Final Design Report

Group 3
Edward Bryner, Sakina Girnary, Ari Messenger, Azwad Sabik

Mechatronics EN.530.421
Professor Charbel Rizk

Our team's goal was to develop a mechatronic system that can 1) autonomously navigate through a horizontal pipe system consisting of multiple straight segments of PVC piping joined together with lateral turns up to 45° in angle and 2) autonomously collect a small number of lightweight spherical objects (ping pong balls) and transport them to a target location. Minimization of completion time for each task was a priority. In approaching the development of this system, our team chose to view the solution as needing two distinct subsystems - one for the pipe navigation, and one for ball scoring. We analyzed the problem by identifying functional requirements for each of the subsystems and subsequently brainstormed a variety of potential solutions (components) for each of the functional requirements. We then discussed the pros and cons of each potential solution both in-and-of itself and with regard to the overall subsystem until we finally decided upon our final design.

In designing the first subsystem, we identified four required elements of function. Our subsystem needed to be able to: move through the pipe from the initial opening to the end (locomotion); identify when it would encounter turns in the pipe (turn sensing); determine how to react to an upcoming turn (turn reaction); and communicate with our second subsystem in order to let it know that the first device has arrived at the end of the pipe course and activate the second subsystem (communication).

We decided on using a ball-caster and two foam wheels coupled with a high speed DC motor to drive our robot, placing a high design value in speed and lesser value in control since navigating the pipe course quickly provided our system with a significant point-advantage. The wheeled system would provide what we considered to be an optimal balance of these two primary factors, with the added bonus of the ball-caster reducing our device's turning precision requirements. To solve the possible problem of going backwards when flipped over (highly plausible, given a wheeled system that is not top-bottom symmetric), we added hubcaps to our design, which allow the racer to slip down the walls back into an upright position when it banks too high up the wall.

When trying to determine how our device should sense and react to turns, we found that all of our active solutions would have added an additional layer of complexity and require response that would potentially reduce our system's time efficiency drastically. We therefore decided upon a rigid-body construction to force the robot to passively turn when encountering a 45-degree face, and attempted to increase the efficiency of these turns by utilizing a ball caster and hubcaps with the hope that they would allow the system to "slide" down the turns.

Combining these functionality considerations together, we developed a pipe-racer consisting of two soft-tired rear wheels with semi-spherical hubcaps powered by a DC motor in the back, and a lightweight front-rounded chassis with a fixed ball caster in the front. Through this, we created a subsystem that reliably achieved great speeds and sufficient control throughout the competition.

Finally, In planning our relay system, we decided to add a limit switch to the pipe racer so that whenever it was held down by the walls of the pipe, the motor ran, and stopped when it was released. This ensured that while it was in the tube, the pipe racer moved forward, and as soon as it exited the tube, the pipe racer came to a halt. We also found that the extra force added by the arm of the limit switch on the top of the pipe bot added additional grip of the foam wheels, thus decreasing our time through the pipe from 3 seconds in the mini challenge, to 2 seconds for the final competition. Additionally, in order to activate the second subsystem, we added a yellow piece of paper to the front of our pipe racer so that when our pipe-racing robot exited the pipe, the Pixy CMUcam5 color sensor placed on the external surface of the ball-scoring robot could detect the yellow signature. Upon detecting the yellow signature, the ball-scoring robot was activated and ready to hunt.

For the design of the ball-scoring robot, we decided that it should hold multiple balls at a time for the purpose of scoring the correct balls efficiently. Due to the size requirement of this design, we chose to have two different robots for the pipe race and the ball game. We also decided that the benefit of having two well-optimized robots outweighed the bonus two points for having a single system. This allowed us to keep our pipe robot small and fast, and allowed our ball game robot to collect many balls and move across the large playing field more efficiently.

The first subsystem of the ball game robot that we discussed was locomotion. In order to accurately traverse the playing field, we considered many different locomotive configurations. To keep our system compact and simple, we wanted a rigid drive train, thus eliminating the possibility of using reoriented wheels like those of a car. To further simplify the path planning code we wanted to design a system that would allow for zero-degree turning, which would allow the robot to make direct, straight-line movements towards a target without a complicated feedback system. Therefore, the only viable options to fit these criteria are a set of omni-wheels, tank-like treads, or 2 bi-directional wheels with casters. We decided against the wheels with casters because the robot would not spin about the center when turning, which would potentially complicate the motion path. We chose omni-wheels over the treads for their low sliding friction as well as their availability in the lab. It also adds the benefit of being able to move sideways or diagonally, which is a benefit we didn't initially consider.

Next, we considered ball acquisition, sensing, and scoring. We did not want to simply push the balls around, since the robot would have limited control over the ball's motion. We also felt that controlling a single ball via a claw, clamp, or arm would be a waste of our precious 135 seconds per period of game time, since a design like that would require the robot to attempt to score every time a ball was collected. For that reason, we decided that we wanted to pick up as many scoring balls as possible and score them all at once, for a point apiece. This reduced the amount of time moving back and forth between the goal and balls, leading to a more efficient and higher scoring robot.

In order to collect the balls prior to sorting, we considered a variety of mechanisms for loading the balls into our system. We settled on a flexible rubber paddle system that rotated on a single horizontal axis, mainly because it could pick up balls against the wall and in corners, whereas the other options could not do so easily or at all. This system also pushed the balls into the ground when moving into them, thus preventing them from bouncing away from the robot. In addition, we realized that paddles spinning on a horizontal axis (rather than a vertical axis) had the widest collection area of the options we discussed.

Since our ball intake system was wide, the likelihood of non-scoring balls making it through was rather high. We decided that in order to be more accurate and allow for complete control of each ball, our robot would sort the collected balls internally, rather than try and avoid bad balls. Additionally, since the only discernable difference between scoring and distraction balls was the color, we could only sort by color. In order to sort them, we decided to use a color sensing camera system that can sort the balls as they are collected, and inform the brains of the robot so each ball can be sent to the correct storing location. As the balls were being raked in, we would be constantly moving the balls up a conveyer belt away from the collection system, and towards the ColorPAL and servo (our color sensing and sorting mechanism), so we could collect and sort balls simultaneously. Using the servomotor, our robot forced blue non-scoring balls to quickly be expelled through an exit area, while allowing green scoring balls to pass into a tube that could be filled up to a capacity of nine until it was time to score.

With this system, we had many options for the design of the path-planning algorithm. We decided that instead of conducting a pre-determined sweep of the entire field, it would be more effective if we directed our robot towards scoring balls. Since we would also be sensing and sorting the balls internally, we were not concerned about the accidental collection of blue balls, and therefore did not need to take them into account when path planning. We created a path for our robot that involved it methodically searching for balls from one side of the field to the other, so as to reduce our robot from navigating the field randomly and/or inefficiently by covering ground we did not need to. In order to direct our robots to the scoring balls, we used the Pixy CMUcam5 as a means for our robot to detect the green color it needed to orient towards.

