WHEELCHAIR-MOUNTED DOG TREAT DISPENSER

In-Progress Report

v1.0

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

Faculty Supervisor

Dr. Nathalia Peixoto

Client

Service Dogs of Virginia

George Mason University | Volgenau School of Engineering

Service Dogs of Virginia

Table of Contents

Technical Section

- **Hardware Procurement (Began in ECE 492):** ***Week 1***

Raspberry Pi's provided to all team members.

Knowledge/Tutorial provided to all members on bootstrapping through tools like PiBakery.

- **Hardware Development and Module Implementation: Week 2-5**

1.1 Circuit Designs:

1.1.1 Infrared Proximity Sensor:

The IR Proximity sensor will be used for two different functions. The first is to indicate the level of treats in the x and the second is to detect whether a treat has been dispensed on the tray or not.The circuit shown below in Figure 1, depicts how the IR Proximity sensor will operate. This circuit will indicate the level of treats by lighting up an LED, a green LED for high level and a red LED for a low level of treats. This circuit can be broken down into two parts. The first part consists of a push button, 555 timer, transistor and a capacitor. The main functionality of this part of the circuit is to lower the power consumption by acting as an on/off button. The capacitor value can be adjusted and will control the amount of time the output LED stays on.The second part of the circuit consists of a potentiometer, a pair of IR LEDs, an LM358 integrated circuit and an output LED. The potentiometer controls the sensitivity and range of the detection[1]. Finally, pin 1 of the LM358 is connected to the output LED. To determine how long the output LED stayed on, the following equation was used, T=1.1*R*C(seconds). The circuit was tested using a 100µF capacitor and a 10kΩ resistor giving us a theoretical value of 11 seconds. The experimental value was 8.3 seconds, giving us an error of 24.5%.

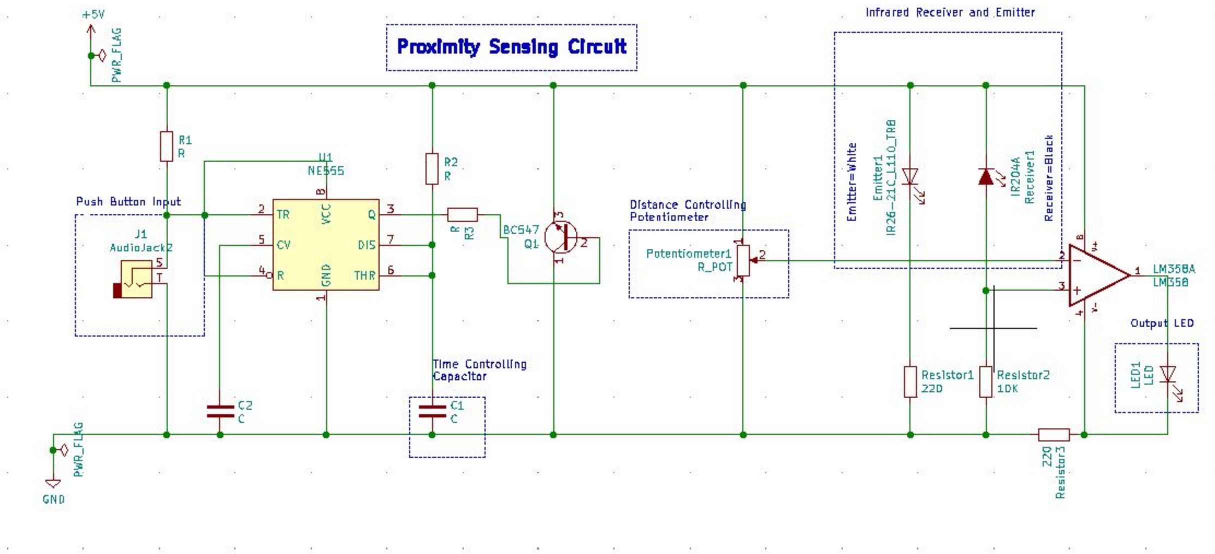

Figure 1: Infrared Proximity Sensor with Push Button

The PCB that detects the level of treat will be placed on the inside of the dispensers lid, which is why we decided to use it to utilize the IR LEDs since they also function in the dark. A video demonstrating this can be shown here(http://treatdispenser.onmason.com/prototyping/).

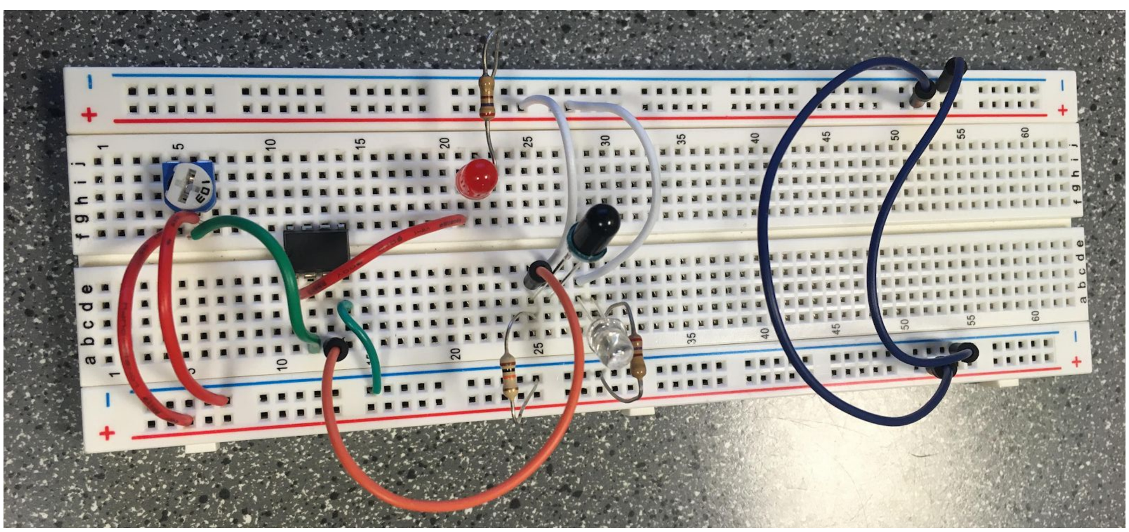

Figure 2: Breadboard Implementation for Proximity Sensor Circuit

1.1.2 555 Timer Circuit :

