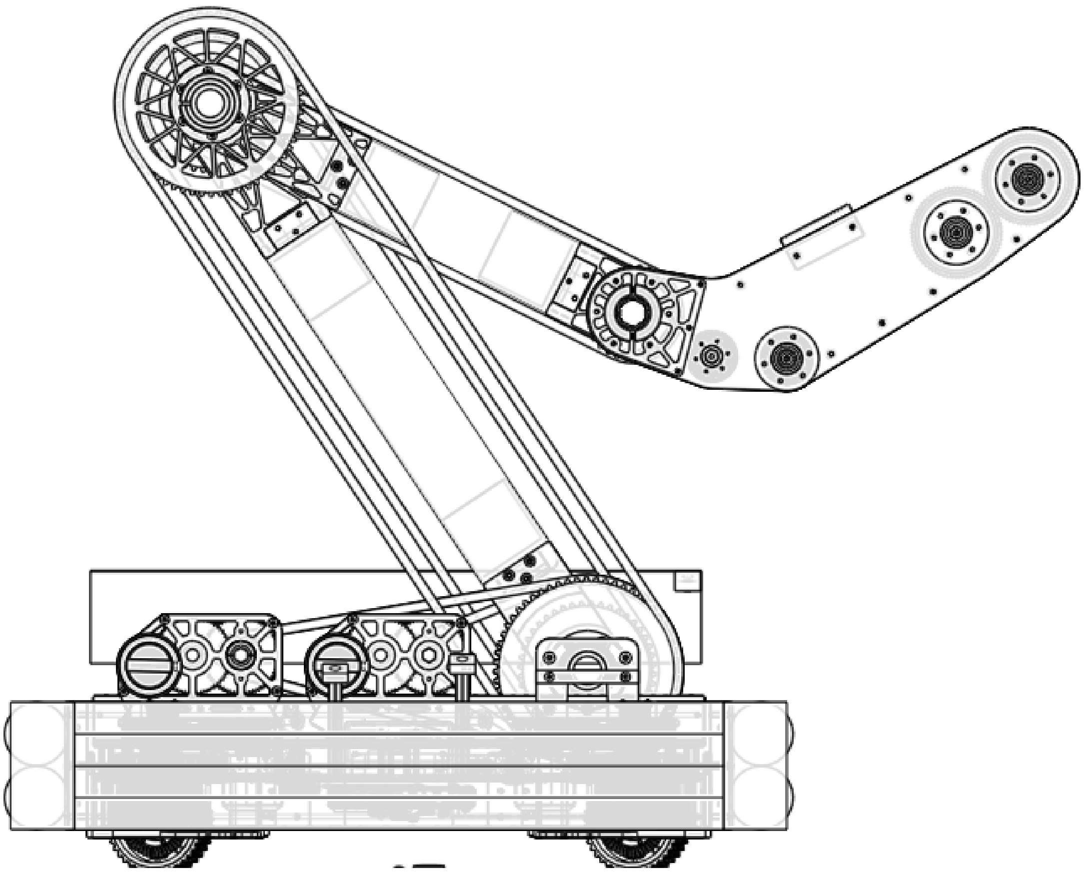
CHARGED UP

PRESENTED BY HAAS Gene Haas Foundation

PROUD MEMBER



FRC WORLD CHAMPIONSHIP EDITION

LUXO

WOLVERINES

1757

2023 TECHNICAL BINDER

FORWARD

Hello, and let us welcome you to FRC Team 1757's 2022-2023 Season. This season has continued the tremendous growth in our robot's design and technical ability that started last year as our team emerged from the hibernation of COVID-19 to become a surprising contender in the New England Region. Continuing to recruit rookie students to supplement our now more veteran team members and Senior Mentors, we have pushed our collective talents to their limits to deliver the competition-worthy robot contained within the pages of this binder.

Our season started in the fall of 2022, introducing a new class of over 10 freshmen, sophomores, and juniors to the world of FRC. We showed off the robot at local town events, built a T-Shirt Cannon to raise school spirit at the prep rally, and hosted weekly technical seminars on everything from the engineering process to CAD, Electronics, Pneumatics, Mechanics, and everything in-between. Over the Summer we got a new OMIO X8 bed router and practiced our CAD and fabrication skills by designing and building an enclosure for the machine. We traveled to Billerica, MA in October to compete in the first-ever New England Robotics Derby. We finished in Second Place, losing in the Finals (The best competitive finish in team history). We piled into our classroom on a cold Saturday morning in January, eagerly anticipating this year's game. 4 CAD models, 8 shared Google Drives, ten weeks, 20 Weekend Build Sessions, 50 Zoom calls, 5799 lines of code, 170 git commits, 19,129 discord messages, and many, many cups of coffee later, we are proud to unveil our robot "LUXO" for the 2023 FRC Season.

Why did we name the robot LUXO? Is it because of the shining lights on its frame that illuminate what game piece we are looking for on the field…no. Is it the bright shining future of the team…no. Is it a reference to solar power and how that ties into the theme of this year's FIRST season…good guess, but no. In truth, we are a bunch of animation nerds, and we thought the robot looked like the lamp in the Pixar Animation title sequence named Luxo. Not every robot name has a deep prophetic meaning…sometimes it's just about the memes.

One very exciting thing about this year is that Team 1757 joined the Open Alliance. We found the Open Alliance teams and their open and timely build season updates so helpful to our team last season that we decided to join so we could help other teams the same way the alliance has already helped us. In addition to frequent updates on our build thread, we also made two appearances on the Open Alliance Show Streamed on twitch. If you want to learn even more about our robot and the design process, beyond what is contained in this manual, please visit our Chief Delphi Build Thread at https://www.chiefdelphi.com/t/frc-1757-wolverines-2022-2023-build-thread/416564

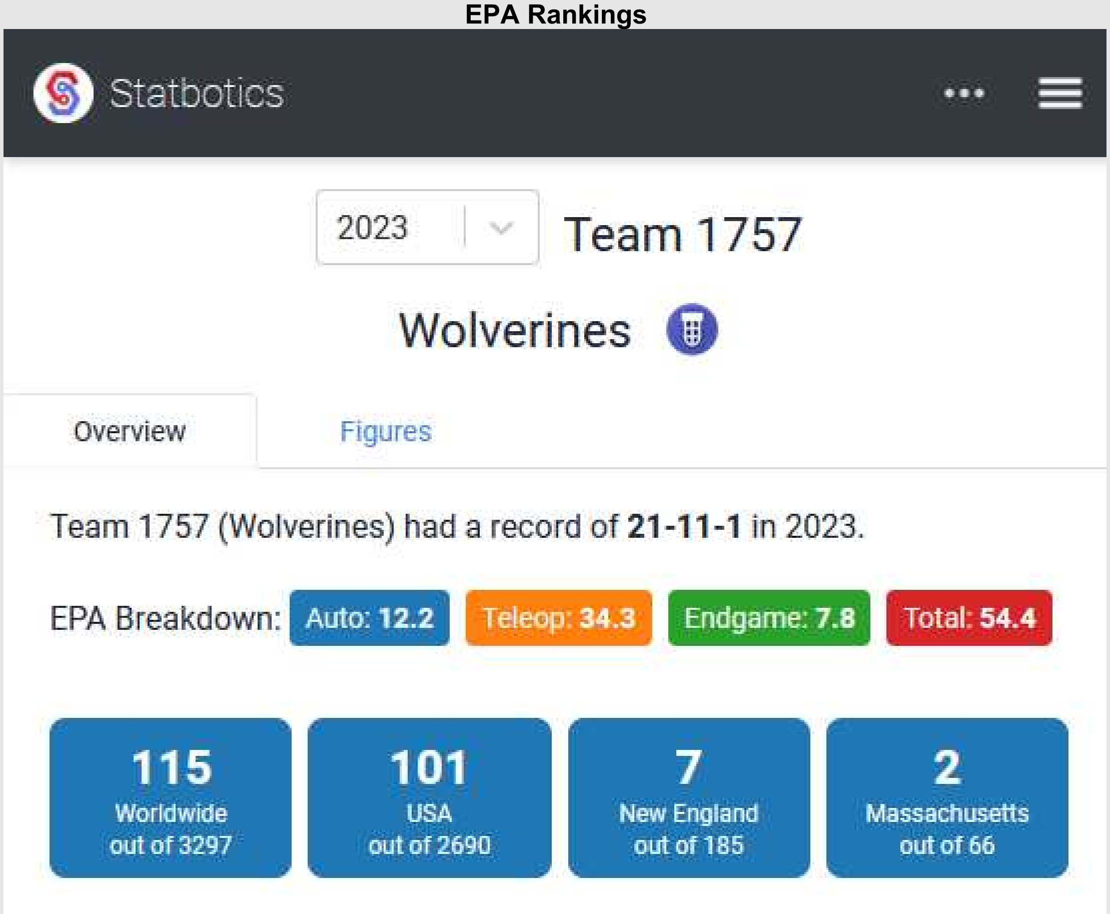
We hope you enjoy this brief look at the design process and technical details that went into this robot, and if you have any questions, look for one of our team members in the stands, in the pits, or on the field. We are always ready to share the knowledge we have gained and share a few hard-learned lessons we learned along the way.

DCMP Update

So it has been a whirlwind of a season so far, after meddeling performance at Greater Boston district we went on win the WPI District Event. Not only were we Alliance captian of the the #2 alliance, we also won the Engineering Inspiration award at WPI. Though out this document you will find various updated information featuring design changes/Repair/modifications that were made during the competition season.

Competitive Record Though District Play:
22-11-1
Greater Boston District - Deans List Semi-Finalist: Sean Tao
WPI District Event – Winners
WPI District Event – Enginnering Inspiration Award Winnerd

EPA Rankings



World Championship Update
We thought our seson couldn’t get any better then taking home the teams first ever blue banner at WPI. We were wrong. We came into Wilson Division at New England Championship a solid middle of the pack Contender, however we quicky proved why we were there, our robots consistent and Reliable play led us to take #1 overall at the end of Qualifications, after picking the highest rated offensive bot on the field 176 Aces High, we picked up 1699 Robocats to round out a great alliance. We went undefeated in the Wilson Divison playoffs, taking home another blue banner before taking on the Mier Divison winners for the New England District championship. With the Championship Tied 1-1, we went into a nail-bitting sudden death match where we came out on top.

Please review our OA thread on Chief Delphi for more details.

Competitive Record Though District Championship Play:
39-13-1
Greater Boston District - Deans List Semi-Finalist: Sean Tao
WPI District Event – Winners
WPI District Event – Enginnering Inspiration Award Winnerd
NE Championship – Wilson Divison – Winners
NE Championship – Wilson Division – Excelence in Engineering
New England District Championship - Winners

EPA Record

TABLE OF CONTENTS

GAME ANALYSIS

Every FRC season starts the same way; we gather together as a team, watch the kickoff stream, then hunker down and break down the game in back-to-back 8-hour build sessions. The hope is that by the time we walk out the door on Sunday night, we understand the game and know what we are doing.

After carefully considering the different ways you can score points, we concluded that placing GAME PIECES on the NODES was the most critical ability in this game, with it having the highest potential points available. Without the ability to DOCK and ENGAGE, however, it will be virtually impossible to remain competitive due to the lack of ranking points.

After two days of deliberation, these are the design Requirements we settled on.

