TEXAS A&M UNIVERSITY

J. Mike Walker '66 Department of
Mechanical Engineering

Garden Cultivation Robot Final Report

Authors: Linus Chin, Alvaro Guerra, Mason Grimsley, Ryan Li, Giselle Martinez

Sponsor Name: Dr. Matthew Elliott

MEEN 402-511: Dr. Suh

4/29/2025

Executive Summary

The Garden Cultivation Robot was developed to address the physical challenges associated with home gardening, especially for elderly and mobility-limited individuals. The team's goal was to create a modular, remotely-controlled robot capable of traversing a 50-foot garden row while delivering water and fertilizer to plants. Key design goals included modularity, maneuverability, durability, and ease of use.

The final prototype successfully achieved its core objectives. The robot features a modular aluminum 8020 chassis with ABS and PETG enclosures, a twin DC motor drivetrain using chain and sprocket assemblies, and two modular subsystems for watering and fertilizing. The electronic control system was built around a dual-Arduino master-slave architecture, ensuring responsive wireless remote control with minimal signal latency. Failure risk assessments, including FMECA and Fault Tree Analysis, identified drivetrain misalignment and battery depletion as key concerns; these risks were addressed through design modifications, rigorous mechanical assembly, and iterative electronic debugging. The robot met major performance targets, including a 0.01 m turning radius, a 26% improvement in consumable volume, and a final project cost of $2,301.97, remaining within the approved contingency budget.

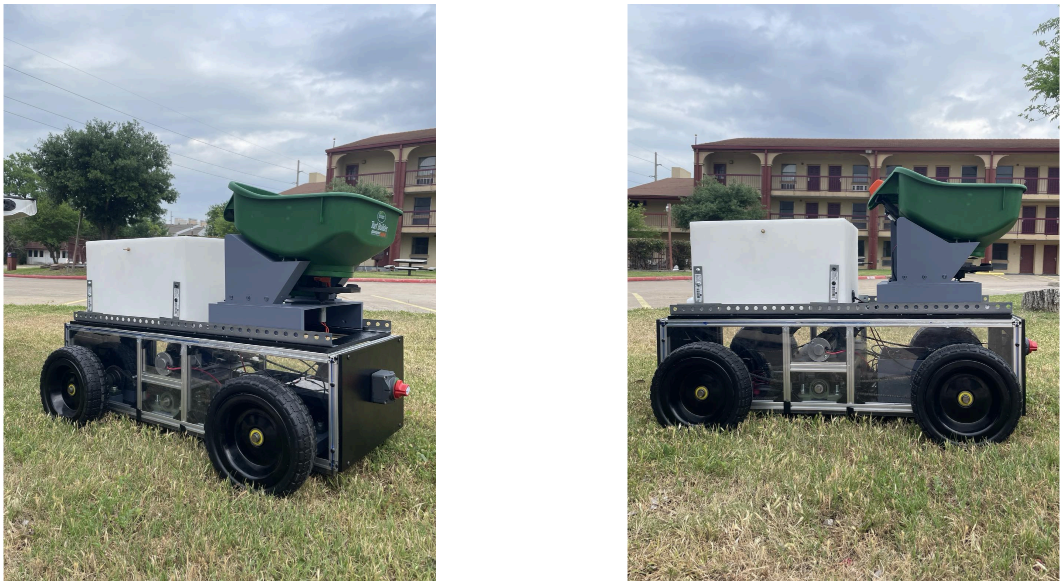


Figure 1: Final Assembly of the Garden Cultivation Robot with Subsystems Attached

Future work includes upgrading the robot's battery system to a 24V $LiFePO_4$ pack for extended run time, installing spliced wire connectors to streamline subsystem swapping, and expanding toward autonomous capabilities through the integration of cameras, soil moisture sensors, ultrasonic row trackers, and GPS waypoints. Longer-term visions include developing a machine learning-based weeding subsystem to further enhance garden maintenance automation.

Table of Contents

Executive Summary....................1
Table of Contents....................3
Glossary Table....................4
1.1 Need Analysis....................7
1.2 Design Parameter Identification.................... 7
1.3 Function Structure....................9
2. System Description....................10
2.1 Design Decisions.................... 10
2.2 Prototype Architecture.................... 11
2.3 Critical Interfaces....................25
3. Analysis for Design.................... 26
3.1.1 Embodiment Design Checklist.................... 26
3.1.2 FMECA....................28
3.1.3 Fault Tree Analysis....................30
3.1.4 Standards and Codes....................31
3.2 DFMA Considerations....................32
3.3 Design Validation (could use more work).................... 33
3.4 Design Requirements Traceability Matrix....................35
3.5 Cost Accounting and Cost Model....................36
4. Broader Impacts of Design....................38
4.1 Design for Lifecycle.................... 38
4.2 Intellectual Property....................39
4.3 Liability Considerations....................39
4.4 Ethical Considerations.................... 40
5. Summary....................41
5.1 Work Breakdown Structure....................41
5.2 Final Gantt Chart....................42
5.3 Technological Development.................... 44
5.4 Design Limitations....................44
5.5 Future Work.................... 45
6. Acknowledgements.................... 46
References....................47
Appendix....................49

Glossary Table

Term	**Definition**
ANSI	American National Standards Institute
ASME	American Society of Mechanical Engineers
BOM	Bill of Materials
CAD	Computer Aided Design
CG	Center of Gravity
CNC	Computer Numerical Control
DC	Direct Current
DFMA	Design for Manufacturing and Assembly
DFx	Design for “x”
DoE	Design of Experiments
FMECA	Failure Mode, Effects, and Criticality Analysis
FTA	Fault Tree Analysis

GPS	Global Positioning System
HOQ	House of Quality
IEC	International Electrotechnical Commission
IMU	Inertial Measurement Unit
ISO	International Organization for Standardization
LDS	Laser Distance Sensor
LiDAR	Light Detection and Ranging
MEEN	Mechanical Engineering
MTBF	Mean Time Between Failure
NFPA	National Fire Protection Association
OSHA	Occupational Safety and Health Administration
PID	Proportional-Integral-Derivative
QC	Quality Control

RC	Remote Control
RPN	Risk Priority Number
SNPS	Solution Neutral Problem Statement
WBS	Work Breakdown Structure

1. Introduction and Problem Definition

1.1 Need Analysis

Home gardening, while a rewarding activity, often poses some physical challenges, particularly for elderly individuals and those with limited mobility. Essential tasks such as watering, fertilizing, and weeding require repetitive bending and sustained periods of manual labor, increasing the risk of fatigue and injury. In response to these challenges, the team proposed the development of a Gardening Robot designed to automate these activities, thus improving safety, efficiency, and accessibility for home gardeners.

The customer needs were established through the initial problem statement and discussions with the team sponsor, Dr. Matt Elliott. A successful design would feature a functional chassis capable of self-propelling along a 50-foot garden row while transporting and operating two modular subsystems. "Modular subsystems" refer to detachable units that execute specific gardening tasks, such as watering or fertilizing, while the "functional chassis" denotes a robust yet lightweight frame capable of supporting mobility and secure subsystem attachment. The robot must satisfy several key requirements: it must self-propel across at least 50 feet of garden row, effectively deploy two modular subsystems, navigate varying terrain conditions, and operate reliably through remote control input.

The functional decomposition of the robot begins with understanding its top-level functions, each critical to meeting the project objectives. The powertrain is responsible for supplying sufficient energy to drive both the chassis and the subsystems. The drivetrain ensures reliable and controlled movement across the garden row, accommodating minor variations in surface conditions. The modular subsystem interface enables the attachment, secure mounting, and operation of at least two distinct modules. Finally, the control and interface system allows the user to operate the robot intuitively through wired or wireless remote control. Collectively, these functions form the foundation of the overall system architecture, ensuring that energy management, mobility, modularity, and user interaction are fully addressed in the final design.

1.2 Design Parameter Identification

To translate the functional decomposition into a realizable system, it was necessary to define clear, quantifiable design parameters for each major function of the Gardening Robot. These parameters serve as measurable targets that ensure the final design not only fulfills the intended functions but also meets specific performance targets. Identifying these parameters early in the design process was crucial in understanding the most effective path for assembly and iteration.

For the powertrain, the primary design parameters include motor power output, energy storage capacity, and overall drivetrain efficiency. Each motor must sustain a minimum output of 100 watts to enable reliable movement across the 50-foot garden row, accounting for terrain variability and system load. Battery selection was guided by the need to balance energy density and system weight, ensuring sufficient operational time without compromising the robot's maneuverability. The powertrain must also maintain high energy efficiency to minimize losses across operating conditions.

The drive system governs the robot’s mobility and maneuverability. Key parameters identified include wheel or track configuration, traction control, turning radius, ground clearance, and center of gravity height. Performance targets specify that the robot must reliably propel itself at least 50 feet of garden row, achieve a turning radius no greater than 0.9 meters to enable tight maneuvering within confined garden spaces, and maintain a center of gravity lower than 0.18 meters to minimize rollover risk on uneven terrain. Wheel selection and drivetrain gearing must be tuned to provide adequate grip and propulsion force without excessive slippage or loss of control.

For the modular subsystem interface, design parameters focus on attachment and detachment speed, as well as consumable volume. Each modular subsystem should be attachable or removable within 30 seconds without the use of specialized tools, supporting a modular architecture. Additionally, subsystems must achieve functional output metrics, such as delivering appropriate watering flow rates or uniform fertilizer distribution, as optimized for garden plant needs. Mechanical interfaces should also ensure secure mounting.

The control and interface system ensures that users can operate the robot reliably and intuitively. Critical parameters include system response time, communication stability, and overall usability of the interface. Commands issued by the user must be processed with a latency of less than 0.5 seconds to maintain responsive control during operation. Wireless communication links must remain robust over the operational range without significant signal degradation. The user interface itself should require minimal training, employing straightforward controls for basic navigation and subsystem deployment, thus enhancing accessibility.

By defining these parameters, the team created a foundation for a more structured design phase. This approach enabled the engineering team to make educated design trade-offs and subfunction validation to ensure that the final prototype delivers efficient operation across its intended use cases.

1.3 Function Structure

The function structure of the Gardening Robot is developed by breaking down each top-level function into its corresponding subsystems and detailed sub-functions. This structured approach ensures a clear understanding of how each component contributes to the robot's overall operation. At the highest level, the robot consists of four primary functional categories: powertrain, drive system, modular subsystem execution, and control/interface system. Each of these top-level functions is further decomposed to define the sub-functions necessary for successful task execution.

The powertrain supplies energy to all components of the robot. Its sub-functions include power generation (battery or fuel-based sources), power distribution, and energy management to optimize system efficiency. The subsystem must ensure that adequate energy is available to support mobility and modular operations throughout the full 50-foot travel distance.

The drive system enables the robot to travel the garden row efficiently. Its sub-functions include propulsion through motors or actuators, steering (differential drive or track control), and stability management (maintaining balance over uneven terrain). Further sub function decomposition addresses torque distribution, traction control, and speed regulation to meet mobility performance requirements.

The modular subsystem execution manages the deployment and operation of interchangeable gardening activities or tools. This function is subdivided into functions such as attachment mechanisms (quick-release or bolt-on designs), deployment mechanisms (fluid control or granular dispersion), and actuation components (pumps or mechanical spreaders). Each sub-function must ensure effective task execution within either specified time and volume constraints or user actuation.

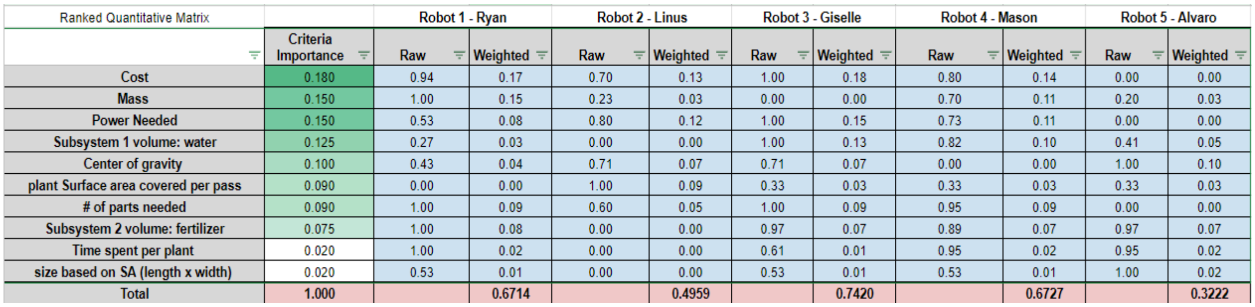
The control and interface system allows users to command and actuate the robot. It includes input processing (wired or wireless commands) or feedback response (additional sensor data). Further function subdivision includes communication protocol management and user interface development to provide an intuitive user experience.

2. System Description

2.1 Design Decisions

During MEEN 401, the team conducted a comprehensive design selection process utilizing Pugh matrices and quantitative matrices, as shown in **Table 1**, to systematically evaluate and rank preliminary design concepts against a defined set of criteria. Each team member submitted an initial robot design, which was assessed based on standardized evaluation metrics. Images of these preliminary sketches are provided in **Appendix 7-12**.

Table 1: Ranked Quantitative Matrix of five different robot concepts



Ranked Quantitative Matrix		Robot 1 - Ryan		Robot 2 - Linus		Robot 3 - Giselle		Robot 4 - Mason		Robot 5 - Alvaro	
	Criteria Importance	Raw	Weighted	Raw	Weighted	Raw	Weighted	Raw	Weighted	Raw	Weighted
Cost	0.180	0.94	0.17	0.70	0.13	1.00	0.18	0.80	0.14	0.00	0.00
Mass	0.150	1.00	0.15	0.23	0.03	0.00	0.00	0.70	0.11	0.20	0.03
Power Needed	0.150	0.53	0.08	0.80	0.12	1.00	0.15	0.73	0.11	0.00	0.00
Subsystem 1 volume: water	0.125	0.27	0.03	0.00	0.00	1.00	0.13	0.82	0.10	0.41	0.05
Center of gravity	0.100	0.43	0.04	0.71	0.07	0.71	0.07	0.00	0.00	1.00	0.10
plant Surface area covered per pass	0.090	0.00	0.00	1.00	0.09	0.33	0.03	0.33	0.03	0.33	0.03
# of parts needed	0.090	1.00	0.09	0.60	0.05	1.00	0.09	0.95	0.09	0.00	0.00
Subsystem 2 volume: fertilizer	0.075	1.00	0.08	0.00	0.00	0.97	0.07	0.89	0.07	0.97	0.07
Time spent per plant	0.020	1.00	0.02	0.00	0.00	0.61	0.01	0.95	0.02	0.95	0.02
size based on SA (length x width)	0.020	0.53	0.01	0.00	0.00	0.53	0.01	0.53	0.01	1.00	0.02
Total	1.000		0.6714		0.4959		0.7420		0.6727		0.3222

A key selection criterion was the power required for operation, with the team prioritizing designs that minimized energy consumption. Lower power requirements generally correlated with lighter designs, reduced material usage, and improved overall system efficiency. Additional evaluation criteria included total mass, center of gravity height, subsystem volume capacity, power consumption, time spent servicing each plant, plant surface area coverage, estimated cost, and overall system size. Given that many design parameters were still flexible at this stage, educated assumptions were made based on preliminary engineering analyses and reasonable extrapolations.

Following the weighted evaluation, Giselle's robot design which featured a tread drive system and adjacent modular subsystem layout achieved the highest overall score according to the quantitative matrix. After selection, further engineering calculations were performed to define specific part and interface requirements, allowing the team to iteratively refine the design into the finalized prototype model. Other key design tradeoffs are outlined in **Table 2**.

Table 2: Primary Design Tradeoffs

Topic	**Chosen Direction**	**Benefit**	**Tradeoff**
Drivetrain Layout	2 DC motors + chain drive	Simple control and assembly, lower cost	Less precise steering vs 4” wheel gearmotors
Mobility	10” off-road wheels	Good obstacle clearance and fits garden row width	Larger wheels raise CG and offer less traction than treads
Control System	Wireless RC	Increases user distance from robot	Connection is less reliable than simple wired connection
Chassis Enclosure	PETG and ABS side panels	Quick prototyping and quick installation to chassis panels	Lower relative corrosion resistance
Subsystem Scope	Watering and Fertilizing	Focuses effort, ensures two functions are fulfilled	Pruning/weeding options are not analyzed for use

2.2 Prototype Architecture

The robot as a whole entity can be subdivided into its core functions: the chassis and enclosure, power, drivetrain, subsystems, electronics, and software. Section 2.2 will also be divided as such. Here, descriptions of each core function’s construction, usage, implementation, design challenges/changes, and figures will be provided.

2.2.1 Chassis and Enclosure

The chassis of the robot serves as the primary structural framework upon which all subsystems are mounted and supported. Given its critical role in ensuring mechanical integrity and protection of internal components, the chassis was designed for high strength and modularity while minimizing weight and simplifying the assembly process.

The frame of the chassis is constructed from 6063-T6 aluminum slotted bars, utilizing 8020 extrusion profiles for ease of assembly, structural rigidity, and compatibility with modular fastening systems.

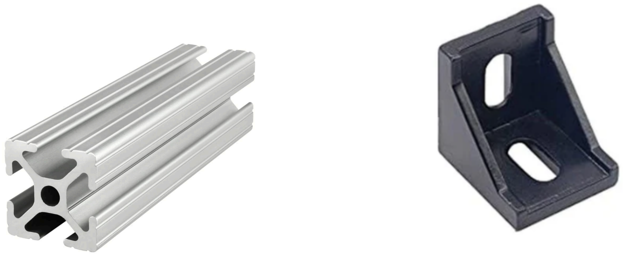


Figure 2: Typical 8020 extrusion used in chassis frame (left); typical corner bracket for securing adjacent extrusions (right)

The outer dimensions of the frame measure 36 inches by 16 inches by 10 inches.. The aluminum bars were cut to specific lengths to form the chassis geometry: six pieces at 36 inches, four pieces at 16 inches, eight pieces at 8 inches, fourteen pieces at 6.5 inches, and four pieces at 4 inches. The bottom frame was assembled using three 36-inch bars, two 16-inch bars, and six 6.5-inch bars, arranged and fastened using M5 bolt aluminum corner brackets to ensure dimensional symmetry. The top frame was constructed using an identical configuration.

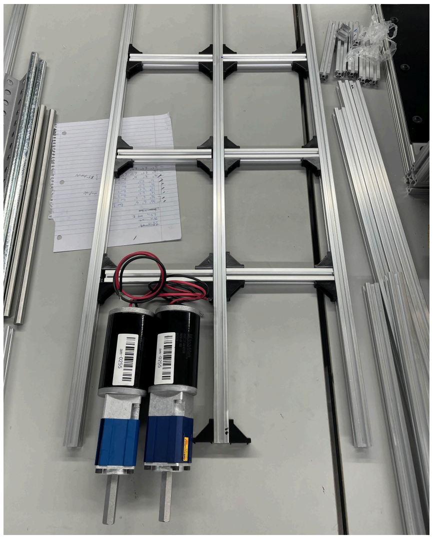


Figure 3: Bottom chassis assembly

Hand-calculations ensured that the 8020 bars would be able to withstand the bending moments induced by the static loading of the subsystems. Assuming that the weight of a fully water tank is roughly 90 lbf (~10 gallons), and the length of the distributed load is roughly 18 inches, then the bending moment is 135 lb-ft. This value was then incorporated into a bending stress calculation:

$$\sigma = \frac{Mc}{I}$$

The moment of inertia *I* can also be calculated:

$$I_{centroidal} = 0.0833\ in^4$$

$$I = I_{centroidal} + Ad^2 = 1.607 * 10^{-5} ft^4$$

This yields a bending stress of around 33.5 MPa, far below the 120 MPa yield strength of 6063 aluminum. Vertical supports were completed using the eight 8-inch bars to connect the top and bottom assemblies, creating a rigid rectangular prism, as seen in **Figure 4**.

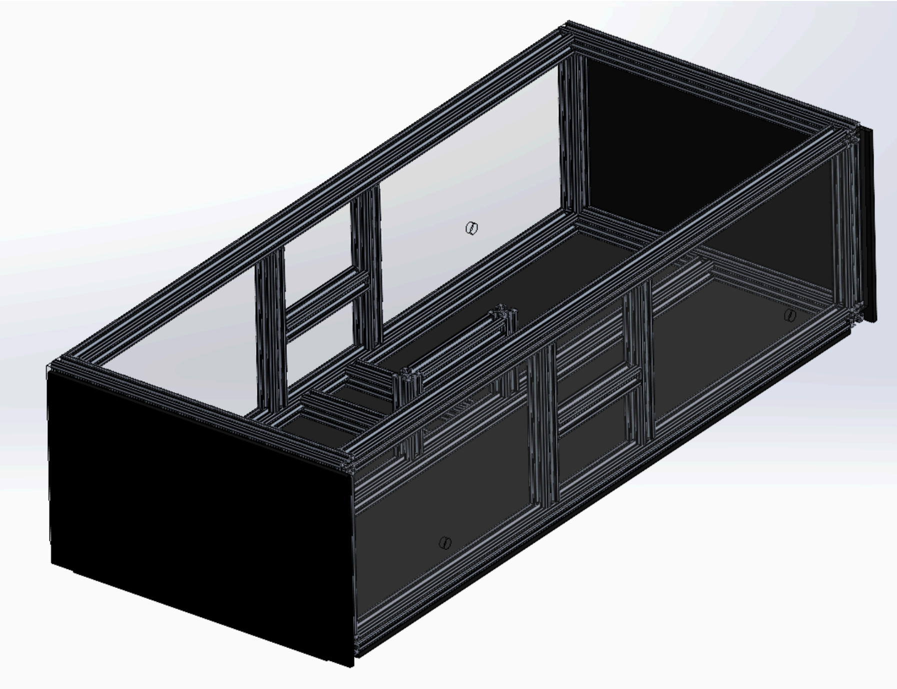


Figure 4: Isometric view of CAD model of chassis assembly with plastic sheets attached

For the enclosure, the chassis is lined with ⅝-inch black ABS plastic sheets on the bottom, top, front, and back faces. The ABS panels were secured directly to the aluminum frame to provide an element-resistant barrier, shielding sensitive internal components such as the drivetrain and electronics from dust and debris. On all the side sheets, four M5 holes were drilled into each corner for attachment. The side panels are constructed from clear ⅛-inch PETG sheets, offering both visibility into the chassis interior and additional lateral protection without significant weight increase. The initial enclosure step involved cutting and attaching the ABS sheet to the bottom of the chassis, while additional plastic panels were installed on all remaining faces to complete the full enclosure after the internal components were secured.

Motor mounts were strategically offset vertically within the chassis to accommodate the size of the drive motors without interfering with each other or internal subsystems. In **Figure 4**, this motor mount sub-assembly can be seen along the chassis mid-length. This design choice allows the motor and gearbox to fit within the width constraint while also maintaining a low center of gravity. Overall, the chassis and enclosure system were designed with a strong emphasis on structural resilience and ease of assembly, allowing for a relatively robust foundation for the robot's drivetrain and subsystems.

2.2.2 Power

The powertrain system of the garden robot is primarily responsible for supplying the necessary torque for propulsion. Specifically, the design decisions behind motor selection involved having enough torque and speed to propel the robot across a 50-ft garden row while maintaining energy efficiency and system robustness.

To determine the power requirements, some preliminary calculations were done based on the weight of the robot and the environment in which it would be operating. For a roughly 150 pound robot with a rolling resistance coefficient of 0.05 for firm soil, the force required is 3.75 lbf/motor, assuming a dual motor setup. Thus, the torque required for 10 inch wheels is 1.56 lb-ft. Since the wheels should spin at roughly 100 rpm (3 mph), the gear ratio is 3:1. Then, the motor torque can be calculated using the following:

$$T_{motor} = (2 * 1.56\, lb\, ft)/(3{:}1) = 1.04\, lb\, ft$$

The power can then be calculated using the following:

$$P = \frac{(1.56\, lb\, ft)(100\, RPM)}{5252} = 0.03\, hp/wheel = 22.06\, W$$

This means that each motor, if connected to two wheels, should be rated for at least $2 * 22.06 = 44.12\,W$ at 300 RPM, with a torque rating of at least 1.04 lb-ft. Incorporating a factor of safety of 2, each motor should be able to support roughly 90 W of continuous power, and 2.08 lb-ft of torque under standard operating conditions.

Based on these calculations, the chosen motor is a 2.5 in. diameter CIM brushed DC motor from AndyMark, selected for its relative affordability and high performance metrics. Each motor features a free speed of 5310 RPM, a maximum continuous power output of 337 W, and a stall torque of 1.79 ft-lb.

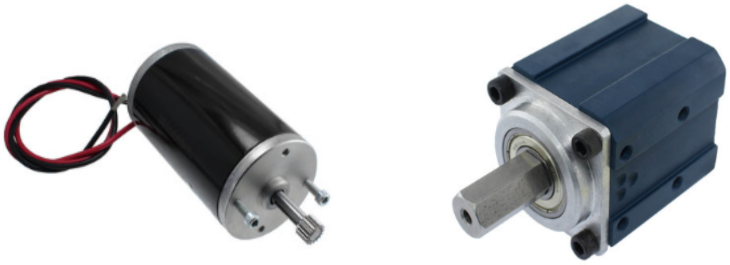


Figure 5: 2.5 in. AndyMark brushed DC motor (left), 16:1 planetary gearbox (right)

To achieve the necessary torque and wheel speed, a 16:1 planetary gearbox compatible with the motor was integrated into the powertrain, significantly amplifying torque to 28.62 lb-ft while also reducing the free speed to 331.9 RPM. This gearing strategy allowed the motors to operate within the desired speed-torque regime.

