24-671 Electromechanical Design
Final Report

Group TBD
Project PupBuddy

May 11, 2018

Peter Li, Lukas Cervenak, Jae-Eun Lim, Xiaorui Li, Lenny Zhang

Team member name	Signature
Xiaorui Li	Xiaorui Li
Jae-Eun Lim	
Lukas Cervenak	
Yunheng Zhang	
Jun Yang Li	

Table of Contents

1. Executive Summary

It is no doubt that dogs can be great friends and bring a lot of happiness to our life, yet spending time to take care of and interact with them, especially during workdays can be an issue for many dog parents. Currently, common solutions include finding a dog-sitter, leaving dogs alone at home, and/or taking dogs to the workplace, which can be either unnecessarily costly or distracting for the owners, and lonely or even unsafe for the dogs. As the number of dog parents increases annually, this problem becomes more urgent and no product on the market today addresses all three issues of monitoring, feeding, and interacting with dogs away from home. To thoroughly solve this problem, we came up with PupBuddy, a mobile remotely-controlled treat launching toy designed to help pet owners who work away from home to monitor, feed and interact with their pets via the Internet, anytime and anywhere.

Throughout the past four months, we have built three prototypes at the product level and conducted much more design iterations at the subsystem level for each prototype. In the end, our final working prototype consists of four major subsystems -- driving, treat launching, treat loading, power and webapp, and each of them more or less needs to be designed with considerations like aesthetics, mechanism, operating principles, control, user interface, manufacturability, communication, environmental impact, etc.

This final report records the entire design process of PupBuddy. Firstly, we defined the problem of attending pets while owners who are at work, and conducted online research and internal discussion to identify stakeholders & customer needs, which were then converted into target specifications with quantitative descriptions. Secondly, we brainstormed as a team to generate ideas for each subsystem given the targeted features and used a series of engineering matrices and methods to assess and select the best concept. Thirdly, detailed design and engineering analysis throughout all prototyping stages were illustrated and explained. The section primarily constitutes CAD models of all components and the final assembly to illustrate design modifications, FEA simulations on the cylindrical shell, the hemispherical wheels and the drive motor shaft, jamming tests on different loader iterations, the MATLAB dynamic simulation and field tests of the largest tilting angle of the body, speed tests of PWM control signal and the product performance, frictions for different rubber coatings, battery charging and recharging, webapp latency tests with and without the camera on board. As a result, the product design has been iteratively improved given the test results and finally led to our final prototype. Besides, we also discussed FMEA, design for environment, manufacturing, and assembly considerations and techniques. Fourthly, a thorough description of our final prototype and subsystems was presented with demonstration, tests and results. Lastly, we concluded the report with lessons learned from this project, reflection on decision making, assessment of unsolved issues and suggestions for future work.

Generally speaking, the final prototype fulfilled most of the design features and worked well as expected. To continue the project, manufacturability, robustness and user testing will be the main foci to start with.

2. Problem Definition

2.1 Problem Statement

PupBuddy is a mobile treat dispensing dog toy, created to solve the problem that pet owners cannot interact, feed, and exercise their dog while away from home.

This issue is especially prevalent with dog owners who work during the day. Currently there are a few products that are trying to address this problem using treat dispensing or remote control movement but no product has addressed all issues. We plan to design a device that can launch treats, be controlled through Wi-Fi, as well as exercise the dog. Our product allows the busy owner to interact with the dog while at work, sufficiently exercise the dog, and finally incentive the dog to move through the reward of treat launching.

2.2 Mission Statement

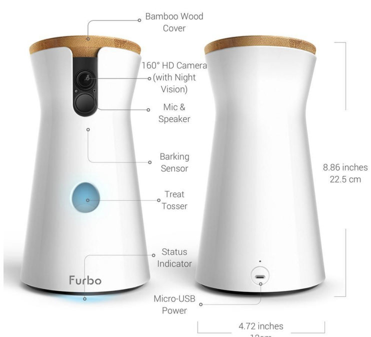
Product Description	A web app controlled dog toy with video streaming that can launch treats
Benefit Preposition	PupBuddy allows dog owners to exercise and interact with their dog remotely.
Primary Market	Dog owners that work
Secondary Market	Other dog owners Other pet owners
Assumptions and Constraints	Constraints: ● $750 budget ● 1 semester ● Limited manufacturing access ● System size is limited due to off-the-shelf electronic components Assumptions: ● Final prototype is not a final product ● Survey is accurate sample
Stakeholders	Customers, manufacturer, competitors, suppliers, pet safety organizations, pet food manufacturers

2.3 Competitors

GoBone($169)

Pebby($189)

Furbo($199)



Petnet ($119)

1. GoBone
    a. Operating Principle: All-day exercise and play
    b. Key Element:
        i. Two independently rotating wheels
        ii. Microcontroller
        iii. Haptic sensor (or similar, to sense the padding)
        iv. Mechanical parts: elastic wheel cases, shaft shell, case for electronics
        v. Rechargeable battery (8 hours in auto mode, ~30 min in manual mode)
        vi. Telecommunication module (Bluetooth smart wireless radio)
        vii. LED
        viii. Software app, machine learning algorithm (adjust to the pace of dogs)

GoBone can interact with pets in both auto and manual modes. It is also pretty easy to assemble, charge and replace any mechanical components. However, the bluetooth connection only allow short-distance interaction. The lack of camera or audio limit the scope of interaction and monitoring if the pet owners are away from home. Few snacks can be put into two sides of the bone-shaped toy.

1. Pebby
    a. Operating Principle: Playtime anytime anywhere
    b. Key Elements:
        i. Mechanical:
            1. Polycarbonate plastic spherical casing
            2. **SHOCK™** suspension system
            3. **EASYfix™** magnetic caps (allow customized the outlook)
            4. **PebbyKennel™** Charging dock
        ii. Electronical:
            1. In-build camera (wide-angle, fish-eye lens)
            2. Laser (e.g. to play with cats)
            3. Two-way audio
            4. Wifi & bluetooth (including wireless charging)
            5. Activity tracker
            6. LED
            7. Rechargeable battery (1.5 hours playtime, 12 hours idle time)
        iii. Software app, bark alert

Pebby focuses more on interaction and activity of pets, so it does not have any treat/food dispensing mechanism. The camera, wifi, and two-way audio greatly augment the interactive experience for both pet owners and pets. Its auto-charging, the user driving it into the dock, could ensure the continuity of playing and monitoring functionalities.

1. Furbo Dog Camera
    a. Operating Principle: Remote Training. Know of emergencies. Never miss a moment. No more lonely dogs at home.
    b. Key Elements:
        i. Mechanical
            1. Food/treat container (body)
            2. Adhesive feet
            3. Bamboo wood lid
            4. Tossing mechanism
        ii. Electronic
            1. 160 degree HD camera & night vision
            2. Mic & Speaker
            3. Bark sensor
            4. Micro-USB power & adaptor (no battery)
            5. Lights (Probably LED)

6. Microcontroller
7. Wifi module

iii. Software app, activity monitoring, alert, AI-powered dog recognition

Furbo is an stationary dog camera capable of tossing treats. It touches all interactive, monitoring and feeding sides a little bit, yet it is not as effective as a mobile robotic product in terms of helping pets exercise. The night view camera and activity alert system do boost the monitoring feature.

1. Petnet SmartFeeder
    a. Operating Principle: Bring you more joy while keeping your pets healthy and happy
    b. Key Elements
        i. Mechanical:
            1. Food container (bucket, up to 7 lb)
            2. (smart) Bowl
            3. Food dispensing mechanism
            4. Exterior casing
        ii. Electronic:
            1. Food dispense control
            2. Nest Camera
            3. Wifi module
            4. Light indicator
            5. Weight sensor (to measure the amount of food)
            6. Build-in rechargeable battery, USB cable, adapter
            7. Compatible with Amazon Alexa
        iii. Software app

Petnet Smart Feeder aims to build a healthy eating habits of pets, so it lacks the interactive aspect. Instead, it is able to hold a large amount of food at once, and to allow users to control the portion delivered each time.

3. Stakeholders and Customer Needs

3.1 Stakeholder Identification

- Customers: Dog owners who go to work during the day
- Competitors:
    - Automated pet toys and Treat dispensers/tossers like GoBone, Pebby, Furbo, Petnet, etc.
- Pet product designers, engineers
- Manufacturers
- Suppliers
- Government regulators:
    - the US Food and Drug Administration (FDA)

        - the Department of Agriculture (or equivalent department) in each State
- Pet safety organizations
    - the American Association of Feed Control Officials, or AAFCO.
- Pet food manufacturers

3.2 Customer Needs

Internal Generation

We brainstormed as a team during regular weekly meetings to generate customer needs for PupBuddy. Because three out of five team members are raising at least one dog at home, we were able to think of requirements from both engineering and customer perspectives. Thus, the internal generation provided many a inspiration for further external outreaches.

External Outreach

In addition to internal idea generation, we also conducted a customer survey, a few customer interview, competitor analysis and literature review.

In literature review, we searched for several technical patents including animal interactive devices, a pet food dispenser, a ball launcher, etc. to sharpen our brains and open our minds for more inspirations, links contained in the Bibliography section. The results from analyzing four major competitors have been shown in sections above. In-person interviews have been conducted with customers with and without technical background to provide a wider range and more variety for customer requirements.

We conducted a survey with 8 questions tailored to help us gather information regarding typical types of treats, preferred system functions, time owners have at work, etc. To date, we received 18 responses from dog owners of various ages and jobs. Based on these responses, we came up with the customer requirements. The results are shown in Appendix A.

Based on our research, we learned that the typical size of dog treats ranged from 0.5-1 inch. The online survey result showed that 77.8% of the dog owners surveyed would prefer medium speed for a mobile dog toy and very negligible percentage preferred fast. About 55.6% of the dog owners preferred shooting treat mechanism to dropping mechanism. 61.1% of the dog owners indicated that their dog toys are typically tennis ball size and 38.9% of the dog owners indicated football size. The top four most important attributes of a mobile dog toy that dog owners indicated were bite proof, easy to use, long battery life and aesthetics.

As for the user interface, 55.6% of the dog owners preferred touchscreen control mechanism and next was keyboard control with 22.2% preference. Although more people desire touchscreen control, we took into consideration that our team is more experienced with keyboard control. Most of the dog owners spent about 1 hour every day to play with their dogs. Most also indicated that they have about 30 minutes of break during their work time. From this information, we conclude that 30 minute to 1 hour of battery life is sufficient.

Summarizing all the research and survey results, we come up with the objective tree for customer requirements listed below.

Objective Tree (Importance is marked by *, ** and *** from the least to the most)

R1 Feeding

R1.a Treat dispensing is tele-operated. ***

R1.b Treat is dispensed at a minimum distance of 0.5 m in order to keep the dog active. **

R1.c The system is able to contain a handful of treats. **

R1.d The system is compatible with one type of treat. **

R1.e Launch the treat***

R2 Entertainment

R2.a Owner is able to monitor the dog remotely via video. ***

R2.b Owner is able to speak to the dog remotely. **

R2.c The system is controlled remotely. ***

R3 Mobility

R3.a The speed of the system is less than TBD in/s so as not to scare the dog or arouse predatory instinct. ***

R3.b The system is able to move in all directions. ***

R4 Safety

R4.a The system is not swallowable. ***

R4.b The system does not contain hazardous materials (chemicals ***,Non-poisonous ***,Not much color bleeding/paint peeling ***)

R4.c The system is not easily breakable. (Difficult to break/shatter/chip, Don't want heavy metals/hard outside)***

R5 Durability

R5.a The system is bite proof. ***

R5.b The system lasts at least TBD days.**

R5.c The system is saliva proof.**

R6 Power (#1 constraint)

R6.a The system lasts at least 8 hours in auto mode **

R6.b The system lasts at leastw1 hour in manual mode ***

R7 Cheap

R7.a The system is no more than $500 per unit. **

R8 Ease of use

R8.a The system is easily refillable. ***

R8.b The web app is intuitive.***
R8.c The system is easy to assemble/disassemble. ***
R8.d The system is easily rechargeable. *
R8.e The system is easy to clean. **

R9 Aesthetics
R9.a The system is aesthetically pleasing. *

R10 Manufacturability
R10.a The system can be built in a semester. ***

Conclusion:
From internal discussion, we collected ideas primarily from an engineering perspective, while from external research we were able to expand the requirements from internal generation with more diversity, and importance levels. For example, one suggestion mentioned that the functionality of tossing a treat instead of releasing meals would be more interesting and interactive for user experience. Movever, dividing customer needs into categories made sub-system identification and concept generation much easier and clearer later on.

4. Target Specifications

By mapping the customer needs from the objective tree in section 3 to a quantitative table for target specifications, including specifications from our major competitor, GoBone. In each matrix, we picked ideal and marginal values according to customer requirements, time limit and our capabilities.

