Autonomous Tennis Ball Collector

Final Design Review (FDR)

By

Frances Belcher

fbelcher@calpoly.edu

Matthew Hoffman

mhoffm12@calpoly.edu

Robert Lutrell

reluttre@calpoly.edu

Alex Petrov

abpetrov@calpoly.edu

Michael Yiu

mwyiu@calpoly.edu

Sponsor

Dr. John Chen

Mechanical Engineering Department

California Polytechnic State University

San Luis Obispo

June 2021

Statement of Disclaimer

Since this project is a result of a class assignment, it has been graded and accepted as fulfillment of the course requirements. Acceptance does not imply technical accuracy or reliability. Any use of information in this report is done at the risk of the user. These risks may include catastrophic failure of the device or infringement of patent or copyright laws. California Polytechnic State University at San Luis Obispo and its staff cannot be held liable for any use or misuse of the project.

Abstract

Practicing tennis often involves hitting many tennis balls from one side of the court to the other without an opponent to hit the balls back. In training sessions like these, the task of collecting the balls is laborious when performed manually. The objective of this project is to develop a robotic tennis ball collector that can automatically collect the balls from one side of the court so that the player can rest rather than collect the balls manually. This document outlines the process of designing such a robot. Included in this report is background research, prototype, and concept modeling, along with a finalized design, and a complete timeline of our process. We will also detail the manufacturing process and the design verification. In the conclusion we will provide you with recommendations for future projects. Throughout our research, we discovered many similar products, but none met all of the customer's requirements, thus opening a window for our product. After copious design consideration, we selected the strongest idea that satisfied our customers' needs and are moving forward with structural modeling and preliminary analysis on it.

After the structural prototype revealed issues in the design we went back to work and finalized a design that we felt confident with and still satisfied all the requirements. As seen in this report the final design utilizes structural framing materials to build the robot and allows for ease of attachment for all the electrical components. The final step in the design process was to test the verification prototype to ensure that it met all our specifications. Unfortunately, our design did not pass as many of the tests as we would have liked, and this is detailed in that section. While at the conclusion of this project, we did not complete as much as we hoped, there is a good foundation in place for the project to continue as our sponsor so desires.

Table of Contents

List of Tables and Figures

Tables

Figures

1.0 Introduction

In this report, we will detail our research, objectives, design process and timeline for the project we have been tasked with completing. This project team is comprised of Frances Belcher, Matthew Hoffman, Robert Luttrell, Michael Yiu, and Alex Petrov. Frances, Matthew, Robert, and Michael are all senior Mechanical Engineering students with an interest in robotics, while Alex is a senior Computer Engineering student who joined a little later into the project to assist with the software. We are all interested in this project for a variety of reasons, but the creativity and ability to see a new product from start to finish really piqued our interest. Our sponsor, Dr. John Chen, is a professor at Cal Poly SLO and an avid tennis player. On his breaks during tennis sessions, he would rather rest than pick up tennis balls first. Consequently, he has tasked us with making him an autonomous tennis ball collector. Manual tennis ball collectors do not satisfy Dr. Chen's needs, and the few fully autonomous tennis ball collectors that are on the market are too expensive. Covered in this report is a comprehensive overview of our entire process from start to finish. In the background and objectives, we cover the preliminary research conducted, along with scope out the responsibilities of the project. The concept design covers all of the work we did before settling on a final design. This includes ideation and some concept prototyping. In the final design section we go through every aspect of our final design and explain how the 10S bot will function as a whole. Moving on, we detail the manufacturing work done to build the robot. After manufacturing we tested the robot, with limited success, and detailed our struggles with that. Finally in the conclusion we offer some recommendations and reflect on the year as whole!

2.0 Background

The background will detail our experience researching autonomous robots and the tennis community. We hope to gain a better understanding of how to utilize the resources currently available to make a better product for our sponsor. In this updated background section, we have included photos of competitor products that we considered when creating our House of Quality, and future comparisons.

2.1 Summary of customer observation

To gain a better understanding of our customers' desires in our product, we conducted an interview with our sponsor and potential customer, Dr. John Chen. Dr. Chen uses a tennis hopper to collect balls on the court, which involves manually dropping a wire-frame basket onto each individual ball to squeeze it through a grate. This process is not only lengthy and tiresome, but it also takes away from his break time between practice sessions to rest and talk with his son. Dr. Chen desires an easily transportable, autonomous device that can pick up the balls for him in a timely manner. He wants it to hold at least 72 balls and run at least 5 full collection cycles before requiring a recharge.

Additionally, we conducted one-on-one interviews with several experienced tennis players to understand the relative importance of key factors we must consider in our design. From the interviews, players stressed the inconvenience of picking up tennis balls by hand and expressed strong interest in a device that could pick up the balls for them [1]. Manually collecting balls typically takes approximately 5 minutes with the help of a hand-operated collection tool, and the need to recollect balls generally occurs every 20-25 minutes [2]. The average number of balls players want a collection device to hold is approximately 70, which is on par with Dr. Chen's requirement as well [3]. The most common complaint players have with collecting balls by hand is the strain put on either their backs from bending down or on their wrists from gathering many balls onto a tennis racket. This back-pain complaint is most prevalent with older individuals [4]. Finally, many players indicated that they do not personally own any kind of tennis ball collection tool because they are too expensive, citing some products such as the MultiMower, which costs $400 [5].

2.2 Discussion of existing designs

We conducted research on five existing designs and one similar product. Images of these designs can be seen in the following figures. In Section 3.4, we discuss the process by which we compare the five existing designs using a quality function deployment.

Figure 1: Tennibot

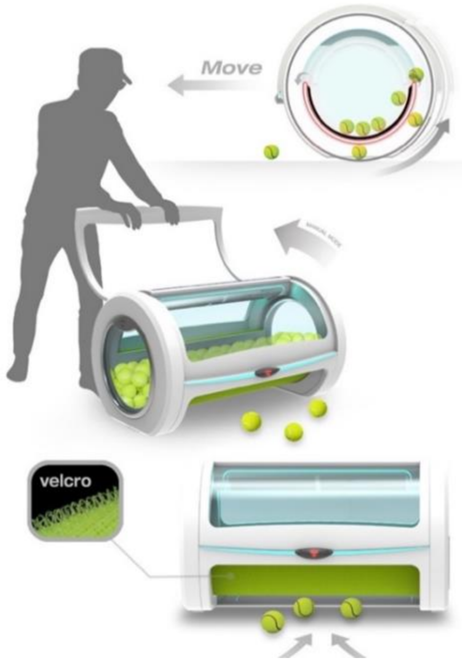


Figure 2: Ball Boy

Figure 3: Tennis Ball Mower

Figure 4: MultiMower

Figure 5: Tomohopper

Figure 6: BallPicker

Tennibot, as seen in Figure 1, is an autonomous, electric powered tennis ball collector [1]. The Ball Boy, as seen in Figure 2, is a concept idea that collects tennis balls using a revolving cylinder wrapped in Velcro [7]. The Tennis Ball Mower, as seen in Figure 3, is a hand operated collector that rolls over each tennis ball and squeezes it into an inner container [8]. The MultiMower, as seen in Figure 4, is a similar concept, but it instead rolls over each ball and scoops them into a basket above [5]. The Tomohopper, as seen in Figure 5, similarly scoops each ball into a basket but also features a sliding mechanism that allows the basket to raise up to serving level [9]. Lastly, the BallPicker, as seen in Figure 6, is another autonomous product, but was designed to collect golf balls on a golf course or driving range; in the scope of this project, it will be used as a reference for autonomous design [10].

2.3 Table of patent search results

One key part of our background research was the investigation of current patents in the field of autonomous tennis ball collectors. There was a variety of patents and concepts available, but we narrowed it down to five of the most applicable patents. A summary of our key findings for each of them can be seen in Table 1. While some patents had similar ideas, such as using suction for collection, they all had differences that expanded our current knowledge on the subject. These patents were found on Google Patent, a resource that lists current and past patents for the general public.

Table 1: Relevant Patents

Patent Number	Patent Name	Key Findings
US 8313396B1	Tennis Ball Vacuum Collector [11]	• Wet-dry vacuum utilized with suction port near ground and funnel to direct balls to port • Able to suck balls up at a rate of 5 balls/second and ability to hold up to 150 balls
US 10676006B2	Dual functional robot and storage bin [12]	• Utilized Ubuntu software library to write code for robot to read • Uses geofencing and sets up boundary to limit the amount of work robot does • Similar to duffel bag on wheels, can be collapsed and easily transported
KR 101873226B1	Device for golf ball retrieving [13]	• Uses suction arm (with suction fan) to collect balls and transfer to large holding tub • Has ultrasonic sensor in front and side to detect balls
US 20060068948A1	Tennis Ball Collection, Dispensing and Transport Apparatus [14]	• Uses sweeping and rolling motion to move tennis balls into collection space • Manual push, use collection vanes to move the ball from court to holding bin
US 20200070015A1	Autonomous golf ball picking system [15]	• Has sensors to determine when collector is full and to detect obstacles on path • Uses predefined path to pick up balls based on computer generated and readable code

2.4 Summary of relevant literature

Lightweight Tennis Ball Pick Up & Hopper (Morales, Degrandis, and Noxon, 2018) is a senior project published by a team at Cal Poly in 2018 [16]. This project sought a manual solution to improve the process of collecting tennis balls from a tennis court. In the report, the authors provide a summary and evaluation of the current mechanisms in use by various competitors to physically pick up a tennis ball from the ground, as well as a set of brainstormed ideas to accomplish the same task. They also demonstrate an integrated basket system that can be removed and replaced with a spare empty basket, allowing the user to conveniently access the balls after they have been collected. This project also serves as a reference point for the cost of one manual solution for picking up the tennis balls.

Tennis Ball Collecting Robot (Sultan, Omar, and Sharabati, 2018) is a research project published by students at Palestine Polytechnic University [17]. The article describes the team's approach to solving the same problem of autonomously collecting tennis balls from a tennis court. This article goes into great depth about different potential mechanisms for picking up the balls, as well as a method for automatically determining an efficient path to collect the balls.

Design and Develop of an Automated Tennis Ball Collector and Launcher Robot for Both Able-Bodied and Wheelchair Tennis Players - Ball Recognition Systems (Wei, 2012) is a report documenting a student team's design process for a similar problem statement at the Universiti Tunku Abdul Rahman [18]. This article provides methods for implementing control algorithms to maintain the robot's path towards the tennis balls based on the balls' location in the frame of the tennis ball's camera. It also provides an outline of a potential system architecture using a single-board computer for path-determination logic driving a microcontroller for robotic control.

Through our research into autonomous tennis ball collectors, a variety of conference papers and articles were found involving different methodologies. One of the largest distinctions between articles was the method for detecting tennis balls and for path planning. According to Cabuk (2018), one advanced methodology was to use image detection and processing to find the balls on the court [19]. With this, there is a camera that scans the area and uses a detection method based on the shape and color to identify tennis balls amongst other objects. For the actual ball collection, there are two types that have been utilized and written about. The first involves using parallel discs with springs to squeeze the tennis balls in between them before depositing them into a holding bin. This method, detailed by Chen (2016), also employs a switch that when triggered would release the balls [20]. The other ball collection idea that was seen a bit more during our research was a long paddlewheel with long blades to sweep the balls into the collection space [21]. In the journal report by Elamvazuthi (2015), using this method yielded a 60% success rate.

Another aspect of our design to be considered is the controller system and how it will interact with the mechanical system. One of the most common components are servo motors, which are small actuators that allow for linear or rotary motion [22]. As detailed by Yeon (2017), the servos can act independently or work together to create motions or power different components. They can interact with software such as Raspberry Pi or Arduino. Most conference papers reference using multiple small motors throughout the autonomous robot to account for camera movement, wheels, and movement in the ball collection component. Another example would be using microcontrollers to control the camera which can identify tennis balls versus other balls and move to collect those, a topic researched by Alwuqayan (2012) [23]. This paper also utilized a claw to collect the balls, which is something to consider.

2.5 List of applicable industry codes, standards, and regulations

According to the United States Tennis Association, a standard tennis court "shall be rectangular, 60 feet long and 21 feet wide for singles and 27 feet wide for doubles play" [24]. In addition, there must be a 10 feet minimum distance from any fixed objects, such as walls or fences. According to the International Tennis Federation, approved tennis balls for play must have a diameter between 2.57 to 2.70 inches and a weight of 1.975 to 2.095 ounces [25]. Both of these regulations are critical in properly designing a mechanism that can consistently collect balls and store the number of balls specified by Dr. Chen.

3.0 Objectives

In this section we will detail the requirements and environment regarding this project, along with a careful analysis of the engineering specifications and how they will be measured. Updated for this report is a better description of the House of Quality, along with more explanation of the boundary diagram and how it will be useful.

3.1 Problem statement

Dr. Chen is an avid tennis player, who after long practice sessions would rather take a break than collect tennis balls. He would like an autonomous device to quickly collect the tennis balls so he can rest, drink water, and talk with his training partner instead of exerting energy to manually pick up the balls himself.

3.2 Boundary diagram

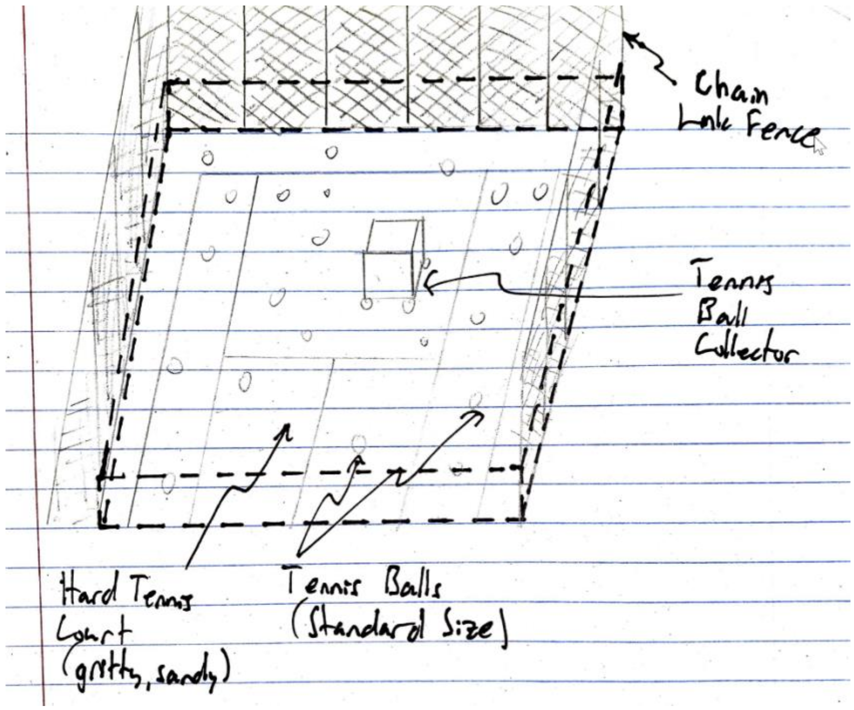


Figure 7: Boundary Diagram Sketch

Figure 7 shows a sketch of the boundary diagram, which outlines the environment the robot will take into consideration when collecting tennis balls. This helps further identify how the robot must interact with the environment. Establishing a boundary diagram gives us a visual representation of what the scope of our work is and allows us to gain a better understanding of the environmental factors that will be at play when the robot is in use.

3.3 Summary of customer wants and needs

In our initial meetings with Dr. Chen, many things were reviewed and considered, but he came up with several key needs for this project. One of his biggest concerns was the size and weight of the tennis collector. He specifically wanted something that could fit into the trunk of a small car and could be lifted and moved around by someone of average height and build. He was also adamant about having the autonomous robot pick the balls up in under 15 minutes, with a preference for an even better time. There were also things that Dr. Chen desired but specified were less critical. For example, having it move autonomously is important, but the methodology behind that is up to our discretion.

3.4 QFD Process Description

The Quality Function Deployment (QFD) is a process used to transform customer wants and needs into quantifiable engineering specifications. Our team used this tool so that we could better evaluate the desires of the customer and compare our solution to the problem statement to existing alternatives. The QFD, which can be found in Appendix A, contains several different sections that serve to translate information starting as the customer voice step-by-step until a quantifiable engineering specification is constructed. In the Who section, the customer is defined. For our project, the primary customers are tennis clubs, tennis coaches, individual tennis players, and our project sponsor, Dr. Chen. The What section is a list of customer needs and wants, which were determined through customer research. The Who vs. What section assigns relative weights to each customer requirement. In the benchmarking section, we evaluate each alternative's performance against each customer requirement. In the How section, we generate a list of engineering specifications designed to capture the customer requirements. In the How vs. What section, we determine relationships between each customer requirement and each engineering specification. Finally, in the How Much section, we set targets for each specification so that we can quantitatively evaluate each solution against the customer requirements. Together, this table provides us with a framework to quantify the performance differences between our own product and the existing alternatives. These differences were examined across a variety of engineering specifications described in Table 2.

Using the QFD we generated, we benchmarked each competitor against the customer needs. Ratings ranged from zero (poor) to five (fantastic). From our analysis, the MultiMower stood out as a clear winner with the highest unweighted score of 38 points. A clear trend emerged that non-automated solutions scored higher in our evaluation across categories that are negatively impacted by the addition of automation hardware like weight, size, and collection cycles per charge. However, the scope of the problem statement is to generate an automated tennis ball collector for customers who value time over factors like weight and size, so we will proceed with a robotic solution.

3.5 Engineering specification table

During the QFD Process, we developed a table of specifications and relevant data about each specification to allow us to evaluate each product against these specifications. This data, found in

Table 2, contains the following information about each specification. In chapter 7.1, these specifications were modified for the design verification plan due to subtle nuances that were not discovered until prototyping. More information can be found in Table 4.

- **Specific Description:** a specific description of the specification.
- **Requirements of Target:** a quantitative value of the specification that serves as a requirement that must be achieved with the given tolerance.
- **Tolerance:** a description of the tolerance of the requirement. Possible values are:
  - **Max:** the product's specification value cannot exceed the requirement value.
  - **Min:** the product's specification value must exceed the requirement value.
  - **Target:** the product's specification value should meet the requirement value plus or minus a specified tolerance.
- **Risk:** how difficult the specification will be to achieve (Low, Medium, High).
- **Compliance:** how the product's compliance with the specification will be determined. Possible values are Test (T), Analysis (A), Inspection (I), and Similarity (S).

Table 2: Engineering Specification

Spec #	Specific Description	Requirements of Target	Tolerance	Risk	Compliance
1	Weight	25 pounds	Max	M	I
2	Time to clear court	15 minutes	Max	H	T
3	Size	10 cubic feet (max length: 3 ft, max width: 5.5 ft, max height: 1.25 ft)	Max	M	I
4	Collection cycles per charge	5 cycles per charge	Min	L	T
5	Ball storage capacity	72 balls	Min	L	T
6	Cost to manufacture	500 dollars	Max	H	A
7	Time to move collected balls to serving basket	30 seconds	Max	L	T
8	Time spent operating device	5 minutes	Max	M	T

3.6 How to measure each specification

- Weight will be measured using some form of a scale. The device should be small enough that a special or larger type of scale should not be necessary.
- The time to clear the court will be measured using a stopwatch. Multiple trials will be conducted with a varying number and spread of tennis balls, and the time it takes to pick up the last ball will be recorded for each trial.
- The size will be measured with measuring tape, and will include the length, width, and height of the device.
- The collection cycles per charge can be recorded at the same time as the "time to clear the court" specification. Trials will be conducted until the device runs out of battery, and the total number of trials will represent the collection cycles per charge.
- The ball storage capacity will simply be measured by placing tennis balls into the collection chamber until it cannot hold any more and recording the amount of tennis balls.
- The cost to manufacture will include all material and labor costs put into building the final design.
- The time to move balls to basket will be measured by recording the time it takes to transfer all the balls in the collection chamber into some form of a readily available ball basket. This basket could either be built into the device itself, or it could be a separate basket that the balls are simply unloaded into.
- The time spent operating device will represent the amount of time a human being has to spend physically preparing or configuring the device per use before it can then operate autonomously. This time will be measured from the moment the device is picked up within the court to be put in place to the moment the person no longer needs to touch or communicate with the device for it to operate as desired.

3.7 High risk specifications

The two high risk specifications we identified for this project are the time to clear the court and the cost to manufacture. The time to clear the court is high risk because it is the specification that is most directly affected by the success and effectiveness of our device. If the device does not work properly, then it will take longer, if not forever, to completely clear the court. The cost to manufacture is also high risk because we do not yet have a full understanding of the components that are necessary to assemble this project. Unforeseen circumstances or events, such as a component being broken and needing to purchase a new one, can also result in a higher cost to manufacture than expected.

4.0 Concept Design

In this section, we detail our extensive concept design process, from initial ideation to our completed final design with sketches, matrices, and CAD models to help visualize our development.

4.1 Concept design process overview

Before we could begin our concept design process, it was necessary to break the project down into more manageable components to allow for easier ideation and concept creation. We did this using functional decomposition, which helps separate the project into subsystems that can be prototyped and analyzed. We came up with four functions after reflecting on our engineering specifications and sponsor needs: generate motion, coordinate movement, transfer balls to storage, and ease of transport. Transfer balls to storage was further broken into two more subsystems, hold tennis balls and pick up tennis balls, which we analyzed separately during our design process. While coordinate movement and generate motion sound similar, they deal with different components of the machine. Coordinate movement is related to how the robot will seek the balls out and the path it takes on the court; generate motion relates to how the machine will move and what components are needed to help it maneuver around the court. The complete functional decomposition can be seen in Figure 8. At the top of this tree is the most general function related to our project: autonomously collect tennis balls. We chose this because it represents an overarching goal that all the sub functions can be related back to.

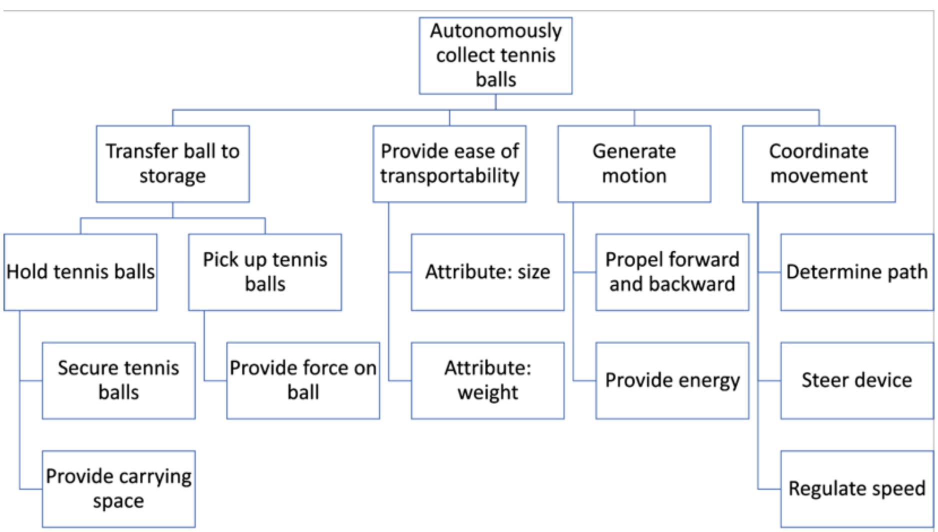


Figure 8: Functional decomposition tree

The concept design process began with a series of brainstorming sessions, known as ideation. We started by learning about various types of brainstorming, and how each one could potentially guide the design of our senior project. We chose three types of brainstorming to engage with: brain dumping, brainwriting, and worst possible idea. Brain dumping was our initial effort at ideation, and it involved taking five to ten minutes to individually come up with as many ideas as possible for individual functions. We then shared all our responses as a team and recorded them for future reference. We followed this with brain writing, which involved sketching ideas for different functions in a short time. Lastly, we tried the worst possible ideas method for ideation, which involved coming up with wacky ideas and then thinking through why they would not work, which allowed us to gain better insight for what could work.

After several brainstorming sessions, the next step in ideation was to start concept prototyping. At this point the ideas were not refined, and the purpose of this modeling was to allow for better visualization and evaluation of ideas, and potentially new ideas to develop as well. Each member of the team completed individual concept modeling on a different function of the overall model. This did make things disjointed and there was some overlap in regard to prototype designs, but this allowed for new ideas and conversations to grow from the experience. The next step in the ideation process was to start refining and selecting our final concept model. To help guide our conversations, each team member created a Pugh matrix for the function we had been prototyping. The point of a Pugh matrix is to allow for ideas to be considered against a datum, or base level idea, and compared for the relevant customer wants/needs. The Pugh matrix was the first in a series of evaluation methods to find a final concept design. The Pugh matrices for each function can be found in Appendix B. Based on the results of our individual Pugh matrices, several concept sketches were created to better visualize our top ideas. This signaled a significant step in the concept design process, allowing us to move forward with final ideas.

4.2 Concept sketches

At the beginning of the ideation process, our group used a variety of brainstorming methods to come up with ideas together. We then moved into an individual phase of work, involving concept prototyping and Pugh matrices. We reconvened and had group discussions to come up with the following sketches. They represent the best ideas to come from the ideation phase and out of our Pugh matrices. Each one has merits that we considered when thinking through our final design concept.

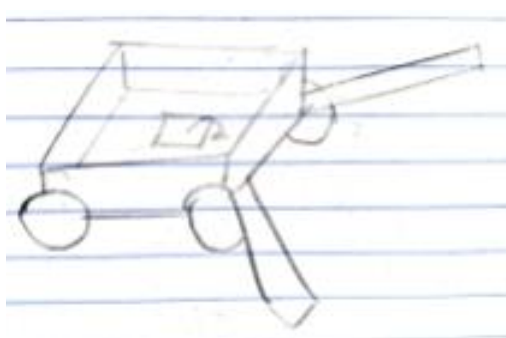


Figure 9: Concept Sketch 1

Idea #1 in Figure 9 visualized the most basic option based on all the ideas presented. It utilized rear wheel drive with basic wheels to generate the motion. It would have long V-shaped arms to corral the balls into the machine, with a trap door to allow for removal of the balls from the machine. It would also follow a simple set path to move around the court. While this idea would satisfy the most basic of our customer needs, it was not creative, nor did it do everything Dr. Chen wanted us to achieve. This design was a good benchmark of how we could improve and what kinds of changes we wanted to make the design stronger.

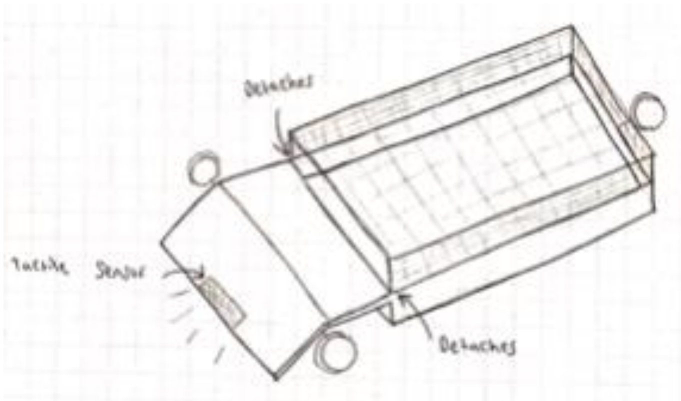


Figure 10: Concept Sketch 2

Idea #2 in Figure 10 involves three wheels, with two in the front of the machine, and a ramp to move the balls into a basket that would be removable. This design also employed tactile sensor to sense the ball or other obstacles on the court. This design excelled in many areas as it is simple, and satisfies many of the design requirements, but in talking through this idea, discovered it would not be as hands-off and autonomous as we strived to provide for our customer.

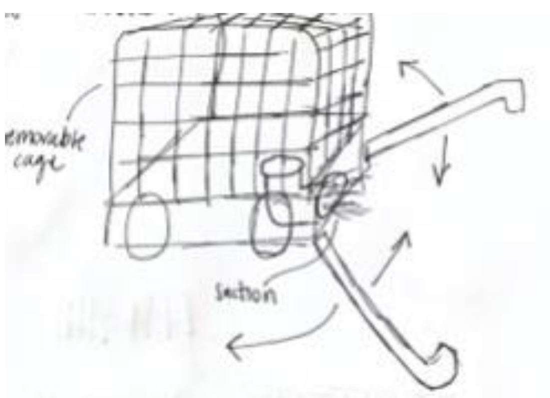


Figure 11: Concept Sketch 3

Idea #3 in Figure 11 was the strongest design we put together, involving all the best ideas from individual Pugh matrices. There are long V-shaped arms with added flexibility and rotation, along with a suction force to move the balls from the court to a detachable basket. The robot would utilize topographical mapping to identify the balls and an ideal route on the court. The wheels on this machine would be mecanum, which allows for 360-degree rotation. While there are still some flaws here, we decided this idea had the most promise moving forward.

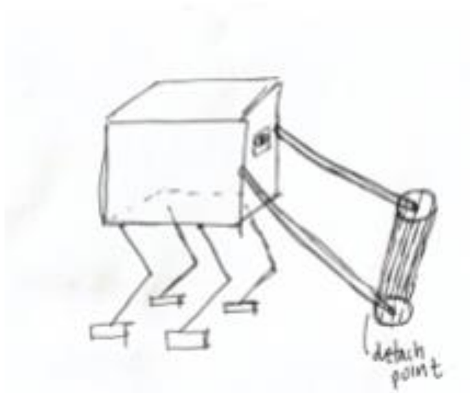


Figure 12: Concept Sketch 4

Idea #4 in Figure 12 was a more ambitious avenue of design, as it involved what we called AT-AT walker legs, which are just advanced robotic legs with a joint for enhanced movement. The robot would be pushing a rolling cage that would collect balls and be detachable. With this idea, we envisioned the robot using LIDAR technology to scan the court and optimize a route to collect the balls.

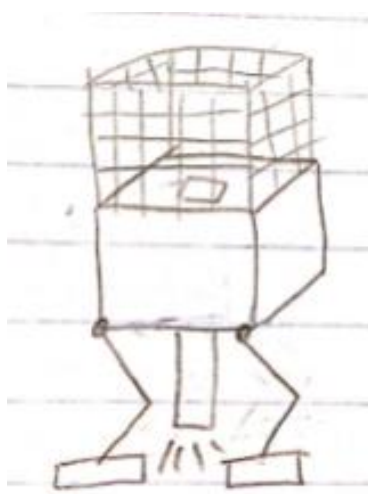


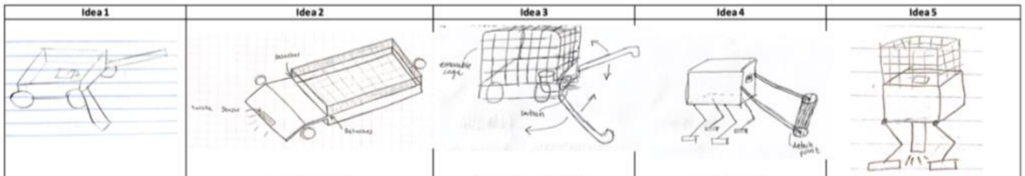
Figure 13: Concept Sketch 5

Idea #5 in Figure 13 was the final one sketched for our analysis and was also the most ambitious in terms of efficiency. It involved the robotic legs once again and used a strong suction force to pick the balls up and move them into a removeable basket that would be resting on top of the machine.

4.3 Concept design decision process

As discussed previously, the initial steps of the design process heavily involved ideation, sketches and considering different concepts for specific functions. More importantly is how to translate these initial components into a final design concept. Within our team, this occurred through a structured selection process. It involved both morphological and decision matrices, along with more concept modeling and analysis to ensure the idea we selected would work. After the creation of four Pugh matrices and the discussion of these within our group, we moved the top 5 ideas from these individual matrices to a morphological matrix to house all the different ideas for each function. This allowed us to visualize different combinations of functions and start creating some rough sketches for what the final concept could look like. This was one of the most important discussions our group had, as it was one of the first times we thought long and hard about how the machine would operate as an entire system and not individual components. From this discussion, we emerged with our top five concepts. These ideas were then sketched out in a clean and readable format to allow for better visualization when we completed our weighted decision matrix.

The weighted decision matrix was the final stop in the decision of our concept design, and can be found in Figure 14, and at the end of Appendix B. It provided us with an analysis of our top concept sketches compared to the engineering specifications we had determined in the QFD. Although this methodology should have provided a clear winner, sometimes there are things that cannot be communicated accurately through simply weighted values and specification.



		Idea 1		Idea 2		Idea 3		Idea 4		Idea 5	
				with Tactile Sensor		with topographical mapping		with LIDAR tech			
Specifications	**Weight**	**Score (1-5)**	**Total**	**Score (1-5)**	**Total**	**Score (1-5)**	**Total**	**Score (1-5)**	**Total**	**Score (1-5)**	**Total**
Lightweight	0.1	4	0.4	4	0.4	3	0.3	3	0.3	3	0.3
Fast	0.1	3	0.3	3	0.3	4	0.4	3	0.3	2	0.2
Hands-off while running	0.14	3	0.42	2	0.28	5	0.7	3	0.42	3	0.42
Easily transported/stored	0.12	3	0.36	4	0.48	3	0.36	4	0.48	3	0.36
Long battery life	0.12	4	0.48	4	0.48	3	0.36	2	0.24	2	0.24
Holds many tennis balls	0.09	3	0.27	3	0.27	4	0.36	2	0.18	4	0.36
Inexpensive	0.07	5	0.35	4	0.28	3	0.21	2	0.14	3	0.21
Durable	0.07	3	0.21	4	0.28	3	0.21	2	0.14	2	0.14
Convenient access to balls	0.08	3	0.24	4	0.32	4	0.32	4	0.32	4	0.32
Minimal initial setup	0.11	3	0.33	4	0.44	3	0.33	2	0.22	3	0.33
Total	1	3.36		3.53		3.55		2.74		2.88	

Figure 14: Weighted Decision Matrix

The results of the weighted decision matrix aligned with our initial thoughts going into the conversation. As a group, we knew that ideas #4 and #5 were ambitious, but they performed the worst out of all the ideas. One of the biggest flaws we saw was that the robotic legs would not move fast enough on the court to collect the balls in an adequate fashion based on our sponsors specifications. Another big issue we saw with both designs was the durability, although it was not a huge concern overall, with a much higher center of gravity, they are much more prone to falling or breaking. One final reason we decided not to move forward with these designs was the cost. They would both require expensive parts to build the robotic legs, along with the technology to coordinate the movement on the court would prove to be too much for our budget. Ideas #1, #2, and #3 all scored higher, with #2 and #3 being particularly close in score. While idea #1 scored better than some, it had several flaws that were confirmed through the decision matrix. While it was a strength on the decision matrix, being too inexpensive could lead to less durable parts and not being as effective during use. It was here we realized the values on the spreadsheet do not always accurately convey all the subtleties involved. As a group we concluded that idea #3 was the strongest, as it excelled in several areas over design #2. Firstly, it had multiple methods of collecting the balls, as the suction and V-shaped arms would double the potential for balls to be collected, whereas the ramp in idea #2 we determined to be ineffective and potentially lead to more issues with the angle needed to ensure balls rolled up it. Along with that, we believe the mecanum wheels to be the strongest option when it comes to figuring out how to move about the space. Thus, even before starting the weighted decision matrix, we had a slight preference for idea #3, but seeing how the weighted decision matrix played out, it proved to be the strongest option.

4.4 Description of selected design

Recalling the functional decomposition, we divided the general goal of autonomously collecting tennis balls into four main subfunctions. These functions were prototyped, considered in the Pugh matrices, and compared in the morphological matrix. All the functions need to work together to produce a fully functional design; however, this was not considered prior to concept prototyping. One of our biggest concerns is how the machine will pick up balls and hold onto them, and after going through several design concepts and iterations on one design, we have decided on a two-part mechanism to pick the balls up. The first component is the V-shaped arms. These will extend from the front of the robot and corral the balls towards the machine. Our goal is to have these arms cover a wide range of motion, allowing for an extended area of collection. One of our sponsor's concerns was having balls stuck in the bottom of the chain-link fence. With an added flexibility of the arm attachments, we hope to use them to dislodge balls from the fence and corral them towards the robot.

The second component of the collect tennis balls function is the rotating cylinders that will project the balls from the ground into the basket. While initially we wanted to use suction force here, it proved too difficult and energy intensive to implement. Instead, we are planning to design rotating cylinders that would be molded to fit the shape of a tennis ball that will propel the tennis balls into the collection basket. To account for how the machine will hold tennis balls, we struggled to find a solution that would hold enough tennis balls, while also being compact and easily transportable. We finally settled on using an off-the-shelf basket and adapting it to the robot. This will allow the basket to be removed from the machine and then used as a stand for serving practice or moved around. This also allows for easier storage and transportation as the basket can be stored separately or moved while on top of the machine.

The generate motion subfunction has two main mechanical components: the wheels and the motors. The robot's motion will be controlled by a single-board computer acting as a leader controlling a microcontroller acting as a follower. The microcontroller will handle synchronous low-level tasks like reading values from sensors, driving the wheels, and controlling other moving hardware involved in the collection of the balls. The computer will run a lightweight operating system that will enable us write high-level code to send signals to move in specific directions to the microcontroller. The computer's ability to run high-level code will allow us to write the software controlling the robot in fewer, more human-readable lines of code. Additionally, the extra computing power will enable us to run CPU-intensive open-source computer vision projects to help the robot formulate the optimal path to collect balls most efficiently. In terms of the software we plan to use to navigate the robot, we need to have a better understanding of the parts we plan to use before choosing a tracking program. This will be implemented early in winter quarter to allow for maximum testing time with the chosen software.

The wheels we selected are a type of omnidirectional wheel called mecanum wheels. These wheels allow the robot to rotate without translating and to translate in any direction regardless of its current orientation. This enhanced mobility over traditional wheels will enable the robot to access the corners and sides of the court more easily, which will save time collecting the balls and decrease the number of balls that are inaccessible to the robot and left behind.

