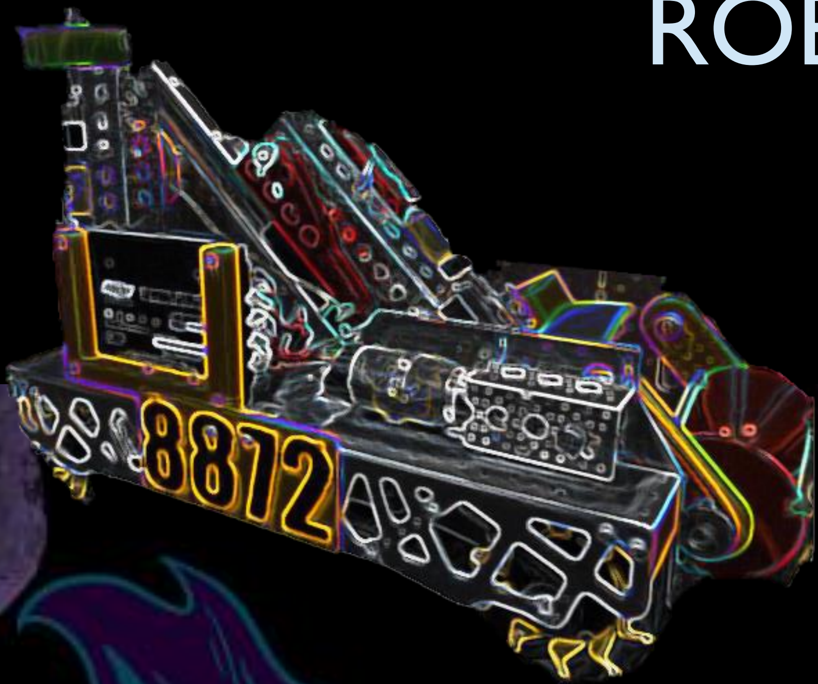
ROBOPOCALYPSE

2021-22 Engineering Portfolio



8 8 7 2

Contents

Sponsored By:

INTUITIVE SURGICAL®

TMK
TMK MANUFACTURING INC.
CNC MACHINING • WIRE EDM

KLA

Be agift to the community
Rotary Club Of Santa Clara

O2Micro® Breathing Life into Mobility

LOCKHEED MARTIN

NORTHROP GRUMMAN

TEAM DIVISION ORGANIZATION

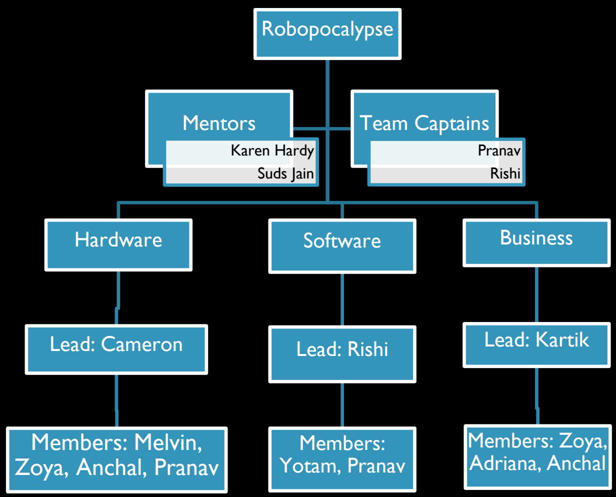


FIGURE 1: TEAM MANAGEMENT

*This is **Team 8872 Robopocalypse's** seventh year competing in FTC. Our team has improved substantially since our first year in 2014. Not only do we have our team, but we have also created a Robotics Club at our school for interested students to learn about the different elements of robotics.*

Our team, 8872 Robopocalypse, is divided into three main divisions: hardware, software, and business.

The **hardware** division handles the build phase of the robot. Responsibilities of hardware members include making a CAD of the robot chassis and mechanisms, building the robot, and communicating with software for testing the built mechanisms.

The **software** division handles the programming for teleops and autonomous.

The **business** division is responsible for budget management and planning local and international outreach events.

Although each of the divisions have their own role, our team runs based on communication - with the divisions and between the divisions. Members of hardware and software coordinate robot testing. Business members coordinate with all team members, so everybody engages in community outreach, as well as managing betting when ordering new supplies.

TEAM MANAGEMENT

This year started with a new challenge - returning from the Pandemic. Due to the pandemic, last year's team was significantly smaller in size, we only had one competing team. This year, however, with the return of in-person schooling, not only did we face a higher number of team members on 8872, but also a greater number of new members for our junior team. This added not only the challenge of managing a larger team, but also taking the responsibility of mentoring our growing junior team.

Our main tool of management was the continuation of using Trello. Trello is a web-based organization tool, which allows our team to use a Kanban-style organization. We have created separated Kanban boards for each division, with separate lists meant to represent tasks that need to be done (TO DO), Tasks that are currently being worked on by the team (IN PROGRESS), and tasks that have been finished (DONE).