Metric #	Needs #	Metric	Importance	Measurement	Ideal Values	Marginal Values	GoBone
1	R1.a, R2.a, R2.b, R2.c	Wireless connection	5	Boolean	hotspot/ Wifi	hotspot/ Wifi	Bluetooth
2	R1.b, R3.a	Min Interaction Distance	3	m	0.8- 1m	.5 m	0 m
3	R1.d	Treat Diameter	5	cm	1-2 cm	1-1.5 cm	1-2 cm

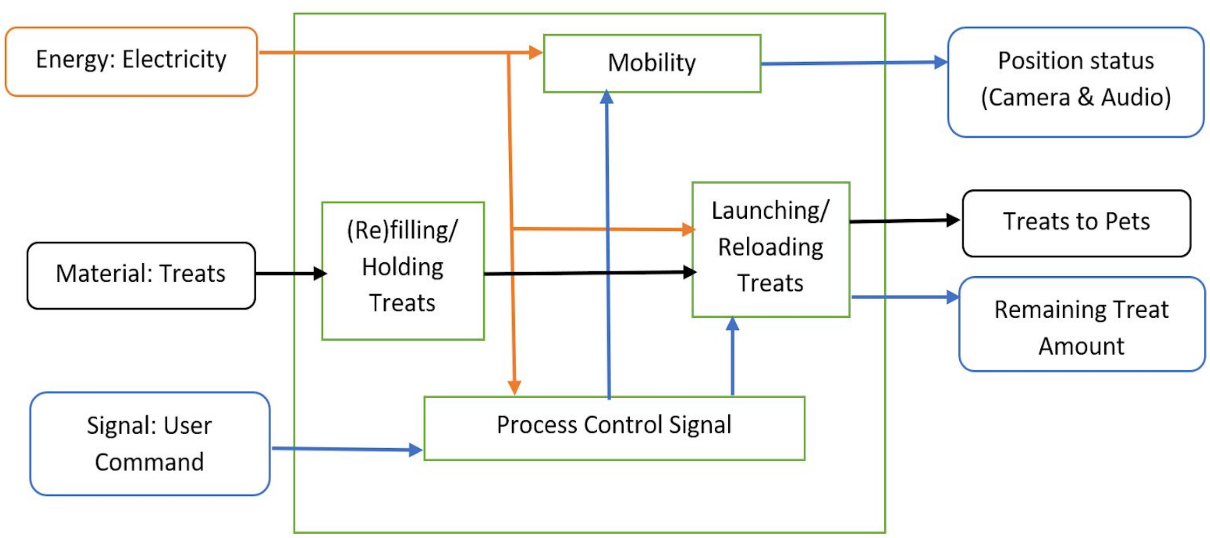
	R4.a	System Diameter	5	cm	15-30 cm	15-30 cm	25.7 x 9.2 x 14.7 cm^3
4	R1.c	Container Volume	3	cm^3	150cm^3	75 cm^3	~25 cm^3
5	R4.c, R5.a	Material yield strength	5	Pa	100 MPa	60 MPa	Wheel - Nylon and TPE;synthetic rubber
6	R4.c	Material elasticity	3	Pa	--	--	--
7	R3.b	Movability	5	dof	2	2	2
8	R3.a	Velocity of robot	3	m/s	1m/s	.5 m/s	~.75m/s
9	R6.a, R6.b , R8.d	Charge and Discharge times	5	Minutes	1 hour discharge	.5 hour discharge	8 hour in Auto-mode; 0.5 hr in idle-mode
10	R8.a	Refill time	1	Seconds	30s	1 min	15s
11	R10	Manufacture time	5	Weeks	1.5 months	2 months	Unknown
12	R8.c, R8.e	Cleaning time	1	Seconds	2 min	4 min	2 min
13	R7.a	Cost	1	Dollars	< $300	<$500	$169
14	R8.b	Intuitiveness	3	subjective	Easy to use for non-technical person	Easy to use for technical person	Easy to use
15	R9.a	Aesthetic	3	subjective	Beautiful	Not Ugly	Nice
16	R5.c	Splash resistance	5	binary	yes	yes	yes
17	R2.c	Wifi Bandwidth	3	Mb/s	15 Mb/s	10Mb/s	N/A

18	R2.a	Video resolution	3	Pixels x pixels	1280x720	800x600	None
19	R2.b	Video bitrate	1	Mb/s	6 Mb/s	3 Mb/s	N/A
20	R1.e	Launch the treat at certain speed/distance/height	5	m/s m cm	1 m/s 0.5 m 20 cm	0.5m/s 0.2m 10cm	N/A

<u>Conclusion:</u>
By using the customer surveys in addition to internal discussion, we were able to come up with a list of metrics that match the customer needs. In particular, our ideal and marginal specification values show an improvement in features over our main competitor, GoBone, since it can't perform tasks such as treat launching and video streaming. These specifications are still somewhat rough and may need to be updated in the future, but they will provide constraints in helping us generate useful concepts, formulate designs, and eventually choose the correct parts to purchase.

5. Concept Generation

5.1 Function Decomposition



Our system's main function is to allow users to remotely interact with their pet. The interactions include video viewing and treat dispensing. Divided further, there are 3 secondary functions: filling and holding treats, reloading/launching treats, and moving the robot, plus the overall control function. The function decomposition above provides a good abstraction of the system we want to build by being at a level to succinctly describe all the important functions, but at the same time not limit ourselves with any particular implementation.

5.2 User Control

Platform

1. Web application. We have experience building a web application, and almost any device can access the web-based interface.
2. Mobile (Android/iOS). There could be more options for control (gyroscope). However, we don't have experience building a mobile app.

Control method

1. Software buttons. This is the most accessible method, and allows all users to interact with the system. However, it is also the least intuitive.
2. Keyboard input. This is easy for people who would prefer to control from a laptop. It is not compatible with mobile app.
3. Joystick. For people with access to joysticks, this could be the control method that is most fun. This could work well as an optional secondary input.

5.3 Mobility

There are a variety of methods with which PupBuddy can move.

Concept 1 is a spherical device found by external search. The entire body of the robot rolls, and is balanced by a inner weight. This concept would always be balanced, and is sealed well, but it is slow and does not leave a good opening for treat launching.

Concept 2 is a four-wheeled vehicle similar to a toy car. This is very conventional looking, and is relatively easy to build, but multiple wheels increase manufacturing complexity, and it could be flipped over unless the wheel diameter exceeds the height of the vehicle.

Concept 3 is a two-wheeled vehicle inspired by one of our competitors: GoBone. This design looks good, cannot be immobilized, but could require extra coding to maintain good movement and controls.
More concepts can be found in the appendix.

5.4 Launch Treats

We thought of several different mechanisms to launch treats.

Concept 1. Use a rack and pinion system to pull the launching platform and store the energy using a spring (similar to a crossbow). This design is powerful, but is not very robust.

Concept 2. Use a rotating block that interfaces with a spring. The block's irregular shape pulls the spring, storing energy and releasing it at the right time to launch treats. This pulling system is more reliable than rack and pinion.

Concept 3. Use spinning wheels to accelerate and launch the treat. This design is simple to implement, but it takes up a large space near the launching entrance.

5.5 Loading

Concept 1. Motor powered spiral: a spinning, enclosed spiral pushes treats from one side of the container to the other. This does not rely on gravity, but it is difficult to reload the spiral.

Concept 2. Motor powered segmented wheel: a wheel with distinct segments can dispense treats. This is easy to make, but does not allow for granular treat dispensing that we want.

Concept 3. Spinning disk with a hole: a disk spins under the container. When two holes of the disk and the container line up, treats can fall out. This is very easy to make, but relies on gravity to function.

5.6 Appearance/Casing Material

Concept 1. Rubber coating with spikes. This allows the system to absorb shock, maintain good friction and contact with the ground, and is safe for the dog.

Concept 2. Smooth plastic shell (eg. polycarbonate). This makes implementing a camera easier, since it is clear. However, this is more prone to scratches and breaking.

Concept 3. Tennis ball fuzz. This material is easy to find, good for impact absorption and friction, but it is difficult to clean.

5.7 Concepts Combination

The Concepts Combination Table above lists all the concepts we derived from Concept Generation process. However, we narrowed down the options to ones that are feasible and in terms of manufacturability, capability, and size. We dropped all the concepts that were either impossible or not suitable. For example, legged locomotion and roly poly shaped system are extremely difficult to manufacture and control. Moreover, pneumatic treat launcher would be too powerful and may hurt the dog. Using fur as the outer material may have shedding issues. The concepts that are not colored are ones that we decided to drop.

From the narrowed down options, we developed four full system concepts. The above table shows the four concepts which are labeled from A to D and color-coded. These system concepts are explained below.

Conclusion

There is a wide design space for creating a physical design for our dog toy. In the Concept Generation phase, we explored some distinct points in that space and fleshed them out into full concepts. We learned about the viability (or lack thereof) of certain designs and mechanisms, so we can use these concepts to further our product development process, and attempt to find an optimal design among these concepts.

Concepts Combination Table

Mobility	Launching	Reloading	User Interface	User Control	Appearance/Materials
Two hemispherical wheels	Bean-spring actuator	Spinning wheel with a hole	Web app	Keyboard	Natural rubber coating
Spherical	Crossbow	Motor-powered segmented wheel	Android app	Joystick	Plastic (polycarbonate)
Four wheels	Rack and pinion	Gravity	iOS app	Button on screen	Tennis ball fiber
Treads	Spinning wheels	Two-chamber pushed by linear actuator	Web app	Keyboard	Cotton
	Catapult	Inclined turning plate with holes			Fur
	Trebuchet	Motor-powered spiral			Polyester
	Pneumatic				

Full-System Concepts	A	B	C	D

Concept A

Concept A is a system with two hemispherical wheels that has a treat launcher with bean-spring actuator and uses uses a spinning disk with a hole to reload the treat before shooting. The outer material is natural rubber coating. It uses a web app with keyboard control.

Concept B

Concept B is a spherical system that has a crossbow treat launcher and uses motor-powered segmented wheels to reload the treat before shooting. The outer material is plastic. It uses an Android app with joystick control.

Concept C

Concept C is a system with four wheels that has a treat launcher made of rack and pinion and uses gravity to reload the treat before shooting. The outer material is tennis ball fiber. It uses an iOS app with screen button control.

Concept D

Concept D is a system with treads that has a treat launcher made of spinning wheels and uses two chambers pushed by linear actuator to reload the treat before shooting. The outer material is cotton. It uses a web app with keyboard control.

6. Concept Selection

6.1 Pugh Matrix

Pugh Matrix						
		Concept Variants (1,3,5)				
Selection Criteria	**Reference Concept (GoBone)**	**Concept A**	**Concept B**	**Concept C**	**Concept D**
Tele-operation	0	0	0	0	0
Treat holding/releasing	0	1	1	1	1
Treat launching	0	1	-1	-1	1
Remote connectivity	0	0	0	0	0
Mobility	0	1	1	-1	-1
Safety	0	1	1	0	-1
Durability	0	1	1	-1	-1
Battery life	0	0	0	0	0
Cost	0	-1	-1	-1	-1
Ease of use	0	0	0	-1	1
Maintainability	0	0	1	1	-1
Aesthetics	0	1	1	-1	-1
Manufacturability	0	0	0	1	0
Rank	0	1	2	3	3
Score	ref	**5**	**4**	**-3**	**-3**
Verdict	ref	Keep	Keep	Drop	Drop

Using the four system concepts, we used a Pugh matrix to narrow down the options. We used GoBone as our industrial reference. We decided to drop the lowest two concepts and kept Concepts A and B. These two concepts scored particularly higher on safety and durability since Concept A and B are both spherical in shape which are safer to dogs than rectangular shapes with sharp edges. Both concepts also scored high on aesthetics. They had relatively low scores on manufacturability, however, the scores for other criteria made it insignificant.

Although Concept C with four wheels may have been easier to manufacture, we decided to drop it because it scored 7 points lower than Concept B with the second highest score. This was a sufficient indication for us that in multiple aspects, Concept C is inferior to Concepts A and B and we would have hard time getting Concept C to meet our customer requirements. Concept D received the same score as Concept C so we decided to drop it. Not only that Concept D was already our least favorite since treads are very difficult to manufacture and work with. We concluded that keeping Concepts A and B was a good decision, especially since they scored significantly higher than the lowest two.

6.2 Selection Matrix

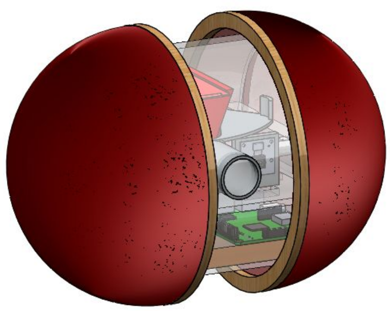
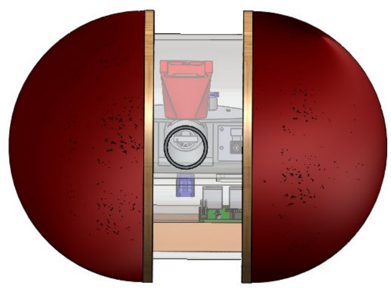
Selection Matrix			
		Concept Variants (1,3,5)	
Selection Criteria	**Weight (%)**	**Concept A**	**Concept B**
Tele-operation	10	3	3
Treat holding/releasing	8	5	3
Treat launching	7	3	1
Remote connectivity	8	3	3
Mobility	9	3	3
Safety	10	5	5
Durability	10	3	3
Battery life	5	3	3
Cost	4	3	5
Ease of use	8	5	3
Maintainability	6	3	5
Aesthetics	5	5	5
Manufacturability	10	3	1
Rank	--	1	2
Score	100	**3.62**	3.16
Continue?	--	Develop	No

<u>Conclusion</u>

To finalize our concept selection, we used the selection matrix as shown above to score Concepts A and B. We selected Concept A, which scored higher. Concept A scored higher on treat releasing because the concept of spinning wheel with a hole was a more feasible concept than a motor-powered segmented wheel which is more difficult to manufacture and more inaccurate in terms of releasing the right amount of treats. Concept A also scored higher on treat launching because Concept B's crossbow launcher is very difficult to implement, whereas Concept A's bean-spring actuator is much simpler to implement. Furthermore, Concept A scored higher on ease of use because more users are familiar with controlling from a computer than smartphones. Most importantly, Concept A scored higher on manufacturability which was one of our high weight criteria. Since, we are not dealing with a full sphere like Concept B which is more difficult to manufacture, Concept A scored higher. We were very satisfied with the result because the selection matrix helped us be more confident that our chosen concept was feasible and can meet our customer requirements.

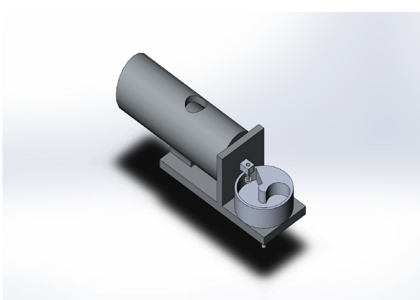
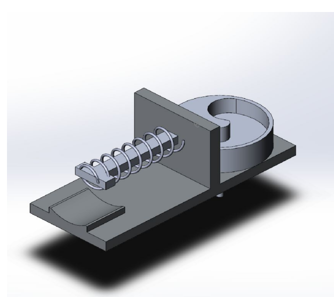
6.3 CAD Models & Prototypes

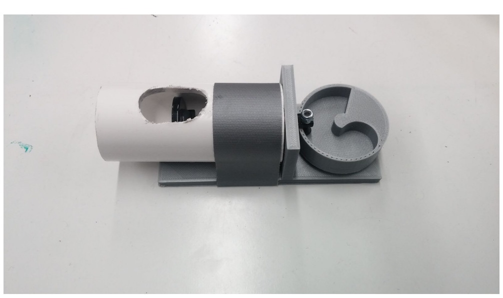
6.3.1 Full System


The drive system has two hemispherical wheels covered with rubber and actuated by two motors. The center cylindrical shell encases all electronics and subsystems and is bottom heavy to self-right.

