Weed Robot
Final Design Report

Team Weed

Claire Franz
cjfranz@calpoly.edu

Lewis Kanagy
lkanagy@calpoly.edu

Carlos Rodriguez
crodr137@calpoly.edu

Kaela Stern
kstern02@calpoly.edu

Team Scouts

Jackie Chen
jchen329@calpoly.edu

Sachi Hiji
shiji@calpoly.edu

Ayush Kakkanat
akakkana@calpoly.edu

William Ta
wita@calpoly.edu

Advisor

Eileen Rossman

Mechanical Engineering Department
California Polytechnic State University, San Luis Obispo

Sponsor
Matthew Meadows
Summer Helmuth
Girl Scouts of the Central Coast
Camp Arnaz, Ventura, CA

June 9th, 2023

Statement of Disclaimer

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

Abstract

Team Weed Scouts has completed our work on a weed-cutting robot for the Girl Scouts of California's Central Coast. The final robot build provides a solid foundation that can be built and improved upon by future teams. We have completed the robot base and structure, including the chassis, drivetrain, and robot shell. We also completed manufacturing a weed storage bin and canvas cover for the robot. Additionally, we have built a weed scooper, the mechanism that cuts weeds and transports them into a storage compartment. The electronics and programming for remote control of the robot are also implemented. After some testing, we found that the robot had limited functionality. It was able to drive around with slight power issues but unfortunately, the weed scooper was not able to fully cut and transport the weeds. Despite these obstacles, we have created a semi-operable foundation for future teams to optimize, test, and debug. The next steps include adding a weed shredder and developing autonomous robot functionality along with weed identification. Upon project completion, the weed-cutting robot will help maintain the grounds at Camp Arnaz and serve as a source of engineering inspiration for Girl Scouts and other camp visitors.

Table of Contents

List of Figures

List of Tables

1 Introduction

This project entailed building a weed-cutting robot for use at the Girl Scouts of California's Central Coast (GSCCC) camp in Ventura, California. Two Mechanical Engineering senior project teams worked on this project: Team Weed and Team Scouts. This document represents the combined Final Design Report (FDR) for both teams. Our goal was unchanged since the Critical Design Review (CDR), apart from the removal of the shredding function. We aimed to create a turtle themed, remote-controlled robot that responded to controller commands and could move around the site and cut weeds when commanded. In brief, Team Weed's primary responsibilities were the drivetrain, chassis, battery, and shell while Team Scouts' focuses were the control system, lighting, and weed cutting and storage of the weeds.

The next phase of this project will be to build upon the existing design and to automate these processes and make the existing functionality autonomous. This report includes the full system design, which has been updated to include all the mechanical and electrical components within the robot. The report also includes sections for design verification, manufacturing and assembly, cost estimation, and our discussions and conclusions of the project.

2 Design Overview

2.1 Design Description

The design of the robot consists of several subsystems. This section will provide a brief overview of the robot subsystems, how they function, and how the user will interact with each. The full User Manual can be found in Appendix A. For a more detailed description of the design of all subsystems, see the Weed Robot Critical Design Report (CDR). The subsystems discussed in this section are labeled in Figure 1 below.

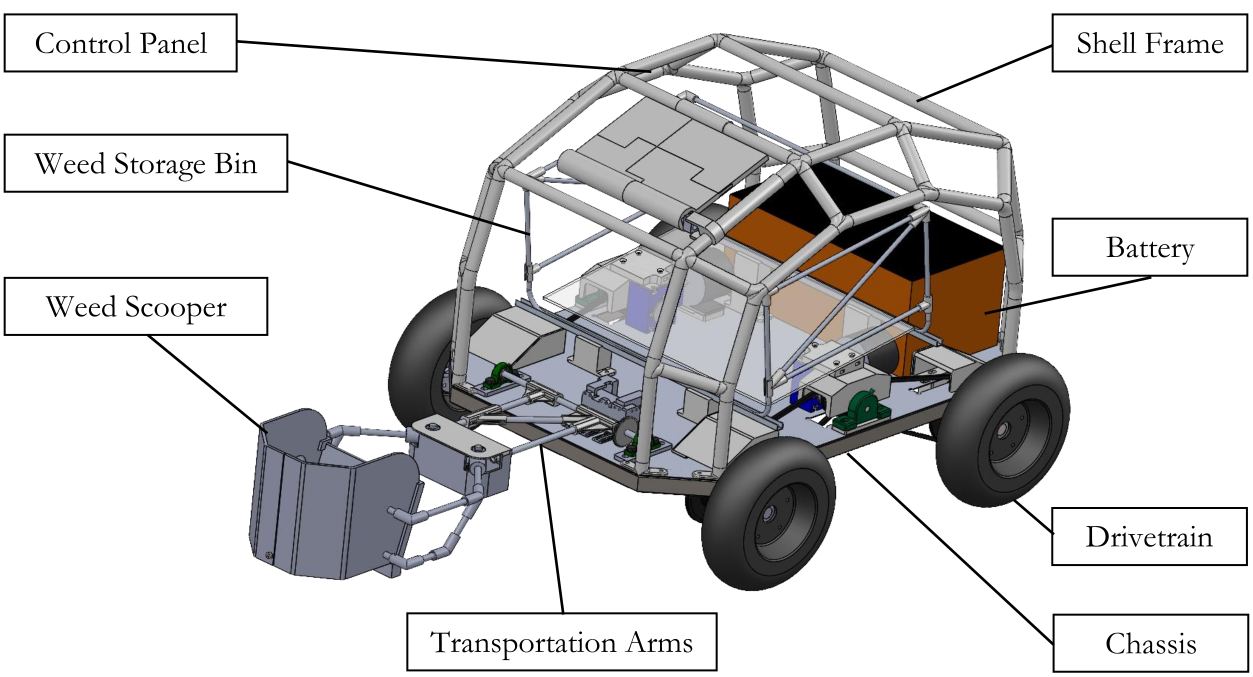


Figure 1. Complete CAD Assembly of the Robot

2.1.1 Shell Frame

The shell frame consists of custom 3D printed ABS fittings, ¾" diameter PVC pipe, and a canvas shell covering. The shell covering includes various flaps and cutouts to facilitate access to the robot's internal components. The robot's control panel is located on the right side and includes the Power, Reset, and Emergency Stop buttons. The flap on the right rear corner provides access to the battery connections for charging, which will be discussed in further detail in the section below. On the robot's left side, there is a large flap door for access to the storage bin.

In the event that a shell fitting, or any other 3D printed component, breaks and needs to be reprinted, SolidWorks and .stl files for all parts are included with this report. Instructions for printing are included in the User Manual.

2.1.2 Chassis & Drivetrain

The chassis and aluminum sheet are the main structural components that support all the other components. The chassis is made from 1.5" x 0.12" carbon steel square tube. The base support plate is manufactured from .080" 6061-T6 aluminum. All robot components are mounted to the chassis and base plate, including the bearings to support the drivetrain.

The drivetrain includes two independently controlled sides, which allow for robot turning. Each contains two loops of #40 standard roller chain, two axle sprockets, one double-single center sprocket, a planetary gearbox, and a drivetrain motor. Additionally, our team cast custom aluminum wheel hubs configured to our 13" diameter non-pneumatic wheelbarrow tires. The drivetrain has possible pinch points due to the roller chains, so in accordance with the Risk Assessment in Appendix B, we printed and applied warning labels to warn users about them.

2.1.3 Weed Storage Bin

The weed storage bin holds a volume of 2.5ft^3, or approximately 18 lbs. of dry weeds. Fresh weeds are expected to weigh more, which would meet our 20 lbs. capacity requirements. The bin has 3/8" aluminum rod frames which are supported by ABS fittings. Handles have been added to the side to provide support for the user. Magnets are attached on one side to secure the storage during operation. To empty the weed storage bin, the user will open the flap on the robot's left side, and pull on the handle to remove the bin along its rails.

2.1.4 Battery

The robot is powered by a 28.6V, 220 Ah, LiFePO4 battery mounted to the back of the robot. The battery sits on the aluminum plate and is held in place by the back chain guards and storage bin rail supports as well as a battery bracket. All motors and electronic components are powered off this battery. The battery can be charged using the included battery charger via a standard 120V power outlet. Like with the drivetrain, warning labels were placed on the battery warning users of the high voltage and to cut the power before starting maintenance on the robot.

2.1.5 Weed Scooper & Transportation Arm

The weed scooper is the primary system that interacts with the weeds with a blade at its base that cuts the weeds by closing with a torque provided by two servo motors. Its walls create a basket-like container that holds the weeds until they are transported into the weed storage. The scooper was designed to be easily 3D printed with ABS filament and assembled using super glue which would serve as a hands-on learning opportunity for the girl scouts. Pinch point and sharp blade warning labels were attached to the outside of the scooper to warn users.

The transportation arms were made using ¾" thick ½" diameter hollow aluminum tubing for low weight and easy machining. The arms are two parallel tubes connected diagonally for extra strength which connects the scooper and the chassis via the rotating shaft which was also made using the same aluminum tubing drilled and turned down as described in the manufacturing subsection (3.2). The transportation system is actuated by a servo motor with a spur gear reduction of 4:1 and is supported by two mounted pillow block bearings on either side of the rotating shaft. These arms lift the scooper upwards to transport the weeds in the scooper to storage. A warning label was placed onto the motor box warning users of the swing area since the arms will rotate up and down to transport the weeds to the storage.

2.1.6 Remote Control

The robot is remote controlled using a PlayStation DualShock 4 controller connected via Bluetooth. See the user manual in Appendix A for a visual mapping of the remote controls. The buttons mapped to a function are the left joystick, the home button, the triangle button, the circle button, and the cross button. The remaining buttons serve no functionality when pressed. The left joystick controls the robot's drivetrain and wheels. Pushing on the joystick in any direction will send a pulse-width modulation (PWM) signal to both motor drivers to move the motors. Vertically upwards causes the robot to drive straight forwards. Vertically downwards causes the robot to drive straight reverse. Since the robot is physically incapable of steering, it turns by turning one side of the wheels slower or in the opposite direction compared to the other. Pushing the joystick nearly horizontally either left or right will cause the entire robot to rotate its orientation left or right. The triangle button opens and closes the storage door, the circle button opens and closes the scooper, and the cross button lifts and lowers the scooper arms. At any point during the robot's operation, the home button can be pressed as a safety measure to disable PWM outputs, disable the enable pins on the motor drivers, and stop the execution of the code.

2.2 Design Changes Since CDR

Since CDR, Team Weeds made the decision to postpone the implementation of the weed shredder. After some discussion with our sponsor and our academic advisor, we decided that the shredder was not a high priority since the robot would still be able to complete most of its intended functions without the shredder. Included in the report deliverables is our completed design for a shredder, with intention for it to be implemented by a future senior project team or by our sponsors.

Team Scouts had a few minor design changes after the critical design review with our sponsor. We realized that creating strong weld joints on thin aluminum tubes was not possible with our welding ability. Instead, we came up with an alternative to join our transportation arms and the rotating shaft that would not involve TIG welding of the aluminum tubes. We designed and 3D printed T, K, and Y connectors from PLA as shown in Figure 2, with M4 holes drilled into the rotating shaft to fasten the connectors to the shaft. The transportation arms were then superglued to the connectors. The transportation arms were reduced to 11" and the diagonal arm was reduced to 7.9" in order for the entire scooper-transportation sub assembly to better interface with the shredder opening. In order to reduce the torque on the DOCYKE servo motor, we added a spur gear pair with a 4:1 reduction. Another idea was to add a cord in tension with carabiners, connecting the motor box and the shell frame. However, on testing this design we realized that the PVC frame is too weak to provide reaction forces strong enough to balance the tension, however, this idea can still be implemented in a future iteration when a load bearing element is added with the shredder onto which the cable can be hooked. A major addition to the motor box was a roller bearing in the lid that would support the L tube connectors along with 3D printed dowel pins. More detailed descriptions of these changes can be found in Section 3.

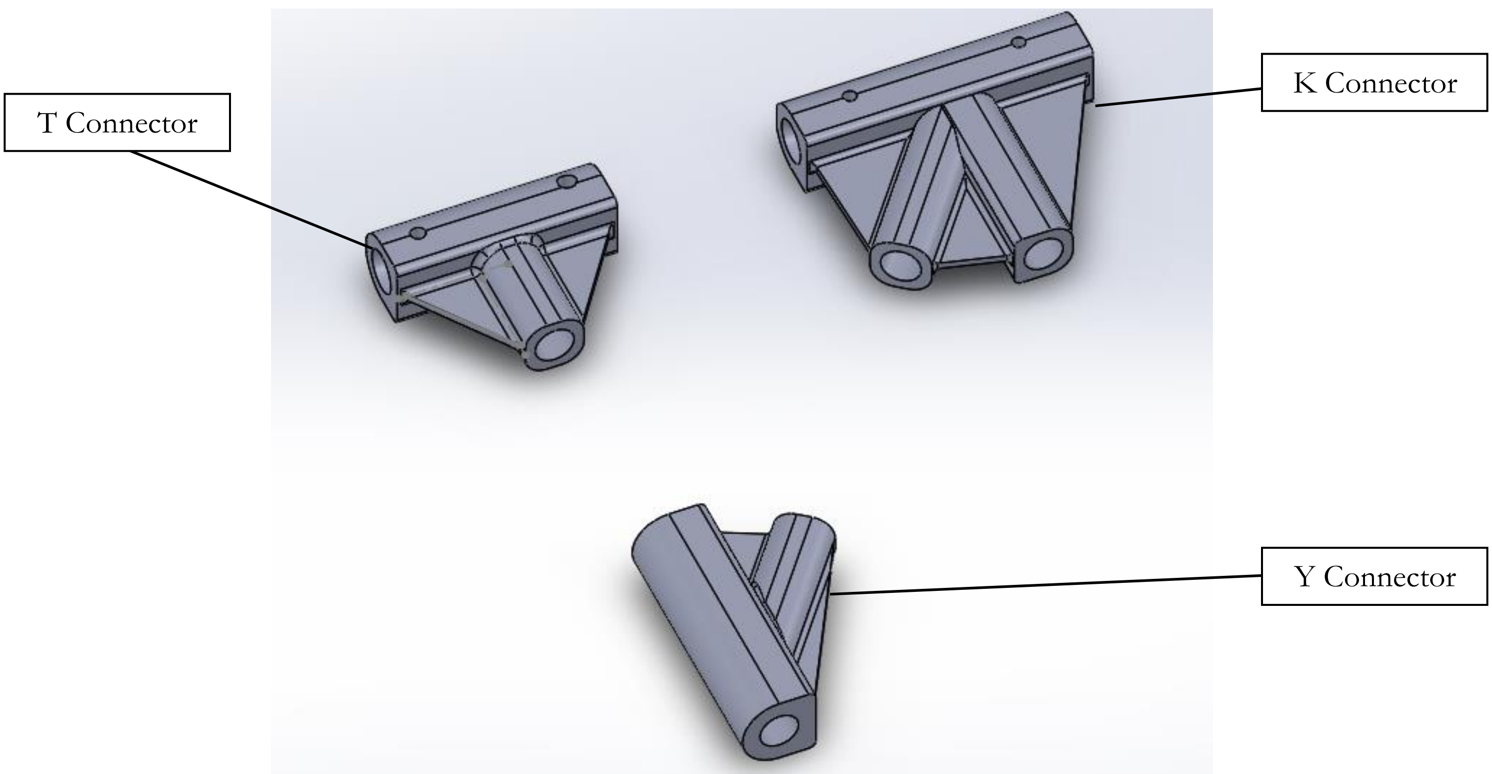


Figure 2. T, K, and Y Connectors

We also designed and implemented the system control panel, and the shredder door mechanism which involved connecting them to the shell frame using 3D printed mounts for the PVC as can be seen in Figure 3.

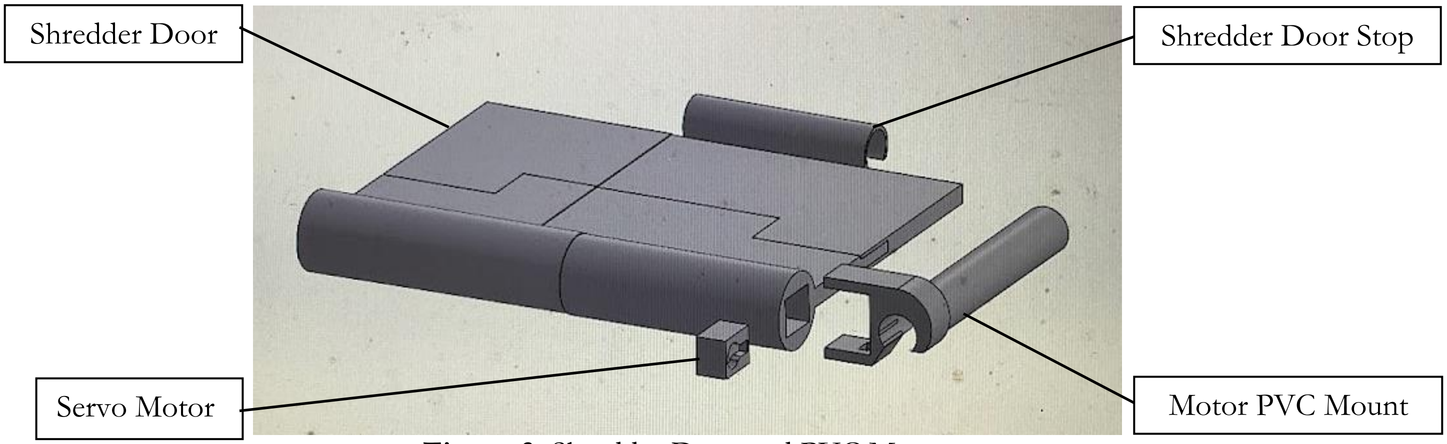


Figure 3. Shredder Door and PVC Mount

Another design change was made to the scooper arms with a secondary connection being added to improve the stability of the system along alignment guides were added to the back panel and the base of the scooper head. Lastly, an unforeseen issue arose with implementation of the DOCYKE positioning which caused the gears to stop meshing. To resolve this, we 3D printed brackets mounted to the aluminum base of the robot to support the motor on its side and prevent it from lifting off the chassis.

3 Implementation

This section outlines the steps taken to procure materials, manufacture custom parts, and assemble the final verification prototype.

3.1 Procurement

Most of the materials used in the verification prototype were purchased from various sellers on Amazon.com and other specialty vendors. Common building materials, such as PVC pipe and fasteners were purchased from local hardware stores like Home Depot and ACE Hardware. A full list of purchased materials and purchase sources is included in the final budget listing in Appendix C. An indented bill of materials (iBOM) can also be found within our drawing packet.

3.2 Manufacturing

The manufacturing process will be outlined in detail for each manufactured component. Team Weed's manufacturing included creation and integration of the wheels and wheel hubs, chassis, base plate, drivetrain, storage bin, shell frame, and canvas cover. Team Scouts manufactured the scooper head, motor box, scooper arms, and transportation arms.

3.2.1 Wheels and Wheel Hubs

The robot's wheel hubs are investment cast aluminum. The first step was to create a 3D printed ABS pattern which would eventually produce the cavity in the mold. We did this by modifying the wheel hub CAD model by adding a sprue section. The purpose of the sprue section is to reduce shrinkage and premature cooling in the part by providing an additional mass of molten metal above the cast part. This section was removed in post-processing. The hub pattern is shown in Figure 4a, with the sprue section identified. Next, we placed the pattern upside down inside a cast iron flask (shown in Figure 4b on top of a silicone mat and poured plaster around it. Once the plaster had hardened, we cycled the flask through a series of furnace temperatures to remove any remaining moisture from the plaster, burn the pattern out, and fully solidify the plaster. Figure 4c shows the plaster mold after the pattern has been burned out. The final furnace cycle heated the mold to about 500 °C so the molten aluminum at about 700 °C did not freeze when it hit the mold. We then removed the mold from the furnace and poured aluminum into the cavity, shown in Figure 4d. After the aluminum solidified and cooled, we removed our cast part from the plaster. The final cast part, before any post-processing, is shown in Figure 4e.

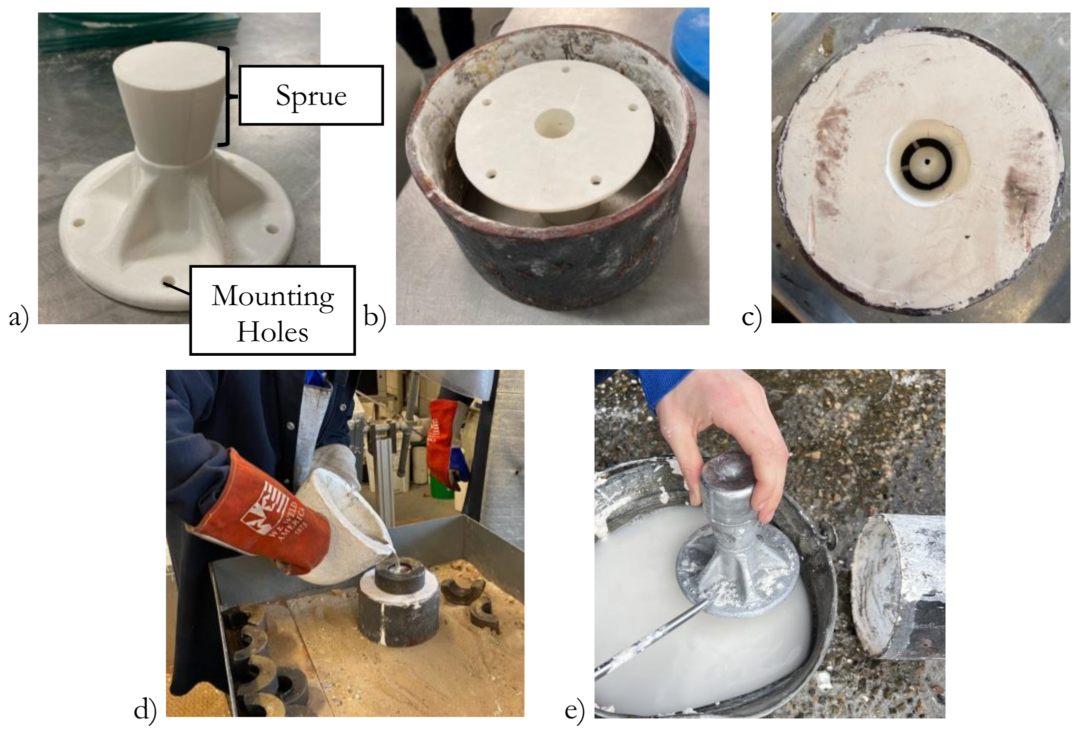


Figure 4. Manufacturing Process for Wheel Hubs

After the hubs had been cast, they required a few post-processing steps. First, we used a grinding wheel to ensure that the bottom surfaces of the hubs were flat where they would mate with the wheels. Next, the sprue sections were removed using a vertical band saw. The mounting holes, which had been undersized in the pattern, were drilled out using a 5/16" drill bit on the drill press. Due to shrinkage and imperfections in the cast, the center hole of the hubs was slightly smaller than the intended 1-inch diameter. We used a boring bar on the manual mill to slowly chip out material from the inside of the holes until the axle shafts fit smoothly inside. Finally, the set screw holes were drilled out and tapped with ¼"-28 threads. Figure 5 shows these processes.

Figure 5. Post-Processing of Wheel Hubs

3.2.2 Chassis and Base Plate

The main chassis of the robot was welded together by our sponsor. After we received the chassis, we did some grinding on the welds to create a flat surface on the top of the chassis where we would eventually mount an aluminum plate. After we finished grinding, we then hand sanded away the small patches of rust that had accumulated after welding. We then wiped down the surface of the chassis frame and spray painted it with two coats of green rust-preventing spray paint as seen in Figure 6.

