Final Document
Version 6 - 120 pages

Trashporter

A Robot that Transports Trash

University of Central Florida
Department of Electrical and Computer Engineering
Senior Design 1
Dr. Lei Wei

Group 7

Jourdan Callahan - Computer Engineering
Andy Kuang - Computer Engineering
Dhanesh Singh - Computer Engineering
AnnaBelinda Zhou - Electrical and Computer Engineering

July 24, 2022

Table of Contents

List of Figures

List of Tables

1. Executive Summary

Today we live in an age where everyone is busy with something that takes higher priority over the household chores, and thus being pushed aside for another time. One of the biggest motivations of the advancement of technology is to make our lives easier by making something that completes a task easier than it currently is. In this case, it is taking out the trash, this is one of those chores that we must do because it is a time sensitive task where you only have one opportunity to take it out a week, else you would have to wait for the next week if pushed to the side. This project will introduce a device that, while it will not entirely take out the trash entirely, will make the process much more convenient by collecting all the trash in each room and gathering them to one location, this is The TrashPorter.

The TrashPorter will be an autonomous trash bot that goes out and seeks specialized bins that it will collect trash from. It starts at a central docking station where it is convenient for the consumer to take out the trash from, gets set off to go to the designated locations with its built-in sensors, sense the bin it needs to collect from, lift the secondary bins, lifts it up and dumps the trash into the main compartment of the trash robot. It then returns to the starting location where it can reset and wait for a command or a specific set time to go out to collect again.

This device was heavily inspired by many autonomous household robots in the market already on sale, but nothing widely known as a trash bot. Robots in the household today will do things like vacuum, wash clothes, and even mowing the lawn, these products are pretty common in the market today and most people have heard or own one themselves because of how well they work within the household, spending less time on what must be done and more time on what really matters. It is uncommon to have seen or heard of indoor trash bots in the market being widely sold yet. There have been people that have built their own robots to bring the large bins outside to the curb of their driveways to be collected by the trash trucks, but this is an outdoor robot as well as something that was not for sold and built by himself, this could carry about 3 times its weight and moves autonomously, it was a great device to help many people that could not do this task on their own and needed external assistance from someone. Our robot will assist people on the insides of their homes and acts as a garbage truck inside your own home.

There will be the implementation of both hardware and software for this device to function as intended. The hardware can be easily seen from consumers as the robot will need to drive from one location to another, extend its claw to the bin, lift the bin up on a motorized track system, and dump the garbage into the main can. The software side can be implemented as first programming the robot to work manually so that the measurements are made so it can correctly and accurately pick up the trash and bring it in the main bin. These precise movements are crucial for it to consistently be successful in its task. Another component to the robot is when it should be sent out and decide when to take out the trash, this will be figured out through the mobile application that will be developed for the scheduling of the robot. We will give it the features of manually sending it out to go collect the trash on the spot or scheduling a set time where it will go out on that date and time. It will also collect metadata where the user can see when it is full, notifications, and weight collected. All these features working together will make something to stay in the households in the days to come.

2. Project Description

In our growing world of technology, it is going towards making people's lives easier when there are so many responsibilities to take care of already. Our team aims to innovate on the idea of home appliances that make these chores faster and more efficient to manage with The Trashporter, an autonomous trash robot that goes around the house and collects garbage from each room to be brought back to a singular unit docking station. This project will be developed throughout 2022 and the process will be detailed within the report.

2.1 Project Motivation

The motivation behind this device was to create a user-friendly device that's lightweight, portable, and convenient to consumers to complete a tedious household task. The device was heavily inspired by other autonomous vacuuming robots already in the market, using the concept of completing a task, then returning to the designated location where the robot can be cleaned out by the user. The device has many avenues to be enhanced with many unique features making the idea very flexible in the direction we would like it to go, some features that could be developed are a trash sorter where it can scan the item or bin to determine if it is considered trash or recyclables, being fully autonomous where it can map each room knowing where to pick up the bins and return back to its original docking location, and many other possibilities. Our team can use previous knowledge on boards to develop this project for example in previous classes we used microcontrollers to control the main functionality of devices, so boards like the Texas Instrument MSP430FR6989 and the Arduino Uno boards knowledge can be directly used in this device where there are many available libraries to help construct it. There were many discussions on how the robot would transfer one bins trash to the main bin, and after brainstorming the pros and cons of each idea, we have concluded that it would be easiest to build and external arm on the trash bot connected to a track to lift the sub-bins and dump the contents into the main bin. There is also a great area to implement software, being a mobile application to either manually be sent out to collect garbage at will or schedule when it should be sent out on a consistent basis. Some exciting things we will get the opportunity to work on is machine learning to train the bot to know how to navigate around the room and reach its desired target, this is possibly one of our biggest challenges that will be faced during our projects, making it that much more satisfying to succeed in this project.

2.2 Goals

The Trashporter plans to satisfy three main goals at the end of the term. It will autonomously seek out the sub-bins from the docked location to the desired location while avoiding obstructing objects. Additionally, it will pick up the targeted bins with a robotic arm, be lifted by a track system, and empty all the contents of the sub-bin into the Trashporter and return to a centralized location to have all the trash in one easily accessible space. Finally, The Trashporter will be connected to a mobile application where it can be set on a schedule to go out and collect all of the trash, or manually told by the user to go out and picked up immediately, the mobile application can also let the user know when the main bin has reached its capacity limit.

2.2.1 Basic Goals

The main goal we are trying to achieve with the robot is that it can move from one stationary position to another on its own with the use of a machine learning algorithm. It will then pick up the trash, raise it up to the right angle and empty the items from the sub-bin into the main robot bin. It will then return to the original spot where it has started so that it is easy and quick for the homeowner to collect all the trash that both have claimed. This will be our most basic goal to achieve and from there we have a lot of options to expand from it. Just from this idea it has a lot of components which need to be carefully planned to be executed correctly. First the robot must be constructed on its own where all these motions will work by a controlled user, this includes motion controls, acceleration, and steering. After motion works properly, we must ensure all the physics behind it are properly calculated so it is stable when the bin is full or empty, it then needs to travel to the smaller bins where it will have a external claw that we have to build upon a preconstructed garbage bin, this claw must extend from the base of the trash bin, grab the bin, then angle it so that everything inside the other bins falls in the right space to go into the main bin, we expect the construction of this will take majority of the time and the biggest challenge because of the physics calculations behind it as well as the programming of the robot. Once all these functionalities work as intended is where our second biggest challenge is going to be, and that's the machine learning aspect of it because we need to develop an algorithm which can accurately and precisely execute the movement, lifting, and returning all on its own with the added precaution of avoiding objects. While our team's knowledge on machine learning is not starting for nothing, it is a big step up from previous classes we may have taken and will have to implement new ideas and concepts that will be having to be researched.

Some of the specifications that will be expanded on throughout the paper is the weight and height. We estimate that the bin should carry an upper limit of 15 pounds, this is at the max capacity of the bin. It is important that it can carry a decent amount of weight as the ideal idea is that it goes to each room of the house and collects all the trash while still being able to perform all its functions. The other requirement of the robot is that it is about 22 inches tall, this allows enough space for the bin to collect a decent amount of trash from all the rooms. As this is a prototype model of what might be desired in the market it will be a scaled down design for this project. The other bins that the main bin will be collecting from are going to be a fraction of the scale of the main bin so that it is able to fit all the contents of it, this allows for a lightweight and compact design.

2.2.2 Advanced Features

The TrashPorter to work on because it allows many different additional features to be added, explaining the basic goal of the project, here we can dive into and add onto the robot. One of the things that we are aiming to achieve and add to the robot is it can go to multiple rooms and pick up bins. Our most primitive goal is that it can go from the home spot to another bin and pick up trash from one bin. The goal to pick up multiple bins should not be too much of a challenge because once we have developed the algorithm to pick up one trash, it is just the matter of having it repeat its same function until it reaches a point where it is full and needs to return to the original location where it has started. This works with the next additional feature being the ability to detect when the bin is full, we will add a feature where it will have sensor at the top of the bin so it knows when it is full and cannot take anymore trash, the only option when this criterion has been met is to return to the original docking location. These two features are solely hardware implementations that we can add additionally to the basic goal, but we can also incorporate software features that compliment what is added here to make the user experience more convenient. An example of one of the futures that can be added is notifications when the trash bin is full, this will be achieved through a mobile application that we will develop So the robot can let the user know when it is time for the trash to be collected and it can continue for its next cycle. The next feature is scheduling, through the mobile application we can let the robot know when it should go pick up the trash. This is going to be either on a scheduled cycle or manually told by the user to go out and collect trash at the user’s command. To recap these features that we are going to add to the robot, we have the robot starting from the docking location going to multiple rooms to collect trash and return to the starting docking location. It then needs to know when the bin is full, it will know because it will have a sensor implemented near the top of the bin where it can sense that the bin is completely full and cannot receive any more trash, this ensures that when it goes out to collect multiple bins in different rooms it will not overflow the robot. From the software side we will develop a mobile application where it has a notification system that sends the user's mobile app a message to let them know that the bin needs to be emptied out before it can proceed once again. Lastly the robot needs to know when it should go out autonomously to go collect trash from each room, the feature that we will implement so that it can do this is a scheduler, we will allow the robot to be scheduled through the mobile app at a certain time and date when it should go out and pick up trash, because of the previous feature it will not go out and pick up trash if it is already full, we will also let it go out and pick up trash manually at the users command. These are the advanced features that we would like to implement to a robot and can be distinguished from our basic features because they mainly act as a convenience more than a requirement for the user. the idea of autonomous robot is very flexible allowing us to add many additional features to make the user experience easier to use and more convenient, while there are already many products like this on the market already, we have not seen many commercial products already released for this idea to pick up trash from multiple rooms and bring it to a main station to be discarded, we believe that this product will make People's Daily household lives much easier when they have many things they have to do each day.

2.2.3 Stretch Goals

We have explored the goals that are required and the goals that can be additionally added to make this product much better than the bare bones. In this section, we will explore the different possibilities if given the time. One of the features that would make this product much better is being able to charge itself without the user's interaction with it. we want the robot to last as long as possible and to be as efficient as it can and the whole idea is to have it do all of these things on its own without any user interaction, very similar to other cleaning autonomous robot products it will be running on battery power supply and it would be very inconvenient to have to either manually plug it in to charge it itself or replace the batteries periodically, therefore we have come up with the idea of the robot with turning back to the docking station where it would find and connect its charging port to the docking location and be ready for the next time it needs to perform.

Another idea that was brought up is the ability to have trash segregation. This means that the robot can distinguish between a bin full of trash and a bin that is meant to be recycled, we have been inspired by many external projects which would scan and read what the item is that it is looking at to see whether it can be recycled or discarded in the trash, but we want to implement this idea within our trash bot. There were many ways to go about this, but many included a vast knowledge of computer vision and this could go out of the scope of what we are looking to achieve here, so we believe the easiest solution is to differentiate multiple bins by giving them a barcode or QR code which we will mount a camera onto the robot where it would scan the code and know whether it is considered trash or a recyclable. This means that instead of a single sub-bin unit there would be multiple bins within a room that it would go in and collect from, this idea acts like your own personal garbage and recycle truck within your home. Trash segregation is a useful idea because it can achieve not only picking up trash but also plastics or papers that can be recycled and brought back to a single location to be collected by the user, this acts as a smaller scale to our real-world situation in which we have days that the truck picks up garbage and other days where it picks up recyclables. Having more flexibility within our project only makes the product more useful to the consumers that investigate the product and decide whether it is worth buying or not. There are many constraints with these additional ideas and we have to keep in mind that if it also collects recycle there will be a bit more manual work towards the user and knowing whether it is a recycle day or a trash day and they will have to adjust accordingly to this additional feature, this can also be incorporated into the app to have it collect a certain been on a certain day depending on what the demand is.

There are many ideas that can be added to this robot, but these are just some of the ideas that we have thought of and can be additionally added if given enough time after completing the basic and advanced features within the project. This is the stretched goal section in which we explore many additional options that can makes the product even better than what it currently already is being the ability to make a chore less tedious by going into each of the rooms collecting with trash and bringing it back to the starting location so it could be collected as well as having it self-charging to the dock. As technology is advancing, it is not a surprise that we will likely see something like this idea in the market in the near future and we hope that we will be one of the few to explore this desired product.

2.3 Objectives

The objectives of the creation of The Trashporter isn't only to create a autonomous robot that collects trash for the convenience of the consumer, but also to demonstrate the knowledge and abilities of the team on how we have come together to collaborate, communicate, and execute all of the hard work and research that went into developing this project. We will connect the hardware aspect of construction to the mechanical mechanisms so that it functions properly with the software logic so that the bot knows how to navigate to the desired target, accurately pick up the sub-bins, collect it, and navigate back to the docking station seamlessly. Our team will research, experiment, and develop a device using the knowledge we have gained throughout our engineering career, as well as furthering our knowledge with new challenges that have not been explored yet within our educational journey.

2.4 Project Constraints

Supply Shortage: Due to Covid-19 and lack of truck drivers and shipments, there is a global supply shortage that is affecting technology and parts vastly. The current supply shortage can affect the time it will take to receive parts and will also affect the prices.

Cost: We have decided to have a budget of $800 dollars for this project which we hope is more than enough to buy everything we need. We are willing to spend more if needed but this should be more than enough for our goals.

Time: As I mentioned before, the possible wait for parts can delay our development process for this project. We will plan to order items as soon as possible to try to avoid this. We have created a timeline that has specific deadlines that state how much progress should be complete.

Trash weight: One thing we have to take in account is the weight of the trash. Some bins we will have to pick up will be heavier than others. Because of this we will make sure the bot is strong enough to pick up most bins and have a weight limit.

2.5 Specifications and Requirements

For Specifications and Requirements, it is important to pinpoint the limitations and expectations of our project. We will be specifying what our robot should be doing in a measurable way to ensure that the goals have been met with the given design. We will give requirements that we can test based on a system which can be measured in a unit of numbers, this refers to things like time, weight, or quantity. There will be different expectations of requirements being the basic goals, advanced features, and stretch goals. Basic goals are the bare minimum of what the device should be able to do, advanced features incorporate something a bit more complex like an external application controlling the robot wirelessly, and stretch goal is what could be potentially added given enough time.

2.5.1 Hardware Specifications

The hardware specifications describe what is physically expected of the device. It will include the dimensions of the device as far as weight, size, and functionality. These will be labeled in numbered units so it can specifically show that it can be physically achieved. The highlighted boxes show which one we would demonstrate for the device to be considered working as intended. These specifications will then be showcased in the future to ensure that it is up to design and successful.

	Requirements	**Units**
1.0	The device shall be within the specified weight and dimensions	≤ 15 lbs ≤ 15" x 11" x 20"
1.1	The device shall detect a trash can with a camera	1 Camera
1.2	The device shall be able to connect to the mobile application via wifi	
1.3	The device shall be able to be controlled from the mobile app	
1.4	The device shall be able to lift up the trash can and dump the trash into the robot and return within a timeframe	20 seconds
1.5	The device pick up mechanism should contain two motors and two servos	≤ 2 Motors ≤ 2 Servos
1.6	The device should receive commands through mobile app within a timeframe	10 seconds
1.7	The device should cost no more than $800.	≤ $800

Table 1: Hardware Specifications and requirements

2.5.2 Software Specifications

The software specifications describe what the software side of things should do within the project. As it can be quite clear what is expected of the device on the hardware side, it can be a bit trickier to know what should be expected of the software. There will be a mobile application that interacts with the functionality of the device. The application can directly control the device or specify when the robot should behave autonomously. This allows a wider range of options in which the device can operate instead of a static design.

2.0	The device will be paired with an application on a mobile device.
2.1	The app shall be able to power it on and off
2.2	The app shall be able to control is manually
2.3	The device shall be able to receive a notification when the trash can is full.
2.4	The device will be able to track how many bins have been collected.

Table 2: Software Specifications and requirements

2.5.3 House of Quality

In this chart, we have measured and evaluated different marketing requirements and engineering requirements. The marketing requirements consist of ease of use, performance, energy cost, and quality. The engineering requirements include weight, cost, quality, dimensions, efficiency, and speed. We have compared those key factors against each other in terms to get a better grasp of our project’s feasibility.

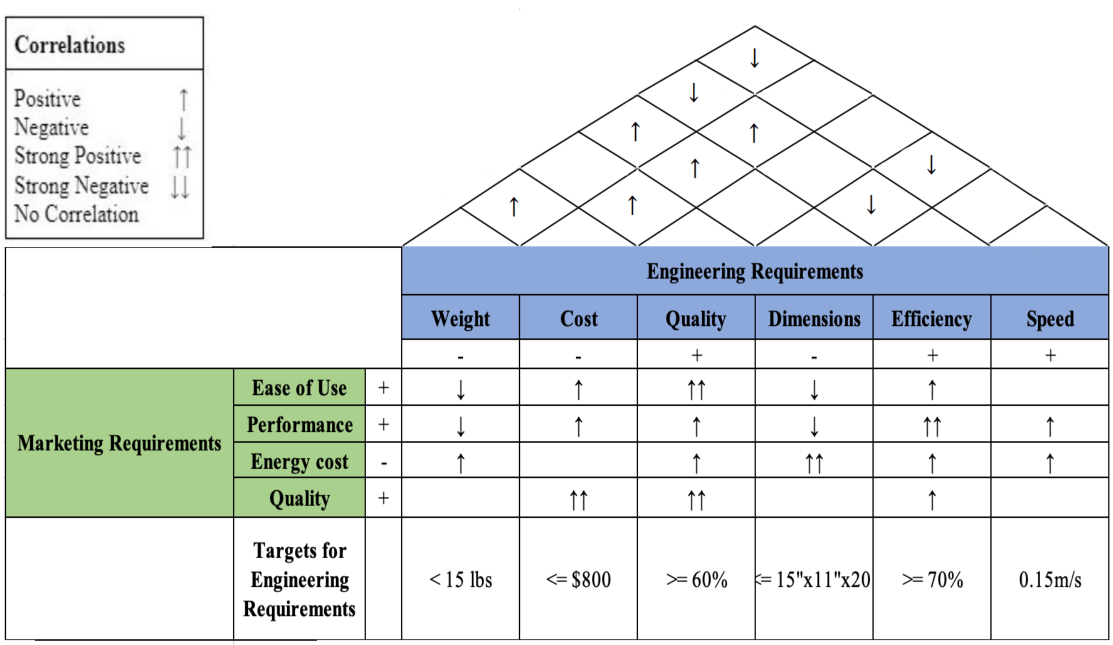


Figure 1: House of Quality

2.6 Block Diagrams

The block diagram explains the different components within the device that need to be worked on. It is good to have a block diagram to indicate who will be working on which part within the project and allows for transparency. The blocks are color coded to indicate which team member will be working on the specified component. There is a hardware and software section because each has their own parts that at the end will be connected. Since the groups have been divided into software and hardware, we can expect to see the hardware team taking on more tasks for the respected parts and software on their parts.

2.6.1 Hardware Block Diagram

The hardware block diagram describes all the physical components that will be worked on for the device. There are subparts that need to be worked on which will be connected to a bigger component, this requires multiple parts to make the overall system to function. The outer components are the independent devices which can function by themselves, but will be mounted to the main driver, giving access to the feature.

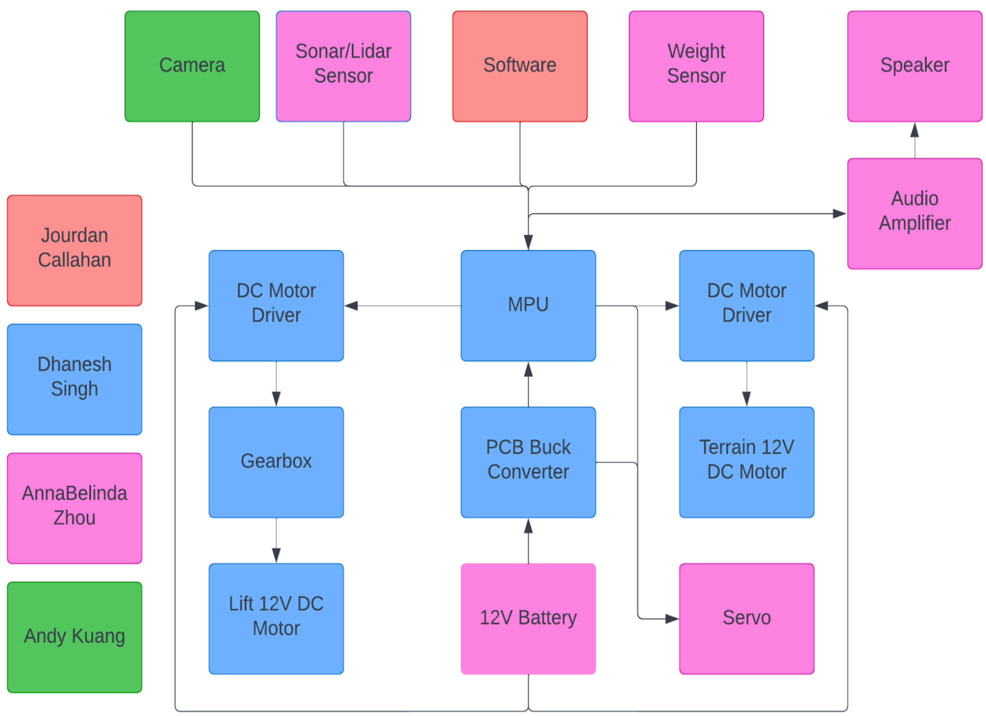


Figure 2: Group Members Roles and Responsibility for Hardware

2.6.2 Software Block Diagram

The software block diagram describes what will be worked on within the software side of things. This is going to refer to the mobile application as well as any computer coding logic. We have the robotic logic of the hardware components, to the external parts mounted on the circuit boards, as well as the pure software user interface that gives commands to the robot externally. A big portion of controlling the robot is the mechanics of it, this will be performed manually through the user, then we would have to develop an AI algorithm for the robot to act upon these functions on its own. Putting the hardware and software together will make the robot function properly and perform as intended.

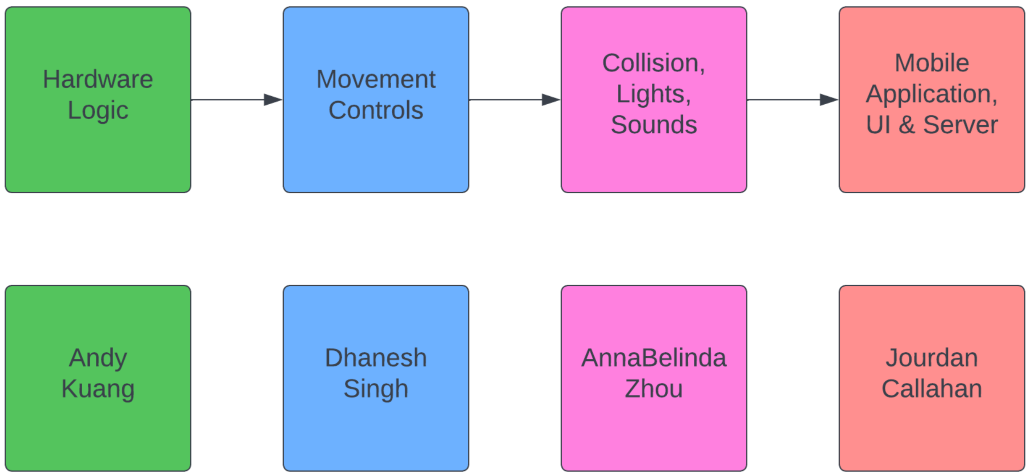


Figure 3: Group Members Roles and Responsibility for Software

2.9 Decision Matrix

The table below displays the several ideas we had among our group members. It evaluates the costs, our knowledge of the idea's technology, our motivation, and the educational goals in Senior Design of each project idea.

Project Ideas	**Costs**	**Familiarity with Technology**	**Motivation**	**Educational Goals**
Trash Pickup Robot	~ $300	80%	80%	100%
Portable Salt Water Distiller	~ $100	25%	60%	60%
Sisyphus Table	~ $200	25%	75%	80%

Table 3: Decision Matrix

All three project ideas are practical and feasible to build in 2 semesters. The main reason that the trash pickup robot was chosen is because all the team members had experience with building a robot. On top of that, there is more motivation for the trash pickup robot because it is more challenging. Especially compared to the sisyphus table since there are already YouTube tutorials made for building sisyphus tables. Looking at the Senior Design project requirements and educational goals, there is more room for hardware and software implementations for the robot compared to the other two project ideas. PCBs can be designed for different functionalities in the robot whereas there is limited functionality for the other two. Overall, the trash pickup robot takes more advantage in functionalities, project requirements, and the familiarity with the technology.

3. Research and Parts Selection

In this section, we will look into different technologies that are embedded in the robot. Different technologies will be researched and compared to find the best one that suits our robot. Parts and components will be compared and selected to match up with our robotics design. This section will look into the mechanism of the arm and lift system, microprocessing unit, power supplies, motors, the different sensors needed to measure environmental data, and cameras to make the robot perform its tasks as described before.

3.1 Claw mechanism

One of the biggest challenges for this project will be the operation of picking up an object, raising it up, and rotating it at the right angle for the claw to dump the items inside the object. There are many designs out there that we can use, and in this section, we will explore some of the different options that would work best for this robot. There are many resources that can be found through research on project websites, and they have slightly different designs. We plan to use some of the concepts of these designs to modify them into something that would suit our project.

3.1.1 Popsicle Stick Design

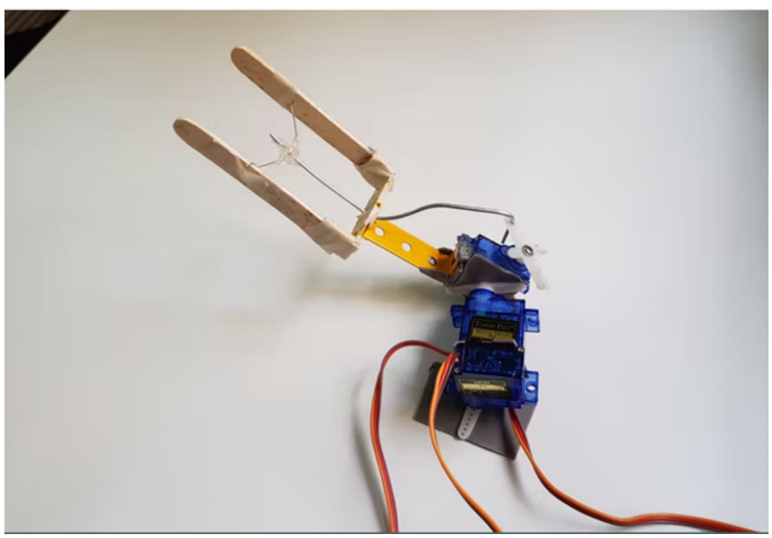


Figure 4: Robotic Claw: Popsicle Stick Design

The first design is going to be a very simple and small scaled version that we could use for testing purposes and upscale for what we need. We would need a microcontroller like an Arduino Uno, Micro-Servo motors, Jumper wires, tape, hot glue gun, and popsicle sticks. The claw will be composed of the popsicle sticks, or any generic wooden sticks glued to a metal rod acting as the joints, the claw will be connected through a thin wire making the arms stable not to move forwards or side to side. Next, we attach the Servo Motor to the end of the metal plate and feed the metal rod connected to the wires pulling in the claws into one of the holes of the Servo Motor which will make the claws open and close when rotating.

This design is simple to implement, and we could use this to test a smaller version of what we intend to build with TrashPorter, we could first use the wooden claws to test the functionality of the robot for it to lift an object then have it transported to the desired location. When we have replicated the design we could use more sturdier material like metal as the arms instead of wood, stronger wire, and motors so that it could lift a heavier object. The website provides the given Arduino code and schematic to easily follow how to construct it and code it to function properly.

The sample flowchart scene and figure 5 demonstrates the coding process of getting the desired angle for a zero servo. The program runs on a continuous process where it constantly checks its state. In this scenario the state is defined by whatever angle it is in. This flow chart also shows how servos are different from motors as they do not rotate for 360° which makes it great for specialty work such as specific movements where the range of motion has to be limited.

Sample code flow chart:

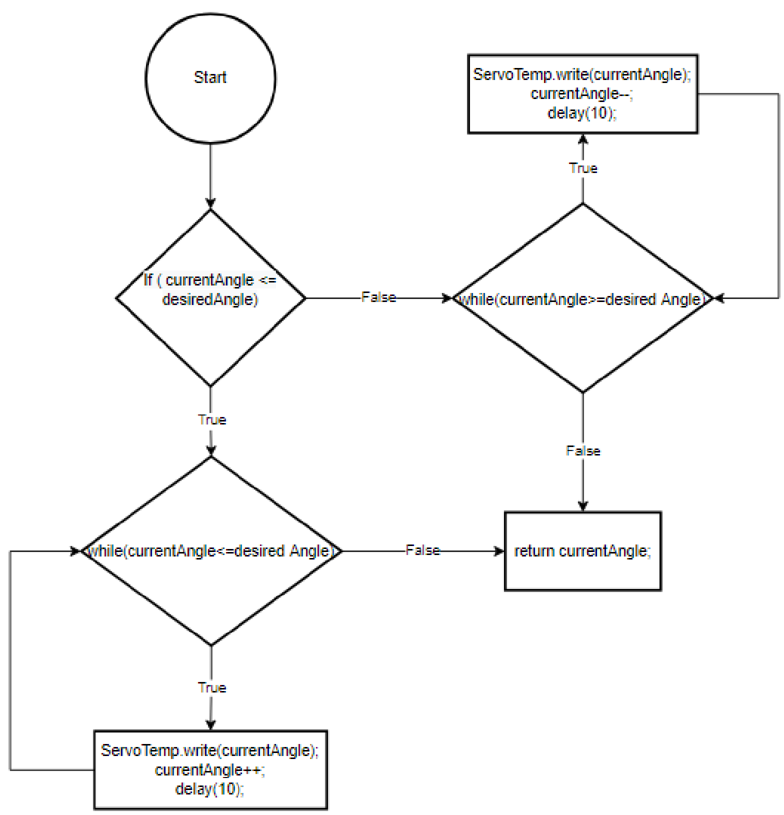


Figure 5: Popsicle Stick code flow chart

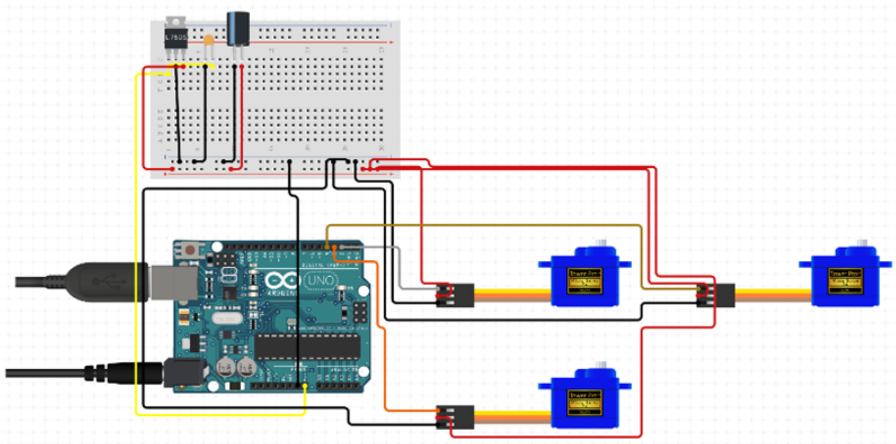


Figure 6: Build Schematic: Popsicle Stick Design

3.1.2 3D Printed Claw Design

The next design we will look at is more upscaled compared to the first version that has been researched, it involves 3D printed materials and gears to contract and retract the system. The original goal of this design for the source of this is to grab a plastic cup, it should pick up the cup with liquid inside it where it is stable enough that the liquid does not spill out to replicate a hand, it can detect if the cup has been grabbed so it will stop at a certain position, this was implemented by using a limit switch sensor to detect the cup, and finally it detects when the cup has been fully lifted up from the stationary position. The design of the claw is 1 cm thick, which is about 20% of the height of the plastic cup, this thickness was necessary to implement the sensor to fit on the claw. There was rubber inserted within the inside of the claw to give the claw more friction to give it more gripping surface so as to not make objects slip when being grabbed. The parts that were used were a 6-axis robotic arm carried by the SG-90 Servo motor. The claws act as traditional claws with gearing making the left and right claw meet in the middle to hold the object steady and flexible to grab different size cups.

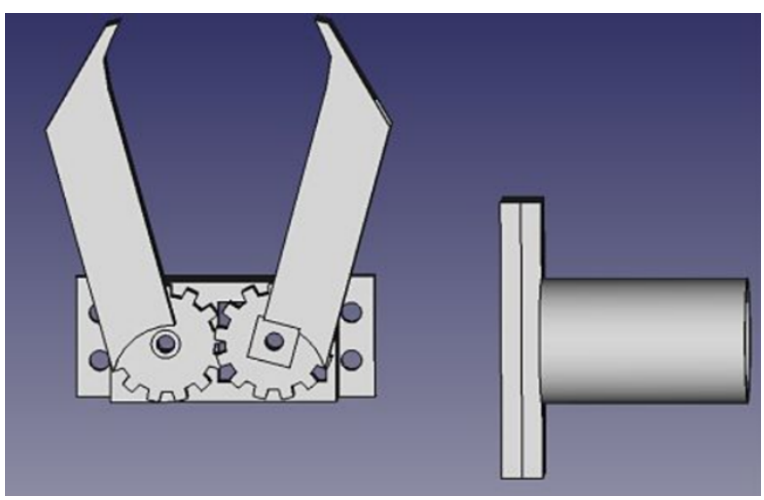


Figure 7: Robotic claw: CAD design

This is a good candidate to choose from because it provides the source code for the mechanism of the claw, the process in which to build it and looks quite sturdy from the start for what will be similar enough of what we need in our implementation. It does not go too much into details of the dimensions and the specifications, but using the concepts given in the blog it can be modified for what we need it to do in picking up trash cans instead of cups.

In the figure below, we describe the process of how we will control the microcontroller in relation to the servo. One or two servos will be used to control the robotic claw. The robotic claw will open and close, in this case it will be used to grab on to the smaller trash can. In this example we measured the time in milliseconds.

Sample code flow chart:

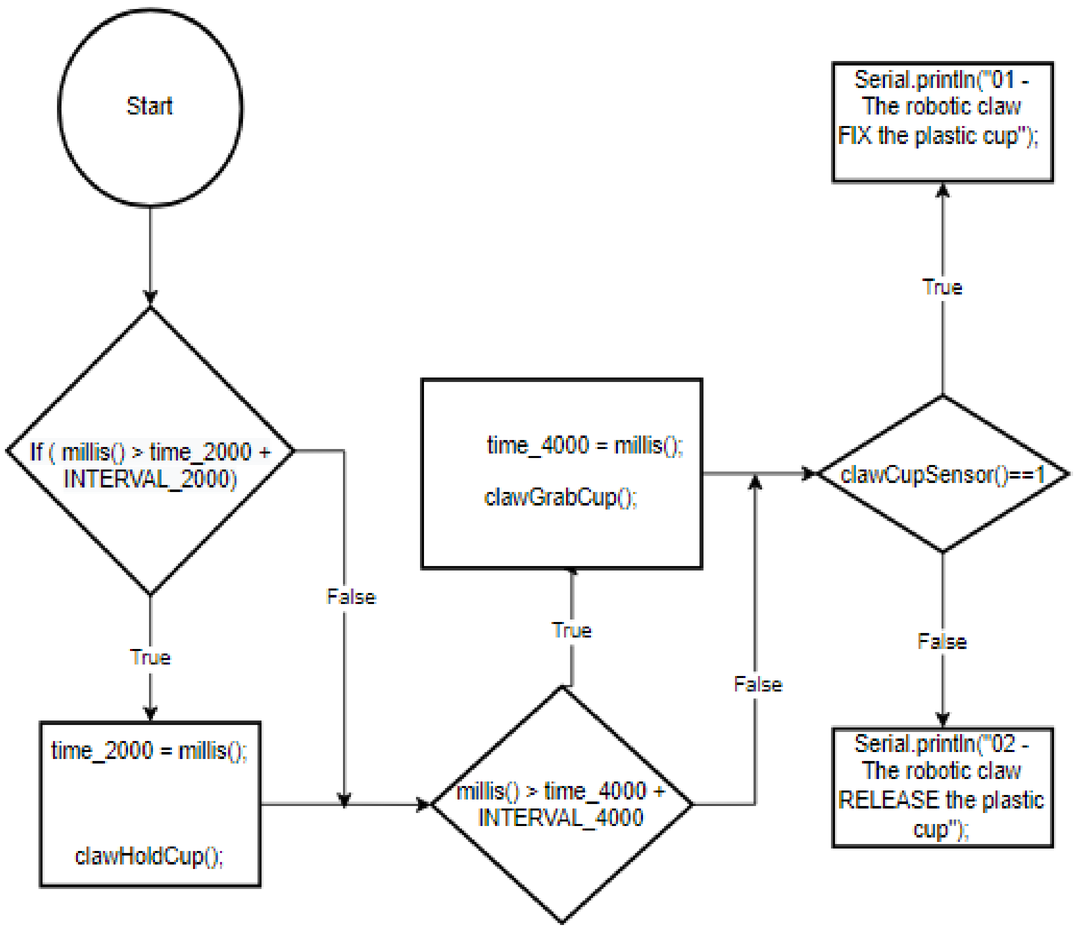


Figure 8: Popsicle Stick code flow chart

3.1.3 Claw Comparison

	Popsicle Stick Claw	**3D Printed Claw**
Materials	- Arduino Microcontroller - SG90 Micro-servo motor - Jumper Wires - Double sided tape - Hot Glue Gun - Metal rod - Metal wires	- 3D Printer - Claw design - Gear design - Servo Motors - Base Housing - Jumper wires - PVC pipe - Sensors - Limit switch
Practicability to desired design	- Good practice design to upscale for actual build - Relatively cheap to other designs - Code provided - Schematic provided - Smaller material list - Step by step instructions - Will require more upscaling - Might be too simple for added track lift system	- Close to actual desired design - 3D Printer available - Flexible object size to grab - Code provided - Dimensions provided - Repurposing design goal - More costly - More complicated - Instructions not provided

Table 4: Claw Comparisons

The 3D Printed Claw design looks more suitable for what we are looking for to pick up trash. It requires less upscaling and redesign because of how similar it is to how we plan to use it. The object it originally picks up is round, as will the trash bins our trash bot will pick up and it is much sturdier than the Popsicle stick design. The Popsicle stick design can still be a good option for testing, but there would need to be a large overhaul to make it work for what we want, so though the printed design is more costly and complex, this is the direction is we would need to go into for the actual designing of the trash bot claw. Since we have the privilege of having access to a 3D printer, we anticipate using this to design the accurate specifications and dimensions to meet what is needed.

3.2 Robot Lift System

While the bulk of the robot's mechanism is going to be the claw grabbing the trash bin and dumping the trash from one bin to the other, we expect the main trash bin to have a large capacity where it can pick up trash multiple times before returning to the main docking position. This means that the bin has to be fairly big to contain the different items of trash that could pile up, the issue arises as the claw would be too low to properly just lift and dump from one bin to the other assuming the height of the main bin is taller than the smaller ones, this is where we will need to implement a lift system to carry the claw up the trash can and dump from the top of the bin. Lifts are used to raise the height of different mechanisms vertically, usually through an activated motor attached to a gear system, sprocket, or chain system which is mounted directly on the body of the robot. This requires a lot of planning as accuracy is crucial for multiple parts to work in sync with either and to avoid structural failure if not calculated correctly. Here we will explore some of the few options that can be used for the lift and determine which is the best fit for our design.

