FollowBot: A motorized platform to follow the user with collision detection

Department of Electrical Engineering & Computer Science

University of Central Florida

Dr. Lei Wei

Group 6

Adil Ali – Electrical Engineering

David Falter – Computer Engineering

Carlos Gonzalez – Electrical Engineering

Abhinav Sharma – Computer Engineering

Table of Contents

Figure Index

Table Index

1.0 Executive Summary

The FollowBot is highly focused around two key aspects: collision avoidance and following a user of the product. These seem simple enough, but are in fact very hard to implement. While there are plenty of products on the market that do some sort of following, the FollowBot fills a unique niche which we feel has not been occupied yet. The target audience being an airport, we feel that the FollowBot can make a great impact in terms of convenience for many people. The security concerns can be resolved through only allowing rented platforms to users, and Bluetooth pairing makes it very easy for us to maintain users and allow updates on the fly through a mobile app. The technology used is cutting-edge for its domain, and we made sure that in no form will the technology limit the device. The goal is to make the unit as least clunky as possible to be able to provide a smooth demo and presentation to the audience.

With collision detection comes a whole host of different problems, especially with motion and adjusting course correctly. Overall, despite the hindrances we could face, we believe we have chosen the best tools at our disposal and that we have a great understanding of what is to come.The need for a device like this has been present in our minds for a long time. The idea in itself is very simple and not quite original, but the execution makes it unique. A motorized following platform allows for greater use than just a drone with camera capability or a smart suitcase that can follow you from your car to the check in window. This product, however, has greater intentions. We want to provide a platform for developers and users alike to use and apply to different objects. While the primary use may be intended for an airport setting, such a product could have so much use elsewhere that we may even see it change its primary audience. The consumer market could effectively take over due to the vast usage we expect to see for personal use.

Currently on the market one option exists for our target project: a smart suitcase with following capabilities. This is exactly along the lines of what we imagine, but solves all the right problems and creates more. The suitcase adds almost 20 pounds to the weight of an empty suitcase, has a limited battery life, and cannot get through security and must be checked in. This is what we chose to actively assess when choosing this project. If a product is aiming to convenience people, it must do so without losing a tradeoff in the same aspect somewhere else. We want to partner with airports and create revenue while allowing people to have their travel times a little easier and a lot less stressful. All of this, however, is reliant on the product’s effectiveness, which is a result of our time spent in development and choosing the ideal parts for such a venture.

The hardware inside FollowBot is powering the whole product and allowing us to do what we need to do. The motors we chose allow us to be able to avoid collisions instantly and actually adjust course as we need. The battery will provide us with enough power to move around and hold a charge to actually make the product useful, and it being rechargeable allows for multiple uses and repeatability without sacrificing cost. The wheels will be able to traverse through regular flat floors with ease and allow FollowBot to maintain a consistent speed while staying close to the traveler for maximum convenience. Our hardware for Bluetooth trilateration will be very effective due to the power of Bluetooth as it is used today. We will be able to use Bluetooth modules on the device for trilateration as well as pairing to a mobile phone to increase maximum effectiveness in the field providing the best user experience possible, which is key. For collision detection and avoidance, very responsive and effective ultrasonic sensors will be used to detect distance and adjust accordingly when needed, while pinging constantly enough to minimize response time. Under the hood, the products our team has chosen for the PCB (printed circuit board) are top of the line for their various uses.

The microcontrollers are a very important part in controlling all of the various modules on the product, while the various modules do all the functionality. The hardware chosen is all very effective at its job, but not at the cost of ease of development, which causes our software to be one of the most efficient parts of our project.FollowBot has a lot of complexity due to its various features, namely the software on the product itself being a top priority. As discussed previously, the collision detection and avoidance software is systematically detecting objects whether moving or stationary while also managing speed and location of the user. The microcontrollers allow us to retain this effectiveness due to their power and versatility. One of the keystones of our efforts is the mobile app, which will allow users from all over the world with a smartphone to be able to connect to their FollowBots anywhere and use them to make life easier. The mobile app controls two very important things: pairing with the FollowBot and maintaining the location of the user.

This information must be constantly exchanged for our product to be effective, and thus heavily relies on the effectiveness of the software. With efficient code, this can be streamlined extremely well allowing the user to experience a lag-free process. We have chosen the best development tools which appeal to our strengths as programmers, further increasing the time to production and reducing the cost of development allowing us to put our time in perfecting the product. The parts used complement the software tools used very well and allow us to write and use open source code with the wide variety of resources available publicly on the internet.

Our choice to enter the public market for a small scope such as airports is due to the potential we saw in such a matter. We knew it was a part of the industry which could see improvement, and our hardware and software could bring it up a notch. The choices we made regarding our hardware and software alike have aligned us with the best tools available to us to finish this product and allow for massive market potential if scaled properly. Our ability to execute will determine the effectiveness of such a large project, but a working demo will show why all of these choices will reflect nicely on the end product. The FollowBot is a simple yet elegant solution to an everlasting problem of convenience which will always be a desire for people around the world. This, in effect, also solves a great issue for disabilities and is indeed a wise choice for airports around the world.

2.0 Project Narrative and Goals

This project aims towards a solution in industry that is long overdue. The platform being built allows users to be completely hands-free when taking along a suitcase at an airport within a few simple steps. The applications for such a solution are endless: making disabled lives easier, simplifying working with large tools, and even avoid carrying around a cooler at events, just to name a few. By building FollowBot, we aim to make a visually representative demo of a working product that could eventually be scaled into a marketable design. The motivation comes from something which we can all relate – having to carry or roll around a suitcase at an airport. This is not too bad but becomes exponentially worse when having to do something else or having to rush to your next flight while dragging along a heavy suitcase.

The solution we propose aims to squash these issues by creating a platform with versatile applications. The platform will be able to carry or mount a suitcase, a cooler, or any reasonable object while maintaining the ability to follow the user wherever they go. The product will use ultrasonic sensors for triangulating the position of the transmitter regularly as well as collision detection to avoid obstacles along the way. In this project, collision detection is one of the main priorities due to its importance in an actual demo environment. The platform should be able to: follow the user close, avoid hitting all obstacles, keep up at a reasonable speed, and maintain connection and tracking throughout the process. The project should be relatively low cost due to ultrasonic sensors being inexpensive, and the motors for the wheels being our main cost. The idea, however, is that the platform itself should be easy to use and portable. The user should be able to simply pair their transmitter to the receiver on the product and it should be ready to follow at command. Along with this, a portable device would allow for flexibility while travelling or moving whenever needed.

The goals and overall objectives of this project are to deliver a product that can resolve a common inconvenience people run into every day. For people that travel very often, a solution such as this at the airport could mean time saved as well as ability to do other stuff on the go. For this reason, the platform needs to be reliable and portable. The user should never have to lift the platform off the ground, even for charging, ideally. The product will include wheels that can maneuver around busy hallways and keep the objects close for maximum security. The PCB will reside under the platform, enclosed, keeping the top surface flat and aesthetically pleasing. There will be railings around the top to ensure that items kept on top will be safe for the duration of the ride. The ultrasonic sensors connected to the PCB will receive signals from the transmitter constantly updating its location. Other sensors will be used to bounce back against objects to determine whether there is an imminent collision, and if so, the wheels and motors need to be adjusted as needed.

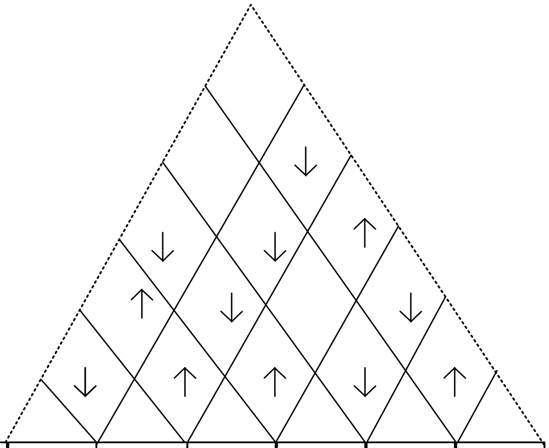
2.1 Project Requirements & Specifications

- Device weight should be less than 25 lbs.
- Should be able to autonomously follow user
- Has collision detection system to avoid obstructions
- Battery powered, allowing it to be recharged
- Has effective range of at least 30 ft.
- Can carry at least 40 lbs.
- Device should be able to fit in a standard sized elevator
  - The width should be less than 36” and the height less than 6’ 8”
  - Length should be less than 40”
- Should be able to carry at least two standard sized suitcases
- Development cost should be no more than $600
- Unit cost should be less than $150
- LED to display when battery needs to be recharged
- Remains in a low power usage mode if not being utilized
- Device should be able to be operated utilizing a smartphone application
- Returns to an owner specified location once its task is complete
- Front wheels should turn to change the carts direction
- Must utilize weight sensors to protect motors
- There must be an LED to indicate when weight limit is reached
- Device should be able to plug into a wall charger
- Wheels do not need to be all terrain, should be suitable for indoor environments
- Battery LED should turn green when fully charged
- Battery LED should turn red when battery is at 10%
- Application should give estimate of how much battery is left
- Application should display when weight limit is reached
- Application should be usable on the Android Operating System
- Application should connect to the cart via Bluetooth
- Cart should be able to move weight atleast 50 ft
- Battery should last cart about 8 hours of on-time
- Anti-theft protocol to prevent cart from leaving specified area
- Weight sensors used to measure the amount of weight on device
  - LED light is triggered when weight limit is reached

2.2 House of Quality Diagram

The house of quality diagram is utilized to determine the tradeoffs, marketing requirements, and engineering requirements needed for the project. The correlation between the marketing requirements and engineering requirements are then determined. This allows for the designer to quickly and clearly see how each requirement affects another requirement.

Table 1: House of Quality Diagram



			Dimensions	Weight	Cost	Load Size	Range	Battery Life
			-	-	-	+	+	+
Marketing Requirements	Durability	+	↑↑	↑	↑↑	↓		
	Ease of Use	+	↓	↓↓			↑↑	↑↑
	Low Cost	-	↓		↑↑	↓		↓↓
	Battery Size	+	↓	↓↓	↑↑	↓	↑↑	↑↑
	Targets for Engineering Requirements		H: <60 in. L: <40 in w: <36 in.	<25 lbs	<$600	≥40 lbs	≥30 feet	<$250

2.3 Hardware & Software Block Diagrams

These block diagrams are useful for quickly showing the responsibilities and tasks that each member of the team is responsible for. These tasks involve both development and research and this section can be referred when determining the author of each section of this paper. The electrical engineering members are responsible for the hardware design while the computer engineering members will be responsible for the user application and the coding for the microcontroller.

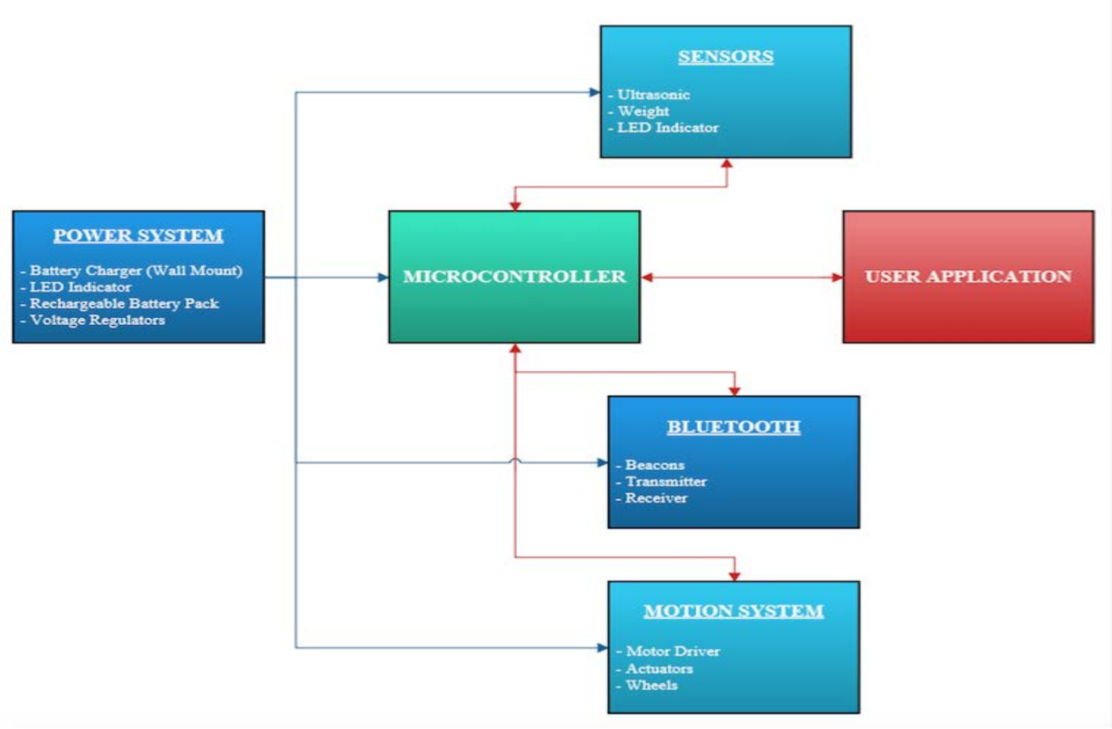
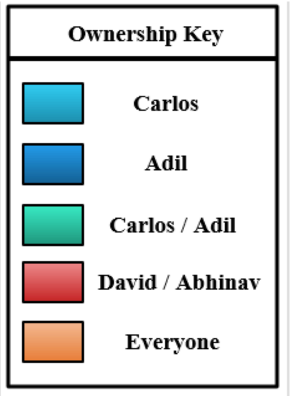


Figure 1: Hardware Block Diagram



Ownership Key	
🟦	Carlos
🟦	Adil
🟩	Carlos / Adil
🟥	David / Abhinav
🟧	Everyone

Figure 2: Ownership Key

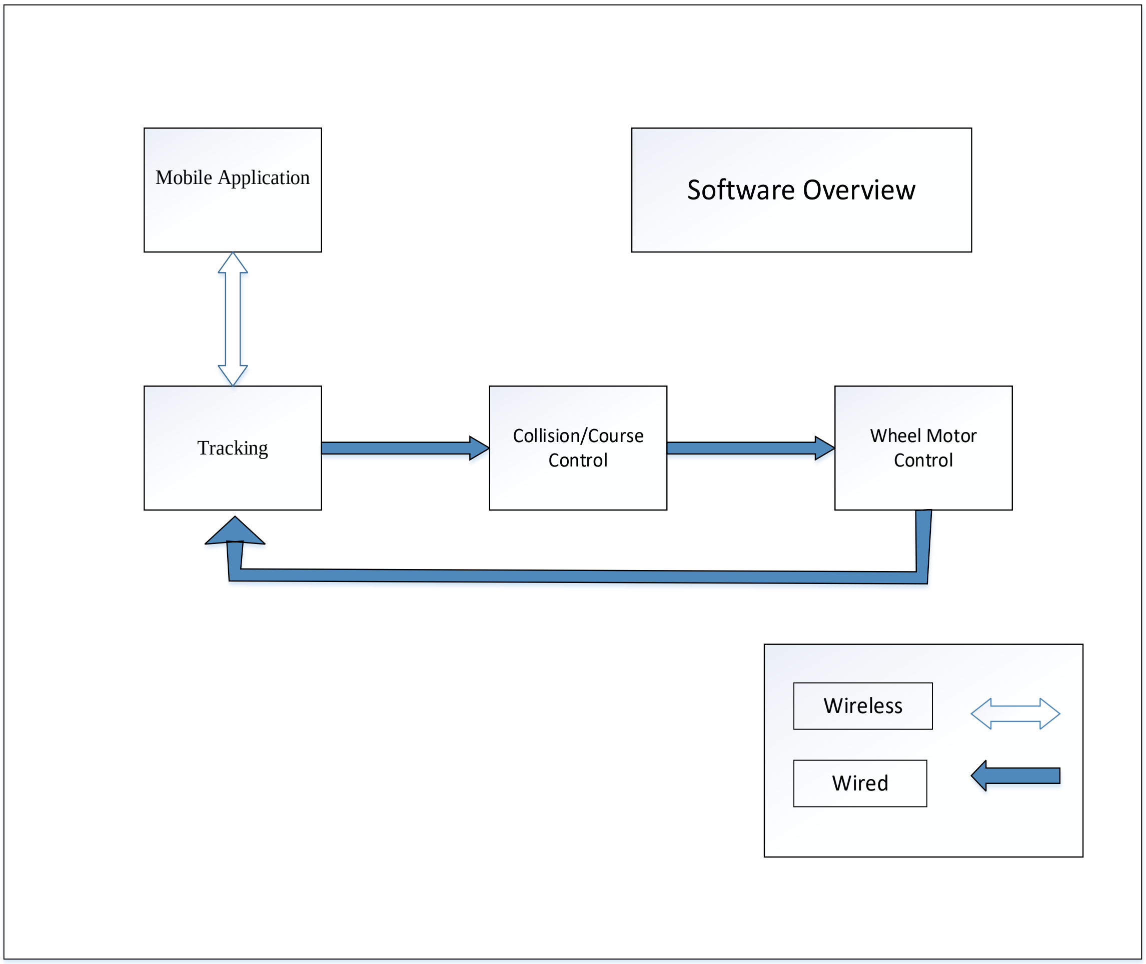


Figure 3: Software Block Diagram

Mobile Application Block

Status: Research

Input: User will input commands such as Connect, Follow, Disconnect, and

Output: Wirelessly communicate to tracking block on who to follow

Owner: David Falter

Additional Details: Will more than likely be developed in Java for android or Objective C

Tracking Block

Status: Research

Input: Information on new direction of vehicle from Wheel Motor Control block and location of mobile phone using GPS or Bluetooth from Mobile Application block.

Output: Suggested direction and distance to Collision/Course Control block

Owner: David

Additional Details: Will likely be using FollowMe GPS based framework frequently used in modern tracking drones

Collision/Course Control Block

Status: Research

Input: Potential threats from collision detection sensors and direction/distance to the target mobile phone from the Tracking block and

Output: Will recalculate path based on foreign object threats so it can pass on the

Information about the desired direction and how far in that direction the vehicle should go to the Wheel Motor Control Block.

Owner: Abhinav

Wheel Control Block:

Status: Research

Input: The Collision/Course Control block will give information change of direction and how far in that direction the vehicle must go.

Output: Give commands to the motors on the speed and number of rotations. Will then give information back to the Tracking or maybe even the Collision/Course control block on the new direction of vehicle

Owner: Abhinav

Each control block is assigned to one group member and although each member is responsible for the task all of the group members must work in coordination in order for there subsystems to function properly. The blocks are not definitive meaning that they may be switched or there may be more than one person working on it depending on the issues or problems that arise during the testing and prototyping phase. These are important to define however as they provide some structure to the design phase.

3.0 Project Summary

Throughout time, convenience through technical solutions has always been a focal point of innovation. The main idea behind the FollowBot also strives for a solution that can make travel easier for those who may need or want it. One of the main applications of this idea is travel in airports. The platform will be manufactured with the idea to be able to support the weight of luggage easily to be able to maneuver around a crowded airport. The convenience of having a following luggage carrier could be very significant to the disabled or people with large families. Often at an airport, people are burdened by having to do multiple tasks while trying to make a flight by a certain deadline. Security lines are unpredictable, and carry-ons must be taken with them all the way to the plane. A following platform able to carry multiple things allows for users to be able to maintain multiple actions and interactions throughout their trip whilst not having to worry about the extra stress of dragging suitcases around. With airports comes a lot of constraints however, due to the restrictions surrounding security issues.

For one, making our technology available for airports has influenced our design choice regarding going with Bluetooth over GPS. GPS is usually blocked in airports for obvious security concerns, whereas Bluetooth is mostly local and only requires receivers throughout the building. Other security concerns around airports include security around sending a platform with so many electrical components through security as well as the issue of having users own a FollowBot. For these reasons, we believe that our technology around airports will not be a consumer product, but rather a modified version provided by airports at a small cost for regular users but complimentary for disabled users, like luggage carts.

A large part of concern around airports is also foot traffic: the large volume of people at airports makes an autonomous robot of this kind hard to produce but makes the functionality of the software that much more important. The "vision" of the bot must be able to navigate through crowds without bumping another person or their luggage as well as being able to maintain a consistent speed to not fall back and hinder the user. The balance required here requires the software to follow the owner close as well as maintain consistent object avoidance. Although the design of the bot is centered around the airport application, there is endless potential for such a product in the consumer market. Another targeted market for application of the design is for personal use at the beach. A motorized platform of this size and purpose can carry supplies such as drinks, coolers, phones, and anything else used for the beach. This sort of consumer application will appeal to plenty of people at the beach looking for convenience and a simple way to move things from point A to point B.

Despite opening a new market here, it brings different sorts of challenges from an airport application. Such a platform would need different wheels to be able to tread through sand easier as well as completely internalized components to make sure water and other natural factors would not render the bot useless. This solution of our software could also be implemented as a cooler product, simply able to carry refrigerated drinks behind the user across multiple terrains. This also translates into many other consumer uses such as help around the house with autonomously being able to carry supplies behind a user working on something. As it can be seen, the FollowBot has potentially infinite applications around the consumer market and industry. The actual technology of basically having an autonomous helper around always is a vision for the future that has been seen for years. A product of this type can be narrowly implemented for a single solution such as airport travel but it would be severely limiting the potential. For our project, navigation around an airport is a focus, but further byproducts of the IP are self-evident.

3.1 Market Analysis

To understand the potential usage of our product, we must first analyze its potential place in today's market. There is always room for convenience, and that is what we are striving for. Today's market in this particular niche can be very hard to find. Things like this seem they should go viral as a "must have" item for every consumer to make their lives easier but truth be told there are not many varieties of this product on the market right now. The idea of a "following suitcase" however, has shown promise in the past and has even been marketed on a relatively large scale, starting through Kickstarter. One current example includes a product from Travelmate Robotics, called the Travelmate. This autonomous suitcase does exactly what it says and follows the user throughout most terrains while connecting to a smartphone. This is an excellent idea of convenience and solves the same problem we aim to. We differentiate, however, because we agree that this idea can and will be bigger in the future.

FollowBot not only seeks to change the way of thinking regarding autonomous robots, but also define a standard for a following bot that seeks to make life just a little bit easier. The competition in the niche on the market seems to aim very specifically at consumers being able to buy a suitcase. There are a few very noticeable problems with this business model. First, in public, theft is a very obvious problem for this solution. Not keeping track of your suitcase because it is following you will attract thieves, especially at such a high price point. Another major flaw in this marketing plan is the sudden loss of convenience when a user must check their bag in at the airport. Sure, the bag will follow someone from their car to the check in booth, but what about your carry-ons?

The whole point of making carrying convenient is the ability to truly relieve this tension across the entire airport. This is specifically why we aim to remedy the problem by aiming to sell the product to airports itself and allow for temporary rental with which the user can automate through the app. This solves a few very important things: a customer does not need to buy a brand-new suitcase to use this functionality, multiple items can be placed on the platform, and theft will be kept low due to the product belonging to an airport and constantly tracked throughout use.

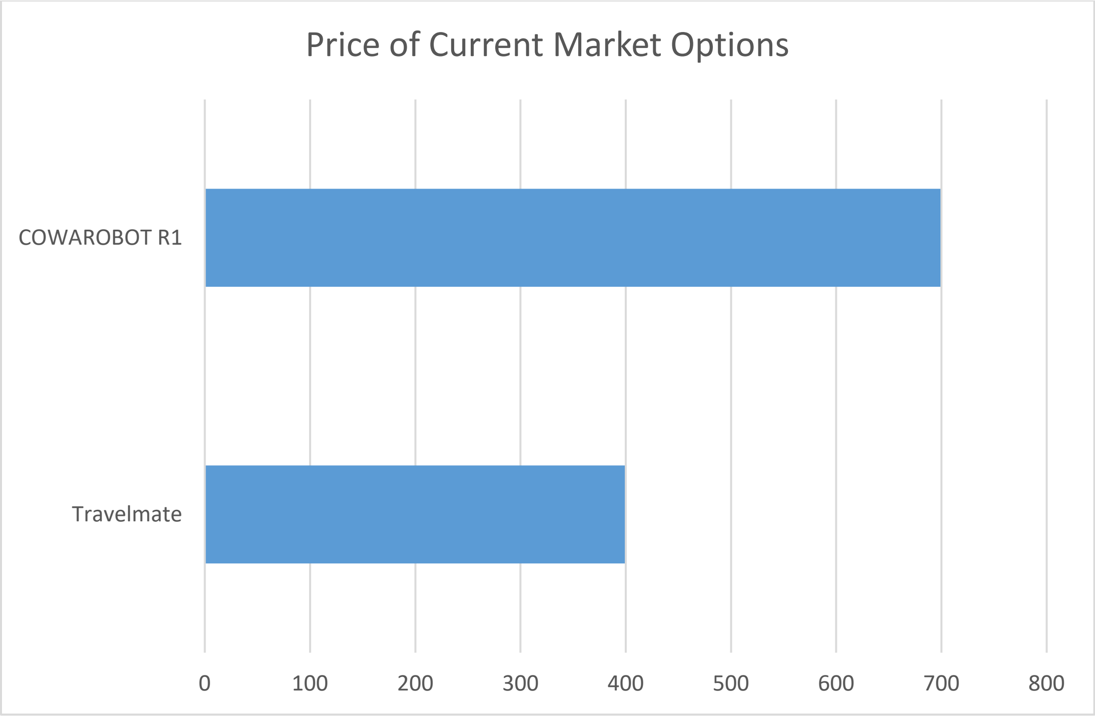


Figure 4: Price Comparison of Similar Market Options

Both options shown highlight why an affordable and reliable solution would greatly benefit the future market. Product rental would put the price on the airports rather than customers and allow for everyone to experience the convenience or luxury should they choose to at a very reasonable price. We believe that FollowBot will ascend these current expectations, however, due to its versatility.

Expanding past the suitcase market, the FollowBot has many other applications. This can be used as a consumer product or marketed to companies because it can be used for practically anything. A few applications which were mentioned include driving around at the beach carrying supplies and drinks or even following you around your home as you make repairs while carrying all your tools. In this regard, no product in our current market rivals the FollowBot and what it aims to bring to the public. If executed correctly, the scope of such a product is endless, and we hope that a lot of people find use of such a project.

As shown the prices of the competition do fall in line with the estimate that are found in the budget section of this report. The FollowBot foundation is the tracking and collision detection which if successful and efficient can also be implemented in a range of other devices such as strollers, etc. Our solution is not and will not be limited to just autonomous luggage carts as we hope to expand to other platforms and other areas of usage.

4.0 Project Research

One of, if not the most crucial design aspects of the autonomous cart is the implementation of the movement and positioning systems. There are a variety of different technologies that were considered such as GPS, Low Energy Bluetooth, and a newly emerging technology known as Chronos. The range as well as operating conditions of these technologies will be discussed and compared.

4.1 GPS

GPS or the global positioning system is a system of satellites that is operated by the US government or more specifically the United States Department of Defense. GPS utilizes at least four satellites to compute three-dimensional positional data as well as the GPS time. The user utilizes a receiver which must have line of sight to at least four GPS satellites and in turn solves mathematical equations to determine the positional data of the receiver.

4.1.1 GPS Positioning Services

While GPS is the system that the US government operates, there are two types of GPS positioning services which are the Precise Positioning Service (PPS) and the Standard Positioning Service (SPS). Previously before the early 2000s the usages and availability of these services varied due to security concerns.

4.1.1.1 Precise positioning Service (PPS)

The Precise Positioning Service (PPS) requires the user to have authorized equipment that is often only available for military uses and can only be obtained for civilian usage by obtaining permission from the US government. The PPS has a higher vertical accuracy than the Standard Positioning Service (SPS) and provides data about the receiver’s velocity as well as positioning and timing data.

4.1.1.2 Standard Positioning Service

The Standard Positioning Service (SPS) is the service that most cell phones and navigational devices utilize today as it is open for public usage and before the 2000s the SPS was unlike PPS because it only sent positional and timing data. The horizontal accuracy of the SPS was once significantly lower than PPS because it was purposefully degraded by the Department of Defense using a timing bias that was known as selective availability which intentionally degrades the satellite clock. However, selective availability has stopped being implemented since 2000. The horizontal accuracy of the SPS used to be around 100 meters and the vertical accuracy approximately 156 meters due to the selective availability however it is relatively close to the PPS today. [1]

Table 2: SPS vs. PPS Comparison [2]

	SPS w/ SA	SPS	PPS
Usage	Civilian	Civilian	Military
Horizontal Accuracy	100 meters	25 meters	22 meters
Vertical Accuracy	156 meters	43 meters	27.7 meters
Timing Accuracy	340 ns	340 ns	200 ns

4.1.2 Possible Design Applications for GPS

GPS is suitable for projects that are operated outside and is beneficial in that it allows for accurate data on an objects position and velocity to be obtained. It is not possible to utilize GPS as the movement system in our design because the autonomous cart will be operated inside a building which means that the cart would not be able to connect to the satellites. GPS is also not accurate enough for this application as both the horizontal and vertical accuracies are over 100 meters.

4.2 Chronos

Chronos is an indoor positioning system that was designed by MIT to provide extremely accurate positional data in environments in which other solutions (such as Bluetooth or RF) would be costly and ineffective. Chronos utilizes a single wireless access point and commercial wireless cards as a receiver to triangulate the position and speed of an object. This is done by applying an algorithm that calculates the time of flight value and then multiplies this value by the speed of light. A multiple input/multiple output (MIMO) or multi antenna access point is then used to calculate the distance between the object and the antennas that are present on the access point.

The usage of these multi antenna access points allows for not only the distance to be calculated but also the angle between the transmitter and receiver. Specifically, both the receiver and transmitter are set to broadcast at frequencies in the standard wireless range (2.4 GHz to 5.8 GHz). There are 35 frequency bands present in the 2.4 GHz and 5 GHz ranges and every 3 microseconds the transmitter changes the frequency it is broadcasting at. The receiver compares the phase differences and computes positional data. [3]

4.2.1 Possible Design Applications for Chronos

Chronos is beneficial when compared to RF and Bluetooth solutions because it allows for the calculation of an absolute time of flight. Bluetooth triangulation on the other hand is performed by calculating the time of flight difference between the Bluetooth receivers and is not as accurate as Chronos. Chronos has the ability to measure an object's position to an accuracy of 65 centimeters and is able to perform these positional calculations incredibly quick (in under a nanosecond).

One problem with Chronos is the fact that the receiver and transmitter must be calibrated to determine the initial distance calculation. Chronos is still in the developmental phase and cannot be utilized in this project however, it was mentioned because it demonstrates a future upgrade path for the movement system and that future technologies might improve both the accuracy and efficiency of the autonomous cart.

4.3 Bluetooth

Bluetooth is a wireless technology that utilizes short-range radio frequencies (RF) that are between 2.400 GHz and 2.4835 GHz (known as the ISM band) in order to create an ad hoc network known as a piconet or a personal-area network (PAN). In a piconet, a singular device acts as a master while the other devices on the network act as slaves or essentially these devices follow commands from the master device. The piconet can include up to eight devices or more specifically one master and seven slaves.

4.3.1 Spread-Spectrum Frequency Hopping

Bluetooth allows for the connection of eight simultaneous devices through a procedure known as spread-spectrum frequency hopping. Spread-spectrum frequency hopping is performed by the master device by randomly determining 79, 1 Mhz frequencies in a different order. The master device can also exclude certain frequencies that are being utilized by connected Bluetooth devices in order to prevent the connected devices from creating interference or crosstalk.

The transmitters are also designed to constantly change between the pseudorandom frequencies every second (more than 1000 times a second) in order to prevent interference from other devices that operate on the 2.4 GHz spectrum. If data is being transmitted over a Bluetooth network, then frequency hopping occurs between package transmission or in essence once a package is transmitted the frequency can then change and another packet of data is sent to the receiver. [4]

4.3.2 Differences between Bluetooth Versions & Classes

Bluetooth has undergone a plethora of transformations due to the introduction of newer and more efficient versions of the standard. Prior to the introduction of Bluetooth 3.0, Bluetooth devices were incredibly unreliable and slow due to the low transfer rates. The arrival of Bluetooth 4.0 includes improvements in both range and speed as well as the Low Energy Bluetooth protocol which was specifically designed to be utilized with IOT devices.

Table 3: Differences between Bluetooth Versions

Bluetooth Versions	**2.1**	**3.0**	**4.0**	**5**
Maximum Range	10 meters	10 meters	60 meters	240 meters
Low Energy Compatible	No	No	Yes	Yes
Maximum Throughput	0.7-2.1 Mbps	0.7-2.1 Mbit/s	305 Kbps	1400 Kbps
Transfer Rates	3 Mbps	24 Mbps	24 Mbps	49 Mbps

There are also three different "classes" of Bluetooth devices which are designed to operate at specific ranges. Personal computers often use Class 1 Bluetooth transmitters which have the largest transmitter range but also consume the most power. Cellular devices and headsets often utilize Class 2 Bluetooth transmitters which have a shorter range than class 1 but also utilizes significantly less power. Class 3 devices have an incredibly short operating distance and are used in specialized cases in which extremely low power usage is necessary.

Table 4: Bluetooth Class Specification

	Maximum Output Power	**Minimum Output Power**	**Range**	**Usage**
Class 1	100 mW (20 dBm)	1 mW	100 m	PCs
Class 2	2.5 mW (4 dBm)	0.25 mW	10 m	Handhelds
Class 3	1 mW (0 dBm)	0.01 mW	1 m	Specialized

4.3.3 Bluetooth Beacons

With the introduction of Bluetooth low energy (Bluetooth 4.0) a new class of devices known as Bluetooth beacons were created. These beacons are hardware transmitters that allow a Bluetooth device to perform a certain action once the device has entered the transmitters range. Bluetooth Low Energy allows for data exchanges to occur in two modes. These modes are the connected mode and the advertising mode. The connected mode transfers data from one device to another device using a layer known as the Generic Attribute layer. Advertising mode on the other hand allows for data to be transferred from one device to a plethora of other devices and utilize a Generic Access Profile.

4.3.3.1 Types of Bluetooth Beacons [5]

There are three commonly used Bluetooth Beacons which are the iBeacon, AltBeacon, URLBeacon (Eddystone). These Beacons all have similar uses and all utilize Bluetooth Low Energy however the ways in which they transfer data slightly differ.

4.3.3.2 iBeacon

The iBeacon was the first Bluetooth beacon which was created and implemented by Apple. The iBeacon transmits four key packets of data the first being a Universally Unique Identifier (UUID) which can trigger beacon specific events such as a certain advertisement being pushed to a user phone depending on which beacon is in proximity. The iBeacon also transmits a Major and Minor number which are used for beacon identification. The Major number is used to identify a specified beacon subset in a larger group of beacons while the Minor number essentially identifies each individual beacon or a specific one.

The final number that is transmitted is the TX power level which measures the signal strength between the beacon and the connected device. This value is then used to determine the distance of the receiver from the transmitter. The TX power level must be calibrated by either the user or the manufacturer in order to be properly utilized. In order to use the iBeacon SDK one must be part of the Apple Development Community which costs $99.

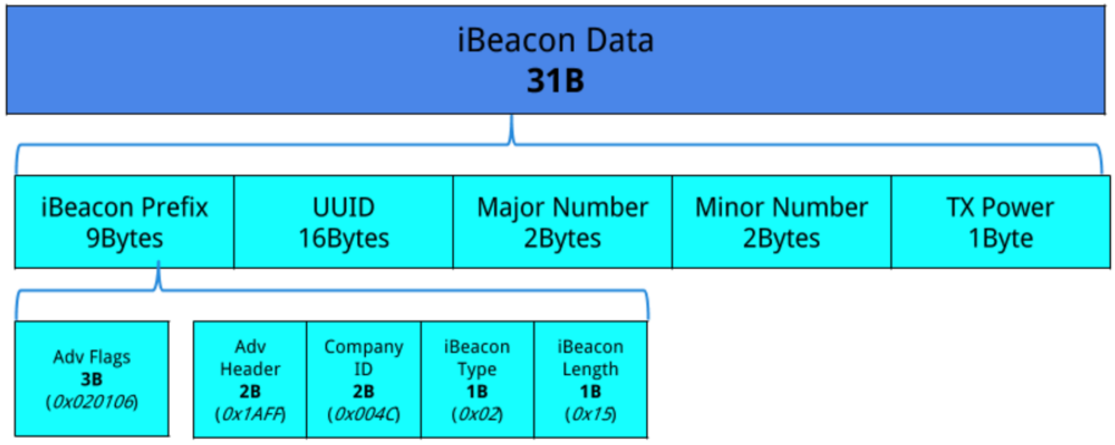


Figure 5: iBeacon Data Structure [5]

4.3.3.3 AltBeacon

AltBeacon is an open source Bluetooth beacon that was created by Radius Networks and is relatively similar to iBeacon except for the fact that the iBeacon is closed source and proprietary. AltBeacon and iBeacon are both essentially used as proximity sensors in order to transmit advertising data or to triangulate positional data. Unlike the iBeacon which has a UUID, Major, and Minor data points, the AltBeacons have an MFGID, BeaconCode, BeaconID, and MFG RSVD. Due to the larger number of bytes available for user data AltBeacon allows for more packages to be delivered per message.