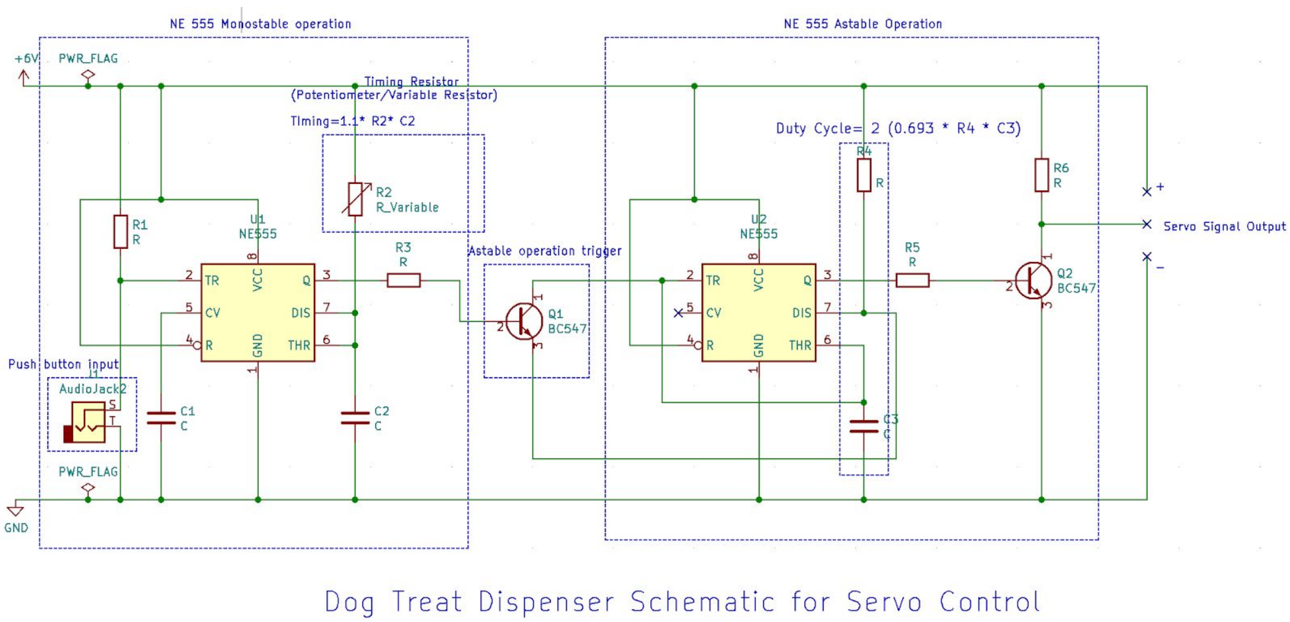

Figure 3: 555 Timer Servo Control Circuit

Servo Control Circuit:

There are two 555 timers used for the operation of servo motor. The first 555 timer is set in monostable mode[2] and is used to trigger and time the following 555 timer. The second 555 timer is set in Astable mode[3] which generates PWM signal required for operation of servo motor. Second 555 timer is active as long as it gets triggered from the first 555 timer. The timing of the first 555 timer is calculated by the equation: Timing= 1.1 * R2*C2 as seen in the figure above. Similarly, the duty cycle of the second 555 timer can be calculated by the equation. Duty cycle= 2*0.693 *R4 *C3.

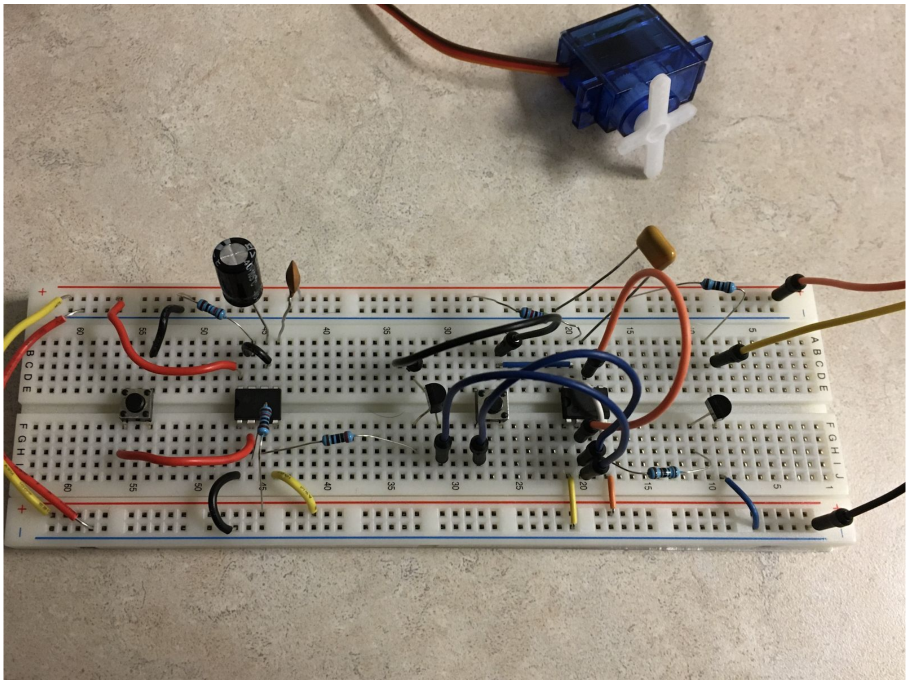

Fig 4: Breadboard Implementation for the Servo Control Circuit.(Working Video on http://treatdispenser.onmason.com/prototyping/)

