Engineering Summary

Walton Robotics

Team 2974

2023 - 2024

Shosty

Table of Contents

Kickoff

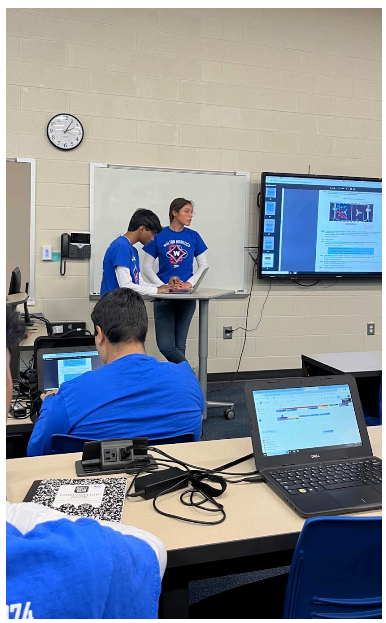
Strategy Session

Our team hosted this year's kickoff in our school's cafeteria and our build site. We invited Peachtree Ridge, Robolions, to join us as well as younger/rookie teams from Lassiter, Wheeler, and a local team the Mechanisms.

We broke the game manual down into different sections and analyzed the information in groups consisting of members from various teams. We invited alumni to come talk to us about how to break down and better understand the game. Mentors were also able to add valuable input throughout the event as we developed our season strategy.

Mechanisms Session

The next day, our team met to discuss ideas we had brainstormed the night before.We split into groups and had our engineering leadership guide groups through the brainstorming process. Each team member was encouraged to contribute their valuable input to the discussions.

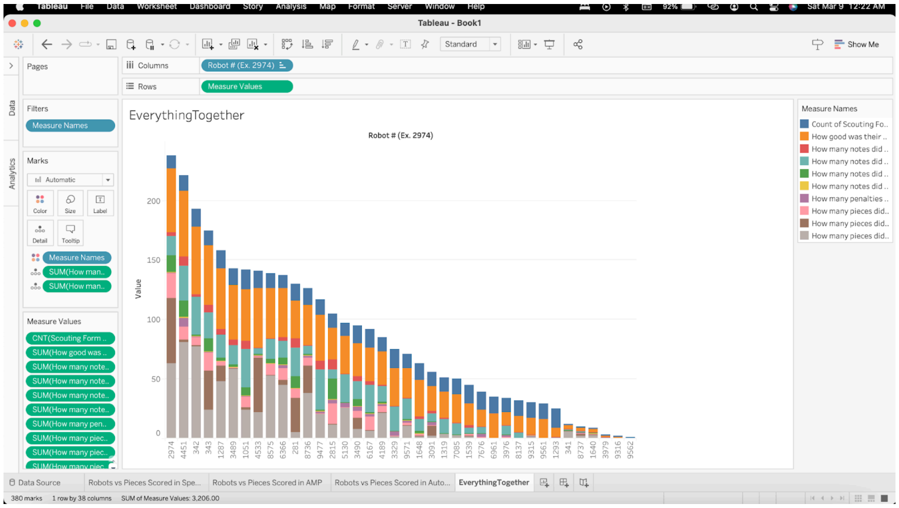


Strategy

Game Analysis and Strategy

Crescendo requires teams to shoot notes into Speakers, while attempting to amplify via the lower scoring option, the Amp. While efficiently completing the secondary challenge of balancing one or more robots on a suspended chain on a truss structure called the stage which can be Spotlit with interaction from the Human Player. Simultaneously, scoring in a trapdoor mechanism called the trap. Our team decided that, in order to achieve our longtime goal of being an alliance captain at worlds, that we must accomplish the following:

- Qualifications
    - Earn all ranking points available by scoring notes in the speaker and amp, climbing onto a spotlit stage, and winning matches.
    - After the points update to 25 notes (21 with coopertition) to earn the scoring ranking point, we have determined that cooperation will be necessary with some partners. Climbing and trapping is also imperative to rank higher through qualification points, especially at championships.
- Alliance Selection
    - The night before we have an extensive meeting, in which we pore through data collected by our 25+ scouters. By utilizing powerful tools like Tableau and our Scouting App, we generate visualizations of metrics we think are important to an alliance partner that would suit us best.
    - This season as teams were picked they were automatically removed from our visualization. After they were removed we had weighted values, including Amp scoring, speaker scoring, and auton flexibility. Based on these weightings teams were ranked, this automatic data driven ranking clearly presents us with a pick that most optimally complements us during elimination, just based on data.



- Playoffs
    - Utilizing scouting data from prior matches and analytical research, we are able to fabricate the perfect match strategy unique to the alliance we go up against to commit to achieving the win.
    - Our most important objective in playoffs is in autonomous; clearing the field puts up 61 points, and synergizing autons with alliance partners allows for a huge head start early in an elimination match.

After establishing ourselves as one of the top scoring teams in the world, we have looked to further refine our game strategy. By looking into options such as other robots ferrying notes over to us, as well as passing to other teams. At our district championship competition while being fed we averaged the most notes we ever had before, this success led us to switching strategies, to a more collect passes and shoot orientation.

Robot and Game Goals

Requirements
Autonomous/ Tele-op
Robot must be able to move
Robot must intake Notes from ground (touch it, own it)
Including midline notes during autonomous
Including scoring zone notes during autonomous
Robot must be able to Score Speaker (within 1/2 second)
Robot must be able to Score Amp (within 1 second)
Robot must have Accurate pathing to game pieces during autonomous
Robot must intake Notes from Source (within 1 second)
Robot must have automated fine tuning position for scoring in speaker
Robot must have automated scoring for scoring in speaker
Robot must have automated fine tuning position for scoring in amp
Robot must have automated scoring for scoring in amp
Robot must have automated pathing during autonomous
Robot must be able to accomplish 12 cycles in one match once autonomous is done(two more cycles in auton)
Fit underneath Stage 2 ft. 3⅞ in
Endgame
Robot must be able to completely support another robot during Onstage (within 2 seconds)
Robot must be able to Onstage (within 10 seconds)
Robot must be able to place note in Trap (within 3 seconds)
Robot must have automating scoring for scoring in trap

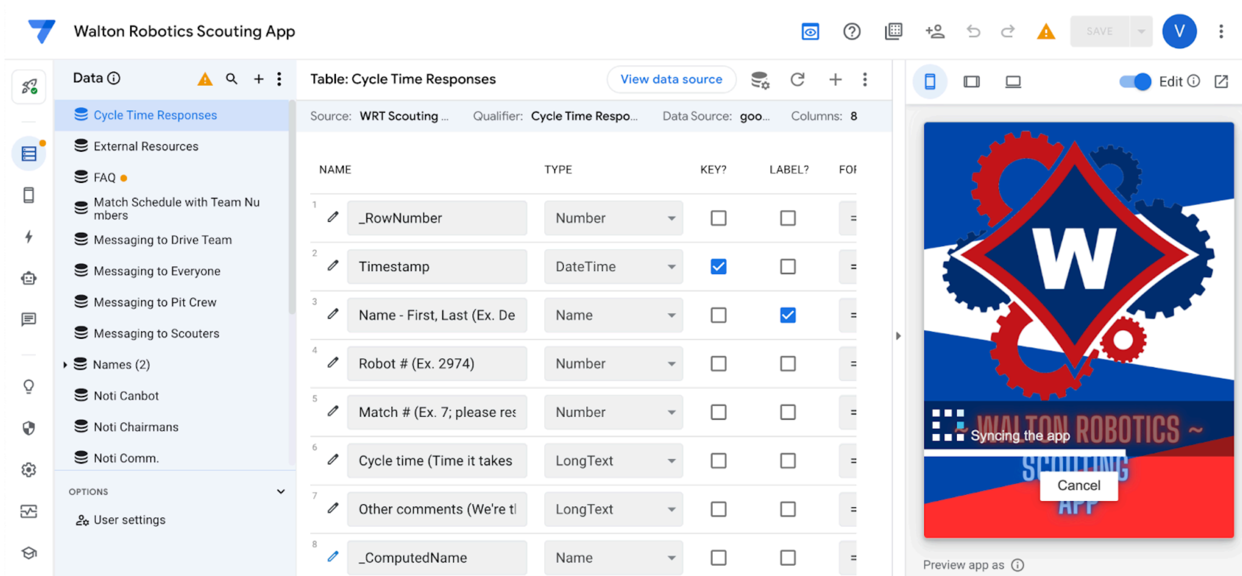
Red = High priority
Blue = Medium Priority
White = Low Priority

Walton Robotics Scouting/Competition Management App

In prior years we have struggled with getting accurate scouting data, we have noticed that other teams suffer from the same problem. Keeping all of this in mind, we wanted to design a solution that solved this problem and went the extra mile while doing it. By giving our JV programmers an opportunity to build an app that addressed our needs; we were able to accomplish much more than originally planned out.