One of the key design challenges was balancing the torque and speed requirements without overloading the motors or introducing excessive mechanical complexity. Initial sizing assumptions underestimated the rolling resistance encountered in soft garden soil, prompting an increase in the target motor torque and incorporation of a higher gear reduction. A significant adjustment was made to the chassis layout to accommodate the large size of the gearbox-motor assemblies while maintaining a low center of gravity to preserve stability.

The motor layout itself is a double stacked setup, where one motor is mounted directly above the other via the powertrain chassis assembly described in **Section 2.2.1**.

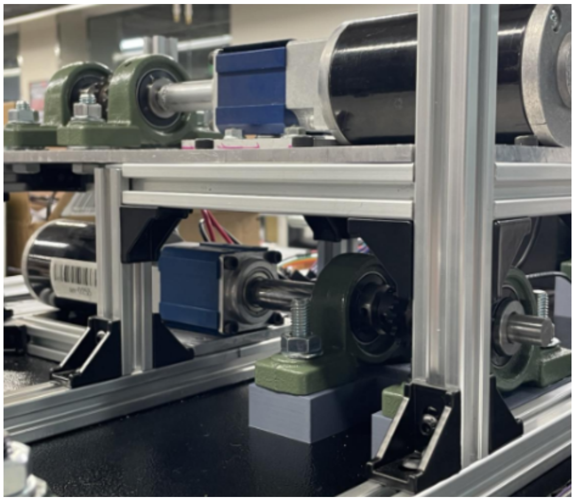


Figure 6: Stacked motor layout during assembly process

Each gearbox has four fastening points for M4 bolts. As seen in **Figure 6**, both top and bottom motors are secured to a custom cut and drilled aluminum sheet, which themselves are mounted on the chassis assembly’s 8020 extrusions.

2.2.3 Drivetrain

The Gardening Robot's drivetrain was designed to deliver reliable, controllable propulsion across a 50-foot garden row. Each rear wheel is powered by a CIM brushed DC motor connected to a 16:1 planetary gearbox, which increases torque output and reduces rotational speed to meet mobility requirements. Power is transferred through a chain and sprocket system, using 15-tooth sprockets on the motor shafts and 45-tooth sprockets on the wheel shafts to achieve a further 3:1 reduction. The sprockets were chosen as ANSI 25 (¼” pitch), with a compatible ANSI 25H Heavy Duty drive chain.

Pillow block bearings (UCP202-10 and UCP201-8) were mounted to the chassis to support the driven shafts and maintain drivetrain alignment. Chain tension was carefully adjusted during assembly by positioning the motor mounts and bearings to minimize slack and prevent chain derailment or excessive wear during operation. The entire drivetrain layout can be found in **Appendix 1a**, **1b**.

The chain drive configuration was selected for its cost-effectiveness compared to alternative direct-drive or belt-driven systems. Although periodic maintenance such as chain tension checks and lubrication will be necessary over time, the drivetrain reliably met performance targets during testing, providing a stable operating speed of approximately 3 mph with sufficient torque to traverse uneven garden terrain.

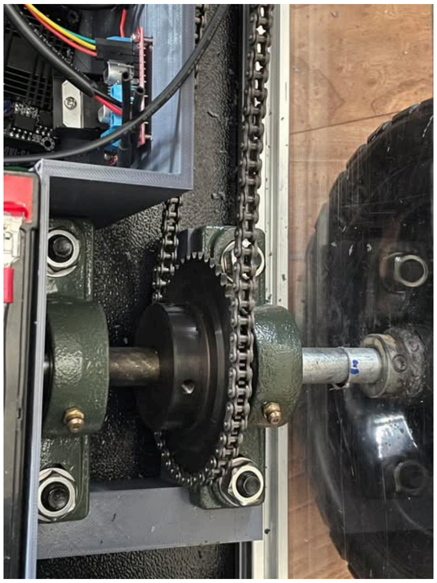


Figure 7: Bearing/sprocket layout for one wheel

From **Figure 7**, it can be observed that there is a 3D printed base for the bearing to sit on.

These were incorporated into the design to allow the shafts to clear the horizontal 8020 extrusions. An initial concern was the ABS plastic not being strong enough to withstand the clamping force of the M10 bolt; however, through a test prototype of the 3D printed platform, the team found that deformation was not an issue as long as the bolts were not tightened excessively.

2.2.4 Subsystems

Two modular subsystems were developed for the Gardening Robot: a watering system and a fertilizing system. Both subsystems were designed to mount securely onto the aluminum chassis using the integrated modular rail system, enabling flexibility, maintenance, and future upgrades.

The watering subsystem consists of a 10-gallon tank mounted at the center of the chassis, dual submersible pumps housed inside the tank, and adjustable outlet hoses routed through waterproof cable glands (**Appendix 1d**). The pumps are powered and controlled remotely through the Arduino-based electronic system, activated via the RC transmitter. Testing confirmed that the system delivers consistent water flow, and the waterproof routing ensures that leaks and electrical hazards are minimized during operation.

The fertilizing subsystem is based on an adapted off-the-shelf broadcast spreader. A gravity-fed hopper mounted onto the chassis stores granular fertilizer, which is dispensed through a 3D-printed platform (**Appendix 1c**) and driven by a 12V DC motor. A linear actuator controls the fertilizer chute, allowing the user to remotely start or stop fertilizer deployment during garden passes. Motor speeds and actuator positions are managed via the robot's Arduino control architecture.

Both subsystems themselves are mounted on a 36 inch long L-bracket, allowing for easy bolt-on. While the modularity concept was successfully achieved, subsystem swapping currently requires manual disconnection and reconnection of wiring, resulting in a longer-than-anticipated swap time of approximately 90 seconds. Future improvements could include spliced quick-connect systems to improve usability. Overall, both the watering and fertilizing subsystems performed reliably during testing and supported the robot's goal of reducing manual garden maintenance.

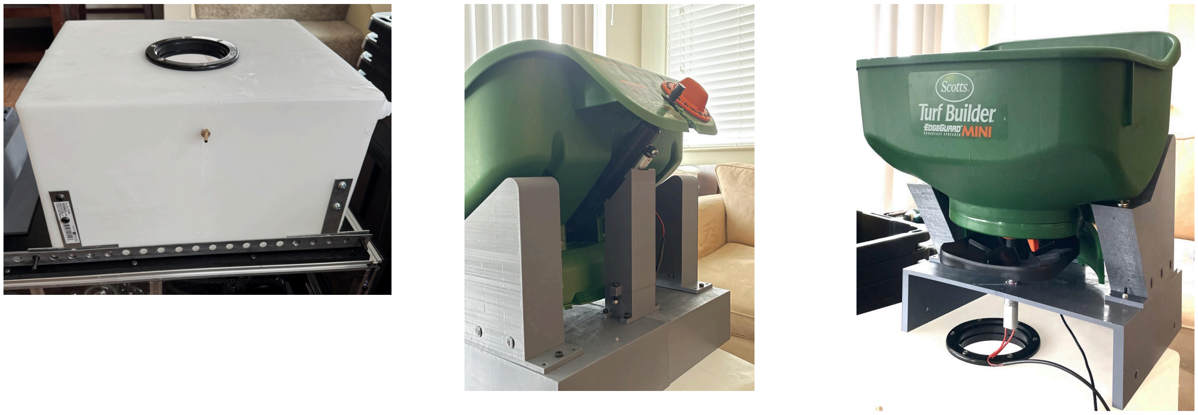


Figure 8: Water subsystem mounted on robot (left), fertilizer base attached to 3D printed platform with linear actuator attached (middle), fertilizer base with motor attached (right)

2.2.5 Electronics

The electronic system for the gardening robot was designed to distribute power, control signals, and subsystem actuation efficiently and safely across multiple components **Figure 9**. A

12V battery serves as the primary power source, supplying energy to both the drive motors and auxiliary subsystems. Positive and ground bus bars were incorporated to organize and manage high-current 12V distribution, reducing wiring complexity and ensuring reliable power delivery. Four L298N motor drivers are used to control two water pumps, a 12V DC motor, and a linear actuator, with each motor driver independently powered from the 12V bus. A 60A motor driver, powered from a separate 12V bus line, controls the main drive motors. A buck converter steps 12V down to 5V to supply regulated power to the Arduino microcontrollers and the FS-iA6B receiver module.

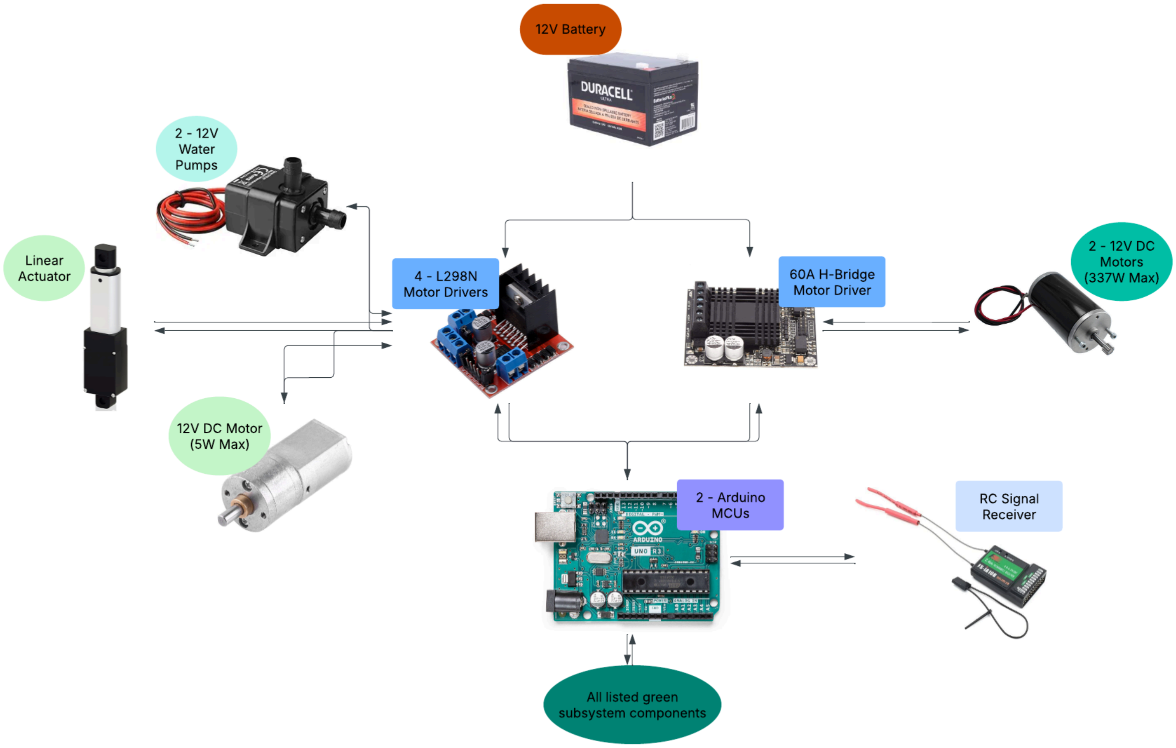


Figure 9: System-level block diagram outlining major electronic components

The FlySky FS-i6X remote controller serves as the user interface for real-time manual operation of the robot **Figure 10**. Each joystick and switch is mapped to a specific function: drive motor control, water pump toggle, fertilizer actuator, and fertilizer motor operation. For example, the left joystick is used to actuate the fertilizer hatch, while the right joystick controls robot motion. Switches are used for subsystem activation such as turning on and off the water pump and the motor attached to the dispersion disk. This layout was chosen to provide intuitive control while maintaining a compact and ergonomic design. The receiver decodes the user's inputs and transmits PWM signals to the Arduino, which then interprets and distributes commands to the appropriate actuators and motor drivers.

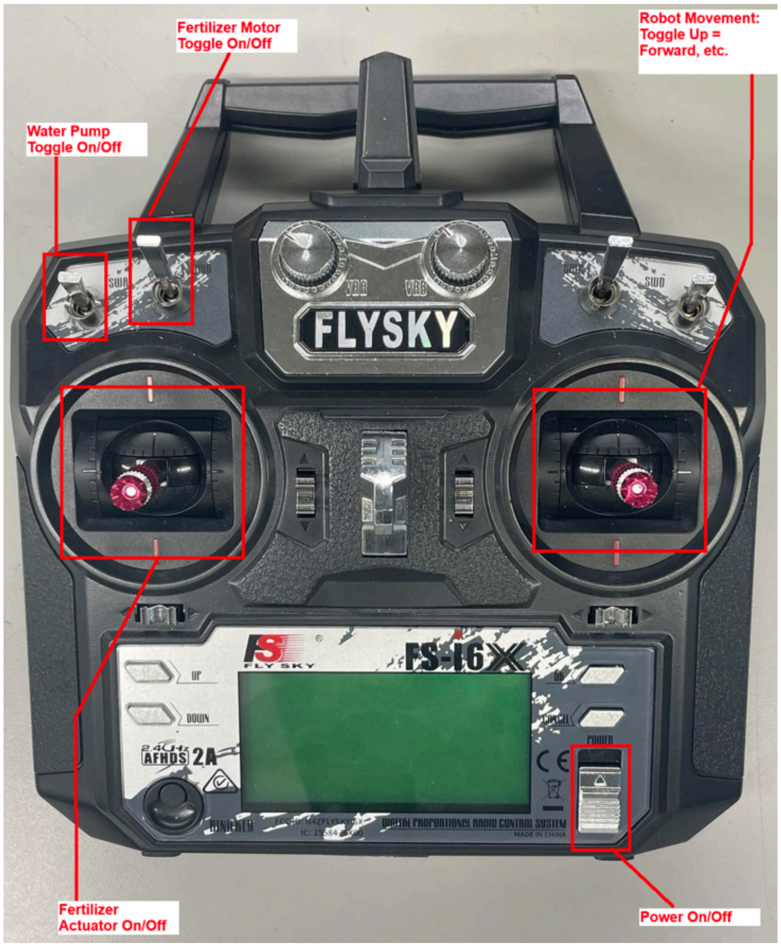

Fertilizer Motor Toggle On/Off
Robot Movement: Toggle Up = Forward, etc.
Water Pump Toggle On/Off
FLYSKY
FS-i6X
Fertilizer Actuator On/Off
Power On/Off

Figure 10: FlySky remote diagram, labeled with robot functions

Several design alternatives were considered for the electronic system. One option was to integrate all components directly through the Arduino's onboard voltage regulator; however, this approach posed significant risks of overcurrent and voltage instability. Another option involved using relay modules instead of motor drivers for simple on/off actuation. Ultimately, the system of dedicated motor drivers and distributed bus bars was selected to ensure safe current handling, support PWM control for variable speed actuation, and maintain organized and scalable wiring.

Selection criteria for evaluating the alternatives included system safety, wiring organization, scalability, and control flexibility. System safety was prioritized to prevent component damage due to high current draws. Wiring organization was considered to facilitate easier troubleshooting and maintenance. Scalability was important to allow for the addition of future subsystems without major redesign. Control flexibility was required to allow both digital on/off switching and analog PWM control of motors and actuators. Based on these criteria, the bus bar distribution combined with dedicated motor drivers emerged as the superior choice.

The final electronic system layout reflects a modular and robust design philosophy **Figure 11**. The master and slave Arduinos coordinate subsystem control, while each L298N motor driver manages an individual motor or actuator independently. High-current devices are connected directly to the 12V battery via bus bars to minimize voltage drops and power bottlenecks. The buck converter ensures that sensitive logic components operate at the correct voltage levels. This structured and redundant approach to power management and actuation control enhances system reliability, simplifies assembly, and provides flexibility for future expansions.

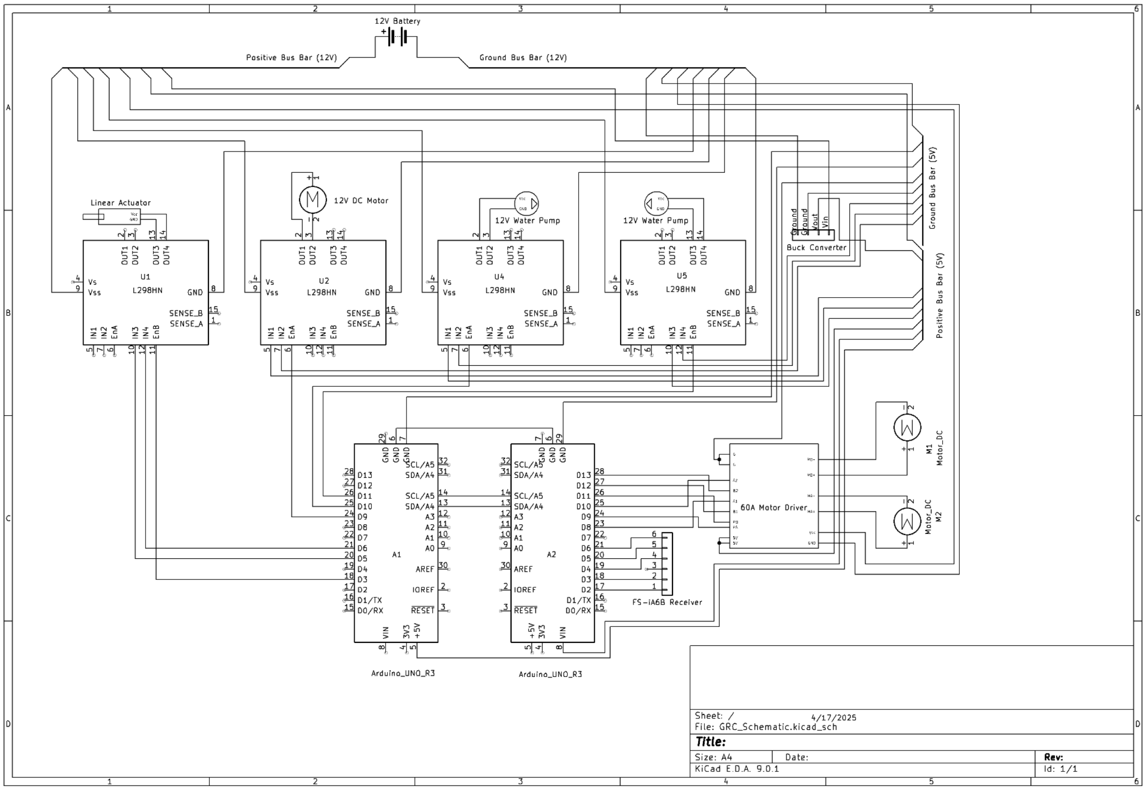


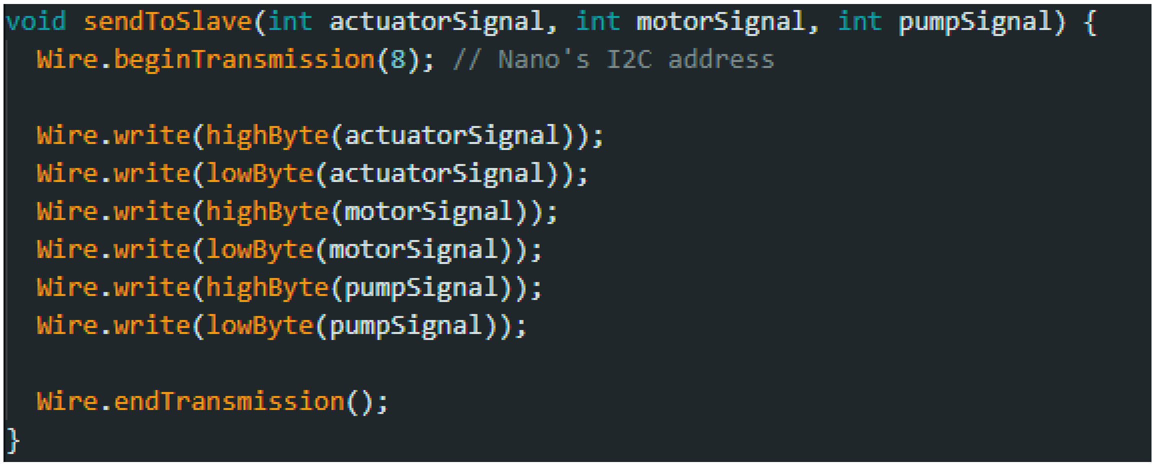
Figure 11: Detailed wiring schematic of the gardening robot's electronic system

2.2.6 Arduino Code

To control multiple motorized components within the gardening robot, a master-slave Arduino architecture was implemented. In this system, the master Arduino is responsible for receiving and interpreting remote control (RC) signals, including steering, throttle, and subsystem actuation commands. Based on these inputs, the master sends corresponding instructions to the slave Arduino through an I2C communication link. The slave Arduino then executes the commands by controlling the motors, pumps, and linear actuators using a variety of motor drivers selected based on the motor's power requirements. This architecture was selected to efficiently manage the limited PWM outputs available on a single Arduino board and to ensure reliable operation of all subsystems.

Several design alternatives were considered during development. One approach involved utilizing a single Arduino with a PCA9685 PWM expander to increase the number of available PWM channels. Another approach involved using a dual-Arduino setup, where control and execution responsibilities were divided between two microcontrollers. Selection criteria for evaluating these alternatives included system reliability, modularity, simplicity of debugging, and hardware availability. Reliability was prioritized to maintain consistent operation without PWM signal degradation. Testing comparing these alternatives showed that the master-slave Arduino configuration provided significantly higher scores across all categories. It offered superior reliability by reducing the risk of PWM interference, greater modularity by cleanly dividing system tasks, and enhanced ease of debugging by isolating control functions. Furthermore, using two Arduinos made debugging code significantly easier. As a result, the master-slave configuration was selected.

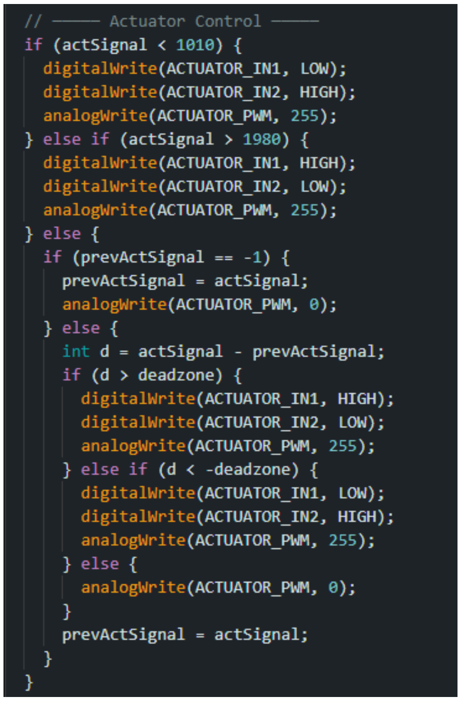
The master Arduino code (**Appendix 13**) is structured to continuously read pulse-width modulation (PWM) signals from the RC receiver, applying dead zones and thresholds to filter minor noise and unintended fluctuations. It interprets these inputs as specific commands for driving, subsystem actuation, or pump control. Commands are formatted into data packets and transmitted over the I^{2}C bus. The function used to send actuator, motor, and pump signals is shown in **Figure 12**, where each value is split into high and low bytes before being transmitted to the slave Arduino.



```
void sendToSlave(int actuatorSignal, int motorSignal, int pumpSignal) {
  Wire.beginTransmission(8); // Nano's I2C address

  Wire.write(highByte(actuatorSignal));
  Wire.write(lowByte(actuatorSignal));
  Wire.write(highByte(motorSignal));
  Wire.write(lowByte(motorSignal));
  Wire.write(highByte(pumpSignal));
  Wire.write(lowByte(pumpSignal));

  Wire.endTransmission();
}
```

Figure 12: Master to slave arduino code snippet

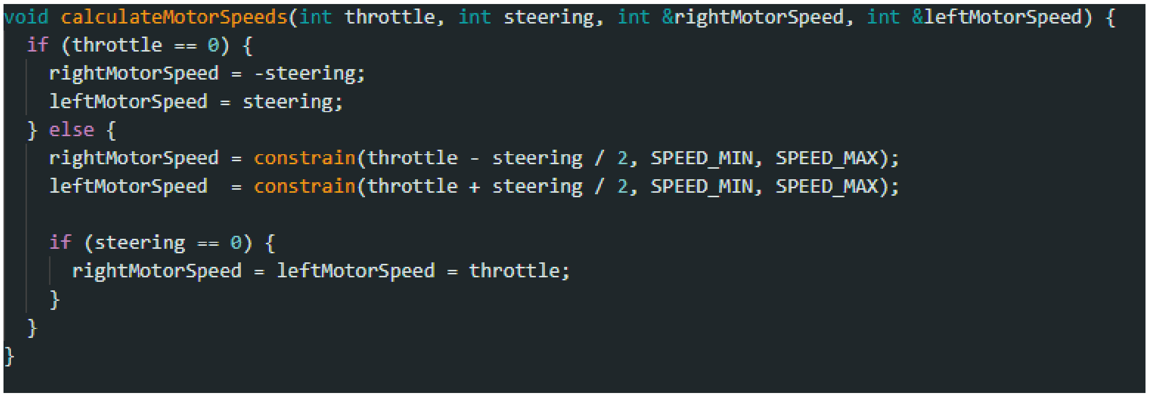
Upon receiving a command, the slave Arduino (**Appendix 14**) decodes the packets and executes appropriate actions. As shown in **Figure 13**, the slave processes signals to control the 12V motor, water pumps, and the linear actuator. Actuator control is designed to respond to direct user inputs while also implementing a dead zone to prevent minor fluctuations from causing unwanted movement.



```
// ───── Actuator Control ─────
if (actSignal < 1010) {
  digitalWrite(ACTUATOR_IN1, LOW);
  digitalWrite(ACTUATOR_IN2, HIGH);
  analogWrite(ACTUATOR_PWM, 255);
} else if (actSignal > 1980) {
  digitalWrite(ACTUATOR_IN1, HIGH);
  digitalWrite(ACTUATOR_IN2, LOW);
  analogWrite(ACTUATOR_PWM, 255);
} else {
  if (prevActSignal == -1) {
    prevActSignal = actSignal;
    analogWrite(ACTUATOR_PWM, 0);
  } else {
    int d = actSignal - prevActSignal;
    if (d > deadzone) {
      digitalWrite(ACTUATOR_IN1, HIGH);
      digitalWrite(ACTUATOR_IN2, LOW);
      analogWrite(ACTUATOR_PWM, 255);
    } else if (d < -deadzone) {
      digitalWrite(ACTUATOR_IN1, LOW);
      digitalWrite(ACTUATOR_IN2, HIGH);
      analogWrite(ACTUATOR_PWM, 255);
    } else {
      analogWrite(ACTUATOR_PWM, 0);
    }
    prevActSignal = actSignal;
  }
}
```

Figure 13: Linear actuator control arduino code snippet

The master Arduino also calculates the appropriate motor speeds for the left and right drive motors based on the throttle and steering input. This calculation, shown in **Figure 14**, adjusts the motor speeds to enable differential drive, ensuring smooth and responsive steering behavior.



```
void calculateMotorSpeeds(int throttle, int steering, int &rightMotorSpeed, int &leftMotorSpeed) {
  if (throttle == 0) {
    rightMotorSpeed = -steering;
    leftMotorSpeed = steering;
  } else {
    rightMotorSpeed = constrain(throttle - steering / 2, SPEED_MIN, SPEED_MAX);
    leftMotorSpeed  = constrain(throttle + steering / 2, SPEED_MIN, SPEED_MAX);

    if (steering == 0) {
      rightMotorSpeed = leftMotorSpeed = throttle;
    }
  }
}
```

Figure 14: Motor speed arduino code snippet

The slave code is further structured to prioritize responsiveness and safety, ensuring that motors and actuators return to a neutral or off state when communication is lost or invalid commands are detected. This structured division of roles between signal processing and actuator control results in a highly organized, modular, and reliable robotic system.

