Engineering Notebook

920B

Team Number

For Zero Dollars

Team Name

San Luis Obispo High School

School

VEX ROBOTICS

REC FOUNDATION

Joseph Lipper, Hunter Dorf, Christopher Kaml

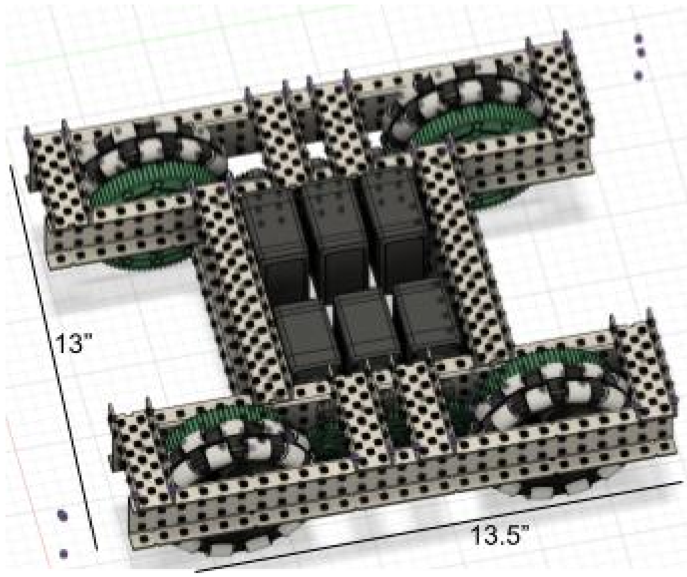
November 21, 2022
Planning Phase

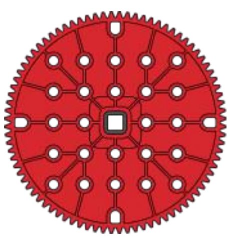
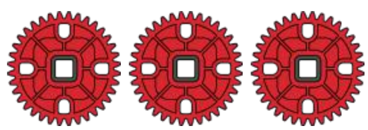
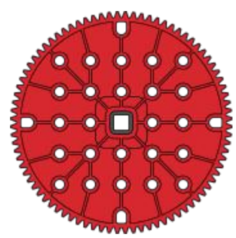
We started off our planning by deciding our goals. That quickly turned out to be a robot that could do everything (launch discs at the high goal, spin the rollers, and expand for points) to maximize our chances of competing at the World Championships. Our next step was choosing how we planned to launch the discs. Because it appeared that teams were mostly opting for a flywheel or a catapult, we had to weigh the pros and cons of each.

	Pros	Cons
Flywheel	• Easier to adjust the angle of the launch • Discs can be controlled individually	• Could require 2 motors • Requires pneumatics
Catapult	• Can shoot 3 discs at a time • Only requires one motor for power	• Doesn't make good use of the limited space • Less control over the discs

After some careful consideration, we decided that the benefits of using a flywheel had more value than the disadvantages. The next issue that we wanted to address was the plastic pipes that held up the net.

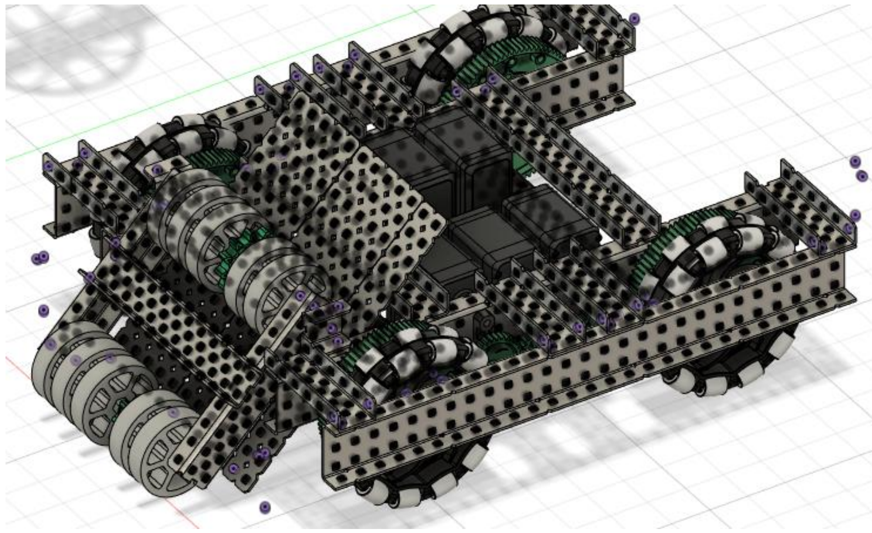
After analyzing videos of early-season matches, we found that robots can hit the pipes and stop their motion. Because every second is important, we concluded that our robot would have to be compact to circumvent this potential issue.



Driven Gear (84 Teeth)	Driving Gears (36 Teeth)	Driven Gear (84 Teeth)
		

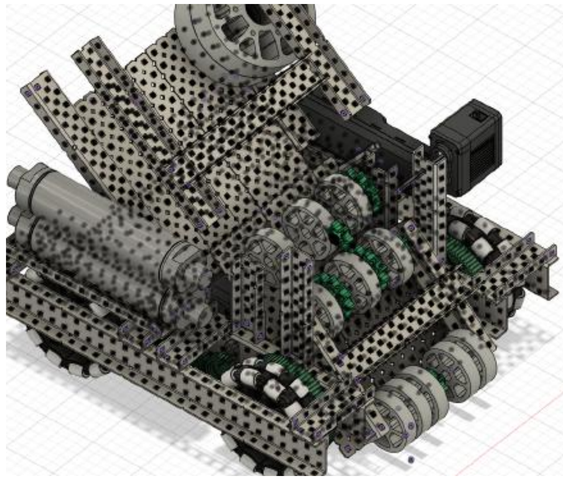
The drivetrain was geared with a ratio of $2.\overline{3}$ using motors that contained 600 RPM cartridges. Excluding friction, the wheels should be spinning at 1400 RPM. The c-channels across the drivetrain are there to add structural support.

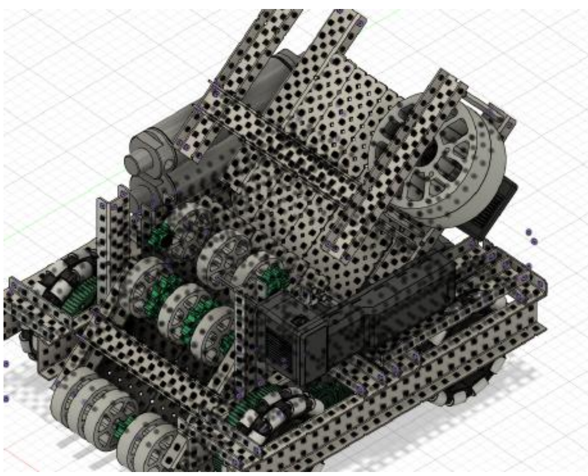
November 23, 2022
Design Phase



We added a disc intake to the three-dimensional model of our robot. The reason why we chose flex wheels was that they compress (which should prevent jamming), and they grip the discs well. The design shows metal plating for the intake, but we plan on using plastic to reduce friction (the software we used didn't have a library for plastic parts). The sprocket is there so that all of the flex wheels can spin, and so we can add a roller mechanism to it in the future.

