

Designing an Automated Material Handler for Agricultural Warehouse Applications

Design Team: Mark Fenn, Brendan Taedter

Instructors: Dr. Kevin Kilty, Professor Lawrence Willey

Faculty Advisor: Dr. Jeff Anderson

Sponsored by Plenty Unlimited, Inc.

May 7th, 2018

Table of Contents

Executive Summary.....	1
Project Background	2
Design Objectives and Specifications	2
Benchmarking	2
Design Concepts and Considerations	3
Functionality	5
New Knowledge	7
Design Factors	7
Design Process Overview	8
Test Results	13
Prototype Evaluation	14
Prior Art	15
Creativity and Innovation	15
Contributions of Each Member	16
Conclusions	16
Recommendations for Future Work	16
References	18
Appendix I: Usage Instructions	20
Appendix II: Full Arduino Code	22
Appendix III: SolidWorks Drawings	30
Appendix IV: Wiring Schematic	32
Appendix V: Parts List	33
Appendix VI: Torque Requirement Calculations	34
Appendix VII: Product Design Specifications (PDS)	35

List of Figures

Figure 1. Automated Guided Vehicle Market Forecast	3
Figure 2. Line Tracking Module with 3 IR Emitter/Receiver Pairs	6
Figure 3. Remote Control used to Operate Material Handler	6
Figure 4. LCD Screen at Startup	7
Figure 5. Small-Scale Prototype on Test Track	9
Figure 6. Full-Scale Prototype on Temporary Test Track	9
Figure 7. Completed Load-Carrying Assembly	10
Figure 8. Completed Bottom Chassis	11
Figure 9. Angled Aluminum Motor Mount.....	11
Figure A1. SolidWorks Drawing of Bottom Chassis	30
Figure A2. SolidWorks Drawing of Loading Platforms	31
Figure A3. Arduino Wiring Schematic	32
Figure A4. Free Body Diagram of a Wheel	34

List of Tables

Table 1. Summary of Design Considerations	5
Table 2. Line Tracking Logic Summary.....	6
Table A1. Parts List	33

Executive Summary

The objective of this project was to design and construct an automated cart, or material handler, with the ability to automatically carry heavy items to any location in Plenty Unlimited, Inc's agricultural warehouses. The material handler will make employee's jobs safer, less physically strenuous, and more efficient. The handler should also be able to pay for itself within one year based off a corresponding increase in labor efficiency.

Design requirements were supplied by Plenty Unlimited and can be summarized as follows: the handler must be able to stop at any designated location in the warehouse, must be able to be sent/summoned from any location, must be able to move forward and backward, and must be made of materials that are FDA approved as food-safe or food-grade. Because of strict warehouse cleanliness requirements, it was determined that the handler should utilize machine vision as its guide mechanism, rather than a mechanical system. Although there are several methods for machine vision, it was decided to guide the material handler by line-following technology due to its ease of implementation and adaptability to warehouse layout. The handler uses three infrared sensors on its undercarriage to keep it moving along a black line on the floor; desired destinations are designated by placing small gaps in the line. The design team elected to use an Arduino Mega microprocessor due to wide familiarity of the Arduino language as well as compatibility with many different sensors and other necessary electronic components. Materials used include high-density polyethylene (HDPE) for the load-carrying platforms, steel connecting rods for added height, standard rubber wheels, and various hardware and electrical components.

The design process essentially consisted of the construction of two prototypes. First, a small-scale prototype was constructed from various off-the-shelf components. This small vehicle allowed the design team to implement fully functional Arduino programming before building the full-scale model. The second prototype is the key product of this project. This version of the material handler uses many of the same electrical components as the small version, and the identical programming. The bottom chassis as well as the two load-carrying platforms are made of custom-cut HDPE and supported by steel connecting rods. Power is supplied by a rechargeable 12V lithium-ion battery and four 12V, 160rpm DC motors. This handler likely would not support the desired 500lbs due to deflection of the bottom HDPE chassis, however, Plenty Unlimited will be able to take this design and incorporate stainless steel or attach high-quality ball transfers to increase the handler's load capacity. Based on a budget of approximately \$700, and only considering the time workers will save, the material handler will pay for itself in 11 months. This is especially impactful when considering commercial automated material handlers cost roughly \$15,000 at the cheap end. Therefore, this material handler will not only be custom-tailored to Plenty's needs but will come at about 5% the cost.

Project Background

Plenty Unlimited is an indoor agricultural tech company and a pioneer in the vertical hydroponic farming industry. Plenty's farms are contained in 100,000ft² warehouses, meaning workers often must manually transport materials over distances of up to 400ft. Clearly, this method is time consuming, inefficient, and introduces safety concerns as workers are manually delivering heavy trays of produce across long walking distances. Currently there are large-scale industrial material handling systems for companies to reduce manual labor and improve efficiency; however, these systems are extremely cost-prohibitive to relatively new companies such as Plenty Unlimited and are riddled with non-essential functionality. However, if an automated material handler was designed with only the specific needs of Plenty Unlimited in mind, a cost-effective and efficient solution could be achieved.

Design Objectives and Specifications

Engineers at Plenty Unlimited approached the design team with an open-ended task: design and construct an automated load carrying mechanism. The relative ambiguity of the task allowed the design team to be creative and brainstorm various options that would fit the specific needs of Plenty Unlimited; however, Plenty engineers provided the following requirements to help guide the design process:

- The device must be able to deliver goods to any location within the warehouse.
- The device must be able to carry up to 500lbs of goods at a time.
- The device must not produce emissions/footprint.
- The device must be made of food-safe and food-grade materials.
- The device must be easy to disassemble and clean.
- The device must be easy and safe to operate.

Clearly, the design team was given a great deal of freedom of thought when determining the best solution to the design problem. Using the parameters listed above, the team came up with several designs which would satisfy the bare minimum requirements. These initial design ideas are discussed in detail in the upcoming "Design Concepts and Considerations" section, and **Appendix VII** contains the full Product Design Specification (PDS) document.

Benchmarking

Automated material handlers already exist, but are laden with non-essential functionality and therefore very expensive. Most commercial material handlers are only available by purchasing an entire fleet, which is helpful for large corporations, but prohibitive for upstart companies like Plenty Unlimited. **Figure 1** on the following page is a projection of the market for autonomous guided vehicles (AGVs) [1].

AGV MARKET FORECAST											2015 -2025	
	2015	2016	2017E	2018E	2019E	2020E	2021E	2022E	2023E	2024E	2025E	CAGR
Autonomous Guided Robotics (AGVs)												
Units Sold	19,000	33,250	49,875	69,825	97,755	127,082	165,206	206,507	258,134	309,761	371,713	34.6%
YY	50.2%	75.0%	50.0%	40.0%	40.0%	30.0%	30.0%	25.0%	25.0%	20.0%	20.0%	
ASPs	\$41,000	\$38,950	\$37,100	\$35,430	\$33,925	\$32,568	\$31,346	\$30,249	\$29,266	\$28,388	\$27,607	-3.9%
YY	1.2%	-5.0%	-4.8%	-4.5%	-4.3%	-4.0%	-3.8%	-3.5%	-3.3%	-3.0%	-2.8%	
Market Value (B)	\$0.78	\$1.30	\$1.85	\$2.47	\$3.32	\$4.14	\$5.18	\$6.25	\$7.55	\$8.79	\$10.26	29.4%
YY	52.0%	66.3%	42.9%	33.7%	34.1%	24.8%	25.1%	20.6%	20.9%	16.4%	16.7%	
Mkt Share	20.7%	26.7%	29.1%	31.0%	33.4%	33.4%	34.1%	34.2%	34.7%	34.4%	34.3%	

Source: Loup Ventures

Figure 1. Automated Guided Vehicle Market Forecast

Note that the current average selling price (ASP) is estimated to be over \$35,000 per vehicle. For perspective, the material handler designed and constructed for this project cost \$700 while keeping all essential functionality.

Design Concepts and Considerations

The following is a discussion of the three most important design considerations encountered throughout the project. Refer to **Table 1** on page 5 for a summary of all noteworthy design considerations.

Choosing a Mechanism

After receiving the general design requirements from Plenty, the design team thought of three possible mechanisms that would accomplish the necessary tasks. The first proposed solution was to have an electric vehicle with a mechanical guide system such as a rail or pulley. This is a widely implemented method and would not be difficult to tailor to Plenty’s needs. The issue that arises with this type of mechanism is that it presents a tripping hazard and/or impedes the strict cleanliness requirements of Plenty’s agricultural warehouses.

The second proposed solution was again an electric vehicle, but this time using camera-based machine vision. This would certainly be a more elegant solution, but also came with a steep increase in complexity that the design team did not feel qualified to implement effectively. Furthermore, with the increase in complexity, it was anticipated that there would be an accompanying increase in nagging functionality problems.

The third proposed solution was a simpler version of the machine vision vehicle. In this solution, the electric vehicle would use infrared (IR) sensors to follow a black line around the warehouse floor. Line tracking technology already exists, and would be particularly useful in that it is adaptable to any warehouse layout, i.e. the line could be made to go wherever it is desired to have the material handler be able to travel. This method is also cheaper and simpler to implement than camera machine vision. The only issue foreseen would be if the black line were to become soiled or damaged, it may cause the handler to fail. However, with Plenty’s use of hydroponic farming and strict cleanliness requirements, this potential problem is believed to be unlikely. For these

reasons, the design team elected to use line-following technology as the basis of the automated material handler.

Material Selection

Once the design team had decided what kind of vehicle to design, it was necessary to determine the optimal materials from which to construct it. Stainless steel is a popular choice for food-safe applications, however can be both prohibitively expensive and heavy. For this reason, plastics were given close consideration. Although there are many kinds of food-safe plastic on the market, high-density polyethylene (HDPE) was deemed to be the best for this application. HDPE is cheaper and lighter than stainless steel, and is more durable, wear-resistant, machinable, moisture and chemical resistant, and easy to clean relative to other plastics with comparable strength/cost ratios. HDPE does compromise on strength (yield strength of 5ksi as compared to 31ksi for stainless steel) [2] but is still sufficiently strong for the intended application of simply carrying goods in a warehouse.

Because HDPE is food-safe, it is used primarily for the load-carrying platforms, as these are the surfaces most likely to contact the produce. Galvanized steel was chosen as the material for the supporting rods, as it is food-safe except when in contact with highly acidic foods such as fruits [3]. It will be necessary to change this material to aluminum or stainless steel on future iterations, but galvanized steel was deemed acceptable for the prototype as a means of cost reduction. All materials that will be at locations on the cart where they will not contact the produce may be only food-grade, meaning they are rated as safe to be in the vicinity of food, but not necessarily in direct contact. These materials most notably include the rubber wheels, aluminum-housed motors, and all electrical components.

Driving Systems

The first important consideration regarding the material handler's electric propulsion was whether each of the four wheels should have its own motor. By having a motor on each wheel, the handler could turn around nearly in place by rotating the wheels on one side the opposite direction as the others. The wheels could then be fixed in relation to the handler, as turning would be accomplished by simply controlling the direction of the motors. Furthermore, the torque rating of each motor could be lower than if only two motors were used. The alternative would be to have a motor on each of the rear wheels and a servo motor for steering on each of the front wheels. This is a more complex solution, and likely to be more expensive due to the need for much higher-torque motors as well as high-quality servos. The design team also anticipated issues arising if the handler needed to turn a sharp corner, since using this method would significantly decrease the handler's turn radius. For these reasons, it was decided to use four motors.

The next consideration was the balance between motor torque, motor speed (rpm), and wheel diameter. From the initial calculations, it was seen that the minimum torque rating required for each motor is 0.143 ft*lb (see **Appendix VI** for detailed torque calculations). Ideally, the motor speed should be such that the handler is able to move just faster than walking speed, or about 1.5 m/s. Using Bemonoc brand 0.18 ft*lb, 160rpm, 12V DC electric motors and 5-inch diameter wheels, a speed of 1.1 m/s is obtained. The team initially selected 220rpm motors, which would

have resulted in the desired 1.5 m/s, but these motors were backordered and would not have arrived in time to be attached and tested.