- The App
    - No internet connection needed to scout
    - Sends SMS notifications to scouters for matches, and Pit Crew for queue
    - Consolidates multiple schedules into one compact setting
        - Canbot, Pit Crew, Pit Presentation, Strategy, Scouting
    - Statbotics integration for data accuracy comparison
    - TBA integration for built in match schedule
    - Direct connection to Tableau data analysis and integration for statistics.

- We were also able to open source it to all of FIRST through the portfolio link posted on our build blog, so any team is able to use and improve on the software.
- The app has been spread to over 80 teams, with 4 teams utilizing the app for all their scouting needs at World Championships. We plan on spreading the message for our scouting app even more at worlds.



Subsystems Overview

Shooting Mechanism

- Spin shooter design
- Instant piece exit
- Certainty with position (multiple sensors to determine angle of shooter and position of piece)

Intaking Mechanism

- Touch it, own it
- High driver tolerance (Driver has large area of acceptable contact to grab piece)

Conveyor

- Instant transfer (Intake to Shooter is instant)
- High torque helps when intaking uncentered notes
- Runs at the same time as the intake rollers

Climbing

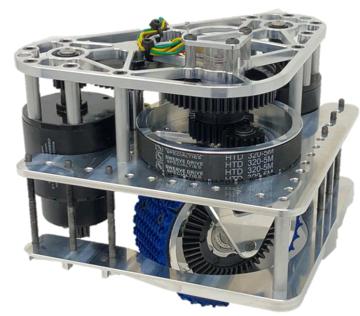
- Winch System
- Hooks attached to winch and bent fiberglass rods

Camera and Global Positioning

- Pigeon, camera camera camera
- AprilTags are love, AprilTags are life

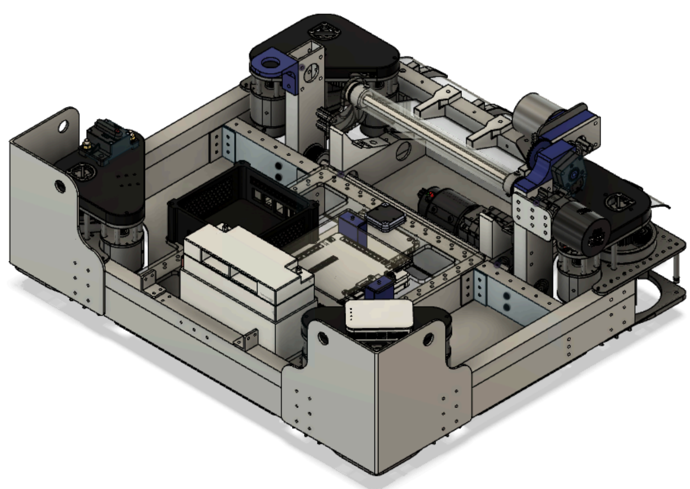
Drivetrain

MK4Is Modules



- 3” spacing between plates
- L3 drive ratio
- Kraken drive motor, Falcon 500 steer motor

Chassis

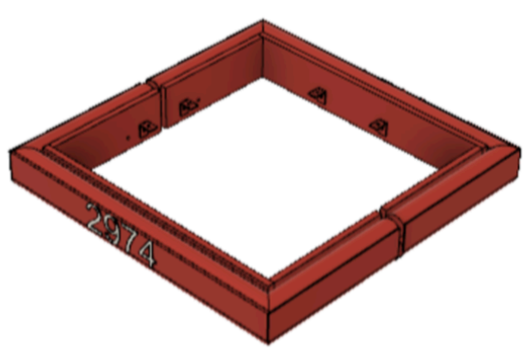


- 28.5” x 28.5” frame perimeter
- 1/8” aluminum belly plate without pockets

Electrical

- PDH on underside of main electrical panel elevated over belly plate to fit within compact chassis structure
- Battery in back of robot to offset CoG of pivoting shooter
- 2 CANbus serials, one for CTRE and one for REV products

Bumpers



- 2 piece bumper
- 2 ¼ above the ground

Intake

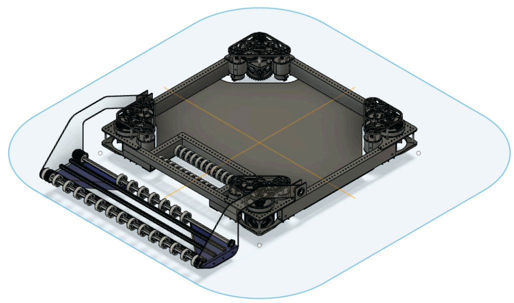
Brainstorming

Introduction:

Before brainstorming, we discussed the functions the intaking mechanism needed to accomplish. To maximize cycle times and efficiency, our intaking mechanism had to be "touch it, own it," we also knew the design must allow for some variance in picking up notes.

Results of Brainstorming:

Our team took the ideas from kickoff and began 3D modeling them to get a better understanding of what each design would look like. On the right are two designs that students came up with.



Decision Matrix:

The team then shared their designs and later consolidated them to reach the list seen below. To narrow down the possibilities for prototyping, we put the ideas in a decision matrix and compared them based on predetermined criteria. Based on their final weightings, we decided to explore further the stacked roller intake, integrated intake shooter, and under the bumper intake.

Intake	Speed	Consistency	Robustivity	Familarity	Build Simplicity	Control Ability	Driver Ease	Integration with rest of robot	Final Weight
Weight	5.00	5.00	4.00	2.00	3.50	2.75	2.50	2.50	
Stacked Roller Intake	7	7	3	1	3	7	7	3	138.75
Expanding Claw	1	1	1	3	7	7	3	1	73.75
Grasping Claw	1	1	1	7	7	7	3	1	81.75
Under the Bumper	7	7	7	1	1	7	3	1	132.75
Integrated Intake Shooter	3	7	3	1	3	7	7	1	113.75
Conveyor Belt Intake	1	1	1	1	1	1	1	7	42.25
2-sided intake	1	1	1	1	1	1	1	1	27.25

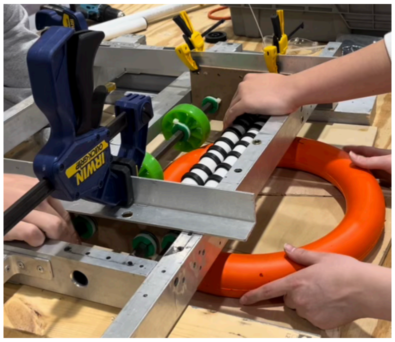
Prototyping and Testing

Our design process started in Fusion 360, where we sketched an idea's geometry and started designing basic intakes. We made sure that the shape of each design was compatible with the pathing of the note.

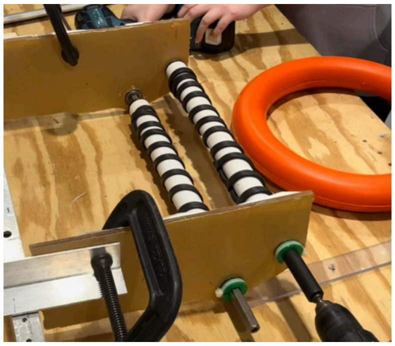
With each design we realized that note pathing and note bend radius would be important to consider with each design.

