**ECE 118L**

Final Project

Instructor: G.H. Elkaim

June 10th, 2024


| *Author* | *Student ID* |
|---|---|
| Shreya Sinha | ssinha12 |
| Jinsung Park | jpark598 |
| Rohan Person | rperson |

# I. Design Constraints

## A. Goals

The goal of this project was to autonomously navigate a field (Image 1) and collect strewn balls. The robot utilized infrared sensors, track wire sensors and limit switches to take input from its environment in the form of digital and analog signals into a PIC32 microcontroller. The robot then sent signals to motor drivers and other peripherals to influence the outside world, like driving around the field or opening a trapdoor.

More specifically, the two towers located in the corners shot balls intermittently over a 2 minute time period. The robot needed to contain or send at least 75% of the balls shot. Sending was defined as navigating to a one-way trapdoor on the field, and pushing the ball fully through the door. In addition to the 75% figure, robots were required to send at least two balls through the trapdoor.

## B. Restrictions

Once energized, the robot must be autonomous for the full two minutes. No energy can leave or exit the field, such as lasers, physical touch, or sound. If a robot fell off the field, defined by the black strip of tape around the edge of Image 1, the robot was disqualified. If a robot could not resolve a collision within 5 seconds, the robot was disqualified. Notably, sliding along a surface did not count as a collision, as controlled translation still occurred.  Only balls pushed through the trapdoor count as being 'sent', and for a ball to count as contained it simply had to be within the robot's footprint and not able to be jostled loose or be removed without significant effort.

Robots themselves had to fit within an 11" cube, but could expand after the round started. Additionally, their main composition had to be of laser-cut MDF sheets, though 3D printing and traditional machining can be used for small necessary mounts or brackets.

## C. Methods

The two most important restrictions we faced were regarding the immediate disqualifications: falling off the field, and getting stuck. Both problems were solved by using sensors to detect events ahead of these failure modes. Since there was a distinct line of tape around the majority of the field, we used a sensor that was able to detect different colors, more specifically detect changes in reflectivity of colors: a combo of an infrared LED and phototransistor. For the collisions, we employed a limit switch similar to those found on 3D printers, or in the Cockroach Lab at the beginning of the quarter.

# II. Approach

## A. Navigation

Our initial approach to the navigation problem was to follow the line in a counterclockwise motion so that the robot will align itself once it detects track wire at the top right corner of the field. Ball collection will take place in a separate state and will predominantly bounce around the map till 100 seconds has run out. This idea would incorporate 3 different states: Meander, Olympic, and Disposal. As we slowly found out, our method, dumping the ball to the other side will prove more cumbersome than expected, thus we had to shift our strategy to dump the ball as often as we possibly could. This led to us eliminating the Meander state and trying to simply find the tape and completing a line trace + disposal as often as we could.

## B. Collection

As we were working on the sub-states for this strategy, we found a group had gotten a near perfect score on a run simply by tracing the wall that is connected to the trap door. As it turns out, the balls conveniently land near the wall because balls lack bounce and thus happen to clutter where it is hit. Now, normally this should not be possible and even possibly goes against what this project is supposed to challenge us to do. This led us to redesign our strategy so that the bot will first find a tape, dispose, and continue tracing the wall til the end of competition. However, this method is limited as it relies heavily on the placement of the obstacle.

The biggest challenge came from obstacle detection. We were having issues with random state changes and timers were going off randomly. Additionally, in our switch-cases we had to incorporate if statements in every sub-state to account for bumps and tape. Looking back, we should have added more conditions so that the bot never escaped past the tape in an unforeseen scenario. Our attempt to catch balls that were near the edge ended up being our demise. We should have made a condition to never even go past the tape in the first place.

## C. Disposal

To avoid unnecessary complication, we designed around using the front roller for both collection and disposal just by reversing the direction of the roller, compared to a rear-access panel that other groups gravitated towards. Once we learned the difficulty of opening the trap-door before sending balls through, we incorporated a servo that would bring an arm down to push the trapdoor open before reversing the roller and disposing of the balls. This meant that a complex series of steps must be undertaken once the track wire is detected, to avoid the arm getting stuck on the wall or the robot not being lined up properly.

# III. Mechanical

## A. Navigation

The robot navigates with two-wheel drive using differential steering to turn. By applying custom densities to the representations of the Pic32, 2 H-bridges, battery, limit switches, and motor, we were able to manipulate the model until the center of gravity (CoG) were in front of the drive motors. This allowed us to only have skids on the front of the robot rather than needing them in the front and back. Additionally, we optimized to have about a third of the weight on the front skids keeping them pressed against the ground: this both prevented balls from high-centering us preventing locomotion, and meant we didn't wheelie during fast accelerations.

