Design Phase Two

Chase Garner, Sawyer Hall, Lexi Sheeler, Daniel Summers
Electrical and Computer Engineering Department
Tennessee Technological University
Cookeville, TN
ncgarner42@tntech.edu, sjhall43@tntech.edu, aasheeler42@tntech.edu, dbsummers42@tntech.edu

I. Introduction

Throughout the development process, a series of experiments, analytical tests, and research has led to a concrete design of the 2022 SoutheastCON Hardware Competition Robot. The purpose of this research was to develop specific artifacts that could be used in the construction of individual subsystems and the overarching system as a whole. This document aims to present the design constraints, research and testing, and justification for the team's design choices. Additionally, deviations and revisions of the design from the conceptual phase to the current system have been explained and justified as well.

II. Chassis

A. Chassis Specifications

The main purpose of the chassis is to house the other subsystems of the robot. Additionally, the combination of the chassis' structure and any external subsystems must conform into a 1 cubic foot form factor at the beginning of the competition. Once a competition run has begun, the robot is allowed to extend beyond this form factor. Another constraint on the chassis is the limited turn radius of the game track. If the chassis' footprint measures one square foot, the robot will not be able to turn around on the narrow portions of the track. The figure below illustrates the necessity of conforming to this specification. The track itself measures 16 inches across, and the robot's width and length are each specified at 12 inches or less. This allows the robot to traverse the track in its initial direction. However, if the robot attempts to turn, its diagonal length of 16.9 inches will exceed the track width. Designing around this constraint involved dividing the chassis into a stationary lower compartment, which is used to house the electronic components (microcontrollers, drivetrain, power supply system), and an upper platform attached to a turntable. This turntable must rotate 180 degrees to allow the robot to fire beads horizontally in either direction perpendicular to the track. Lastly, the chassis was designed to maximize the reach of the robotic arm. In order to do so, a tiered design was incorporated onto the upper platform.

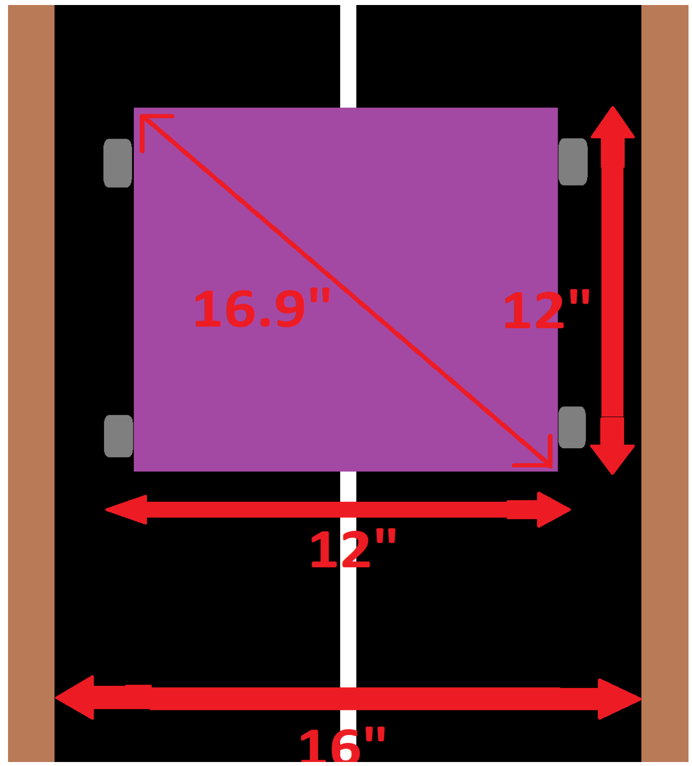


Fig. 1. Chassis and track dimensions

B. Chassis Design

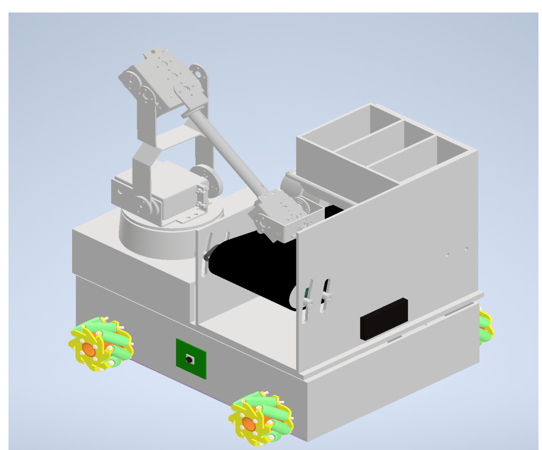


Fig. 2. Chassis design overview

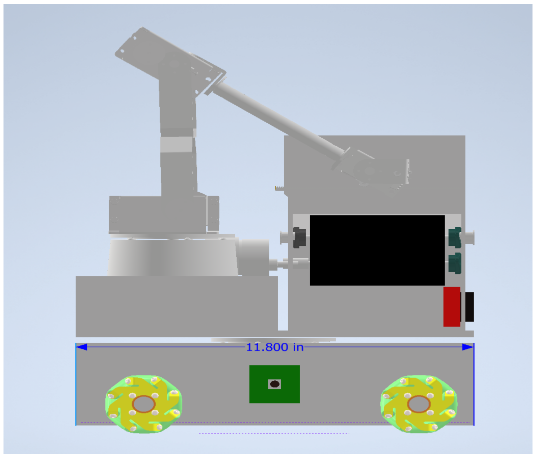


Fig. 3. Chassis length

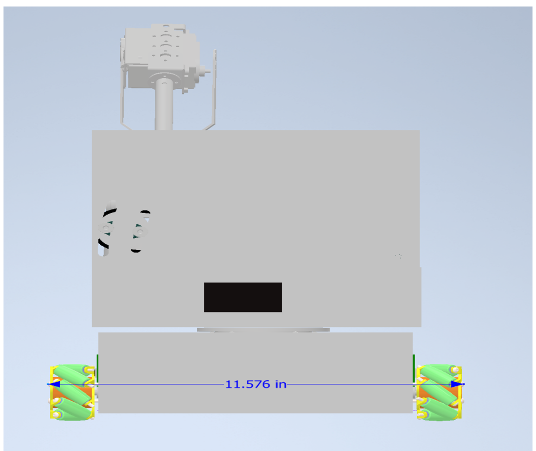


Fig. 4. Chassis width

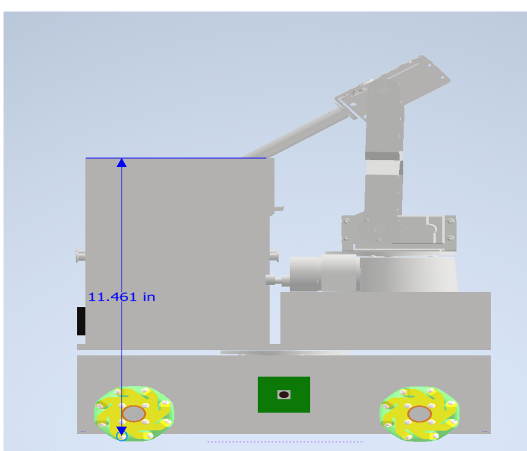


Fig. 5. Chassis and height

The lower compartment of the chassis will measure 8.75 x 11.8 x 3.25 inches and will serve as the housing of the major electronic components including the microcontrollers, drivetrain, and power supply system. This component will be constructed using acrylic sheets. The upper chassis will be constructed as an 9.175 x 11.8 inch acrylic platform that will be affixed to the turntable. The rear half of this platform will be raised an additional 2.15 inches and will serve as a platform for the robotic arm and shooting mechanism motor. A gap in this platform allows the robotic arm to fold into itself before the competition begins. The horizontal, load-bearing platforms of the robot will be constructed from $\frac{1}{4}$ inch thick acrylic sheets, while the remaining acrylic components will be $\frac{1}{8}$ inch thick. This choice was made in order to reduce the flex on the acrylic. The total necessary acrylic is shown in the Figure 6.

	Dimensions	Acrylic (sq. inches)
.25"		
Upper Chassis	9.175 x 11.8	108.265
	6 x 4.920 + 4.25 x 3	42.27
Lower Chassis	9.175 x 11.8	108.265
	Total:	258.8
.125"		
Upper Chassis	2.15 x 6	12.9
	2.15 x 3	6.45
	2.15 x 3	6.45
Lower Chassis	11.8 x 3.25	38.35
	11.8 x 3.25	38.35
	9.175 x 3.25	29.81875
	9.175 x 3.25	29.81875
Shooting Mech. Housing	4.835 x 9.125 + 2.835 x 4.375	56.5225
	5.75 x 9.125	52.46875
	4.835 x 0.849	4.104915
	7.67 x 5.75	44.1025
	Total:	319.336165

Fig. 6. Overview of Chassis surface area. The table is divided by the two different thicknesses of acrylic and in what part of the chassis the material will be placed.

Additionally, the upper chassis will house the shooting mechanism and a 3D printed compartment for sorting and storing the beads. The shooting mechanism housing rests on a 5.75 x 9.25 inch platform. Three vertical walls extend upward to support the belt's rollers and the storage compartment. This compartment measures 5.25 x 3.75 x 2.8 inches and contains a false bottom that will hold the beads in place until a motor slides the false bottom free of the container. A geared tab on one side of the false bottom was included to facilitate this movement. A motor will be mounted to the compartment to drive the false bottom. The compartment will be 3D printed due to the tight tolerance between the compartment and the false bottom as well as to reduce weight. Models of the bead compartment and false bottom can be seen in Figures 7 and 8.

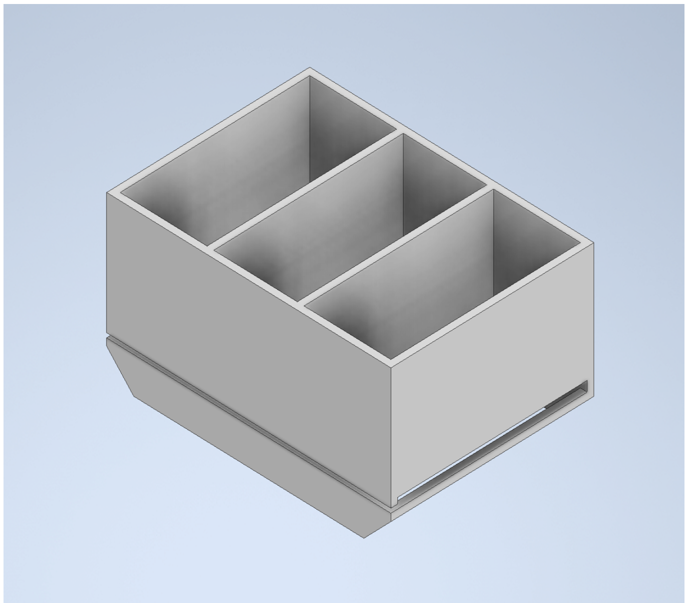


Fig. 7. Bead compartment design.

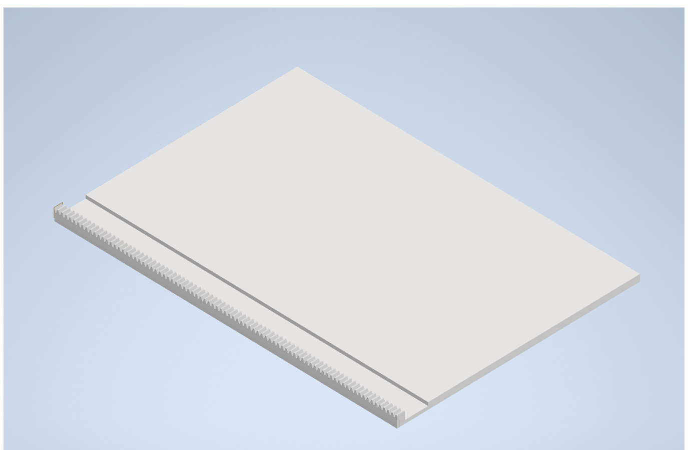


Fig. 8. Bead compartment false bottom design.

The various aspects of the chassis will be laser cut to include finger joints for connecting the acrylic sheets together. Additionally, a 3D Printed mounting bracket (Figure 9) will be included in each corner to connect the pieces. A brass press-fit insert will be inserted into the hole in the acrylic, and a bolt will be threaded through the mounting bracket and into the insert, securing it to the acrylic. The total width of the acrylic and mounting bracket is $\frac{1}{4}$ inch, and the threaded insert is 2-56 sized thread, so $\frac{1}{4}$ inch Long 2-56 thread size screws are being used as the fasteners. For smaller components to be mounted to the chassis, such as the navigation cameras, peripheral display, and motors, screw holes will be drilled into the acrylic frame. This will allow the exact placement of these components to be finalized after the structure itself is built. This decision was made to prevent misplacement and wasting acrylic by laser cutting these smaller mounting holes in the wrong place.

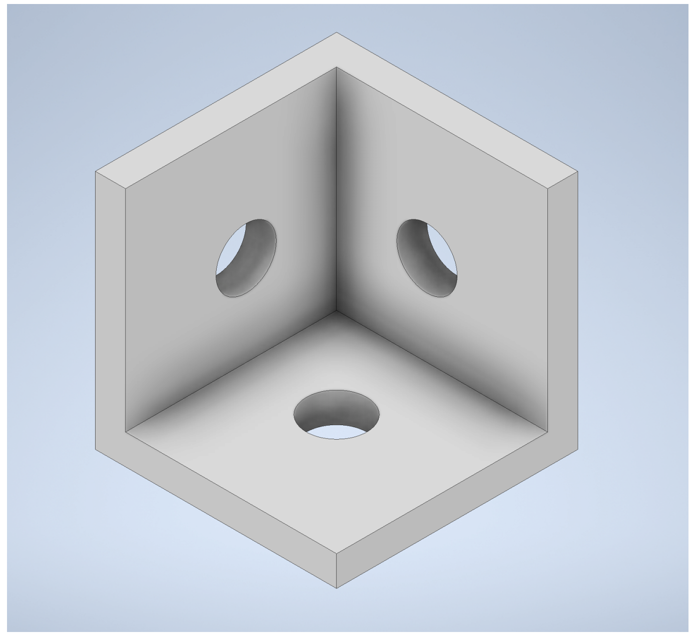


Fig. 9. Chassis mounting bracket.

C. Verification

The most crucial design constraint of the chassis is that it must conform to the 12 cubic inch specification as outlined by the competition. Without meeting this constraint, the robot will be disqualified from the competition, rendering all other design aspects pointless. As evidenced above, this compliance was verified by constructing a 1:1 scale 3D model of the chassis and external subsystems. To assist with the more complex elements, third-party 3D models of the robotic arm and mecanum wheels were imported and used. More detailed models and designs of these subsystems are located in their respective sections of the document. Using the CAD software's measurement tool, the chassis in its current design iteration was measured to be 11.8 x 11.576 x 11.461 inches which complies with the specification. Additionally, the upper chassis begins at a height of 3.25 inches off the ground, which allows it to freely turn above the 2 inch barricades that surround the track.

D. Bill of Materials

Item	Description	Unit Cost	Qty.	Total Cost
1/8" Acrylic Sheeting	12" x 24" Acrylic Sheeting	$16.46	2	$32.92
1/4" Acrylic Sheeting	12" x 24" Acrylic Sheeting	$22.18	2	$44.36
Brass Screw to Expand Insert for Plastic	0.156" Length Brass Insert (Pack of 50)	$12.06	1	$12.06
Screws	¼" Long 2-56 Thread Size Screws (Pack of 100)	$8.32	1	$8.32
			Total	**$97.66**

Fig. 10. Chassis Bill of Materials.

III. Turntable

During the competition, nets will be placed along either side of the track. The robot, as it is designed, will not be able to turn in place. This poses a challenge, as the shooting mechanism is not bidirectional, and the robot will not be able to turn to face the nets in either direction. Early in the design process, it was decided that the robot will utilize a turntable in order to accomplish aiming. The turntable will be responsible for turning the upper chassis (which contains the shooting mechanism) in either direction. The turntable will need to be able to support the weight of the upper chassis; a breakdown of this weight is shown in Figure 11.

Component	Weight (pounds)
Motors x4	1.65
All Controllers/Sensors	0.5
Shooting Mechanism	2
Wheels x4	0.5
Battery	1.5
Robotic Arm	2
Acrylic Chassis	4
Turntable	0.2
Total	12.35

Fig. 11. Breakdown of robot weight.

Being that the turntable only needs to support the weight of the upper chassis, the drivetrain motors, wheels, $\frac{1}{3}$ of the acrylic chassis, and the turntable itself can be subtracted. It is reasonable to assume that the turntable will need to support at least 7 lbs.

The turntable will be need to be able to rotate the entire upper chassis in order to aim the shooting mechanism. It will need to be able to rotate 180 degrees to fire in both directions. Based on emperical evidence collected from testing a similar turntable model made by the same manufacturer, the turntable will easily be able to achieve the desired angle of rotation.

In addition to the turntable, a servo motor is also needed to drive the rotation. Calculating the load of the servo is a matter of calculating the torque required to rotate a rectangle (the chassis volume) around its height axis. This torque is equal to the moment of inertia multiplied by the angular acceleration. The equation for moment of inertia is shown below in Equation 1:

$$I_h = 112 * 3.2 * (0.32 + 0.232)$$

This gives the moment of inertia at approximately 0.0381 kg*m^2. The desired acceleration of the servo is 2π radians/s^2. Thus, multiplying these values gives a resultant torque of 0.2394 N*m. Converting to kg*cm gives 2.44 kg*cm, which is the minimum torque requirement of the motor.

Using the documentation of the turntable, a servo motor with the appropriate torque output was selected. This servo requires 4.8-6 V. The design already includes step down circuitry that outputs 5 V, so the voltage requirement is met. The servo will be controlled using the Adafruit servo shield that was purchased for the robot arm. The servo shield will allow for control of the servo from the microcontroller via PWM. A wiring diagram is shown in Figure 12.

Figure 13 shows the CAD drawing of the turntable as well as the physical layout of the robot. The turntable's small form factor means that a custom base was designed, which will be 3D printed, to mount it in order to achieve the correct height.

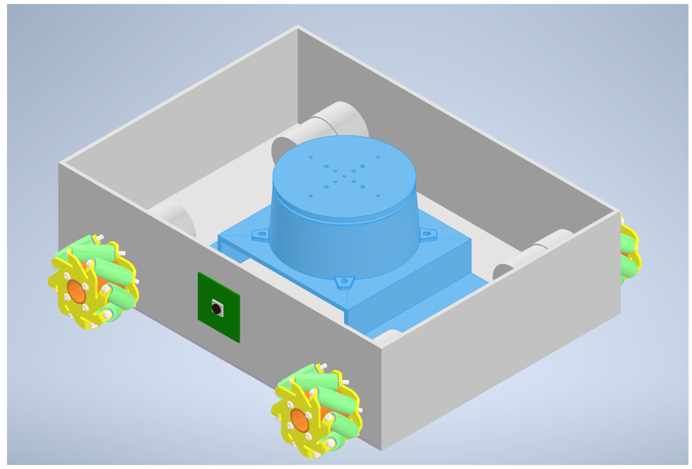


Fig. 13. Three-Dimensional model of turntable.

A. Bill of Materials

Item	Description	Unit Cost	Quantity	Total Cost
Hitec HS-645MG Servo Motor	Turntable Motor	$34.99	1	$34.99
Lynxmotion Base Rotate Kit	Turntable	$19.95	1	$19.95
#4 x 3/8" Phillips Head Self Tapping Screws	Mounting Screws (Pack of 25)	$7.58	1	$7.58
			Total	$62.52

Fig. 14. Turntable Bill of Materials.

IV. Shooting Mechanism

The shooting mechanism is a vital aspect of the overall design, being that it is the primary way that the team will be gaining points. It will need to be able to aerially launch beads a horizontal distance of between 5 and 6 inches and a vertical distance of between 6 and 13.5 inches. It is desirable that the beads reach their "apex" height near the middle of the diameter of the net. To accomplish this objective, the team has designed a conveyor system that will be able to launch the beads at the required distance and height.

To better understand the trajectory of the beads, a MATLAB program was created to plot the flight path of the beads with different parameters. The parameters that were allowed to be edited were the launch angle in degrees, the distance from the end of the conveyor to the net in inches, and the RPM of the motor that is driving the conveyor. Changing these three parameters and observing how they affected the flight path of the beads allowed the team members to gain a better understanding of how the conveyor should be designed. An example of the output of the program is shown in Figure 15.

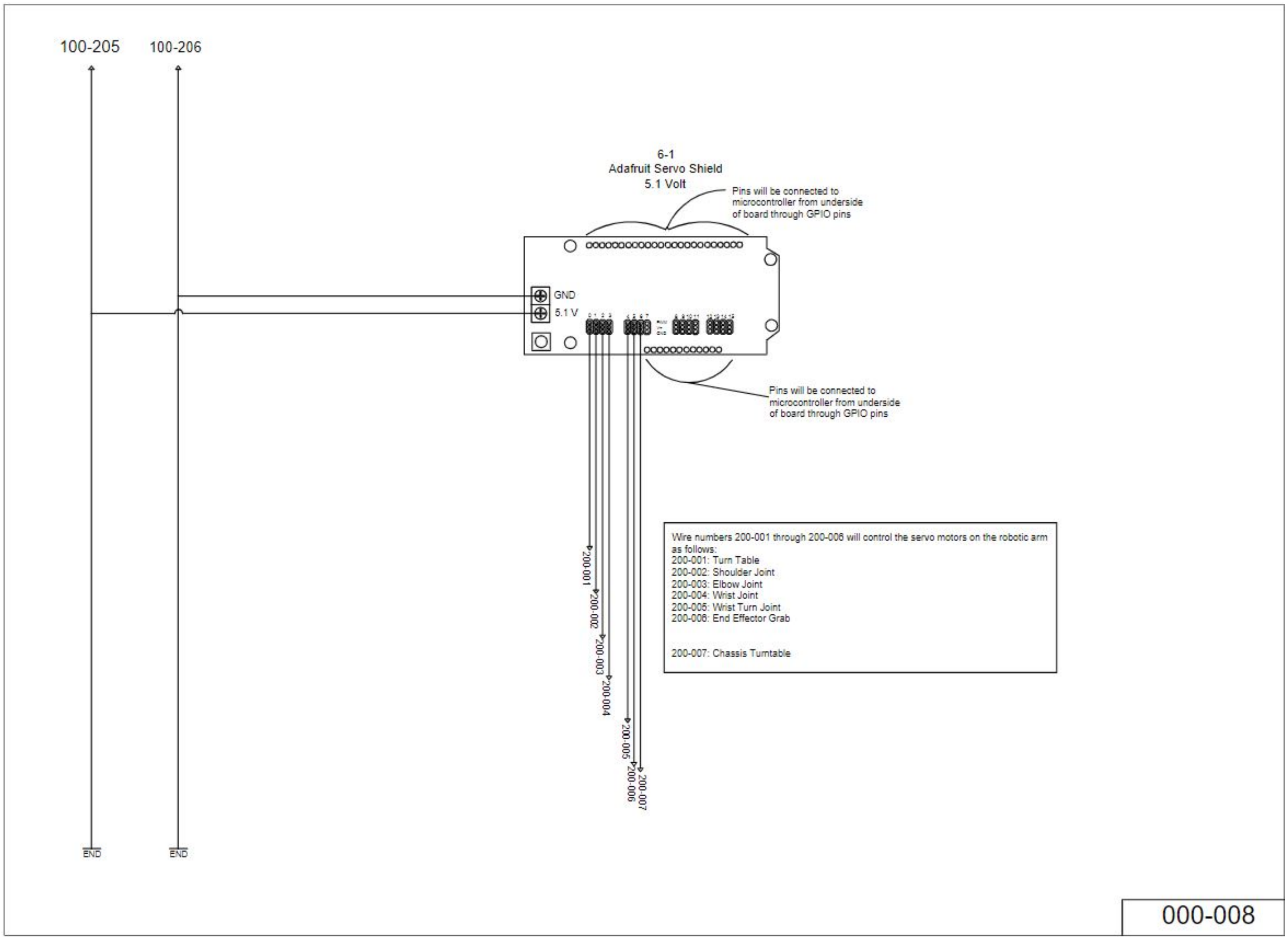


Fig. 12. Turntable wiring diagram.

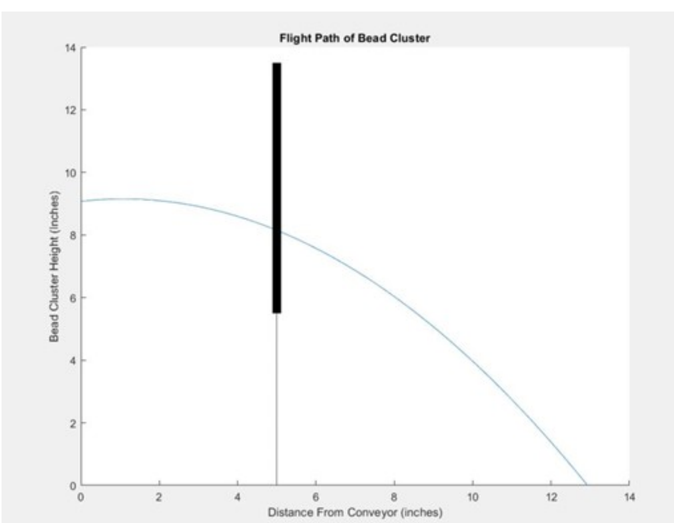


Fig. 15. Projected bead trajectory.

Figure 15 is using 8 degrees for the launch angle, 700 RPM for the motor speed, and 5 inches for the distance to the net. It is clear to see that with these parameters, it is achievable to launch the beads into the net. The limiting factor in the actual design of the conveyor will be the speed of the motor.

In order to choose an appropriate motor, the potential torque load that the motor will experience needs to be approximated. There are a few different factors that affect the overall load, including the mass of the beads and the friction of the ball bearings that the conveyor is mounted to. It is not expected that the conveyor belt itself will contribute much to the load due to the fact that it is both pushing and pulling the rollers at the same time, effectively cancelling out this force. In addition, the friction of the ball bearings is expected to be negligible, as they are specifically designed to be able to rotate with minimal friction. It is anticipated that the main contributor to the load will be the mass of the bead bracelets. Being that there is no official bead bracelet being used, the mass has been estimated at around 0.822 grams per bracelet. This number was based on similar products from Amazon. The maximum number of bracelets in the competition will be 30, giving a potential total mass of 24.66 grams. The approximate length of the conveyor will be 22.86 centimeters. Using these values to approximate the load returns a value of 0.5637 kg-cm. The selected motor has a rated torque of 0.66 kg-cm, which exceeds the estimated required torque for the belt. In addition, it has a rated speed of 850 RPM, which has been proven using the MATLAB simulation to be fast enough to launch the beads into the nets. It will be coupled to the rotary shaft via a 4mm to 5mm set screw motor couple, and mounted to the chassis via a custom 3D printed bracket. The motor will be controlled through an H-bridge via PWM. This method of control was chosen because it gives the team the ability to change the speed of the motor if necessary. The team already possesses an extra H-bridge that can be used for this purpose. A circuit diagram of the motor and H-bridge is shown in Figure 16.

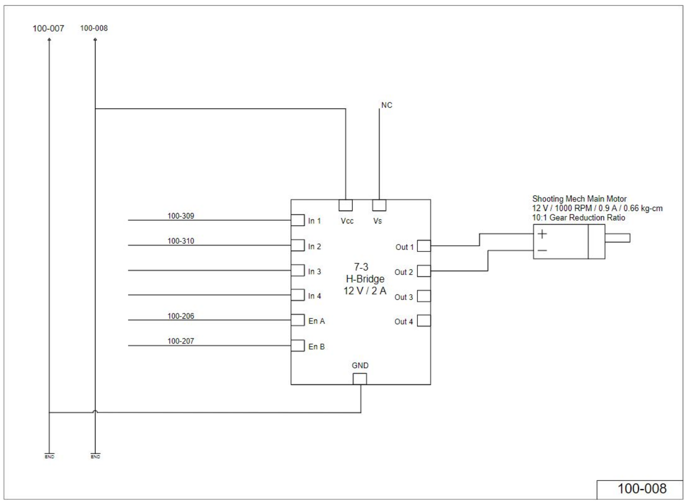


Fig. 16. Shooting mechanism conveyor circuit with th H-bridge.