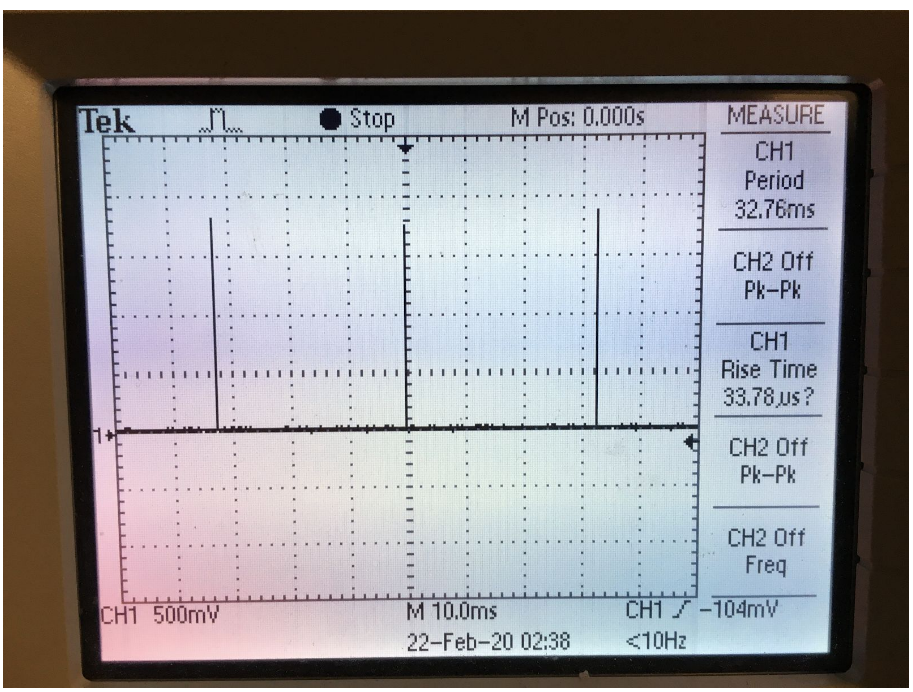

Fig 5: Duty Cycle Waveform(PWM) Observed from The Second 555 Timer

1.2 PCB:

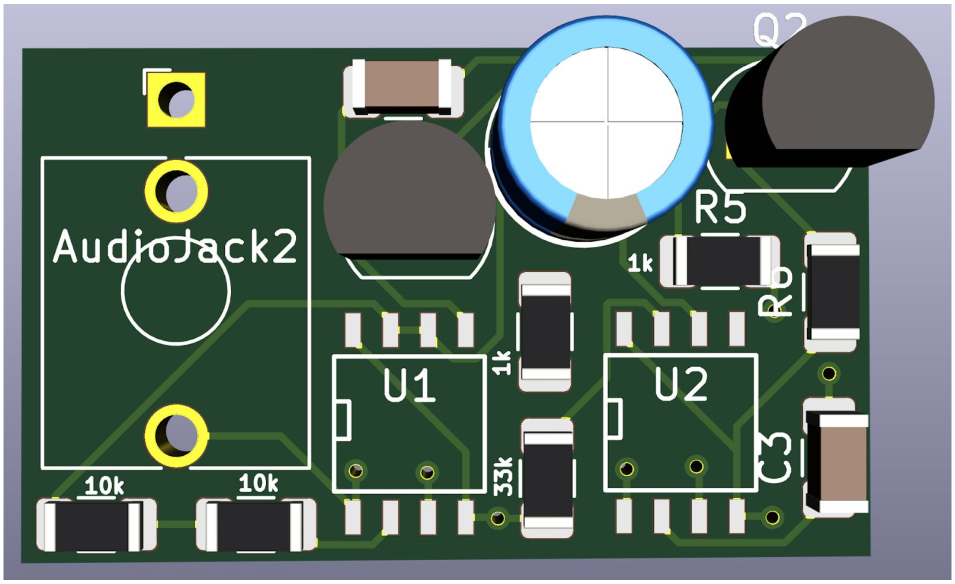

Fig 6: PCB Implementation of the 555 Timer Circuit (Final Component Placement will Differ)

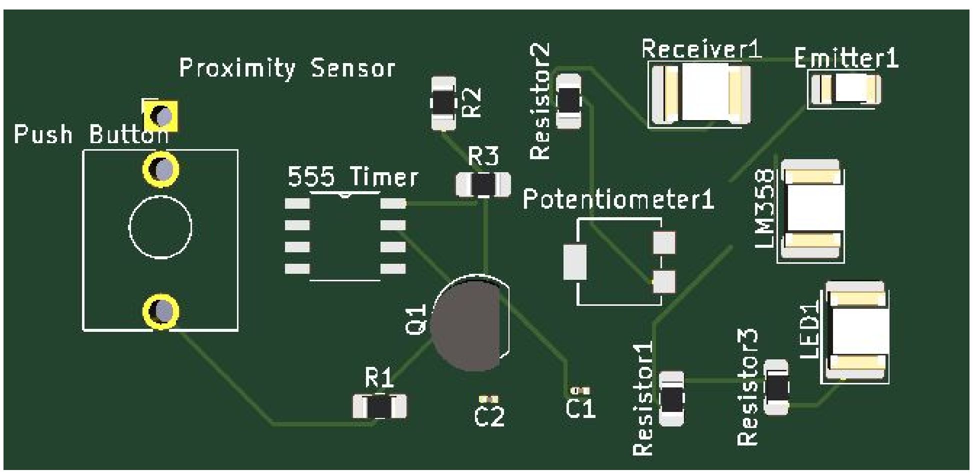

Fig 7: PCB of the Proximity Sensor Circuit(Final Component Placement Will Differ)

1.3 Power Evaluation:

Power evaluation was conducted for four different modes using a multimeter for the 555 Timer circuit design.

Normal mode: Operating the servo motor only.

Continuous mode: Standby current draw by circuit.

Proximity Sensor On: Operating servo motor with feedback LED.

DC Motor On: Powering on vibrating motor.

Based on the table below, the largest power draw is going to be when the vibrating motor is used to dislodge treats. This has a power consumption of 4.392 Watts based on initial tests. This will be used as an alternate input so it cannot consume power draw accidentally.

The treat dispenser is planned to be used 30-50 times per day, which will involve the Normal and Continuous mode or the Continuous and Proximity Sensor On mode. More power testing will need to be conducted to find the required battery sizing.

Mode	**Voltage(V)**	**Current(A)**	**Power Consumed(W)**
Normal	6	0.22	1.32
Continuous	6	0.04	0.24
Proximity Sensor On	6	0.305	1.83
DC Motor On	6	0.732	4.392

As seen in Figure 8 below, the proximity sensor will illuminate an LED while operating the servo motor. However, the clicker sound speaker has not been taken into account yet. The proximity sensor will need to also work as the feedback and tell the circuit whether or not to activate the clicker sound. That way, the sound is not activated unless a treat is actually dispensed.

There will be a proximity sensor used in the treat storage compartment that will activate once the treat compartment gets low and there will be one used at the end of the dispensing tube to confirm a treat is dispensed. See Section 2.8 for more details on the housing and sensor placement.

