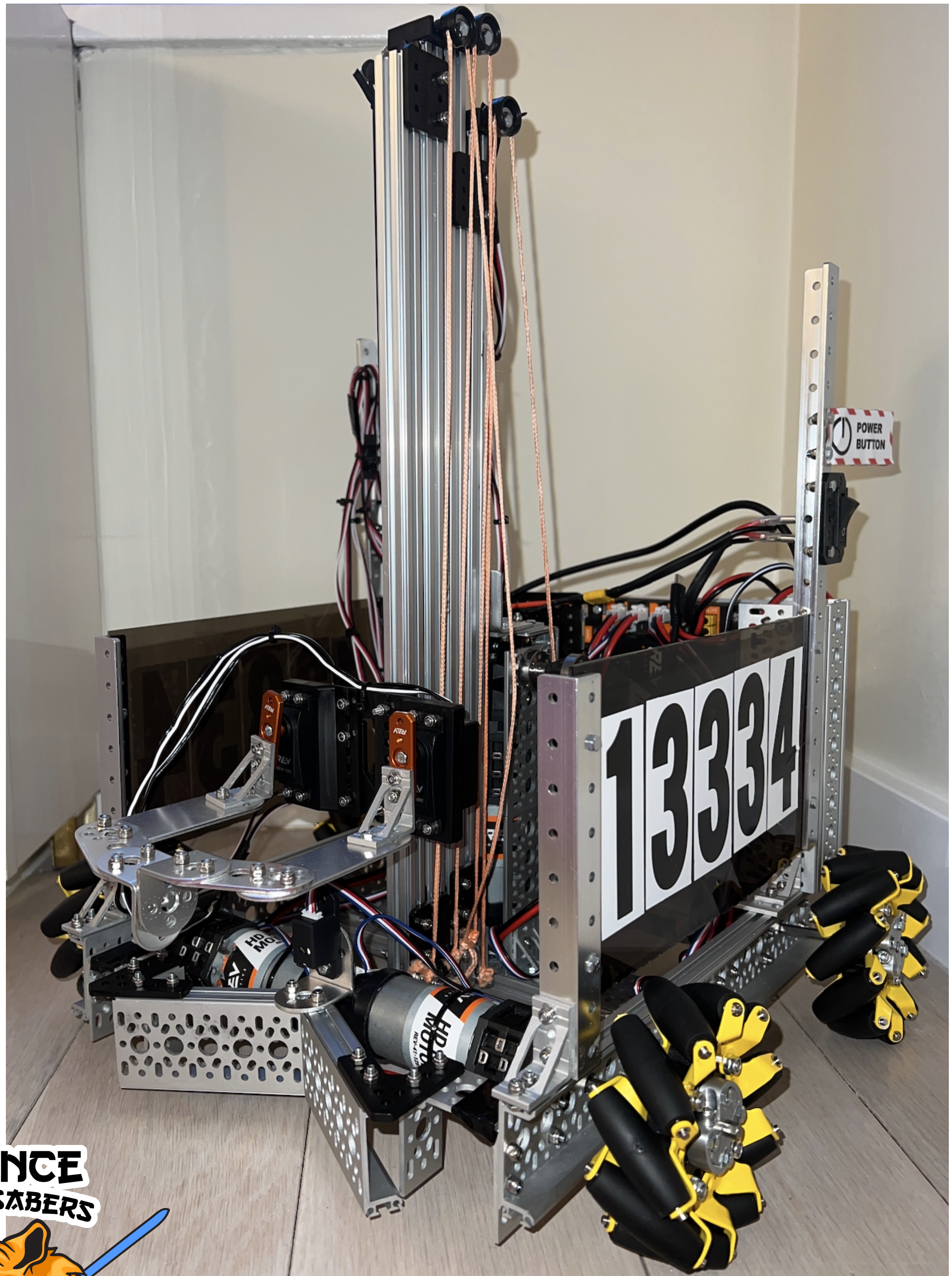
Engineering Portfolio

2022 - 2023



SPENCE LIGHT SABERS

13334

Team 13334

Meet the Spence Light Sabers!

The Spence Light Sabers team was founded in 2018 and first competed in the *FIRST* Tech Challenge Rover Ruckus Season. As a completely student-run team with This season, our team is composed of 18 students from 9th to 12th grade who all attend The Spence School, an all-girls independent school in New York City. Although most of our team members joined without prior programming and engineering experience, we have become a closely knit team brought together by a common love for robotics. Our team's aim is to inspire and encourage women and gender non-conforming individuals to pursue STEM, a historically male dominated field. Last year, we even created our own website, which can be visited at spencelightsabers.org.