2.3 Critical Interfaces

The most critical interfaces are listed below in **Table 3**. It outlines the interface and the mechanical or electrical requirements at each interface.

Table 3: Critical Interfaces and Requirements

Interface	**Requirement/Criticality**
Wheel hub - ½” axle	Double set screw must resist all tractive torque to the wheels (roughly 3-4 lb-ft during average operation)
45-tooth sprocket - ½” axle	Set screw through sprocket hub must resist torque driven by chain (roughly 3-4 lb-ft during average operation)
Drive motor	Must have a torque of at least 1.56 lb-ft
Drive motor	Must sustain 12 V with at least 44.12 W of power
Drive motor gearbox	Must reduce RPM to 300 at motor shaft

Motor/gearbox platform (M10 bolts)	Must resist a torque of at least 0.39 lb-ft
Bearing/sprocket platform (M5 bolts)	Must resist a torque of at least 0.39 lb-ft
Wheel sprocket	Must reduce RPM to 100 at wheels
12V battery - positive/ground bus	12 AWG wires and 8 wire fastening screws must handle 12V/60A surge
Bus bars - 60A motor driver	12 AWG wire must sustain 12V/2x30A from motor driver channels
Bus bars - 4 x L298N motor drivers	18 AWG wire must sustain 12V/2x2A from each of the L298N motor drivers
5V rail - FS-iA6B RC Receiver/Arduino	Jumper wires must sustain 5V@100 mA/400 mA from the receiver and Arduino, respectively
Water Pump	Must have pressure of 2.36 kPa

3. Analysis for Design

3.1.1 Embodiment Design Checklist

The full embodiment design checklist can be found in **Appendix 3, 4**. The checklist is fully complete, with critical attributes addressed.

Customer needs were satisfied across the major performance target, including maneuverability, system cost, and robot power, with some deviation from the initial targets. These parameters are explored in more detail in **Section 3.4: Design Requirements Traceability Matrix**. The selected form solutions enable the robot to effectively dispense both fluids and granular solids as designed. The chassis is primarily configured as a rectangular prism, with modular rails mounted on top to allow for easy attachment and removal of subsystems. Due to the use of two high-torque motors for propulsion, heat generation and noise were anticipated during operation. To mitigate thermal buildup, the design minimized the surface area in direct contact with the motors, thereby reducing conductive heat transfer into the chassis. Additionally, the ⅝-inch plastic sheets enclosing the frame provide a degree of sound dampening, helping to reduce operational noise levels.

The final layout and material choices, primarily built upon 8020 extrusions and easily sourced components, provided a durable yet lightweight architecture, effectively meeting strength requirements. The 8020 extrusions allowed for ease of assembly and installation, while also providing a secure environment for torque transfer between the internal components. During assembly of the gear train, careful measurements were carried out to ensure sprocket alignment, resulting in a properly aligned and tensioned chain drive. The M4 bolts that hold the gearboxes in place to their aluminum plates were torqued down to ensure the entire system is mechanically sound and can resist the high torque generated by the motors. Also, the 3D printed platform on which the bearings sit on are able to support the compression from the M10 bolt connection.

The selected combination of motor, gearbox, and sprocket system enables efficient torque transmission to the drivetrain. The 16:1 gearbox increases the motor torque to approximately 28 lb-ft, significantly exceeding the required value, while reducing the output speed to around 331 RPM. An additional 3:1 chain-driven reduction further lowers the wheel speed to approximately 100 RPM, optimizing the system for controlled propulsion. The drivetrain assembly integrates sprockets, shafts, bearings, and chains to complete the mechanical power transmission. Motor actuation is controlled via an Arduino microcontroller interfaced with a motor driver specifically rated to handle the system's power demands.

As for safety, most of the robot's moving parts are enclosed within a chassis enclosure made of ⅝" thick ABS and PETG sheet, fastened to the 8020 extrusions. The subsystems are the only parts that the user would primarily be in contact with are the subsystems, which pose little risk even under abnormal operating conditions. The main safety concern is ensuring that the electronics suite is properly calibrated and connected to prevent any unwanted brownout, power spikes, or excessive discharge. Furthermore, the weight of the robot itself poses a safety risk, any small children, pets, or other small and fragile items must be removed from the robot's area of operation.

In terms of aesthetics, the robot was not designed with a certain theme or aesthetic in mind. The main aesthetic feature is the design choice of adding clear PETG sheets to the sides instead of black ABS sheets, so that the inner mechanisms of the robot can be seen. The user would be interacting with the RC FlySky Transceiver during most of the robot's normal operating activities.

The primary production and fabrication involves cutting the 8020 extrusions to their intended lengths, and drilling holes for fasteners along the chassis bottom deck. These require the use of a bandsaw and mills. The proper precautions and safety measures were taken to ensure safety standards were met while operating the equipment. Another component that required prototyping was the fertilizer base. It consisted of a variety of different 3D printed pieces, so some fasteners were needed to secure all of them together.

Quality control measures have been implemented throughout the chassis to ensure that future iterations and potential production-ready models maintain consistent tolerances and dimensions. Critical fasteners, particularly those associated with the powertrain, were secured using a torque wrench to verify proper preload and structural integrity. Additionally, all major components are standardized and sourced from readily available online suppliers, significantly reducing the need for custom in-house manufacturing.

The assembly process for the robot follows a structured top-down approach, beginning with the installation of the largest components, namely the chassis frame and the external chassis casing. Once the primary structural framework is secured, subassemblies such as motor mounts, bearing mounts, and subsystem attachment rails are installed to complete the drivetrain and modular support systems. Final assembly involves the integration of smaller components, including wiring harnesses and other electronic components.

Although recyclability was not a primary focus during the initial design phase, the robot's construction allows for relatively straightforward disassembly. The chassis, composed largely of 80/20 aluminum extrusions and modular fasteners, can be easily repurposed for future projects. Furthermore, the ABS plastic used in the chassis casing is recyclable, and the majority of components were sourced as standardized parts, minimizing waste associated with custom manufacturing.

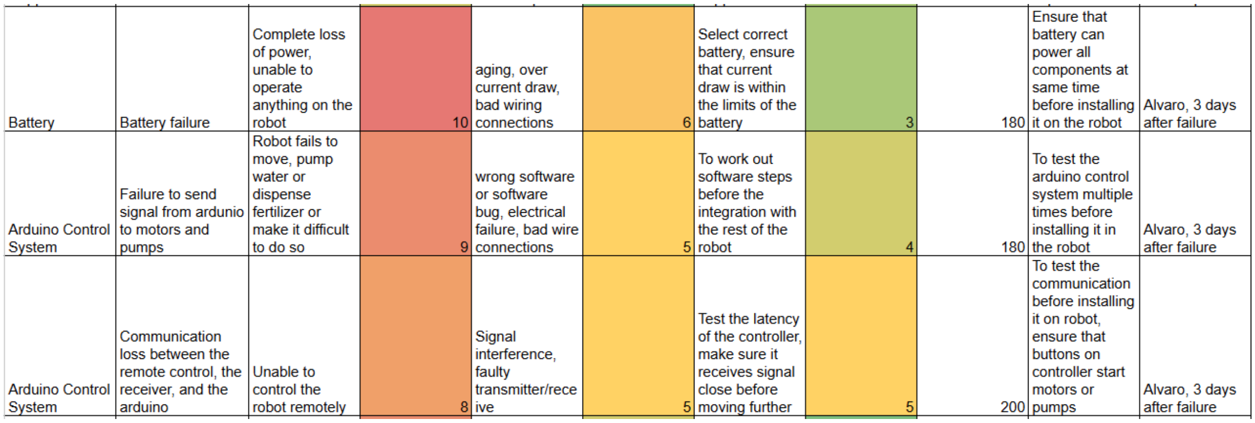
The modular design of the robot significantly simplifies maintenance procedures, allowing for quick access to internal components if malfunction occurs. If an issue arises during operation, the user can easily detach the modular subsystems and remove the upper chassis casing without the need for specialized tools. This would then expose the internal drivetrain, wiring, and electronic assemblies, enabling straightforward repair. The accessibility of the robot's internal structure reduces downtime.

The cost limits were also observed, outlined in **Section 3.5 Cost Accounting and Cost Model**. The team was careful during the assembly process to not overspend on critical components, ensuring the allocated budget was not exceeded. The total cost for all components procured was $2,301.97, which is within the 30% safety margin granted at the end of MEEN 401.

Finally, the lead times and delivery schedules for components were relatively manageable. Most suppliers, whether that be Amazon or McMaster-Carr, had a maximum lead time of roughly a week. Since most were common, off-the-shelf components, waiting on delivery was generally not an issue. It is worth noting that while submitting prototyping requests for parts such as the fertilizer base, the lead times increased significantly as the semester progressed, due to a higher volume of total requests from other Capstone teams.

3.1.2 FMECA

A comprehensive Failure Modes, Effects, and Criticality Analysis (FMECA) was conducted to systematically identify potential failure points within the robot's major systems, evaluate their severity, occurrence likelihood, and detection difficulty, and propose mitigation strategies. The analysis covered the drivetrain, water and fertilizer subsystems, electronics, and chassis structure. Key takeaways from the FMECA include the identifying high-priority risks such as drivetrain misalignment, battery failure, and pump wiring issues, which exhibited high Risk Priority Numbers (RPNs) and thus needed immediate design attention.



Battery	Battery failure	Complete loss of power, unable to operate anything on the robot	10	aging, over current draw, bad wiring connections	6	Select correct battery, ensure that current draw is within the limits of the battery	3	180	Ensure that battery can power all components at same time before installing it on the robot	Alvaro, 3 days after failure
Arduino Control System	Failure to send signal from ardunio to motors and pumps	Robot fails to move, pump water or dispense fertilizer or make it difficult to do so	9	wrong software or software bug, electrical failure, bad wire connections	5	To work out software steps before the integration with the rest of the robot	4	180	To test the arduino control system multiple times before installing it in the robot	Alvaro, 3 days after failure
Arduino Control System	Communication loss between the remote control, the receiver, and the arduino	Unable to control the robot remotely	8	Signal interference, faulty transmitter/receive	5	Test the latency of the controller, make sure it receives signal close before moving further	5	200	To test the communication before installing it on robot, ensure that buttons on controller start motors or pumps	Alvaro, 3 days after failure

Figure 15: Examples of failure modes with high RPNs

As observed in **Figure 15**, high-priority risks were most pronounced with the electronics, specifically the battery and Arduino control system. The Arduino code malfunctioned multiple times during the final assembly and testing phases, so iterative programming was conducted to ensure the code was compatible with how the wires were connected between the board and the motor drivers, among other components. For example, a common failure was "bad wire connections". Connecting wires entailed using a precision screwdriver to ensure that the proper wires were secured within the terminals of the motor drivers. Sometimes, the physical connection would come loose, effectively discontinuing the circuit. Similar to the iterative programming done with the Arduino code, various testing phases were done to ensure solid physical connections between all of the electronic components. The full FMECA matrix can be found in **Appendix 5, 6**.

3.1.3 Fault Tree Analysis

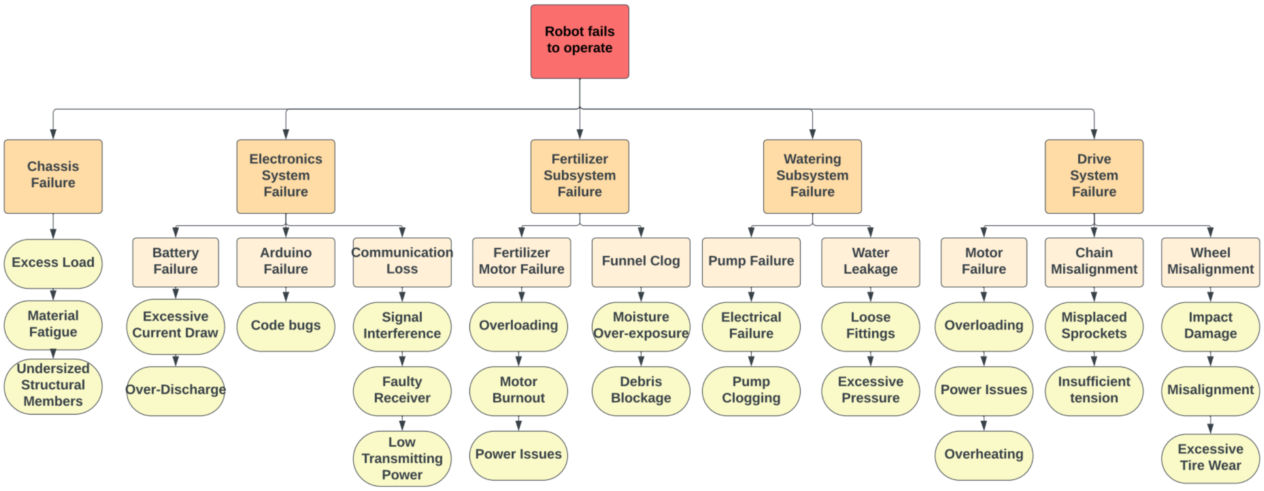


Figure 16: FTA of Garden Robot outlining critical risk relationships

A Fault Tree Analysis (FTA) was conducted to systematically identify potential failure modes and the cause-and-effect relationships between them. The analysis categorized failures into five primary categories: chassis failure, electronics system failure, fertilizer subsystem failure, watering subsystem failure, and drive system failure. For each failure path, design decisions and mitigation strategies were used to decrease the chance and severity of these states.

For chassis failure, excess load resulting in material fatigue or undersized structural members was identified as a critical risk. To mitigate this, the chassis was built from 8020 aluminum extrusions. This material was selected for its relatively high strength-to-weight ratio. Bending stress hand calculations verified for adequate safety factor against expected loads. Conservative design margins were also built into the chassis to account for unanticipated dynamic loading during normal operation.

Electronics system failure, encompassing battery failure, Arduino failure, and communication loss, was another significant risk area. To address battery failure due to excessive current draw and discharge, proper batteries (Duracell SLA) and motor drivers were selected. Arduino failure from code bugs was mitigated through incremental code development and debugging sessions after testing. Communication loss from signal interference or faulty receivers was addressed by ensuring the 2.4 GHz receiver link was properly secured to the Arduino before testing.

Fertilizer subsystem failure, such as motor burnout from overloading or funnel clogs due to moisture and debris, was considered during the design of the fertilizer base mount. An off-the-shelf fertilizer spreader was chosen, since the team would be able to forego any testing for debris blockage. A motor for the distributor disk was selected with a sufficient torque rating and duty cycles to prevent overloading, and the distribution system itself was designed with easy-access bolt-on points to allow for easier maintenance and debris clearance.

In the watering subsystem, pump failure and water leakage were identified as potential failure modes. Pumps were selected with ratings appropriate for expected duty cycles and flow pressure. Water leakage risks due to loose fasteners were mitigated by adding a silicone sealant along all cable glands and drilled openings. Redundancy in the cable glands also reduced the risk of leakage. Multiple trials for water leakage were conducted before the team was satisfied with the sealing integrity of the tank.

Finally, drive system failures, including motor failure, chain misalignment, and wheel misalignment, were considered. Motors were selected with enough power margin to prevent overloading, and they are connected to 60A H-bridge motor drivers to ensure their current and power draw is effectively distributed. Chain alignment was carefully managed during drivetrain assembly, with sprocket placement considered and tightened to their respective drive shafts before finally attaching the chains. Wheel misalignment and impact damage risks were addressed by selecting large, off-road wheels and giving them enough clearance from the chassis enclosure to operate without obstruction.

Ultimately, the FTA proved critical in guiding the team in selecting components and assembly procedures that would mitigate the most risk. Of course, not all risks could be eliminated in their entirety, but this process of identifying the most critical ones significantly reduced the probability of operational failure.

3.1.4 Standards and Codes

The first and arguably most crucial standard the team considered was the Failure Modes and Effects Analysis: SAE Standard J1739 (2021) [9]. This FMEA standard outlines processes for identifying and mitigating risk through various tools, such as rating charts and worksheets. Thus, the FMEA documentation for the project complied with this standard.

Other structural standards the team referenced were ANSI B18.2.1: Standard for Bolts and Fasteners [11] and ANSI MH28.2: Structural framing standards [10]. These standards guided the design and correct use of various fasteners used in the assembly process, such as M5 bolts for the 8020 extrusions and M10 bolts for the bearings. ANSI MH28.2 was referenced for design and loading assumptions for the aluminum chassis and frame connections. General machinery requirements were referenced from ISO 10218-1: 2025: Industrial robots safety [12] and OSHA 1910 Subpart O: Machinery and Machine Guarding [13]. These standards guided accepted practices involving rotating parts, pinch points, and moving systems, such as keeping the entire drivetrain system enclosed within the chassis.

For the electrical and control systems, design validation criteria were informed by NFPA 70: National Electric Code [15] for wiring safety and proper wiring color-coding. IEC 62133-2: Li-ion cell and battery safety [14] was referenced for battery selection; the chosen Duracell SLA battery is certified and is connected to bus bars within the robot to control proper power flow. Finally, FCC Part 15 Subpart C: Unlicensed intentional radiators was referenced to guide selection of the RC transmitter and receiver system; the chosen FlySky FS-i6A 2.4 GHz link does not exceed power limits, disrupt wireless signals, and is within regulated frequencies.

3.2 DFMA Considerations

Design for Manufacturing and Assembly principles were strongly considered throughout the design and assembly process to ensure fabrication efficiency and serviceability of the final prototype. These principles were especially important for this project due to the relatively high number of distinct components. The chassis and primary subsystem frames were built using standardized 8020 aluminum extrusions and off-the-shelf corner brackets, which allowed for easy cutting and bolting without the need for specialized fabrication equipment. Many parts were built off of the extrusions themselves, such as the plastic sheets enclosing the chassis. This material selection reduced manufacturing costs and allowed for greater flexibility for modifications. Fastening methods were deliberately kept simple, relying predominantly on common M5 bolts and T-slot channel nuts to minimize the number of different tools and fasteners required during assembly. Most parts that were fastened to the extrusions only required an allen wrench. In subsystem design, parts were standardized where possible to reduce the overall part count, although late-stage design changes led to an increase in unique components beyond the original plan. For example, the team chose to use an off-the-shelf Scotts TurfBuilder EdgeGuard Mini [8] to reduce the amount of 3D printed material. Although this reduced the need for extensive prototyping, the team encountered increased complexity when modeling the base and various other mounts to be fastened to the spreader.

The team also considered reducing fastener count when modeling parts for 3D print. Some parts were too big to fit in a singular print job, so the part had to be separated into multiple pieces. Instead of modeling holes for fasteners to put these pieces back together, various clamping and snap-fit features were added instead, as shown in **Figure 17**.

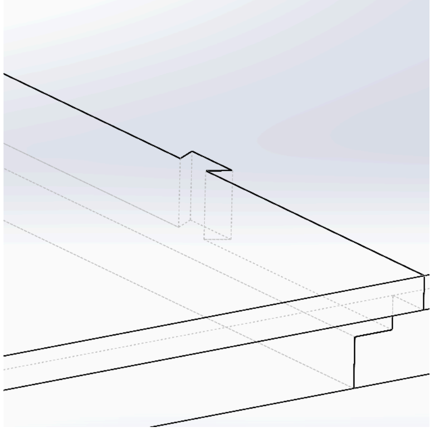


Figure 17: Dovetail features incorporated into chassis sheet interface

Other DFM principles that were incorporated into the design included allowing for enough space between fasteners, so that there is ample room for a fastening tool. Furthermore, slots and channels were machined into fastened components, accounting for tolerance. Lastly, major structural components like the chassis frame were assembled first, representing the order of assembly in which the most reliable goes first, and the least reliable last. The team followed a roughly top-down approach when assembling, where the main frame of the robot was assembled, followed by drivetrain and power implementation, and finally the least reliable subfunctions: the electronics and wiring suite.

3.3 Design Validation

The design of the Gardening Robot was validated through a combination of analytical modeling, risk assessment, and physical testing. Early-stage validation included hand calculations and material selection analysis to ensure the chassis could support anticipated static and dynamic loads without exceeding allowable stress limits. Additional validation was performed through Failure Mode, Effects, and Criticality Analysis (FMECA) and Fault Tree Analysis (FTA), which identified drivetrain misalignment, battery failure, and wiring connection issues as critical risks. Mitigation strategies such as improved alignment procedures, iterative code debugging, and physical wire connection checks were implemented to address these concerns.

A 2^k factorial DoE was conducted using appropriate independent variables based on what the team deemed most viable given the validation window and available resources. Assumptions of the experiment include: stable and roughly constant environmental conditions across each trial (ambient temperature T ≈ 85°F, 76% humidity), robot configuration (chain tension, pump and fertilizer Arduino code, etc.) is constant across each trial, measurement systems are repeatable and unbiased, and task success is treated as a binary result instead of a statistical response. Four independent variables were identified, along with their high and low values:

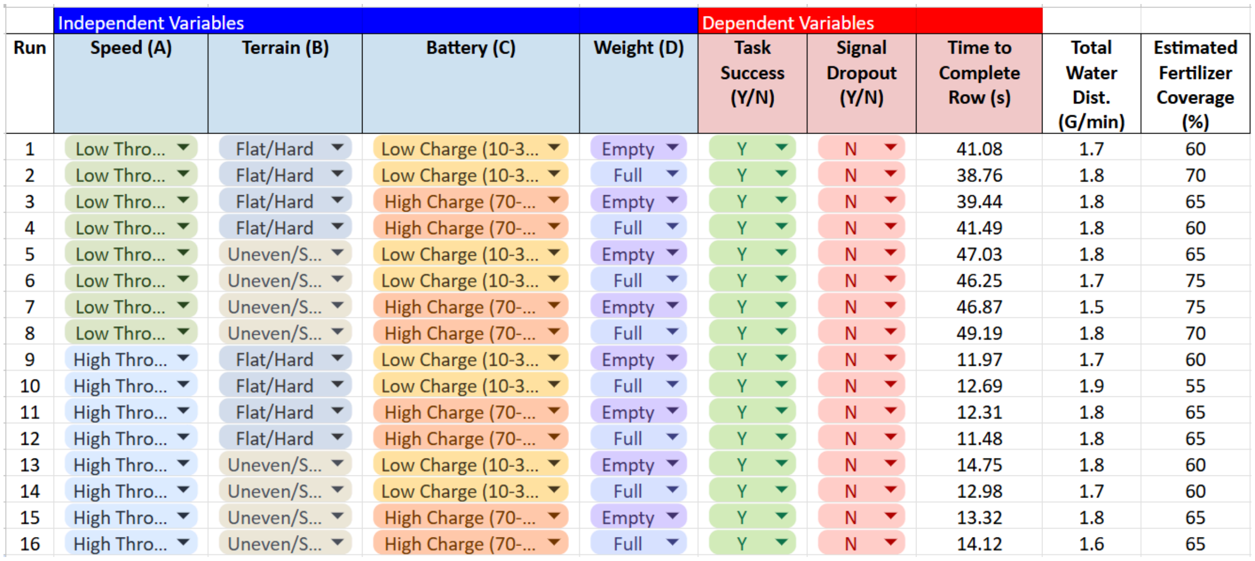
- Speed: Low Throttle vs High Throttle
- Terrain: Flat/Hard vs Uneven/Soft
- Battery Charge: Low Charge (10-30%) vs High Charge (70-100%)
- Weight: Empty Subsystems vs Full Subsystems

Since k = 4 independent variables, 16 total trials were conducted. Varying speed involved pushing the joystick slightly on the RC for low throttle and pushing it all the way for full throttle. Flat/hard terrain involved testing the robot's movement on concrete, while uneven/soft terrain involved testing the robot on soft dirt, with patches of grass. Three dependent variables were also identified based on the independent variables:

- Task Success: Y/N
- Signal Dropout: Y/N
- Time to Complete Row

The last two columns, as seen in **Table 4**, correspond to subsystem validation. Since there were no independent variables that can easily vary the flow rate and distribution of consumables, the subsystems were simply tested in conjunction with the DoE for a total of 16 trials. The total water distribution was measured by noting how much water left the tank, taking the difference between water level at the beginning of the trial vs at the end of the trial. The fertilizer coverage has a significantly more varied distribution than the fluid subsystem, so its data was mostly qualitative based on how much fertilizer had effectively covered the testing area (50 ft x 2 ft).