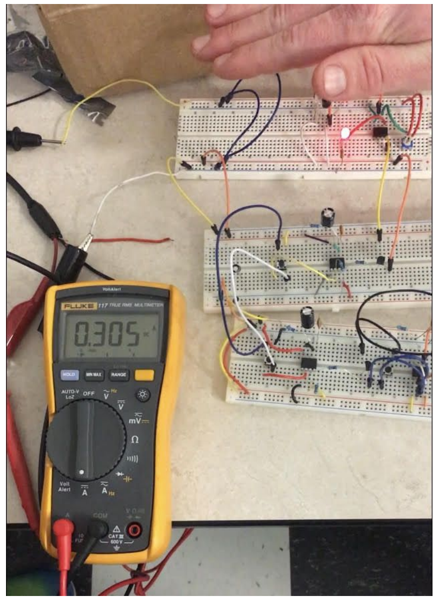

Figure 8: Current with Motor Activated and Proximity Sensor On

1.4 Microcontroller Evaluation:

1.4.1 Raspberry Pi:

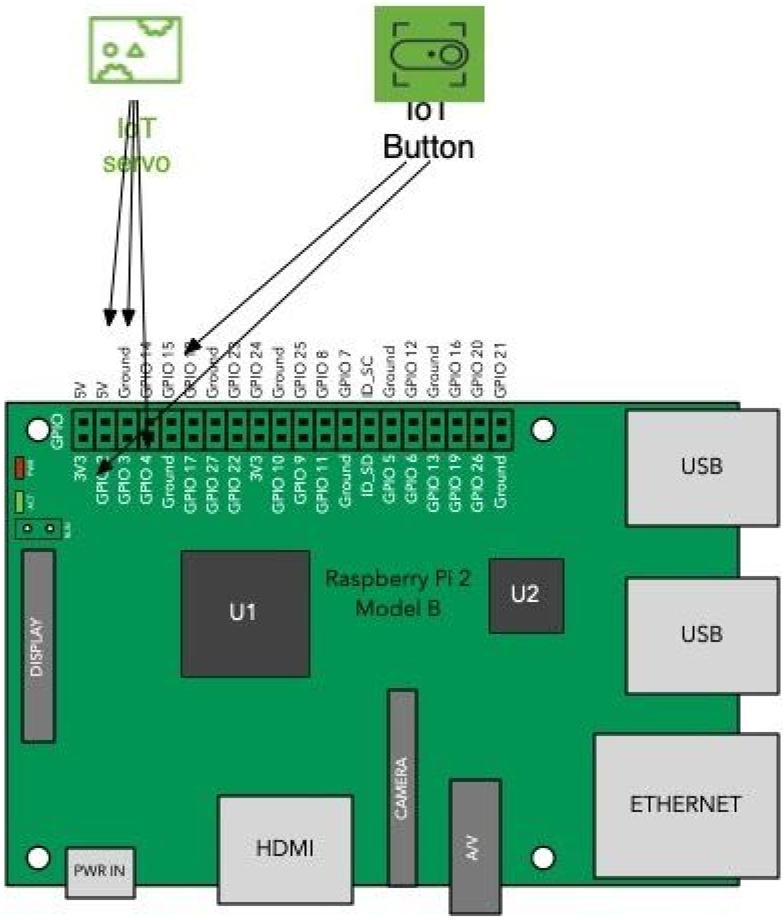

Figure 9: Raspberry Pi

During the initial evaluation of the Raspberry Pi we had to ensure that we could leverage the available power supply without causing it to short circuit. Because none of the tasks the Raspberry Pi will be in parallel then we do not need to worry about pulling too much power from the board itself. This can be verified in our workflow code as we never execute anything in parallel with each other. (https://github.com/asoccer/Treat-Dispenser/blob/master/TDCode/adam/TreatDispenserCode.py) The workflow that the RaspberryPi will take when evaluating actions will be the following.

Figure 10: Raspberry Pi workflow diagram

As an illustrative example, provided below is the raspberry pi command window (pictured in the upper left) and the code that successfully spins the servo exactly 360 degrees on every rotation command as Figure 10. Moreover, a picture of the raspberry pi set up with the activation button and the servo is pictured below as Figure 11 [4].

