Final Report

Team #11
Robo-Weeder
4/8/2016

Submitted to: Dr. Gupta (Faculty Advisor)
Jeff Phipps (Sponsor)

Authors: Steven Miller (ME)
Christopher Murphy (ME)
Arriana Nwodu (ME)
Aquiles Ciron (EE)
Steven Williamson (EE)
Zhang Xiang (ME)

Table of Contents

Table of Figures

Table of Tables

Abstract

Team 11's Robo-Weeder senior design project is sponsored by Jeff Phipps, the owner of Orchard Pond Organics. Orchard Pond Organics as well as all organic farms worldwide have a pressing need for assistance in weed control. Due to the conventional, labor intensive methods of weed removal, the cost of highly nutritious crops is high rendering it inaccessible to low income families. The purpose of the Robo-Weeder is to design and create a robotic system that will satisfy this need by using an auger style shearing mechanism to remove weeds on the seedbeds of planted vegetables. After many iterations, and guidance from the sponsor and faculty advisors Dr Gupta and Dr Hooker, the final design has been approved and the Robo-Weeder has been built. Controlled testing of each subsystem has been completed and the Robo-Weeder is working as intended. Team 11 is now transitioning into the final stages of the project with coding optimization as well as field testing.

Acknowledgements

Team 11 would first like to thank Dr. Chiang Shih, the Mechanical Engineering Chair as well as our Faculty Advisors, Professors Nikhil Gupta and Jerris Hooker. They've been invaluable in the development and progress of the Robo-Weeder senior design project. They've guided us not only on the technical aspects, but also in team morale and because of that we learned to perform in an optimal and cohesive manner. We would also like to thank our team's sponsor Mr. Jeff Phipps of Orchard Pond Organics. Due to his vision of providing more nutritious food at a lower cost, thousands if not millions of low income families can have the option of a healthier diet. Also, we would like to thank the FAMU-FSU College of Engineering Machine shop and Mr. Keith Larson for aiding in the fabrication and assembly processes. Finally team 11 would like to the senior design teaching assistants Yuze Liu, Jhamal Holliday, Andrew Panek and Obie Abakporo as they have been a constant resource throughout the design process providing valuable feedback.

1. Introduction

As a continuation of the previous year's Robotic Weeding Harvester, Team 11 has been tasked with improving upon or redesigning the final design prototype that was generated. Team 11's project, appropriately named "Robo-Weeder" has a primary function of aiding in the removal of unwanted weeds that plague the seed beds of Jeff Phipps' organic farm, as well as organic farms worldwide.

1.1 Problem Statement

Organic farming techniques rely heavily on labor intensive methods which create large production costs for high nutrition organic produce.

1.2 Design Requirements & Constraints

The sponsor gave several parameters that Team 11's design must be confined in. The first constraint was the robotic system shouldn't disrupt more than 1" of the soil. This will ensure there is a reduction in weed removal without destroying the adjacent crops. The second constraint given was the robotic system shouldn't use a till-like cutting implement, and must have the option to have interchangeable weeding implements. Tilling results in soil erosion, degrading the helpful nitrogen content of the soil which lowers the nutritional value of organic crops. The final constraint given by our sponsor was the robotic system must be remotely operated. Last year's Team 11: The Robotic Harvester was given too many design constraints and although their system was fully autonomous their shearing implement incorporated a tilling machine that not only contributes to soil erosion but causes a moment whenever there is an obstruction blocking the tiller. Professor Gupta, one of the team's faculty advisor gave another design constraint that the robotic system must be splashproof to properly house all electrical components in a safe and durable manner. For further clarification, the design constraints for the mechanism are listed below:

- Function in "No-Till" fashion
- Must be mobile
- Must be remotely operated
- Must use Auger style shearing mechanism
- Not disturb more than one inch in depth of soil
- Must be tolerant to minimal water (splash proof).

1.3 Goal Statement and Objectives

Thus far, the established goal for Team 11 is to develop a 'proof of concept' robotic system that will enhance the production of organic crops. The final design came about by meeting several objectives. In no particular order the design objectives Team 11 established are: the robotic system must be remotely operated, it must have independent front and rear steering, it must have the option to interchange weeding implements, it must be splashproof and finally the design must be robust.

Robo-Weeder will be to help facilitate the production of crops with high nutritional value. The Robo-Weeder will: be remotely operated, remove weeds through the application of a shearing force and also have an interchangeable battery source. The primary challenge that will be solved by team 11 is the amount of force to apply to the seedbed to achieve the desired amount of shear. The team dynamics include four mechanical engineers and two electrical engineers. The project is presented by the FAMU-FSU College of Engineering Department of Mechanical Engineering and is sponsored by Mr. Jeff Phipps, of the Orchard Pond Organics farm. The project is advised by Dr. Gupta and Dr. Hooker.

2. Background Research and Literature Review

2.1 Fundamentals - Soil Analysis

The main function of the Robo-Weeder project is to develop a robotic platform that is capable of removing weeds from seedbeds on an organic farm. The removal of the weeds is accomplished by applying a shear force to the surface of the soil in such a manner as to disrupt the root systems of the weeds.

In order for Team 11 to accomplish the objective of shearing the soil to remove weeds, it is important to understand the force that needs to be applied to the soil to achieve the desired soil shear. To understand these forces, Team 11 consulted with the Civil Engineering department to better understand the field of Soil Mechanics. During the consultations, Team 11 spoke with Adjunct Professor Sal Arnaldo, P.E. as well as Professor and Chairman Kamal Tawfiq, Ph.D., P.E. who recommended soil testing to understand the exact forces needed during this project.

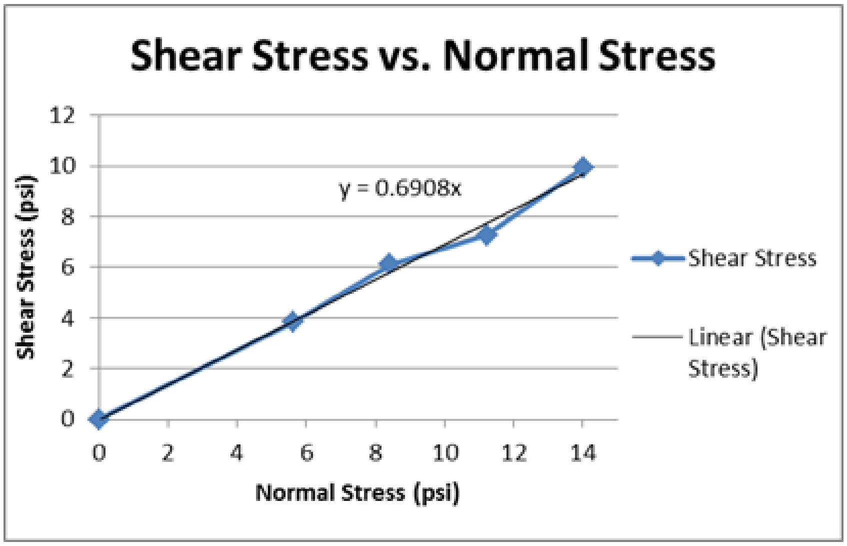


Figure 1: Shear Stress data collected from soil testing.

At the conclusion of the soil testing team 11 was able to use **Figure 1** above to estimate the required force to apply. To estimate the stresses needed to achieve soil shear the design team first needed to estimate the footprint size of the shearing mechanism, as well as the weight of the robot to determine the normal stress on the soil. From the known normal stress, Team 11 can use the relationship displayed by the linear trend line to estimate the shear stress that would need to be supplied by the Robo-Weeder.

2.2 Conventional Farming Research

Current methods of farming use technologically advanced cultivating tools and genetically modified crops. This method is the currently used to maximize the possible yield of the crops without considering the nutritional value. However, these processes are not only destructive to the environment, but hurts crops by destroying microorganisms along with ground insects that would further contribute to the development of high yielding soil. Another flaw in conventional farming techniques is that there's large scale production of one single crop on a parcel of land also known as monoculture. This cultivation method is the main reason synthetic fertilizers and pesticides are used since there is a dramatic decrease in crop diversity on a land plot, eliminating the natural biological controls that would maintain pest levels, disease, and soil degradation.[1]

Another negative attribute of using pesticides, herbicides, and insecticides, is many 'pests' have already evolved and will continue to evolve to resist new developing chemicals. These pesticides, herbicides, insecticides, and fertilizers are derivatives of fossil fuels which are a limited natural resource. In addition to being a natural resource; fossil fuels also contribute to water contamination which is problematic because farms require the use of vast water irrigation systems. Currently irrigation systems extract water from reservoirs faster than they can be replenished, rapidly depleting this resource.

Due to the known fact that traditional farming leads to some serious consequences, organic farming has become a growing trend around the world. For clarification, organic farming is done without the use of any chemically derived fertilizers, pesticides, herbicides, or is grown with genetically modified organisms (GMO).[2] There are many different methods to subsidize the effect of not using traditional fertilizers, pesticides, and herbicides such as using organic compounds as fertilizer, pesticides, and herbicides. Some of these organic compounds include rotenone and pyrethrin.

Another method of organic farming is uses crops covers such as clover, a legume, to reduce unwanted weeds. Legumes also put nitrogen back in the soil once they are tilled out of the earth. That being said, legumes are natural fertilizers and promote healthy soil as well as improvements of antioxidant levels or a highly nutritious crop.[3] In an effort to combat the effects of pests while not using pesticides, organic farms do away with monoculture and diversify the crops. This variation in crops allows certain immunity to pests that would target a particular crop under monoculture practices. The final method used by organic farmers is crop rotation which enhances the quality of the soil by placing vital nutrients back into the soil.

However, the downside of organic farming is the precise removal of undesired plants that grow near crops, and pest control. There are many different ways to combat these efforts but none of which work well with monoculture. Existing weeding machines are heavy, bulky and use gasoline engines which can adversely affect the crop yield.

2.3 Need Statement

After enough background research Team 11 determined that the chassis desired must be robotic in nature, and must apply an adequate shear force to the roots of undesired plants without disturbing adjacent crops. The platform must be able to accommodate future cutting implements that the sponsor develops and must be safe, easily maintained and user friendly. In order to aid Mr. Phipps on his organic farm, an effective, reliable, and well-functioning remotely controlled platform must be created.

3. Concept Generation

3.1 House of Quality

An important tool that is at the disposal of the Robo-Weeder design team is the House of Quality. The House of Quality (HOQ) allows one to relate requirements that the customer has for the final product, or sponsor in the case of the Robo-Weeder project, to key engineering characteristics. It claimed the name House of Quality due to its graphical nature resembling a house.

Importance/Weight	Customer Requirements	Mass	Material	Durability	Stability	Strength of Components	# of Tires/Tracks	# of Motors	Operation Mode (Wired, Radio, etc…)	Battery System
4	Safe to Operate	4	20	20	40	20	4	4	4	4
1	Cost Efficient	5	10	5	5	5	10	10	5	5
5	Effective	5	25	50	5	25	25	5	25	25
5	Reliable	5	25	50	50	50	5	5	5	25
2	Simple to Operate	2	2	2	2	2	2	2	10	2
2	Interchangable Implements	20	10	10	20	10	2	2	2	2
3	Weight	30	30	30	30	30	30	30	3	15
2	Marketability	2	2	20	20	10	2	2	10	2
4	Irregular Terrains	4	4	20	40	4	40	4	4	4
Totals		**77**	**128**	**207**	**212**	**156**	**120**	**64**	**68**	**84**
Rank		**7**	**4**	**2**	**1**	**3**	**5**	**9**	**8**	**6**

Figure 2: House of Quality constructed for the Robo-Weeder project.

In the above figure, **Figure 2**, the decision matrix has been removed to show the primary data within the HOQ. The key features shown include the customer requirements, the weight or importance of the requirement, the key engineering characteristics shown in blue and the decision matrix relating each section. The conclusion of the analysis yields a ranking of the engineering characteristics and advises the design team which characteristics could be the most important when designing the final product. For the Robo-Weeder project, the House of Quality advises Durability, Stability and Strength should be considered during the design process. It should be noted that a full scale House of quality with the "Roof Section" is located in Appendix A.

3.2 Design I

During the early stages of the design process, Team 11 had general ideas of how the Robo-Weeder needed to function. They knew the robot needed a robust design as made evident from the HOQ results. They also knew that they needed advanced control over the movement of the robot to ensure the robot did not make contact with the very crops the project is trying to help. Lastly, the team knew that they needed a method to raise and lower the weeding mechanism, which was composed of two 8 inch augers in a horizontal orientation.

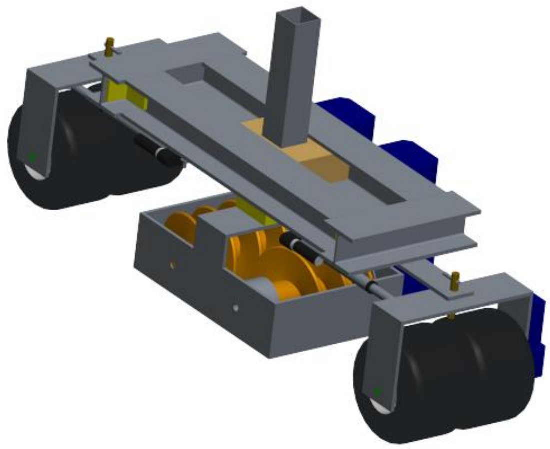


Figure 3: First rendition of the Robo-Weeder.

After consideration of all of the design features that were outlined during the research phase of the project, **Figure 3** above illustrates a general CAD of the Robo-Weeder. This particular design housed independent front and rear steering which answered the question of enhanced control. The steering system was driven by two linear actuator, one on each drive assembly. The glaring flaw with this design was the mounting location of all of the DC gear motors. All motors were on one side, which created an uneven distribution of weight.

