Introduction to Engineering Design with Professional Development 1

Final Report for Autonomous LCD

A completely autonomous floor scrubber used to simplify the cleaning process.

Team: Autonomous LCD
Section 15
Instructors: Paul Moon, Karyn Dyer

Due May 2, 2018

Authors

Grant Tragni (2020 / Industrial and Management Engineering)
Joseph Catalano (2020 / Computer Systems Engineering)
Jonathan Corbin (2020 / Aeronautical Engineering)
Daniel Ray (2020 / Electrical Engineering)
Haocheng Zheng (2021 / Mechanical Engineering)

Executive Summary

Autonomous LCD (*Little Cleaning Dude*) is an innovative robotic cleaning prototype that is designed to scrub floors in an indoor environment in order to save people time and energy. The prototype is broken down into five subsystems: motor control, cleaning system, room navigation, fluid control, and external communication. The motor control is the system that controls all of the motors involved in helping the Autonomous LCD to work. The cleaning system is primarily responsible for the actual cleaning aspect of the robot in which a sponge attached to a motor scrubs out any dirt on the floor. The room navigation system relies on the motor power delivered to the wheels and the ultrasonic sensor used to detect walls in front the Autonomous LCD. The fluid control system is primarily responsible for distributing the cleaning fluid onto the floor. The external communication system allows the Autonomous LCD to communicate easily with the user.

The team began work by first researching existing products on the market in order to see if there were any robots like Autonomous LCD. This research was useful in that the team found no existing products that scrub floors. The only products found similar to Autonomous LCD were products such as Roomba, iLife, and iRobot. The major difference with Autonomous LCD and the similar products to it were that no other robot scrubs floors, but other products actually only vacuum the floor. Each team member was given a subsystem to work on, but each team member also had to work in harmony with the other members to combine each aspect of the project. Testing and data collection was completed during subsystem development as well as after the final prototype had been designed.

Autonomous LCD was a functional model with every subsystem working together well. The prototype was tested repeatedly to ensure safety and that the Autonomous LCD performed to the specifications set forth. The team feels that a device similar to the Autonomous LCD could revolutionize cleaning and retire the traditional mop.

Table of Contents

Introduction:

Cleanliness is an essential part of healthy living. However, in today's fast paced world, people find there is not enough time in the day to clean thoroughly. Inadequate cleaning habits allow dirt and bacteria to build up on the floor creates an unsafe environment that facilitates disease and illness. Currently, a variety of machines exist that are designed to help people keep floors clean. Equipment such as mops, brooms, brushes, and the more modern designs, such as the Roomba, serve the purpose of assisting individuals and families in maintaining a safe and healthy environment. However, each of the previously mentioned machines have individual strengths and weaknesses.

The broom and the mop are some of the simplest options to ensure a clean home. However, items such as the broom and mop have become outdated and dormate due to the energy the items take to use. The natural improvement on the broom is the vacuum cleaner but even that is time consuming and tiring to use. Also, these items perform the same task, but they still fail to completely clean the floor. In order to disinfect the floor completely, a mop is needed to wipe away dirt and grime after a broom removes larger dust and debris.

As technology advances, so too do the ways in which society maintains safe and clean environments. The Roomba and iLife are the newest way to keep floors clean. These autonomous robots will drive around the floor and clean up dirt and dust. While the iLife can only vacuum carpets, the Roomba has a wider range of abilities to handle both carpet and hardwood floor. Advancements in robotics can take a great deal of strain off of any individual that wants to keep floors clean. By simply leaving these devices on, the room becomes cleaner. Having cleaning tasks be performed by robots significantly reduces the amount of manual labor involved in cleaning. However, the limitations involved in the more advanced technology become readily apparent when compared to similar shortcomings in previous technology. Just as a broom can only remove dust and debris, so too can the iLife and Roomba only clear small amounts of dirt left on the floor. Furthermore, just as the broom had the mop, the iLife and Roomba have the iRobot.

The iRobot functions as the modern counterpart to the mop. It has a pad on the underside and is used to clean up spills like a normal paper towel would. However, the iRobot is not without its own flaws; the greatest of which being that it cannot clean as deeply as a mop would. Without a strong motor system there is no way for the robot to clean into tougher grime and dirt that a mop would be able to. In this way the iRobot is not a suitable counterpart to the Roomba and iLife as it cannot act as the corresponding wet cleaner to their dry cleaning.

The next development in cleaning would be an autonomous robot that can provide the scrubbing power needed for a deep clean. The group came together to build a robot that would allow people to sit back and relax while a robot scrubs the floors.

Individuals would no longer have to settle for the cleaning power of the iRobot and would no longer need to rely on the time and exertion involved in using a mop to clean a floor by hand.

The Autonomous LCD would function as a necessary floor scrubbing robot that would provide the safe, clean, grime free environment that people need to flourish.

Objectives and Scope:

The team decided upon a variety of objectives which are provided below:

- Design a cleaning robot that can clean floors efficiently and finish the job quickly
- Meet all customer requirements and technical specifications
- The robot is user friendly
- Complete the project within the due date
- Fully understand and utilize the design process throughout the project
- Fully utilize each team member’s strength and improve as a team
- Obtain an “A” for the final grade

Mission Statement

The purpose of Autonomous LCD is to fill every households need of a cleaner environment with an autonomous floor scrubbing system. Additionally, team members will learn skills in areas in which they might be less comfortable to complete the project in a timely manner.

The team has set out to create an entirely autonomous cleaning device that has the main capability of getting rid of any mess on the floor.

Customer Requirements/Specifications

Customers for the Autonomous LCD are primarily going to include busy individuals who find that there is little time in the day for cleaning floors. The customers could be anyone from busy college students to families with children. The Autonomous LCD intends to efficiently tackle messes in a large variety of environments, making it suitable for a large customer base. However, the Autonomous LCD will be limited to customers that have hard floors, not carpet. By keeping a low production cost, the Autonomous LCD should prove to be inexpensive to the average consumer, widening the customer range to include people with little budget room for autonomous cleaning robots.

Safety is an essential part of the design that must placed under great scrutiny regardless of the intended customer base. However, the Autonomous LCD will be built with the nuclear family in mind as the primary baseline for safety standards. This means making the Autonomous LCD safe for adults by making it not trip people or make floors excessively slippers. Furthermore, it must be safe for children by being sturdy so that it cannot be knocked over and fall on them and it must use a nontoxic cleaning solution so that young children who are often on or close to the floor will not be in any danger from accidental ingestion. In general, the Autonomous LCD should have quite a large customer base due to how commonplace mopping is in today’s society.

The target audience for the Autonomous LCD is directed towards busy homeowners who do not have the time or energy to clean hard floors. Team members asked peers, family, and others about what they would want out of an autonomous cleaning product through a Google survey. The survey included questions such as: “Do you have time to clean your floors?”, “What are some features that you would want in a robotic floor cleaner?”, and “Would you spend money on an autonomous floor cleaner?”. Due to the answers the team received, the Autonomous LCD group decided to go forth with the idea.

Table 1: Customer Requirements and Technical Specifications

Customer Requirement	**Technical Specification**	**Target Value / Range of Values**
Effectively cleans floor	Efficiency	90% of dirt cleaned
Finishes cleaning quickly	Speed of cleaning	1 hour cleaning time
Doesn't hurt users	Safety	0 casualties
Non-toxic to the user	Safety	0 illness'
Ground covered	Area of flooring	$100ft^2$
Easy to operate	Age	~10+ years
Convenient/Simple	Number of moving parts.	3
Portable	Weight	6-10lb
Not too large	Volume	1ft-1ft-6in
Inexpensive from a college student's perspective	Dollars per person	50-100$
Long lasting	Time before failure arrives	2 years
Fluids easily replaceable	Accessibility	Takes less than 1 minute to refill
Battery Easily Rechargeable	Time	1 hour
Low risk of shock	Percent chance of getting shocked	Less than 1%

Concepts Selection & Development

Brainstorming:

To start brainstorming on a project idea, the team members were told to go through a full day and see what problems the team members ran into. The team then met in order to look over the ideas and talk about which ones the team could pursue. After an hour, there were a total of ten ideas on the list ranging from sustainability solutions to assistive technologies for the disabled. The team then began to narrow the list down to select one to solve. The top three ideas the team came up with were a fully autonomous wheelchair that could climb stairs, workout equipment that could generate energy, and an autonomous floor scrubber. These ideas were researched by the team and ranked based on originality and if the team would be able to complete it. After the team had further discussion regarding the task at hand, the team decided that the autonomous floor scrubber was the best idea due to its feasibility and potential usage.

The idea for the Autonomous LCD came from another group brainstorming session. The group came to agreement that an autonomous floor scrubber would help alleviate the age-old problem of dirty floors. From there, the team moved to brainstorming the best ideas of how the task could be accomplished.

Concept Combination:

Existing products on the market are similar to the Autonomous LCD, but none quite tackle the issues in the same way the Autonomous LCD intends to. Existing products that are similar to the Autonomous LCD include the Roomba, iLife, iRobot, and mops. Out of the current products, only the iRobot and a mop can be utilized to clean floors in a similar way to the Autonomous LCD. In addition, the Roomba, iLife, and iRobot all have an autonomous functionality that will be similar to that of the Autonomous LCD.

The uniqueness of the Autonomous LCD combines the autonomous functions of the Roomba with the scrubbing aspect of a mop or the iRobot. The Roomba and iLife fail to mop/scrub floors and only vacuum particles off the floor, while the iRobot and mops scrub floors, but are expensive, time consuming, and have poor autonomous functionality. The Autonomous LCD integrates the best aspects of those four products, while improving on those aspects in a new and unique way.

Table 2: Concept Selection Matrix

Design Type	Hexagon Shape	Circular Shape	Acrylic	Wood	Circular Brushes	Rolling Brush	Cheese cloth dryer	Squeegee
Cost	1	1	1	0	1	1	-1	1
Difficulty	1	-1	0	1	0	0	1	0
Creativity	1	-1	1	0	-1	1	1	0
Efficiency	0	0	0	-1	0	0	1	-1
Sustainability	0	0	1	-1	1	0	1	-1
Safety	0	0	0	0	0	1	0	0
Size	0	0	0	0	0	1	0	0
Durability	0	0	1	1	0	1	-1	1
Net Score	3	-1	4	1	1	5	2	0
Rank	1	2	1	2	2	1	1	2

As seen in (*Table 2*), the team's original design choice was the hexagonal shape, mainly because of the creativity (as the shape is not like any other product), but also because a hexagon shape allows for the Autonomous LCD to go into tight corners and clean anything collecting in the corners. However, during construction, the chassis shaped was changed from a hexagon to a square. The change allowed for a more efficient build process and saved time. For the body material, acrylic was preferred over wood because of the durability and resistance to liquid that acrylic provides. Wood would soak and rot, leading to the design eventually reaching a point where catastrophic failure would occur. For the the floor scrubber, the team decided to use a rolling brush since rolling brushes can be lowered down to the floor to maintain constant contact with the floor, providing even and thorough cleaning for the duration of the Autonomous LCD's lifespan. The table also shows that a cheesecloth drying system would be used, but the component was removed during the build process to simplify construction and to save time.

Table 3: Concept Combination Table

Design Type	Method	Use
Scrubber	Continuous drum	Spin drum with high torque to clean floor
Liquid Tank	Holds cleaning fluid	Spray cleaning fluid
Motors	Provides mechanical movement	Spins wheels, scrubbers, cheesecloth

As seen in (*Table 3),* the concepts that the team decided on will all coexist and work together. The scrubber will be a rotating drum covered in scrubbing material.The liquid tank will be used to hold all of the cleaning fluid and there will be spray nozzles to release the cleaning solution directly onto the scrubbing brush. Lastly, the motors will be the power behind the robot and will put the wheels and scrubber into action.

Concept Sketches

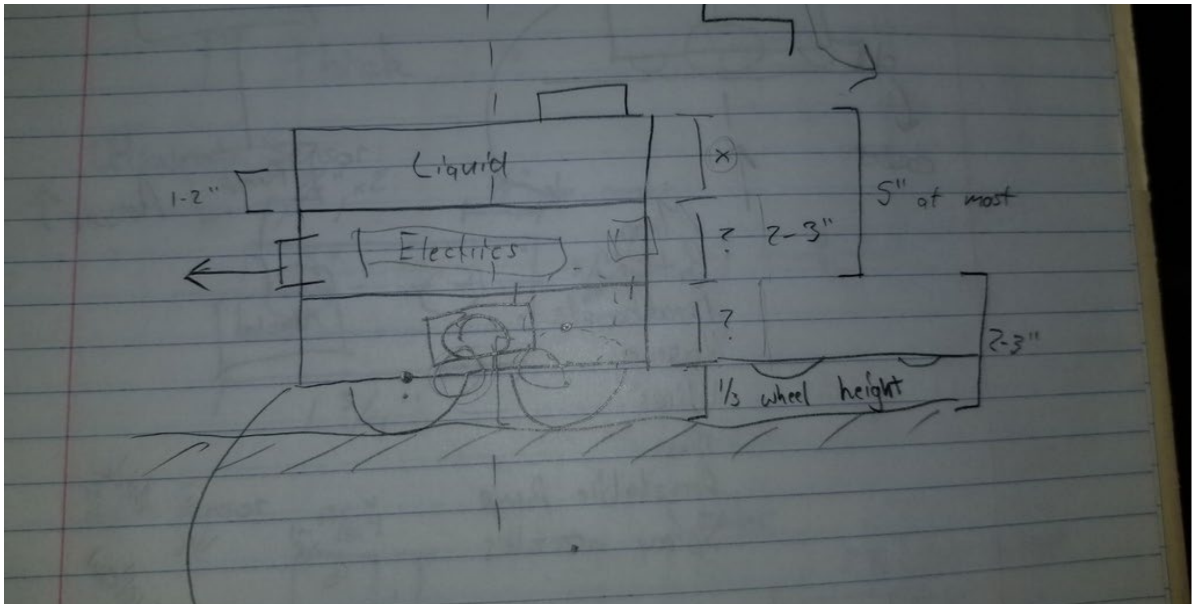


Figure 1 - Concept Exterior Dimension Sketch

Figure 1 was the first preliminary design with the idea of putting an idea on paper to see how it would look. The design included three horizontal panels of the robot: one liquid tank, one with electrics, and one with the wheels and spray nozzles of the bottom. The top panel would be the liquid tank which would be filled with a solution designed to break apart dirt. The middle panel would hold the motors, arduino, and most of the wiring, and the bottom panel would include the wheels, some type of cloth to pick up the excess solution, spray nozzles, and the scrubber. As a team, the first sketch was known to be an extremely rough sketch just to see what the team had come up with.

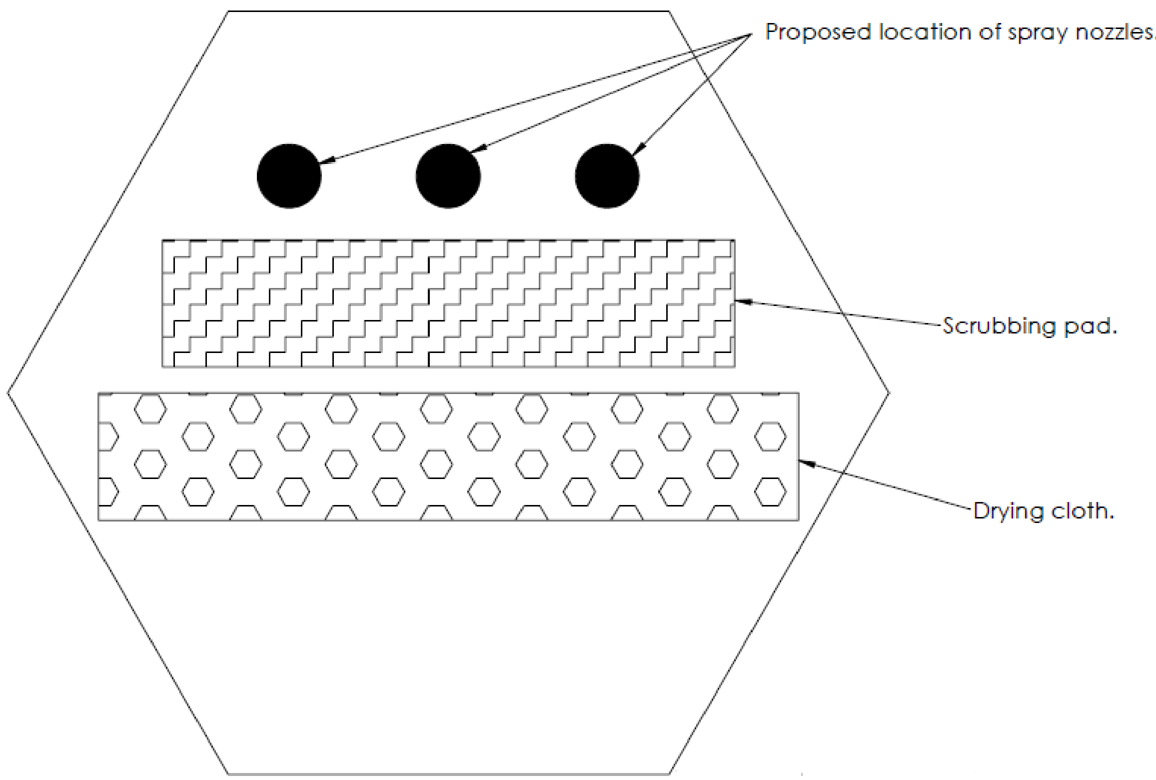


Figure 2 - Bottom Layer Concept Sketch

Figure 2 is what the team's original base of the Autonomous LCD would look like. The figure is a very basic sketch that included three spray nozzles, a scrubbing pad, and a drying cloth. There were no wheels put on this sketch due to the fact that the team originally wanted to place the wheels on the exterior on the sides. The team was still unsure of what material to use for a drying cloth, and what material was needed for a scrubbing pad.

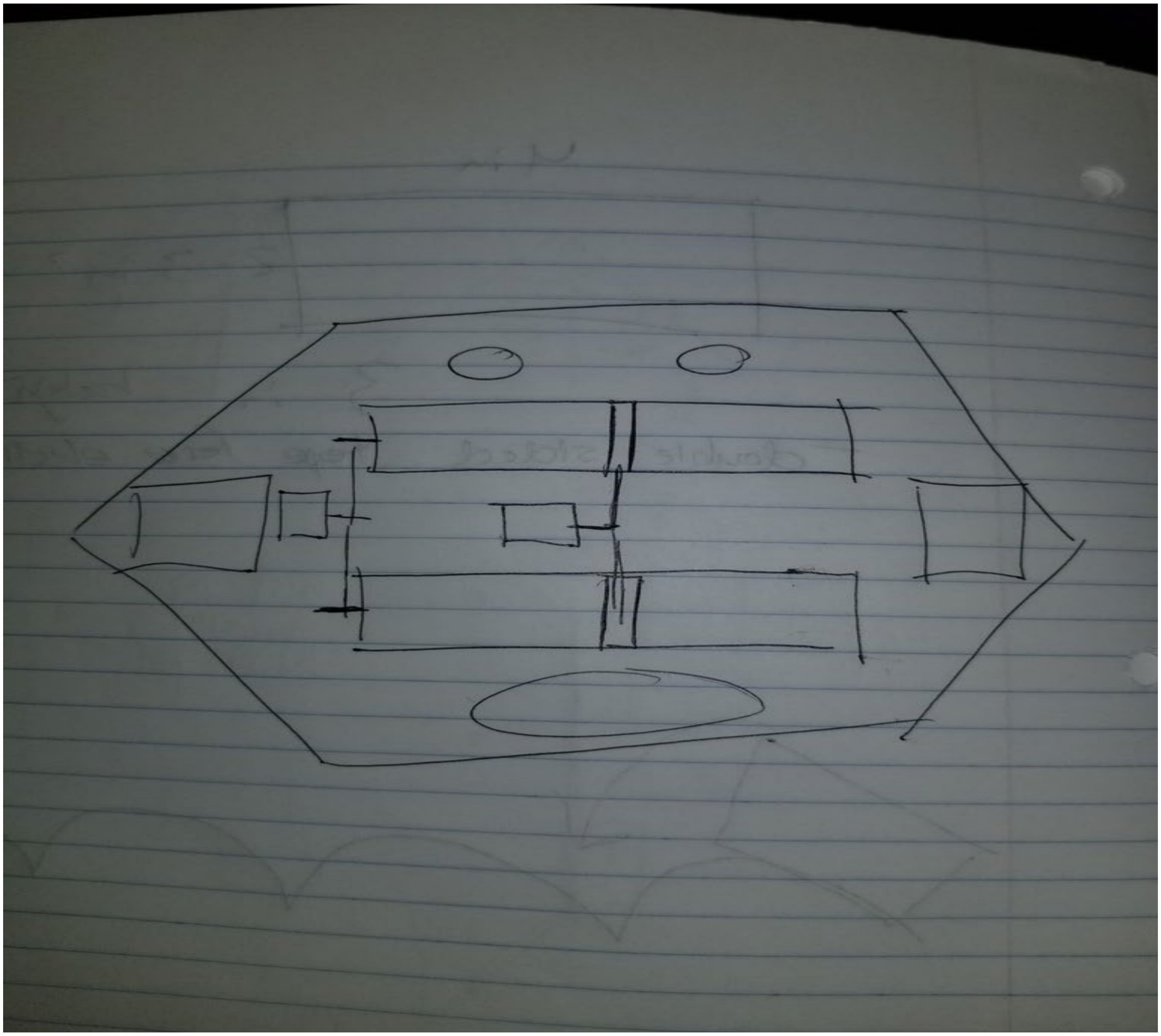


Figure 3 - Bottom Layer Concept Sketch

Figure 3 includes the two wheels now on the bottom layer, a drying cloth, a scrubbing pad, and two spray nozzles. The issue with the above design was that the team did not have enough room on the acrylic sheets to fit all of the materials on the bottom. Due to the overcrowded nature of the figure, the team decided to remove the drying pad to save space.

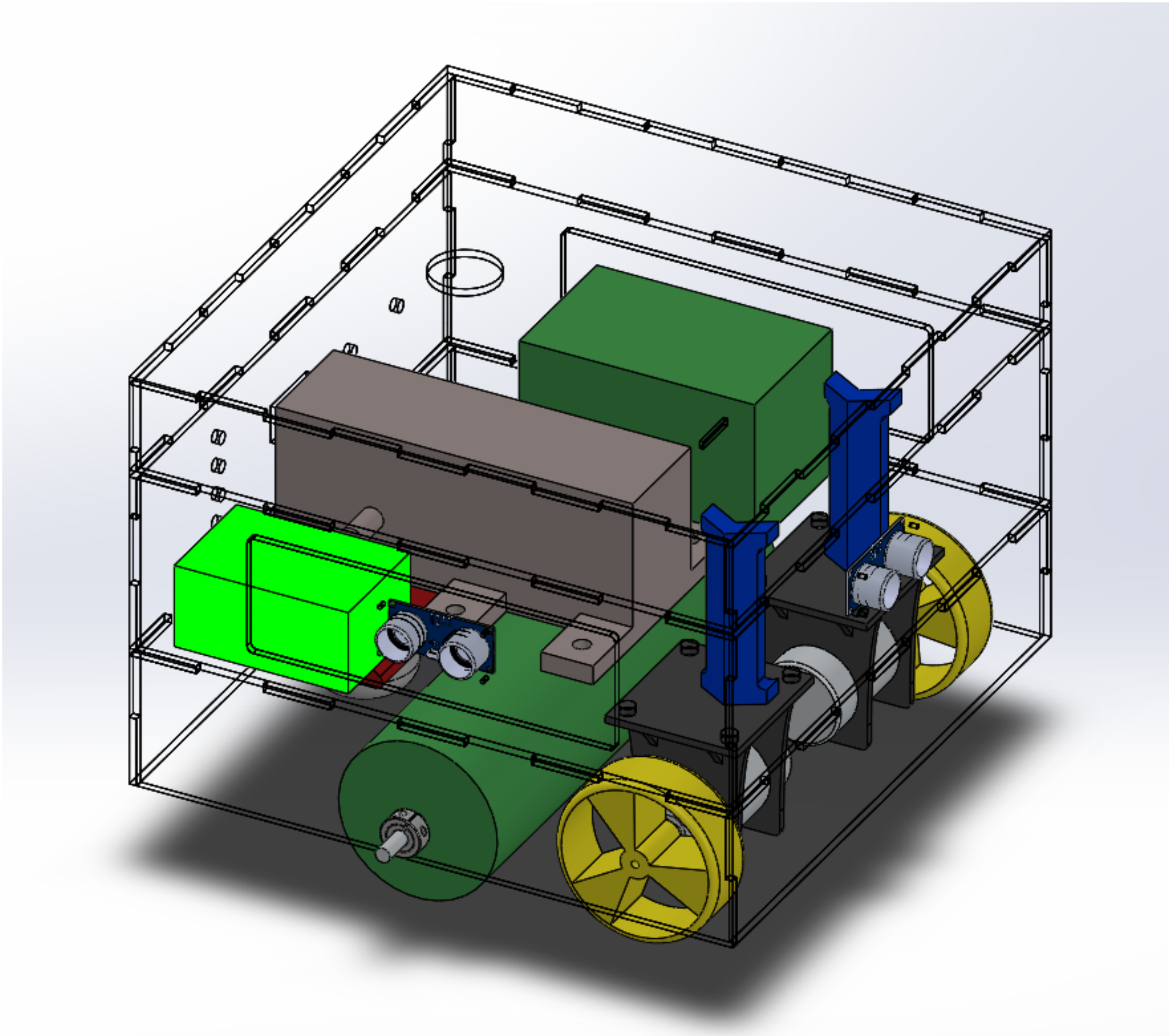


Figure 4 - Final CAD

Figure 4 is a close approximation of the final design. The sketches have since moved from being hand drawn to a CAD *(computer-aided design*) format. Several design changes were implemented, including changing the overall shape of the robot. *Figure 4* shows the robot from the front side.