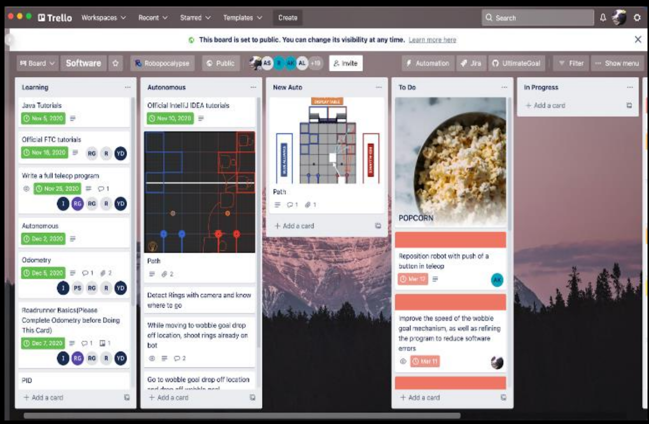


Figure 2: Task Assignment On Trello

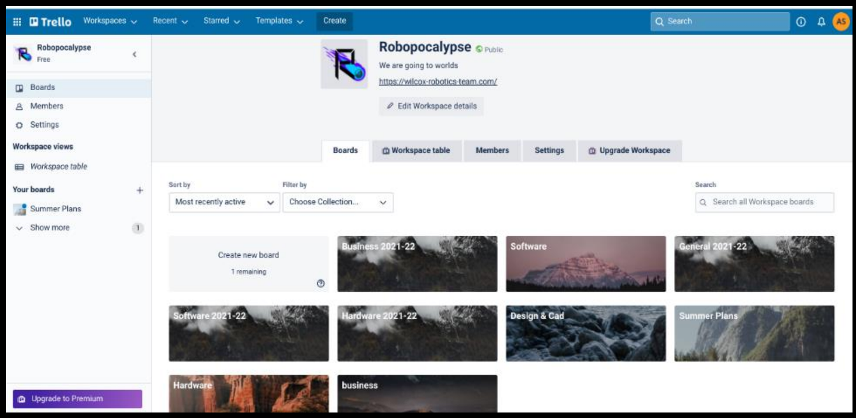


Figure 3: Trello Management for Divisions

REBUILDING TEAM 13190

Along with managing our team, we also had to mentor the newly recruited people. Prior to the pandemic, we had two teams: Team 8872 Robopocalypse and our sister team, 13190 Roblivion. However, during the Ultimate Goal we were unable to effectively have two teams while adhering to COVID protocols, so 13190 was temporarily disbanded. This year, though, we worked to rebuild the team through a variety of events and constant mentoring.

Figure 4: Team 13190 This Year

Our rebuild for the team this past season was very successful. We recruited members for the team by presenting and demonstrating our Ultimate Goal robot, collecting sign-ups at our school's Club Fair, Freshman Orientation, and Back to School Night. From these events we recruited roughly 30 new members to 13190 and our robotics club.

Once recruited, the prospective members were given a briefing about FIRST and about our robotics program. We used slideshows created by our business team and followed by a Q&A session with our veteran members, allowing the new members to learn more about each division and ultimately choose what division they wished to work in. These members were then trained in skills such as software, hardware, and CADing, with daily project-based workshops under guidance of more experienced members. Some things those new members helped with were a small multi axis arm, our test chassis, and field setup. These projects were organized by our captains through IntelliJ Simulator, CodeAcademy, and Fusion 360, and progress was recorded through Trello.

MEMBER ENGAGEMENT

The main forms of member engagement included the Parade of Champions, several side projects, and learning opportunities.

In addition to the parade, we assigned our junior team with several side projects such as building, CAD, 3D printing, and programming for continuing to teach them and assist them in becoming valuable assets for their respective team.

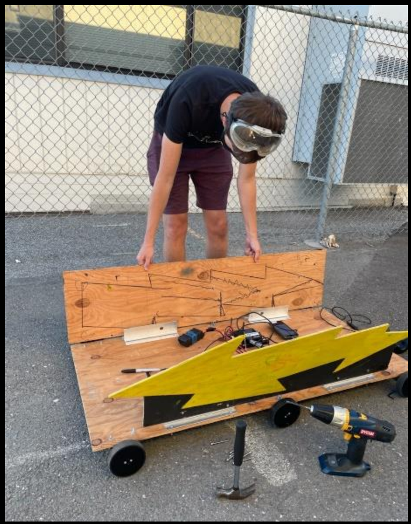


Figure 5: Junior Team Members Working on the Parade Robot

Parade Of Champions

We had the pleasure of participating in The Parade of Champions which is a traditional parade of the Santa Clara City. This was Wilcox Robotics' third year taking part in the parade. The Parade of Champions served not only as an outreach event, but also as a major form of starting to involve our new members. We used this opportunity to mentor the new club members of robotics to the basics of building a drive train. Our goal was to use a scrap plywood for the base of a large format robot with 6 direct wheels, using low rpm motors to reduce skidding and provide ample torque, and a large waving arm, with a support structure using an Actobotics C channel and a long chain to power the waving hand. We wanted the new members to have an opportunity to assemble and program a robot, while gaining the experience of facing failures in the process. Consequently, new members learned several hardware activities such as attaching motors and drilling, to software exercises, such as programming a basic tank drive, and learning about basic mechanical components. Members learned about the iterative process of prototyping. Although issues were faced, the event went well, and the community was impressed by our team.

LEARNING OUTREACH

Cruise Automation

- One of our most exciting experiences was at the Cruise facility. Cruise is a company that is building an autonomous ride service for people. They build autonomous cars which can drive around San Francisco gathering data and providing ride services. Our team was offered a unique opportunity to tour Cruise's office in San Francisco and see some of their cars and manufacturing facilities up close.

Guest Speaker: Daniel Myberg

- Daniel Myberg, from one of our sponsors, O2 Micro, visited us several times and taught us about management and structuring our team and mentored our teams, guiding us with organization.