Once we had the balls separated, with the green stored inside of our robot, we needed to score. To do so, we needed to first detect the goal. Since the goal had a red-colored border, which was a different color than the balls and pipe robot, we used the Pixy to locate the goal. Specifically, we decided that the best way for our robot to optimally position itself to score, would be if it lined up to the right post of the goal, and then make a slightly larger than 180-degree turn, to release the balls through the goal. We decided that it would be more reliable to align with one of the sides of the goal, since the top red bar of the goal was not always detectable when the robot was close to the goal.

Once the goal was detected, and the robot's tube with the scoring balls was aligned appropriately, we used another servomotor to release the balls from the tube. Therefore, our mechanism for the controlled dump relied on the pull of gravity to slide our balls out of the tube and straight into the goal.

Throughout our design process, we made several iterations to our original pipe-racing and ball-scoring robot subsystems.

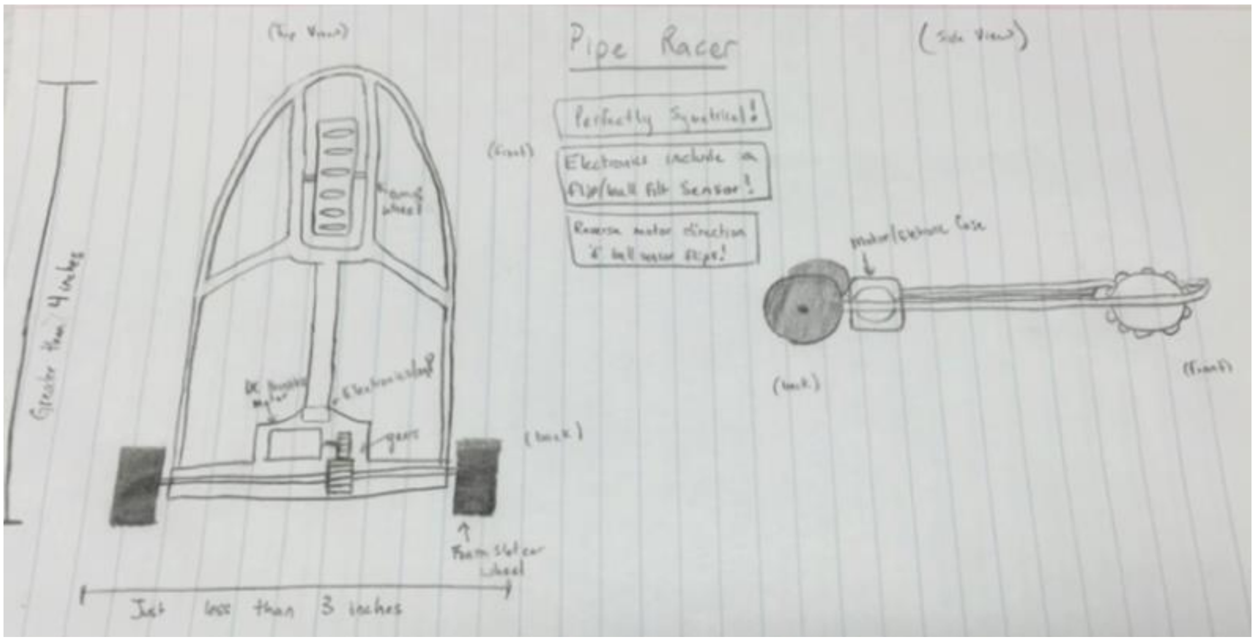

Photo 1: Original pipe racer design

Our original design of the pipe racer included an Omni wheel in the front to reduce our device's need for turning precision, and a brushless RC motor and motor driver to propel the robot forward at a fast speed. After building our initial prototype, we realized something very important: wheeled devices like ours inside a pipe have a high likelihood of flipping. In order to take the flipping into account, we initially planned to add a ball-tilt switch to sense when the robot had flipped, and thus allowed us to reverse the direction of the motor so that the robot could continue moving forward. However, due to the large size of the front Omni wheel, when our robot flipped, it was not able to do so fully, and instead had a tendency to lie on its side. We decided that a small ball caster would perform better in comparison to the Omni wheel, and the addition of rounded rails on each side of the racer, would allow it to flip the full 180 degrees it needed to.