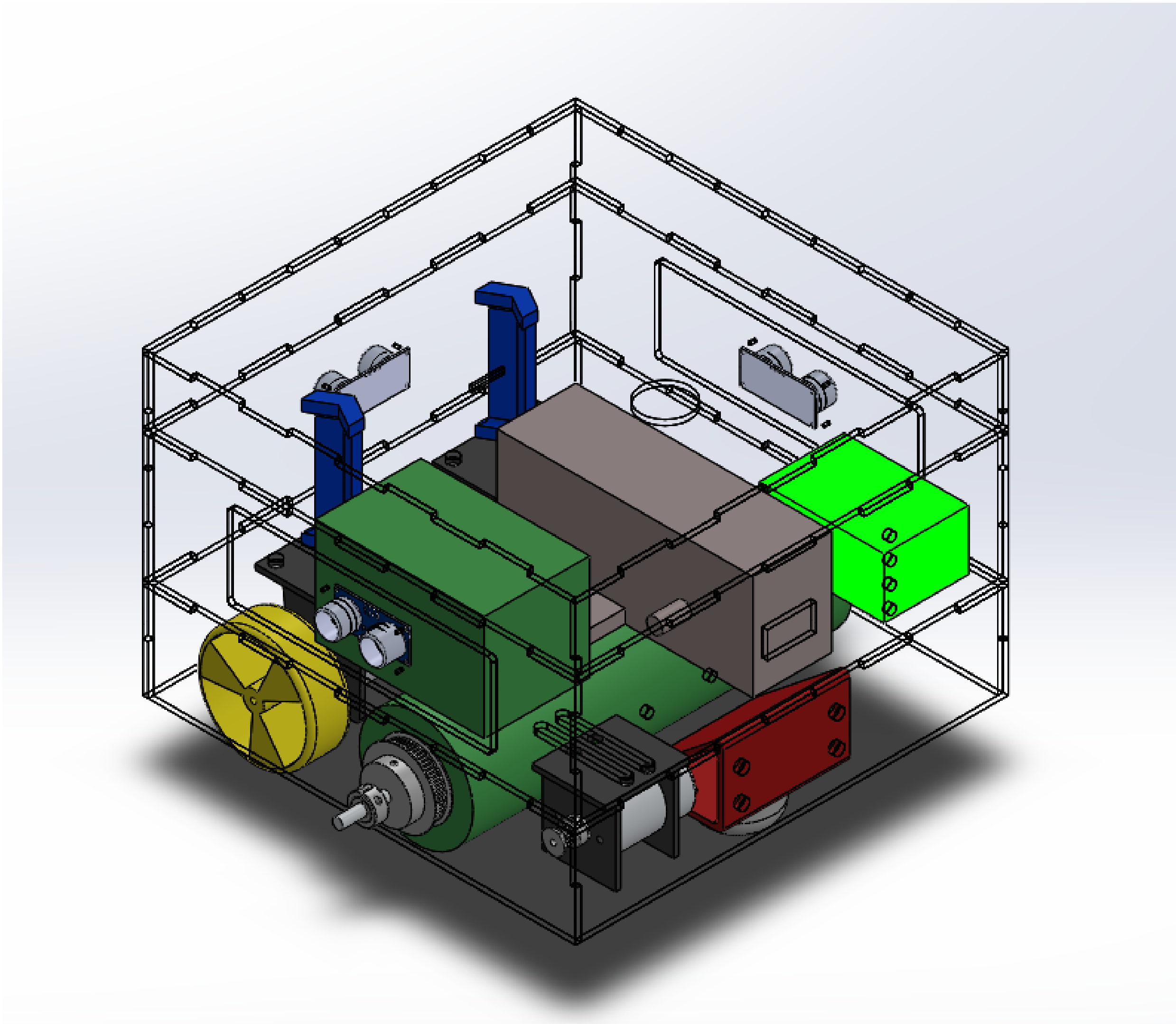


Figure 5 - Rear View CAD

Figure 5 is nearly identical to *Figure 4*, but rather shows the robot from the back.

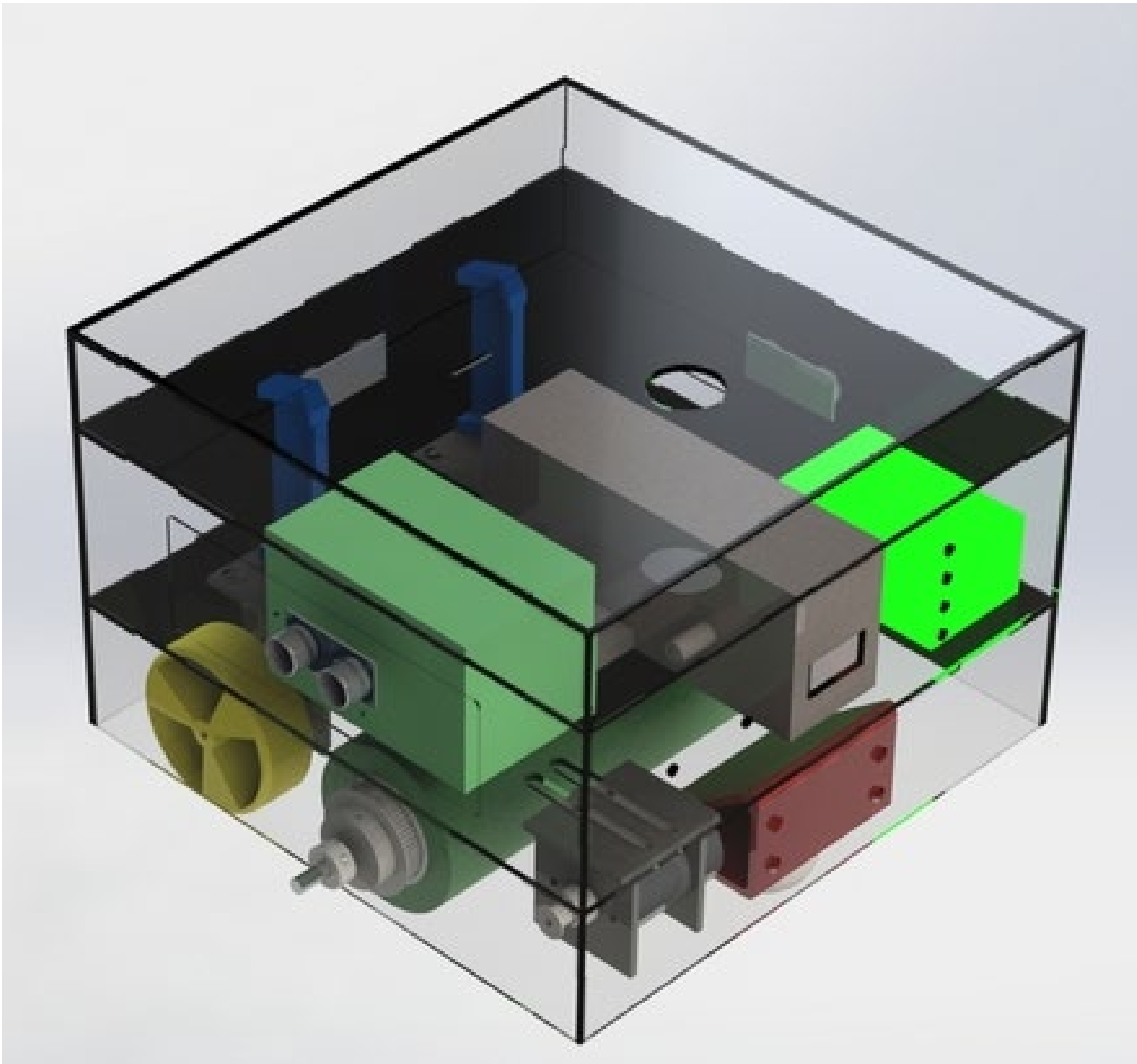


Figure 6 - Rendered CAD

Figure 6 shows the same view as *Figure 4*, but the material has been rendered using a process to provide more realistic textures and to gain a better understanding of how the completed robot will look and feel to an average user.

Subsystems:

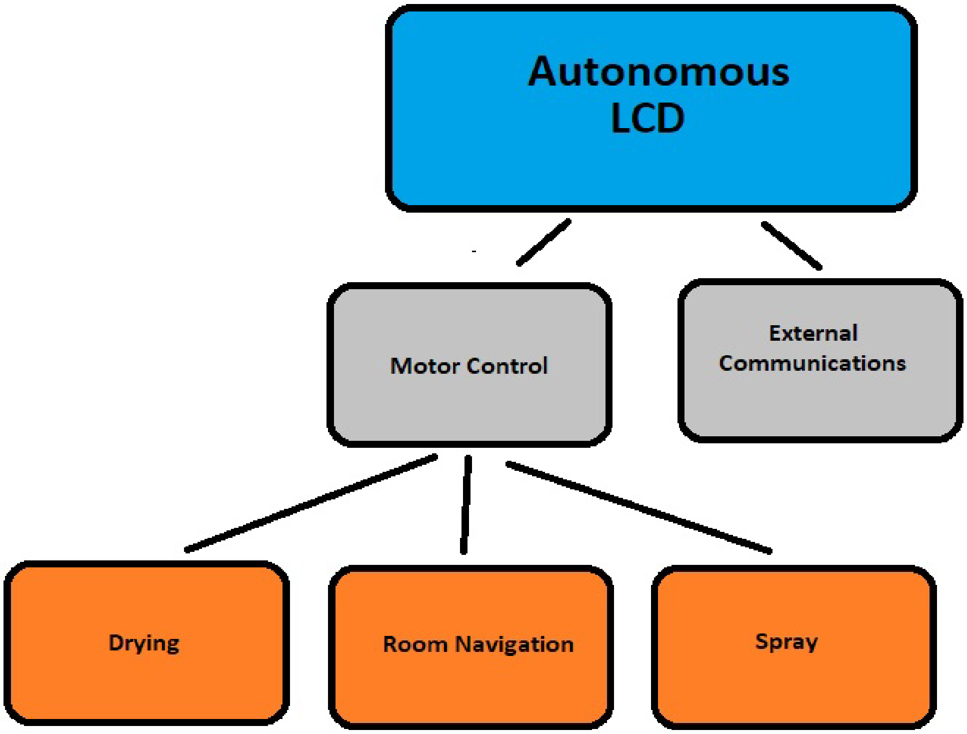


Figure 7 - Subsystem Hierarchy

Overview

The best solution available to the problem is an automated robot that will scrub floors using a unique cleaning solution. In order to accomplish those tasks, the robot, called the Autonomous LCD, will drive across the entire room and clean all of the floor that is travelled over. The Autonomous LCD does accomplishes the task by dispensing a light mist of cleaning fluid that will accompany a scrubber to clean the floor. The fluid will be alcohol based, non-toxic, and a strong disinfectant. Once the fluid has been spread onto the floor, a scrubber will be spinning just behind the application point of the fluid. The scrubber will make sure that the fluid is utilized efficiently, ensuring that the Autonomous LCD cleans any dirt, grime, or sticky messes that are present on the floor. Instead of the leaving the floor wet, the machine will then use another system to dry the ground behind the scrubber. The Autonomous LCD will also have a system to determine ways to drive across the floor in order ensure even and thorough cleaning while avoiding any possible collisions or falls off stairs. Furthermore, the system by which the motors run will be crucial to helping the Autonomous LCD drive, maintaining the drying system, and powering the pump that will allow the Autonomous LCD to distribute the fluid. Lastly, the Autonomous LCD will use series of LEDs to communicate with the user so that the user will know when something in the machine needs to be replaced.

Room Navigation

Objective:

The room navigation system used in the Autonomous LCD is meant to be simplistic in nature. This system relies on the motor power delivered to the wheels and the ultrasonic sensor used to detect walls in front the Autonomous LCD. This is essential to prevent the Autonomous LCD from completely stopping once it reaches an obstacle. This requires a solid detection system that makes sure the device does not have any trouble with the walls in front of it. Beyond this it must turn evenly and carefully to prevent the Autonomous LCD from being offset from its desired path. This will require appropriate motor control and carefully measured pulsewidth modulations. Lastly, the turning must be done at a speed that allows for the Autonomous LCD to clean the floor evenly as it is turning.

Customer requirements

The Autonomous LCD is not like most other floor cleaning robots. Since the Autonomous LCD uses fluid to clean floors, it must have a different method by which it cleans the floor for the sake of customer convenience. The erratic path used by the Roomba would leave an irritating pattern of liquids around the floor. Furthermore, similar products have unpredictable paths that cause them to unexpectedly bump into users. By making the path easily predictable, these collisions could be avoided.

Design Process/Concept Selection:

The ideal path for the Autonomous LCD is a grid shape. This is very different from the erratic patterns used by devices such as the Roomba, but it is essential given the very nature of this device. Since the Autonomous LCD uses fluid to clean the floor, an erratic shape is undesirable for a variety of reasons. Non-uniform paths will mean that the device is cleaning the same locations multiple times. This will result in the Autonomous LCD using up things like battery power and cleaning spray too quickly. Furthermore, since the Autonomous LCD will leave behind some fluid on the ground once it cleans, it would leave behind many irritating wet patches that customers would not like. The grid shape, shown in *Figure 8* and *Figure 9*, will still leave behind wet patches, but the locations of these patches will be predictable. The predictability of the grid design also means that it will not clean the same places more than once, and users will know where it is going to clean next so they can avoid it. Predictability is important in this case because the object is spraying fluids. Erratic motions are not possible given the very nature of the Autonomous LCD.

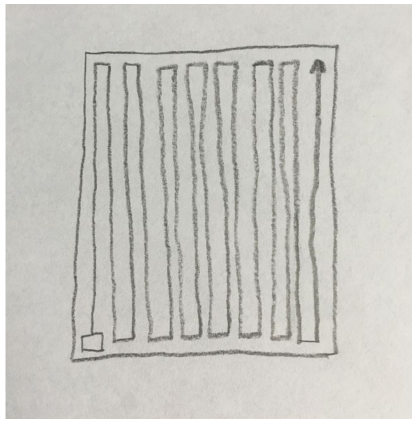


Figure 8 - First Half of Grid Path (No Obstacle)

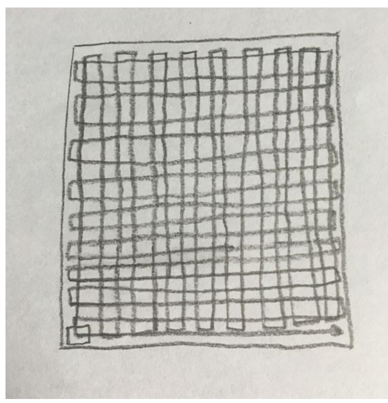


Figure 9 - Full Grid Path (No Obstacle)

Part of the design that changed the most was how the system would deal with obstacles. Initially, the team had planned on having the Autonomous LCD drive around each obstacle it encountered so as to efficiently cover the largest amount of floor space. However, as development continued, the team realized that this method was overly complicated and more likely to fail than other methods. The method that wound up becoming more useful to the group involved treating all objects as walls. This concept is demonstrated in *Figure 10* and *Figure 11*. By doing this, the robot will encounter obstacles then turn around and continue as if it had just come in contact with a wall at the end of the room. Then, once the Autonomous LCD reaches the end of the room in the final corner, it will stop and turn itself to move in another grid perpendicular to the one that it had just completed. By doing this the design can be greatly simplified and the likelihood of the design missing many areas of the floor is greatly decreased.

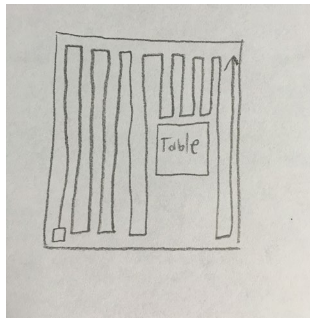


Figure 10 - First Half of Grid Path (With Obstacle)

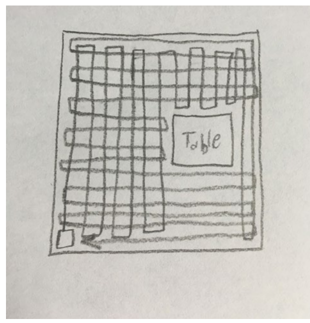


Figure 11 - Full Grid Path (With Obstacle)

Lastly, the driving subsystem relies heavily upon the ability of the Autonomous LCD to orient itself accordingly and to locate all nearby obstacles. In order to do this properly the team found it optimal to add additional ultrasonic sensors so as to allow for the Autonomous LCD to view its surroundings from multiple sides. By adding additional sensors on the sides of the Autonomous LCD, the robot will no longer have to turn 90 degrees to know what is next to it. This gives the Autonomous LCD fewer turns where it has the likelihood to move slightly away from its desired path. Furthermore, this saves time and code complexity that allows this to be more desirable for potential customers. Also, maintaining simple code allows for future designs to be added and adjusted easily by creating fewer areas where the code can go wrong. However, the Autonomous LCD has other methods that it uses to maintain its path. The method that the group went with was to use the ultrasonic sensors on the side of the Autonomous LCD to allow the robot to track and follow a straight path along the wall so as to prevent the Autonomous LCD from deviating from its desired course at the beginning and end of its journey.

Dimensions:

The best metric for this system is whether or not all of the code can successfully be downloaded onto the Arduino’s limited code storage space. The maximum code storage space for the Arduino was 250 KiloBytes. The Room Navigation fit well within this limit.

In addition to code requirements, the Room Navigation subsystem also requires that the Autonomous LCD track nearby walls at a set distance of 15cm. This is to make sure that the Autonomous LCD maintains a straight line of motion and does not collide with nearby walls. This distance was properly maintained after being examined in testing.

Subsystem Testing:

In order to test the room navigation subsystem, a variety of things had to be checked. First, this subsystem would not be able to function if the Autonomous LCD could not detect obstacles in its way. Therefore, the first things that needed to be tested were the ultrasonic sensors. Individually testing these sensors would allow the team to guarantee that the robot could detect walls and adapt to them.

The remainder of the room navigation subsystem comes down to how accurate the robot is in following the assigned path. The Autonomous LCD's ability to remember previous turning directions and follow the programmed grid-shaped path as well as its ability to adjust its course based on its collisions detection software and hardware are the main basis for this subsystem. Additionally the Autonomous LCD must know when it has finished the first half of its journey so that it can begin the second half, and when it has completed its job and covered the entirety of the room.

Integration:

This subsystem relies heavily on other subsystems to function. Testing for this system could not begin in earnest until the robot had been constructed, as many variables had to be tested and modified based on the performance of the Autonomous LCD. Due to these prerequisites, integration for this portion of the Autonomous LCD was one of the last to occur. However, since this portion of the Autonomous LCD was the last to be added to the project, it was the most flexible as it was the part expected to change to the needs of other subsystems.

The Room Navigation subsystem was designed and coded alongside the creation of other subsystems so as to minimize integration time. As other subsystems were finished, so too was the necessary integration of that subsystem with the room navigation subsystem. Once the entirety of the Autonomous LCD had been completed, the Room Navigation subsystem was added. Once the Room Navigation subsystem was added it could begin testing and modification to guarantee functionality.

Challenges:

The main challenges that the team faced in the development of the driving subsystem are: covering the largest area possible, covering a large space efficiently, and managing collisions with walls and objects in a room. Each of these challenges led to a different area of focus in the design. Covering a large area led to the design where the Autonomous LCD doubles back on itself to clean perpendicular to its original path. By using this method, the Autonomous LCD will be able to cover most of a given floor so long as the layout of furniture is not such that the Autonomous LCD cannot enter the area between pieces of furniture. By using a grid method, the Autonomous LCD will be able to cover its cleaning space efficiently. Using a more erratic path would have been inefficient, as the Autonomous LCD would be more likely to accidentally cover the same spots on the floor several times over. Lastly, the Autonomous LCD deals with collisions with furniture in the same way that it deals with collisions with walls. The Autonomous LCD contains several ultrasonic sensors. The Autonomous LCD can use these sensors to locate nearby walls and furniture and adjust its course by turning around and continuing its grid pattern whenever it detects furniture or walls.

Fluid Control

Objective:

The core objective of the fluid control subsystem is to provide fluid to the cleaning system to ensure thorough cleaning is achieved. To provide an even flow of fluid on demand, the fluid control system contains a liquid holding tank, tubing, and a pump. The chassis is built in such a way that the water tank is an extension of the chassis itself, rather than a separate tank being added to the chassis. As such, the chassis also falls under the fluid control subsystem.

To accomplish the required tasks for the subsystem, a pump transports liquid via tubing from a tank in the chassis to a cleaning system. Above the cleaning system, liquid is sprayed through minute pores in the tube to provide even coverage of liquid over the scrubbing wheel. By controlling the speed and duration of the pump, the amount of liquid used in the system can be accurately deduced and the perfect amount of fluid can be applied to ensure proper cleaning without excess fluid.

Customer Requirements:

As with any subsystem, core customer requirements are affordability, safety, and efficiency. The fluid control system needs to supply just the right amount of liquid. Supplying too much liquid will result in a wet floor and increased costs by needing to supply more fluid, while too little fluid will result in floors being insufficiently cleaned. In addition, the fluid needs to be an adequate cleaner and disinfectant, without being harmful to humans or flooring.

In addition to the basic requirements, the survey the team sent out at the beginning of the project returned several other key requirements, including: an easy way to refill the liquid, a quiet pump, lightweight, and durable over long periods of time. The fluid control subsystem aimed to tackled all of the specified requirements in a simple and elegant manner.

Design Process/Concept Selection:

Initial designs for the fluid system started out with a concept selection matrix (*Table 2*) the team discussed. From the initial concepts, a system was developed involving a small tank of water adjacent to the electronics. A pump would then be used to transport the liquid from the tank to three small spray nozzles. The spray nozzles would directly apply a fine mist of fluid to the floor directly in front of the robot (*Figure 12*). After brief consideration, the initial design concept was determined to be inadequate and a new design was conceptualized.

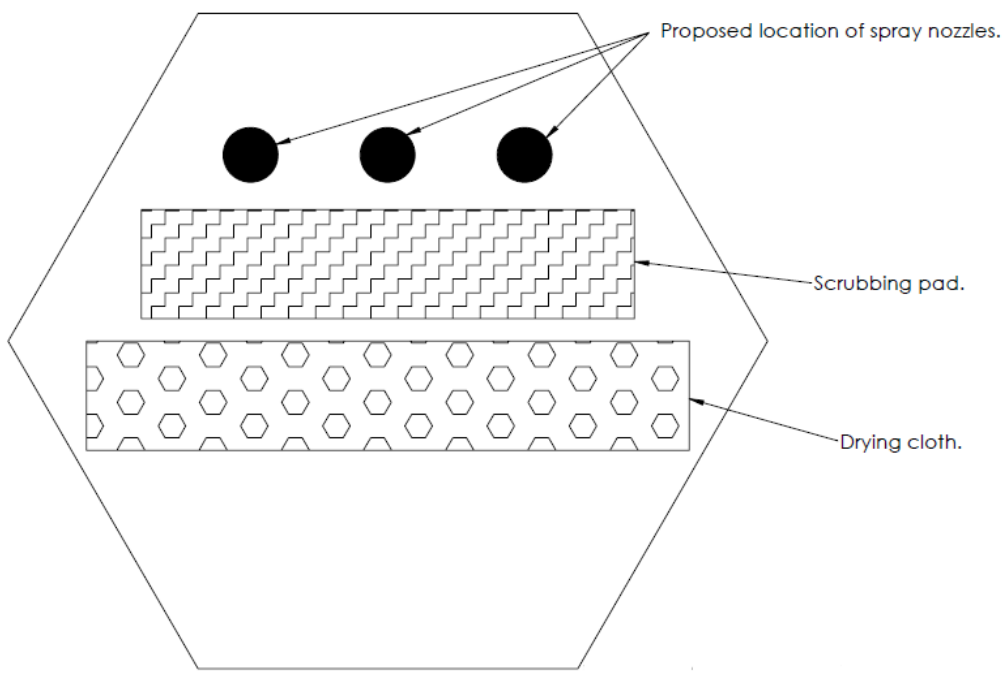


Figure 12 - Initial Nozzle Location

The secondary design made one major change, the water tank was moved from the electronics enclosure, to a dedicated tank located above the electronics. As such, the chassis design was extended upwards and the new walls would double as the sides of the tank. The pump, tubing, and spray nozzles all remained the same and would function in the same capacity. The new tank design incorporated larger dimensions to hold more liquid and a slot to insert a water level sensor to indicate when fluid levels reached a specified level. The second concept and design ended up being the final design for the whole subsystem.

Dimensions:

The fluid control subsystem has a variety of components each with unique dimensions. As an overview, the system contains a fluid tank, a water pump, vinyl tubing, brass fittings, and a water sensor.

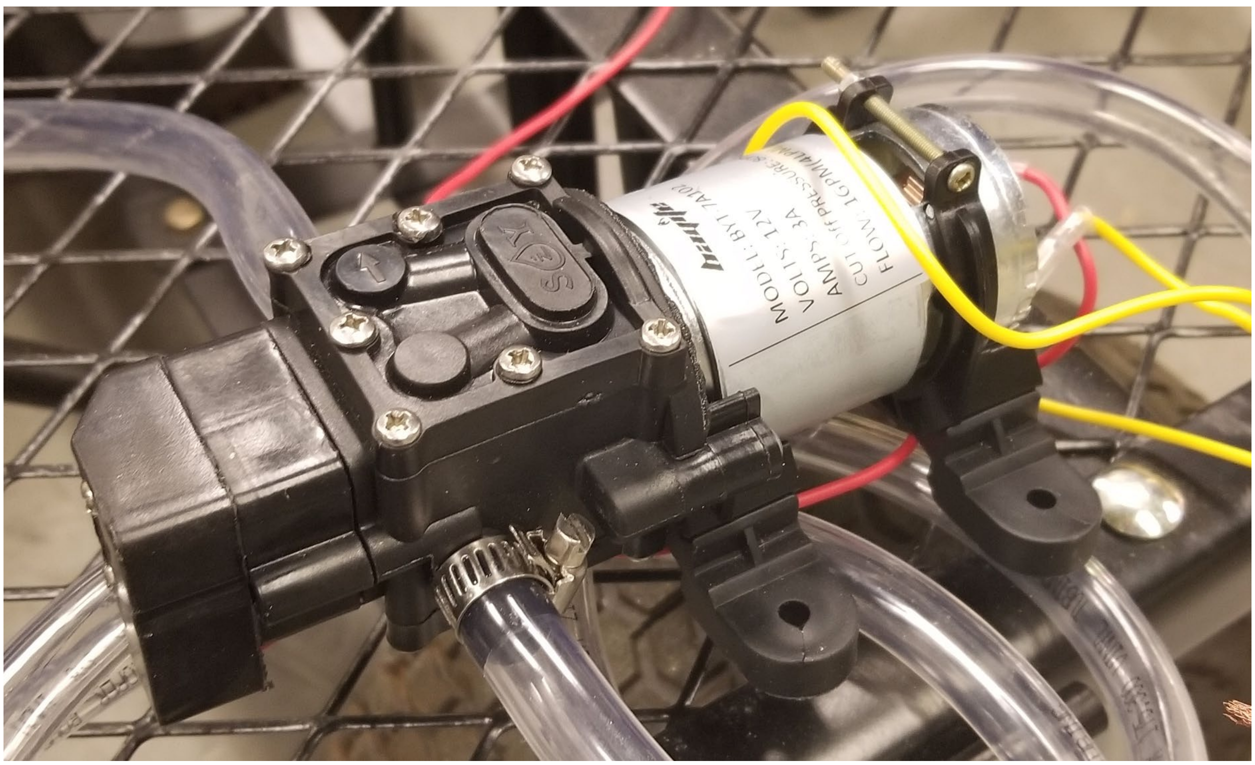


Figure 13 - Fluid Pump

The liquid tank is the largest component in the subsystem. The tank shares four walls with the chassis, each wall being 10.25” in length. Due to walls overlapping for seams, the total volume of liquid the tank can accomodate is a 10” x 10” x 1.5” box, for a total of 150 cubic inches, or 0.649 gallons of liquid. As seen in *Figure 14*, a hole is also installed in the upper plate of the fluid tank allows that the user can utilize to refill liquid whenever necessary.