Guest Speaker: Kevin Yamada

- Kevin Yamada taught us about different classifications of materials, such as metals, non-metals, ceramics, and plastics like thermoplastics and polycarbonates. We learned about different applications of materials. We also learned about different 3D printing materials and their features.

Guest Speaker: John Detriech

- We were also given the opportunity to have a speaker meeting with John Detriech from Northrop Grumman. He went over how to build reliable systems, effectively deal with problems, and manage a proper schedule. Additionally, he discussed the rule of three by relating it to how on a probe that was designed they had three computers that all chose a decision. If one of the computers disagreed it was taken offline. They ran extensive statistical models to calculate failure rate and this proved this to be significantly more fail-proof. We related this to our three forms of position tracking and altered our code to better fit his more reliable description.

Hosting a Qualifying Tournament

- On January 15, we hosted a tournament at our school. The tournament accommodated 14 teams and was held in person. Watching and interacting with the teams at the tournament helped us gain a better understanding of how to mentor our junior team and increase gracious professionalism. Throughout the tournament our team did tasks like robot inspector, field inspector, queuer, and field resetter.
- Doing tasks like robot and field inspection guided us to improve our robot and better comprehend the regulations of Freight Frenzy for our future competition. In addition, we got the opportunity to observe various other designs. Previously, our robot was able to go over the barrier. Our team spent a lot of time redesigning and building, to make sure that the robot is able to go over the black barrier. However, we noticed that the best and highest-scoring robots did not have that feature. In fact, they were smaller and went through the 13-inch gap. Seeing this, we redesigned our robot, making it smaller to 13 inches, and changed the side plates of our robot.

FIGURE 6: KEVIN YAMADA TALKING ABOUT MATERIALS USED IN ENGINEERING

FIGURE 7: DANIAL MEYBERG GIVING TEAM MEMBERS ONE-ON-ONE ADVICE

FIGURE 8: JOHN DIETRICH GIVING PRESENTATION

EXPANSION OF FIRST

Local Outreach with Elementary Schools

Our team collaborated with Santa Clara Unified School District to help local elementary school STEM programs in our school district. Twice a week, we taught them about science topics, and gave talks about engineering, robotics, and FIRST. This gave us an opportunity to connect with younger students in our community, while spreading the values of FIRST.

International Outreach – Israeli Outreach

One of our Israeli members was able to reach out to a school in Israel to help spread the ideas of robotics and FIRST. During the presentation, we taught the values of FIRST as well as the engineering process and how it was applied to both the hardware & software of our robot. Additionally, we provided an interactive activity involving Vuforia and Vision: how to identify the location of a digital image based on relative location on the screen. Finally, we wrapped up the presentation by demonstrating the functionality of robots and giving the students time to ask questions about FIRST and robotics. We had plenty of interesting and on topic questions and were generally satisfied with the event.

International Outreach – Korean Outreach

Along with the online local mentoring, we also wanted further our mentorship initiative, spreading the values of FIRST. With the help of one of our member's Korean connections, we are in the works of organizing a Korean Outreach program, mentoring a Korean high school's robotics club. We plan on holding virtual sessions teaching them about engineering, building, CAD, programming, financing, and team management. We intend to teach them about FIRST and robotics, so their club can have the opportunity to participate in next year's FTC challenge.

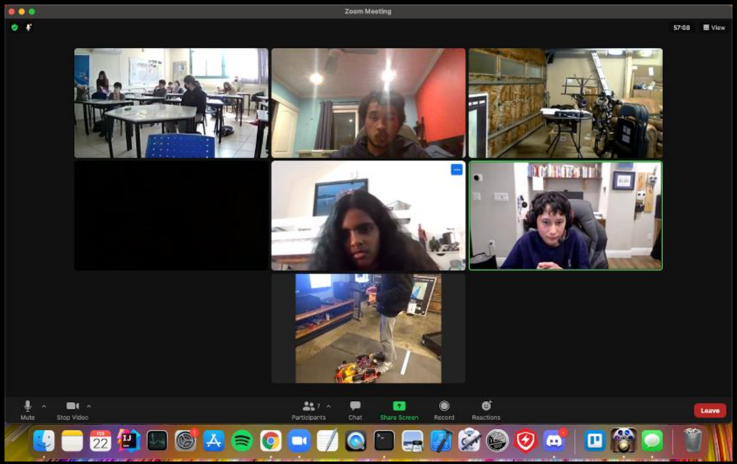


FIGURE 9: PRESENTING TO ISRAEL STUDENTS

Continuing Outreach in India and Malaysia

With the popularity of our India International Outreach last year, our team hosted another Indian Outreach at the same school to help introduce FIRST and robotics to an English-medium school in India. We created a comprehensive slideshow with an explanation of this year's competition followed by our general design philosophy and process. Later, we talked about the software element of the robot used to automate scoring with the implementation of a feedback loop to keep the robot in the correct location. Finally, we gave a live demonstration of our robot scoring points. In general, our team was very satisfied with the event as the audience remained engaged and had the opportunity to ask multiple questions.