The rollers will be 3D printed, which will allow the team to control bore precision. The surface of the 3D printed material also provides a good amount of friction to turn the belt. The diameter will be 1.5 inches with a bore size of $\frac{3}{16}$ inches to match that of the bearings and axles. The rollers will need to be attached via ball bearings to the supporting structure. To attach the rollers to the surrounding support structure, pillow block bearings will be utilized. The pillow block bearings have outward-facing mounting holes. These holes will allow for mounting as well as adjusting the angle of the conveyor. Brass wells will be inserted into slots in the acrylic in order to hold the bearings in place. Additional nuts and screws will be needed to mount the wells to the bearings. The rotary axle will pass completely through the roller bore and be mounted on each side to the bearings/motor. It will have a $\frac{3}{16}$ inch diameter to match that of the bearing and roller. The belt material thickness was chosen to be $\frac{1}{8}$ inch to keep the conveyor height at a minimum. Rubber has been chosen for the belt material because of its tackiness. The team also added room in the conveyor design to attach 1-inch protrusions on the belt to prevent slip if it is an issue. The belt will be held to the rollers by its own tension. Finally, the supports of the conveyor will be made of acrylic, the same material as the rest of the chassis.

The shooting mechanism will also need a method to reload, as there will be three separate nets. As each new net is found, the next set of beads to be fired off will need to be dropped onto the belt. The team has designed a reloading mechanism that will be fixed atop the shooting mechanism. This mechanism will contain three compartments with a slab on the bottom that is able to be moved, and in turn allowing each set of beads to be dropped. The slab will contain a toothed rack on its side and will be moved by a motor attached to a gear. Because the slab will be 3D-printed, it is expected that the torque required to move it will be miniscule. A small servo motor will be used in this application to have precise control over the slab movement. This servo will be controlled through the Adafruit servo shield via PWM. A custom gear will be 3D printed and attached to the servo motor shaft. The wiring diagram of the Adafruit servo shield is shown in Figure 17.

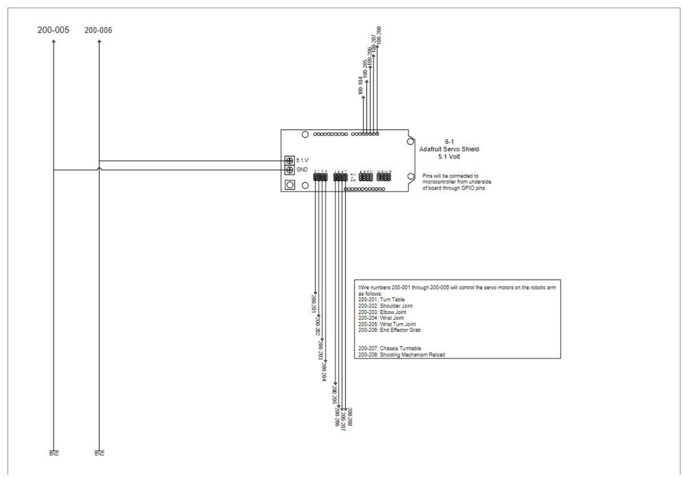


Fig. 17. Shooting mechanism servo shield wiring diagram.

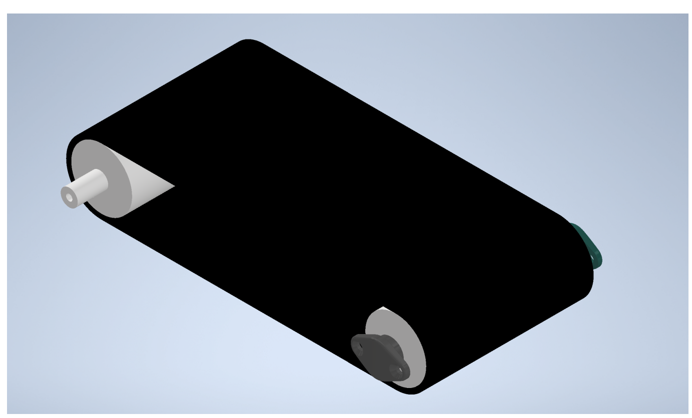


Fig. 18. Shooting mechanism conveyor model

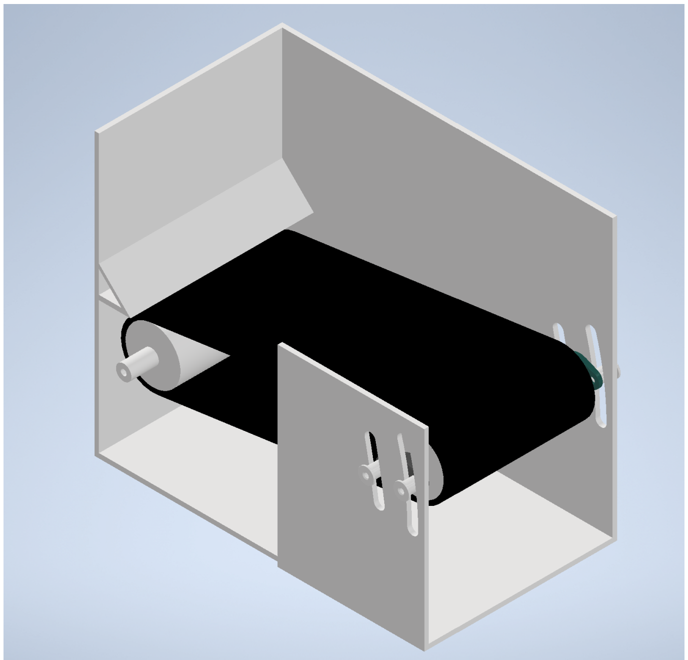


Fig. 19. Detailed Shooting Mechanism Model

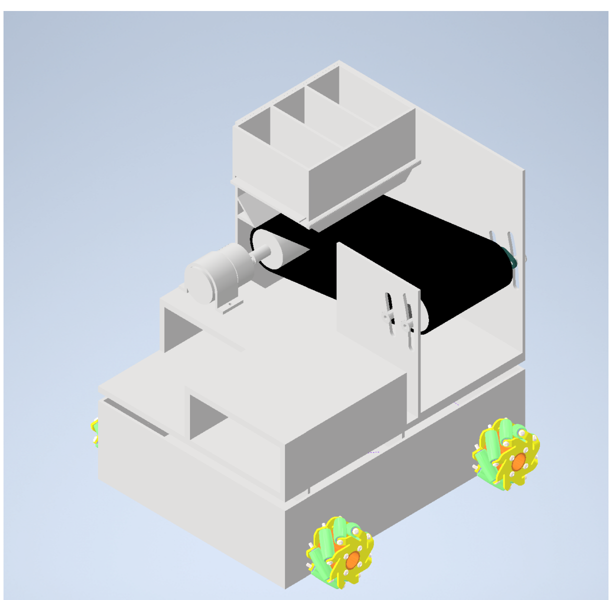


Fig. 20. Detailed Shooting Mechanism Model

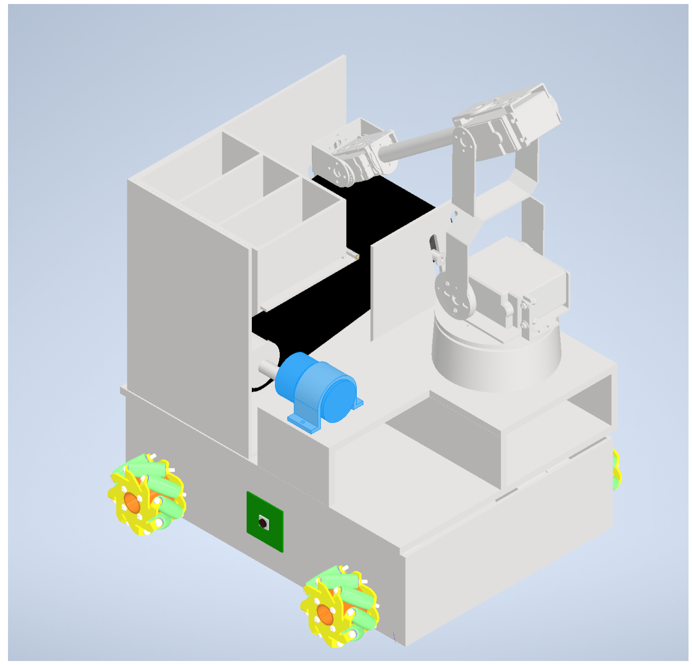


Fig. 21. Shooting mechanism driving motor with mounting bracket.

Note: The robot arm has been removed from this picture to enhance visuals.

A. Bill of Materials

The total BOM is shown in Figure **??**.

Item	Description	Unit Cost	Qty	Total Cost
Conveyor Motor	Pololu 12 V 1000 RPM DC Motor	$24.95	1	$24.95
Pillow Block Bearing	Pillow Block Bearing 3/16" Bore	$9.56	3	$28.68
Bearing Rotary Shaft	Rotary Shaft 3/16" diameter 1ft long	$8.02	2	$16.04
Motor Coupling	4mm to 5mm Motor Coupling	$4.99	1	$4.99
Conveyor Belt Material	1/8" 5 ft long PVC 120	$20.98	1	$20.98
Reloading Motor	Parallax (Futaba) Continuous Rotation Servo	$19.95	1	$19.95
			Total:	**$115.59**

Fig. 22. Shooting mechanism Bill of Materials.

V. Main Controller Specifications

The biggest specification that determined the design of the main controller was the large number of GPIO pins required. Each motor driving circuit for the drivetrain requires six pins from the main controller, and two motor driving circuits will be onboard for a total of twelve GPIO pins for the drivetrain alone. Additionally, the line following sensors, which were originally conceptualized to be connected to the navigation system, have been moved to directly connect with the main controller. The purpose of this design change was to reduce the feedback delay from the line following array to the control algorithm. However, this change increased the number of GPIO pins required of the main controller. Due to this high pin requirement, an Arduino Mega 2560 was selected for its 54 GPIO pins, 15 of which can be used to generate Pulse Width Modulation signals.

Figure 23 along with Figure 57 demonstrates the connnections between the main controller and its subsequent systems. To accomplish the required servo controls, an Adafruit servo shield will be connected to the main controller. This shield generates its own PWM signals based on commands sent to it from the main controller via an I2C connection. This greatly reduces the necessary number of connections made directly to the Arduino. The Adafruit shield also contains header pins that are directly connected to the corresponding Arduino pins below, allowing for those pins to still be accessed although physcally blocked by the shield. More thorough circuit schematics of this and other subsystems are included in Appendix B.

A. Verification

The following table in Figure 24 outlines the subsystems that will utilize the main controller's GPIO pins. The subsystem, total number of pins, and number of PWM pins are each listed. In order for this microcontroller to comply with our design, the total number of pins and number of PWM pins must not exceed 54 and 15, respectively.

Subsystem/Subroutine	Description	GPIO Pins Required	PWM Pins Required
Drivetrain	2 Motor Control Circuits	12	4
Line Following Algorithm	3 Light Arrays	24	0
Robotic Arm	Servo Control	2	0
Shooting Mechanism	Motor Control	4	2
	Totals	**42**	**6**

Fig. 24. Main controller GPIO Pins.

B. Bill of Materials

Item	Description	Unit Cost	Qty.	Total Cost
Arduino Mega 2560	Microcontroller	$34.25	1	$34.25
			Total	**$34.25**

Fig. 25. Main controller Bill of Materials.

VI. Power

The needed power specifications are defined by the data sheets of the selected components of each subsystem. Due to the nature of the project and its many electrical components, providing stable power that is free of noise is important to ensure proper functionality of all components. Power will be supplied to most of the components through secondary devices such as a servo shield or H-bridges.

The first step in defining the specifications of a power system is to determine the required voltage and current draw of each subsystem. Each value was obtained from the components' datasheet. The main controller, an Arduino Mega 2560, has a needed input voltage of 7 Volts. The navigation system will be running from a Raspberry Pi 4 Model B. The navigation sensors will be connected to this board, which will serve as their power supply. The Raspberry Pi requires an input of 5 Volts. The drivetrain will operate through H-Bridges that will drive the four motors. The four motors have an operating voltage of 12 Volts which will be supplied to the H-Bridge thus powering the motors. The robotic arm will function in a similar way that the motors do; however, the servo motors controlling the robotic arm will be driven and powered by the Adafruit servo shield. The servo shield requires an input voltage ranging from 2.3-5.5 Volts. Finally, the turntable servo motor will be run by a similar servo motor to the turn table. For this reason, the turntable can be run through the servo shield. The shooting mechanism will be run by a servo motor operating at 5.1 Volts. A maximum voltage of 12 Volts is required. These values are can be seen in Figure 26.

Component	Voltage (V)	Amperage (A)
Raspberry Pi	5.1	3
Motors x4	12	2.4 (0.6)
Adafruit Shield	5.1	0.25
Arduino MEGA	7	0.5
Arduino UNO	7	0.3
Turntable Servo	5.1	1
Shooting Mech Servo	5.1	1

Fig. 26. Voltage and Amperage values per data sheets.

Finding the needed amperage is not as simple as the voltage. Finding the amperage requires utilizing Figure 26 and the MATLAB code found in Appendix A. The MATLAB code outputs a total current of 4.8287 Amps, which means that everything running at once will require a 12 Volt battery to output at least 4.8287 Amps. For this reason, the team selected the LiFePO4 12V 6.6Ah battery. Using this battery required the team to find how long one battery charge will last, the C value. 6.6Ah means that the battery will give one hour of 6.6 Amps, giving a C value of 1. Using this ratio, the team found that the current ratio (referenced in Appendix A) gives a value of 1.3932. Changing this number to minutes gives the team 82.0096 minutes or 27.337 3-minute competition iterations. The calculation is also assuming that every component will be in operation for every one of the 82.0096 minutes; however, this is not the expected use. Because of this, the battery is expected to last longer than the calculated value.

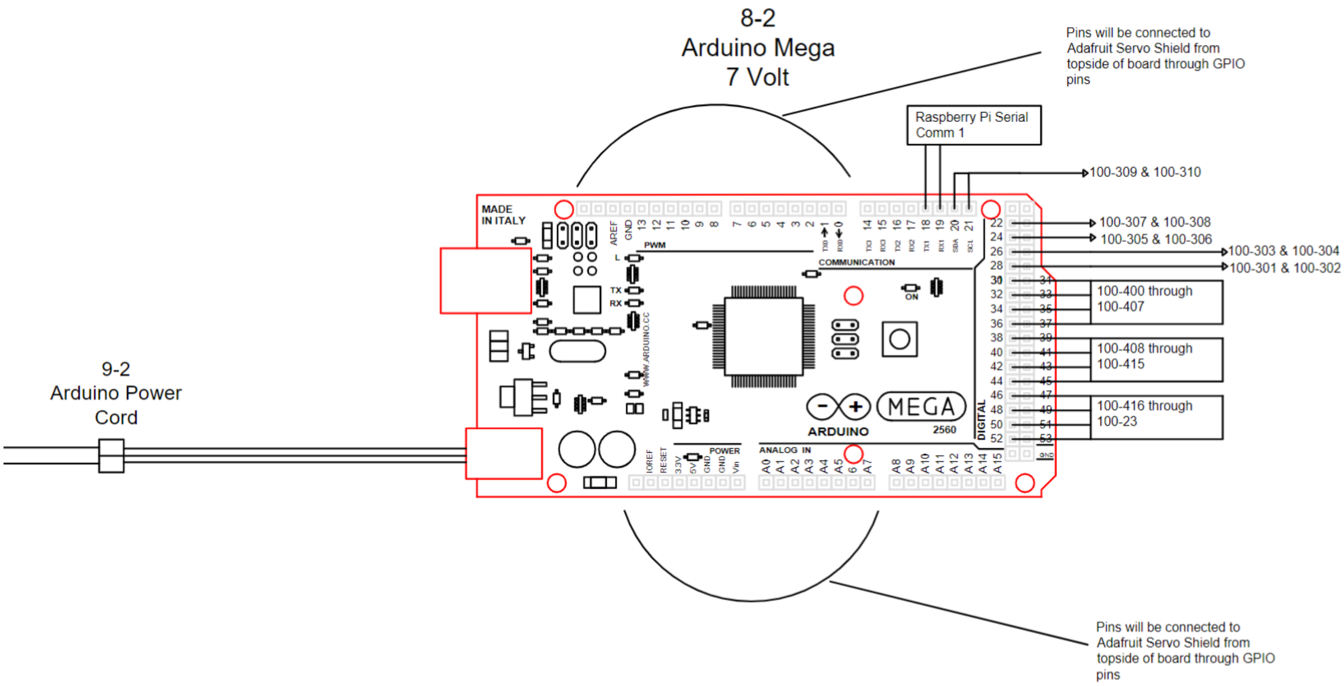


Fig. 23. Main controller pinout.

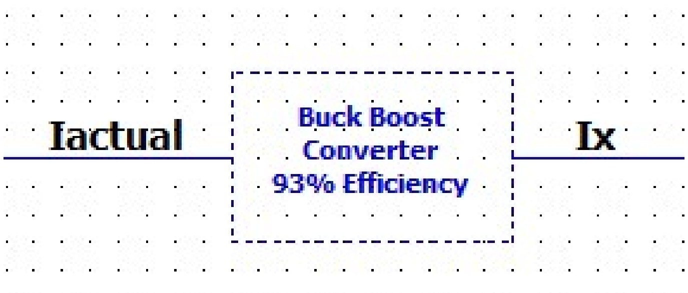


Fig. 27. Example Circuit Modeled by MATLAB Code

The 6.6Ah surpasses the maximum possible current draw (4.8287 A) of all the components running; however, a problem arises due to the battery not being able to give a constant 12 Volts. For this reason, the team will utilize a buck-boost converter. In theory, a buck-boost converter can take a ranged input voltage and transform the voltage to a certain output voltage. The converter can take an input voltage from 9-36 Volts and transform the voltage to 12 Volts with 10 Amps.

One concern is the fact that noise may come off the converters. The data sheet verifies that the noise from the component will be no bigger than 115-200mV$_{\text{p-p}}$. For this reason, the team configured a π-type LC filter that resembles the filter found in the data sheet of the converter. The team simulated the potential noise to verify the design of the filter. The LTSpice Simulation schematic and graph can be seen in Figures 29 and 30. Using this LC-filter will eliminate the noise created by the boost converter. The circuit itself will be built utilizing the components listed and a solderless breadboard.

Another concern was the initial and final switching on and off with the L1 inductor shown in Figure 29. The inductor itself takes time to dissipate current. A sudden change of voltage, like a switch turning on and off, would cause a massive voltage spike within the circuit. The Schottky diode prevents this massive spike of voltage in addition to preventing a current arc across the switch. The diode will continually feed the circuit so that the voltage and current decay to zero.

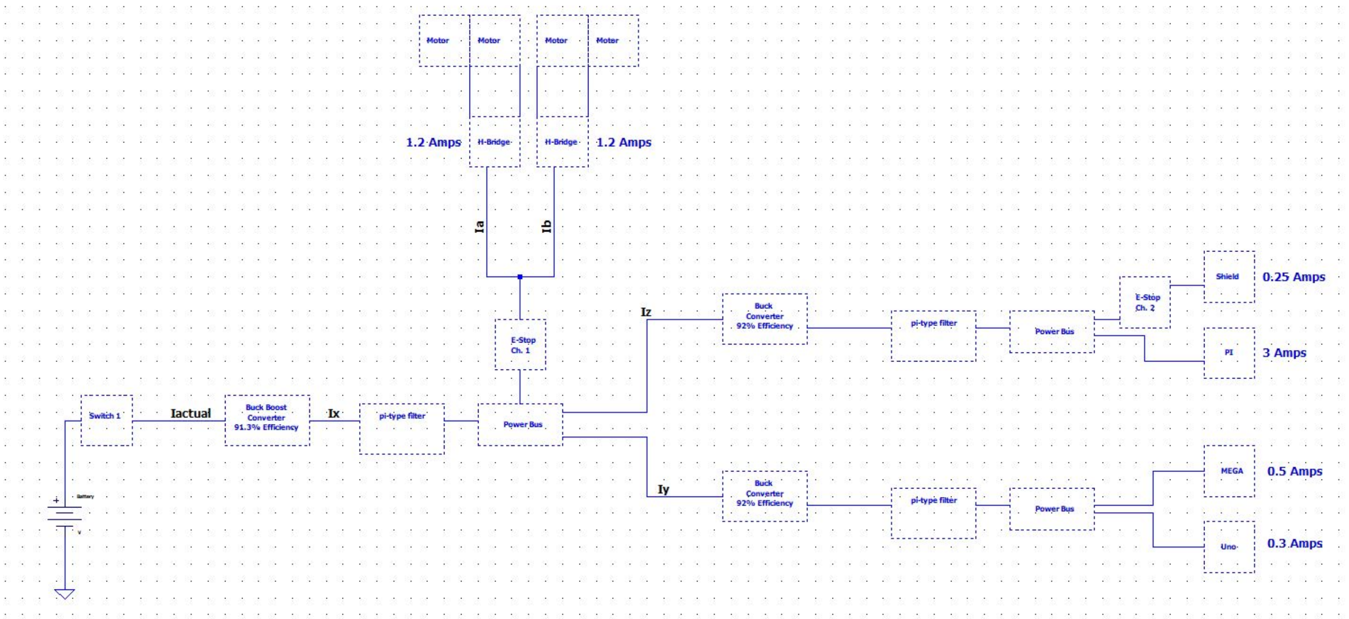


Fig. 28. Full power system schematic.

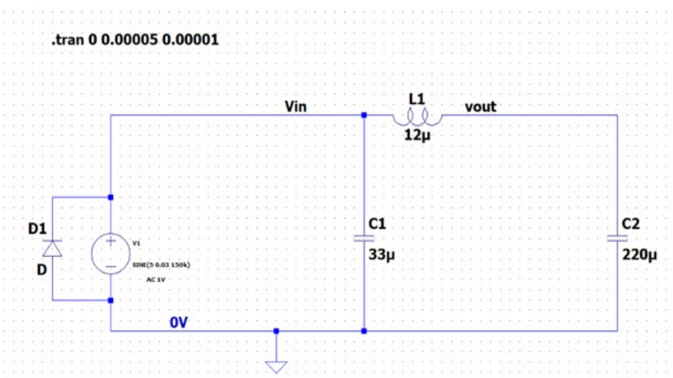


Fig. 29. Circuit for noise simulation.

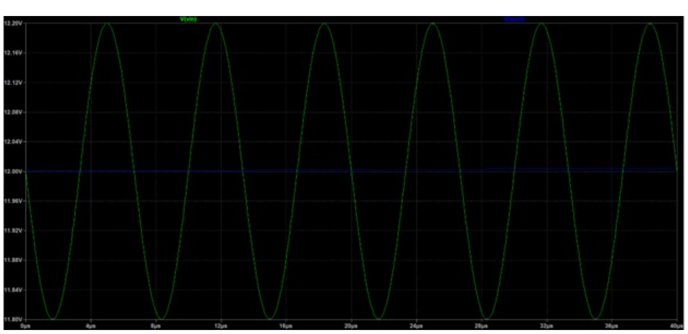


Fig. 30. LTSpice Schemtatic 1 noise simulation graph.

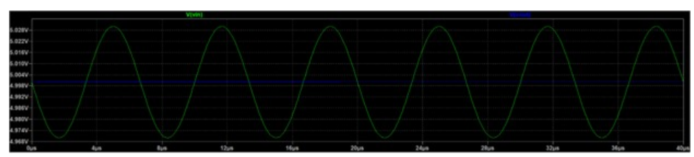


Fig. 31. LTSpice Schematic 2 noise simulation graph.

By implementing this filter, the battery and converter will be able to safely supply the components with the needed input voltage. Figure 26 shows that multiple components need the same input voltage to operate. For instance, one set of components needs 12 Volts, while other components need 7 and 5 Volts, respectively. This brings new constraints to the power circuit. For one, the initial voltage needs to be converted into two seperate lower voltages, and said voltages need to go to multiple components. This will be solved through the use of a buck converter.

The data sheet verifies that the buck converter can take an input range of 3-40 Volts and transform it into a set range of 1.5-35 Volts. The two ranges fit the needed specifications set by the components described before. The team will utilize the regulated output voltage knob to set the needed transformation. Like before, the team will utilize the same π-type LC filter to ensure that the output voltage is a noiseless 7 and 5 Volts. The V_{in} and V_{out} graph can be seen below in Figure 31. Along with transforming the voltage, the second constraint will be met by utilizing power buses. A power bus can take the input voltage and distribute that voltage to other components. Figure 32 depicts the circuit diagram for the selected power bus which can operate at up to 24 Volts.

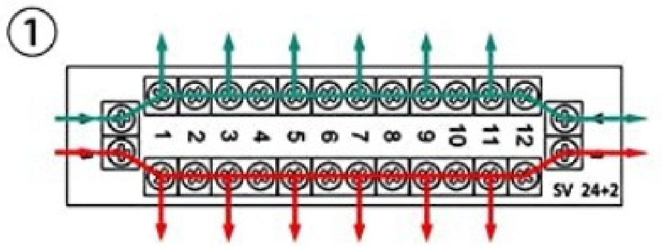


Fig. 32. Power Bus circuit diagram.

The team needs a combination of the two previous ideas so that the power circuit will be able to supply the needed voltages to each component. A combination of all the components previously described will build the power circuit. The connections described among all the components can be seen in the attached CAD document in Appendix B.

There is also concern about the DC motors feeding noise back into the circuit past the H-bridges. To counteract this, three 1 μF capacitors will be placed across the leads of the DC motors as recommended by the manufacturer. One capacitor between the positive and negative terminals, one capacitor from positive terminal to the casing of the motor, and one capacitor from the negative terminal to the casing of the motor, illustrated below.

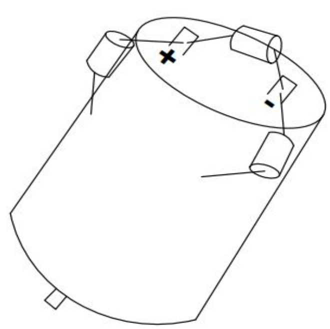


Fig. 33. DC Motor Filtering Capacitors

A. Power Switch and Emergency Stop

The competition rules state that the robot must have a clear on/off switch, and strongly recommend adding an e-stop into the design. For this reason, the team has picked out a switch and an emergency stop that will meet these requirements. Two switches will be utilized in the circuit. One switch will be the interim between the battery and the boost converter while the second will be an input that the microcontroller will watch for to start the competition. The data sheet clarifies that the switch can withstand 10 Amps which is above what the team needs.

The emergency stop's data sheet justifies that the channels can withstand 10 Amps, which is well above the team's currents within the circuit. The emergency stop was picked due to the two-channel capability. The team will utilize this feature so that the 5 Volts going into the Adafruit servo shield and the 12 Volts going into the H-Bridges will have to clear the emergency stop before powering the robot to move. Likewise, pushing the emergency stop will stop the movement from the robotic arm and dc motors of the wheels, but leave the microcontrollers still running. Doing this, will let the robot handler stop the robot from causing more damage if havoc ensues. Both the switch and E-stop connections can be seen in Figure 34 below. A more zoomed in depiction can be seen in Figures 35 - 37.