Table 1. Summary of Design Considerations

Item	Consideration	Other Constraints	Possible Solutions	Final Decision
Guide Mechanism	Must be self-guiding (i.e. no mechanical guide system)	Cost, complexity, customizability	Machine vision using camera, line following with IR sensors	Line following with IR sensors
Primary Material	Must be food-safe and strong enough to transport 500lbs	Cost, weight, durability	Stainless steel, food-safe plastics	Food-safe plastic (specifically HDPE)
Microprocessor	Must support motor driver, IR module, proximity sensor, LED display, IR receiver	Cost, customizability, user-friendliness	Arduino, RaspberryPi	Arduino
Platform supporting rods	Handler should support up to 500lbs, but not be excessively heavy itself	Weight, cost, ease of assembly/disassembly	Internally threaded connecting rods (x12), externally threaded rods (x6)	Externally threaded rods
Electric propulsion	Motor(s) must be powerful enough to drive a fully-loaded handler	Cost, weight	One or two large motors, four smaller motors	Four smaller motors
Motor speed / wheel size	Attempt to find the best combination of motor RPM/torque and wheel diameter. Handler must move at nearly walking pace	Torque, cost, voltage	High RPM motor with smaller wheel, vs. lower RPM motor with larger wheel	160RPM motor, 5" diameter wheels
Battery	Battery should be rechargeable, powerful enough to operate handler for entire work day	Cost, battery specs	Different sizes of batteries	12V, 20AH, \$56 rechargeable Li-Ion battery

Table 1, above, shows a complete account of all noteworthy design considerations encountered during this project. Of special notice are those considerations that did not receive a full discussion, such as the choice of microprocessor, platform supporting rods, and battery.

Functionality

The material handler utilizes an Arduino Mega microprocessor and IR line tracking technology to navigate a warehouse. The current design uses three IR sensor/emitter pairs mounted on the underside of the bottom chassis (**Figure 2**) to orient itself on a 2 inch wide black line on a light-colored floor. IR light travels from the emitter to the floor. If the floor beneath a given sensor/emitter pair is light-colored, the IR sensor will receive feedback; otherwise, the sensor

receives no feedback as all IR light is absorbed by the black line. In order for the material handler to keep moving forward along the line, all three sensor/emitter pairs must be above the black line. A summary of the material handler’s driving logic is listed in **Table 2** below.

Table 2. Line Tracking Logic Summary

Left Sensor	Middle Sensor	Right Sensor	Function
On	On	On	Go Forward
Off	On	On	Adjust Right
On	On	Off	Adjust Left
Off	Off	Off	Count Room or Stop

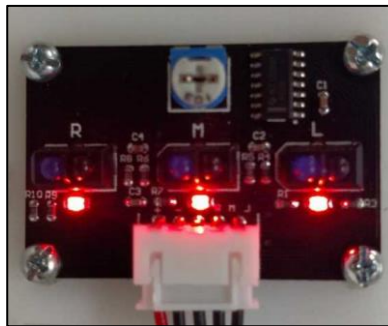


Figure 2. Line Tracking Module with 3 IR Emitter/Receiver Pairs

The material handler will navigate the warehouse by following a black line that loops around the warehouse’s central hallway. Each room within the warehouse is indicated by a 2-inch-long break in the track. The user will input a destination via remote control (**Figure 3**) and the handler calculates which direction of travel will provide the shortest path to the destination. The material handler then orients itself with the optimal direction of travel and begins to move along the line. The handler counts breaks in the line until it reaches its destination and stops.



Figure 3. Remote Control used to Operate Material Handler

Two key safety features are implemented into the programming to ensure employee wellbeing. The material handler uses an ultrasonic proximity sensor to detect obstacles, including humans, in its path. If the proximity sensor is triggered, the handler will stop until the obstacle is removed. Additionally, if the material handler malfunctions and gets off the line for more than 0.5 seconds, it is programmed to stop moving and enter “Manual Mode” in which the user can use the remote to move the handler back onto the line. Refer to **Appendix I** for complete usage instructions.



Figure 4. LCD Screen at Startup

In addition to the basic functionality required for normal operation, the material handler includes some extra features to improve the overall user experience. An LCD screen (**Figure 4**) mounted on the bottom chassis displays the handler's direction of travel, current position, and destination; the screen is useful for troubleshooting if the handler is not behaving as expected. The current design of the handler is approximately 25" tall, which allows for easy loading and unloading. The top platform is equipped with a rubber mat to prevent the load from slipping while the handler is in motion. With the exception of the bottom chassis, the height of each platform is completely adjustable, allowing for Plenty Unlimited to customize the design day-to-day to fit their exact needs. Possible design orientations include many closely-spaced platforms for carrying large flat trays or widely-spaced platforms with pull out drawers for larger goods or supplies.

New Knowledge

This project relied less on the development of new knowledge, and more on the integration of existing technologies to produce the desired final product. To the design team, however, much of the technicalities of line tracking was completely new. Line tracking has existed for some time now, but to the knowledge of the design team, has never been used for an application exactly like this project. Therefore, it was necessary to conduct many tests while attempting to make line tracking a viable solution to the design problem, which ultimately led to discoveries relating to optimizing line width, IR receiver sensitivity, break length, and the sharpness of curves. Ultimately the goal was to make the handler travel as smoothly as possible with all the desired functionality.

Design Factors

This section provides a summary of how the design of the material handler addresses several important factors.

Cost and Profitability

The importance of producing a material handler that is as cheap as reasonably possible cannot be overstated. If cost were not an issue, Plenty Unlimited could simply purchase their own fleet of material handlers and not bother with designing and producing their own. Because the material handler is only able to pay for itself by increasing labor efficiency, it is critical that its production cost is as low as possible, particularly when considering the goal is to have the handler pay for itself within only one year. A full Return on Investment (ROI) analysis can be found on page 44. The easiest way to keep the cost down is through material selection. The current design

incorporates HDPE as a means of avoiding the use of stainless steel, which is very costly (nearly 2.3 times as costly as HDPE) [3].

Safety

Because the material handler requires human interaction and will be used in warehouses where workers constantly move about, safety is a critical aspect of the design. During construction of the full-scale prototype, it was ensured that all sharp edges were filed until appropriately rounded, and that all electrical connections were secure and sufficiently far from where the operator's hand would be. The handler's programming includes two safety features: an ultrasonic proximity sensor stops the handler if it detects an obstacle, as well as a feature built into the programming which stops the handler after 0.5 seconds of motion without detection of the line (i.e. the handler is off the line rather than passing over a "room gap"). As a final safety measure, there is an easily accessible power switch on the battery pack which can be turned off if the programming unexpectedly fails.

Sustainability

A feature of the material handler's design is that it is made entirely from commonly available materials and hardware. None of the materials used will stop being produced any time soon, and on the off chance this did happen, every piece can be replaced with something similar. The simplicity and adaptability of the design ensures that Plenty Unlimited will be able to produce similar if not identical material handlers for the foreseeable future.

Impact on environment/society

One of the initial stipulations of the design must be that it produces no emissions, largely because it will operate in agricultural warehouses. All materials are recyclable, including the HDPE sheets. Its impact on society will admittedly be very indirect, however it is intended to greatly improve the quality of life of the Plenty Unlimited warehouse workers. As discussed previously, the current situation relies on the workers manually transporting heavy loads over distances of up to 400ft. Allowing the material handler (or material handlers) to conduct these tasks would save the worker's backs and allow them to spend their workday more productively.

Design Process Overview

Once the line tracking solution was deemed viable, the design team elected to construct a small-scale prototype and test track of black duct tape to test all coding and automation features (**Figure 5**). This prototype was constructed entirely from off-the-shelf components, which were not necessarily food-safe or food-grade, as it is too small to carry a load and was only used to test the functionality of the automation. Once this prototype was functioning correctly, a full-scale prototype was constructed (**Figure 6**). This larger version did incorporate food-safe and food-grade materials where necessary, in addition to other off-the-shelf parts and electronic components taken directly from the small prototype.

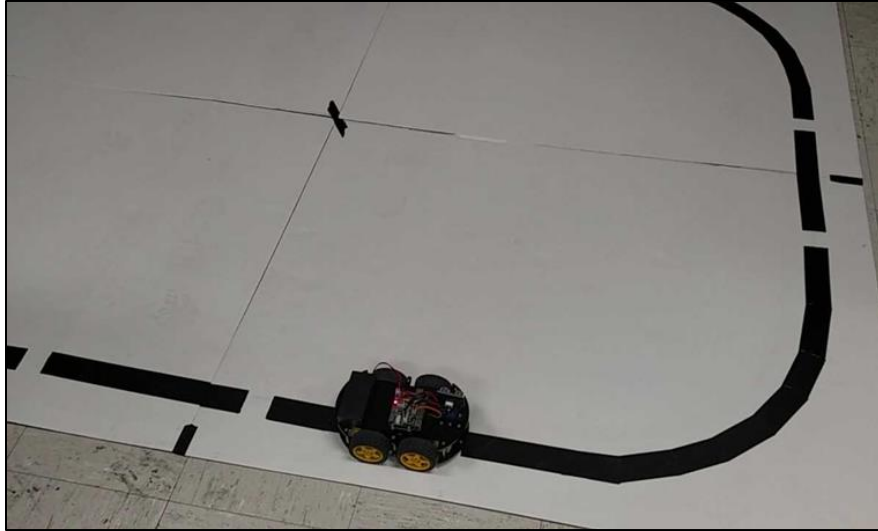


Figure 5. Small-Scale Prototype on Test Track

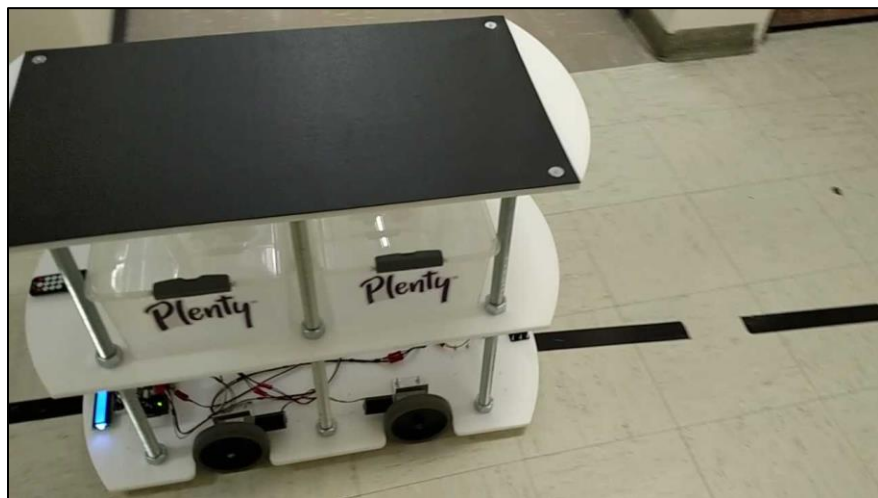


Figure 6. Full-Scale Prototype on Temporary Test Track

Construction of Small-Scale Prototype and Test Track

The small-scale prototype was constructed from an Elegoo Smart Robot Car kit purchased by the design team on Amazon.com. The kit contained all the supplies the design team needed to successfully test the automation component of the material handler including an Arduino Uno microprocessor, motors and wheels, a line tracking module, a remote, and an ultrasonic proximity sensor. Assembly instructions were provided by Elegoo Inc. Although the kit provided a Bluetooth module for the prototype, the design team deemed Bluetooth capabilities as out of the current project scope. If desired, Bluetooth capabilities can easily be implemented in future iterations of the material handler. The design team replaced the Arduino Uno microprocessor with an Arduino

Mega processor to increase memory and provide additional IO pins, giving the user the ability to add additional sensors or outputs. The larger processor allowed for installation of an LCD display, as seen previously in **Figure 4**. The LCD display was especially useful in troubleshooting during the testing phase of the small-scale prototype.