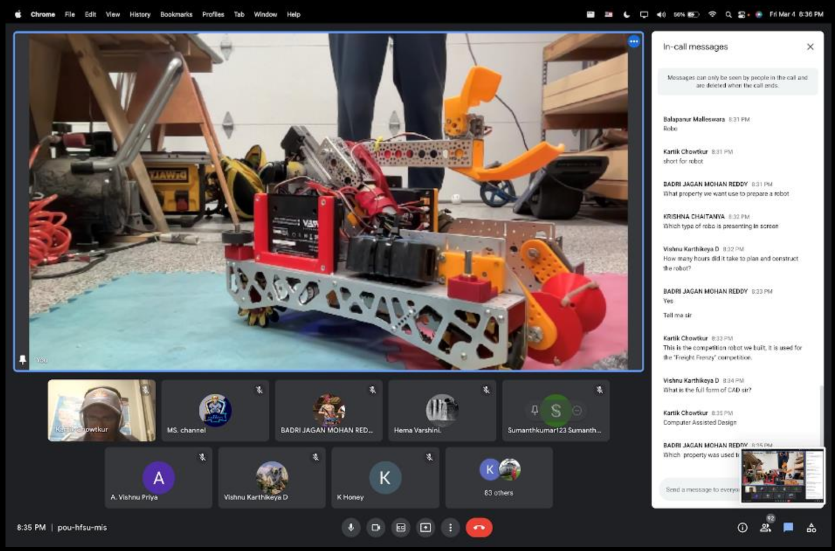


Figure 10: Presenting to Indian School

Additionally, we decided to continue our collaboration with Tutoring Malaysia, an organization made to teach children in Malaysia English from a young age. Like the India outreach, we gave a quick presentation and followed up with a live demonstration, although this time we made sure to put things in a way that was understandable to all ages, as we were not sure of the knowledge base of our audience. We were generally happy with how the presentation went, as we had good questions from the audience, especially during the live demonstration portion.

(Collaboration with Sister Team 13190) Assisting Rookie Team Legot

*Because this was their first year participating in the FTC, we provided team Legot with a full field to test the functionality of the robot (crossing the barrier, grabbing and delivering freight). We also helped to identify core issues with their robot, such as an extended c-channel preventing them from crossing the barrier and provided ideas to apply the fixes. Finally, we mentored many of the members in 3D modeling through **Fusion 360,** teaching them 2D sketching with constraints, creating 3D bodies, and modeling with various FTC parts.*

Figure 11: Demonstrations to Team Legots

Assisting Rookie Santa Clara Teams - Team Zenith and Team Serenity

With our team CNC machine, we were able to produce plates and metal parts for Local Santa teams such as Team Zenith and Team Serenity. Additionally, we provided tutoring on designing plates and traded FTC resources (mecanum wheels, hex axles) that were low in stock.

FIGURE 12: OUTREACH TO SANTA CLARA TEAM

Lockheed Martin Presentation

To give our team a feel of how robotics is applied to the real world, we were able to listen to a presentation given by three guest Speakers from Lockheed Martin. They gave us an introduction to their experience working in the company as well as some of the current and past projects, such as the **Hubble Space Telescope,** the **Orion SpaceCraft,** etc. We were to also learn about features used to improve reliability of aircraft, such as the **Orion Heatshield,** and various uses of manufacturing. At the end of the presentation, we asked questions and got insight into how they manage their team, building and developing prototypes, and softwares used to simulate stress loads and create 3d models for different parts of the spacecraft.

Prototyping the Base Chassis

Ultra Chassis (Version 1)

Initial Goals and Requirements:

There were a few goals and design requirements that we had set before the first version of the Ultra Chassis. We wanted to create a chassis that could be used over *multiple* years and *adapt* to each season's needs. Another important feature we incorporated was adding plates on all four sides to prevent game elements interfering with wheels. Finally, we maximized the area in between the wheels for the mechanisms necessary for that year.

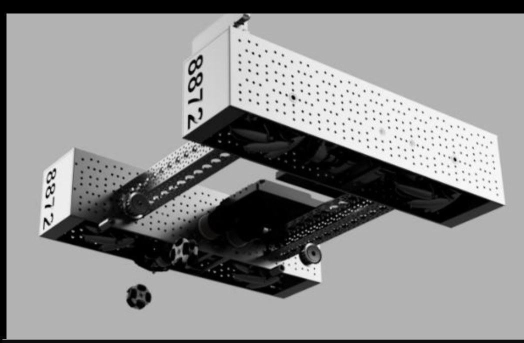


FIGURE 13: CAD OF OUR ULTRACHASSIS' S PREVIOUS VERSIONS

Initial Design Elements to Achieve Goals:

We accomplished the above design requirements through a few design choices. These include using a consistent **Actobotics-Inspired** hole pattern through the entirety of the chassis. This allows high levels of customizability, as we can mount various mechanisms easily. In order to keep the wheels from interfering with game elements we added large flat plates on either side of the wheels and closed off the wheels from all sides except the bottom. Finally, in order to create as much room for mechanisms we pushed the belts to the outside of the bot and moved the motors inwards.