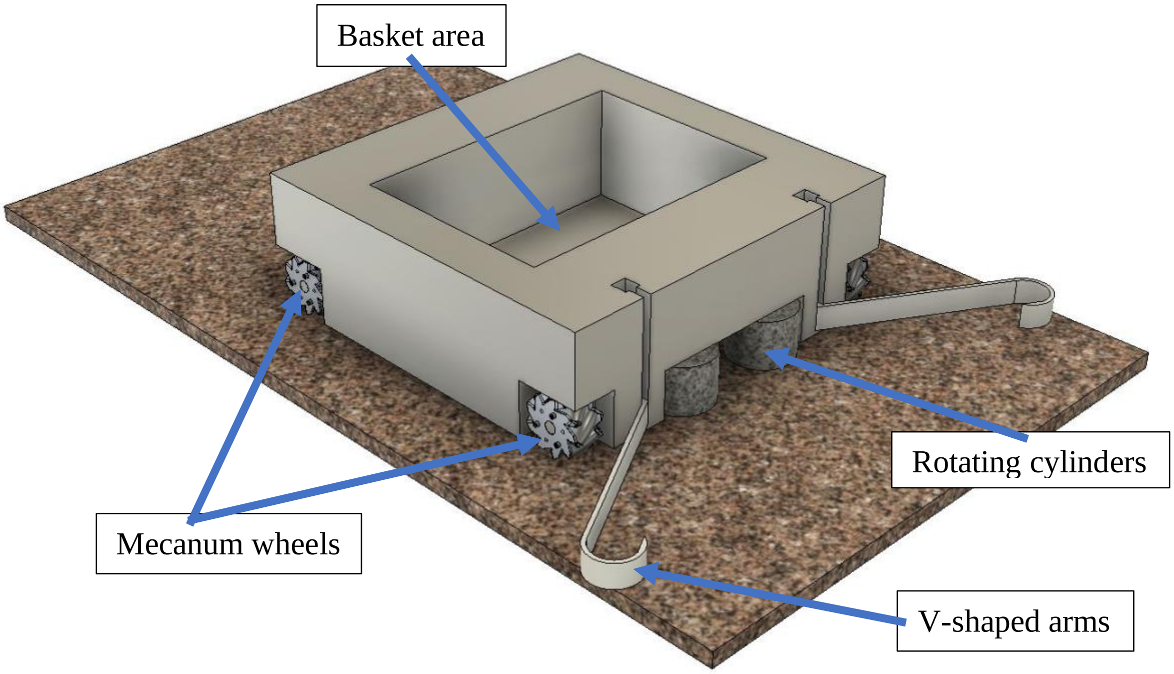


Figure 15: Front View of the Concept CAD

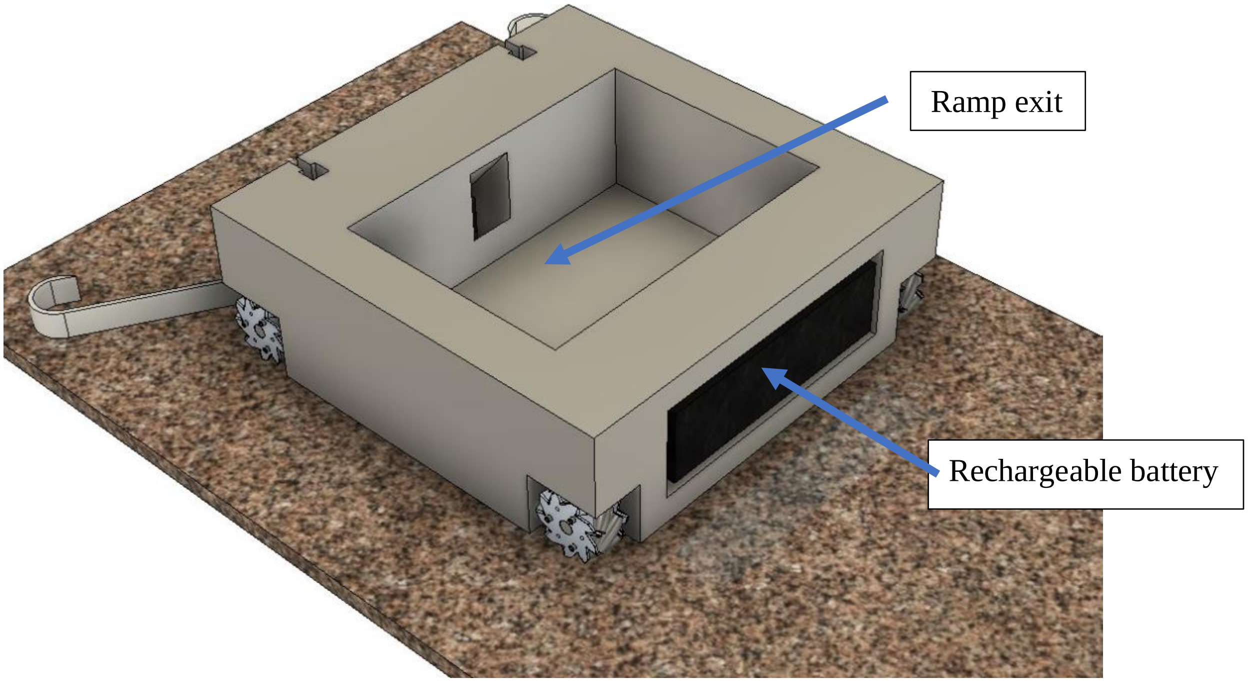


Figure 16: Back View of the Concept CAD

As seen in Figures 15 and 16, the robot will operate close to the ground. The battery and mecanum wheels are both contained within the robot to minimize its dimensions. The V-shaped arms can slide upwards and be detached to save space. These arms will assist in corralling tennis balls towards the rotating cylinders in the center. The curved hooks at each end will prevent balls from rolling away if the robot changes directions. Once the balls reach the cylinders, they will be projected into a basket located within the robot. The entrance to the collection area is designed to be slightly above the base of the basket area to prevent balls from spilling back into the rotating cylinders.

Figure 17: Front Isometric View of Concept Prototype

Figure 18: View of Ramp and Basket of Concept Prototype

Figures 17 and 18 showcase the final concept prototype, with several of the functions modeled to allow for further design and analysis throughout the coming weeks. Production of the concept prototype gave way to several important conversations, such as how the ramp would operate, what sort of force we would use instead of the suction power, and the basket functionality Through these discussions we settled on this final design. As seen in Figure 17, there are orange concave rotating cylinders present inside of the box; these cylinders will propel the ball from surface of the tennis court, up the ramp, and into the collection basket. The long V-shaped arms are able to rotate in this model, although we do not intend to allow for free rotation during customer use. An important aspect that we were unable to prototype successfully was the basket. We plan to use a detachable off-the-shelf tennis basket that also doubles as a serving basket. This will allow the customer greater ease when transporting the machine and ease of access to tennis balls during practice.

4.5 Preliminary analyses & tests

Preliminary analysis was conducted for four specifications: ball storage capacity, size, weight, and time to clear court. The remaining specifications were not analyzed at this moment, as they can only be verified through testing of our future fully functional device. Hand calculations for each specification can be found in Appendix C.

Assuming that the goal is for the device to hold at least 72 tennis balls, the desired information to obtain for the ball storage capacity specification is the volume of 72 tennis balls. From these calculations, the storage container volume must be at least 1,200 in^3. Compared to the maximum size specification of 10 ft^3, or 17,280 in^3, it is safe to assume that reaching the necessary ball storage capacity within the size limitations will not be an issue. A device with the dimensions of 2.5 ft x 4 ft x 1 ft and a separate partially embedded collection container of 2 ft x 3.5 ft x 1 ft should be acceptable within the size and ball storage capacity specifications.

Using these dimensions with aluminum as the base material for the robot's frame and a collection cage similar to that of a typical 72-ball ball picker, the overall weight of the tennis ball collector reaches approximately 21.4 lb. This falls within the 25 lb weight specification, leaving room to adjust key weight sensitive parameters, such as the type and thickness of the material and the size of the motors.

Finally, to address the time to clear the court specification, it is desired to know the required average velocity of the device if it must traverse the entire square area of a tennis court within the 15-minute requirement. Assuming the device covers a wingspan of three feet, the device must travel at an average velocity of approximately 2.67 ft/s, or 1.82 mph. This is comparable to the speed of the Tennibot, which can reach up to 1.4 mph.

4.6 Discussion of risks, challenges, and unknowns

Moving forward with the project, as a group we have reviewed the challenges and risks associated with the project, and feel we are in a good place. There are several specifications that we know will be harder to achieve. These being the time limit to collect the balls, the cycles per charge, and potentially the hands-off while running. Because we can identify these early on, we can come up with a best plan for success, along with a backup plan in case of failure. With the case of the time limit, during our initial testing with the prototype we know of some modifications that can be made to increase the speed. With the cycle per charge, this will heavily depend on the battery we decide on, along with the energy required by the motors and electronics in our system. We can research and test different batteries before settling on the strongest option. While hands-off while running was one of the biggest requests from our sponsor, it will be harder to confirm this specification until we are in the testing phase. Because of this, we will ensure the research presents strong results before moving forward to guarantee success.

A hazard that was discussed with our project sponsor was having the robot run into people while collecting balls. While this is a serious issue, we all agreed it was not something that will heavily impact our design process. The robot should be able to detect people, and we can include an emergency stopping mechanism to ensure that it does not run into anyone. We also do not anticipate the robot moving fast enough to not allow for people to move out of the way if they see the robot on the court. The complete design hazard checklist can be found in Appendix D. A completed risk assessment, as it relates to the final design, can be found in Appendix E. We had some insightful discussions with our project coach about the material used for the structure of the mechanism. This is something no one in the group was familiar with and will need to be heavily researched moving forward. By recognizing this, we can put in adequate research into finding the best solution.

5.0 Final Design

In this section, we will walk through our final design. This will include a comprehensive overview of how we expect the design to function. It will also detail how the mechanical and electrical subcomponents will work together. We are also providing the design justification we have come up with thus far. Also considered are the safety concerns regarding robot-user interactions and how to mitigate those, along with an overview of the final budget. Finally, we detail the structural prototype design and the testing results.

5.1 Description of final structural design

Figure 19: Aluminum 8020 T-slot

Based on feedback we received from our sponsor, and the general direction our project was heading, we split into two sub teams. One was tasked with the mechanical components and the other was dedicated to the electronics. From this emerged a strong, cohesive design that was able to provide expansive technical components, along with the structure necessary for a long-lasting robot. The frame is made up of structural framing aluminum 8020 T-slots. A sample piece of the T-slot can be seen in Figure 19. This was a conscious design decision as it is quite versatile and will make assembly simple. One benefit of the T-slots is that they allow for easy connection of other parts to the frame. Additionally, the T-slots are extremely strong and lightweight; they will easily be able to support the weight of the tennis balls. There are a variety of options when attaching T-slots, but we went with the standard end fasteners, as they are easy to work with and fit our needs. The frame is made up of 14 T-slot pieces, with additional supports across the base and where the rotating cylinders will go. Two rubber wheels in the front are attached to motors that drive the robot. These motors are attached to the T-slots through motor mounts, which are secured with screws. The motors are connected to motor controllers, which are driven by an Arduino Nano. A battery is also attached to the bottom of the frame, which is removable and rechargeable, and secured using fasteners and a panel for support. The Omni wheels in the back are secured by an axel with bearings to allow for free rotation. Omni wheels can translate laterally, which will allow the robot to steer about its front. These will allow for a wider range of motion and make movement across a tennis court faster and easier. This encompasses our generate motion functionality. A labeled CAD model of our final design, with the components mentioned previously, can be seen in Figure 20.

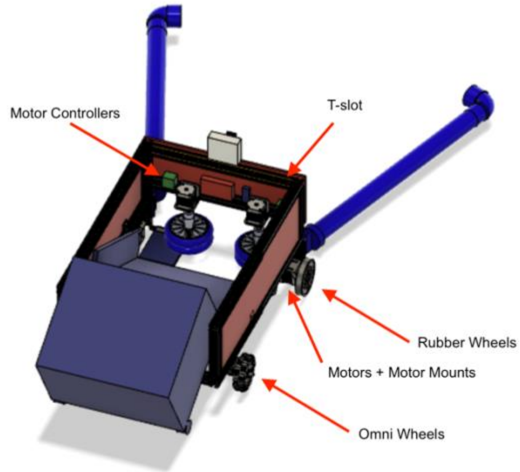


Figure 20: Isometric view of CAD model with labels for the generate motion function

In terms of the collection of tennis balls, the system is twofold. External to the robot are a set of arms made from PVC pipe and fixed to the frame with a 3-D printed part, and then secured in place with a spring-loaded pin. This allows for the arms to be detached by simply pulling the pin off the 3D printed part and then removing the arms from the frame, as picture in Figure 21.

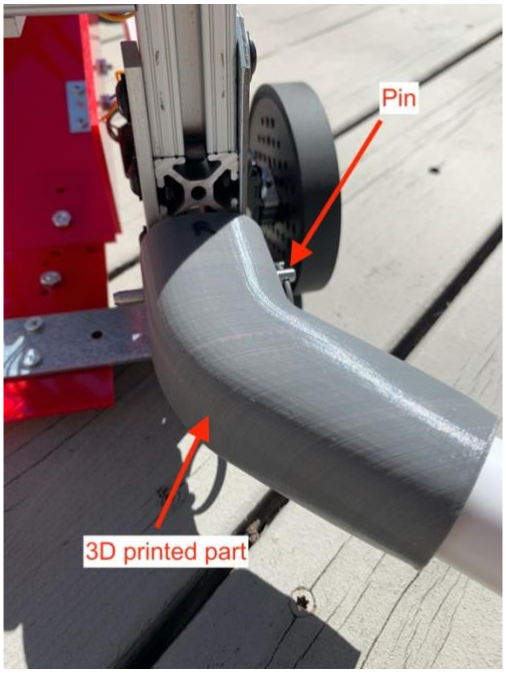


Figure 21: PVC Arms

At the end of these arms is a PVC elbow functioning as a hook to make sure the balls cannot roll away from the arms. As seen in Figure 22, the PVC arms will corral the balls towards the rotating cylinders, providing momentum and structure for the tennis ball movement. At the front of the robot is a set of rotating rubber wheels. These are attached to motors with a coupling and a metal shaft. When a ball approaches them, they quickly suck in the tennis balls and shoot them up a ramp into a basket using the force generated by the motor. We tested using 3-D printed cylinders and 5V motors, but this did not provide enough power. Thus, we found 125mm rubber wheels that can be stacked on top of each other, with a shaft to support them. This, along with a stronger motor will provide the necessary power to move the tennis balls.

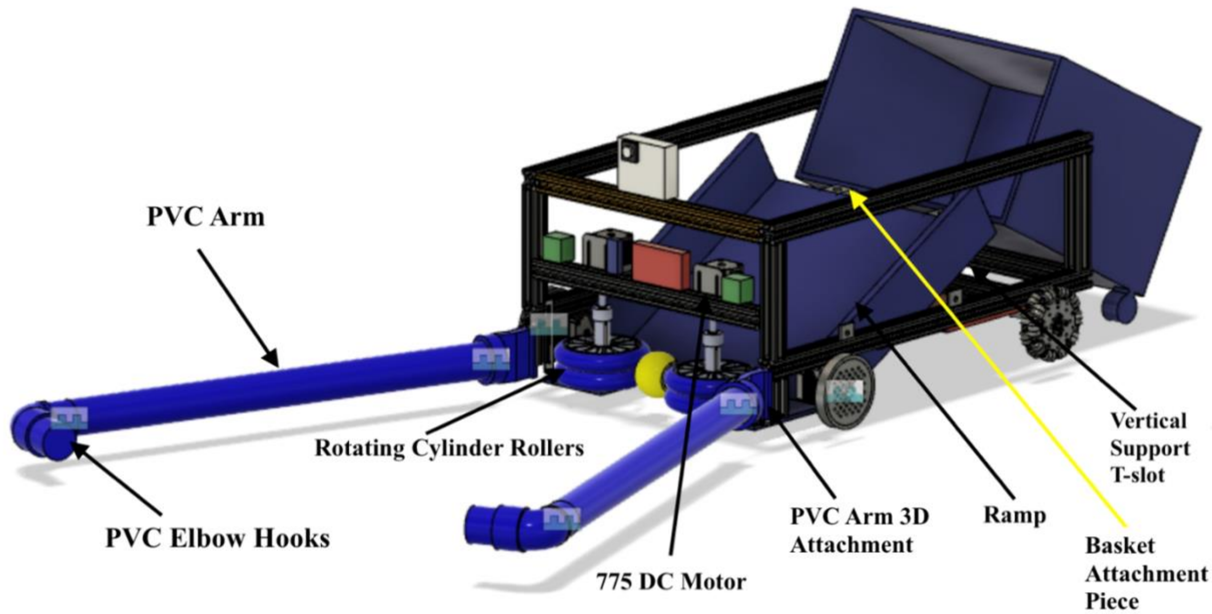


Figure 22: Isometric View of CAD Model with Labels for the Ball Collection Subsystem

The basket is connected to the robot with two 3-D printed brackets. These will sit on top of two vertical pieces of T-slots towards the back of the robot. The brackets will have space for s-hooks and will be designed based on the precise angle needed for the ramp. The intended design is to allow for the basket to be removed when it fills up with balls and then transform into a serving basket, making it multipurpose. The balls move up a ramp made of an acrylic panel, with walls to prevent the tennis balls from flying off the ramp. The basket is situated at an angle complementary to the ramp so that the balls simply move far enough up the ramp, and they will then be deposited into the basket. Having the basket at an angle provided additional support for the tennis balls and takes some of the weight from the robot off the motor shafts. The ramp will also be connected to the 3-D printed bracket at the top, along with latches so that the battery can be easily accessed. At the other end of the ramp, there are two specialized brackets that will be secured to the side of the frame, and then hold the ramp and rotating cylinders in place. This can be seen in Figure 23.

Figure 2323: 3-D printed basket attachment piece and ramp connect piece (not visible) highlighted in final build picture.

As we were developing our final design, we carefully referenced a failure modes and effect analysis worksheet that guided our work. When filling this out we explored potential failure modes for each functionality of the subsystems. From here we were able to come up with some current preventative activities and how we will be able to detect them. From this analysis, we came up with a better method to mount the ramp to the rest of the robot and ensure that the rotating cylinders remain stable. A complete data sheet for the failure modes and effects analysis we did can be seen in Appendix F.

5.2 Functionality of design

The overarching goal of our senior project is to build a robot that can move around a tennis court and move balls from a spot on the court, into a basket. Walking through the functionality, we will use an image detection software to view the court and identify the tennis balls present. Then the robot will adjust its position and move to the correct location to pick up the next closest tennis balls. As the robot is moving, two long arms will be sticking out of the sides and be corralling the balls towards the center of the robot. Because of the movement of the tennis balls, and the power of the rotating cylinders, the tennis balls will be pulled into the moving cylinders and shot up a ramp into a basket. The ramp will be designed and then modified so that the tennis balls do not have the ability to escape as they are moving towards the basket. The basket and ramp will be removable so our sponsor can have easy access to the basket for other purposes, and so the battery can be reached as well.

5.3 Electrical and software final design

The electrical subsystem consists of two parts: a computer vision and processing portion and a motor controlling portion. The computer vision part is comprised of a Raspberry Pi 3B+, a Pi Camera V2 Module, and a 9V power source connected to a 9V to 5V step down module. The motor control system consists of an Arduino Nano that is powered by the Pi via a USB to mini-B cable, four L288 motor controllers, two stepper motors, two 12V DC motors, and a 12V NiMH battery to power the motors. Each motor controller has four input logic pins that need to be connected to the Nano. This means 8 of the 22 pins on the Nano will be taken up by the motor controllers. One dual H-bridge is required for each two-coil motor. An H-bridge is a connection of diodes that allows the motor’s coils to induce alternating magnetic forces on the rotor. One motor should be connected to one H-bridge, and the other motor to the other H-bridge. Lastly, the motor controllers take 12V and ground inputs from the NiMH battery in order to power the motors. For specified wire routes, refer to Appendix G.

To navigate the robot to the tennis balls, the Raspberry Pi will run custom software to detect tennis balls seen by the camera, determine an instantaneous optimal path to collect all the balls seen by the camera, and send commands to the Arduino to follow such a path. Figure 24 shows an overview of the software control flow. In Appendix H, all of the completed code implemented in this project can be found.

To perform the first step – ball detection – the Raspberry Pi will run custom software that utilizes the open-source computer vision library OpenCV. This library provides a built-in function that performs a mathematical operation called the Hough Circle Transform to detect imperfect circles in an image and return coordinates and radii of detected circles. This transform serves as the basis for our ball-detection algorithm. In order to improve efficiency and accuracy, the program first preprocesses the image before performing the Hough Circle Transform. This preprocessing includes downscaling the image from HD resolution to a less performance-intensive resolution, then applying either an edge detection method called Canny edge detection or transforming the image to HSV colorspace and applying a threshold mask to filter out all colors outside of a range aimed to capture just the tennis balls. Both the Canny edge detection and HSV thresholding methods work well beyond the required framerate on a desktop computer. Further testing with the reduced processing power of the Raspberry Pi will likely expose performance differences between the two preprocessing methods and lead us to a final selection.

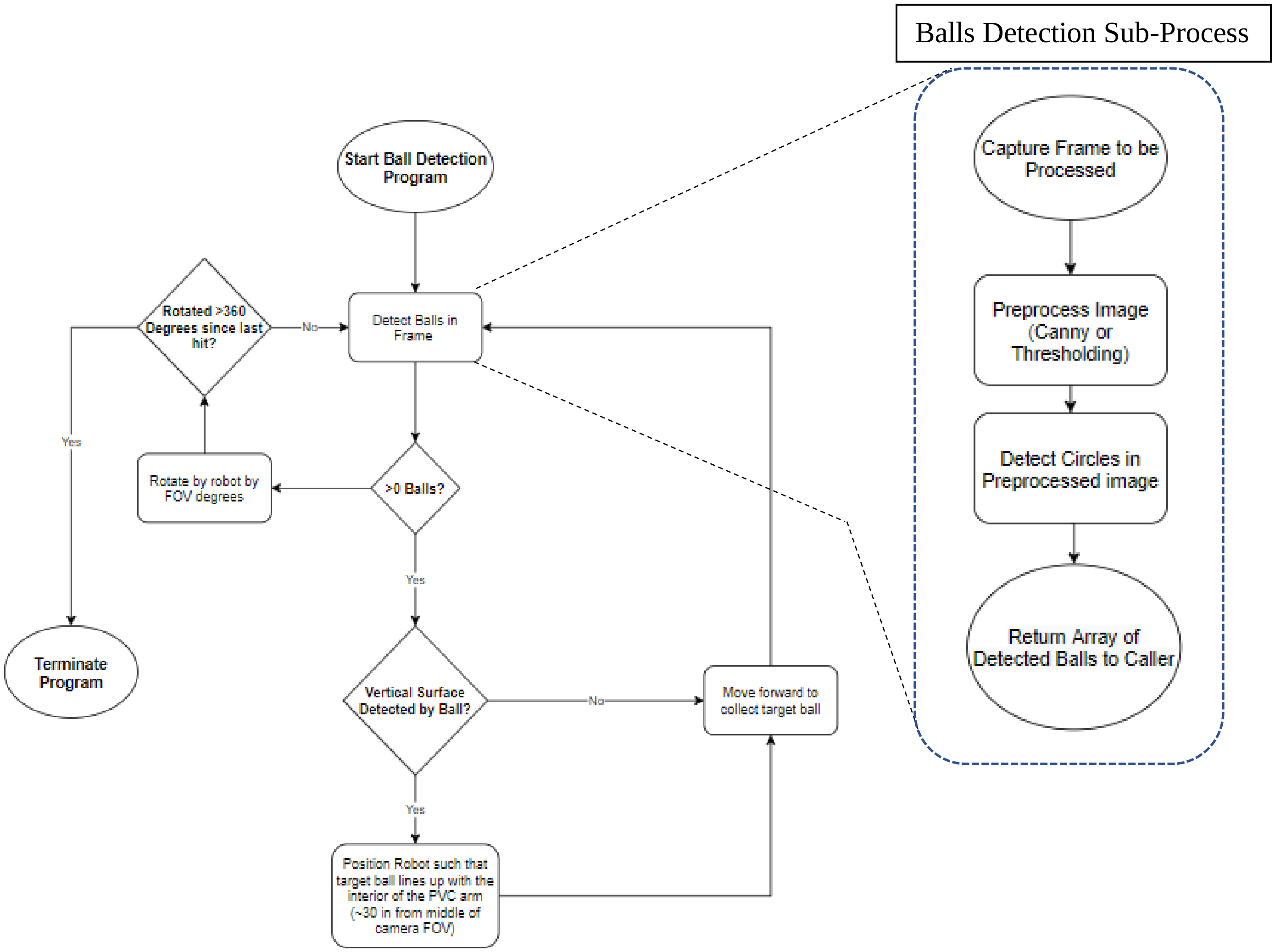


Figure 24: Top-Level Image Detection and Control Software Flowchart

The second step – determining the instantaneous path of the robot – uses the coordinates and radii of the detected circles produced by the Hough Circle Transform to generate a path for the robot to follow. The current pathfinding algorithm is a greedy algorithm that seeks to optimize the entire path by optimizing each step along the path. Although this greedy algorithm does not produce the absolute optimal path, the process of producing such an optimal path closely parallels the Traveling Salesperson Problem, a heavily studied problem in graph theory for which no known solution that works in $O(1.9999^n)$ time exists. Due to the limited computational power of the Raspberry Pi, the greedy algorithm is the sensible choice. At every frame in the video, the algorithm selects the closest ball by detecting the ball that is lowest in the frame.

The final step in navigating the robot is to send the commands to the Arduino to rotate and move the robot. The Arduino will support commands to rotate in either direction as well as to move forwards and backwards. After the software has selected a tennis ball to navigate to, it commands the Arduino to begin rotation. The camera system continues to monitor the position of the tennis ball in the frame until it is in the middle of the frame horizontally. Then, the Raspberry Pi commands the Arduino to stop rotation and begin driving forwards. The software determines that the ball has been collected once it is outside of the bottom of the frame. Then, it starts over with step 2 to identify the new closest ball.

To allow for a user-controlled application, we chose to use the Blue Dot API. The Blue Dot API uses a client/server model. By downloading the API on the Pi 3B+ or any Bluetooth Enabled Raspberry Pi, the API can set up a communication server on the Pi. Then, the user must download the Blue Dot application which can be found for free on the Google Play Store on any Bluetooth Enabled Android device. Once the app is installed, is acts as a client that the already running server can recognize. The user simply taps on the name of the server through the app and the connection is established. The app by default starts with a large blue button that will execute whatever function is specified in the server-side code. However, the app is extremely customizable through the server-side code as well. The documentation for setup, customization, and use can be found online at bluedot.readthedocs.io.

5.4 Design justification

To meet the design specifications, careful consideration has been placed into the material selection and geometry of the design. For the construction of the chassis and frame, aluminum 8020 1.0 in x 1.0 in T-slots have been selected as the material. These T-slots boast an incredibly high yield strength of 35,000 psi for a just a fraction of the weight, weighing a measly 0.0424 pounds/inch of length. In total, approximately 174 inches of T-slot is required for the robot, equating to approximately 7.38 pounds or 29.4% of the weight specification. Acrylic plates have been selected as the frame's panels as they are lightweight, sturdy, and most importantly, are nonconductive. All of the electrical components, such as the motor controllers, Raspberry Pi, and Arduino Nano will be stored between these panels and a protective 3D printed casing. Lastly, the general geometry of the robot was constrained with a specific collection basket so as to minimize weight and allow for ease of replacement.

Mecanum wheels were initially selected during the concept prototype, however they were replaced with a pair of Omni wheels and a pair of rubber wheels in the final design. This was done to reduce the manufacturing cost, since mecanum wheels are much more costly. Rubber coated wheels were selected rather than plastic wheels due to their higher coefficient of friction required to operate on the surface of the tennis court.

A preliminary torque analysis was conducted to select the two 230 mN-m motors used to drive the wheels. The torque analysis, done in Appendix I, assumes a conservative weight of 45 pounds evenly split between the four wheels of the robot. To generate a moment to cause the robot to turn either left or right, 78 mN-m was required. However, we chose to size up to increase the overall speed of the robot, as motor velocity and torque are inversely related.

The Raspberry Pi 3B+ and Arduino Nano were respectively selected as the robot's single-board-computer and microcontroller. The Raspberry Pi boasts a powerful 1.4 GHz quad-core processor, Bluetooth 4.2, a camera port, and 40 GPIO pins. These are all the components needed to efficiently process the machine vision and remotely command the robot in a single low-cost package. Although the Raspberry Pi has the capability to drive the wheels' motors itself, we have elected to move the motor driving software to the microcontroller instead; this will free up precious processing power for the machine vision algorithm. Selecting the Arduino Nano as the microcontroller was an easy choice because of its tiny size, cheap cost, and extensive library documentation.

To power all of the electrical components, a rechargeable 3,000 mAh NiMH battery was selected. A preliminary analysis done in Appendix I estimates that 3,150 mAhs are required to power the components for 75 minutes. This duration of time was estimated from the five collection cycles per charge specification, with the assumption that the motors are constantly running at maximum speed and each cycle taking 15 minutes. Although it may seem that a larger battery capacity is needed, the estimate assumes the maximum time and current draw. Realistically, the robot will not be running at max speed the entire duration of each collection cycle. Efforts will be implemented to conserve energy, such as optimizing the machine vision to reduce the collection time and placing the robot on standby while not actively collecting balls.

5.5 Discussion of Safety, Maintenance and Repair considerations

While the robot may not necessarily be in motion while people are playing tennis, it may be in motion while people are present on the tennis court. As such, it is important that the robot does not present any immediate safety hazards, especially to children.

There are a couple of notable mechanical safety concerns that arise when the robot is in use. First, the robot operates at a low height, making it a tripping hazard in the event that it crosses paths with a person. However, its maximum speed will be less than two mph, so the likelihood of a person accidentally being hit or run over by the robot is very slim. Additionally, it is not very heavy, so there is not a crushing hazard if the robot does run over somebody's feet.

User safety was an important consideration for the electrical subsystem. The processors chosen both run at a 5V input which means they pose no threat of electrocution. The motors were chosen to have 12V input. The motors are high-torque, low-RPM stepper motors so it is extremely unlikely that they could cause any serious, bodily damage. Additionally, the Raspberry Pi, Arduino Nano, and L298 motor controllers will be connected to each other in a plastic box such that any electrical junctions cannot be accessed without first opening the plastic casing. The 12V NiMH battery, however, does pose some potential risk. This battery should not be exposed to any liquids and its charging instructions should be followed carefully to prevent a potential fire hazard. The battery has a large warning label on its body in order to inform the user of these instructions. Additionally, the charger comes with an instruction manual that details the charging procedure for this specific battery. This manual will be explicitly provided to the user of this product. The battery must be disconnected from the system to be charged. To increase safety, the battery and system are connected via a PP45 to PP15 connector set. The battery is housed and mounted under the ball ramp, as shown in Figure 25 so that it may be further protected from any outside factors. Any cable runs, such as between the motor and logic board housing or between the motor drivers and motors, will be routed through the aluminum T-slots to minimize human contact.

These safety choices were made alongside some corresponding maintenance decisions. In order to keep the robot running, the battery must be easily accessible and removable. This inevitably makes the battery the most accessed component of the electrical subsystem. Having the battery mounted underneath the ball ramp allows the user to easily access it by removing a section of the bottom panel. In the case that the battery runs its lifetime, it will be easy to order a new battery in its place to power the robot.

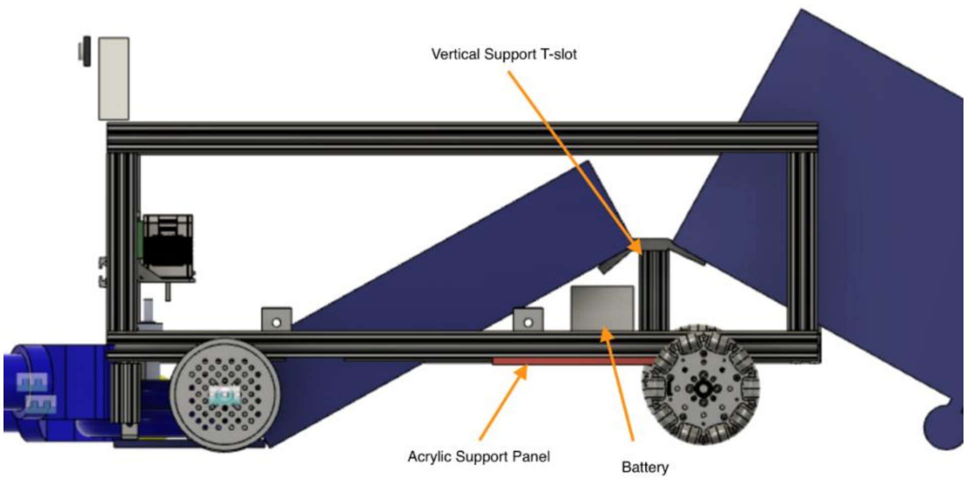


Figure 25: Side view of robot with labels for the battery, support panel, and vertical T-slot.

The next most accessed components, according to the team's assessments, will be the logic boards. These include the Raspberry Pi, Arduino Nano, and L298 motor controllers. It will be important for the team, especially during testing, to access these chipsets to implement quick hotfixes and tweaks to the control system. It will be less important for the final user to access these boards. The boards are mounted to the front of the chassis as shown in Figure 26**.** They are housed in a plastic casing, as described above, such that important inputs, like USB ports and HDMI ports, and outputs will be accessible on the side of the casing. The logic housing is mounted to the aluminum 80/20 t-slots such that some common tools are required to unmount it. The removable lid of the logic housing is held to the body via traditional Philips's head fasteners so that it can be easily opened if necessary. In case someone needs to access the electronics in the housing, all electrical connections to logic boards are properly insulated to protect the board from potential static discharge. The boards are fastened to the housing with standard fasteners through the fastening holes on-board. The Arduino Nano is set into female pin connectors that are soldered to a circuit board that is fastened to the housing. If it needs to be replaced, a new Nano can simply be plugged in.

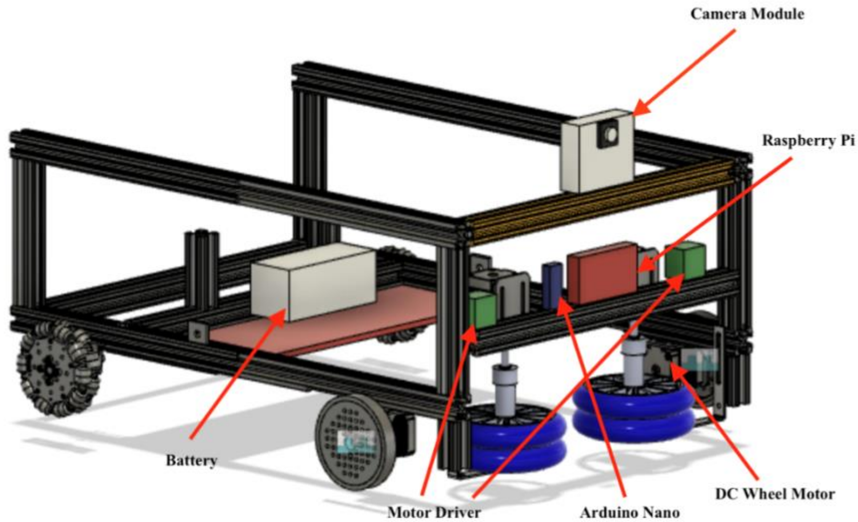


Figure 26: Labeled Electronic Layout on Chassis

The least accessed component will be the motors. Therefore, the team did not find it necessary to make the motors and motor mounts particularly accessible. To replace one of the DC motors driving the ball collection cylinders, some of the acrylic side panels will have to be removed. The stepper motors will be much easier to replace as they are mounted to the underside of the chassis. The front acrylic panel will have to be removed for either motor type replacement because the electrical connections will have to be updated as well.

5.6 Summary cost analysis

The projected costs of all the major components are summarized in Table 3. The components are organized by their relation to either the mechanical or the electrical subsystems. These costs do not take into consideration minor purchases that may be necessary, such as glue and electrical tape. The final total cost is $941.40, and that value can be found in more detail in Appendix J. Also in that appendix is an overall hierarchy of components with their part numbers along with details about each purchased part.

Table 3: Preliminary Subsystem Cost Analysis

Subsystem	Component	Cost
Mechanical	T-slots	$132.48
	Panels	$ 53.15
	Fasteners/Brackets	$ 57.50
	Spinning Cylinders	$ 17.14
	Basket	$ 43.00
	PVC Arms	$ 10.00
	Subsystem Total:	**$341.95**
Electrical	12V Motors	$ 69.80
	Machine Vision	$ 91.69
	Wheels	$ 77.38
	Battery	$ 88.74
	Subsystem Total:	**$327.61**
	Overall Total:	**$669.56**

The mechanical components comprise the chassis and the ball collection subsystem. The 1” x 1” T-slots, which are purchased from 8020.net, vary from 3 inches to 24 inches in length and require various access holes to be drilled to allow for connection with other T-slots. These T-slots come at a cost of $0.23 per inch, with an additional $1.95 per cut and $1.95 per additional drilled hole. The acrylic panels are $3.70 per square foot with an additional $10.50 per cut. The 10 Series standard end fasteners for the T-slots are $1.25 per fastener, and the 10 Series panel-mount brackets for the acrylic panels are $3.50 each. There are four wheels required for the spinning cylinders component at a cost of $8.57 each. The basket, which is a Ballport Tennis Ball Pickup with wheels, will be purchased directly from Walmart and costs $43.00. Finally, the PVC arms can be purchased from Home Depot and will cost approximately $10.

The electrical components are comprised of four main categories, the motors, machine vision, wheels, and battery. The cost of each category can be seen in Table 3. The motors category consists of two 12V DC motors and two 12V stepper motors, respectively $19.82 and $15.64. In the machine vision category is the Raspberry Pi 3B+, Arduino Nano, and the Camera Module, roughly $30 each. Four wheels in total are required, two Omni wheels and two rubber wheels. Omni wheels cost $30.57 each due to requiring many smaller parts to construct a single wheel, while rubber wheels are much cheaper, only $8.12 each. Though the battery category is costly, it is a rechargeable 12V 3,000 mAh battery, accompanied by a charger, that is expected to last for the duration of the design specification.

5.7 Structural prototype analysis

For the structural prototype analysis, each sub team created a separate model to test the validity of different things. The mechanical sub team created a wooden frame and 3D printed cylinder to test the efficiency of moving tennis balls through the cylinders. The electrical sub team started by attaching their motors and wheels to a piece of plywood to work on the configuration of components and initial tests with the motors.

While the idea of two rotating cylinders was not a new one, to ensure that it would function as desired, it was critical to create a structural prototype to see how this design would fare within the scope of the project. To do this we decided to focus on the rotating cylinders themselves as it would be critical that they are able to pull the tennis balls in with enough force. We started by building a frame for the cylinders to be tested within out of scrap wood. We also procured some small 5V DC motors and adequate batteries. For the rotating cylinders we chose to the 3D print them as it was both cheap and easy to manufacture. At the time we had not considered sourcing out wheels to use instead and were hoping this test would reveal the validity of using PLA as the material. The frame and rotating cylinders can be seen in Figure 27.

Figure 27: Disassembled Rotating Cylinder Structural Prototype

In the end it was critical that we spent the time going through this structural prototyping process as during the testing we realized that the PLA cylinders would not be the right material or shape to accurately move the tennis balls. We also discovered that there would need to be a lot more power coming from the motors. It was after this testing that we did some more research into different ways to propel tennis balls. Something that we had not considered earlier but proved to be fruitful in our later searches was the idea of a pitching machine and how those wheels are designed. Finally, from this structural prototype we learned that the current design for the rod we had secured through the cylinders would need to be secured at the bottom of the frame. When we were testing and did not have it secured, the bottom of the cylinders was flying around. In Figure 28, a tennis ball can be seen moving through the two rotating cylinders. While this looks promising, it was not the cylinders moving the tennis balls and just the slant of the surface that propelled them through the opening. Another issue with this prototype is the concavity of the cylinders was too severe and it did not match up with the tennis ball shape correctly. This was another reason that the structural prototype did not function as desired.

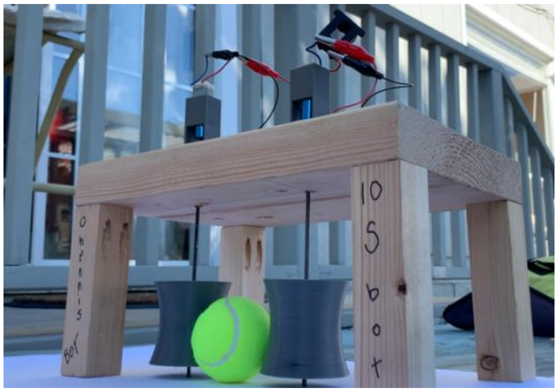


Figure 28: Rotating Cylinder Structural Prototype

On the electrical subteam, we created a structural prototype out of plywood to test the strength of the motors. Due to not yet receiving a large majority of the components, temporary substitutions had to be made, such as wooden dowels for the shaft and a large 12V DC power supply. The power supply was not very optimal as it constrained the movement to a length of the cord attached to the wall, but the weight of it was useful in simulating the weight of the robot. From this structural prototype, we learned that the relationship between the speed of the motors and its torque are inversely proportional. The proportionality is not linear, and the torque quickly drops off as the speed increases. Additionally, we learned that the wheels must be as perpendicular as possible to the ground, otherwise the motors have immense difficultly driving the prototype. An image of our electrical prototype can be seen in Figure 29.

Figure 29: Electrical Structural Prototype

6.0 Manufacturing

Due to the circumstances under which this project was completed, most of the parts needed to assemble the robot were purchased off-the-shelf. Parts that could not be found through a retailer, such as the attachment piece for the V-shaped arms, were 3-D printed and/or modified through simple manufacturing methods. The following sections describe how these parts were procured and the steps for assembling them. For in-depth part drawings see Appendix K.

