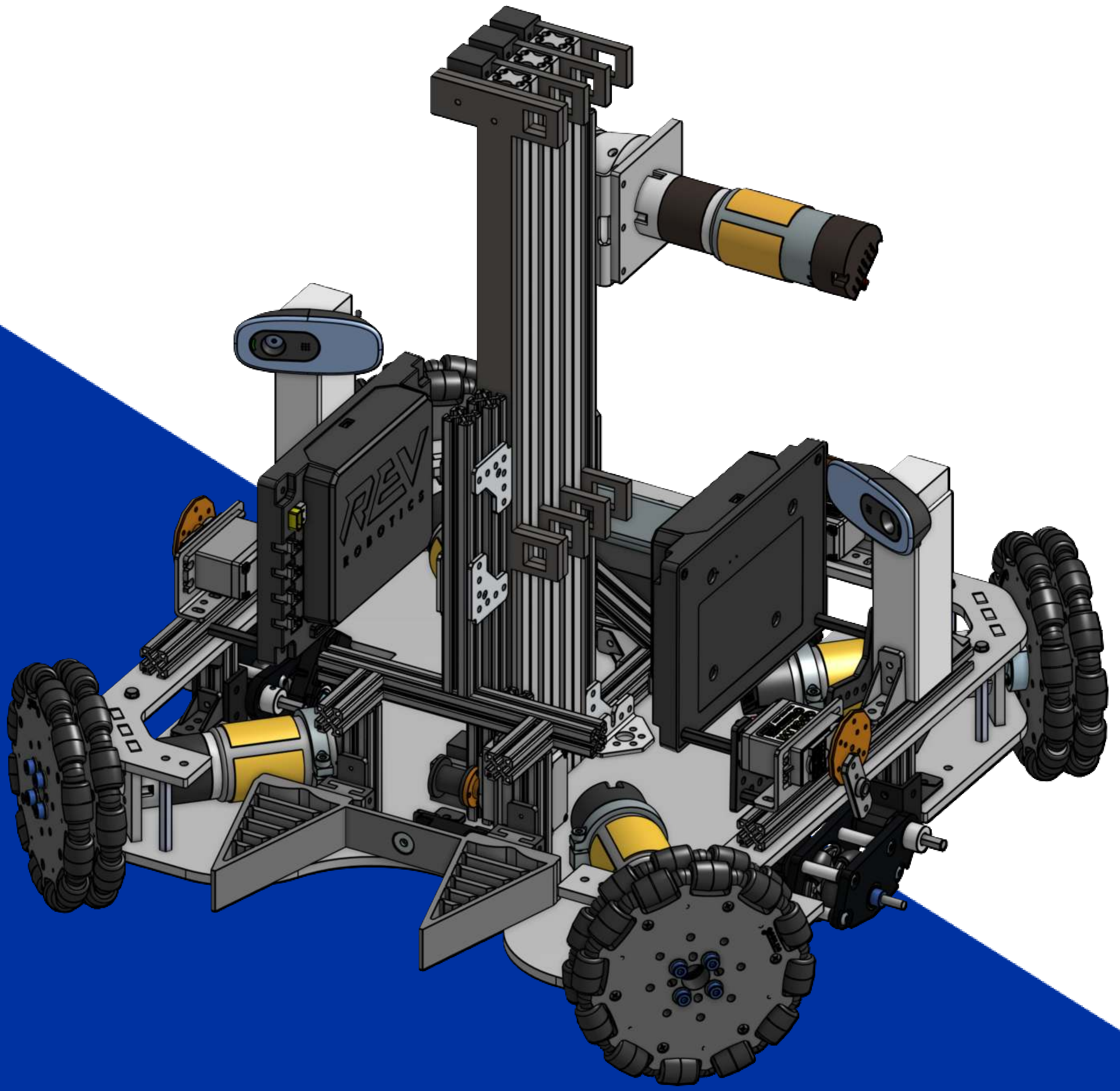


SUMMIT ROBOTICS

TEAM 14365

OUR TEAM

MEMBERS

Logan Miller
Team Captain, Programming Lead, Driveteam

Alex McDulin
Team Co-Captain, Build Co-Lead, CAD Co-Lead, Pit Lead

Mason Moyle
Team Co-Captain, Eng Notebook Lead, Driveteam

Nate Bailey
Build Co-Lead, Driveteam

Johnathan Breazeale
Scouting Lead, Build, CAD, Programming

Meghana Curran
Build, Driveteam

Cole Hiller
Programming, Pit Team

Rovio Chen
Programming, Pit Team

Joey DiPaola
Programming, Build, Scouting

Airy Barnes
Outreach Co-Lead, CAD Co-Lead, Scouting

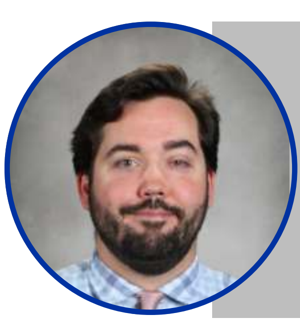
Kelly Qiu
Marketing, Outreach, Scouting

Gracie Sanders
Programming, Scouting

Coleton Chadwell
Build, Scouting

Runako Muvirimi
Programming, Scouting

MENTORS



Ed Buyniski
Coach; Computer Science Teacher

Michael Moyle
Coach; Software Engineer Manager at Crowdstrike

Jack Crane
Studying ME & CS at SLU; team alumni

Jacob Locke
Studying ME at University of Cincinnati; team alumni

Connie McDulin
Chemist; Assisted with Documentation Process

Mike Crane
Previous Coach; Industrial Sales & Project Management

DESIGN PROCESS

Problem Definition
We gather as a team to discuss which challenges our strategy will focus on.

Concept Development
Each member, regardless of subteam, contributes component ideas on notecards that are decided as a team.

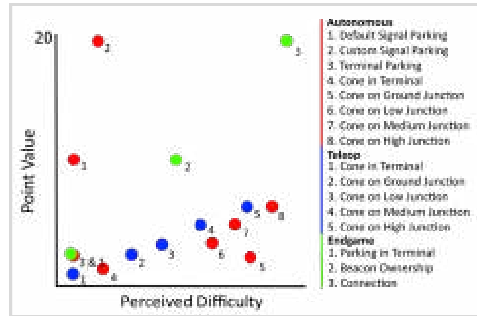
CAD & Prototyping
Make prototypes out of cardboard, then design with CAD to see how our ideas work in a 3D environment.

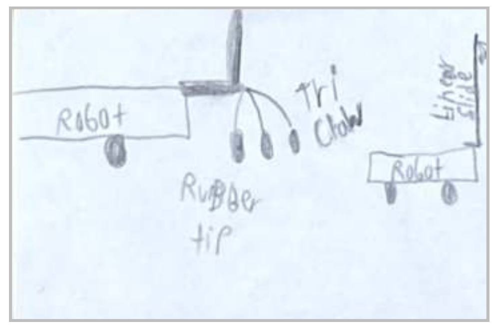
Build
We turn our prototypes into real components.

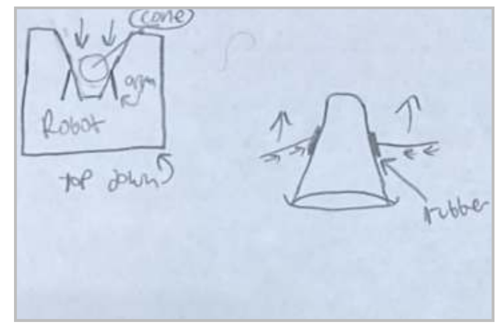
Testing
Evaluate pros and cons of our design and see what may improve functionality.

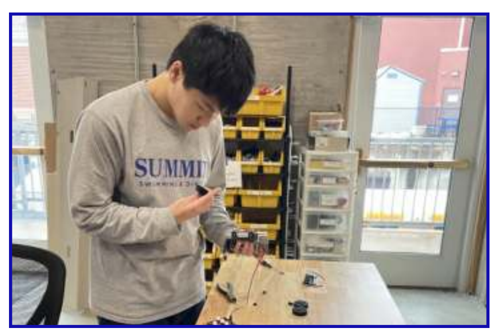
Iterate
If we determine that there is a potentially more functional solution, then the component is redesigned.

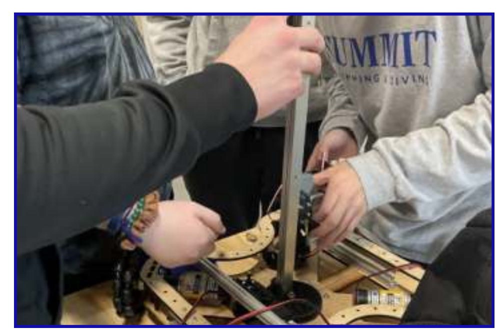
Compete
Observe how our robot interacts with others and the game in real life to develop ideas so we can further refine our robot.

