Team 18015

The Omega Squad

http://theomegasquad.org

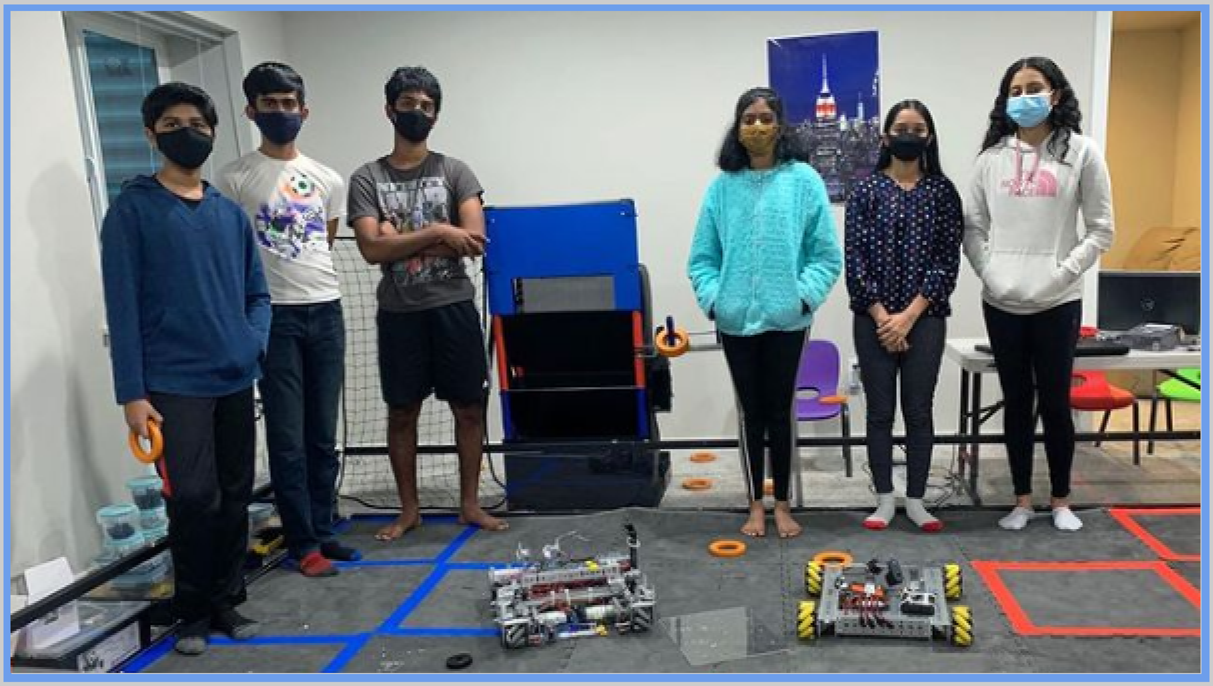
Table of Contents

Our Team 3
Our Coaches 4
Season strategy 4
Game strategy 5
Outreach 5
Outreach Flyer 6
Robot Overview 7
- Robot Drive Train 7
- Ring Intake 7
- Carousel Mechanism 7
- Ring Shooter Mechanism 7
- Wobble Goal Mechanism 7
- Webcam 7

Robot Design Process 8
- Brainstorming Robot Design 8
- Attachment Prototype 8
- Programming Prototype 9
- Hardware Integration 9
- Program Integration 10
- Sensors and Computer Vision Programming 11
- Testing and Improving Robot Hardware 13
- Testing and Improving Robot Software 13

Website Design 14
Future Improvements and Ideas 15

Team Picture: (From Left to Right - Dharsan, Gaurav, Amogh, Swetha, Anvi, Inchara)



Our Team

Inchara is in 10th grade and goes to North Creek High School. She joined FTC because recently, she found an interest in robotics and she wanted to further explore this topic. She decided that doing hands-on robotics projects and joining a robotics team would be the best way for her to learn more and expand her understanding. She's specifically interested in applying robotics to the medical field and she's also interested in web design.

Amogh is in 9th grade and goes to North Creek High School. He joined FTC to learn more about programming, engineering, and other related fields. He also wanted to improve his business knowledge through sponsorships, and learn how to design robots through CAD. He likes programming, web design, and engineering.

Anvi is an 8th grader at Skyview Middle School. She has done FTC for a year and she wants to continue to do FTC because she learned a lot last year and wants to expand her knowledge. She also loves coding, engineering, and working in teams. Building new things and experimenting has always been one of her passions, and she hopes participating in FTC will grow those passions into a greater ambition.

Dharsan is an 8th grader at Skyview Middle School. This is his first year doing FTC. He is doing FTC because he enjoys expanding his knowledge about programming and CAD modeling and wants to learn more about this. He has always enjoyed building things as well, and he aspires to take that joy and build it into a passion..

Gaurav is an 8th grader who goes to Finn Hill Middle School. He wants to be in FTC because it is an interesting learning experience and he enjoys learning how the robot pieces together and works and how the code works to move all of the mechanisms on the robot. He also likes meeting new people and being part of a team.

Swetha is in 7th grade and goes to Redmond Middle School. She's been doing FLL for three years and this is her first time doing FTC. She's always liked seeing how things work and usually wonders how different machines are put together and built. She also likes doing the programming part of robotics, and overall coding.