3.2.1 Scissor Lifts

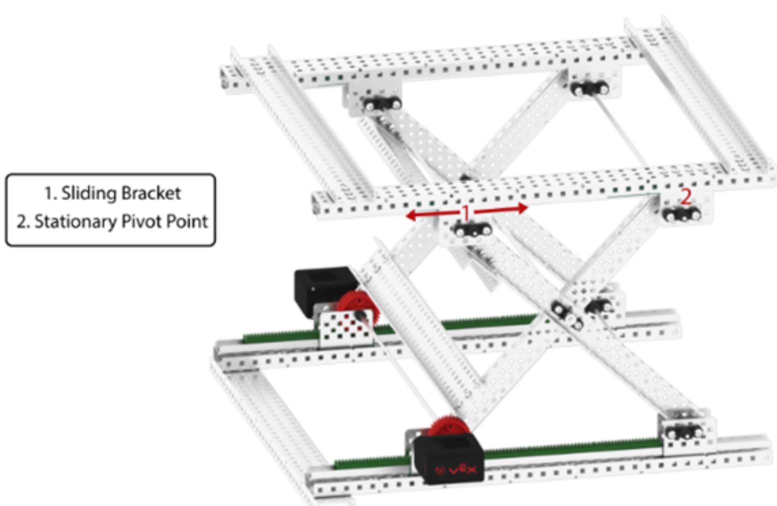


Figure 9: Vex: Scissor Lift

Scissor Lifts are built by having a pivot point in the middle of the two structural pieces of metal that cross each other. One end of the piece of metal is fixed to the pivot point of the chassis, while the other piece can freely slide from the first end to the chassis, when the two points are closed, the structure raises the object or part. The two sides should be constructed to have an equivalent amount of force on each side to ensure stability when being raised and stationary. The Scissor Lift can be raised by having a spur gear activated by motor alongside a Rack Gear to pull one side of the scissors to make it slide towards the pivot side. In this example they use an 84T High strength Gear connected to the metal pivot point between the two metal sides. It is powered by an a12T High Strength Gear attached to a motor, The 12T Gear moves the 84T Gear to lift the device.

This system is considered one of the most difficult lifts to implement because of the part requirements and the amount of space it will take up for the design but makes for an efficient way to lift an entire mechanical system. This would be more useful for a device that needs to be lifted like on a hydraulic machine. This design would be too big for how we plan to utilize a lift system connected to the claw because of the amount of space it takes up, as there would not be an easy way to mount this contraption to the side or entire frame of the trash bin.

3.3.2 Linear Slide

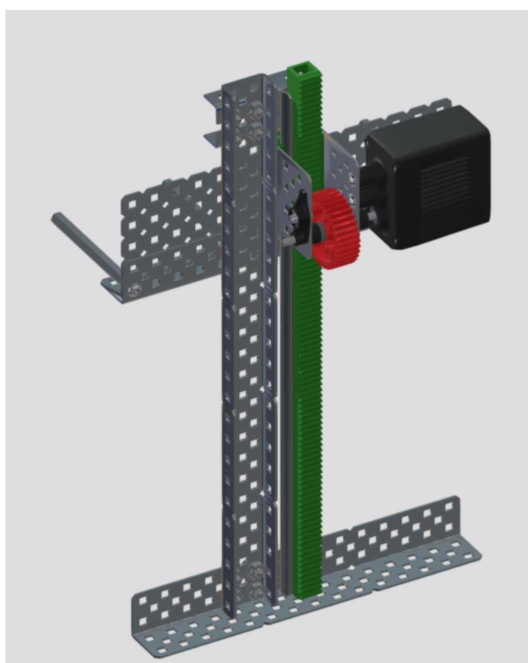


Figure 10: Vex: Linear Slide

The Linear Slide is a vertical track system operated by a gear shaft. The Rack Gear is mounted to a Linear Slide Track, the Rack Bracket is attached to an Acetal Rack Truck, then a motor is connected to the Rack Gearbox Bracket with a spur gear on the shaft. This lets the rack bracket move up and down the Linear Slide track as the spur gear allows

movement on the Rack Gear. This is directly attached to the chassis of the system, additional racks or platforms can be attached for stabilization or functionality.
This choice looks like a good option for our design because it can be directly attached to the body of the robot as the claw is attached to the lift system to move up and down. It is flexible enough to build on the trash bin itself and does not require a platform for the device to work. It would require research into the dimensions of the components and how they should work together within the robot.

3.2.3 Telescoping Lift

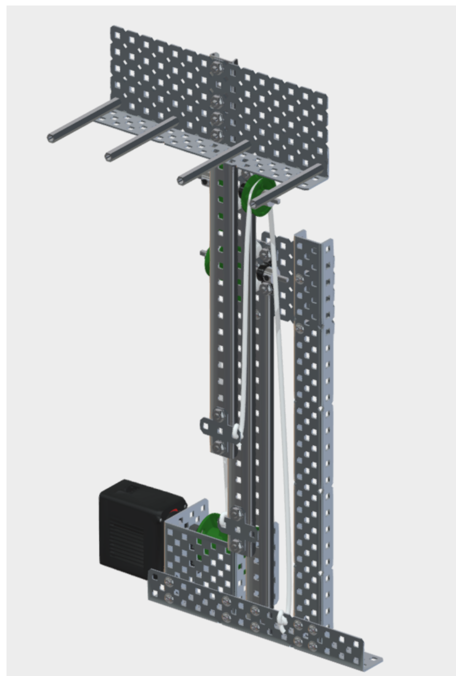


Figure 11: Vex: Telescoping Lift

The Telescoping Lift utilizes pulleys and Winches to power the lift upwards and gravity to move downwards. The pulley is mounted at the top of the Linear Slide Track, this track is attached to the main body chassis, and an Acetal Rack Truck is mounted at the bottom of the second Linear Slide Track. The Linear Track slides into the first track, then a rope from a winch assembly connected to the top of the pulley gets attached to the bottom of the second rack, the rope gets pulled by the winch and the second track moves upwards, this design allows the device to reach great heights.

This could be a mechanism that could possibly work for our design because of the small linear lift design being attached to the chassis of the robot. It allows the claw we built to be placed at the top end of the first track to be lifted, then another joint on the claw will proceed to angle upwards to dump the trash into the main bin. There could be a concern of the bottom of the device having a second rack taking too much room on the trash bin as we must make space for the wheels to have enough clearance to move and not bump into its surrounding while moving. Overall, this device will be considered with the idea of making modifications to the amount of space it takes up on the bottom.

3.2.4 Chain Lift

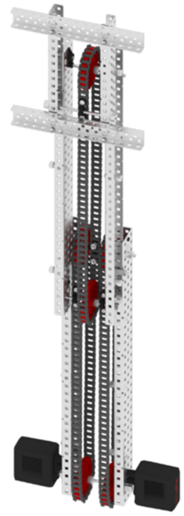


Figure 12: Vex: Chain Lift

Chain Lifts are closely related to telescoping lifts but can be powered to move up and down. These are also sometimes called cascading lifts; these are constructed in a pair to equalize the amount of force on the lift. It is built by having a pair of C-Channels together and connecting a set of C-Channel to the chassis body with a sprocket and chain system in the middle of the two C-Channels. The other pair of C-Channels are connected to the chain and sprocket system, these are powered with one or multiple motors that spin the chain and sprocket, the second pair of C-Channels will slide the first pair of C-Channels moving the system. This design is great for reaching large heights because of the flexibility of the system where the parts are interchangeable to larger C-Channels and chains to modify the design.

This looks like a good option for the design of our project because it is quite compact and can be installed directly on the body of the trash bin where it protrudes minimally to the sides. It could seem a bit bulky to build a square surface onto a rounded surface, but there can always be adjustments made to make new joints to stabilize the mechanism.

In conclusion, the best options are going to be the Linear Slides and the Chain Lifts because of limited space it will take up horizontally and mostly take up vertically which works with the dimensions of the trash bin. We will attempt to use a combination of both and see which design would suit the robot best due to the number of parts that will be attached to the bin, these must be compatible with the claw and work in conjunction with each other, as they have their own independent motor, it should allow the lift to be raised up and down on command by the microcontroller. The Scissor Lift system is too bulky for our design, and it is more suited to picking up an entire device from the bottom up rather than just a single component on a robotic device. It is also one of the most difficult designs to implement because of the required frame, making a constraint of having to think about the lift itself being the main chassis of the device. The Telescoping Lift looks slightly feasible, but the secondary rack sitting on the bottom of the chassis could raise an issue on how it will be mounted on the bottom of the trash bin as well as the amount of space it will take up when the bot is moving around causing possible collision as it turns.

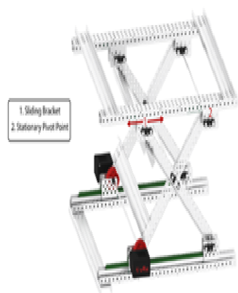
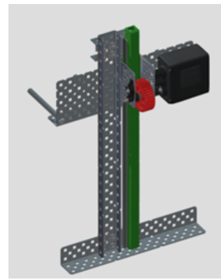
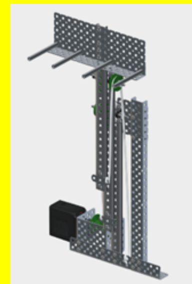
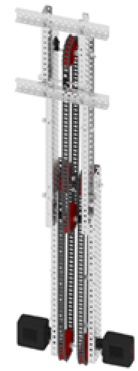
				
Name	Scissor Lift	Linear Slide	Telescoping Lift	Chain Lift
Materials	- Spur gear - Motor - Linear Motion Kit - 84T High strength Gear - 12T Gear	- Linear Motion Kit - Rack Gears - Linear Slide Track - Rack Bracket Acetal Rack Truck - Spur Gears - Power supply	- Linear Motion Kit - Winch Kit - Pulley Kit - Linear Slide Track - Acetal Rack Truck - Rack Truck - Rope	- C Channel - Vex Chassis - Sprocket - Chain system - Motors - Spacers
Practicability to desired design	- Too bulky for our design - Uses small number of gears needed	- Side power supply might be too big to design - Could slide the claw up and down along the track - Shape does not conform to our design	- Claw connected directly to end of lift - Moves along track as intended for our design - Could be modified for claw to change angles	- Could possibly fit our design with the space required - Vertical design - Lot of parts could be complex

Table 5: Lift system comparison

3.3 MCU

The microprocessing unit (MCU), will be the heart of our robot. It will handle the inputs coming from the sensors and camera, and process this information to send the output signals to the motors and lift of the robot. For this project we decided to investigate two types of MCUs, the Arduino and the Raspberry Pi.

3.3.1 Raspberry Pi Zero 2 W

The Raspberry Pi Zero 2 W is a beloved controller around the world. The Raspberry Pi company has allowed their products to operate on the open-source platform, which allows the public to modify, enhance, and develop their product. The Raspberry Pi isn't best described as a microcontroller, but rather as a computer because it has significant computing power that operates on Linux and programs are usually written in Python. The Raspberry Pi Zero 2 W operates with a 64bit quad core 1Ghz Arm Cortex-A53 CPU, and comes equipped with 512MB of SDRAM, 2.4GHz 802.11 b/g/n wireless LAN, a mini HDMI port, a microSD slot, a CSI-2 camera connector, and OpenGL ES 1.1, 2.0 graphics. Not only does the Zero 2 W come packed with powerful features, it only costs $15. Although the Raspberry Pi might seem like the best choice for our project, it might be too powerful for our purposes and we most likely would not be utilizing the full power of the Raspberry Pi. Instead we need a microcontroller that can handle the inputs and outputs of the various sensors and components of the Trashporter. Therefore, we decided to take a look at the Arduino microcontrollers as they are better suited for our project and several of our team members are already familiar with the Arduino.

3.3.2 Arduino Uno Wifi Rev 2

The Arduino Uno is another beloved microcontroller around the world. With tons of tutorials and documentation online, the Arduino is not very hard to pick up and learn. Several of our team members have experience working with the Arduino and using it for this project wouldn't be unfamiliar. We need a microcontroller that is powerful enough to compute our tasks as well as have internet connectivity. The Arduino Uno Wifi Rev 2 is a possible option for our project. This board is equipped with the ATmega4809 microcontroller that comes with 48KB of flash memory, 6KB of SRAM, 256 Bytes of EEPROM and operates at a clock speed of 16Mhz. The Uno is equipped with the u-blox NINA-W102 radio module for Wifi and Bluetooth. Although the Arduino Uno Wifi Rev 2 delivers some great features, it might be a bit over priced at $54, especially when it is compared to other microcontrollers.

3.3.3 NodeMCU ESP32

The ESP32 was developed by Espressif based in Shanghai. Although the ESP32 isn't a Arduino product, the open source and open documentation availability allowed the Arduino community to program the ESP32 with the Arduino IDE. This version of the ESP32 is the ESP32-WROOM-32D that is equipped with 448 KB of ROM, 520KB of SRAM, 16 KB of SRAM for the Real-Time Clock. This board contains the dual-core Xtensa 32-bit LX6 microcontroller, which should run up to 240 Mhz. The ESP32 is superior to the Uno Wifi Rev 2 in terms of hardware and price, the HiLetgo ESP32 cost only $15.

3.3.4 Final MCU Choice

Although the Raspberry Pi might seem like the perfect choice, the team decided to work with the ESP32 because several team members already have experience with the Arduino IDE, and the Raspberry Pi might be overkill for our project. We have decided to use the ESP32 for this project instead of the Arduino Uno Wifi Rev 2 because it is faster and it has more memory to process more complicated code, not to mention it is cheaper than the Uno. A team member already possesses an ESP32-WROOM-32D, which allows us to distribute our budget elsewhere. We can utilize the ESP32 dual core processor to handle the IoT section of our project while processing the inputs and outputs of our robot. The Pixycam 2, which will be explained in later sections, also has its own processor to handle machine learning, which will reduce the stress on our MCU. We have prepared *table 6* below to better visualize the comparisons between the Raspberry Pi Zero 2 W, the ESP32, and the Arduino Uno Wifi Rev 2.

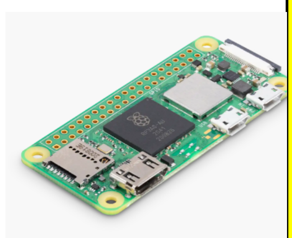
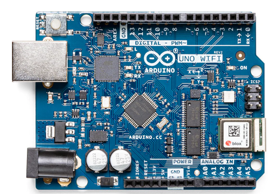
			
Name	Raspberry Pi Zero 2 W	HiLetgo ESP32	Arduino Uno Wifi Rev 2
Processor	1Ghz Quad-Core 64-bit CPU Arm Cortex-A53	240MHz Xtensa dual-core 32-bit LX6 MCU	16 MHz ATmega4809
Operating Voltage	5V	3.3V - 5V	5V
Operating Current	2A	370mA Peak	50mA
Memory	512 MB SDRAM	448KB ROM + 520KB SRAM	6KB SRAM + 48KB Flash
Wireless	2.4Ghz 802.11 b/g/n wireless LAN	2.4Ghz 802.11 b/g/n wireless LAN	2.4Ghz 802.11 b/g/n wireless LAN
Display	Mini HDMI	N/A	N/A
Pins	40	38	32
Dimensions	65mm x 30mm	114mm x 88mm	69mm x 53mm
Cost	$15	$11	$54

Table 6: Comparing several MPUs

3.4 Battery

We are going to need 12 volts to power our motors, we also need 5 volts to power our sensors and MCU. This can be achieved by using a 12 volt power source and a buck converter (explained in a later section) to drop the 12 volts down to 5 volts. A robot tethered to a wall for power is quite inconvenient, therefore we need our power source to be portable. A battery powerful enough to power our project will be perfect. A 12 volt battery source will provide enough power for the Trashporter and majority of 12 volt batteries are for automobiles. Automobile batteries are extremely heavy and will not meet our design specifications. Thankfully, there are 12 volt battery packs that utilize small cylindrical battery cells that are wired in series to create 12 volts. We have prepared *table 7* below to better visualize the comparisons between two battery packs that we have investigated.

		
Name	SPARKOLE 12 V Battery Pack	Tenergy 12 V Battery Pack
Operating Voltage	9-12v	12v
Output Current	4 Amp Max	2 Amp Max
Capacity	5200 mAh	2000 mAh
Max Charge Current	3 Amp	1 Amp
Dimensions	3.14” x 2.7” x 1.5”	2.89”x2.11”x1.2”
Weight	338g	255g
BMS protection	Yes	No
Cost	$34.99	$42.99
Charger included?	No	Yes

Table 7 : Comparing the Tenergy and Sparkle Batteries

Originally we decided to work with the Tenergy 12 V battery pack, however this battery pack lacks a Battery Management System (BMS). A BMS manages important features for batteries, such as; overcharge protection, over discharge protection, cell balancing, and short circuit protection. Without these features a battery may fail and can cause catastrophic damage. After further research we discovered the SPARKOLE 12 V pack, this battery pack offers a BMS, as well as an overall improvement in capacity compared to the Tenergy pack, more than twice the capacity. However, this pack doesn't come with a battery charger, thus we need to purchase a charger separately. In addition, this battery pack seems a little too good to be true because there really isn't a battery pack on the market that has similar features and statistics, therefore further testing will be required of this battery pack. Due to the overwhelming features of the SPARKOLE 12 V battery back we decided to opt for it rather than the Tenergy pack. We hypothesize that the entire project will draw about 1.2 amps of current under max load, and with the 5200 mAh capacity we should have about 3.5 hours of uptime before the battery needs to recharge.

3.5 Motor

There are two types of fundamental motors: AC and DC. We investigated each type of motor to discover how they work, what their benefits and drawbacks are to determine which one would be ideal for our project.

3.5.1 AC Motor

An AC motor transforms AC current or alternating current into rotational motion. *Figure 13* below shows a two-phase AC motor that utilizes two separate AC currents that are 90 degrees out of phase to power two pairs of electromagnetic coils.

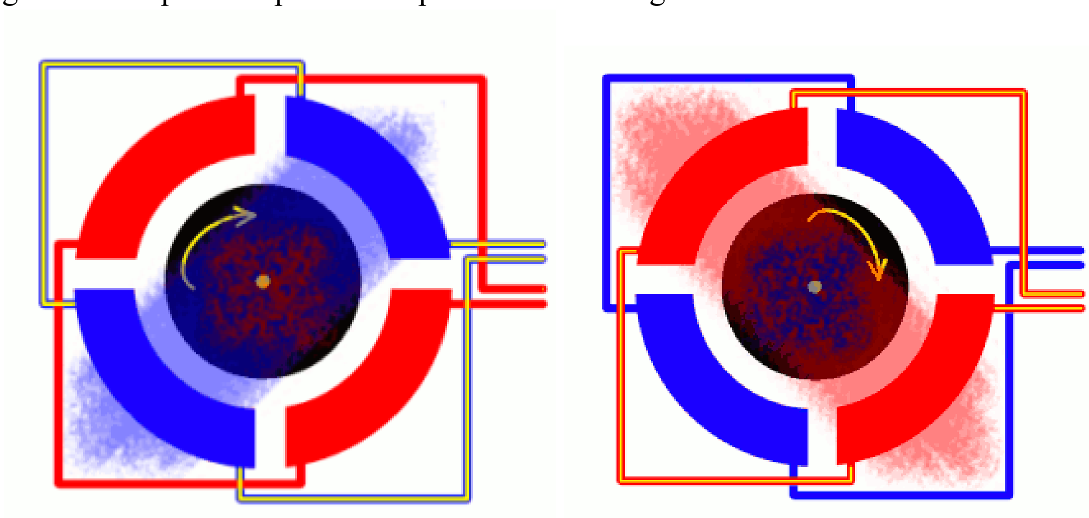


Figure 13: Interaction Between AC current and an AC Motor. Image Courtesy of ExplainthatStuff

Our equipment will be powered by batteries, which can only provide DC or direct current. Converting direct current to alternating current (AC) for an AC motor is inefficient, therefore we decided to use a DC motor for this project.

3.5.2 Brushed DC Motor

A DC motor converts direct current into rotational motion. There are three types of DC motors; brushed DC motor, brushless DC motor, and stepper motor. *Figure 14* demonstrates the components of a brushed DC motor; the stator magnets, windings, brushes, armatures, commutators, and the motor shaft.

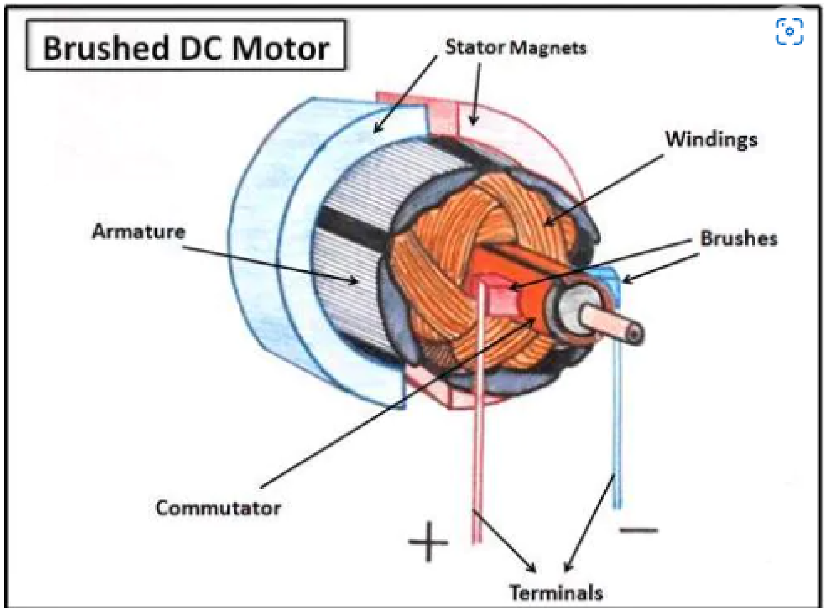


Figure 14: Internals of a Brushed DC Motor. Image Courtesy of The Clemson University

The stator magnets are individual magnets with opposite poles that create a magnetic flux within the stator's housing. The brushes are brushed against the commutators, which are attached to their appropriate coil windings. By applying a direct current to the brushes, they transfer the current to the windings, which, according to the lawrence law, causes an electro-magnetic force. This force, combined with the magnetic flux induced by the stator magnets, turns the motor shaft, producing mechanical motion. Armatures are added to the coil windings to increase the flux within the system to improve efficiency.

3.5.3 Brushless DC Motor

A brushless DC motor is illustrated in *figure 15*, below. Permanent magnets are attached to the motor's shaft also known as the rotor. The stator of the motor is coiled-wrapped. An electromagnet is created by passing current through two opposing side coils. The electro-magnetic force causes the rotor to revolve towards the coils. This procedure is then repeated with another pair of coils, and the DC current is eventually converted into rotational motion using magnetic sensors (hall sensor).

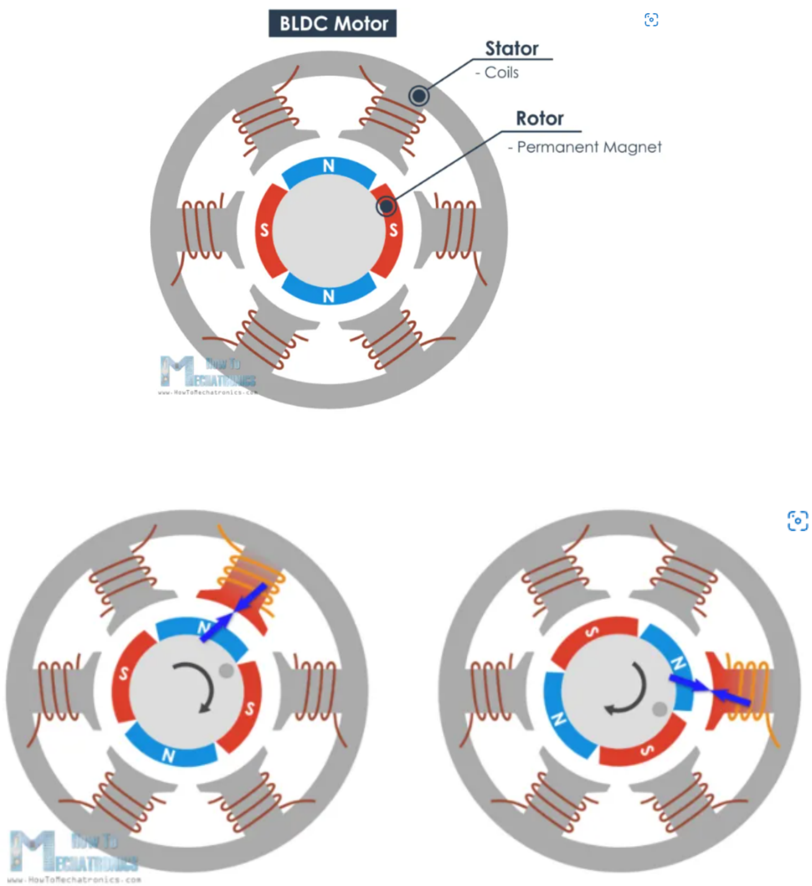


Figure 15: Internals and Demonstration of a Brushless DC Motor. Image Courtesy of Howtomechatronics

3.5.4 Stepper Motor

The internal components of a stepper motor are depicted in *figure 16* below. Brushless DC motors are quite similar stepper motors. The rotors of a stepper motor are made up of a stacked array of metal gears.

Referring to *figure 16* below, applying a current to coils B and B' creates an electro-magnetic force, and the rotor rotates to align teeth 6 and 2 to coils B and B', respectively. Next, Coils D and D' will be energized, causing an electromagnetic influence on the rotor. This forces the rotors to revolve in order to align teeth 1 and 4 with coils D' and D, respectively. This process is repeated for each pair of coils, and the motor rotates 30 degrees, each rotation is referred to as the step angle. The step angle decreases as the number of rotor teeth and coils increases. For example, a stepper motor with 50 rotor teeth and 48 stator teeth has a step angle of 1.8 degrees. The accuracy of a stepper motor increases as the number of teeth increases.

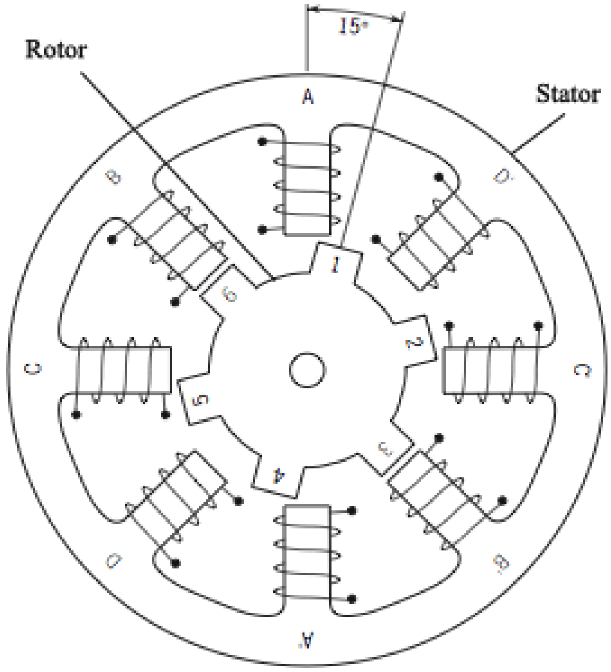


Figure 16: Internals and Demonstration of a Stepper Motor. Image Courtesy of CircuitDigest

3.5.5 Motor Requirements

We have already determined that an AC motor would not be ideal for this project, therefore, we must take a deeper look into what type of DC motor would fit best. Our motors have to perform certain jobs; a pair of identical motors needs to maneuver our entire robot efficiently and effectively (locomotion motor), and another motor must lift the trash bin it intends to pick up (lift motor).

The locomotion motor must meet certain requirements; firstly it must be able to produce enough torque to move the robot at max load (<=15 lbs), secondly it must be able to operate at a max speed of about 0.10 m/s, thirdly the motor must operate in both directions, and finally the movements must be precise, especially when adjusting for pick up. The lift motor must also meet certain requirements; firstly it must produce enough torque to lift the trash, secondly it must hold its position when needed, thirdly the motor must operate in both directions, and finally the movements on the lift must be precise. After we investigated the several types of motors, we needed to determine what type of DC motor we should use. We created a table below to compare and contrast the several types of motors to better visualize and understand our selection.

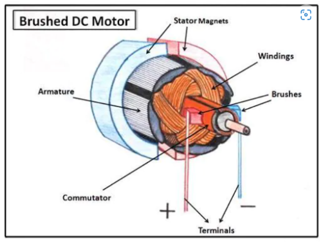
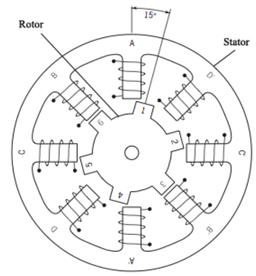
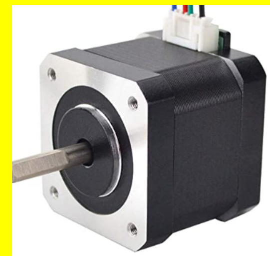
	Brushed DC Motor	Brushless DC Motor	Stepper Motor
			
Pros	⅓ Cheaper than other motors <= $60	Lifetime longer than Brushed Motor	Lifetime longer than Brushed Motor
	No driver board needed, can be controlled by either PWM or Voltage control	High torque at low speeds	Produce high Torque at low speed
	Noisy ~48dB	Quieter than Brushed Motor ~25dB	Holds rated torque at 0 speed
	High RPM 12000 RPM	High RPM 12000 RPM	Low RPM (200-2000 RPM)
Cons	Lifetime is shorter than other motors	Driver Board Needed	Driver Board needed
		Expensive >= $60	Expensive >= $40
			40% less efficient than brushless motor

Table 8 : Table comparing the different types of DC Motors

The brushed motor might seem like the best choice because it does not need a driver board and the physical motor is cheaper than the other DC motors. However, we still need our motors to be precise and hold their torque at certain times. We could use PWM to adjust the speed of a brushed DC motor to create precise movements, however the accuracy of the movement is impacted by different frictional surfaces, and brushed DC motors can't hold its torque at zero speed. After careful review of our options, we decided to work with stepper motors for our lift and locomotion motor because they meet our design requirements.

3.5.6 Motor Specification

Once we decided that the stepper motor was the best choice for our robot, our next step is to decide which stepper motor is the correct choice. Under max load (<= 15 lbs), we want our robot to move at a speed of 4 inches per sec, which is about 0.10 m/s. With this information, we can calculate how much power our motor needs.

1) $Desired\ Speed$: $v = 4\ in/sec = 0.10\ m/s$
2) $Max\ Weight\ Under\ Load$: $M = 15\ lbs = 6.8kg$
3) $Wheel\ Radius$: $r = 0.05m$
4) *Wheel Speed:* $\omega = \frac{v}{r} = \frac{0.10\ m/s}{2\pi(0.05m)} = 0.32\ rps = 19\ rpm$
5) *Thrust Force*: $F = M * g = (6.8\text{kg})(9.81\ \text{m/s}^2) = 66\ \text{N}$
6) *Torque:* $T = F * r = (66\text{N})(0.05\text{m}) = 3.3\text{Nm}$
7) *Power*: $P = \omega T/9550 = (19\ \text{rpm})(3.3\text{Nm})/(9550) = 0.0065\text{KW} = 6.5$ watts

The wheels radius should be about 0.05m (3) and under full load (2) the speed of the robot should be about 0.10 m/s (1). To achieve this speed, the wheels must complete about 19 rotations per minute (4). To overcome the thrust force/weight of the robot, which is about 66 Newtons (5), a torque of 3.3 Newton-meters must be applied on the drive wheels (6). To apply the 3.3 Nm torque on the wheels at a sufficient speed of 0.10m/s, a stepper motor rated at a minimum of 6.5 watts (7) should be used. To reduce wear and stress of our system we decided to shop for two stepper motors each rated at least 3.25 watts of power.

3.5.7 Choosing the Stepper Motor

The table below represents the three different motors that we have researched. After some discussion, we decided to use the Adafruit stepper motor for our project. Although the Adafruit motor is more expensive, it is a higher quality motor. Also noteworthy, is that the Adafruit motor is in fact weaker than the other two motors in terms of its power, we believe we don't need a high power motor for our project, a 1 - 1.5 amp motor will draw too much power and will drain the battery faster.

However, during our future testing phase, if the Adafruit motor falls short of its task, we may need to purchase the STEPPERONLINE 1.5A stepper motor, and return the Adafruit motor to help our budget. After some further research, if we decided to use a more powerful stepper motor, we can limit the current to about 0.7A - 1A with a motor driver, which decreases the power draw from the battery, which reduces losses through heat, and inherently increasing the runtime and lifetime of the Trashporter and the 12V battery pack.

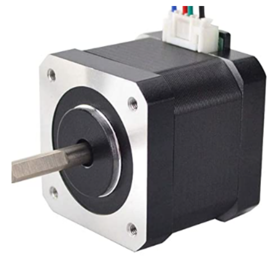
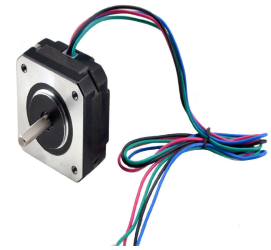
			
Name	Adafruit Stepper Motor	STEPPERONLINE Stepper Motor	STEPPERONLINE Stepper MOtor
Cost	$17.21	$13.99	$10.99
Voltage	12V	12 - 24V	12 - 24V
Max Current	350mA	1.5A	1.0A
Speed	50 RPM	N/A	N/A
Dimensions Inches	3 x 2 x 1	4.45 x 2.95 x 2.72	N/A
Weight	85 grams	280 grams	<80 grams
Source	Amazon	Amazon	Ebay

Table 9 : Table representing three different stepper motor

3.5.8 Motor Driver

Motor drivers are necessary to control the three stepper motors (2 terrain and 1 lift motor) for our project. There are various driver setups for stepper motors, and the Adafruit stepper motor is configured for an H-Bridge driver configuration. The table below displays an H-Bridge stepper motor driver that utilized the L298N chip. We will be needing three driver boards, each powering their own stepper motor, and we will purchase the four bundle driver for $11.49.

In the event that we do decide to use a higher power motor, we can use a motor driver with a current limit like the A4988 stepper motor driver as shown in *table 10* below. By utilizing the current limit, we can control how much power the motor needs. The A4988 driver can limit how much current is being drawn from our battery pack, which can increase efficiency and allow the Trashporter to run longer. We will want to use a motor driver as we can control how much power the motors will use by setting it to our desired voltage. By doing this, we will be able to save power from our batteries which will be connected to our PCB board.

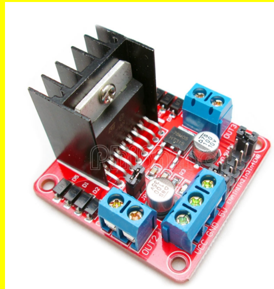
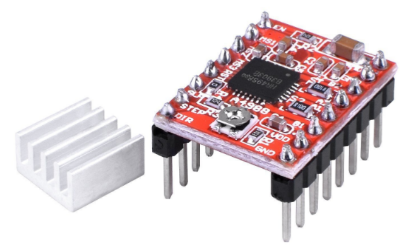
	HiLetgo L298N Stepper Motor Driver 	HiLetgo 5pcs A4988 Stepstick Stepper Motor Driver 
Logic Voltage	5 V	3 - 5.5V
Logic Current	0mA - 36mA	N/A
Operation Mode	H-Bridge	N/A
Drive Voltage	5V - 35V	8V - 35V
Drive Current	2A Max	1A without heatsink 2A with heatsink
Dimensions (mm)	43x43x27	1.5x2
Microstepping	No	Full, 1/2, 1/4, 1/8, and 1/16
Cost	$11.49 for 4	$9.89 for 5 with Heatsink

Table 10 : HiLetgo L298N Stepper Motor Driver Specifications

3.6 Object Detection Sensor

In general, there are seven types of technologies to sense and detect objects. They are electro-mechanical, pneumatic, magnetic, inductive, capacitive, photoelectric, and ultrasonic sensors. Electro-mechanical sensors use mechanical actuators to detect objects by switching the states (polepositionmarketing@kellertechnology.com, 2021). Pneumatic sensors generate compressed air and use diaphragm valves to sense the change in air pressure when an object is close. Magnetic sensors detect magnetic objects within a specific range. Inductive sensors utilize electromagnetic fields to detect metallic objects. Capacitive sensors utilize electrostatic fields in the air to detect objects such as wood, paper, plastic, and more. Photoelectric sensors detect objects by the reflection of the transmitted light beam. Ultrasonic sensors use sound wave reflections to detect a wide range of dense objects except foam and cloth materials that can absorb sound waves.

Looking into distance measurement technologies with Lidar, ultrasonic sensors, and radar. Lidar stands for Light Detection And Ranging. They can be used in the air or on the ground. For use on the ground, the wavelength of the lasers would range from 500 to

600 nanometers (Hani, 2021). Lidar can detect the position, size, and shape of the objects. It usually works alongside GPS, a camera, and IMU sensor. Lidar in general is expensive compared to ultrasonic sensors. Lidar uses laser beams to detect objects rather than sound waves used by ultrasonic sensors. Thus, they will produce detailed data about the objects detected like creating 3D object models with static environments and object contour and topography.

They have a wide measuring range and good accuracy. They use smaller wavelengths as opposed to sonar and radar. Their narrow point detection can miss small glasses or little things on the floor. Lidar will not work under the condition of heavy rain, snow, and fog. However, these issues will not be a problem since the robot will operate inside a room rather than going outside. They can also be harmful to our eyes. For lidar, the distance is calculated using the speed of the light and the time duration of sending and receiving the signal.

Radar stands for Radio Detection And Ranging. Instead of using light wave pulses like lidar and sound waves like ultrasonic sensors, radar uses radio waves to detect objects. They usually have a fixed or rotating antenna attached. However, due to the size of the antenna, they are not the best to use for the purpose of our robotic design. The transmitter of the radar generates the electromagnetic wave while the antenna sends out the wave, so, the transmitter and the antenna have to work together. With the long wavelength of radar, the data results will not be as detailed as lidar since lidar sensors can produce detailed data with high accuracy and resolution to create 3D models of objects. Instead, radar can be used for 2D imaging.

To choose the right radar, the choice of the frequency is critical as it affects the wavelength, detection range, and the antenna size. Also, the image resolution is influenced by the frequency and antenna size. For short range detection and high resolution data, it needs a high frequency and attenuation and granular detection with low power. There are advantages to radar sensors. With their long wavelength, they are able to detect objects in long distances compared to ultrasonic sensors and lidar. They help detect large objects and moving targets. They can operate in bad weather conditions compared to lidar. Radar results are less impacted by temperature than ultrasonic sensors.

Sonar stands for Sound Navigation And Ranging and it is mainly used for underwater detection using low frequencies ranging from 100 to 500 Hz. Ultrasonic sensors are like the same thing as sonar in terms of how they work. However, ultrasonic sensors are used on land and use only high frequency sounds. Ultrasonic sensors would produce erroneous readings when dust and air-borne particles interfere with the sound waves emitted by the ultrasonic sensor.

In the table below, ultrasonic sensors, lidar, and radar will be compared to identify the best technology to use for the design of the Trashporter robot.

