JAK-HD

Jovanni Shen, Aleida Diaz-Roque, Karan Humpal

ECE118/218 Intro to Mechatronics

Professor Gabriel Hugh Elkaim

UCSC - Spring 2024

Date: 06/07/2024

Table of Contents

Introduction and Project Overview

Image 1: UCSC 2024 Mechatronics Poster

As a dedicated team of engineering students from the University of California, Santa Cruz, we successfully completed the ECE-118/L Mechatronics final project. This project was a testament to our skills and creativity, as we designed and built an autonomous robot capable of efficiently performing specific tasks on a playing field. This experience allowed us to demonstrate the practical application of our mechatronics education, showcasing our technical abilities in a challenging and competitive environment.

Our team successfully developed an 11”x11”x11” autonomous robot, which competed in a game where the objective was to navigate a standardized 4'x8' playing field, locate, trap, and dispense 25mm chrome balls. The challenge was to store these balls within our robot and push at least two of them through a one-way door into the opponent's field.

Over the course of five weeks, we designed, implemented, tested, and iterated our robot to ensure it met the project's rigorous requirements. The playing field, marked with 2” black tape boundaries and a low wall separating the two sides, included two towers that discharged the chrome balls at a rate of 1 ball every 2 seconds from alternating towers. Our robot was tasked with having to send or collect a total of at least 30 balls out of the 40 that were dispensed. On the field was also an 11”x11”x11” “Dead-Bot”, which served as an obstacle for our bot. We had to resolve any collision with either the wall, tape, or obstacle within 5 seconds in order to not be disqualified.

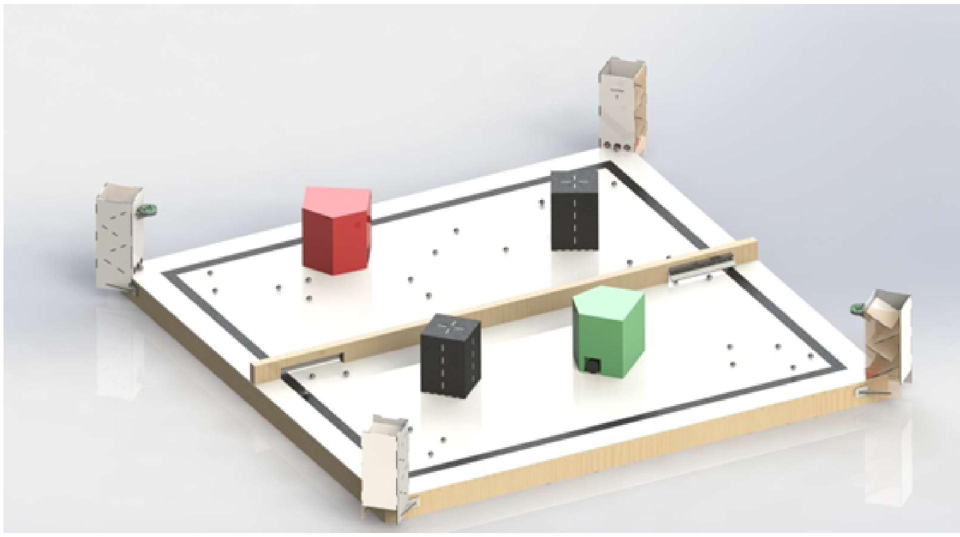

Image 2: Playing Field

Our primary objective was to clean our field more effectively than our opponents by the end of each two-minute round. Points were awarded based on the number of balls removed from our field, emphasizing the importance of precise navigation and efficient ball handling. The project involved multiple design reviews, weekly check-offs, and culminated in a public tournament where we demonstrated our robot's capabilities.

This project was a culmination of our learning in ECE-118, requiring us to apply our knowledge in electronics, mechanics, and programming. It challenged us to think creatively, work collaboratively, and manage our time effectively. We were motivated by the opportunity to innovate and compete, successfully building a robust and effective robot that exemplified the skills we developed throughout our coursework.

Background

Our team successfully completed a series of labs that provided the essential background needed for our final project during the 5 weeks of the course. These labs laid the groundwork for the development and implementation of our autonomous robot.

Lab 0: The ES_Framework and State Machines

In Lab 0, we were introduced to the ES_Framework, which is fundamental for implementing state machines in our project. This lab involved programming a two-wheeled robot, the "Roach," and learning the basics of event checkers, test harnesses, finite state machines (FSM), and hierarchical state machines (HSM). By soldering a small PCB and coding the robot's behavior, we gained practical experience with the ES_Framework, which proved invaluable when designing the state machine for our final robot.

Lab 1: Building a Filter

Lab 1 focused on constructing an active bandpass filter using op amps. The goal of the lab was to allow a frequency of 2kHz but completely attenuate frequencies below 1.5kHz and above 2.5kHz. This was done on a breadboard and included stages of amplification without saturation, a peak detector circuit, and a comparator. This lab aimed to enhance our understanding of signal processing and filtering techniques. Although we initially intended to use this filter for detecting the 2kHz beacon from one of the towers in our final project, we ultimately decided against it. Nonetheless, the skills and knowledge acquired from this lab were crucial in developing our robot's sensing capabilities.

Lab 2: Mechanical Prototyping and SolidWorks

Lab 2 provided us with hands-on experience in mechanical prototyping and the use of SolidWorks for designing parts. We learned to use the LaserCutter, build a prototype robot chassis out of MDF and foamcore, and solder the 2kHz signal detector from Lab 1 onto a perf board. This lab emphasized the importance of precision in mechanical design and the practical application of SolidWorks for creating detailed components. The experience gained from this lab was critical in constructing the mechanical aspects of our robot, ensuring robust and reliable performance.

Lab 3: Motors and Motor Drivers

In Lab 3, we delved into the world of motors and motor drivers. This lab taught us how to drive various types of motors, an essential skill for our final project. We explored different motor types, control methods, and the integration of motor drivers with our robot. Understanding how to effectively control motors was key to developing a robot capable of precise movements and accurate navigation on the playing field.

Together, these labs equipped us with the necessary skills and knowledge to tackle the final project. By building on the foundations laid in these preliminary labs, we were able to design, implement, and refine an autonomous robot that met the project's rigorous requirements.

List of Materials

Materials	Quantity	From	Cost
MDF 16” x 24” Sheet	2	BELS	0
4” Stealth Wheels	2	Andy Mark	16.50
Nubs for Mounting wheels	2	Andy Mark	7
DC Brushless Motor	3	BELS	0
3D Printed Roller Mounts	2	Slug Works	0
Limit Switches	6	Amazon/BELS	8
H-Bridge (L298N)	2	BELS	0
IR Tape Sensor	6	Amazon	7
4” Linear Rod Actuator	1	DigiKey	27
9.9V Battery	1	BELS	0
Uno32 Board/ Power Distribution Board	1	BELS	0
Wires	N/A	BELS	0
Hot Glue	N/A	BELS	0
4-40 Screws/ Washers/ Nuts	N/A	BELS	0
Perfboards	3	BELS	0
Electrical Components	N/A	BELS	0
Ball Bearing	1	Amazon	5
Zip Ties	2	Home	0
Rubber Bands	20	Home	0
TOTAL			**70.5**

Mechanical Design Concept

The goal of our project was to design a robot capable of collecting and dispensing a total of 30 chrome balls, each 1 inch in diameter. The 4’ by 8’ field of play, with boundaries denoted by tape, included a wall separating the two sides, which was 3 inches tall, and an obstacle, represented by an 11x11x11 inch box. Given these parameters, we developed a comprehensive mechanical design using MDF (medium-density fiberboard) and adhered to the constraints of our available tools, primarily a laser cutter that could only cut 2D shapes.

Design Constraints and Materials

The primary material used for constructing our robot was MDF, which we selected for its strength and ease of laser cutting. The laser cutter's limitation to 2D shapes meant that we had to employ a tab and slot technique to assemble our parts. We used hot glue to secure the connections, ensuring a sturdy and reliable build.

Collection Mechanism

After careful consideration, we decided that the most efficient method for collecting the balls was a roller system, similar to the mechanisms used by golf range servants to pick up golf balls. The roller was powered by a high RPM DC motor, whose axle was press-fitted into a 3D printed piece. This piece housed a small shell that allowed a thin piece of MDF, with a section hollowed out to accommodate the motor axle, to fit seamlessly into another 3D printed piece at the opposite end. This second piece featured a boss that was press-fitted into a ball bearing, which was mounted concentrically with the motor axle on the other side, allowing for smooth rotation.

Both 3D printed pieces had 12 indents around their edges to facilitate the placement of rubber bands, completing the roller. To collect balls, the roller spun clockwise, and to dispense them, it spun counterclockwise.

Storage and Dispensing Mechanism

For storage, we designated the entire bottom floor of the robot. A small ramp, made from thin cutting board material, was angled towards the roller. As balls were collected, they were flung up the ramp, which angled down towards the roller. This design ensured that, in collect mode (clockwise spin), the balls remained in storage. To dispense the balls, the motor's polarity was reversed, causing the roller to spin counterclockwise and eject the balls and due to the angle of the ramp, the balls were always pushing against the roller.

Wheel and Drive System

We used relatively large wheels, 4 inches in diameter, to ensure that the motors did not interfere with the storage space on the first floor. The wheels were directly driven by DC motors, which were mounted on the first floor. This arrangement provided ample space for the balls to fit without obstruction.

Slot Door Mechanism

Opening the slot door to dispense balls required a decent amount of force and a mechanism to keep it open. We considered several options, including a servo-controlled drop-down bar, but ultimately chose a linear rod actuator for its reliability and simplicity. The actuator had a 4-inch stroke and was mounted at an angle to achieve the required extension while minimizing operation time.

Collision Detection

To detect collisions with walls or "dead bots," we used limit switches. We designed bumpers with two levels at the front of the robot—one at the height of the wall and one taller and extending farther out than the lower level (to detect obstacle collision). Each bumper had two limit switches to determine the direction of impact (left, right, or head-on). A similar bumper was added to the back of the robot.