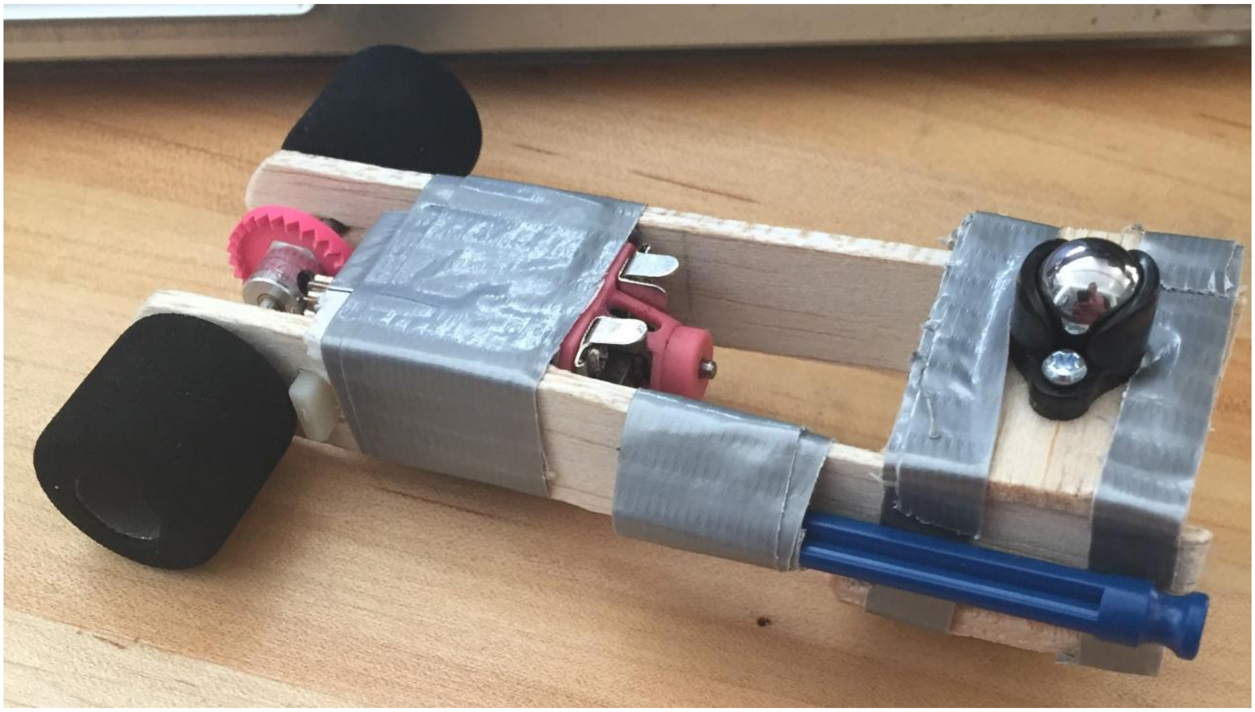

Photo 2: Prototype with ball casters and side rails

When we tested our new prototype, the rails actually impeded the forward motion as our pipe-racer consistently got stuck on the ridges between connecting pipes. We still wanted to add a mechanical aspect to the racer that would allow for easy flipping, and yet be low profile enough that the racer would not get caught on the edges of the joined pipes. This is when our team had a stroke of genius, and we decided to add rounded hubcaps to the back foam wheels in order to handle this. Little did we know at the time that it would also enhance the sliding motion of our pipe-racer down the side of the pipe. Thus, whenever our pipe-racer began to climb up the sides of the pipe, it slid back down the hubcaps until the foam wheels made contact again, thus preventing the racer from ever flipping over. This fortunately solved almost all of our problems; we got rid of the side rails that kept getting stuck on the ridges, the ball-tilt switch used to reverse the motor when the racer flipped, and the bulky motor driver and microcontroller. This decreased weight and complexity, and resulted in a racer capable of crossing the finish line in less than two seconds! With the addition of a limit switch and a yellow piece of paper, we were able to implement a simple stop mechanism and court-bot activator, respectively.