3.3 Design II

Faced with an instability issue due to uneven weight distribution, the design team sought a way to keep the weight along the central plane of the robot. To address this, the gear motors that operated the weeding mechanism were moved onto the front and rear of the auger housing. Also, the team set out to find new drive motors that would fit between between the wheels in the front and rear. These changes proved to help with weight distribution, but proved impractical due to the increased length of the Robo-Weeder's new design.

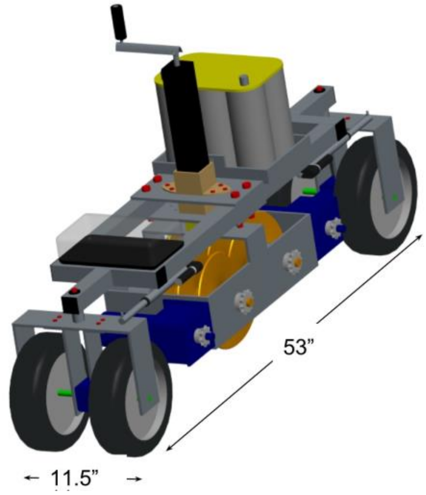


Figure 4: Robo-Weeder design 2; Redesign to address weight distribution.

When the motors that operated the weeding mechanism were moved to the front and rear of the auger housing, **Figure 4**, it forced the design team to lengthen the Robo-Weeder to allow for proper turning. With a new length to width ratio of 5:1, the design team headed back to the drawing board in an attempt to solve the instability issue that was introduced.

3.4 Design III

After consulting with faculty advisors and the team's sponsor, the design team determined that the biggest factor affecting the design of the Robo-Weeder was the size of the auger style weeding mechanism. During early meetings with the team’s sponsor, Jeff Phipps, it was suggested that an 8 inch auger system be used for the weeding mechanism. However, in the third design of the Robo-Weeder (**Figure 5)**, the design team introduced a new 4 inch auger system that allowed the chassis of the robot to be shortened. Also, due to the decrease in size of the augers, the design team was also able to locate new gear motors of much smaller size and weight. The new gear motors were a direct product of the decreased torque requirement of the new smaller augers.

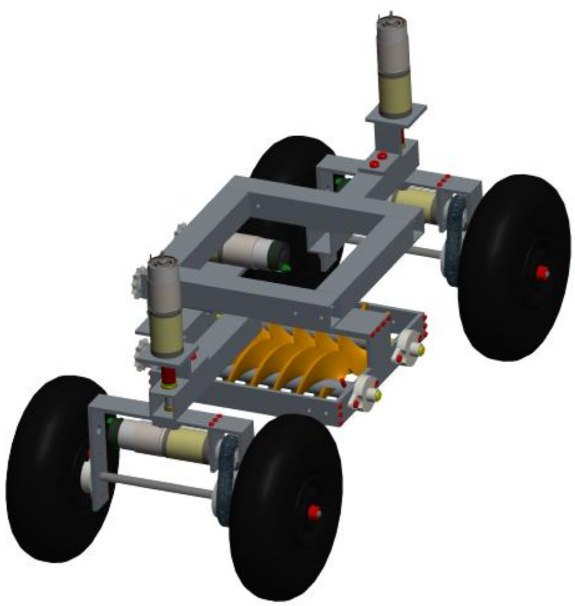


Figure 5: Third design of the Robo-Weeder with the 4 inch auger system.

During the third rendition of the Robo-Weeder, the team decided a redesign of the drive and steering subsystems was needed. During the redesign, it was suggested by the team's advisors to replace the linear actuator steering system with a gear motor design. Also, the team determined that the drive motors could be moved between the wheels of each drive assembly, further increasing stability by slightly increasing the width of the robot. It was evident at the conclusion of the third rendition that the design team had a solid platform to proceed with.

4. Final Design

4.1 Mechanical Design Concepts

The mechanical design phase of the project evolved as the team progressed through the fall semester and was solidified with a final design at the conclusion of the semester. During the spring semester, the design team further enhanced the mechanical design to adequately support all of the Robo-Weeder's functionality. Key aspects described in the following mechanical design section will include the steering and drive assemblies, auger style weeding mechanism, lift system, frame and the final Robo-Weeder assembly.

4.1.1 Steering and Drive

During meetings with the project's sponsor Jeff Phipps, Mr Phipps voiced a need for the robot to be able to be steered very accurately. This is due to the close proximity to the crops as well as the environment that the robot would be operated in. Taking this information into account, the design team created a steering and drive system, **Figure 6**, which would mirror functionality in both the front and rear of the robot.

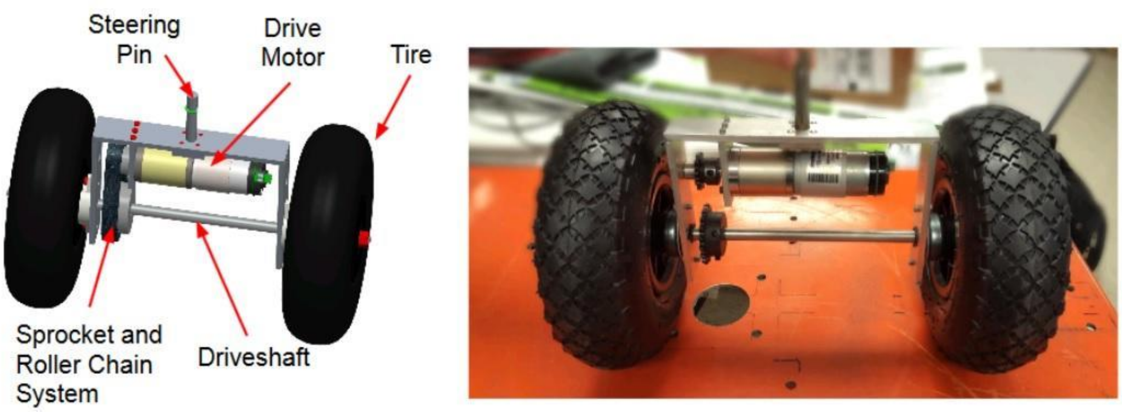
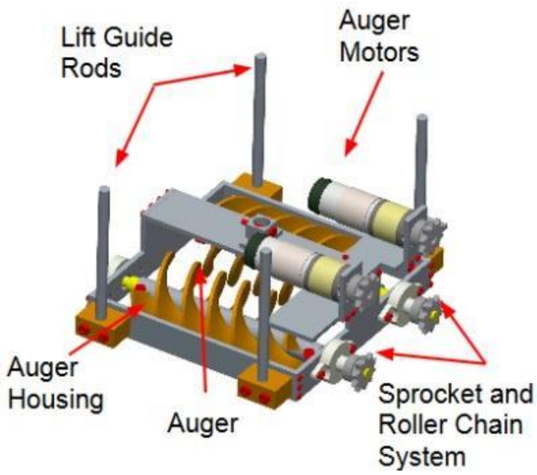


Figure 6: Steering and Drive assembly used by the Robo-Weeder.

The above figure shows the Computer Automated Drawing (CAD) of the steering and drive system with component labels while the right image shows the actual subsystem after fabrication. Having mirror steering and drive assemblies allowed for simplification during the fabrication stages but also allowed for the enhanced control that the sponsor requested as the robot now has all-wheel drive and also has independent front and rear steering allowing for traditional and parallel steering.

4.1.2 Weeding Mechanism

The weeding mechanism that was desired by the sponsor was an auger style system which would use the auger flighting to apply the needed shear force to the soils surface. The final design for the weeding mechanism uses two four inch augers which are driven independently by two DC gear motors. The weeding mechanism can be seen in **Figure 7** below.



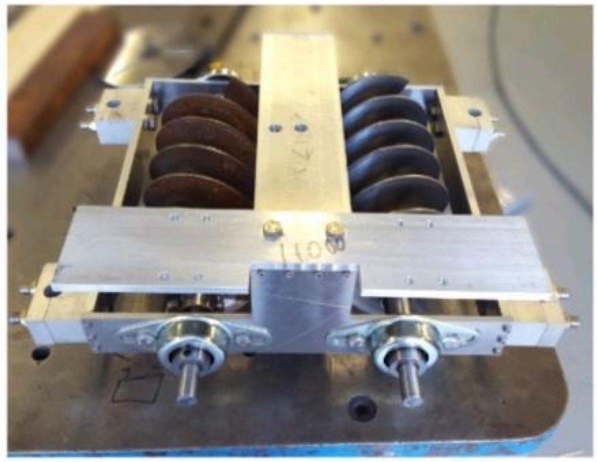
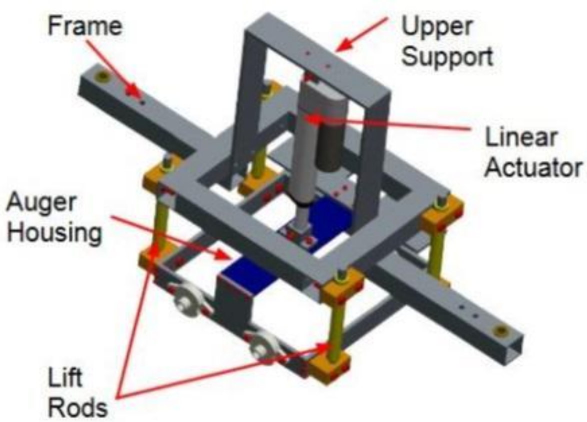


Figure 7: Auger style shearing mechanism.

In the left image of **Figure 7**, the CAD graphic is displayed with the appropriate component labels, while the right image is the actual auger housing after fabrication. Independent motors ensured the required torque needed to sufficiently shear off the one inch of soil was generated. Also, the auger housing was designed to incorporating different size augers from four to five inches.

4.1.3 Lift System

During operation of the Robo-Weeder, there will be a need to adjust the vertical position of the weeding mechanism. To provide this functionality, the design team implemented a lift system that provides six inches of travel. The lift system also delivers the needed stability to the auger housing through the lift rods.



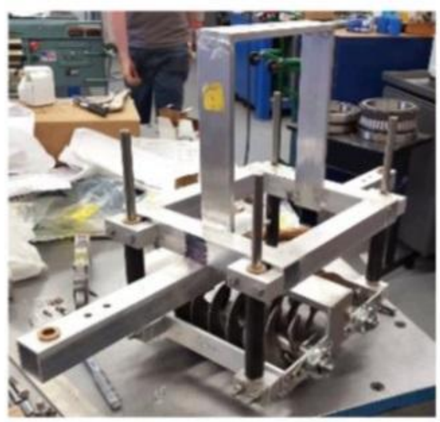


Figure 8: Lift system CAD and fabricated parts.

Above in **Figure 8**, the lift system is shown in detail. The left image shows the CAD graphic with the component labels. The image on the right shows the final lift system conjoining the frame and the auger housing.

4.1.4 Frame

The frame of the Robo-Weeder is the backbone for the project. The primary functionality is to provide strength and stability as well as provide a central attachment point for all of the Robo-Weeder's many subsystems.

4.1.5 Final Mechanical Design

During the evolution of the Robo-Weeder, the design team has seen many significant changes. The most notable change is to the overall size. The early design for the Robo-Weeder embodied an unstable length to width ratio of approximately 5:1. Other noteable changes made to the final design of the Robo-Weeder include the method of steering and also the functionality of the lift system. The early design of the lift system were manual in nature and required the operator to physically lower and raise the auger housing by rotating a handle attached to the lift system. Also, the steering system used a linear actuator to rotate the drive assembly in order to steer the ROV. The linear actuator was replaced due to the slow nature of extension and retraction.

After the redesign of the overall size of the Robo-Weeder to address stability issues, and the steering and lift system redesigns, the final Robo-Weeder embodies a wider, more stable 2.4:1 length to width ratio.

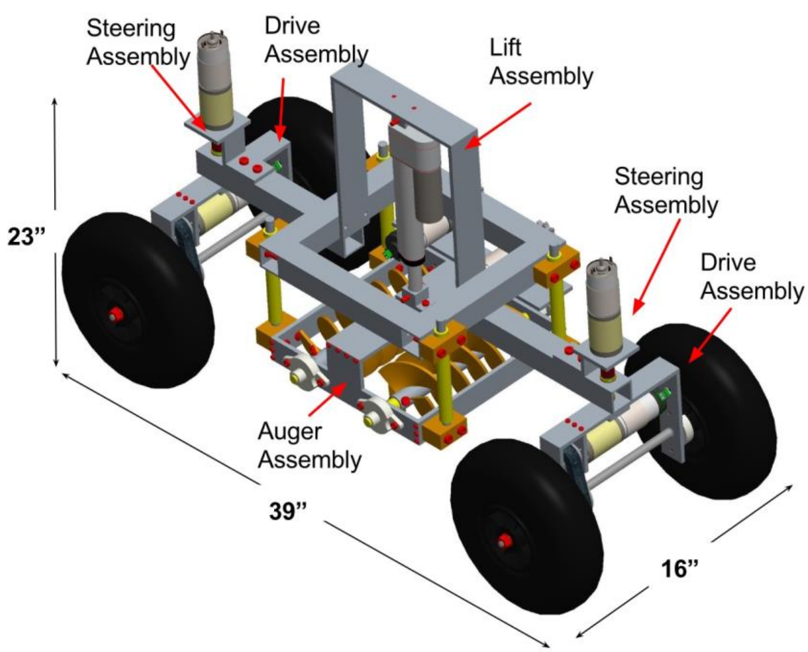


Figure 9: Final Robo-Weeder Design

To make the steering system more conventional, the linear actuator was removed and replaced by a direct coupling of a steering motor. This change can be seen in **Figure 9** above and is labeled "Steering Assembly." Also, the "Lift Assembly" labeled in the above figure shows the added linear actuator which serves to automate the lift system allowing for precision control of the auger depth and ground pressure.