DRIVE

- Need to be a Small Bot – The smaller the bot, the easier it is for 3 robots to balance on CHARGE STATION
- Need a low center of gravity
- Need to be able to drive and balance on the CHARGE STATION.
- Preferably autonomous balancing on CHARGE STATION
- Use of vison (April Tags) to provide feedback to the onboard odometry system
- Use of vision to identify and seek out game pieces on the field.

ARM

- Arm needs to be strong and durable
- Use Encoders on the input and output of gearboxes to monitor and minimize backlash.
- Either 2 or 3 Degrees of Freedom Further testing will be needed.
- Needs to score at all 3 levels BOTTOM, MIDDLE and TOP Nodes.

INTAKE

- Quickly acquire GAME PIECES (Touch It – Own It)
- MUST pick up CONES and CUBES from the LOADING STATION
- MUST pick up CUBES and upright CONES from the ground.
- Would like to be able to pick tipped-over CONES from the ground.

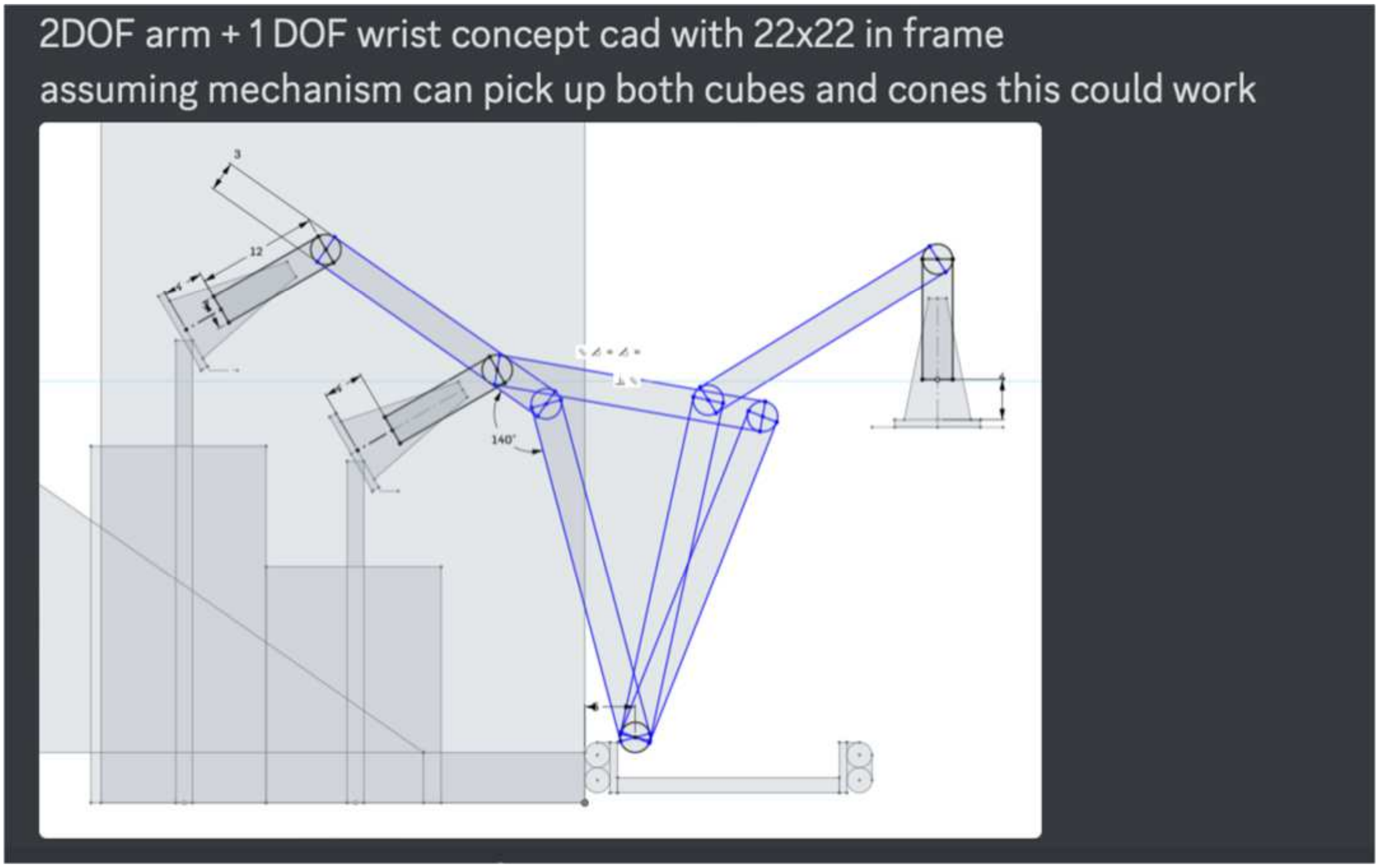
GENERAL DESIGN CONCLUSIONS

Our overall goal for the season was to be a competitive bot in district-level play and qualify for New England Championship. To accomplish this, we need to, at the bare minimum, make it to Elimination at both our district events, hopefully as an Alliance captain or 1st pick.

We approached our design as trying to build a highly reliable jack-of-all-trades bot, focusing on gaining one of the two performance-based ranking points in either match.

Inspired by the cost-effective production strategies of the Hass Formula 1 racing team and our limited team members and design resources, we prefer to use pre-engineered solutions wherever possible to focus our design resources on critical complex components.

Identifying Design Constraints



We are thinking about using an arm as a manipulation mechanism. We potentially envision a 2DOF arm + 1 DOF wrist that can pick up both cubes and cones, with a high range of motion on the wrist joint. As we can utilize the bot's movement, we do not need the Arm to move from side to side. An important note is that with an arm the starting configuration poses a good challenge, as it will need to fit inside of the robot's frame before activation. We have found that the shoulder joint only needs to move 90 degrees max, the elbow joint 210 degrees, and the wrist joint somewhere like 270 (at least in the configuration, lots to play with) to achieve all necessary motion.

THE 1757 RAPID DEVELOPMENT MODEL

DEFINE

- Clearly Identify the design requirements of the system

PROTOTYPE

- Design and Build a prototype that can be used to test design assumptions and Test

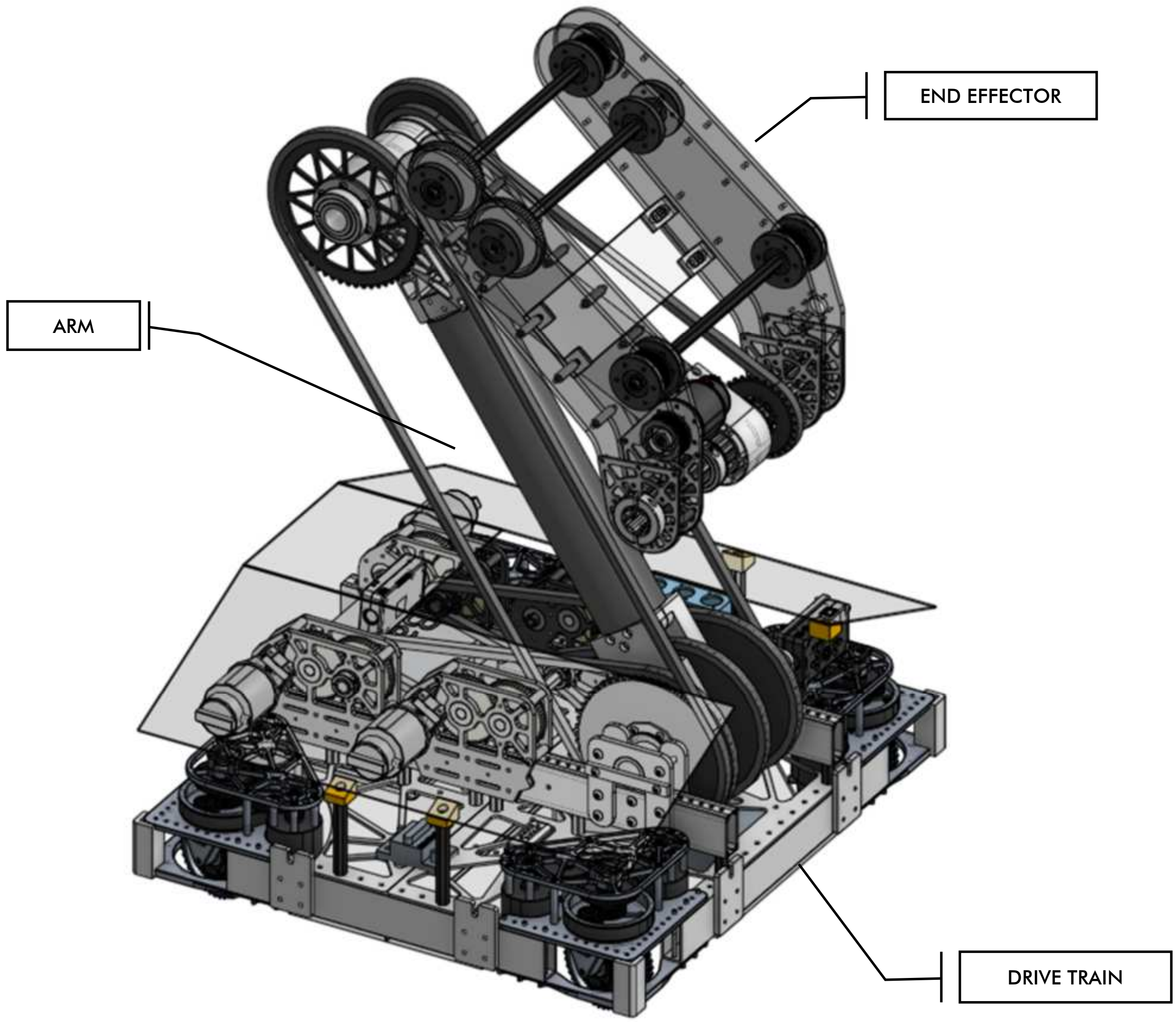
REFINE

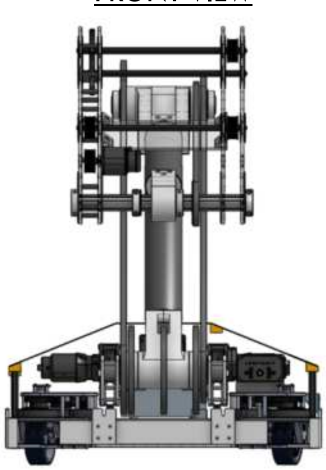
- Use what we learned from testing to develop a final design

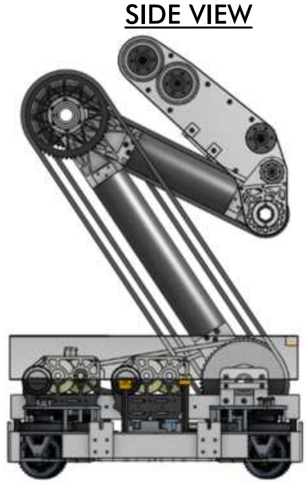
DEPLOY

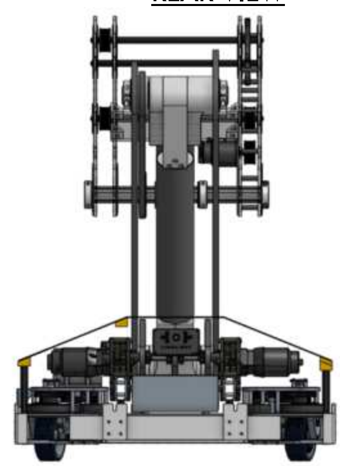
- Fabricate final version and intergate into overall robot systems