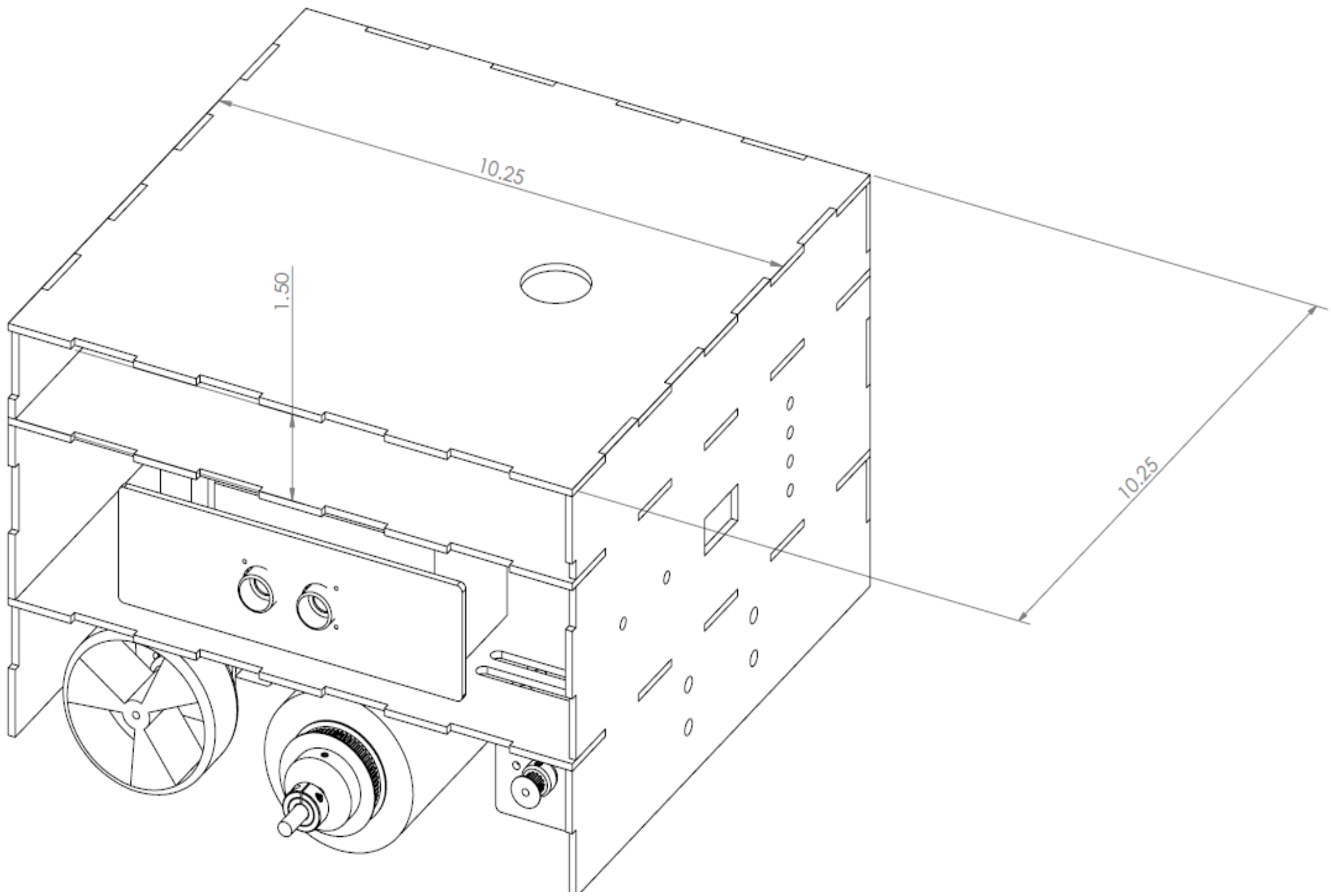


Figure 14 - Fluid Tank Dimensions

The water pump is manufactured to run at 12 volts and to draw a maximum of 3 amps. For more information on the electrical aspect of the pump, refer to the *Motor Control* subsystem. The pump is capable of producing flows of one gallon per minute. The pump has an inlet and an outlet port, both fitted for a ⅜” inner diameter tube. Lastly, the pump stands at 2.5” tall, 6” wide, and 4” long.

The tubing used to transport the liquid is a food-grade polyvinyl tube. The inner diameter of the tube is ⅜”, to mesh with the inlet and outlet ports on the pump for the subsystem. The walls of the tube are ⅛” thick, ensuring stability when under pressure. The brass fittings that mesh with the tubing have ⅜” NPT threads on one side and a ⅜” hose barb on the other to easily fit and clamp into the tubing.

The water sensor roughly an ⅛” thick, ¾” wide, and 2” long. The sensor is inserted vertically up through a hole in the lower plate of the liquid tank. The water sensor then detects changes in resistance to determine how much liquid is currently left in the tank.

Subsystem Testing:

A very strict requirement of the fluid control subsystem is the ability of the water tank to maintain a water-tight seal. If liquid is allowed through the tank, the potential to short electronics and cause fires increases significantly. As such, extensive testing was done on the system to verify that all components worked properly and efficiently.

For the tank, every component that enters the tank and every seam needs to be fully watertight. As such, extensive tests were performed to ensure the waterproofing. The tank was repeatedly filled and emptied with water, while visual inspections were performed to assess the level of fluid leaks. When a leak was noticed, additional sealant was applied to the area and the process was repeated until every seam and component in the tank was found to hold a full watertight seal over multiple fillings and emptyings of the tank. *Figure 15* shows several components that pass through the lower tank plate and the sealant around the pieces to ensure a watertight fit.

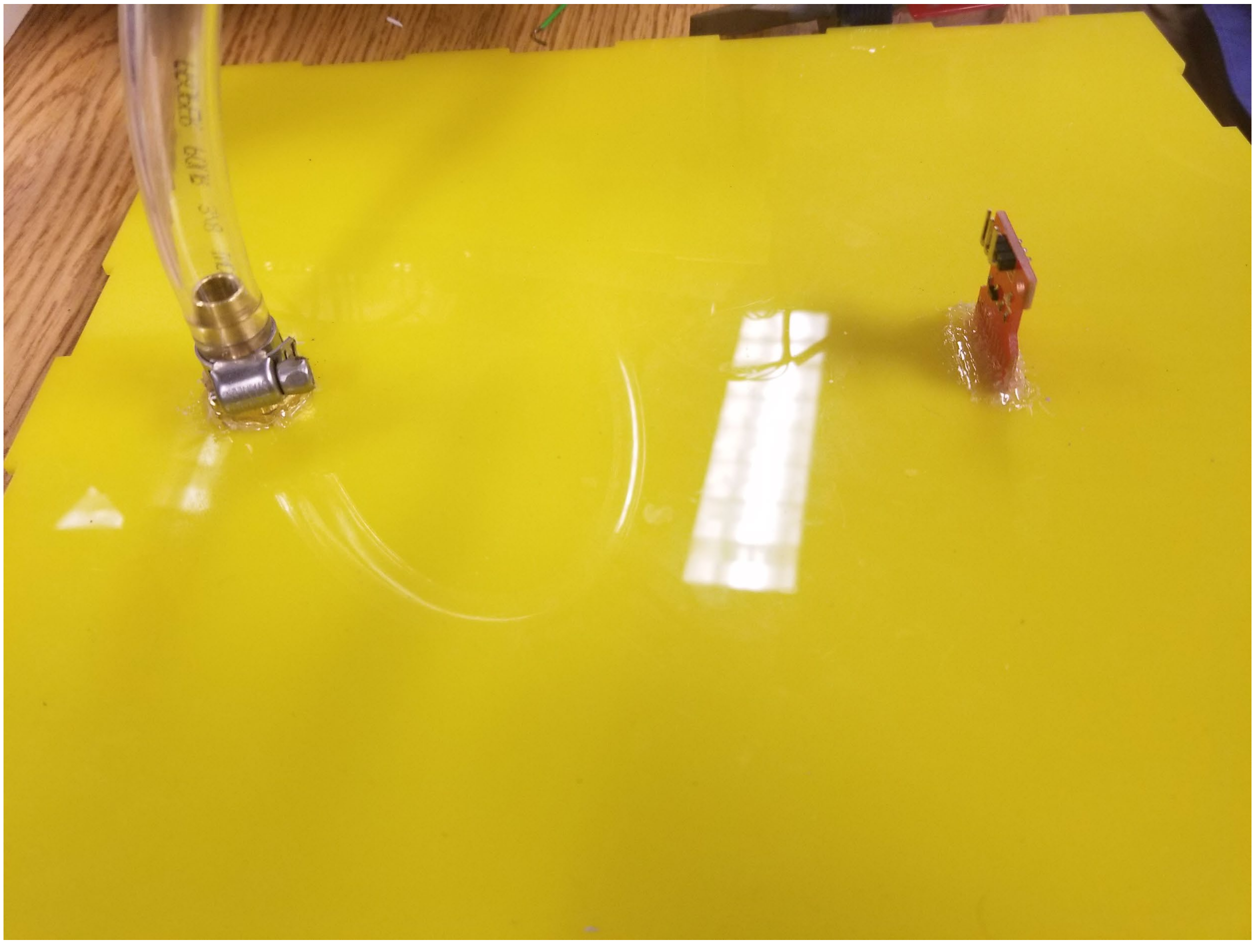


Figure 15 - Underside of Tank Plate

Tests for the pump needed to verify the current draw and the actual pressure and fluid rates provided by the pump. To test the pump, containers were filled with a known level of liquid. The pump was then supplied full power via a power supply and allowed to pull liquid from the container. By timing how long the pump needed to move all the fluid, the max fluid flow rate was found. Next, tests were done while the pump was under an extreme load (had difficulty moving fluid due to high back pressure around 500 kPa) to determine the maximum current draw the system could expect to see. Lastly, a variety of different nozzles and outlets were tested to determine which one would give the best fluid coverage. Each test for the nozzle involved attaching the new nozzle and running liquid through via the pump to determine the spray pattern and coverage area.

As per one of the previously stated customer requirements, the system needed to be efficient in running. An important aspect of the efficiency of the system is running for extended periods of time without needing to constantly add more fluid. As such, a specific mass flow rate of the system was desired. The appended derivations in *Appendix H* show the maximum mass flow rate and total run time the pump can achieve. After the analysis, the total pump run time before the system runs out of fluid was discovered to be 1.55 minutes. The metric was then tested by filling the fluid tank and timing the duration of time required to empty the tank. The testing found that the actual flow rate and the predicted flow rate were nearly identical, thus verifying the accuracy of the analysis.

Integration:

The integration of the overall subsystem is quite simple. A tank suspended at the top of the chassis is filled with liquid. The tank contains an exit nozzle and a water level sensor. A fluid is pumped from the tank, through a pump, and down to the base of the chassis. At the base of the chassis, the fluid experiences a velocity and pressure change before being sprayed out through pores in the tubing onto the cleaning system. In addition, a pressure relief hole was installed on top of the tank to expose the tank to atmospheric pressure, preventing the buildup of vacuum in the tank. As a result, the tank is less likely to collapse from the negative pressure and the pump performs less work fighting a vacuum and achieves the same results.

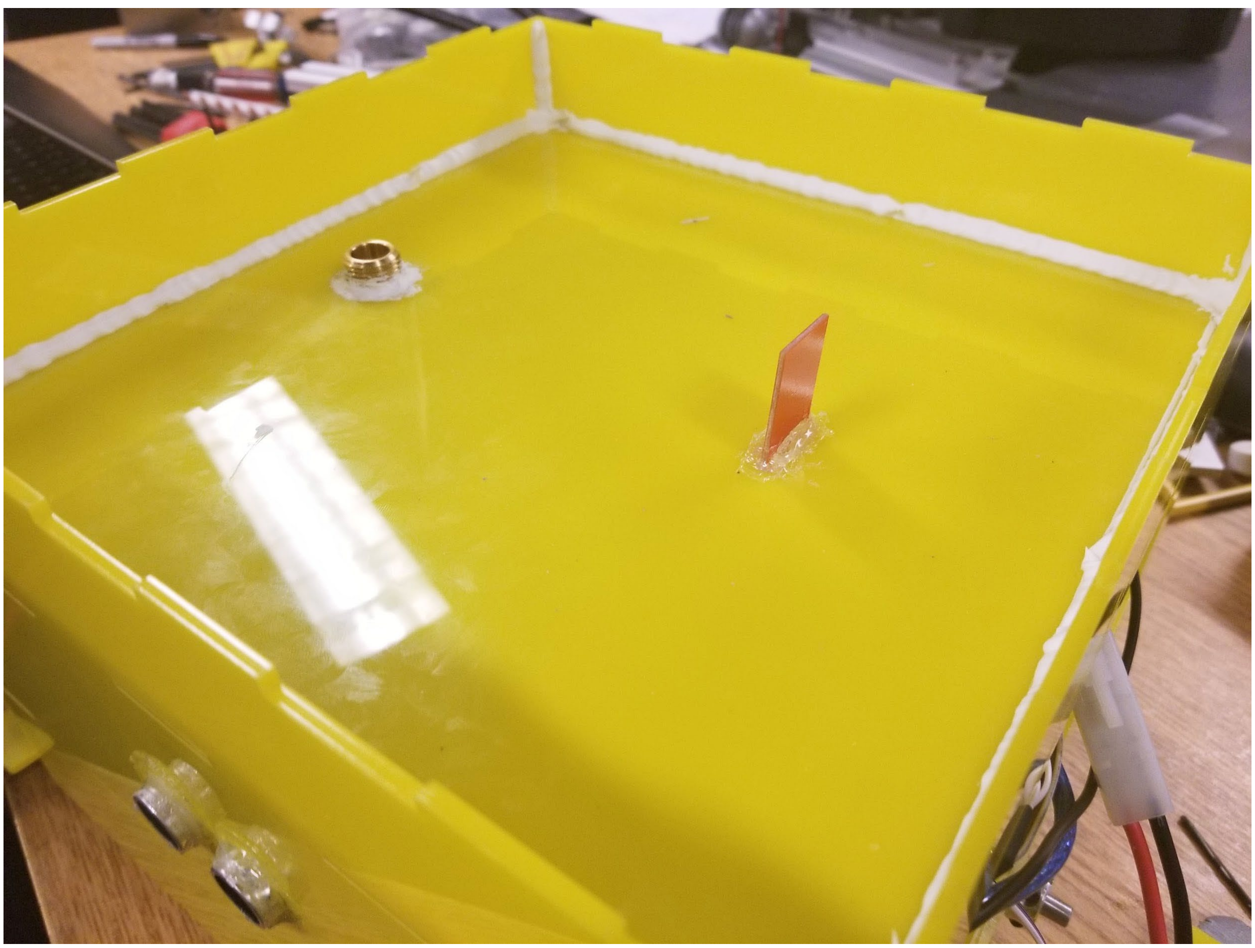


Figure 16 - Internal Waterproofing of Liquid Tank

By using the dynamics analysis (*Appendix H*) as a basis (verified by the team's test results), an on/off cycle was decided upon for the pump. The pump has a total run time of 1.55 minutes before it runs out of fluid. However, the robot does not need to run the pump continuously until fluid is depleted. As such, the pump was determined to be most efficient by having periods of fluid pumping followed by periods of no fluid dispersion. The cycle of three seconds of pumping followed by five seconds of no pumping allows for the amount of fluid dispersed to be accurately predicted. With this new pump cycle, the true run time of 1.55 minutes can be greatly increased to any desirable amount at the sacrifice of total fluid dispersed per unit time.

Challenges:

Several challenges were present in creating the fluid control subsystem and ensuring proper operation. Some of the challenges included: waterproofing the system, controlling speed of fluid dispersion, and maintaining a safe pressure within the tank.

The location of the fluid tank is directly above the electronics enclosure for the whole system. As such, great care needed to be taken to ensure that the tank would not leak water onto the electronics, causing a short. To prevent leaks, the entire tank was acrylic welded on all sides and every seam. In addition, plumbing caulk (*Figure 16*) was also added to every seam to create better seal. Various tests were performed on the seams of the tank to verify that the tank was fully watertight before final assembly was complete.

To control the speed of fluid dispersion, various changes were made to the exit nozzles. As such, the size and shape of the nozzles allowed the team to specifically control how much fluid passed through the nozzles. In addition, a system was developed to pulse the pump for periods of pumping followed by periods of no pumping. The pulse to the pump allowed for activation and deactivation on command, which greatly increased the speed at which fluid was dispersed from the system.

In the fluid tank, a vacuum forms as fluid is being pulled from the reservoir. If the vacuum is allowed to build, there is potential for the low pressure to cause the tank to collapse in on itself. As such, a small hole was added to the top of the fluid tank. The purpose of the whole is to expose the contents of the fluid tank to atmospheric pressure. The new pressure in the tank offsets the fluid being pulled out and prevents the tank from collapsing. In addition, the increased air pressure on the liquid increases the efficiency of the pump since there is no longer a vacuum to fight against.

Cleaning System

Objective

The cleaning system is primarily responsible for cleaning up various kinds of dust and dirt on the floor. In this subsystem, a sponge is the primary tool to accomplish the cleaning. While choosing the materials to clean the floor, the Autonomous LCD team found and tested various kinds of materials, such as: steel wool, microfiber dusters, and various sponges. After careful consideration, sponges were determined to be an excellent fit for various customer requirements provided to the team. Among the requirements, which are discussed in detail later in the subsystem, are a low cost, easy maintenance, and a simple design.

While the Autonomous LCD is cleaning the room, the sponge roll, which is located at the bottom of the robot, will keep rotating at a high speed. The high speed allows for a more thorough clean within a smaller time frame. A timing pulley is attached to the same dowel as the sponge and then connected by a belt to the motor. The motor allows precise control over the speed of rotation of the sponge and also provides an increase in torque through a change of gear ratios. The increased torque allows the entire sponge and dowel system to be operated off a single motor. By utilizing only one motor, the system will save power, increasing battery life and ultimately saving more time and money.

Customer requirements

Customers' core requirements are affordability, safety, and efficiency. The sponge should be set to the right spin speed so dust, dirt and grime will be thoroughly cleaned, but at the same time, the sponge can absorb an excess amount of water on the floor.

Once the sponge gets dirty, customers are able to easily wipe off the dust on the sponge. If the sponge is damaged or becomes too dirty, customers are able to take the cleaning system apart and replace it with a new, clean sponge very easily with the use of some hex-keys.

In addition to the easy replacement, the cleaning sponges are extremely inexpensive, which allows each and every robot owner to afford the afterward costs.

Design Process/Concept Selection

At the beginning of the designing phase, the team found different types of materials that excel at collecting liquids, such as: sponges, cotton cloth, and paper towels. Paper towels were quickly dismissed, since once the paper towel becomes saturated, the towel would fall apart and interfere with the robot's movement and create a larger mess. The sponge is good at sucking up the fluid, but once it is saturated, a customer may have to take it out manually and let the sponge dry, which could be a pain for the users. The cheesecloth was the best material moving forward, so the Autonomous LCD team designed a system that will allow the cheesecloth fully contact the floor and collect dirty liquid that is left. Once the cheesecloth is filled with liquid, the cloth should be collected and stored in a place that will allow new cheesecloth to replace the dirty cloth. The team decided to use two dowels for the cheesecloth mechanism; one for sending in the new cheesecloth and the other for collecting the dirty cheesecloth (Cleaning System Concept Sketch 1). There will be a motor attached to one of the dowels which allows the cheesecloth to spin. Additionally, strings will be attached to the dowels in order to absorb stress from the floor. In that way, the drying job can be done all at once. With the overall design of the chassis, space quickly became the greatest challenge, because the inside of the robot where the system can be place is so small that it is almost impossible to fit the whole cheesecloth mechanism in. Also when the robot carries the cheesecloth mechanism and moves around, the motors will do a lot of mechanical work, which will end up consuming more power than needed.

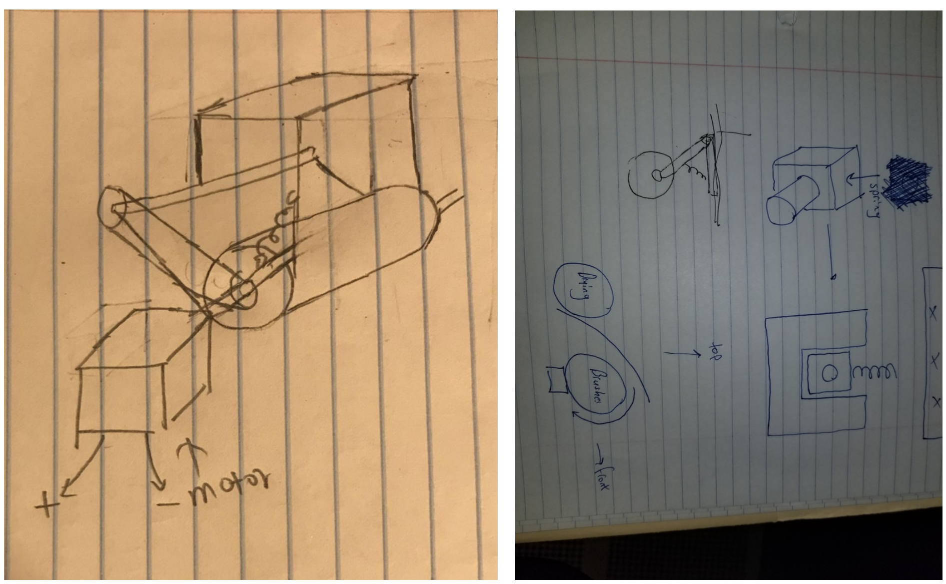
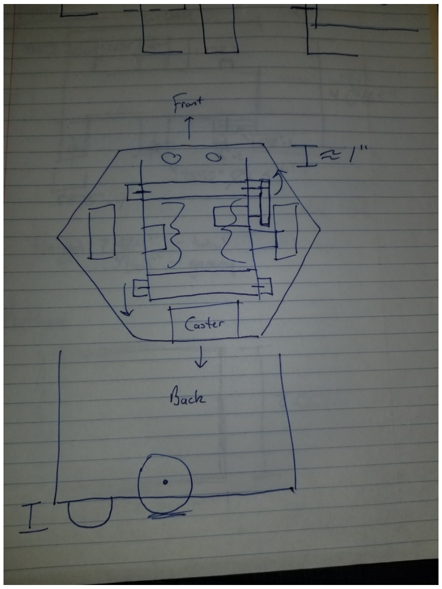


Figure 17 - Scrubbing System Concept Sketch

The team changed the design to replace the cheesecloth mechanism with a rotating sponge (*Figure 17*); the sponge and the brush will be connected to the same motor by two separate belts. This will allow the sponges to move freely, while making the drying and cleaning process more efficient. When calculating the dimensions of this mechanism, results showed that the system would require much more space than is available. Because both the brush and the sponge will be powered by each of their own belts, at least 6 inches of length was required to install the whole system. But total length of the robot is 6 inch so the team decided to abandon this idea and then decided to just power the brush and simply let the sponge roll on the floor on its own (without motor power).



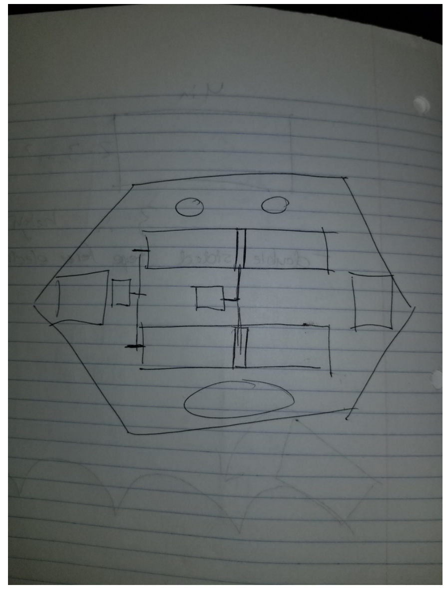


Figure 18 Figure 19

Cleaning System Concept Sketch

The design seemed to work well, but when the team assembled the chassis, it was almost impossible to connect the sides together in order to form a hexagon. So, the team redesigned the shape of the chassis and the mechanism of the cleaning system. In the end, the team decided to the change the shape of the chassis from a hexagon to a cube; the drying sponge was also abandoned and changed the cleaning system to just using a brush, which was then powered off of a motor. At the same time, there was no worry about the excess amount of cleaning fluid left on the floor (due to pump) since the team had adjusted the code and set the water spraying speed to a lower value which was just enough for brushing the floors without soaking floors. The brush was built by using multiple pieces of scotch brite pads which were rolled together. A metal rod goes through the center of the brush roll and at the end of this rod, a timing pulley system was attached and connected it to a motor which rotates the brush.

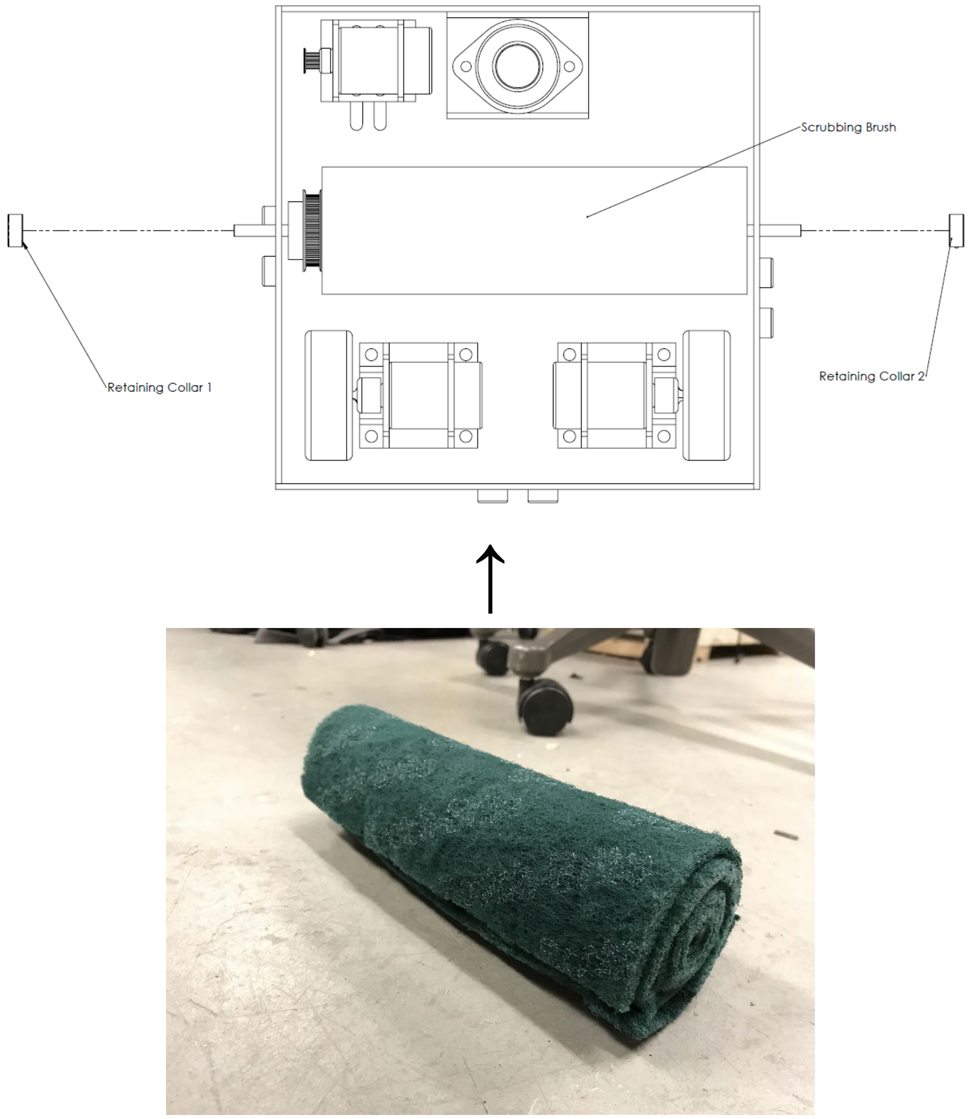


Figure 20 - Cleaning System Final Sketch and The Brush

After the design was finalized, the pads were built and tested, but the team realized that the brush is influencing the movement of the robot due to friction from the floor. The team then tested the cleaning ability of a sponge and saw it is actually able to suck up fluid and brush the surface at the same time. Due to the lightweight sponge and minimal amount of the friction it's generating, this design was chosen.