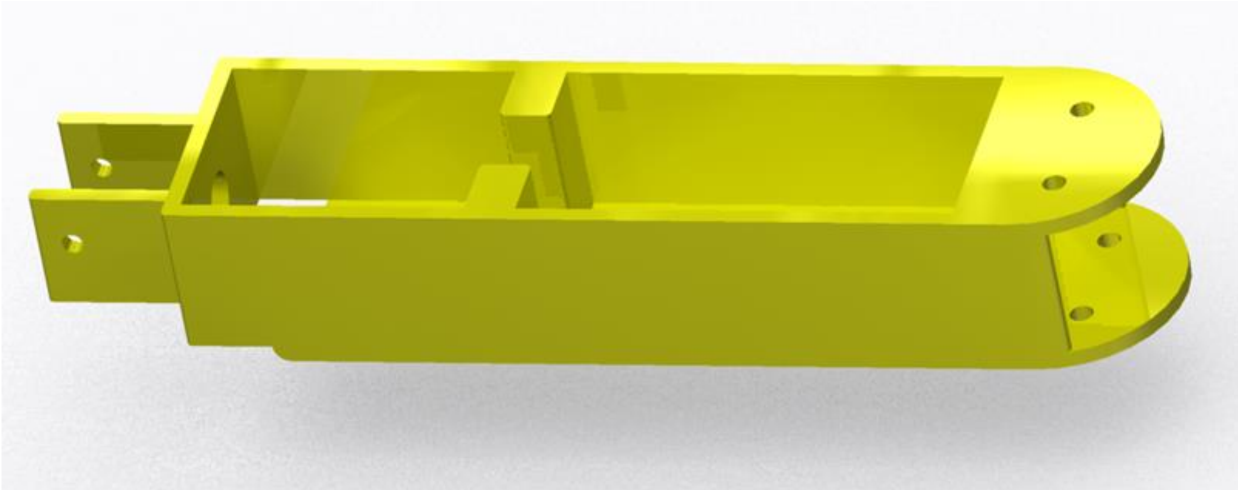

Photo 3: Pipe bot chassis, CAD render

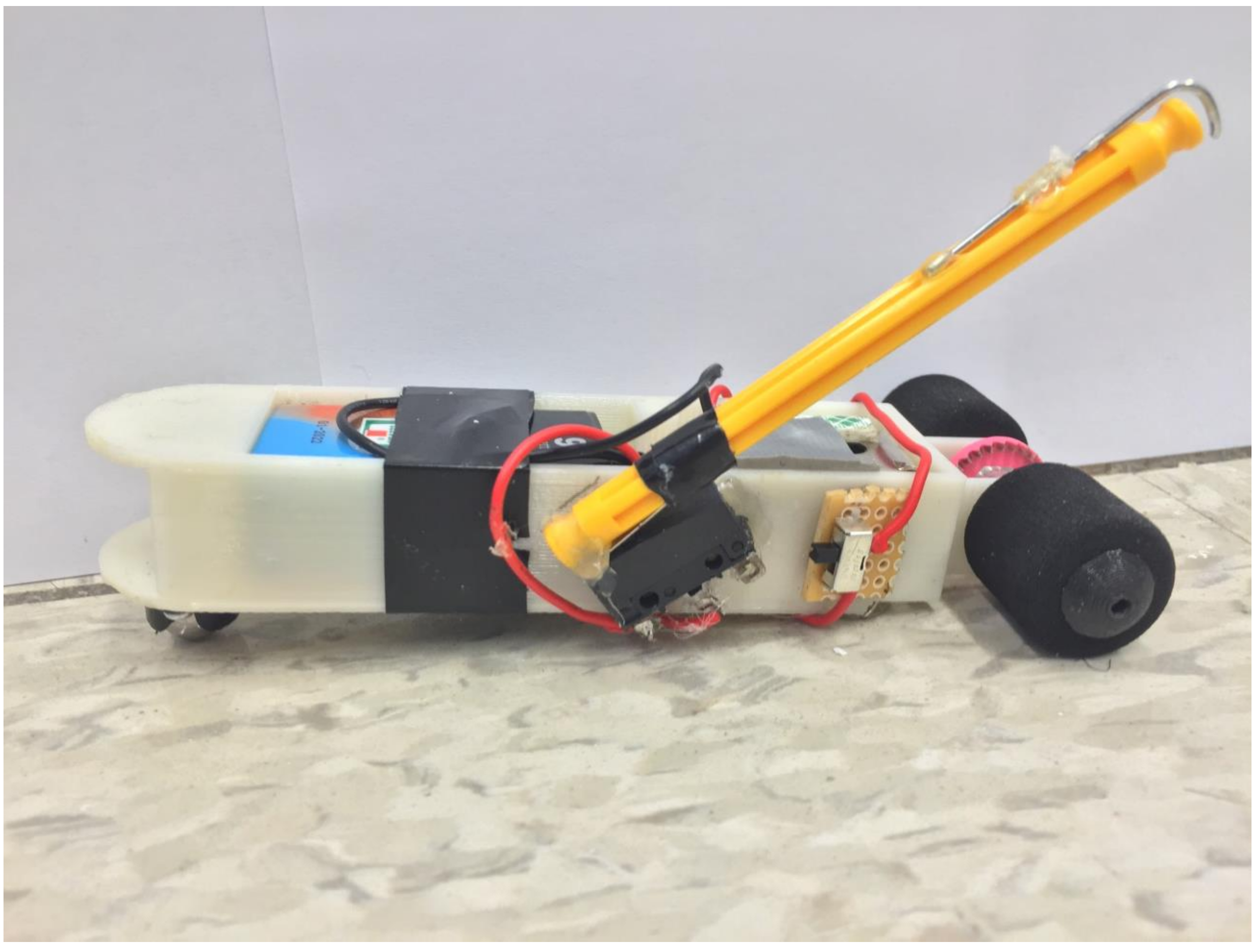

Photo 4: Completed pipe racer, with rounded hubcaps and limit switch activation arm

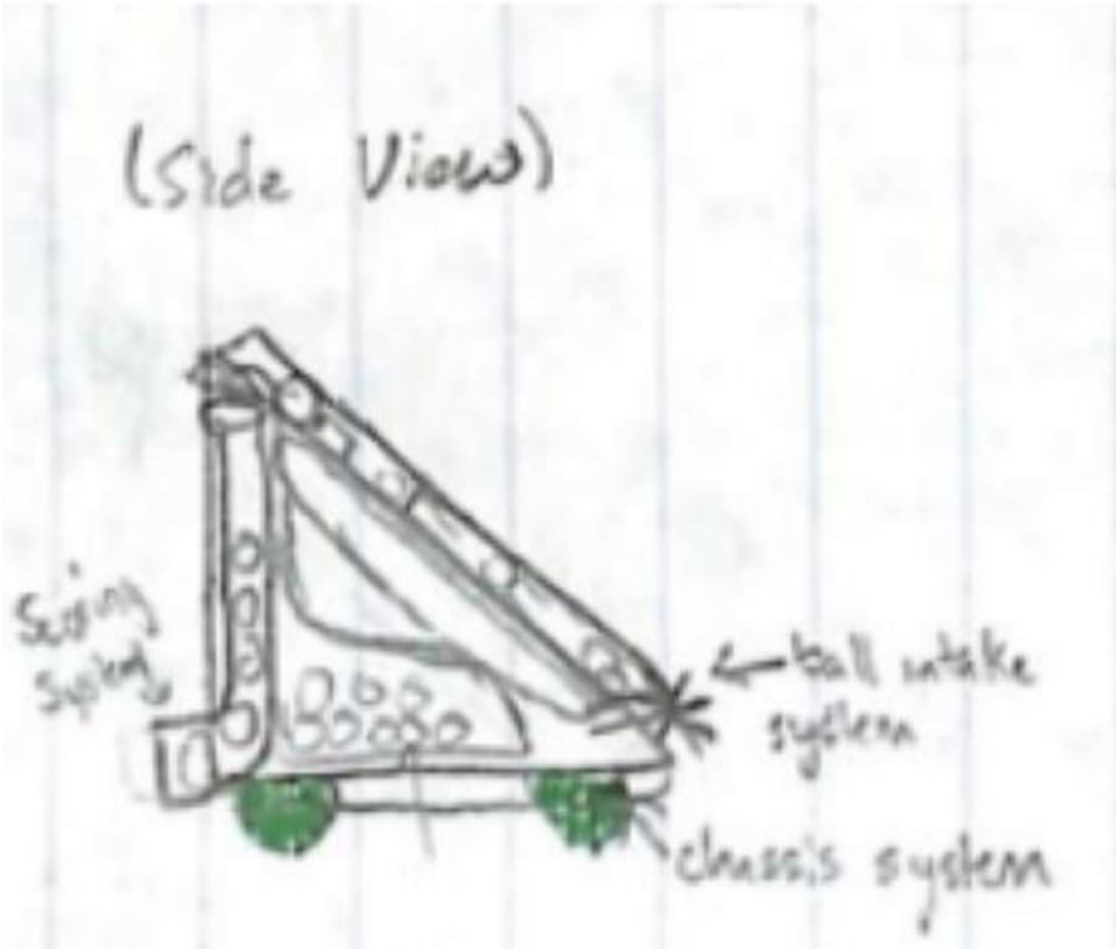

Photo 5: Initial ball robot design sketch

Although we stayed relatively true to our original design, the design process for the ball-scoring robot was more involved than the pipe-racer. We had far more obstacles to overcome due to the number of subsystems we were trying to integrate and the general complexity of the robot. We made two significant changes to our robot from our original design. Firstly, instead of storing the non-scoring balls in a compartment to keep them out of our way, we decided to eject them back out on to the playing field. This was done to stay within the size constraint of our robot and also to reduce the complexity of our already elaborate design. We also filled much of the needed space to store non scoring balls with weighted bags to ensure that all of the wheels of our locomotion system had enough normal force applied to them as to move accurately and as expected around the court. Secondly, instead of having our robot perform a pre-planned sweep through the entire field like a Zamboni, we decided to make use of the very convenient Pixy CMUcam5 device and leaned toward a more targeted approach. By placing the Pixy higher on our robot we could even see scoring balls behind non scoring balls thus keeping our sorting method inside of the robot an effective way of verifying we were scoring the proper balls since we had such a wide area with which to pick up balls.

Other than the two significant design changes, we made minor alterations to the design as we progressed. We did not plan the entire robot out, piece by piece, before attempting to construct it, which was a good thing in our case. Instead, we designed and built the most important or geometrically constraining parts first. For example, we prototyped the ball-intake system before the rest of the robot so we could account for the amount of space it would take up and to reduce its geometric footprint before committing to the design of any other part. We built the robots intake, ramp and escalator out of Vex Metal. We then moved these ideas into a wood structure for the mini challenge, and once we were happy with all of these parts, we laser cut acrylic and 3D printed to make our design perform more consistently and look much cleaner.