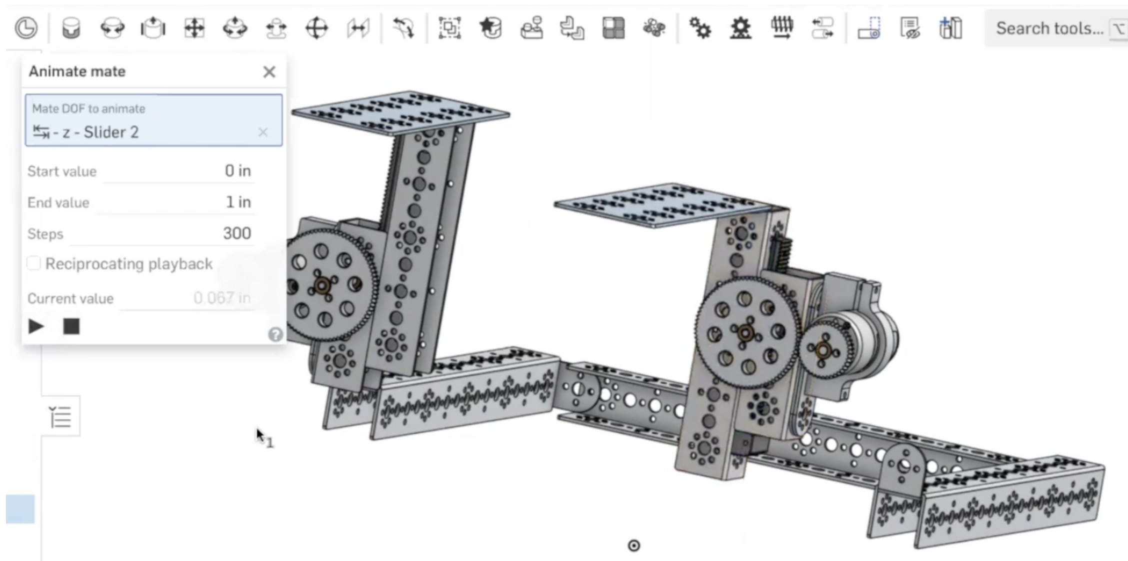
Team Plan

Our Plan: Sustainability, Self-Taught Skills, & Inspiring Young Women in STEM

Our team's goals are to create a sustainable team, teach ourselves new robotics-related skills each season, and inspire young women to pursue STEM.

In 2018, our team's four founding members' primary concern was developing a sustainable team, due to a lack of interest in STEM at our school and limited opportunities to educate students about robotics. As a result, we devised new recruitment strategies to encourage other students to join our team and empower our peers to explore engineering. This included hosting interest meetings that were open to all Upper School students, presenting our team's work to the Computer Science I and II classes at Spence, and actively recruiting 8th grade students from our middle school's FIRST Lego League team. Additionally, we present to 8th graders and invite them to attend our meetings during our school's annual Upper School Visiting Day, which gives middle-school students the opportunity to visit Upper School clubs. This early exposure to our team ensures freshman recruitment in the following year.

For the past two seasons, we've continued to implement these strategies into our team's sustainability plan, which has allowed us to retain membership and expand our team substantially since its founding in 2018. As membership is no longer a concern, we have now focused our efforts on teaching our members foundational skills in engineering, programming, and design. In particular, we like to emphasize that new members do not need any prior robotics or STEM experience, which has encouraged students with a large variety of interests to join the team. Since many of the newer members have little-to-no experience in robotics, we have created guidelines, reference sources, and video tutorials to teach ourselves the skills we need in coding, computer-aided design, engineering, academic writing, and researching. Below is a snapshot from one of our self-made CAD video tutorials that shows how to implement a rack and pinion mechanism.



Finally, we introduce people to STEM and FIRST both inside and outside of our school's community. Our active engagement with our school's larger community has been centered around providing insight into careers in STEM, highlighting female engineers, and promoting existing organizations that grant young women the opportunity to explore their interests in engineering. In addition to mentoring our school's FLL team, we host discussions with Spence alums who work in robotics that are open to both students and faculty *(pictured below)*, work with admissions to advertise our robotics team to all prospective students, and organize presentations to teach our school about our team's commitment to FTC.

Notably, we have been able to extend our outreach beyond both our school and local community this season. Last November, we began collaborating with Women in STEM (WiSTEM), an organization supported by young women and professionals who are passionate about celebrating women working across STEM. Currently, we are creating easily accessible infographics for WiSTEM that highlight FIRST and its mission. Additionally, we are crafting digestible lessons to introduce readers to different skills used in robotics such as building and object-oriented programming. This year, we plan to host discussions at Spence featuring ambassadors from WiSTEM to inspire both Middle and Upper School students with an interest in STEM to explore opportunities in engineering. We are also working towards collaborating with Nightingale's team, one of the only other all-girls teams in our league, to host an event this season. (For more details on our outreach efforts, please visit the *Outreach* section.)

Robot Design Process

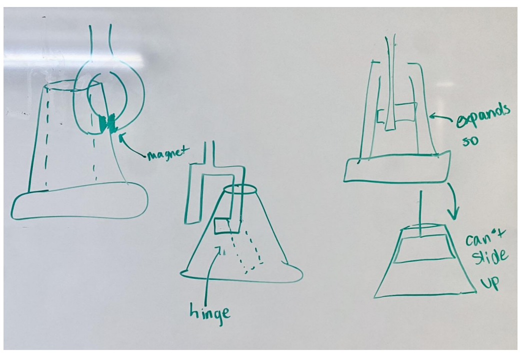
As our team is fully student-run, we were tasked with developing a design process that did not revolve around the guidance of a mentor. We modified our design process to include the core steps of a typical engineering design process, as well as ensure that every member shares their ideas and collaborates with the rest of the team to push our robot's design to its greatest potential.