6.1 Part Procurement & Final Budget Status

The part procurement process was originally going to be spearheaded by 80/20 Inc since we ordered all of the frame components and many of the other necessary mechanical parts from them. Unfortunately, they had severe complications at their warehouse due to COVID-19 delays and were unable to manufacture our parts in a reasonable timeline. We only discovered this during the second week of the quarter when our components were meant to arrive, which then required us to drastically shift our plans. We instead ordered the raw materials from McMaster-Carr and started the manufacturing process ourselves. The reordered parts arrived almost immediately and allowed for us to start the manufacturing process relatively quickly. All of our parts were shipped to the Mustang 60 machine shop and we were able to drop in and pick them up. In terms of the other components, all the electrical parts arrived as ordered and we had no trouble passing the parts off to the correct members of the team to start building and initial testing. Additionally, several of the smaller components were 3D printed in-house using PLA that we already had. This alleviated some of the issues we faced with getting the correct sizing of these components as some had a tight tolerance. Thus, it was not a big issue to reprint parts as needed.

In terms of our budget, we were given a very generous amount from our sponsor, Dr. Chen, and were able to purchase everything without any issues. The largest expense then came from the raw T-slots and large acrylic panels we ordered. We spent a grand total of $941.40 which is well under our budget allocation of $3,000. We could have potentially alleviated some of the costs since several purchased components were not used in the final build of the project. At the time of ordering, we deemed them necessary, and it was important to our team that we were able to build the robot without having to wait to repurchase parts frequently.

6.2 Manufacturing Processes

The manufacturing process primarily consisted of cutting parts to length, drilling holes, and sanding parts down to slightly decrease their size. This section provides more in-depth descriptions of all of the manufacturing steps that are necessary for constructing each sub-assembly of the robot. A summary table of these manufacturing steps can be found in Appendix L.

6.2.1 Electrical Housing

The electrical components housing required manufacturing. A 3-D printed housing was created to protect the electrical components, to hold the 12V battery in place, and to mount the camera module to the 8020. These components can be seen in Figures 30, 31, and 32.

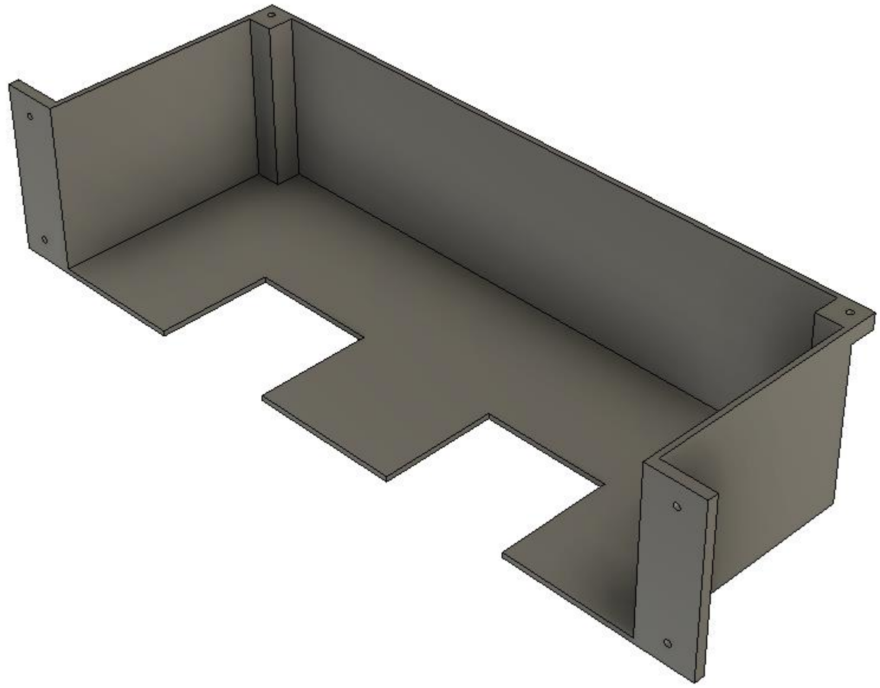


Figure 30: Electrical Components Covering Panel

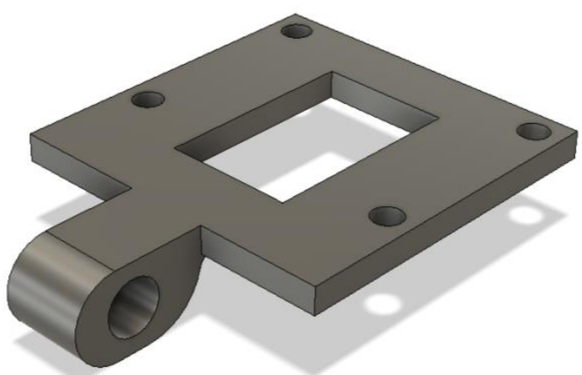


Figure 31: Raspberry Pi Camera Mount

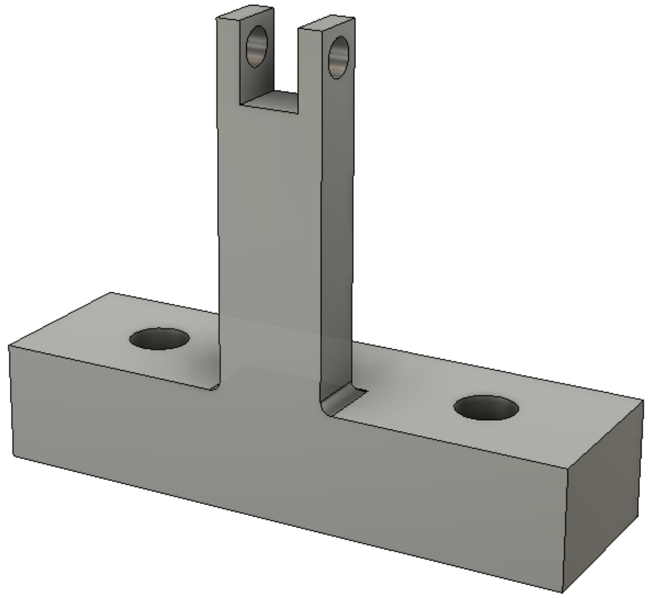


Figure 32: 8020 Camera Mount

6.2.2 PVC Arms

1. Attach curved PVC pipe to the long, straight PVC pipe to create a hook shape. Repeat for the second set of pipes.
2. 3-D print two attachment pieces from the “PVC Pipe Connector.stl” file, as seen in Figure 33.

Figure 33: 3-D printed PVC pipe connector

6.2.3 Rotating Cylinders

1. Using a miter saw for cutting steel, cut the 12" long 10 mm diameter carbon steel rod into two 4" long segments.
2. Use a belt grinder to smooth out the cut ends of the rods.
3. Attach two set screw shaft collars back-to-back about 2" from one end of each rod. If necessary, use a belt grinder to decrease the diameter of the shaft collars until each one can fit comfortably in the bearing well of the skate wheels.
4. Slide a skate wheel onto each shaft collar and snap them into place.

6.2.4 Frame

1. Using a miter saw for aluminum, cut available 12" T-slots into four 6" segments.
2. Use any remaining scraps or 12" T-slots to cut five 3" segments and two 2" segments.
3. Using a 24" T-slot, drill 3/16" holes 0.5" from one end and 0.5" from the other end. Repeat for a second 24" T-slot.
4. Repeat step 3. After drilling the second hole, turn the T-slot 90 degrees and drill one more hole 0.5" from one end. Repeat for a second 24" T-slot.
5. Using a 6" T-slot, drill a 3/16" hole 2" from one end.
6. Using a 12" T-slot, drill 3/16" holes 4" from one end and 1" from the other end. Repeat for a second 12" T-slot.
7. Using a 2" T-slot, drill a 3/16" hole 0.5" from one end. Repeat for a second 2" T-slot.
8. Make ¼ 20" taps on all end profile holes on all T-slots that are 12" long or less using the process shown in Figure 34.

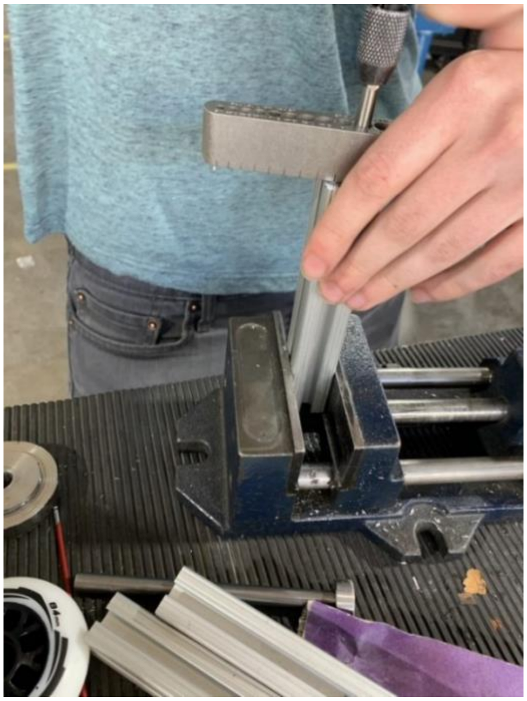


Figure 34: Tapping the end profile hole of a 6" T-slot.

6.2.5 Panels

1. Select a table saw for cutting wood.
2. Cut out nine rectangular pieces from a stock piece of acrylic panel according to the cut list show in Figure 35.

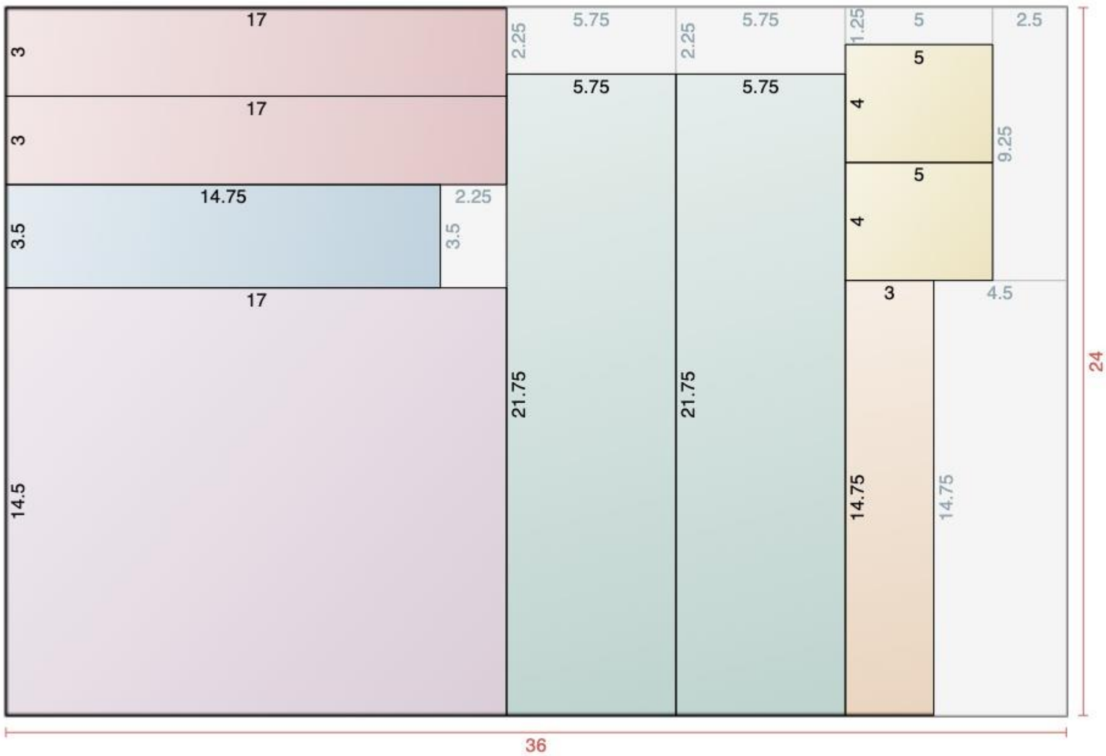


Figure 35: Cut list for the acrylic panels.

3. The 4"x5" pieces are the left and right flaps of the ramp. On the left flap, drill 11/64" holes at the locations listed in Table 4, with the top left corner as the datum:

Table 4: Hole alignments for left ramp flap.

Hole #	Horizontal Alignment (inches)	Vertical Alignment (inches)
1	7/8	5/16
2	7/8	1 3/16
3	31/32	2 11/16
4	31/32	3 7/16
5	1 15/16	3 1/16

4. On the right flap, drill 11/64" holes at the locations listed in Table 5, with the top left corner as the datum:

Table 5: Hole alignments for right ramp flap.

Hole #	Horizontal Alignment (inches)	Vertical Alignment (inches)
1	15/16	3 13/16
2	15/16	2 15/16
3	1 3/8	1
4	1 3/8	1 11/16
5	2 3/16	1 11/32

5. On the left flap, drill a 13/32” hole at (3 ¼”, 1 ¼”) using the same top left datum.
6. On the right flap, drill a 13/32” hole at (4 3/16”, 2 13/16”) using the same top left datum.

6.3 Assembly

Since most parts are being purchased off-the-shelf, the assembly processes for most components are simple. This section describes the assembly steps for these major components.

6.3.1 Computer Vision and Motor Control

NOTE: When assembling the electrical components, refrain from connecting components to the 12V battery source until fully wired.

1. While wiring components, please refer to the wiring diagram in Appendix G. First wire the 12V motors to the motor controllers.
2. Wire the motor controllers to the Arduino Nano.
3. Attach the camera module to the Raspberry Pi’s dedicated camera ribbon-cable input, referring to Figure 36.
4. Connect the Arduino to the Pi using the USB-B to mini-B cable.
5. Once all these connections have been made, wire the power and ground cables to each component.
6. Plug in the battery to the system power distribution cable.

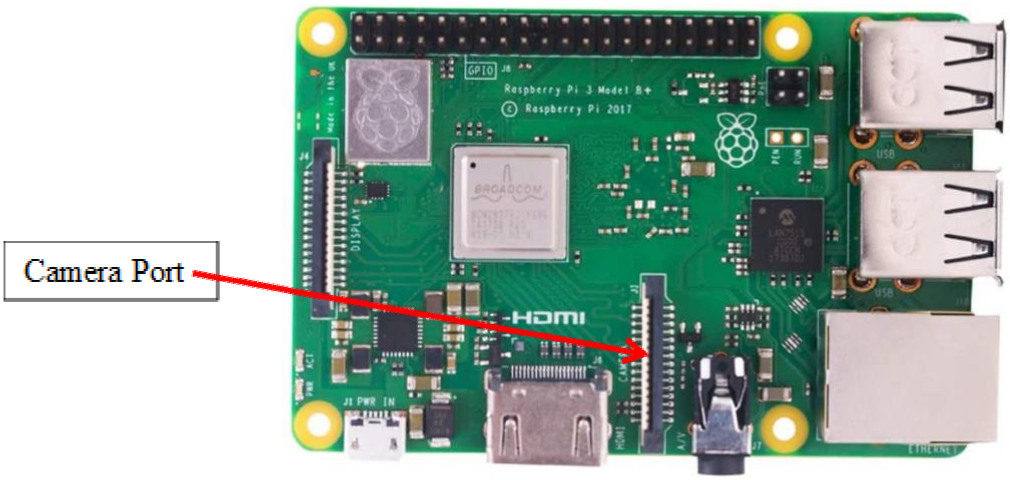


Figure 36: Raspberry Pi Model 3B+

6.3.2 Omni Wheels

1. Attach motor mounts and motors to the side of the chassis frame using M4 30 mm screws.
2. Using Figures 37 and 38 as reference, assemble the Omni wheel by:
   a) Placing one Omni wheel on top of the sonic hub with the center holes aligned.
   b) Placing one pattern spacer on top of the previous Omni wheel with the center hole aligned.
   c) Placing a second Omni wheel on top of the spacer, at a 90-degree rotation from the first Omni wheel. Ensure the center holes are aligned.
   d) Fasten the wheel together using four M4 30mm screws in a square pattern.
3. Attach the assembled wheel to the motor shaft and tighten the sonic hub.

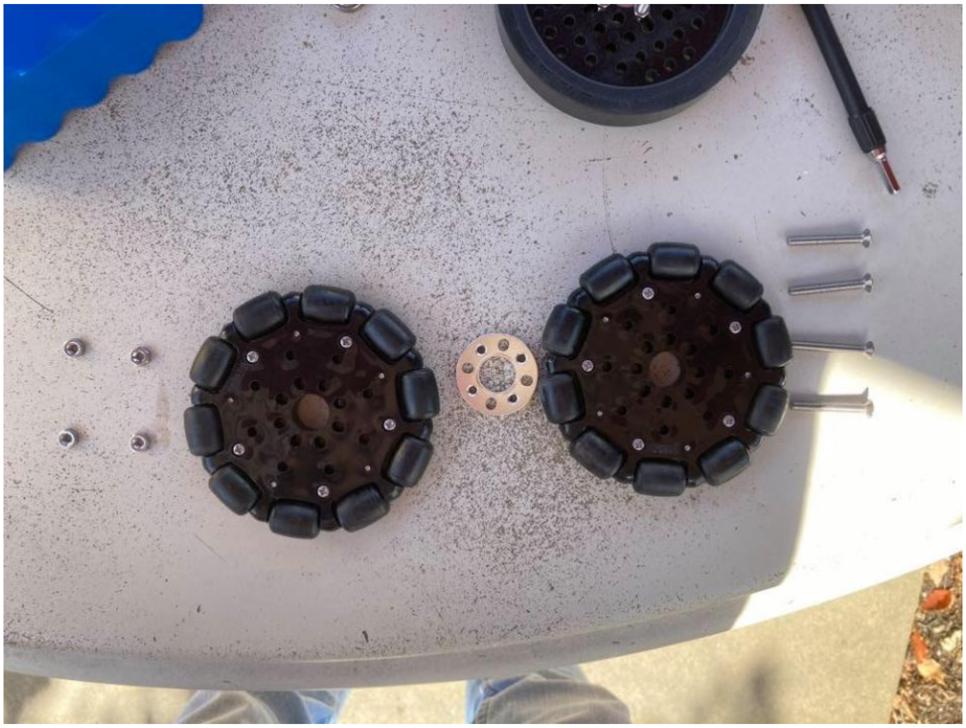


Figure 37: Omni wheel exploded view

Figure 38: Omni wheel assembly

6.3.3 Rubber Wheels

1. Attach shaft mount onto the side of the chassis frame using M4 30 mm screws.
2. Fasten ball bearing into the center bore of the rubber wheel shown in Figure 39.
3. Press-fit the steel shaft into the center of the ball bearing.
4. Attach the opposite end of the shaft onto the shaft support.
5. Repeat for each wheel.

Figure 39: Rubber wheel assembly

6.3.4 PVC Arms

1. **1.** Attach curved PVC pipe to the long, straight PVC pipe to create a hook shape. Repeat for the second set of pipes.
2. **2.** 3-D print two attachment pieces from CAD model.
3. **3.** Attach attachment pieces to the end of each long PVC pipe.

6.3.5 Chassis

1. **1.** Reference the 3-D CAD model provided for arrangement of 80/20 T-slots.
2. **2.** Assemble 80/20 T-slots using the standard end fasteners.
3. **3.** Latch the basket onto the elevated T-slot in the middle of the chassis and rest the back of the basket on the back frame.

6.3.6 Ball Collection

1. **1.** Attach motor mounts and motors to the front of the chassis frame using M4 30 mm screws.
2. **2.** Attach motor shaft and cylinder shaft to either end of a coupling. Tighten shafts in place using the set screws. Repeat for the second motor.
3. **3.** Attach rotating cylinders to each cylinder shaft, leaving a small amount of clearance off the ground (< 1/2”).
4. **4.** Stage ramp to position with the opening of the basket.
5. **5.** Slide PVC pipes into their respective slots.

6.4 Outsourcing

The team originally planned to outsource 80/20 Inc. to pre-cut and pre-drill the T-slots and pre-cut the acrylic panels. However, due to complications with COVID-19 related delays, 80/20 Inc. could not start our order by the time that we needed our materials. Consequently, we chose instead to order stock length T-slots and panels from McMaster Carr and completed all the outsourcing manufacturing ourselves. While this did cause some unplanned delays in our project timing, this inconvenience ultimately did not set our overall project off track.

6.5 Major Build Operations

Since only a few manufacturing processes were necessary for manufacturing all of the parts, only a handful of machines were needed to complete those processes. The first machine used in the manufacturing process was the miter saw, as seen in Figure 40. A miter saw for aluminum was used for cutting all of the 80/20 T-slots, and a miter saw for steel was used for cutting the steel rods for the rotating cylinders. A table saw for cutting wood was used to cut the acrylic panels.

Figure 40: Miter saw for cutting steel, located in the Aero Hangar.

After all of the parts were cut, a drill press was used to drill all necessary holes. The drill press we primarily used can be seen in Figure 41.

Figure 41: Drill press, located in the Aero Hangar.

The final machine we used in the manufacturing process was the belt sander. The belt sander served two purposes. The first purpose was to smooth the edges of the cut ends of the steel rods for the spinning cylinders. The second purpose was to slightly shave off and decrease the diameter of the shaft collars until they fit snugly into the bearing wells of the skate wheels.

6.6 Challenges & Lessons Learned

Our team encountered a variety of challenges throughout the process, but through each of these we were able to learn something new and continue to make progress. One of the earlier issues during our build was not receiving the 80/20 order. We were relying on 80/20 Inc. to do most of the manufacturing for our team since we knew space would be limited in the shops. This pushed our team members to come up with a new manufacturing plan at the drop of a hat and get into the shops to start the manufacturing right away. While this resulted in a longer manufacturing process, we were still able to finish all of the manufacturing within a reasonable time to start testing. Fortunately, most of the manufacturing we had to do was easily possible with the use of the basic shop machinery. Figure 42 shows the robot with all manufactured components attached and the electrical assembly included.

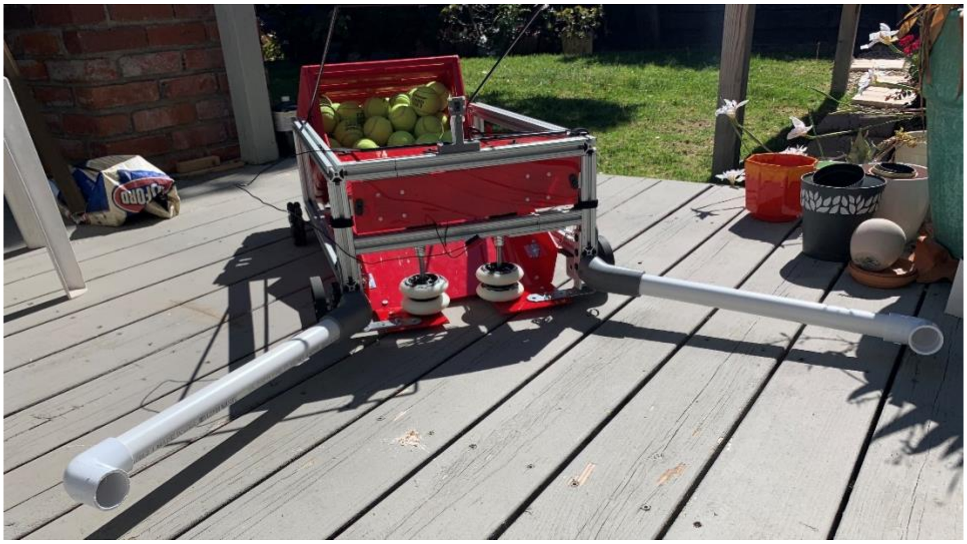


Figure 42: Fully Assembled Robot

Another challenge our team faced was the intricacy of our project. We have a lot of moving parts and getting all of those to work together was much more difficult than we expected. We had been working in two sub teams, mechanical and electrical, for the majority of the time. This often times resulted in a lack of clear communication about how the robot would look with all the components attached. For example, the bottom of the ramp is situated very close to the motors and we have a tight tolerance to ensure they do not interfere with each other. This was not something the two teams had considered and thus did not get resolved until the whole team was looking at the issue together. Another example of this difficulty was the fact that there is only one robot and both teams should ideally have been working on it during the entirety of the manufacturing process.

This was quite difficult since we frequently had to exchange the robot or coordinate which members would have it at one time. A positive side effect of this was that communication between the two teams became more streamlined over time. While allowing each team to have more individual time with the robot would have been nice, we also learned valuable communication and team dynamics skills that will prove useful in team environments in the future.

6.6 Recommendations for Future Production

The overall electrical subsystem that drives the robot is a solid foundation, but there are some parts that would be beneficial to upgrade. The first and most obvious electrical replacement would be the motors that drive the robot. Although their spec sheets make it seem like the motors should be powerful enough to drive the robot, they cannot. We recommend looking into 12 VDC, NEMA17 motors that provide more holding torque than the current motors. An alternative option is to add more motors. The downside to this is that it will decrease battery life. At the moment, all the electronics are connected through a breadboard attached to the front panel. This is not a good long-term solution as the vibrations from the robot operating might shake a cable loose. The ideal long-term solution would be to solder the connections together on a perf board and attach the perf board to the acrylic panel with nylon hardware. Another recommendation that we would make is to find a way to step down the voltage going to the DC motors for the ball collection cylinders. With the full 12 V powering them, they spin much too fast. This can be accomplished most easily by using a DC/DC converter module that can easily be found online.

7.0 Design Verification Plan

This chapter details how we tried to verify that our design met all the specifications, along with test results and numerical data as appropriate. We will also explain why some of the tests were not able to be completed and what we learned through this process.

7.1 Specification Discussions

With the implementation of complex electrical components and software, came additional uncertainties. This prompted us to review and revise the design specification table to address these new components. A revised design specification list can be seen in Table 6.

Table 6: Design Specification Table

Spec #	Specification Description	Result
1	Weight of fully assembled robot without basket or balls	Pass
2	Time to clear court	Fail
3	Collection cycles per charge	Fail
4	Time to empty collected balls to serving basket	Fail
5	Time spent operating device	Pass
6	Accuracy of ball detection software	Fail
7	Range of robot controller	Pass
8	Robot's ability to collect tennis balls	Fail

Spec 1: Dr. Chen has specified that anyone who is able to play tennis should be able to transport the robot. With that, the weight of the fully assembled robot must be less than 25 pounds.

Spec 2: The robot must quickly and efficiently collect tennis balls. Dr. Chen has specified that he would like the robot to clear a tennis court of tennis balls in no longer than 15 minutes.

Spec 3: Dr. Chen conducts multiple serving drills in a single session; therefore, the robot must have the capacity to complete at least five collection cycles before requiring a recharge.

Spec 4: The collected balls must be easily accessible from their storage unit and should take no longer than 30 seconds to empty into a serving basket.

Spec 5: With complicated electrical components and machine vision software, the robot should be intuitive in design. It should take no longer than five minutes to lightly assemble and operate the device.

Spec 6: Since we are using machine vision to detect and collect the balls, the machine vision software should detect over 95% of the tennis balls it sees.

Spec 7: Dr. Chen has expressed interest in a mobile app that would allow him to remotely command the robot. This app should have a range of at least 25 feet.

Spec 8: Dr. Chen has expressed concern in the ball collection method, which employs two rotating cylinders to catch and accelerate incoming tennis balls into the storage unit. To address his concern, the robot's current ball collection method must collect 100% of balls it receives.

7.2 Description of Completed Tests and Results

For the first specification, we used a scale the Mechanical Engineering department has on campus. There was a freight scale in the back of the HVAC labs that we gained access to and were able to weight the robot there. All the results from our testing can be seen in Table 7. All the weights recorded under 25 pounds and thus met our specification we had set forth. Along with that, the weight was close to the calculated number set earlier in the quarter based on estimated weights of all the parts. While the robot did meet the weight specifications, it is clunky and at times hard to transport. One recommendation we have for future development would be to install handles on top of the robot to provide an easier method of transporting. Currently, the user has to pick up the robot by the frame, which can prove difficult on occasion.

Table 7: Measured Weight for each Trial

Test #	Measured Weight (lbf)
1	24.5 ± 0.25
2	24.0 ± 0.25

The fourth specification involved testing the movement of tennis balls from the robot to a different collection location. The test would involve several volunteers and they would be given limited instructions and then asked to set up the ball collection basket in its standing form. As seen in Figure 43, the basket we chose for this project is able to stand up with the help of two wire legs that will remain attached to the basket during normal collection operations. This makes things simple for the user as all they have to do is detach the basket from the hooks and then flip the legs around, so they are supporting the weight of the tennis balls. An image of the hooks in as they are attached to the basket can be seen in Figure 44. For this test we utilized members of our households and asked them to be a part of this test. While all of them had seen the robot before, we did not provide any guidance when they were performing the test.

Figure 43: Ballport Tennis Ball Basket

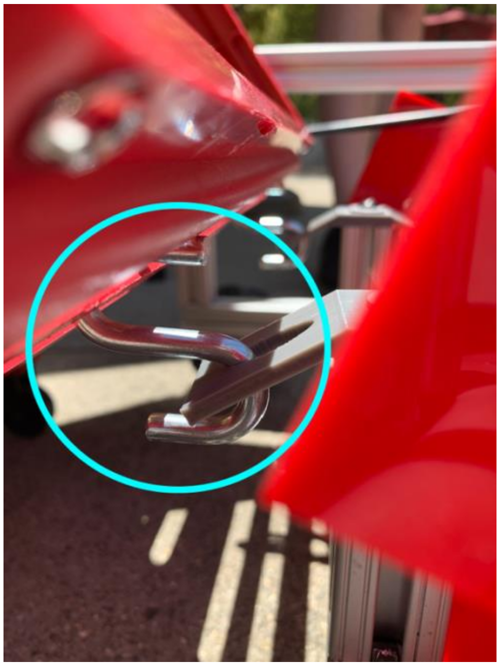


Figure 44: Ball Basket Connection Hook

The timed results can be seen in Table 8. The test results ranged from a high of 62 seconds to a low of only 18 seconds, with the average coming out to 36.12 seconds. This does not pass the specification we set forward of an average time of under 30 seconds. Some things we could have changes that would have influenced the results would be to show these volunteers how to detach the basket before having them go through the process or providing them with the user manual before usage. All of these would be things our sponsor will have access to, but if this product were to go to market, there would be consumers that are much more unfamiliar with the robot.

Table 8: Elapsed Time of Each Test Run

Test #	Elapsed Time (seconds)
1 (Thalia)	00:53.20
2 (Kahye)	00:23.94
3 (Michael)	00:18.86
4 (Sudeep)	01:02.43
5 (Galen)	00:22.15
Average Time	**00:36.12**

The fifth specification involved another user test that would measure the amount of time it takes to set the robot up before use. As our robot is a complex system that involves both mechanical and electrical components to set up, we wanted to ensure that our sponsor would not have to waste time setting up the robot each time he arrived at the tennis courts. For this test we set out the components necessary for the user to assemble and then asked them to put the robot together and start it up with limited guidance from our team. This involved attaching the basket, the PVC arms, and then opening the app on their phone and starting the robot's sequence of commands. A user assembling the robot can be seen in Figure 45. With limited support from our team, the user testers did very well with this specification.

Figure 45: PVC Arms Assembling Test

A list of the completion times can be seen in Table 9, with an average time of 2:44 minutes. As our sponsor and any other users will have access to the User Manual in the future, and these participants did not, we are satisfied with the results and feel confident that all users will be able to set up the robot in under five minutes.

Table 9: Elapsed Time of Each Run

User #	Elapsed Time (seconds)
1 (Thalia)	02:33.95
2 (Kahye)	01:24.76
3 (Michael)	01:42.40
4 (Sudeep)	03:45.20
5 (Galen)	04:16.41
Average Time	**02:44.54**

The seventh specification tested the maximum range our created app could control the robot. We tested this by pinging the Raspberry Pi and having it send a message back to the app. We continuously pinged the Raspberry Pi while stepping backwards until messages where no longer sent back to the app, indicating the connection has been lost. The distance from the Raspberry Pi and the controller was then measured. A list of the measured distances can be seen in Table 10. We are satisfied with these results, as the average maximum distance was 79 ft, which is approximately the length of a standard tennis court.

Table 10: Measured Maximum App Distance

Test #	Measured Distance (ft)
1	80 ± 1
2	79 ± 1
3	81 ± 1
4	80 ± 1
5	75 ± 1

Previously in the design process, Dr. Chen had expressed some concerns regarding the rotating cylinders and their ability to pick up the tennis balls. To mitigate this, we performed a test related specifically to their efficiency at propelling the tennis balls into the basket. While we had to modify this test slightly due to the current state of our robot, we were able to get some test results. For this test we decided to scale down the number of tennis balls we were using to 15, as throughout the building of the prototype, the battery being used to power the cylinders was not working as much as we hoped it would. While our robot was not able to move on its own for this test, we were able to push it forward to simulate that motion. The scaled-down results from this test can be seen in Table 11. Looking at the test results, while it did not pass our specification, we are satisfied with the results. None of the tennis balls got left behind, and all of them were able to pass through the cylinders. While we would have hoped more made it into the basket, with some modifications in the future we are confident that will be the case.

Table 11: Raw Data of Ball Collection Effectiveness

Test #	Ball stored securely in basket	Ball went through cylinders but could not make it up the ramp	Ball could not pass cylinders
1	10	5	0

A list of our completed test procedures can be seen in Appendix M, and the Design Verification Plan & Report can be seen in Appendix N.

7.3 Description of Uncompleted Tests and Results

Specification 2: Time to clear the court, was unable to be completed. This test was the most important and challenging to complete as it required all system components to be fully functional. Unfortunately, the assumptions made during the torque calculations of the motors were not conservative enough. We assumed a conservative total weight of 45 lbs evenly split between the four wheels, however after manufacturing and assembly, we noticed that most of the weight was concentrated in the front of the robot. Due to this, the oversized motors initially chosen to drive the wheels were having difficulty moving the robot, moving at too slow a pace necessary for testing. Additionally, the machine vision component outline in specification six, was never integrated. Code was written for the machine vision, however due to unforeseen circumstances within the team, it was not able to be implemented.

Specification 3: Collection cycles per charge, was also unable to be completed. Two tests were constructed for this specification, but neither were completed. The first test was to allow the robot to run freely on the tennis court and measure the time until it ran out of power. The second test was to spread tennis balls around the court and to have the robot collect the balls and measure how many collection cycles could be completed in one battery charge. Test two was not completed due to the system not yet being fully integrated. Without the machine vision component, the robot could only collect balls by driving in a straight path instead of seeking and collecting them. Test one could have been conducted, but as a team we decided against it as the machine vision component was not yet integrated. Additionally, although the battery is rechargeable, we felt that it was best to not needlessly drain the battery to conduct this test and harm the battery's overall capacity.

Lastly, Specification 6: Accuracy of ball detection software, was not completed. Software was written for the machine vision during the construction of our structural prototype. This preliminary machine vision was able to accurately detect all of the balls from a prerecorded video which was promising. However due to internal team issues, we were unable to implement this code onto the Raspberry Pi and test it on the tennis courts. Ideally, the software already written would translate well onto the courts, but this is not necessarily true. The software only provides a good base to work off of, as the detection is affected by a variety of variables such as lighting and background.

7.4 Discussion of challenges and lessons learned

In the end, our group did not get a chance to complete all the tests we had set out to run, due to unforeseen circumstances with both our internal team, and issues with the robot. We were not able to fully integrate the machine vision with the rest of the mechanical systems, and this meant that several of the specifications could not be tested. Along with that, the motors cannot move the robot in a manner that would be conducive for testing. Throughout the quarter our team encountered issues with the build involving addition and unexpected manufacturing problems, difficulty procuring the necessary code to implement the machine vision and struggles with the motor specifications given to use. While none of these are valid excuses as to why we did not finish the testing, hopefully they provide some insights. In terms of the lessons learned, I think our team realized that we should have allotted more time and energy to the actual testing. For so much of this last quarter we were focused on finishing the build and working out any issues we had there and did not have enough time to do sufficient testing at the end of the quarter. Another lesson we learned from this experience is that writing test procedures that are relevant and give you the information needed is a crucial skill. We spent a large chunk of time writing tests that would provide us with the data to confirm our results and that was a learning process for everyone involved.

8.0 Project Management

This section details the project design process and a timeline of key deliverables for the project in the form of a Gantt chart.

8.1 The Design Process

The overall design process began with a group interview with Dr. Chen to determine his specifications on the design as well as his wants and needs. Afterwards, research was conducted on similar products and existing designs to gather more information on what is already available on the market. College aged students were also interviewed to gather additional information on a typical customer’s needs and wants. After gaining a comprehensive understanding of the project, we created a scope of work report and presented it to our peers for feedback.

Following this was a functional decomposition and group brainstorming session to generate ideas and prototypes for each of the robot’s functions. Functions were combined to create many preliminary concepts. These designs were scored with a weighted decision matrix to determine which design performed best. A concept prototype of the best design was created and modeled using CAD. This design was presented to our peers for review and improved based on their feedback.

With the feedback of our peers, we moved forwards towards the final design prototype. It was determined early on that splitting into mechanical and electrical sub teams would most benefit the project. The mechanical subteam focused on the frame and ball collection, while the electrical subteam focused on coordinating and generating movement. Preliminary analyses were conducted to determine the required battery capacity and the required motor. Parts were sourced and a structural prototype was built. From testing, it was discovered that higher rpm motors and a better fixture were required to adequately collect tennis balls, and that the motors selected are able to drive the structural prototype. Additional preliminary testing was conducted on the machine vision software, which proved that the Raspberry Pi can not only process videos for the machine vision, but it also does so relatively quickly at eight frames per second.

With the success of the structural prototype and identification of potential issues, we continued moving forwards to construct the verification prototype. Custom manufactured Aluminum 8020 was ordered from 8020.net, however due to delays in manufacturing, the order was not due to arrive until much too late. As a result, the order was canceled, and stock 8020 parts were procured from McMaster-Carr. This set the project back a modest amount, as the team was not planning on conducting any manufacturing. Approximately four weeks was required for the mechanical subteam to finish manufacturing the 8020 and an additional two weeks for the electrical subteam to attach all of the electrical components. Afterwards we began full system testing, however due to time constraints and internal issues, none of the testing involving the machine vision was able to be completed. Table 12 outlines a summary of key deliverables throughout the project's timeline.

Table 12: Due Dates of Key Deliverables

Deliverable	Due Date
Preliminary Design Review	11/10/2020
FMEA & DFMA	11/17/2020
Design Analysis	11/21/2020
Material Selection	11/24/2020
Interim Design Review	01/25/2021
Source Materials	01/25/2021
Manufacturing Plan	02/02/2021
Critical Design Review	02/09/2021
Risk Assessment	02/17/2021
Manufacture Final Prototype	05/02/2021
Assemble Final Prototype	05/16/2021
Complete Prototype Testing	05/30/2021
Final Design Review	06/04/2021

Overall, as a team we find that the design process outlined above as well as the structure given to us by our advisor was very effective. In the Fall and Winter terms, we felt that we were always progressing forwards and improving upon the project. However, in the Spring, we ran into unexpected complications with manufacturing and assembly. The original Gantt chart created in the Fall and updated in the Winter was not beneficial as we greatly underestimated the time required to not only manufacture all of the individual parts from stock components but also assemble them. In the future, we would increase the allotted time for manufacturing to account for any unexpected issues. Additionally, we would include alternative tasks that can be done concurrently; the electrical subteam was both unable to assist the mechanical subteam with manufacturing, and to proceed forward as a completely assembled robot was required to test software.

8.*2 Gantt chart*

Our team decided to create a Gantt Chart using Microsoft Project to manage the progress of the project. This allowed us to identify the dependencies between deliverables and break down large deliverables into a series of smaller tasks. Each deliverable is marked by the team member in charge of it. A detailed image of the Gantt chart can be seen in Appendix O.