Electronics Mounting

The second floor of the robot was designed with precise screw holes to mount the electronic perf boards and the Uno32 microcontroller stack. This ensured that all electronic components were securely positioned and easily accessible for maintenance.

IR Sensors

Our robot was equipped with six IR sensors. Four sensors faced the ground (front right, front left, back right, and back left) to detect the tape on the floor, while two sensors were mounted on the sides, each facing outward to detect the wall.

In summary, our mechanical design integrated robust materials and precise assembly techniques to create an efficient and reliable robot capable of collecting and dispensing chrome balls while navigating the field and avoiding obstacles.

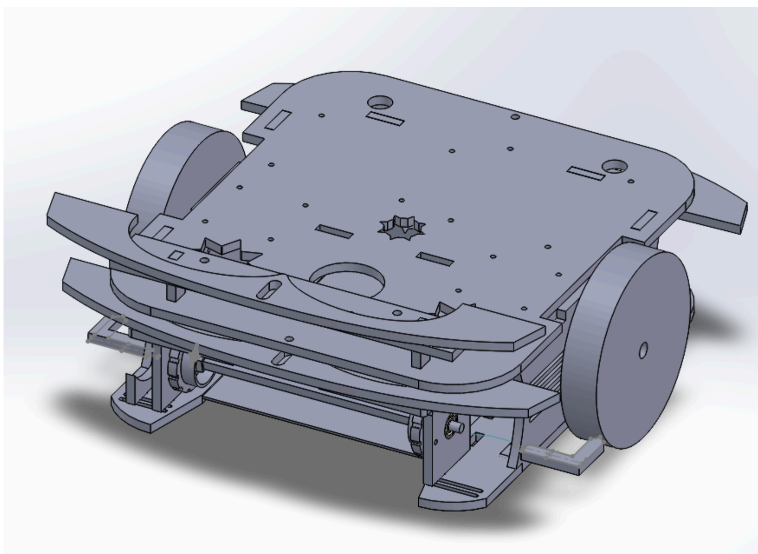

Image 3: Isometric Front View of the Autonomous Robot

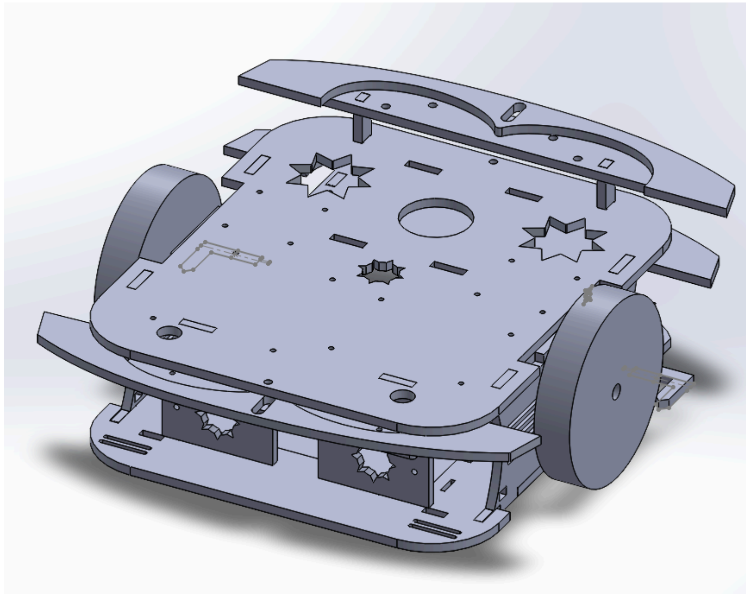

Image 4: Isometric Back View of the Autonomous Robot

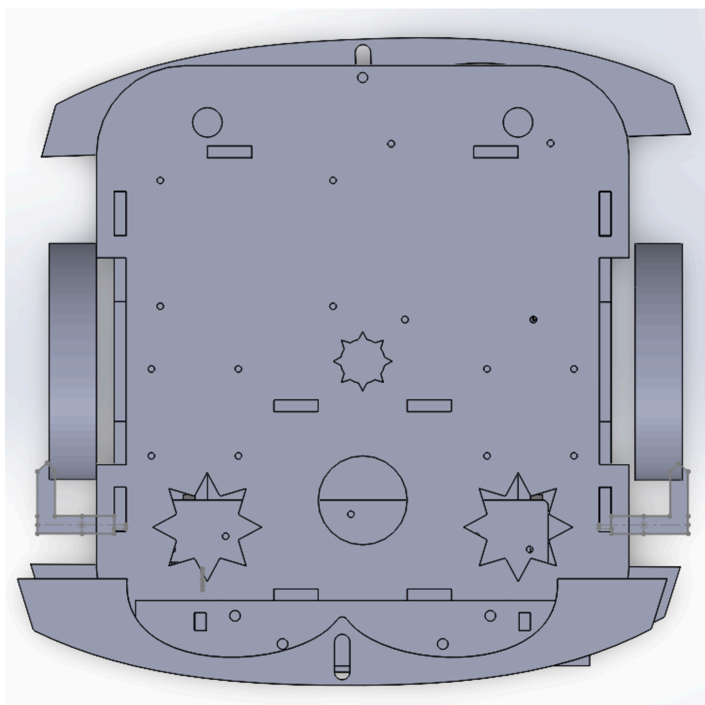

Image 6: Top View of the Autonomous Robot

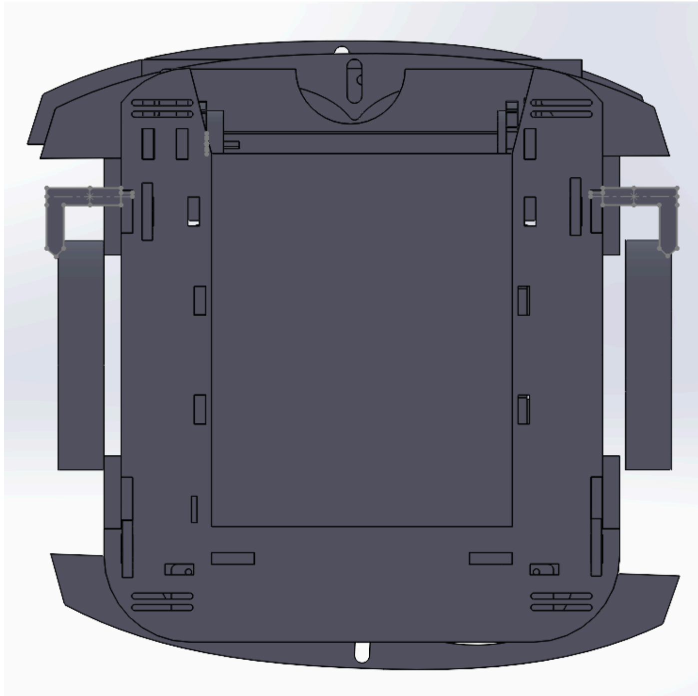

Image 7: Bottom View of the Autonomous Robot

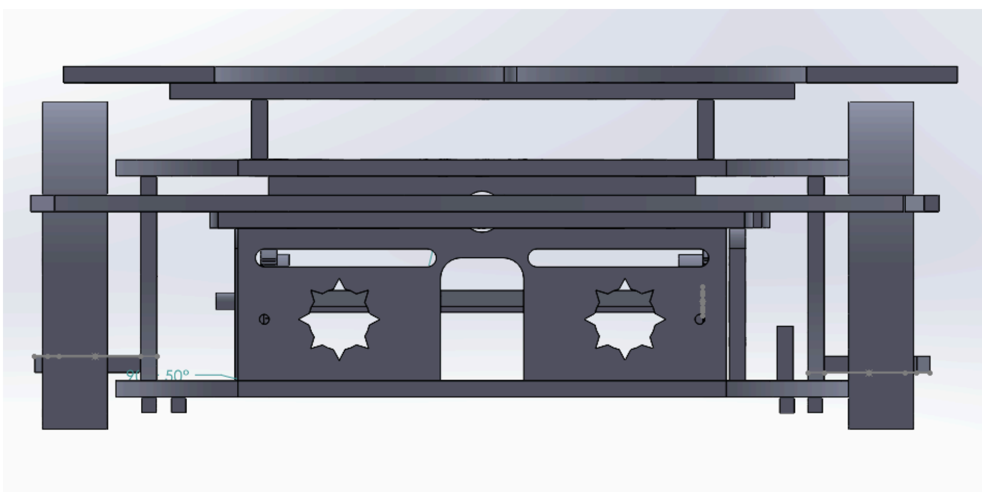

Image 8: Back View of the Autonomous Robot

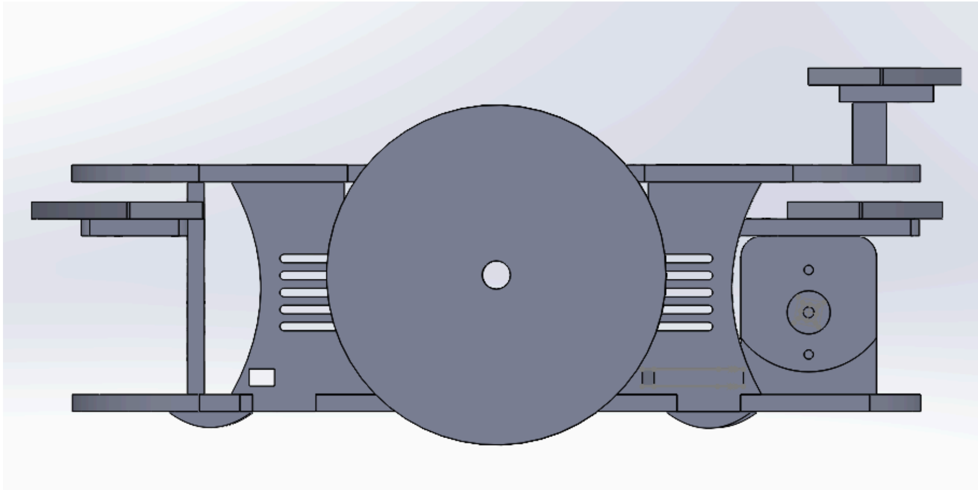
Image 9: Side View of the Autonomous Robot