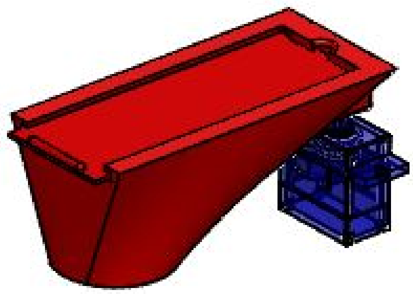
6.3.2 Launcher

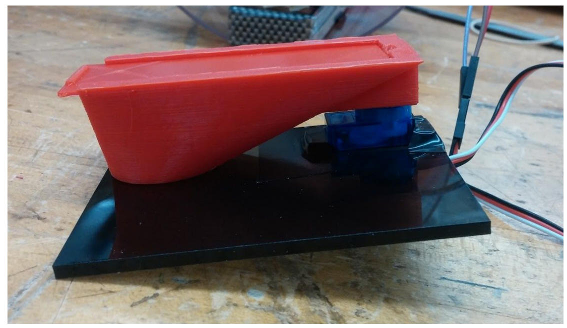





The launcher utilizes the bean-spring mechanism to compress and release the spring to shoot treats.

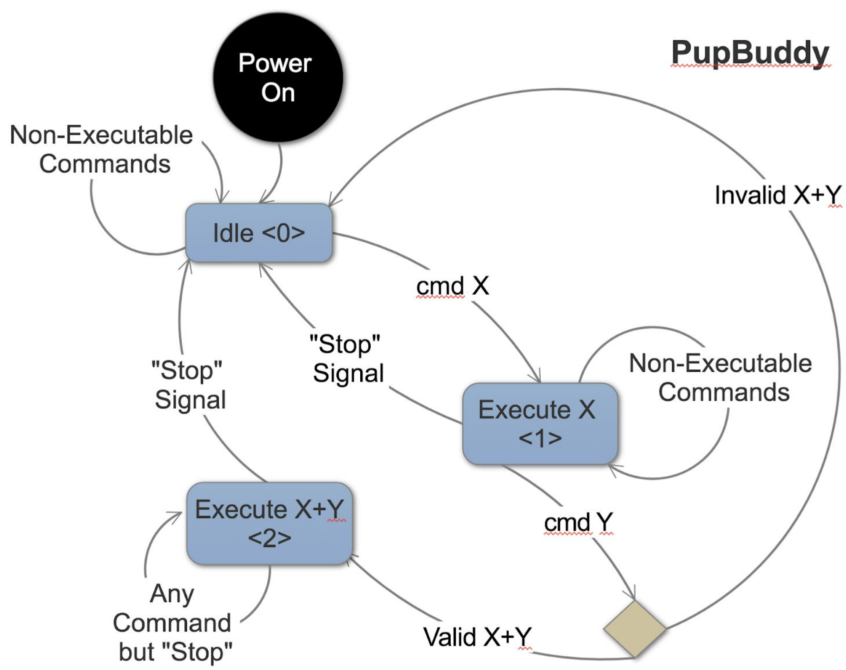
6.3.3 Treat Holder/Loader





The treat loader is a narrow container with a slidable lid. The servo is attached to one end which swings the loader over a stationary plate with a hole. When the exit of the loader and the hole on the plate aligns, the treats drop.

6.3.4 Web App State Machine



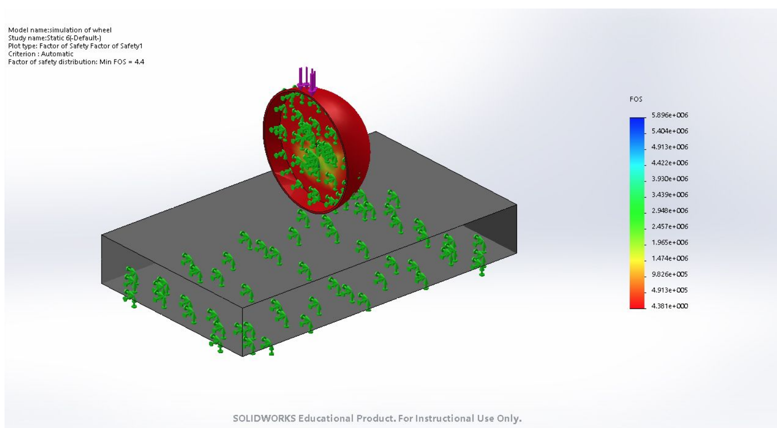
7. Detailed Design and Engineering Analysis

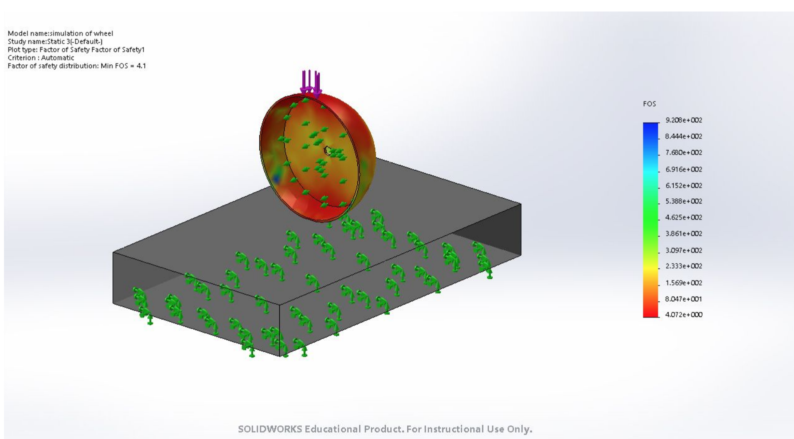
7.1 Drive System

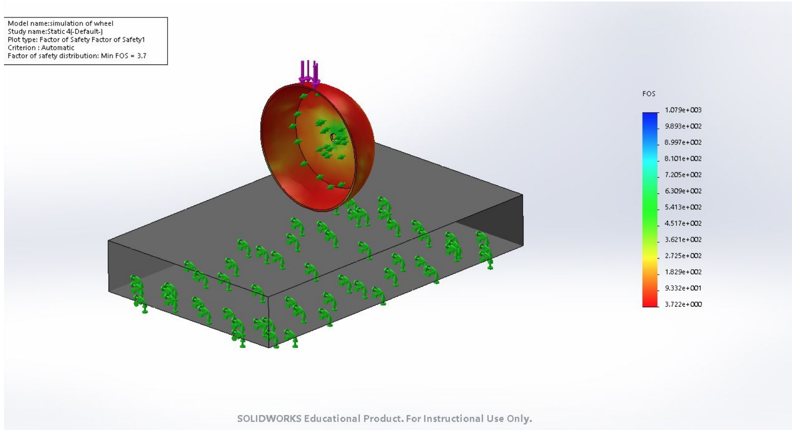
7.1.1 Hemispherical Wheels Static Stress Analysis

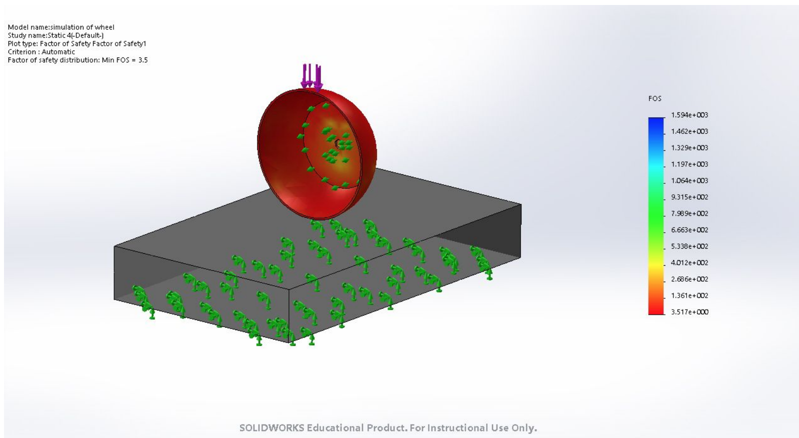
We planned to purchase an off-the-shelf 7” diameter acrylic hemispherical shell in order to simplify manufacturing process. In order to increase resistance to compression, we added a circular acrylic strut to the wheels. Since we have more room for the internal components when we place the strut further in from the edge of the hemisphere, we did FEA with varying distances from the edge in order to determine the optimal distance with sufficient resistance to compression due to a dog pressing down on the wheel. Below are five factor of safety plots from the FEA for strut being placed 1”, 1.5”, 2”, 2.5” and 3” from the edge.

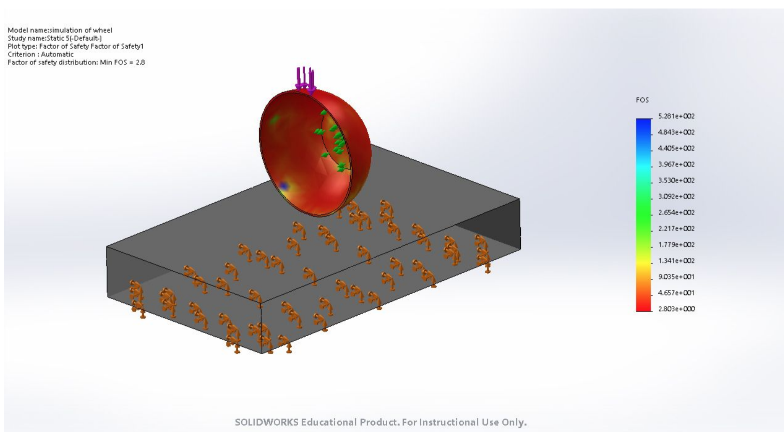
For the simulation setup, we placed a plain carbon steel plate underneath the wheel to simulate the floor. We imposed roller fixture on the surface of the strut and applied 40 lb downward force from the top of the wheel, which we considered the maximum limit of dog weight.



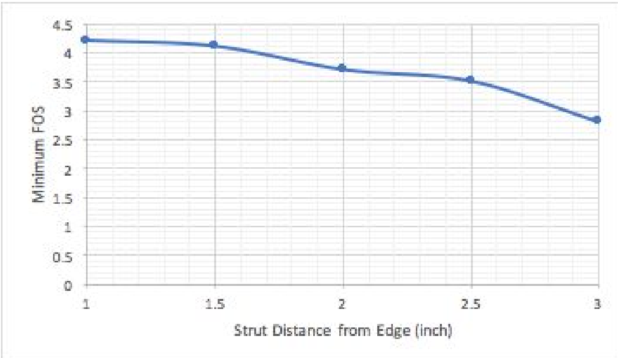


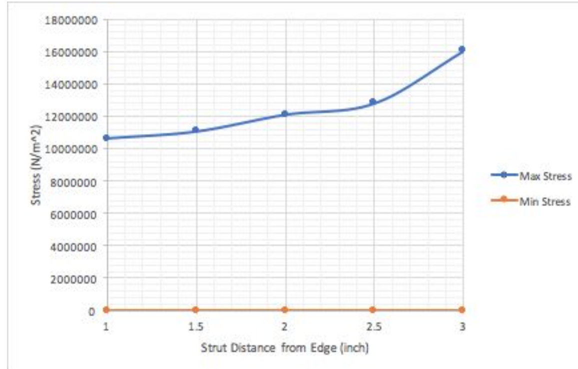


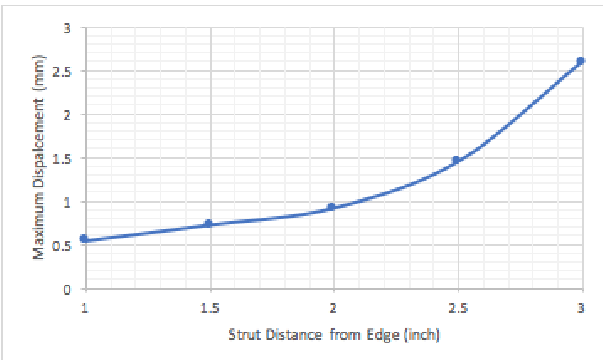


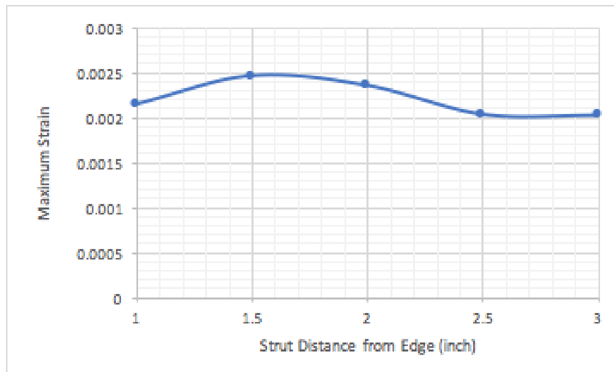


We also plotted the results as shown below. We noticed that the factor of safety ranged from about 2.5 to 4.5. These values were good enough for us and we chose 1.5” which gave us FOS of 4.1. This gave us enough space for the internal components.