9.0 Conclusion & Recommendations

While writing this report has provided an opportunity for reflection, it also gives us a space to acknowledge that we did not fully achieve the goals we set for ourselves as a team. First, we would like to focus on our achievements. Building a robot is not easy and being able to seamless integrate both complex mechanical and electronic components is hard. While we didn't fully succeed with that, we do have a lot of smaller components fully fleshed out. We were able to get the tennis ball collection subsystem working, and the rotating cylinders can pick up tennis balls and propel them into the basket. This is one of the most fundamental things that our sponsor, Dr. John Chen asked us for, and we were able to deliver on that. We also got a strong start on the electrical components. We installed a Raspberry Pi on the robot, along with several other components to be able to communicate with the motors and control the wheels. In a broader sense, I think as a team we struggled to continuously push each other to be better. It sometimes felt like we were always working towards the next deadline and never thinking about project as something to build and improve upon. Along with that, we suffered difficulties working together on occasion, and that often meant that not everyone was participating in the report writing. Moving on to other things that didn't work, we did not set up a clear plan for combining the two sub systems and that was detrimental for our group. We also did not fully consider the weight of the robot and how the motors would react to that. I think we could have done more engineering analysis to get a better understanding of our system and how different forces would act on that. In speaking with our sponsor, he is obviously upset at the progress thus far, but optimistic that we have built a strong enough base to continue the project with other groups in the future. To help guide whatever work is done next, along with provide some resources to Dr. Chen, we have created a user manual that will provide some critical help if any challenges arise using the robot. This user manual can be found in Appendix P.

While as a team we are proud of the progress and work put into this project, there are things we would do differently if given the chance. The biggest difference would be how we structured the team and our interactions. By splitting the group of five into two sub teams we were both creating specialized groups to provide the best chance for success and creating a divide that would be hard to navigate through. One challenge we faced during the manufacturing process was coordination who had the robot and who needed to be working on it. Because we had to manufacture the frame and ramp first, that required a lot of time in the Machine Shop. Because of this the electrical sub team did not get as much time as they would have liked to run preliminary tests and get the electrical components properly set up. This caused us some issues when doing testing and set the group back. Another area we would change if given the opportunity revolves are the complexity of the project and the sheer number of different components. With the integration of both electrical and mechanical sub systems, issues arose when trying to secure parts and hardwire the connections. Looking back this might have been something we could have avoided, but it was in the scope of the project for us to consider both electrical and mechanical components. One thing we could have done to fix the issue was come up with an integration plan that would have considered the different factors required to ensure both sub teams have equal access to the robot and can spend time developing their components into the final design.

There are several specific things that can be done moving forward with the robot to get it working better. The first would be to add caster wheels to the robot. These would help redistribute the weight of the robot and alleviate some of the pressure on the front wheels and motors. Another step that could be taken with this issue to do a stronger engineering analysis of the weight distribution of the tennis balls and electrical components on the wheels. We did not consider that so much of the total weight of the robot would be concentrated in the front of the frame. Because this is where we also have the motors, it made it difficult for the motors to get enough torque to drive the robot forward in any meaningful way. Another area that needs improvement is the machine vision. While we have a great base for this topic, it has not been fully implemented on the robot. To be successful with that endeavor the machine learning code will need to be revisited and integrated with the Raspberry Pi that is currently on the robot. While this is not time consuming, we were not able to get this completed due to internal team issues. One final recommendation we have is to lower the voltage being supplied to the rotating cylinder motors. While there is currently a 12VDC battery supplying all the power, the rotating cylinders only need 9 volts. There are electrical components that can modify the amount of power being transmitted, it just was not in our means at the time to successfully integrate that. While this might seem like a lot of recommendations, it would completely transform the robot from its current state to being almost fully operational.

References

[1] Anno, Riley. Personal Interview. 24 September 2020.

[2] Attal, Kush. Personal Interview. 24 September 2020

[3] Chen, Vincent. Personal Interview. 26 September 2020

[4] Davenport, Maddie. Personal Interview. 26 September 2020

[5] “MultiMower.” OnCourt OffCourt, oncourtoffcourt.com/multimower/.

[6] “Introducing Tennibot®.” Tennibot, www.tennibot.com/.

[7] Seth, Radhika. “Radhika Seth.” Yanko Design - Modern Industrial Design News, 30 July 2014, www.yankodesign.com/2014/07/30/the-ball-boy/.

[8] “Vermont Tennis Ball Roller Mower & Hopper [85 Ball Capacity].” Vermont, www.vermontsports.com/vermont-tennis-ball-roller-mower-hopper-85-ball-capacity.

[9] “Store.tomohopper.com.” Ball Mower Tomohopper, store.tomohopper.com/category-s/1814.htm.

[10] “BallPicker.” Belrobotics, 9 Jan. 2020, www.belrobotics.com/en/mowers/ballpicker-connected-line/.

[11] Charles J. Mailman, “Tennis Ball Vacuum Collector”, US 8313396B1, April 20, 2012.

[12] Haitham Eletrabi, “Dual Function Robot and Storage Bin”, US 10676006B2, June 6, 2018.

[13] Byungwoon Park, “Device for golf ball retrieving”, KR 101873226B1, December 19, 2017.

[14] Lloyd Mendoza, “Tennis ball collecting, dispensing, and transport apparatus”, US 20060068948 A1, September 30, 2004.

[15] Xiaoshi Zhang, “Autonomous golf ball picking system”, US 20200070015A1, September 2, 2019.

[16] Morales, Alejandro, et al. California Polytechnic State University, San Luis Obispo, 2018, pp. 1–137, Lightweight Tennis Ball Pickup and Hopper.

[17] Sultan, Majdi, et al. Palestine Polytechnic University, 2018, pp. 1–98, *Tennis-Ball Collecting Robot*.

[18] Wei, Foo Shi. Universiti Tunku Abdul Rahman, 2012, pp. 1–78, *Design And Develop Of An Automated Tennis Ball Collector And Launcher Robot For Both Able-Bodied And Wheelchair Tennis Players – Ball Recognition System*

[19] Cabuk, Vehbi Umur, et al. “Design and Control of a Tennis Ball Collector Robot.” 2018 6th International Conference on Control Engineering & Information Technology (CEIT), 25 Oct. 2018. IEEE, doi:10.1109/ceit.2018.8751917.

[20] Chen, Hung-Kuang, and Jyun-Min Dai. “An Intelligent Tennis Ball Collecting Vehicle Using Smart Phone Touch-Based Interface.” 2016 International Symposium on Computer, Consumer and Control (IS3C), 4 July 2016, pp. 362–365. IEEE, doi:10.1109/is3c.2016.100.

[21] Elamvazuthi, I., et al. “Development of an Autonomous Tennis Ball Retriever Robot As an Educational Tool.” Procedia Computer Science, vol. 76, 2015, pp. 21–26. ScienceDirect, doi:10.1016/j.procs.2015.12.270.

[22] Yeon, Seong Ho, et al. “System Design for Autonomous Table Tennis Ball Collecting Robot.” 2017 17th International Conference on Control, Automation and Systems (ICCAS), 18 Oct. 2017, pp. 909–914. IEEE, doi:10.23919/iccas.2017.8204354
[23] Alwuqayan, Wuqayan and Neha Chawla, Ahmed Faizan, Sabri Tosunoglu. "Robotic Tennis Ball Collector." 2012 Florida Conference on Recent Advances in Robotics, 10 May 2012.
[24] “USTA Court Standards.” USTA, www.usta.com/content/dam/usta/pdfs/usta27 standards for establishing 60 tennis lines on pavement 12-5-10.pdf.
[25] “ITF Approval of Tennis Balls.” The International Tennis Federation, 10 May 2019, www.itftennis.com/media/2236/2020-itf-ball-approval-procedures.pdf.

Appendix [A] QFD House of Quality Table

Correlations	
Positive	+
Negative	–
No Correlation	

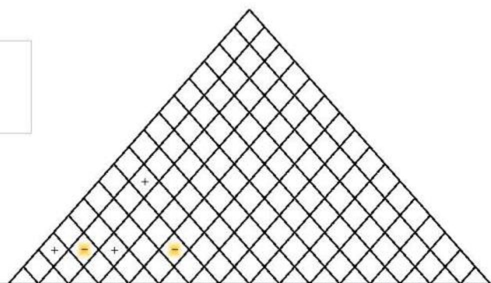
Relationships	
Strong	●
Moderate	○
Weak	

Direction of Improvement	
Maximize	▲
Target	
Minimize	▼

QFD House of Quality

Project: F71 Tennis Ball Collector

Revision Date: 10/06/2020



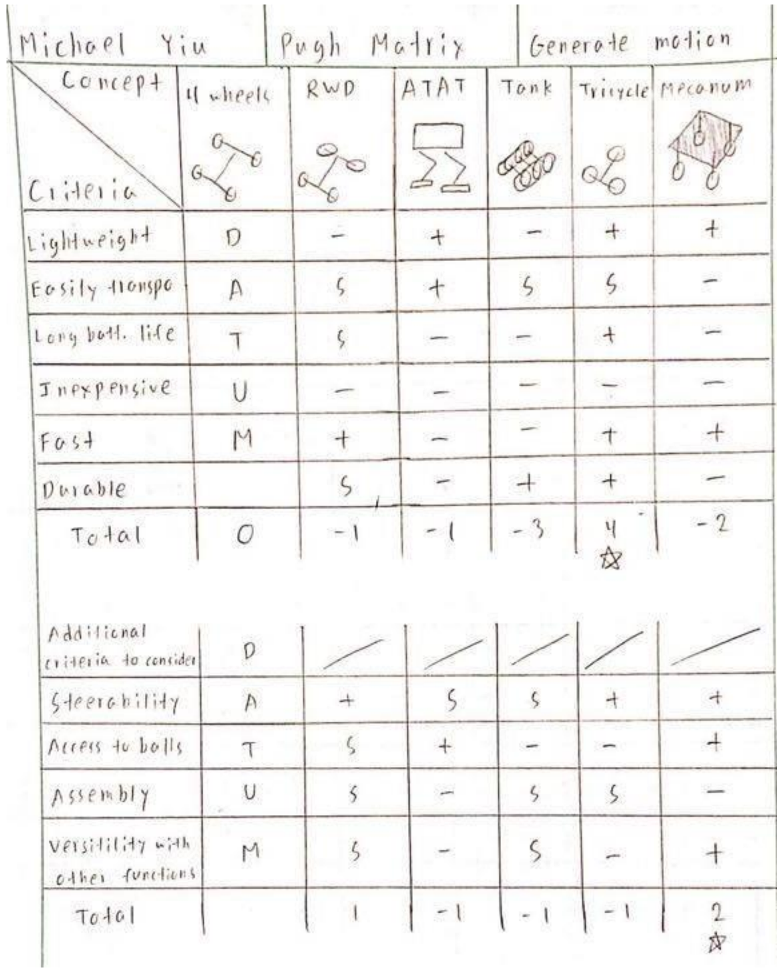
Row #	Weight Chart	Relative Weight	Individual Tennis Players	Tennis Clubs	Tennis Coaches	Dr. Chen (Sponsor)	Maximum Relationship	WHAT: Customer Requirements (Needs/Wants) \ HOW: Engineering Specifications (Tests)	Weight	Time to clear court	Size	Collection cycles per charge	Ball storage capacity	Cost to manufacture	Time to move balls to basket	Time spent operating device									Our Current Product	Tennibot	Multi Mower	Tomohopper	Basket Collector	Row #
								Column #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16						
								Direction of Improvement	▼	▼	▲		▲	▼	▼	▼														
1	■	10%	7	3	5	8	9	Lightweight	●		○	○														3	3	4	5	1
2	■	10%	6	3	8	6	9	Fast	○	●		○														5	4	4	2	2
3	■	14%	7	6	8	9	9	Hands-off while running		○						●										5	0	0	0	3
4	■	12%	8	3	7	8	9	Easily transported/stored	●		●															3	4	4	4	4
5	■	12%	5	7	8	5	9	Long battery life	○		○	●														3	5	5	5	5
6	■	9%	4	5	6	4	9	Holds many tennis balls			●		●													4	5	4	4	6
7	■	7%	6	3	3	4	9	Inexpensive						●												0	1	2	4	7
8	■	7%	3	5	4	3	3	Durable	○																	4	4	4	4	8
9	■	8%	4	3	5	5	9	Convenient access to balls					○		●											3	5	5	5	9
10	■	11%	6	7	5	6	9	Minimal initial setup								●										3	3	4	5	10
11		0%																												11
12		0%																												12
13		0%																												13
14		0%																												14
15		0%																												15
16		0%																												16

WHO: Customers — NOW: Curr. Products

HOW MUCH: Target Values	25 lbs	7 minutes	5 cubic feet	5 cycles	72 balls	500 dollars	30 seconds	5 minutes								
Max Relationship	9	9	9	9	9	9	9	9								
Technical Importance Rating	294.2	153.3	260.7	167	124.7	84.84	85.71	225	0	0	0	0	0	0	0	0
Relative Weight	21%	11%	19%	12%	9%	6%	6%	16%	0%	0%	0%	0%	0%	0%	0%	0%
Curr. Products: Our Current Product																
Tennibot	5	4	5	5	4	2	5	4								
Multi Mower	4	4	3	5	5	4	5	2								
Tomohopper	4	4	5	5	4	4	5	2								
Basket Collector	5	3	5	5	4	5	5	1								
Column #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Appendix [B] Decision Matrices

Pugh Matrices

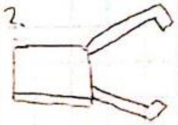
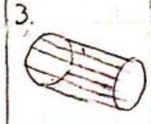
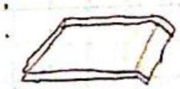


Michael Yiu	Pugh Matrix		Generate motion			
Concept / Criteria	4 wheels	RWD	ATAT	Tank	Tricycle	Mecanum
Lightweight	D	–	+	–	+	+
Easily transpo	A	S	+	S	S	–
Long batt. life	T	S	–	–	+	–
Inexpensive	U	–	–	–	–	–
Fast	M	+	–	–	+	+
Durable		S	–	+	+	–
Total	0	-1	-1	-3	4 ☆	-2

Additional criteria to consider	D	/	/	/	/	/
Steerability	A	+	S	S	+	+
Access to balls	T	S	+	–	–	+
Assembly	U	S	–	S	S	–
Versitility with other functions	M	S	–	S	–	+
Total		1	-1	-1	-1	2 ☆

Matthew Hoffman

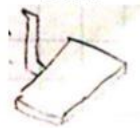
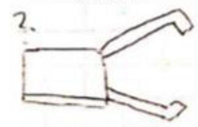
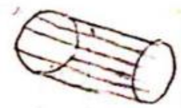
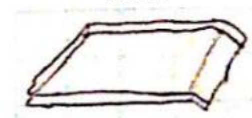
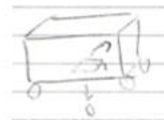
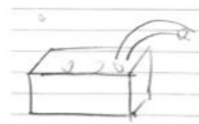
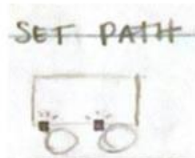
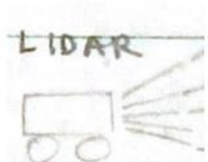
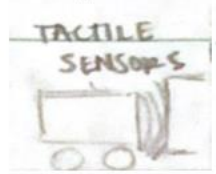
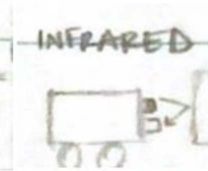
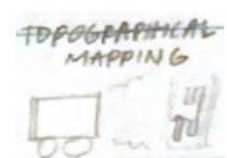
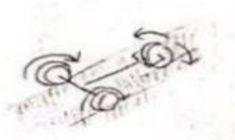
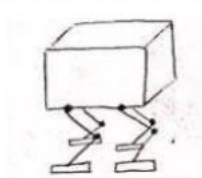
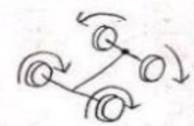
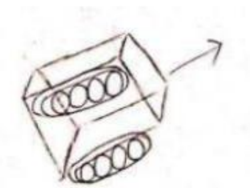
Collecting Tennis Balls Pugh Matrix

Criteria \ Concepts:	1. 	2. 	3. 	4. 	5. 
Lightweight	+	+	S	+	−
Easily transported / stored	+	S	S	S	−
Holds many tennis balls	−	+	S	−	+
Inexpensive	+	+	S	+	S
Durable	−	−	S	S	+
Convenient Access to Balls	−	+	S	S	+
Minimal Initial Setup	S	S	+	+	+

CONCEPTS / CRITERIA	SET PATH	LIDAR	TACTILE SENSORS	INFRARED	HEAT MAP	TOPOGRAPHICAL MAPPING	TENNIBOT
HANDS-OFF WHILE RUNNING	–	S	S	S	S	S	D
MINIMAL INITIAL SET UP	S	S	S	S	–	–	
INEXPENSIVE	+	–	+	+	–	+	A
FAST	–	+	–	–	+	+	
							T
SOFTWARE EASY TO FIGURE OUT	+	–	+	–	–	+	
MATERIALS EASILY ACCESSIBLE	+	S	+	+	–	+	U
COMPLEXITY OF SOFTWARE & DEVICE	+	+	+	S	–	+	M
TOTALS	2	0	3	0	-4	4	X

Coordinate movement

Morphological Matrix

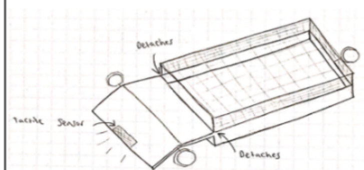
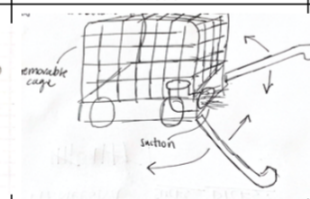
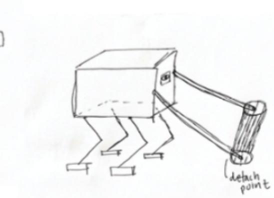
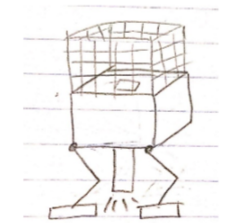
Function	Idea 1	Idea 2	Idea 3	Idea 4	Idea 5
Pick up balls	 Flexible attachment arm	 V-shaped collection arms	 Rolling Cage	 Paddlewheel	 Static collection tray (ramp)
Hold balls	 Removable Basket	 Removable Net	 Trapdoor	 Transfer Tube	
Coordinate motion					
Generate motion	 Tricycle	 ATAT walker	 RWD	 Mecanum	 Tank treads

Resultant Idea Combinations

1) RWD with V-shaped arms running a set path collecting balls into trapdoor
2) Tricycle with a ramp with tactile sensors that collect balls into a removable basket
3) Mecanum wheels with flexible arm/v-shaped arms + suction with topographical mapping and a removable basket
4) ATAT walker with rolling cage using LIDAR, the rolling cage acts as a removable basket
5) ATAT walker with suction that transports balls through a tube into a removable basket on top

Considerations: Suction tube connecting bottom bin into top bin
ATAT walker with suction tube

Weighted Decision Matrix

		Idea 1		Idea 2		Idea 3		Idea 4		Idea 5	
				 with Tactile Sensor		 with topographical mapping		 with LIDAR tech			
Specifications	**Weight**	**Score (1-5)**	**Total**	**Score (1-5)**	**Total**	**Score (1-5)**	**Total**	**Score (1-5)**	**Total**	**Score (1-5)**	**Total**
Lightweight	0.1	4	0.4	4	0.4	3	0.3	3	0.3	3	0.3
Fast	0.1	3	0.3	3	0.3	4	0.4	3	0.3	2	0.2
Hands-off while running	0.14	3	0.42	2	0.28	5	0.7	3	0.42	3	0.42
Easily transported/stored	0.12	3	0.36	4	0.48	3	0.36	4	0.48	3	0.36
Long battery life	0.12	4	0.48	4	0.48	3	0.36	2	0.24	2	0.24
Holds many tennis balls	0.09	3	0.27	3	0.27	4	0.36	2	0.18	4	0.36
Inexpensive	0.07	5	0.35	4	0.28	3	0.21	2	0.14	3	0.21
Durable	0.07	3	0.21	4	0.28	3	0.21	2	0.14	2	0.14
Convenient access to balls	0.08	3	0.24	4	0.32	4	0.32	4	0.32	4	0.32
Minimal initial setup	0.11	3	0.33	4	0.44	3	0.33	2	0.22	3	0.33
Total	1	3.36		3.53		3.55		2.74		2.88	

Resultant Idea Combinations

1) RWD with V-shaped arms running a set path collecting balls into trapdoor
2) Tricycle with a ramp with tactile sensors that collect balls into a removable basket
3) Mecanum wheels with flexible arm/v-shaped arms + suction with topographical mapping and a removable basket
4) ATAT walker with rolling cage using LIDAR, the rolling cage acts as a removable basket
5) ATAT walker with suction that transports balls through a tube into a removable basket on top

Concept Idea Discussion

3) We decided to proceed with idea 3. It proved to have the strongest design from the weighted decision matrix. We believe that the mecanum wheels will allow for the best mobility around the court. Coupled with the v-shaped arms and suction, this will maximize the ball collecting efficiency. Although idea 2 ranks only slighly below idea 3, the ramp design and its potential for ineffectiveness makes its an inferior design overall.

PDR Preliminary Analysis

Ball Storage Capacity

Required Target: 72 balls

Size of 1 tennis ball:

Diameter ≈ 2.7 in

$$Volume = \frac{4}{3}\pi r^3$$

$$V = \frac{4}{3}\pi\left(\frac{2.7\ in}{2}\right)^3$$

$$V = 10.3\ in^3 / ball$$

22.4 x 18.5 x 12.5
9.5 lb

9.5 x 9.75 x 13.0
7.7 lb 6.5 lb

Total Volume of tennis balls:

$$V_{Tot} = 72 \times 10.3\ in^3$$
$$V_{Tot} = 741.6\ in^3$$

Accounting for gaps between balls:

According to Jaeger and Nagel, the maximum packing density of random close-packed spheres is 64%

Therefore, the total volume of balls is only 64% of the total volume of the container in which they are stored.

$$V_{Tot} = 0.64\ V_{cont}$$

$$V_{cont} = \frac{741.6\ in^3}{0.64}$$

$$V_{cont} = 1159\ in^3$$

Recommended container volume: at least 1200 in^3

PDR Preliminary Analysis

Size

Recommended container volume: $1200\ in^3$

Max Dimensions:
L 3'
W 5.5'
H 1.25'

Max Volume = $35640\ in^3$

Container will likely match the length and width of the robotic device it is attached to. The maximum size of the device is significantly larger than the required ball storage size, so sizing of the device is not expected to be an issue.

Let's assume the robot will be 2.5' × 4' × 1' (L×W×H)

If a separate container is used, it will be approximately 24" × 42" × 12", assuming it is partially embedded into the frame of the robot.

PDR Preliminary Analysis

Weight

1 tennis ball ≈ 2 oz.
72 tennis balls ≈ 144 oz.

$$144 \text{ oz} \times \frac{1 \text{ lb}}{16 \text{ oz}} = \underline{9 \text{ lb}}$$

Must be able to carry at least 9 lb (72 tennis balls) of additional burden.

Assumptions:
- 4 Motors
- 4 Mecanum wheels
- 2.5' x 4' x 1' base (robot)
- 2' x 3.5' x 1' cage (container
- Cage made of carbon steel wire (similar weight to typical 72-ball ball pickers)
- The robot's frame will be made of Aluminum

Approximate Weights
Motors ≈ 1 lb ea.
Wheels ≈ 1/4 lb ea.
Aluminum ≈ 0.3 lb/ft^2
Cage ≈ 6.5 lb (using Ball Picker 72 as reference)

Total Weight

$$4(1 \text{ lb}) + 4\left(\tfrac{1}{4} \text{ lb}\right) + 6.5 \text{ lb} + (0.3 \text{ lb/ft}^2)\underbrace{\left[2(4\text{ft} \times 2.5\text{ft} + 1\text{ft} \times 2.5\text{ft} + 1\text{ft} \times 4\text{ft})\right]}_{\text{Approx. Surface Area of Robot}}$$

Weight = 21.4 lb

Using Aluminum w/ a weight of 0.3 lb/ft^2, the device will be within the 25 lb. requirement

PDR Preliminary Analysis

Time to Clear Court

Required clear time: 15 min

Assumptions:
- Full sized tennis court including outer boundaries is 120' x 60'.
- Device is operating at constant speed
- No balls are missed as it runs its course
- Wing span of collecting arms is 3 ft

$$\text{Area of court} = 120\text{ ft} \times 60\text{ ft}$$
$$= \underline{7200\text{ ft}^2}$$

$$\text{Velocity by area} = \frac{\text{Area}}{\text{Time}}$$
$$= \frac{7200\text{ ft}^2}{15\text{ min}\left(\frac{60\text{ s}}{1\text{ min}}\right)}$$
$$= \underline{8\text{ ft}^2/\text{s}}$$

Account for wing span:

$$V_{avg} = \frac{8\text{ ft}^2/\text{s}}{3\text{ ft}}$$

$$V_{avg} = 2.67\text{ ft/s}\left(\frac{1\text{ mile}}{5280\text{ ft}}\right)\left(\frac{3600\text{ s}}{1\text{ hr}}\right)$$

$$\boxed{V_{avg} = 1.82\text{ mph}}$$

Appendix [D] Design Hazard Checklist

Y	N	
Y		1. Will any part of the design create hazardous revolving, reciprocating, running, shearing, punching, pressing, squeezing, drawing, cutting, rolling, mixing or similar action, including pinch points and sheer points?
	N	2. Can any part of the design undergo high accelerations/decelerations?
Y		3. Will the system have any large moving masses or large forces?
	N	4. Will the system produce a projectile?
	N	5. Would it be possible for the system to fall under gravity creating injury?
	N	6. Will a user be exposed to overhanging weights as part of the design?
	N	7. Will the system have any sharp edges?
	N	8. Will any part of the electrical systems not be grounded?
	N	9. Will there be any large batteries or electrical voltage in the system above 40 V?
Y		10. Will there be any stored energy in the system such as batteries, flywheels, hanging weights or pressurized fluids?
	N	11. Will there be any explosive or flammable liquids, gases, or dust fuel as part of the system?
	N	12. Will the user of the design be required to exert any abnormal effort or physical posture during the use of the design?
	N	13. Will there be any materials known to be hazardous to humans involved in either the design or the manufacturing of the design?
	N	14. Can the system generate high levels of noise?
	N	15. Will the device/system be exposed to extreme environmental conditions such as fog, humidity, cold, high temperatures, etc?
	N	16. Is it possible for the system to be used in an unsafe manner?
	N	17. Will there be any other potential hazards not listed above? If yes, please explain on reverse.

For any "Y" responses, on the reverse side add:
(1) a complete description of the hazard,
(2) the corrective action(s) you plan to take to protect the user, and
(3) a date by which the planned actions will be completed.

Description of Hazard	Planned Corrective Action	Planned Date	Actual Date
1: The rotating cylinders used to propel the balls into the basket could potentially drag a person’s limb into the machine if a limb is inserted far enough to dislodge a stuck ball or for any other reason	Create a design that will stop the rotating cylinders if the robot is not on the ground and moving		
3: The robot itself will be a large, moving mass	Determine and implement a safe maximum ground speed to prevent the robot from causing significant bodily harm if it runs into a person		
10: Energy will be stored in a battery. Depending on the battery technology, a puncture could result in a fire	Design a protective shroud to cover the battery so that it cannot be punctured		

Appendix [E] - Risk Assessment

10S Bot

4/28/2021

designsafe Report

Application:	10S Bot	Analyst Name(s):	Michael Yiu, Alex Petrov, Frances Belcher, Matthew Hoffman, Robert Luttrell
Description:	F71 - Autonomous Tennis Ball Collecter	Company:	Cal Poly San Luis Obispo Senior Project
Product Identifier:		Facility Location:	
Assessment Type:	Detailed		
Limits:			
Sources:			
Risk Scoring System:	ANSI B11.0 (TR3) Two Factor		

Guide sentence: When doing [task], the [user] could be injured by the [hazard] due to the [failure mode].

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment Severity Probability	Risk Level	Risk Reduction Methods /Control System	Final Assessment Severity Probability	Risk Level	Status / Responsible /Comments /Reference
1-1-1	operator normal operation	mechanical : drawing-in / trapping / entanglement Rotating cylinders	Moderate Unlikely	Low	warning label(s), instruction manuals /Turn off electrical power	Moderate Unlikely	Low	Action Item [5/8/2021] Michael
1-1-2	operator normal operation	mechanical : pinch point Around the panels and attaching the V-shaped arms, ramps, and basket	Minor Likely	Low	instruction manuals, warning sign(s) /Not Applicable	Minor Unlikely	Negligible	
1-1-3	operator normal operation	mechanical : break up during operation Improper storage of V-shaped arms or too much pressure exerted on arms	Minor Likely	Low	instruction manuals, prevent energy buildup /Not Applicable	Minor Unlikely	Negligible	
1-1-4	operator normal operation	electrical / electronic : energized equipment / live parts Charging the battery	Minor Unlikely	Negligible	warning label(s), warning sign(s), instruction manuals /Not Applicable	Minor Unlikely	Negligible	
1-1-5	operator normal operation	electrical / electronic : water / wet locations Pushing the physical buttons to start the robot	Minor Remote	Negligible	warning label(s), warning sign(s) /Not Applicable	Minor Unlikely	Negligible	
1-1-6	operator normal operation	slips / trips / falls : trip Tripping over the robot	Moderate Unlikely	Low	on-the-job training (OJT) /Not Applicable	Moderate Unlikely	Low	Action Item [5/8/2021] Michael
1-1-7	operator normal operation	ergonomics / human factors : lifting / bending / twisting Transporting robot to and out of a vehicle	Minor Unlikely	Negligible	warning label(s), on-the-job training (OJT) /Not Applicable	Minor Unlikely	Negligible	

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment Severity Probability	Risk Level	Risk Reduction Methods /Control System	Final Assessment Severity Probability	Risk Level	Status / Responsible /Comments /Reference
1-1-8	operator normal operation	fire and explosions : hot surfaces Excessive heat generated from electrical componenets	Minor Unlikely	Negligible	warning label(s) /Not Applicable	Minor Unlikely	Negligible	
1-1-9	operator normal operation	fire and explosions : spontaneous combustion Improper handling of NiMH battery	Moderate Unlikely	Low	warning label(s), warning sign(s) /Not Applicable	Moderate Unlikely	Low	Action Item [5/8/2021] Michael
1-1-10	operator normal operation	environmental / industrial hygiene : hazardous waste Improper disposal/handling of NiMH battery	Moderate Unlikely	Low	warning label(s), instruction manuals /Not Applicable	Moderate Unlikely	Low	Action Item [5/8/2021] Michael
1-2-1	operator clean up	mechanical : pinch point Around the panels and attaching the V-shaped arms, ramps, and basket	Minor Unlikely	Negligible	instruction manuals, warning sign(s) /Not Applicable	Minor Unlikely	Negligible	
1-2-2	operator clean up	ergonomics / human factors : lifting / bending / twisting Transporting robot to and out of a vehicle	Minor Unlikely	Negligible	warning label(s), on-the-job training (OJT) /Not Applicable	Minor Unlikely	Negligible	
1-3-1	operator basic trouble shooting / problem solving	mechanical : drawing-in / trapping / entanglement Rotating cylinders	Moderate Unlikely	Low	warning label(s), instruction manuals /Turn off electrical power	Moderate Unlikely	Low	Action Item [5/8/2021] Matthew
1-3-2	operator basic trouble shooting / problem solving	electrical / electronic : energized equipment / live parts Seeing if there are any loose connections	Minor Unlikely	Negligible	instruction manuals, fixed enclosures / barriers /Not Applicable	Minor Unlikely	Negligible	
1-4-1	operator load / unload materials	mechanical : pinch point Attaching and detaching the basket	Minor Unlikely	Negligible	warning sign(s), instruction manuals /Not Applicable	Minor Unlikely	Negligible	
1-4-2	operator load / unload materials	ergonomics / human factors : repetition Repeatedly lifting basket	Minor Unlikely	Negligible	on-the-job training (OJT), instruction manuals /Not Applicable	Minor Unlikely	Negligible	

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment Severity Probability	Risk Level	Risk Reduction Methods /Control System	Final Assessment Severity Probability	Risk Level	Status / Responsible /Comments /Reference
1-4-3	operator load / unload materials	ergonomics / human factors : lifting / bending / twisting Lifting basket full of tennis balls	Minor Unlikely	Negligible	on-the-job training (OJT), instruction manuals /Not Applicable	Minor Unlikely	Negligible	
2-1-1	maintenance technician parts replacement	mechanical : pinch point Replacing the V-shaped arms	Minor Unlikely	Negligible	warning sign(s), instruction manuals /Not Applicable	Minor Unlikely	Negligible	
2-1-2	maintenance technician parts replacement	electrical / electronic : energized equipment / live parts Replacing the battery	Minor Unlikely	Negligible	warning sign(s), instruction manuals, warning label(s) /Not Applicable	Minor Unlikely	Negligible	
2-1-3	maintenance technician parts replacement	electrical / electronic : shorts / arcing / sparking Shorts produced from improper resoldering	Moderate Unlikely	Low	wire to electrical codes (NEC/IEC) /Turn off electrical power	Moderate Unlikely	Low	Action Item [5/8/2021] Alex
2-1-4	maintenance technician parts replacement	electrical / electronic : improper wiring Unexpected behavior due to improper wire connections	Moderate Unlikely	Low	wire to electrical codes (NEC/IEC) /Not Applicable	Moderate Unlikely	Low	Action Item [5/8/2021] Alex
2-1-5	maintenance technician parts replacement	fire and explosions : spontaneous combustion Improper handling of the battery	Moderate Unlikely	Low	warning label(s), warning sign(s), instruction manuals /Not Applicable	Moderate Unlikely	Low	Action Item [5/8/2021] Robert
2-1-6	maintenance technician parts replacement	environmental / industrial hygiene : hazardous waste Improper handling of the battery	Moderate Unlikely	Low	warning label(s), warning sign(s), instruction manuals /Not Applicable	Moderate Unlikely	Low	Action Item [5/8/2021] Robert
2-2-1	maintenance technician adjust controls / settings / alignment	mechanical : drawing-in / trapping / entanglement Adjusting robot's wheels	Minor Unlikely	Negligible	warning sign(s), instruction manuals /Not Applicable	Minor Unlikely	Negligible	
2-2-2	maintenance technician adjust controls / settings / alignment	electrical / electronic : energized equipment / live parts Electric shock while accessing logic boards due to battery not being disconnected	Minor Unlikely	Negligible	regularly inspect electrical connections to ensure proper grounding/connections /Not Applicable	Minor Unlikely	Negligible	

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment Severity Probability	Risk Level	Risk Reduction Methods /Control System	Final Assessment Severity Probability	Risk Level	Status / Responsible /Comments /Reference
2-3-1	maintenance technician trouble-shooting / problem solving	mechanical : drawing-in / trapping / entanglement Seeing if the motors are driving the wheels correctly	Moderate Unlikely	Low	instruction manuals /Not Applicable	Moderate Unlikely	Low	Action Item [5/8/2021] Matthew
2-3-2	maintenance technician trouble-shooting / problem solving	electrical / electronic : energized equipment / live parts Seeing if there are any loose connections	Minor Unlikely	Negligible	regularly inspect electrical connections to ensure proper grounding/connections /Not Applicable	Minor Unlikely	Negligible	
2-3-3	maintenance technician trouble-shooting / problem solving	electrical / electronic : improper wiring Seeing if wires are not connected to the appropriate terminals	Minor Unlikely	Negligible	regularly inspect electrical connections to ensure proper grounding/connections /Not Applicable	Minor Unlikely	Negligible	
3-1-1	passer by / non-user work next to / near machinery	slips / trips / falls : trip Walking into the robot	Moderate Unlikely	Low	on-the-job training (OJT), supervision /Not Applicable	Moderate Unlikely	Low	Action Item [5/8/2021] Frances
3-2-1	passer by / non-user walk near machinery	slips / trips / falls : trip Walking into the robot	Moderate Unlikely	Low	on-the-job training (OJT), supervision /Not Applicable	Moderate Unlikely	Low	Action Item [5/8/2021] Frances

Appendix [F] Failure Modes & Effects Analysis

Design Failure Mode and Effects Analysis

Product: **Autonomous Tennis Ball Collector** Prepared by: ______________________

Team: **F71** Date: ______________ (orig)

System / Function	Potential Failure Mode	Potential Effects of the Failure Mode	Severity	Potential Causes of the Failure Mode	Current Preventative Activities	Occurence	Current Detection Activities	Detection	Priority	Recommended Action(s)	Responsibility & Target Completion Date	Action Results: Actions Taken	Action Results: Severity	Action Results: Occurence	Action Results: Criticality
Coordinate motion/ Provide directions	Robot is not provided correct directions	Tennis balls are unable to be collected	7	1) microcontroller reads directions incorrectly 2) wires come unplugged	1) solder wires 2) unit test the communication	3	Unit tests and logging communication	3	63						
Coordinate motion/ Determine location of tennis balls	Robot does not determine location of tennis balls	Robot misses tennis balls	6	1) balls look different from training set used on computer 2) camera is obstructed, damaged, or dirty 3) camera is unplugged	1) Use wide variety of ball and court colors for training set 2) Protect camera in housing 3) solder connections	3	Record video to computer and manually verify that balls are detected	3	54						
General/ Support weight of balls	Robot is too heavy	Catastrophic error, robot falls apart. Robot is too heavy and cannot move	7	1) structural material is too dense 2) motors and batteries are too heavy 3) tennis balls push the weight over the max	1) find a compromise between strength and density 2) find motors and batteries that are not as heavy	2	Using a scale. Stress analysis	2	28						
Collect balls/ Move balls to storage	Robot does not store balls properly	Balls are not sent to the basket, effectively not collected	7	1) balls get stuck on the ramp 2) balls flow back down the ramp 3) balls never go up the ramp	1) include tolerances for tennis ball size 2) design ramp to prevent overflow	2	test by rolling balls up the ramp and seeing what happens	2	28						
Collect balls/ Provide storage	Robot runs out of space	Balls are collected but not stored	5	1) balls overflow 2) storage area is not large enough	1) design a larger volume storage 2) design a method of prevent overflow	2	test by rolling many(72) balls up a ramp and seeing there is still space	2	20						
Generate Motion / Receive direction from computer	Directions are not recevied	1) robot is unable to move on the court 2) robot is unable to collect the balls from the court	7	Signal between computer and robot is broken, components on robot are broken and can't relay information	Check all device connections and ensure that components on robot are functioning throughout prototyping	3	Run unit tests from the computer, regularly check components on robot to ensure they are up and running.	2	42						
General / Provide energy to other systems	Robot is not able to run, no energy means no movement	1) robot is unable to move on court 2) robot is unable to transmit or receive signals 3) robot is unable to collect balls	7	1) battery is broken or out of charge 2) Wires connecting battery to other components are torn	1) make sure battery is rechargeable or check the power of it before usage 2) solder the wires to ensure they can't come loose or tear	2	1) test the battery during prototyping and usage using multimeter 2) during prototyping, check wires for dislodged connections	3	42						