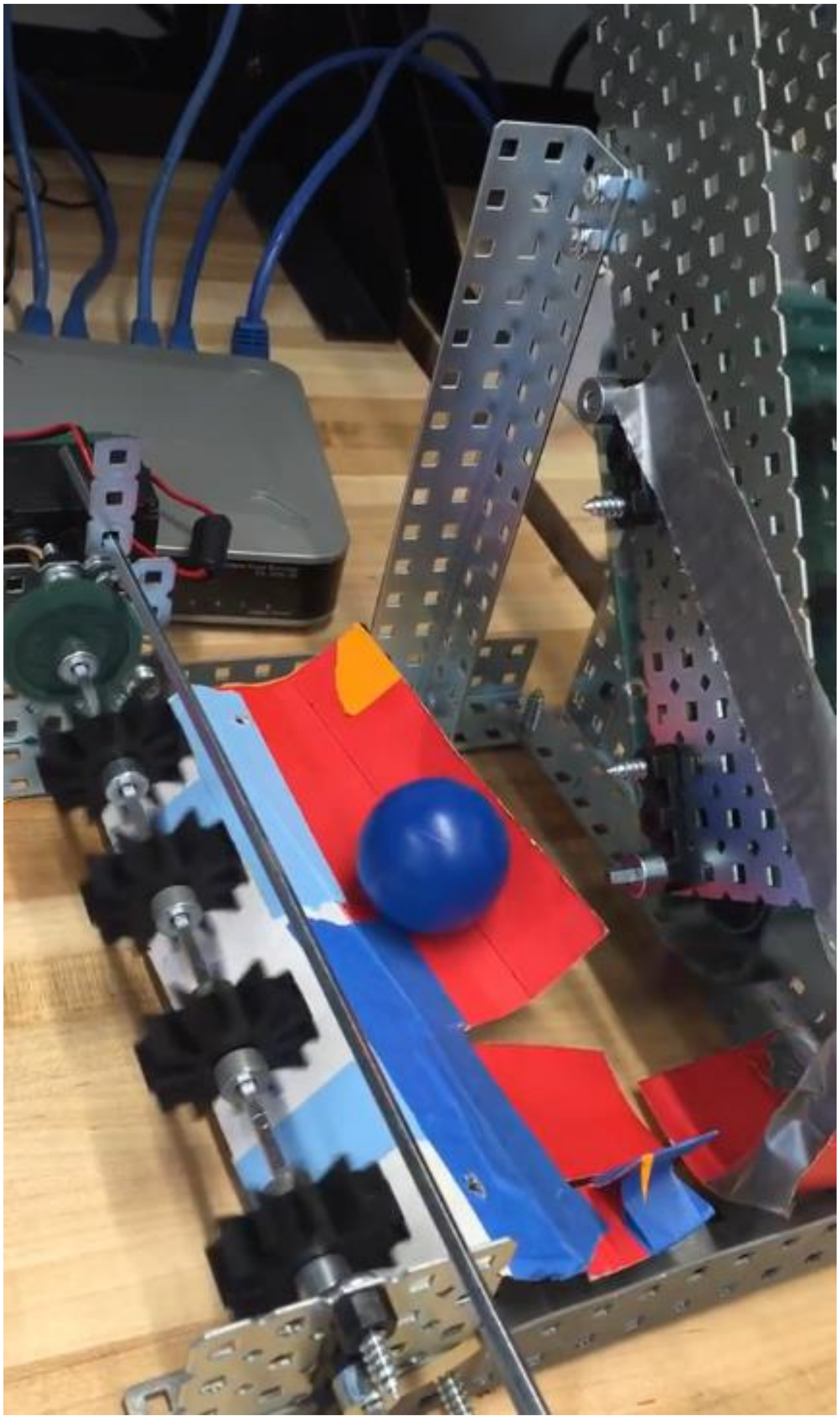
Photo 6: Ball intake system prototype - made with Vex metal