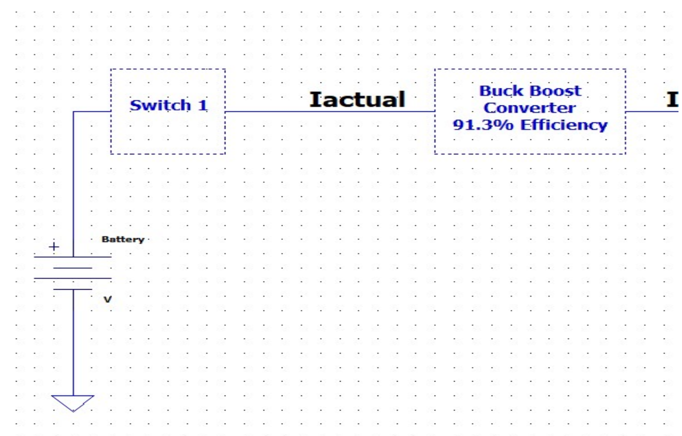


Fig. 35. Switch 1 Placement

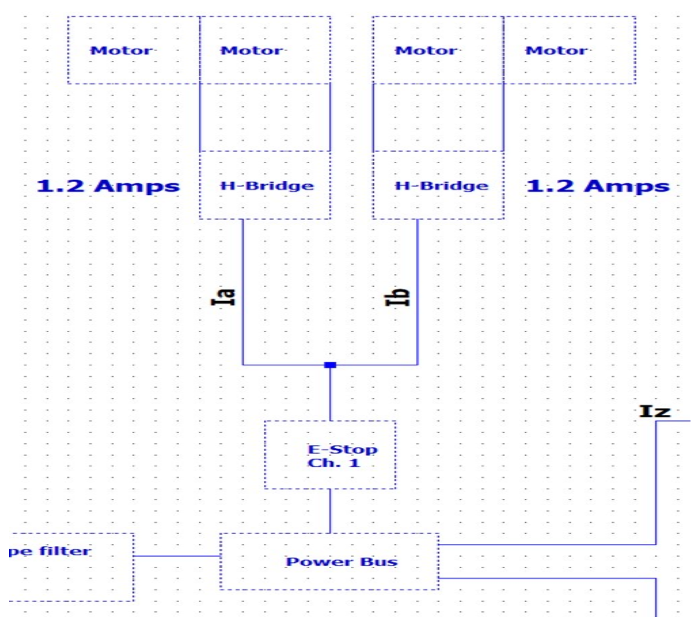


Fig. 36. E-Stop Placement 1

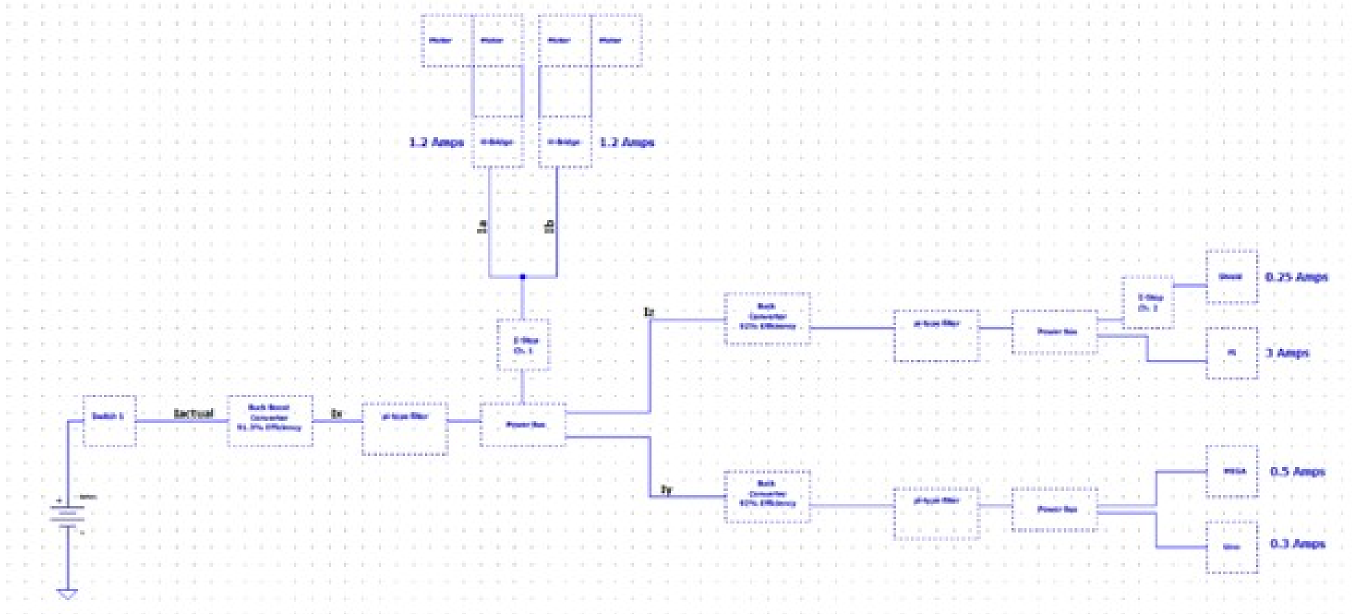


Fig. 34. Switch Schematic Overview

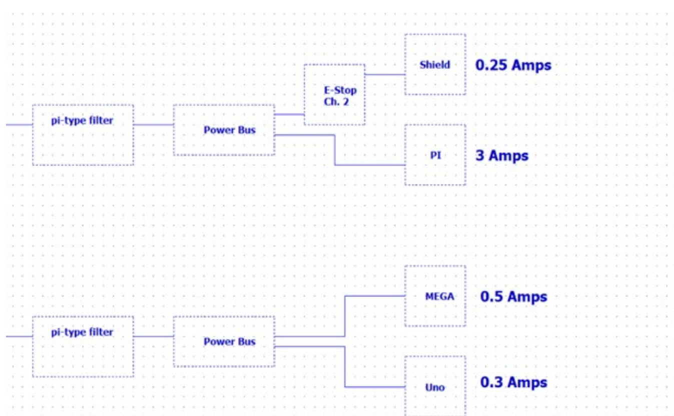


Fig. 37. E-Stop Placement 2

B. Bill of Materials

Item	Description	Unit Cost	Qty.	Total Cost
LiFePO4 Battery	12.8V 6.6Ah Battery	120	2	240
12.8V LiFePO4 Battery Charger	Smart Charger	35.95	1	35.95
12 Position Power Distribution Board	Power Bus	9.49	3	28.47
Boost Converter	DC to DC Converter	55.99	1	55.99
Buck Converter	DC to DC Converter	15.99	1	15.99
Wall Adapter	AC to DC Wall Adapter	16.99	1	16.99
Power Cable Adapters	18 AWG DC Power Cable	5.99	1	5.99
Solder-able Breadboard	Gold Plated Finish Proto Board PCB	10.28	1	10.28
33uF Capacitor	Aluminum – Polymer Capacitor	1.79	6	10.74
220uF Capacitor	Aluminum – Polymer Capacitor	3.29	6	19.74
12uH Inductor	Fixed Inductor	1.36	6	8.16
Schottky Diode	Schottky Diode	1.19	3	3.57
0.1uF Capacitor 6 pack	Ceramic Capacitor	0.99	3	2.97
Switch	1 Channel Switch	14.85	2	29.70
Emergency Stop	2-Channel Emergency Stop	12.99	1	12.99
			Total	497.53

Fig. 38. Power system Bill of Materials.

VII. Navigation

The navigation system has undergone several changes since the initial design phase, now consisting of a Raspberry Pi 4, two cameras, and four IR transmitter/receiver pairs. A Raspberry Pi, Figure 39, allows for the use of Robot Operating System (ROS), which is optimized for robotics, and computer vision techniques, as well as having a large amount of RAM. The Pi will run Ubuntu 20.04 as the main operating system and use the ROS Noetic distribution to handle communication between the main and navigation controllers. Communication will take place via serial pins using the rosserial library. ROS Noetic is the most recent distribution and has long-term support from the developers.

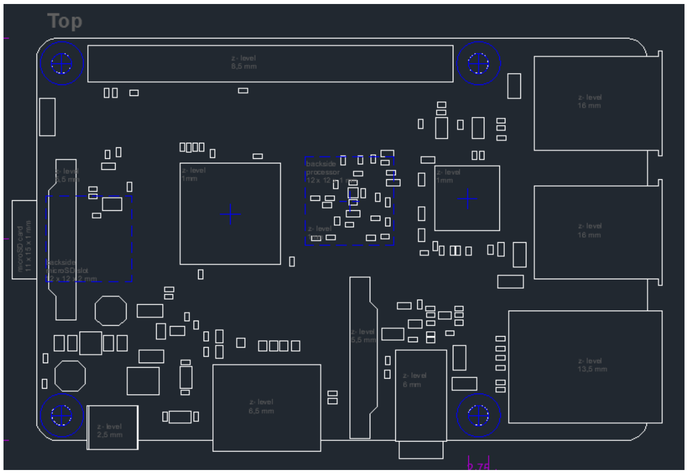


Fig. 39. Raspberry Pi 4 schematic.

This system is vital to performance in the competition. It is vital to identifying the beads and nets, which is the main way the team will be earning points in the competition. The system is designed to recognize the nets, with the camera, and finely adjust, using the sensors and camera, the robot's position on the track to align the shooting mechanism.

A. Sensors

Instead of having two IR arrays per side of the robot, there will now only be two sensors per side, both at the same height, tentatively aligned under the shooting mechanism. This allows for fine tuning adjustments before shooting the beads. The selected sensor, Sharp GP2Y0A21YK0F IR Distance Sensor has a range of 4-32 inches. Based on provided images of the competition track setup, the nets appear to be less than 18 inches from the track. Therefore, this is expected to be enough range to reach the nets and provide adequate distance readings. If competition specifications are changed, the sensor will be revisited to determine if it still satisfies the required functionality. This sensor includes both the transmitter and receiver, which will give more accurate readings without exhaustive testing. This sensor requires 3 GPIO pins: output, power, and ground. The Pi has enough pins to support 4 sensors (12 GPIO pins), but, if necessary, measures can be taken to consolidate the power and ground wires.

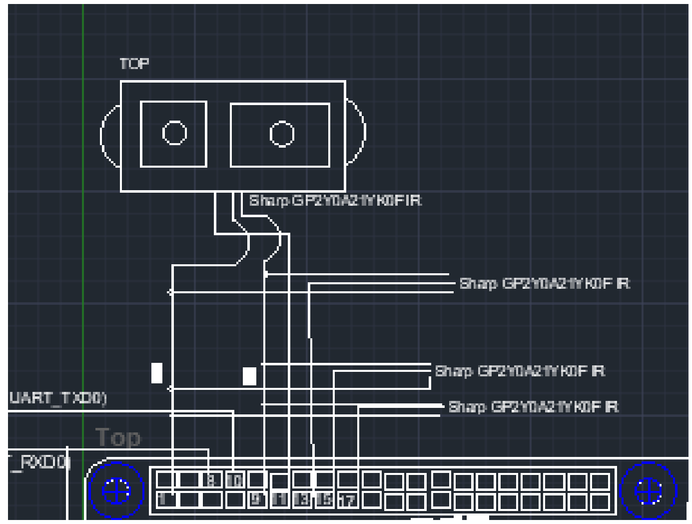


Fig. 40. Sharp GP2Y0A21YK0F IR Distance Sensor schematic.

A new component to the system will be the Arducam USB Camera V2 Module. Initially, the system was designed to use two arrays of sensors along the left and right sides of the lower and upper chassis. As more research was completed, it was determined that this idea was not going to give accurate information for positioning, as well as major difficulty differentiating between the nets and power poles. After looking into several ideas, the design now incorporates computer vision as the primary source for net identification. This specific model was chosen for several reasons: small size, adequate focus range and resolution, field of view (FOV), and USB operation. Given the size constraints of the robot, it was necessary to find a camera that took up a minimal amount of space. This camera is only 38mmx38mm, in addition to the weight being negligible. It has a resolution of 8MP 3264H x 2448V, and per the device specification sheet, the focus ranges from 200 mm to infinity. As the nets will be more than 200 mm and less than three feet from the track, this focal length is sufficient. The camera has a FOV of 62.2 degrees horizontal and 48.8 degrees vertical. The FOV should be adequate as the camera is intended to be directly facing the nets. The camera is also going to be operated via the on-board USB ports. This was done because the Pi has only one connection for a camera, and would require additional hardware to include two cameras. There are four on-board USB ports, which leaves two remaining for any additional hardware that may be necessary in the future. The cameras will be mounted on the left and right sides of the lower chassis, as shown in 42.

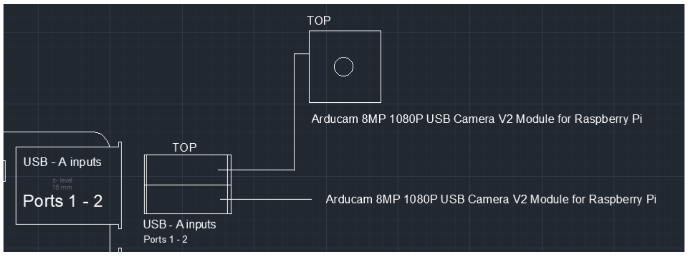


Fig. 41. Arducam USB Camera V2 Module schematic.

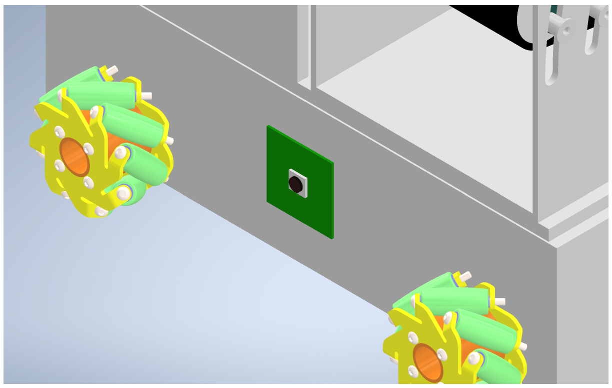


Fig. 42. 3D model for camera attachment to chassis.

A system schematic, illustrated in Figure 43, has been created using AutoCAD Electrical showing the hardware connections. It shows the eighteen needed GPIO pins for the IR sensors, two USB port connections for the cameras, power supply, and serial comminication between the microcontrollera, as well as a full capture of the Raspberry Pi.

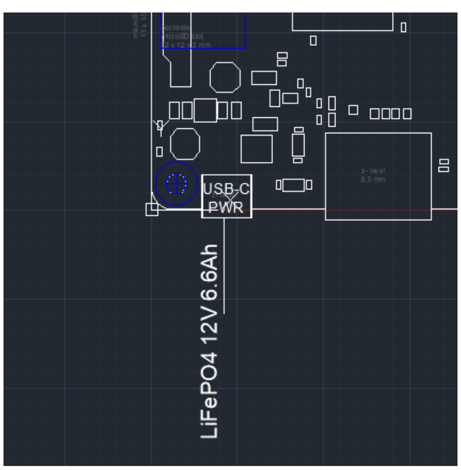


Fig. 44. Navigation system to power schematic.

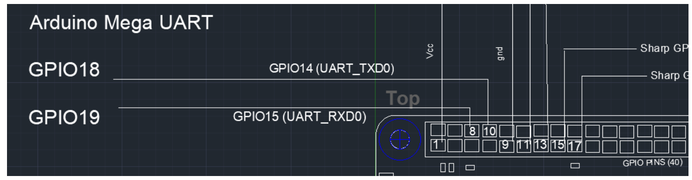


Fig. 45. Navigation system serial connections to Arduino Mega.

B. Bill of Materials

Figure 46 shows a complete breakdown of pricing per component as well as a total system cost.

ITEM	DESCRIPTION	UNIT COST	QUANTITY	TOTAL COST
Raspberri Pi 4 8GB	Microcontroller	$ 149.99	1	$ 149.99
Sharp IR Distance Sensor	IR sensor	$ 8.99	4	$ 35.96
Arducam USB Camera V2 Module	Camera	$ 29.99	2	$ 59.98
			Total:	**$ 245.93**

Fig. 46. Navigation system Bill of Materials.

VIII. Drive Train

The drivetrain subsystem of the project has been designed in order to maximize maneuverability and increase the effectiveness of other subsystems within the robot. This subsystem will be comprised of three primary components: DC motors, H-bridge motor drivers, and wheels.

A. Wheels

Previously, it was intended that the design would utilize standard friction wheels. However, after more consideration, it was finally decided that mecanum wheels will be used in the final design. The primary advantage of using these wheels is that they will allow the robot to traverse in any direction, including horizontally. It is advantageous for the robot to move horizontally because it will make obtaining and firing beads easier and more effective. Specifically, moving horizontally will reduce the distance from the robot to the trees and nets.

When researching different wheels, it is important to consider weight capacity. The robot will have a certain mass, and the wheels will need to be able to support it among the four of them. The team has approximated that the robot will weigh about 12 lbs. The weight breakdown of the robot can be seen in Figure 11.

After researching wheels, it was decided that the team will utilize the same type of mecanum wheels from a previous SECON robot iteration. The wheels that will be used have a radius of 26mm, and can together support up to 30 lbs, which far exceeds the anticipated need of this application.

B. Motors

The next component needed in the design of the drivetrain subsystem is the DC motor model. The primary specifications of the motor include the maximum speed of the entire robot as well as its acceleration to reach this speed. Earlier in the semester, the desired speed of the robot was set at 2 ft/s, but this has since been cut down. The team believes that 1.5 ft/s is a reasonable speed for the robot to traverse the track.

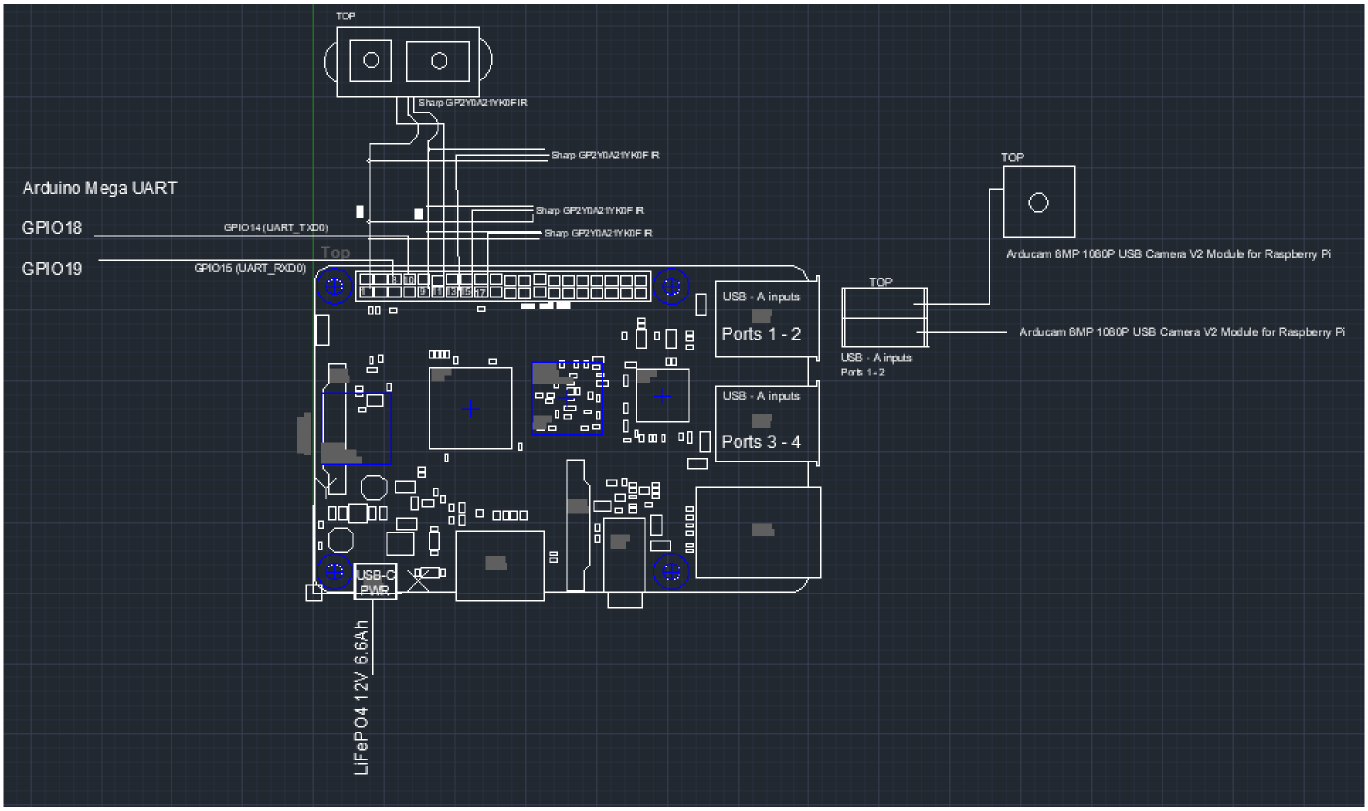


Fig. 43. Full navigation system schematic.

Additionally, the original 2 ft/s parameter would have been harder to accomplish when considering that the robot will have to stay centered on the white line on the track. Reducing the speed will allow for more accurate navigation capabilities. The maximum speed that the robot is able to achieve will ultimately depend on the accuracy and speed of the navigation control system that the team is able to produce.

In addition to maximum speed, robot acceleration is also a parameter that affects the specific motor hardware that the project utilizes. It is desirable that the robot reach its maximum speed in one second. Thus, with the top speed set at 1.5 ft/s, an acceleration of 1.5 ft/s^2 is required. Similar to the wheels, weight is also a crucial parameter that will affect the specific motor hardware used. The weight of the robot directly affects the total load seen by the motors, and thus a motor is needed that will be able to move this weight at the speed required.

Maximum incline is something that should be considered for motor sizing as well. In this specific application, the maximum incline is expected to be relatively small. The environment that the robot will be competing in should be a fairly level board, and so the team estimated that the maximum incline that the motors will have to overcome would be about 15 degrees.

Wheel radius is another factor that will change the requirements of the motors. Increasing the wheel radius will in turn increase the torque requirements of the motors, and vice versa. Conversely, increasing the wheel radius will decrease the number of revolutions per minute (rpm) that the motors will have to turn to turn in order to achieve a given traversal speed. The wheel radius in this application was mentioned earlier, and is 26 mm. This smaller radius will mean that the motors used in this project will need to have a higher rated rpm in order to achieve the top speed of 1.5 ft/s.

Finally, the last parameter that will affect the rating of the motor is the number of drive motors used. As one would expect, increasing the number of motors utilized will decrease the torque requirements of each motor. This application will require that four motors be used to independently drive each mecanum wheel, allowing for the desired lateral motion.

The effects that all of the aforementioned parameters will have on the motor requirements can be seen below in Equation 2:

$$T = \frac{100}{e} * \frac{(\alpha + g * \sin(\theta)) * M * R}{N}$$

It is important to note that the International System of Units (SI) must be used in the equation above. In the equation, α represents the desired acceleration of the robot in m/s^2. When converting from the previously mentioned desired acceleration of 1.5 ft/s^2, the acceleration becomes 0.452 m/s^2. The symbol g represents the acceleration due to gravity and is constant. θ represents the maximum incline, and as mentioned earlier is set to 15 degrees. M represents the mass of the robot in kilograms. When converting the anticipated robot weight of 12.35 lbs, the result becomes 5.60 kg. The wheel radius is already in SI units, and is 0.026 m. N represents the number of motors, and is set to 4.

It is worth noting that an addition variable, e, is used in Equation 2. This variable represents the total efficiency of the system, including power lost to heat, friction of the gears, etc. When considering the entire system, this parameter can be difficult to estimate. It is desirable that the system be as efficient as possible, but the team believes that using a more pragmatic value is beneficial. Many websites use a standard value of 65% for e, however it was decided that a value of 50% would be used in this application in order to allow for more inefficiencies.

When using Equation 2 with the aforementioned parameters, the resultant required torque is calculated as 0.19270 Nm. When converting to kgf-cm, the result is 2.22 kgf-cm. Additionally, the required rpm of the motors in order to achieve the desired top speed will be 168 rpm. Using these parameters, the team was able to find a motor model that matches quite closely to the calculated requirements. The motor is a 12 volt DC gear motor with a gear reduction ratio of the motor is 50:1. The rated operating points of the motor can be seen below in Figure 47.

Maximum Efficiency At 12 V	**51%**
Speed At Maximum Efficiency	**180 RPM**
Torque At Maximum Efficiency	**2.2 kg-cm**
Current At Maximum Efficiency	**0.66 A**
Power at Maximum Efficiency	**4 Watts**

Fig. 47. Rated operating points of the motor.

As can be seen from Table above, the rated operating point of this motor matches very closely to the requirements of this application. Further, this motor slightly exceeds the requirements of this project, which is advantageous because it will allow for some tolerance if the robot weighs slightly more than calculated.

The output shaft of the motor is 6 mm in diameter. However, the wheels that this project will utilize have a hub diameter of 3 mm. The team will utilize the machine shop in order to further drill the hubs of the wheels to match the required 6 mm diameter. It is worth noting that the back up wheels from the previous robot have already been drilled to 6 mm.

C. H-Bridge Circuitry

In order to get the best performance out of the drivetrain subsystem, it is necessary to utilize motor driving circuitry that will allow the robot to move in different directions and at variable speeds. The type of circuit that will be utilized in this application is known as an H-bridge.

When considering the H-bridge model that would be utilized, it was important to ensure that the chosen circuitry matched the power requirements of the motors. The drivetrain motors used in this application are rated for 0.66 A at maximum efficiency, or the rated operating point. As stated earlier, the total load of the robot is anticipated to be less than what the motors are rated for. However, in order to provide some room for error, an additional 15% has been added to the current draw per motor, which places the current draw for each motor at 0.76 A. In addition, the motors being rated for 12 V means that the H-bridges will also have to be able to support that voltage.

Additionally, the H-bridge needed to be PWM compatible in order to drive the motors at variable speeds. PWM enables speed control by varying the average voltage level that the connected motors see.

The specific model that will be used is the L298N motor driving board and is developed by Qunqi. The current rating for each channel of the board is 2 A, which greatly exceeds the worst-case scenario of 0.76 A. The voltage rating of this board is 5 V to 35 V, so it will be able to support the 12 V needed by each of the motors. In addition, this motor driver board has input pins to change the polarity of the voltage to the load, which will allow for direction control of the motors. Finally, this particular L298N board is PWM compatible, and in fact is very commonly used in Arduino projects that involve motor control.

D. Drivetrain Schematic

All of the components that will be required to construct the drivetrain have been compiled, and the interconnects of each component are shown in the design artifact in Figures 48 and 49.

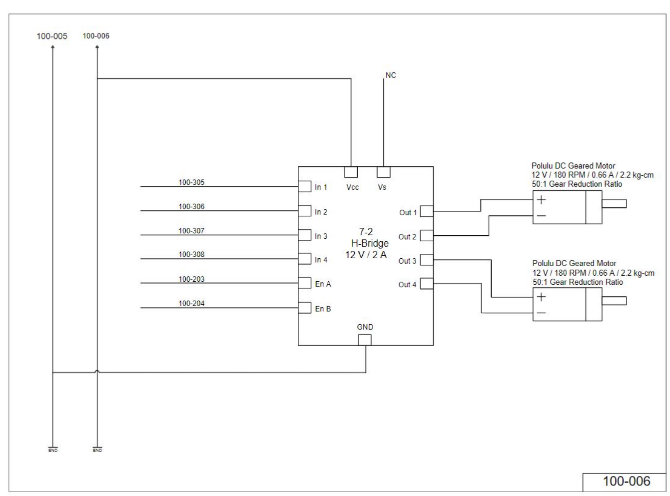


Fig. 48. Drivetrain CAD model 1.

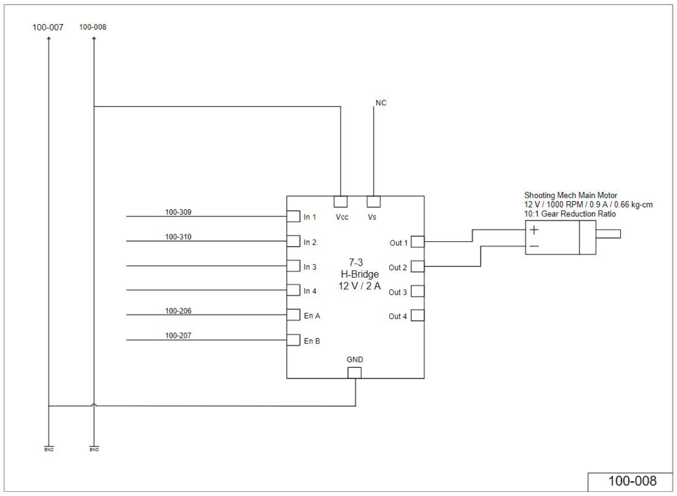


Fig. 49. Drivetrain CAD model 2.

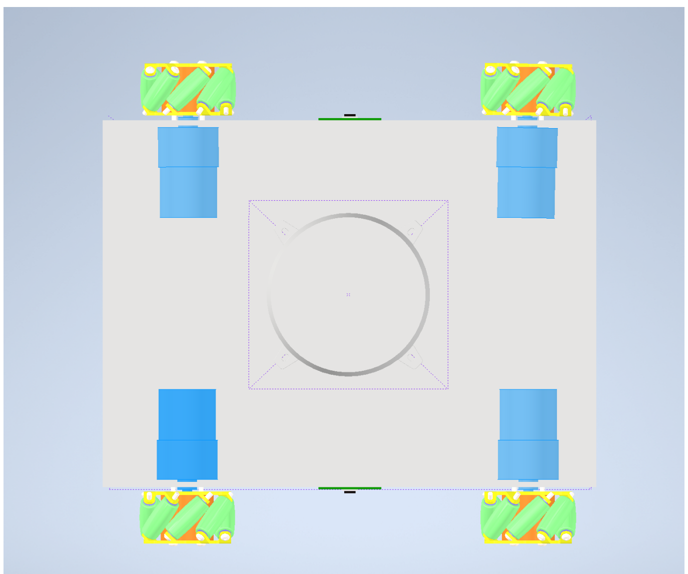


Fig. 50. Motors mounted to lower chassis.

Figure above is a visual demonstration of the interconnectivity of the various drivetrain components. On the left are the individual signals that the L298N will be receiving from the main microcontroller. Inputs one through four will control the polarity of each motor, while inputs ENA and ENB will be the PWM inputs to control the speed. The input V_{CC} is the drive voltage which will need to come from a 12 V regulator. Outputs one through two will control the first motor, while outputs three and four will control the second motor. It's worth noting that two identical versions of this schematic will be implemented in the final design, the only difference being the pins coming from the microcontroller in both cases. Each motor will be physically mounted to the lower chassis via screws through the chassis acrylic and threaded into the front face of the motor. The motors are manufactured with M3 threaded holes for this specific purpose. The holes in the acrylic will be drilled as previously discussed.

