WHEELCHAIR-MOUNTED DOG TREAT DISPENSER

DESIGN DOCUMENT

v1.0

Team Members

Zainab Abdullahi
Adam Dost (PM)
Gage Moore
Jachan Shrestha
Robby Wignall

Faculty Supervisor

Dr. Nathalia Peixoto

Client

Service Dogs of Virginia
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of Contents</td>
<td>2</td>
</tr>
<tr>
<td>Problem Statement</td>
<td>4</td>
</tr>
<tr>
<td>Previous Designs and Patents</td>
<td>5</td>
</tr>
<tr>
<td><strong>Treat Dispenser Project Requirements</strong></td>
<td>10</td>
</tr>
<tr>
<td>Mission Requirement:</td>
<td>10</td>
</tr>
<tr>
<td>Functional Requirements:</td>
<td>10</td>
</tr>
<tr>
<td>Operational Requirements:</td>
<td>10</td>
</tr>
<tr>
<td>Input Requirements:</td>
<td>10</td>
</tr>
<tr>
<td>Output Requirements:</td>
<td>10</td>
</tr>
<tr>
<td>Technology:</td>
<td>11</td>
</tr>
<tr>
<td><strong>System Architecture</strong></td>
<td>11</td>
</tr>
<tr>
<td>Dog Treat Storage Area</td>
<td>12</td>
</tr>
<tr>
<td>Circuitry</td>
<td>12</td>
</tr>
<tr>
<td>Dog Treat Staging Area</td>
<td>13</td>
</tr>
<tr>
<td>Dog Treat Dispensing Area</td>
<td>13</td>
</tr>
<tr>
<td><strong>Background Knowledge/Phenomenology</strong></td>
<td>13</td>
</tr>
<tr>
<td>Servo Motor</td>
<td>13</td>
</tr>
<tr>
<td>Electrical System</td>
<td>14</td>
</tr>
<tr>
<td>Mechanical System</td>
<td>14</td>
</tr>
<tr>
<td>555 Timer</td>
<td>15</td>
</tr>
<tr>
<td>Astable Operation</td>
<td>16</td>
</tr>
<tr>
<td>Monostable Operation</td>
<td>17</td>
</tr>
<tr>
<td>Dispensing Analysis</td>
<td>18</td>
</tr>
<tr>
<td>Servo Motor Timing Analysis</td>
<td>18</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>19</td>
</tr>
<tr>
<td>Power Analysis</td>
<td>20</td>
</tr>
<tr>
<td><strong>Detailed Design</strong></td>
<td>24</td>
</tr>
<tr>
<td>Overall Design</td>
<td>26</td>
</tr>
<tr>
<td>Treat Staging and Dispensing Mechanism</td>
<td>26</td>
</tr>
<tr>
<td>Dog Treat Storage Compartment</td>
<td>29</td>
</tr>
<tr>
<td>Circuitry and Activation Mechanism</td>
<td>30</td>
</tr>
<tr>
<td>Dispensing Tube</td>
<td>31</td>
</tr>
<tr>
<td>Functional Block Diagrams</td>
<td>32</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>----</td>
</tr>
<tr>
<td>Level 0:</td>
<td>32</td>
</tr>
<tr>
<td>Level 1:</td>
<td>33</td>
</tr>
<tr>
<td>Level 2:</td>
<td>34</td>
</tr>
<tr>
<td>Software Flowchart</td>
<td>38</td>
</tr>
<tr>
<td>Software Explanation</td>
<td>39</td>
</tr>
<tr>
<td>Material and Components</td>
<td>40</td>
</tr>
<tr>
<td>Prototyping Progress Report</td>
<td>44</td>
</tr>
<tr>
<td>Schematic &amp; Simulation Result</td>
<td>44</td>
</tr>
<tr>
<td>PCB Board using SMD Components - 3D View</td>
<td>45</td>
</tr>
<tr>
<td>Breadboard Circuit Testing</td>
<td>45</td>
</tr>
<tr>
<td>Cardboard Prototyping</td>
<td>46</td>
</tr>
<tr>
<td>Raspberry Pi Prototyping</td>
<td>47</td>
</tr>
<tr>
<td>Options to be Considered (Not Implemented)</td>
<td>50</td>
</tr>
<tr>
<td>Solenoid Control Module</td>
<td>51</td>
</tr>
<tr>
<td>Testing Plan for ECE 493</td>
<td>52</td>
</tr>
<tr>
<td>Functional Requirements:</td>
<td>52</td>
</tr>
<tr>
<td>Preliminary Project Plan</td>
<td>53</td>
</tr>
<tr>
<td>Gantt Chart for ECE-493</td>
<td>54</td>
</tr>
<tr>
<td>References</td>
<td>55</td>
</tr>
</tbody>
</table>
Problem Statement

The wheelchair is one of the most commonly used assistive devices for increasing and enhancing personal mobility, which is a necessity for being an independent and productive member of society. There are many conditions and afflictions which may result in the need of a wheelchair such as multiple sclerosis, cerebral palsy and muscular dystrophies. People with who are wheel-chair bound accompanied by service dogs account for .9 percent of the U.S. population[1]. These highly trained canines can provide independence to their owners and significantly enhance their quality of life. These dogs are capable of completing a wide range of tasks from opening doors, retrieving dropped items and pushing their partners up ramps. These dogs are an integral part of these individuals lives and must go through extensive training to earn this role[2]. In order to achieve maximum performance from these dogs, frequent rewards in the form of verbal affirmations and treats are required. Rewarding the dog is essential. It not only increases the emotional relationship between the dog and the owner but most importantly, provides the dog with much needed positive reinforcement.

This project was undertaken at the request of The Service Dogs of Virginia, a non-profit organization that raises, trains, and places dogs to assist people with disabilities due to the failure of several designs that have attempted to ameliorate the burden to create an effective standard of assistance. The main challenge users who have limited or no hand mobility at all is giving the dog a treat. There have been several great ideas and attempts to create an adaptive treat dispenser that could be integrated seamlessly to a wheelchair, but there are none that are commercially available as of today. Despite the numerous attempts made to make this merchandise commercially accessible to the public it has failed repeatedly due to design limitations such as; the dispenser being too large to be seamlessly embed into a wheelchair as well as there being malfunctions with getting the dispenser to release the desired number of treats appropriately. While there have been few designs that have overcome these setbacks, they too have proven to be futile as a result of the dispenser bowl being too small, and fiscal challenges due to the expenses of printing and dipping into a food safe coating.
Previous Designs and Patents

Previous attempts at this project have been done over the course of several semesters. To validate and ensure the design would succeed where other projects failed the design group evaluated each previous design and took note of each fault’s that were found so that the current design iteration would solve any and all problems that the previous designs failed on so that this design would be successful. Examining the current design shown in Figure 1 below carried issues with not storing the treats in a safe food storage container and was prone to jam quite easily upon use and no way of actually unjamming the device without human manual intervention. One of the key early design evaluations done was to ensure that the material that is used for food storage is not only safe material but won’t leave a pungent smell and can easily be washed when needed. A materials report was created to address this specific problem to ensure that the current design won’t fall victim to this.