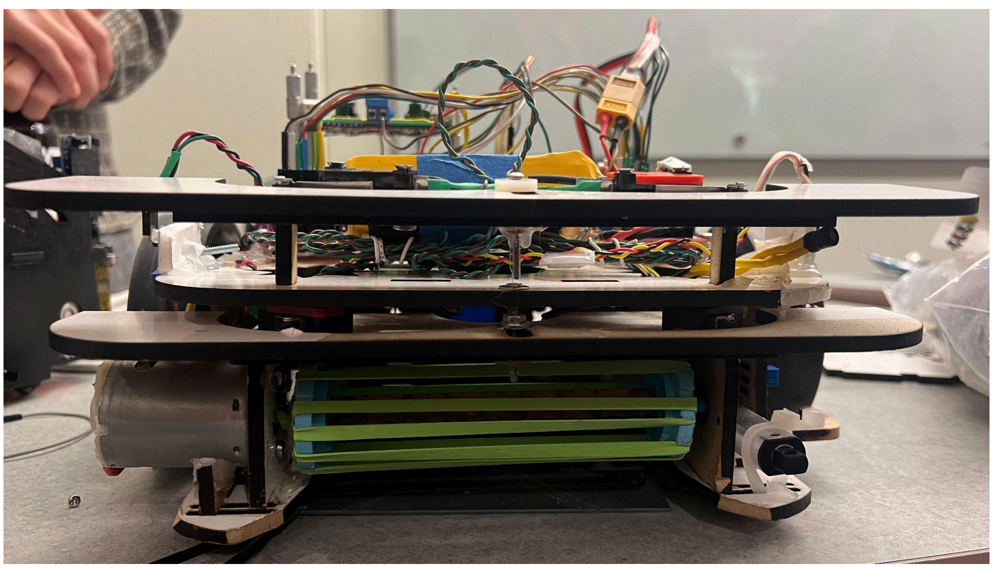
Image 10: Front View

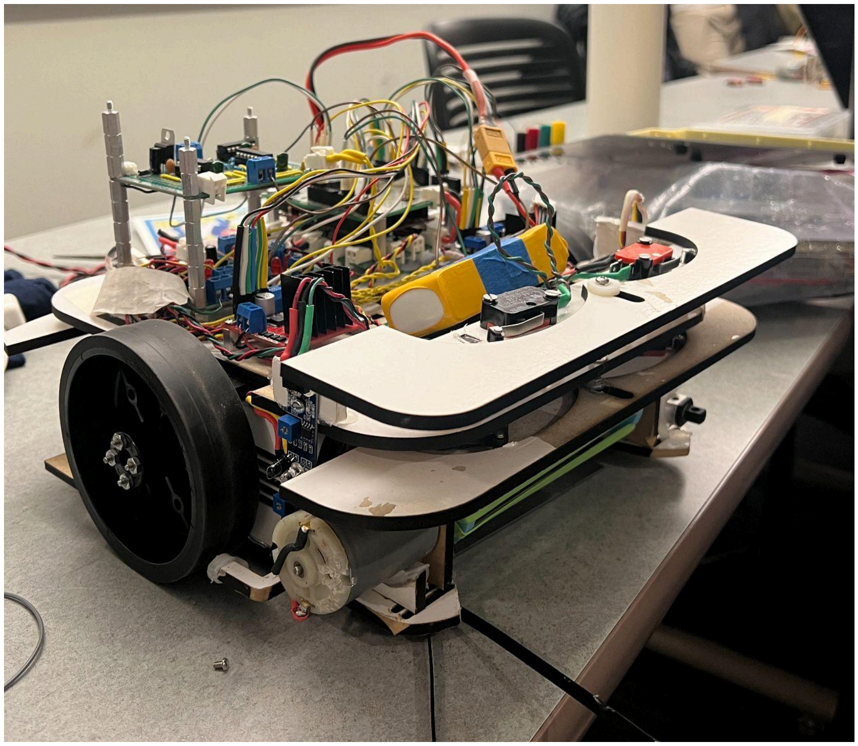

Image 11: Isometric View

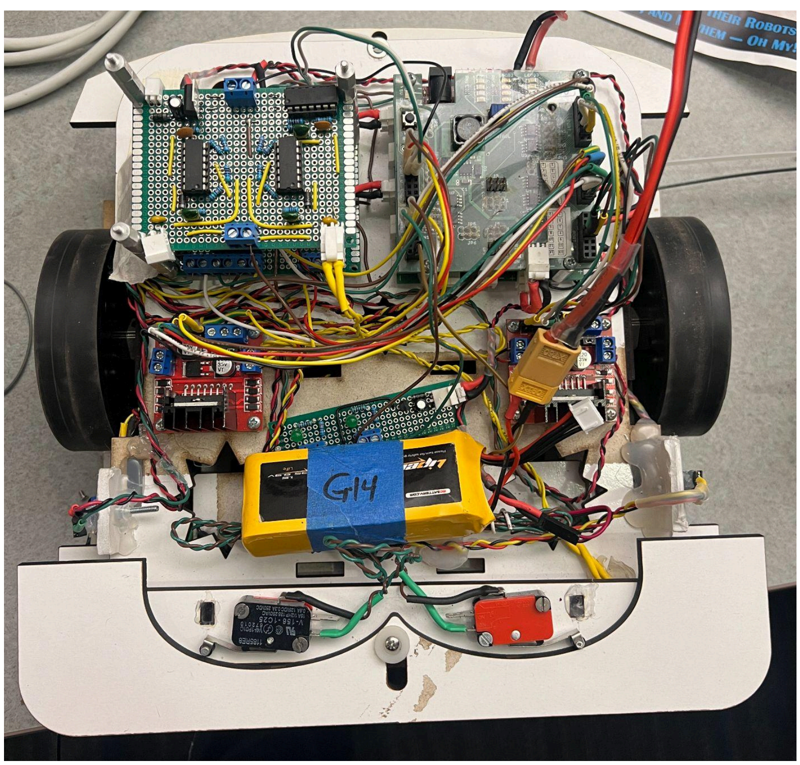

Image 12: Top View

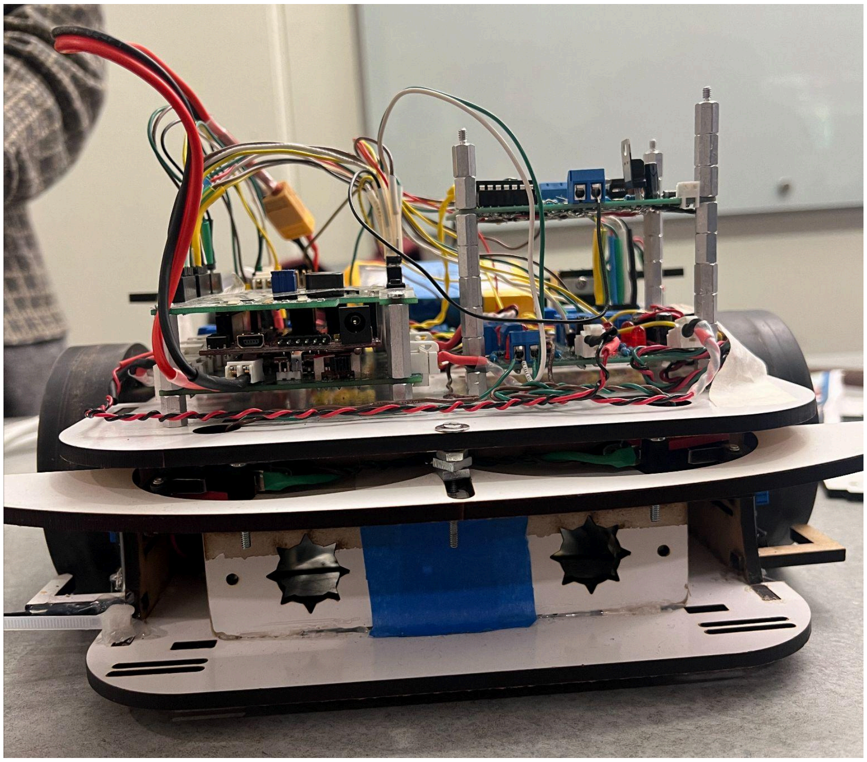

Image 13: Back View

Electrical Schematics

When it came to the electrical needs of our robot, the main things that we needed to design were the sensors used to help the robot navigate throughout the field and deposit the balls effectively. The sensors that we used to accomplish this were the IR tape sensors, track wire sensor and bump switches. Originally we also planned to use the beacon detector to help our robot orient itself with the field; however, after further planning and testing we decided to not utilize the beacon detector in order to simplify our overall searching algorithm and decrease our overall chance for failure. Furthermore, the IR tape sensors that we ended up using were purchased online where all we needed to do was wire in 3.3V, GND and have a signal wire going out into our microcontroller; these tape sensors also came with a potentiometer on them to tune the threshold in which they would output a high or a low.

Track Wire Detector

The first circuit that we set out to implement was the track wire detector. The track wire was a wire that had a 25kHz signal oscillating throughout it, in order to detect this signal we used a tank circuit to electromagnetically couple to the track wire itself and provide us a signal without any physical contact to the wire.