Our Coaches

Our coach, **Dr. Srujana Karlapalem** is a Physician Assistant Specialist in Bothell, Washington. She graduated with honors from University Of Washington School Of Medicine in 2013. Dr Srujana Karlapalem graduated with Bachelor of Medicine, Bachelor of Surgery (MBBS) from Gandhi Medical College, Secunderabad, India. Having more than 8 years of diverse experiences, especially in PHYSICIAN ASSISTANT, Dr. Srujana Karlapalem affiliates with Evergreenhealth Medical Center, cooperates with many other doctors and specialists in many medical groups including King County Public Hospital District No 2, Healthpoint. She has been coaching First LEGO League teams for the last seven years and First Tech Challenge team for the last two years.

Our coach, **Mr. Babu Govindarajan** is working as Lead Software Architect with Accenture. He graduated with a Master of Engineering (Mechanical) from Guindy College of Engineering in 2000. He has about twenty years of software development experience in India and the USA. He has been coaching First LEGO League teams for the last seven years and First Tech Challenge team for the last two years.

Season Strategy

The first part of our design process was the brainstorming aspect where we thought of ideas and designs for our robot. Everyone on our team came up with an idea for the robot, and we then debated the pros and cons of each design and came up with our initial robot design. After we came up with our ideas for the robot, we had to put them in action. Our team split up into three groups of two, and each group worked on one aspect of the robot. One did the programming, another did the shooter/carousel, and the last did the wobble goal mechanism prototype.We did this because having both attachments will give up the maximum points we can get, and the divide and conquer method helped us finish our robot three times as fast. One team later changed to image processing, and the other became testing. Throughout the season, we have switched teams so everyone can contribute their own ideas.

Another part of our strategy was to have back ups of our working code. We checked in our code on GitHub regularly, in case the working versions accidentally got messed up. This way, we can ensure that we always have working code and can revert back to our original program if we ever need to.

At the start of every day, we planned out the goals for the day, and later reflected upon how well we achieved them. When planning the goals for the day, each group discussed how they were going to achieve their goals. Sometimes, the day went to plan, but most of the time, we ended up not achieving all of our goals, or our solutions led to more problems. The solutions to these problems improved our robot greatly. This goes to show that in FTC, and in real engineering careers, finishing a design is not the end. New problems will arise, and new solutions will need to be made. Engineering is not a straight line, like design, build, and test. It is a cycle: find problems, and redesign again and again, until your robot is the best it can be.

Game Strategy

In Autonomous Mode, we decided to use image processing to see how many rings are in front of us instead of guessing randomly. Based on the amount of rings there are, the robot will move to the corresponding square and release the wobble goal using the attachment. We use the wall to align the robot so it's as accurate as possible. Then, the robot will shoot 3 pre-loaded rings into the high goal, and move forward to park on the white line.

Then comes the TeleOps period. We have decided to go with an attachment design that will grab the rings from the front with surgical tubings spinning at a high speed. This will pull the rings up to the mechanism known as the carousel, which will spin the rings 180 degrees, resulting in it facing front again. Using the surgical tubing, the ring will be placed into a chamber with a very fast spinning wheel, and launch the ring at a very high speed, with controllable altitude, allowing us to change targets. When the End Game starts, the drivers move towards the wobble goal that we placed in the square during autonomous mode. We use the attachment to pick it up and move to drop it outside the playing field.

Outreach

On April 17th, 2021, we had a virtual outreach event with people in our community (including people who professionally work in the STEM field). We had handed out flyers and posted on social media about this event. We talked to them about FTC and what we do as a team. We also told them about our strategy and process, our robot design, our code, and gave them a demonstration of our robot with the hope to inspire them to join our team next season. In the demonstration, we showed them our robot doing the Autonomous, TeleOps, and Endgame parts of a match. We then answered some questions the audience had about our robot and team.

On April 30th, we also had another virtual outreach where we showed our robot to roughly 150 students studying Mechanical Engineering and intend on pursuing STEM topics professionally, as well as college professors that teach Mechanical Engineering. We talked about what our robot does, what our competition is about, explained our robot's attachments and functions, explained the code behind our robot, and gave them a demonstration of our robot. We then answered questions about our robot and team as well.

While our FTC season was going on, we also helped another FLL team. By the name of Quad Squad, members of our team occasionally helped them out. They were using Python Programming instead of the block based code, meaning they were inexperienced in that field. We decided to help them learn about python programming for some parts of their year.

We were contacted by a local police force early into the season. They asked us if we could fix a robot they had. After we went to the station, they explained that it was a robot that they had gotten from the military as a hand me down. They gave it to us to try and fix it. We made some progress, like figuring out how to turn on the robot, but we ran into a roadblock. The robot and the computer couldn't link up, and we looked everywhere on the internet for a fix for the problem. Unfortunately, the only information that we could find was that it would have to be reprogrammed by an employee of the company that made it, and the line was discontinued long ago. We returned it to the police, but we were still able to make a lot of progress in fixing the robot.

Because of Covid-19, there were not many opportunities to provide outreach to the community in person. However, we used social media to share our robot's progress with the city we live in, Bothell. By using Facebook and Instagram, we are able to come in contact with many people, and expose them to the FIRST program. You can check out our social media on our website.