- **Over the bumper**
    - One design tested- stacked wheel rollers
    - Bottom rollers were good for grabbing and owning
    - Not as good at conveying note between the bars into the chassis
    - Time lag to flip outside of bumper

- **Under the bumper**
    - Under the chassis
        - Would also need an over the bumper design to transfer the note to roller wheels under the bumper
        - More a form of conveyor, directly under/into shooter
        - Angle of intake to shooter would be too sharp
        - Chassis shape would restrict max size of opening

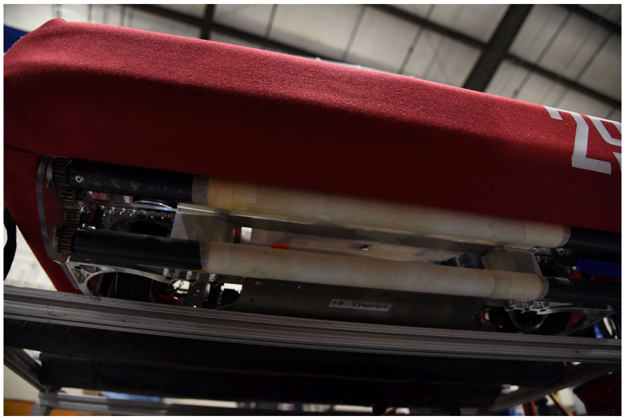


    - Behind the bumper beater bar- Winner
        - Most efficient intake
        - Ideal angle up into robot body
        - Tested with .2” compression on note diameter, good grip
        - Allows for full frame width span
        - Frame perimeter will not be the same as drive train



Final Design

Our final design is a behind the bumpers dual beater bar. The design features carbon fiber rods powered by a kraken and has two 3D printed side pieces to funnel the note through the intake and into the conveyor.



Components:

- 2 Carbon fiber rods
- Grippy tube rubber
- Slick Tape
- 1 Kraken
- 18T pulleys
- Polycarbonate plates

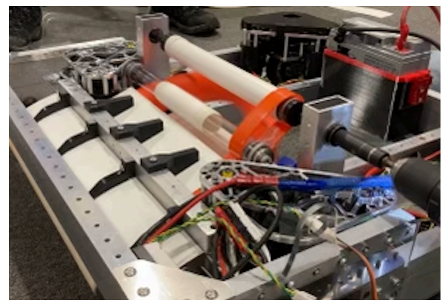
Conveyor

Introduction:

With designing the conveyor we decided to make a decision that we would finish designing the intake and shooter separately and then design the conveyor to accommodate the geometry of the separate designs.

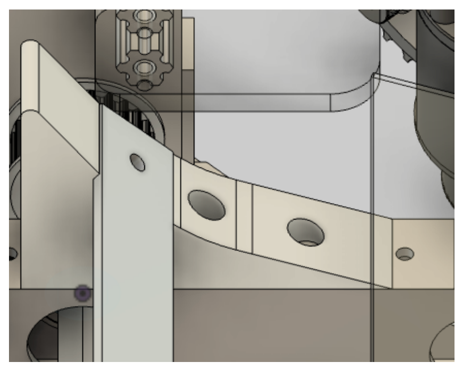
Initial Design:

- 3D printed curved guide up to shooter
- Top gripping roller



Iteration #1:

- Top roller powered from belt to bottom shooter rollers
- 1/16” HDPE under and over guides
- Bottom idle roller with compliance wheels for grip and small compression

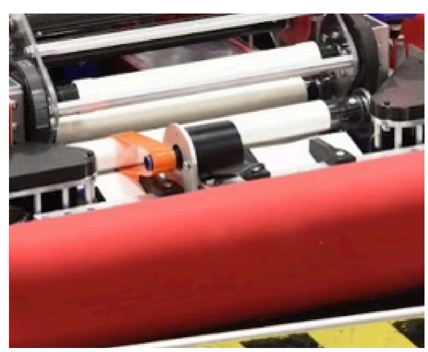


Iteration #2:

- Wheeled idle roller was slowing down note, removed and extended bottom plate
- Bar added to prevent HDPE bowing
- Steeper curve for HDPE bottom supports, once shooter was attached and we could see geometry better

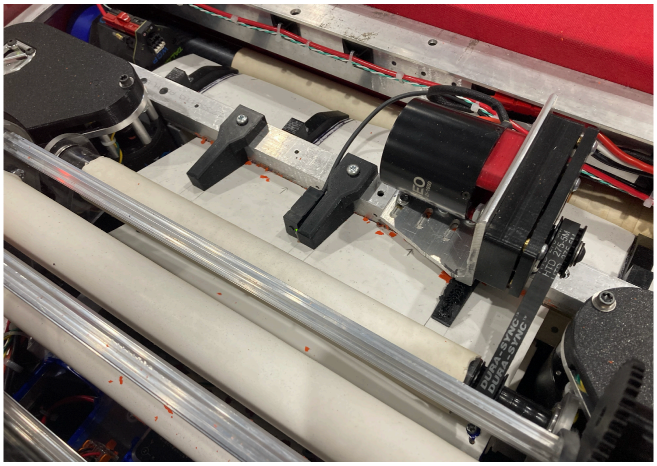
Iteration #3:

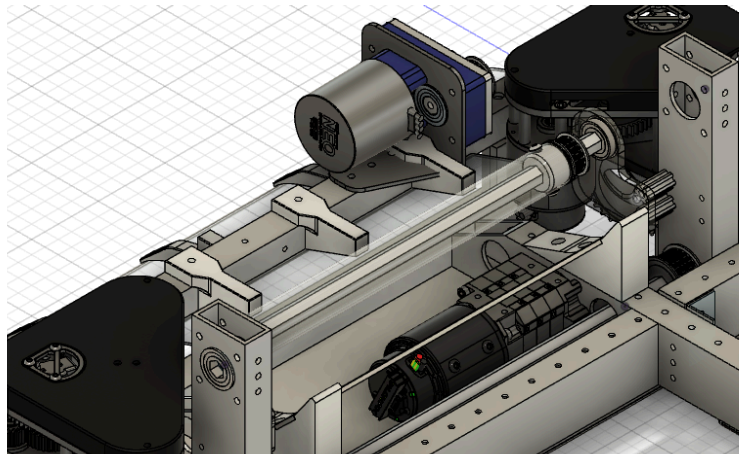
- Motor mounted above intake to power roller
- Polybelt between motor and roller with crowned pulley



Final Design

Our final design consists of bottom and top guide plates and a powered top gripping roller the same width as our shooter. We switched from the polybelt in our previous iteration to pulleys with a timing belt because the motor was turning so rapidly that the polybelt was melting and snapping.





Components:

- Polycarb roller with through axle
- Polycarb roller brackets with 1” spacers
- Neo motor
- 2:1 ratio custom gearbox
- 18T pulleys, 55T belt
- 1/16” HDPE

Shooter

Brainstorming

Before we brainstormed concepts, we decided that as the money maker of the robot, the shooter needed to be compatible with various intake concepts while still retaining its functionality as an accurate, fast shooter. Based on our strategy requirements, coupled with our shooter requirements, we brainstormed multiple concepts that we knew would work for quickly shooting notes with high power and accuracy.

Decision Matrix:

Like with the intake, we brainstormed many ideas with the whole team, then condensed the realistic ideas, and compared them with our predetermined criteria for the shooting mechanism.

Intake	Speed	Consistency	Robustivity	Familarity	Build Simplicity	Control Ability	Range	Integration with rest of robot	Final Weight
Weight	5.00	5.00	4.00	2.00	3.50	2.75	4.00	2.50	
Integrated Intake Shooter	3	3	1	1	3	3	1	7	76.25
Side roller shooter	7	7	7	7	7	7	7	3	191.25
Top/Bottom roller shooter	7	3	7	3	3	7	7	3	149.25
Frisbee shooter	7	3	7	3	7	7	7	1	158.25
Turret	1	1	1	1	1	1	1	1	28.75
Adjustable angle	1	1	1	1	1	1	1	1	28.75

Other Factors We Considered:

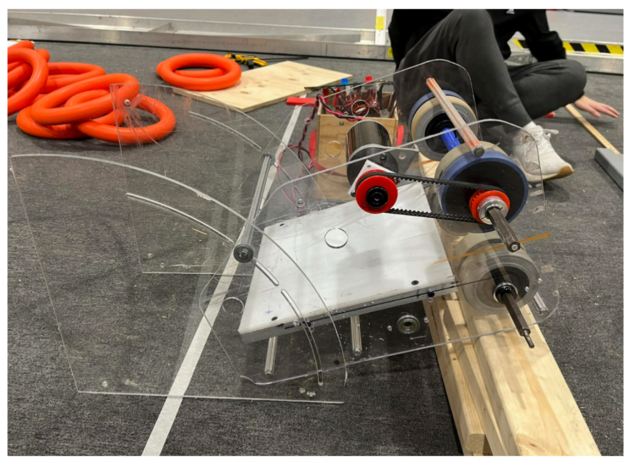
- The shooter mechanism and intaking mechanisms could work as one mechanism
- Shooter needs to shoot EVERY intaken note consistently
- The shooter pivot mechanism had to support the weight of the whole shooter while pivoting on an axis that allowed for a smooth note path
- Shooter pivot had to move instantly with minimal backlash

Our Final Choice: A top and bottom flywheel shooter.