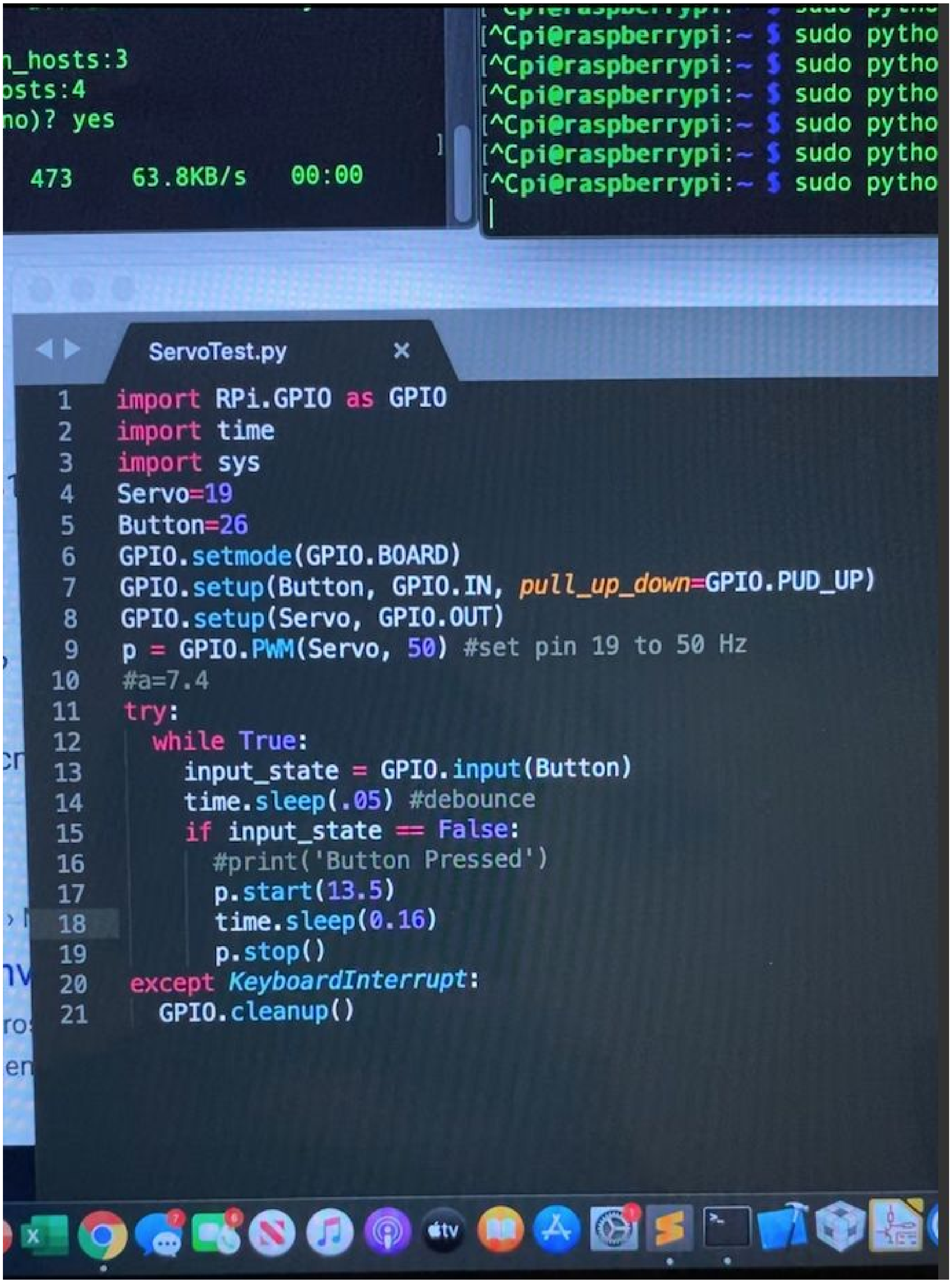

Figure 11: Raspberry Pi Command Window and Servo Code

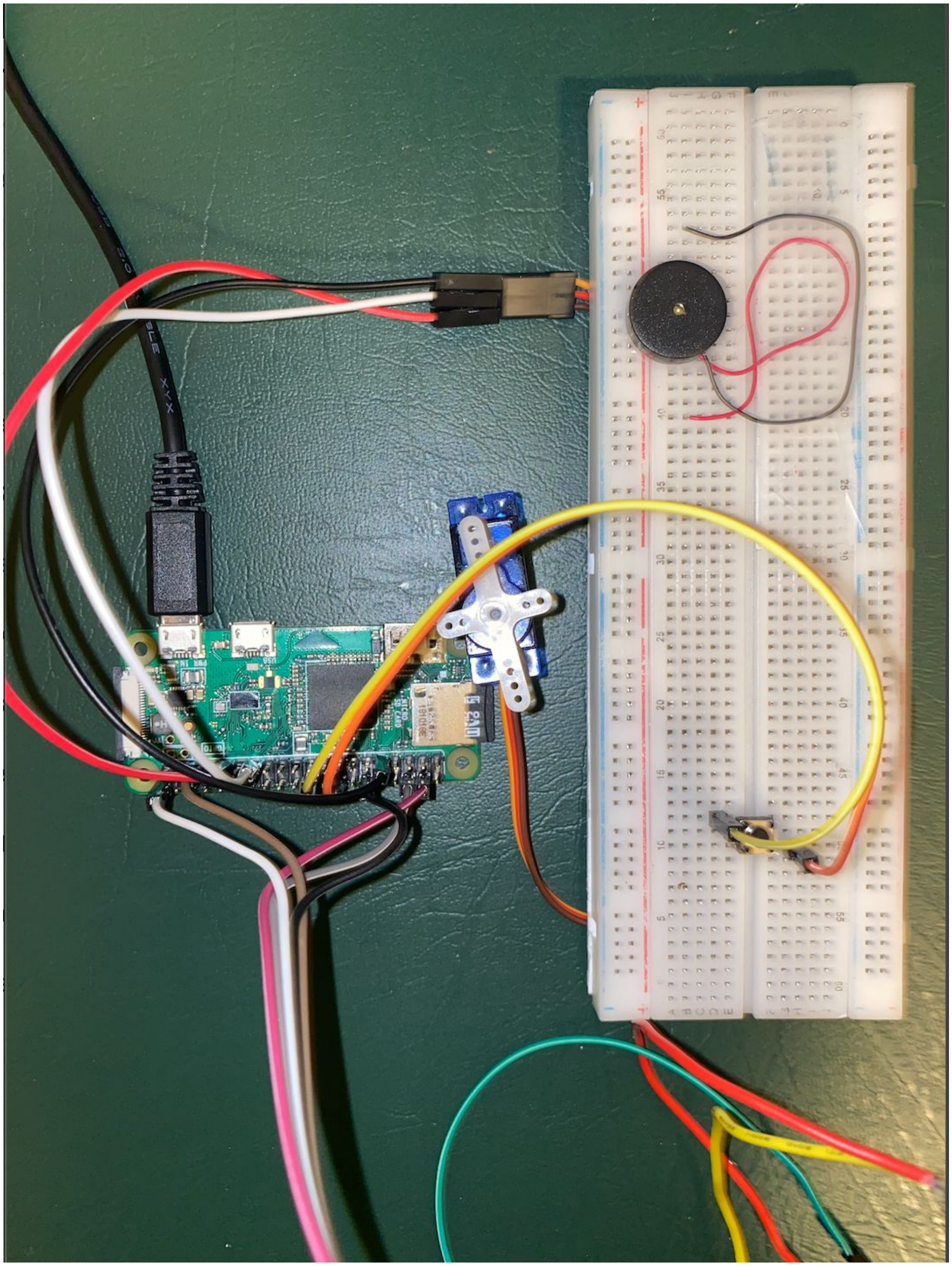

Figure 12: Raspberry Pi Servo Control Setup

1.6 Clicker Evaluation:

In order to replicate the same clicker sound that the service dogs are most familiar with we needed to emulate the same sound from both the PCB and microcontroller side. For the microcontroller we successfully were able to play a .wav file of the click sound through a speaker connected via GPIO to the Raspberry Pi. A video demonstrating this can be found here: (http://treatdispenser.onmason.com/prototyping/). For the written report however, an image of the button, speaker, raspberry pi, and adafruit MAX98357 amplifier working together as Figure 12 [5].