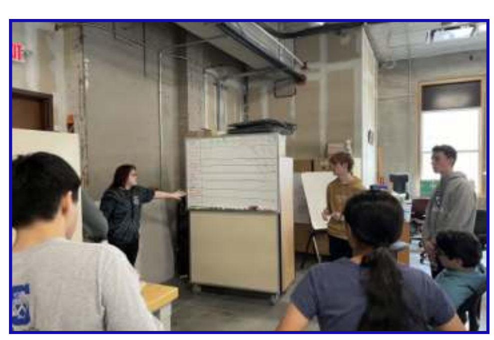


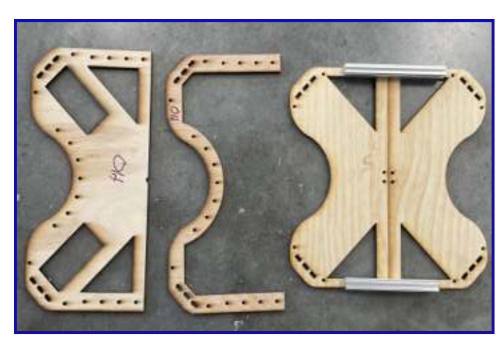








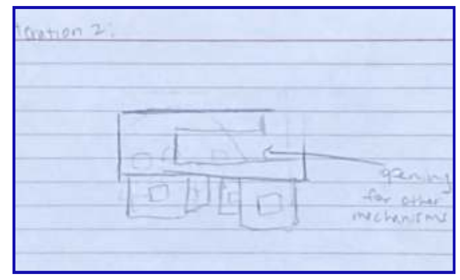


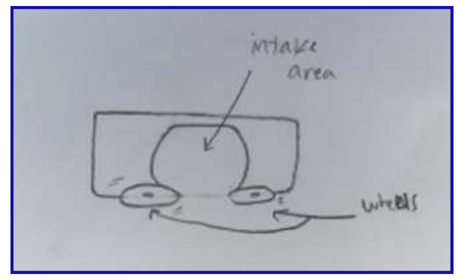


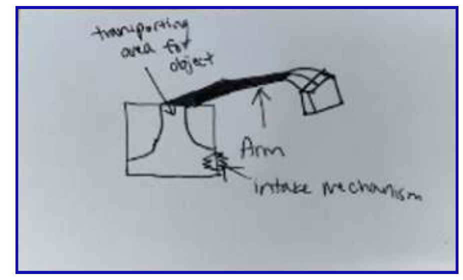
BRAINSTORMING

Ideation

- We begin by researching concepts to find the best way to achieve each task.
- Every member of the team, regardless of subteam, draws their ideas on notecards, which is followed by discussion of the pros and cons of each idea.
- We go through the mechanisms of the robot and review all notecards to ensure we have the most effective design for each component.

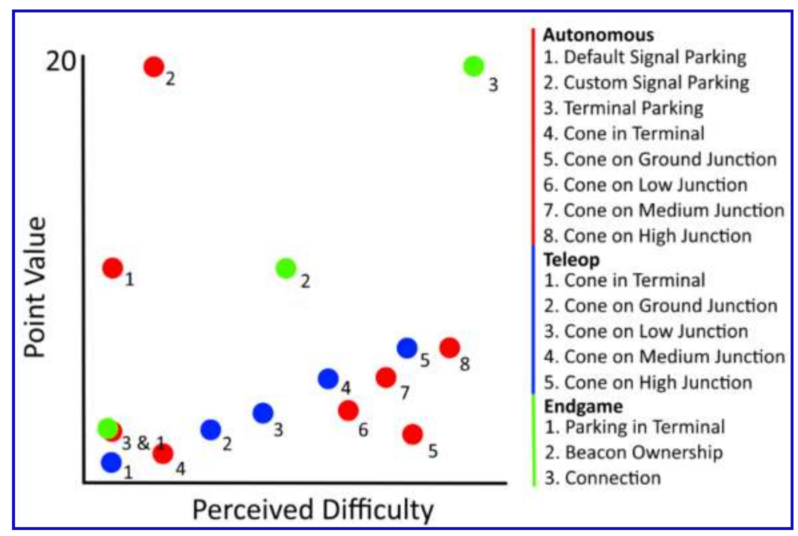






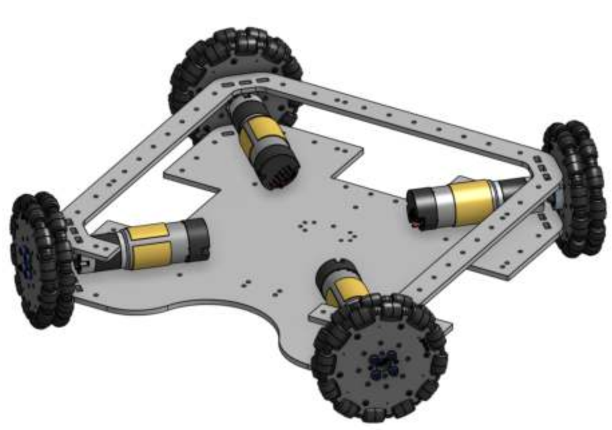
Prioritization

- We discuss as a team and decide which scoring opportunities to pursue. No matter the subteam, everyone participates in the idea process.
- We make a graph which compares point values and perceived difficulty for each phase of the match.



Prototyping

- Cardboard and wood prototypes are developed to visualize how different systems work together.
- Our Glowforge laser cutter allows for rapid prototyping.
- The designs are then transferred into CAD.



DRIVETRAIN

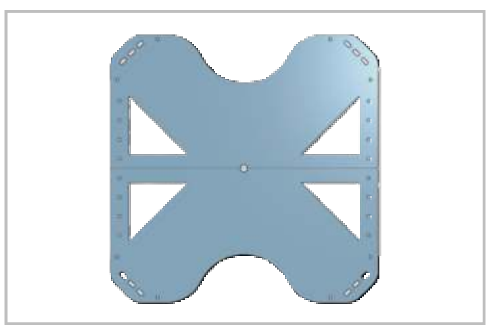
Our Design Strategy

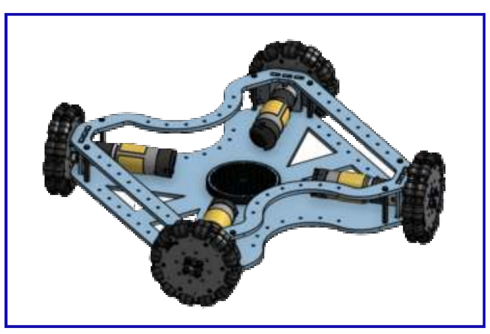
We opted to use a Holonomic X-Drive base with two omni-wheels offset at a 45° angle for our drivetrain.

- The X-Drive allows us to go √2 faster in the cardinal directions and when turning.
- The robot moves √2/2 slower when going diagonally, but with the way the poles are positioned in a grid, it is never optimal to move in this manner.
- We decided to go with a 14”x14” robot frame to easily navigate the narrow field.
- We used in-house laser-cut wood to allow for rapid prototyping and easy modifications.

Once we were confident in our design, we had it machined from aluminum to improve rigidity and overall strength.

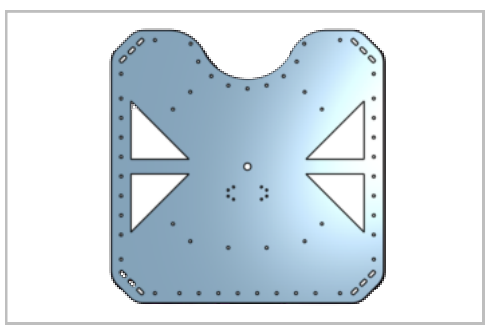
Version 1.0

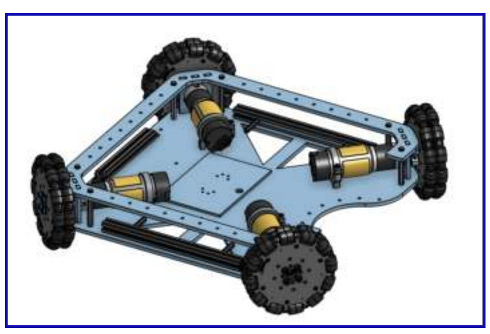




Basic X-drive drivetrain with various holes for standoffs. Lazy suzan prototype in the middle and two circular cutouts.