Dimensions

The sponge is 6.5 inch long and its radius is 1.25 inch.

The belt used has 130 teeth and its pitch is 0.4inch.
The timing pulley we used on the motor is 0.08in MXL pitch, 15 tooth, 0.4in pitch diameter, 3/16in bore, oversize hub with flange.

Subsystem Testing

04/16/2018
The team first tested the brushing ability of the scotch brite pads by securing the pads to a metal rod and rotating the rob via the gearing system. The pads could clean dirty messes on the floor extremely well at the speeds tested.

04/22/2018
The whole robot has been built, and the cleaning system has been successfully installed. But when first testing the movement of the robot, the robot was not moving forward because the brush was spinning in the opposite direction of the wheels.

04/23/2018
The team fixed the brush spin direction by switching the polarity of the motor leads in the circuit. After this fix, the robot slightly moves but the brushing is not as smooth and even as the team would like.

04/25/2018
The team replaced the scotch brite pads with the sponge, and the robot started moving forward without too much resistance due to friction. The sponge is also cleaning and collecting dust on the floor. At this point, the sponge replacement is up to par.

Integration

The integration of the cleaning system was fairly straight forward. The sponge is used to brush and dry the floor. The belt is used to connect to timing pulleys together and transfer power from the motors to the metal rods which allow the sponges to spin. The timing pulleys, which are attached to the rod and the motor are held together by a belt, which transfers power to the rod (and increases torque).

Challenges

The main challenge faced when building the cleaning system was having to change the design over and over again due to budget limitations, space limitations and time limitations. Before designing the robot, the team only plans to spend \$100 to \$150 on this project, this means that the team do not have tons of money to spend just on the acrylic itself, and that causes some space issues. Fitting everything into a small box is no easy job, and in order to make things perfect it takes the team to redesign the robot over and over again. At last, the time that is given is only 3 months, and due to the heavy amount of designing and building that the team have, there is not much time being left for refining the robot.

External Communication

Objective

The objective of the external communication subsystem is to simplify the way the Autonomous LCD interacts with the user. By having an external communication subsystem, the user will understand when problems with the robot occur, or just as simple as letting the user know that the Autonomous LCD is on. This subsystem will make the robot extremely easy to understand and will make the experience of autonomous cleaning easy and fun.

Customer requirements

Safety should be the number one goal of every product, and that stays true with Autonomous LCD. External communication will assist with safety by alerting the customer when it is on, so the customer knows not to pick it up or refill it which may be dangerous.

After taking a survey through google forms of fellow RPI students, family members, and friends, the Autonomous LCD team found that many people would purchase an autonomous cleaning robot. Also, the survey discovered that nearly all answers wanted to have a system in the robot that could communicate with the user. Due to that, the Autonomous LCD team took full advantage of the opportunity, and the team felt that implementing an external communications system was completely necessary. The survey said that customers wanted to have some sort of communication regarding battery life and liquid tank capacity.

Overall, the goal of this subsystem was safety and ease of use for all customers. The Autonomous LCD team feels the external communication system satisfied this need from the customers.

Design Process/Concept Selection:

The process of concept selection was integral to the external communication subsystem. This was because the team had a lot of ideas for this subsystem with a common goal of safety and easily understandable communication. Due to all these needs, but space constraints, the team had to make a couple of changes to this subsystem so that it coexists well with the other subsystems.

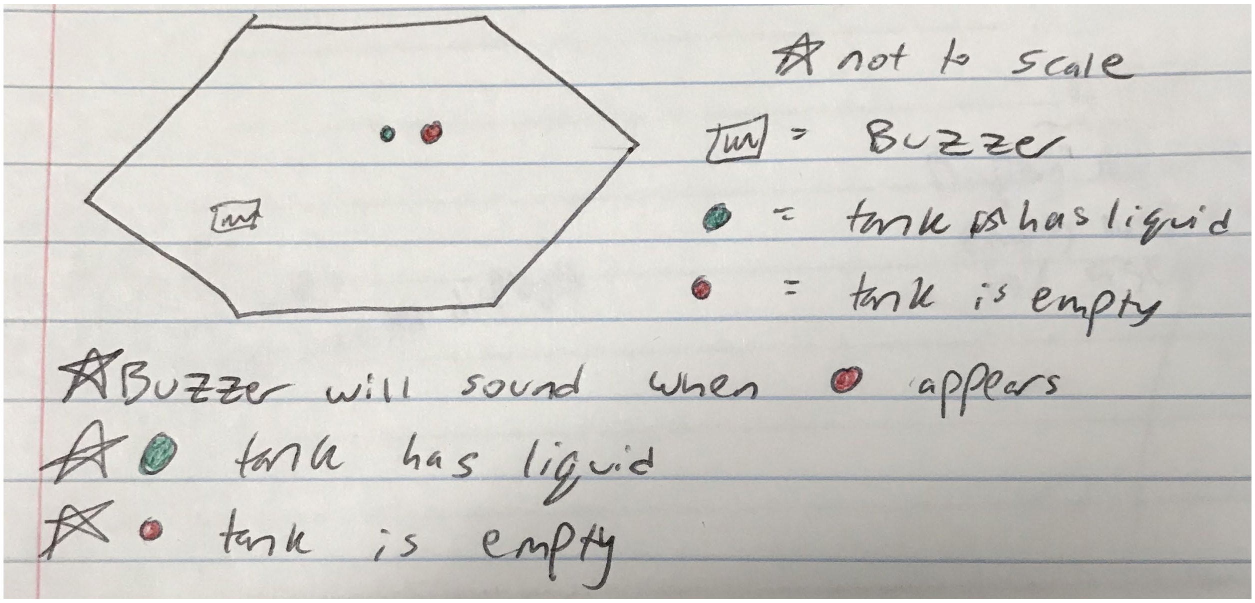


Figure 21 - Initial External Communications Sketch

The first sketch of the external communications subsystem includes a green LED when the liquid tank is a full and a red LED for when the liquid tank is empty. A buzzer will sound when the red LED is displayed to notify the user about the empty liquid tank. This preliminary sketch failed to include battery life LED's, a on/off switch, and an LED to tell the user that the machine is on.

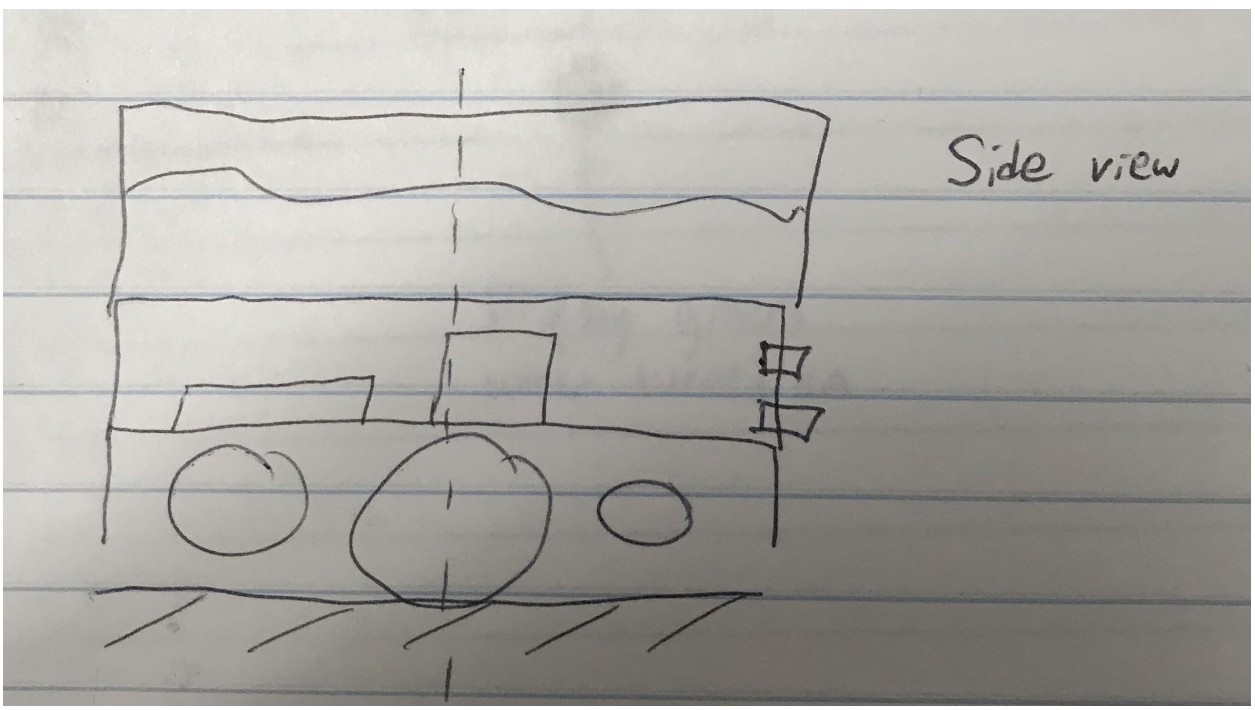


Figure 22 - Side view concept sketch

The second sketch shows the side view of the original sketch. The side view shows that the LED's will be extruding on the side panel of the robot.

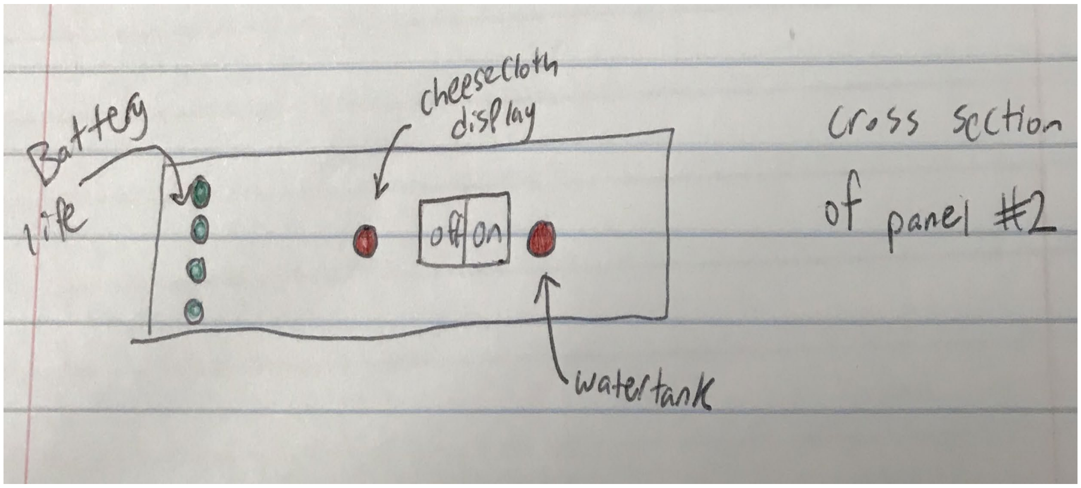


Figure 23 - Side panel concept sketch

The third sketch shows the external communications subsystem concept sketch number two. This sketch now includes, four vertical LEDs that represent battery life, an on/off switch, a LED for the cheesecloth spool, and an LED for the water tank. The four vertical battery life LEDs will display all green when the battery is >75%. With each 25% decrease in battery life, each corresponding LED will turn off. The cheesecloth spool display will indicate a red LED when the spool has run out. The on/off switch will be completely user friendly and will be used to turn the Autonomous LCD on and off. The water tank LED will indicate red when the liquid has run out. When any red LED occurs, a buzzer will sound which will let the user know that a problem has occured.

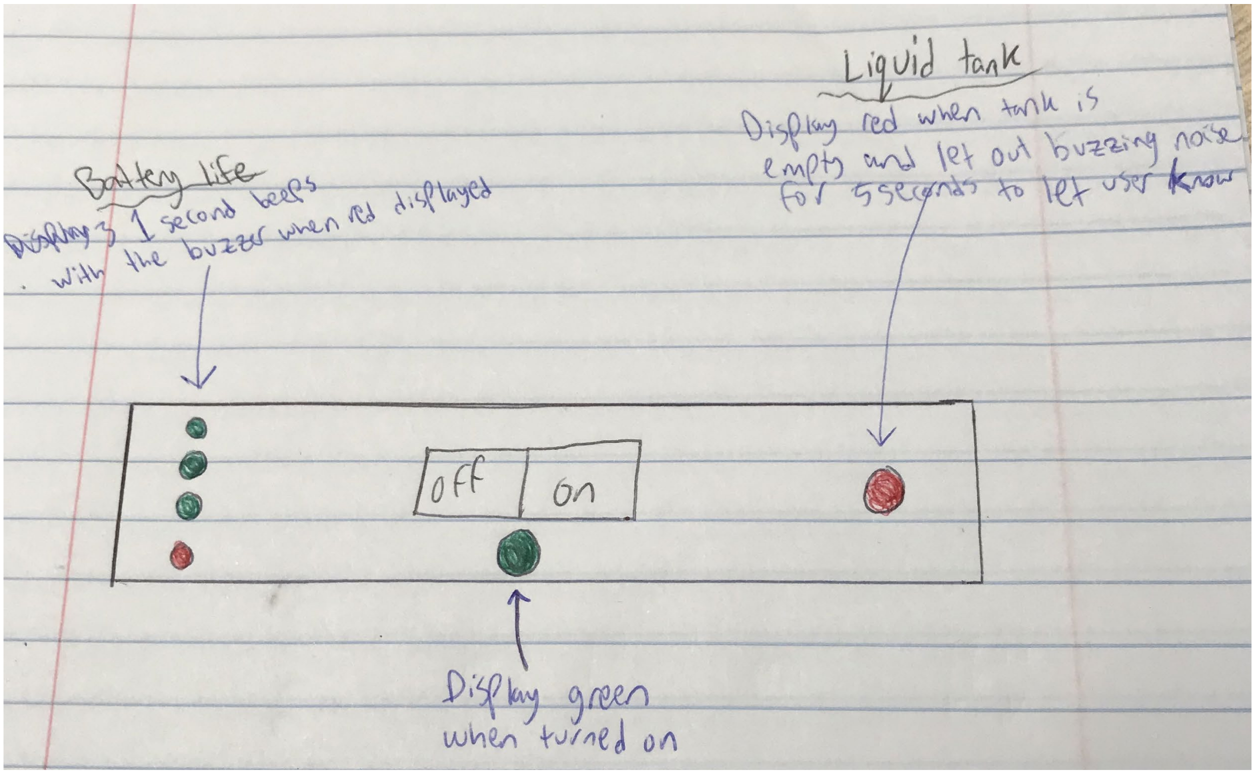


Figure 24 - Revised concept side panel sketch

The sketch above shows the final rough sketch of the external communication subsystem. This sketch is different from sketch two due to the elimination of the cheesecloth display LED, a red battery life LED, and an LED that correlates with the on/off switch. A LED will display green when the Autonomous LCD is turned on by the user using the on/off switch. The battery life LEDs will be placed vertically on the side panel and will display green LEDs when the battery is >25%. When the battery goes below 25%, then a singular red LED will display and alert a buzzer in order to let the user know the robot has a low battery life remaining. A red LED will be displayed when the liquid tank is empty. This signal will also alert a buzzer and stop the robot from cleaning which will tell the user that the machine needs more liquid.

The battery voltage was obtained by using the analog to digital converter on the arduino. This electronic device allowed analyzing battery voltage by using digital data. The digital data was then used to turn on/off LEDs when the battery voltage changed.

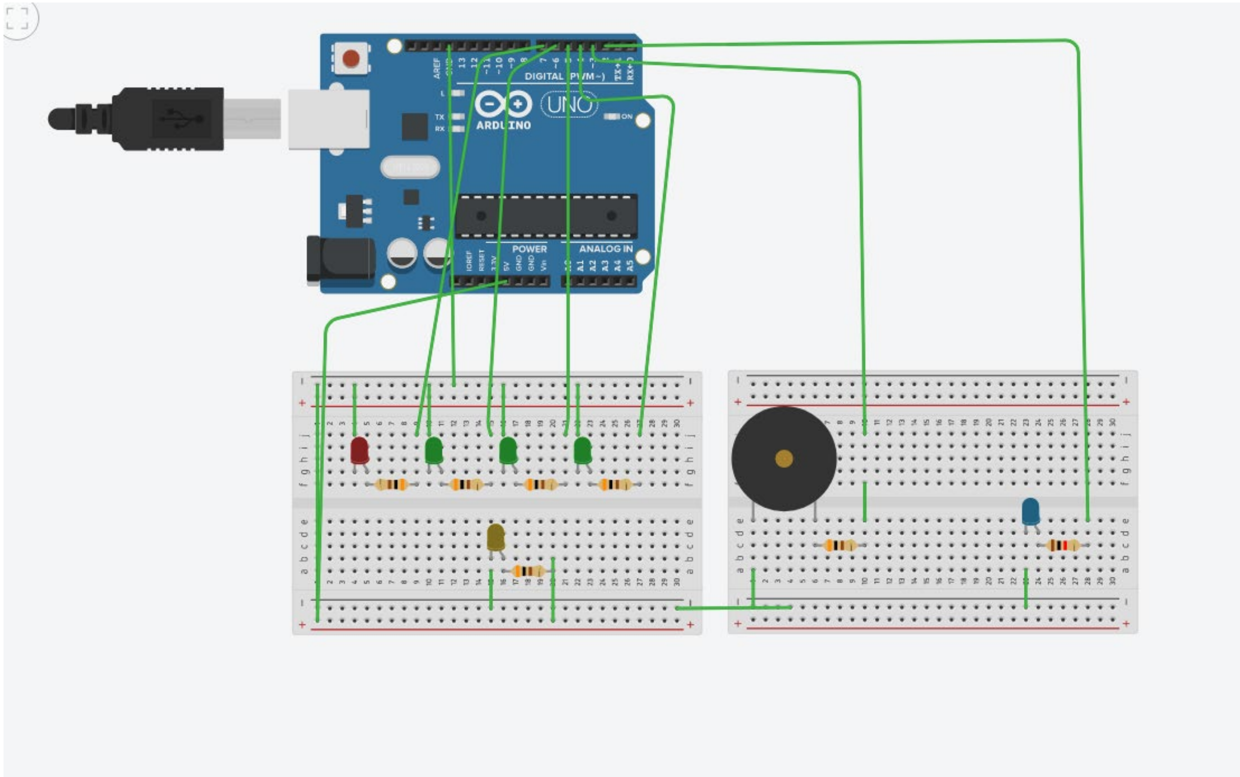


Figure 25 - Circuit board wiring

The figure above shows the wiring to make the external communications subsystem run smoothly. A 100Ω resistor is wired in series with the LED to prevent too much current from going into the LED. This prevents the LEDs from being burned out. Wires from the arduino's digital output pins will be connected to each LED. A digital pin was also wired to a buzzer, which indicated standby mode. More information on standby mode can be found in the user manual.

Figure 26 - Side panel LEDs and switch

The picture above shows the final design of the external communications Subsystem. All of the LEDs fit perfectly into the design and lit up as expected. The LEDs were hot glued into the acrylic sheet and wired accordingly.

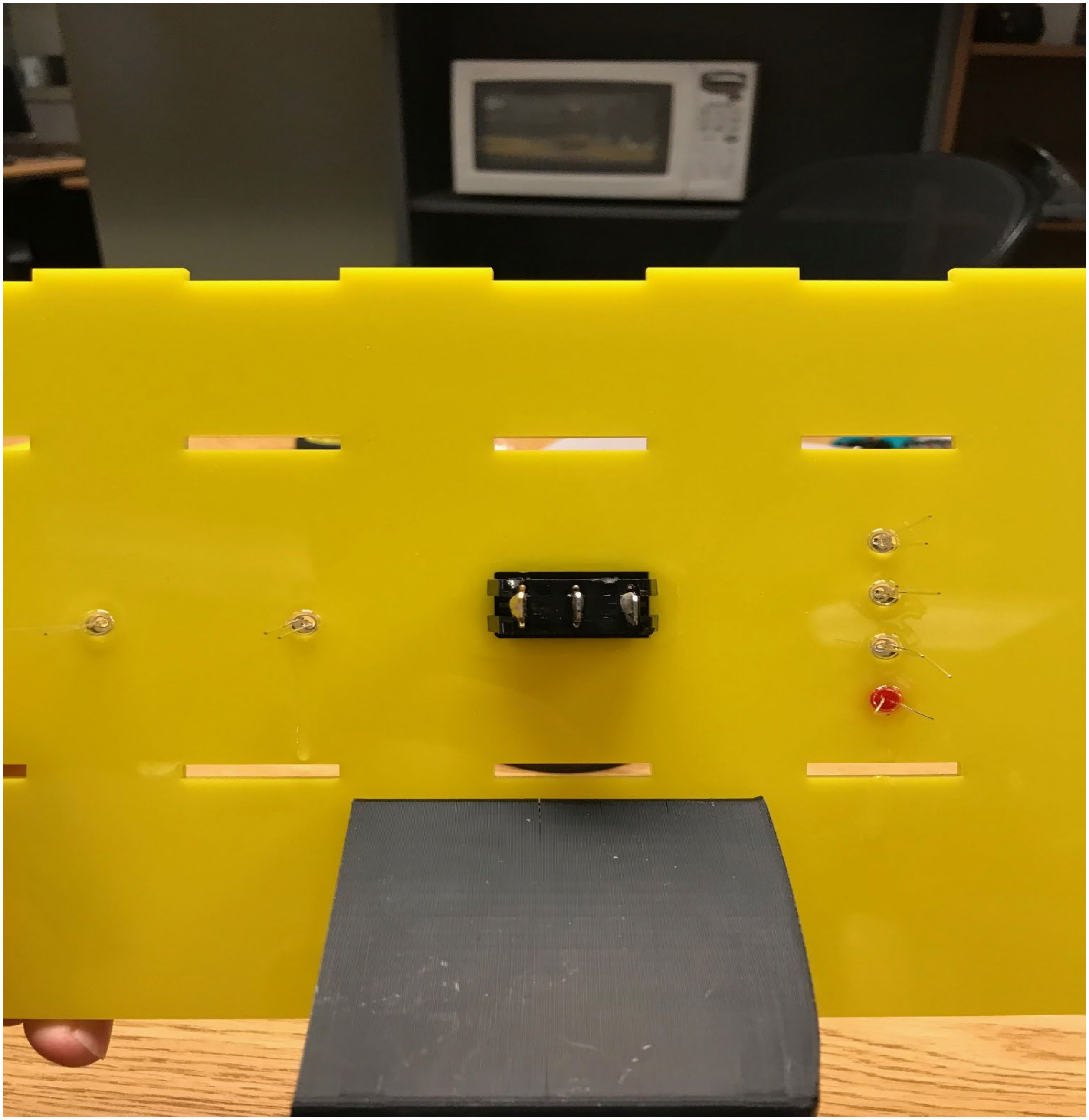


Figure 27 - Side panel LEDs and Switch back view

Shown in the picture above is the final design of the external communications subsystem from the back view. The wiring was done from the back and attached to a circuit board.

Figure 28 - Fully integrated subsystem

Above is the final external communications system with all of the LEDs turned on fully. The water tank LED is on due to the fact that there is no water. The four battery life LEDs are all on in order to show the battery at full charge.

Dimensions

The dimensions of each of the LEDs are .5cm in diameter for each of them. The on/off switch is 1.5 inches long and 1 inch wide.

Subsystem Testing

04/19/2018

All of the LEDs were tested to see if they lit up when activated by a wire that was connected to an arduino. The on/off switch has yet to be tested due to the full robot not being completed. The water tank LED was activated when the team went from the water sensor touching water to not touching anything. The team has yet to integrate the external communications subsystem with the other subsystems due to the Autonomous LCD not being complete at this moment in time.

04/27/2018

The LEDs and the on/off switch all work when activated. When the switch is turned on, the battery life LEDs indicate 3 green and 1 red LED. Also, the liquid tank LED is off when water is present, but indicates a blue LED when water is no longer present. When the switch is turned on, the green power LED lights up to notify the user that the Autonomous LCD is now on. Every aspect of the external communications subsystem works to the fullest of its capabilities.

Integration

The integration of the external communication subsystem was extremely simple. The LEDs and the switch had resistors soldered onto them. Wires were then connected from the arduino to the LEDs. A code was developed to activate the LEDs when needed. The external communications subsystem was necessary to be integrated with each subsystem because it acts as the voice of the robot. The external communications subsystem is the bridge from Autonomous LCD to the user.

Challenges

A major challenge that the team encountered was connecting the water sensor to a blue LED to indicate when the liquid tank was empty. The water sensor did not display the blue liquid tank LED at the right times. Due to this, the team had to rewire the LED to the water sensor in a different manor. After this, the LED worked correctly when the water sensor stopped sensing water.

Motor Control

Objective

The main objective of the motor control subsystem was to provide electrical power to electric, brushed DC motors. The motors were used to provide mechanical power to wheels, scrubs, and pumps. These motors needed to be controlled via software (from an Arduino), in order to integrate the room navigation subsystem. The motor direction and speed was required for wheel motors. Speed was required for the scrubber and pump. Safety was also a main objective, because liquids were present. Circuits were tested and developed that maintained safety standards, while also providing enough power that the cleaning/spraying systems required.

Customer Requirements

Safety was a large concern when designing the subsystem. The Autonomous LCD should not seriously injure any user, either from motor entanglement or from electrical shock. Since liquid is involved and is a central part in the overall design, steps were taken to prevent shorting of the electrical equipment via water spillage. All tubing was firmly secured and sealed, while all water was contained away from electrical equipment. Electrical tape was used excessively to prevent shorting. All motors were also soldered and then insulation was added to the solder joints so that water could not get into the motors. Heat sinks were added onto transistors on circuit boards, in order to dissipate heat and vent. This prevented potential burning of users. 22 gauge wiring was used for circuit board connections, while 16 gauge wire was used for all motors. This wiring system was used so that users would not burn themselves if they ever touched a motor wire, as the maximum current of our motors would never exceed two amperes. Two amperes through a 16 gauge wire will not produce much heat. The voltage of the battery used was 12V (~14V when fully charged), and the current capacity 2000 mAh, which met our expected standards for safety, with regards to voltage and current maximums.

Design Process/Concept Selection

Concept selection of the motor control subsystem was integral to the design process. The design process was split into two parts: speed/direction control, and speed control. The Autonomous LCD makes use of four total motors: two motors that drive the wheels, a motor that drives the scrubber, and a motor that controls a pump that transfers water from the tank to the scrubber. The motors that drive the wheels require both speed and direction control, while the pump and scrubber motors require only speed control. Each wheel motor also needed to have its own speed/direction control, so that left and right turns could be performed. In order to do this, a circuit called an H-Bridge was developed for speed/direction control. The first designed H-Bridge is shown below.

