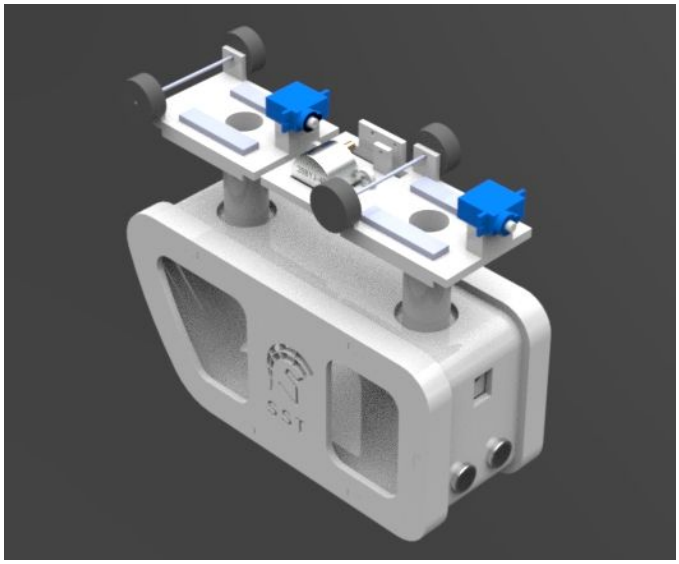
Small Scale Team Fall 2019 Report

ME 195A, Section 4



By:

Asmaa Darwish, Lissette Romero,

Julio de Pereda Banda, Justin Ghieuw, Shane Fatehi

Dr. Burford J. Furman

December 14, 2019

Charles W. Davidson College of Engineering

San Jose State University

Abstract

This semester SPARTAN Superway was given the task to improve prior years designs. As it will be discussed in further detail in this report, the Small Scale Team is divided into three subteams: guideway, bogie and controls. Each of these subteams had team specific goals which set a clear path as to how to proceed throughout the semester. The guideway team made progress on designing a reliable, interchangeable and practical track with an inner and outer loop. The bogie team designed a bogie and pod which will reduce the printing time and make the pod easier to fix if needed. The controls team designed an anti-collision system for the pods along with an iOS app to control the pods. Overall, the main goal of the 2019/2020 Small Scale Team is to design a complete small scale model which will gain the interest of the public, as well as investors.

Each subteam made progress this semester by designing and prototyping different aspects of the small scale model. The guideway team designed several 3D CAD iterations of the track. A sample piece of one of the outer and inner turns was also manufactured by hand cutting leftover plywood from previous years. Based on this first prototype, it was determined that a better manufacturing process was needed to create precise and clean cuts on the plywood. The bogie team also designed a 3D CAD model of what the pod and bogie will look like once 3D printed. Meanwhile, the controls team created several Arduino codes to test the functionality of the anti-collision system by using ultrasonic sensors. In addition to this, two iterations of the iOS app were developed and tested using Apple's app building software Xcode. A lot of research has been done to understand how to connect the iOS app to the arduino using HM-10 BLE module. Different arduino codes has been conducted to connect the BLE.

From the design stage to the prototyping stage, the Small Scale Team found several pieces of knowledge and information which will facilitate the continuation of the project in the upcoming Spring 2020 semester. With the design and first prototype of the guideway, it was found that the current plywood supplied at the SPARTAN Superway center was not suitable for water jet cutting due to warping, and that a better grade of plywood was needed. In addition to this, some of the curves of the inner and outer turns were adjusted so that they were not as sharp of a turn. The bogie team found that the new bogie and pod design will increase the versatility and accessibility of past bogies and pods. Lastly, the controls team found that it was a challenge to connect the developed iOS app with the Arduino. Overall, the small scale team analyzed positive aspects of the project, as well as the parts of the project which will need fine tuning.

Acknowledgments

The Small Scale Team would like to thank and acknowledge Dr. Burford Fuman, Mr. Ron Swenson and Eric Hamstrong for their help this first semester. A special thanks to Mr. Swenson for providing SPARTAN Superway teams with a place to work on the project. Thank you Jacques-Hariel Ango and Bill James for providing suggestions and examples for building the guideway.

Table of Contents

Abstract 2

Acknowledgments 3

Table of Contents 4

List of Figures 6

List of Tables 9

Executive Summary 10

1. Project Description 12
    1.1 Current problems 12
    1.2 Societal Needs and Benefits 14
    1.3 Objectives 16
    1.4 Project Structure 16

2. State-of-the-Art Review 17
    2.1 ATN around the world 17
        2.1.1 Wuppertal Suspension Railway 18
        2.1.2 Morgantown Personal Rapid Transit System 19
    2.2 ATN in SJSU 21

3. Design Solutions 25
    3.1 Requirements and Specifications 26
        3.1.1 Guideway Specification 26
        3.1.2 Bogie Specifications 27
        3.1.3 Controls Specifications 27
    3.2 Prime Design 28
        3.2.1 Guideway Design 28
        3.2.2 Bogie Design 30
        3.2.3 Controls Design 34
            3.2.3.a iOS App 34
            3.2.3.b Arduino Code 39
    3.3 Supporting Analysis 42
        3.3.1 Guideway Analysis 42
        3.3.2 Bogie Analysis 45
        3.3.3 Controls Analysis 47

3.2.3.a iOS App Analysis 47
3.2.3.b Arduino Code Analysis 60
3.4 Plans for Fabrication 62
3.4.1 Guideway Fabrication 62
3.4.2 Bogie Fabrication 63
3.4.3 Controls Fabrication/Testing 64

4. Conclusion and Next Steps 64

References 66

Appendices 69
Appendix A: Ultrasonic Sensor DataSheet 69
Appendix B: Pixy2 DataSheet 71
Appendix C: Servo Motor Data Sheet 71
Appendix D: HM10 DataSheet 72
Appendix E: Motor DataSheet 73
Appendix F: Ultrasonic Test Code 73
Appendix G: BLE Configuration Code 75
Appendix H: BLE Test Code 76
Appendix I: BLE and Ultrasonic Anti-collision Code 77
Appendix J: Small Scale Team Gantt Chart 80
Appendix K: Guideway Material Properties 81
Appendix L: Guideway Bill of Materials 81
Appendix M: Guideway Drawings 82
Appendix N: iOS App First Iteration Code 83
Appendix O: BLE Connection Test Code: 85

List of Figures

Figure 1. Assembly of the first model of the track …….. 9

Figure 2. 3D Modeled Pod and Bogie Assembly ……..... 10

Figure 3. The iOS app loading screen ……..….... 10

Figure 4. Total U.S. Greenhouse Gas emissions in 2017 ……..... 12

Figure 5. Division of Small Scale Team sub teams ……..….... 16

Figure 6. Wuppertal Suspended Railway above traffic ……...… 17

Figure 7. The Wuppertal Suspension Railway above the city river ……..……..… 17

Figure 8. Morgantown vehicle steering system ……..…….. 18

Figure 9. Urban Light Transit PRT in London Heathrow Airport ……...…… 19

Figure 10. Cabinentaxi in Germany stopped developments in …… 20

Figure 11. SPARTAN Superway Full-Scale Cabin ……..

21

Figure 12. Small-Scale Mobile Application in 2016 ……...…… 22

Figure 13. Connection between two tracks ……..... 23

Figure 14. Bill James first prototype of mini-scale model using threaded studs ……..... 28

Figure 15. Flat washer and nut clamping on top and bottom clamping track pieces …….. 29

Figure 16. Guideway visual using threaded studs to suspend..……..... 30

Figure 17. The old switch design and the new switch design …… 29

Figure 18. 3D model of snap fit design ……... 31

Figure 19. Exploded View of Pod Car Design ……..... 32

Figure 20. Assembly of the swivel design …… 33

Figure 21. Render of isometric view of pod and bogie assembly ……... 34

Figure 22. App loading screen ……...…….. 35

Figure 23. Main tab with project and team information …….. 36

Figure 24. Rough design UI design of the app ……..... 37

Figure 25. Current third tab UI …… 38

Figure 26. iOS app logo ……... 39

Figure 27. Anti-collision system electrical configuration ……..... 40

Figure 28. Flowchart of the implementation of the BLE and the anti-collision …….. 41

Figure 29. Electrical hardware configuration for testing the anti-collision code ……..... 41

Figure 30. Fritzing diagram of desired electrical configuration …….. 42

Figure 31. FEA analysis of a straight track in ANSYS, using symmetry ……..... 43

Figure 32. Equivalent stress found using FEA …… 43

Figure 33. Simple fixed beam example ……... 44

Figure 34. Swivel connection press fit …… 46

Figure 35. The current bogie platforms ……... 47

Figure 36. SwiftUI on Xcode and output on desired device ……... 48

Figure 37. Template options for a new app in Xcode ……. 49

Figure 38. Options page in Xcode …….. 49

Figure 39. Snippet of the edited code for the first iteration …… 50

Figure 40. SPARTAN Superway app layout of first app iteration …….. 51

Figure 41. Storyboard layout of iOS app ……..... 52

Figure 42. First tab the user sees after app has loaded …… 53

Figure 43. Meet the Small Scale Team "page" embedded in home tab …….. 54

Figure 44. Map tab ……. 55

Figure 45. Order pod tab …… 55

Figure 46. Order Pod 1 screen ……..... 56

Figure 47. Order Pod 2 screen ……...…. 56

Figure 48. Example of how to read Storyboard map …….. 57

Figure 49. Central and peripheral visual representation ……. 59

Figure 50. Flowchart of BLE test code ……... 61

Figure 51. Connection between Arduino and BLE ……..... 61

Figure 52. Hand cut track pieces ……...….. 63

List of Tables

Table 1. Guideway Design Specifications ……25

Table 2. Bogie Design Specifications ……...26

Table 3. Controls Design Specifications ……...27

Executive Summary

SPARTAN Superway or Solar Powered Automated Rapid Transit Ascendant Network Superway is a project composed of several teams who are all working on developing a new sustainable form of public transportation for Silicon Valley. This years' team consists of full-scale, half-scale, small scale, power module , and wayside power team. The Small Scale Team in particular is developing a 1/12 scale model which will mimic how the full scale Automated Transit Network (ATN) system will operate. The purpose of this model is to serve as a proof of concept, which can be used to market the SPARTAN Superway during fundraising and educate the public about the project by having a physical working system. The final model will include a fully functional guideway, bogie, and a controls system. The small scale team is developing a system which will allow multiple pods to be operating on the track without failure.

So far the Small Scale Team has made significant progress on the development of the guideway, bogie and controls of the 1/12 scale model. The guideway team completed their first 3D model of the track as shown below in figure 1. This design is made to be easy to assemble and have pieces that are interchangeable.

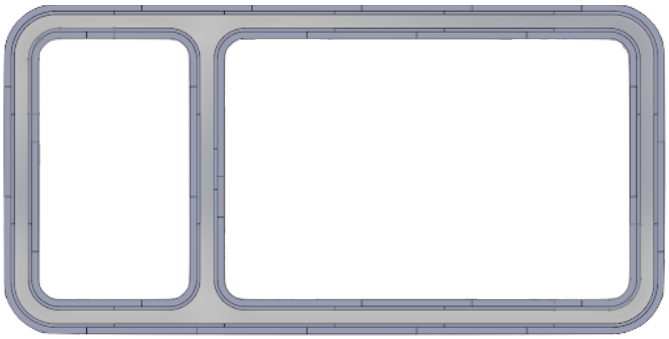


Figure 1. Assembly of the first model of the track.

A refined 3D CAD model of the pod and bogie, shown in figure 2, was also made. This design successfully improves upon the designs of previous work. It incorporates snap fits, swiveling connections and an easily disassemblable body.

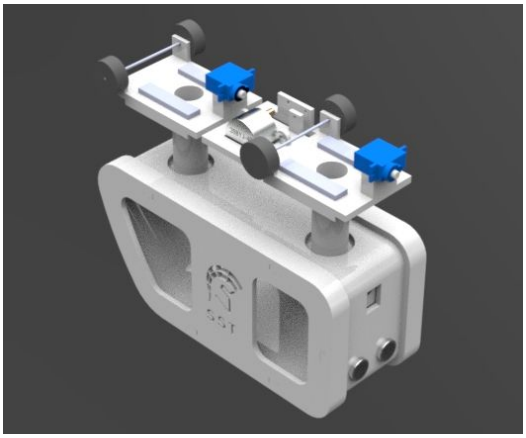


Figure 2. 3D Modeled Pod and Bogie Assembly.

Lastly, the controls team made progress in the app, arduino code, and electrical hardware. The controls system will use an iOS app in conjunction with a bluetooth module to communicate with the pods on the track. The iOS app is in its development stages and the main UI has been built. The loading screen can be seen in figure 3. Several parts of the final arduino code have also been made and tested with the electrical hardware.

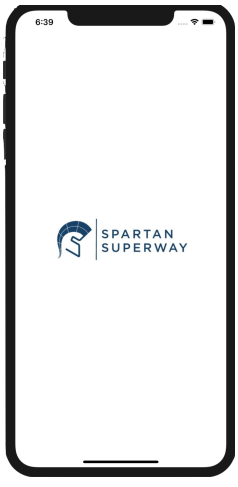


Figure 3. The iOS app loading screen.

1. Project Description

SPARTAN Superway is a project started in 2012 by professor Burford Furman and Mr. Ron Swenson along with students at the SJSU college of engineering. This project is an implementation of automated transit networks or (ATN). This section will discuss the current problems with transportation and its societal impact in the Silicon Valley and the Bay Area. It will explain how SPARTAN Superway is a solution to those problems and how the small scale team plays a role in it. It will also discuss the objectives and structure of the small scale team for the 2019/2020 year.

1.1 Current problems

At the current moment, the planet is facing problems which have not been seen before in earth's history. This is largely due to human activity and the need for an increased amount of resources, specifically fossil fuels. This increased amount of resource usage is due to the massive increase in global population. With an increasing population, there is a higher demand for every kind of resource that humans need to keep advancing socially, technologically, and economically. One of the most demanding sectors of resources used is transportation. In 2017, emissions from transportation accounted for about 28.9 percent of total U.S. greenhouse gas emissions, making it the largest contributor to U.S. greenhouse gas emissions (Sources of Greenhouse Gas Emissions, 2019). Figure 4 illustrates how this percentage compared to other sources of greenhouse gas emissions.

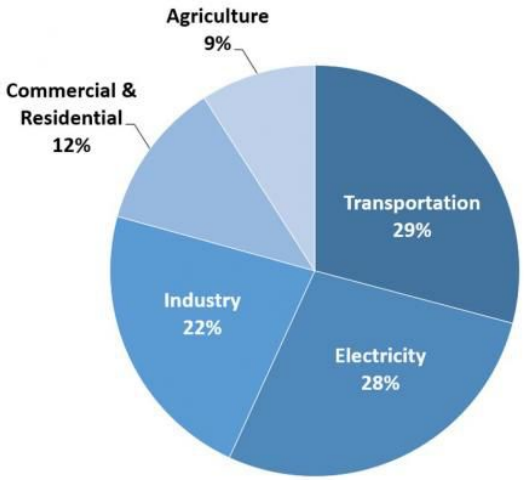


Figure 4. Total U.S. Greenhouse Gas emissions in 2017

With more humans populating the planet, there is an urgency for quicker and more efficient transportation. At the current moment in the United States, people's top choice of transportation is still the automobile. This presents a significant issue in cities such as San Jose, San Diego, San Francisco, and other highly populated areas with the amount of traffic on the road. Not only does traffic congestion present environmental issues, but it increases the stress levels of the people who have to endure it on a day-to-day basis. Traffic also causes people who commute to work on a daily basis to lose anywhere from 42 to 84 hours per year due to sitting in traffic (Andrews, n.d.). This is valuable time that could be used for people to invest in themselves or with their loved ones. Happier and less stressed humans have been proven to be more productive at their respective jobs, as well as with their own personal goals.

In major cities, specifically in the Bay Area, current public transportation alternatives such as BART, VTA Light rail, and buses are not ideal solutions as they are crowded, inefficient, and do not provide an overall pleasant experience that encourages people to continue using public transportation. Other alternatives to public transportation are services like Uber and Lyft;

however, these methods still contribute to the greenhouse gas emissions. In addition to this, parking is a hassle in any major city, this is especially true for San Jose and in particular SJSU. On the first day of classes this Fall 2019 semester, more than 22,600 cars entered the University's parking services (Faas, 2019). This meant that students and faculty who drove to campus, not only had to worry about their first day of class going well, they had to worry about being able to find parking. The theme with all of these issues is that there is not an environmentally friendly alternative at the current moment that is quick, efficient, and provides a pleasant experience. This is exactly where SPARTAN Superway fits into the picture. The positive societal impacts of the Superway will be discussed in the following subsection.

1.2 Societal Needs and Benefits

With the problems discussed in the prior subsection of the report, there is a dire need for a quick and green solution. As briefly mentioned in the first section of this report, SPARTAN Superway is a suspended railway system which aims to provide a quick, efficient, and pleasant transportation experience for the City of San Jose. This project as a whole is not like any of the present alternatives because it moves transportation above the ground. Furthermore, SPARTAN Superway will be fully powered via solar panels, so apart from the production and transportation of the solar panels, once fully operational, the SPARTAN Superway will not produce any carbon emissions. The goal is to have several stations at specific points of interest spread across San Jose. To reach each of these stations, there will be pods which can be requested from a mobile app. The pod nearest to the station will pick up the customer and take them to the station closest to their desired destination. With a project such as this one being implemented in the city of San Jose, the quality of life for the people will improve dramatically. Having a suspended

transportation network means that traffic can be reduced significantly. If an increased amount of people are electing to use SPARTAN Superway, the amount of people driving is reduced, hence reducing the amount of traffic and the emissions that would otherwise happen. Having pods which traverse to specific points of interest is beneficial because it creates a “carpool” experience in which up to four people are able to be in a pod at once. This maximizes the amount of people which can be on a pod.

In addition to this project having a positive impact in the city of San Jose, it can have a greater impact in the SJSU community. As stated in the current problems section, parking is a huge problem for any person who needs to commute to the university. Adding drop off stations for SPARTAN Superway near the university, would allow for students, professors and any SJSU affiliate to avoid the need to search for parking every morning. It could also create a tighter bond within the SJSU community as students, professors, or really anyone who decides to use this form of transportation would be able to interact with new people who they may have not met otherwise. Overall, the societal impact of SPARTAN Superway will be positive in terms of the reduced: emissions, traffic and parking issues. Meanwhile, it will create a close knit community in which people look forward to choosing SPARTAN Superway as their transportation of choice.

As mentioned briefly in the project description section, SPARTAN Superway as a whole has many requirements and goals. The Small Scale Team plays a very important role in this project because it is a way of piquing people’s interest as to what the project is about. This is done by providing a fully functional small scale model which depicts how the full scale would work once fully operational. This can gain the interest of both investors and younger students who are

interested in this type of technology. This will further be discussed in the following sections of the report.