Outreach Flyer

Robot Overview

Robot Drivetrain

This year we used GoBilda's Strafer Chassis kit. We used four Planetary Gear Motors to power 96mm in diameter Mecanum wheels. Our robot has a REV Robotics Control Hub and Expansion Hub to control the robot. We use Logitech C270 USB Camera for computer vision and machine learning.

Ring Intake Mechanism

Robot intake mechanism consists of a ramp and a carousel. Surgical rubes pull the Ring and push them over the ramp. For the ramp we used a chain drive with a single DC motor to power two shafts having surgical tubes to push the Ring over the ramp. We threaded the chain through two gears connected to the different shafts.

Carousel Mechanism

The robot's carousel mechanism allowed us to turn the Ring 180 degrees, so we can shoot the Rings while driving the robot without turning the robot. Carousel's central shaft is powered by a single DC motor. The central shaft spun surgical tubes to turn the Ring around in the semi-circle closed enclosure.

Ring Shooter Mechanism

The shooter mechanism allows the robot to shoot the Rings to the desired height. The shooter mechanism consists of a smaller ramp and a 4 inches compliant wheel to shoot Rings. We used a high RPM DC motor to power the shooter wheel. We used projectile Motion formula to calculate the desired shooting ramp angle.

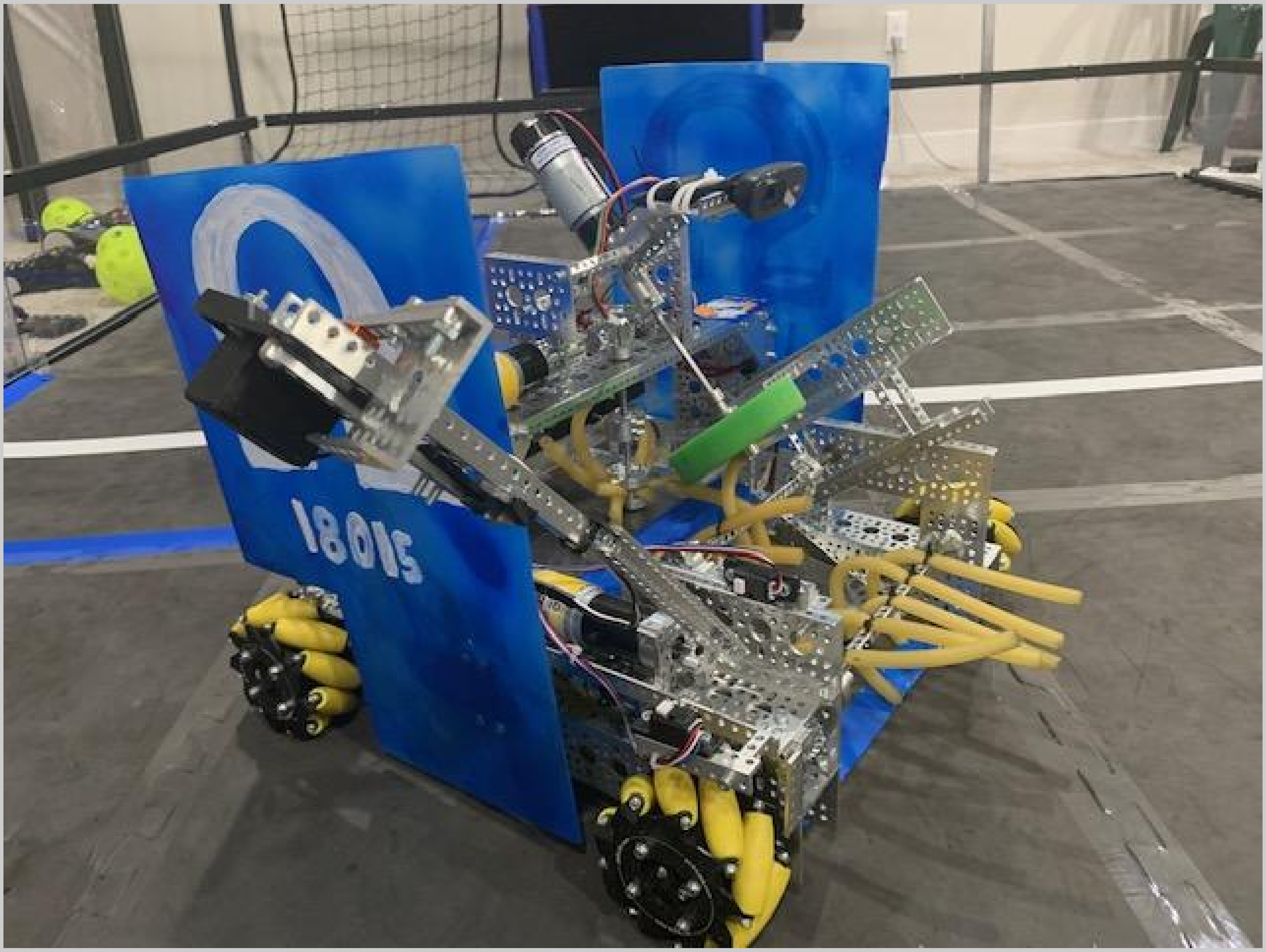
Wobble Goal Mechanism

The wobble goal mechanism enables the robot to grab and lift the wobble goal. It consists of a Servo motor to hold the wobble goal and a DC motor to lift the wobble goal using the arm and place it outside of the field.