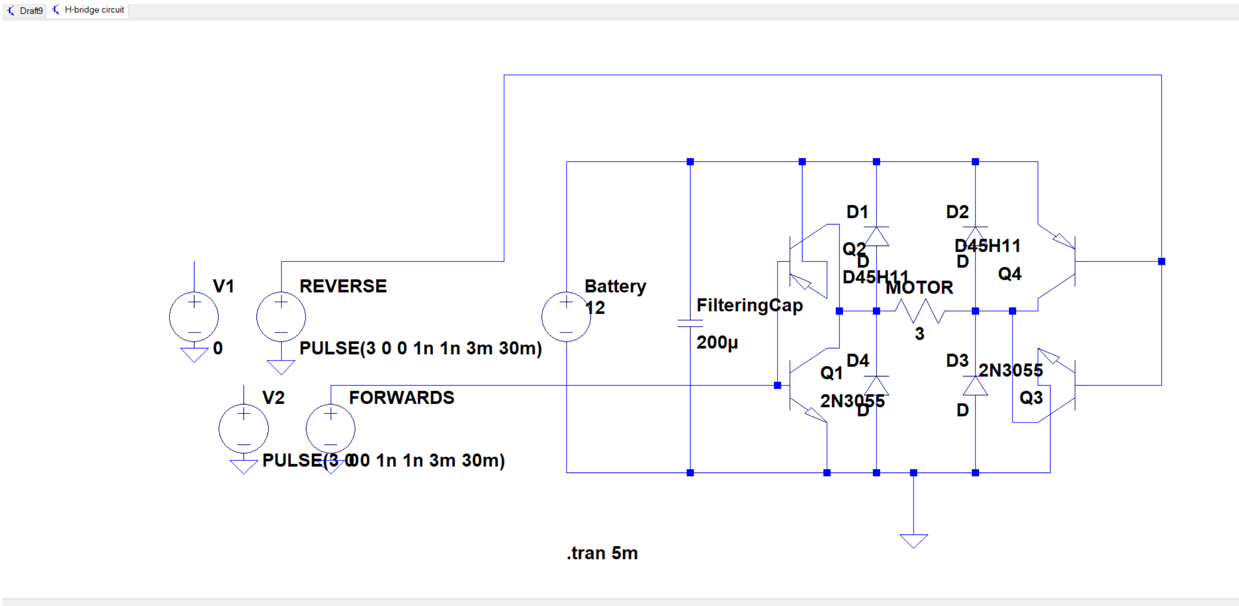


Figure 29 - Motor Control H-bridge Design#1

Figure 29 shows the first H-bridge design, which consists of four total transistors. This first design made use of BJT (bipolar junction transistors). Two are NPN, and two are PNP (top two PNP, bottom two NPN). Each NPN transistor collector is connected to ground, and each PNP transistor collector is connected to 12V. The base pins of each transistor are connected to a PWM analog output from the arduino. Finally, the emitter pins the PNP transistors are connected to an NPN collector pin. Diodes are added from each of the transistors emitter pins to their collector pins for protection against reverse direction currents. The PWM input from the arduino that is connected controls how each transistor operates. In this circuit's case, transistors act as switches. NPN transistors will turn on (emitter will equal collector voltage) when the base voltage is a low value (close to zero), and PNP transistors will turn on when base voltage is a low value. To control the direction, the upper left and lower right transistors must be on, and the others off, so current will flow to the right in the motor. Likewise, if the upper left and lower right transistors are on and the others off, current will flow to the left through the motor. The voltage of the base determines how much voltage a transistor will draw from the battery. A higher base voltage equals higher speed. This operation allowed for easy use of an arduino, with only two output pins used, which made coding easier. This circuit worked as a speed/direction controller, but it was deemed not good enough, so another circuit was designed that worked much better.

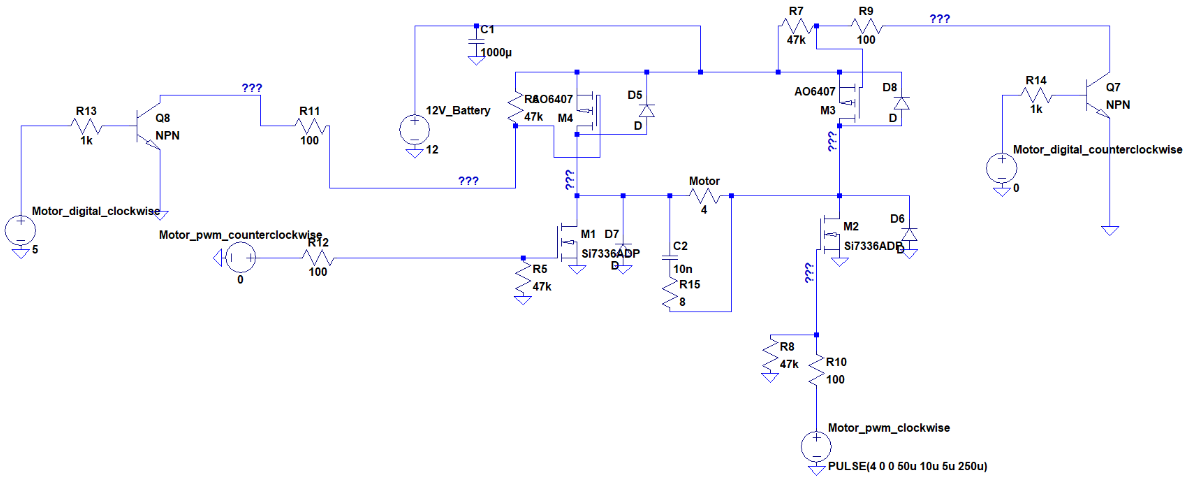


Figure 30 - Motor Control

The circuit above shown in *Figure 30* makes use of BJTs, but the main transistors used in the h-bridge were MOSFETs. MOSFETs are generally very good at switching high voltages and high currents, and are very good transistors to use in an h-bridge. As with BJTs, there are two types of MOSFETs: NMOS and PMOS, which function in similar ways to NPN and PNP BJTs. BJTs in this circuit were used to prevent a large amount of current to flow out of the arduino. This is very important, because if the arduino supplies too much current, it can burn out electronics on itself, such as communication circuits (TX, RX pins). The BJTs prevented this. The overall operation of the circuit is the same as the before one, except that there were four overall inputs. This may complicate coding for driving the motors, as well as crowd the pins on the arduino itself, but this was needed to obtain higher motor speeds than with the earlier designed circuit. The two BJTs are connected to a digital output pin on each arduino. These pins and their voltages directly control the motor spin direction. A zero turns the BJTs on, since they are both NPN types, so for the circuit to work, one BJT must be 0V, and the other must be 5V. The output of the BJTs is connected to a PMOS transistor. PWM signals are connected to each NMOS transistor, which directly control the speed of the motor. The MOSFETs used in the circuit are capable of switching 30V and 15A, which is well above values used in the Autonomous LCD.

Testing via simulation was done, to confirm the direction of current flow through the motor. The motors DC steady state impedance was tested using an ohm-meter, and was found to be $2-4\Omega$, so a resistor of 4Ω was used, because if the impedance of the motor was that high, a test would show the worst case where internal impedance would reduce overall current going through the motor via Ohm’s law. The motors used in the actual design were tested to be 1.9Ω, but testing the worst case possible was important because it showed how much current the circuit could source under those circumstances. Input “speed” voltages to the NMOS transistors were pulsewaves, just like a PWM output from the arduino. This allowed for better analysis of the circuit response.

Subsystem Testing

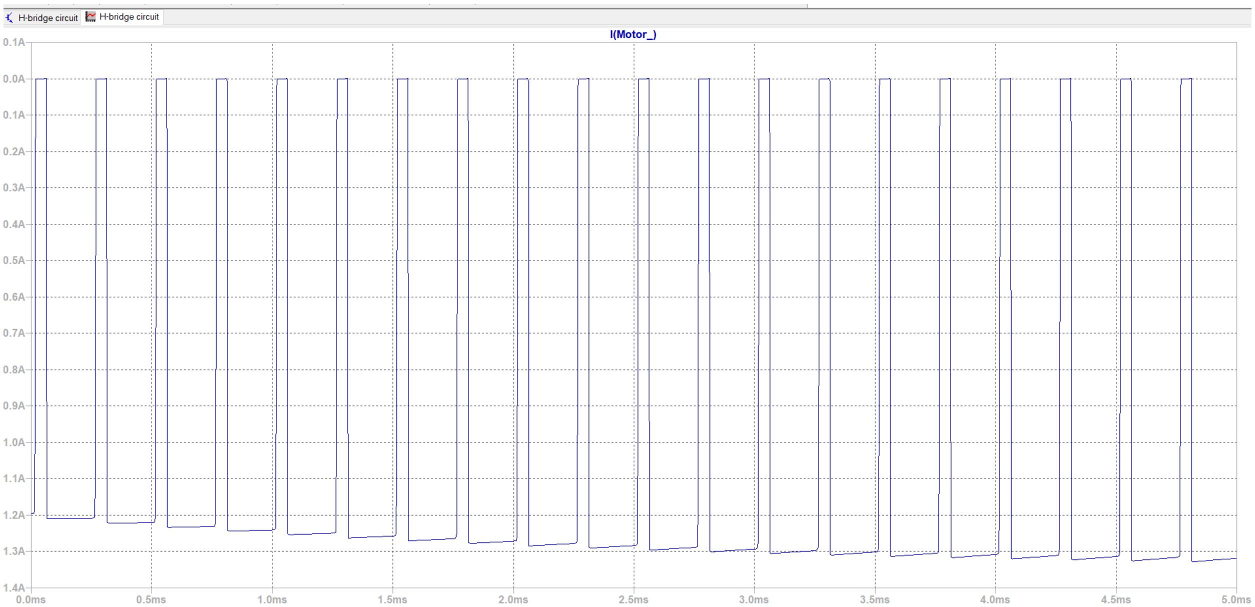


Figure 31 - Motor Counterclockwise Direction Control Testing

As seen *Figure 31*, the current through the motor is a pulse wave ranging from -1.3A to 0.0A. The simulation software did not have a feature to add the current supply of the voltage source, so an 6Ω series resistance was added with the voltage source, effectively making it a 2.0A current source with a simple source transformation. The battery used in the model was a 12V, 2000mAh NiMh battery, so this added series resistance in the simulation provided the exact current the motors would theoretically draw in a real setting. The speed setting in this simulation was at 4V, which is fairly high (max is 5V from the arduino). If the speed was turned down to 3V, the current going through the motor was reduced, as seen in the below figure:

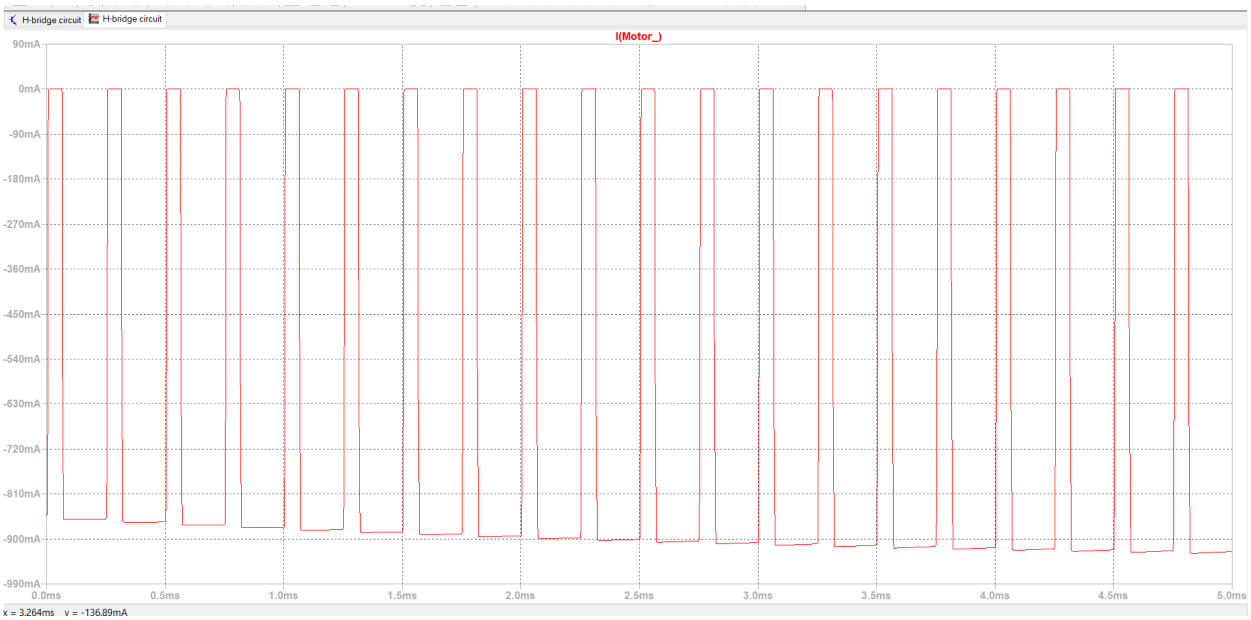


Figure 32 - Motor Counterclockwise Speed Control Testing

This simulation confirmed that speed control had been ascertained, however, the current value was negative, which meant that the motor would be spinning counterclockwise, so the clockwise direction was tested next, with speed at 4V.

Figure 33 - Clockwise Direction Control Testing

Figure 33 shows that, at the same speed (4V), the motor draws 1.3A to 0.0A in a pulsewave, and the current values are positive, which means that current is flowing through the motor the opposite direction as the counterclockwise simulations (clockwise). This simulation confirmed that the circuit's bidirectional modes worked. The clockwise speed was also tested, just to make sure everything worked:

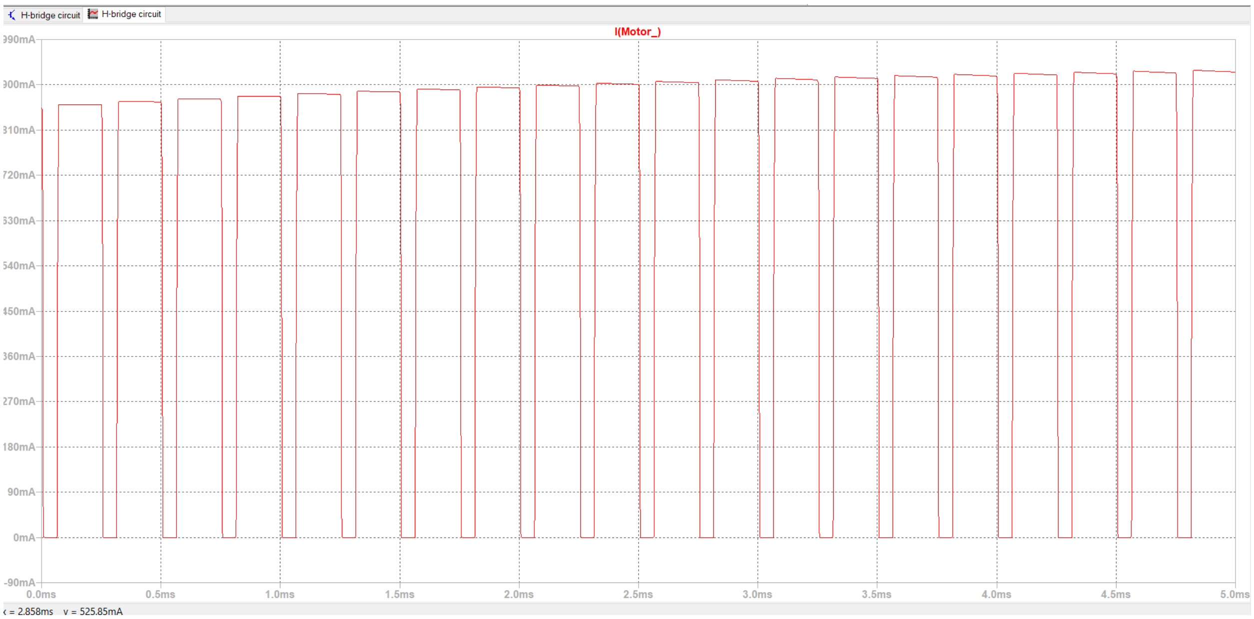


Figure 34 - Clockwise Speed Control Testing

At a "speed" (input voltage) of 3.5V, the motor drew ~0.85A of current, which had a similar value as in the earlier simulation charts. Overall, the currents were fairly ideal, as nothing over 2A was desired for safety concerns, as well as the overall time that the robot should ideally be active was around 45-60 minutes before a battery recharge. The battery can supply 2A for one hour before a recharge, and if each motor is running at 0.5A, the robot should be able to last for an hour (although putting two batteries in parallel to increase current capability was considered). The circuit was capable of running each motor at 0.5A, which was enough current to run each motor at a fairly high speed (this speed did move the robot), while still maintaining the desired time period of 45-60 minutes of usage before recharging the battery.

Next, purely speed control circuits were designed and tested. Since the member who was assigned to the subsystem had done the line follower previously, the member had developed speed control circuits for the motors on the line follower, and similar circuits were designed, but more thoroughly tested before use.

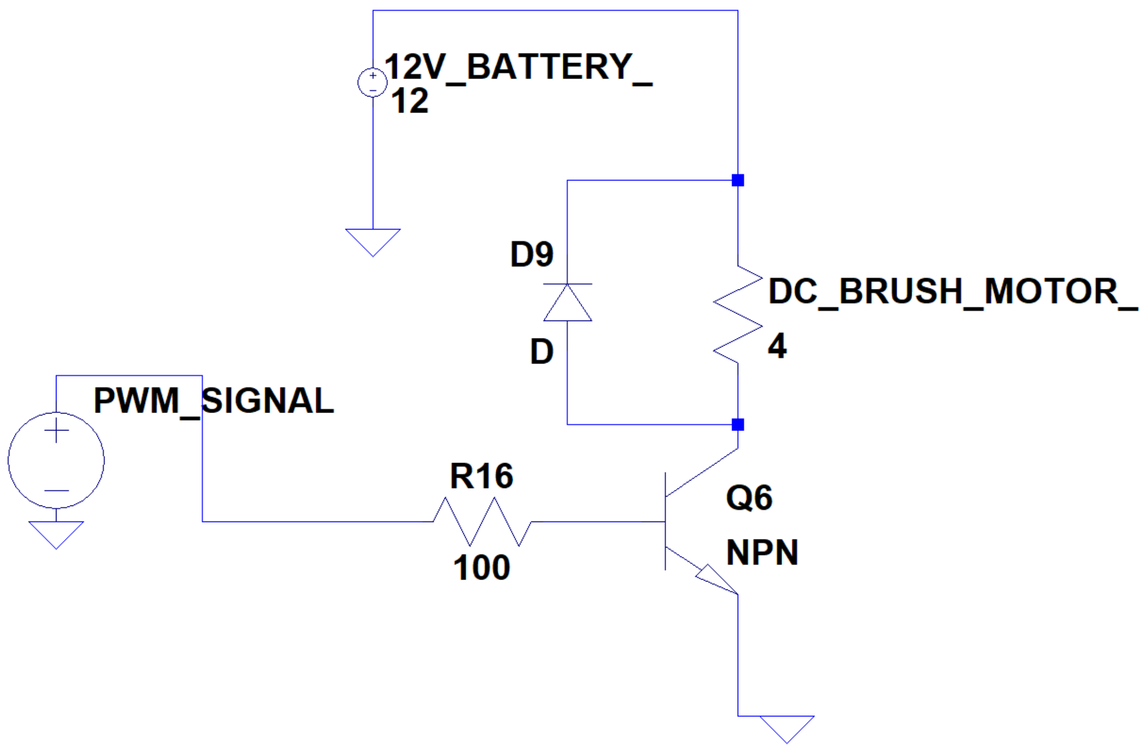


Figure 35 - Motor Speed Control Circuit

The designed circuit consisted of an NPN BJT transistor, which was wired in a very similar way as in the H-bridge circuit. The circuit uses the transistor as a switch that supplies voltage. As the voltage of the PWM signal increases, the transistor will let less and less voltage through to the collector pin. Since the signal is a pulse wave, it will always go from a high value to a low value. NPN transistors turn on when the base voltage is a low value, so when the voltage is 0 from the
PWM signal, the transistor turns on and the voltage across the motor is 12V for a pulse time, and then the transistor turns off. The voltage drop across the motor is then a pulse wave. As the voltage going into the transistor increases, so does the current through the motor, which means an increase in speed happens.

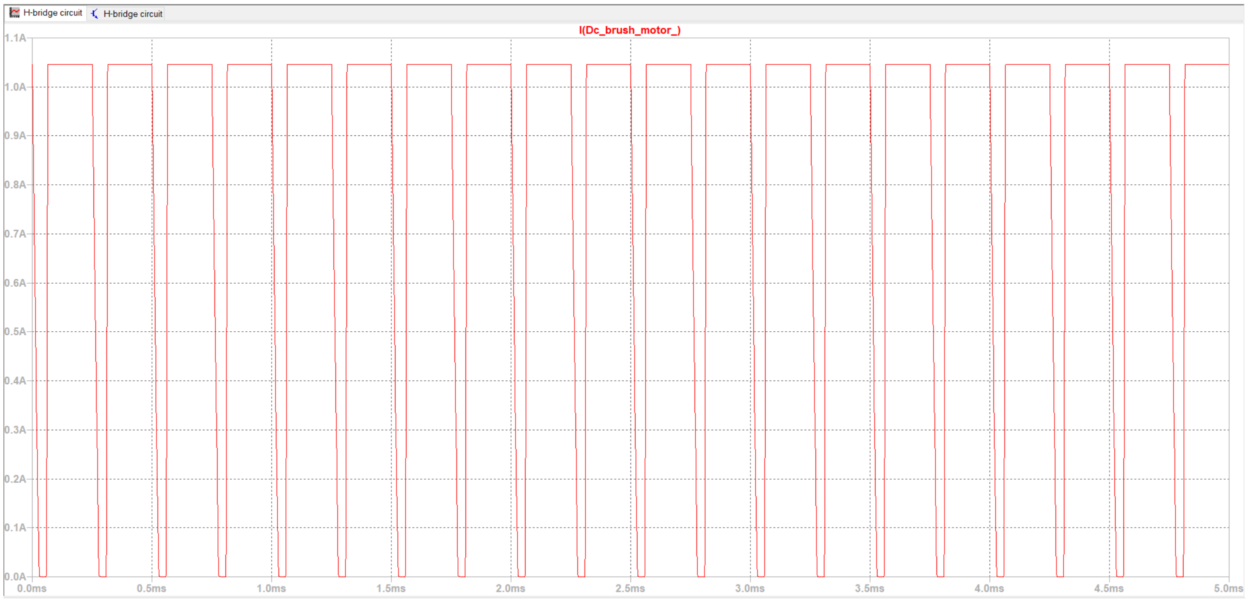


Figure 36 - Motor Control Testing

Figure 36 shows current going through the motor, which is a pulse wave varying from 1.05A to 0A. The base voltage of the transistor was 4V.

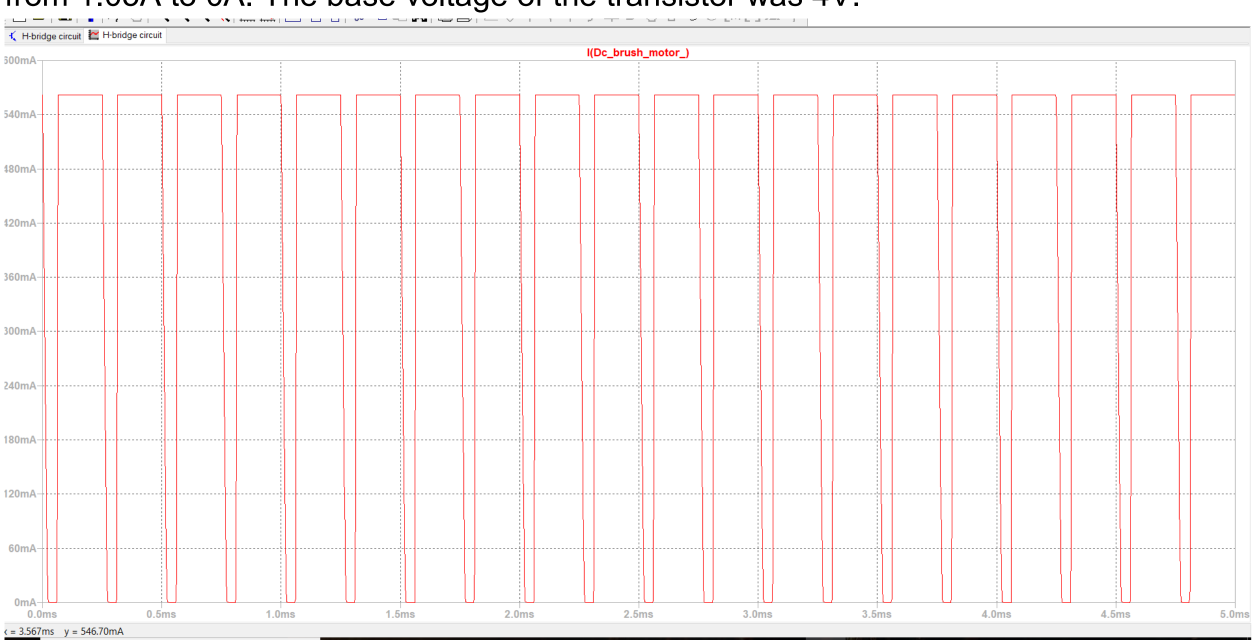


Figure 37 - Motor Control Testing

A speed of 2.5V was tested, with current going through the motor peaking at 0.560A. The simulations showed that speed control could be attained with the circuit designed.

After testing using simulations, circuits were developed. Each circuit developed was tested using motors without any load on the shafts. Once the circuits had produced the correct speeds and/or direction, the circuits were added to the robot's electrical component.

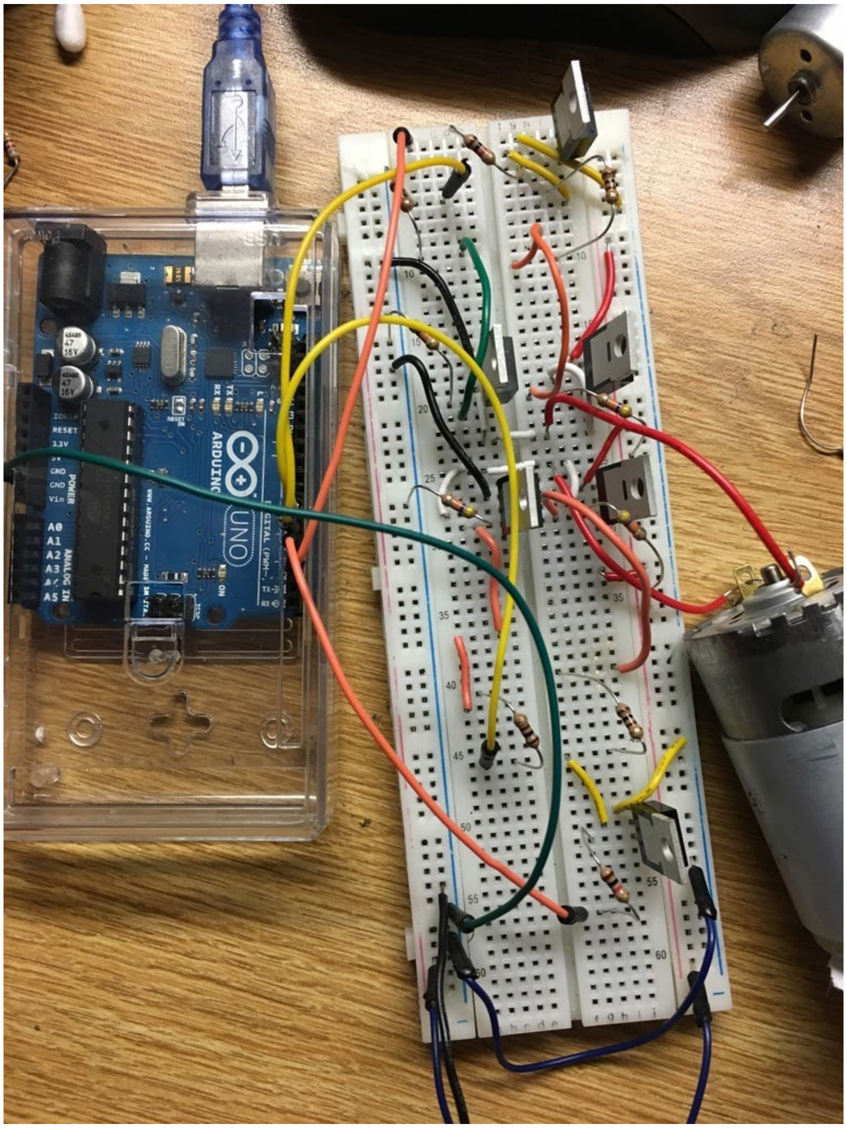


Figure 38 - Motor Control Wiring

Motors were soldered and insulation (heat shrink) was then added onto all open connections to prevent shorting anywhere in the circuit. 22 gauge wiring was used for the circuits, which has a current rating of 7 amperes. Motors were soldered with 16 gauge wire, rated for 22 amperes. These wire gauges were selected to insure that the system could withstand the desired current set by safety standards. Fuses were not added to the system because it could've been too much of a burden for a consumer to change fuses everytime the system exceeded the current rating of the fuses, and ease of use of the Autonomous LCD was a key factor in the design process.