December 1, 2022
Design Phase

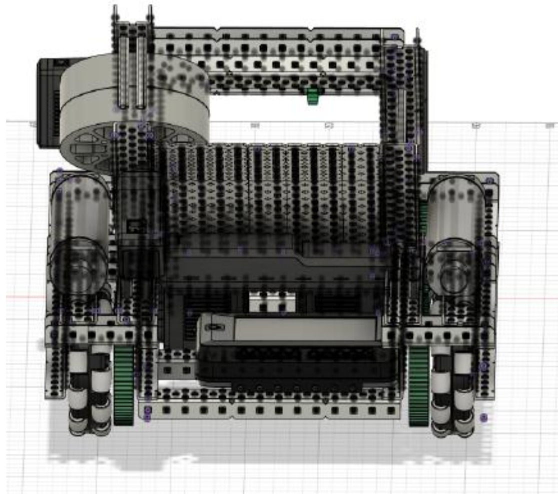


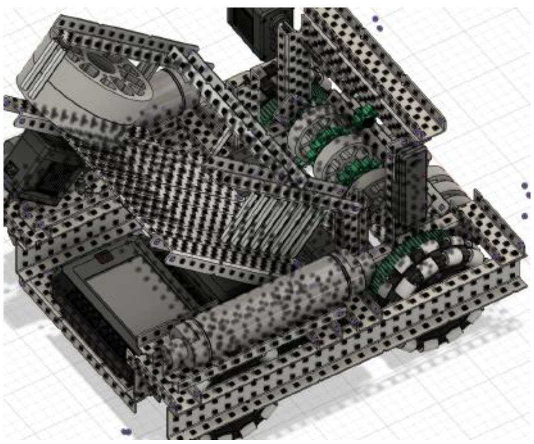


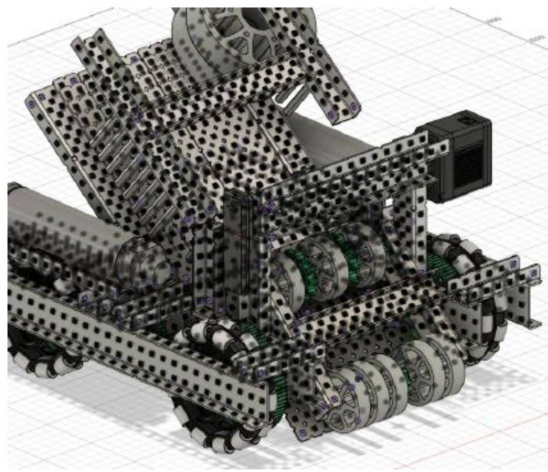
We added most of our functional parts to the design: an intake motor, a flywheel motor, two pneumatic cylinders, the battery placement, the brain placement, the radio, and more sprockets on the intakes. The only functional parts remaining are a pneumatic cylinder and a roller mechanism.

The intake uses a compound sprocket ratio to power the movement of the flex wheels. The wheels on the shaft directly connected to the motor should be spinning at 600 RPM. The middle row should spin at 1200 RPM because it is chained up from twelve teeth to six teeth. The flex wheels at the base should spin at 2400 RPM because there was another sprocket overdrive with a 1:2 ratio.

December 2, 2022
Design Phase

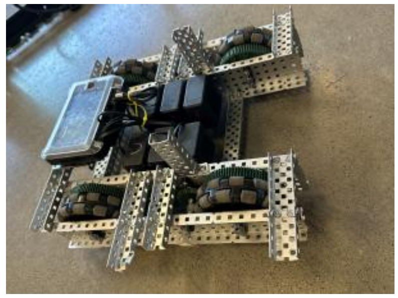
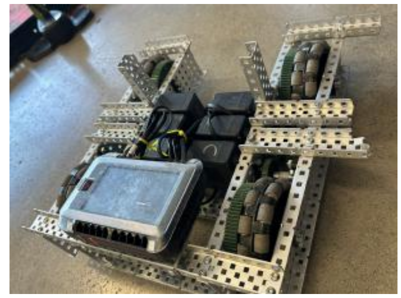






Due to the time limitation that we had today, only a few changes were made. We placed the pneumatic cylinders on opposing sides for balance and ended up centering the battery beneath the flywheel. Standoffs were also added so that discs would move consistently within the indexer and flywheel. We believe that the positioning of the standoffs will have to be changed because early-season experimenting showed us that the amount of compression between the flywheel and the opposing wall has a large impact on the direction and power of the shot.

December 5, 2022
Build Phase

The drivetrain was fully assembled for testing. We added the following code to control our drive:

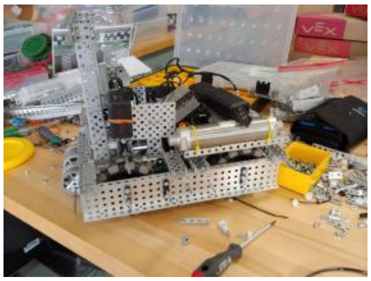
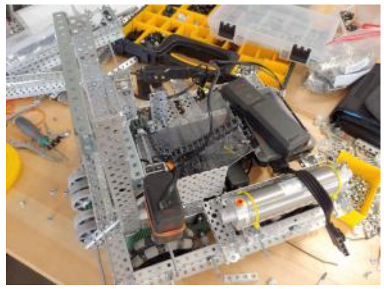
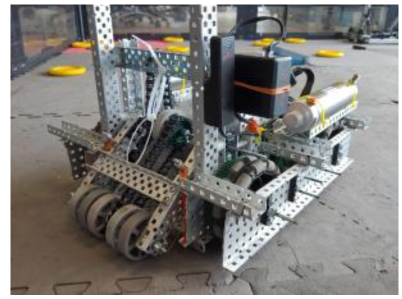
```
def teleopDrive():
    a1 = controller_1.axis1.position()
    a3 = controller_1.axis3.position()
    leftVelo = (a1 * abs(a1) / 100) + (a3 * abs(a3) / 100)
    rightVelo = -(a1* abs(a1) / 100) + (a3 * abs(a3) / 100)
    drive_leftFront.spin(FORWARD, leftVelo, PERCENT)
    drive_leftMiddle.spin(FORWARD, leftVelo, PERCENT)
    drive_leftRear.spin(FORWARD, leftVelo, PERCENT)
    drive_rightFront.spin(FORWARD, rightVelo, PERCENT)
    drive_rightMiddle.spin(FORWARD, rightVelo, PERCENT)
    drive_rightRear.spin(FORWARD, rightVelo, PERCENT)

controller_1.axis1.changed(teleopDrive)
controller_1.axis3.changed(teleopDrive)
```

After driving around for a bit, we found out that the robot didn't drive straight. Testing showed that the right side of our robot had an excessive amount of friction. We had to try multiple solutions:

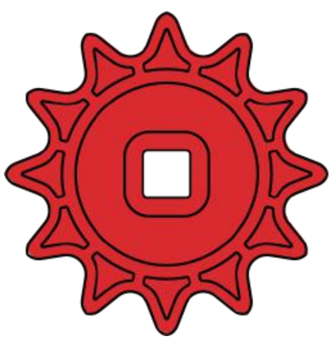
1. ~~Replace all of the shafts~~
2. ~~Align the bearing flats~~
3. ~~Readjust the c-channels so that all of the holes lined up correctly~~
4. Check the spacing between the wheels, gears, and metal

It turned out that gears were constantly rubbing against the metal, so we adjusted the spacers to prevent it from happening again.