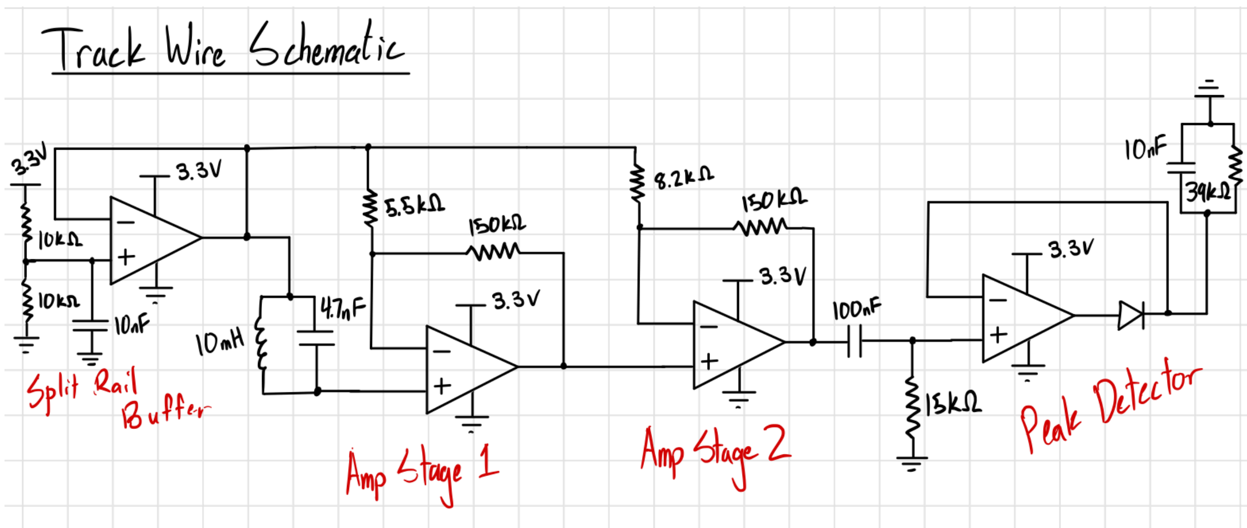

Image 14: Track Wire Schematic

The tank circuit was simply an inductor in parallel with a capacitor, we used a 10mH inductor and a 4.7nF capacitor to line up with the 25kHz signal that we were trying to receive. Other than that part of the circuit we simply just amplified the signal and provided some DC offset to it with our split rail buffer. After the amplification we sent the signal into a peak detector to rectify it into a DC signal rather than a sinusoidal signal which our microcontroller wouldn't be able to read. In this circuit we didn't implement a comparator because we planned to feed an analog signal into our microcontroller and manage the thresholds and hysterisis through software rather than hardware. We planned to do this because we found that the different track wires in different rooms were at slightly different frequencies which made the response of our detection circuit different depending on which wire it was trying to sense. Because of this inconsistency we decided that it would be best to deal with all of the bounds through software since we would be able to change the bounds much quicker and be able to adapt to different scenarios.

Since we had the overall design from Lab 1 already, we could get started by prototyping the circuit on the breadboard first. This process was just putting down what we already had from Lab 1 and then changing and adjusting the resistor and capacitor values to suit what we were trying to do. Since we wanted to have more room to adjust and be safe we increased the gain of our circuit so that we could detect the trackwire from roughly 5-6 inches away. Once we had the circuit tuned to what we were looking for we could then go to implement it onto the perfboard for the final design.

On the perfboard we simply had to just transfer the circuit that was already designed and proven from the breadboard onto a soldered perfboard. When doing this it was important to make each section of the circuit incrementally and test after making each section to make sure that there was no mess ups that would break the entire circuit after we had already spent a lot of time soldering the entire thing.

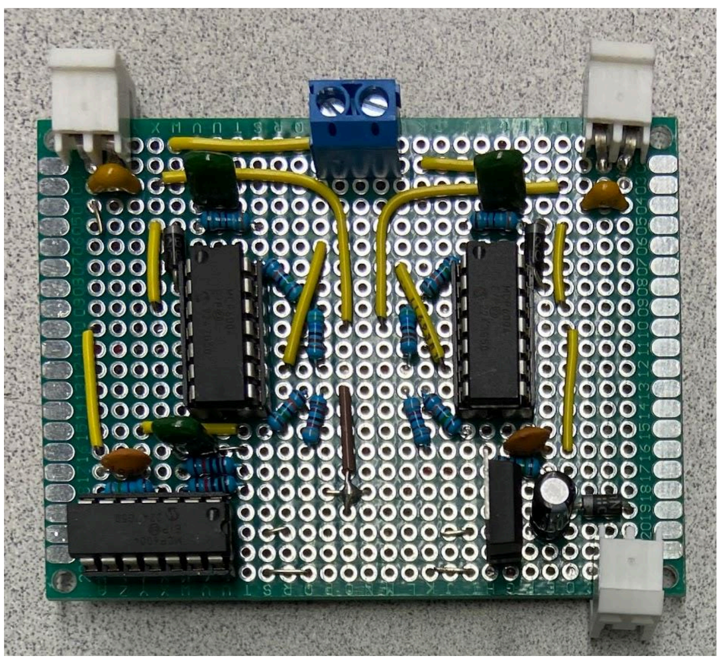

Image 15: Track Wire on Perfboard

Our final circuit contained two different track wire circuits because we originally planned to use two different track wires so that we could ensure that our bot was properly lined up with the track wire and not far off to one side or the other. However, the right track wire that we were planning on implementing ended up having way too much noise interference from the DC motor that we used for the roller to provide us a readable signal. There was a large amount of noise from the motor since DC motors are run using inductive coils and the inductive coils would couple to the inductor on our tank circuit, which would provide us a very noisy signal if our inductor was within 6 inches of the DC motor. Because of this we ended up just not using the right track wire on our robot and relying on our tape following to ensure that we were lined up with the trap door instead.

Bump Sensor and Tape Detectors

For our bump sensors and tape detectors the circuits were much more simple since we just needed to give them power and ground and have a signal wire coming out. However, we decided to also solder a board for these with indicator LEDs on them so we could see in one place whenever each of the switches or tape sensors are being actuated. This board also held the voltage regulators for all of our bump switches and tape detectors so that we could run the power and ground all from one place rather than having everything come off of the power distribution board. Our power delivery for our entire robot is something that we will cover more extensively later. To wire all of these indicator LEDs the overall circuit was really simple since we just needed an LED from our signal node connected to a pullup resistor going to 3.3V. This way whenever our signal going into the microcontroller would change so would the LED, showing us whether or not our sensors were properly actuating. The only challenge that we encountered when creating this circuit was having to find a pullup resistor that wouldn't affect the voltage of the signal wire too much. If we had too small of a pullup resistor the low voltage sent from the signal wire wouldn't cross the low voltage threshold of our microcontroller. This was a problem because even though our LED would turn on and off and the voltage from the signal wire would be changing, the microcontroller wouldn't register a change since the voltage was still too high for the digital input thresholds.

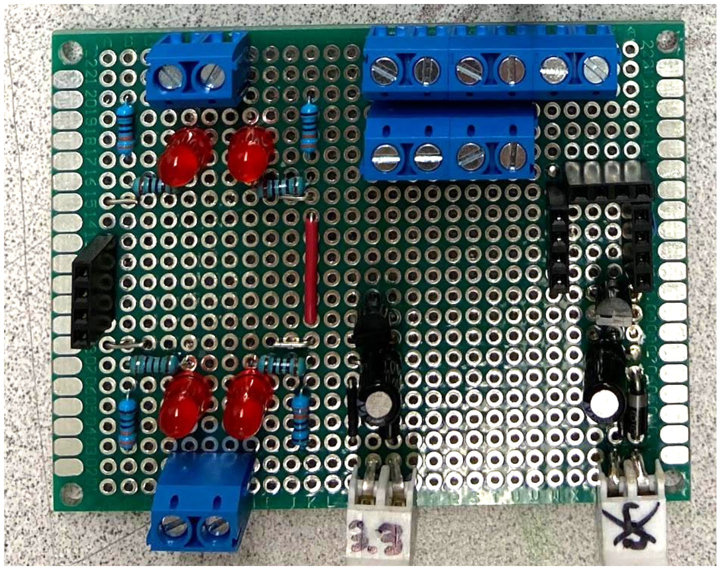

Image 16: Tape Sensor and Bump Switch Power Board

On each of these circuits we had a power regulator for each of them to regulate the voltage down to 3.3V since that's what the op-amps use and that's the maximum input voltage of our microcontroller as well. The voltage regulators would take the 10V input from the power distribution board which was powered by our 10V battery. All the power distribution board did was basically take the battery voltage and separate it into different outputs that were all protected by fuses so we don't burn anything out. Each of our power regulators also had a diode on the input for reverse polarity protection and a capacitor going from the output voltage to the input voltage to smoothen the power delivery and reduce noise.

Throughout this lab we had a lot of issues with power delivery since we had trouble with our power distribution board not properly distributing power. On top of that we also had unsecure connectors that would connect our battery to the power distribution which caused us a lot of problems as well, especially with collisions. To fix these issues we learned that we needed to balance the power draw on the voltage distribution board so that we aren't trying to source too much current from a single side and try to keep it as balanced as possible on both sides of the board. We also remade the connector for our battery so that it was more properly secured to our distribution board and the connector wasn't half falling out. These changes made a big difference and solved a lot of the issues that we had when it came to power delivery on our robot.

Electrical Mapping

Another big help when it came to the electrical components of our robot was having a clear and detailed wiring diagram of what wires went where. Many times we had to change something with the physical design of our robot which led us to take apart our robot and put it back together from scratch. Having this wiring diagram made it a lot easier for us to reassemble and rewire our robot since we had a single neat document that showed us exactly what went where.