E. Bill of Materials

The bill of materials includes an additional set of M3 screws that will be used to mount the motors to the acrylic chassis. Mounting and screw holes will be intentionally left in the design when cutting the acrylic.

Item Description	Quantity	Cost Per Unit	Total Cost
12 V DC Geared Motor	5	$24.95	$124.75
M3 Screw Set	1	$0.99	$0.99
Mecanum Wheel 4 Pack	1	$70.94	$70.94
L298N H-Bridge Motor Driver (2 Pack)	2	$8.99	$17.98
		Total:	**$214.66**

Fig. 51. Drivetrain Bill of Materials.

IX. Robotic Arm

The team plans to utilize the robotic arm to grab the beads off the tree and place said beads into a magazine to be loaded into the shooting mechanism. One reason for choosing the Lynxmotion AL5D is for the 5 degrees of freedom. There is one servo motor for the shoulder joint, one for the elbow joint, one for the wrist, one for the wrist turning motion and one for the gripping mechanism (end effector). The shoulder, elbow, wrist and turning motion can all move 180 degrees. The end effector has the capability to open to 1.25 inches. This opening length is perfect for grabbing hanging beads off the tree; however, other aspects of the stock end effector are not ideal for the given task. This includes the short length of the gripping mechanism. Because of this, an extension to the end effector may be 3D printed if trial runs show that its current length is unsatisfactory. It is common engineering practice to design a custom end effector to be used with a third-party robotic arm. The basic robotic arm extension length can be seen below. Figure 52 shows the minimum height the chassis needs to be for the robot arm to reach the 21.5 inch tree branch.

Height of Chassis Minimum without extension	3″
Height of Chassis Minimum with extension	1.6750″

Fig. 52. Robot Arm extension lengths.

With these specifications the team solved for the height needed on the chassis to be able to reach the tree branch.

Height without extension add on	18.5″
Height with extension add on	19.8250″

Fig. 53. Minimum height required of chassis for arm to reach

The distance from the tree to the robotic arm was varied to verify how far the robot can be and still reach the tree limb. The below graphs show the lengths that can be achieved with the Lynxmotion AL5D Robotic Arm. The values across the top show the height at which the base of the robotic arm is sitting on the gameboard. The values down the side of the graph show the distance away the base of the arm is from the tree. Both values can be seen in blue. Number values in red show lengths that the robot cannot reach from the given height and distance. Number values seen in yellow appear -$\frac{1}{4}$ inches away from the max arm reach distance. Number values seen in green are distances that the robot can reach from the given height and distance.

Robot Chassis Height	1	2	3	4	5	6	6.5	7	7.5
Inches Away From Tree	************No Extension Arm Length = 16.5"************								
3	20.7183	19.74294	18.7417	17.7553	16.7705	15.7877	15.2971	14.8071	14.3178
3.25	20.756	19.769	18.7833	17.7992	16.817	15.8371	15.348	14.8598	14.3723
3.5	20.7966	19.8116	18.8282	17.8466	16.8671	15.8902	15.4029	14.9164	14.4309
3.75	20.8402	19.8573	18.8762	17.8973	16.9208	15.9472	15.4616	14.9771	14.4935
4	20.8866	19.906	18.9275	17.9513	16.9779	16.0078	15.5242	15.0416	14.5602
4.25	20.9359	19.9578	18.9819	18.0087	17.0386	16.0721	15.5905	15.11	14.6309
4.5	20.9881	20.0125	19.0394	18.0693	17.1026	16.14	15.6605	15.1822	14.7054
4.75	21.0431	20.0702	19.1001	18.1332	17.1701	16.2115	15.7341	15.2582	14.7839
5	21.1009	20.1308	19.138	18.2003	17.2409	16.2865	15.8114	15.3379	14.8661
5.25	21.1616	20.1944	19.2305	18.2705	17.3151	16.365	15.8922	15.4212	14.952
5.5	21.225	20.2608	19.3003	18.3439	17.3925	16.4469	15.9765	15.5081	15.0416
5.75	21.2911	20.3301	19.373	18.4204	17.4732	16.5322	16.0643	15.5985	15.1348
6	21.36	20.4022	19.4487	18.5	17.557	16.6208	16.1555	15.6924	15.2315
6.25	21.4316	20.4771	19.5272	18.5826	17.644	16.7126	16.25	15.7896	15.3317
6.5	21.5058	20.5548	19.6087	18.6682	17.7341	16.8077	16.3478	15.8902	15.4353
6.75	21.5827	20.6352	19.693	18.7567	17.8273	16.906	16.4488	15.9941	15.5423
7					17.9234	17.0074	16.5529	16.1012	15.6525
7.25						17.1118	16.6602	16.2115	15.7659
7.5						17.2192	16.7705	16.3248	15.8824

Fig. 54. Arm Length No Extension Calculations

Robot Chassis Height	1	2	3	4	5	6	6.5	7	7.5
Inches Away From Tree	************Extension Added Arm Length = 17.825"************								
3	20.7183	19.74294	18.7417	17.7553	16.7705	15.7877	15.2971	14.8071	14.3178
3.25	20.756	19.769	18.7833	17.7992	16.817	15.8371	15.348	14.8598	14.3723
3.5	20.7966	19.8116	18.8282	17.8466	16.8671	15.8902	15.4029	14.9164	14.4309
3.75	20.8402	19.8573	18.8762	17.8973	16.9208	15.9472	15.4616	14.9771	14.4935
4	20.8866	19.906	18.9275	17.9513	16.9779	16.0078	15.5242	15.0416	14.5602
4.25	20.9359	19.9578	18.9819	18.0087	17.0386	16.0721	15.5905	15.11	14.6309
4.5	20.9881	20.0125	19.0394	18.0693	17.1026	16.14	15.6605	15.1822	14.7054
4.75	21.0431	20.0702	19.1001	18.1332	17.1701	16.2115	15.7341	15.2582	14.7839
5	21.1009	20.1308	19.138	18.2003	17.2409	16.2865	15.8114	15.3379	14.8661
5.25	21.1616	20.1944	19.2305	18.2705	17.3151	16.365	15.8922	15.4212	14.952
5.5	21.225	20.2608	19.3003	18.3439	17.3925	16.4469	15.9765	15.5081	15.0416
5.75	21.2911	20.3301	19.373	18.4204	17.4732	16.5322	16.0643	15.5985	15.1348
6	21.36	20.4022	19.4487	18.5	17.557	16.6208	16.1555	15.6924	15.2315
6.25	21.4316	20.4771	19.5272	18.5826	17.644	16.7126	16.25	15.7896	15.3317
6.5	21.5058	20.5548	19.6087	18.6682	17.7341	16.8077	16.3478	15.8902	15.4353
6.75	21.5827	20.6352	19.693	18.7567	17.8273	16.906	16.4488	15.9941	15.5423
7					17.9234	17.0074	16.5529	16.1012	15.6525
7.25						17.1118	16.6602	16.2115	15.7659
7.5						17.2192	16.7705	16.3248	15.8824

Fig. 55. Arm Length Extension Calculations

There are two big takeaways from the two figures above. For one, the robot arm becomes more versatile when the extension is added. For instance, the robotic arm could only be 5.5 inches away from the tree; however, the extension to the arm allows the robot to be 7.5 inches away from the tree. The next big takeaway is that the robotic arm can accomplish the needed task of grabbing the beads from the tree. The Lynxmotion AL5D also gives the team options about chassis height placement. The most versatile placement is having the arm at 7.5 inches above the chassis. To do this, the team will have to have the robot start in a resting position that points down within the robot. The team will still be able to fully utilize the robotic arm at 6.5 inches if the chassis design does not permit the 7.5 inch starting position. The exact placement of the robotic arm will be seen in the CAD model presented with the chassis.

The next aspect of the arm will include the electrical components. Each servo motor has three wires. Black for ground, red for power (4.8-6V) and yellow for the PWM signal. The best way to control the servo motors will be through the utilization of the Adafruit 16-Channel 12-bit PWM/Servo Shield – 12C interface. The board will easily plug into our Arduino MEGA based on the design. Through this, the team will be able to control each of the servo motors while also providing reverse polarity protection. The servo motor will be connected to the Adafruit shield seen in Figures 56 and 57. These images are better represented in the CAD package pages 200-006 and 200-201 in Appendix B.

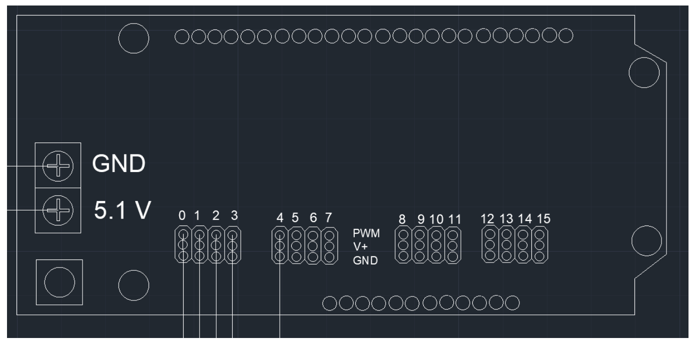


Fig. 56. Servo Shield CAD Model

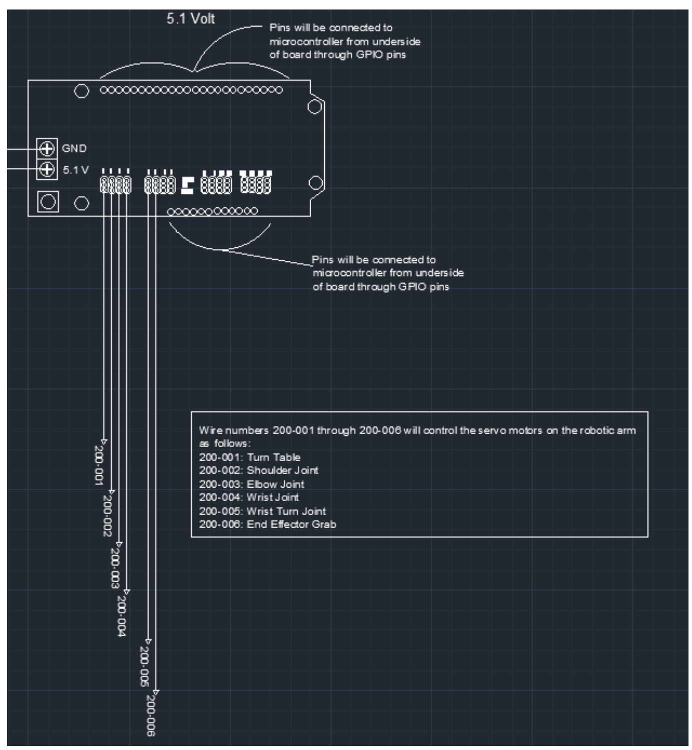


Fig. 57. Robotic Arm Servo Connections

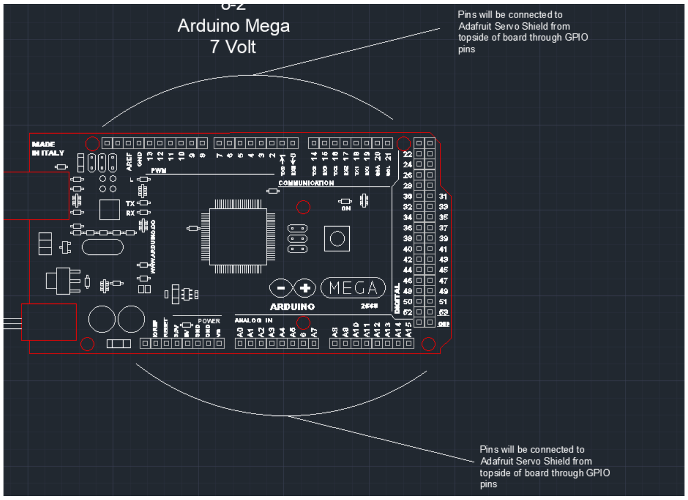


Fig. 58. Arduino Mega GPIO Pin Representation

The team will use the first five three-set pins (0,1,2,3,4,5) to control the robotic arm. The board itself will be powered from the 5.1 V power bus (2-2). The board will use the outer rim pins, located on the top and bottom of the board, to plug into the Arduino MEGA. By doing this, the team will be able to give the desired PWM signals to the board. The PWM signals will need to be communicated to the board in durations that last between 0.5 ms to 2.5 ms followed by a delay between 20 ms to 30 ms. The servo motors then translate the signal through onboard electronics. The pulse will need to be repeated for the servo to hold the desired position. For these reasons, the Adafruit component will be the best option to translate these signals from the microcontroller.

A. Bill of Materials

Item	Description	Unit Cost	Quantity	Total Cost
Lynxmotion AL5D	Robotic Arm	399.00	1	**399.00**
Adafruit Servo Shield	16-Channel PWM Servo Shield	17.50	1	**17.50**

Fig. 59. Robot arm Bill of Materials.

X. Peripheral System Specifications

The team can be rewarded points for adding a different variety of decorations to the exterior of the robot. Additionally, the team has recruited the Tennessee Tech IEEE chapter in designing a peripheral system that will help maximize the awarded points. The peripheral system was designed to play audio over a speaker, integrate a moving display onto the existing turntable, and include a blinking LED analog circuit. The LED Circuit and Arduino will be powered by the 12 Volt and 7 Volt power busses, respectively. An audio shield was chosen which fits on top of an Arduino Uno, which was chosen for its compact size. The shield is equipped with an SD card reader and audio amplifier allowing the Uno to read and play sound data using existing Arduino Libraries. A TFT LCD display was chosen for the moving display because it allows for more detailed pictures to be displayed through bitmap imagery. The front of the shooting mechanism housing will be cut to allow the screen to be pressed through while obscuring the additional electronics. The screen will be held in place with pre-drilled mounting screws similar to the navigation cameras. A detailed illustration of this is shown in Figure 65. Finally, the NPN transistor circuit seen in Figure 61 allows a grouping of LEDs to blink in a “chase” pattern. This circuit was simulated in LTSpice in order to determine the current draw and blinking frequencies, as well as to verify that the circuit would create the necessary voltage drop across the diodes to make them blink. Simulation results are displayed in Figures 62- 64. An analog circuit was chosen to drive the LEDs instead of utilizing the Arduino in order to reduce the use of GPIO pins and provide the IEEE members with a variety of design experiences. The overall system can be seen in Figure 60.

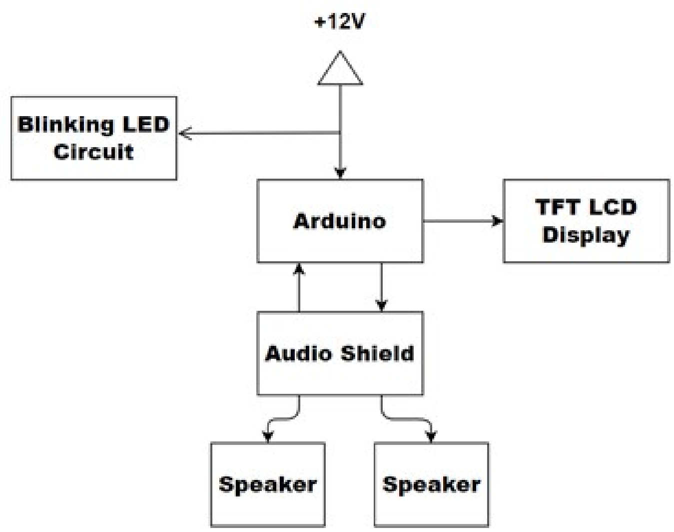


Fig. 60. Block diagram of peripheral system.

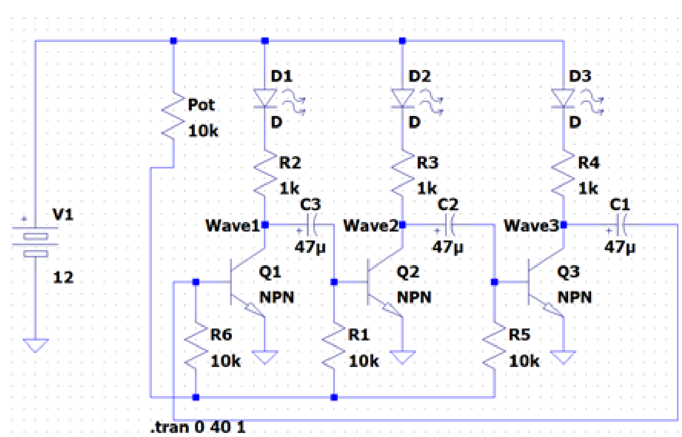


Fig. 61. Circuit diagram of peripheral system LEDs.

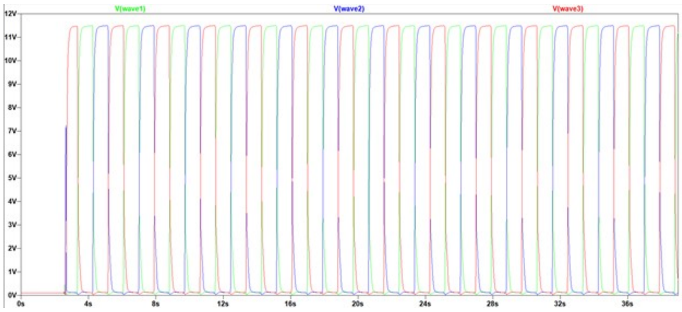


Fig. 62. LED Circuit Voltage Verification

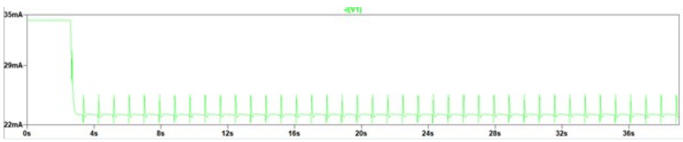


Fig. 63. LED Circuit Current Simulation

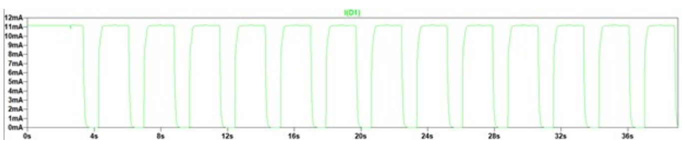


Fig. 64. LED Circuit Current Simulation

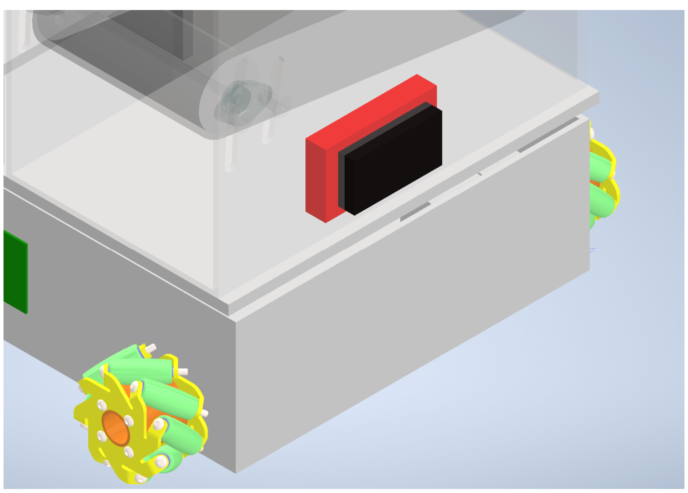


Fig. 65. Peripheral display mounted on front of chassis.

A. Bill of Materials

Item	Description	Unit Cost	Qty.	Total Cost
Assorted LED's	200 Light Emitting Diodes	7.99	1	7.99
Adafruit MP3 Shield	Arduino with Stereo Amp	34.95	1	34.95
DSD Tech Display Module	Arduino Display Module	9.99	1	9.99
Arduino Uno Rev 3	Arduino Uno	20.98	1	20.98

Fig. 66. Peripheral system Bill of Materials.

XI. References

[1] Vaidya, Kapijal. AL5D Robotic Arm. (2016) [Solidworks]. Place: GrabCAD. Available: https://grabcad.com/library/al5d-robotic-arm-1, Accessed on: 12. 15. 2021

[2] V, Vanchinathan. Mechanum Wheel. (2021) [Solidworks]. Place: GrabCAD. Available: https://grabcad.com/library/mechanum-wheel-2, Accessed on: 12. 15. 2021

[3]"Drive Motor Sizing Tutorial," RobotShop Community, Feb. 01, 2014. https://www.robotshop.com/community/blog/show/drive-motor-sizing-tutorial (accessed Dec. 15, 2021).

[4]"Pololu - 50:1 Metal Gearmotor 37Dx54L mm 12V (Helical Pinion)," www.pololu.com. https://www.pololu.com/product/4743 (accessed Dec. 16, 2021)

Appendix A : Simulation Source Code

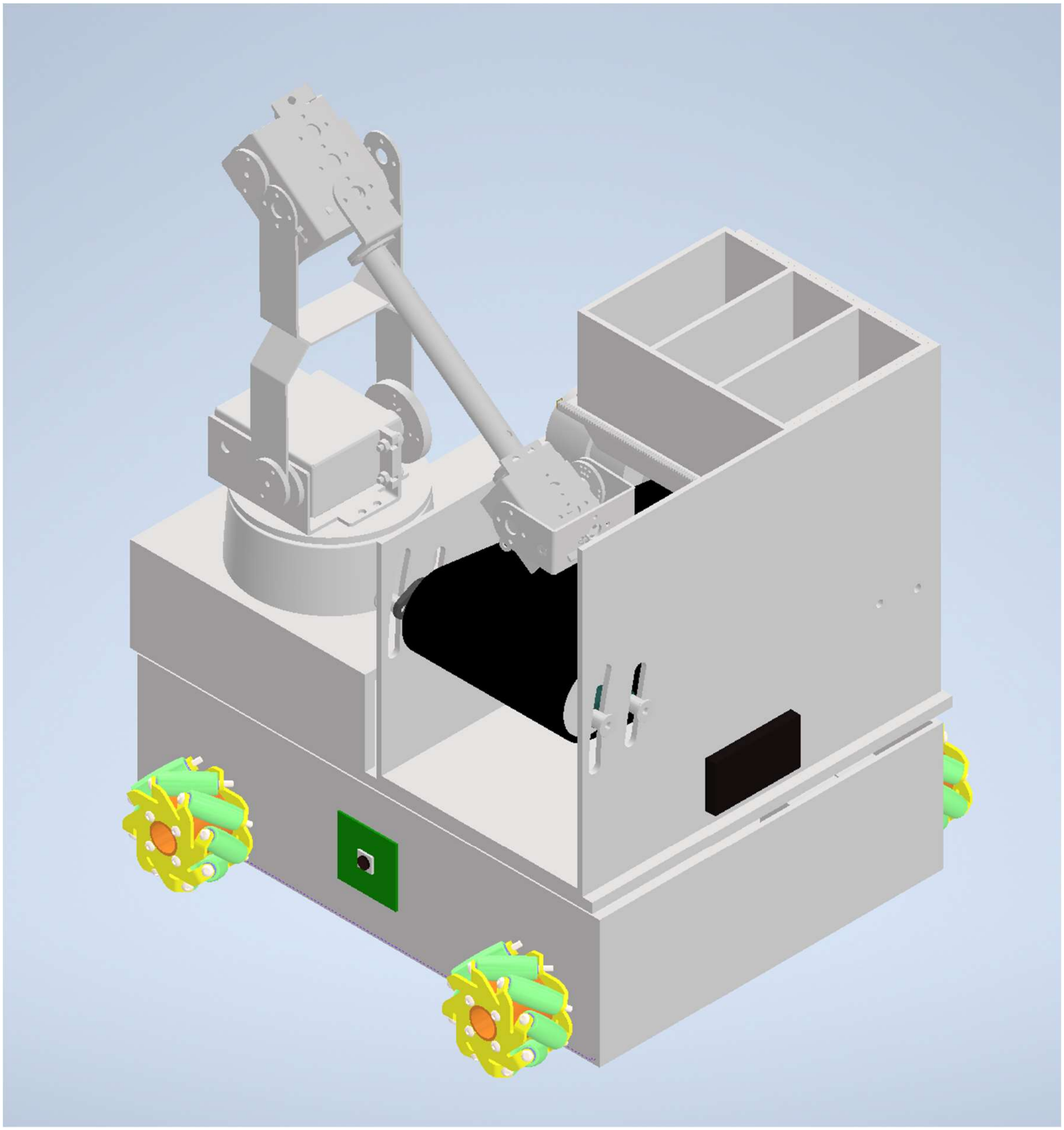
1. Current Draw Calculation

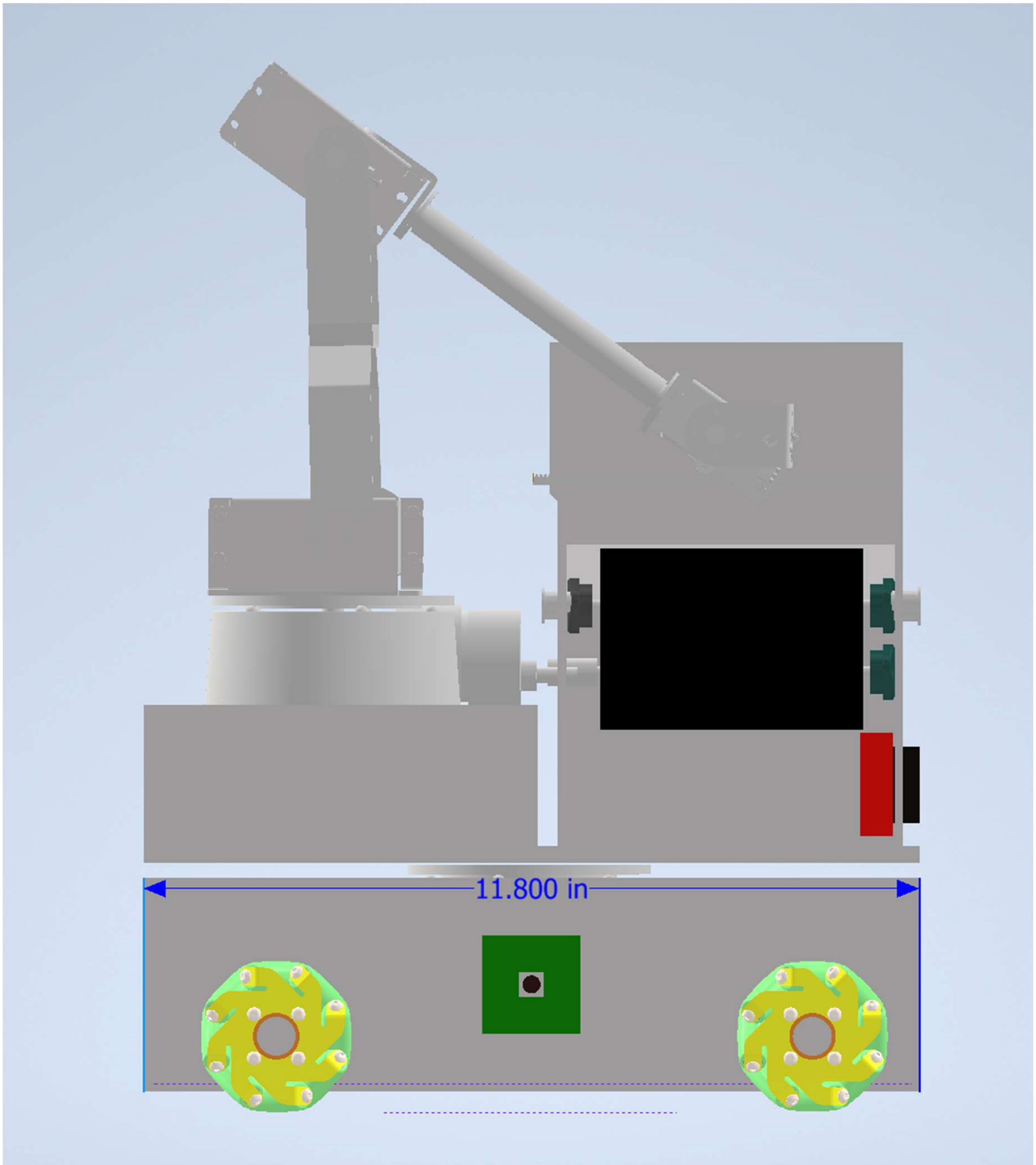
```
*************************************************************************************************
clear all;
close all;
clc;


%The code below calculates the needed current supply from the battery for
%Capstone Power Signoff justification
%The code also calculates the C value of the battery to find out how long
%the battery will be able to operate

%Set values for each component to find current consumption
Pi = 3;
Shield = 0.25;
MEGA = 0.5;
Uno = 0.3;

%Set Values for Buck Boost and Buck Converter Efficiency
BuckboostE = 91.3/100;
BuckE = 92/100;

%Powerbus loss is neglible therefore no measurable loss will be accounted
%for

%12 Volt to 5.1 Volt Buck Converter
Iz = ((5.1*(Pi+Shield))/12)/BuckE

%12 Volt to 7 Volt Buck Converter
Iy = ((7*(MEGA+Uno))/12)/BuckE

%We can now find the needed current that the battery will have to supply
Ia = 1.2;
Ib = 1.2;
Ix = (Iz+Iy+Ia+Ib);
Iactual = ((12*(Ix))/12)/BuckboostE

%C value calculations
%Per datasheet the battery can run 6.6 Amps for 1 Hour at 1 C
Idata = 6.6;
CurrentRatio = Idata/Iactual;
TimeinMinutes = CurrentRatio*60
*************************************************************************************************
```