Prototyping and Testing

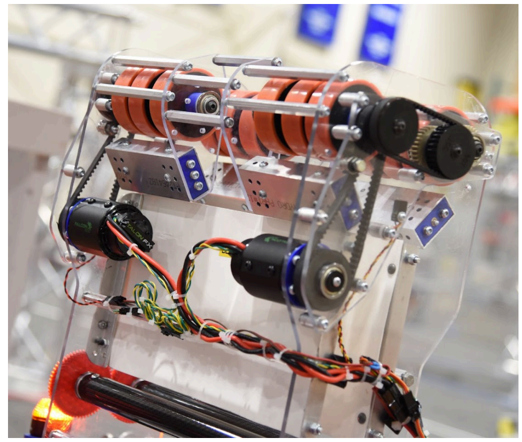
- Side Roller Shooter
    - The side roller shooter used 4 sets of 3 mini flywheels to shoot out the notes.
    - It also allowed us to test different note compressions while keeping a constant vertical compression.
    - Though it looked like a promising initial design, after performing kinematic equations for the maximum distance a note exiting could travel and by testing the shooter, we determined that it would not be sufficient enough to shoot from the distances we wanted to shoot from.

- Top and Bottom Shooter
    - The top and bottom shooter was more promising than the side roller shooter as in this design, the notes went much further.
    - This design used two motor powered axles on the top and bottom as well as a polycarb glide plate to slide the note into the wheels.
    - However this design used colson wheels and was very heavy, leading to concerns over pivoting and movement.
    - This design allowed us to conduct tests to get data for ideal shooting angles, as well as giving us a strong foundation to design our final shooter. With the use of a custom belt, and a strong spinning design our future shooter drew heavy inspiration from this design.



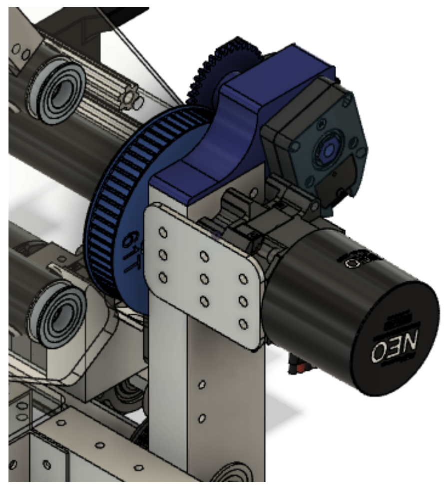
Final Design Specs

Shooter



- Top and Bottom Dual Flywheel assembly
    - Height: 20.25”
    - Width: 16.50”
- 2 Falcon 500 motors power flywheels
    - 2:1 pulley upduction for flywheels
- 85T belts transfers power to top flywheels while 30T belt and 32T gears transfer that power to bottom flywheels

Shooter conveyor rollers

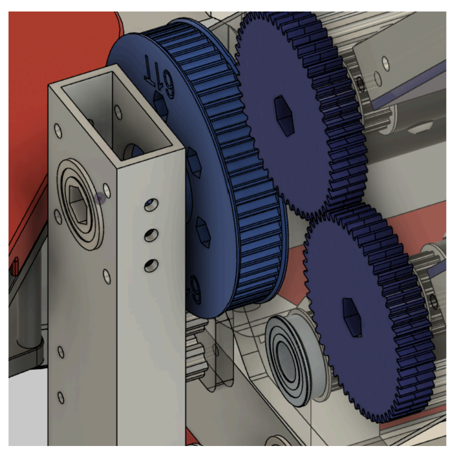


- 4 carbon fiber rollers with 18T 9mm pulleys connected by 60T belts
- NEO with 3:1 ratio spins these rollers

Encoder assembly

- 61T gear behind shooter 61T pulley meshes with 36T pulley that has encoder shaft
- Mounts inside and to the side of shooter tower

Shooter Pivot



- 61T pulley mounted to shooter side plate
    - 100T belt connects 61T pulley to 24T pulley below; LBracket tensions belt
    - Falcon 500 with 125:1 reduction spins 24T pulley
- Backlash dampening System
    - 1” 3D printed axle attachment rubs against old tread
    - Mitigates the large amounts of backlash in the system

Climber and Trap

Because of the complexity of this year's game mechanics, we decided to prioritize our note scoring design and leave space in the robot for a climber until the rest of the robot was complete.

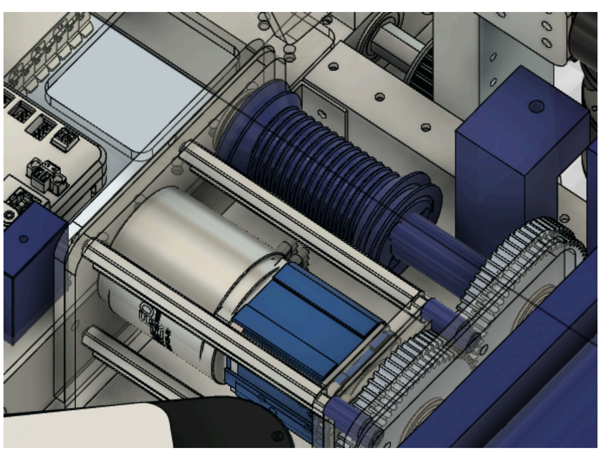
Design Requirements:

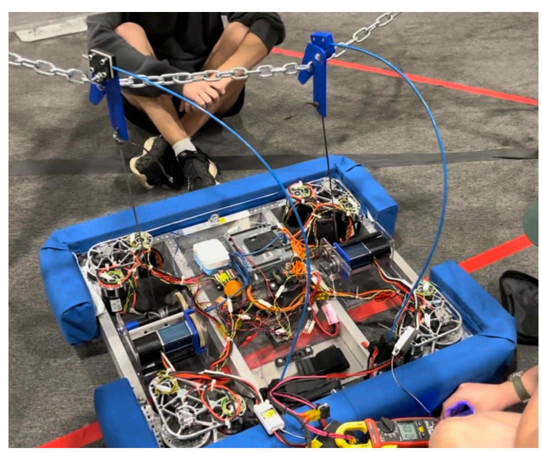
- Accessibility to parts: The design should be made of parts we already have or can easily manufacture.
- Strength: The design should be able to lift the robot and stay lifted so the trap can be scored easily into.
- Durability: The design should be robust and not be a wear point for the robot.
- Attachability: The design should be easily attachable and removable from the robot.
- Trap: The design should allow the robot to climb as high as possible to make the robot closer to the trap

Based on these requirements and different climber designs we saw at the beginning of the season, we decided to further explore a winch system.

Initial design

We decided on using a two winch system system with carbon fiber rods that flip out to be able to attach to the chain then the two winches are used to pull it down.





This was the design we had before going into Anderson. However, we did not end up attaching it until DCMP as we were still in the testing phase until that point.