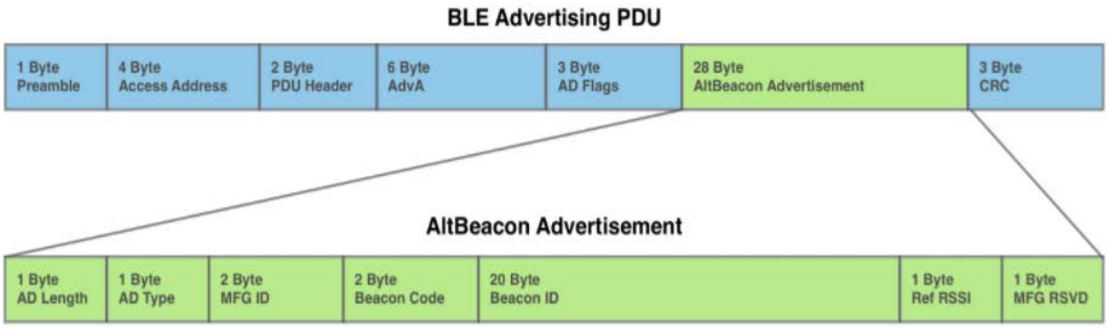


Figure 6: AltBeacon Data Structure [5]

4.3.3.4 URiBeacon

URiBeacons are unbelievably similar to iBeacons however their advertisement packets contain a URL allowing the beacon to link the receiver to a website. For example, if the beacon detected that you were standing in front of a specific product in the store it could link you to that products main page. iBeacons and AltBeacons have a one-time setup and configuration however the URiBeacons have a special service that allows the beacon to be constantly updated. The beacon was developed by Google as an open source alternative to iBeacon. The URiBeacon also does not require a specific database like the other two beacons and can utilize any created website. The beacon’s advertising packets are 28 bytes with 19 of these bytes being used to transmit the URL. Both the URL prefix and the URL suffix are encoded using one byte and only www addresses are compatible however, URL shortening services can be used to limit the size of the URL data transmission.

4.3.3.5 EddyStone

The URiBeacon has been succeeded by Eddystone-URL which can transmit three different types of advertisement packets. The first type is the Eddystone-UID which is a unique identifier for the beacon and can be used to push notifications to a user device or send signals to perform certain actions in applications. The next type is the Eddystone-URL which is in essence the same advertising packets that the URiBeacon uses to push URLs to the user’s device. The final advertising packets is the Eddystone-TLM which transmits telemetric data that is collected by the beacons’ sensors. These packets can trigger events based on a variety of different factors. These factors include but are not limited to temperature, humidity, sound levels, etc.

Table 5: Beacon Comparison

Beacon	iBeacon	AltBeacon	URiBeacon	Eddystone
Compatible OS	iOS	Android	Android, iOS	Android, iOS
Available User Data Space	20 B	25 B	28 B	Up to 31 B**
Trigger Push Notifications	Yes	Yes	No	Yes
Cost	$99+$15 per Beacon	$29	$29.95	$22.60

**Depending on which type of advertising packet is transmitted

4.3.4 Bluetooth Modules

Bluetooth modules are compact and relatively cheap devices that are manufactured for usage with microcontrollers. These devices can be found in a variety of form factor and some microcontrollers even have modules built into the board. These modules come Bluetooth certified and are created with APIs to be easily accessible.

4.3.4.1 RN4020

The RN4020 is a Bluetooth Low Energy (Bluetooth 4.1) module that is created by Microchip. The module utilizes a PCB antenna that has a maximum range of approximately 100 meters and the module is compatible with all microcontrollers. The module has 64 KB of internal flash and has an incredibly low power usage due to the Bluetooth Low Energy standard. The device seems the most suitable for our autonomous cart however other options will be discussed.

4.3.4.2 RN52N

The RN52N was the second Bluetooth module that was considered however the module is only Bluetooth 3.0 compatible which would not operate well with our Low Energy Bluetooth Beacons. The module is also a class 2 Bluetooth device which means it has a maximum operating range of 10 meters which is much lower than the RN4020 Bluetooth module. These modules while similarly priced are meant more for usage in an audio device such as a Bluetooth speaker or headset.

4.3.4.3 CC2564MODx

The CC2564MODx are a series of Bluetooth modules that are constructed and distributed by Texas Instruments for use with their MSP43x boards. These modules are compliant with the Bluetooth 4.1 standard and therefore supports Bluetooth Low Energy devices. This makes them perfect for usage with Bluetooth Beacons. The CC2564MODA has an integrated PCB antenna while the CC2564MODN has an external antenna however most the main specifications such as power usage or size are relatively the same.

Table 6: Bluetooth Module Comparison

Modules	RN4020	RN52	CC2564MODA
Operating Voltage	1.8 V to 3.6 V	3.0 V to 3.6 V	2.2 V to 4.8 V
Size	11.5x19.5x2.5 mm	13.4x25.8x2.4 mm	7.0x7.0x1.4 mm
Power Consumption (Idle)	<1.5 mA	12 mA	40 uA
Power Consumption (Active)	16 mA	40 mA	41.2 mA
TX Power	-19 dBm to +7.5 dBm	4 dBm	10 dBm
Bluetooth Version	4.1	2.1	4.1
Operating Range	100 meters	10 meters	100 meters
Antenna Type	PCB	PCB	PCB

4.4 Ultrasonic Sensors

The sensors are the next key part of object detection and collision. We are considering different technologies as well as different types and brands of each technology. The standard route for collision detection usually relies on a camera as seen in modern cars but is also often reliant on ultrasonic sensors. We immediately crossed off the idea of using a camera because it does not fit our vision for the product. A camera in a crowded place can pick up plenty of interference and will not always function correctly in terms of detecting distance. Another common technology is infrared sensing which is commonly used in remotes and various motion detection sensors.

While infrared is great at detecting whether an object is imminent or not, it is poor at detecting distance and sending that information to the MCU. The strongest contender for collision detection is ultrasonic, purely due to the versatility and accuracy it provides. Ultrasonic sensors work by essentially pinging a surface and receiving information based on time travelled to determine relative location. This is perfect for our application: a constant stream of data that can be used to determine how close the FollowBot is to an object in front of it allows us to control its movement in real time. Ultrasonic sensors are often used in industry for this exact purpose, and we feel as if they are the best solution in terms of technology for real time collision detection. We have considered a few options as well for sensors, all of which are similar, but provide different advantages

4.4.0.1 Ultrasonic v. Infrared Sensing

An ultrasonic sensor utilizes a time of flight measurement to calculate the distance the object is at. This is done by measuring the time it takes for a high frequency auditory waveform to travel to the object in question and back to the receiver. An infrared sensor on the other hand measures light waves that are present in the IR spectrum (760 nm). Similar to the ultrasonic sensor, a IR wave is sent to the object and the change of intensity of the returning IR wave determines the distance the object is at. The weakness of this method is that the intensity measurement could be affected by the material color. IR sensors often have an incredibly low response time as light travels faster than sound. It is important to consider the environmental constraints of an ultrasonic sensor such as the fact that the temperature or humidity can affect the speed of the sound wave. When choosing sensors, it is important to consider what material will your main obstacle as the material type (plastic, wood, etc.) can affect the efficiency of the sensors. [6]

4.4.1 HC-SR04 Ultrasonic Ranging Sensor

This sensor provides up to 400cm of measurement with ranging accuracy up to 3mm. In a practical application of our collision detection system, we ideally want our product to stop at least 1ft away from the object in its path. By taking a fairly large cushion to allow for braking, the software should detect the object at 18 inches away and come to a stop by 12. With such a large range, the HC-SR04 sensor provides plenty of leeway and allows us to execute our needs within a great margin. A bonus of this item is ease of use: the sensor only has four pins and will do exactly what it needs to do. This product comes in at only $3.95 which is very cheap and allows us to order multiple parts in case we need to replace the sensors or we decide to go with another brand.

Figure 7: HC-SR04 Sensor

4.4.2 Maxbotix LV-EZ3

Now to discuss one of the higher end options, the Maxbotix Ultrasonic Rangefinder LV-EZ3. This device to start comes in at $24.95 which is significantly higher than its counterpart, the cheap HC-SR04 sensor. Price is not always the most important factor though, and the LV-EZ3 does offer a very strong advantage in certain areas. The LV-EZ3 first provides us with up to 254 inches of range information, allowing us to detect collisions from a very impressive 21ft away. The LV-EZ3 is easily usable with an Arduino, and Maxbotix even provides sample code to be able to use it with your own project. This product is compact, efficient, and very accurate, along with significantly better build quality than the HC-SR04. This is a valid option, but the cost just may be too much for our purposes considering how inexpensive the alternative is.

Figure 8: Maxbotix LV-EZ3 Sensor

4.4.3 Parallax PING))) Ultrasonic Distance Sensor

This sensor makes the list due to being a valid competitor but is not one of the strongest contenders. This Parallax sensor provides a range from approximately 1 inch to 10ft and similar to the other options, is accurate to about 1-3cm. The reason this sensor is not as impressive as the others largely resides with its functionality compared to its price of $29.99. The sensor itself has great reviews and seems to work quite well but just does not compare well to the other competitors. An advantage this Parallax sensor possesses is documentation and free source code that comes along to help with implementation. The support site for purchasing this sensor provides almost 8 different small projects to help with implementation, which could be very useful for our project. This sensor is considerable due to the implementation support that Parallax seems to offer, but it still stands at the most expensive and does not provide the range benefits that the Maxbotix does.

Figure 9: Parallax Ping))) Sensor

Table 7: Ultrasonic Sensor Comparison Chart

Sensor	Range	Cost	Resolution	Support
HC-SR04	2cm – 400cm	$3.95	4cm	Little to none
LV-EZ3	0cm – 645cm	$24.95	2.5cm	Good
PING)))	2.5cm – 304cm	$29.99	4cm	Excellent

All information provided by manufacturer websites.

The software that will go into steering our product for collision detection and avoidance is one of the pillars of development for our project to be successful. If the device is not able to avoid obstacles well, it will be practically useless in any demo setting. All three ultrasonic sensors discussed provide solutions to our problems at hand, but some more effectively than others. For this project, I believe price will be the steering factor here, giving the HC-SR04 sensor an upper hand above all others. The price point is just too good to pass up and allows for a lot of room for error on our part. The implementation can be improvised without resources and the range can be managed in such a controlled environment such as ours.

4.5 Microcontrollers

The microcontroller is a key part of this project: it will drive the hardware because of our software programming and allow it to do all the functionality we plan on implementing. The microcontroller is an important facet we cannot overlook, and thus the solution must be given much consideration. The microcontroller will be used for quite a few tasks: motion controls of the motors because of software input, collision detection and avoidance, and Bluetooth pairing and functionality. The MCU (microcontroller unit) should handle these tasks with ease and maintain high functionality in real time to not hinder the use of the product. Since the MCU is the backbone of combining our software and hardware, it is very important to understand the benefits and drawbacks of several types and brands.

4.5.1 MSP432 Microcontrollers

The first MCU in consideration is a new edition of a very familiar line of MCUs that Texas Instruments produces. The MSP 432 line of microcontrollers brings high-precision signals with outstandingly low power usage while still maintaining a high level of functionality. TI's proprietary SimpleLink technology provides wireless connectivity applications and easy development with plenty of documentation through the platform. The MSP 432 is a leading contender in the race to decide a MCU due to TI's prowess in the MCU industry. The MSP 430 line has been an industry leading standard for years due to its respective power usage and functionality.

The MSP 432 provides responsive out-of-the-box development with TI-RTOS and an immense library of provided APIs to achieve any level of implementation with support from the manufacturer. The MCU provides up to 256kB flash memory and 64kB of SRAM which is above and beyond our applications. Another bonus of this device is the relatively low cost. The unit price of an MSP 432 chip hovers around $10, very inexpensive for an MCU and perfect for our planned costs. A bonus of this TI family of chips is due to UCF's class instruction in the past, which has brought us all experience with programming on an MSP 430 family chip before. The development on this MCU is relatively easy to learn and expansive due to resources provided by TI, furthering it in the list of MCU's considered.

4.5.2 ATmega328P (Arduino Uno)

The second MCU in our list of considerations is often considered a fan favorite for tinkerers and small project designers around the world. The Arduino Uno MCU board which is based on the ATmega328P is a very familiar microcontroller board that supports a wide range of applications. The Arduino Uno MCU board is thus one of the top contenders for the choice of microcontroller. The biggest bonus of going with Arduino to begin with is the overwhelming amount of resources available on the internet to develop with the Arduino Uno. People have been working on Arduino MCUs for years now and it has always had the upper hand for people looking to get into custom projects due to its ease of use. This, of course, means the Uno comes at a slightly higher cost than its major contender, the MSP 432. The Arduino Uno board package comes in at $24.95 excluding tax and shipping. This is by no means a ridiculous price for an MCU board package like this, but is not quite as inexpensive as the MSP 432.

One of the biggest reasons this board is near the top of our list is due to its inseparable relationship with Java. The Arduino IDE is even written in Java, and everything on the board can be used to communicate in programs written in Java. The software engineers on our team feel very comfortable with the language and thus feel as though the Arduino Uno could expedite the development process greatly. This microcontroller solution feels easy, but reliable, and thus makes it a very strong contender. Despite being a "user friendly" choice in the market, it is hard to underestimate the shadow that it casts. Arduino has been around for a long time taking the reigns as the leader of the public consumer market. This board may not see as much use in industry, but it certainly seems like a strong option for a project such as ours.

4.5.3 BeagleBone Black

The next product in consideration is the BeagleBone Black development board. This is very different than our other MCU contenders, and for good reason. The BeagleBone kits are essentially small computers, and not as much microcontrollers. They host actual ARM processors on them and can run operating systems such as Linux with ease. The BeagleBone Black comes with WiFi, Bluetooth, servo and motor controllers, 9 axis IMU sensors, a barometer, and a wide range of support for additional add-on sensors. This product is a beast, and for good reason: the BeagleBone Black kit comes in at a whopping $54.95, which is 5.5 times the cost of the MSP 432 and over 2 times the Arduino Uno. This price could be justified as the BeagleBone Black has a plethora of UART and I2C ports as well as an upto 720 MHz processor.

Using a strong board like this has a lot of advantages, of course, but does not come without its drawbacks. The cost, namely, is the largest. Followed by cost, the board draws a lot of power compared to our other contenders. This may not seem that big an issue for a product such as ours but minimizing power usage where we can will greatly enable us to make the product lighter and allow for longer actual product usage on a full battery charge. The kit that BeagleBone provides does give us a lot of options regarding development and housing a lot of the sensors we are planning to add onboard, but it seems to severely skimp out on our priorities. Although this option may be our strongest competitor in power, it seems to be our weakest in efficiency, and thus becomes one of the “probably not” boards on our list.

4.5.4 Raspberry Pi 2

Like our last contender, the Raspberry Pi 2 is technically not a microcontroller, but rather an entire computer. The device offers many of the same benefits that a BeagleBone Black provides, and can solve our problems very efficiently as well. Similar to Arduino, the Raspberry Pi brand has been a namesake in the hobbyist realm for years and is often regarded as a “must have” for young engineers to work on and experiment with. This, of course, brings us back to the reason Arduino was so high on this list, and that is community involvement. The tremendous amount of resources we have at hand going with such a board raises the stakes of such a choice a bit every time.

The development cost in terms of time goes down significantly when we can find APIs and libraries that can make development smoother and more applicable to our specific need. The reason this solution is so low on our list is 2 major things: the cost does not justify the power of the board at ~$40 and everything the RP2 can do, the BeagleBone Black can pretty much do better. If we were to decide to go with a minicomputer solution such as this, it would almost unequivocally be the BeagleBone Black. The reason this unit still holds consideration though is because it might eventually win in providing us with enough readily available solutions to problems we may run into in development. The community around the Raspberry Pi products has always been keen and we are almost certain that our problems have a solution somewhere in the community at this point.

Table 8: MCU Solution Comparison Chart

MCU Solution	**Cost**	**Active Power Consumption**	**Bluetooth?**	**Community**	**Development**
MSP 432	~$10	80uA/MHz	Add-on	Good	Fair
Arduino Uno	~$25	20mA/MHz	Add-on	Excellent	Quite Easy
BB Black	~$55	460mA/MHz	Onboard	Average	Easy
Raspberry Pi 2	~$40	240mA/MHz	Onboard	Excellent	Easy

All information provided by manufacturer websites.

4.5.5 Microcontroller Comparisons

Overall, all the above solutions will get the job done in the end. There is much importance, however, in achieving the best solution possible. Our top priorities are stable cost and ease of use and development. Considering there are two clear winners in cost and two clear winners in ease of development, the issue of decision becomes a lot harder. There is a fine line between picking the efficient choice and the easy choice. They are all great solutions by very good companies but they each offer their own advantages. The ultra-low power consumption of the MSP 432 makes it a very tempting choice for battery use and long lasting performance along with its low cost allowing us to swap it out if need be. The Arduino Uno provides top-of-the-line community support with a lot of public source code and examples available as well as native Java support in the IDE for easy development.

The BeagleBone Black and the Raspberry Pi 2 are mini-computers capable of much more than just microcontrollers and give us a wide array of options to use onboard modules and condense our cost into one part, but raise overall cost and make development a little less straightforward. We believe that the Arduino Uno and the MSP 432 are strong contenders for the place on the project and a case could be argued for both. The bigger boards are most likely out purely due to the cost but could still potentially be surprise options if there seems to be an opportunity to reduce cost overall by implementing one of them. By comparing the MSP 432 and Arduino Uno, however, we can see that we must decide on whether we value cost and power efficiency more or ease of development at the cost of those things. Development on the MSP 432 is not hard, per se, but is a little less streamlined for regular consumers and hobbyists like us in this case and more targeted towards companies and industry.

Table 9: Comparison of Microcontroller Specifications

	MSP432P401R	**ATmega328P**	**BeagleBone Black**	**Raspberry Pi 2**
Max. Frequency	48 MHz	20 MHz	720 MHz	900 MHz
RAM Memory	256 KB	32 KB	256 KB L2	1 GB
Voltage Range	1.62 V to 3.7 V	1.8 V to 5.5 V	1.71 V to 3.47 V	1.8 V to 5 V
Temperature Range	-40 to 85	-40 to 105	0 to 90	-40 to 90
Cost	$7.89	$5.95	$54.95	$35.00

Table 10: Specifications for Selected Parts

Components of FollowBot	**Voltage**	**Current (Idle)**	**Current (Active)**	**Power**
RN4020 Bluetooth*	3.3 V	<1.5 mA	16 mA	52.8 mW
ATmega328p	3.3 V	16.32 mA	17.52 mA	57.816 mW
HCSR04	5 V	15 mA	15 mA	75 mW

4.6 Motion System Analysis

The design of the motion system for this project is relatively complex as there are a variety of different motors and actuators that can be used. The types of DC motors as well as the types of servo motors will be extensively discussed and compared in this section and a choice will be made as to which motor is suitable for our requirement specifications.

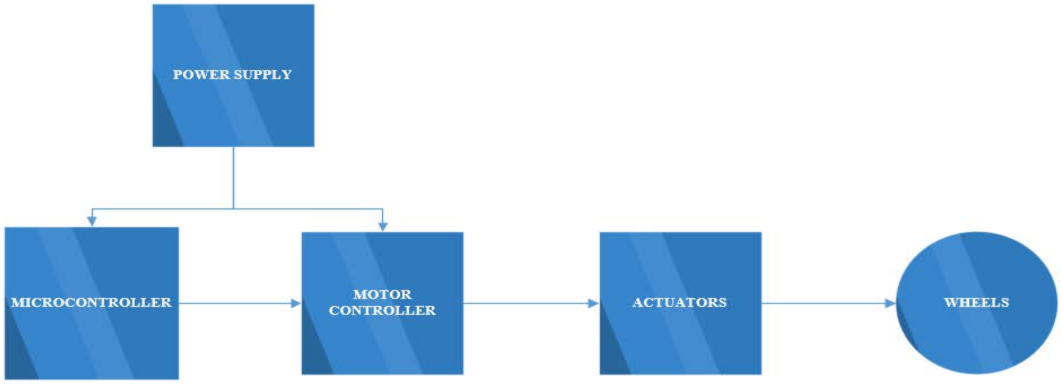


Figure 10: Motion System Block Diagram

4.6.1 Actuators

Actuators are an important aspect when designing a robot. An actuator is a functional element which connects the information processing part of an electronic control system with a technical or nontechnical process [7]. For actuators to work, they require some type of control signal and a source of energy. Once the control signal is received, the actuator responds by converting energy into mechanical motion. For this project, the converting energy will be electrical and the control system will be software-based. Multiple actuators can be used for a mechanism depending on the requirements needed. Below are backgrounds and specifications of actuators that will be considered for this project.

4.6.2 DC & Geared DC Motors

The first motor that will be discussed are DC (direct current) motors. DC motors come in an array of shapes and sizes, however most of them are cylindrical. They can be found in many portable home appliances, automobiles, and types of industrial equipment. Most of them come in both 12V (volt) and 24V DC. Nearly all types of DC motors have some internal mechanism to periodically change the direction of current flow in part of the motor. They possess loops that are made of simple coils. This is called the armature.

The armature is connected to a DC power source through a pair of commutator rings. When current flows through the coils, an electromagnetic force is induced on it according to the Lorentz Law. This will allow the loops to start rotating. Numerous loops are added in the DC motor because of torque. If there were only one loop of coil, the result would be irregular motion of the rotor. Furthermore, the more such loops, the smoother the motor rotation will be. For DC motors, they can operate in clockwise (CW) and counter clockwise (CCW) rotation [8]. This is an important aspect to consider for a robotic system.

4.6.2.1 Physics of Motors

Below shows the formula and a description of the Lorentz Law.

Lorentz Law: $\mathbf{F} = q\mathbf{E} + q\mathbf{v} \times \mathbf{B}$

- q – charged particle
- **v** – velocity
- **E** – electric field
- **B** – magnetic field
- **F –** electromagnetic force (Lorentz Force)

In physics, the Lorentz Law is the combination of electric and magnetic force on a point charge due to electromagnetic fields. A charged particle moving with a velocity in the company of an electrical and magnetic field creates the Lorentz Force. In a DC motor, a current is passed through wires shaped in coils. Since the current produces a movement of charges in the wire, the Lorentz force acts on the moving charges [9].

DC motors exhibit a unique characteristic, the production of back electromotive force (BEMF). BEMF is a voltage that emerges in the opposite direction to current flow because of the motor's coils moving corresponding to a magnetic field. A rotating loop in a magnetic field will produce an EMF according to the principle of the electric magnetic induction. This case is the same with rotating armature loops. An internal EMF will be induced that opposes to the applied input voltage. Magnitude of BEMF is directly proportional to the speed of the motor [10]. The BEMF reduces armature current by a huge amount and is proportional to the speed of the rotor. An equation of the armature current relative to the input EMF and the BEMF is shown below.

$$\mathbf{I = (EMF_{in} – EMF_{back}) / R}$$

At the starting of the motor BEMF is too low, thus, the armature current becomes too high. Ultimately, this leads to the burnout of the rotor. For large DC motors, a cooling fan is added to the mechanism to control the applied voltage.

There are two different kinds of DC motor constructions, a shunt and a series-wound motor. The series-wound motor has a good starting torque, but speed drops immensely with weight. The shunt motor has a low starting torque, but it can run at an almost constant speed regardless of the weight acting on the motor.

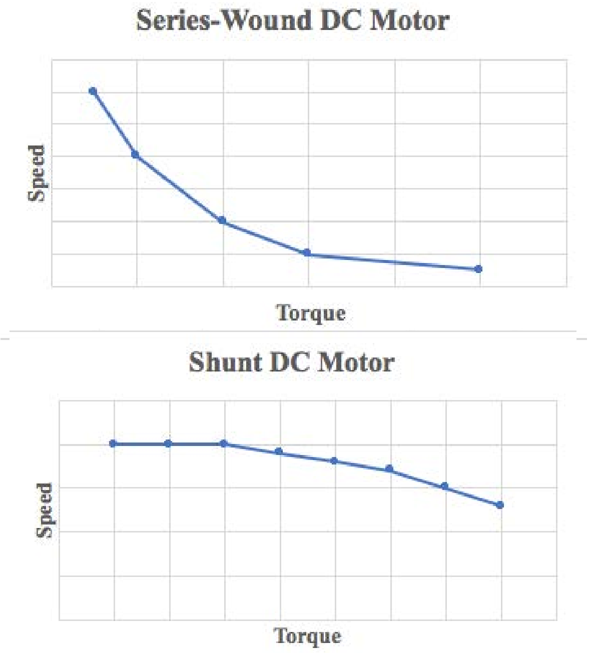


Figure 11: Torque v. Speed for Series-Wound and Shunt DC Motors

Torque has been mentioned plenty of times now and that is because it is obviously an important concept to include when designing the motion of a robotic system. Torque is known as a rotational force. In other words, it can be known as a twist to an object. It can be calculated by a force that is acting at a distance away from a pivot multiplied by the distance.

DC motors rotate rapidly and for most cases, they have low torque (not strong) [8]. To increase the torque in the DC motor, a gear may be added. However, the trade-off of adding a gear to a DC motor is a decrease in the motor's speed. We will call this type of motor, geared DC motors. Just like a DC motor, they can rotate clockwise and counter-clockwise. They have four types of gearing mechanisms called spur, helical, planetary, and worm. These gearing mechanisms are described below.

4.6.3 Spur Geared DC Motors

Spur gear motors are in great demand across various industries. They are made with straight teeth mounted on a parallel shaft [11]. They are the cheapest gearbox to manufacture. They are considered for applications in the fields of robotics, industrial actuators (locking mechanism), or animated static displays. Features such as voltage, torque, speed, gear ratio, body diameter, and the composition of materials classify them. While spur gears are common and cheap, they are not the best solution when high torque or speed is needed. They can be boisterous to operate and the stress on the gears can be damaging if a lot of force is applied.

Advantages:

- Cheap to manufacture
- Dependable when used at low speeds
- Simple in operation

Disadvantages:

- Not good at handling large amounts of torque
- Noisy when rotational speeds are high
- Needs many gears for large reductions

4.6.4 Helical Geared DC Motors

Helical gear motors are the next evolutionary step from spur gear motors. Their design incorporates teeth that are cut at an angle to the leading edge of the gear. This allows the gears to mesh together over the length of the helical cut. Helical gears have a much stable transfer of torque and less of a noise generation compared to spur gears at similar speeds. Applications using helical gears are applicable in high horsepower [11].

Advantages:

- Simple in operation
- Good torque and noise properties
- Cheap to manufacture

Disadvantages:

- More complex and expensive
- Requires many gears for large reductions
- Less noise than spur gears at high speeds

4.6.5 Planetary Geared DC Motors

Planetary gear motors are also known as Epicyclical gearing. They consist of one or more spur gears revolving around a central gear and the planet carrier (the outer gear). Companies have tested conditions with these motors, which include vibration levels, temperature, humidity, structural load, altitude, electrical loading, and mechanical loading. They are a more complex gearbox but can contribute great torque transmission, low noise, and high efficiency.

Advantages:

- Small size
- High efficiency
- Good load distribution which gives great torque capability
- Can handle extremely large reductions

Disadvantages:

- Noisy when rotational speeds are high
- Stress is high on the output shaft bearings.
- Expensive to manufacture

4.6.6 Worm Geared DC Motors

Worm gear motors have great attributes, which makes them convenient for a wide range of applications. They are generally low speed, but are capable of immensely high torque. By the nature of their design, break features are offered. This is possible because the worm can turn the gear, but the gear cannot turn the worm [11]. Since there is a break option for this motor, there is no power applied to the worm drive, thus the load will not be able to turn the motor. The downfall of worm drives is efficiency. The average efficiency of a worm drive is around 65%, which means a high-power loss.

Advantages:

- Huge torque capability
- Good torque and noise properties
- Benefit of braking
- Can handle extremely large reductions
- Simple in operation and easy to maintain

Disadvantages:

- Low efficiency
- Can generally be quite large
- Not suitable for high speed rotation

Below are images of a DC motor and a DC geared motor.

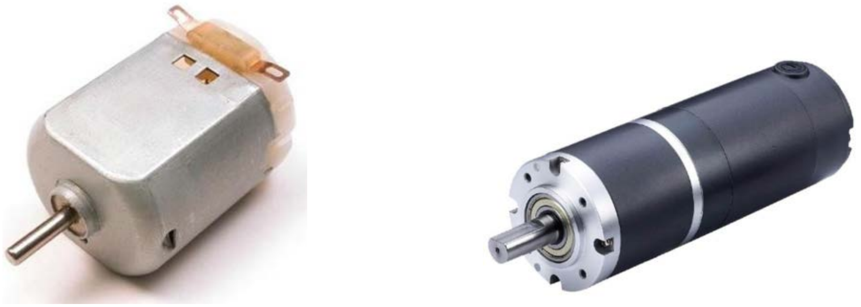


Figure 12: DC Motor (Left) and DC Geared Motor (Right)

4.6.7 Servo Motors

Servo motors are a rotary or linear actuator that allows for precise control of angular or linear position, velocity, and acceleration. They are utilized in many applications such as industrial, robotics, automated machinery, in-line manufacturing, and even pharmaceutics and food services. They are used the most in more upscale remote controlled vehicles for steering or controlling flight surfaces [8]. Furthermore, the price of servo motors has decreased significantly as aspects such as sizes, technology, and strength has increased. For servo motors, they can operate in clockwise and counter clockwise rotation.

There are several components that make up servo motors. Servo motors include a DC motor, potentiometer, gearing, and a control circuit. The motor is attached by gears to the control wheel and is coupled to a sensor for position feedback. The potentiometer is connected to the output shaft. The control circuit and potentiometer work together to activate the motor and to stop the output shaft at a specific angle. As the motor rotates, the potentiometer’s resistance changes. This will allow the control circuit to accurately determine how much movement there is and the direction.

Servo motors require a relatively sophisticated controller that is often uniquely designed for them. They generally have three wires for ground, voltage in, and a control pulse. The control pulse is usually developed with the controller [8]. Through the wire used for the control pulse, servo motors can be controlled by sending an electrical pulse of variable width. This is also known as pulse width modulation (PWM).

More information about potentiometers and pulse width modulation is described below.

4.6.8 Potentiometer

A potentiometer is a device that measures electric potential (voltage). It can be used as a voltage divider to attain a manually adjustable output voltage. There are several types of potentiometers. Some of the types are digital, membrane, linear, and logarithmic. Depending on the type, it can be used in applications such as volume control, brightness control, audio control, etc. There are three terminals that are available in a potentiometer. These terminals are generally connected to ground, a current source, and a sliding contact that goes with a strip of resistor. A potentiometer can also determine the amount of current flow in a circuit.

4.6.8.1 Pulse Width Modulation

Pulse width modulation is a unique modulation technique used to encode a message into a signal. The signal is a square wave that is continuously switched on and off. This pattern of turning on and off can simulate a range of voltage by changing the amount of time the signal spends on against the time the signal spends off. The continuation of the time that the signal is on is called, the pulse width.

There is a minimum pulse, a maximum pulse, and a repetition rate [12]. When the signal is minimum pulse, the pulse width will be around 1ms (millisecond) and move the motor in a direction of zero degrees. When the signal is maximum pulse, the pulse width will be around 2ms and move the motor in a direction of 180 degrees. There is also a neutral position and this occurs when the pulse width is 1.5ms. This will make the motor turn to the 90-degree position.

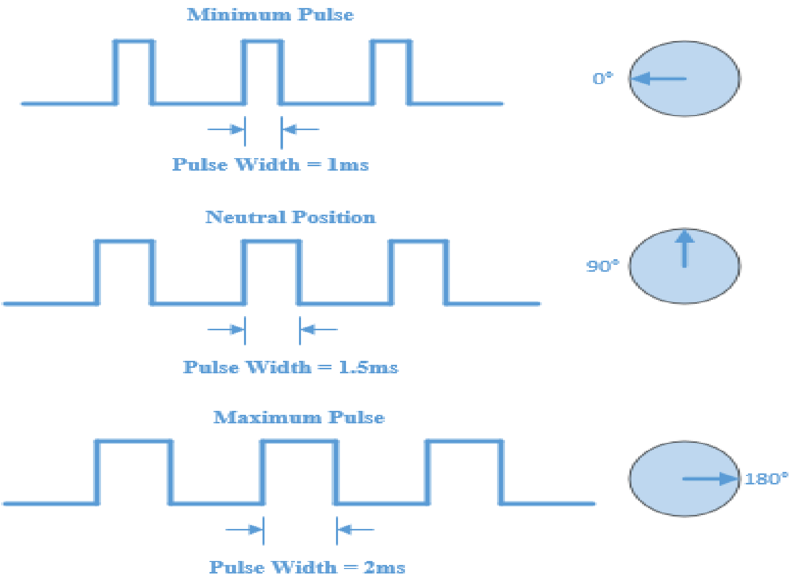


Figure 13: Servo Position via Pulse Width

There are two types of servo motors, DC and AC. DC servo motors are used more for designs such as this project because of their ability to operate continuous rotation. They are also way less expensive compared to AC motors. AC motors can control high surges of current [6]. This makes them more suitable for designs in areas like CNC machinery or automated manufacturing.

Figure 14: DC (Left) and AC (right) Servo Motors

Below are some advantages and disadvantages for only the DC servo motor since it is more suitable for this type of project.

Advantages:

- Higher power output than the motor's size and weight
- Not much noise at high speed
- Minimum vibration and resonance
- Higher torque to inertia ratio which leads to a quick response to the control signals
- Extremely efficient

- Can provide 3-10 times their rated torque for a short time
- Current drawn is proportional to load which allows the motor to stay cool
- Encoder establishes accuracy and resolution

Disadvantages:

- Too much load can damage motor
- Requires encoder which can be complex
- The peak power of the motor will only be generated at high speeds and requires consistent gearing
- Motors can get easily get contaminated because of an inefficient cooling mechanism
- Takes 10 times the average power supply to use peak torque

4.6.9 Stepper Motor

Stepper motors are DC motors that can rotate in specified degrees or “steps”, hence the name of the motor. They are a unique type of actuator because they can control the angular position of the rotor without the need of a closed feedback loop. They are designed with various coils that are coordinated in groups called phases. The phases are powered in sequence which will allow the motor to rotate one step at a time [13]. There are several factors that determine specifications of the stepper motor like torque, resolution, speed, and the amount of degrees the shaft rotates. Some of these factors include motor size and type, step count, shaft style, and gearing. These factors are further explained below. Furthermore, stepper motors can also rotate clockwise and counter clockwise.

Size and Type: Stepper motors come in several sizes and types. With the increase in size, power and other electrical characteristics are considered. The most common types of stepper motors are permanent magnet or hybrid.

4.6.9.1 Hybrid Stepper Motor

They contribute great performance in areas such as step resolution, speed, and torque. The number of steps per revolution (SPR) can range from around 200 to 400. They are designed with a magnetized rotor with two steel caps at each end of the magnet. The polarity is determined by the ends of the magnet which are classified as stator poles. These poles are energized by the coils relative to them which then produces the step angle resolution. The accuracy of this motor comes from the clever arrangement of the rotor and stator teeth.

4.6.9.2 Permanent Magnet Stepper Motor

They use permanent magnetic rotors that are controlled by electrical pulses. They require no “teeth” (the edges of a rotor) and consist of coils, stator yoke, and input/output terminals and stators that are molded together. To improve the resolution of rotor rotation or decrease the step angle, more pole pairs or phases are added. Rotation, in a precise direction, is accomplished by applying a voltage to the individual phases in a precise sequence. As current begins to flow it will generate magnetic polarity on each stator. Some applications that include this type of motor are printers, scanners, copiers, etc. Permanent magnetic stepper motors carry characteristics such as being able to comply with automated motor assembly processes, reduction of steps, and even increase of holding and effective torque.

4.6.9.2.1 Step Count

This factor is very imperative because it is the positioning resolution required for the project. The total number of steps per resolution usually ranges from around 4 to 400. Degrees per step is another way of interpreting resolution. For example, if a motor is classified as 1.8 degrees, this is also stating that the motor has a 200 SPR [13]. Step counting grants high resolution, however there are some trade-offs. These trade-offs are a reduction in speed and torque.