Table 4: Design of Experiments Validation Table



	Independent Variables				Dependent Variables				
Run	Speed (A)	Terrain (B)	Battery (C)	Weight (D)	Task Success (Y/N)	Signal Dropout (Y/N)	Time to Complete Row (s)	Total Water Dist. (G/min)	Estimated Fertilizer Coverage (%)
1	Low Thro...	Flat/Hard	Low Charge (10-3...	Empty	Y	N	41.08	1.7	60
2	Low Thro...	Flat/Hard	Low Charge (10-3...	Full	Y	N	38.76	1.8	70
3	Low Thro...	Flat/Hard	High Charge (70-...	Empty	Y	N	39.44	1.8	65
4	Low Thro...	Flat/Hard	High Charge (70-...	Full	Y	N	41.49	1.8	60
5	Low Thro...	Uneven/S...	Low Charge (10-3...	Empty	Y	N	47.03	1.8	65
6	Low Thro...	Uneven/S...	Low Charge (10-3...	Full	Y	N	46.25	1.7	75
7	Low Thro...	Uneven/S...	High Charge (70-...	Empty	Y	N	46.87	1.5	75
8	Low Thro...	Uneven/S...	High Charge (70-...	Full	Y	N	49.19	1.8	70
9	High Thro...	Flat/Hard	Low Charge (10-3...	Empty	Y	N	11.97	1.7	60
10	High Thro...	Flat/Hard	Low Charge (10-3...	Full	Y	N	12.69	1.9	55
11	High Thro...	Flat/Hard	High Charge (70-...	Empty	Y	N	12.31	1.8	65
12	High Thro...	Flat/Hard	High Charge (70-...	Full	Y	N	11.48	1.8	65
13	High Thro...	Uneven/S...	Low Charge (10-3...	Empty	Y	N	14.75	1.8	60
14	High Thro...	Uneven/S...	Low Charge (10-3...	Full	Y	N	12.98	1.7	60
15	High Thro...	Uneven/S...	High Charge (70-...	Empty	Y	N	13.32	1.8	65
16	High Thro...	Uneven/S...	High Charge (70-...	Full	Y	N	14.12	1.6	65

The first takeaway from the validation process was that the location of the throttle joystick was the dominant factor for time efficiency, as expected. Varying the joystick position will vary the motor control, as outlined in **Figure 14**. At low throttle, the time to complete the row is roughly between 39 and 49 seconds, while at high throttle, the time drops to roughly

11-14 seconds. Another notable characteristic of the validation data is that uneven terrain increases the time more significantly for lower throttle runs. Standard deviation for lower throttle runs $s = 4.001$ while standard deviation for higher throttle runs $s = 1.092$. Under typical garden conditions, the robot will be at low throttle on uneven/soft surfaces; therefore, speed will vary during operation. Another key takeaway is that the fertilizer coverage is lower at high throttle, where the average is 61.875%, compared to coverage at lower throttle, averaging 67.5%.

It can also be observed that task success is 100% across all trials, and signal dropouts are always No. The robot can handle its loads across the 50 ft testing row with no complications, indicating a reliable system under typical operating conditions. Furthermore, the RC system does not drop out within the 50 ft testing area. Additional testing has not been performed to test the control system's maximum allowable range; however, as the FlySky receiver and transmitter were originally designed for hobby airplanes, the team foresees no issues with signal.

Surprisingly, the battery charge and robot weight have very limited effect on robot speed and other critical performance metrics. As long as the battery charge is above roughly 10%, there is no observable drop in battery performance. Furthermore, the team had expected the speed of the robot to decrease with heavier loads, but there was no noticeable change when full and empty weight was varied. This is likely due to the dual motor setup’s high torque rating, along with the 3:1 sprocket reduction.

Future work in the validation of the robot includes adding more trials for each run to ensure replication and statistical confidence. On top of the existing trials, the team recommends constructing a dummy garden to test the robot; this will make validation of agronomic benefits more observable. Moreover, it would be beneficial to validate operations under varied real world conditions. For example, having a soil moisture sweep, ambient temperature sweep, or crosswind sweep can be beneficial in determining the robot’s performance in suboptimal environmental conditions.

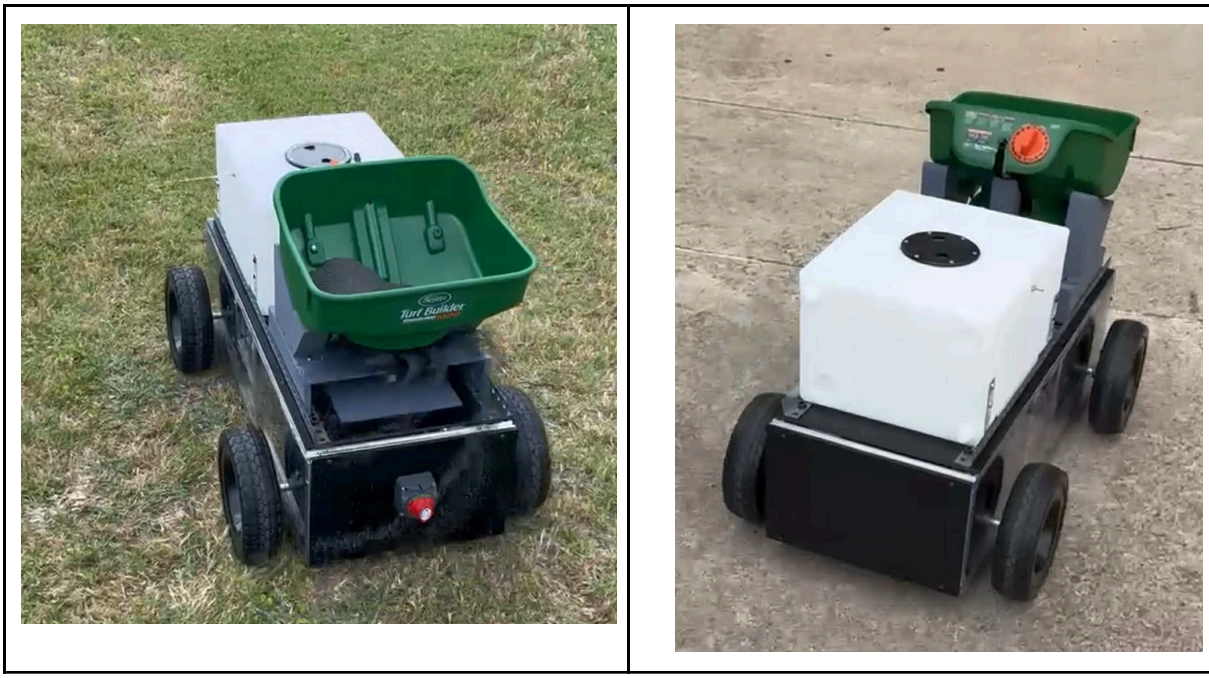


Figure 18: Robot testing on uneven terrain; note fertilizer distribution in spiral pattern and water spray (left), robot testing on hard terrain (right)

3.4 Design Requirements Traceability Matrix

Table 5: Requirements Traceability Matrix

Requirement Metric	**Initial Target Value**	**Actual Value**	**% Improvement/ % Decline**
Cost	$2500	$2122.90	15%
Power	150 W	674 W	349%
Mass	70 lbs	84 lbs	20%
CG	0.6 ft	0.53 ft	11.67%
Consumable Volume	0.03 m^{3}	0.038 m^{3}	26.67%
Complexity	15 unique parts	33 unique parts	120%
Efficiency	10 s per plant	Variable	N/A
Modularity	30 s	90 s	200%
Maneuverability (Turning Radius)	0.9 m	0.01 m	89%

To guide the development and validation of the project, the team established a requirements traceability matrix to compare with the customer requirements early on in the design phase. Several predictive models were developed to set initial values and identify potential design trade-offs.

A power and drivetrain model was constructed based on estimates of robot mass, terrain friction, and desired operating speeds. By conducting power and torque calculations, the team was able to identify that at least 150 W of total power was needed to drive the robot. After identification of the suitable motors (in this case, an AndyMark Brushed DC motor), the robot was able to produce 674 W of power, since each motor had a power rating of 337 W [7]. This represented a 349% increase from target power values. While beneficial in outputting higher power, this also exacerbates power draw from the battery, decreasing its battery life.

Mass estimations were made using preliminary CAD models assuming lightweight 8020 aluminum extrusions and minimal subsystem mass, targeting a dry mass of 70 pounds. The final assembled mass of the robot without consumables was 84 pounds, a 20% increase, reflecting the addition of structural reinforcements and heavier drivetrain components. A center of gravity estimation was used to minimize the risk of rollover and allow for stability design, initially targeting 0.6 ft from the ground. Due to heavier than anticipated drivetrain components near the bottom of the robot, the actual CG was 0.53 ft (pulled from SolidWorks), an 11.67% improvement.

The consumable volume model, which accounted for the capacities of the water tank and fertilizer hopper, targeted 0.03 m³. The final value achieved was 0.038 m³, a 26.67% improvement, which allows for longer operation before requiring refills. Complexity was estimated based on early subsystem sketches, with a goal of limiting the system to 15 unique parts. The final design contained 33 unique parts, a 120% increase, primarily due to an underestimation of how many different components were required for the drive system and electronics suite. This value does not reflect the various fasteners and wires.

The target for attaching and detaching subsystems was 30 seconds on average, reflecting a modular architecture. However, final testing showed swap times of around 90 seconds, due to the increased complexity of attaching wires to the motor driver within the electronics housing. Lastly, a maneuverability estimate was developed, targeting a turning radius of 0.9 meters. This estimate assumed the implementation of a steering rack, which was ultimately not incorporated in the final design. Instead, a tank-style steering system was developed, where wheels on each side counter-spin.

Overall, the predictive models provided valuable guidance in establishing design expectations. While some metrics, such as power consumption and complexity deviated from initial targets, others, such as stability, consumable capacity, and maneuverability, exceeded expectations. The primary causes of deviation were unanticipated factors during the assembly process. Some parts were not accounted for in the initial design phase, and thus parameters like their additional mass and complexity were not considered from the start. Despite these challenges, the customer requirements matrix proved pivotal in supporting the design process.

3.5 Cost Accounting and Cost Model

During MEEN 401, the team developed a preliminary budget based on the estimated parts necessary to construct the gardening robot. A comprehensive list of materials was created in accordance with the robot's design requirements. The estimated project expenditure was $1,900, and with a 30% contingency margin, a final budget request of $2,500 was submitted for approval.

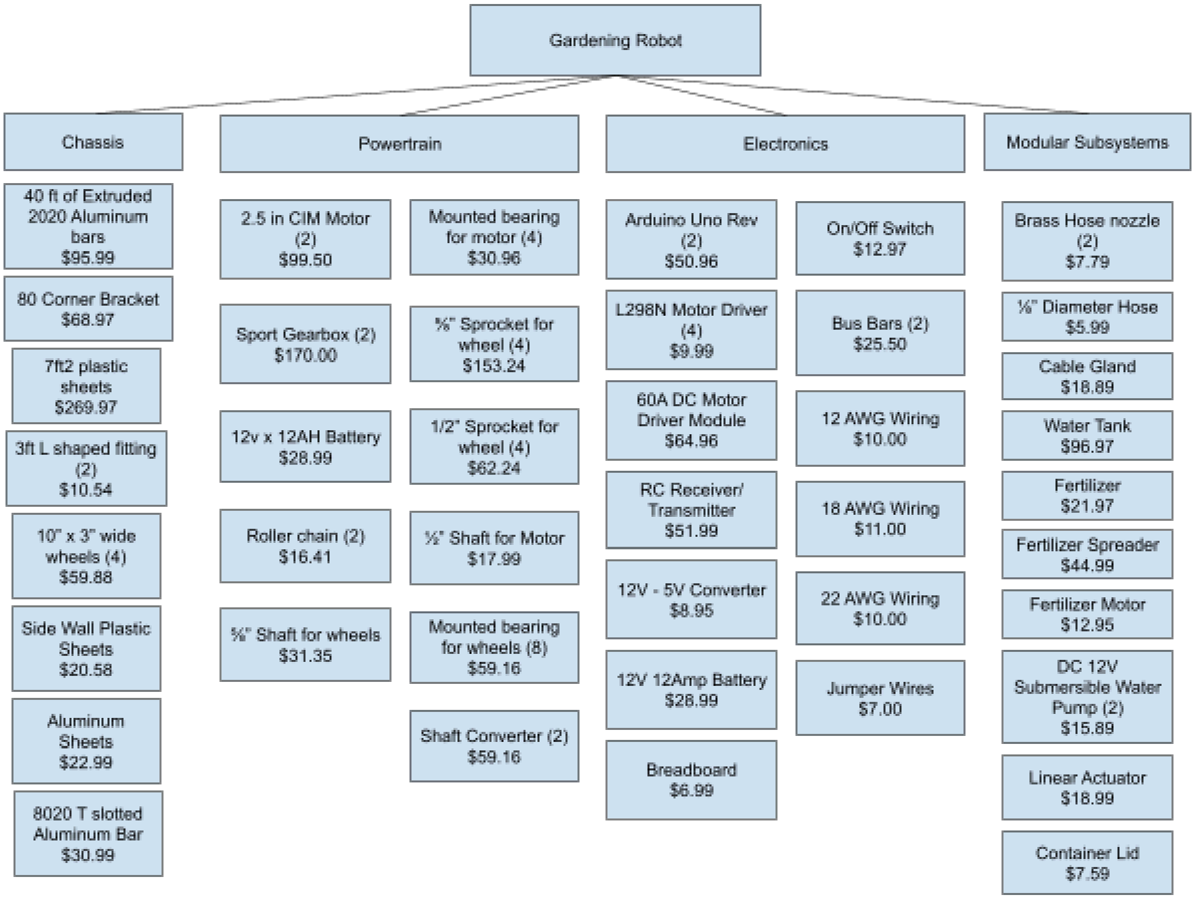

Gardening Robot
Chassis
Powertrain
Electronics
Modular Subsystems
40 ft of Extruded 2020 Aluminum bars $95.99
80 Corner Bracket $68.97
7ft2 plastic sheets $269.97
3ft L shaped fitting (2) $10.54
10" x 3" wide wheels (4) $59.88
Side Wall Plastic Sheets $20.58
Aluminum Sheets $22.99
8020 T slotted Aluminum Bar $30.99
2.5 in CIM Motor (2) $99.50
Sport Gearbox (2) $170.00
12v x 12AH Battery $28.99
Roller chain (2) $16.41
⅝" Shaft for wheels $31.35
Mounted bearing for motor (4) $30.96
⅝" Sprocket for wheel (4) $153.24
1/2" Sprocket for wheel (4) $62.24
½" Shaft for Motor $17.99
Mounted bearing for wheels (8) $59.16
Shaft Converter (2) $59.16
Arduino Uno Rev (2) $50.96
L298N Motor Driver (4) $9.99
60A DC Motor Driver Module $64.96
RC Receiver/ Transmitter $51.99
12V - 5V Converter $8.95
12V 12Amp Battery $28.99
Breadboard $6.99
On/Off Switch $12.97
Bus Bars (2) $25.50
12 AWG Wiring $10.00
18 AWG Wiring $11.00
22 AWG Wiring $10.00
Jumper Wires $7.00
Brass Hose nozzle (2) $7.79
⅛" Diameter Hose $5.99
Cable Gland $18.89
Water Tank $96.97
Fertilizer $21.97
Fertilizer Spreader $44.99
Fertilizer Motor $12.95
DC 12V Submersible Water Pump (2) $15.89
Linear Actuator $18.99
Container Lid $7.59

Figure 19: Cost model of final prototype

The cost model outlines all essential components required for the robot's construction. For clarity, the components are categorized into four groups: chassis, encompassing the structure and frame; powertrain, comprising the mechanical systems that drive the robot; electronics, consisting of the actuation and control systems; and modular subsystems, responsible for distributing materials onto the garden. Most standard components were sourced from Amazon to take advantage of expedited shipping and cost efficiency. However, specialized, high-performance parts, such as motors and sprockets, were procured from vendors like AndyMark and McMaster-Carr. The cost to reproduce the gardening robot, based purely on parts and 3D-printed components, is $1,963.22. This estimate does not account for manufacturing tools used or labor expended during assembly.

The final project expenditure totals $2,301.97, encompassing all parts purchased and 3D-printed through the FEDC. Although this amount exceeds the original $1,900 estimate, it remains within the approved 30% contingency margin. A detailed review of the bill of materials reveals that several components were either unused or not fully integrated into the final design. For instance, certain electronic components, such as MOSFET controllers, were initially anticipated but ultimately proved unnecessary. Similarly, a GPS tracking system was considered but not implemented due to time constraints. In addition to material costs, several tools were purchased to supplement the fabrication shop's capabilities. Notably, a chain breaker was acquired to properly size and install chains onto the sprockets, a critical step in assembling the drivetrain.

4. Broader Impacts of Design

4.1 Design for Lifecycle

When designing the garden robot, the life cycle of the robot was factored in the material selection, maintenance and modularity of the robot. The first decision that was considered for the life cycle is the maintenance of the robot. The robot was designed with modularity in design making it easy to work on or replace parts. Most parts were also off the shelf components meaning that it should be easier finding replacement parts.

Another key lifecycle consideration was the robot's repurposability. Its modular design makes it easy to adapt for other uses beyond gardening. If the two subsystems are removed, anything under 100 pounds can be mounted on top, whether for carrying materials or supporting other tools. The watering and fertilizing subsystems are also versatile and could be used in different applications. Core components like the pumps, Arduino boards, and motors can be reused in future projects.

Recyclability was another decision factor that went into the design. The frame was made out of aluminum with stainless steel corner brackets which are recyclable. Also the ABS plastic siding of the robot is also recyclable making it the reason why it was chosen over other plastics. The components of the robot that are not recyclable like the battery and some of the electronic components were kept to a minimum.

The environmental impact of the design is low as the manufacturing process only used cutting, drilling and minimal welding. When the robot is running it also does not produce any emissions as it is electronically powered. While this design took life cycle decisions into consideration, future designs could improve on these decisions to better improve the life cycle as well as environmental impact.

4.2 Intellectual Property

The garden robot design does not have anything that would be deemed intellectual property. Intellectual property generally includes patents or new technology that are made in the design process. This project design is based on well known engineering designs and uses off the shelf components like pumps, motors and a remote control system. None of the components are a new concept meaning that there is no intellectual property related to the project.

4.3 Liability Considerations

Designing a gardening robot has many potential liability considerations because it reacts with not only the user but also the outdoor environment. To reduce these risks the design was carefully thought out for the prototype. One of the primary concerns for liability was the electrical system. The robot uses a 12 volt battery and even though 12 volt is deemed low voltage, it still can pose a safety concern when it is exposed to water or not handled properly.

Mechanical risk is another liability concern. Moving components like the motors, chains and fertilizer spreader disk are also a concern due to it being a potential pinch point or hits operator when performing maintenance. To reduce these liability concerns, a kill switch was installed on the robot. When the kill switch is off, the electronics are disconnected from the battery and no electronics will have power provided to them and all moving parts will come to a stop.

To ensure safe operation of the gardening robot, the user **must** power on the remote control before turning on the robot itself. This sequence is critical because if the robot is turned on first, it may receive undefined or residual PWM signals from the receiver, causing the motors or actuators to engage unintentionally. Such unexpected behavior can pose safety hazards or damage the components. Activating the remote control first establishes a stable communication link with valid signal outputs, ensuring that the robot initializes in a controlled and idle state.

Other mitigation steps taken include ensuring that all wires meet current loads to prevent overheating or wire failure. The electronic housing has a lid to increase the waterproofing capabilities of the electronic container, reducing the chance of a short circuit. Finally a part of the fertilizer spinning disk was covered to reduce risk of potential harm. While the current robot design has used many measures to reduce liability concerns, future versions should include even more safety measures to minimize liability as much as possible. Potential improvements include improving battery management and the overall housing of the robot to better prevent water and dust from getting inside the robot. Through risk evaluation, the robot was able to minimize liability concerns making it better for real world use.

4.4 Ethical Considerations

The development of the Gardening Robot raised several important ethical considerations related to user safety, environmental impact, and responsible design practices. First, ensuring the safety of the end user was a key priority. The robot contains moving mechanical parts, such as motors, chains, and sprockets, which pose potential pinch and impact hazards during operation and maintenance. To mitigate these risks, the drivetrain was fully enclosed within a protective chassis, and a master kill switch was installed to allow users to quickly shut down all electronic systems in case of emergency.

One of the most significant concerns involves the weight of the robot. At approximately 80 pounds dry weight, the robot is heavy enough to cause serious injury if it is dropped, mishandled, or operated improperly. Ethical design decisions were made to minimize this risk, including the addition of a master kill switch to immediately disable the system in emergencies, and enclosing moving parts like the drivetrain to prevent pinch point injuries. However, users must still be properly trained to safely move, transport, and operate the robot, particularly around vulnerable individuals such as children or pets.

Electrical safety was also considered in the design. Although the robot operates on a relatively low 12V system, improper wiring or component failure could lead to electrical shorts or overheating. Proper gauge wiring, secured bus bars, and waterproof enclosures were used to minimize the risk of electrical hazards.

Environmental responsibility was another ethical concern. The robot incorporates recyclable materials such as aluminum and ABS plastics where possible. However, components like the SLA battery must be properly recycled at the end of their life to prevent environmental contamination. Future designs could further improve environmental sustainability by selecting components with lower environmental impact and increasing the recyclability of electronics.

Lastly, ethical use of the robot must be considered. The robot is designed for controlled gardening environments and is not intended for use in hazardous conditions or in areas with vulnerable populations such as children or pets without supervision. Proper training and responsible operation are necessary to ensure that the robot does not cause unintended harm.

Overall, ethical decision-making guided the project's focus on minimizing harm, ensuring user safety, and promoting environmental responsibility throughout the design process.

5. Summary

5.1 Work Breakdown Structure

The Work Breakdown Structure (WBS) breaks down the key steps necessary to fully implement the gardening robot prototype. It organizes manufacturing, chassis fabrication, drivetrain assembly, subsystem installation, electronics integration, software setup, and final validation into a structured hierarchy. This structured approach focuses on building, testing, and deploying the final robot and ensures that all critical steps are logically sequenced for efficient implementation.