Webcam

During the autonomous period, our program uses the webcam image to identify the number of Rings placed on the field and place the wobble goal to the right square. Our program uses Vuforia and Tensor flow APIs to identify the number of rings.

Our Robot:



Robot Design Process

Brainstorming Robot Design

During this phase, each team member came up with design ideas to do the game mission. Every team member presented their ideas and as a team we discussed the pros and cons of each design. As an outcome of the discussion, we decided to develop a mechanism to grab and move the wobble goal and to shoot rings to the top slot to get a higher game score.

Attachment (Mechanism) Prototype

We formed three groups and one group worked on the shooter prototype. Our first prototype used 4 wheels and one DC motor, and gears to transfer power. The mechanism was heavy and many moving parts. So we came up with a new idea to have 2 wheels on one side and a wall on the other. There was an improvement and the mechanism was able to shoot the Rings, but we were unable to shoot them to the required height and distance. So we decided to connect the shooter wheel directly to the motor, so we won’t lose the speed and torque in the motion transfer through the gears. We realized that this mechanism can not use more than one motor as other mechanisms and wheels need other motors. The group successfully tested the prototype with a single motor to shoot the Rings.

The other group was given the job of prototyping the wobble goal attachment. We first had to build our idea (the original prototype) and make sure the servo motor worked and that the wobble goal could be picked up (by figuring out how to program it). We then made changes to our prototype. First we had to come up with the design for the arm, where it was going to be placed, and how it was going to lift the wobble goal. Once we got a design that was visibly pleasing and would properly function, we started making the prototype by using pieces from Andymark and GoBilda. The group successfully tested the prototype to grab and move the wobble goal across the field.

Programming Prototype

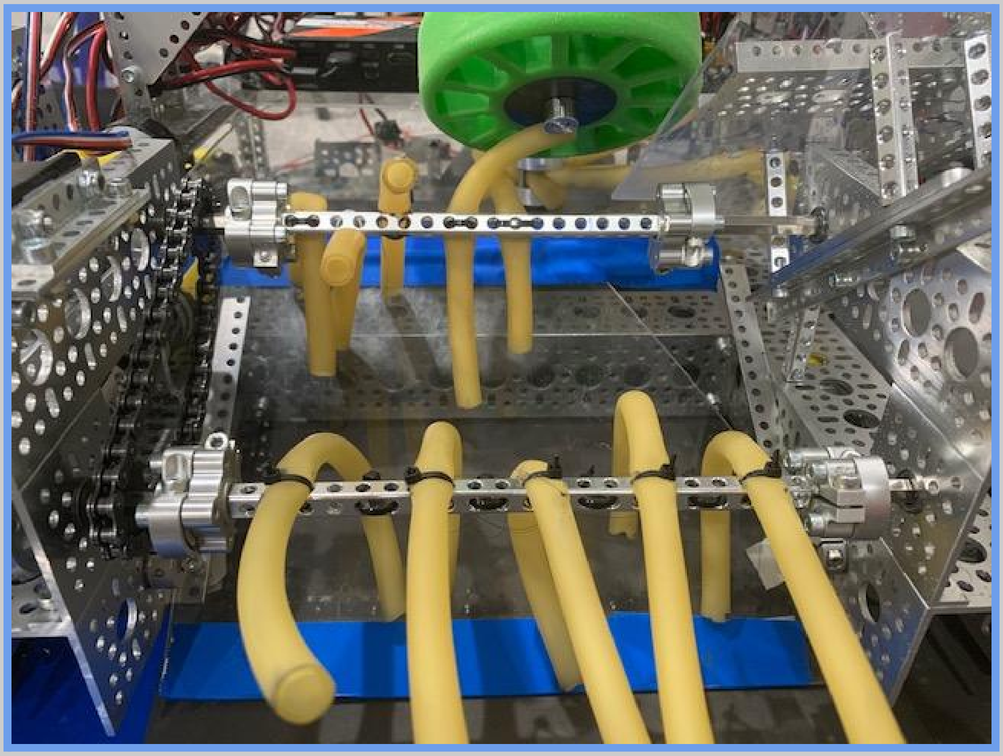
The third group worked on building an identical base robot so the team can do programming experiments while the main robot being used in mounting attachments. The group used the second base robot to experiment with Vuforia and Tensorflow programming. The group focussed on programmatically identifying the number of Rings placed in the field and driving the robot the right square. The group also experimented on using IMU gyro sensor, color sensor and encoder.

Hardware Integration

After we successfully tested our attachments prototype, we had to move on to integrating the prototypes with the main robot. When we worked on the wobble goal prototype, we had to figure out how and where we had to mount the wobble goal to the robot. We had to plan firstly how the wobble goal arm was going to be attached to the robot and secondly where it needed to go so it wouldn't interfere with the other attachments. During integration of wobble goal attachment, we realized the attachment is not tall enough to put the wobble goal over the fence, so we increased the length of the arm.

The Ring intake mechanism and the carousel were directly built on the main robot as these are low risk components of the robot. Later we integrated the shooting mechanism with the main robot. For intake and the carousel we tested surgical tubes having different flexibility and finally settled with one in the robot. We use one motor to rotate two shafts having surgical tubes using a chain and sprocket mechanism.