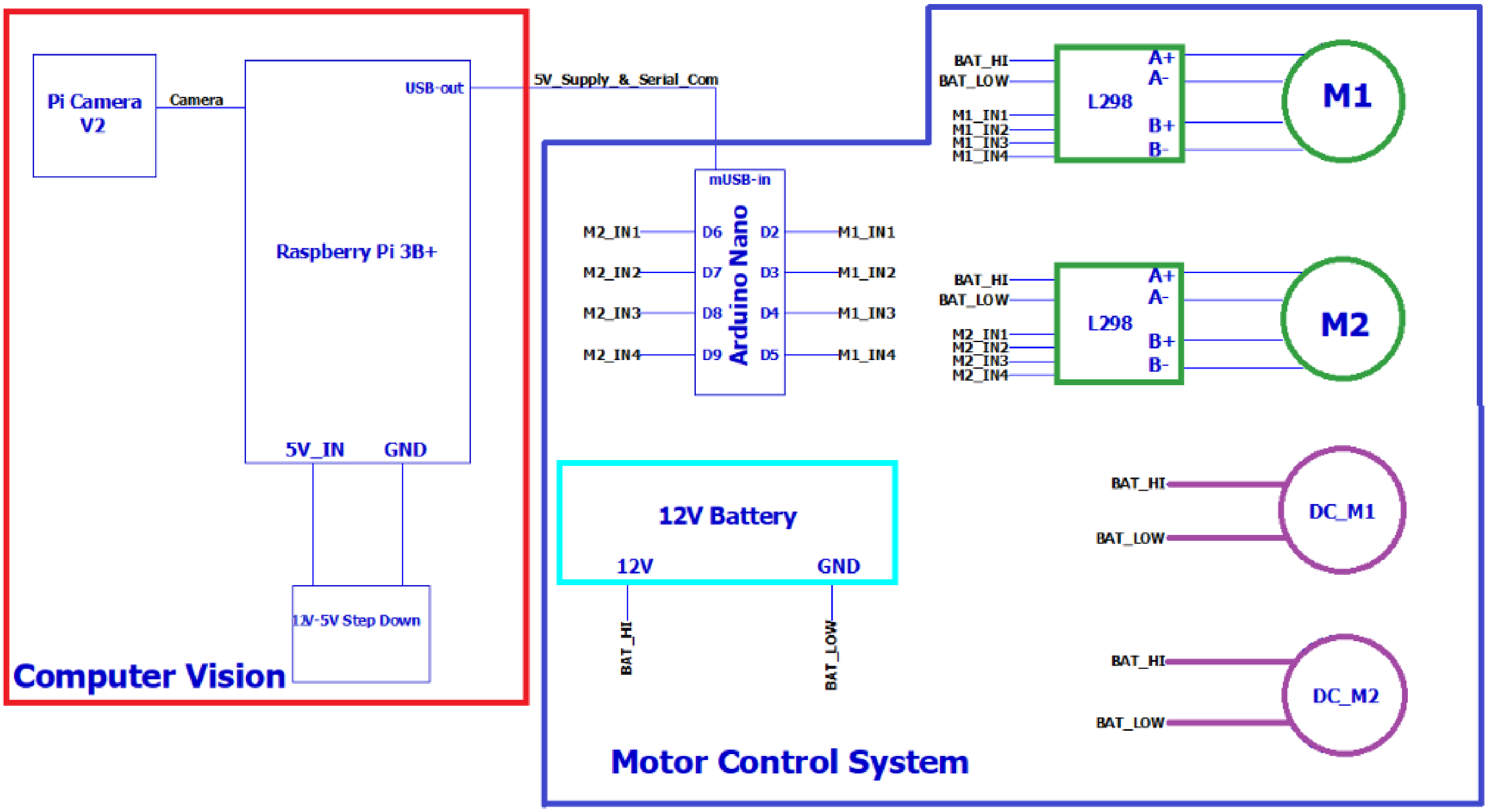
Design Failure Mode and Effects Analysis

Product: ____________________________ Prepared by: ____________________________

Team: ____________________________ Date: ________________ (orig)

												Action Results			
System / Function	Potential Failure Mode	Potential Effects of the Failure Mode	Severity	Potential Causes of the Failure Mode	Current Preventative Activities	Occurence	Current Detection Activities	Detection	Priority	Recommended Action(s)	Responsibility & Target Completion Date	Actions Taken	Severity	Occurence	Criticality
General / Hold parts together	Parts separate during use & robot falls apart	Robot is unusable	7	Fasteners are not secured in place during construction or are faulty	1) make sure fasteners are quality products 2) make sure fasteners are secure and won't move during use	1	Buy fasteners from a reputable website and do some testing before we install them in the robot. Tighten fasteners correctly and make sure they are secure.	3	21						
Collect balls / Corral balls	V-shaped arms break or get stuck on fence or net	1) robot is unable to move on court 2) balls are collected less efficiently 3) robot is unable to collect balls	5	V-shaped arms get caught in the fence or net while moving and break off as a result	Design curved ends on the arms to prevent them from sticking through links in fence or net	4	Test the arm design against a chainlink fence and the net	2	40						
Collect balls / Catch and move balls to storage	Ball grabber does not pick up balls and/or move the balls to ramp	1) robot is unable to collect balls 2) robot is unusable 3) balls do not reach storage	8	1) Not enough friction between ball grabber and ball 2) Ball grabber does not oprate at a fast enough speed 3) Balls get jammed in grabber or ramp channel	1) Choose material with good friction with tennis balls 2) Ensure device operates at fast enough speed to get balls up the ramp	3	Test interaction between balls and grabbing device	2	48						
Generate motion / Propel robot in a given direction	Robot does not move properly according to messages received from computer	1) Robot moves in wrong direction 2) Robot does not move at all	6	Errors in code or mechanical jam of rollers	1) Test code thoroughly for specific types of motion on the court	3	Observe motion for irregularities	3	54						

Appendix [G] Electrical Subsystem Wiring Diagram



Appendix [H] All Completed Code

We were unable to contact Robert, and thus this code is not commented. To see more information, check the Github. https://github.com/AlexPetrov75/10s_Bot

BallCollector.py

```
import BallDetector
import cv2


class BallCollector:

    def __init__(self, cap, hsv_thresh_lower, hsv_thresh_upper, filter_mode,
show_frames_on=True, input_scale=None,
                 output_scale=None, print_frametime=False):

        self.ball_detector = BallDetector.BallDetector(cap, hsv_thresh_lower, hsv_thresh_upper,
filter_mode,
                                                show_frames_on=show_frames_on, input_scale=input_scale,
                                                output_scale=output_scale, print_frametime=print_frametime)
        self.detected_balls = None

    def loop(self):
        while True:
            self.ball_detector.detect_balls()
            self.detected_balls = self.ball_detector.detect_balls()
```

BallDetector.py

```
import cv2
import numpy as np
import detection_utils
import time
import ImageSource
import PreprocessorInterface
import CannyPreprocessor
import ThresholdPreprocessor


class BallDetector:

    def __init__(self, cap, hsv_thresh_lower, hsv_thresh_upper, filter_mode,
show_frames_on=True, input_scale=None, output_scale=None, print_frametime=False):
        self.hsv_thresh_lower = hsv_thresh_lower
        self.hsv_thresh_upper = hsv_thresh_upper
        self.cur_detected_balls = None
```

```
        self.show_frames_on = show_frames_on
        self.input_scale = input_scale
        self.output_scale = output_scale
        self.print_frametime = print_frametime
        self.output_img = None

        self.image_source = ImageSource.ImageSource(cap)

        self.preprocessor = self.get_preprocessor(filter_mode, hsv_thresh_lower, hsv_thresh_upper)

    @staticmethod
    def get_preprocessor(filter_mode, hsv_thresh_lower, hsv_thresh_upper):
        if filter_mode == "canny":
            return CannyPreprocessor.CannyPreprocessor()
        elif filter_mode == "thresh":
            return ThresholdPreprocessor.ThresholdPreprocessor(hsv_thresh_lower,
hsv_thresh_upper)
        else:
            raise Exception("Unknown preprocessing mode: " + str(filter_mode))

    def draw_circles(self):
        if self.cur_detected_balls is not None:

            # Convert the circle parameters a, b and r to integers.
            detected_balls_int = np.uint16(np.around(self.cur_detected_balls))

            min_b = detected_balls_int[0, :][0][1]
            min_index = 0
            i = 0
            for pt in detected_balls_int[0, :]:
                a, b, r = pt[0], pt[1], pt[2]
                if b > min_b:
                    min_index = i
                    min_b = b

                # Draw the circumference of the circle.
                cv2.circle(self.output_img, (a, b), r, (0, 255, 0), 2)

                # Draw a small circle (of radius 1) to show the center.
                cv2.circle(self.output_img, (a, b), 1, (0, 0, 255), 3)
                i += 1
            pt_min = detected_balls_int[0, :][min_index]
            a_min, b_min, r_min = pt_min[0], pt_min[1], pt_min[2]
            cv2.circle(self.output_img, (a_min, b_min), r, (255, 0, 0), 5)

    def show_frames(self):
```

```
        if self.output_scale:
            self.output_img = detection_utils.resize_with_aspect_ratio_rel(self.output_img,
self.output_scale)

        cv2.imshow("output", self.output_img)
        if cv2.waitKey(1) & 0xff == ord('q'):
            quit()

    def detect_balls(self):
        prev = time.time()
        raw = self.image_source.get_frame()

        if self.input_scale:
            raw = detection_utils.resize_with_aspect_ratio_rel(raw, self.input_scale)

        self.output_img = raw.copy()

        frame_to_hough = self.preprocessor.preprocess(raw)
        self.cur_detected_balls = cv2.HoughCircles(frame_to_hough,
                                    cv2.HOUGH_GRADIENT, 1, 10, param1=50,
                                    param2=20, minRadius=5, maxRadius=15)

        self.draw_circles()

        if self.show_frames_on:
            self.show_frames()

        if self.print_frametime:
            t = time.time()
            print("Frametime (ms): " + str((t - prev) * 1000))
            prev = t

        return self.cur_detected_balls
```

CannyPreprocessor.py

```
from PreprocessorInterface import PreprocessorInterface
import numpy as np
import cv2


class CannyPreprocessor(PreprocessorInterface):

    def __init__(self):
        super().__init__()

    def preprocess(self, raw_img: np.ndarray):
        return cv2.Canny(raw_img, 50, 100)
```

ImageSource.py

```
import cv2
import numpy as np


class ImageSource:

    def __init__(self, cap: cv2.VideoCapture):
        self.cap = cap

    def get_frame(self) -> np.ndarray:
        _, frame = self.cap.read()
        return frame
```

PreprocessorInterface.py

```
import numpy as np


class PreprocessorInterface:

    def __init__(self):
        pass

    def preprocess(self, raw_img: np.ndarray) -> np.ndarray:
        pass
```

ThresholdPreprocessor.py

```
from PreprocessorInterface import PreprocessorInterface
import cv2
import numpy as np


class ThresholdPreprocessor(PreprocessorInterface):

    def __init__(self, hsv_thresh_lower: np.array, hsv_thresh_upper: np.array):
        super().__init__()
        self.hsv_thresh_lower = hsv_thresh_lower
        self.hsv_thresh_upper = hsv_thresh_upper

    def preprocess(self, raw_img: np.ndarray) -> np.ndarray:
        hsv = cv2.cvtColor(raw_img, cv2.COLOR_BGR2HSV)
        mask = cv2.inRange(hsv, self.hsv_thresh_lower, self.hsv_thresh_upper)
        masked = cv2.bitwise_and(raw_img, raw_img, mask=mask)
        gray = cv2.cvtColor(masked, cv2.COLOR_BGR2GRAY)
        gray_blurred = cv2.blur(gray, (3, 3))
        return gray_blurred
```

detection_utils.py

```
import cv2
```

```
def resize_with_aspect_ratio_rel(image, scale_ratio, inter=cv2.INTER_AREA):
    width = int(image.shape[1] * scale_ratio)
    height = int(image.shape[0] * scale_ratio)
    dsize = (width, height)
    output = cv2.resize(image, dsize)
    return output


def resize_with_aspect_ratio_abs(image, width=None, height=None, inter=cv2.INTER_AREA):
    dim = None
    (h, w) = image.shape[:2]

    if width is None and height is None:
        return image
    if width is None:
        r = height / float(h)
        dim = (int(w * r), height)
    else:
        r = width / float(w)
        dim = (width, int(h * r))

    return cv2.resize(image, dim, interpolation=inter)
```

main.py

```
import cv2
import os
import BallDetector
import BallCollector
import numpy as np

if __name__ == "__main__":
    test_vid_path = os.path.join(os.getcwd(), "Test", "tennisballwizard.mp4")
    cap = cv2.VideoCapture(test_vid_path)
    lower_yellow = np.array([23, 10, 187], dtype=np.uint8)
    upper_yellow = np.array([89, 190, 255], dtype=np.uint8)
    bc = BallCollector.BallCollector(cap, lower_yellow, upper_yellow, "thresh", input_scale=0.5,
print_frametime=True)
    bc.loop()
```

range_detector.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# source: https://github.com/jrosebr1/imutils
# USAGE: You need to specify a filter and "only one" image source
#
# (python) range-detector --filter RGB --image /path/to/image.png
# or
```

```
# (python) range-detector --filter HSV --webcam

import cv2
import argparse
from operator import xor


def callback(value):
   pass


def setup_trackbars(range_filter):
   cv2.namedWindow("Trackbars", 0)

   for i in ["MIN", "MAX"]:
      v = 0 if i == "MIN" else 255

      for j in range_filter:
         cv2.createTrackbar("%s_%s" % (j, i), "Trackbars", v, 255, callback)


def get_arguments():
   ap = argparse.ArgumentParser()
   ap.add_argument('-f', '--filter', required=True,
               help='Range filter. RGB or HSV')
   ap.add_argument('-i', '--image', required=False,
               help='Path to the image')
   ap.add_argument('-w', '--webcam', required=False,
               help='Use webcam', action='store_true')
   ap.add_argument('-p', '--preview', required=False,
               help='Show a preview of the image after applying the mask',
               action='store_true')
   args = vars(ap.parse_args())

   if not xor(bool(args['image']), bool(args['webcam'])):
      ap.error("Please specify only one image source")

   if not args['filter'].upper() in ['RGB', 'HSV']:
      ap.error("Please speciy a correct filter.")

   return args


def get_trackbar_values(range_filter):
   values = []

   for i in ["MIN", "MAX"]:
      for j in range_filter:
         v = cv2.getTrackbarPos("%s_%s" % (j, i), "Trackbars")
         values.append(v)
```

```
    return values


def main():
    args = get_arguments()

    range_filter = args['filter'].upper()

    if args['image']:
        image = cv2.imread(args['image'])

        if range_filter == 'RGB':
            frame_to_thresh = image.copy()
        else:
            frame_to_thresh = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    else:
        camera = cv2.VideoCapture(0)

    setup_trackbars(range_filter)

    while True:
        if args['webcam']:
            ret, image = camera.read()

            if not ret:
                break

            if range_filter == 'RGB':
                frame_to_thresh = image.copy()
            else:
                frame_to_thresh = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

        v1_min, v2_min, v3_min, v1_max, v2_max, v3_max = get_trackbar_values(range_filter)

        thresh = cv2.inRange(frame_to_thresh, (v1_min, v2_min, v3_min), (v1_max, v2_max, v3_max))

        if args['preview']:
            preview = cv2.bitwise_and(image, image, mask=thresh)
            cv2.imshow("Preview", preview)
        else:
            cv2.imshow("Original", image)
            cv2.imshow("Thresh", thresh)

        if cv2.waitKey(1) & 0xFF is ord('q'):
            break


if __name__ == '__main__':
    main()
```

command.h

```
#ifndef COMMAND_H
#define COMMAND_H
typedef struct Command
{
   char command_char;
   float arg;
} Command;
#endif
```

main.c

```
#include <stdio.h>
#include <string.h>
#include "command.h"
#include "command_utils.h"

int main(int argc, char *argv[])
{
   Command **cmds = parseCommands("asdf");

   float testNum = 123.456;
   printf("testNum: %f\n", testNum);
   char bytes[4];
   float2Bytes(bytes, testNum);

   char bytesToUse[5] = {bytes[0], bytes[0], bytes[1], bytes[2], bytes[3]};
   float testNumCalc = getCommandArg(bytesToUse, 0);

   printf("testNumCalc: %f\n", testNumCalc);

   return 0;
}
```

command_utils.h

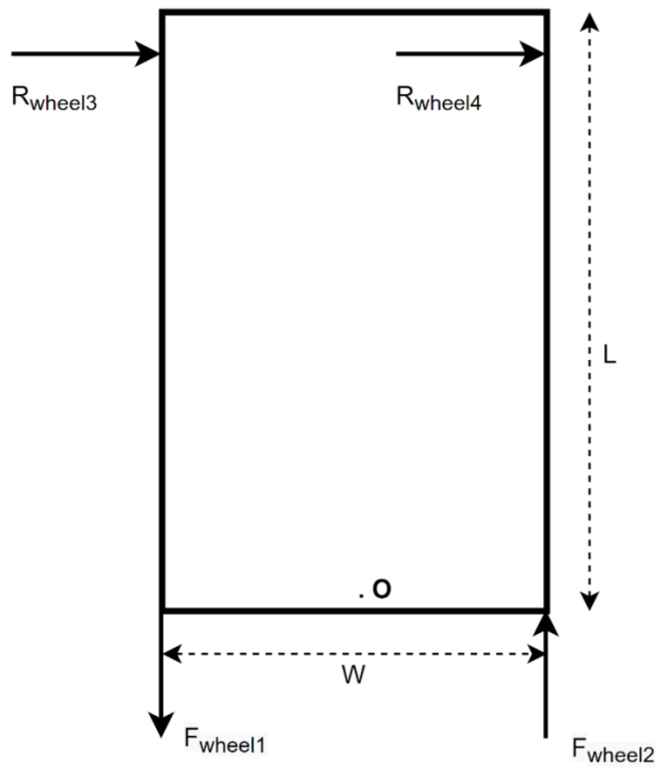
```
#ifndef COMMAND_UTILS_H
#define COMMAND_UTILS_H
#include "command.h"
#define MAX_CMDS_PER_ARR 5

Command** parseCommands(char *byteArr);
float getCommandArg(char *byteArr, char curLetterIdx);
void float2Bytes(char bytes[4], float floatVariable);

#endif
```

command_utils.c

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "command_utils.h"

void float2Bytes(char bytes[4], float floatVariable)
{
   /* Used to populate byte arrays for testing */
   memcpy(bytes, (unsigned char*) (&floatVariable), 4);
}

Command** makeEmptyCommandArr(char *byteArr, char numCommands)
{
   /* Malloc array of struct pointers with one extra spot for null terminator */
   Command** emptyCommandArr = malloc(sizeof(Command *) * (numCommands + 1));

   /* Malloc each struct in array */
   for (int i = 0; i < numCommands; i++)
      emptyCommandArr[i] = malloc(sizeof(Command));

   /* Add null terminator */
   emptyCommandArr[numCommands] = NULL;

   return emptyCommandArr;
}

float getCommandArg(char *byteArr, char curLetterIdx)
{
   float f;
   char b[] = {byteArr[curLetterIdx + 1],
            byteArr[curLetterIdx + 2],
            byteArr[curLetterIdx + 3],
            byteArr[curLetterIdx + 4]};
   memcpy(&f, &b, sizeof(f));
   return f;
}

Command** parseCommands(char *byteArr)
{
   char numCommands = byteArr[0];

   Command **result = makeEmptyCommandArr(byteArr, numCommands);

   for (char i = 0; i < numCommands; i++)
   {
      Command *cmd = result[i];
      char curLetterIdx = 5 * (i - 1) + 1;
      cmd->command_char = byteArr[curLetterIdx];
      cmd->arg = getCommandArg(byteArr, curLetterIdx);
   }
```

```
    return result;
}
```

Appendix [I] – Design Analysis
Required Torque Calculations



When opposite torques are applied to the two driven rubber wheels, forces f_{wheel1} and f_{wheel2} are produced at the interface between the wheels and the ground. To produce rotational motion, the torque produced by these forces must produce a moment strong enough to overcome the static rolling resistance of the rollers on the omnidirectional wheels as they begin to translate transversely.

Spinning a moment about O to calculate the required force,

$$\Sigma \mathrm{M_o} = \frac{W}{2}(F_{wheel1} + F_{wheel2}) - L(R_{wheel3} + R_{wheel4}) = 0$$

Where

$$F_{wheel1} = F_{wheel2} = F_w$$
$$R_{wheel3} = R_{wheel4} = R_w$$

$$0 = \frac{W}{2}(2\mathrm{F_w}) - \mathrm{L}(2\mathrm{R_w})$$
$$0 = WF_w - 2LR_w$$
$$F_w = \frac{2R_wL}{W}$$

Equation 1

Assuming an equal weight distribution among the four wheels, R_w can be expressed in terms of the coefficient of rolling resistance.

$$R_w = F_{weight} \cdot \mu_R$$
$$R_w = \frac{mg}{4} \mu_R$$

Equation 2

Plugging Equation 2 into Equation 1 yields

$$F_w = \frac{mgL\mu_R}{2W}$$

Equation 3

Where F_w is the friction force required to be exerted by each wheel on the ground in order to overcome the rolling resistance of the omniwheel rollers and induce rotation. Equation 3 can be expressed more usefully in terms of the torque required by the motors if the wheel's radius *R* is known.

$$T_{motor} = \frac{L}{W} \frac{Rmg\mu_R}{2}$$

Substituting known values in,

$$T_{motor} = \frac{0.508\text{m}}{0.457m} \cdot \frac{(0.048m)(18.1kg)(9.81m/s^2)(0.30)}{2}$$
$$T_{motor} = 7.85Ncm$$

Battery Calculations

Approximate how many Amps our system will be pulling (based on data sheets):
x2 DC Motors Max Amp Pull => 0.16 A
x2 Stepper Motors Max Amp Pull => 0.35 A
Average Raspberry pull with 2 I/Os => 1.5 A
This gives us: 0.16 + 0.16 + 0.35 + 0.35 + 1.5 = 2.52 A

We want to run the robot for about 75 minutes which is 1.25 hours.
2.52 A x 1.25 h = 3.15 Ah = 3150 mAh

We need to buy a battery sized around 3150 mAh

Appendix [J] - Project Budget & iBOM

F71 - Autonomous Tennis Ball Collecting Robot

Indented Bill of Material (iBOM)

Assembly Level	Part Number	Description	Qty	Cost	Ttl Cost	Source	More Info
		Lvl0 Lvl1 Lvl2 Lvl3 Lvl4					
0	100000	Final Assembly				------	
1	110000	Carriage Assembly				------	
2	111000	8020 T-slots	13	$6.44	83.72	McMasterCarr	part # 47065T101
2	112000	Standard End fasteners	20	$1.60	32.00	8020.net	part 3682
2	113000	Motor Mounts with Hardware	2	$8.95	17.90	Adafruit	part no 1297
2	114000	8020 Panels	7	$10.28	71.96	8020.net	part no. 2669
2	115000	3D printed basket attachment	2	$0.00	0.00	Custom	PLA
2	116000	Roll-in Panel Mounts	14	$3.50	49.00	8020.net	part no. 2489
2	117000	Straight Flat Plate	6	$5.55	33.30	8020.net	part no. 4117
2	118000	T-nuts	24	$0.50	12.00	8020.net	part no. 3321
1	120000	Wheel Assembly				------	
2	121000	Rubber Wheel Assembly				------	
3	121100	Rubber wheels	2	$5.99	11.98	Gobilda	item #3612-0014-0096
3	121200	Sonic Hub	2	$6.99	13.98	Goblida	part #1309-0016-1006
2	122000	Omniwheel Assembly				------	
3	122100	Omniwheels	4	$6.99	27.96	Gobilda	item #3604-0014-0120
3	122200	Shaft (8mm by 100mm)	2	$2.09	4.18	Gobilda	2100-0008-0100
3	122300	Spacers	2	$2.49	4.98	Goblida	part #1504-0032-0040
3	122400	Linear Rail Shaft Support	2	$3.95	7.90	Adafruit	part no 1182
3	122500	Flanged Ball Bearings	1	$3.49	3.49	Gobilda	1611-0514-0006
3	122600	U-Block Channel	2	$2.49	4.98	Gobilda	1121-0001-0048
2	123000	M4 Nuts	16	$0.85	13.60	Home Depot	part #810208
2	124000	M4 Bolts	16	$0.85	13.60	Home Depot	part #843798
1	130000	Electrical Assembly				------	
2	131000	Computer Vision and Communication				------	
3	131100	Raspberry Pi 3B+	1	$35.00	35.00	Digikey	part #1690-1025-ND
3	131200	Camera	1	$25.00	25.00	Digikey	part #1690-1011-ND
2	132000	Motor Control System				------	
3	132100	Ardunio Nano	1	$22.00	22.00	Digikey	item #1050-1001-ND
3	132200	L298 Motor Controllers	2		0.00	Donated	part #L298N
3	132300	Stepper Motors	2	$14.00	28.00	Digikey	part #1528-1062-ND
3	132400	DC Motors	2	$18.88	37.76	Amazon	FBA_TRS-775W
3	132500	Nylon Fastener Set	1	$11.96	11.96	Amazon	B0744MMJ9V
3	132600	Cable Management Set	1	$14.99	14.99	Amazon	B08P5VH157
2	133000	Battery	1	$44.99	44.99	Studica	part #70018
3	133100	Charger	1	$29.99	29.99	Studica	part #70019
1	140000	Collection Assembly				------	
2	141000	Cylinder Assembly				------	
3	141100	Rotating Cylinder Wheels	4	$8.57	34.28	Amazon	87048
3	141200	Tourna Ballport Deluxe	1	$43.00	43.00	Amazon	BPD-80W
3	141300	Ball Ramp	1	$ 10.28	10.28	8020.net	part no. 2669
4	141310	3D printed angled ramp bracket	2	$ -	0.00	Custom	PLA
3	141400	1/8 in. x 1 - 1/4 in. Zinc Plated S-Hook	2	$1.50	3.00	Home Depot	Store SKU #566440
3	141500	30 degree Angle Backet	2	$2.95	5.90	McMasterCarr	33125T44
3	141600	Set Screw Shaft Couplings	2	$3.41	6.82	McMasterCarr	part #539T111
3	141700	8mm Shaft	2	$2.09	4.18	Gobilda	2100-0008-0100
3	141800	Corner Bracket	2	$4.15	8.30	McMasterCarr	part # 1556A49
		2 in. Zinc-Plated Inside Corner Brace	2	$2.76	5.52	Home Depot	Store SKU #1002316496
2	142000	Arm Assembly				------	
3	142100	PVC Piping	2	$19.30	38.60	McMasterCarr	part #49035K47
3	142200	3D printed connector	2	0.00	0.00	Custom	PLA
3	142300	PVC Elbow	2	3.49	6.98	McMasterCarr	part #9102K113
		Holding Pin	2	2.13	4.26	McMasterCarr	part #98404A020
1	150000	Wires	TBD		823.08	Donated	22 AWG
	Total Parts		**178**				

Final Cost	
parts cost	$823.08
shipping	$83.72
tax	$34.60
Total	$941.40

Appendix [K] Drawing Package

1000 - Final Assembly

1001 - Exploded top level assembly

- **1100** - Carriage Assembly (1)
  - **1110 -** 8020 T-slots (2)
  - **1120 -** Standard End fasteners (3)
  - **1130 -** Motor Mount with Hardware (4)
  - **1140** - 8020 Panels (5)
  - **1150 -** 3D Printed Basket Attachment (6)
  - **1160 -** Roll-in Panel Mounts (7)
- **1200 -** Wheel Assembly (8)
  - **1210 -** Rubber Wheels (9)
    - **1211 -** Sonic Hub (10)
  - **1220** - Omni Wheel Assembly (11)
    - **1221 -** Omni Wheel (12)
    - **1222 -** 8mm Shaft (13)
    - **1223** – Spacers (14)
    - **1224 -** Linear Rail Shaft Support (15)
    - **1225 -** Flanged Ball Bearings (16)
    - **1226 -** U-block channel (17)
  - **1230 -** M4 Fasteners (18)
  - **1240 -** M4 Nuts (19)
- **1300 -** Electrical Assembly (20)
  - **1310 -** Computer Vision and Communication
    - **1311 -** Raspberry Pi 3B+
    - **1312 –** Camera
    - **1313** – 12 V to 5 V Stepdown
  - **1320 -** Motor Control System Wiring Diagram
    - **1321 -** Arduino Nano
    - **1322 -** L298 Motor Controllers
    - **1323 -** Stepper Motors
    - **1324 -** DC Motors
    - **1325 –** Nylon Fastener Set
    - **1326 –** Cable Management Set
  - **1330 -** Battery Specs
    - **1331 -** Battery Charger Specs
- **1400 -** Collection Assembly (21)
  - **1410 -** Cylinder Assembly (22)
    - **1411 -** Rotating Cylinder Wheels (23)
    - **1412 -** Basket

1413 - Ramp (24)

14131 - 3-D printed angled ramp bracket

1414 - 1/8 in. x 1 - 1/4 in. Zinc Plated S-Hook (25)

1415 - T-slot Stopper (26)

1416 - Set Screw Shaft Coupling (27)

1417 - 8mm Shaft (28)

1418 - Corner Bracket (29)

1419 – 2 in. Zinc-Plated Inside Corner Brace

1420 - Arm Assembly (30)

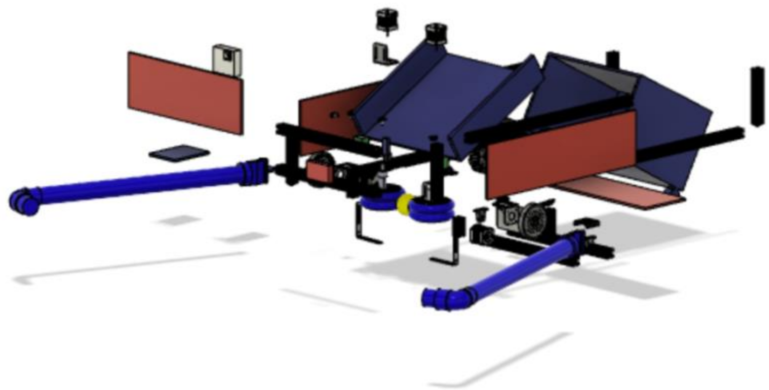
1421 - PVC Piping (31)

1422 – 3-D printed connector (32)

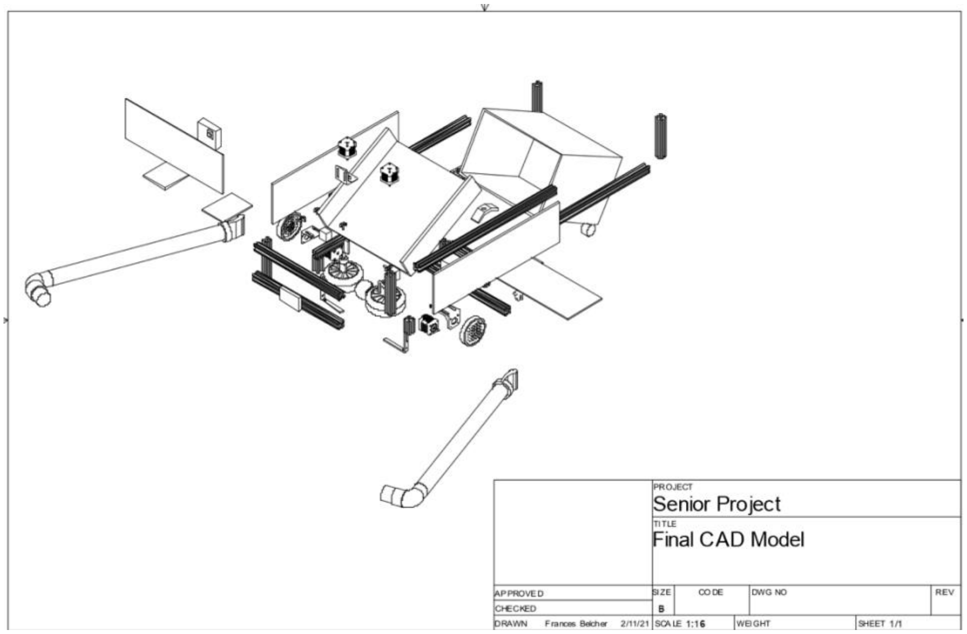
1423 - Curved PVC end piece (33)

1424 - Holding Pin

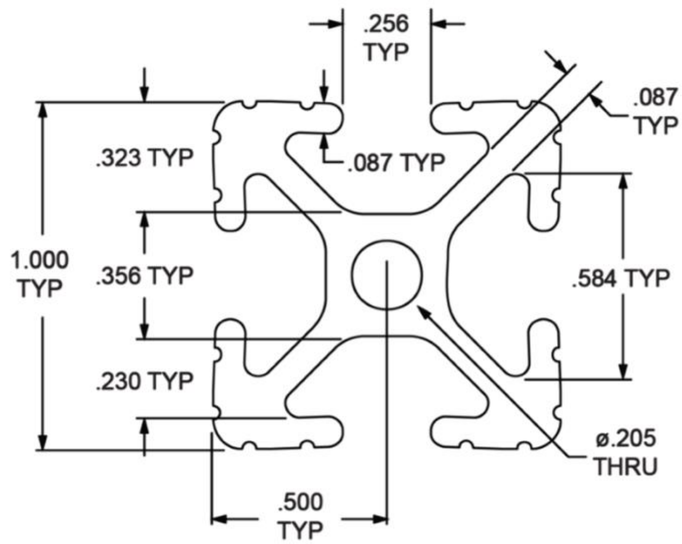
Part 1000 - Final Assembly



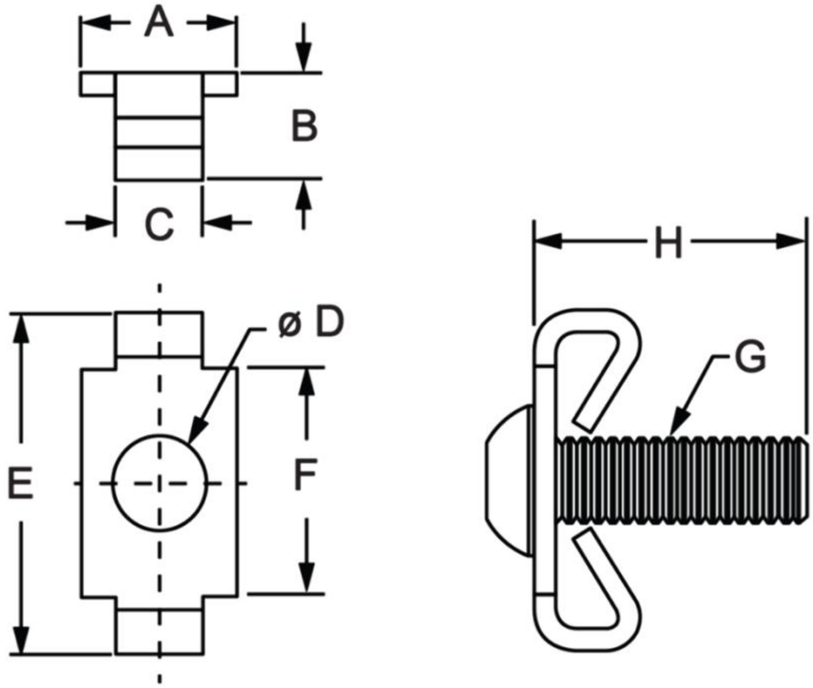
Part 1001 - Exploded Top Level Assembly



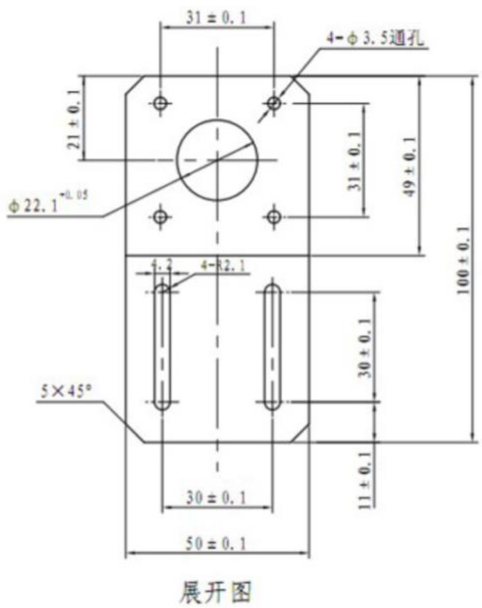
Part 1110 - 8020 T-slots

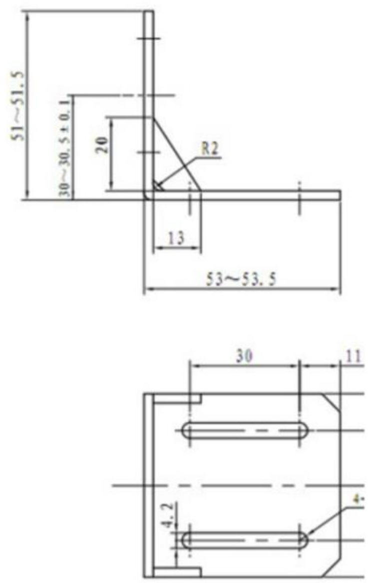


Part 1120 - Standard End fasteners

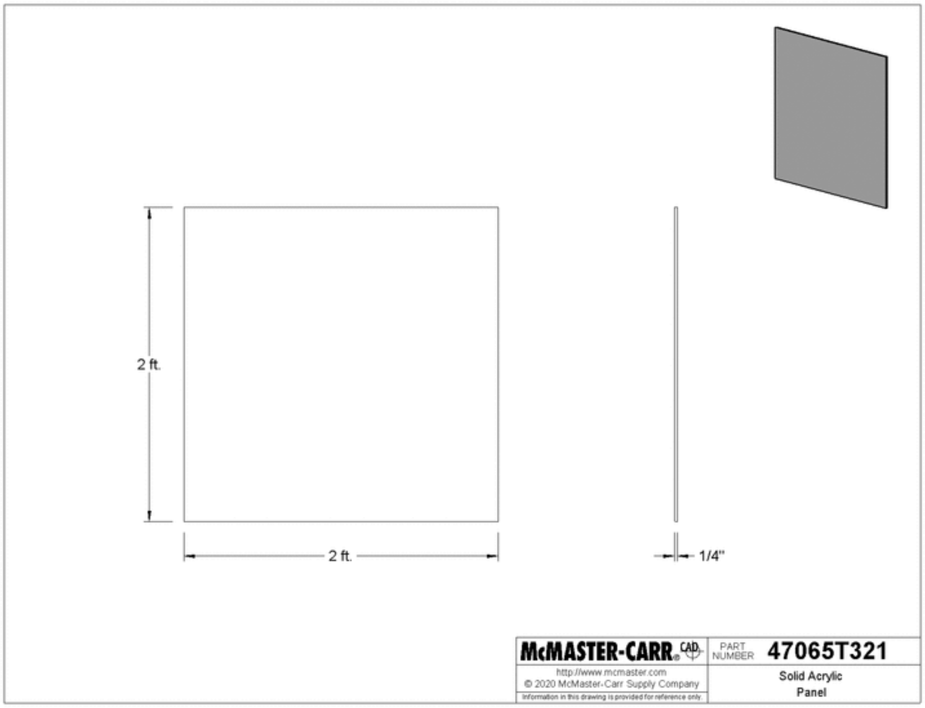


Part 1130 - Motor Mount with Hardware

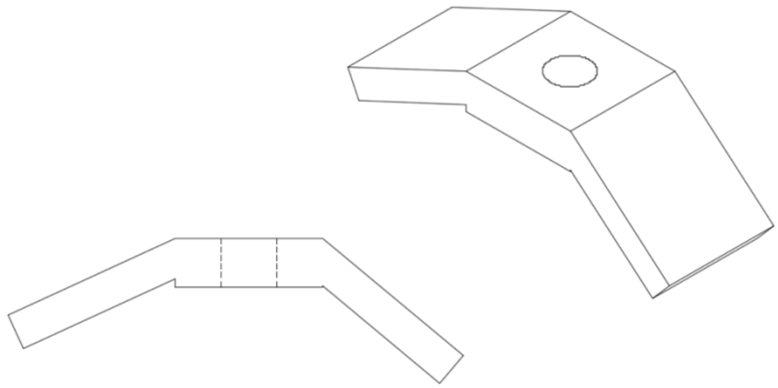




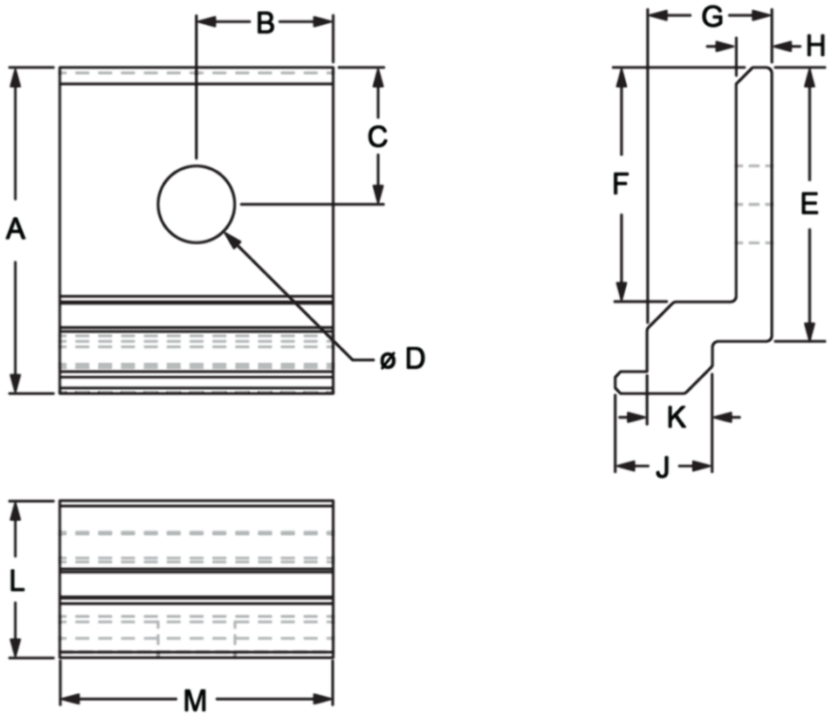
Part 1140 - 8020 Panels



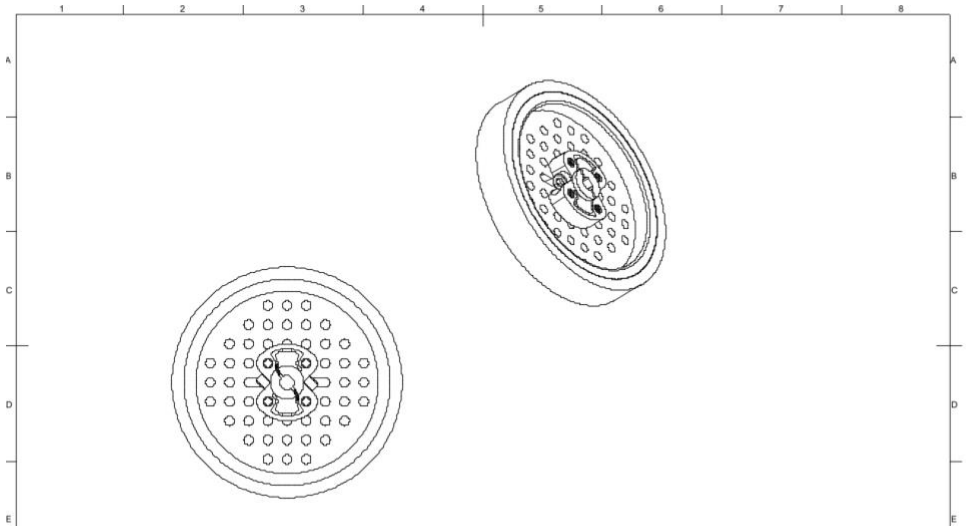
Part 1150 - 3D Printed Basket Attachment