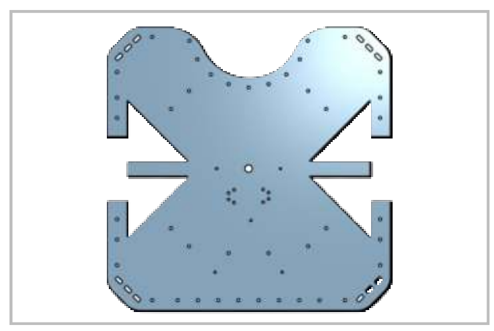
Version 2.0

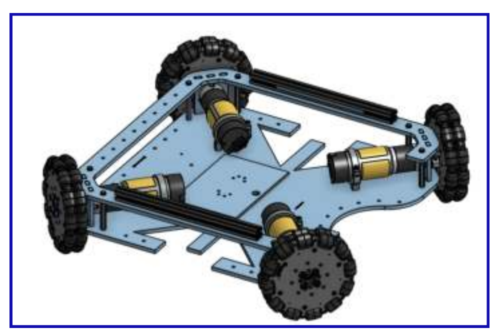




Removed second cutout once we decided to remove the lazy susan (see next page). The holes for the linear slide were added.

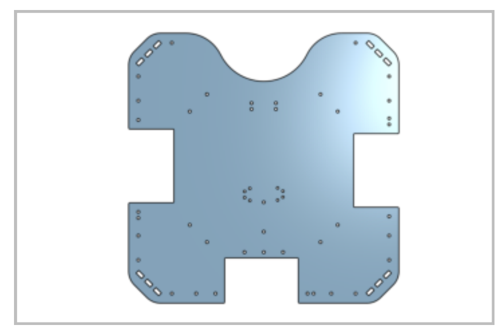
Version 3.0

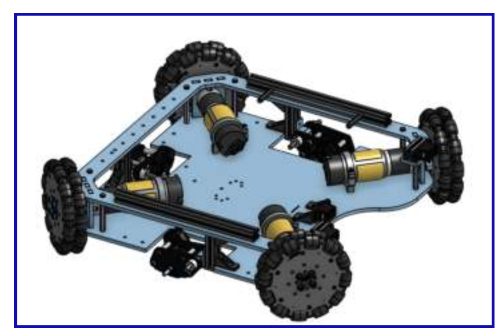




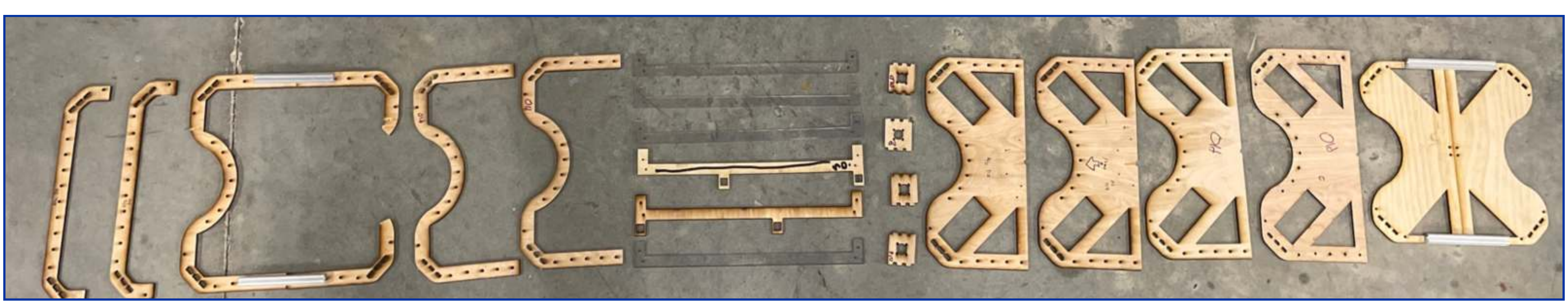
Added cutouts for odometry wheel implementation.

Version 4.0





Removed unnecessary holes and changed cutout size for odometry to account for a new design.



LINEAR SLIDE

Our Design Strategy

- We use a three stage Misumi linear slide with custom wood slide inserts and custom 3D printed slider caps.
- The slide can reach all pole lengths, ground junctions, and terminals.
- Wires are encapsulated in a wire sheath kept close to the robot and away from essential components through badge clips.
- Extrusion is attached on front and back to secure the slide & claw.
- Custom covered spool (to ensure the string won't become tangled) is driven by a goBuilda 5202 435rpm motor.

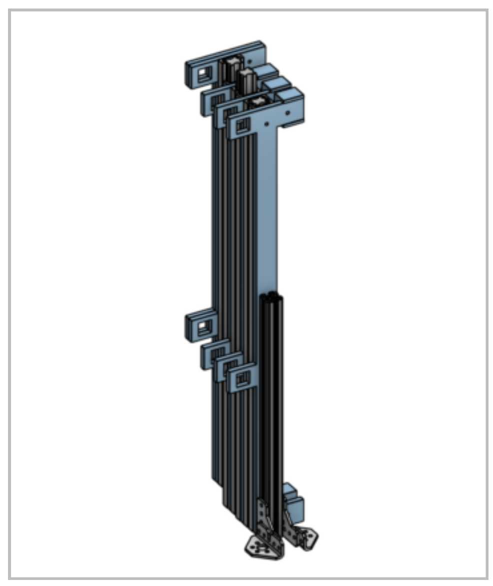
Prototype

Cardboard linear slide used to visualize different claw designs and to ensure that our robot fits within the height requirements.

Version 2.0

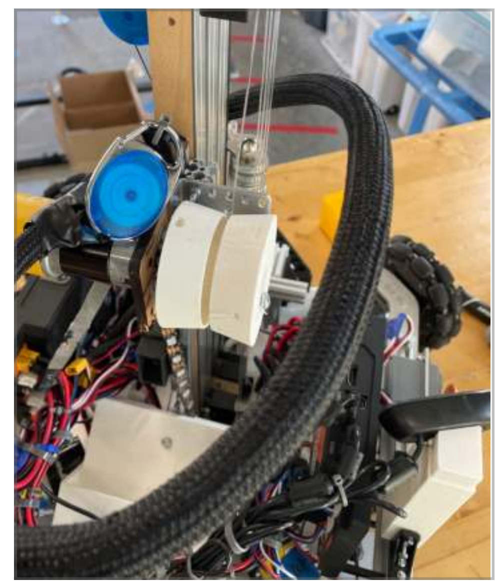
Prototype translated into CAD for design vision; custom inserts developed.

Version 3.0



Added another stage to the linear slide to make sure we can reach all pole heights; decision to scrap lazy susan (description below).

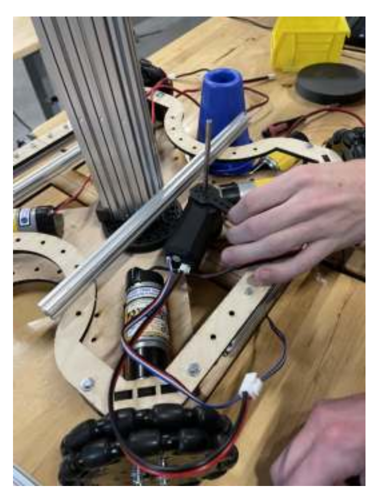
Version 4.0



Added black mesh wire sheath attached through retractable blue badge clips; removed wire management squares. Added white covered spool to keep string from tangling.

Lazy Susan

Our original plan was to have a 180° spinning "lazy susan" component on our robot. Going into the development, we knew this would be a challenge. Once several prototypes were constructed, the decision was made to scrap this component as there was not enough time to perfect the design and because of severe stability issues when raising the linear slide. We compensated for not having the lazy susan by leaning into the turning speed and cardinal direction speed of our robot.

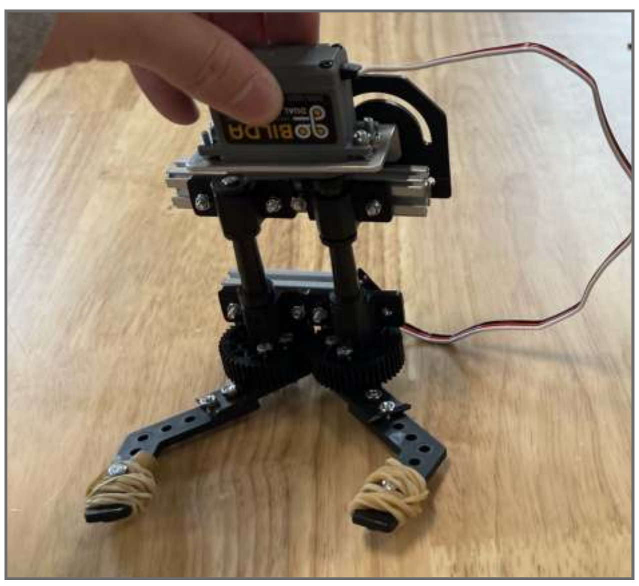


CLAW

Our Design Strategy

- We took inspiration from a fin ray design and modified it to fit our needs.
- Printed in TPU filament by a past team member at a Saint Louis University research laboratory.
- Smaller spool was attached to open and close the claw, the diameter of spool was adjusted to provide the gripping strength we needed.
- Cut tubing was added to the inside of the claw to get a secure grip and to have more grip surface area.
- Many different materials were used to prototype and test different designs (cardboard, wood, plastic, metal).