4.2 Final Electrical Design

The final electrical design for the Robo-weeder was based on the choices of motors that will power the drive train, steering, and the augers. The heart of the electrical design is the Arduino MEGA microcontroller. The motor-controller selection was based on the voltage and amperage rating of the dc motors selected. This ensured that each motor-controller will not overheat due to the excess load requirement of the dc motor, in order to avoid any design failure. The 12V power system for this design will provide the necessary system requirements in terms of the operating voltages and current, in addition to being able to maintain an acceptable amount of charge for a given length of operational use in between charging. The transmitter selected will perform all necessary functions such as driving, steering, raising/lowering the weeding mechanism, as well as controlling the augers (turning them ON and OFF).

4.2.1 Transmitter/Receiver

Figure 10: FlySky FS-T6 6CH Transmitter w/ Receiver

Table 1: FlySky FS-T6 6CH Transmitter Specifications

Transmitter	FlySky FS-T6
Frequency	2.4GHz 6 channel transmitter
Power supply	8 AA batteries
Current drain	250mA, 87mA (Energy Saving)
Range	400 m
Frequency	2.4 GHZ

Power supply	12V

The radio transmitter that was selected is a Radio controlled transmitter that is typical of those in the aerial quadcopter transmitter application. It was determined that for the Robo-Weeder application, having six channels for a transmitter will serve the purpose of being able to provide all of the functions that is required for the Robo-Weeder to being operated. The FlySky FS-T6 transmitter has a range of 400 m that is suitable in the outdoor farming environment. The FlySky transmitter is simple to program and can be done easily when paired with its receiver. The radio receiver is more than capable of handling all the channels that will operate the Robo-Weeder; having six channels paired with the transmitter allows for a better transmission and receiving capability.

4.2.2 Microcontroller

Figure 11: Arduino Mega 2560

Table 2: Microcontroller Specifications

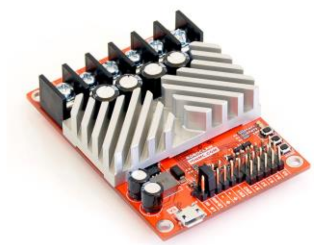
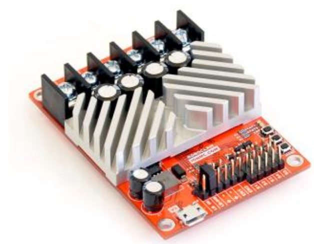
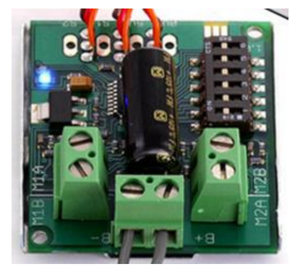
Models	Arduino Mega 2560 R3
Price	$45.95
Input Voltage	7-12V
Digital I/O Pins	54 (14 PWM)
Analog Input Pins	16
DC Current per I/O Pin	40mA
Flash Memory	128KB
SRAM	8KB

Clock Speed	16MHZ
Processor	ATmega2560

The Arduino Mega in **Table 2** will function as the "brain" of the electrical design. The Arduino Mega was selected due to the many pins it has on board. For our design there will be, in total, 6 motors and 1 linear actuator. The extra pins on board will be necessary if additional features are added if this project is continued in the following years.

4.2.3 Motor Controllers

Table 3: Motor controller Specifications

Motor Controllers	 RoboClaw 2x45 (x2)	 RoboClaw 2x30	 Sabertooth 2x5
Channels	2	2	2
Maximum Current/Ch.	45A	30A	5A
Operating Voltage	6V - 34V	6V - 34V	6V - 18V

In order to control the Robo-Weeder's many functions, 4 motor controllers are utilized. The Robo-Weeder consists of two 2x45A Roboclaw motor controllers, one 2x30A motor controller, and one Sabertooth 2x5 motor controller. The 2x45A Roboclaw motor controllers will be used to control the two shearing motors as well as the two steering motors. The shearing motors have the highest stall current out of all our motors maxing out at 44 amps, making the 2x45A an ideal choice. The steering motor has a lower stall current, however it was decided the 2x45 would be ok due to the Roboclaw's ability to limit the current output it produces as well as the opportunity for upgrading the motors in the future. The 2x30A Roboclaw will control the two drive motors of the Robo-Weeder which each have a stall current of 22 amps. The smaller motor controller was selected due to the Robo-Weeder navigating through soft soil which would not cause much current draw from the motors. Lastly, the Sabertooth 2x5 motor controller was selected to control the linear actuator onboard the Robo-Weeder to lift/lower the auger housing. This specific motor controller was recommended by the vendor from where the linear actuator was purchased and works well in conjunction with the Arduino MEGA.

4.2.4 Electrical System Design

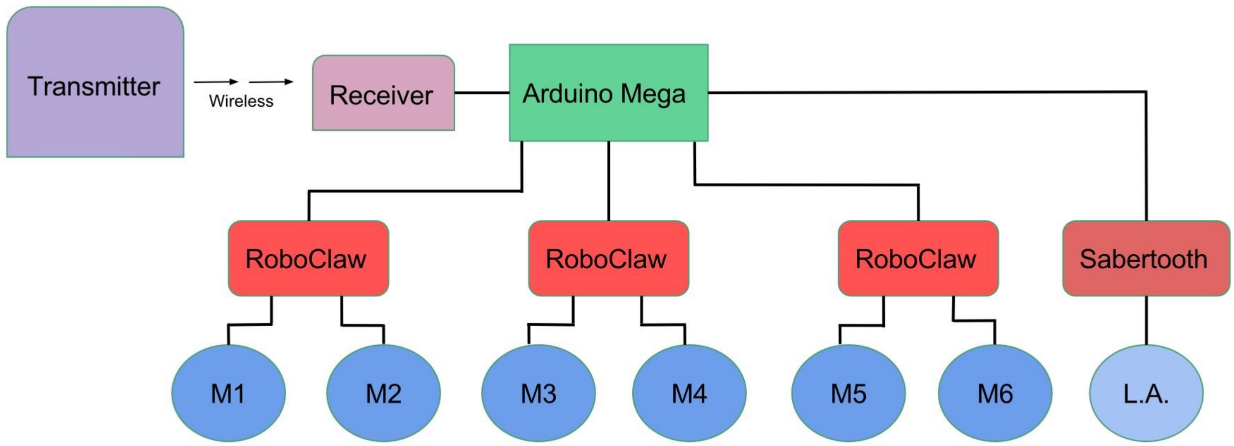


Figure 12: Communication between user and Robo-Weeder

Figure 12 above portrays the communication between the user and the Robo-Weeder. The user will have the FlySky transmitter in hand which will transmit a wireless signal to its compatible receiver powered by the Arduino MEGA. This signal will be interpreted by the Arduino microcontroller and will perform the desired function. The first Roboclaw motor controller will control the front and rear drive motors, M1 and M2. They will both be regulated by the y-axis of the right joystick on the transmitter in hand allowing for the Robo-Weeder to move forward or reverse.

The second Roboclaw motor controller will control the front and rear steering, M3 and M4. The x-axis of both joysticks regulate this specific function with the right joystick controlling the front steering and the left joystick controlling the rear steering. With the ability to steer both the front and rear assembly, this gives the Robo-Weeder enhanced control with the capability of parallel steering.

The third Roboclaw motor controller controls both auger motors, M5 and M6. On the top right of the transmitter is a knob which will serve as the operation point for the auger functions. The augers will be at standby when the knob is set to zero and will accelerate from zero to max speed in correlation with how far and also the direction that the knob is turned. During operation, the two motors will always be rotating at the same speed, but the operator can adjust the speed and direction of rotation by using the aforementioned control knob.

Lastly, the Sabertooth motor controller will control the linear actuator giving the operator the ability to raise and lower the weeding mechanism from a distance. Similar to the control of the augers, a control knob on the top left of the transmitter will raise and lower the mechanism. At the zero position on the left knob, the linear actuator will be locked in the current position that it is in. When the knob is turned clockwise, the housing is lowered. When the knob is turned counterclockwise, the housing is raised. This feature allows for the Robo-Weeder to perform its desired function on multiple types of terrain and also delivers enhanced control to the operator.

4.2.5 Power System

Figure 13: Battery for Robo-Weeder

Table 4: Keyko KT-12120 Battery Specifications

Brand	Keyko
Model	KT-12120
Voltage	12V
Amp-hour Rating	12Ah
Weight	7 lbs.
On-Board	3
Cost	$19.60 each

The battery that Team 11 has selected to power the Robo-Weeder design is the Keyko KT-12120 battery. This is a 12V battery, which is compatible with all of the motors and controllers to be used. It was decided to go with three of these batteries for multiple reasons. First, it was more weight-friendly to the Robo-Weeder design than to go with one large lead-acid battery. Having three small batteries gives flexibility to distribute the weight around the Robo-Weeder rather than having it at one central location. Secondly, putting these three batteries in a parallel configuration will give the Robo-Weeder an approximate runtime of 30 minutes using 36Ah. The battery system can always be upgraded, however, the more ideal batteries would come at a higher expense. Due to this, the battery was the final selection of the design. The 3 main criteria evaluated when selecting the battery were: cost, weight, and amp-hour rating. The cost of the battery system had to be within Team 11's budget, and the weight had to be within a reasonable amount to prevent any unbalance on the Robo-Weeder. Then finally, the amp-hour rating needed to obtain at least a minimum runtime of 30 minutes.

In order to control the power transmission on the Robo-Weeder, a main safety switch will be located on the Robo-Weeder itself. The switch will be in series from the positive terminal of the batteries to the distribution block with a fuse in between. This switch will also act as an emergency shut off, cutting off all power to the electrical components when turned off. The fuse will also be a safety feature in case too much current is drawn by the electrical components, which will result in it being blown and breaking the connection.

4.3 Finite Element Analysis

With the completion of the design process, the design team needed to ensure that key load bearing components could withstand the the stresses that the Robo-Weeder could experience during real world application. It was determined that the component that faced the largest loads were the lift system. These loads are concentrated on the lift rods and their perspective tubes in which they mate.

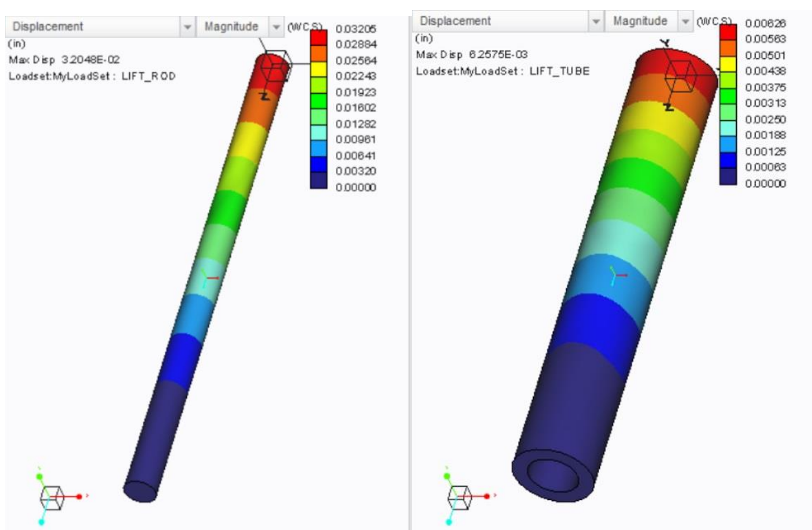


Figure 14: Results of the FEA analysis of the lift rods and mating tubes.

It was determined after the analysis was completed, **Figure 14**, that the neither component could be made from aluminum, and that steels were needed due to the material strength.

4.4 Failure Modes and Effects Analysis

Design teams often use failure modes and effects analysis or FMEA tools to analyze a given design to determine modes of failure. Team 11 conducted an FMEA of the Robo-Weeder and the full analysis can be seen in Appendix B. From the failure modes and effects analysis, it was determined that that DC motors and bearings were the most susceptible to failure during operation.

4.5 Dimensions and Tolerances

The overall dimensions of the Robo-Weeder are 39” x 16” x 23” and the overall size can and location of components can be seen in **Figure 9.** The primary system of the Robo-Weeder is the auger style shearing mechanism. The auger flighting is made of 7 gauge (0.1875”) thick carbon steel and each auger is approximately 10” in length.

The frame of the robot is made of 1.5” square aluminum tubing with 0.125” wall thickness. Square tubing enabled for a robust frame design and the aluminum material kept the design light in nature. Most other subsystems that were fabricated in the machine shop were made of 1.5” wide aluminum flat bar of 0.25” thickness. These subsystems include both drive assemblies as well as the auger housing. The team attempted to use aluminum material in as many places as possible in order to keep the weight of the Robo-Weeder as light as possible.

All load bearing shafts on the Robo-Weeder are created out of 0.5” diameter stainless steel rod. These load bearing shafts include both axles, all 4 of the lift rods and the steering pins. Using stainless steel in these locations drastically lowers the chance of material failure and also helps with material reactions (ex: oxidation) at bearing and bushing locations

4.6 Design Calculations

In order to determine the amount of shear force that an auger will place on the soil, it is first important to understand how to model the auger to get accurate force data. After consultations with Dr. Hollis, it was determined that the auger can be modeled as a large worm gear and would deliver an axial force (F_{axial}) and a tangential force (F_{tan}) when rotated. To determine the axial force, equation 1 below can be used.

$$F_{axial} = F_{tan} * \frac{cos(\alpha)cos(\gamma) - \mu * sin(\gamma)}{cos(\alpha) * sin(\gamma) + \mu * cos(\gamma)} \qquad \text{Eq: 1}$$

The tangential force exerted by a worm gear can be determined by equation 2 below.

$$F_{tan} = \frac{2 * M}{d_1} \qquad \text{Eq: 2}$$

In the following equations, gamma refers to the lead angle of the auger, alpha is the pressure angle, mue is the coefficient of friction, M is the auger torque and $d_1$is the reference diameter of the auger.