We meet daily after school from 3:30 p.m. to 8:00 p.m. to both build and program the robot. Frequently hosting these longer meetings accommodates members that would otherwise have conflicting schedules. We also meet semiweekly for 30 minutes within the regular school day to troubleshoot engineering issues that occurred during our after school meetings, map our progress, and set new goals for the ongoing week.

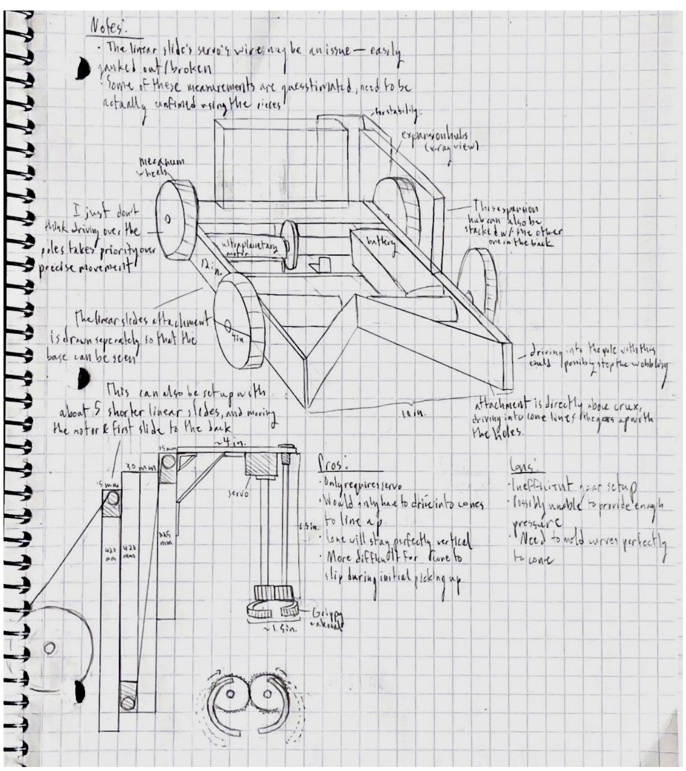
We take notes on all our meetings in our extensive Engineering Notebook to keep everyone updated on the team's progress and share potential ideas for the robot's design. Whenever members help to build the robot, they are tasked with writing a detailed notebook entry to update the rest of the team, which helps us immediately pick up where they left off at the next meeting. All of our members share the responsibility of writing in the Engineering Notebook, which gives both newer and older members the chance to develop their skills in detailed note taking as part of the engineering process. The notebook is also a great way to help new members acclimate to the club's routine.

In our first few meetings of the season, we identify the challenges of the FTC season competition, which range from dimension constraints of the robot to maneuvering around obstacles on the playing field. We use past team experiences to design realistic goals for the gameplay while keeping in mind time-restraints and the relative novelty of our team compared to others. At the same time, we challenge ourselves by creating team objectives and implementing new strategies/mechanisms, which advances our learning throughout the season.

As we begin to devise our game strategy, we task each member with independently designing a sturdy and space-efficient foundation for our robot. Then, we present our ideas at the following meeting and choose a foundation to build from, or combine several ideas. Once we decide on the robot's foundation, we plan the intake and outtake systems, and any attachments needed for that season. From this point, we continue to assign weekly, individual assignments where members brainstorm, sketch, and design the robot for specific objectives (i.e. designing an arm to place a cone onto the junctions). Our members then present each of their ideas to the team. These show-and-tell style meetings allow everyone to be up-to-date on our design and practice valuable presentation and communication skills.



We discuss the pros and cons of each design and are often conflicted between a few options. For instance, one arm design might be strong, but heavy, and therefore strains the robot when fully extended. Another arm might be lighter and space-efficient, but flimsy and less reliable. Specifically, when designing our intake mechanism, we were conflicted on whether picking up cones from the outside or inside would be more reliable. (For more details on the design process of our intake, please visit the *Designing the Intake/Outtake* section.) So, we discussed the dilemma as a team and sketched up multiple designs during one of our in school meetings as seen on the left. While we do our best to combine our designs, sometimes we end up scrapping them altogether and considering new ones. It is crucial that by the end of the process, we have an efficient, reliable, and structurally



sound design. Eventually, after discussing every possibility we can think of, we vote on a final design and begin the process of building our robot. The image above on the left depicts the peer-reviewed and edited blueprint of the robot that both inspired and guided our current design.

Once we generate a final design, we use our after school meetings to build and program the robot. As we build, we often run into new challenges that cause us to redesign and rebuild aspects of our robot. (For more details on the design process, please visit the *Designing the Foundation*, *Designing the Intake/Outtake*, and *Improvements When Building the Robot* sections.) We also split up the team into smaller groups during our longer meetings where members will collaborate on various tasks. The sub-team each member partakes in matches their interest while also giving them the opportunity to learn or practice a new skill. For example, some members are more interested in working on outreach projects while others prefer building the robot.

Once the building process is over, we begin to code. As many members were unfamiliar with Java programming, we dedicated time during our after school meetings to teach ourselves Java. We used tutorials we created last year to learn Java's syntax and the structure of hardware, teleop, and autonomous classes. We also assigned ourselves light homework to practice coding in Java and supplemented these assignments with weekly quizzes to refresh syntax fluency. An example from one of our Java guides can be seen on the left.