The programming of the small-scale prototype combined parts of sample code provided with the Elegoo Smart Robot Car kit with original programming created by the design team. Sample code was provided for many components of the prototype including line-following, ultrasonic proximity sensing, and IR remote input. The sample codes were integrated and customized by the design team to ultimately meet the automation requirements of the material handler. The full Arduino code can be found in **Appendix II**. A test track was also constructed for the purpose of evaluating this prototype. The track was simply made of four pieces of white poster board with a circuit of black duct tape placed around the outside.

Construction of Full-Scale Prototype

Once the small-scale prototype was functioning as desired, a full-scale prototype was constructed. This version of the material handler is comprised of two main sections: the load-carrying assembly and the bottom chassis. Refer to **Appendix V** for the complete full-scale prototype parts list.

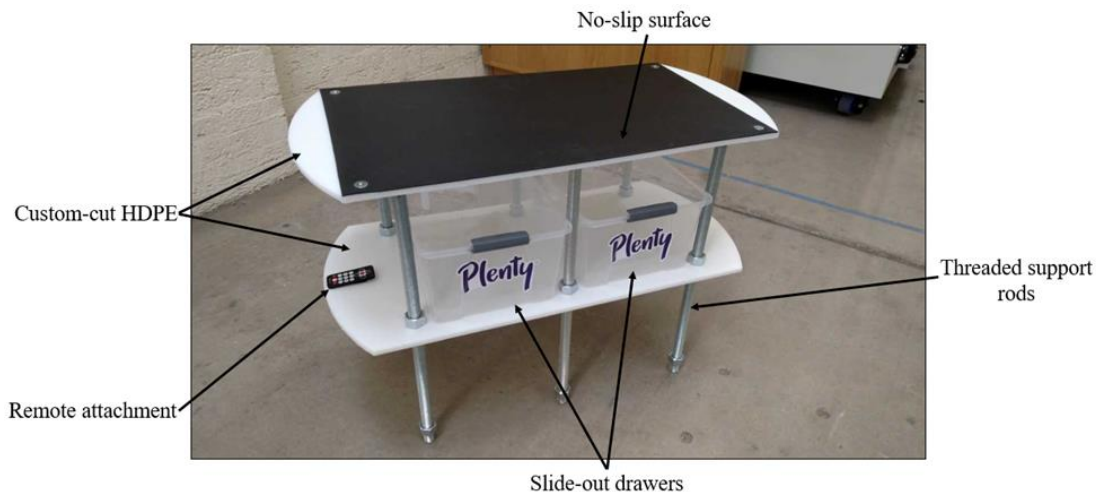


Figure 7. Completed Load-Carrying Assembly

The load-carrying assembly, as depicted in **Figure 7**, is a structure of 3/8” thick HDPE platforms and 7/8” diameter galvanized steel supporting rods. This is the section of the material handler where goods are loaded for transportation. The bottom chassis, as shown in **Figure 8**, is another sheet of HDPE which carries all the essential electronics and is the platform on which the load-carrying assembly rides. The bottom chassis can run independently of the load-carrying assembly, which is helpful when troubleshooting.

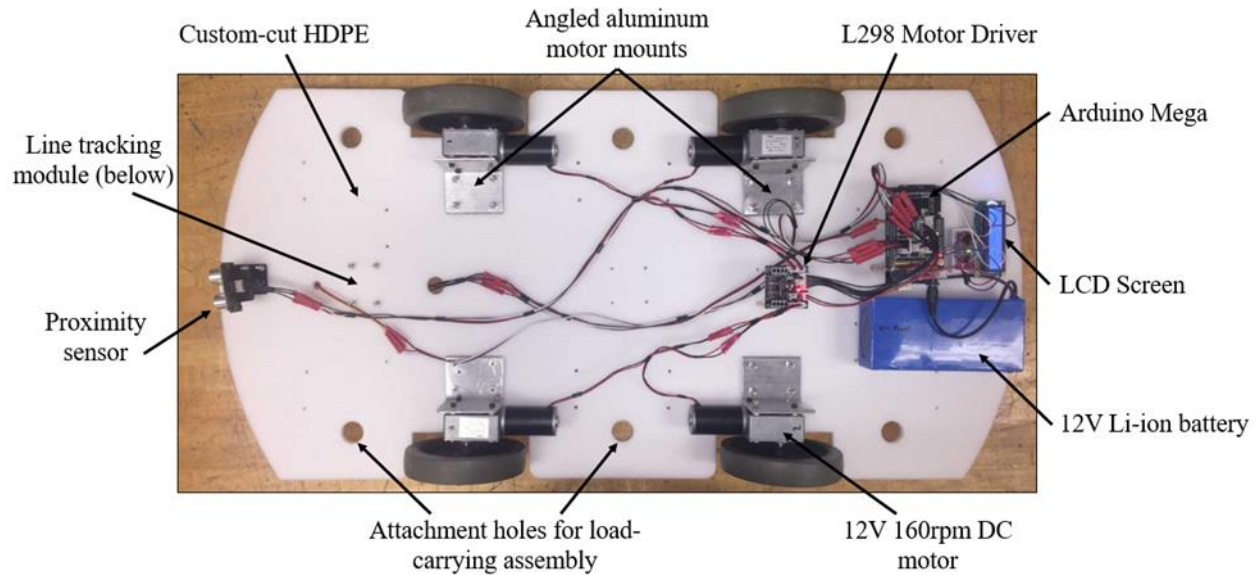


Figure 8. Completed Bottom Chassis

The first step in the construction of the full-scale prototype was to cut out the customized HDPE platforms. This was done according to the SolidWorks drawings found in **Appendix III** using the UW College of Engineering Shop’s high-pressure water jet. The water jet also cut the spaces for the wheels, the holes for the support rods, the holes for the proximity sensor wires and line tracking module wires to come up through, and the rectangular mounting hole for the proximity sensor.

The next step was to mount the motors to the bottom chassis (see **Figure 9** below). To do this, the team cut a bar of 2” by 2” by 1/8” angled aluminum into four sections, each 2.5” long.

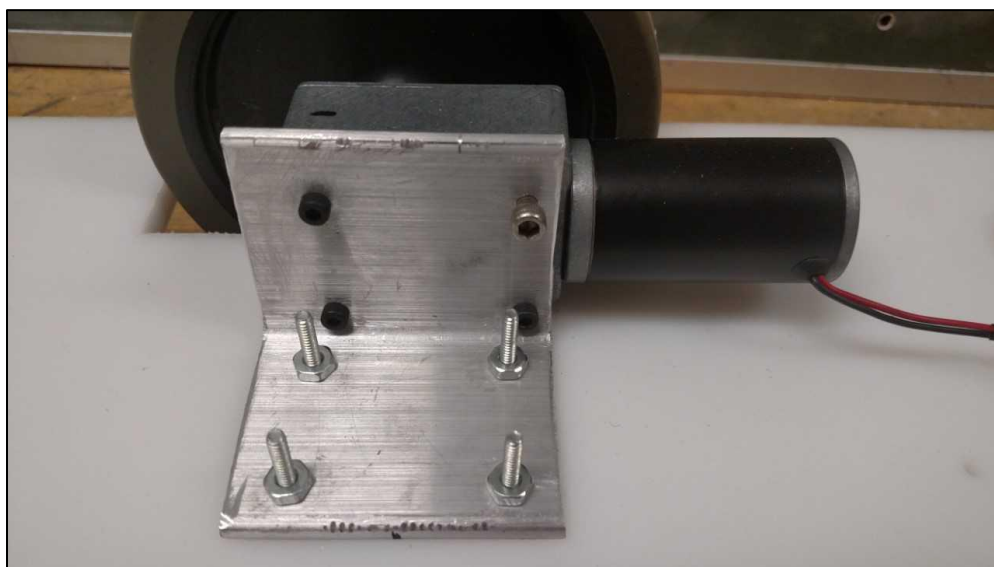


Figure 9. Angled Aluminum Motor Mount

Using a drill press, four holes were then drilled into one side of the angled aluminum, one hole near each corner, approximately 3/8" from the sides. These holes should be just large enough to accommodate 6-32 NC hardware. On the other face of the angled aluminum, four holes were drilled such that they lined up with the four holes on the backside of the motors, and such that the motor is no more than 1/8" above the chassis surface. Because the holes on the motors were initially unthreaded, it was necessary to tap three of them such that they accommodate 6-32 NC bolts. The fourth hole, being larger than the other three, was tapped such that it would fit 8-32 NC bolts. The larger hole is easily identifiable as being the one with a metal protrusion around it. The corresponding hole on the motor mount was also tapped to fit 8-40 bolts. At this point, the motors were fastened to the mounts. To fasten the mounts to the chassis, the team simply centered the motor shaft in the wheel well and marked locations on the HDPE through the holes already drilled in the mount. All holes drilled into the HDPE can be easily made with a powered hand drill. Now the motors can be fully mounted to the bottom chassis.

After the motors have been affixed, the wheels can be attached to the motor shafts. The wheels ordered by the design team have 5/16" diameter round shaft holes (0.3125"). The motors were said to have 8mm (0.315") diameter D-shaped shafts, such that the wheels could be press-fit onto the motor shafts. However, the motor shafts are closer to 6mm in diameter, meaning a temporary solution had to suffice. The team elected to fill the wheel hole with hot glue and press it onto the shaft, thus creating a smaller D-shaped hole inside of the round wheel hole. This method is clearly not robust and is not recommended for future iterations of the design. It did, however, work well enough to allow for testing of the full-scale prototype.

The electronic components, including the Arduino Mega, LCD, L298 motor driver, and proximity sensor were mounted using hardware from the Elegoo Smart Robot Car kit. The proximity sensor should have its hole already cut out using the water jet. The locations of the other components are not critical, as long as the IR receiver on the Arduino is visible to the remote, the LCD is visible by the operator, and the battery power cord can reach the Arduino port. The motor driver and Arduino are elevated about 2" from the chassis using small internally threaded connecting rods from the Elegoo kit, as those components can get relatively hot after prolonged use. The battery is attached to the chassis by four squares of Velcro to allow for easy removal for charging. The battery should be located as far to the rear as possible, such that the power switch is easily accessible in an emergency. The line tracking module should be centered side-to-side on the underside of the chassis and located somewhere in between the front support rod holes and the front wheels (more tests are necessary to determine the optimal front-to-back location). The IR sensors should be approximately half an inch above the floor when the bottom chassis is alone, and should deflect approximately 1/8" when the load-carrying assembly is attached. The height of the module can be adjusted by adding a spacer between the module and the chassis. If the sensors are too far from the floor, they may not register the line; if they are too close, the chassis could deflect enough under loading that they drag. The module also includes a sensitivity dial that can be adjusted depending on the distance to the floor. Some wiring extension is necessary, for which the design team simply used wire nuts to join wires to their corresponding extensions. A full wiring schematic can be found in **Appendix IV**.

To construct the load-carrying assembly, each of the six 24" long threaded support rods should be inserted into the holes in the bottom chassis and secured on both sides with nuts. Then, nuts can be placed on the support rods where the middle platform is desired to rest, and the middle platform lowered down onto said nuts. A Velcro square was placed at the rear center of this platform as a place to store the remote. Add nuts on the top side of this platform for extra security. At this point, it is helpful to fasten the rubber mat or other load security measures to the top platform. The design team used 1.5" long 4-40 bolts with the largest possible washers that still gripped the bolt head to secure the rubber mat. Finally, place nuts on the threaded rods so that approximately 3/8" of threaded rod is exposed at the end. Then the top platform can be lowered down onto the rods, while making sure that none of the rods are pushing up through the rubber mat.

Test Results

The following is a discussion of the tests each prototype underwent, how it performed, and the conclusions drawn.