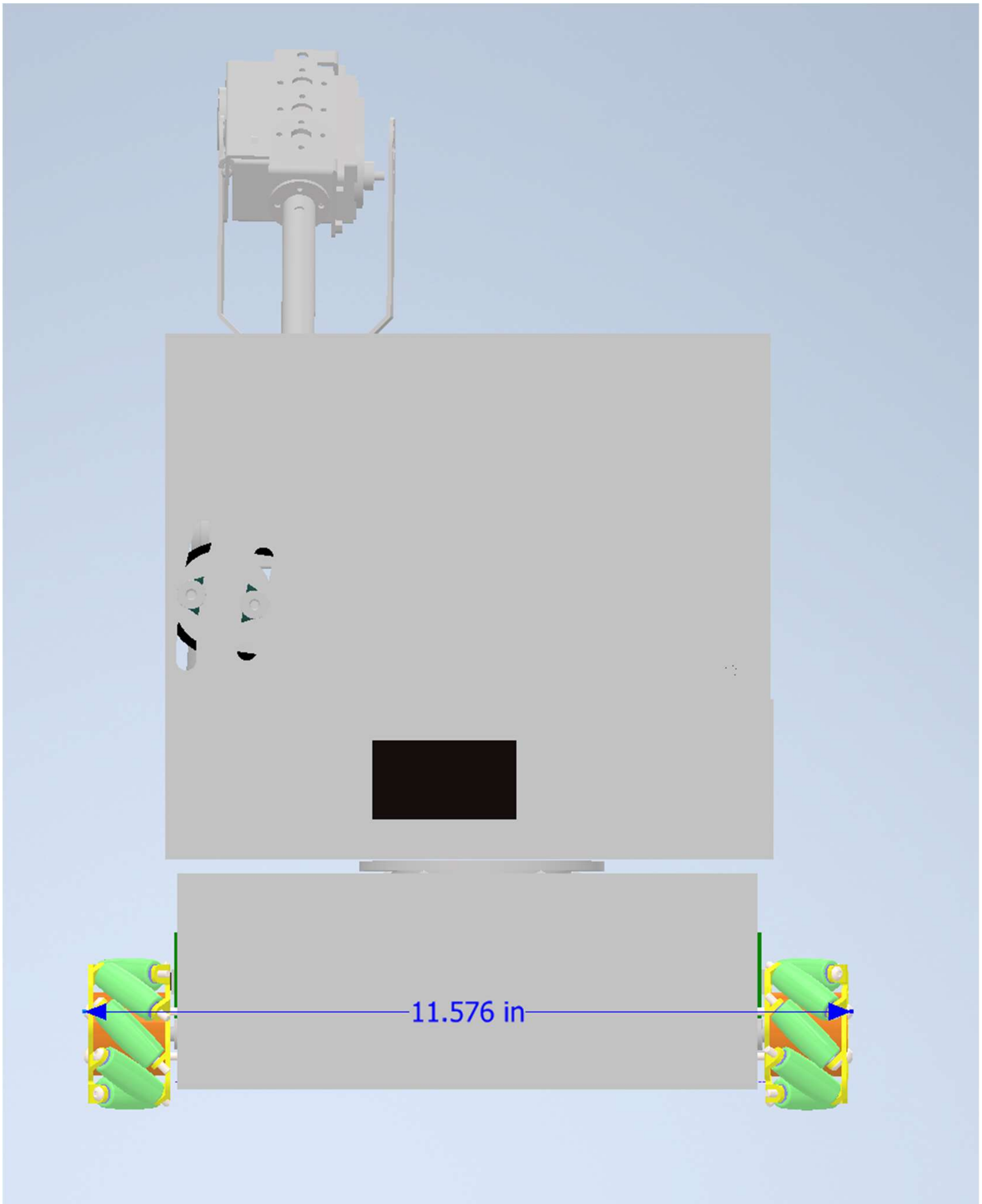
2. Shooting Trajectory Calculation

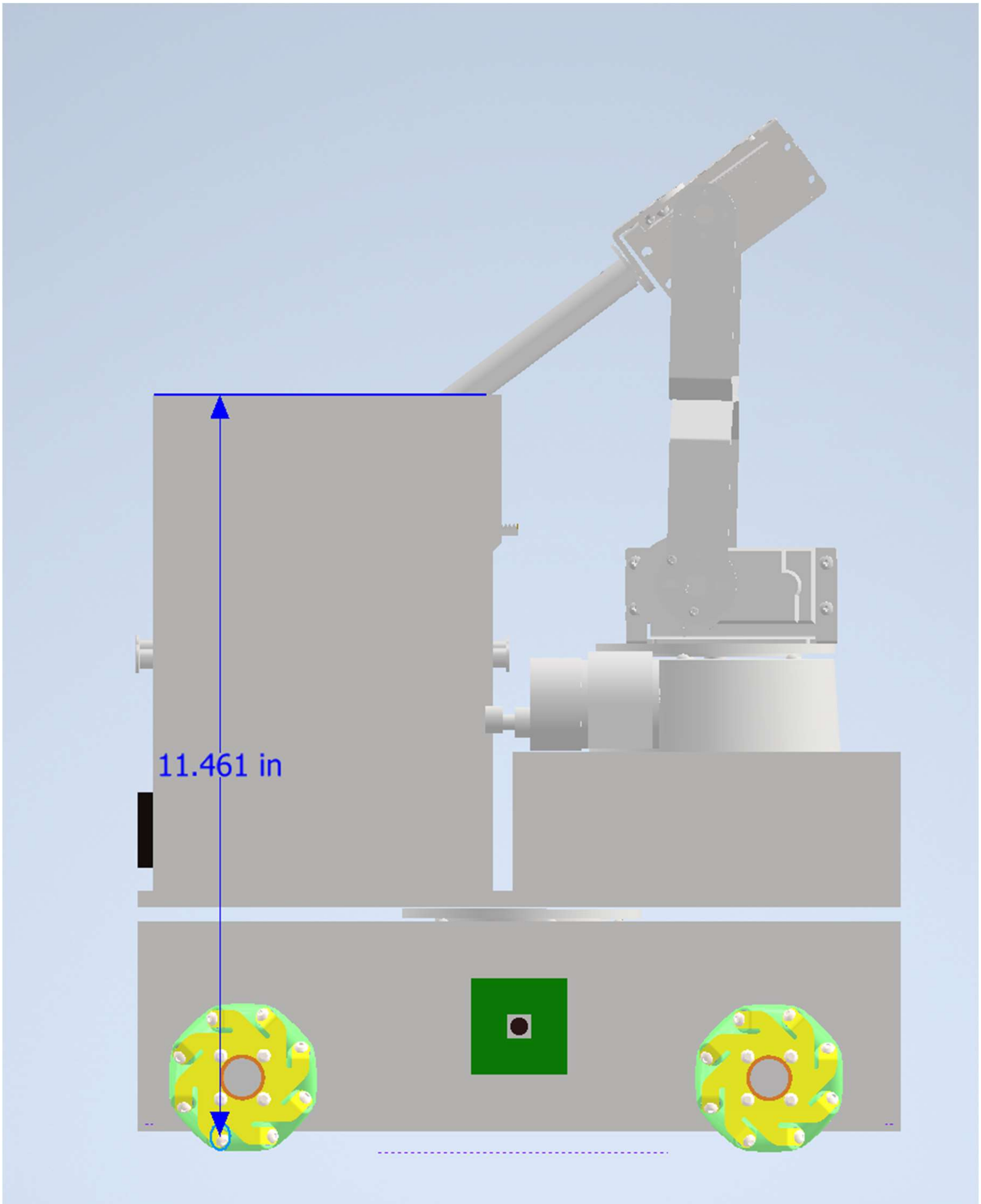
```
%Script to optimize launch angle and conveyor height

%These variables are changeable
motor_rpm = 700;
launch_degrees = 8;
distance_to_net = 5;

%Physics
gravity_inches = -386.22;

%Acceleration has been converted to in/s^2
exit_height = sind(launch_degrees) * 9.17 + 7.8;

exit_velocity_inches_second = (2*pi*0.75*motor_rpm)/60;

additional_height_gained = -
(sind(launch_degrees)*exit_velocity_inches_second)^2/(2*gravity_inches);

time_max_height = 2*additional_height_gained/(sind(launch_degrees)*exit_velocity_inches_second);

time_of_flight = time_max_height + sqrt(2*(additional_height_gained+exit_height)/386.22);

distance_traveled = (exit_velocity_inches_second*cosd(launch_degrees))*time_of_flight;

max_height = exit_height + additional_height_gained;

%Plot The Trajectory
t = linspace(0, time_of_flight);
d = linspace(0, distance_traveled);

height = exit_height +
sind(launch_degrees)*exit_velocity_inches_second.*t+.5*(gravity_inches).*t.^2;

hold onplot(d, height);

degstr = sprintf('Launch Degrees %.1f', launch_degrees);

%legend(degstr);
ylim([0 14]);
xlim([0 6]);

%xlim([0 6]);
xlabel('Distance From Conveyor (inches)');
ylabel('Bead Cluster Height (Inches)');
title('Flight Path of Bead Cluster');

%Draw the net for visual aid
line([distance_to_net distance_to_net], [5.5 13.5], 'linewidth', 8, 'Color', 'black');
line([distance_to_net distance_to_net], [0 5.5], 'Color', 'black');
```