Figure 1: Previous Design 1

Figure 2: Previous Design 2
The second design shown in Figure 2 above was praised as being one that had easily procured parts that could be sourced locally in a single store. Procurement and accessible parts is very important in this design as cost is a major factor and per our design cost of parts was evaluated to ensure that an operator could replace and repair broken parts in the event the device. Another key design flaw in this was managing the output of the treats. During Dr. Neuber’s briefing of what Service Dogs of Virginia does for training which includes varying the number of treats that the dog will receive depending on the task they complete or what the owner believes that they should be rewarded.

A third previous design that was evaluated had the most success of the three is shown in Figure 3 above. Both its placement on the wheelchair and the arduino storage area were successful and the storage device had no issues. One of the key issues the group had ran into was coating the storage for treats and found that the cost of it would have increased the entire cost of the device’s creation beyond what was acceptable by the design team. Another key issue was the size of the storage device was far too small for what it was supposed to do in live operation.
During our storage device measurement phase this design storage device flaw will play key in helping understanding what this iteration of the device will need for storage size. Currently, there is a need for an improved dog treat dispenser. The available prior art is limited by several factors such as being bulky, cumbersome, mechanical, non-food safe, and not adaptable to a wheelchair. Many handicapped people suffering from advanced disabilities do not have the dexterity as people without disabilities do. This leads to a need for a new design in order to meet their needs. In researching and examining existing patents, elements that were common with our design and elements that could not be used in our design were found. To meet the ADA (Americans with Disabilities Act) accessibility standards, the width of a door must be between 32 inches minimum to 48 inches maximum [3]. Many of the patents were not wheelchair adaptable and could not be mounted onto the side or even the back of a wheel-chair. Our goal was to ensure that the dispenser would not interfere with the user's day to day activities or limit them in any way.

![Figure 4: Patent One](image1)

![Figure 5: Patent Two](image2)

The patents illustrated in Figures 4 and 5 were too large and bulky, in addition to not being able to be mounted onto a wheelchair [4],[5]. Another crucial requirement was for the material used to be food safe and eco-friendly. Figure 6 below shows a patent that could be mounted onto a wheelchair, but was made using 3D printed parts which were not food safe [6]. Some elements that shall be used in our design are illustrated in Figures 7, 8 and 9 below.
These patents used a button for activation and a rotary mechanism which shall be incorporated into our design. The need for a generic switch to activate the dispensing of a treat is clearly needed for people with advanced disabilities as they will be able to use a switch, button, or other form of activator that they already utilize in everyday life to complete tasks.
In addition to the proposed design having a generic switch input, it will also be electronically controlled to remove the need for a mechanical input to dispense a treat. This will allow people with advanced disabilities to train their dog in order to comply with the testing requirements to keep their dog “in service.”

Upon reading the available prior art that has been registered, there is no indication that the proposed automatic treat dispenser design discussed in this document will infringe on any of the prior art. While there exist a few key elements such as a rotational food compartment, a tray portion, and a button, none of the elements in combination with our design would infringe on existing patents [4-7]. As all the designs lack the use of a circuit and/or microcontroller to activate the dispensing of a treat coupled with an adaptable size food storage area that is mountable to a wheelchair. Thus, based on our review there exists a novelty within our design, and it is okay to proceed with implementing it as described herein.
Treat Dispenser Project Requirements

Mission Requirement:

The device shall assist people with wheelchair-binding disabilities by providing them a method to reward their service dogs.

Functional Requirements:

1. The device will hold up to at least one cup of treats at a time in order to allow for treats to be provided to the service animal throughout at least one full day.
2. The device shall use timers to control treat dispensing mechanisms
3. The device shall utilize a mono-jack input so the user may use their own control mechanism. If time allows, this project will investigate an accelerometer-based control input.

Operational Requirements:

1. The device will dispense 1-5 treats, depending on initial setup, with a single activation of the user’s input device.
2. The device will be integrated seamlessly to a wheelchair without being too bulky.
3. The materials used will be food compatible.
4. The device will be easy to take apart and easy to clean.

Input Requirements:

1. The device will accept input from the operator. Though the physical input device may vary, the interface with the device will stay consistent (mono-jack).
2. The device will accept dog treats into a storage container integral to the device.

Output Requirements:

1. The device will output dog treats through a pipe system leading to a food tray.
2. The device will output a “click” sound to notify the service dog a treat has been dispensed.
3. The device will provide notification if the treat storage compartment is empty and no treats can be dispensed.
**Technology:**

1. Shall use some Surface Mounted Technology (SMT) 555 Timers or a Microcontroller.
2. Shall use some form of food-safe pipe/tube to move food from storage to dispensing tray.
3. Shall use some form of coating/hardening on the circuit to ensure durability.
4. Shall use food-safe materials for all surfaces that come in contact with dog treats.
5. Shall use rechargeable batteries to keep the economic impact minimal.
6. Shall use a servo motor or solenoid to be activated by the controlling circuit.

**System Architecture**

The System Architecture for the automatic treat dispenser is shown in Figure 10 above. The overarching blocks such as “Dog Treat Storage Area,” “Circuitry,” “Dog Treat Staging Area,” and “Dog Treat Dispensing Area” are the four main components utilized in this design. This shows how all components are interconnected and related at a high level to ensure that all required module functionality is achieved. The dashed lines show where the four modules relate
to each other. For example, the Circuitry is connected to the Dog Treat Staging Area by the following. When the 555 timer circuit is activated, the treat staging area is implemented, which in turn connects to the rotating plate portion of the Dog Treat Staging Area. Once prototyping begins, this diagram will be used as a reference so no piece is missing. This diagram will be modified throughout the prototyping and testing portion as new and different methods may be implemented to achieve the best results.

**Dog Treat Storage Area**

As shown in the figure above, the dog treat storage area will hold at least a cup full of dog treats at a time. This will ensure that the service dog will never be without a treat whenever the dog completes a correct task and shall be rewarded. The storage area will also be funnel shaped and made out of stainless steel. This will not only help with jamming problems (along with the vibrating motor), but will also help overcome past design flaws such as using non food safe materials.

**Circuitry**

At a high level, the circuitry will either utilize a 555 timer circuit or a microcontroller to be used as the control system for the treat dispenser. The type of microcontroller that will be used is the Raspberry Pi Zero W, and a 555 timer circuit utilizing Surface Mount Technology (SMT) will be used for the 555 timer based design in order to keep the printed circuit board (PCB) as small as possible. This will help to keep the treat dispenser’s footprint small.

The circuitry comprises the entirety of the controller portion of the design, but will further need to be comprised with a mono-jack input for switch connection. This will allow for any switch with a mono-jack to be plugged into the controller and used in conjunction with the treat dispenser. Since disabilities can vary from only being able to move a finger to only being able to move your head, this simple means for device input connection is necessary so the end user may utilize their custom switches. The circuitry component controls all of the moving parts in the dispenser design, and is the heart of the project. Not only will the circuitry control the spinning of the rotational disk, but it will also control the vibrating motor to help dislodge treats, activate the “click” sound module, and provide any required feedback to the user.
**Dog Treat Staging Area**

Essentially, the dog treat staging area is the part of the design where a treat will simply wait to be delivered to the service dog. It works in conjunction with the rotational mechanism described later to hold the treats in their respective positions so that, when the rotational disk is spun, the treat will fall into a dispensing hole which will then drop to the service dog.