Hardware Classes

Template you should follow: HardwarePushbot

1. Change file name and match class & constructor name to said name
2. Declare your OpMode Members
   - "public DcMotor leftDrive = null;"
   - public [CLASS NAME] [name] = null;
3. Initialize OpMode Members
   - "leftDrive = hwMap.get(DcMotor.class, "left_drive");"
   - [name] = hwMap.get([CLASS NAME].class, "[NAME ON PHONE]");
4. Set Modes/Powers/Positions etc. (depends on what you are coding, I'm using a DcMotor class here as an example)
   - "leftDrive.setPower(0);"
   - [name].setPower(0);
   - "leftDrive.setMode(DcMotor.RunMode.RUN_WITHOUT_ENCODER);"
   - [name].setMode([CLASS NAME].RunMode.RUN_WITHOUT_ENCODER);
5. Delete what you don't need and adjust the comments

As we approach competitions, we split our time between rehearsing our presentation, writing the portfolio, creating the poster, finalizing the code for our robot, practicing driving, etc.

In summary, we start the season with individual assignments where we discuss and debate our ideas with each other; transition to more rigorous meetings where we focus on outreach, building, and programming; and then finish with assigning members specific tasks in the weeks leading up to the competition.

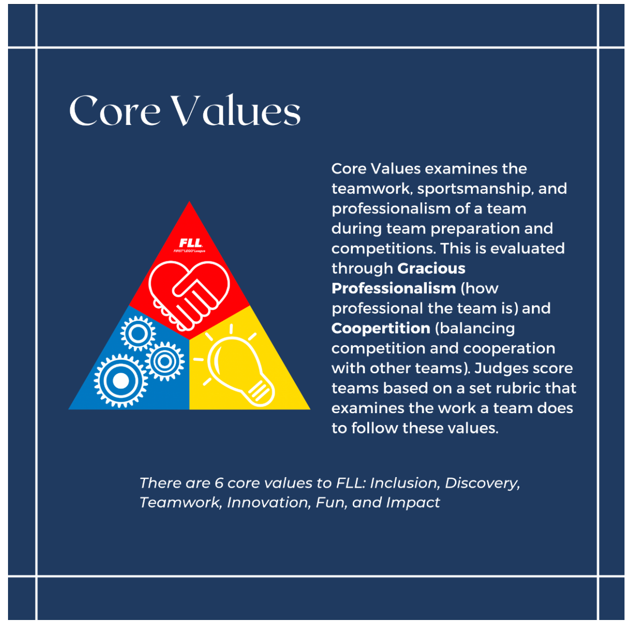
Outreach

Once a week, we spend time mentoring and working with the middle school FIRST Lego League robotics team. Our team helps them improve their programming and building skills; teaches them problem solving strategies and how to use logical reasoning when encountering difficulties as a team; and encourages them to explore their interests in STEM.

One example of how we helped the MS FLL robotics team was when they were trying to complete a mission *(pictured on the left)*. In this mission, they had to maneuver their robot to flip a switch. The ending position of their robot differed each time, so they were struggling to complete the mission accurately. Therefore, referencing our own experiences, we suggested that they slow down the robot on turns so that the momentum wouldn't cause the robot to go further than intended. We also suggested that they turn more when trying to flip the lever so they could face it head on which would make the completion easier and consistent.

In addition to encouraging young women in our own community to pursue STEM, we reached out to WiSTEM to help educate and empower a larger range of young women who want to pursue STEM but don't know where to start or fear being pushed out of majority male STEM spaces. In collaboration with *Women in Stem*, we are working to provide an easy and accessible way for girls to learn the fundamentals of engineering, programming, and robotics as well as how to get involved with FIRST. To accomplish this we are creating a series of recorded lessons to give an introduction to FTC, FLL, building, and coding to educate the next generation of female roboticists. Additionally, we are designing infographics *(pictured above on the left and below)* for WiSTEM's social media aimed to answer questions about FIRST, the engineering design process, problem-solving, how to get started with programming, and more. Finally, we are working with WiSTEM to choose a speaker from their organization to discuss their

experience as a woman in STEM and offer advice to the young female engineers and programmers in the audience.



Through our collective outreach efforts, we have learned to market our team; share our knowledge with younger students; and educate young women inside and outside our school about FIRST, programming, engineering, and robotics by offering our support and creating accessible educational resources for anyone to use.

Designing, Building, & Coding the Robot

Designing the Foundation

This season's game field requires robots to navigate through a complex maze of junctions and quickly pick up cones. To ensure our robot could do this easily, we employed a light/compact design, gear ratios, holonomic drive, and built the front of the foundation inwards to resemble a "V" shape.

While the maximum dimensions for the robot are 18 inches x 18 inches, we aimed for the frame of our robot to be just 12 inches x 10 inches to leave plenty of space for the robot to drive through two adjacent junctions. Additionally, we avoided unnecessarily using U channels for the foundation of our robot as they are much larger and heavier relative to C channels and flats. Lightening our robot meant an increase in speed, and therefore efficiency when transporting cones.

To further amplify the speed, we decided to optimize each motor's gear ratio. We researched the advantages and disadvantages of a high gear ratio and a low gear ratio, and learned that a low gear ratio would allow for a greater max speed while a high gear ratio would be better for carrying heavy weights. Using this information, we specifically chose to increase our gear ratio to 60:1 for the arm motor because we wanted to increase torque while we lowered our gear ratio to 20:1 for our drive motors to increase speed. After finalizing the ratios of each motor, we then attached them to the foundation.