Small-Scale Prototype

The small-scale prototype was tested using a black loop of duct tape on a large, white piece of cardboard. The prototype was first programmed to follow the black line indefinitely using the provided sample code. The prototype was able to successfully navigate around the loop five times, at which point the design team stopped the test and concluded the line-following sample code worked sufficiently well. The design team then integrated and adapted sample code for the IR remote and developed original code for multi-directionality and locating a destination. Breaks in the loop of black duct tape were created to indicate potential destinations. Upon initial testing, it was discovered that the wheels of the small-scale prototype did not spin fast enough to make the 180° turn necessary for multi-directionality. After the design team increased the turning speed, the prototype performed exactly as desired. Sample code for ultrasonic proximity sensing was integrated in the programming and various obstacles were placed on the testing track. The small-scale prototype successfully navigated the track without hitting any obstacles; the prototype simply stopped until the obstacles were removed from its path, at which time it continued to its intended destination. It should be noted that the width of the duct tape (approximately 2") should be considered a minimum line width. If the line is too thin, it is possible that it could be underneath the line tracking module but between the sensors, so it is not detected. Furthermore, the radius of curvature of the corners should be at least 12" to allow the prototype to navigate them smoothly.

Full-Scale Prototype

The initial design of the full-scale prototype included 12 ball transfers attached to the underside of the bottom chassis. These ball transfers did not touch the ground until the handler was partially loaded, as their presence was intended to more evenly distribute the load as opposed to making the wheels and motor mounts bear it entirely. However, it was soon discovered that the ball transfers did not roll smoothly on the hard floor and impaired the motion of the handler. The design team removed the transfers one pair at a time, testing the functionality after each removal, but unfortunately ended up removing all of them.

Once able to roll smoothly, the full-scale prototype was placed on a straight line of black duct tape running along a hallway. It was given the command to go to “destination 1”, which it reluctantly did. A strong right bias was noticed, as the handler was supposed to be moving straight, but kept needing to correct to the left. Upon further investigation it was seen that one of the motors on the right side of the handler was only turning at 100rpm instead of the rated 160rpm. This is likely attributed to poor quality control of the motor manufacturer, and unfortunately there was not sufficient time to order a new motor before Undergraduate Research Day. By slightly curving the tape line, the handler was made to travel smoothly until its destination. This allowed for testing of the proximity sensor, which worked as intended.

The main issue facing the full-scale prototype is its current inability to turn. It had a hard time simply adjusting its path along the line, and cannot begin to execute a full turn as the functionality requires. In addition, it was tested and seen that the handler can only carry a load of 45lbs; any more, and it will not begin to move. This is because the motor torque, although calculated to be sufficient for the design with ball transfers, is insufficient to carry more than 45 pounds now that the ball transfers have been removed. The inability to turn is likely because the torque calculations did not account for the fact that the motors must overcome the friction slip the wheels experience when the handler attempts to turn around. It is also highly possible that the motors are not receiving enough current from the battery, thus limiting their torque output (see **Appendix VI**). It will be necessary to determine if the motors are receiving the full amount of current they need; if they are not, directly connecting the battery to the motor driver could solve the torque problem. However, if the motors are receiving the right amount of current, then they are simply too weak for the current design and new, higher-torque motors will be necessary. When placing the handler on blocks and manually manipulating the line tracking sensors, it was seen that all of the functionality from the small prototype is still working. This includes counting rooms, stopping, turning around, and manual control.

Prototype Evaluation

Small Scale Prototype

The small-scale prototype exceeded the design team’s expectations with its performance. Despite some minor imperfections with the overall performance, the prototype successfully completed all tasks necessary for full functionality. Some imperfections with the performance of the small-scale prototype that should be noted are:

- Turn navigation
 - The prototype makes a series of jerky corrections instead of travelling smoothly around the curve.
- Line sensor sensitivity
 - The sensitivity knob on the line tracking module is so low to the floor that it often gets bumped and the sensitivity changes causing the prototype to travel off the line.

- Destination detection
 - If the prototype, for any reason, gets off the line and enters manual mode, it does not always accurately keep track of its position causing the prototype to stop at the incorrect destination.
 - This problem could be solved by using a more robust indication of which destination is which, such as a barcode at each stop and a scanner on the prototype.

Full-Scale Prototype

Although the full-scale prototype is suitable as a proof of concept for some aspects of the material handler, it exhibits some critical flaws that inhibit its overall performance. The prototype does travel at approximately 1 to 1.5 m/s as expected based on the motor speed and wheel diameter. Additionally, the full-scale prototype, when equipped with 12 ball transfers, is able to support the required 500lbs as requested by Plenty Unlimited. The materials used for the prototype are food-grade and food-safe; the HDPE platforms are food-safe, and the galvanized steel supporting rods are food-safe unless in the presence of highly acidic foods such as fruits. This does pose an issue, however, this full-scale prototype will likely only be used in Plenty Unlimited Inc.'s Laramie facility. Consequently, it does not need to meet all FDA standards since the Laramie location does not distribute produce for public consumption.

There are several flaws with the full-scale prototype that must be addressed prior to it being used in a warehouse application. The motors do not produce sufficient torque to consistently move the prototype along the line. Furthermore, the provided torque is not sufficient to allow the full-scale prototype to turn around for multi-directional capability. In addition to moving in a jerky motion, the prototype is not able to navigate turns without becoming stuck on the track. Because the full-scale prototype is too wide for one ultrasonic proximity sensor to safely cover, more proximity sensors must be implemented to ensure employee safety. Due to the questionable quality of the current motors, the design team was unable to test all aspects of the prototype. Several design revisions are needed in order to produce a fully functioning full-scale prototype.

Prior Art

Automated material handlers are already in existence. Additional information about existing automated material handlers can be found in the “Benchmarking” section. Currently, there are no intellectual property (IP) aspects associated with our design. Our design is unique in that it has only fundamental functionality, is made of food-safe materials, and uses line following technology for navigation. The design will simply continue with Plenty Unlimited, Inc. as a trade secret. The design team does not plan on protecting the design with a patent.

Creativity and Innovation

The primary creative aspect of the current material handler design is the use of IR line following technology for navigation. Although IR line following technology may not be as elegant or reliable as machine vision, using IR line tracking as a means of navigating a warehouse is much easier, and more importantly less expensive, than utilizing image processing.

A secondary creative aspect of the design is the use of threaded support rods between platforms. The threaded rods allow for complete customization with regards to the quantity and spacing of the platforms. Using such rods will allow Plenty Unlimited to alter the design based on the types of goods they need to transport.

Contributions of Each Member

Due to having a small group of two members, a substantial amount of the design process was performed jointly during regular scheduled meetings. However, each member was responsible for a primary aspect of the design. The design was divided into two main components: automation/electronics and physical prototype design/construction. Although there was joint collaboration between team members for each step in the design, Mark Fenn was primarily responsible for the automation/electronic components and Brendan Taedter was primarily responsible for prototype design and construction.

While working with the automation aspect of the design, Mr. Fenn learned a substantial amount about efficient programming, especially in the C/C++ languages that Arduino microprocessors use. Additionally, Mr. Fenn learned how to effectively debug and troubleshoot when aspects of the automation did not work as expected.

Mr. Taedter learned a great deal about the design process as a whole; specifically, that there are always unforeseen complications when making the transition from concept to prototype, and from one prototype to the next. Speaking more technically, Mr. Taedter learned much about appropriate material selection, and all the factors included in such a decision. He thoroughly enjoyed developing solid models of the full-scale prototype, and editing these models based on updated material selections and design considerations.

Conclusions

The design team successfully developed a functioning automated material handler with essential functionality for a fraction of the cost of commercial material handlers. The design is adaptable to almost any warehouse layout and is completely customizable to fit the exact needs of Plenty Unlimited. The design team was able to develop a fully functional automation program designed specifically for agricultural warehouse applications using existing technology and available sample code. The design team was also able to manage the project throughout the entire design process; this allowed the team members to witness and overcome the various hurdles that arise during a project of this caliber. Although some modifications to the full-scale prototype are required before implementing the material handler in a warehouse setting, the design is a solid foundation for engineers at Plenty Unlimited to build upon.

Recommendations for Future Work

The design team has developed the fundamental aspects of the material handling system. With minor modifications, the full-scale prototype will be fully-functional and able to be implemented into Plenty Unlimited's warehouses.

Motors and Driving Systems

The most pressing alteration required to attain full functionality is modification of the current motors and driving systems. As stated in the “Test Results” section, the motors are either not receiving sufficient current or do not provide the required torque to move the material handler successfully. To solve this issue, the design team suggests directly connecting a battery to the motor driver to achieve the necessary current. If the required current cannot be achieved, or the motors are still too weak even provided with the necessary current, the motors should be replaced with higher quality motors with higher torque. The driving system can be further improved by replacing the current wheels; implementation of omni-wheels would eliminate the sliding friction created while the material handler is turning. Modification of the driving systems should allow the material handler to achieve full functionality.

Additional Improvements

In addition to the required modifications, there are minor changes to the design that would greatly increase the user experience and performance of the material handler. The current design of the material handler is stronger than required and thus heavier than necessary. Performance of the material handler could be greatly improved by optimizing the design and reducing the overall weight. The current full-scale prototype weighs 50lbs, which could be cut down by using thinner supporting rods and removing some HDPE from the centers of the load-carrying platforms. The material handler could be further optimized by having modified chassis designs specifically designed to transport certain goods. To ensure the safety of employees, additional proximity sensors need to be implemented on the front corners of the material handler; the current design only utilizes one proximity sensor which does not have a wide enough field of vision to detect all obstacles in the material handler’s path. The design could be further improved by implementing a more robust system for detecting destinations. Instead of using breaks in the track, each room could be designated by a unique barcode which would ensure correct destination identification.

References

- [1] Murphy, Andrew. "AGV Deep Dive: How Amazon's 2012 Acquisition Sparked a \$10B Market." *Loup Ventures*, loupventures.com/agv-deep-dive-how-amazons-2012-acquisition-sparked-a-10b-market/.
- [2] "HDPE (High Density Polyethylene)." *Plastics International*, Plastics International, www.plasticsintl.com/datasheets/HDPE.pdf.
- [3] "In Contact with Food." *American Galvanizer's Association*, galvanizeit.org/hot-dip-galvanizing/how-long-does-hdg-last/contact-with-food/.
- [4] Online Metals. "316/316L Sheet / Plate." *OnlineMetals.com*, www.onlinemetals.com/merchant.cfm?id=325&step=2&top_cat=1.

List of Appendices

Appendix I: Usage Instructions	20
Appendix II: Full Arduino Code	22
Appendix III: SolidWorks Drawings	30
Appendix IV: Wiring Schematic	32
Appendix V: Parts List	33
Appendix VI: Torque Requirement Calculations	34
Appendix VII: Product Design Specifications (PDS)	35
PDS Attachment 1. FMEA Spreadsheet	42
PDS Attachment 2. Product Roadmap Analysis	43
PDS Attachment 3. Return on Investment Analysis	44
PDS Attachment 4. Gantt Chart	45
PDS Attachment 4. Work Breakdown Structure	46

Appendix I: Usage Instructions

Creating the Track

The track should be a 2” wide black line of paint or sturdy tape placed on a relatively level, light-colored surface. Each room or desired destination shall be indicated with a 2” long break in the track. With the exception of the breaks indicating rooms, the black line should make a continuous loop.

Rooms shall be numbered 1 through 9 and increase in the counter clockwise (CCW) direction. Additional rooms may be added by editing the code.

Turns in the loop should be smooth curves with radii of curvature as large as possible in the warehouse. The minimum radius of curvature that the prototype can successfully navigate needs to be tested, but the design team anticipates the radius to be approximately 48”.

Start Up

- 1) Place the material handler on the warehouse floor with the line sensing module directly over the black line. The handler must be oriented in the CCW direction and be positioned between Room 1 and the highest number room.
- 2) Connect the charged battery to the Arduino Microprocessor and switch the battery on. Lights on the Arduino Microprocessor should light up.
- 3) Push the “1” button on the remote. The material handler should start moving until it reaches Room 1.
- 4) The material handler is now ready to be used.

Normal Operation

- 1) Load the cart with the desired goods.
- 2) Press the button on the remote corresponding to the destination to which the goods shall be delivered.
- 3) The handler should travel in the direction of shortest distance to the desired destination.
- 4) If the handler gets off the track and stops, the LCD screen will show an error and the handler will enter “Manual Mode” and will need to be placed back on the track (*See “Manual Mode” instructions below*).