4.6.9.2.2 Shaft Style

Shafts are a mechanical component for conducting rotation and torque. They are used so the motor can interface with the rest of the motion drive system. There are several types of shafts that are used for designs such as this project. Some of these types are round shafts, geared shafts, and lead-screw shafts. Round shafts have one side that is flattened to avoid slippage [13]. These shafts are perfect when high torque is needed. Geared shafts contain teeth that are combined with the shaft. They provide rotation that allows one gear to engage with and turn another gear. Lead-screw shafts are used to build linear actuators. They are used in applications such as disk drives.

4.6.9.2.3 Gearing

Stepper motors obtain a certain gear design called gear trains. In a mechanical system, gear trains are created by mounting gears on a frame. This will allow the teeth, the outer part of the gears, to engage providing a gentle transmission of rotation from one gear to the next. Gear trains increases the torque of the motor so small geared steppers can provide remarkable torque [13].

Previous in this section we discussed how stepper motors are designed with phases. Stepper motors are classified as two or four phase, which states how many groups of coils are designed in the motor. In this paper, we will go more in depth with the two different types of two-phase stepper motors, bipolar and unipolar. They are two basic winding arrangements and are further explained below.

4.6.9.3 Bipolar Stepper Motors

Bipolar stepper motors have only a single winding per phase and a low winding resistance. They require a more complex circuitry to reverse the current in the winding [14]. For this to happen, a H-bridge circuit is conducted. H-bridge circuits enables a voltage to be applied across a load in either direction. Even though bipolar stepper motors are more complicated, they produced more torque than unipolar stepper motors.

4.6.9.4 Unipolar Stepper Motors

Unipolar stepper motors have one winding with a center tap per phase [14]. In certain areas of the winding, the winding is switched on for each direction of the magnetic field. Windings are made quite straightforward with the commutation circuit because the magnetic pole can be reversed without switching the direction of the current. Unipolar stepper motors are quite easy to operate compared to bipolar stepper motors. They are becoming very popular and are commonly the cheapest way to get accurate angular movements.

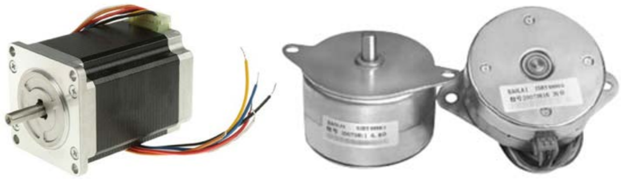


Figure 15: Unipolar Hybrid Stepper Motor (Left) and Permanent Magnetic Stepper Motor (Right)

To conclude, below are some advantages and disadvantages for the stepper motor.

Advantages:

- Precise positioning
- Excellent control of rotational speed
- Maximum torque at low speeds
- No feedback required (this can also be a disadvantage)
- Low cost

Disadvantages:

- Low efficiency – current consumption is independent of load
- Torque drops when speed is increased
- Low output power for size and weight
- Very noisy from moderate to high speeds
- Low accuracy

4.6.10 Motor Specifications and Comparisons

Now that we have considered the different motors that are being considered for this project, it is time to discuss the specifications that are important to know before buying a certain motor. The table below describes the specifications for actuators and a description for them. Also, shown below are questions to indicate the type of motor that should be used depending on what is needed for the robot.

Table 11: Specifications for Actuators

ACTUATORS	
SPECIFICATION	DESCRIPTION
Nominal Voltage	The voltage that corresponds to the motor's highest efficiency. It is imperative to choose a main battery pack that matches closely with the nominal voltage of the motor.
Power Rating	The motor's power can be approximated by using the formula P = IV if it is not listed. The motor's output power can be calculated by using the no load current and nominal voltage. The motor's maximum power can by calculated by using the stall current and nominal voltage.
Stall Torque	The maximum torque a motor can maintain without the shaft rotating. The motor chosen should have no more than 1/4 to 1/3 stall torque.
Stall Current	The amount of current the motor will draw at maximum torque conditions. The value of the current can be very high so make sure the motor controller selected can provide this current. If neither the stall nor nominal current are provided, the current can be estimated by using the motor's power rating and nominal voltage (P = IV).
No Load RPM (revolutions per minute)	How fast the angular velocity of the final output shaft will rotate assuming nothing is connected to it.
Gear Down Ratio	The gear down acts to increase torque and reduce the revolutions per minute. The ratio means the amount of revolutions that the driver gear must take to rotate the driven gear once.

1. **Does the robot need to lift or turn heavy weight?**
- DC Geared Motor
2. **Does the robot need a range of motion limited to 180 degrees?**
- R/C Servo Motor
3. **Does the angle of rotation need to be precise for the robot?**
- Stepper Motor

This concludes the research for the type of motors that will be considered for our design. It is now time to discuss the device that will drive the motors and control its speed and direction.

4.7 Motor Controllers

A motor controller, or sometimes-called motor driver, is an electronic device that works between the microcontroller, power supply, and motors. The microcontroller of a system determines the speed or direction of the motors; however, it cannot run the motors because of a very limited power output [15]. Fortunately, motor controllers can provide the current at the required voltage for the motors. For the motors to operate precisely, the microcontroller and the motor controller must work together. Motor controllers vary in sizes and weight depending on the amount of current that is needed to control the motors.

Before choosing a motor controller, it is important to already know the type of motors that are going to be used for the robot. The reason for this is that certain motor controllers are needed to meet the specifications of certain motors. Below is a description of things to consider for each type of motor that was mentioned in the actuators section of this paper.

4.7.1 Process for Controlling Motors

Each type of motor has a specific way in which it can be controlled and this varies depending on the type of motor. Some motors have a more complex control method while others such as servo motors are relatively simple. These processes will be discussed and analyzed in this section.

4.7.1.1 DC Motor Control

1. Nominal voltage. DC motor controllers usually have a range of voltage in their product description. Make sure that the motor’s nominal voltage meets within the range of voltage that the motor controller can supply.
2. Continuous current. Make sure that the motor controller selected provides a current equal to or greater than the motor’s continuous current. Most motor controller manufacture companies do not specify the motor’s continuous current but instead specify the stall current. If this happens, an easy way of estimating the motor’s continuous current is to take 20% to 25% of the stall current given for the motor.
3. Control method. This includes PWM, R/C, UART, or analogue voltage. Make sure to check the pin types that are available for the microcontroller selected.
4. Single versus dual. Dual DC motor controllers can operate the direction and speed of two identical dc motors. They only have one power output, so controlling motors at different voltages is not possible. Make sure to select the motor controller based on the motor with the most voltage.

4.7.1.2 Servo Motor Control

1. Nominal voltage, continuous currents, and the control method should be considered for this type of motor.
2. Servos. Make sure to check the number of servos that the servo motor controller can control. Most motor controller manufacture companies have this listed in their specifications.

4.7.1.3 Stepper Motor Control

1. Nominal voltage, continuous currents, and the control method should be considered for this type of motor.
2. Unipolar versus Bipolar. Select the stepper motor controller based on the winding arrangement. There are stepper motor controllers that are capable of both.
3. Current per coil. Make sure the stepper motor controller selected can provide the amount of current per coil for the motor.

4.7.2 L298N Dual H-Bridge DC Stepper Motor Driver Controller

This motor controller can drive one 2-phase stepper motor, one 4-phase stepper motor, or two DC motors. It contains an H-Bridge circuit, as mentioned in the name, which is used to control the speed and direction of multiple motors. In addition, the circuit can be controlled by pulse width modulation and can drive a current in either polarity. Below are specifications about this motor controller that are important to know before deciding on the motors.

Specifications:

- Low heat, and outstanding anti-interference performance
- More stable and reliable with a large capacity filter capacitance, and after flow protection diode
- Logical voltage: 5V
- Logical current: 0mA – 36mA
- Drive voltage: 5V-35V
- Max Stall current: 3A
- Continuous current can reach 2A (MAX single bridge)
- Max power: 25W
- When drive power is over 12V, external 5V power supply is available to use

4.7.3 MDD10A Dual Channel DC Motor Driver Controller

This motor controller can operate two brushed DC motors. It supports PWM signal with two types of operations, sign-magnitude and locked-antiphase. Sign-magnitude PWM uses two control signals to control the speed and direction of the motor. The PWM pin is supplied by the PWM signal to control speed while the DIR pin controls the direction. Locked-antiphase PWM uses one control signal to control the speed and direction of the motor. For this operation, the PWM pin is set to logic high while the DIR pin is supplied by the PWM signal. The direction of how the motor will turn depends on how much duty cycle percentage the PWM signal has. Depending on the connection, the motor will turn clockwise or counter-clockwise if the PWM has less or more than 50% duty cycle. If the PWM has exactly 50% duty cycle, the motor will stop running. Below are specifications about this motor controller that are important to know before deciding on the motors.

Specifications:

- NMOS H-Bridge for great efficiency and requires no heat sink
- Sign-magnitude and locked-antiphase PWM operations are supported
- Obtains solid state components for faster response time and minimize tear from mechanical relay
- Push button is on the board to manually operate the motor
- Drive voltage: 5V-25V
- Max PWM frequency is 20kHz (Hertz)
- Max continuous current up to 10A and 30A peak for each channel

4.8 Wheels

Wheels are a circular object that rotates around an axle. They were invented to create movement and drive for many applications. Some of these applications include motor vehicles such as cars, motorcycles, and even tanks that are used in the military. Airplanes have wheels so they can take off and land at their destinations. Wheels are now becoming more capable of movement on different terrains. With the increase of technology, robotics is now an application in which wheels are an important aspect of its design. Fortunately, wheel designs are evolving tremendously for all types of engineering disciplines.

For this project, we are only concerned in wheels that are used for robotic designs. Wheels provide speed, accuracy, and stability for robots. There are several types of wheels that are available for robots. Some of the types that are being considered for this project are standard, orientable, and omnidirectional (multi-directional) wheels. Before choosing a wheel, it is important to select the type of motors first for the robotic design. The reason for this is that motors and wheels must work together to have a successful design. For example, a small motor with low torque will most likely not be able to control a large wheel. Also before choosing a wheel, the texture and width of the wheel, and the types of terrain that is expected for the robot to move on should be considered. For example, smooth wheels will have less traction than rough wheels and thin wheels will turn easier than wide wheels. Below is a description for the different types of wheels that are available for robotic designs.

4.8.1 Standard/Fixed Wheels

Standard and fixed wheels are the most common used in robotic applications. They have two degrees of freedom, and can traverse forward and backward. The center of the wheel is fixed to the robot's frame, also known as the chassis [16]. The angle within the robot's chassis and the wheel is constant. Fixed wheels are usually connected to motors to steer and run the robot.

4.8.2 Orientable Wheels

Orientable wheels are usually used to balance a robot. They are not the best fit of wheels to provide drive to the robot. To hold these wheels in place, they are designed to be mounted to a fork. These wheels branch off to two different types: centered and off-centered oriented wheels.

- **Centered Oriented:** For centered orientable wheels, the vertical axis of the wheel goes through the center of the wheel. All that is needed for this design is a couple of screws to connect the wheel to the chassis.

- **Off-Centered Oriented:** The design for off-centered orientated is a little bit different compared to centered oriented. The vertical axis of the wheel does not go through the center of the wheel, instead it is somewhat off-centered. They can rotate 360 degrees of freedom when a swivel joint is connected between the wheel and the fork [10]. There is a disadvantage that can happen with these swivel wheels and that is flutter. Flutter happens when the wheel is not touching the ground and instead is rotating freely in any direction. However, when the wheel is in contact with the ground, it can be orientated and move in any direction.

4.8.3 Omnidirectional Wheels

Omnidirectional wheels are the best decision for a robot that needs to move in multiple directions. These wheels are designed with small wheels (rollers) that are attached to a larger center wheel. The axis of the small wheels is perpendicular to the axis of the larger center wheel which allows the wheel to rotate parallel to its own axis [16]. Some advantages that these wheels provide is drive, steering, and a low resistance when moving in any direction.

This concludes the discussion of the motion system. We will now transition into the radio-frequency identification system (RFID).

4.9 Radio-Frequency Identification System (RFID)

4.9.1 RFID Analysis

As the name implies RFID utilizes radio waves to identify and monitor objects through the use of RFID tags which contain identification and other data. There are two types of RFID tags: passive and active. Passive RFID tags are powered by the radio waves from the RFID reader while active RFID tags are battery powered. The RFID tags always contain a circuit that processes the modulation of the RF signals as well as an antenna to broadcast the information stored on the tag to the receiver. Utilizing RFID for positional tracking can be difficult and inefficient because one would need a tag for each positional location that needs to be tracked. It is not possible to use RFID for positional navigation as the technology is more suitable for object identification and tracking. RFID tags do have a larger range than most Bluetooth beacons. While RFID tags are relatively cheap the reader itself is expensive. Ultimately, RFID is not suited for the task at hand however it could be implemented in our system if there were multiple autonomous carts and then we could track their location or use the tag to gather statistical data on the usage and paths of the carts in order to further optimize and streamline the movement algorithms and systems. [52].

4.10 Magnetometers

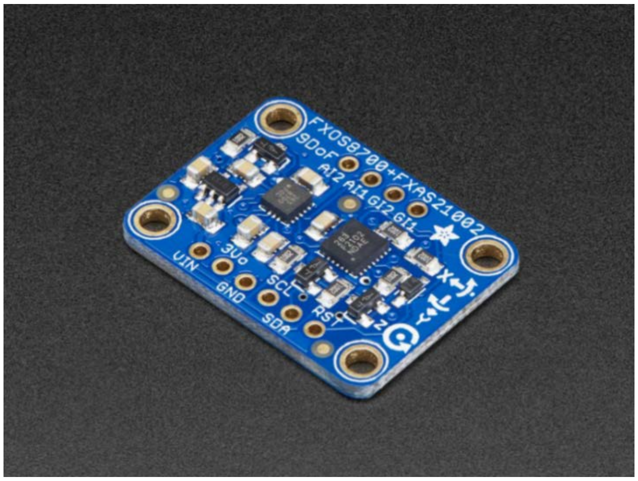


Figure 16: Magnetometers FXAS

Down in section 5.2.1.4, Particle filtering is described as a means for keeping the "pose" or really the position in space and direction in which the FollowBot is facing. While particle filtering is great because it needs no extra hardware, the need to calibrate the initial position and orientation of the FollowBot can be cumbersome and the algorithms to support it are complicated. With these factors in mind, another solution for finding the orientation of the FollowBot could be useful if it could cut down on the complexity and the need for calibrating a starting orientation. This solution could be an integrated magnetometer that is compatible with a micro-controller such as the Arduino. In this section, we will discuss some of the potential Magnetometers.

4.10.1 LSM303DLHC

This is a triple axis accelerometer/magnetometer compass module. Inside are two sensors, one a classic 3-axis accelerometer which can tell you if the FollowBot is pointing up or down through the measurement of gravity. It then has a compact I2C sensor which is used for all information including information on the direction the FollowBot is facing using the strongest magnetic force which will generally be the force coming from magnetic north. There is an Arduino library already written for this chip[20] making it a strong candidate for a project where time is such a limiting factor. The chip is 3.3V max and costs $14.95.

4.10.2 NXP Precision 9DoF breakout

This part actually has multiple chips and sensors, being made from the best sensors to date at Adafruit using the FXOS8700 3-Axis accelerometer and magnetometer, and the FXAS21002 3-axis gyroscope. This chip actually allows for orientation AND motion sensing. The sensor for orientation is powerful for its price because the calculations are actually done on the Arduino. Adafruit provides a version of AHRS that is compatible with Bluetooth LE giving you Euler angles and quaternions. At $14.99 this chip is a steal

Specs:

- $14.99

FXOS8700 3-Axis Accelerometer/Magnetometer

- 2-3.6V Supply
- ±2 g/±4 g/±8 g adjustable acceleration range
- ±1200 µT magnetic sensor range
- Output data rates (ODR) from 1.563 Hz to 800 Hz
- 14-bit ADC resolution for acceleration measurements
- 16-bit ADC resolution for magnetic measurements

FXAS21002 3-Axis Gyroscope

- 2-3.6V Supply
- ±250/500/1000/2000°/s configurable range
- Output Data Rates (ODR) from 12.5 to 800 Hz
- 16-bit digital output resolution
- 192 bytes FIFO buffer (32 X/Y/Z samples)

4.10.3 HMC5883L

The HMC5883L is purely just a magnetometer. It is a triple axis compass module that finds orientation by comparing against the strongest magnetic force, usually magnetic north. It is a 3.3V max chip and uses I2C to communicate making it very easy to use similar to the LSM303DLHC above. It costs only $10 and is a great option if a chip with only a magnetometer is desired. There are many online sources showing examples of using this chip.

Specs:

- I2C interface
- 1-2 degree heading accuracy
- Integrated 12-bit ADC
- 160Hz max data rate
- Range of -8 to +8 Gauss
- $10

4.10.4 Adafruit 9-DOF Accel/Mag/Gyro+Temp Breakout Board – LSM9DS0

This chip is the most powerful of the four chips coming with three different sensors. A classic 3-axis accelerometer which can help with finding orientation in a 3D plane, a 3-axis magnetometer which gives NESW and a 3-axis gyroscope that can track spin and twist.

Specs:

- I2C interface and SPI interface
- 1-2 degree heading accuracy
- $24.99

4.10.5 Magnetometer Conclusions

The primary purpose of the FollowBot is to follow its dedicated user with his/her luggage. For this reason, it will not be necessary for it to ever move at fast speeds. With this as a consideration, although having a gyroscope and accelerometer would help the bot orient faster, a simple magnetometer may suffice. For the price and compatibility, the HMC5883L is

4.11 Voltage Regulators

Voltage regulators are electric devices that are used to regulate a DC or AC voltage so that it maintains a constant value. There are two types of voltage regulars and these are known as linear voltage regulators and switching voltage regulators.

4.11.1 Linear Voltage Regulator [17]

A linear voltage regulator is the most commonly found voltage regulator as it is used in almost every power supply unit and it is incredibly inexpensive and simple to implement. A linear regulator utilizes a voltage-controlled current source to fix the DC voltage that is on the end of the output terminal. Linear voltage regulators can be classified as one of three basic topologies which are the standard regulator, the low dropout regulator, and the quasi-low dropout regulator. These regulator topologies have two significant differences with the first being the dropout voltage of the regulator.

Each topology has a varying dropout voltage which is the smallest voltage drop that sustains the desired output voltage regulation. The low dropout regulator, as the name implies, has the lowest dropout voltage while the NPN Darlington or standard regulator has the largest dropout voltage. A small voltage across the regulator is desirable because this means that the least amount of power is being used. Another significant difference between the regulator topologies are the ground pin currents. The ground pin currents are currents which are necessary to regulated a rated load current. A higher ground pin current would be that the current is being unused because although it is being supplied to the load from the source it essentially does contribute to powering the load. Linear regulators are advantageous in that they provide an output with little noise and are only more efficient than switching regulators as the output voltage approaches the dropout voltage. [17]

4.11.2 Switching Voltage Regulator

Switching voltage regulators are comprised of different topologies however the three most common topologies are step-down (buck), step-up (boost) and step-up/step-down (buck boost). Switching regulators often utilize an inductor, diode, and switch to manipulate the energy transferring from the input node to the output node. Linear regulators can only be used to step-down the input voltage to a lower output voltage therefore limiting their functionality. [53]

4.11.2.1 Advantages of Switching Regulators

When compared to linear regulators switching regulators are advantageous in the fact that they are much more efficient. Switching regulators are efficient except when the load current is very low. Switching regulators are also beneficial in terms of heat management as they stay much cooler but due to the requirement of more complex parts, such as inductors, are significantly more expensive. [53]

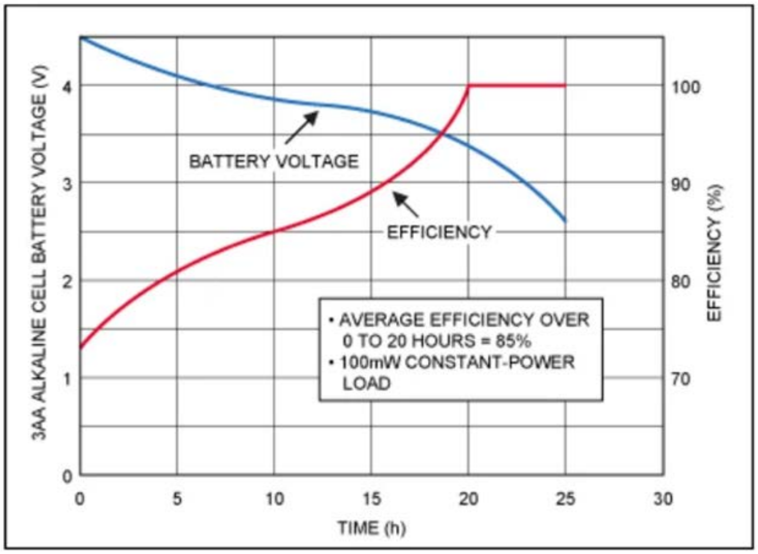


Figure 17: Regulator Efficiency Comparison (Maxim)

Table 12: Regulator Comparisons

Regulator Type	**V78 Switching**	**78xx Linear**	**LM2578**
Efficiency	84-96%	40-60%	~80%
Short Circuit Protection	Yes	Yes	Yes
Ripple Noise (pp)	25 mV	<1 mV	10 mv
Heatsink	None	Yes	None

4.12 Battery Types [18]

The battery type and size was an important aspect of our design as the type and size of battery would not only determine the travel distance but also dramatically alter the weight of the total system. Nickel Cadmium (NiCd), Nickel-Metal Hydride (NiMH), Lead Acid, and Lithium Ion were all considered for usage in the design. It is not only important to consider the capacity and cost of the battery but also the operating temperature and the batteries susceptibility to environmental conditions (temperature, humidity, etc.)

4.12.1 Nickel Cadmium (NiCd)

Nickel Cadmium batteries are incredibly reliable in that they can provide almost 1000 cycles of usage (charge-discharge) if and only if they are operated properly. They also have the lowest cost per cycle of any of the discussed battery types because of how quickly they charge. Nickel Cadmium batteries perform efficiently even when stressed or operated in rigorous conditions however, they have one of the lowest gravimetric energy densities. This means that larger capacity batteries would weigh much more when compared to similar sized batteries of other types.

Nickel Cadmium batteries are also vulnerable to the memory effect which essentially means that the battery must be fully discharged periodically or else the cell plates will undergo crystallization. If the batteries undergo crystallization, then the performance and lifespan of the battery would be greatly degraded. Nickel Cadmium batteries also discharge relatively quickly if left on a shelf but the battery does have one of the fastest charge times. Nickel Cadmium batteries are often used in

4.12.2 Nickel-Metal Hydride (NiMH)

Unlike nickel cadmium batteries, nickel-metal hydride batteries have a very high gravimetric energy density however its performance suffers greatly if the battery is deeply charged and discharged (higher load currents). If the battery is discharged under conditions where the discharge current is higher than the lifespan of the battery would be reduced. For optimum performance, the load currents should be between a fifth to a half of the rated total capacity. These batteries are also extremely sensitive to heat therefore they must be stored in proper environments or else performance degradation will occur. Nickel-metal hydride batteries are also non-toxic unlike the nickel cadmium batteries. The battery toxicity is important if a country or company has strict environmental regulations.

4.12.3 Lead Acid

The lead acid batteries, also known as small sealed lead acid (SLA), are a battery type with an incredibly low gravimetric density; the lowest of all battery types discussed. Lead acid batteries also experience performance degradation in very cold temperatures however, they are incredibly useful in that the batteries do not suffer from the memory effect. This means that the battery can retain charge incredibly well without undergoing any form of damage. It also has a 5% self-discharge rate per month which is extremely low when compared to the Nickel Cadmium and Nickel-Metal Hydride batteries.

Lead acid batteries are commonly used in automobiles due to its low maintenance and efficiency when shelved for a long time. Lead acid batteries should not be fully discharged as every time the battery is discharged the capacity is slightly lowered. The battery should not be stored when fully discharged or else a sulfate may develop on the battery which in turn hinders the recharging process. When compared to other battery types lead acid batteries have a short cycle life (200 to 300 cycles) but is incredibly cost effective.

There are two commonly known types of lead acid batteries which are flooded lead-acid batteries and sealed lead-acid batteries. When being charged lead-acid batteries produce gases (hydrogen and oxygen) which can either be sealed or released. This is where the two types of lead-acid batteries come in. The flooded lead acid batteries have electrolytes on the electrodes of the battery and during electrolysis the hydrogen and oxygen gas are released. Automobile batteries are the most commonly used flooded lead acid batteries. Sealed lead acid batteries are different due to the fact that they only release the oxygen and hydrogen gas if the pressure inside the battery becomes too much. If this limit is not exceeded, then the oxygen gas that is produced is recombined.

4.12.4 Lithium Ion (Li-Ion)

Lithium Ion is perhaps the most popular and well recognized type of battery due to its growing usage in a plethora of electronic devices. The popularity of lithium ion batteries is because it requires little to no maintenance unlike the other battery types. Lithium ion batteries are not subject to the memory effect and do not need to be cycled. The batteries have an incredibly high gravimetric energy density making them suitable for smaller electronic devices such as mobile phones. The cycle life of lithium ion batteries is also incredibly high when compared to lead acid and nickel-metal hydride.

One drawback of lithium ion batteries is that the batteries require a circuit to be built into every battery to ensure that the voltage and discharge current do not exceed twice the rated capacity. As the lithium ion batteries age, they also become vulnerable to a deterioration in their total capacity. Lithium ion batteries one of the most expensive batteries due to the complex manufacturing procedure and costs almost 4 times as much per cycle that the nickel cadmium batteries. Although expensive, the reliability and high energy density is the reason that this is the battery type that we will be using on our autonomous luggage cart.

4.12.5 Lithium Polymer

While Lithium Polymer batteries are incredibly popular they are mentioned due to their low-profile form factor. Lithium polymer batteries utilize a non-conductive film as its electrolyte. Other battery types utilize a “wet” electrolyte or essentially the separator is cover in electrolytes. This dry polymer has an extremely thin geometric profile however it is a poor conductor due to the high internal resistance. To counteract this issue, manufacturers added a gel electrolyte which in turn increases the conductivity of the ions. If size and weight are an issue, then one should strongly consider using lithium polymer batteries however they offer no other advantage over lithium ion batteries as it has a lower gravimetric energy density and is incredibly expensive due to the manufacturing processes not yet being streamlined. This type of battery is not convenient or plausible for our design but is mentioned due to its increasing popularity and the fact that it could very well be a viable option in the years to come.

Table 13: Battery Type Comparison [18]

Battery Type	**NiCd**	**NiMH**	**Lead Acid**	**Lithium Ion**
Charging Time	~1 hour	3 hours	8-16 hours	2-4 hours
Discharge Cycles	1500	300 to 800	500 to 800	500 to 1000
Temperature Range(C.)	-40 to 60	-20 to 60	-20 to 60	-20 to 60
Storage Temperature	-20 to 45	-20 to 45	<10	-20 to 25
Energy Density (Wh/kg)	45-80	60-120	30-50	110-160
Cutoff Voltage (V)	0.9 to 1.05	0.9 to 1.05	0.9	2.5
Cost per Cycle	$0.04	$0.12	$0.10	$0.14
Discharge Rate Per Month*	10%	13.9 to 70.6%	3%	8%

*Discharge Rate is when battery is stored at room temperature

4.13 Types of Charging

One of the most important but often overlooked features of a battery powered device is the type of charger it uses or more specifically the way in which the charging occurs. Charging is a valuable research topic because it can potentially prolong the operational life of the device as well as determining longevity of the battery. This can dramatically impact costs as certain batteries can be incredibly expensive and proper charging allows the user to maximize the battery for better performance and efficiency. A mismatched charger can cause a battery to experience significant wear and can even overheat or explode therefore it is vital to understand how charging occurs and which type of charging is best for each type of battery.

4.13.1 Constant Voltage Charging

Constant voltage charging is typically used for lead acid batteries and lithium ion batteries. These chargers provide the battery with a relatively constant input voltage throughout the charge cycle of the battery as well as a higher initial current. Constant voltage charger is disadvantageous in the fact that they often require longer charging periods to fully charge the battery. The initial input current is high until the power supply reaches a fixed voltage and once this occurs then the current starts to decrease.

4.13.1.1 Float Charging

A float charger is a type of charger that charges the battery at a rate that is close to the self-discharge rate of the battery thus allowing the battery to maintain a full charge without any severe negative impacts. Float charging is useful for cases in which the battery is very rarely discharged such as a backup battery. Most float chargers are in fact constant voltage chargers but have a low operating voltage; this low operating voltage prevents the battery from getting overcharged which in turn prevents any damage from occurring to the cells of the battery. If float charging is utilized for a sealed lead acid battery, then actions should be taken to prevent thermal runaway which is when the heat generated by a battery is much greater than the dissipated heat. This leads to overheating and can cause the battery to melt or catch on fire. Temperature-compensated chargers are utilized in such cases as these chargers vary the float voltage in order to compensate for an increase or change in temperature of the battery.

4.13.2 Constant Current Charging

A constant current charger is often used for nickel-metal hydride batteries and provides a uniform charging current. This current does not vary by temperature or charge percentage but is instead constant throughout the charging process. This allows for battery cells to be evenly charged however the user must take great care to not overcharge the batteries as the batteries could overheat.

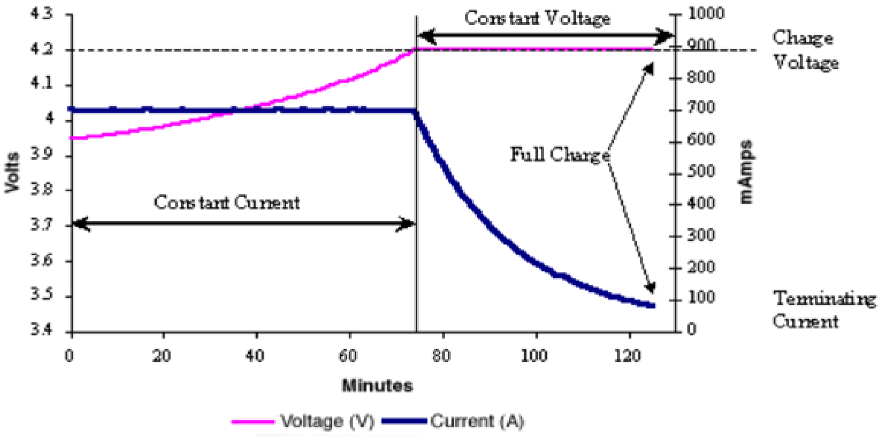


Figure 18: Examples of Constant Current & Constant Voltage Charging [19]

4.14 Final Part Selection & Changes

In Senior Design 2 a few parts were found to not meet our requirement specifications and were there changed. These new parts as well as the different specifications are displayed and shown below.

4.14.1 Motor Selected

The motor selected for the FollowBot was the Magnolara 12V DC 25MM 120RPM Powerful High Torque Motor. They are used in applications such as robotics, household appliances, electric tools, etc. We went with this motor compared to the others mentioned because they have the perfect balance between torque and RPM. Another reason is because the shaft diameter matches with the diameter of the shaft hub which is needed to provide a connection between the motor and the wheel. The specifications about this motor is shown below.

Specifications:

- Spur Gear Motor
- Nominal Voltage: 12V
- No Load RPM: 120RPM
- Stall Current: 1.8A
- Stall Torque: 111 oz-in. = 8 kg-cm.
- Shaft Diameter: 4mm
- Shaft Length: 12mm
- Shaft Type: D-Shaped
- Size: 25D x 52L mm
- Weight: 100g

Figure 19: Magnolara 12V High Torque Motor

4.14.2 Battery Selected

The battery selected for the FollowBot is the Turnigy 2200mAh 3S 20C Li-Po Pack. These batteries have great performance, reliability, and are inexpensive. They are equipped with heavy duty discharge leads that helps reduce resistance and sustain high current loads. Some applications that these batteries are used for are aerobatic flight vehicles and RC vehicles. The specifications about this battery is shown below.

Specifications:

- Minimum Capacity: 2200mAh
- Configuration: 3S1P/ 11.1V / 3Cell
- Constant/Peak Discharge: 20C / 30C
- Weight: 208g

Figure 20: Turnigy 2200mAh Li-Po Pack

4.14.3 Wheel Selection

The FollowBot consists of three wheels. The two front wheels are 6 in. Semi-Solid Tire with Polypropylene Hubs. They are puncture-proof semi-solid rubber tires with strong weather resistant polypropylene inner hubs. They have excellent tract in any terrain. The diameter of the wheel is 6 inches and the weight is 1 lb. The third wheel used is the 3 in. Hard Rubber Light Duty Swivel Caster. This wheel also has solid rubber for durability and smooth movement. It has a rotation of 360 degrees and includes double ball bearings and a zinc plated steel frame for more durability. The diameter of the wheel is 3 inches and the weight is 0.90 lb. The swivel caster wheel's purpose is to provide a more efficient turning mechanism to the vehicle. Both of these wheels provide a maximum working load of 94+ lbs. which exceeds our desired weight. Below are images of the wheels used on the FollowBot.

Figure 21: 6 in. Semi-Solid Tire (left) and the 3 in. Hard Rubber Swivel Caster (right)

Prior to our 3 wheel design for the FollowBot, our design consist of four Dagu Wild Thumper Wheels with four motors. The main reason why these wheels were not used for the FollowBot was because they are filled with foam material. As weight was added on to the FollowBot, these wheels would become flat and lose traction. This would then make it impossible for the FollowBot to turn in any direction. We immediately changed to wheels that provide stronger traction, and changed the entire vehicle design to become more efficient with carrying weight and turning. One image below shows a front view of the FollowBot with dimension values of length, width, and height. The other image shows an above view of the FollowBot.

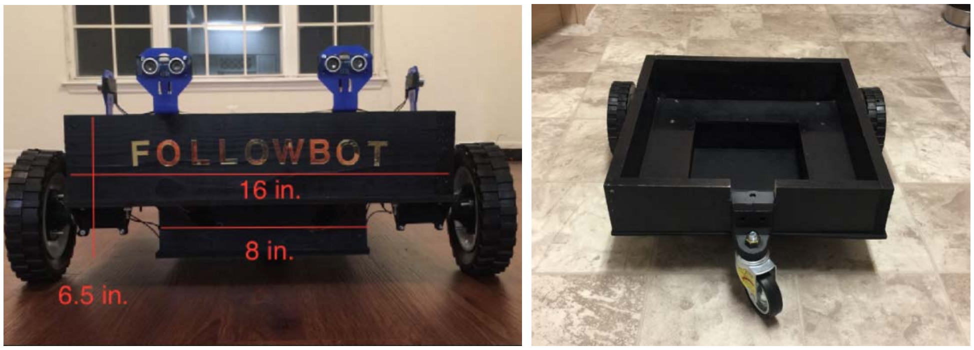


Figure 22: Front view (left) and rear view (right) of the FollowBot

4.15 Motion System Design

The FollowBot consists of two motors that are each controlled independently. They are connected to the output terminals of the L298N Dual H-Bridge Motor Controller. The input pins, PWM signal pins, and the 5V input/output voltage regular are connected to our designed printed circuit board. Below is an image of the motion system connections between the motors, power supply, and the Atmega328.

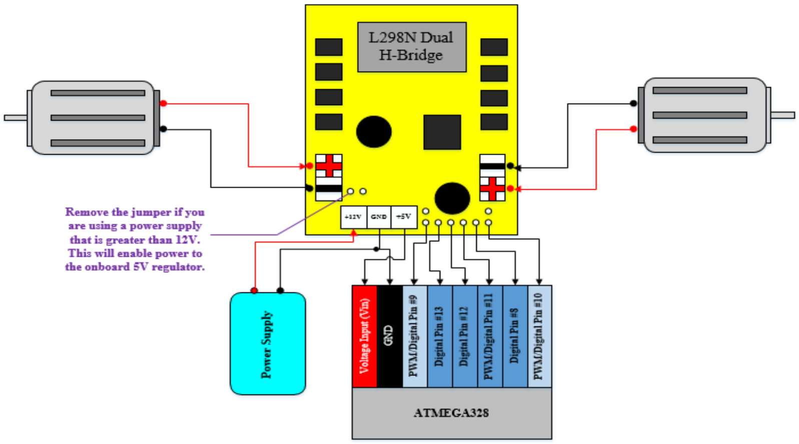


Figure 23: FollowBot's Updated Motion System

5.0 Application to Device (Middleware)

The mobile app will serve as a consumer facing way to communicate with the product. Mobile apps are a great way to control hardware and have seen increased usage in the past few years. Since there is free internet available almost everywhere in the USA, a mobile app will be no issue, especially in airports. Limited functionality can also be designed offline due to the product only using Bluetooth, which is not dependent on internet connectivity. The app itself will focus on (1) connectivity to the hardware, (2) payment for rental of the carts, (3) storing customer information. The phone will connect to the device through Bluetooth: the sensors on the cart will not only achieve trilateration through signals but also use pairing to determine what to follow. The communication itself will be done using simple pairing using Bluetooth libraries that are provided by Android and iOS themselves. The phone will be paired with each unique device which will then be connected to all the triangulation points as well. These receivers will also need to pair, and the only thing the app needs to keep track of in this regard is the FollowBot it is connected to.