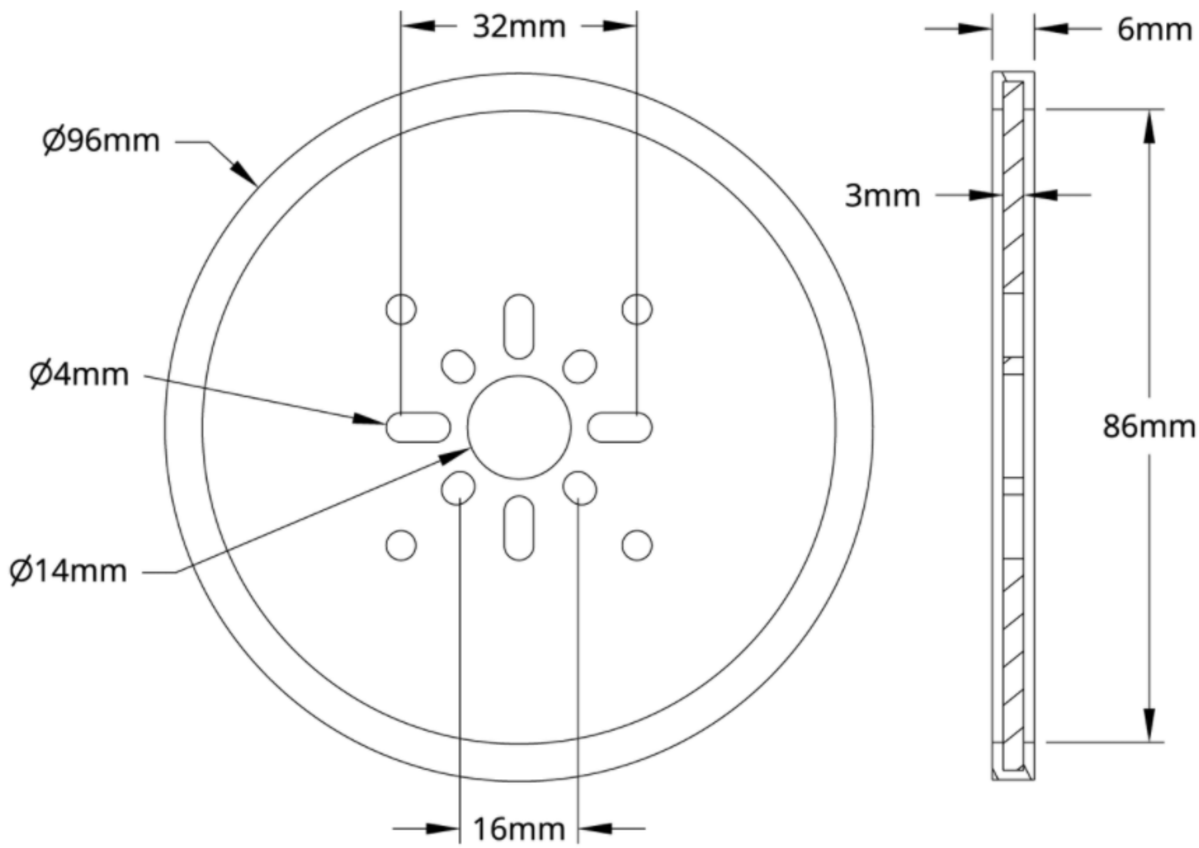
Part 1160 - Roll-in Panel Mounts



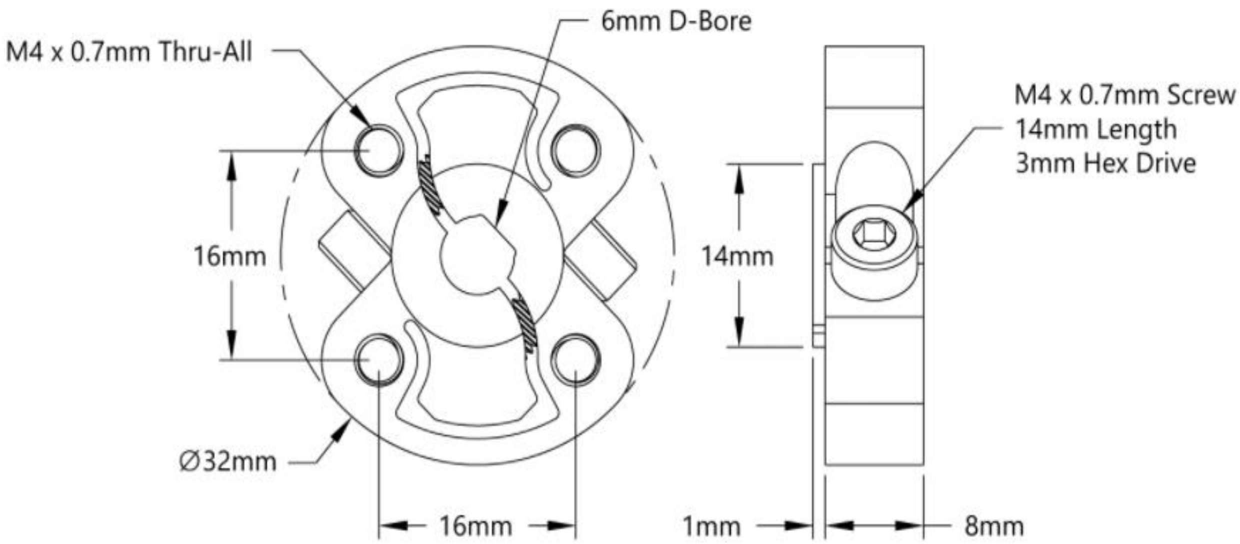
Part 1200 - Wheel Assembly



Part 1210 - Rubber Wheels

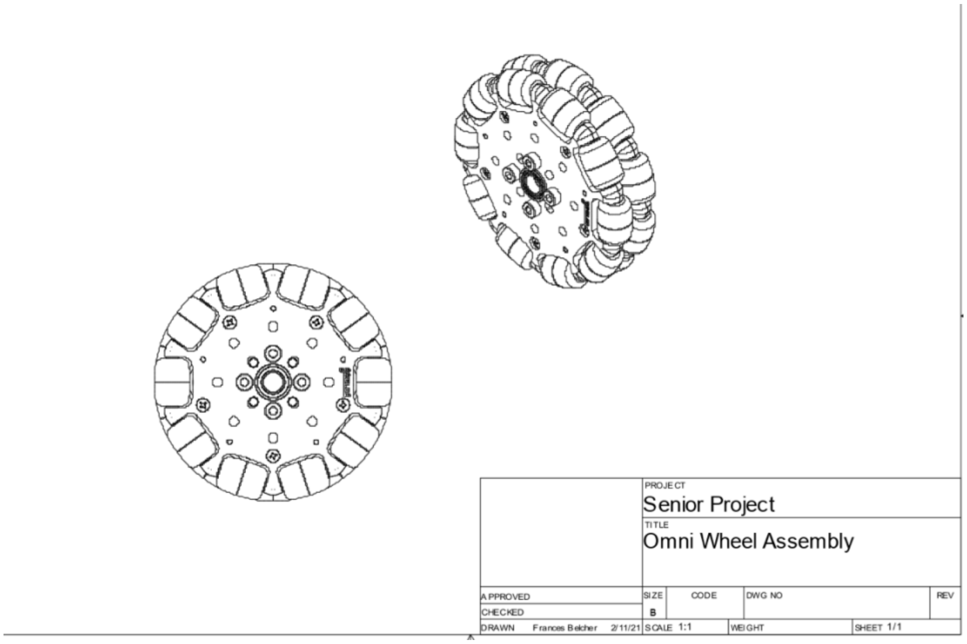


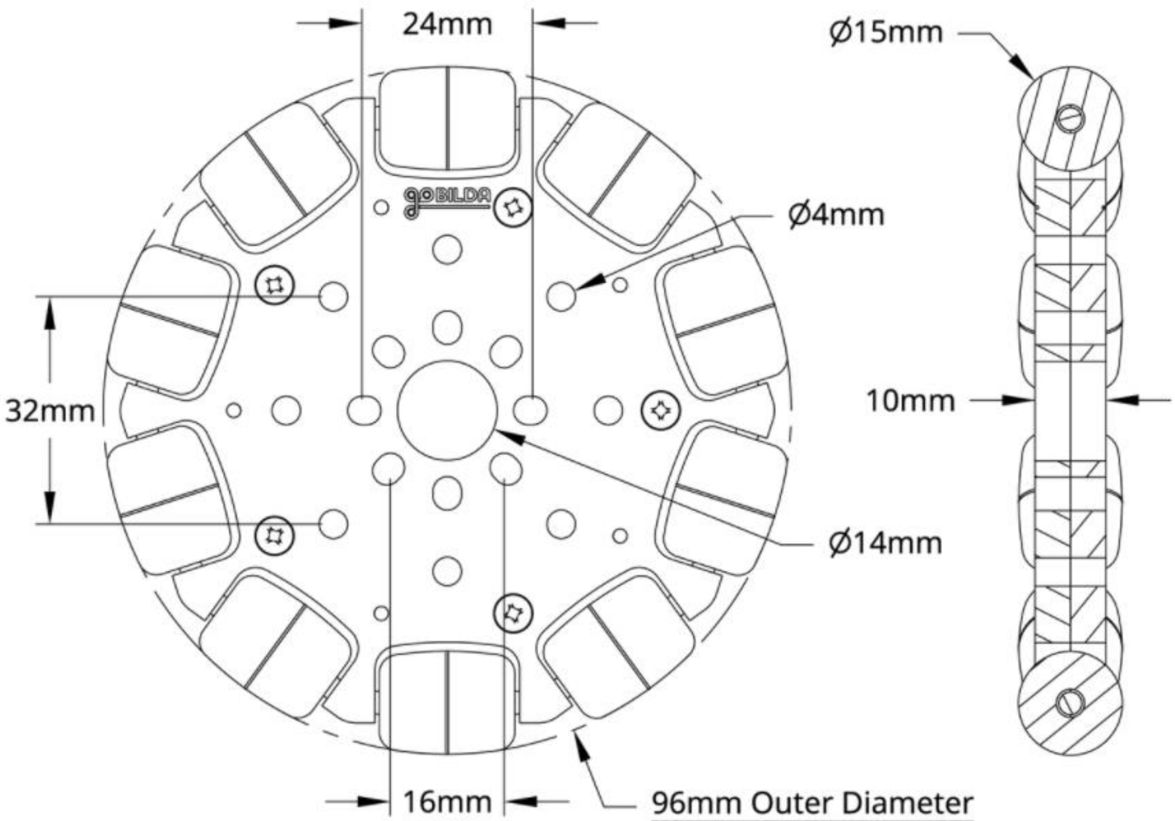
Part 1211 - Sonic Hub



Patents Pending

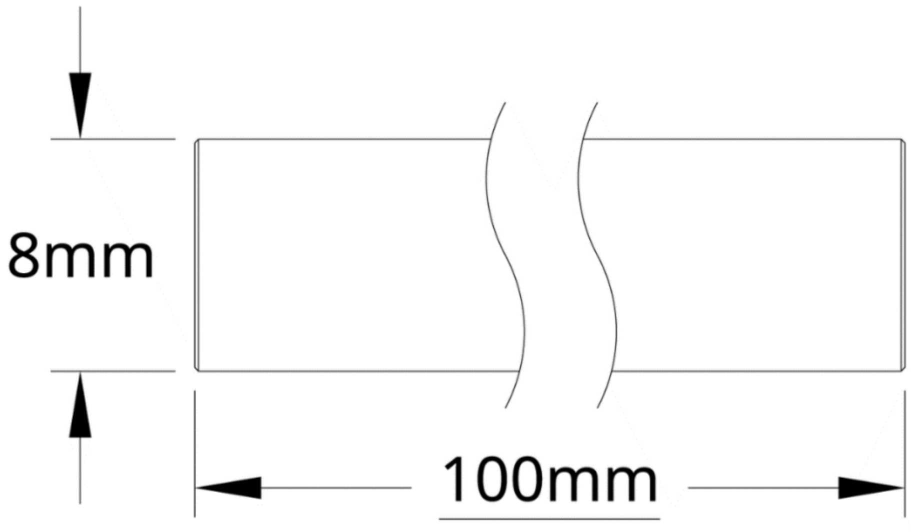
Part 1220 - Omni Wheel Assembly





24mm
Ø15mm
Ø4mm
32mm
10mm
Ø14mm
16mm
96mm Outer Diameter

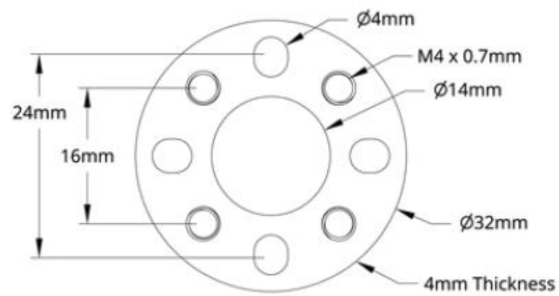
Part 1222 – Shaft



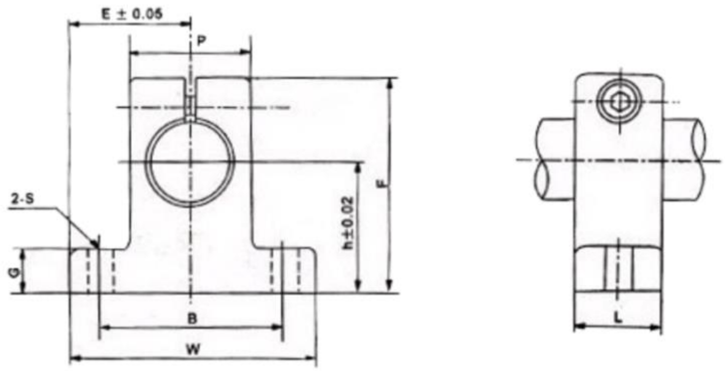
Part 1223 – Spacers

1504 Series 32mm OD Pattern Spacer (4mm Length)

SKU: 1504-0032-0040

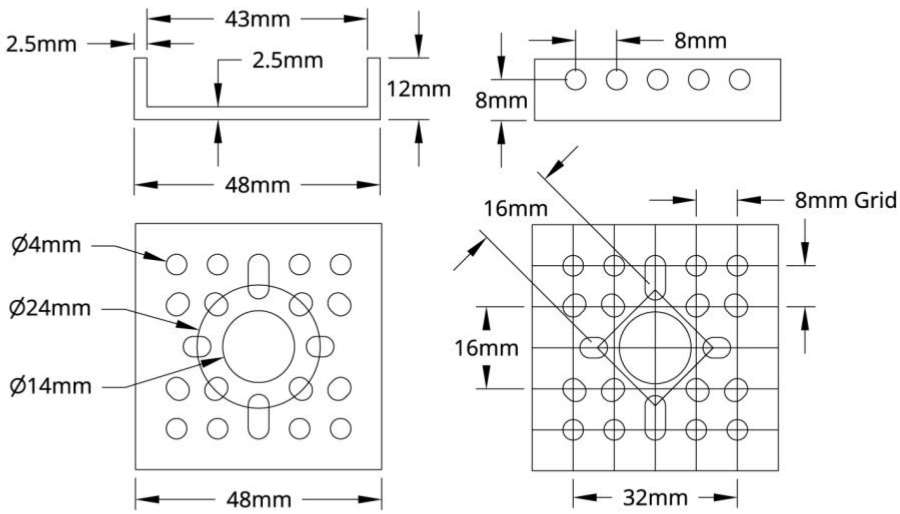


Part 1224 - Linear Rail Shaft Support



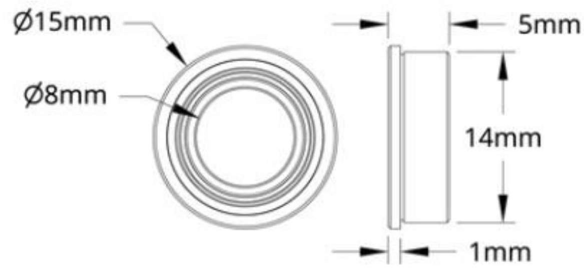
	Rod Dia.	h	E	W	L	F	G	P	B	S	Set Screw	Attach Screws	Weight (kg)
SK8	8	20	21	42	14	32.8	6	18	32	5.5	M4	M5	0.024

Part 1226 - U-block channel

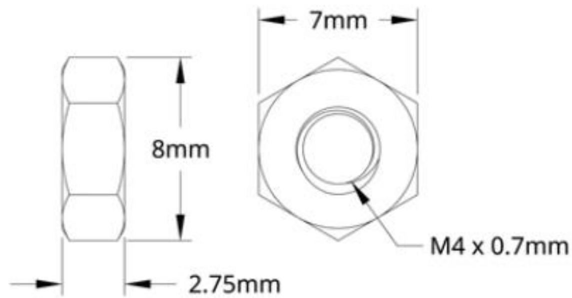


Patents Pending

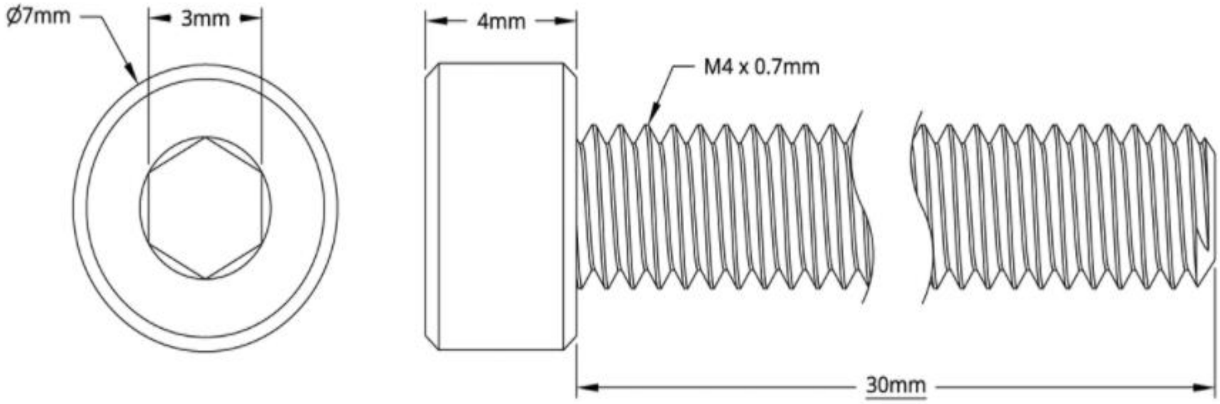
Part 1225 - Flanged Ball Bearings



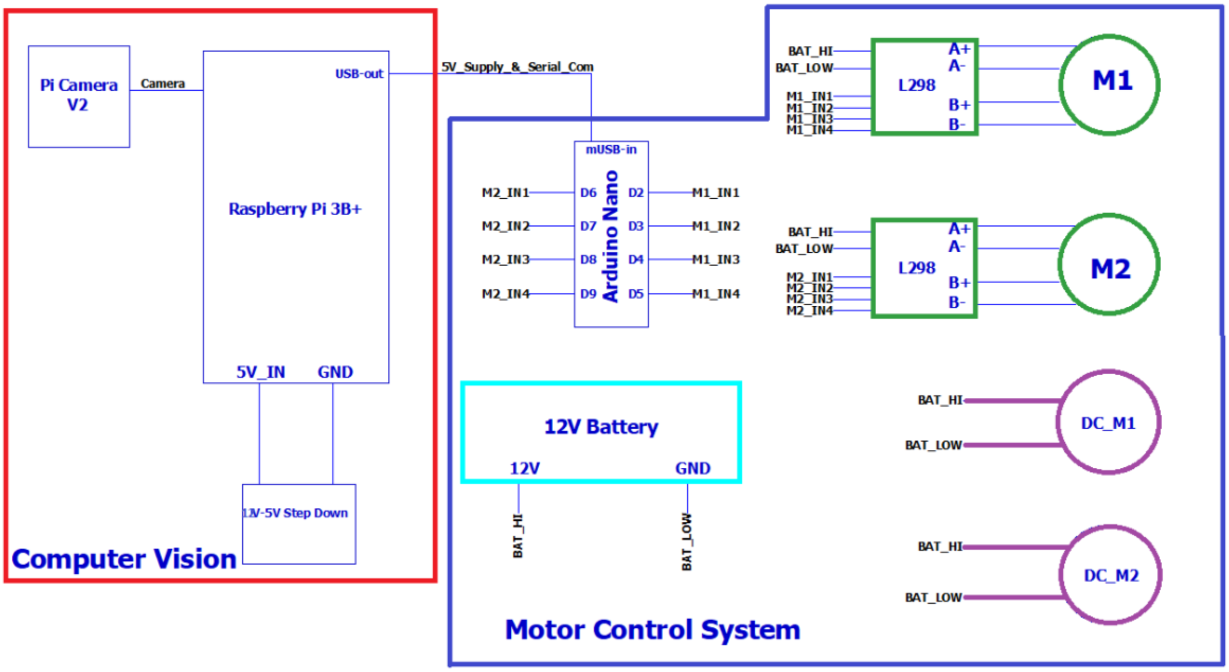
Part 1230 - M4 Fasteners



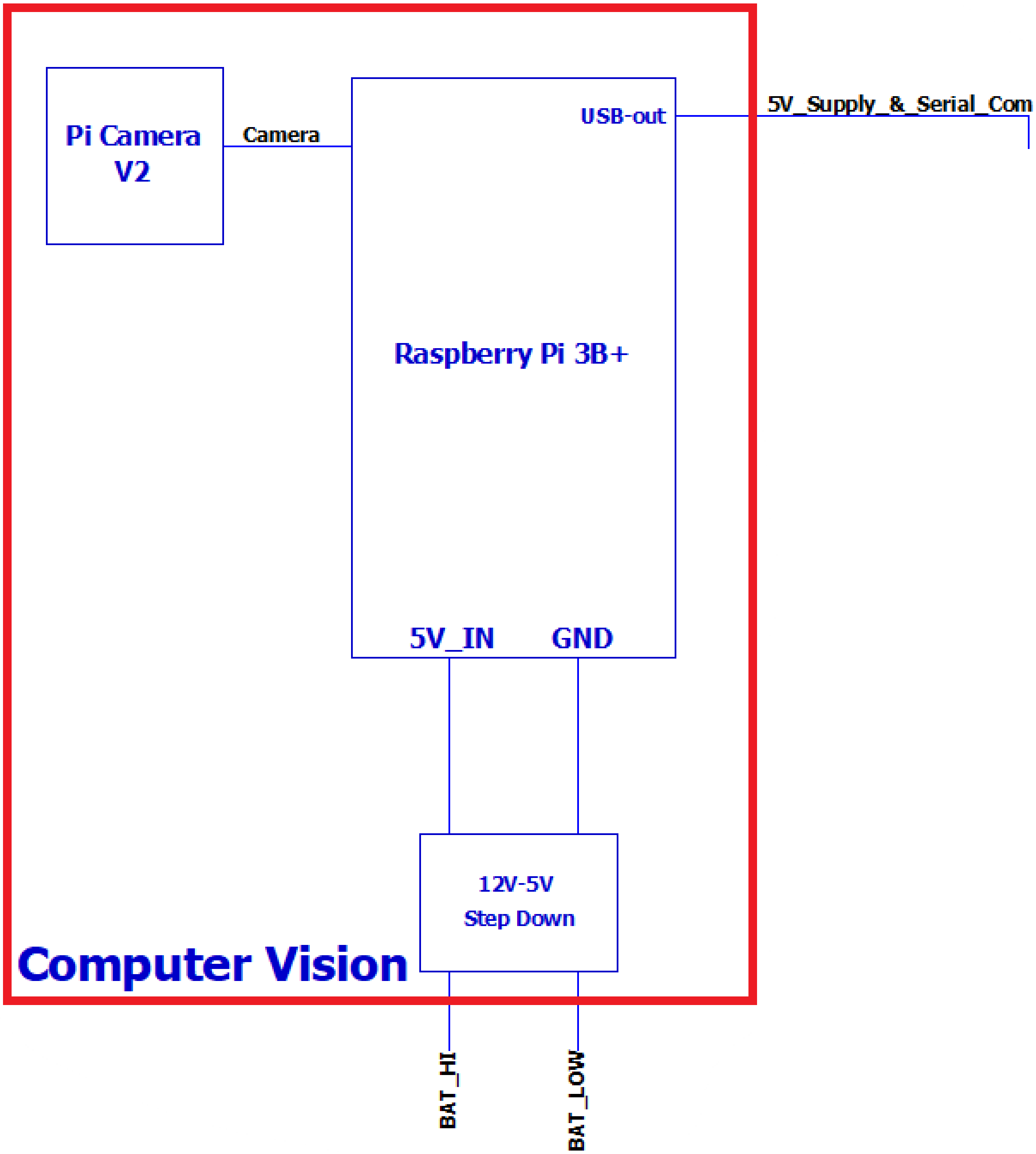
Part 1240 - M4 Nuts



Part 1300 - Electrical Assembly



Part 1310 - Computer Vision and Communication



Overview

The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range, boasting a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE capability via a separate PoE HAT

The dual-band wireless LAN comes with modular compliance certification, allowing the board to be designed into end products with significantly reduced wireless LAN compliance testing, improving both cost and time to market.

The Raspberry Pi 3 Model B+ maintains the same mechanical footprint as both the Raspberry Pi 2 Model B and the Raspberry Pi 3 Model B.

raspberrypi.org

Part 1312 - Camera

Camera Module V2

The Raspberry Pi Camera Module v2 replaced the original Camera Module in April 2016

Buy now

Specification

The Raspberry Pi Camera Module v2 replaced the original Camera Module in April 2016. The v2 Camera Module has a Sony IMX219 8-megapixel sensor (compared to the 5-megapixel OmniVision OV5647 sensor of the original camera).

The Camera Module can be used to take high-definition video, as well as stills photographs. It's easy to use for beginners, but has plenty to offer advanced users if you're looking to expand your knowledge. There are lots of examples online of people using it for time-lapse, slow-motion, and other video cleverness. You can also use the libraries we bundle with the camera to create effects.

You can read all the gory details about IMX219 and the Exmor R back-illuminated sensor architecture on Sony's website, but suffice to say this is more than just a resolution upgrade: it's a leap forward in image quality, colour fidelity, and low-light performance. It supports 1080p30, 720p60 and VGA90 video modes, as well as still capture. It attaches via a 15cm ribbon cable to the CSI port on the Raspberry Pi.

The camera works with all models of Raspberry Pi 1, 2, 3 and 4. It can be accessed through the MMAL and V4L APIs, and there are numerous third-party libraries built for it, including the Picamera Python library. See the Getting Started with Picamera resource to learn how to use it.

The camera module is very popular in home security applications, and in wildlife camera traps.

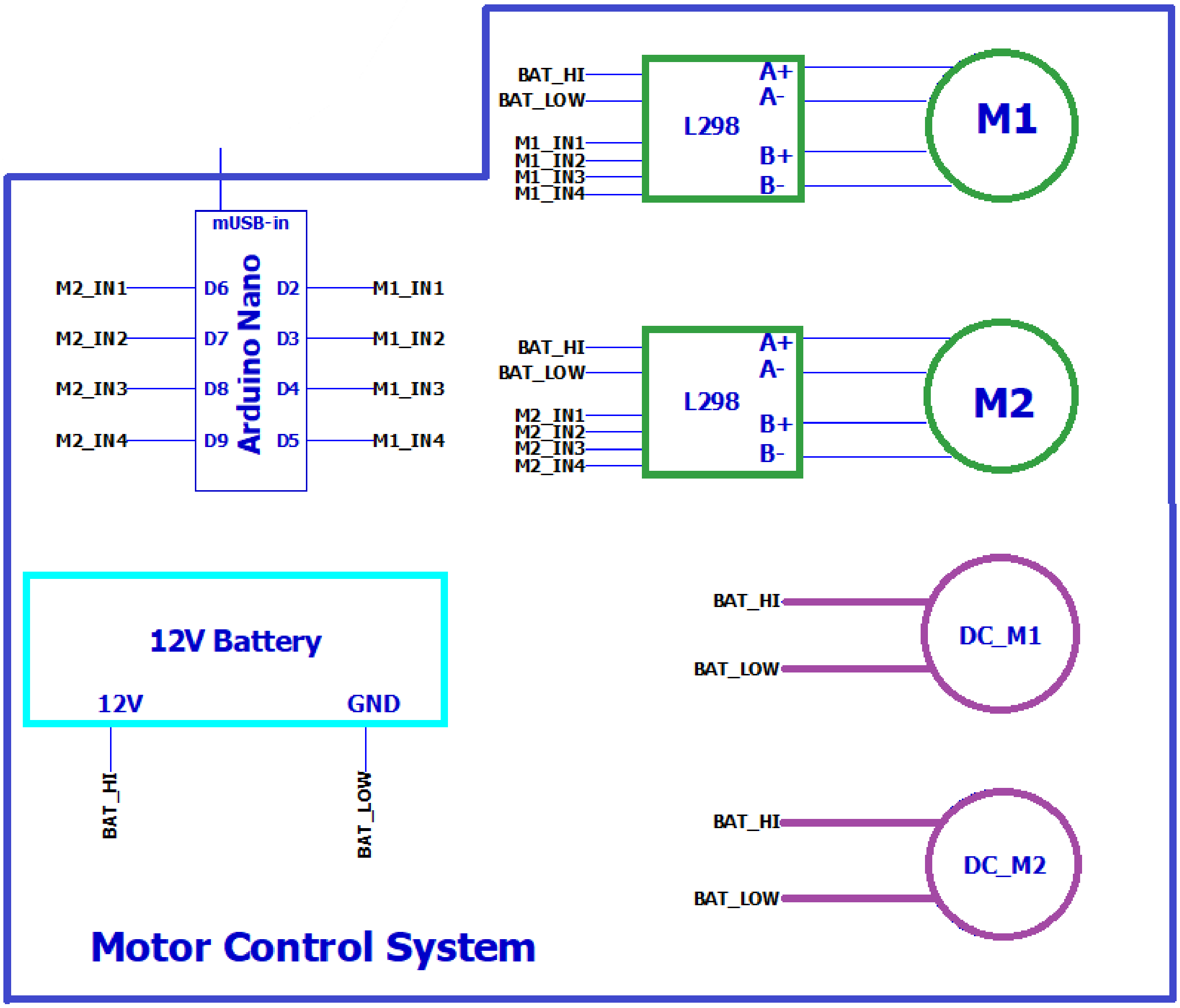
1313 – 12 V to 5 V Stepdown

DC-DC 12V to 5V 3A Micro USB Converter Voltage Step Down Regulator Waterproof Power Converters for Car Smartphone

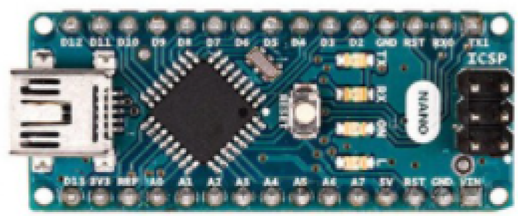
Brand: GLOGLOW

164 ratings | 15 answered questions

Part 1320 - Motor Control System Wiring Diagram



Part 1321 - Arduino Nano





OVERVIEW | TECH SPECS | DOCUMENTATION | FAQ

Microcontroller	ATmega328
Architecture	AVR
Operating Voltage	5 V
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
Analog IN Pins	8
EEPROM	1 KB
DC Current per I/O Pins	40 mA (I/O Pins)
Input Voltage	7-12 V
Digital I/O Pins	22 (6 of which are PWM)
PWM Output	6
Power Consumption	19 mA
PCB Size	18 x 45 mm
Weight	7 g
Product Code	A000005

L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

Multiwatt15

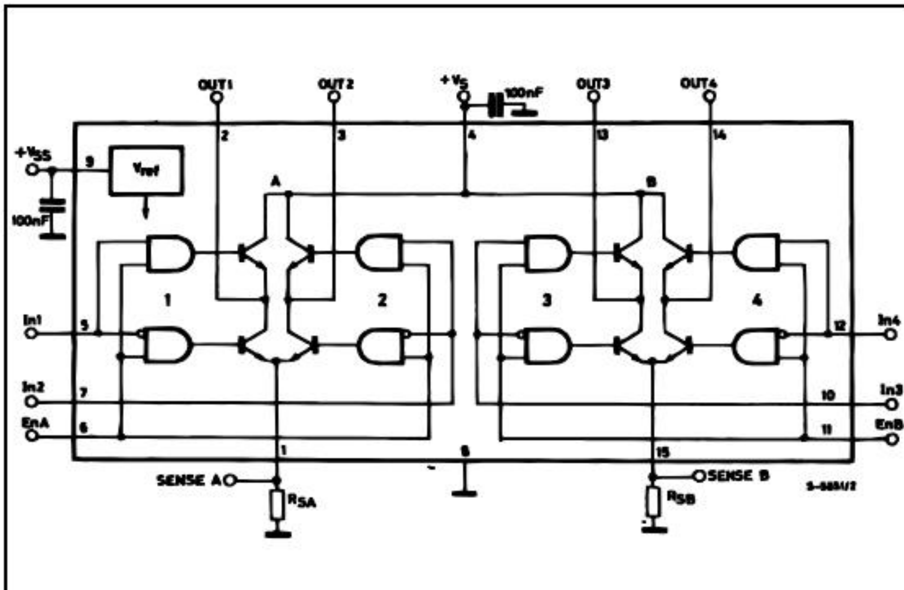
PowerSO20

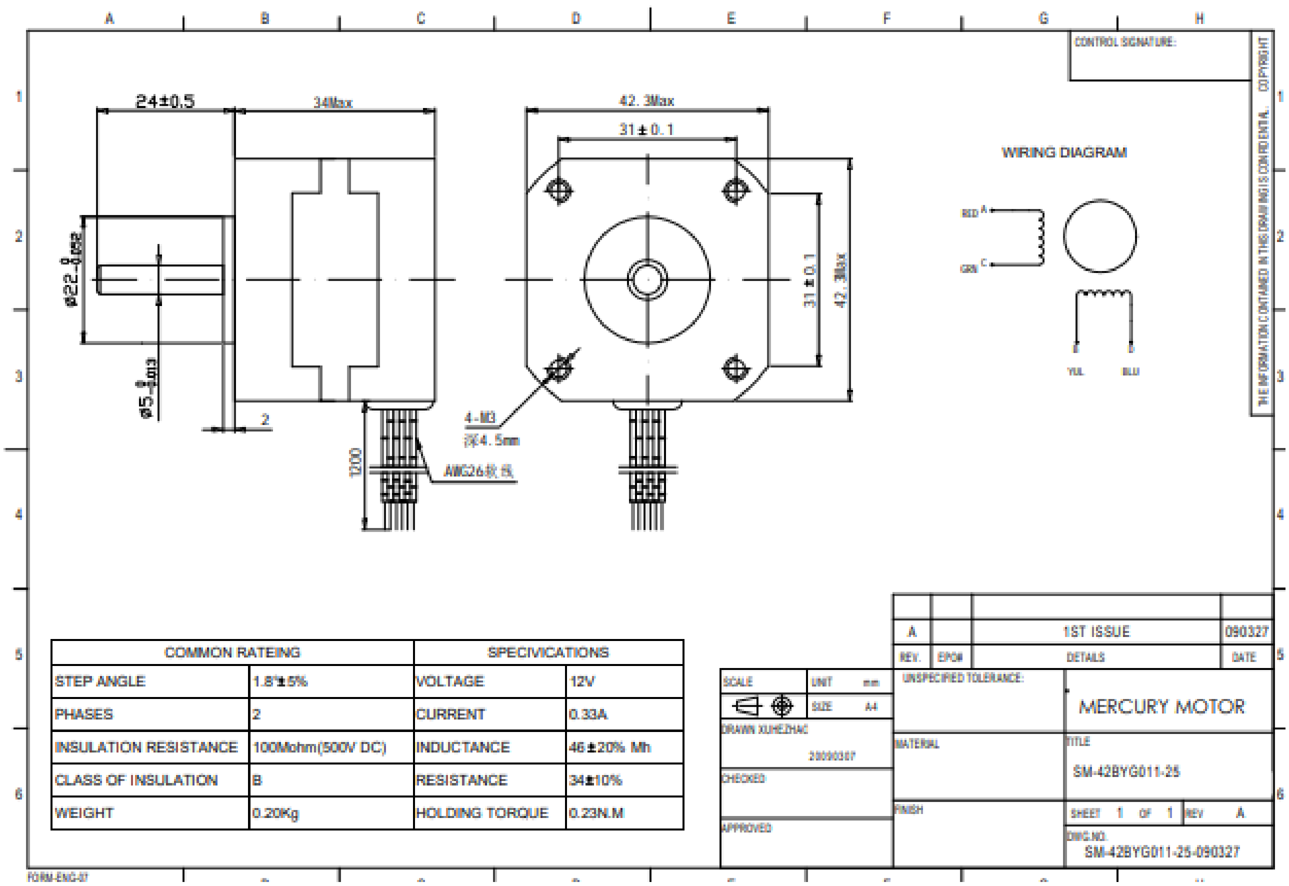
ORDERING NUMBERS : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM





COMMON RATEING		SPECIVICATIONS	
STEP ANGLE	1.8°±5%	VOLTAGE	12V
PHASES	2	CURRENT	0.33A
INSULATION RESISTANCE	100Mohm(500V DC)	INDUCTANCE	46±20% Mh
CLASS OF INSULATION	B	RESISTANCE	34±10%
WEIGHT	0.20Kg	HOLDING TORQUE	0.23N.M

Part 1324 - DC Motors

Roll over image to zoom in

775 DC Motor DC 12V - 24V Max 6000-12000 RPM Ball Bearing Large Torque High Power Low Noise Gear Motor Electronic Component Motor

Brand: SP

24 ratings

Best Deal

Price: $17.88 prime & FREE Returns

Get $70 off instantly: Pay $0.00 ~~$17.88~~ upon approval for the Amazon Prime Rewards Visa Card. No annual fee.