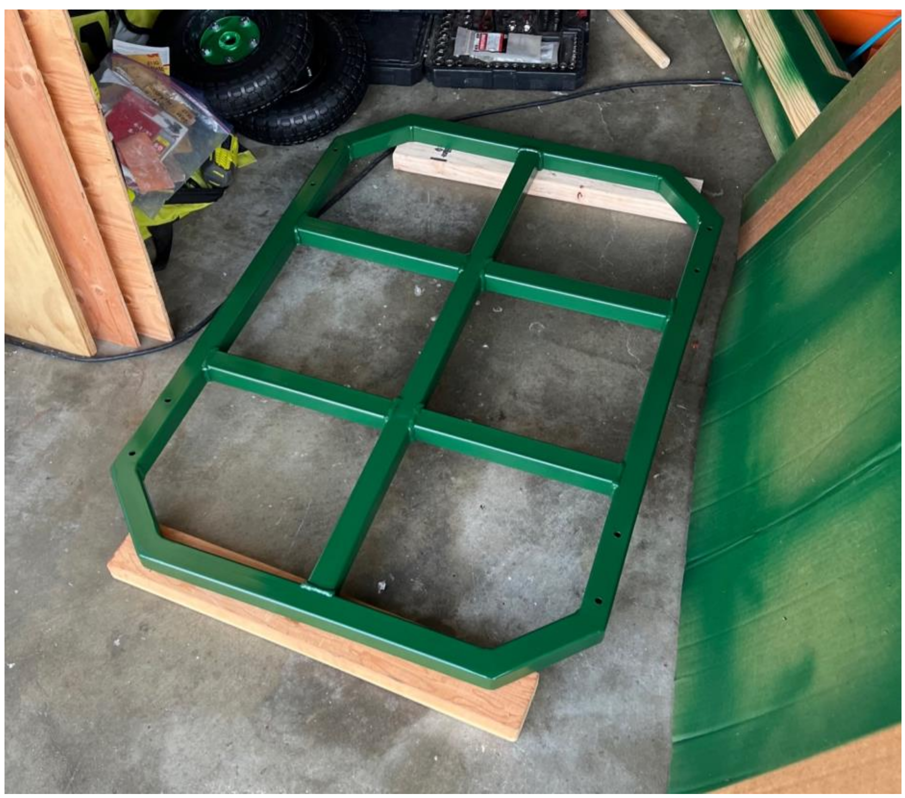


Figure 6. Spray Painted Chassis Frame

Further processing included fitting an aluminum plate to the top of the chassis and bolting them together. To allow the drivetrain chain to pass from the bottom to the top of the chassis, we needed to cut slots in the aluminum plate. We began by measuring and marking the cutouts with a Sharpie, then using a centerpunch to mark the corners of each cutout. We drilled pilot holes at each corner, then enlarged the holes with a drill bit large enough to allow our jigsaw blade to be inserted cleanly in the hole. Then, with the aluminum plate supported by wood blocks, we used a jigsaw to cut each slot out. Figure 7 shows the process of cutting the chain slots into aluminum.

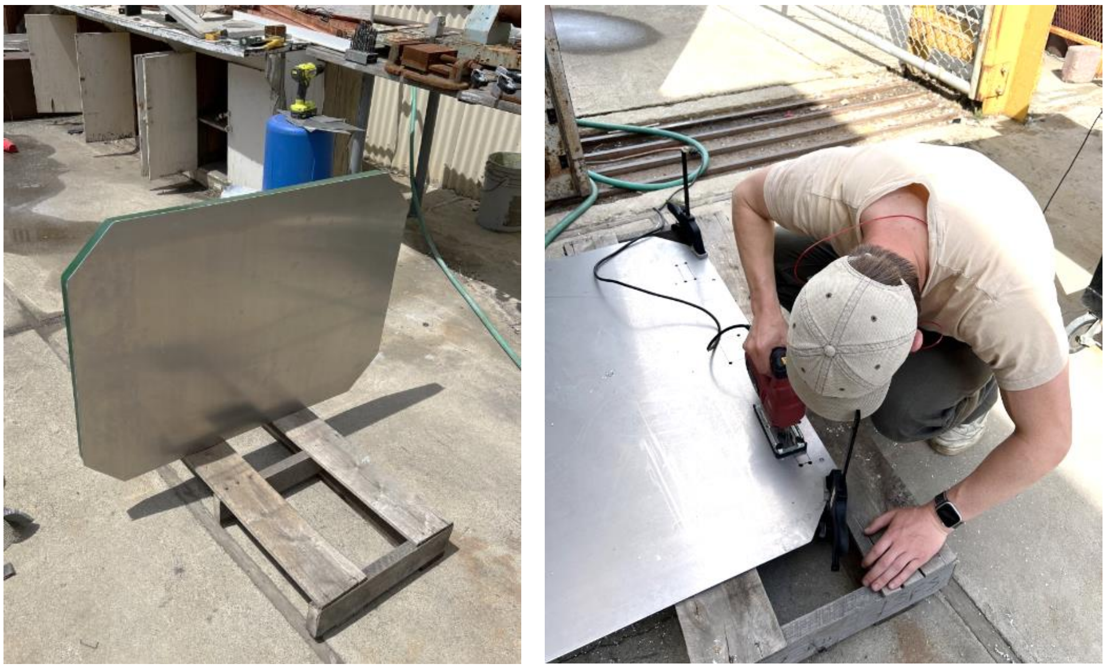


Figure 7. Processing of Chassis and Aluminum Plate

Throughout the assembly process, we drilled holes into the aluminum plate to mount the other drivetrain components including the motor, worm gearbox, bearings, storage rail mounts, PVC frame joints, and cutting arm bearings.

3.2.3 Drivetrain

A large portion of manufacturing for the drivetrain subassembly involved machining shafts on a lathe and manual mill. The drivetrain contains a total of eight shafts (four on each side) that required machining.

3.2.3.1 Motor – Gearbox Shaft

We cut the two 11mm diameter stock shafts to length using an abrasive chop saw. We placed each shaft in the lathe for turning down the coupling diameter. We then moved to the mill where we mounted the shafts in a collet block and cut out the keyway for the worm gearbox and the flat for the shaft coupling. Flats were also manufactured on the motor shafts to better fit into the coupler. These two shafts are shown in Figure 8.

Figure 8. Motor to Gearbox Shafts

3.2.3.2 Center Sprocket Shaft

We cut the two 5/8" stock shafts to length using an abrasive chop saw. We then placed each shaft in the lathe and turned down the diameters for the worm gearbox and then flipped the shafts around to turn down the diameters for the gearbox support bearings. After turning was completed on the lathe, the shafts were moved to the mill. We placed the shafts in a collet block to mill out the keyway for the worm gearbox. After that we milled out the keyway for the center sprockets. We used a machinist's jack to support the end opposite the end that was in the collet block since the center sprocket keyway requires a large portion of the shaft to stick out of the collet block. We then cut out the keyway for the center sprockets. The fixturing for machining this keyway is shown in Figure 9.

Figure 9. Machining Center Sprocket Keyway

3.2.3.3 Axle Shaft

Each of the four axle shafts began as a 12-inch long, 1-inch diameter carbon steel rod. We used an abrasive saw to cut the shafts to length. Since the shafts were slightly oversized, we used a lathe to reduce the diameter by about .0015" for the shafts to fit through the outside axle shaft bearings. When this process was complete, we flipped the shafts around and turned down the diameters to fit the wheel sprockets, and the inside axle shaft bearings. After we completed turning on the lathe for all 4 axle shafts, we moved them to the mill. We used a 3/16" endmill to mill out the keyway for the wheel sprockets and the flats for the wheel hub. Figure 10 shows parts of the turning and milling process for the axle shafts.

Figure 10. Turning Axle Shaft Diameters and Cutting Keyway Slots

3.2.3.4 3D Printed Motor and Bearing Spacer Brackets

On each side, the drivetrain contains 3 bearings and 1 motor that must be spaced off the top or bottom of the build plate to ensure that they align with the other components of the drivetrain (other bearings and the worm gearbox). We therefore needed 3D printed spacer brackets for the motor, center sprocket bearing, and inside axle shaft bearings. An example of one of these spacers can be seen in Figure 11.

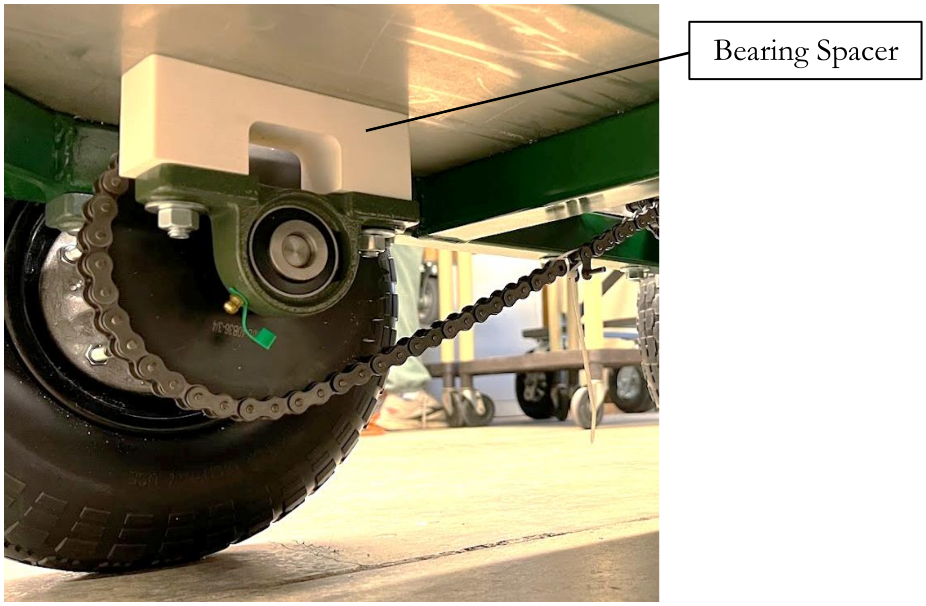


Figure 11. 3D Printed Bearing Spacer used to Align Components

3.2.3.5 Keys

We purchased a metric sized key set that contained several sizes of metric keys. The drivetrain assembly contains mostly standard unit system components except for the worm gearbox which has metric dimensions. The keys were only used on the shafts going in and out of the gearbox and to mount the sprockets to their shafts. The sprockets used standard unit 3/16" keys. We made the sprocket keys by modifying the 5mm keys that came in our metric key set. We ground down the 5mm keys until they had a 3/16" square cross section to fit the sprocket keyways.

3.2.4 Storage Bin

We began manufacturing the metal frame of the storage bin by cutting 3/8" diameter aluminum rod to the required lengths. We then used a pipe bender to bend two 90° angles in each of the two longest pieces. We also 3D printed a variety of custom fittings out of ABS plastic to connect the pieces of aluminum.

The storage bag was created from three rectangles of canvas – two for the short sides of the bin and one to wrap around the long sides and the bottom. The three pieces were sewn together, and channels were sewn into the top of the long sides for the frame pieces to fit through. Figure 12 shows the completed storage bin and bag.

Figure 12. Storage Frame Connectors and Storage Bin and Modifications

The storage bin slides in and out of the robot body on two pieces of ½” x ½” aluminum C-channel. Since the storage bin must slide over the drivetrain components in the center of the chassis, each piece of C-channel is supported by two 3D printed storage rail supports at the required height. We first cut the C-channel to length, then drilled two counter-sink holes where the supports would be screwed in. Fittings on the storage were later modified to add magnet inserts that would connect to the added buffer stop.

3.2.5 Shell Frame

To construct the shell frame, we started by cutting PVC stock from Home Depot to the specific lengths needed. We 3D printed custom PVC fittings out of ABS plastic to connect the segments of PVC at the necessary angles to create the curved frame. Figure 13 shows the 3D printing process and the resulting fittings and segments of PVC pipe.

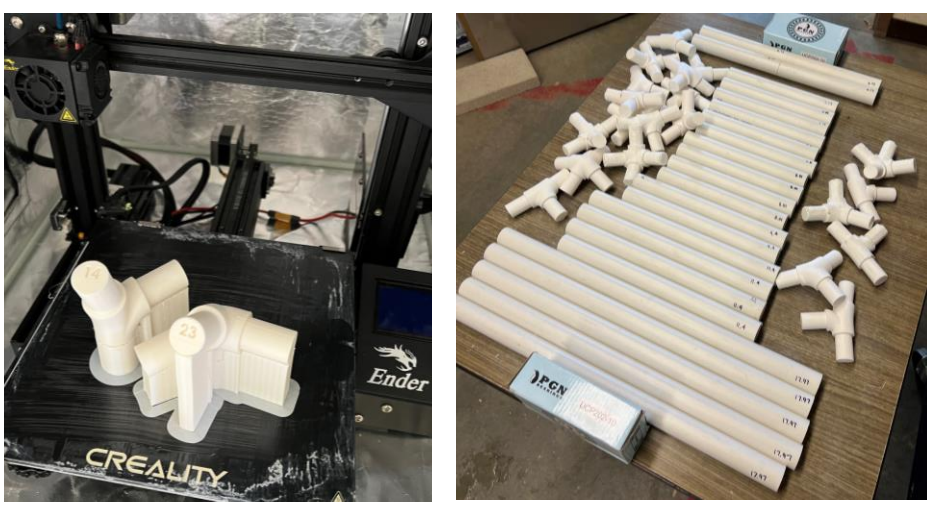


Figure 13. Creation of PVC Segments and 3D Printed Fittings

3.2.6 Canvas Cover

The canvas shell covering consists of five pieces sewn together – one center piece that stretches from front to back, two flat side pieces, and two pieces connecting the angles between the two others. A sewing pattern was developed based on the dimensions of the Shell Frame to cut the fabric to size. To keep the bottom edge of the canvas secured to the chassis, strip magnets were sewn into the hem of the cover. As component placement was finalized, flap doors were integrated into the canvas cover for easy access to the battery and storage bin, as well as cutouts for the shredder door, robot control panel, and headlight and taillight. These doors are also secured by magnets sewn into channels in the canvas. A slit was also created in the front to allow Team Scout's transportation arm to move up and down freely. The processes involved in creating the canvas cover and the finished cover are shown in Figure 14.

Figure 14. Canvas Shell Cover

3.2.7 Scooper Head

The scooper head consist of four pieces: female front and rear piece, and male front and rear piece. The parts were designed in SolidWorks and each piece is printed with ABS with 40% infill, and each half is separated by front and rear piece due to the size of the component. The two parts of the half are conjoined by puzzle-like teeth that were modeled to ensure better rigidity after the parts are connected as shown in Figure 15.

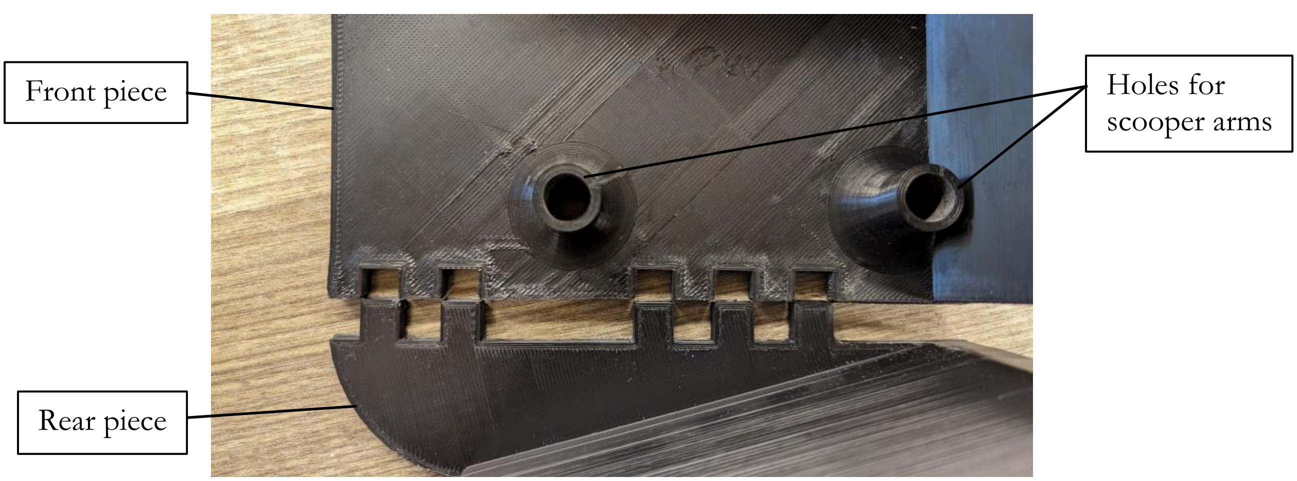


Figure 15. Scooper Pieces

Alignment fixtures were 3D printed to provide support on the lining up of the closing, and they are attached to the back wall of the scooper and latched onto the side walls of the part as shown in Figure 16. When the scooper closes, these fixtures interface first and guide the blade and the rest of the scooper halves to properly mesh together.

Figure 16. Alignment Fixtures on the Scooper

3.2.8 Scooper Arms

The scooper arms were joined using three types of connectors shown in Figure 17. The two connectors at the rigid supports at the scooper head are at an angle of 90°. The connector at the bottom arm piece is angled at 126.87°. The last three-way connector joining the top and bottom arm pieces to the motor box system has critical angles of 53.13° and 135°. The arm connectors were created by 3D printing.

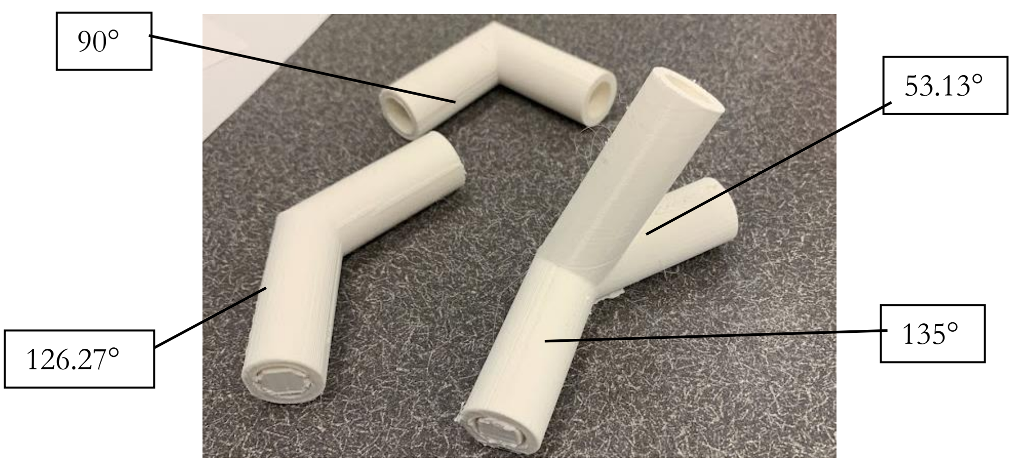


Figure 17. Scooper arm connectors

The scooper arm pieces were cut from the ½” diameter aluminum tubing to the dimensions labeled in Figure 18 using an abrasive chop saw followed by sanding it down to length on a metal belt sander.

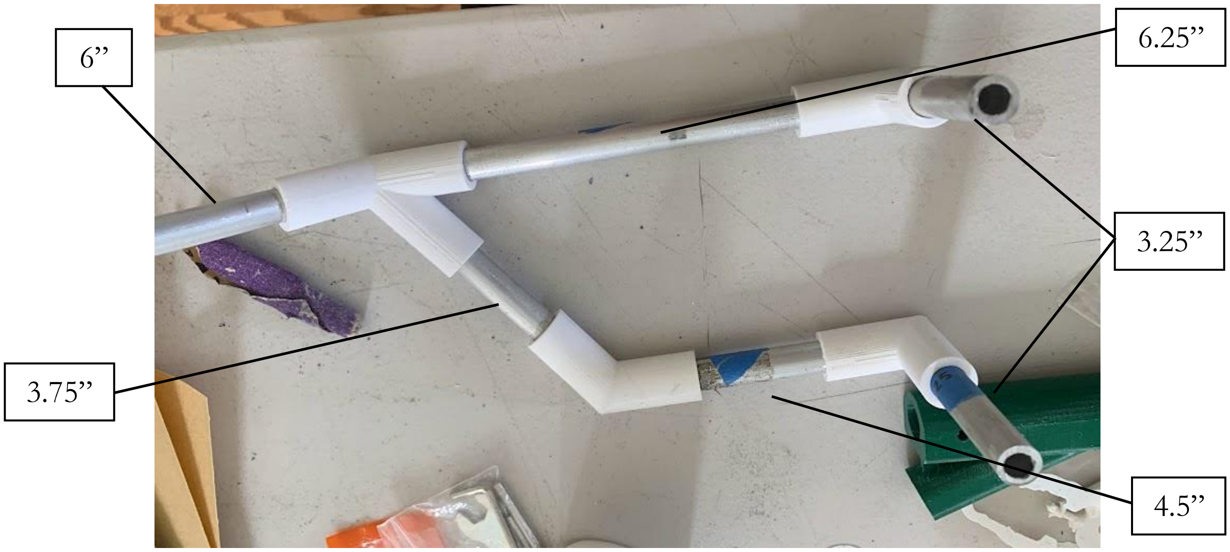


Figure 18. Assembled Scooper Arms

3.2.9 Motor Box

Manufacturing of the motor box and its lid involved designing the parts in SolidWorks and 3D printing them in ABS at Mustang 60 shop. The box printed with features to hold the two scooper motors and attach to the transportation arms. After printing, 3D printed supports needed to be removed from the box and lid, and heat set inserts were installed using a soldering iron. Some sanding needed to be done as well to ensure proper fitment of the box and the lid. The aforementioned features include the dowel pins and the L-tube connectors which were printed in a similar fashion. The motor box and its components are shown in Figure 19.

Figure 19. Motor Box with Slide-On Lid Containing Holes for Roller Bearings and Dowel Pins

3.2.10 Transportation Arms and Rotating Shaft

The transportation arms and the rotating shaft were cut from the ½" aluminum tubing to the lengths indicated in Figure 20 using an abrasive chop saw followed by sanding on a metal belt sander. To allow the scooper servo motor wires to be threaded through and not be exposed to sharp edges, a deburr tool was used on the inner diameter.

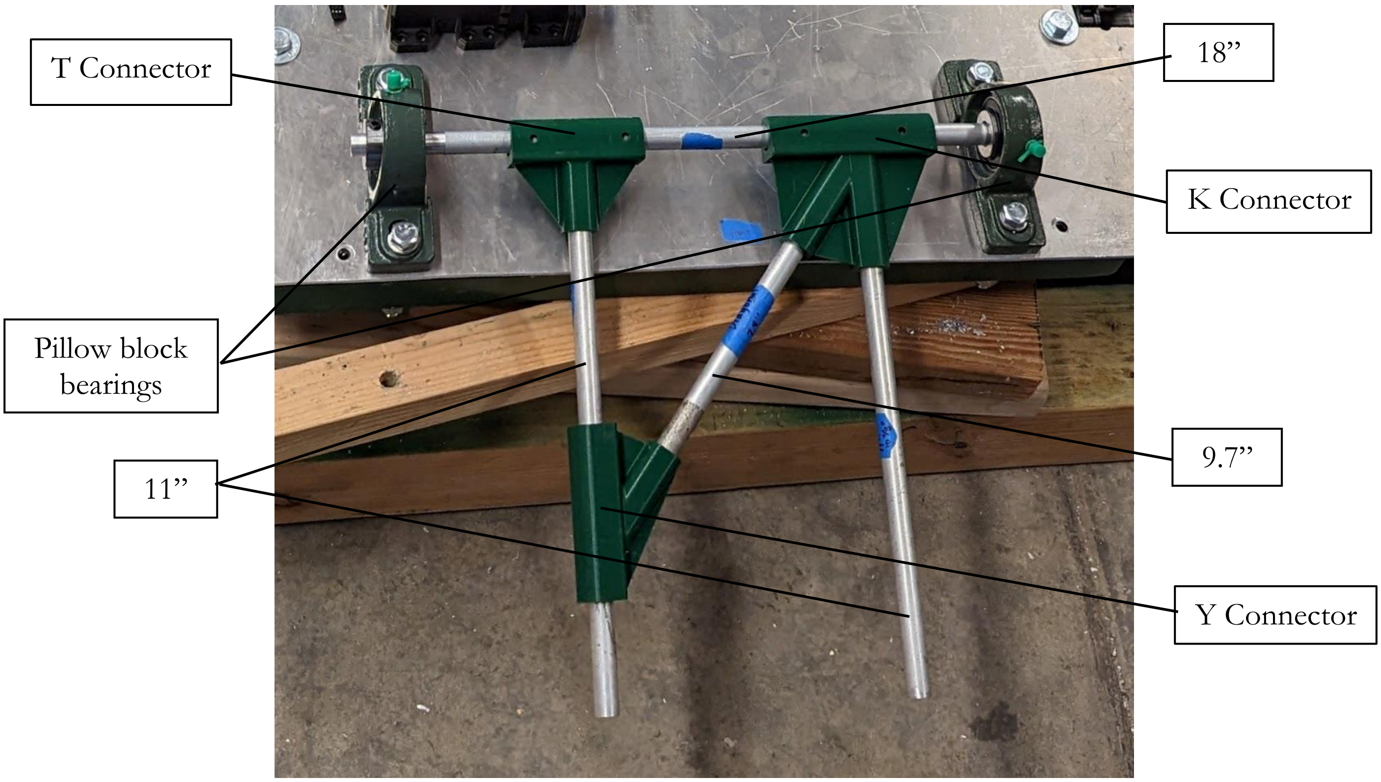


Figure 20. Transportation Sub-System Assembly

Further manufacturing of the rotating of the was conducted in the form of drilling M4 holes using a drill press and clamping the part on a vise as shown in Figure 21. The shaft was then rotated 90° and 3/8" holes were drilled to allow the wires to be threaded through. In order to interface with the mounted pillow block bearings, the shaft was turned on a lathe by taking 0.0035" off the diameter through a length of 1.5" from the ends using the tailstock with a live center attachment to support the part as pictured in Figure 21.