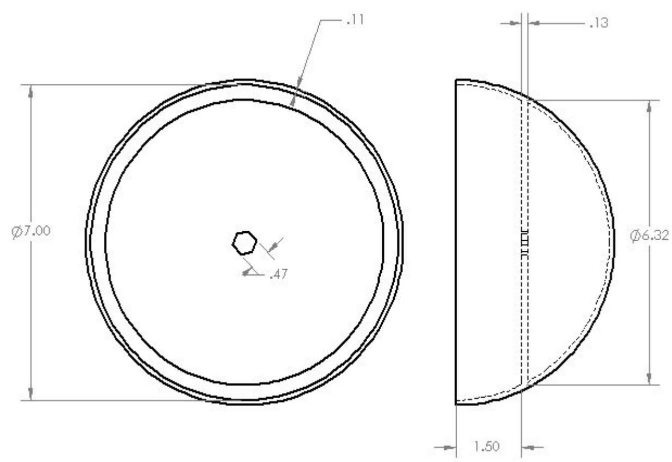








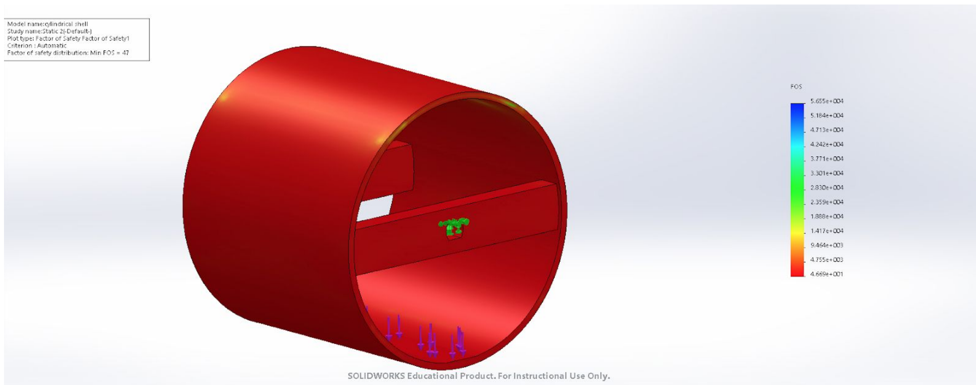
So our final dimensions for the wheels are as shown in the engineering drawing below. The strut is ⅛" thick and will be laser-cut in acrylic. The tolerance is ±0.05".

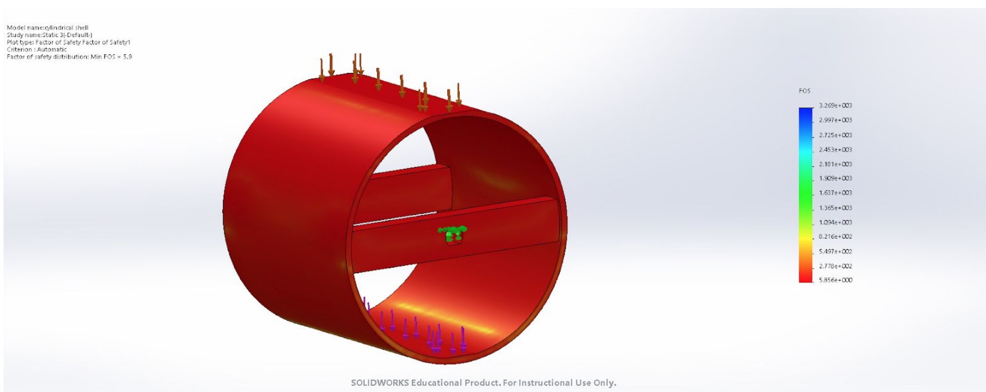


7.1.2 Cylindrical Shell Static Stress Analysis

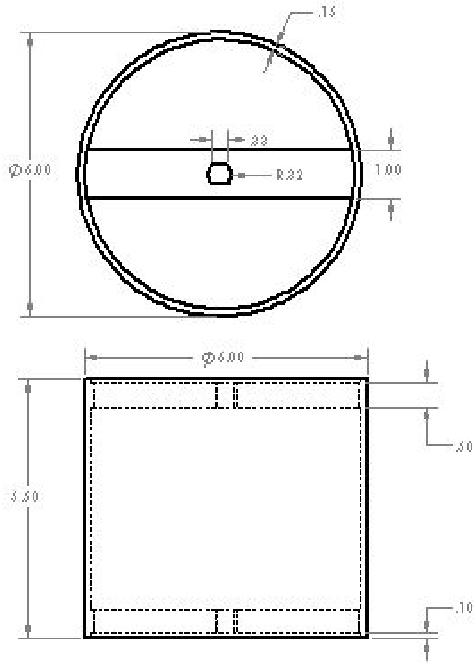
We planned to purchase an off-the-shelf 6" diameter acrylic cylindrical shell in order to simplify manufacturing process. In order to attach motors to the cylinder, we added a strut to both ends of the cylinder. We did FEA on the whole cylindrical shell structure with struts in order to ensure that it can resist the force applied by the dog. We conducted two simulations: one with just the weight of the system and one with a 40 lb dog pushing down on the cylinder.

For the simulation setup, we fixed the hole where motor is placed since the motor will be kept in place by the wheels. We applied 3 lb downward force at the inner bottom part of the cylinder to simulate weight of the system due to internal components. We applied 40 lb downward force at the top of the cylinder to simulate a dog's weight. With the dog's weight included, the factor of safety came out to be 5.9 which was more than sufficient.





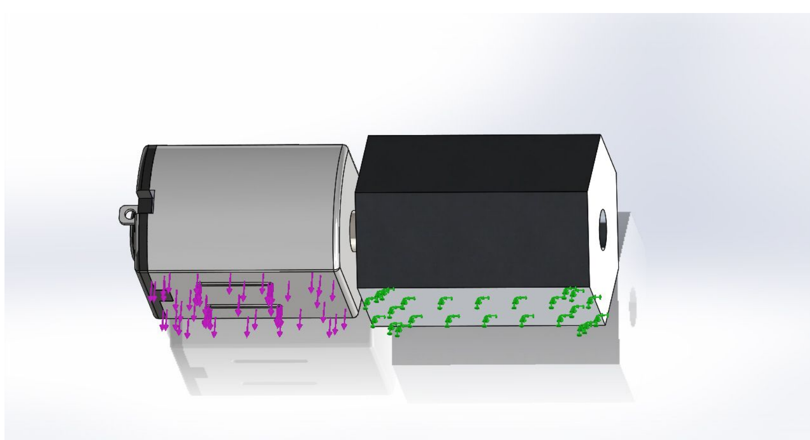
So our final dimensions for the cylinder are as shown in the engineering drawing below. The strut is 1/2" thick and 1" wide and will be laser-cut in acrylic. The tolerance is ±0.05".



7.1.3 Motor Shaft

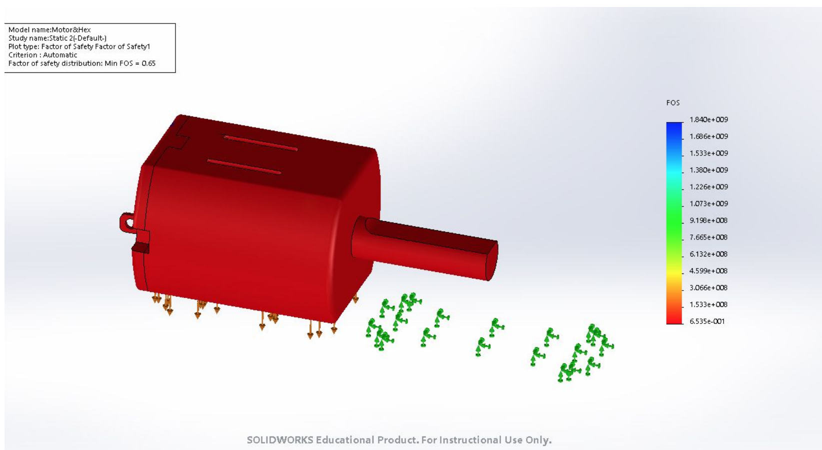
We realized that the motor shaft was only 3 mm in diameter. In order to ensure that the shaft can resist the weight of the system as well as possible external force due to a dog's weight pressing down on the system, we did FEA on the motor shaft.

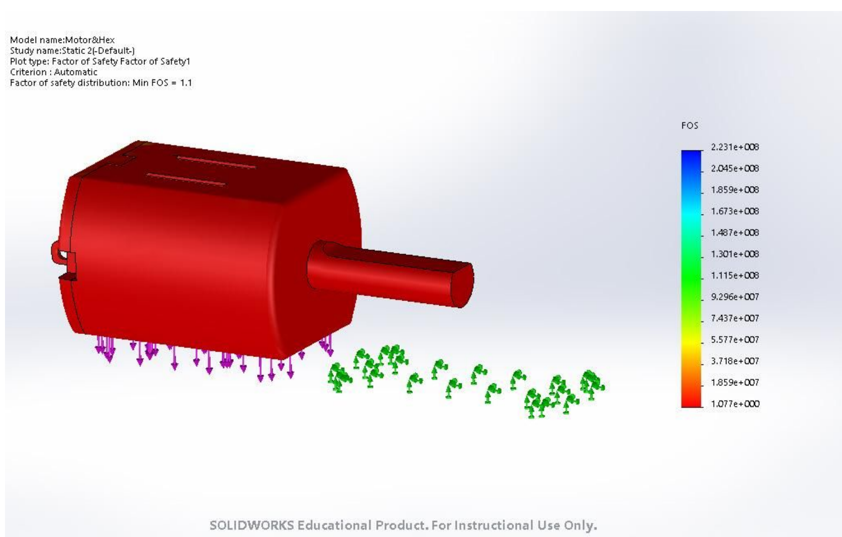
For the simulation setup, we attached the hex adapter on the shaft and fixed the bottom of the hex adapter since it is sitting on the wheel strut. We applied downward force on the motor body which is the part that is sitting on the struts for the cylindrical shell. The table below shows the results of applying difference forces.

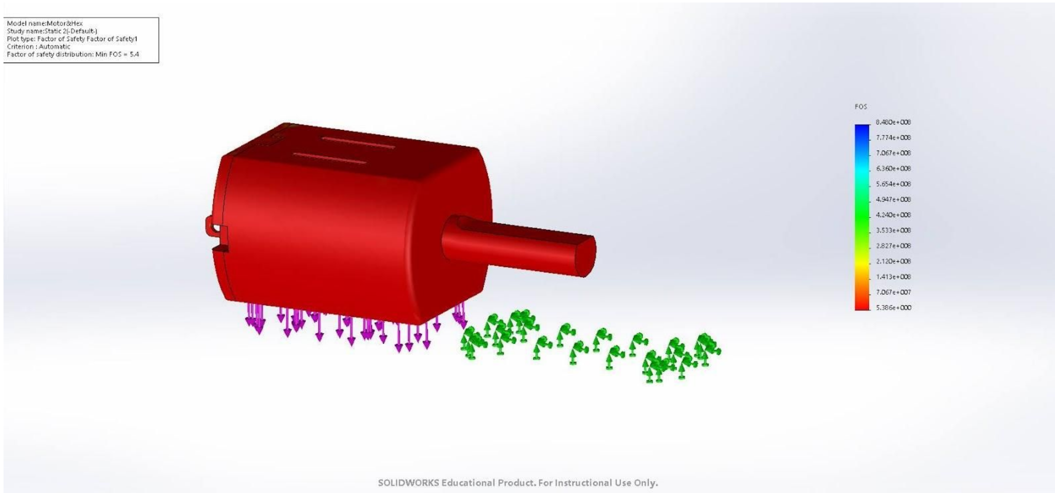


Applied Force (lbf)	Min. FOS	Max Strain	Max Displacement (mm)	Max Stress (10^8 N/m^2)	Min Stress (N/m^2)
40	0.65	0.0007049	0.04837	4.198	301.1
15	1.1	0.0004276	0.02934	2.546	182.4
3	5.4	0.0000855	0.0059	0.5093	36.64

When we applied 40 lb force, the FOS came out to be only 0.65. With just the system weight of 3 lb, we got FOS of 5.4. So we tried 15 lb force which corresponds to a small-sized dog. This gave us FOS of 1.1 which was good enough. From this FEA, we decided to limit the dog weight to less than 15 lb.



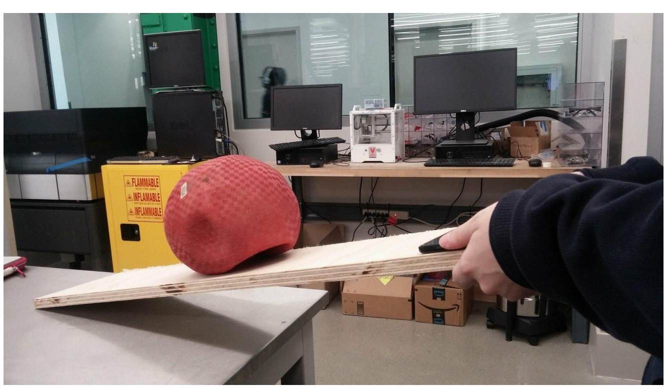




7.1.3 Coefficient of Friction of Rubber

To make sure that the friction between the rubber we want to use for the wheels and the ground is high enough, we conducted a brief test to determine the coefficient of friction. We used the dodge ball rubber and placed in on a plate of plywood which simulated a wooden floor. We did five consistent trials of measuring the minimum angle of tilt of the plywood at which the dodge ball started to slide. Below is a table that summarizes the results. The average coefficient of friction came out to be 0.54 with a standard deviation of 0.05. We concluded that this is high enough friction.

Degree	Radian	Coefficient
28	0.4887	0.5317
31	0.5411	0.6009
26	0.4538	0.4877
30	0.5236	0.5774
27	0.4712	0.5095



7.1.4 Drive System Dynamics

Due to the design of the drive system, the middle section will rotate whenever we apply a torque onto the wheels. We wanted to make sure that angular rotation is sufficiently low to allow the user to comfortably see out of the camera. Thus, we created a Matlab model that described the system dynamics, and ran an ODE solver to simulate how the system would accelerate.