Manual Mode

The material handler will enter “Manual Mode” if it gets off the track and stops. “Manual Mode” should only be used when necessary and is not an efficient means of making the material handler transport goods.

- 1) The arrow buttons on the remote indicate directions in which the handler can move. Pressing the “Up” arrow will move the handler forward for 0.5 seconds. Similarly, pressing the “Down” arrow will move the handler backward for 0.5 seconds. Pressing the “Left” or “Right” arrows will turn the handler CCW or CW, respectively, for 0.25 seconds.

- 2) Use the arrow buttons to place the handler back on the track facing the same direction as its most recent direction of travel.
- 3) Once the handler is back on the track, press the number button corresponding to the room that the handler just passed (e.g. If the handler just passed Room 4, press the “4” button on the remote).
- 4) Push the “OK” button on the remote and the material handler will proceed to its destination.

Appendix II: Full Arduino Code

```
#include <IRremote.h>
#include <LiquidCrystal.h>
#include <Servo.h>
Servo myservo;

//Line Tracking IO define
#define LT_R !digitalRead(10)
#define LT_M !digitalRead(4)
#define LT_L !digitalRead(2)

#define ENA 5
#define ENB 6
#define IN1 7
#define IN2 8
#define IN3 9
#define IN4 11

#define carSpeed 200
#define turnSpeed 225

//IR Remote Sequences
#define RECV_PIN 12 //IR signal receiving pin
#define ONE 16738455
#define TWO 16750695
#define THREE 16756815
#define FOUR 16724175
#define FIVE 16718055
#define SIX 16743045
#define SEVEN 16716015
#define EIGHT 16726215
#define NINE 16734885
#define ZERO 16730805
#define STAR 16728765
#define POUND 16732845
#define UP 16736925
#define DOWN 16754775
#define LEFT 16720605
#define RIGHT 16761405
#define OK 16712445

//LCD Display initialize interface pins
LiquidCrystal lcd(53,51,37,35,33,31);

unsigned long val;

//Warehouse Characteristics
int pos;
int dest;
int maxpos;

int curdir; //current direction
int desdir; //desired direction
//direction will be either stored as a 0 for counter clockwise (CCW) or as a 1 for clockwise (CW)

int distCCW;
int distCW;
```

```

int ok; //for use in manual mode

//FOR OBSTACLE DETECTION
int Echo = A4;
int Trig = A5;
int middleDistance=0;

IRrecv irrecv(RECV_PIN);
decode_results results;

void forward(){
  analogWrite(ENA, carSpeed);
  analogWrite(ENB, carSpeed);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  Serial.println("go forward!");
}

void back(){
  analogWrite(ENA, carSpeed);
  analogWrite(ENB, carSpeed);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  Serial.println("go back!");
}

void left(){
  analogWrite(ENA, carSpeed);
  analogWrite(ENB, carSpeed);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  Serial.println("go left!");
}

void right(){
  analogWrite(ENA, carSpeed);
  analogWrite(ENB, carSpeed);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  Serial.println("go right!");
}

void stop(){
  digitalWrite(ENA, LOW);
  digitalWrite(ENB, LOW);
  Serial.println("Stop!");
}

int Distance_test(){
  digitalWrite(Trig, LOW);
  delayMicroseconds(2);
  digitalWrite(Trig, HIGH);
  delayMicroseconds(20);
  digitalWrite(Trig, LOW);
}

```

```

float Fdistance = pulseIn(Echo, HIGH);
Fdistance= Fdistance / 58;
return (int)Fdistance;
}

```

```

void turnCCW(){
  analogWrite(ENA, turnSpeed);
  analogWrite(ENB, turnSpeed);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  Serial.println("Turn CCW");
}

```

```

void turnCW(){
  analogWrite(ENA, turnSpeed);
  analogWrite(ENB, turnSpeed);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  Serial.println("Turn CW");
}

```

```

void lcdDisplay(){
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("pos:");
  lcd.setCursor(4,0);
  lcd.print(pos);
  lcd.setCursor(9,0);
  lcd.print("dest:");
  lcd.setCursor(14,0);
  lcd.print(dest);
  lcd.setCursor(0,1);
  lcd.print("direction:");
  lcd.setCursor(10,1);
  if (curdir==0){
    lcd.print("CCW");
  }
  else if (curdir==1){
    lcd.print("CW");
  }
}

```

```

void offTrackError(){
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("ERROR: OFF TRACK");
  lcd.setCursor(0,1);
  lcd.print("Manual Enabled");
}

```

```

void ManualMode(){
  if (irrecv.decode(&results)){
    val = results.value;
    irrecv.resume();
    switch(val){
      case ONE: pos=1; break;
      case TWO: pos=2; break;
      case THREE: pos=3; break;
    }
  }
}

```

```

    case FOUR: pos=4; break;
    case FIVE: pos=5; break;
    case SIX: pos=6; break;
    case SEVEN: pos=7; break;
    case EIGHT: pos=8; break;
    case NINE: pos=9; break;
    case ZERO: break;
    case STAR: curdir=0; break; //CCW
    case POUND: curdir=1; break; //CW
    case UP: forward(); delay(500); stop(); break;
    case DOWN: back(); delay(500); stop(); break;
    case LEFT: turnCCW(); delay(250); stop(); break;
    case RIGHT: turnCW(); delay(250); stop(); break;
    case OK: ok=1; break;
  }
}
else {
  ok=0;
}
}

void setup(){
  Serial.begin(9600);
  myservo.attach(3);
  myservo.write(90);
  pinMode(Echo, INPUT);
  pinMode(Trig, OUTPUT);
  pinMode(LT_R,INPUT);
  pinMode(LT_M,INPUT);
  pinMode(LT_L,INPUT);
  irrecv.enableIRIn();
  lcd.begin(16,2); //set up LCD's number of columns and rows (16 columns, 2 rows) for current display.
  maxpos=9; //number of rooms
  pos=0; //store current position as 0 (right before room 1 going counter clockwise)
  dest=0; //current destination as 0 so that input destination will work (note can't go back to maxpos right off the start)
  curdir=0; //initial direction will be set to be counterclockwise
  desdir=0; //desired destination will be set to counterclockwise to start
  distCCW=0; //both distances set to zero to start
  distCW=0;
  ok=0;
  lcdDisplay();
}

void loop() {
  middleDistance = Distance_test();
  if (middleDistance <= 30){
    stop();
  }
  else {
    if (pos==dest){
      stop();
      if (irrecv.decode(&results)){
        val = results.value;
        irrecv.resume();
        switch(val){
          case ONE: dest=1; delay(1000); if (dest==pos){ break;}
            else {
              if (dest<pos){distCW=pos-dest; distCCW=maxpos-pos+dest;}
              else if (dest>pos){distCW=pos+maxpos-dest; distCCW=dest-pos;}
              if (distCCW<distCW){desdir=0;}
              else if (distCW<distCCW){desdir=1;}
              else {desdir=curdir;}
            }
        }
      }
    }
  }
}

```

```

        if (desdir==curdir){ forward(); delay(300); lcdDisplay(); break;}
        else if (desdir==1 && curdir==0){turnCCW(); delay(500); turnCCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}
        else if (desdir==0 && curdir==1){turnCW(); delay(500); turnCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}}

    case TWO: dest=2; delay(1000); if (dest==pos){ break;}
    else {
        if (dest<pos){distCW=pos-dest; distCCW=maxpos-pos+dest;}
        else if (dest>pos){distCW=pos+maxpos-dest; distCCW=dest-pos;}
        if (distCCW<distCW){desdir=0;}
        else if (distCW<distCCW){desdir=1;}
        else {desdir=curdir;}
        if (desdir==curdir){ forward(); delay(300); lcdDisplay(); break;}
        else if (desdir==1 && curdir==0){turnCCW(); delay(500); turnCCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}
        else if (desdir==0 && curdir==1){turnCW(); delay(500); turnCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}}
    case THREE: dest=3; delay(1000); if (dest==pos){ break;}
    else {
        if (dest<pos){distCW=pos-dest; distCCW=maxpos-pos+dest;}
        else if (dest>pos){distCW=pos+maxpos-dest; distCCW=dest-pos;}
        if (distCCW<distCW){desdir=0;}
        else if (distCW<distCCW){desdir=1;}
        else {desdir=curdir;}
        if (desdir==curdir){ forward(); delay(300); lcdDisplay(); break;}
        else if (desdir==1 && curdir==0){turnCCW(); delay(500); turnCCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}
        else if (desdir==0 && curdir==1){turnCW(); delay(500); turnCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}}
    case FOUR: dest=4; delay(1000); if (dest==pos){ break;}
    else {
        if (dest<pos){distCW=pos-dest; distCCW=maxpos-pos+dest;}
        else if (dest>pos){distCW=pos+maxpos-dest; distCCW=dest-pos;}
        if (distCCW<distCW){desdir=0;}
        else if (distCW<distCCW){desdir=1;}
        else {desdir=curdir;}
        if (desdir==curdir){ forward(); delay(300); lcdDisplay(); break;}
        else if (desdir==1 && curdir==0){turnCCW(); delay(500); turnCCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}
        else if (desdir==0 && curdir==1){turnCW(); delay(500); turnCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}}
    case FIVE: dest=5; delay(1000); if (dest==pos){ break;}
    else {
        if (dest<pos){distCW=pos-dest; distCCW=maxpos-pos+dest;}
        else if (dest>pos){distCW=pos+maxpos-dest; distCCW=dest-pos;}
        if (distCCW<distCW){desdir=0;}
        else if (distCW<distCCW){desdir=1;}
        else {desdir=curdir;}
        if (desdir==curdir){ forward(); delay(300); lcdDisplay(); break;}
        else if (desdir==1 && curdir==0){turnCCW(); delay(500); turnCCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}
        else if (desdir==0 && curdir==1){turnCW(); delay(500); turnCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}}
    case SIX: dest=6; delay(1000); if (dest==pos){ break;}
    else {
        if (dest<pos){distCW=pos-dest; distCCW=maxpos-pos+dest;}
        else if (dest>pos){distCW=pos+maxpos-dest; distCCW=dest-pos;}
        if (distCCW<distCW){desdir=0;}
        else if (distCW<distCCW){desdir=1;}
        else {desdir=curdir;}
        if (desdir==curdir){ forward(); delay(300); lcdDisplay(); break;}

```

```

        else if (desdir==1 && curdir==0){turnCCW(); delay(500); turnCCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}
        else if (desdir==0 && curdir==1){turnCW(); delay(500); turnCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}}
        case SEVEN: dest=7; delay(1000); if (dest==pos){ break;}
        else {
            if (dest<pos){distCW=pos-dest; distCCW=maxpos-pos+dest;}
            else if(dest>pos){distCW=pos+maxpos-dest; distCCW=dest-pos;}
            if (distCCW<distCW){desdir=0;}
            else if(distCW<distCCW){desdir=1;}
            else {desdir=curdir;}
            if (desdir==curdir){forward(); delay(300); lcdDisplay(); break;}
            else if (desdir==1 && curdir==0){turnCCW(); delay(500); turnCCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}
            else if (desdir==0 && curdir==1){turnCW(); delay(500); turnCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}}
        case EIGHT: dest=8; delay(1000); if (dest==pos){ break;}
        else {
            if (dest<pos){distCW=pos-dest; distCCW=maxpos-pos+dest;}
            else if(dest>pos){distCW=pos+maxpos-dest; distCCW=dest-pos;}
            if (distCCW<distCW){desdir=0;}
            else if(distCW<distCCW){desdir=1;}
            else {desdir=curdir;}
            if (desdir==curdir){forward(); delay(300); lcdDisplay(); break;}
            else if (desdir==1 && curdir==0){turnCCW(); delay(500); turnCCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}
            else if (desdir==0 && curdir==1){turnCW(); delay(500); turnCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}}
        case NINE: dest=9; delay(1000); if (dest==pos){ break;}
        else {
            if (dest<pos){distCW=pos-dest; distCCW=maxpos-pos+dest;}
            else if(dest>pos){distCW=pos+maxpos-dest; distCCW=dest-pos;}
            if (distCCW<distCW){desdir=0;}
            else if(distCW<distCCW){desdir=1;}
            else {desdir=curdir;}
            if (desdir==curdir){forward(); delay(300); lcdDisplay(); break;}
            else if (desdir==1 && curdir==0){turnCCW(); delay(500); turnCCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}
            else if (desdir==0 && curdir==1){turnCW(); delay(500); turnCW(); while(!LT_M); stop(); curdir=desdir;
lcdDisplay(); break;}}
        case ZERO: dest=dest; break;
        case STAR: dest=dest; break;
        case POUND: dest=dest; break;
        case UP: forward(); delay(500); stop(); break;
        case DOWN: back(); delay(500); stop(); break;
        case LEFT: turnCCW(); delay(250); stop(); break;
        case RIGHT: turnCW(); delay(250); stop(); break;
        case OK: dest=dest; break;
    }
}
else {
    dest=dest;
}
}
else {
    if(LT_L && LT_R) {
        forward();
    }
    else if(LT_L && !LT_R) {
        left();
        while(LT_L && !LT_R);
    }
}