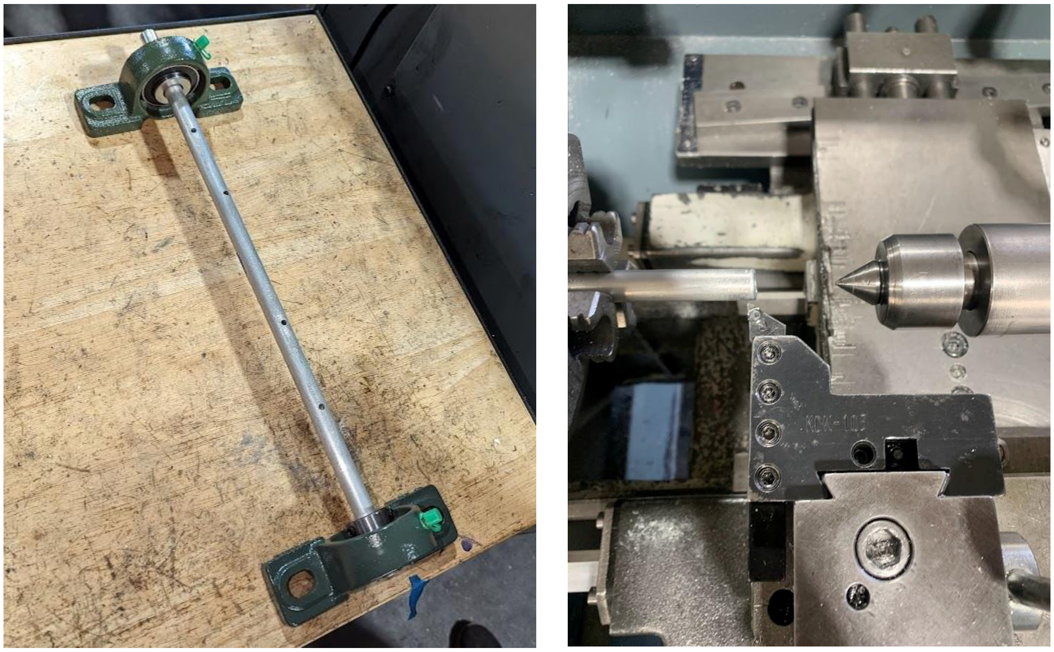


Figure 21. M4 Drilled Holes and Turning the Shaft

K, Y, and T connectors were 3D printed and used to join the shaft to the transportation arms. To secure the connections, we glued the rod and connectors together and used M4 screws and nuts to further secure the rotating shaft.

For installation of the mounted bearings, DOCYKE servo motor, and motor mounts, holes were drilled into the chassis and aluminum plate. The position of the motor was dependent on meshing the gears on the motor shaft and rotating aluminum shaft. Also, bearing mounts were 3D printed to lift the bearings and avoid a collision between the big gear and aluminum plate.

3.2.11 Manufacturing Challenges & Changes

While some steps of the manufacturing process were more time consuming than anticipated, the build phase of this project went smoothly for most of our parts. Apart from minor hiccups (such as small manufacturing mistakes, delays on parts arriving, equipment issues, etc.), both teams had a couple of challenges. These challenges exclude equipment inaccessibility due to limited or damaged equipment.

One of Team Weed's challenges was in post-processing our cast aluminum wheel hubs. We realized the inner bore diameter in the cast part was not exactly 1" and needed to be drilled to size to fit the corresponding shaft. To do this, we initially tried to drill it out using a 1" drill bit on a drill press. However, there was significant runout in the drill bit as it rotated, and the resulting hole would be larger than 1". This resulted in too much play between the shaft and the wheel hub. To solve this problem, we transitioned to the manual mills and used a boring bar to slowly chip material out of the holes. This allowed us to achieve a more exact, concentric 1" inner bore diameter to fit snugly with the shaft.

Another challenge was experienced with machining our keyways. After passing the learning curve of using the mill, we still had some difficulties positioning the shafts property and squaring the part with the mill. We tried different locking mechanisms to keep the shafts in the proper position, but even after professional help, our cut shifted from its planned path. The shaft had to be processed again on the lathe to remove burs from the damaged section and allow for a different key slot to be created. This is shown in Figure 22, using a dial indicator to ensure the shaft was concentric with the lathe's axis of rotation.

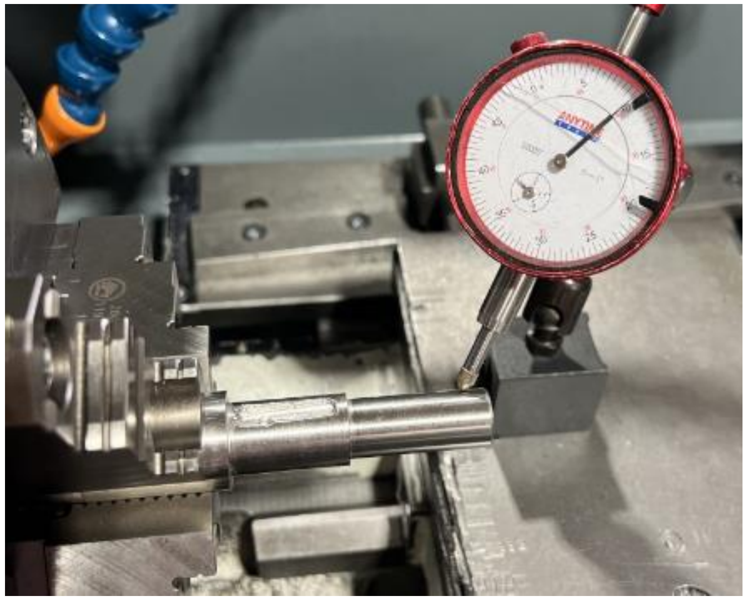


Figure 22. Fixing Misaligned Key Shaft Finish

Another change we made was our selection for the storage bin rail. Initially, we had selected a much heavier and wider rail than necessary. When we went to pick it up, we realized that a slimmer rail option that was in stock would be more suitable for our needs.

Team Scouts faced manufacturing and design challenges in the scooper sub assembly for the most part. The most prominent concern and challenge was the stability of our systems, considering that the scooper-transportation system behaves like a cantilever beam. Furthermore, the scooper subsystem itself acts as a cantilever beam with respect to the motor box. Firstly, there were great difficulties in stabilizing the motor connection which we were optimizing since our structural prototype. The final system consisted of the L-tube connector being supported by the screw for the motor attachment along with washers that acted as spacers since we had to get longer screws than what came with the motors themselves. In order to ensure that the moment on the L-tubes did not affect their spatial alignment, we printed dowel pins that would align the L-tubes with the motor box lid vertically.

The next difficulty was with mounting the transportation system onto the aluminum plate. The size of the DOCYKE motor was 0.1" larger than the provided space which caused an interference that would involve machining of the rotating shaft or the motor to overcome it. This would have been too dangerous in terms of functionality which made us motor on its side. This introduced further complications in terms of stability of the motor. The moment caused by the scooper-transportation system was too large for the minimal surface area and mounting holes on the servo to overcome and would get lifted when it tried to lift the scooper head. In order to fasten it down, we created custom brackets and used the ones that came with the motor itself. The next issue was with the fact that the base plate was made of aluminum that would deflect once the motor was secured. In order to minimize the deflection, we cut sheet metal squares and bolted them underneath the area facing maximum stress.

The last issue we faced involved the meshing of the gear teeth. While the brackets and the sheet metal prevented any more deflection, the moment still caused the gear teeth to mesh poorly which was worsened by the fact that the rotating shaft was made of hollow aluminum tubing which indented due to the set screws and lost its circularity. At the end, however, we did assemble the system with utmost care such that the lifting mechanism worked as intended.

3.3 Assembly

The robot required significant assembly time after all parts were manufactured individually. Some small manufacturing steps such as drilling mounting holes in the aluminum plate took place alongside the assembly process to ensure that all shafts and other components lined up correctly.

Figure 23 shows the fully assembled robot. The canvas shell was not included to allow for visibility of internal components.

Figure 23. Final Robot Assembly Without Canvas Covering

3.3.1 Drivetrain Assembly

We first located and drilled holes to mount the outside wheel bearings. We drilled these holes through the frame and the aluminum plate and then fastened the bearings to the chassis.

The following applies to each side of the drivetrain. We placed the motor shaft coupler on the motor and then placed the shaft from the motor to the gearbox in the coupler. Then we lightly tightened the set screws on the coupler. We then inserted the motor shaft along with a 4mm key into the input of the worm gearbox. Then we inserted the center shaft along with a 4mm key into the output of the gearbox. We then placed the two center sprockets onto the center shaft along with two 3/16" keys. We then inserted the center shaft bearing onto the other end of the center shaft. After creating this sub assembly, we placed it on the aluminum plate with the 3D printed spacers below the motor and center shaft bearing. We then marked and drilled holes and fastened the motor, gearbox, and center shaft bearing to the build plate. We left the set screws on the sprockets loose for now.

Then we placed all four axle shafts into the previously mounted outside axle bearings. We fastened the wheels to the wheel hubs then placed the hubs on the shafts and tightened the set screws. After that we placed the wheel sprockets along with a key on the inside of the axle shaft making sure that the spacer on the sprocket was facing the outside of the robot. Then we placed the inside axle shaft bearing along with the spacer on the shaft. We then drilled holes in the aluminum plate for the inside axle shaft bearings and fastened them in place.

Next, we measured out the necessary length of chain and used the half link to attach each of the four chain loops from each axle shaft sprocket to each center sprocket. At this point we connected as few as possible chain links in each chain loop. The chain still needed to be tense. Each bearing mount in our assembly has slotted holes. We loosened the fasteners on each axle bearing and held the axle shaft and wheel towards one end of the robot by hand to make the chain as tight as possible. With the shaft and wheel in this position, we then tightened the axle shaft bearing fasteners to hold the axle shaft in place. This process resulted in the chain being properly tense.

3.3.2 Shell Frame

After all ABS fittings were finished printing, the PVC pieces and fittings were connected to build the final frame. The original plan was to use an ABS/PVC glue to glue the frame together, but we realized during assembly that this step would be unnecessary because the press fits were already secure enough. Many of the fittings needed to be sanded down in order to fit inside the PVC pipe. Figure 24 shows this process in progress.

Figure 24. Frame Shape Manufacturing in Progress

Once the frame was put together, the last step was to bolt it to the chassis on top of the base plate. We drilled the necessary holes in the chassis and aluminum plate and secured the frame with 1/4" bolts. The final frame on the robot is shown in Figure 25.

Figure 25. Completed Shell Frame

3.3.3 Scooper

The complete scooper assembly involved putting together all of the components of the motor box and assembling the scooper halves. For the motor box assembly, first the servo motors were screwed in with 8 total M3 screws. Then the L-shaped connectors were attached to the motor attachments. A washer was put in the top hole of the connector, and then the connectors were screwed onto the motor. The motors wires were strung through the transportation arms holders before the lid was slid on. The bearings were installed into the lid with superglue. Finally, the lid was put on the box, and the cylinders to align the bearings and L connectors could be installed. Figure 26 shows the interior of the motor box.

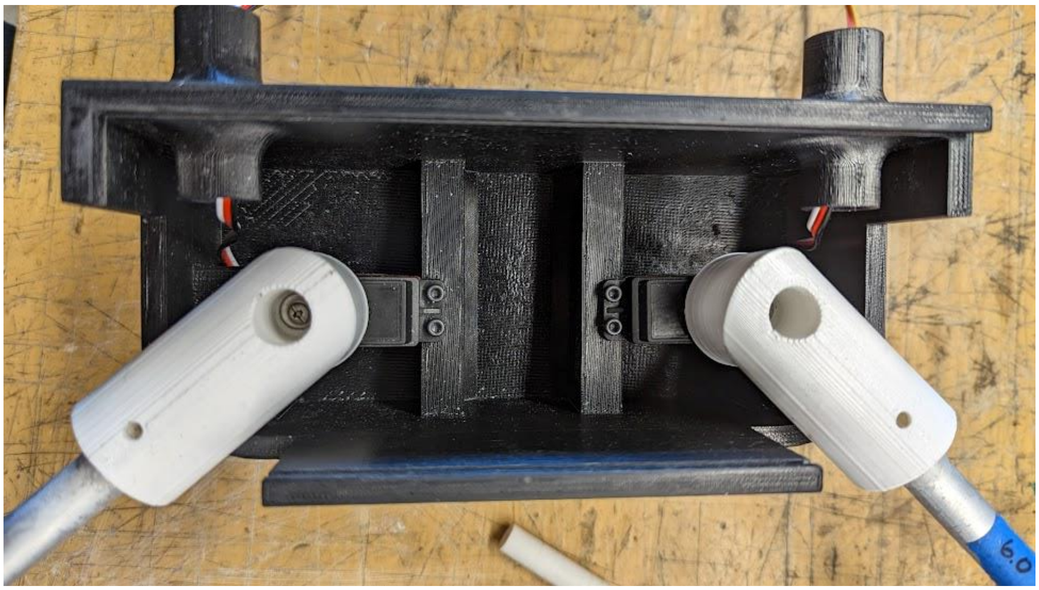


Figure 26. Inside the Motor Box

To put together the 3D printed scooper pieces, we applied superglue. The scooper arms pieces and connectors were also attached with superglue. To prevent inward rotation of the scooper, we also installed a #6 screw through the 6in and 3-pronged connector, shown in Figure 27. The completed scooper arms were installed into the side of the scooper halves and secured with superglue. Two clamps were fastened to the male scooper half with M8 bolts and nuts, and the blade was inserted into the slot and clamped into place. The alignment guides for the scooper pieces were also superglued onto the scooper.

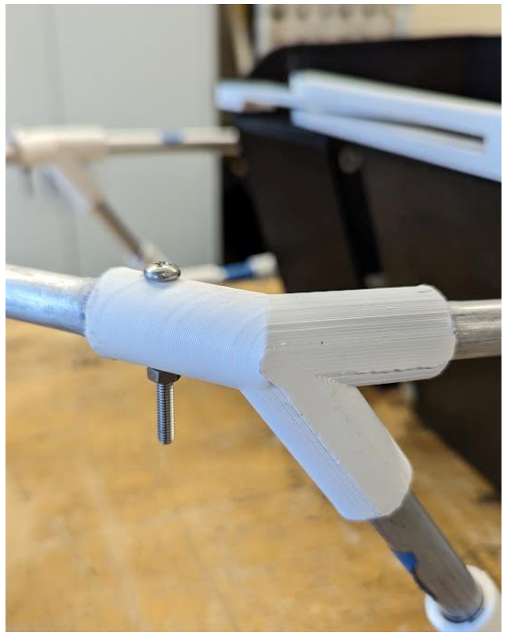


Figure 27. #6 Screw through the 3-Prong Connector

The scooper subassembly was attached to the motor box by inserting the 6in scooper arm piece into the L-shaped connector. Then, that connection was secured with superglue. Finally, the scooper assembly was finished with coats of green spray paint and UV protection spray. Figure 28 shows the scooper assembly before and after the spraying process.

Figure 28. Scooper Assembly Before and After Spraying

3.3.4 Transportation System

Assembly of the aluminum tubes and transportation arm connectors is detailed in the Manufacturing section. The motor was first installed onto the aluminum plate with #6 screws, washers, and nuts. Then, the two 3D printed DOCYKE mounts were screwed onto the motor and chassis. The small gear was tightened on the motor shaft with set screws. Before mounting the bearings onto the chassis, the rotating shaft of the transportation arms needed to be inserted into the bearings. Once the bearing spacers and bearings were placed on the aluminum plate and aligned with the drilled holes, the 3/8" bolts, washers, and nuts are used to secure the transportation system so that the two gears are properly meshed. The set screws on the bearings also needed to be tightened onto the shaft. Figure 29 shows us installing the screws for the transportation system.

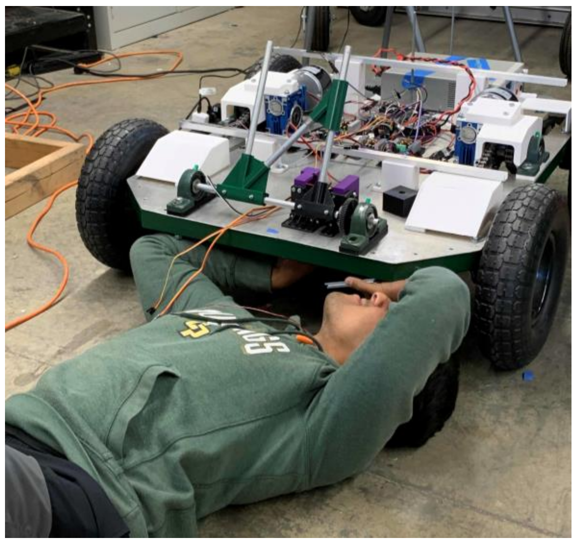


Figure 29. Installing the Transportation System onto the Aluminum Frame

3.4 Electronics

Most of the electronics in the robot are contained in an acrylic box located in the center of the chassis. Figure 30 shows the electrical components within the electronics box. The major components include the 25.6V battery, NVIDIA Jetson Nano microcomputer, PCA9685 pulse-width modulation (PWM) driver, DOCYKE servo motor, two BTS7960 DC motor drivers, two LM2596 DC buck converters, three Anmbest constant current buck converters, two 24V DC motors, three DS3225 servo motors, one 470 μF capacitor, and one emergency stop button. All of these components were purchased from Amazon. The robot was controlled remotely through Bluetooth, through a PlayStation DualShock 4 controller connected to a Bluetooth on the Jetson Nano via a Bluetooth adapter. The Bluetooth adapter was also purchased from Amazon, while the PS4 controller was bought on eBay. A fully detailed purchase list can be found in Appendix C.

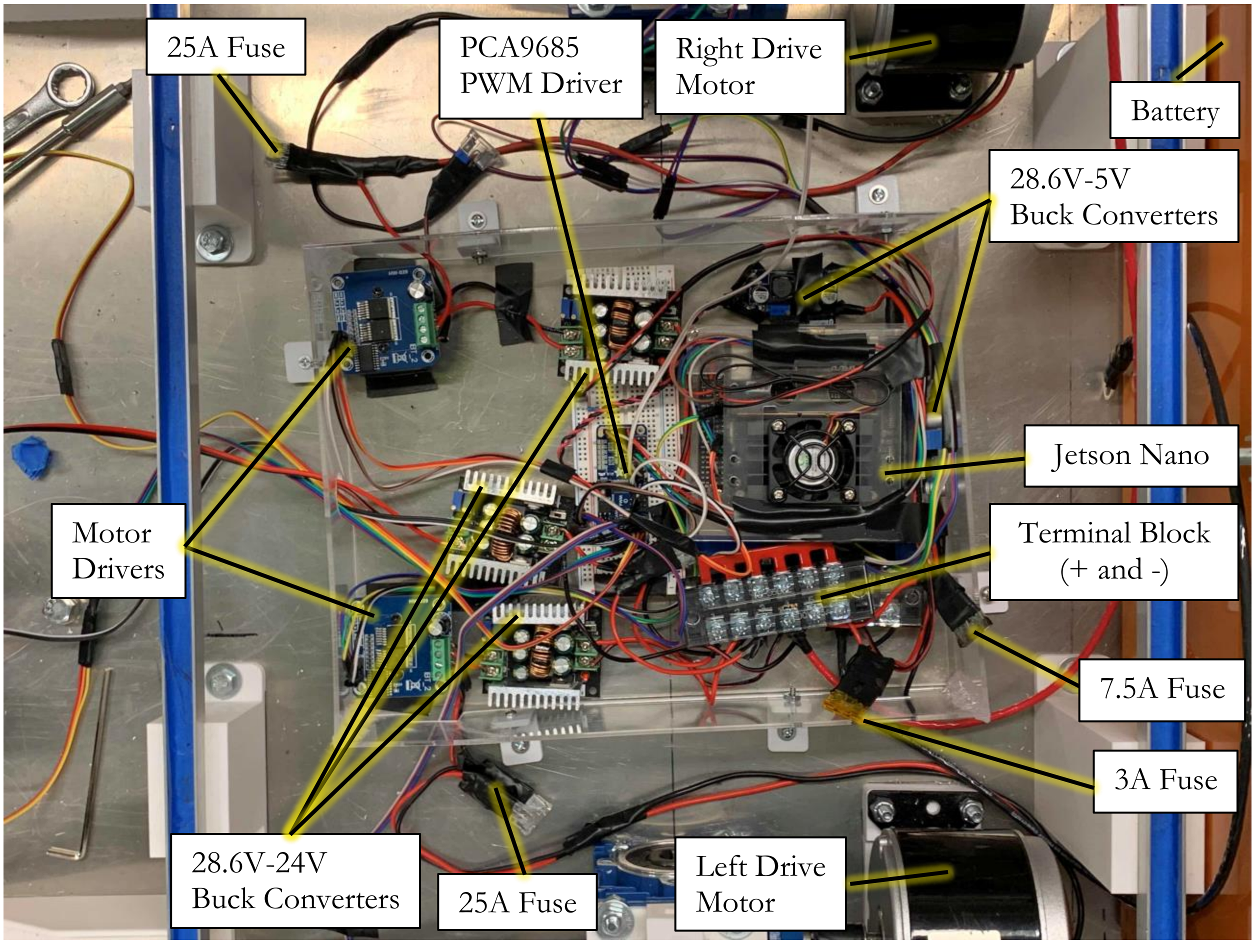


Figure 30. Electronics Box

The battery has a nominal voltage of 28.6V and is positioned in the back of the robot and connected to two terminal blocks, one positive and one negative. This voltage is too high for the electrical components to handle, so we stepped down the output using DC buck converters. With the Anmbest voltage converters, we could provide 24V to the DOCYKE servo and the two DC motor drivers to power the 24V rated motors with a maximum current output of 20A. To provide the correct amount of power to the Jetson Nano and the PWM driver, two DC buck converters were inserted in between each to step down the voltage from 24V to the 5V required by both components with a maximum output of 3A. The motor drivers allow the DC motors to go forward and reverse with an H-bridge by reversing the polarity of current flowing to the motors. The driver is connected to the battery and motors for power, accepts PWM inputs from the PWM driver, and is connected to a 5V logic level from the Jetson Nano. To protect the robot from overcurrent, fuses were also placed in between each component.

Since the Jetson Nano has only had 2 dedicated PWM pins on its 40-pin expansion header and we needed to control 5 motors, the PWM driver with its maximum of 16 output pins was implemented to provide the necessary amount of PWM. The four servo motors each require one PWM input, while the DC motors require two each (one for forward drive and one for reverse drive). The driver also provides power to the three DS3225 servo motors since they require 5V. A 470 µF capacitor is attached to the driver to smooth out the abrupt jumps in current caused by servo movement. The amount of PWM is controlled by the Jetson Nano, which communicates to the driver through the

Inter-Integrated Circuit Protocol (I2C) by the data (SDA) and clock (SCL) lines. A full wiring diagram detailing the power and logic connections between each major component as well as the pins on the Jetson Nano for the electrical subsystem is included in Appendix D. Figure 31 shows the lifting servo, storage door servo, and scooper servos, which lie outside the electronics box. The lifting servo controls the lifting and lowering of the scooper arms, the scooper servos control the opening and closing of the scooper, and the storage door servo controls the opening and closing of the storage door.

Figure 31. Placement of Servo Motors

An emergency stop button was implemented between the battery and the terminal block, which is shown in the User Manual in Appendix A. When the button is not pressed, the circuit is closed and allows current to flow through the entire circuit. When the button is pressed, the circuit is open and prevents current from flowing to the terminal block. The emergency stop button completely cuts off power to the robot until the button is disengaged. In addition to standard wiring, the input/output connections for the DC buck converters have all been soldered.

We had some issues integrating the electronics system with the rest of the robot. We didn't realize that the battery, which was rated and advertised to be 25.6V, was actually outputting 28.6V. This was an issue because our motor drivers were rated for 27V so they ended up burning out. The Anmbest buck converters were implemented into the circuit to step the battery voltage down to a more manageable 24V. Other issues came from accidentally swapping the positive and negative leads of certain components, which caused the connection to short out and send a large amount of current the components, burning them out. Once we realized the connection was wrong, we fixed it and bought new components to replace them.

3.5 Software

All the code for the weed robot was written using Python 3.7.5. This code referenced and used existing libraries for interfacing with the Adafruit PCA9685 driver [1] and PS4 controller [2]. Before beginning to code the robot, these libraries and other system dependencies were installed onto Python using pip.

The code initially sets up the PCA9685 driver to communicate with the Jetson Nano and establishes the channels it needs to send PWM signals to each motor. Then, several GPIO pins on the Nano were set up as output pins to enable the enable pins for the motor drivers. Initial positions were set for each servo motor to keep the scooper closed, the storage door closed, and the transportation arms lowered. The code then tries to find a connected controller. If found, the status LED on the control panel is set to turn on to tell the user that the robot is ready to operate. If no controller is detected, the code will continuously loop until a controller is paired. In this state, the code waits for an input from the controller and performs the corresponding action. Appendix G contains the finite-state machine showing all each individual state in the code. All actions were programmed using a state approach, meaning that each function was attached to a different state. There were nine states: waiting for controller input, forward drive, reverse drive, turn left, turn right, open/close scooper, lift/lower scooper arms, open/close storage door, and stop. A fully commented script of the code is included in Appendix H along with a link to a GitHub page with the documentation for reference.