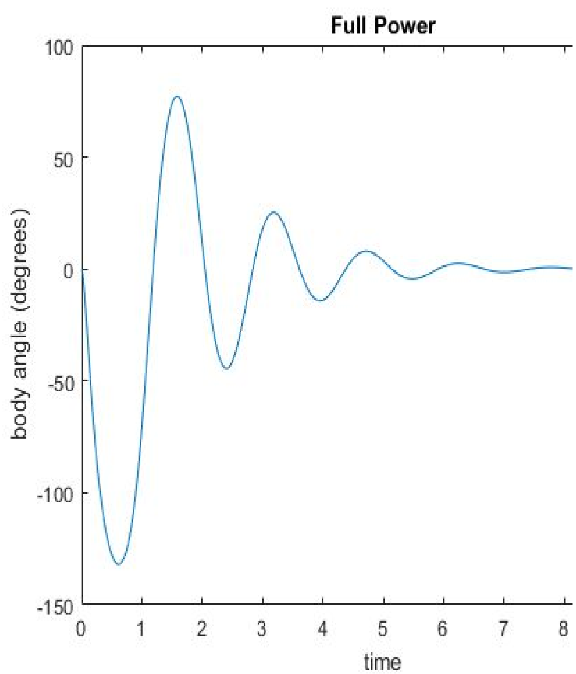
We assumed a linear motor model, and used the following equations of motion for the cylindrical body and the two wheels.

$$\ddot{\theta}_b = \frac{\tau - m_b g r \sin(\theta_b)}{I_b}$$

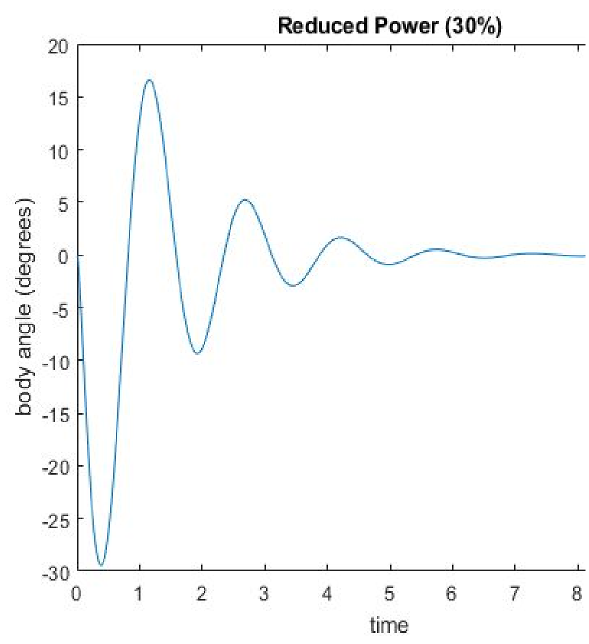
$$\ddot{\theta}_w = \frac{\tau - F_{friction} r}{I_w}$$

$$F_{friction} = m_{total} a = m_{total} \ddot{\theta}_w r$$

Body angle rotation at full power



Body rotation at 30% power



As we can see, at full power, the system is not very stable, and the center body would rotate at a nausea-inducing 130 degrees. Therefore, to reduce the rotation, we reduce the motor output. At 30%, the body rotation is reduced below 30 degrees, which we consider sufficiently low for comfortable remote driving.

In conclusion, using the simulation, we know that we can balance the system by reducing the power of the motors.

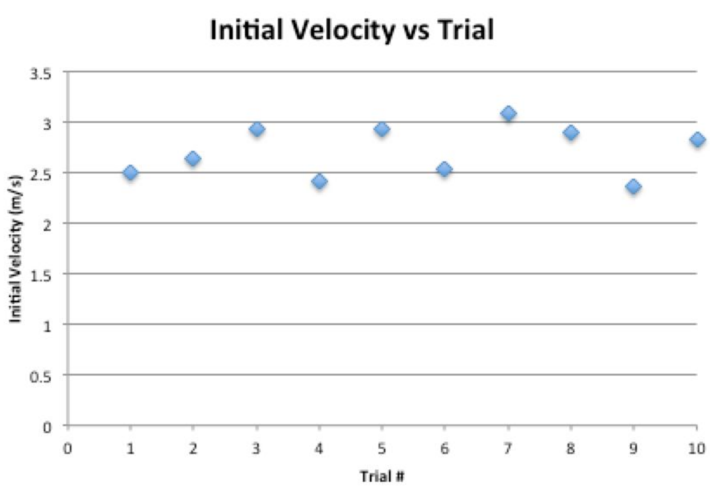
7.2 Treat Launcher

Treat launcher design started with the selection of a spring. This is done by setting a desired velocity, the treat mass, and the spring displacement. Energy balance is used to compute the required stiffness value.

$1/2\ mv2 = 1/2\ kx2$
$m = 0.001kg$, $v = 1m/s$, $x = 0.02m$
Solved: $k = 2.5\ N/m$

We can then select a soft spring with similar stiffness to the calculated value, and verify its effectiveness in physical testing.

Further analysis was done using experimentation values of height measurements from vertical launches.
$mgh = 1/2\ mv2$
$v = 2gh$

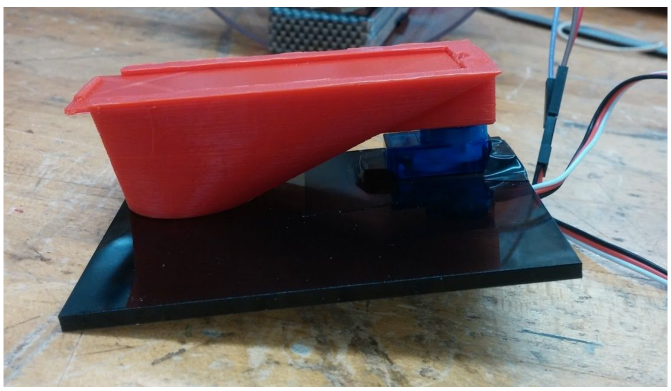


With a starting height of 0.076cm (3 inches) and a marginal horizontal distance of 0.5m, and use the initial velocity of 2.5m/s, the minimal angle can be calculated as follows:
$y = vt\ \sin() - g2t2 = -0.076$
$x = vt\ \cos() = 0.5$
$= 0.26\ rad = 14.9$ degrees

From these data, we can conclude that our treat launcher is sufficiently powerful. We can easily reach the marginal value for launch distance, and our initial velocity is quite high. We will continue to use this design for prototype 2.

7.3 Treat Loader

With the final treat loader design, we conducted 10 trials to test whether or not all treats would filter out completely. We started with the container completely filled with dry treats and swinging the container over the hole on the stationary plate until all treats were dispensed. The table below shows the results of the experiment. Only in three out of 10 trials some treats were permanently clogged and could not filter out. These three cases were when the treats were particularly shoved into the container more randomly. From this experiment, we learned that placing the treats in a line instead of randomly shoving them in is important in reducing jamming. Although, it would give the user more rule to follow, we decided that this design was good enough for the project.



Trial	Success
1	Y
2	N
3	Y
4	Y
5	Y
6	Y
7	N
8	Y
9	Y
10	N

7.4 Web App

We conducted several experiments to analyze our design choices regarding web app and user control, including:

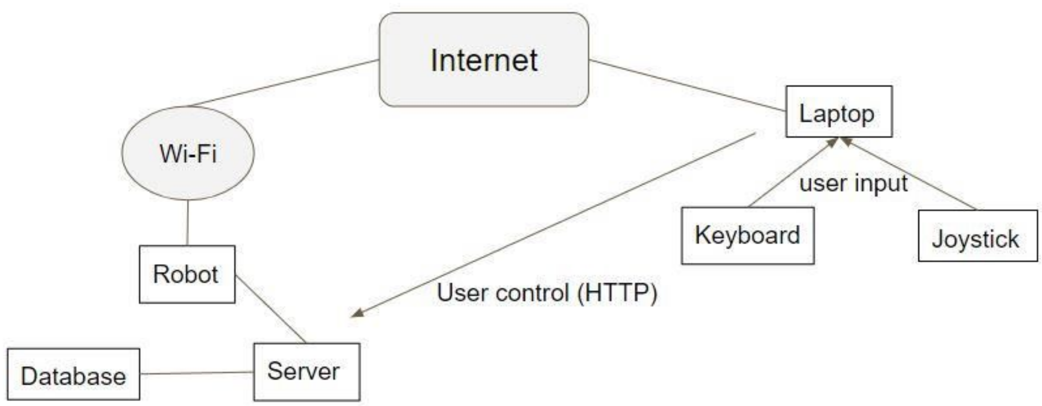
1. Testing the latency of Raspberry Pi connected to CMU Wi-Fi to determine if there would be observable delay between user input and device execution of commands.
2. Testing the upload speed of Raspberry Pi connected to CMU Wi-Fi to determine if the speed could support real-time video streaming at a speed of about 1Mb/s from device to user.
3. Testing the user input filter to determine if the communication channel would be jammed by continuous user input.

For experiment 1, our goal is to determine if the total latency between user input and device execution of command would be so large that user would observe a visible delay. The total latency in this process is the sum of the following:

1. Time spent by browser to send command to Internet
2. Time spent by server on Raspberry Pi to obtain command from Internet
3. Time spent by device to process and execute command

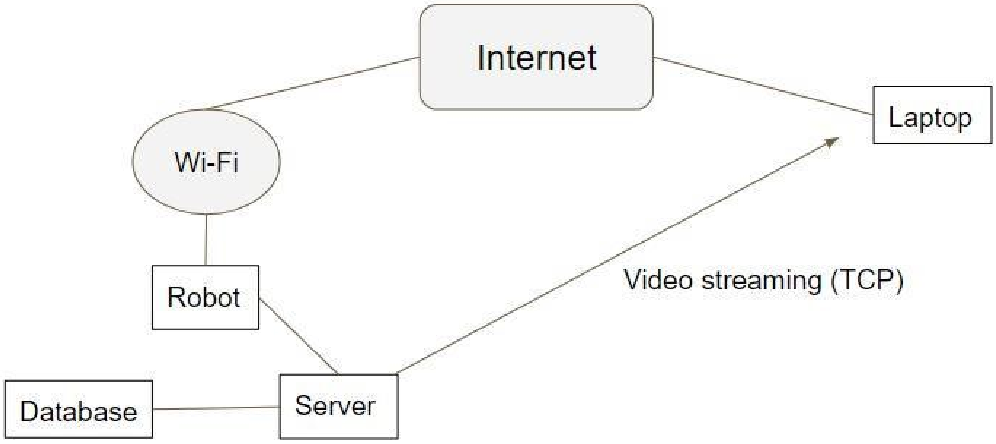
We know that modern browsers usually come with negligibly small latency, and the microcontroller also takes negligibly small amount of time to process and execute command. Thus, the majority of latency would be induced in step 2, when our server on Raspberry Pi downloads command from the Internet.

We used speedtest.net to test the latency of our Raspberry Pi connected to CMU Wi-Fi, and the result was about 92ms, which is well below the threshold that human could notice. Thus, we could conclude that user would likely not notice any delay between user input and device execution of commands and it supports our design decision to host our server on Raspberry Pi. The architecture diagram for user control is shown below:



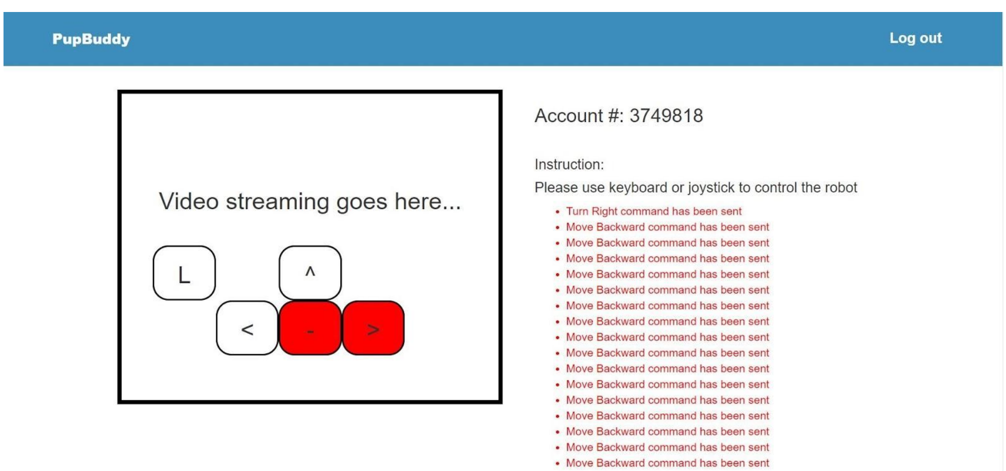
For experiment 2, our goal is to determine if the upload speed of our Raspberry Pi would be sufficient to support video streaming to user, which would require an upload speed of about 1Mb/s at 600 by 800 resolution and 24 frames per second.

Similar to experiment 1, we used speedtest.net to test the upload speed of our Raspberry Pi connected to CMU Wi-Fi, and the result shows that the average upload speed was about 1.20Mb/s, which is slightly more than the required upload speed. Thus, we could conclude that a server on Raspberry Pi would support live video streaming for users and it supports our design decision to host our server on Raspberry Pi. The architecture diagram for video streaming is shown below:

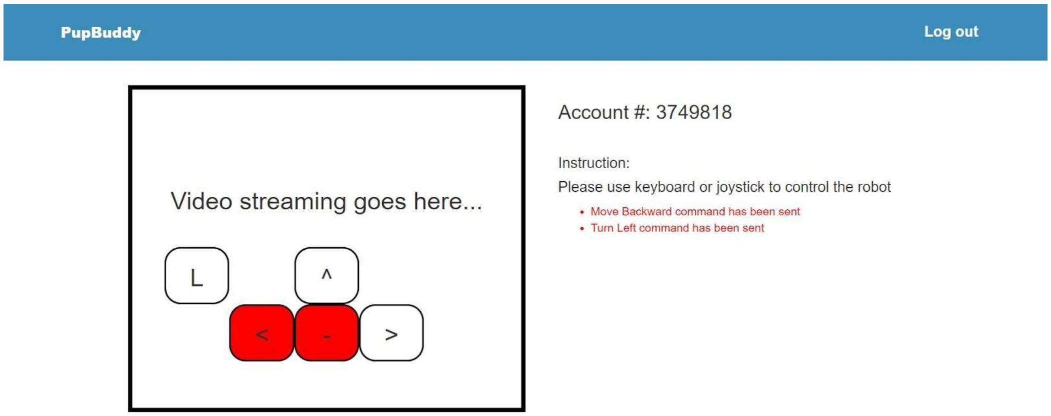


For experiment 3, we were facing a problem that when a user presses a button on the keyboard for an extended period of time, multiple commands would be sent and it would jam communication protocol and confuse the device. Thus, we imposed certain rules on user inputs and tried to filter them before sending commands to device.