The WBS was developed by first outlining the main system components of the robot and then identifying the smaller work packages required to complete each one. These work packages represent the specific tasks needed to complete each subsystem and provide a clear structure for managing complexity during the final build phase. This structure helps prevent missed steps, manages dependencies between components, and ensures the robot is assembled and tested systematically. The WBS is presented in **Figure 20.**

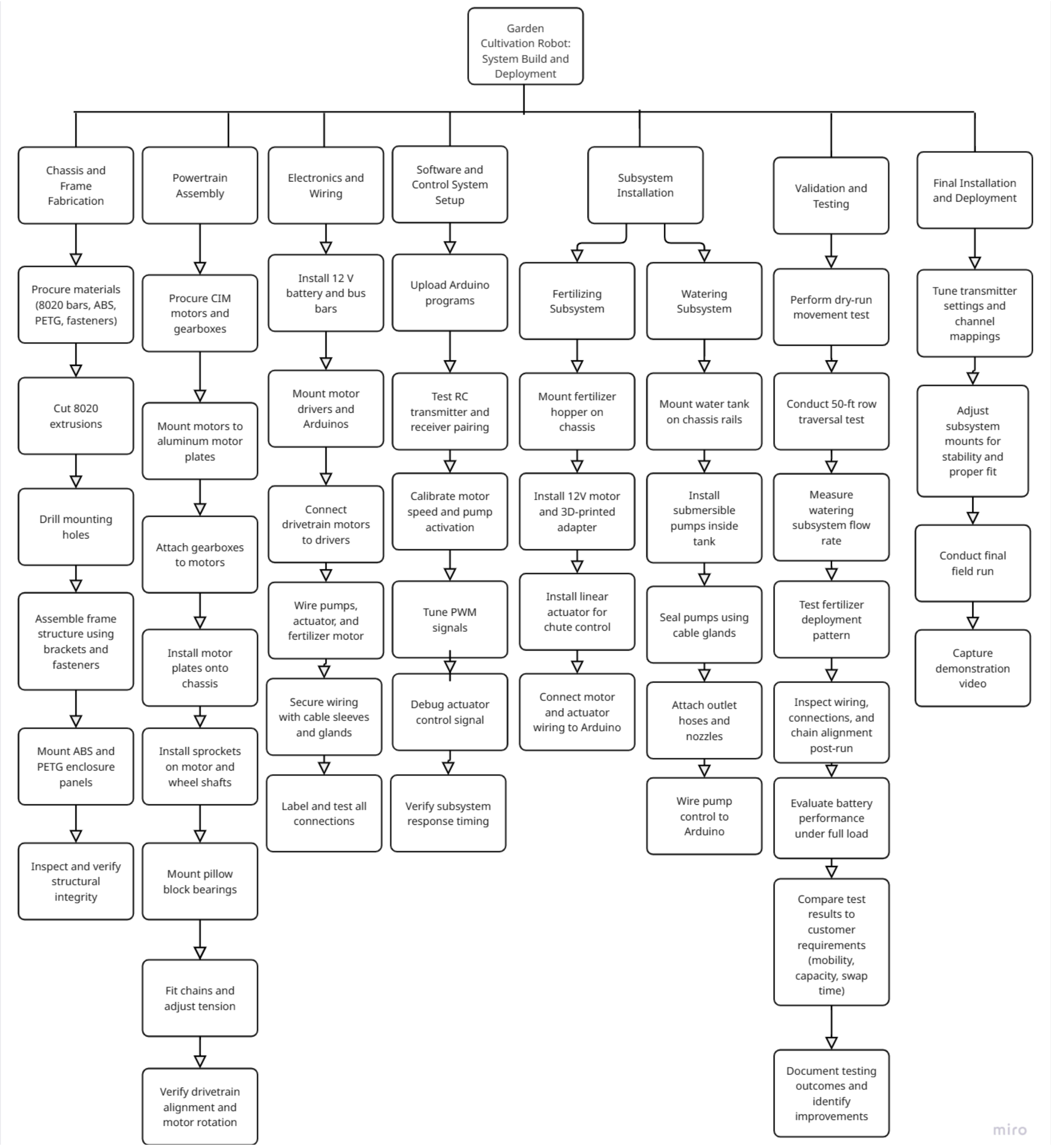

Garden Cultivation Robot: System Build and Deployment
Chassis and Frame Fabrication
Procure materials (8020 bars, ABS, PETG, fasteners)
Cut 8020 extrusions
Drill mounting holes
Assemble frame structure using brackets and fasteners
Mount ABS and PETG enclosure panels
Inspect and verify structural integrity
Powertrain Assembly
Procure CIM motors and gearboxes
Mount motors to aluminum motor plates
Attach gearboxes to motors
Install motor plates onto chassis
Install sprockets on motor and wheel shafts
Mount pillow block bearings
Fit chains and adjust tension
Verify drivetrain alignment and motor rotation
Electronics and Wiring
Install 12 V battery and bus bars
Mount motor drivers and Arduinos
Connect drivetrain motors to drivers
Wire pumps, actuator, and fertilizer motor
Secure wiring with cable sleeves and glands
Label and test all connections
Software and Control System Setup
Upload Arduino programs
Test RC transmitter and receiver pairing
Calibrate motor speed and pump activation
Tune PWM signals
Debug actuator control signal
Verify subsystem response timing
Subsystem Installation
Fertilizing Subsystem
Mount fertilizer hopper on chassis
Install 12V motor and 3D-printed adapter
Install linear actuator for chute control
Connect motor and actuator wiring to Arduino
Watering Subsystem
Mount water tank on chassis rails
Install submersible pumps inside tank
Seal pumps using cable glands
Attach outlet hoses and nozzles
Wire pump control to Arduino
Validation and Testing
Perform dry-run movement test
Conduct 50-ft row traversal test
Measure watering subsystem flow rate
Test fertilizer deployment pattern
Inspect wiring, connections, and chain alignment post-run
Evaluate battery performance under full load
Compare test results to customer requirements (mobility, capacity, swap time)
Document testing outcomes and identify improvements
Final Installation and Deployment
Tune transmitter settings and channel mappings
Adjust subsystem mounts for stability and proper fit
Conduct final field run
Capture demonstration video
miro

Figure 20: Work Breakdown Structure (WBS)

To illustrate how the WBS translates into real tasks, the watering subsystem path demonstrates the full process of assembling and testing the system responsible for delivering water along the robot’s path. The watering subsystem began with mounting the tank, installing and sealing the pumps, and routing tubing to spray nozzles. The pumps were wired to the Arduino through a motor driver and controlled via the RC transmitter.

5.2 Final Gantt Chart

At the start of MEEN 402, our team made a detailed Gantt chart that gave a timeline for all major deliverables or tasks. The start of the semester started with refining both the validation plan as well as the CAD. By the second week, parts were being ordered with the priority being on the robot chassis, powertrain or electronics as the focus. During weeks 4 through 8, the parts came in and the chassis and powertrain were being assembled. Also at this time, the arduino code was being worked on and refined. In weeks 9 and 10, the integration of the powertrain and electronics was done. By the end of week 10, the robot was able to be controlled remotely allowing for it to travel 50 feet and turn. In the last weeks of the semester, the team focused on building the subsystems as well as integrating the electronic components for the subsystems with the rest of the electronic system.

While most of the project stayed on schedule, some parts of the project did take longer than expected. Lack of fasteners in initial orders and long wait times caused the assembly of the robot to be prolonged by a few days. Another major delay that was faced was troubleshooting the arduino code as it could not reliably turn on the drive motors or operate the linear actuator. Fixing this problem required more time for testing and debugging than what was scheduled for it.

Despite these challenges, the team was able to complete the robot with good results. Overall the team was able to stay close to the original gantt chart timeline due team effort and initial early planning for the semester schedule. In the future, adding more time for testing and debugging as well as integration of all components will allow for less risk.

5.3 Technological Development

While the Gardening Robot met its primary design objectives, future development would require additional technological upgrades to fully optimize its performance and usability. A major area identified for improvement is the power system; upgrading from a 12V sealed lead-acid (SLA) battery to a 24V $LiFePO_4$ battery pack would significantly enhance runtime, reduce weight, and improve energy efficiency. Implementation of this higher-voltage system would also require redesigning the drivetrain's motor drivers and potentially selecting new motors compatible with 24V operation.

Expanding the robot's capabilities toward partial autonomy would require development of an onboard vision system, likely integrating low-cost cameras and microcontrollers capable of processing visual data. This would enable features such as plant recognition, soil moisture assessment, and basic GPS waypoint navigation for self-guided row traversal.

One challenge in implementing these enhancements is system complexity. Adding sensors, vision processing units, and autonomous subsystems increases wiring, weight, and software demands, which could strain the robot's structure and existing control architecture. Additionally, ensuring waterproofing and dust resistance for these added electronic components would require new enclosure designs. Addressing these challenges would be essential to achieving a fully autonomous gardening robot in future iterations.

5.4 Design Limitations

Although the Gardening Robot met its primary functional objectives, several limitations remain within the final prototype. The system operates entirely through manual remote control and does not feature any autonomous sensing or navigation capabilities. All movement and subsystem activation depend on continuous user input.

The power system presents another limitation. The 12V sealed lead-acid battery provides sufficient energy for short-term operation but depletes quickly during extended use, especially when running multiple motors simultaneously. Future improvements would require a higher-capacity or higher-voltage battery solution to extend operational time.

While the modular subsystem design was achieved, swapping subsystems currently requires manual wiring connections, resulting in longer swap times than originally targeted. A fully integrated quick-connect system was not developed within the project timeline.

Mechanically, the chain-and-sprocket drivetrain is sensitive to misalignment and chain tension. Over time, operational vibrations may loosen components, requiring regular maintenance to maintain optimal drivetrain performance.

Additionally, the final assembled weight of 84 pounds exceeded the original design target, making the robot heavier and potentially more difficult to transport for some users. Despite these limitations, the Gardening Robot provides a strong foundation for future improvements in modularity, autonomy, and usability.

5.5 Future Work

The gardening robot developed through this capstone project successfully meets the primary objective: traversing a 50-foot garden row and delivering both water and granular fertilizer to plants. Although the project's initial scope has been achieved, the team has identified several opportunities for improvement that future MEEN 401 Capstone teams can pursue to further enhance the robot's performance, durability, autonomy, and usability.

One area for improvement is power management. The robot currently operates six motors on a single 12V battery, leading to rapid depletion during heavy usage. Upgrading to a 24V $LiFePO_4$ battery pack is recommended to approximately double the runtime while offering improved energy density and cycle life. In addition to improving battery life, this will also increase the capabilities of the robot if more complicated subsystems are to be added on to the robot such as a weeding mechanism.

Building off of the increase of voltage from the battery pack, the team has also envisioned a new subsystem that can be implemented to the gardening robot: a weeding subsystem. Though complicated and may be its own capstone project, a weeding subsystem can entail utilizing a claw mechanism to pull out specific weeds that the robot identifies via camera lenses. The weeding subsystem can be taken further and utilize machine learning and ai to identify what plants are weeds and what are crops that the gardener wants to grow. In addition to improving the subsystems of the robot, spliced wire connectors are recommended to be installed for the subsystems to drastically reduce the time to install each subsystem on the chassis of the robot and increase user friendliness.

To further expand the robot's autonomous capabilities, the integration of camera systems is recommended. Cameras would not only support the development of the weeding subsystem but also enable the use of additional sensing technologies. With onboard vision, the robot could incorporate soil moisture probes for targeted watering, use ultrasonic sensors for precise row tracking, and implement GPS waypoints to autonomously navigate larger garden plots. Together, these additions would create a more intelligent and adaptable robot capable of operating with minimal human intervention.

6. Acknowledgements

We would like to express our sincere gratitude to those who supported and guided us throughout the duration of this capstone project. Several individuals went above and beyond to contribute to its success.

First, we thank Dr. Matt Elliott, our project sponsor, for originating the project idea, meeting with us biweekly, and dedicating his time and expertise to guide the project's direction. We are also deeply grateful to Dr. Steve Suh, our MEEN 402 Studio instructor, for believing in our team, keeping us focused, and serving as an exceptional mentor throughout the semester.

We extend our thanks to Dr. Adolfo Delgado, our MEEN 401 Studio professor, for helping us refine our presentation skills and offering thoughtful feedback that strengthened both our technical work and communication. Finally, we thank Dr. Joanna Tsenn for helping us launch this journey in MEEN 401, addressing early concerns, providing valuable initial guidance, and assisting with placement for our robot following MEEN 402.

References

[1]Society of Automotive Engineers (SAE). (2001). *Recommended failure modes and effects analysis (FMEA) practices for non-automobile applications (ARP5580).* SAE International.

[2]Robotis e. Manual. Accessed February 24, 2025. https://emanual.robotis.com/docs/en/platform/turtlebot3/features/#components

[3]Sebastian. (2021, August 10). *Microcontroller connection protocols: W1, I2C, SPI, UART*. Medium. Accessed February 24, 2025. https://medium.com/geekculture/microcontroller-connection-protocols-w1-i2c-spi-uart-7625ad013e60

[4]Embedic. (2022). *ESP32 vs STM32: Which is better and how to choose?* EmbedIC. Accessed February 24, 2025. https://www.embedic.com/technology/details/esp32-vs-stm32--which-is-better-and-how-to-choose-2022.

[5]ResearchGate. n.d. "Centrifugal Spreader Principle Using Two Spinning Discs." Accessed February 24, 2025. https://www.researchgate.net/publication/220050101/figure/fig1/AS:393943167586309@1470934778554/Centrifugal-spreader-principle-using-two-spinning-discs.png.

[6]Sod Solutions. n.d. "Granular vs Liquid Fertilizers." Accessed February 24, 2025. https://sodsolutions.com/lawn-garden-nutrition/granular-vs-liquid-fertilizers/?srsltid=AfmBOooiQy1CDbLkq4HOdE4XkpgZrbyNPlt_RqvosTMH1HU5lRSwQKua.

[7]AndyMark Inc., 2025, "2.5 in. CIM Motor," AndyMark, Inc., https://www.andymark.com/products/2-5-in-cim-motor (accessed Apr. 26, 2025).

[8]Scotts, 2025, "Scotts® Turf Builder® EdgeGuard® Mini Broadcast Spreader," Scotts, https://scotts.com/en-us/shop/lawn-care-deals/scotts-turf-builder-edgeguard-mini-broadcast-spreader/76121B.html [Accessed: Apr. 26, 2025].

[9] SAE International, 2021, *SAE J1739_202101: Potential Failure Mode and Effects Analysis (FMEA) Including Design FMEA, Supplemental FMEA-MSR, and Process FMEA*, SAE International, Warrendale, PA, https://www.sae.org/standards/content/j1739_202101/ (accessed Apr. 27, 2025)

[10] Material Handling Industry of America (MHIA), 2022, "ANSI MH28.2-2022: Design, Testing, and Utilization of Industrial Steel Boltless Shelving," ANSI, Washington, DC, https://webstore.ansi.org/standards/MHIA/ansimh282022 (accessed Apr. 27, 2025).

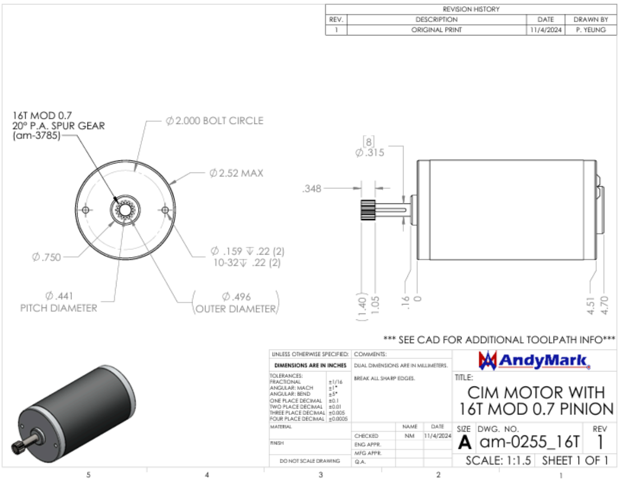
[11] ASME, 2012, "ASME B18.2.1-2012 (R2021): Square, Hex, Heavy Hex, and Askew Head Bolts and Hex, Heavy Hex, Hex Flange, Lobed Head, and Lag Screws (Inch Series)," ASME, New York, NY, https://www.asme.org/codes-standards/find-codes-standards/b18-2-1-square-hex-heavy-hex-askew-head-bolts-hex-heavy-hex-hex-flange-lobed-head-lag-screws (accessed Apr. 27, 2025).
[12] International Organization for Standardization (ISO), 2021, "ISO 73933:2021: Fine bubble technology — General principles for usage and measurement of fine bubbles," ISO, Geneva, Switzerland, https://www.iso.org/standard/73933.html (accessed Apr. 27, 2025).

[13] Occupational Safety and Health Administration (OSHA), 2025, "29 CFR Part 1910 Subpart O – Machinery and Machine Guarding," U.S. Department of Labor, Washington, DC, https://www.ecfr.gov/current/title-29/subtitle-B/chapter-XVII/part-1910/subpart-O (accessed Apr. 27, 2025).

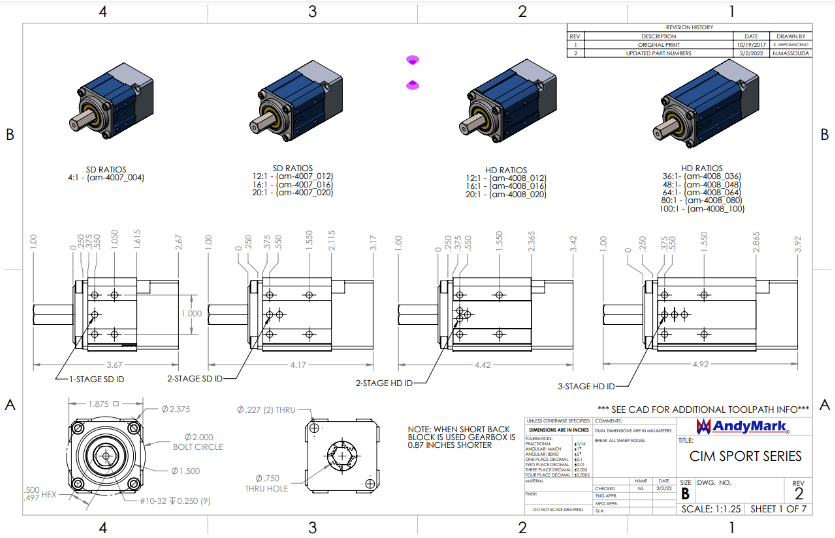
[14] International Electrotechnical Commission (IEC), 2017, "IEC 62133-1:2017: Secondary Cells and Batteries Containing Alkaline or Other Non-Acid Electrolytes—Safety Requirements for Portable Sealed Secondary Cells, and for Batteries Made from Them, for Use in Portable Applications—Part 1: Nickel Systems," IEC, Geneva, Switzerland, https://store.accuristech.com/standards/iec-62133-1-ed-1-0-b-2017?product_id=1945431 (accessed Apr. 27, 2025).

[15] National Fire Protection Association (NFPA), 2022, "NFPA 70: National Electrical Code, 2023 Edition," NFPA, Quincy, MA, https://www.nfpa.org/product/nfpa-70-national-electrical-code-nec/p0070code (accessed Apr. 27, 2025).

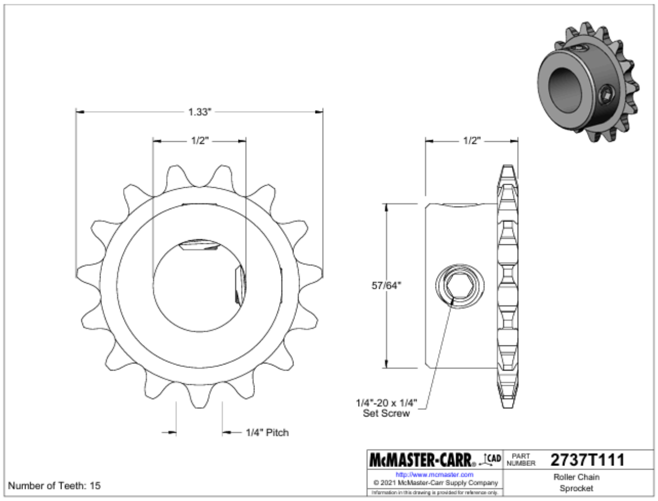
Appendix



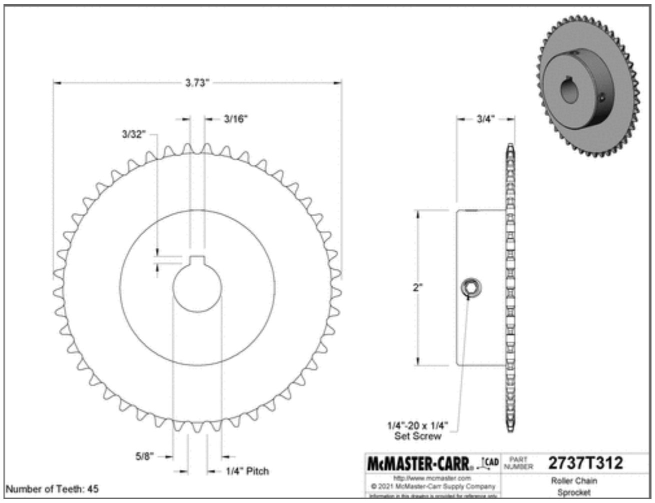
Drawing 1: AndyMark CIM Motor with 16-tooth pinion



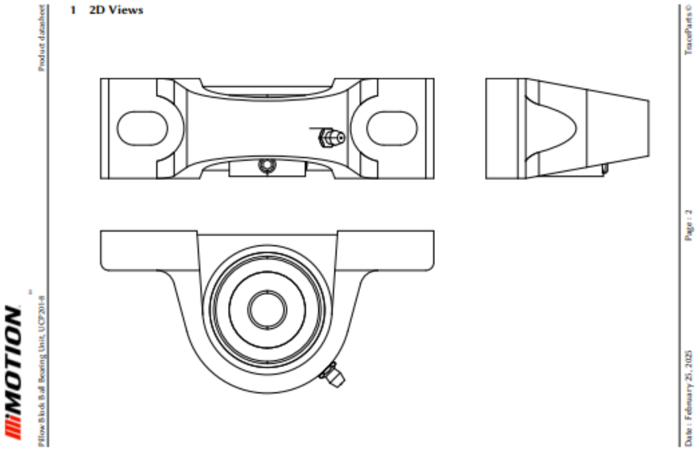
Drawing 2: AndyMark CIM Planetary Gearbox



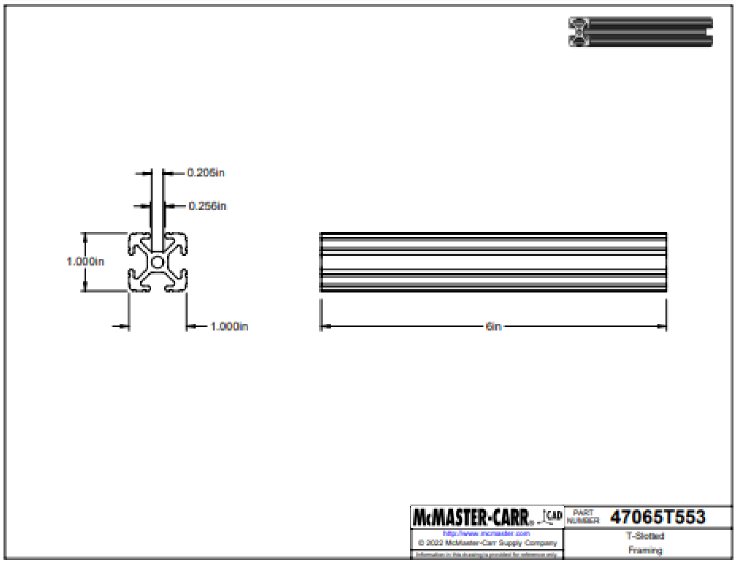
Drawing 3: 15-tooth motor shaft sprocket



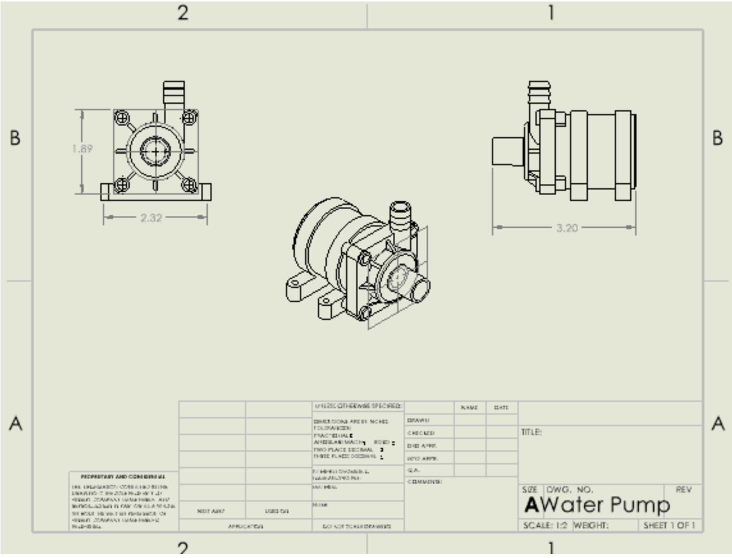
Drawing 4: 45-tooth wheel shaft sprocket



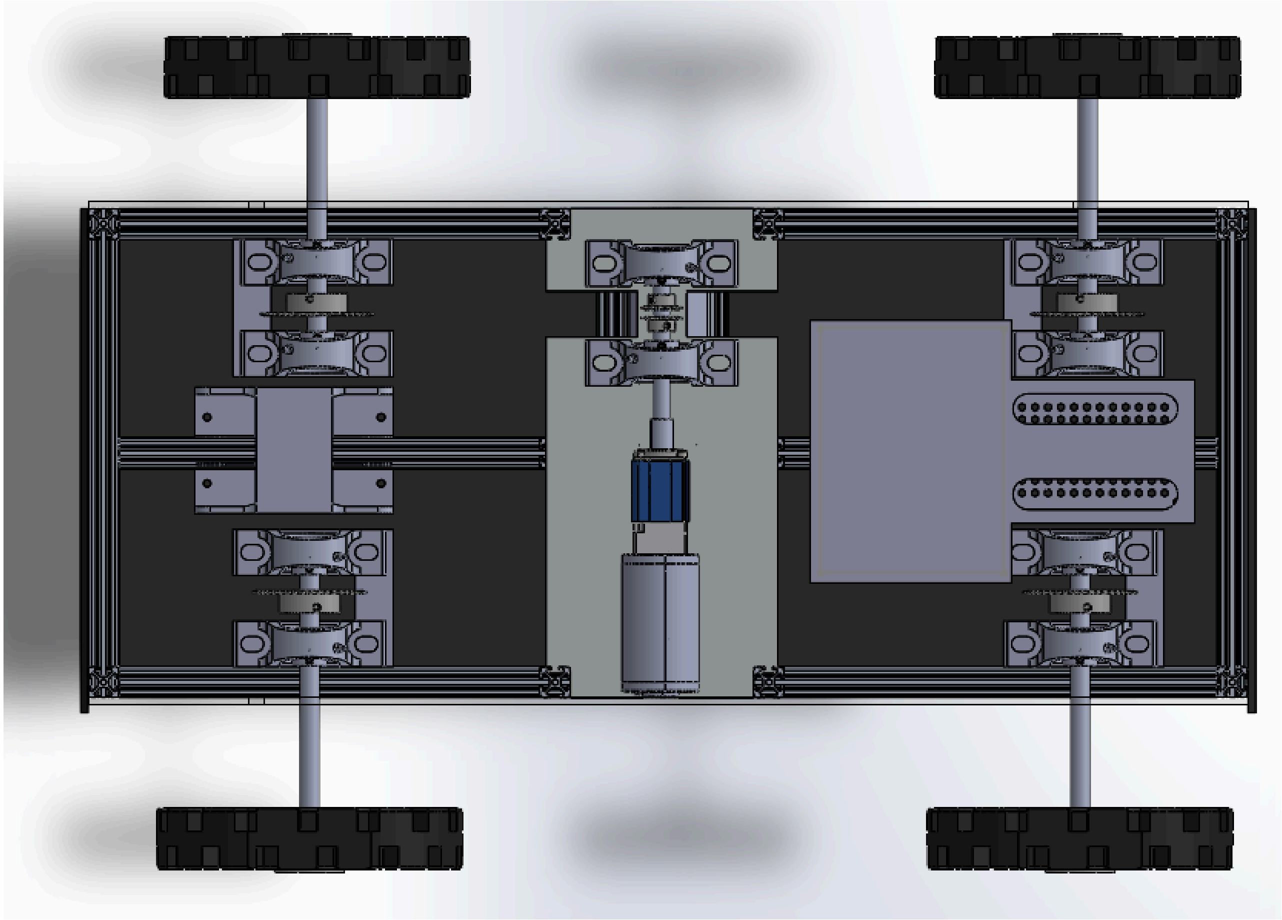
Drawing 5: Block Bearing for Wheel and Motor shafts