OpenCV, a real-time computer vision library, was also installed onto the Jetson Nano through Python in anticipation of a future CPE/CS team to take up the task of improving and implementing weed identification. While it was no longer within our scope to implement identification into the robot, some preliminary work was done training an image detection model to detect certain types of weeds with the help of the Jetson Inference library [3]. Future implementation of this feature can either build upon this model or use the pre-installed OpenCV library to pursue another object detection method.

4 Design Verification

This section verifies the design choices made to fulfill our engineering specifications and shows the results of testing to determine which specifications were met and which were not.

4.1 Specifications

Table 2 shows the engineering specifications that the robot is expected to meet, as well as how each was completed.

Table 1. Design Specifications

	Specification	Plan for Completion
1.	Operating Time per Charge: **3 hr.**	We have done analysis to estimate the average power usage of the robot while it is running and selected the battery size based on the 3-hour estimate.
2.	Weight of Weeds Stored: **20 lbs**	We have done testing and analysis to determine the volume of a container to store 20lb of shredded yellow star thistles and sized the storage bin accordingly.
3.	Exposure Temperature: **20 °F - 110 °F**	We have verified that all materials we have selected are rated to be exposed to this temperature change.
4.	Height of Obstacle to be Driven Over: **± 2 in**	We utilized a simple clearance test to determine the robot's capabilities to drive over at height of at least 2 inches.

Table 1. Design Specifications cont.

	Specification	Plan for Completion
5.	Time for Full Charge: **6 hours**	We have selected a 24V, 200 Amp-hour battery that can be charged in under this time using a standard 120V outlet.
6.	Max Speed: **2 MPH**	We will design the drivetrain gear ratio for the maximum speed based on motor speed.
7.	Accelerates to Max Speed in: **5 Seconds**	We have selected two 24V, 250W drivetrain motors based on the robot's weight to allow the robot to accelerate to 2 MPH in 5 seconds or less. Calculations for this specification are found in Appendix C4.
8.	Max Chassis Dimensions: **3 x 4 ft**	We have limited our chassis frame to these dimensions that will fit into the back of a standard truck, modifying raw material to meet requirements.
9.	Robot Lifespan: **5 years**	We will do obstacle endurance analysis and design for infinite life for all mechanical parts out of an abundance of caution but some non-mechanical such as the battery will have a more finite life.
10.	Robot Displays **"low battery" and "weed storage full" indicators.**	These indicators will be designed based on sensors implemented into the storage container and electrical system.
11.	% of 3D Printable Parts: **20%**	Shell frame fittings, safety features, and other components will be 3D printed whenever possible, for a total of about 39% of parts.
12.	Weed Elimination Rate: **60%**	We will do rough estimates of the weed extraction mechanism with cutting force and strength of plant matter. Physical testing will occur to verify estimates and analyze capabilities.
13.	Ratio of Weeds Cut to Weeds Stored: **>80%**	We will measure the weight of weeds in the storage and the weight of the weeds that fall out of the scooper to ensure proper ratio.
14.	Remote Control Range: **30 ft**	We will measure the farthest distance in which the remote controller can still communicate with the robot.
15.	Total Weight of Weeds Extracted in a Day: **20 lb**	We will measure the weight of the weeds when the storage is at full capacity and also see how fast the robot can collect weeds.
16.	Maximum Cost: **$6000**	Management of budget and proper allocation of funds. Total cost estimate for the project is found in Appendix B.

Table 1. Design Specifications cont.

	Specification	Plan for Completion
17.	Typical Maintenance Time: **10 min**	Physical maintenances and test run to ensure easy maintenance and optimal times.
18.	Nighttime Robot Visibility: **20 ft**	Nighttime testing with surveying different personnel's ability to see the robot and measuring the distance to contrast.

4.2 Testing and Results

To verify whether our design met all the engineering specifications, we designed test procedures that correspond to our desired functions and conducted them after the verification prototype was built. Appendix E summarizes the Design Verification Plan and Appendix F lists all the test procedures. Table 2 summarizes each test procedure and its results:

Table 2. Testing Results

Test	Description	Criteria	Results
Operating Time	This test is to verify the amount of time that the robot can operate on a single charge.	3 hours minimum	Incomplete
Waterproofing	This test involves sprinkling the robot with water to simulate light rainfall and check for adequate waterproofing.	Operable in light rain	Incomplete
Motor Parameterization	This test is to parameterize our drive motors, specifically to determine the motor constant k_m and stall torque T_{stall}.	Determine motor constant k_m and stall torque T_{stall}.	k_m = 250 RPM/V T_{stall} = 8.5 Nm
Time for Full charge	This test checks the amount of time it takes for the battery to fully charge.	6 hours maximum	Incomplete
Obstacle Stability	This test is to determine how well the robot can move around uneven terrain. It evaluates how well the robot can drive over obstacles of different heights.	2 inch obstacle minimum	Max obstacle height of 1.5 in. More testing recommended.

Table 2. Testing Results contd.

Test	Description	Criteria	Results
Weed Elimination Rate	The test is to determine the percentage of weeds cut out of the number of cuts attempted. The test will determine the cutting efficiency.	60% minimum	Incomplete
Weed Collection Efficiency	The test is to determine the ratio of weeds cut to weeds stored.	80% minimum	Incomplete, 0 weeds stored
Remote Control Range	This test determines the farthest remote control range distance.	30 feet minimum	Control Range = 160.1ft
Nighttime Visibility	This test measures the visibility of the robot at night.	300 feet minimum	Visibility distance = 370.2 ft
Maintenance Time	This test measures the time it takes to carry out normal maintenance of the robot (charging the battery and replacing the blade).	5 minutes maximum	Maintenance time = 1 min 57 sec
Transportation Arm Operation	This test ensures that the DOCYKE servo is strong enough to lift the scooper assembly and measures the lifting sequence.	8-15 second rotation time	Average total rotation time = 8.59s
Multiple Motor Control	This test validates that each motor functions with its respective remote control input.	Pass/Fail for each motor	100% pass rate
Scooper Alignment	This test measures the alignment of the two scooper halves, both horizontally and vertically.	0.125 in maximum	Average blade in slot distance = 0.0125 in Average scooper base distance = 0.324 in

The following sections describe each test and briefly discuss the procedure and results. Refer to Appendix F for full descriptions of the tests, data collected, and outcomes.

4.2.1 Operating Time

The Operating Time test required the drive motors to function over the entire design voltage range, up to 24V. This is important for the robot to receive maximum power and be able to overcome obstacles, traverse uneven ground, etc.

Unfortunately, a motor driver issue prevented the completion of the time to full charge and operation time tests. On the day we attempted to complete these tests, the max output we were getting from the motor drivers was 7V. The drive motors are 24 volt motors and need an input of 24V to operate at max power. We were able to manually drive the robot around (as depicted in Figure 32) and discharge the battery to some extent however these test results cannot be used to conclude anything about the operation time of the robot since the motors were consuming much less energy than they would have been had they been getting 24V.

Figure 32. Operating Time Test

4.2.2 Waterproofing

Team Weeds was unable to complete the Waterproofing test. While planned to be encased by the time of senior project expo and project turnover, the electronics were still exposed throughout the duration of the testing time. This was partially due to the issues outlined above, which required significant troubleshooting. Subjecting the robot to a water test became unrealistic because of this.

4.2.3 Motor Parameterization

The purchased drive motors came with almost no specifications other than voltage and current draw. Therefore, it was necessary to perform a motor parameterization to get motor characteristics such as stall torque. The motors' no-load speed was recorded for various DC voltage inputs and motor parameters were extracted from this data. Table 3 shows relevant motor Parameters.

Table 3. Drive Motor Parameters

Parameter	Value
Stall Torque (at the motor)	0.848 Nm
Torque at Max Power	0.424 Nm
Stall Current	22.26 A
No Load Speed	629.9 rad/s
Motor/Speed Constant	250.63 RPM/V

Figure 33 shows the motor torque for a given voltage to the motor.

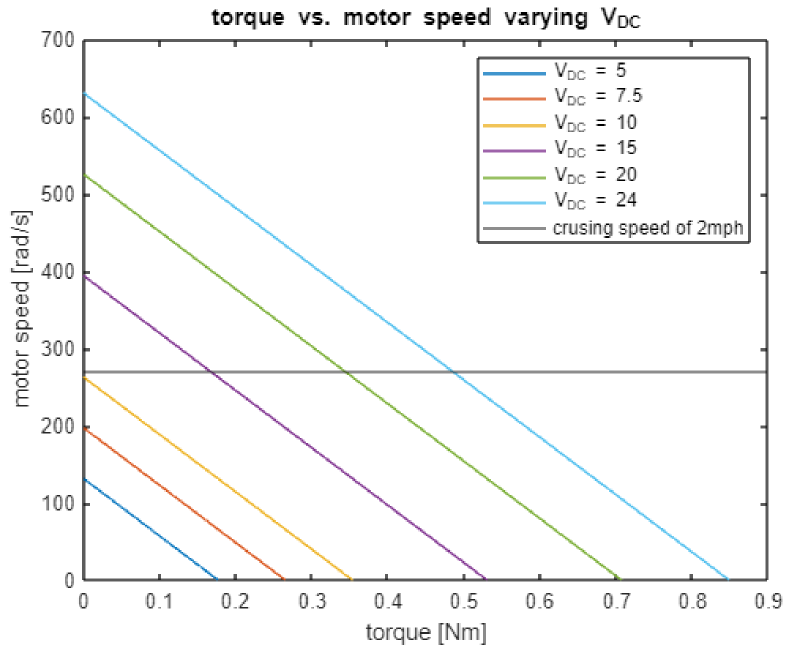


Figure 33. Motor Torque vs. Speed for various Voltage Inputs

4.2.4 Time for Full Charge

Because of the issues discussed for the Operating Time test, the battery was not able to be fully discharged in a reasonable timeframe. As a result, we were unable to complete the Time for Full Charge test.

4.2.5 Obstacle Stability

For this test, the drive motors were connected directly to a 24V power supply and driven over obstacles of increasing size until the robot got stuck. The largest obstacle that the robot was able to drive over was 1.5 inches tall, which does not meet the minimum height of 2 inches specified by the test. Figure 34 shows the robot about to traverse a 1.5 inch tall rock. Since the power supply has a built-in power limiter, the motors were significantly underpowered during execution of this test and the robot was therefore unable to traverse an obstacle larger than 1.5 inches in height. For this reason, further testing is recommended for the Obstacle Stability test.

Figure 34. Obstacle Test

4.2.6 Weed Elimination Rate

The weed elimination test was incomplete due to the blade being incapable of cutting weeds. The scooper damaged the weed tested, but results proved the scooper ineffective in cutting and collecting the weeds. This could have been due to the blade itself not being sharp enough to shear the weed, or due to flaws in our design.

4.2.7 Weed Collection Efficiency

The weed collection efficiency test was incomplete as well due to the scooper being incapable of cutting weeds and thus collecting them.

4.2.8 Remote Control Range

This test was done by having a testing the motor response at intervals of 25 ft until the motor ceased to respond and the remote control disconnected from the Jetson Nano. The test was conducted multiple times to obtain the average value of 160.1 ft which exceeded the criteria of 30 ft. Figure 35 shows the test being conducted in the Bonderson High Bay courtyard.

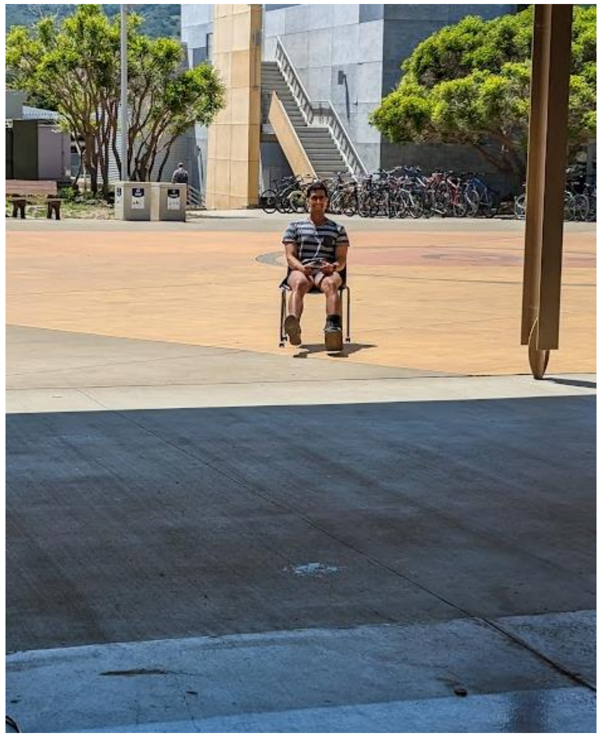


Figure 35. Remote Control Range Test

4.2.9 Nighttime Visibility

The Nighttime Visibility test was conducted by having a person hold the forward and reverse lights and increasingly walk away from a stationary observer until the person was not visible with the light anymore. The distance was recorded, and the test was repeated several times to obtain an average of 370.2 ft which surpassed the 300 ft minimum criteria. Figure 36 shows the visibility of the forward light in the center of the image from a distance of 314 ft.

Figure 36. Nighttime Visibility Test

4.2.10 Maintenance Time

The maintenance time was tested within our work room. Replacing and sharpening the blade of the scooper and plugging in the battery were the primary objectives. The maintenance time was found to be 59 seconds when replacing the blade and was timed to be 2 minutes 57 seconds, both of which adhere to the criteria of 5 minutes. Our systems thus were proven to have a quick maintaining process. Figure 37 shows a snippet of the maintenance process.

Figure 37. Maintenance Test Procedure

4.2.11 Transportation Arm Operation

Using the remote controller, we lifted the transportation system and measured the RPM of the lifting process, which passed our desired range of duration to lift to its maximum set position. Figure 38 shows the scooper-transportation system in its elevated state before the weeds get dropped to the shredder. In the future, this parameter can be optimized for a faster and smoother lifting operation due to the fact that as of this report, the all the motors have undefined initialization processes.

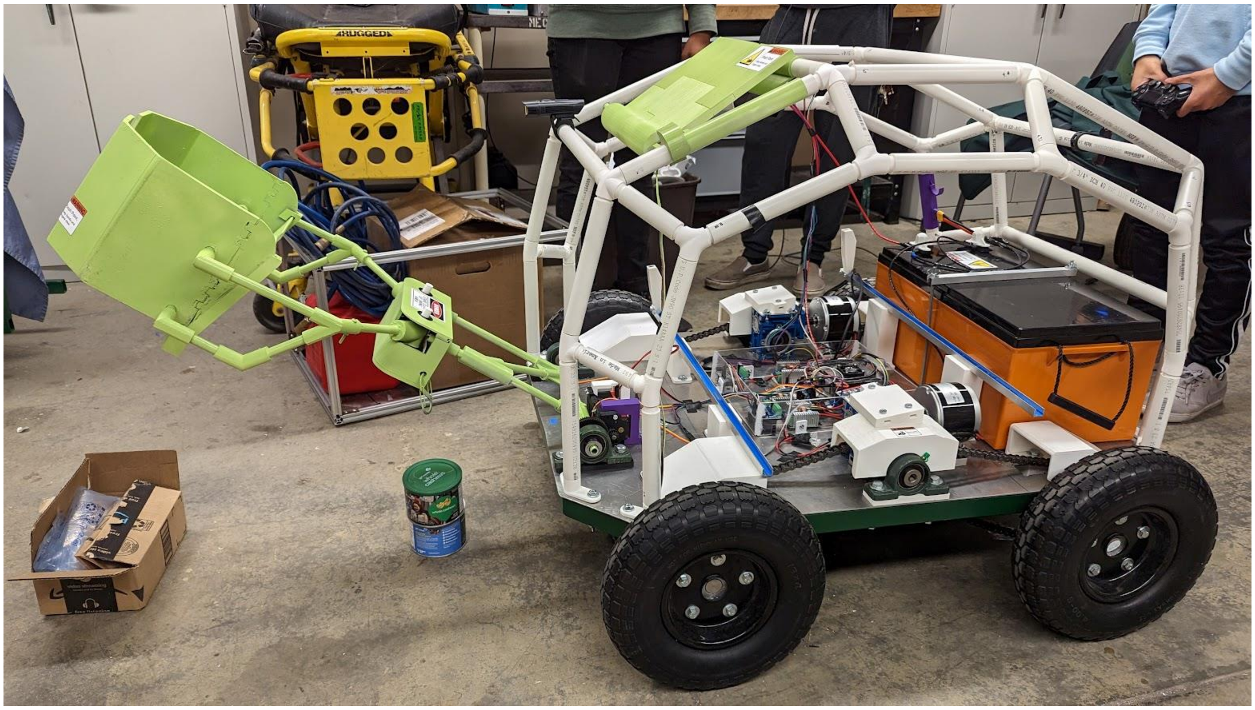


Figure 38. The Scooper-Transportation System in its Elevated State

4.2.12 Multiple Motor Control

We conducted this test iteratively as we wrote the code for easy debugging. All the motors responded to their respective button presses as expected at the end of the testing phase.

4.2.13 Scooper Alignment

We tested the alignment of the scooper by measuring the misalignment, if any, at two locations: blade height in the slot and horizontal distance between the two scooper halves. Although our acceptance criteria for vertical displacement was 0.125”, the scooper had difficulties with closing at an average displacement of 0.0125”. This difficulty can be attributed not only to the vertical misalignment but also the horizontal gap which we measured to be an average 0.324”. Figure 39 depicts the measurement process for the alignment test.

Figure 39. Measuring Front Scooper Alignment

For future testing, the continuation of the cutting and collecting of the weeds are of the highest priority. In addition, characterizing the motors for optimum performance with the code should be conducted.

5 Discussion & Recommendations

This section provides commentary about some of the challenges and learnings that came out of this project. It also lists some of the changes that we would make if we were to design the

5.1 Discussion

This project was a valuable learning experience for our team. Some of the main challenges we faced had to do with component integration and combining the separate work of each team. These challenges helped teach us the importance of consistent testing and cross-team communication. Our recommendations and next steps involve some troubleshooting and debugging of current components, as well as further expanding robot functionality. This is further detailed in the following section.

5.2 Recommendations and Next Steps

Automation

This project will be handed off to a Computer Engineering (CPE) senior project team at Cal Poly in the coming year. The CPE team will work to implement several systems necessary for autonomous operation, including geofencing, path planning, and computer vision for weed identification.

Shredder

To fully complete the mechanical aspect of the robot, design of a shredder mechanism must also be implemented. This component is necessary for the robot to carry a substantial amount of weeds. The shredder mechanism should be mounted inside the robot behind the shredder door and the exit of the shredder should deposit shredded weeds into the storage bin.

We partially designed a shredder for this project but were unable to complete or manufacture this design. We would recommend a shredder that has similar characteristics to a plastic recycling shredder. We recommend using steel water jet cut steel shredder blades. Although using steel blades will result in a heavy shredder, it will also ensure that the blades do not dull quickly and increase the longevity of the robot. Although some shredders/chippers that are often used to shred plant materials spin at high rpm, these shredders often have different blade designs and operate with gas powered engines. We think that the lower rpm plastic recycling shredder design is preferable to avoid vibrations from high rpm. In addition, we recommend the future implementation of a weed shredder to increase the weight capacity of weeds stored.

Drivetrain

If design changes were to be made to the drivetrain, we would recommend adding a neutral setting to the drivetrain. Currently the only way to roll the robot without driving the motors is to disconnect and remove the chain. This can be done by disconnecting a master link on each drivetrain chain. We think that a shaft disconnect between each center shaft and the output of the worm gearbox would be the best location for a neutral mechanism to be installed. This may require a new center shaft design and may require that the worm gearbox and motor be shifted towards the center of the chassis.

Weed Elimination Mechanism

Since the weed scooper mechanism was unable to eliminate weeds, we have some suggestions to make the scooper more effective. First, the blade that we chose was too dull to cut the weeds, so we would recommend getting sharper and easier-to-replace blades, like X-acto blades. Additionally, cutting might be more effective if there were blades attached to both scooper halves. With two blades, the weed could be sheared like how scissors work. In order to integrate the second blade, redesign of the scooper halves would have to be done to include clamps in the female scooper half and a blade slot in the male scooper half.

Transportation Arms

For the transportation arm subsystem, we still had issues regarding the gear ratio and stability. The current gear ratio is too big, and the rotating arms do not lift the full 90 degree path to the shredder door. In order to get the correct gear ratio of about 3, the future team should install a 45 or 48 tooth Module 1 gear on the rotating shaft. Since this gear will be smaller than the existing one, a future team may need to reposition the DOCYKE motor closer to the rotating shaft.

Another issue we had involved the stability of the transportation arms in its "off" state, as the DOCYKE motor was only able to lift the scooper and transportation assemblies when it was on. One suggestion we have to improve stability is to install additional support in the form of bungee cords that connect from the motor box to the shredder frame that will be implemented in the future.

Under the situation for more drastic changes, the scooper-transportation system can be refined to be more stable and provide a stronger shearing force. Furthermore, the system mounting could be made easier such that it can be used with different attachments such as a weed whacker or a weed pulling system. Another major redesign could be to the drivetrain to make it a better steering system and for it to be able to traverse higher slope grades.

6 Conclusion

The design, manufacturing, and building accomplished this year by Team Weed Scouts has resulted in tremendous progress on the weed robot. With the combined efforts of both teams, we completed manufacturing and implementing the body, cutting mechanism, electronics, and software of the robot. However, as discussed in the previous section, we had some issues with the functionality of our subassemblies. Due to poor early design choices such as only having one cutting surface or selecting a dull blade, the scooper was not able to fully cut weeds as intended. This also prevented us from testing weed transportation. Another aspect of the robot that we didn't achieve is creating and implementing the shredder. We underestimated the amount of time it would take to design and build a shredder from scratch that would be suitable for our needs. Ultimately, we think that most of the issues could have been avoided or reduced if we had started testing our designs earlier so that we knew what worked and what didn't. This would give more time for redesigning problematic components and subassemblies. The work outlined in this report expresses the design decisions made and procedures followed for manufacturing and building. Future work that builds upon this work will move the robot closer to its goal of autonomously extracting weeds throughout Camp Arnaz.

References

[1] Adafruit, Adafruit_CircuitPython_PCA9685, (2023), GitHub repository, https://github.com/adafruit/Adafruit_CircuitPython_PCA9685

[2] Tom Oinn, approxeng.input, (2021), GitHub repository, https://github.com/ApproxEng/approxeng.input

[3] Dustin Franklin, jetson-inference, (2023), GitHub repository, https://github.com/dusty-nv/jetson-inference

Appendices

Appendix A – User Manual

This user's manual includes instructions for product use and important safety information. Read this section entirely including all safety warnings and cautions before using the product.

Operational Guides
The following instructions refer to the robot's control panel:

Turning the robot on:

1. Make sure the emergency stop button is disengaged by rotating it clockwise. Press the power button. The robot is on when a blue circle light is shown on the power button as shown below.

2. Press the button on the PS4 controller to connect it to the robot. The controller is fully paired when the light at the top is solid blue and not flashing. A button map of the controller's buttons and respective functions are included at the end of this user manual.

3. Wait for the status LED to light up before operating the robot. During this time, the robot may move a little after initializing its motors.

Turning the robot off/resetting:

1. Press the button on the PS4 controller to stop the robot and disable the motors.
2. To power off just the robot's computer, press and hold the power button for about 7 seconds until the blue light disappears. To reset the computer, quickly press and release the reset button and the blue light should disappear briefly before showing up again.
3. To cut power from the battery, press the emergency stop button. This should ideally be done after the above two steps.

Engaging and disengaging the emergency stop:

1. Press the button on the PS4 controller at any moment during the operation of the robot to instantly stop the motors and stop the robot from running. In this state, the robot and motors are still connected to power.

OR

Press the physical emergency stop button on the robot. This shuts off power completely to the robot. To restore power to the robot, rotate the button to disengage it.

Driving the robot:

1. Use the left joystick on the PS4 controller to move the robot forward, backward, right, or left.
2. The joystick angle determines the speed of the relevant drivetrain motor. The speed of the robot is proportional to how far the joystick is pushed from the idle position. The robot cannot steer but turns by rotating its entire body.

Operating the scooper/cutting weeds:

The scooper starts in a closed position and opens/closes depending on the current state.

1. Align front towards weed/plant.
2. Press to open scooper.

3. Move forward until the scooper encompasses the weed/plant.
4. Press again to begin the cutting process.

Operating the transportation/lifting mechanism:

1. The arms start at a lowered position. To lift the scooper up when it is in a lowered position, press the button. The arms should lift up slowly over a 5 second interval before reaching the storage door.
2. To lower the scooper when it is in a raised position, press the button again. The arms should lower slowly over a 5 second interval before reaching a level position, horizontal with the robot's chassis.