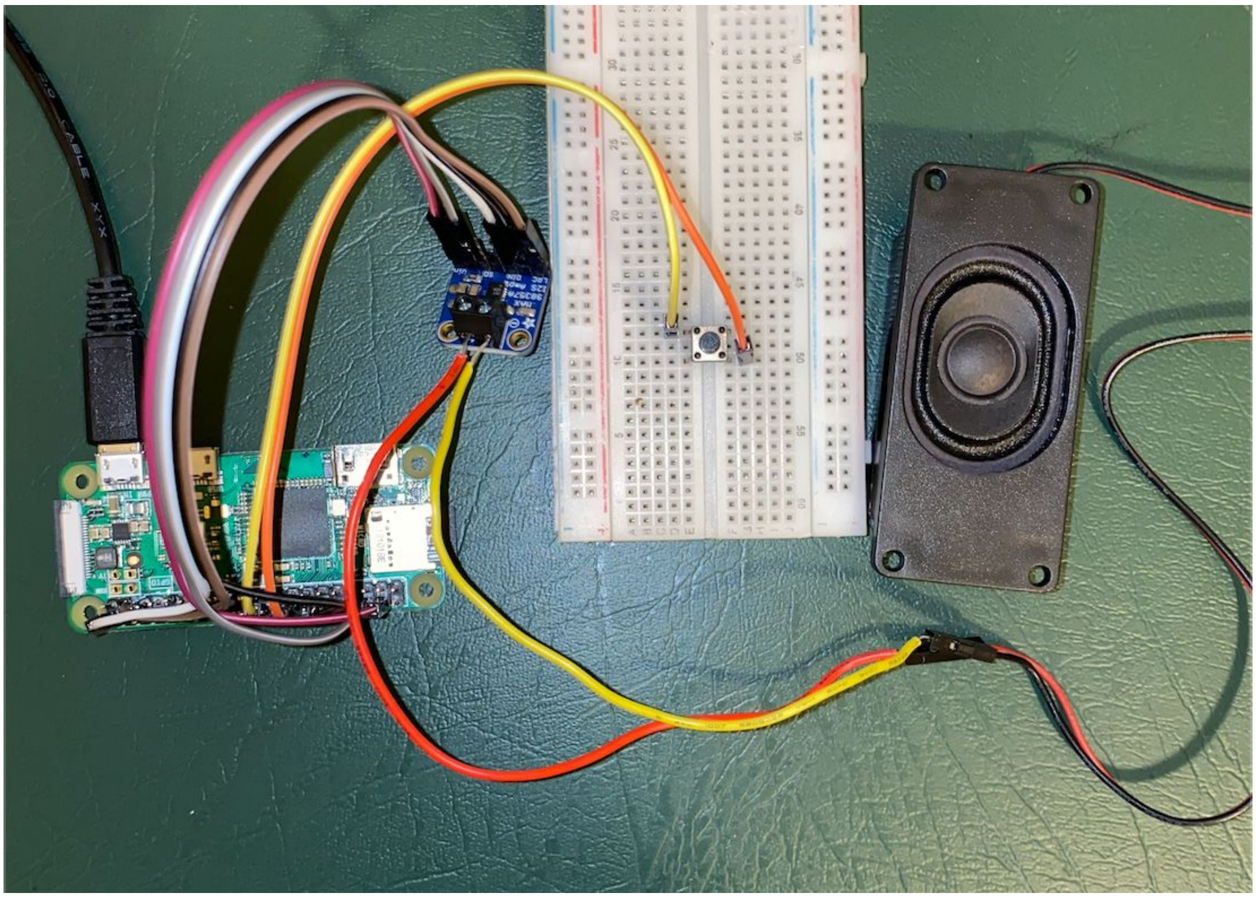

Figure 13: Raspberry Pi and Associated Hardware for Click Sound Module

In order for the speaker to work, we utilized the MAX98357 amplifier to ensure that the dog can hear the noise. Based on our review, we believe that the click noise is sufficient for a dog to react to. Furthermore, after we complete testing, if we need to adjust the volume of the click sound we will be able to through the Raspberry Pi [5,6]. Provided below as Figure 13 is a picture of the Raspberry Pi's command line to test the speaker sound. This test ran successfully.

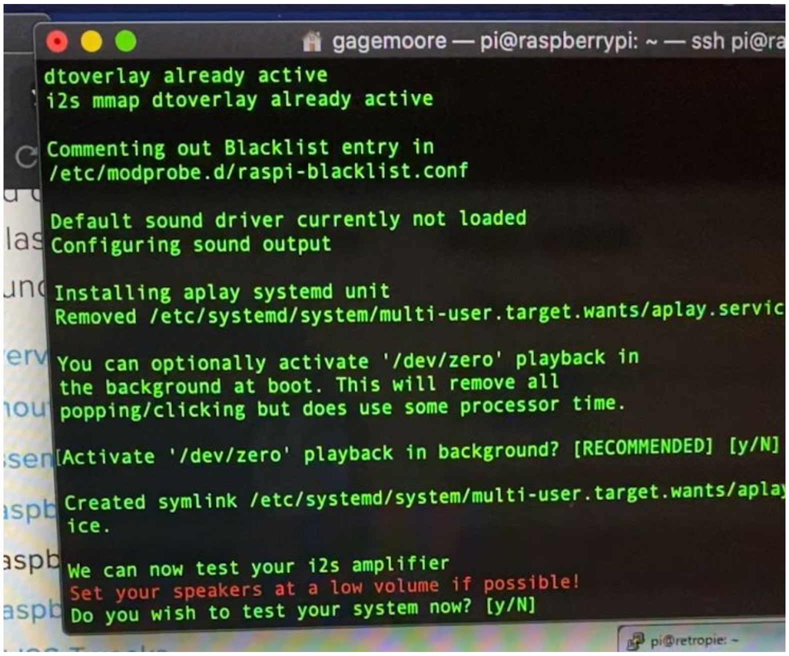

Figure 14: Sound Module Test

1.7 Vibrating Motor Evaluation:

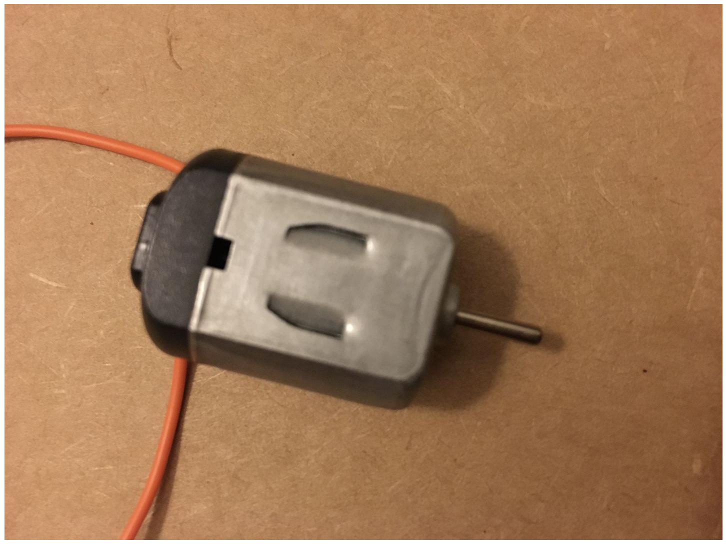

Figure 15: Vibrating Motor

The vibrating motor will be placed on the outer-side of the treat storage area. A DC motor will be used. This motor will be connected to a switch, allowing the user to turn the motor on whenever treats become jammed. The working can be seen on our website (treatdispenser.onmason.com/prototyping.)

1.8 Housing and Tray:

The gears are taken directly from the gumball machine as the machine provides consistent results. The idea is to rebuild these gears in a treat dispenser that works automatically. The gumball machine requires too much force to activate so the gears need to be on bearings and freely spinning. Images of the gears to be used are shown in Figure 16 below.