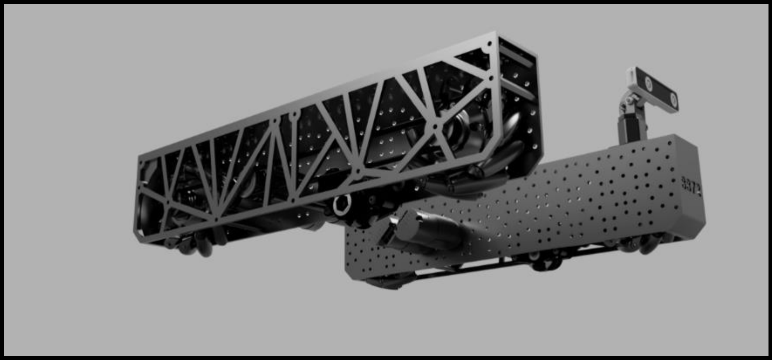


FIGURE 14: FIRST REVISION OF ULTRA CHASSIS

Revision after Initial CAD:

This first version itself went through several revisions. The first version used live axles where the wheels were attached to an axle that spun and was supported by two bearings. This worked but took up loads of space giving less room for the intake and making it difficult to support the shafts axially. To take care of this we used dead axles where the wheels spun around a standoff which was fixed in place. This was much more compact, giving us the desired space and made it easier to manufacture.

We then replaced Actobotics mecanum wheels with Gobuilda mecanum wheels. This was because the Actobotics mecanum wheels were not reliable and were much harder to work with. The Gobuilda mecanum wheels were the gold standard for accuracy and reliability and helped with keeping the bot compact with their slim profile.

Initial Build:

Eventually, we got around to building the chassis right before our competition and managed to complete it just before the competition. Surprisingly, the CAD work seemed to pay off, as the chassis went together without any major struggles. However, we were unaware that the chassis would prove useless for this year's competition.

Ultra Chassis (Version 2)

After the competition release, our previous ultra chassis would not be viable as for the new competition. The main problem was that the chassis did not have the required clearance for the pipes. Whenever the robot tried to clear the pipes, the pipes would hit the front block of the chassis or get high-centered in the middle. In order to account for this, we had to cut clearances in chassis, lowering the wheels relative to the chassis, and moving motor positions. Although the solutions were simple it meant redesigning the entire chassis. However, we took the opportunity to learn and make a robot that was more durable and reliable.

We added a large, rounded cut out to the front and back of the bot along with a long slice out of the middle in order to make sure that we would never get stuck on the pipes. To further improve the design, we added a flat plate to the top of the robot to help mount mechanisms easier. However, we would continue to tweak the chassis well into the season to increase tolerances and make the robot easier to work with. In general, the design of the chassis taught us that investing time into projects before the competition release is not always a good idea and designing/assembling something many times is the only way to get a perfect mechanism.

Open Odometry

Odometry is a position tracking tool we use to update the robot's position even when it is *skidding*.

Previous versions of our custom designed odometry experienced certain types of failures

- **Version 1 (Original)** - NO spring-based system, couldn't always remain in contact with the ground
- **Version 2 & Version 3** - Parts of wheel scraped the ground, wheel was angled toward the side which resulted in bouncing and incorrect position tracking
- **Version 4** - Tried out *Open Odometry*, eliminated the two previous problems (pod perpendicular to the ground during position tracking

FIGURE 15: CAD VERSION 1

FIGURE 16: CAD VERSION 2

FIGURE 17: CAD VERSION 3

FIGURE 18: CAD VERSION 4

Due to the complexity of mounting the open odometry pod with the additional obstacles of the barrier, our team made the decision to instead opt toward using V-Slam (Visual Slam) with the T-265 Camera as the main form of position tracking.

Intake Prototyping

During the brainstorming portion of the engineering process, we came up with many different ideas that we wanted to test. This included designs such as the entrapment star, a compliant wheel intake (like Ultimate Goal), a roll cage, a surgical tubing intake, and multiple claw designs. In order to filter these designs to figure out what design we were going to implement, we 3D modeled and built a simple test bench where we could easily swap intake designs to see what would work the best.

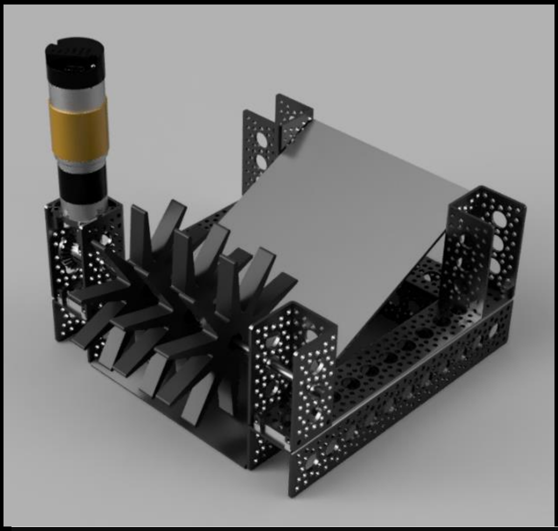


FIGURE 19: CADDING OF THE ENTRAPTION STAR DESIGN - WE USED THIS OVERALL INTAKE

We quickly began to filter down the long litany of designs we had brainstormed, looking at estimated effectiveness, practicality, and ability to build. We ruled out the compliant wheel intake because it wouldn't work with both types of freight and wouldn't be particularly effective in picking up freight off of the field. We also omitted all claw designs, as we knew from previous experience in Ultimate Goal that lining up and picking up freight with a claw normally increases cycle time and reduces points.

Unfortunately, we had to scrap even more designs simply because we didn't not have and couldn't make the materials necessary to build them. One popular design that many other teams were using was one or two high RPM Andymark Entraption Stars mounted in front of a ramp, which we had to rule out because they were out of stock from all vendors. We considered 3D printing the stars instead but reasoned that they wouldn't be as rigid as the store-bought product and would thus bend instead of moving freight into the robot.