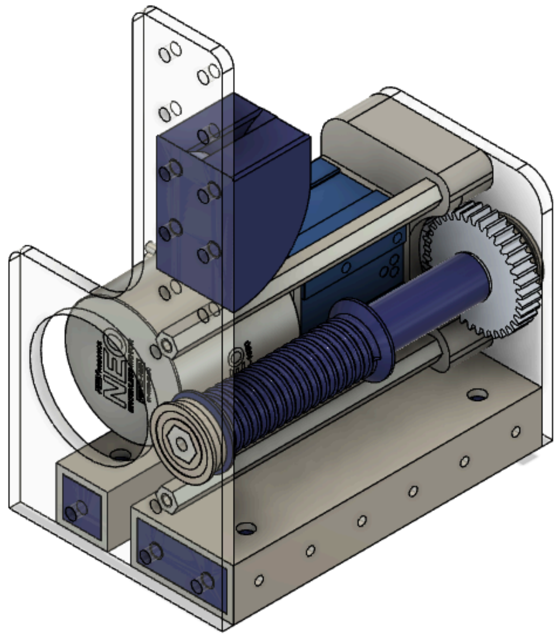
The biggest change of design from beginning to end is that we clocked the winch system 90 degrees and made it more compact. This allowed us to more easily fit and attach the design into the robot.

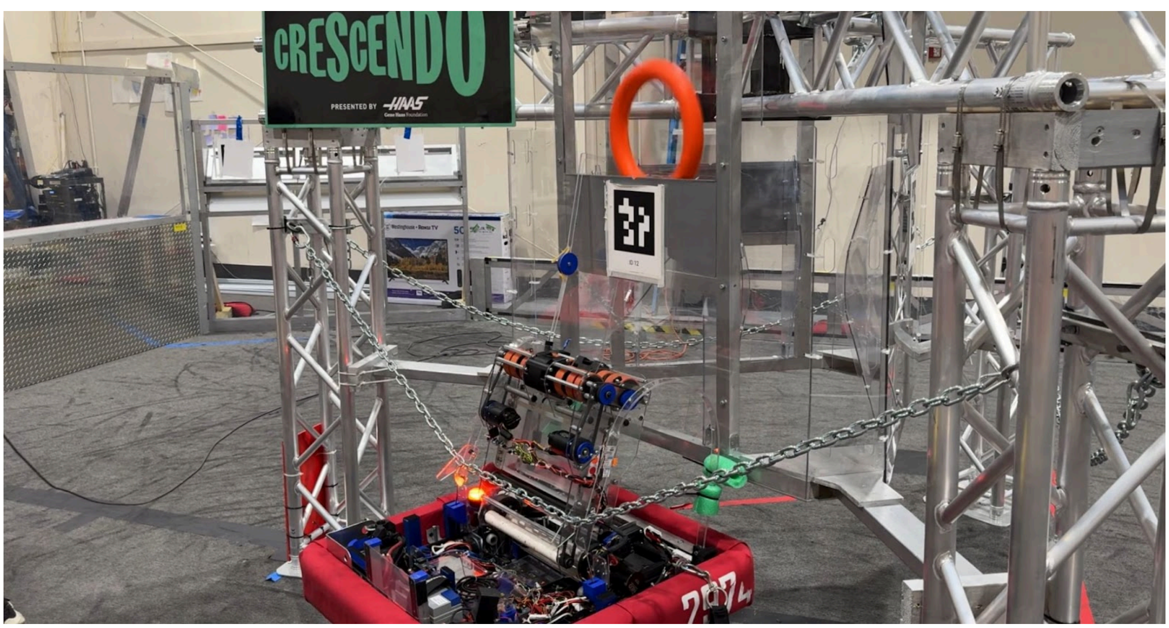
The 2x1 and 1x1 aluminum tubes also have 3D printed inserts that contain places to embed nuts. This allows us to more easily attach and remove the design.

Final Design

Components:

- Winches
- Hooks
- ¼ Polycarbonate rods
- 1x1 aluminum tube
- 2x1 aluminum tube
- 3D printed spool
- 36:1 motor gearbox
- Neo
- Motor attachment





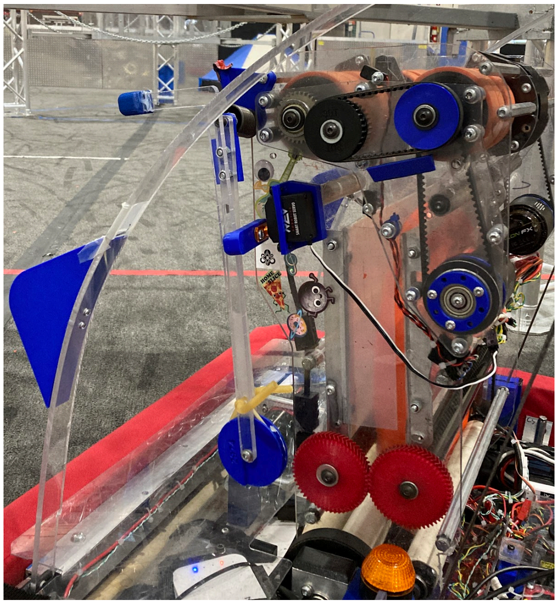
Trap

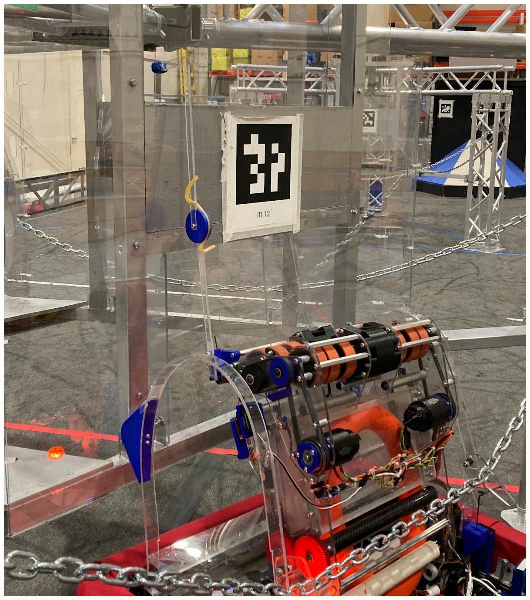
For the trap we initially tried shooting into it at different speeds and angles. We could not get this approach to work because the note would stick to the polycarbonate trap door. We realized because of this that we would need a separate mechanism to open the door when climbing so we could shoot into it.

Design Requirements:

- Must be lightweight and fit on the shooter
- Must extend from the top of the shooter to the bottom of the trapdoor
- Must not get in the way of other robot mechanisms

The solution we came up with was a double actuating arm. The first arm pivots using an axial spring connected to the shooter itself, and the second arm pivots using an elastic band around a pulley.





Programming

Intake photoelectric sensor

- Detects that a note has entered the intake and triggers an automatic intake and conveying system to ensure that the note is in the correct position to shoot even if the driver releases the button early

Conveyor beam break sensor

- Detects when the note has reached the top of the conveyor to automatically stop the intake process and retract the note slightly to keep ensure it is not prematurely contacting the shooter flywheels

Shooter beam break sensor

- Detects when the note has reached the top of the shooter to detect when the note has left the shooter to manage the state machine

Shooter pivot absolute encoder

- Absolute encoder is directly coupled to the angle of the shooter so that position can be automatically controlled while ignoring backlash in the planetary gearbox

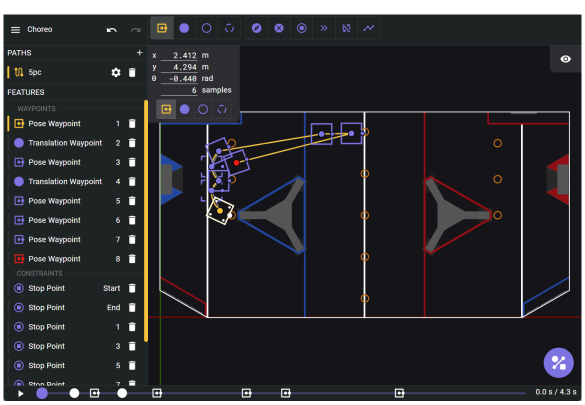
Basic Commands

We started out programming swerve on our test chassis.

Our basic teleop drive command was generated by Phoenix Tuner X.

After generating this code, we used PathPlanner's path follower to follow our Choreo trajectories. For easier testing, we added a method to hold a button and drive to the auton starting position.

Autonomous Code



We created our potential auton routines on Choreo, which we ran on our test chassis before getting the robot. We run our autons using full wheel odometry.

We have several autonomous routines, starting from the center, the amp side, and the source side.