1.3 Objectives

The main objective of the Small Scale Team is to improve upon previous guideway, bogie, and controls designs in order to achieve a fully integrated and working system. The guideway team will focus on designing and building a y-junction and track that it is easy to assemble and will allow the bogie to travel smoothly. Another objective for the guideway is to improve the interchangeability and manufacturing for the parts of the track. For the bogie, the main objective is to design a switching mechanism which will eliminate failure at the y-junction. The next objective for the bogie team is to develop a pod and bogie which is easy to assemble and modify if needed. In other words, the pod and the bogie should be able to be disassembled to allow for easier access to electronics. For the controls team, the main objective is to have two pods on the track at the same time, each following its own path without colliding with one another. Another objective for controls is to develop an iOS app which allows the user to control the bogies from their mobile device. This app will control every aspect of the bogie, but will also keep the user in mind to create a pleasant and aesthetically pleasing user interface. This concept of creating a great user experience is crucial for getting people excited about the project. The more excitement and awareness regarding the project, the higher the chance of success.

1.4 Project Structure

The Small Scale Team can be organized into three main sub teams: bogie, guideway, and controls. Each subteam represents an important aspect of the project, all of which play a large role in the teams' overall success. Because of this, members of the Small Scale Team were

divided into three subteams. Asmaa Darwish, Lissette Romero, and Julio de Pereda Banda are on the controls team. Their primary responsibilities are to develop a control system for the pods mainly the app and the Arduino code. Justin Ghieuw and Shane Fatehi are designing and building the track for the guideway. Whenever deemed necessary members from either the controls or guideway team will come together to work on the bogie to ensure that progress is also made in this area. A visual representation of this division of team members is shown in figure 5.

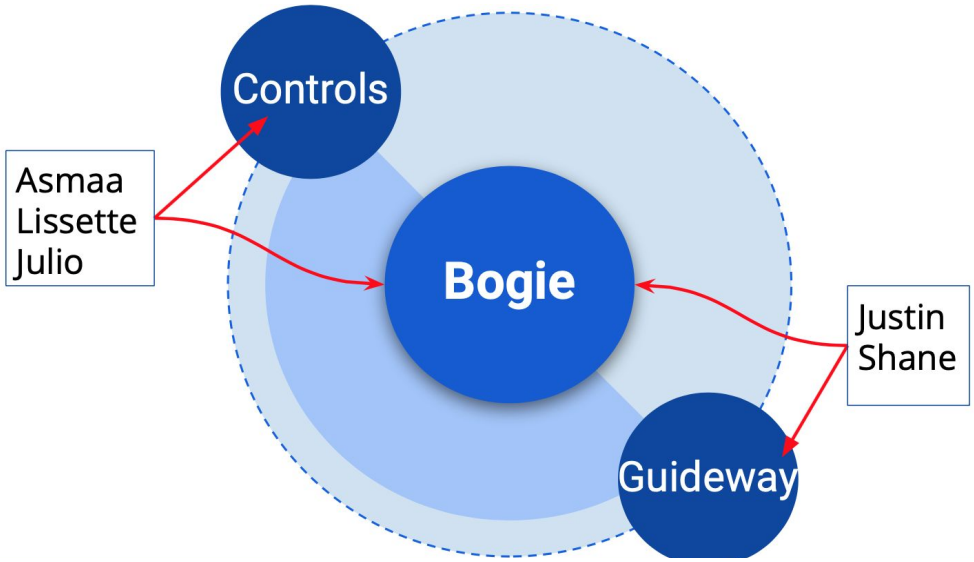


Figure 5. Division of Small Scale Team sub teams.

2. State-of-the-Art Review

2.1 ATN around the world

Several Automated Transit Networks exist in the world and each of these networks offer a unique feature that differentiates them from other networks. Automated Transit Networks offer a common feature where the steering and propulsion is part of the bogie and that the guideway has no moving parts (Furman, 2016). The guideway not having any moving parts is a step towards

reducing the long term maintenance costs. None of the Automated Transit Networks that are implemented today operate on a suspended guideway.

2.1.1 Wuppertal Suspension Railway

The Wuppertal Suspension Railway is not an ATN system; however, it is an example of one of very few guideways around the world which are elevated. It is located in Wuppertal, Germany and has been operational since 1901. It is comprised of one loop which covers a distance of 13 km and transports approximately 85,000 people. Since its first day of operation, there have been a total of 1.5 thousand million people who have used this form of transportation (Wuppertal Tourism and City Information , n.d.). The City of Wuppertal reports that people from all over the world visit the suspension railway on a yearly basis as it has become a popular point of interest. An image of this suspended railway can be seen in figures 6 and 7:

Figure 6. Wuppertal Suspended Railway above traffic.

Figure 7. The Wuppertal Suspension Railway above the city river.

This suspended railway is electrically powered and it can achieve a top speed of 37 miles per hour, with the whole journey from beginning to end taking a total of 35 minutes. Throughout the route across the city, there are a total of 20 stops at specific points of interest where people can either get on or off the railway (Wuppertal Tourism and City Information , n.d.). It was recently

reopened after being closed for around nine months due to a collapse of a part of its railway. The people who use the Wuppertal Suspension Railway states that it is a convenient form of transportation as the railway weaves through the city avoiding all of the traffic (Wuppertal Tourism and City Information , n.d.).

This elimination of traffic is a trend with these suspended railways as they provide substantial benefits by reducing ground real estate, which allows these guideways to be implemented anywhere, even above river as seen in figure 7 above.

2.1.2 Morgantown Personal Rapid Transit System

The Morgantown Personal Rapid Transit system is another example of ATN systems that resides in West Virginia and features a driverless system that transports students between campuses. The Morgantown Transit system is responsible for transporting 15,000 riders per day (Boeing, n.d.). Personal Rapid Transport (PRT) offers a smaller people mover, which can hold up to 4 or 6 passengers. The benefit of PRT is that it offers a more personal ride without making unnecessary stops, all the passengers will go to one location quickly. The Morgantown system pods have a similar steering system design as cars shown in Figure 8 (Morgantown, n.d.).

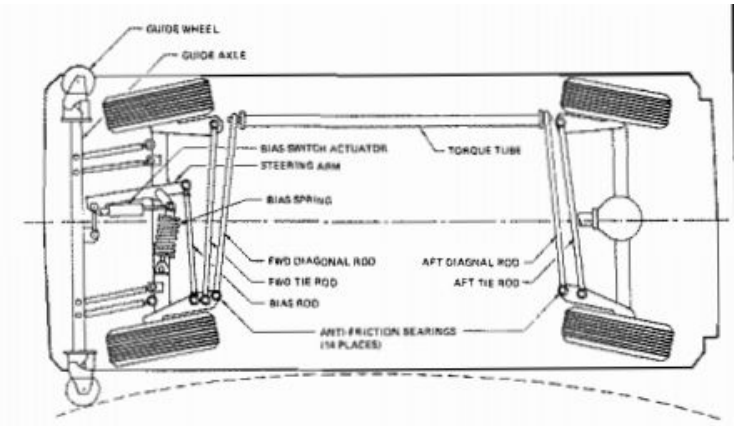


Figure 8. Morgantown vehicle steering system

2.1.3 ULTra Rapid Transit at London Heathrow Airport

Each one of the twenty-one driverless vehicles at London Heathrow Airport can carry up to four passengers. Each one of these autonomous vehicles can travel up to 25 miles per hour and transport passengers from terminal 5 to the car park in less than 5 minutes on its 2.1-meter wide guideway (Nordstrom, 2009). On the guideway, each pod travels in a single direction on rails as shown in Figure 9. The network is controlled and monitored by personnel that overlook the system inside a control center (Furman, Ellis, Mueller, & Swenson, 2014). Each vehicle is equipped with sensors to prevent collisions and a control system that allows each pod to take people to their destination without interruptions. It achieves this by taking the vehicles offline to avoid disruptions when the PRT is carrying passengers.

Figure 9. Urban Light Transit PRT in London Heathrow Airport

2.1.4 Cabinentaxi PRT in Hagen, Germany

The ATN PRT system to have a similar suspended guideway as SPARTAN Superway was developed by a joint-venture between two German companies and sponsored by a German

(Furman, Ellis, Mueller, & Swenson, 2014). This guideway system offers point-to-point PRT service to increase the time it takes to deliver passengers to each station. This guideway is unique because it uses two types of podcars. The first podcar is suspended on the guideway as shown in Figure 10. The other podcar can ride on top of the guideway rails like a car and is not suspended.

Figure 10. Cabinentaxi in Germany stopped developments in

This system went through all sorts of endurance testing to ensure it can withstand the real-world conditions for 11 months (Furman, Ellis, Mueller, & Swenson, 2014). The Cabinentaxi project development and funding ended in the 1980s.

2.2 ATN in SJSU

SPARTAN Superway is developing an ATN to be implemented at the SJSU campus and its surrounding areas. Each year since Superway's inception in 2012, senior project teams and summer interns have collaborated to demonstrate a proof of concept ATN system. The 2013-2014 team developed a full-scale 4.9 m guideway, improved on previous years 1/12 scale, and presented their work at Maker Faire and InterSolar (Cowley, et. al., 2014). In 2015, an

autonomous full-scale guideway with a working switch mechanism and a bogie fitted with motors (Ornellas, et. al., 2015). In addition, a new 1/12th scale was designed and manufactured that resembles the full-scale design more than the previous 1/12th scale models. A full-scale proof of concept cabin was made as shown in Figure 11.

Figure 11. SPARTAN Superway Full-Scale Cabin

Analysis was completed on the guideway in 2016 to determine the designs that need improvement and areas within the structure that may cause problems. The team featured over 40 students from different majors. Their most notable developments were improving the 1/12th scale controls. With the help of graduate software engineering students, the 1/12th scale was fitted with a mobile application user interface that allowed different users to create accounts, schedule rides, and save all pertinent information into a database (Alvarez, et. al., 2016). The interface of the app that was developed in 2016 is shown below in figure 12. The team developed a complex system that uses an asynchronous programming language (NodeJS) to spawn threads and communicate with the Firebase database to store each user's ticket. This year the controls team will be using Xcode, a much simpler programming language.

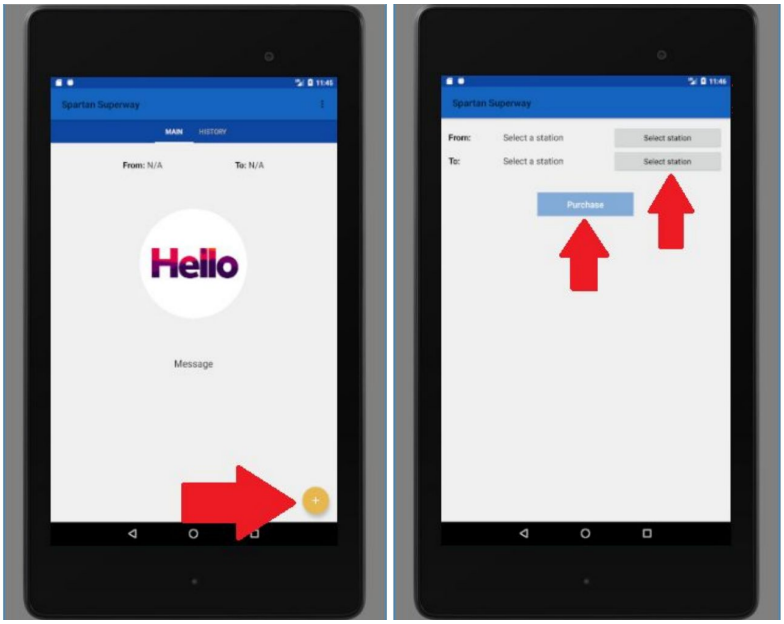


Figure 12. Small-Scale Mobile Application in 2016

In 2017, Superway made progress by upgrading the 1/12th scale track and podcar, improved propulsion and control system for the half-scale model, and small models of rails designs were made (Aquino, et. al., 2017). The small-scale team researched implementing a position sensing for each podcar that would be able to provide real-time location tracking. Due to the wheels of the bogie slipping and producing inaccurate position data, this method was not implemented. In 2018, the small-scale re-designed the bogie and implemented a master-slave bogie design, which prevents the bogie from getting stuck at round turns or at y-junctions (Halabi & Jocson, 2018). The 2018 team suggested making the bogie more robust by designing a dual-switch mechanism. The small scale SPARTAN Superway teams in the past have designed a guideway that featured 1/12th scale bogies to traverse along the guideway and effectively switch over rails on the y-junctions.

Over the past six years, the guideway has been made out of aluminum bar stock and the two rails separated by a fixed distance using 3D printed PLA clamps as shown in Figure 13 below. Previous teams were unsuccessful to get a working guideway due to several issues with the design. First, the aluminum bar stock needs to be bent to a specific radius for the corners of the guideway. The tool used to bend the aluminum was not precise, thus all the bent pieces of the guideway were different from each other. This caused the guideway pieces to not be interchangeable furthermore increasing the time it takes to take-apart and build the guideway. The 2018-2019 team spent the first half of the semester designing and manufacturing a tool that can precisely bend each aluminum bar stock for the corners of the guideway.

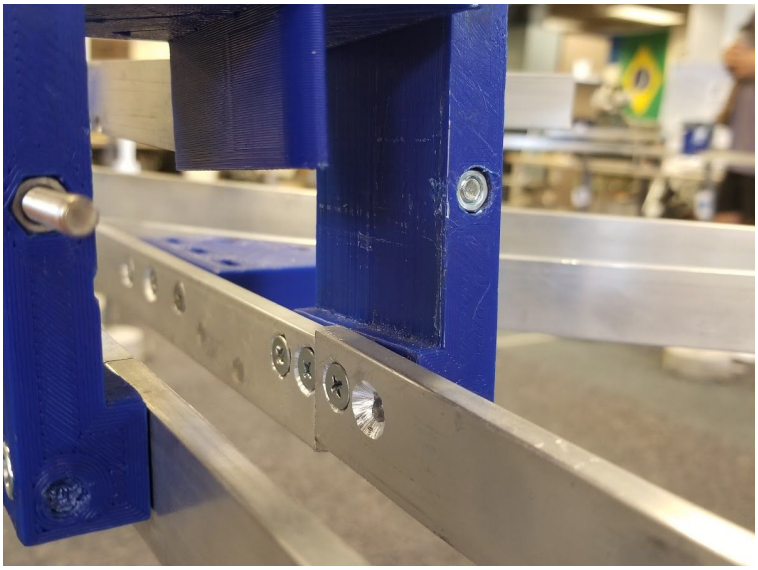


Figure 13. Connection between two tracks

The previous years bogie team developed a master and slave bogie assembly to ensure the system does not fall through the guideway when it switches. Previous teams stated that this implementation works better than a single bogie assembly. In addition, the bogie is equipped with inductive charging and tapered wheels. Using an inductive charger resulted in limitations

into the types of DC motors that can be used for propulsion. Tapered wheels were used to allow the bogie to easily traverse through the round corners. Non-tapered wheels resulted in the bogie getting stuck. The 2018-2019 controls team developed an application to be used with a Raspberry Pi tablet. The Raspberry Pi tablet was found to be inconvenient to use and determined that a phone application would be a preferable solution.

3. Design Solutions

The final design for this years' small scale team model will include a fully functional bogie, guideway, and controls system. The design will improve upon previous small scale work, while also meeting the goals and specifications which have been set by the team. This section will list and explain all design specifications and each subteams' designs. It will discuss the prime designs which have been developed, and explain the teams progress with supporting analysis.

3.1 Requirements and Specifications

3.1.1 Guideway Specification

Table 1. Guideway Design Specifications

Parameter	**Value**	**Unit**	**Comments**
Track (Plywood)	96 x 48	in.	Overall length and width of the SST
Base (Plywood)	96 x 48	in.	Overall length and width of the base
¼ Base	48x24	in	Length and width of the ¼ base part
Thickness	0.5	in.	Thickness for bottom and top tracks
Top Track	3	in.	Width of tracks (top & bottom)
Bottom Track	3.5	in.	
Straights	16	in	Length of top/bottom straights
Outer Curve Top Track Inner/Outer Radius	7.5 8.5	in.	Inner and outer radius of the top track for the outer curves, respectfully
Outer Curve Bottom Track Inner/Outer Radius	7 8.5	in.	Inner and outer radius of the bottom track for the outer curves, respectfully
Inner Curve Top Track Inner/Outer Radius	4 5	in.	Inner and outer radius of the top track for the inner curves, respectfully
Inner Curve Bottom Track Inner/Outer Radius	4 5.5	in.	Inner and outer radius of the top track for the outer curves, respectfully
Threaded Rods (Steel)	3	ft.	Length of threaded rods
	⅜-16	in.	Diameter of threaded rods

3.1.2 Bogie Specifications

Table 2. Bogie Design Specifications

Parameter	**Value**	**Unit**	**Comments**
Speed	1	m/s	The speed without stopping or slowing
Max Weight of Pod	3	lbs	Including mechanical and electrical hardware
Max Weight of Bogie	2	lbs	Including mechanical and electrical hardware
Max Weight of System	5	lbs	Including mechanical and electrical hardware
Pod Dimensions	3.5 x 9 x 6	in	
Bogie Dimensions	9 x 2 x 2	in	
Y switch	2	/bogie	2 fully functional y switches per bogie
Stepper Motor	1	/bogie	1 motor will be used per bogie

3.1.3 Controls Specifications

Table 3. Controls Design Specifications

Parameter	**Value**	**Unit**	**Comments**
Sensor Response Time			Maximum
iOS app	N/A	N/A	Fully Functional (Communicates w/ BLE)
Ultrasonic sensor	1	/pod	Each pod will have 1 ultrasonic sensor
PixyCam	1	/pod	Each pod will have 1 PixyCam
HM 10 (BLE)	1	/pod	Each pod will use an HM 10

3.2 Prime Design

3.2.1 Guideway Design

One of the major issues with the previous Superway guideways was manufacturing each piece of the track within specified tolerances to allow for interchangeability. It was determined that to ensure precision, each piece must be machined using a high precision tool. To improve the accuracy of the rail dimensions the guideway will be made up of laser cut pieces of plywood. There will be three major parts of the guideway: the track, stands, and base. The base will be the foundation upon which the stands and track will lie on. The dimensions of the base will be 96 inches by 48 inches, or 8 feet by 4 feet. To ensure portability, the base will be divided into four pieces each being 48 inches by 24 inches. The thickness of the base will be 1 inch to allow the threaded rod to be inserted. This will also allow the base to be easily disassembled and convenient to carry. The idea for using threaded studs was inspired by Bill James' first prototype of the mini-scale track as shown in Figure 14.

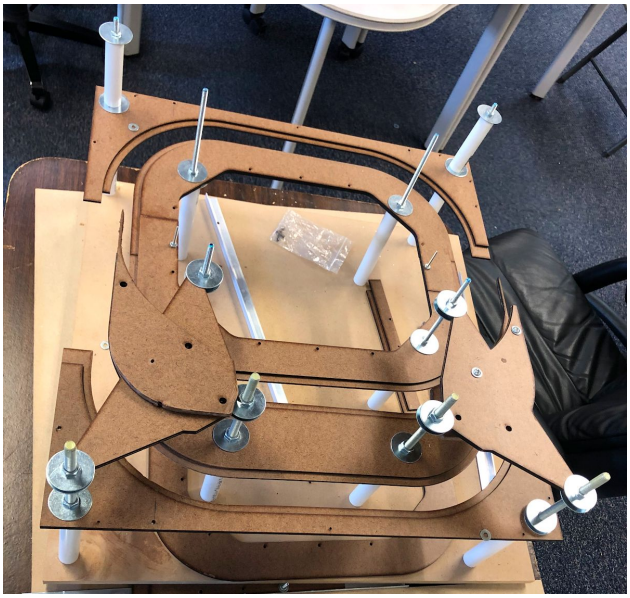


Figure 14. Bill James first prototype of mini-scale model using threaded studs.