	Ultrasonic Sensors	Lidar	Radar
Type of Emission	Sound Waves	Light Waves	Electromagnetic Waves
High Reading Frequency	No	Yes	-
Range Detection	Small	Small	Long
3D Imaging Compatible	No	Yes	No
Resolution	High	High	Low
Size	Small	Medium	Big
Cost	Low	High	Medium

Table 11: Comparing Different Object Detection Sensors

Concluding from the table above, lidar sensors are for detecting to get to know all the details of the object, ultrasonic sensors are used for short range detection, and radar sensors are used for long range detection. The purpose of using object detection sensors in this robot is to detect walls and big objects to keep the robot from crashing into them. Thus, only short-range detection is needed. Meanwhile, it is better to keep the size to be small on the robot. Looking at the features of ultrasonic sensors, they fit what our robot needs in a small and simple form. Out of all the different types of object sensors, ultrasonic sensors best suit the robotic design. Ultrasonic sensors are small and fast. They are immune to colors and transparency of objects, dust, moisture, and ambient light. They are great at detecting ranges and can act as proximity sensors. Looking at the specifications of this project, this sensor will be needed to restrict the robot from bumping into walls and obstacles. Ultrasonic sensors are also our cheapest option out of the three choices.

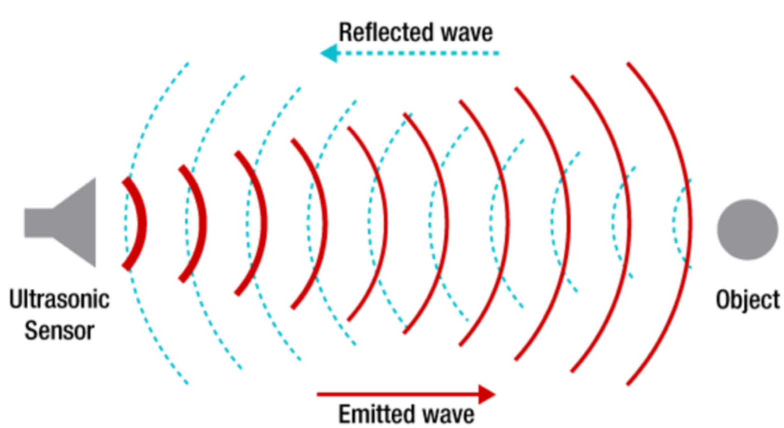


Figure 17: Principle of Ultrasonic Sensors. Image Courtesy of Texas Instruments

Looking specifically at ultrasonic sensors, they generate high frequency sound waves (usually above human hearing frequency range) and send them out to detect the object from the echo reflected by the object (See Figure 16). The distance is obtained through the time between the emission and reception of the ultrasonic wave. That means the time is proportional to the distance between the sensor and the object. Thus, the relationship between the time and distance of the sensor to the object is linear. This makes the principle easy to understand and implement.

The advantages of ultrasonic sensors include their sensory independence of the colors, sizes, materials, transparency, surface texture, and optical reflection of the object. They are accurate in detection measurements. They are endurable in dirty and dusty environments. The detection is used without any physical contact involved. Their cuboid and cylinder structure provide more degrees of freedom (Calin, 2020). On the contrary, the disadvantages of ultrasonic sensors include they have a minimum detection distance requirement. The sensor gets better output with objects that have high surface density, thus, foam and cloth materials with less surface density won't be easily detected due to their ability to absorb sound waves. The environmental factors such as humidity and pressure can affect the sensor's response. Loud noises can cause the sensor to produce false responses. Despite these disadvantages, the robotic design and specifications won't be greatly affected by them. For the design scenario, the robot will operate in an open space room of a house. Thus, humidity, pressure, and loud noise would not be a big problem on the performance of the sensor. Small foams and cloth should not be an issue for the robot since it will mainly detect walls, trash bins, and obstacles that can damage the robot when it comes in contact. Therefore, it is okay to work with ultrasonic sensors in this robotic design.

While it is decided that ultrasonic sensors are the best device to use for the robot to detect walls, trash bins, and obstacles, there are four different types of ultrasonic sensors to choose from. They are ultrasonic proximity sensors, ultrasonic two point proximity switches, ultrasonic retro-reflective sensors, ultrasonic through beam sensors (Lab, 2017). With ultrasonic proximity sensors, the time between emitting and receiving the sound wave and the distance between the sensor and the object are inversely proportional. The sensor output is obtained only within a certain detection range, this can be adjusted with the sensor's potentiometer. The concept of how the ultrasonic proximity sensor is shown in Figure 17 below. Ultrasonic two point proximity switch is similar to the ultrasonic proximity sensor except it has two switches that can be programmed within the sensing range. Ultrasonic retro-reflective sensor uses the propagation time to measure the distance to the object. A stationary object would be chosen as the reflector point to determine the maximum range of detection. When the object appears within the range of detection, it changes the propagation and activates the sensor. This sensor can also detect sound absorbent objects and objects that can deflect sound waves. The concept of how the ultrasonic retro-reflective sensor is shown in Figure 18 below. Ultrasonic through beam sensor places the transmitter and receiver on the opposite side and are in two different housings while the other three ultrasonic sensors have the transmitter and receiver on the same device (2022). Ultrasonic through beam sensors detect objects when the objects interrupt the sound waves. This sensor gives the shortest response time and longer detection ranges compared to the other three ultrasonic sensors. The concept of how the ultrasonic through beam sensor is shown in Figure 19 below.

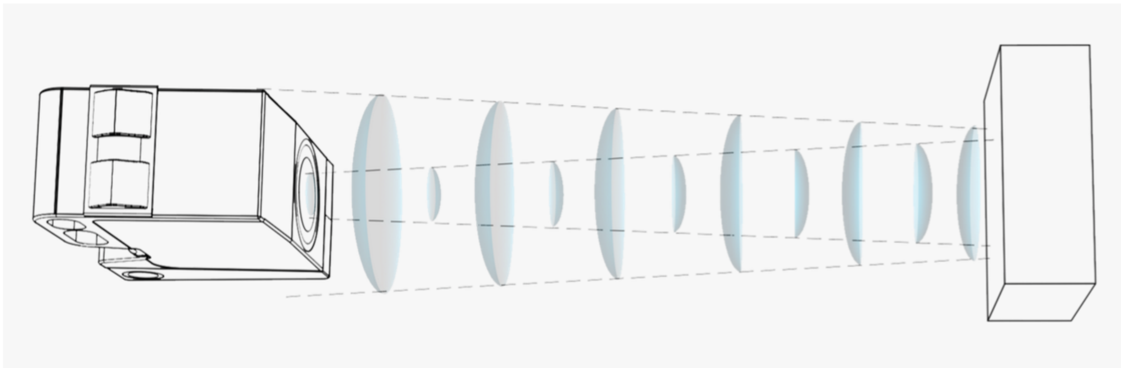


Figure 18: Principle of Ultrasonic Proximity Sensor. Image Courtesy of Baumer

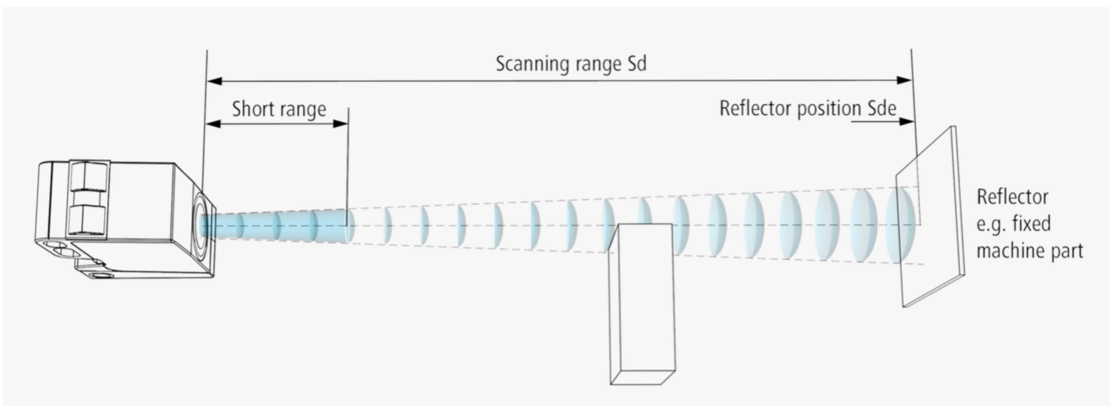


Figure 19: Principle of Ultrasonic Retro-Reflective Sensor. Image Courtesy of Baumer

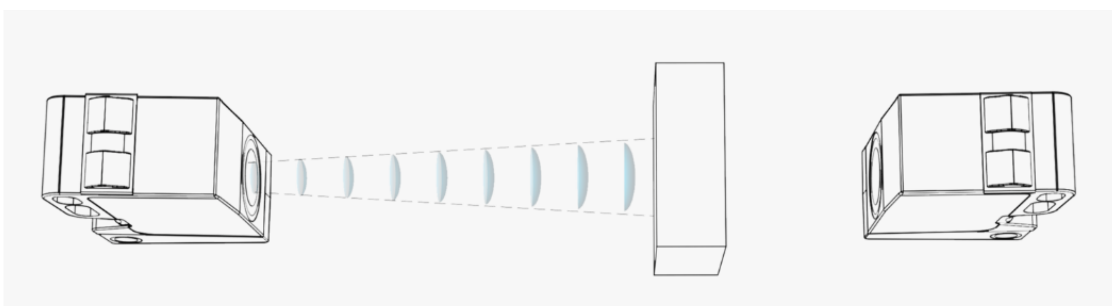


Figure 20: Principle of Ultrasonic Through Beam Sensors. Image Courtesy of Baumer

Looking at the four different types of ultrasonic sensors mentioned above, it is best for the robot to use the ultrasonic proximity sensors or ultrasonic retro-reflective sensors. The ultrasonic retro-reflective sensor is better at detecting objects that can absorb or deflect sound waves than the ultrasonic proximity sensor. It is also a bit more complex than the ultrasonic proximity sensor with its functions and features. However, the robotic design does not necessarily need a sensor to detect specific objects but big dense objects like walls and trash bins. The ultrasonic proximity sensor just fits the needs and goals of the robotic design, for object detection and distance measurements. Thus, to keep things simple, ultrasonic proximity sensors will be used. The different ultrasonic proximity sensors can be seen compared in the table below.

	CUS Ultrasonic Proximity Sensors	**HC-SR04 Ultrasonic Distance Sensor**	**SU Series Ultrasonic Sensors**	**TU Series Ultrasonic Sensors**
Features	- Housed in compact aluminum or plastic cases - Featuring through-hole, wire leads, or wire leads with connector mounting styles - Deal with moisture and environmental contaminants - Options of transmitter, receiver, and transceiver	- Screw hole diameter: 1 mm - Compatible with Arduino and Raspberry Pi	- M18 (0.71”) plastic - High resolution - Complete overload protection - 2 DC models with adjustable sensitivity and LED status indicator - 3 analog models	- M30 (1.18”) plastic - High resolution - Complete overload protection - DC output model with adjustable sensitivity and LED status indicator
Sensing Range	0” to 708”	0.78” to 157.48”	3.94” to 59.06”	11.81” to 98.43”
Ultrasonic Frequency	23 KHz to 400 KHz	40 KHz	180 KHz, 300 KHz	130 KHz
Ultrasonic Beam Angle	75 to 80 degrees	30 degrees	8 degrees	8 degrees
Max. Response Time	-	-	150 ms	100 ms
Load Current		15 mA	≤ 5 mA	≤ 5 mA
Operating Voltage	80 to 180 V (DC)	3 to 5 V (DC)	15 to 30 V (DC)	19 to 30 V (DC)

Output Type	Analog	0 to 5 V (DC)	DC: PNP logic Analog: 0 - 10 V (DC)	DC: PNP logic Analog: 0 - 10 V (DC)
Output Connection Type	-	-	- 6.5’ axial cable - 0.47” connector	- 0.47” connector
Temperature Range	-	5 to 158 F	-13 to 158 F	-13 to 158 F
Temperature Compensation	-	-	Yes	Yes
Tightening Torque	-	-	2.21 lb./ft.	2.21 lb./ft.
Dimension	0.49” x 0.37”	1.77” x 0.78” x 0.59”	5.47” x 0.71”	4.88” x 1.18”
Weight	< ~2 g	8.5 g	54 g	124 g
Cost	< ~5 ($)	0 - 4.50 ($)	305 - 333 ($)	$357

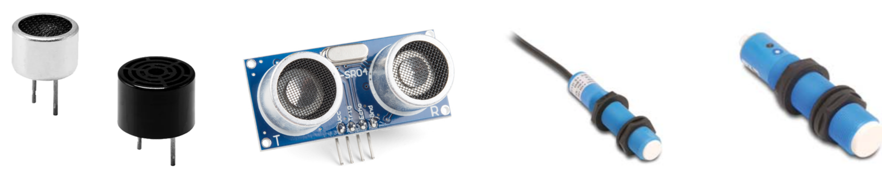


Table 12: Table Comparing Different Ultrasonic Proximity Sensors

Since we had experiences in working with HC-SR04 Ultrasonic Distance Sensor in Junior Design before and it is compatible with Arduino, it is prioritized to use at first in the list. However, the disadvantage of it is the size of it is too big and adds weight to the robot. Also, it would have to be placed on the tip on the sides of the robot to be able to get better distance reading. On the other side, the CUS Ultrasonic Proximity Sensors have decent small sizes, making them easier to place on the robot. However, the CUS Ultrasonic Proximity Sensors would have to be placed on the PCB as a SMD. In this case, we are not sure of the PCB board placement yet, which will affect the function of the sensor in sensing objects. Plus, we would need at least four ultrasonic proximity sensors on four sides of the robot. This would limit us, since we don’t want a big PCB board on the robot nor having four small PCB boards dedicated to ultrasonic proximity sensors as both strategies add a lot of unnecessary weight onto the robot. With the SU and TU Series Ultrasonic Sensors, we won’t take consideration in using them due to their expensive cost, large dimensions, and heavier weight. Comparing HC-SR04 Ultrasonic Distance Sensors to the SU and TU Series Ultrasonic Sensors, HC-SR04 Ultrasonic Distance Sensors have better advantages in compatibility, sensing range, beam angle, voltage, dimension, weight, and cost. This also makes HC-SR04 Ultrasonic Distance Sensors a better part to use in our robotic design with the design specifications.

Furthermore, to determine the right ultrasonic proximity sensor, the sensing distance required don't have to be too large as the space would be limited in a room. Thus, the sensing distance can be as small as one inch. Also, we have to take into consideration to place the sensors as far away from heat sources as they can cause the sensors to fail to work. Thus, for the CUS Ultrasonic Proximity Sensor SMDs, they would either be placed far away from the power supplies on the PCB since they emit ambient heat to their surroundings or on a separate PCB. Also, temperature sensors might be needed to detect high temperatures inside and outside of the robot to prevent damages to the sensors, components, and the robot. In addition, the robot would have to be designed without any parts that can get in the way of the ultrasonic sensor or else it would affect the actual detection of objects.

On top of considering the weight, distance, and placement variables for the ultrasonic proximity sensors, we need to consider having a shielded or unshielded ultrasonic sensor. For the robot design, it is best to use shielded or embedded ultrasonic proximity sensors for electrical protection and flush mounting. Factors and variables for later that will affect which ultrasonic proximity sensor to use are the 1, 2, 3, or 4 -wire discrete outputs, what the sensor output will be connected to, and the output connection type. Overall, HC-SR04 Ultrasonic Distance Sensor will be the first in our list to use and then the CUS Ultrasonic Proximity Sensors. As a backup plan, there are other ultrasonic proximity sensors that can be looked into if the two mentioned do not work or work well with our robotic design. They include uxcell Ultrasonic Module US-100 Distance Sensor, Adafruit 3942 HC-SR04 Ultrasonic Sonar Distance Sensor + 2 x 10K resistors, and Ultrasonic Sensor for Robots - People Detection.

3.7 Trash Bin Monitoring System

In this section, it will look into two methods of monitoring the weight and the level of garbage filled in the trash bin. The first method is to choose weight sensors and height sensors individually so we will have more control over designing and what to do with the measurement values. The most trouble of this method would likely be interfacing and moving the data onto the software. The second method would be getting an IOT garbage monitoring system that is already designed and fits the purpose of what is needed to control the trash bin. However, there will not be too much freedom on our side to change features and functions if we need to. In general, the purpose of this trash bin monitoring system is to notify the user when the weight limit or the height level limit are met to be able to empty the trash bin on the robot for the robot to continue to do its task.

3.7.1 Weight Sensor

Weight sensor is a weight transducer that converts the mechanical input such as load, weight, tension, compression, force, torque, or pressure into electrical output signals (Futek).

There are ten types of weight sensors. They are photoelectric, hydraulic, electromagnetic force, capacitive, magnetic pole variation form, vibration, gyro, resistance strain, annular plate, and digital weight sensors (R, 2020). Photoelectric weight sensors can be optical grating or code disc type. Photoelectric weight sensors work by converting angular displacement into photoelectric signals and further convert into electric signals. Hydraulic weight sensors measure by the gravitational pull on the objects. As the gravitational force increases, the pressure of the hydraulic oil also increases and in a proportionally way to the gravitational force. Hydraulic weight sensors are simple, firm, and measure different ranges. However, the accuracy is not that great. Capacitive weight sensors are measured by the proportional relationship between the oscillation frequency of the capacitor oscillation circuit and the distance between two electrode plates. One electrode plate is fixed and the other electrode plate can move. When the objects are placed on the weighing platform, the leaf spring deflect and cause the distance between the two electrode plates to change, which in return changes the oscillation frequency of the circuit. Capacitive weight sensors use less power and are cheaper to buy. They are also higher in accuracy compared to the hydraulic weight sensors. Electromagnetic force weight sensors measure with electromagnetic balance force. As the object is placed on the weighing platform, one side of the lever will tilt upward. The tilt is detected by the photoelectric component, then amplified and creates electromagnetic force through a coil to return the lever to the balanced state. The weight measurement is then quantified by the current created to generate the electromagnetic balance force. Electromagnetic force weight sensors have a high accuracy in measurement but the measurement ranges are within milligrams and ten kilograms, that is under 22 pounds. We are looking for a decent amount of trash, which with an average trash bag weighs around 22 pounds. That needs a bit of a wiggle room for sensor measurements. Thus, electromagnetic force weight sensors will be eliminated from the part's list since the robotic design requires the robot to pick up items in pounds of twenties.

Magnetic pole variation weight sensors measure by the voltage change generated by the ferromagnetic element. The ferromagnetic element is changed by the weight of the object. This is due to the internal stress created by the object which changes the permeability of the ferromagnetic element or magnetic poles and further causes the induced voltage on both sides of the ferromagnetic element's coil to change. Magnetic pole variation weight sensors also do not have a high accuracy as the capacity weight sensor. However, it does have the ability to measure various weights ranging from pounds to tons. Vibrating weight sensors measure by the proportional relationship between the natural vibration frequency and the square root of the applied force on the elastic material. Vibrating weight sensors also have two types: vibrating wire and tuning fork type. Vibrating weight sensors have extremely high accuracy and high cost as well. The weight measurements range from hundred grams to hundred kilograms. However, taking into consideration the ultrasonic distance sensor, vibrating weight sensor and the ultrasonic distance sensor may interfere with each other's measurements in terms of the close relationship between frequency of sound and vibration. Thus, the vibrating weight sensors will be crossed off from the list for weight sensors. Gyro weight sensors measure by the proportional relationship between the force applied and the precession angular velocity. They have a high accuracy in measuring the frequency and fast response time.

The basic principle behind the resistance strain weight sensor measurement is the change in electric resistance with applied forces. To understand how weight sensors work deeply, we need to understand the physics and working principle of strain gauge or strain gauge. Strain gauges are electrical conductors and are tightly attached to the zigzag lines on the film shown in the figure below. The conductors stretch when the film is pulled and contract when the film is pushed. This change in the shape causes the resistance of electrical conductors to change. Thus, the load cell or transducer is determined by the strain gauge resistance change. Strain gauge resistance increases with the stretch or applied force and decreases with contraction. The weight sensor is usually metal made of aluminum or stainless steel. This material helps it to hold out against high loads with elasticity to return to its shape without deformation when forces are applied. Annular plate weight sensors have high precision and long-term stability but low accuracy. Digital weight sensors use gravity measurements and turn them into electrical signals. They are simple and have good adaptivity.

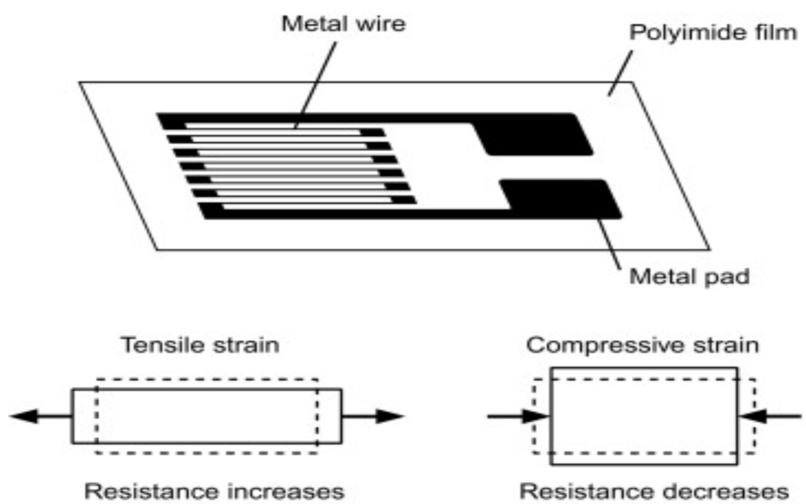


Figure 21: Principle of Weight Sensors. Image Courtesy of Futek

In general, electromagnetic force weight sensors, magnetic pole variation weight sensors, vibrating weight sensors, and annular plate weight sensors can be eliminated from the list of parts to use for weight sensors. Electromagnetic force weight sensors and magnetic pole variation weight sensors can produce the wrong measurements since we are looking into using metal materials for making the frame of the robot, which have the potential to interfere and affect how these sensors take measurements. Annular plate weight sensors have low accuracy due to the error of calculation and they are mainly used for heavy objects. Plus, they are also very expensive to obtain. Thus, that leaves the choices of photoelectric weight sensor, hydraulic weight sensor, capacitive weight sensor, gyro weight sensor, and resistance strain weight sensor.

Overall, due to the large weight range measurement and the big cost of material, photoelectric weight sensor, hydraulic weight sensor, capacitive weight sensor, and gyro weight sensor will not be considered to be used. This is also due to the fact that they are used in specific fields too. In summary, resistance strain weight sensors will be looked into to use in the robot. This is because resistance strain weight sensors are popularly used and the cost of them are pretty cheap and manageable for this team. They are high in accuracy, reliability, performance, and are not much influenced by temperature. They also have a wide range of capacity, from 10 grams to 100,000 pounds (Futek).

Looking more into resistance strain weight sensors, they are also referred to as load cells. There are eight types of load cells. They are in-line load cells, column load cells, load buttons, S-beam load cells, thru-hole load cells, pancake load cells, and rod-end load cells. The in-line load cells are used to measure tension and compression. They have male thread for mounting. They are high in accuracy and endurance. However, they are used in press applications, which won't be applicable in this robotic design. Column load cells offer the weight range measurement from 2,000 to 30,000 pounds (Futek). Thus, column load cells won't be used since the weight measurement range exceeds what we expected.

Load button has a button on top to measure the compressive force. This won't be ideal for the robotic design since the trash bin will be laying directly on top of the frame of the robot. Using this load button will offset the balance of the trash bin and affect the accuracy of the measurement. Thus, the load button will not be a strong candidate to use for the weight sensor.

S-beam load cells or Z-beam load transducers have female thread for mounting. They are high in accuracy and mainly used in in-line processing and automatic control feedback practice. Thru-hole load cells or washer load cells have a hole in the middle and a rod goes through it to measure the weight. They are mainly used for bolt-loading. Thus, thru-hole load cells won't be very applicable to our robotic design, also due to the same concern as the load button and there's no way of using the rod. Pancake load cells look the same as the thru-hole load cells, but with a smaller hole in the center and more holes around the edge of its circular shape. Rod-end load cells have the option of male and female thread for mounting. This will be useful for fixing the position of the sensor on the robot.

To select the right weight sensor for this robotic design, we need to consider how we are measuring the weight of the trash bin on the robot, the way of mounting the sensor, the minimum and maximum weight capacity, the size and weight of the sensor, and the output type of the sensor. First of all, the weight sensor should be mounted at a fixed position on the robot. The weight capacity of the weight sensor should be ranging from zero to around 50 pounds. The dimensions and the weight of the weight sensor should be small to be easily placed on the robot.

In the table below, two different types of load cells are compared. Both of them use a HX711 A/D converter. However, they both have different strategies in terms of how they measure the weight of the trash in the trash can.

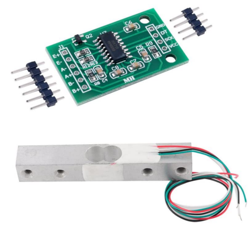
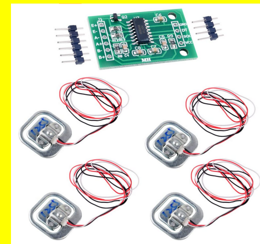
		
Name	**20 kg Load Cell with HX711 Load Cell Amplifier**	**50 kg Half-Bridge Strain Gauge Load Cell with HX711 A/D Converter**
Features	- Two selectable differential input channels - Low noise - On-chip oscillator, power supply regulator, and power-on-reset - Compatible with Arduino	- 3 different ways of using it - Low noise - On-chip oscillator, power supply regulator, and power-on-reset - Compatible with Arduino
Weight Capacity	20 kg/44 pounds	50 kg/110 pounds
Operating Voltage	2.6 - 5.5 V	2.6 - 5.5 V
Operating Temperature Range	-20 to 85 degrees	-20 to 85 degrees
Current Consumption	-	< 1.5 mA
Dimensions	0.5” x 0.5” x 3.15”	1.33” x 1.33”
Weight	36 g	28 g
Cost	$9.99	$8.99

Table 13: Weight Sensor Comparisons

Comparing the weight sensors shown in the table above, the 20 kg load cell and 50 kg half-bridge strain gauge load cell have a lot of similar features. However, they do differ in the weight capacity range, dimensions, weight, and cost. The 50 kg half-bridge strain gauge load cell is more in favor compared to the 20 kg load cell as of their ways of measuring the weight. The 20 kg load cell needs to be set up like a weight scale for vegetables and food like shown in Figure 21 below while the 50 kg half-bridge strain gauge load cell needs to be set up like a weight scale for body weight like shown in Figure 22 below. The weighing platforms and the frame foundation like shown in the two figures below won’t be able to be purchased. Thus, we will need to make them by 3D printing or other technologies. For the 20 kg load cell, it needs more parts than the 50 kg half-bridge strain gauge load cell. It needs a supporting leg under the sensor and the weighing platform on top of the sensor to be able to measure the weight as shown in Figure 21. The 50 kg half-bridge strain gauge load cell only needs the weighing platform. Thus, it will be easier for setting up the 50 kg half-bridge strain gauge load cell than the 20 kg load cell for measuring the weight, since it needs less part. Furthermore, it will be much more difficult to set up the 20 kg load cell weighing technique on top of the robot since the 20 kg weight sensor does not have a complete solid platform set up under it. Thus, the 20 kg weight sensor could potentially break into two when the weight is exceeded and the measurement could be impacted by the movement of the robot going to places since the sensor would not be stably fixed, as the 20 kg weight sensor can shake up and down as the robot moves. Furthermore, the weight and cost of the 50 kg half-bridge strain gauge load cell is lower than the 20 kg load cell. The 50 kg half-bridge strain gauge load cells will just need to be placed under the big weighing platform for weight measurements. The placement of the 50 kg half-bridge strain gauge load cells under one big weighing platform is a lot easier than placing the 20 kg load cell in between two platforms. In conclusion, the 50 kg half-bridge strain gauge load cell will be used instead of the 20 kg load cell due to its easy weighing method, light weight, and less cost.

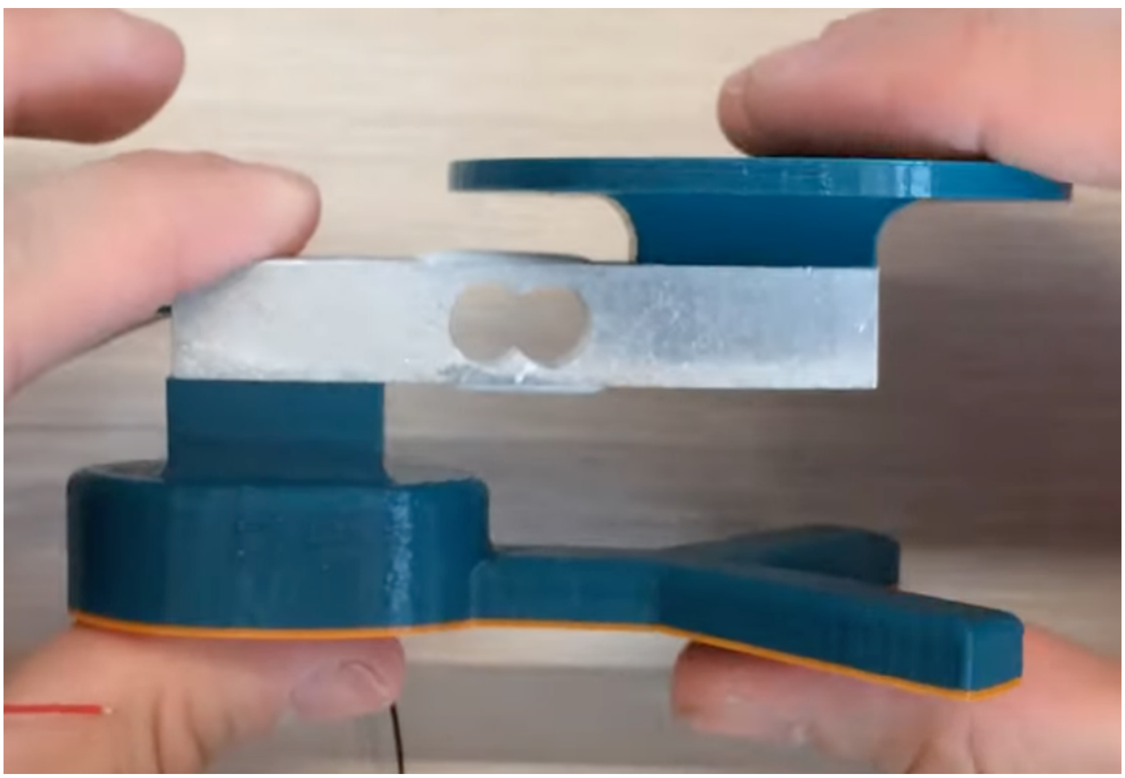


Figure 22: Set Up of the 20 kg Load Cell

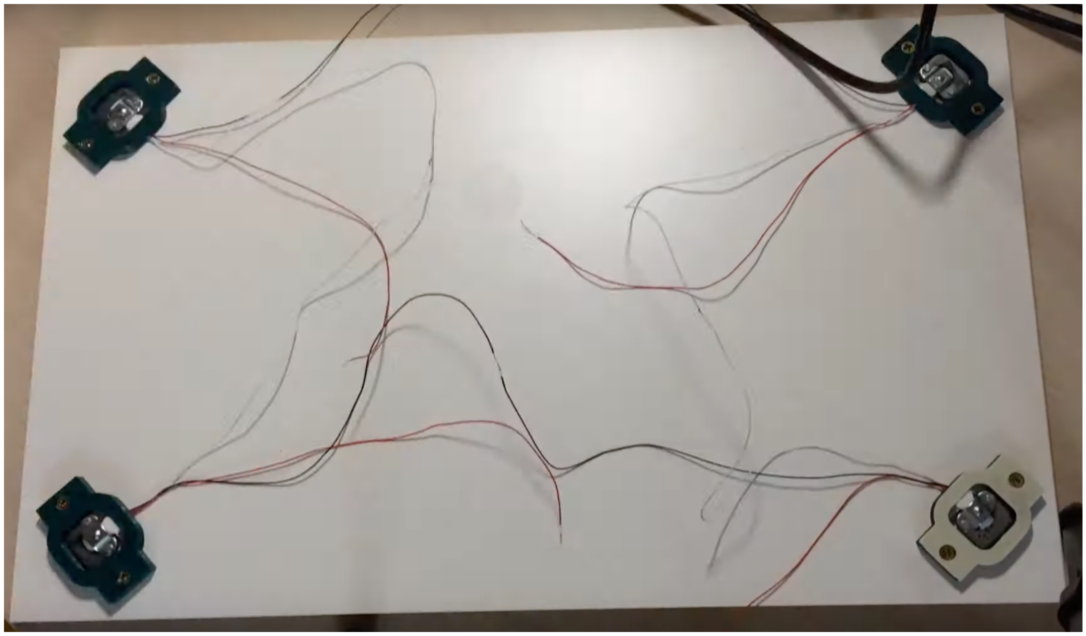


Figure 23: Set Up of the 50 kg Half-Bridge Strain Gauge Load Cell

For weight sensors, they will have measurement errors no matter what. The errors can be caused by the zero balance of the weight sensor, linearity or non-linearity output characteristic curve, hysteresis, repeatability or non-repeatability of the inputs, and temperature shift. To reduce the errors mentioned above, we can do weight sensor calibration to reduce the error measurements with hysteresis, repeatability, and temperature shift. All of the weight sensors should have a calibration curve or characteristic curve like shown in the graph below. They define the sensor's output to the input. To calibrate the weight sensor, we need to make sure the zero offset of the weight sensor falls under the reference weights.

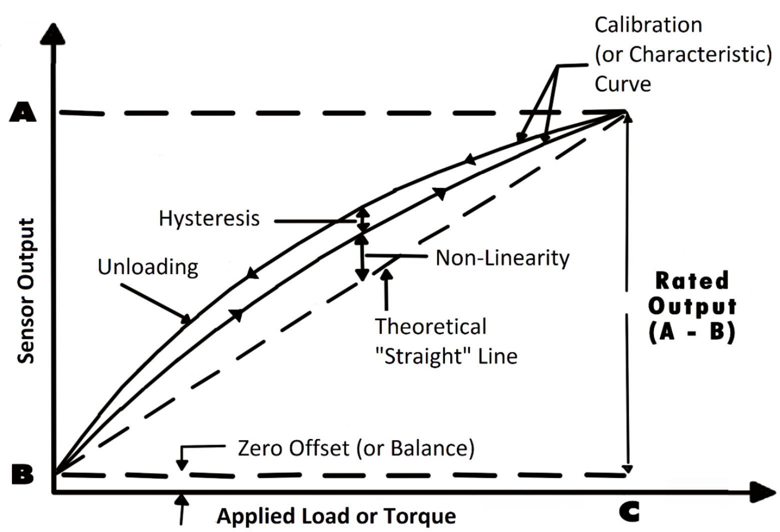


Figure 24: Weight Sensor Calibration Curve. Image Courtesy of Futek

3.7.2 Height Sensor

In order to detect the amount of trash in the trash bin, we need to check for its weight and height. The ultrasonic sensors discussed above can also be used as height sensors since it measures distance. Usually, weight sensors and height sensors have a little error in terms of measurements but also close enough to the correct measurements. There are two options of monitoring the weight and height of the trash can to detect when the trash can becomes full, either place the weight sensor and height sensor or implement the IOT garbage monitoring with weight sensing. As the weight sensor and height sensor method have been mentioned above, IOT garbage monitoring system will be discussed next. This could be a backup plan in case the weight sensor and height sensor we selected don't work well together.

3.7.3 IOT Garbage Monitoring

IOT garbage monitoring with weight sensing will be able to measure the weight and know when the trash reaches the top of the trash bin. The system collects information on the trash bin and displays the data over on a web. The IOT garbage monitoring system places ultrasonic sensors over the trash bin to detect the garbage level by comparing the depth of the trash bin. It also has weight sensors below the garbage bin. The IOT monitoring system uses the internet to send over the weight and the height of the garbage filled in the trash bin.

The foundation of how this system works also matches up with what we plan to do with monitoring the trash bin and conducting its tasks. The combo of height and weight measurements can limit each other to prevent the overweighting and overfilling the trash bin. When the trash bin is not filled, but the weight has reached the maximum of what the robot can carry, the robot should stop looking for more trash bins in the room and put an alert on the user's phone to warn that the trash in the trash bin on the robot needs to be taken out. Also, when the height of the garbage has reached the top of the trash bin, the robot no longer should look to pick up more trash from other trash bins and stop executing any more tasks or return to its home position.

This IOT garbage monitoring system uses LCD to display the status of the weight and level of the garbage collected in the trash bin. The website also will display the weight of the garbage in the trash bin and pictures that demonstrate how full the trash bin is with different levels like shown in the figure below. This monitoring system uses AVR family microcontroller, wifi modem to send data, and LCD screen to display data. This IOT monitoring system can also create alerts by using the buzzer to indicate that the limit of the trash bin has reached. The software of this IOT garbage monitoring system is done through the IOT Gecko web development platform using the language of C and Arduino compiler.

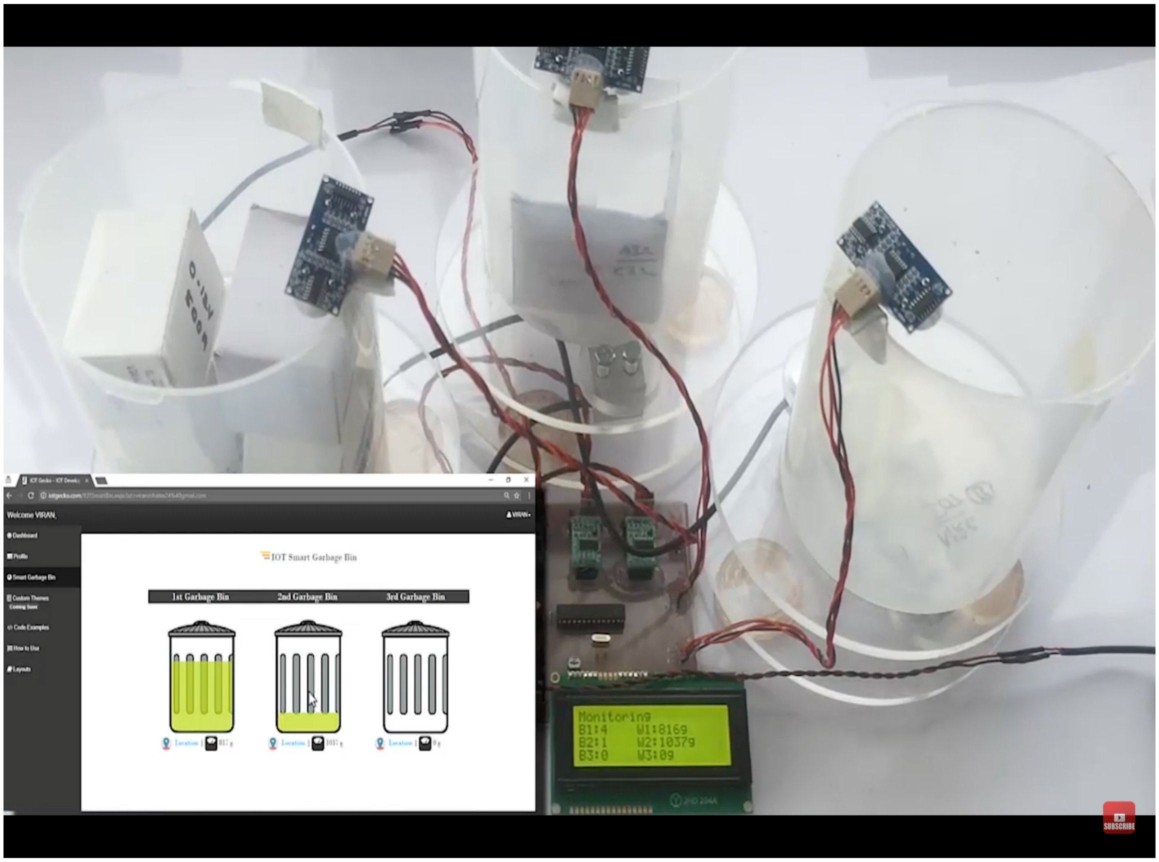


Figure 25: IOT Garbage Monitoring System

The problem with this system if we do resort to using it would be the software features of it. We would need to have everything displayed on our application rather than going to a third-party website while using our software. However, this would be the best choice of a back-up plan just in case our plan of implementing our own trash bin monitoring system does not work out. Thus, it would be the problem when we reach that point.