Once the frame was finished, we debated over whether to implement a holonomic or non-holonomic drive. After outlining potential pathways from an alliance substation to various junctions, we concluded that a holonomic drive would be the most beneficial as we often found it more efficient for the robot to drive side-to-side rather than around a junction. Next, we debated whether to use mecanum or omni wheels. The mecanum wheels we had at the time were an older, heavier model that would frequently fall off their respective motors, so we considered omni wheels,

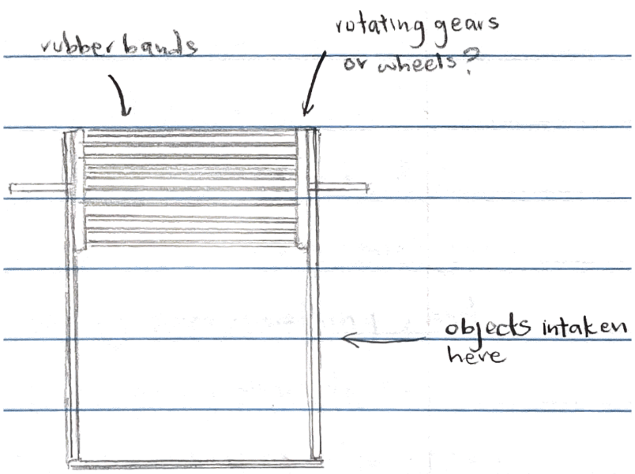
However, since our robot was already lightweight, we wanted to have sturdier wheels. Additionally, implementing omni wheels would interfere with the design of the robot's frame because we built the front inwards and thus wouldn't have a way to attach them. Ultimately, we decided to invest in newer mecanum wheels (*pictured on the left*), because they were the most compatible with our design and needs.

We chose to preserve our original foundation design when choosing which wheels to use because it is built inwards to funnel cones directly underneath the intake system. Since picking up cones requires a lot of precision, we worried about the risk of human error when driving the robot. We wanted a design where we could simply drive up to a cone and pick it up without having to reorient the robot until the cone was perfectly in line with the intake system. Because our intake system had to be in a fixed position, we needed a way to funnel cones into the perfect spot each time we wanted

to pick one up. Using two C channels, we built a V shape in the front of the robot that pushes cones right under the intake system as the robot drives into it, consequently saving us valuable time that would have otherwise been wasted on precisely orienting the robot. This design feature can be seen on the left.

Designing the Intake/Outtake

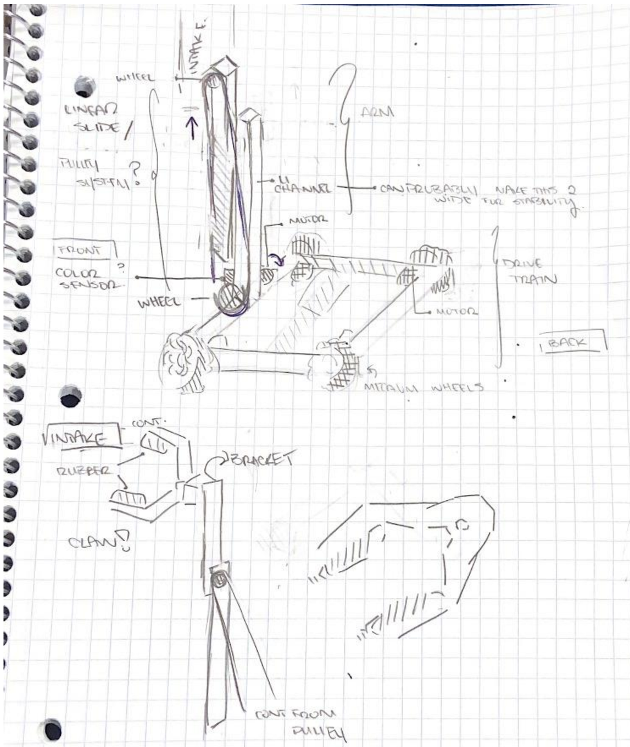
The first step in designing our intake and outtake systems was debating whether or not to use a joint or separate system. An example of one of our separate system designs was to have a "spintake" to pick up the cones and an elevator to deliver them