Robot Intake Mechanism



Program Integration

For the robot's movement during TeleOps mode, we copied code from last year (using GitHub) and changed it to fit this year's robot. In that code, we used the joystick to move the robot in the direction we wanted it to by measuring the x and y positions of the joystick.

We then moved on to the attachment programs and started with the intake, carousel, and shooter. After experimenting with various buttons to use, we finally settled on using the right bumper to suck in the rings and operate the carousel, and the left bumper for the shooter. We had to change the speeds of the motors multiple times to get the most accurate shots. We had some trouble with making the motors stop after we stopped pressing the button, but we eventually figured it out. Our robot's motors would also sometimes rapidly start and stop, but we soon found out that it was because of a misplaced sleep command in our code. After fixing the bugs in our shooter/intake/carousel code, we moved on to work on the wobble goal attachment code. The main struggle we had with the wobble goal attachment was coding for the servo motor that opens and closes the claw. It used to resist whenever we tried to open the claw, and we eventually found out that it was because we accidentally coded for the servo motor twice. Once we deleted one version, it worked and didn't resist us anymore. Throughout our code, we used telemetry to help us test and locate errors as well as comments to keep track of what part does what.

One group programmed image processing for the autonomous period. We used sensors and cameras to sense how many rings are in front of the robot to make it go to the correct target zone. We built and used a second robot for this so we didn't interfere with the other 2 teams' work. To start things off we copied the sample code from GitHub(which was used last year) then made some changes to the variables to match our previous programs.Once we had changed the variables we did a blank test. We just downloaded the program and ran the code to see what would happen. It didn't work, and it showed multiple errors, so we had to tinker with the values and statements to see what they did, which was very time consuming.

We used Vuforia, a machine learning tool, to see how many rings are in front of the robot. Then we started to change what the webcam saw, like the height of the rings. If the height of the pixels is greater than or equal to 200 then there are 4 rings in the field, or if the height detected was less than 200, there's only 1, and if the height was 0 then there are 0 rings. Once it detects the rings it will go to the corresponding target zone. We made a lot of mistakes with the code but we learned from them and fixed the mistakes to make the code work properly. Then we uploaded the code into GitHub so that the other part of the team could integrate it into the actual robot.

Sensors and Computer Vision Programming

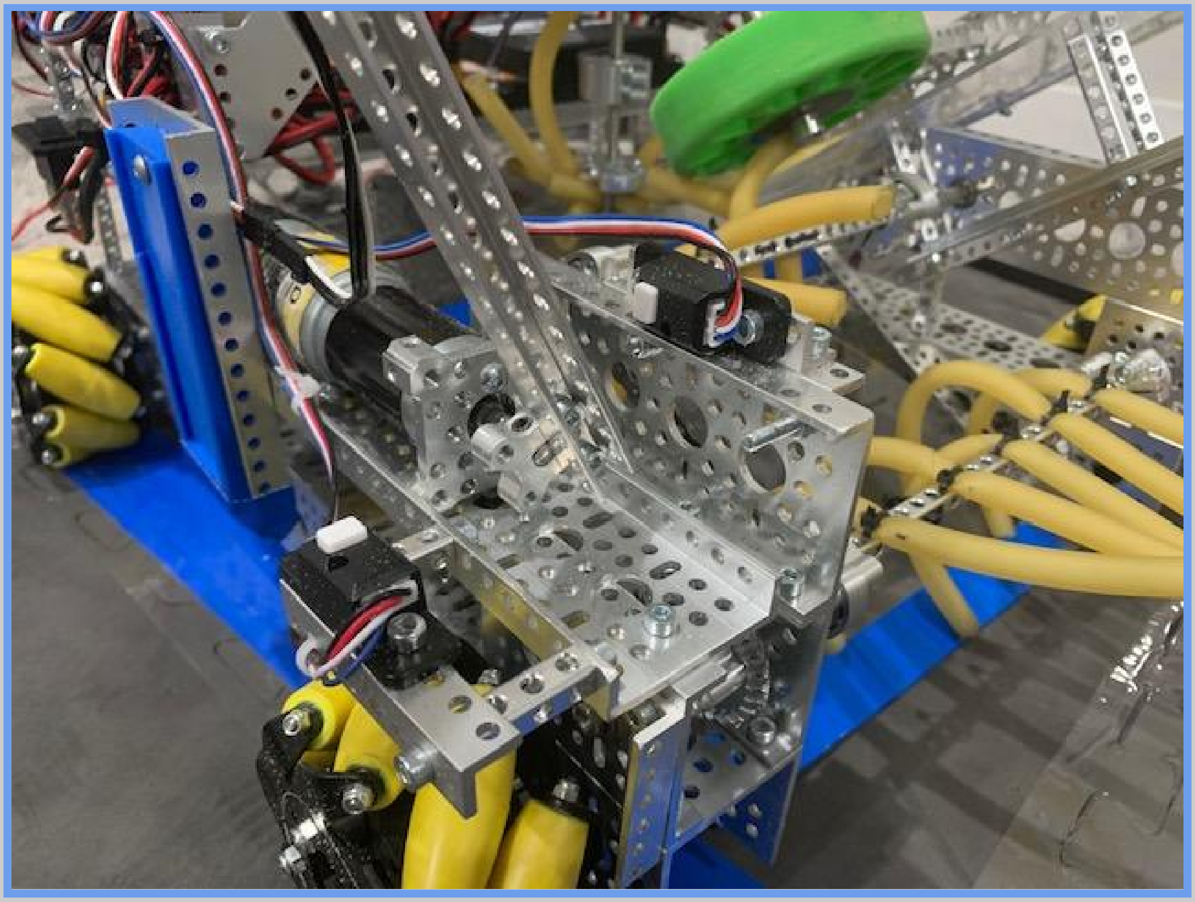
Using sensors in the program - autonomous and tele-op - was essential to the efficiency and accuracy of the robot. While the main and most advanced control feature we used was the Vuforia machine learning tool, to detect the rings in front of us, we still used other sensors. A major problem that occurred many times was the wobble goal arm not stopping on time during the tele-op, due to human error. This would cause the motor's shaft the chip off, causing major instability to the arm and attachment as a whole. Our fix to this was to add touch sensors to limit how far we were allowed to move the arm. This was also a major programming hurdle for us, because it required some complex logic that took some time to wrap our heads around. Regardless, this program fixed the problem, and inadvertently increased our score, by allowing us to pick up and drop both the wobble goals outside the field in the endgame.