After all wiring and sufficient testing of motor controllers had been completed, the circuits were mounted into the electrical compartment of the Autonomous LCD.

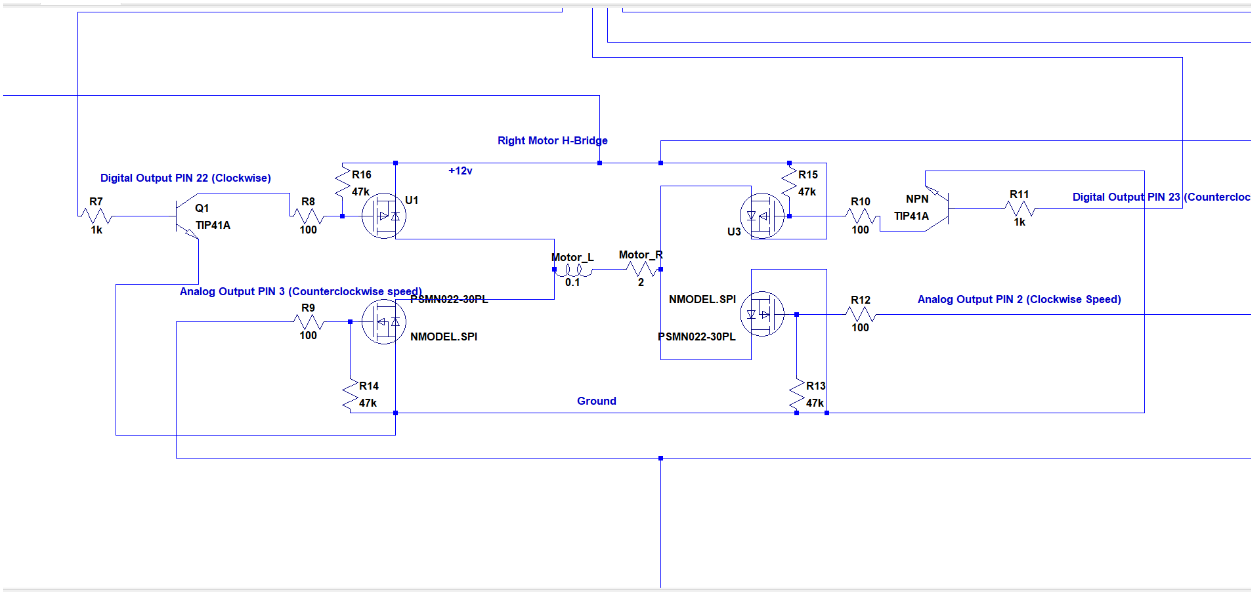


Figure 39 - Final H-bridge Circuit diagram

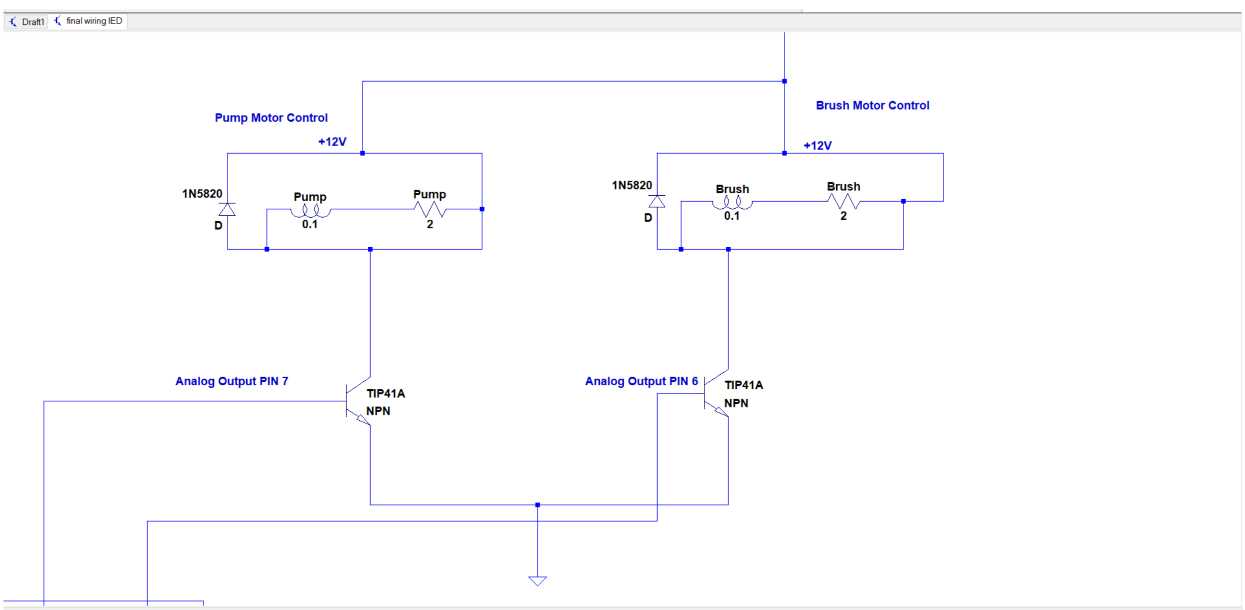


Figure 40 - Final Speed control Circuits

Dimensions

Each motor is a cylinder 4 inches tall with 1 inches diameter. The motors are on the bottom compartment, and 20 inches of 16 gauge wires were soldered onto each motor lead and then snaked up to the electrical compartment. Each wheel motor has a breadboard 4 inches long by 3 inches wide. The pump and scrub motor circuits are combined into one breadboard the same dimensions. Components on each board added 2 inches to the height of the circuit, because of wiring and transistors.

Challenges

The actual design and testing of circuit had to be done fairly quickly, because the drive system was more or less the most integral part of the system. Getting the circuitry done on time was challenging. The preliminary h-bridge circuit was done during Spring Break, but had to be reassessed. The final circuitry used in the actual final design was tested and found to be working to standard on March 28th 2018, which was somewhat behind schedule. Because the design took longer than anticipated, final circuitry could not be manufactured via designing a PCB. Designing a PCB and then printing the product and soldering could've reduced overall final wiring in the final weeks, when wiring was done. This process could've also reduced chances of disconnections on the circuit board, which was a problem with the breadboards used. Overall the system was completed to the team's standards, but further design steps could've been taken to simplify the system overall

Results

During the 3-month long design process, every Autonomous LCD team member learned lessons that will be kept forever. The design the team chose was not the best, but was chosen to integrate every team member's strength so that every team member could provide as much as possible to the design process.

The final product performed fairly well, to the teams expected standards and specifications. The external design is extremely sleek and appealing. The LEDs and start switch provide an extremely easy way for users to interface with the robot. The circuits developed provided above expected power to all motors, and the H-bridge circuit allowed for full control of motor speed and direction. The fluid system succeeded, flow rate was above standards, and no leaks were seen in the tank or pump. The cleaning system also succeeded to standards. The spin speed of the sponge worked well in tandem with the spraying system's high flow rate, which scrubbed basic dirt away easily. The system also cleaned sticky messes, which was one main customer requirement. The navigation of the Autonomous LCD was successful; the wall tracking algorithm was extremely successful. The robot did not bump into any walls when the final software had been uploaded to the arduino.

The product met all technical customer requirements; safety, speed, cleaning, weight, size. The one requirement that was not met was cost. The team may have overspent for products, specifically the acrylic used for the chassis. These costs may be reduced when estimating bulk sales for mass production of the Autonomous LCD.

In conclusion, the design team feels like the project is a success, because all members learned from each other and the design process itself to create a functioning product that passed the team's expected standards.

User Manual

Table of Contents

1. Safety:
    a. Wet Floors:
        i. This device will leave behind a liquid trail as it cleans. Be aware that this liquid may be slippery. Slipping on this liquid can lead to pain and injury.
    b. Electronics:
        i. These electronics generate a strong electric currents. Do not touch these wires while the device is active as the current can lead to potential harm.
        ii. The electronics in this device are in close proximity to conductive fluids. If a leak occurs, do not touch electronics or fluids on the device while the device is active.
    c. Heavy Materials:
        i. The Autonomous LCD is a heavy device and is such can cause harm to small children or individuals if handled improperly.
2. Setup
    a. Checking Battery Charge
        i. Check the charge remaining in the battery of the car before activation.
        ii. To test battery charge, empty the water tank then hold the Autonomous LCD off the ground and activate it.
        iii. Examine the LEDs for battery level indication. If the battery level indicated is low, then the batteries in the Autonomous LCD must be charged.
        iv. For more information on charging look under the Charging Batteries section of Care and Maintenance.
    b. Filling Water Tank

    i. The water tank should be refilled before each use. Simply remove the top cap and fill with water before activating.

3. Placement
    a. Assess Room Layout
        i. Assess the layout of objects in the room and attempt to reduce the number of individual objects in the room so as to allow the Autonomous LCD to drive around the room with greater ease.
        ii. Be aware that the room layout. Non rectangular rooms will have areas skipped over or left uncleaned.
    b. Proper Initial Placement
        i. The ideal place to start the driving of the Autonomous LCD is in a corner of the room facing inward towards another corner of the room.
4. Activation
    a. Power Switch
        i. To turn on the Autonomous LCD simply flip the activation switch.
    b. Active Indicator
        i. When the Autonomous LCD is active, the indicator light will light up to indicate that the Autonomous LCD is working.
5. Monitoring
    a. Standby Mode
        i. If the Autonomous LCD stops driving before it reaches a corner, then it has entered standby mode.
        ii. Standby mode indicates that the Autonomous LCD has insufficient materials to continue cleaning.
        iii. And LED will light up in standby mode to indicate what materials are needed to continue.
    b. Water Levels
        i. If the water level is below the minimum amount required to run, then the Autonomous LCD will answer Standby Mode.
        ii. While in Standby mode the “Low Water” light will be illuminated.
        iii. Once the water tank is refilled to a level above the water level minimum, the Autonomous LCD will continue along its path.
    c. Power Levels
        i. The power level of the Autonomous LCD is indicated by a series of LEDs on the device.
            1. When 4 LEDs are lit up the power is at 100%
            2. When 3 LEDs are lit up the power is at 75%
            3. When 2 LEDs are lit up the power is at 50%
            4. When 1 LED is lit up the power is at 25%

   ii. If the power level is below the minimum amount, then the Autonomous LCD will enter Standby Mode.
   iii. While in Standby Mode, the Autonomous LCD will stop moving and wait to be recharged before it will continue.

6. Care and Maintenance
   a. Charging Batteries
      i. The batteries need to be removed from the unit in order to be recharged. Use the following procedure:
         1. Place the Autonomous LCD upright and verify power is turned off.
         2. Carefully remove the left hand side access panel.
         3. The battery is located directly behind the panel and care should be taken to disconnect and remove the battery without disturbing other electrical components.
         4. Charge the battery with a verified charging device.
         5. Replace battery back into the side access panel and verify connector is secure.
         6. Close access panel.
   b. Cleaning the Brush
      i. Over time the cleaning brush will accumulate dirt and debris from its time cleaning the floors.
      ii. When this happens, the dust and debris must be removed.
      iii. In order to do this, use the following instructions:
         1. Make sure that the Autonomous LCD is switched off.
         2. Empty the water tank by turning the robot upside down over a drain or sink so as not to get any important items or surfaces wet. Continue this until all water has been drained out.
         3. Once all of the water has been drained, leave the robot upside down and locate the scrubber.
         4. Use a rag or paper towel to remove an dust and debris visible on the scrubber. The scrubber will need to be rotated so that all sides can be seen to be cleaned.
         5. Once this is done, the Autonomous LCD can be used again.
   c. Cleaning the Tank
      i. To prevent mold and mildew buildup, rinse the water tank with a disinfectant.
      ii. This should be done regularly so as to maintain the integrity of the Autonomous LCD’s cleaning system.

Appendix A: Selection of Team Project

At the beginning of the project, the group pondered issues in the world that could be solved. Initially, the list that the team came up with a long list of many ideas; some of these ideas were not feasible with a budget of a college student, and the time duration of this project. Eventually, the group all agreed on a common issue that could be solved. This problem was dirty floors. Time is not in favor of a college student, so an autonomous floor scrubber would help tremendously. A group member then came to a realization to make an autonomous floor scrubber. From there, the team discussed ways that this project could be completed. But first, the team needed to see what people would want in an autonomous floor scrubber. The best way to obtain that information was through various interviews and surveys performed on the student population of RPI.

Appendix B: Customer Requirements and Technical Specifications:

Below is a table of information gathered from the interviews.

Interview #	Information Gathered	Customer Need Interpretation
1	• Not too big • Easily Stored • Able to clean messes without breaking	• Small • Sustainability
2	• Able to clean a whole floor • Safe around pets • Does not fall down the stairs	• Safety • Ultrasonic Sensors
3	• Not expensive • Easy to use • Can handle all floors of a house • Completely autonomous	• Can work on multiple surfaces • Not complicated to use • Worth the price • No human assistance needed
4	• Ability to work around furniture • Big liquid tank • Long battery life	• Long run time • Can be used in a crowded room
5	• Can communicate with the customer to know when something is wrong • Can be hit but not break	• Durability • External communications subsystem

Below is a complete list of all the customer requirements and their corresponding specifications.

Customer Needs	**Priority (1-3)**	**Metric**	**Target Value/ Range of Values**
Safety	1	N/A	--
Affordability	3	Cost	$100-150
Small	2	Size	Height: <1.5ft Weight: <15lb Width: <1.5ft
External Communications subsystem	1	N/A	--
Durability	1	Strength	>1000 psi
Battery life	1	Run time	>2 hours
Versatile	2	N/A	--
Ultrasonic Sensors	1	N/A	--
Sustainability	2	Last a long time	>1 year

Easy to use	2	Age to use	>10 years old

Appendix C: Team Members and Their Contributions:

Johnathan

Johnathan was responsible for the design and creation of the fluid control subsystem. The system included a liquid reservoir, pump, tubing, and the chassis of the robot. Johnathan also helped provide specifications necessary to the programming and assisted with minor aspects in writing the software for the robot.

Edward

Edward was responsible for the cleaning system. This system is assigned to Edward because of his major in mechanical engineer, and the building of the cleaning system requires that. This system includes: timing pulley sets, dowels, and sponges. Edward also helps with the building and the testing of the robot. And in the end, Edward helps the team working on the final memo.

Daniel

Daniel was responsible for the motor control subsystem. This subsystem was assigned to Daniel because of his strong background in electronics and circuit, and controls, which was useful when modeling the digital controller for the autonomous LCD. Daniel also spearheaded the final wiring process and provided feedback for the software development phase of the project.

Grant

Grant was responsible for the external communications subsystem. The team felt that Grant would be the best fit for this subsystem due to the fact that he was the only extrovert on the team. The team wanted to have an extrovert in charge of this subsystem to ensure that the Autonomous LCD could communicate with the user in a way that would accommodate everyone. Grant also helped work alongside the other team members with soldering wires, constructing the chassis, interviewing peers, and mounting various items on the robot.

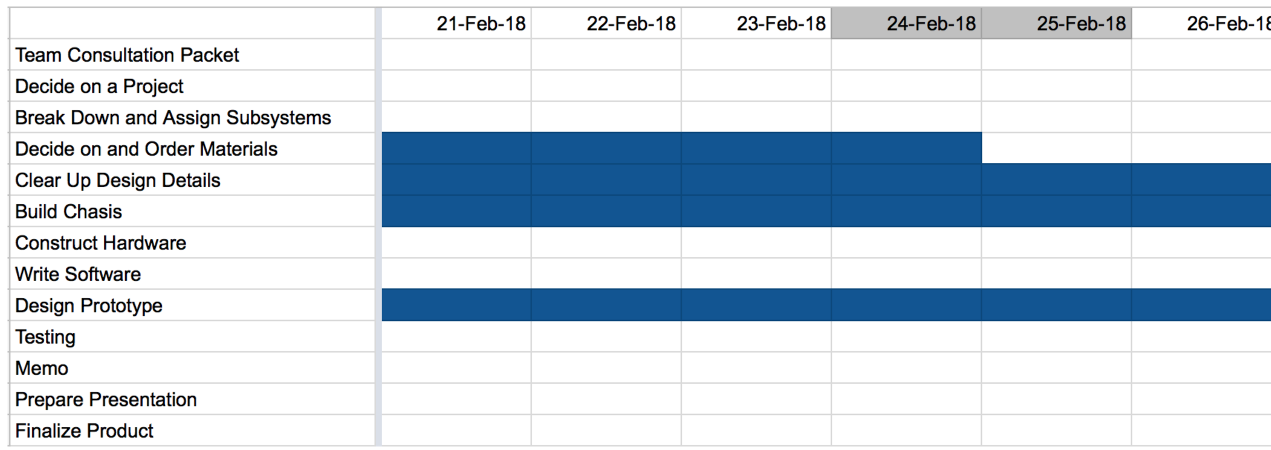
Joseph

Joseph was responsible for the room navigation subsystem. The team put Joseph in this subsystem given his strong coding background and knowledge of algorithms. Aside from his work in coding Joseph also spend much of his time assisting others with their subsystems, constructing the chassis, and helping the team overall by working on the final memo during the building and design process.

Appendix D: Gantt Chart

	Status	Owner	Start
Team Consultation Packet	100%	All	02/12
Decide on a Project	100%	Grant	02/12
Break Down and Assign Subsystems	100%	Edward	02/15
Decide on and Order Materials	100%	Joseph	02/15
Clear Up Design Details	100%	Daniel	02/20
Build Chasis	80%	Johnathan	02/20
Construct Hardware	90%	Grant & Edward	03/10
Write Software	60%	Daniel, Joseph, & Johnathan	03/10
Design Prototype	100%	Johnathan and Joseph	02/18
Testing	35%	Edward	04/05
Memo	40%	All	03/01
Prepare Presentation	15%	Grant	04/18
Finalize Product	20%	Daniel	04/05

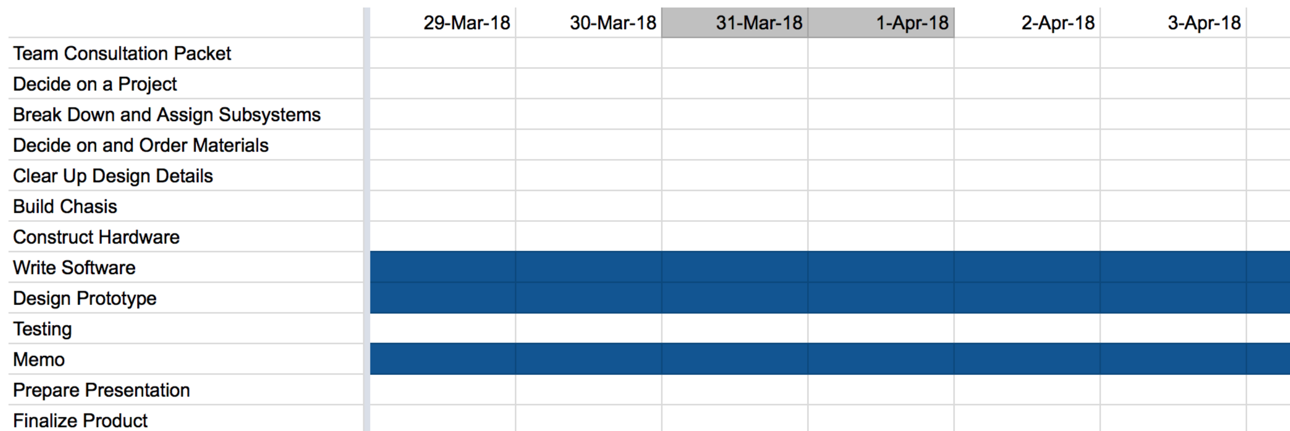
	12-Feb-18	13-Feb-18	14-Feb-18	15-Feb-18	16-Feb-18	17-Feb-18
Team Consultation Packet						
Decide on a Project						
Break Down and Assign Subsystems						
Decide on and Order Materials						
Clear Up Design Details						
Build Chasis						
Construct Hardware						
Write Software						
Design Prototype						
Testing						
Memo						
Prepare Presentation						
Finalize Product						


21-Feb-18
22-Feb-18
23-Feb-18
24-Feb-18
25-Feb-18
26-Feb-18
Team Consultation Packet
Decide on a Project
Break Down and Assign Subsystems
Decide on and Order Materials
Clear Up Design Details
Build Chasis
Construct Hardware
Write Software
Design Prototype
Testing
Memo
Prepare Presentation
Finalize Product

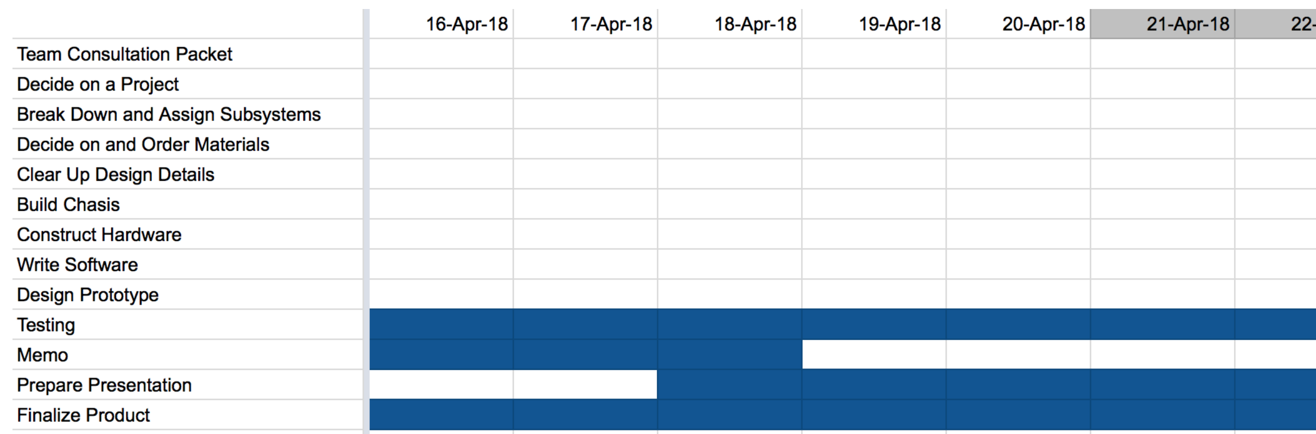
	2-Mar-18	3-Mar-18	4-Mar-18	5-Mar-18	6-Mar-18	7-Mar-18
Team Consultation Packet						
Decide on a Project						
Break Down and Assign Subsystems						
Decide on and Order Materials						
Clear Up Design Details						
Build Chasis						
Construct Hardware						
Write Software						
Design Prototype						
Testing						
Memo						
Prepare Presentation						
Finalize Product						

	11-Mar-18	12-Mar-18	13-Mar-18	14-Mar-18	15-Mar-18	16-Mar-18
Team Consultation Packet						
Decide on a Project						
Break Down and Assign Subsystems						
Decide on and Order Materials						
Clear Up Design Details						
Build Chasis						
Construct Hardware						
Write Software						
Design Prototype						
Testing						
Memo						
Prepare Presentation						
Finalize Product						

	20-Mar-18	21-Mar-18	22-Mar-18	23-Mar-18	24-Mar-18	25-Mar-18
Team Consultation Packet						
Decide on a Project						
Break Down and Assign Subsystems						
Decide on and Order Materials						
Clear Up Design Details						
Build Chasis						
Construct Hardware						
Write Software						
Design Prototype						
Testing						
Memo						
Prepare Presentation						
Finalize Product						


29-Mar-18
30-Mar-18
31-Mar-18
1-Apr-18
2-Apr-18
3-Apr-18
Team Consultation Packet
Decide on a Project
Break Down and Assign Subsystems
Decide on and Order Materials
Clear Up Design Details
Build Chasis
Construct Hardware
Write Software
Design Prototype
Testing
Memo
Prepare Presentation
Finalize Product

	7-Apr-18	8-Apr-18	9-Apr-18	10-Apr-18	11-Apr-18	12-Apr-18
Team Consultation Packet						
Decide on a Project						
Break Down and Assign Subsystems						
Decide on and Order Materials						
Clear Up Design Details						
Build Chasis						
Construct Hardware						
Write Software						
Design Prototype						
Testing						
Memo						
Prepare Presentation						
Finalize Product						


16-Apr-18
17-Apr-18
18-Apr-18
19-Apr-18
20-Apr-18
21-Apr-18
22-
Team Consultation Packet
Decide on a Project
Break Down and Assign Subsystems
Decide on and Order Materials
Clear Up Design Details
Build Chasis
Construct Hardware
Write Software
Design Prototype
Testing
Memo
Prepare Presentation
Finalize Product

	25-Apr-18
Team Consultation Packet	
Decide on a Project	
Break Down and Assign Subsystems	
Decide on and Order Materials	
Clear Up Design Details	
Build Chasis	
Construct Hardware	
Write Software	
Design Prototype	
Testing	
Memo	
Prepare Presentation	
Finalize Product	

Appendix E: Expense Report

Subsystem	**Item**	**Cost**
Fluid Control	Spray Nozzles (x2)	$16.78
Fluid Control	Acrylic Sheet	$42.90
Motor Control	Flange-Mount Ball Transfer	$9.12
Fluid Control	⅜” ID Vinyl Tubing	$7.79
Fluid Control	12V DC Water Pump	$19.99
Motor Control	Arduino Mega	$14.99
Fluid Control	Rubber Cement	$7.00
Fluid Control	Brass Adapter	$11.22
Fluid Control	PVC Cap	$.49
Fluid Control	PVC Bushing	$.59
Fluid Control	Nylon Tee	$2.87
Fluid Control	Hose Clamps	$4.98
Motor Control	Motor (x4)	$31.00
Motor Control	Transistors, various electrical components	$12.98
Cleaning System	Scotch Brite Pads	$4.99
Cleaning System	Metal Rod	$5.29
Cleaning System	Timing Pulley Set	$9.99

Appendix F: Statement of Work

Team:
Grant Tragni, Jonathan Corbin, Joseph Catalano, Daniel Ray, Haocheng (Edward) Zheng

Semester Objectives

1. Design and build an autonomous floor scrubber similar to a Roomba, but instead of a vacuum function, it scrubs and dries a floor.
2. Organize a professional technical memo.
3. Communicate with the public for ideas of improvement by taking surveys.
4. Complete the project at least 3 weeks before the deadline for time to beta test.