Today we started experimenting with our intake. We knew from past years of experience that there was a good chance that the design in the CAD wouldn't work perfectly, but we figured it was a good starting point.

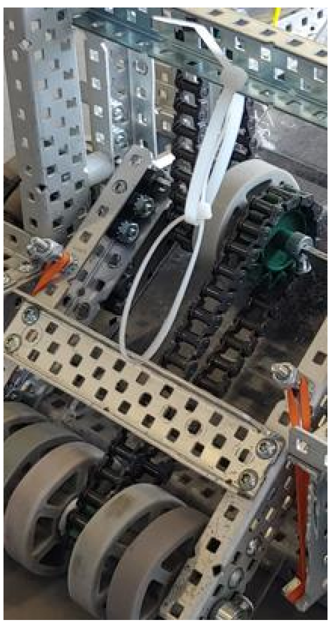
The idea behind the intake was two 600 RPM rollers with 2" flex wheels. To pick up the disc off the ground we are using the conventional "knife edge" lexan piece. It is a piece of lexan secured super close to the ground so that when a disc goes into it, the disc gets scooped up off the ground.

Driving Sprocket	Driven Sprocket
	

For the flex wheel rollers, we initially started with two statically positioned rollers powered by one 600 RPM motor. However, we soon realized that the bottom roller wasn't picking up the discs very well. It was inconsistent and took a lot of alignment, but the main issue was that it needed to be at a shallow angle to intake the disc quickly. This was a problem because we were trying to build a compact robot, and we only had about 5” of space for the intake to bring the disc to the top, and with the current angle, we were still about 2” short from the top of the intake. So, we analyzed the situation and came up with a few options:

1. ~~Shift around the geometry of the first roller some more~~
2. ~~Use a different size/durometer of flex wheel~~
3. ~~Use a different number of flex wheels~~
4. Try a different design for the first roller

We figured that several of these options were fairly simple and quick to test, and they had the potential to make the intake work, so we’d see what we could do with the first three options.



We spent a good hour and a half experimenting with different sizes, positions, numbers, and durometers of the flex wheels. However, we didn’t have much luck. We were able to make the intake work a little better and it brought the disc a little higher, but still not high enough to feed it into the indexer. We needed an intake that could get the disc from flat on the ground to a steep angle to stay within the 5” allotted space.

We’d already exhausted all but one of the choices on our list, so we decided to try that one. We started by researching on YouTube and looked at what kind of intakes the winning teams used. We figured if it was good enough for teams who won signature events, it was good enough for us too.

One common theme we noticed among all of the top intakes was that the first roller wasn’t static like ours; it was able to sort of move up and down. We did some more research, mainly on the VEX Forum, and found that it was called a “floating intake”. We found out that it was initially created by team 9457B within the first week after the game release.

We knew the floating intake would take a good deal of refinement, as is the case with all intakes. We ended up putting some c-channels together and bearing flats in a position that allowed us to move around the parts if we needed to change anything. After some testing, we noticed that the longer the distance between the knife edge and the floating intake roller, the better the intake picked it up. This did have the unfortunate downside of a longer intake since it needed to grab the disc from further out.

We did notice a trend in our testing of some things that tended to make the intake work better. It seemed that the shallower of an angle that the lexan was, the easier the intake picked up the discs. Also, we noticed that spinning the bottom intake roller at 1200 RPM seemed to give us the best results, as it seemed to be a good balance between speed and torque.

We needed a way to combine the shallow lexan intake angle with still having it steep enough to get all the way to the top of the intake in just 5”. We decided to research how other teams were able to intake at such steep angles.

After watching some reveal videos, we realized that teams bent the lexan partway up the intake, so they could get a shallow angle at the bottom, but then steepen it as it went up. This got the benefits of both a shallow and steep intake, while avoiding the downsides for either one. It had a nice shallow knife edge to pick up the disc, and a steep angle to bring it to the top in just 5”.

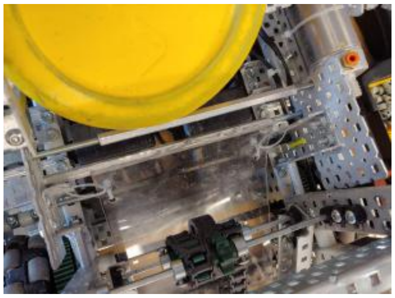
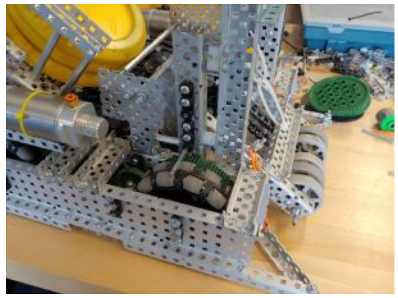
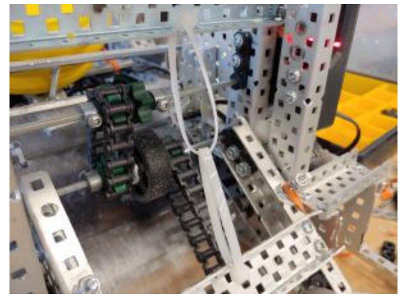
```
def forwardIntake():
   intake.spin(FORWARD)

def reverseIntake():
   intake.spin(REVERSE)

def stopIntake():
   intake.stop()

controller_1.buttonL1.pressed(forwardIntake)
controller_1.buttonL2.pressed(reverseIntake)
controller_1.buttonL2.released(stopIntake)
```

December 13, 2022
Build Phase

As we drove the robot around the field, picking up discs, we noticed another problem, the discs still weren't quite coming out of the top of the intake high enough. It came up about a half-inch short. This made it so that we couldn't fit three discs in the indexer. We could only fit two, and the third one would have to stay in the intake until we were ready to shoot. Then, we would run the intake while shooting and get a smooth three-in-a-row.

We added a standoff to simulate the height we'd need for the intake to be able to stack 3 discs in the indexer until we had finalized the intake design and were able to cut a piece of lexan for the final revision. We also added zip-ties on the front as an adjustable hard stop so we can fine-tune it and then bolt it in place once we found a good distance.