**Dog Treat Dispensing Area**

The dog treat dispensing area is a simple but important piece of the design. The dispensing area comprises of a long flexible tube for easy customizability for the end user. This would potentially allow the whole treat dispenser to be mounted on the back of the wheelchair, out of the way of obstacles, and still be able to provide a treat to the service dog at the side of the wheelchair, which should be their ready position and keep the dog from having to walk around the wheelchair. Furthermore, the dog treat dispensing area is comprised with a food dish or bowl in order for the dog to receive a treat without it scattering across the floor. As mentioned above, service dogs are used to consistency when in training, so delivering the treats to the side of the wheelchair will help maintain their training. These four main modules are described more in depth in the Detailed Design section of the report.

---

**Background Knowledge/Phenomenology**

**Servo Motor**

A servo motor utilizes a brushless DC motor technology, which helps keep the motor’s efficiency high (no friction and lost energy due to brushes). The efficiency is about 10% higher than for a brushed motor[8]. There are magnets connected to the rotor and electromagnets connected to the stator (the stationary part of the motor). Every time the rotor spins 180 degrees, the polarity of the electromagnets on the stator needs to be flipped (or turned off in some cases) to allow the rotor to keep spinning. Otherwise, the magnets would constantly negate each other
and provide a total torque of zero. The system model as shown in Figure 11 below will be used to analyze the servo motor [9].

![System Model for Servo Motor](image_url)

**Figure 11: System Model for Servo Motor**

When analyzing servo motors there are two important time constants, the electrical time constant and the mechanical time constant. The time constants are usually listed in the data sheets, but are for the motor alone with no load connected to the shaft of the motor. Thus, it is important to know the actual values of the time constants under the actual load conditions. This will need to be determined during prototyping. Once the treat dispensing module is built, tests can be run to measure the torque required for consistent results.

**Electrical System**

\[ e_i = i_a R_a + K_e V_m \]  \hspace{1cm} (Voltage Equation)

The steady state equation for the electric part of the motor is derived using Kirchhoff’s voltage law. Where \( e_i \) is the input voltage, which is equal to the sum of the voltage drop across the resistor \( R_a \) and inductor (resistance of zero), \( L_a \), as well as the motor when activated using the given electrical time constant, \( K_e \), and angular motor velocity, \( V_m \).

**Mechanical System**

\[ T = \text{Torque} = i_a K_T = J \alpha \]  \hspace{1cm} (Torque Equation)

Torque is what causes objects to obtain angular acceleration. It is a force that causes an object (the circular plate) to rotate about an axis. The torque, \( T \), of the motor is equal to the torque constant, \( K_T \), times the current, \( i_a \). This is also equal to the total inertia of the motor, \( J \), multiplied by the acceleration of the arm, \( \alpha \). Once the required torque to activate the treat dispenser module is found, the servo motor needs to be utilized at a current that meets or, preferably, exceeds the measured required torque.
A 555 timer integrated circuit (IC) is made up of two operational amplifiers (op-amps), a flip-flop, output driver, transistor, and series of resistors [10]. The timer gets its name from the series of three 5 kΩ resistors between ground at Pin 1 and the positive voltage supply at Pin 8, producing a voltage divider network so that ⅔ of the supply voltage is supplied to the inverting input of op-amp two and ⅓ is supplied to the non-inverting input of op-amp one. The trigger at Pin 2 is connected to the inverting input of op-amp one so that, when the trigger is higher than ⅓ of the supply voltage, the op-amp’s output activates the set terminal on the flip-flop. The threshold at Pin 6 is used to override the ⅔ input voltage and reset the flip-flop. The control voltage at Pin 5 can be used to override the RC timing network, but will be ignored in this project and connected to ground with a capacitor to eliminate noise. The flip-flop’s output is connected to a driver to amplify the signal some, and then is connected to the output at Pin 3. The reset at Pin 4 is connected directly to the flip-flop’s reset. Lastly, there is the discharge at Pin 7, which is used to discharge the timing capacitor to ground when the output is active. There are three main modes of operation for a 555 timer circuit: astable, monostable, and bistable (Schmitt Trigger).
In astable mode, there is no “stable” state; the output is an always oscillating square wave with period and duty cycle determined by the values of resistors and capacitors in the circuit. In monostable mode, the stable mode is when the output is low. When the 555 IC’s connection to the trigger pin goes low, the output goes high for a certain amount of time as determined by the impedance of the circuit. After that timeframe, the output goes low again until a signal is seen at the trigger again. In bistable mode, also known as a Schmitt Trigger, the 555 timer behaves similarly as it does in monostable mode, but the output will remain high until the input to the Reset pin goes low. Thus, the timing aspect of the circuit is essentially bypassed, and two inputs would need to be utilized. Based on the requirements for the automatic treat dispenser, the astable and monostable modes of operation will be the most useful.

**Astable Operation**

![Figure 13: 555 Timer Circuit for Astable Operation [11]](image)

As mentioned previously, this circuit’s output will generate a square wave with controllable frequency and duty cycle [11]. This mode’s output is also known as pulse width modulation (PWM) and may be useful for activating a motor intermittently and dislodging treats by vibrating the treat storage compartment. Here, the period $T$ is given by the sum of the time the output is high and the time the output is low ($T = T_h + T_l$), where:
\[ T_h = 0.7 \times (R1 + R2) \times C1 \]
\[ T_i = 0.7 \times R2 \times C1 \]

Combining these two equations, we can find the period \( T \) directly using the following:

\[ T = 0.7 \times (R1 + 2 \times R2) \times C1 \]

To vibrate a motor well, a small duty cycle is desired. A duty cycle less than or equal to 50% would provide a “sputtering” type effect that can also help to save power.

\[ Duty \ Cycle = 100 \times \frac{T_h}{T_h + T_i} \]

**Monostable Operation**

![555 Timer Circuit for Monostable Operation](image)

*Figure 14: 555 Timer Circuit for Monostable Operation [11]*

Under these conditions, the 555 timer’s output will remain high for an amount of time based on the external RC system after triggered, and return to a low state until triggered again [11]. It also gives the option of an “override,” which will keep the output high while a trigger is applied, providing an option for a “long push.” Bringing the reset pin low can also cut the output signal short if needed, which could be required if the treat compartment is empty. This signal can
cancel the treat dispenser and also be used to notify the user to refill the treat dispenser.

The amount of time the output goes high after a trigger can be found by:

$$T = 1.1 \times R1 \times C1$$

To design the circuit, a capacitor is selected first as resistors are much easier to find in different sizes that can fit the required $T$. This output $T$ should be equal to the amount of time it takes to turn the motor around one time. In order to determine the required $T$, an analysis of the dispensing module must be made.

**Dispensing Analysis**

Since the design utilizes a circular dispensing module divided into three equal compartments, the goal is to turn the module $\frac{1}{3}$ of the way, or $120^\circ$, around so a new compartment lines up with the dispensing hole each activation. Using the equation for circumference $C$ of a circle, $C = 2\pi r$, where $r$ is the radius of the dispensing module, the size of the motor’s gear can be determined using the following logic:

To dispensing module must turn $\frac{1}{3}$ of the way around to dispense the treats, so the distance needed to turn is given by:

$$\frac{1}{3} (2\pi r) = \frac{1}{3} \pi r$$

Therefore, the gear attached to the servo motor that will turn the dispensing module can be lined up with the outside perimeter and have the circumference given by:

$$Circumference \ of \ Motor \ Gear = C_{motor} = \frac{1}{3} \pi r$$

Since gears are being used as shown in Figure 23 of the Detailed Design portion below, the outer diameter will be $C_{motor}$ (to the gear tips) while the inner diameter should be large enough for the gear to grip the teeth of the dispensing module turn the module accordingly. Now that the size of the gears are determined, timing requirements can also be established.