The hardware components of this system include ATmega328P AVR MC, ESP8266 Wifi module, HC-SR04 ultrasonic sensors, weight sensors, LCDs, LEDs, transformer or adapter, push buttons, and buzzers. The hardware block diagram for the IOT garbage monitoring with weight sensing is shown in Figure 26 below showing how the components are connected together with cables or connectors. The rectifier, regulator, and regulator blocks would mainly include the resistors, capacitors, transistors, and diodes.

In the figure below, it shows the overall block diagram for this IOT garbage monitoring system. In our case, we won't have the same diagram as that. The number of ultrasonic sensors would not be four but rather one of it is enough to detect the level of the trash in the big trash can.

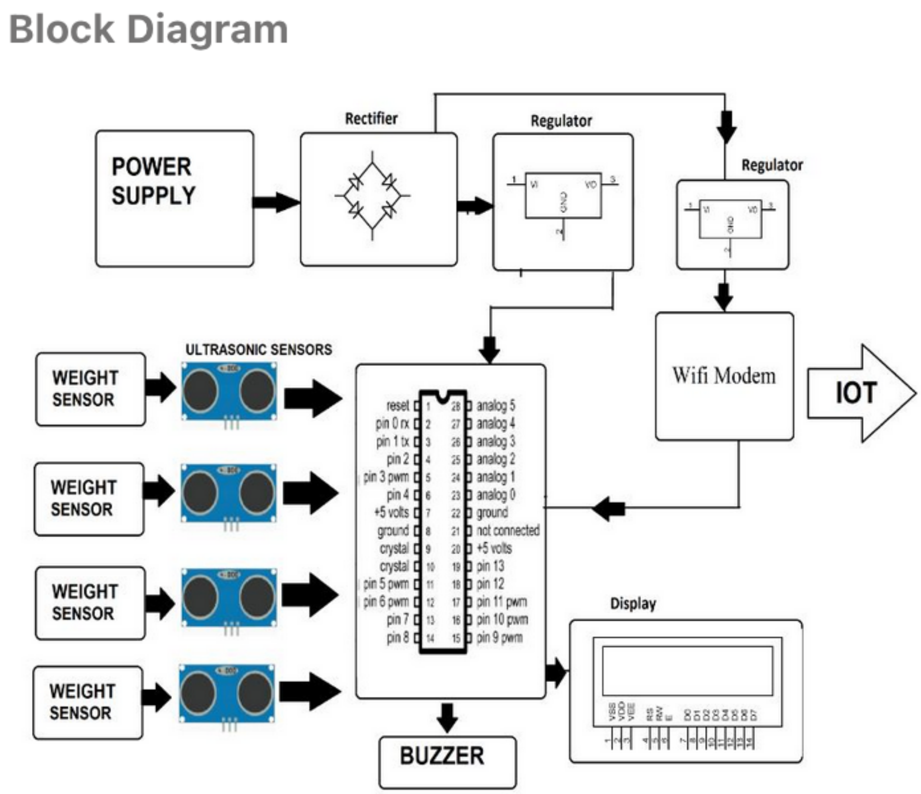


Figure 26: The IOT Garbage Monitoring with Weight Sensing

3.8 Camera

In this section, we will be exploring the different options of cameras for the robot to detect objects. The robot needs to detect objects for multiple reasons, mainly to identify that the object it is looking at is the correct object to pick up from. This needs to be very precise and accurate as it does not need to pick up random items in front of its view. There are many options in the market for cameras as they vary in price and functionality, we would want something that has object recognition programmed inside already which makes the building process a lot easier as we would not have to develop the code from scratch. Another thing that the camera can do for us is object avoidance, this would be considered a stretch goal as it is slightly past the basic goals, but it would need to avoid any object that is not our intended target, we would not want our robot to hit random objects that just happen to be between its path, possibly damaging the bot.

3.8.1 ESP32 Camera

The ESP32 Camera module is a cheap camera that costs only $10 dollars. This camera does not have as many pins as other cameras and it doesn’t have a USB port that we could use so we would have to use an external adapter to work with it. We will also need another adapter for this device's micro SD card. To program this, we will have to use the Arduino IDE but modded so that it can include all the ESP32 boards. This module includes a white LED for a flash and is capable of having an external antenna. This device also has fewer pinouts than other cameras. A lot of people report a lot of problems with getting this device to work. We will also have to use machine learning software to get it to function for our specific means which is identifying the object and using it for our autonomous process.

3.8.2 Raspberry Pi Camera

Unlike the other USB cameras, these types of cameras are connected directly to the GPU of a raspberry pi board thus it does not need to use the CPU to process the camera's data and is much more efficient. This camera can also be very high quality if needed. It has 15 pins that allow the data to stream to the GPU. There are also a bunch of lenses that are compatible with this module. Because of this, we can switch and test many lenses for our personal needs. The connectors use the CSI-2 protocol which is also used in mobile devices and tablets. Most Raspberry Pi cameras use smaller adapters (vs USB ports). Because of this we would need an adapter cable that is required for Pi Zero if we choose to use it. Although these cameras are very cheap and deliver high-quality images and easily send data over from the camera to the GPU, our team is more comfortable and familiar with Arduinos. Also to implement machine learning we will also have to use third-party software to get it to function for our specific means which will be more challenging if we decided to use a Raspberry Pi.

3.8.3 OpenMV Camera

This camera actually came from a Kickstarter with the intention of making machine vision easy and fun to use. This module includes a mediocre-quality camera (640x480 pixels). According to the OpenMV designers, the module is capable of Color Tracking, Face Detection, Eye Tracking, QR code Detection, AprilTag Tracking, and Line detection. This additional functionality already puts this camera above the others mentioned earlier. This board can be programmed with an IDE developed specifically for this board. Since this board runs the MicroPython operating system, we can program the board using Python, a language that some people on our team are familiar with. We can train images with this camera but it is harder to train it to recognize real objects. The accuracy of this camera is not great at all. Also, this camera is more expensive than the others. However, with extra work, we can get this to be able to function the way we need it to for this device.

3.8.4 HuskyLens Sensor

The HuskyLens is a sensor that implements the Kendryte K210 processor which can be used for more advanced algorithms such as neural networks and can even be used for audio recognition. There are two models which vary in camera quality. This device can use many devices but can also be used independently. This module has a camera, LEDs, a selector wheel to navigate through the menu, a training button to have the Husky Lens learn an object, and a microSD card slot. The module is powered by a micro USB. This module also contains a screen used for object tracking.

3.8.5 Pixy Cam 2

The Pixy2 Camera is a great camera. Unlike a regular camera, the Pixy2 is an intelligent camera. This device has an onboard processor that allows it to do things such as object recognition, line-following, intersection detection, and even simple barcode reading. The Pixy2 has all kinds of outputs that can be used with all sorts of microcontrollers. In our case, we will be using this device with an Arduino. The Pixy2 can learn to detect objects that we will teach it (our smaller bin). The camera can also track every object in its field of view. To control the Pixy2 Camera we will use the PixyMon application which uses the USB port. With this we will see the video, configure, and debug the camera. This device supports C/C++ and Python and everything is open source. We choose to use the Pixy2 camera as it out weighed the others with the requirements needed for this project. The Pixy2 Cam has many functionalities that will make the autonomous development of our project much easier. The Pixy2 Camera also offers all this functionality at an affordable price which also satisfies our economic constraint for the project. The camera has built in wifi and bluetooth, ability to seamlessly connect to a computer and the Arduino which we will use for development. The built in object learning and detection using colors is perfect for our autonomous process where the device will seek out the smaller trash can that will be picked up and dumped into the body of the device itself.

Specs/Board	ESP32	Raspberry Pi Camera	OpenMV	HuskyLes Sensor	Pixy2 Cam
WiFi	Yes	No	No	No	Yes
Bluetooth	Yes	No	No	No	Yes
Camera	No	1920x1080	640x480	No	1296x976
Built-In Machine Vision?	No	No	Yes	Yes	Yes
Cost	$17.99	$18.99	$69.95	$66.90	$64.50

Table 14: Parts Consideration Camera Sensor

3.9 DC to DC Converter

Majority of electric circuits are designed to operate at a specified voltage range. A voltage regulator provides a constant DC output voltage regardless of fluctuating input voltages and various output loads. A voltage regulator automatically regulates its output voltage by utilizing a feedback technique. There are two types of voltage regulators, linear and switching regulators.

3.9.1 Linear Voltage Regulator

Figure 27 below demonstrates the fundamental topology of a linear voltage regulator. Essentially, RLDO provides an impedance that is presented to Rload (intended device), and the excess voltage is dropped across Rload. A linear voltage regulator is very similar to a voltage divider with a couple more added features, which will be explained a bit later.

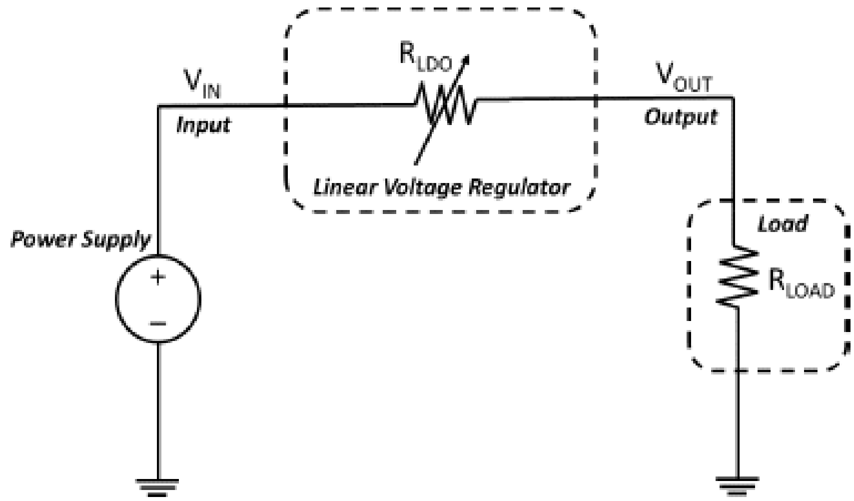


Figure 27: General Schematic Architecture of Linear Voltage Regulator. Image Courtesy of Vidatronics

Referring to *figure 27* above, if we assume the power supply is a constant 12 volts, and our Rload is about 3 kOhms, and we want a Vout of 5 volts, we can use Ohm's Law and Kirchhoff's circuit laws, to determine what RLDO we should use:

1) $Vout = Vin \frac{Rload}{Rload + RLDO}$

2) $RLDO = Rload\,(\frac{Vin}{Vout} - 1) = 3k\Omega * (\frac{12v}{5v} - 1) = 4.2k\Omega$

However, *figure 27* isn't a practical implementation of a voltage regulator to power most electronic circuits. During operation, the output current demand might change which can change Rload, also the input voltage may also fluctuate for various reasons, such a battery losing energy, due to these factors if we use the above implementation of a voltage regulator, the output voltage may be too little or too much for our device. Well designed voltage regulators use a feedback loop and a voltage reference to ensure the output voltage stays independent of the output load and input voltage. *Figure 28* below demonstrates the high level architecture of a proper voltage regulator. Usually the feedback control of a voltage regulator utilizes an operational amplifier to correct the voltages within the regulator.

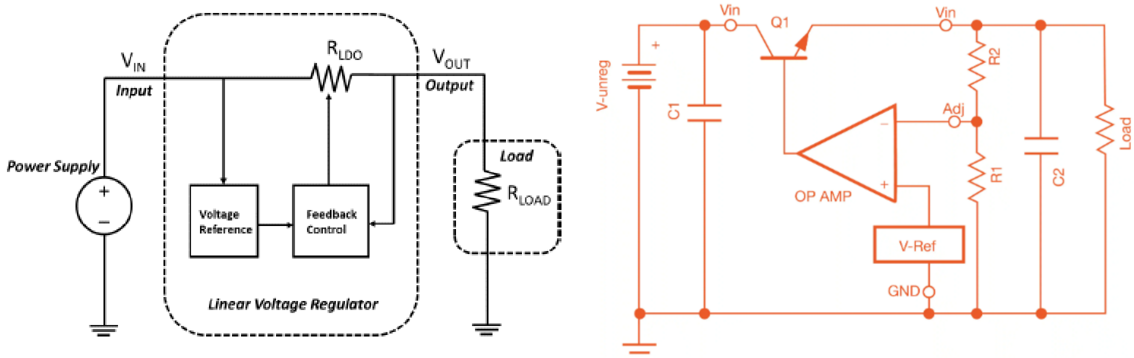


Figure 28: General Architecture of Linear Feedback Voltage Regulator (Left) and Detailed Schematic of Linear Feedback Voltage Regulator (Right). Image Courtesy of Vidatronics and Aspencore

Linear voltage regulators are fairly cheap because they don't use complicated parts, however, there are a few drawbacks of a linear voltage regulator. As stated earlier a linear voltage regulator uses the voltage divider as its foundational concept, thus the excess voltage across RLDO is transferred into wasted heat, which creates a highly inefficient process (<= 50% efficiency), and more energy is lost when input voltage rises. Another characteristic about the linear voltage regulator is that it can only supply a current that is less than or equal to the input current, therefore if the input power supply cannot provide sufficient current then the linear voltage regulator would waste resources. To combat these problems we decided to take a look at switching voltage regulators, more specifically the buck converter.

3.9.2 Buck Converter

A buck converter is a DC to DC converter that steps up the input current, while stepping down the input voltage. This type of converter is perfect for our project because we need to step down 12 volts to 5 volts, while having enough current to supply our MCU, PixyCam, and other components which is around 2 Amps. A buck converter utilized a technique called switching to achieve its output.
Figure 29 below demonstrates the basic topology of a buck converter. A buck converter usually consists of several parts; the input power supply, a switch, a diode, a inductor, a capacitor, and the resistive load, which is the resistor in *figure 29* above. A buck converter can be described in two states, the open state, when the switch is open, and the closed state is when the switch is closed.

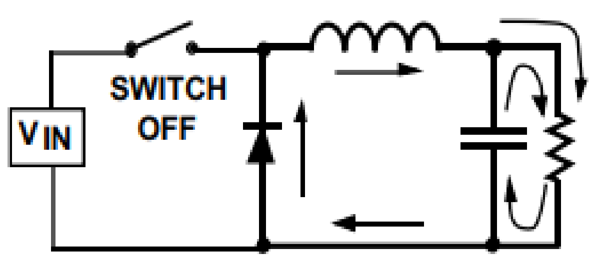


Figure 29: General Topology of a Buck Converter. Image Courtesy of Texas Instruments

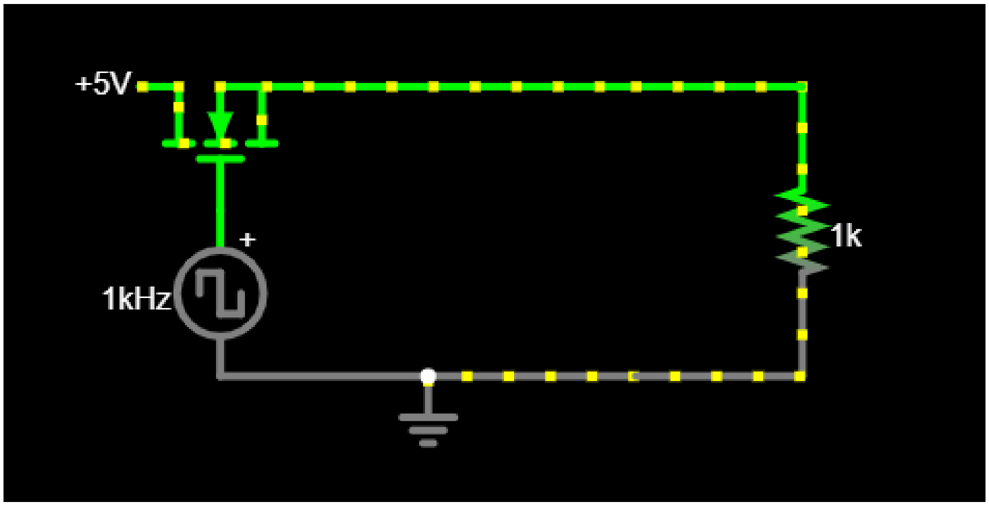


Figure 30: Schematic with a MOSFET, function generator and 1KOhm load

By removing the inductor, diode, and capacitor from the buck converter, and swapping the manual switch with a fast switching transistor as shown in *figure 30*, we can begin to understand how a buck converter works. If we apply a 5 volt input and a 50 % duty cycle we can observe that output voltage across the resistor has decreased to an average of about 2.5 volts as shown in *figure 31* below. By decreasing the duty cycle, the output voltage should also decrease.

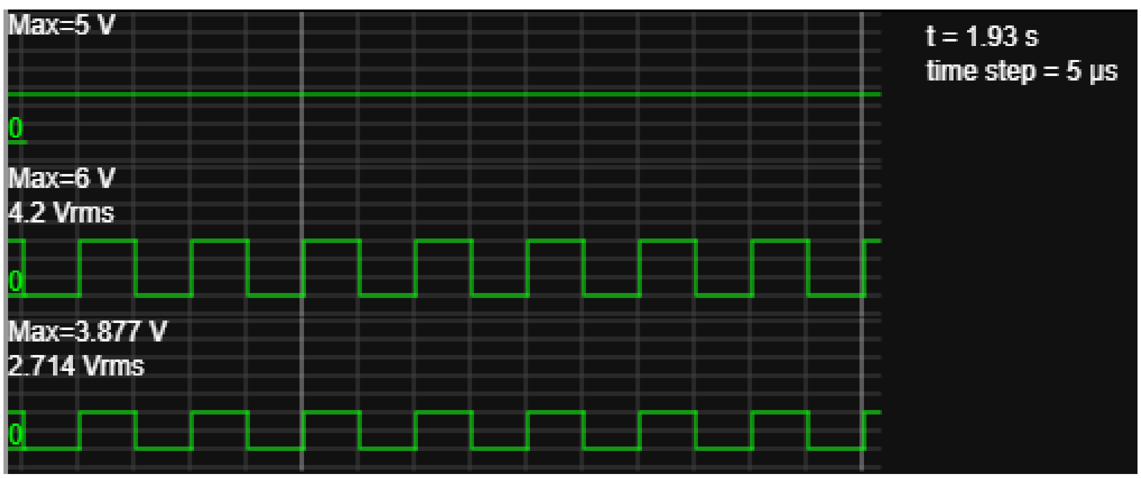


Figure 31 : Voltage plots of figure 30, Top: 5V input, Middle: Function generator with 50% duty cycle at 1kHz, Bottom: Reduced output voltage

However, the output voltage is around 3.8 volts, which is higher than the intended 2.5 volts, and this can damage lower voltage rated components. To get around this, an inductor is placed in series with the resistive load, as shown in *figure 32*.

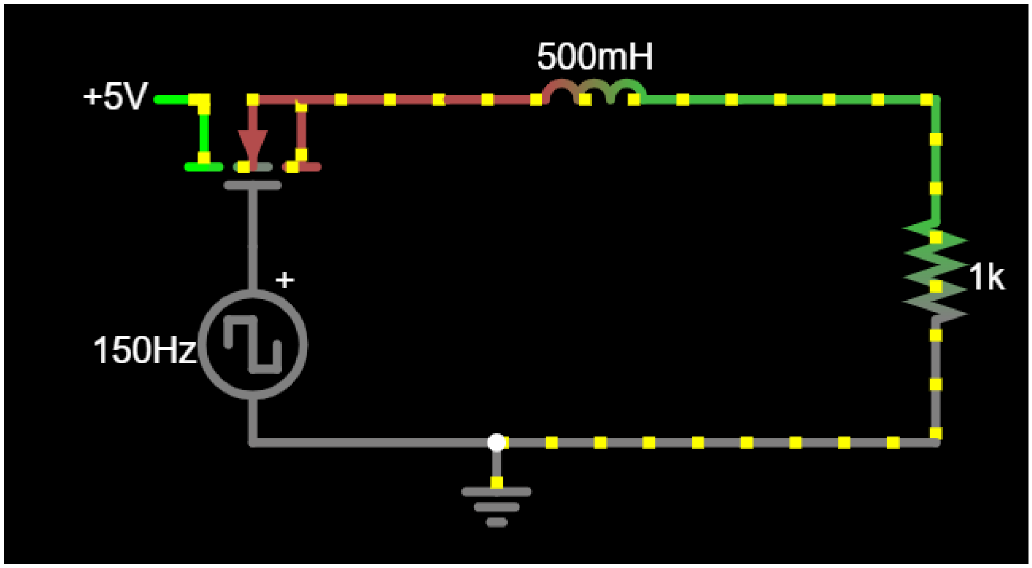


Figure 32: Schematic of figure 30 with an Inductor added

With the inductor in place, the voltage spikes on the output are removed, and this is because an inductor wants to keep the current across it the same. When the switch is off, no current is flowing through the inductor, and when the switch is on, the inductor tries its best to resist the current change but eventually the inductor will have to give and slowly increase its current. While the current increases across the inductor, so does the magnetic field generated by the inductor, eventually when the magnetic field stabilizes the inductor acts like a wire and allows all the current to flow. When the switch is immediately turned off, the inductor wants to resist the current change, but it eventually gives in again, and slowly decreases its current by slowly decreasing the generated magnetic field. By reducing the frequency of the function generator, this phenomena can be observed clearer in *figure 33* below.

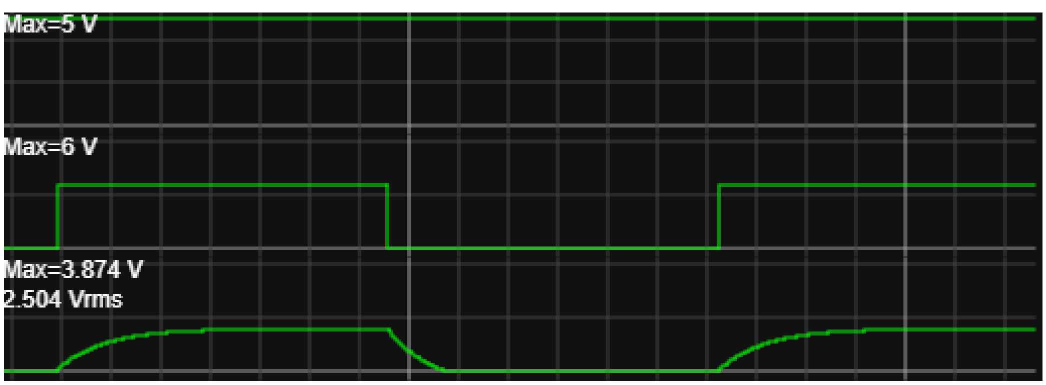


Figure 33 : Voltage plots of figure 32, Top: 5V input, Middle: Function generator with 50% duty cycle at 150Hz, Bottom: Reduced output voltage

By increasing the frequency of our switching, we can decrease how long it takes the inductor to charge and discharge, which creates a more stable output voltage as shown in *figure 34* below. This is the reason that the buck converter utilizes high frequency.

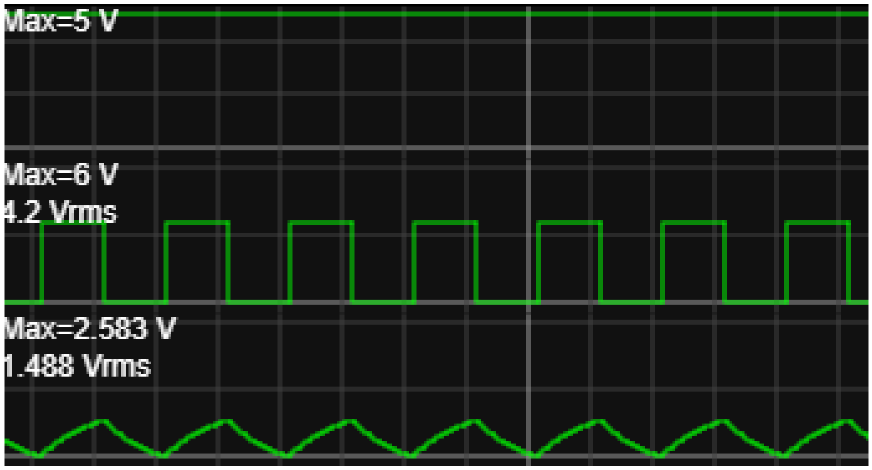


Figure 34 : Voltage plots of figure 32, Top: 5V input, Middle: Function generator with 50% duty cycle at 1kHz, Bottom: Reduced output voltage

There still is a small problem with this circuit. When the switch is turned off, the inductor's magnetic field starts to collapse which causes excess electrons to accumulate at the beginning of the inductor, and this can be observed in *figure 32* above where the line segment is red. This abundance of electrons causes a high negative voltage spike as shown in *figure 35* below, which can be catastrophic to other components.

To fix this, a low forward voltage diode is added to the circuit so that the excess electrons can flow out of the circuit

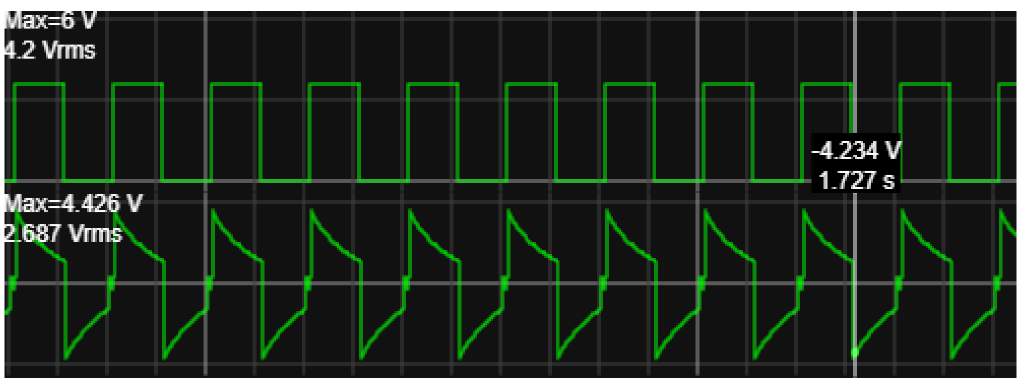


Figure 35: Voltage plots of figure 32, Top: 5V input, Bottom: Voltage across the inductor

Finally, an electrolytic capacitor is added in parallel to the resistive load to smooth out the output voltage ripple in *figure 34* above. *Figure 36* below shows the final schematic of the buck converter as well as the input and output voltages of 5 volts and 1.81 volts respectively.

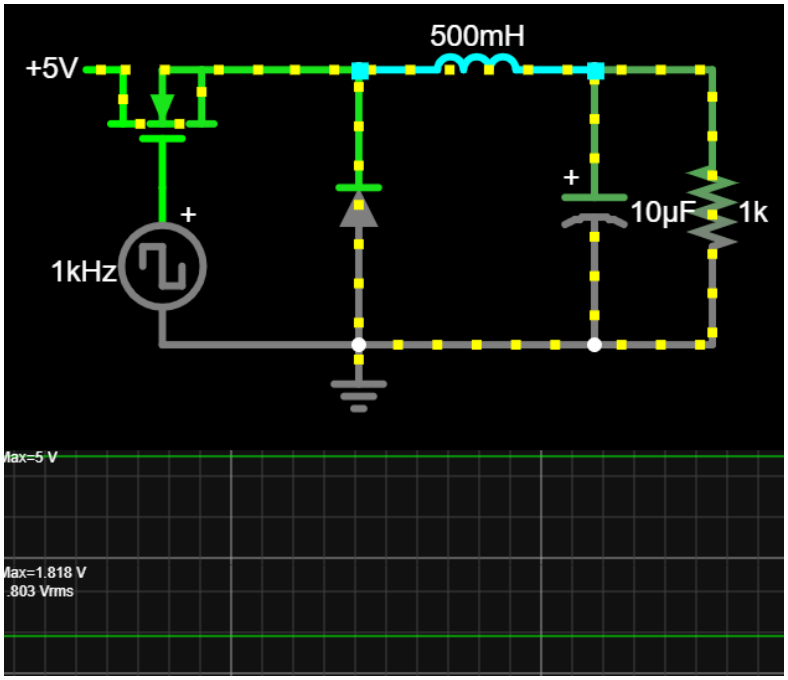


Figure 36: Final schematic of Buck Converter with input and output voltages

There still is one flaw with this circuit; the output voltage is not independent of the load or the input voltage. This problem was also shared with the linear voltage regulator that we explained in the earlier sections. To solve this problem, we have to add a feedback loop that is able to adjust the duty cycle when the load changes. We can do this by using the Texas Instruments LM2673 adjustable voltage regulator. This regulator would act as our function generator, as well as our feedback loop for our buck converter. The LM2673 also comes with a few bells and whistles that enhance our buck converter, such as high input voltages and soft start.

4. Related Standards and Realistic Design Constraints

In engineering, there must be a consistency for how things are done, this goes for measurements, units, organization, testing, execution, and much more. The main standard that engineers follow in America is ANSI or IEEE, these are highly sought out in a company and brings credibility to those who follow them. They provide a detailed description on standards for unit testing and is what is expected for most high-level development. In this field, each step has to be quite detailed to ensure that the project is a success, there has to be clear communication between the client, provider, and the employees so everyone is on the same page on what to expect and the current progress of the project. The client does not need to know the details of how the project is executed in the construction of it, but it is crucial that the company understands exactly what the client is expecting as a product, otherwise many people would be taken off guard on what was expected of it.

4.1 Related Standards

For this project we will be following the standards of the IEEE, the Institute of Electrical and Electronics Engineers. The IEEE has over 400,000 members and is the world's largest publisher of technical documents. There are also members in over 100 countries. The IEEE standards let consumers enjoy many things in their life like wifi. The IEEE has standards that cover a wide variety of engineering. Standards allow for people to compete, create, and invent, and with insurance, the underlying system will prevent anything that will cause whatever is being developed inoperable.

The IEEE standards of development are based on five principles. The right of appeal, due process, openness, consensus, and balance. These could be seen by taking a look at the IEEE standards development process. The process to make a standard has to go through a rigorous path and has to get approved by many people. This is crucial in developing standards that will benefit companies and consumers.

When developing this project and even in our professional life we will be working with standards almost daily as most aspects of a business are impacted by standards, if not, the quality will dramatically increase which would not benefit the consumer and can ruin companies that develop products for them. This is another reason why having engineering standards is important for development.

Our project will be using the said standards from the IEEE website. Our device, the trash collector will be designed and built with these standards in mind as if a consumer will be using the device on a daily basis. This is important as we want the device to be as functional as possible even if something goes wrong. If something goes wrong we will be able to pinpoint the issue quickly as the standards we will implement will be known and established from the beginning. To find standards for this device we will use the IEEE website to search for the correct standards and testing for development. There are many other standards from other organizations however we chose the IEEE standards to use as they have a proven track record of providing consumers and companies with great standards and the standards that made it went through a long rigorous process that is used in the technology and devices we use in our everyday lives. Also, the IEEE is an international organization in over 100 countries and has many members which also gives credibility to their standards as well.

4.1.1 IEEE 1013

The IEEE battery standard goes into how the batteries should be used and tested and any sort of engineering project you would use among other details surrounding batteries. The standard also goes into how to safely implement batteries in your engineering project and product. The standard is used in many devices that are on the market today and companies use it for personal use as well. The IEEE standards are used all around the world and this particular standard is no different as it is referred to in a variety of things. The IEEE standard explains that we should properly set up the battery system in our project to make sure that it is safe for anyone who will use it to use and operate. This standard also talks about different kinds of batteries and talks about how to select the correct batteries relating to different sizes and efficiency overall in order to make sure that the battery is up to standard.

The battery in our device will be selected from a website that generates designs for different kinds of batteries for a PCB board based on your needs such as cost, efficiency, power, and budget. The battery will be connected directly to our PCB board which will be used to power the entirety of our project. The battery will only be on when we turn the device on and off and we turn the device off. The battery will not be running while not in operation which will ensure that no potential overheating will happen which could lead to the device being potentially dangerous if we did not set up the hardware part of our project properly.

4.1.2 Battery Standard implementation

The device will not be in operation for long periods of time so the battery will not be pulling as much power as other devices on the market that follow the IEEE standard. The device on average would be running for a minute or two. The device would be running for 10 minutes solely for demo purposes. This makes sure that the battery uses as much power and limits the time that we have to potentially charge up the battery or even replace the battery with a new one. The point of this device is to simply find a smaller trash can to pick up and then empty the contents into the bin which will be attached to the main part of the device. Because of this, the maximum amount of energy that the batteries use will be kept to a minimum. The only factors we will have to consider are when we start the device where the current starts flowing through the system and while the current is running during operation. Overall our device will not use that much energy overall for this device. We have a maximum voltage volume of 12 volts and we would like to stay under that when the device is fully complete if possible.

For the voltage load, we have calculated all the necessary values calculated by using the Kirchhoff circuit laws to determine all the factors such as our load resistance and more.

We considered using a linear voltage regulator but decided not to use it as when we did the calculations we determined that it wasn't suitable for our case. In fact, we found out that even though the voltage regulator is cheap, we would not get enough power to operate most electronic devices that can be found out now. Also, the input voltage may fluctuate as well which is not good when using a linear voltage regulator.

Our device battery system will completely power all our moving components and all our hardware components such as the PCB, the microcontroller, and the Pixy2 camera together. The PCB will connect all our components together and have the batteries we chose from the website to determine the best efficiency and price for a connected project as well. The PCB will have a battery system connected to it. The PCB will connect the circuit to all our components on our device such as a sensor which will be attached directly to it. The sensor will be used to detect walls and which we will use and incorporate into our autonomous algorithm to help avoid going into the walls as well.

The device will also use the batteries to power the motors which will control each individual wheel to act as a four-wheel drivetrain such as the one you can find utilized in many vehicles. We will also have motors that will be used for our mechanical lift that will be lifting the smaller trashcan full of contents as well. There will be two motors that will be under stress so if you are lifting up the smaller trash can, this will probably take the most energy to power as the trashcan could possibly be heavy (we will determine the max weight in our testing).

The device will also power the servos which will have many functions in our device that will be attached to the motors and the moving components on the mechanical lift. These servos will have multiple uses. One will be used to grip onto the trash can. The other two will help with flipping the trashcan to be able to dump the contents contained inside of the smaller bin into the bigger bin that will be attached to the main body of the device.

The last two things are the batteries will power the microcontroller which will act as the "brain" or the main unit of the device. The microcontroller will be able to control all our motors, and all the servos, and will also have access to the Pixy2 camera. The Pixy2 camera will be able to send information to the microcontroller that it will be able to use to be able to control the motors and servos when it comes to our autonomous algorithm. The Pixy2 camera will be powered by the battery source as well and will be used to be the main driver of the autonomous algorithm using robot vision and working closely with the microcontroller.

As you can see we have many components that have to be powered by the batteries that we will use. Because of this, we have to make sure that we choose the correct size battery and the correct battery set up to be able to get the best efficiency and output while staying within our budget. The IEEE standards help us do this as they have guidelines for picking the battery size and design. Then we compared it to what we found on the line. The IEEE standard also goes over any safety precautions we have to take while working with the battery and developing a circuit that will let the current reach all the components efficiently. We will follow all of the safety guidelines listed in the standard closely as this will also help with our safety constraints as well. We want to make sure that the device is as safe as possible so anyone can operate it without possibly getting anyone hurt because of incorrect battery implementation.

For this device, we will be using a 12-volt battery pack that contains a battery management circuit. This is called a BMS. A battery management circuit lets us have features for the batteries such as overcharge protection, over-discharge protection, cell balancing, and short circuit protection. Without having these features in place this can make the battery a very dangerous power source that can possibly even damage our device. The IEEE standard states that whenever working with a power source such as batteries you would need to have a battery management circuit to make sure that these issues do not occur. This battery pack is perfect for the needs of this project and meets our needed voltage for the way our PCB is set up for our device. We believe that our battery setup and implementation completely comply with the standard from the Institute of Electrical and Electronics Engineers.

4.1.3 IEEE 829-2008

The IEEE has developed the 829 Standard for Software Test Documentation for any type of software testing. The industry has come up with this standard that has been widely accepted in the engineering community for software testing. There are eight existing document types in the IEEE 829 standard, which can be used in three distinct phases of software testing. The three groups that the documents can fit into are the preparation of tests, the running of said tests, and the completion of testing.

When preparing the tests of the software, the IEEE 829 standard has broken it down in simple steps: Test Plan, Test Design Specification, Test case Specification, Test Procedure, and Test Item Transmittal Report. Following the guideline given by this standard will ensure we will create great, viable tests to determine if our device will function as expected under a multitude of variables and outside environmental factors.

In our device software wise, we will need to test the movement of the device to make sure our motors are working correctly and that we are able to control the device manually. To do this, this will also involve testing of the mobile app as well to make sure a connection is established to the device and that it is able to receive notifications when the status of the device changes such as if the power is on or off or if the trash bin attached to the device is full. We will also need to test the anti wall collision of the device so it does not run into a wall or so that it knows to redirect itself if it does. A great amount of our software testing will be involving the autonomous feature of our robot which will include the correct movements being made to seek the smaller trash can to be picked up and disposed of. This will definitely include trial runs of object detection by just having the device be able to recognize an object by camera, then having the object be able to rotate itself to see the object, then eventually having the object be able to move to the object will keep the object in the center of its point of vision. Then last we will test the device actually being able to pick up and dispose of the smaller trash can by itself. We will also need to do a multitude of testing with the Arduino and Pixy2 camera.

The Pixy2 camera must be tested to make sure that it recognizes the correct object by training it in the Pixy2 Camera software IDE. Once that is done we will have to do testing to make sure that the Pixy2 Camera is sending the correct coordinates based on the X, Y, and Z axis to make sure that the Arduino knows how to control the motors to be able to properly run the autonomous algorithm with conjunction of the motors to find and seek out the object to pick up and dispose of which is the smaller trash can that will be brightly colored so it is easier for the Pixy2 Camera. Based on our test design specification we will go into depth and describe how each test will run after creating test cases for each aspect of what needs to be tested for our device.