```

```

}
else if(!LT_L && LT_R) {
  right();
  while(!LT_L && LT_R);
}
else if(!LT_L && !LT_R) {
  if (curdir==0){
    if (pos==maxpos){
      pos=1;
      lcdDisplay();
      if(pos!=dest){
        forward();
        delay(500);
        if (!LT_L && !LT_R){
          stop();
          ok=0;
          offTrackError();
          while(ok==0){
            ManualMode();
          }
        }
      }
      else{
        //Do nothing
      }
    }
    else{
      //Do nothing
    }
  }
  else {
    pos=pos+1;
    lcdDisplay();
    if(pos!=dest){
      forward();
      delay(500);
      if (!LT_L && !LT_R){
        stop();
        ok=0;
        offTrackError();
        while(ok==0){
          ManualMode();
        }
      }
      else{
        //Do nothing
      }
    }
    else{
      //Do nothing
    }
  }
}
else if (curdir==1){
  if (pos==1){
    pos=maxpos;
    lcdDisplay();
    if(pos!=dest){
      forward();
      delay(500);
      if (!LT_L && !LT_R){
        stop();
        ok=0;

```


Appendix III: SolidWorks Drawings

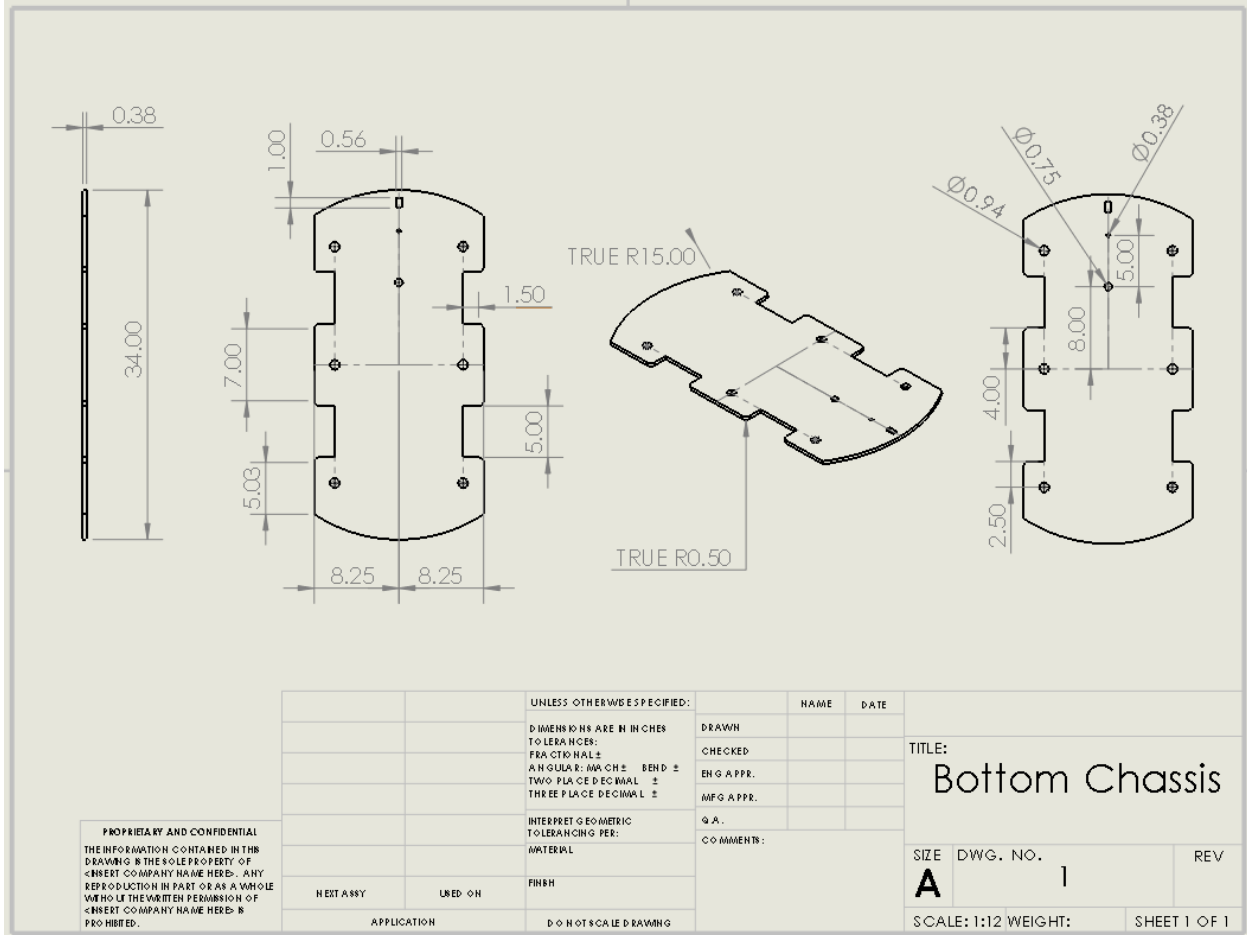


Figure A1. SolidWorks Drawing of Bottom Chassis

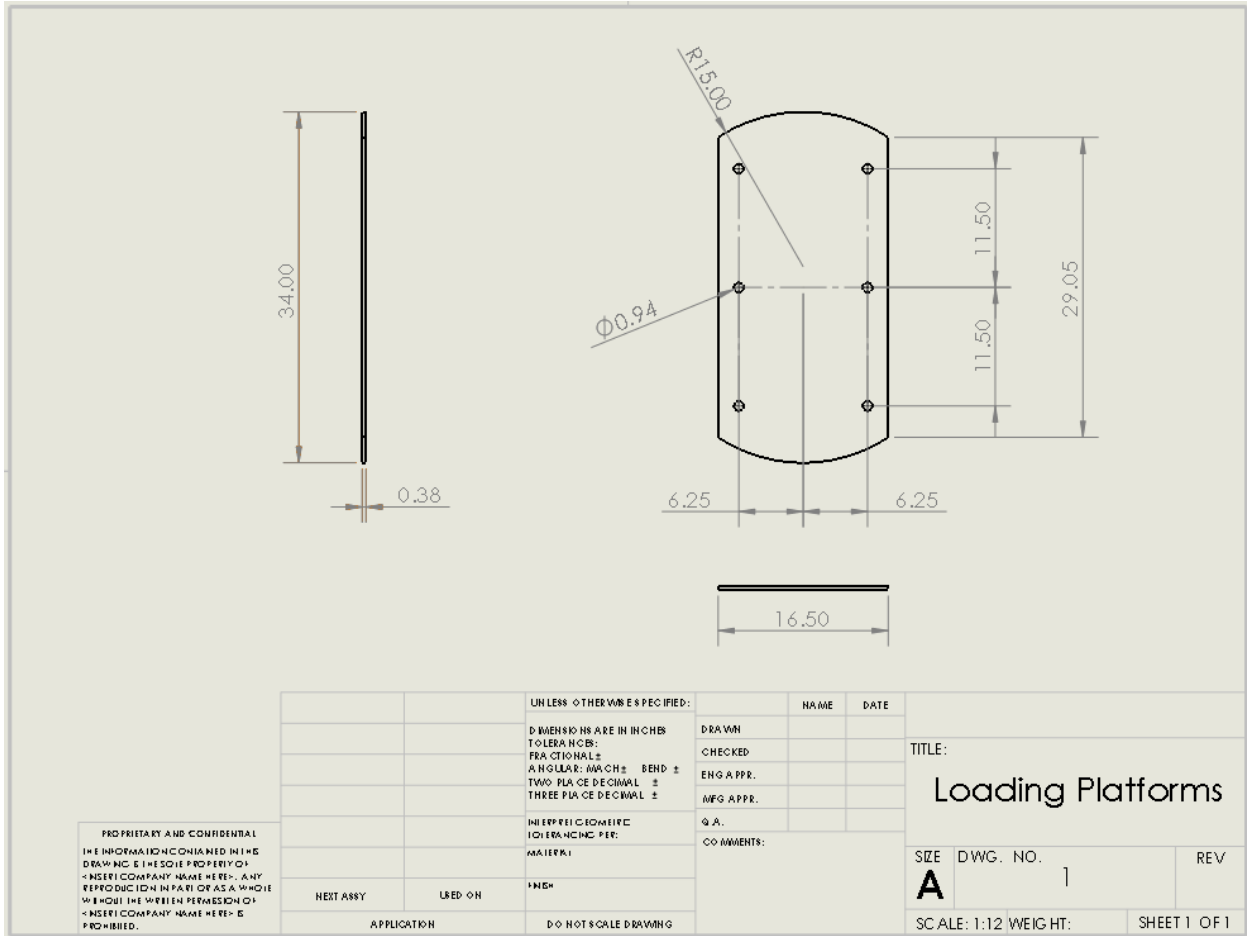


Figure A2. SolidWorks Drawing of Loading Platforms

Appendix IV: Wiring Schematic

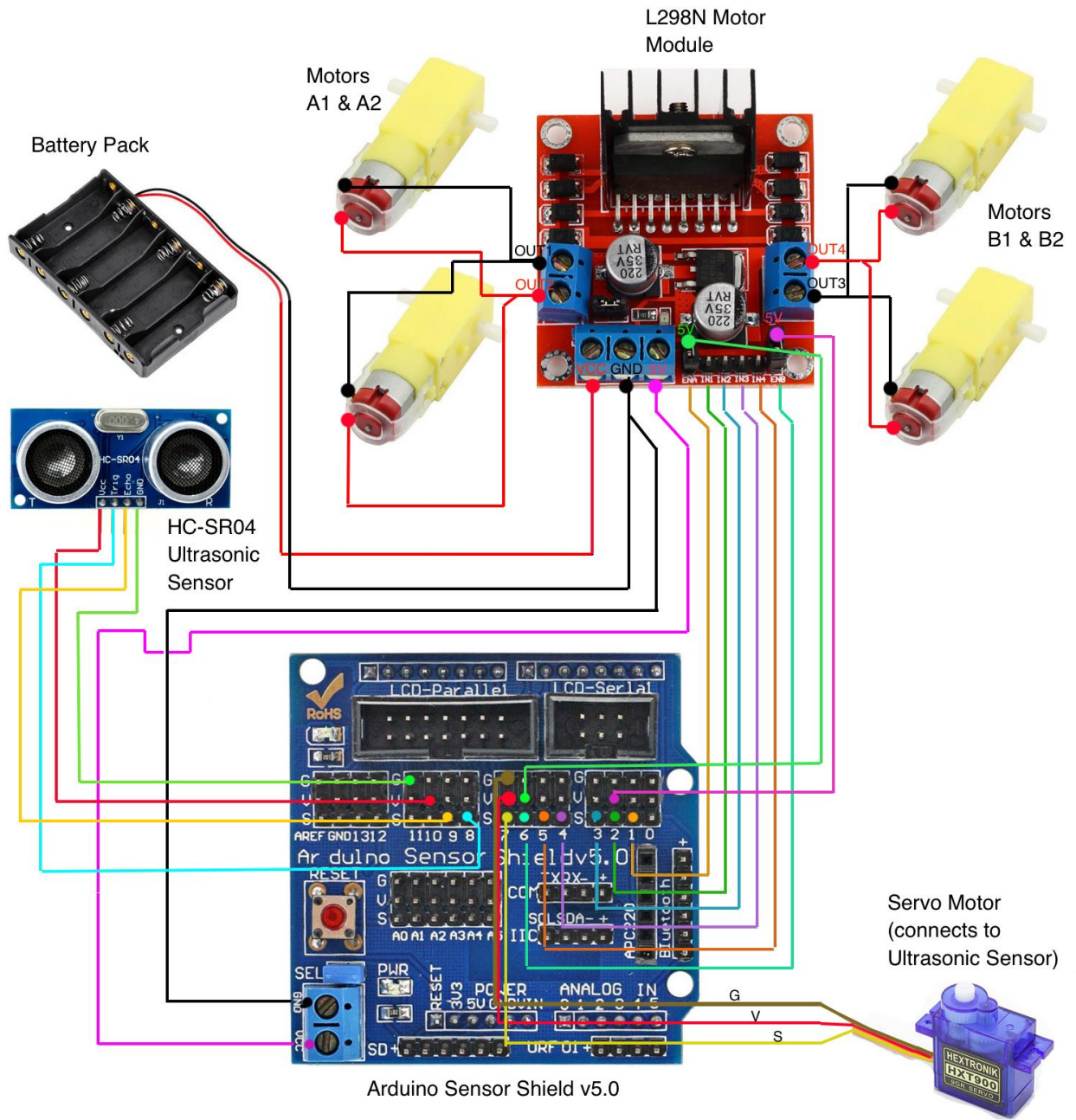


Figure A3. Arduino Wiring Schematic (from Elegoo, Inc.)

Appendix V: Parts List

Table A1. Parts List

Item Description	Quantity
Structural	
6' long 7/8" galvanized steel threaded rods	2
7/8" nuts for threaded rods	30
36" by 18" by 3/8" natural HDPE sheets	3
Drive Systems	
5" rubber wheels	4
Bemonoc 12V 2.5kgcm 160rpm DC motors	4
2" by 2" by 1/8" by 2.5" long angled aluminum mounts	4
6-32 5/8" long NC socket head bolts	12
8-32 3/4" long NC socket head bolts	4
Electrical Components	
12.6V DC 20000mAh Li-ion rechargeable battery	1
Arduino Mega with expansion board*	1
L298 quad motor driver*	1
Ultrasonic proximity sensor*	1
Proximity sensor servo motor*	1
LCD screen and breadboard	1
Remote control*	1
IR line tracking module (with 3 IR pairs)*	1
Wire nuts	32
Misc. wiring extensions	variable
Other Hardware	
6-32 1.125" long bolts (any head)	16
4-40 1/2" long bolts (any head)	4
4-40 1" long bolts (any head)	3
4-40 1.5" long bolts (any head)	4
4-40 nuts	11
6-32 nuts	16
adhesive velcro squares	5
3/8" long bolts (for Arduino and L298)*	16
2' long internally threaded spacers*	8
4-40 large washers	4
Rubber load security mat	1
Plastic slide-out drawers	2
*from Elegoo Smart Car kit	

Appendix VI: Torque Requirement Calculations

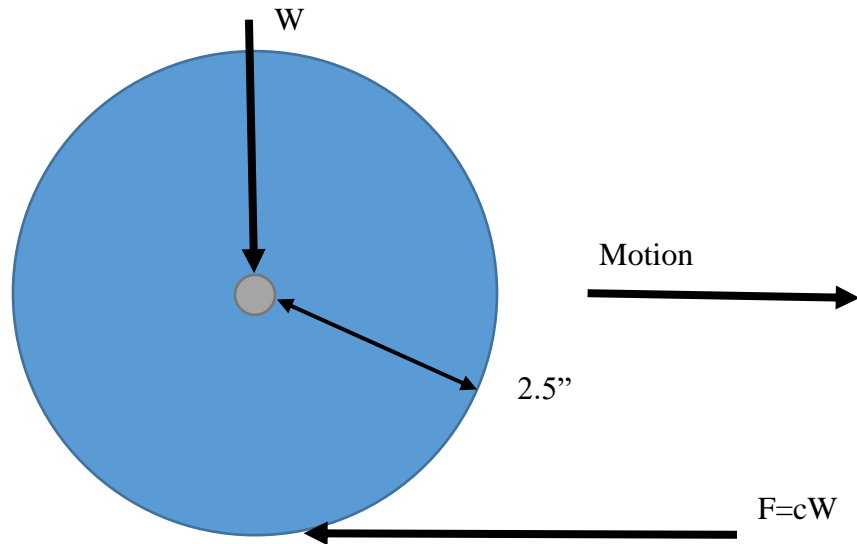


Figure A4. Free Body Diagram of a Wheel

W=weight applied to wheel

F=force opposing wheel rotation due to rolling resistance

c=coefficient of rolling resistance (=0.002 from engineeringtoolbox.com)

T=torque

Rated torque=2.5kgcm=2.5Ncm (kg and N are often sloppily interchanged on motor specs)
=.025Nm=.0184ftlb

With 12 Ball Transfers

$$W=550/16=34.4\text{lb}$$

$$F=.002(34.4)=0.069\text{lb}$$

$$T(\text{necessary})=0.069(2.5/12)=\mathbf{0.0143\text{ftlb}}$$
 (motors have sufficient torque)

Without Ball Transfers

$$W=550/4=137.5\text{lb}$$

$$T(\text{necessary})=\mathbf{0.057\text{ftlb}}$$
 (current motors are not capable of this torque requirement)

Loaded with 45lbs, no Transfers

$$W=95/4=23.75\text{lb}$$

T(necessary)=**0.010ftlb** (45lbs is the experimental weight limit of the current handler. Any more and it will not begin to move. Because the torque necessary to move 45lbs is only about half the rated torque, the team suspects the motors are not receiving sufficient current, thus limiting their torque output.)

Appendix VII: PDS (Revised February 2018)

Product-Project Design Specification (PDS) 2017-2018

ME 4060 & ME 4070

Product-Project Title: Plenty Unlimited Material Handler

PDS Version No: 3.0

PDS Version Date: 2/8/18

Background:

This PDS provides a checklist that supports development of this Product or Project. It is a living document and shall be updated as new information or influences are discovered that impact this development. To the extent practical, it should be written using indented numbered list format for brevity and clarity with liberal inclusion of images and graphics as applicable. This document shall be the prime focus very early in the product-project development process. The PDS lays the groundwork for all early planning and ahead of all design activities – it ensures all factors are accounted and all stakeholders are heard from. A PDS is not a record of what has been done, but rather a comprehensive collection of all of the influence parameters and considerations that must be continuously re-visited throughout the development process to ensure a successful realization of this Product or Project. Consider that there will generally be a phases to products or projects such as Prototype, Pilot-Production and Full-Production. For purposes of progress evaluation and assessment of continued efforts, early focus on the Prototype and Pilot-Production phases is recommended – other downstream phases may be included by making adjustments to the considerations listed further below. This PDS has been checked and agreed by the development team, faculty advisors, client/customer and stakeholder/sponsors as applicable. In addition, signatures from both Professor Kilty and Professor Willey are mandatory (see signature section at the end of this document).

PDS Version History:

Track development and distribution of the PDS, up to the final point of approval (all signatures).

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Mark Fenn, Brendan Taedter	11/30/2017	Self	11/30/17	Initial Design Definition draft
2.0	Mark Fenn, Brendan Taedter	12/5/2017	Self	12/5/17	Substantial Completion