For Android, Google provides extensive documentation and great support regarding Bluetooth connection. The Bluetooth network stack is internally supported by the operating system, and this framework allows for applications made for Android to be easily streamlined to connect to any Bluetooth device. The APIs allow for scanning for devices, pairing with devices, transferring data, and managing multiple connections. In the case of the app, the user is not guaranteed to connect with the same FollowBot on multiple occasions, and in most cases, they would not be. This calls for multiple connections being very important on a user's end. Connection through the app must be seamless and efficient while handling multiple connections.

For iOS, the support is very similar and just as extensive. Apple provides a core library with the API and functionality to build a variety of applications that connect with the technology and send/receive data. BLE devices (Bluetooth Low Energy) are a large focus of application technology in the iOS sphere, and thus the framework provided to implement this in an iOS application is expansive. Like Android, iOS Bluetooth framework allows for scanning, pairing, data transfer, and connection management. This is key for development as it allows FollowBot to appeal to the large number of users of both platforms. The documentation provided by Apple is a little clearer but does not have nearly the amount of open source code available as Android does, for obvious reasons.

The whole basis around connecting the mobile phone to the platform relies on a few things: reliable Bluetooth connection without interference, the user having access to a relatively new Android or iOS phone, and being able to pair the two together with ease. The technology that allows the product to do the tracking and trilateration is completely detached from the user however, and is far more reliant on the hardware in place at the establishment it is being used. For smart following of a person, Bluetooth trilateration must occur. Though we have touched on this in other sections, this is the calculation of position of the device relative to the person it is following. This information must constantly be transferred to ensure that the product can follow the target in real time. The other side of this, however, is ultrasonic triangulation. With this solution, the user would simply have to wear a transmitter to allow the device to follow the user, and would be far more suited to areas where Bluetooth transmitters cannot be setup, like a beach.

5.1 Software for Object Collision Detection

The FollowBot heavily revolves around object avoidance and collision detection due to its purpose: maneuvering through crowds while still staying close to its target. The paramount part of our software is thus actually programming this intricate system of avoidance. While detecting collisions might seem simple enough, real time avoidance is a different issue altogether. The whole process can be broken down into three steps: (1) the bot "sees" an object it will collide into on its current path, (2) the bot calculates and decides the best direction to avoid this object while still staying relatively close to the owner, and (3) the bot executes the maneuver to safely avoid the object and continue following its original path. All of these tasks must constantly be completed to show that our demo can successfully be executed in a crowded environment. Haste is an important factor in determining the actual usability of the product, and if these tasks cannot be carried out swiftly, it renders the convenience of the product useless.

For the demo's sake, we can identify a few ways that would be most common in presenting an object to avoid for the software. First and foremost, would be a person – very common in airports – and would require the most attention. The last thing we want to do with our cart is bump into a person and although most people do a good job of weaving through traffic at an airport, this is still a very possible occurrence. To manage people avoidance, we must consider a few things: how close we are to the obstacle, whether we need to stop for the obstacle, and if we can avoid it all together. In an ideal case, a person might just be passing through the negative space between the user of the FollowBot and the bot itself and there would be no slowdown required. The more likely case, however, is someone not noticing an autonomous cart and interrupting its motion, requiring a stop-and-go maneuver. The second most likely obstacle would be a barrier – a wall, a corner, etc. – of some kind that would prevent the bot from getting across. The best way to solve this issue is most likely mimicry.

By analyzing the user's movements to avoid said object, the pathing would simply be followed by the bot and assuming the person is not running into walls constantly, it would be fine. Despite having this "soft" way of managing the obstacles, the "hard" cushion of maneuvering around barriers would be detecting proximity of a non-moving object and once a threshold is reached, stop and find the nearest path that would lead to efficiently following the user. The last common object to avoid would be the user itself. This is fairly easy: since the bot will be mimicking the user's actions, the user stopping should tell the bot to stop with proximity a certain threshold behind. This will also be backed up by the hard cushion of having sensors stop at hard barriers ahead. The software must essentially diagnose every use case where avoidance is needed and constantly avoid it. For the demo, only a few objects may need to be avoided, but in a consumer environment, real-time collisions must actively be managed at all times. In the figure below, a rough representation of object collision as viewed by the software can be seen:

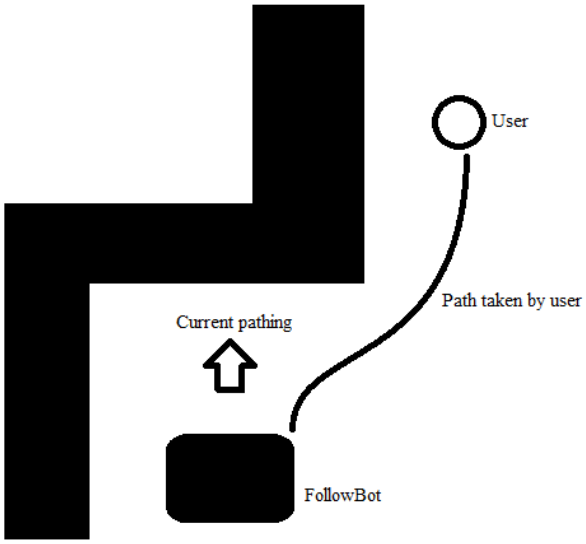


Figure 24: Object Avoidance Example

As seen above, the current pathing of the software instructions the bot to go straight due to being the last action taken by the user at the current location of the bot. This however changes because the user took a different path to avoid the obstacles – the walls of the hallway – which the bot now knows. This will inform the software to make a change in course and follow the pathing set by the user which will, in turn, perform avoid collision. This is what we call "blind avoidance". This is considered blind because the bot does not actually know that a wall is there, it is just performing actions based on the user it is following. It could be taking this pathing regardless of an object being in the way, and it would perform similarly. "Smart" collision avoidance, however, would have still occurred if, for example, the user went into a door in that wall straight ahead and closed the door behind them.

The robot would recognize the pathing as going through the door but would be overridden by the impending collision because it hit the threshold of being too close to an object. In this case, the cart would stop and try to analyze a possible way around this object, and once it tries to maneuver around and then goes out of range of the user, it would simply return and stop, failing to see a feasible way around the barrier. This is exactly how this situation should go: object avoidance should only go so far or else the bot may end up straying too far from the user and causing unnecessary inconvenience in the long run. Convenience is a major factor for this project, not just in overall goal but also minor aspects of usability that may turn the tide on whether someone decides to use a FollowBot.

This factor of convenience thus heavily affects the programming of the software. The avoidance needs to not only be efficient, but convenient. The user should not have to wait a long time for their suitcase to search for an effective path to keep moving and it should rather stop and reset. Object avoidance should follow a basic decision tree. At one particular time when the bot needs to make a decision on what to avoid, it should be prioritizing some decisions over others. There are many considerations that go into this, namely avoiding higher priority collisions such as people first. There may not be as big an issue if the FollowBot hits a wall, but it could cause some trouble if it bumps into another person. The decision tree is shown in the diagram below, which highlights key decisions now.

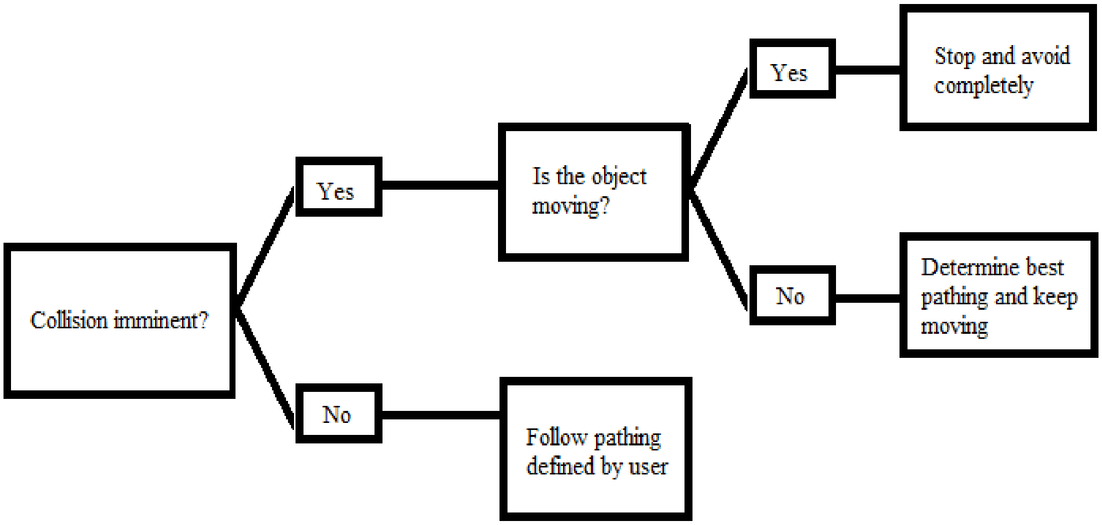

Collision imminent?
Yes
No
Is the object moving?
Follow pathing defined by user
Yes
No
Stop and avoid completely
Determine best pathing and keep moving

Figure 25: Decision Tree for Collision Detection System

This is a basic example of a decision tree on a very broad level, but we feel as if these are the right priorities in terms of avoidance and efficiency. If we forsake efficiency and convenience for the sake of safety, we lose a lot of the appeal of the product. On the other side, however, we cannot compromise the safety of the product to a large degree or it would fail miserably in places such as an airport and for the most part on the consumer market. The software will use conditional cases to determine what course of action is appropriate and will make all of these decisions in real time. This is a crude interpretation of what our software may look like, but should roughly follow these specifications. The FollowBot will use the ultrasonic sensors constantly pinging to determine distance as well as motion of an object in front of it with live tracking, thus enabling it to make these choices. The collision detection will perhaps be the hardest system to implement as the system has a lot of choices to make in a very little amount of time. This means that the decision tree must not only be precise but incredibly quick. One of our major concerns is reducing the amount of latency in our solution. This is one of the major challenges in developing such software, but enables us to do a lot more in terms of scaling to a large market.

6.0 Mobile Application Overview

The primary objective of the mobile application is to seamlessly link to a FollowBot giving its user the ability to control the mode of the bot and view important information about the it and the cargo that it is carrying. Additionally, the mobile application will handle the vital calculations for locating the mobile device. These calculations will be passed to the FollowBot so that is can calculate the direction and distance it needs to go to follow its target device. Although not required, the goal is that the application will follow industry standards for frameworks and languages as well as for mobile software architectural design. In the following sections, we will discuss various languages and their advantages. We will also discuss algorithms for locating devices using several different potential hardware configurations.

6.1 Mobile Application Languages

Before beginning work on any application, it is important to consider the available languages and their advantages for helping deliver the desired product. There are several factors that go into which languages will be used for this mobile application. These factors include overall viability, available resources, the operating systems of potential users' mobile devices, and industry share. Due to limitations on development time, the available frameworks and APIs just after overall viability are the most important factors that will be taken into consideration. The less effort needed to build the framework of the application, the more time the team will have to spend on other aspects of the project. This is a common consideration among many development teams. Next in importance are the operating systems of the intended users. Different software languages have distinct advantages in ease and functionality when developing applications for the various mobile operating systems. Last, we have industry share which is important because of increased available documentation and future hiring viability for team members. The mobile development languages that with be discussed and compared are Java, Objective-C, C++, C#, HTML5, and JavaScript[20, 21]. At the end of the caparisons the chosen languages will be chosen and the potential locators will undergo similar scrutiny.

6.1.1 Java

Java is likely one of the first languages that would come to mind for a seasoned developer. It has been around since 1995 and has been a prevalent language for desktop applications, web application and most importantly mobile applications since its creation. According to VersionEye, a company that tracks open source libraries for software, Java in the language has the most complete and open source resources available [22]. It is also the most searched language on Google. Having all the free mathematical and physics based libraries we need, Java is certainly viable for handling advanced calculations. In addition, there are existing GitHub projects with open source code for trilateration location using Bluetooth[23] and Follow Me GPS. Making Java even more convenient, it integrates flawlessly with JavaScript and HTML, two of the most powerful and frequently use front-end development languages[7]. With regards to its market share and operating systems of its users, Java is the desired language for android applications which is the most widely used operating system among international mobile device owners[8]. The breakdown for market share of operating systems can be seen in figure 2.

It is clear to see that Java has much to offer for our project. Its wide use will make resources plentiful and give the team time to dedicate on hard problems such as effective collision detection and avoidance. Java's large presence also means that companies are searching to hire developers who have experience with it. On top of all that, Java is viewed as the best language to use for a multi-platform application[19]. Although developers are payed less on average as will be discussed in the section on Swift, this language right off the bat has everything this project needs to be built and function efficiently.

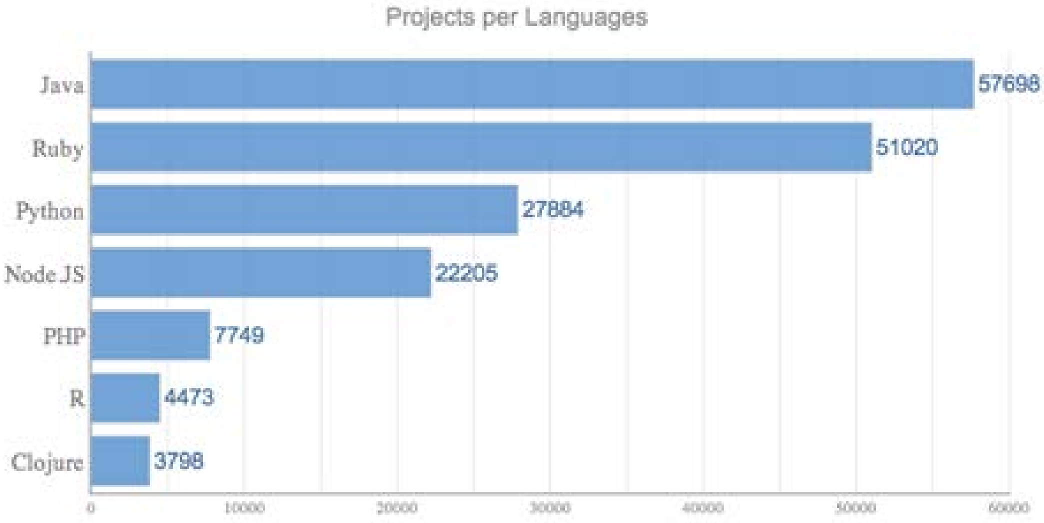


Figure 26: Graphs of Project Distribution per Language

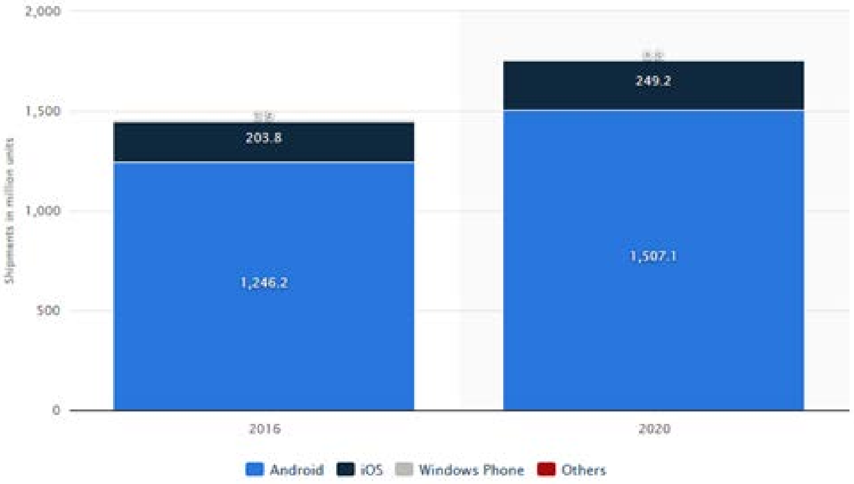


Figure 27: Application Creation Using Native Mobile OS Languages

6.1.2 Objective C

Objective C is an object-oriented language based on C. It also one of two native languages for iOS and is currently ahead of Swift in the number of GitHub repo creations and Stack Overflow questions. This gives it a distinct advantage over swift in readily available internet sources. It also has lots of in-depth how-to resources made available by apple[2]. There are complete GitHub repositories for Bluetooth trilateration[9] as well as repositories for GPS location of phone. Unfortunately, no GitHub or libraries for Follow Me GPS tracking could be found.

As was shown above in figure 2, iOS devices are vastly outnumbered by summary devices making Objective-C less attractive for an application that will be used by sorts of devices at airports. There are]ays to convert Objective-C code to java for use on android such as objc2j[28] and Apportable[29], but the conversion is certainly an increase is effort that needs to be considered with the time constraints on the project. Despite its disadvantages, there is a high job demand for Objective-C and the developers are paid slightly more than their Java counterparts[30].

Overall Objective-C would be a decent choice for developing the app. Its popularity might be decreasing with respect to Swift[20], but iOS isn't going over and even once Swift takes over there will be many important libraries from Objective-C that will still be utilized. It is currently trailing Java however in terms of market share and resources available.

6.1.3 Swift

Swift is widely believed to be a language that can reshape the future and has its name because of it speed[31], It is a C based language and can just about perform function that at C based languages can. In addition, demand for swift developers has increased by over 600% recently making it the language[20]. See Figure 3 for breakdown of average salaries various developers according to indeed.com. Furthermore, Google is considering to switching to Swift as its primary development language which would further improve swift developer demand.

Unfortunately, despite the promising career opportunities and its potential for future market share; Swift is currently has significantly less resources available. This would make developing the mobile application a resource hog which is an important factor to consider as we are not using incredibly powerful boards. The increase in resource usage would also increase the power consumption of the microcontroller which may in turn reduce the amount of power available for the motor and motor controller. This could result in an increase in latency as well as a decrease in the speed of the cart.

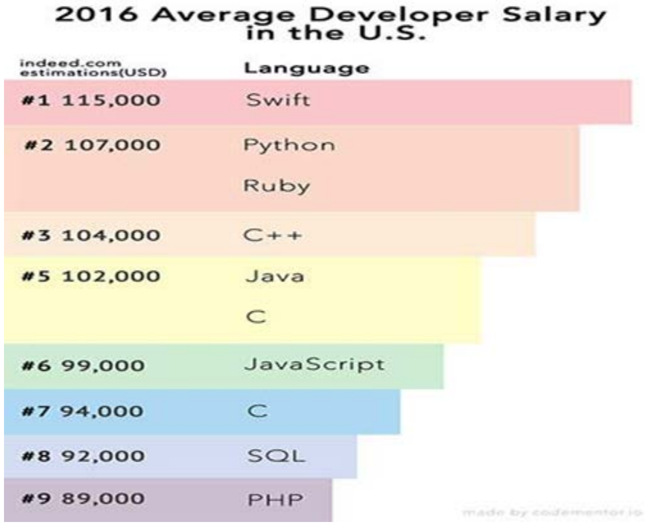


Figure 28: Average Developer Salaries

6.1.4 C# Language

C# is an object-oriented garbage collecting language similar to Java but is based on C. It is a language capable of most anything Swift, C++, and Objective-C are capable with and is a pleasure to use with the power IDE visual studios and developer helper Intellisense. There are Git Repositories for GPS tracking and Bluetooth. C# also is a great language if the target users' mobile devices will have different operating systems. Additionally, it integrates easily with the powerful front-end languages Javascript and HTML5 and has native user interface frameworks for all operating systems. In many ways C# just like a newer version of Java and will quite possibly take over eventually. However, Java still dwarfs C# in market-share and has several times more free, open-source software, APIs, and Frameworks available.

Overall C# is viable option as the primary development language for this project, but does not appear to have any advantage over Java except for having a superior IDE and development helper.

6.1.5 MATLAB

MATLAB is multi-paradigm numerical computing language that is primarily used for mathematical and physics related tasks. Using simulink you can add graphical multi-domain simulations for statics, dynamics, embedded system, you name it. In this project there will be quite a bit of computing that will need to be accomplished quickly such as trilateration of our Bluetooth module on the FollowBot with respect to our Bluetooth beacons. Although, MATLAB is not a language that can stand alone for application development, MATLAB code can be exported into Java packages so that the Java application code can call upon its functions. This could be incredibly handy for all parts of the project that require tedious computation.

6.1.6 Conclusions on Mobile Application Development Languages

Using the research and discussion from the above sections a language needs to be picked based on its overall viability, available resources, the operating systems of potential users' mobile devices, and industry share.

The most important of these factors going into the discussion was overall viability, however after the discussion is clear that all of the options could realistically handle the tasks and calculations that the application needs to handle. So, the new primary factor for choosing a language becomes the resources available for the language. In this category, Java for android beats out all of its competitors by a landslide with its compatability with MATLAB, and broad selection of open source code. Next, when looking at the operating systems of potential users mobile devices, Java is at the top along with C#. Both these languages let you deploy applications for all operating systems with limited effort. Finally, we have industry share which a complicated form of measuring these development languages. On one hand, the demand for iOS based languages like Swift and Objective-C is high and looks like it will only be getting higher with the chance of windows adopting swift as their primary mobile development language. On the other hand, Java for Android is currently the most used language for mobile development. When combining the final factor of average pay, the iOS languages definitely come out on top with respect to market share.

With time being the teams most severe limitation, the choice for primary mobile application development is Java for Android. With thousands of APIs and open-source code available and applicable to our project this is the wisest decision.

6.2 Mobile Based Tracking Implementations

The mobile application will contain most of the algorithms and logic needed to locate the linked FollowBot and mobile device as well as the distance and angle between them. It will then communicate this information to the FollowBot locomotive control system so that it can follow the mobile device while moving around obstacles and avoiding potential collisions. There are several technologies available today that have been used with success for tracking. The most notable for its market being Follow Me GPS which is being widely used in drones for carrying gear and video recording. Bluetooth technology although less prevalent than GPS for tracking, has been implemented quite successfully by mobile device manufacturer, Nokia.

It uses trilateration, or in some cases multilateration to pinpoint the location of a blue tooth device using Bluetooth beacons. Lastly, there is Chronos, Wi-Fi based technology that has recently been introduced by MIT researchers. Chronos is poorly documented and has not been used for any large commercial purposes due to its recent emergence, but is boasts quite a few impressive advantages over its competitors. The most notable of these advantages being its ability to work using a single Wi-Fi access point and its proven ability to locate devices within a decimeter of accuracy. When considering the software side of these location finding technologies we will be using several factors including accuracy within a large building and complication of implementation. In the following sections, each of the mentioned tracking technologies will be explored in-depth.

6.2.1 Bluetooth (Software)

In this section, we will be discussing principles of radio-based indoor location and how it can be implemented using Low Energy Bluetooth technology. Bluetooth is strong option for this project as it is cheap, well documented, and work just about as well inside of an airport as it will anywhere else. We will also be discussing our planned software implementation and the pros and cons of utilizing this system of movement.

6.2.1.1 Inputs Needed for Follow Me Bluetooth

For the FollowBot to locate and navigate towards the desired mobile device, there are several inputs that it need. These inputs are the angle it needs to turn in a direction and distance is needs to move in that direction. In order to know what angle, it needs to turn, it needs to know what direction it is facing as well location of itself and the mobile device. These problems can be solved in many ways, as there are different methods of trilateration, finding distances, and even keeping track of the orientation of the FollowBot. Below we will be discussing these methods in further detail

6.2.1.1.1 Finding Delta Direction

The first problem that needs to be solved to find Delta direction is knowing what direction the FollowBot is currently facing. There are three viable ways we will discuss of achieving this. All three being simple concepts.

1. The first option would be to track the number of times wheels turn and use that information to calculate the current direction of the FollowBot by running the calculations against some calibrated starting direction. This is the very first thought that popped into the team’s mind, but there are quite a few issues with its application. The most major of these issues being the near impossible task of

accounting for bumps or other intrusions that could cause a single rotation of a wheel to have a less or more significant change in vehicle direction. Another issue is the unfavorable need to calibrate a starting direction when there are other methods that don't require it. On top of all that, this implementation would probably be the most complicated and costly on development time and processing power.

2. Another option is to ping for the FollowBot's location for every time interval t. Using the two locations, you could use the it's coordinates to find the exact direction the bot is facing. Although this method would not require calibrating an initial direction, it would cause a delay and force the bot to move in a straight line for whatever the time interval t. Even more unfortunate is that this system would rely on Bluetooth generated coordinates, when Bluetooth is already guaranteed to be the least reliable part of the system.
3. The third and best option would be to simply integrate a compass into our hardware design. This idea was inspired by Hacker Houses's Arduino powered follow me cooler[36]. This would allow the FollowBot to move more fluidly and would cut down significantly on the processing power needed to constantly calculate direction in the above two solutions.

With the direction of the bot being known, you have the angle between it and true north, we call this the heading(a1). If the mobile device(p1) and the FollowBot(p2) are localized correctly using the methods described in the following sections, we will also be able to find the angle between true North and the line that is formed between p1 and p2. This angle is called a bearing(a2). You can now find delta direction as it will be equal to a2 – a1.

6.2.1.1.2 Accurate Localization using Bluetooth

According to the Indian Institute of Technology Madras, the following are the steps that the algorithm will need to accurately localize a Bluetooth device[32].

1. Obtain signal strength indicator values for Bluetooth beacons using some method of inquiry such as variation of RSSI values
2. Use signal strength indicators to generate a rough estimate of distance between Bluetooth device and beacons
3. Use trilateration algorithm to find positions that are in line with the above distances within some decided minimum error range
4. Pass this information to the locomotive control block of project. Potentially use particle filter to use location from previous steps and integrate it with the motion model for the FollowBot. Then repeat step one.

The coming sections will describe in detail how each of these steps can be accomplished. If one correctly we will be able to localize a Bluetooth device with 0.427+-0.229 m accuracy using just three beacons [32]. This level of accuracy is plenty within the acceptable distance for a FollowBot to be within its user.

6.2.1.1 Bluetooth Signal Inquiry

RSSI (Receive Signal Strength Indicator) is the measure of the power level a RF device is receiving from a radio source. GRPR is the power range defined by an upper and lower threshold value. Bluetooth uses something called Adaptive Power Control which increases or decreases the power level if it goes outside of the GRPR thresholds. So, based on whether or not the RSSI is within the limits of the GRPR, the power level is changed accordingly. We can use this built-in infrastructure to inquire RSSI. The method Inquiry_with_RSSI can be used to find exactly what we are looking for, the RSSI. Since we are only using the beacons to gain RSSI information we can set them to Inquiry_Mode to 0x01 which disables Adaptive Power Control and makes the RSSI values more precise.

To gain the values of RSSI needed the following commands need to be sent to the Bluetooth controller. Set inquiry mode equal to 0x01, set parameters for inquiry with rssi(duration, number of responses, discovery mode, devices to discover), request for inquiry with rssi from controller, wait for result, extract RSSI values from result and repeat until the proper number of values are received. Now we have the information we have the RSSI information we need to find the rough distance from the beacon and the device.

In the experiments and projects conducted at the Indian Institution of Technology, RSSI almost completely died out at about 5m, as you can see in figure 18. However, we will be using a far more powerful Bluetooth module on our FollowBot, so are hopes are that we will get readings at quite a larger distance. If this is accomplished, then our FollowBot could viably be used in Airports around the country.

6.2.1.2 Correlation between RSSI and Distance

Distance can be found based on the assumption that RSSI signal strength received by a device will decrease as the distance between the device and the beacon increases. Unfortunately, due to the effects of noise in the signal, an accurate mapping of distance only using a single beacon is not possible[33]. Figure 18 below is borrowed from a test performed by researchers at the Indian Institute of Technology[32]. It shows the RSSI variation as a device moves away from a beacon. They use a class-2 USB dongle 2.5mW/4dBm which has a max range of 10m. The RN4020 we will be using has a max range of 100m, so while applicable, better results could be found in the future.

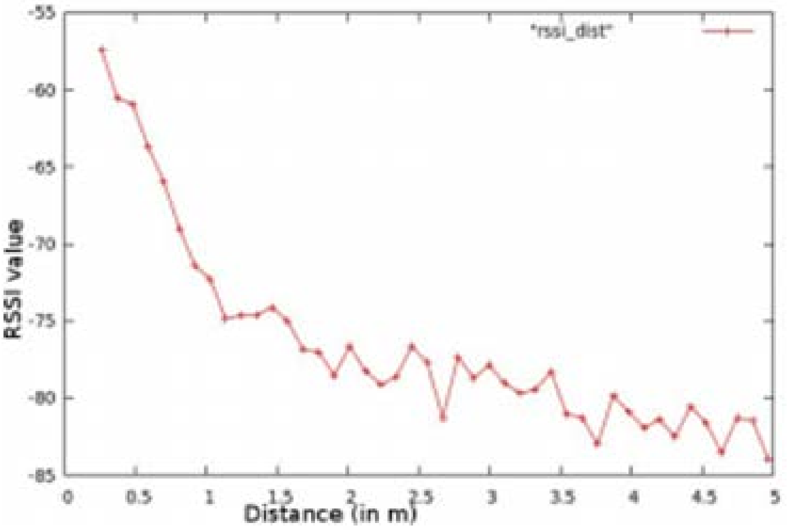


Figure 29: RSSI vs. Distance

Figure 18 proves that there is noticeable and measurable correlation between RSSI and Distance. It also shows the limited range at which this technology will work meaning that on a large scale many beacons would be needed for a marketable solution.

6.2.1.3 RSSI Discrepancy at Same Distance

Unfortunately, unlike what figure 18 seems to show, RSSI is not an easy unit to measure accurately. In fact, in another figure created by Indian Institute of Technology, it is clear to see that although RSSI correlates with distance, at any distance there are many distance triplets (if you have three beacons) that correspond. This means that the trilateration algorithms and particle filtering algorithms must take into account all of the triplets and make a best guess as to which is the most accurate. This adds a large layer of complexity. See Figure 19 below

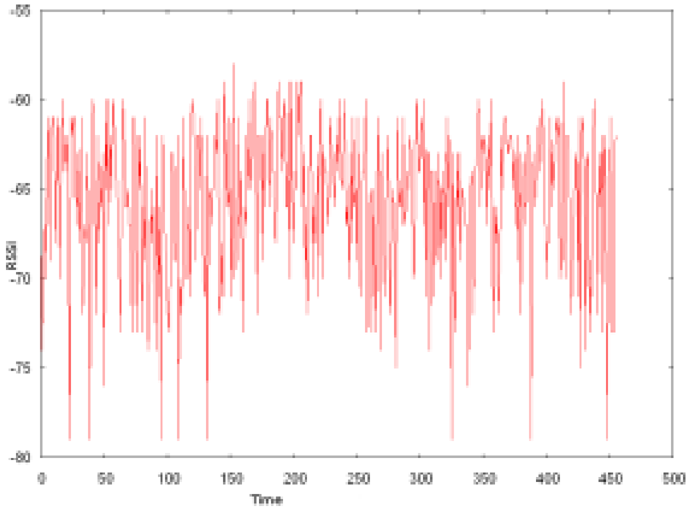


Figure 30: RSSI Variability at 3 meters

6.2.1.4 Trilateration

Trilateration is similar to the more discussed localization algorithm, triangulation, that is used to in GPS. The difference is that trilateration uses the distances between three beacons and a device instead of using angles. Most algorithms create three or more circles with their center the one of the stationary Bluetooth beacons $(x_i, y_{i)}$ and their radius being the distance between the correlating beacon and the device, r_i. The point at which they intersect is likely the location of the target device. Unfortunately, due to noisy environments, the circles created by the various r_i and (x_i, y_i) can sometime intersect in an area or not intersect at all. In these cases, an algorithm must be able to reduce the margin of error despite it being impossible to completely rid of it. These algorithms will be explained further below.

6.2.1.4.1 Over-Estimated System of Equations

The over-estimated system of equations can be explained as A X = B. Variables are explained as followed:

- A is an n X m coefficient matrix
- X is m X 1 variable matrix
- B is n X 1 contant matrix.

If A^{-1} exists and m=n then X will equal $A^{-1}B$. If m<n, then there is an over determined system and the problem can be solved with inverse method X equals A^{+1} B where A^{+1} is defined as the pseudo inverse of A. This will give at least a square eroor when m < n. The system of circle equations is represented as[32]:

$$X = \begin{pmatrix} s \\ x \\ y \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & -2x_1 & -2y_1 \\ 1 & -2x_2 & -2y_2 \\ . & . & . \\ . & . & . \\ 1 & -2x_n & -2y_n \end{pmatrix}$$

$$B = \begin{pmatrix} {r_1}^2 - {x_1}^2 - {y_1}^2 \\ {r_2}^2 - {x_2}^2 - {y_2}^2 \\ . \\ . \\ {r_n}^2 - {x_n}^2 - {y_n}^2 \end{pmatrix}$$

Although promising this method for finding looks promising, it isn't as it relies too heavily on absolute error which we know does not exist in the realm of RSSI from figure 19. The next method will however use this method as a starting point

6.2.1.4 Iterative Over-Estimation Trilateration

This method starts with basic estimate of a center and then a more accurate reference point using an iterative algorithm based on gradient descent to find the point of least error. These are used together and then an algorithm similar to the one explained in the second above is used to find it. The reference points and their matching distances are denoted as (x_i , y_i) and di. The basic point can is denoted as (x_e, y_e). The equation in for error in estimated distance then becomes[32]

$$|f_i| = \left| d_i - \sqrt{(x_i - x_e)^2 + (y_i - y_e)^2} \right|$$

Now using Taylor series approximation you can find the adjustment (Δx, Δy) used for the reference point (x_e, y_e) by using this matrix based equation:

$$\Delta = (B^T B)^{-1} B^T f \quad or \quad \Delta = \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

B being define as:

$$B = \begin{pmatrix} \frac{\partial f_1}{\partial x_e} & \frac{\partial f_1}{\partial y_e} \\ \frac{\partial f_2}{\partial x_e} & \frac{\partial f_2}{\partial y_e} \\ . & . \\ . & . \\ \frac{\partial f_i}{\partial x_e} & \frac{\partial f_i}{\partial y_e} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{(x_1 - x_e)}{\sqrt{(x_1-x_e)^2+(y_1-y_e)^2}} & \frac{(y_1 - y_e)}{\sqrt{(x_1-x_e)^2+(y_1-y_e)^2}} \\ . & . \\ . & . \\ \frac{(x_i - x_e)}{\sqrt{(x_i-x_e)^2+(y_i-y_e)^2}} & \frac{(y_i - y_e)}{\sqrt{(x_i-x_e)^2+(y_i-y_e)^2}} \end{pmatrix}$$

To simplify your new equation with adjustment becomes for $x_{e \text{ and }} y_e$ becomes $x_e = 0.05\Delta x + x_e$ and $y_e = 0.05\Delta y + y_e$. The coefficient is at .05 to keep the convergence from being too large.

Finally, f_i at the beginning of this section deals with absolute error. To decrease this error we want to transform it into an equation that considers fractional error so that all circles are given more equal weightage to all circles. This is vital as small circles (smaller values of r_i) will be more accurate. The fractional error can be expressed as:

$$|f_i| = \left| \frac{d_i - \sqrt{(x_i - x_e)^2 + (y_i - y_e)^2}}{d_i} \right|$$

6.2.1.4.3 Simple 2D Trilateration with Cramers[24]

The equations above are complicated and come from a stream of Masters and PhD students building off of each-others research. We will now discuss a basic algorithm for trilateration.