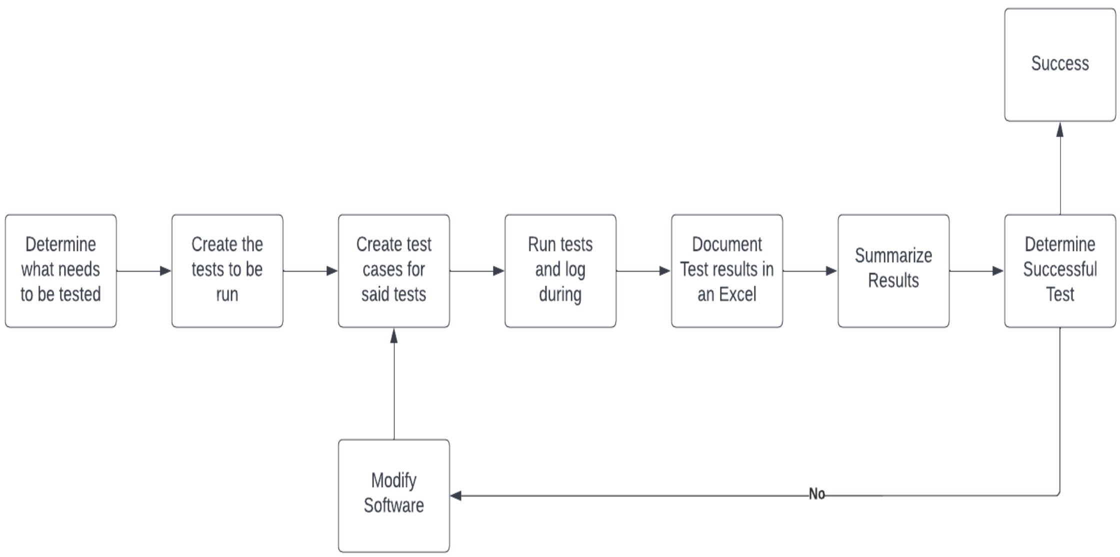


Figure 37: The Testing Process for Software Development

After each test case is done we will document our results and if necessary will create another test case while possibly redesigning and adjusting our software until we have reached success for our main goal which is established in our specifications. We will do anything we observe during testing with details in time order while running said test. We will also use the test incident report defined in this standard to record details of events that will need to be investigated before we give the green light and decide that the device has successfully passed and completed a test case according to the standard. After all testing is completed on one aspect of our software, we will summarize and evaluate the tests. During the running of test cases we will track any significant events in an excel and write a paragraph that will summarize the results of the testing. The figure below is a simplified flowchart of the process we will use to test the software of this device.

4.1.4 IEEE Std 1666-2011

The IEEE std 1666-2011 is the IEEE Standard for Standard SystemC® Language Reference Manual. This standard is for a library called SystemC which is a C++ class library that connects both software and hardware together with helpful built-in functions. This standard also includes valuable information when it comes to programming our Arduino by having many libraries that work with the hardware side of our device and that will communicate with our PCB. This will make a complex system of connecting software and hardware much easier. This standard is also a guide which we can refer back to during our process of developing the software for the Arduino and Pixy2 Camera that will lead to the code following the required standards documented in this manual. We will also use this manual to refer to any functions we will need during the development process that we might not know to do a task during our code and even has potentially to even help us fix and debug and issues that we might find along the way in our code. This document goes into depth of these built in functions as well as standards for developing software in the language of C++. It is good to have consistent coding standards and even a style guide in order to be able to read code clearly and easily. This is important when it comes to testing the software, fixing any errors and bugs that might come up while developing and refactoring so the code has great readability just in case someone else needs to come in and fix a problem or to use previous code for another aspect of the project.

Regarding coding style, we will be following the coding style guide provided by google to avoid turning our code into something that is full of bugs and hard for other developers to read. The whole point of us following a style guide is to make sure our code is readable to anyone who might come across it for reasons as fixing something that went wrong that they haven't worked on, refactoring existing code to do the same task another way or even in a shorter amount of time, for someone to try to replicate our software, or even for manufacturing purposes. The google style guide is similar to the IEEE Standard for Standard SystemC® Language except that it solely focuses on style and syntax. Some examples that this document has is how you would initialize variables, how to layout and format functions, and also how to name variables. Overall, our code for this software should prioritize how to write it with a reader in mind and being consistent with whatever existing code is available. We will also prioritize optimization and avoid any potential dangerous or lines of codes that might lead to problems or bugs later on in development. These issues will be taken care of early on to avoid any potential issues and conflicts later down the line in development.

4.2 Realistic Design Constraints

In this section, we will be looking at the different types of constraints for the project. We should understand the things that could hinder us from getting the project done within the timeframe. Without the organization of knowing the difficulties we face; it could make us unprepared for the problem and set us back much further than we would be expecting for this outcome. Though, it might be difficult to list out all the possible constraints on the project, the clear big issues should be thought of ahead of time and should try to figure out a solution or adjust accordingly. Making these preparations will save us from running into these problems in the future, saving us time and possibly money that could be invested into something that would not work out.

4.2.1 Project Design Constraints

For this project it is essential for us to set realistic design constraints in order to complete this project efficiently and in a timely manner. Design constraints also give us a guideline to use while developing our project which makes the process of developing it much easier for our team. In this section, we will go over all of the constraints we came up with and then look at the overall to have a clear picture Of all our constraints combined.

4.2.2 Economic and Time Constraints

At the beginning stages of our project, we came together and had several brainstorming meetings. During this, we came up with multiple project ideas and weighed the pros and cons of each idea. When we settled on the robotic trash can device, we decided to have a budget of $200 per team member. Thus our total budget for the project is $800. We preferably would like to be under that number, but we are willing to spend more if needed. However, this is the worst-case scenario and we shouldn't have to spend any more than our budget. For the time constraint, we have two semesters to complete everything involving senior design. In the first semester see your design one we will write the report and also start and finish testing for the device. During the time between Senior design one and two during August, we will be buying and continuing testing parts for senior design which starts in the fall semester. In the Fall semester we will start development of the device with our set timeline with key deadlines where we should be done with our progress.

To make sure we are able to finish the device and meet all the deadlines we will make sure to prioritize shipping times for all our necessary components to build the Trashporter. This is very important and crucial to us as there is currently still a shortage from the pandemic that is affecting the supply chain. The websites that had the fastest shipping times we have found are Amazon, eBay, and other websites that specialize in selling specific parts and components. Because of this, we will be sticking to mostly these websites to order parts in a timely manner to help us with our goal of meeting our deadlines. We will also make sure that we have check ups throughout the process and weekly meetings to make sure everything is going smoothly and coming together on time.

To make sure we reach our economic constraint which is our budget of $800, we are shopping very methodically for only the parts and components we absolutely need and limit any parts that might go unused and other excess components. We are also sourcing from many different sites to make sure that we are getting the best price period. We also want to make sure we have enough in our budget just in case a component breaks or we need to switch a component while developing. To do this we will make sure that at least one-fourth of our budget is put away for any emergency situations that might come up while developing the device. However, we would like to stay well under our budget of $800. We also want to make sure to keep our materials cost low as well as we will be using a metal frame to build out the robot foundation. We have been looking at many different kinds of robotic frames such as c-channels from the company Tetrix robotics and other robot parts and components websites. We are only looking for one that is affordable. Some websites sell bundles of frames of metal c-channels that we could buy, however, these bundles are hundreds of dollars so we believe that we will use our prototype as a guide. We will only buy a specific amount of parts needed to build out the frame. To do this we will pre-design our frame with minimalism in mind to save the most money and only buy the parts we need and maybe a couple of extras just in case.

4.2.3 Environmental, Health, and Safety Constraints

We have size constraints for this device so that it will be small enough to be able to be picked up and moved around by someone of an average size. Also we want to be able to transport this device easily. Because of this, this device should not cause any potential harm to people around it. However, motors and gears can possibly be exposed on this device, so we will make sure it is covered up or out of reach so no one will be able to access it and potentially get injured such as hurting their fingers. One potential cause of harm could be children as their curiosity might mess with the process of picking up trash can autonomously. So parents should be aware of their child while this device is doing its job of picking up trash. Besides that, we have in mind that safety should be a number one priority when designing and building this device. This device should not harm the environment in any sort of way as we won't have to worry about potential pollution etc. The only constraint we have is the size and making sure components are protected.

A possible safety concern that we could have with this device is the battery. The battery can potentially cause harm to our device and other people if we don't properly design it two to fit our system. Also, we need to make sure the implemented wiring of the battery connected to the PCB is correct. The battery could be a potential fire hazard if the following isn't correct by potentially causing the battery to by causing the battery to potentially overcharge or something else like a fire or explosion. To prevent this we will be using a battery that has a battery management system built into it the pack that helps it prevent the battery from overcharging, discharging, or any other common possible complications you can get from working with batteries. This will also potentially protect our components from possibly shorting out as well which will be beneficial for our project.

Another safety hazard that we should avoid is potentially getting our fingers stuck or cut by the mechanical lift as this is going to have a significant part of a robot be moving up and down and that can cause potential hazards while working on the lift such while possibly fixing any issues such as malfunctions or any part of the mechanical lift that has to be readjusted. So to make sure that we do not get hurt or any other potential people that might work on our project, we must take the necessary safety precautions. We will make sure that everything is off while working on the device and make sure the motors and servos do not run while working. Another precaution we can take is making sure the mechanical lift does not fall on our fingers while working on it because the lift might not be suspended in the air by itself, especially when working with a sliding lift that can come back down at a high speed.

4.2.4 Manufacturability and Sustainability Constraints

Manufacturability should be no problem for the device as it will use many components that it's already manufactured as it is. Therefore this device will have components that I have already gone through industry standards and should be replicated easily if our device were to be manufactured. Our device is also sustainable as parts will be able to be swapped out easily such as the motors and service this device will use. We also plan on having a panel that will give easy access to any person who wants to work with the electrical components of the device. This gives advice sustainability as if anything breaks or malfunctions it will be easy to fix the issue with little to no problems. The two components that will be harder to manufacture are the Pixy2 Camera and our custom made mechanical lift. The Pixy2 camera already is not produced in mass and will have code specifically programmed for our needs that will need to be replicated. The mechanical lift will have to be produced exactly the way we will design down to the measurements for proper operation and use with the Pixy2 camera.

4.2.5 Ethical, Social, and Political Constraints

We do not believe our device does not violate any ethical constraints. This device simply uses motors and services to pick up a trash can autonomously. Some people see automation as a threat as it can cause job instability for many people. Automation does not need breaks or need to be paid for their work which would lead to a new level of efficiency the world hasn't seen since the industrial revolution. Our device can potentially take away a task that a janitor would do which is going around and collecting and emptying trash cans to be dropped off in the dumpster. However, we do need to believe our device will ever reach that level of efficiency that it could be considered as something to use by an actual person. Besides this factor, we do not believe our device violates any ethical, social, or political constraints.

There are some ethical questions that have been growing recently with the use of autonomous machinery and algorithms in devices as well as artificial intelligence. There is an increasingly growing fear among people when discussing the great strides the field of technology will cause. Some people are afraid to potentially lose their jobs and be essentially useless when it comes to manual labor careers. This is something we have thought about and considered as technology as the years past is becoming more advanced quickly.

This is a valid question when discussing potential jobs being lost because of artificial intelligence and autonomous machinery and algorithms. In our case, potentially our device could contribute to it if our robot functions very well. It could possibly remove the job that regular people have of throwing their trash out into a bigger bin in their household which will make their life easier but not a significant amount besides convenience. However, this device can potentially be built upon to potentially replace garbage men who go around the neighborhood collecting trash. However, we believe that our device will not be potentially unethical as the technology that we will be implementing in our project, specifically with the Pixy2 camera, has been around for years. Although the Pixy2 camera's robot vision technology performs very well at detecting objects and being able to use that detection software to autonomously do things, it is not a threat to potentially put people's jobs in danger or affect their way of living.

5. Project Hardware Design Details

Here, we will describe the details on the physical hardware design and what must be added to the device. We will have to research the circuity of the entire and individual components that will be interacting with each other to see if they are compatible. Having a good understanding of each component and how they work will speed up the process in connecting them together. Some components will require a voltage drop or rise, which will need a regulator to control such. The calculations will be researched to understand what the maximum and minimum threshold for the parts are to be careful of not to damage them.

5.1 Initial Design Architectures and Related Diagram

We originally decided to work with the Texas Instrument TPS56424 4A synchronous buck converter IC and created a schematic and PCB design for the TPS56424 shown in *figure 38* and *39* below. However we overlooked the size of the IC, and when it arrived we realized it was too small to practically test and work with. The IC would only be possible to test with the right equipment, which would go over budget. Therefore we decided to stick with through hole components for our buck converter design, and decided to work with the Texas Instrument LM2673 buck converter which is explained in the next section below.

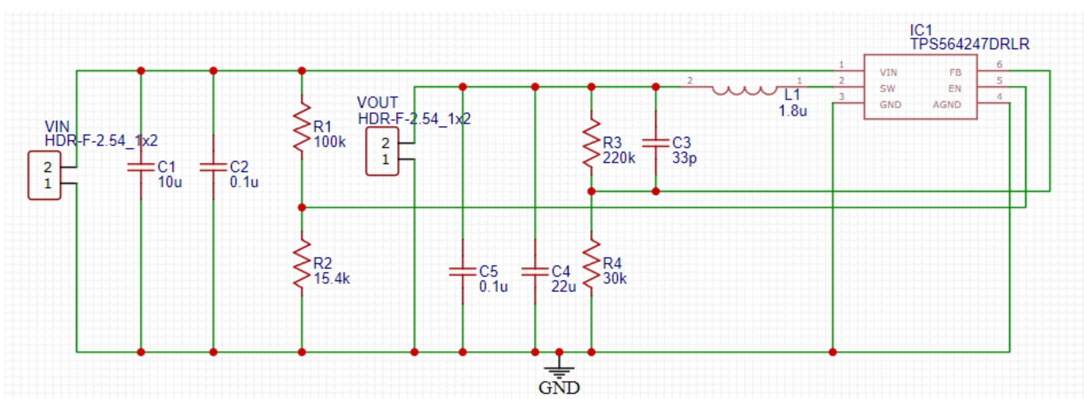


Figure 38: TI TPS56424 4A Synchronous Buck Converter Schematic

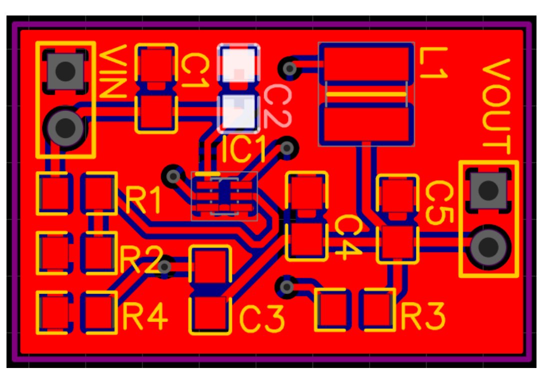


Figure 39 : PCB design for Schematic in Figure 38

5.2 ESP32

With the help of the pinout diagram in *figure 70 in Appendices*, we can connect our various sensors to the analog or digital pins of our ESP32, depending on what signal is being received. The same can also be done when transmitting signals to other components, like the motors for example. In this case the ESP32 will send the proper PWM signal to the HiLetgo L298N stepper motor driver, and the driver would send the corresponding voltage switching to the motor. The ESP32 also comes equipped with a wifi chip, therefore we don't need an external board to process the IoT of our project.

5.2.1 ESP32 + Pixycam2

Figure 71 in Appendices comes from the PixyCam2 documentations, it demonstrates the pinout diagram of the Pixycam. With the help of these pinouts we can easily connect the ESP32 with the Pixycam. The PWM pins of the Pixycam would connect to the digital pins of the ESP32, the I2C, the SPI, and the UART pins would connect to the corresponding pins on the ESP32. The power supplying the PixyCam and ESP32 will be coming from the PCB buck converter that we designed earlier in section 5.1.

5.3 Buck Converter Design

With the help of the Texas Instrument LM2673 documentation we were able to create a 12 volt to 5 volt buck converter. *Figure 40* below displays the general schematic that TI provides to create a buck converter.

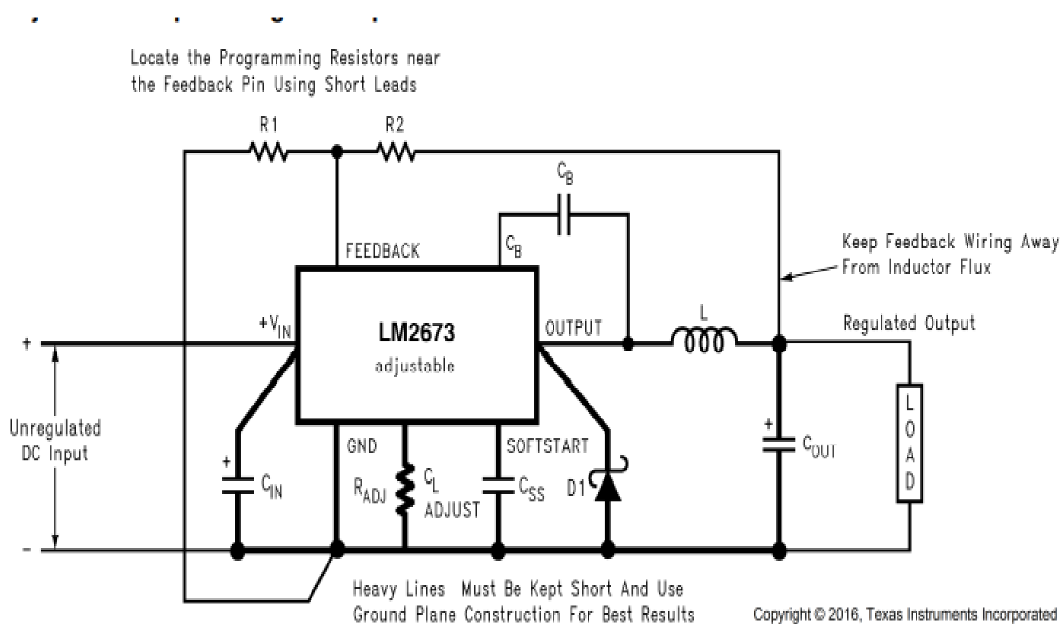


Figure 40 : TI General Schematic of Buck Converter. Image Courtesy of Texas Instruments

Using the LM2673 documentation, we were able to define what values we needed and what values that needed to be calculated.

Defined Values:
$Vout = 5v$
$Vin\,Max = 13v$
$Iload\,Max = 3A$
$R1 = 1k\Omega$
$CB = 0.01uF$

Using the defined parameters above we can calculate the rest of our components values.

Calculated Values:
$R2 = R1(\frac{Vout}{VFB} - 1) = 1k\Omega(\frac{13v}{1.21v} - 1) = 3.1k\Omega,\ where\,VFB = 1.21v$
$E * T = (Vin(max) - Vout - Vsat) * (\frac{Vout + VD}{Vin(max) - Vsat + VD}) * \frac{1000}{260}$
$= (13v - 5v - .45v) * (\frac{5v + 0.5v}{13v - .45v - 0.5v}) * \frac{1000}{260} = 13.25\ (v * us)$

VD is the voltage drop across the diode, which is about 0.5 volts. Vsat is the voltage drop across the internal power switch of the LM2673 which is Rds(ON) time Iload ($0.15\Omega * 3A = 0.45V$). Using our calculated E*T value, we can determine what inductor we should use by using the diagram provided by Texas Instruments shown in *figure 41* below.

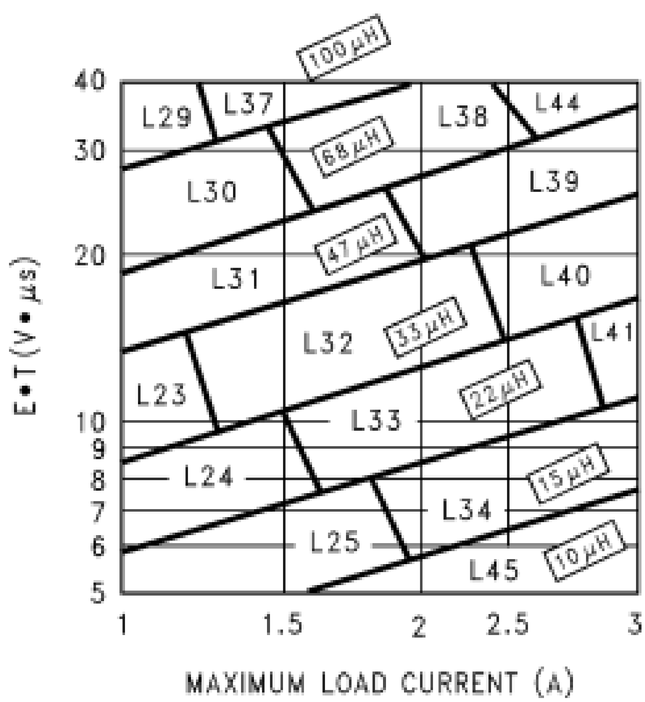


Figure 41 : Texas Instruments inductor nomograph. Image Courtesy of Texas Instruments

The intersection of E*T = 13.25 and load current of 3A gives us the label L41. Identifying L41 in the documentation describes a inductor 22uH rated at a current of 5.22 Amps. Once we have our inductor value, we can use the two tables below in *figure 42* provided by TI to choose our output capacitor. We decided to use a 330uF capacitor.

OUTPUT VOLTAGE (V)	INDUCTANCE (μH)	SANYO OS-CON SA SERIES	
		NO.	C CODE
1.21 to 2.5	33 (3)	2	C3
	47 (3)	2	C2
2.5 to 3.75	33 (3)	1	C3
	47 (3)	1	C2
3.75 to 5	22	1	C3
	33	1	C2
	47	1	C2

CAPACITOR REFERENCE CODE	SANYO OS-CON SA SERIES		
	C (μF)	WV (V)	Irms (A)
C1	47	6.3	1
C2	150	6.3	1.95
C3	330	6.3	2.45
C4	100	10	1.87

Figure 42 : TI Capacitor Chart. Image Courtesy of Texas Instruments

Next, we need to pick our output capacitor. For this capacitor, we need a capacitor that is rated at least 35V and has an RMS of 1.5A (½ Iout Max). Using the same TI capacitor chart, we picked a 680uF capacitor rated at 63V. Next, we need to pick a diode for the excess electrons, so we decided to opt for a 3A low forward voltage diode. Finally, we need to find RADJ for our peak switch current limit.

Calculate RADJ:

$$X = Iload + (Iload * 0.5) = 3A + (3A * 0.5) = 4.5A$$

$$RADJ = \frac{37,125}{4.5} = 8.25k\Omega$$

Component Chart:

Type	Specifications	Manufacturer	Size	Price
LM2673 Voltage Regulator	–Buck Topology –8-40 Volts Input –1.2-37 Volts Output –3A Output –260kHZ Switching Freq.	Texas Instruments –LM2673T-ADJ	Through-Hole, 24mm x 10mm	$5.46
Schottky Diode	–3A Forward Current –40V Vrrm –480mV Vf –120A Ifsm –Ir 500uA	Vishay –SB340-E3/73	Through-Hole, 5.3mm x 9.5mm	$0.47
Toroidal Inductor	–Radial –22uH –DRC 7mOhms –16.4A Max Current	Bourns –2305-V-RC	Through-Hole, Diameter 35.5mm, Wire Length 16.51mm	$4.54
Aluminum Organic Polymer Electrolytic Capacitor	–Radial –330uF –25 VDC –ESR 14mOhms –1.58A Ripple Current	Panasonic –25SEF330M	Through-Hole, Diameter 10mm, Length 13mm	$2.88
Aluminum Electrolytic Capacitor	–Radial –680uF –63 VDC –ESR 50mOhms –2.38A Ripple Current	Nichicon –UHW1J681MHD6TN	Through-Hole, Diameter 16mm, Length 20 mm	$2.65
Resistor	–1/4 Watt –1KOhm –Axial	KOA Speer –MF1/4DCT26A1001F	Through-Hole, 6.3mm	$0.13
Resistor	–1/4 Watt –3.16KOhm –Axial	KOA Speer –MF1/4DCT52R3R16F	Through-Hole, 6.3mm	$0.18
Resistor	–1/4 Watt –8.25KOhm –Axial	KOA Speer –MF1/4DCT52R8251F	Through-Hole, 6.3mm	$0.13

Table 15 : Components chosen for Buck Converter

A couple things to notice is that we picked electrolytic capacitors for the input and output voltage pins because they are better for smoothing out voltages, and they can handle high voltage spikes. Another thing to notice is that we used a toroidal inductor made of iron, which is a better choice for high frequency circuits compared to ferrite inductors.

5.4 Schematic

In this section we will dive into our process into making the schematic for the Trashporter more specifically the buck converter schematic as well as the overall schematic.

5.4.1 Buck Converter Schematic

Now that we have all our component values and general schematic, we can finally create our buck converter schematic, which is shown in *figure 43* below. Pin 1 of the LM2673 is the switching output, pin 2 is the power supply input, pin 3 is the boost feature of the LM2673, pin 4 is ground, pin 5 is the current adjust, pin 6 is the feedback pin and finally pin 7 is the softstart pin which we didn't use for this project.

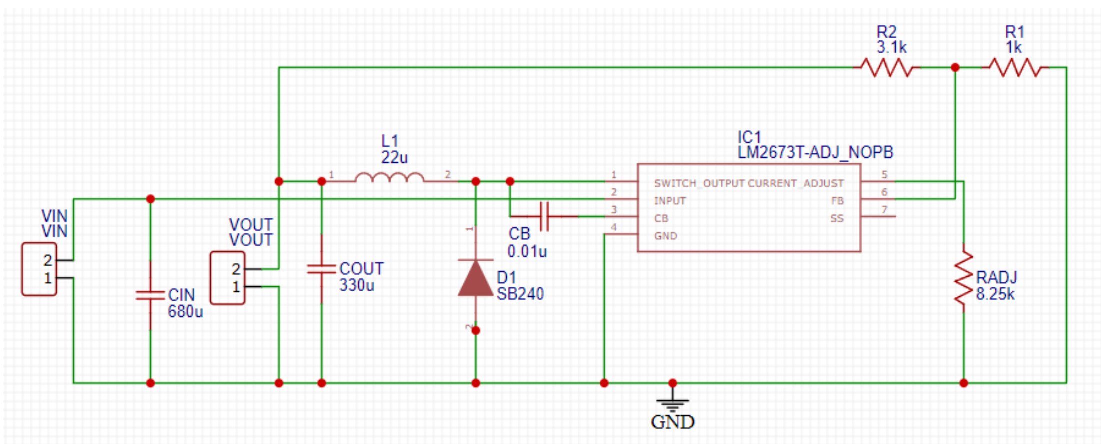


Figure 43 : Schematic of the LM2673 Buck Converter using EasyEDA

5.4.2 Overall Schematic

After creating the buck converter schematic, we then created the overall schematic in *figure 44* below. The following schematic contains the ESP32, the buck converter, and the pinouts for other components in our system; the three stepper motors, the load cell sensor, the sonar sensor and the PixyCam 2. Notice that we used ports to connect the GPIO pins of the ESP32 to the buck converter and the 2.54 mm male pin headers, we did this because it is easier to read the schematic and more convenient to add more components if needed.

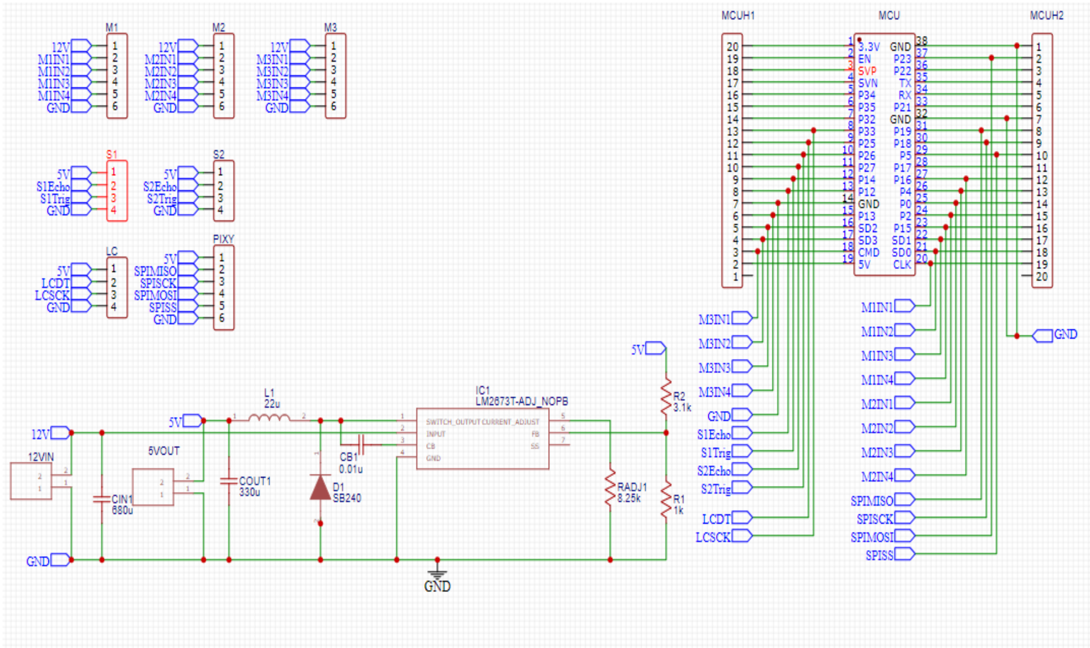


Figure 44: Final overall schematic

5.5 PCB Design

In this section we will dive into our process into making the PCB for the Trashporter more specifically the buck converter PCB as well as the overall PCB.

5.5.1 PCB Buck Converter Design

Using EasyEDA we have created a PCB for our buck converter schematic design in *figure 45*. Notice that we try to keep the inductor L1 as far away from the LM2673 IC as possible, we did this because we don't want the flux of the inductor influencing our circuit traces.

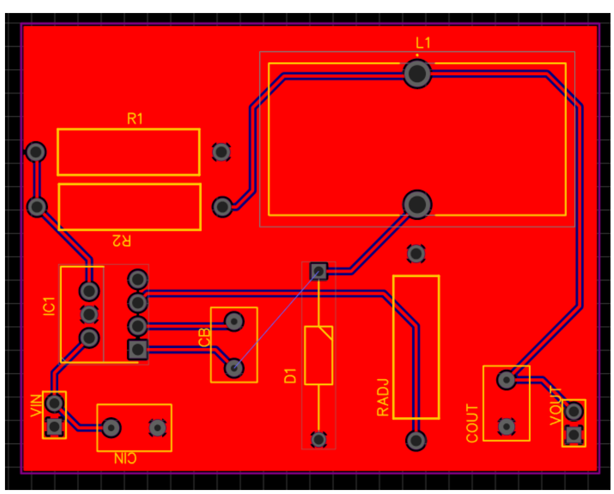


Figure 45 : PCB Design of the Schematic in figure 43

5.5.2 Overall PCB Design

Once we had the overall schematic designed, we finally designed the overall PCB for our project in *figure 46* below using EasyEDA. The PCB will be a two layer design, the red PCB on the left will be the top layer and the blue PCB on the right will be the bottom layer. Notice that the 12 volt traces are thicker than the rest, this helps the trace carry larger currents, which should be around 3 amps. Also, notice that the inductor L1 is placed farther away from the rest of components because the magnetic flux can unintentionally influence the circuit. Furthermore, notice that the buck converter components are placed far from the ESP32 because we don't want the high frequency influencing the ESP32. Finally, the pinouts are placed close to the ESP32 because it helps increase efficiency.

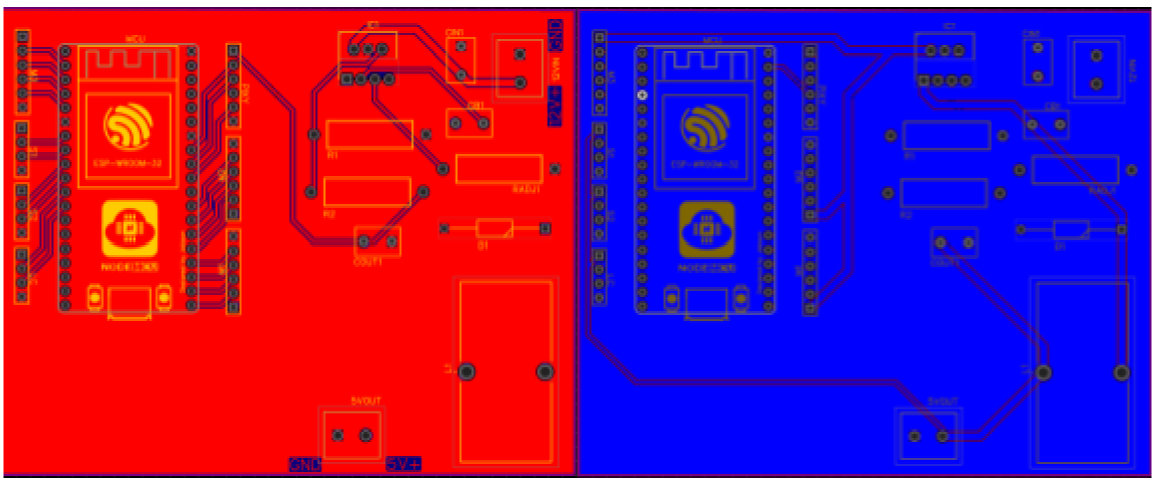


Figure 46 : Final PCB design of the overall schematic.

5.6 Hardware Design Summary

Throughout the previous subsections of section 5 the hardware design of the project has been modified from the original design. Each design component has been researched, tested and discussed throughout the process thus far. The results of our fruitful research and testing has led to the creation of our final hardware design diagram in *figure 47*, below.

In the diagram below in *figure 47*, there are four main blocks which are outlined with dashed lines. The first block is the PCB block which demonstrates the MCU and buck converter that will reside on the PCB. The second block is the lift mount block which represents the components that will reside on the lift mechanism of the Trashporter. Thirdly, the last block represents the entire Trashporter chassis and this is where all the hardware components will be placed except the AC to DC charger. Finally, the last block represents the possible block, which represents the part in the system that may be implemented later if time permits, which is the audio system in this case. The possible block may or may not make it into the final implementation of the Trashporter as it is not an immediate subsystem in our robot.

Overall, as we do more research and testing of the hardware design of the Trashporter, it has changed overtime to meet the requirements and design constraints. The final hardware design may or may not change as we begin to build the Trashporter due to complications, change of plans, more research or other factors.

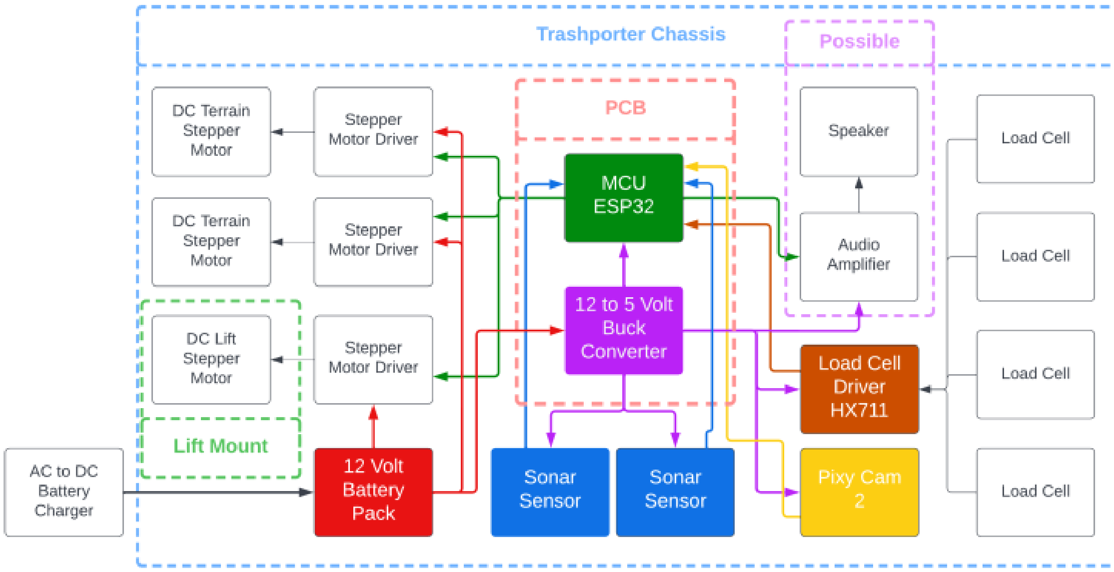


Figure 47 : Final Hardware Design Diagram

6. Project Software Design Details

The software design details are going to describe how the software is going to function within our project. Software is not something that we can physically see and sometimes test, so ensuring we have the details before uploading the logic is essential. Here we will develop the code, and research how we would implement artificial intelligence to automate the device. We will explore the different algorithms so that the bot can perform what is needed. Each component has a different logical system, so we would have to understand how each part works within the given board and access that functionality to have it executed within a given time.

6.1 Introduction

This device will have multiple components that will be used and/or involved with our software development. The movement of this device will be controlled by 4 motors. This device will also have servos that are used in the arm for the movement of the pick-up of the smaller bin. To control these motors and servos we will use an Arduino. An Arduino is a microcontroller that we will use with the PCB to control all the movements of this device. We will also use the Arduino to use Robot Vision to identify and seek the smaller trash bins that the device will collect. To do this we will use a specialized sensor camera called the Pixy Cam 2.

The Pixy Cam 2 comes with built-in functionality which is perfect for the functionality of this device. The camera will be connected to the Arduino and will work together to find the smaller bin and will control the movement of the robot towards it. The other major component of our software development for this device will be a mobile application. This application will be able to power on and off the device, receive notifications of status updates on the device, and will be able to control the device manually with the mobile application. This device will be autonomous as the device will be able to seek a smaller trash can, move towards said trash can, gauge the distance for the length of the arm movement, and pick up and empty the contents of the trash can into its main bin.

6.2 Arduino

The Arduino are integrated circuits that are basically tiny computers that can run small simple software programs. There are a lot of different kinds of Arduinos that all have different functions. We will use more powerful ones for this device with a better processor. There is also the Arduino software environment which makes the Arduino great for beginners as it Is easier to program. This is one reason why we decided to use this in the device. The Arduino is one of the easiest and most seamless programming experiences when working with circuit boards. Also, the Arduino is cheap which is great for keeping us under our budget. To work with Arduino, we will use the Arduino software. Once downloaded you will connect the Arduino to the computer and launch the Arduino environment so each can communicate with the other respectively. Below you can see the IDE in Figure 47.

Figure 48: Example of the Arduino Software. Image Courtesy of Charmed Labs

6.3 Movement

In the IDE, since we are using stepper motors we will include the stepper motor library in the code. For stepper motors, it is required to define a variable that is the number of steps per revolution. Then in the loop function, we will simply have the stepper motor rotate for each one of the stepper motors. We will program the motors to go forward, backward, and turn left and right. For the servos, we will initialize the servo pin location and the servo initial position. To program the servo we will set an angle that the servo will turn to and also call a function to run it in a loop. We will add functionality where we can control our written code by mobile application.

6.4 Robot Vision

This device will implement Robot vision. Robot vision will use cameras that act as sensors that will act similar to eyes to see things such as humans. Cameras are ideal to do this as they already act as eyes, are relatively inexpensive, and have plenty of algorithms available to extract information from images. The process of robot vision is as simple as taking a picture, processing it, and calculating the information. When designing, we initially thought about implementing LIDAR for this device, which stands for "Light Detection And Ranging" which uses lasers to measure the things around. In this case, we wanted to use it for room mapping with an Arduino. However, we decided not to as using LIDAR can get expensive, complicated, and all-around inconvenient for our device design. Instead, we found an amazing inexpensive device that is perfect for our robot called the Pixy 2 Camera.