Approach

A regular meeting schedule with constant team communication through Slack, Google Drive, and Facebook messenger. This communication is necessary due to the overlap with all of our subsystems and individual workload.

Deliverables and Dates

1. Have all materials ordered (03/01/2018)
2. Finalize Design Details (03/03/2018)
3. Have the chassis built (03/10/2018)
4. Start working on hardware/software (03/10/2018)
5. Finish Hardware (03/20/2018)
6. Finish Software (04/04/2018)
7. Complete Prototype (04/05/2018)
8. Testing (04/05/2018)
9. Memo done (04/18/2018)
10. Presentation preparation (04/18/2018)
11. Presentation complete (04/25/2018)
12. Finalized product (04/25/2018)
13. Review (04/25/2018)
14. Testing Day (04/26/18)
15. Presentation Day (04/30/2018)
16. Hand in Memo Electronically (05/02/2018)

Appendix G: Lessons Learned

The team learned many skills and lessons in the product development process. Perhaps the most important one being team communication, especially between subsystems. For example, the motor control subsystem involved creating circuits that could be controlled by software. The circuit had to be developed and then was demonstrated to the team, with example software developed that told the team how to turn specific motors on, off, turn forwards, backwards, etc. Without this communication, developing software would've taken much longer.

The second most important skill/lesson learned in the designing process was modeling systems. Modeling via computers can save extensive time and testing. The team modeled the whole chassis by designing every mechanical part in Solidworks 3-D. A complete assembly was made, which made the building process much easier, because ensuring that every piece would fit just right in CAD meant that the pieces would most likely fit very well when putting everything together, which meant that no pieces needed to be remade, which saved using more acrylic stock than needed (acrylic price was high). The modeling process for the Autonomous LCD's digital controller was also extremely helpful. The digital controller modeling provided feedback when deciding control constants used in the algorithm, which saved time physically testing the algorithm by seeing the response of the Autonomous LCD's wall tracking, which was time that the team did not have.

The third most important skill/lesson learned was how to convey information to people who knew nothing about the Autonomous LCD. Even if an engineer develops an extremely useful device, if there is not documentation on how to use the device, there is almost no point to sell the device. The team feels like after writing this memo and giving a presentation about the Autonomous LCD, customers have learned the basics behind the design and process of the Autonomous LCD.

Appendix H: Fluid Control System Dynamics Model

Pump Statistics:

Pump	Voltage (Volts)	Current (Amps)	Flow Rate (m^3/s)	Pressure (kPA)
	12	3	6.31*10^-5	552

Additional Known Information:

- Total Volume = .00246 m^3
- Pressure One = Atmospheric Pressure = 101.3 kPa
- Height Change = .0889 m
- K Entrance = .78
- K Bend = .7

Equation Derivation and Modeling:

$$\frac{P_1}{\rho * g} + \frac{V_1^2}{2 * g} + z_1 + h_p = \frac{P_2}{\rho * g} + \frac{V_2^2}{2 * g} + z_2 + h_T + \sum h_L$$

$$V_1 = 0 - due\ to\ large\ area\ of\ tank$$

$$h_T = 0 - due\ to\ no\ turbine\ head\ loss$$

$$h_p = \frac{W}{\rho * \gamma * g} - pump\ head\ equation$$

$$\sum h_L = (K_{entrance} + 4 * K_{bend}) * \frac{V^2}{2 * g} - sumnation\ of\ the\ total\ minor\ losses$$

$$W = I * V$$

$$V = \frac{4 * \gamma}{\pi * D^2}$$

Substituting into the original equation and solving for the second velocity yields:

$$V_2 = \sqrt{\frac{2 * (P_1 - P_2)}{\rho} + 2 * g * (z_1 - z_2) + \frac{2 * W}{\rho * \gamma} - (K_{entrance} + 4 * K_{bend}) * (\frac{4 * \gamma}{\pi * D^2})^2}$$

After substituting in known values:

$$V_2 = 33.76\ \frac{m}{s}$$

Then:

$$\gamma_{final} = V_2 * \pi * D^2 = .000027 \frac{m^3}{s}$$

$$Mass\ Flow\ Rate = \rho * \gamma_{final} = .0265 \frac{kg}{s}$$

$$Total\ Mass = 2.46\ kg$$

$$Total\ time = \frac{Total\ Mass}{Mass\ Flow\ Rate} = 92.8\ s = 1.55\ min$$

The previous equations are all derivations of the steady, incompressible flow energy equation. The result shows the total amount of time required to completely empty the tank of 2.46 kg of fluid. The total run time of the pump can then be extended by introducing an on/off cycle into the system to offset the total continuous run time of the pump. In theory, the on/off cycle would decrease fluid usage and costs, while increasing pump life and reliability.

Appendix I: PID Control

The transfer function of the Autonomous LCD is as follows:

$$H(s) = \frac{K_{Motor}}{Ts + 1} \frac{Speed}{Volts}$$

Where controlling the motor speed is the objective, so:

$$H(s) = \frac{Speed(s)}{Voltage_{In}(s)}$$

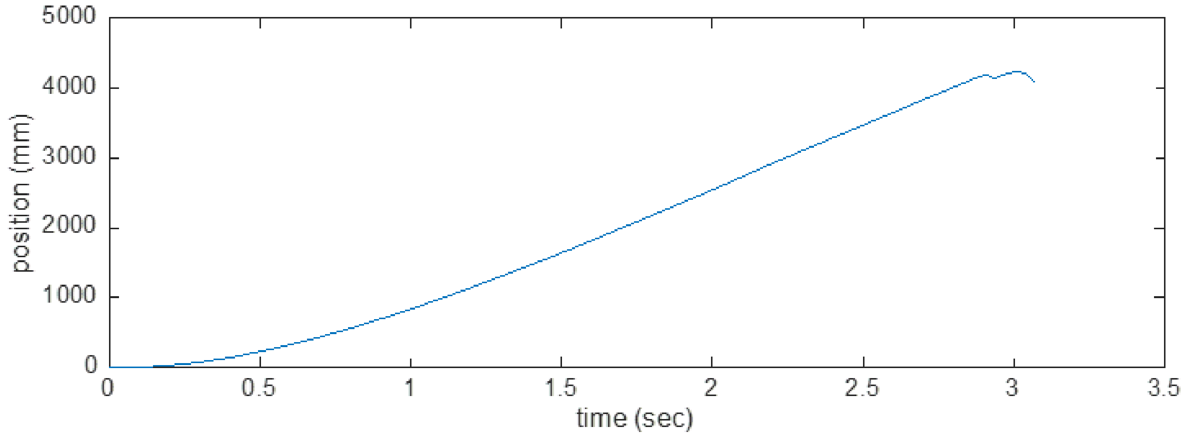
Where the input voltage is a pulse width modulated signal coming from the Arduino's analog output pins. The speed output is not directly speed, but is a voltage correlating to speed. As this value rises, motor speed rises, and as this speed decreases, motor speed decreases. More information on how this works can be found in motor control subsystem documents.

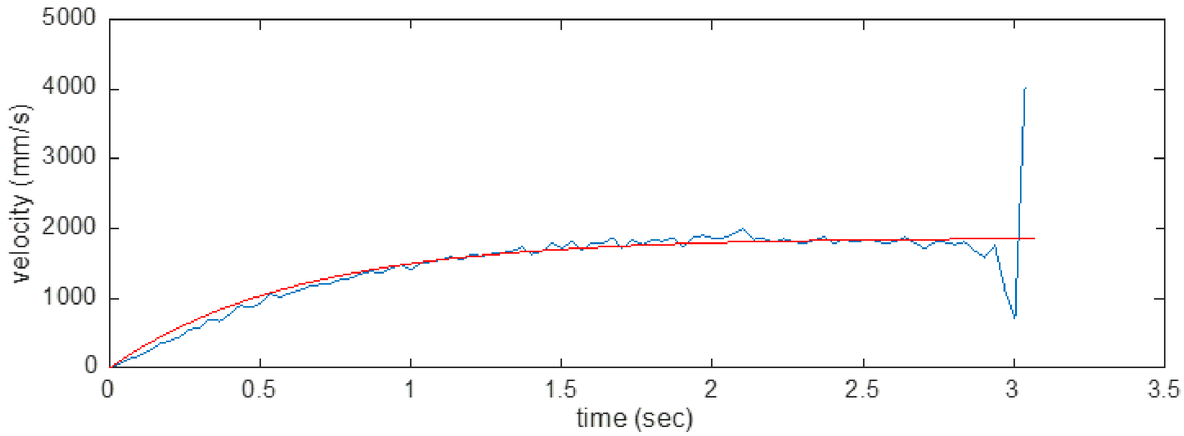
In the transfer function,

$$T = Time\ Constant\ of\ motor\ (seconds) = 0.62s$$

$$K_M = Motor\ DC\ gain\ \left(\frac{m}{\frac{s}{V}}\right) = 0.406 \frac{m}{\frac{s}{V}} (approximate)$$

$$s = complex\ frequency\ (the\ transfer\ function\ is\ modeled\ using\ Laplace)$$





Measured step-inputs of position and speed of the motor showing first order responses. The time constant of the motor was obtained by measuring the time when the velocity had reached 63.2% of the steady state speed, which was 0.62 seconds (this is the motor time constant). The top graph was obtained via filming a vehicle that was turned on at t = 0, and then a voltage of 12V was applied to the motor to obtain max speed. The position of the vehicle was obtained via tracker software, and then was differentiated to obtain velocity vs time, which allowed for examination to obtain the time constant of the motor.

The transfer function is then:

$$H(s) = \frac{0.406}{0.62s + 1} \frac{Speed}{Volts}$$

The inductance of the motor was incredibly small, so the electrical time constant was neglected.

This transfer function only represents the system without any control loops or gain. As a result, the system does not perform well, so control is needed.

Using basic laplace transforms, the transfer function can be modified so that the gain of the system can be modified so that a good response can be attained:

$$\mathcal{L}(K_P) = K_P, proportional\ term$$

$$\mathcal{L}\left(\int K_I * f(t)dt\right) = K_I \frac{1}{s}, Integral\ term$$

$$\mathcal{L}\left(K_D * \frac{d}{dt}\right) = K_D s, Derivative\ term$$

These terms are then multiplied into the transfer function to obtain a compensated system with gain:

$$H(s) = \frac{0.406(K_d s^2 + K_p s + K_I)}{s * (0.62s + 1)} \frac{Speed}{Volts}$$

The transfer function is put into polynomial form, as fractions cannot be analyzed.

These control terms can be used to compensate the system, which allows tuning. This was done by using the MATLAB software, to simulate responses so that a good estimate could be used when implementing the actual code to the Arduino.

Since the control is done in a 100% digital system, the output can be viewed in two ways: either the output is a digital voltage in bits (which is also an analog voltage), that directly represents the speed of the motor, or one can view the output as a speed directly. However, the speed cannot be exactly measured, because a tachometer or digital encoder was not used, which prevented speed measurements.

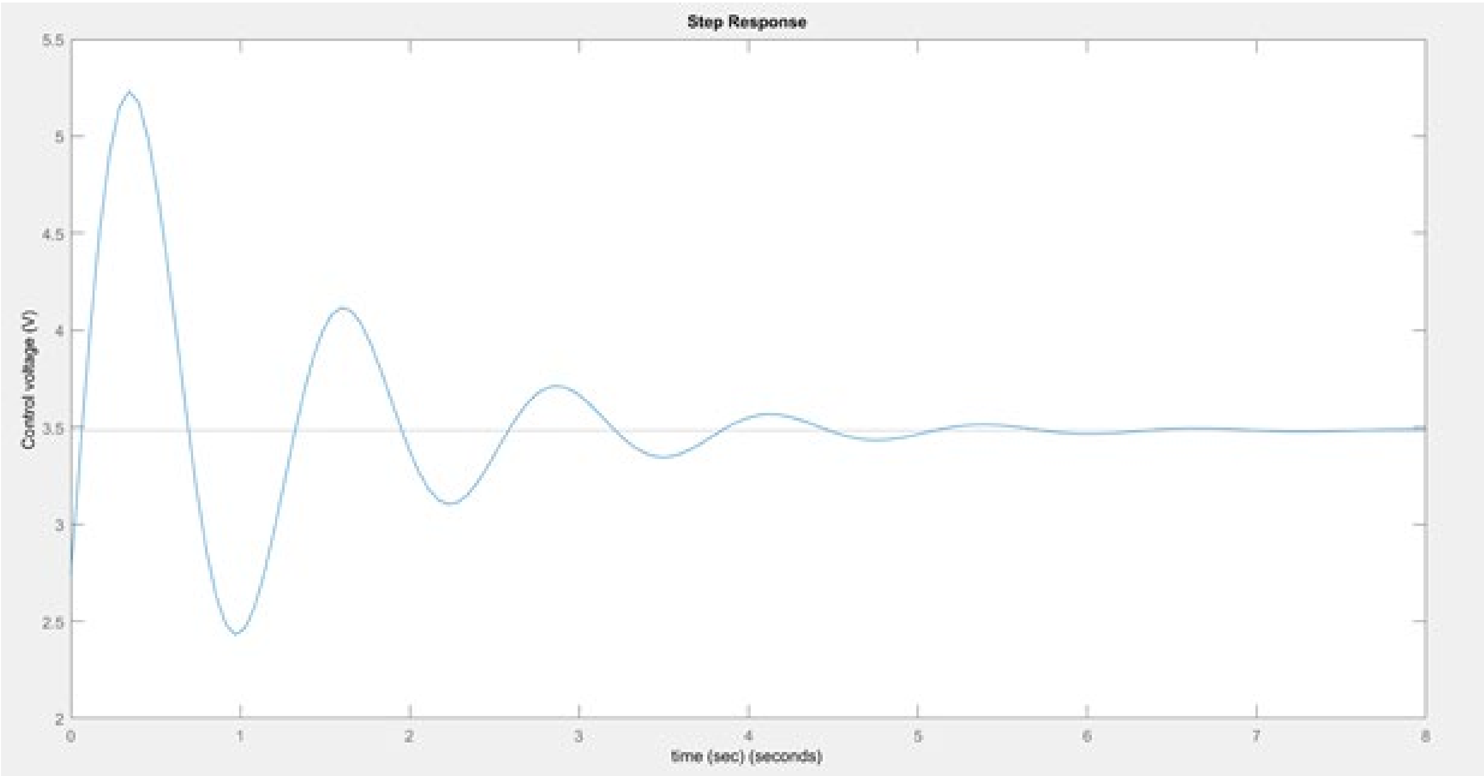
The desired speed of the system was set to a digital value of 180 bits (out of 255). This directly corresponds to a voltage of 3.86V, which outputs on an analog pin.

The idea of the control system is to measure the distance of one of the autonomous LCD's sides from a wall or obstacle, by using the ultrasonic sensors. The system wants to stay 15 cm away from a wall. If the left sensor reads a value from 45 to 0 cm, software will activate a loop where the system will keep 15 cm away from the left wall. The system does this by controlling the motor speeds of the left and right motors. If the robot is 30 cm away, the "error" is -15 cm, and the motor speeds are compensated so that the Autonomous LCD will steer slightly left until the error value is zero (the ultrasonic rangers read 15cm). Since the Autonomous LCD needs to move left, the right motor's speed is increased and the left motor's speed is decreased, until the robot is 15 cm away from the left wall. If the Autonomous LCD is less than 15 cm away from the left wall, the "error" will be positive, and the left motor's speed will increase, while the right motor's speed will decrease, turning the robot away from the wall. A similar algorithm is used for the right wall. If the center sensor detects a wall or obstacle, the Autonomous LCD will drive backwards for ~1 second, and then turn right if on a left wall, or turn left if on a right wall to prevent collision. The control algorithm is directly implemented in the wall detection software.

The control algorithm required extensive testing to find suitable constants for the proportional, integral, and differential gains. Actual testing using the Autonomous LCD itself was fairly difficult, because of time constraints. MATLAB simulation results provided feedback as to what terms needed to be.

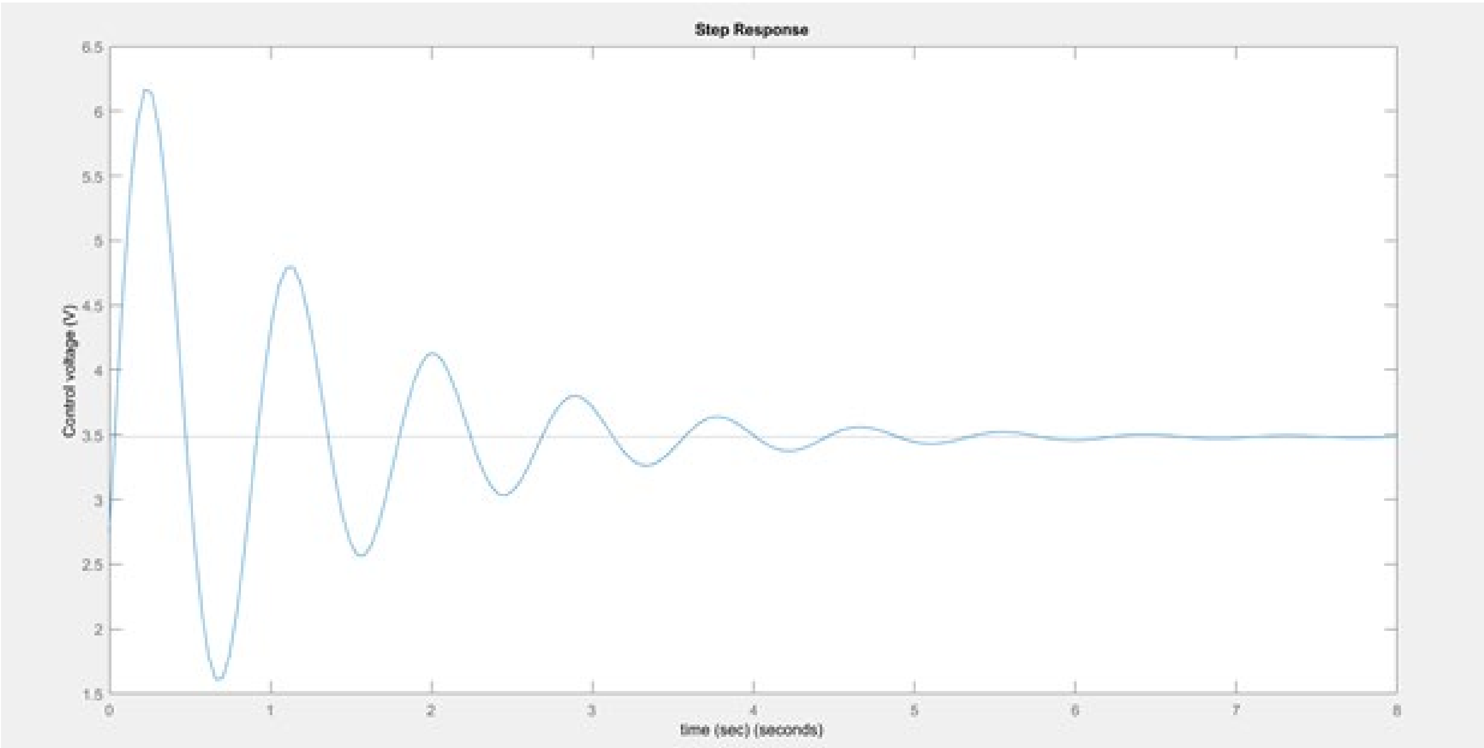
Different constants were tested, with results below:
A step function where the starting voltage of a motor was 2.7415V (140 bits). The algorithm then aims to correct itself and settle at a voltage of 3.5V (180 bits), which would correlate to a straightened out robot.

$$K_p = 13, K_i = 3, K_d = 0$$



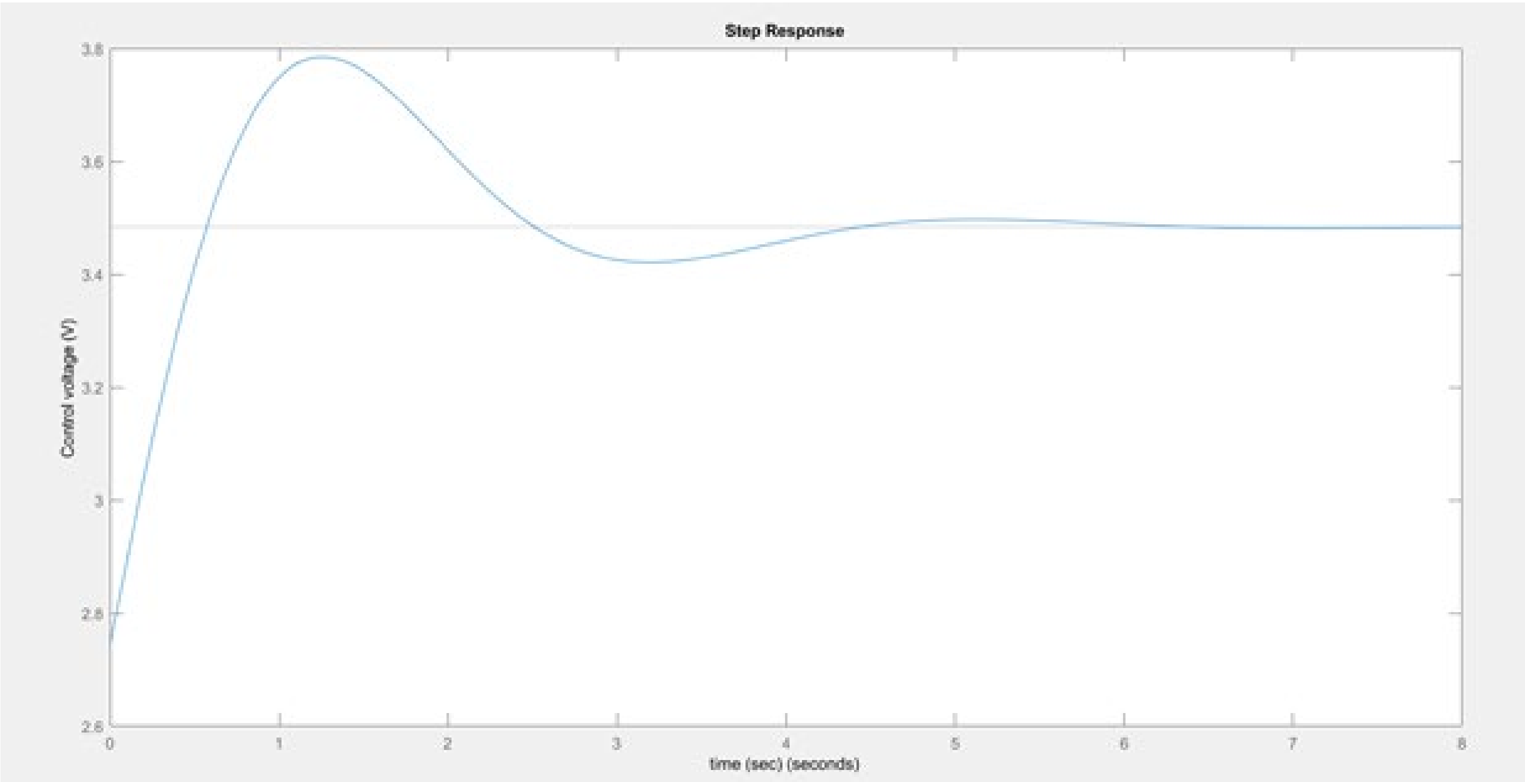
The response is terrible. The voltage overshoots to ~5.25V, which means that the motor speed is "overestimated." The system will most likely bump into a wall, which is not desired. The system does settle at the correct speed, but it is after 7 seconds, which is much too long.

Kp = 13, Ki = 6, Kd = 0



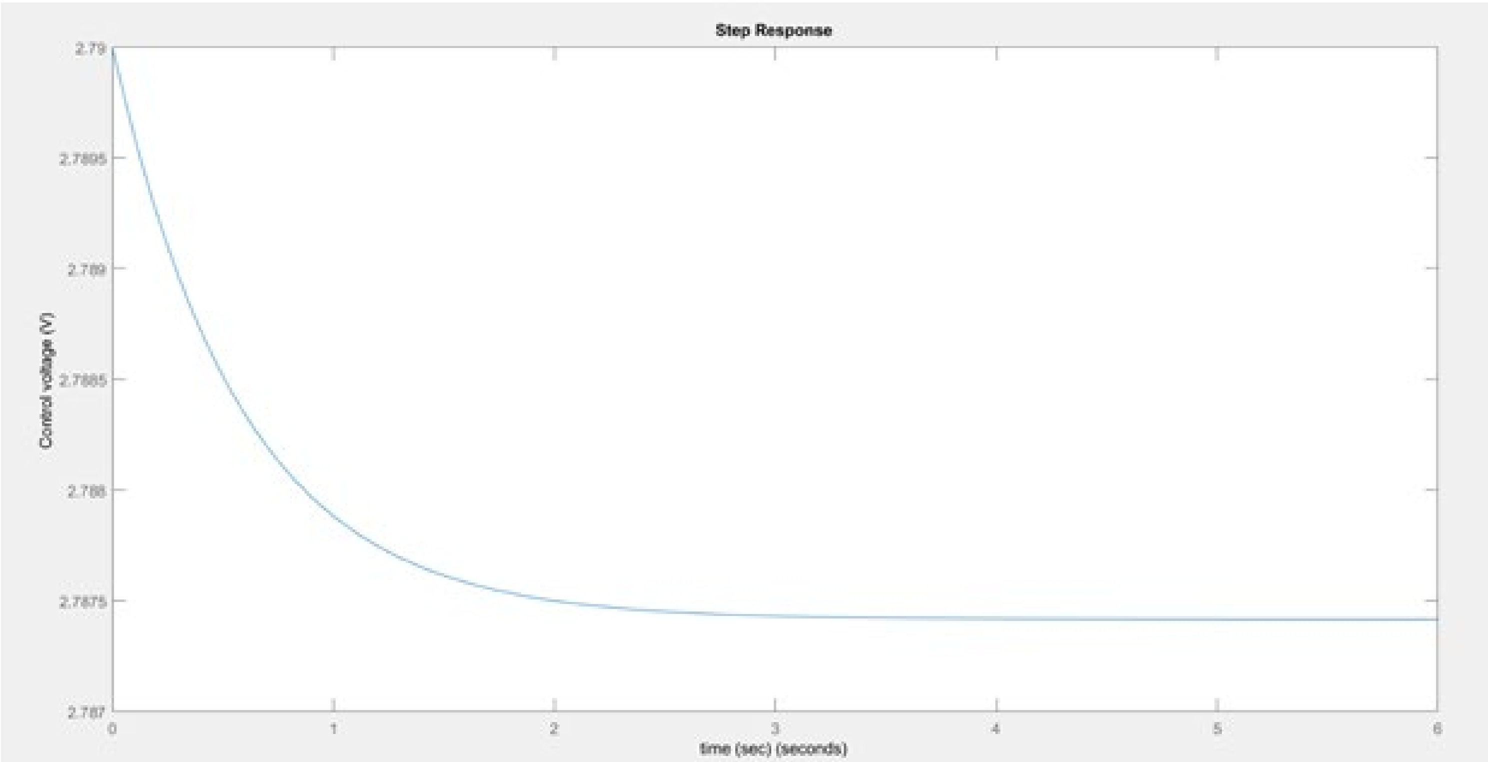
The response is even worse. The time it takes the system to settle is about the same as before, but overshoot is terrible. Doubling the integral term made the system more unstable.