- ★ Type: Mini DC motor; Rated voltage: 12V 24V; No-load speed: 12000 ± 10% RPM;
- ★Reversing: brush; protection function: fully enclosed; certification: CE;
- ★Weight: 340g (APPROX); CAD size reference picture; is_customized: Yes.
- ★Application: printer, rearview mirror / front beam level adjuster, toy car;
- ★ Packaging: 1PCS x DC Motor; one year can be returned if there is a quality problem

Specifications for this item

Brand Name	SP
Ean	0708082091861
Part Number	775 12V 24V Gear Motor
UNSPSC Code	32000000
UPC	708082091861

1325 – Nylon Fastener Set

Hotusi M2 M2.5 M3 M4 M5 (Total 270pcs) Nylon Screw/Nut/Washer Assortment Kit Come with Plastic Box

Brand: Hotusi

80 ratings

Amazon's Choice for "plastic screws"

1326 – Cable Management Set

Part 1330 - Battery Specs

Click on images to enlarge

12V 3,000 mAh NiMH Battery Pack, PP45

Part#: 70018

Related Categories ▾ | ♡Add To Wishlist

☆☆☆☆☆ Write a Review

The Studica 12V 3,000 mAh NiMH (nickel metal hydride) Battery Pack, PP45 comes with a w/20 amp fuse. It is a 10-cell, 12-volt rechargeable battery pac k that brings long-lasting power to motors and electronics that are used in the WorldSkills Mobile Robotics Collection. The Studica NiMH Battery Pack Charger, PP45, is required to charge the battery.

Specifications,

- Dimmensions: 5 x 2.1 x 2.1 inches (115 mm x 45 mm x 50 mm)
- Weight: 1.32lbs (579 g)
- Built-in 20-amp replaceable fuse for safety
- 10-cell pack provides 3,000-mAh capacity for longer life
- PP45 connector for easy and safe connection to controller

Part 1331 - Battery Charger Specs

Click on images to enlarge

NiMH Battery Pack Charger, PP45

Related Categories ▾ ♡Add To Wishlist

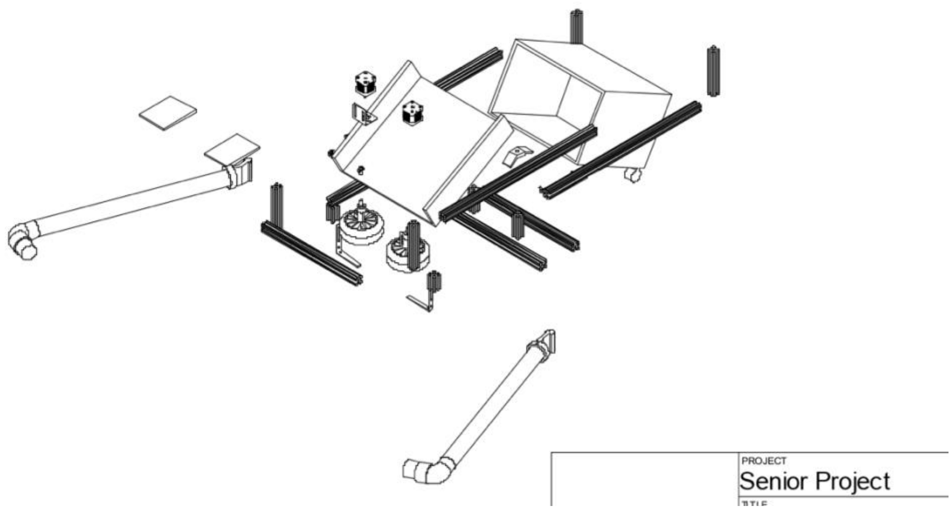
☆☆☆☆☆ Write a Review

The Studica NiMH Battery Pack Charger, PP45, is used to recharge the 12V 3,000 mAh NiMH Battery Pack, PP45. It is a smart charger that auto detects battery pack voltage, sets up proper charging voltage and cuts off power automatically by negative delta V IC when the battery pack is fully charged.

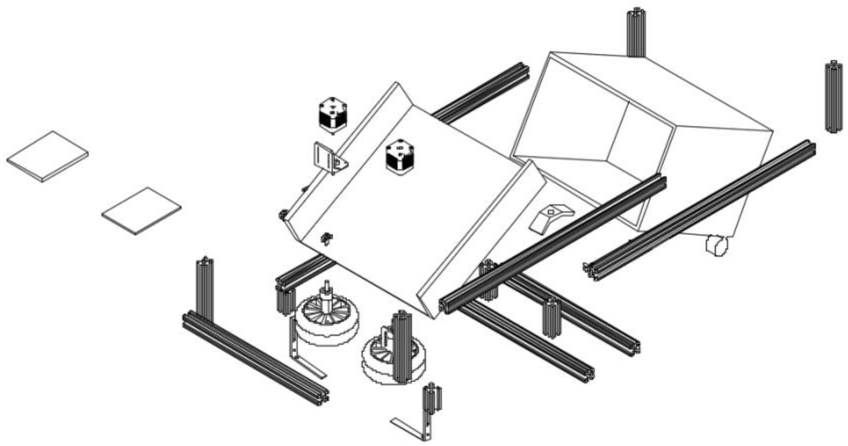
Specification:

- PP45 connector for quick and safe connection to battery pack.
- Input: AC 100-240V
- Output Voltage: 7.2V-12V
- Charging Current: 0.9A/1.8A
- Applicable Battery Types: NiMH/NiCd
- Do not overcharge
  - Disconnect the battery from the charger once it indicates a full charge.
  - Typical charge time does not exceed 2 hours.
  - Do not charge a battery that hasn't been discharged significantly. Running the robot under minimal load for a few minutes will not significantly discharge the battery.
- Minimum no-load voltage: 9.0V
  - Discharging the battery past 9.0V can reduce the lifespan of the battery and can permanently damage the cells.
  - Periodic dips below 9.0V when under load is expected and OK. Don't forget to unplug your battery after you are finished running the robot and don't run your robot until it completely stops responding!
- Operating Temp: 0 ~ 40°C
  - Let the battery cool before and after charging.
  - The battery may feel warm after heavy loading or after charging. This is norma

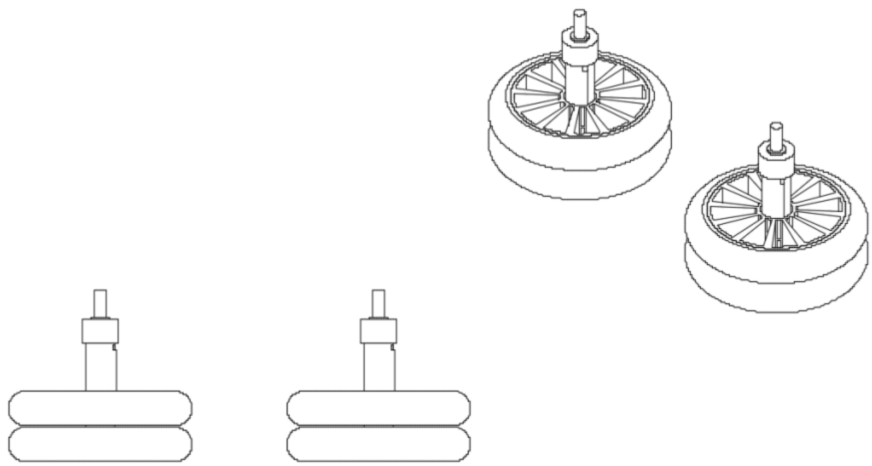
Part 1400 - Collection Assembly



Part 1410 - Cylinder Assembly

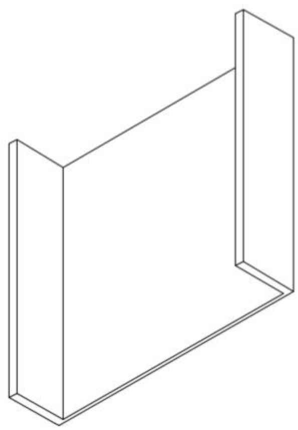


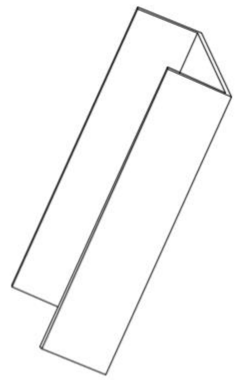
Part 1411 - Rotating Cylinder Wheels



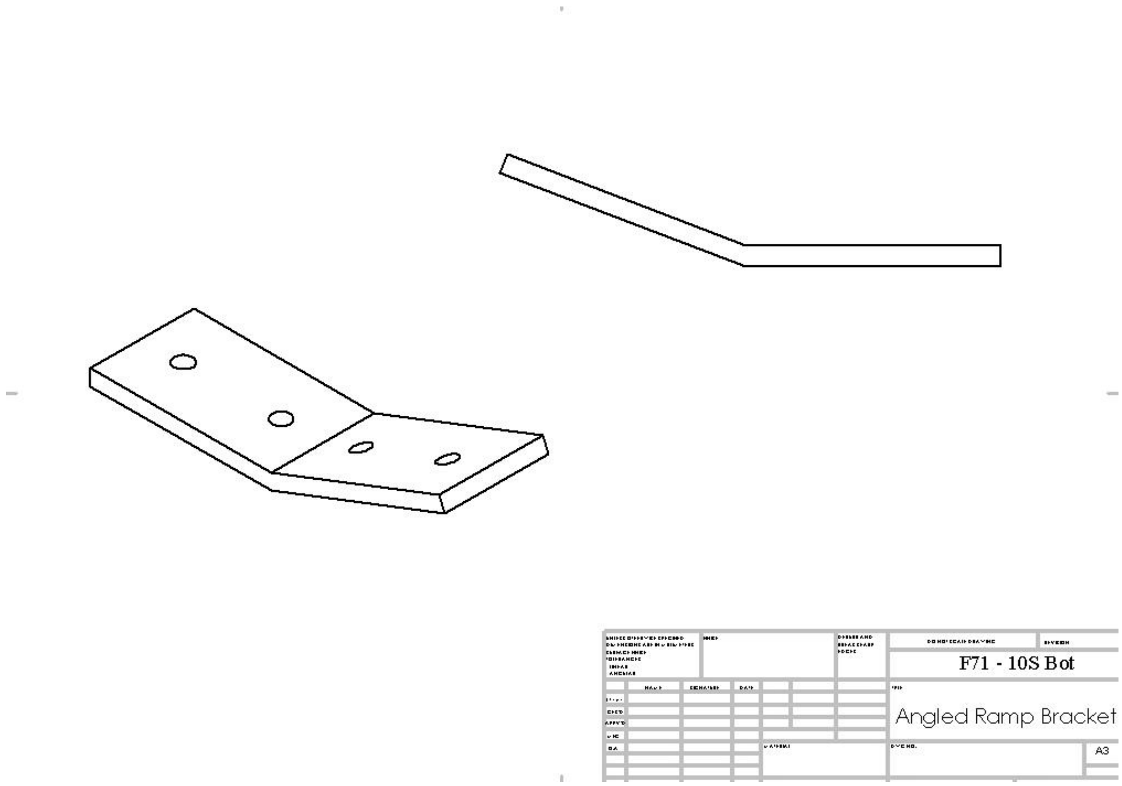
Part 1412 – Basket

Part 1413 – Ramp



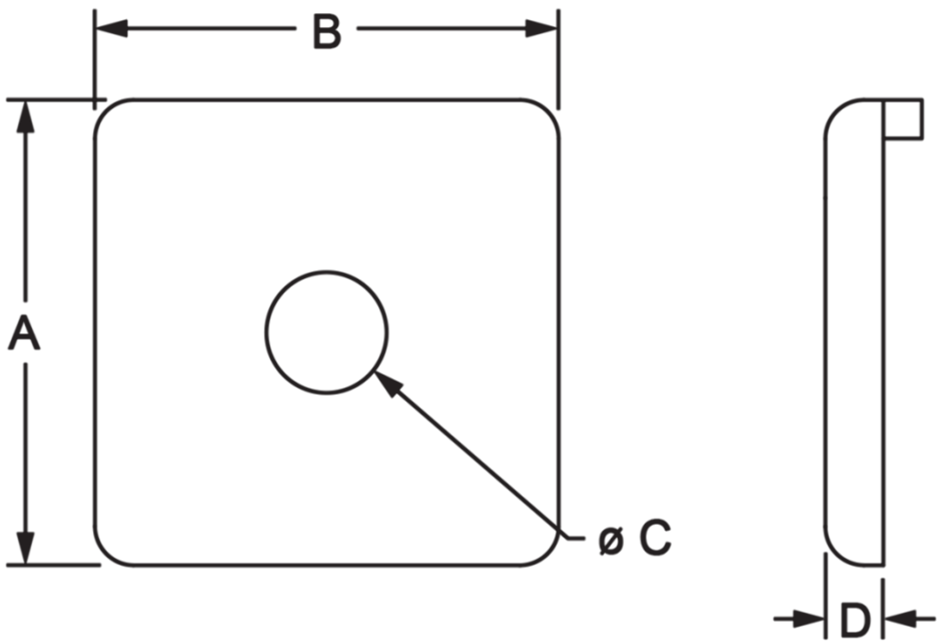


Part 14131 – 3-D printed angled ramp bracket

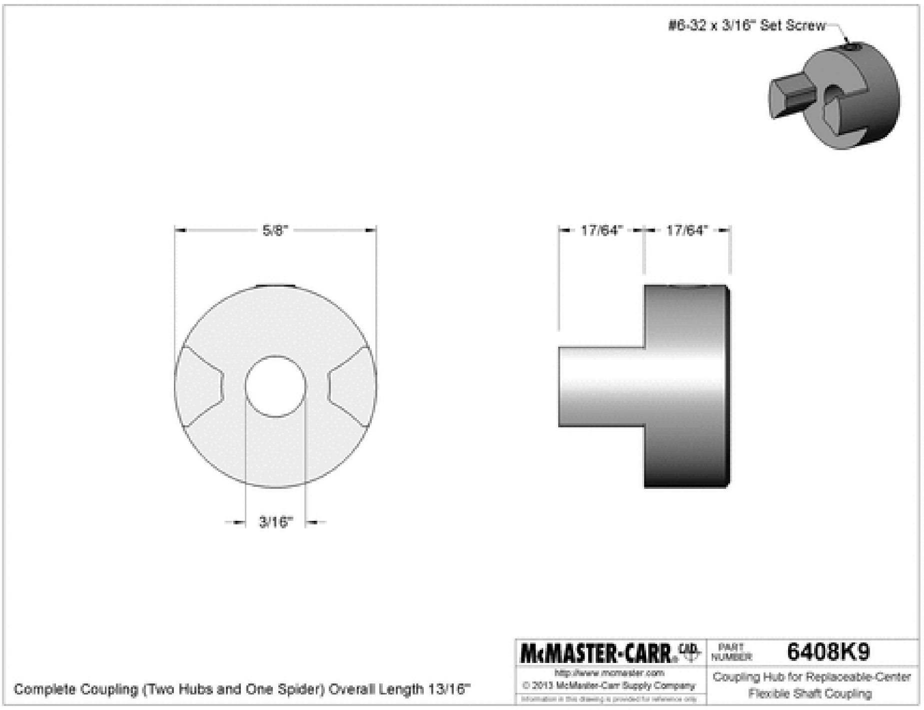


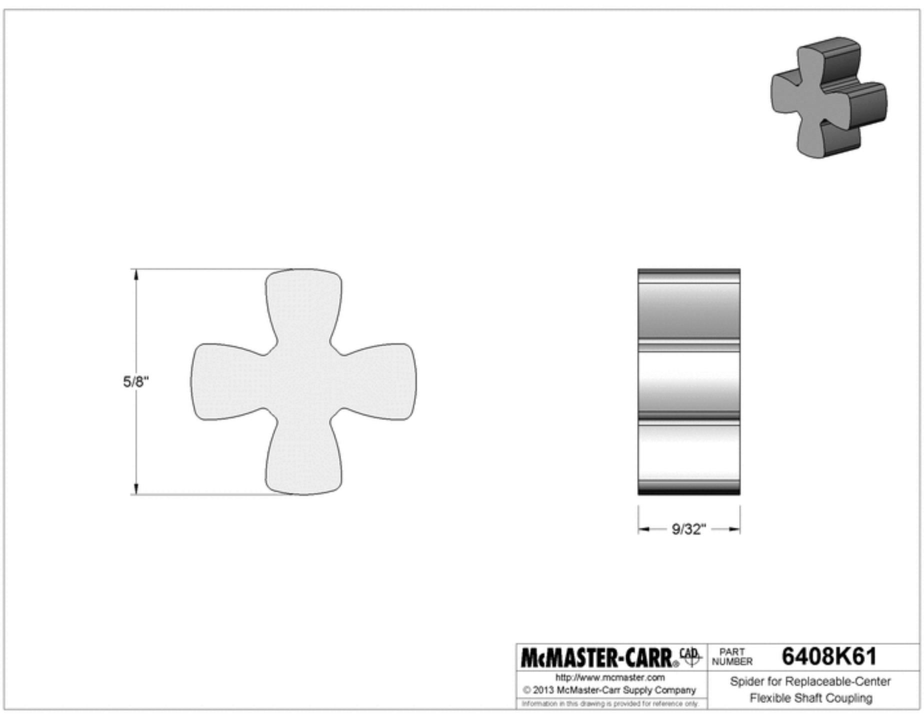
Part 1414 – 1/8 in. x 1 - 1/4 in. Zinc Plated S-Hook

Part 1415 - T-slot Stopper

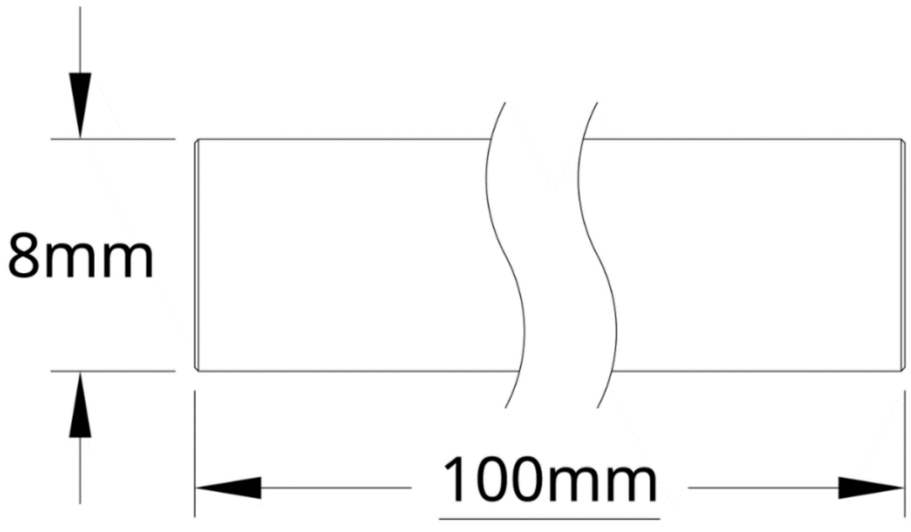


Part 1416 - Set Screw Shaft Coupling

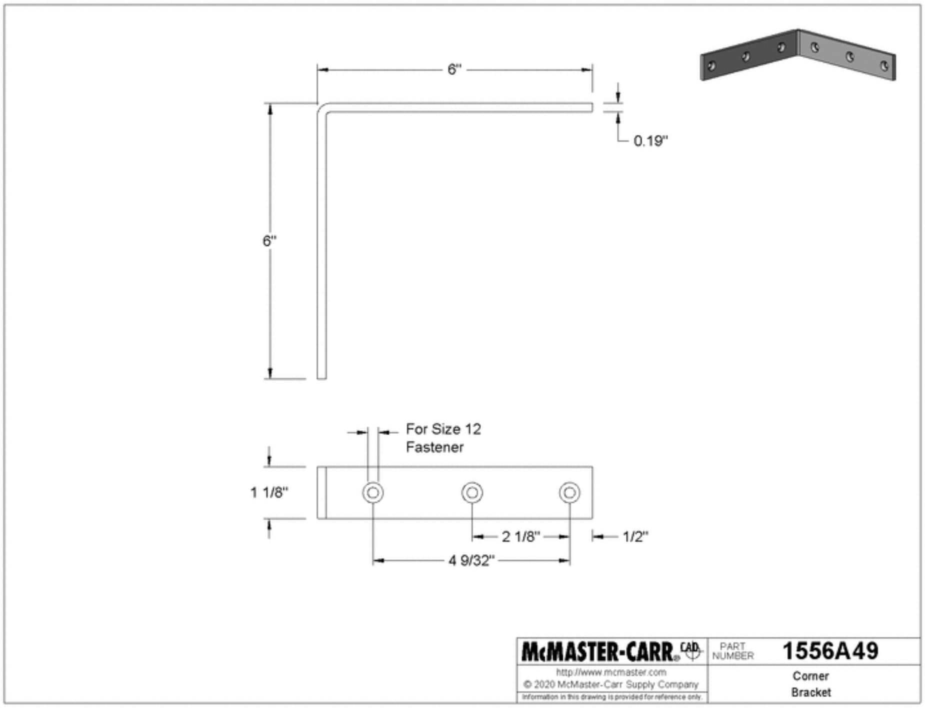




Part 1417 - 8mm Shaft

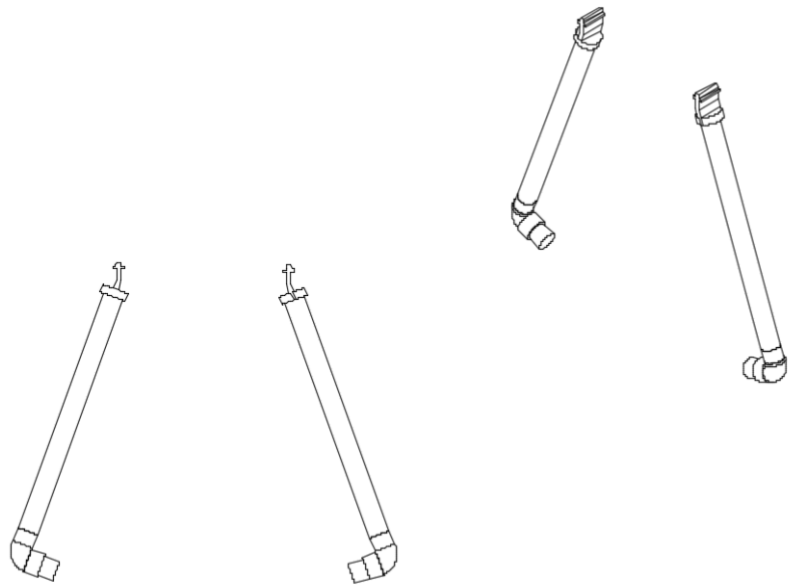


Part 1418 - Corner Bracket

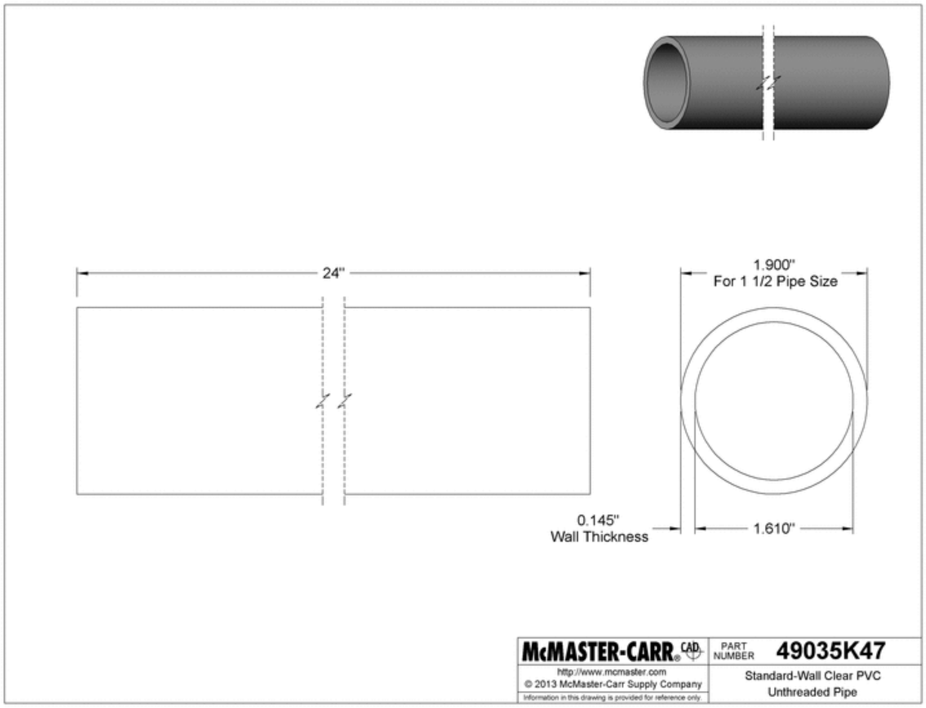


1419 – 2 in. Zinc-Plated Inside Corner Brace

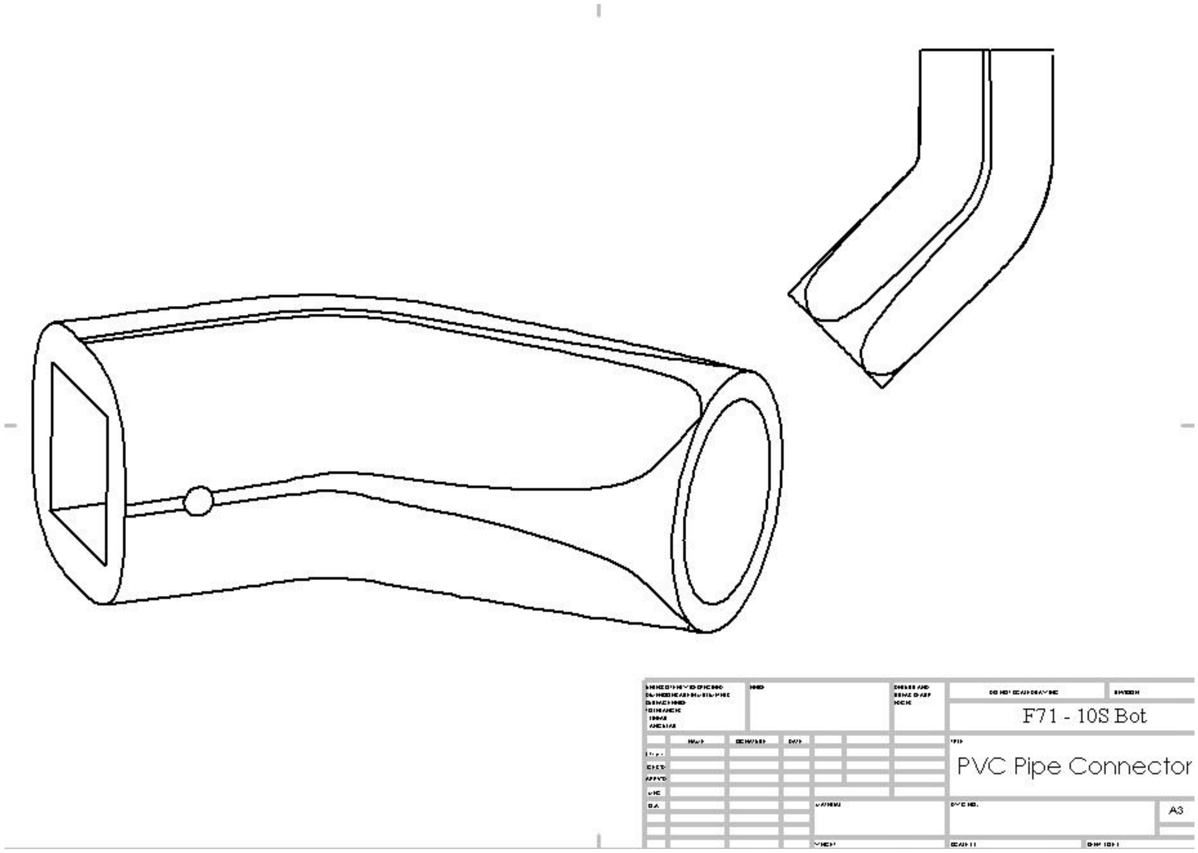
Part 1420 - Arm Assembly



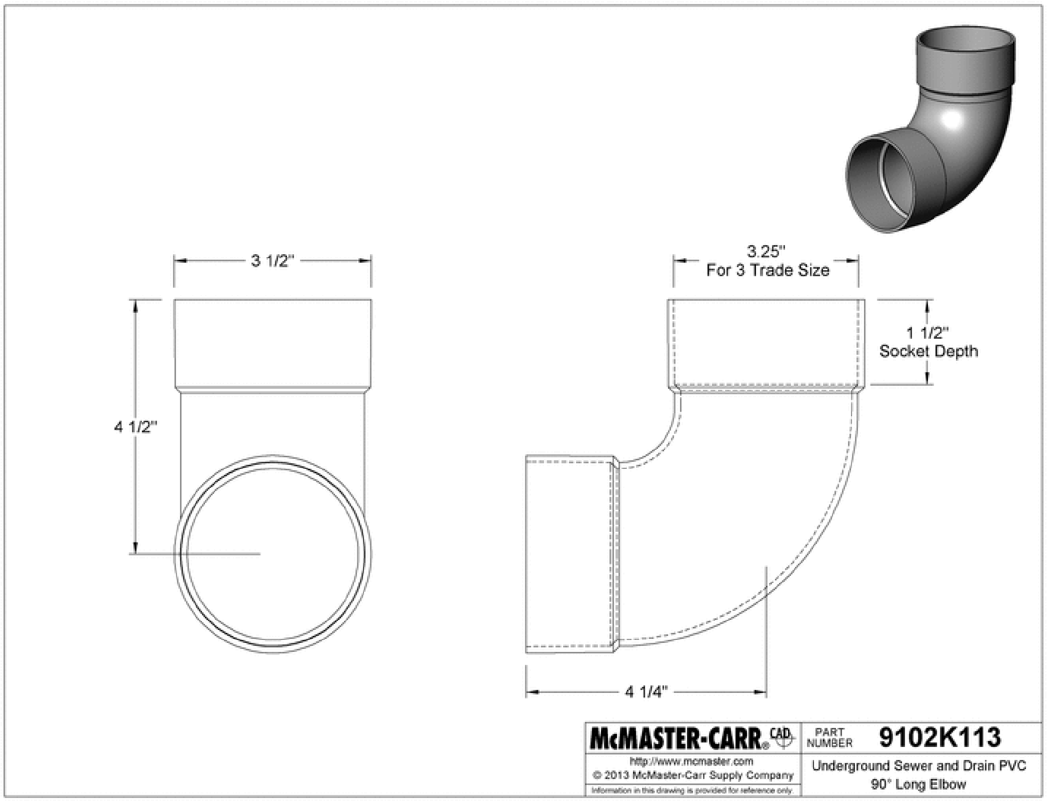
Part 1421 - PVC Piping



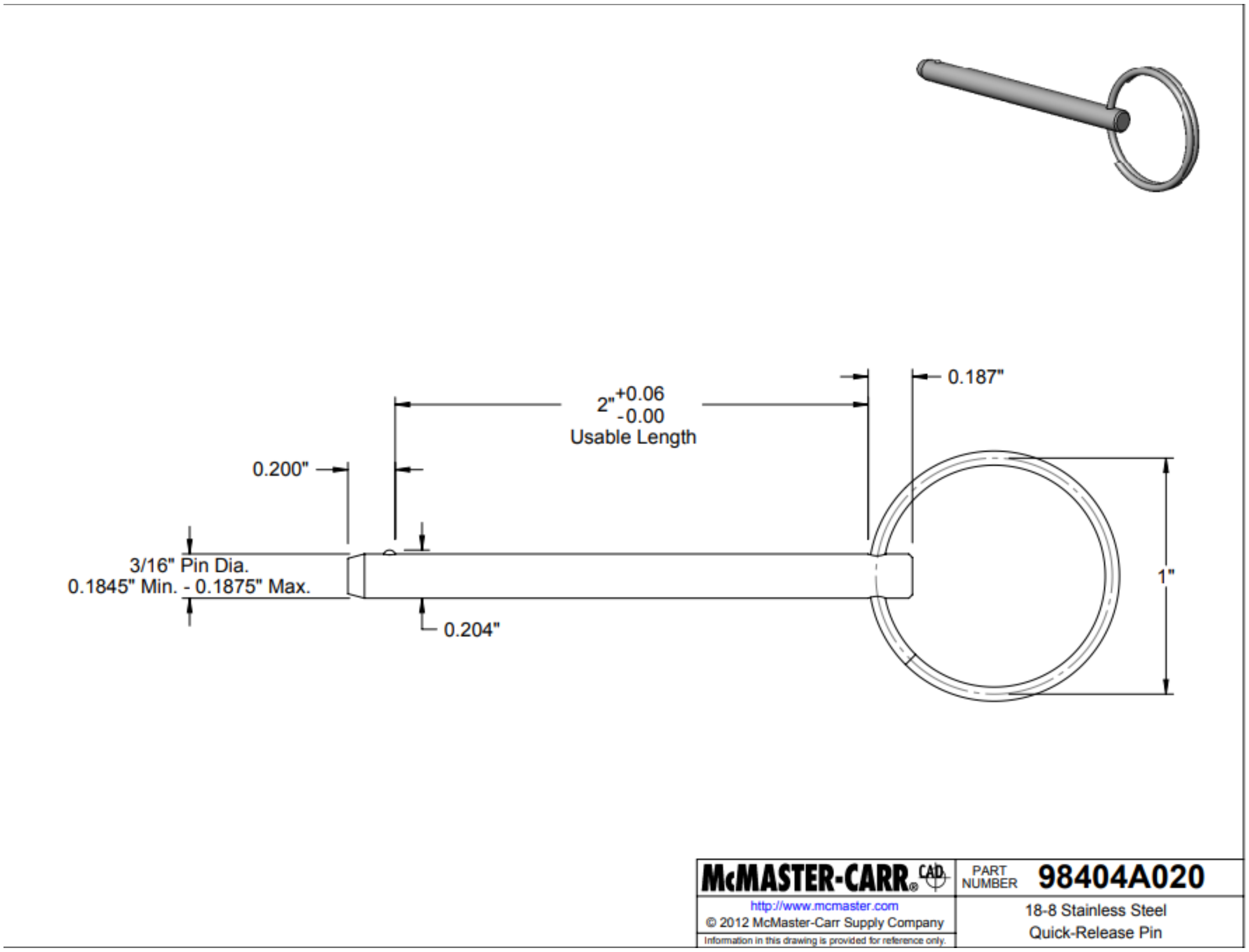
Part 1422 - 3D printed connector



Part 1423 - Curved PVC end piece



1424 - Holding Pin



Appendix [L] – Manufacturing Summary Table

Manufacturing Summary Table

PVC Arms	**1.** Attach curved PVC pipe to the long, straight PVC pipe to create a hook shape. Repeat for the second set of pipes. **2.** 3-D print two attachment pieces from the "PVC Pipe Connector.stl" file. **3.** Attach attachment piece to the end of each long PVC pipe.
Rotating Cylinders	**1.** Using a miter saw for cutting steel, cut the 12" long 10 mm diameter carbon steel rod into two 4" long segments. **2.** Use a belt grinder to smooth out the cut ends of the rods. **3.** Attach two set screw shaft collars back-to-back about 2" from one end of each rod. If necessary, use a belt grinder to decrease the diameter of the shaft collars until each one can fit comfortably in the bearing well of the skate wheels. **4.** Slide a skate wheel onto each shaft collar and snap them into place.
Frame	**1.** Using a miter saw for aluminum, cut available 12" T-slots into four 6" segments. **2.** Use any remaining scraps or 12" T-slots to cut five 3" segments and two 2" segments. **3.** Using a 24" T-slot, drill 3/16" holes 0.5" from one end and 0.5" from the other end. Repeat for a second 24" T-slot. **4.** Repeat step 3. After drilling the second hole, turn the T-slot 90 degrees and drill one more hole 0.5" from one end. Repeat for a second 24" T-slot. **5.** Using a 6" T-slot, drill a 3/16" hole 2" from one end. **6.** Using a 12" T-slot, drill 3/16" holes 4" from one end and 1" from the other end. Repeat for a second 12" T-slot.

	7. Using a 2" T-slot, drill a 3/16" hole 0.5" from one end. Repeat for a second 2" T-slot. **8.** Make ¼ 20" taps on all end profile holes on all T-slots that are 12" long or less
Panels	**1.** Select a table saw for cutting wood. **2.** Cut out nine rectangular pieces from a stock piece of acrylic panel according to the cut list provided in Figure 1 of this Appendix. **3.** The 4"x5" pieces are the left and right flaps of the ramp. On the left flap, drill 11/64" holes at the locations listed in Table 1 of this Appendix with the top left corner as the datum: **4.** On the right flap, drill 11/64" holes at the locations listed in Table 2 of this Appendix, with the top left corner as the datum: **5.** On the left flap, drill a 13/32" hole at (3 ¼", 1 ¼") using the same top left datum. **6.** On the right flap, drill a 13/32" hole at (4 3/16", 2 13/16") using the same top left datum.
Computer Vision and Motor Control	**NOTE:** When assembling the electrical components, refrain from connecting components to the 12V battery source until fully wired. **1.** Wire the 12V motors to the motor controllers. **2.** Wire the motor controllers to the Arduino Nano. **3.** Attach the camera module to the Raspberry Pi's dedicated camera ribbon-cable input. **4.** Connect the Arduino to the Pi using the USB-B to mini-B cable. **5.** Once all these connections have been made, wire the power and ground cables to each component.

	6. Plug in the battery to the system power distribution cable.
Omni Wheels	**1.** Thread the M6 holes on the shaft support. **2.** Attach the 8mm steel shaft to the shaft support and use the set screw to tighten it in place. **3.** Attach motor mounts and motors to the side of the chassis frame using M4 30 mm screws. **4.** Assemble the Omni wheel by: **a)** Placing one Omni wheel on top of the sonic hub with the center holes aligned. **b)** Placing one pattern spacer on top of the previous Omni wheel with the center hole aligned. **c)** Placing a second Omni wheel on top of the spacer, at a 90-degree rotation from the first Omni wheel. Ensure the center holes are aligned. **d)** Fasten the wheel together using four M4 30mm screws in a square pattern. **5.** Attach the assembled wheel to the motor shaft and tighten the sonic hub.
Rubber Wheels	**1.** Attach shaft mount onto the side of the chassis frame using M4 30 mm screws. **2.** Fasten ball bearing into the center bore of the rubber wheel. **3.** Press-fit the steel shaft into the center of the ball bearing. **4.** Attach the opposite end of the shaft onto the shaft support. **5.** Repeat for each wheel.