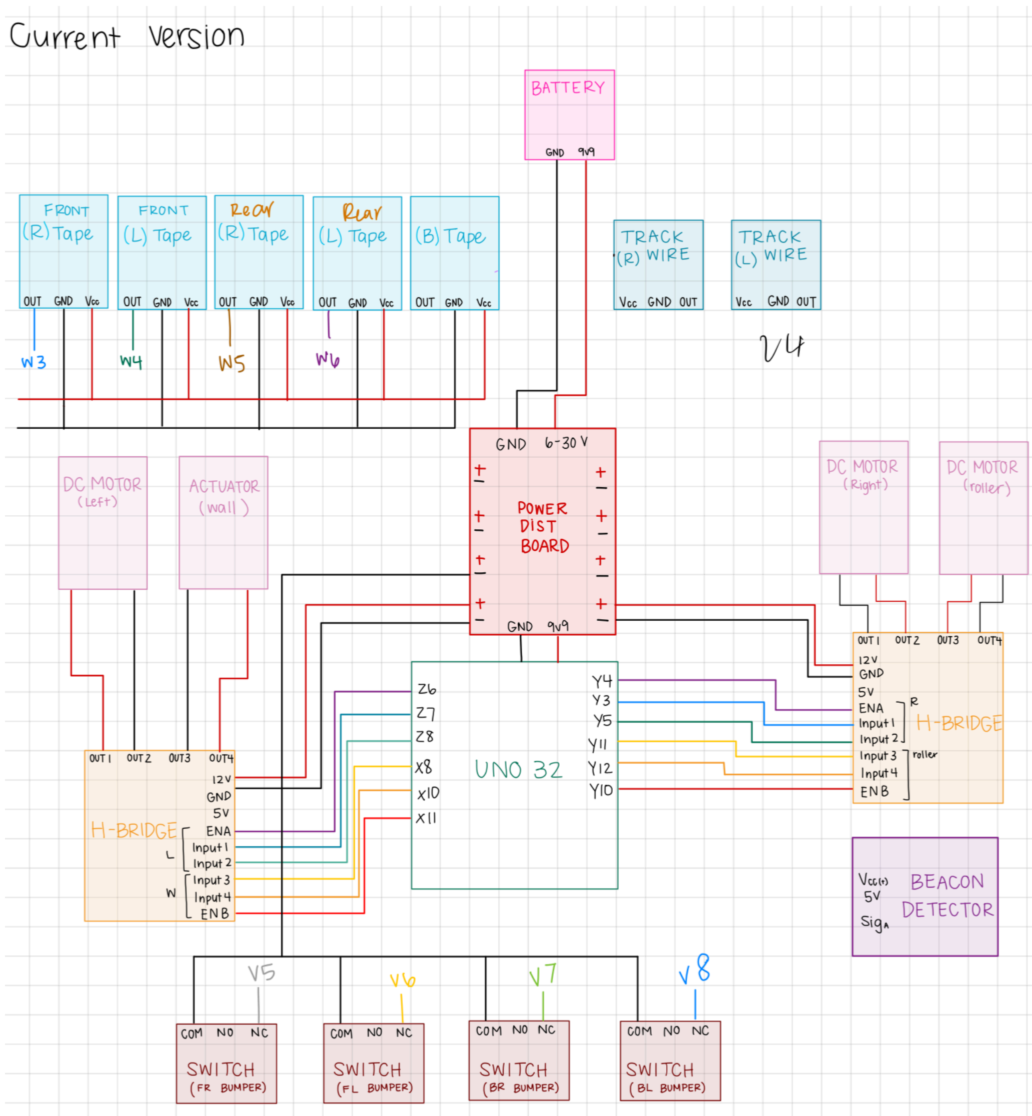

Image 17: Robot Component Pinout

To make all of these connections that are shown in this wiring diagram we originally just used the pre-crimped jumper wires that we purchased. However, these wires would easily pop out of the ports and sometimes didn't create the most solid connections between our microcontroller and our circuits. Eventually, after we were having problems with loose connections causing false signals we switched to making our own connections with the male to male headers that were provided to us in our lab kits. These connectors allowed us to make neat ribbons that would populate an entire port of our microcontroller, rather than having to plug each individual wire into the correct pin tediously. Not only did these connectors make it much easier to wire our robot, but they also were a lot more secure as well and we didn't have any more issues with wires unplugging and false signals being sent.

Voltage Regulator

As our project evolved there were parts of the electrical design that we didn't end up using in our final design. Two of the circuits that we didn't keep the whole way through were the 5V regulator for our servo that we iterated away from and the beacon detector. We had a separate 5V regulator circuit for our servo when we initially were planning on using a servo to push the balls out of our storage compartment; however, when we switched from using a servo to using a ramp we no longer had a need for that regulator. The bigger circuit that we ended up ditching was the beacon detector, originally we were planning on using a beacon detector that would detect the 2kHz IR transmission from one of the towers that would drop the balls; however, due to simplicity and doubts in reliability we decided to use tape and wall detection as our method for field orientation instead. We decided to make this change after we had already designed, tested and created our final beacon detector and added its event into our software's service routine.

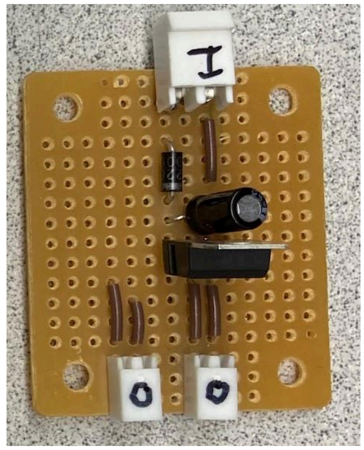

Image 18: 5V Voltage Regulator

Beacon Detector

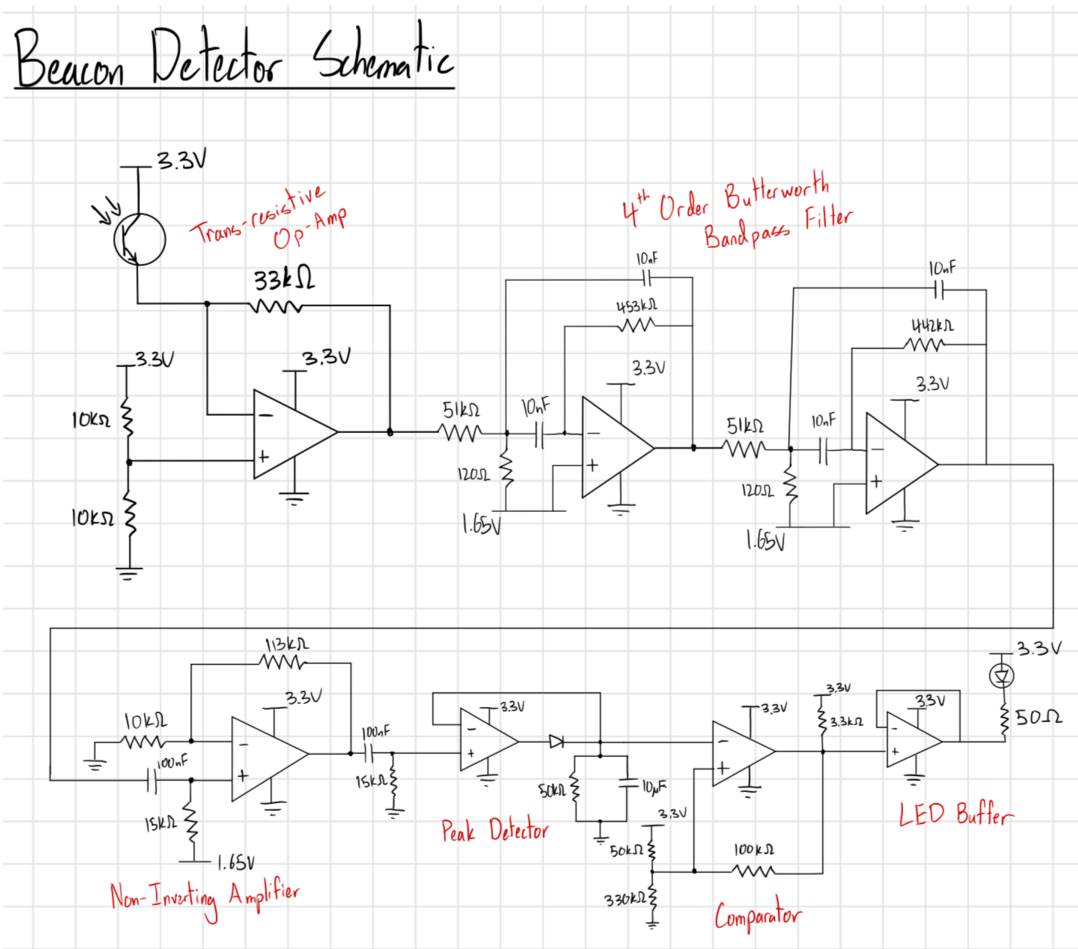

Image 19: Beacon Detector Schematic

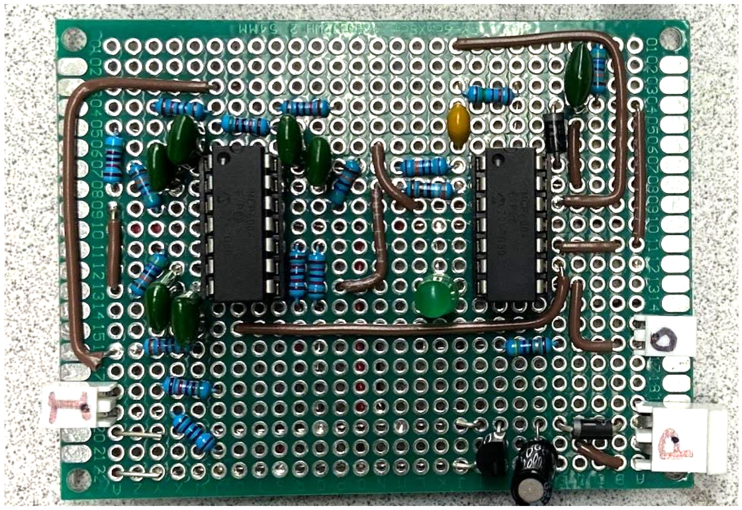

Image 20: Beacon Detector on Perfboard

With the amount of problems that we were already facing and our time constraints, keeping as many aspects of our robot as simple as possible was the best thing for us to do. We decided to work towards having the essential parts of our robot work as reliably as possible rather than have a lot of parts to our robot that worked more sporadically. This turned out to be a good strategy for us especially when it came to the electrical design where everything is extremely difficult to troubleshoot. Overall, when it came to electrical design we ran into a lot of problems with power delivery throughout our robot and tried to keep everything as simple and reliable as possible.