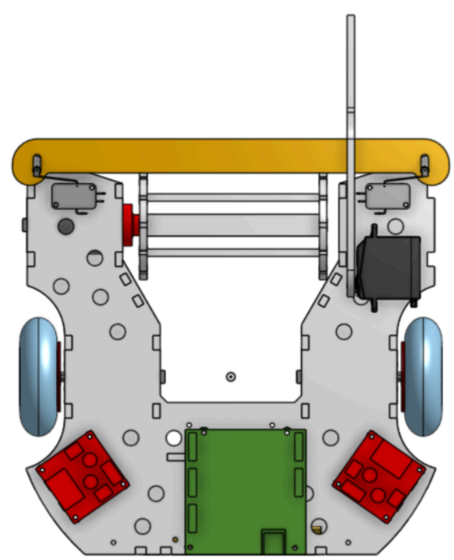
To avoid collisions, two limit switches were mounted to a bumper on the front of the robot. These are triggered when the robot runs into an obstacle or the wall, letting us know that we need to maneuver around the blockage. Additionally, four tape sensors were mounted on the corners, triggering whenever the robot approached the tape surrounding the play area.

## B. Collection

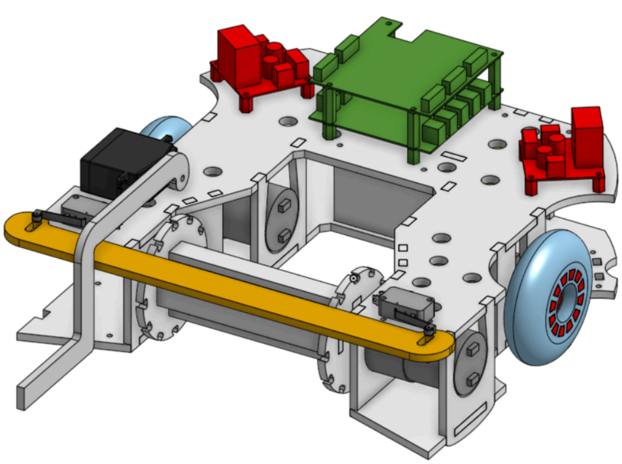
The robot collects balls randomly, and does not actively seek them out. An actuated roller at the front of the robot uses stretched compliant bands to pull the balls into the center collection area. Two wedges on either side of the roller sweep balls into the roller, and additional zip-tie "whiskers" are affixed to scrape out balls from hard-to-reach nooks like next to the wall or stuck against the obstacle. Once pulled inside, the balls are unable to escape as there are no holes in the chassis wide enough for the balls to pass through, and they cannot escape through the front while the roller is spinning.

## C. Disposal

The robot senses the trap-door by use of a track wire detector that detects the track wire around the trapdoor. Once the analog input value is greater than a certain threshold, the robot backs up, brings down the arm attached to the servo meant to hold the trap-door open, and reverse the direction of the roller motor. This now shoots all the balls out through the trap door. An angled piece of foam core ensures that all the balls roll to the front and get disposed of.



## D. CAD Screenshots



# IV. Electrical

## A. Overview

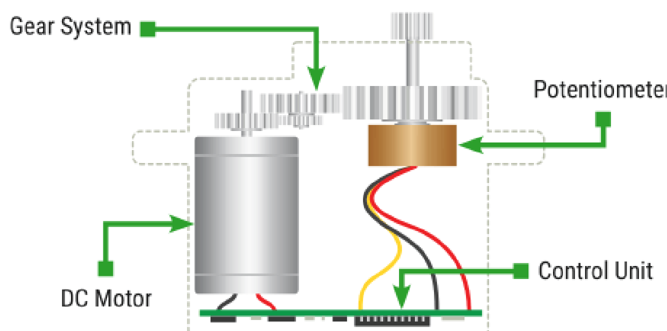
The "brain" of our robot or the main source of distributed power and signal control was through the PIC32 microcontroller alongside the power distribution board provided by the UNO32 stack. We used a 9.9V LiFO battery to power our robot although we were allowed the option to use a higher voltage battery (we were recommended to use a bomb bag for that in case of explosions). This lower voltage had contributed to the somewhat slow speed of our robot and having a higher powered battery might have mitigated that issue.

The power distribution board of the UNO32 stack provided 9.9V which we then distributed into 3.3V and 5V channels with appropriate voltage regulators to provide voltage to power our various sensors and actuators. Each separate electrical component was on a separate perfboard just for ease of reference and change if we found it necessary to modify a certain component without affecting any of the other ones. The various digital and analog pins on the PIC32 provided signals the rest of the motors, IR sensors, bumpers, RC servo, the two H-bridges we used, as well as our track wire detector.

## B. Sensors

We used IR sensors to detect tape on the ground which represented the boundaries of the playing field as well as bumpers to discover when an obstacle was in the path of the robot.

## C. Actuators




Four motors were used in total: two to drive the robot using differential steering, another to power the combine roller, and a fourth incorporated into a servo used to open the trap-door to deposit. To drive the three DC brushed motors, two H-bridges were used as mentioned above. Each channel of a two-sided H-bridge takes in three inputs: Enable, and In 1 and In 2. In 1 and In 2 create the potential difference in voltage that will drive the motor, which controls the direction that the motor spins. The Enable pin can be driven with a pwm signal to achieve variable speed without sacrificing power as you might by varying input voltage. Both the H-bridge and the servo take in a pwm signal with pulse-width of 1000 to 2000 μs, but while the H-bridge uses it to control whether the motor draws current, the servo uses the signal and a potentiometer attached to the output shaft as seen in the image on the right to drive the motor bidirectionally, achieving fairly accurate closed-loop positional control.