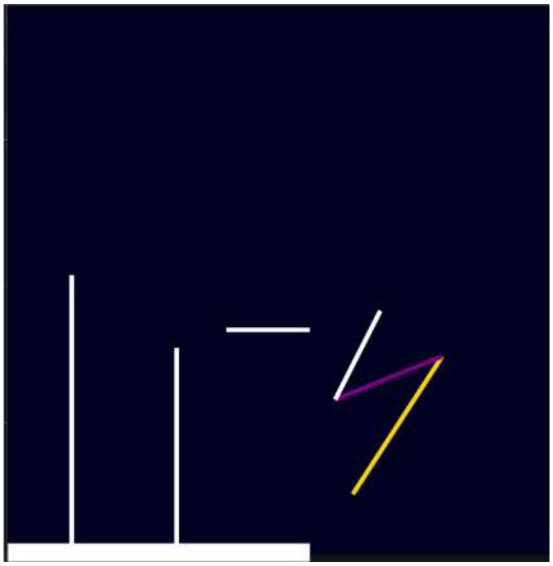
FINAL ROBOT DESIGN



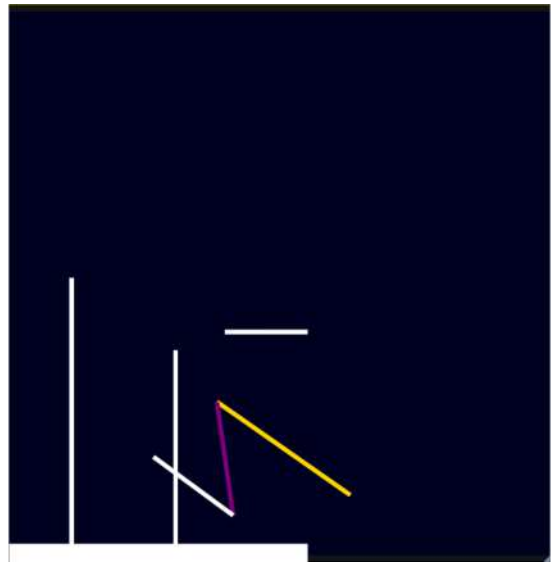




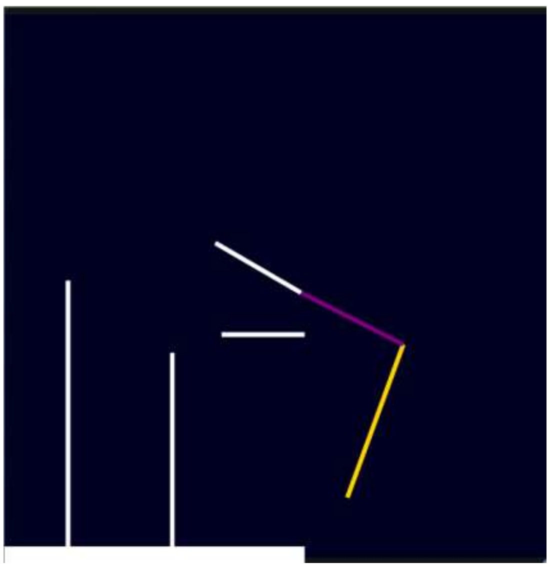




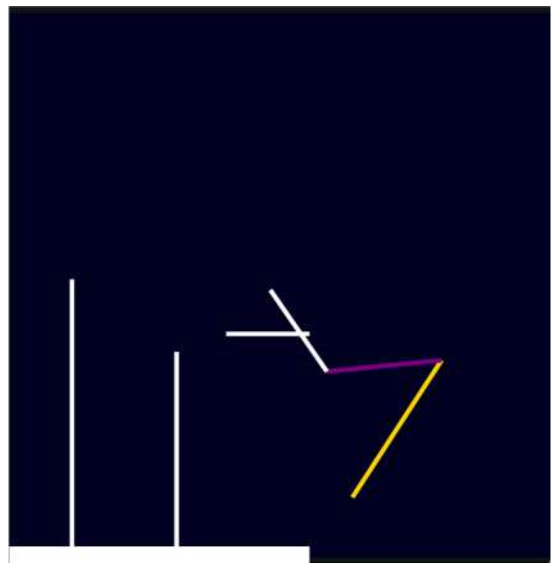
Default Configuration



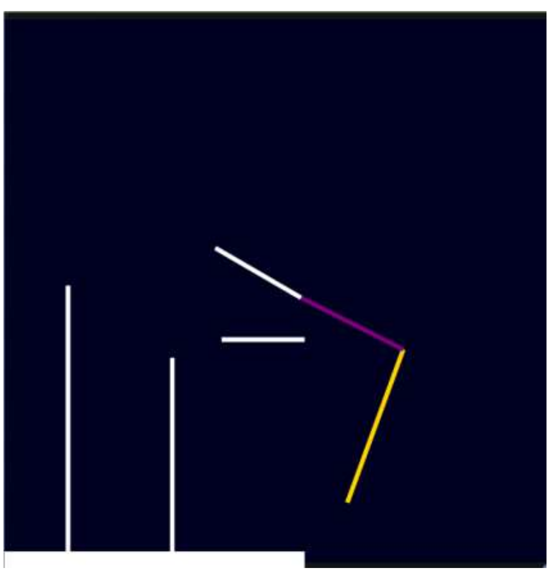
Floor Pickup



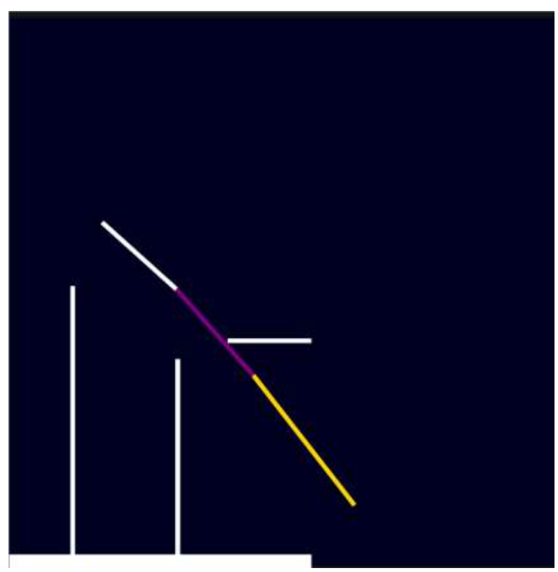
Substation Pickup



Score – Bottom



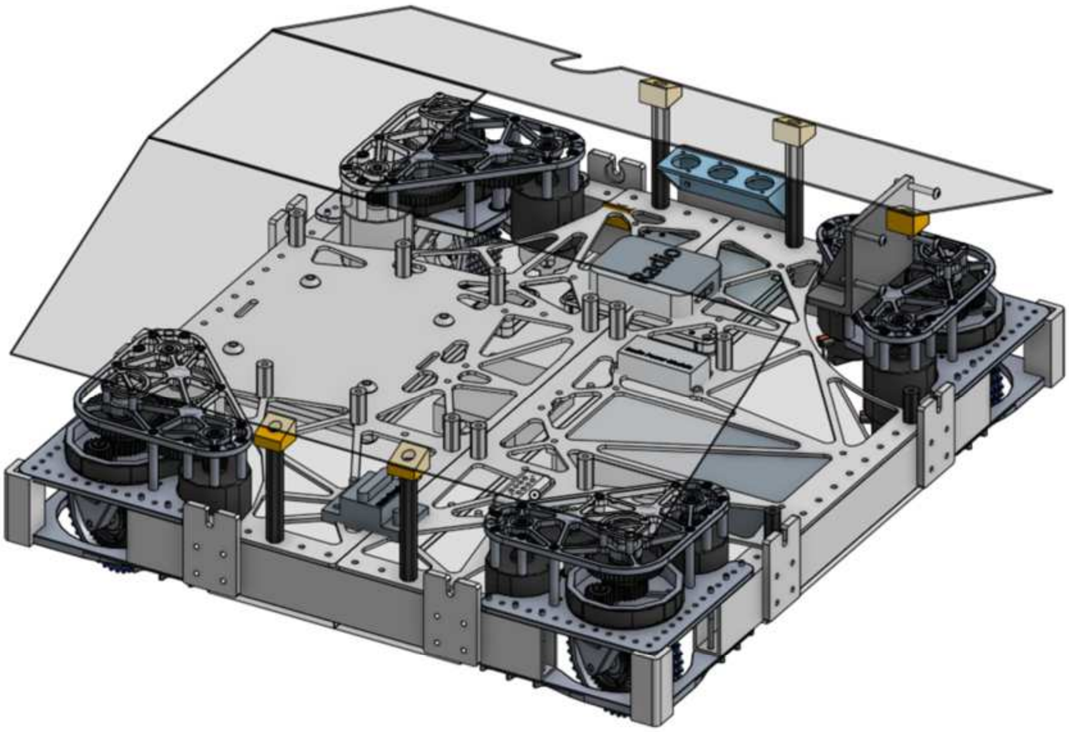
Score – Middle

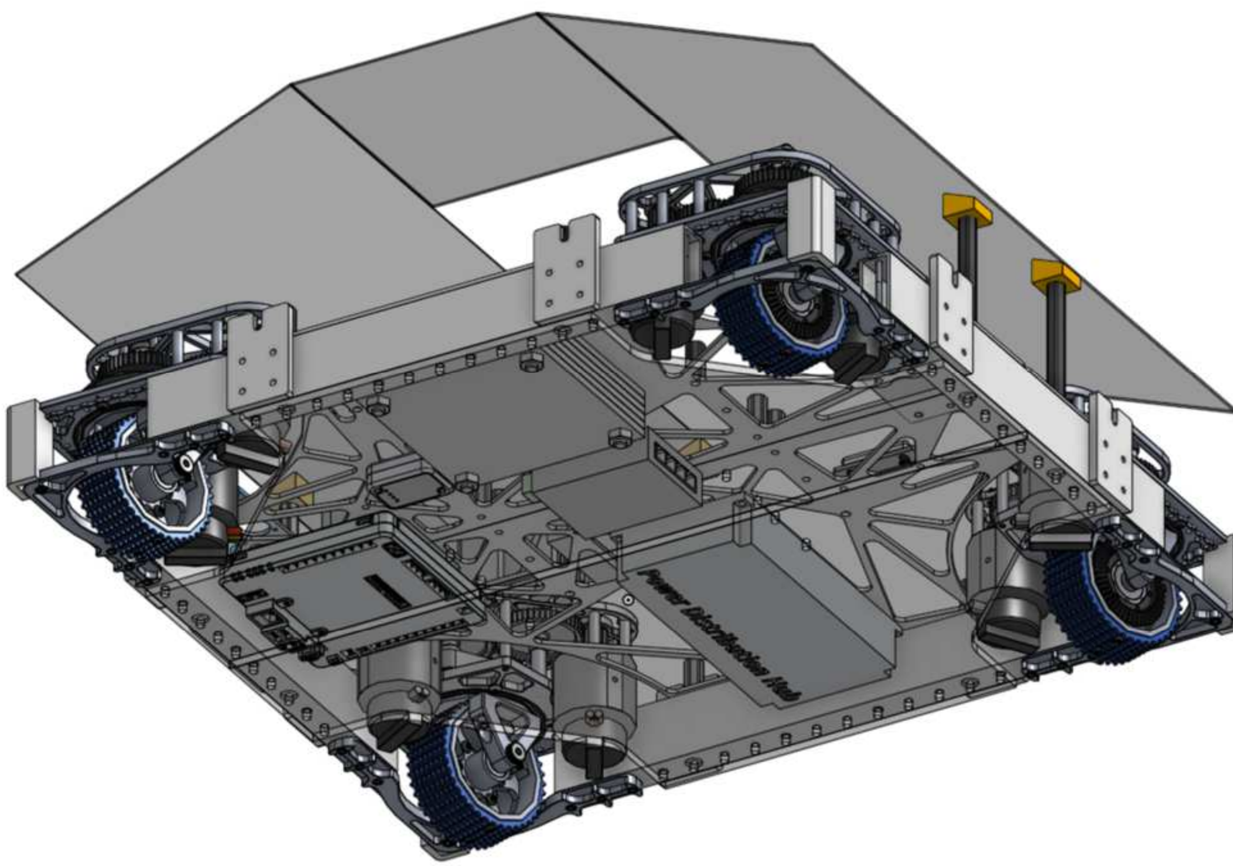


Score - Top

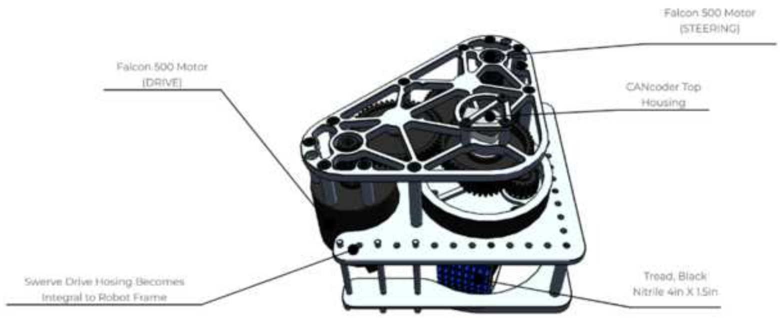
Above – 3D Simulation of Arm Joints in all of its various Arm Configurations

MAJOR SYSTEM #1: DRIVE TRAIN





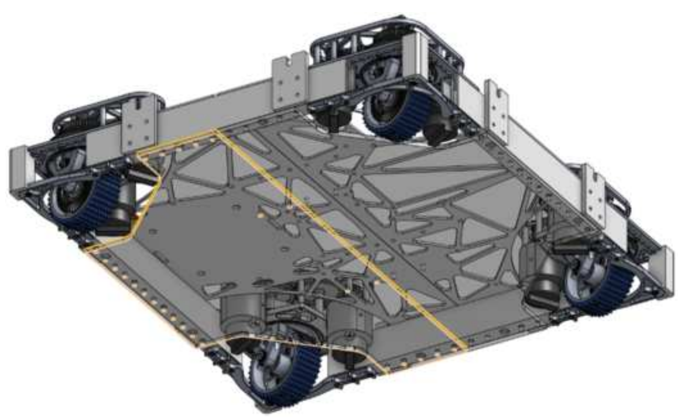
1.1 - SWERVE DRIVE MODULES



Last season was our first competitive season using swerve drive and we could not be happier with the results, because of the equal power between steering and driving there are none of the performance trade offs inherent in other drive systems. We can still run circles around the field when we need to and we can still push another robot across the field when they are in our way. One of our favorite features exclusive to swerve drive is what we call the park feature, by turning all 4 wheels to a 45° angle relative to the corners of the robot the robot effectively parks itself in place and wont move, another robot can push against us all match long and we wont move. Last year we used Swerve Drive Specialties Mk4 units, and this season we upgraded to the newly released Mk4i units. This revised design points the motors downward into the bot instead of mounting above the module. This allowed us to eliminate ≈2" of vertical space in our robot between the drive frame and the major systems.

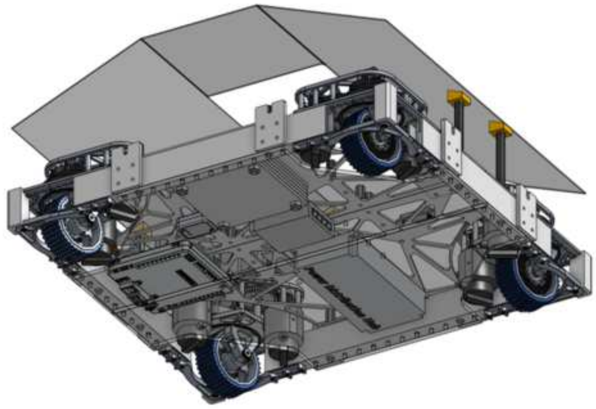
SDS MK4i Swerve Modules	
Powerplant	Falcon 500
Gearbox Configuration	L2
Overall Gearbox Ratio	6.75 : 1
Unadjusted Free Speed	16.3 ft/sec

1.2 - ROBOT FRAME



Left – CAD – Isometric Top View of robot frame
Right – CAD – Isometric Bottom View of Robot frame

1.3 - ELECTRONICS SUBSYSTEM



One thing we struggled with on our 2022 robot was how inaccessible most electrical components were. On our 2022 robot, all the electronics sat in a belly pan at the base of the bot, and the only way to access most of the components required you to remove the majority of the Robot Systems. Inspired by another team's design from 2022, we decided to hang all of our electronic components upside down and face the ground. Now to access the electronics, we tip the bot on its side, remove the ¼" protective polycarbonate plate, and you have full easy access to all the electronics components. (Credit to Team 125 for the Idea, they have the thanks of a grateful drive team and Pit Crew)

ELECTRONICS SYSTEM MAJOR COMPONENTS

- (1 ea) National Instruments - RoboRio 2
- (1 ea) REV Robotics - Power Distribution Hub
- (1 ea) Navex 2 – RoboRio MXP expansion Board
- (1 ea) CTRE CANivore
- (1 ea) CTRE CANdel
- (1 ea) BrainBoxes – SW-015 5 Port Gigabit Switch
- (1 ea) Generic Passive POE Injector
- (1 ea) Limelight 2 Camera
- (12 ea) Falcon 500 Motors
- (1 ea) REV Robotics Sparkmax brushless motor controller
- (1 ea) REV Robotics NEO 550 Brushless DC Motor
- (1 ea) Open Mesh Access Point [Insert Model Number]

1.4 - TESTING PORTS

We added a convenient patch panel to the upper side of the robot to allow for quick access to essential data ports when we don't want to access the underslung electronics.

PATCH PANEL SLOT 1 – USB TYPE A