3.0	Mark Fenn, Brendan Taedter	2/8/18	Self	2/8/18	Revisions

Target Market: Agricultural Technology Companies, specifically Plenty Unlimited, Inc.

Key Market Assumptions & All Influence Parameters:

- Automated material handlers already exist, but are cost-prohibitive to upstart companies. Furthermore, these material handlers won't satisfy the exact needs of Plenty.
- Creating an automated material handler for Plenty Unlimited, Inc. will be much cheaper, as well as specifically tailored to Plenty's needs.
- The material handler will save Plenty's employees time, and therefore save the company money.
- The material handler will make employee's jobs safer, less physically strenuous, and more efficient.

Market Entry Requirements: Because this product will be completed specifically for Plenty Unlimited, Inc., the only requirements are those that have been supplied by Plenty Unlimited, Inc.

System Boundary:

- Internal
 - Electric propulsion
 - Load-carrying platform
 - Controls to summon/send
 - Guide sensors to follow a black line around the warehouse floor
 - Safety
- External
 - Loads carried (different kinds and weights)
 - Hard, smooth floor
 - Moisture/temperature
 - User
 - Expected obstacles
- Outside
 - Unlikely/unexpected obstacles
 - Price of electricity

Out of Scope:

- Warehouse layout (determined by Plenty Unlimited, Inc.)
- Fully autonomous functionality
- Mechanical guide system
- Performing the necessary regular maintenance
- Voice activation

Primary Function: Automatically deliver trays of produce to specified locations in a warehouse

Secondary Functions: Delivery of tools and other materials to warehouse locations. Also doubles as a scale.

User 1 (decision-maker/buyer): Administration of Plenty Unlimited, Inc.

User 2 (primary operator): Warehouse workers at Plenty Unlimited, Inc.

User 3 (secondary operator): Office workers at Plenty Unlimited, Inc.

Wonderful, Outstanding & Way Beyond (WOW) – Customer/User Delight: Built-in scale to avoid overloading. User-friendly button interface for sending and summoning material handler. Will include slide-out trays for additional carrying capacity of miscellaneous items.

WOW Factor – User Characteristics Analysis (attitude, belief, need & want): Essentially, if the handler functions as planned, the users will realize what a vast improvement it is over manual transportation of goods. The scale mentioned above could log and display a running total of the total weight it has transported, as a way to make the benefits more tangible to the users.

WOW Factor – User Situational Analysis (planning influences; e.g., size, occasion, significance & timing): Material handler will be a convenient size; will be large enough to carry significant loads while being compact enough to not impede other warehouse operations. Will be usable all day, every day due to rechargeable power supply. Significance will be realized over time, as the handler saves employee time and energy, and company money.

Prior Art (Intellectual Property (IP)): There are similar material handlers already on the market. These are very expensive and are used in larger-scale, more industrial applications. Our design will likely not have any proprietary components.

This Project's Initial Approach to IP (e.g., Patent, Trade Secret): The design and its implementation would best be categorized as a trade secret. There will likely be no patentable technology.

Benchmarking and Competitive Analysis: Other designs on the market have more programmed functionality but are much more expensive. Our design will have similar base functionality (enough to satisfy the Primary Function above) at approximately 1/10th the cost.

Prevailing Price and Value Landscape and Trends: The material handler will be somewhat of a barebones version of more expensive and complex designs currently on the market. Therefore, it will be competitive as far as cost is concerned while still maintaining similar profitability and relatively similar performance.

Product-Project Design Objective: Create a material handler that automatically delivers supplies to various warehouse locations in order to minimize labor costs and maximize efficiency.

Product Environmental Requirements; e.g., temperature, humidity, shock, contaminants, corrosives: Material handler will need to be able to function within the farming warehouse - a warm and moderately humid environment. There should be no issues as long as the material handler does not encounter a highly corrosive environment.

FMEA (Failure Modes and Effects Analysis): See Attachment 1 for FMEA spreadsheet.

Critical to Function: Power supply, user input, autonomous guide system, microprocessor programming

Critical to Quality Attributes (CTQs) – all phases:

- Handler moves at least as fast as a human can walk, but not so fast that it endangers the safety of the load or the warehouse workers
- Handler is able to be summoned or sent from any location to any other location

- Handler can support more than as much weight could be reasonably loaded onto it while still functioning properly
- Handler “knows” where to go and when to stop using sensors, instead of a mechanical guide system
- Handler has a manual control feature that allows the user to drive it by remote control
- Powered by rechargeable battery to avoid dragging a cord

Overall Sizes: Preliminary design is for a cart that measures 30 inches long by 18 inches wide. The height of the load-carrying platform, which will be the top surface of the cart, is currently flexible to allow for placement of electric motors, battery, and other working components. However, the height will need to be between 30 and 42 inches (2.5 to 3.5 feet) to allow for ease of loading/unloading.