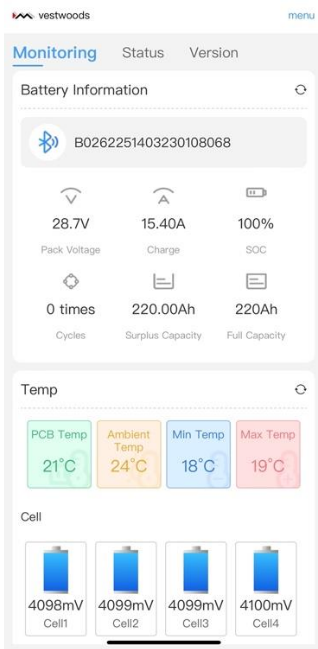
Operating the storage door mechanism:

1. The storage door starts in a closed state. When the door is closed, press the button to open the door 110 degrees.
2. When the door is open, press the button again to close the door.

Charging the battery:

1. Turn off the robot.
2. Plug battery charger into standard wall outlet (120 V).
3. Plug the charger wire from the battery to the charger wire of the battery charger.
4. Turn on the battery charger and ensure the battery charger slider is all the way to the right (35A/18A) and that it is set to LiFePO4 as shown below:

5. Monitor charging status by downloading the Vestwoods app and following the instructions to connect to the battery. When prompted, scan the barcode on the top right of the battery. The Vestwoods app interface is shown below:



6. Once charging has been completed, remove the leads and battery charger from the robot.

Maintenance Guides

Emptying the storage bin:

1. Turn off the robot.
2. Flip the side door open and over the top of the robot shell.
3. Grasp the handle on the top of the storage bin and pull the bin straight out along the tracks.
4. When the handle on the opposite side of the bin becomes reachable, grasp it and lift the storage bin out of the robot.
5. Turn the bin upside down and shake slightly to remove all cut weeds.
6. Once empty, place one end of the storage bin back onto the tracks inside the robot and slide it all the way in.

Replacing 3D Printed Parts:

1. .STL files for all 3D printed parts can be found in the CAD package.
2. Using slicer software, select desired print settings. The current parts on the robot are ABS and most are printed at 45% infill.

Adjusting Chain Tension:

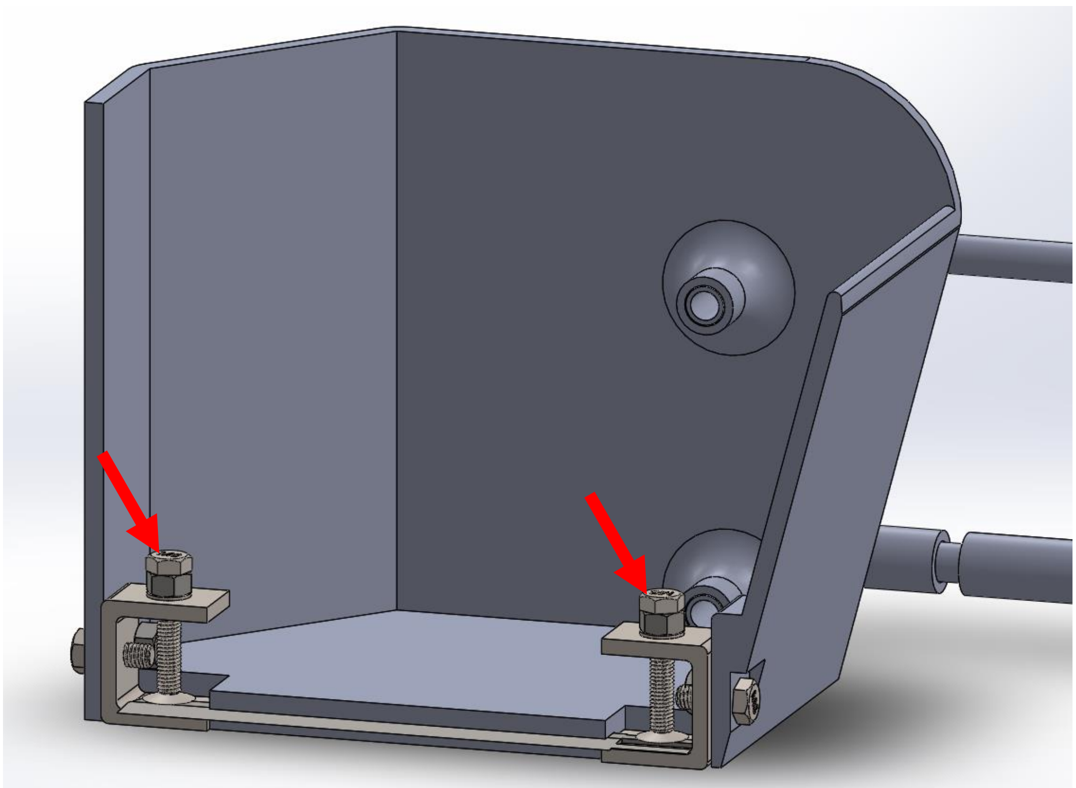
This task requires at least two people.
For each of the 4 chains,

1. Turn off the robot.
2. Support the robot on 4 jack stands such that all 4 wheels are off the ground.
3. Remove the axle chain guards.
4. Loosen but do not remove the four fasteners that hold each of the two axle bearings to the chassis.

5. To tension the chain, pull the wheel away from the center sprockets. Slightly rotate the wheel so that the top and bottom of the chain are tensioned roughly the same amount.
6. Re-tighten the bearing bolts while holding the wheel in the tensioned position.

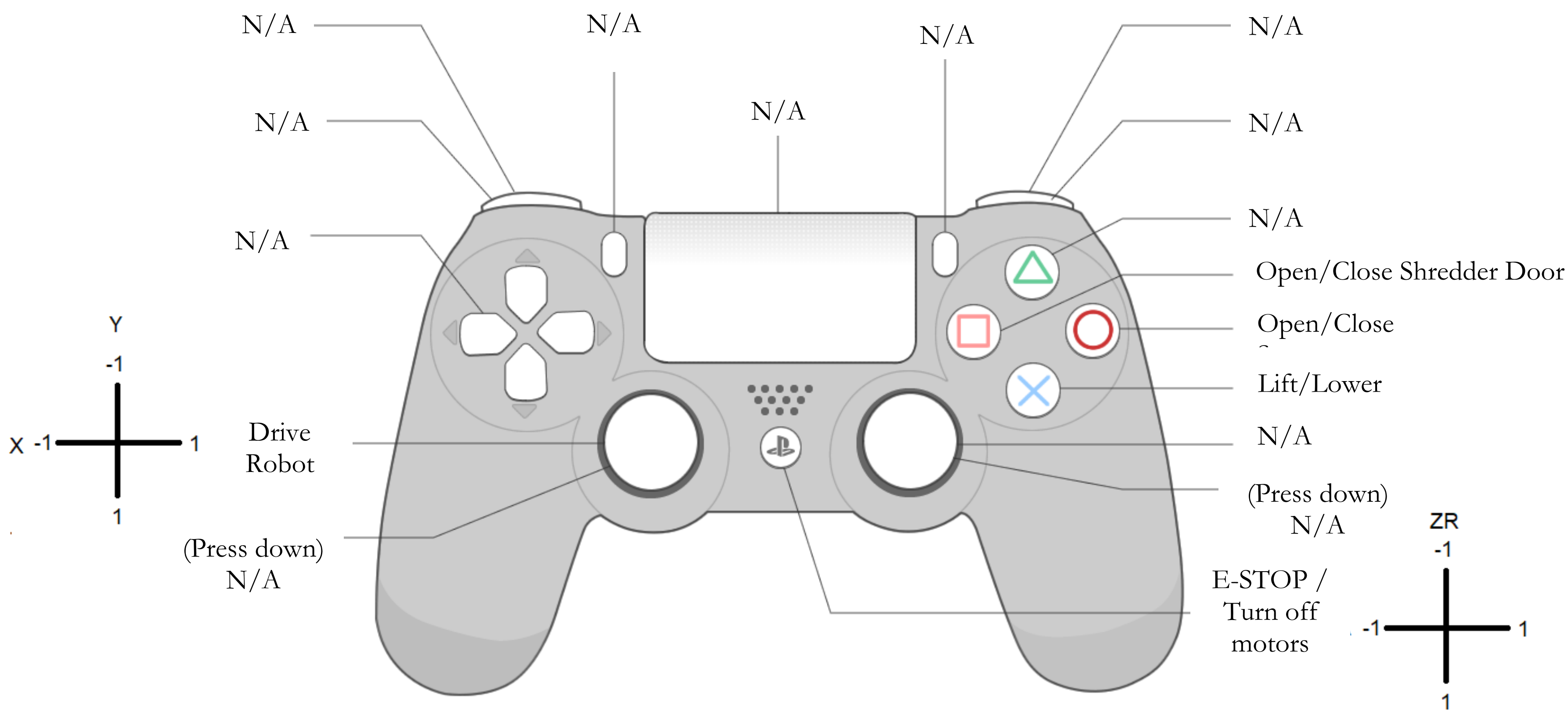
Charging the lights:
1. Lower the scooper to the horizontal position using the controller to allow access to the light sources.
2. Uncover the canvas and loosen the strip of the lights.
3. Charge the lights via USB.
4. When fully charged, situate the light back to its original position.

Changing the blade:
Gloves are required to prevent cuts.
1. Turn off the robot.
2. Loosen the bolt on top of the clamp.



3. Carefully remove the dull blade.
4. Insert a new blade in the clamp.
5. Tighten the clamp.

Controller Mapping:



Appendix B – Risk Assessment

designsafe Report

Application:	Weed Robot	Analyst Name(s):	Jackie Chen, Claire Franz, Sachi Hiji, Ayush Kakkanat, Lewis Kanagy, Carlos Rodriguez, Kaela Stern, William Ta
Description:		Company:	F75 F76
Product Identifier:		Facility Location:	Cal Poly, SLO
Assessment Type:	Detailed		
Limits:			
Sources:			
Risk Scoring System:	ANSI B11.0 Two Factor		

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

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment Severity Probability	Risk Level	Risk Reduction Methods /Control System	Final Assessment Severity Probability	Risk Level	Status / Responsible /Comments /Reference
1-1-1	adult first use / test	mechanical : crushing moving object	Serious Remote	Low		Serious		
1-1-2	adult first use / test	mechanical : cutting / severing blades	Serious Unlikely	Medium		Serious		
1-1-3	adult first use / test	mechanical : drawing-in / trapping / entanglement rotating shafts	Serious Unlikely	Medium		Serious		
1-1-4	adult first use / test	mechanical : pinch point rotating components	Moderate Likely	Medium		Moderate		
1-1-5	adult first use / test	electrical / electronic : energized equipment / live parts battery	Moderate Remote	Negligible		Moderate		
1-1-6	adult first use / test	electrical / electronic : shorts / arcing / sparking wiring issues	Moderate Unlikely	Low		Moderate		
1-1-7	adult first use / test	electrical / electronic : water / wet locations rain/cleaning	Moderate Likely	Medium		Moderate		

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment Severity Probability	Risk Level	Risk Reduction Methods /Control System	Final Assessment Severity Probability	Risk Level	Status / Responsible /Comments /Reference
1-1-8	adult first use / test	slips / trips / falls : falling material / object falling from scooper	Minor Likely	Low		Minor		
1-1-9	adult first use / test	ergonomics / human factors : lifting / bending / twisting robot weighs >35 lb	Minor Likely	Low		Minor		
1-1-10	adult first use / test	environmental / industrial hygiene : corrosion	Minor Unlikely	Negligible		Minor		
1-2-1	adult normal use	mechanical : crushing moving object	Serious Remote	Low		Serious		
1-2-2	adult normal use	mechanical : cutting / severing blades	Serious Unlikely	Medium		Serious		
1-2-3	adult normal use	mechanical : drawing-in / trapping / entanglement rotating shafts	Serious Unlikely	Medium		Serious		
1-2-4	adult normal use	mechanical : pinch point rotating components	Moderate Likely	Medium		Moderate		
1-2-5	adult normal use	electrical / electronic : energized equipment / live parts battery	Moderate Remote	Negligible		Moderate		
1-2-6	adult normal use	electrical / electronic : shorts / arcing / sparking wiring issues	Moderate Unlikely	Low		Moderate		

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment Severity Probability	Risk Level	Risk Reduction Methods /Control System	Final Assessment Severity Probability	Risk Level	Status / Responsible /Comments /Reference
1-2-7	adult normal use	electrical / electronic : water / wet locations rain/cleaning	Moderate Likely	Medium		Moderate		
1-2-8	adult normal use	slips / trips / falls : falling material / object falling from scooper	Minor Likely	Low		Minor		
1-2-9	adult normal use	environmental / industrial hygiene : corrosion battery corrosion	Minor Unlikely	Negligible		Minor		
1-3-1	adult maintenance / lubrication	mechanical : cutting / severing blades	Moderate Unlikely	Low		Moderate		
1-3-2	adult maintenance / lubrication	mechanical : drawing-in / trapping / entanglement rotating shafts	Moderate Unlikely	Low		Moderate		
1-3-3	adult maintenance / lubrication	mechanical : pinch point rotating components	Moderate Likely	Medium		Moderate		
1-3-4	adult maintenance / lubrication	environmental / industrial hygiene : corrosion battery corrosion	Minor Unlikely	Negligible		Minor		
1-4-1	adult repair tasks	mechanical : cutting / severing sharp blades	Moderate Unlikely	Low		Moderate		
1-4-2	adult repair tasks	mechanical : drawing-in / trapping / entanglement rotating shafts	Moderate Unlikely	Low		Moderate		

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment Severity Probability	Risk Level	Risk Reduction Methods /Control System	Final Assessment Severity Probability	Risk Level	Status / Responsible /Comments /Reference
1-4-3	adult repair tasks	mechanical : pinch point rotating components	Moderate Likely	Medium		Moderate		
1-4-4	adult repair tasks	environmental / industrial hygiene : corrosion battery corrosion	Minor Unlikely	Negligible		Minor		
1-5-1	adult trouble-shooting / problem solving	mechanical : crushing large moving object	Moderate Unlikely	Low		Moderate		
1-5-2	adult trouble-shooting / problem solving	mechanical : cutting / severing blades	Serious Unlikely	Medium		Serious		
1-5-3	adult trouble-shooting / problem solving	mechanical : drawing-in / trapping / entanglement rotating shafts	Serious Likely	High	Shredder door will be closed at all times unless a weed is being dumped into the shredder. Shredder will not turn on unless door is closed.	Serious Unlikely	Medium	
1-5-4	adult trouble-shooting / problem solving	mechanical : pinch point rotating components	Moderate Likely	Medium		Moderate		
1-5-5	adult trouble-shooting / problem solving	electrical / electronic : energized equipment / live parts battery	Moderate Unlikely	Low		Moderate		
1-5-6	adult trouble-shooting / problem solving	electrical / electronic : shorts / arcing / sparking wiring issues	Serious Unlikely	Medium		Serious		
1-5-7	adult trouble-shooting / problem solving	electrical / electronic : water / wet locations rain/cleaning	Moderate Unlikely	Low		Moderate		

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment Severity Probability	Risk Level	Risk Reduction Methods /Control System	Final Assessment Severity Probability	Risk Level	Status / Responsible /Comments /Reference
1-5-8	adult trouble-shooting / problem solving	slips / trips / falls : falling material / object falling from scooper	Minor Very Likely	Medium		Minor		
1-5-9	adult trouble-shooting / problem solving	environmental / industrial hygiene : corrosion battery corrosion	Minor Unlikely	Negligible		Minor		
1-6-1	adult cleaning	mechanical : pinch point rotating components	Moderate Likely	Medium		Moderate		
1-6-2	adult cleaning	electrical / electronic : energized equipment / live parts battery	Moderate Unlikely	Low		Moderate		
1-6-3	adult cleaning	electrical / electronic : water / wet locations rain/cleaning	Moderate Unlikely	Low		Moderate		
1-6-4	adult cleaning	slips / trips / falls : debris weeds	Minor Likely	Low		Minor		
1-6-5	adult cleaning	environmental / industrial hygiene : corrosion battery corrosion	Minor Unlikely	Negligible		Minor		
1-7-1	adult storage	electrical / electronic : energized equipment / live parts battery	Moderate Unlikely	Low		Moderate		
1-7-2	adult storage	environmental / industrial hygiene : corrosion battery corrosion	Minor Unlikely	Negligible		Minor		

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment Severity Probability	Risk Level	Risk Reduction Methods /Control System	Final Assessment Severity Probability	Risk Level	Status / Responsible /Comments /Reference
1-8-1	adult misuse	mechanical : crushing large moving object	Serious Unlikely	Medium		Serious		
1-8-2	adult misuse	mechanical : cutting / severing blades	Serious Likely	High	Warning labels on scooper. Scooper motors were selected with a low enough torque limit to prevent serious injury.	Serious Unlikely	Medium	
1-8-3	adult misuse	mechanical : drawing-in / trapping / entanglement rotating shafts	Serious Likely	High	Shredder door will be closed at all times unless a weed is being dumped into the shredder. Shredder will not turn on unless door is closed.	Serious Unlikely	Medium	
1-8-4	adult misuse	mechanical : pinch point rotating components	Moderate Likely	Medium		Moderate		
1-8-5	adult misuse	mechanical : unexpected start coding bug	Serious Unlikely	Medium		Serious		
1-8-6	adult misuse	mechanical : product instability rough terrain	Moderate Unlikely	Low		Moderate		
1-8-7	adult misuse	electrical / electronic : energized equipment / live parts battery	Moderate Unlikely	Low		Moderate		
1-8-8	adult misuse	electrical / electronic : arc flash wiring issues	Serious Unlikely	Medium		Serious		

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment Severity Probability	Risk Level	Risk Reduction Methods /Control System	Final Assessment Severity Probability	Risk Level	Status / Responsible /Comments /Reference
1-8-9	adult misuse	electrical / electronic : shorts / arcing / sparking wiring issues	Serious Unlikely	Medium		Serious		
1-8-10	adult misuse	electrical / electronic : water / wet locations rain/cleaning	Moderate Unlikely	Low		Moderate		
1-8-11	adult misuse	electrical / electronic : unexpected start up / motion coding bug	Serious Unlikely	Medium		Serious		
1-8-12	adult misuse	slips / trips / falls : falling material / object falling from scooper	Minor Likely	Low		Minor		
1-8-13	adult misuse	environmental / industrial hygiene : corrosion battery corrosion	Minor Unlikely	Negligible		Minor		
1-8-14	adult misuse	chemical : skin exposed to toxic chemical battery corrosion	Moderate Unlikely	Low		Moderate		
2-1-1	very young (0-7 yrs) walk near	mechanical : crushing large moving object	Catastrophic Unlikely	Medium		Catastrophic		
2-1-2	very young (0-7 yrs) walk near	mechanical : cutting / severing blades	Serious Unlikely	Medium		Serious		
2-1-3	very young (0-7 yrs) walk near	mechanical : drawing-in / trapping / entanglement rotating shafts	Catastrophic Unlikely	Medium		Catastrophic		

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment Severity Probability	Risk Level	Risk Reduction Methods /Control System	Final Assessment Severity Probability	Risk Level	Status / Responsible /Comments /Reference
2-1-4	very young (0-7 yrs) walk near	mechanical : head bump on overhead objects scooper	Moderate Unlikely	Low		Moderate		
2-1-5	very young (0-7 yrs) walk near	slips / trips / falls : falling material / object falling from scooper	Minor Likely	Low		Minor		
2-2-1	very young (0-7 yrs) observe/watch	slips / trips / falls : falling material / object falling from scooper	Minor Likely	Low		Minor		
3-1-1	child / youth (8-18 yrs) walk near	mechanical : crushing large moving object	Serious Unlikely	Medium		Serious		
3-1-2	child / youth (8-18 yrs) walk near	mechanical : cutting / severing blades	Serious Unlikely	Medium		Serious		
3-1-3	child / youth (8-18 yrs) walk near	mechanical : drawing-in / trapping / entanglement rotating shafts	Serious Unlikely	Medium		Serious		
3-1-4	child / youth (8-18 yrs) walk near	slips / trips / falls : falling material / object falling from scooper	Minor Likely	Low		Minor		
3-2-1	child / youth (8-18 yrs) observe/watch	slips / trips / falls : falling material / object falling from scooper	Minor Likely	Low		Minor		
3-3-1	child / youth (8-18 yrs) cleaning	mechanical : pinch point rotating components	Serious Unlikely	Medium		Serious		

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment Severity Probability	Risk Level	Risk Reduction Methods /Control System	Final Assessment Severity Probability	Risk Level	Status / Responsible /Comments /Reference
3-3-2	child / youth (8-18 yrs) cleaning	electrical / electronic : energized equipment / live parts battery	Moderate Unlikely	Low		Moderate		
3-3-3	child / youth (8-18 yrs) cleaning	electrical / electronic : water / wet locations rain/cleaning	Moderate Unlikely	Low		Moderate		
3-3-4	child / youth (8-18 yrs) cleaning	slips / trips / falls : debris weeds	Minor Likely	Low		Minor		
3-3-5	child / youth (8-18 yrs) cleaning	environmental / industrial hygiene : corrosion battery corrosion	Minor Unlikely	Negligible		Minor		
4-1-1	passer-by / non-user walk near	mechanical : crushing large moving object	Serious Unlikely	Medium		Serious		
4-1-2	passer-by / non-user walk near	mechanical : cutting / severing blades	Moderate Unlikely	Low		Moderate		
4-1-3	passer-by / non-user walk near	mechanical : drawing-in / trapping / entanglement rotating shafts	Serious Unlikely	Medium		Serious		
4-1-4	passer-by / non-user walk near	mechanical : pinch point rotating components	Moderate Unlikely	Low		Moderate		
4-1-5	passer-by / non-user walk near	slips / trips / falls : falling material / object falling from scooper	Minor Likely	Low		Minor		

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment Severity Probability	Risk Level	Risk Reduction Methods /Control System	Final Assessment Severity Probability	Risk Level	Status / Responsible /Comments /Reference
4-2-1	passer-by / non-user observe / watch	slips / trips / falls : falling material / object falling from scooper	Minor Likely	Low		Minor		

Appendix C – Final Project Budget

Complete Budget for Senior Project - Weed Robot				
Senior Project:	Weed Robot			
	F75 - Weed	F76 - Scouts		
Team members:	Claire Franz, cjfranz@calpoly.edu; Lewis Kanagy, lkanagy@calpoly.edu; Carlos Rodriguez, crodr137@calpoly.edu; Kaela Stern, kstern02@calpoly.edu	Sachi Reiko Hiji, shiji@calpoly.edu; Ayush Sudhee, akakkana@calpoly.edu; Jackie Chen, jchen329@calpoly.edu; William Ta, wita@calpoly.edu		
Faculty Advisor:	Eileen Rossman, erossman@calpoly.edu	Total Expenses:	$	4,812.82
Sponsor:	Matthew Meadows & Summer Helmuth Girl Scouts of California's Central Coast - Camp Arnaz	Cal Poly Expenses:	$	886.83
Project Start:	Fall 2022 - Spring 2023	Sponsor Expenses:	$	3,925.99
Sponsor Budget:	**$6,000.00**	**Remaining Sponsor Balance:**	**$**	**2,074.01**
Cal Poly Budget:	**$1,000.00**			

Date purchased	Vendor	Items purchased	Amount	
20-Jan-23	Amazon - Hatchbox	3D ABS Filament	$	68.97
25-Jan-23	Amazon - Hatchbox	3D PLA Filament	$	24.99
25-Jan-23	Amazon - CEE	6061 T651 Aluminum Sheet Metal	$	63.96
25-Jan-23	Amazon - Sony Brook Hams	C - Clamp	$	10.99
26-Jan-23	Amazon - NVIDIA	Jetson Nano	$	200.00
26-Jan-23	Amazon - Gorilla	Epoxy	$	10.76
30-Jan-23	Amazon - Hatchbox	Filament	$	27.18
31-Jan-23	Chassis	1-1/2 x 1-1/2 H.S.T. 20'	$	133.29
5-Feb-23	The Home Depot	5 PVC & Glue	$	44.82
6-Feb-23	Amazon -Hatchbox	Filament	$	27.18
7-Feb-23	Harbor Freight	13in Tire Steel	$	26.09
9-Feb-23	Amazon - PGN Bearings	Bearing Mounts	$	45.34
11-Feb-23	Travel to pick up Chassis	N/A	$	191.26
11-Feb-23	Harbor Freight	3 - 13in Tire Steel	$	78.63
11-Feb-23	The Home Depot	Hex Bolt 5/16 x 1 1/2, Hex Bolt 3/8 x 3, Hex Nut 3/8, Flat Washer, Grit Sheet, Triangle Sheet, Hole Saw, Diameter Dowel	$	86.98
11-Feb-23	Amazon IMS Metal Made Easy	1" 304 SS Round Rod	$	130.44