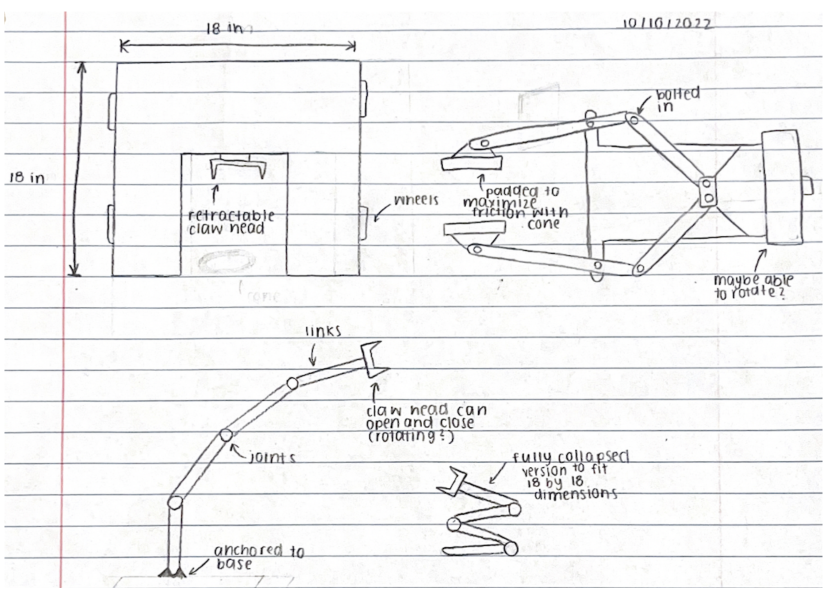


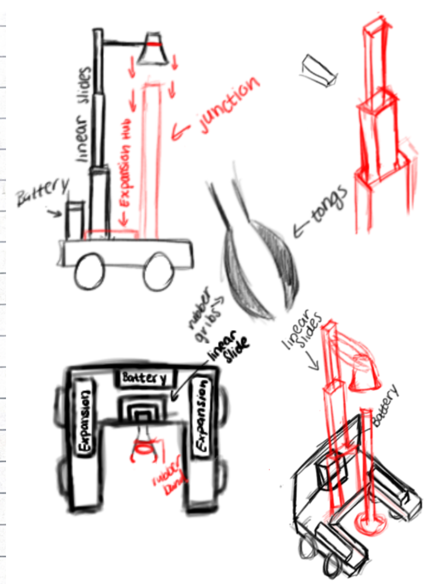
to each junction. The "spintake" would compose of a motor connected to two wheels where rubber bands are spanned across them with some tension *(pictured on the left)*. As suggested by the name, this mechanism would spin inwards to pick up the cone by having the bands conform to the conical shape. The issue we ran into, though, is that there would be no way to guarantee the orientation of the cone before it arrived at the elevator platform. Not only was this design unreliable, but as we realized with most of our separate intake and outtake ideas, the transportation time between the mechanisms would inefficiently extend the amount of time needed to pick up and deliver cones. Additionally, the more systems, the greater the weight and size of our robot, which we were trying to minimize. So we officially decided on designing a joint system, meaning that the contraption would pick up and deliver the cone under one system instead of two separate ones.

Next, we outlined the objective and needs of our intake/outtake system. In order to both pick up and drop cones onto each junction, the mechanism needed to be lightweight, easy to drive, able to deliver cones at different heights, and have a strong grip. Keeping this in mind, team members were tasked with independently brainstorming different designs before collaborating with the rest of the team to choose one. When it came time to present, almost everyone thought of similar if not the same mechanisms. Each arm was made of linear slides, extrusion channels, or a scissor lift to adjust to the junctions' heights and had some iteration of a claw with variations in shape. We quickly decided on using extrusion channels to make our lift system because they were the lightest and simplest. We wanted to ensure our arm wasn't unnecessarily heavy so as to not lessen the robot's speed or balance since the lift would have to extend even higher than the tallest junction whilst holding a cone.

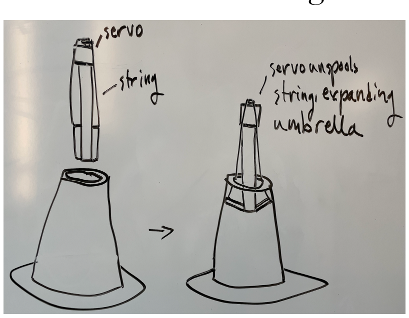
Our claw design, however, took significantly more time to design since we found ourselves debating between a claw that gripped the cone from the outside versus the inside. Three of our claw ideas can be seen to the left and below. The first was made out of straight brackets and had a rubber-padded interior to both conform to the shape of the cone better and apply friction to better grip the sides of the cone. However, we worried that even with the added rubber, the cone would simply slip right out, especially during transportation. The second claw is similar to the first while the third would be 3D printed to conform to the sides of the cone perfectly. While this would reduce the likelihood of slipping, we were still concerned that the claw wouldn't be able to grasp the cone with enough pressure.



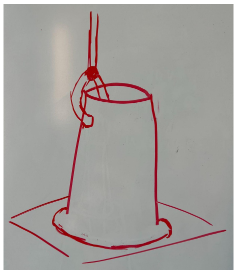




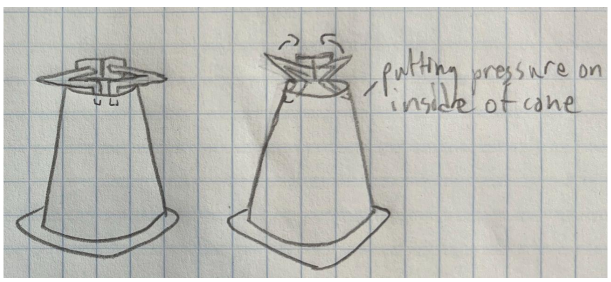
Our other designs involved picking up the cone from the inside. For example, we were inspired by how an umbrella opens to design a mechanism made of thin pieces that would be lowered into the top of the cone and then expanded *(pictured on the left)*. The expanded section would catch on the lip of



the cone as the attachment lifted, carrying the cone with it. A limitation with this design, though, is that the inner lip on the cone is not solid, so the pieces could get