To apply the above equations, the axial force is determined from the soil analysis data, **Figure 1,** which was gathered in conjunction with the Civil Engineering Department. The axial force corresponds to the shear force that is delivered to the soil’s surface. With the needed shear force determined, the design team then solves the system of equations for the torque value in equation 2. It should be noted that the design team used a factor of safety of 2 when designing the auger system.Therefore the supplied torque should be well over the actual needed torque ensuring proper functionality of the auger soil shearing system.

4.7 Design for Manufacturability

4.7.1 Complete Mechanical Assembly

The Robo-Weeder prototype can be broken down into four main subsystems: Frame, Steering & Drive, Auger, and Lift. Each subsystem needs to be assembled individually, then mounted to the central frame.

4.7.2 Tools

- Common and Phillips Screwdriver
- Socket Set
- Allen Key Set
- Adjustable Wrench
- Channel Lock or Vice Grip Pliers

4.7.3 Frame Assembly

The frame of the Robo-Weeder is made of Aluminum square tubing that was welded prior to assembly. The vertical portion of the lift system also needed to be welded to the central portion of the frame. All other subassemblies will be attached to this central component of the chassis. The frame only needs four ½ inch bushings into the front and rear drive steering assembly mounting locations.

4.7.4 Drive & Steering Assembly

There will be two separate drive and steering assembly for the front and rear ends of the Robo-Weeder. For this assembly we needed, the socket set, Allen key set, and a pair of pliers. First we attached the steering pin into the main wheel housing by passing the pin up through the inside and use four bolts. Next, attach one flange bearing to the outer edge of each side of the main wheel housing using two bolts each. With the bearings now attached, we inserted the axle shaft through each bearing, making sure to mount the 18 tooth sprocket inside the wheel housing. We then secured the axles' position by pressing two e-clips into the two small grooves located on the inside of the wheel housing and secured the set screws onto the axle located on the flange bearings. With the axle and sprocket installed, next we inserted the ¼ inch shaft bushing in the upper hole on the wheel housing. We proceeded to mount the drive motor to the drive motor plate using 4 M4 screws, and installed the shaft motor shaft extension with the 10 tooth sprocket. Next we inserted the drive motor extension into the ¼ inch bushing and use three 5-32 screws to mount the motor to the wheel housing. Ensuring that the sprockets are aligned, we tighten the set screws on the sprockets to lock them into their prospective shaft. We finally, mounted each 10 inch diameter pneumatic wheel on either side of the wheel housing and secured the wheels with a locking nut. Next we prepared the steering motor for installation by using 4 M4 screws to attach the motor to the steering motor mount and installed the coupler onto the steering motor shaft. We repeated the process for the second drive and steering assembly.

4.7.5 Auger Housing Assembly

For the Auger Housing assembly we first prepared the socket set, Allen keys and pliers. It should be noted that the auger housing was designed to allow removal of the augers. We located the front, back, left, right, motor mount and top plates that make the main housing structure. We attached four two flange bearings onto both the left and right housing plates. We attached the right plate to the front, top and rear plates using ten 5-32 screws. At this time, we inserted the extension pins into their perspective auger and inserted the extended augers into the bearings installed on the right plate. Note: we needed to ensure that the auger with right hand flighting is installed towards the front of the auger housing. With the augers inserted into the right housing plate, we inserted the auger extensions into the left housing plate and attached it using ten 5-32 screws. We then proceeded to align the augers and tighten the flange bearing set screws. With the augers installed and the main housing assembled, we attached the auger motors to the motor mount plate, and attached 10 tooth sprockets to both the motor shafts and the auger motor extension pins and finally mounted the auger motor mount onto the auger housing top plate.

4.7.6 Lift Assembly

The lift system is a relatively simple structure that was mounted directly to the auger housing and the frame. The socket set is the only tool needed for assembly. We first located the four ½ inch diameter stainless rod and the four 1 inch steel tubes. These are the stabilizing posts of the lift system and serve to keep proper alignment. We loosely installed the inner and outer aluminum brackets to the auger housing front and rear plates using eight 5/16 inch bolts and locking hardware. We then inserted the ½ inch stainless steel rods between the aluminum brackets ensuring the rod is flush with the bottom of the bracketry and tighten the bolts. With the rods attached to the auger housing, we aligned the 1 inch steel tubes flush with the top of their perspective mounts, then install ½ inch bushings inside the steel tubing. Using eight 5/16 bolts and locking hardware, we installed the tubes onto the central frame such that the tubes protrude downward. Finally, we installed the bracketry on the vertical portion of the lift and the auger top plate for installation of the linear actuator.

4.7.7 Final Assembly

With each of the sub-assemblies completed, we raised the frame over the auger housing and slid the ½ inch rods through the bushings in the 1 inch steel tubing. We then took the auger housing to the highest position and installed the linear actuator using the provided pins. Next we, inserted the steering pin of each drive and steering assembly through the bushings located on the front and rear ends of the frame. We then secured each assembly into their prospective locations by installing two e-clips into the small grooves on the steering pin just above the bushing. Locating the steering motor, we inserted the mounted coupler over the steering pin and mounted the motors of the frame using four 5/16 bolts and locking hardware. Finally, we locked the coupler to each perspective shaft by securing the set screws located on each coupler we also, installed the drive chain for each of the drive and auger motors.

4.7.8 Completed Prototype and Components

The design team in charge of the Robo-Weeder project completed a single prototype but had plans to construct a wooden model for testing functionality. However, due to time constraints, the wooden model never came into fruition. The prototype took longer to design than originally expected due to the complex nature of the robots many components and subassemblies.

Components

QTY	Component Name
4	10” Pneumatic Wheels
8	½” Flange Bearings
6	10 Tooth Sprocket
2	18 Tooth Sprocket
4	Roller Chain (Variable Lengths)
8	E-Clip ½“ Shaft
2	AndyMark 71:1 DC Gear Motor with Encoder
2	AndyMark 188:1 DC Gear Motor
2	BaneBots 104:1 DC Gear Motor
2	Optical Encoder
2	Absolute Encoder
1	Linear Actuator - Firgelli Automations
2	Linear Actuator Mounting Bracket
12	5/16 - 18 Bolt (2.5” Length)
8	5/16 - 18 Bolt (1.75” Length)
30	5-32 Cap Head Screw
8	10-32 Cap Head Screws
10	1/4 - 20 Set Screws
24	1/4 - 20 Bolt (0.5” Length)
10	1/4 - 20 Locking Nut
15	5/16 - 18 Locking Nut
12	½” Brass Bushing

- 2 ¼” Brass Bushing

Total: 176

Note: Raw Materials were purchased for the fabrication of unique components through the College of Engineering Machine Shop and the materials are listed below.

Materials

Material Description

- 1.5” Square Aluminum Tubing (0.125” Wall Thickness)
- 1.5” Aluminum Flat Bar (0.25” Thick)
- 2.5” Aluminum Flat Bar (0.375” Thick)
- 0.5“ Diameter Stainless Steel Rod (7’ Length)
- 1” Diameter Carbon Steel Tubing (3’ Length)
- 0.75” Diameter Stainless Rod (1’ Length)

The current Robo-Weeder prototype houses approximately 72 fabricated components from the machine shop bringing the final component count to 233. It should be noted that this count takes into consideration welded assemblies as 1 component. A large portion of the included components is comprised of bolts, screws and nuts and further consideration could be taken to weld more components therefore reducing the total number of components in the final prototype. The reasoning for not welding more components was for the ease of replacement or augmentation of the current prototype if any design changes were necessary.

4.7.9 Preparation for Electrical Assembly

The electronic system housed on the Robo-Weeder is diverse in nature and allows for the control of up to 6 DC motors. To control these 6 DC motors, three Roboclaw motor controllers are implemented along with an Arduino Mega. Each Roboclaw motor controller is able to control up to 2 motors each: 2 for the drive function, 2 for the steering function, and 2 for the shearing function. All of the electrical components will be housed inside of a sealed electrical box located centrally on the Robo-Weeder chassis.

In order to correctly attach all of the electrical components to their corresponding electro-mechanical components, wiring and connectors need to be prepared. To deliver power to all of the DC gear motors, 8 AWG power and ground cable is used to ensure proper current carrying capability. Furthermore, after sizing of the cabling, crimp style connectors are affixed to either end to ensure proper power transmission.

The coding process for the Robo-Weeder started out with different coding for each separate components; a separate code for the drive, the auger, and the steering system. The final code will be the combination of the entire components into a single master code, later to be downloaded into the microcontroller for field operational use.

4.7.10 Electrical Components

QTY	Component Name
1	Arduino Mega
3	Polulu Roboclaw Motor Controllers
1	Sabertooth Motor Controller
1	Flysky 6 Channel Wireless transmitter/Receiver
1	Polycarbonate Electrical Housing
1	25’ 8 AWG Power Cable - RED
1	25’ 8 AWG Power Cable - BLACK
2	8 AWG 40A Power Distribution Block
40	8 AWG Crimp Connectors

4.7.11 Electrical Installation

To begin the installation and assembly of all of the electrical components onto the Robo-Weeder, first attach the following components onto the floor of the electrical housing: 1 Arduino Mega, 3 Robo-Claws, 1 SaberTooth, and 1 Distribution Block. With the components in place, attach the appropriate power cabling between each motor controller and the power distribution block. Next, attach the appropriate power cabling between each motor controller and their perspective motors. Before connecting the distribution block to the battery, connect the power switch and fuse to the positive terminal of the battery, respectively. Finally, connect the negative terminal of the battery and the end of the fuse cable, from the positive terminal, to their respective distribution block.

The electrical system is a moderately simple system that uses a single microcontroller as the central processing point. The transmitter is used to control the Robo-Weeder’s many functions. The receiver is directly connected to the Arduino Mega which will interpret the signal from the transmitter and send the corresponding signal to appropriate motor controller. To simplify the system, 2 channel motor controllers were used so that the total number of components could be reduced.

4.8 Design for Reliability

The primary electrical concern that is still remaining for the Robo-Weeder project is the available runtime provided by the power system. Three areas of interest were evaluated while determining the appropriate battery system: size and weight, capacity and cost. The primary factor affecting the battery system was the cost. This was due to the optimal battery with the appropriate size and capacity was determined to be too expensive.

Another electrical concern for the Robo-Weeder would be the steering function. The steering function is controlled using an absolute encoder attached to a DC motor and it is designed to turn approximately 20°. However, during testing of the steering function, it was hard to control due to the absence of a load. During testing of the steering motor, it would tend to over shoot its reference location when returning to center. It was suggested that the team test the steering function with a load on the DC motor in hopes of preventing the overshoot.

4.8.1 Reliability of Design

During a single test of the individual components, there was no visible stress on any single component. All mechanical components performed as expected. After inspection, there were no loose bolts or bent/failed components. After 100 uses, due to the quality of materials used, it is highly unlikely that any of the mechanical components would have failed. The component that would be under the highest stress is the auger. The auger assembly is made from high strength carbon steel and will be very resilient to stresses seen under normal operating conditions. However, due to the nature of the carbon steel some oxidation could be present if the protective coating has worn off. It is recommended that the parts of the auger that are not in contact with the soil be closely monitored for oxidation. When considering 1000 uses of the prototype, there is expected to be little to no physical damage to the structural components of the Robo-Weeder. Due to the operating environment of a farm and the moisture and dirt that is present, it is possible that after 1000 uses the flange bearings could begin to fail and that minor oxidation could be observed on the augers. However, due to the sealed nature of the bearings that were selected, their failure along with unwanted oxidation is still an unlikely occurrence if proper maintenance procedures are followed and the robot is used as intended. Finally, when considering a usage rate of 10,000 times, the design team still anticipates very little wear on the mechanical structure of the Robo-Weeder. However, it is entirely possible for motor failure to occur due to the rate of use. The Andymark motors used in the drive function of the prototype can tend to reach elevated temperature levels which could lead to a loss in performance or failure. Also, the flange bearings and oxidation of the augers could potentially be a threat at this level of usage. Finally, at a usage rate of 10000, wear of the absolute encoder could be a concern due to the physical rotation of the encoder shaft.

From the above analysis, the primary concern would have to be the flange bearings. These bearings are used with the drive shafts and the auger shafts and are in close proximity to the soil. Under dry conditions, they will most certainly be exposed to large amount of dirt that could potentially cause failure if the dirt managed to pass the bearing seals. For this reason, regular cleaning should be practiced to ensure dirt is flushed from problem areas.

Under prolonged usage, motor failure would be a significant concern. The motors that were used in the Robo-Weeder design may not be ideal as this prototype is mainly a “Proof of Concept” design. An upgrade of the drive motors would be the appropriate action to ensure reliability of the product under prolonged usage.

4.9 Operations Manual

The Robo-Weeder is designed to be controlled through wireless communication using a transmitter and receiver. The transmitter used for this is the FlySky FS-T6. In order to drive the Robo-Weeder, the y-axis of the right joystick of the transmitter will control both drive motors transmitting a signal telling the machine to drive forward or reverse. The x-axis of the right joystick will operate the front steering of the Robo-Weeder while the x-axis of the left joystick will operate the rear steering of the Robo-Weeder, allowing for the capability of parallel steering. To power on the augers, the knob on the top right of the transmitter can be twisted in order to speed up the augers from neutral to maximum speed. Once the desired auger speed is set, the augers will continue to spin at a constant RPM until the knob is twisted again. The knob on the top left of the transmitter will operate the linear actuator, lifting or lowering the auger housing bringing it closer or farther from the ground. **Figure 15** below gives a schematic view of the transmitter and its functions.