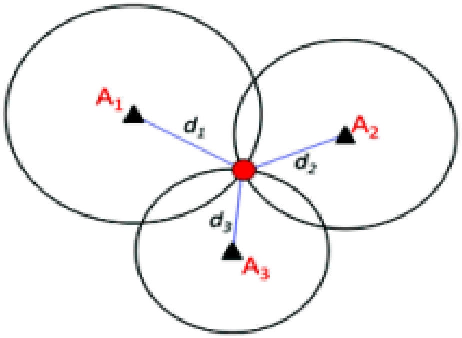


Figure 31: Simple Trilateration

Equations:

$$(x - x_1)^2 + (y - y_1)^2 = d_1^2 \qquad (Equation\ 1)$$

$$(x - x_2)^2 + (y - y_2)^2 = d_2^2 \qquad (Equation\ 2)$$

$$(x - x_3)^2 + (y - y_3)^2 = d_3^2 \qquad (Equation\ 3)$$

Equations 1, 2, and 3 can all be simplified into two linear equations

$$2(x_2 - x_1)x + 2(y_2 - y_1)y = (d_1^2 - d_2^2) - (x_1^2 - x_2^2) - (y_1^2 - y_2^2) \qquad (Equation\ 4)$$

$$2(x_3 - x_1)x + 2(y_3 - y_1)y = (d_1^2 - d_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) \qquad (Equation\ 5)$$

The X and Y coordinates of the center can then be found using Cramer's

$$X = \frac{\begin{vmatrix} (d_1^2 - d_2^2) - (x_1^2 - x_2^2) - (y_1^2 - y_2^2) & 2(y_2 - y_1) \\ (d_1^2 - d_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) & 2(y_3 - y_1) \end{vmatrix}}{\begin{vmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) \end{vmatrix}} \qquad (Equation\ 6)$$

$$Y = \frac{\begin{vmatrix} 2(x_2 - x_1) & (d_1^2 - d_2^2) - (x_1^2 - x_2^2) - (y_1^2 - y_2^2) \\ 2(x_3 - x_1) & (d_1^2 - d_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) \end{vmatrix}}{\begin{vmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) \end{vmatrix}} \qquad (Equation\ 7)$$

6.2.1.5 Particle Filtering

Particle filtering is a complicated method of localization that helps pinpoint the location of an object by merely knowing the approximate distance from the object at several sampling points. For this to work, a virtual grid needs to be created with as many squares as possible. With each roll of the wheel, the FollowBot could get RSSI data from each becon, approximate the distance, and then make an educated guess where each of the beacons could be in relation to the FollowBot using the approximate distance and the acceptable margin of error. Imagine a circle around the FollowBot its radius equal to the approximate distance between the bot and the relevant beacon. On the squares that lie on the parameter of the circle have the highest chance of holding the beacon relative to the FollowBot, the squares that are within the margin of error have a lower chance. So, as the FollowBot moves, squares are given higher or lower weight in their chance of holding a beacon relevant to the starting point of the bot. Over time, the exact location of the beacon with regards to the FollowBot will be known, meaning the location of the Followbot with regards to the beacon is also known.

There are methods we will be exploring: Bayesion Filtering and Monte Carlo Localization.[32]

Bayesian filtering in theory believes the posterior density(likelihood for an object being in any given square) can be computed recursively in two stages, predict and update. It uses all the information possible to find the posterior. The Chapman-Kolmogorov equation is used to for the prediction which is the prior posterior density x_k:

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1}, z_{1:k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1}$$

The update stage is using Bayes rules once again gets predicted error $z_{k:}$

$$p(x_k|z_{1:k}) = \frac{1}{\eta}p(z_k|x_k)p(x_k|z_{1:k-1})$$

Normalizing Constant:

$$\begin{aligned} \eta &= p(z_k|z_{1:k-1}) \\ &= \int p(z_k|x_k)p(x_k|z_{1:k-1}) \end{aligned}$$

Monte Carlo Localization uses the posterior density found above to make a weighted field. Based on each sample the likelihood of an object existing on any given particle is updated. Particles with higher weight have the most likely chance of hosting the object in question.

6.2.1.6 Mobile Resources Available for Bluetooth trilateration

There is a Github project calling lemmingapex[4] with source code for Bluetooth trilateration, or more realistically multilateration. This project using a nonlinear least squares optimizer to hone in and solve the n-D trilateration problem. Using the input of positions of stationary beacons, which will be a piece of cake to find precisely and the radius between those beacons and the receiver on the phone and craft and gives a margin of error.

This project takes advantage of Levenberg-Marquardt algorithm and Apache Comons Math. Implementing it is simple, which will give developers more time to spend on the particle filtering aspect of the project.

With this multilateration problem solved as an optimization problem using non-linear least squares methods, the more beacons you have the more accurate it will be finding the Bluetooth receiver. For three beacons to work, none of them could afford to have degenerative distances due to Bluetooth's poor ability to find accurate distance between a transmitter and a receiver. If all three beacons manage to give decent estimates of distance, then you will have a well-formed R^2 ellipse solution. With 4 good distance estimates you would have an even better R^3 ellipsoid solution. Figure 19 below shows a visual of how 5 beacons would locate the receive

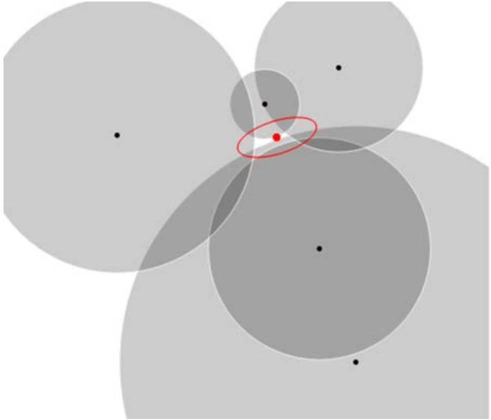


Figure 32: Multilateration

This project has a Maven dependancy for Trilateration. Thankfully, Maven is free and implementing the dependency is as simple as the follow:

```
<dependency>
        <groupId>com.lemmingapex.trilateration</groupId>
        <artifactId>trilateration</artifactId>
        <version>1.0.2</version>
</dependency>
```

And using Gradle adding the dependency by using the following:

```
dependencies {
  compile 'com.lemmingapex.trilateration:trilateration:1.0.2'
}
```

6.2.1.7 RN4020 Resources Available for Bluetooth trilateration

Unlike for mobile devices, there is no API that could be located to trilaterate the location of our FollowBot carrying an RN4020. However, thankfully there is framework in place to make the process easier.

The RN4020 comes with a built in function F that has two hex values as inputs. The first input is the interval for how often to ping out to peripheral Bluetooth devices and second is the window to scan and find important information about the devices. Two of the pieces of information that are sent back are the BTName and the RSSI. Using a map between devices BTName and their fixed location will allow RSSI feeds to be correllated with the correct device. We will then use the RSSI feeds to get a rough estimated of the distance between the Bluetooth beacons and the RN4020.

Another resource that will make the trilateration easier is MATLAB. MATLAB will greatly simplify the code for computing distance via RSSI correlation and will also easily compute the location of the RN4020 using that information. Furthermore, unit tests can be used to test the MATLAB code prior to exporting it into a java package to be used in the project.

6.2.2 Follow Me GPS

In this section, we will be discussing GPS based tracking using the Follow Me Framework and how it can be implemented. Follow Me GPS solutions are being used widespread to have drones carry equipment[17] and to autonomously video record athletes and others who require similar problem solving. Hacker House recently made a cooler that follows people quite accurately and made their build and source code completely available.[18]

6.2.2.1 Inputs Needed for Follow Me GPS

In order to accurately locate and navigate towards a mobile device, the FollowBot is going to need the angle it needs to turn towards the mobile device and the distance it needs to travel. In order for this to happen, the Followbot must always know what direction it is facing, as well as it and the mobile devices locations. The methodology for this would be the same as in Section 5.2.1.1.1.

6.2.2.1.1 GPS Localization

GPS coordinates are found using extremely high frequency radio-waves being beamed out from satellites orbiting the earth. Using the angles between the GPS device and three satellites, the location of the device can be triangulated. Thankfully, this is a widely frequently used localization method, so there are many Java APIs available to handle all the heavy lifting on the development side. The one we considered using will be discussed below in 5.2.2.2.

6.2.2.2 Resources Available for GPS Software Development

Out of all of the radio-wave options, GPS by far has the most options for APIs and free open source code available for mobile applications. In one of the examples, Flytbase, requires only a phone, an Arduino, and a GPS module and contains the following functionality:

- Marker showing location of vehicle with gps module
- Marker showing location of phone
- "Follow on" button that activates the vehicle to start following the phone
- "Follow of" to have the vehicle stay in place
- If you have a drone, it has an automated landing feature
- Desired distance to follow from settings

With all of this functionality coming bundled, the team would have a vast amount of time to focus on course tracking and collision avoidance.

6.2.3 Chronos Wi-fi

Chronos Wi-Fi is the newest and arguably the highest most sophisticated solution for indoors tracking. Developed by at MIT, it is the first system to allow a Wi-Fi card to locate another Wi-Fi card within 10 centimeters of accuracy. This solution is handled completely by the two Wi-Fi cards with no additional hardware needed. The technology has already been used by the MIT team to smart home occupancy, Wi-Fi geofencing, and a personal drone that flies at a safe distance from its user. These are things that before Chronos could only be accomplished by Wi-Fi if there were 3 if not more Wi-Fi routers.

The MIT's team was inspired to undertake this project because of the knowledge that there are systems that use specialized ultra-wideband radios that span multiple GHz that can accurately localize a device because of they can easy measure time-of-flight with sub-nanosecond accuracy. Accurate time-of-flight measurement means accurate localization of target device, in this case being within 10 centimeters[34]. In the following we sections we will discuss the components of Chronos technology as explained in MIT teams technical report[15].

6.2.3.1 Chronos Major Algorithmic challenges

In the Chronos technical paper[33], they explain the major issues that need to be overcome the weaknesses of one-on-one Wi-Fi localization are resolving phase offsets, eliminating packet detection delay, and combatting multipath.

6.2.3.1.1 Resolving Phase Offsets

Chronos must combine channel state information from individual packets being transmitted in different frequencies starting at different times in order to simulate the required wideband radio signal. This switching of frequency bands leads to an initial phase offset as the hardware switches to a new frequency. These phase-offsets are random in nature making the software needed to perform the task of getting an accurate time of flight for the positioning quite difficult.

6.2.3.1.2 Eliminating Packet Detection Delay:

In addition to the random phase-offsets, the software will need to account for the delay in hardware detecting packets. This detection delay is often many times hundreds of time longer than the time-of-flight itself. Furthermore, these detection delays are also random in nature like the phase-offsets that occur due to the changing hardware frequency. Chronos must be able to piece accurate time-of-flight back together despite all the issues

6.2.3.1.3 Multipath

Multipath is an issue that occurs from radio based signals ricocheting off the walls of an indoor building, causing receivers to pick up on several of the same signal with different time-of-flight for each signal that is purposely transmitted. In order to maintain accurate time-of-flight, Chronos must maintain the integrity of the main time-of-flight by filtering out these extra signals.

6.2.3.2 Algorithms for Chronos to function

The following sections will go into depth about the algorithmic problems that need to solved in order for Chronos to work properly

6.2.3.3 Chronos Resources

The source code for MIT's Chronos personal drone is unfortunately not available. This would mean our software team would have to start from scratch trying to implement the very complex algorithms. While the MIT team does provide some of the algorithms and all of the math needed, implementing the algorithms so that they run at similar speeds would take months of development work on its own.

6.2.3.4 Conclusion on Localization Methodology

Deciding on the radio based methodology for localization for this project is difficult as each of methodologies have their perks and their short fallings. Without a methodology standing out as a clear winner, the various strengths and weaknesses of each must be summarized for clarity

6.2.3.4.1 Bluetooth Strengths

1. Available APIs
2. Simple Development
3. Accurate inside as much as outside
4. AltBeacons work well with Java

6.2.3.4.2 Bluetooth Weaknesses

1. Incredibly short distance effectiveness
2. The best implementations still have large margin error(atleast a meter)
3. Needs 4+ beacons giving semi accurate estimations of distance to localize device

6.2.3.4.3 Follow Me GPS Strengths

1. Available APIs
2. Simple Development
3. Long range effectiveness
4. No need for any beacons

6.2.3.4.4 Follow Me GPS Weaknesses

1. Will not work inside buildings with thick walls
2. Will not work in areas that have poor gps signal
3. Fairly large margin of error even in places would strong gps signal

6.2.3.4.5 Chronos Wi-Fi Strengths

1. Most advanced and cutting edge technology discussed
2. Accuracy within a decimeter using a single Wi-Fi access point
3. High speed transfer of information
4. A drone that flies a safe distance from owner has been implemented

6.2.3.4.5 Chronos Wi-Fi Weaknesses

1. There is no open source code, and the development work would drain more resources then the team can likely afford.

The intention of the FollowBot is to have it carry luggage within an airport which makes GPS inabilities to function properly inside large building a deal breaker. Unfortunately, the Chronos Wi-Fis complete lack of open source code also makes it an ineligible methodology, as the development work to recreate it would be immense. There are other options of using multiple Wi-Fi access points to perform indoor localization, but the price for the routers would be far too great. This leaves Bluetooth as the only viable option for a radio signal based localization method. Bluetooth's short range and need for 4, 5, even 6 beacons is quite frustrating. However, on the bright side, this might mean there is room for improvement via software.

6.3 Communication between Arduino and Mobile Device

In order for the project to execute correctly, the mobile application must be able to connect to the Arduino and transfer information about its current location as well as other information used for commands. This process should be streamline so that the information flow is incredibly quick which will reduce the latency of the application. The communication between the Arduino and the mobile device is important as it provides the operator of the cart with information such as the weight as well as the limit of the

6.4 Microcontroller Programming

To successfully create our platform, efficient and bug-free code sent to the microcontroller is one of our key foundations that our project rests on. We are still between the MSP 432 and the Arduino Uno as far as microcontrollers go, so both will be discussed in terms of programming. Programming an MCU comes with quite a few challenges that can show up later in production. UCF has provided us with a decent amount of knowledge when it comes to programming chip-level functions in C. The language is by far the most important and widely used when it comes to calling pins and functions directly connected to an MCU. A microcontroller's job can change but will always need to function in order for the product to not break apart. Overall, a microcontroller is quite simple: it controls inputs and outputs but most of the functionality is quite diverse. One microcontroller of the same type could be doing a drastically different job than the same elsewhere. A microcontroller is very basic: it does not know what its role is or what it's supposed to do unless you explicitly state it. Each microcontroller also has its own options for programming, which gives us a lot of choices for this aspect. Since the short list of microcontrollers is now just the Texas Instruments MSP432 and the Arduino Uno, the options are limited slightly but still not narrow enough to highlight one clear option.

6.4.1 Texas Instruments MSP 432

To begin, the Texas Instruments MSP432 is an ultra-low power microcontroller unit that offers a lot of benefits as well as tradeoffs. The chip itself is very cheap and efficient, which allows us to be able to experiment more and not worry about breaking it as much. It is possible to brick an MCU by flashing bad code and we would rather not risk it with an expensive MCU. For the TI microcontroller, we have two major options for development: TI Code Composer Studio and Energia. These options are the two leading competitors for writing code on the MSP430 family and are largely used around the world.

Figure 33: A MSP432 Launchpad

6.4.1.1 Energia Rapid Prototyping Platform

To begin, Energia is not technically considered a tool meant to write production code. It is intended to be a prototyping platform used by mostly hobbyists to quickly and efficiently write code and deploy to their microcontroller for on-the-fly testing. Energia is immensely helpful, however, in bringing the main Arduino framework over to the MSP430 family, allowing developers to write code easily even if they are coming from only using Arduinos. Although Energia is intended as a free tool to help rapid prototype development, it should not be underestimated in raw power. Energia can be used to not only prototype but also develop entire code for very complex projects. It can deploy these changes fast and effectively while also reflecting necessary changes or errors to the developer. This is one of the most popular options for this very reason, and a true consideration in our search for a MCU code development platform.

Advantages:

- Rapid prototyping allowing for maximum tests and builds
- Very familiar Arduino framework which is easy to use
- Many examples available with our microcontroller being one of the choices supported

Disadvantages:

- Not supported by TI itself, and lacks a lot of the support for most of the MCU's in the family
- Does not allow for plenty of the inbuilt functionality to be changed that the Arduino framework may not offer in its base itself
- Not all TI libraries are supported

6.4.1.2 Texas Instruments Code Composer Studio

Code Composer Studio has reigned as the most popular IDE for the TI code family for as long as it has existed, and not without reason. The company provides this powerful tool for users to write code on their platforms, largely because it is so effective. The IDE writes C code but marks it up and makes it very efficient to determine errors and debug the code due to its native support of the device. Code Composer Studio also offers a huge advantage due to its versatility: it can natively program the whole family of MSP430 MCU's whereas the Energia has limited support based on what is added by the creators. Code Composer also offers the ability to debug and test code similar to what most developers are used to in other C IDEs. The library offered by Texas Instruments in CCS is also very powerful compared to the simple framework ported over from Arduino by the developers. Energia is renowned for this, but it will still never compare to a completely native IDE that the company itself provides. Energia still heavily functions if the user attaches extension libraries, but it's still not as simple as CCS in this regard. Another bonus that Code Composer Studio holds is the fact that it makes you program every part and peripheral attached to the MCU.

Advantages:

- Texas Instruments creates CCS as well as the MSP432, having company support
- Natively supports all MSP430 family MCU's
- Familiarity of using universal C code as used in TI's examples

Disadvantages:

- Can cause trouble if explicit errors exist due to low time to production
- Error reporting is not that great – slows down development

6.4.2 Arduino Uno

The Arduino Uno, as previously discussed, is a very powerful MCU capable of every function and use our FollowBot would need. One of the key benefits it brings is the ease of development we possess when using the Uno. The MCU itself is not very cheap, but has so much background documentation and codebase around it that flashing and installing software has never been easier. This benefit can severely outweigh some of the MSP432's benefits due to the time to publish code being faster and the overall software lifecycle being quicker. The Arduino Uno also has two major options for development being the Arduino IDE and Arduino for Visual Studio. The Arduino IDE is, of course, offered by the creator of the board while Visual Studio is a very large IDE that supports development for the platform.

Figure 34: An Arduino Uno Board

6.4.2.1 Arduino IDE

The software made by Arduino itself is by far the most popular IDE for the product for plenty of reasons. The reason that trumps all is, of course, support from the brand creator itself. The tremendous amount of documentation and example code available for Arduino IDE sets it apart from its “competition”. This aside, the IDE itself is actually quite powerful, visually pleasing, and easy to understand. Arduino makes development easy on the user due to their focus on hobbyists and their like in making hardware development understandable and useful. Running Arduino code can be as simple as taking a snippet of written code, entering it into a file in the IDE, compiling, and running the board to do whatever you wish. Rapid prototyping is always a factor as you never have to worry about re-flashing hardware. Arduino IDE is simple, efficient, and effective, which is the obvious reason it is the most popular for the microcontroller.

Advantages:

- Powerful IDE which natively supports and powers Arduino devices
- Simple and aesthetic yet effective
- Community resources are tremendous

Disadvantages:

- Lacks code prediction, error highlighting, skipping to code definition
- Is a very simple IDE, almost a text editor with a compiler

6.4.2.2 Arduino for Visual Studio

Visual Studio has long been considered one of the best IDEs of all time when it comes to general programming. Microsoft executes all of their products with state-of-the-art production quality and this shows above most companies that are not quite of the scale in developing development tools. One of the large bonuses Visual Studio holds is familiarity: we have all used it at some point and it holds one of the most recognizable user interfaces on the web. The features include code prediction, error highlighting, code definition skipping, all which Arduino IDE lacks. Being able to develop for a device in an IDE you are already familiar with ascends its usefulness to the next level. Although the support and documentation Arduino provides for its own board through its IDE is immense, Arduino for Visual Studio is just such a strong tool due to its potential and ease of use. Arduino for Visual Studio seems most popular for a production build of code, with version management and great debugging all built in whereas Arduino IDE seems like a most worthy option for prototyping and execution.

Advantages:

- Supported by one of the largest development companies in the world
- Very familiar IDE and has long been at the top of the list
- Quick publishing and version management

Disadvantages:

- Not native to Arduino
- Lacks support of some functions that Arduino IDE may have due to regular updates

6.5 Phone to FollowBot Connection

One of the main functionalities we have outlined regarding the FollowBot of course is the Bluetooth pairing between the mobile phone of the user and the device. This is important due to the very business model we aim to market to. We want one FollowBot to work for many people as they come and go and we wish for people to seamlessly pair and unpair when they need to use one. This will be achieved through a Bluetooth module connecting the Arduino Uno board to the onboard Bluetooth module in most smartphone devices. Since we are making our app cross-platform and available to use on either Android or iOS, we must make sure that we are able to maintain these pairs without interference on either device and tie this to user authentication on a remote server. Security is only a slight factor here as the location of the device is always tracked and mishaps can easily be located. The Bluetooth module we decided to go with was the RN4020 which will provide us with a range of 100m to be able to operate within, and connections to phones will be easy to maintain within this distance. The pairing can be done in development this way as Bluetooth connection APIs are very common and easy to find on the internet. We will be able to simply ask a user to pair their device to the FollowBot and maintain connection until they need to leave or return the bot. This step will be done after initial authentication, and the user’s account will keep track of all of the FollowBots that have been used and display any existing rentals. Ideally, a user should be able to connect and catch all their flights while having FollowBots at each airport ready to go for maximum convenience. We believe this could be achieved in the future with strategic marketing and analysis.

The link between the mobile phone and the FollowBot is very important, and as developers, we must protect this. A device always possesses the potential to lose Bluetooth connection at any time for various reasons: dead battery, interference, phone crash/problems. Whatever the problem may be, the bot must have a default state and absolutely must remain functional in terms of object avoidance and collision while it is in this unconnected state. This can be resolved by setting a timeout for the FollowBot, and basically stay in place until the user corrects their Bluetooth issue, or if the issue is on the bot’s end, allow them to reboot.

The timeout will cause the product to stay in place until a link is re-introduced to the same phone or account as earlier and immediately start the follow process again. We must understand that the security of this connection is important because if we lose it, we lose the premise of the convenience all together. The phone and device must always stay connected, and our choices in material will allow this to happen at a maximum. Although this is the thought process for our airport design, we believe consumer bots will work differently. Consumer grade FollowBots will be aimed at also having a Bluetooth connection between devices, but we agree that an offline mode is more appropriate in a home setting and rules regarding connections can be a little more relaxed. An owner will be established at the first connection, and the link will not need to be refreshed every time the bot is activated.

As we can see, the mobile to FollowBot connection is key in determining success in most markets. Most hardware devices that are considered “smart” in today’s market rely on mobile connection and rightfully so. Most people in the world have access to smartphones and this allows us to take advantage of gaining users through our platform. Signing up just to use a luggage cart might seem like a drag but not having to carry a suitcase around an airport might just succeed that. Even better, if a user purchases a FollowBot for home, having a permanently connected bot with a 100m range in a house can be very convenient, appearing at the touch of a button. The convenience of having a mobile app severely outweighs the fallbacks, and as developers it allows for a huge deal of product maintenance and ability to push software updates remotely. Being able to monitor usage and spot defects is also huge for furthering development, and this cannot be done without remote connecting requiring internet. The Bluetooth connection between mobile devices and FollowBots will make the product even better.

6.6 Mobile and Microcontroller UML Diagrams

The team decided that we would use hardware that is highly compatible with Java, the most popular of object-oriented development languages. When tackling large object-oriented projects, software engineers generally start out using UML (Unified Modeling Language) diagrams before starting on the actual development work. The reason for this is that there are clear goals for functionality, understanding of potential blockers, and interaction between parts of the system. The first diagram we will be using is called a Use Case Diagram. The Use Case Diagram is a high-level representation of the interaction between parts of the system. It clarifies which parts of the system perform which actions, as well as the order in which the needed and/or optional actions must be executed.

After that, the Sequence Diagram shows interactions and how objects work with each other and in what order. The important factor here is that is shows interactions with respect to time. The last UML pictorial will be the Class Diagram. This diagram lists out the classes, abstract classes, and interfaces needed to achieve the project. Methods are expressed showing the type returned and inputs needed. The Class Diagram will give the framework needed to effectively create manageable tickets for the software team to systematically develop the final product. These diagrams are of utmost importance to the success of the software team and understanding of other team members who do not have backgrounds in software. They will be the backbone of the software design process.

6.6.1 Use Case Diagram

In Figure 30 below the two stick figures are the users performing the actions. The ovals have the actions listed inside of them. Include means that the action must occur for the action pointing towards it to occur. Extend means that the action can occur and is related to the action that is pointing towards it. A solid black line shows the connection between a primary action and a user.

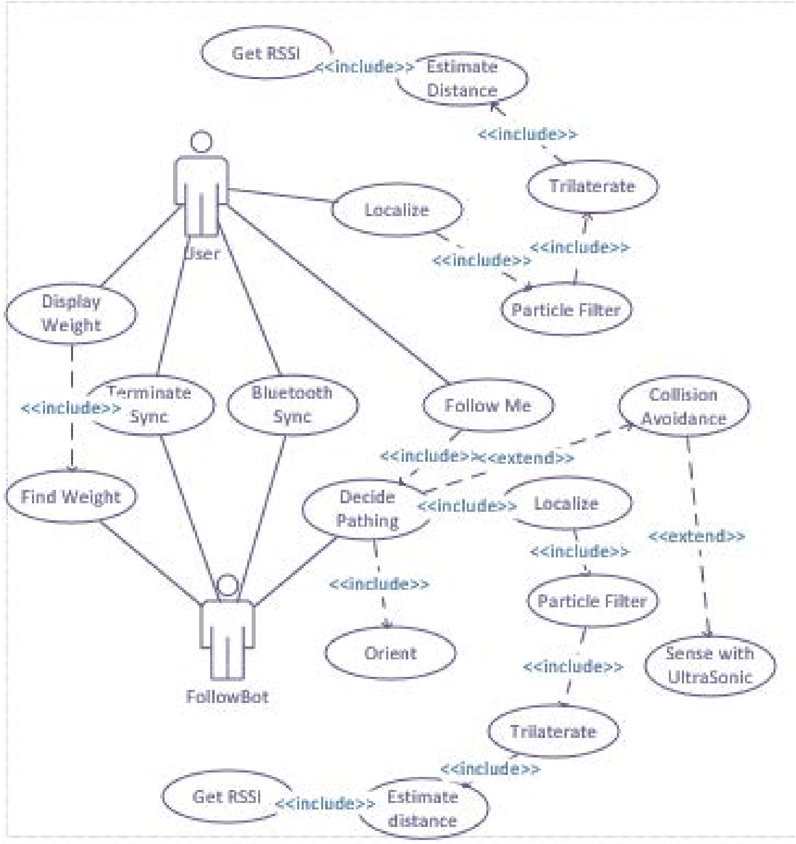


Figure 35: User Case Diagram

6.6.2 Sequence Diagram – Bluetooth

In the Sequence diagram below, the little man represents the User. The other three rectangles are other objects, or sections of the Bluetooth transfer of information. The Arrowed lines are a series of requests and responses from between the different parts.

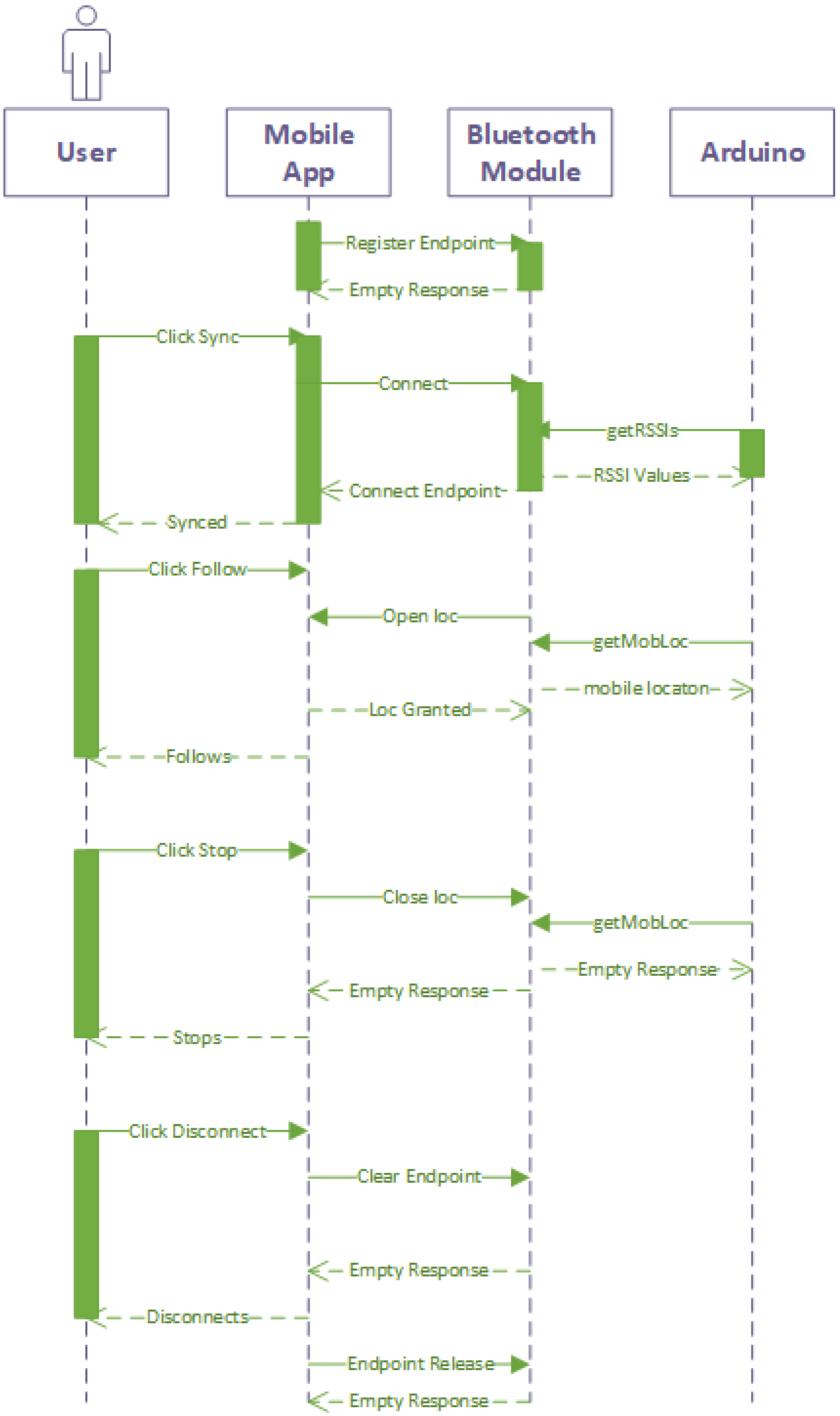


Figure 36: Bluetooth Sequence Diagram

7.0 Design Constraints & Standards

As engineers, it is vital to understand the constraints that the team has for the project as these can impact one's choice of specific parts or lead to delays if the constraints are not met. Operating within limitations is a vital part of being an engineer in the real world as you must meet certain criteria such as physical design limitations (size, weight, shape, etc.) or even environmental limitations (use of non-toxic materials, rugged for outdoor environments). These constraints can even be software related such as a certain response time or with a certain programming language. The most considered constraints pertain to the pricing of your product as the price will determine how popular your product is as well as what audience the product is designed for (consumer, enthusiast, commercial, etc.). For example, enthusiast and commercial products tend to be significantly more expensive than regular consumer products due to their higher performance as well as more rugged designs.

7.1 Standards

Standards are essentially guidelines that manufacturers can follow in order to increase efficiency and safety as they are compiled and issued by non-profit organizations which are comprised of experts. Standards serve to streamline the engineering of a product as well as ensuring that the product can easily meld into the ecosystem of existing products. Standards however, are not regulations and therefore are not mandatory they instead act as a common criterion for producers. By standardizing processes, procedures, or ideas products become consistent and uniform across states as well as countries. [38]

7.1.1 Physical Material & Size Constraints

Our autonomous cart is meant to be operated in an airport or an extremely busy environment therefore the cart should be extremely compact. The cart should be compact enough to allow it to pass into elevator doors and should have wheels large enough for it to go over small doorstops and bumps. There should be no sharp edges on the cart as this will ensure that if something or someone bumps into the cart then they will not be injured. The material should be smooth to avoid damaging the packages or luggage that are being carried but should have grooves or notches to prevent the luggage from sliding around. The material should be durable as luggage will be coming on and off the cart at a relatively quick pace.

Durability of the materials and electronic components are a key factor in the design of this cart as it needs to be rugged and can handle continuous usage without a break. The most vital component that should be considered is the battery as the durability of the battery will in turn determine the feasibility of the cart as more frequent battery changes would increase the overall price of the cart. It is important to choose a battery that has both a long total lifespan as well as a relatively high capacity.

The luggage cart should be able to operate two to three hours continuously on a single charge. Another important component which should be screened or tested for durability is the motors for the wheels as these will be constantly stopping, turning, and adjusting to ensure the cart is on the proper course without colliding with anything. A faulty or failing motor can severely injure a person. Even if the collision detection is working properly this means nothing if the cart cannot readjust itself to avoid the obstacle.

7.1.2 Financial Constraints

The most important constraints are the financial constraints as the creators of this project are only college students therefore the parts that should be used should be relatively cheap and easy to obtain. When determining the parts that we were going to use the cost became an extremely crucial aspect of the screening process. This was because we would have to purchase three of that component to ensure that if one failed we would have a backup part to replace it. In order to limit financial cost, we were considering using sanded wood as our prototype material as it is both sturdy and cheap. However, for the actual product we were considering using a synthetic material such as polybutylene succinate (PBS) or another similar plastic.

When operating with strict financial constraints it is important to identify where corners can be cut and the benefits and tradeoffs of cutting that corner. The price to performance ratio is the most important factor as you want to find components that are both cheap but perform adequately for the designated tasks and that have a large lifespan. If the components perform well enough to meet your requirement specifications, then it is acceptable. Researching components is vital as we do not have the financial resources to buy and test each individual part for compatibility or efficiency. Datasheets are useful for finding operating requirements as well as part limitations and overheads. These datasheets provide the backbone of our research as it allows for direct comparison between parts from different manufacturers to be made.

7.1.3 Environmental & Situational Constraints

When designing a product one must consider the environment that the product will be used in as well as the effects that the product will have on the natural environment. Environmental constraints are important in most if not all first world countries and if a product does meet these requirements then the creator could be fined or even jailed. Our autonomous cart is battery operated therefore it is important to choose a non-toxic battery. Some battery types such as Nickel Metal Hydride batteries are banned in certain countries due to the toxic chemicals it releases upon disposal. Since the autonomous cart is being designed to be operated in an airport the cart must be secure.

This means that there should be no large opening or gaps where unauthorized items could be stored. The cart must also be all the federal standards for an airport. These standards will be discussed later in the report as meeting them are vital to the profitability of this product. Solar was considered for this project however, it would be an unnecessary cost as this product operates indoors. Even if the cart did operate outdoors solar panels might still not be viable due to the high cost of some of the higher efficiency panels and the fact that the cart may not have line of sight with the sun a large percentage of the time since airports often have lots of shady areas and concrete structures.

Our product will utilize parts from manufacturers that are compliant with the Restriction of Hazardous Substances (RoHS) which aims to restrict the usage of hazardous materials in electrical equipment and in the production processes of these electronics. Under this restriction toxic materials: lead, mercury, cadmium, polybrominated biphenyls and diphenyl ethers as well as a few more substances are banned. These substances are banned because they can be harmful to those who are exposed or come into contact with objects that contain these materials. These cannot be properly disposed of without causing pollution and in Florida these materials can pollute the groundwater due to the water table being incredibly high. Restriction of Hazardous Substances compliance is determined by utilizing special screening equipment (XRF metal analyzers) that determine if the product contains more than the allowed amount of these banned substances.

7.1.4 Safety Constraints

FollowBot should be extremely safe and follow all safety regulations because as previously stated it will be operating in an environment that places a high value on security and has a large volume of people. That is the reason why we are using ultrasonic sensors instead of optical cameras as our collision detection because they provide a low response time. In an airport people, could walk in front of the cart at any given time therefore both the collision detection as well as the wheels need to be able to stop or change direction rapidly. Optical cameras would be able to detect an object once it enters the frame however that object may not necessarily be close enough to warrant a response from the autonomous cart. Another important safety constraint that should be accounted for is heat generation. Heat generation is important as if a component in the cart becomes too hot then it could be a fire hazard or it could potentially burn someone.

Heat generation may not seem to be a large safety concern however, it is important to remember that this product is intended to carry customer's valuable belongings therefore it should also be safe for their items. This means that the motors and wheels should be smooth and should not have a large amount of bounce or jitter when moving the object. This could possibly be implemented by padding the loading zone with a softer sponge material. A sponge material or a rubber would ensure that the luggage does not slide around when it is being transported and this minimizes the risk of injury as the luggage would be very unlikely to fall off.

The balance of the cart is key as if the cart is not balanced and there is too much weight on one side then the cart could tip. The carts weight should be evenly distributed and the platform should also help to distribute the weight of the luggage. The team will research different designs and materials to determine which provides the best stability and safety. While performance is important, this should not be at the cost of safety. All parts of the electrical system should be covered securely and insulated to prevent unnecessary dangers from electrical shocks and the battery should be housed to prevent moisture or humidity from collecting. The safety of our product will be intensively tested as it is one of the selling points of our design and all of the parts will be purchased from reputable distributers such as Digikey or Microchip.

7.1.5 Social Constraints

The FollowBot should be incredibly easy to use as this product is targeted towards the sickly and elderly. This means that the application should be easy to understand and provide clear and detailed instructions of how to operate and sync the cart. Any images that are displayed on the application should be large and easily readable and the UI of the app should be streamlined so that even a child can learn how to use it. It should be easy to load luggage onto and off the cart therefore when designing the cart a ramp or easy access point should be created to allow for this to occur.