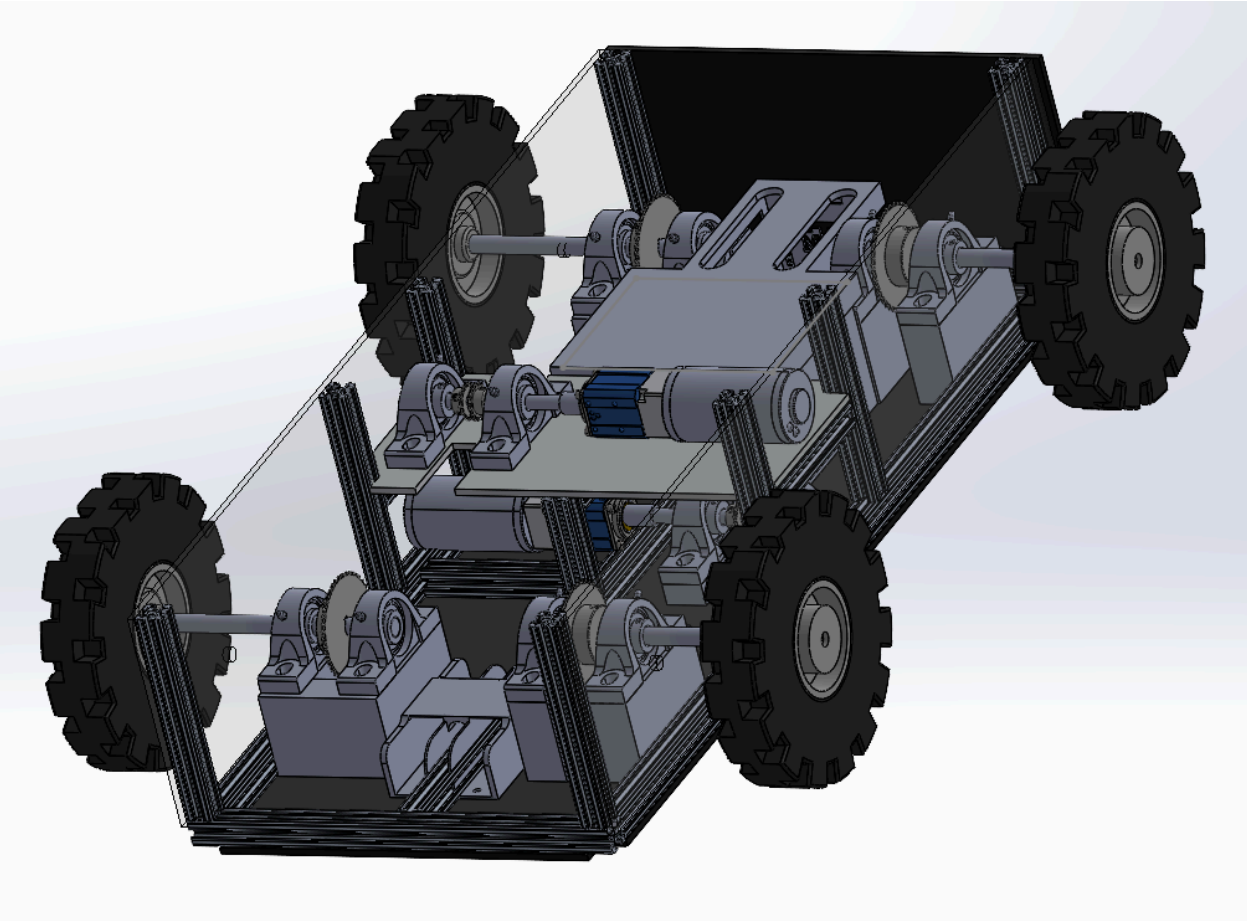
Drawing 6: 6063 T-6 Aluminum Slotted Bars



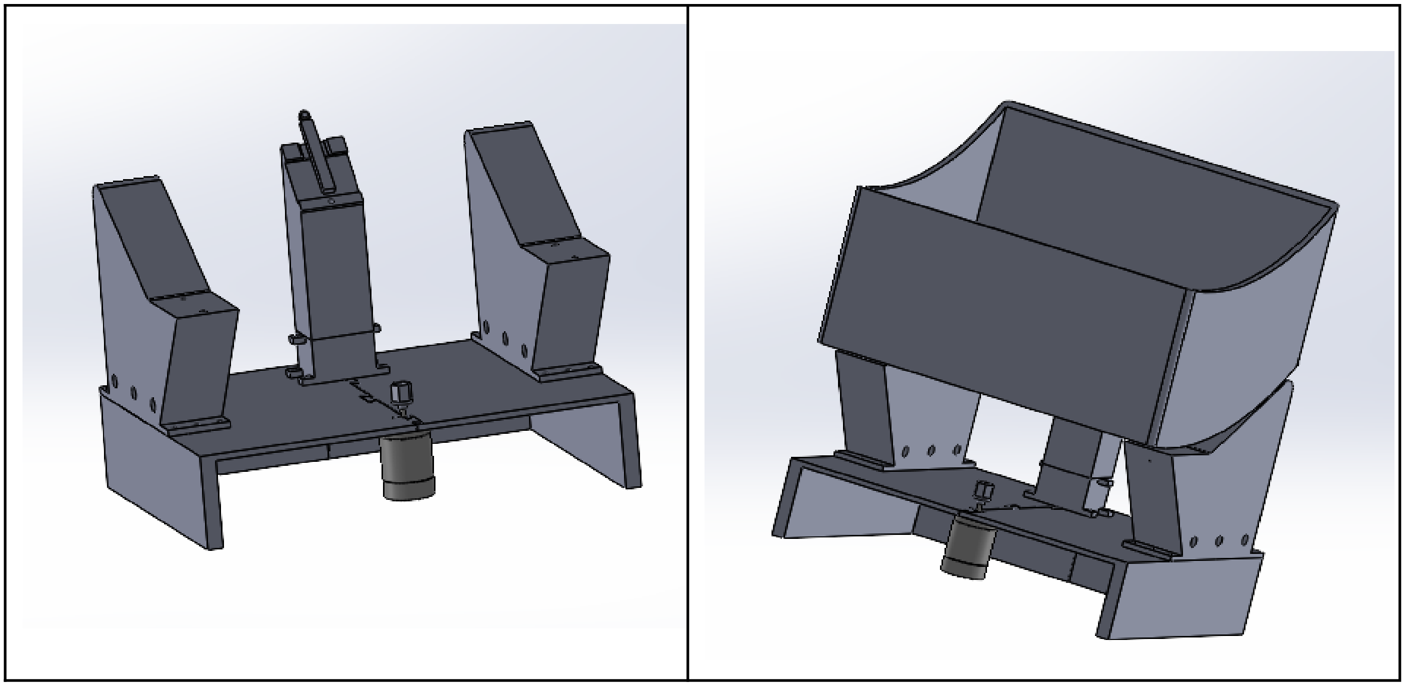
Drawing 7: Submersible Water Pump



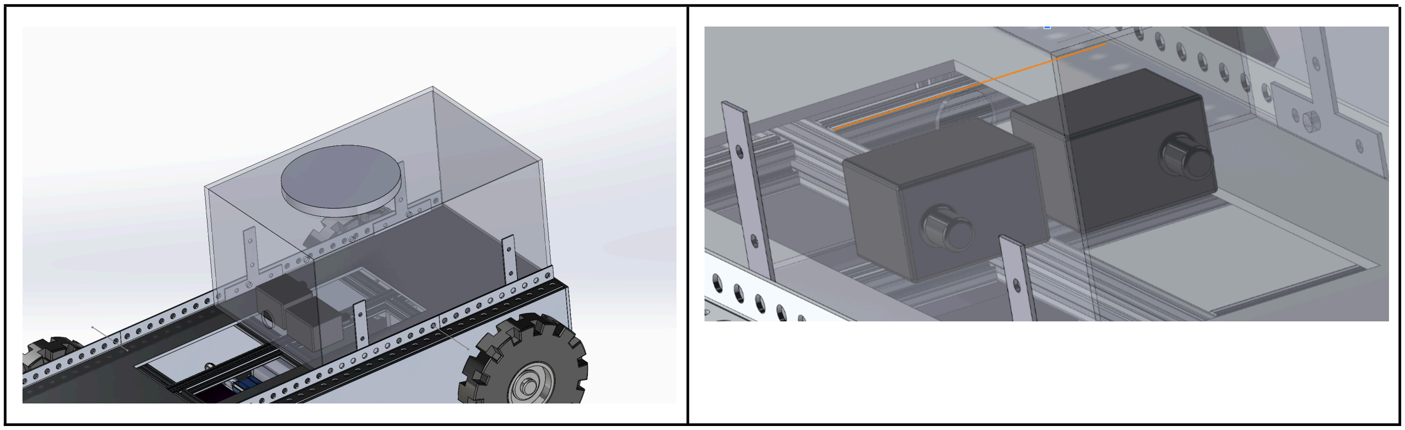
Appendix 1a: Chassis Layout, Top View



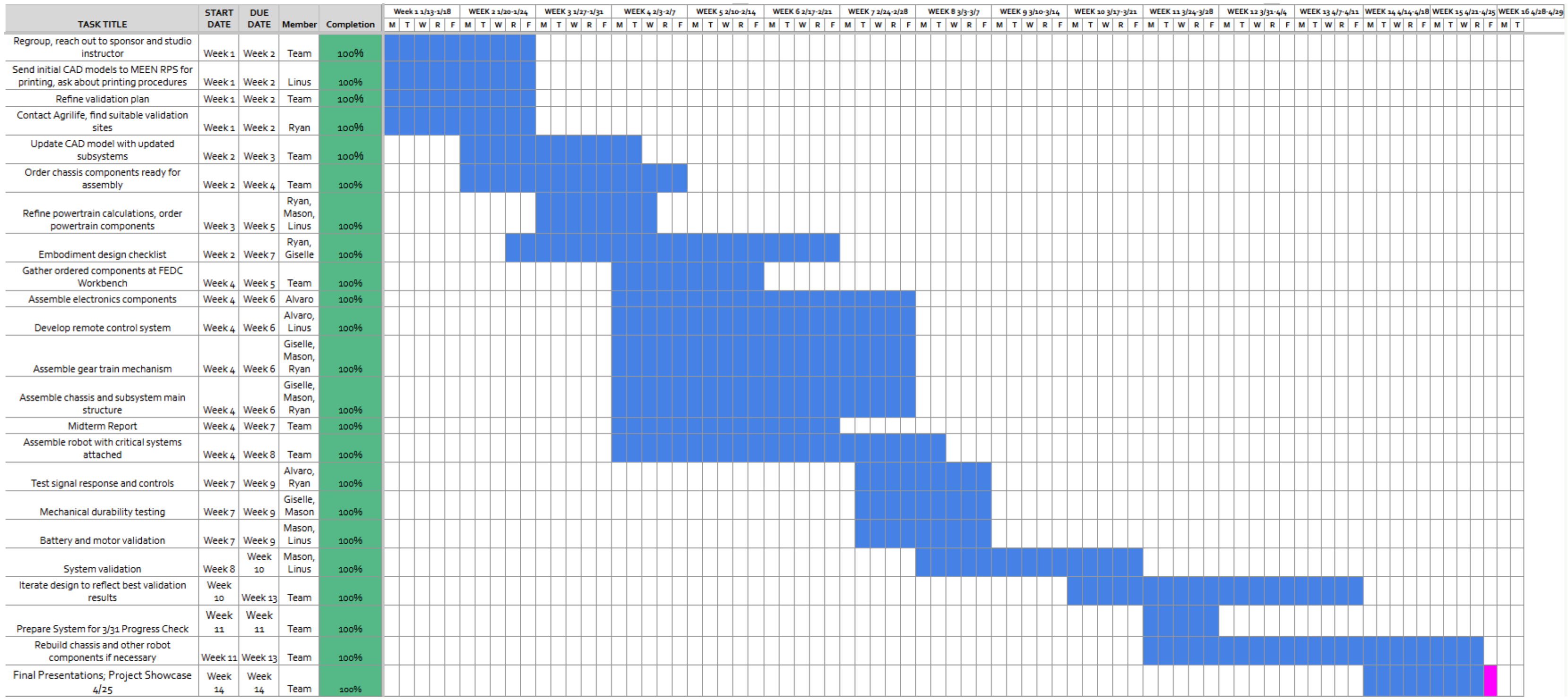
Appendix 1b: Chassis Layout, Isometric View



Appendix 1c: Fertilizer base with motor and linear actuator

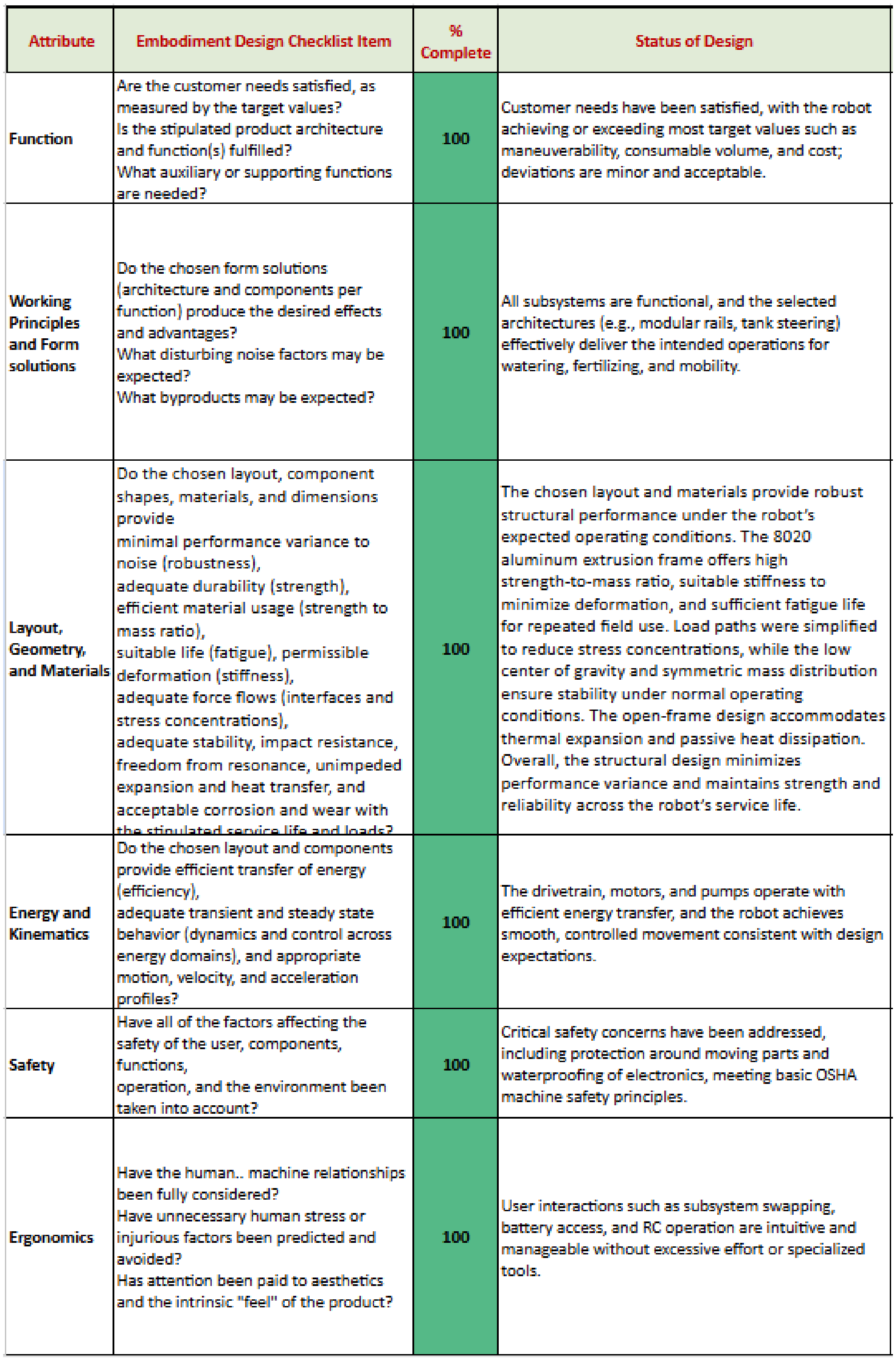


Appendix 1d: Water tank mounted on chassis (left), water pump positioning (right)



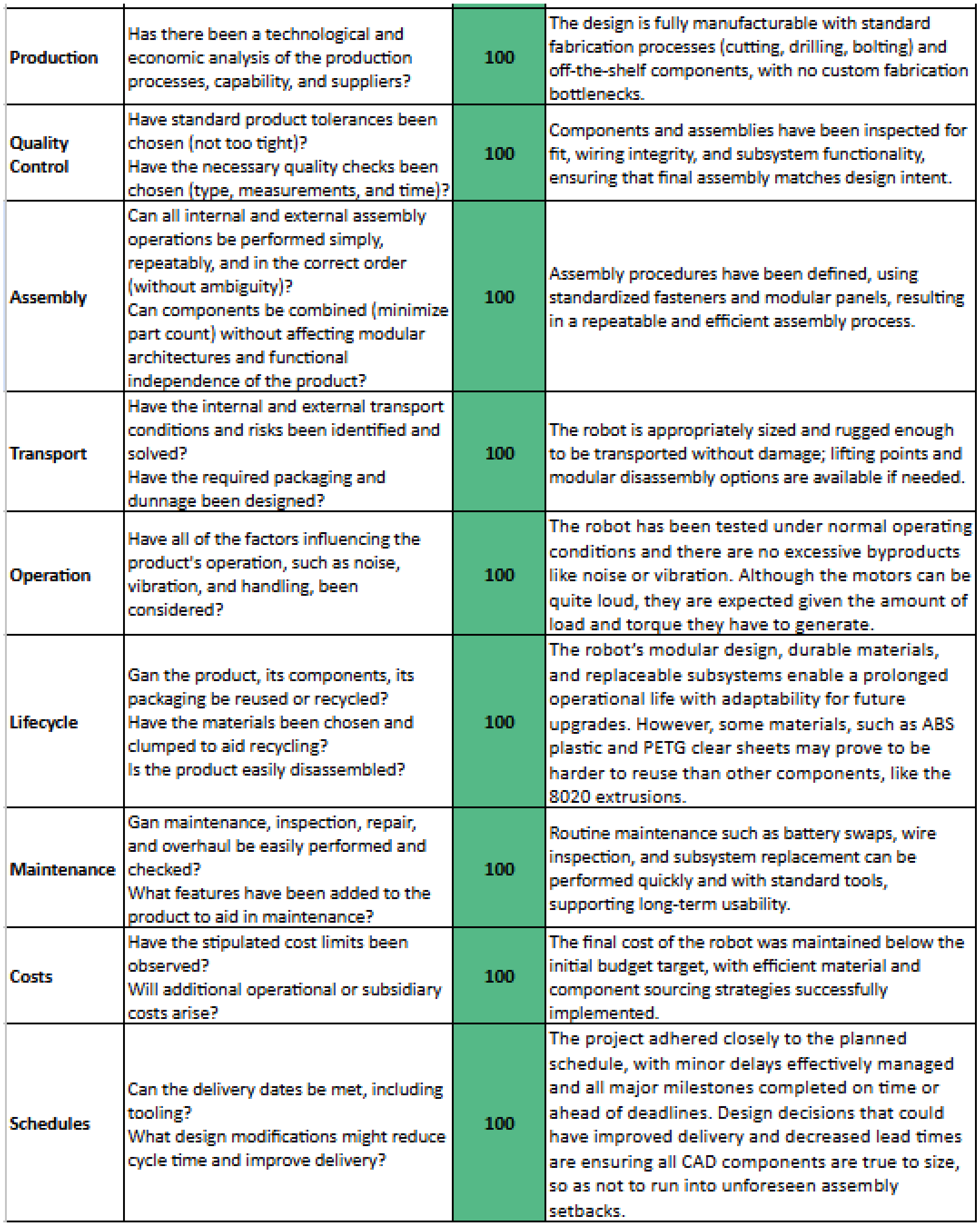
TASK TITLE	START DATE	DUE DATE	Member	Completion
Regroup, reach out to sponsor and studio instructor	Week 1	Week 2	Team	100%
Send initial CAD models to MEEN RPS for printing, ask about printing procedures	Week 1	Week 2	Linus	100%
Refine validation plan	Week 1	Week 2	Team	100%
Contact Agrilife, find suitable validation sites	Week 1	Week 2	Ryan	100%
Update CAD model with updated subsystems	Week 2	Week 3	Team	100%
Order chassis components ready for assembly	Week 2	Week 4	Team	100%
Refine powertrain calculations, order powertrain components	Week 3	Week 5	Ryan, Mason, Linus	100%
Embodiment design checklist	Week 2	Week 7	Ryan, Giselle	100%
Gather ordered components at FEDC Workbench	Week 4	Week 5	Team	100%
Assemble electronics components	Week 4	Week 6	Alvaro	100%
Develop remote control system	Week 4	Week 6	Alvaro, Linus	100%
Assemble gear train mechanism	Week 4	Week 6	Giselle, Mason, Ryan	100%
Assemble chassis and subsystem main structure	Week 4	Week 6	Giselle, Mason, Ryan	100%
Midterm Report	Week 4	Week 7	Team	100%
Assemble robot with critical systems attached	Week 4	Week 8	Team	100%
Test signal response and controls	Week 7	Week 9	Alvaro, Ryan	100%
Mechanical durability testing	Week 7	Week 9	Giselle, Mason	100%
Battery and motor validation	Week 7	Week 9	Mason, Linus	100%
System validation	Week 8	Week 10	Mason, Linus	100%
Iterate design to reflect best validation results	Week 10	Week 13	Team	100%
Prepare System for 3/31 Progress Check	Week 11	Week 11	Team	100%
Rebuild chassis and other robot components if necessary	Week 11	Week 13	Team	100%
Final Presentations, Project Showcase 4/25	Week 14	Week 14	Team	100%

Appendix 2: Final Gantt Chart



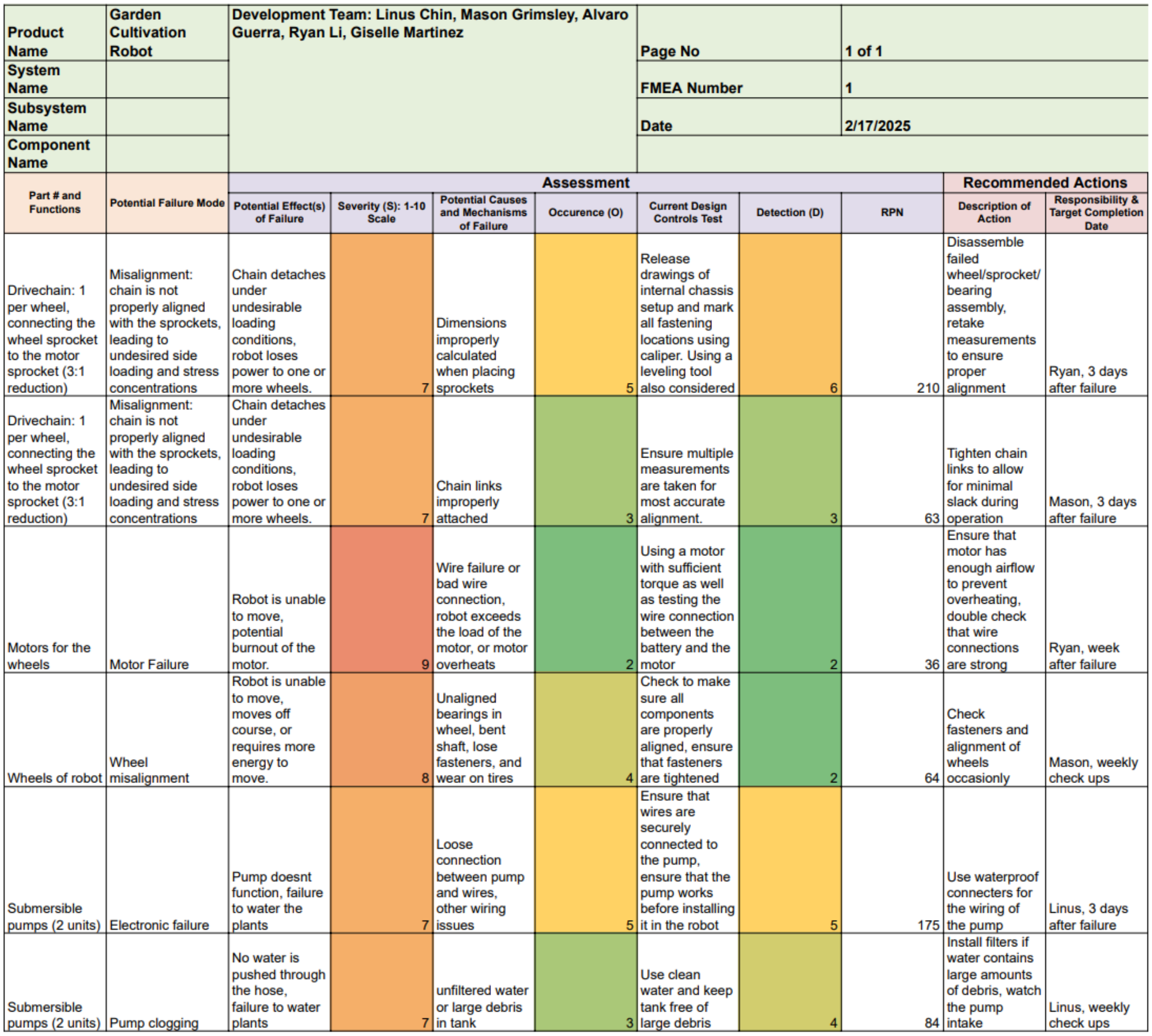
Attribute	Embodiment Design Checklist Item	% Complete	Status of Design
Function	Are the customer needs satisfied, as measured by the target values? Is the stipulated product architecture and function(s) fulfilled? What auxiliary or supporting functions are needed?	100	Customer needs have been satisfied, with the robot achieving or exceeding most target values such as maneuverability, consumable volume, and cost; deviations are minor and acceptable.
Working Principles and Form solutions	Do the chosen form solutions (architecture and components per function) produce the desired effects and advantages? What disturbing noise factors may be expected? What byproducts may be expected?	100	All subsystems are functional, and the selected architectures (e.g., modular rails, tank steering) effectively deliver the intended operations for watering, fertilizing, and mobility.
Layout, Geometry, and Materials	Do the chosen layout, component shapes, materials, and dimensions provide minimal performance variance to noise (robustness), adequate durability (strength), efficient material usage (strength to mass ratio), suitable life (fatigue), permissible deformation (stiffness), adequate force flows (interfaces and stress concentrations), adequate stability, impact resistance, freedom from resonance, unimpeded expansion and heat transfer, and acceptable corrosion and wear with the stipulated service life and loads?	100	The chosen layout and materials provide robust structural performance under the robot's expected operating conditions. The 8020 aluminum extrusion frame offers high strength-to-mass ratio, suitable stiffness to minimize deformation, and sufficient fatigue life for repeated field use. Load paths were simplified to reduce stress concentrations, while the low center of gravity and symmetric mass distribution ensure stability under normal operating conditions. The open-frame design accommodates thermal expansion and passive heat dissipation. Overall, the structural design minimizes performance variance and maintains strength and reliability across the robot's service life.
Energy and Kinematics	Do the chosen layout and components provide efficient transfer of energy (efficiency), adequate transient and steady state behavior (dynamics and control across energy domains), and appropriate motion, velocity, and acceleration profiles?	100	The drivetrain, motors, and pumps operate with efficient energy transfer, and the robot achieves smooth, controlled movement consistent with design expectations.
Safety	Have all of the factors affecting the safety of the user, components, functions, operation, and the environment been taken into account?	100	Critical safety concerns have been addressed, including protection around moving parts and waterproofing of electronics, meeting basic OSHA machine safety principles.
Ergonomics	Have the human.. machine relationships been fully considered? Have unnecessary human stress or injurious factors been predicted and avoided? Has attention been paid to aesthetics and the intrinsic "feel" of the product?	100	User interactions such as subsystem swapping, battery access, and RC operation are intuitive and manageable without excessive effort or specialized tools.