Another addition that we added was the use of the IMU sensor. The IMU is a gyroscope in the Rev hub. We were originally confused by the code, as the sample code given gave lots of values. However, through online tutorials, we were able to come up with two complex functions that take into account the angle of the IMU to turn specific amounts of degrees. This was confusing at first, because we had to sift through all the values. People on our team had tried using the sensor several times, but it turned out to be difficult. However, we were able to make use of the IMU to turn properly in our autonomous code. Before the addition, the robot ended up in a position that made shooting the rings difficult. However, using the gyroscope, we were able to straighten out the robot and get points for shooting in the goal during autonomous. However, with success also comes failure. While it took many attempts to make these control inputs work (especially with Vuforia), there were some ideas we couldn't implement.

One of these was with the motor encoders. We only had one encoder cable that was compatible with our GoBilda motors, and we couldn't order more as a result of time pressure. We attempted to use the encoder to make shooting more consistent. This problem was derived from the motors being too powerful when the battery was just charged, but to slow when it was tapering out. The speed set by the program doesn't change, by the power does. This problem is made even worse because our runs take up a lot of battery power, and we have to switch batteries twice in a hour. We tried to use the encoder to solve this problem, but it only made it worse. It took more time to get to maximum speed, and it couldn't seem to stay there. We decided to scrap this idea for next year, as the deadline for submitting robot scores was almost up. We hope to use encoders for all motors, including the drive motors, for next year. Another control method that didn't make it into the final robot code was the use of color sensors. Originally, the idea was to have a downward facing sensor to detect the white line of the shooting zone. This would prevent us from getting penalties in the autonomous.

However, this idea later developed into a substitute for the Vuforia ring checker system. The plan was to have two color sensors at various heights. One at the height of one ring and one at the height of 4. The robot would have gone close to the rings and scanned the color sensors. If there were no sensors lit up, there would be no rings. If the bottom most one detected an orange color, there was one ring. Finally, if both detected a ring, there would be four. We designed and installed the color sensors at the desired heights. However, it was then that the Vuforia system started working more efficiently and accurately. We decided this would be the better idea, as it doesn't require such precise movement. We removed the color sensors from the robot and adapted the code for the Vuforia system.

Touch Sensor Setup



GitHub link for our code:: https://github.com/16113-ftc/16113_2020-21

Testing and Improving Robot Hardware

We had to improve the placements of the parts, like when we were putting the large motor on the robot, we realized the amount of space we had left, and we made numerous changes to the placement of the larger motor. Just like finding the placement of the larger motor, we ran into a lot of obstacles. For example, when we were testing the claw part of the lifter to see if it held the goal properly, we found out that the parts we used did not have enough friction, so the wobble goal kept falling off. We fixed this by finding some parts that had enough friction and would fit on the claw.

After testing the wobble goal attachment, we realized that we needed touch sensors to prevent the arm from harming the base robot's structure. We changed some sample code and added touch sensors onto the robot. Later on, we changed the placing of the sensors so they were in line with the arm and it would consistently hit them. We kept changing the code until it did exactly what we wanted it to, but we had another issue with the claw not opening and closing fully. We fixed this by making sure to hold down the button that opened and closed the claw to make sure that the claw wouldn't accidentally drop the wobble goal.The ramp ended up cracking after getting rammed into the wall, so we had to cut and attach a new ramp. We also ended up adding tape at the bottom of the ramp because it was too high off the ground.