The guideway team plans to purchase ¼-20 threaded studs that are three feet in length that will be purchased from McMaster-Carr. The threaded studs will be clamped onto the base board and track pieces by using a steel flat washer and a steel flanged nut as shown in Figure 15. A hole with a diameter of 0.26 inches will be drilled in the base of the board in addition to the track pieces for the threaded stud to be inserted through. The underside of the base will have counterbored holes, which will allow the track to sit flat on the floor without the nut protruding from the other side. The track will consist of sections that are straight and curved. For each part of the track, there will be a layer cake design implemented to increase rigidity and prevent too much deflection. Each layer of the track will have a thickness of half an inch and the width will be different for the lower and upper level. The lower layer track pieces will have a width of 1.5 inches and the upper layer will have a width of 1 inch. The layering of the track will allow the bogie to be placed nicely on the guideway, with enough space to hold wheels with a width of up to half an inch.

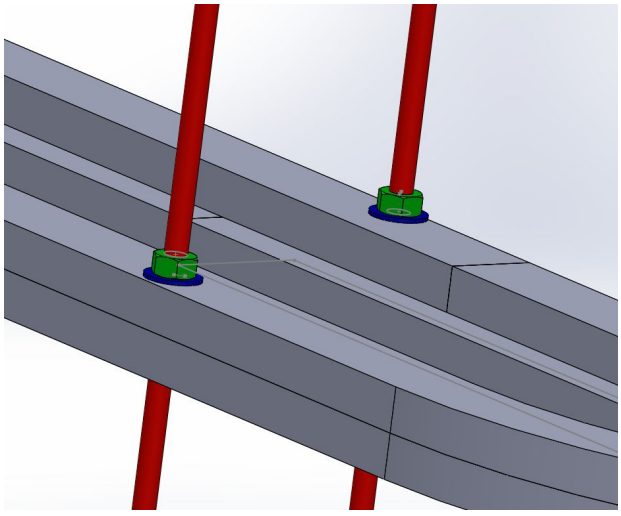


Figure 15. Flat washer and nut clamping on top and bottom clamping track pieces.

An idea of how the guideway will look using threaded studs to suspend the track pieces can be seen in Figure 16. The track will be elevated between two to three feet using the hardware described above. The guideway team plans to incorporate pvc tube in addition with the threaded stud to increase rigidity and minimize the track from wobbling.

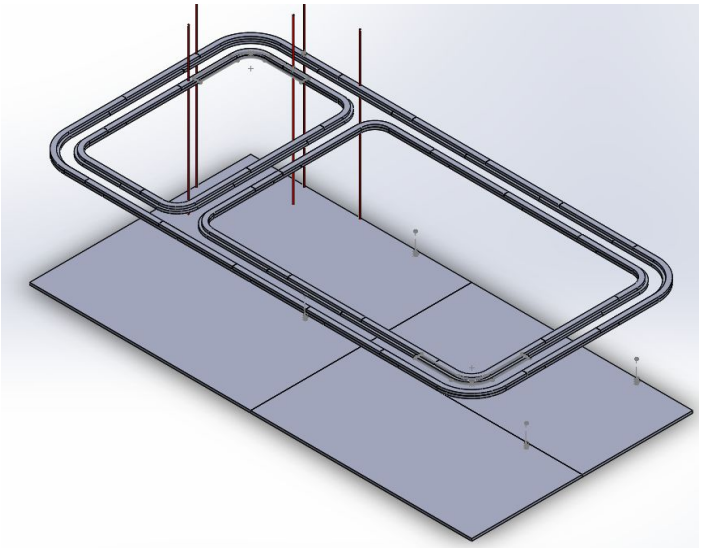


Figure 16. Guideway visual using threaded studs to suspend.

3.2.2 Bogie Design

There are 3 main issues with previous bogie design which will be addressed in this year's prime design of the bogie. One of the biggest issues that the previous teams have faced is high failure rate at the y-junctions. This failure was mostly due to the design of the switching mechanism that was implemented in the bogie. In all prior designs the switching mechanism relied solely on the pressure that the switch exerted on the y-junction to complete the turn, so in many cases the bogie would fall while making the switch. To solve this problem the new bogie design will implement a more realistic switching mechanism, which will support the weight of the pod while traversing across the y-junction. Instead of a tab like switch that flips from one side to the other,

the new design will have two arms, which when switched will grapple onto a third rail. This way, as the bogie travels across the y-junction, the pod will be supported. In order for the switch to hold onto the rail, the rail will also have to be redesigned accordingly. Figure 17 below illustrates the differences between the old design (on the left) and the new design (on the right).

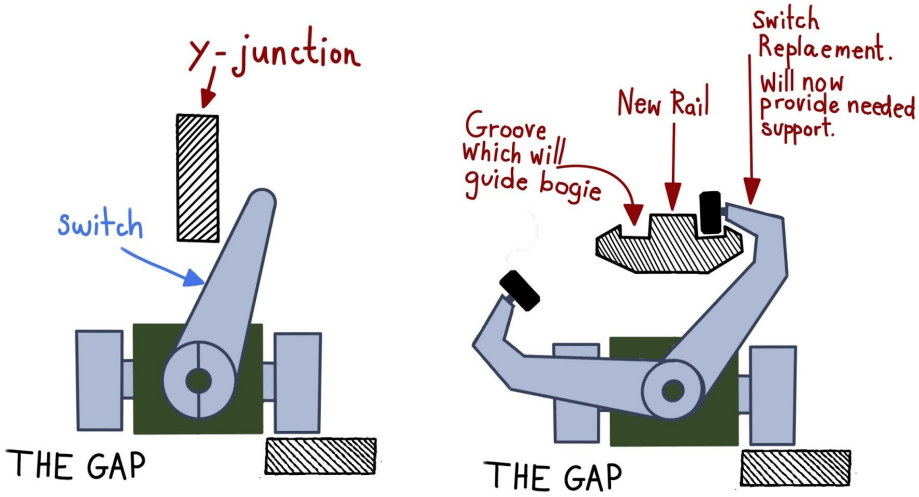


Figure 17. The old switch design and the new switch design

The second issue that will be discussed is the lack of a maintenance friendly design. In previous years the bogie and the pod were made as two solid pieces rigidly attached together by screws. Because of this, the only way to get the bogie and the pod in and out of the guideway was to either disassemble the track or the pod. Ultimately this made it difficult to continuously test and make changes to the code. To solve this issue, and improve the ability to make quick changes to the bogie and control designs, a snap fit was designed as shown in figure 18.

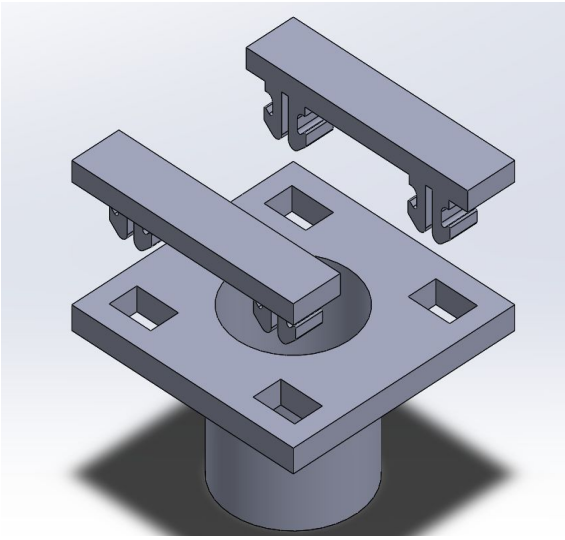


Figure 18. 3D model of Snap fit design

The purpose of the snap fit is to allow for quick connection and disconnection between the bogie to the pod car which eliminates the need to disassemble any parts. Building upon this idea, the connections between the pod and the bogie are designed with as a hollow tube, which is where the wires go through to connect to the bogie's electrical components. This will improve the overall aesthetics of the final product. To further improve the workability of the design, the body of the pod car was broken up into multiple parts, which, with a series of keys and press fits, will be able to easily assemble and disassemble when needed. Another advantage to dividing the car into smaller parts is that it allows for a shorter printing time. As a result, if any changes need to be made to a part of the pod design, only one of the parts will have to be reprinted, and less time and material will be wasted. The prime pod car design consists of a total of five parts as shown in figure 19.

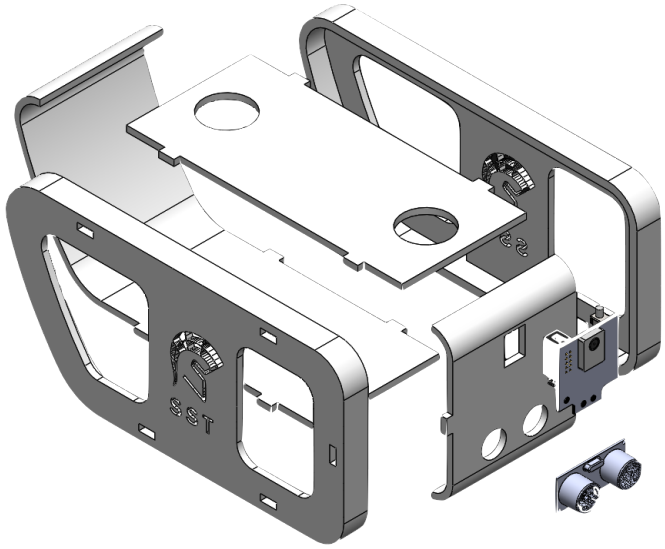


Figure 19. Exploded View of Pod Car Design

Finally, with inspiration from Bill James' design, it was decided that the bogie should be allowed to swivel freely. This will allow for the bogie to effectively clear the turns and the y-junction with ease. Figure 20 shows the concept of the swivel design which will be implemented.

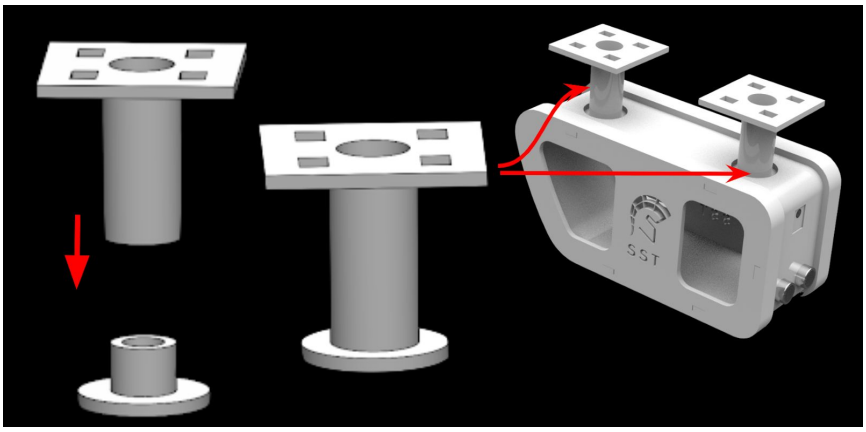


Figure 20. Assembly of the swivel design

Each pod will be equipped with two swiveling connections. The connections are made of two pieces which will be assembled in the openings at the top of the pod. The pieces were designed such that a gap will remain between the bore in the pod and the outer diameter of the connection. This will allow the connection to swivel as the pod travels around the track. At the top of the

swivel connection the snap fit design was implemented to be able to connect to the bogie platform. A full 3D CAD of the prime design of the bogie and pod car was developed. A render view was created to show the final design of the pod integrated with a rough sketch of the bogie shown in figure 21.

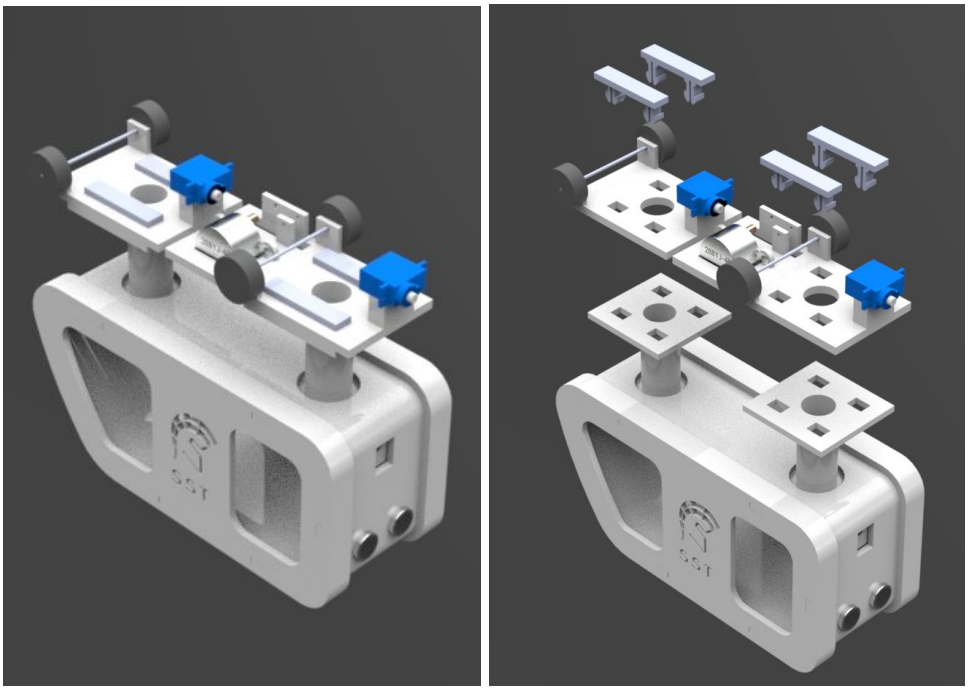


Figure 21. Render of isometric view of pod and bogie assembly

3.2.3 Controls Design

The Small Scale control design consists of two parts, an iOS app and an Arduino code.

3.2.3.a iOS App

For the iOS app, there is an ideal design in mind which serves two distinct purposes simultaneously. The first is to be able to fully control the bogies and pods via an HM10 BLE module. As stated in the design specification section, the goal is to have two fully functional anti-colliding bogies which traverse the track successfully. The first goal which the iOS app is meant to accomplish is to be able to control both of the bogies on the track from any team

member who has the iOS app. The ideal concept of the app is that as soon as either of the two pods are requested from the app, the pods will traverse to the designated pick up location. The user will be able to specify which station the pod needs to travel to on the small scale guideway. This simulates what would similarly happen if this was to be implemented into the full scale model. The app aims to serve almost as a remote control for Arduinos.

In addition to the iOS app being used to control the pods and bogies, the app is meant to provide a pleasant user interface in both a functional and aesthetic manner. This will be achieved by having anywhere from three to four tabs, with different information within each tab. The information within two of these tabs is for certain, and the remaining two will be decided over winter break. Prior to reaching any of these tabs, the users will be met with fairly quick splash (loading) screen which can be seen below in figure 22.

Figure 22. App loading screen.

The first tab which has been decided to be a part of the prime design is a welcome tab in which users can find information regarding what SPARTAN Superway is about, as well as a brief introduction to the Small Scale Team. In this brief intro of the team, a picture of each team member, along with short "biographies" will be provided so that any user of the app can learn more about both the team and the project. A rough draft of what the final design of this tab can be seen below in figure 23:

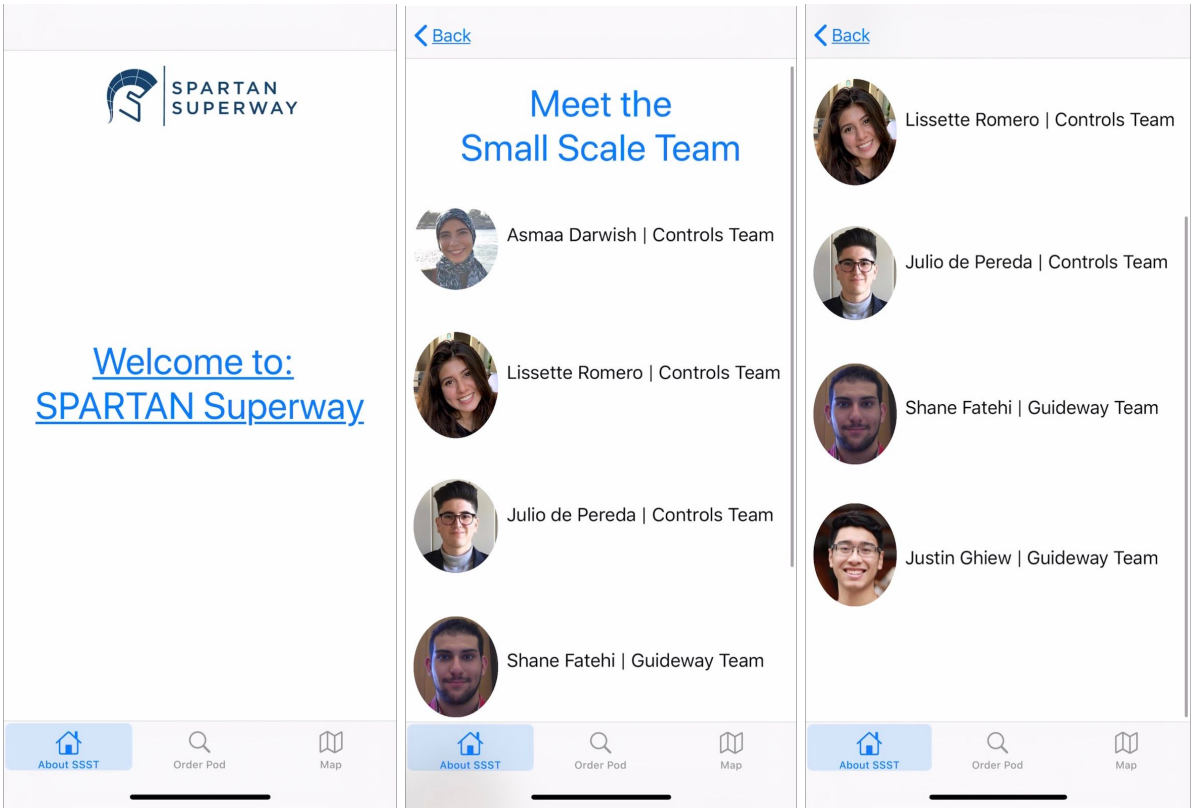


Figure 23. Main tab with project and team information.