Software Implementation / Search Strategy

Once the essential navigation elements of the robot were in place we began implementing a library which controlled the navigation of all the motors in the robot. These included the two motors controlling the wheels, the motor controlling the roller which was our intake mechanism and the motor controlling the linear rod actuator used to open the trap door. The library also included a test harness so that we could periodically tune the behavior of the motors independently. The motos library initialized the registers used to control the behavior as well as provide clear and easy to use functions like moveMotor(motor).

After thoroughly testing all the motors we created another library for the sensors used in the robot. These included six IR sensors used to detect the black tape on the field, six limit switches used in the bumpers, and a track wire sensor. The sensors library was designed to initialize and read the various sensors used by configuring the Analog-to-Digital (AD) pins and set the TRIS registers for the digital I/O ports on the UNO32 as well as provide functions to read the values from these sensors.

*** EVENTS AND SERVICES

With the motors and sensor libraries in place, we needed a robust method to manage the robot's behavior to ensure a smooth transition between different actions and implement a strategy to meet all the required specifications. A hierarchical state machine allowed us to define a clear and structured way to handle various operational modes of the robot, such as searching for spheres, navigating obstacles and performing specific tasks.

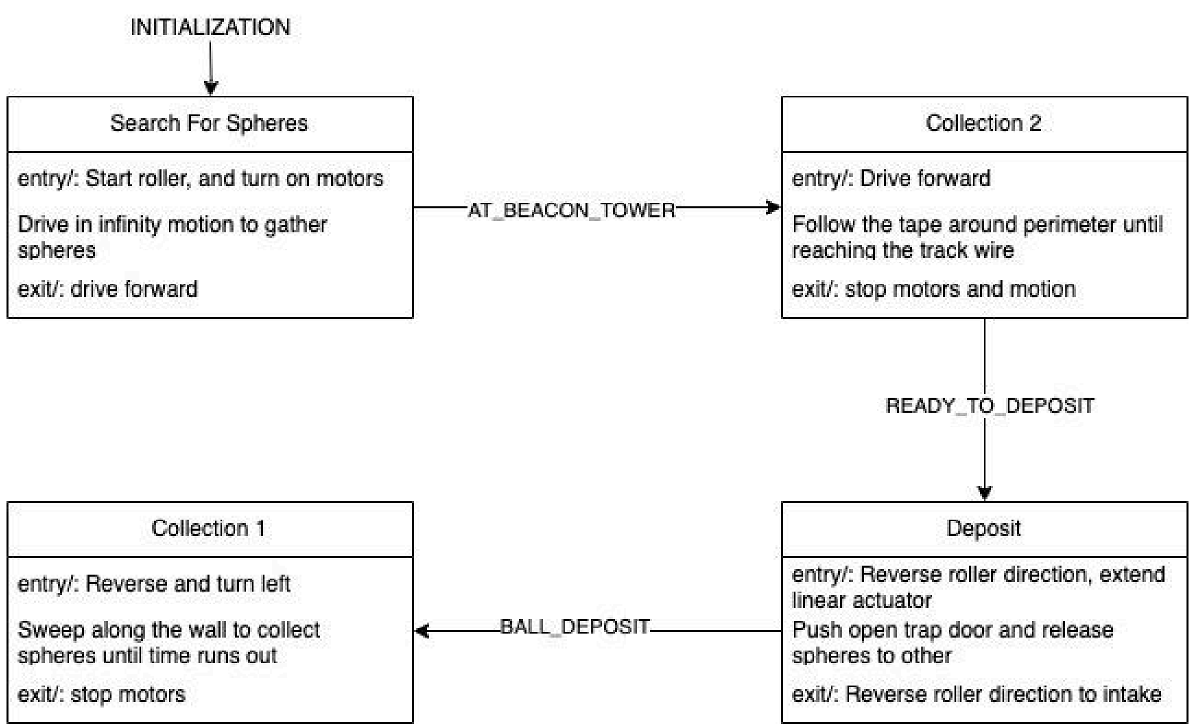

Image 21: Top Level Hierarchical State Machine

By implementing a hierarchical state machine (HSM), we could break down the robot's overall behavior into manageable subtasks, each represented by specific states. These states would handle different aspects of the robot's operation, ensuring that it responded appropriately to sensor inputs and other events reliably. For instance, the state machine can manage the transition from searching for spheres to driving along the perimeter one tape is detected and then parking to perform the deposit and afterwards driving along the wall to continue collecting.

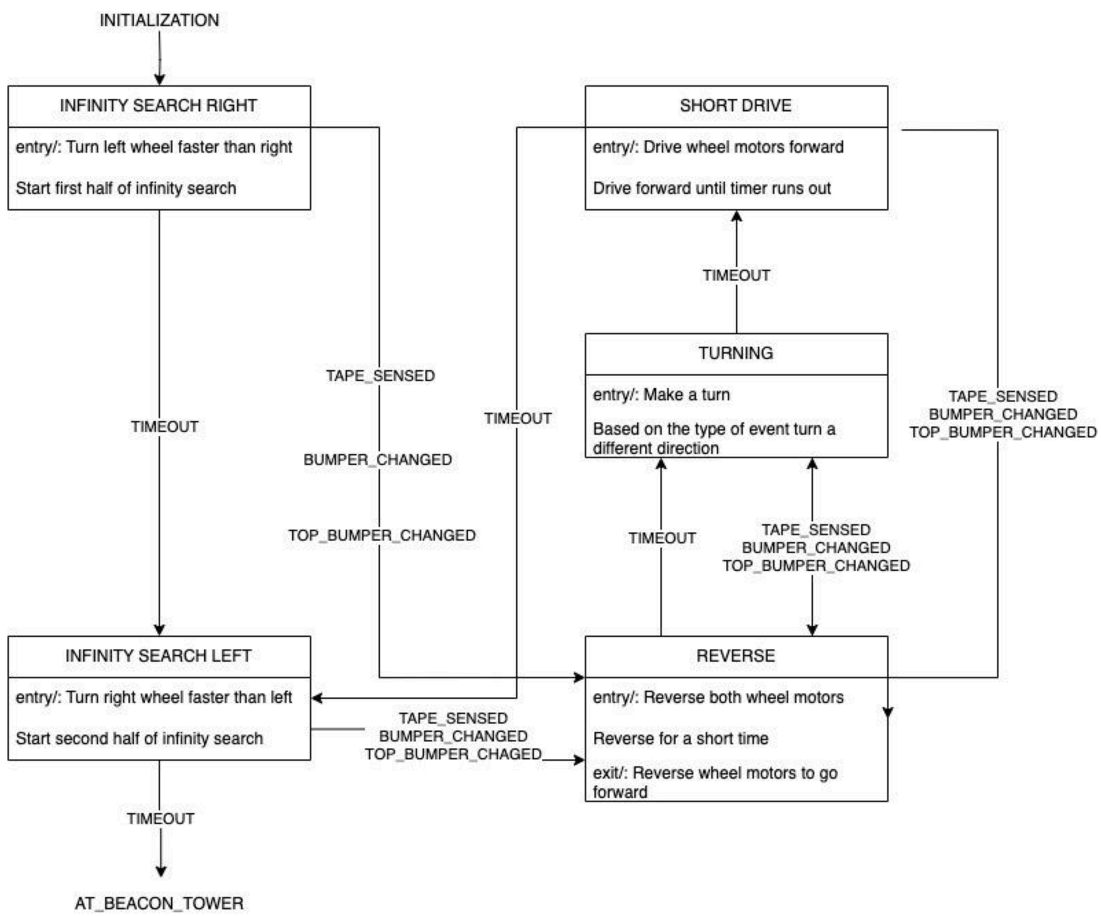

Image 22: Searching for Spheres Sub State Machine

The first sub state machine (SSM), Search For Spheres, was set as the initial SSM in which the robot would perform a short roam around the field allowing enough time to collect at least two spheres. Collecting at least two spheres is essential so that once it moves on to the next SSMs it can deposit at least the minimum amount of required spheres through the trap door. The state machine begins in a pseudo initialization state in which any behavior like setting the speed of the motors can be done before entering the actual state machine. In the active search states of Infinity Search Right and Left, the robot rotates on the right and left wheel as their axes. This rotation allows the robot to cover a 360-degree field of view and increase likelihood of finding spheres. These states direct the robot to perform an infinity or figure-eight search pattern, systematically covering a wired area. Through the process, the state machine implements other states in which it is capable of handling special conditions and unexpected events. For example if the robot encounters an obstacle like the “Dead-Bot” or the tape marking the border of the field it can adapt its behavior to move away from the objects. Once a successful search has been completed the SSM will conclude and return back to the Top Level HSM with an event indicating it to move on to the next SSM.