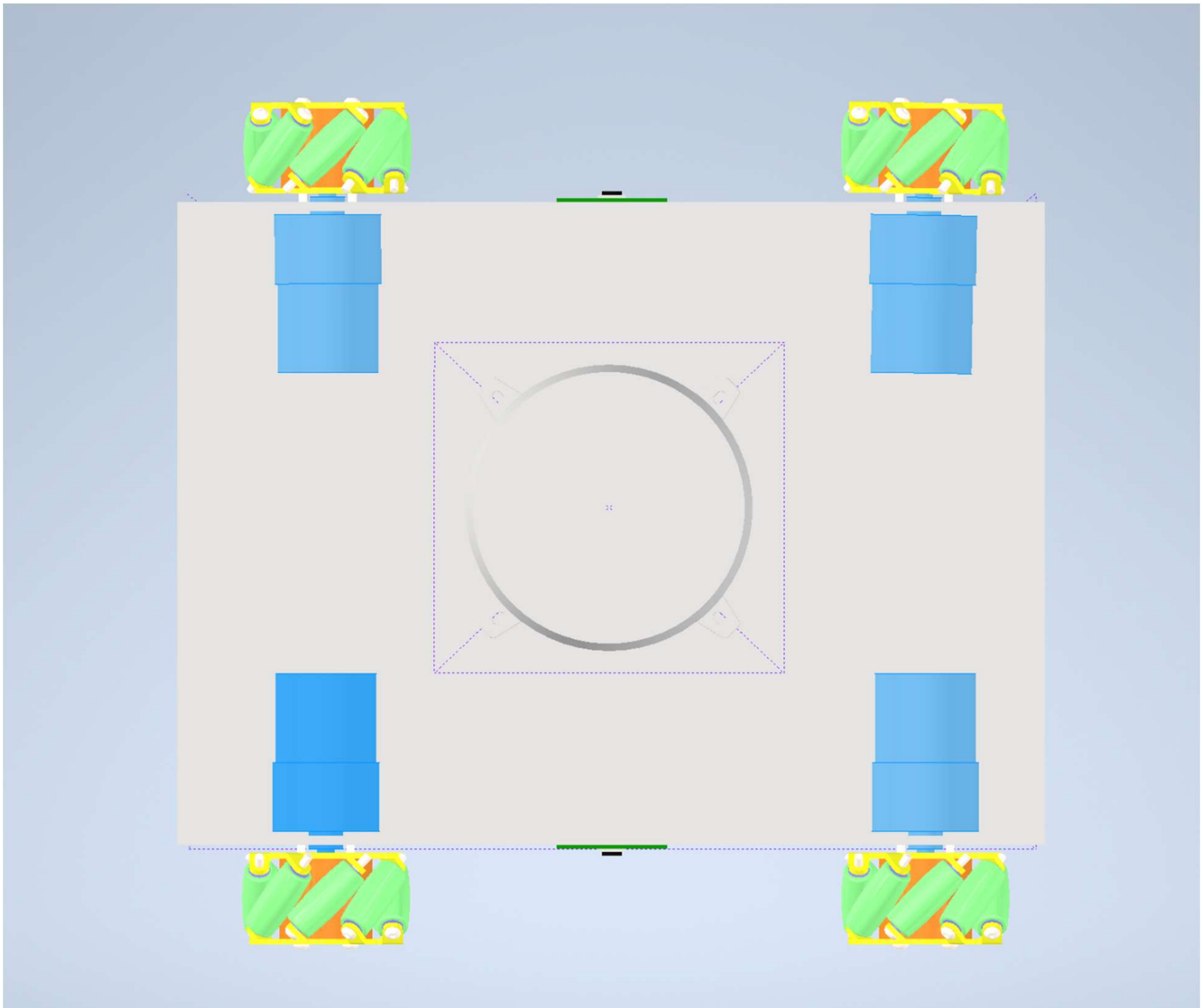
Appendix B: Full CAD Documentation

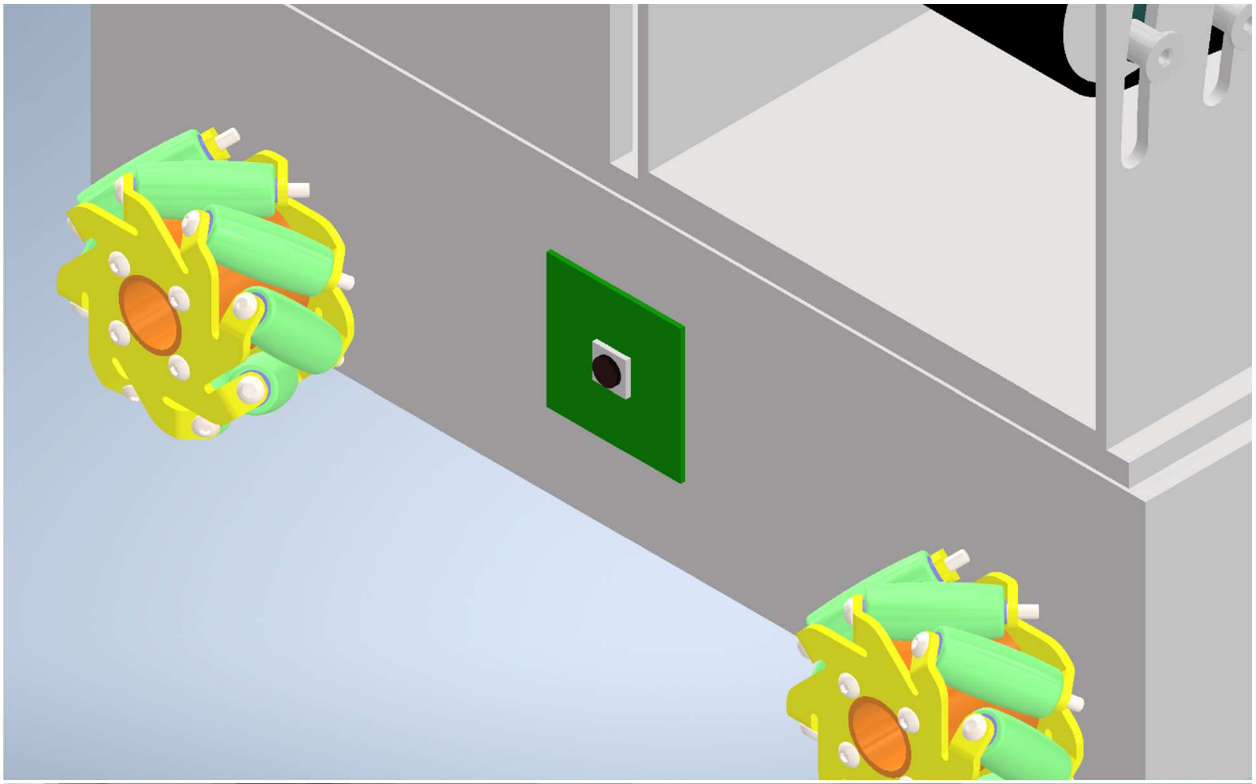


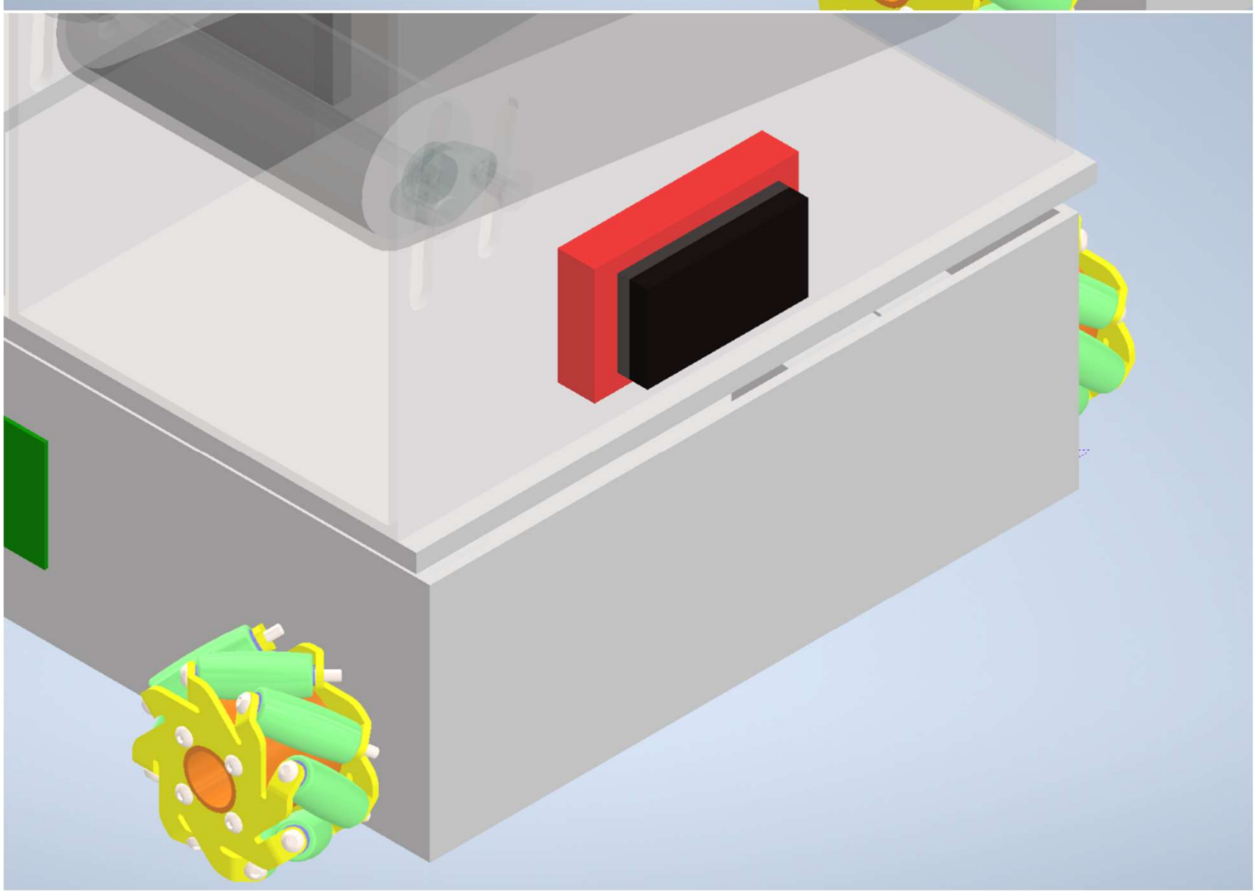

11.800 in

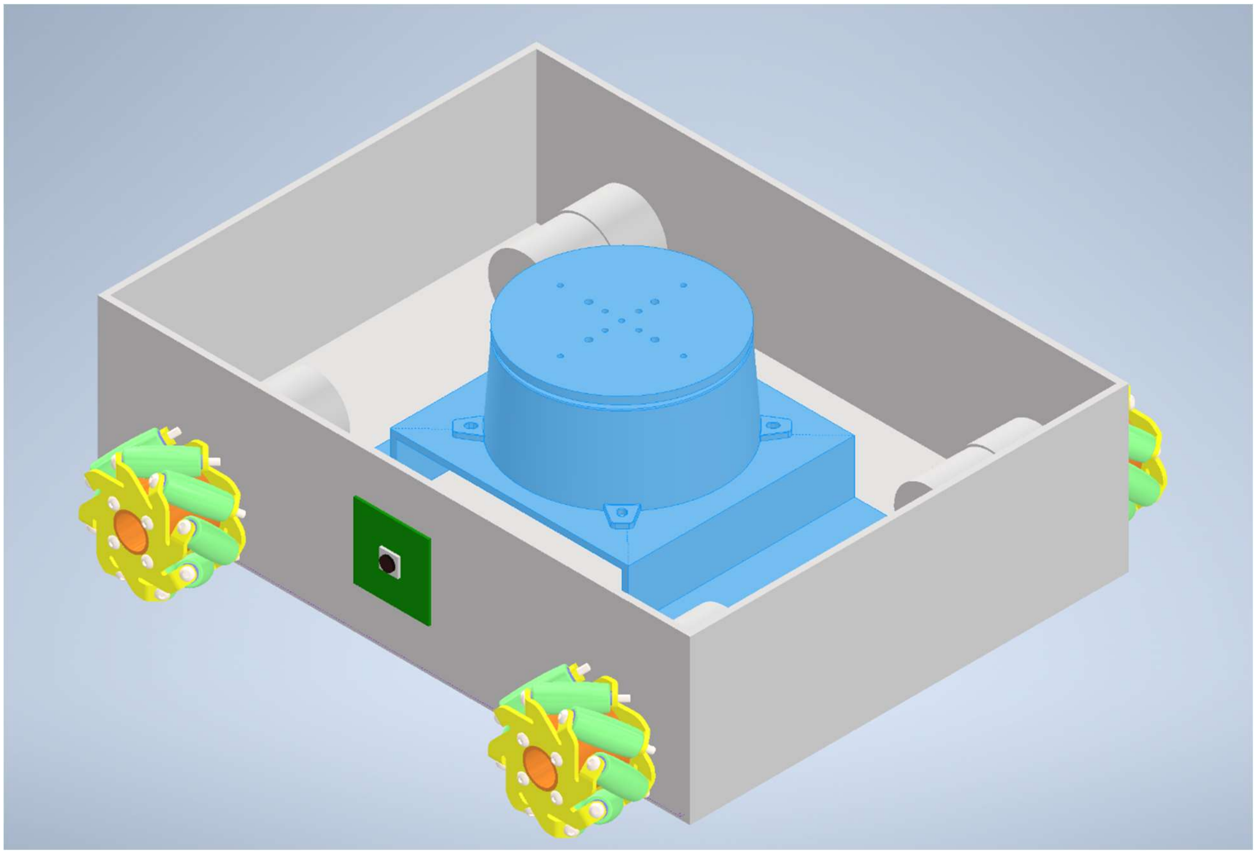

11.576 in

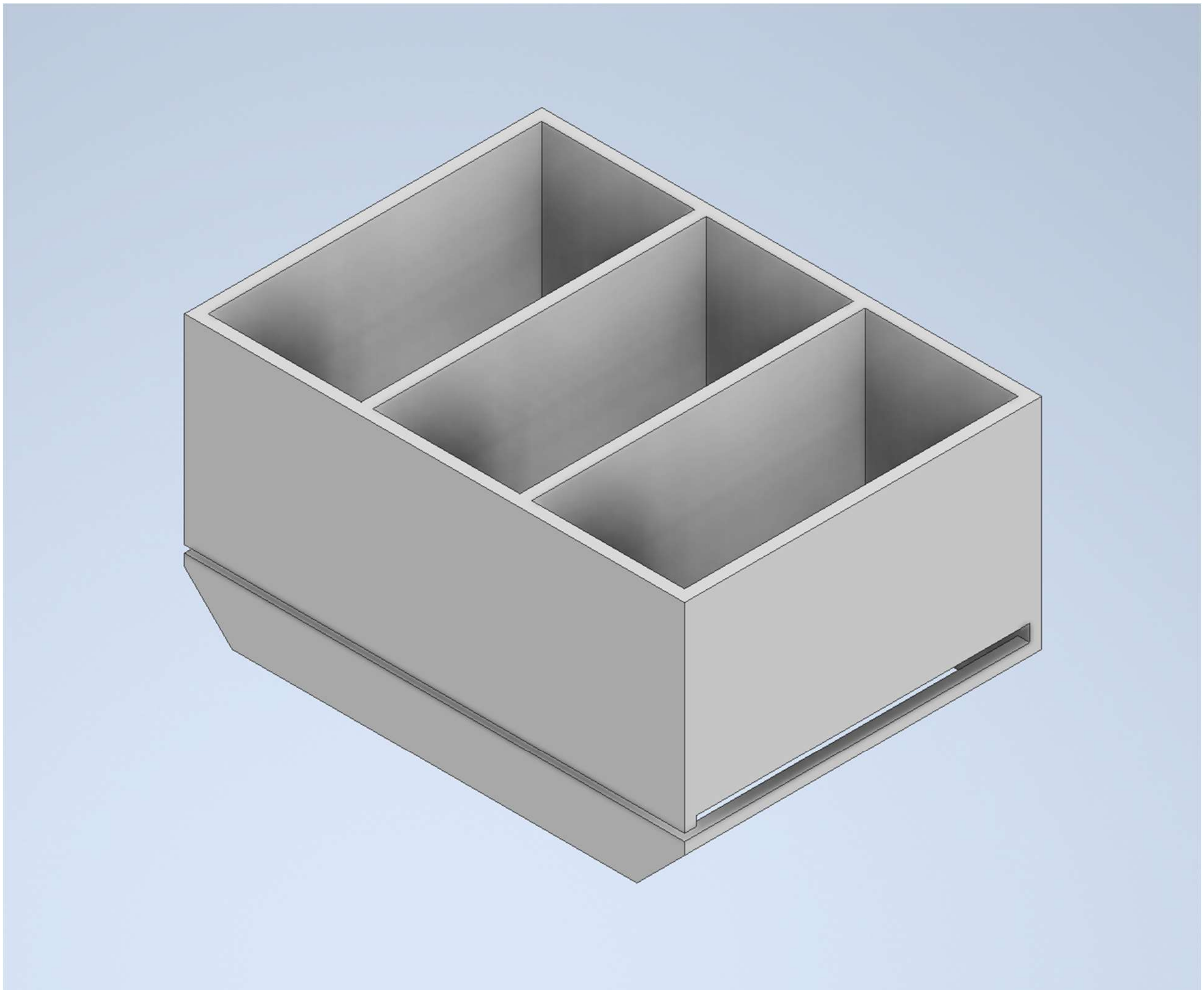

11.461 in

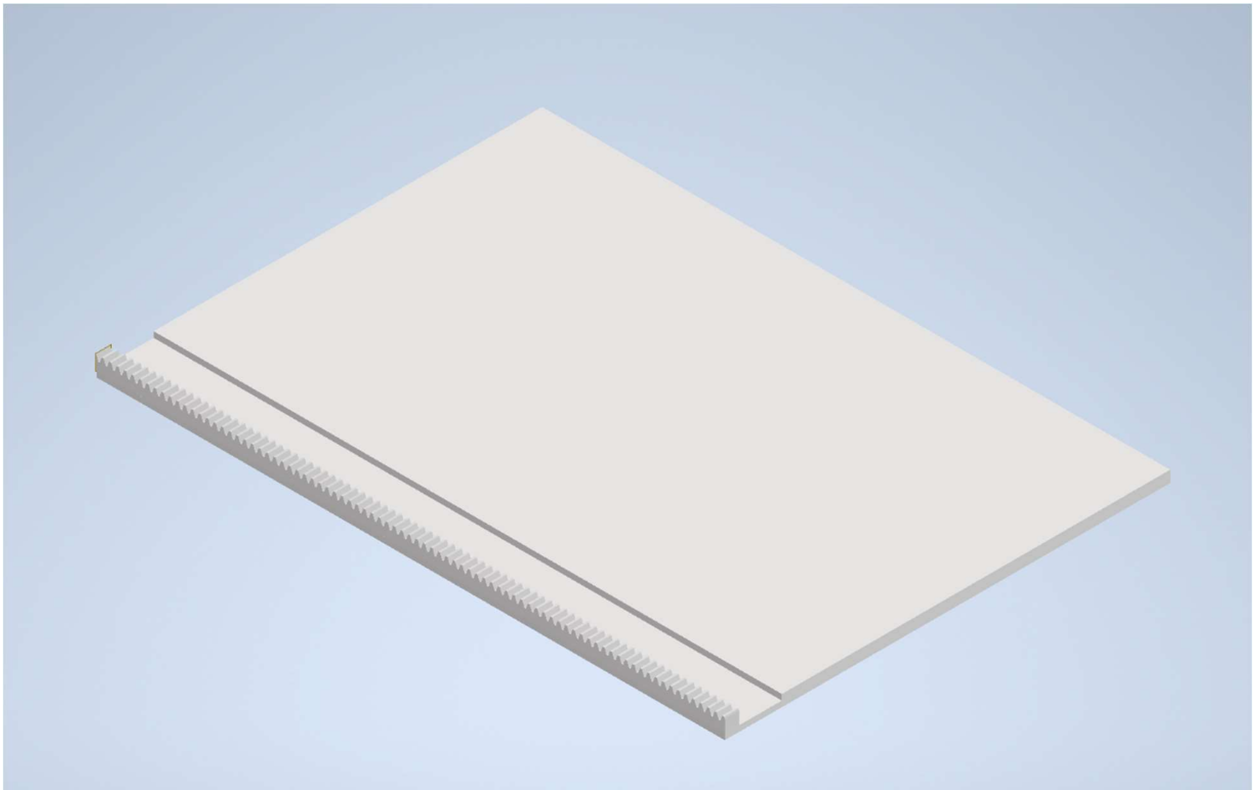


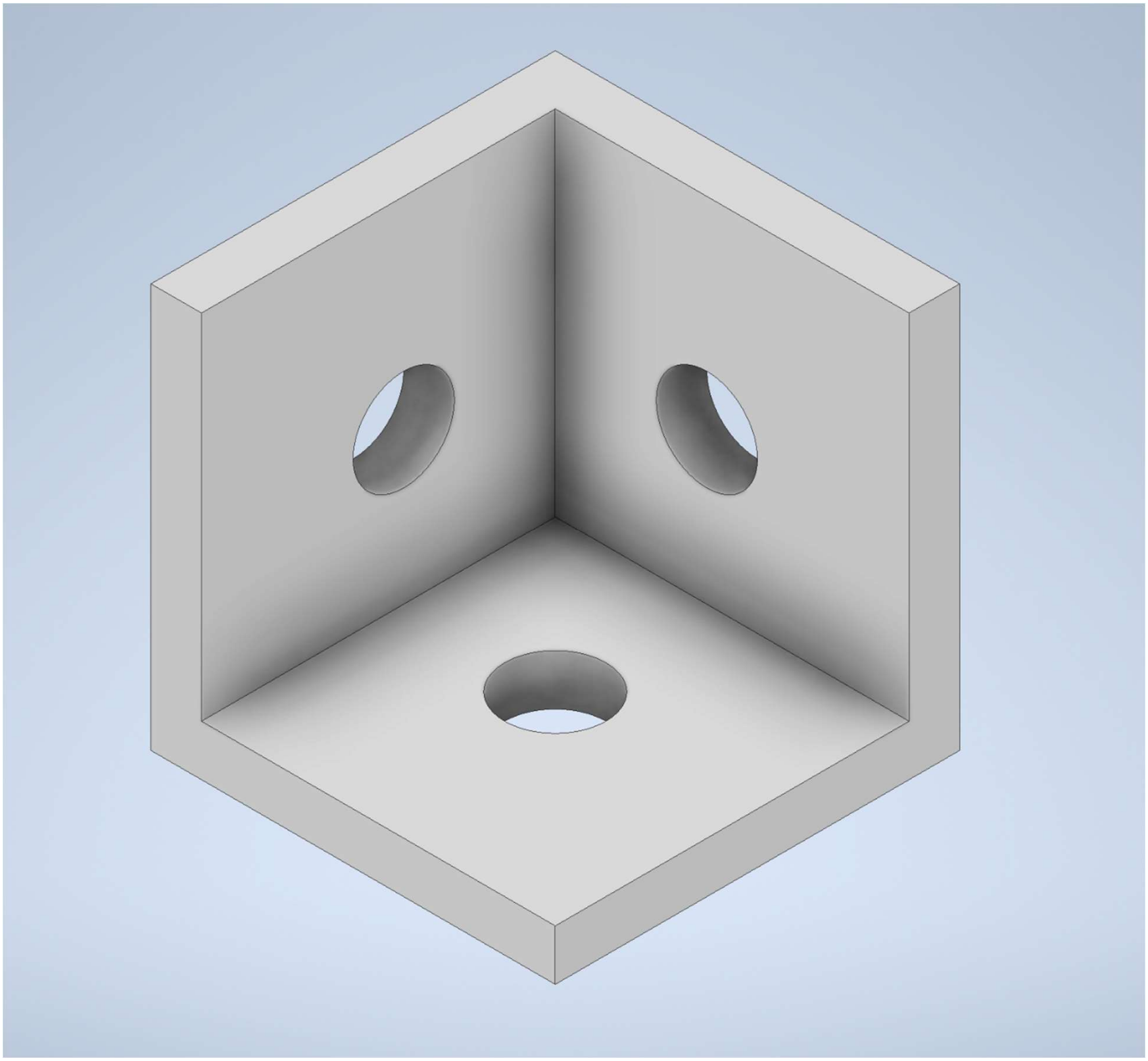


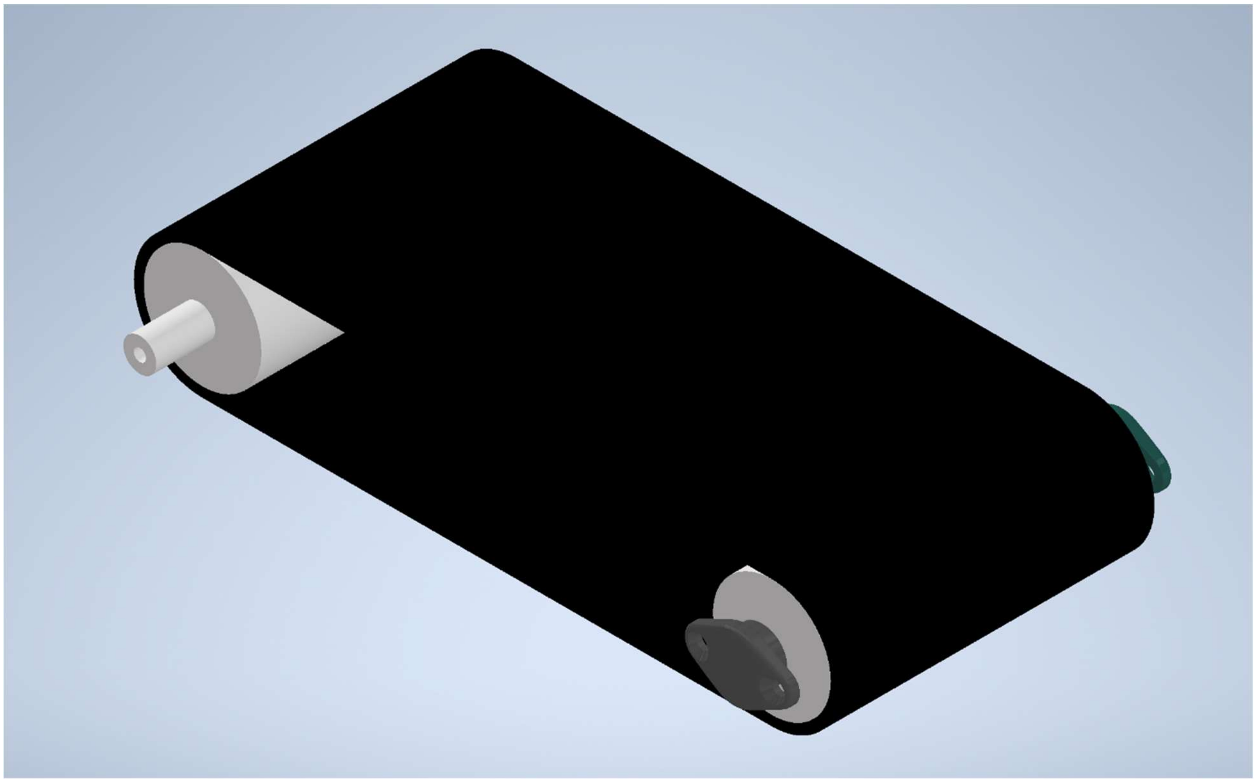


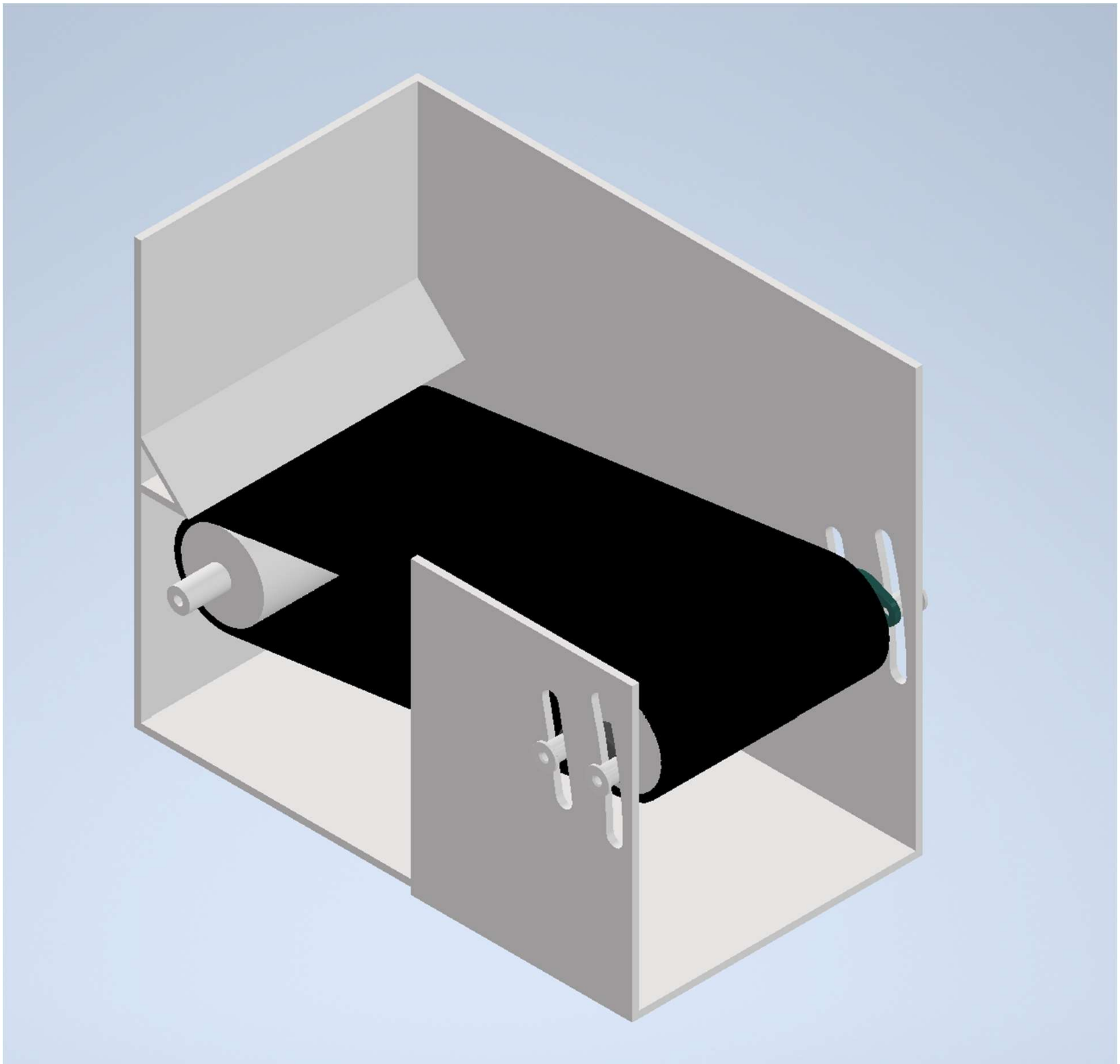


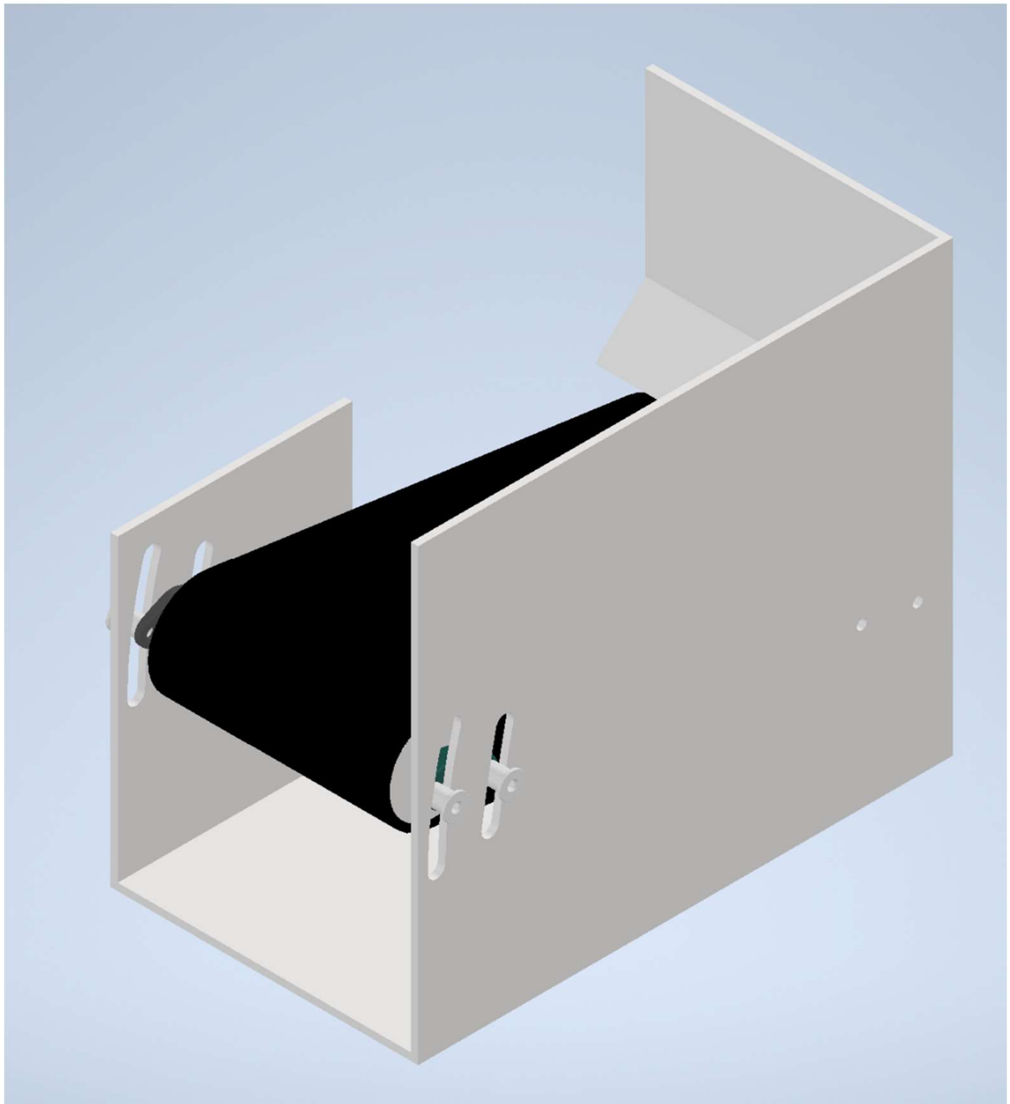


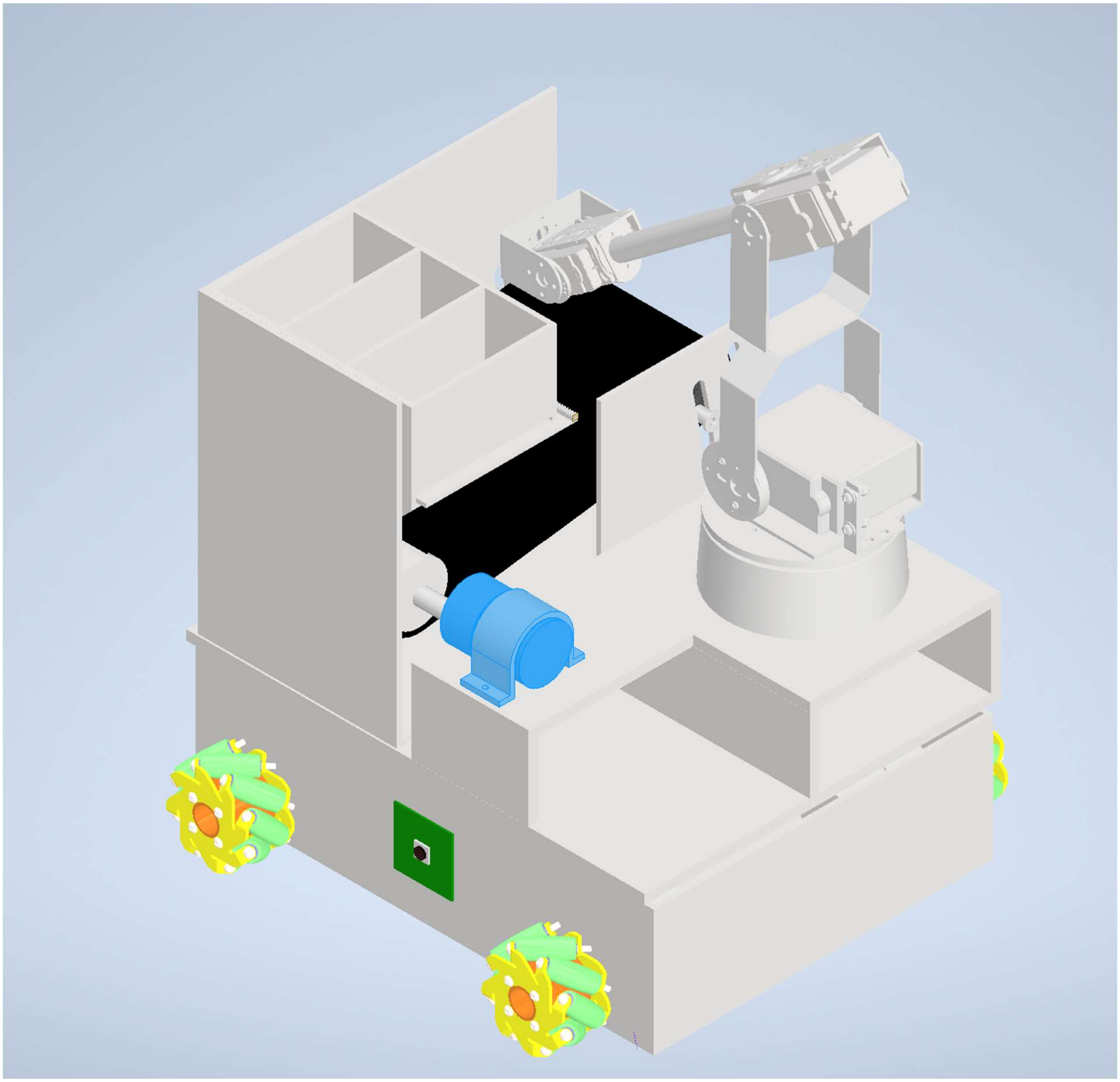










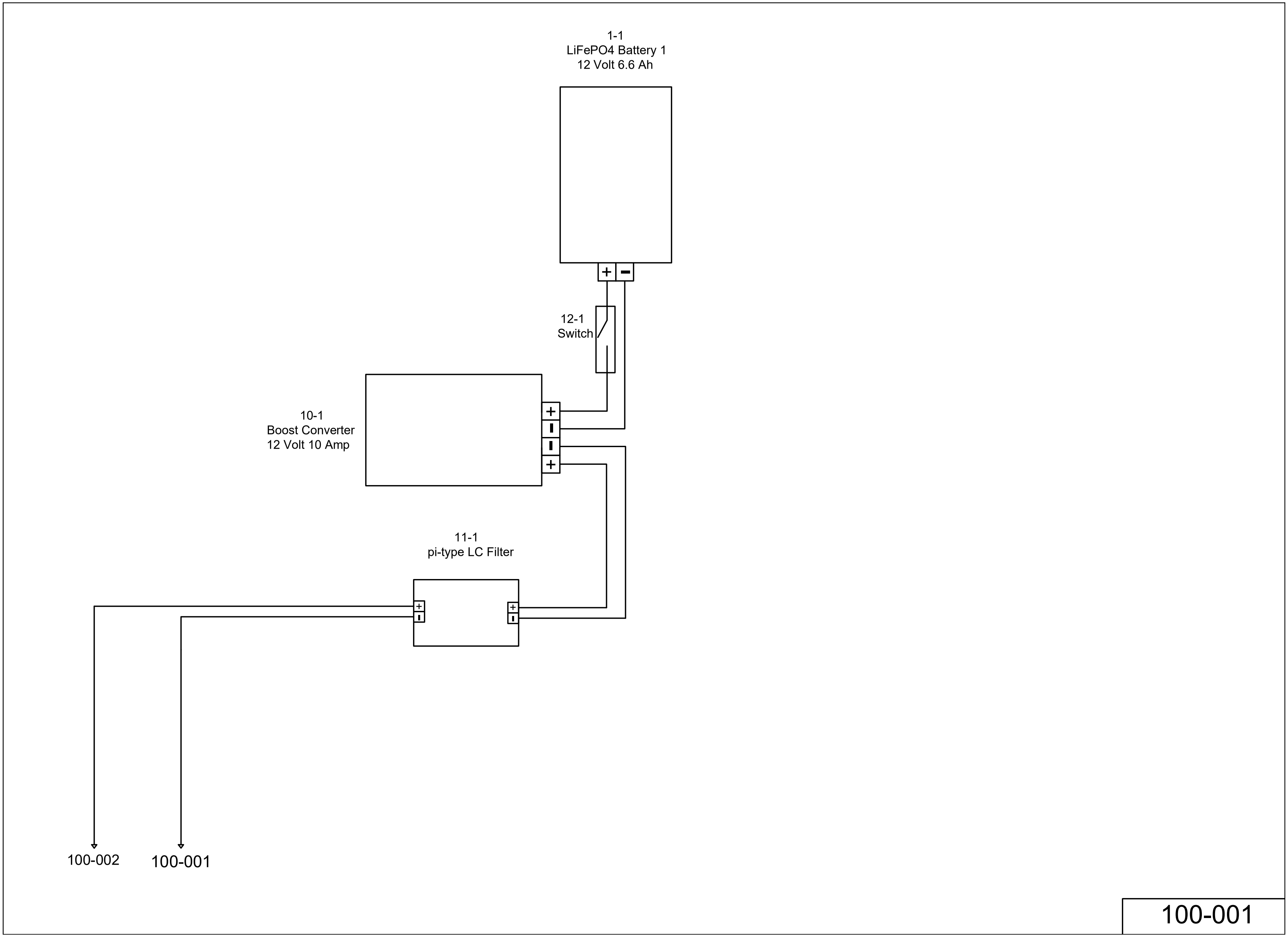


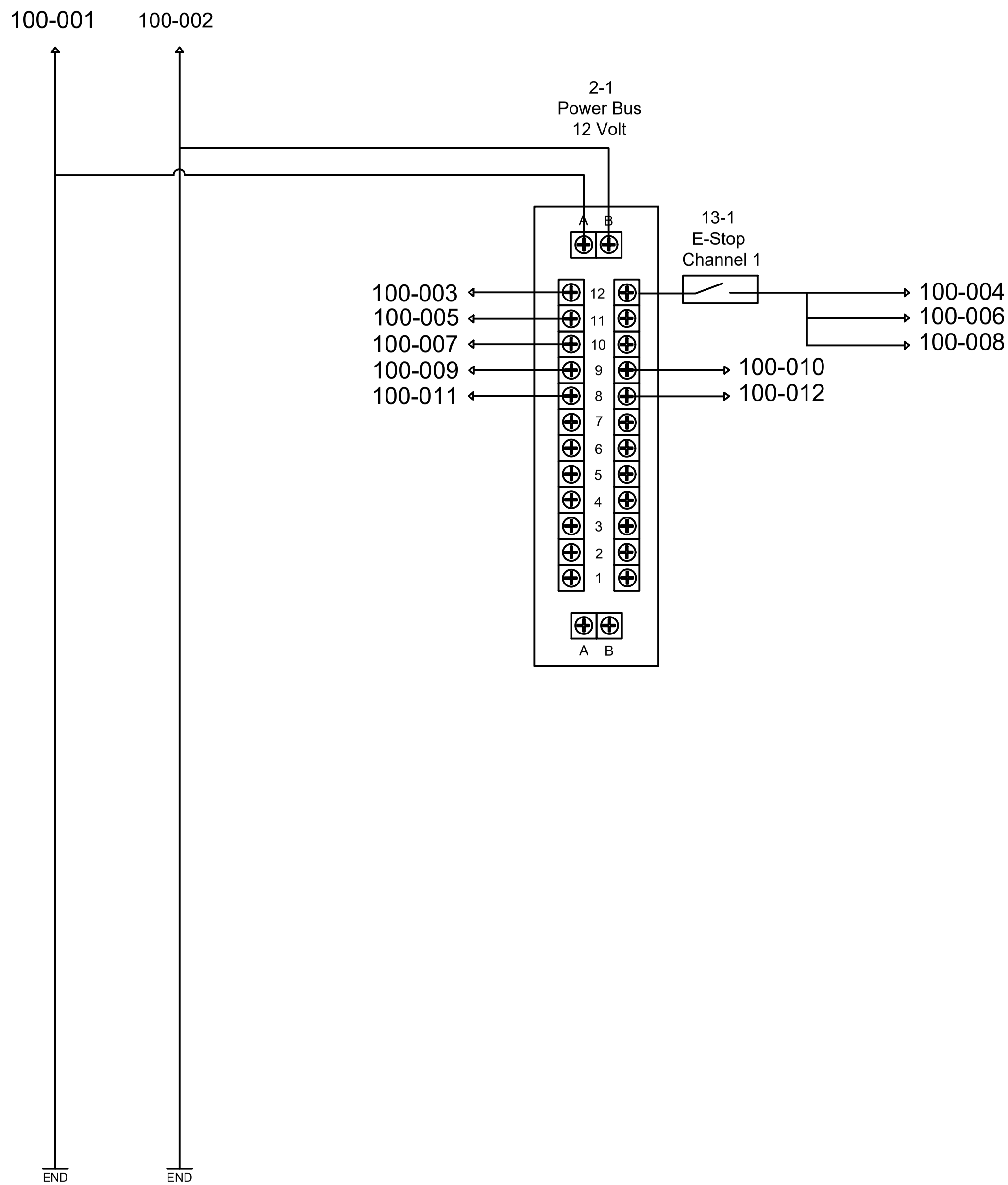
SouthEast Con Robotic Competition 2022 Robotic Drawing Package

000-001

#	Product Name	Quantity	#	Product Name	Quantity
1	LiFePO4 Battery 12 V 6.6Ah	1			
2	Power Bus	3			
3	Buck Convertor	2			
4	Raspberry Pi 4B	1			
5	USB-C cord to Raspberry Pi	1			
6	Adafruit 16-Channel 12-bit PWM/Servo Shield	1			
7	H-Bridge L298N Motor Drive Controller Board	3			
8	Arduino Mega 2560	2			
9	Arduino Mega 2560 Power Cord	2			
10	Boost Converter	1			
11	Pi-type LC filter	3			
12	Switch	2			
13	2-Channel Emergency Stop	1			
14	Line Following Array	3			