Finally, we were left with two ideas: the roll cage and the surgical tubing intake. The roll cage consisted of two circular wheels with strips of an elastic material in between them which met the freight. The surgical tubing design consisted of many pieces of staggered surgical tubing of equal length attached perpendicular to the axle. After building the test rig and conducting multiple long days of test trials, we concluded that the roll cage would work better for our robot's design as it more easily got around to the backside of the freight, allowing us to lift the freight from the ground easily and in turn score more points.

FIGURE 20: ROLL CAGE INTAKE PROTOTYPE

Decision made, we modeled the design in Fusion 360 and made the final intake using custom CNC machined aluminum side plates and ramp. To make the intake work effectively and still fit in the 18-inch robot size requirement, we attached the roll cage on a hinged platform, using a pulley and belt system to power it.

Top Mechanism Prototyping

Prototyping Top Section of Robot

This year, we wanted to prototype two independent designs for the top section of our robot to figure out which would be the most reliable and efficient.

FIGURE 21: DESIGN 1 - LINEAR-RAIL BASED DESIGN

- Used a large linear rail with a vertically rotating arm mounted to it
    - The linear rail created most of the movement, and the arm was used to hit different shipping hub heights / move the game elements backwards
    - This eliminated the need for a linear actuator
- The linear rail was powered by a stationary Gobuilda motor that spooled in a cord to move the arm and its mount up and down the rail

- This was much simpler than a stacked linear slide design, as the cord only had to go around one pulley and didn't need a retract
- The weight of the arm mount, which was ½ inch aluminum, was heavy enough to slide down the rail on its own

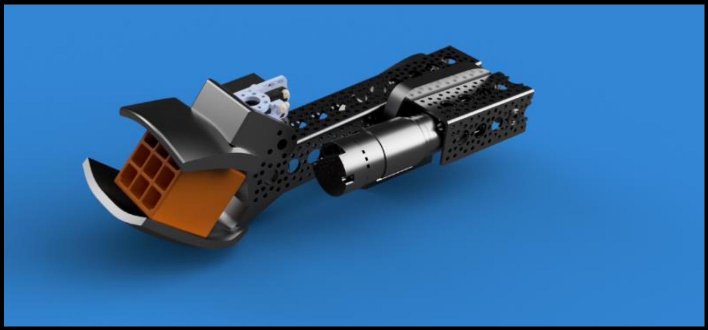


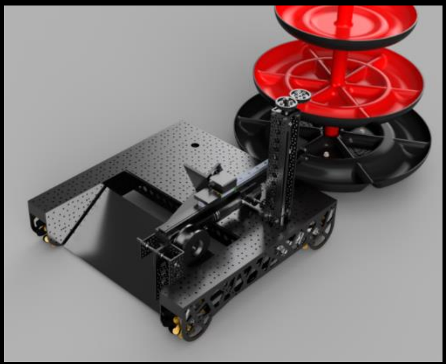
FIGURE 22: CAD OF THE ARM

- The arm was powered by a moving Gobuilda motor attached to a 2:1 bevel gearbox
  - The gearbox converted excess speed into more torque
  - The entire motor moved, so the cables had to be moveable but also not get in the way of any other mechanism
  - There was already quite a bit of stress on the quarter inch D shaft axle

FIGURE 3: CAD OF THE CLAW PART OF THE ARM

FIGURE 23: CAD OF THE CLAW FROM THE ARM

- The arm itself was just a 1 ½ by 1 ½ inch Actobotics C channel, which was modular and lightweight
- The game elements were held in place on the end of the arm by a clamshell-like cage for the blocks powered by a single high-torque servo
- The entire mechanism was 3D printed, as its complex curves could not have been machined any other way.
  - This was weaker and heavier than we would have liked, but it worked out well in the end.
- We went for a clamshell-like shape because we thought it would make it easier for a block to fall into it, while staying secured while they were lifted up towards the shipping hubs
  - The entire mechanism attached to the end of the arm using square nuts



The game elements were pushed into the clamshell using a large intake / ramp at the front of the bot. In order to deposit minerals in each stage of the Alliance Shipping Hub, we planned to use a linear actuator with a telescopic slide above it

FIGURE 29: CAD OF LINEAR-ACTUATOR DESIGN

- There would be a pivot point at the front of the robot
- The height would be set by a linear actuator at the back of the robot

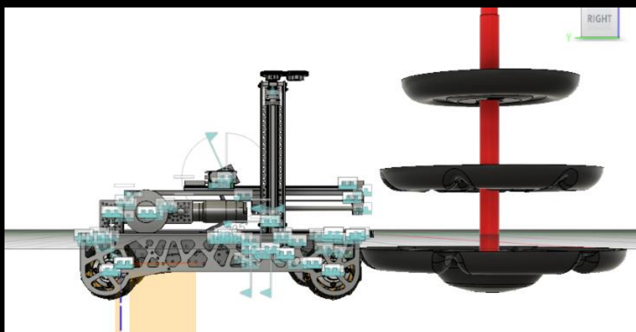


FIGURE 30: EACH OF THE BLUE SIGNS REPRESENTS THE TYPE OF JOINTS USED. COMPARED TO THE LINEAR RAIL DESIGN, THERE WERE MORE MOVING JOINT LOCATIONS.