7.1.6 Political Constraints

Our product is being designed at a time where airport security has become increasingly strict due to potential foreign threats and an increased in drug trafficking therefore it is important to consider the potential malevolent uses of FollowBot and ways in which to prevent it. This means that no components should easily accessible thus preventing tampering from occurring. The cart should be designed so that there are no gaps or spaces in which harmful objects such as weapons or drugs can be hidden. The luggage on the cart should be easily viewable so that both airport security and the consumer can ensure that nothing is being removed or placed into their luggage.

7.1.7 Ethical Constraints

As of July 2017, there is no autonomous luggage cart and our idea does not seem to infringe on any current patents. The only similar idea that was discovered was an autonomous suitcase known as TravelMate and while both have the same relative goal of moving luggage they are accomplished in completely different ways. TravelMate has a patent for their depth sensing camera which we will not be utilizing or infringed in any way by our design. If any idea is drawn upon then the express written consent of the inventor or creator of said idea will be obtained through the proper channels to ensure the

7.1.8 Time Constraints

Time constraints are also incredibly important for this projects design as our group only has about 40 weeks to design, test, and implement our autonomous cart. Our project must be completed by the first week of December 2017 which is essentially when the Senior Design 2 class ends. By the third week of Senior Design 2 it is required for our PCB design to have been submitted for production and for the serious prototyping to begin. This is an extremely tight window so our group has set consecutive deadlines and are aiming to finish the project about three weeks before the final deadline. The final draft of the research paper is due August 1st, 2017 and must be a minimum of 120 pages however the team can also edit the paper during Senior Design 2 if any changes are made or if the group wants to add any more research or design sections.

7.2 Standards

In order to utilize Bluetooth technology and to ensure proper environmental precautions are taken our product must comply to certain standards that are created by a plethora of foundations across the country such as IEEE, TOC, etc. These standards change as the location of the market changes in other words, the standards in one country may slightly differ from the standards of another country which means that your product may have to be altered to be compliant with the standards of that region. If the product is non-compliant then the product will not be able to be marketed or sold in that region.

7.2.1 Bluetooth SIG Standards

In order to brand a product or the Bluetooth symbol or trademark one must be a member of the Bluetooth Special Interest Group (SIG). The Bluetooth SIG monitors the usage, licensing, and development of the Bluetooth standards. The RN4020 Bluetooth module that we will be using complies with the Bluetooth core specifications version 4.1 and in order for our product to be marketable it must meet the requirements of the Federal Communications Commission (FCC) which are discussed in later sections. The Bluetooth Qualification Process includes 3 steps which are radio qualification, software qualification, and end product listing. [43]

1. Radio Qualification
   - The first step of the qualification process is to verify that the Bluetooth radio conforms to the specifications. This process is often done by a certified testing facility and tests if the radio is operating at the correct frequencies as well as a plethora of in depth tests that are detailed in the core specifications of each version.
2. Software Qualification
   - Bluetooth devices protocols and stack are tested and a Qualified Design Identification will be issued by SIG. The protocols that are tested for a Bluetooth v4.1 device include GAP, GATT, ATT, SMP, L2CAP, RF PHY
3. End Product Listing
   - The final step in the Bluetooth SIG qualification process is the product being list in end product listings that can be found on the Bluetooth website. The end product listing costs $8000 for adopter members and $4000 for promoter members.

It is important to remember that the Bluetooth SIG standard only needs to be utilized if the product will be utilizing Bluetooth trademarks, images, or logos. In essence, if the product is being advertised as a Bluetooth compliant product then this strict qualification process needs to be undergone or else there will be violation of trademarks and patents. This violation could lead to potential legal action and it was concluded that our product would not need to undergo this process.

7.2.2 FCC Part 15

For a product to be marketed in the United States it must be FCC certified and for this section a plethora of FCC standards will be discussed as they may be relevant to our product or design. The module that is being utilized as already been FCC certified and can maintain certifications for Subpart C once no modifications to antenna are made and as long as the module supply voltage is properly grounded. Any other transmission modules or equipment must also comply with FCC regulations. Subpart B and C will be discussed in this paper due to their relevance to our motors and Bluetooth modules however the other subparts are equally as important.

A typical Bluetooth Low Energy module has to pass a variety of tests which such as:

1. Emission Tests
2. Peak Power & Spectral Density of Peak Power
3. Spurious Emissions
4. Occupied Bandwidth
5. Antenna Requirements
6. Measurements of Band Edges

7.2.2.1 CFR 15.23 Home Built Devices

This subsection from the Code of Federal Regulations states that products that are constructed for personal use and are not marketed do not require equipment authorization because the manufacturer of this device may not have the financial means or equipment to determine whether their device is in compliance with federal regulations. However, good engineering practices such be utilized in order to maximize the extent in which the product does comply to the standards specified. This means that our prototype does not need to be certified but the device must still follow the CFR 15.5 conditions.

7.2.2.2 CFR 15.5 General Conditions of Operation

The CFR 15.5 section is relevant because it aims to prevent the harmful interference caused by objects which the CFR refers to as incidental radiators. According to the CFR incidental radiators include DC motors which will be utilized in this experiment. The code states that the no harmful interference should be caused by the operation of an incidental radiator or that intentional radiators, devices which purposefully generate radio frequencies through radiation, should not product any damped waves.

7.2.3 IEC60950 (Relevant Power Supply Standards)

This standard is intended to ensure that products are insulated correctly in order to prevent serious injuries from occurring due to fire hazards, electrical shock, or high temperatures. A hazardous voltage according to this standard is any AC voltage that is larger than 42.2 volts peak or any DC voltage that is larger than 60 volts. These apply as long as the circuit is not a limited current circuit which is a circuit that is guarantees that a hazardous current will not be reached even if there is a fault. This standard separates equipment into three different classes with each being isolated through different methods: [38,39]

1. Class I: For class I devices, shock protection is performed by grounding to protective earth or using basic insulation material to separate live wire. The protective earth grounding must have a certain colored insulation (green, yellow, or clear) and must not contain a switch or fuse.
2. Class II: Class II devices, unlike Class I devices, do not require a protective earth ground and only requires that shock protection be performed by utilizing reinforced insulation (double insulation)
3. Class III: Class III devices must contain a secondary circuit known as a safety extra-low voltage (SELV) circuit. This circuit makes it impossible for dangerous currents to be produced by the equipment and the SELV circuit should be separated from primary circuits and other dangerous voltages by two other types of insulation or protection.

Hazardous voltages are stopped by utilizing insulation of which there are different types. The most important types of insulation specified in IEC 950 are the basic insulation, supplementary insulation or operational insulation.

1. Basic Insulation: Protects against shocking hazards by utilizing a single level of protection. There is no minimum thickness for this insulation type however, there must be a secondary protection layer such as a protective earth ground.
2. Supplementary Insulation: A layer of protection that is often utilizing as a failsafe for basic insulation. This type of insulation must be at least 0.4 mm
3. Operational Insulation: This type of insulation is the minimum level of insulation needed for the equipment to work correctly and are not meant to provide any protection from fire or shock hazards.

Ultimately, IEC 950 outlines the basis for hazard protection for power supplies and demonstrates the importance of clearance as well as solid insulation in ensuring the safety of the user of the product. These standards are vital in protecting the consumer from short circuits and shocks which could in turn cause fires. The reason this standard was discussed was due to the importance of power supply protections. This can be viewed in the relatively recent “Hoverboard” fires or Samsung Galaxy Note 7 fires that were caused by overheating power supplies and batteries.

7.2.4 Department of Energy Battery Maintenance Standards (DOE-HDBK-1084-95)

The IEEE standard 450 aims to set forth a list of rules in order to ensure that lead-acid batteries are stored and maintained correctly. These rules allow for users to maximize the lifespan of the battery and ensure that the full amount of charging cycles is reached before the battery needs to be replaced. It also ensures the safety of the battery as lead-acid batteries need to be stored in specific conditions. These rules most commonly apply to flooded lead acid batteries as these types of batteries require the addition of distilled water every so often. Sealed lead acid batteries on the other hand need to be periodically cleaned and tested to ensure safe operation. The standard proposes six maintenance rules for lead acid batteries which are:

1. Ensure that a matching charger is used or that the current and voltage rates of the charger are in line with those of the battery.
2. Be careful to not excessively discharge the battery as this shortens the lifespan of lead acid batteries. These batteries have a limited amount of times that they can be fully discharged. One should keep the battery charging as often as possible.
3. The electrolyte levels should be maintained by adding distilled water when necessary. As previously stated this only applies to flooded lead acid batteries.
4. Ensure the contacts of the battery are cleaned as they can get dirty which can cause debris or rust to develop on the contacts.
5. The battery's temperature should be maintained so that it does not overheat. One should ensure that the environment that the battery is operating in is suitable for that type of battery. This means that one should be knowledge of the operating temperature range of the batteries.
6. If the battery is weak or low on charge, then the charge should be equalized.

If the charger that is being used has mismatched specifications when compared to the battery, then this can result in an excessive production of hydrogen and oxygen gas or can cause the battery to overheat. Lead acid batteries are incredibly flexible when it comes to the rate at which they can be charged at and can be charged in a variety of different ways. Lead acid batteries should not be over-discharged as this not only shortens the lifespan but also makes it difficult to recharge the battery due to an increase in the internal resistance of the battery. It is also vital to control the temperature of the battery as high temperatures can cause the metal components to breakdown and corrode quicker as well as causing the self-discharge rate of the battery to greatly increase. If the batteries have an unequal distribution of voltage in the cells, then an equalizing charge can be used to correct this issue. An unequal discharge voltage can lead to one cell being completely discharged while another cell still has charge and this in turn could potentially lessen the lifespan of the battery.

One concern when utilizing flooded lead-acid batteries is maintaining the levels of electrolyte in the battery. This change in electrolyte levels is caused by the loss of water due to the gassing that occurs when the battery is overcharged. The electrolytes are important because help heat transfer from the plates and without properly maintained levels the battery could overheat. These lead-acid batteries also needed to be cleaned as they often get dirty and these dirty spots can cause the terminals of the battery and the steel trays to corrode. According to the standard, a hot water-baking soda solution can be used to wife the battery down. Ultimately, lead-acid batteries while very rugged still require proper maintenance in order to function properly and to extend the lifespan of the batteries. [42]

7.2.4 Battery Storage Standards (IEEE Std 450)

Battery Storage

The IEEE Std 450 addresses a variety of different aspects of lead-acid batteries including how they should be stored as well as their disposal and transportation procedures. According to the standard, the batteries must be stored in “open-circuit condition with the terminals insulated”. The batteries must be stored in a container or area that has a relatively cool temperature and where the air is dry as humidity and heat will damage the battery even if it is only being stored for a short time.

The battery should be stored when it is at a full charge as this allows it to be stored for a longer period of time. As previously mentioned, lead-acid batteries are very negatively impacted when stored when fully discharged. If one is storing a battery for a long time, then the battery should be checked and recharged regularly. When batteries are stored for a long time they can undergo what is known as sulfation which can occur when a battery is being self-discharged. Sulfation occurs when the materials of the electrodes are converted into lead sulfate which then crystallizes and can cause an increase in the battery’s internal resistance which in turn decreases the capacity of the battery. [40,41]

8.0 System Testing & Design

This chapter will consist of the testing and design of each system as well as parts that we’ve acquired so far for each system will be introduced. The first system discussed is the motion system and the fabrication and design of the PCB will be discussed.

8.1 Bill of Materials

This section will provide a list of parts currently purchased as well as a breakdown of the manufacturer, unit cost, and total cost of each item. This provides a framework for any outsider attempting to replicate our project for educational purposes as well as instructors aiming to analyze the efficiency of our proposed solution. It is important to note that one of the Arduino Uno units was donated by a family member of one of the group members therefore the cost is not included in the total costs even though two units are listed. The material that will be used to construct the main structure of the luggage cart was provided free of charge by a group member. All other items were purchased by the group and a picture of all the acquired parts are included in the figure below the table.

Table 14: Bill of Acquired Materials

Level	**Description**	**Manufacturer**	**Part Number**	**Quantity**	**Unit Cost**	**Total Costs**
2	Microcontroller	Arduino	ATmega328	2	$19.95	$19.95
2	Bluetooth Module	HYY	HM-10	2	$6.495	$12.99
2	Ultrasonic Sensor	Elegoo	HC-SR04	5	$1.972	$9.86
1	Jumper Wires	Haitronic	HR0291	120	$0.05825	$6.99
2	Lead Acid Battery	ExpertPower	EXP1270	1	$16.94	$16.94
2	Switching Regulators	Qunqi	MP1584EN	5	$1.798	$8.99
2	Magnetometer	SMAKN	HMC5883L	2	$9.26	$18.52
1	Bluetooth Beacons	Estimote	B01MU7YC87	3	$22.99	$68.97
2	Motor Controllers	Daoki	L298N	5	$2.996	$14.98
2	Motor	Cytron	RB-Cyt-28	4	$14.61	$87.66
1	Wheels	Harbor Freight	Solid 6 in.	2	$7.475	$14.85
1	Battery Charger	UPG	D1761 SLA	2	$7.01	$14.02
2	PCB	PCBWay	Two Layer	10	$27.00	$54.00
				Total		$348.78

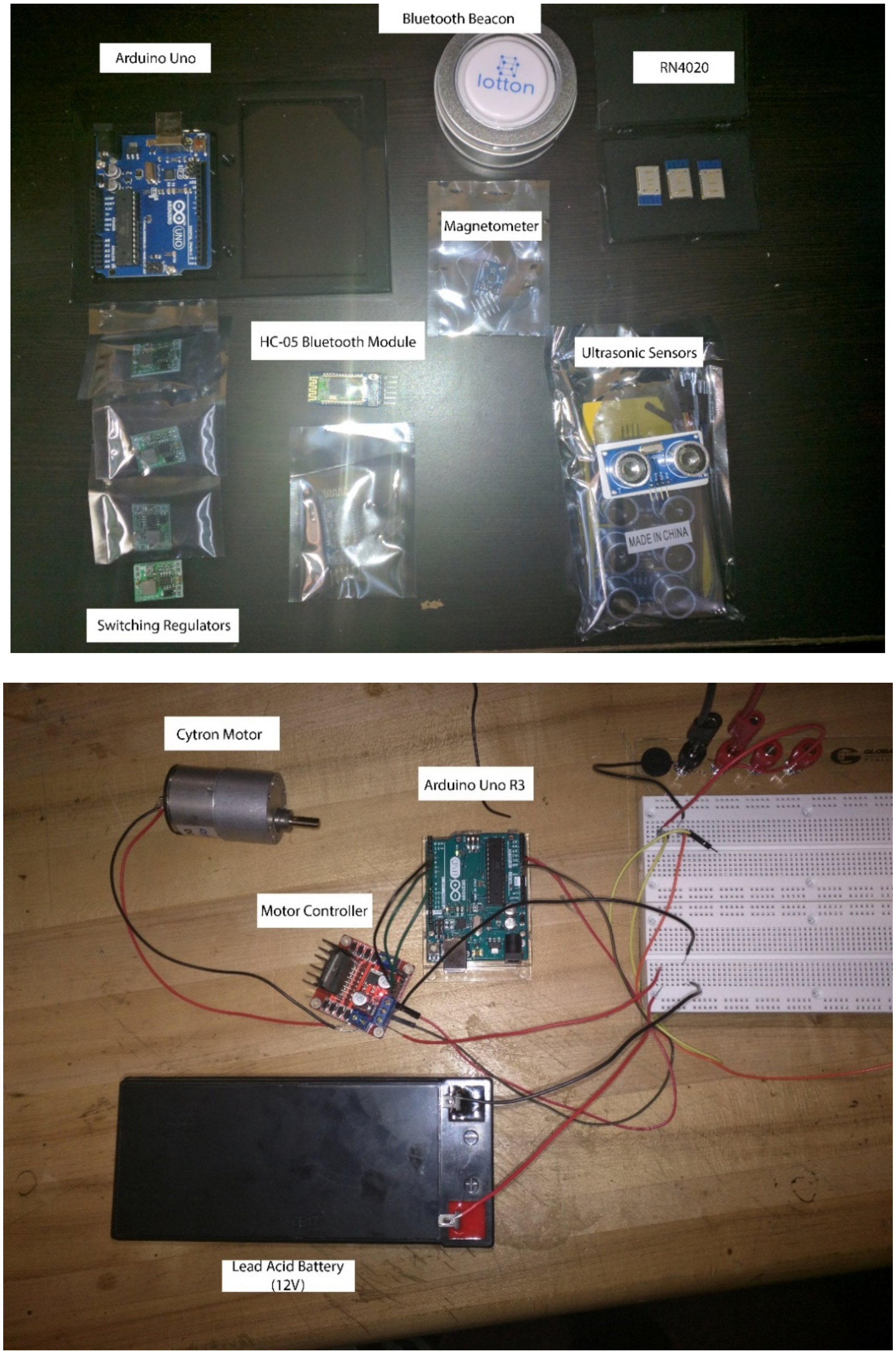


Figure 37: Picture of Acquired Parts

8.2 Motor Orientation Design

In this section, we will discuss the two types of spur gear motors that we decided to test for our design. Before we go in depth on the two types of spur gear motors, we want to ensure that the specifications and curves came from each motor's datasheet. The datasheets are cited in the reference section of this paper. To reiterate what was discussed in the research section of this paper, spur gear motors are not good at handling large amounts of torque and can be noisy when rotational speeds are high. However, they are dependable at low speeds and are relatively cheap. At this moment, only one of each motor was bought for testing.

8.2.1 Cytron 12V 17RPM 194.4oz.-in. Spur Gear Motor

The first motor that was selected to be tested for our design is the Cytron 12V 17RPM 194.4oz.-in. Spur Gear Motor. This motor is designed to control high amounts of torque for the vehicle. The price of this motor is $14.61. This motor is used in applications such as printers, welding machines, ovens, slot machines, coffee makers, and many more. Below are the specifications of this motor that are important to know before starting the design of the motion system.

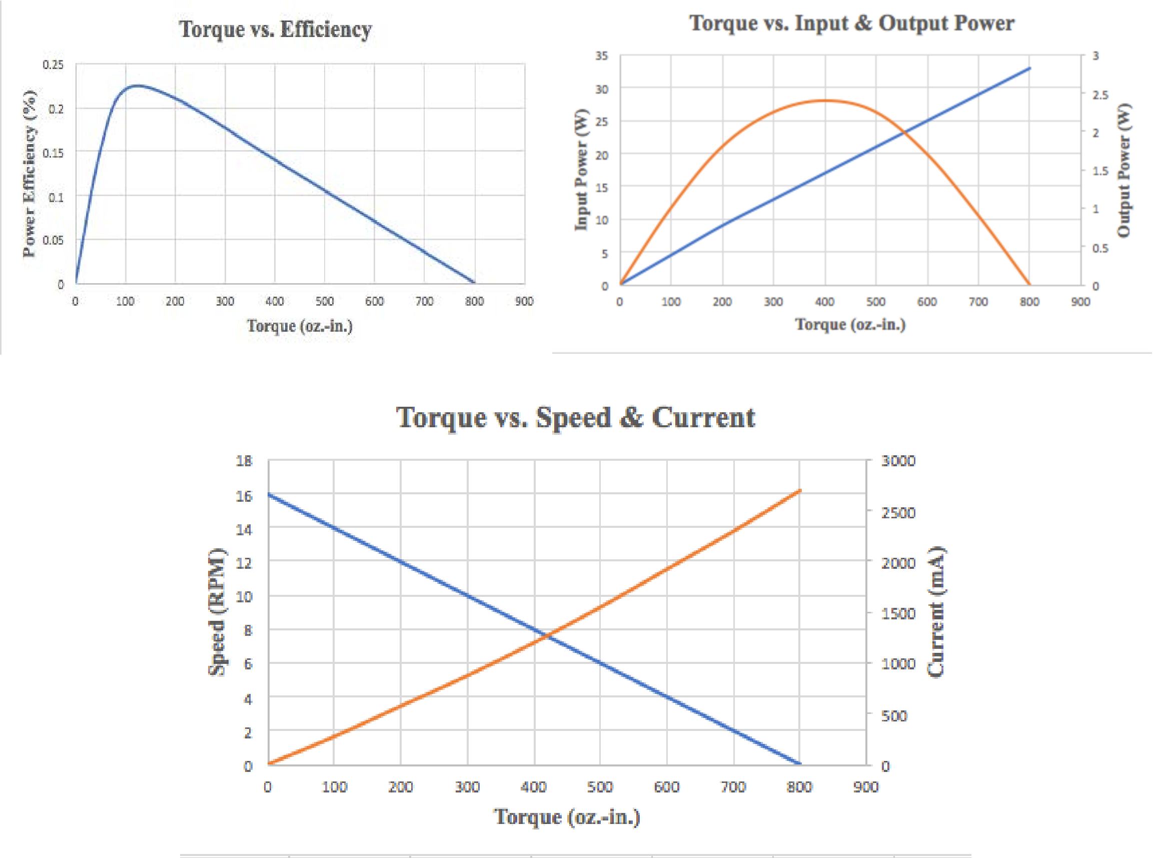
Specifications:

- Nominal Voltage: 12V
- No Load Current: 0.09A
- Power Rating: 1.08W
- No Load RPM: 16.7
- Rated RPM: 12.5
- Rated Torque: 194oz.-in.
- Stall Current: 2.6A
- Stall Torque: 778oz.-in.
- Diameter of shaft: 6mm
- Shaft Type: D-Shaped
- Gear Ratio: 270:1

Predicted Errors and Complications: Now that we have the specifications, we want to discuss the problems we may experience with this motor. Foremost, the values of the no load and rated RPM are low. This may cause the motor's speed to not move fast enough. A way to solve this problem could be to increase the speed variable in the source code used to operate the motor. Another problem could be the high gear ratio. A 270 to 1 gear ratio means that for every 270 turns, the output shaft turns once. This will reduce the speed by a factor of 270 but will also increase the torque by a factor of 270.

Important Factors to Consider: First, the nominal voltage of this motor is 12V. This means that the battery pack used must have a voltage equal to or a little lower than 12V. If the voltage of the battery pack is higher than the voltage of the motor, this may cause damage to the motor. Second, the stall current of this motor is 2.6A. This means that the motor controller used must be able to provide this current. If the motor controller is not capable of providing this current, it is possible that it will be damaged. Lastly, the stall torque of this motor is 778oz.-in. This means that the motor can put out a torque of 778 ounces of force at a distance one inch from the shaft.

Before designing, it is imperative to remember and keep updating the goals that are being pursued. In the project requirements and specifications section of this paper, we stated that we wanted our design to be able to carry a weight of at least 40 pounds and be able to follow the user. For this to be possible, the torque of the motion system needs to be focused on. The graphs shown below are curves of how the torque of the Cytron 12V 17RPM 194.4oz.-in. Spur Gear Motor will be effected with other variables such as speed, current, and power. These curves are approximated so they can be used for guidance.


Torque vs. Efficiency

Torque vs. Input & Output Power

Torque vs. Speed & Current

Figure 38: Torque vs. Efficiency, Power, Speed, and Current for the Cytron 12V 17RPM 194.4oz.-in. Spur Gear Motor

For the torque vs. efficiency curve the power efficiency increases as torque increases. Once around the rated torque is reached which for this motor is 194oz.-in., the power efficiency plummets immediately and reaches zero once the stall torque is reached. For the torque vs. input and output power curve, the input power increases linearly as torque increases indefinitely. The output power increases until a certain torque is reached and then starts to drop. For the torque vs. speed and current curve, the rotations per minutes starts at 17 and decreases linearly as torque increases. Lastly, the current increases as torque increases.

8.2.2 Cytron 12V 150RPM 18oz.-in. Spur Gear Motor

The second motor that was selected to be tested for our design is the Cytron 12V 150RPM 18oz.-in. Spur Gear Motor. This motor is designed to provide speed to the vehicle. There are some qualities of this motor that are the same with the first motor tested. These qualities are price, applications, and some specifications. However, there are differences between their specifications. The specifications for the Cytron 12V 150RPM 18oz.-in. Spur Gear Motor are shown below.

Specifications:

- Nominal Voltage: 12V
- No Load Current: 0.09A
- Power Rating: 1.08W
- No Load RPM: 150
- Rated RPM: 103
- Rated Torque: 10oz.-in.
- Stall Current: 1.8A
- Stall Torque: 97oz.-in.
- Diameter of shaft: 6mm
- Shaft Type: D-Shaped
- Gear Ratio: 30:1

Predicted Errors and Complications: Now that we have the specifications, we want to discuss the problems we may experience with this motor. Foremost, the values of the rated torque and stall torque are low. This may cause the motor’s to not be able to carry a certain amount of load. Since this motor is mainly meant for speed, another factor that could be a problem is the gear ratio. Even though it is significantly lower than the first motor’s gear ratio, the speed will still be decreased by a factor of 30. However, on the bright side, since this motor needs a little help with torque, the gear ratio will increase the torque by a factor of 30.

Important Factors to Consider: First, the nominal voltage of this motor is 12V. Second, the stall current of this motor is 1.8A. We already know that the battery pack and the motor controller must meet the requirements for these two specifications or else there could be damage. Lastly, the no load RPM and rated RPM are 150 and 103, respectively. The no load RPM is the maximum speed that the motor can handle when no load is attached to it. The rated RPM is the speed at which the motor will first produce its maximum designed power output. A reduction ratio should be applied on the RPM if a load is attached to the motor. If the load applies the same torque as the motor’s rated torque, then the reduction should be applied to the rated RPM.

The amount of stall torque for this motor is significantly less compared to the previous motor. Just like we said before, torque is one of our main priorities that need to be focused on if we want our vehicle to carry a certain amount of weight and move at a decent speed. The graphs shown below are curves of how the torque of the Cytron 12V 150RPM 18oz.-in. Spur Gear Motor will be effected with other variables such as speed, current, and power. These curves are approximated so they can be used for guidance.

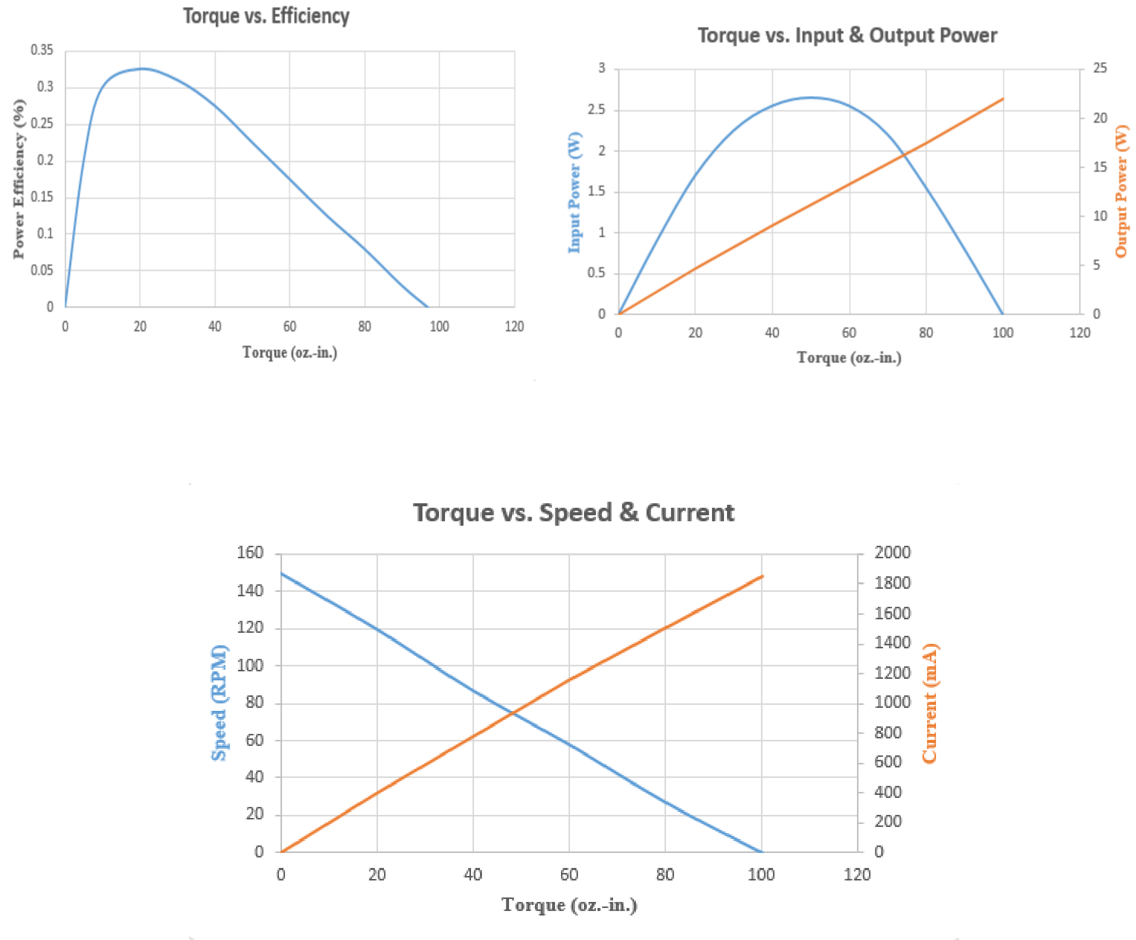


Figure 39: Torque vs. Efficiency, Power, Speed, and Current for the Cytron 12V 150RPM 18oz.-in. Spur Gear Motor

The torque curves for this motor are quite like the Cytron 12V 17RPM 194.4oz.-in. Spur Gear Motor. The main difference between them is the amount of torque they can handle. Since both torque curves are similar, we will not go in depth in describing each graph.

Now that we have tested and collected data for both DC geared motors, it is time to discuss the type of motor controller that was selected that the motors will connect to for testing.

8.3 Motor Controller Testing

In this section, we will discuss the motor controller’s components and design options that we are considering for the motion system.

8.3.1 L298N Dual H-Bridge DC Stepper Motor Drive Controller

The motor controller that was selected to be tested for our design is the L298N Dual H-Bridge DC Stepper Motor Drive Controller. When dealing with a motor controller, many issues may rise because of the many connections and factors that are contributed. That is why it is smart to buy several pieces of the same motor controller just in case an issue occurs. This motor controller was available with a set of five pieces for a cost of $15.78. The specifications for this motor controller was already mentioned in chapter 3.7. However, even though it was mentioned, it is a good to compare the specifications of both motors and the motor controller to see if they both properly match with each other. Below is a table that shows a comparison of the motors and motor controller that were selected.

Table 15: Comparison between the Cytron Spur Gear Motors and the L298N motor controller

L298N Dual H-Bridge DC Stepper Motor Drive Controller	**Cytron 12V 17RPM 194.4 oz.-in. Spur Gear Motor**	**Cytron 12V 150RPM 18oz.-in. Spur Gear Motor**	**Check Box**
Can handle a drive voltage between 5V and 35V	Nominal voltage is 12V	Nominal voltage is 12V	✓
Can handle a max stall current of 3A	Stall current is 2.6A	Stall current is 1.8A	✓
Can handle a max power of 25W	Power rating is 1.08W	Power rating is 1.08W	✓
Can drive one 2-phase stepper motor, one 4-phase stepper motor or two DC motors	DC gear motor	DC gear motor	✓

All the requirements between both motors and the motor controller properly match. Now that we have concluded that it is possible to use all the devices with each other, it is time to discuss what the L298N Dual H-Bridge DC Stepper Motor Drive Controller offers. Below is an above view image and labels of each component that the motor controller offers.

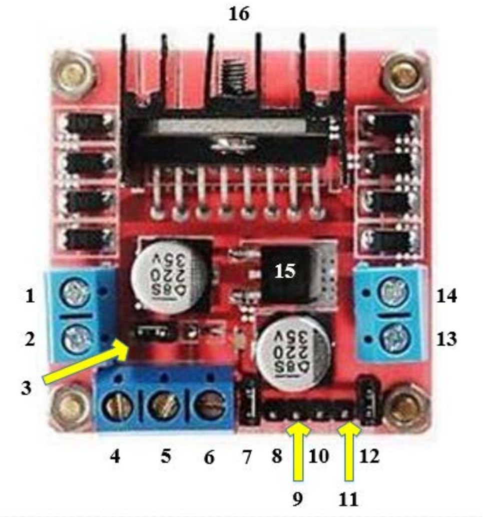


Figure 32: Labeled Components for the L298N Dual H-Bridge DC Stepper Motor Drive Controller

L298N Dual H-Bridge DC Stepper Motor Drive Controller Components:

1. DC Motor A positive lead out
2. DC Motor A negative lead out
3. 12V jumper: If a supply voltage is used that is greater than 12V DC then make sure to remove this. This will enable the 5V voltage regulator.
4. Connect the supply voltage here. The range of voltage can be from 5V to 35V.
5. GND
6. 5V input/output from voltage regulator which is used to connect the micro-controller (Arduino Uno).
7. Enables DC Motor A PWM signal to control speed and direction. Leave in place if using a stepper motor. We will label this pin as ENA
8. IN1
9. IN2
10. IN3
11. IN4
12. Enables DC Motor B PWM signal to control speed and direction. Leave in place if using a stepper motor. We will label this pin as ENB
13. DC Motor B positive lead out
14. DC Motor B negative lead out
15. Internal 5V voltage regulator
16. L298N Dual H-Bridge

In the components list, numbers 5, 6, 7, 8, 9, 10, 11, and 12 will be connected to the micro-controller which in this case will be the Arduino Uno R3. Shown below are images of the digital PWM pins and the power pins that are available on the Arduino Uno R3.

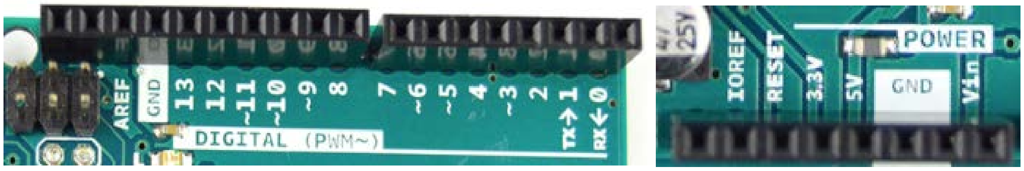


Figure 33: Digital PWN Pins (Left) and Power Pins (Right) on the Arduino Uno R3

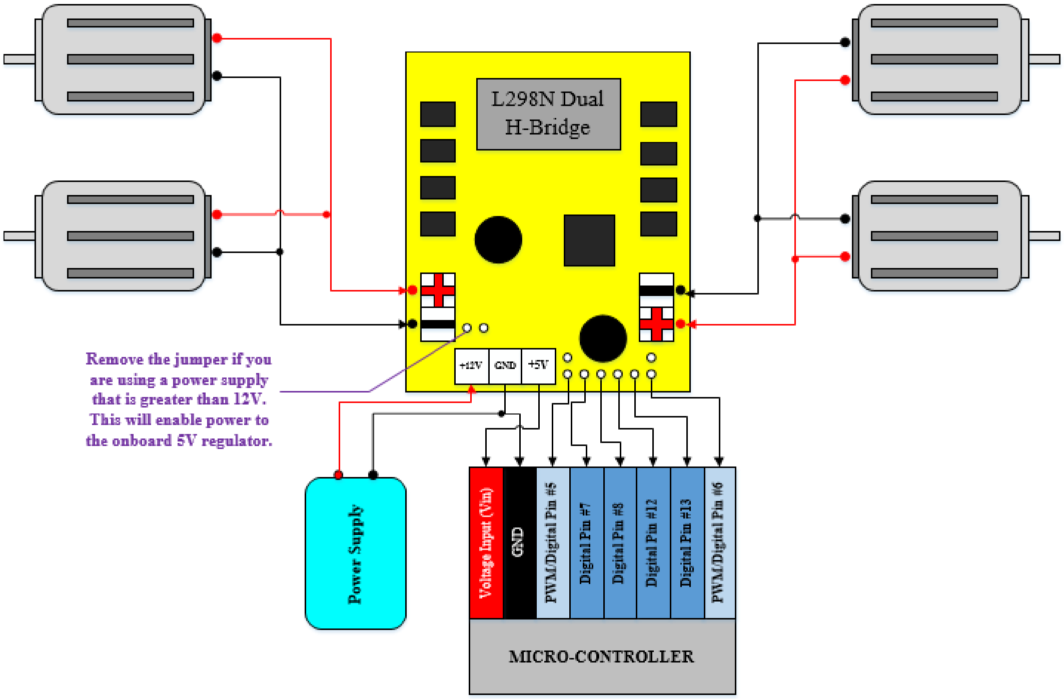
In the image with the digital pins, the pins that have the symbol ~ are specifically used for PWM signals. It is imperative to connect the EN pins on the motor controller(s) to those pins on the Arduino. The IN pins on the motor controller(s) should be connected to the digital pins that don't have the ~ symbol. However, if needed, it is possible to connect the IN pins on the motor controller(s) to the digital pins with the ~ symbol.