Teleop Code

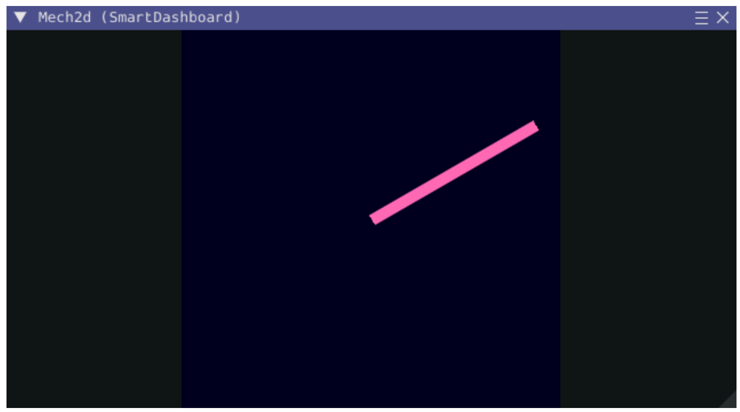
We determined our teleop functionality partially by the robot's design and partially by what the drivers asked of us. We have a button for intaking and a button for shooting, which start a line of states so that the robot knows where the note is at all times and won't make any errors like turning on the shooter wheels before the note has retracted, ejecting the note before the shooter has spun up.

We have the controllers rumble to indicate where the note is in the robot, rather than using LEDs, not only because they are less costly, but also because we have found that it's easier for the drivers to register than having to watch LEDs on the robot.

Aim Code

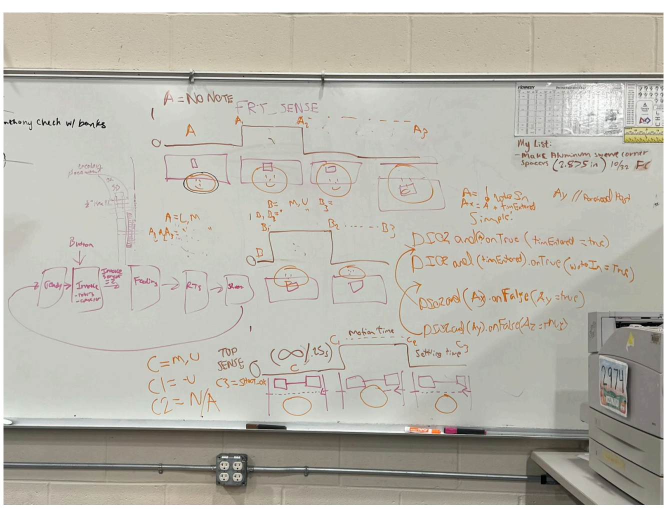
We started out simulating our robot's aim as if it were an arm mechanism using WPILib's arm simulation. In simulation, we tested automatic and manual aiming.

The physical aim has predetermined setpoints that the drivers can shoot from.



Intake → Shooter Code

Once the intake button is pressed, the intake runs until the upper beam break detects a note, and then it stops and our conveyor retracts to prepare the note for shooting.



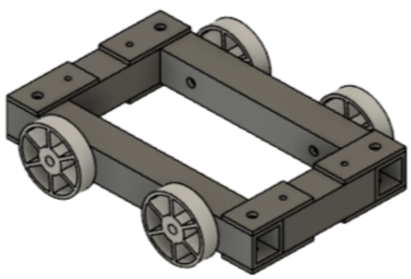
There is another button press for shooting, which will start the spin up only if the beam break detects that the note is no longer in the way of the shooter.

Once the shooter is spun up and aimed, the conveyor will run and shoot the note out.

This whole process is done using a state machine, with several triggers based on the current state of the robot.

Training

Tool Training

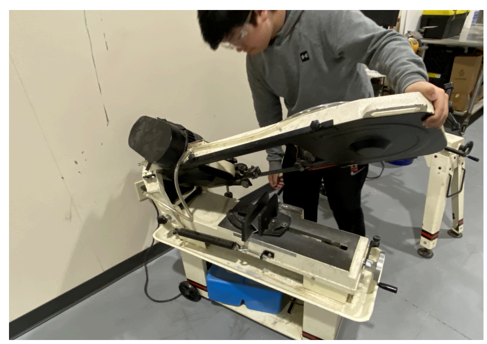


The first experience our members had with manufacturing was a project consisting of reading drawings, cutting parts, and assembling. Using Fusion 360, we designed a basic car out of 1x1 aluminum box stock and 1” aluminum flat stock.

The car was specifically designed to teach students how to use the tools and machines we commonly utilize when building our competition robots. We had our entire team, including our presentation and programming departments, participate in this training because we believe it is essential that everyone appreciates the hard work and dedication of the other departments.

With the guidance of veteran students and mentors, team members learned to use tools and machines such as:

- Combo squares and scratch-alls
- Hammers and punches
- Files
- Clamps and vices
- Wrenches
- Horizontal bandsaws
- Vertical bandsaws
- Belt sanders
- Drill presses and Mill
- Lathe
- Rivet guns

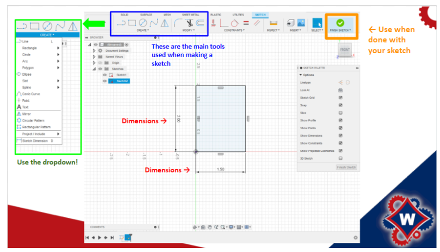


Advanced Tool Training

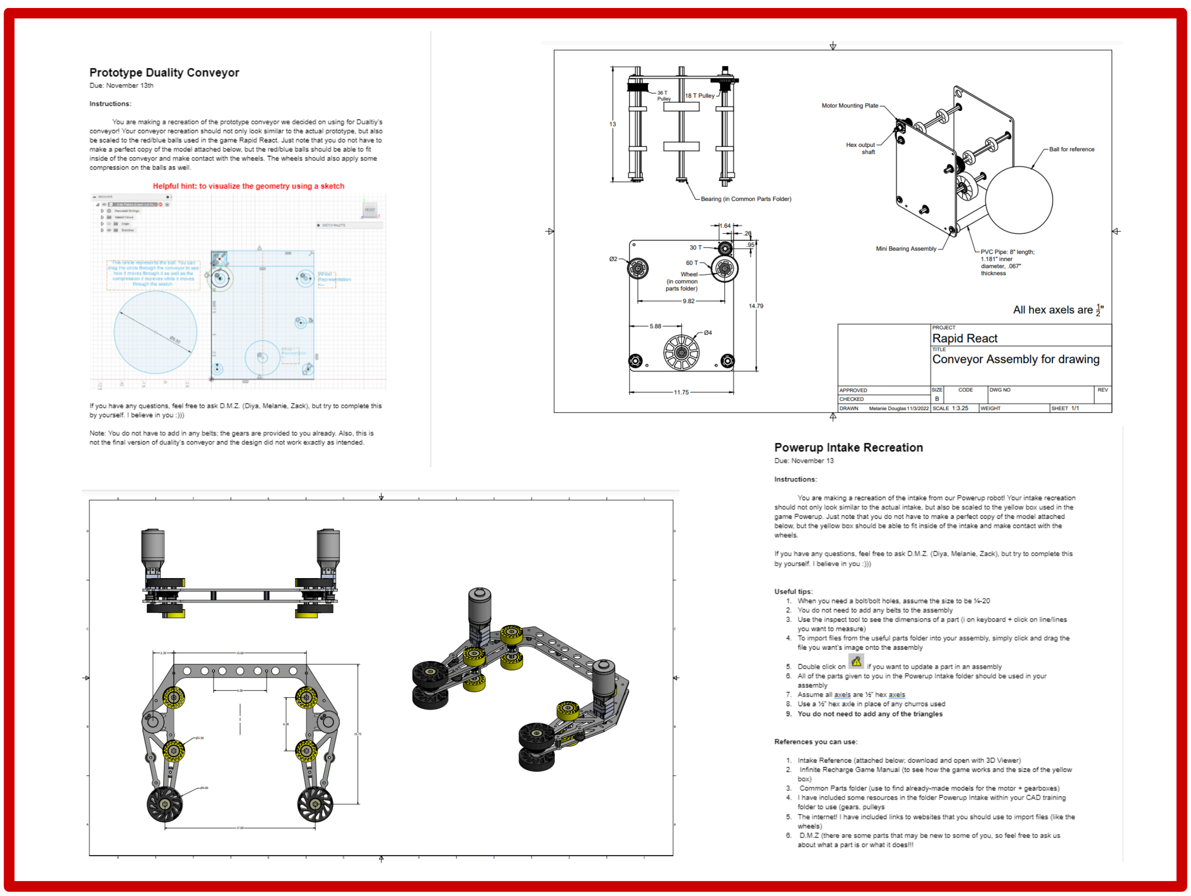
Aside from the tool training that was given to the entire team, advanced tool training was given to those who were part of our varsity build team. We taught important skills such as how to use the mill, lathe, jigsaws, heat guns, etc.