6.5 Pixy Cam 2

The Pixy2 Camera is a great camera. Unlike a regular camera, the Pixy2 is an intelligent camera. This device has an onboard processor that allows it to do things such as object recognition, line-following, intersection detection, and even simple barcode reading. The Pixy2 has all kinds of outputs that can be used with all sorts of microcontrollers. In our case, we will be using this device with an Arduino. The Pixy2 can learn to detect objects that we will teach it (our smaller bin). The camera can also track every object in its field of view. To control the Pixy2 Camera we will use the PixyMon application which uses the USB port. With this we will see the video, configure, and debug the camera. This device supports C/C++ and Python and everything is open source. In Figure 12, you can find an example of how the Pixy2 Camera learns objects through the PixyMon software.

6.6 Autonomous Functionality

To train the Pixy2 camera, we will simply press a series of buttons on the Pixy2 and then the camera will try to lock on the object and determine its color. When you repeat this process the Pixy2 will successfully learn the object which is our smaller bin. When we have successfully identified the smaller bin. We will retrieve the information from PixyMon for the Arduino to use. We will control the motors in such a way that once our smaller bin is identified, this device will start moving towards it. We will also use the Pixy2 to gauge the distance in order to make sure our device successfully picks up our bin. Once this step is down the arm of the device will successfully pick up and empty our found smaller trash bin. In Figure 13, you can see the logical steps of this.

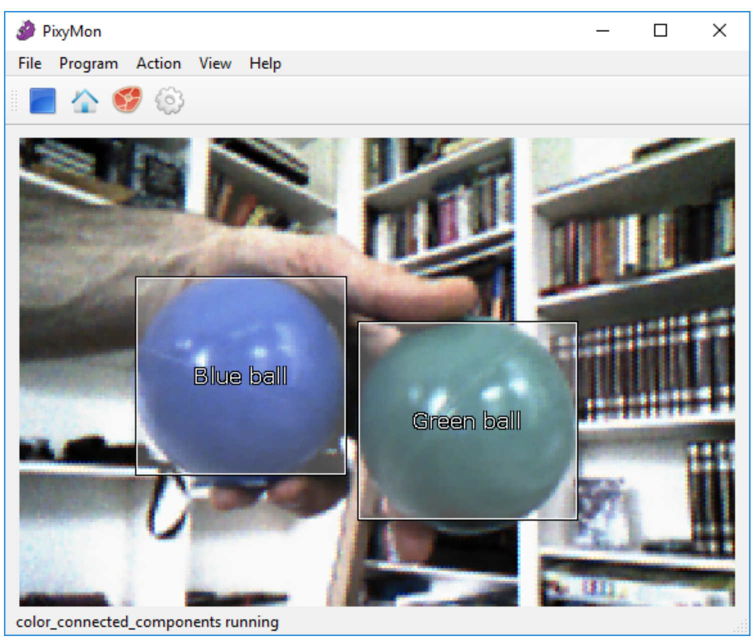


Figure 49: Pixy2 Camera Object Detection. Image Courtesy of Charmed labs

The figure above shows an example of the Pixy2 camera software called PixyMon. We will use this software to use the robot vision functionality to be able to identify the smaller trash that will be picked up and then empty into the main bin attached to the body of the Trashporter device. This function will be able to send information to our microcontroller such as coordinates. With this information, we will be able to control the motors of this device to be able to move towards our smaller trash can.

Using this software we will be able to save objects we will identify using the robot vision functionality. Having the ability to save objects will let us be able to recognize the smaller trash can in any circumstance even if the device has been turned off, or even if there are multiple objects in the field of view within the camera. The Pixy2 camera will be able to identify the object no matter what. There is an example from the Charmed website that created the Pixy2 camera that demonstrates how it is able to distinguish items from one another which would be helpful if we need our Trashporter device to be able to avoid obstacles that are in its way. This can be something we add after we successfully complete developing and implementing the autonomous algorithm that will be able to complete the main task of the device. Being able to collect the contents of the smaller trash can.

6.7 Mobile Application Overview

The other major component of our software development is a mobile application. This application will contain several buttons. The power button, and the controls when we want to control the device manually. The device will also receive a notification when the device is full and when the device has successfully emptied a smaller bin. The application will control the device manually as we will use this functionality for testing. To do all of this, we will use the Blynk app.

6.8 Blynk

Blynk is a software platform that lets us control, monitor and interact with our Arduino with our mobile device. Blynk provides a set of Arduino libraries and a cloud server. The Arduino will be connected to the internet to the Blynk cloud server which we will be able to use with the Android and/or iOS Blynk app. The Blynk app has a widget box that we will use to create controls and buttons to interact with the device. We decided to use this platform because it has a lot of free resources and works seamlessly with the Arduino. With the Blynk app, we will be able to completely control and program our Arduino any way we want it to be. We will first go into the Blynk development IDE on the browser to start programming our Arduino. We will first create a new template and then we will name the component that we want to program on the Blynk IDE online application. Then we will select our hardware which will be an Arduino Microcontroller or any other hardware someone would use, but in our case, it will just be the controller. Then we will select a connection type. There are many connection types such as Wi-Fi, Bluetooth, USB, micro USB, etc. However, we will be using a Wi-Fi adapter to connect the blink app to it as it is the fastest way to get a connection right now. There is also an option to use a description to describe what the component is and what it is supposed to do. We will be writing a description for each component for ease of use so we can look back and check if something goes wrong.

Then we will add a template if needed. Blink app has many templates to use to control microcontroller applications such as a switch, slider, buttons, and for many different other uses. Then we will select a manufacturer which would just be our organization in our case it will be the “Senior design team”. Then we can even define the category this component will be under and this will be helpful to define as the Blynk app suggests other functions and buttons to use to control it automatically while developing. Then we will define the meta-data which is basically extra information for the component that the app will use. The meta-data contains data such as the device name, the device alias, the location, time zone, hotspot name, and more.

After that is defined we will then define the data streams. Data streams are a way to structure data that regularly flows in and out from a device. For this particular section, we will use it to get the sensor data of the sensors that will detect if the device is close to a wall or not so that it can move away from it. The virtual pin data stream is what we will select for the section. The virtual pin data stream is how we will decide where to read the data from. It is similar to the pins on the microcontroller however it is virtual. The virtual pin data stream will have the name, alias, virtual pin type, and maximum and minimum values.

There are also advanced settings we can use as well but we will not. When that is all done we can finally start the development of the blink app in the PC browser. There is one option called a web dashboard however we will not be using that option as the web dashboard lets you control your microcontroller from your PC. The section we will be using for our project is the mobile dashboard, in this section we will be able to develop the mobile app and even add widgets that will automatically return code in C and import any needed libraries for our software development.

Then we will have to export the code we developed to the Arduino software with the necessary imports so it recognizes the wet Blynk software. Once that is finished we will download the Blynk app. On the Blynk app, we will first log in with our username and password and then we will type in the Wi-Fi password and name in order to connect to the microcontroller that will have Wi-Fi capability on it that will be connected to the same Wi-Fi spot. On the Blynk main app screen, we will simply be able to add widgets that we can use to control the defined components we did earlier on the Blynk browser software IDE. This will make us able to have complete control of our components so we can control our device manually for testing purposes and be able to receive notifications of status updates such as if the bin is full or if a collection was completed.

In the next page you will see our Autonomous Functionality flow chart for the collection process of the smaller trash bin as well as the structure of the Blynk software we will be using to create the mobile application.

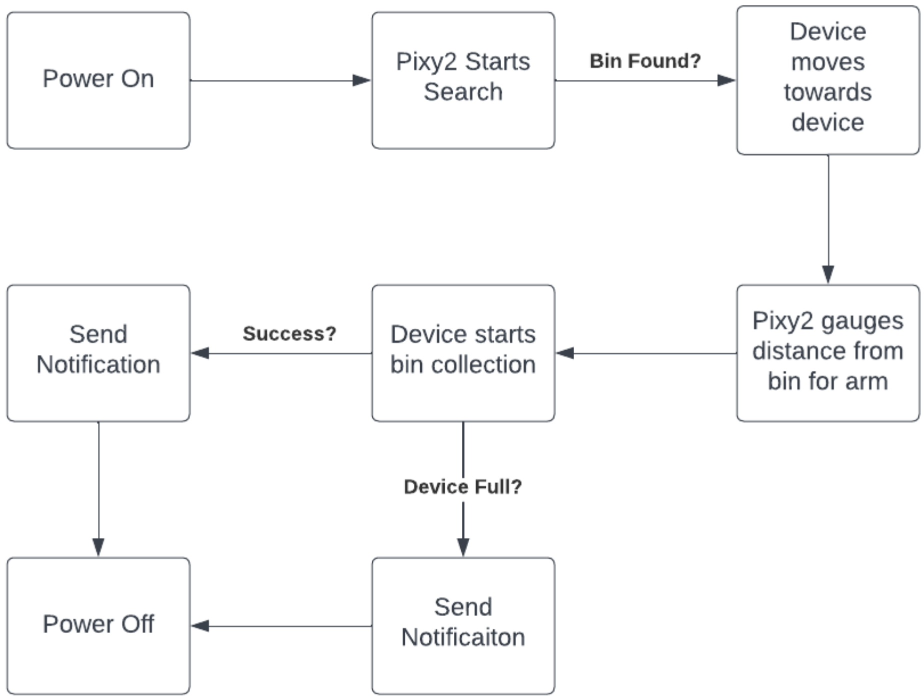


Figure 50: Autonomous Functionality Flow Chart

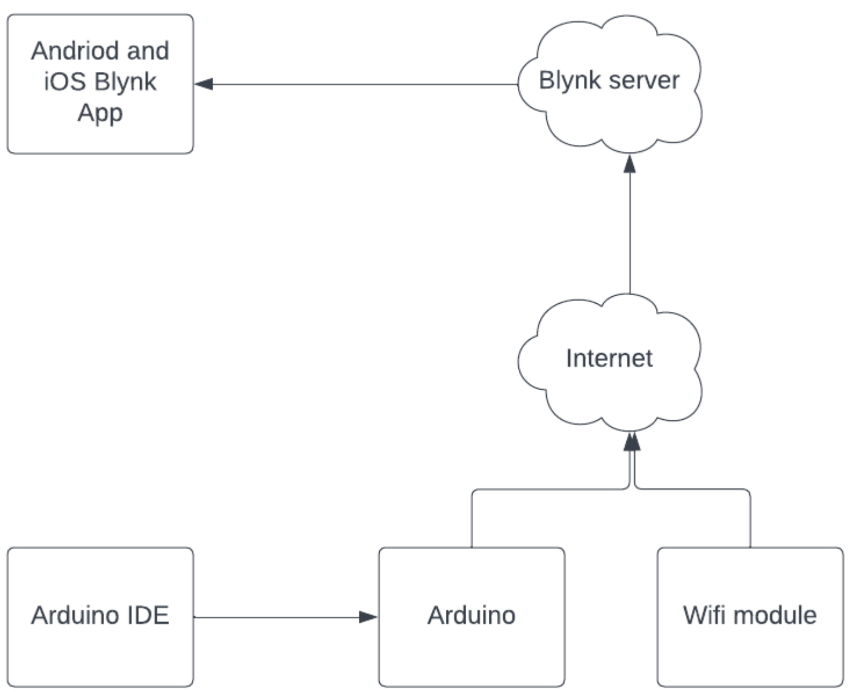


Figure 51: High Level of Blynk Structure

6.9 Autonomous Algorithm

To make this device autonomous we will be using an algorithm that will allow the device to seek the smaller trash can and start moving towards it until it reaches a specified distance to be able to pick up and dispose of the trash. To do this we will be using the Pixy2 camera. The Pixy2 camera is a smart camera that allows you to identify objects based on color and use it as a sensor to judge the distance of any object you want to. Using the Pixy2 camera we will be training it to recognize a smaller trash can that the device will regularly pick up and put back down. To do this we will put the object in front of the camera and have the camera focus on the object multiple times until it has Marn the object and it's built an algorithm. Once the object is learned we will save the object as a "smaller trash can".

Once we have correctly added the object to the pixie cam enough so that we are able to recognize it off of plants we will be using the demo provided with the pixie to the camera to help develop our algorithm. The Pixy2 camera has a demo called the pan until demo. The point of this demo is to show how the Pixy2 to the camera can track an object with movement. In order to do this, we will use the pan and tilt mechanism which can be found on a variety of retail websites. The pan-tilt mechanism can move side to side and up and down and pretty much move in a range of motion where the camera will be able to track any object in sight. The pan-into algorithm works by using the information received by the pixie to two accordingly move the motor to servos in the correct direction to have the object locked in the center of the frame of the camera.

The Pixy2 camera gives many types of information to our Adriano such as the position in the X and Y position and also in the Z position. Basically, we will have the position of each object we want to identify in integers and decimals that we would incorporate in our Arduino. With this demo algorithm, we can modify it where we send our Pixi to camera information, including the object's coordinates, to our Arduino. With this information, we will program the Arduino to control the motors to go in the direction of the object. To break it down PR, if the object is on the right side of the point of you we will have the motors from the front to Mulders turn the opposite way such that the robot will move in the right direction. If the object is in the field of view on the left side we will have it so the front wheels will move an opposite way to the left side. If The object is farther from our set range distance that is needed to properly pick up and dispose of the smaller trashcan we will have the device move forward until it reaches the set variable distance in the IDE code. In the diagram below (Figure 51), you can see a visual representation of our algorithm that will be used to control the device in order to seek the object (the smaller trash can) that will be picked up. The first step is to turn on the device. Then the device turns in a 360 motion until the object is in its field of view. Once the object is in the field of view, the device shall move forward. If the object is not centered in the field of view, the device will center itself by moving right or left. Once the set distance needed is reached, the pickup process of the arm shall begin. This algorithm will be running every time the device is started from the mobile application. For now, this will only work for one object but in the future, we might adjust it to be able to pick up multiple objects. Below you can find the autonomous algorithm as a flowchart in Figure 52.

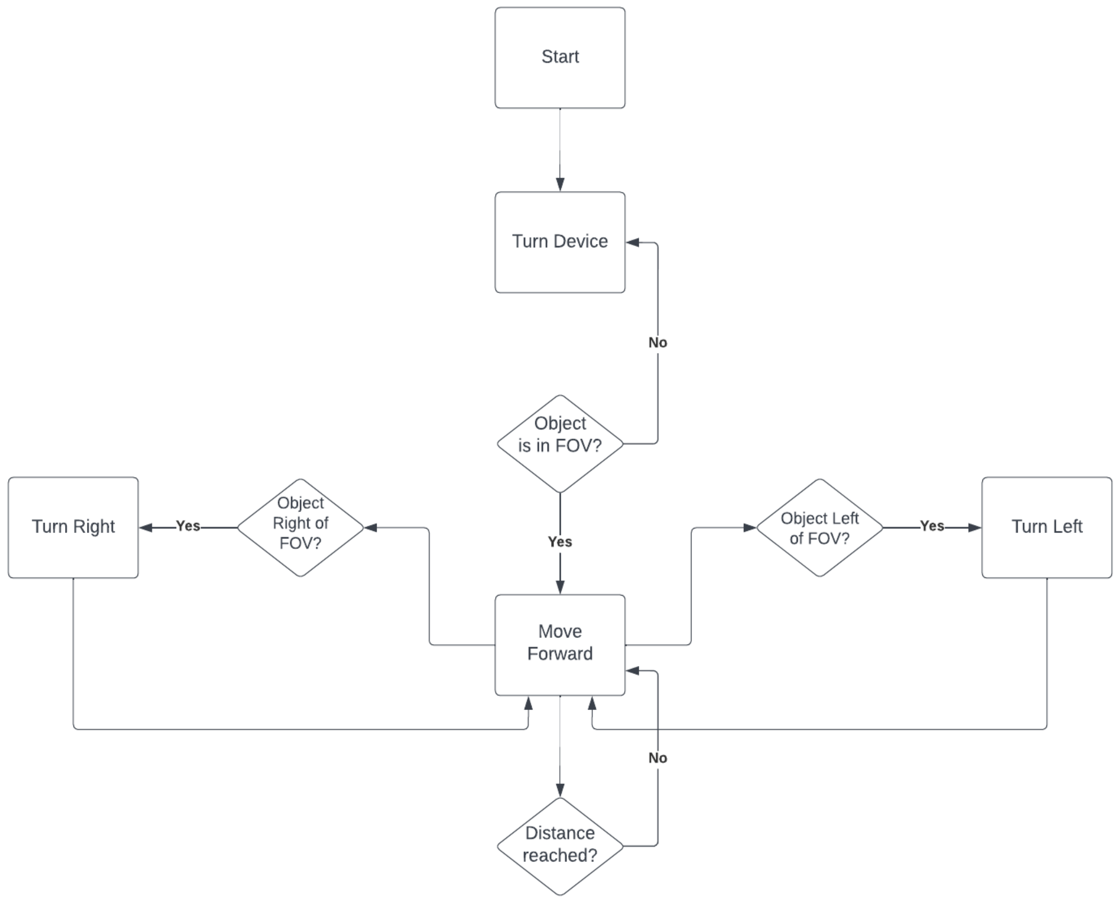


Figure 52: Autonomous Algorithm Flowchart

7. Overall Design

In this section, we will explain how the device will be constructed and how it should function. When researching the different designs and ways that this would work, there were many ways to go about this. We came to an agreement on how the design should be and look like, as well as function, as we do more research, and we meet it things could always be changed as new ideas developed because it would be easier to implement a certain way or some other ways that we think it should be designed is unnecessary. This was not an easy discussion to have because all of us would have to agree on one uniformed design, as well as the vast number of ways that this has been done. It was difficult to find something that is the same as what we envisioned, so we would have to take the liberty of brainstorming modifications to a preexisting idea to fit our project.

7.2 Design Overview

We initially went over many different types of designs for this trash device. One design we considered was having an actual trash can that can be found in the kitchen or a classroom that is tall metal with a lid. We like the idea of having a lid on top of the trash can because we could potentially attach a servo on the hinge to control the bottom of the trashcan to be able to open and close the lid manually after a device has picked up trash. However, we decided that it would be easier to not have a lid. We were also going to attach two wheels on each side of the trashcan so it can move similar to a Hoverboard you can buy in stores. It would've had a self-balancing component where it would balance itself out while moving. Then we would have a lift that would have moved up and down attached to the front of it that would be able to pick up and put back down any smaller trash can. The lid would open and close whenever this action happened.

We decided to not go with the design because the lift would have to move too far of a distance to empty the contents of the trashcan into the bedroom and we also believe that it would have caused many complications while developing in building the left. Also only having two wheels leads to a higher chance of the device tipping over on itself because the front left could possibly be too heavy and pull it to the front which will lead to the device falling. The device could also possibly fall over if it knocks itself over by running into a wall or if a person accidentally bumps into it. Also, we would like to make the device small enough with our defined size constraints to be able to be picked up, moved around, and transported easily. If we were to use a tall metal trash can, it would not be easy to move around or transport at all. Also, this can lead to the trash can being too much too heavy for operation. We decided on having a smaller design in order to be able to save money and be able to transport and move the robot easier and also to help with the efficiency of operation and functionality of the robot which could be seen in our HOQ diagram.

In the figure below, we have one of our early designs we came up with during one of our meetings while brainstorming. We made this design in AutoDesk inventor, a CAD modeling software where you can create, design, build, and assemble parts that you create within the program. In this example we worked with the measurements to make sure that part is in scale with each other to make sure that we will not go over our size constraint. The parts that we created are the bin, the base, all four wheels, an arm mount that will be attached to the base, and two arms with circles on the end to represent the grips that will grip onto the sides of the trashcan then rotate. Then we assembled it all together in the program and made sure the measurements and constraints are what is needed for the specifications of this project.

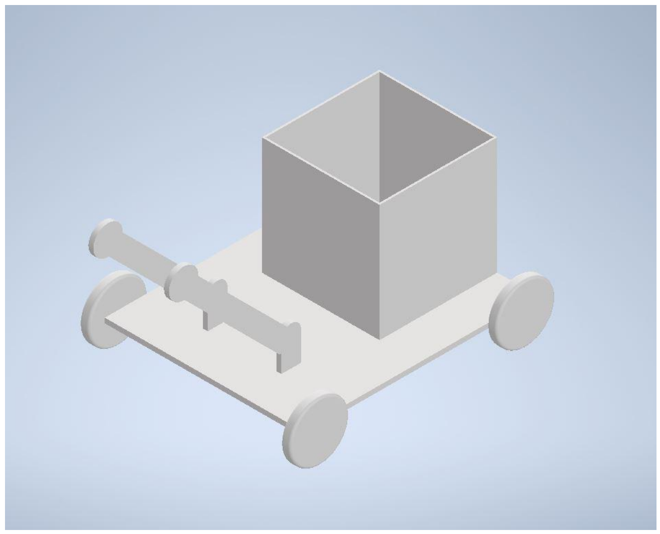


Figure 53: One of our early designs in Autodesk Inventor

7.3 PCB Design

We will create a custom PCB to connect all our components for this device. Before we actually create the PCB we will plan the function of the circuit based on the requirements of the device such as having a power switch, connections to the microcontroller, battery source, motors, and servos. To source components, we will use multiple sources such as websites like LSCS that have a large selection and we could also buy a PCB board on the same website as well. To create the schematic for the PCB board we will use the EasyEDA. EasyEDA has built-in libraries that contain many parts. However, for any parts, we do not find we will use the Internet and websites to find them such as EDA. Then once we place the parts on the schematic we will organize them in a way that they can be routed easily. We will use the autoroute tool on all of our eagle two to get as close as 100% routing completed. If not we will route manually by hand any components that did not get routed by the auto-router. Once the routing is done then we will start setting up any vias and pours for the PCB and then once that is finished we will test the PCB with the EasyEDA software and then it is ready to be created.

7.4 Claw Design

One of the most crucial hardware components is going to be the claw, it is what makes our robot functional and brings the project together. Because of how unique the measurements are going to be for our specifications this would not be something that could be standardly bought from outside means, so we intend to 3D print the design. We would design this on a modeling software program, ensure the dimensions and calculations are correct for what we need it for, and put in an order to print it out. The claw will be driven with a couple of motors and sensors to know when it has been considered as holding an object. It will act as a two-part system where there would be an arm attached to the main chassis of the robot, then a static arm that is used to extend the distance of the claw, it will then have the claw line up with the target and a servo motor will run to contract the claw until the object is firmly held.

As for the construction of the claw, it will consist of an arm, multiple servo joints, metal rods for structural stability, and a connected lift system. The claw will be connected at the end of the joints acting as a wrist so it can flip the bin backwards, dumping the trash into the main bin. A small but powerful enough servo will control the angle in which the claw is always positioned, making sure it is stable enough to hold a still position and flip backwards when needed. It will then be connected to a long metal rod that holds the claw where it would have enough distance for it to reach the object being grabbed, this part is mostly static as it is connected to the main chassis and lift system. At the end of the arm section will be another servo that is connected to a gear system which will allow the lift to move the height of the claw system. This gives a full range of motion as it can move vertically as well as the independent wrist, for this design it will simplify the design that we do not have to have the claw move horizontally because the robot will align the position for it. The claw should be big enough where it can pick up variable sized bins, but in our demonstration, it will pick up the same sized bin, this is allowed because there will be a functionality where the claw will detect that it has fully grasped an object, making sure not to crush the object, but also firmed gripped as to not drop it when in motion.

There were many designs to research from and have been inspired from, the design we chose has to be a mix of a few of the researched topics. We can use some of the simpler designs to test the software so that we can ensure it can be upscaled to something more complicated when being implemented. The first design that was looked at was a basic project where it consisted of two wooden sticks that were angled, a metal rod in the middle of the claw which acts as tension for the claws to be pulled in, and a servo connected to the end of the rod so when it moves in one direction, it adds tension, pulling in the claws to the center, this is good to test, but ultimately needs to be adjusted so we have something stronger suited for what we are trying to achieve. The other designs were based off of pre-built kits that had all the parts included and instructions to follow along to recreate what was intended in the project, we could definitely use some of the side parts, but the scaling for it did not seem to fit what our actual project needed. Using the design, we could get an idea of how it should function and try to replicate the code as best as we could with the observations made. We have decided to do so and mix the different design ideas to fit the idea of making the project not too overly complicated by using a sort of tension technique and 3D printing a custom claw so that it can be strong and sturdy enough to carry large loads without any structural failures.

7.5 Lift Design

The Lift Design consists of the connection between the end of the arm and the system used to raise the height of the claw. The lift uses a gear and track system that has the arm attached at the end of the joint, connected to the belt, and when the motor runs, it pulls on the belt system to raise the level of the claw. This will be programmed to sense the height of the bin and grab the central part of the bin to make sure when it is flipped vertically, the contents of the bin will be intact with the main bin. When being lifted, it relies on a pulley system to easy the amount of load being carried, then the motor will generate power to pull the belt along the vertical beam, when falling, it mostly uses gravity to have it retract downwards, but with a slight force applied so that it does not just fall out right, potentially damaging the parts. The program should know where it should be lifted based on the information provided by the camera when tracking the bin as well as identifying it should be the bin picked up. This will measure the height and know that the claw should stop at the correct height of the bin and pick it up from there, because for testing and demonstration, the bins will be the same height and weight, so because this is constant the results may be different if we were to test other types of bins. Other sets of bins used in the future could be considered edge cases in which it can run into some problems because of unexpected height or weight when the bot tries to lift up a different type of bin. One rule that it should always follow is that the sub trash bins should always be smaller than the main bin, being that its purpose is to collect and deposit multiple bins from each room and bring it back to an easily accessible place by the user to discard.

When researching different types of lift systems, there were many options to choose from. The reason that we chose the one we did is because it closely fits the requirements that it needs to be to function correctly within our project. The other lift systems like the Scissor Lift would be too bulky for our design because it has a solid four-sided base for stability as well as taking a large amount of space when trying to design it. This would be impractical for the purposes of this project as the main focus should be the bin itself and not have the lift be the foundation of our robot. Other designs like the Linear Slide could work as intended, but one thing to keep in mind is the width of the mechanism, because the robot will be moving around corners and might go by objects, having parts stick out, adding to the width of the robot can have it bump into these objects and potentially damage a major component making the robot inoperable. We have to keep in mind how much space the robot is taking up as it can easily run into things, though the camera can detect objects and avoid them, the lens of the camera is only so big that it does not know the dimensions of the robot, as well as possible blind spots it cannot detect. It is important to be hyper aware of these edge case situations because it makes the difference between a potentially working product and a consistently working product. From the research that was conducted, many of the parts utilize pre-built kits that came from the website which they sell, this could be a large time saver as the materials used to stabilize the system is already provided and we would not have to make it from scratch. It also comes with the mechanism of the motor and gears, so learning more about the schematics and source code will be a large help in not having to develop the code ourselves. Reading up on the libraries can let us manipulate the code so we can have it adapt to the functionality we need in our project. Having less to develop from scratch will save us time in the future when building it as well as ensuring that the device works as intended.

7.6 Base

Because now we are prioritizing size and cost we decided on a different design where the device has a bin instead of a trashcan so the arms will have a shorter distance to travel in order to dump the contents of the trashcan into the bin attached to the device. Also having a bin will lead to more capacity to collect garbage collection. A bin instead of a standard trash can will also let us be able to detect when the trash is full easier so that we can send a notification to the mobile app that we will create. The bin will have more surface area which will let our weight sensors be able to get a better reading. The bin will be attached to the base that we will create custom ourselves with a light, strong material which will most likely be made out of plastic that will go on top of our metal frame. This base contains the majority of our components such as the batteries, the hardware, and all the electrical components including the microcontroller and the PCB board. It will also contain the Pixy2 camera with a gimbal, pan, and tilt mechanism if needed. We need to make sure that the electrical components are protected from possible human intervention and make sure that the moving components will not interfere or even damage the other components.

Under the base, we will have four mounts and four motors that will be used to move the robot. In this motor configuration, the drivetrain will be a four-wheel drive which should be able to be quick and agile. These motors will be connected to the microcontroller on the base along with the PCB board. We will also use special wheels that will be connected to the stepper motors as the wheels can move in many directions easier which will be perfect for our Trashporter robot. As the Trashporter device will make a lot of turns and adjustments to be able to seek out the smaller trash can, having a setup with our stepper motor and special wheels will help with the centering functionality of the autonomous algorithm. The autonomous algorithm will automatically adjust the center of the device with the center of the smaller trash can so if we can have more concise moments, it will make it easier to do said adjustment. Our stepper motor and wheel combination will be perfect for movement under the base.

7.7 Bin

The bin will simply be in the shape of a box that we will either custom make with wood, plastic, or buy a box of our required size online or at a store. This is one of the less important parts as it doesn’t really have a function besides storage of the trash that we have collected. However, one element of this part we want to consider is the weight. We want the device to be as light as possible. It is important that we use a light enough material so that it does not interfere with the functionality of the robot. Having extra weight can significantly decrease the performance of the robot. However, even though we do want something that is very heavy, we will need something that is strong enough to hold the items collected from the trash can that will be picked up from the autonomous process. We also do not want the bin to be too big so the arms do not have to be massive or very long to just be able to lift the smaller trash can over it.

We initially decided to use a taller trash bin like one that can be found in a kitchen. Then we would have planned to have two wheels on each side of this trash can to act as a self balancing moving robot. We decided not to go with this design as the lift we would have had to design for this size trash bin would have to be very tall. The lift would have a long way to go and would not be within our size constraint. Also, since we originally planned on making it a self balancing robot, the lift would be too heavy if attached to one side of the bin which would cause the Trashporter device to be tipped over easily.

7.8 Safety

In this section, we will be discussing the safety precautions that we will take while developing the Trashporter device. Taking the appropriate safety precautions is very important as we want all our team members to be safe, as well as anyone that might be operating our device. While developing our device we will make sure to have safety in mind. This will be our first priority. We will be focusing on three different aspects of safety in relation to this device. The first one will be electrical safety which will include the safety measures while developing our PCB board and our electrical connections, circuits, etc. We will also want to make sure that we safely use the equipment while developing the electrical components that will be used to create the necessary connections and hardware components such as the PCB board for this device. Then we will also go into the safety precautions of the mechanical moving components of the Trashporter device to make sure that it doesn't possibly hurt our team members while developing it or someone else while operating our device. We will also discuss how to make sure that components are protected from various outside forces. Then the last thing we will work on in relation to safety precautions is the actual movement of the device to make sure that it doesn't possibly bump into anyone or anything. We will do this by going into the subject of the autonomous algorithm and object detection.

7.8.1 Electrical safety precautions

We will be taking many safety precautions and be very careful when working with the electrical components of a Trashporter device. There are many things that could possibly go wrong while developing a circuit, PCB board, working with the batteries, and more such as the equipment. Working with electricity can be very dangerous as you can possibly run into things such as discharge, short circuits, and even potentially get electrocuted while working.

There are many factors that we have to take into account when working with electricity. Just to start, we will need to make sure that we are wearing the proper safety equipment while developing our circuit. We will wear shock-resistant gloves and goggles whenever we are developing components. We will also have to make sure that we are not wearing any electrical conductive clothing and use an electric anti-static conductive mat that will also protect us when working with electrical components. This might seem like a lot to do for ensuring that we are taking the necessary safety precautions, however, we believe that it is better to be safe in the first place rather than to be sorry later.

When it comes to soldering, we will make sure that we are wearing the necessary protective equipment to make sure that we don't possibly burn ourselves from the sparks and the heat. We will be wearing heat-resistant gloves and will have our goggles and/or a face shield with long sleeve clothes to prevent any possible sparks to fly onto us which could lead to potentially burns which we would not want. We also want to make sure when we are soldering the components that we are in an open, well-ventilated area as soldering can lead to potentially harmful fumes which could cause us to pass out or even die which is not what we want. Soldiering could be a very dangerous thing to do without the necessary safety measures in place.

The last thing we want to make sure of is that we are properly using the equipment that will be used for testing electrical components and hardware. We will be using a multitude of devices to do our testing on our trashporter device. Some of the devices we will use is a multimeter which we will supply electrical current to the hardware components to test if the voltage we are all putting out is correct. When doing this we want to make sure that none of our hardware components possibly heat up or start smoking. If this happens we should stop immediately and make sure that the smoke and hardware are getting cool air and not in a heated state. We will also have an extinguisher on hand just in case anything sets on fire. This can easily cause a fire if we continue the testing. The proper procedure is just to scrap that part as it will be burnt and check our math and design to make sure we are not accidentally making a component break or possibly causing a fire.

When actually assembling our electrical and hardware components together, we want to make sure that we are placing the components correctly where they should be. When we do this we avoid all sorts of issues that can come up that would cause the safety of the device and others to be at risk. Proper assembly of the electrical opponents connected to the hardware components will make sure that nothing will short circuit, spark, or even discharge. These negative effects could potentially damage other components on our device that will affect the full operation. We also want to make sure that nothing on a device conducts electricity. This is why we will make sure to use a material that is not conductive on our device to avoid any potential electrocution when working on the device.

We also want to make sure that the wiring of our electrical components to the hardware components is not interfered with. To make sure it's not interfered with we will make sure that our wire placement is placed in a way that is neat and organized and out of the way of the moving components. If the moving components are possibly hit or even cut our wires this can lead to complete malfunction of a device and even potentially cause sparks or even a fire. So making sure that the placement of our wires and possession of them on a device is also something we will take into account that will make sure that we have reached all the proper safety precautions for working with electrical components.

This part of our safety precautions is potentially the most important part of the three sections we will discuss as working with electrical components can possibly have many dangerous aspects to it. This is why it's important for us to follow all our safety precautions. Also, we will be using the safety precautions that are given by the IEEE standards that we wrote about in the related standards and constraints sections of our paper. We would not want to cause a fire or potentially have sparks flying or potentially burn ourselves while developing the hardware components of the device. To summarize, we will take the necessary precautions while working with the equipment, while soldering the PCB board, and also working with the wiring and setting up of our electrical parts.

In the figure below you can see some of the safety equipment we will use when working with the electrical components of the trashporter device. We have gloves to protect our hands from any potential heat and sparks, and also protect us from possibly getting cut. The gloves will also protect us from getting potentially shocked from working on the components as well. We also have goggles that will protect our eyes just in case anything goes in our direction such as sparks while soldering components on our board. The last thing in this picture that we have is a fire extinguisher which is always useful to have while working with hardware and electrical components as they can possibly heat up, smoke, or even cause a fire. If we notice that this is happening to our components, we will immediately use the extinguisher on it to make sure that the fire is completely out. This is just some of the equipment that we will be using to make sure that we are safe. This is a major part of our overall safety precautions.

Figure 54: Some of our Safety Equipment

7.8.2 Mechanical Safety Precautions

Our project will have many different moving components that can potentially be harmful if we don't take the necessary safety precautions. In this section, we will be discussing the appropriate safety precautions that we will take for moving components and for any potentially exposed components. We will also talk about our mechanical lift that will be used to grab, lift up, flip over, and then empty the contents of the smaller trash can into the larger bin which will be attached to the main body of our trashporter device.

When the device is ready to be moved around, we will be using four motors and four motor mounts to act as a drivetrain for the trashporter device. These four motors will act as individual motors that will move in sync which will be programmed by the microcontroller. The four motors act as a four-wheel drive similar to what could be found in vehicles currently for sale on the market. The only possible safety issue that we are worried about is the physical axle that will be sticking out connected inside the stepper motors we will be using. This axle would be stuck out of the physical motor body. While moving around on the floor this axle can possibly catch onto items that were lying down on the ground. For example, if the axle gets intertwined with a string it can completely disrupt the movement of the robot, which can potentially make the robot unsafe as it has other moving mechanical components that can possibly fall over. Also in general the movement of the robot might become erratic which we don't want as this robot is supposed to be autonomous and have precise movements.

For this robot, we will have all our hardware components and electrical components connected to the main base of our trashporter device. The base will contain components such as our PCB board, our microcontroller, our battery packs, and more. It is important that these components are protected while the device is in use. The device will be moving around and will also have a moving component attached to it which will be our mechanical lift which we will go into more detail on later. If the components are interfered with it can potentially damage our robot and cause sparks to fly or even cause a fire. It's also important to protect these components so we don't possibly get electrocuted while working on the robot. The overall theme of our safety precautions is we have to make sure that nothing smokes and nothing is set on fire. This is why I keep repeating this sentiment when discussing our safety precautions. Something is more likely to smoke or catch on fire than our device actually physically injuring someone like our team members or someone else that operates our trashporter device.

One thing we could do is to make sure that when we are working on the robot the power is off. However, another thing we can do is to make sure that we are taking the necessary safety precautions for our hardware by making sure that the hardware is not exposed at all. One option we have is having another base cover that will go on top of it. We also discussed having a compartment that can open and close so we will have easy access to our hardware components just in case we have to fix something or diagnose an issue if the trashporter device is malfunctioning. Also, we will have a moving mechanical arm that will go up and down and we do not want it to accidentally snag on any wires or mess up anything with the gears so we will have to make sure that we're placing the wires in their appropriate spots and are making sure they are hidden outside. We also want to make sure that the wires are long enough for when the lift will be going up and down.

This project will also use a mechanical lift. This lift will have three different functions. The first function will be to grab onto the smaller trash can. The second function will be to lift the smaller trash can while it is gripping. The last function of the mechanical lift is to rotate the wrist component of the lift to be able to empty the contents of the smaller trash can into the larger bin attached to the main body of the trashporter device. This brings up a possible safety concern as while we are doing our mechanical work our fingers can potentially get hurt. When we want to work on the lift it is important to make sure that the lift is properly lowered down so it cannot drop down suddenly if something fails. If our slider lift is carrying the weight of itself and the weight of the bin filled with the contents of trash, the slider lift can become something very dangerous if it suddenly comes down. We will want to make sure when we are working on our device, especially if we are working with the mechanical lift, that it is fully down to avoid potentially hurting ourselves. Specifically our finger.

Another thing that we will have to worry about is the wires as it will be connected to the motors and servos. The motors and servos will be the way we will move each component of the mechanical arm. Motors will move the arm up and down and the servos will be used to grip onto the smaller trash can and another servo would be used to flip the trashcan into the bigger bin connected to the main body of the trashporter device. We do not want the motor or servo to get in any contact with the wires that will be connected to our microcontroller or else it can potentially cause harm and mechanical failure by having the slide lift go down quickly or even cause a spark which could lead to a fire.

This project will also incorporate gears for the movement. The gears will be implemented mainly with the lift mechanism. We will select a gear ratio that is strong enough to lift the slide lift with the weight of the smaller trash can. Because we are going to be using gears to lift the slide lift, we will need to make sure that the gears are not exposed from the device. The gears cannot be revealed as we do not want anyone touching the gears as it can lead to injury. The safety precautions that we will take to make sure that no one gets injured from the gears is making sure that the gears are covered up with material or with the frame of the mechanical lift. We will also want to make sure that the gears are not sharp by sanding them down just in case someone accidentally touches the gears.

The last safety precaution that we would want to worry about is the actual movement of the transporter device itself while the autonomous algorithm is running. We will implement a level of object avoidance and the trashporter device will have extra precautions made to make sure that the device cannot potentially injure someone. When we test and demo the trashporter device, the room and space have to be clear of people and obstacles that could potentially get in its path. Here, we plan on having the trashporter device have a metal frame which we would not want to potentially run into anyone with. Since the device will be lower, it can potentially hurt someone’s shins or a different part of their legs. Also, we want to make sure that no one is in the area as we want to grip onto the smaller trash can and do the necessary procedure to empty the contents into the main bin connected to the main body of the transport device. We also do not want this device to move too fast as it can get out of control easily and cause damage to other objects or even people in the room.

8. Integration and Prototype Testing

In this section we will describe how we will integrate the components to our Trashporter device. We will also go into extensive detail of the testing procedures of our prototype such as testing the software that involves our microcontroller controlling the moving components such as the motors that will drive the device and the servos that will be used in the mechanical lift. We will also go step by step to test out the autonomous algorithm that will let our device find and collect the smaller trash can autonomously.