Version 1.0



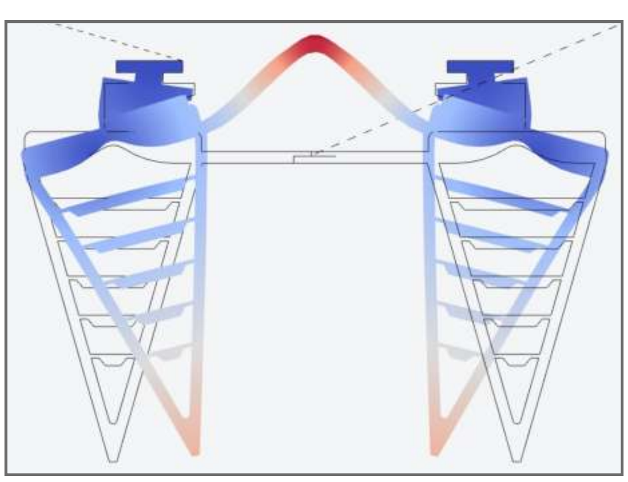
We developed a prototype based off our notecards containing REV mounting brackets, gears, and rubber bands for traction.

Version 2.0

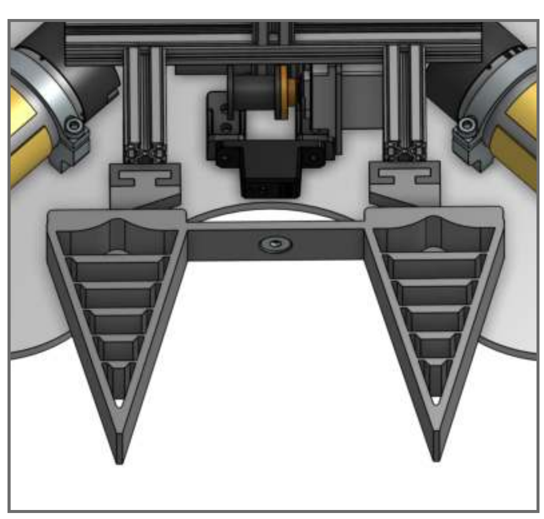
The previous design was not as precise as we needed, so we designed a new claw. In this new design, we used a cut compliant wheel for traction, bent metal pieces to get the shape, and added foam to get a secure grip.

Version 3.0

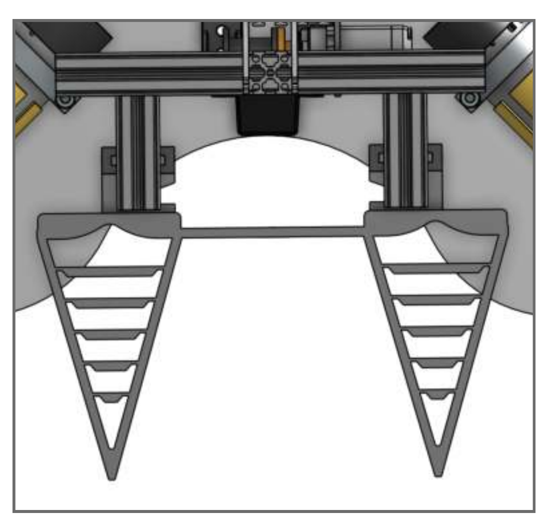
As the previous design did not have the versatility we needed to grab the cones in different positions, we chose our new silicone claw shape because its form fits to the cone and has a strong grip. We use a small spool to open and close the claw.



We performed a soft-body physics simulation to see the angle of the mounts necessary to get the grip we needed.



Front view of claw.



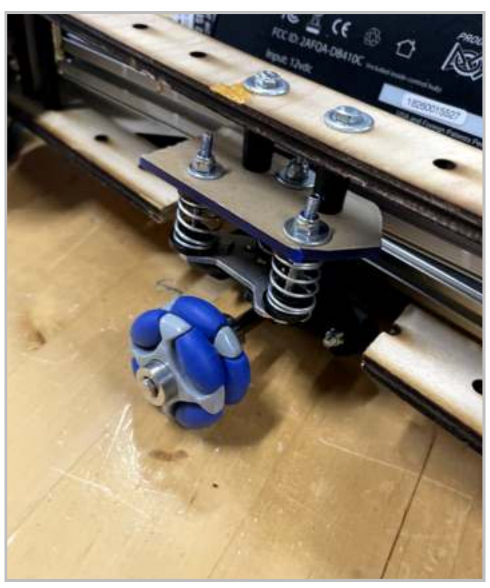
Top-down view of claw.

SENSORS

Odometry Wheels

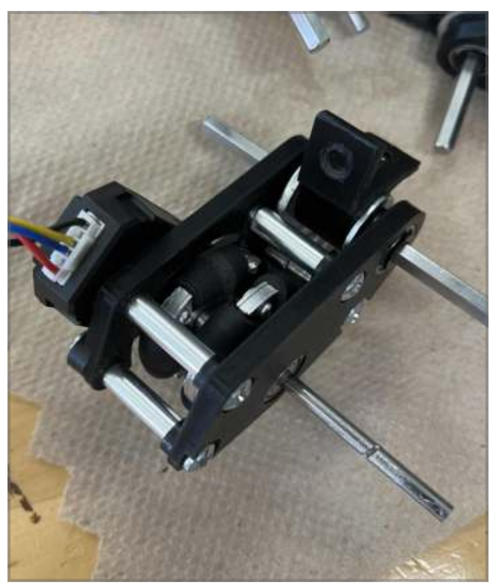
- Open Odometry design derived from OpenOdometry designs made by Team 18219.
- The design was adjusted to fit our standard wheels, bearings, and standoffs.
- We switched from a standard 5mm REV shaft to a 4mm round shaft to improve wheel alignment.
- Made sure the springs were the proper length and force to drive the odometry into the ground.
- We added a servo to retract our odometry during TeleOp so go over the ground junctions without risk of breaking.

Version 1.0



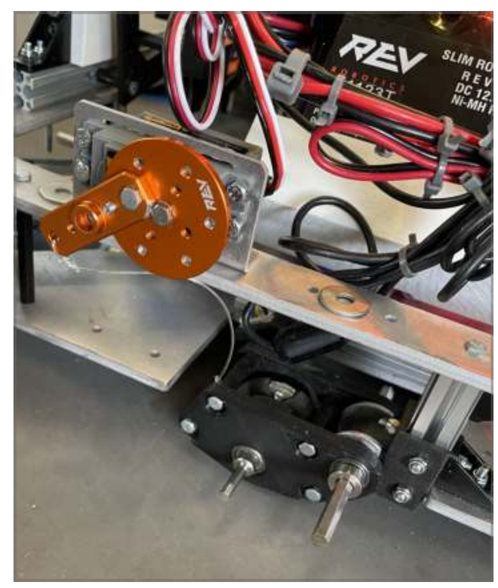
Acrylic and metal panel connected by two springs to drive the wheel into the ground. This design had skipping issues and was not parallel.

Version 2.0



New open odometry design with two 3D printed panels connected by goBuilda standoffs. Used modified bearing pillow block to allow the odometry to move up and down.

Version 3.0



We switched to REV standoffs to get the exact distance we needed between panels. We added a servo, servo horn, and servo arm to retract our odometry during TeleOp.

Webcam Mount Process

Our first version for the webcam mount had a small 3D printed piece holding the camera. We soon realized that this design would not work as the design was not strong enough. The second version had a more robust connecting system to ensure no breakage. Our third version, which we currently use, has two L brackets on each side for stability. We also shaped the camera mount to fit the bottom of the camera.

PROGRAMMING - INTRO

To perform the objectives required in PowerPlay, we chose a number of strategies, technologies, and systems. Our programming enables us to synthesize the data from twelve independent systems, allowing our robot to self-navigate, drive, and take real-time actions as required by the rules and objectives. All components help the robot during Autonomous, TeleOp, and Endgame play periods.