However, we quickly realized that this was not ideal for a competition as it gave the driver more to think about when controlling the robot and maneuvering around defense. So, we decided on a few potential options: we could just deal with it, we could add another intake roller to give the intake additional height, or we could try to adjust the current roller positions for the intake so that it could get the height with just two rollers. We went for the second option because it seemed like the most robust solution. We were also running out of the time we'd allotted to work on the intake and we needed to move on to the other subsystems. The main potential downside to this solution that we saw was that it could introduce some extra friction into the system; however, our intake wasn't in danger of having too much friction as the motor didn't overheat for about 15 minutes, so we decided that wasn't an issue worth worrying about.

After adding the third roller, it surprisingly still wasn't enough. The disc got to the top of the intake just fine, but it still had a hard time sometimes getting the last little push it needed to get into the indexer. To try and give it that extra nudge, we made the top/third roller out of flaps so it reached out further to push the discs better. It also had extra compliance and still worked when propelling the disc up the intake. We found that one flap worked best so the disc would go fully back against the roller and the other flap would come around and jump it into the indexer. However, there was a major flaw in this design: the flap hit a nearby piece of metal and consequently caused a good amount of friction. The metal was structural, so we couldn't move it or cut a big enough hole in it for the flap to go through. However, it wasn't a huge problem, as the motor still took ten minutes of continuous use to overheat and the intake still worked well (and, more importantly, got the discs into the indexer). That's why we decided to keep it as is, and would only change it only if it became a major problem or a major source of friction.

December 16, 2022
Build Phase

We finally moved on to the flywheel design after perfecting the intake design to our preference and making a prototype-level indexer.

Our goal with the flywheel was to be able to shoot three discs in under two seconds, and consistently make all three in the goal from close range, and at least two out of three from long range.

To accomplish this goal, we realized that we needed an angle adjuster for long-range (autonomous shots) and short-range (driver control shots), or one angle that could do both pretty well. We agreed to do a static angle because we realized that we didn't have room on the end of our robot for a protected angle adjuster.

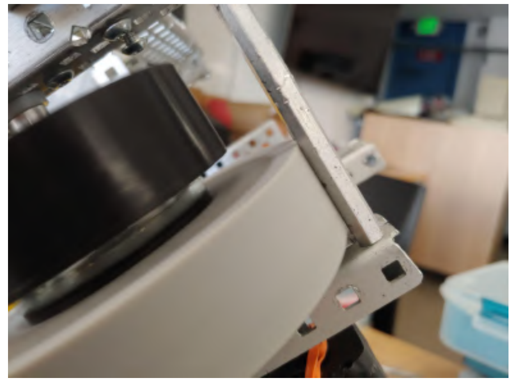
To precisely tune our flywheel angle, we made it easily adjustable. We did this by using standoffs with the extenders going into a shaft collar, so we could adjust the flywheel angle to any angle.

The initial design for the indexer and flywheel was at a 45-degree angle, so we decided to start with that (and adjust it later if necessary). Next, we needed to build up a mock flywheel to test the angle and adjust it to the correct angle if it was incorrect. We quickly built a flywheel and ran into a major problem. When we increased the flywheel speed past a certain point, it wouldn't go any faster. The motor also was overheating a lot faster than we expected. However, we didn't know for sure how fast it should overheat since there was only one motor to carry the entire load.

This lead us to believe that something was wrong with our flywheel. We decided to look deeper into it. Upon closer inspection, the large 4" diameter flywheel expanded when it spun up, something we hadn't taken into account when we were 3D modeling the robot. The flywheel expanded enough to hit the supporting standoff in the flywheel superstructure. This caused a large amount of friction, which made the flywheel not able to spin faster once it started rubbing against the standoffs (see photo below).

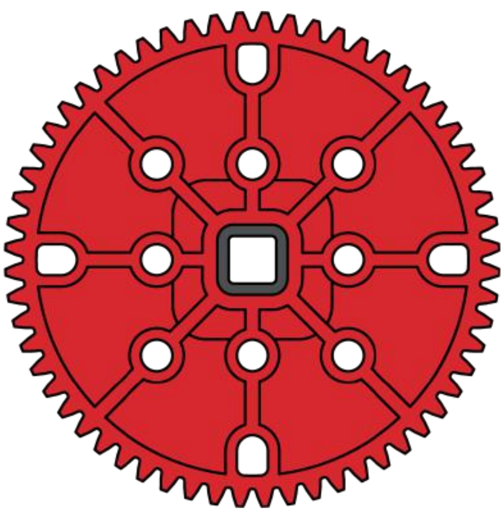
We thought about it and decided we could either use a smaller diameter flywheel, try using a higher durometer so it would expand less, or move the standoffs out of the way.

We chose to move the standoffs since we wanted to stay as close to the CAD as possible, and the CAD used a 30A 4” flex wheel. Changing the position of the flywheel would majorly affect the use of the indexer and would likely mess up the compression between the flywheel and the other side of the shooter. Changing the durometer would also potentially affect the compression and make that not work as intended in the CAD. So, we ended up changing the position of the standoffs. This seemed to fix the issue, and we continued tuning the flywheel.



After shooting just over two hundred discs, we determined that our initial calculations in the CAD were incorrect and that the optimal angle for the flywheel was 46 degrees. This was determined through testing of long-range, short-range, and just around the outside of the low goal. Through our game analysis earlier in the season, we discovered this was the best place to shoot from, especially under heavy defense.

Using a 600 RPM cartridge, the flywheel spins at 3000 RPM because of a 6:1 gear ratio.

Driving Gear	Driven Gear
 60 Tooth Gear	 12 Tooth Pinion

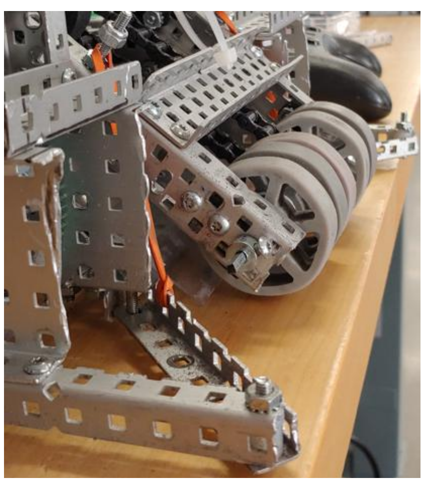
We used a “Bang-Bang” control loop in the code (which sets a high voltage at low speeds, and a low voltage at high speeds). This allows the flywheel to have fast spin-up times while still maintaining speed after a shot.