**Servo Motor Timing Analysis**

To figure out the timing needed to keep the Servomotor active long enough to turn our plate $120^\circ$ as described above, the RPM of the servo motor must be taken into account. The
specified RPM for the Servo motor can be used to figure out how long it will take to rotate around one time as:

\[
Rotation\_Time = \frac{60}{RPM}
\]

This \textit{Rotation\_Time} will then be used to design the 555-Timer circuit as well as the code for the microcontroller. The output from both of these designs should stay high for at least one \textit{Rotation\_Time} after an activation. Of course, for different functionality, this time may need to be extended (i.e., vibrating to dislodge treats). Thus, from above, the required \textit{T} for monostable operation will be given by \textit{T} = \textit{Rotation\_Time}. This same time \textit{T} can also be used for the astable operation.

In the case where it is determined that the servo motor provides inconsistent results or an overly complicated circuit, a solenoid with arm will be used in its place. As described later in the prototyping section, this lateral motion can be translated into rotational motion so the rest of the design can remain the same. When a solenoid is used, the time \textit{T} will be given by the amount of time to fully activate the solenoid, extending the arm fully. This will be called \textit{Solenoid\_Time} and \textit{T} > \textit{Solenoid\_Time}.

\textbf{Raspberry Pi}

Raspberry Pis are an inexpensive “off the shelf” solution that provides diverse functionality in a small package. The small form factor (66.0mm x 30.5mm x 5.0mm) will allow this part to be integrated to the treat dispenser without causing any obtrusions[12]. Raspberry Pis offer several configuration options due to the power of the microprocessor. This device can be utilized with a screen and user interface (Human Machine Interface - HMI), or blindly using button commands to make adjustments to programming. No matter how the RaspberryPi is used, the program can be saved and backed up in a separate location so that it can be copied to replacements in the event of damage or a failure. Using the tool PiBakery, the Raspberry Pi can be pre-configured with the design’s required code already ready to operate on each startup. This can drastically reduce any repair time and allow for more tech-savvy users to customize their experience if wanted.
Power Analysis

555 Timer: In order to begin analyzing the power requirements of a 555 Timer Circuit, a simulation or the information from its datasheet can be used. For this project, a basic simulation of the circuit in Astable operation was performed as shown in Figure 15 below. By measuring the current leaving the positive terminal of the power supply, the total current for the circuit can be determined (Shown in Figure 16). Since the total current and supply voltage is known, the total power consumption can then be calculated as well. The output PWM signal shown in Figure 17 coincides with the amount of time needed to spin the servo motor, so this circuit could potentially be used, making the simulated numbers a worthwhile analysis tool.

![Figure 15: 555 Timer Circuit in Astable Mode for Simulation](image)
Using the total current waveform from these initial simulations of the 555 Timer circuit in astable conditions, the circuit should draw around 0.5 mW without any load. This is based off
the 9V power supply and 50 μA peak shown in the waveform. Next, the motor will need to be considered. The motor the design will leverage is an FS90R, which, based on the datasheet, will draw about 5W per activation [13]. This is determined by the maximum stall current of 650mA at 6V which will be at around 4W and adding in a slight safety factor. To determine if the treat dispensers have treats or not, a sensor, like a photosensor, will need to be used. Also, if an accelerometer activation switch is created, it will need some power to provide activations to the treat dispenser. An estimate of 5 watts will be added by each sensor. Thus, the final estimation for the total power consumption will be around 20-30 W per activation.

**Microcontroller:** The Raspberry Pi Zero W requires a 1.2A/5V input which gives a total power consumption of 6W max [14]. The team will also leverage additional LEDs and sensors to provide feedback to the user of the device, each at around 5W per sensor. Thus, the total estimated power consumption using this design would be between 26-36 W. This is another number that will need to be verified during prototyping and testing.

**Battery Requirements:** From the power analysis above, it can be seen that either design will require around 2 amps per activation at 12 Vdc. In order to determine the number and size of batteries needed, the device usage needs to be taken into account. The design requires the ability to store a minimum of one cup of dog treats at any time, which is about 15-30 treats depending on the brand used. Thus, with 2-3 treats being dispensed on average, the device will be used between 8 and 15 times before a refill is required. When a refill is required, the batteries can also be changed out for a freshly charged set, though going 2-3 days between charges may be favorable. Therefore, the worst case scenario will be around 45 activations of the device between battery charging, and each activation should have the circuit draw power for less than 5 seconds (in the case of the 555 timer design since it is comprised of passive components; microcontroller design may have higher power needs).

Therefore, the batteries will need to provide power to the device for around 240 seconds, or 4 minutes, between charges. 4 minutes is the same as 0.0667 hours. A double A (AA) battery has about a 2 Ah capacity at 1.2 Vdc. If ten AA batteries are used in series, a voltage of 12 Vdc can be achieved, which should power the device for about:
Where the \textit{charges} were determined to be the worst case of 3 days above. Thus, these ten AA batteries would be able to run the device for an estimated 45 days before charging is needed.

The Raspberry Pi Zero W can be powered by USB cable, and there are many USB battery packs available that would be able to power the microcontroller for at least a day at a time. The battery requirements for the microcontroller design would include the constant power to the Raspberry Pi at about 6 W Maximum for the entire day. This can be estimated to be about 2 W on average throughout an entire day (including activations) as the idle power will be much less than when trying to activate modules. At 12 Vdc, this comes out to be about 0.1666 Amps. Multiplying this by a 12 hour day shows that 2 Amp-hours of battery will be required. As determined above, this can be accomplished with ten AA batteries, or with a rechargeable battery pack that will have an even higher capacity. Therefore, no matter which design is utilized by the customer, they will be able to use the automatic dog treat dispenser all day long on a single charge. The general operation of either style design is shown in Figure 18 below.

\[
\frac{1}{0.0667} = 15 \text{ charges}
\]
As shown in the flowchart, every device operation should be achievable from a single input. There should be one feedback letting the user know whether or not treats are present in the storage or dispensing area, but everything else is directed towards processing and dispensing treats to a service dog. Once the treats are dispensed, the process starts over and waits for user input.

**Detailed Design**

Previously in the proposal document, there existed four different designs. This stemmed from two major design components such as a rotational dispenser mechanism, and a dual slot design dispenser mechanism as shown below as Figures 19 and 20, respectively.

*Figure 19: Design with Rotary Dispensing Mechanism*
As depicted above, Figure 19 shows a rotational dispensing disk that rotates upon activation of a button to dispense a treat. Figure 20 shows how the dual slot mechanism operates by actuating a first arm in order to drop a treat into the placeholder, and then closes to prevent other treats from dropping in. The second arm would then open in order to allow the treat to drop into a bowl below to allow the dog to receive a treat. These are the two main designs that we previously have chosen to base the dog treat dispenser off of.

However, based on a review of both designs, it has been decided to move forward with the rotational mechanism based dog treat dispenser. Not only will this design require half the servos to dispense a treat, but it will also allow for a more simplistic design that saves money and energy. The major difference in the two designs the team is moving forward with will be what actually controls the system.

The first design will utilize passive components on a printed circuit board (PCB) and 555 timer chips to control the system. The second design will utilize a RaspberryPi Zero W microcontroller in order to control the system.