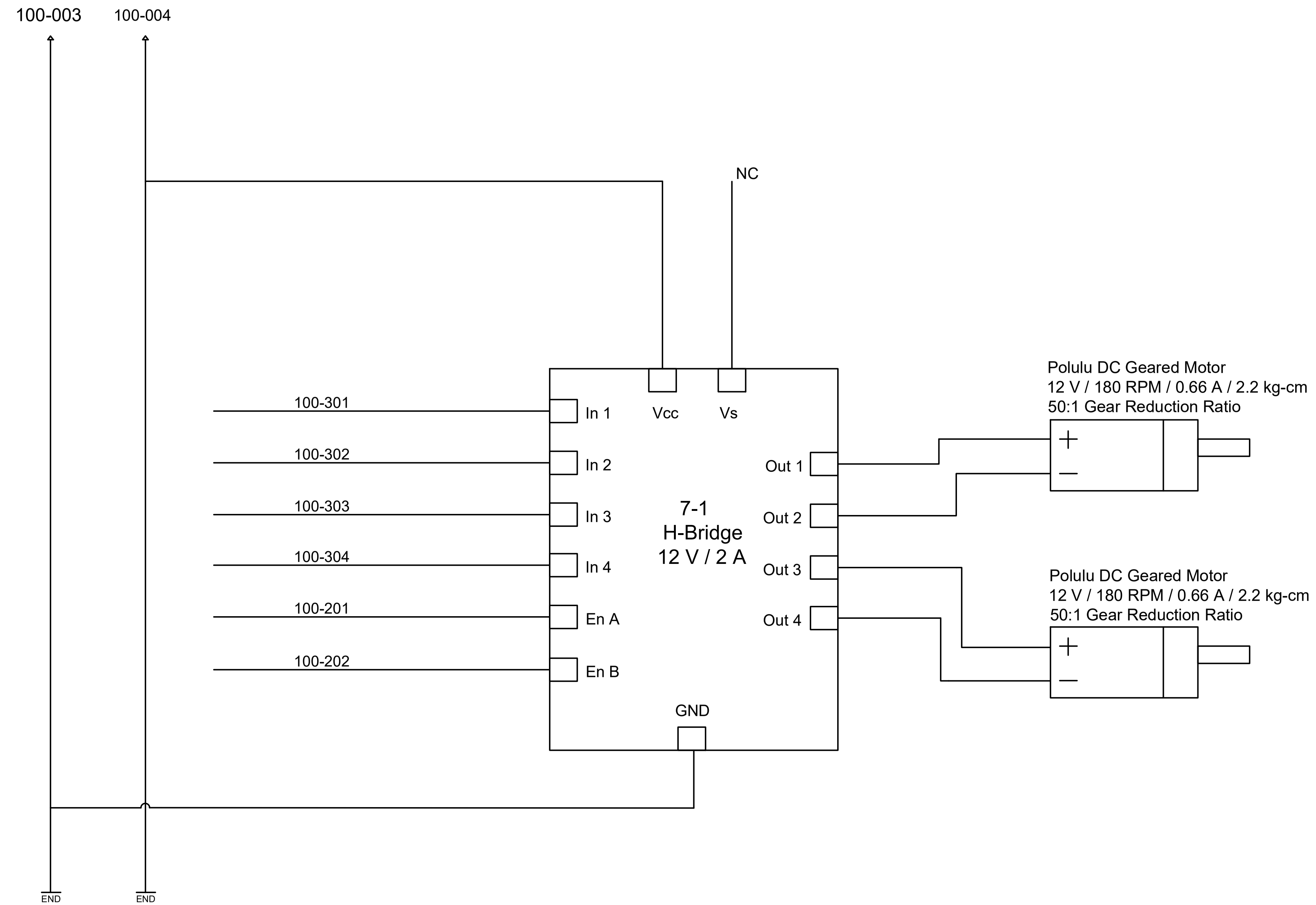
000-002


1-1
LiFePO4 Battery 1
12 Volt 6.6 Ah
12-1
Switch
10-1
Boost Converter
12 Volt 10 Amp
11-1
pi-type LC Filter
100-002
100-001
100-001



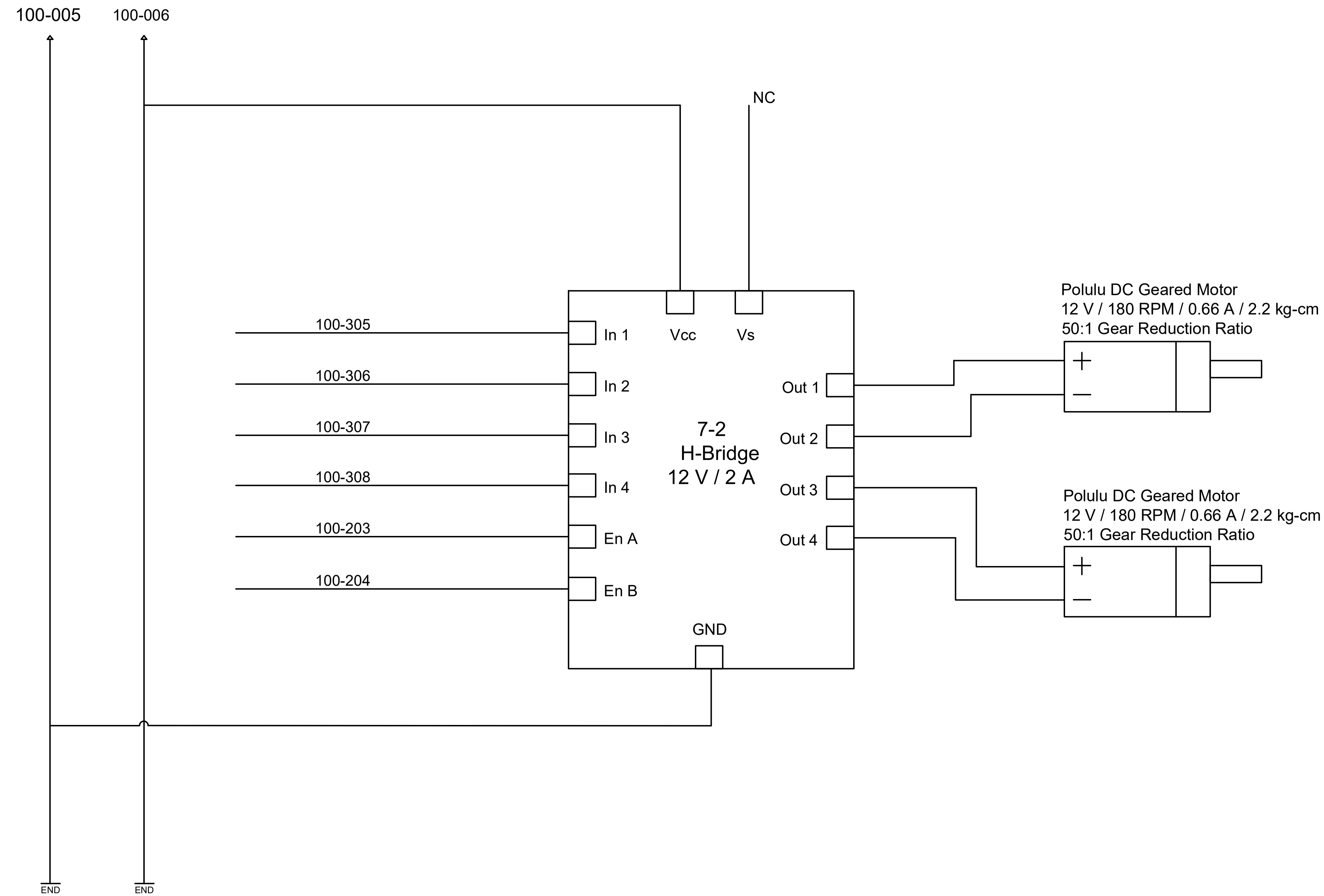
100-001
100-002
2-1
Power Bus
12 Volt
A
B
13-1
E-Stop
Channel 1
100-003
100-005
100-007
100-009
100-011
12
11
10
9
8
7
6
5
4
3
2
1
100-004
100-006
100-008
100-010
100-012
A
B
END
END

100-002

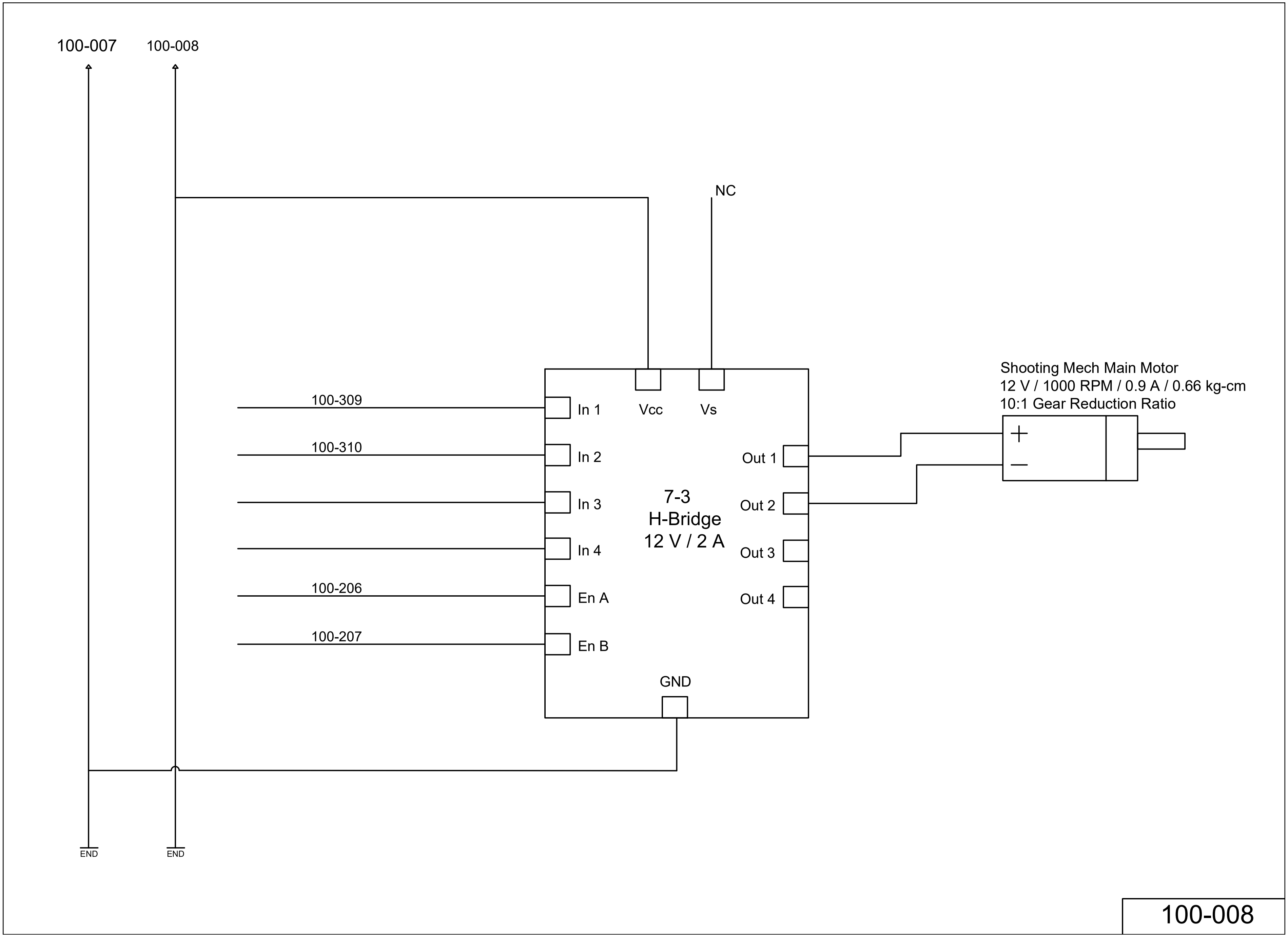


100-003
100-004
NC
100-301
100-302
100-303
100-304
100-201
100-202
In 1
In 2
In 3
In 4
En A
En B
Vcc
Vs
7-1
H-Bridge
12 V / 2 A
Out 1
Out 2
Out 3
Out 4
GND
Polulu DC Geared Motor
12 V / 180 RPM / 0.66 A / 2.2 kg-cm
50:1 Gear Reduction Ratio
Polulu DC Geared Motor
12 V / 180 RPM / 0.66 A / 2.2 kg-cm
50:1 Gear Reduction Ratio
END
END

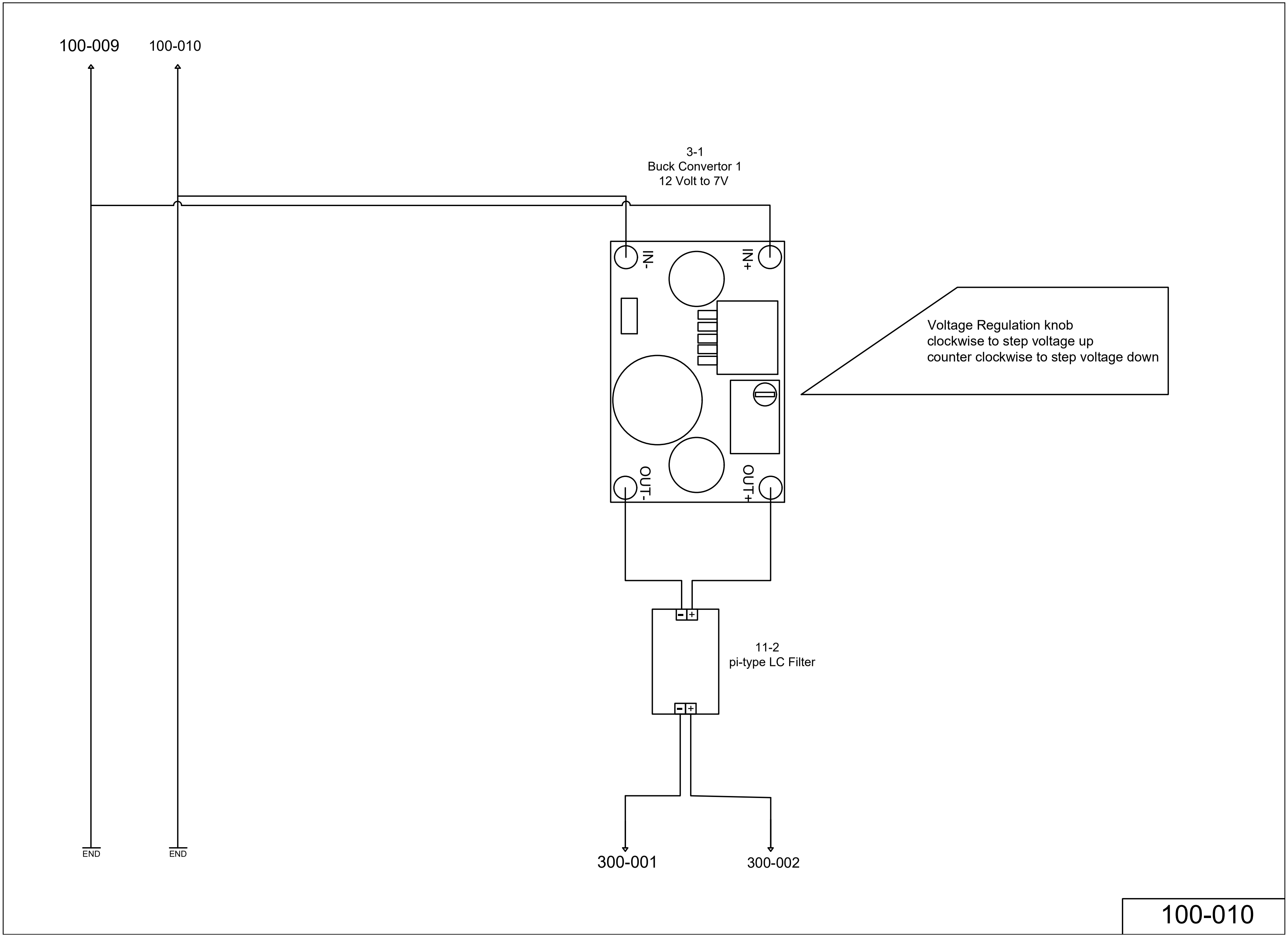
100-004

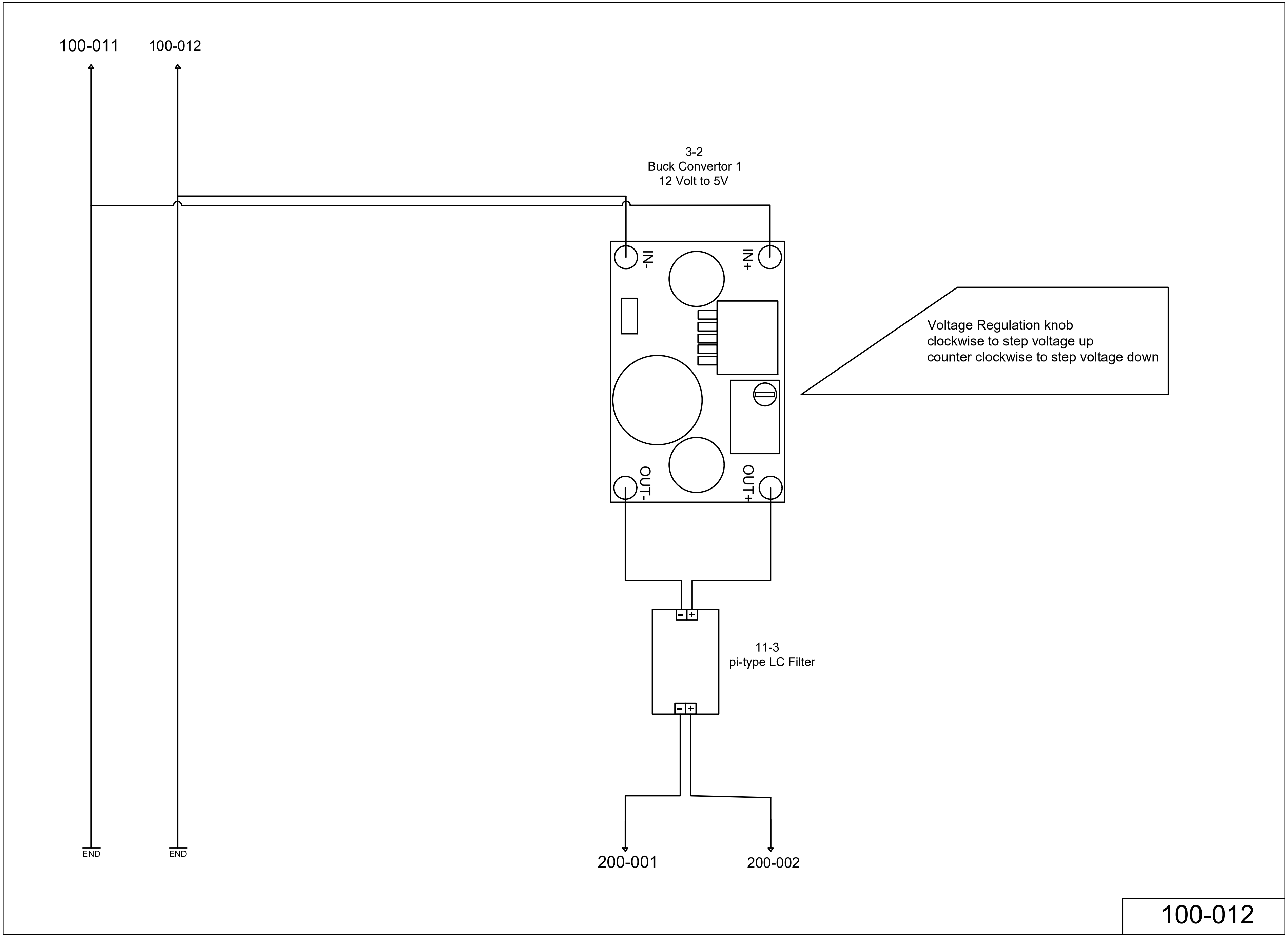

100-005
100-006
NC
100-305
100-306
100-307
100-308
100-203
100-204
In 1
In 2
In 3
In 4
En A
En B
Vcc
Vs
7-2
H-Bridge
12 V / 2 A
Out 1
Out 2
Out 3
Out 4
GND
Polulu DC Geared Motor
12 V / 180 RPM / 0.66 A / 2.2 kg-cm
50:1 Gear Reduction Ratio
Polulu DC Geared Motor
12 V / 180 RPM / 0.66 A / 2.2 kg-cm
50:1 Gear Reduction Ratio
END
END

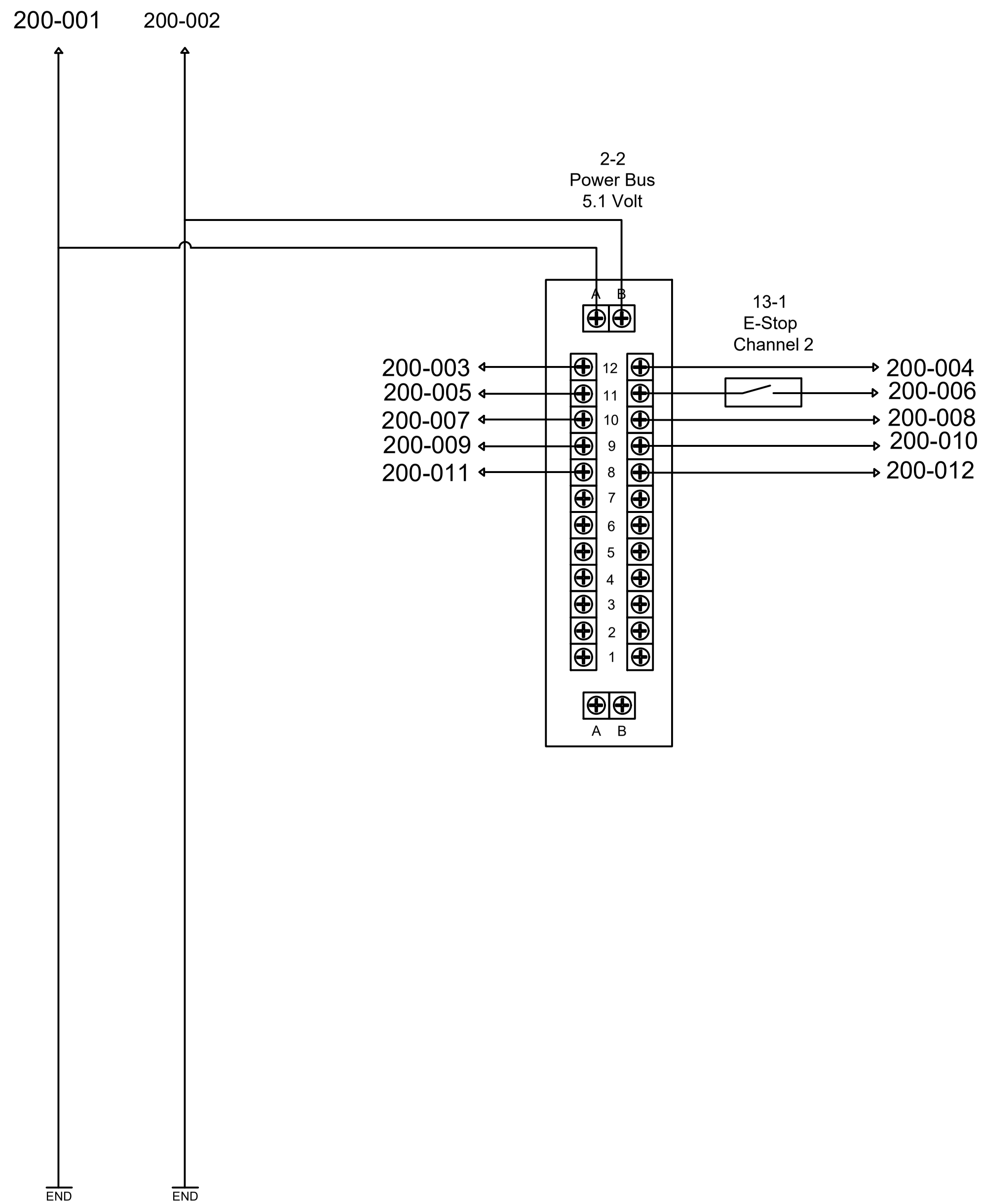
100-006



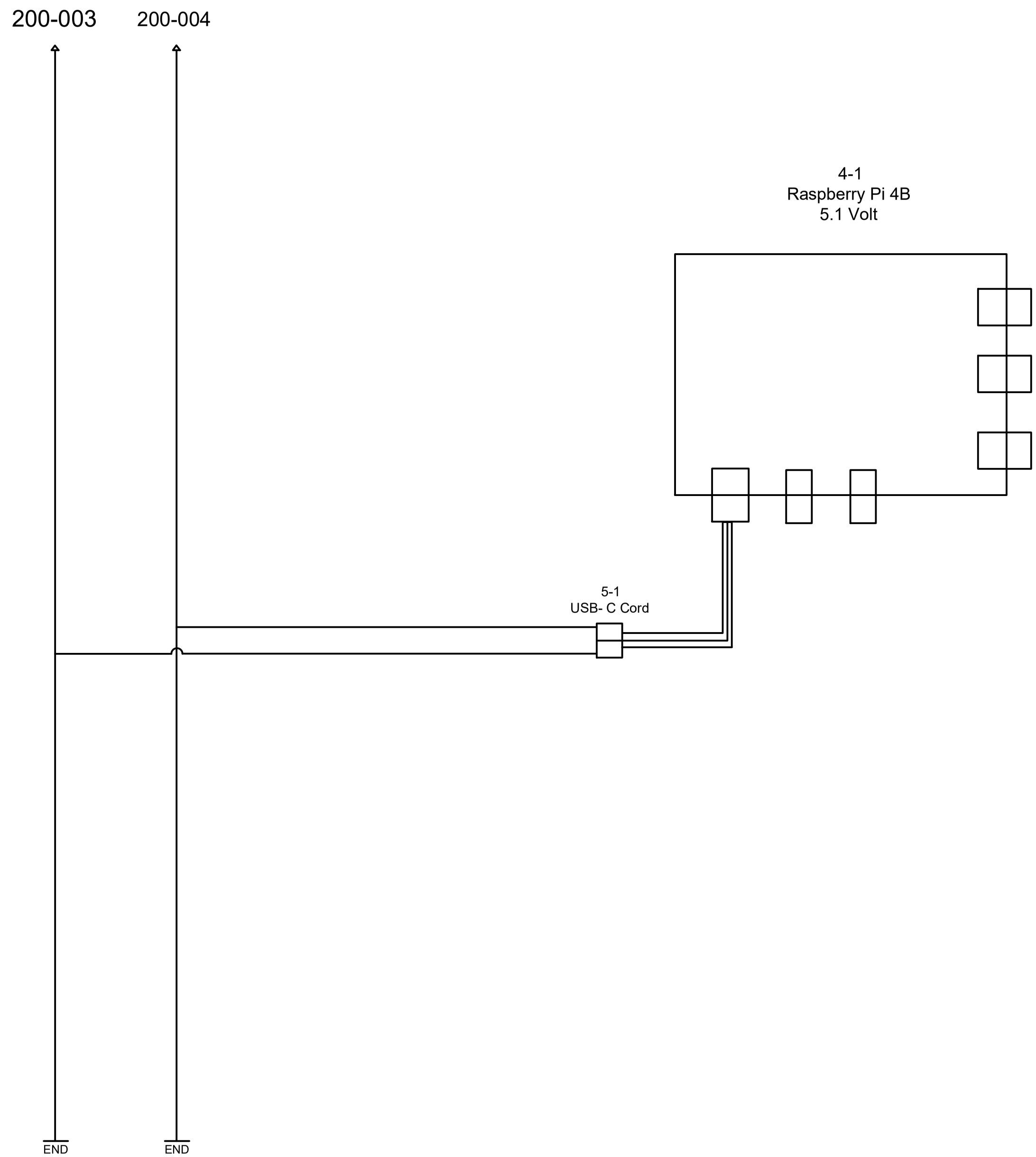
100-007
100-008
NC
100-309
100-310
100-206
100-207
In 1
In 2
In 3
In 4
En A
En B
Vcc
Vs
Out 1
Out 2
Out 3
Out 4
GND
7-3
H-Bridge
12 V / 2 A
Shooting Mech Main Motor
12 V / 1000 RPM / 0.9 A / 0.66 kg-cm
10:1 Gear Reduction Ratio
END
END
100-008


100-009
100-010
3-1
Buck Convertor 1
12 Volt to 7V
IN-
IN+
OUT-
OUT+
Voltage Regulation knob
clockwise to step voltage up
counter clockwise to step voltage down
11-2
pi-type LC Filter
END
END
300-001
300-002
100-010