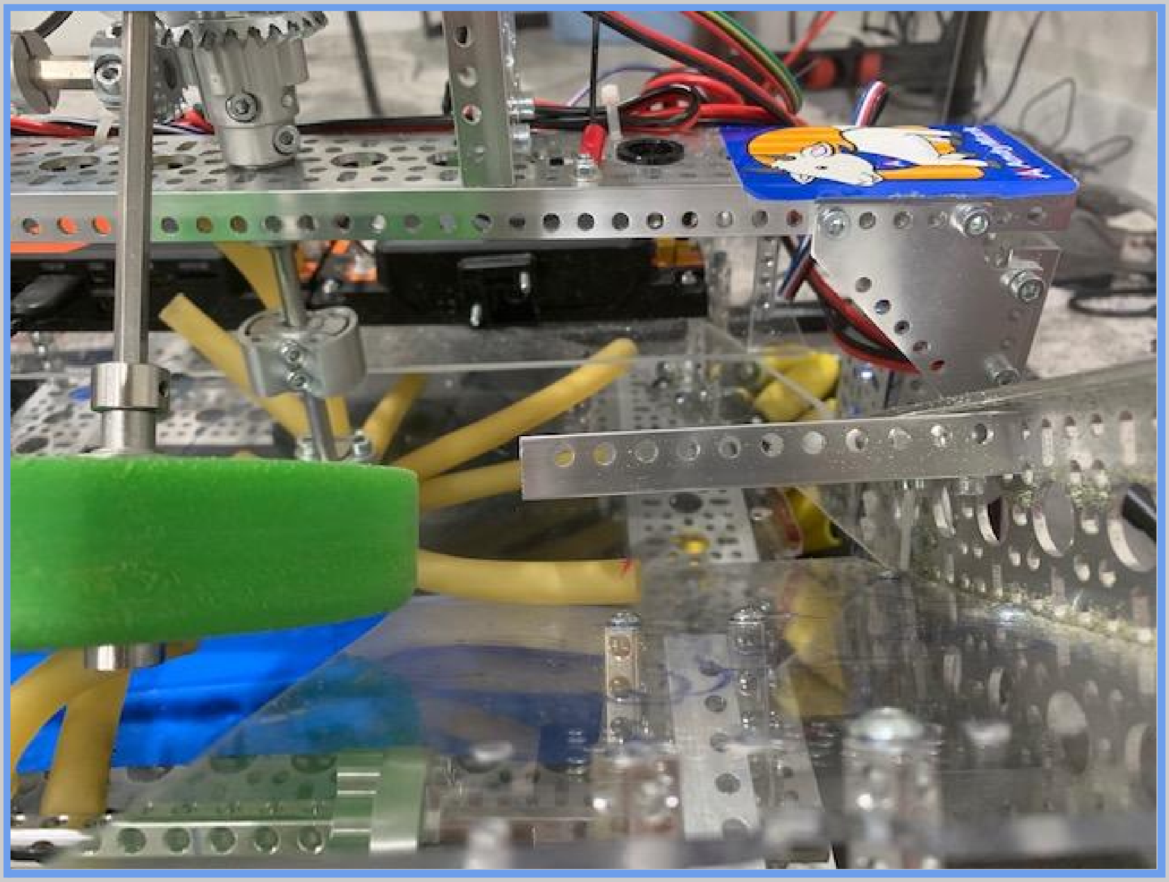
The intake stopped working consistently after a while and we discovered it was because of the short and unequal lengths of the surgical tubes. To fix this, we cut longer tubes and added them to the robot. The carousel also wasn't working correctly because the tubes kept bending upward and they didn't push the ring far enough. We made the new design of the carousel and added it to the robot. We also switched the surgical tubing we used to intake the rings, and made the intake ramp more thin, allowing for easier intake. This made it very easy to grab the ring in the driver controlled period. By cutting a slit in the shooting ramp and adding a barrier to keep rings from popping up (a problem that we consistently had), we were able to get a more predictable shot based on the speed at which the motor spins. This allowed us to do accurate tests and get the best shots possible.

Testing and Improving Robot Software

We wanted 2 different places we could shoot from, so we made 2 parts in the main TeleOps program that would do that. We made 2 different positions with different shooter speeds in case we went back in person and our alliance team or the other teams' robots were in the way. Even though we mainly use the first position for shooting to the highest tower goal right up to the line, we could also use the 2nd position for shooting power shots even though it isn't as accurate.

In the autonomous mode, we made many changes to our program, trying to make it as exact as possible. We had to change the amount of time that the robot moved for, and had to change the shooter speed multiple times in order to get it almost perfect. We dealt with a big problem when we changed out a battery with half power to a fully charged battery. The robot was now at full power, and all our speeds and timings were off. We had to change all of our numbers to fix this problem, but we got the hang of it eventually. We also had to make the robot turn 90 degrees during the autonomous mode so the wobble goal would land in the square fully.

A inside look at the shooter

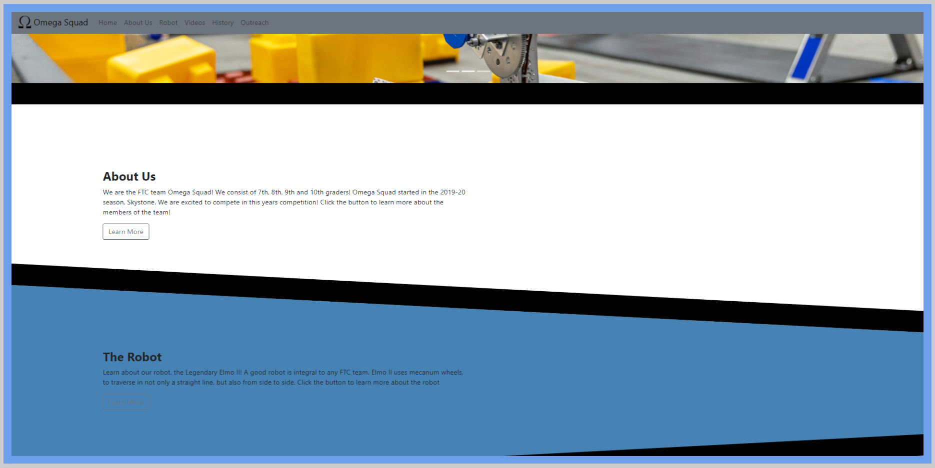


Most of the robot improvements involved improving the efficiency and accuracy of the shooter