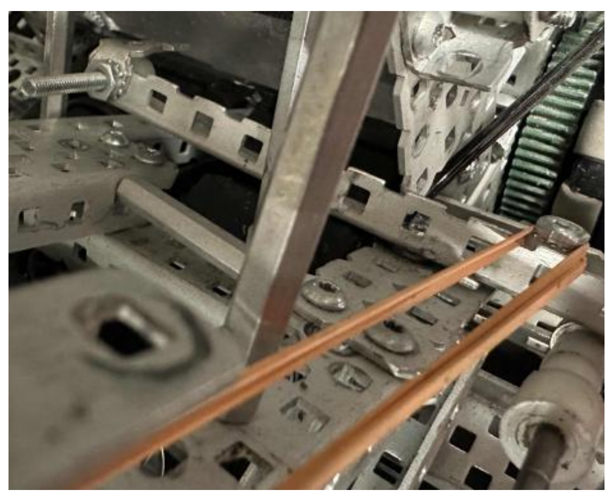
```
while (flywheelEnabled):
    if (flywheel.velocity(RPM) < target):
        flywheel.spin(FORWARD,12, VOLT)
    else:
        flywheel.spin(FORWARD, 5, VOLT)
    wait(5, MSEC)
flywheel.stop(COAST)
```

December 17, 2022
Build Phase

We drove the robot around to stress test the intake and found one major flaw: it stuck out of the front of the robot. This caused a few major potential problems, including the intake caught in other robots, bending the bottom shaft, and making it overall awkward to drive and pick up discs.

We brainstormed and came up with two choices. We could either redesign the intake or make the front of the robot longer. We decided to make the robot longer since we'd just gotten the intake to work super well and we didn't want to mess with it in case it stopped working. Also, we thought back to our reasoning for making a small robot, and remembered that it was because we wanted to be easily maneuverable around defense. To still achieve that outcome, we needed to maintain a skinny robot, but the length didn't matter nearly as much as the width. We initially started putting just straight bars out of each side of the robot to make the bot longer, but after we drove around we had the idea that we could double purpose making the robot longer by adding little horizontal ramps on the front of the robot. These helped center the disc, making it a lot easier to drive as the intake was now effectively the entire front side of the robot because it centered itself into the intake when you drove into it. After adding these, the robot drove much better and didn't get the intake caught in things. We tried it under defense from another robot, and it held up well.





The next major mechanism on the robot still left to design was the indexer. The indexer is the mechanism that holds the discs after going through the intake and feeds them into the flywheel. We decided on a pneumatic indexer as we had no more motors left, and we didn't see any major downsides to it. We briefly researched the topic and found some interesting designs on pneumatic indexers. We used the basic idea from many of the teams we saw, but we added our little twist to it to make it work for our robot. We put the pneumatic cylinder on the side, like in the CAD, and had it extend to actuate the little arm that pushed the bottom disc into the flywheel. It uses a lever and is the common mechanism that we saw many other high-performing teams using.

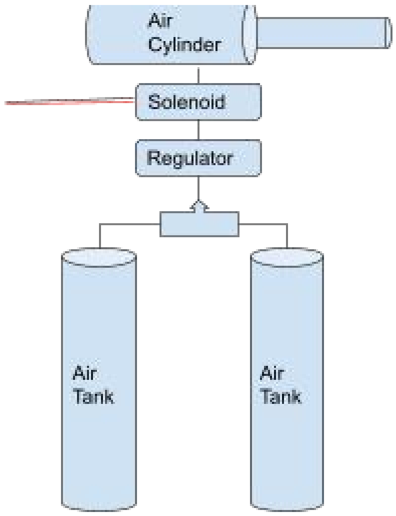
However, when we tried it out for the first time, we ran into a problem; the pneumatic indexer didn't reach far enough to push the disc in the flywheel. We solved this by adding a little screw on the end of the lever to extend its reach enough to cause contact with the flywheel. It had plenty of room for this as it went back into the robot pretty far because the cylinder had such a long throw.

Now, the indexer worked well, but there was still a problem with it; when doing more than one disc in a stack, the disc on top of the one that was currently being shot would get caught on the indexer sometimes. This would make the indexer not be able to retract all the way, preventing you from shooting the next disc. We realized that it was getting caught on the lip of the disc falling on top of it when it was retracting. Once we realized this, the solution was obvious. We just make the top of the indexer flat and extend back past the lip of the discs This makes the disc on the top ride smoothly on top of the indexer until it retracted all the way. Then, and only then, is the disc allowed to come down and prime itself for shooting.

As we were practicing though, we realized that we quickly were running out of air after about 30 shots. We determined this was okay, but not ideal, since we could shoot about 30 shots in a match, and still would need air for when we implemented an endgame mechanism. To solve this, we did a few things. First, we made the air cylinder single-acting instead of double-acting, and rubber banded it back, so it only used air one way. Second, we added a regulator. This solved two issues.

First, it saved us air by only giving out just barely enough to shoot. The second thing was it gave out a consistent amount of air every time. Before having a problem where when the air tanks were full, the disk would shoot further than when they were empty because the indexer would push them out faster. Now, that isn't an issue because the regulator always gives out the same amount of air, so we get the same shot every time (until we run out of air of course, which takes about 60 shots).

Lastly, we added spacers on the piston of the cylinder to limit how much it retracts so it is just past the point where the next disc drops down. This has two benefits. First, it saves air by having the cylinder not actuate as far, and it also saves a little tiny amount of time per shot because the indexer is closer to the disc, so it can shoot it quicker. All in all, we are very happy with this indexer and are excited to move on to the next system.




```
def actuate():
    indexer.set(True)
    wait(75, MSEC)
    indexer.set(False)
```

The roller mechanism was the last major system on our robot. The initial design had one roller in the middle, but we wanted to do some research and see what other top-performing teams were doing at this point in the season. We watched a recent signature event and found out a new roller mechanism meta had emerged. The new meta was to have two wheels on the mechanism, one on each side of the robot. This allows for a much wider area to be able to reach the roller, from any point along the front of the robot. It makes complete sense why this is the superior solution, and we decided to implement it right away.

We started with two 30A 3” flex wheels, one on each side of the robot, and chained it directly from the top intake roller, which was running at 300 RPM. As soon as we tested it on the rollers, we found two very important things. First, that it was way too fast, but second, and most important, that not all the rollers are the same. New rollers are a lot stiffer and harder to turn. However, when we tested it on the older rollers that our sister teams had been using, we discovered that they were much easier to spin. Our roller mechanism worked well on the easy-to-spin rollers, however, it was going too fast to work on the new, harder-to-turn rollers. Because the rollers at competitions would likely be new, we should make our roller mechanism able to work well on any stiffness of rollers. We carefully analyzed the situation and decided there were a lot of variables that could affect how well the roller mechanism worked. We brainstormed and came up with:

1. Wheel durometer
2. Height
3. Wheel size
4. Angular velocity

With so many variables potentially affecting our roller mechanism, we decided to experiment with one at a time. We first decided to slow the roller mechanism down since our research showed that teams were running theirs at less than one hundred rpm. We geared ours to 100 RPM, and it worked a little better on the stiff rollers, but not anywhere close to the way we'd like it to.