Benefits with Design

- *Linear Actuator allows us to fine tune the location for dropping the freight in each location for the box*
- *Less bulky & heavy compared to use the linear rail mechanism*

Issues with the Design

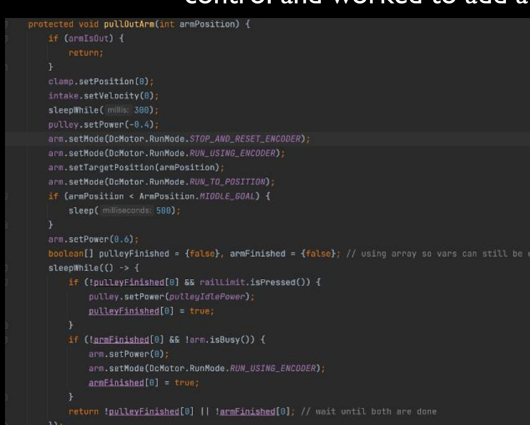
- *Used a multi string-based pulley system (linear rail & telescopic slide rail*
- *Lack of mounting holes on telescopic slide rail & Linear rail, made it difficult to mount pulleys to route string*
- *Complicated to build in the time span we had allocated for hardware*
- *With the linear rail moving diagonally, would have to keep the actuator in a constant position throughout each run for the intake system to function*

Based on our Gantt Chart and Allocated Time we had left to build for the Top Section of our robot, we decided to omit building this design instead opting for the Linear Rail system which was simpler.

Software

Teleop Programming & Automation

Our Teleop program has many software features to assist the driver and reduce possible errors. We realized early on that having multiple mechanisms on the robot would be hard for a driver to control and worked to add automations and failsafe to every mechanism to reduce load on the driver and error rate.



```
protected void pullOutArm(int armPosition) {
    if (armIsOut) {
        return;
    }
    clamp.setPosition(0);
    intake.setVelocity(0);
    sleepWhile( millis: 300);
    pulley.setPower(-0.4);
    arm.setMode(DcMotor.RunMode.STOP_AND_RESET_ENCODER);
    arm.setMode(DcMotor.RunMode.RUN_USING_ENCODER);
    arm.setTargetPosition(armPosition);
    arm.setMode(DcMotor.RunMode.RUN_TO_POSITION);
    if (armPosition < ArmPosition.MIDDLE_GOAL) {
        sleep( milliseconds: 500);
    }
    arm.setPower(0.6);
    boolean[] pulleyFinished = {false}, armFinished = {false}; // using array so vars can still be
    sleepWhile(() -> {
        if (!pulleyFinished[0] && railLimit.isPressed()) {
            pulley.setPower(pulleyIdlePower);
            pulleyFinished[0] = true;
        }
        if (!armFinished[0] && !arm.isBusy()) {
            arm.setPower(0);
            arm.setMode(DcMotor.RunMode.RUN_USING_ENCODER);
            armFinished[0] = true;
        }
        return !pulleyFinished[0] || !armFinished[0]; // wait until both are done
    });
```

Our Automation

Arm automation

- With a press of a button our robot can clamp down on a payload, move the payload towards the shipping hub with the help of a linear slide, and drop off the payload by moving our arm and releasing the clamp. It can then automatically retract the arm as well to prepare for intaking the next item.

- The robot is fully drivable during this time and other mechanisms are still operable to allow efficiency when cycling freight.

3 Fusion Backup System

In case a position tracking system fails or becomes unreliable over time, we build a system that makes use of 3 methods of position tracking.

1. **Motor Encoders(Main)** : Most reliable, but position tracking degrades over time or when going over bumps
2. **Navigation Targets(Secondary)** : Unreliable in certain robot positions in the field (unable to identify targets), provides perfect pose when identified. Embedded with motor encoders to apply position correction when identifying either 4 navigation targets
3. **Intel T265 Camera(Tertiary)**: Tracks pose for long distances reliably, takes time to update positioning, embedded to encoder system to remove noise (similar to **kalman filter**)

Navigation Targets

In order to detect a navigation target, the camera lens has to be facing toward the nav target. To serve the dual purpose of identifying barcodes with tracking pose through navigation. The logitech camera is mounted on a 270-degree servo.

Intel Camera (T-265)

- With the inclusion of barriers in this year's competition, we decided to use the intel camera for tracking the position compared to odometry, which would have experienced a position tracking error
- During the summer, we found **V-Slam (Visual Slam)** was able to track the position better compared to using motor encoders, as well as track the heading more accurately than the imu
- During implementation we encountered an issue with the setPose() functionality provided by the **FTC 265 Wrapper** for the **Intel Realsense Camera**
    - To account for the issue, we used a translation for the position
- **Dual Camera (Logitech C920 Pro)** - used to create custom object detection using built in Vuforia software provided by the FTC APK

Fail-safes

Throughout the process of developing and testing our robot, we realized there are many single points of failure. To account for this, we make use of limit switches and sensors to ensure mechanisms are not moved past their breaking point and can always be easily reset.