13-Feb-23	Amazon	High Torque Motors	$	36.96
20-Feb-23	Amazon - SanDisk	Memory Card	$	11.05
20-Feb-23	Amazon - Adafruit	PWM Driver	$	19.71
20-Feb-23	Amazon - Noctua	Fan	$	13.95
20-Feb-23	Amazon - Intel	Wifi Card	$	24.00
20-Feb-23	Amazon - EDGELEC	Breadboard Jumper Wires	$	11.99
23-Feb-23	Amazon - I-faster & YaeTek	24V Motor 250W & Controller 2650RPM	$	99.78
24-Feb-23	Amazon	USB Bluetooth Adapter	$	10.86
27-Feb-23	Amazon - Fire Sale Merchant	3D PLA Filament	$	29.35
28-Feb-23	MidwestSteelSupply.com	.080 6061 Aluminum Sheet	$	132.83
5-Mar-23	The Home Depot	Spray Paint	$	5.44
8-Mar-23	Amazon	Bearings, inserts, coupling, motors+	$	162.14
9-Mar-23	Amazon	Step down converters	$	10.86
16-Mar-23	Amazon - Mybecca	5 Yards Hunter Green Denier	$	43.49
6-Apr-23	The Home Depot	Round Alum Tube	$	49.89
6-Apr-23	The Home Depot	Steel Machine Screws	$	1.48
7-Apr-23	Amazon - PGN Bearings	Pillow Block Bearings 1/2" Bore	$	198.53
7-Apr-23	Amazon - L-Star Motor	2 - Worm Gear Reducer 25:1		-
7-Apr-23	Amazon - Uncell	8mm to 10mm Coupling Set Screw		-
7-Apr-23	Amazon - Uncell	11 mm Steel Bar Stock 100mm Long		-
7-Apr-23	Amazon - PGN Bearings	2 - # 40 Roller Chian - 10ft		-
7-Apr-23	Amazon -Small Parts	1018 Carbon Steel 0.625" Dia. - 12"	$	18.08
7-Apr-23	Nitro Chain	4 - #40 Sprocket 5/8	$	245.33
7-Apr-23	Nitro Chain	4 - #40 Sprocket 3/4		-
10-Apr-23	Amazon - FOIIOE / Vestwoods	24V 220Ah Lithium Battery	$	1,339.99
10-Apr-23	Amazon	Filament	$	25.00
11-Apr-23	Amazon - Swpeet	Carbon Steel Keys	$	14.12
11-Apr-23	Amazon - Forney	Mild Carbon Steel Alloy - 1/2" Diameter	$	14.12
12-Apr-23	The Home Depot	Metic Cap Screws	$	6.94
15-Apr-23	Ebay - OEM Sony	Playstation Dualshock 4 Controller	$	27.18
18-Apr-23	Amazon - Makeronics+	Jetson Nano Case, Tape, Bearings	$	29.20
20-Apr-23	McMaster-Carr	Alloy Steel Set Screws	$	15.76
20-Apr-23	Metals Depot	3/8" Dia, 6061 Aluminum Rod	$	81.91
22-Apr-23	Amazon	64GB MicroSDXC	$	10.57
23-Apr-23	Amazon - HTRC	35 Amp Smart Charger	$	143.43
23-Apr-23	Amazon - PGN Bearings	4 Mounted Pillow Block 5/8" Bore	$	30.56

25-Apr-23	Amazon	Motor Drivers	$	17.39
27-Apr-23	Amazon	Fuses	$	8.69
27-Apr-23	Amazon	Gears	$	19.92
27-Apr-23	CVS	Glue	$	8.69
1-May-23	ACE	Fasteners +	$	43.15
1-May-23	The Home Depot	Sandpaper	$	7.59
3-May-23	Amazon - Unxcell	6 #40 Chain Offset Hald Link 1/2"	$	8.15
4-May-23	The Home Depot	1/2 Aluminum C Channel	$	13.97
4-May-23	The Home Depot	Washer, Hex Nut +	$	23.11
4-May-23	The Home Depot	Hex Nuts +	$	4.57
8-May-23	The Home Depot	Hex Nuts, Screw+	$	8.70
9-May-23	Miners Ace Hardware	Fasteners +	$	18.92
9-May-23	Amazon	Connector wires	$	10.86
9-May-23	Amazon	Terminal Strip	$	8.69
9-May-23	Amazon	24V Wire	$	13.04
9-May-23	Amazon	Connectors	$	11.95
11-May-23	Amazon - Hatchbox	3D ABS Filament	$	52.18
11-May-23	Miners Ace Hardware	Fasteners +	$	15.61
14-May-23	Amazon	Flashlight	$	20.65
16-May-23	Amazon	Motor Drivers	$	11.95
17-May-23	Amazon - HiLetgo	Arduino - Current Limit	$	11.95
18-May-23	Amazon	Red Emergancy Stop	$	10.65
21-May-23	Amazon	Motor Drivers	$	17.39
23-May-23	Miners Ace Hardware	Fasteners & Threadlock	$	12.60
23-May-23	The Home Depot	Screws +	$	24.87
23-May-23	Amazon	Spray Paint	$	15.36
23-May-23	Amazon	Step-Down Converters	$	47.28
24-May-23	The Home Depot	Socket Set	$	19.37
25-May-23	Miners Ace Hardware	Fasteners +	$	5.51
25-May-23	The Home Depot	Washes, Bolts, Nuts	$	40.46
25-May-23	Amazon - Hatchbox	3D ABS Filament	$	23.91
25-May-23	Amazon - Hatchbox	3D ABS Filament	$	26.09
25-May-23	The Home Depot	Screws +	$	1.50
26-May-23	Amazon	Driver Modules	$	11.95
27-May-23	McMaster-Carr	1/4Thick 1/2Wide 2Long 28lb magnet	$	71.52
28-May-23	The Home Depot	Washers+	$	45.46
28-May-23	Amazon	Driver Modules	$	17.39
31-May-23	Michaels	Velcro	$	5.21
31-May-23	Staples	AVY LSR	$	16.30
1-Jun-23	Ace Hardware	Fasteners +	$	24.89

Appendix D – Wiring Diagrams

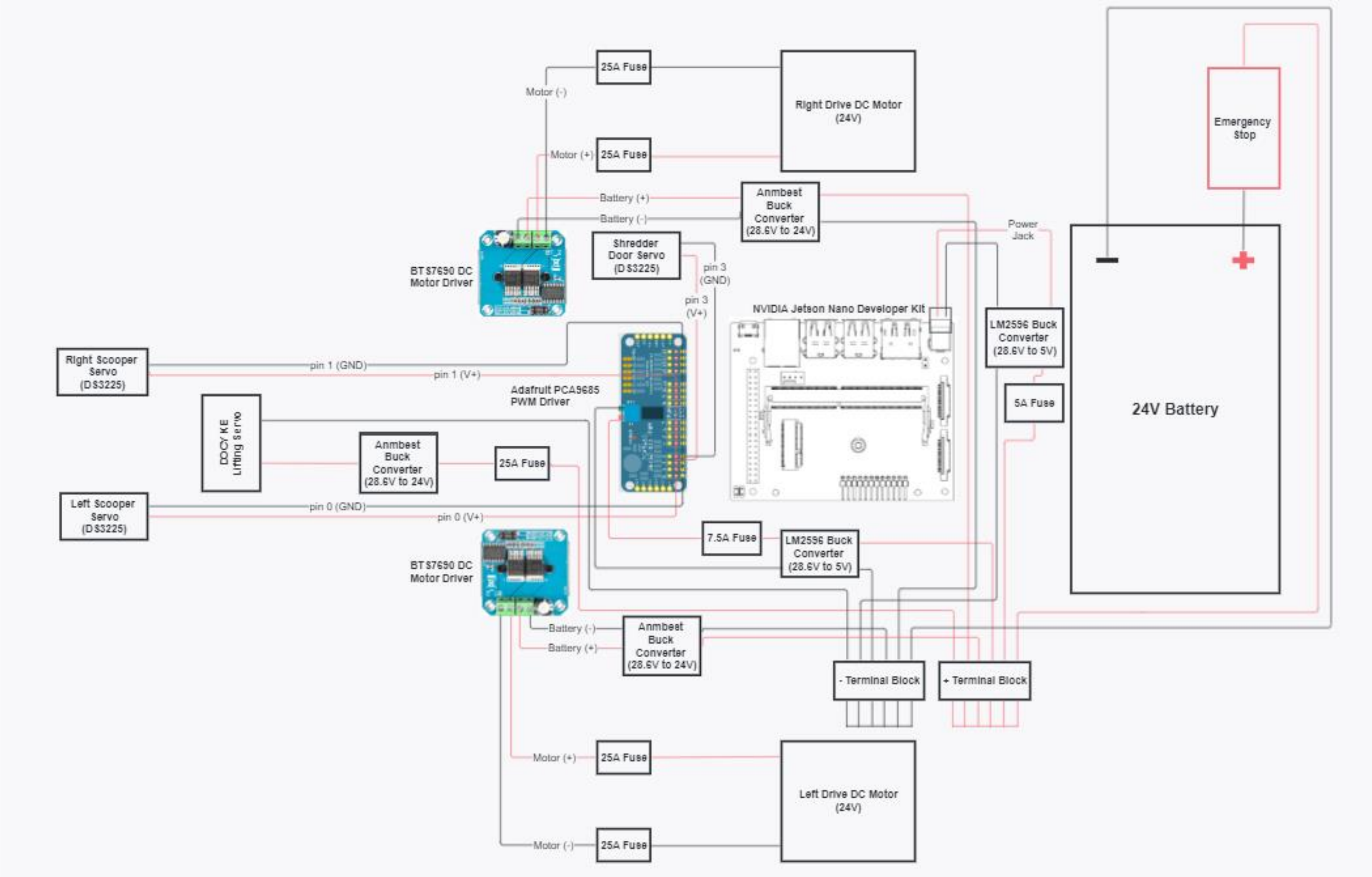


Figure D1. Power Connections

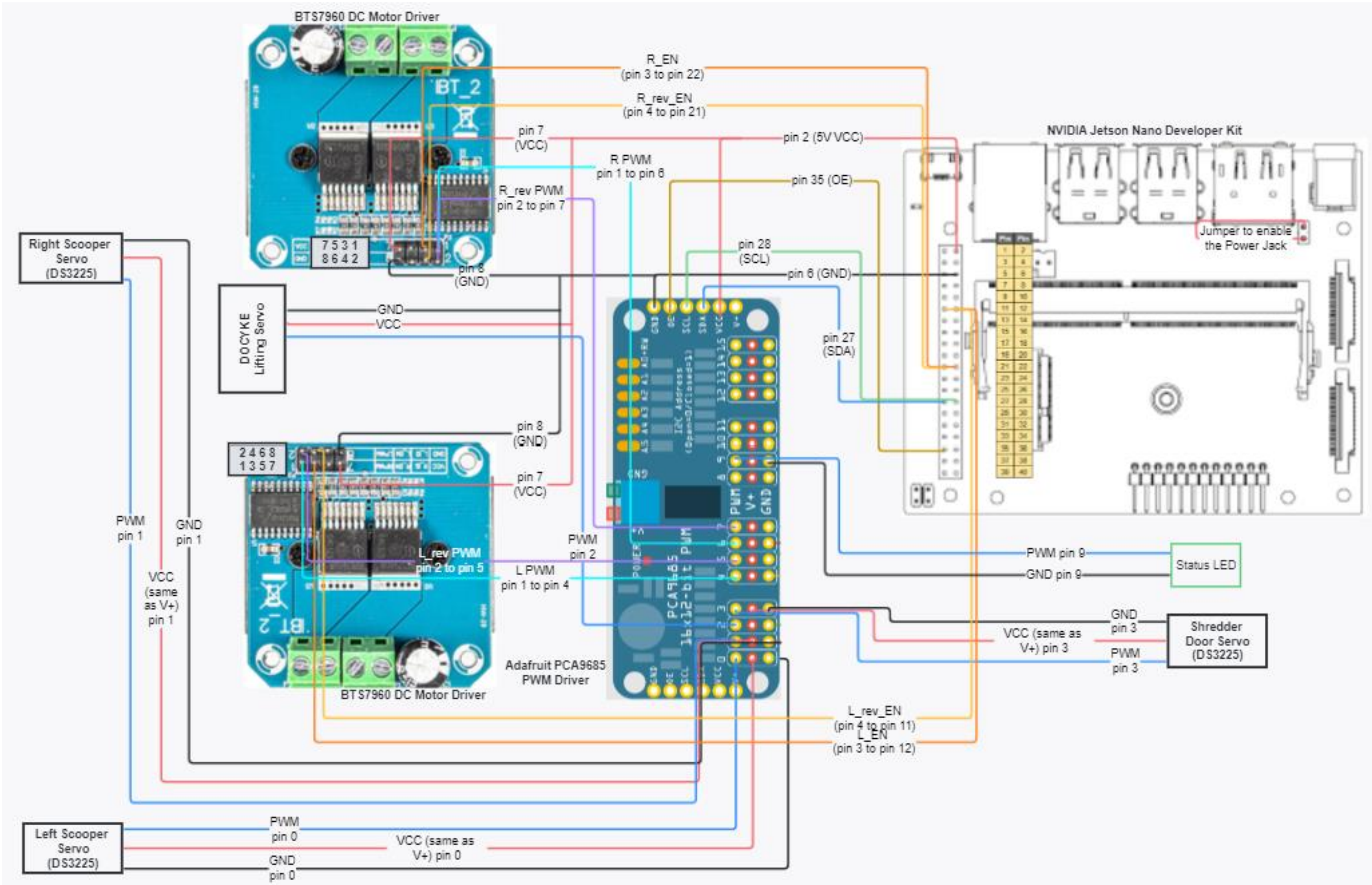


Figure D2. Logic Connections

Appendix E – Design Verification Plan & Report (DVPR)

DVP&R - Design Verification Plan (& Report)

Project:	F76: Weed Robot	Sponsor:	Girl Scouts of California's Central Coast		Edit Date:	5/30/2023

TEST PLAN										TEST RESULTS	
Test #	Specification	Test Description	Measurements	Acceptance Criteria	Required Facilities/ Equipment	Parts Needed	Responsibility	TIMING		Numerical Results	Notes on Testing
								Start date	Finish date		
1	Weed Elimination Rate	Carry out cutting operation and measure the amount of weed cut versus the amount of weed that remains.	Weed weight before and after cutting	60% of Weeds Eliminated	Scale	Scooper (SP)	Sachi	5/11/23	5/30/23	N/A	Scooper was not able to cut the weeds due to blade design. Future design iterations should be steadier and have better interfacing of the blade with the slot for cutting.
2	Ratio of Weeds Cut to Weeds Stored	Carry out cutting and weed transporting and measure amount of weeds cut vs amount of weeds stored	Weed weight in storage and weight of weeds that fall out of the scooper	80% of Weeds Stored	Scale	Scooper and Storage (VP)	Willliam	5/11/23	5/30/23	N/A	Scooper was not able to cut the weeds due to design. Future design iterations should be steadier and have better interfacing of the blade with the slot for cutting.
3	Operation Area Size/ Remote Control Range	Measure farthest remote control range	Farthest distance between remote controller and robot	>30 ft	Tape Measure	Remote Controller, Receiver (VP)	Ayush	5/1/23	5/1/23	164 ft	The controller will disconnect past this distance

4	Nighttime Detection Range	Survey personnel's ability to see the robot at night	Distance from robot at night	>20ft	Tape Measure	Lights (VP)	Ayush	5/25/23	5/25/23	370 ft	The light was very bright and visible.
5	Maintenance Time	Replace blade and battery	Amount of time it takes for typical maintenance	<10 minutes	Stopwatch	Blade and Battery (VP)	Sachi	5/26/23	5/27/23	2 min 59s	Includes replacing and sharpening the blade
6	Speed of Lifting Arms	Run the motor at several speeds	Amount of time it takes for the arms to lift up or lower the arms	Between 8 and 15 seconds	Stopwatch	Transport Arms (VP)	Jackie	5/6/23	5/30/23	8.59s	Arms do not completely lift up to the storage door due to issues with gear meshing. Future design should stabilize the servo by securing it more firmly onto the chassis.
7	Multiple motor control	Run any motor with remote control	-	100% Pass Rate	Power Supply	Motors	Jackie	5/12/23	5/30/23	100% Pass	

8	Scooper Alignment	We will investigate the alignment of the blade and the slot	The distance of the bases of the front and back of the scooper in the closed position	The gap between the halves is < 1/8"	Calipers Power supply	Scooper Head, Motors	Ayush	5/1/23	5/30/2023	0.013"	
9	Operating time per charge is 2 hours.	We will operate the robot and measure the amount of time it can function before running out of battery.	Elapsed time before battery needs to be charged.	Elapsed time >= 2 hours.	Stopwatch	Entire robot (VP)	Claire and Lewis	5/4/23	N/A	N/A	This test was not complete due to the controls not being finished. See the individual test sheet for mor details.
10	Motor Torque Curve	We will run a test to collect voltage and motor speed data to performa a motor parameterization on our motor to then determing the motor torque curve.	Voltage and RPM	>0.5 nm of stall torque	Power Supply	Drive Motor	Lewis and Carlos	5/4/23	3/19/23	0.85 Nm motor stall torque	More details can be found in the individual test sheet. This test was necessary for the motor parameterization.

11	Robot can withstand rainfall and hose spray cleaning (shell and housing protects component s).	We will place paper towels inside the robot body and spray the outside with a garden hose.	Presence of water on inside of robot shell after spray down with hose.	No water present on inside of robot.	Hose	Robot shell (VP)	Kaela	5/4/23	N/A	N/A	The electronics for the robot were not safely covered in time to complete this test without risking significant damage to the electronics.
12	Height of obstacles that can be driven over is plus or minus two inches.	We will roll the robot over obstacles of increasing size until the robot either becomes unstable or exceeds an obstacle size of 2 inches.	Maximum obstacle size	Robot remains stable and bottom of chassis does not scrape the 2-in obstacle	Tape Measure	Chassis, bearing mounts, axles, tires (SP)	Kaela and Carlos	2/16/23	5/25/23	1.5in	Tests were completed with a power supply that was unable to provide full power to the motors.
13	Time for full charge is less than or equal to 6 hours.	We will measure the amount of time it takes to charge the robot's batteries from dead to fully charged.	Total time to charge batteries	total time <= 6 hours	Power source, stopwatch	Batteries (VP)	Claire and Lewis	5/4/23	N/A	N/A	This test was not complete due to the battery not being able to be fully discharged.

Appendix F – Test Procedures

Test Procedure for Weed Elimination Rate

Test Name: Weed Elimination Test

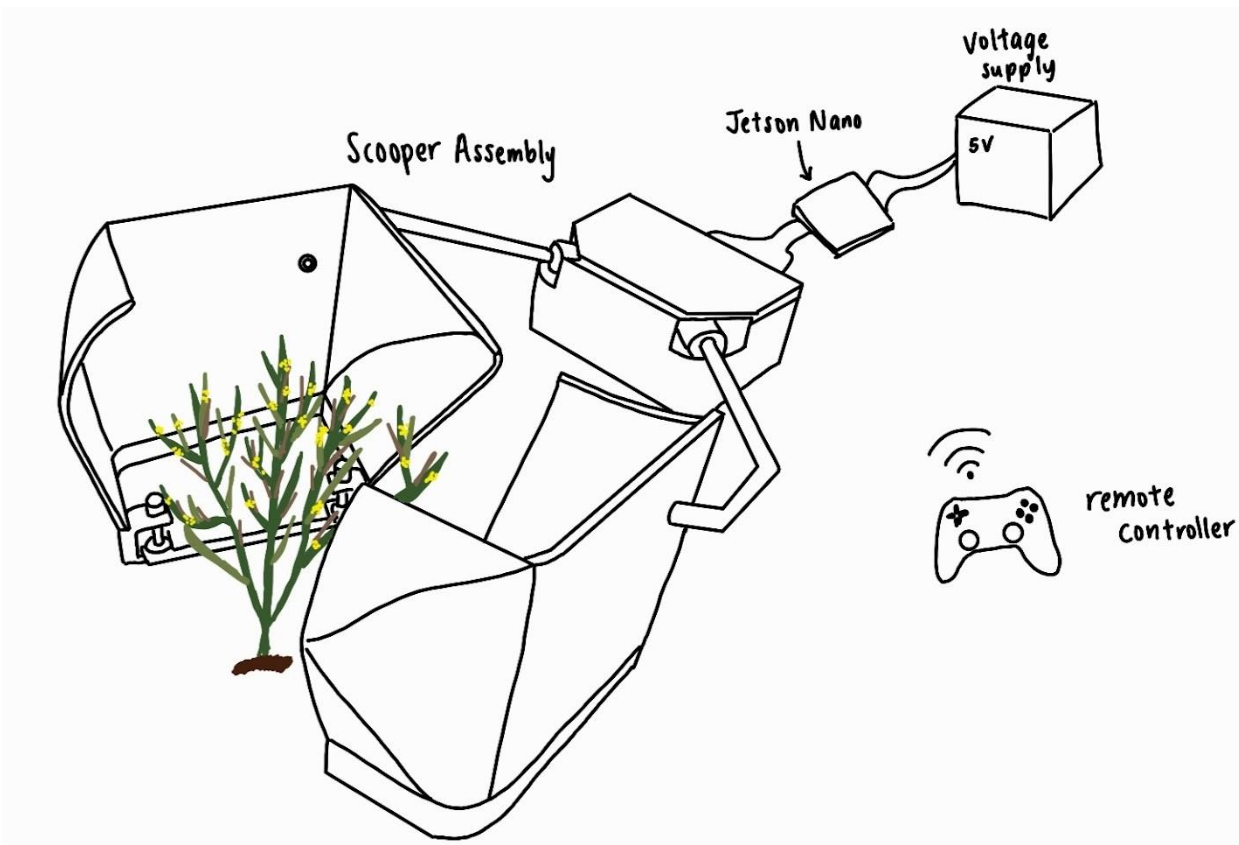
Purpose: The purpose of this test is to measure how effective one cutting operation of the weed scooper is.

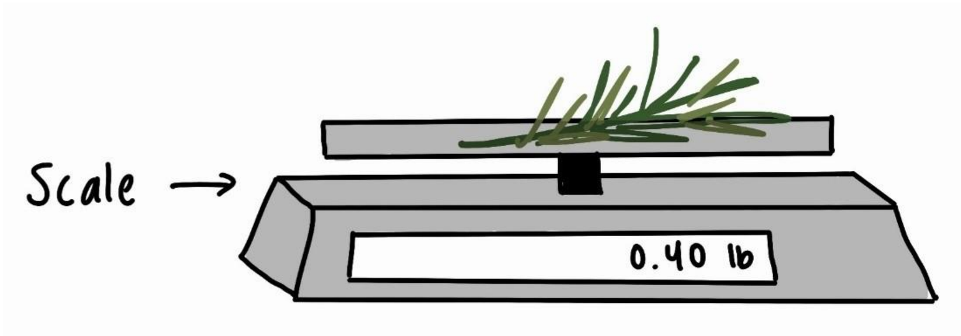
Scope: The weed cutting function which involves the entire scooper assembly will be tested.

Equipment:

- Mass Scale
- Voltage supply for Jetson Nano and motors
- Scooper head
- Shears to cut leftover weed

Diagram of Apparatus:





Hazards: Blades will be included with the scooper head halves so there are risks of cuts and pinch points

PPE Requirements: Gloves

Facility: The test can be conducted outside in a group member's backyard where there are weeds.

Procedure:

1. Run motor program and perform one cutting operation
2. Zero the scale
3. Weigh the portion of weeds in the scooper in lbs and record in table
4. Cut the remaining part of the weed above the ground, weigh that section (lb), then record the leftover weight in the table
5. Repeat cutting operation 9 more times, recording weed section weights each time
6. Calculate total weed weight and weed elimination rate for each trial to fill out the rest of the table

Results:

We will repeat the cutting procedure a total of 10 times. The acceptance criteria we want to achieve is to eliminate 60% of the weed after it is cut. This percentage will be a ratio of the scooped weight to the total weight of the scooper.

$$\text{Weed Elimination Rate} = \text{Scooped Weight}/\text{Total Weight} \times 100,$$

where,

$$\text{Total Weight} = \text{Scooped Weight} + \text{Leftover Weight}$$

After 10 trials have been conducted, we will calculate the average weed elimination rate to determine if the weed scooper is overall successful at achieving the 60% weed elimination rate.

