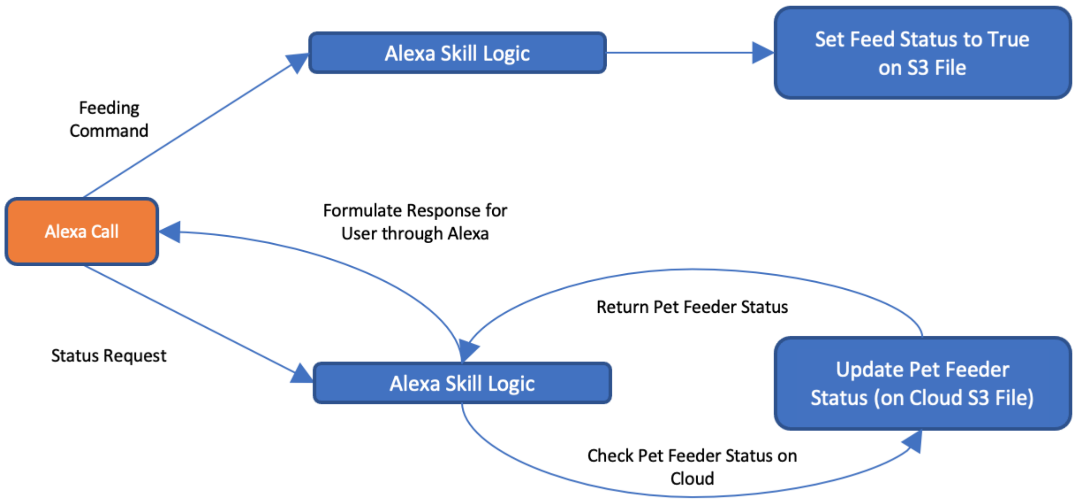


Figure 73: Alexa Integration Flow Diagram

5.1.4 Achieving Low Power Consumption

One of the main goals with the Pet Feeder is to have battery operation. This allows for the user to be able to place the device anywhere in their living space. However, with these many processing units it can be difficult to achieve said functionality. Next are some of the ways the Pet Feeder will achieve low power consumption.

Even though this is a stretch goal, the device that consumes the most power in the project is the Jetson Nano. At 5V and 3 Amps, the Jetson Nano can draw up to 15 Watts, which is quite high. However, this will only occur if the Jetson Nano is being pushed to the limits, but this is not difficult to do when running object recognition frameworks. This obstacle will be mitigated by adding a motion sensing device. This device will be connected to the Jetson Nano. As soon as motion is detected, an interrupt will be thrown to wake up the Jetson Nano and engage object detection. If the Jetson does not detect a dog within 10 seconds, the Jetson can go back to sleep. This approach is much more efficient because most of the time, there will be no movement in front of the pet feeder, so much energy can be saved. Moreover, the Jetson can also be placed into a low power consumption mode, where it runs at 5V 1 Amp. This alternative can also significantly reduce power consumption of the pet feeder.

5.2 Hardware Design

This part of the report will talk about the hardware aspects of our project and how the wiring for each of the parts that we have chosen will work. Lastly, we will talk about the schematic and board layout for the designed DC/DC converter.

5.2.1 Hardware Layout

The following sections/sections will outline the design and implementation of the sensors and other hardware that will be used during this project. All of the components that are chosen to need to make sure that they do not require a lot of power but while keeping the accuracy of each sensor and motor at high so that we do not lose any efficiency of our sensor.

5.2.1.1 Ultrasonic Sensor

The HC-SR04 ultrasonic sensor was chosen for the food level detection of our project. This specific sensor is going to be placed onto the housing unit and be able detect when there is not an object in front of it, in this case the object will be the pet food. An ultrasonic sensor measures a distance by using ultrasonic waves. The sensor emits an ultrasonic wave and receives a wave back from the target, and in this case the target will be the back of the housing unit. For example, if the distance between the front and back of the housing unit is 20cm, the ultrasonic sensor will wait until it detects a distance of greater than or equal to 20cm meaning that there is no food in its path and that the food level in general is low. The specific ultrasonic sensor that we have chosen for our project is a 4-pin sensor. One of the pins is a Vcc pin that will be hooked up to the +5V pin of our microcontroller. The next pin a ground pin and that will be a common ground with all other sensors in the project. The next pin is a Trigger pin which is an input pin, and that pin has to be kept high for 10 microseconds to initialize measurement by sensing US wave. Lastly, the fourth pin an Echo pin, an output pin, and this pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor. Both, the trigger, and echo pins are digital pins. The wiring diagram between the ultrasonic sensor and the MCU can be found below. For the sake of a good image, the Arduino UNO board is used but this will not be used for the project. Instead, we will just be using the microcontroller on the Arduino UNO which the Atmega328 chip.

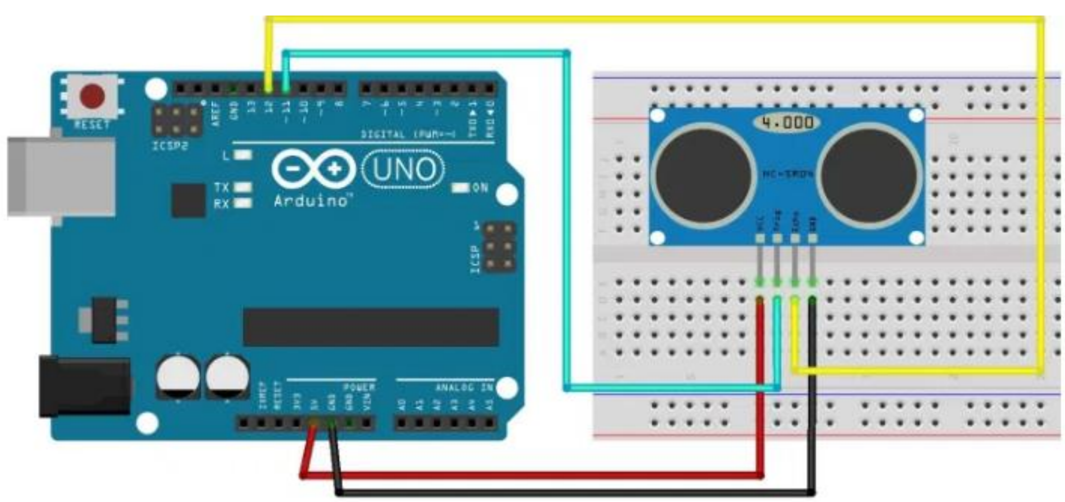


Figure 74: Ultrasonic Sensor Wiring Diagram

The table below will show a better description of how the pins need to be connected to the microcontroller.

Ultrasonic Sensor HC-SR04 Pins	Arduino UNO Pins
VCC (Pin 1)	+5V
Trigger (Pin 2)	Pin 11
Echo (Pin 3)	Pin 12
GND (Pin 4)	GND

Table 43: HC-SR04 and Arduino UNO Connections

As seen in the table below, the trigger and echo pins are connected to pins 11 and 12 of the Atmega328 respectively but they can be connected to any of the digital pins on the board.

5.2.1.2 Weight Sensor

The load cell with the amplifier was chosen to do the weight sensing part of our project. This load cell sensor is going to be placed underneath a bowl and will accurately be able to detect the weight that is on top of the bowl. So, what a load cell is that it is a transducer that will measure the force and then output this force as an electrical signal. This setup will also have an HX711 load cell amplifier and whose working principle is to convert the measured changes in resistance value through the conversion circuit and then output an electrical signal. For the setup of the Atmega328p to the HX711, the +5V will go to the VCC of the amplifier, ground will connect to ground, then we have an SCK and a DT pin that will connect to two digital pins of the microcontroller. We then take the four output pins of the amplifier and connect them to the four pins of the actual load cell so that it can properly detect the weight that is attached to the load cell. This specific design is going to be very important aspect of our project and we need to make sure that the design is done properly, or it could screw up the entire project. Eventually we will need to also hook up the motor to the microcontroller as well and then test to make sure that once the load cell reaches a desired wight, the motor will keep spinning that thus ejecting more food into the bowl. Finally, once the load has reached that desired weight, we need to have so that the motor will stop dispensing food without much overshoot so that we have an accurate amount of food and that we are not overfeeding the pet.

Below, we can see the wiring diagram of how the microcontroller will be hooked up to the HX711 amplifier and then how the HX711 amplifier will be hooked up to the actual load cell. Obviously for our project, it will be only the Atmega328P chip that will be connected to the microcontroller but for the sake of simplicity, we decided to do a wiring diagram for with the entire Arduino UNO board.

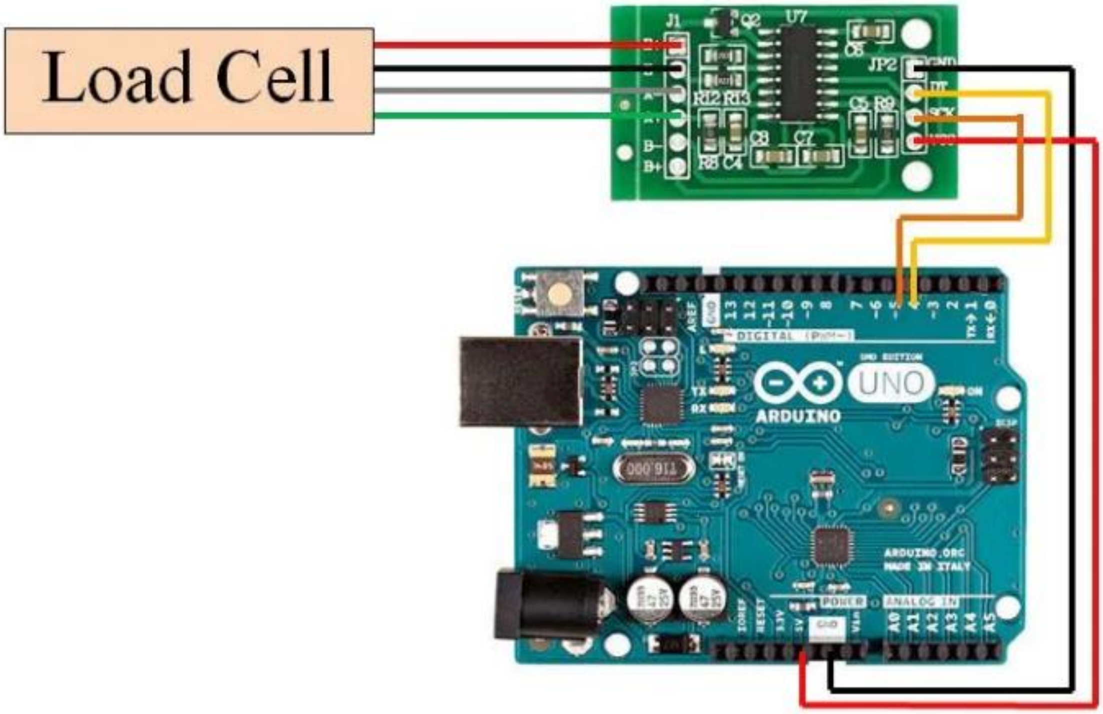


Figure 75: Load Cell Wiring Diagram

Below, you can see the pinout descriptions and the connections that need to be made in order to connect the HX711 amplifier to the Arduino UNO board/Atmega328P microcontroller chip.

HX711 Pins	Arduino UNO Pins
VCC	+5V
GND	GND
SCK	Pin 5
DT	Pin 6

Table 44: HX711 and Arduino UNO Connections

As seen in the table above, the serial clock and DT pins are connected to pins 5 and 6 of the Atmega328 respectively but they can be connected to any of the digital pins on the board.

Below, you can see the pinout descriptions and the connections that need to be made in order to connect the HX711 amplifier to the Load Cell. This connection needs to be correct in order to ensure that the load cell can accurately know the correct weight that is on top of the sensor.

HX711 Pins	Load Cell Pins
E+	Red
E-	Black
A-	White
A+	Green

Table 45: HX711 and Load Cell Connections

5.2.1.3 Motor/Motor Driver

The setup for the motor, motor driver and the mic are very similar to the setup of the load cell and the HX711 amplifier. The reason that we need to use a motor driver for the NEMA motor is because these motors require a high amount of current in order to operate and the controller circuit deals with low current signals so a motor driver is used in order to take those low current signals and turn it into a high current signal that can operate the motor. So, from the microcontroller we have a 5-pin connection to the motor driver. We see four input pins which go to any four digital pins and then we have a 12V power supply that is going straight the Vmot on the A4988 driver and a connection of ground to the motor driver and the microcontroller. Next, we are able to setup for the NEMA motor. From the motor driver pins, we have an A+, A-, B+ and B- that are going to connect to the Red Green, Yellow and Blue pins of the NEMA motor. It is very important that we properly connect the motor driver pins to the correct motor pins, or we are not going to be able to accurately program the motor to do what we want it to do.

Below, we can see the wiring diagram of how the microcontroller will be hooked up to the A4988 motor driver and then how the A4988 motor driver will be hooked up to the actual NEMA 17 stepper motor. Obviously for our project, it will be only the Atmega328P chip that will be connected to the microcontroller but for the sake of simplicity, we decided to do a wiring diagram for with the entire Arduino UNO board. Later on, we take all of the designs that we have talked about and add them all onto a PCB with their appropriate pins. This will be important aspect of the project because if the pin connections are wrong on the PCB, then we are not going to be able to connect the pins on any of our sensors/motors how we want leading to us not having a properly working project at the end of Senior Design 2.

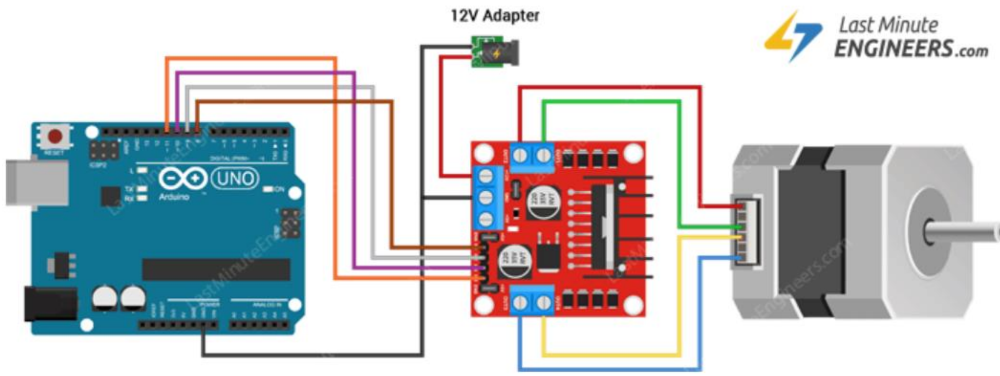


Figure 76: Stepper Motor Wiring Diagram

Below, you can see the pinout descriptions and the connections that need to be made in order to connect the A4988 motor driver to the Arduino UNO board/Atmega328P microcontroller chip.

A4988 Pins	Arduino UNO Pins
IN1	Pin 8
IN2	Pin 9
IN3	Pin 10
IN4	Pin 11
GND	GND

Table 46: A4988 and Arduino UNO Connections

The connections above can be making to any four digital pins on the microcontroller but in this case, we input the fours input pins of the motor driver into pins 8, 9, 10, and 11, respectively.

Below, you can see the pinout descriptions and the connections that need to be made in order to connect the A4988 motor driver to the actual motor. This connection needs to be correct in order to ensure that the motor can the current actuated and thus we make sure not to supply overcurrent.

A4988 Pins	Nema Motor Pins
A	Red
A-	Green
B	Yellow
B+	Blue

Table 47: A4988 and Nema Motor Connections

5.2.1.4 NodeMCU and ATMega328p Comms

As previously discussed, the Pet Feeder makes use of the NodeMCU for Alexa Integration and the ATMega for operating the actual components of the Pet Feeder (i.e., Motor, Ultrasonic Sensor, and Scale. Since the NodeMCU is the broker for all of the requests, the ATMega and NodeMCU need to communicate. In this section, the communication between the two will be discussed. All in all, there are 4 items that the ATMega328P and NodeMCU have to communicate about. These are the bowl status, reservoir status, food bowl weight, and feed command. In the pet feeder, the bowl status, reservoir status and feed command are all communicated with the use of digital pins. Essentially, the reservoir status and food bowl status are digital outputs to the ATMega328p and digital inputs for the NodeMCU. Since the ATMega328P monitors the status of the reservoir and the food bowl, it will set the reservoir digital output to HIGH if the reservoir is full and LOW if the reservoir is almost empty. Similarly, the ATMega328P will set the digital output pin to HIGH if the food bowl is full and to LOW if the food bowl is empty. Since the NodeMCU receives the Feed Requests from Alexa, a digital output pin is used from the NodeMCU to a digital input on the ATMega328P. Whenever a feed request comes through, the NodeMCU sets that digital output to HIGH, and then sets it to LOW again after the feed request goes through. The ATMega328P will detect this digital Input immediately and dispense food. Finally, the only request that isn’t digital (0 or 1) is the food weight. For this UART had to be used. Here the NodeMCU is communicating the information to the ATMega328P, therefore only the Tx (transmission) pin of the NodeMCU had to be connected to the Rx (receive) pin to the ATMega328P to communicate the information. Finally, both the NodeMCU and ATMega328P shared a common ground to allow for UART Serial Communication. Below are how both of these devices were connected to each other to communicate:

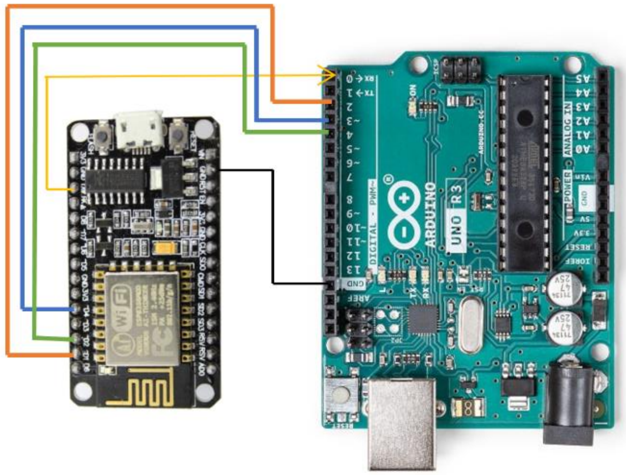


Figure 77: Stepper Motor Wiring Diagram

5.3 Power Schematic/PCB Design

Designed from WEBENCH, we are able to put in our design requirements for a DC/DC convertor. The design requirements that are required are a 12V input voltage with a 5V output voltage. The schematic that was built using EasyEDA can be found in the image below. The switching regulator that is used for this design is a LM2576HVSX, the design also requires two capacitors, a 100uF and a 1000uF along with one diode. We can then take this schematic and convert that to a PCB and then trace the parts on the PCB and then finally add a copper ground plane on the top and bottom layers on the board. The reason to include a ground layer is to have the shortest possible path to ground for all parts that require a ground connection in the design.