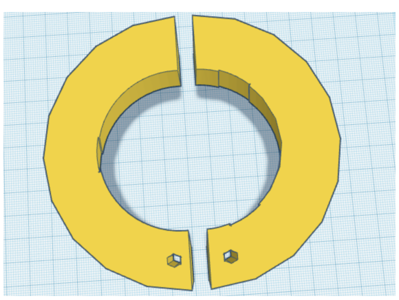
stuck to the inside when trying to drop the cone onto a junction. Another idea was to create a "mini-claw", that rather than solely grabbing the outside of the cone, it would lower into the cone and grasp its side *(pictured on the left)*. However, we found the narrow opening at the top of the cone too small for a design like this to work consistently. Our final and most stable idea was to create an "inverted-claw" where we would insert brackets into the cone and turn them outwards to apply pressure on the inside *(pictured below on the left)*. The main advantage of this design is that it would be extremely difficult for the cone to fall while being lifted. The main disadvantage of picking up a cone from the inside is that the top of the cone is a



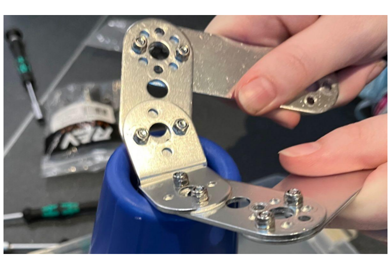
very small target to consistently aim for and would require extreme precision from the driver, which is why we built our foundation inwards (as mentioned earlier) to funnel the cone into the same spot repeatedly.

In the end, we decided to design two attachments: a custom made claw to conform to the cone's shape to grasp it from the outside, and the "inverted-claw" to pick up the cone from the inside. We wanted to create both to see which one performed better and have a backup plan in case one didn't work.

We built our first claw in Tinkercad *(pictured first on the left)* and matched its

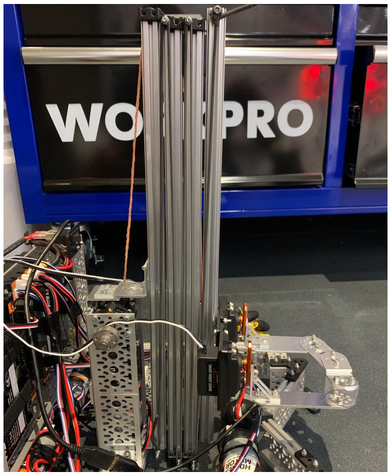


slope to that of the cone as well as later lined the inside with rubber tubing to increase the friction between the claw and cone. In order to pick up the cone, we designed a gear system that kept one side of the claw stationed while the other would open and close to grasp the cone. For our second design, we attached two 90° brackets to servo



motors to turn them outwards to pick up the cone and inwards to drop it *(pictured second on the left)*. Originally, we used one servo attached to a gear system to control the brackets, however, we found that using two servos worked better with our current inventory of parts.

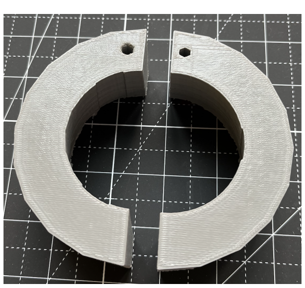
Improvements When Building the Robot



As we began building, we had to make some modifications to our design to adapt to the challenges we faced. When we first began testing our lift system, we realized that three extrusion channels were not tall enough to lift the cone above the highest junction. We decided to add another extrusion channel *(pictured on the left)*, extending the reach of the lift system. However, this required us to reconstruct the foundation in order to allow room for a fourth extrusion channel by replacing the 10 inch C channels with 12 inch C channels. We also noticed a lot of strain and cracks on the end caps attached to the channels, which jeopardized the structural integrity of our lift system. After conducting research, we discovered that switching from a continuous lift to a cascading one would lessen the strain on each channel and instead distribute it more evenly across the four channels. This switch not only made the system more robust and eliminated breakage, but also significantly increased the speed and smoothness of the lift.

Another issue we encountered was when the wires for the servo motor would get caught on various parts of the robot because they had to be long enough to reach the highest junction. To contain the wires, we added two long L beams on both sides of the robot and zip tied the wires to them. Additionally, we attached plexiglass walls to the robot to ensure the wires would remain inside our robot and not interfere with our wheels or other robots. These walls also provide a place to write our team number as well as attach our alliance markers and warning signs.

The biggest alteration we had to make to our robot, however, was finalizing our



intake/outtake design. We were unable to 3D print our custom claw *(pictured on the left)* without defects as we do not have access to newer printer models. The printer struggled with smaller details, and so it couldn't create holes for a hex shaft to fit through. Moreover, the slope of the printed claws did not match their digital models or the walls of the cone.

As 3D printing was an imprecise and tedious process, we decided to use our "inverted-claw" claw design *(pictured below)*. Initially, we had difficulty implementing this mechanism as we tried using gears and a single servo, which resulted in over complicating our idea and required parts we didn't have. Instead, we added a second

servo, minimizing the weight of and simplifying the overall design to optimize efficiency and speed when placing a cone. Since this design applied so much force to the cone's inner walls, it was virtually impossible for the cone to slip out. This made the mechanism incredibly reliable and consistent regardless of how far or high the robot transported the cone.