Next, we experimented with height. We'd designed our roller mechanism with lots of extra room so that we could adjust the height if necessary. We tried several different hole positions, but height didn't seem to be helping it work better. However, we still had other variables to test. Next, we tried wheel size. We tried the 4" wheels since those seemed to be the most common size for roller mechanisms other than the 3" ones. We compared the two by putting the 4" one on one side of the roller mechanism, and the 3" one on the other side, so we could see side by side how each one performed compared to the other. After messing around with the height some more, we determined the 3" wheel worked better than the 4" one. We briefly tried the 2" flex wheels, but we quickly found out the 2" ones have an absurdly little amount of traction on the rollers, so we quickly took those out of the running. By now we'd determined that we'd started with pretty much the optimal height for the rollers and that the 3" wheels were the best.

The only variables left were the speed and the durometer. We tried both the 45A and 60A 3" flex wheels, but neither of them worked as well as the 30A ones, so the only thing left to change was the speed. Slowing down from 300 RPM to 100 RPM did help a fair amount to make it work better. So we tried slowing it down even more to $66.\overline{6}$ RPM. This seemed like the best combination of all, as it still tuned the easy rollers nicely, but it also got the hard rollers fairly well.

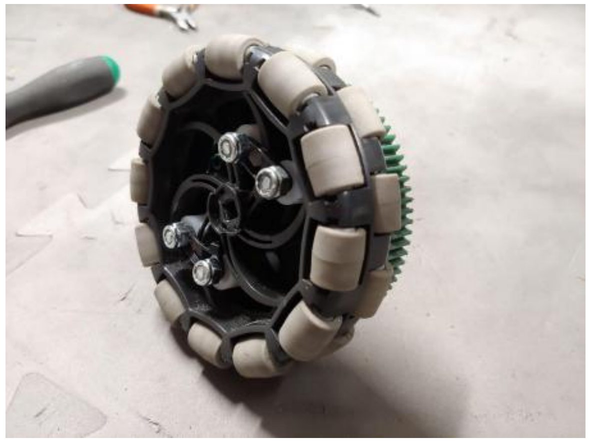
To have high-accuracy movement during autonomous, we wrote a Proportional-Integral-Derivative control loop for our turns.

```
def turnToDegrees(target):
    currentRotation = inertial.rotation(DEGREES)
    error = currentRotation - target
    derivative = 0
    integral = 0
    while (abs(error) > 0.5 or abs(derivative) > 1):
        currentRotation = inertial.rotation(DEGREES)
        error = target - currentRotation
        if (abs(error) < 8.6):
            integral += error
        derivative = -inertial.gyro_rate(AxisType.ZAXIS, VelocityUnits.DPS)
        setDriveVelocity((error * 0.88) + (integral * 0.0008) + (derivative
* 0.033), PERCENT)

        drive_left_front.spin(FORWARD)
        drive_left_middle.spin(FORWARD)
        drive_left_rear.spin(FORWARD)
        drive_right_front.spin(REVERSE)
        drive_right_middle.spin(REVERSE)
        drive_right_rear.spin(REVERSE)
        wait(1, MSEC)
    setDriveVelocity(0, RPM)
```

The code allows the robot to accelerate as it takes large turns, slow down as it reaches the target angle, and then move the slightest bit to correct any potential error that may have occurred during the turn.

December 26, 2022
Build Phase



As we were trying to fine-tune our autonomous functions, we noticed that we weren't getting consistent turns. We had an inertial sensor and were using it, but there were still a few problems. As we investigated the issue, we had our programmer look over the code again, and our builder looked at the robot and see if there was anything mechanical that could be causing something like what we were seeing. It turns out it was a mechanical issue. Our builder noticed that there was a lot of play between how far the drive wheels spun, and how much the gears were turning. This made sense as the issue we were having was that it wouldn't execute the same turn the same every time, which made tuning it impossible. The large amount of slop between the shaft and the wheel on the drive was negatively game-changing.

We decided that we needed to develop a way to attach the drive wheels directly to the gear, rather than on the shaft. Ideally, we would have used VEX's new Omni wheels that have mounting holes in them, but we didn't have any of those, and they were back ordered for several months at least. So, we needed to find another solution. We looked carefully at the holes in the gears and tried to see if they could potentially line up somehow with the curved supports in the wheel. After switching out standoffs, spacers, zip-ties, and nuts, we figured out the best possible solution. A screw in each side of a spoke, with large spacers on it with a zip-tie on the end to prevent the screws from bending outwards, with nuts on the end, would effectively clamp the wheel in place, centered on the gear. We did this to all four drive wheels and tried tuning our turns again, and got much better results, thus confirming that the slop was indeed the issue.

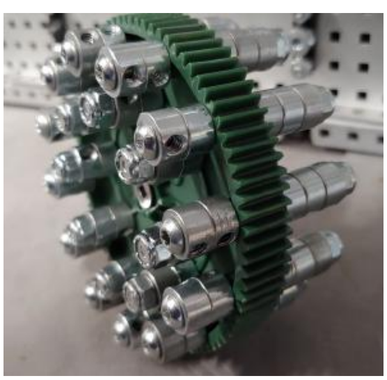
As we were practicing and scrimmaging under defense with other teams in our organization, we noticed that our shooter was slow, and took a fair amount of time to spin up again after each disc is shot. This was not optimal for heavy defense matches because you often only have a second or two to shoot after they back off. We brought the robot back to the pit and took a look at what was going on. It seemed that we had the wrong amount of weight on the flywheel. We researched and came up with four possible solutions to the issue:

1. Add more weight
2. ~~Take away some weight~~
3. ~~Add a kicker wheel~~
4. ~~Lower gear ratio~~

As we discussed the options, we determined that option three would not fit with the current design because we didn't have room for a kicker flywheel behind our main flywheel. We then imagined a lower gear ratio. However we realized this would limit our ability to consistently make long-range shots, so we ruled out that idea. We then had two ideas left: adding and subtracting weight. We first tried adding weight, in the form of a second 4" flex wheel stacked on top of the first one solely for weight to help prevent RPM drop in the flywheel between shots.

We'd previously had a 3" flex wheel as our additional weight, but after researching how weight and centrifugal force works, and how the proportions work, we determined that the 4" flex wheel would be about twice as effective as the 3" one. This is because weight further out increases the energy stored in it when spinning by a factor of four. However, when we tested it on the field, it still wasn't able to shoot at the speed we'd like, and it took around three seconds for three consecutive shots. We thought about it and wondered if our last idea could help: taking away as much weight as possible. So we took off the second flex wheel and any extra screws, shaft collars, and spacers that weren't necessary for the functionality of the flywheel. This produced similar results to the aggressive weight, and we were able to shoot three discs in about three seconds.