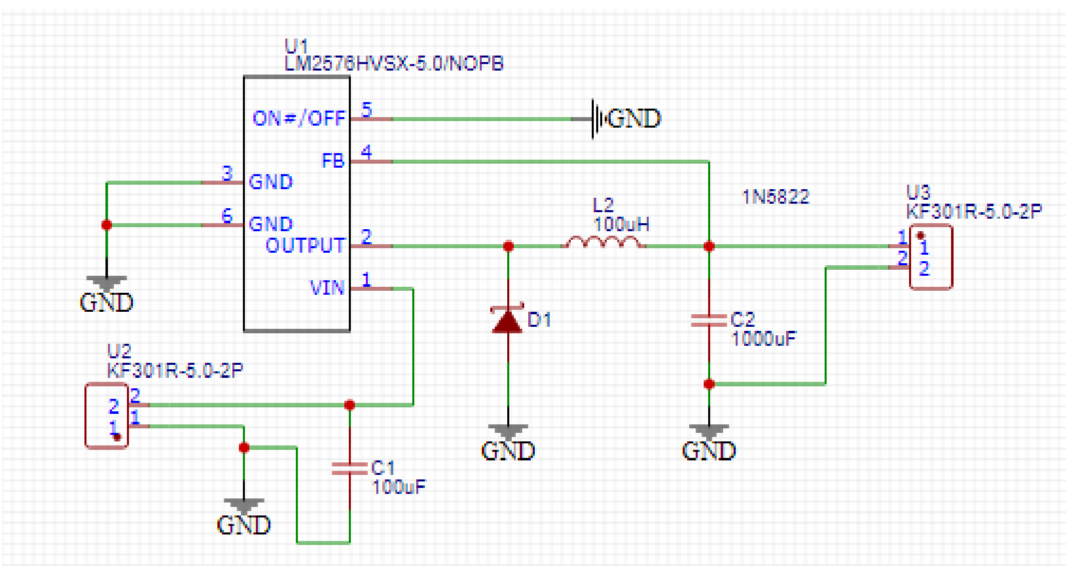


Figure 78: 12V to 5V DC/DC Schematic

The top layer of the board can see below with a copper ground plane also included. Also, below, we can see the initial traces with a ground plane so we can see how a via was implemented. We needed to use a via since we could not trace underneath an existing trace. Because we did not want a very long trace, the use of a via allowed us to trace underneath the board and then finally trace back on the top layer.

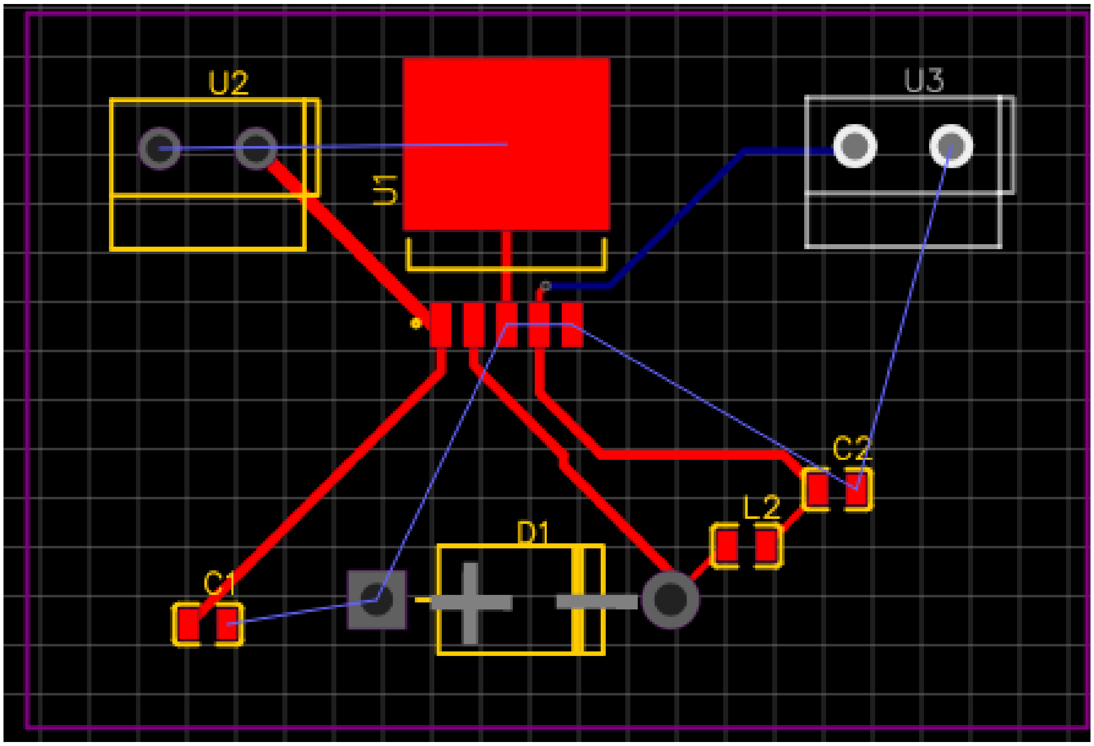


Figure 79: 12V to 5V DC/DC Board Layout w/o Ground Plane

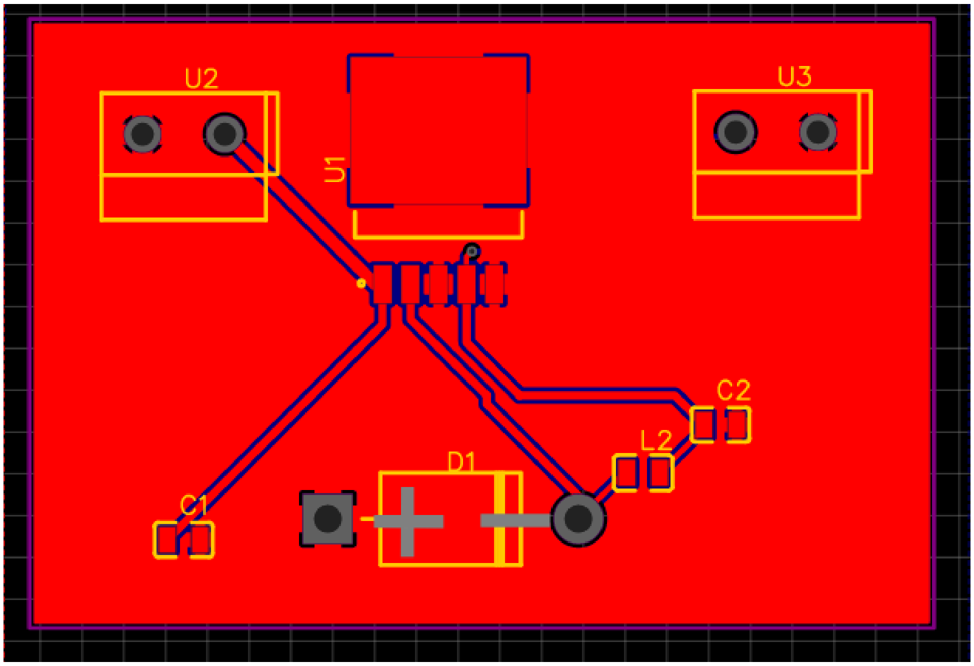


Figure 80: 12V to 5V DC/DC Board Layout w/ Ground Plane

5.4 Main Schematic/PCB Layout

Below is the main PCB schematic which utilizes plenty of pin headers for our sensors along with the motor and motor driver. It also has the Atmega328P soldered onto the board and 3 debugging LEDs which will be very useful for the user.

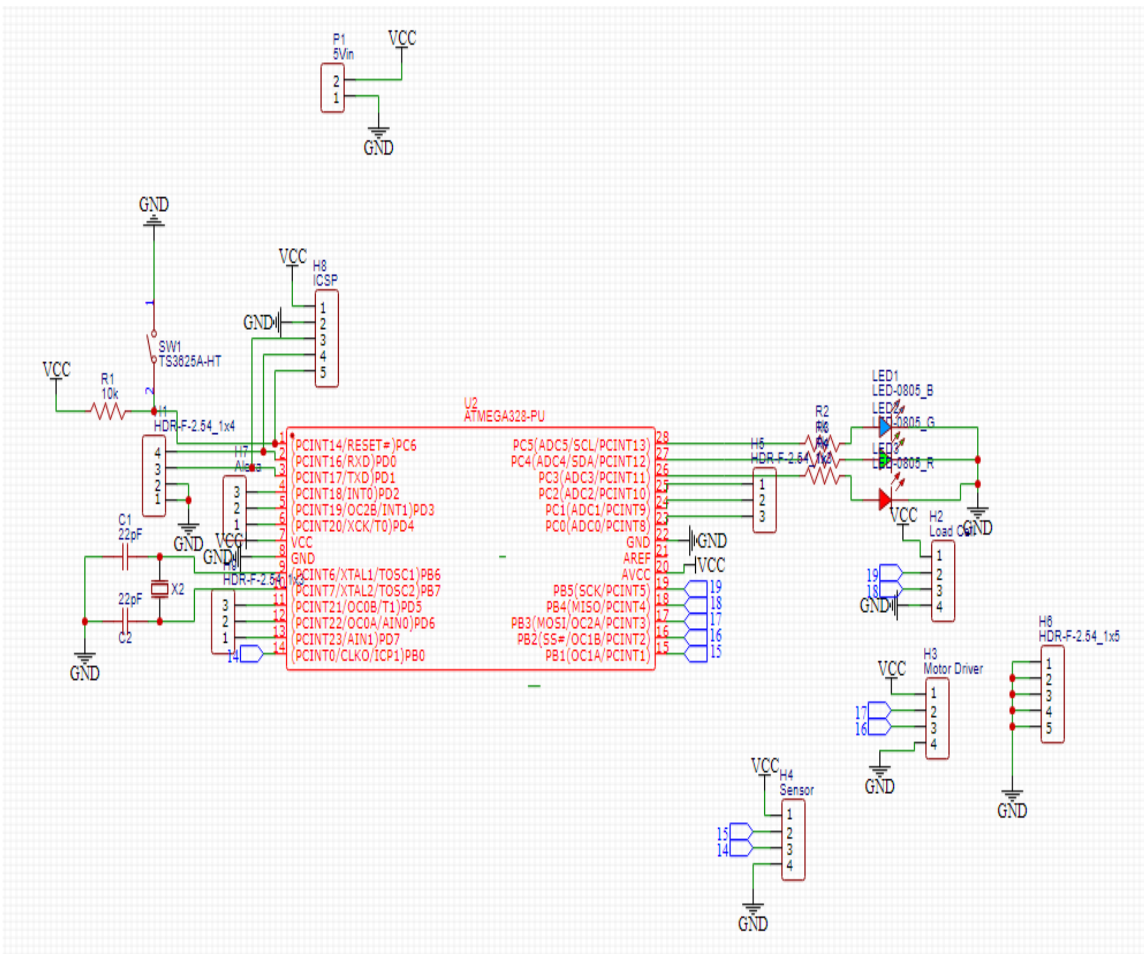


Figure 81: Main PCB Schematic

The top layer of the board can see below with a copper ground plane also included. Also, below, we can see the initial traces with a ground plane so we can see how a via was implemented. We needed to use a via since we could not trace underneath an existing trace. Because we did not want a very long trace, the use of a via allowed us to trace underneath the board and then finally trace back on the top layer.

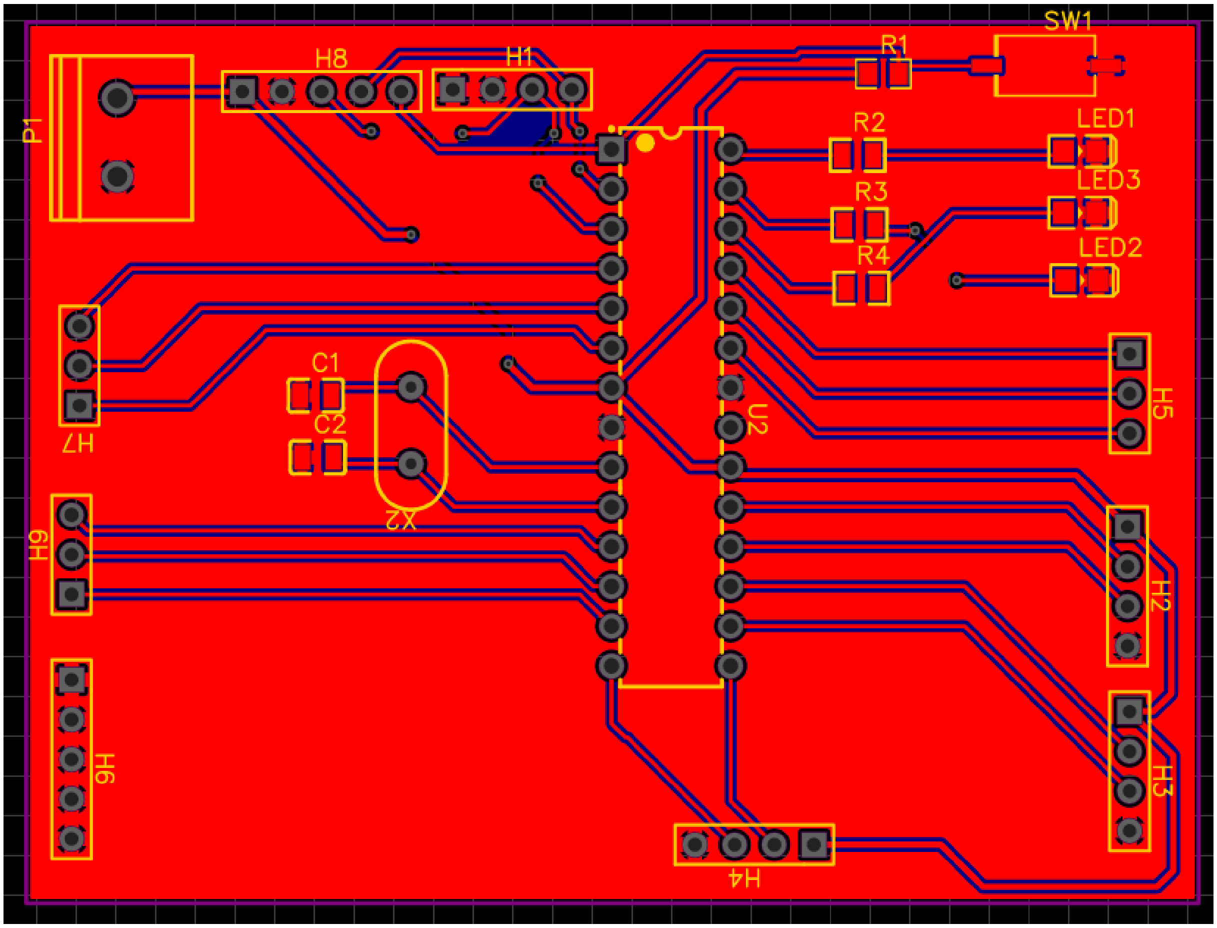


Figure 82: Main PCB Layout

6.0 Testing Plans/Results

This section aims to perform some preliminary testing of the Hardware and Software that has been developed at this point. Some notable items are computer vision, power supplies, Alexa integration, and comms between devices. All of these items, as well as others, are crucial to ensuring that the Pet Feeder works as intended. These items, and more, will be covered in this section.

6.1 Software Testing

In this section, the idea is to cover some of the initial testing of some of the software components in the project. These are most notably the computer vision component and Alexa integration with the Node MCU.

6.1.1 Computer Vision Testing

In order to ensure that the Computer Vision component of the Pet Feeder works correctly, the Jetson's ability to detect a dog must be tested. Below are a couple of screenshots of some initial testing with dogs. This is using the Jetsons Detectnet framework, that takes advantage of the Jetson's Maxwell GPU to output steady 15 fps during object detection.

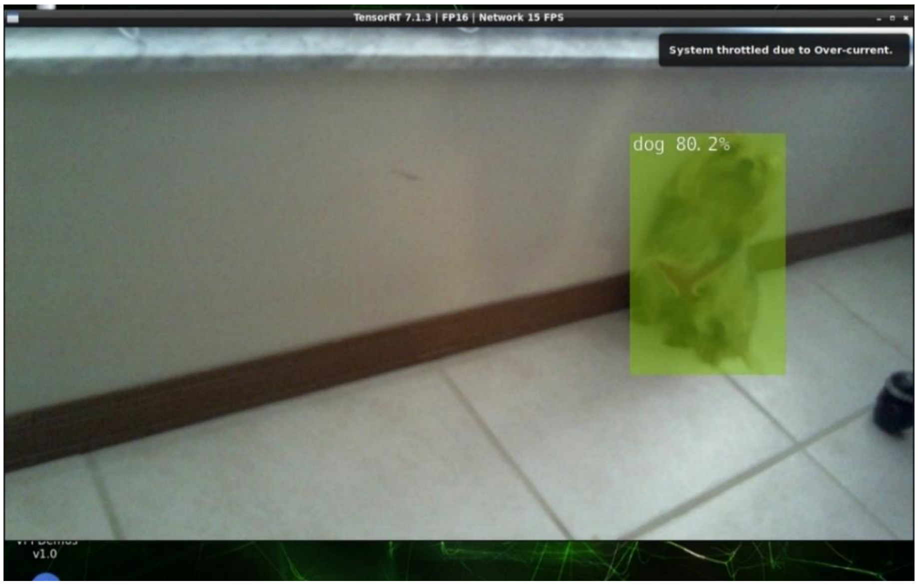


Figure 83: Jetson Nano Computer Vision Test #1

As can be seen above, the Jetson confidently detected the dog. On this image, the dog was farther away though, this caused the confidence levels to drop. Below is another screenshot that showed higher confidence levels:

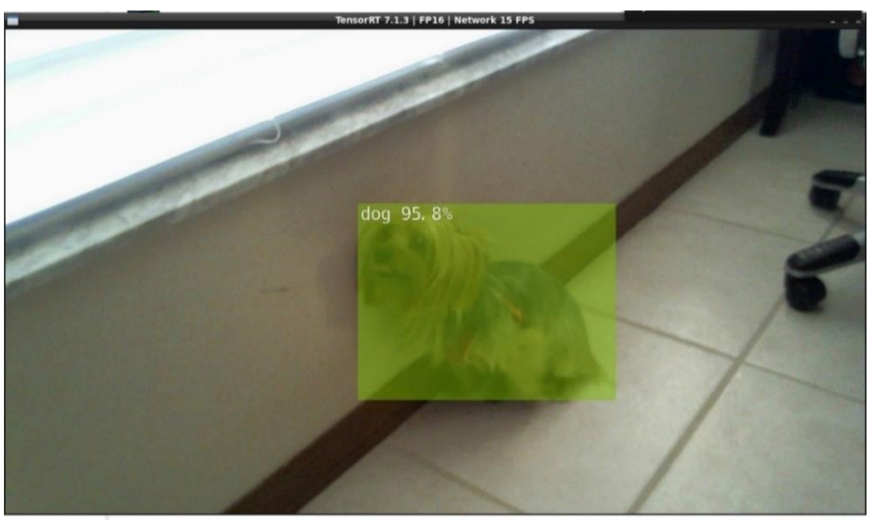


Figure 84: Jetson Nano Computer Vision Test #2

As can be seen above, as the dog got closer, the confidence level increased. The detection is plenty accurate to be used for the Pet Feeder. Finally, below is another screenshot, in which there is two dogs, and they are directly in front of the camera. As can be seen the Jetson detected both of them with high confidence levels:

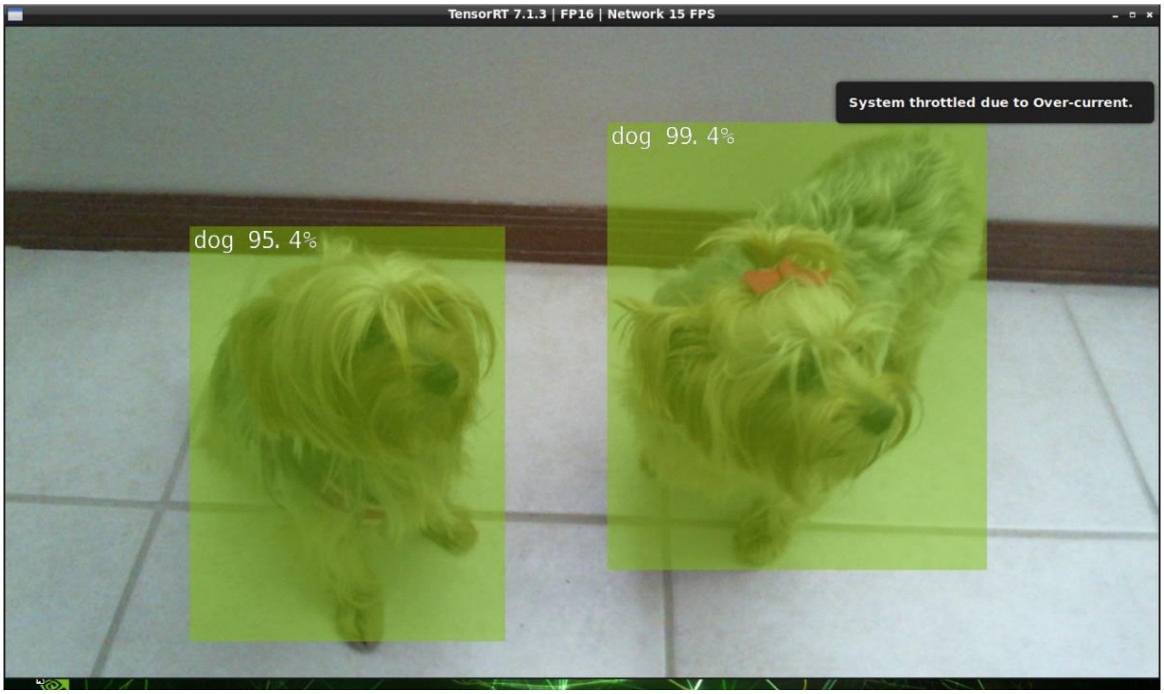


Figure 85: Jetson Nano Computer Vision Test #3

As can be seen by the testing completed, the Jetson can confidently detect dogs consistently. This preliminary test showed that the Jetson Nano was the perfect choice for the computer vision portion and that it will successfully bring this feature of the pet feeder to life.

6.1.2 Alexa Integration Testing

Alexa is the interface that the Pet Feeder uses. All of the Pet Feeder's capabilities will be used through Alexa. For this reason, some preliminary testing of the software design must be conducted. The device that we are going to use for Alexa integration is the NodeMCU since we decided that it is best for power consumption. As described in the Software Design Section (5.1), the NodeMCU communicates with a Lambda Function to acquire Alexa Status. Moreover, this Lambda Function Also updates the status of the Pet Feeder in the Cloud. To ensure that this design works, a simple LED circuit will be put together to perform the Alexa Integration Testing.

We are going to test the Alexa Integration using an LED that is going to be connected to the breadboard. The other materials that we need are going to be in the shown in the table below. Moreover, the use cases for each item will be shown.

Material	Use Case
NodeMCU	Using the wireless connection to the Wifi which is be used to receive the Alexa signals.
NPN Transistor	Current amplifier for the low current of the NodeMCU
Relay	This is used to turn on and off switches. In this case, it is used to turn the light on and off.
Diode	Used as a flyback diode. To prevent damage to the NodeMCU
Resistor	Prevent short circuit.

Table 48: Material for Alexa Integration

The schematic below shows the configuration of testing layout. This will be tested on a breadboard. The pin 4 is the GPIO 4 on the NodeMCU and we are going to output the pin based on the signal that the Alexa is going to send, this will result in the switch being turned on. As you can see, the GPIO 4 has a resistor that leads to the base of the transistor and the signal is sent to the relay so that the switch can be turned on and off. The relay will be connected to an LED that will turn on with the Alexa command. The diode has to be connected to the relay to prevent the sudden spike through the inductor, this is called a flyback diode.

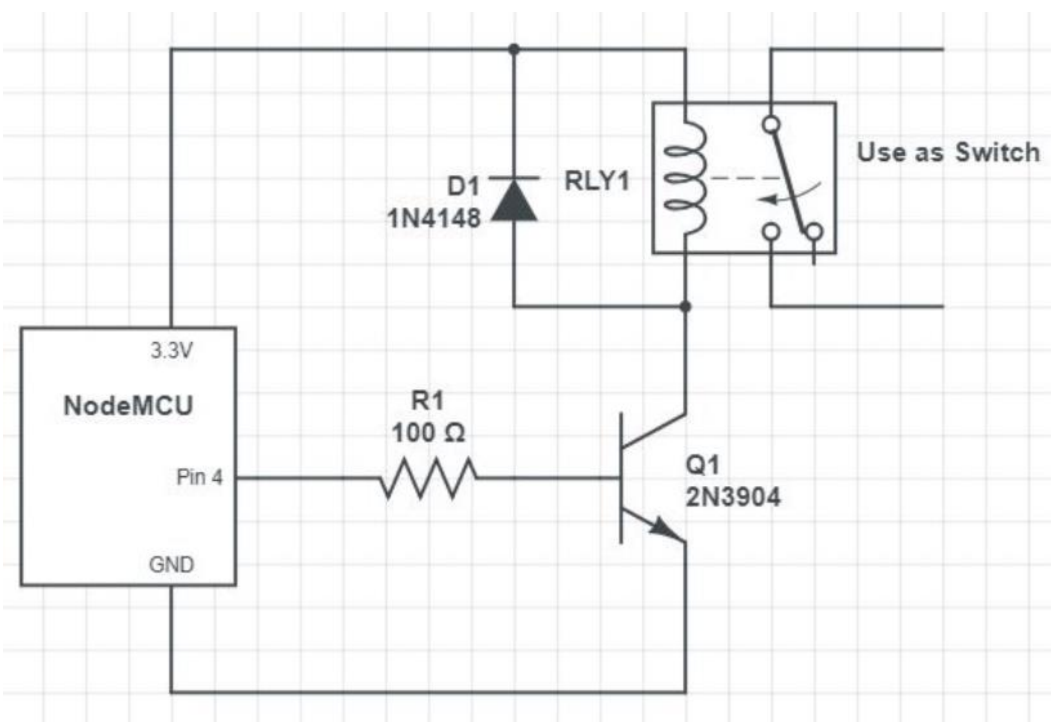


Figure 86: Schematic for Alexa Integration Testing

6.1.3 NodeMCU/Arduino UNO SPI Communication Testing

The NodeMCU has pins that are multiplexed so it does not matter what communication protocol is chosen since each GPIO pin can support PWM/UART/SPI. Since the Arduino UNO only have a set of TX and RX pins we are going with SPI, so all the testing will be done with that communication protocol. Since the NodeMCU has the four SPI hardware pins, this enable the NodeMCU to be able to communicate with the Arduino UNO. The NodeMCU is exclusively a four wire SPI communication device since the four pins are (SD1, CMD, SK0, and CLK).

The figure below shows the pinout layout for the SPI pins connections for both the NodeMCU and the Arduino UNO that we are going to be using for testing purposes. The CLK or chip select is also known as the SS, which is the slave select, this pin is used so that the master can select the slave that they want to communicate with through that pin. The MOSI is where the master will send the data and the slave will receive the data. The MISO is the opposite of MOSI and the SCLK is where the master will generate the clock which the slave will use to keep transmitting data. The GPIO pins that contain the 4 SPI pins on the right side of the board are not available for use since they are used by the internal flash. This means that if you were to do test the SPI communication then you would have to use the pins that have the prefix "H" before the four SPI pins. These pins are on the right side of the board and are GPIO12 to GPIO15.

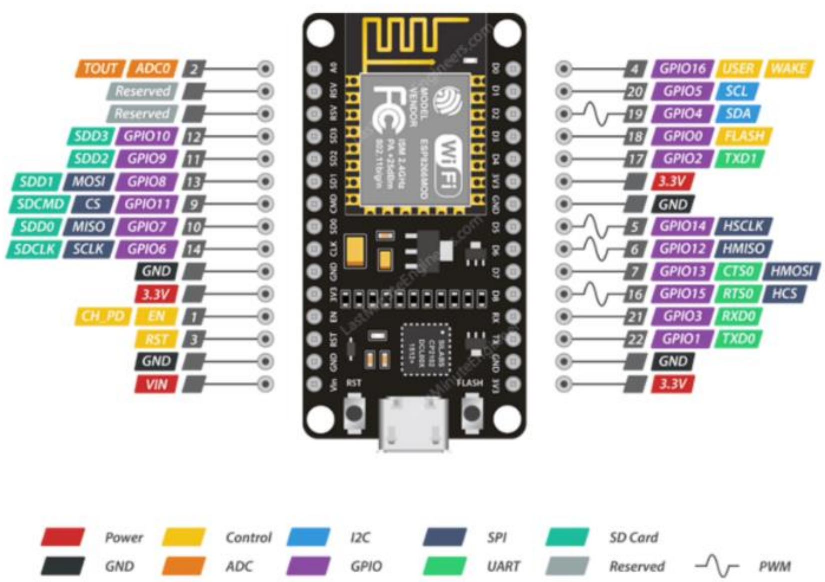


Figure 87: NodeMCU Pins for SPI

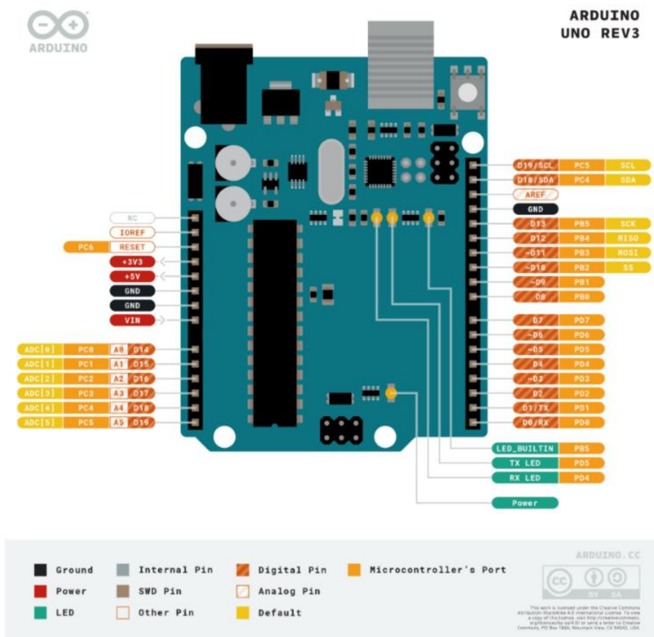


Figure 88: Arduino UNO Pins for SPI

The figure below shows the wiring for the connection of the SPI communication between the NodeMCU and the Arduino UNO.

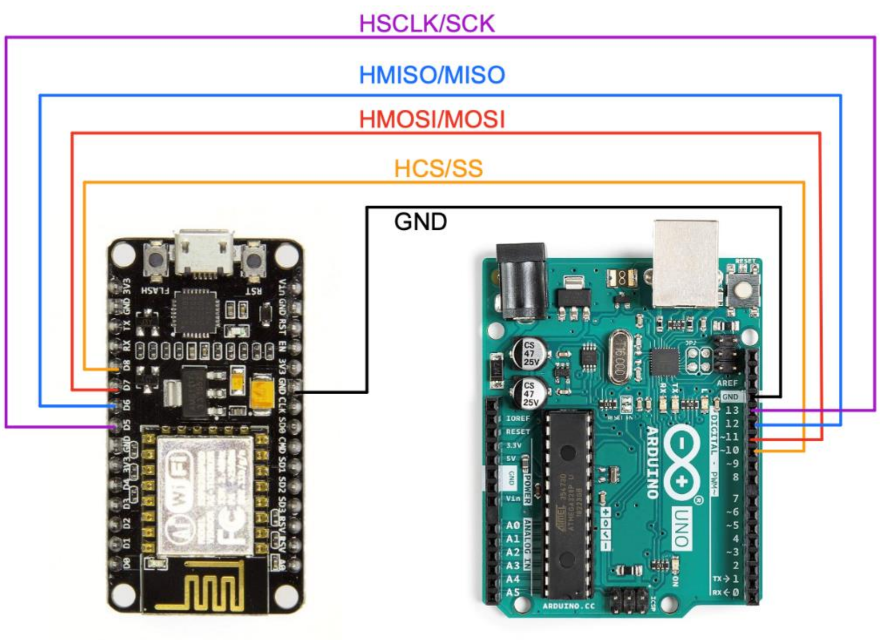


Figure 89: SPI Communication between Arduino UNO and NodeMCU

We can start doing some testing using the Arduino IDE and write a simple sketch, which just means a program in the Arduino environment. The sketch that we are going to use is to test the communication between the NodeMCU in this example will act as the master, while the Arduino UNO will be the slave. The figures below show how each sketch works and the communication between the NodeMCU and Arduino UNO.

The sketch on the NodeMCU is going to send a message that says, "Sending message to the slave" that is shown in the figure below. The message will keep sending since it is in an infinite loop.

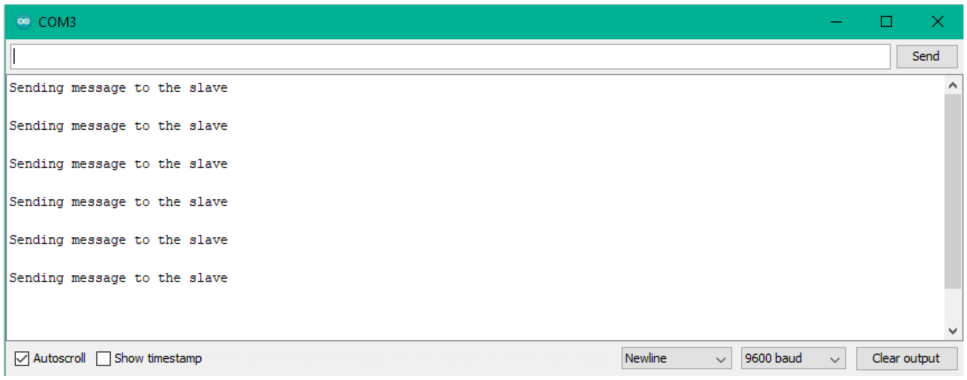


Figure 90: Arduino UNO Example Communication Sketch

In the function called the setup, you set the baud rate to 9600 and start the SPI communication with the built-in function. This will be the end of the setup function, the next function that is written is just a loop where is going to transmit each bit of the message if there is power supplying the board, there could be a case where it stops looping if it hits the null character or you can use a for loop that takes the size of the string.

The sketch on the Arduino will also have the setup in which you have to set the baud rate as the same value as the master. Set the MISO pin as the output pin through the built-in function, set up an interrupt that will be later used, and make sure to enable SPI communication. For the final implementation, we decided to use the TX and RX pins for UART communication protocol. We used this protocol so that user was able to use Alexa to set the user desired weight for how much food would be dispensed.