The second tab will be where the most effort is required in order for it to work successfully. This is the tab in which the iOS app will be able to connect to the Arduinos in each pod via the bluetooth module. Ideally, in this tab, both pods can be called, directed, and controlled. A rough draft of what the interface for the final design would look like can be seen in figure 24:

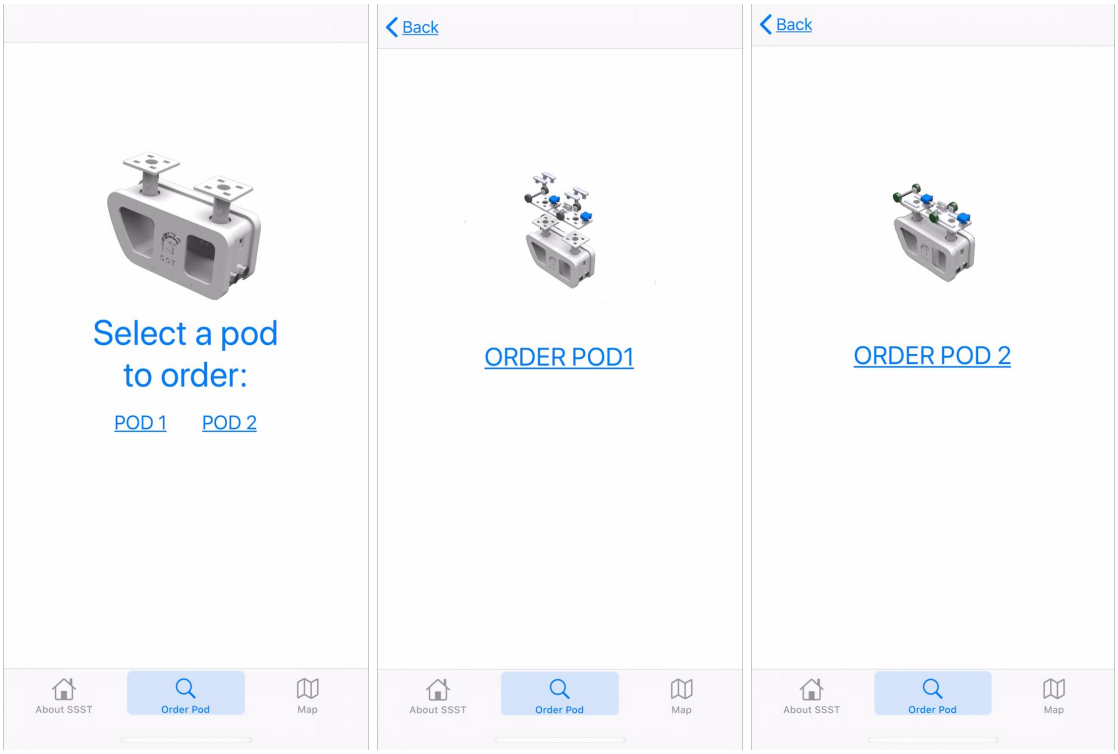


Figure 24. Rough design UI design of the app

As it will be discussed in greater detail in the analysis section, this tab is going to be where the most actual coding goes into. Once the user clicks on either the "POD 1" or "POD 2" buttons, the respective "tab within a tab" will be presented to the user. Based on their pod of choice, the user will either "ORDER POD 1" or "ORDER POD 2". Behind the scenes of this tab through the coding, this will prompt the respective Arduino code to run and the pod will begin to go through it's journey. This part of the app is more complex because in order for the iOS app to connect to the Arduino, a lot of requisites between the iPhone and the BLE module have to be met. The images shown in Figure 24 are just current test tabs, the ideal tab will have more buttons to give different directions to the pods. The last two or three tabs are still being thought of and developed. There are several ideas being bounced around as to what is the ideal information to be depicted in these tabs. Some of these ideas are a map locating where SPARTAN Superway is

located, or maybe a live feed into the Pixie cam on the pod. These tabs are still being thought of and developed. At the current moment, the third tab of the latest version of the iOS app can be seen in the following figure 25.

Figure 25. Current third tab UI

The prime app design will also feature different animations and transitions when the loading screen of the app appears, in between tabs, whenever buttons are clicked, and others. This will provide a great user interface experience which is vital when presenting these kinds of ideas to investors or younger students. A product that not only functions well, but looks and feels good, is something which will gain more interest from the public.

An app logo was designed to be uniquely identifiable as the SPARTAN Superway app. The app icon can be seen in figure 26.

Figure 26. iOS app logo.

The team is happy with the overall design of the logo for the app icon, so it seems that this will be the prime design icon for future iterations of the app. A great feature of using Xcode to build the iOS app is that there is infinite freedom as to how anything looks and functions. Thus, if any changes need to be made to meet a specific look or feel, it can be done very easily. As mentioned earlier, a big key of the second tab is getting the iOS app to connect to the Arduino, the process for this connection will be briefly touched upon in the following section regarding the Arduino code. The concept behind BLE and the relation to the app will also be covered in the iOS app analysis section.

3.2.3.b Arduino Code

For the Arduino code, one of the objectives of the controls team is to allow two pods to move on the track without colliding with one another. The team was able to successfully create an anti-collision system using ultrasonic sensors. A code shown in appendix F was designed to slow down the motor to three different speeds depending on the distance calculated from the ultrasonic sensor. At a distance greater than 25 cm, the motor will run at full speed, which is set to be 160 RPM. When the distance is between 25 cm and 15 cm, the motor speed is lowered to

35 RPM, and when the distance is between 15 cm and 3 cm, the speed is lowered to 15 RPM. Any distance that is equal to or less than 3 cm, the motor will stop completely. Figure 27 below shows the connection between the Arduino mega, the ultrasonic sensor, and the motor.

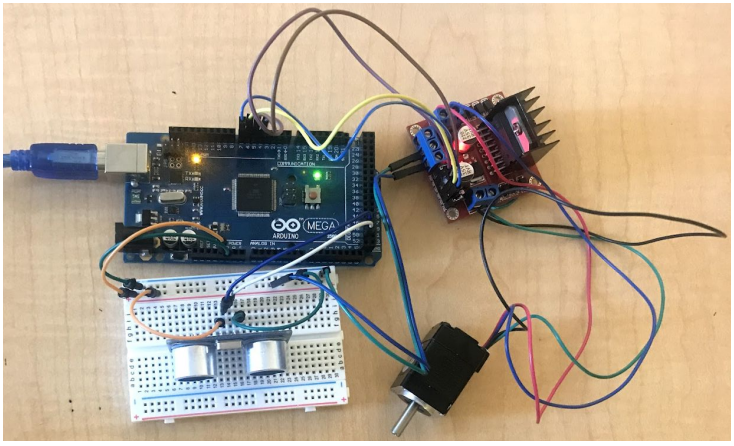


Figure 27. Anti-collision system electrical configuration

To be able to communicate with the iOS app, the team is using HM-10, a Bluetooth Low Energy (BLE) module. To identify the module in Arduino and the iOS app codes, the BLE had to be configured first. Using an existing Arduino example "SoftwareSerial" shown in appendix G, the module can accept AT commands that can change some configurations like the renaming of the module. The first HM-10 was renamed BLE1, which would make it easier to find later on to connect to the iOS app. A second HM-10 will be renamed to BLE2 so that it can be easily distinguished from BLE1. Each pod will consist of one BLE and one Arduino.

Integrating the BLE into the anti-collision Arduino code created the new code which can be found in appendix I. This code reads an input signal from the BLE and runs different functions depending on the value received. While the SPARTAN Superway iOS app is being prepared to connect to the BLE, a free iOS app was used that connects to the BLE and sends signals through a text field. More explanation of how the BLE connection was made can be found in the analysis

section. A flowchart of the code is shown in figure 28, which illustrates the actions taken by the app to use the BLE in conjunction with anti-collision system. The motor is able to run and read the ultrasonic signal simultaneously when an input of 1 is received from the app.

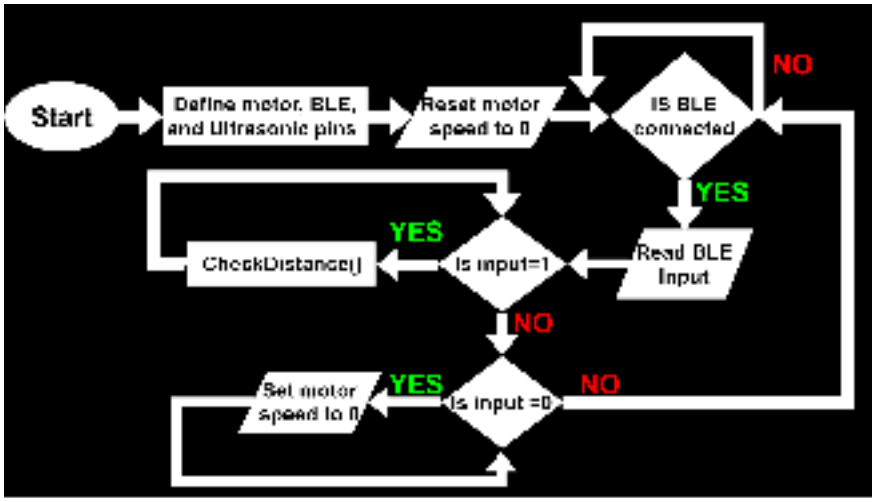


Figure 28. Flowchart of the implementation of the BLE and the anti-collision

While testing the implementation of the BLE and the anti-collision code in Arduino electrical components were configured as shown in figure 29 below.

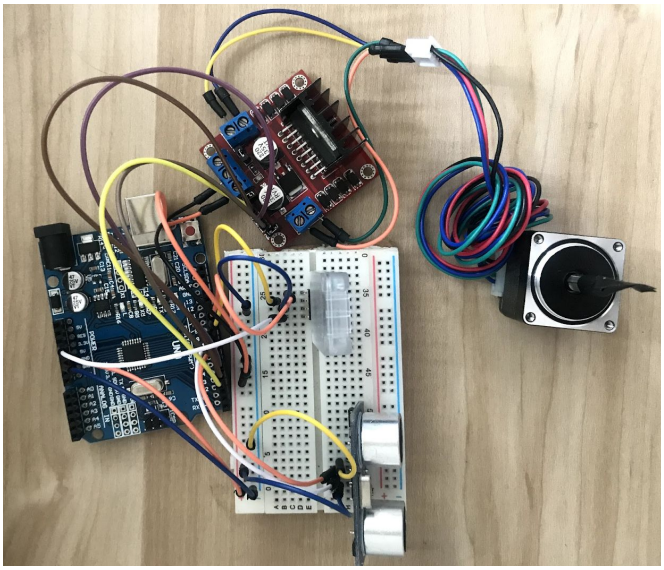


Figure 29. Electrical hardware configuration for testing the anti-collision code

Although the controls team has yet to test the other sensors which will be implemented in the final product, a fritzing diagram was made with the desired final electrical configuration. The diagram is shown in figure 30. All the electrical components have been assigned to a specific and appropriate pin number on the ArduinoMega. From this diagram an electrical analysis can be made on this design, flowcharts can be developed, and a rough arduino script can be made.

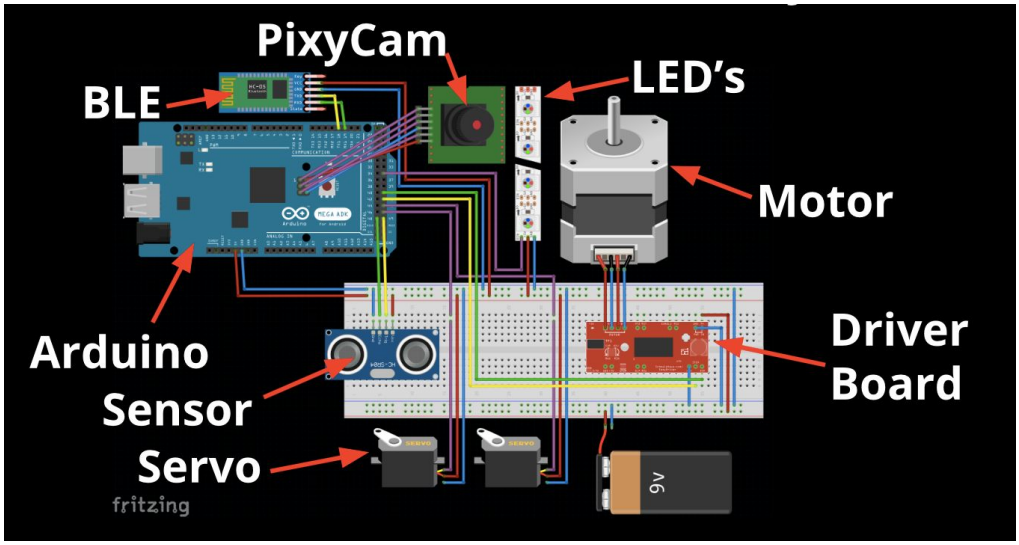


Figure 30. Fritzing diagram of desired electrical configuration.

3.3 Supporting Analysis

3.3.1 Guideway Analysis

The straight track piece underwent FEA analysis using ANSYS to determine the maximum deflection. The track was made using plywood, which has material properties as listed in Appendix K. A combined bogie and pod system weight of five pounds for the simulation was used. For the analysis, the length of the beam was designed to be 16 inches, and an assumption was made that the beam would be fixed at both ends. Using a bogie and pod weight of five pounds means that 2.5 pounds distributed between two wheels for each side of the bogie. Each

rail for the analysis has an applied external force of 2.5 pounds. Applying symmetry, the length of the track was cut down to 8 inches, with one side having a fixed support. On the other side, a force of 2.5 pounds was applied and an automatic generated mesh was used. Results from the analysis showed that the max deflection would be 0.0017 inches max. The von-Mises stress was also found to be 67.03 pounds per square inch max.

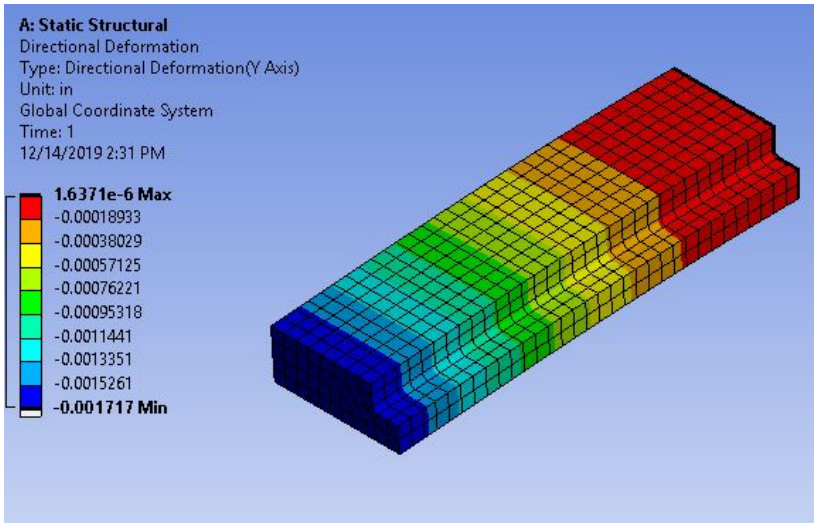


Figure 31. FEA analysis of a straight track in ANSYS, using symmetry.

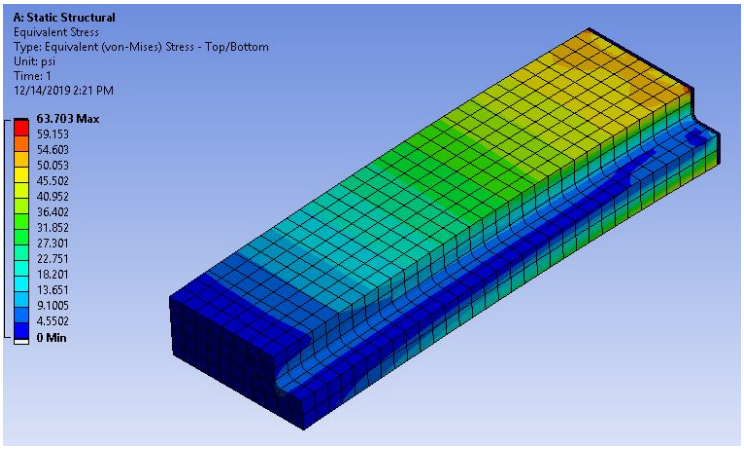


Figure 32. Equivalent stress found using FEA.

3-D, structural beam analysis was primarily used to analyze the track. Similarly, if the track is treated as a simple beam, 2-D analysis can also be applied. The main goal of this analysis is to

find maximum deflection and bending moment along the track. If the model is approached such that it follows Figure 33, then an element stiffness matrix can be created.

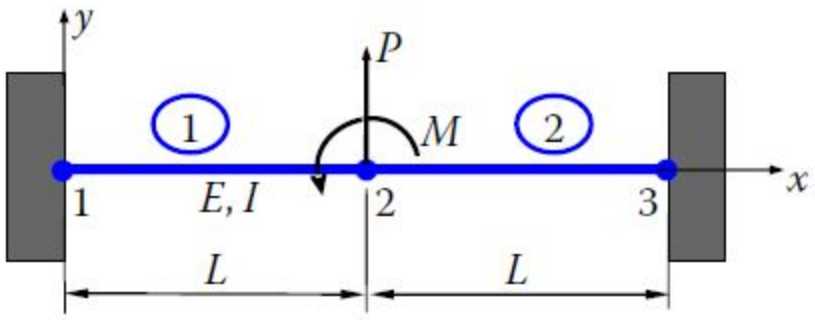


Figure 33. Simple fixed beam example.

Here, a global stiffness matrix can be formed from the beam example, similar to the tracks (Chen, Liu).

$$\frac{EI}{L^3}\begin{bmatrix} 12 & 6L & -12 & 6L & 0 & 0 \\ 6L & 4L^2 & -6L & 2L^2 & 0 & 0 \\ -12 & -6L & 24 & 0 & -12 & 6L \\ 6L & 2L^2 & 0 & 8L^2 & -6L & 2L^2 \\ 0 & 0 & -12 & -6L & 12 & -6L \\ 0 & 0 & 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \begin{matrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \\ v_3 \\ \theta_3 \end{matrix} = \begin{matrix} F_{1Y} \\ M_1 \\ F_{2Y} \\ M_2 \\ F_{3Y} \\ M_3 \end{matrix} \quad (1)$$

where,

$E =$ *Young's Modulus of Elasticity* $I =$ *Moment of Inertia of the cross − sectional area*

$L =$ *Length of the beam* $v =$ *Deflection* $\theta =$ *Rotation of the beam*

$M =$ *Internal bending moment* $F_i, M_i, F_j, M_j =$ *External lateral forces and moments at nodes i and j*

Loads and constraints can also be applied later, in this case:

$$F_{2Y} = -P,\ M_2 = M,\ v_1 = v_3 = \theta_1 = \theta_3 = 0$$

Applying boundary conditions to the global stiffness matrix, the finite element equation reduces to:

$$\frac{EI}{L^3}\begin{bmatrix} 24 & 0 \\ 0 & 8L^2 \end{bmatrix}\begin{Bmatrix} v_2 \\ \theta_2 \end{Bmatrix} = \begin{Bmatrix} -P \\ M \end{Bmatrix} \quad (2)$$

Solving for the equation, an equation for deflection and rotation can be obtained.

$$\begin{Bmatrix} v_2 \\ \theta_2 \end{Bmatrix} = \frac{L}{24EI}\begin{Bmatrix} -PL^2 \\ 3M \end{Bmatrix} \quad (3)$$

Plugging in our known values for the track, the max deflection that should occur along the beam would be 0.0018 inches, which is almost the exact same as our model in ANSYS has given. Although 3-D and 2-D analysis has been performed on part of the straight track, other parts of the track will be much different. Because of the complex shapes of the curved track, the y-junction, and the overall track, analysis will primarily be done in ANSYS in the future. A thorough analysis of the beam connections should also be considered.