This slot connects to one of the USB Type A ports on the RoboRio. This typically has a USB flash drive plugged in. During a match all the system logs are copied to the USB drive. After a Match, the USB drive can be pulled and opened up in AdvantageScope on the debug machine for post-game analysis. It's our version of a Blackbox on an airplane.

PATCH PANEL SLOT 2 – USB TYPE B
This connects to the USB Type B Port on the RoboRio—a redundant method for tethering the robot for control and debugging at events.

PATCH PANEL SLOT 3 – RJ45 CONNECTOR
This connects to the Ethernet Switch Via CAT5e for network access. Used for tethered connections to the bot during testing. Ethernet tethering is preferred, but we have encountered software reliability issues in the past.

DCMP Update

At the Revere District event we ran into serious problems tethering to the robot via ethernet and via USB B. we traced the ethernet tethering problem to a problem with the network configuration issue on the driver station laptop. We were unable to determine a definite cause of the USB-B connection issue, but, we think it most likely to to poor quality of the 90° usb connector used on the robot. From that point on we connect a USB-B cable directly into the port on thr

At the same time we realized we needed a button to manually put the arm motors in coast mode for serviceability when the bot is not connected to the driver station. Since we are no longer using the USB-B testing port we replaced it with a momentary push button switch.

1.5 – CAMERA/VISON SYSTEMS

LIMELIGHT 2 – CAMERA

We are utilizing a Limelight 2 Camera for a variety of tasks on the robot mostly devoted to sensor fusion and automation of systems using computer vision. The Limelight's field of vision (FOV) is essentially parallel to the floor and at the height of the April Tags.

Please refer to the Software section of this document for more information on how we use the limelight and April Tags to improve the Onboard odometry of the robot.

1.6 – COUNTERWEIGHT

Not originally intended as part of the design, upon testing of the robot with the Arm fully extended in the scoring position, we realized that robot was prone to falling forward. To resolve this, we looked to add ballast to the bot. First we thought of lead but didn't want to deal with the potential health risks of improperly encapsulated lead. We investigated tungsten; however, a review of the current price of tungsten plate (≈$40/kg) quickly ruled it out as a potential candidate. We settled on 6" x6" x1/4" steel plates mounted directly under the robot battery. After testing with different #'s of the plate, we decided on 6 Plates with a total weight of ≈25 lbs. Now the bot is highly stable even when the Arm is fully extended.

1.7 – PROTECTIVE COVERS

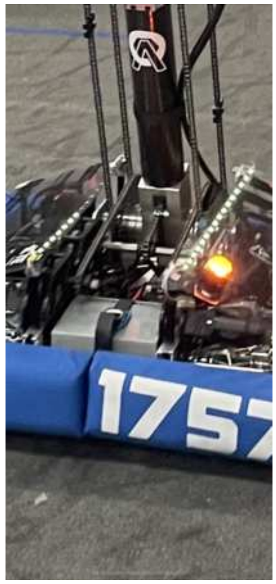
We added protective covers that Slope away from the central arm structure down to the bumpers. Not only do these plates provide a valuable location to display all of our great sponsors they serve to prevent errant game pieces from getting stuck inside the robot during a match.

DCMP Update

Originally the protective covers were only hled on with 3M™ Dual Lock™ SJ3560, this material is nice because it very strong but eaisily removable. Durring qualifying matches in Revere however, these panels kept falling off and dragging around the fieild. The Dual Lock strips were reinforced with zip ties and these held through all of playoffs in revere, and all of qualifications at WPI. Then in Playoff Matches we shered off 3 of the metal standoffs holding up the protective covers. We made quick repairs to keep going however prior to DCMP we will be swaping out all the ½ thunderhex standoffs with 1" 80/20 extrusion with hardened bolts for strength.