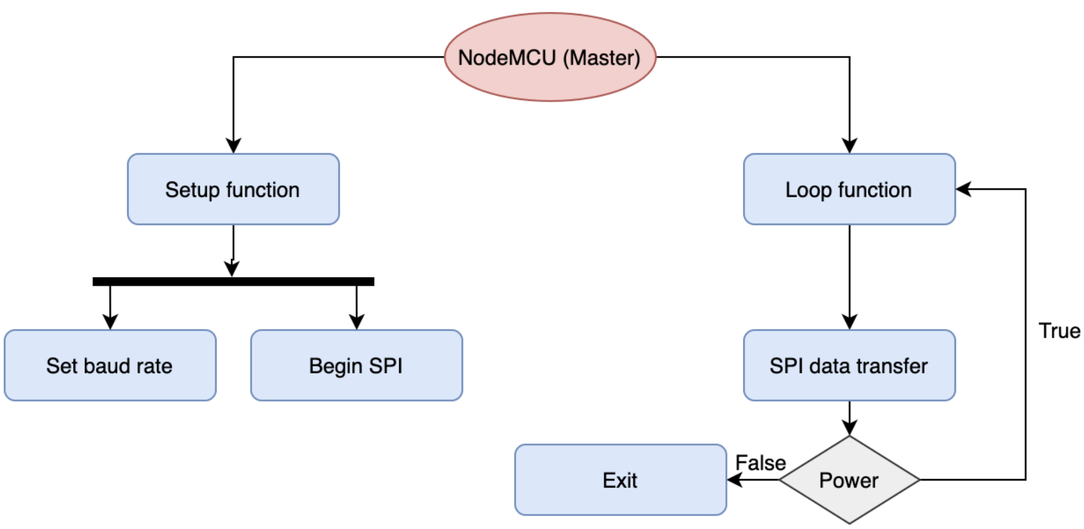


Figure 91: NodeMCU SPI Flow Diagram

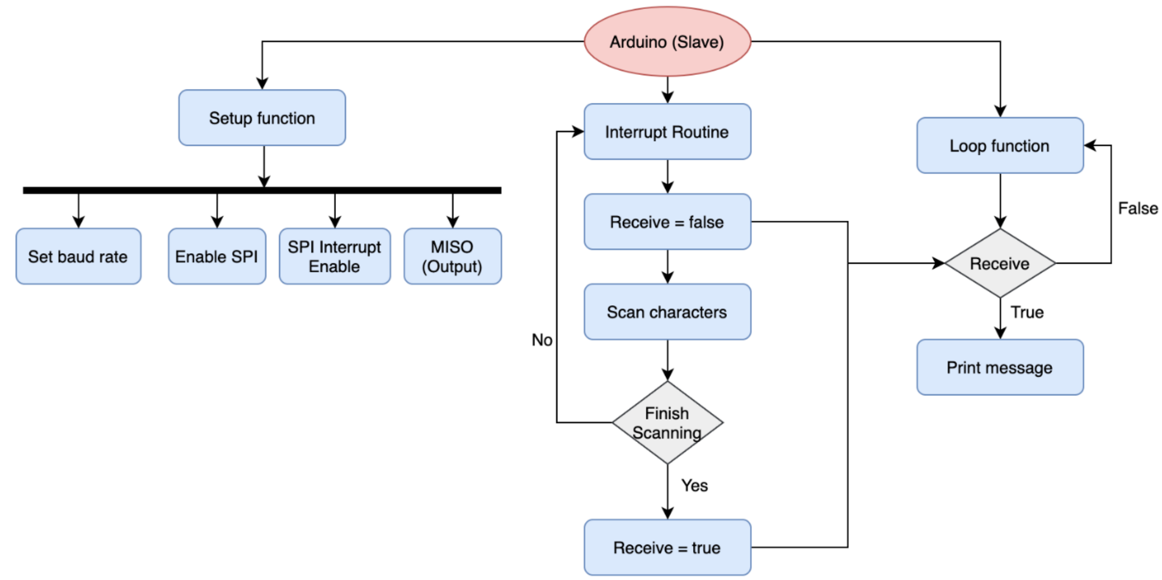


Figure 92: Arduino SPI Flow Diagram

6.1.4 NodeMCU/Arduino UNO UART Communication Testing Plans

The protocol that will be used to communicate between devices is UART. This protocol is supported by all the devices and is, arguably, the easiest one to deploy. As discussed earlier in the research section, this protocol makes use of the Rx and Tx pins on the device. These two are used for the serial communication. Here the communication between the NodeMCU and the Arduino UNO through UART will be covered. In the previous section Figure 82 and Figure 83 show the pin layout of both the Arduino UNO and the NodeMCU. Keeping this layout in mind, below is the setup for UART serial communication between said devices:

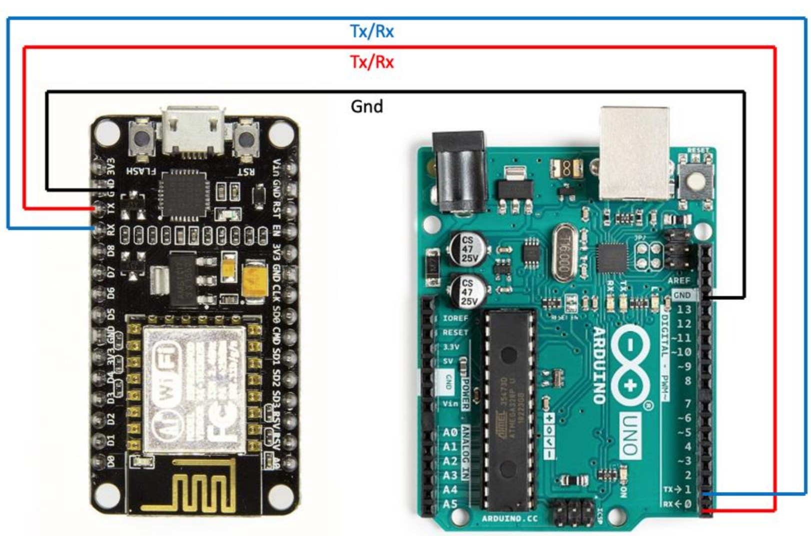


Figure 93: UART Communication between Arduino UNO and NodeMCU

As can be seen above, the Rx Port of the Arduino UNO is connected to the Tx port of the NodeMCU and the Rx port of the NodeMCU is connected to the Tx port of the Arduino UNO. Lastly, the Ground lines of each device are also connected. With these wires' setup, the communication of these two devices can be setup. In order for UART communication to begin, a setup function must initialize both devices for UART communication. This requires both devices to have the same baud rate as well as a couple of other parameters. After that, communication can begin. Below is a code flow diagram for the communication among both of these devices:

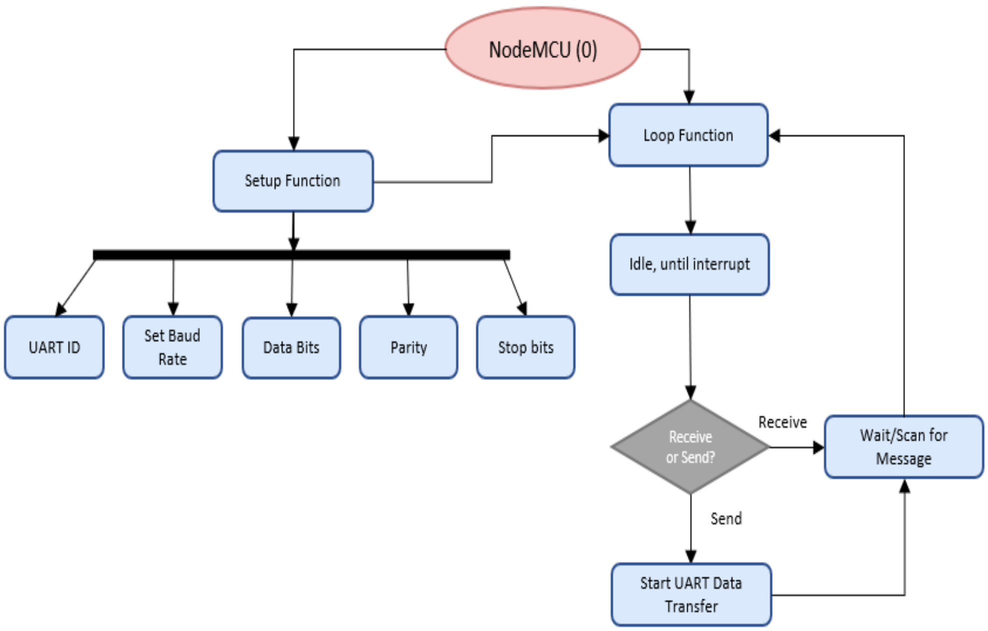


Figure 94: NodeMCU UART Flow Diagram

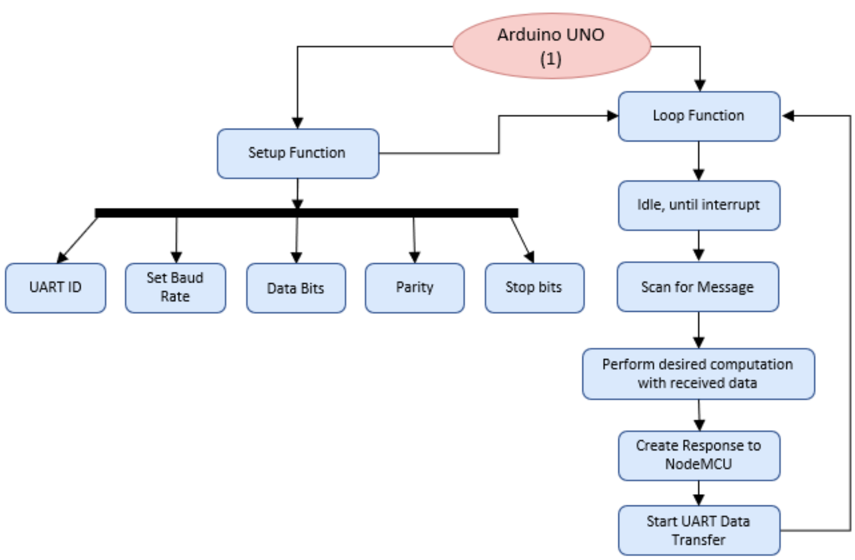


Figure 95: Arduino UNO UART Flow Diagram

Notice above that both devices require a setup function. This basically ensures that both devices communicating through UART are on the same page. The UART ID serves as an identification for both devices communicating. Here, the Node MCU is device 0 and the Arduino UNO is device 1. The baud rate is essentially the speed at which data is transferred among devices. The baud rate must match between the devices to ensure that data bits are not lost during transmission. Moreover, the amount of data bits lets the device know how many bits to expect per packet. The parity bits per packet can be used to detect errors during transmission. Finally, the stop bits let the devices know when the end of the packet has been reached. After UART communication is initialized, the Node MCU will send information to the Arduino UNO, the Arduino will process the information and then respond. Through this method the functionality of UART can be tested for the Pet Feeder. Proper communication among devices is crucial for the implementation of the Pet Feeder, since all the devices need to work in unison.

6.2 Ultrasonic Sensor Testing Plans/Results

The ultrasonic sensor is going to be used for food level detection in this project. It will be mounted on the inside of the housing unit where the food will be stored. We want to have it so once it detects the other side of the housing unit, it will communicate with the user that the food is low. So, let us say the distance between 1 side of the housing unit to the other is 1ft or 12in, once the ultrasonic sensor reads that the distance between the two sides is greater than or equal to 12 inches, it will raise a flag and let the user know. The wiring diagram for how the ultrasonic sensor should be hooked up can be found in the screenshot below.

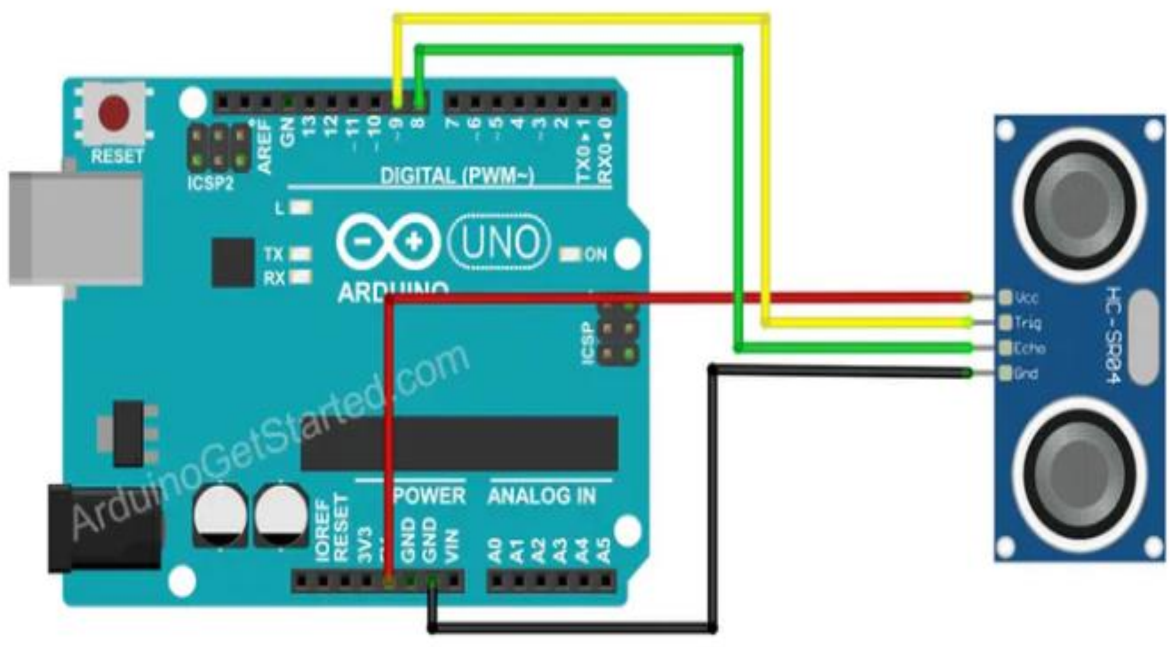


Figure 96: Ultrasonic Sensor Testing Diagram

We will run two separate and simple tests to test the accuracy of the ultrasonic sensor. The testing descriptions for the first test can be found in the table below along with the results of the test.

Test Identifier	Ultrasonic Sensor Test #1
Objective	Ultrasonic Sensor Accuracy Test
Description	Place an object six inches away from sensor and read the distance from the Ultrasonic Sensor
Expected Results	The Ultrasonic Sensor should give a reading of a distance of at 90% accuracy

Table 49: Ultrasonic Sensor Test #1 Specifications

The figure below shows the testing set up for the ultrasonic sensor. This set up includes a solid mounted platform for the sensor, a distance measurement from the edge of the sensor, and a flat object to reflect the sensors waves back to itself.

Figure 97: Ultrasonic Sensor Test #1 Setup

A second test will also need to be run with an object placed at a further distance to ensure that the sensor is still accurate enough to give us a good reading and those testing descriptions can be found in the table below. In order for this sensor to ensure the user that it is not going to have any issues when using this product, it should be able to detect up to 400cm which way overkill for what we want to accomplish with this product.

Test Identifier	Ultrasonic Sensor Test #2
Objective	Ultrasonic Sensor Accuracy Test
Description	Place an object twelve inches away from sensor and read the distance from the Ultrasonic Sensor
Expected Results	The Ultrasonic Sensor should give a reading of a distance of at 90% accuracy

Table 50: Ultrasonic Sensor Test #1 Specifications

The figure below shows the testing set up for Ultrasonic Sensor Test #2. This set up is identical to the set up for Test #1 with the exception of the distance being measured at 12 inches instead of 6 inches.

Figure 98: Ultrasonic Sensor Test #2 Setup

The results from both Test #1 and Test #2 are provided below, from screenshots of the serial output monitor. The Ultrasonic Sensor uses centimeters as its preset measurement. This is because the ultrasonic waves produced and read by the sensor are measured in meters per second. Instead of changing the measurement of the sensor, we can simply convert the outputs to inches.

```
Distance: 14 cm        Distance: 29 cm
Distance: 14 cm        Distance: 29 cm
Distance: 14 cm        Distance: 29 cm
Distance: 14 cm        Distance: 29 cm
Distance: 14 cm        Distance: 29 cm
```

Figure 99: Ultrasonic Sensor 6 inch and 12 inch Test Outputs

Using the outputs from test 1 and test 2 shown above, we can calculate the accuracy of the measurements. 1 cm is equivalent to 0.3937 inches. Test #1's distance in inches is equal to 5.512 inches and Test #2's distance in inches is equal to 11.417 inches. The table below provides the calculated accuracy of each test and states whether the test was passed or failed.

Test Identifier	Accuracy	Pass/Fail
Ultrasonic Sensor Test 1	91.9%	Pass
Ultrasonic Sensor Test 2	95.1%	Pass

Table 51: Ultrasonic Sensor Test Results

If you recall from the testing descriptions, we are looking for at least 90% accuracy in the ultrasonic sensor. The tests conclude that the sensor accommodates our required accuracy. Some points to consider when analyzing the accuracy of the ultrasonic sensor is the rounding factor of the test code used. The output was shown to the nearest whole unit, without decimal values. This reduces the perceived accuracy of the sensor. The precision of measurement can be adjusted in the code but was deemed unnecessary for this use case. The reason a tight tolerance of accuracy is not demanded from the ultrasonic sensor is because the sensor is going to be acting as a binary signal. Specifically, the sensor will be set to low while food is directly in front of it. When the food gets too low, a hardcoded threshold will be exceeded, the sensor will sense a distance equal to seeing the other side of the hopper, and the sensor will set its communication pin to high. This will alert the system of low food. In short, the testing results for the ultrasonic sensor meet our expectations and will suffice for our use case in the pet feeder.

6.3 Weight Sensor Testing Plans/Results

The weight sensor is going to be used for food weight detection in this project. It will be mounted underneath a food bowl. We want to have it so once a certain weight is detected on the sensor, the motor will then stop turning and thus stop dispensing any food. It is particularly important for the weight sensor to be accurate because we do not want the motor to keep turning and thus overfeeding the pet. The wiring diagram for how the weight sensor should be hooked up can be found in the screenshot below.

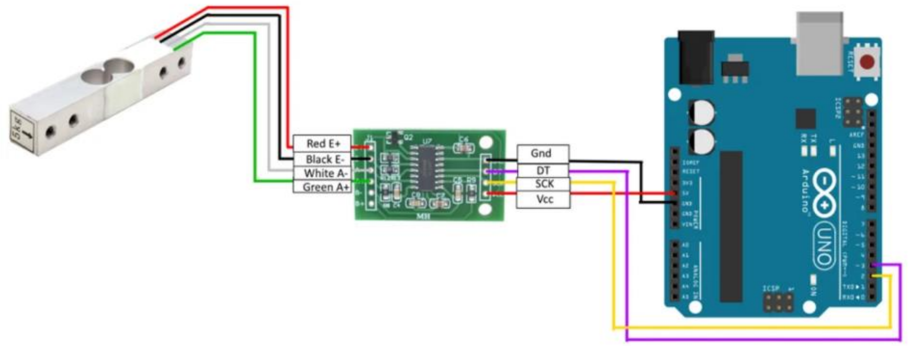


Figure 100: Load Cell Testing Diagram

Most load cells will not have an identical output due to the extreme sensitivity of the circuitry that measures the voltage differential in the load cell. Because of this, the load cell needs to be calibrated before we can start to test its accuracy. To calibrate the load cell along with the HX711 ADC converter, we need to run a calibration code from Arduino and find the calibration number for our specific sensor, which can then be entered into the code to measure weight and test the accuracy. To calibrate the load cell, we first hooked it up to the Uno following the wiring diagram shown above, with the Uno being powered by a laptop computer. Next, we uploaded the calibration code to the Uno, and used a serial monitor on the laptop computer to display the output from the load cell, through the Uno. To calibrate the load cell, we needed to use a known weight. At the time of calibrating, we did not have access to a precision scale, so we used an iPhone as our known weight because its exact weight can be found online. The program was used to adjust the calibration factor until the output from the load cell matched the exact weight of the iPhone, at 0.43 pounds. The calibration factor that was settled on the produce a precise measurement using our known weight is 234500. This number is used in the measurement code to calibrate the scale with the code to ensure an accurate and precise reading. We used pounds for the calibration because the code was preset to that, but the units can be easily converted in the final code.

The description for the test can be found in the table below. This test is designed to demonstrate accuracy across a range of weights, using one lighter weight and one heavier weight. This will ensure the scale is measuring in a linear fashion, which is important for our case because the weight in the bowl needs to be measured at any point of given user input and not just at one specific weight.

Test Identifier	Weight Sensor Test
Objective	Weight Sensor Accuracy Test
Description	Using two known weights, place the lesser of the two on the scale and record the measurement, then place the heavier weight on the scale and record.
Expected Results	The Weight Sensor should give a reading of a weight at 95% accuracy

Table 52: Load Cell Test #1 Specifications

The test was completed successfully and the testing configuration along with the measured objects can be seen in the figures below. The accompanying serial outputs can also be seen depicted next to their set up pictures. Both measured weights fall well within the expected 95% accuracy range, with the I phone's known weight being 0.43 lbs. and the MacBook's known weight being 3.02 lbs.