Kp = 5, Ki = 1, Kd = 0



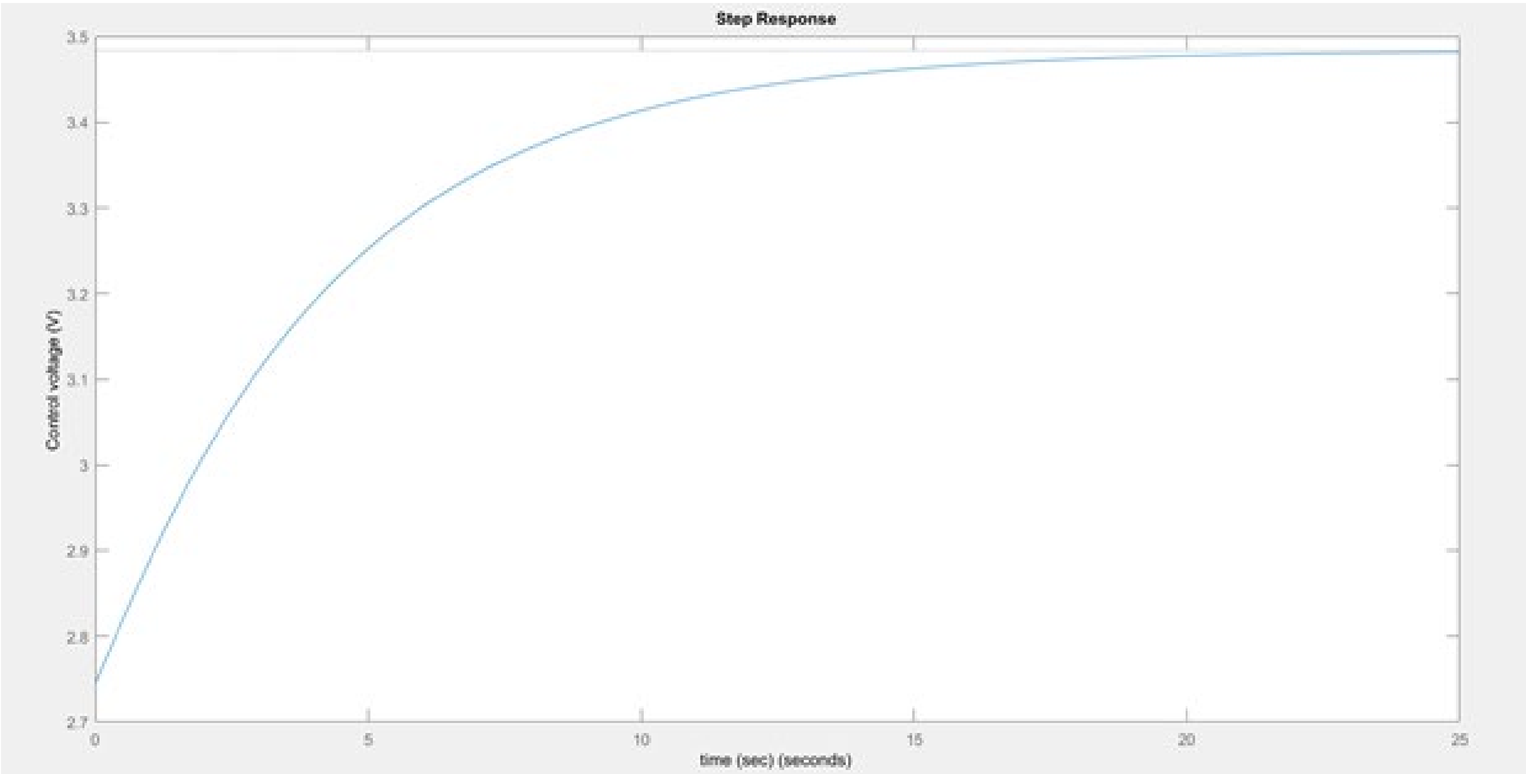
The system is better. The response is less oscillatory, but settling time is still too long. More damping could be added to compensate for the system's oscillatory form.

Kp = 5, Ki = 1, Kd = 3



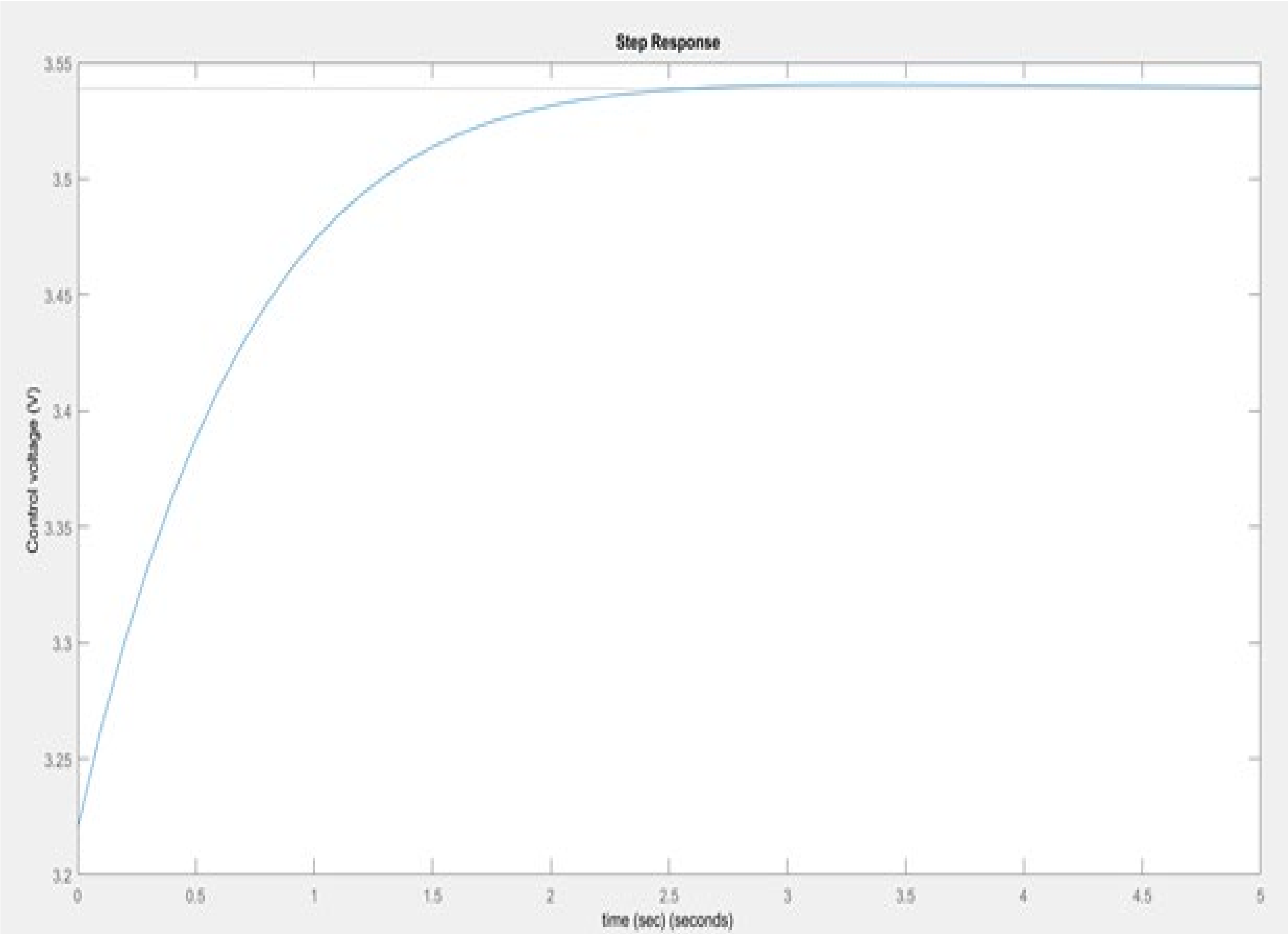
The result is puzzling. Control laws say derivative control adds damping, but this is a very overdamped system. The motor speed barely changed, because the voltage decays slightly. The system has too much derivative control.

Kp = 5, Ki = 0.1, Kd = 0



This result looks much, much better. There is no overshoot. However, looking closer at the time-axis, one can see that the settling time is too long. Increasing the proportional control constant and increasing damping (derivative) should help.

Kp = 28, Ki = 0.01, Kd = 0.005



The result of the control algorithm is fairly successful. Settling time is 2.5 seconds, which is longer than anticipated, but there is no overshoot, ensuring that the Autonomous LCD should not bump into a wall under the algorithm. These control constants and these simulations gave the team a very good starting point to determine how to proceed when implementing this data into actual software, and finalizing the digital controller.

Appendix I: Pseudocode for Room Navigation

```
define turn
        turn 90 degrees in the desired direction

go = 1
```

```
turn_direction = left
While go == 1
	update battery reading
	move forward
	if (the robot has insufficient cleaning fluid or battery power)
		stop
		turn on buzzer
		led for corresponding need is lit
	if (something is blocking the robot)
		stop
		read sensor in turn_direction
		if (robot is still blocked)
			go = 2
			exit the while loop
		else
			turn turn_direction
			move forvard one robot width
			turn turn_direction
			change turn_direction to left if it was right and right if it was left
	else
		keep moving
change turn_direction to left if it was right and right if it was left
turn turn_direction
while go == 2
	update battery reading
	move forward
	if (the robot has insufficient cleaning fluid or battery power)
		stop
		turn on buzzer
		led for corresponding need is lit
	if (something is blocking the robot)
		stop
		read sensor in turn_direction
		if (robot is still blocked)
			go = 0
			exit the while loop
		else
			turn turn_direction
			move forvard one robot width
			turn turn_direction
```

change turn_direction to left if it was right and right if it was left
else
keep moving

Appendix J: Demo Code

```
/*
 * Arduino Code for Autonomous LCD Cleaner Robot
 *
 */
```

```
//left drive motor pins
#define left_drive_cw_bjt  22 //controls bjt for cw
#define left_drive_ccw_bjt  23 //controls bjt for ccw
#define left_drive_cw_speed  2 //controls speed for ccw
#define left_drive_ccw_speed  3 //controls speed for cw

//right drive motor pins
#define right_drive_ccw_bjt  24 //controls bjt for cw
#define right_drive_cw_bjt  46//controls bjt for ccw
#define right_drive_cw_speed  10 //controls speed for ccw
#define right_drive_ccw_speed 11 //controls speed for cw

//Brush/scrubber motor
#define brush_speed  6

//pump motor
#define pump_motor  7

//water level sensor
#define h20_sensor  A0
unsigned int water_level; //variable to store data

//ultrasonic rangers

//center
#define trigger_center  44
#define echo_center  36
long duration_center;
unsigned int distanceCM_center;

//right
#define trigger_right  48
#define echo_right  49
long duration_right;
unsigned int distanceCM_right;
unsigned int prev_right_distance;
//left
#define trigger_left  30
#define echo_left  32
```

```
long duration_left;
unsigned int distanceCM_left;
unsigned int prev_left_distance;
#define pulse 1500

//LED's and buzzers, battery analog input
#define battery_100  50
#define battery_75  42
#define battery_50  53
#define battery_25  38
#define buzzer  33
#define liquid_tank  18
#define battery_input  A1
unsigned int battery_voltage;


unsigned char r_motor_speed;
unsigned char l_motor_speed;


signed long error; //error in distance
signed long prev_error;

char turn; //1 or 0, 1 for right turn, 0 for left




void setup()
{
  Serial.begin(9600);
  //left drive motor
  pinMode(left_drive_cw_bjt, OUTPUT);
  pinMode(left_drive_ccw_bjt, OUTPUT);
  pinMode(left_drive_cw_speed, OUTPUT);
  pinMode(left_drive_ccw_speed, OUTPUT);
  //right drive motor
  pinMode(right_drive_cw_bjt, OUTPUT);
  pinMode(right_drive_ccw_bjt, OUTPUT);
  pinMode(right_drive_cw_speed, OUTPUT);
```

```
  pinMode(right_drive_ccw_speed, OUTPUT);
  //scrubber motor
  pinMode(brush_speed, OUTPUT);
  //pump motor
  pinMode(pump_motor, OUTPUT);
  //water level sensor
  pinMode(A0, INPUT);
  //rangers
  pinMode(trigger_center, OUTPUT);
  pinMode(echo_center, INPUT);
  //2
  pinMode(trigger_right, OUTPUT);
  pinMode(echo_right, INPUT);
  //3
  pinMode(trigger_left, OUTPUT);
  pinMode(echo_left, INPUT);
  //LED's and buzzers
  pinMode(battery_100, OUTPUT);
  pinMode(battery_75, OUTPUT);
  pinMode(battery_50, OUTPUT);
  pinMode(battery_25, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(18, OUTPUT);
  //battery voltage level
  pinMode(A1, INPUT);
}

void loop(void)
{


  //Read battery and water level
  read_water();
  read_battery();



  //read ultrasonics
  distance_center();
  distance_right();
```

```
  distance_left();
  //pump & brush motors
  pump_water();
  turn_brush();

  //run main code

  wall_check();

 /*reading data for demo
 //Serial.print("Left Distance: ");
 //Serial.print(distanceCM_left);
//Serial.print("\n");

 //Serial.print("Right Distance: ");
// Serial.print(distanceCM_right);
 //Serial.print("\n");

  //Serial.print("Center Distance: ");
  //Serial.print(distanceCM_center);
  //Serial.print("\n");

  Serial.print("Left Motor Speed: ");
  Serial.print(l_motor_speed);
  Serial.print("\n");

  Serial.print("Right Motor Speed: ");
 Serial.print(r_motor_speed);
  Serial.print("\n");

 // Serial.print("Error Signal: ");
 //Serial.print(error);
 //Serial.print("\n");
 */

}
//backup function, very basic
void clean(void)
{
  go_backwards();
```

```
  if(distance_center < 30)
  {
    go_straight();
    delay(500);
    turn_right();
    delay(1000);
    go_backwards();
  }
  if(distance_right < 30)
  {
    go_straight();
    delay(500);
    turn_left();
    delay(1000);
    go_backwards();
  }
  if(distance_left < 30)
  {
    go_straight();
    delay(500);
    turn_right();
    delay(1000);
    go_backwards();
  }
}

//PID controller, we ended using just a PD controller
void wall_check(void)
{

  if(distanceCM_left < 45) //close to a left wall
  {
    error = (15 - distanceCM_left);
    prev_error = (15 - prev_left_distance);
    r_motor_speed = 190 + 3*(error);
    l_motor_speed = 230 - 3*(error); //kp = 3

    //move forwards with adjusted speeds

  analogWrite(right_drive_ccw_speed, r_motor_speed);
```

```
analogWrite(left_drive_cw_speed, l_motor_speed);
analogWrite(left_drive_ccw_speed, 0);
digitalWrite(left_drive_ccw_bjt, LOW);
digitalWrite(left_drive_cw_bjt, HIGH);

  if(distance_center > 30)
   {
    go_backwards();
    delay(500);
    turn_right();
    delay(2000);
    go_straight();
   }


   prev_left_distance = distanceCM_left;
 }
 if(distanceCM_right < 45) //close to a rght wall
 {
  error = (15 - distanceCM_right);
  prev_error = 15-prev_right_distance;

  r_motor_speed = 190 - 3*(error);
  l_motor_speed = 230 + 3*(error); //kp = 1.25, kd = 3

 //move forwards with adjusted speeds

 analogWrite(right_drive_ccw_speed, r_motor_speed);
 analogWrite(left_drive_cw_speed, l_motor_speed);
 analogWrite(left_drive_ccw_speed, 0);
 digitalWrite(left_drive_ccw_bjt, LOW);
 digitalWrite(left_drive_cw_bjt, HIGH);

  while(distance_center > 30)
   {
    turn_all_off();

    delay(500);
    turn_left();
```

```
      delay(2000);
      go_straight();
    }

    prev_right_distance = distanceCM_right;

  }

   //if not close to a wall, "roam" basically hope that it moves straight, but if it veers off it
will gravitate towards a wall

  if(distance_center > 25)
    {
      turn_all_off();
      delay(1000);
      if(turn == 1)
      {
        turn_right();
        delay(2000);
        go_straight();
      }
      if(turn == 0)
      {
        turn_left();
        delay(2000);
        go_straight();
      }
      if(turn == 1)
      {
        turn = 0; //set next turn to left after a right
      }
      if(turn == 0)
      {
        turn = 1;
      }


    }
```

```
}



void Standby(void)
{
  turn_right_off();
  turn_left_off();
  analogWrite(pump_motor, 0);
  analogWrite(brush_speed, 0);
  digitalWrite(buzzer, HIGH);
  delay(5000);
}

//reading ultrasonic sensors: returns a distance in centimeters
bool distance_center(void)
{

   digitalWrite(trigger_center, LOW);
   delayMicroseconds(2);
   digitalWrite(trigger_center, HIGH);
   delayMicroseconds(10);
   digitalWrite(trigger_center, LOW);
   duration_center = pulseIn(echo_center, HIGH);
   distanceCM_center = (duration_center / 2)*0.0344;

   if(distanceCM_center<= 15){ //15 cm away
    return true;
  }
  else{
    return false;

}
}

bool distance_right(void)
{

   digitalWrite(trigger_right, LOW);
```

```
  delayMicroseconds(2);
  digitalWrite(trigger_right, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger_right, LOW);
  duration_right = pulseIn(echo_right, HIGH);
  distanceCM_right = (duration_right/2)*0.0344;
  if(distanceCM_right<= 15){ //15 cm away
   return true;
 }
 else{
   return false;

}
}

bool distance_left(void)
{

  digitalWrite(trigger_left, LOW);
  delayMicroseconds(2);
  digitalWrite(trigger_left, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger_left, LOW);
  duration_left = pulseIn(echo_left, HIGH);
  distanceCM_left = (duration_left/2)*0.0344;

  if(distanceCM_left<= 15){ //15 cm away
   return true;
 }
 else{
   return false;

}
}
//EXTERNAL COMMS CODE
bool read_water(void)
{
 water_level = analogRead(h20_sensor);
 if(water_level <= 10) //if tank empty, number that it is empty at yet to be determined
 {
```

```
    digitalWrite(liquid_tank,HIGH);
    digitalWrite(buzzer, HIGH);


    // flash led/buzzer on off for 1 second
  }
  if(water_level>=100)
  {
    digitalWrite(liquid_tank,LOW);
    return true;
  }
  else
  {
    return false;
  }
}
bool read_battery(void)
{
  //~186 = 13V
  //135 = 9.75V (75%)
  //90 = 6.5V (50%)
  //45 = 3.25V (25%)
  battery_voltage = analogRead(A1);
  if(battery_voltage >= 135) //if battery voltage greater than 75%
  {
    //turn on the leds you want, flash them, etc..
    digitalWrite(battery_100, HIGH);
    digitalWrite(battery_75, HIGH);
    digitalWrite(battery_50, HIGH);
    digitalWrite(battery_25, HIGH);
  }
  else if(battery_voltage >= 90) //if battery voltage greater than 50%
  {
    //turn on the leds you want, flash them, etc..
    digitalWrite(battery_100, LOW);
    digitalWrite(battery_75, HIGH);
    digitalWrite(battery_50, HIGH);
    digitalWrite(battery_25, HIGH);
  }
  else if(battery_voltage >= 45) //if battery voltage greater than 25%
```

```
  {
    //turn on the leds you want, flash them, etc..
    digitalWrite(battery_100, LOW);
    digitalWrite(battery_75, LOW);
    digitalWrite(battery_50, HIGH);
    digitalWrite(battery_25, HIGH);
  }
  else if(battery_voltage >= 0) //if battery voltage greater than 0%
  {
    //turn on the leds you want, flash them, etc..
    digitalWrite(battery_100, LOW);
    digitalWrite(battery_75, LOW);
    digitalWrite(battery_50, LOW);
    digitalWrite(battery_25, HIGH);
  }

}

//MOTOR DRIVER CODE
//speeds = numbers in analogwrite functions

void turn_right_motor_forwards(void)
{
  //clockwise

  analogWrite(right_drive_cw_speed, 200);
  analogWrite(right_drive_ccw_speed, 0);
  digitalWrite(right_drive_ccw_bjt, LOW);
  digitalWrite(right_drive_cw_bjt, HIGH);

}

void turn_right_motor_backwards(void)
{
  //ccw
  analogWrite(right_drive_cw_speed, 0);
  analogWrite(right_drive_ccw_speed, 200);
  digitalWrite(right_drive_ccw_bjt, HIGH);
  digitalWrite(right_drive_cw_bjt, LOW);
}
```

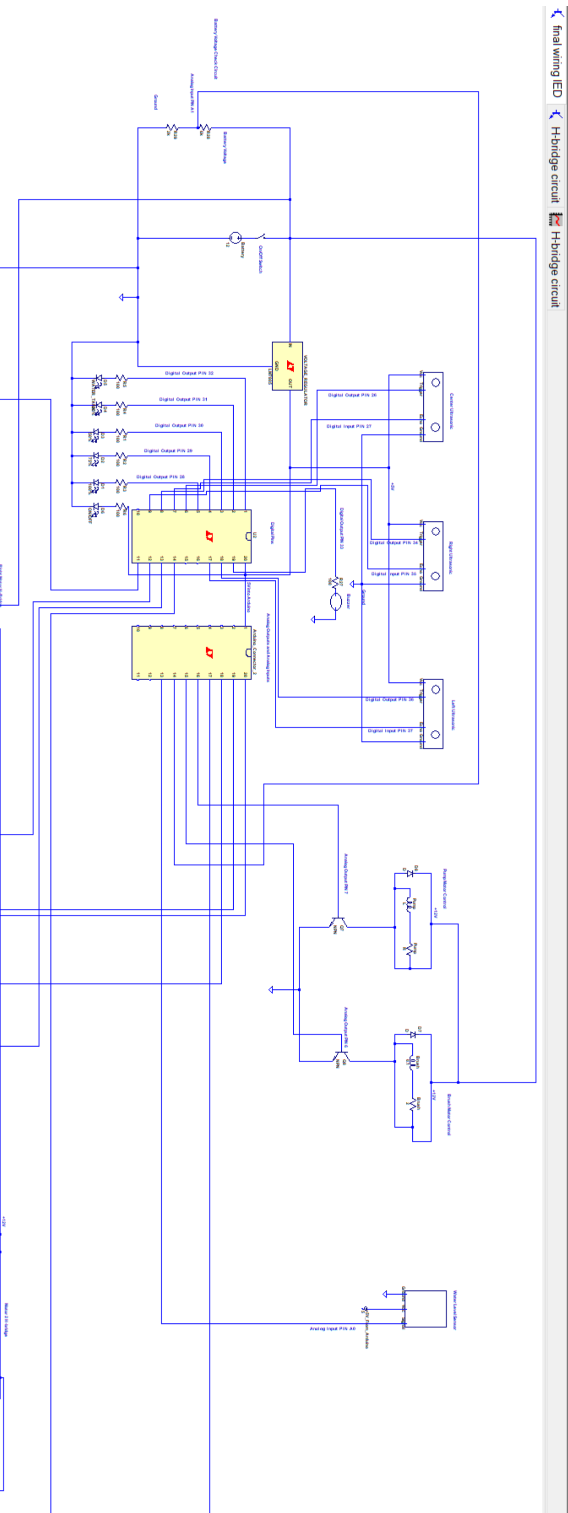
```
void turn_right_off(void)
{
  analogWrite(right_drive_cw_speed, 0);
  analogWrite(right_drive_ccw_speed, 0);
  digitalWrite(right_drive_ccw_bjt, LOW);
  digitalWrite(right_drive_cw_bjt, LOW);
}

void turn_left_motor_forwards(void)
{
  //clockwise
  analogWrite(left_drive_cw_speed, 130);
  analogWrite(left_drive_ccw_speed, 0);
  digitalWrite(left_drive_ccw_bjt, LOW);
  digitalWrite(left_drive_cw_bjt, HIGH);
}
void turn_left_motor_backwards(void)
{
  //counterclockwise
  analogWrite(left_drive_cw_speed, 0);
  analogWrite(left_drive_ccw_speed, 230);
  digitalWrite(left_drive_ccw_bjt, HIGH);
  digitalWrite(left_drive_cw_bjt, LOW);
}
void turn_left_off(void)
{
   analogWrite(left_drive_cw_speed, 0);
  analogWrite(left_drive_ccw_speed, 0);
  digitalWrite(left_drive_ccw_bjt, LOW);
  digitalWrite(left_drive_cw_bjt, LOW);
}
void turn_left(void)
{
  turn_right_off();
  turn_left_motor_forwards();
}
void turn_right(void)
{
  turn_left_off();
```

```
  turn_right_motor_forwards();
}
void go_straight(void)
{
  turn_left_motor_forwards();
  turn_right_motor_forwards();
}
void go_backwards(void)
{
  turn_left_motor_backwards();
  turn_right_motor_backwards();
}
void turn_all_off(void)
{
  turn_left_off();
  turn_right_off();
  analogWrite(brush_speed, 0);
  analogWrite(pump_motor, 0);
}
void turn_brush(void)
{
  analogWrite(brush_speed, 220);
}
//function turns the pump on every 3 seconds periodically
void pump_water(void)
{
    unsigned long timer_two;
    unsigned long timer_one;
    timer_one = millis();
    timer_two = millis();
    analogWrite(pump_motor, 0);
    while(timer_two < timer_one + 3000){
      analogWrite(pump_motor, 200);
      timer_two = millis();
    }

}
```

Appendix K: Final wiring schematic


final wiring IED
H-bridge circuit
H-bridge circuit

Appendix L: Gear Ratio of Cleaning System

In the cleaning system gears are used to transfer power from the motors to the rotating sponge in order for it to clean. Sometimes, a motor itself can be powerful enough for an application, but the motor's maximum torque may not be high enough for the application. A motor that is rotating very fast may not have a lot of torque, which was required to scrub the floor efficiently. In regards to motors, a tradeoff is made between motor speed and torque. As one increases, the other decreases. In order for the cleaning system to be functional, the speeds and torques of the motor need to be analyzed using the following methods:

In most simple term, the formula will be the following:

Torque = F $\times r\ (N*m)$

F:Force Vector (N)

r: is the position vector (m) (a vector from the origin of the coordinate system defined to the point where the force is applied)

But in this mechanical system, the following equation can be used to calculate the amount of torque that's is generated by the motor.

*τ = (I * V * E *60) / (rpm * 2π)*

I: Current (amperes)

V: Voltage (volts)

E: Motor Efficiency (unitless)

The voltage maximum was found to be 9V, and the current was found to be 0.83 amperes. The motor efficiency was approximately 0.701, and the nominal rpm for the motor was 3,855 rpm (nominal rpm and efficiency found from datasheets for the specific motor used in the design).

The maximum torque produced is then: $(0.83 * 9 * 0.701 * 60)/(3{,}855 * 2\pi) = 0.01297\ N*m$

Since a timing pulley set is used, the gear ratio must be taken into consideration, which could affect the total amount of torque that the mechanical system can produce. The following function can be used to calculate the exact torque:

Motor Torque x Gear Ratio = Torque At the Wheel

Since the input gear(which is attached at the motors) has less teeth than the output gear(which is attached to the sponge via metal rod), more torque will be generated to turn the metal rod, which turns the sponge. This increased torque means that the sponge can scrub more efficiently.

The input gear has 14 teeth, and the output gear has 60 teeth, which results in a gear ratio of 4.285. This gear ratio effectively increases the motor torque from 0.01297

N*m to 0.05558 N*m, which adds enough torque to insure the scrubbing mechanism works.