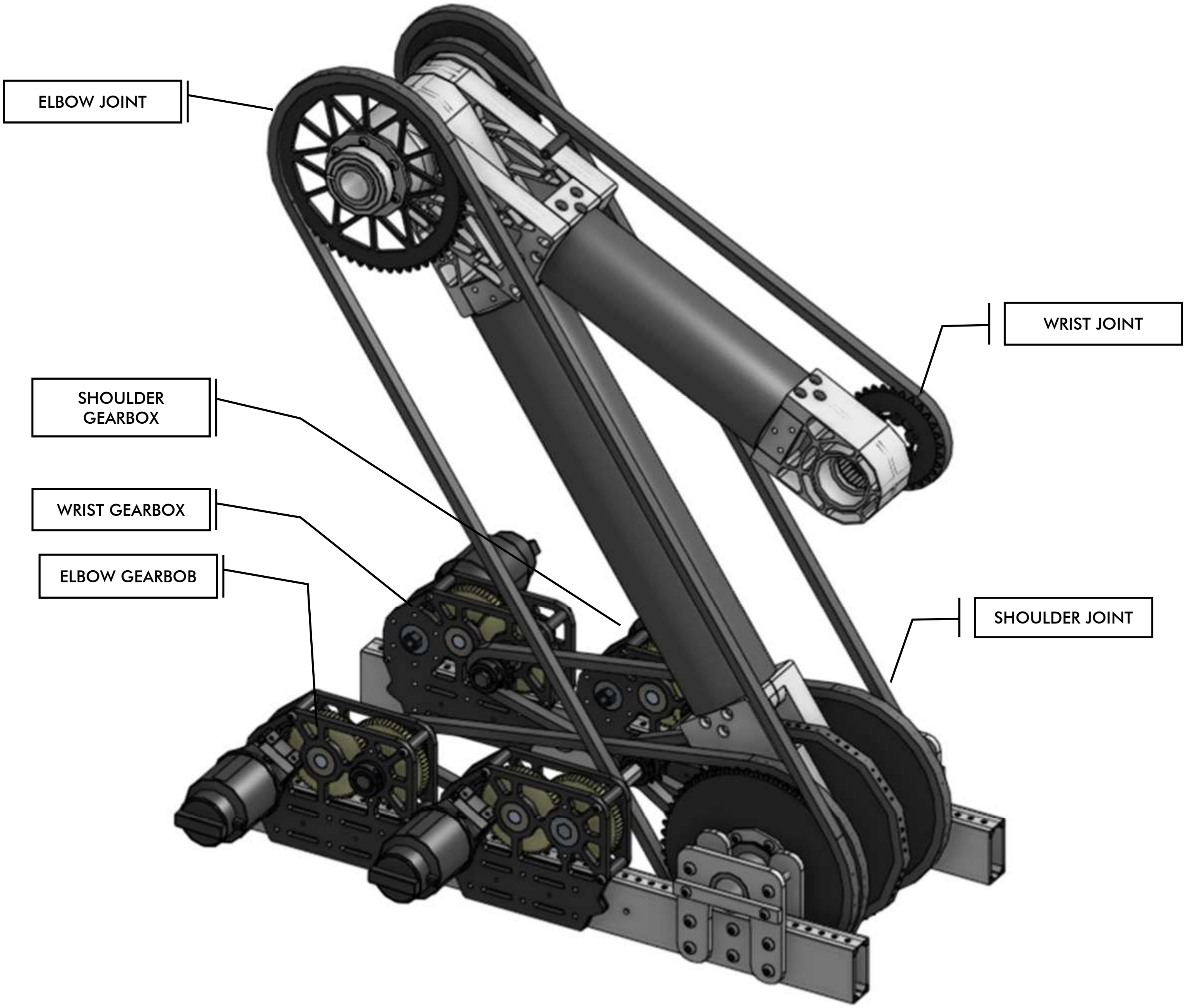
1.8 – GAMEPIECE INDICATORS

One of the hardest things to do in a match is how to signal between the driver station and the human player what game piece you want them to load into the robot. People use Hand Signals, Colored pieces of paper or guess. We wanted to take the guesswork out of the equation, so we mounted 2 LED Strip lights along the top of the protective covers. The driver controls what color these strips are so he can communicate to the human player which game piece to feed to the bot – Yellow for Cones and Purple for Cubes.



Left – Robot Displaying "I Want a Cone"
Right – Robot displaying "I Want a Cube"

MAJOR SYSTEM #2: ARM



2.1- MOTORS & GEARBOXES

To keep the robot's center of gravity low and keep the Arm as simple as possible we decided to locate all of the heavy motors and gearboxes at the base of the superstructure.

GEARBOX # 1 – SHOULDER GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into Custom Gearbox based on WCP Gearbox design.

GEARBOX # 2 – ELBOW GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into COTS WCP Single Speed Gearbox.

GEARBOX # 2 – WRIST GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into COTS WCP Single Speed Gearbox.

2.2 - CHAIN DRIVE

Using a combination of dead and live axels we transfer the power of the gearboxes up though the Arm to power each of the individual joints. For Reliability and durability, we chose to use #35 roller chain rated for 11,000 lbs of force.

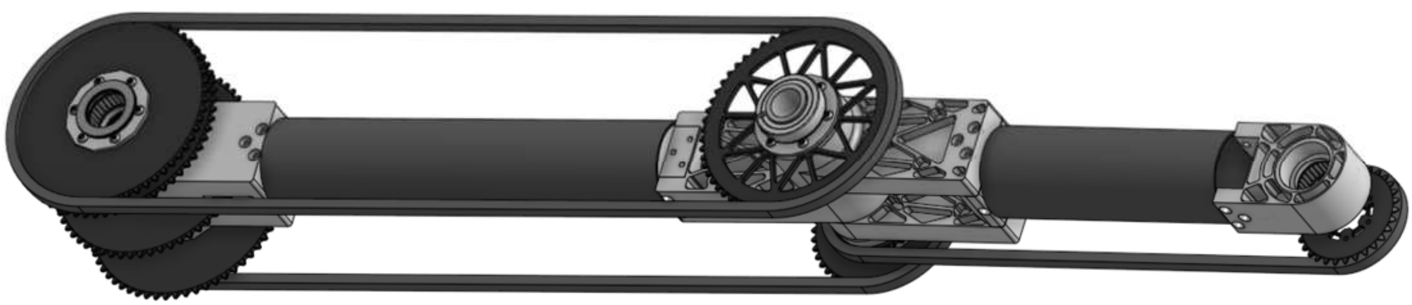
Below is a summary of the different chain runs on the Arm

CHAIN DRIVE 1 – SHOULDER
Shoulder Gearbox Output 12t Sprocket → 60t sprocket on Shoulder (Dead Axel)

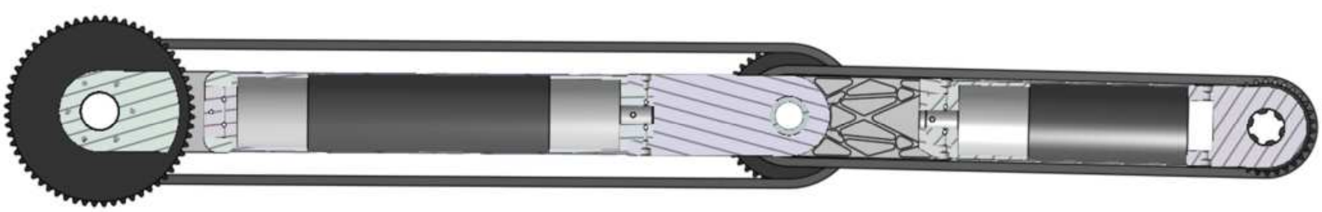
CHAIN DRIVE 2 – ELBOW
Elbow Gearbox Output Shaft 12t Sprocket → 60t Sprocket on Shoulder (Live Axel) → 70t Sprocket on Elbow (Dead Axel)

CHAIN DRIVE 3 – WRIST
Wrist Gearbox Output Shaft 12t Sprocket → 60t Sprocket on Shoulder (Live Axel) → 60t Sprocket on Elbow (Live Axel) → 40t Sprocket on Elbow (Live Axel) → 32t Sprocket on the Wrist (Dead Axel)

2.3 - ARM STRUCTURE



Above - 3-D View of Outstretched Arm



Above - Section View Though Center of Carbon Fiber Arm

CARBON FIBER ARMS

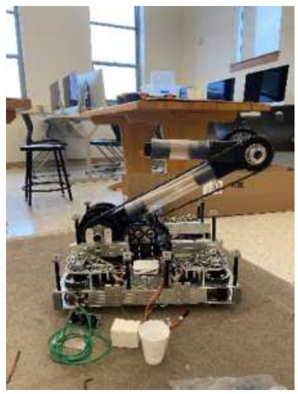
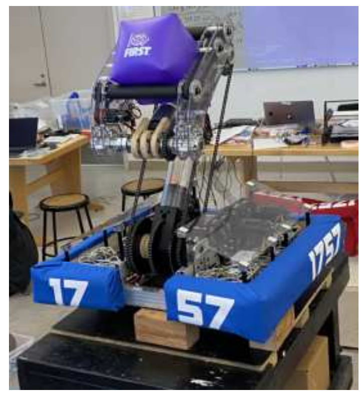
We chose to use carbon fiber tubes as the main structure of the Arm due to its strength and lightweight, the more weight we could save on the Arm the lower we could push the robot Center of Gravity. Carbon Fiber tubes are a stock McMaster item 3" Ø. Carbon Fiber is Epoxy bonded to 3" hollow aluminum plugs bolted to the aluminum joints.


Left – Final carbon fiber arm links after final glue up

Right – Final aluminum plugs used in the ends of the carbon fiber tubes

3D Printed and Polycarbonate Prototypes

Because we knew the carbon fiber and machined aluminum would take time and money to manufacture, we heavily used 3D-printing to make prototypes of the Arm and test and confirm critical geometry before placing final fabrication orders. These prototypes are two fragile to be used on a competition bot but worked well for their intended purposes. We learned very Important lessons about where the concentrations of forces were along the axels and what parts needed reinforcement.

Left – 3D printed Prototype of the wrist joint, printed on a FormLabs 2 SLA Printer

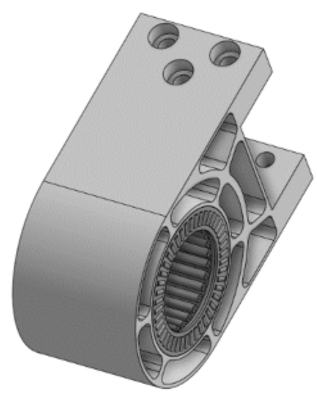
Center – Polycabonate Prototype arm Mounted on bot for the First Time

Right – Fully Assembled "Alpha" Robot build

2.4 – JOINT STRUCTURE

CAD - Shoulder Joint

CAD - Elbow Joint

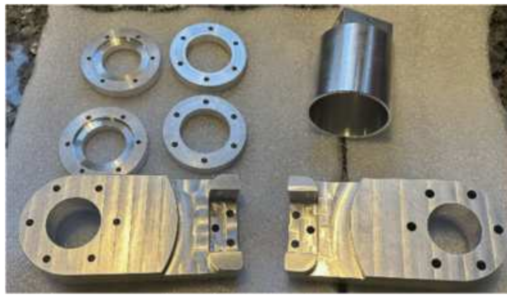


CAD - Wrist Joint

Needle and Thrust Bearings

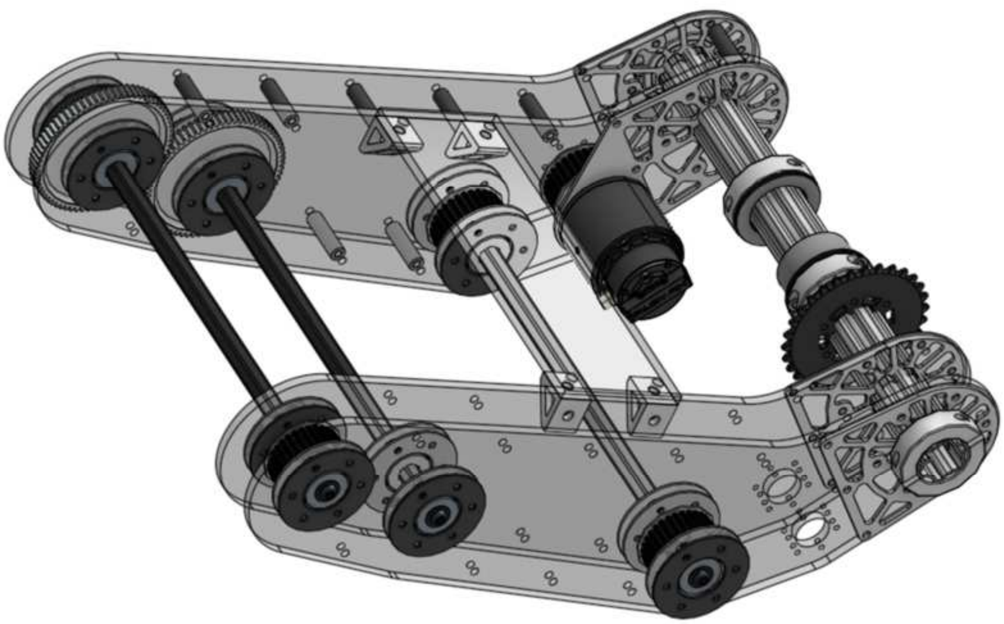
Used in All three joints to allow for smooth rotary motion in each joint.