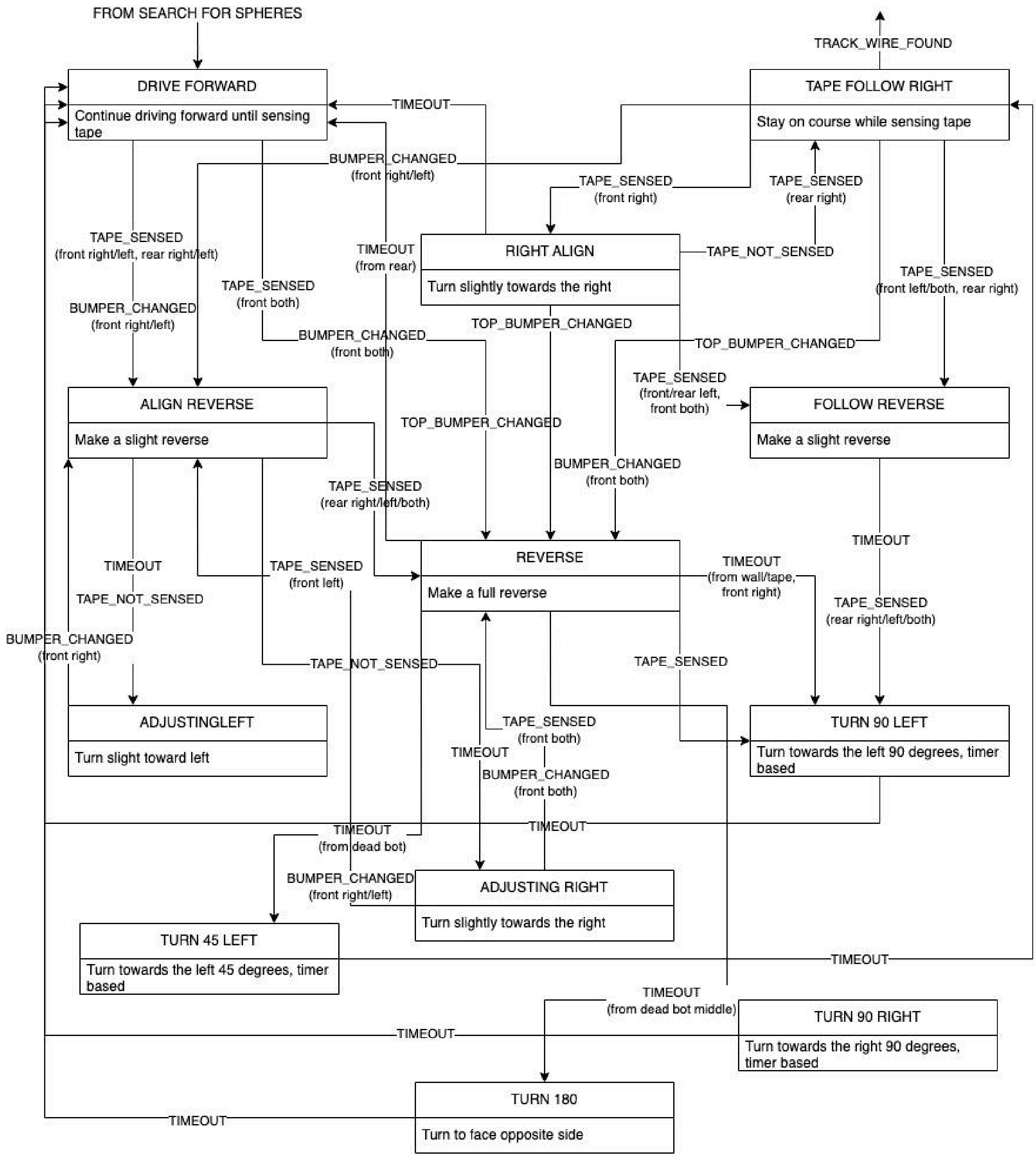

Image 23: Collection 2 Sub State Machine

After completing the infinity search, the HSM will start the second SSM, Collection 2. In this SSM, the primary focus is driving toward the black tape marking the perimeter and then following along until the track wire sensor detects the track wire outlining the trap door on the wall. Initially, the SSM begins in the Drive Forward state in which its goal is to find the tape at the border. Once sensing the tape it will align perpendicular to it and then turn parallel to it and follow the tape around the border using the IR tape sensors on the right side of the robot. If at any point, tape was sensed from the front left IR sensor, it would indicate that the robot was at a corner and it would turn left 90 degrees to continue following along the tape. Once a track wire was sensed, the SSM would return back to the Top Level HSM. Throughout the SSM, the robot also had to deal with collisions. A couple possible scenarios would be in the initial driving forward state in which it would hit a wall in which case it would alight with it and then turn 90 degrees left in the direction of the tape. All other collisions would likely be with the “Dead-Bot” obstacle during the tape follow algorithm. The algorithm would handle these collisions by reversing slightly and then making a semi circle turn around the object in hopes of avoiding it. In case it would still collide with the object in that turn it would repeat the same behavior until it sensed tape again. Once sensing tape it would realign with it to continue tape following. After reaching the track wire at the trap door, it would be ready to deposit any balls collected during its run.

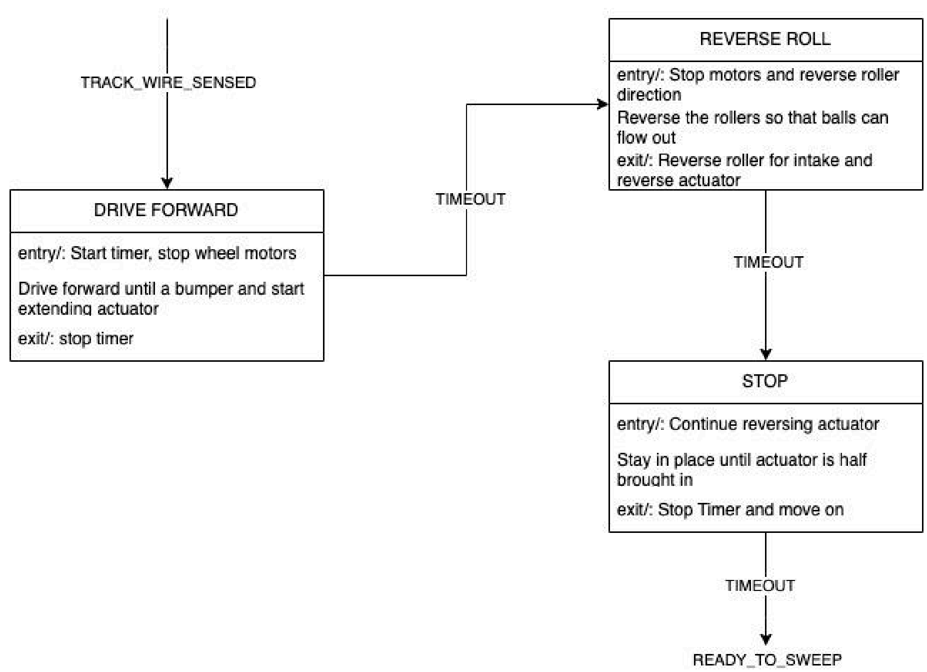

Image 24: Deposit Sub State Machine

Once a track wire was sensed in the previous SSM, the HSM would transition into the next SSM, Deposit. Upon entering the robot would drive forward state until sensing a bumper event indicating that it has reached the wall where the track wire was located. This event would intricate it was time to deposit. The transition between the SSMs would happen very quickly as our sensor would detect the track wire enough time before hitting the wall. Once the wall was detected the linear rod actuator would begin to extend in order to open the trap door. The actuator was controlled by a timer so that it would extend only the necessary distance and then the roller would begin to reverse so that the spheres would freely flow out to the other side of the field. After a couple seconds all the balls trapped inside the cavity of the robot would roll out since they were on a ramp. Once the deposit was finalized the SSM would return back to the Top Level.