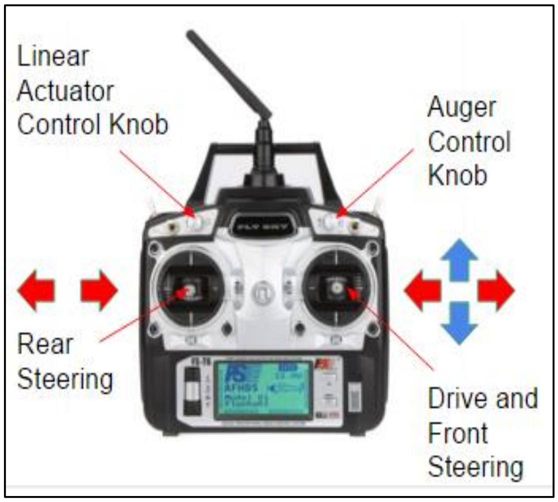


Figure 15: Transmitter Functions

With the transmitter in hand, the operator must first rotate the main power switch to the on position signified by a green window on the side of the power switch. The power switch is located on the side of the lift support tower on the robot. With the main power switch in the on position, the Robo-Weeder is now activated and ready for use. At this time the operator should move the power switch on the transmitter to the on position which will activate the transmitter allowing control of the robot. As seen in **Figure 15** above, the operator can now operate the Robo-Weeder.

With the robot in the correct position on the seedbed, rotate the top left knob to lower the linear actuator into position. The augers should be in loose contact with the soils surface to begin. With the augers in position, set the auger speed by rotating the top right knob until the desired speed is reached. Begin the weeding operation by using the joysticks to drive the robot down the seedbed next to the crop. Adjustment of the auger system can be made at any time during operation to achieve the correct amount of ground contact and angular velocity. When the weeding operation is complete, return both knobs to the off position to lift and stop the rotation of the augers.

4.10 Troubleshooting

Trouble shooting is a valuable part of the design process and allows for accurate repair and ultimate improvement of Robo-Weeder's components. In this section, the possible problems, methods of detection and their solutions will be outlined.

NOTE: Power should be disabled from the Robo-Weeder before engaging in any troubleshooting actions!

Problem: *Failure to move when drive motor is engaged.*

- Action: Check the locking hardware that secures the wheel to the axle shaft and ensure it is tight.
    - Repair: Tighten locking hardware that holds tire onto the axle.
- Action: Check the roller chain system to ensure chain is intact and properly aligned on sprockets.
    - Repair: Remove master link on roller chain, realign chain on the sprockets and reinstall link.
- Action: Check the roller chain system to ensure sprockets are secured to shaft.
    - Repair: Using an Allen key, tighten the setscrews onto the axle shaft.
- Action: Check the battery voltage.
    - Repair: Charge the batteries and ensure 12V.

Problem: *Failure to turn when steering motor is engaged.*

- Action: Check to ensure steering pin coupler is secured to both motor and steering shaft.
    - Repair: Tighten the setscrews onto both the steering motor shaft and the steering pin.
- Action: Check the battery voltage.
    - Repair: Charge the batteries and ensure 12V.

Problem: *Augers fail to turn on when auger motors are activated.*

- Action: Check the roller chain system to ensure chain is intact and properly aligned on sprockets.
    - Repair: Remove master link on roller chain, realign chain on the sprockets and reinstall link.
- Action: Check the roller chain system to ensure sprockets are secured to shaft.
    - Repair: Using an Allen key, tighten the setscrews onto the axle shaft.
- Action: Check to ensure augers are free of debris.
    - Repair: Remove debris from the auger.
- Action: Check the battery voltage.
    - Repair: Charge the batteries and ensure 12V.

4.11 Maintenance

Regular maintenance and inspections should be conducted on the Robo-Weeder in order to maintain optimal performance during operation. This section will outline the critical components that need to be maintained.

Auger Housing

The auger housing has the largest amount of cap screws which can become loose after prolonged use. It is recommended that each week, a full inspection of all screws in the auger housing be checked to ensure they are properly tightened. The housing is also the location of the lift rods and their subsequent mounts. It is recommended that the bolts associated with the attachment of the lift rods be checked weekly. If it is determined that any screw or bolt is loose, tighten immediately before using the Robo-Weeder.

Another key component of the auger housing is the flange bearings which are secured to the auger shafts. It is recommended that after every use that these bearing are inspected to ensure sand and debris has not become lodged around the bearing. If debris of any kind is present, remove to insure premature failing of the bearings does not occur. The roller chain and sprockets associated with each auger should also be checked to insure proper tightness and alignment. Oil should also be added to the roller chains monthly to ensure proper lubrication.

Steering and Drive

The steering and drive system on the Robo-Weeder is fairly robust and only minor maintenance will need to be performed to maintain proper. As mentioned with the Auger housing, inspect both the flange bearings connected to the axle shaft as well as the roller chain and sprocket system. Ensure proper lubrication and tension of the roller chains and check that the bearings are free of dirt and debris.

Electrical System

Regular maintenance of the electrical system comes in the form of ensuring that all the wires are in good condition, no frayed wires or damages in the distribution box. The plastic housing for the electrical system has to be cleaned regularly and ensure that proper seal is intact, ensuring protection from the element. The battery itself has to be kept in a constant state of charge, never leave the battery in a state of discharge for an extended period of time. The terminal connections to the motors, batteries, motor controllers have to be inspected regularly after each use of the Robo-Weeder, ensuring that no wires are loose from its connection. The transmitter has to be kept in a secure place as well as kept clean at all times.

Routine maintenance of the electrical system needs to be completed after each use of the Robo-Weeder. Always keep the electrical system clean at all times. The key feature to be kept clean constantly are the motor encoders that are built-in within the motor component itself, always make sure that the encoder is kept clean at all times, no visible damage to the wires or the encoder itself.

5. Design of Experiments

5.1 Electrical Component/Communication

During the design process of the Robo-Weeder, it was necessary to conduct experiments to test functionality of the many electrical components of the robot. The most abundant testing conducted was that of the electrical system and communication. In the below figure, **Figure 16**, the electrical engineers tested communication between the hand held transmitter and the microcontroller in order to rotate a simple DC motor.

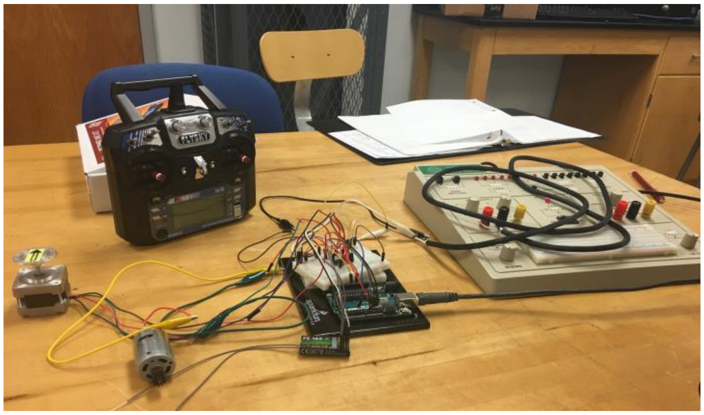


Figure 16: Early communication testing between the transmitter, Receiver and Microcontroller.

As the design complexity increased, it was important to keep testing the electrical design to ensure the desired functionality was obtained from the Robo-Weeders 6 DC gear motors and 1 linear actuator.

5.2 Mechanical Functionality

Once the arduino code that controls the different aspects of the Robo-Weeder was completed, mechanical testing of individual components commenced to ensure proper function. These individual tests included testing the steering, weeding mechanism and lift system. All of the initial tests were conducted in a controlled environment in a laboratory setting.

5.2.1 Steering Testing

The steering function of the Robo-Weeder is controlled by an Andymark 188:1 gear motor that is directly coupled to the steering pin of the drive assembly. To ensure that proper control was achieved, team 11 utilized an absolute encoder coupled to the gear motor which gave the motor a reference point. This reference point was used as the robot's forward position, and signified that the robot was not turning in either direction. Utilizing the nature of the transmitter, the joystick that controlled the front steering and drive, returns to center when not in use. The team tested this functionality to ensure that the Robo-Weeder’s front drive assembly would rotate 30 degrees and upon release of the joystick, return to its reference position.

5.2.2 Weeding Mechanism Test

The primary function of the Robo-Weeder is to remove weeds by applying a shear force to the soil. The each auger style weeding mechanism is driven by a Banebot’s 104:1 gear motor attached by way of roller chain and sprocket. The design team built the Arduino code in such a way that allowed the augers to operate at variable speeds depending on the operator's preference and need.

The transmitter is equipped with two knobs which fit the augers function perfectly. The knob was programmed in such a way that the center position is the off position and if rotated left and right, different directions of rotation could be achieved. Upon testing this function, the design team achieved proper transmission of power as well as enhanced control through the rotation of the knob.

5.2.3 Lift System Test

Just as important as the weeding mechanism, the lift system also plays a very important role in the functionality of the Robo-Weeder. The lift system is centered around a linear actuator which is operated by the user by way of the handheld transmitter. The lift system also serves as the direct coupling of the weeding mechanism to the frame of the robot.

The linear actuator on board of the Robo-Weeder is designed in such a way that the retracted most position ensures no contact between the weeding mechanism and the robot frame. This is accomplished by utilizing the factory installed limit switches that stop the actuator from over extending and retracting too far.

The linear actuators primary function is to give the user increased control over the weeding mechanism by allowing more ground pressure, less ground pressure and to raise it completely to remove contact all together. To give the desired control, a second knob on the transmitter is utilized to control this function. Again, the center most position signifies for the linear actuator to hold position, while rotating the knob left or right extends and retracts the actuator.

With the linear actuator installed on the Robo-Weeder, the major point of testing was to ensure proper function of the guide rods and their perspective tubes during operation. Upon completion of testing, the team achieved seamless linear travel of the auger housing during operation of the linear actuator.

5.2.4 Integrated Functionality Testing

At the conclusion of the testing of the individual subsystems, the design team set its next testing phase as a full integrated test of the Robo-Weeder’s complete functionality. This integrated test was completed in a laboratory setting, **Figure 17** below, and its primary purpose was to test the integrated arduino code and ensure proper simultaneous functionality.

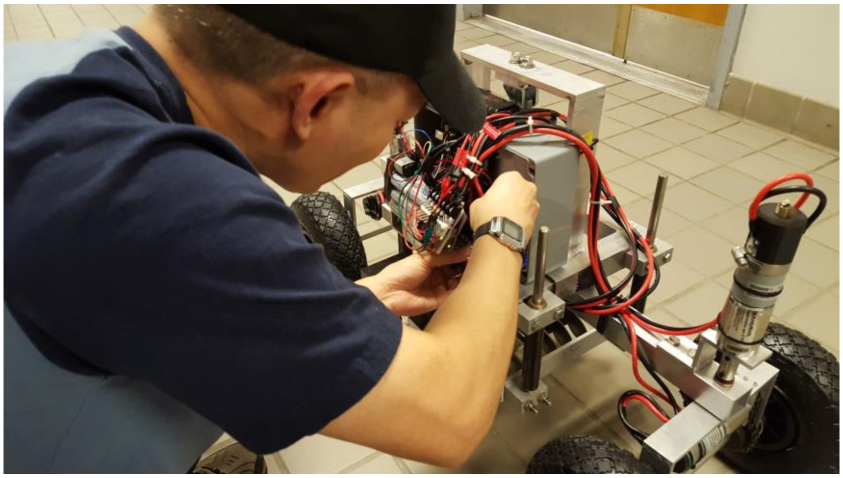


Figure 17: Preparing the Robo-Weeder for an integrated test.

At the conclusion of the integrated testing, the design team observed near perfect performance of the Robo-Weeder in the controlled environment. The only flaw that was observed was in the steering. The flaw seemed to indicate that the code controlling the steering function was not reading the encoder properly. After further research it was determined that the absolute encoder being used to read the steering motor position is incompatible due to the input speed to the encoder. To remedy this, the design team replaced the absolute encoder with a Hall Effect encoder. While not ideal for the situation, the new encoder will suffice for the given application.

5.2.5 Field Testing

The final testing phase to be completed with the Robo-Weeder is the field tests in real world conditions. The importance of this testing phase is to monitor all of the Robo-Weeder's many components under a load and ensure proper functionality. The field testing phase is currently scheduled to be completed April 12th, and was delayed due to a lack of proper power source and the surfacing of the error with the steering functions. The goal for the design team during field testing is to ensure proper shearing of the soil during forward motion of the Robo-Weeder.

6. Considerations for Environment, Safety and Ethics

The primary function of the Robo-Weeder project is to remove weeds from the seed beds of organic produce. The key constraint when considering the environmental impact for our project is that the Robo-Weeder was to not disturb more than 1 inch of the top soil. This primary constraint served as a key means to preserve the environmental habitats for key microorganisms which would aid in the growth of organic produce. Also, the minimal impact on the soil would prevent mass soil erosion that is sometimes present on traditional farms that employ mass tilling of the soil.