Using a PCB will allow for an overall lower cost (in components and materials) and allow someone to simply “plug and play” with our design in regards to reproducing it. This “plug
and play” action will be suitable for someone with little to no engineering experience to build the automatic treat dispenser design so many people will have access to it. On the other hand, the microcontroller based design (Raspberry Pi Zero W) will require some minimal background in electronics when it comes to programming the system and making sure the General Purpose Input and Output (GPIO) pins are hooked up correctly to allow the dispenser to function properly. With the two major design controllers noted, the decomposition of the designs are as follows.

**Overall Design**

The automatic dog treat dispenser will implement a design that is robust while being sleek and lightweight. Since the treat dispenser will be installed on a wheelchair, it cannot interfere with the normal operations of the wheelchair, nor limit locations that the user can go in the wheelchair. It will be low profile and able to mount anywhere with adjustable tubes to accommodate a low or a high mounting of the device. The device will be battery powered to work autonomously from the rest of the wheelchair. It will not be responsible for draining the wheelchair’s power as there are more important functions of a wheelchair than a dog treat dispenser.

The dog treat dispenser will integrate seamlessly with the end user’s switches/buttons using a 3.5mm mono-jack as connection. This connection is simply needed to close a circuit that will trigger the dispenser when activated and remain open otherwise. The project will also explore the creation of a switch using accelerometers or some other method (perhaps using fiber and detecting vibrations due to movement). This portion is still to be discussed and can be further investigated once the main treat dispenser has been completed.

There are four main areas for the automatic treat dispenser: the dog treat storage compartment, the treat staging and dispensing area, the circuitry and mechanism to activate the treat dispenser, and the treat deployment tube.

**Treat Staging and Dispensing Mechanism**

The treat staging and dispensing area influences what the mechanism to activate the treat dispenser should be, as well as the dog treat storage compartment. The dog treat storage compartment feeds directly into the dog treat staging and dispensing area, and the circuit has to
work for whatever mechanism is established, so the design must start here. The style in which the design will be based off of is the gumball machine style (rotational mechanism).

The gumball machine style consists of a horizontal, circular plate that spins on an axis perpendicular to its center with cut outs to drop the treats through a hole when they are to be dispensed (Figure 21). This plate would have three to four openings (dependent on rotational disk) around the outside that are just large enough to hold two to three treats side-by-side[15].

This spinning plate would then sit on top of another plate that has only one hole in it the same shape as the openings in the spinning plate. This plate would be static and should be able to line up with one of the openings in the spinning plate at a time. There would then be one more plate above the spinning plate with an opening that is large enough to let treats filter into one or two of the openings on the spinning plate at a time. The opening would have to be offset from the opening of the bottom plate and towards the start of the rotation after the bottom opening.

*Figure 21: Dispensing Plates*
These three circular plates would be at the bottom of the dog treat storage compartment so that all the treats sit on top of them and can be filtered in as the spinning plate is activated. The spinning plate should turn 1/3 of a full spin at a time (due to the 3 evenly sized compartments) and have an opening line up with an opening in the bottom plate each turn.

Thus, the treats are loaded into the spinning plate one or two openings at a time and then dispensed one opening at a time for each activation. The number of openings will have to be tested to determine what works the best and provides the most consistent results.

The great fact to note about this style of dispensing mechanism is that the gumball machine style rotational disk is usually adjustable as shown below as Figure 22.

![Figure 22: Dispensing Mechanism](image)

The screws on the plate will allow for the user to adjust the spacing in order to let a treat (shown as a gumball above) drop into the space so the user can drop as many treats per activation as they wish.
In order for the disk to rotate, the normal gumball machine handle is normally activated by a user twisting the handle. As shown below as Figure 23, the handle will be replaced by a servo to allow for an electronic activation to spin the disk.

![Figure 23: Rotary Mechanism](image)

The servo will be attached to the handle portion of this spur gear in order to drive the larger rotational disk above to allow a treat to be dropped and be provided to the dog.

Although this design will allow for great control over the treats being dispensed, this design may be susceptible to jamming, and dog treats can sometimes get sticky in humid areas, so preventative measures must be taken. These measures can be taken within the dog treat storage compartment.

**Dog Treat Storage Compartment**

The storage compartment will have smooth, easy to clean surfaces, and be detachable for cleaning. The material used will need to be food safe. It should also be transparent or translucent enough for the user to see the amount of dog treats still available at any time. The bulk of the storage compartment will be of stainless steel which is a food safe material. The compartment
will be slim and wide to hold at least a cup of dog treats at a time and ensure the wheelchair’s profile is minimally impacted.

The storage compartment will have a funnel type bottom to feed the staging area of each dispensing mechanism with dog treats. The funnel will also have sharp enough slopes so the treats do not get stuck.

The dog treat storage compartment will also have an internal vibrator that activates when the automatic treat dispenser is activated. This is to help ensure that the treats feed into the staging area and jams are broken free. The vibrator can also be attached to the outside of the storage compartment (to avoid stopping treats from falling down within the compartment).

There will also be a hinged lid on top of the storage compartment with an easy to use latch/clasp for simple refills. The lid will have a seal to make sure the dog treats do not spill out the top of the compartment.

The dog treat storage compartment will be the largest part of the dispenser so it will also have a flat face with some mounting tabs to make it easy to mount anywhere on a wheelchair. The storage compartment will be tightly secured to the dispensing mechanism and the rest of the treat dispenser so that no other mounting points will be needed.

**Circuitry and Activation Mechanism**

The circuitry in the PCB based design style will require the use of a timer circuit. This timer circuit will need to activate a motor for long enough to spin the gumball style dispensing mechanism.

The circuitry both of the designs will require a motor to spin the plate/wheel far enough to dispense the treats. The motor speed will be used to time how long it takes for it to spin the plate/wheel to the correct position and the timer circuit will be built accordingly to keep the motor activated for that amount of time [16].

Another method would be to use limit switches that are bypassed when the device is activated and then activated right after so that the device stops when a limit is reached. These can be user adjustable for troubleshooting purposes [17].
The circuitry will be installed within a sealed container to protect the components from damage. The PCB will be designed so components can be easily replaced and will be mounted on rubber mounts to protect the circuit from vibration damage [18]. All connections will be solid enough for a long life under normal use.

The circuit will have a 3.5mm mono-jack input to activate the circuit (in place of the shown input switch) and will be battery operated for autonomous operation. The 3.5 mm mono-jack will be the connection point for the switch of user’s choice. The circuit itself will look at this portion as a simple switch that closes the circuit when the switch is activated. The lower right circuit depicts the activation circuit. When a button/switch is pressed, it will keep the circuit on for a specified period of time (via the time constant of the 1000uF capacitor or time constant of the Raspberry Pi zero w), and will turn off the circuit when the capacitor is discharged (or time constant is reached).

As previously mentioned, the circuit that controls the motor speed will use a 555-timer chip to create a Pulse Width Modulation signal to drive the motor [19], [20], [21], [22]. This will spin the motor/servo in a direction (e.g. clockwise or counterclockwise) in order to actuate the dispensing mechanism.

An accelerometer type switch will also be investigated as a method of activating the treat dispenser. This device will be wireless for easy use and help those individuals that lack any kind of dexterity. A simple nod could be enough to trigger the accelerometer and activate the treat dispenser. This activation switch will be implemented once the treat dispenser is functioning as needed.

The circuit will also include a small speaker that emits the dog training clicker sounded to pique the interest of the service dog.