Appendix 3: Embodiment Design Checklist



Production	Has there been a technological and economic analysis of the production processes, capability, and suppliers?	**100**	The design is fully manufacturable with standard fabrication processes (cutting, drilling, bolting) and off-the-shelf components, with no custom fabrication bottlenecks.
Quality Control	Have standard product tolerances been chosen (not too tight)? Have the necessary quality checks been chosen (type, measurements, and time)?	**100**	Components and assemblies have been inspected for fit, wiring integrity, and subsystem functionality, ensuring that final assembly matches design intent.
Assembly	Can all internal and external assembly operations be performed simply, repeatably, and in the correct order (without ambiguity)? Can components be combined (minimize part count) without affecting modular architectures and functional independence of the product?	**100**	Assembly procedures have been defined, using standardized fasteners and modular panels, resulting in a repeatable and efficient assembly process.
Transport	Have the internal and external transport conditions and risks been identified and solved? Have the required packaging and dunnage been designed?	**100**	The robot is appropriately sized and rugged enough to be transported without damage; lifting points and modular disassembly options are available if needed.
Operation	Have all of the factors influencing the product's operation, such as noise, vibration, and handling, been considered?	**100**	The robot has been tested under normal operating conditions and there are no excessive byproducts like noise or vibration. Although the motors can be quite loud, they are expected given the amount of load and torque they have to generate.
Lifecycle	Gan the product, its components, its packaging be reused or recycled? Have the materials been chosen and clumped to aid recycling? Is the product easily disassembled?	**100**	The robot's modular design, durable materials, and replaceable subsystems enable a prolonged operational life with adaptability for future upgrades. However, some materials, such as ABS plastic and PETG clear sheets may prove to be harder to reuse than other components, like the 8020 extrusions.
Maintenance	Gan maintenance, inspection, repair, and overhaul be easily performed and checked? What features have been added to the product to aid in maintenance?	**100**	Routine maintenance such as battery swaps, wire inspection, and subsystem replacement can be performed quickly and with standard tools, supporting long-term usability.
Costs	Have the stipulated cost limits been observed? Will additional operational or subsidiary costs arise?	**100**	The final cost of the robot was maintained below the initial budget target, with efficient material and component sourcing strategies successfully implemented.
Schedules	Can the delivery dates be met, including tooling? What design modifications might reduce cycle time and improve delivery?	**100**	The project adhered closely to the planned schedule, with minor delays effectively managed and all major milestones completed on time or ahead of deadlines. Design decisions that could have improved delivery and decreased lead times are ensuring all CAD components are true to size, so as not to run into unforeseen assembly setbacks.

Appendix 4: Embodiment Design Checklist, cont'd.

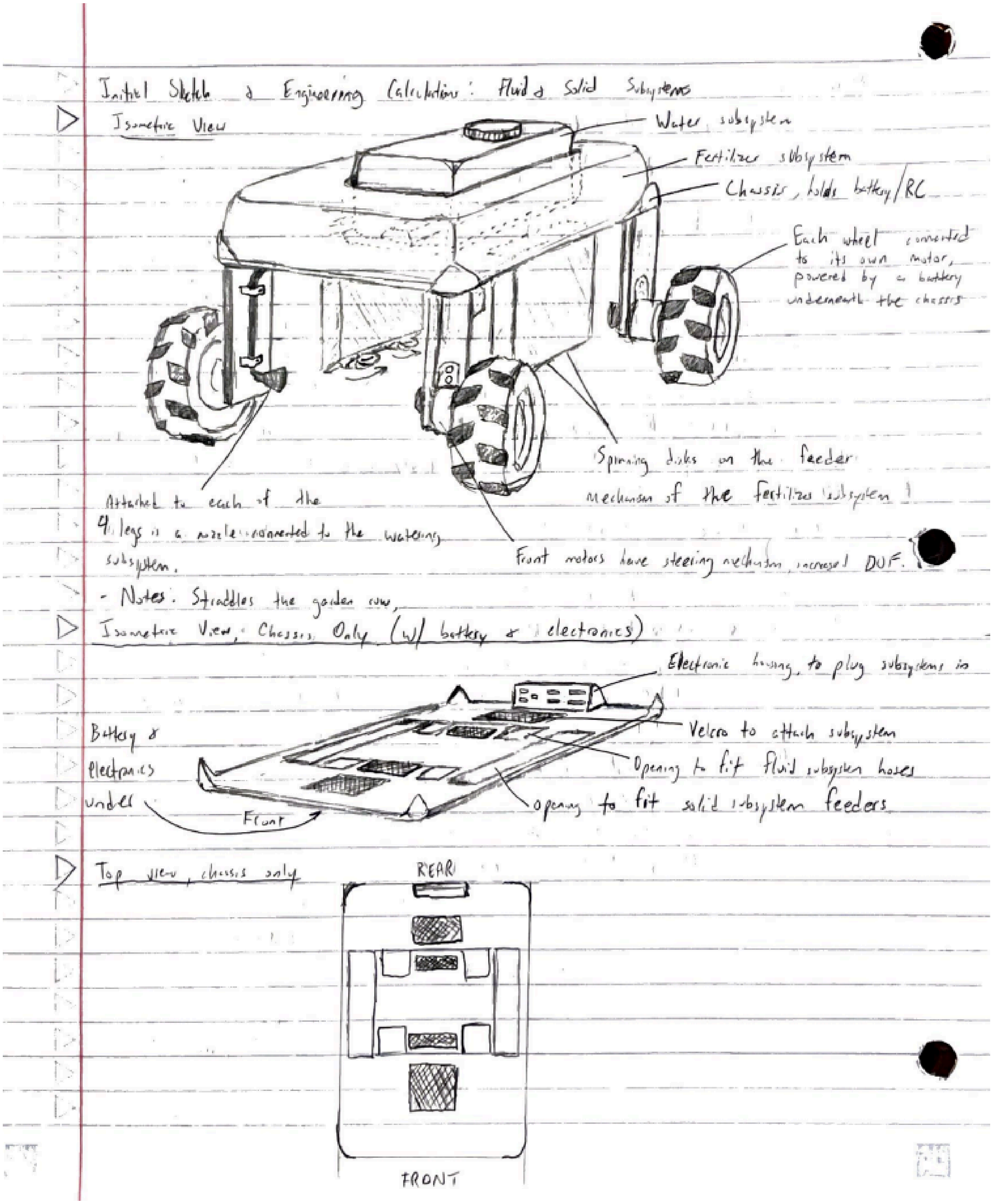


Product Name	Garden Cultivation Robot	Development Team: Linus Chin, Mason Grimsley, Alvaro Guerra, Ryan Li, Giselle Martinez				Page No		1 of 1		
System Name						FMEA Number		1		
Subsystem Name						Date		2/17/2025		
Component Name										
Part # and Functions	Potential Failure Mode	Assessment							Recommended Actions	
		Potential Effect(s) of Failure	Severity (S): 1-10 Scale	Potential Causes and Mechanisms of Failure	Occurence (O)	Current Design Controls Test	Detection (D)	RPN	Description of Action	Responsibility & Target Completion Date
Drivechain: 1 per wheel, connecting the wheel sprocket to the motor sprocket (3:1 reduction)	Misalignment: chain is not properly aligned with the sprockets, leading to undesired side loading and stress concentrations	Chain detaches under undesirable loading conditions, robot loses power to one or more wheels.	7	Dimensions improperly calculated when placing sprockets	5	Release drawings of internal chassis setup and mark all fastening locations using caliper. Using a leveling tool also considered	6	210	Disassemble failed wheel/sprocket/bearing assembly, retake measurements to ensure proper alignment	Ryan, 3 days after failure
Drivechain: 1 per wheel, connecting the wheel sprocket to the motor sprocket (3:1 reduction)	Misalignment: chain is not properly aligned with the sprockets, leading to undesired side loading and stress concentrations	Chain detaches under undesirable loading conditions, robot loses power to one or more wheels.	7	Chain links improperly attached	3	Ensure multiple measurements are taken for most accurate alignment.	3	63	Tighten chain links to allow for minimal slack during operation	Mason, 3 days after failure
Motors for the wheels	Motor Failure	Robot is unable to move, potential burnout of the motor.	9	Wire failure or bad wire connection, robot exceeds the load of the motor, or motor overheats	2	Using a motor with sufficient torque as well as testing the wire connection between the battery and the motor	2	36	Ensure that motor has enough airflow to prevent overheating, double check that wire connections are strong	Ryan, week after failure
Wheels of robot	Wheel misalignment	Robot is unable to move, moves off course, or requires more energy to move.	8	Unaligned bearings in wheel, bent shaft, lose fasteners, and wear on tires	4	Check to make sure all components are properly aligned, ensure that fasteners are tightened	2	64	Check fasteners and alignment of wheels occasionly	Mason, weekly check ups
Submersible pumps (2 units)	Electronic failure	Pump doesnt function, failure to water the plants	7	Loose connection between pump and wires, other wiring issues	5	Ensure that wires are securely connected to the pump, ensure that the pump works before installing it in the robot	5	175	Use waterproof connecters for the wiring of the pump	Linus, 3 days after failure
Submersible pumps (2 units)	Pump clogging	No water is pushed through the hose, failure to water plants	7	unfiltered water or large debris in tank	3	Use clean water and keep tank free of large debris	4	84	Install filters if water contains large amounts of debris, watch the pump intake	Linus, weekly check ups

Appendix 5: FMECA Matrix

Watering system: hose and nozzle	Water leakage in the hose and nozzle	Drop in pressure and velocity, water loss, potential electrical damage	4	loose fittings between the nozzle and hose, lose fittings between hose and pump, high water pressure	3	Used material that wont degrade under high pressure, use nozzle that is barbed to ensure tight fit	2	24	Ensure that connectors and nozzle are leak proof	Linus, 3 days after failure
Fertilizer Motor	Motor Failure	No fertilizer spread, uneven spread of fertilizer	7	Overloading the motor, motor burnout, wire connection issues	5	Use correct power supply, test motor/wire connection before installing on robot	4	140	Test the motor under loaded conditions	Giselle, week after failure
Fertilizer System: Hopper	Clogged fertilizer funnel	No fertilizer spread, uneven spread of fertilizer	4	Moisture exposure or fertilizer not broken up	2	Make sure to break up fertilizer before putting it in hopper	3	24	Install ways to prevent moisture exposure	Giselle, weekly check ups
Battery	Battery failure	Complete loss of power, unable to operate anything on the robot	10	aging, over current draw, bad wiring connections	6	Select correct battery, ensure that current draw is within the limits of the battery	3	180	Ensure that battery can power all components at same time before installing it on the robot	Alvaro, 3 days after failure
Arduino Control System	Failure to send signal from ardunio to motors and pumps	Robot fails to move, pump water or dispense fertilizer or make it difficult to do so	9	wrong software or software bug, electrical failure, bad wire connections	5	To work out software steps before the integration with the rest of the robot	4	180	To test the arduino control system multiple times before installing it in the robot	Alvaro, 3 days after failure
Arduino Control System	Communication loss between the remote control, the receiver, and the arduino	Unable to control the robot remotely	8	Signal interference, faulty transmitter/receive	5	Test the latency of the controller, make sure it receives signal close before moving further	5	200	To test the communication before installing it on robot, ensure that buttons on controller start motors or pumps	Alvaro, 3 days after failure
Chassis (Structural Support)	Chassis fails under robot weight	Frame bending or breaking. Unalignment of major components that could effect efficiency of the robot	9	Excessive load, undersized brackets and fasteners, excessive load	4	Perform load analysis and choose high strength/low weight material	2	72	Reinforce aluminum frame at stress points and conitinued load analysis of the robot	Mason, week after failure

Appendix 6: FMECA Matrix, cont'd.



Initial Sketch & Engineering Calculations: Fluid & Solid Subsystems

Isometric View

Water subsystem
Fertilizer subsystem
Chassis, holds battery/RC
Each wheel connected to its own motor, powered by a battery underneath the chassis
Spinning disks on the feeder mechanism of the fertilizer subsystem
Attached to each of the 4 legs is a nozzle connected to the watering subsystem.
Front motors have steering mechanism, increased DOF.

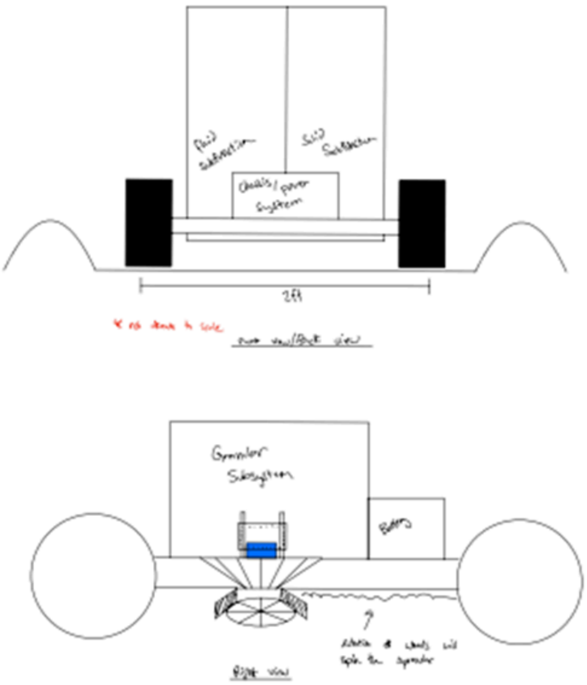
- Notes: Straddles the garden row,

Isometric View, Chassis Only (w/ battery & electronics)

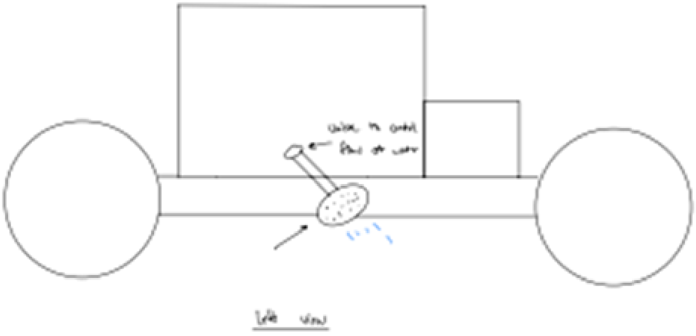
Electronic housing, to plug subsystems in
Velcro to attach subsystem
Battery & electronics under
Front
Opening to fit fluid subsystem hoses
opening to fit solid subsystem feeders

Top view, chassis only

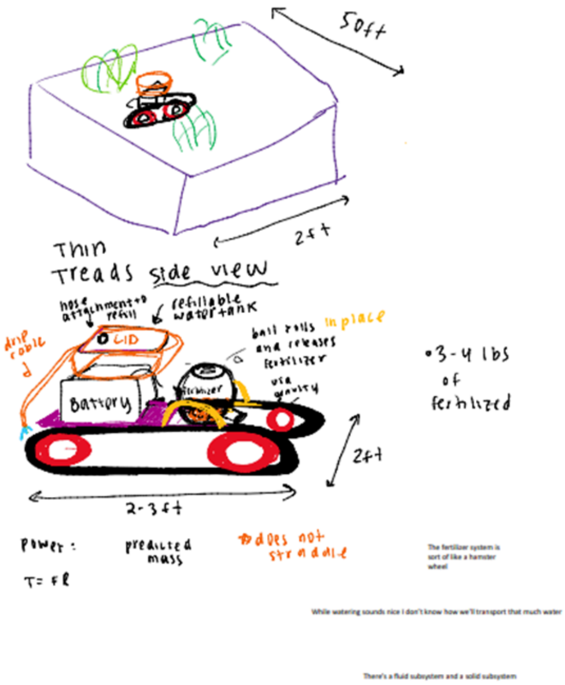
Appendix 7: Ryan’s initial robot design concept



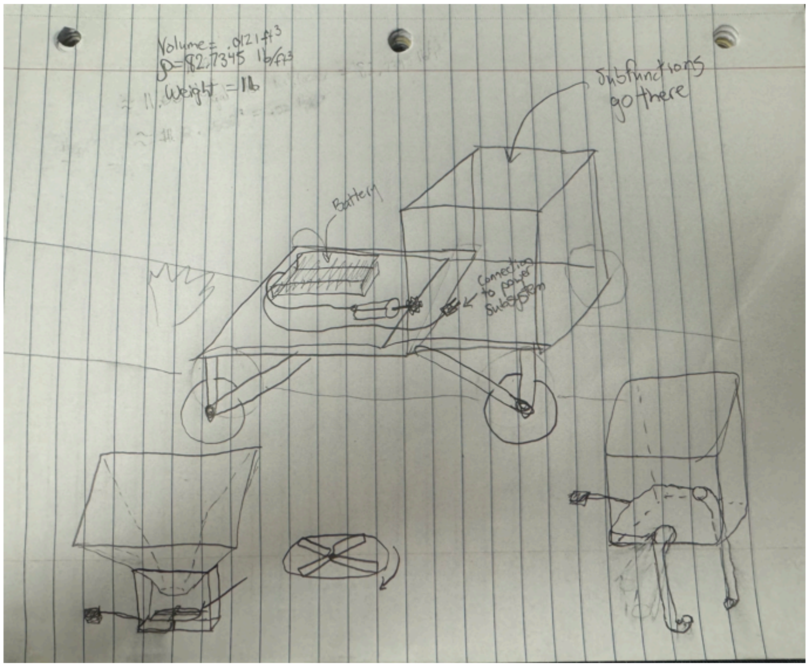
Appendix 8: Linus' initial robot design concept (front, side view)



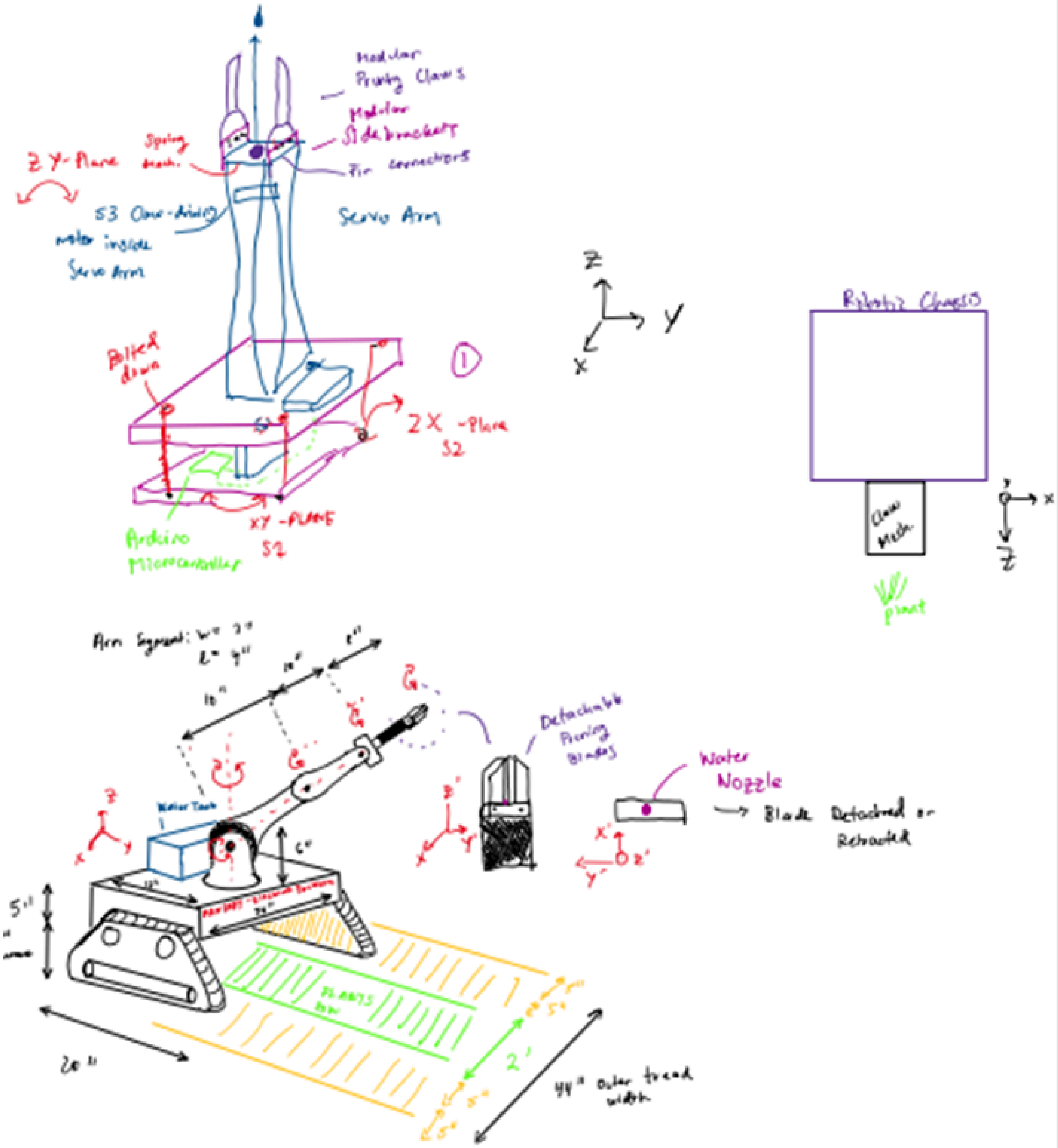
Appendix 9: Linus' initial robot design concept (side view)


50ft
2ft
Thin
Treads side view
hose attachment to refill
refillable watertank
drip cable
LID
Battery
fertilizer
ball rolls in place and releases fertilizer via gravity
2ft
2-3ft
•3-4 lbs of fertilized
Power:
T= Fℓ
predicted mass
*does not straddle
The fertilizer system is sort of like a hamster wheel
While watering sounds nice I don't know how we'll transport that much water
There's a fluid subsystem and a solid subsystem