7. Project Management

7.1 Schedule

During the early stages of the design teams forming, it was important to properly coordinate the team's resources. The first step in coordinating the team's efforts was to set up weekly meetings so that the team could reflect progress made in their individual tasks as well as ask questions of others if difficulty was encountered. Also, to ensure that the team stayed on schedule, the team utilized a gantt chart to plan each semester. The Gantt chart was broken into fall and spring semesters for simplicity. The fall semester Gantt chart focused on the mechanical and electrical design aspects of the project while the spring Gantt chart focused primarily on procurement, fabrication and testing. Both Gantt charts can be viewed in Appendix D.

7.2 Resources

One of the benefits to all senior design projects completed at the FAMU/FSU College of Engineering is the vast amount of resources available to students. During the fall semester, the resources that were utilized were vastly different than those used in the spring. During the fall semester, faculty and staff were a huge resource that was used by the design team. Dr Hollis and Dr Moore provided a great amount of knowledge to the mechanical design process. While Dr Hooker and Dr Gupta provided the team with advice on the many electrical components that would eventually be incorporated on the Robo-Weeder. Also, the different engineering disciplines that are studied at the College of Engineering allowed the design team to expand their knowledge outside of their area of study. During the fall semester, the design team consulted with the Civil Engineering department to test and study soil shear effects that proved useful when designing the auger style weed shearing mechanism requested by the sponsor.

During the spring semester, the largest resource that was utilized by the design team was the College of Engineering machine shop. Team 11 consulted regularly with the machine shop staff regarding fabrication details that would need to be accounted for during the machining process. Coupled with the machine shop, the design team also utilized Mr. Keith Larson to ensure the CAD was properly formatted.

7.3 Procurement

The Robo-Weeder senior design team began the semester with a total budget of $3000.00. At the project's current state of completion, the design team has spent approximately $3,028. However, there was a discrepancy with the shipping cost of all of the parts, and the total that the team was aware of spending was $2,792. This budget discrepancy has been discussed with the FAMU-FSU College of Engineering and all overages has been authorized.

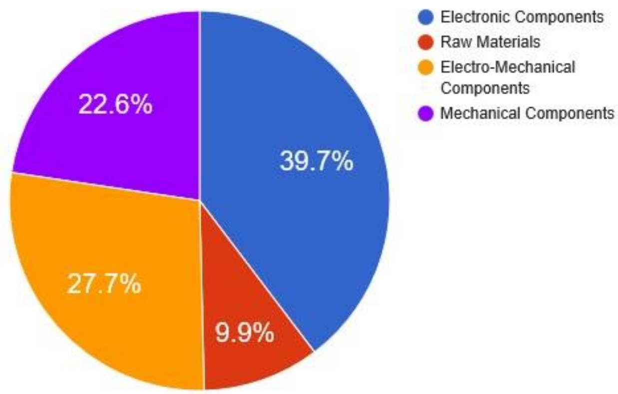


Figure 18: Spending report and breakdown of funding.

The breakdown of spending can be seen in the above figure, **Figure 18**. Currently team 11 has consumed the total budget allotted to them. A full description of the team's purchases can be seen in Appendix C: Procurement.

8. Conclusion

The Robo-Weeder senior design projects primary goal was to design and fabricate a remotely operated robotic system to aid in weed control on organic farms. The weeding mechanism that is housed on the Robo-Weeder is an auger design that will remove weeds by applying a shear force to the soils surface resulting in the uprooting of the weeds. The Robo-Weeder's many functions are controlled by six DC gear motors. The central unit controlling the electronics located on the Robo-Weeder is an Arduino Mega microcontroller which in turn sends signal to three Roboclaw motor controllers. The Robo-Weeder is designed to be operated by a user that is in close proximity to the robot allowing for quick corrections in the weeding mechanisms output as well as the direction of travel. With the ability to manipulate auger speed and ground contact, the Robo-Weeder will greatly reduce the physical labor required to control weed growth on organic farms.

Being that the Robo-Weeder project is a continuation of the previous year's senior design team, the 2015-2016 design team opted to start the project fresh due to several design concerns regarding the previous year's prototype. This required the design and fabrication of a completely new prototype. The design team only used the data and information from the previous year. If in turn the Robo-Weeder project is again continued next year, we feel that the current prototype and its current state would give the next team a valuable head start in the design process if the prototype is used. With the continuation of the Robo-Weeder project, the next design team could ultimately merge both years' designs into one singular robotic system making use of the autonomous nature of the 2014-2015 prototype and the weeding mechanism and chassis of the 2015-2016 prototype.

Although the prototype is complete, there are future design changes that could be completed to make the overall project function better. The first design alteration would be to extend the frame of the Robo-Weeder to increase the degrees of turning of each drive assembly back to the original 30 degrees. During the design of the lift system, the location of the supports for the lift rods reduced the turning angle of the drive assembly from the original 30 degrees to 20 degrees. The design of the lift system supports were completed after the fabrication of the robot's frame and the design team lacked sufficient time and resources to make the necessary modifications. Another improvement that could be made would be to increase the thickness of the metal of the auger housing. The auger housing on the current Robo-Weeder is 0.25 inches thick and requires small bolts to fasten the auger housing together. The small bolts creates an issue due to the moderately low amount of force required to shear the bolt heads during installation.

Over the course of the fall and spring semesters, the senior design team has experienced its fair share of setbacks that could have been avoided. During the fall semester, the design team needed to consult with their faculty advisors more often with problems encountered during component selection. Due to this lack of communication, team 11 found itself behind during the fall semester during the motor selection phase. Dr Gupta is a very good resource that should have been used more effectively early on in the design phase.

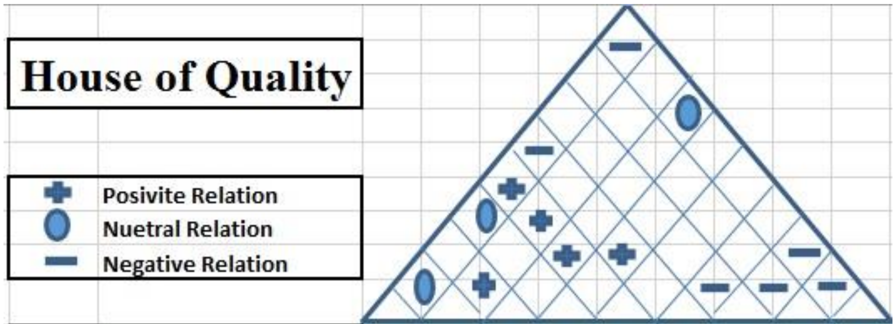
References

1 http://12.000.scripts.mit.edu/mission2014/problems/ineffectiveinadequate-agricultural-practices

2 http://www.ofrf.org/organic-faqs

3 https://www.ocf.berkeley.edu/~lhom/organictext.html

Appendix A: House of Quality



Importance/Weight	Customer Requirements	Mass	Material	Durability	Stability	Strength of Components	# of Tires/Tracks	# of Motors	Operation Mode (Wired, Radio, etc...)	Battery System
4	Safe to Operate	4	20	20	40	20	4	4	4	4
1	Cost Efficient	5	10	5	5	5	10	10	5	5
5	Effective	5	25	50	5	25	25	5	25	25
5	Reliable	5	25	50	50	50	5	5	5	25
2	Simple to Operate	2	2	2	2	2	2	2	10	2
2	terchangable Implemen	20	10	10	20	10	2	2	2	2
3	Weight	30	30	30	30	30	30	30	3	15
2	Marketability	2	2	20	20	10	2	2	10	2
4	Irregular Terrains	4	4	20	40	4	40	4	4	4
Totals		**77**	**128**	**207**	**212**	**156**	**120**	**64**	**68**	**84**
Rank		**7**	**4**	**2**	**1**	**3**	**5**	**9**	**8**	**6**

Figure 19: House of Qualities.

Appendix B: Failure Modes and Effects Analysis

Process or input	Failure Mode	Failure Effect	SEV	Cause	PROB	Control	DET	RPN	Preventive Measures	Responsibility
Augers	Stops from tangled debris	Stalling of Motor	8	Large amount of debis present	3	Auger design prevents tangling	4	96	Ensure field is free of large debris items to include rocks and roots.	Operator
		Bending of Flighting	5	Large amount of debis present	3	High strength materials prevent failure	4	60		
DC Motors	Burns Up	Loss of function	8	Over Loading	3	Over Design	7	168	Ensure proper alignment	Operator
		Loss of function	8	Bearing Wear	2	Accurate Alignment	7	112	Clean bearings	Operator
Roller Chain	Derailing from sprocket	Loss of Drive/Weeding	6	Chain too loose	4	Proper spacing of sprockets.	2	48	Install a chain tensioner	Operator
	Link Breakage	Loss of Drive/Weeding	6	Over Loading	2	Over Design of Chain	8	96	Operate as intended	Operator
Tires	Flat Tire	Insufficient ground clearance	5	Wear, Age, Puncture	7	No Control	1	35	Remove debris items	Operator
	Lock Nut Failure	Loss of Drive	3	Wear	3	Locking compound on threads	5	45	Remove debris items	Operator

Figure 20: Failure modes and effects analysis for Robo-Weeder.

Appendix C: Procurement

Ordered Items							
Vendor	Item Description	Part Number	Quantity	Price	Total Cost	Status	Dept
Robot Shop	Arduino Mega 2560 Microcontroller	RB-Rlk-03	1	34.99	34.99	Received	EE
Robot Shop	Radiolink Transmitter and Receiver	RB-Ard-33	1	36.81	36.81	Received	EE
Bloom MFG	Auger Flighting - Right Hand	528	1	61.00	61.00	Received	ME
Bloom MFG	Auger Flighting - Left Hand	528L	1	61.00	61.00	Received	ME
McMaster	Material - AL Flat Bar and Tubing, Steel Rod and Tube	N/A	-	120.00	120.00	Received	ME
Northern Tool	10" Pneumatic Tire/Wheel	2252	2	9.99	19.98	Received	ME
Amazon	12V 30A DC Universal Power Supply 360W	S-360-12	1	23.97	23.97	Received	EE
Amazon	Heavy Duty Power Cord - 6 Feet	N/A	1	9.99	9.99	Received	EE
Amazon	16 Pack 2800 mAh Rechargable Batteries w/ Charger	N/A	1	39.99	39.99	Received	EE
Robotshop	PG188 Gearmotor - No Encoder (Steering)	am-2193a	1	79.00	79.00	Received	EE
Robotshop	PG71 Gearmotor w/ encoder (Drive)	am-2971	1	89.00	89.00	Received	EE
Andy Mark	Absolute Encoder w/ Cable	am-2899	1	45.00	45.00	Received	EE
Pololu	Roboclaw Motor Controller 45A continuous	2397	2	169.95	339.90	Received	EE
McMaster	Materials - 304 Stainless Rod and Flatbar	N/A	-	35.63	35.63	Received	ME
Grainger	1/2 Flange Bearing	4X727	2	21.22	42.44	Received	ME
Robot Shop	Arduino Mega 2560 Microcontroller	RB-Rlk-03	1	34.99	34.99	Received	EE
Amazon	1/2 Brass Steering Bushing	EXEF081008	2	8.99	17.98	Received	ME
McMaster	1/2 Chain & 1/2 Sprockets	N/A	-	73.90	73.90	Received	ME
Amazon	FlySky FS-T6 2.4ghz Digital Transmitter	N/A	1	49.99	49.99	Received	EE
Andy Mark	PG188 Gearmotor - No Encoder (Steering)	am-2193a	1	79.00	79.00	Received	EE
Andy Mark	PG71 Gearmotor w/ encoder (Drive)	am-2971	1	89.00	89.00	Received	EE
Andy Mark	Absolute Encoder w/ Cable	am-2899	1	45.00	45.00	Received	EE
Andy Mark	Optical Encoder Kit	am-3132	2	42.00	84.00	Received	EE
Banebots	DC Motor - RS540 - 12V	M5-RS540-12	2	4.37	8.74	Received	EE
Banebots	P60 Gearbox: Standard, 104:1, for RS500 motor	P60S-554-5	2	67.95	135.90	Received	ME
polulu	Roboclaw Motor Controller 30A continuous	2396	1	124.00	124.00	Received	EE
Firgelli	Linear Actuator		1	119.99	119.99	Received	EE
Firgelli	Mounting Bracket		2	7.00	14.00	Received	ME
Dimension	Sabertooth 5A Motor Driver		1	59.99	59.99	Received	EE
McMaster	Nuts/Bolts/Materials/Sprockets/Chain/		-	272.89	272.89	Received	ME
Grainger	1/2 Flange Bearing	4X727	6	21.22	127.32	Received	ME
Digikey	Power Wire/Connectors		-	194.08	194.08	Received	EE
Vellaman	Electrical Housing	G378	1	24.60	24.60	Received	EE
Andy Mark	Encoders and Cables			60.00	60.00	Ordered	EE
McMaster	Grommets, Shrink Tubing, Cable Sheathing			63.55	63.55	Ordered	ME
FastBatt	Lead Acid Battery: 12V 12AH	KT-12120	3	19.60	58.80	Ordered	EE
Batteryspace	Battery Charger	CH-LA1210UL	1	15.95	15.95	Ordered	EE
				235.83	207.63		
				Total	3,000.00		

Figure 21: Procurement of Robo-Weeder.