CAD Training

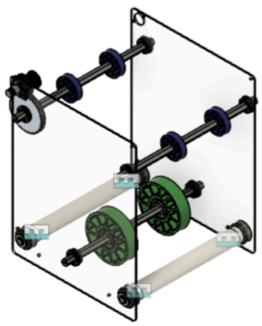
During the fall semester, our other focus is teaching students how CAD can be used to optimize the design process. We started with presentations that taught students basic 3D design functions such as sketching and extruding.

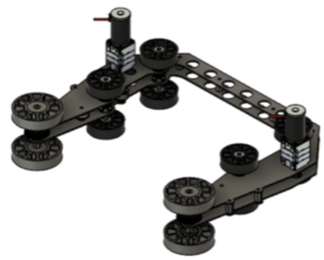


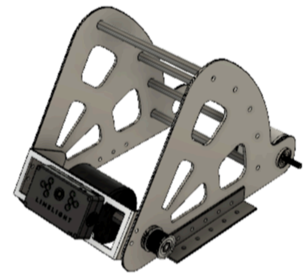
Their next project was to reverse engineer a mechanism from an old robot. We offered three different levels of difficulty and created documents to guide them through the process.



Below are a few examples of student designs based on the instructions provided during our CAD trainings:

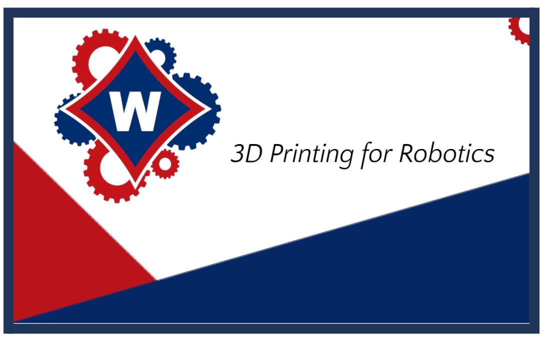


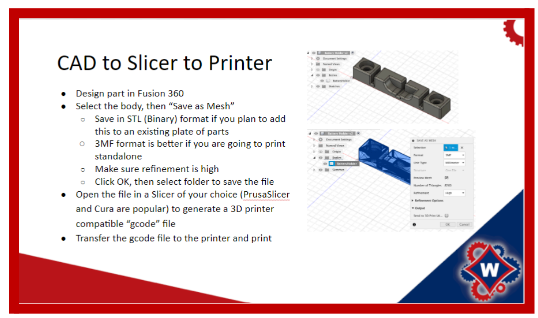




3D Printing

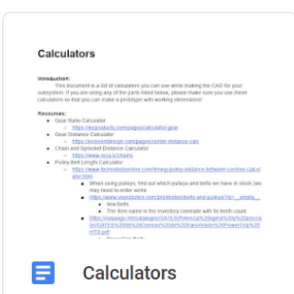
We taught students how to convert parts in Fusion into STL files and then use a Slicer to 3D print different parts.

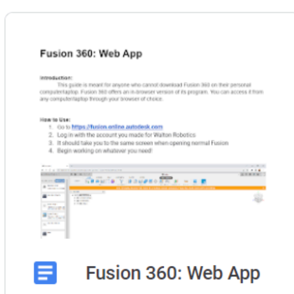


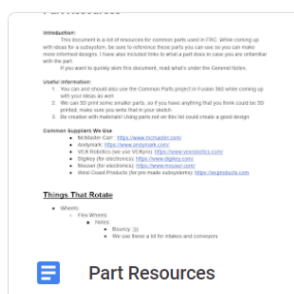


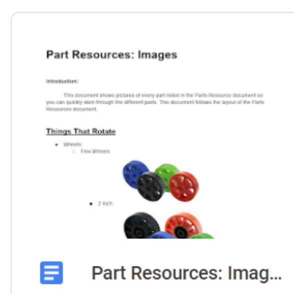
Resources

Our department's leadership also created documents outlining resources to help students while they design and build.









Safety Training

Safety is a very important part of our team's culture. We held two mandatory safety training sessions at the beginning of the year. We took members on a tour around the build site and showed them how to act during an emergency, where our first aid kit and ear/eye protection are located, as well as how to behave around the tools and field in our build site. We then had every single member of the team take a mandatory safety quiz. If they got anything less than a perfect score on the quiz, we gave them the same build site tour as before and had them retake the quiz until they scored a 100%.

Programming Training

Team-wide Training

To get our team (especially rookies) exposed to programming, we held a two hour, mandatory team-wide programming training that covered all the basics.

- It helped get many rookies excited about programming, and we achieved a record amount of programmers on the team this year!
- We used Lego Prime Spikes to simulate a mini-drivetrain and introduce members to basic programming principles, coding in Java, and interacting with robots.

Beginner (JV) Training

After we established who wanted to continue programming on the team, we split up the trainees into beginners and advanced so they could learn at an appropriate pace.

We started with basic Java, with a minor focus on robot programming.

- We instructed them on how to install WPILib VScode, and had them complete the training using this IDE.
- We went through basic Java slides, so they would have a proper foundation in Java by next year.
- We also talked about GitHub, since we use it to collaborate with our fellow programmers.

We wanted them to know about robots and their different parts, so we had them complete a quiz about different types of motors, motor controllers, and other components.They were encouraged to use their resources, such as WPILib.

Advanced (Varsity) Training

For the people who already had Java experience, we jumped right into robot-specific training.

- We taught the advanced programmers about the different hardware/software components, command-based programming, and explored the latest technologies such as PathPlanner and PhotonVision.
- We also used the task of updating previous year's robots as a training tool to help our varsity programmers get accustomed to our team's programming style.

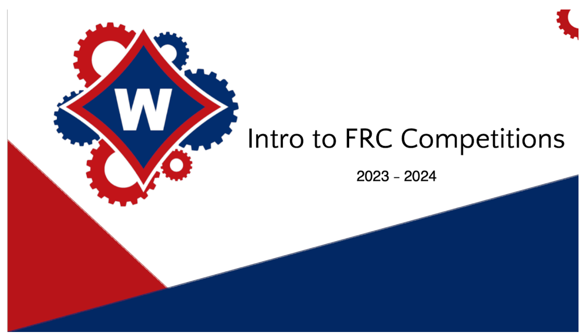
Strategy/Scouting Training

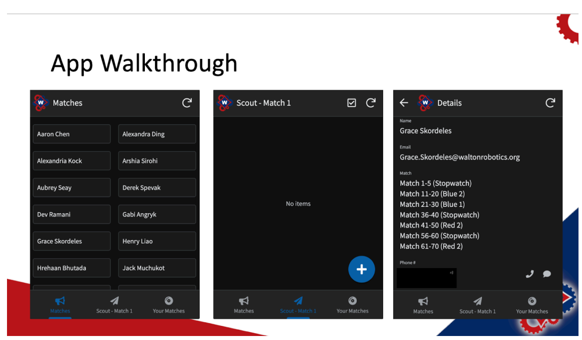
Team-wide Training

To get our whole team involved with strategy and learn about how deep FRC games really are we had a strategy training paired with scouting training with the whole team.

- It helped many rookies learn about past and present FRC games while also seeing that scouting has a profound impact on our robots performance in any given year.
- We used Google Sheets, with an emphasis on data analysis, as well as Google Forms with a focus on data entry and scouting in order to prepare for competition season.

- We also had training to have our team walkthrough our Scouting/Competition Management app so that everyone knew how to use the app to a high standard by our first competition.