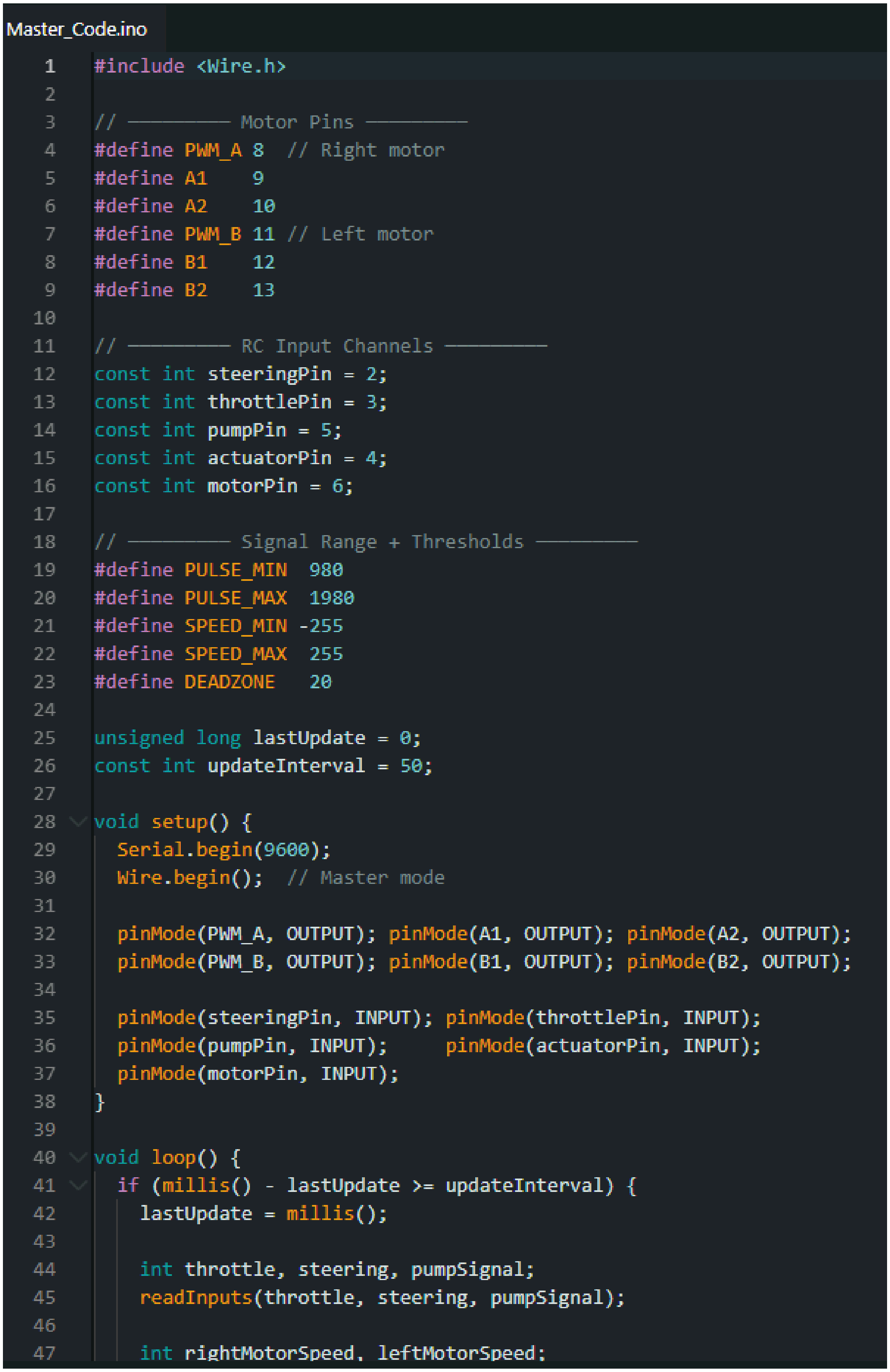
Appendix 10: Giselle's initial robot design concept



Appendix 11: Mason's initial robot design concept

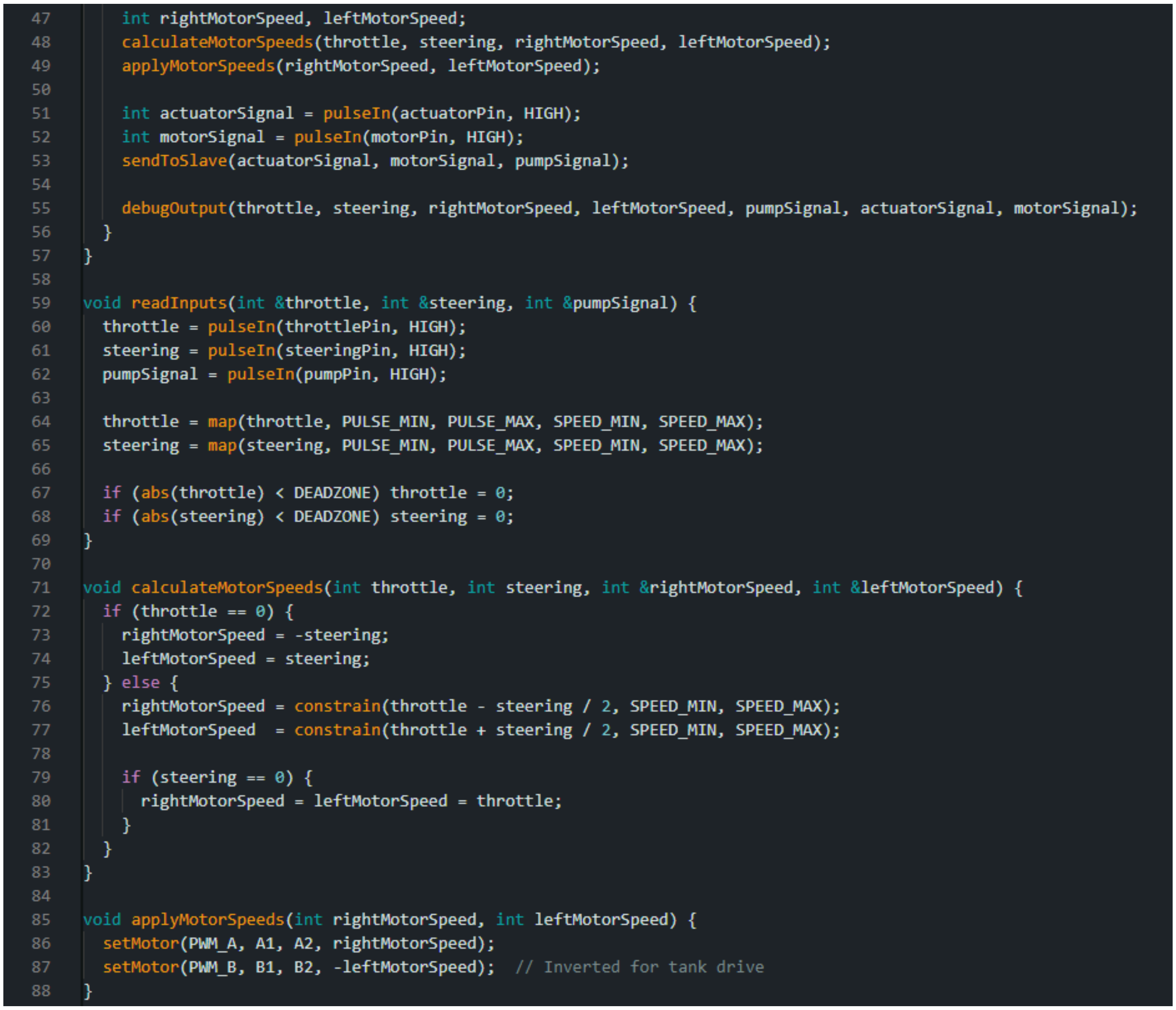


Appendix 12: Alvaro's initial robot design concept

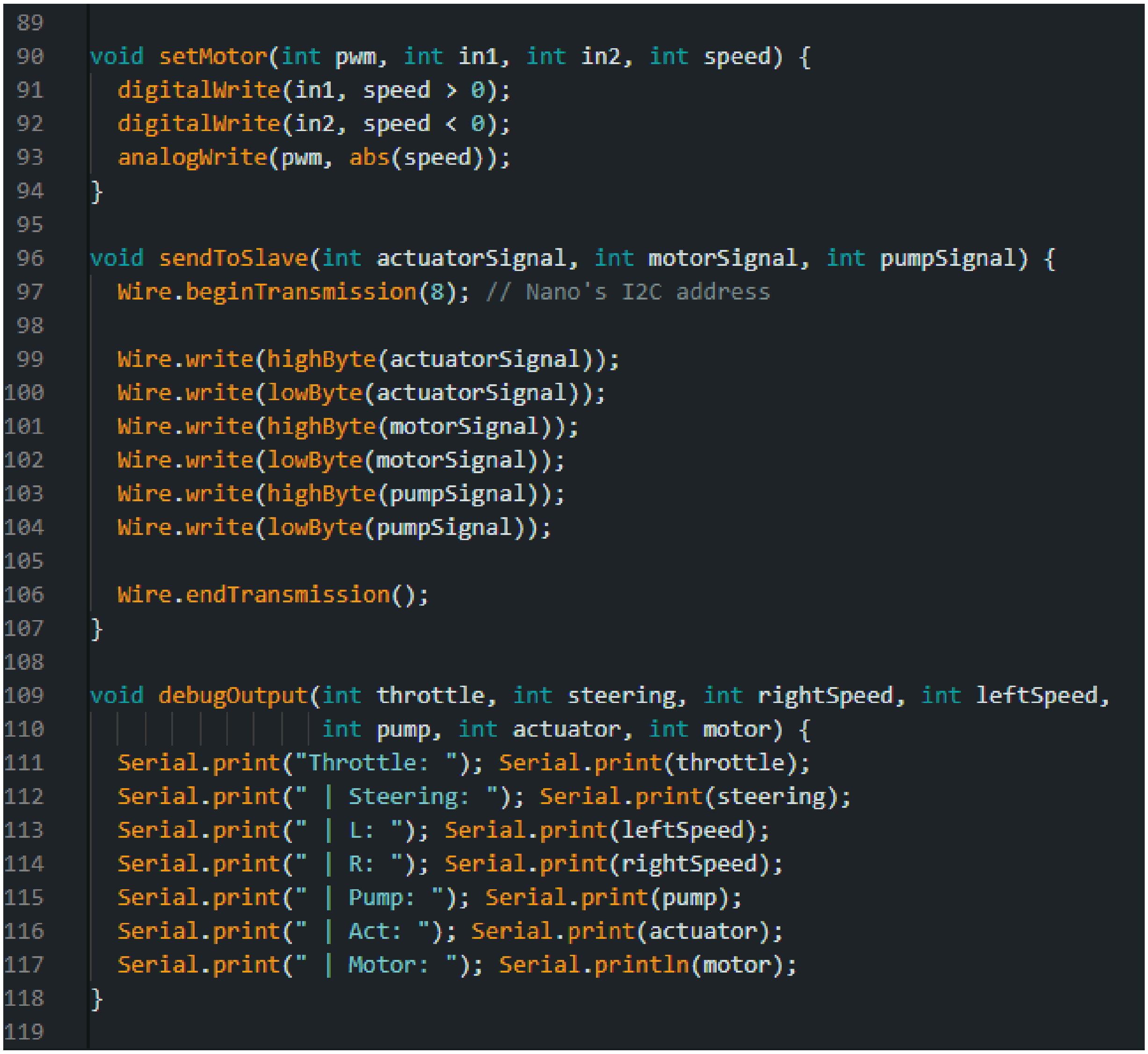


Master_Code.ino

```
#include <Wire.h>

// ────────── Motor Pins ──────────
#define PWM_A 8  // Right motor
#define A1    9
#define A2    10
#define PWM_B 11 // Left motor
#define B1    12
#define B2    13

// ────────── RC Input Channels ──────────
const int steeringPin = 2;
const int throttlePin = 3;
const int pumpPin = 5;
const int actuatorPin = 4;
const int motorPin = 6;

// ────────── Signal Range + Thresholds ──────────
#define PULSE_MIN  980
#define PULSE_MAX  1980
#define SPEED_MIN -255
#define SPEED_MAX  255
#define DEADZONE   20

unsigned long lastUpdate = 0;
const int updateInterval = 50;

void setup() {
  Serial.begin(9600);
  Wire.begin();  // Master mode

  pinMode(PWM_A, OUTPUT); pinMode(A1, OUTPUT); pinMode(A2, OUTPUT);
  pinMode(PWM_B, OUTPUT); pinMode(B1, OUTPUT); pinMode(B2, OUTPUT);

  pinMode(steeringPin, INPUT); pinMode(throttlePin, INPUT);
  pinMode(pumpPin, INPUT);     pinMode(actuatorPin, INPUT);
  pinMode(motorPin, INPUT);
}

void loop() {
  if (millis() - lastUpdate >= updateInterval) {
    lastUpdate = millis();

    int throttle, steering, pumpSignal;
    readInputs(throttle, steering, pumpSignal);

    int rightMotorSpeed, leftMotorSpeed;
```

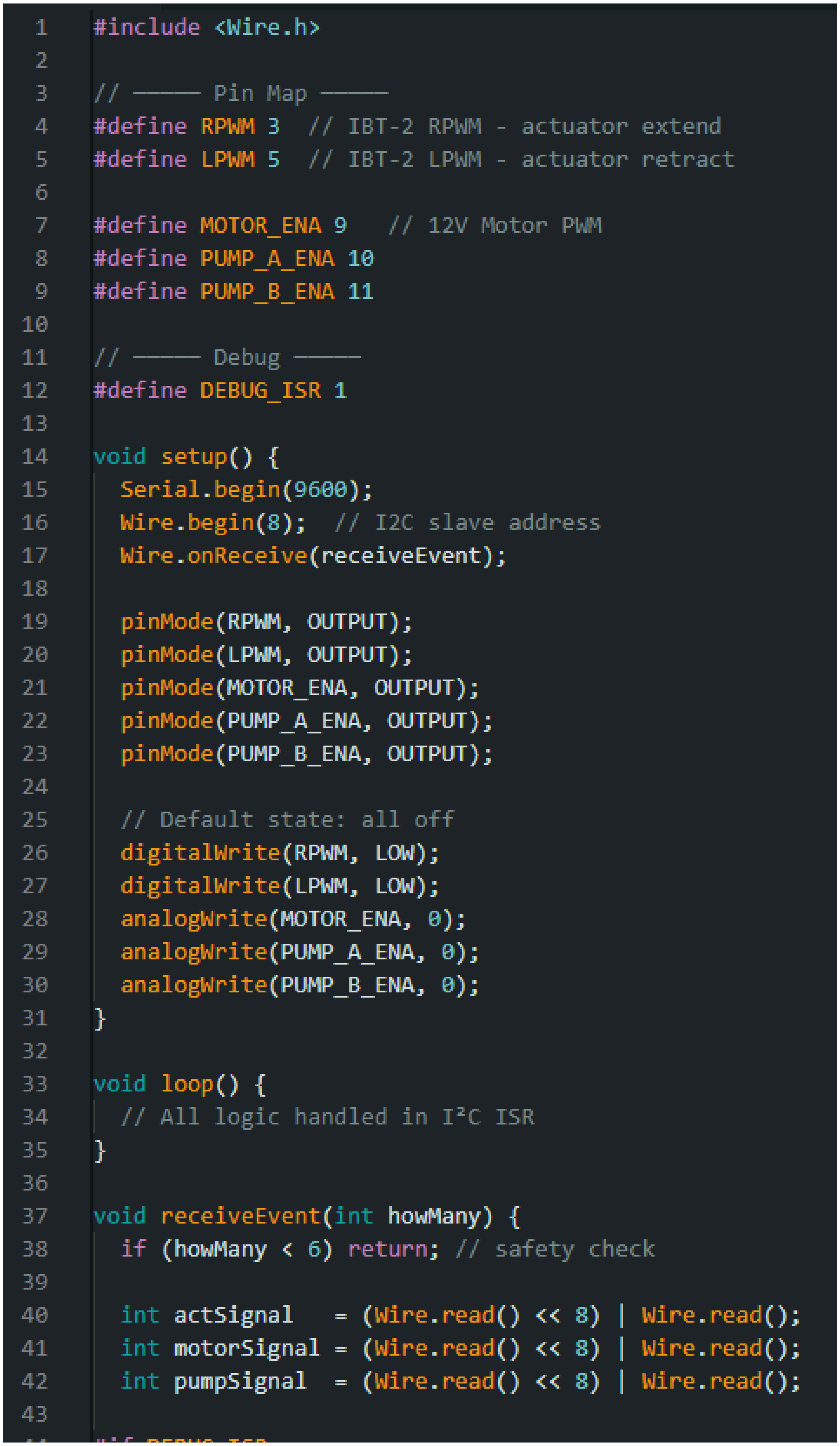


```
    int rightMotorSpeed, leftMotorSpeed;
    calculateMotorSpeeds(throttle, steering, rightMotorSpeed, leftMotorSpeed);
    applyMotorSpeeds(rightMotorSpeed, leftMotorSpeed);

    int actuatorSignal = pulseIn(actuatorPin, HIGH);
    int motorSignal = pulseIn(motorPin, HIGH);
    sendToSlave(actuatorSignal, motorSignal, pumpSignal);

    debugOutput(throttle, steering, rightMotorSpeed, leftMotorSpeed, pumpSignal, actuatorSignal, motorSignal);
  }
}

void readInputs(int &throttle, int &steering, int &pumpSignal) {
  throttle = pulseIn(throttlePin, HIGH);
  steering = pulseIn(steeringPin, HIGH);
  pumpSignal = pulseIn(pumpPin, HIGH);

  throttle = map(throttle, PULSE_MIN, PULSE_MAX, SPEED_MIN, SPEED_MAX);
  steering = map(steering, PULSE_MIN, PULSE_MAX, SPEED_MIN, SPEED_MAX);

  if (abs(throttle) < DEADZONE) throttle = 0;
  if (abs(steering) < DEADZONE) steering = 0;
}

void calculateMotorSpeeds(int throttle, int steering, int &rightMotorSpeed, int &leftMotorSpeed) {
  if (throttle == 0) {
    rightMotorSpeed = -steering;
    leftMotorSpeed = steering;
  } else {
    rightMotorSpeed = constrain(throttle - steering / 2, SPEED_MIN, SPEED_MAX);
    leftMotorSpeed  = constrain(throttle + steering / 2, SPEED_MIN, SPEED_MAX);

    if (steering == 0) {
      rightMotorSpeed = leftMotorSpeed = throttle;
    }
  }
}

void applyMotorSpeeds(int rightMotorSpeed, int leftMotorSpeed) {
  setMotor(PWM_A, A1, A2, rightMotorSpeed);
  setMotor(PWM_B, B1, B2, -leftMotorSpeed);  // Inverted for tank drive
}
```

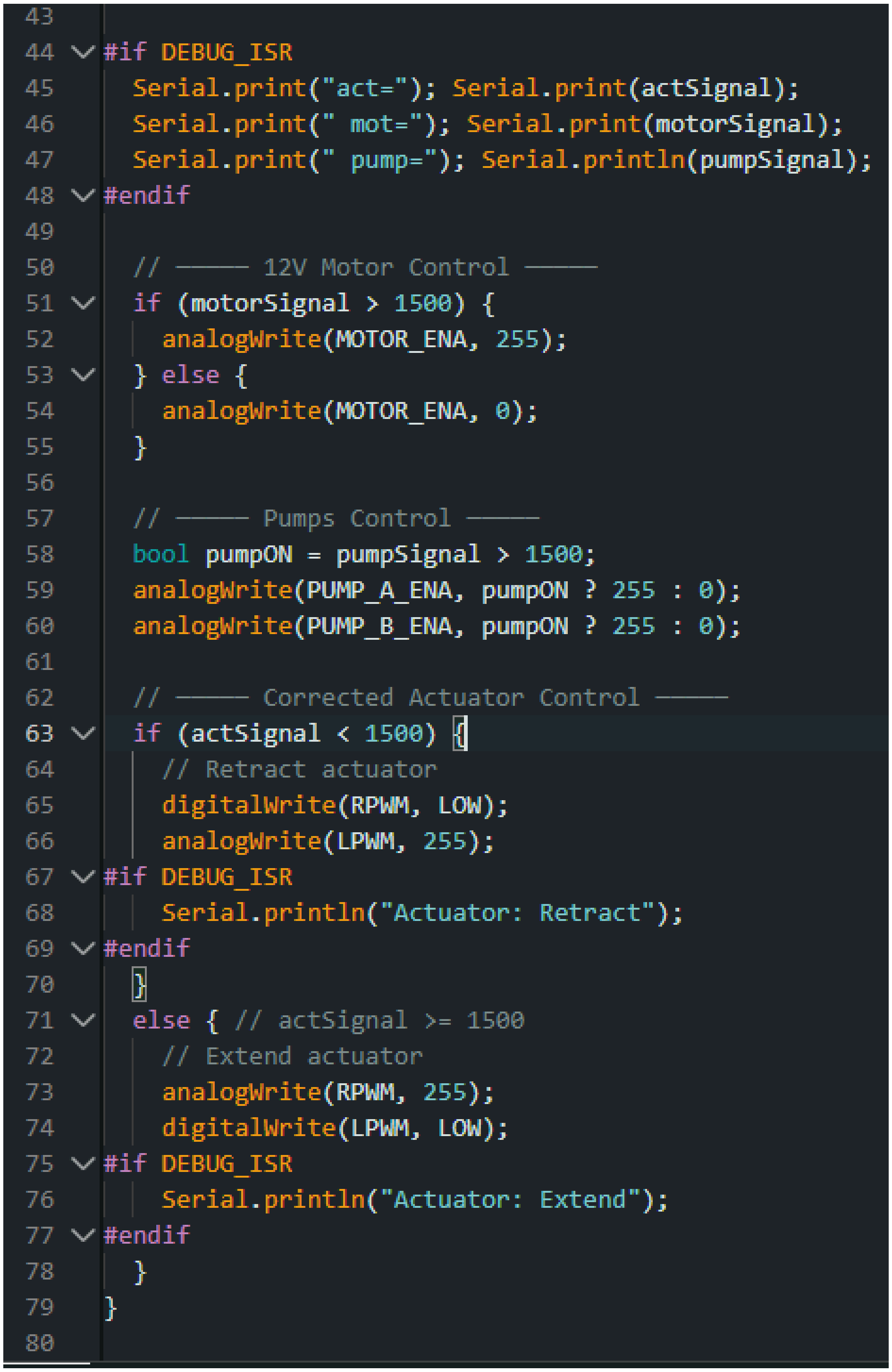


```

void setMotor(int pwm, int in1, int in2, int speed) {
  digitalWrite(in1, speed > 0);
  digitalWrite(in2, speed < 0);
  analogWrite(pwm, abs(speed));
}

void sendToSlave(int actuatorSignal, int motorSignal, int pumpSignal) {
  Wire.beginTransmission(8); // Nano's I2C address

  Wire.write(highByte(actuatorSignal));
  Wire.write(lowByte(actuatorSignal));
  Wire.write(highByte(motorSignal));
  Wire.write(lowByte(motorSignal));
  Wire.write(highByte(pumpSignal));
  Wire.write(lowByte(pumpSignal));

  Wire.endTransmission();
}

void debugOutput(int throttle, int steering, int rightSpeed, int leftSpeed,
                 int pump, int actuator, int motor) {
  Serial.print("Throttle: "); Serial.print(throttle);
  Serial.print(" | Steering: "); Serial.print(steering);
  Serial.print(" | L: "); Serial.print(leftSpeed);
  Serial.print(" | R: "); Serial.print(rightSpeed);
  Serial.print(" | Pump: "); Serial.print(pump);
  Serial.print(" | Act: "); Serial.print(actuator);
  Serial.print(" | Motor: "); Serial.println(motor);
}

```

Appendix 13: Master Arduino code



```
#include <Wire.h>

// ───── Pin Map ─────
#define RPWM 3  // IBT-2 RPWM - actuator extend
#define LPWM 5  // IBT-2 LPWM - actuator retract

#define MOTOR_ENA 9   // 12V Motor PWM
#define PUMP_A_ENA 10
#define PUMP_B_ENA 11

// ───── Debug ─────
#define DEBUG_ISR 1

void setup() {
  Serial.begin(9600);
  Wire.begin(8);  // I2C slave address
  Wire.onReceive(receiveEvent);

  pinMode(RPWM, OUTPUT);
  pinMode(LPWM, OUTPUT);
  pinMode(MOTOR_ENA, OUTPUT);
  pinMode(PUMP_A_ENA, OUTPUT);
  pinMode(PUMP_B_ENA, OUTPUT);

  // Default state: all off
  digitalWrite(RPWM, LOW);
  digitalWrite(LPWM, LOW);
  analogWrite(MOTOR_ENA, 0);
  analogWrite(PUMP_A_ENA, 0);
  analogWrite(PUMP_B_ENA, 0);
}

void loop() {
  // All logic handled in I²C ISR
}

void receiveEvent(int howMany) {
  if (howMany < 6) return; // safety check

  int actSignal   = (Wire.read() << 8) | Wire.read();
  int motorSignal = (Wire.read() << 8) | Wire.read();
  int pumpSignal  = (Wire.read() << 8) | Wire.read();

#if DEBUG_ISR
```



```

#if DEBUG_ISR
  Serial.print("act="); Serial.print(actSignal);
  Serial.print(" mot="); Serial.print(motorSignal);
  Serial.print(" pump="); Serial.println(pumpSignal);
#endif

  // ----- 12V Motor Control -----
  if (motorSignal > 1500) {
    analogWrite(MOTOR_ENA, 255);
  } else {
    analogWrite(MOTOR_ENA, 0);
  }

  // ----- Pumps Control -----
  bool pumpON = pumpSignal > 1500;
  analogWrite(PUMP_A_ENA, pumpON ? 255 : 0);
  analogWrite(PUMP_B_ENA, pumpON ? 255 : 0);

  // ----- Corrected Actuator Control -----
  if (actSignal < 1500) {
    // Retract actuator
    digitalWrite(RPWM, LOW);
    analogWrite(LPWM, 255);
#if DEBUG_ISR
    Serial.println("Actuator: Retract");
#endif
  }
  else { // actSignal >= 1500
    // Extend actuator
    analogWrite(RPWM, 255);
    digitalWrite(LPWM, LOW);
#if DEBUG_ISR
    Serial.println("Actuator: Extend");
#endif
  }
}

```

Appendix 14: Slave Arduino Code