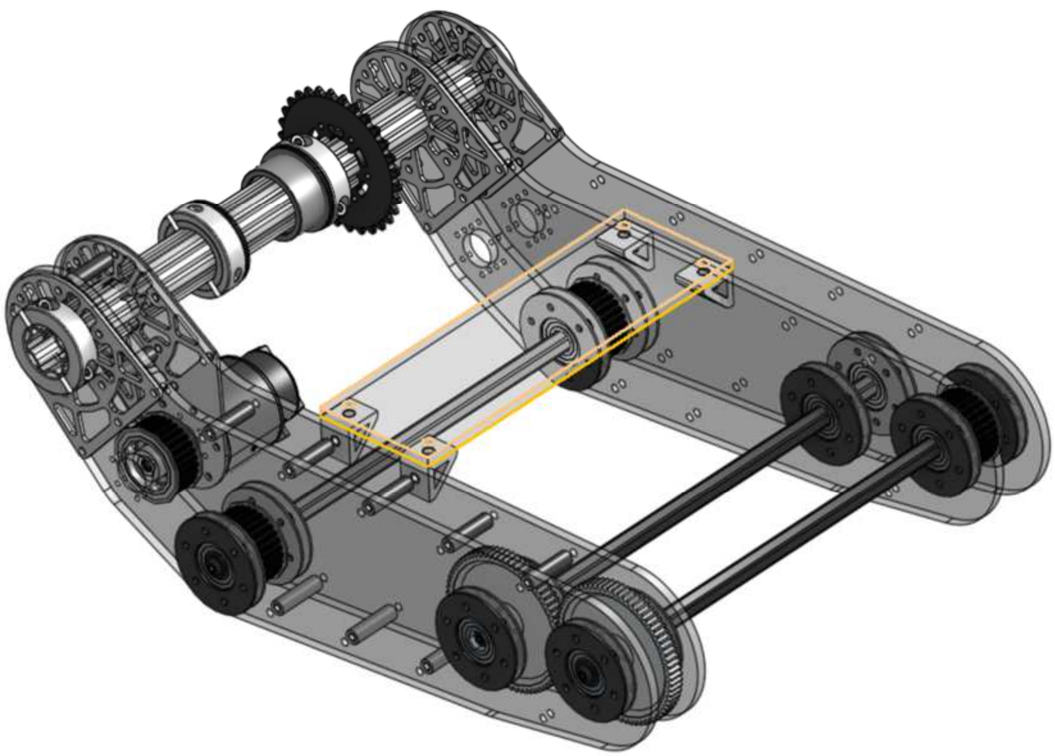





Photos - Final Machined Parts

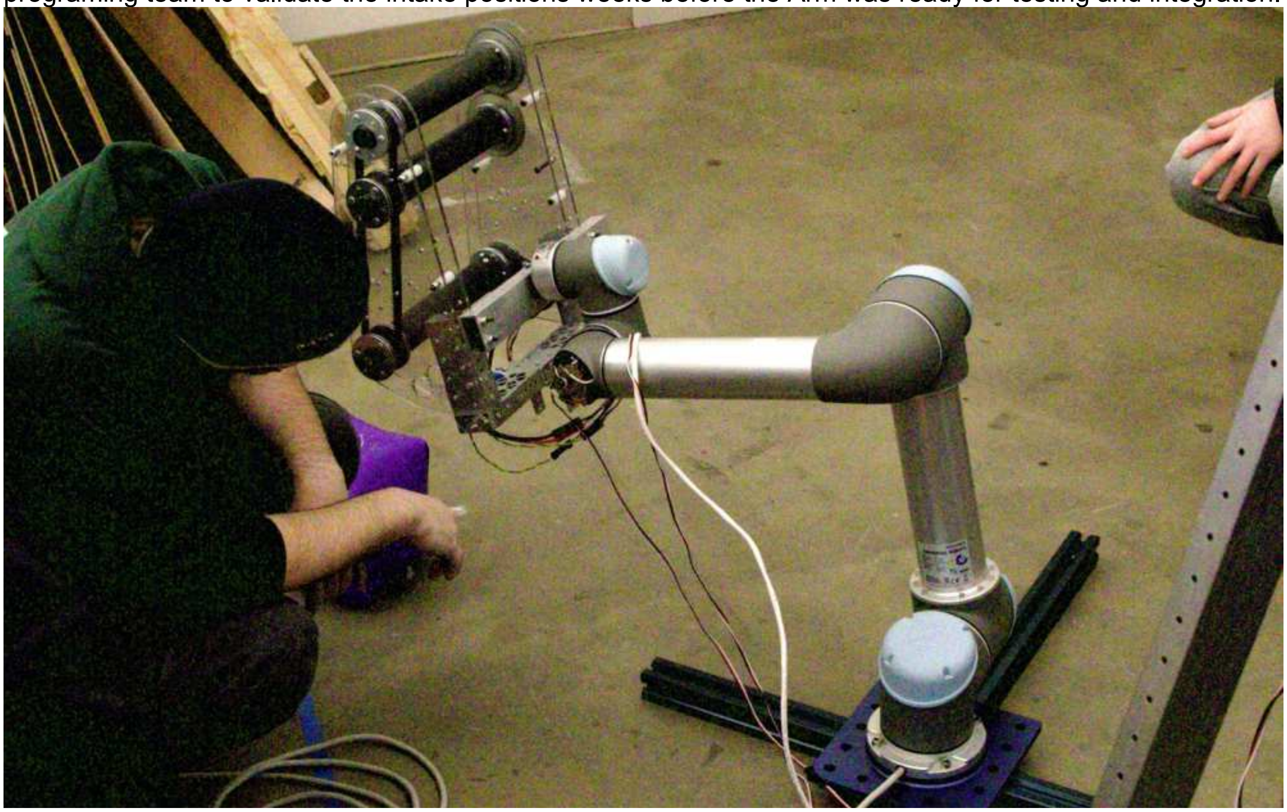
MAJOR SYSTEM # 3: END EFFECTOR





Intake Prototyping – Fun With Industrial Robots

We had a lot of fun prototyping this mechanism, since it was the first major mechanism that we finished we had lots of time to put it trough its paces. One of our mentors has access to a Universal Robots UR3 robot and brought into our lab during one of our weekend build sessions (see images below). This actually proved to be essential because it allowed our programing team to validate the intake positions weeks before the Arm was ready for testing and integration.



3.1 – ROLLERS

The rollers we are using are Vex VersaHub Rollers with ¼" neoprene tubing stretched to cover them, they are very grippy and hold the cube verry securely. We started with the dimensions of the everybody roller for our prototype then made some modifications before settling on final separation differences. The neoprene tubing is undersized for the OD of the polycarbonate roller. We learned a fun trick to clamp off one end of the neoprene tube and inflate it with an air compressor to stretch it over the polycarbonate tube, when the air is released, it makes a perfect friction fit between the Neoprene and the polycarbonate. We have had no detectable slippage after weeks of testing with the rollers.

DCMP Update

After 33 competitive matches one thing is clear, we have problems picking CUBES up off the floor and in order to maintain our competitive edge at DCMP we know we need to be able to get CUBES up off the floor. We thing the majority of the problem is related to how narroe our end effector is, the original design was only 1” wider than the width of the CUBE. To aliviate this issue we are planning to widen the end effector by 3”.

3.2 – MOTORS & GEARBOXES

The intake is powered by a REV Robotics NEO550 Brushless motor into a REV Robotics Ultraplanetary gearbox. The motors small size is nice however because we mounted the Sparkmax Motor controller on the intake as well there is no significant weight savings compared to using a Falcon 500 with an integrated Talon SRX. We may end up swaping this out for simplicity sake in the future.

REV Ultraplanetrary	
Powerplant	NEO550
Gearbox Configuration	4:1, 5:1
Overall Gearbox Ratio	20:1

SOFTWARE

SOFTWARE: OUR DEVELOPMENT ENVIRONMENT

WPILib

The perineal stalwart, we still rely on core elements of WPIlib for robot communications and debugging. WPILib's new Logging features have greatly enhanced our Debugging capabilities

RobotPy

We have found that students have a lot easier time learning python then they do Java or C++ so with the growing support for RobotPy we migrated our Codebase from Java to Python in 2020. As of this March we are an official contributor to the RobotPy project

GitHub

Without Github our level of remote work and collaboration just wouldn't be possible.

SOFTWARE: NEW AND UPDATED TOOLS THIS YEAR

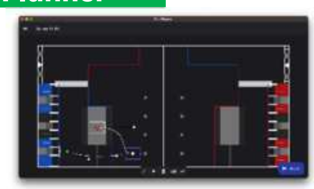
AdvantageScope

WE LOVE ADVANTAGE SCOPE! Not only does it log *everything* but does it in away that is intuitive and easy to review. No more searching though 10000 lines of log files to find the one piece of information we need. Huge thanks to team 6328 for building such a great tool.

PhotonVision

We are using PhotoVision as our native development framework for Computer vision due to its growing wide support inside the FRC community. It does not include native support for RobotPy however so as an offseason project our lead programmer wrote a custom wrapper for PhotonVivion so it can work inside our RobotPy environment

PathPlanner



Last year we used PathWeaver, but we were disappointed in the lack of native support and increased complexity in the development stack so starting with the off season we transitioned all of our Autonomous path planning to PathPlanner. We had much fewer issues with this system.

SOFTWARE: DRIVE

Taking off of last year, the drivetrain codebase has stayed the same. We are running field oriented drive with robot relative rotation to allow for quick maneuverability. A button to align to the nearest 90 degree angle was added to help with driver alignment. This state slightly reduces the speed and snaps the angle of the robot in order to have perfect alignment to the double substation, single substation, and grid every time. For our automated balance sequence, we work in robot relative space on the robot relative gyro.

A Brief Tangent - Absolute Relative Drive

Last year our lead programmer had a new idea for drive control, an absolute relative drive. The common swerve drive control method was to have a field relative translation for the bot, and a robot relative rotation. What this meant is a left input on the rotation axis would result in the robot rotating to the left at a constant speed. A translation action was not affected by rotation but instead was in "field relative" space. The difference of absolute drive is that the rotation is also field relative. A left input on the rotation stick will yield the robot turning to face left. This year we expect this type of robot control to be very important for drivers when they have to be able to turn to specific positions for collection and scoring on swerve drives. You can see this in action in any one of our videos from last year. Having fixed controlled rotation will allow for precise driver input and less fiddling with controls when cycle time is very important.