Data:

Trial #	Scooped Weight (lb)	Leftover Weight (lb)	Total Weed Weight (lb)	Weed Elimination Rate (%)
1	Inconclusive	Inconclusive	Inconclusive	Inconclusive
2				
3				
4				
5				
6				
7				
8				
9				
10				

Test Date(s): 5/3/2023

Test Results: Average Weed Elimination Rate = Inconclusive

Performed By:
Jackie Chen
Sachi Hiji
Ayush Kakkanat
William Ta

Test Procedure for Weed Cutting Efficiency

Test Name: Weed cutting efficiency test

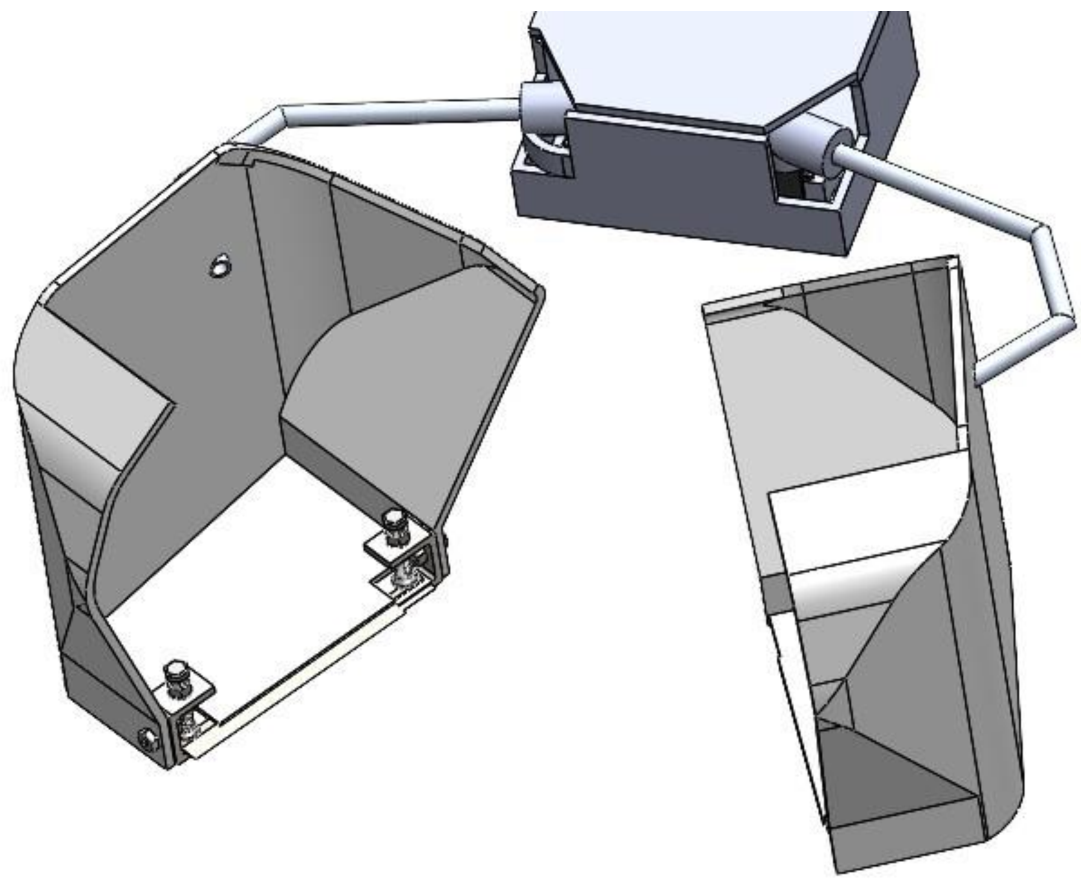
Purpose: The purpose of this test is to test whether the cutting mechanism can cut weeds and the success rate of cutting weeds.

Scope: The ratio of weeds cut to weeds stored is tested.

Equipment:

- Scale
- Scooper Head
- Plant Matter

Diagram of Apparatus:



The scooper head shown above will be tested with any type of weed to see if it is capable of cutting the weeds. This test will show any problem that arise from the cutting process.

Hazards: Blade from scooper head

PPE Requirements:
- Safety Glasses
- Gloves

Facility: Engineering 4 courtyard by Bonderson.

Procedure:
1. Secure weed stems to a fixture and ensure it is stable and fixed.
2. Actuate the cutting process (closing the scooper head) and try to cut at the base of the stem.
3. After the cutting process, examine the weed stem and determine whether it was cut and record it with a table
4. Repeat the experiment 25 times.

Data:

Attempted Cuts on Weeds	Weeds Stored
25	0

Test Date(s): 5/11/2023

Test Results: Percentage of weeds stored = 0%

Performed By: William Ta

Test Procedure for Remote Control Range

Test Name: Remote Control Range Test

Purpose: The purpose of this test is to measure the farthest remote control range distance.

Scope: The remote control function which involves the control of the motors actuating other sub functions will be tested.

Equipment:
- PS4 Controller
- Jetson Nano with receiver, power and a motor plugged in
- Two individuals

Diagram of Apparatus:
- PS4 Controller and Jetson Nano communication

Hazards: N/A

PPE Requirements: N/A

Facility: The test can be conducted at any open location such as Mt. Bishop road by the crop science unit.

Procedure:
1. Place the Jetson Nano on a stable surface with one individual observing the motors.
2. Starting at a distance of 5 feet away from the Jetson Nano, the other individual should control the motors using the PS4 controller.
3. Increase the distance in increments of 5 feet until the motors stop responding to the signal.
4. Reduce the distance in increments of 1 foot until the motors respond again.
5. Note the distance.
6. Repeat the process 10 times and calculate the average.

Data:

Trial #	Control Range (ft)
1	164
2	156
3	155
4	163
5	164
6	160
7	162
8	163
9	159
10	155

Test Date(s): 5/16/2023

Test Results: Average Control Range = 160.1 ft

Performed By: Ayush Kakkanat

Test Procedure for Night-time Detection Range

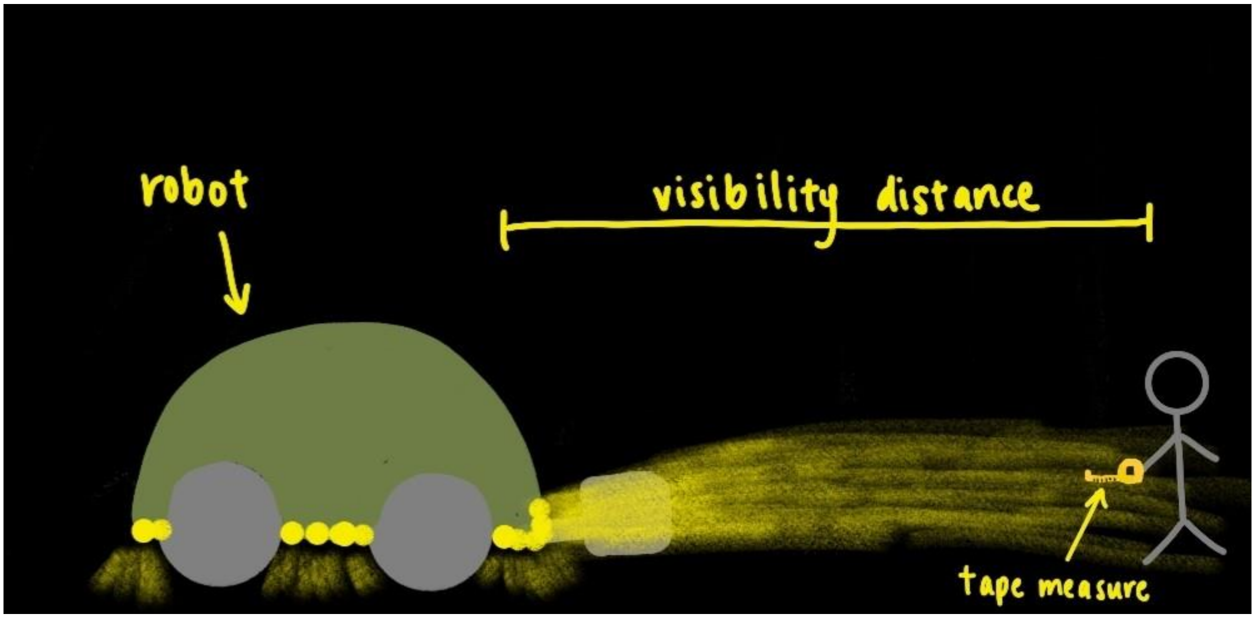
Test Name: Night-time Detection Range Test

Purpose: The purpose of this test is to measure the visibility of the robot at night.

Scope: The lighting function which involves LED lights will be tested.

Equipment:
- Tape Measure
- Battery or power supply for the LEDs

Diagram of Apparatus:



Hazards: N/A

PPE Requirements: N/A

Facility: The test can be conducted at any open location such as Mt. Bishop road by the crop science unit.

Procedure:
1. Take the robot out to an open field at night
2. Turn on the LED lights
3. Have one person stand at the robot holding the end of a tape measure
4. Have another person hold the base of the tape measure and walk away from the robot until the robot is no longer visible
5. Record the distance on the table
6. Repeat distance test 4 more times

Results:
We will get 5 distance measurements that represent the farthest visibility of the robot at night. We can take the average of the measurements to see what we may expect for night-time visibility in general. The acceptance criterion for this test is 300 ft.

Data:

Trial #	Distance from Robot (ft)
1	314
2	359
3	403
4	396
5	379

Test Date(s): 5/29/2023

Test Results: Average night-time visibility = 370.2 ft

Performed By:
Jackie Chen
Sachi Hiji
Ayush Kakkanat
William Ta

Test Procedure for Maintenance Time

Test Name: Maintenance Time Test

Purpose: The purpose of this test is to measure the time it takes to replace the blade and plug in battery.

Scope: The maintenance function which involves the quick replacement of critical components.

Equipment:

- Blade
- Battery Cable
- Stopwatch

Hazards: The blade can be sharp and cause injuries. Unplug the battery to avoid any loose cables.

PPE Requirements: Gloves

Facility: The test can be conducted at any location.

Procedure:

1. Start a stopwatch.
2. Open the scooper head.
3. Unscrew the C clamps.
4. Slide out blade.
5. Insert a new blade.
6. Screw the C clamps.
7. Close the scooper head.
8. Open the back panel to access the battery.
9. Plug in the battery charger.
10. Close the back panel.
11. Stop the stopwatch.
12. Repeat 5 times.
13. Repeat steps 1 through 12 but sharpen the existing blade using a knife sharpener instead of inserting a new one.

Data:

Variation	Trial #	Time
Blade Replacement Trial Data	1	1 min 6 sec
	2	42 sec
	3	1 min 22 sec
	4	49 sec
	5	57 sec
Blade Sharpening Trial Data	6	1 min 57 sec
	7	2 min 59 sec
	8	3 min 14 sec
	9	2 min 43 sec
	10	3 min 38 sec

Test Date(s): 5/28/2023

Test Results: Average blade replacement time ~ 59 sec
Average blade sharpening time ~ 2 min 54 sec

Performed By: Ayush Kakkanat

Test Procedure for Operation of Transportation Arms

Test Name: Transportation arms operation test.

Purpose:
- Confirm that the DOCYKE servo is strong enough to move the scooper assembly.
- Ensure that the lifting arms complete a rotation of 90 degrees within a period between 8 to 15 seconds.

Scope: Transport cut weeds to the shredder.

Test Equipment:
- Scooper assembly (lifting arms, motor box, scooper head, pillow block bearing)
- DOCYKE servo motor
- Controller setup (Jetson Nano, power cable, monitor, HDMI cable, mouse, keyboard, and Bluetooth adapter)
- PCA9685 PWM driver
- Pin connector wires
- PS4 remote controller
- Stopwatch
- 24V power supply

Hazards:
- An object in the scooper can be flung out if the speed is too high.
- An object or person in the path of the rotation can be hit by the scooper.

PPE Requirements: Safety glasses

Facility: Area with open space without people and a nearby outlet for power.

Procedure:
1. Verify that all participants are wearing proper safety glasses and clear the area of any unsecured objects.
2. Plug the Jetson Nano to power and set up its accessories.
3. Wire the PCA9685 PWM driver and DOCYKE Servo.
4. Run the python script needed to control the servo motor on the Jetson Nano.
5. Press the button on the remote controller that is mapped to moving the lifting arms and start the stopwatch at the same time.
6. Once the rotation ends, stop the stopwatch, and record the elapsed time.
7. Repeat steps 5 and 6 for lowering the arms.
8. Modify the time delay in the python script to adjust the rotation speed.
9. Repeat steps 5 through 7 until the elapsed time is within the desired range.

Results: Pass criteria: 8 to 15 second rotation time

Data:

Run	Time Elapsed – Lifting [s]	Time Elapsed – Lowering [s]	Rotation Speed [RPM]
1	3.52	3.83	1.59
2	4.57	4.22	1.32
3	4.36	4.71	1.28
4	4.51	4.64	1.28

Test Result:
Average Lifting Time: 4.24s
Average Lowering Time: 4.35s
Average Rotation Speed: 1.37s
Average Total Rotation Speed: 8.59s

Rotation speed was modified after the first trial to ensure >8 second rotation time.

Test Date(s): 5/30/23

Performed By: Jackie Chen

Test Procedure for Multiple Motor Control

Test Name: Multiple Motor Control Test

Purpose: The purpose of this test is to validate the functioning of a motor with its respective remote control input.

Scope: The remote control function which involves the control of the motors actuating other sub functions will be tested.

Equipment:

- PS4 Controller
- Jetson Nano with receiver, power and the motors plugged in

Diagram of Apparatus:

- PS4 Controller and Jetson Nano communication to control motors

Hazards: There are powerful motors which people should be weary of and stay at a distance from.

PPE Requirements: N/A

Facility: The test can be conducted at any location.

Procedure:

1. Place the Jetson Nano on a stable surface with one individual observing the motors.
2. Activate the desired motor to be actuated using its relevant remote control input button press.
3. Repeat 10 times with different motors.

Result: Pass criteria = 100% pass rate

Data:

Trial #	Motor	Pass/Fail
1	Scooper	Pass
2	Shredder Door	Pass
3	Drivetrain	Pass
4	Transportation	Pass
5	Scooper	Pass
6	Shredder Door	Pass
7	Drivetrain	Pass
8	Transportation	Pass
9	Scooper	Pass
10	Scooper	Pass

Test Date(s):
3/15/2023
5/12/2023

Test Results: Pass rate = 100%

Performed By: Jackie Chen

Test Procedure for Scooper Alignment

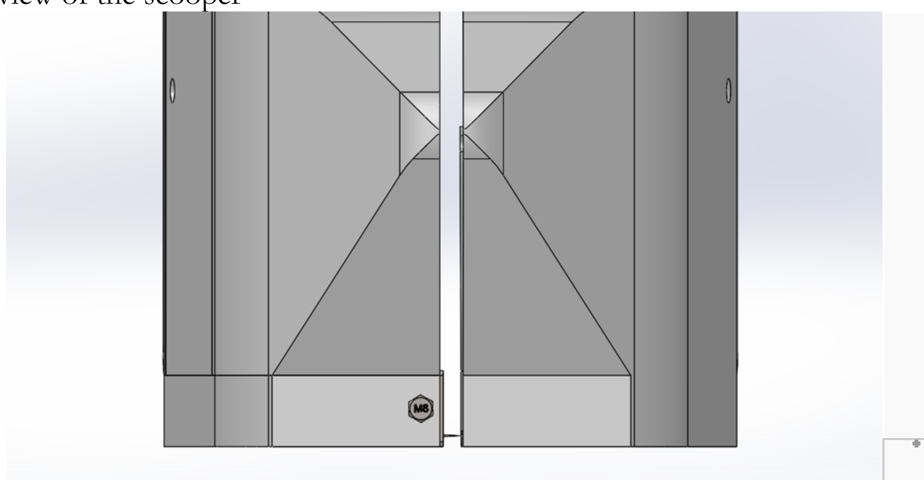
Test Name: Scooper Alignment Test

Purpose: The purpose of this test is to investigate the alignment of the two scooper halves and see if the blade aligns with the slot.

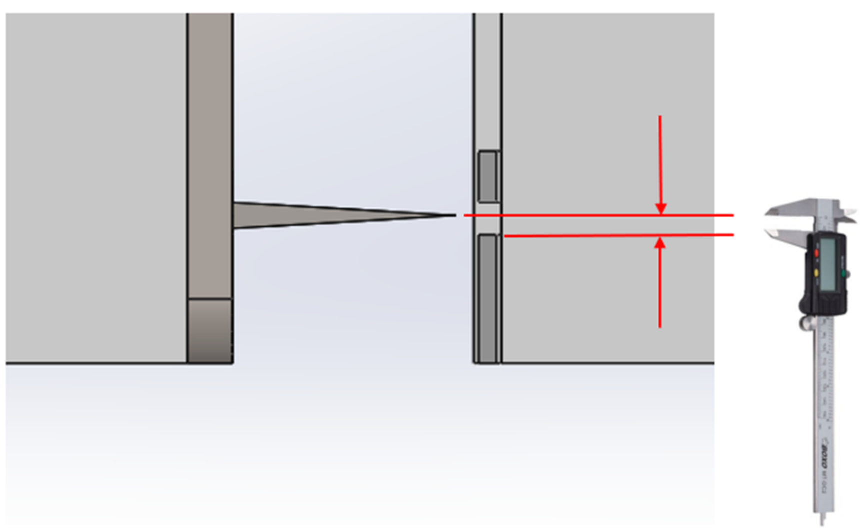
Scope: The weed cutting function which involves the entire scooper assembly

Equipment:
- Calipers or tape measure
- Voltage supply for Jetson Nano and motors
- Scooper head

Diagram of Apparatus:
- Front view of the scooper



- Close up view:



Hazards: Blades will be included with the scooper head halves so there are risks of cuts and pinch points.

PPE Requirements: Gloves

Facility: The test can be conducted in the High Bay in Bonderson where our storage room is located.

Procedure:
1. Zero the digital caliper by closing it completely and pressing the "zero" button.
2. Run motor program.
3. Take the distance measurement at the blade-slot interface and the horizontal distance between the bases of the scooper with the caliper.
4. Repeat 10 times.

Results:
Each time the scooper is closed, the distance from the blade point to the bottom of the slot will be measured at the horizontal distance of the scooper halves along with vertical blade distance and recorded on the table. We will get statistical uncertainty from our measured distances and reading uncertainty from the caliper or tape measure precision.
Data analysis equations/spreadsheet with uncertainty:

$$u_{tot} = [{u_{reading}}^2 + {u_{stat}}^2]^{1/2}$$

Area created from the vertical blade distance and horizontal distance from the scooper halves' base distance to determine if blade is level:

$$A_{displacement} = h_{blade} * l_{base}$$

Propagated uncertainty:

$$u_{A_{displacement}} = \pm\sqrt{\left(\frac{\partial A_{displacement}}{\partial h_{blade}} u_{h_{blade}}\right)^2 + \left(\frac{\partial A_{displacement}}{\partial l_{base}} u_{l_{base}}\right)^2}$$

Data:

Trial #	(Vertical) Blade in Slot Distance (in)	(Horizontal) Scooper Base Distance (in)
1	0.0625	0.3125
2	0	0.25
3	0.0625	0.375
4	0	0.25
5	0	0.25
6	0	0.25
7	0	0.25
8	0	0.43
9	0	0.375
10	0	0.5

$U_{reading}$= 1/32" = .03125

Test Results:

Slope of Area/Blade Height (in)	Slope of Area/Base Length (in)
0.323	0.062

Average of Blade Height (in)	Average of Base Length (in)
0.016	0.296

Root Sum Square of Blade Height (in)	Root Sum Square of Base Length (in)
0.002197	0.0002743
0.0002441	0.002110
0.002197	0.006251
0.0002441	0.002110
0.0002441	0.002110
0.0002441	0.002110
0.0002441	0.002110
0.0002441	0.01797

Standard Deviation of Blade Height (in)	Standard Deviation of Base Length (in)
0.02893	0.07076

Blade Height (in)	Base Length (in)	Area (in^2)
0.0625	0.3125	0.01953
0	0.25	0
0.0625	0.375	0.02344
0	0.25	0
0	0.25	0
0	0.25	0
0	0.25	0
0	0.43	0

Statistical Error of Blade Height (in)	Statistical Error of Base Length (in)
0.01023	0.02502

Reading Error (in)	Total Uncertainty (in)
0.03125	0.04132

Propagated Uncertainty (in)
0.003658

Average blade in slot distance = 0.0125 in
Average scooper base distance = 0.324 in

Test Date(s): 5/30/2023

Performed By:
Jackie Chen
Sachi Hiji
Ayush Kakkanat
William Ta

Test Procedure for Operating Time per Charge

Test Name: Test for Determining Operating Time

Purpose: The purpose of this test is to evaluate whether our prototype meets the operating time requirement. According to our project specifications, the goal was to reach an operating time of 3 hours. This test will determine if our prototype passes or fails this requirement.

Scope: This testing is important for basic robot functionality of moving and extracting weeds. Understanding the operating time is useful for determining realistic robot usage, design scalability, and whether modifications need to be made to the electrical or mechanical systems.

Equipment:

- Complete robot prototype and controller
- Robot battery and charger
- Stopwatch

Hazards:

The hazards present in this test are the same hazards present in general robot operation. Safety precautions will be in place for both electrical and mechanical safety, including things such as chain guards, a chain guard, electrical fuses, and electrical insulation. Additionally, adequate signage will be used to ensure operator safety in regard to the shredder and cutting scooper.

PPE Requirements:

- Ensure that all loose clothing and long hair is tied up
- Safety goggles

Facility: The test will be conducted at the Camp Arnaz property or other field equivalent.

Procedure:

1. Charge battery to full capacity.
2. Note the actual state of charge (SOC) of the battery at the beginning of the test.
3. Manually drive the robot around varying terrain for 30 minutes.
4. After 30 minutes of operation, note the battery SOC.

Results:

Time	Charge
Starting SOC	100%
SOC after 30 minutes of driving	98.6%
Total extrapolated operating time	N/A (see results paragraph)

Test Date(s): 5/30/2023

Test Results:
Not Complete.
We were unable to complete this test in a way that would indicate whether the battery will last through 3 hours of full operation. On the day we attempted to perform this test, there were still several issues with the controls, one of which being that the motor drivers were only outputting 2-7 volts. To operate the motors at full power, the motor drivers need to be able to output 24 volts. We manually drove the robot around for 30 minutes and the battery dropped from a SOC of 100% to a SOC of 98.6%. This is a very small SOC decrease for 30 minutes of operation, but no conclusions can really be drawn from this test since the motors were unable to operate at full capacity.

Performed By:
Claire Franz
Lewis Kanagy
Carlos Rodriguez
Kaela Stern

Test Procedure for Waterproofing

Test Name: Test for Waterproofing

Purpose: The purpose of this test is to evaluate the waterproofing capabilities of the 3D printed boxes and compartments used to contain the robot's electrical components and other sensitive components. These components include the battery, motors, motor controllers, and other electronics that Team Weed is responsible for.

Scope: This test is important to ensure that the robot's internal components will be able to withstand exposure to water due to rain or cleaning processes. The results of the test will allow us to redesign the boxes if necessary to prevent water damage to the electronics.

Equipment:
- Assembled Robot.
- Garden hose or other water source

Hazards: N/A

PPE Requirements: N/A

Facility: The test will be conducted on Cal Poly's campus.

Procedure:
1. Close component boxes and place them right-side-up on the ground
2. Remove connection to battery terminals and place plastic terminal caps on battery terminals.
3. Remove Jetson Nano MCU.
4. Using a garden hose, lightly spray the robot from all sides to simulate rainfall or spray from cleaning
5. Open the boxes, being careful not to drip any water from the outside of the box into the inside.
6. Examine all areas listed in the table below to determine t water was able to penetrate sensitive areas.

Results:
If there is any amount of water inside any of the component boxes, that box will fail the test. If the inside of a box is completely dry, that box will pass the test.

Data:

Component	Was there water in this area? (Y/N)
Drive Motor Windings	Inconclusive
Water on Base Plate	
Battery Terminal	
Worm Gearbox	
Jetson Nano MCU	
Motor Drivers	
Fuse Box	

Test Date(s): 5/25/2023

Test Results:
Unfortunately, we were unable to perform this test. Due to Team Scouts needing to troubleshoot electrical/control issues, they did not have an electrical box or waterproof housing for their components. Performing this test would have leaked water on everything and caused significant damage.

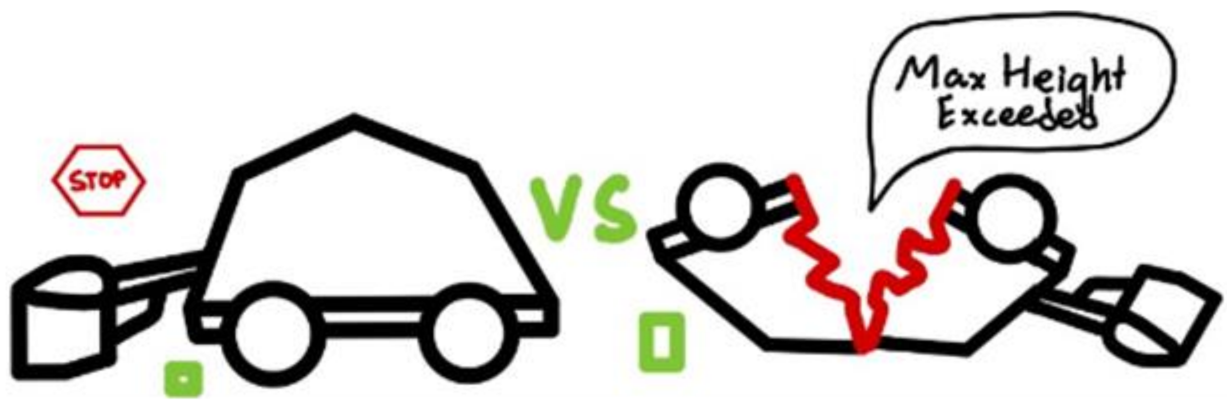
Performed By:
Claire Franz
Lewis Kanagy
Carlos Rodriguez
Kaela Stern

Test Procedure for Obstacle Stability

Test Name: Obstacle Stability Test

Purpose: The purpose of this test is to evaluate the maximum obstacle our robot can drive over. This will give us insight into our chassis load distribution and the strength of our motors so that modifications can be made if needed.