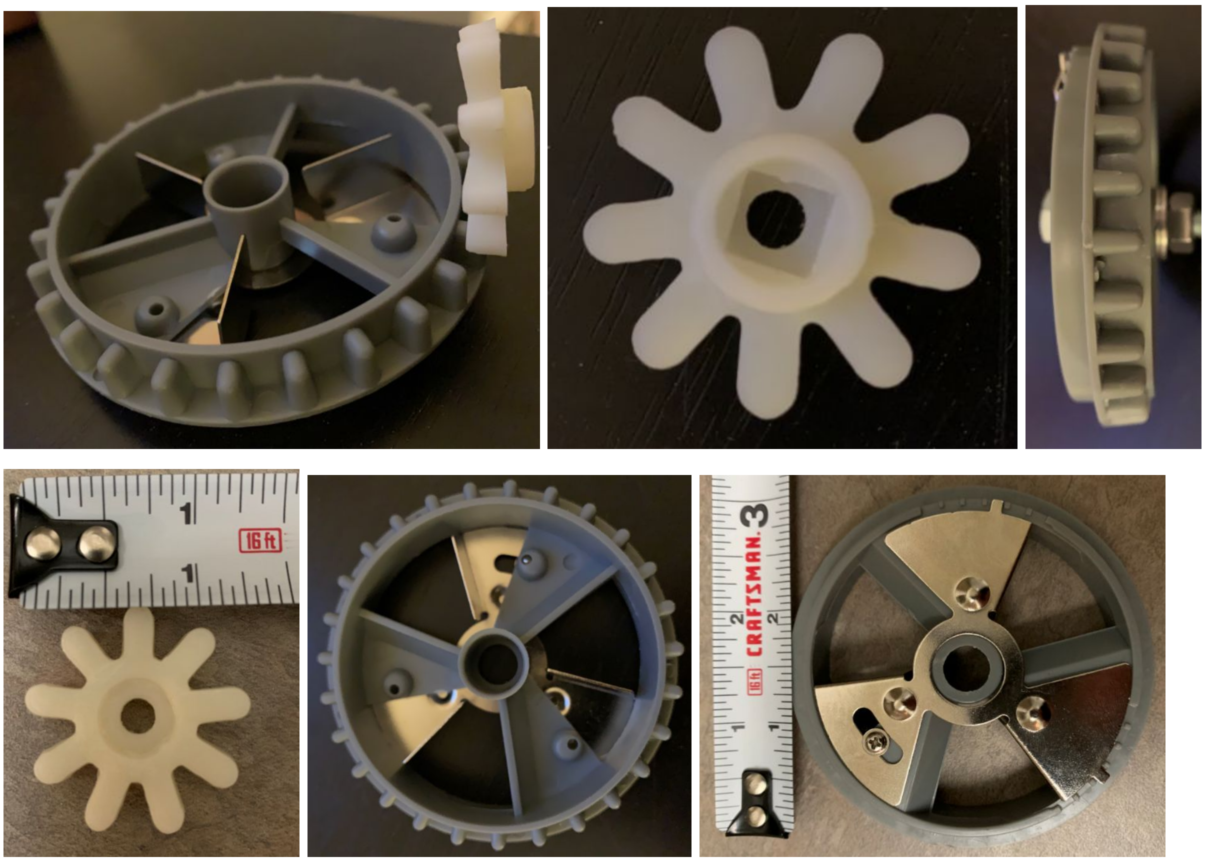

Figure 16: Gear measurements

The dog dish to be used will be a small steel dog food dish. That way, it will be resilient to dogs retrieving their treats and easy to clean. It will be attached to the vinyl dispensing tube using a small bolt. There will also be a mounting bracket and velcro straps attached to the bottom of the dispensing tray to provide solid mounting to the wheelchair. Images of the dog dish to be used can be seen in Figure 17 below.

Figure 17: Dog Bowl

The dog treat dispenser housing will be mostly made of pvc and plastic. Initially, stainless steel was the plan, but this is a difficult material to work with. It will still be used with the dog tray so the material the dog has direct contact with is as resilient as possible. There will be PVC screw connections between modules as shown by the red outlined cylinders in Figure 18 below. The electronics will be housed in a waterproof container separate from the rest of the treat dispenser. There will be a clear cover so diagnostic LEDs can be seen and circuitry as well. Vinyl tubing will be used for the dispensing tube as it is flexible and provides many options for mounting the device.