3.3.2 Bogie Analysis

Upon analyzing the 3D model of the bogie and the pod, several conclusions were made about the current design. Overall, there are many strong points throughout the design that inspire confidence about the success of the overall design. One part of the design that has been tested and proven to be functional is the snap fit design. The reliability and robustness of the snap design was tested by 3D printing the design. The printed snap fit worked well, the strength of the snap fit exceeded the expectations of the team. From this it was concluded that the implementation of the snap fit into the bogie design will be successful. From the 3D printed pieces it was observed that resolution of the print was not as fine as anticipated, thus increasing

the interference of the snap fit components. With this interference in mind, the press fits in the pod were designed, however, because this connection relies solely on the interference of the two mating objects, the exact reliability of these fits are still unknown. The press fits on the swivel connections, shown in figure 34, is of particular concern.

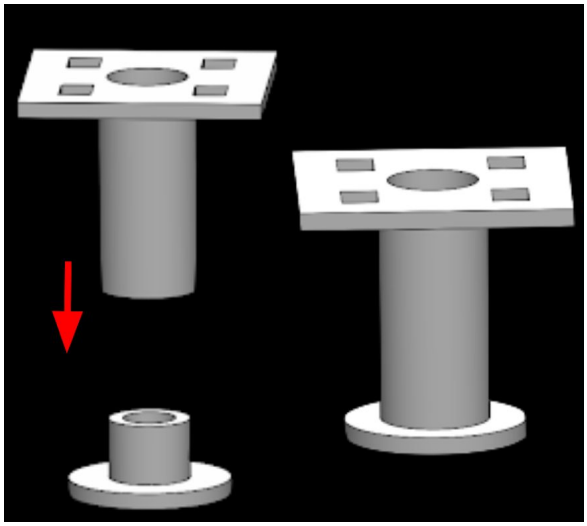


Figure 34. Swivel connection press fit.

This concern stems from the fact that the entire weight of the pod will be resting on the bottom connection of the press fit. Because of this the press fit connection must be stronger than the force due to the weight of the bogie. To test the strength, a sample of the swivel connection will be printed. If the results conclude that the fit is not strong enough, then a pin will be placed through the press fit, parallel to the bogie platform. Adding this pin will ensure a secure and safe connection.

The next point of concern is the clearance between the two bogie platforms. The two pieces that make up the the bogie are shown in figure 35. As the bogie traverses across the guideway the platforms will rotate independently from each other and, due to the small clearance between the

two platforms, they will collide. To fix the bogie team has already begun to redesign the bogie such that there is a larger clearance between the platforms.

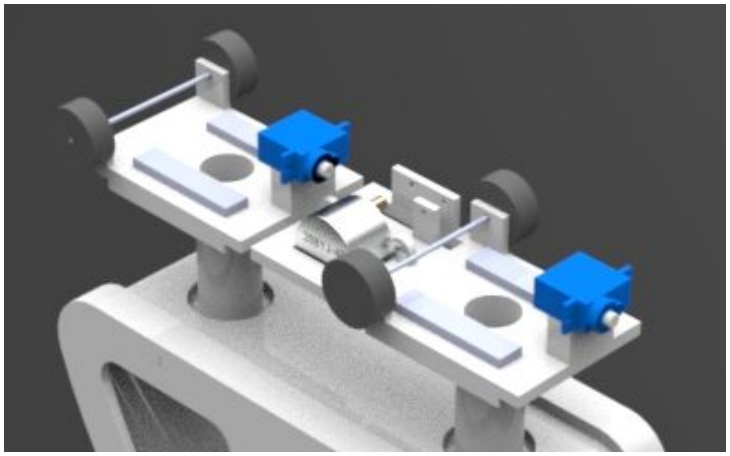


Figure 35. The current bogie platforms

For the pod, the design is final and 3D printing will begin next semester. On the other hand, for the bogie, some more designing will have to be made before reaching a satisfactory design.

3.3.3 Controls Analysis

3.2.3.a iOS App Analysis

Designing an app from scratch presents many great challenges and opportunities. As stated earlier in the iOS app prime design section, the software used to develop the different iterations of the SPARTAN Superway app is Xcode. The programming language which Xcode utilizes is called Swift. Apple claims that Swift was designed to be anyone's first language and that it can easily be learned. A lot of the language is based on objective C and C++. Having experience with either of these languages allows the transition to using Swift much easier. However, like any language, there is a learning curve and it takes a lot of practice to become proficient. The versatile thing about Xcode is that it allows for an app to be developed using two different techniques, either SwiftUI which is mostly coding, or Storyboard, which is more of a visual

approach. SwiftUI uses essentially every aspect of any other programming language, from integers, string character, functions, cases, and others. This provides an infinitely customizable experience in which every facet of a desired app can be manipulated meticulously. An example of what SwiftUI looks like from the coding, to the actual user interface on a device, can be seen in figure 36:

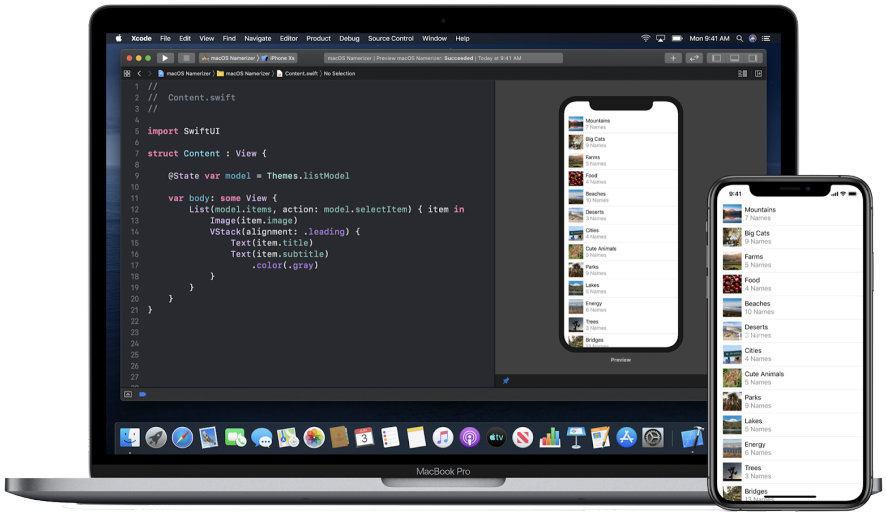


Figure 36. SwiftUI on Xcode and output on desired device. (SwiftUI, n.d.)

The coding in Swift can be seen on the computer screen, along with a simulation of what the end result will look like on an actual device.

The other mentioned approach to building an app in Xcode is to use the Storyboard mode. Using the Storyboard mode, it allows for visual elements and controls to be added to a controller view, which is what the user will ultimately be seeing when using the app. In this mode, a very basic app can be built without writing a single line of code. All that has to be done is add labels, images, buttons, scrolling features, and other cosmetic features to the desired app views. In the developing of the SPARTAN Superway iOS app, both approaches were utilized for two separate

iterations. When first creating an app in Xcode, a prompt is presented which can be seen below in figure 37:

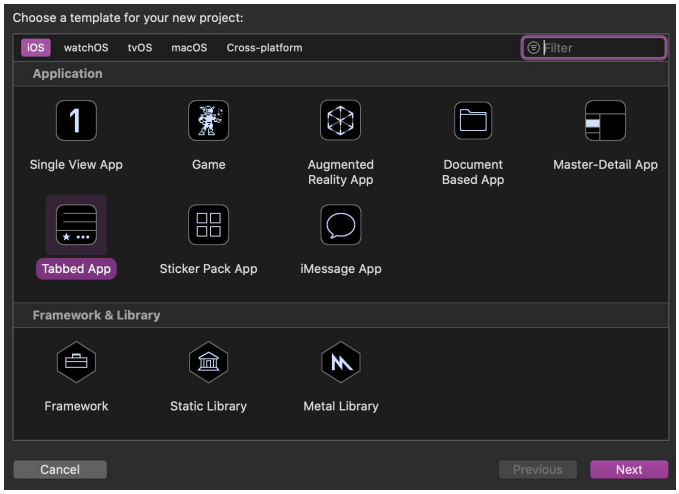


Figure 37. Template options for a new app in Xcode.

In this prompt, several templates are given with regards to the type of app which the user wants to be structured as, this ranges from a single view app, to a game, and to a tabbed app. The latter option was selected for the first iteration of the app. The next prompt that is presented when setting up a new app project is the project settings. This options page can be seen below in figure 38:

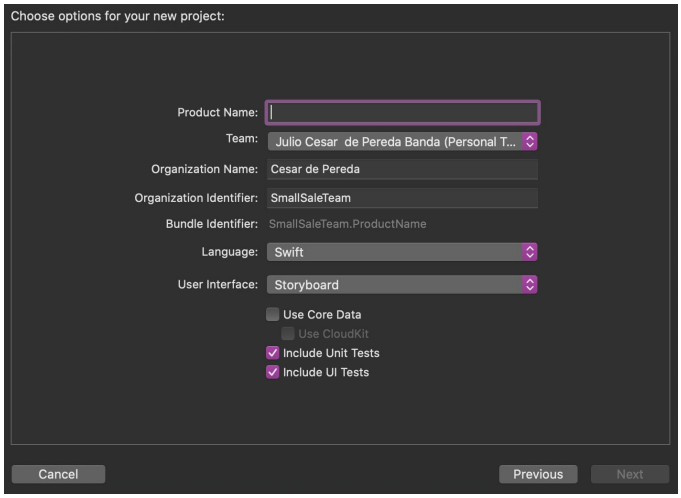


Figure 38. Options page in Xcode

In this page, the name of the app, the team that is working on the app, the language (Swift or Objective C can be chosen), and the desired user interface can be chosen. What is of interest in this options page is choosing whether to use the Storyboard or SwiftUI. This will take the app in two different directions as discussed earlier. The first iteration of the app utilized the SwiftUI approach to build a very simple three tabbed app. Choosing the SwiftUI option for the tabbed app template, the code is already pre-set in the files of the new project, which structure the set up of the three tabs. Doing some research, allows for the code to set up these tabs to be easily modified to add images, text, and specific layout options. A snippet of what this edited code looks like for the first iteration of the app can be seen below in figure 39:

```
import SwiftUI

struct ContentView: View {

    init(){
        UITabBar.appearance().backgroundColor = UIColor.white
    }

    @State private var selection = 0

    var body: some View {
        TabView(selection: $selection){
            Text("Welcome to: \r SPARTAN Superway")
                .font(.title)
                .foregroundColor(Color.blue)
                .multilineTextAlignment(.center)
                .tabItem{
                    VStack {
                        Image("first")
                        .padding(.top)
                        Text("Home")
                         .multilineTextAlignment(.center)
                    }
                }
```

Figure 39. Snippet of the edited code for the first iteration

The rest of the code for the layout of the first iteration of the app can be seen in Appendix N. Xcode allows for this code to be ran and simulated on their simulator software called "Simulator". Running the code prompts an option as to which Apple device to simulate the app on, this ranges from iPhones to iPads, etc. The other option is to connect a personal device and

upload the app to that specific device to test the app. For the first iteration of the app, the Simulator was used with an iPhone 11. The choice of device to simulate the app on does not matter too much, the app should look the same on any device when formatted right. After running the code shown in the snippet in figure 39, the app layout corresponding to this code can be seen below in figure 40:

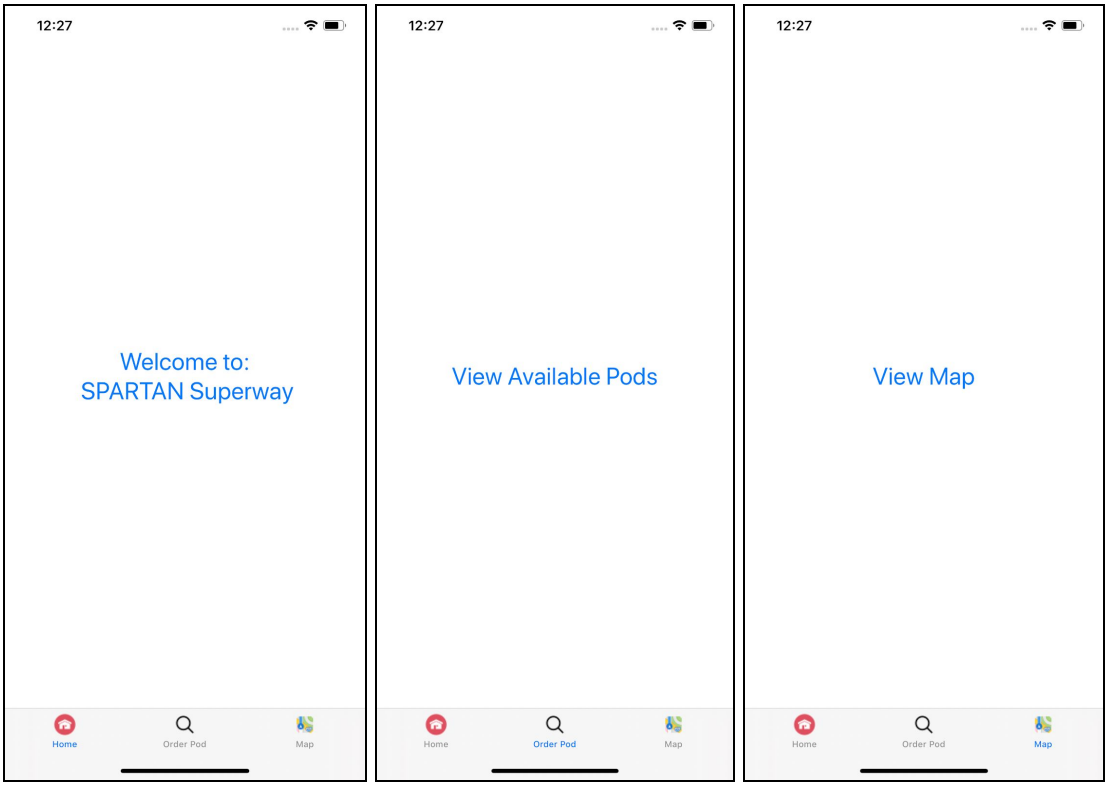


Figure 40. SPARTAN Superway app layout of first app iteration.

This was a very basic first iteration and was just a test to see how the app building process would be.

The second iteration of the app utilized the Storyboard mode. In this mode, more features were added without needing to write any code. In this mode, different controllers for different views of the app can be modified visually. Images, labels, buttons and a plethora of other options can

be dragged and dropped into the desired controllers. As more view controllers are added, they can be mapped out in a specific order. An image of what the Storyboard layout looks like for the second iteration of the app can be seen below in figure 41:

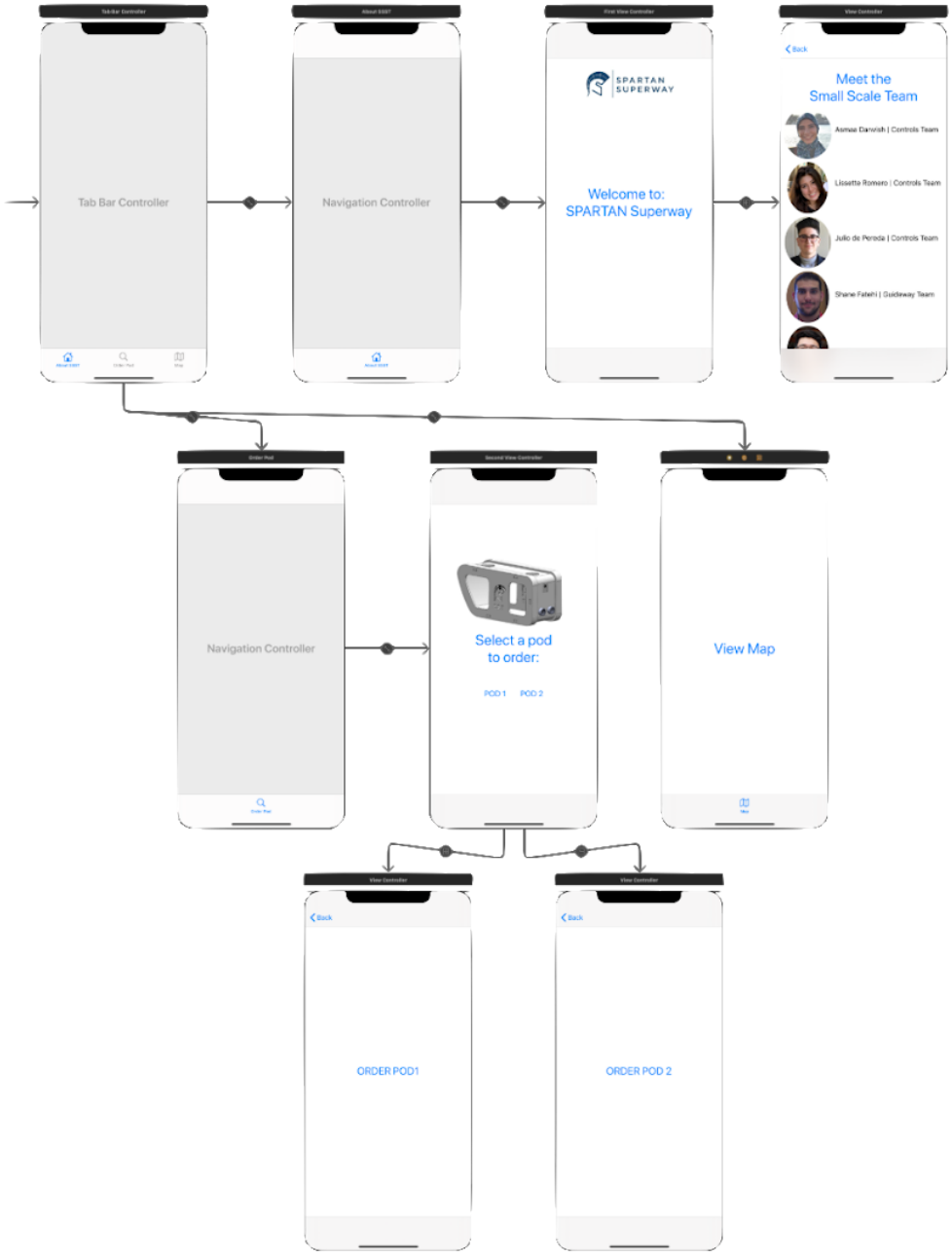


Figure 41. Storyboard layout of iOS app.

Looking at figure x for the first time without any prior knowledge as to how the layout works and what order the view controllers are in, can be a bit tricky. However, it is not quite as complex to explain. When running the app on a physical iPhone, the nine view controllers (screens with info), are just three seperate tabs: “About SSST”, “Orde Pod”, “View map”. A specific tab can be selected to be the first tab the users sees, in this case, the “Home” tab was selected as such. Hence, after the app is opened, the first tab which the user sees after the loading screen is the home tab as shown in figure 42:

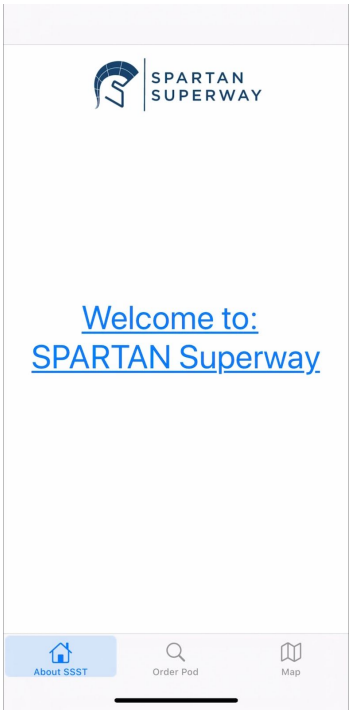


Figure 42. First tab the user sees after app has loaded.