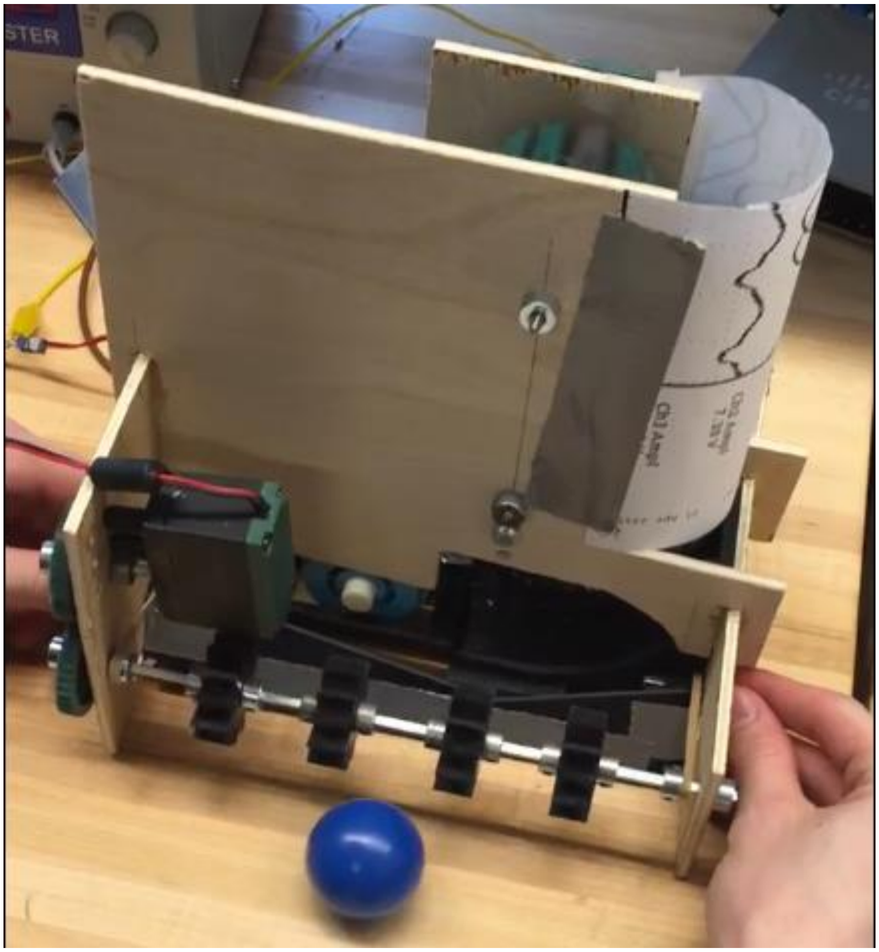

Photo 7: Wood ball bot prototype

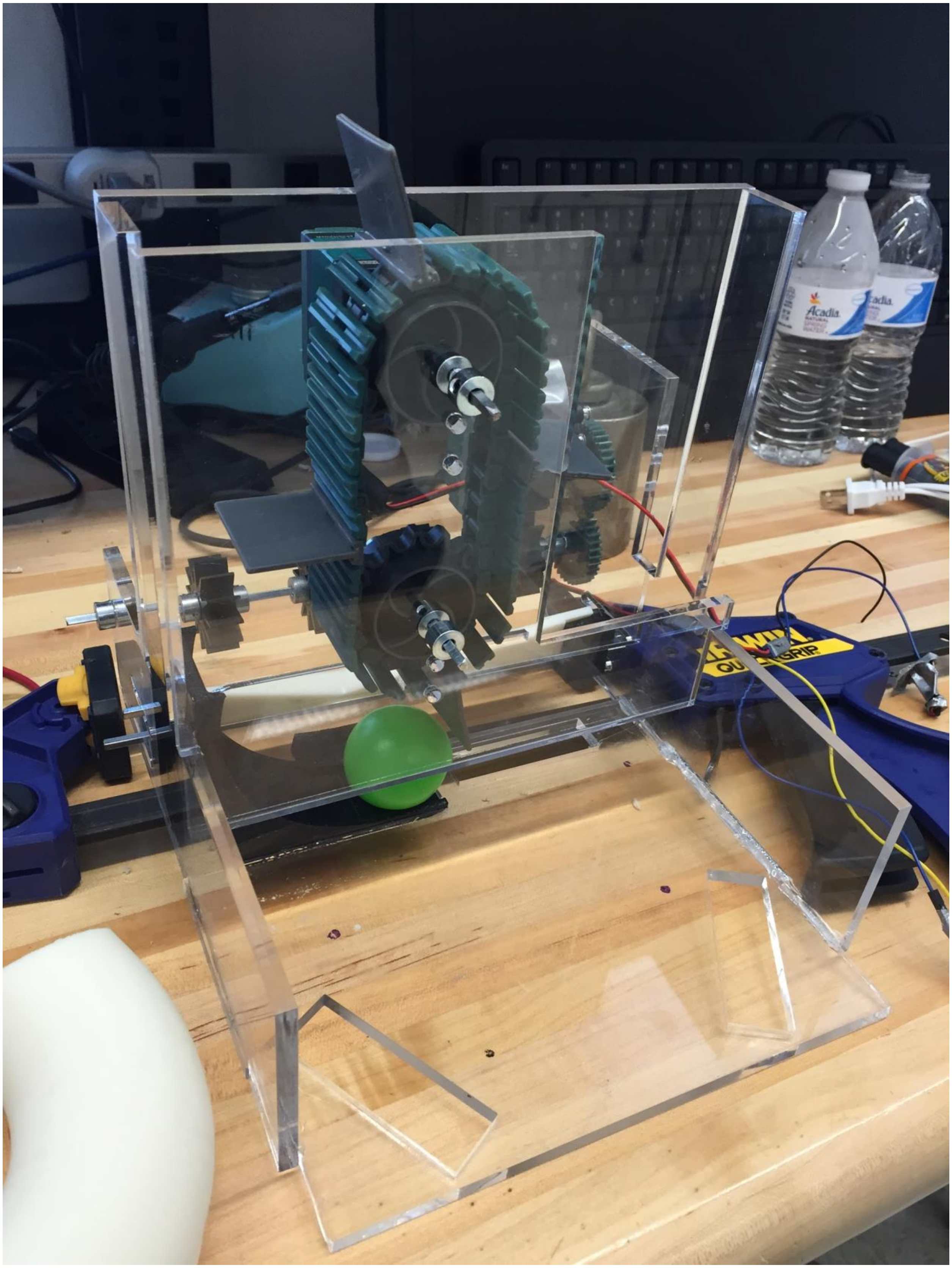

Photo 8: Laser cut acrylic frame and 3D printed ramp

One of the challenging parts of the design was the scoring mechanism. We wanted to maximize our use of the allotted space given to us, as aforementioned, and thus we wanted to have a scoring pipe that used a servo motor to store as many balls as possible before releasing them, so we could store a large number of balls, and we wanted to ensure that no ball would get stuck in the scoring system. With this in mind we discussed using a hopper to store the balls with a single angled chute to score, a helical pipe to store and score the balls, or a simple chamber with a door that would release the balls out the back of the robot. We decided upon the helical pipe because we felt that it would have the smallest chance of having balls get stuck. We also designed a sorting mechanism at the top of the pipe, where a small section would be connected to a servo and would let good (green) balls into the pipe and let bad (blue) balls fall down and out of the robot. To differentiate between the good and bad balls we would use a ColorPAL sensor.