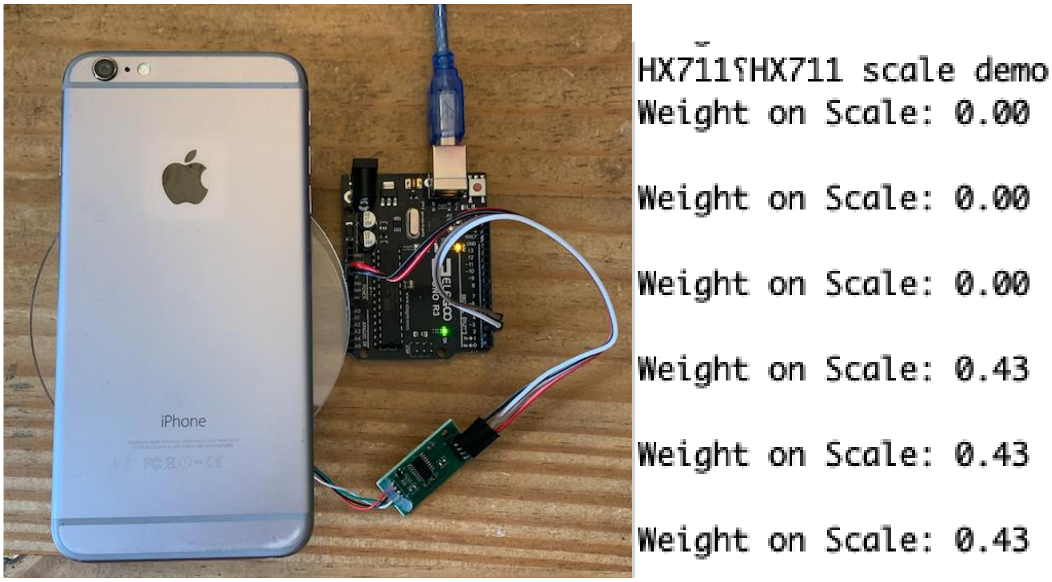


Figure 101: Load Cell Test #1 Setup and Results

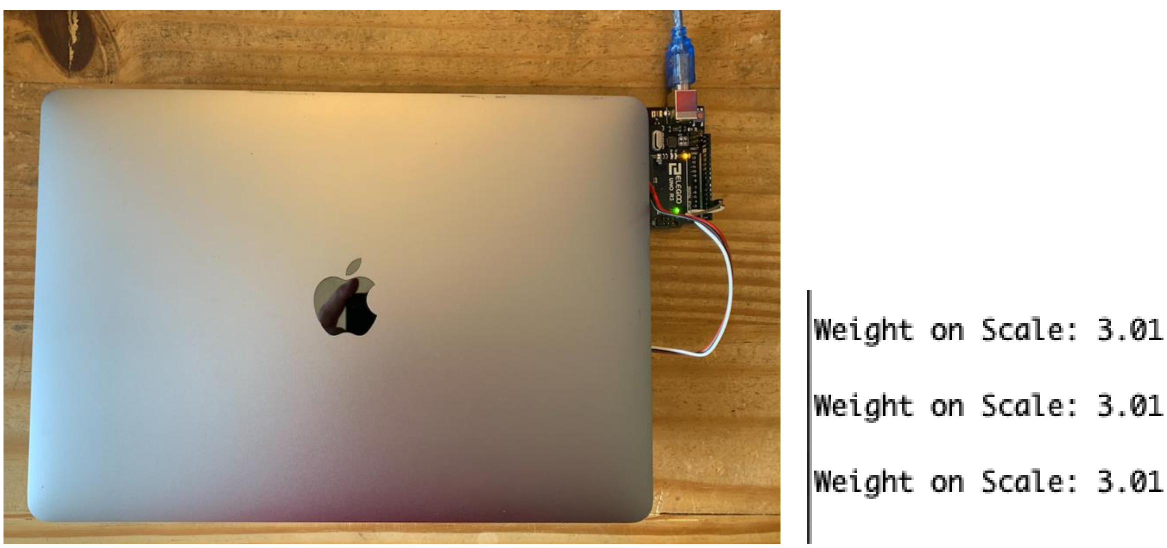


Figure 102: Load Cell Test #2 Setup and Results

This testing procedure ensured the load cell weight sensor along with the HX711 ADC amplifier and converter operates successfully and as expected. The benefit of completing this individual test is to obtain an accurate calibration number, so when it comes time to integrate the weight sensing module together with the rest of the components, it will be ready to go and be easily adapted with some code to work along with the rest of the components in the unit.

6.4 Motor Testing Plans/Results

The motor is obviously going to be used to turn the auger for the pet feeder and dispense food into the bowl. The motor needs to be able to turn and stop accurately in order to make sure the proper amount of food is dispensed. The wiring diagram can be found below with a motor driver and the external voltage source of 12V in order to power the motor.

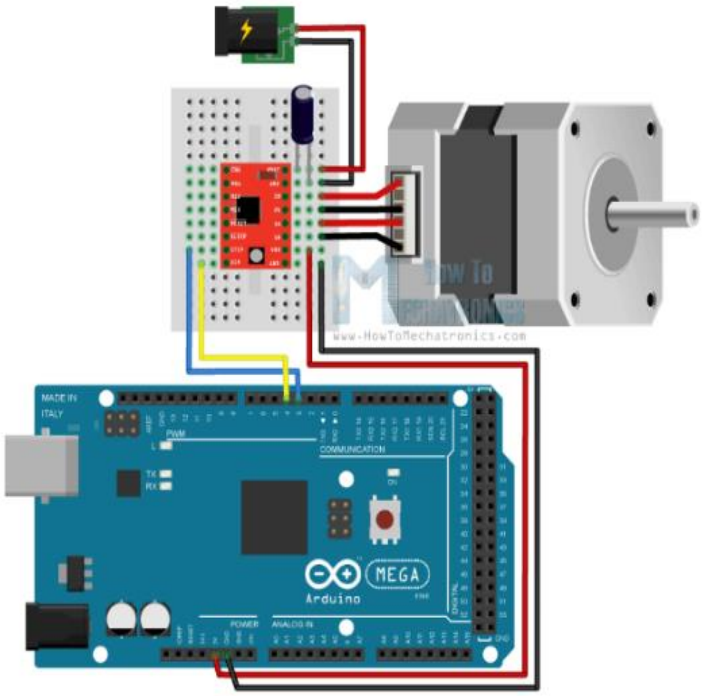


Figure 103: NEMA Motor Testing Diagram

The figure below depicts the testing set up used to perform the following tests to ensure proper operation of the stepper motor and driver. This testing set up utilized the 12V SLA battery that will be used to power the pet feeder. Voltage regulation was not necessary for this set up due the motor driver circuitry being able to accept an input voltage range of 8V to 35V. The stepper motor driver was able to be connected directly to the battery with the addition of a decoupling capacitor placed from the positive to negative connections to the driver motor, from the battery. This protects the driver from any unexpected voltage spikes or shorts. One additional piece of wiring not included in the wiring diagram above is the “Sleep” and “Reset” pins were both set to high (5V).

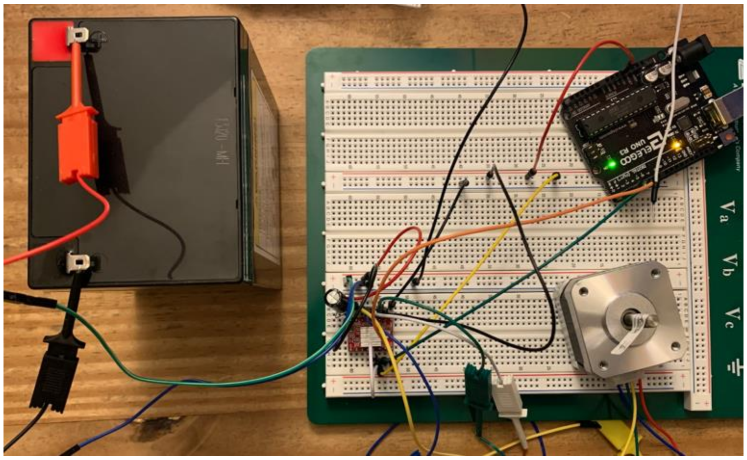


Figure 104: Stepper Motor and Driver Testing Set-up

We will run two tests, one that will turn the motor approximately two times before stopping and another to do one turn in the counterclockwise direction and then stop and do a turn in the clockwise direction and then stop. The testing descriptions for the first test can be found in the table below.

Test Identifier	NEMA Motor Test #1
Objective	NEMA Accuracy Test
Description	The stepper motor shall do one full revolution in a predetermined direction and then stop while not going past the starting point
Expected Results	The NEMA should be 100% accurate and do one turn and then stop turning

Table 53: NEMA Motor Test #1 Specifications

With the knowledge that each step in the Nema 17 motor is 1.8 degrees, we can calculate how many steps need to be commanded to the motor to complete one full circle. One revolution is 360 degrees, so 360/1.8 gives 200 steps. In the simple code used to test the stepper motor, the Arduino sends command for 200 steps. This should result in an accurate and precise full rotation of the motor spindle. It was difficult to determine the accuracy of rotation when attempting to view the bare motor spindle, so a sticker was attached to the spindle as depicted in the figure below. This allowed a starting point to be determined, and to compare to the ending point.

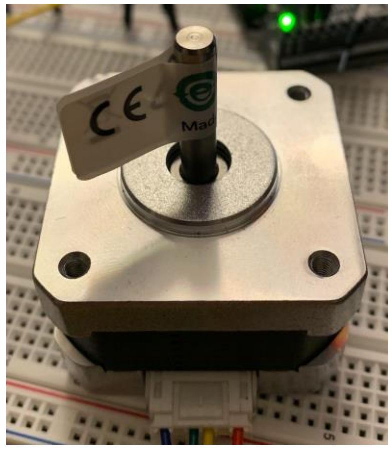


Figure 105: Nema 17 Motor with Marking Sticker

A second will be run as previously mentioned. This time we will have the motor turn once in the counterclockwise direction then turn once in the clockwise direction and eventually stop at the original starting point. The second test descriptions can be found in the table below.

Test Identifier	NEMA Motor Test #2
Objective	NEMA Accuracy Test
Description	The stepper motor shall do one full revolution in a counterclockwise direction then one full turn in the clockwise direction and return to the original starting point
Expected Results	The NEMA should be 100% accurate and do one turn and then stop turning

Table 54: NEMA Motor Test #2 Specifications

To accomplish this testing procedure, the motor was first run as it was in test 1, then the direction pin on the motor driver was set to high (it was set to low for test 1), this reverses the stepping direction of the motor, and a command to step 200 times is sent in the same statement.

The testing results for the motor can be found in the table below. There is no way to accurately show the results without taking a video of the motor so instead we have opted to put the results into a table.

Test identifier	Accuracy	Pass/Fail
NEMA Motor Test 1	100%	Pass
NEMA Motor Test 2	100%	Pass

Table 55: NEMA Motor Testing Results

As seen in the table above, both tests were successful. These tests proved to us that the functionality of both the A4988 motor driver along with the 17HS2408 stepper motor will work in unison to accomplish our desired task in the complete unit. The rotation of the stepper motor is fully controllable, accurate, as well as reversible. The motor driver is capable of running directly off of our 12V power supply and the reference voltage to adjust the output to the stepper motor is easily adjustable.

6.5 Display Testing

To ensure that the display is working correctly with the microcontroller/host, a test plan must be developed. In this section a plan to test the display will be conducted. Below are how the connections will be routed to test the display functionality:

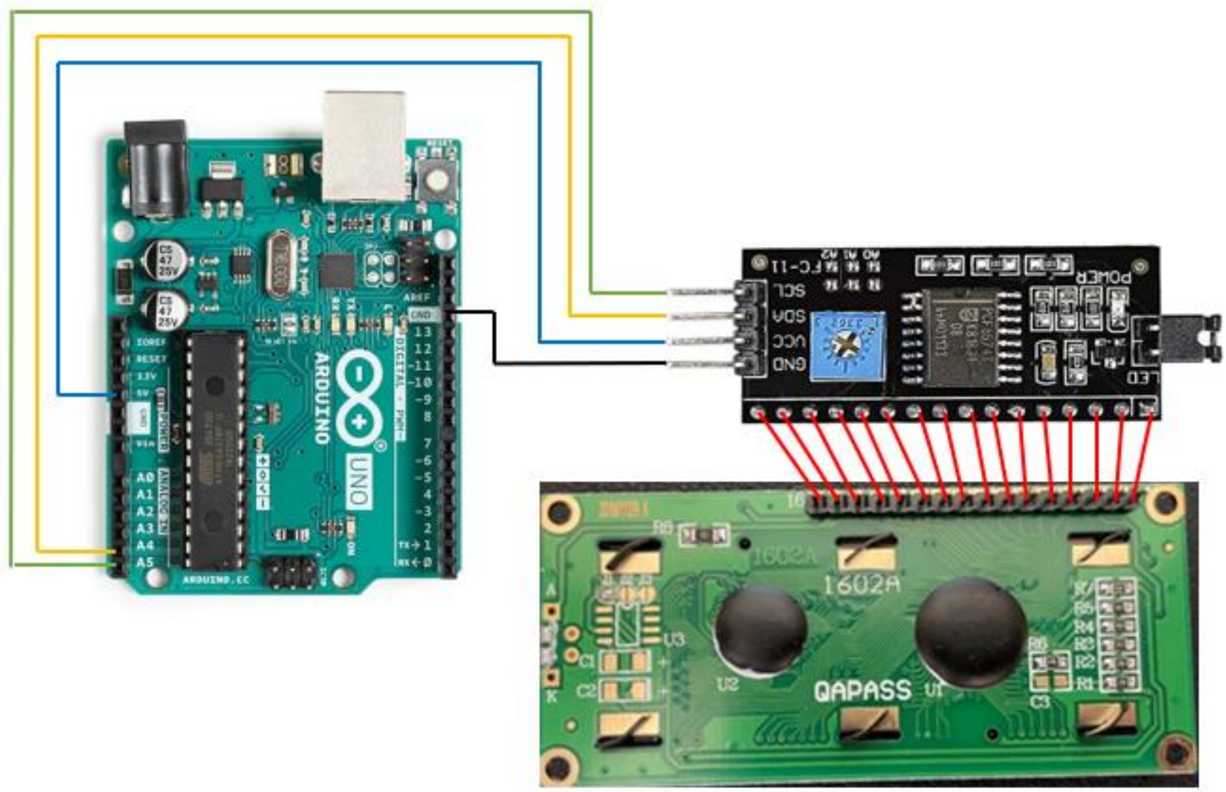


Figure 106: LCD Testing Diagram

The setup above will allow to test the functionality of the display to be tested. A simple message of "Welcome to the Pet Feeder" will be sent to check that the liquid crystal display is working as intended. This will also allow for proper tuning of the display brightness and contrast. With the blue potentiometer located on the daughter board of the LCD display, the proper brightness/contrast will be set. The main goal with the addition of this display (recall that this is a stretch goal for the project) is to allow the user to view the status of the microcontroller. This test plan will allow for testing of this functionality.

6.6 Final Design Testing

Below is the final testing with our final design of our Alexa Automated Pet Feeder. The following sections will go over four of the specifications and the results from the testing that was done.

6.6.1 Reservoir Capacity Testing

Reservoir Capacity specification states, that it must be able to hold 3 days' worth of food, or 5 cups. As seen in the image below, it is only halfway full when 5 cups of food were added into the bowl. The pet feeder could hold approximately 10-11 cups of food when completely filled. Therefore, this specification was all passed successfully.

Figure 107: Reservoir Capacity Testing Results

6.6.2 On Demand Feeding Testing

Specification states that the pet feeder shall start dispensing food with 20 seconds after acknowledging the command from Alexa. As seen in the photo, it took approximately 12.9 seconds for the motor to begin turning/dispensing after telling Alexa to feed the pet. This specification was passed successfully.

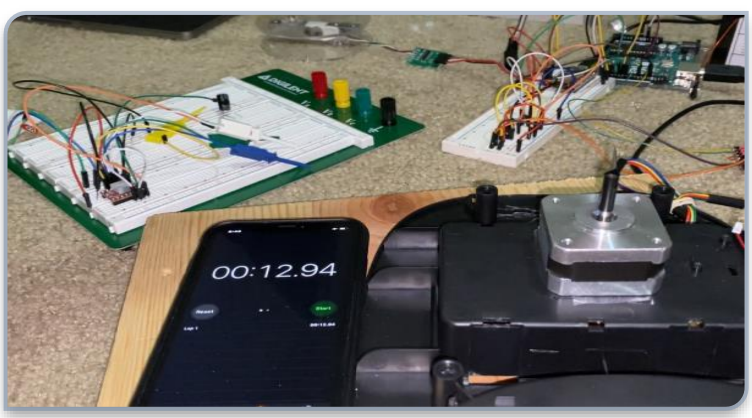


Figure 108: On Demand Feeding Testing Results

6.6.3 Food Reservoir Status Testing

Reservoir Sensor specification states. "Pet Feeder shall alert the user, via Alexa, that the reservoir is low once the sensor sees a distance greater than 15cm." For testing purposes, we placed the ultrasonic sensor face up to emulate the sensor seeing a distance greater than 15cm. After this, we then asked Alexa to dispense food and it gave us an error stating, "Food Reservoir is low, please refill to dispense food." Therefore, the Reservoir Sensor Status specification was passed successfully.

Figure 109: Food Reservoir Status Testing Results

6.6.4 Weight Sensor Status Testing

The weight sensor status specification says that the pet feeder shall stop dispensing once a weight greater than or equal to the threshold weight. For testing purposes, we set a maximum weight of 80g. Once the motor began to turn, we placed an 80g weight on the load cell and we saw the motor stop instantly. The weight sensor status specification was passed successfully.