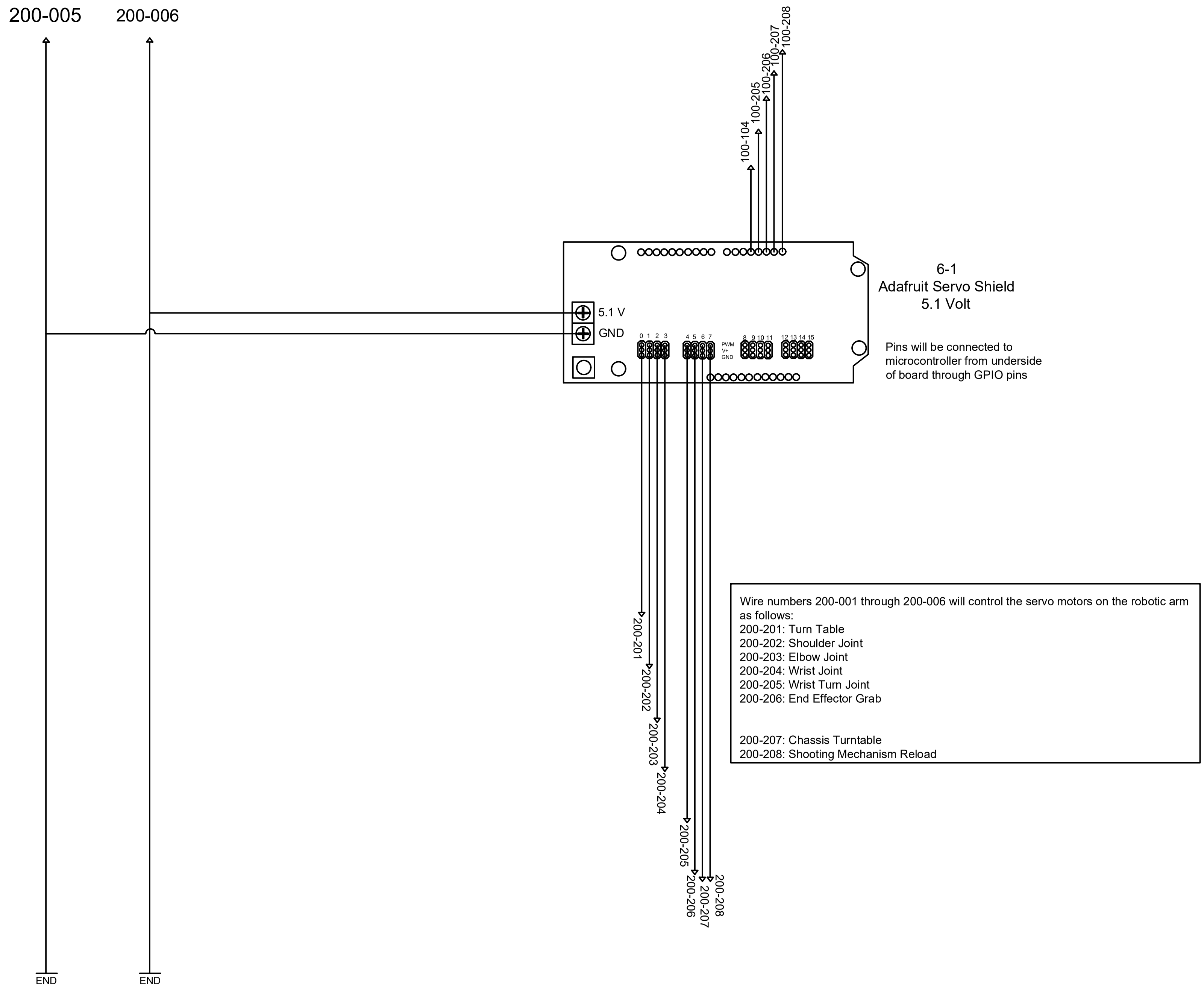

100-011
100-012
3-2
Buck Convertor 1
12 Volt to 5V
IN-
IN+
OUT-
OUT+
Voltage Regulation knob
clockwise to step voltage up
counter clockwise to step voltage down
11-3
pi-type LC Filter
END
END
200-001
200-002
100-012


200-001
200-002
2-2
Power Bus
5.1 Volt
A B
13-1
E-Stop
Channel 2
200-003
200-005
200-007
200-009
200-011
12
11
10
9
8
7
6
5
4
3
2
1
200-004
200-006
200-008
200-010
200-012
A B
END
END

200-002



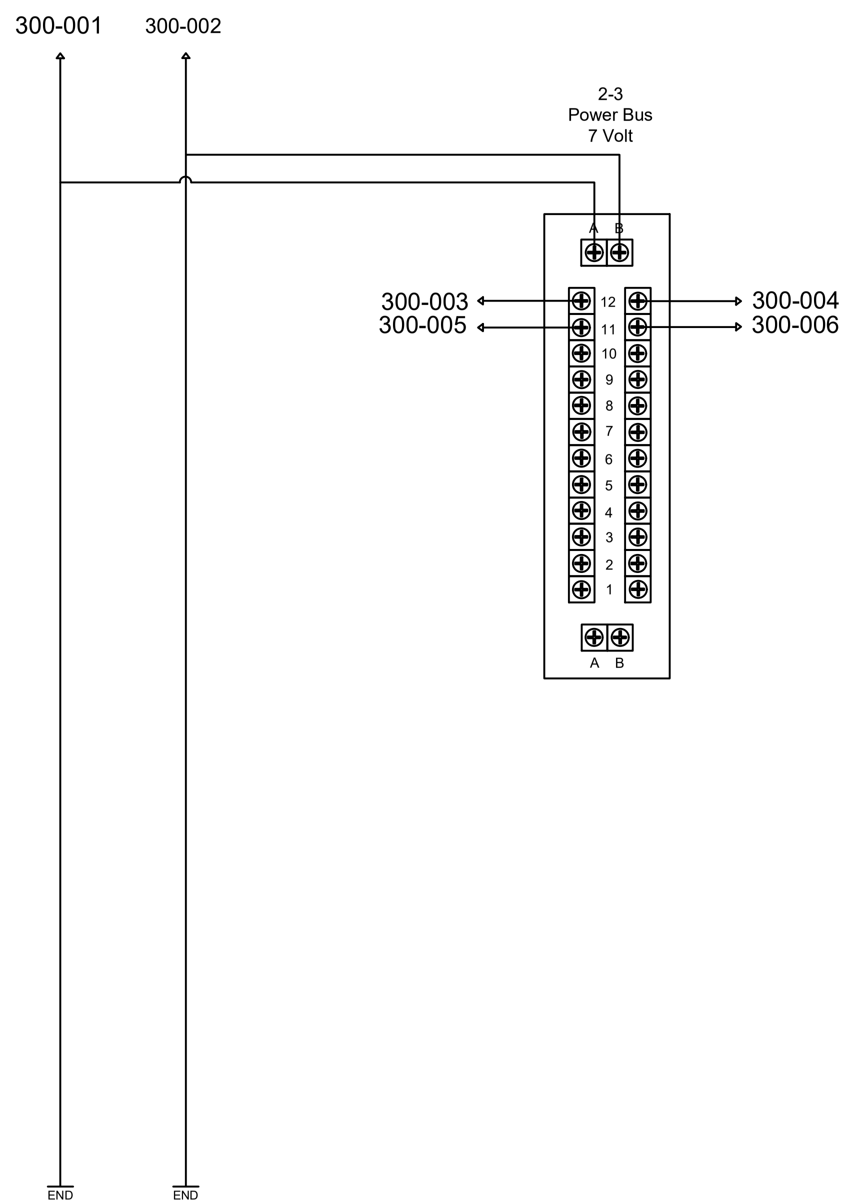
200-004


200-005
200-006
100-104
100-205
100-206
100-207
100-208
6-1
Adafruit Servo Shield
5.1 Volt
5.1 V
GND
0 1 2 3
4 5 6 7
PWM
V+
GND
8 9 10 11
12 13 14 15
Pins will be connected to
microcontroller from underside
of board through GPIO pins
200-201
200-202
200-203
200-204
200-205
200-206
200-207
200-208
Wire numbers 200-001 through 200-006 will control the servo motors on the robotic arm
as follows:
200-201: Turn Table
200-202: Shoulder Joint
200-203: Elbow Joint
200-204: Wrist Joint
200-205: Wrist Turn Joint
200-206: End Effector Grab
200-207: Chassis Turntable
200-208: Shooting Mechanism Reload
END
END

200-006

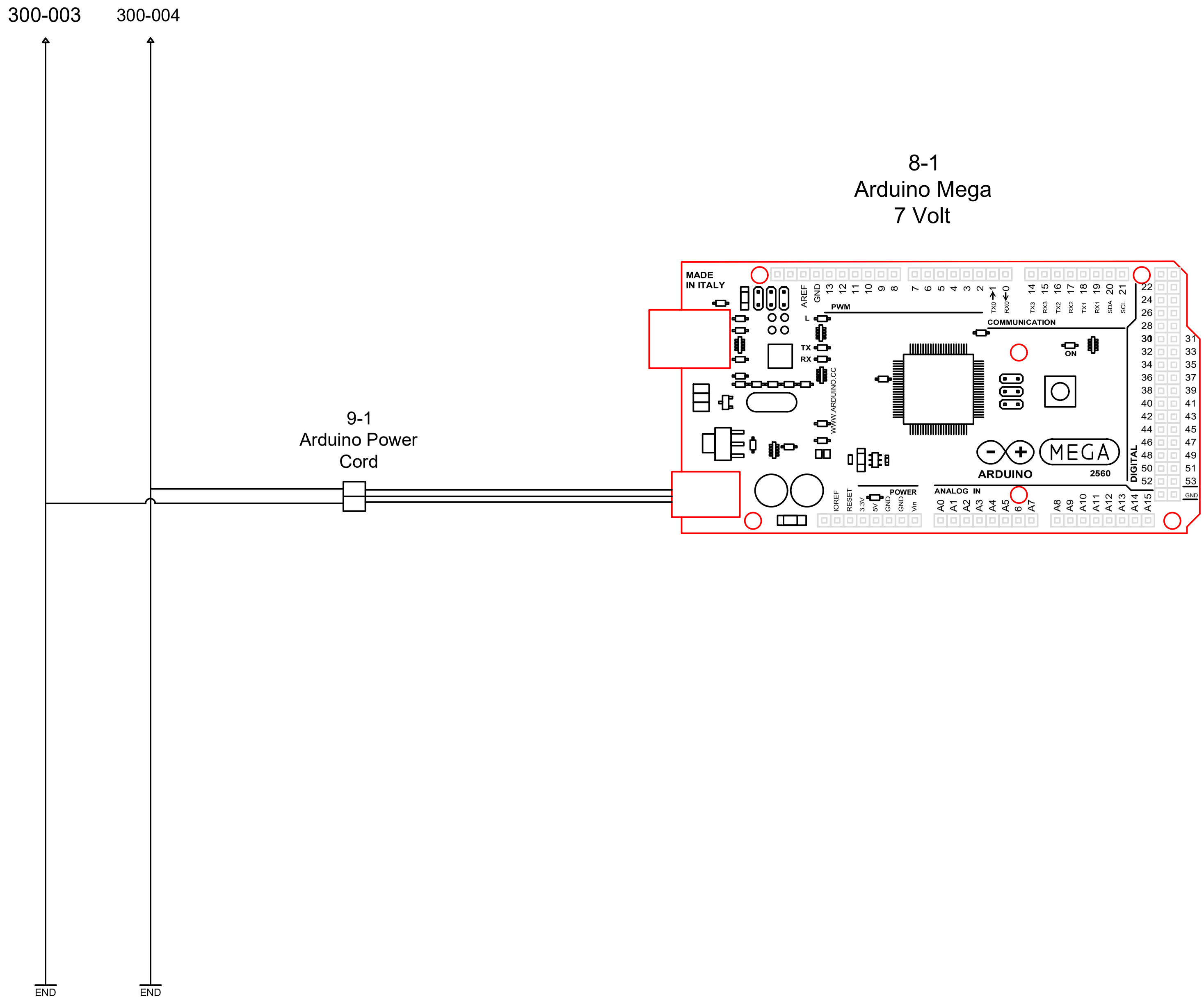
Wire numbers 200-001 through 200-006 will control the servo motors on the robotic arm as follows:
200-201: Turn Table
200-202: Shoulder Joint
200-203: Elbow Joint
200-204: Wrist Joint
200-205: Wrist Turn Joint
200-206: End Effector Grab
200-207: Chassis Turntable
200-208: Shooting Mechanism Reload

200-201

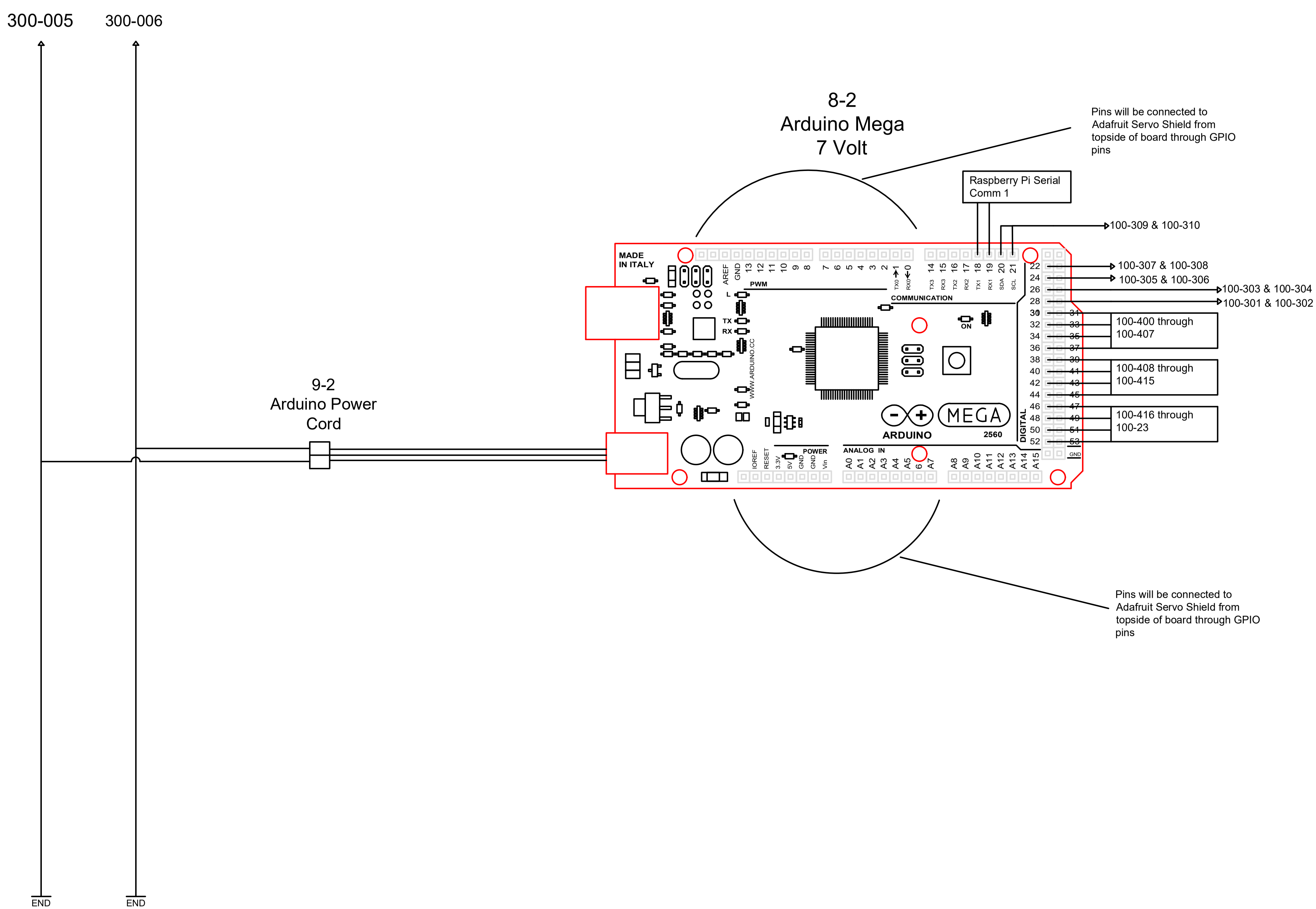


300-001
300-002
2-3
Power Bus
7 Volt
A B
12
11
10
9
8
7
6
5
4
3
2
1
A B
300-003
300-005
300-004
300-006
END
END

300-002

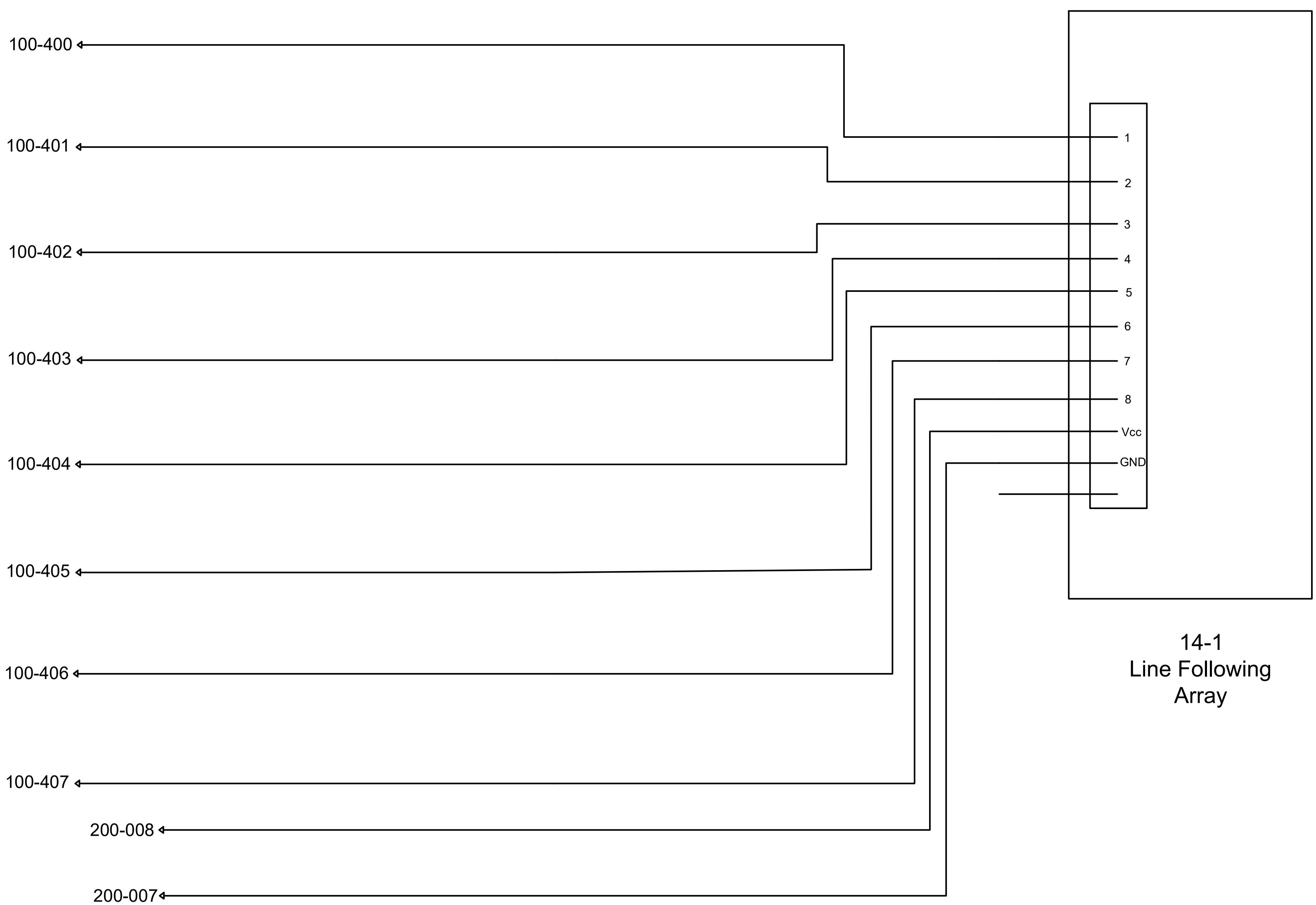

300-003
300-004
8-1
Arduino Mega
7 Volt
9-1
Arduino Power
Cord
END
END

300-004

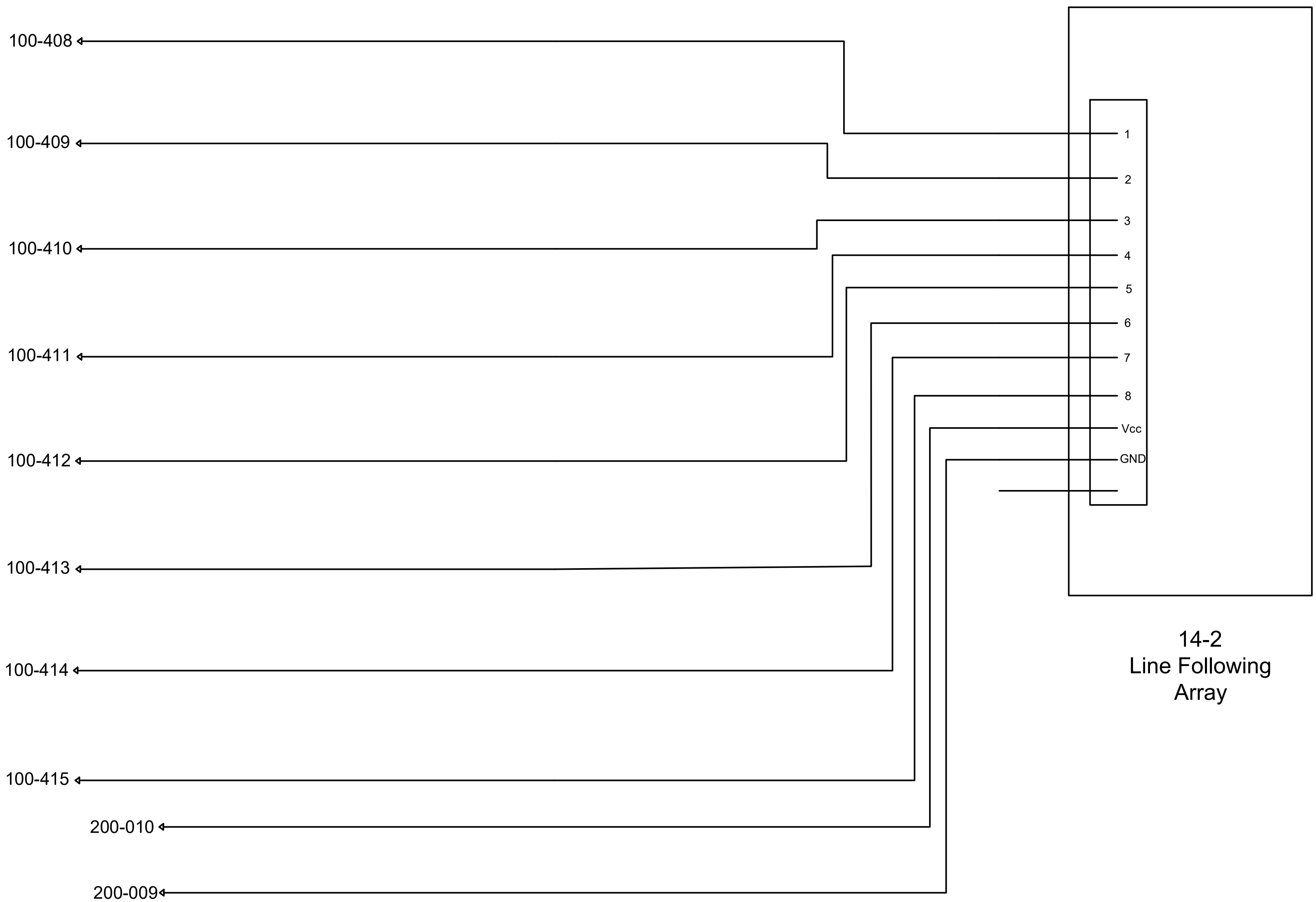


300-005
300-006
8-2
Arduino Mega
7 Volt
Pins will be connected to Adafruit Servo Shield from topside of board through GPIO pins
Raspberry Pi Serial Comm 1
100-309 & 100-310
100-307 & 100-308
100-305 & 100-306
100-303 & 100-304
100-301 & 100-302
100-400 through 100-407
100-408 through 100-415
100-416 through 100-23
9-2
Arduino Power Cord
Pins will be connected to Adafruit Servo Shield from topside of board through GPIO pins
END
END

300-006

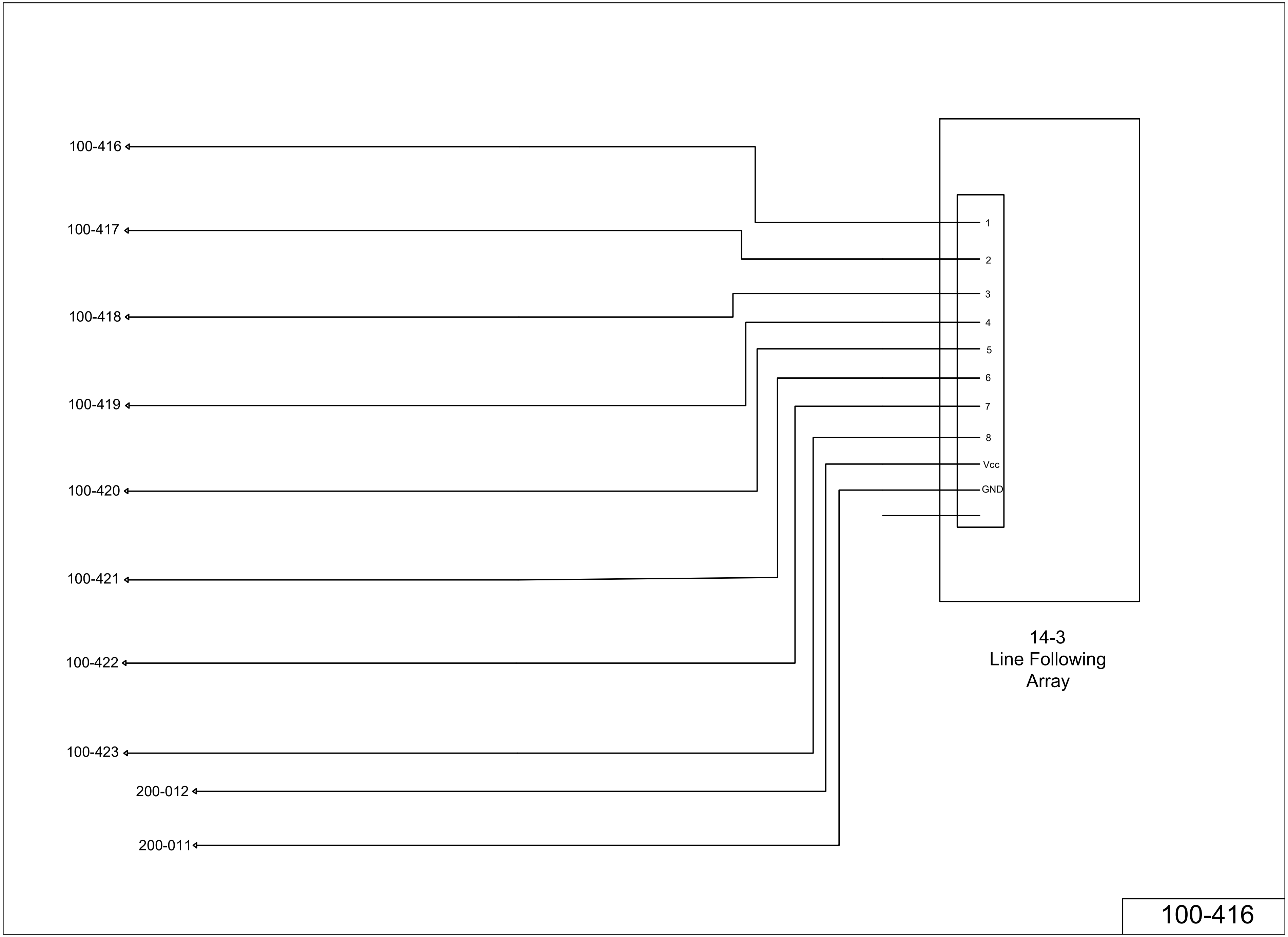

100-400
100-401
100-402
100-403
100-404
100-405
100-406
100-407
200-008
200-007
1
2
3
4
5
6
7
8
Vcc
GND
14-1
Line Following
Array

100-400



100-408
100-409
100-410
100-411
100-412
100-413
100-414
100-415
200-010
200-009
1
2
3
4
5
6
7
8
Vcc
GND
14-2
Line Following
Array

100-408


100-416
100-417
100-418
100-419
100-420
100-421
100-422
100-423
200-012
200-011
1
2
3
4
5
6
7
8
Vcc
GND
14-3
Line Following
Array
100-416