Photo 9: Ball scoring chute, CAD render

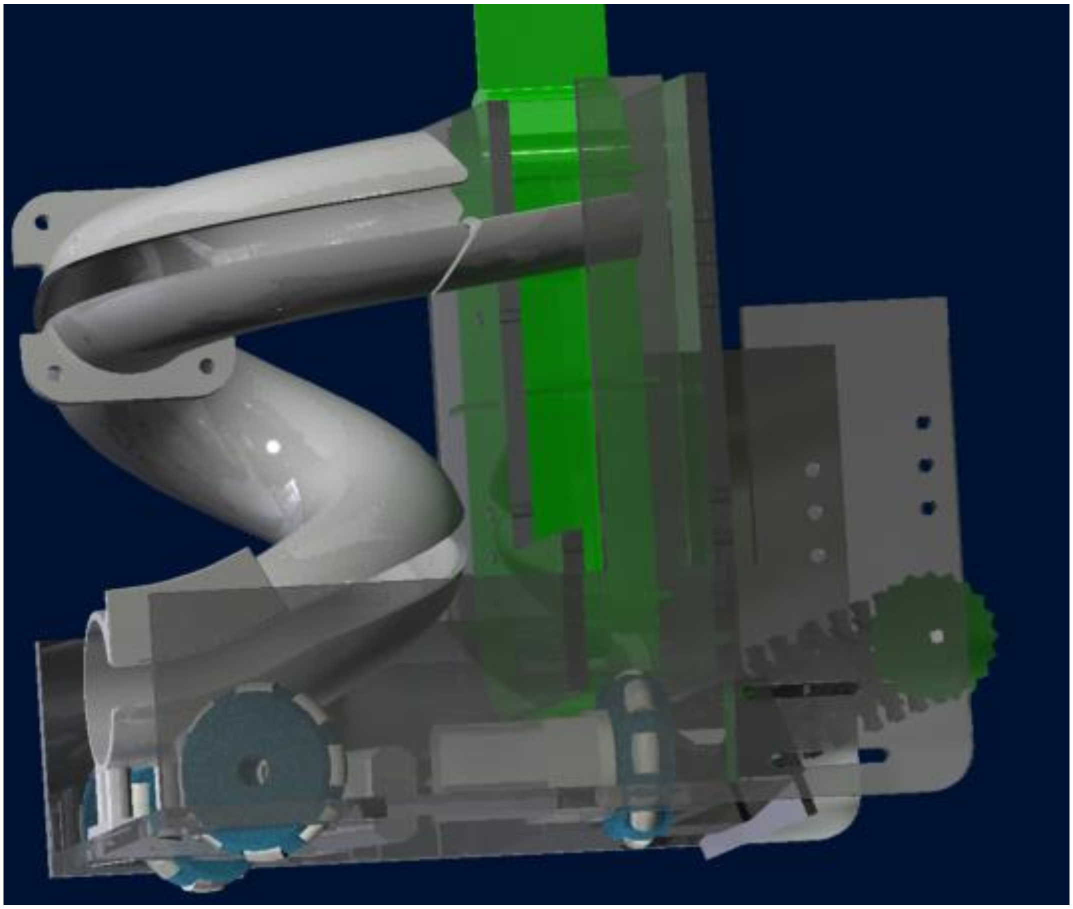
Photo 10: Full ball robot CAD render

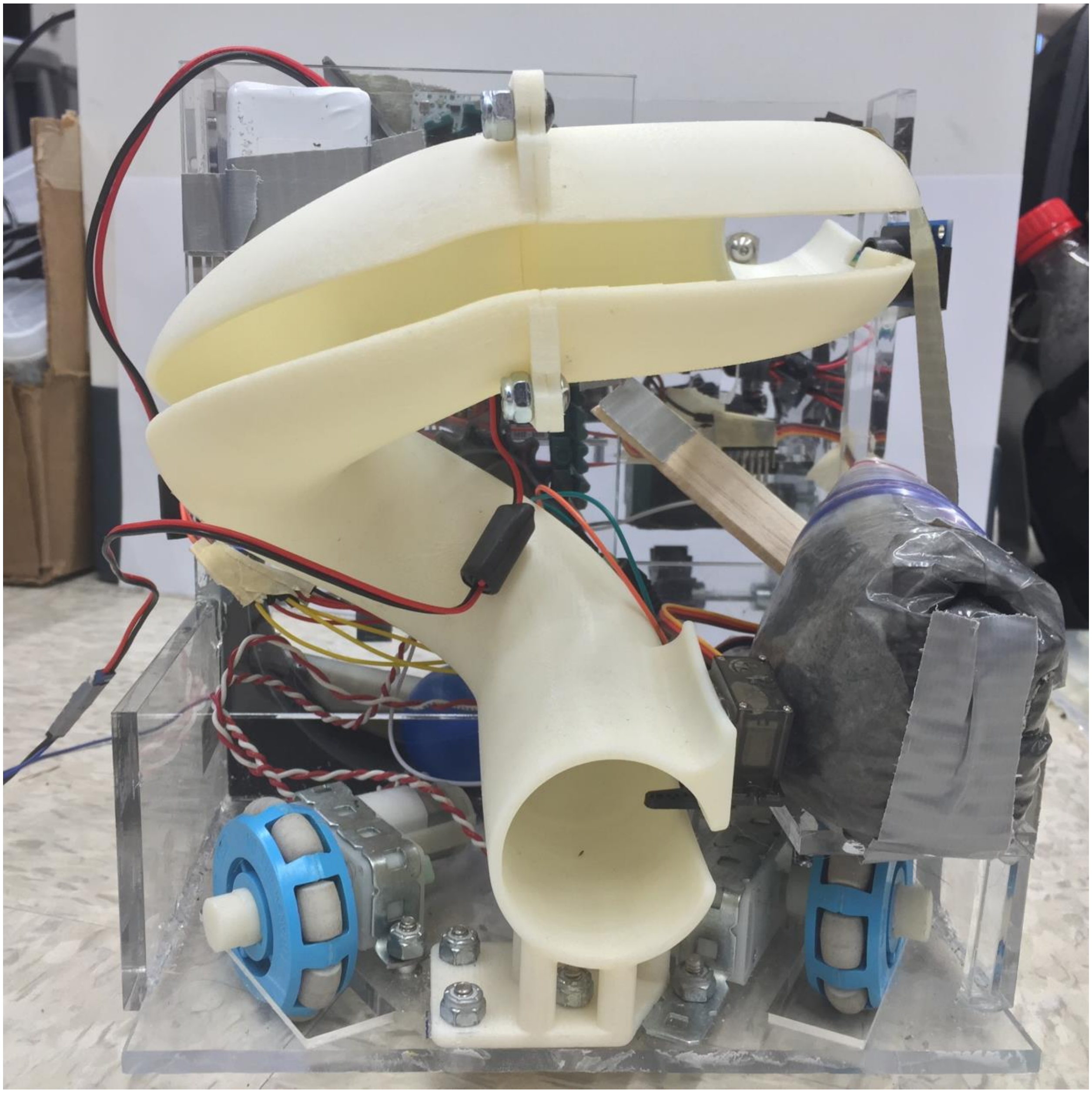

Photo 11: 3D printed ball chute on robot

Another significant roadblock we encountered was the necessity of constantly querying information from the ColorPAL in order to get accurate readings - however, implementing a constant query-feedback loop with the ColorPAL from the main microcontroller made it near impossible to maintain control of other critical elements of our system, such as the servos which needed to respond to the ColorPAL. This was obviously a significant problem since our design depended on reliably controlling two servos to sort and score. To solve this problem, we added an Arduino Micro whose only job was to read data from the ColorPAL and send this information to our main Arduino Mega so it could operate the servos independently. Unfortunately, with the addition of another Arduino came the problem of powering it. We spent hours trying to figure out why the Micro would work properly when plugged into the computer via USB but not on battery power like the Mega. We finally discovered that the Micro is very specific about the voltage supply that it uses, and that we should be supplying it with a steady 5 volts instead of the ~7.5 we were getting from the battery, and that we would have to use a voltage regulator to solve this problem.

We also made small modifications to the ball-intake system as we went along. One minor issue we encountered was that the balls would sometimes slip on the plastic ramp, instead of rolling up, which meant that those balls would be stuck at the front of our robot occupying the space where we might collect new balls. To solve this issue we epoxied 220 grit sandpaper to the surface of the ramp and inside faces of the acrylic walls to ensure that the balls would roll up the ramp and into the conveyor region.