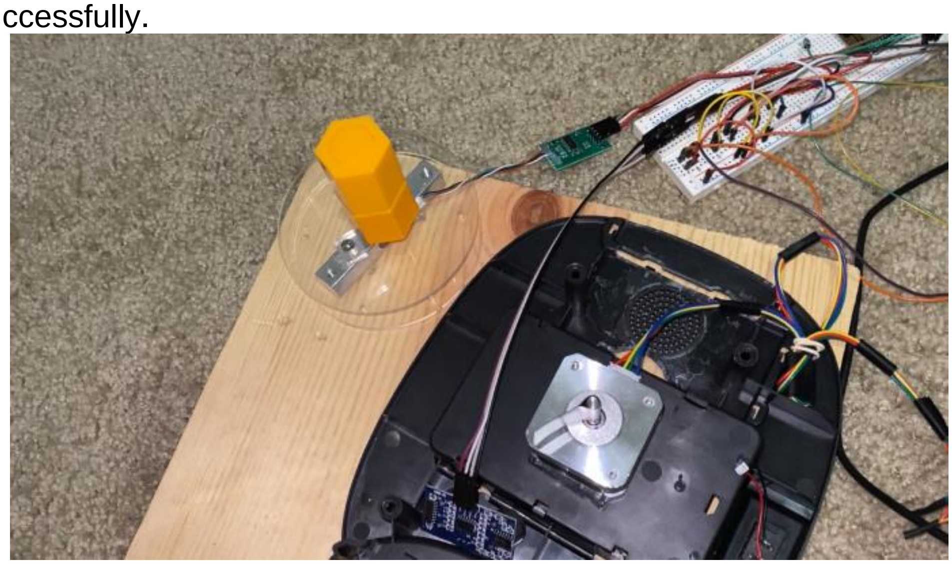


Figure 110: Weight Sensor Status Testing Results

7.0 Communication Platforms

Throughout the course of this project, one of the main goals of the group was to maintain high levels of communication. This was especially difficult with the Health Constraints faced due to COVID-19 this semester. For these reasons the group decided to use Discord and Zoom for the bulk of the communication throughout the project. These platforms will be discussed here. Also discussed in this section is OneDrive, the platform used for document development throughout the project.

7.1 Discord

Discord is an extremely useful online communication tool that has a lot of benefits compared to other platforms such as skype. Discord operates based on servers, for the Pet Feeder Project, a server was created. Within the server there are several text channels and voice channels. This allowed the group to stay organized among different topics surrounding the Pet Feeder. For example, the main text channels used were general, session-planning, and off-topic. The general channel was used to discuss overarching topics of the pet feeder. These can pertain to any aspect of the Pet Feeder development process. On the other hand, session planning channel was used to discuss meeting times as well as deadlines. Finally, off-topic allowed for discussion of stretch goals that could be added to the pet feeder. The voice channels allowed for easy deployment of weekly meetings as well as impromptu meetings. Here team members can easily chat to discuss Pet Feeder development and progress. Moreover, it is easy to share the screen for other viewers and/or a camera feed. Overall, the great communication among the team members of the project can be attributed to the ease of use and deployment of Discord.

7.2 Zoom

The other great platform that the group used to communicate was Zoom. This platform allowed for easy discussion of group progress. It is very easy to deploy, as simply sharing a link can allow anyone to access the meeting. Moreover, the platform has camera feed, screen share, and chat features built in. All in all, it allowed for more formal meetings with group members, since it allowed for viewing all the camera feeds simultaneously. This allowed for a more in-person feel to the conversation. This platform was very helpful throughout project development and organization.

7.3 OneDrive

OneDrive is a service available from Microsoft. It is a cloud platform that allows for easy document sharing and collaboration. This service is actually available for free to UCF students, so it was very easy to deploy. This allowed the team to create a folder on the cloud that all team members had access to at all times of the day. Moreover, it allowed for team collaboration as all team members could be on the document(s) and view the changes being made by the other members live. Team members could also highlight text and leave comments for other members to evaluate. Finally, having all the documents backed up to the cloud ensured that no work would be lost, and the automatic save feature allowed for constant backups to be uploaded to the cloud. Overall, OneDrive allowed for the group to work in unison throughout the development of the Pet Feeder.

8.0 Project Summary and Conclusion

To discuss some concluding remarks, we would like to talk about what has been accomplished and our projections for the future of the Alexa Enabled Pet Feeder. As a group, we have completed extensive research on individual components as well as the system as a whole. It is imperative that the correct components were selected to be used in this project to avoid component related issues down the line. While completing research for the most ideal components, we were required to adhere to a strict budget due to our collectively limited resources. So far, we have been able to remain well within our budget. A final expenditure report has not been produced because we have not yet ordered a final PCB design nor built the housing for the unit. The reasoning for this is also budget related, we want to ensure everything will fully function together as a whole before completing these last few, heavily cost reliant steps.

As for overall system function, we have purchased the key components and tested them individually using prototype codes that will be easily adapted and pieced together to work in unison. We have a general overview of the flow of the program, along with how the communication protocols between the components will be interpreted and handled.

Through careful deliberation and consideration of all aspects apparent to us, and even those that took a more careful eye to see, we are confident in our trajectory towards the assembly stage and completion of the Alexa Enabled Pet Feeder. Working as a team has been a great advantage. We all bring different skills to the table along with various life experiences that have proved to be beneficial in developing this project in a progressive and positive direction.

9.0 References

[1] "ATmega328P-PU DIP-28 Chip with Arduino Uno Bootloader and Addicore Pinout Sticker." *Www.addicore.com*, www.addicore.com/ATmega328P-PU-Arduino-Uno-Bootloader-Addicore-p/132.htm.

[2] "C8051F98x." *C8051F98x - Silicon Labs - 8051 Family | Online Catalog | DigiKey Electronics*, www.digikey.com/catalog/es/partgroup/c8051f98x/20892.

[3] "EAGLE: PCB Design And Electrical Schematic Software." *Autodesk*, www.autodesk.com/products/eagle/overview?plc=F360&term=1-YEAR&support=ADVANCED&quantity=1.

[4] "An Easier and PowerfulOnline PCB Design Tool." *EasyEDA*, easyeda.com/.

[5] "Infrared Proximity Sensor - Sharp GP2Y0A21YK." *SEN-00242 - SparkFun Electronics*, www.sparkfun.com/products/242.

[6] Last Minute Engineers. "In-Depth: What Is RFID? How It Works? Interface RC522 with Arduino." *Last Minute Engineers*, Last Minute Engineers, 18 Dec. 2020, lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/.

[7] "Linear vs. Switching Regulators." *Renesas*, www.renesas.com/us/en/products/power-power-management/linear-vs-switching-regulators.

[8] "MSP430G2553." *MSP430G2553 Data Sheet, Product Information and Support | TI.com*, www.ti.com/product/MSP430G2553.

[9] "MSP432P401R." *MSP432P401R Data Sheet, Product Information and Support | TI.com*, www.ti.com/product/MSP432P401R.

[10] "PETKIT ® 'Element' Wi-Fi Enabled Smart Pet Food Container Feeder." *Petkit.us*, petkit.us/products/petkit-element-wi-fi-enabled-smart-pet-food-container-feeder?_pos=4&_sid=25660a8d3&_ss=r.

[11] "Petnet SmartFeeder." *Petnet.io*, petnet.io/smartfeeder.

[12] "RFID Card Reader - Serial." *Parallax*, 13 Apr. 2021, www.parallax.com/product/rfid-card-reader-serial/.

[13] "Ultrasonic Distance Sensor - HC-SR04." *SEN-15569 - SparkFun Electronics*, www.sparkfun.com/products/15569.

[14] "Ultrasonic Range Finder - HRLV-MaxSonar-EZ0." *SEN-11307 - SparkFun Electronics*, www.sparkfun.com/products/11307.

[15] “Ultrasonic Sensor HC-SR04 with Arduino Tutorial.” *Arduino Project Hub*, create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6.

[16] “WEBENCH- Power Designer.” *Power Designer*, webench.ti.com/power-designer/switching-regulator.

[17] “Alexa Connected Devices Official Site: Create Voice-First Devices.” *Amazon (Alexa)*, developer.amazon.com/en-US/alexa/devices/connected-devices. Accessed 27 Apr. 2021.

[18] “Arduino - Wire.” *Arduino*, www.arduino.cc/en/reference/wire. Accessed 27 Apr. 2021.

[19] “GPIO - Raspberry Pi Documentation.” *Raspberry Pi Documentation*, www.raspberrypi.org/documentation/usage/gpio/#:%7E:text=A%2040%2Dpin%20GPIO%20header,a%20wide%20range%20of%20purposes. Accessed 27 Apr. 2021.

[20] “Jetson AGX Xavier Developer Kit.” *NVIDIA Developer*, 5 Jan. 2021, developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit.

[21] “Jetson Nano 2GB Developer Kit.” *NVIDIA Developer*, 29 Oct. 2020, developer.nvidia.com/embedded/jetson-nano-2gb-developer-kit.

[22] “---.” *NVIDIA Developer*, 29 Oct. 2020, developer.nvidia.com/embedded/jetson-nano-2gb-developer-kit.

[23] “Jetson Nano 2GB Developer Kit User Guide.” *NVIDIA Developer*, 12 Nov. 2020, developer.nvidia.com/embedded/learn/jetson-nano-2gb-devkit-user-guide.

[24] “Jetson Nano Developer Kit.” *NVIDIA Developer*, 14 Apr. 2021, developer.nvidia.com/embedded/jetson-nano-developer-kit.

[25] “Jetson Xavier NX Developer Kit.” *NVIDIA Developer*, 19 May 2020, developer.nvidia.com/embedded/jetson-xavier-nx-devkit.

[26] Phongchit, Nikko. “What Is Baud Rate & Why Is It Important?” *Setra*, www.setra.com/blog/what-is-baud-rate-and-what-cable-length-is-required-1. Accessed 27 Apr. 2021.

[27] “Techdoc-Template.” *Amazon (Alexa)*, developer.amazon.com/en-US/docs/alexa/devconsole/about-the-developer-console.html. Accessed 27 Apr. 2021.

[28] “---.” *Amazon (Alexa)*, developer.amazon.com/en-US/docs/alexa/alexa-voice-service/get-started-with-alexa-voice-service.html. Accessed 27 Apr. 2021.

[29] “---.” *Amazon (Alexa)*, developer.amazon.com/en-US/docs/alexa/alexa-voice-service/register-a-product-with-avs.html. Accessed 27 Apr. 2021.

[30] "---." *Amazon (Alexa)*, developer.amazon.com/en-US/docs/alexa/alexa-voice-service/build-the-avs-device-sdk.html. Accessed 27 Apr. 2021.

[31] "---." *Amazon (Alexa)*, developer.amazon.com/en-US/docs/alexa/alexa-voice-service/get-a-refresh-token.html. Accessed 27 Apr. 2021.

[32] "---." *Amazon (Alexa)*, developer.amazon.com/en-US/docs/alexa/avs-device-sdk/raspberry-pi.html. Accessed 27 Apr. 2021.

[33] The Raspberry Pi Foundation. "Buy a 3 Model A+ –." *Raspberry Pi*, www.raspberrypi.org/products/raspberry-pi-3-model-a-plus. Accessed 27 Apr. 2021.

[34] ---. "Buy a 3 Model B+ –." *Raspberry Pi*, www.raspberrypi.org/products/raspberry-pi-3-model-b-plus. Accessed 27 Apr. 2021.

[35] ---. "Buy a 3 Model B –." *Raspberry Pi*, www.raspberrypi.org/products/raspberry-pi-3-model-b. Accessed 27 Apr. 2021.

[36] ---. "Buy a 4 Model B –." *Raspberry Pi*, www.raspberrypi.org/products/raspberry-pi-4-model-b. Accessed 27 Apr. 2021.

[37] "Uart - NodeMCU Documentation." *NodeMCU*, nodemcu.readthedocs.io/en/release/modules/uart. Accessed 27 Apr. 2021.

[38] "Attention Required! | Cloudflare." *What Is Layer 7? | How Layer 7 of the Internet Works*, www.cloudflare.com/learning/ddos/what-is-layer-7. Accessed 27 Apr. 2021.

[39] "Automation Using Amazon Echo Alexa and NodeMCU Without IFTTT." *Hackster.Io*, 20 Aug. 2017, www.hackster.io/shiva-siddharth/automation-using-amazon-echo-alexa-and-nodemcu-without-ifttt-08b6c8.

[40] Ewald, Wolfgang. "Using the ATmega328P Standalone." *Wolles Elektronikkiste*, 5 Dec. 2020, wolles-elektronikkiste.de/en/using-the-atmega328p-standalone.

[41] "Flashing Arduino Bootloader on Atmega328p Microcontroller." *Electronics-Lab.Com*, 24 Sept. 2018, www.electronics-lab.com/project/installing-the-arduino-bootloader-on-the-atmega328p-microcontroller.

[42] "NodeMCU SPI with Arduino IDE | NodeMCU." © *2018 ElectronicWings*, www.electronicwings.com/nodemcu/nodemcu-spi-with-arduino-ide#:%7E:text=NodeMCU%20based%20ESP8266%20has%20Hardware,pins%20available%20for%20SPI%20communication.&text=We%20can%20use%20the%20Hardware%20SPI%20interface%20for%20user%2Dend%20applications. Accessed 27 Apr. 2021.

[43] “Programming Atmega328p Microcontroller with Arduino IDE.” *Electronics-Lab.Com*, 24 Oct. 2018, www.electronics-lab.com/project/programming-atmega328p-microcontroller-with-arduino-ide.

[44] Santos, Rui. “Alexa (Echo) with ESP32 and ESP8266.” *Random Nerd Tutorials*, 5 Mar. 2020, randomnerdtutorials.com/alexa-echo-with-esp32-and-esp8266.

[45] Squix78. “ESP8266 Module Comparison: ESP-01, ESP-05, ESP-12, ESP-201, Test Board And...” *Squix - TechBlog*, 22 Oct. 2017, blog.squix.org/2015/03/esp8266-module-comparison-esp-01-esp-05.html.

[46] “Techdoc-Template.” *Amazon (Alexa)*, developer.amazon.com/en-US/docs/alexa/avs-device-sdk/overview.html. Accessed 27 Apr. 2021.

[47] “Wifi - NodeMCU Documentation.” *WiFi Module*, nodemcu.readthedocs.io/en/release/modules/wifi. Accessed 27 Apr. 2021.

[48] Yida. “UART vs I2C vs SPI – Communication Protocols and Uses.” *Latest Open Tech from Seeed Studio*, 16 Nov. 2020, www.seeedstudio.com/blog/2019/09/25/uart-vs-i2c-vs-spi-communication-protocols-and-uses.

[49] “Amazon.Com: ML5-12 - 12 Volt 5 AH Rechargeable SLA Battery - Mighty Max Battery Brand Product: Electronics.” *Amazon*, 2020, www.amazon.com/Mighty-Max-Battery-ML5-12-Rechargeable/dp/B079ZCJYP3/ref=pd_bxgy_img_3/138-9210625-9689615?_encoding=UTF8&pd_rd_i=B079ZCJYP3&pd_rd_r=45ef6d70-e78c-4087-adff-ef75d5b7b445&pd_rd_w=1Vhj2&pd_rd_wg=VwIv4&pf_rd_p=bd257e40-8799-42ac-88d0-f76e0faf3886&pf_rd_r=4TRBC6896VX7TDVBXBN2&psc=1&refRID=4TRBC6896VX7TDVBXBN2.

[50] “Amazon.Com: Mroinge 6V / 12V 1A Fully Automatic Trickle Battery Charger/Maintainer for Automotive Vehicle Motorcycle Lawn Mower ATV RV Powersport Boat, Sealed Deep-Cycle AGM Gel Cell Lead Acid Batteries: Automotive.” *Amazon*, 2016, www.amazon.com/Mroinge-maintainer-Automotive-Motorcycle-powersport/dp/B01FTPZ6P4/ref=sr_1_5?crid=22BFQCKFTK43B&dchild=1&keywords=12v+1+amp+battery+charger+maintainer&qid=1618790152&sprefix=12v+1+amp+battery+charger%2Caps%2C177&sr=8-5.

[51] “Auger Drill Bit for Planting, 7"-30" Garden Plant Flower Bulb HEX Shaft Auger, Earth Auger Bit Post Hole Digger for 3/25" to 1/2" Hex Drive Drill (1.57x17.7): Amazon.Com: Industrial & Scientific.” *Amazon*, 2020, www.amazon.com/Planting-Garden-Flower-Digger-1-8x14-56/dp/B08832YY7J/ref=sr_1_8?crid=3V1GNV34N7VJD&dchild=1&keywords=2%2Binch%2Bauger%2Bbit%2Bfor%2Bdirt&qid=1617292963&sprefix=2%2Binch%2Bauger%2Caps%2C178&sr=8-8&th=1.

[52] Cavallo, Christian. “Stepper Motors vs. DC Motors - What’s the Difference?” *Thomasnet*, 2021, www.thomasnet.com/articles/machinery-tools-supplies/stepper-motors-vs-dc-motors.

[53] “Different Types of Batteries and Their Applications.” *Circuit Digest*, 12 Nov. 2020, circuitdigest.com/article/different-types-of-batteries.

[54] Engineering, Omega. “How to Use Load Cells for Weight Measurement.” *Https://Www.Omega.Com/En-Us/*, 2 July 2020, www.omega.com/en-us/resources/load-cell-application-examples.

[55] “How Much Food Should I Feed My Dog Explained by Charts.” *Dog Food Guru*, 14 Oct. 2020, dogfood.guru/how-much-food-should-i-feed-my-dog-explained-by-size-in-charts.

[56] Khan, Zak. “FAQ: What Are Stepper Drives and How Do They Work?” *Motion Control Tips*, 29 June 2015, www.motioncontroltips.com/faq-what-are-stepper-drives-and-how-do-they-work.

[57] “Lead Acid Battery: Working, Construction and Charging/Discharging.” *Circuit Digest*, 19 Dec. 2018, circuitdigest.com/tutorial/lead-acid-battery-working-construction-and-charging-discharging.

[58] “Lithiem-Ion Charge/Discharge Image.” *VectorStock*, 2021, www.vectorstock.com.

[59] “MakerHawk Digital Load Cell Weight Sensor HX711 AD Converter Breakout Module 5KG Portable Electronic Kitchen Scale for Arduino Scale: Amazon.Com: Industrial & Scientific.” *Amazon*, 2018, www.amazon.com/MakerHawk-Converter-Breakout-Portable-Electronic/dp/B07GQPV6C4/ref=pd_sbs_2?pd_rd_w=hFusN&pf_rd_p=527ea27c-adf6-4b67-9c5f-265eb29e0622&pf_rd_r=EV4H8XRGHZ21M6K29781&pd_rd_r=43438b95-8b2f-42bf-ae86-d5dd826f1cb0&pd_rd_wg=7ctp6&pd_rd_i=B07GQPV6C4&psc=1.