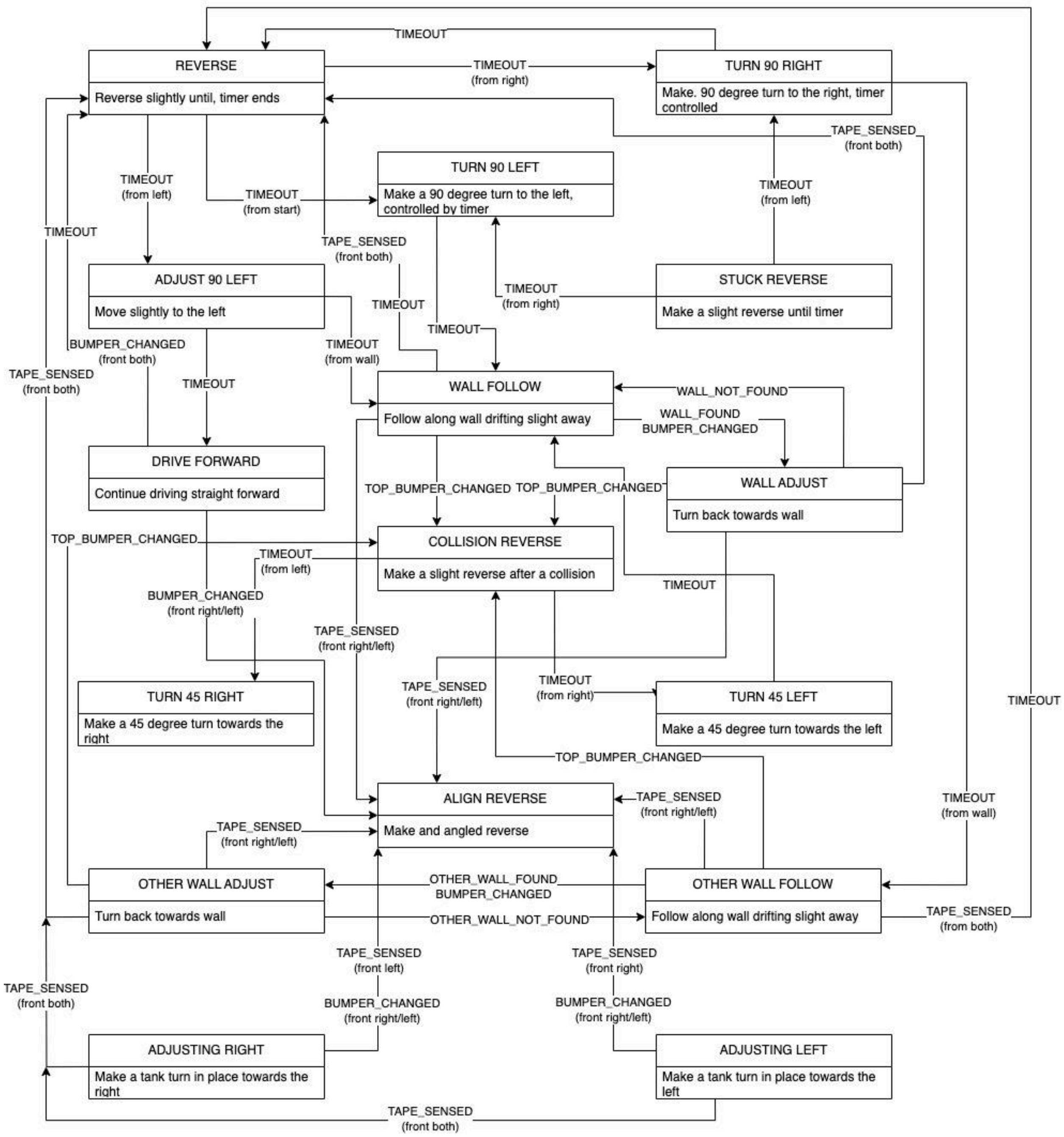

Image 25: Collection 1 Sub State Machine

Following the deposit through the wall, the Top Level HSM would enter the final SSM, Collection 1. In this SSM, the focus was driving back and forth along the wall to collect as many balls as possible. As many balls would lose momentum after hitting the wall and due to the field being unlevel and favoring balls toward the wall. This bias caused us to focus this last collection algorithm to traverse only along the wall as to increase our chances of collecting at least 30 balls during the two minutes. To traverse the wall we used the two IR sensors located on the side of the robot. After finishing the deposit, the robot would reverse turn 90 degrees to the left and then use the IR sensor to align itself continuously. Once it reached the tape on the other end of the wall it would turn until the IR sensor on the other side of the robot would be triggered. This would indicate it was aligned with the wall again and it could continue along the wall until it again reached the other end. It would continue this behavior until the end.

Breaking up the robot's complex behavior made it manageable and allowed us to test each state machine individually and make sure it would handle collisions and maintain reliable behavior.

Challenges Encountered

Open Cavity VS Roller

Throughout the design and testing of our robot we encountered many challenges that lead to redesigns or various work arounds. The first challenge that we faced was the difficulty of closing the servo door for collisions before we implemented the roller; this problem led us to redesign our bot with a roller rather than just having an open cavity. Having a roller solved a lot of problems for us since it prevented balls from being lost and also helped simplify our ball ejection method.

Ball Ejection Servo

When it came to the ball ejection method we initially had the idea of using a servo attached to a wall to hit the balls out of our bot. However, this method would only deposit a small fraction of the balls that were actually stored in the bot. To fix this problem, we added in a ramp that all of the balls would sit on and slide down instead of having the balls just riding on the ground underneath the robot. This was a change that we definitely wished we made earlier because it would have made the clearance for the wheels and wheel motors a lot easier.

Wheel and Wheel Motor Problems

The wheels of our robot and the motors that drove the wheels were a big challenge for us throughout the project. Our original design had the balls riding on the ground underneath the robot, so we needed to find wheels that were large enough so that the balls could roll under the motors, this led us to getting 4 inch wheels. However, the first set of 4 inch wheels that we got were way too soft and squishy. Because of how soft and squishy these wheels were, they had so much friction on them that our motors weren't able to generate enough torque to move them, especially with our laser cut wheel mounts. Once we got harder wheels that had collar locks to prevent them from slipping on the shafts a lot of our problems were solved. The issue that we ran into with the wheel motors was that we needed the balls to go under them so that we would have more storage space, which became an even bigger problem when we decrease the clearance to the motors by adding in a ramp underneath our robot as well. It ended up taking a lot of trial and error to find the perfect ramp height where the balls could still fit underneath the motors but also would slide to the roller.

Actuator Speed

Even though the actuator was very simple and easy to implement, a large issue with it was that it took 18 seconds to fully extend and 36 seconds total to fully extend and retract its 4 inches. Even if we were not using all of its length it would take a very long time, considering the fact that we only have 2 minutes total per round. We ended up just getting the timing for the extension and retraction as little as possible and placing the actuator at the best possible angle and position so that it would have to do the least stretching and retracting as possible. From this point we saw it as pretty much useless to try and further optimize what we had because we might only gain a second or two at most and those extra seconds weren't going to make or break our design. If we wanted to make a more substantial improvement we would have to go for a much larger redesign, which we did not have the time and energy to undertake.

Balls Going Against the Wall

When we started actually doing our testing on the actual field we discovered that the field wasn't perfectly flat and that almost all of the balls would end up by the wall rather than being evenly distributed throughout the field. This was a problem because we designed our robot and our state machine to do an even sweep of the field since we were under the assumption that the balls would be evenly distributed. After trying to achieve the minimum specification a couple of times with our even sweep, we saw that we would have to get way too lucky since almost all of the balls would just go by the wall. To solve this problem we decided to implement wall following as our collection method rather than using an even sweep. To have our robot effectively wall follow we also needed to add in 2 new tape detectors on the sides of our robot since our bumpers couldn't effectively actuate when coming into the wall at such a shallow angle. These changes weren't that difficult to make since we already knew how the tape detectors worked and also had a method for tape following that was very similar to what we ended up implementing for wall following.

Ghost Events

We had an issue where we were getting a lot of unwanted events sent to our microcontroller from loose connections from the sensors and the bad connection from the battery to the voltage distribution board. This was a big challenge for us since we had no idea what was causing these events and had to go through so many different things to try and hunt down the problem. Eventually we found that it was just a series of bad connections whether it be to the ports of our microcontroller or to the power distribution board from our battery. These events made it really difficult and discouraging to test since our robot would be working perfectly for multiple runs in a row and then out of nowhere it would completely break and stop working even though we changed nothing. For a while we thought that this was just a problem with having a bad battery, but we eventually realized from observing our robot while wiggling various connectors that it was a connector issue.

Fuses / Power Distribution

Throughout this project we went through a couple of fuses and power distribution boards. Like we previously mentioned we always thought that we had problems with our batteries or power distribution boards, when testing to try and find out what the exact problem was we would often probe and test our power delivery with methods that weren't very proper or hygienic. This would lead to us either blowing fuses and burning the power distribution board itself. Luckily this was a rather rare occurrence since these mistakes lead us to have to rewire our entire robot and would cost us a lot of time.

Skid Height

Getting the perfect skid height for our robot was a large challenge for us. This was difficult since we needed to balance getting traction to our wheels and keeping the bot high enough to lead balls in through the roller. This was a very fine balance of fractions of inches. We ended up reprinting and filing our skids an uncountable amount of times to achieve the perfect skid height. This was even more of a problem when we added in the ramp to our design. Our ramp rode so low to the ground due to our motor tolerances that it could get caught on the edge of the tape on the field. Because of this, we needed to have the perfect skid and ramp height for our robot to be able to navigate the field effectively. However, after lots of trial and error we were able to find the perfect height that was also robust.

Zip Tie Funnel

Like we previously mentioned, our robot evolved to use wall-following as its collection method; however, the wall following by itself wouldn't get the robot close enough to the wall to pick up every single ball . To solve this problem, we added zip ties to funnel the balls that were right up against the wall either into our robot's collection or into the gutter to be dispensed. Having the perfect zip tie length and shape was very crucial to getting a robot that could effectively grab balls off of the wall. By just changing how our zip ties were bent and attached to our robot we were able to increase the amount of balls collected/deposited by our robot by an average of 10 and successfully achieve the minimum specifications.

Performance

Image 26: Competition during our first round

Image 27: Competition, winning one round

The robot was able to meet the minimum specifications by collecting at least 30 out of 40 balls and depositing balls through the trapdoor within 2 minutes in 2 out of 3 test trials. In the first trial that we passed we collected/deposited 37 of 40 balls and in the second trial that we passed we collected/deposited 32 out of 40 balls. In the competition itself our robot made it to the semifinals; however, we encountered a couple of issues that we had not seen before. During the competition we were able to collect and deposit balls; however, we had an issue where our roller was getting jammed so we were unable to collect as many balls as we usually could. During the competition we also had problems with our tape sensors not being able to detect the tape or the wall when we expected them to. This caused our bot to fall off of the stage and get stuck in certain wall following states for too long. This inconsistency with the tape detectors could be because of the heat lamps in the lecture hall the competition was in, which were not present during any of our testing. We were unable to identify exactly what was causing the roller to get jammed, but our best guess was that it was a lack of current from the different battery that we were using and hadn't previously tested with before. Despite these issues our robot was still able to make it into the semifinals during the competition and performed very consistently during the checkoff, where it was in the conditions that it was designed for.

Conclusion

Our project successfully demonstrated the integration of mechanical, electrical and software systems to create a functional and competitive and autonomous robot. Throughout this project we applied our knowledge from coursework, labs, and practical experience to tackle intricate engineering challenges across electrical, mechanical and software domains. This project not only honed our technical skills but also emphasized the importance of teamwork, time management, and adaptability. The successful completion and performance of our robot in the competition showcased our ability to innovate and apply mechatronics principles effectively. By breaking down complex behaviors into manageable tasks and using hierarchical state machines we could ensure that our robot could reliably perform essential tasks such as navigating, collecting, and dispensing balls. The iterative design process coupled with thorough testing and troubleshooting allowed us to address various challenges and optimize the robots performance.

Overall, this project has provided us with a solid foundation in mechatronics, enhancing our understanding of integrated systems and preparing us for future endeavors in the field of robotics and automation. The experience gained from designing, building and competing with our robot has been invaluable, equipping us with the skills and knowledge to tackle complex engineering challenges in the future.