Before filtering, when a user presses a button for an extended period of time, the system would behave as shown below:



After filtering, even if a user presses a button for an extended period of time, only one command would be sent. The system would behave as shown below:



We observed that the filtering system could effectively prevent identical commands from firing multiple times, and such system would support our design decision to use keyboard input as our primary user control method.

7.5 Failure Modes and Effects Analysis

Failure Mode	Effect	S	Causes	O	Current Controls	D	RPN	Mitigation Plans
Treat loader jams	Treat does not dispense	4	Multiple treats come down at once, spring gets stuck	7	Unjamming control mode	7	196	Unjamming control mode, robust design, add spring to push the treats out
Wheels coming off	System cannot move, electronics exposed	6	Dog bites the wheel aggresively	5	Secure attachment with screws, camera for monitoring	2	60	Secure attachment with screws, camera for monitoring, install remote shutdown mode
Overheated Chips	Raspberry Pi stop working/catch fire	3	Executing too many tasks; high temperature	4	Add external heat sinks	4	48	Monitor the CPU frequency and keep it below a threshold; heat sink and/or fan
Wheels breaking	Unpredictable movement of the system, broken pieces can choke the dog	7	Excessive stress on the wheels	3	Design to overcome stress	2	42	Design to overcome stress
Too many treats being dispensed per cycle	Launcher may jam	2	Opening for the treat loader is too large, opening interval is too long	4	Small hole for the opening	5	40	Small hole for the opening, refinement of treat loader design
Motor shaft breaking	System stops moving	5	Dog applies too much weight on the system	3	Need to manually detach wheels and replace shaft	2	30	Set recommended weight limit for dogs
User control latency too long	Delay between user input and robot execution	2	Latency in Raspberry Pi Wi-Fi or server	5	Use Wi-Fi network with low latency	3	30	Make sure server side codes can be executed with low latency
Unfiltered user control input	Communication channel jammed with identical commands	2	User presses a button for a period of time	10	None	1	20	Filter user input
Wifi disconnection	User cannot control the system	8	Problem with Wi-Fi network or Wi-Fi module on Raspberry Pi	2	None	1	16	Contigency code: when Wi-Fi disconnects, stop all robot actions
Camera view is blocked	Cannot see the dog or the environment	5	Debris on the camera, dust	2	Isolated camera location	1	10	Isolated camera location, better surface finish on shield plastic, clear camera view
System flips over	Feeder stops functioning because it relies on gravity	2	Dog crashes into the system	2	Designed to passively upright itself	1	4	Pendulum design

The failure mode with the highest RPN of 196 was treat loader jamming issue. Based on the tests we have conducted on the treat loader, we realized that the occurrence of jamming is pretty high. It is difficult to detect for a user who is remotely controlling the system. While we gave a severity of 4, treat jamming disables one of the main functions of the system so it is important to address the issue. Based on our tests, the main cause was due to multiple treats piling up on top of each other. We plan to mitigate this failure mode first by designing exclusively for one type of treats to prevent the treats from piling up. If this fails, we plan to add some agitator that shakes out the treats or a spring to push out the treat one by one.

The second highest failure mode was wheels coming off with an RPN of 60. The severity of it was particularly high since the system would become immobile and the internal electronics would be exposed when the wheels come off. This may happen when a dog bites off the wheels aggressively. As a mitigation plan, we plan to make the attachment of wheels and motors particularly secure. The onboard camera can also help the user to detect if the dog is biting off the wheels and try to stay away from the dog in such situations.

The third highest failure mode was chips overheating. This could happen due to sending too many commands to the microcontroller which can cause temperature to rise. As a result, the microcontroller can break or even catch on fire. As a mitigation plan, we plan to install heat sink and monitor the CPU frequency to keep it below a threshold.

The fourth highest failure mode was wheels breaking which can happen due to dog pressing down upon the wheels. To account for this issue, we added strut to the wheels to increase resistance to compression. Moreover, we determined to limit the dog's weight to less than 15 lb. Other lower RPN failure modes are shown in the table above.

7.6 Design for Environment

Some of the materials used in the product may be harmful to the environment. Lithium batteries contain potentially harmful materials which can produce environmental pollutions. Solders on the electronics can leach lead into the soil when thrown out in landfill. During the production stage, plastic injection molding can produce parts with defect. The energy and time consumed to make these parts become wasted. Also, if the factory layout is inefficient, transporting molded parts from the molding machine to a storage area can take longer time. In the distribution stage, styrofoams used for packaging contain carcinogen called styrene. During use, injection molded polycarbonate can leach bisphenol A (BPA), chemical used to make plastics, which can cause heart disease, diabetes, and infertility. Rubber, which is used as cover the wheels, biodegrades slowly and would build up if thrown into the landfill after use. Similarly, electronics are non-biodegradable and, when incinerated, electronics release chlorinated dioxins, hazardous chemicals.

To minimize impacts in the materials stage of the life cycle, we should use lead-free solders for circuits and recyclable materials for components in order to reduce non-biodegradable materials from being dumped into the landfill. In the production stage, we can design the system for weight and install long lasting battery so that the overall system consumes less power. For distribution, we can use only cardboard packaging instead of foams to reduce exposure to carcinogens. During use, we can provide a method of system charging via renewable energy such as solar or wind energy. In the recovery stage of the life cycle, rubber should be refurbished for reuse or incinerated which would provide energy for use during manufacturing processes. Similarly, polycarbonate can be melted down and reused.

7.7 Manufacturing and Assembly Techniques

7.7.1 Bill of Material

Component	Total
Raspberry Pi	$34.80
2 Servo motors	$15.90
2 Gear motors	$31.90
Motor shield	$23.25
LiPo battery	$49.99
Battery charger	$39.90
Pi camera	$22.75
PC protective case	$6.90
SD card reader	$6.59
UHS-I card with adapter	$12.12
Micro USB adapter	$2.79
Dean's T adapter	$8.99
XT60 to USB adapter	$10.89
Motor shaft hub	$3.95
Voltage regulator	$30.11
2 Acrylic hemispheres	$45.98
Cylindrical shell	$10.99
Dodge ball	$13.20
Acrylic plate	$2.91
2 Acrylic wheel struts	$2.92
2 Resin struts	$18.30
Launcher 3D print	$0.30
Loader 3D print	$0.90
Spring	$0.15
Total	**$376.83**

7.7.2 Mass Production

For mass production, most of the components including wheels, launcher and loader will be injection molded in polycarbonate. Printed circuit board will be made to integrate the controller, motor shield and voltage regulator in one piece. To reduce cost of assembly, the struts will be integrated to the wheels and the cylindrical shell. The launcher will be one piece with the plate for the loader also integrated to it. To maximize ease of assembly, the system will use only one type of screw with threaded inserts and most of the parts will have snap-fit feature. The system will contain an external switch and charging port so that it does not have to be disassembled to power on or charge the battery. A cheaper controller with GPU will be selected instead of Raspberry Pi. Lastly, a cloud server will be setup as opposed to using the Raspberry Pi.

7.7.3 Manufacturing Methods

For the final prototype, we bought off-the-shelf acrylic hemisphere and cylinder for the wheels and center casing. We used laser cutter and drill press to cut out holes on acrylic parts. Most of the flat acrylic parts such as the struts and the separating plate were laser cut. We 3D printed the loader and launcher using PLA and struts using resin. Most of the wires have been soldered. We mainly used hot glue or super glue to attach the struts to the wheels, put pieces of the launcher together, and attach the rubber to the wheels. We used velcro to lock the lid onto the cylindrical shell. We used multiple types of screws for the struts and hex adapter for the motor shaft. Different electronic components such as the controller, motor shield and voltage regulator were separate pieces and wired together. Lastly, everything was hand assembled.

Unlike the final prototype, in mass production, we will replace all of the laser cutting and 3D printing with injection molding. Instead of using velcro, hot glue or super glue, we will make most of our parts snap-fit. We will use hot melt adhesive to attach the rubber to the wheels so that it can be easily detached when necessary. We will try to replace any screw attachment with snap-fit attachment. If any, we will use only one type of screw for the entire system. Instead of having multiple pieces of electronic circuits, we will integrate them into one printed circuit board.

The new manufacturing methods come with some challenges. For instance, in order to design for injection molding, we need to add ribs and supports to the parts, maintain constant wall thickness, and prevent shrinking and warping of the parts. Tolerance control is another challenge during manufacturing, especially for parts that use snap-fit. There is cost for injection molding and to train workers to assemble the parts. Layout of the facility for efficient manufacturing can pose a challenge as well.

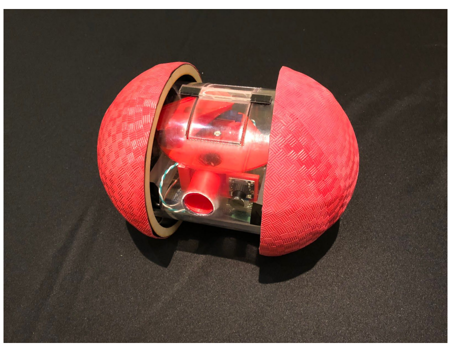
8. Final Prototype Description

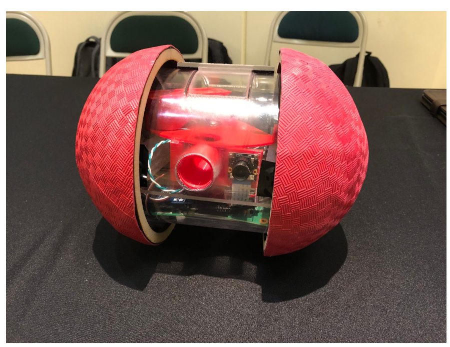
The final prototype retains most features from prototype 2. It is composed of two hemisphere wheels and a center, cylindrical shell that houses the loading and launching subsystems, as well as the electronics that enable those subsystems. The hemispherical wheels are acrylic covered by dodgeball rubber material to increase friction. Each wheel also has a laser cut wooden rim to improve aesthetics and enhance structural integrity. The cylindrical shell is also made from acrylic. It has two additional openings -- both laser cut -- for putting pet treats into the loader and shooting those same treats out of the launcher. Two 3d printed resin struts, one for each wheel, are screwed in to the shell to support the DC motors that power the wheels.

The electronics are housed underneath the plate that holds the loader and launcher assembly. The battery, being the heaviest component, is placed at the bottom to optimize for stability. The raspberry pi (with the motor shield attached) and the 12V to 5V regulators are secured on top of the battery.

When the user wishes to use PupBuddy, he/she would need to first remove the left wheel, which is attached by press fit. Then the user would attach the battery cable (with a dean-style T connector) to the cable leading to the voltage regulators. The user then puts the wheels back. To launch treats, the user would remove the top hatch, which is secured by velcro straps, and push on the sliding door of the loader. Treats can then be inserted through the opening. When the hardware preparation is complete and the system is powered on, the user can then connect his/her laptop to the internet at its IP address: 128.237.247.17:80, assuming that the server has been set up.

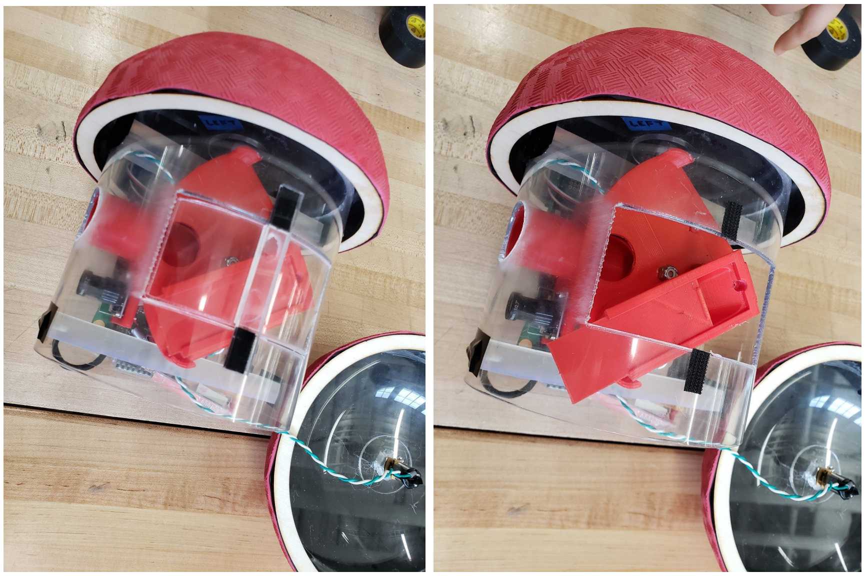
8.1 Demonstration of Design





Pictures of final prototype

Picture of PupBuddy with left wheel removed, showing a strut and the electronics



Pictures of PupBuddy with loader hatch closed (left) and open (right)

A short video of the working system demonstrating video streaming, device movement and treat launching is recorded and the video can be accessed through this link: https://youtu.be/2p1DyIICegY.

Another longer video of the working system, demonstrating specifically its mobility capability can be accessed through this link: https://youtu.be/Jph2ILvdFYs.

8.2 Testing and Results

A number of tests have been conducted to examine the software component of the system, including latency test, upload speed test, and user input filtering correctness test.

Latency is crucial for our system as a key feature of it is real-time, and any latency more than 200ms would cause user to notice observable delay between command input and command execution, or notice observable delay in real-time video streaming. On the other hand, upload speed for the system is also important as the 600 by 800 resolution at 24fps video streaming feature needs a minimum of 1Mb/s upload speed. Using the service provided by speedtest.net, we were able to measure the latency and upload speed of the server running on Raspberry Pi connected to CMU Wifi network.

The results are shown below:

Latency	Upload speed
90.81ms	1.21Mb/s

Table of latency test result and upload speed test result

Note that latency is lower than the threshold of 200ms and upload speed is higher than the threshold of 1Mb/s.