Appendix D: Gantt Charts

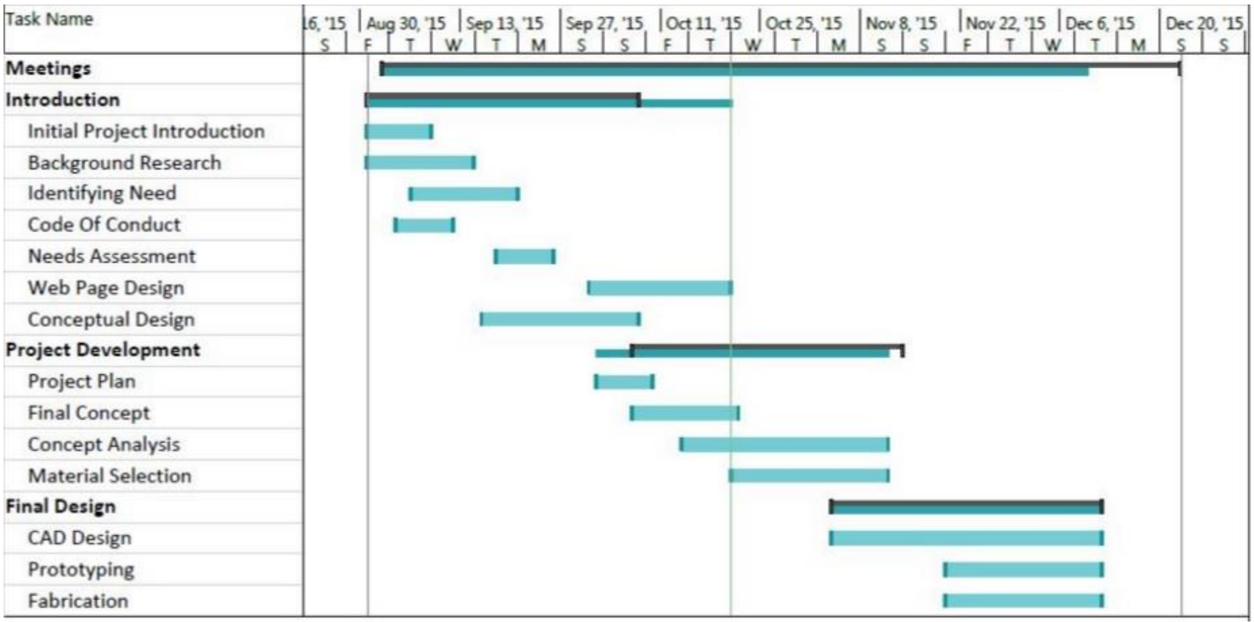


Figure 22: Fall Gantt Chart.

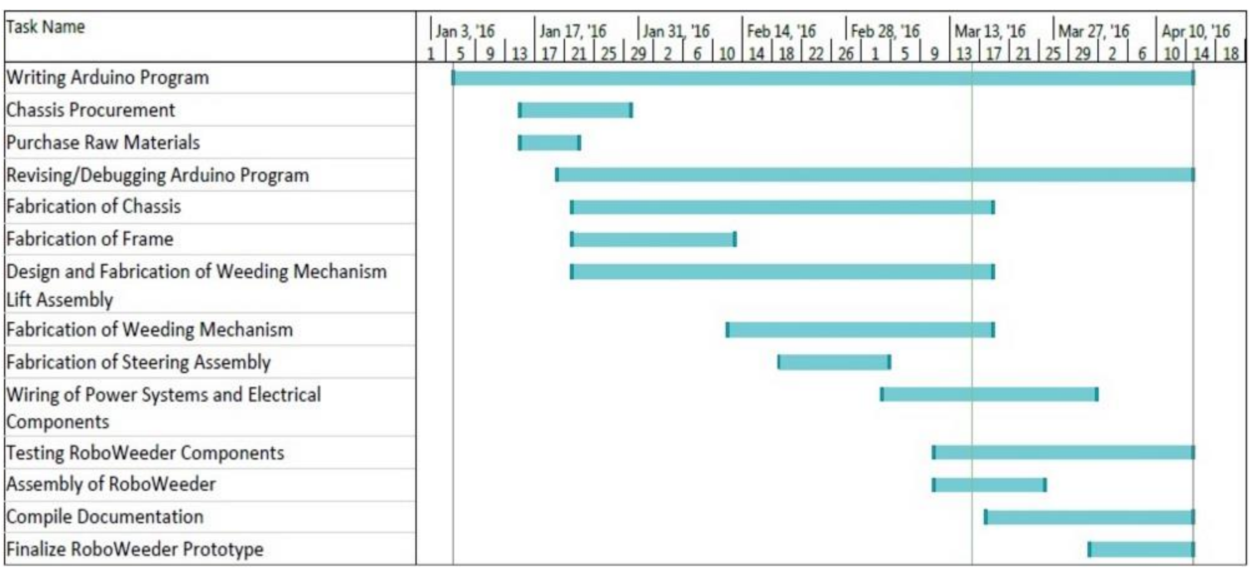


Figure 23: Spring Gantt Chart.

Appendix E: Auger Housing Exploded View and Disassembly

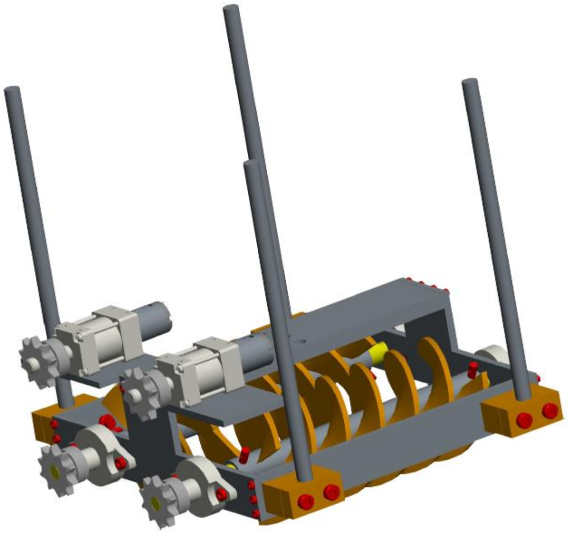


Figure 24: Completely assembled auger assembly.

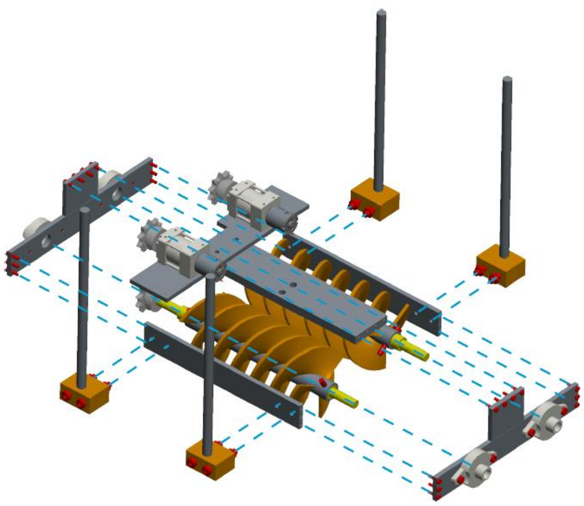


Figure 25: Completely disassembled auger assembly.

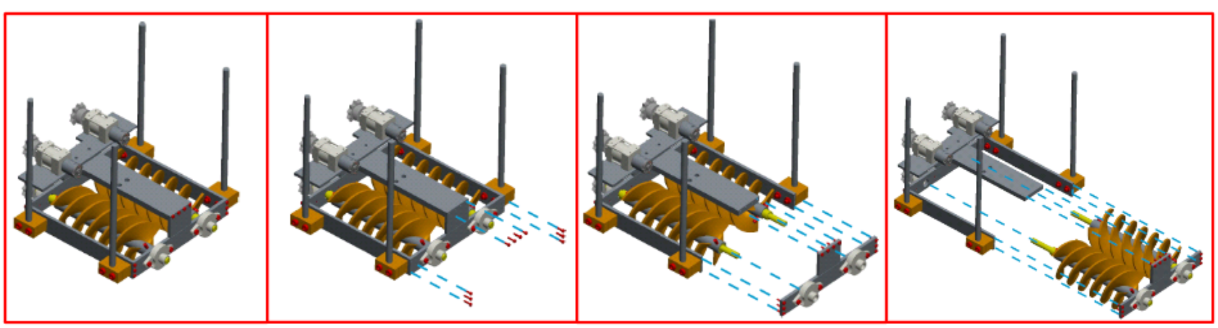


Figure 26: Step-by-step removal of Auger from Auger assembly.