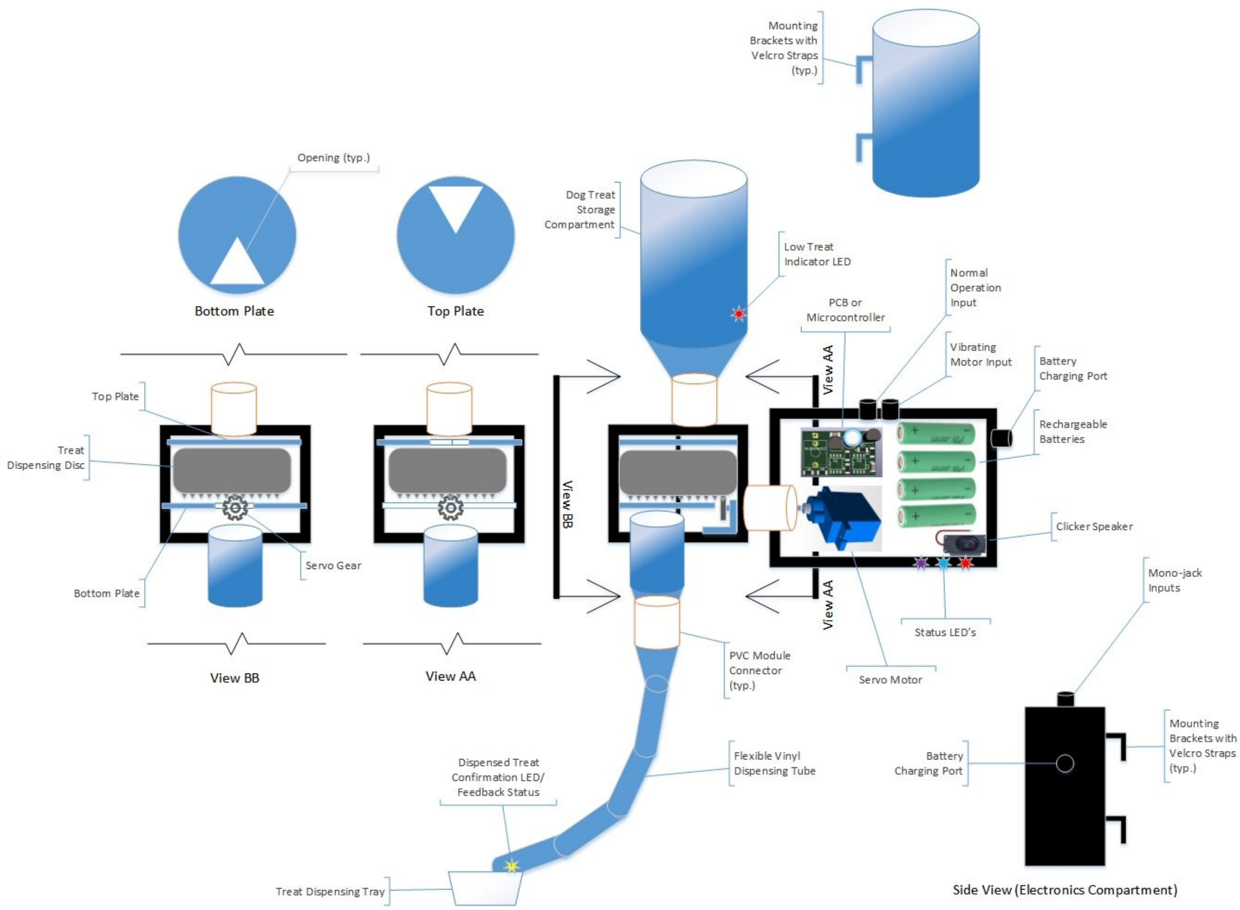

Figure 18: Container Overview

- **Software Development: Week 4-5**

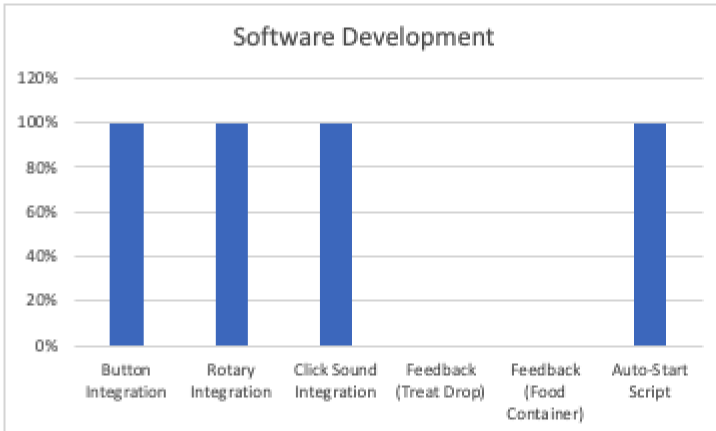

Figure 19: Completed Software Development

Code Repository: https://github.com/asoccer/Treat-Dispenser

Software Development	Progress	Validation
Button Integration	100%	Source Code
Rotary Integration	100%	Source Code
Click Sound	100%	Source Code
Feedback (Treat Drop)	0%	
Feedback (Food Container) Auto-Start Script	0% 100%	 Source Code

Administrative Section

• Funds Spent at This Point

- BC 547 Transistor pack: $5.28
- Resistor Kit: $7.42
- Capacitors, Breadboards, Jumper wires kit: $13.78
- 5x 9v-Batteries: $5
- 3x 9v Battery Connectors: $6
- 2x Adafruit I2S 3W Class D Amplifier Breakout - MAX98357A: $11.99 each
- CQRobot Speaker 3 Watt 4 Ohm Two Pack: $7.99
- Dog Treat: $4.22

• Man-hours to This Point

- 555 timers study: 5 hrs
- Sensor study: 3 hrs
- Circuits study: 10 hrs
- Circuit simulations: 6 hrs
- Breadboard circuit building: 15 hrs
- PCB designs: 5 hrs
- MicroController Rotary Code: 2 Hours
- MicroController BootStrap Code: 1 hour
- MicroController Sound Module: 2 Hours
- MicroController Rotation Timing: 4 Hours
- Visio Sketches: 4 Hours
- Material Procurement & Encasing for Device: 10 Hours
- Weekly Meetings and Discussions: 25 Hours
- Group Working Sessions: 24 Hours

Plans for the Next Reporting Period (In-Progress Review)

• Which tasks to work on?

For the next reporting period, the tasks we would like to work on is to fully integrate all of the separate parts together to make one cohesive unit. For example, although the servo is timed correctly with no load on it to spin exactly 360 degrees, we do not know for sure if this will change with the load on it. The mentioned load would be simply spinning the handle of the gumball machine and adjusting (if needed) the timing cycle on how long the servo should spin for. Thus, this is a task we want to fine tune for the final design. Another task we would like to work on is ensuring (through testing) that the “click” sound is loud enough for the dog to hear and react to once a treat is dispensed. Based on our testing thus far, we believe that the “click” sound produced by the speaker is loud enough for most people to hear even when in a noisy environment. As of now, the only other tasks on the microcontroller side of things to complete will be properly interfacing the proximity sensor to measure the amount of dog treats currently in the treat staging area, and to integrate all of these parts together. Another task we would like to work on for the PCB side of the design would be to finalize the PCB board we will print, and solder the resistors, capacitors, and associated chips to the fabricated PCB.

• What kind of demo(s) to be presented?

The demos to be presented for the next progress report would hopefully be a fully working project. If we are able to make this happen, we will have more than enough time to fully test each system and polish off any remaining issues with the project we are currently facing. This demo would involve not only making sure that the treat dispenser dispenses a treat, but also ensuring that it won’t scare the dog (due to noise) or make sure the dog can respond to the “click” sound (which indicates a treat has been dispensed). In order for us to present this demo, we hope that Grady, Dr. Nuebers service dog, would be available for such a demo. If everything goes well with Grady, we would like to take a trip down to the Virginia Service Dogs center located in Charlottesville to present a demo of a fully functional treat dispenser once functional and safety testing is complete.

• Which problem areas to be solved?

Other than the aforementioned problems with the load of the servo possibly being an issue once we integrate it with the gumball machine, we would like to work more with the PCB. The PCB as is currently will be using SMT components. With the size of SMT being so small, we believe that we will have a couple issues come up when initially trying to solder them.

On the microcontroller side the current challenge is going to be interfacing different inputs with each other. Currently we have tied the button’s switch to the rotation of the servo and will need to then add evaluation boolean’s for the proximity sensor and the container level sensor to ensure that the proper values are being evaluated and not false positives. Click sound module should never activate until the proper checks have passed.

On the hardware portion of the design, the team is currently researching on a shaft or a bolt to integrate to servo for fitting a gear that will be rotating the dispensing disk.

Provide Answers to the Following Questions

a) Is the project on schedule?

Breadboard simulations and PCB designs on schedule
Proximity sensors on schedule
Raspberry Pi control, sound modules on schedule

b) Are there any problem areas causing project delays, etc.?

Mechanical activation of the click sound module for the PCB (because the force is too strong at the moment).

c) Plan to deal with problems/delays.

As an option we are planning on possibly using a thinner piece of aluminum that is stamped as the same shape in order to reproduce the sound with less force. As another option we will go with using a sound recording device that stores the sound in order to replicate the click sound.

References

[1] Mohammad, T. (2009). Using Ultrasonic and Infrared Sensors for Distance Measurement. WorldAcademy of Science, Engineering and Technology.

[2] *555 Timer Monostable Circuit Diagram*, 29-Sep-2015. [Online]. Available: https://circuitdigest.com/electronic-circuits/555-timer-monostable-circuit-diagram. [Accessed: 22-Feb-2020].

[3] Instructables, "Drive Servos With a 555 Timer IC," *Instructables*, 20-Oct-2017. [Online]. Available: https://www.instructables.com/id/Drive-Servos-with-a-555-timer-IC/. [Accessed: 22-Feb-2020].

[4] R. V1.1, "Raspberry Pi Zero W V1.1", *Smart Prototyping*, 2019. [Online]. Available: https://www.smart-prototyping.com/Raspberry-Pi-Zero-W-V1.1. [Accessed: 01- Nov- 2019].

[5] "Continuous Rotation Micro Servo [FS90R] ID: 2442 - $7.50 : Adafruit Industries, Unique & fun DIY electronics and kits." [Online]. Available: https://www.adafruit.com/product/2442?gclid=Cj0KCQjw0brtBRDOARIsANMDykZM_o-SLs9Gf7AtLB_AFkEGiJs20nIDk34L36z1GMfcGGiof2Btzl0aAvcYEALw_wcB. [Accessed: 22-Oct-2019].

[6] "FAQs - Raspberry Pi Documentation." [Online]. Available: https://www.raspberrypi.org/documentation/faqs/#pi-power. [Accessed: 22-Oct-2019].