In addition to fulfilling the requirements of PowerPlay, the team identified and managed the following Non-Functional Requirements:

- **Performance** - This is a major requirement due to time constraints. We had many challenges optimizing processing time with RoadRunner and dual webcam logic. We tested, re-tested, and optimized our code and algorithms in order to ensure fast operation on the field. Our optimizations allowed for a dual webcam refresh rate of 147 ms at a high accuracy and reliability, as well as six seconds saved during autonomous through optimizing Roadrunner trajectory building.
- **Reliability** -After early-season connectivity issues, the programming team partnered with the build team and programming mentors. After code reviews and benchmarking, we determined the issue was related to grounding. After installing grounding equipment, we have had no issues in the last eight matches.
- **Usability** - We incorporated LED lights and the gamepad rumble to reflect the robot's real-time state. The lights will change color to let the driver know the robot's situation; see "TeleOp" on page 11 for details. This allows the drive team to keep their eyes on the robot at all times. This assists the team during autonomous tuning and TeleOp driving.
- **Maintainability** - We modulized our code to cut down on duplicate code and to help stay organized when adding new features or debugging existing solutions.
- **Quality and Testability** - We have a number of routines that test the accuracy and reliability of our webcam and localization systems. Additionally, we have incorporated Pair Programming strategies when preparing and writing code. This helps improve code quality, reduce the chance of logic errors, and help to educate future programmers.

Code Details

Architecture - We fully utilize the object-oriented nature of Java, using modulization and helper classes wherever possible. Our code has a Robot class to hold all the basics of our robot, including Vision, Sensors, Delivery, and Drive subsystems. Each contain relevant methods and objects to control those systems. This ensures that development time is optimized, as a change to one subsystem will reflect in all programs, Autonomous or TeleOp, where it is utilized.

Version Control - We utilize GitHub in our development process. New features are developed on a separate branch to be tested before being merged back into our competition main branch. This also enables members to work simultaneously on different experimental features without breaking the project at large, as well as reverting back to any prior version. Logan, Rovio, and Cole have direct access to the repository on their personal accounts, and all other members of the team have access through the shared Summit Country Day Robotics account saved on the computers in our workspace and lab.

Third-party Libraries

Library	Version	Author	License
Road Runner	0.5.5	ACME Robotics	MIT License
FTC Dashboard	0.4.4	ACME Robotics	MIT License
EOCV AprilTag Plugin	1.1.1	OpenFTC	Custom, as defined in source
EasyOpenCV	1.5.3	OpenFTC	Custom, as defined in source

Sensors and Usage

Autonomous	Autonomous & TeleOp	TeleOp
Dead Reckoning Wheels Motor Encoders PID Settings	Webcams LED Lights	Distance Sensor IMU Gyroscope

PROGRAMMING - AUTO

Programming enables our robot to process multiple, real-time sources of data that will influence the logic and commands needed to score PowerPlay points. This data allows our robot to process location, distance, and the precision required for success. Our autonomous program utilizes data from three sensors to determine its autonomous path and commands. These sensors are:

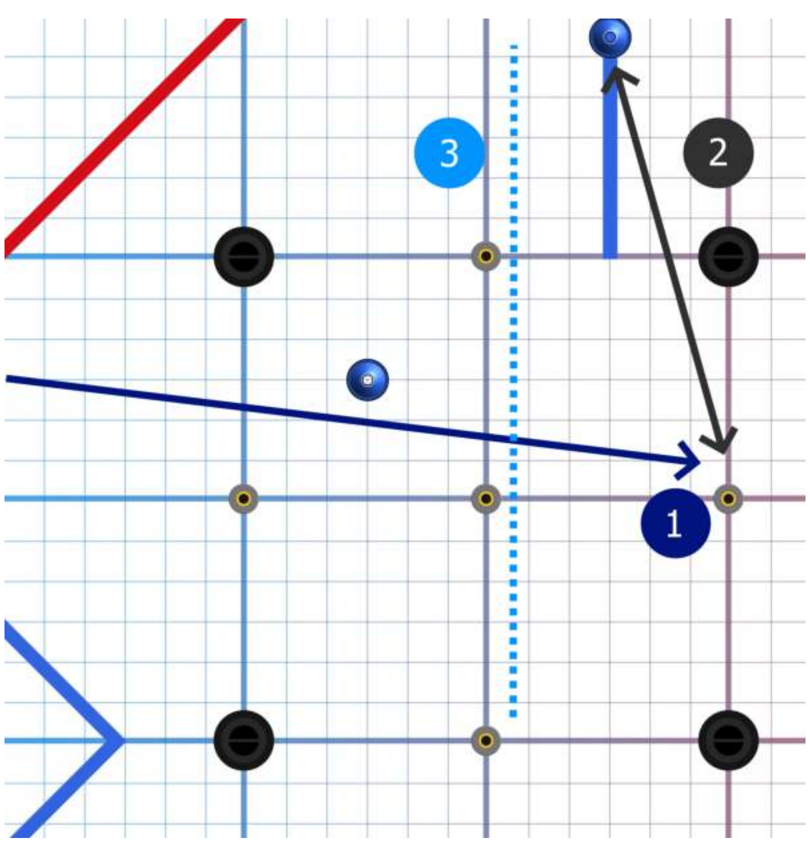
1. **Webcams** - Two cameras used for position triangulation; see 'Dual Webcam' section on page 12 for details.
2. **Motor Encoders/PID Control** - We utilize two distinct kinds of PID values- Velocity and Follower. Velocity PID works on the scale of individual motors by comparing the ideal and actual velocity returned by the motor encoder. This ensures that the motor quickly adjusts to any discrepancy in the velocity it should be going using Proportional, Integral, and Derivative modifiers. The Follower PID works on the scale of the entire robot and it ensures the robot quickly adjusts to the ideal path using the same modifiers, comparing the robot's ideal position and actual position returned by the dead reckoning wheel localization. These two forms of PID ensure that, both on an individual motor and broad field scale, the robot follows its path accurately and is able to respond to any unexpected forces.
3. **Dead Reckoning Wheels** - Our three dead reckoning wheels ensure consistent and reliable path following. By comparing the change in position from the dead reckoning wheels to each encoder's feedback(as described above) we are able to more accurately calculate our robot's current location on the field.

As the match is starting, we hit the 'initialize' button to tell the robot to build motor velocity trajectories. Then once the match starts, we hit 'play' and the robot follows the sequence until we need live webcam feedback for the cone drop. After the live-built cone drop, the robot will use the Follower PID and Dead Reckoning localization to transition itself to back to the pre-built path. The robot will transition from pre-built and live-built modes depending on the goal - movement and cone grabbing, or dropping on the pole. This ensures a consistent, versatile, and performant autonomous within the thirty-second time allotment.

Autonomous Path

1. Preloaded cone drop on high junction
2. Cone pickup, same drop
3. Parking based on custom sleeve

Functions on all 4 possible starting positions



leads to . . .

Autonomous Results

10 pts - 2 Cones on High Junction

20 pts - Parked in zone denoted by custom element

+ 10 pts - Placed cones are scored once again within TeleOp

30 PTS IN AUTO CATEGORY, 40 PTS TOTAL EFFECT ON THE MATCH

ensures consistency in...

Autonomous Reliability

Webcams: Pole Alignment

- Custom yellow isolation and pole center detection processes
- Custom formulas that provide angle and position alignment measurements

Roadrunner: Path Following

- Velocity PID constraints for consistent and reliable motor behavior
- Dead Reckoning Wheels and Follower PID for reactive and consistent adherence to the set path

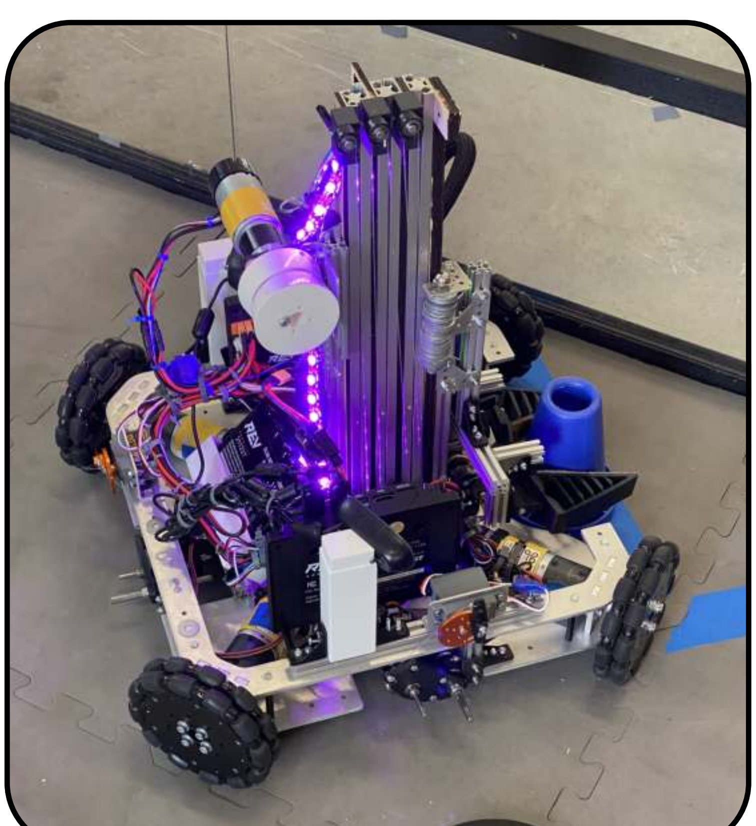
PROGRAMMING - TELEOP

Our team's TeleOp strategy is going for junction ownership across the field, as we have identified this as the most efficient method to score points. During drive practice, we have identified several critical areas of driver input that can slow down our cycle time and, as a result, decrease how many points we score. These critical moments include:

1. **Cone Pickup** - Getting a secure grab from both the alliance substation and the side cone stacks
2. **Cone Delivery** - Ability to secure a good drop as well as quickly manuever the slide between height levels
3. **Robot Maneuverability** - Traveling the field quickly as well as having the ability for fine, precise movements

Our programming assists TeleOp performance through realtime sensor feedback during the situations above. The specific sensor use cases, as well as which critical area listed above they assist with, are as follows:

Driver Challenge	Solution
#1 - Cone Pickup It is difficult to see when a cone is sufficiently far enough in the delivery area for an ideal, secure pickup	Our **LEDs** light up PURPLE and our **gamepads rumble** when our slide is retracted, determined by **motor encoders**, our claw is open, and our frontal **distance sensor** detects an object within 2.5 inches.
#2 - Cone Delivery It is difficult to tell when we are in line with the pole for a successful drop	Our **webcam algorithms** run when our slide is extended, determined by **motor encoders**, which turns our **LEDs** YELLOW when we are not in line with a pole, and makes them GREEN, as well as makes our **gamepads rumble**, when we are in line with the pole.
#2 - Cone Delivery It is difficult and time consuming to manually adjust the position of the slide to each pole height	We utilize the slide's **motor encoder** to use a D-Pad toggle to automatically adjust the slide to the heights indicated by how many times the D-Pad was pressed up or down, with manual control still enabled for fine adjustments.
#3 - Robot Maneuverability Field-centric driving makes cone pickup and drop off challenging, while robot-centric driving makes maneuverability of the field difficult	We have a button which is able to automatically swap between robot-centric and field-centric driving to cover both of their weak areas. We use the **IMU** to constantly track our angle and adjust the direction of our wheel movements based on its outputs when we are in field-centric mode.



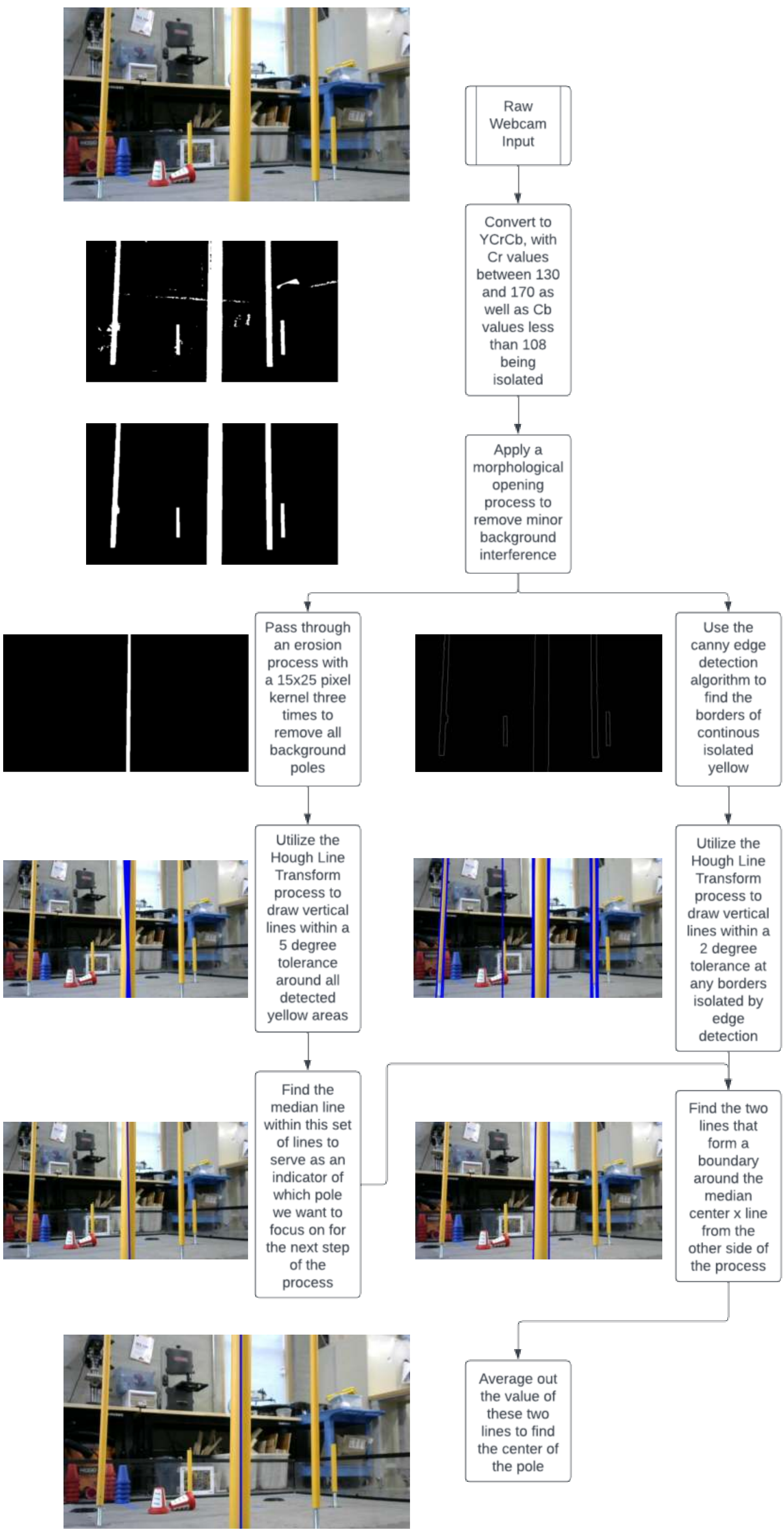
Left: Here, our LEDs are turning purple to signal that there is a cone far enough in the delivery cut-out for a successful grab, utilizing our frontal distance sensor.

Right: Here, our LEDs are turning green to signal that we are in line with the pole, utilizing our webcam stereo processing algorithm (detailed on the next page).

PROGRAMMING - DUAL WEBCAM

OpenCV Pole Isolation

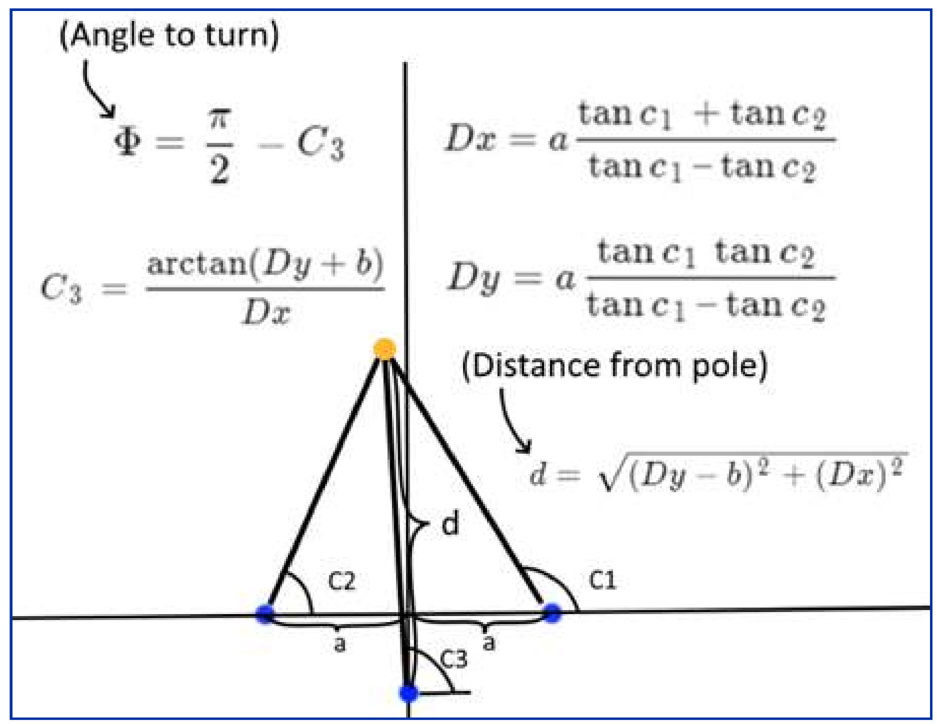
The first step in our pole alignment process is isolating the closest pole to find its center x coordinate pixel value in each camera. We do this through a fine-tuned, in-depth process utilizing OpenCV, detailed in the flow chart below:



Stereo Processing

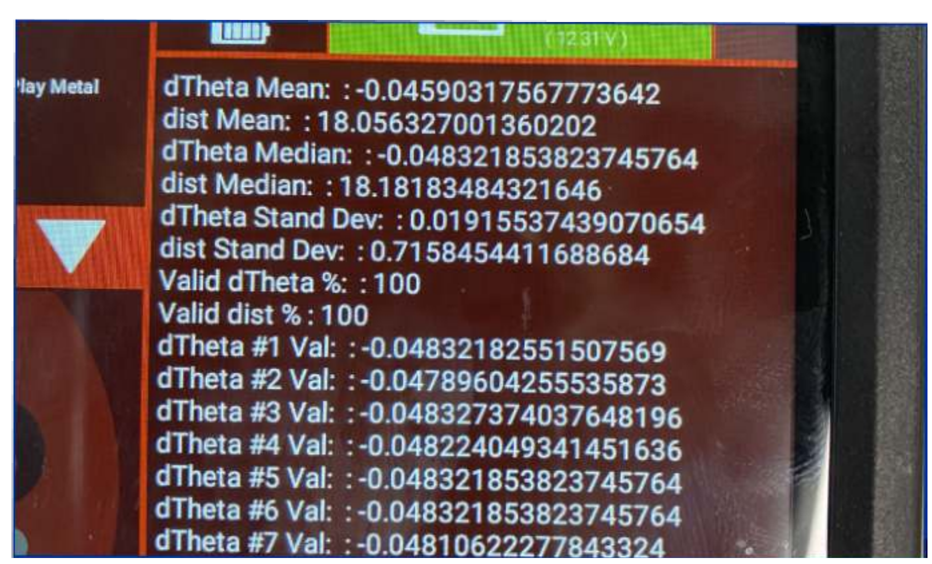
Once we have the x coordinate of the center of the pole for both webcams, we plug this information into the custom formula detailed below to give us an angle we need to turn as well as a distance we need to move forwards to have an ideal cone drop onto the center of the pole.

This process works due to having two webcams and comparing the difference in position of the pole between each camera to determine where the pole is located within a 3D space relative to the center of our robot. Below is a diagram detailing these calculations.



Telemetry Analysis

In order to develop and test the accuracy of this system, we created an advanced telemetry test program which runs the process n amount of times and provides mean, median, standard deviation, and valid calculation percentages across those n number of trials. This has guided our yellow isolation tuning and been a helpful indicator of how reliable this system is. Below is an image of some of this telemetry feedback for 100 trials:



FINANCES

Team Income

Source	Amount
The Summit Country Day School	$4000.00
Summit Community Donations	$1000.00
Total	**$5000.00**

Team Spendings

Item	Amount
FIRST Team Registration & Assorted REV Parts	$1434.06
FTC Field Preorder	$526.06
REV Robotics Parts	$156.42
GoBuilda Parts	$1547.02
SendCutSend Metal Drivetrain Parts	$136.24
FedEx Shipping & Presentation Posters	$800.00
Other Assorted Parts	$290.65
Total	**$4890.45**

Bill of Materials

5	goBuilda 5202 435rpm Motors (w/ encoders)	8	goBuilda Omni-Wheels
3	goBuilda Torque Servo	12	goBuilda Patter Spacer
2	Logitech C270 Webcams	4	goBuilda Sonic Hub
3	REV 2m Distance Sensors	8	45mm M3 Standoff
1	REV Control Hub	4	40mm M3 Standoff
1	REV Expansion Hub	2	3D Printed Camera Mount
1	Drivetrain Bottom Panel (aluminum)	4	3D Printed Motor Clamps
1	Drivetrain Top Panel (aluminum)	3	Misumi SAR 340 Linear Slides
4	Custom Motor Panels (aluminum)	5	Custom Linear Slide Inserts
16	goBuilda 30mm Socket Head Screw	5	3D Printed Linear Slide Caps
13	Linear Slide Screws	1	3D Printed Battery Holder
4	Motor Clamp Screws	6	REV Metal Bracket
50	M3 Assorted Screws	13	Assorted REV Extrusion
50	Assorted Bolts	1	Custom Sillicon Claw
30	M3 Washers	2	Assorted Metal Pieces
1	Super Speed Servo	2	REV Brackets
5	Assorted REV Hardware	3	REV Metal Servo Bracket
1	REV Metal Motor Bracket	1	Counter Weight
2	REV Gears	6	Custom Odometry Panels
1	REV Power Switch	3	Through Bore Incoder
1	3D Printed Custom Covered Spool	6	Assorted Ball Bearings
6	8mm Screw Collar	6	REV M3 4mm Shaft Collar
9	goBuilda 16mm Standoffs	3	Axel
24	Socket Button Head Cap Screw	3	Through Bore Incoder Insert
3	Hex Shaft	1	3D Printed Spool

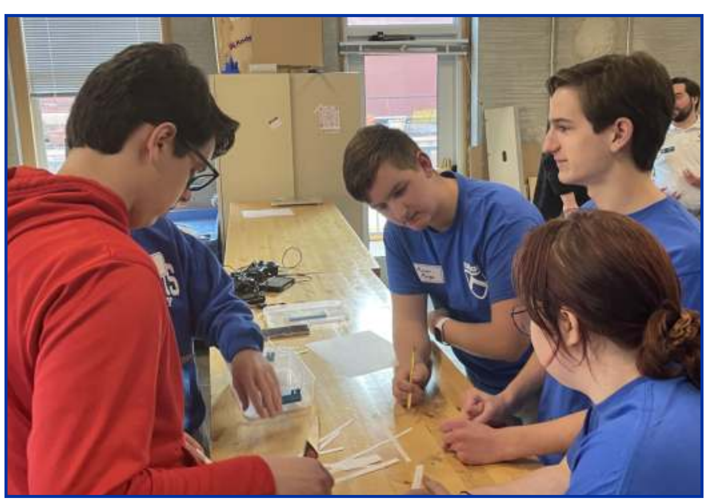
TEAM ORGANIZATION

Mentorship Program

- Once a new member joins our team, they get to choose the subteams that they would like to be on. Once on these subteams, the member will work with upperclassmen that are experienced in that role.
- This experienced mentor will guide our new team members through the process of what that subteam entails.
- Once a subteam has been assigned a task, the new member will work alongside our experienced member, learning the design strategy, terminology, and overall skills.

Communication

- Our team leaders start our meetings using a whiteboard task-tracker to set goals for each subteam to achieve. These tasks are then assigned to a team member and a deadline is set.
- Discord is our main communication platform where we make announcements about future meeting times, subteam announcements, and important tournament information.



Team Sustainability

Recruiting Members

Club Fairs, open houses, Campus Day, two FLL teams, apprenticeship, no cut/ no pay policy.

Retention

Member coaching program, skills development program, free choice for sub-teams, and parent involvement if desired.

Securing Financials

We receive generous support from our school and school community

Building a Legacy

Proficient members are role models for younger students. They build team bonds, demonstrate positive attitudes, and create a demand to join the team.

OUTREACH

Current Events

Summit Middle School Robotics

Our school has an FLL team in the middle school. We make frequent visits to their meetings to help in all aspects of their LEGO robot. We also demonstrate our robot to them so they have a greater interest in robotics when they begin high school.

Mentoring Ursuline Robotics Team

We invited Ursuline Robotics Team #20317 to our space so they can utilize our field, receive advice from our build and programming subteams. As last year was their rookie year, they have gained experience that would normally taken longer to acquire.

Open Houses & Welcome Receptions

We attend multiple admissions events from our school to promote our school's image and gain more members for our club. We have acquired most of our members from these events, and we get a chance to talk to the incoming students and advise them on computer science and engineering interests they have.

Summit Teacher Exposition

We invited the teachers at our school to view our presentation. In addition, they got to see a demonstration of our robot and give us valuable feedback. Additionally, we educated our school community on how our team operates and what we do.

Summit Community Exposition

↓

We invited members of the professional community (including engineering students, software engineers, CEOs, CFOs, surgeons, as well as other professionals) to view our presentation, design, strategy, and to give feedback on things that can be improved. We found this feedback to be vital to the further development of the team.