The drivers have also experimented with alternate driving methods on swerve to get used to interesting control schemes such as a curvature drive, standard tank drive, standard field relative drive, and full robot relative drive.

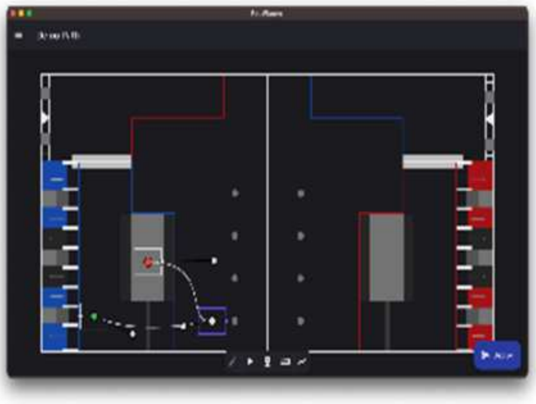
SOFTWARE: INTAKE

The intake is using a state machine in order to regulate its expected behavior. There are 2 enumerable values: one for the gamepiece intended direction (intake, outtake, hold) and one for the desired gamepiece type (cube, cone). The transition between each state is dictated by a user input to any given category. If no input is given, the system holds its position and keeps the desired gamepiece remembered. The state value of the desired gamepiece is displayed to the driver and to the human player through pulsing leds of the respective color.

SOFTWARE: ARM

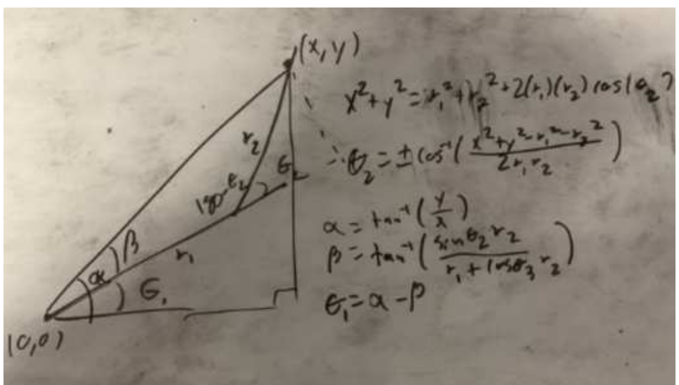
A triple jointed arm is no easy feat in order to program smoothly. From cad, states are given about the end effector's desired position and rotation relative to the floor. From there we use inverse kinematics to determine each per-joint relative rotation at any given position. A cartesian control on the wrist joint's position is added using a trapezoidal PID profile to lay out a path for the Arm to follow. For the wrist it has another trapezoidal PID profile controller. When going state per state on the Arm's motion, we check for if the relative angle goes over software end limits in order to prevent running the Arm into itself. These are done in joint relative space. Since the Arm is controlled from the base through chain and sprockets there is a virtual 4 bar created in which the rotation of any given joint is given relative to the ground. These are converted into motor space and passed onto each motor where they have a position PID controller onboard. For simulating the system we are using mechanism2d to view the expected values of the Arm and be able to run through positions. This simulation first approach has allowed minimal revision and a solid foundational codebase that is mostly complete before the bot is finished. Furthermore the position of the Arm is logged in 3d with advantagescope based on the position it believes the bot is in. Logging was also a priority for a complex system and therefore we log the instructed position and actual position per joint and each end position in cartesian space.

SOFTWARE: AUTONOMUS



We experimented in the offseason with pathplanner and use it extensively for our autonomous. Each necessery command is placed into a potential pool of events for pathplanner to fire. At the beginning a sequence determined solely in pathplanner is fired. Going off of last year we had a large time gap in order to make slight changes so instead for this year we are using the waypoint system and opting to have any given autonomous contained solely in pathplanner. This has increased our ability to construct autos and tweak any given aspect as needed. For the library itself of parsing, the lack of first class robotpy support meant we had the freedom to reimplement autonomous however we pleased based on the path. We follow a changing trajectory and the swerve drive using onboard odometry and a weighted vision estimate determines its bot position relative to the global field and follow through it between each section.

SOFTWARE: SIMULATION



armsubsystem.py

```
    def setEndEffectorPosition(self, pose: Pose2d):

        twoLinkPosition = Translation2d(
            pose.X() - constants.kArmwristLength * pose.rotation().cos(),
            pose.Y() - constants.kArmwristLength * pose.rotation().sin(),
        )

        endAngle = math.acos(
            twoLinkPosition.X() * twoLinkPosition.X()
            + twoLinkPosition.Y() * twoLinkPosition.Y()
            - constants.kArmtopLength * constants.kArmtopLength
            - constants.kArmbottomLength
            * constants.kArmbottomLength
            / (2 * constants.kArmtopLength * constants.kArmbottomLength)
        )

        startAngle = math.atan2(twoLinkPosition.Y(), twoLinkPosition.X()) -
math.atan2(
            math.sin(endAngle) * constants.kArmtopLength,
            constants.kArmbottomLength + math.cos(endAngle) *
constants.kArmtopLength,
        )
        wristAngle = pose.rotation().radians() - startAngle - endAngle

        bottomArmEncoderPulses = (
            startAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kBottomArmGearRatio
        )
        topArmEncoderPulses = (
            endAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kTopArmGearRatio
        )
        wristArmEncoderPulses = (
            wristAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kWristPivotArmGearRatio
        )

        self.topArm.set(Falcon.ControlMode.Position, topArmEncoderPulses)
        self.bottomArm.set(Falcon.ControlMode.Position,
bottomArmEncoderPulses)
        self.wristArm.set(Falcon.ControlMode.Position, wristArmEncoderPulses)
```

Due to our team's resources, virtual simulation is a huge part of our ability to quickly and reliably construct the bot's codebase. Some key examples of simulation are a wrapped onto a simulated falcon motor. Given our team's extensive use of falcons on the robot, a wrapper that provides simulation support allows for the programming team to iterate much easier and creates a cleaner codebase. Each falcon is logging the values of the motor % and the encoder position, as well as an override value to allow the user to manually in simulation change the value for sensor readings. Entire robot configuration is done on a single call and the getting of velocity, position, and percent and the setting of velocity, position, and percent are easy to access functions to allow interfacing with the motors more accessible than the CTRE library. Given this robot also has a NEO550, the simulation system was adopted to have a similar interface for ease of replacement from a falcon to a motor on the intake. We geometrically derived the inverse kinematics for 3 links with a fixed Pose endjoint. Each of these poses actually allows for two configurations of the proximal 2 arm joints (they can simply be mirrored over the line created from the wrist joint to the shoulder joint, however by forcing the sign on the elbow joint they can all be consistent.

SOFTWARE: VISION

NOW WITH APRILTAGS AND PHOTONLIB

We have a vision system complete with sensor fusion for complete robot localization. Last year,we worked with our first complete vision system as a team that resulted in significantly enhanced system performance, and using apriltags will be very important to account for combined sensor error as well as for being able to reliably use sensor data for automated alignment to various points on the field such as the double substation and the grid.

THE HOW

Photonvision generates camera-relative 3d transforms of each apriltag. Since the position of the camera is known and the position of the apriltag is known, the position of the robot can be determined from a single apriltag datapoint. These transforms are fed into a RobotPoseEstimator in order to create a sense of where the robot could be at a given time, this is combined with the gyro and wheel encoder information to get an accurate sense of where the robot is on the field at any given time. This is used in other subsystems when needed, as well as results being logged to AdavantageScope through the usage of each known pose and ghost poseseepaste

GOING FURTHER

We plan on using this odometry data to have automated alignment in complete robot space for important precision actions such as placement of gamepieces on the grid and collection of those gamepieces. Autonomous will also use this data. Perhaps an automatic engagement on the charge station by using the rotation gained from the apriltags will be possible. Overall having a sense of where the robot is on the field is beneficial to aid in other systems.

ENGINEERING TEAM

WHS Faculty Advisors

James Looney & Raul Madera

Team Captain

Charley Marsland*

Team Business Lead	*Team Technical Lead*
Sean Tao	**Luke Maxwell**

Senior Mentors

Dwight Meglan | **Chris Aloisio°** | **Steve Harrington°**

Mentors

Anthony Gelsomini | **Manny Barros°** | **Sean Lendrum°** | **Mark Holthouse** *Mentor Emeritus* | **Amber Maxwell**

STUDENT MEMBERS

Safety Captain	*Graphic Design Lead*		*CAD Lead*
Jacob Kaplan	**Ivan Cai**		**Landon Bayer**
John Santasuoso*	**David Girard***	**Declan MacDonald**	**Liam McWeeney**
Alex Theofilou	**Claire Peng**	**Andrew Gong**	**Rachel Qu**
Baili Jiang	**Nolan folmar**	**Jeffery Pan**	**Melissa Yang**
Divya Siddi	**Vinny Milinazzo**	**David Confoey**	**Adrianna Cirillo**
Elizabeth Lowney	**Jacob Liu**	**Anthony Yang**	**Constantina F.**
Luke Li	**Owen Monahan**	**Erik Curlli**	**Kevin Bai**
Henry Marsland	**Nina Pappas**	**Rachel Lee**	**Hanya Yang**
Sophia Patrick	**Lauren Buza**	**Kaylee Phu**	**Elias Mukdissi**

Westwood Robotics, Inc
Board Of Directors

Steve Harrington° *President*	**Catherine Tseng** *Treasurer*	**Manny Barros°**	**Cheten Gopal** *Clerk*	**Jack Tseng**

**Denotes High School Seniors*

°Denotes FRC Alumni

SPONSORS

Heartlander Surgical	Westwood High School	Medtronic	Act Fasteners
Gene Haas Foundation	TE Connectivity	General Dynamics	IRA Motor Group
Bayer Familt	Capital Group	Yamaguchi Family	Plymouth Rock Assurance
Deadham Savings	Maxwell Family	Schneider Electric	Marsland Family
Porier Apliances	Franklin Templeton Investments	Jake and Joes Resturant	McGolf
Roche Bros		Capers Catering	

VENDORS

Andymark	West Coast Products	Vex Pro	Swerve Drive Specialties
Discord	Google Work	Onshape	Robo Promos
McMaster-Carr	The Home Depot	Cross the Road Electronics	MS Visual Studio
Python	GitHub	RobotPy	Java

1757

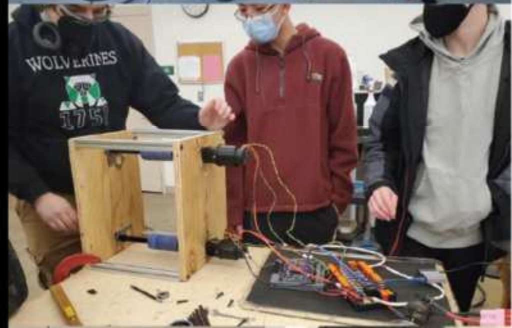
Special thanks to everyone who makes this team possible Including