Overall Weights: Preliminary approximations predict the material handler’s weight to be around 80 pounds, although this parameter isn’t particularly important since normal use wouldn’t involve lifting the device.

Power Requirements: (1) 12V, 20Ah Rechargeable Lithium Ion Battery (at most).

Materials:

- Load-carrying platform: stainless steel or high-strength plastic – must be food-safe
- Load-carrying surface: rubber or equivalent “grippy” surface – must be food-safe
- Electric motors / sensors / wires / batteries
- Assorted assembly hardware

Health & Safety Requirements:

- Materials must be food safe
- System must have minimal footprint
- System must be safe to operate, free of pinch points and sharp edges, and easily stoppable in case of emergency
- Handler will have a proximity sensor to detect any obstacles in its path

Potential Sources of Liability Litigation, Hazards (all phases) and Misuse/Abuse: Handler will be free of pinch points and equipped with proximity sensors to avoid unintentional human contact. Electric components will be isolated to avoid the risk of electric shock. An emergency stop button to avoid further liability.

Codes & Standards Compliance Requirements: Handler will meet FSMA standards.

Ergonomics Considerations:

- Buttons to send/summon the handler will be easy to locate and use
- Loading platform will be at a height such that it is easy to load/unload
- Secondary loading platform will include sliding trays for additional carrying capacity
- Manual control feature will prevent anyone from having to physically move the relatively heavy machine

Motion and Time Study Requirements: Because the warehouse workers have very repetitive work cycles, it would be valuable to conduct a time and motion study. Although no such study has been conducted for this project yet, the material handler will undoubtedly be an improvement over manual transportation. Workers will be able to use the time the handler is in motion to take care of personal needs

or short work-related tasks, and will need less rest time to overcome fatigue. Because the handler will be able to move two trays of produce at a time, whereas a person can only carry one, it would reduce the number of trips a worker takes by half.

Aesthetic Appearance (Shape, Color, Texture): Low profile, either stainless steel with metallic finish or food-safe plastic. Generally rectangular with rounded corners and edges. Approximate dimensions 30”L x 18”W x 30”H.

Product Life Span & Life Cycle Requirements: Life of approximately 10 years with minimal routine scheduled maintenance. Will experience cyclic loading and will need a high fatigue strength.

Possible Off-The-Shelf Components: Microprocessor, Electric Motors, Electronic Components, Wheels, Cart Frame; nearly all components will be purchased rather than manufactured with the exception of some attachments and the custom chassis.

Maintenance Requirements: Routine maintenance every 1000 hrs. Occasional maintenance to batteries, tires, chassis, and electronic components as needed.

Assembly Considerations: Easy access to mounting hardware for electric motors and other electronic components. Must be able to be easily disassembled and cleaned.

Serviceability Requirements: Motors and power supply must be easily accessible through access door. Handler can be easily disassembled for replacing parts and electronic components. Must be able to be thoroughly cleaned.

Reliability Requirements: Handler will be responsive to summoning and sending programs with reset button for fixing coding issues. Motor, chassis, tires, and electronic components shall be high quality to ensure longevity and reliability.

Decommissioning or End-Of-Life Recycling Considerations: All components can be recycled or reused in newer models of the handler.

Product Roadmap Analysis: Please see Attachment 2 for the Product Roadmap Analysis.

Development Equipment Requirements: To be determined by Design Team by 2/16/2018. Will likely include welding equipment, drill press, electronic development equipment, and sheet metal forming equipment.

Development Tools Requirements: To be determined by Design Team by 2/16/2018. Will likely include wrenches, cutting tools, screwdrivers, adhesives, and other basic construction tools.

Design Validation and Testing Requirements: The material handler will be designed to hold more than the required weight of 100 lbs. Specifics regarding FOS will be determined by the sponsor and design team by February 16, 2018. Motors will be sized accordingly. Testing will be simple – the material handler will be loaded with approximately 150 lbs. and will be given a series of commands to test programming and performance.

Manufacturing Processes (Prototype): Basic assembly of various off-the-shelf components. The custom chassis as well as some attachments may be machined in the mechanics shop.

Manufacturing Processes (Pilot-Production): Basic assembly of various off-the-shelf components. The custom chassis as well as some attachments may be machined in the mechanics shop.

Quality Assurance Requirements (Pilot-Production): It is doubtful that many products will be produced, so each material handler produced shall be inspected individually for the first 50 units. After the first 50 units, one in every ten subsequent units will be tested at random for quality assurance.

Quality Control Requirements (Pilot-Production): Assuming the products we purchase to manufacture the material handler have undergone some sort of quality assurance during their manufacturing process, a brief visual inspection of each component will be conducted during its assembly.

Launch Quantities (Used for Pilot-Production Costs): Our team will construct only one material handler for Plenty Unlimited, Inc. However, we will include all necessary documentation such that Plenty Unlimited, Inc. will be able to replicate the material handler as many times as they see fit.

Packaging and Transportation Requirements: The goal is that these material handlers will be able to be assembled on-site, given a complete parts list, instruction manual, and full Arduino code. Many, if not all, of the parts will be ordered from online retailers, and therefore will be packaged and shipped accordingly. If parts are transported by Plenty Unlimited, Inc., proper measures should be taken so that the parts are not subjected to extreme vibration, extreme temperatures, or crushing.

Costs (Prototype): The expected prototype cost estimates are approximately \$1800.

Costs (Pilot-Production): Although it is unlikely for a high number of these products to be produced, it is estimated that the 500th unit cost would be approximately \$1000/unit. We use a 500th unit cost due to the unlikelihood of more than 1000 units being produced.

Profitability (Pilot-Production): It is estimated the handler will save Plenty Unlimited Inc. approximately \$15 to \$20 dollars per week; once the material handler pays for itself, the company will be looking at an additional \$15 to \$20 in profits per week per unit. See Attachment 3 for Return on Investment analysis.

Time to Breakeven (Pilot-Production): No more than 3 years. Preliminary ROI calculations indicated time to breakeven was approximately 100 weeks. However, as units are produced at lower costs, this breakeven time will likely decrease.

Project Time Scales and Planning (Gantt chart with Work Breakdown Structure (WBS)): Please see attached Gantt chart (Attachment 4) and WBS (Attachment 5).

Key Events or Proofs Required:

- Detailed notes of design meetings/ideas and progressive design iterations
- Thorough documentation of design (including but not limited to detailed CAD model, BOM (bill of material), maintenance instructions, and a user manual
- Construction of a small scale working prototype for proof of concept
- Final report outlining full design process and final design

Anything Considered High-Risk or Roadblock (Prototype & Pilot-Production): The main potential roadblock in the design is coding the microprocessor to interpret machine vision, and then integrating the programming with the electric motor and steering components.

Any New Technology that could make this Product-Project Better: Voice activation, more precise GPS systems.

Brand or Branding Considerations: This product will not be its own brand. It will likely include components from several other existing brands.

Marketing Plan: The product will not be marketed; it will simply be provided to the sponsor and stakeholder, Plenty Unlimited Inc.

Voice of the Customer Plan: We have maintained open lines of communication with our contacts at Plenty Unlimited, Inc, in order to stay current with their expectations and ideas for product improvement.

Control Plan: Start with the programming of the microprocessor; the design specifications for the programming have been well defined already and are least likely to be changed. Once programming is complete, it can be integrated to a handler design that is compatible with the already coded software.

Attachments:

1. FMEA Spreadsheet
2. Product Roadmap Analysis
3. Return on Investment Analysis
4. Gantt Chart
5. Work Breakdown Structure

PDS Attachment 1. FMEA Spreadsheet

Failure Modes and Effects Analysis								
Process Step	Potential Failure Mode	Potential Failure Effect	Potential Causes	SEV (Severity)	OCC (Frequency of Occurrence)	DET (Probability of Detection)	RPN (Risk Priority Number, calculated as SEV*OCC*DET)	Action Recommended (Actions for reducing the occurrence of the cause or improving its detection)
User Inputs	Motors do not start	Non-functional machine	Wiring disconnection (remote-microcontroller-motors/battery)	5	2	4	40	Secure connections of wires in all locations - soldering where possible
			Dead battery	1	4	1	4	Use two batteries - charge one while the other is in use
			Not receiving remote summoning signal	8	2	6	96	Ensure summoning device has ample power to communicate to the handler, and that no obstacles are interfering
Handler Motion	Handler does not move	Non-functional machine	Disconnection between motor shafts and wheels	5	1	5	25	Wheels should be attached to the shafts as robustly as possible. Periodically ensure proper fit of the wheels.
			Obstacle blocking handler	1	2	1	2	Ensure no obstacles are in the handler's path
	Handler moves too slowly	Inefficiency	Drained battery	1	6	2	12	Replace with the fully charged battery
			Weight overload	1	3	2	6	Remove weight until handler functions properly
	Handler does not stop at designated location	Incorrect functionality - handler wastes time, and is dangerous	Location detection sensors blocked/disconnected	10	1	7	70	Secure connections of wires in all locations - soldering where possible. Ensure all sensors are not blocked in any way
	Handler does not turn corners	Damage to handler or property of Plenty, Inc.	Location detection sensors blocked/disconnected	7	1	7	49	Secure connections of wires in all locations - soldering where possible. Ensure all sensors are not blocked in any way
Weight overload - too much friction for handler to rotate			4	2	3	24	Remove weight until handler functions properly	

PDS Attachment 2. Product Roadmap Analysis

September 2017	October 2017	November 2017	December 2017	January 2018	February 2018	March 2018	April 2018
PHASE 1—Preliminary Design and Brainstorming			PHASE 2—Programming and Construction			PHASE 3—Testing	

Programming and Electronic Components

<ul style="list-style-type: none"> • Practice programming with Arduino • Identify best microprocessor to use with image processing • Research image processing and source code for microprocessor • Test possible code to verify application • Identify other necessary electronic components • Order all necessary components 	<ul style="list-style-type: none"> • Create pseudocode to direct program in each anticipated situation encountered in the warehouse • Integrate pseudocode into working program for machine vision • Create electronic assembly to carry out the code • Integrate steering and driving motor to electronics 	<ul style="list-style-type: none"> • Test each situation previously identified to prove code works • Identify any errors and fix programming • Seek opportunities for optimization
--	---	---

Physical Prototype

<ul style="list-style-type: none"> • Finalize all desired design specifications • Research food safety standards and brainstorm possible designs • Compare possible designs and rank • Order necessary components • Note components to be self-manufactured 	<ul style="list-style-type: none"> • Manufacture necessary components • Assemble prototype in accordance with all specifications and standards • Verify all components fit and mesh as anticipated in design • Replace any components that do not fit as anticipated with revised components 	<ul style="list-style-type: none"> • Load prototype to design specifications and ensure proper performance • Identify any issues and resolve • Seek opportunities for optimization or weight reduction
--	--	---

PDS Attachment 3. Return on Investment Analysis

Return on Investment Analysis (ROI)				
Item	Cost	Parameters		
HDPE Sheets	\$220	Avg Walking Speed	3	mph
Threaded Steel Rods	\$30		4.4	ft/s
Li-Ion Battery	\$60	Avg Round Trip	100	ft
12V Motors (4)	\$120	Avg Trip Time	22.7	s
Wheels (4)	\$30	Hourly Wage	\$ 20.00	\$/hr
Misc. Electronic Parts	\$80	Cost per Trip	\$ 0.13	\$/trip
Misc. Hardware	\$160	Trips per Day	30	trips
		Trips per Week	150	trips
		Cost per Week	\$ 18.94	\$/wk
Total	\$700	Weeks Until Paid	37.0	weeks

PDS Attachment 5. Work Breakdown Structure