Name	Profession/Company	Name	Profession/Company
Andrew Crall	Software Engineer at GE Aviation	Connie McDulin	Chemist at Clinipace Worldwide
Ashley Miller	Teacher at Summit Country Day	Dan Arant	Engineer
Brian Bailey	Program Manager at The IRS	Dustin McDulin	CFO of Galls
Chris Chadwell	Physician at Christ Hospital	Jackie Ardrey	CEO of Vera Bradley
Chris Miller	Research Specialist at P&G	Jason Hiller	Software Developer at Fifth Third
Christine Chadwell	Nurse Practitioner at Christ Hospital	Manisha Patel	Heart Surgeon at Mercy Heart
Christy Bailey	MRI Tech. at Christ Hospital	Mike Curran	General Surgeon

ENG NOTEBOOK

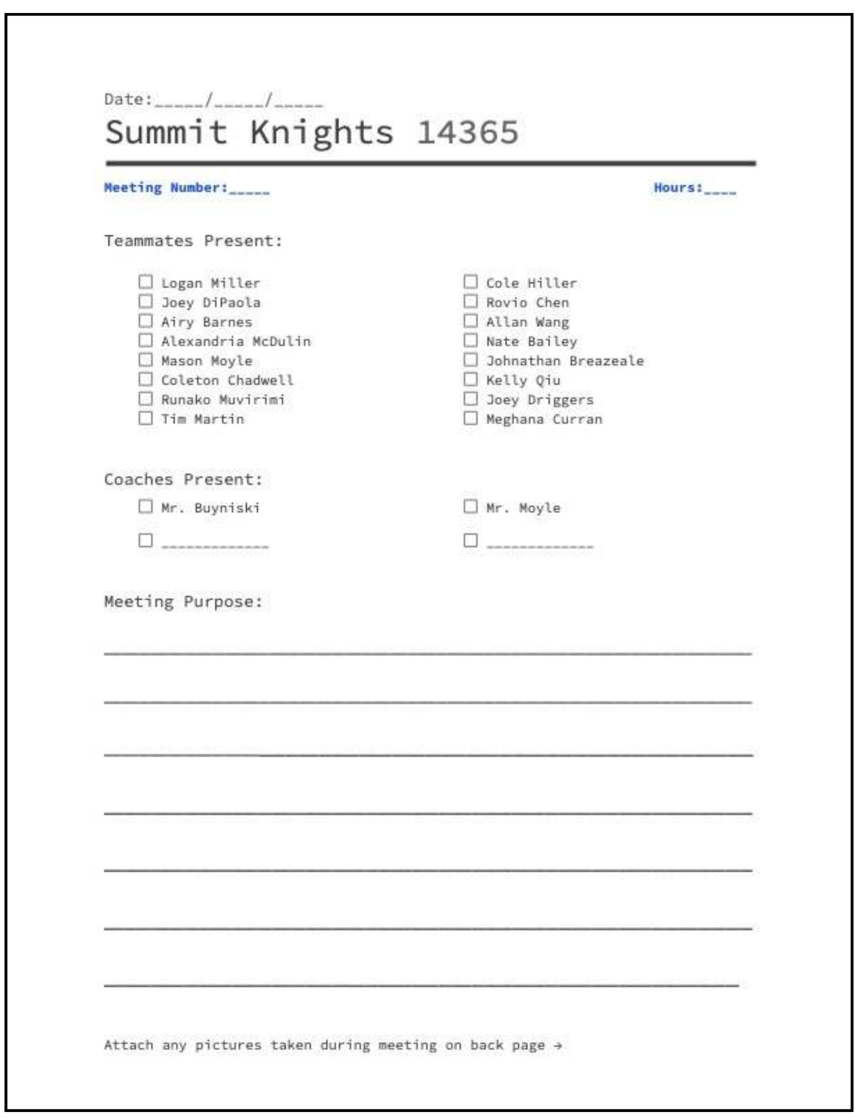
Quality & Accountability - Each subteam is required to fill out one entry per meeting that explains what was accomplished that day. The teammates and coach sign, and then it is approved by the engineering notebook team.

Everyone Participates - Every day we alternate who writes the page, which ensures that everyone gets involved in the engineering process.

General Questions - These help guide the team members to write their entry and helps to formulate their plan of progression moving forward.

Organization - Each page is included in the engineering notebook, which is organized by subteam and meeting number.

Progression of Ideas - All the sheets together show the progression of the team's work from brainstorming, prototyping, to final product.


Date:_____/_____/_____

Summit Knights 14365

Meeting Number:_____ Hours:____

Teammates Present:

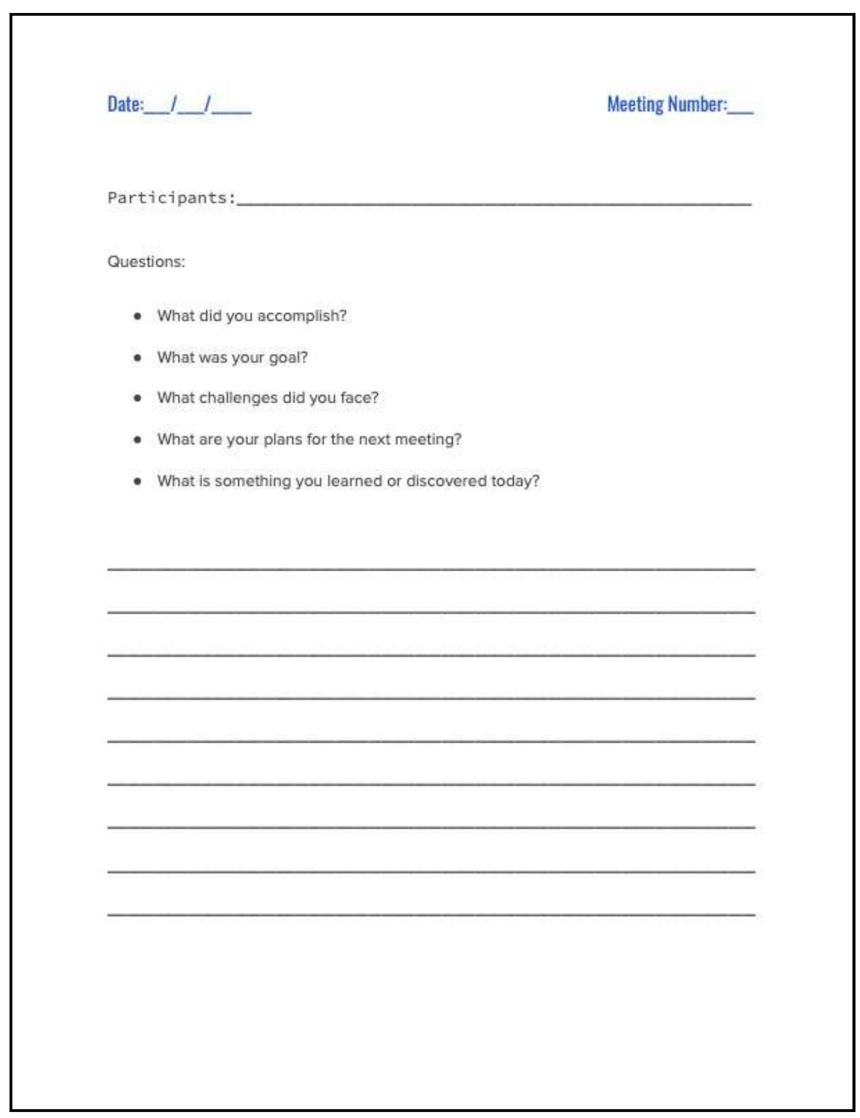
- ☐ Logan Miller
- ☐ Joey DiPaola
- ☐ Airy Barnes
- ☐ Alexandria McDulin
- ☐ Mason Moyle
- ☐ Coleton Chadwell
- ☐ Runako Muvirimi
- ☐ Tim Martin
- ☐ Cole Hiller
- ☐ Rovio Chen
- ☐ Allan Wang
- ☐ Nate Bailey
- ☐ Johnathan Breazeale
- ☐ Kelly Qiu
- ☐ Joey Driggers
- ☐ Meghana Curran

Coaches Present:

- ☐ Mr. Buyniski
- ☐ Mr. Moyle
- ☐ ____________
- ☐ ____________

Meeting Purpose:

Attach any pictures taken during meeting on back page →


Date:___/___/____ Meeting Number:___

Participants:

Questions:

- What did you accomplish?
- What was your goal?
- What challenges did you face?
- What are your plans for the next meeting?
- What is something you learned or discovered today?