Dispensing Tube

The Dispensing Tube would be the simplest part of the dog treat dispenser. It will consist of a tube that allows the dog treats to easily slide through, and a tray attached at the bottom that will catch the treats and provide a constant location for the service dog to go to for the treats. The tray will be constructed of stainless steel that will last with constant use by the service dog. It will be big enough for a fairly large service dog’s nose to fit in so that it accommodates as many
service dogs as possible. The tray will also have provisions for mounting it to a wheelchair, though these mounts will not be responsible for any of the device’s weight. It will simply help to hold the tray in place and ensure the treats are dispensed to the tray.

The dispensing tube will be tightly attached to the bottom of the dispensing mechanism and should be only slightly larger than the opening the treats will be dispensed through. The tube will be flexible so that the storage compartment and dispenser portion can be mounted in a variety of locations and still allow the use of the same tube.

### Functional Block Diagrams

#### Level 0:

![Figure 24: Level 0 Functional Block Diagram](image)

Shown above in Figure 24 is the Level 0 Functional Block Diagram. This is a high level representation of how the treat dispenser will operate via input(s), outputs, and the “black box” in between where all the design functions and dispenser modules operate. The input signal will be the activator for the whole control system. This will control each of the modules contained within the design: the Click Sound, Spin Dispenser Motor, Drop Treat, and Dislodge Treats modules. The outputs of the system are the treats being delivered to the service dog, the audible “click” sound that will entice the dog, and any other notifications needed to be provided to the user based on the status of the treat dispenser (e.g., empty treat storage area).
The Level 1 functional block diagram in Figure 25 above depicts how the modules within the “black box” from the Level 0 diagram are connected. The input signal will be routed to the click sound component, spin dispenser motor, and the dislodge treats module. This input signal triggers the treat dispenser and lets it know that a user would like to provide a treat to their service dog. In turn, this trigger will produce the “click” sound, spin the dispenser motor, and dislodge any stuck treats in the mechanism. The spin dispenser motor will then process the treat (move the treat over to the hole to drop a treat) where, finally, the drop treat module will simply provide the treat to the service dog. A more in-depth analysis on each of the components / modules is shown below in Figures 21 through 24, the Level 2 functional block diagrams.
Level 2:

Figure 26: Level 2 Functional Block Diagram for Drop Treat Module

Figure 26 above represents a cascaded system that will serve as the main function of the design. The input signal is now denoted as “Rotate” as the input signal will now make the drop treat module physically rotate the gumball style mechanism to provide a treat to the dog. A servo will act as the component that will create the rotational force to spin the rotating disk. Once the disk is rotated, the treat will then enter the “catch treat” portion so the service dog will be able to receive/obtain the treat. Once the device is in the “Catch Treat” mode of operation, the treat will be simply caught, and the dog will be able to take the treat from a receptacle, which completes the “Treat Delivered” portion of this module.

Figure 27: Level 2 Functional Block Diagram for Dislodge Treats Module
The diagram in Figure 27 above shows the dislodge treats function of the treat dispenser. The input signal will activate a time delay circuit that will allow the motor to spin for a specified amount of time. This time delay will primarily come from the use of a large capacitor so when the input signal is received, the circuit will allow the voltage to flow to an input of another timing circuit. This time delay signal is then routed to the motor control unit which will utilize a Pulse Width Modulation control signal (motor control signal) to spin the vibrating motor. This will be provided by either a 555 timer chip, or the raspberry pi depending on which design is in use. Next, this signal will simply drive the vibrating motor so that if there are any treats stuck in the system, they will be dislodged and returned to normal treat storage, ready to be dispensed to a service dog.

![Figure 28: Level 2 Functional Block Diagram for Click Sound Module](image)

The next block to be discussed will be the Click Sound module. This will utilize the input signal from the user’s switch (connected to mono-jack port) to control the 555 timer circuit or the RaspberryPi to produce a sound that is audible to the service dog. The Pulse Width Modulation control block comprises of the input signal to initiate a PWM signal that will generate an audible sound signal comparable to the clicker sound used in dog training. In the case of the RaspberryPi microcontroller, a saved waveform depicting the audible sound to be passed through to the speaker as a result of the input trigger. The PWM control block also consists of a reset signal so that further commands for dropping a treat will have a sound ready for the dog to hear indicating a treat has been dropped. This audio signal will then be passed to the speaker so that the dog can actually hear the audible sound produced from the PWM control block.
The spin dispenser motor will be the last module discussed. This module will receive an input signal from the user based on a button / switch actuation, which will then be routed to a time delay circuit. This time delay circuit will work in the same manner as the dislodge treats module will, although this time delay signal will then control how long the motor will actually be spun to rotate the disk. The motor control unit will then receive the output signal from the time delay circuit and set the time constant for the motor to spin. The motor control unit will also comprise a reset signal to ensure it is ready for further treat dispensing commands from the user. This will be done either through the 555 timer, or the raspberry pi as previously mentioned based on which design is in use. Furthermore, the motor will receive its input signal as a steady pulse for the duration it should be active, which will result in the motor spinning the rotational disk the 120° needed to drop a treat(s).

All of these modules interconnect to create a cohesive design. At each step of the process, the device is in different states, so a state diagram can also be utilized to explain how all the modules come together. This is shown in Figure 30 below. The initial state is to go from an unknown state to “Waiting” in the top left marked with 0. There is only one thing that can happen from this state, which happens when the dispenser is activated. The state of the device then moves to
position 1, “Spin Motor” where there are two possible outcomes. The motor either spins for a
given amount of time and comes back to this state until a treat can be dispensed. Once a treat is
dispensed, the device goes to state 2, “Dislodge Treats.” Here, the user can determine whether or
not treats were dispensed. If not, the dislodge treat button is pressed and the device goes back
through states 0, 1, and 2. Once the treat is dispensed correctly, the treat is dropped and the
device moves to state 3, “Dispensing,” which is the final state of the device. Here the treat is
either held and the device remains in state 3, or the treat is retrieved by the service dog, which
triggers the device to return to state 0, “Waiting.” This process repeats itself until the treat
storage compartment is empty and needs to be refilled, or the batteries need to be recharged.
After one of these breaks, the device reinitializes as shown by the arrow at the top left and goes
to state 0, “Waiting.”

![State Diagram of Automatic Dog Treat Dispenser](image)

*Figure 30: State Diagram of Automatic Dog Treat Dispenser*
Software Flowchart

Figure 31: Software Flowchart
Software Explanation

The software flowchart shown above in Figure 31 depicts how the RaspberryPi will operate. First, the RaspberryPi will go through a startup session. During this start up session, the device will set itself up by using RaspberryPi libraries for GPIO ports to designate the pins to their correct functionalities. Not only will the device be set up, but the interrupts will also be set. The interrupts will allow for the RaspberryPi to “wait” for an activation signal that will initiate dispensing a treat without the need for button polling.

Immediately following the startup session, the Raspberry Pi will monitor via interrupts the activation of the switch. If the Raspberry Pi detects a switch activation, this is when dispensing a treat will commence. If the switch has not been activated, it will simply stay in a continually monitoring period, wherein the Raspberry Pi will wait for the activation signal. If the switch has been activated via an interrupt, this will cause the dislodge treat function to start.

The dislodge treat function will activate the spinning of a vibrating motor located on the dog treat container. This will allow for any “stuck” treats to be shaken loose by the force of the vibrating motor. The motor will continue to spin for a time constant before going back into a waiting period to be activated again.