We decided that there had to be a better way. Other teams were getting it to work, so there had to be a secret; something we were missing. We tried as our next step to go back to adding weight, but this time add as much as possible. We looked around for things we could use as weight that could also be securely attached when spinning around 3000 RPM. First, we tried Keps nuts on long bolts, but that was about the same weight as the 4" flex wheel. We considered zip-tying on pneumatic nuts or something of the sort, but we determined that could be a safety concern, so we tried to find a safer solution. Next, we tried shaft

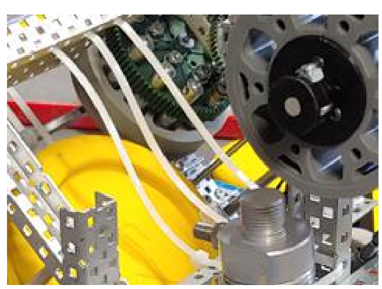


collars. We weighed them and found they were more than twice as heavy as Keps nuts. This was inspiring and we filled up an eighty-four tooth gear with as many shaft collars as we could fit on it. 70 shaft collars later, we had our flywheel weight. We put it on the robot and tested it out. The flywheel took a few seconds to spin up, but that also meant it was storing a huge amount of energy. When we shot three discs with it, it didn't slow down much at all. We'd found our solution.

December 31, 2022
Build Phase

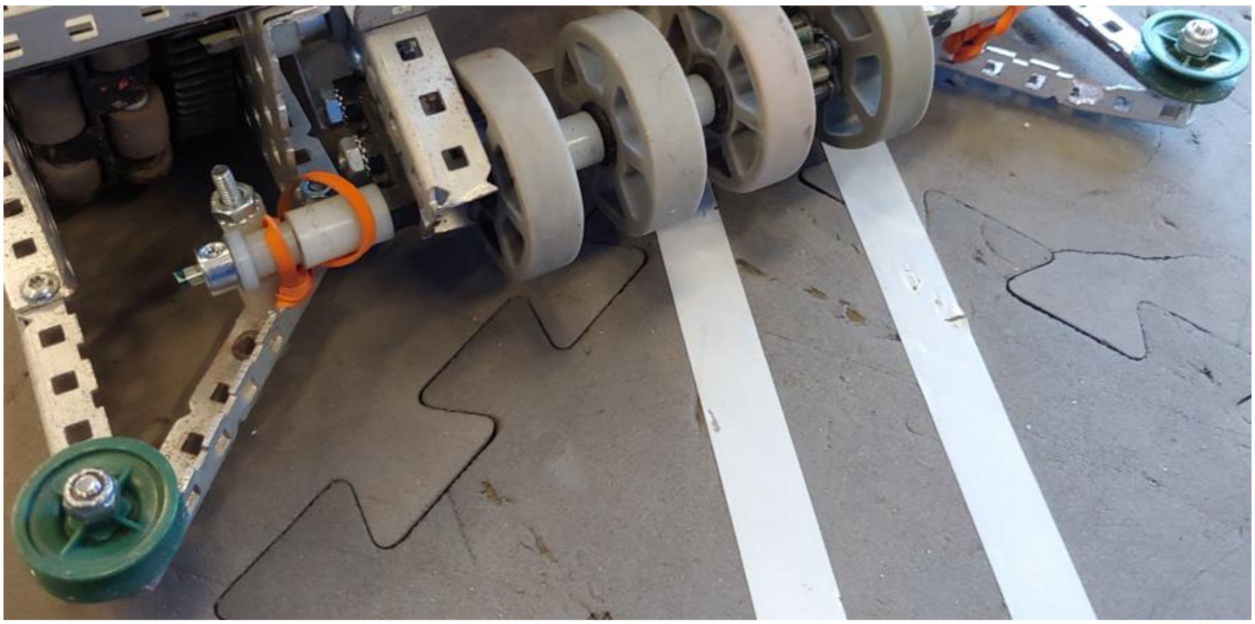
Now that we were nearly done with the robot, we decided we needed to give it a name. We brainstormed different names, but nothing caught our fancy. We needed something that represented us and our robot. Our secondary builder/programmer was the one who came up with the name. A few meetings ago he'd been trying to throw the discs into the high goal, and said "for $100 I'll make this." He'd miss, and then go down to "for $50 I'll make this." After a bunch of misses, he finally said, "for $0 I'll make this." He made it. From then on, whenever we shoot at the goal, we say "for zero dollars," and when we say it, we for some reason make it in the high-goal more often than probability says we should. So then, we had our robot name: "For Zero Dollars". We 3D printed a non-functional decoration to spice up the robot a little bit and then moved on with the build.

As we practiced more and more aggressively under defense, minor issues that we hadn't noticed before (because we were dealing with bigger issues) came to light. One of these was with the indexer. Every once in a while, it would get jammed. We looked into it and tried holding pieces of metal in places to see if that would help. We eventually found out that a few zip-ties across the top helped prevent the issue.



January 9, 2022
Build Phase

Today we mainly worked on the endgame, except for a minor improvement to the intake. For the intake, we noticed that another team was using little rollers on the edges to help them glide along the wall and edge of the low goal better to help them pick up the discs there. We decided we'd try them out, and they worked great. They do break every once in a while if you hit them wrong, but we have a bunch of them, and we are only planning to replace them before the competition to conserve resources.

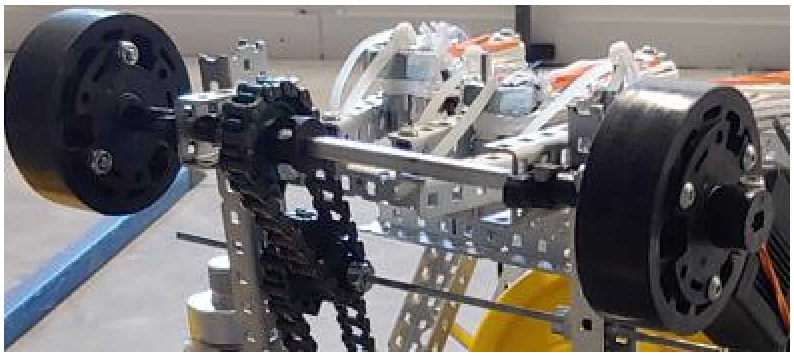


After closely analyzing endgame designs that have scored lots of points, we realized that one mechanism was being used by a lot of teams, and it seemed good. It was a way of doubling up storing the string and holding back the projectile. It was a very innovative design. We also added our own little twist to it. We added separate parts for storing the string and the rubber bands so the two wouldn't get tangled, reducing the chances of it not deploying when we release it because it got tangled in itself.