In the image with the power pins, the pins labeled GND will be connected to the GND terminal(s) on the motor controller(s). The last pin is V_{in} which will be connected to the 5V input/output terminal on the motor controller.

The pins discussed will be connected to the L298N Dual H-Bridge DC Stepper Motor Drive Controller(s) depending on the type of design that we decide. There are a couple of options that we are considering for our design which are explained below.

8.3.1.1 Motor Controller Design Option 1

Our first design consists of four DC motors attached to one L298N Dual H-Bridge DC Stepper Motor Drive Controller. The schematic of how our design will connect is shown below.



Remove the jumper if you are using a power supply that is greater than 12V. This will enable power to the onboard 5V regulator.

Figure 34: Schematic of Four DC Motors Connected to One Motor Controller

The positive and negative wires of the two DC motors on the left are screwed to the positive and negative terminals on the left of the motor controller. We will call these terminals OUT1 (+) and OUT2 (-). These two motors will be controlling the front wheels of our design. The connection will be the same for the two DC motors on the right side. We will call these terminals OUT3 (+) and OUT4 (-). These two motors will be controlling the back wheels of our design.

There will be one power supply for this design. The positive wire of the power supply is connected to the +12V input terminal and the negative wire is connected to the GND terminal on the motor controller.

The last hardware details are the connections between the pins on the motor controller and the pins on the Arduino Uno R3 micro-controller. Since there are plenty of connections between these two devices, a table is shown below for a better visual representation. Please refer to the components list for the L298N Dual H-Bridge DC Stepper Motor Drive Controller.

Table 16: Hardware Details between Arduino Uno and L298 Dual H-Bridge DC Stepper Motor Drive Controller

L298N Dual H-Bridge DC Stepper Motor Drive Controller Pins	**Arduino Uno R3 Pins**
GND	GND
+5V input/output	Vin
ENA	PWM/Digital Pin #5
IN1	Digital Pin #7
IN2	Digital Pin #8
IN3	Digital Pin #12
IN4	Digital Pin #13
ENB	PWM/Digital Pin #6

Below are some advantages and disadvantages with connecting four DC motors to one L298N Dual H-Bridge DC Stepper Motor Drive Controller.

Advantages:

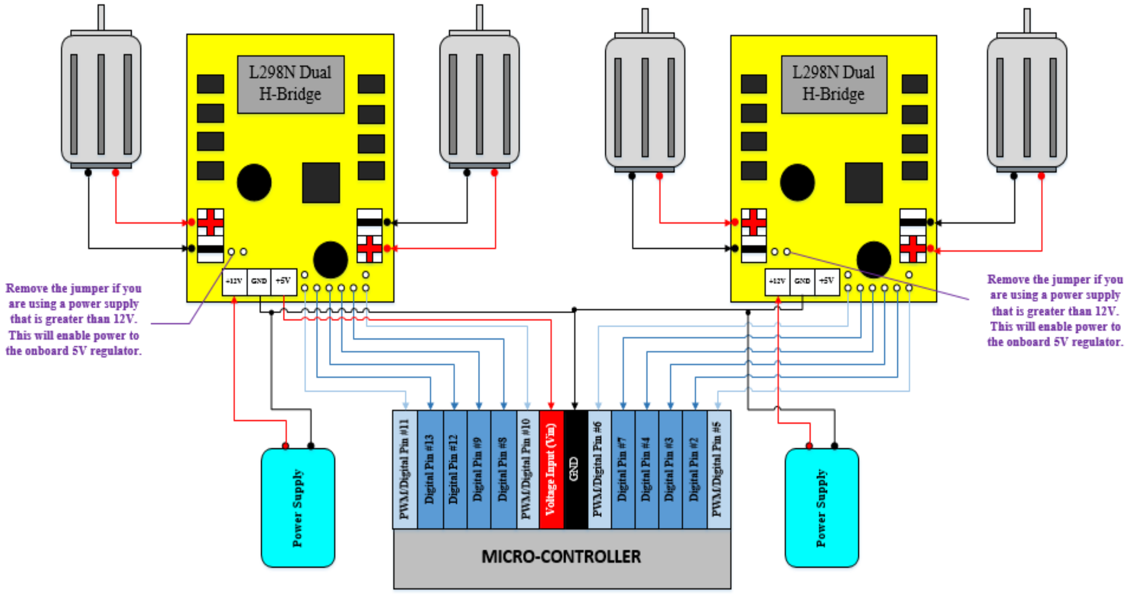
- The two motors that are connected to the front wheels will have the same control and speed. This will be the same for the two motors that are connected to the back wheels as well. To reiterate, we will have control over the front and back wheels. This is useful if the front wheels are for some reason not in contact to the ground, so instead we can use the back wheels.
- Takes up less space on physical design.

Disadvantages:

- The motors will run slowly.
- The motors will not be able to be controlled independently. The motors may only be controlled independently if two motor controllers are used.

8.3.1.2 Motor Controller Design Option 2

Our second design consists of two L298N Dual H-Bridge DC Stepper Motor Drive Controllers containing two DC motors each. The schematic of how our design will connect is shown below.


Remove the jumper if you are using a power supply that is greater than 12V. This will enable power to the onboard 5V regulator.

Remove the jumper if you are using a power supply that is greater than 12V. This will enable power to the onboard 5V regulator.

Figure 35: Schematic of Two Motor Controllers Containing two DC Motors each connected to Microcontroller

For this design, each motor's positive and negative wires are connected to its own positive and negative terminals on a motor controller. We will label the left motor controller CON1 and the motors connected to it Motor A (left) and Motor B (right). We will label the right motor controller CON2 and the motors connected to it Motor C (left) and Motor D (right). A table is shown below to give a better visual representation of the positive and negative terminals that each motor is connected to on the motor controllers.

Table 17: Hardware Details between the DC Motors and Motor Controllers

	CON1	**CON2**
Motor A	OUT1 (+) & OUT2 (-)	------------
Motor B	OUT 3 (+) & OUT4 (-)	------------
Motor C	------------	OUT5 (+) & OUT6 (-)

Motor D	------------	OUT7 (+) & OUT8 (-)

There will be two power supplies for this design. The positive wire for each power supply are connected to a +12V input terminal on a motor controller. The negative wire for each power supply are connected to a common GND between both motor controllers and the micro-controller. The last hardware details are the connections between the pins on the motor controllers and the pins on the Arduino Uno R3 micro-controller. Since there are plenty of connections between these three devices, a table is shown below for a better visual representation. Please refer to the components list for the L298N Dual H-Bridge DC Stepper Motor Drive Controller.

Table 18: Hardware Details between Arduino Uno and Motor Controller

CON1 Pins	**CON2 Pins**	**Arduino Uno R3 Pins**
GND	------------	GND
+5V input/output	------------	Vin
ENA	------------	PWM/Digital Pin #11
IN1	------------	Digital Pin #13
IN2	------------	Digital Pin #12
IN3	------------	Digital Pin #9
IN4	------------	Digital Pin #8
ENB	------------	PWM/Digital Pin #10
------------	GND	GND
------------	+5V input/output	N/A
------------	ENC	PWM/Digital Pin #6
------------	IN5	Digital Pin #7
------------	IN6	Digital Pin #4
------------	IN7	Digital Pin #3
------------	IN8	Digital Pin #2
------------	END	PWM/Digital Pin #5

Notice that the +5V input/output terminal on CON2 is not applicable. The reason for this is because the +5V input/output terminal on CON1 is already supplying power to the Vin pin on the Arduino Uno. Below are some advantages and disadvantages with connecting four DC motors to two L298N Dual H-Bridge DC Stepper Motor Drive Controllers.

Advantages:

- Each motor can be controlled independently.
- Motors will run quickly.

Disadvantages:

- Front and back wheels will not able to be control simultaneously.
- Takes up more space on the physical design.

8.3.2 Motor Breadboard Testing

In order to verify the functionality of the motor prior to the construction of the prototype unit, the motor was connected to the motor controller which was then powered by the 12-volt lead acid battery. The motor controller was connected to the Arduino unit and code was utilized in order to make the motor turn. The current as well as the rotations per minute (RPM) were measured to ensure that the actual motor performance was very close the specifications provided in the datasheet. A schematic of this connection can be found in the motor controller testing section of this laboratory report and an image of the breadboard test is provided below. It was determined that the motor controller did in fact provide enough torque and RPMs to meet the projects requirements specifications.

Table 19: Prototype Motor Controller Testing Procedure

Process	Procedure
Step 1	Begin by connecting your test device (phone) to the luggage cart through Bluetooth
Step 2	Once connected attach Bluetooth module to person and ensure that triangulation modules are set
Step 3	Ask test operator to move in a straight line to begin motor test
Step 4	Operator should then begin to move in different directions
Step 5	Test the voltage and wave signals of the motor and motor controller to ensure proper syncing

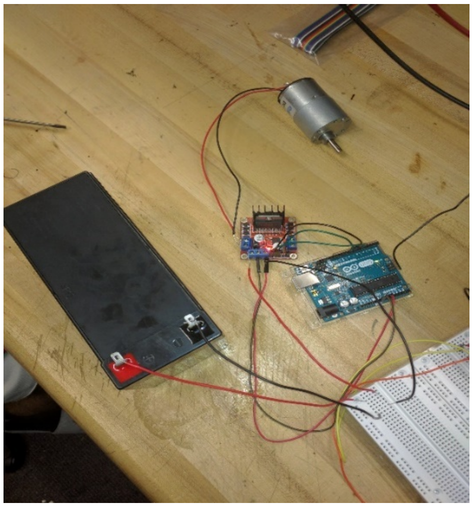


Figure 40: Breadboard Configuration for Motor Testing

Once the Bluetooth module on the Arduino and the Bluetooth beacons are synced then the testing of the motor controller and motors can begin. The operator is asked to move in a straight line in order to test the motors operation because this displays that the Beacons and motors are in sync and that the positional tracking is being correctly implemented. The operator should then begin to strafe in order to display the motors ability to turn left or right. This functionality is vital because it will be coordinated with the collision detection system in order to avoid obstacles. This is possibly the most important step of the procedure as if the motors are not turning properly or quickly enough then this could be disastrous as the cart could strike the obstacle. The testing of the voltage and waveform of the motor controller is solely to ensure that the voltage being provided to the motors is enough and that the heat that the motors generate is being limited through proper operating conditions.

8.4 Dagu Wild Thumper Wheels

The Dagu Wild Thumper Wheels will be used for our design. They are large wheels with a size of 120 x 60mm (4.72 x 2.36in). They are used for robots that need to traverse complex, rugged terrain [47]. The black tires on these wheels are soft, spike rubbers for increase traction, and the plastic hubs can come in colors of metallic red or chrome [47]. These wheels are relatively cheap and come at a price of $14.95 a pair. To reiterate, we want the FollowBot to be capable of mounting a suitcase, cooler, or any reasonable object. This means that high amounts of weight needs to be moved at all times. The Dagu Wild Thumper Wheels were chosen because they will be strong enough to move high amounts of weight, and maneuver the weight through all sorts of terrain. Below is an image of a Dagu Wild Thumper Wheel.

Figure 41: Dagu Wild Thumper Wheel (120 mm)

Now that we have discussed the type of wheel that we want to use for our design, it is time to discuss how the wheels will be connected to the motors.

It is important to recognize the shaft diameters for both the wheel and the motor. The shaft diameter for the Dagu Wild Thumper Wheel is 4mm, and for both Cytron 12V Spur Gear Motors the shaft diameter is 6mm. Since the difference between them is 2mm, a shaft adapter will be needed to make the connection between the motor and the wheel possible. We decided to go with the 12mm Hex Wheel Adapter for 6mm Shaft. They are 20mm long aluminum adapters that convert a 6mm motor output shaft to a 12mm hex shaft [48]. They are compatible to fit with most common hobby RC wheels, such as the Dagu Wild Thumper Wheels [47]. They come in a pack of two at a price of $3.95. Below is an image of the 12mm Hex Wheel Adapter for 6mm Shaft.

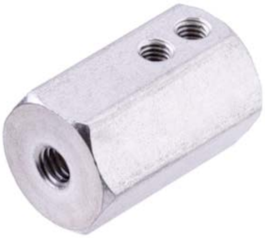


Figure 42: 12mm Hex Wheel Adapter for 6mm Shaft

Now that we have discussed how the wheels will be connected to the motors, it is time to discuss how the motors will be connected to the physical design.

We have decided to go with the Pololu Stamped Aluminum L-Bracket Pair for 37D mm Metal Gearmotors. They are light weighted brackets that are specifically designed for motors with a diameter of 37mm [49]. This type of bracket is perfect for the Cytron 12V Spur Gear Motors. They come in a pack of two at a price of $7.95. Below is an image of the Pololu Stamped Aluminum L-Bracket Pair of 37D mm Metal Gearmotors.

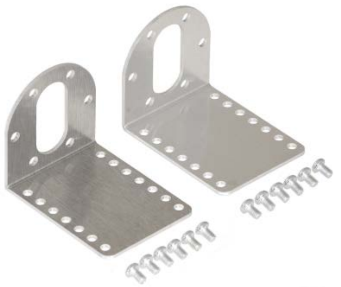


Figure 43: Pololu Stamped Aluminum L-Bracket Pair of 37D mm

8.4.1 Turning Mechanism

In the previous section, we discussed the design options that are considered between the motors and motor controller(s). We know that with design option 1, the motors that are connected to the front wheels will have the same direction and speed, and this concept is the same for the motors connected to the back wheels. We know that with design option 2, all four motors will be independent to each other when it comes to direction and speed. With each design option, comes a different technique for how the vehicle will turn and be connected to the chassis of our design.

The first technique that we want to discuss is independent driving. In this technique, each wheel is driven and controlled independently, so the only way this will be possible is with design option 2. This approach may cause some issues. Each wheel will be driven in its own direction, so coordination between the wheels may be challenging. If coordination breaks down, then the motors will fight with each other which may affect the whole motion system. However, if coordination is accomplished, then movement on uneven terrain will be possible. For example, let’s say one wheel stops working, the other wheels will be able to position the robot and put it in the right direction.

The second technique that we want to discuss is differential driving. This type of technique is usually most common for people that are just getting into robotic design. The way this technique works is a difference between the velocity of two motors. This will drive the robot in any direction and path that is required.

With both techniques, the chassis is a very important aspect when it comes to the robot being able to turn, especially with weight added on. The distance between the wheels can determine how efficient the robot will be able to turn. Below are three examples of basic designs for how a chassis can be designed.

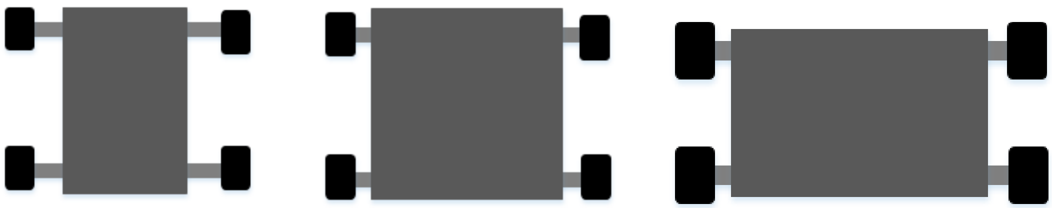


Figure 44: Three Possible Chassis Design Shapes & Sizes

Before we discuss about each design, we will label each wheel with a letter. The top left wheel of each design is A, the top right is B, the bottom left is C, and the bottom right is D.

Design 1: In this design, the distance between wheels AC and BD is much greater than the distance between wheels AB and CD. Let's say that wheels CD start to rotate in opposite directions (the wheel C rotates backwards and D forwards), the wheels AB will start dragging to the left. During this process, there is such a high traction occurring on the front wheels. This exerts a conflicting force which makes it even harder for the robot to turn left. On top of this, adding weight to the chassis will make turning very difficult.

Design 2: In this design, the distance between wheels AC, BD, AB, and CD are equal. If we apply the same concept that was stated in design 1, the front wheels will have a better performance turning than design 1, but there will still be a wheel slippage. Wheel slippage happens because the front wheels will come in the way of circular pull created by the back wheels. The robot will barely be able to rotate at its position because of the wheel slippage that happens to the front wheels. Just like in design 1, adding more weight to the chassis will again make turning very difficult.

Design 3: In this design, the distance between wheels AB and CD is much greater than the distance between wheels AC and BD. Again, using the concept that was stated in design 1, the front wheels will give almost to no resistance to the circular motion created from the back wheels. The torque that is being applied by the motors will be properly harnessed which will result in a better performance turning. Some advantages for this design are quick turning, better traction, no loss of torque, and no wheel slippages.

8.5 Ultrasonic Sensor Testing

In order to test the functionality of the ultrasonic sensors it is first vital to understand the function of each of the four pins present on the sensor module. The four pins present are: VCC, GND, Echo, and Trigger. The trigger pin is essential as it tells the sensor when to send the 40-kHz pulse of ultrasonic waves. The trigger pin also allows for the pulses to be evenly spaced (10 microseconds). The echo pin allows for the retrieval of the returning digital pulse that is then used to calculate the distance of the object by the amount of time it has taken for the ultrasonic waves to return. The ultrasonic sensor is a vital part of our design as it functions as the backbone of our collision detection system. If one of the five ultrasonic sensors are malfunctioning, then the cart can injure a person as it will not be able to avoid them. The breadboard test of the ultrasonic sensor was conducted by activating the sensor through Arduino code and then moving the sensor closer to a stationary object (such as one's hand). The distance was then obtained and from the test it seems that the Arduino can measure objects as close as 4 cm and as far as approximately 180 cm before providing incorrect values.

Table 20: Testing Process for Ultrasonic Sensors

Process	**Procedure**
Step 1	Connect the cart to the tester's phone or Bluetooth device and ensure that the connection is stable
Step 2	Place a cone at least 10 feet away from the cart and beginning moving towards the cone
Step 3	The tester should pass in front of the cone and if the ultrasonic sensors are working then the cart will immediately stop
Step 4	Repeat this step for each direction or in essence place an object within the range of the sensors for each of the four sensors
Step 5	The cart only passes if and only if all of the sensors pass the test

8.6 Switching Voltage Regulator Testing

The switching voltage regulator is perhaps the easier part to test as all that needs to be done is to connect the input end of the voltage regulator to a triple channel power supply and to connect the output end of regulator to a multimeter. The potentiometer should then be adjusted so that it matches the required output voltage. For example, if one wanted to connect the 12-volt sealed lead acid battery directly to the Arduino Uno then a switching voltage regulator with an output of 5 volts should be utilized. If not, then this could result in severe damage to the components of the board. The battery can also be tested quickly using the switching voltage regulator because if battery is connected and the regulators output does not return a voltage then it can be assumed that the battery is dead.

Table 21: Testing Process for Switching Voltage Regulators

Process	**Procedure**
Step 1	Connect the input end of the switching regulator to the ground and VCC of the triple channel power supply or the battery
Step 2	Attach a multimeter to the output end of the switching regulator
Step 3	Adjust the potentiometer until the desired output voltage is reached

8.7 Bluetooth Testing

The testing of the Bluetooth consists of testing both the Bluetooth modules that are being utilized on the board as well as the Bluetooth beacons that are being utilized for triangulation. The beacons serve as the guide for the cart as it transmits the position of the user to the cart allowing the cart to follow the user therefore it is important that this connection not only functions properly but also has incredibly low latency. If the modules are running slow then the movement of the cart will be delayed therefore a sample file should be transferred via the Bluetooth connection of the receiver in order to test the efficiency of this device. The Bluetooth syncing procedure will be done directly from then application and if possible our group might try to implement a signal strength monitor so that the user knows if the Bluetooth strength is weak or strong.

Table 22: Bluetooth Testing Procedure

Process	**Procedure**
Step 1	Power on the luggage cart and turn on Bluetooth on the cellphone
Step 2	Connect the Bluetooth module to the phone and verify connection is secure
Step 3	Send small image or file to test that connection is working
Step 4	Check if Bluetooth module is connected to the four beacons (this will be indicated by four LEDs) which are meant to indicate the status of each beacon
Step 5	If the beacons are all connected, then the module is functioning properly and can be utilized.

8.8 Microcontroller Testing

The microcontroller will be indirectly tested by through the testing of the functionality of the other components. This is done because most if not all of the pins of the microcontroller are being utilized by either the motor controller, ultrasonic sensors, magnetometer, or another component. This means that if these components function properly with the proper connections and efficiency then the microcontroller is also functioning properly.

8.9 PCB Vendors

In this section, we will discuss two companies that manufacture printed circuit boards. These two companies are Advanced Circuits, also known as 4PCB, and Custom Circuit Boards. Before we go in depth about these companies, we want to ensure that most information and data provided came from the company's very own website. Their websites are cited in the reference section.

There are several aspects that we are looking for before choosing a company to produce our PCB such as a low price, great reviews, great quality, and being trustworthy. Below is information and data on both companies that we are considering to manufacture our PCB.

Advanced Circuits or 4PCB: Advanced Circuits has been around since 1989 and are ranked among the top three circuit board fabricators in the United States. They have fabricated PCBs for customers in the military, aerospace, defense, and medical fields. Below are some things that Advanced Circuits offers.

- Reliability. Advanced Circuits get more than 200 new company costumers each month.
- Quality. All orders that are placed receive a free engineering file review to make sure the customer gets exactly what they want.
- On Time. They have a record of being the best on-time shipping record in the industry.
- 24 Hour Tech Support.
- No Tooling charges on new part numbers.
- Instant PCB quotes.
- Free PCB design software (PCB Artist).

There is one special that Advanced Circuits offers that really caught our attention and that is the New Customer PCB Special. This special allows new PCB customers to receive 50% off their first and second order up to $250. In addition to all of that, Advanced Circuits also gives student sponsorships & discounts on PCBs. Below are some specifications of what they offer for PCB manufacturing.

Table 23: Advanced Circuit's Standard & Custom Specs for PCB

Specifications	Standard Specs	Custom Specs
Layer Count	0 – 10 layers	0 – 40 layers
Turn Time	Same Day – 5 Days	Same Day – 4 Weeks
Quantity Requirement	1 - 10,000+	1 – 10,000+
Materials	FR-4	FR-4, Rogers, Polyimide, Aluminum Clad, High-Temp. FR4, Others
Plate Finish	Lead-Free HAL*	Electrolytic Hard Gold, Soft Gold, ENIG, Nickel, Immersion Silver, Leaded & Lead-Free HAL
Cert. / Qualifications	IPC Class 2 – A600	IPC6012 Class 2-3A, IPC6018 Class 3, MIL-PRF-31032, MIL-PRF-55110, ISO 9001:2008, AS9100C, Others
Board Thickness	.031", .062", .093", .125"	Full Range Available
Copper Weight	1 oz. Inner, Up to 2 oz. Outer	0.5 – 4 oz. Inner, 1 – 20 oz. Outer
Trace / Space	5 / 5 Mils	Down to 2.75 / 3 Mils

Custom Circuit Boards: Custom Circuit Boards is celebrating 35 years of prototype PCB fabrication and manufacturing expertise. Since 1982, they have provided good customer service because of their ability to listen, communicate, and react. Similar to their competition, they have done business with customers in the military, aerospace, medical, automotive, industrial, and commercial fields. Students and hobbyists have also relied on them. Below are some things that Custom Circuit Boards offers.

- Trusted on time delivery.
- Qualified engineering in the USA for support.
- Extremely competitive pricing.
- No need to source other offshore manufacturers.
- Fast turnarounds.

- Low minimums.

Custom Circuit Boards does offer a PCB special as well for their new customers. Customers that are trying out their first order will receive a discounted price. There was one aspect about this company that we have noticed they do not offer that their competition does, and that is their own PCB design software. This is not much of a problem because there are several free PCB design software programs; however, it could make the difference of choosing their competition. Below are some specifications of what they offer for PCB manufacturing.

Table 24: Custom PCB Manufacturing Specs

Specifications	**Technology**
Number of Layers	1 – 40 Layers
Board Materials	FR4, FR406, 370HR, FR408, Rogers Ultralam 200, Rogers RO4350, Rogers RO4003, Polyimide, Teflon, Black FR4. Arlon AR350, Getek Copper Clad Thermal Substrates, Hybrid (Rogers and FR4) BT Epoxy, Nelco 4013, Metal Core Materials
Stiffeners	Thermo Set and PSA Based Aluminum, FR4, Stainless Steel, Polyimide
Maximum PCB Size	2 Layer 20” x 28” Multilayer 16” x 26”
Tolerances	PTH Hole Size – +/- .002” Front to Back – +/- .002” NPTH Hole Size – +/- .001” Solder Mask – +/- .002” Tooling Holes – +/- .001” Hole to Pad – +/- .005”
Other PCB Services	Blind and Buried Vias, Plated Slots Specified Dielectric, Tented Vias Controlled Impedance, Solder mask Plugged Vias Via Caps (Solder Mask), Conductive Filled Vias

Both companies offer several more specifications for their PCB manufacturing. Please refer to their websites in the reference section of this paper.

8.9 Physical Product Design

The physical characteristics of the FollowBot must take a few very important things into consideration. One key element of designing the actual product is size and carrying capacity. Since the whole point of the FollowBot is the ability to carry weight effectively while maintaining in motion, the design must complement this. To allow for maximum convenience while travelling with suitcases, the FollowBot should ideally be able to hold a suitcase in its upright position to conserve space, and remain low profile in nearly every other aspect. What we want is essentially a platform with maneuverable wheels and almost a shopping cart style container to hold a suitcase or other personal items.

One of the other major characteristics to consider is safety, both user and other people in the bot's vicinity. Although the object collision detection system should take care of most of this issue, the design should still be safe as to not harm anyone that may run into it. Other than efficiency and safety, the design should be generally aesthetically pleasing due to being marketed to the public. While it does not have to be an innovative marvel, an aesthetically pleasing design will promote more use and will be easier to market. The design will be centered around functionality over aesthetics, however, and will focus on making travel convenient for the user. For the materials used to build the design, we decided to go with a wood base for the main platform due to the efficient weight to strength ratio. Wood is cheap and reliable and can withstand wear and tear while also being replaceable.

A wood platform will allow us to house the electronics underneath and screw all the necessary parts on without issue. The dimensions in mind are 36" by 18" with the platform itself being approximately 6" in thickness. Under the wood base will reside the wheels and electrical components needed to power and run the product. The wheels will be used with standard rubber coating to provide traction for easy navigation in public places. Above the platform will be an approximately 24" tall metal grate surrounding the cargo on all sides. This will allow the user to place any form of luggage or personal items into the carrier without items falling off easily. In accordance to being safe for the user, the wood platform will be covered in protective plastic to serve as "bumpers" in case of collision. As simple as the design may look, it employs all the necessary elements of an aesthetically simple but effective product. The electronics and PCB will be housed in a cavity within the wood base itself, shrouding it from external interference as well as making it more pleasing to the eye.

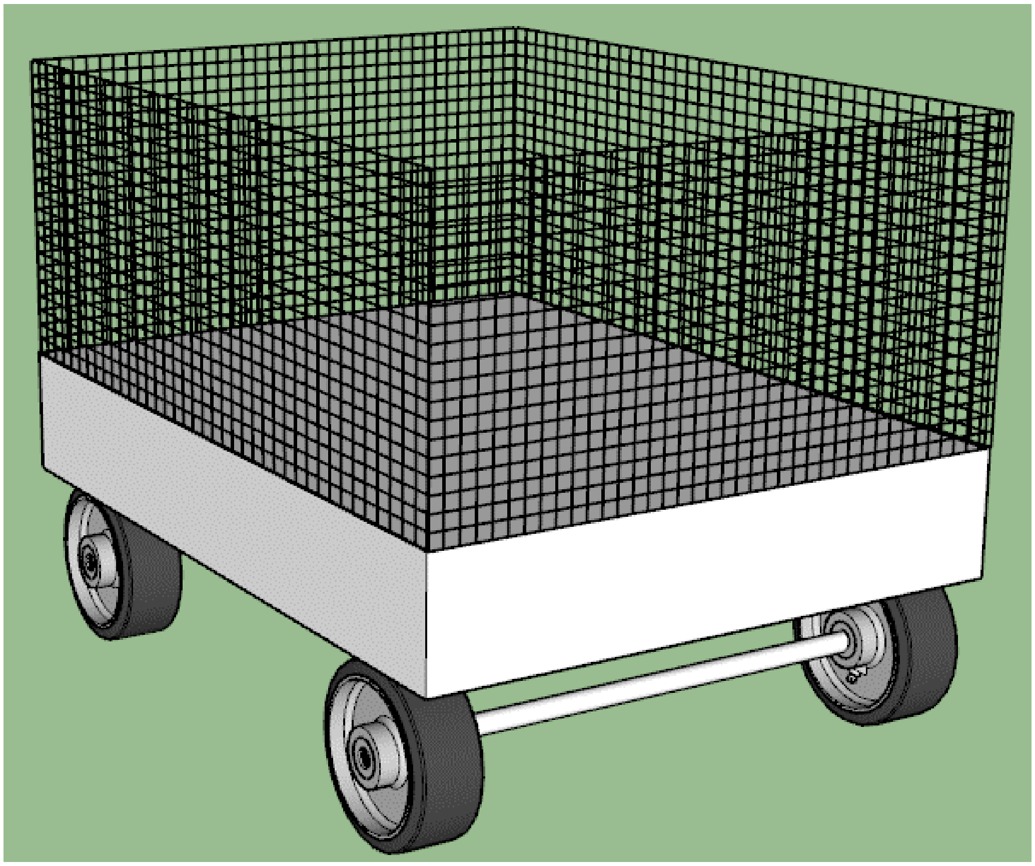


Figure 45: Sample Render of Product Design

The design above shows a simple example of how we want the end product design to look to users. The sketch is over-simplified but still exemplifies the key components of the requirements. This does not, however, include the bumpers or the not visible electric components housed inside. The overall design reflects effectiveness and the prototype will show exactly that.

8.10 Prototype Construction

To build the basic working prototype of the FollowBot, very specific design specifications must be met to maintain efficiency on all fronts. This starts with the robustness and stability of the structure. The wood base will be cut precisely and molded to fit the dimensions. We will ensure that the weight of the wood will not undermine the motors' strength and diminish the quality of the bot's functionality. The wheels will be screwed into the base through a basic system which will allow for swift turns and stops. The metal grate surrounding the user's cargo will be screwed in place and will be able to withstand bumps and falls. To build this design, we will be using a few key facilities: the senior design lab, the Texas Instruments Innovation Lab, and wood cutting services from Home Depot or Lowe's. We will be constructing the physical prototype ourselves and apart from the wood, all of the parts will be cut and shaped in house. The wheels are able to withstand most types of land and will allow us to traverse several types of conditions, but for the demo and prototype our aim is to practice and perform on standard tile or hard wood flooring. The construction of the prototype can mainly be broken down into three steps: gathering the parts, finalizing the design, and assembling the product.

The parts we need for the physical build of the prototype are still relatively up in the air; the type of wood is largely dependent on what we find suitable at a hardware store and the individual pieces such as screws and bolts are largely dependent on the brackets or fixtures we use them with. To begin, the metal grate we need will most likely not be available in exact measurements as we specify. To ail with this, we can purchase a large piece of such structure and cut it to our specifications for all 4 pieces. While this will be significantly harder to work with, the result will be far cleaner and easy for us to adjust. The wood as spoken to before will be attained from a hardware store and the main issues at hand to consider include price, durability, and weight. The perfect balance of these three will yield our best piece for the project, and will lead to a successful demo. The wheels will be attached to the motors along a small metal chassis allowing for maneuverability and sturdy connection to the base. The underside of the wood platform will house a cavity that can store and protect the large amount of electronics going to this device. This will provide a storage for peripherals and allow connections without damaging the aesthetics of the device.

The design is for the most part situated, but elements of this prototype could always change, thus the term prototype. As the construction goes on, we will be testing various parts to see how they react with the parts we have at hand. We want to ensure success by testing at each step, in case calculations were wrong or we found a more elegant solution somewhere. The design process is constantly changing and can constantly be updated for reliability. Once we have a perfect design completed, we can move to the final step, but the design is the most important part and should hardly be overlooked.

To finally assemble the prototype, we are under two assumptions: the design we chose is the end-all final design, and all of the parts we have will fit and succeed in that design. Once we have reached this point, we will be taking all of our parts to the Innovation Lab or a common meeting area and begin assembly. The precision of wheeled robots such as the FollowBot is very important, and each measurement must be meticulously completed and re-checked. We will begin by attaching the chassis to the wood block along with the wheels. To follow, we need to ensure that the housing for the electronic parts successfully makes it onto the frame, and holds sturdy. Once we have the core and underside of the platform done, we will screw on the metal grate enclosing the top. Although the process seems simple, each step must be executed with care and precision to ensure a smooth run. Once we have the prototype built, we will regularly stress test it in weeks leading up to the demo to make sure that no parts falter, and that we can make any changes if necessary. A smooth demo is best, and we do not plan on stumping that due to poor construction quality.

8.11 Prototype Schematic

This section will contain the overall schematic and connections for the motor controller, Bluetooth module, magnetometer, and the ultrasonic sensors. This schematic will utilize the Arduino Uno however the final project design and PCB will utilize the Atmega328p MCU unit instead of the Arduino. The pin layouts are essentially the same.

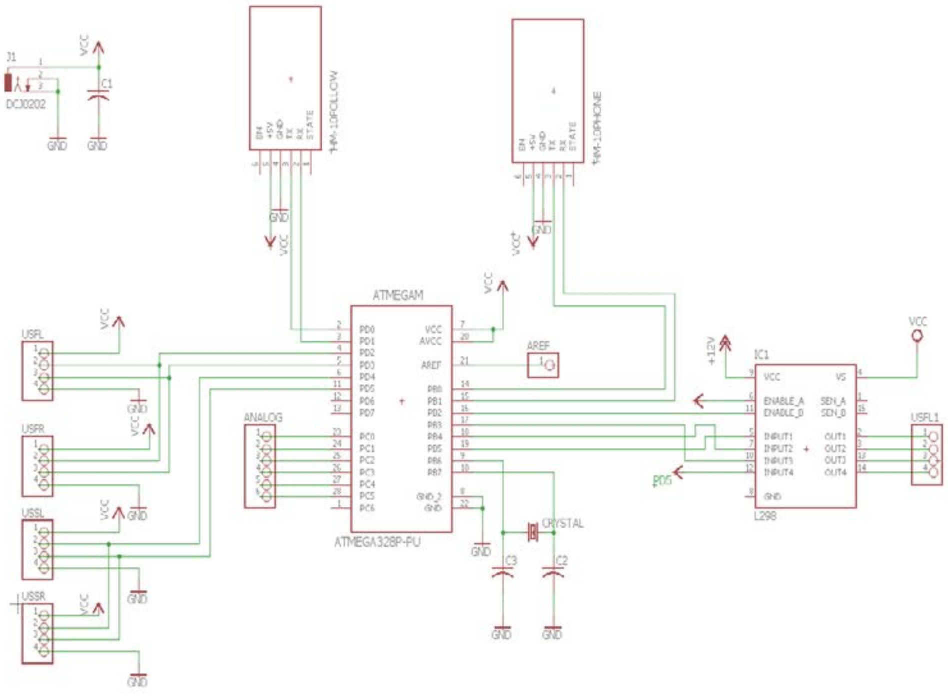


Figure 46: Schematic for Prototype Unit Connections

8.12 PCB Draft

For the PCB design, there were a plethora of programs that could be utilized in order to create a PCB schematic. These include but are not limited to Eagle, Diptrace, Orcad, etc. Diptrace has three tiers of software packages which include the starter, standard, and extended packages. The only difference between each tier is the total number of pins that can be placed for production as well as the pricing. The starter package costs $75, the standard package costs $495, and the extended package costs $695. Orcad in comparison has no pin limit when purchased however, it costs $2300. For this design, Eagle will be used as it is both free for students and offers a large library of parts. The community for Eagle is what is impressive as it is very easy to get help as well as well as find a library that contains a part you are purchasing. The PCB shows a complete and final design that includes not only the DC jack but also mounting holes as well as a crystal oscillator for the MCU. While the PCB is relatively simple this is due to the fact that we had a plethora of peripheral devices extending from the PCB to various areas of the cart.