Scope: This testing is important due to the outdoor nature of our project. This will guild our moving capabilities to prevent future damage to any of our components, testing the height of obstacles our robot can handle without becoming unstable with a 2in threshold goal.



Equipment:
- Camera
- Different sized obstacles (rocks) [0.5, 1, 1.5, 2, 2.5] in.
- Measuring Tape

Hazards: N/A

PPE Requirements: Safety Glasses

Facility: Our test will be conducted on Cal Poly's campus.

Procedure:
1. Prepare a path for robot.
2. Place an obstacle in front of one wheel starting with the smallest.
3. Place the connect the power supply to the drive motors.
4. Place the power supply on the robot.
5. Plug the power supply into a wall outlet.
6. Have someone ready to unplug power from the power supply.
7. Slowly increase the voltage to the motors so that the robot drives forward over the obstacle while examining all components to ensure stability.
8. Repeat steps 1-2 and 6-7 for each sized obstacle and record the results below.

Results:
The results will be a spreadsheet with varying fixtures and observations. Each fixture should have observations accounted for, with the biggest fixture having three additional observations.

Data:

Fixture Size (in)	Observations
0.5	PASSED: Robot went over a 0.5 in obstacle with relative ease.
1	PASSED: Robot went over 1 in obstacle smoothly.
1.5	PASSED: Robot went over obstacle. There was significant vertical movement, but the robot continued forward.
2	FAILED: Robot did not go over obstacle. Unfortunately, the electrical system was not complete for full testing, so this was run with just the power supply. This test would need to be rerun with the battery supplying full power to the motors. Due to electrical issues, the motors were not able to get the necessary voltage from the battery, so this test could not be rerun.
2.5	FAILED: Robot did not have enough power from power supply to go over obstacle. Same reasons as above.
Max Height: 1.5 in	Robot was able to traverse unstable ground, but it did not have maximum power because this test was conducted with the power supply, which was not supplying max current or full voltage to the motors.

Test Date(s): 5/11/2023

Performed By:
Claire Franz
Lewis Kanagy
Carlos Rodriguez
Kaela Stern

Test Procedure for Charge Time

Purpose: Determine if the battery can charge in under 6 hours.

Location: Perform this test in team F75's reserved Room 108 in Benderson Project Center

Equipment:
- 24V 220Ah Robot battery.
- Robot Battery Charger.
- Access to 120V Standard Power Outlet.
- Stopwatch.
- Charge Indicator or multimeter.

Hazards: The person(s) conducting this test will be exposed to a 24V battery. Voltages this low are generally safe but creating a short from the positive and negative battery terminals could cause sparking or electric shock and should be avoided.

PPE Requirements: N/A

Safety Procedures:
1. Ensure that battery terminals are clear of any loose wires or any unwanted contacts that could create a short.
2. Ensure that battery charger leads will not contact any loose wires or any unwanted contacts that could create a short.

Test Procedure:
1. Note the battery's state of charge (SOC).
2. Prepare a stopwatch or note the current time.
3. Plug the battery charger into a standard 120V power outlet.
4. Switch the main battery connector on the robot to the "disconnected" position.
5. Connect the positive lead of the battery charger to the robot battery.
6. Connect the negative lead of the battery charger to the robot battery.
7. Start the stopwatch.
8. Wait 20 minutes to ensure that the battery is charging properly and that there are no thermal issues.
9. Wait an additional 5 hours and 40 minutes until the battery has been charging for a total of 6 hours.
10. Check to ensure that the battery is fully charged.
11. If the battery is fully charged, move to step__
12. If the battery is not fully charged, wait until the battery is fully charged and note the time when the battery reaches full charge.
13. Once the battery is fully charged, Disconnect the negative lead of the battery charger from the robot battery.
14. Disconnect the positive lead of the battery charger from the robot battery.
15. Unplug the battery charger from the 120V outlet.
16. The test is complete.

Data:

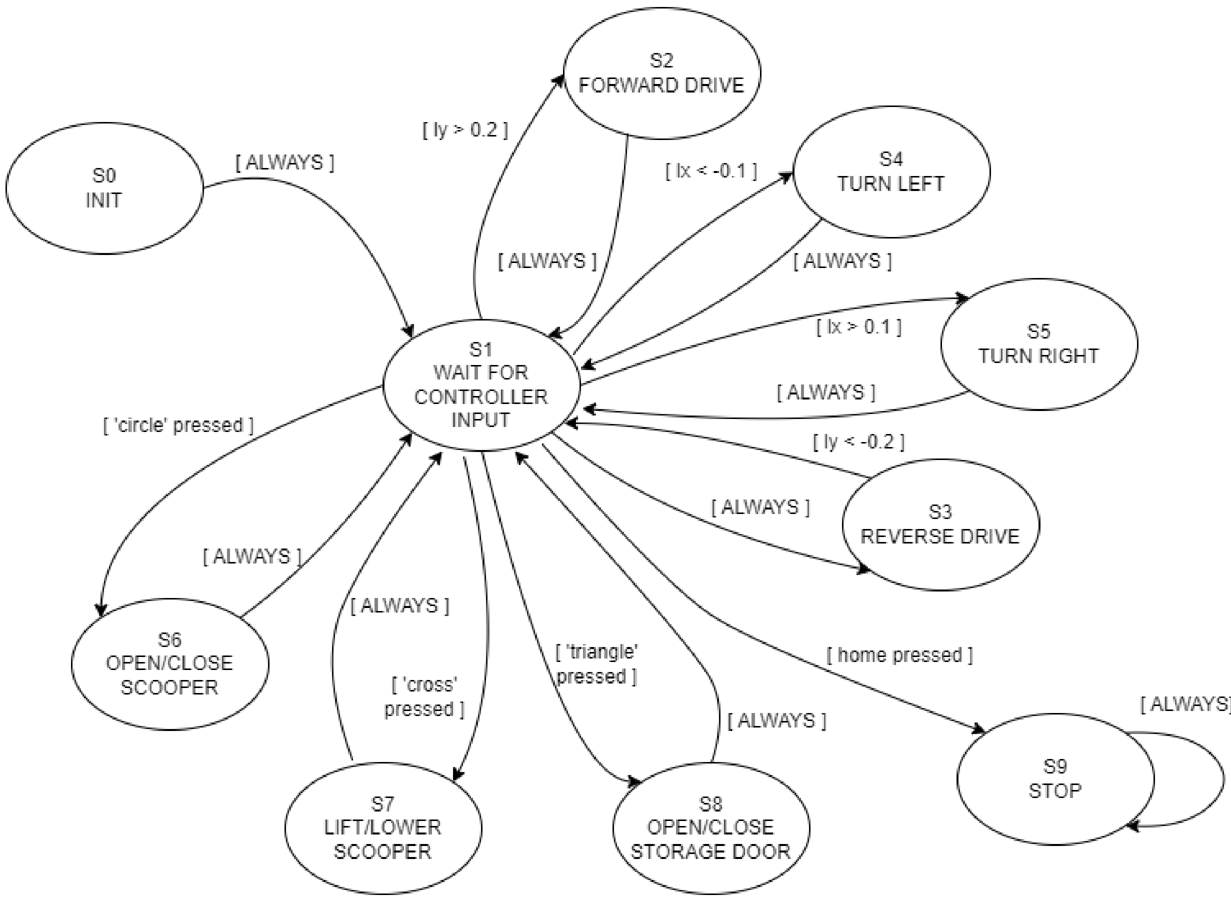
Charge	Time
Initial SOC	Inconclusive
Time to Charge from Initial SOC	
Extrapolated Time to charge from zero	

Test Date(s): 5/11/2023

Test Results:
We were unable to discharge the battery safely (via the drive motors) due to the issues mentioned in the operating time test. Since the battery was not discharged any significant amount, we did not feel that we could draw any meaningful conclusions from charge time since we would be charging the battery from 98.6% to 100%.

Performed By:
Claire Franz
Lewis Kanagy
Carlos Rodriguez
Kaela Stern

Appendix G – Finite State Machine



Appendix H – Commented Python Code

For better visibility and documentation, view the generated GitHub page for this project:
https://bobbybuttins.github.io/WEEDROBOT/turtlelord_8py_source.html

```
'''!
@turtlelord.py
Cal Poly Mechanical Engineering Senior Project - WEED ROBOT

@author Jackie Chen
@date   September 2022 - June 2023
'''

import time
import busio
import board
import adafruit_pca9685
import digitalio
from adafruit_servokit import ServoKit
from approxeng.input.selectbinder import ControllerResource
from approxeng.input.switch import SwitchJoyConLeft
from approxeng.input.dualshock4 import DualShock4
from approxeng.input.controllers import find_matching_controllers,
ControllerRequirement
global u

# Set channels to the number of servo channels on the kit
# For the PCA9685 driver, we have 16 channels
# If using bus 0 (pins 26/27), will need to create bus object like this:
i2c_bus0 = busio.I2C(board.SCL_1, board.SDA_1)
# and include it in the parameters for ServoKit: i2c=i2c_bus0
kit = ServoKit(channels=16, i2c=i2c_bus0)
# Don't need to setup bus if SCL and SDA are connected to pins 3/5
# which are for bus 1 (default)

# Setup bus for DC motor control through PCA9685 driver
i2c = busio.I2C(board.SCL_1, board.SDA_1)
pca = adafruit_pca9685.PCA9685(i2c_bus0)
# Set frequency for board
pca.frequency = 60
# Specify channels for each DC motor
# Channel 4 for left wheel, channel 5 for right wheel, channel 6 for shredder
# More channels to be added later for LED control
l_wheel = pca.channels[4]
l_wheel_r = pca.channels[5]
r_wheel = pca.channels[6]
r_wheel_r = pca.channels[7]
shred = pca.channels[8]
init_LED = pca.channels[9]
# Setup for GPIO pins on the Jetson 40-pin expansion header
# Used to set the enable pins on the DC motor for forward and reverse drive
# Jetson pin 12
l_en = digitalio.DigitalInOut(board.D18)
l_en.direction = digitalio.Direction.OUTPUT
l_en.value = True
# pin 11
lrev_en = digitalio.DigitalInOut(board.D17)
```

```
lrev_en.direction = digitalio.Direction.OUTPUT
lrev_en.value = True
# pin 22
r_en = digitalio.DigitalInOut(board.D25)
r_en.direction = digitalio.Direction.OUTPUT
r_en.value = True
# pin 21
rrev_en = digitalio.DigitalInOut(board.D9)
rrev_en.direction = digitalio.Direction.OUTPUT
rrev_en.value = True
# OE pin on PCA9685 for disabling all PWM outputs (Low/False = On, High/True
= Off)
OE = digitalio.DigitalInOut(board.D19)
OE.direction = digitalio.Direction.OUTPUT
OE.value = True

def move_left(angle):
    '''!
    @brief Controls the rotation of the left scooper servo.
    @param angle The desired angle for the left servo (int), from 0 to 180.
    '''
    kit.servo[0].angle = angle

def move_right(angle):
    '''!
    @brief Controls the rotation of the right scooper servo.
    @param angle The desired angle for the right servo (int), from 0 to 180.
    '''
    kit.servo[1].angle = angle

def move_transport(angle):
    '''!
    @brief Controls the rotation of the lifting servo.
    @param angle The desired angle of the lifting servo (int), from 0 to 360.
    '''
    kit.servo[2].angle = angle

def move_shreddoor(angle):
    '''!
    @brief Controls the rotation of the shredder door.
    @param angle The desired angle of the door servo (int), from 0 to 180.
    '''
    kit.servo[3].angle = angle

def ldc(pwm):
    '''!
    @brief Controls the forward duty cycle of the left DC motor.
    @param pwm The desired pwm duty cycle (int), from 0x0000 to 0xFFFF.
    '''
    l_wheel.duty_cycle = pwm

def ldc_r(pwm):
    '''!
    @brief Controls the reverse duty cycle of the left DC motor.
    @param pwm The desired pwm duty cycle (int), from 0x0000 to 0xFFFF.
    '''
    l_wheel_r.duty_cycle = pwm
```

```
def rdc(pwm):
    '''!
    @brief Controls the forward duty cycle of the right DC motor.
    @param pwm The desired pwm duty cycle (int), from 0x0000 to 0xFFFF.
    '''
    r_wheel.duty_cycle = pwm

def rdc_r(pwm):
    '''!
    @brief Controls the reverse duty cycle of the right DC motor.
    @param pwm The desired pwm duty cycle (int), from 0x0000 to 0xFFFF.
    '''
    r_wheel_r.duty_cycle = pwm

def estop():
    '''!
    @brief Emergency stop command. Disables the enable pins for the DC motors
and stops operation of the servos.
           Purely logical stop that locks out controller input. Does not cut
off the power to any of the components.
    '''
    # bring all DC motors to a stop by setting pwm to 0 and disabling enable
pins
    ldc(0x0000)
    ldc_r(0x0000)
    rdc(0x0000)
    rdc_r(0x0000)
    l_en.value = False
    lrev_en.value = False
    r_en.value = False
    rrev_en.value = False

    # stop the transportation arms and hold them in place
    global u
    move_transport(u)
    # turn off the "STATUS" LED
    init_LED.duty_cycle = 0x0000
    OE.value = True
    print('EMERGENCY STOP')
    exit()

def new_input():
    '''!
    @brief Detects and reads new inputs from the controller. Changes the
speed ratio between the left
           and right motors depending on how far horizontally the left
analogue stick is pushed.
    '''
    global lx
    global ly
    global turn_ratio
    global turn_ratio_r
    lx = joystick['lx']
    ly = joystick['ly']
    # turn ratios for skid steering (turns by one side of the wheels moving
faster than the other)
```

```
    # pwm ratio for forward drive
    turn_ratio = 1 + 2*abs(lx)
    # pwm ratio for reverse drive
    turn_ratio_r = 2 - 0.5*abs(lx)

if __name__ == "__main__":
    try:
        print("Initializing motor default positions...")
        # Change PWM range for all the servos to get the full range of motion
        kit.servo[0].set_pulse_width_range(1000, 3000)
        kit.servo[1].set_pulse_width_range(1000, 3000)
        kit.servo[2].set_pulse_width_range(1000, 4000)
        kit.servo[3].set_pulse_width_range(1000, 3000)

        # Set starting positions for the scooper servos
        OE.value = False
        move_left(60)
        move_right(40)
        scoop = "closed"

        # Set range of motion for DOCYKE servo to 360 degrees (default is 180
degrees)
        kit.servo[2].actuation_range = 360
        u = 5
        lift = "down"

        # Set starting position for shredder door servo
        move_shreddoor(120)
        shreddoor = "closed"
        shred_weed = "off"
        print("Initialized")

        while True:
            with ControllerResource() as joystick:
                print(type(joystick).__name__)
                print("Waiting for input from controller")
                # turn the "STATUS" LED on to convey that the robot will
start taking controller inputs
                init_LED.duty_cycle = 0xFFFF
                while joystick.connected:
                    # read controller inputs
                    new_input()

                    # duty cycle calculations for driving
                    # calculation dependent on y position (for angles of 45-
135 degrees on the left analogue stick)
                    duty_y = int(round(abs(ly)*100)*0x028F)
                    # same calculation as above but divided by the turn ratio
to offset the wheel speeds for turning
                    duty_ty = int(round(abs(ly)*100)*0x028F//turn_ratio)
                    # calculation dependent on the x position (for angles of
0-45 and 135-180 degrees on the left analogue stick)
                    duty_x = int(round(abs(lx)*100)*0x028F)
                    # same calculation as above but divided by the turn ratio
to offset the wheel speeds for turning
                    duty_rev = int(round(abs(lx)*100)*0x028F//turn_ratio_r)
```

```
                    # ROBOT MOVEMENT CONTROL
                    # forward drive (no turn ratio, full drive)
                    if ly > 0.2:
                        if lx > -0.2 and lx < 0.2:
                            # disable reverse pwm on both sides
                            # FORWARD AND REVERSE PWM MUST NOT BE ON AT THE
SAME TIME OR THE MOTOR DRIVERS WILL BURN OUT
                            ldc_r(0x0000)
                            rdc_r(0x0000)
                            # forward pwms enabled
                            ldc(duty_y)
                            print(duty_y)
                            rdc(duty_y)
                            print(duty_y)
                            # wait for new inputs
                            new_input()
                            print('forward')
                            # check for estop (for future implementation use
threading instead to listen to the button press?)
                            presses = joystick.check_presses()
                            if presses['home']:
                                estop()

                        # skid turn left (left wheels at reduced/reverse
drive, right wheels at full speed)
                        if lx < -0.2:
                            # if the left stick angle is within 90-135
degrees
                            if ly > abs(lx):
                                # disable reverse pwm on both sides
                                # FORWARD AND REVERSE PWM MUST NOT BE ON AT
THE SAME TIME OR THE MOTOR DRIVERS WILL BURN OUT
                                ldc_r(0x0000)
                                rdc_r(0x0000)
                                # forward pwms enabled
                                ldc(duty_ty)
                                print(-duty_ty)
                                rdc(duty_y)
                                print(duty_y)
                                # wait for new inputs
                                new_input()
                                print('left')
                                # check for estop (for future implementation
use threading instead to listen to the button press?)
                                presses = joystick.check_presses()
                                if presses['home']:
                                    estop()
                            # if the left stick is within 135-180 degrees
(right wheel will drive forward and left will reverse)
                            elif abs(lx) >= ly:
                                # disable forward pwm on left and reverse pwm
on right
                                # FORWARD AND REVERSE PWM MUST NOT BE ON AT
THE SAME TIME OR THE MOTOR DRIVERS WILL BURN OUT
                                ldc(0x0000)
                                rdc_r(0x0000)
```

```
                                # reverse left pwm and foward right pwm
enabled
                                ldc_r(duty_rev)
                                print(-duty_rev)
                                rdc(duty_x)
                                print(duty_x)
                                # wait for new inputs
                                new_input()
                                print('left-rev')
                                # check for estop
                                presses = joystick.check_presses()
                                if presses['home']:
                                    estop()

                        # skid turn right (right wheels at reduced/reverse
drive, left wheels at full speed)
                        elif lx > 0.2:
                            # if the left stick angle is within 45-90 degrees
                            if ly > lx:
                                # disable reverse pwm on both sides
                                # FORWARD AND REVERSE PWM MUST NOT BE ON AT
THE SAME TIME OR THE MOTOR DRIVERS WILL BURN OUT
                                ldc_r(0x0000)
                                rdc_r(0x0000)
                                # forward pwms enabled
                                ldc(duty_y)
                                print(duty_y)
                                rdc(duty_ty)
                                print(duty_ty)
                                # wait for new inputs
                                new_input()
                                print('right')
                                # check for estop
                                presses = joystick.check_presses()
                                if presses['home']:
                                    estop()
                            # if the left stick angle is within 0-45 degrees
                            elif ly <= lx:
                                # disable reverse pwm on left side and
forward pwm on right
                                # FORWARD AND REVERSE PWM MUST NOT BE ON AT
THE SAME TIME OR THE MOTOR DRIVERS WILL BURN OUT
                                ldc_r(0x0000)
                                rdc(0x0000)
                                # foward left pwm and reverse right pwm
enabled
                                ldc(duty_x)
                                print(duty_x)
                                rdc_r(duty_rev)
                                print(duty_rev)
                                # wait for new inputs
                                new_input()
                                print('right-rev')
                                # check for estop
                                presses = joystick.check_presses()
                                if presses['home']:
                                    estop()
```

```
                    # reverse drive (no turn ratio, full reverse drive)
                    # turning in reverse is effectively the same as turning
in forward drive
                    # to back up at an angle, turn in forward drive and then
go straight reverse drive (not smooth, will have to optimize later)
                    elif ly < -0.2:
                        if lx > -0.2 and lx < 0.2:
                            # disable forward pwm on both sides
                            # FORWARD AND REVERSE PWM MUST NOT BE ON AT THE
SAME TIME OR THE MOTOR DRIVERS WILL BURN OUT
                            ldc(0x0000)
                            rdc(0x0000)
                            # reverse pwms enabled
                            ldc_r(duty_y)
                            print(-duty_y)
                            rdc_r(duty_y)
                            # wait for new inputs
                            new_input()
                            print('reverse')
                            # check for estop
                            presses = joystick.check_presses()
                            if presses['home']:
                                estop()

                    # idle state when the robot is not driving
                    # the conditions for x and y are offset from 0 due to
input noise from the cheap controller
                    else:
                        while ly > -0.2 and ly < 0.2:
                            # set all pwms to 0 to stop the motors
                            ldc(0x0000)
                            ldc_r(0x0000)
                            rdc(0x0000)
                            rdc_r(0x0000)
                            # continue to wait for new inputs
                            new_input()
                            presses = joystick.check_presses()

                            # OPEN/CLOSE SCOOPER
                            if presses['circle']:
                                # starts closed, will cycle between closed
and open on button press depending on the previous state
                                if scoop == "closed":
                                    print("Opening scooper")
                                    time.sleep(0.5)
                                    presses = joystick.check_presses()
                                    # check for estop
                                    if presses['home']:
                                        estop()
                                    move_left(77)
                                    move_right(15)
                                    scoop = "open"
                                else:
                                    print("Closing scooper")
                                    time.sleep(0.5)
                                    presses = joystick.check_presses()
```

```
                                # check for estop
                                if presses['home']:
                                    estop()
                                move_left(45)
                                move_right(47)
                                scoop = "closed"

                        # MOVE SCOOPER UP/DOWN
                        elif presses['cross']:
                            # starts at down position, will cycle between
lowered and raised positions on button press depending on the previous state
                            if lift == "down":
                                print("Lifting scooper")
                                u = 30
                                time.sleep(0.5)
                                # moves at small intervals to avoid a
sudden swing that could damage/injure surroundings and/or the scooper arms
                                while u <= 345:
                                    u += 15
                                    move_transport(u)
                                    presses = joystick.check_presses()
                                    # check for estop
                                    if presses['home']:
                                        estop()
                                    time.sleep(0.3)
                                lift = "up"
                            else:
                                print("Lowering scooper")
                                u = 345
                                time.sleep(0.5)
                                # moves at small intervals to avoid a
sudden swing that could damage/injure surroundings and/or the scooper arms
                                while u > 30:
                                    move_transport(u)
                                    u -= 15
                                    presses = joystick.check_presses()
                                    # check for estop
                                    if presses['home']:
                                        estop()
                                    time.sleep(0.3)
                                lift = "down"

                        # OPEN/CLOSE SHREDDER DOOR
                        elif presses['triangle']:
                            # starts closed, will cycle between closed
and open on button press depending on the previous state
                            if shreddoor == "closed":
                                print("Opening shredder door")
                                time.sleep(0.5)
                                presses = joystick.check_presses()
                                # check for estop
                                if presses['home']:
                                    estop()
                                move_shreddoor(10)
                                shreddoor = "open"
                            else:
                                print("Closing shredder door")
```

```
                                time.sleep(0.5)
                                presses = joystick.check_presses()
                                # check for estop
                                if presses['home']:
                                    estop()
                                move_shreddoor(120)
                                shreddoor = "closed"

                        # TURN ON/OFF SHREDDER (not implemented)
                        # elif presses['square']:
                        #     if shred_weed == "off":
                        #         print("Turning on shredder")
                        #         shred.duty_cycle = 0x0100
                        #         shred_weed = "on"
                        #     else:
                        #         print("Turning off shredder")
                        #         shred.duty_cycle = 0x0000
                        #         shred_weed = "off"

                        elif presses['home']:
                            estop()

                # if controller is disconnected, stop all motors
                # bring all DC motors to a stop by setting pwm to 0 and
disabling enable pins
                ldc(0x0000)
                ldc_r(0x0000)
                rdc(0x0000)
                rdc_r(0x0000)
                l_en.value = False
                lrev_en.value = False
                r_en.value = False
                rrev_en.value = False

                # stop the transportation arms and hold them in place
                move_transport(u)
                init_LED.duty_cycle = 0x0000
                OE.value = True
                print('Controller disconnected')
                exit()

    # Keyboard Interrupt exception for testing and debugging code
    except KeyboardInterrupt:
        print("Operation terminated.")
        ldc(0x0000)
        ldc_r(0x0000)
        rdc(0x0000)
        rdc_r(0x0000)
        l_en.value = False
        lrev_en.value = False
        r_en.value = False
        rrev_en.value = False

        # stop the transportation arms and hold them in place
        move_transport(u)
        init_LED.duty_cycle = 0x0000
        OE.value = True
```