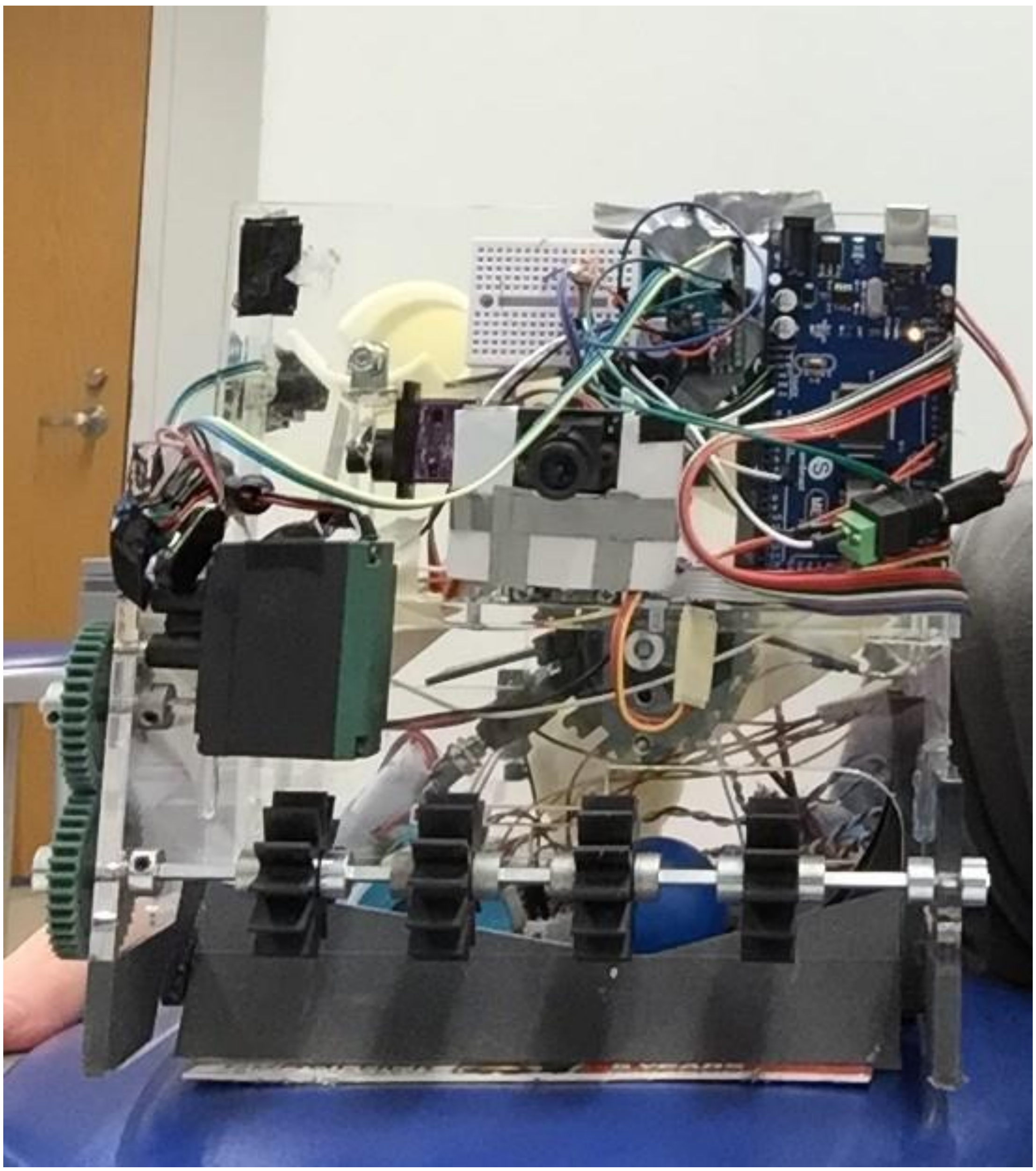

Photo 10: Sandpaper added to intake system to increase friction

Overall, our robots achieved the objectives very well. Our pipe robot consistently navigated the pipe course in under 2 seconds and our ball bot collected, sorted, and almost always scored at least 2 balls. Our pipe robot was consistent because we carefully accounted for the likelihood of flipping over by designing a way of preventing it all together. In addition, we reduced the sliding friction of the body of the pipe racer by minimizing the points of contact to increase its speed. The ball robot consistently collected and sorted balls because the mechanical aspects of the design were very robust once they had been optimized. Once the robot was programmed and functioning properly, we were able to achieve a high probability of scoring by optimizing the distance and orientation to the goal. The only issue we encountered during the competition was that of accidentally damaging a ball. This was due to the height of our intake rollers, which allowed the ping pong balls to go directly under a single roller, instead of between them as originally intended, which sometimes dented the ball.

Throughout the project, we learned a lot about each of the components that went into the robots. The intake paddles, for instance, were very tricky to get right. We quickly learned that the spacing and height of the paddles was the key to their success. We had to carefully adjust the rollers' positions until the sub-system was reliable.

In addition, we realized how important reliable batteries and power supply are to our sensors and microprocessors. Without sufficient power supply to the micro controllers or pixy camera, the robot would sporadically fail to operate or the pixy would "forget" the color signatures that we had assigned or would think that other objects (such as the wall of the court) were one of the assigned colors, which led to the robot attempting to score against a wall. Also, if the motors were not supplied with consistent power, the time it would take for each of the movement operations would increase and the coded path planning algorithms would not function properly because we would not move or rotate far enough.

Lastly, we realized while optimizing our path planning algorithm that it was very important to maintain consistent wheel friction. Since the floor was uneven and smooth, we had to add a significant amount of weight to the robot to ensure that the three omni-wheels would always make contact with the ground and not slip while traversing the court. If the wheels slipped or spun freely, the robot would not be able to reliably capture balls, align with the goal, or score. We also added positional feedback from the pixy camera into the path planning algorithm to ensure that the robot was correcting for small movement inaccuracies while approaching the goal and balls.

We learned many lessons as a team. Each and everyone of us learned a lot from each other. There were many various team and technical issues that we had and learned from as well. For example, soldering is a pain when the tip is burnt and there is only lead free solder to use! Soldering became a much easier process once new tips were placed on the soldering irons. With so many wires for all of our sensors and actuators, we learned quickly we needed to plan most efficient way to wire things up so it takes up least amount of space and is neatest. This was particularly important when wires were near or attached to moving parts of our robot. The use of flexible wires for our breadboards or Arduino pins was a mush so that wires did not easily get pulled out when the robot was moving or being adjusted.

One of the biggest lessons our team learned was to never underestimate the amount of time it will take to figure something out/get something done. For instance, the 3D printing/laser cutting always had lines and limited accessibility. We also should have ordered our parts earlier so that we did not have to wait around as long as we did. Also, ordering spares can prevent a lot of wasted time and struggles. This especially hit us hard when the Pixy seemed to burn out on us right before our second mini challenge.

Integrating the hardware with the software was more strugglesome than we imagined in that the code didn’t run as straightforwardly as we expected it to. We learned that we needed to account for things like friction on the wheels and the inconsistency of many of the sensors we were using. We did not expect to have to do this, but by making our code more robust and simply not using some of the sensors that were giving inaccurate readings, we were able to make our robot perform more consistently. All in all, dividing the responsibilities between members of the group as to prevent anyone from sitting around waiting for another member to finish their work was a very valuable lesson and one that we implemented well by the end of the semester.