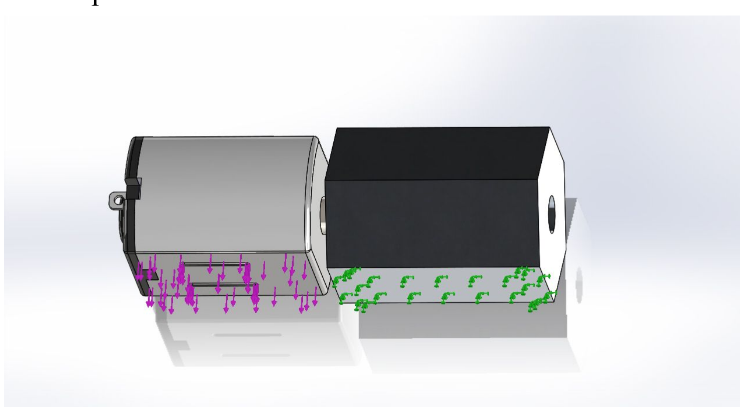
As discussed in the previous section, user input from keyboard would be filtered based on the logic outlined in the state machine, and we conducted test to examine the correctness of user input filtering. We categorized possible user inputs into three categories, one input, where the input should be executed, two inputs, where both inputs should be executed simultaneously, and three or more inputs, where only the first two inputs should be executed simultaneously, and all other inputs should be ignored. The correctness of the input execution was measured and summarized in the table below.

	Total inputs recorded	Number of Inputs correctly filtered and executed	Number of inputs incorrectly filtered and executed
1 input	20	20	0
2 inputs	20	20	0
3+ inputs	20	20	0
total	60	60	0

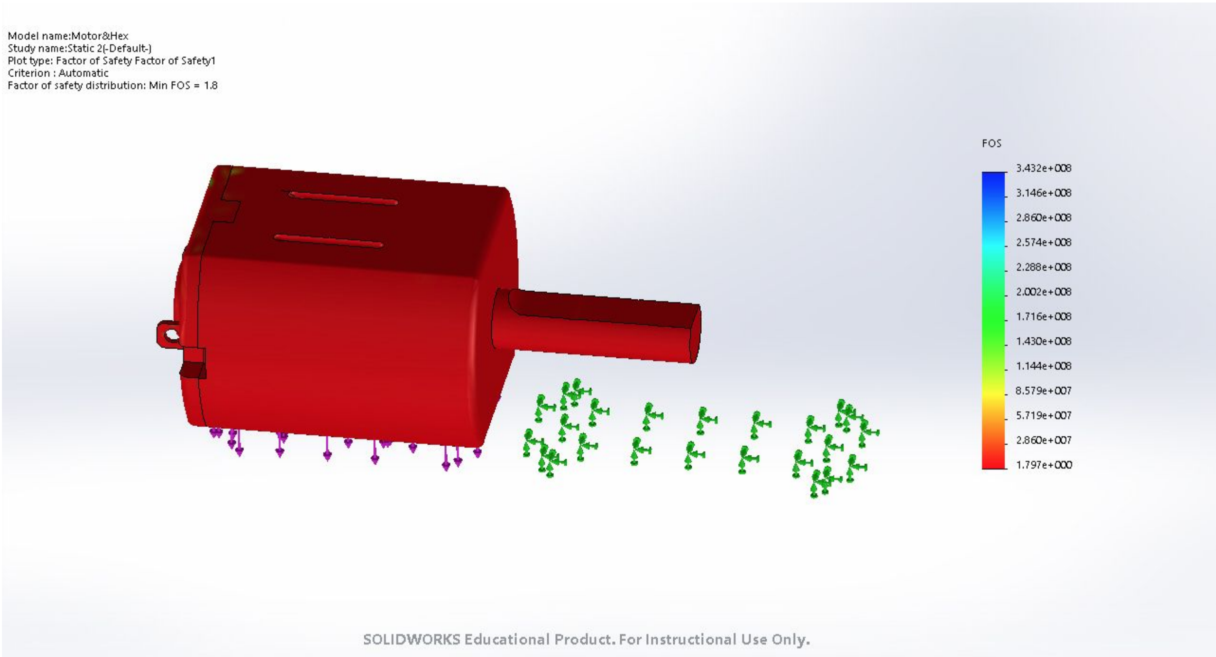
Table of user input filtering correctness test results

Note that the system is able to filter and execute user input with a 100% correctness rate in all cases.

The structure of the final prototype did not change significantly from prototype 2. Its weak points remain the same, and so the same computer simulations can be used to determine the allowable loads and factors of safety. The biggest weak point is located at the motor shaft.



FEA setup with fixed hex shaft and a force on the motor



Factor of safety plot for FEA simulation

Applied Force (lbf)	Min. FOS	Max Strain	Max Displacement (mm)	Max Stress (10^8 N/m^2)	Min Stress (N/m^2)
40	0.65	0.0007049	0.04837	4.198	301.1
15	1.1	0.0004276	0.02934	2.546	182.4
3	5.4	0.0000855	0.0059	0.5093	36.64

As shown by the results above, PupBuddy is able to support at a dog whose weight does not exceed 15lb.

The tilting analysis of the final prototype does not differ from that of prototype two, since the analysis of prototype 2 is done with the final setup in mind. By setting up differential equations that describe the cylinder’s tendencies to tilt, an ODE45 simulation can be run.

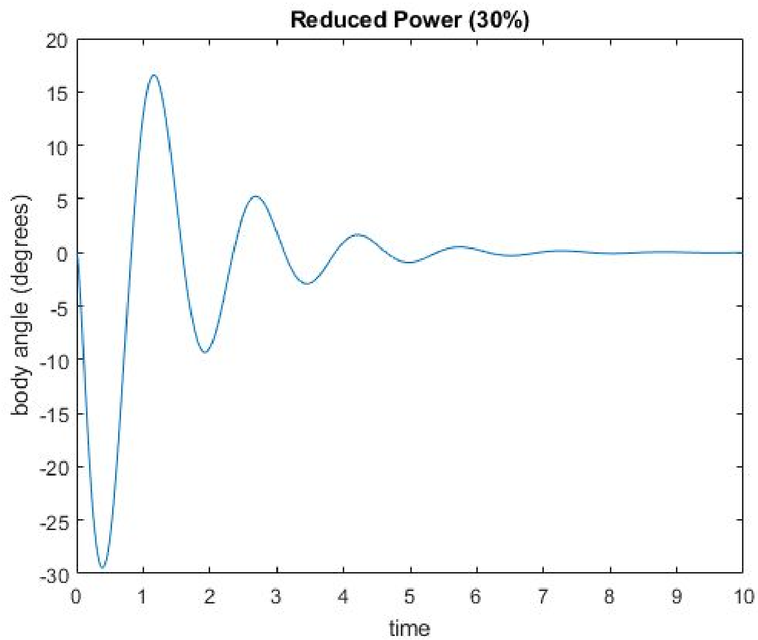


Figure showing rotation of the body due to moving the wheels at reduced power

The above figure shows that lowering the motor power to 30% will give an acceptable (<30°) tilt, and that the tilt will dampen out few oscillations. In real testing, we found out that the damping is higher than in the model, since the motors' gearboxes require significant torque to backdrive, resulting in a slower and smaller oscillation.

Physical capabilities of the final prototype were measured to examine the effectiveness of the loader, launcher, and mobility subsystems.
Velocities were tested by running the robot at the maximum speed (while maintaining acceptable tilt angle) over a fixed distance. Time is measured on a stopwatch.
Launcher and loader were tested together. The launch distance is distance the farthest treat reaches before it hit the ground.
Battery life was measured by running the prototype continuously until the battery voltage drops below 10V (as indicated by the LCD screen on a voltage regulator)

Metric	Value
Linear velocity	0.57 m/s
Turn velocity	166°/s
Tilt angle	30°
Horizontal launch distance	0.16 m

Treats per launch	1.8
Treat capacity	10
Battery life	>1.5 hours constant use

Overall, the physical testing data shows that PupBuddy's subsystem performances are good. It can move at a decent speed. It won't be able to chance after a fast dog, but it is definitely fast enough for pet owners to have meaningful interactions with their dogs. The turn velocity is excellent, and there is no tilt at turning, so the robot is agile and nimble. The launching distance is adequate, as the treats will definitely go far enough to catch a dog's attention. Furthermore, the launcher test excludes bouncing, which significantly increases the treat launching distance on hard surfaces. The only caveat of PupBuddy is that the loader's capacity is quite low, as it can store at max 10 large treats. This would limit the users' options to interact with their dog since they would likely shoot all the treats very quickly. In terms of battery life, the results are surprisingly good. We expected a 1 hour battery life, with a minimal viable value of 30 minutes, but the robot is surprisingly efficient, and continues to perform even after 1.5 hours. By that time, the voltage of the battery drops to 10V. The lowest voltage that our 3 cell LiPo battery could reach safely is 9V, but we decided not to push its limits.

9. Conclusions

As a team, we learned that selecting concept based on the scope of the team's knowledge and skills as well as manufacturability makes the project much more achievable within the time constraint. A large portion of the design process was spent on learning the skills we lacked, for example, using Raspberry Pi and figuring out how to make it work. We also learned that when purchasing parts, taking time to select the right components such as battery and motors saves more time then buying the incorrect one and repurchasing. Also, for parts that are difficult to manufacture such as the cylindrical shell or the hemispherical wheel, it is good to purchase off-the-shelf parts. Moreover, it is important to leave sufficient time to order materials and manufacture because these typically take at least a week. In terms of building, we learned that it is efficient to focus on resolving the more challenging tasks first. In our case, we made a good decision of quickly figuring out how to use Raspberry Pi by Prototype 1 and finalizing the launcher and loader designs by Prototype 2. Lastly, many of the parts were interdependent. For instance, we needed the controls completed in order to verify the functionality of the drive system. Thus, we learned that it is important to think ahead and determine the order of tasks.

One interesting thing we learned is that a seemingly simple subsystem can pose a challenging problem. We spent more time finalizing the loader design than the launcher design due to treat jamming issue. We had to explore multiple designs and conduct multiple experiments to figure out the optimal design that would minimize treat jamming. Starting off with a flat bottom, we moved to a funnel design and narrowed the space to reduce treat clogging. Then we decided to swing the container over a stationary plate to create vibration in the loader.

The integration of loader and launcher also initially caused an issue because the spring was too long which meant that we had to compress the spring first, load the treat and then release the spring. This made the implementation of control more difficult due to error accumulation in servo rotation. We decided to solve it mechanically by replacing the spring with shorter one and quickly redesigning the spring holder, which saved us more time because tuning power level for the servo would have taken much longer time. Resetting the launcher was another issue we faced because the disk that pulls the spring occasionally got stuck at specific orientations. This time, we found a software solution by commanding the servo to revert back to an initial position after launching.

Accessing internal components posed yet another challenge. We resolved this issue rather unexpectedly by tight fitting the motors into the struts. This made the wheels easily slide in and out of the strust and thus made the internal components more easily accessible without needing to use any tools.

One of the things we would have done differently is getting a servo with encoder to make rotations more accurate for loader and launcher. Moreover, installing an external charging port and a switch would have reduced the need for us to disassemble the wheels every time we had to charge the battery or power the system. If we had more time, we would also have redesigned the loader to increase treat holding capacity. Lastly, we would have integrated accelerometer to further optimize the tilt angle of the cylinder so that the camera view is not too blurry during motion.

There are some unresolved issues in our system. The treat loader currently has very small treat holding capacity and dispenses multiple treats at a time as opposed to one. These are important aspects that need to be addressed so that users have more control over how much treats to give to their dogs. Our current camera has low resolution and the latency may be slower than desired by consumers. This would definitely need to be improved for the final product. Furthermore, the wheel attachment has to be more secure so that it doesn't become loose easily especially when interacting with dogs. Lastly, as mentioned earlier, our current system does not have the best charging and powering method because the user has to take one of the wheels off to access the charger. Installing external charging port and a switch would fix that issue.

For future design teams who would continue our project, we would suggest that they find a better way to secure the wheels and protect the motor gearbox which becomes loose very easily. Also, to reduce the overall size of the system, they can build a printed circuit board to integrate controller, motor shield, voltage regulator and other electronics into one piece. As for the treat loader, we would suggest increasing the treat holding capacity and do more tests to figure out how to release one treat at a time. Additional feature they can add is speaker that would allow the dog owner to talk to their dog.

Overall, our problem description, planning, and concept generation and selection were good, which helped us make our system work. Using laser cutter and 3D printer which are a few of the most accessible machines on campus was a good choice as well given the resource and time constraint. In the end, we have successfully created a minimum viable product, unique among all competitors, that performs all three functions: pet feeding, monitoring, and remote interaction.