Trajectories

Using a simulator, we were able to create and test trajectories without using the actual robot. By doing so we could make the trajectories more quickly and efficiently. Furthermore, we created multiple trajectories to

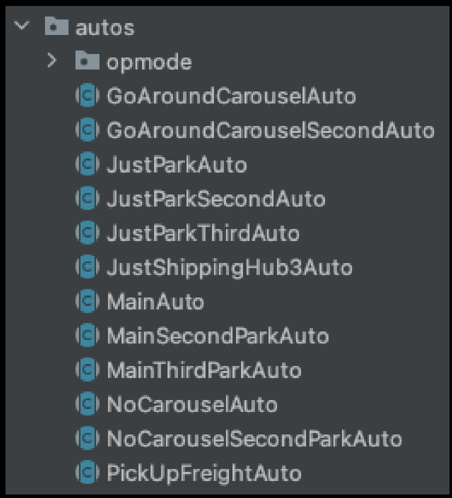


FIGURE 4: TRAJECTORIES

accommodate for any course of action that the other team may take during the competition.

```java
private void armLoop() {
    // Functionality for gamepad 1
    if (gamepad1.dpad_up && !lastDpadState && armIsIn()) {
        poolFuture = pool.submit(gamepad1.a ? () -> fullArmSequence(ArmPosition.TOP_GOAL) : () -> pullOutArm(ArmPosition.TOP_GOAL));
    } else if (gamepad1.dpad_right && !lastDpadState && armIsIn()) {
        poolFuture = pool.submit(gamepad1.a ? () -> fullArmSequence(ArmPosition.MIDDLE_GOAL) : () -> pullOutArm(ArmPosition.MIDDLE_GOAL));
    } else if (gamepad1.dpad_down && !lastDpadState && armIsIn()) {
        poolFuture = pool.submit(gamepad1.a ? () -> fullArmSequence(ArmPosition.BOTTOM_GOAL) : () -> pullOutArm(ArmPosition.BOTTOM_GOAL));
    } else if (gamepad1.dpad_left && !lastDpadState && !armIsIn()) {
        poolFuture = pool.submit(this::retractArm);
    }
    lastDpadState = gamepad1.dpad_up || gamepad1.dpad_right || gamepad1.dpad_down || gamepad1.dpad_left;
```

```java
protected void pullOutArm(int armPosition) {
    if (!armIsIn()) {
        return;
    }
    clamp.setPosition(0);
    intake.setVelocity(0);
    sleepWhile( millis: 300);
    pulley.setPower(-0.4);
    arm.setMode(DcMotor.RunMode.STOP_AND_RESET_ENCODER);
    arm.setMode(DcMotor.RunMode.RUN_USING_ENCODER);
    arm.setTargetPosition(armPosition);
    arm.setMode(DcMotor.RunMode.RUN_TO_POSITION);
    if (armPosition < ArmPosition.MIDDLE_GOAL) {
        sleep( milliseconds: 500);
    }
    arm.setPower(armPower);
    boolean[] pulleyFinished = {false}, armFinished = {false}; // using array so vars can still be effectively final
    long endTimeout = System.currentTimeMillis() + 5000;
    sleepWhile(() -> {
        if (!pulleyFinished[0] && railTopLimit.isPressed()) {
            pulley.setPower(pulleyIdlePower);
            pulleyFinished[0] = true;
        }
        if (!armFinished[0] && !arm.isBusy()) {
            arm.setPower(0);
            arm.setMode(DcMotor.RunMode.RUN_USING_ENCODER);
            armFinished[0] = true;
        }
        if (gamepad1.dpad_left || gamepad2.dpad_left) {
            arm.setPower(0);
            arm.setMode(DcMotor.RunMode.RUN_USING_ENCODER);
            pulley.setPower(0);
            retractArm();
            return false;
        }
        if (System.currentTimeMillis() > endTimeout) {
            arm.setPower(0);
            arm.setMode(DcMotor.RunMode.RUN_USING_ENCODER);
            pulley.setPower(pulleyIdlePower);
            RobotLog.addGlobalWarningMessage( msg: "Arm/pulley movement timed out. Check the rail limit switch or arm encoder.");
            return false;
        } else {
            return !pulleyFinished[0] || !armFinished[0]; // wait until both are done
        }
    });
}
```

```java
protected final void retractArm() {
    if (armIsIn()) {
        return;
    }
    clamp.setPosition(1);
    sleepWhile( millis: 400); // drop element
    pulley.setPower(0);
    arm.setPower(armPower);
    if (arm.getCurrentPosition() < ArmPosition.MIDDLE_GOAL - 100) {
        sleepWhile( millis: 500);
    }
    if (arm.getCurrentPosition() < ArmPosition.TOP_GOAL + 100) {
        sleepWhile( millis: 500);
    }
    if (!railBottomLimit.isPressed()) {
        pulley.setPower(0.5); // retreat downward
    }
    boolean[] pulleyFinished = {false}, armFinished = {false}; // using array so vars can still be effectively final
    long endTimeout = System.currentTimeMillis() + 9000;
    sleepWhile(() -> {
        if (!pulleyFinished[0] && railBottomLimit.isPressed()) {
            pulley.setPower(0);
            pulleyFinished[0] = true;
        }
        if (!armFinished[0] && armLimit.isPressed()) {
            arm.setPower(0);
            arm.setMode(DcMotor.RunMode.RUN_USING_ENCODER);
            armFinished[0] = true;
        }
        if (gamepad1.dpad_left || gamepad2.dpad_left) {
            arm.setPower(0);
            arm.setMode(DcMotor.RunMode.RUN_USING_ENCODER);
            pulley.setPower(0);
            retractArm();
            return false;
        }
        if (System.currentTimeMillis() > endTimeout) {
            arm.setPower(0);
            arm.setMode(DcMotor.RunMode.RUN_USING_ENCODER);
            pulley.setPower(0);
            RobotLog.addGlobalWarningMessage( msg: "Arm/pulley movement timed out. Check the limit switches or arm encoder.");
            return false;
        } else {
            return !pulleyFinished[0] || !armFinished[0]; // wait until both are done
        }
    });
    arm.setPower(0);
}
```