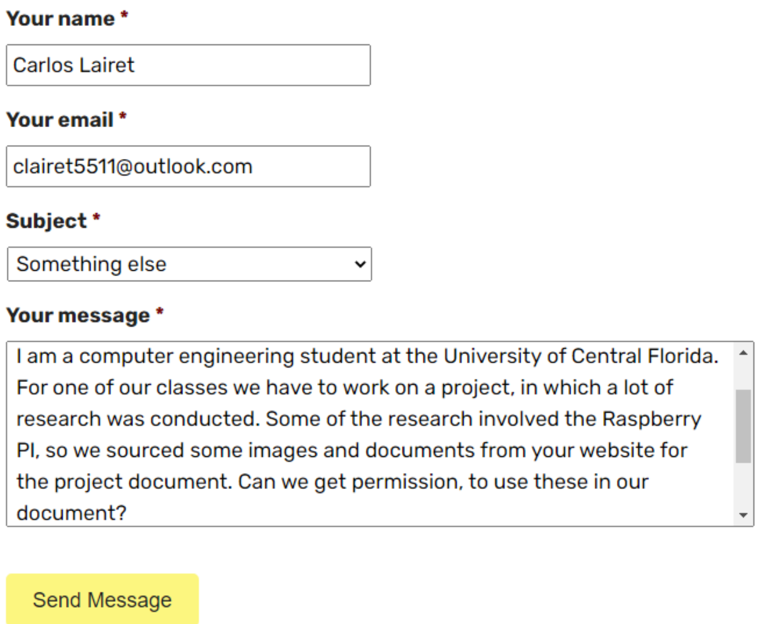
[60] Sirajul. “Electrical Blog - Electrical Engineering Blog.” *Electrical Blog*, 2021, eblogbd.com.

[61] The Organic Chemistry Tutor. “Battery Capacity - Amp-Hours, MAh, and Watt-Hours.” *YouTube*, 18 Dec. 2019, www.youtube.com/watch?v=5JZ8E7B4FAY.

[62] “Understanding Battery Amp Hours.” *Mkbattery*, 2021, www.mkbattery.com/blog/understanding-battery-amp-hours.

[63] “Overview FHD 1080p Webcam.” AUKEY Online, www.aukey.com/collections/wecam/products/fhd-webcam-1080p-live-streaming-camera-with-stereo-microphone. Accessed 27 Apr. 2021.

10.0 Copyright Permissions



Your name *

Carlos Lairet

Your email *

clairet5511@outlook.com

Subject *

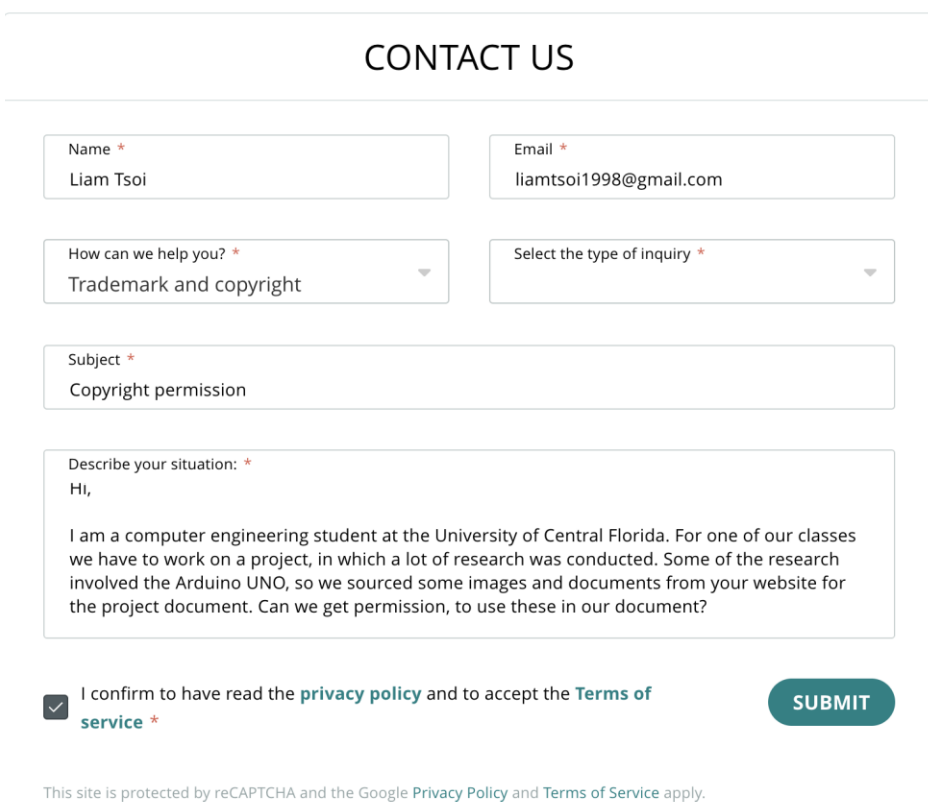
Something else

Your message *

I am a computer engineering student at the University of Central Florida. For one of our classes we have to work on a project, in which a lot of research was conducted. Some of the research involved the Raspberry PI, so we sourced some images and documents from your website for the project document. Can we get permission, to use these in our document?

Send Message

Figure 111 Permission for Raspberry Pi Support Documents and Pictures (Permission Approved)



CONTACT US

Name *

Liam Tsoi

Email *

liamtsoi1998@gmail.com

How can we help you? *

Trademark and copyright

Select the type of inquiry *

Subject *

Copyright permission

Describe your situation: *

Hi,

I am a computer engineering student at the University of Central Florida. For one of our classes we have to work on a project, in which a lot of research was conducted. Some of the research involved the Arduino UNO, so we sourced some images and documents from your website for the project document. Can we get permission, to use these in our document?

I confirm to have read the **privacy policy** and to accept the **Terms of service** *

SUBMIT

This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

Figure 112: Permission for Arduino UNO Support Documents and Pictures (Permission Pending)

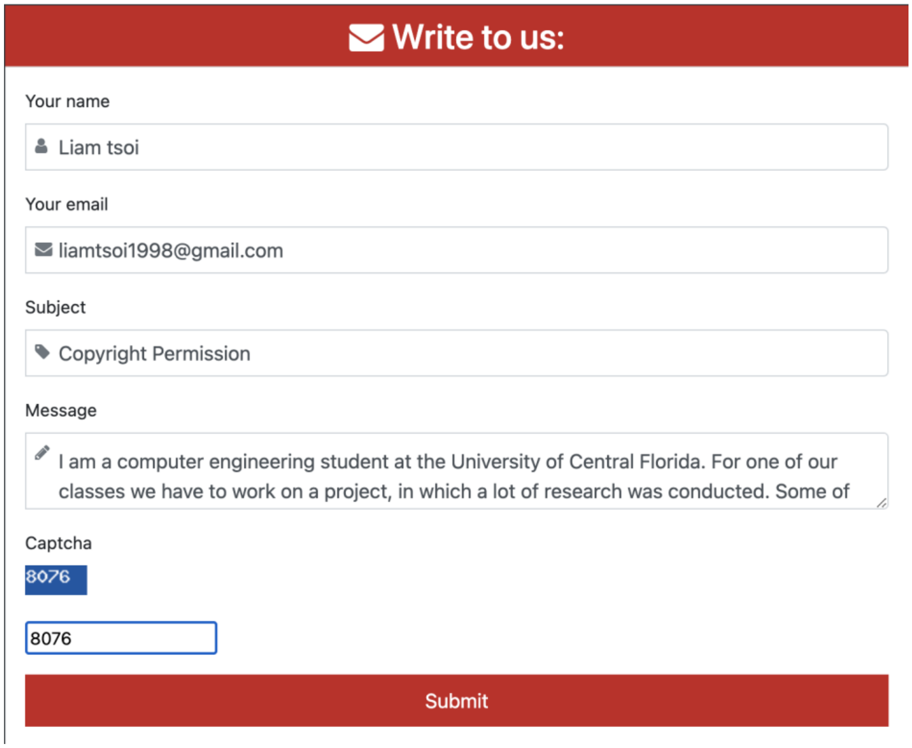


Figure 113 Permission for NodeMCU Support Documents and Pictures (Permission Approved)

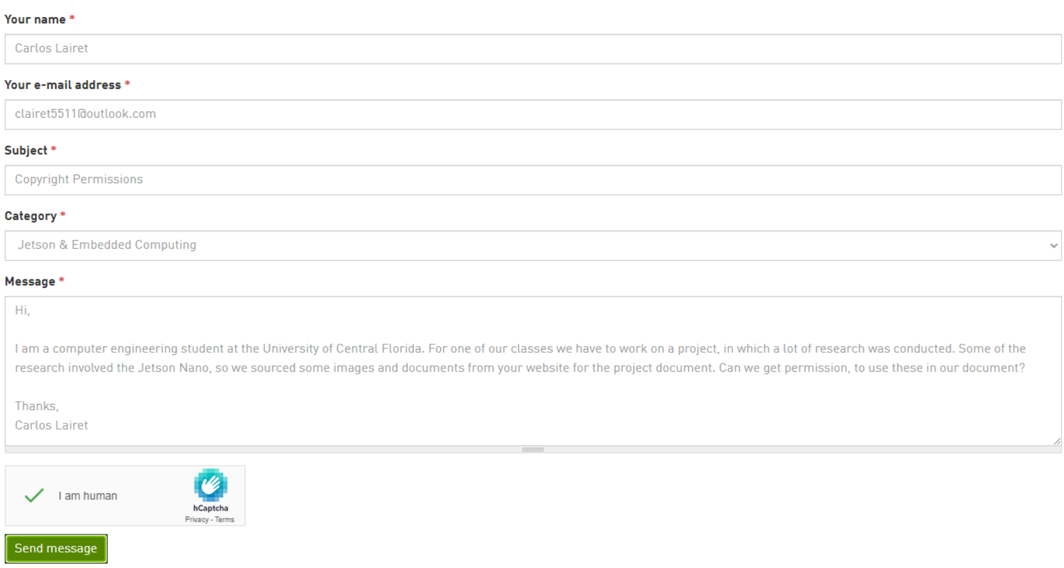
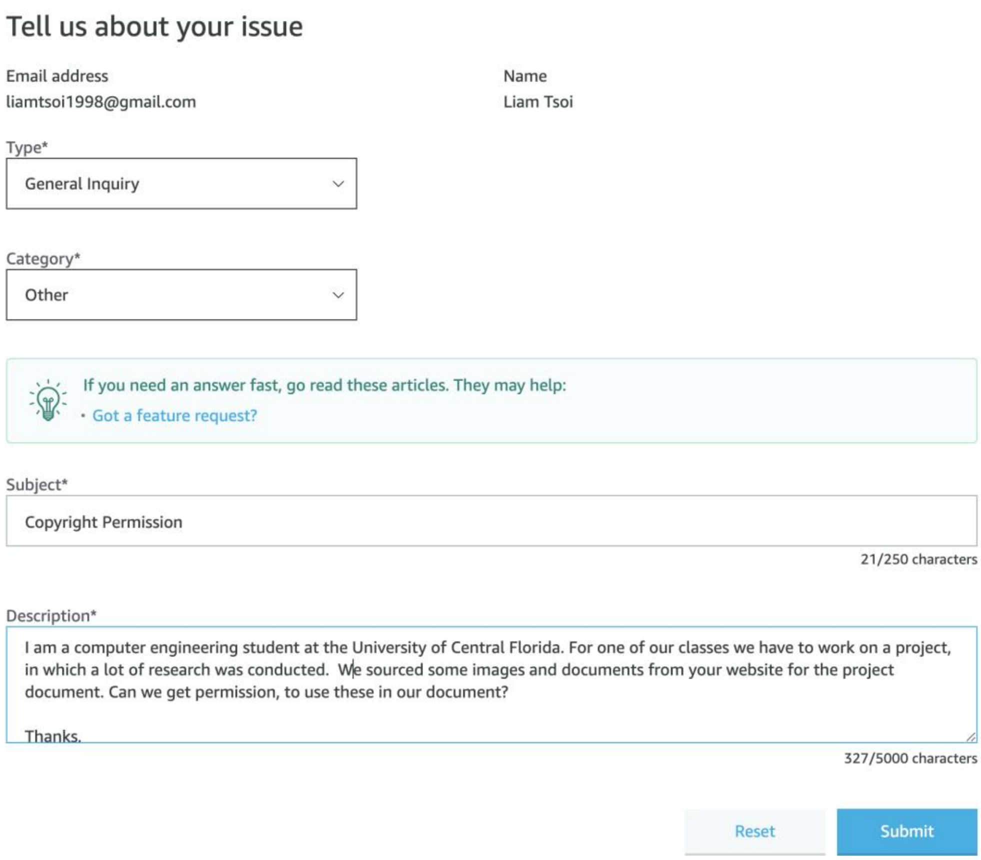


Figure 114 Permission for Jetson Nano Documents and Pictures (Permission Pending)



Tell us about your issue

Email address
liamtsoi1998@gmail.com

Name
Liam Tsoi

Type*
General Inquiry

Category*
Other

If you need an answer fast, go read these articles. They may help:
- Got a feature request?

Subject*
Copyright Permission

21/250 characters

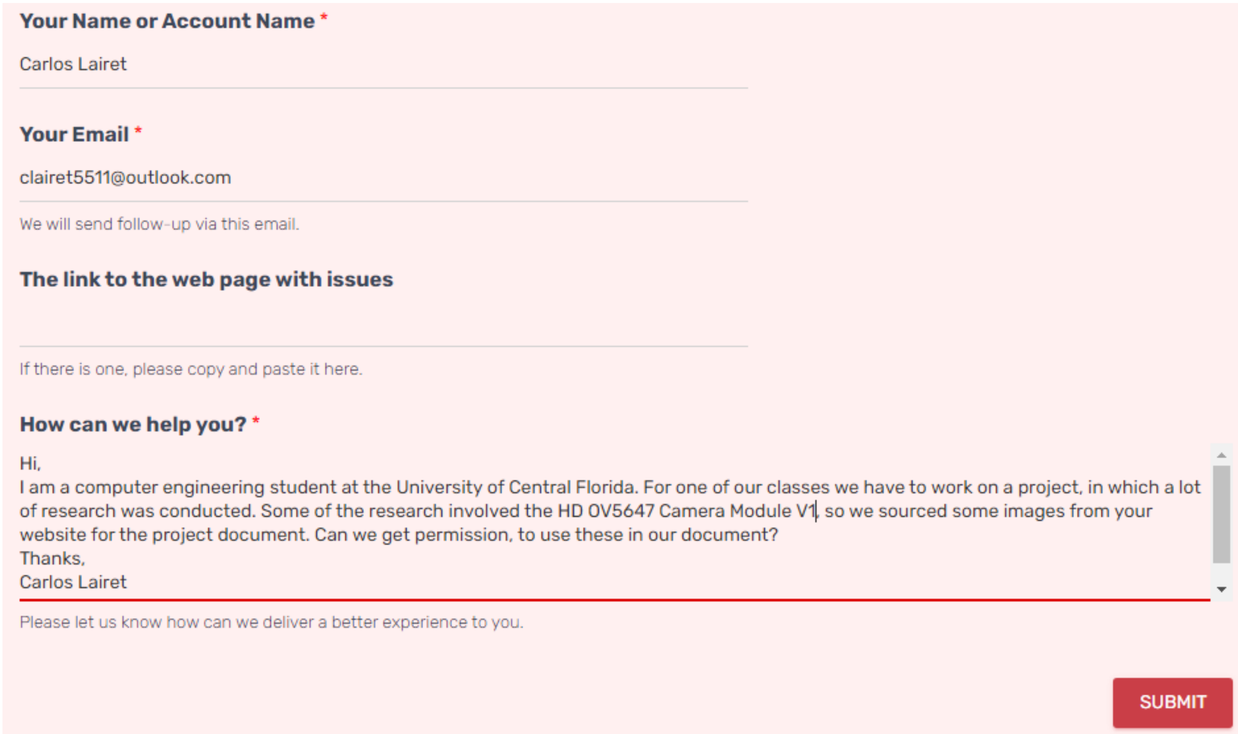
Description*
I am a computer engineering student at the University of Central Florida. For one of our classes we have to work on a project, in which a lot of research was conducted. We sourced some images and documents from your website for the project document. Can we get permission, to use these in our document?

Thanks,

327/5000 characters

Reset Submit

Figure 115 Amazon Developer Permission for Pictures (Permission Approved)



Your Name or Account Name *

Carlos Lairet

Your Email *

clairet5511@outlook.com

We will send follow-up via this email.

The link to the web page with issues

If there is one, please copy and paste it here.

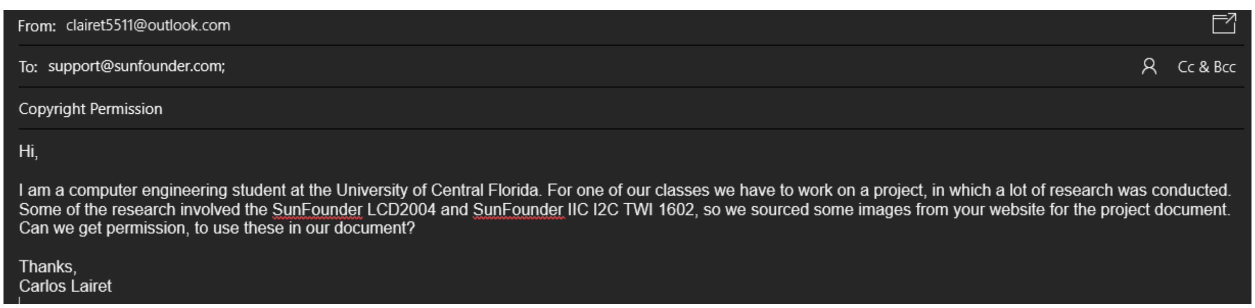
How can we help you? *

Hi,
I am a computer engineering student at the University of Central Florida. For one of our classes we have to work on a project, in which a lot of research was conducted. Some of the research involved the HD OV5647 Camera Module V1, so we sourced some images from your website for the project document. Can we get permission, to use these in our document?
Thanks,
Carlos Lairet

Please let us know how can we deliver a better experience to you.