Website Design

This is our team's second year in FTC. In our first year, we mainly focused on the robot aspect of the challenge. However, due to us being more experienced and having more time (due to Covid), we were able to start to branch out and focus on things other than the mechanical and program design. One of the main aspects that we knew we needed to make was a website for the team. We found three members of our team who knew enough about HTML and CSS to make a website. We met every Wednesday via zoom as discussed how we should make the website, what style it should be in, and what content should be in it. After a while, we settled on the existing design. Our main inspiration was the Garfield robotics website, which has the aesthetically pleasing slants in the pages. We split the pages up into our main tabs and worked on them. After we finished the code, we brought it together and used Amazon Web Services, or AWS, to publish the site on the domain name we had bought. We are very pleased with how the website turned out.

The Omega Squad website home screen



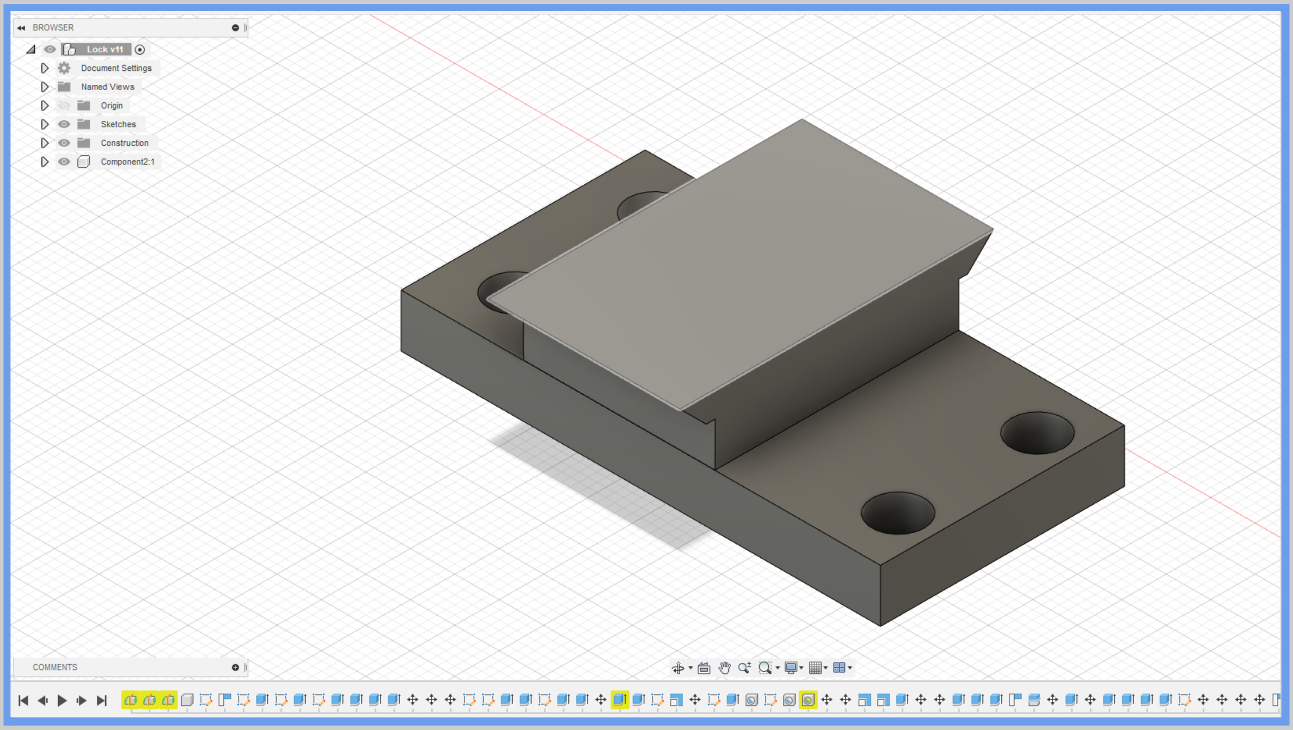
Link to the website: http://theomegasquad.org

Future Improvements and Ideas

As for next year, we hope to do a lot more outreach and events that could qualify us for awards and inform the community starting at the beginning of the season. In the coming years, we want to make use of CAD models, after seeing that building prototypes without a plan is not as efficient. We also plan to make use of 3D printing for unique pieces that we cannot purchase and that we need urgently. 3D printed parts will also help our robot stand out and look professional. We want to get more people to learn about FTC, bring in more members to our team, and get sponsored by businesses and STEM organizations in our community. We want to reach out to professionals in STEM fields around us to learn new skills to help us throughout the season.

In the coming years we need to elaborate on our daily log and have a plan that we develop in the beginning of the season. This plan should be specific to our needs and include the timeline we want to work by, fundraising and business plans, detailed prototype designs, and roles for each team member.We will also start posting frequently on social media about FTC to get more people our age interested. We will talk to businesses who might be interested in supporting our team.

The beginnings of CAD in Omega Squad



This model was designed in Fusion 360 to make a dovetail lock for decorative plexiglass on the sides of the robot