Appendix F: Master Code

```
/*  Robo-Weeder
 *  Master Code
 */
int analogOutput = 0;          //##############################################FRONT- Steering
int encoder_value = 0;
int n = 0; //counter
int rc = 0;
int A = 0; // Encoder spin
int a = 0;
int b = 0;
int c = 0;
int d = 0;
int e = 0;
int pulse = 0;
int z = 0;  // reference turn
int ref = 0;
int Rcounter = 0;
int Lcounter = 0;
int x = 0;  // counter stop
int CC = 0;
int CCW = 0;
int count = 0;
int ReturnCounter = 0;
int CR = 0;  //turn completion left side
int CL = 0;  //turn completion right side
//int pulse = 0; //counter pulse

int B = 0;
int C = 0;
int cc = 40; //center


int rcpin1 = 31;  // input pin from rc receiver channel 1 Front Steering
int rcpin2 = 33;  // input pin from rc receiver channel 2 Drive
int rcpin4 = 35;  // input pin from rc receiver channel 4 Rear Steering
int rcpin5 = 37;  // input pin from rc receiver channel 5 Auger
int rcpin6 = 39;  // input pin from rc receiver channel 6 Linear Actuator

#include <Servo.h>
#define MIN 1410
#define MAX 1590
#define STOP 1500
Servo myservo1; // create servo object to control a RoboClaw channel
int ch1;  //Varibles to store and display the values of each channel
int mapCh1; //ch1 mapped to 255

int analogOutputR = 0;          //##############################################REAR- Steering
int encoder_valueR = 0;
int nR = 0; //counter
int rcR = 0;
int AR = 0; // Encoder spin
int aR = 0;
```

```
int bR = 0;
int cR = 0;
int dR = 0;
int eR = 0;
int pulseR = 0;
int zR = 0;  // reference turn
int refR = 0;
int RcounterR = 0;
int LcounterR = 0;
int xR = 0;  // counter stop
int CCR = 0;
int CCWR = 0;
int countR = 0;
int ReturnCounterR = 0;
int CRR = 0;  //turn completion left side
int CLR = 0;  //turn completion right side

int BR = 0;
//int C = 0;
int ccR = 40; //center

Servo myservo2; // create servo object to control a RoboClaw channel
int ch4;  //Varibles to store and display the values of each channel
int mapCh4; //ch1 mapped to 255

//##############################DRIVE
#include <Servo.h>
#define MIN 1400
#define STOP 1500
#define MAX1 1600
#define MAX2 1675
#define MAX3 1750
Servo myservo3; // Drive Motor 1
Servo myservo4; // Drive Motor 2
int posd = 0; // variable to store the servo position
int ch2;  //Varibles to store and display the values of each channel

#define encoder0PinAD  2  // figure out mega interrupt
#define encoder0PinBD  3  // figure out mega interrupt

volatile long encoder0PosD=0;
long newpositionD;
long oldpositionD = 0;
unsigned long newtimeD;
unsigned long oldtimeD = 0;
long velD;

//#############################AUGER
Servo myservo5; // Drive Motor 1
Servo myservo6; // Drive Motor 2
int pos = 0; // variable to store the servo position
int ch5;  //Varibles to store and display the values of each channel

#define encoder0PinA  18  // figure out mega interrupt
#define encoder0PinB  19  // figure out mega interrupt
```

```
volatile long encoder0Pos=0;
long newposition;
long oldposition = 0;
unsigned long newtime;
unsigned long oldtime = 0;
long vel;

void setup()          //##########################################VOID SETUP
{
pinMode(A1, INPUT);   //Encoder front Steering ##############STEERING
pinMode(rcpin1, INPUT);
myservo1.attach(4); // attaches the RC signal on pin 2 to the servo object
pinMode(A2, INPUT);//  pinMode(encoder_pin, INPUT);
pinMode(rcpin4, INPUT);   //Encoder rear Steering
myservo2.attach(5); // attaches the RC signal on pin 3 to the servo object
Serial.begin(115200);

  pinMode(encoder0PinAD, INPUT);      //####################DRIVE
  digitalWrite(encoder0PinAD, HIGH);  // turn on pullup resistor
  pinMode(encoder0PinBD, INPUT);
  digitalWrite(encoder0PinBD, HIGH);    // turn on pullup resistor
  attachInterrupt(0, doEncoderD, RISING);  // encoDER ON PIN 2 3

  pinMode(rcpin2, INPUT);
  myservo3.attach(6); // attaches the RC signal on pin 4 to the servo object
  myservo4.attach(7); // attaches the RC signal on pin 5 to the servo object

  pinMode(encoder0PinA, INPUT);      //######################AUGER
  digitalWrite(encoder0PinA, HIGH);       // turn on pullup resistor
  pinMode(encoder0PinB, INPUT);
  digitalWrite(encoder0PinB, HIGH);       // turn on pullup resistor
  attachInterrupt(0, doEncoder, RISING);  // encoDER ON PIN 18 19

  pinMode(rcpin5, INPUT);
  myservo5.attach(8); // attaches the RC signal on pin 6 to the servo object
  myservo6.attach(9); // attaches the RC signal on pin 7 to the servo object
}

void loop()          //####################################################LOOP BODY
{
 analogOutput = (analogRead(A1));  //encoder ADC analog 1024####################### FRONT Loop
 A = analogOutput;
 encoder_value = map(analogOutput, 0, 1024, 0, 255);// encoder_value = pulseIn(5, HIGH);

  Serial.print("Encoder: ");
  Serial.print(A);//encoder ADC analog
  Serial.write(10); //ascii line feed
  ch1 = pulseIn (rcpin1,HIGH);  //Read and store channel 1

  Serial.print ("Ch1:");  //Display text string on Serial Monitor to distinguish variables
  Serial.println (ch1);     //Print in the value of channel 1
if (CC == 1)         //###########################################################ClockWise
{
  if ((A >= 0) && (A <= 1024))            // CENTER
  {
   if ((A > 0) && (A <= 512))
```

```
  { a = a + 1;

     if ((a == 2) && (x != 1))          // will not stop counting when not running
      { count = count + 1;
        a = 0;
        x = 1;  // counter stop when rotation stop
  }  else if ((a >= 1) && (b >= 1))
       {  count = count + 1;
          a = 0;
          b = 0;
          n = 0;
          x = 0; }

  }
  if ((A > 512) && (A <= 1024))
  { b = b + 1;
    if ((x != 1) && (b == 2))
      { count = count + 1;
       b = 0;
       x = 1;  // counter stop
  } else if ((a >= 1) && (b >= 1))
       {  count = count + 1;
          a = 0;
          b = 0;
          n = 0;
          x = 0; }
  }
           Serial.print ("CC: ");
           Serial.println (count) ;
 }   }
 if (CCW == 1)        //#################################################Counter-ClockWise
{
 if ((A >= 0) && (A <= 1024))            // CENTER
 {
  if ((A > 0) && (A <= 512))
  { a = a + 1;

     if ((a == 2) && (x != 1))
      { count = count - 1;
        a = 0;
        x = 1;  // counter stop when rotation stop
  }  else if ((a >= 1) && (b >= 1))
       {  count = count - 1;
          a = 0;
          b = 0;
          n = 0;
          x = 0; }

  }
  if ((A > 512) && (A <= 1024))
  { b = b + 1;
    if ((x != 1) && (b == 2))
      { count = count - 1;
       b = 0;
       x = 1;  // counter stop
  } else if ((a >= 1) && (b >= 1))
```

```
        {  count = count - 1;
           a = 0;
           b = 0;
           n = 0;
           x = 0; }
  }
            Serial.print ("CCW: ");
            Serial.println (count) ;
}   }
if (ch1 == 0)
  {  myservo1.writeMicroseconds(STOP); //stop
  }
if ((ch1 >1400) && (ch1 <1505))        //###############################################CENTER
  { CR = 1;
    CL = 1;
    if (cc != 40)
    {   z = 40 - ref; //calculate logic
        if (z < 0)
          { CCW = 1;  //CCW countdown
           myservo1.writeMicroseconds(MIN);
           Serial.print ("ReturnCounter: ");
           Serial.println (count) ;

        if (count == 0)  // pulse = 20, 3 pulse/rev  (20/3 = 10 degrees)
          { myservo1.writeMicroseconds(STOP);
           ReturnCounter = 0;
           CCW = 0;  //CCW countdown
           cc = 40;  //transmitter position
           ref = 40; //pulse reference
          }   }
        else
          { CCW = 1;  //CCW countdown
           myservo1.writeMicroseconds(MAX);
           Serial.print ("ReturnCounter: ");
           Serial.println (count) ;

        if (count == 0)  // pulse = 20, 3 pulse/rev  (20/3 = 10 degrees)
          { myservo1.writeMicroseconds(STOP);
           CCW = 0;   //CCW countdown
           cc = 40;   //transmitter position
           ref = 40;  //pulse reference
          }
          }

  }   else { myservo1.writeMicroseconds(STOP);   // stop
            count = 0;     }           //####################################### manual steering

  }
 //
if ((ch1 >= 1505) && (ch1 <2000) && (CR == 1))      //############################################RIGHT TURN
  { if (cc != 50)

     {  z = 50 - ref;
           CC = 1;
           myservo1.writeMicroseconds(MAX);
           Serial.print ("Right Turn: ");
```

```
            Serial.println (count) ;

        if (count == 5)  // pulse = 20, 3 pulse/rev  (20/3 = 10 degrees)
          { myservo1.writeMicroseconds(STOP);
          //  ReturnCounter = 18;
           // count = 0;
            CC = 0;
            cc = 50;
            CL = 1;   //Turn left enable
            ref = 50; //pulse reference
           }  }
      else {myservo1.writeMicroseconds(STOP);  }
    }
  if ((ch1 >= 950) && (ch1 <= 1400) && (CL == 1))     //##############################################LEFT TURN
    { if (cc != 30)

      {  z = 30 - ref;
            CC = 1;
            myservo1.writeMicroseconds(MIN);
            Serial.print ("Left Turn: ");
            Serial.println (count) ;

        if (count == 5)  // pulse = 20, 3 pulse/rev  (20/3 = 10 degrees)
          { myservo1.writeMicroseconds(STOP);
          //  ReturnCounter = 18;
           // count = 0;
            CC = 0;
            cc = 30;
            CR = 1;   //Turn right enable
            ref = 30; //pulse reference
           }  }
      else { myservo1.writeMicroseconds(STOP);  }
    }
 analogOutputR = analogRead(A2);  //encoder ADC analog 1024
 AR = analogOutputR;
 encoder_valueR = map(analogOutputR, 0, 1024, 0, 255);// encoder_value = pulseIn(5, HIGH);

  Serial.print("EncoderR: ");
  Serial.print(AR);//encoder ADC analog
  Serial.write(10); //ascii line feed
  ch4 = pulseIn (rcpin4,HIGH);  //Read and store channel 1

  Serial.print ("Ch4:");  //Display text string on Serial Monitor to distinguish variables
  Serial.println (ch4);     //Print in the value of channel 1
if (CCR == 1)         //########################################################ClockWise
{
  if ((AR >= 0) && (AR <= 1024))            // CENTER
  {
   if ((AR > 0) && (AR <= 512))
    { aR = aR + 1;

       if ((aR == 2) && (xR != 1))          // will not stop counting when not running
         { countR = countR + 1;
           aR = 0;
           xR = 1;  // counter stop when rotation stop
    }  else if ((aR >= 1) && (bR >= 1))
```

```
        {  countR = countR + 1;
           aR = 0;
           bR = 0;
           nR = 0;
           xR = 0; }

   }
   if ((AR > 512) && (AR <= 1024))
   { bR = bR + 1;
     if ((xR != 1) && (bR == 2))
       { countR = countR + 1;
        bR = 0;
        xR = 1;  // counter stop
   } else if ((aR >= 1) && (bR >= 1))
        {  countR = countR + 1;
           aR = 0;
           bR = 0;
           nR = 0;
           xR = 0; }
   }
             Serial.print ("CCR: ");
             Serial.println (countR) ;
 }  }
 if (CCWR == 1)        //################################################Counter-ClockWise
{
 if ((AR >= 0) && (AR <= 1024))            // CENTER
 {
  if ((AR > 0) && (AR <= 512))
   { aR = aR + 1;

     if ((aR == 2) && (xR != 1))
       { countR = countR - 1;
         aR = 0;
         xR = 1;  // counter stop when rotation stop
   }  else if ((aR >= 1) && (bR >= 1))
        {  countR = countR - 1;
           aR = 0;
           bR = 0;
           nR = 0;
           xR = 0; }

   }
   if ((AR > 512) && (AR <= 1024))
   { bR = bR + 1;
     if ((xR != 1) && (bR == 2))
       { countR = countR - 1;
        bR = 0;
        xR = 1;  // counter stop
   } else if ((aR >= 1) && (bR >= 1))
        {  countR = countR - 1;
           aR = 0;
           bR = 0;
           nR = 0;
           xR = 0; }
   }
             Serial.print ("CCWR: ");
```

```
            Serial.println (countR) ;
}  }
if (ch4 == 0)
  {  myservo2.writeMicroseconds(STOP); //stop
  }
if ((ch1 >1400) && (ch1 <1505))        //################################################CENTER
  { CR = 1;
    CL = 1;
    if (cc != 40)
    {   z = 40 - ref; //calculate logic
        if (z < 0)
          { CCW = 1;  //CCW countdown
            myservo2.writeMicroseconds(MIN);
            Serial.print ("ReturnCounter: ");
            Serial.println (count) ;

        if (count == 0)  // pulse = 20, 3 pulse/rev  (20/3 = 10 degrees)
          { myservo2.writeMicroseconds(STOP);
           ReturnCounter = 0;
            CCW = 0;  //CCW countdown
            cc = 40;  //transmitter position
            ref = 40; //pulse reference
          }   }
        else
          { CCW = 1;  //CCW countdown
            myservo2.writeMicroseconds(MAX);
            Serial.print ("ReturnCounter: ");
            Serial.println (count) ;

        if (count == 0)  // pulse = 20, 3 pulse/rev  (20/3 = 10 degrees)
          { myservo2.writeMicroseconds(STOP);
            CCW = 0;   //CCW countdown
            cc = 40;   //transmitter position
            ref = 40;  //pulse reference
          }
          }

  }   else { myservo2.writeMicroseconds(STOP);   // stop
           count = 0;     }           //###################################### manual steering

  }
 //
if ((ch4 >= 1505) && (ch4 <2000) && (CR == 1))      //############################################RIGHT TURN
  { if (cc != 50)

     {  z = 50 - ref;
            CC = 1;
            myservo2.writeMicroseconds(MAX);
            Serial.print ("Right Turn: ");
            Serial.println (count) ;

        if (count == 5)  // pulse = 20, 3 pulse/rev  (20/3 = 10 degrees)
         { myservo2.writeMicroseconds(STOP);
         //  ReturnCounter = 18;
          // count = 0;
            CC = 0;
```

```
            cc = 50;
            CL = 1;   //Turn left enable
            ref = 50; //pulse reference
          }  }
      else {myservo2.writeMicroseconds(STOP);  }
    }
 if ((ch4 >= 950) && (ch4 <= 1400) && (CL == 1))     //##############################################LEFT TURN
   { if (cc != 30)

      {  z = 30 - ref;
            CC = 1;
            myservo2.writeMicroseconds(MIN);
            Serial.print ("Left Turn: ");
            Serial.println (count) ;

        if (count == 5)  // pulse = 20, 3 pulse/rev  (20/3 = 10 degrees)
         { myservo2.writeMicroseconds(STOP);
         //  ReturnCounter = 18;
          // count = 0;
            CC = 0;
            cc = 30;
            CR = 1;   //Turn right enable
            ref = 30; //pulse reference
          }  }
      else { myservo2.writeMicroseconds(STOP);  }

             }
 newpositionD = encoder0PosD;   //###########################LOOP DRIVE
 newtimeD = millis();
 velD = (newpositionD-oldpositionD) * 1000 /(newtimeD-oldtimeD);
 Serial.print ("speed = ");
 Serial.println (velD);
 oldpositionD = newpositionD;
 oldtimeD = newtimeD;
 delay(250);
 ch2 = pulseIn (rcpin2,HIGH);  //Read and store channel 2
 Serial.print ("Ch2:");
 Serial.println (ch2);

 if (ch2 == 0)
   {  myservo3.writeMicroseconds(STOP);
      myservo4.writeMicroseconds(STOP); }    //stop
 if ((ch2 >1475) && (ch2 <1505))
   {  myservo3.writeMicroseconds(STOP);
      myservo4.writeMicroseconds(STOP);}     //stop
 if ((ch2 >= 1505) && (ch2 <1700))
   {  myservo3.writeMicroseconds(MAX1);
      myservo4.writeMicroseconds(MAX1); }    // forward 30%
 if ((ch2 >= 1700) && (ch2 <1850) )
   {  myservo3.writeMicroseconds(MAX2);
      myservo4.writeMicroseconds(MAX2); }    // forward 60%
if ((ch2 >= 1850) && (ch2 <2000) )
   {  myservo2.writeMicroseconds(MAX3);
      myservo3.writeMicroseconds(MAX3); }    // forward 100%
if ((ch2 > 0) && (ch2 <= 1475))
   {  myservo3.writeMicroseconds(MIN);
```

```
     myservo4.writeMicroseconds(MIN); }     //full reverse

  newposition = encoder0Pos;     //###########################LOOP AUGER
  newtime = millis();
  vel = (newposition-oldposition) * 1000 /(newtime-oldtime);
  Serial.print ("speed = ");
  Serial.println (vel);
  oldposition = newposition;
  oldtime = newtime;
  delay(250);
  ch5 = pulseIn (rcpin5,HIGH);  //Read and store channel 5
  Serial.print ("Ch5:");
  Serial.println (ch5);

 if (ch5 == 0)
   {  myservo5.writeMicroseconds(STOP);
      myservo6.writeMicroseconds(STOP);}    //stop
 if ((ch5 >1475) && (ch5 <1505))
   {  myservo5.writeMicroseconds(STOP);
      myservo6.writeMicroseconds(STOP);}    //stop
 if ((ch5 >= 1505) && (ch5 <1700))
   {  myservo5.writeMicroseconds(MAX1);
      myservo6.writeMicroseconds(MAX1); }    // forward 30%
 if ((ch5 >= 1700) && (ch5 <1850) )
   {  myservo5.writeMicroseconds(MAX2);
      myservo6.writeMicroseconds(MAX2);}     // forward 60%
if ((ch5 >= 1850) && (ch5 <2000) )
   {  myservo5.writeMicroseconds(MAX3);
      myservo6.writeMicroseconds(MAX3);}     // forward 100%
if ((ch5 > 0) && (ch5 <= 1475))
   {  myservo5.writeMicroseconds(MIN);
      myservo6.writeMicroseconds(MIN); }     //full reverse
  }                          //########################LOOP BODY

  void doEncoderD()   //##################InterruptCall DRIVE
{
  if (digitalRead(encoder0PinAD) == digitalRead(encoder0PinBD)) {
    encoder0PosD++;
  } else {
    encoder0PosD--;
  }  }
 void doEncoder()    //##################InterruptCall AUGER
{
  if (digitalRead(encoder0PinA) == digitalRead(encoder0PinB)) {
    encoder0Pos++;
  } else {
    encoder0Pos--;
  }  }
```