SUBMIT

Figure 116 Permission for Arducam Camera Pictures (Permission Pending)



From: clairet5511@outlook.com

To: support@sunfounder.com;

Cc & Bcc

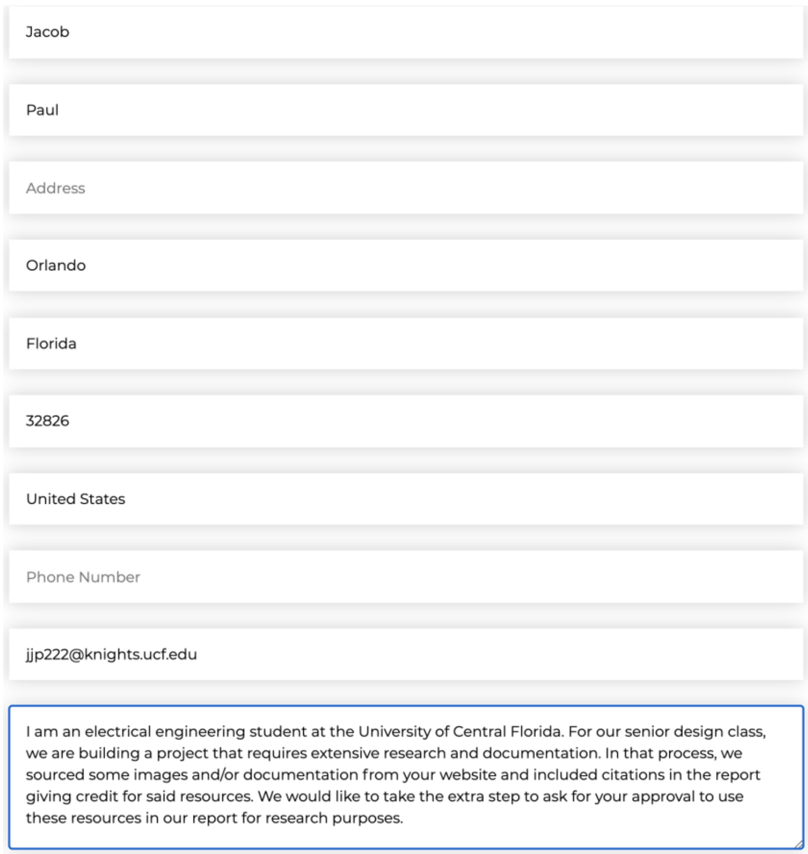
Copyright Permission

Hi,

I am a computer engineering student at the University of Central Florida. For one of our classes we have to work on a project, in which a lot of research was conducted. Some of the research involved the SunFounder LCD2004 and SunFounder IIC I2C TWI 1602, so we sourced some images from your website for the project document. Can we get permission, to use these in our document?

Thanks,
Carlos Lairet

Figure 117 Permission for SunFounder LCD Images (Permission Pending)


Jacob

Paul

Address

Orlando

Florida

32826

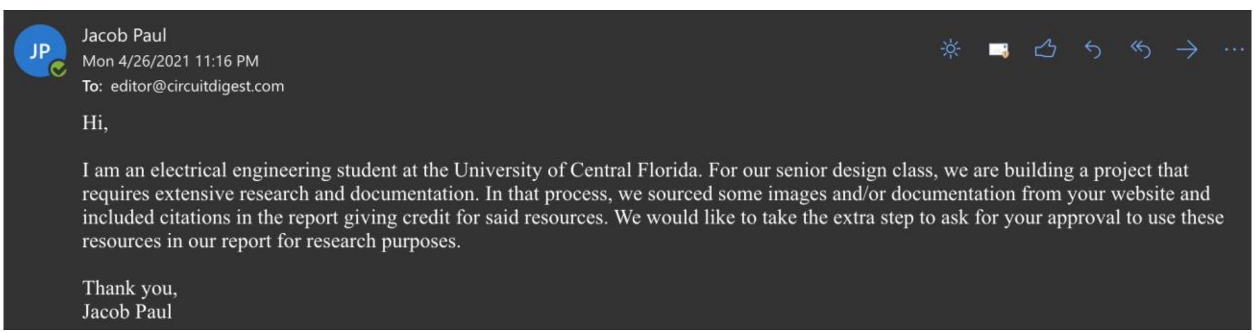
United States

Phone Number

jjp222@knights.ucf.edu

I am an electrical engineering student at the University of Central Florida. For our senior design class, we are building a project that requires extensive research and documentation. In that process, we sourced some images and/or documentation from your website and included citations in the report giving credit for said resources. We would like to take the extra step to ask for your approval to use these resources in our report for research purposes.

Figure 118: Permission for MK Powered Batteries Specs and Image (Permission Pending)


JP

Jacob Paul

Mon 4/26/2021 11:16 PM

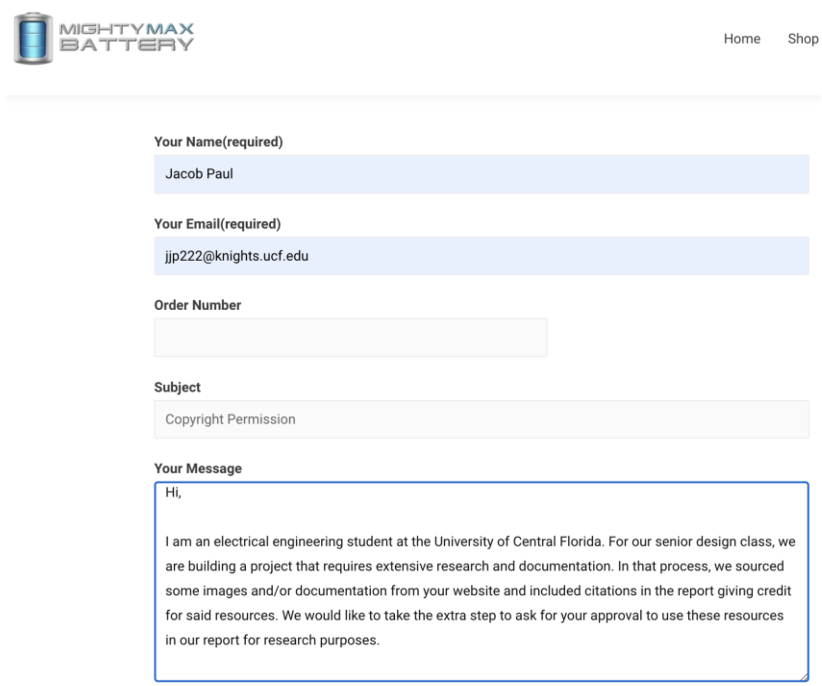
To: editor@circuitdigest.com

Hi,

I am an electrical engineering student at the University of Central Florida. For our senior design class, we are building a project that requires extensive research and documentation. In that process, we sourced some images and/or documentation from your website and included citations in the report giving credit for said resources. We would like to take the extra step to ask for your approval to use these resources in our report for research purposes.

Thank you,
Jacob Paul

Figure 119: Permission for Circuit Digest Image (Permission Approved)



MIGHTYMAX BATTERY

Home Shop

Your Name(required)

Jacob Paul

Your Email(required)

jjp222@knights.ucf.edu

Order Number

Subject

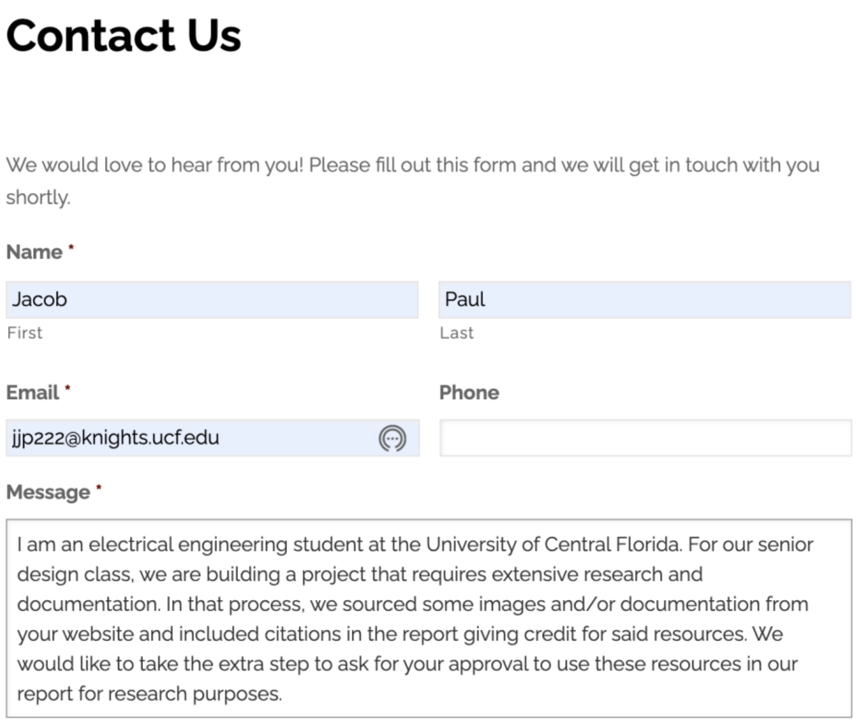
Copyright Permission

Your Message

Hi,

I am an electrical engineering student at the University of Central Florida. For our senior design class, we are building a project that requires extensive research and documentation. In that process, we sourced some images and/or documentation from your website and included citations in the report giving credit for said resources. We would like to take the extra step to ask for your approval to use these resources in our report for research purposes.

Figure 120: Permission for Mighty Max Battery Specs and Image (Permission Pending)



Contact Us

We would love to hear from you! Please fill out this form and we will get in touch with you shortly.

Name *

Jacob — First

Paul — Last

Email *

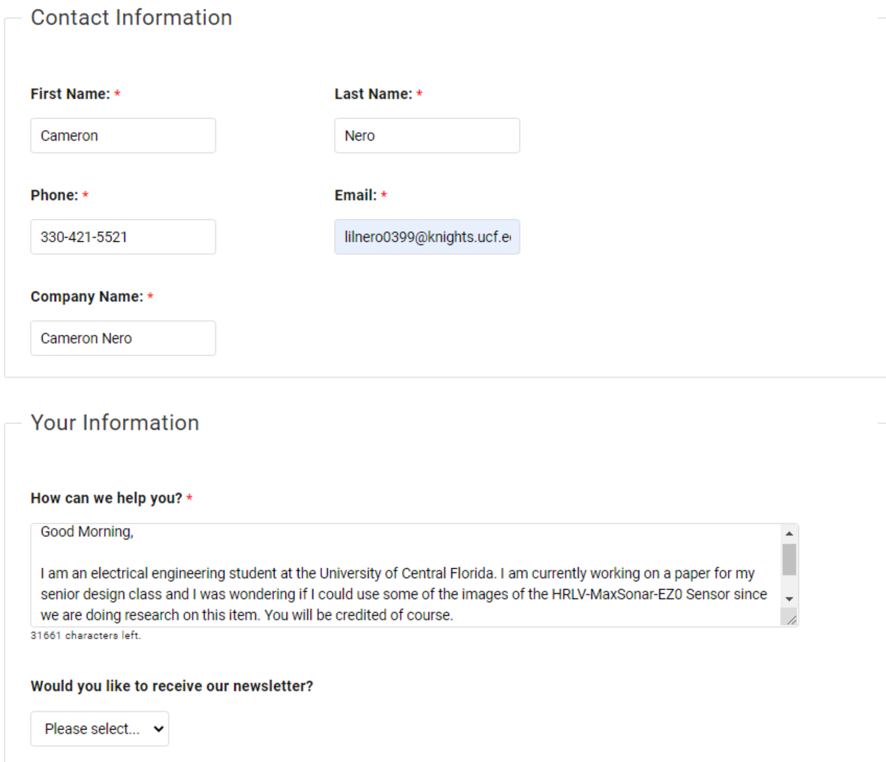
jjp222@knights.ucf.edu

Phone

Message *

I am an electrical engineering student at the University of Central Florida. For our senior design class, we are building a project that requires extensive research and documentation. In that process, we sourced some images and/or documentation from your website and included citations in the report giving credit for said resources. We would like to take the extra step to ask for your approval to use these resources in our report for research purposes.

Figure 121: Permission for Motion Control Info and Image (Permission Pending)



Contact Information

First Name: *
Cameron

Last Name: *
Nero

Phone: *
330-421-5521

Email: *
lilnero0399@knights.ucf.e

Company Name: *
Cameron Nero

Your Information

How can we help you? *

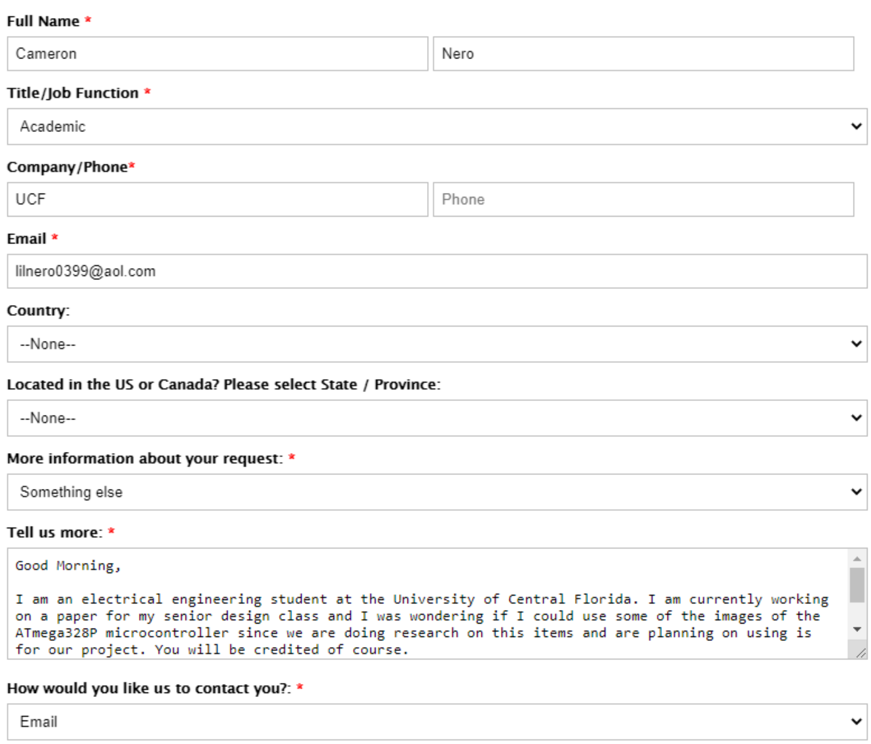
Good Morning,

I am an electrical engineering student at the University of Central Florida. I am currently working on a paper for my senior design class and I was wondering if I could use some of the images of the HRLV-MaxSonar-EZ0 Sensor since we are doing research on this item. You will be credited of course.

31661 characters left.

Would you like to receive our newsletter?
Please select...

Figure 122: Permission for MaxBotix Sensor (Permission Approved)



Full Name *
Cameron | Nero

Title/Job Function *
Academic

Company/Phone*
UCF | Phone

Email *
lilnero0399@aol.com

Country:
--None--

Located in the US or Canada? Please select State / Province:
--None--

More information about your request: *
Something else

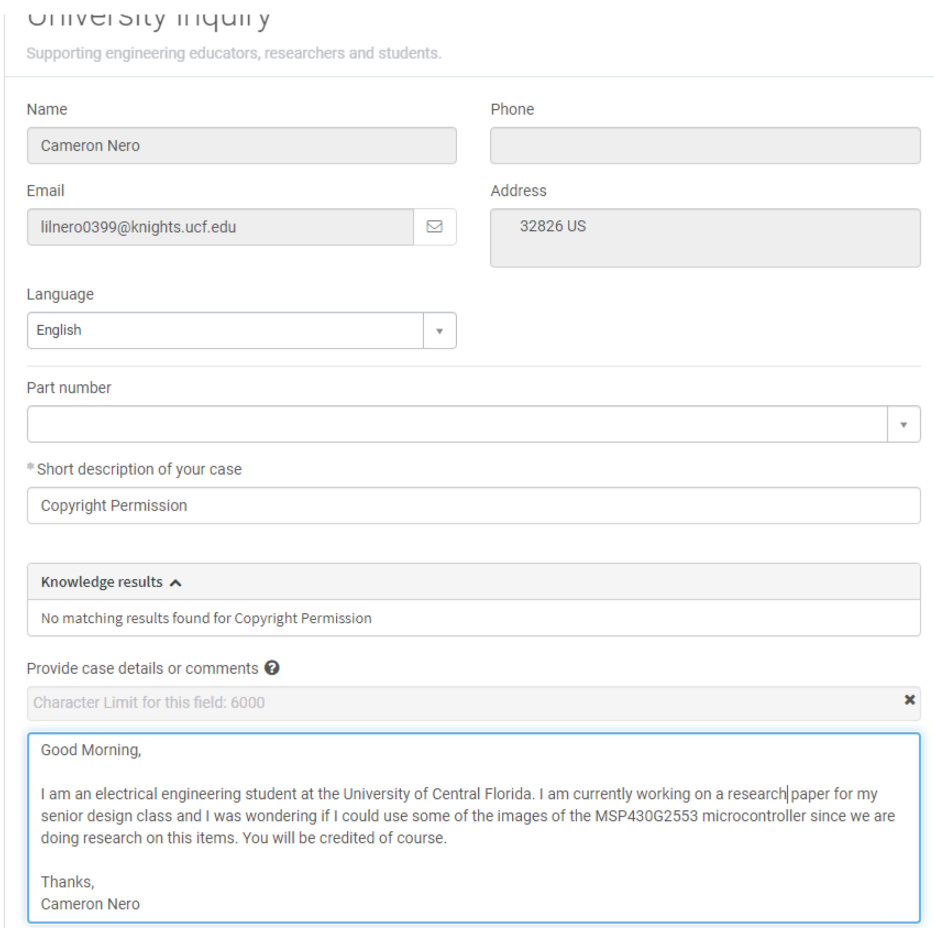
Tell us more: *

Good Morning,

I am an electrical engineering student at the University of Central Florida. I am currently working on a paper for my senior design class and I was wondering if I could use some of the images of the ATmega328P microcontroller since we are doing research on this items and are planning on using is for our project. You will be credited of course.

How would you like us to contact you?: *
Email

Figure 123: Permission for Microchip MCU (Permission Approved)


University Inquiry

Supporting engineering educators, researchers and students.

Name: Cameron Nero

Phone:

Email: lilnero0399@knights.ucf.edu

Address: 32826 US

Language: English

Part number:

*Short description of your case: Copyright Permission

Knowledge results

No matching results found for Copyright Permission

Provide case details or comments

Character Limit for this field: 6000

Good Morning,

I am an electrical engineering student at the University of Central Florida. I am currently working on a research paper for my senior design class and I was wondering if I could use some of the images of the MSP430G2553 microcontroller since we are doing research on this items. You will be credited of course.

Thanks,
Cameron Nero

Figure 124: Permission for TI Microcontroller (Permission Approved)

Good Morning,

I am an electrical engineering student at the University of Central Florida. I am currently working on a paper for my senior design class and I was wondering if I could use some of the images of the HC-SR04 ultrasonic sensor and the Sharp GP2Y0A21YK Infrared Proximity Sensor since we are doing research on these items. You will be credited of course.

Thanks,
Cameron Nero

Figure 125: Permission for SparkFun Sensors (Permission Approved)

*Design Cycle

Concept

*Annual Usage

Less than 1K

Distributor Preference

(Please select from the list below)

*Application Field

Others - Write in value

Application Field in text

EE

Description

Good Morning,

I am an electrical engineering student at the University of Central Florida. I am currently working on a paper for my senior design class and I was wondering if I could use some of the images from the website below where it is talking about switching vs linear regulatos since we are doing research on these items. You will be credited of course.

https://www.renesas.com/us/en/products/power-power-management/linear-vs-switching-regulators

Thanks,
Cameron Nero

Figure 126: Permission for Renesas Images (Permission Approved)