8.1 Testing Procedure

Testing our software is important for us to make sure all of our software components on the device work correctly. We will be implementing many tests while developing this device to ensure we deliver the product with the utmost quality. We will also test the software to make sure there are no random bugs or errors that come up depending on the situation. Some of these bugs may come up when we try to work the autonomous feature of the device or when we try to use the mobile application. The mobile application may fail to connect or control the device so the mobile application will be tested as well. We will also be testing the software of our microcontroller to make sure it is able to control all the components of our device properly.

We also have to make sure all of our hardware components are working correctly to ensure that the PCB and circuits that we will be developing are correctly powering and connecting all of our hardware components together. We also want to make sure that we are getting the correct voltage output from our batteries in relation to our PCB board and to make sure we are avoiding any other issues related to our circuits. To test this will be using equipment in the lab and equipment that we have at home.

It is crucial to test all of our mechanical components to ensure that everything is working as it should to prevent any damage or possible malfunction to our Trashporter device. We will test each individual mechanical component such as our motors, servos. We will also make sure any gears we use have the correct ratio for what we will need for our project. The majority of our testing will involve our mechanical lift as there will be a lot of moving parts which can lead to multiple points of failure. Because of this, there will be extensive testing of our mechanical lift to ensure that our device will be able to complete the main task or our autonomous functionality which will be picking up a smaller trashcan than lifting it in the air to be flipped back in order to empty the trash contents of it into the main bin that will be attached to the frame/body of our device.

8.2 Software Testing

In this section we will be talking about testing the software related to the Trashporter robot device such as the arduino when controlling the moving components, tests relating to the basic movement of the device, and extensive testing for anything that is related to the autonomous algorithm. We also have testing for our mobile application that will be used to receive notifications and status updates from the device and will be able to manually control the device for testing purposes to see that all our mechanical components are all in proper working order. This makes sure that the components will not possibly fail or malfunction.

8.2.1 Microcontroller testing

To test the microcontroller we will first develop the needed code to move all the components connected to devices such as the motors, the servos, and even the Pixy2 camera. We will run tests on each of the individual components to make sure that it's operating in the proper working order. We will run the motors forward and backward to make sure that they are moving well. We will also move the servos in the specified range of motion to make sure that the servos are in working order. Then we will combine the four motors together on the drivetrain of our device. We will test all the motors by moving them forward and backward together. This will make sure the robot is able to move forward and backward correctly. We will also test the turning of the device's motors by programming it to run the two front wheels in opposite directions (back and forth).

8.2.2 Wall detection testing

If the device successfully passes the basic movement test we will be able to move to the next tests. We will then start testing the movement of the robot under multiple conditions where there are walls present. The sensors that will be connected to our electrical system will then be able to detect walls on the robot. The device shall be able to detect walls and move away from them. We will test this by having the device move toward the wall head-on. To pass this test the device will have to come to a complete stop before hitting the wall. This means that the sensor on the edge of the robot has properly detected a wall in front of it. We will also have another contact test where we will have a multitude of tests where the device will be turning into a wall. To pass this test successfully the device must not hit the device while turning into a wall.

8.2.3 Autonomous testing

Once the device has passed all necessary tests related to detecting a wall, we will test the device's ability to find a small trash bin by using the autonomous algorithm that we developed. As stated before, the device will move in the opposite direction of the wall to avoid running into the wall in any situation. This test will make sure that the device will have ample time and no collisions on the way when trying to find a smaller trash can. During testing, we will put the device in many scenarios to ensure that it will not be able to run into any walls. Most of the testing will be done at the senior design center on the UCF campus or at one of our team members' houses. We will also purchase cardboard to act as walls and create custom mazes for the device to go through as a smaller-scale test before moving to the actual building and room walls. This is similar to how scientists used to have mice find their own way to go through mazes for a piece of cheese but instead in our case it will be a robot finding a trash can.

We do not feel that the testing of the movement of the device will be complete when the device is able to move forward and backward, and turn left and right by having all the motors connected and programmed correctly to the microcontroller. Also, the motors being in sync with each other as this while causes a lot of problems with moving the device in sync. Then we will continue to test the device to make sure that it does not run into a wall at all as this will affect the autonomous feature of the device. For the demo, we do not have many walls that the device might run into as the main goal of this device is to simply find, collect, and empty the contents of a smaller trash can without any human interference.

For the stages of software development testing involving hardware components, I will be working closely with my teammates responsible as those two are working on the hardware components of this device. When they set up the circuit, the electrical wires, and the components I will be testing the hardware on the side by running various basic programs to make sure each component is functioning as expected. Then when I finish my test I will report back to them if the device says they are in correct working order or not. We will follow the IEEE standard to log and record the results of our testing as well. If the device does not work they will fix electrical components and then come back to me. Then I will test it again on my computer and software.

For the Pixy2 camera, we have to do a multitude of tests to make sure the software and the hardware of the Pixy2 camera is working correctly and as expected. The Pixy2 camera will be connected to our microcontroller which acts as the brain of our device. The main thing we want to test with the Pixy2 camera is having the ability to identify an object with robot vision that is built in the camera. We also want to be able to receive the correct coordinates of the object once it is identified. And the last thing we want to test is whether the microcontroller is receiving the correct color coordinate positions and sending the data to the microcontroller that it will use to control the motors and service of the device.

To test the functionality of the robot vision of the Pixy2 camera, we will have the camera set up across from the trash can. To make the Pixy2 camera identify and find the object easier we will spray paint the smaller trashcan a bright color such as pink or yellow to help it find the object faster. In the first initial test we will simply place some object in front of the lens and then the Pixy2 camera software should pop up with the notification which looks like a square around the object. If the square shows itself around the object then we know that the object is correctly identified. Once we get this information we can name this object a “trashcan” in the variable settings of the Pixy2 camera. With this data collected, and now that the Pixy2 camera recognizes a trash can, we will be able to continue further testing.

The next set of tests we will use for the Pixy2 camera will also be related to identifying the object. We will have a multitude of objects together in the field view of the lens. This will test if the Pixy2 camera recognizes the correct trash can out of a group of other objects. This ensures that when using the Pixy2 camera in combination with the autonomous function, that it will make sure the device does not pick up the wrong or seek and locate the wrong object which is the smaller trash can. To test this we will have the smaller trash can spray painted pink or another bright neon color that we have used in the previous testing and random objects. First, the objects will start off by being smaller than the correct trashcan to make it easy on the Pixy2 camera's robot vision function to identify the correct object. Then we will use objects of similar sizes to continue testing the Pixy2 camera's ability to seek out the correct smaller neon trashcan object. Once we do that, we will actually use an assortment of smaller trash cans and see if the Pixy2 camera selects and identifies the correct one. The other trash cans will be any common household once you can find the house section of a store with all different colors as a Pixy2 camera relies on mostly color to identify and seek objects with its built in robot vision functionality. Once we believe the device and the Pixy2 camera have passed all the tests, we will then continue working on the autonomous algorithm and functionality testing of the device. To start testing the autonomous functionality of our device will simply start with the basic drive train and the pixie camera attached to it. The Pixy2 camera will act as the eyes of the robot that will seek out the smaller trash can that will be picked up. To start the initial test, we will have an empty room with just a device without the lift and just the basic drivetrain and the Pixy2 camera attached together with one small trash can in the same room. The goal of this test is for the device to simply run into the trashcan by seeking it out using robot vision and our autonomous algorithm. When the device is started, the Trashporter robot should scan the surrounding area by moving in a 360 motion using the gimble that we will attach to the camera or physically spin the whole device around with the motors attached to the wheels. Once the camera's field of view contains the smaller trashcan of the device it will stop moving when centered with it. The Pixy2 camera will send information to the microcontroller to move the device forward. Then the bare bones main body of the device shall control the motors to move forward into the direction of the trashcan. If the smaller neon trashcan is not in the needed center of the field of view for the mechanical lift, the device will auto adjust itself by moving itself left or right to perfect the centering. This basic test ensures that the Pixy2 camera is functioning correctly and also tests the initial first step of our autonomous algorithm that will make our device autonomous and that it moves through the correct steps.

We will continue testing the auto-adjusting feature of the autonomous robot vision function by having the smaller trash can placement in many different angles to make sure that the different placement of the object will not affect the centering algorithm. We will be able to judge how successful the device is with this test and we will see how well in line the center of the device is with the center of the trashcan. It is crucial that the device is as center as possible with the center of the smaller trash can as we want it to be almost exactly in the middle when we start incorporating our mechanical robotic lift/arm system that will lift up the trashcan and dump the contents of it into the main bin attached to our main device. This test also helps us determine if the Pixy2 camera is sending the correct color and positioning coordinates information to ESP microcontroller, which will be controlling each of the motors connected to the drivetrain of the device and also the servos which is incorporated in the mechanical lift that will be discussed later.

The next step of our testing is to repeat the previous test discussed earlier except we will have the device stop a certain distance before it hits the smaller trash can. We will do this because we want to give space for the robotic arm system to be able to lift up the smaller neon colored trash can. Also, this will be crucial for the next step which will be testing the robotic claw which will lift up the smaller trash can. The robotic claw gripper and the robotic arm system will need enough space in order for the Trashporter device to be able to properly center itself in relation to the smaller trash can so that it is able to be gripped properly. The robotic gripper will use one or two strong servos so the servos will have to be able to receive information about the distance and coordinates from the Pixy2 camera as well.

Autonomous steps of the robot should be

1. Scan room for the smaller trash can
2. Stopped movement when the smaller trash can is identified
3. Move forward towards the smaller trash can
4. Auto adjusts itself left or right if needed to center itself in relation to the smaller trash can
5. Start gripping onto the smaller trash can with robotic gripper
6. Lift the smaller trash can up with the lift system while still being gripped by the claw
7. Flip the smaller trash can with one or two servos attached to the robotic gripper to empty the contents of the smaller trash can into the main bin attached to the device.
8. Flip the servos back
9. Lower down the lift system
10. Release the robotic grip on the smaller trash can

8.3 Hardware Testing

In this section we will share our discoveries and results during our testing phase, more specifically testing the buck converter.

8.3.1 Breadboard Testing

The figure below displays the LM2673 IC through hole package. Due to the packaging type it is impossible to place on a breadboard, therefore, to test our design we temporarily soldered the IC on a perfboard with breakout wire to place on the breadboard as shown in *figure 55* below.

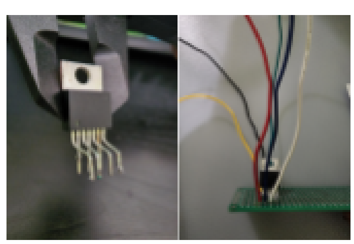


Figure 55: LM2673 Pin Package and Temporary Perfboard

Once we completed the schematic and we were confident that the part sizes were not going to be an issue, we ordered the rest of the parts. Once the parts arrived, we connected all the components on a breadboard in *figure 56* below and tested our output and input voltages.

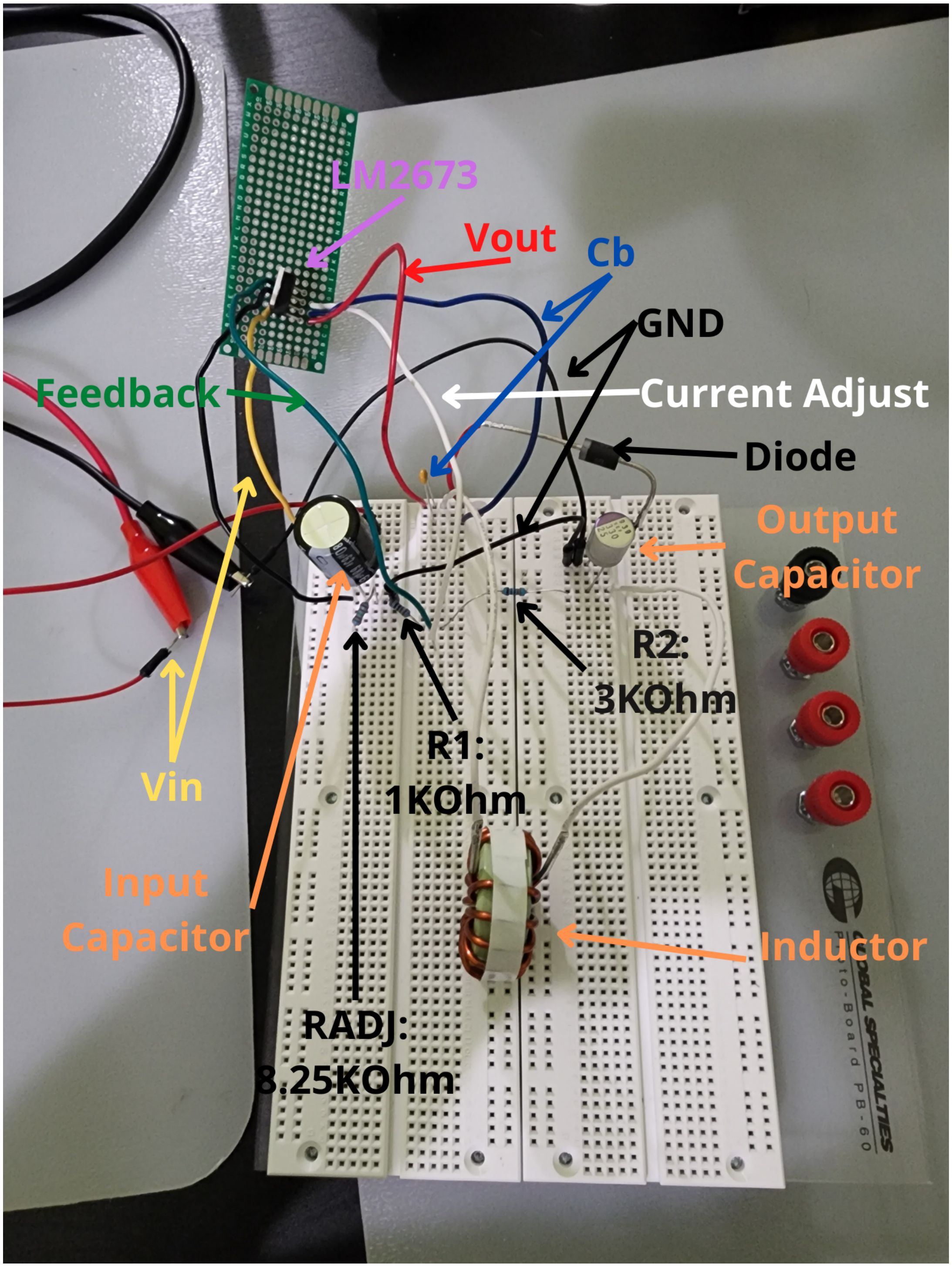


Figure 56 : Breadboard layout of our schematic in figure 43

8.3.2 Perfboard Testing

We then applied 12 volts on our input leads and tested our output using a multimeter and got a reading of a constant 5 volts. However, breadboards are not the best testing boards for high frequency circuits because they can create noise. Also the circuit looks messy on the breadboard and in *figure 57* below, we decided to temporarily solder our components on a perfboard for further testing.

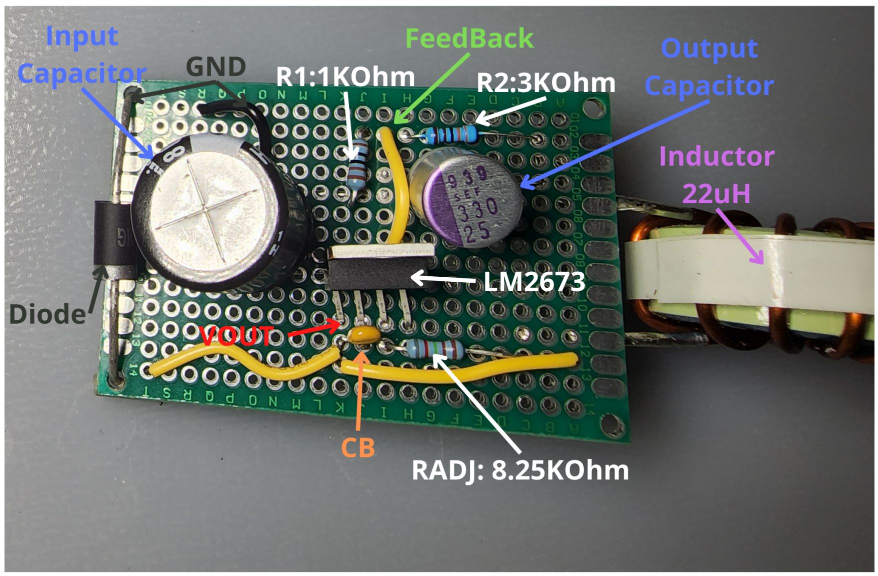


Figure 57 : Temporary perfboard circuit of our buck converter in figure 43

Figure 58 below successfully demonstrates the input and output voltages of our buck converter. In the next section we perform further testing of the other components using an oscilloscope to ensure our buck converter works as intended.

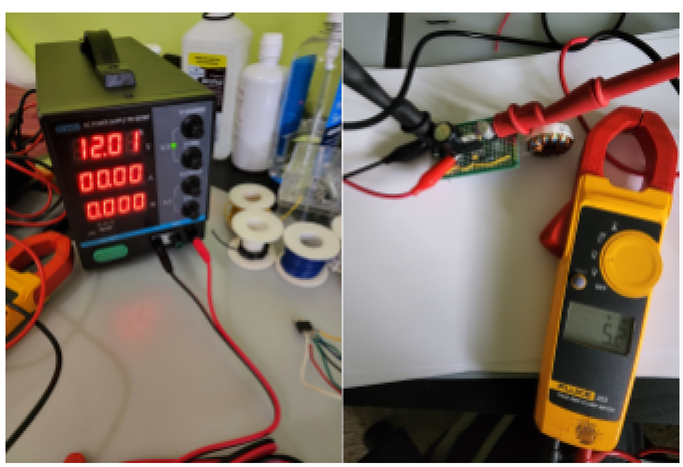


Figure 58: Input vs Output Voltage of our Buck Converter in figure 55

8.3.3 Oscilloscope Testing

Using the Digilent Analog Discovery 2 Kit provided by the University Of Central Florida, we were able to do more in depth testing of our buck converter. We first connected a 12 volt power source to the buck converter with no load then connected the oscilloscope to the input power supply as shown in *figure 59* below.

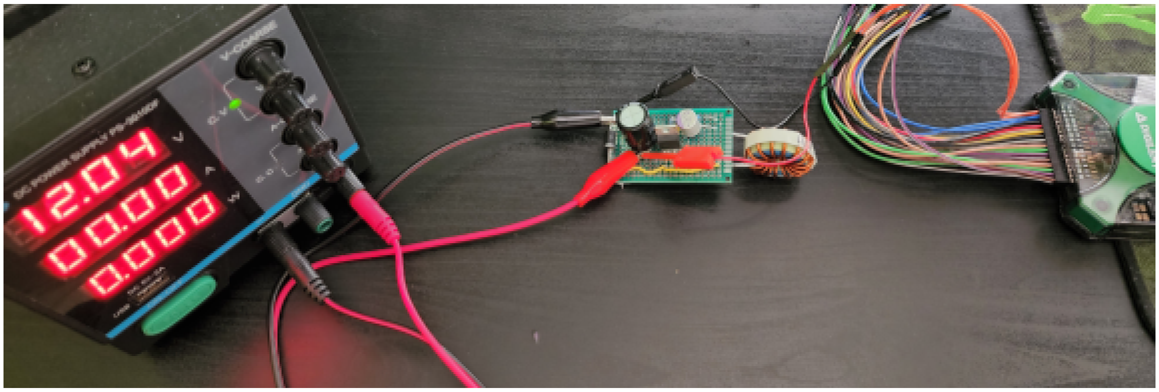


Figure 59 : Connecting the Power Supply to the Analog Discovery 2

Using the Waveform software provided by Diligent we measured the voltage overtime of the input power supply and our results are shown in *figure 60* below. Upon further inspection of the voltage plot we can confirm that the power supply provides clean and stable 12 volts.

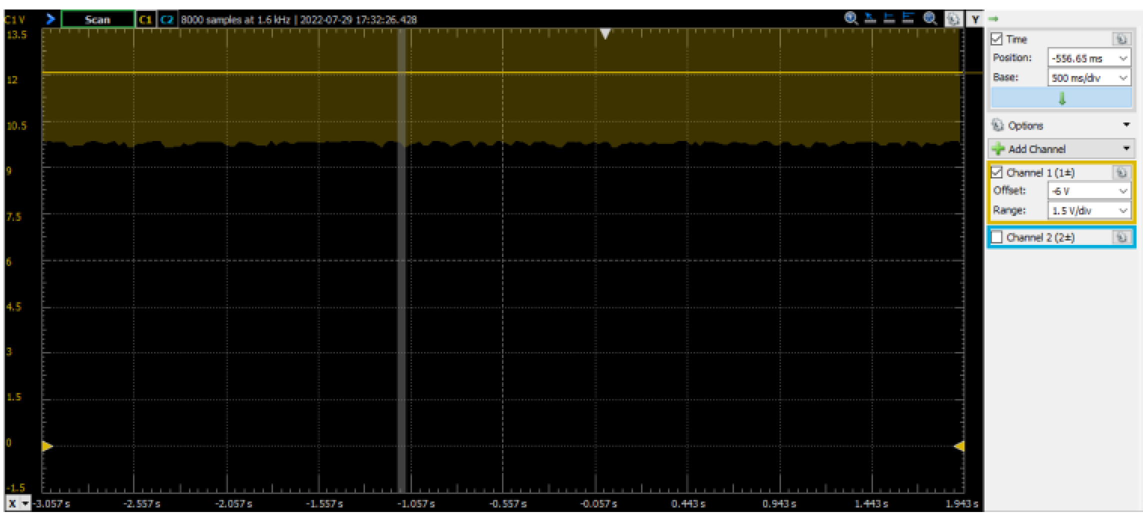


Figure 60 : Power supply input voltage

We then connected the Analog Discovery 2 oscilloscope to the output terminals of the buck converter, which are the leads of the output capacitor, as shown in *figure 61* below.

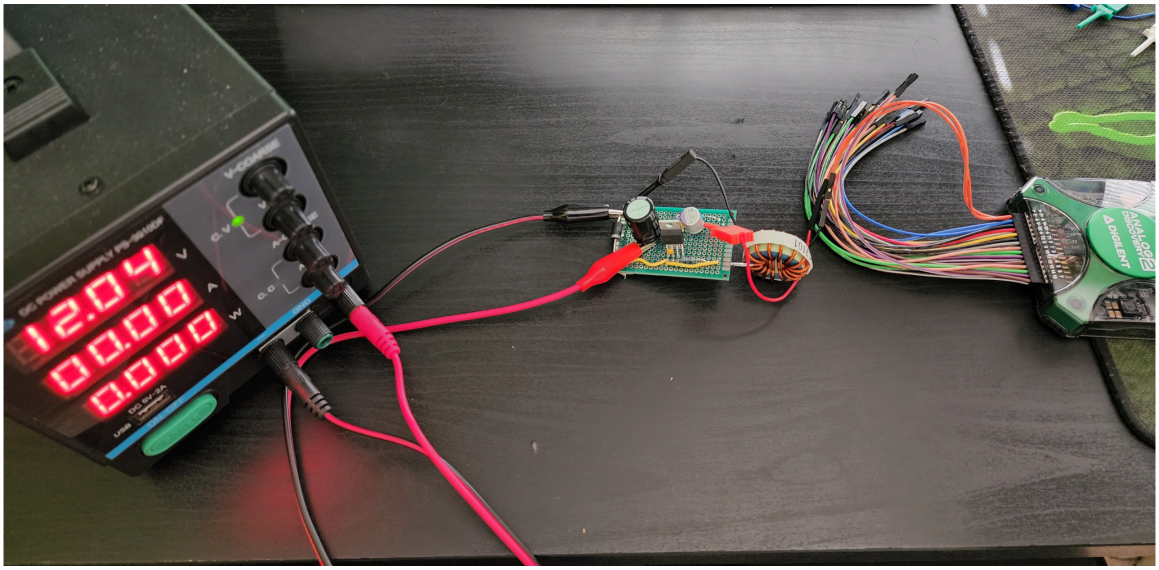


Figure 61 : Connecting the Buck Converter to the Analog Discovery 2

After connecting the oscilloscope to the output of the buck converter we used the Waveform software provided by Diligent to measure the voltage overtime and our results are shown in *figure 62* below. Upon further inspection of the voltage plot, we can confirm that the buck converter provides clean and stable 5 volts under no load.

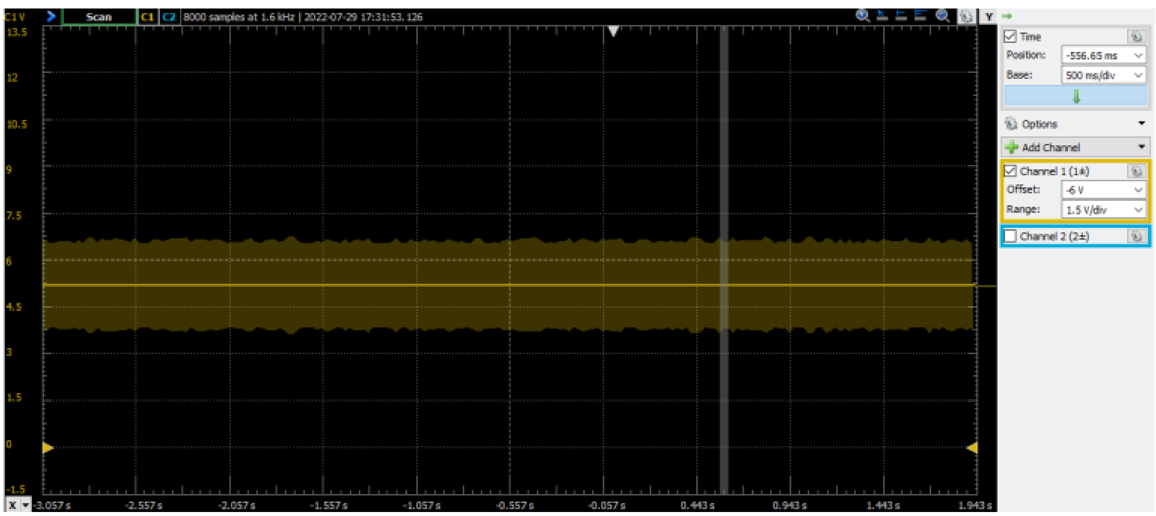


Figure 62 : Buck Converter Voltage Output

In order to determine if our buck converter works properly we need to apply a load to our buck converter. To stress test the load we connected our buck converter to the output of our ESP32, which ran a simple LED blinking sketch as shown in *figure 63* below.

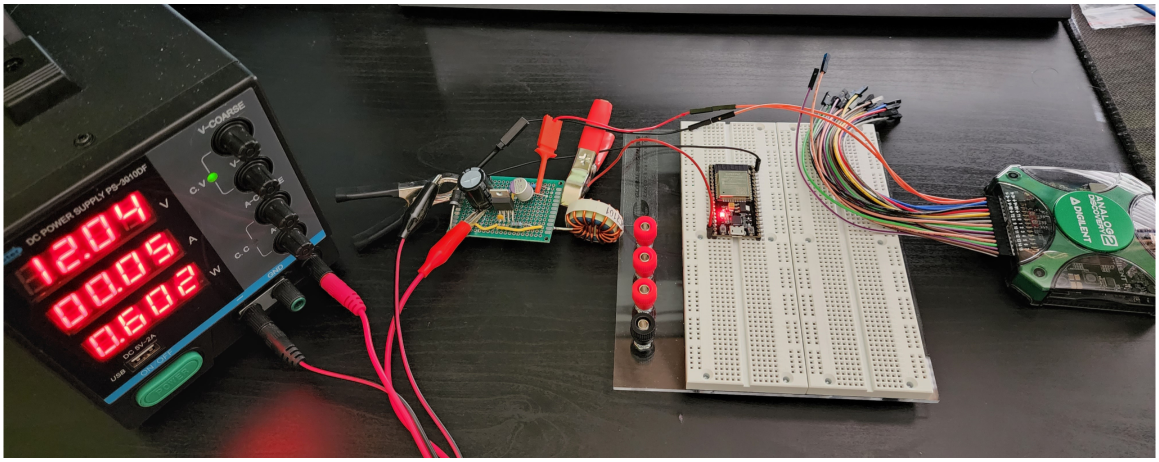


Figure 63 : Connecting the ESP32 and Oscilloscope to the Buck Converter

We then connected the oscilloscope to the output of the buck converter as shown in figure 63 above. *Figure 63* above also demonstrates that the ESP32 draws about 50 mA at 12 volts, which is 0.6 Watts of power. We then took the reading of the output voltages as shown in *figure 64* below. Using *figure 64* below, we can confirm that the output of the buck converter still delivers stable 5 volts under load.

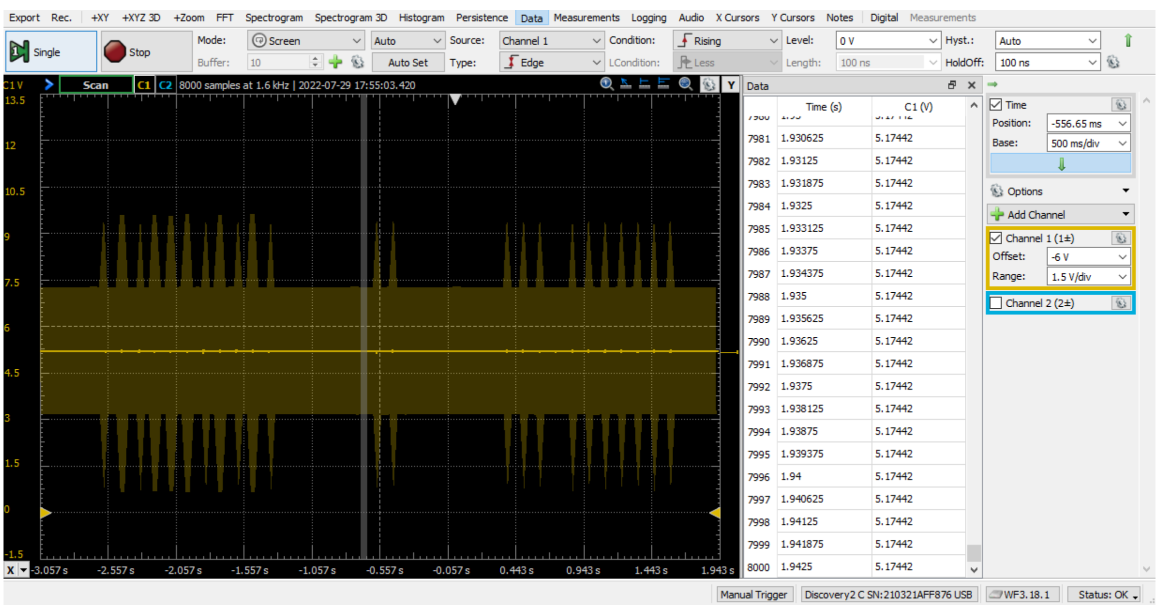


Figure 64 : Output Voltage of Buck Converter under a 50mA load

Although we tested the buck converter under load, we still need to test the buck converters' behavior with different input voltages. To do this we changed the voltage to 9 volts and 14 volts of our input power supply as shown in *figure 65* and *figure 66* below.

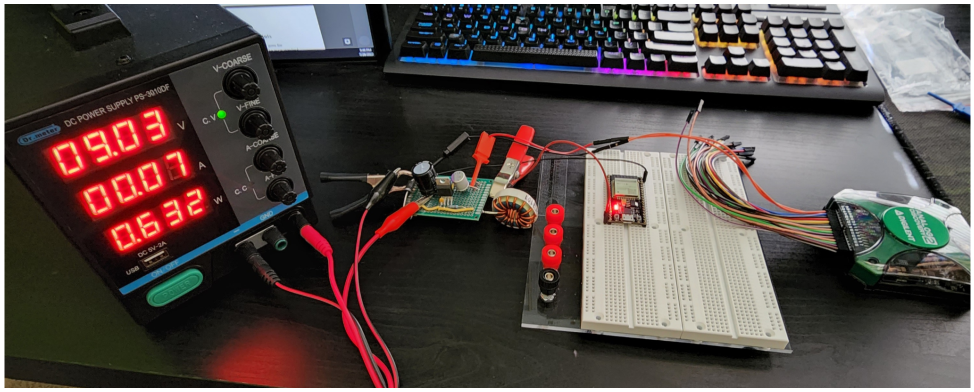


Figure 65 : Buck Converter with 9 Volts Input

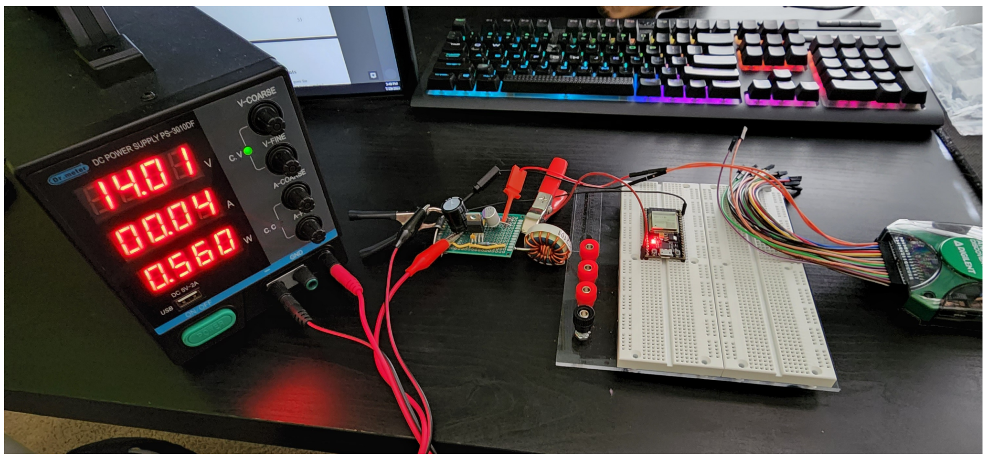


Figure 66 : Buck Converter with 14 Volts Input

We then measured the output voltage of the buck converter using the oscilloscope, and showed our results in *figure 67* below. The graph on the left in *figure 67* below demonstrates an input voltage of 9 volts, and the graph on the right demonstrates an input voltage of 12 volts. After closer inspection we can confirm that the output voltage of the buck converter still holds a stable 5 volts regardless of the input voltage.

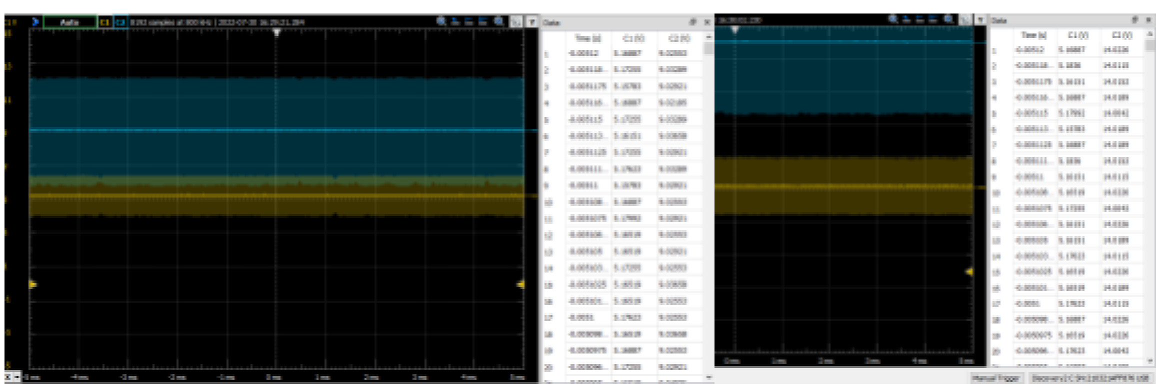


Figure 67 : Power Supply Voltage (Blue), Buck Converter Output Voltage (Yellow)

8.3.4 Hardware Testing Conclusion

After a thorough investigation of our buck converter we can confidently conclude that our buck converter works as intended because it still delivers a stable 5 volts regardless of the load size and input voltage. We would like to note that the input power source to the buck converter should never be below 5 volts because it would not work properly, and for our project this scenario is not possible because we will be using the SPARKOLE 12 volt battery pack as a power source and it should have at least a minimum of 9 volts when completely drained. We would also like to note that we used an AC to DC switching power supply to simulate a battery source for our buck converter because we don't have our SPARKOLE 12 volt battery pack yet. Finally, we would like to note that when we changed the input voltage of the power supply, the current adjusted to still deliver 0.6 watts, which is also another sign that the buck converter works because it adjusts its current when the voltages change to meet the power demand.

8.4 Mechanical Testing

The mechanical testing of our Trashporter device will mostly be down on our mechanical lift. The mechanical lift will consist of two motors that will be used to lift up the smaller trash can from the ground with the use of a lift system. Then we will make a custom robot gripper that will be able to grip onto the smaller trash can tightly with the use of one or two strong servos. The lift system we will be using will lift the entire robot gripper holding the smaller trash can in the air, the "wrist" of the robot gripper will do the rest. The "wrist" of the robot gripper will be one or two strong servos that will flip the gripper along with the smaller trash can to empty the contents of it into the main bin which is attached to the main body of the device. All of our testing we will do in this section will be done to make sure that our mechanical lift is able to operate correctly and efficiently.

8.4.1 Mechanical lift testing

We have to ensure that our robot device is able to lift up the smaller trash can and dump the contents into itself in the bin efficiently. To do this we will be using our robotic lift system that was designed and explained previously. To make sure that the system works correctly and does not possibly break in the future we will put it under rigorous testing as well. The main point of these tests is to make sure that one it's strong enough to lift a smaller trash can, to make sure the movement is fluid and smooth, and that the lift is able to complete the main task which is lifting the trashcan and emptying the contents of it into the device's bin.

The system will have a two-joint design. The main joint will be like a shoulder joint on a human arm which will be powered by two motors on each side of the device. Then the other joint component will act as wrists that will bend back and empty the contents into the bin attached to the device. We will also have one or two servos that are strong to control the robot claw which will grip onto the smaller trash can before it physically lifts it.

To make sure the arm is mechanically sound we will simply do multiple movements of where the robot has to pick up the small trash can at different angles. After that, we will repeatedly move the arm up and down to make sure no components can possibly break while demonstrating our demo to the professors. After that, we will start testing the strength of the mechanical lift/arm system by starting with a lightweight smaller trash can and increasing the load to continue the strength test. We will first start with an empty basket, then we will fill our smaller trash can with lightweight items such as paper and plastic wrappers. Our device does not have to be that strong, but knowing the weight limit of the mechanical lift would be beneficial for presenting our specifications to the professor and for creating a list of specifications for the device as a whole.

The overall objective of all these tests is to make sure that the mechanical lift goes through each step of movement which is: gripping onto the smaller trash can, lifting the smaller trash can up, then flipping the “wrist” which will be servos attached the robotic claw that will flip the claw to be able to empty the contents of the smaller trash can into the main bin of the device.

8.5 Project Operation

The Trashporter robot will search for little trash bins laying around the room and pick them up individually to empty them into the big trash can on the robot. After the actions are finished, the Trashporter robot will send a notification to the software application on the phone. This device should be used within households only to help collect trash from multiple trash bins. It will reduce the time and alleviate stress of worrying about getting trash everywhere in the house during fast-moving times of peoples’ daily lives in modern days where they can focus on more important events. In this section, we will look into the safety precautions for the process of when we are using the Trashporter robot. We will produce a manual for how to operate the trash bin pick-up robot. There will also be a few troubleshooting tips in case some issues occur.