At this point, the user can either click on the “Welcome to: SPARTAN Superway” button, or they can click either of the two other tab buttons located at the bottom of the screen. If the user clicks the “Welcome to: SPARTAN Superway” button, they will be directed to the following screen shown in figure 43:

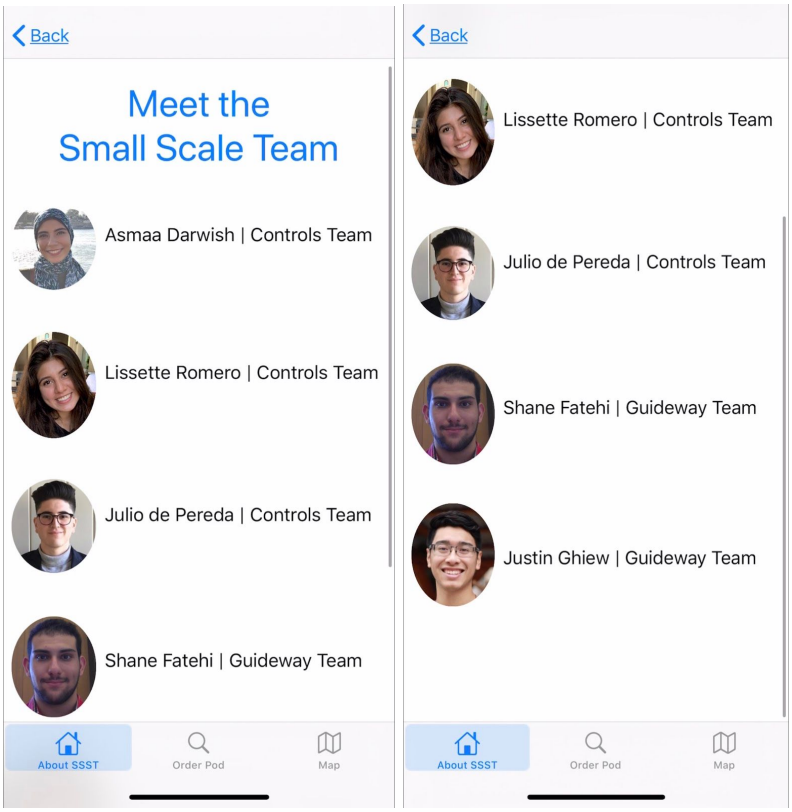


Figure 43. Meet the Small Scale Team "page" embedded in home tab.

In this screen, the user is greeted with a small introduction to the Small Scale Team. A scrolling feature was added so that all five team members could fit into this view controller. This allows the user to scroll up and down on this screen without any issues. Once they are done with this screen, the user can either click the "Back" button at the top left corner of the screen to go back to the main tab screen, or they can click on either of the two other tabs. If the latter option is selected, depending on whether the "Order Pod" tab or the "Map" tab, that respective tab screen will load. Pressing the "Map" button, will load the following screen shown in figure 44:

Figure 44. Map tab.

At this point since there is no other information or screen embedded in this tab, the user can go to any of the other two tabs. Selecting the "Order Pod" tab, directs the user to the following screen as shown in figure 45:

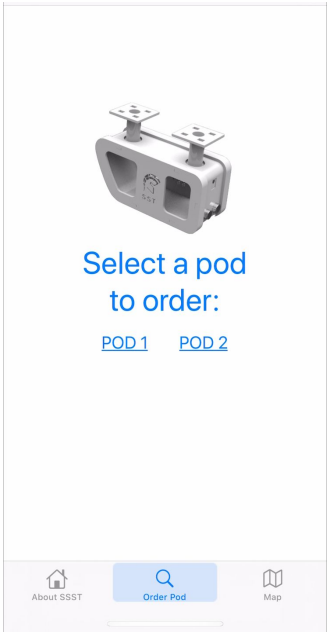


Figure 45. Order pod tab.

On this tab, the user can press either of the two buttons: “POD 1” or “POD 2”. Doing so will direct them to either of the two following screens shown in figures 46 and 47, depending on which button is pressed.

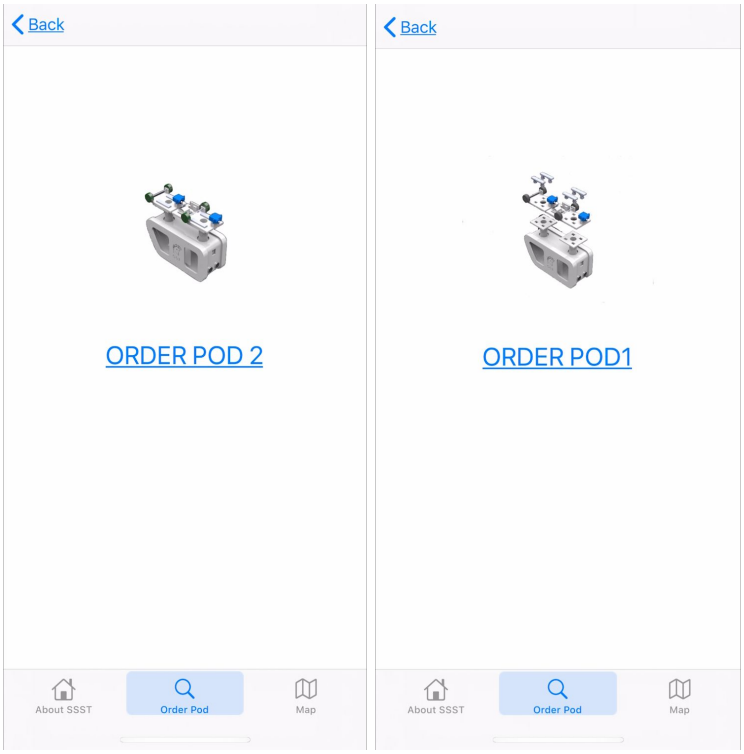


Figure 46. Order Pod 1 screen. **Figure 47.** Order Pod 2 screen.

At this point, depending on whether the user is viewing the screen on the left or on the right in the images shown above, the user will be able to either order pod 1 or pod 2. This feature is yet to be implemented. After this description of the Storyboard layout and looking back at figure 41, a clearer picture of how the Storyboard functions can be understood. The Storyboard map in figure 44 can be followed along, and the different choices a user can make can be pictured by following the arrows from view controller to view controller. As a brief example of this manner

in which the Storyboard map can be “read”. The following image 48, can be utilized as practice:

Figure 48. Example of how to read Storyboard map.

Starting from the left, the tab bar controller is the main controller which contains the tabs. The navigation controller enables the “Back” buttons to be implemented when a button is clicked and the user is directed to a screen within a specific tab. In this case, the two controllers on the left cannot be seen by the user, but they are “background” tools which are running. Greeted by the “Welcome to: SPARTAN Superway”, the arrow connecting this screen with the “Meet the Small Scale Team”, signifies that if the user will be directed from the main tab to the embedded team info screen. The rest of the Storyboard in figure X can be read in the same manner.

As mentioned briefly, the goal is to have the “Order Pod” tab be able to control the pods via the BLE module on the Arduino. To do this, a lot of research had to be done to see how this connection between an iOS app and an Arduino can take place. A clear guide is difficult to find; however, there are a lot of different iOS apps which are already able to get this connection. The code for these apps is available from Github and can be analyzed to see how the connection was

done. There are also several YouTube videos that attempt to explain this process. Apple also has some brief presentations with information about connecting to a BLE via an Xcode created app. However, there is no clear set of instructions, it takes a lot of piecing together to understand how this connection can be accomplished. In order to get an idea as to how to go about connecting the iOS app to the Arduino, some background research on how BLE technology works and how it can be utilized in Xcode.

BLE stands for Bluetooth Low Energy, it differs from normal bluetooth because as the name implies, BLE uses significantly less energy than standard bluetooth. This is the case because BLE devices, will be in a dormant phase, unless they are they are searched for with their Universally Unique Identifiers (UUID). This dormant phase allows for the BLE's battery life to last a lot longer, anywhere from months to a couple of years. In order to understand how to search for the BLE devices via the specific UUID, the concept of central and peripheral devices must be understood.

A central is a device which is seeking information from the peripheral device. Meanwhile, a peripheral is a device which has information or data that the central device can utilize. Apple explains this relationship between centrals and peripherals as a client and server relationship. In which the central is a client who is seeking information from a server. This relationship can be seen in figure 49:

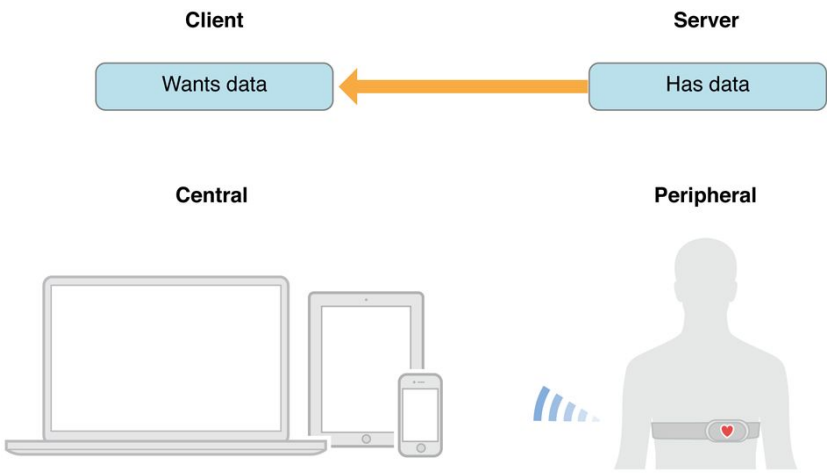


Figure 49. Central and peripheral visual representation. (Core Bluetooth objects on the central side, 2013)

The central device searches for the peripherals who are advertising themselves. Once a central device connects to a peripheral, then the peripheral can send information which is required by the central device. This is a very brief introduction to the concept of centrals and peripherals, but it is necessary as the iPhone via the iOS app will be acting as a central device, scanning for the peripheral device which would be the Arduino via the BLE. This is an abstract concept which the team is still working on understanding. Once this connection can be made between the Arduino and the iOS app, then more code on the Arduino side can begin to be developed.

After the intensive research regarding how BLE works and the manner in which it can be implemented into Xcode, the best source of information which was found was this article which listed the "12 Steps of BLE" (Hoyt, 2017). In order to test this, a separate test app was quickly created with a single view, which aimed to attempt to follow the steps presented in the article, this was done to see if a connection would be achieved. The twelve steps which the article presented can be summarized into: searching for BLE, connecting to BLE, reading or sending

information to and from the BLE. The twelve steps were followed but a connection was not successful, meaning that there is an error somewhere in the code. The test code for the connection written by following the 12 steps guideline can be seen in Appendix O. A challenge was also presented in the sense that some of the functions in Swift which the article was suggesting to use, were from a previous version of Xcode and Swift. So a lot of the commands and functions were outdated. Luckily, Xcode lets you know if you have functions or commands which are from a previous version of Xcode and it fixes them, or does its best. Sometimes these functions still need to be changed, so there is some discrepancy which could lead to potential errors. Apple also provides their own library of functions and commands called "CoreBluetooth" which is meant to facilitate this connection.

The attempt to connect the created test app to the BLE was not successful. At the current moment, the issue has not been detected as to why the iOS app running on the iPhone is not able to connect to the BLE. This is an issue which will be worked on during the break in between fall and spring semester. Overall, great progress has been made on the iOS app, however, there is a lot of progress to be made in the upcoming semester to meet the required goals. The next subsection will discuss the analysis of the Arduino code which has been created thus far.

3.2.3.b Arduino Code Analysis

To test and get familiar with how the BLE module works, a simple code was created to control an LED with an iPhone app. A suggested iOS app called "HM10 bluetooth Serial Lite" was used that allows for a serial input from any iphone. The Arduino code shown in appendix H is programmed to receive signals from the BLE. A flowchart of the code is shown in Figure 50. If the input signal from the app has a value of one, the LED will turn on and if the value is zero, it

will turn off. Connections between the Arduino mega, LED, and the BLE module are shown in figure 51. With very few components, an LED, Arduino, and the BLE module, it was easy to create a simple logic that provided the basic knowledge and enough practice with the BLE module before integrating it into a more complex code. By conducting this simple code, the team is able to prove the functionality of the HM-10 and its ability to connect the arduino to an iOS app.

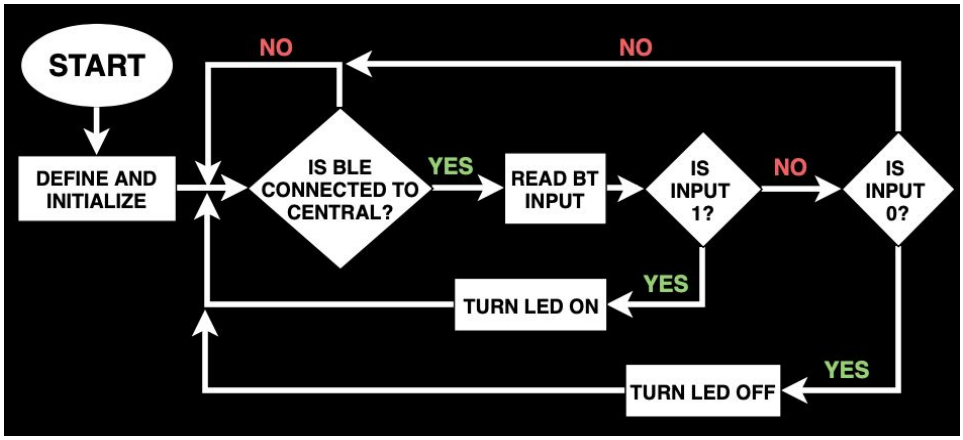


Figure 50. Flowchart of BLE test code

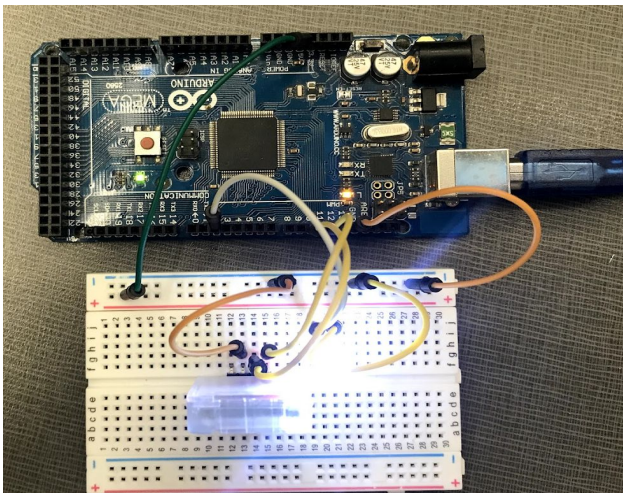


Figure 51. Connection between Aduino and BLE

After getting familiar with the BLE module, the team was confident in integrating it into a more complex code. The BLE Test Code in Appendix H and the Ultrasonic Test code in Appendix F were merged together to create the anti-collision system in Appendix I. A function called CheckDistance() was created to add all the logical steps of the ultrasonic test code. By calling this function, the code reads the ultrasonic value and sets the speed of the motor accordingly. Figure 28 shows the flowchart that explain the code in detail. The use of functions in the Arduino code will be used often to create a clear and easy to follow code.

3.4 Plans for Fabrication

Each respective subteam has their own specific plans for fabricating.

3.4.1 Guideway Fabrication

It was determined that waterjet cutting is the best mode of fabricating the track pieces. Initially, the team considered machining each track piece manually. Several components of the track were made manually using a Skil electric saw and Dremel as shown in Figure 52. The round corners were difficult to cut using the Dremel, which resulted in an uneven cut. Furthermore, the guideway will be fabricated using cutting equipment from the Makerspace and Central Machine Shop. Each track piece and the base will be fabricated using a waterjet cutter. The plywood used for the track pieces and base will be purchased from Home Depot or Lowes. Threaded rods, studs, washers, and nuts will be purchased from McMaster-Carr, an online site with over 580,000 products.

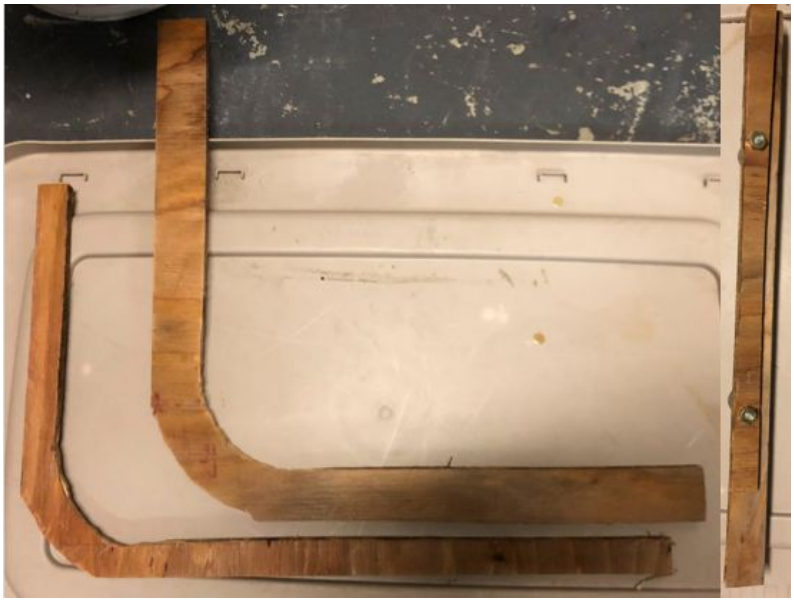


Figure 52. Hand cut track pieces.

The waterjet cutter offered at Makerspace is OMAX series and all documentation can be found on the manufacturers website. The OMAX MAXIEM waterjet cutter offers a linear positional accuracy of 0.006” (MAXIEM, 2019). The accuracy of the track pieces only need to be within ten-thousandths of an inch tolerance. The x-y cutting envelope is five feet by five feet, which is plenty of space to cut our desired guideway pieces.

3.4.2 Bogie Fabrication

All parts of the bogie will be fabricated using a 3D printer. The material used will be PLA, which is one of the cheapest 3D filament materials and one of the easiest to work with. Different prototypes will be printed before achieving the desired final design. The electrical components on the boogie will be secured using the proper hardware, such as screws.

3.4.3 Controls Fabrication/Testing

All connections of the electronics will be done using jumper wires and a solderless breadboard. This is best for constant troubleshooting and testing. Once a final connection diagram is made, connections can be converted to a soldered breadboard for a more permanent and stable connection between the components. In terms of the app, Xcode will continue to be used for the development and testing of the app. Once a successful connection is made between the iOS app and the BLE, along with the overall design, the app will be submitted to Apple so that it can be published in the app store. Afterwards, whoever is interested or future senior project teams that take over the project in the following years can download it. In the meantime, the app will be tested by transferring the app directly from Xcode to the desired iPhone.