References

8.1 Competitors

1. https://mygobone.com/?v=7516fd43adaa
2. https://www.getpebby.com/
3. https://shopus.furbo.com/
4. https://petnet.io/smartfeeder

8.2 Literature (Technical Patents)

1. https://patents.google.com/patent/US7703447B2/en?q=treat&q=launcher&oq=treat+launcher
2. https://patents.google.com/patent/USD782125?oq=furbo
3. https://patents.google.com/patent/US6817351

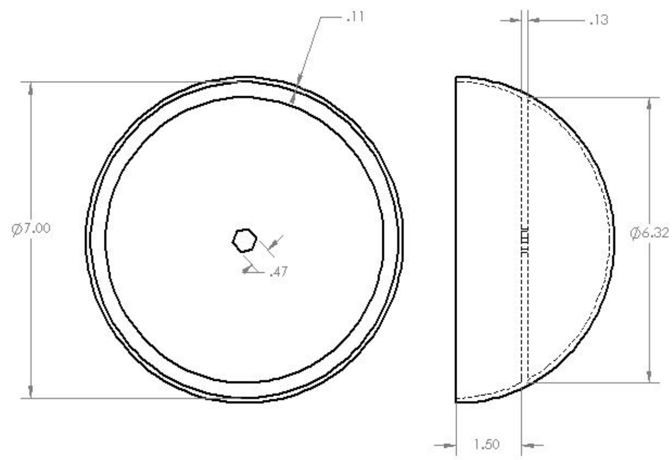
8.3 Design for Environment and Manufacturing Methods

1. http://homeguides.sfgate.com/rubber-recyclable-79758.html
2. https://www.improve-your-injection-molding.com/7-wastes.html
3. https://greenliving.lovetoknow.com/How_Styrofoam_is_Bad_for_the_Environment
4. https://ieeexplore.ieee.org/document/5712793/
5. https://www.doityourself.com/stry/the-environmental-impact-of-electronic-waste
6. https://www.ncbi.nlm.nih.gov/pubmed/23638841

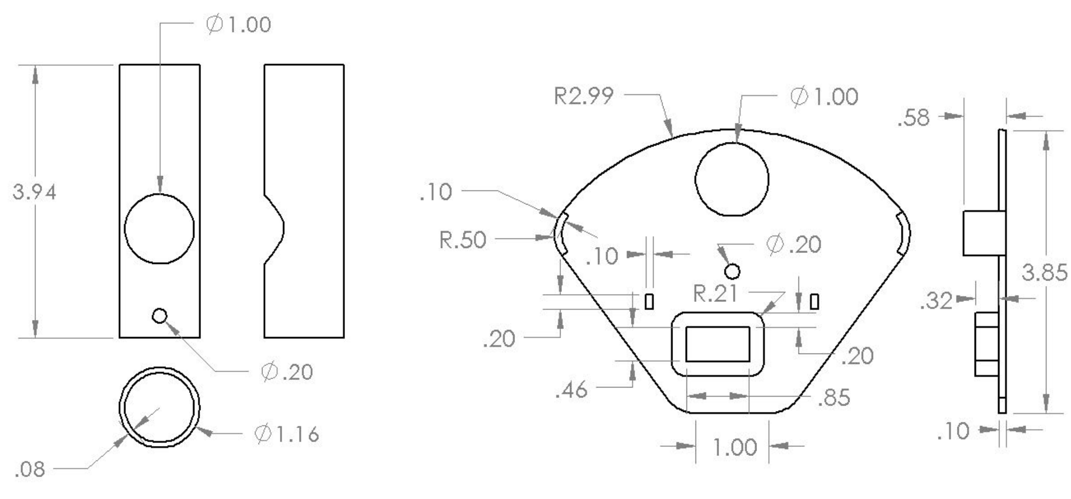
Appendices

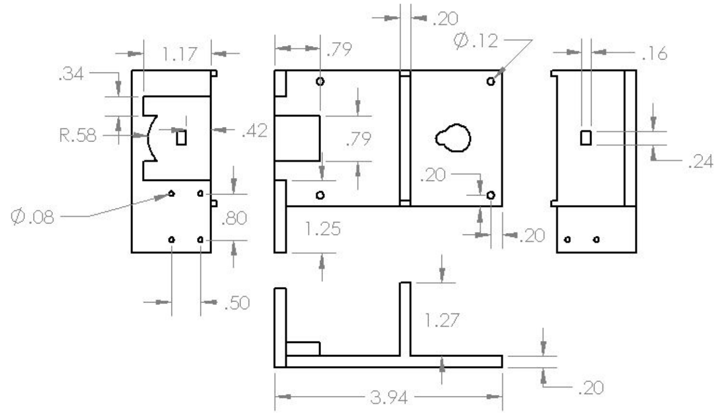
A. CAD Drawings

Wheels

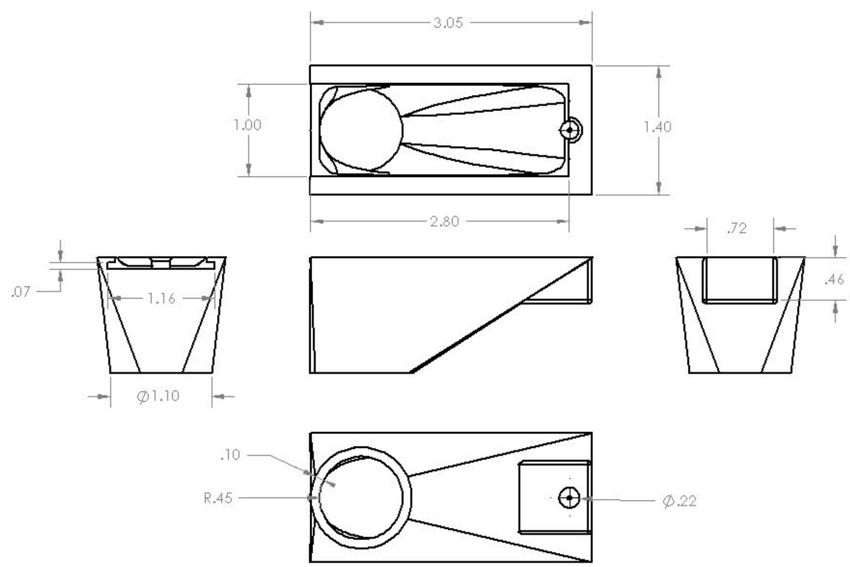


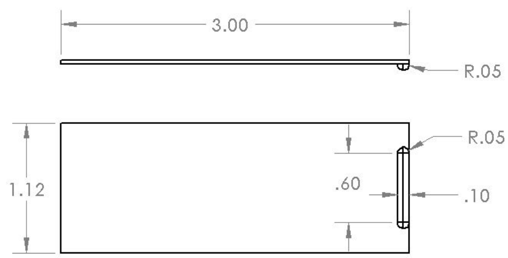
Launcher



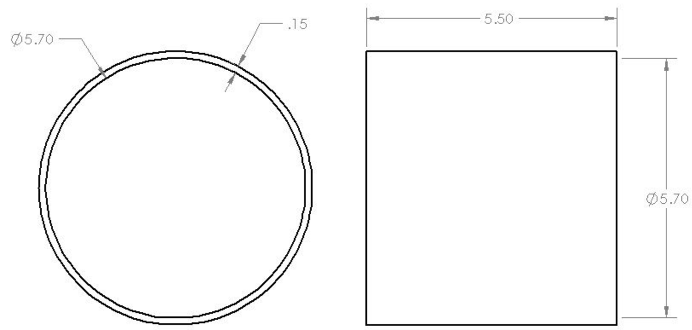


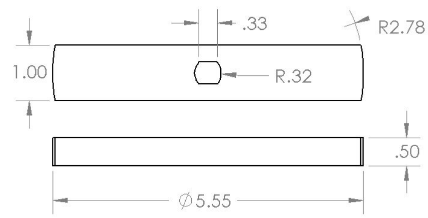
Treat Holder/Loader

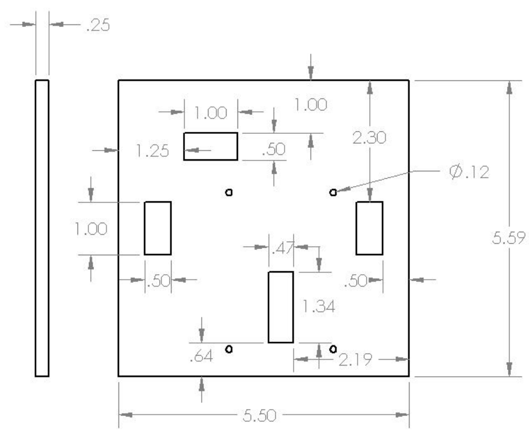




Cylindrical Shell

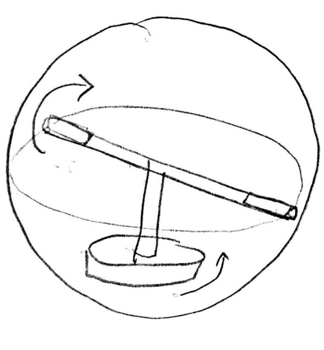


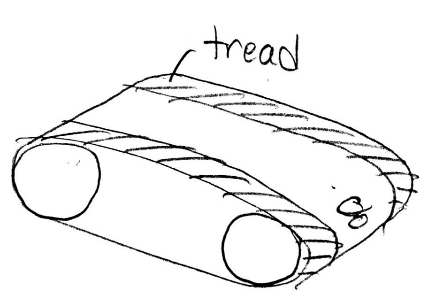


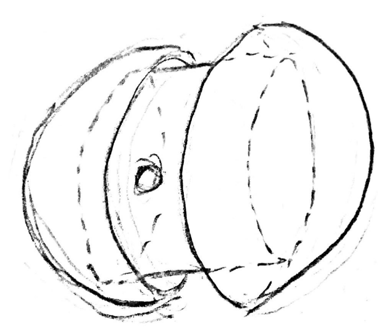


B. Concept Sketches

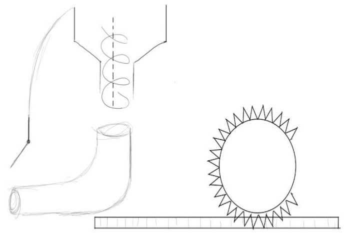
Mobility



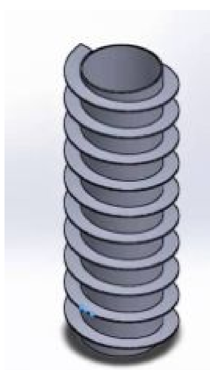


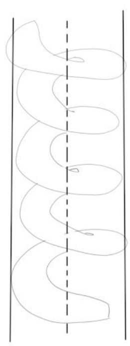


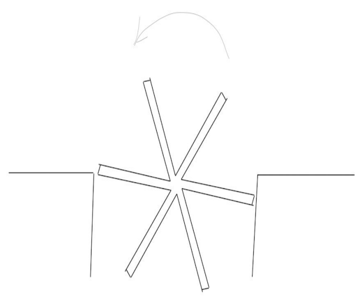
Launcher

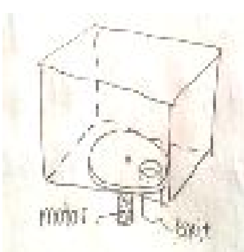


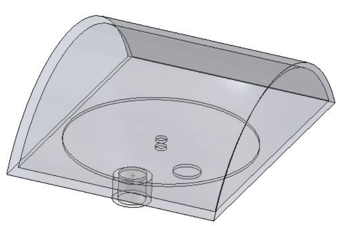
Loader











C. Code

Link to our public GitHub repo where all codes running on our microcontroller (Raspberry Pi) are stored: https://github.com/lennyzhang/PupBuddy

Important scripts and related functionalities:

1. Controlling code for DC motors (drive system) and servo motors (load+launch)
2. Web Server Structure

D. Link to Off-the-Shelf Components

Motor (mobility)	https://www.pololu.com/product/995/specs
Raspberry Pi	https://www.amazon.com/Raspberry-Pi-RASPBERRYPI3-MODB-1GB-Model-Motherboard/dp/B01CD5VC92/ref=sr_1_3?s=pc&ie=UTF8&qid=1519420890&sr=1-3&keywords=raspberry+pi+3&dpID=51Vt9f26ryL&preST=_SX300_QL70_&dpSrc=srch
Motor Shield	https://www.adafruit.com/product/2348?gclid=EAIaIQobChMI4r2zhIC92QIVjFmGCh3JBgXmEAQYBiABEgI4UPD_BwE
Motor (launcher, laoder)	https://www.pololu.com/product/2817
Spring	https://www.amazon.com/Forney-72599-Compression-Assortment-100-Pieces/dp/B003XESMS2/ref=pd_sim_60_3?_encoding=UTF8&pd_rd_i=B003XESMS2&pd_rd_r=3RD1JX3JE8B00JAY36A9&pd_rd_w=7WGM9&pd_rd_wg=1eG7c&psc=1&refRID=3RD1JX3JE8B00JAY36A9&dpID=51LQ7aPtFHL&preST=_SX300_QL70_&dpSrc=detail
Battery	https://www.amazon.com/Floureon-Dean-Style-Connector-Helicopter-5-35x1-69x0-83/dp/B00L2V09VQ/ref=sr_1_7?s=toys-and-games&ie=UTF8&qid=1519419030&sr=1-7&keywords=lipo%20battery
Raspberry Pi Camera	https://www.amazon.com/dp/B01ICLLOZ8/ref=sspa_dk_detail_2?psc=1
PC Protective Case with 2x Heatsinks	https://www.amazon.com/Raspberry-Model-Protective-Heatsinks-Clear/dp/B01CDVSBPO/ref=sr_1_8?s=electronics&ie=UTF8&qid=1520307845&sr=1-8&keywords=raspberry%2Bpi%2B3b%2Bcase&th=1
SD Card Reader	https://www.amazon.com/gp/product/B01KFXS83W/ref=ox_sc_act_title_1?smid=A1SFQ4T7XMP3XM&psc=1
SanDisk Ultra 32GB microSDHC UHS-I Card with Adapter	https://www.amazon.com/gp/product/B010Q57T02/ref=ox_sc_act_title_2?smid=ATVPDKIKX0DER&psc=1
Dean Style T to XT60 Adapter (battery connector)	https://www.amazon.com/Readytosky-Connector-Adapter-Battery-Connectors/dp/B01FZZNPWY/ref=sr_1_3?s=toys-and-games&ie=UTF8&qid=1521668726&sr=1-3&keywords=dean%27s+t+connector+adapter
XT60 to USB Adapter (battery connector)	https://www.amazon.com/Shaluoman-Battery-Converter-Charger-Adapter/dp/B012C6R8Z4/ref=cts_ap_1_vtp#HLCXComparisonWidget_feature_div

Voltage regulator	https://www.digikey.com/products/en?keywords=296-44888-ND
Acrylic hemisphere	https://www.amazon.com/Acrylic-Dome-Plastic-Hemisphere-Pre-Drilled/dp/B01NHEUGXW/ref=sr_1_1?s=pet-supplies&ie=UTF8&qid=1521909371&sr=1-1&keywords=7+inch+acrylic+hemisphere
Body shell	https://www.amazon.com/Acrylic-Decorative-Centerpiece-Royal-Imports/dp/B016V20IJS/ref=sr_1_1?s=home-garden&ie=UTF8&qid=1521748088&sr=1-1&keywords=acrylic+cylinder+6+inch
motor shaft hub	https://www.pololu.com/product/2682
Dodge Ball	https://www.amazon.com/Champion-Sports-7-Inch-Playground-Ball/dp/B002LATNT6/ref=sr_1_fkmr0_2?ie=UTF8&qid=1523885772&sr=8-2-fkmr0&keywords=7%22+rubber+dodgeball