Next, the dog treat will actually begin to be dispensed via the “Spin Rotational Motor” function. This will simply spin the motor to allow a dog treat to be provided to the dog. The rotational motor will also spin for a designated time constant to allow a treat to be moved into place to align with the hole below. Upon finishing spinning for the time constant, the motor will stop spinning, and await for its next activation. Once the treat is dropped into place, the familiar click sound will then be played in order to alert the dog of a treat being dispensed. At this point, the treat will then be provided to the dog.

As a final check, in order to ensure that the dog is never without a treat, the system will check to see if any more treats need to be added. If the treat container has a low amount of treats, the system will send a signal to a red LED to alert the user that they are low on treats. If the container has a sufficient amount of treats left, then the Green LED will turn on. Once the function has completed, the Raspberry Pi will go back into its waiting period and wait for another user input to initiate a treat to be dispensed.
Material and Components

The main components of the automatic treat dispenser design as described so far are the storage compartment and treat staging/dispensing area, the circuitry, the motor, and the power source. To determine the best material and devices for the design, the following decision matrices shown in Tables 1 through 4 were completed. For each of the four categories listed, 3-5 possible contenders were researched and compared based on the listed criteria. The criteria was also ranked from 1 to the number of criteria based on the importance for the intended design.

![Table 1: Decision Matrix for Servo Motor](image)

The first component to be analyzed is the Servo Motor. This piece is integral to the design and the entire dispensing mechanism relies on it. Thus, the most important aspects are for the motor to do its job in being able to turn the required 360°, and to have enough torque to turn the dispensing mechanism and dispense treats to the service dogs. Next most important is the power requirement. In order to keep the cost and size of the device low, the power requirement needs to be low so as few/small batteries as possible are needed. Then, the speed of each motor is compared. This is less important since all the motors have speeds that will dispense the treats in around 1 second. Finally, the least important aspects are cost and weight since these are normally very cheap and light. Thus, these are non-issues for the design. After implementing these requirements, the FS90R Servo Motor was found to be optimal.

![Figure 32: FS90R Servo Motor](image)
The next main component to be evaluated is the Microcontroller. The decision on this part selection could be the difference between a successful project and a failed experiment. If the microcontroller is picked that no one on the team is familiar with, there will be a challenge to learn the language and implement the design as needed. Thus, it is crucial to pick a microcontroller that someone on the team is familiar with. Next, the power and size are the most important features to keep battery requirements and bulk down. Finally, the cost and availability are compared against each other and it was found that the Raspberry Pi Zero W is the optimal choice for this design.

Table 2: Decision Matrix for Microcontroller

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RPI 3B</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>RPI ZERO</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>71</td>
</tr>
<tr>
<td>RPI 4B</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>54</td>
</tr>
<tr>
<td>TNKBRD</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>39</td>
</tr>
<tr>
<td>MSP430</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>48</td>
</tr>
</tbody>
</table>

Figure 33: Raspberry Pi Zero W

Table 3: Decision Matrix for Housing / Food-Safe Material

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Stainless Steel</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>46</td>
</tr>
<tr>
<td>Glass</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>37</td>
</tr>
<tr>
<td>Plastic</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>36</td>
</tr>
</tbody>
</table>

Table 3: Decision Matrix for Housing / Food-Safe Material

Figure 34: Stainless Steel
The largest component is the one responsible for keeping the device together as well as protecting it is the material for the storage compartment and circuitry housing. This material should also be used for the dispensing disk as it will be food-safe and easy to clean. The main requirement here is for the material to be food-safe and durable to ensure the device lasts. The winner here is stainless steel, but another metallic alloy like aluminum could also be used.

<table>
<thead>
<tr>
<th>Battery Type</th>
<th>Voltage</th>
<th>Cost</th>
<th>Availability</th>
<th>Weight</th>
<th>Capacity</th>
<th>Environmental</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alkaline</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>65</td>
</tr>
<tr>
<td>Nickel-Metal Hydride</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>86</td>
</tr>
<tr>
<td>Nickel Cadmium</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>55</td>
</tr>
<tr>
<td>Lithium ion</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>86</td>
</tr>
<tr>
<td>Nickel Zinc</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>69</td>
</tr>
</tbody>
</table>

*Table 4: Decision Matrix for Rechargeable Batteries*

*Figure 35: Nickel-Metal Hydride (Left) and Lithium Ion (Right) Rechargeable Batteries*

Probably the most difficult material to compare and decide on is the type of rechargeable battery to use. Here, the most important aspect is the environmental impact. Since the point of new technology is to improve lives, unnecessary waste is not an option. Next, since batteries can be heavy and weigh down the device, the weight is taken into consideration. Availability of batteries can also be a huge issue. There are many unique batteries so it is important to choose those that can be easily attained [23]. Next, we look at capacity, cost, and voltage. Again, the costs of almost every component is very small as compared to some electronics so the cost is not an issue. The capacity, however, is important in ensuring that the device will stay powered for at least a day at a time between battery charging [24]. The least important component is voltage since batteries can be added in series to get to required voltages, though fewer batteries is better.
Based on this criteria, two batteries tied in the decision matrix: Lithium Ion, and Nickel-Metal Hydride. This is actually good since the 555 Timer design and the Microcontroller design have different power requirements. Based on research, the Nickel-Metal Hydride batteries will be better suited for the 555 Timer design since the power requirements are less than that of the microcontroller design [25]. The Lithium Ion batteries make a good fit for the Microcontroller since they can be purchased as a battery pack with 5 Vdc USB outputs (power for Raspberry Pi) and 12 Vdc mono-jack output, providing flexibility and ease of use.
Prototyping Progress Report

Schematic & Simulation Result

After performing some initial prototyping, it was determined that the circuit as simulated at the beginning of the design phase (Figure 15) was not going to be feasible. After some testing, the circuit as shown in Figure 36 below was built and found to operate the circuit as needed at a very basic level. The timing is still not correct and will need to be modified to match the servo motor, but it allows for the treat dispensing area to be designed and tested with the actual driving mechanism. This new circuit was simulated as shown in the right portion of Figure 36.

![Figure 36: 555 Timer Circuit in Astable Operation](image)

As can be seen in Figure 36’s Digital Oscilloscope, the PWM (yellow) signal is successfully generated for driving the servo motor. The servo motor made a rotation of positive 90 degrees during the testing phase.
Breadboard Circuit Testing

Figure 38: Breadboard Prototype

Comments: Breadboard simulation was carried out to check the functionality of the schematic shown in Figure 36 above. Real time breadboard simulation differed with the computer simulation as slightly different capacitor values were used.
Cardboard Prototyping

Utilizing cardboard, the treat dispensing mechanism was able to be visualized, and functionality tested as shown in Figures 39 through 41 below.