4. Conclusion and Next Steps

At the end of this semester, each subteam has made good progress toward the overall teams' goals. The team is on track with the gantt chart in Appendix J. The guidway team was able to design and model each guideway piece in SolidWorks. Analysis of the track will be performed in ANSYS and Solidworks, with careful consideration to what variables are being applied. For example, instead of just testing the load of one pod, the team can analyze how the track will react to multiple pods being used at once. In addition, all of the planning for how the guideway will be assembled has been organized and any challenges that may arise have been thought through. The track pieces in SolidWorks will be sent to the waterjet for cutting and assembling for the guideway will begin early next year. To minimize cost as much as possible, the guideway team will use hardware and plywood from the Superway Design Center and purchase any additional parts from Home Depot or McMaster-Carr. The next critical step is to design a way to mount the

y-switch hanging third rail onto the guideway. The design needs to be rigid enough to withstand the weight of the pod and bogie. Furthermore, the guideway team is expecting to have a test track assembled by the end of January.

For the bogie, a completed CAD model of the pod car is completed and the first prototype is ready to be 3D printed. The next critical step for the bogie is designing the internal gears and the mounting parts for the motor as well as the servos for the switch. Once the first iteration of the guidway is built, the team will start designing a bogie that integrates with the track. The bogie team and the guidway team will work together to come up with a reliable switching mechanism that reduces the risk of falling.

For the controls team, the layout of the app is complete and a lot of progress has been done into the arduino code. For the app, the next steps are to continue researching and testing the connection between the iOS app and the Arduino. This will be the most difficult and crucial task for the app. The UI of the app will also continue to be developed, animations will be added when transitioning from tab to tab or when pressing buttons. A more detailed team introduction will be implemented and the third and fourth tabs will be finalized. The app will also be submitted to Apple so that it can be published in the App Store.

For the Arduino code, the anti-collision system is completed. The team has enough knowledge of BLE and how it works with the arduino. The next step is to program the Pixy2 camera with the Arduino to recognize station names and different junction signs. A code for the switching mechanism using servos will be conducted as well. A combined Arduino code will consist of the anti-collision system, sign recognition system, switching mechanism, and a connection with the BLE.

References

Alvarez, R., et. al., 2016. Spartan Superway: A Solar Powered Automated Public Transportation System. Retrieved from http://tinyurl.com/jbla3hz

Andrews, L. W. (n.d.). How to Stress Less in a Traffic Jam. Retrieved December 7, 2019, from https://www.psychologytoday.com/us/blog/minding-the-body/201509/how-stress-less-in-traffic-jam

SwiftUI. (n.d.). [Illustration]. Retrieved from https://developer.apple.com/xcode/swiftui/

Core Bluetooth objects on the central side. (2013). [Illustration]. Retrieved from https://developer.apple.com/library/archive/documentation/NetworkingInternetWeb/Conceptual/CoreBluetooth_concepts/CoreBluetoothOverview/CoreBluetoothOverview.html

Aquino, B., et. al., 2017. Spartan Superway: A Solar Powered Automated Public Transportation System. Retrieved from https://www.inist.org/library/2017-06-02.Aquino%20etal.Spartan%20Superway%202016-2017%20Final%20Report.SJSU%20ME195AB.pdf

Boeing. (n.d.). Personal Rapid Transit System Historical Snapshot. Retrieved from http://www.boeing.com/history/products/personal-rapid-transit-system.page

Bogies | The Railway Technical Website | PRC Rail Consulting Ltd. (n.d.). Retrieved September 29, 2019, from http://www.railway-technical.com/trains/rolling-stock-index-l/bogies.html

Chen, X., & Liu, Y. (2019). *Finite Element Modeling and Simulation with Ansys Workbench*. Boca Raton, Fl: CRC Press.

Cowley, A., et. al., 2014. Spartan Superway: A Solar Powered Automated Public Transportation System. Retrieved from http://tinyurl.com/hcvl893

Faas, C. (2019). *Back to School Parking and Traffic*. Retrieved from http://www.sjsu.edu/parking /?mkt_tok=eyJpIjoiTVdRMU5qazFPVGRpWVRJMiIsInQiOiJwWlByRkZyZTZJMTZEVHZuQXplNzhcL1pVRjBidEhjdERpYlwvc0VodjNqM29qaGdVd0g4NlFJN3pKblRFUHRnU2Q4T01ZR1lya2srRU53bE53OUJpRkZzVTV1YXNOVzBQWTZrY1wvQVord1ZJSVwvZFBoSVhZWjhjSlJaQ3V6ejNyS20ifQ%3D%3D

Furman, B. J. (2016). The Spartan Superway: A Solar-Powered Automated Transit Network. Retrieved from http://proceedings.ises.org/paper/solar2016/solar2016-0019-Furman.pdf

Furman, B., Ellis, S., Mueller, P., & Swenson, R. (2014). Automated Transit Networks (ATN): A Review of the State of the Industry and Prospects for the Future. *Mineta Transportation Institute*, 1–220. Retrieved from https://transweb.sjsu.edu/sites/default/files/1227-automated-transit-networks.pdf

Halabi, I & Jocson, B, 2018. Spartan Superway: Small-Scale Bogie Design. Retrieved from https://drive.google.com/drive/folders/1G644XhphRf8OwJpp2lZjsDeMial7oxoP

Hoyt, K. (2017, October 12). The 12 Steps of Bluetooth (Swift). Retrieved December 15, 2019, from https://www.kevinhoyt.com/2016/05/20/the-12-steps-of-bluetooth-swift/

Leiding, K. (n.d.). Gleismannsbahnhof platform 15.3 The cabin taxi test facility in the Hagen lobby. Retrieved from http://www.gleismann.de/15.cabinentaxi/c3.html

MAXIEM. (2019, October 30). MAXIEM 1515: Abrasive Water Jet Cutter Machine. Retrieved from https://www.omax.com/maxiem-waterjet/1515

Morgantown Personal Rapid Transit System. (n.d.). System Operations Description Manual. Retrieved from http://www.advancedtransit.org/wp-content/uploads/2011/09/M-PRT_1-1_SYSTEM_OPERATIONS_DESCRIPTION_MANUAL.pdf

Nordstrom, R. (2009). Heathrow Tests Personal Rapid Transit System. *Airport Improvement* . Retrieved from https://web.archive.org/web/20160304025447/http://prtconsulting.com/docs/Heathrow.pdf

Ornellas, D., et. al., 2015. Spartan Superway: A Solar Powered Automated Public Transportation System. Retrieved from http://tinyurl.com/gnfsqt

Appendices

Appendix A: Ultrasonic Sensor DataSheet

ELEC Freaks

Tech Support: services@elecfreaks.com

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:
(1) Using IO trigger for at least 10us high level signal,
(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
(3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.
Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

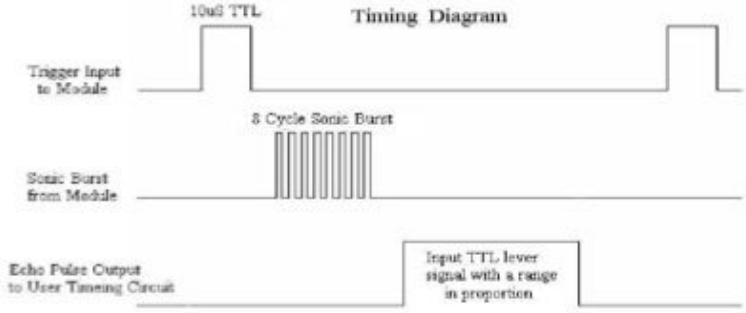
Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

Vcc Trig Echo GND

Timing diagram

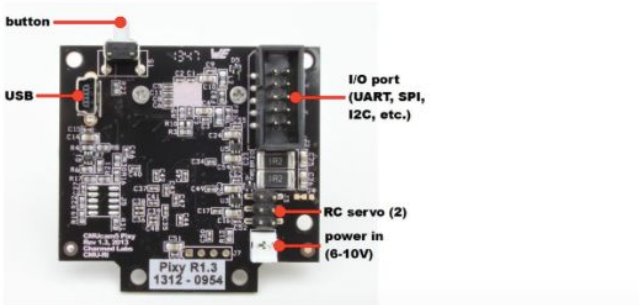
The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



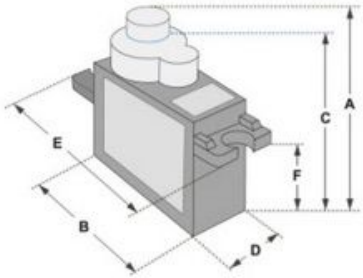
Appendix B: Pixy2 DataSheet

Technical specs

- Processor: NXP LPC4330, 204 MHz, dual core
- Image sensor: Omnivision OV9715, 1/4", 1280x800
- Lens field-of-view: 75 degrees horizontal, 47 degrees vertical
- Lens type: standard M12 (several different types available)
- Power consumption: 140 mA typical
- Power input: USB input (5V) or unregulated input (6V to 10V)
- RAM: 264K bytes
- Flash: 1M bytes
- Available data outputs: UART serial, SPI, I2C, USB, digital, analog
- Dimensions: 2.1" x 2.0" x 1.4
- Weight: 27 grams

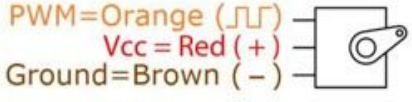


Appendix C: Servo Motor Data Sheet



Dimensions & Specifications
A (mm) : 32
B (mm) : 23
C (mm) : 28.5
D (mm) : 12
E (mm) : 32
F (mm) : 19.5
Speed (sec) : 0.1
Torque (kg-cm) : 2.5
Weight (g) : 14.7
Voltage : 4.8 - 6

Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "-90" (~1ms pulse) is all the way to the left.



Appendix D: HM10 DataSheet

HM-10 DataSheet

Welcome to DSD ECH branded products!

If you have any questions, please contact us: Info@sihaicorp.com

1. Product Parameters

BT Version: Bluetooth Specification V4.0 BLE

Working frequency: 2.4GHz ISM band

Modulation method: GFSK(Gaussian Frequency Shift Keying)

RF Power: -23dbm, -6dbm, 0dbm, 6dbm

Speed: Asynchronous: 2-6K Bytes

Synchronous: 2-6K Bytes

Security: Authentication and encryption

Service: 0xFFE0 (Modifiable use AT+UUID command)

Characteristic: 0xFFE1 (Modifiable use AT+UUID command)

Characteristic: Notify and Write (Modifiable use AT+UUID command)

Power: +2.5V~3.3VDC 50mA

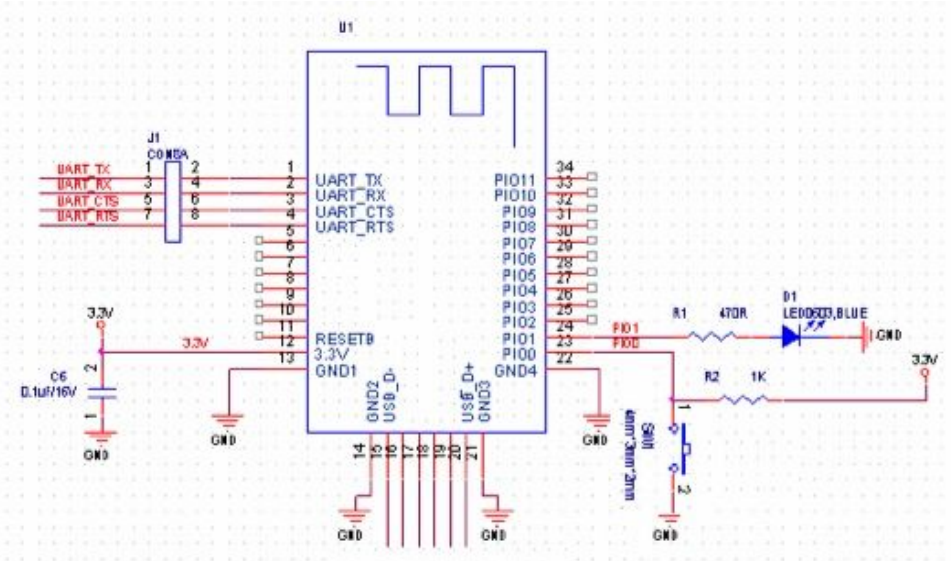
Power: Active state 8.5mA; Sleep state 50~200uA

Working temperature:–20 ~ +95 Centigrade

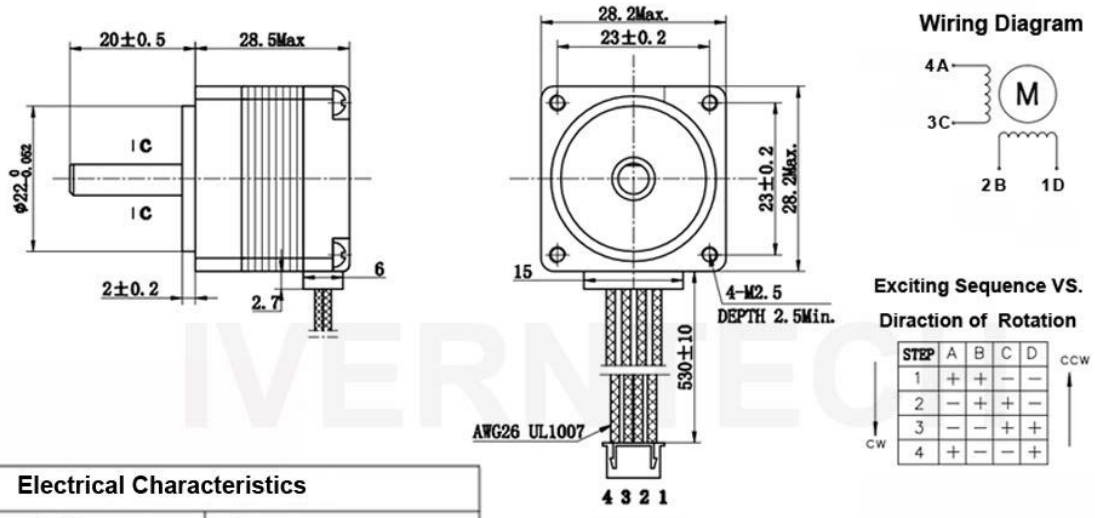
Size: HM-10 27mm x 13mm x 2.2 mm

Size: HM-11 18mm x 13mm x 2.2mm

Size: HM-15 65mm x 32mm x 16mm



Appendix E: Motor DataSheet



Electrical Characteristics	
1. Number of Phase	2
2. Step Angle	1.8 °
3. Rated Voltage	3.8 V DC
4. Rated Current	0.8 A
5. Holding Torque	65 mN.m Min(2 Phases)
6. Phase Resistance	4.8 Ω ± 10% (20° C)
7. Phase Inductance	2.8mH±20%(1kHz 1V rms)
8. Motor Inertia	7.5g.cm^2
9. Motor Weight	80g Ref.
10. Insulation Resistance	100MΩ Min.(DC 500V)
11. Insulation Class	B(130° C)

Appendix F: Ultrasonic Test Code

Ultrasonic

```
#include <Stepper.h>

const int stepsPerRevolution = 200;      //RPM
Stepper myStepper(stepsPerRevolution, 4, 5, 6, 7);

const int trigPin = 9;
const int echoPin = 8;
long duration;
int distance;

void setup()
{
```

```
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT);  // Sets the echoPin as an Input
  myStepper.setSpeed(100);
  Serial.begin(9600);       // Starts the serial communication
}

void loop()
{
  digitalWrite(trigPin, LOW); // Clears the trigPin
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  /* Reads the echoPin, returns the sound wave travel time in
  microseconds */
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2;

  if (distance > 25) //safe distance of 25cm
  {
    myStepper.setSpeed(160);
    myStepper.step(stepsPerRevolution);
    Serial.print(distance);
    Serial.println(" >25");
  }
    else if ( distance <= 25 && distance > 15)
    // slow speed distance between 25cm and 15cm
  {
    myStepper.setSpeed(35);
    myStepper.step(stepsPerRevolution);
    Serial.print(distance);
    Serial.println(" <25");
  }
    else if ( distance <= 15 && distance > 3)
    // slow speed distance between 15cm and 3cm
  {
    myStepper.setSpeed(15);
    myStepper.step(stepsPerRevolution);
    Serial.print(distance);
    Serial.println(" <15");
  }
  else //collision distance of 3cm, motor stops
  {
    myStepper.step(0);
```

```
    Serial.print(distance);
    Serial.println(" < 3");
    delay(100);
  }
}
```

Appendix G: BLE Configuration Code

SoftwareSerialExample

```
/*
 Software serial multiple serial test
 Receives from the hardware serial, sends to software serial.
 Receives from software serial, sends to hardware serial.

 The circuit:
 * RX is digital pin 10 (connect to TX of other device)
 * TX is digital pin 11 (connect to RX of other device)

 Note:
 Not all pins on the Mega and Mega 2560 support change interrupts,
 so only the following can be used for RX:
 10, 11, 12, 13, 50, 51, 52, 53, 62, 63, 64, 65, 66, 67, 68, 69

 Not all pins on the Leonardo and Micro support change interrupts,
 so only the following can be used for RX:
 8, 9, 10, 11, 14 (MISO), 15 (SCK), 16 (MOSI).

 created back in the mists of time
 modified 25 May 2012
 by Tom Igoe
 based on Mikal Hart's example

 This example code is in the public domain.
 */

#include <SoftwareSerial.h>

SoftwareSerial mySerial(10, 11); // RX, TX

void setup()
{
  // Open serial communications and wait for port to open:
  Serial.begin(57600);
  while (!Serial)
  {
```

```
    ; // wait for serial port to connect. Needed for native USB port
only
  }
  Serial.println("Goodnight moon!");
  // set the data rate for the SoftwareSerial port
  mySerial.begin(4800);
  mySerial.println("Hello, world?");
}

void loop()
{ // run over and over
  if (mySerial.available())
  {
    Serial.write(mySerial.read());
  }
  if (Serial.available())
  {
    mySerial.write(Serial.read());
  }
}
```

Appendix H: BLE Test Code

HM10

```
#include <SoftwareSerial.h>
#define LED_PIN 2
int BT_input;
SoftwareSerial mySerial(10, 11); // RX, TX

void setup()
{
  Serial.begin(9600);
  mySerial.begin(9600); // BLE serial
  pinMode(LED_PIN, OUTPUT);
}

void loop()
{
  if (mySerial.available())
  {
   BT_input = mySerial.read();

    if (BT_input== 49) //1 in ASCII from char to dec
    {
      // Non-zero input means "turn on LED".
      Serial.println("Got input:");
      Serial.println("  on");
```

```
      digitalWrite(LED_PIN, HIGH);
    }
    if (BT_input== 48) //0 in ASCII from char to dec
    {
      // Input value zero means "turn off LED".
      Serial.println("Got input:");
      Serial.println("  off");
      digitalWrite(LED_PIN, LOW);
    }
  }
}
```