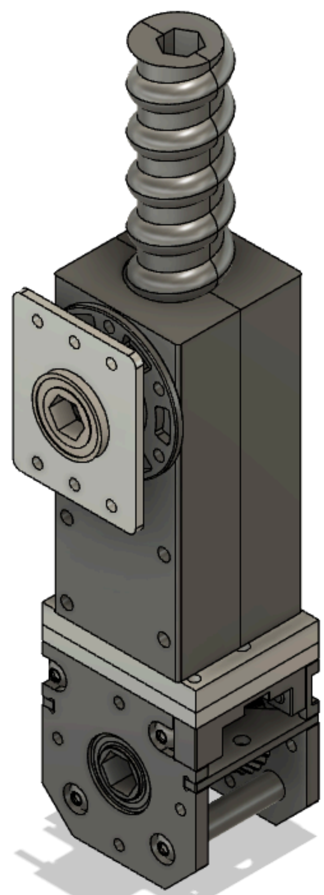


Off-Season Projects: Lead Screw and Wrist

Introduction:

To begin our offseason, we discussed upgrades we wanted to make to our robot for our offseason competitions. We talked about different ways to make our robot's functionality more efficient, and we decided that we wanted to make our elevator and wrist pivots more efficient.

Lead Screw



Problem:

Our robot last year utilized a pulley and spring system that would raise and lower the tilt of the elevator. The strings could easily get tangled up and were hard to tension.

Solution:

We had brainstormed some other possible solutions that had the same functionality as the current pulley and spring system and determined that the lead screw would be the best option.

Instead of using a traditional lead screw, we decided to go with a 3D printed leadscrew so we could customize it to our needs.

To make it work with the current pivoting system, we used a 90 degree gearbox to change the direction of rotation. Connected to the 90 degree gearbox, two plates sandwich a bearing and e-clip to hold in the hex axle.

The screw portion is permanently attached to the hex shaft. The nut portion of the lead screw is attached to the side of the elevator. It is attached through a custom bracket and versaplanetary hub.

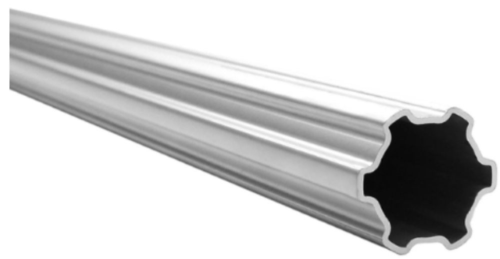
Wrist

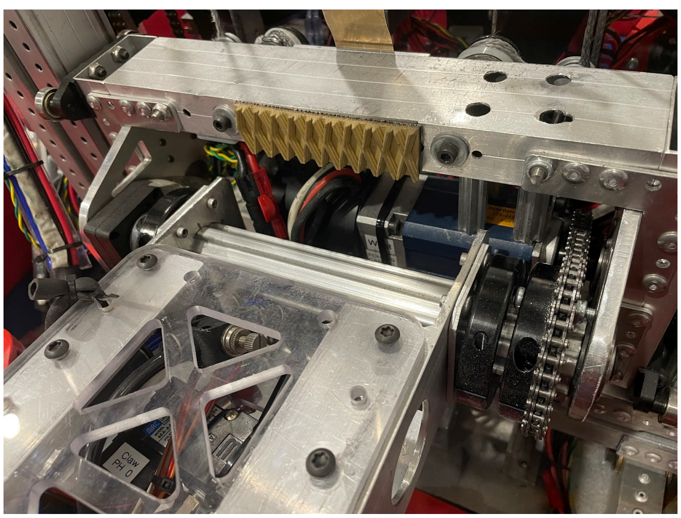
Problem:

With our wrist on our 2023 robot, we found that there was quite a bit of slop between the Versa Planetary Hub and the hex shaft it was mounted to. This was quite bad as this messed up how the encoder read the angle, meaning the wrist was inconsistent with the angle it was going to.

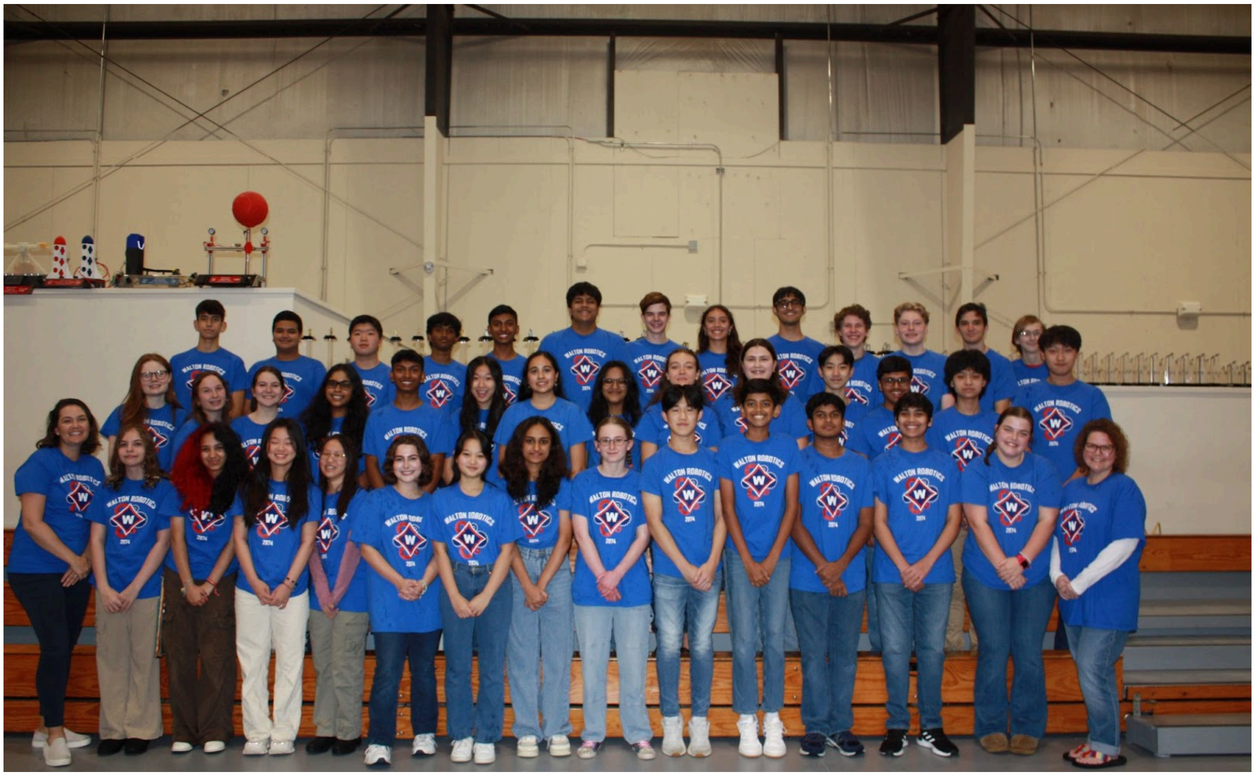
Solution:

To get rid of this slop, we decided to get rid of the Planetary mount and utilize the new REV Ion MaxSpline shaft, as the purpose of this shaft was to transfer torque without any slop. We used the MaxSpline mounting plates to transfer torque through the shaft and onto the shooter, and we made a custom sprocket to continue using a chain to drive the pivot.





Also, we added hex to MaxSpline adapters onto the ends of the MaxSpline shaft, and ran the existing hex shaft through the max-Spline. This way, we could essentially make the wrist hot-swappable, and we could keep the old design as a spare. The redesign made the wrist more consistent and robust.



In Conclusion …

Thank you to our mentors, coaches, and parents! Without their support, we would not have had the amazing season that we did.

If you have any questions regarding our team, robot, strategy, training, etc., feel free to reach out and we will be more than happy to answer!

Chief Programmer: Nikita Chacko (Nikita.Chacko@waltonrobotics.org)

Chief Engineer: Zachary Evans (Zach.Evans@waltonrobotics.org)

Chief Engineer: Samit Kadasinghanahalli (Samit.Kadasinghanahalli@waltonrobotics.org)

Strategy Director: Vishruth Thallam (Vishruth.Thallam@waltonrobotics.org)