8.5.1 Safety Precautions

In general, the Trashporter robot is a safe device to operate at home with its specified application. However, to avoid potential safety risks, please read the following safety precautions below before turning on or using the Trashporter robot.

1. Keep Trashporter robot away from water or any liquid to prevent any electrical damage to the robot.
2. Make sure no wires are visible to be seen and covered with insulation to prevent any electrical shock.
3. Do not throw hot ashes into little trash bins.
4. Wear protective gloves when handling malfunctioning Trashporter robot.
5. Power off the Trashporter robot when it is not functioning correctly.
6. Do not climb into the Trashporter robot or sit on it.

7. Do not let children make direct contact with the device when it is in operation to prevent injuries.
8. Do not let children touch Trashporter robot with wet hands.
9. Under the circumstance to take apart the Trashporter robot, make sure the device is turned off .
10. If the Trashporter robot is broken or damaged in any form, stop using it.

The Trashporter robot can be dangerous, especially to toddlers and children because of potential electrical shocks and its dimensions to knock over objects. However, it is achievable to prevent the harms and lessen the hazards while using this Trashporter robot by reading the foreseeable safety measures mentioned above.

8.5.2 General Information

The Trashporter robot is a useful tool that has many capabilities provided in a list as shown below:

1. Collects trash from small trash bins in the room.
2. Can map the room through the PixyCam.
3. Detects the weight of the trash in the big trash can on the robot.
4. Detects and alerts when the big trash can is full.
5. Create alerts when the battery is about to die.
6. Transfer data from the Trashporter to mobile devices through Wi-fi connection.
7. Displays the weight software application on the phone.
8. Displays notification when the big trash can is full on the phone application.

8.5.3 Using the Trashporter

1. Read the safety precautions above before using the Trashporter robot to reduce risks and safety concerns.
2. Download the Blynk application on the mobile device to control the Trashporter robot.
3. Set up the room with one or multiple small trash bins.
4. Open the Blynk application and connect to the Wi-fi.
5. To turn on the Trashporter robot, go to your mobile device and press the power button in the Blynk software.
6. (Optional) Start using the device manually
7. Select the “Start” button
8. The device will start searching for small trash cans with the Pixy camera.
9. The Trashporter robot will grab onto the small trash can it finds, lift it up to a certain height, and dump the trash from the small trash bin to the bigger trash can on the Trashporter robot.
10. It will send a notification to the mobile device indicating that the trash is collected.

11. Then, it will place the small trash can back down and send a notification to the mobile device saying the task is completed.
12. The Trashporter robot will turn off automatically to save energy.

8.5.4 Troubleshooting Tips

When you run into problems using the Trashporter robot, look over the following troubleshooting tips to help diagnose the issues and solve the problems presented:

Problem	Solution
1. The device won't turn on.	• Change the batteries. • Check the buck converter. • Check for loose wires and connections.
2. The output of the device is not correct.	• Make sure the big trash bin on the robot aligned correctly and not mispositioned.
3. The Trashporter robot won't work.	• Press and hold the power button for 15 seconds to reset the device and turn it back on. • Remove the battery for 5 minutes and reconnect it.
4. The wheels are moving in an unusual way.	• Clean any dirt and debris lodged around the wheel or the axel.
5. Sensors are not responding correctly.	• Make sure the ultrasonic distance sensors are not blocked by any objects on the Trashporter robot. • Replace the sensors.
6. The Trashporter robot cannot connect to the Wi-fi.	• Reboot the device by pressing and holding the power button for 15 seconds and turn it back on. • Check your Wi-fi connection.
7. The phone will not connect to the robot through Wi-fi.	• Turn the Trashporter off for 15 seconds and turn it back on. • Turn off the Wifi-connection on the phone and reconnect it.

Table 16: Troubleshooting Tips

8.6 Blynk Mobile Application Integration

The Blynk app will originally act as a way we will test if our microcontroller and components are communicating with each other and working in proper order. Once we establish that all the connections on our microcontroller are correct we will be able to completely control the device manually and also be able to receive notifications from our device such as when trash is collected or when the bin attached to our trashporter device is full using a sensor.

9. Administrative Contents

In this section, we will look over the description and responsibilities of the team members, view the past timeline and plan the future timeline, sort the budget of buying parts and components for this project, choose our PCB vendor, and look forward into project roles on building the robot in Senior Design 2.

9.1 Team Description and Delegation

This team has three computer engineers and one electrical engineer. This is a good experience to work with engineers of different disciplines, just like in the workforce. It is a good way to develop collaboration and teamwork skills and interact with people of different backgrounds. Having a diversity of backgrounds will be beneficial to us as some of us will be stronger in specific areas compared to others. This will let us be able to work easier as a team with specific roles.

Jourdan Callahan is majoring in computer engineering. He has quite a bit of experience on the software side as he does work a full-time job for a software company. He has experience programming in Java, C, C#, Javascript, PHP, and Python.

Andy Kuang is majoring in computer engineering. He has done multiple projects throughout his learning career that includes different types of microcontrollers that use parts like sensors, displays, and motors. Software knowledge includes being on a project team for software engineering that implemented AR technology, which he contributed to the backend and database. Some programming languages that he knows are C, PHP, SQL, Erlong, Haskell, Java, MATLAB, and Python.

Dhanesh Singh is majoring in computer engineering. He has experience in programming in C, Java, C#, Python, and MIPs. He has worked on Arduinos before and has experience with working with other electronics. He has a part-time job as an IT and has handled computer repairs. He also has experience with soldering and PCB manufacturing and he can apply those skills to this project.

AnnaBelinda Zhou is majoring in electrical and computer engineering. She has done programming in C, Java, python, and MIPs. She has worked with 3-D printers and reflow ovens that could help with building the arms of the robot and creating our PCB. She has soldering experience that can be put to work. She has some art skills that could help decorate the robot as well.

As previously mentioned, Jourdan and Andy will work on the software side of the project building the codes for the application while Dhanesh and Anna will implement the hardware side of the project by putting the parts of the robot physically together and interface with the software side.

9.2 Project Milestones

The table below will look at our progress of achievement and see how far we have reached, as well as the timeline planned for tasks in Senior Design 2. It summarizes the plan created mainly for the semester of Senior Design 1 that goes a little over to Senior Design 2. We have spread all the tasks equally among the group members. We also decided that Dhanesh and Anna will mainly work with the hardware side while Jourdan and Andy will focus on the software side of building this robot.

Number	Task	Start	End	Status	Responsible
Senior Design 1					
1	Project Ideas	5/9/2022	5/20/2022	Completed	Group 7
2	Project Selection and Role Assignments	5/9/2022	6/3/2022	Completed	Group 7
	Project Report				
3	Initial Document - Divide and Conquer	5/30/2022	6/3/2022	Completed	Group 7
4	Table of Contents and First Draft	5/30/2022	7/8/2022	Completed	Group 7
5	Final Document	6/6/2022	8/2/2022	In Progress	Group 7
	Research, Documentation, and Design				
6	Map Construction	6/6/2022	7/7/2022	Researching	Jourdan
7	Localization and Navigation Module	6/13/2022	7/7/2022	Researching	Andy
8	IMU	6/13/2022	6/27/2022	Researching	Jourdan
9	Object Detection Sensors	6/6/2022	6/30/2022	Completed	Anna
10	Weight Sensors	6/20/2022	7/15/2022	Completed	Anna
11	Height Sensors	7/11/2022	7/29/2022	Researching	Anna

12	Camera Detection	6/13/2022	6/30/2022	Completed	Jourdan
13	Motor Driver	6/6/2022	6/15/2022	Completed	Dhanesh
14	Robotic Arm	6/6/2022	6/30/2022	Researching	Andy
15	Lift System	7/4/2022	7/14/2022	Completed	Andy
16	Power Supply	6/13/2022	7/4/2022	Completed	Dhanesh
17	Schematics	6/13/2022	7/25/2022	Researching	Dhanesh
18	Microcontroller	6/20/2022	7/25/2022	Completed	Dhanesh
19	PCB Layout	6/27/2022	7/25/2022	Completed	Dhanesh
20	List of Materials and Parts	8/1/2022	8/19/2022	Researching	Group 7
21	Order and Test Parts	7/18/2022	8/19/2022	Researching	Group 7
Senior Design II					
22	Build Prototype	8/15/2022	9/2/2022	In Progress	Group 7
23	Testing and Redesign	TBA	TBA		Group 7
24	Finalize Prototype	TBA	TBA		Group 7
25	Peer Presentation	TBA	TBA		Group 7
26	Final Report	TBA	TBA		Group 7
27	Final Presentation	TBA	TBA		Group 7

Table 17: Project Milestone

9.3 Material List and Budget

This project is fully funded by the students. Thus, we are looking at the cheapest component parts while they also are compatible with what we need for the robotic design. Overall, each of us will contribute a maximum of $200 towards this project, totalling $800 for the entire project. It will be important for us to take an inventory of the items we intend to purchase because of the current high demand and short supply. This will make the price vary at any given moment within the construction process, thus the earlier we order, the better it will be in the long run. We will ensure to keep track of the amount being spent on each part and who is purchasing the item as we can evenly distribute money for the entire project, making it fair amongst the members.

Parts	**Quantity**	**Cost per part ($)**	**Link**
12V Stepper Motor	2	$14	12 V Stepper Motor Adafruit
12V Stepper Driver	4	$2.87	12 V Stepper Motor Driver
12V Battery	1	$36	SPARKOLE 12V Battery Pack
MCU	1	Free $12	Already Owned NodeMCU ESP32
Chassis and Bin		$100-250	TBD
Ultrasonic Sensor	4	Free $3	Already Owned Arduino Sonar Sensor
Weight Sensor	1	$8	Load Cell Amazon
Speaker	1	$10	TBD
Audio Amplifier	1	$10	TBD
Pixy 2 Camera	1	$69	PixCam2 Amazon
Fixed Inductors 22uH	1 : 22uH	$4.54	Inductor Mouser
Capacitor	1 : 330uF 25V 1 : 680uF 63V 1 : 0.01uF 50V	$2.88 $2.65 $0.27	330uF Capacitor Mouser 680uF Capacitor Mouser 0.01uF Capacitor Mouser
Metal Film Resistors	1 : 3.16KOhms 1 : 1KOhms 1: 8.25KOhms	$0.15 $0.11 $0.11	3.16kOhm Resistor Mouser
Schottky Diodes	1 : 3 A 40 Volt	$0.47	Diode Mouser
Switching Voltage Regulators	1 : LM2673 ADJ	$5.46	LM2673 Mouser
PCB	1-5	$10-60$	JLCPCB
Total		$380	

Table 18: Material List and Cost

9.4 Supplier

During our research, we have noticed that much of our supply for the Trashporter will be purchased from Amazon, Ebay, and Mouser Electronics. We do have some parts already in our possession before the beginning of this project and we won't need to purchase some items like the ESP32 and sonar sensors. We have already purchased much of our electronic components for the buck converter from Mouser for testing. Ebay offers parts that are about ¾ cheaper than Amazon, however Amazon offers two days free shipping with Prime, and the majority of the group have Amazon prime. Depending on our situation, we need to determine when and where to buy our products. We can purchase items from Amazon for the fast shipping times, but it will be more expensive. On the other hand, we can purchase our items from Ebay to save our budget but it may take weeks to arrive, and sometimes Amazon doesn't have the item we need in stock, while Ebay almost always has the item we need. We do plan on purchasing more of our components from the sellers mentioned earlier but it's also possible that we may purchase products from other sellers in the future due to changes in inventory, price and shipping.

9.5 PCB Vendor

Choosing a PCB vendor to fabricate our PCB design is important because we need our PCB to be affordable and satisfactory. The vendor must have a highly reliable reputation for producing high quality products in a timely manner at an affordable cost. To satisfy our requirements we decided to work with JLCPCB to fabricate our PCB. We believe JLCPCB is a reputable company because we have experience with JLCPCB, and JLCPCB is a well known provider in the electronic hobbyist scene.

The prices for fabricating PCBs from JLCPCB are as follows:

- 1 & 2 Layer PCB - $2
    - Includes 5 boards
    - 100x100mm
    - 20$ 2-4 Day Shipping
    - $3 10-18 Day Shipping
    - FR-4 or Aluminum Base Material
    - Test offered for Industrial, Aerospace or Medical purposes boards
    - PCB Thickness from 0.4 to 2.0 mm
    - Colors offered in Green, Purple, Red, Yellow, Blue, White and Black
    - Offered lead free RoHS compliant surface finish
    - Copper Weight traces offered in 1oz or 2oz
    - Included trace testing to test for short-circuits or open-circuits
    - Free Electronic Component assembly to new customers
    - Offered SMT soldering stencil

Assembling the components on the PCB will require us to solder the joints of the electrical components to the PCB according to the schematic. We will place the correct components on the PCB and then hand solder them with a soldering iron because we have personal experience with soldering from personal projects and other educational activities. The pinouts from the PCB will be connected via 2.54mm cables that we will purchase from other vendors. These cables will connect the PCB to the other components in our system such as the stepper motor driver, and PixyCam 2.

9.6 Facilities and Equipment

At the University of Central Florida, there are plenty of resources available to us students to be successful in this project. This includes things like the library, the Senior Design Lab, and professors just to list a few. Places like the library give us a place to meet up and brainstorm ideas as we work towards completing the project. This place also allows us to rent things like laptops and tablets for a period to have a portable way for us to do research and documentation. There are many floors to the library for each function we would need it for, like having an open discussion meeting can be done on the lowest floor because sound would not be an issue in this place. If silence is needed, we could travel to the top floors where it is strictly quiet study or a place to work alone at. Working in the engineering labs has many things for students to use like oscilloscopes, soldering irons, computers, engineering applications, and PCB boards that students have access to. This is a great place to work at if students do not have this equipment themselves and is open to them as much as needed. Lastly, we can ask for the support from professors during office hours or a scheduled time. This is a huge resource because they are professionals in this field and could give a ton of advice which saves time and possibly money when thinking of how to tackle a problem. They can guide us through these projects and advise what would be the best solution to our problems compared to trying to face it alone, as this will not be as efficient because for many students this is the first time, they have gone through something at this scale. Taking advantage of all these resources can make our time easier and cost effective. We would not need to see out other places with these materials or must invest in the equipment needed to complete the project.

The other facilities that will be used during the time of the project will be at groupmates homes or personal labs. Because of our engineering background, a few of our group members are hobbyist when it comes to engineering which will be very useful knowledge and resource wise, this saves a lot of upfront cost because we will have some of the parts required to construct the device and save time as some of us have experience making parts of the project from previous classes or in our own time. In a previous class Junior Design, we were required to build a range finder using an ultrasonic sensor which would have an output sent to a LCD display, all powered by a microcontroller, meaning that we have all these components already which can be repurposed into our new project. We could use things like the ultrasonic sensor to gauge the distance between the top of the trash bin and the number of items inside the bin to observe how much trash the bin is filled up to, this will then send a notification letting the owner know that the bin is full and needs to be disposed of. Other opportunities that we have in our hands is labs that we have access to based on job opportunities, this has things like 3D printers, computers, modeling software, and electronic parts that we could use. Having access to our own printer is going to be very convenient for us because though the University does have one that is provided for us, there would have to be a request to build what is needed, and since there are plenty of other students working on this project as well that would want to use the printer, this would have us in a queue to get access to the materials we need to print costing time in the process. Planning out what we have access to in this stage is important as it shows us the options we have and how we should execute our plans of constructing the device as to figuring out the options we have available to us when we are currently working on it because there will be many other things that we would have to be thinking about when focused on the project.

The equipment that we will be using are an Oscilloscope, soldering iron, electrical power supply, 3D printer, multimeter, Autodesk software, and computers just to name a few. All these things are available to us in the Engineering lab or something that we have in our own personal space. It is important to get an accurate measurement of the specifications when building the device because that determines how successful the device will operate, which these tools will help guide what is needed.

The Oscilloscope will aid us in measuring electronic signals in the form of waveforms to accurately display the voltage as a function of time. This lets us know if the device is outputting too much or too little voltage which could indicate that it is not being regulated correctly for the circuit to function properly.

Figure 68: Digital Oscilloscope

We have access to a digital oscilloscope at the University's lab as well as one of our team member's personal device, this gives us easy access to the tool to use to measure current and voltage levels in our device without. We have experience using the device because there have been previous classes that this was required to use to measure the outcome of the voltages, currents, and resistance on our circuit boards when constructing different types of circuits.

A soldering iron is used to melt metal alloy to meld a piece of electrical equipment to a board, it provides an electrical supply of current to heat up the iron hot enough where it can make the metal malleable. We need this to either mount components onto our custom-built PCB or edit parts of the board.

Figure 69: Soldering Iron

We have access to a soldering iron also in the University's lab as well as a personal owned device. The experience we have with soldering is from a previous class where some students had to mount components to their board, as well as personal experience doing side projects.

To ensure our project will be a success, we are going to need proper equipment and a safe environment to help construct the Trashporter. For senior design students the University of Central Florida provides free and open access to their Senior Design Laboratory. The lab is open during business and non-business hours, and weekends. Senior design students can use their identification card to gain access to the lab, which is located in the Engineering 1 building, room 456. According the UCF ECE website "the lab provides a workshop space with instrumentation, equipment, and software for Senior Design students to design and build their Senior Design project." (UCF ECE) The lab comes equipped with Tektronixs' oscilloscopes, function generators, and digital multimeters, as well as Keithley triple-channel power supplies, Dell Precision 3420 computers, SMD reflow workstations, soldering stations, desoldering stations, and digital microscopes. UCF also provided us with the Digilent Analog Discovery 2 Oscilloscope and Function Generator kit to use at home. The UCF SD lab will be a great workspace as it provides the necessary tools and environment to safely construct our project in the near future.

Dhanesh also owns his own set of power tools such as power drills, a reciprocating saw, an oscillating saw, a sander, a compressor, and a couple of other power tools. He also owns the standard set of manual tools such as hammers, various tipped screwdrivers, and a handheld saw, etc. He also owns tons of passive electronic components such as capacitors, resistors, diodes, LEDs, and wires, etc. He also owns several tools that can aid with the electrical aspect of the project such as breadboards, perfboards, a soldering station, a 300 watt AC to DC switching power supply, and a multimeter. Dhanesh's home is also another great option for us to safely construct our project in the near future because not only does he provide the majority of the tools that UCF SD lab provides, but he has additional tools as well, not to mention a possible place to sleep for those long overtime hours.

9.7 Look Forward

We look forward to building this project physically soon. We do finally get some hands-on experience with touching machines and building something from scratch. We do not look forward too much to the troubleshooting part of this project. However, that is the process of learning and gaining experience and knowledge. Hopefully after completing this project successfully, it will be able to move on to get modified to have better versions and features. We will incorporate skills that we know that whatever each team member is strong in. Then we will also incorporate new skills that we will learn during the development of this Trashporter device. We do expect this to be challenging and we believe with our research and planning defined in this document, that we will be able to successfully complete this project with our specified requirements and functions that we defined.

10. Project Summary and Conclusion

Overall, the point of this project is to make a robot that is autonomous. Our trashporter device will be able to locate and pick up a smaller trash can to be able to empty its contents into the main bin connected to the main body of the device. We believe that this device can be replicated and manufactured easily as we are using components that can be found easily online besides our PCB board and some other hardware components. The main parts of the device will be the main bin, which will be used to hold all of the trash contents, the main base chassis, which will hold our electrical and hardware components along with being attached to our four motors and motor drivers, and the last component of this device will be our mechanical lift system which will be a robot claw attached to a servo which will be used to grip onto the smaller trash can, and this assembly will be attached to a slider lift which will be able to move up and down. The assembly will be able to rotate back to be able to flip the smaller trash can along with the robotic claw to be able to empty the smaller trash can contents into the main bin.

We have encountered some problems while coming up with our development plan while researching. We were originally going to use an Arduino microcontroller but decided to try to use a different one as one of our team members already owns a microcontroller. We also brainstormed a lot of ideas on how we should make the mechanical lift for this device. We wanted the mechanical lift to be simple enough so that not many things could possibly break while in use. We believe that if the mechanical lift had many points of failure it would not be a good idea to choose that option as we want this device to be working when we demo it. We also wanted to make sure that the option we chose was simple enough that it would fall within our budget range and our size constraint as well. Another problem we had was figuring out the sensors that will be used for the trashporter.

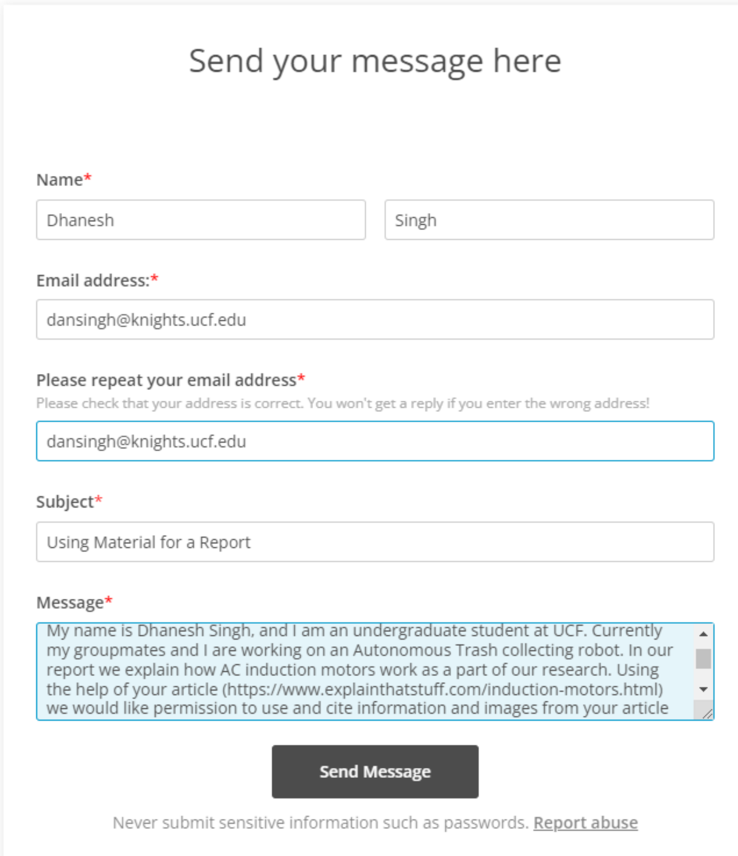
For our autonomous device, we will have two sensors. The first sensor will be used to detect the weight of the trash contents in the main bin. We plan on having this sensor send a notification to our mobile device if the trash bin is full. The second sensor that we will use is a sensor for detecting walls or obstacles while the robot is running the autonomous algorithm. We went over many different types of sensors for the weight sensor and decided on the one discussed above. We had the same process for the object detection sensors and decided to use the same sensors that were used in our rangefinder projects from Junior design. When researching both of these sensors, we went through many different types. For the weight sensor specifically, we wanted to physically weigh the main bin of the device, however, we decided that it would be easier on us if we approached it a different way. We decided that having a regular sensor would be a smart decision and we could use it in two different ways. The first way would be to aim the sensor into the bin and measure the distance between the trash contents and the sensor. The second way would be having a sensor detect when the main bin is full by having a sensor pointed horizontally acting like a “fill line”.

Overall, we believe that this project will let us use the skills specific to our strengths for each team member as well as learn new skills for the development of the device. We understand that this could be a difficult task to get this trashporter device to work, however, we believe with this document and our extensive research that we will be able to tackle this project in a timely manner. We will also gain valuable experience that any future employers would love to see as we will show that we are able to learn new things on the fly and also work well in a team setting. We look forward to developing the trashporter device. We believe that it will be a very interesting way to show off robot vision and autonomous functionality while being able to collect trash around the house!

11. Appendices

This section is used to show the bibliography containing all the material that was referenced throughout this paper. Also a section for permissions follows to show permission for each outsource image used.

11.1 Copyright Permissions


Send your message here

Name*

Dhanesh

Singh

Email address:*

dansingh@knights.ucf.edu

Please repeat your email address*

Please check that your address is correct. You won't get a reply if you enter the wrong address!

dansingh@knights.ucf.edu

Subject*

Using Material for a Report

Message*

My name is Dhanesh Singh, and I am an undergraduate student at UCF. Currently my groupmates and I are working on an Autonomous Trash collecting robot. In our report we explain how AC induction motors work as a part of our research. Using the help of your article (https://www.explainthatstuff.com/induction-motors.html) we would like permission to use and cite information and images from your article

Send Message

Never submit sensitive information such as passwords. Report abuse

Re: Using Material for a Report

ET Explain that stuff <ets7@explainthatstuff.c
To Dhanesh Singh
4:03 PM

You replied to this message on 7/31/2022 4:13 PM.
We removed extra line breaks from this message.

Hi Dhanesh,

Yes, of course you're very welcome to do that. If you let me know exactly which image or images you're interested in, I might be able to supply bigger and clearer versions than the ones on the website (if it's one of the animated images, I can let you have the frames of the animation as separate images - if you don't know how to extract them for yourself).

Best wishes,

Chris.

Permission to Use Images from Article

DS Dhanesh Singh
To CVEL-L@clemson.edu
3:14 PM

Hi,

My name is Dhanesh Singh and I am an Undergraduate student at UCF. My group and I are working on an Autonomous trash collecting robot. In our report we explain how a brushed DC motor works as part of our research. We would like permission to use and cite your article and image of the internals of a brushed DC motor for our report.

Thanks,
Dhanesh Singh

HowTo MECHATRONICS ARDUINO PROJECTS TUTORIALS HOW IT W

The information contained in this website is for general information purposes only. The information is provided by How to Mechatronics and while we endeavor to keep the information up to date and correct, we make no representations or warranties of any kind, express or implied, about the completeness, accuracy, reliability, suitability or availability with respect to the website or the information, products, services, or related graphics contained on the website for any purpose. Any reliance you place on such information is therefore strictly at your own risk. Every material on this website may not be the original idea of the webmaster and may have been borrowed ideas and/or submitted by visitors.

The information provided on the HowToMchatronics.com may be used, copied, remix, transform, build upon the material and distributed for any purposes only if provided appropriate credit to the author and link to the original article.

Your name*

Dhanesh Singh

Your email address*

dansingh@knights.ucf.edu

Subject*

Permission to Use Material

Message*

Hi,

My name is Dhanesh Singh and I am an Undergraduate student at UCF. My group and I are working on an Autonomous trash-collecting robot. In our report, we explain how a stepper DC motor works as part of our research. We would like permission to use and cite your article and image of the internals of the stepper motor (https://circuitdigest.com/tutorial/what-is-stepper-motor-and-how-it-works).

Thanks,
Dhanesh Singh

Contact *

Dhanesh Singh

E-mail *

dansinghcoll@gmail.com

Message Content *

Hi,
My name is Dhanesh Singh and I am an Undergraduate student at UCF. My group and I are working on an Autonomous trash-collecting robot. In our report, we plan to use the NodeMCU ESP32 as well as the LN298 stepper motor driver. We would like permission to use and cite your images of the products as well as the data sheet in our report.

Security Code

2220 | 2220 | Refresh the code

Permission to Use Material

DS Dhanesh Singh
To copyrightcounsel@list.ti.com

3:35 PM

Hi,

My name is Dhanesh Singh and I am an Undergraduate student at UCF. My group and I are working on an Autonomous trash collecting robot. In our report we create a buck converter with the help of the Texas Instruments LM2673 buck converter data sheet. We would like permission to use and cite your datasheet, images, and charts of the two TI linked articles below, to further explain how we used the help of your articles to create the buck converter.

Switching regulator fundamentals (Rev. C) (ti.com)
LM2673 data sheet, product information and support | TI.com

Thanks,
Dhanesh Singh

REQUEST OF INFORMATION

Contact Vidatronic, Inc.

Fill out this form for contacting a *Vidatronic, Inc.* representative.

Your Name:	Dhanesh Singh
Your E-mail address:	dansingh@knights.ucf.edu
Your Company address:	University of Central Florida (Student)
Your Phone Number:	3478815459

Write your message:

Hi,

My name is Dhanesh Singh and I am an Undergraduate student at UCF. My group and I are working on an Autonomous trash-collecting robot. In our report, with the help of your article (https://www.design-reuse.com/articles/42191/low-dropout-ldo-linear-voltage-regulators.html) we explain how a linear voltage regulator works as part of our research. We would like permission to use and cite your images and information as part of our report.

Thanks,
Dhanesh Singh

send

EMAIL *

dansingh@knights.ucf.edu

INQUIRY TYPE *

Editorial/Content question

FIRST NAME *

Dhanesh

COMPANY TYPE *

Academic

LAST NAME *

Singh

PHONE NUMBER

3478815459

COMPANY NAME *

University Of Central Florida

YOUR MESSAGE *(TELL US HOW WE CAN HELP)* *

Hi,

My name is Dhanesh Singh and I am an Undergraduate student at UCF. My group and I are working on Autonomous trash-collecting robot. In our report with the help of your article (https://www.eetimes.com/signal-chain-basics-part-19-exploring-and-understanding-linear-voltage-regulators/) we explained how a linear voltage regulator works. We would like permission to use and cite your images, and information that helped our project development.
Thanks,

COUNTRY *

United States

STATE/PROVINCE *

Florida

Yes, I Would Like To Receive AspenCore Emails About Products, Updates, Offers, And Events. (I Understand I Can Opt-Out At Any Time).

☐ By Submitting This Form, You Agree To The Use Of Your Personal Information In Accordance With The AspenCore Terms Of Use And Privacy Policy.

Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)

This is a human-readable summary of (and not a substitute for) the license. Disclaimer.

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Jourdan Callahan <jourdancallahan8@gmail.com> — Jul 17, 2022, 9:07 AM

to support

Hello,

My name is Jourdan and we plan on using the Pixy2 camera for our school project. I am just reaching out to ask for permission to use some pictures from your website in our Senior Design report.

Thank you!

Charmed Labs Support — Jul 18, 2022, 1:01 PM

to me

That should be fine! Go ahead

Hope your project goes well

Sincerely

ROBOTIC CLAW 1 months ago

Hey joshlunde! I am working on a school project and was wondering if it would be okay to use this photo as a reference? Thank you for sharing your project and having this public for everyone!

Name *

Andy | Kuang

First | Last

Email *

akuang@knights.ucf.edu

Comment or Message *

Hi, my name is Andy Kuang, I am working on a school project and was wondering if it was okay to use of the photos on your robot claw blog - Robotic claw: CAD design and 3D printed, this was a large help to my research. Thank you for sharing!

Submit

Andy Kuang — required

University of Central Florida — required

Your phone number

akuang@knights.ucf.edu — required

Hi, this is Andy Kuang, I am working on a school project and doing research to create a robotic arm, I was wondering if it would be okay to use of of the pictures on the Building V5 Robot Lift Systems blog? Thank you for all the information that you have provided!

Send

AnnaBelinda Sent - Exchange July 28, 2022 at 10:41 AM
Permission to use pictures
To: sales.us@baumer.com

Hello,

My name is Annabelinda Zhou. I'm on a Senior Design project group that will be using ultrasonic sensors. I wanted to ask your permissions to use some of the pictures from your website in our project report.

Thank you,
AnnaBelinda Zhou
(She/Her/Hers)
Email: az999@knights.ucf.edu
University of Central Florida
Bachelor of Science in Electrical & Computer Engineering

AnnaBella Zhou Sent - Google 8:15 PM
Picture Permission
To: futek@futek.com

Hello,

I'm a college student that's doing a senior design project right now. I am writing a project proposal with my senior design group. I would like to ask your permission to use one of the pictures from your website in my report.

Sincerely,
AnnaBelinda Zhou

11.2 Data-Sheets

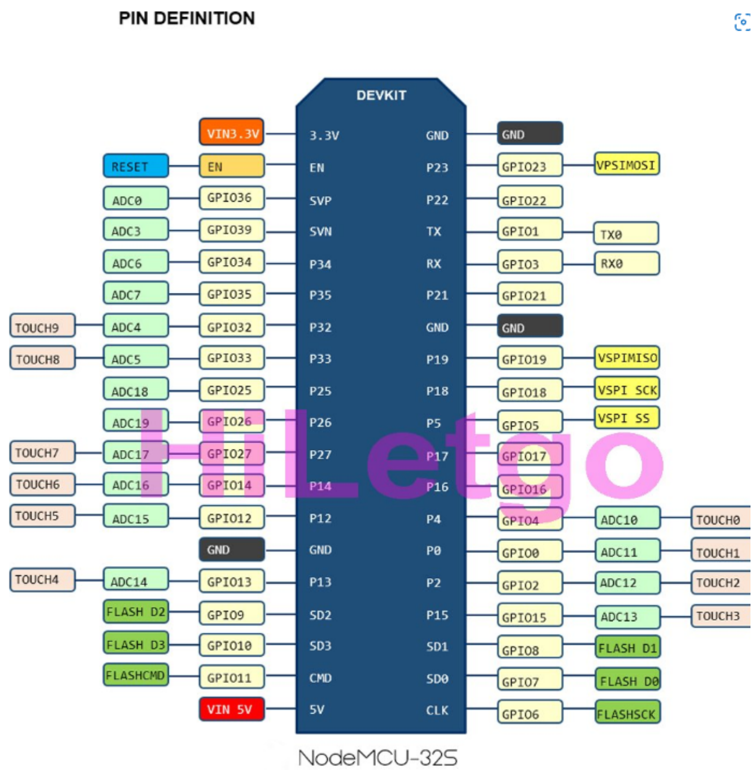


Figure 70: NodeMCU ESP32 Pinout Diagram. Image Courtesy of HiLetGo

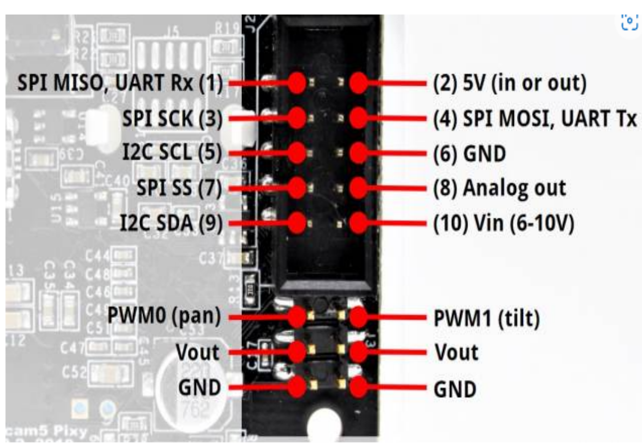


Figure 71: Pixy Cam 2 Pinout Diagram. Image Courtesy of Charmed Labs

11.2 References

1. polepositionmarketing@kellertechnology.com. (2021, December 6). *7 Types of Sensors for Object Detection*. Keller Technology Corporation. Retrieved from https://www.kellertechnology.com/blog/7-types-of-sensors-for-object-detection/
2. Calin, D. (2020, April 15). *Types of sensors for target detection and tracking | Into Robotics*. Into Robotics | ROS, Robots and Stuff. Retrieved from https://www.intorobotics.com/types-sensors-target-detection-tracking/
3. Lab, M. (2017, May 30). *Ultrasonic Sensor working applications and advantages*. Microcontrollers Lab. Retrieved from https://microcontrollerslab.com/ultrasonic-sensor-working-applications-advantages/
4. *Ultrasonic sensors – the most versatile object detection*. (2022). Baumer. Retrieved from https://www.baumer.com/us/en/product-overview/object-detection/ultrasonic-sensors/c/281
5. *What is a weight sensor, what are the different types of sensors and how do they work?* Futek. Retrieved from https://www.futek.com/weight-sensor
6. R. (2020, July 27). *What are Weight Sensors?* Utmel. Retrieved from https://www.utmel.com/blog/categories/sensors/what-are-weight-sensors
7. *Robotic Claw*. (2020, August 8). Arduino Project Hub. Retrieved from https://create.arduino.cc/projecthub/aryamankumar02/robotic-claw-61ed7b
8. Dragos-George, C. (2020, April 18). *Making a Robot Claw | Into Robotics*. Into Robotics | ROS, Robots and Stuff. Retrieved from https://www.intorobotics.com/robot-claw/
9. *Building V5 Robot Lift Systems.* VEXV5. Retrieved from https://kb.vex.com/hc/en-us/articles/360037388692-Building-V5-Robot-Lift-Systems
10. Hani, D. (2021, April 19). *LiDAR vs RADAR: Detection, Tracking, and Imaging*. Wevolver. Retrieved from https://www.wevolver.com/article/lidar-vs-radar-detection-tracking-and-imaging
11. Klein, B. (2008, August 2). *Signal Chain Basics (Part 19): Exploring and understanding linear voltage regulators.* EETimes. Retrieved from https://www.eetimes.com/signal-chain-basics-part-19-exploring-and-understanding-linear-voltage-regulators/
12. Veeravalli, A., & Nolan, S.M. (2022). *Introduction to Low Dropout (LDO) Linear Voltage Regulator.* Design & Reuse. Retrieved from https://www.design-reuse.com/articles/42191/low-dropout-ldo-linear-voltage-regulators.html
13. *LM2673.* Texas Instruments. Retrieved from https://www.ti.com/product/LM2673
14. *L298N Motor Driver controller Board Module Stepper Motor DV Dual H-Bridge*

For Arduino. (2018). HiLetgo. Retrieved from http://hiletgo.com/ProductDetail/1915475.html

15. Castaldo, A. (2021, September). *Switching regulator fundamentals.* Texas Instruments. Retrieved from https://www.ti.com/lit/pdf/snva559?keyMatch=SWITCHING%20REGULATOR%20FUNDAMENTALS
16. Raj, A. (2018, October 1). *What is Stepper Motor and How it Works.* Circuit Digest. https://circuitdigest.com/tutorial/what-is-stepper-motor-and-how-it-works
17. D. *How Brushless Motors and ESC Work.* How To Mechanics. Retrieved from https://howtomechatronics.com/how-it-works/how-brushless-motor-and-esc-work/
18. *Brushed DC Motors.* Automotive Electronics. Retrieved from https://cecas.clemson.edu/cvel/auto/actuators/motors-dc-brushed.html
19. Woodford, C. (2021, June 28). *Induction motors.* Retrieved from https://www.explainthatstuff.com/induction-motors.html
20. *Raspberry Pi Zero 2 W.* Raspberry Pi. Retrieved from https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/
21. *Documentation.* ESPRESSIF Systems. Retrieved from https://www.espressif.com/en/support/documents/technical-documents
22. *Arduino Uno WiFi Rev2.* (2021). Arduino CC. Retrieved from https://store-usa.arduino.cc/products/arduino-uno-wifi-rev2?selectedStore=us
23. *Documentation*. wiki:v2:start [Documentation]. (n.d.). Retrieved from https://docs.pixycam.com/wiki/doku.php?id=wiki%3Av2%3Astart
24. *IEEE SA - IEEE recommended practice for sizing lead-acid batteries for stand-alone photovoltaic (PV) systems*. IEEE Standards Association. (2019). Retrieved from https://standards.ieee.org/ieee/1013/7165/
25. *IEEE SA - IEEE Standard for software and System Test Documentation*. IEEE Standards Association. (2008). Retrieved from https://standards.ieee.org/ieee/829/3787/
26. IEEE Standards Association. (2008). *IEEE SA - IEEE standard for Standard Systemc Language Reference Manual*. IEEE Standards Association. Retrieved from https://standards.ieee.org/ieee/1666/4814/
27. *Senior Design Laboratory.* Department of ECE. Retrieved from https://www.ece.ucf.edu/academic-laboratories/senior-design-laboratory/