Appendix I: BLE and Ultrasonic Anti-collision Code

HM10 and Ultrasonic

```
#include <Stepper.h>
#include <SoftwareSerial.h>

int BT_input; // BLE input variable

const int stepsPerRevolution = 200;  //stepper motor

Stepper myStepper(stepsPerRevolution, 4, 5, 6, 7); // stepper motor
pins
SoftwareSerial mySerial(10, 11); // BLE pins RX, TX

const int trigPin = 9;  // ultrasonic
const int echoPin = 8;  // ultrasonic

// defines variables for ultrasonic
long duration;
int distance;

void setup()
{ // Clears the trigPin
  pinMode(trigPin, OUTPUT);  // Sets the trigPin as an Output
  pinMode(echoPin, INPUT);   // Sets the echoPin as an Input

  myStepper.setSpeed(100);   // reset motor speed
  Serial.begin(9600);
  mySerial.begin(9600);      //BLE serial
}

void loop()
{
  if (mySerial.available())   // send data only when data is received
  {
```

```
    BT_input = mySerial.read();  // read the incoming byte from BLE

    while (BT_input == 49)       //1 in ASCII from char to dec
    {
      Serial.print("Got input:");
      Serial.println(BT_input);
      CheckDistance();           //read ultrasonic and run motor
      Serial.println("motor on");
    }
    while (BT_input == 48)       //0 in ASCII from char to dec
    {
      Serial.print("Got input:");
      Serial.println(BT_input);
      myStepper.step(0);        //turn motor off
      myStepper.setSpeed(0);
      Serial.println("motor on");
    }
  }
}
void CheckDistance()               // check distance and slowing
down
{
  digitalWrite(trigPin, LOW);       // Clears the trigPin
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;  // Calculating the distance

  if (distance > 25) //safe distance of 25cm
  {
    myStepper.setSpeed(160);
    myStepper.step(stepsPerRevolution);
    Serial.print(distance);
    Serial.println(" >25");
  }
  else if ( distance <= 25 && distance > 15)
  // slow speed distance between 25cm and 15cm
  {
    myStepper.setSpeed(35);
    myStepper.step(stepsPerRevolution);
    Serial.print(distance);
    Serial.println(" <25");
```

```
  }
  else if ( distance <= 15 && distance > 3)
  // slow speed distance between 15cm and 3cm
  {
    myStepper.setSpeed(15);
    myStepper.step(stepsPerRevolution);
    Serial.print(distance);
    Serial.println(" <15");
  }
  if (distance <= 3) //collision distance of 3cm, motor stops
  {
    myStepper.step(0);
    myStepper.setSpeed(0);
    Serial.print(distance);
    Serial.println(" < 3");
  }
}
```

Appendix J: Small Scale Team Gantt Chart

				September 2019			October 2019				November 2019			
No.	Owner	Task Description	Due	16-Sep	23-Sep	30-Sep	7-Oct	14-Oct	21-Oct	28-Oct	4-Nov	11-Nov	18-Nov	25-Nov
1	All	Identify Issues	16-Sep											
2	All	Develop Scope of Work	16-Sep											
3	All	ME195A Presentation 1	25-Sep											
4	All	Research	14-Oct											
5	Asmaa , Lissette and Julio	Indentify Electrical Hardware	7-Oct											
6	Asmaa , Lissette and Julio	Purchase Electrical Har	14-Oct											
7	Shane and Justin	CAD the Bogie body	21-Oct											
8	Shane and Justin	Print the Bogie body	11-Dec											
9	Shane and Justin	CAD the Track	14-Oct											
10	Shane and Justin	CAD the Y-junction	14-Oct											
11	Shane and Justin	CAD the switch mechanism	14-Oct											
12	Shane and Justin	CAD the track connection	14-Oct											
13	Shane and Justin	Indentify Mechanical Hardware	21-Oct											
14	Shane and Justin	Purchase Mechanical H	28-Oct											
15	Shane and Justin	Fabricate Track	11-Nov											
16	Shane and Justin	Assemble Track	18-Nov											
17	Shane and Justin	Assemble Bogie	18-Nov											
18	Asmaa , Lissette and Julio	Create Arduino circuit s	14-Oct											
19	Asmaa , Lissette and Julio	Develop a Flow chart Diagram of Arduino Code	21-Oct											
20	All	ME195A Presentation 2	23-Oct											
20	Asmaa , Lissette and Julio	Develop Flow Diagram of App	28-Oct											
21	Asmaa , Lissette and Julio	Develop draft 1 of Arduino code	28-Oct											
22	Asmaa , Lissette and Julio	Test Arduino code 1	4-Nov											
23	Asmaa , Lissette and Julio	Develop draft 2 of Arduino code	18-Nov											
24	Asmaa , Lissette and Julio	Test Arduino code 2	2-Dec											
25	Asmaa , Lissette and Julio	Develop draft 1 of iOS code	4-Nov											
26	Asmaa , Lissette and Julio	Test iOS code 1	18-Nov											
27	All	ME195A Presentation 3	20-Nov											
27	Asmaa , Lissette and Julio	Develop draft 2 of iOS code	2-Dec											
28	Asmaa , Lissette and Julio	Test iOS code 2	9-Dec											
29	Asmaa , Lissette and Julio	Get app to connect to Bluethooth	9-Dec											
30	All	Project Report	9-Dec											

				Februrary 2020				March 2020				April 2020					
No.	Owner	Task Description	Due	5-Feb	12-Feb	19-Feb	26-Feb	4-Mar	11-Mar	18-Mar	25-Mar	1-Apr	8-Apr	15-Apr	22-Apr	29-Apr	
31	Asmaa , Lissette and Julio	Develop draft 2 of iOS code	12-Feb														
32	Asmaa , Lissette and Julio	Test iOS code 2	12-Feb														
33	All	Test Bogie on Track with Arduino	12-Feb														
34	All	Test 2: Bogie on track with current Arduino Script	12-Feb														
35	Asmaa and Lissette	Fix bugs in Arduino	26-Feb														
36	All	Minor adjustments to bogie and track	26-Feb														
37	Asmaa , Lissette and Julio	Get app to control the bogie w/ BLE	11-Mar														
38	All	Test 3: control bogie on track with iOS app	25-Mar														
39	Asmaa and Lissette	Final debugging in Arduino	25-Mar														
40	Julio	Finial debugging in swift	8-Apr														
41	Shane and Justin	Final adjustments to guieway	8-Apr														

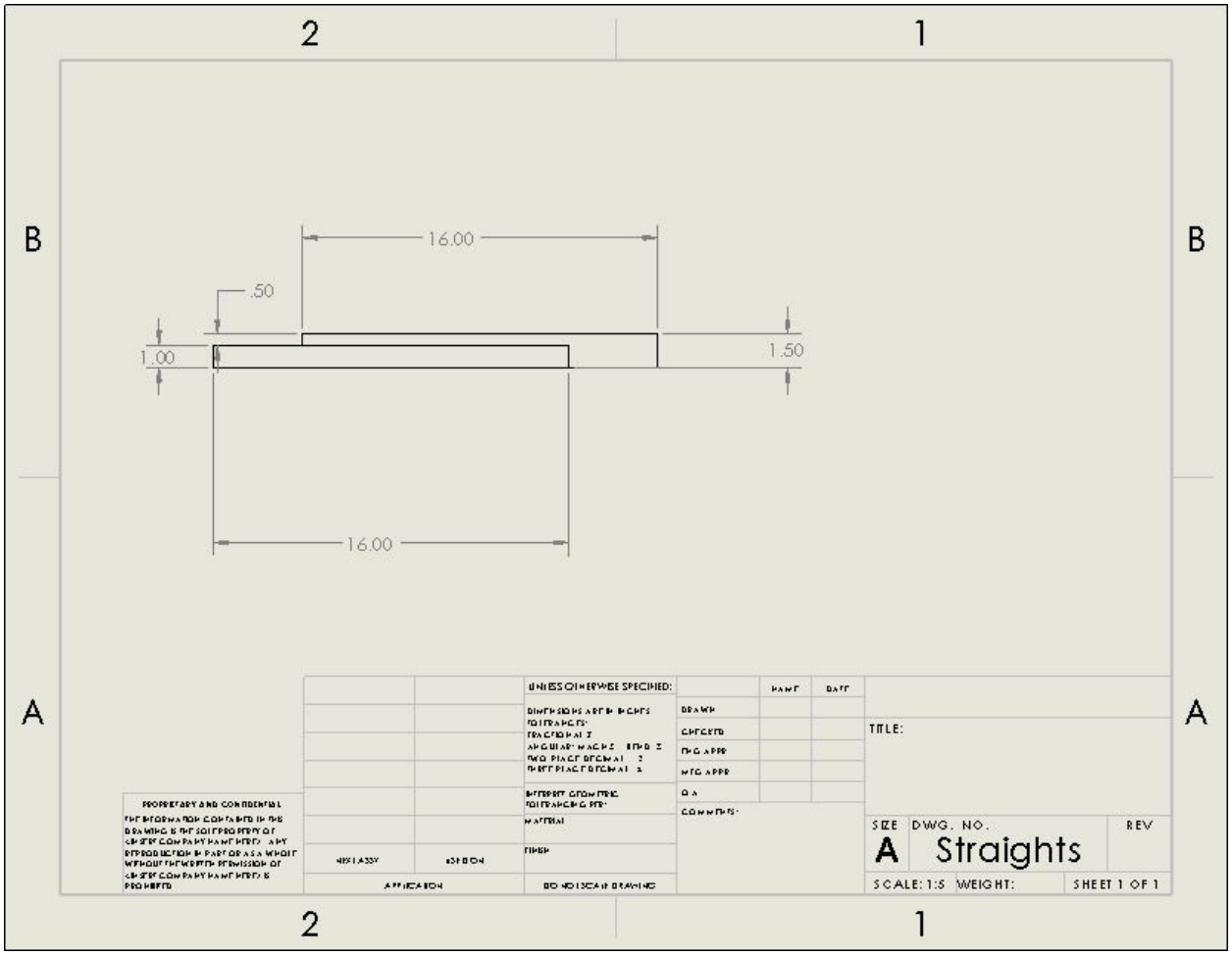
Appendix K: Guideway Material Properties

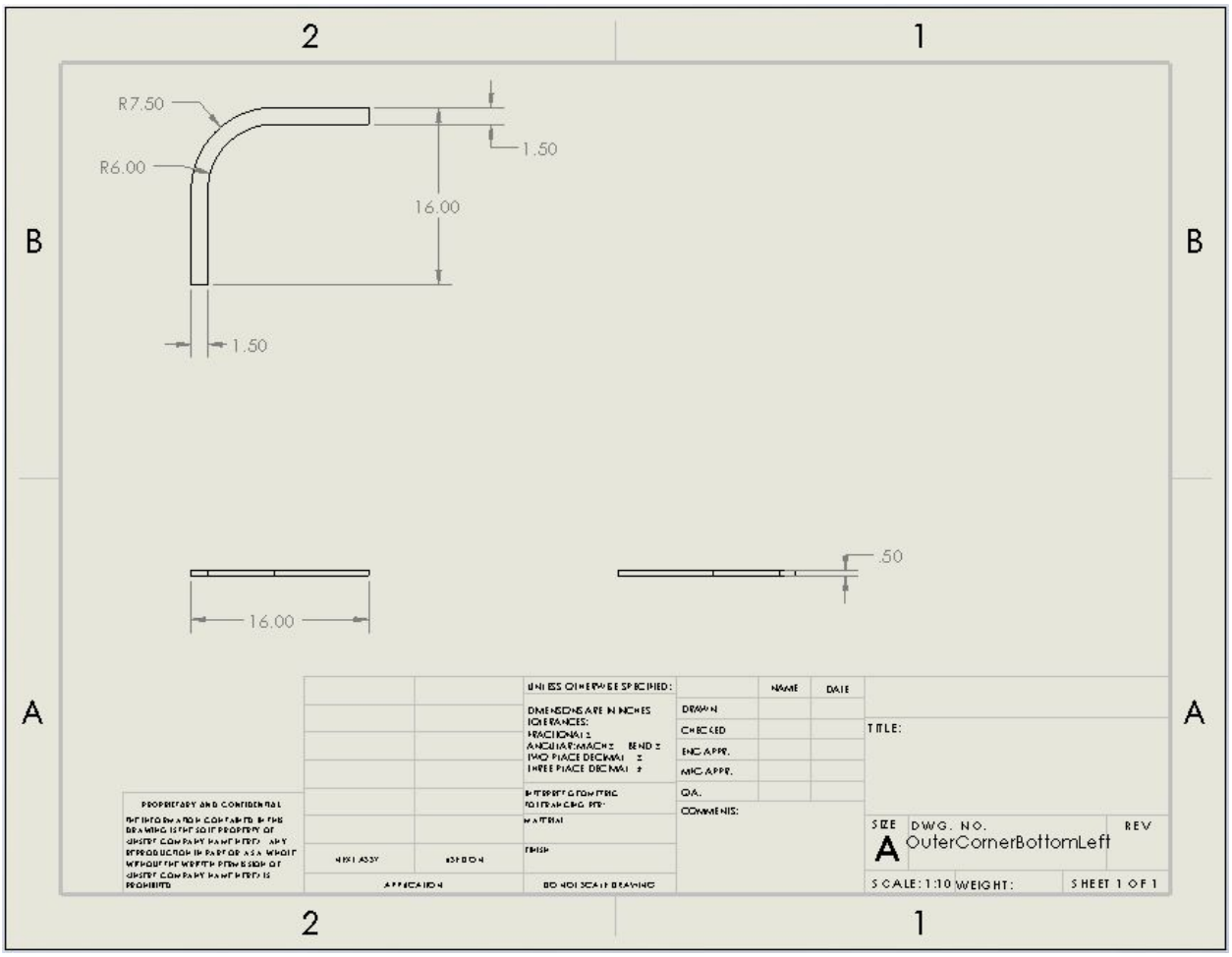
Material	Property	Value/Range	Unit
Plywood	Flexural Modulus	1000-1400	ksi
	Poisson's Ratio	0.1-0.3	-
	Shear Modulus	20.0-30.0	ksi
Structural Steel	Tensile Strength	150	ksi

Appendix L: Guideway Bill of Materials

Item	Description	Qty.
1	Outer Corner Top Left (OCTL)	2
2	Outer Corner Bottom Left (OCBL)	2
3	Top Left Inner Corner (TLIC)	2
4	Bottom Left Inner Corner (BLIC)	2
5	Top Right Inner Corner (TRIC)	2
6	Bottom Right Inner Corner (BRIC)	2
7	Bottom Inner Corner (BIC)	2
8	Top Inner Corner (TIC)	2
9	Straights	30
10	Base (Plywood)	4
11	Threaded Rods (Steel)	30
12	Nuts	120
13	Washers	120
14	PVC Tube	30

Appendix M: Guideway Drawings





Appendix N: iOS App First Iteration Code

```
//
//  ContentView.swift
//  SPARTAN Superway
//
//  Created by Cesar de Pereda on 10/8/19.
//  Copyright © 2019 Cesar de Pereda. All rights reserved.
//

import SwiftUI

struct ContentView: View {

    init(){
        UITabBar.appearance().backgroundColor = UIColor.white
    }
```

```
  @State private var selection = 0

  var body: some View {
     TabView(selection: $selection){
        Text("Welcome to: \r SPARTAN Superway")
           .font(.title)
           .foregroundColor(Color.blue)
           .multilineTextAlignment(.center)
           .tabItem{
              VStack {
                 Image("first")
                 .padding(.top)
                 Text("Home")
                  .multilineTextAlignment(.center)
              }
           }

           .tag(0)
        Text("View Available Pods")
           .font(.title)
           .foregroundColor(Color.blue)
           .tabItem {
              VStack {
                 Image("Image")
                    .padding(.top)
                 Text("Order Pod")
                    .multilineTextAlignment(.center)
              }
           }
           .tag(1)
        Text("View Map")
        .font(.title)
        .foregroundColor(Color.blue)
        .multilineTextAlignment(.center)
        .tabItem {
           VStack {
              Image("second")
               .padding(.top)
              Text("Map")
               .multilineTextAlignment(.center)
           }
        }
        .tag(2)
     }.accentColor(Color.white)
  }
}
```

```
struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```

Appendix O: BLE Connection Test Code:

```
import UIKit
import CoreBluetooth


class ViewController:-
    UIViewController,
    CBCentralManagerDelegate,
    CBPeripheralDelegate {

    var manager:CBCentralManager!
    var peripheral:CBPeripheral!

    let BLE1 =  "BLEONE"
    let BLE1_SCRATCH_UUID =
        CBUUID(string: "59E8F9AC-279B-3E8B-8420-662D02174DE6")
    let BLE1_SERVICE_UUID =
        CBUUID(string: "59E8F9AC-279B-3E8B-8420-662D02174DE6")

    override func viewDidLoad(){
        super.viewDidLoad()
        manager = CBCentralManager(delegate: self, queue: nil)
    }

    func centralManagerDidUpdateState(_ central: CBCentralManager){
        if central.state == CBManagerState.poweredOn{
            central.scanForPeripherals(withServices: [BLE1_SERVICE_UUID], options: nil)
        }
        else
        {
            print("Bluetooth not available")
        }
    }

    private func centralManager(
        central: CBCentralManager,
        didDiscoverPeripheral peripheral: CBPeripheral,
        advertisementData: [String: AnyObject],
        RSSI: NSNumber){
```

```
    let device = (advertisementData as NSDictionary)
    .object(forKey: CBAdvertisementDataLocalNameKey)
    as? NSString

    if device?.contains(BLE1) == true {
        self.manager.stopScan()

        self.peripheral = peripheral
        self.peripheral.delegate = self

        manager.connect(peripheral, options: nil)

    }
}

// Get Services
func centralManager(_ central: CBCentralManager, didConnect peripheral: CBPeripheral) {
    peripheral.discoverServices(nil)
}

// Get characteristics
func peripheral(_ peripheral: CBPeripheral, didDiscoverServices error: Error?) {
    for service in peripheral.services! {
        let thisService = service as CBService

        if service.uuid == BLE1_SERVICE_UUID{
            peripheral.discoverCharacteristics(nil, for: thisService)
        }
    }
}

// Setup Notifications
private func peripheral(
    peripheral: CBPeripheral,
    didDiscoverCharacteristicsForService service: CBService,
    error: NSError?){
    for characteristic in service.characteristics! {
        let thisCharacteristic = characteristic as CBCharacteristic

        if thisCharacteristic.uuid == BLE1_SCRATCH_UUID{
            self.peripheral.setNotifyValue(true, for: thisCharacteristic)
        }
    }
}

//Changes are coming
```

```
    func peripheral(_ peripheral: CBPeripheral, didUpdateValueFor characteristic: CBCharacteristic, error:
Error?) {
        var count:UInt32 = 0; //need to see what this is

        if characteristic.uuid == BLE1_SCRATCH_UUID{
            func copyBytes(to: UnsafeMutableBufferPointer<UInt32>, count: Int)
            {
            accessibilityLabel =
                NSString(format: "%llu", count) as String}
        }
    }

    func centralManager(_ central: CBCentralManager, didDisconnectPeripheral peripheral: CBPeripheral,
error: Error?) {
        central.scanForPeripherals(withServices: [BLE1_SERVICE_UUID], options: nil)
    }

        // Do any additional setup after loading the view.
}
```