Panels Manufacturing Summary

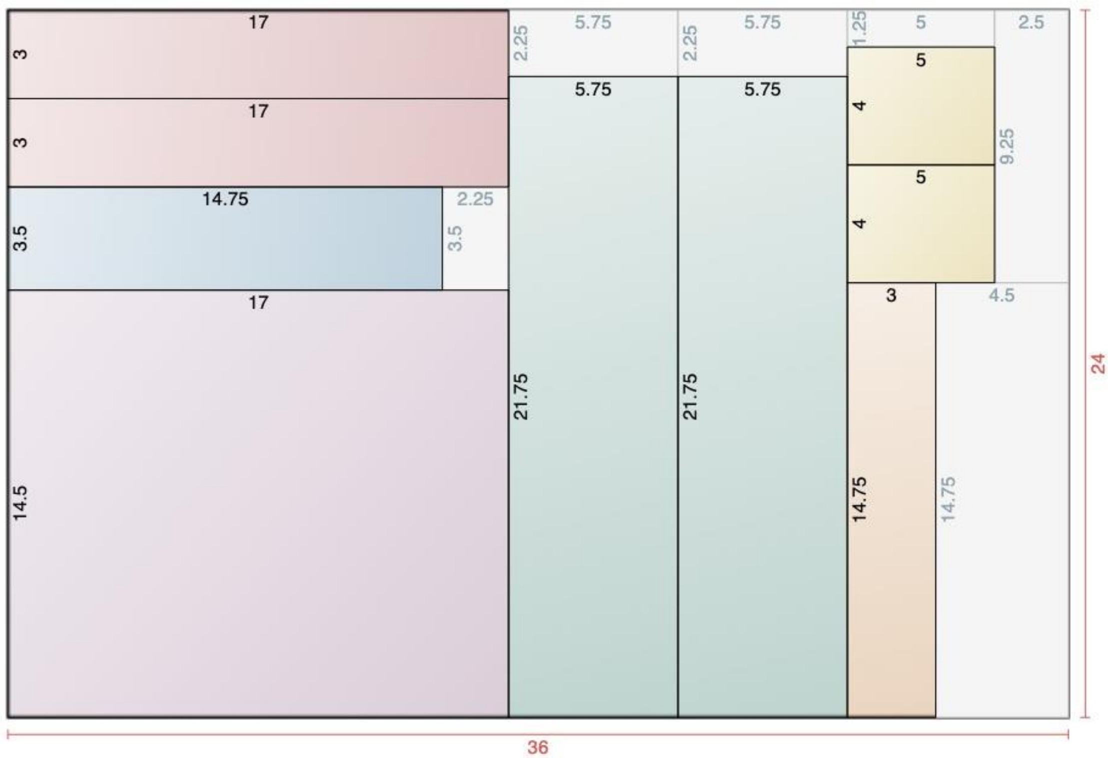


Figure 1: Cut list for the acrylic panels.

Table 1. Hole alignments for left ramp flap

Hole #	Horizontal Alignment (inches)	Vertical Alignment (inches)
1	7/8	5/16
2	7/8	1 3/16
3	31/32	2 11/16
4	31/32	3 7/16
5	1 15/16	3 1/16

Table 2. Hole alignments for left ramp flap

Hole #	Horizontal Alignment (inches)	Vertical Alignment (inches)
1	15/16	3 13/16
2	15/16	2 15/16
3	1 3/8	1
4	1 3/8	1 11/16
5	2 3/16	1 11/32

Appendix [M] Completed Test Procedures

F71 – Autonomous Tennis Ball Collector

Test Name: Test for Specification 1 – Weight
Purpose: The purpose of this test is to get an accurate measurement of the robot's total weight, without carrying any tennis balls.
Scope: This test will help us better estimate the total weight of the robot. This will help us better understand the weight distribution, as an equal distribution is needed for the omni wheels.
Equipment: Weight scale
Hazards: Crushing hazard while lifting robot
PPE Requirements: None
Facility: Cal Poly's HVAC Lab for the large scale located within
Procedure:

1. Fully assemble the robot and attach all components.
2. Place the robot on the scale and record the weight.
3. Remove the robot from the scale then repeat step 2 to obtain a second reading.

Results: This test will be conducted a total of five times to gather an approximate estimate on the weight of the robot. The criteria we will be checking for is the weight reported from the scale.
Finish Date: May 10th
Test Results: See Table 1.

Table 1: Measured Weight for each Trial

Test #	Measured Weight (lbf)
1	24.5 ± 0.25
2	24.0 ± 0.25

Performed By: Any able-bodied person able to lift up to 25 pounds.
Run By: Frances and Matthew

Test Name: Test for Specification 4 – Time to move collected balls to serving basket.
Purpose: The purpose of this test is to get an accurate estimate of the time it takes for the operator to transport a full load of tennis balls from the collection basket to the serving basket.
Scope: This test will help us better understand the difficulties associated with removing and reattaching the collection basket as well as the intuitiveness of the design.
Equipment: 72 tennis balls, collection basket.
Hazards: Pinch hazard when removing and attaching collection basket.
PPE Requirements: None
Facility: None
Procedure:

1. Load 72 tennis balls into tennis ball collector's basket.
2. Attach basket onto the robot.
3. Start timer and have the user remove the storage basket from the robot and prop up the basket's legs.
4. Stop the timer as soon as the basket is securely propped up. Record the elapsed time.
5. Repeat steps 1-4 with a total of five different users to compute the average time.

Results: This test will be conducted a total of five times to gather an approximate estimate on the duration required to transport balls from the collection basket to the serving basket.
Finish Date: May 24th
Test Results: See Table 1.

Table 1: Elapsed Time of Each Test Run

Test #	Elapsed Time (seconds)
1 (Thalia)	00:53.20
2 (Kahye)	00:23.94
3 (Michael)	00:18.86
4 (Sudeep)	01:02.43
5 (Galen)	00:22.15
Average Time	**00:36.12**

Performed By: Frances and Matthew

Test Name: Test for Specification 5 – Time spent operating device
Purpose: The purpose of this test is to get an understanding of the intuitiveness of the robot's design
Scope: This test will help us better estimate if individuals, who have no prior experience with the robot, are able to setup and operate the device in a reasonable amount of time.
Equipment: Stopwatch
Hazards: None
PPE Requirements: None
Facility: None
Procedure:

1. Lay the robot on the ground with the basket and PVC arms detached and layed out separately.
2. Start timer. Have the user attach the basket and PVC arms, followed by commanding the robot to collect tennis balls through the phone app.
3. Stop the timer. Record the elapsed time.
4. Repeat steps 1-3 with a total of five different users to compute the average time.

Results: This test will be conducted a total of five times to gather an approximate estimate on intuitiveness of the robot's design. Time lengths of over 5 minutes indicates identifying markers will be required to guide the setup of the robot.
Finish Date: May 24th
Test Results: See Table 1.

Table 1: Elapsed Time of Each Run

User #	Elapsed Time (seconds)
1 (Thalia)	02:33.95
2 (Kahye)	01:24.76
3 (Michael)	01:42.40
4 (Sudeep)	03:45.20
5 (Galen)	04:16.41
Average Time	**02:44.54**

Performed By: Frances and Matthew
Testing notes:
users figured out the hooks pretty easily.
Suggestion to keep the hooks attached to the basket instead of the robot

Test Name: Test for Specification 8 – Ability to Collect Tennis Balls
Purpose: The purpose of this test is to determine how effective the robot is at gathering, transporting, and storing the tennis balls.
Scope: This test will help us determine the effectiveness of the ball collection subsystem when the robot is in motion. It involves moving the robot around a court full of tennis balls (either by hand or through automation, depending on the full robot functionality at the time of this test), and counting the number of balls that are successfully shot through the spinning cylinders, up the ramp, and into the basket. Each ball will be classified by how far it made it through the collection process: into the basket, onto the ramp, or never made it through the cylinders.
Equipment: 72 tennis balls
Hazards: Tripping hazard while robot is in motion.
PPE Requirements: None
Facility: SLO High School tennis courts - 1499 San Luis Dr, San Luis Obispo, CA 93401
Procedure:

1. Scatter all 72 tennis balls around one side of the tennis court.
2. Activate the robot to cause the rotating cylinders to start spinning.
3. Push the robot around the court to collect the tennis balls.
4. When a tennis ball is collected or attempted to be collected, use a tally mark to record the ball's progression in its corresponding column in Table 1.
5. Repeat until all 72 tennis balls are collected or attempted to be collected.
6. Repeat steps 1-5 a second time.
7. Deactivate the robot.

Results: We will run the test 2 times with 72 tennis balls to get enough data to feel confident with our results. The goal is to get 95% of tennis balls securely stored in the basket. If this goal is not reached, then necessary adjustments will be made to improve the effectiveness of the design.
Finish Date: May 25th
Test Results: See tables.

Table 1: Raw Data of Ball Collection Effectiveness

Test #	Ball stored securely in basket	Ball went through cylinders but could not make it up the ramp	Ball could not pass through cylinders
1	10	5	0

Performed By: Frances and Matthew

Appendix [N] - Design Verification Plan & Report

DVP&R - Design Verification Plan (& Report)

Project:	F71-10S Bot	Sponsor:	John Chen	Edit Date:	5/25/21

TEST PLAN											TEST RESULTS	
Test #	Specification	Test Description	Measurements	Acceptance Criteria	Required Facilities/Equipment	Parts Needed	Responsibility	Test Plan Author	TIMING Start date	Finish date	Numerical Results	Notes on Testing
1	Spec 1, Weight	Place robot on a scale. Read and record resulting measurement.	The weight of the fully assembled robot, without tennis balls, in lbf	Under 25 lbf	Tennis court, and a scale	VP	Frances	Michael	5/6/21	5/6/21	Average weight was: 24.25 pounds. This is right under the accepted criteria. In the future it would be good to try and lighten the weight of the robot through other means.	Weight matched the hand calculations done by Matthew. Very close to the max weight specification.
2	Spec 2, Time to clear court	Scatter 72 tennis balls randomly across half a tennis court. Run the robot and measure the time it takes to collect at least 95% of the tennis balls.	The run time in minutes and at least 95% (68/72) of the balls have been collected.	The run time is less than 15 minutes	Tennis court, 72 tennis balls, and a stopwatch	VP	Michael	Frances	X		Not able to complete this test due to a variety of reasons. Robert has not given us the necessary machine vision code to run the test. Our motors are also not working as expected and cannot provide enough power to move the robot around the court. This test relies heavily on both of these factors to get the necessary data, and thus we couldn't perform the test.	
3A	Spec 3, Collection cycles per charge	1: Fully charge the battery and run the robot with the Spec 2 test. Repeat this until battery is depleted.	The number of cycles completed.	The number of cycles completed is greater than or equal to 5	Tennis court, 72 tennis balls, and a stopwatch	VP	Alex	Alex	X		This test was not completed as the machine vision component was unable to be integrated with the rest of the robot. Without the machine vision component, the device is unable to seek and collect tennis balls.	
3B		2: Fully charge the battery then run robot continuously until the battery is depleted. Measure the time needed to deplete the battery	The measured time.	The time measured is longer than 75 minutes	Tennis court and a stopwatch	VP	Michael	Michael	X		This test was not completed as well due to the machine vision component not yet being integrated. Without the machine vision component, less current will be drawn and any data collected on the time required to deplete the battery will be inaccurate. As a team, we chose to avoid proceeding forwards without the machine vision component fully functional to unnecessarily damage the battery's health.	
4	Spec 4, Time to move collected balls to serving basket	Remove internal storage and empty the tennis balls into the serving basket.	The time it takes to remove the basket and pour it into the serving basket.	The time measured is less than 30 seconds	Stopwatch, 72 tennis balls, and two baskets	VP	Frances	Robert	5/23/21	5/24/21	Average time to complete this test was 36 seconds. Times ranges from 18 seconds to slightly over 1 minute.	The average time was higher than the acceptance criteria. To mitigate this issue, we plan to put together a more comprehensive guide for the user manual to ensure the time measured is shorter.
5	Spec 5, Time spent operating device	Hand the robot to someone unfamilar with the project, and measure the time it takes for them to figure out how to operate the device.	The measured time.	The time measured is less than 5 minutes	Stopwatch	VP	Alex	Robert	5/23/21	5/24/21	The average time to set up and operate the device was 2:44.54 seconds. This is well under the acceptable measured time of 5 minutes. Along with this everyone who participated in the user testing	Some interesting points were to see how users that had some previous knowledge of the robot performed better overall in the tests. We also could provide more detail in the user manual for initial set up and to guide new customers in the future.
6	Spec 6, Accuracy of ball detection	Run the detection algorithm on different types of tennis courts. These courst ideally will differ in size, color, and lighting.	Assess the conditions in which the algorithm cannot detect the tennis balls.	Detects 95% of tennis balls	Tennis court, camera, 72 tennis balls	SP	Robert	Robert	X		This test was not able to be completed as we were not able fully implement the machine vision to be integrated with the rest of the robot. We believe the machine vision to be working, but due to a lack of communication, we have not been able to combine it with the rest of the elecontric components. Thus, the test could not be completed.	
7	Spec 7, Robot range of communication with controller (phone)	Stand far away from the robot and attempt to command it with the phone device. Slowly approach the robot while continuously attempting until the robot responds.	Distance at which the robot beings to recieve commands.	The distance measured is greater than 25 feet.	Tennis court, phone device	VP	Alex	Robert	5/1/21	5/24/21	Maximum operating range of 80 feet.	Test was completed with a clear line of sight to the robot, with minimal nearby towers, poles or obstructions. Tennis courts are approximately 80 feet in length, so this operating range far exceeds the expected operating distance.

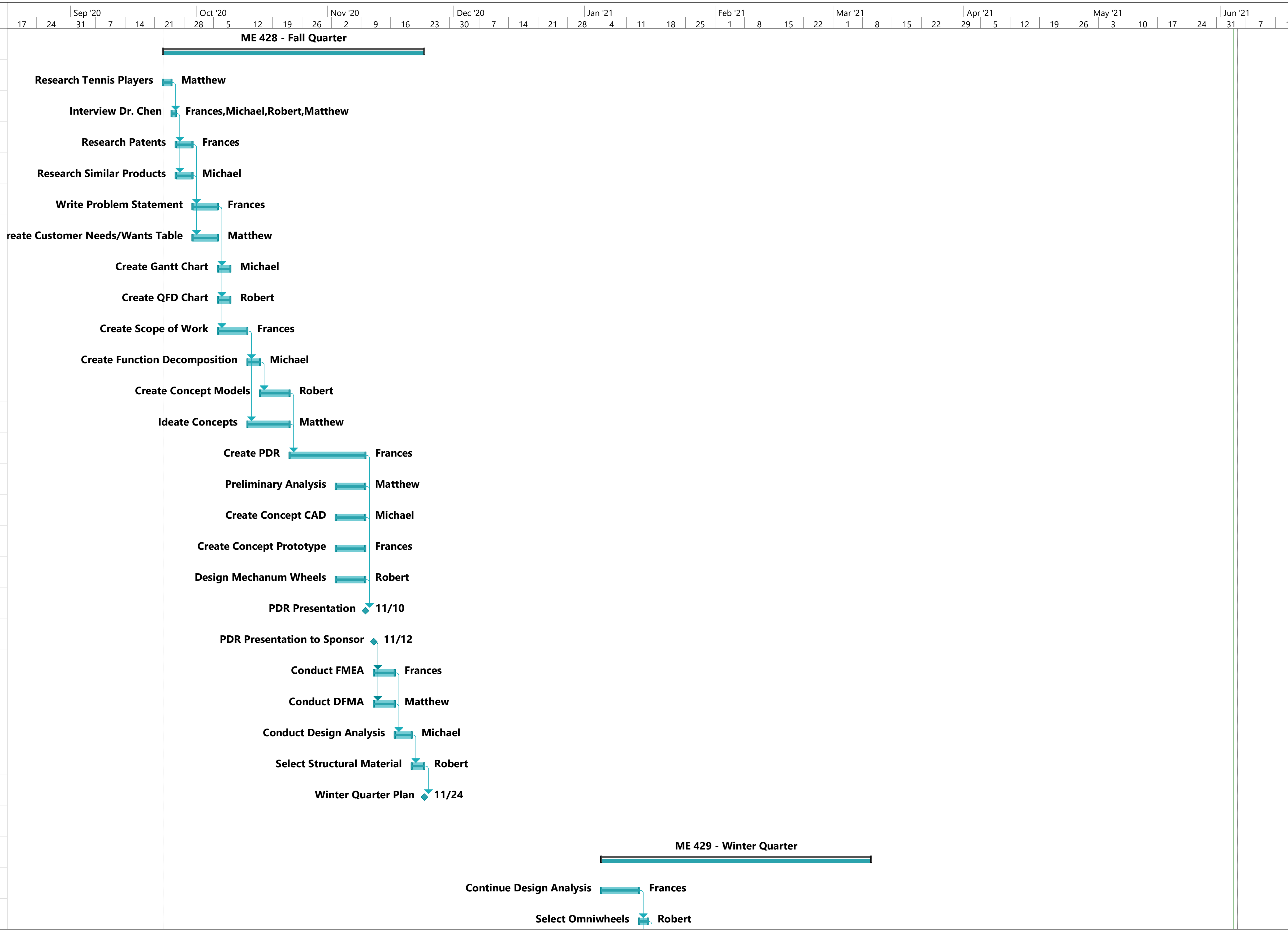
DVP&R - Design Verification Plan (& Report)

Project:	F71-10S Bot		Sponsor:	John Chen							Edit Date:	5/25/21
TEST PLAN											**TEST RESULTS**	
Test #	Specification	Test Description	Measurements	Acceptance Criteria	Required Facilities/Equipment	Parts Needed	Responsibility		TIMING		Numerical Results	Notes on Testing
								Test Plan Author	Start date	Finish date		
8	Spec 8, Ability to collect tennis balls	Set up structural prototype and push balls towards rotating cylinders to determine collection ability	Feasibility to collect tennis balls, how many are are able to move through cylinders	Moves tennis balls from ground past cylinders	Structural prototype, tennis balls	SP	Frances	Matthew	2/6/21	2/18/21	The structural prototype was not able to successfully collect any tennis balls, although it didn't shoot the tennis balls in the wrong direction either. Video footage of tests in OneDrive	Need to have stronger motors, need to secure the rod to the bottom of the frame so it is more secure. Used metal rod instead of wooden dowel and saw improvements.
9	Preliminary test for Spec 8 - Ability to collect tennis balls	Set up verification prototype and spread out 15 balls in a nearby vicinity to the robot, film the footage to analyze later and watch specifically for the ball collection functionality.	Feasibility to collect tennis balls, how many are are able to move through cylinders	Moves tennis balls from ground past cylinders	Verification prototype, tennis balls	VP	Frances	Frances	5/24/21	5/24/21	During the first test that we ran, the roller were able to pick up 100% of the balls from the ground and able to propel them into the basket with 66% efficiency. For the second test we made some modifications to the cylinders and the efficiency of propelling the balls into the basket increased to 90%	We got varying results based on the location of the tennis balls in relation to the rollers. When the tennis balls were directly in front of the rollers, it was much easier to pick them up. Overall, the robot did not have any issues moving the tennis balls off the ground, and with some slight modifications to the roller location, it should be able to get them all in the basket.

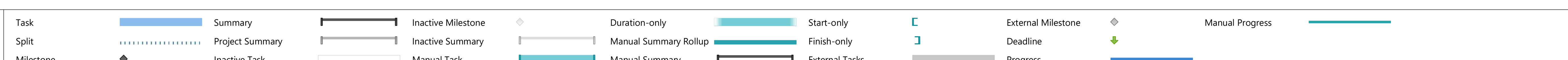
N - 2

Appendix [O] - Gantt Chart

ID	Task Name	Start	Finish	Duration
1	**ME 428 - Fall Quarter**	**Wed 9/23/20**	**Tue 11/24/20**	**62 days**
2	Research Tennis Players	Wed 9/23/20	Fri 9/25/20	2 days
3	Interview Dr. Chen	Fri 9/25/20	Sat 9/26/20	1 day
4	Research Patents	Sat 9/26/20	Wed 9/30/20	4 days
5	Research Similar Products	Sat 9/26/20	Wed 9/30/20	4 days
6	Write Problem Statement	Wed 9/30/20	Tue 10/6/20	6 days
7	Create Customer Needs/Wants	Wed 9/30/20	Tue 10/6/20	6 days
8	Create Gantt Chart	Tue 10/6/20	Fri 10/9/20	3 days
9	Create QFD Chart	Tue 10/6/20	Fri 10/9/20	3 days
10	Create Scope of Work	Tue 10/6/20	Tue 10/13/20	7 days
11	Create Function Decomposition	Tue 10/13/20	Fri 10/16/20	3 days
12	Create Concept Models	Fri 10/16/20	Fri 10/23/20	7 days
13	Ideate Concepts	Tue 10/13/20	Fri 10/23/20	10 days
14	Create PDR	Fri 10/23/20	Tue 11/10/20	18 days
15	Preliminary Analysis	Tue 11/3/20	Mon 11/9/20	7 days
16	Create Concept CAD	Tue 11/3/20	Mon 11/9/20	7 days
17	Create Concept Prototype	Tue 11/3/20	Mon 11/9/20	7 days
18	Design Mechanum Wheels	Tue 11/3/20	Mon 11/9/20	7 days
19	PDR Presentation	Tue 11/10/20	Tue 11/10/20	0 days
20	PDR Presentation to Sponsor	Thu 11/12/20	Thu 11/12/20	0 days
21	Conduct FMEA	Thu 11/12/20	Tue 11/17/20	5 days
22	Conduct DFMA	Thu 11/12/20	Tue 11/17/20	5 days
23	Conduct Design Analysis	Tue 11/17/20	Sat 11/21/20	4 days
24	Select Structural Material	Sat 11/21/20	Tue 11/24/20	3 days
25	Winter Quarter Plan	Tue 11/24/20	Tue 11/24/20	0 days
26				
27	**ME 429 - Winter Quarter**	**Tue 1/5/21**	**Wed 3/10/21**	**64 days**
28	Continue Design Analysis	Tue 1/5/21	Thu 1/14/21	9 days
29	Select Omniwheels	Thu 1/14/21	Sat 1/16/21	2 days

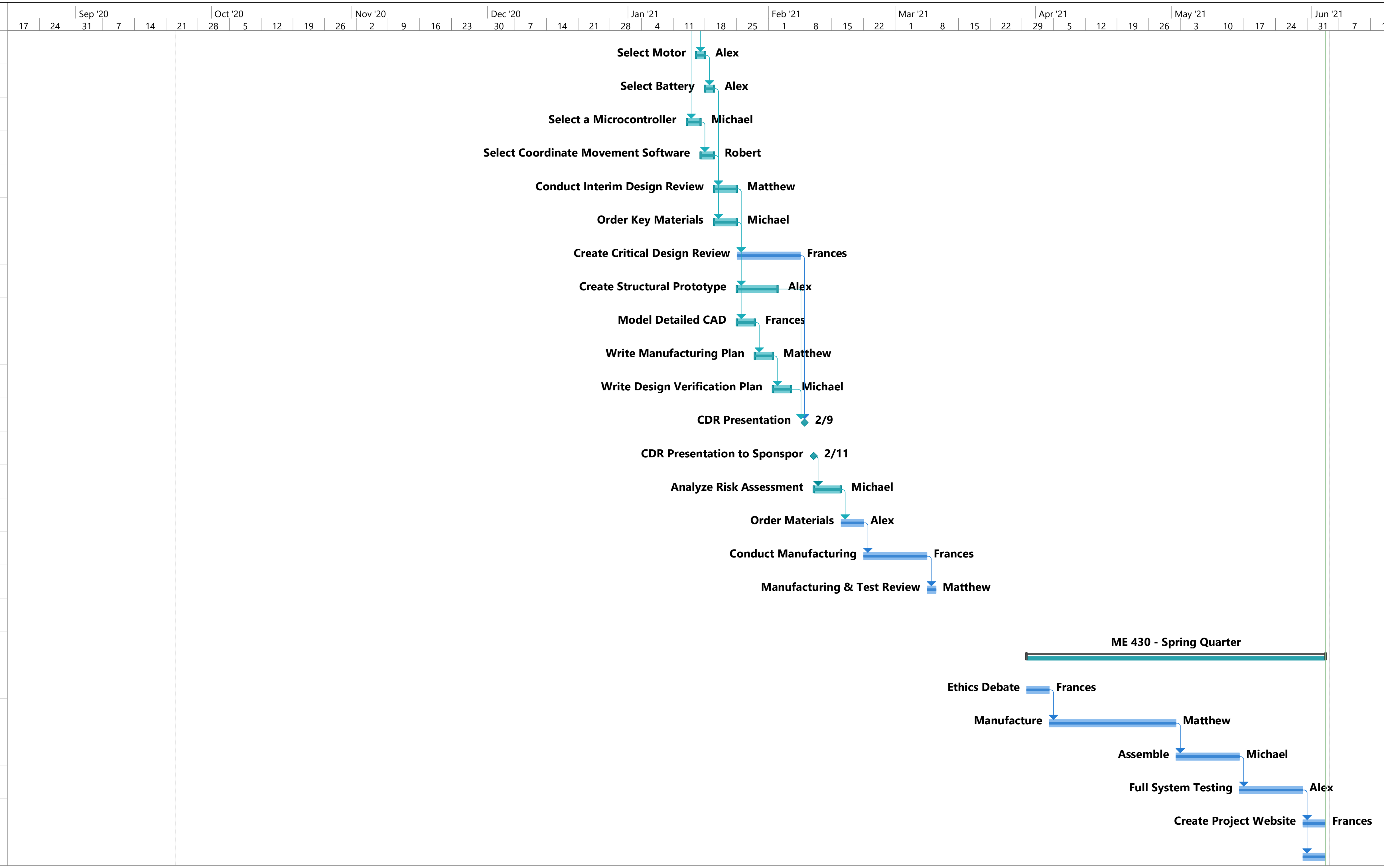


Project: F71 Gantt Chart
Date: Fri 6/4/21



Page 1

ID	Task Name	Start	Finish	Duration
30	Select Motor	Sat 1/16/21	Mon 1/18/21	2 days
31	Select Battery	Mon 1/18/21	Wed 1/20/21	2 days
32	Select a Microcontroller	Thu 1/14/21	Sun 1/17/21	3 days
33	Select Coordinate Movement	Sun 1/17/21	Wed 1/20/21	3 days
34	Conduct Interim Design Review	Wed 1/20/21	Mon 1/25/21	5 days
35	Order Key Materials	Wed 1/20/21	Mon 1/25/21	5 days
36	Create Critical Design Review	Mon 1/25/21	Mon 2/8/21	14 days
37	Create Structural Prototype	Mon 1/25/21	Wed 2/3/21	9 days
38	Model Detailed CAD	Mon 1/25/21	Fri 1/29/21	4 days
39	Write Manufacturing	Fri 1/29/21	Tue 2/2/21	4 days
40	Write Design Verification Plan	Tue 2/2/21	Sat 2/6/21	4 days
41	CDR Presentation	Tue 2/9/21	Tue 2/9/21	0 days
42	CDR Presentation to Sponspor	Thu 2/11/21	Thu 2/11/21	0 days
43	Analyze Risk Assessment	Thu 2/11/21	Wed 2/17/21	6 days
44	Order Materials	Wed 2/17/21	Mon 2/22/21	5 days
45	Conduct Manufacturing	Mon 2/22/21	Mon 3/8/21	14 days
46	Manufacturing & Test Review	Mon 3/8/21	Wed 3/10/21	2 days
47				
48	**ME 430 - Spring Quarter**	**Tue 3/30/21**	**Fri 6/4/21**	**66 days**
49	Ethics Debate	Tue 3/30/21	Sat 4/3/21	5 days
50	Manufacture	Sun 4/4/21	Sun 5/2/21	28 days
51	Assemble	Sun 5/2/21	Sun 5/16/21	14 days
52	Full System Testing	Sun 5/16/21	Sun 5/30/21	14 days
53	Create Project Website	Sun 5/30/21	Thu 6/3/21	5 days
54	Create Final Design Review	Sun 5/30/21	Thu 6/3/21	5 days



Project: F71 Gantt Chart
Date: Fri 6/4/21

Task	Summary	Inactive Milestone	Duration-only	Start-only	External Milestone	Manual Progress	
Split	Project Summary	Inactive Summary	Manual Summary Rollup	Finish-only	Deadline		
Milestone	Inactive Task	Manual Task	Manual Summary	External Tasks	Progress		

Appendix [P] –User Manual

Maintenance

Replacing the Basket Attachment

If a new basket attachment piece is required, refer to the "Basket Attachment.stl" file to 3-D print a new one. Regarding the 3-D printing setting, any pattern is acceptable, but it is recommended to use a zig-zag pattern with an infill of between 30-45%. The holes included in the 3-D printed part will need to be drilled to remove supporting material. Finally, using the screws previously provided with the project, secure the basket attachment in place to the 8020 chassis and the ramp. When attaching to the vertical 8020, ensure the washer is included and secure.

Replacing the PVC Pipe Attachment

If a new PVC pipe attachment is needed, refer the "PVC Pipe Attachment.stl" file to 3-D print a new one. Regarding the 3-D printing setting, any pattern is acceptable, but it is recommended to use a zig-zag pattern with an infill of between 30-45%.

Replacing the Angled Ramp Bracket

If a new angled ramp bracket is needed, first determine which side needs replaced, then refer to the corresponding Left or Right "Angled Ramp Bracket.stl" file to 3-D print a new one. Regarding the 3-D printing setting, any pattern is acceptable, but it is recommended to use a zig-zag pattern with an infill of between 30-45%.

Charging the Battery and Checking its Capacity

Proper Charger Figuration

Proper Terminal Connections

Using the charger, attach the connections red to red and black to black. Ensure that they “snap” as they are connected. On the blue charger, be sure to flip the switch to the 0.9 Amp setting to ensure that the battery is properly charged. Never leave the battery unattended while charging. Disconnect the battery immediately after the LED on the blue charger changes from red to green. To check the capacity of the battery, use a voltmeter set on VDC and connect red to red and black to black; a typical reading at full capacity is approximately 12 to 14 V. If the battery shows signs of puffing up or ballooning, immediately dispose of it according to the manufacturer’s suggestions.

Trouble Shooting

Electronics and Wiring

WARNING:	Do not attempt to troubleshoot the electronics while the battery is connected. Be sure to unplug the battery before troubleshooting.

Before powering the device, be sure to check that all wirings are secure to prevent entanglement and proper operation. If any wires are disconnected, consult the wiring diagram in Appendix [H] and attempt to reconnect the wires. If the connection was originally soldered together, seek a qualified electrical technician to repair the connection. If all wires are connected and the device is not operating as expected, check for signs of corrosion and visible shorts, verify using a multimeter if necessary. If no shorts are detected, verify that the LEDs on the electrical components, such as the Raspberry Pi, Arduino, and motor drivers, are brightly lit. If most are lit but one or some are not, this could indicate that the chips are damaged. In this case, further troubleshooting should be done according to the particular chip's manual. If the chip and shield are not recoverable, a new chipset can be purchased and installed according to the wiring diagram.

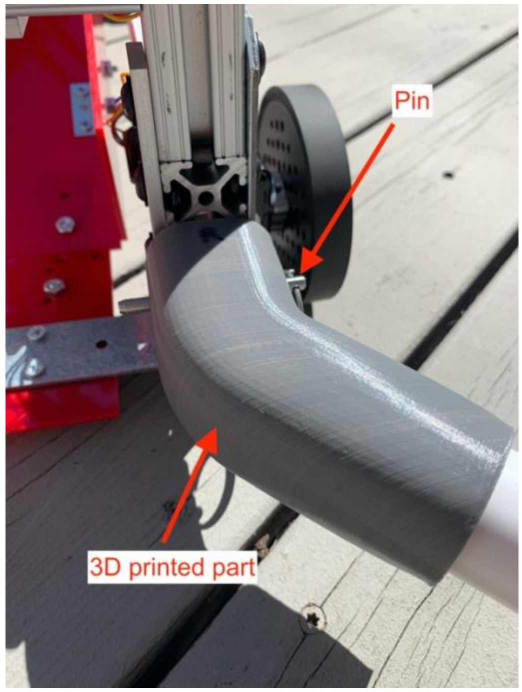
Rotating Cylinders not Working as Expected

If the rotating cylinders are not working as expected, first ensure that the flaps underneath the cylinders are not interfering with the cylinders as they attempt to spin. If the flap is interfering, this likely indicates that the flap has either started to warp or come loose from its fasteners and is now higher up than it should be. To troubleshoot, first try loosening and reattaching the screws to the ramp flap, focusing on keeping the flap low enough to not interfere with the cylinders. If this does not result in any change, the flap may need to be glued directly to the front edge of the ramp using a strong adhesive such as Gorilla Glue. Tape may also work as a temporary solution. If the flaps are not interfering with the cylinders, check that the battery has sufficient charge and is properly connected. If the battery is not sufficiently charged, reference the "**Charging the Battery and Checking its Capacity**" section of the User Manual for charging instructions. If the battery is sufficiently charged, there is likely a problem with the motor that is spinning the cylinder in question and needs to be replaced.

Assembly

Attaching PVC Arms

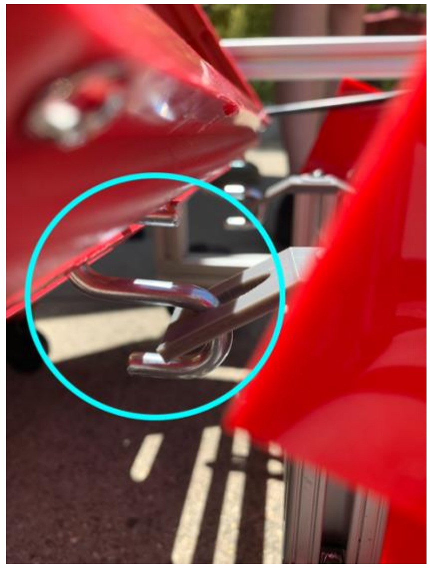
Take a PVC arm (the 3-D printed connector should already be attached) and orient it so that the hooked end is facing inwards when attached to the robot. Each connector piece should have either an "L" or an "R" marking on its upward face indicating which side of the robot it attaches to. Slide the square end of the connector onto the small protruding T-slot at the bottom left (or right) area of the front of the robot. Ensuring that the small hole on the sides of the connector lines up with the hole in the T-slot, secure the connector in place by inserting a quick release pin fully through the hole on the connector. Repeat for the second PVC arm.



Assembled PVC attachment with pin.

Attaching Basket

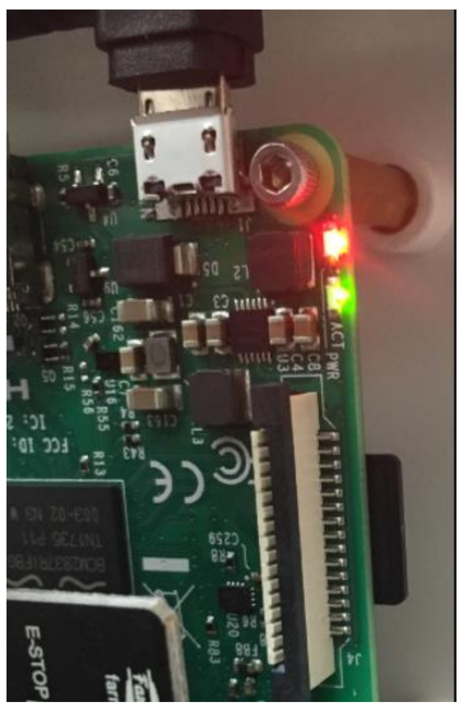
Check the 3-D printed basket attachment piece is connecting the ramp to the vertical 8020 supports. If this is not the case, reference the **"Replacing the Basket Attachment"** section of this user manual. Once the basket attachment piece is secured take one of the S-hooks and secure it to the free end of the basket attachment piece. Attach the other S-hook to the other basket attachment piece. Slide the basket underneath the basket attachment piece and secure the other end of both S-hooks to the basket. These should attach to the spaces between the slats in the basket and the S-hooks should be quite secure.



S-hook attached to basket and 3-D printed attachment piece

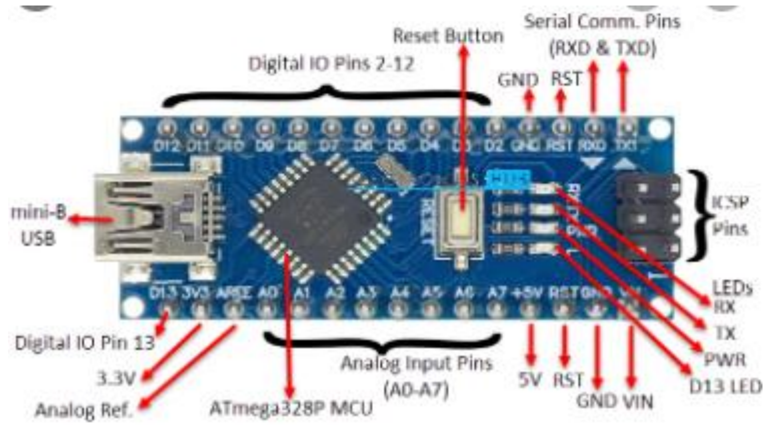
Bootup

Plug the battery connector into the battery. This will deliver power to all the systems on the robot. Wait 60 seconds for the Raspberry Pi to boot up.



LED Lights On Pi Indicating Power ON

The Arduino begins booting up as soon as the systems receive power and takes less than 10 seconds to boot up.



PWR LED Will Indicate Arduino is Powered ON

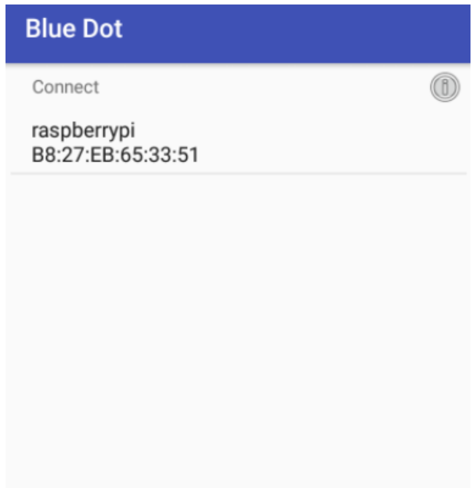
Once the Pi has booted up, the autonomous collection software will automatically run. To begin collection of the tennis balls, see the ”**App Instructions”** section.

General Setup

To begin setup of the robot ensure it is resting on a flat surface and there is about 1ft of space all around it to attach the PVC arms. To attach the PVC arms, reference the **"Attaching PVC arms"** section of this manual. To attach the basket, reference the **"Attaching Basket"** section of this manual. Finally, the last step in the general setup is to connect the robot to the app on your phone via Bluetooth. This can be achieved by following the **"App Instructions"** section. You may then begin collecting tennis balls with the 10S bot.

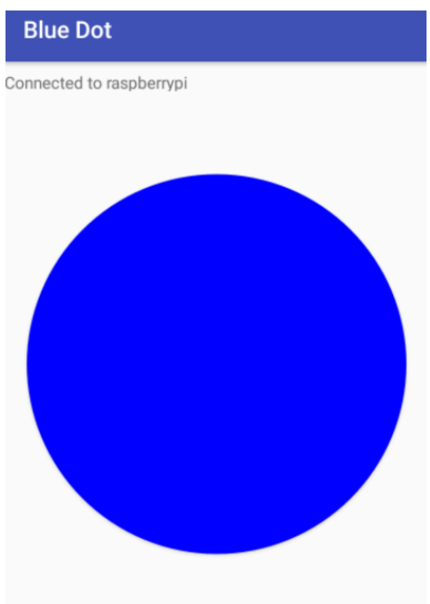
App Instructions

After turning on the robot and placing it in the desired starting position, open the Blue Button app on the Android powered phone. The app will prompt the user to select a Bluetooth enabled device to pair with.



Blue Dot App Will Ask User to Select Server-side Host

The robot will show up on the app as "raspberrypi". Select this option and wait for the devices to pair. Once they have paired, a blue button will show up on the screen of the phone. Simply press the button to begin autonomous ball collection.



The User-Controlled Blue Dot