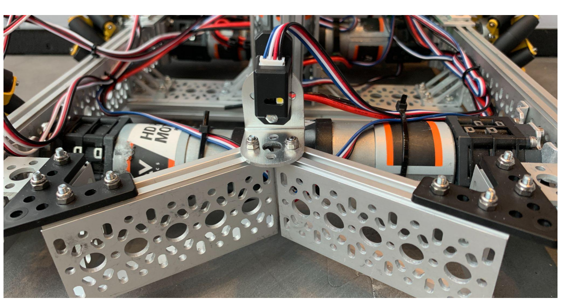
Programming

While creating pseudocode, we focused on using automation and sensor input to lessen the precision required by the driver and boost efficiency. To accomplish this, we used encoders to preset heights for the lift system to place cones on each junction; implemented a color sensor to read our custom signal sleeve to then move to the indicated Signal Zone; and programmed a button to slow the speed of all the motors. We configured a second gamepad that only controls the lift to make it easier for the drivers to align the robot with a junction to place a cone.

```
// Use dpad to raise and lower arm to preset levels.
// Low.
if (gamepad2.dpad_left || gamepad1.dpad_left) {
    robot.arm.setTargetPosition(7700);
    robot.arm.setMode(DcMotor.RunMode.RUN_TO_POSITION);
    robot.arm.setPower(1);
}
// Medium.
else if (gamepad2.dpad_up || gamepad1.dpad_up) {
    robot.arm.setTargetPosition(11000);
    robot.arm.setMode(DcMotor.RunMode.RUN_TO_POSITION);
    robot.arm.setPower(1);
}
// Tallest.
else if (gamepad2.dpad_right || gamepad1.dpad_right) {
    robot.arm.setTargetPosition(15000);
    robot.arm.setMode(DcMotor.RunMode.RUN_TO_POSITION);
    robot.arm.setPower(1);
}
// Ground.
else if (gamepad2.dpad_down || gamepad1.dpad_down) {
    robot.arm.setTargetPosition(2735);
    robot.arm.setMode(DcMotor.RunMode.RUN_TO_POSITION);
    robot.arm.setPower(-1);
}
```

We pre-programmed the four directional pad buttons to lift the arm either to the lowest, middle, or highest junction as well as bring the whole lift system to the ground *(pictured on the left)*. Preset heights are not only crucial to the Autonomous period, but also improve efficiency and precision during the Teleop period as well as make it easy to cap a junction with our custom, rocketship inspired beacon. However, we also coded two separate buttons to manually lift and lower the arm in case we needed to precisely adjust the height. In order to prevent the lift from extending beyond the height of the tallest junction when controlled manually, we programmed a maximum height.

As we are a self-taught team, we taught ourselves how to configure, program,



and troubleshoot the color sensor through syntax databases and tutorials online. Testing our sensor, we noticed major discrepancies in the sensor's values, which meant that the robot would often park in the wrong zone.

Eventually, we found that the color sensor had to be in close range to our custom signal sleeve to return proper values. So, we used the V shaped foundation to funnel the cone directly in front of the color sensor, which we attached right above the V *(pictured on the left)*. Using the V allowed for a larger margin of error during the autonomous mode so that if the cone was placed slightly off-center, the angle of the V would push it back into place toward the sensor.

After test driving our robot, we found it difficult to properly align the robot with a junction to drop a cone, because the robot would either turn, drive forward/backward, or lift the arm too much when we wanted minimal movement. In order to allow for more driving precision and ease, we programmed the left bumper on the gamepad to slow down the driving and arm motors as long as it is pressed. When the driver releases the button, the robot returns to its original speed. We used

```
double spin = gamepad1.left_stick_x; // For controlling spin.

double spinPower1 = spin * 0.5;
double spinPower2 = spin * 0.3;

if (Math.abs(spin) > 0.1) {
    if (gamepad1.left_bumper) {
        // If someone is moving the right joystick and want to move slowly, then spin.
        robot.rightFrontDrive.setPower(-spinPower2);
        robot.rightBackDrive.setPower(-spinPower2);

        robot.leftFrontDrive.setPower(spinPower2);
        robot.leftBackDrive.setPower(spinPower2);
    } else {
        // If someone is moving the right joystick and want to move quickly, then spin.
        robot.rightFrontDrive.setPower(-spinPower1);
        robot.rightBackDrive.setPower(-spinPower1);

        robot.leftFrontDrive.setPower(spinPower1);
        robot.leftBackDrive.setPower(spinPower1);
    }
```

variables such as spinPower1 and spinPower2 *(pictured on the left)* to make it easier to update/change the robot's speed throughout the competition. To further improve driving fluency, the additional gamepad allows for the one driver to just focus on lifting the arm while the other focuses on aligning the robot itself.

Conclusion

We are incredibly proud to have achieved so many of our goals this season. We have spread interest in FIRST and encouraged young women to pursue STEM both inside and outside our community from mentoring our school's FLL team to connecting with WiSTEM; developed a collaborative design process that uses our ingenuity to overcome engineering challenges, resulting in innovative, outside the box, and reliable robot solutions like our V shaped funnel and inverted-claw mechanisms; created an overall functional robot that easily navigates through the field of junctionsa , yet remains compact and uses a plexiglass structure, ingto give it an aesthetic and clean appearance; and incorporated sensors and automation for an intelligent robot.

While we have accomplished so much as a self-taught team, we hope to find a mentor with FIRST experience to assist and guide us next season to advance our knowledge even further.