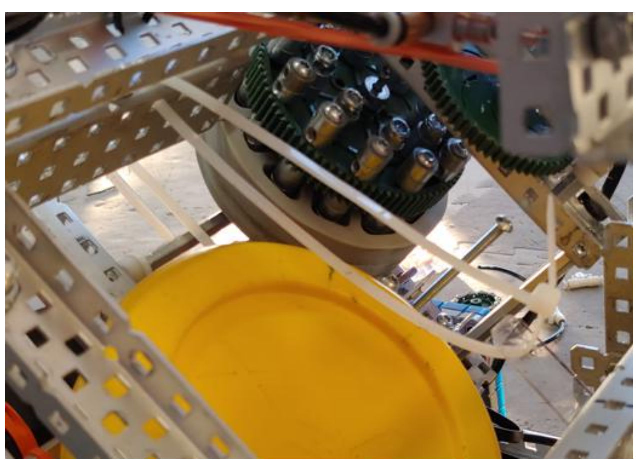
One major problem we had with it is that the strings wouldn't always spread evenly. We tried using different strings and adding zip-ties onto the projectile (pneumatic nuts) to make it not bounce around as much when it hit the ground. But the real magic was when we changed the actual angle of the launchers. Angling the launchers far to the side resulted in a consistently good spread, touching about fifteen tiles with our three strings. We also realized that we needed to be cognizant that there was no chance the strings could go out of the field. We did this by angling the launchers slightly downward. This not only makes the pneumatic nuts not go out of the field because they are launching too high, but the shallow angle we are angling the launchers at makes the nuts not bounce upward much so they can't go out of the field that way. This was a common way we saw teams get disqualified in matches, so we wanted to make sure it wouldn't do that. After our modifications, our endgame seemed to work well, scoring us an additional ~45 points in the last second of the match.

January 12, 2022
Program Phase

Our roller mechanism wasn't working as well as when we first designed it, so we decided it was time to redesign it a little bit and see if we could get it working again. We first tried different durometers of flex wheels since that was the easiest thing to test. We quickly found out that the black 60A wheels worked better than the 45A ones. So, we switched over to the harder durometer ones and kept practicing and coding autonomous routines.

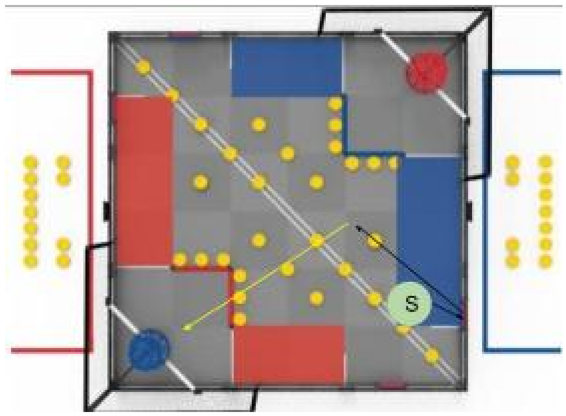


We were also having issues with an edge case with the indexer. The three zip-ties we put on earlier weren't working as well as they were when we first put them on. So we decided to play around with the position and number of zip-ties and even add a piece of lexan for some more rigidity until we could prevent it completely. After a large amount of testing, which is an unfortunate side-effect of edge cases, we determined that a single zip-tie in the center with a lot of play in it along with a small piece of lexan at the top of the intake to help guide the discs into the indexer was the best at preventing the issue.

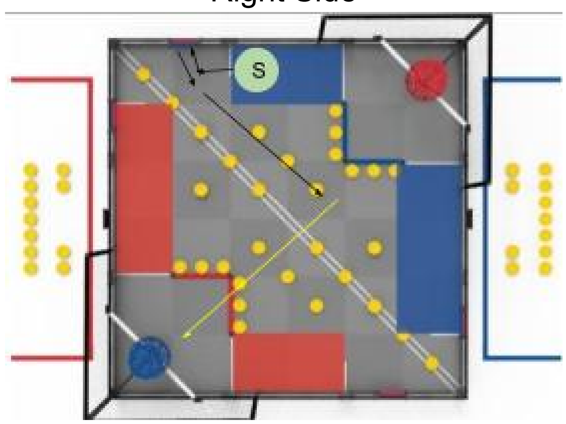


These are the autonomous routes that we are programming for the competition on January 17 (black arrows represent movement, yellow arrows represent shooting exactly three discs, and the “S” is the starting position of the run):

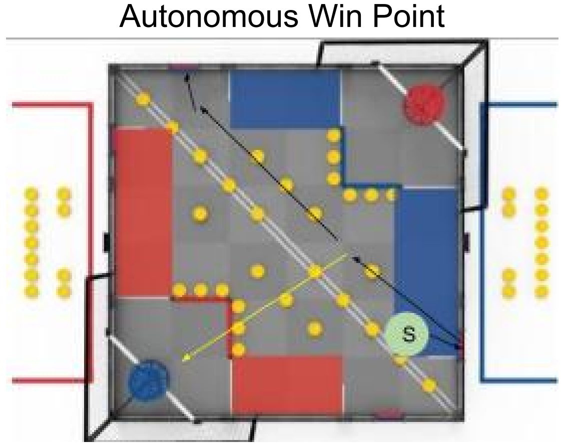
Left Side:



Right Side



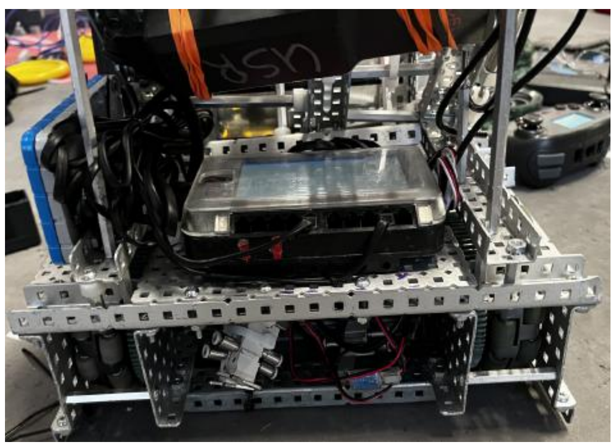
Autonomous Win Point



January 23, 2022
Build Phase

We won as a tournament champion at the competition!

Because the head referee felt that the positioning of our brain was questionable, he asked us to move it before the next competition. We moved it to a location just under the flywheel so that the screen is a lot more visible.



We now had a very competitive robot, proven by our win at the recent competition, and now it was time to work out any minor bugs or problems that hadn’t been an issue, but might cost us just a few seconds in matches. We need to be at the top of our game for our upcoming signature event in February. One issue we were having in the low goal, under defense, was that the high-strength shaft collars (high-strength because we found that low-strength would easily bend) sticking out the end of the roller mechanism would sometimes get a little tangled up in the net, allowing a defender to catch up to us. Once in a while, we would even get stuck on top of a roller. We decided that we had to get rid of the collars sticking out like that, but we needed them to be able to support the wheel. We decided to research and see if any other high level teams were having this problem, and how they solved it. We watched a recent signature event and noticed that some teams had the same roller design as we did, but didn’t have shaft collars sticking out the end. Upon closer inspection, they were only using one versa hub on the flex wheel and were putting the shaft collar inside the wheel. Once we realized this, we implemented it immediately, and it worked beautifully. Since it was only supported on one side, there was a small issue that the wheel moved around a little bit. We solved this by adding a small piece to lexan to the end to support the wheel without having anything sticking out the end that could get potentially caught in the net and hinder us in any way.