# V. Software

## A. Sensing & Driving

Our sensing mechanism consists of 4 IR sensors on either “corner” of the robot and 3 bumpers on the front and back. The IR sensors are calibrated such that it can detect the black tape since the reflection of the dark outline is much weaker than that of the field itself which is white. The IR sensors in particular are finicky to use because they need to be positioned in a way such that it wouldn’t pick up false positives off of the MDF which was a big part of the issue we were facing. Again, all of this could’ve been prevented had we considered our design options a little better but this wasn’t such a terrible issue to fix since we later opted to move the IR sensor closer to the ground so nothing else would interfere with it. It is worth noting that we designed our robot such that the IR sensors were aligned parallel. This allows us to track the tape sensor and follow the line straight.

I found that the bumpers don't require Vdd to be connected to the power source for the detection to work. I assume this is the case as the residual electricity on the wire is detected by the port. As long as the bumper GND is connected to ground, there should be a noticeable difference in the voltage. However, after more testing we did eventually connect Vdd to power which resulted in better detection. This could’ve been all placebo but I just wanted to point that out.

## B. State Machine

The state machine was discretely described in a prior section but I shall elaborate further here to better illustrate the procedure better. There are 3 states total: Meader, Olympic, and Disposal. The meander state is the beginning stage where the bot will idly bounce around the map to collect as many balls as possible. Then once a certain time period has passed, it will transition to Olympic state. In the Olympic state, the bot will attempt to go straight and maybe evade obstacles if needed. Once it finds a tape, it will trace the line in counter clockwise motion until it finds track wire. The track wire will trigger a state change to Disposal state as it means the robot is not facing the trap door to dump the balls on the opponent’s side.

In Olympic state, the bot finds a tape, then aligns itself so that it is facing parallel to the line. Once the bot is aligned, it tries to go forward but in an arc such that it will again pass the line where it can repeat the cycle. Corner turn poses less of an issue with this approach as the aligning of the bot depends on the way that the bot crossed the black tape. The bot will attempt this motion to path in a circle until it comes facing the track wire.

Track wire service will activate an event in which the HSM can use to state change to the Disposal state. In the Disposal state, the bot will move back a bit, move down the hand, and push forward so that the trap door can open. Open trap door will allow the intake motor to shift reverse and let the balls out. 5 seconds is all we need to let all the balls out usually. Once that motion is complete, the bot will pull out, move the hand up, and do the left turn. Aligning once it completes the left turn is essential because we want to move straight while not grinding against the wall. This sweeping motion will continue until it reaches the end, then the bot makes another round back to the trap door side of the field. The cycle repeats two times then enters Olympic state again.

# VI. Conclusion

In conclusion, this project challenged us in both perseverance and teamwork as we were tasked with building a fully functioning robot within four weeks. The hurdles we faced included Rohan being sick with Lyme disease and the PIC32 board failing just minutes before the competition. Despite these setbacks, we managed to push through, and the robot turned out alright.

Our initial strategy to get a minimum specification checkoff was to attempt runs as many times as needed to achieve a perfect layout, allowing the bot to pick up enough balls and dump them to the other side. However, this approach proved unsuccessful, as our bot was not versatile enough to navigate a field it wasn't designed for. Our manual tests, where we let the bot wander in a quarter of the field, were inadequate for realistic scenarios where obstacles would get in the way.

It was only when we began to reevaluate our strategy and develop a better wall-following system that we found success. The takeaway from this experience is that one shouldn't be too fixated on a "working" system and should always remain open to new possibilities.

## VII. Bill of Materials

| Item | Amt | Cost | Total Cost |
|---|---|---|---|
| Uno Stack | 1 | $0.00 (Provided) | $0.00 |
| L298N H-Bridge | 1 | $0.00 (Provided) | $0.00 |
| 1500mAh 9.9V 7C LiFe Battery | 1 | $0.00 (Provided) | $0.00 |
| BELS L298N H-Bridge | 1 | $0.00 | $0.00 |
| BELS DC Brushed 150rpm motor | 3 | $0.00 | $0.00 |
| BELS Omron Limit Switch | 3 | $0.00 | $0.00 |
| BELS Tape Sensor | 4 | $2.00 | $8.00 |
| BELS Servo | 1 | $0.00 | $0.00 |
| Through-hole 5V regulator | 1 | $0.00 (Provided) | $0.00 |
| Through-hole 3.3V regulator | 1 | $0.00 (Provided) | $0.00 |
| Zip-Ties | 2 | $0.00 (Scavenged) | $0.00 |
| Rubber-Bands | 6 | $0.00 (Scavenged) | $0.00 |
| ¼" OD 1" mild steel rod | 1 | $0.00 (Scavenged) | $0.00 |
| ¼" Shaft Collar | 1 | $0.16 (Bulk Order) | $0.16 |
| 8mm Plastic Bearing | 1 | $0.80 | $0.80 |
| M3 Hardware Kit | 1 | $6.71 | $6.71 |

Total Cost: $15.67 USD