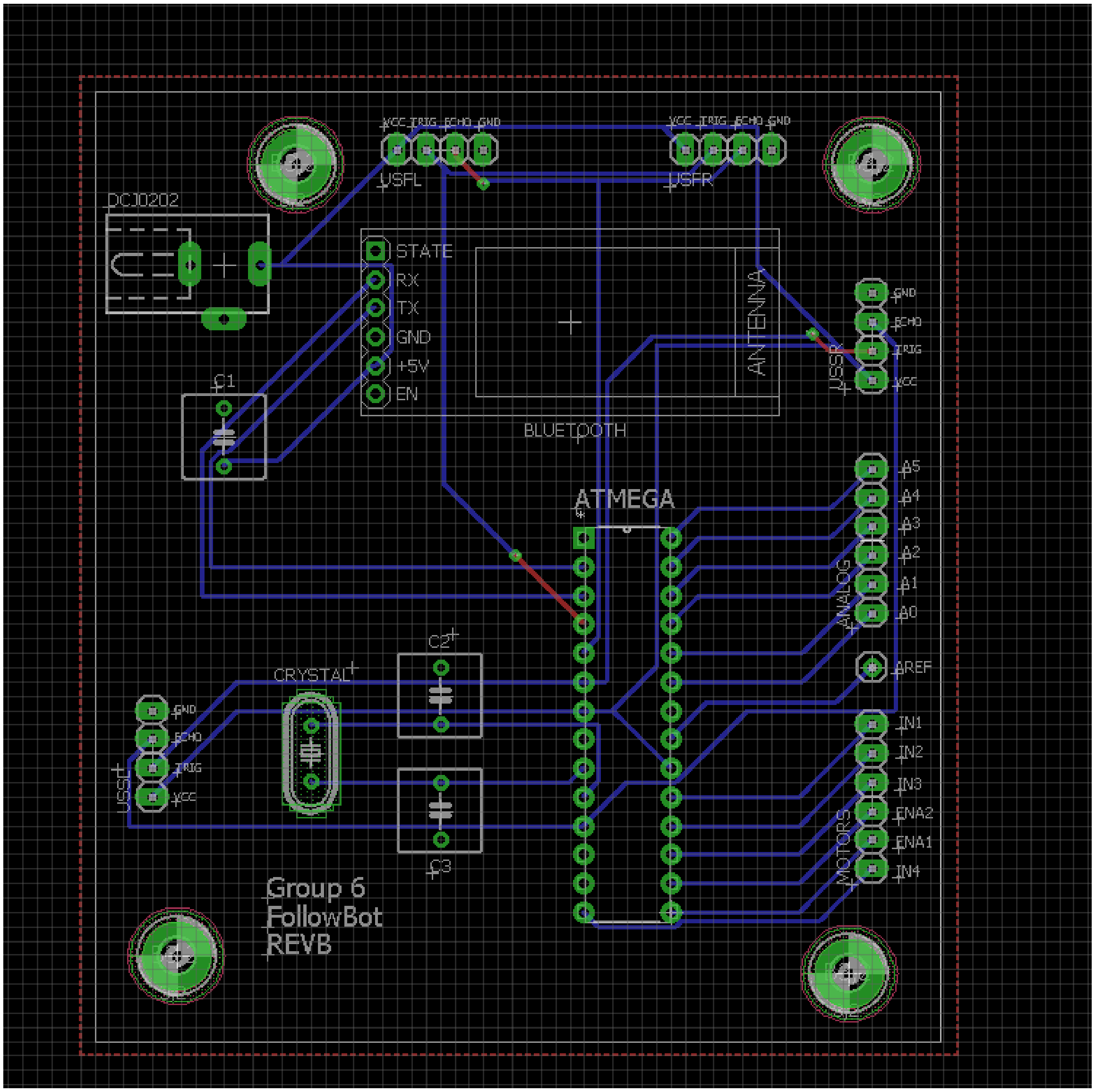


Figure 47: Eagle Layout of Movement Subsystem PCB

8.13 Application Mockup

Creating an exemplary user interface is one of our biggest priorities, and our mobile application is the forward-facing product to the majority of the audience. People of all ages and backgrounds should be able to download the app and use it to their best ability almost instantly. The user should be able to sign up and log in immediately with an internet connection and a nearby FollowBot station. When a user goes on their app and checks out a FollowBot, they will be presented with a screen to pair to their bot and manage it directly through the app. As seen below, the user interface for the log in screen is relatively pedestrian, and follows a template set by most apps on the market. The main control screen shows some of the relevant controls a user might find useful while using the FollowBot, and allows total control over their paired device. After a user is done using the product, they can unpair and go about their day while the app takes care of the rest.

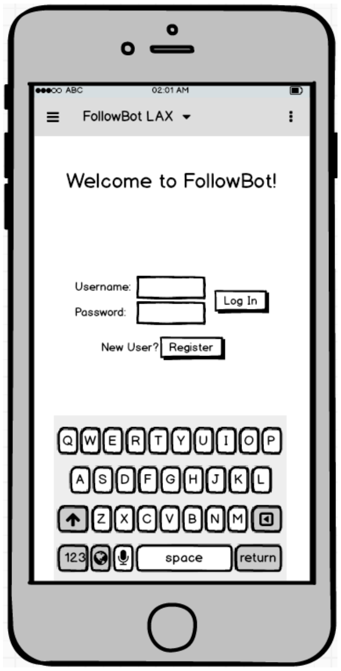
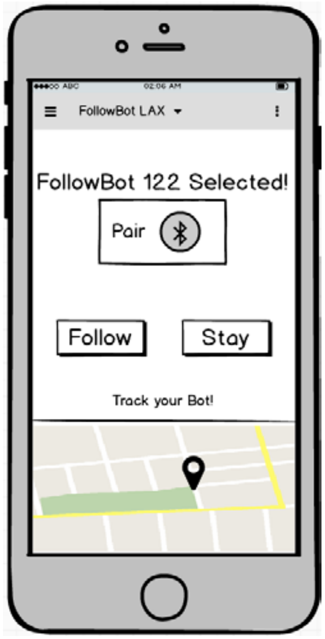

Welcome to FollowBot!

Figure 48: Login Screen Mockup


FollowBot 122 Selected!

Track your Bot!

Figure 49: Control Screen Mockup

8.14 Project Operation

The FollowBot prototype aims to provide a user with a seamless carrying companion that can assist them in carrying burdensome cargo across various distances. User experience is a priority, and achieving seamless integration between host and device is both exciting and rewarding. To begin, a user should approach the idle FollowBot with their handheld Bluetooth capable device. The user must first download the FollowBot app from their desired platform application store and prepare for pairing. The user should now be able to sign in to their account on the mobile interface and begin the pairing process. An idle FollowBot will be constantly searching for new devices to pair with, and associate links based on information relayed by the Bluetooth connection through the app.

Once the pairing process is complete, the user is now on their way to burden-free travel. After the device and the user's mobile phone are paired, the FollowBot will search for beacons around the area to gauge location and initialize its following capabilities. The user will designate the follow mode and will then proceed to be closely shadowed. Travelling through crowds and obstacles, the FollowBot should keep up with haste and avoid any possible collisions along the way providing the user with an efficient and safe experience. The user should limit the use of FollowBot sessions to avoid draining too much battery and should always stay a safe distance away from the moving object while walking. Once the user has reached their destination, they will be able to disengage the device with a simple touch of a button on their handheld device, severing the link and sending the FollowBot back on its way. This will then be sent back to holding to allow for the next person to re-engage the FollowBot while it charges and readies for a new journey.

Procedure:

1. *Find the nearest FollowBot station in your area and download the official app from the Android or iOS App Stores*
2. *Sign up for a FollowBot account which will then be connected to your mobile device*
3. *Use the interface to find available FollowBots at the nearest station*
4. *Checkout and use the pair button to connect to your linked FollowBot*
5. *Once the devices are linked, the robot should immediately begin trailing*
6. *Feel free to walk to your desired path now as your mobile companion follows*

9.0 Cost Analysis & Budget

This project will not be supported by any sponsors. Each group member will contribute to the total amount needed to develop this project which is shown in the table above. However, the amount that each group member will contribute has not been discussed yet. As shown below, the maximum amount predicted for this project is around $800. Our goal is to not exceed pass the predicted maximum total. Since at this moment our budget is being estimated, we plan to minimize the entire cost of our project as much as possible.

It is expected that the majority cost of this project will be spent on PCB manufacturing. This will be determined by the number of components we design on our PCB, and the company that we choose to manufacture our PCB. Miscellaneous components such as resistors and capacitors can and should be purchased in quantities. For hardware components, this is an important factor to consider for minimizing the overall cost. Software assets are expected to require absolutely no dollar value. This is because our project will assimilate a free programming software.

It was important for us to maintain a reasonable budget but make sure that the parts we were getting could handle our specifications. We wanted to make sure we did not just go for cheap and ineffective parts because it would reduce the overall functionality and quality of our design. We aimed for the best sensors for what we needed, the best motors we could find in our spec range, and decided on the microcontroller that fit our needs the best. By agreeing on a budget that was still high but not past our affordability, we believe we can build the product to exactly how we want it. Since we are operating without a sponsor, we cannot spend with the same intention as most others. We must be careful about wasting parts, and we must be especially mindful about wasting tools. Our hardware is very important and must be structurally sound as well as able to carry the weight we put on it so wasting any tools that may be in our advantage would be very negatively impactful.

In terms of software, most development will be completely free. The IDE’s have no cost for students and publishing an app is free for everything but iOS. We may decide to deploy to iOS but may skip this for the demo. The budget includes development costs, but since these are basically non-existent, all our budget can go towards the actual hardware. This is helpful, since the components are not cheap, and each part can go a long way. Our budget includes parts that we might not necessarily use however, we increased the quantity for certain parts as well as overestimated the PCB to purposefully provide a high estimate. This allows for a worst-case budget or essentially a budget that accounts for the failure of multiple parts and components. It also does not include capacitors, diodes, and resistors and other minor components such as jumper wires or solder/flux.

Table 25: Budget Breakdown

BUDGET			
PARTS	AMOUNT	QUANTITY	TOTAL
Platform (Physical Design)	$50	1	$50
Wheels	$10	4	$40
Motors	$15	4	$60
Motor Drive Controller	$3.16	5	$15.78
Ultrasonic Sensors	$1.8	1-5	$9
Weight Sensor	$10	1	$10
Bluetooth	$20	1	$20
Arduino Uno R3	$20.99	1	$20.99
Voltage Regulators	$1.50	5	$7.5
PCB	$150-300	1	$150-300
Battery Charger (Wall Mount)	$52	1	$52
Rechargeable Battery	$4.50	5	$22.50
Subtotal			$607.77
Predicted Maximum Total			$800

9.1 Milestones

Given that in today's day in age many engineering companies operate in Scrum due to its flexible and highly productive nature, we will be planning our process in sprints rather than in milestones. These sprints will take between one and two weeks.

<u>Senior Design One Sprints:</u>

Sprint 1: Begin research on compatible parts for each block of the project and software options for each of major hardware components. Create a spreadsheet for potential microcontrollers.

Have final recommendation on which components we should use.

June 11th-June18th

Sprint 2:

Software- Research, design, and document languages, framework, architecture and connections to potential GPS tracking frameworks such as follow me. Research the App and how it will work

Hardware- Research and design for wireless receiver and gps tracker or Bluetooth receivers. Create spreadsheet for potential Bluetooth/gps systems. Have final recommendation on which components we should use.

June 18th-June25th

Sprint 3:

Software- Research, design, and document of the tracking block and the logistics behind connecting to the Collision/Course Control Block.

Hardware- Research and design for the ultrasonic sensor components and how those components will talk with the microcontroller. Create spread sheet for ultrasonic collision detection components. Have final recommendation on which components we should use.

June 25th-July 2nd

Sprint 4:

Software- Research, design, and document motor control software and logistics behind connecting to the Collision/Course Block

Hardware- Research and design the platform, motor, and tires for the system. Create spreadsheets for each component needed. Have final recommendation on which components we should use.

July 2nd-July 9th

Sprint 5:

Software- Research, design, and document the Collision/Course Control software and functions to take in inputs and deliver commands to the rest of the project

Hardware-Research and design the PCB, power source requirements and voltage regulators.

July 9th-July16th

Sprint 6: Review and refine the designs and documentation from the earlier sprints.

July 16th-August 1st

9.3 Conclusion and Closing Statement

The goal we set forth from the beginning of this idea was to make a creative way for more convenience at the airport. We kept expanding upon this until we realized that FollowBot was an elegant solution to a popular problem. Not only are we aiming to improve the public market and general happiness across airports, we believe it can expedite foot traffic and relieve blockages in many of our country's biggest airports. By creating a mobile platform capable of carrying luggage and much more, we believe that FollowBot can even have applications beyond Senior Design. We want to prioritize safety and efficiency – this has been repeated and reiterated – because it is so important to the idea of our design. Areas with large foot traffic can easily be seen as a huge challenge for an implementation such as this, and we aim to surmount that in our demo. We believe that our choices in design have been consistent with the consumer market as well as strategically viable in real-world production. As this idea expands, we hope it will be applied to different fields due to its versatility.

The challenge of FollowBot starts and ends with survivability in a large traffic environment such as an airport. A working product with limitations such as amount of people would render it useless for a large number of cases. We are directly tackling this issue by ensuring that the development of the code for collision detection and avoidance is state-of-the-art. With avoiding collisions being the priority of our code, maneuverability is the priority of our hardware. The decisions for parts made for the hardware are specifically related to being able to manage multiple conditions as well as being able to maintain and ensure the success of the overall product. The wheels were specifically chosen to be adaptable to multiple environments while the battery was chosen to last long enough for an effective demo and even more. The microcontroller was chosen to speed up development and even small components such as ultrasonic sensors were chosen with very specific applications in mind. Ultimately, while Bluetooth tracking is incredibly useful for proximity our final project and results demonstrates that it is not suitable or reliable for precise indoor tracking and positioning. It is equally as bad as the backbone for the motion system of a robot and other types of indoor tracking such as Chronos will surpass Bluetooth easily in popularity in the near future.

10.0 References

[1] Rhodes, Trey. "Global Positioning System Overview." *The Global Positioning System*. University of Texas, n.d. Web. 01 Aug. 2017.

[2] "GNSS Frequently Asked Questions - GPS." *FAA seal*. N.p., 20 Dec. 2016. Web. 01 Aug. 2017.

[3] Vasisht, Deepak, Swarun Kumar, and Dina Katabi. "Decimeter-Level Localization with a Single Wifi Access Point." *Proceedings of the 13th USENIX Symposium*(2016): 1-3. Web. 6 June 2017. <https://www.usenix.org/system/files/conference/nsdi16/nsdi16-paper-vasisht.pdf>.

[4] Curt Franklin & Julia Layton "How Bluetooth Works" 28 June 2000. HowStuffWorks.com. <http://electronics.howstuffworks.com/bluetooth.htm> 1 August 2017

[5] "Understanding the Different Types of BLE Beacons." *Understanding the different types of BLE Beacons | Mbed*. Blackstone Engineering, 24 Mar. 2015. Web. 01 June 2017.

[6] S, Adarsh, and Mohammed Kaleemuddin. "Performance comparison of Infrared and Ultrasonic sensors for obstacles of different materials in vehicle/ robot navigation applications ." *Materials Science and Engineering*149 (2016): 1-5. *IOPScience*. Web. 1 July 2017.

[7] Janocha, Hartmut. *Actuators: Basics and Applications.* Berlin: Springer, 2010. Print.

[8] Benson, Coleman. "How to Make a Robot – Lesson 3: Making Sense of Actuators." *RobotShop.* N.p., 2 Sept. 2010. Web.

http://www.robotshop.com/blog/en/how-to-make-a-robot-lesson-3-actuators-2-3703

[9] "Lorentz Force." *Encyclopædia Britannica.* Encyclopædia Britannica, Inc., n.d. Web. https://www.britannica.com/science/Lorentz-force.

[10] "How a DC Motor Works?" *Electricaleasy.com*. N.p., n.d. Web.http://www.electricaleasy.com/2014/01/basic-working-of-dc-motor.html

[11] "Gearbox Guide." *Gearboxes | Planetary and Spur Designs from Anaheim Automation*. N.p., n.d. Web. http://www.anaheimautomation.com/manuals/forms/gearboxguide.php#sthash.oMbkEMKE.dpbs.

[12] Reed, Frances. "How Do Servo Motors Work." *How Servo Motors Work*. N.p., n.d. Web. http://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html.

[13] Earl, Bill. "All About Stepper Motors." *What Is a Stepper Motor? | All About Stepper Motors | Adafruit Learning System.* N.p., n.d. Web. https://learn.adafruit.com/all-about-stepper-motors/types-of-steppers.

[14] George, and Roy Schahrer. "Unipolar Stepper Motor vs Bipolar Stepper Motors." *Simply Smarter Circuitry Blog*. N.p., 30 Mar. 2016. Web. https://www.circuitspecialists.com/blog/unipolar-stepper-motor-vs-bipolar-stepper-motors/.

[15] Benson, Coleman. "How to Make a Robot – Lesson 5: Choosing a Motor Controller." *RobotShop*. N.p., 10 Nov. 2010. Web. http://www.robotshop.com/blog/en/how-to-make-a-robot-lesson-5-motor-controller-3695.

[16] "Robot Platform | Knowledge | Types of Robot Wheels." *RSS*. N.p., n.d. Web.http://www.robotplatform.com/knowledge/Classification_of_Robots/Types_of_robot_wheels.html.

[17] Keeping, Steven. "Understanding the Advantages and Disadvantages of Linear Regulators." *Understanding Advantages Linear Regulators | DigiKey*. DigiKey, 08 June 2012. Web. 01 July 2017.

[18] Buchmann, Isidor. "What's the Best Battery?" *Basic to Advanced Battery Information from Battery University*. Battery University, 21 Mar. 2017. Web. 01 July 2017.

[19] "Can someone clarify how to charge a battery without an off-the-Shelf battery charger?" *Batteries - Can someone clarify how to charge a battery without an off-the-Shelf battery charger? - Electrical Engineering Stack Exchange*, electronics.stackexchange.com/questions/76621/can-someone-clarify-how-to-charge-a-battery-without-an-off-the-shelf-battery-cha.

[20] Mobile Tech Force, "2016's Best Programming Languages For Mobile Development. (2017, September 10). Retrieved July 7, 2017. Web. http://www.mobiletechforce.com/programming-languages-for-mobile/

[21] Velji, Jamil. "14 Programming Languages for Mobile App Development"(2016, December 19). Retrieved July 7, 2017. Web. https://buildfire.com/programming-languages-for-mobile-app-development/

[22] Version Eye. “Language Statistics : Java and Ruby are leading”(2013, January 17) Retrieved July 7, 2017. Web. https://blog.versioneye.com/2013/01/30/language-statistics-java-and-ruby-are-leading/

[23] “Lemmingapex/Trilateration”(Ongoing). Retrieved July 7, 2017. Web. https://github.com/lemmingapex/Trilateration

[24] PYPL Github “Popularity of Programming Language” Retrieved July 7, 2017. Web. http://pypl.github.io/PYPL.html

[25] Khanna, Nitin “Building Hybrid Android Apps with Java and Javascript” Retrieved July 7, 2017. Web. https://www.safaribooksonline.com/library/view/building-hybrid-android/9781449361907/ch01.html

[26] Statista.com “Forcasted Unit shipments of smartphones from 2016to 2021” Retrieved July 7, 2017. Web. https://www.statista.com/statistics/309448/global-smartphone-shipments-forecast-operating-system/

[27] “N00D13/Indoor GPS” (Ongoing). Retrieved July 7, 2017. Web. https://github.com/N00D13/IndoorGPS

[28] “objc2j” (Ongoing). Retrieved July 7, 2017. Web. https://code.google.com/archive/p/objc2j/

[29] Apportable. Retrieved July 7, 2017. Web. http://www.apportable.com/

[30] Rowinski, Dan. “The programming and engineering skills with highest salaries” (2015, March 13). Retrieved July 7, 2017. Web. http://www.businessinsider.com/the-programming-and-engineering-skills-with-the-highest-salaries-2015-3

[31] Swift “About Swift” Retrieved July 7, 2017. Web. https://swift.org/about/

[32] Raghavan N Aswin “Accurate Mobile Robot Localization in indoor environments using Bluetooth” Retrieved July 7, 2017. Web. http://www.cse.iitm.ac.in/~ravi/papers/Ashwin_ICRA10.pdf

[33] Vasisht, Deepak “Decimeter-Level Localization with Single WiFi Access Point” (2016, March 16-18). Retrieved July 7, 2017. Web. https://www.usenix.org/system/files/conference/nsdi16/nsdi16-paper-vasisht.pdf

[34] Setlur P. “Multipath Exploitation in Through-Wall Radar Imaging Via Point Spread Functions.” IEEE Transactions on Image Processing, 2013.

[35] Turner, Craig "Homemade Flying Cooler Drone" (2017, June 3). Retrieved July 7, 2017. Web. https://www.youtube.com/watch?v=NZDYx31nDsI

[36] Mashabe News "Smart Cooler" (2017, June 17). Retrieved July 7, 2017. Web. https://www.youtube.com/watch?v=wpRD0m9_1ys

[37] "47 CFR Part 15, Subpart B - Unintentional Radiators." *LII / Legal Information Institute*. N.p., n.d. Web. 01 Aug. 2017.

[38] "Power Supply Safety Standards." *CUI Inc*. N.p., 1 Sept. 2014. Web. 1 July 2017.

[39] Mammano, Bob, and Lal Bahra. "Safety Considerations in Power Supply Design." (n.d.): n. pag. *IEEE*. IEEE, 05 Jan. 2005. Web. 1 July 2017.

[40] ANSI/IEEE Std 1106-1987, IEEE Recommended Practice for Maintenance, Testing, and Replacement of Nickel-Cadmium Storage Batteries for Generating Stations and Substations.

[41] ANSI/IEEE Std 1115-1992, IEEE Recommended Practice for Sizing Nickel-Cadmium Batteries for Stationary Applications.

[42] Yoder, John A . "Primer on Lead-Acid Storage Batteries." *DOE Handbook*95 (1995): 27-34. *Energy.gov*. Web. 1 July 2017.

[43] Rossi, Sergio. "How to get your Bluetooth Design FCC Certified." *ST Life*. N.p., 25 Jan. 2017. Web. 1 July 2017.

[44] "Wiring & Driving the L298n H Bridge with 2 to 4 DC Motors." *14Core.Com*, www.14core.com/wiring-driving-the-l298n-h-bridge-on-2-to-4-dc-motors/.

[45] Datasheet made by RobotShop for the Cytron 12V 150RPM 18oz-in Spur Gearmotor.http://www.robotshop.com/media/files/pdf2/034-rb-cyt-28.pdf.

[46] Datasheet made by RobotShop for the Cytron 12V 17RPM 194.4oz-in Spur Gearmotor.http://www.robotshop.com/media/files/pdf2/009-rb-cyt-29.pdf.

[47] "Pololu - Dagu Wild Thumper Wheel 120x60mm Pair with 4mm Shaft Adapters - Metallic Red." *Pololu Robotics & Electronics*, www.pololu.com/product/1555.

[48] "Pololu – 12mm Hex Wheel Adapter for 6mm Shaft (2-Pack)." *Pololu Robotics & Electronics*, www.pololu.com/product/2686.

[49] "Pololu – Pololu Stamped Aluminum L-Bracket Pair for 37D mm Metal Gearmotors." *Pololu Robotics & Electronics*, www. www.pololu.com/product/1084.

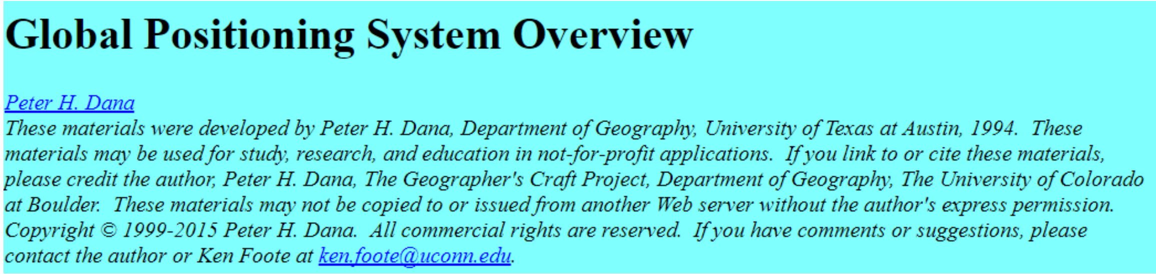
[50] "Advanced Circuits." *Printed Circuit Board Manufacturer & PCB Assembly | Advanced Circuits*, www.4pcb.com/.

[51] (CustomCircuitBoards), Zachary Burruel. “Custom Circuit Boards.” *Custom Circuit Boards*, www.customcircuitboards.com/.

[52] Cawley, Christian, et al. “How Does RFID Technology Work?” *MakeUseOf*, 31 May 2017, www.makeuseof.com/tag/technology-explained-how-do-rfid-tags-work/.

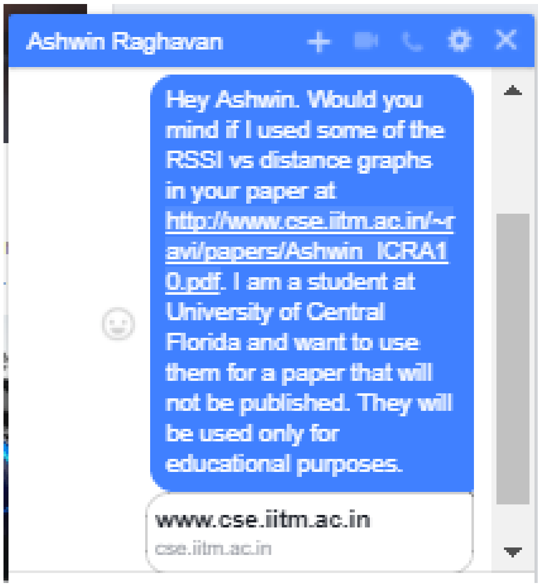
[53] Schweber, Bill. “Understanding the Advantages and Disadvantages of Linear Regulators.” *Understanding Linear Regulator Advantages | DigiKey*, 20 Sept. 2017, www.digikey.com/en/articles/techzone/2017/sep/understanding-the-advantages-and-disadvantages-of-linear-regulators.

10.1 Appendix A: Permissions



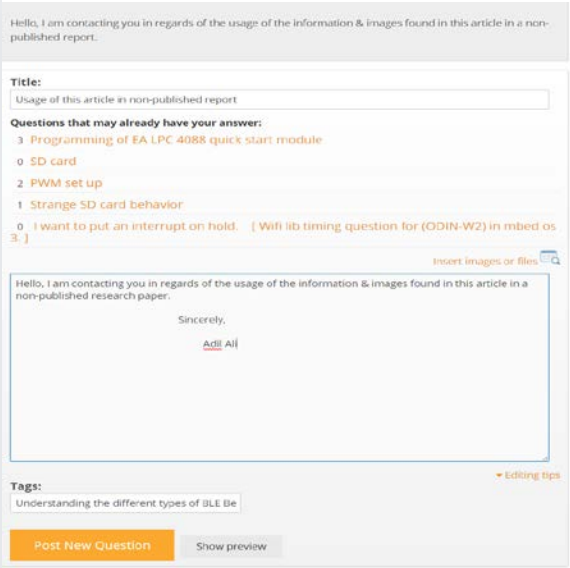
Global Positioning System Overview

Peter H. Dana
These materials were developed by Peter H. Dana, Department of Geography, University of Texas at Austin, 1994. These materials may be used for study, research, and education in not-for-profit applications. If you link to or cite these materials, please credit the author, Peter H. Dana, The Geographer's Craft Project, Department of Geography, The University of Colorado at Boulder. These materials may not be copied to or issued from another Web server without the author's express permission. Copyright © 1999-2015 Peter H. Dana. All commercial rights are reserved. If you have comments or suggestions, please contact the author or Ken Foote at ken.foote@uconn.edu.



Hey Ashwin. Would you mind if I used some of the RSSI vs distance graphs in your paper at http://www.cse.iitm.ac.in/~ravi/papers/Ashwin_ICRA10.pdf. I am a student at University of Central Florida and want to use them for a paper that will not be published. They will be used only for educational purposes.

www.cse.iitm.ac.in
cse.iitm.ac.in


Hello, I am contacting you in regards of the usage of the information & images found in this article in a non-published report.

Title: Usage of this article in non-published report

Questions that may already have your answer:

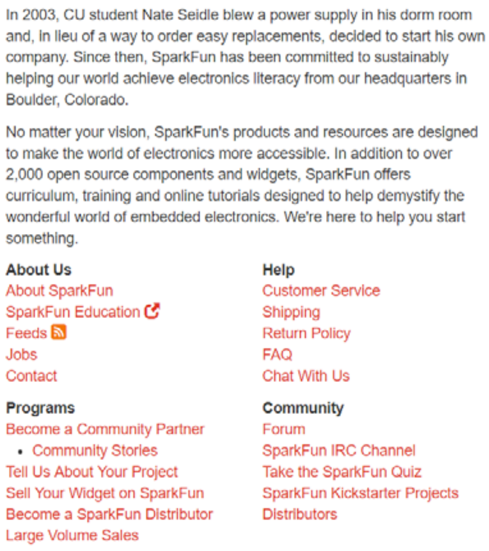
3	Programming of EA LPC 4088 quick start module
0	SD card
2	PWM set up
1	Strange SD card behavior
0	I want to put an interrupt on hold. [Wifi lib timing question for (ODIN-W2) in mbed os 3.]

Hello, I am contacting you in regards of the usage of the information & images found in this article in a non-published research paper.

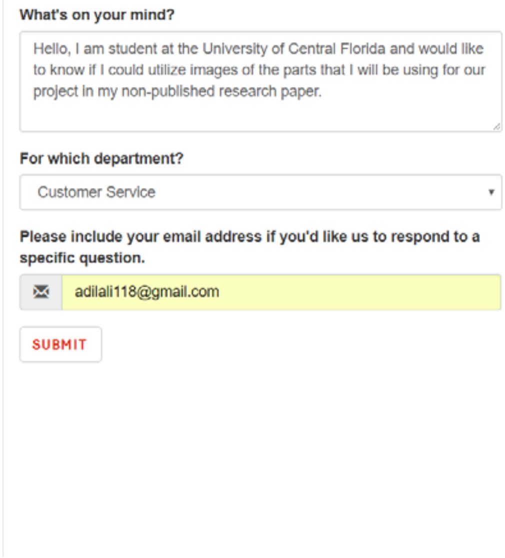
Sincerely,

Adil Ali

Tags: Understanding the different types of BLE Be


In 2003, CU student Nate Seidle blew a power supply in his dorm room and, in lieu of a way to order easy replacements, decided to start his own company. Since then, SparkFun has been committed to sustainably helping our world achieve electronics literacy from our headquarters in Boulder, Colorado.

No matter your vision, SparkFun's products and resources are designed to make the world of electronics more accessible. In addition to over 2,000 open source components and widgets, SparkFun offers curriculum, training and online tutorials designed to help demystify the wonderful world of embedded electronics. We're here to help you start something.


What's on your mind?

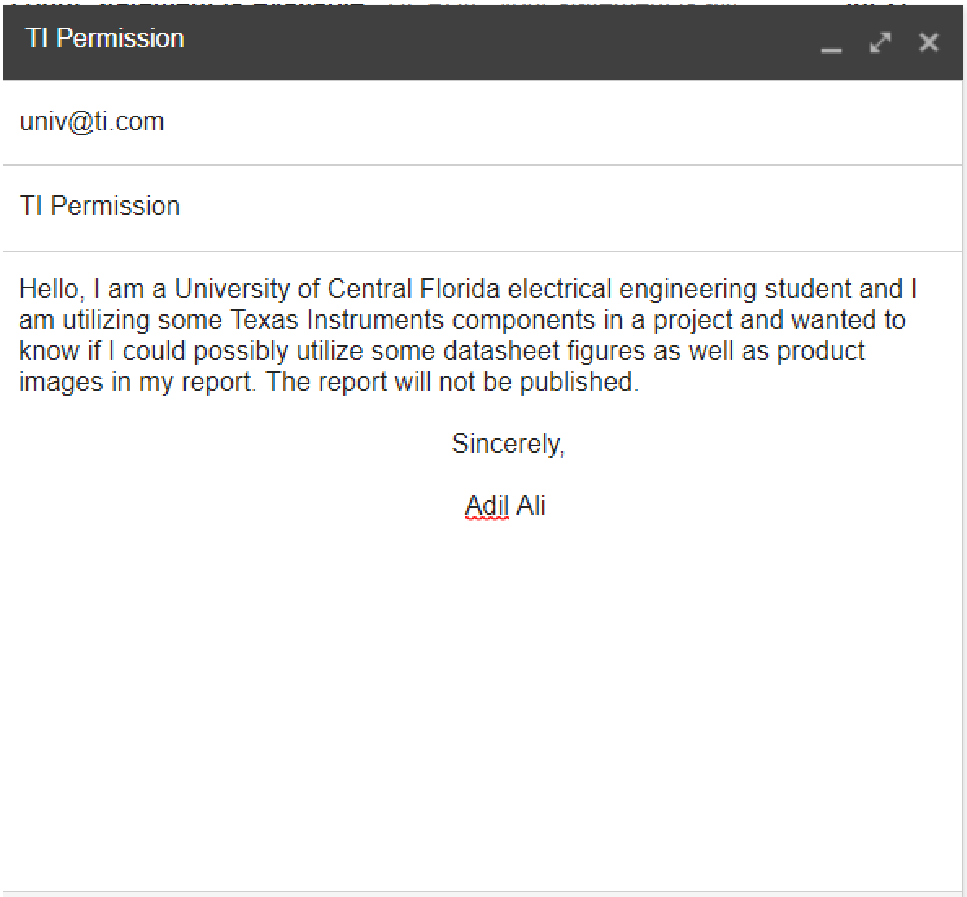
Hello, I am student at the University of Central Florida and would like to know if I could utilize images of the parts that I will be using for our project in my non-published research paper.

For which department?

Customer Service ▾

Please include your email address if you'd like us to respond to a specific question.

✉ adilali118@gmail.com


TI Permission

univ@ti.com

TI Permission

Hello, I am a University of Central Florida electrical engineering student and I am utilizing some Texas Instruments components in a project and wanted to know if I could possibly utilize some datasheet figures as well as product images in my report. The report will not be published.

Sincerely,

Adil Ali

10.2 Breadboard Testing Pictures

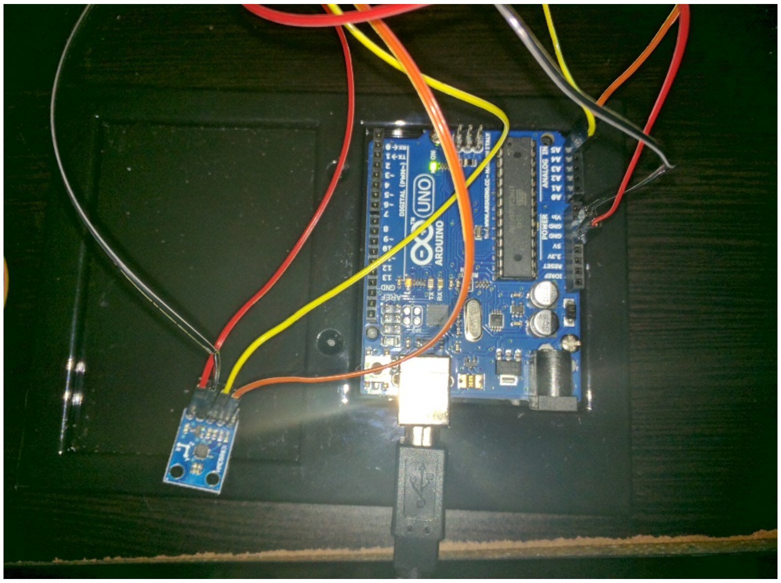


Figure 50: Bluetooth Test Setup

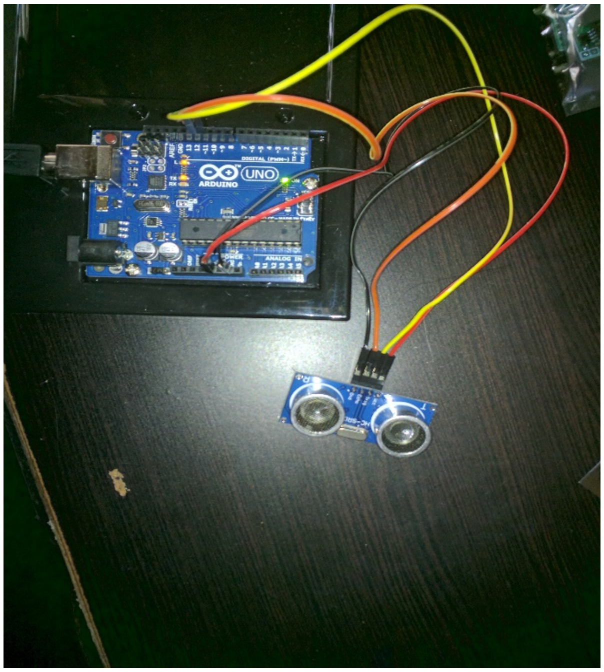


Figure 51: Ultrasonic Sensor Test Setup