![Figure 39: Top View of the Dispensing Disk]

![Figure 40: Bottom View of the Dispensing Disk]
Raspberry Pi Prototyping

```
MacBook-Pro-5:~ gagemoore$ ssh pi@10.0.0.99
pi@10.0.0.99's password:
Linux Gages-pi 4.19.75+ #1270 Tue Sep 24 18:38:54 BST 2019 armv6l

The programs included with the Debian GNU/Linux system are free software:
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Nov 2 06:52:23 2019
pi@Gages-pi:~ $ ls
Desktop Downloads Music Public Videos
Documents MagPi Pictures Templates
pi@Gages-pi:~ $ cd Desktop
pi@Gages-pi:~/Desktop $ ls
*pi@Gages-pi:~/Desktop $ cd ..
pi@Gages-pi:~ $ ls
Desktop Downloads Music Public Videos
Documents MagPi Pictures Templates
pi@Gages-pi:~ $ /home
-bash: /home: is a directory
```

Figure 41: Prototype Testing of the Dispensing Mechanism

Figure 42: SSH into Raspberry Pi
Shown above in figure 42, the raspberry pi was effectively connected to via Secure Shell (SSH). SSH is secure shell encrypted protocol which allows for secure file transfer between the raspberry pi and a computer to load the files onto the microcontroller. This allows for our python files to be transferred to the raspberry pi itself in which it can run the appropriate code to control the treat dispenser.

![Servo Driver Code](image1)

**Figure 43: Servo Driver Code**

Shown above is the actual code that is used to control the motor during our prototyping efforts. The code uses a series of import statements to help set up the General Purpose Input and Output ports of the Raspberry Pi. The code simply runs a loop three times that spins the motor clockwise for a second and then turns it off.

![FileZilla SSH Client Usage](image2)

**Figure 44: FileZilla SSH Client Usage**
Figure 44 above shows how the application FileZilla (an SSH client to help migrate code over to the Raspberry Pi) is used. All that is needed to be known prior to using the tool is the ip address of the Raspberry Pi, along with the login credentials. Once we were able to log into the Raspberry Pi, we were able to successfully migrate the appropriate motor driver code onto the Raspberry Pi.

![Image of command prompt output]

*Figure 45: Successful Operation of Code*

Figure 45 above shows how the motor driver file was successfully run on the Raspberry Pi, and how exactly to do it using the command prompt from the Raspberry Pi. Figure 36 below depicts the microcontroller hooked up to the servo motor in order to control the device. The code successfully ran on the Raspberry Pi indicating that we were able to control the motor through this code as a good prototyping effort.
Options to be Considered (Not Implemented)

Tests were carried out to test the servo motor. However, team has considered using Linear Solenoid Actuator to replace servo motor, should the circuit or the design fail. This actuator can be simply used to create a linear dispensing mechanism or create a similar rotary dispensing mechanism like of servo by converting linear motion to rotary motion.

Solenoid Actuators convert electromagnetic energy into mechanical energy resulting in a push/pull mechanism. It provides excellent force, but requires higher voltage and power (12V or more depending on its size) and is slightly more expensive than a Servo motor. This device could be easily implemented to activate a physical clicker in the device if the customer determines it is needed.
Solenoid Control Module

Comments: Solenoid control module prototype was created using cardboards to visualize the motor and parts movements. Working video of this module can be found on our website treatdispenser.onmason.com under the prototyping tab. The above prototype translates the lateral motion of the solenoid actuator to a rotational motion that can be used to activate the treat dispensing module.
Testing Plan for ECE 493

There are several test cases that the design team will perform in a controlled lab environment before involving service dogs and their owners. In order to start building our test cases we look back into our original functional and operational requirements and begin framing our test cases along those lines to ensure that we meet both the customers expectations and that the scope of the work does not fall victim to scope creep.

Functional Requirements:

1. The device will hold a minimum of a cup of treats at all times to provide rewards for the service animal.
2. The device shall use an accelerometer-based control input so that someone with limited physical mechanics can easily use the dispenser.
3. The device shall use timers to time and control dispensing mechanism.

From these requirements the first basic test cases can be derived. The first test case will be to ensure that the storage mechanism can properly hold a cup of treats, receive input from a switched device, and dispense a treat. This test case can fail in the following ways:

1. The storage device fails to hold the minimum amount and tips over and spills food
2. The storage device collapses after multiple dispenses
3. The storage device hole that food goes through regularly gets stuck
4. The device fails to accept accelerometer input
5. The device misinterprets the accelerometer input
6. The timers fail to perform as expected
7. The timer integration with the dispensing mechanism fails to behave as expected

Each function in our design will need to have incremental testing done on it from the simplest operations to the most complex operation. With testing involves using alternative solutions like the solenoid as a replacement incase the current servo motor isn’t the most optimal solution.
Preliminary Project Plan

List of Major Tasks

- Hardware procurement (1 Week)
  - 555 Timers
  - Rotary Device
  - Switches

- Hardware development (3 Weeks)
  - PCB Design
  - PCB Assembly
  - Microcontroller evaluation
  - Power evaluation
  - Storage Container

- System Integration (3 Weeks)
  - System rotary design
  - Variable output functionality
  - Accelerometer functionality

- Wheelchair Integration (1 Week)
  - Placement verification

- Testing (2 Weeks)
  - Experiment #1 and #2

- Data Analysis and Refactor (1 Week)
  - Feedback review and data review

- Reporting (2 Weeks)
  - Initial Progress Report
  - Mid-Flight Report
  - Final Report

- User Acceptance Testing (2 Week)
  - Experiment #1 and #2

- Faculty Demos (Part of reporting)
  - Demo #1, 2, 3

- Customer Demo (Part of reporting phase)
  - Demo #1
# Gantt Chart for ECE-493

**Treat Dispenser**  
FS: Nathalia Pelucio  
PM: Adam Dost

Members: Jachin Shrest, Zainab Abdullahi, Robert Wignall, Gage Moore

<table>
<thead>
<tr>
<th>TASK</th>
<th>NOTES</th>
<th>PROGRESS</th>
<th>START</th>
<th>END</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial integration</td>
<td></td>
<td>0%</td>
<td>1/15/20</td>
<td>1/25/20</td>
</tr>
<tr>
<td>Hardware Procurement</td>
<td></td>
<td>0%</td>
<td>1/15/20</td>
<td>1/25/20</td>
</tr>
<tr>
<td>Hardware Development</td>
<td>PCB / MCU</td>
<td>0%</td>
<td>1/15/20</td>
<td>1/25/20</td>
</tr>
<tr>
<td>System Integration</td>
<td>DESIGN 1 &amp; DESIGN 2</td>
<td>0%</td>
<td>1/23/20</td>
<td>1/29/20</td>
</tr>
<tr>
<td>Wheelchair integration</td>
<td>DESIGN 1 &amp; DESIGN 2</td>
<td>0%</td>
<td>2/15/20</td>
<td>2/20/20</td>
</tr>
<tr>
<td>User Acceptance Testing</td>
<td>DESIGN 1 &amp; DESIGN 2</td>
<td>0%</td>
<td>3/1/20</td>
<td>3/2/20</td>
</tr>
<tr>
<td>Analysis and refactor</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Analysis and Refactor</td>
<td></td>
<td>3/8/20</td>
<td>3/15/20</td>
<td></td>
</tr>
<tr>
<td>User Acceptance Testing 2</td>
<td></td>
<td>3/16/20</td>
<td>3/17/20</td>
<td></td>
</tr>
<tr>
<td>Reportings</td>
<td></td>
<td></td>
<td>3/15/20</td>
<td>3/26/20</td>
</tr>
<tr>
<td>Additional Features</td>
<td>MCU Specific Focus</td>
<td>3/20/20</td>
<td>4/20/20</td>
<td></td>
</tr>
<tr>
<td>Presentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faculty Demo</td>
<td>TBD</td>
<td>TBD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer Demo</td>
<td>TBD</td>
<td>TBD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
References