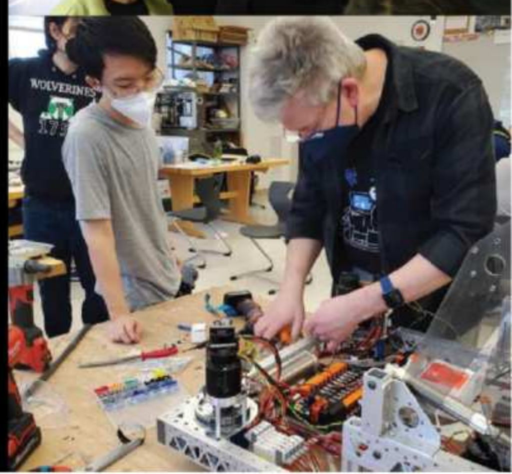
Our Mentors
Our Students
Our Teachers
Our Parents

Our Amazing Sponsors

Westwood Public Schools
Emily Parks - Superintendent
Allison Borchers - Asst. Superintendent
Deb Nolan - Building Use Coordinator
Tom Carey - Director of Facilities
Ron Smith - Assistant Director of Facilities
Amy Davenport - Principal
Aishleen Flanagan - Assistant Principal
Tom Millett - Assistant Principal
Andrew Miller - Science Dept. Chair
WHS Janitorial/Facilities Staff
WHS Library Staff
The Open Alliance

#FRC1757
@WWRobotics1757

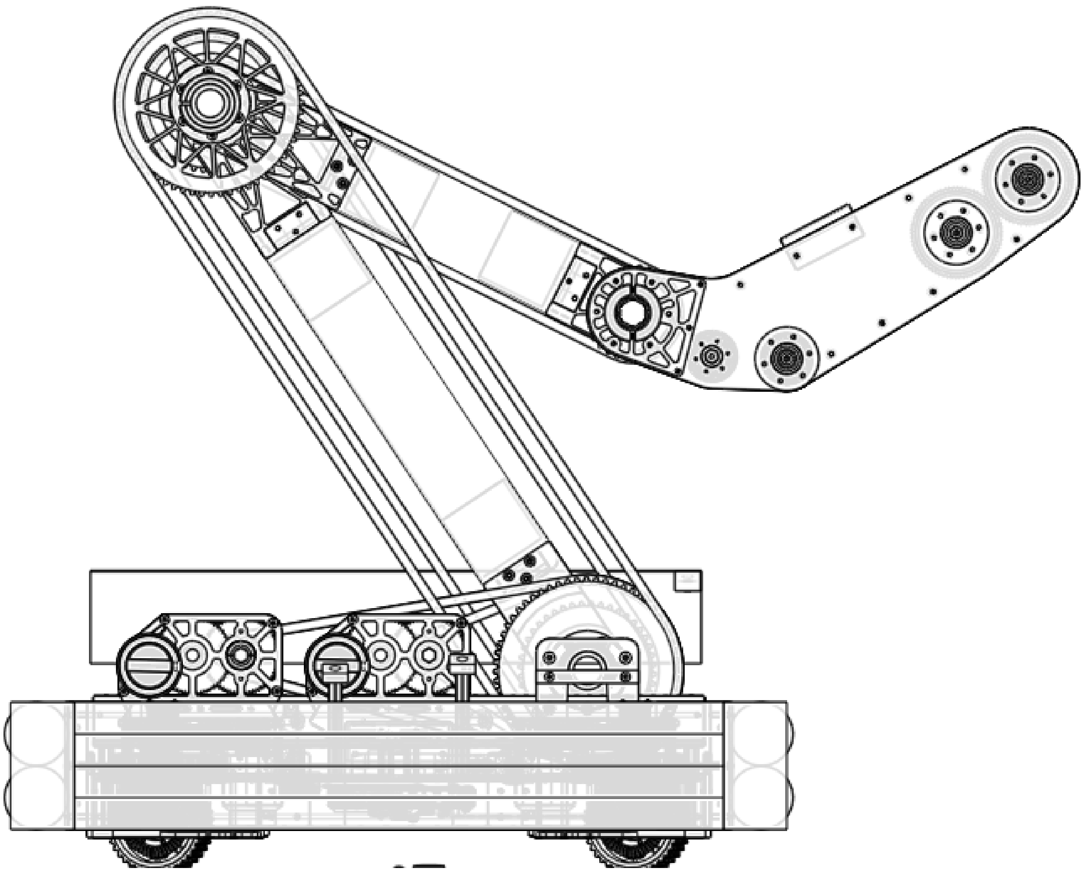




CHARGED UP

PRESENTED BY HAAS Gene Haas Foundation

PROUD MEMBER



FRC WORLD CHAMPIONSHIP EDITION

LUXO

WOLVERINES 1757

2023 TECHNICAL BINDER

FORWARD

Hello, and let us welcome you to FRC Team 1757's 2022-2023 Season. This season has continued the tremendous growth in our robot's design and technical ability that started last year as our team emerged from the hibernation of COVID-19 to become a surprising contender in the New England Region. Continuing to recruit rookie students to supplement our now more veteran team members and Senior Mentors, we have pushed our collective talents to their limits to deliver the competition-worthy robot contained within the pages of this binder.

Our season started in the fall of 2022, introducing a new class of over 10 freshmen, sophomores, and juniors to the world of FRC. We showed off the robot at local town events, built a T-Shirt Cannon to raise school spirit at the prep rally, and hosted weekly technical seminars on everything from the engineering process to CAD, Electronics, Pneumatics, Mechanics, and everything in-between. Over the Summer we got a new OMIO X8 bed router and practiced our CAD and fabrication skills by designing and building an enclosure for the machine. We traveled to Billerica, MA in October to compete in the first-ever New England Robotics Derby. We finished in Second Place, losing in the Finals (The best competitive finish in team history). We piled into our classroom on a cold Saturday morning in January, eagerly anticipating this year's game. 4 CAD models, 8 shared Google Drives, ten weeks, 20 Weekend Build Sessions, 50 Zoom calls, 5799 lines of code, 170 git commits, 19,129 discord messages, and many, many cups of coffee later, we are proud to unveil our robot "LUXO" for the 2023 FRC Season.

Why did we name the robot LUXO? Is it because of the shining lights on its frame that illuminate what game piece we are looking for on the field…no. Is it the bright shining future of the team…no. Is it a reference to solar power and how that ties into the theme of this year's FIRST season…good guess, but no. In truth, we are a bunch of animation nerds, and we thought the robot looked like the lamp in the Pixar Animation title sequence named Luxo. Not every robot name has a deep prophetic meaning…sometimes it's just about the memes.

One very exciting thing about this year is that Team 1757 joined the Open Alliance. We found the Open Alliance teams and their open and timely build season updates so helpful to our team last season that we decided to join so we could help other teams the same way the alliance has already helped us. In addition to frequent updates on our build thread, we also made two appearances on the Open Alliance Show Streamed on twitch. If you want to learn even more about our robot and the design process, beyond what is contained in this manual, please visit our Chief Delphi Build Thread at https://www.chiefdelphi.com/t/frc-1757-wolverines-2022-2023-build-thread/416564

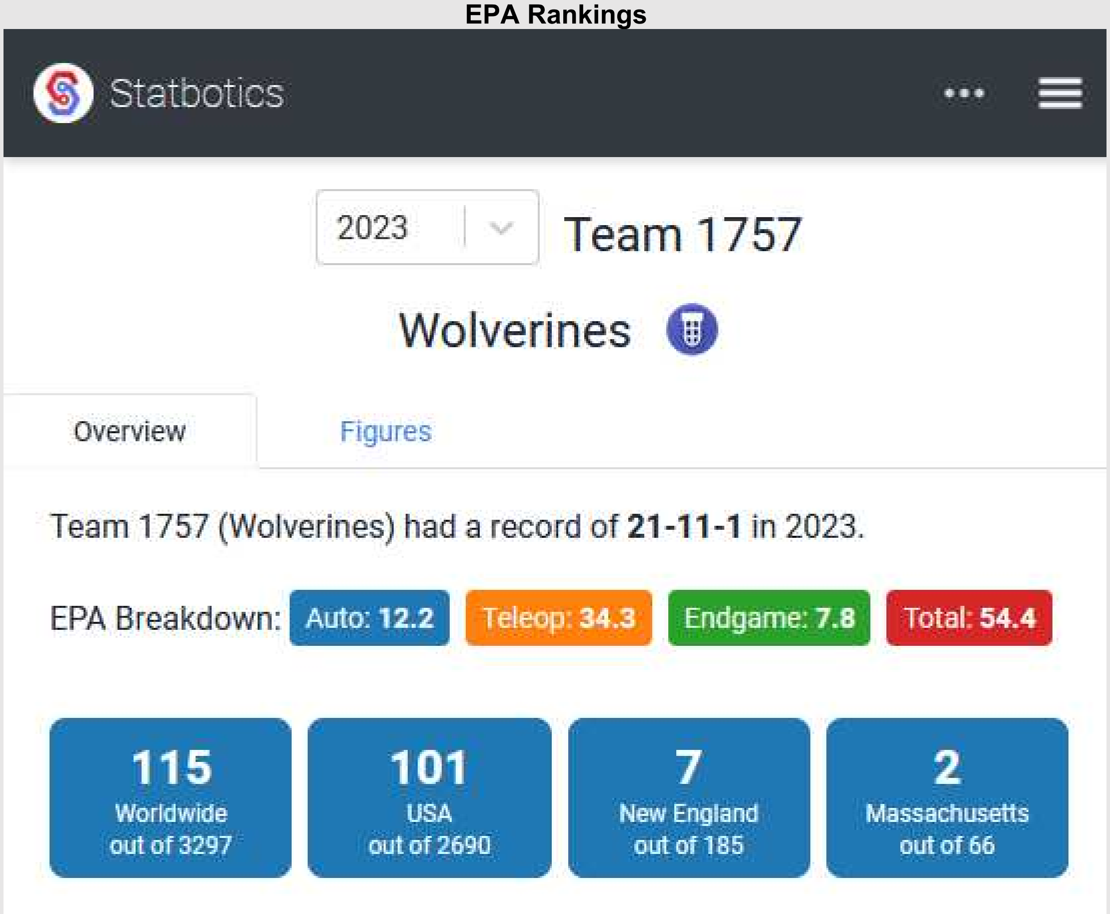
We hope you enjoy this brief look at the design process and technical details that went into this robot, and if you have any questions, look for one of our team members in the stands, in the pits, or on the field. We are always ready to share the knowledge we have gained and share a few hard-learned lessons we learned along the way.

DCMP Update

So it has been a whirlwind of a season so far, after meddeling performance at Greater Boston district we went on win the WPI District Event. Not only were we Alliance captian of the the #2 alliance, we also won the Engineering Inspiration award at WPI. Though out this document you will find various updated information featuring design changes/Repair/modifications that were made during the competition season.

Competitive Record Though District Play:
22-11-1
Greater Boston District - Deans List Semi-Finalist: Sean Tao
WPI District Event – Winners
WPI District Event – Enginnering Inspiration Award Winnerd

EPA Rankings



World Championship Update
We thought our seson couldn't get any better then taking home the teams first ever blue banner at WPI. We were wrong. We came into Wilson Division at New England Championship a solid middle of the pack Contender, however we quicky proved why we were there, our robots consistent and Reliable play led us to take #1 overall at the end of Qualifications, after picking the highest rated offensive bot on the field 176 Aces High, we picked up 1699 Robocats to round out a great alliance. We went undefeated in the Wilson Divison playoffs, taking home another blue banner before taking on the Mier Divison winners for the New England District championship. With the Championship Tied 1-1, we went into a nail-bitting sudden death match where we came out on top.

Please review our OA thread on Chief Delphi for more details.

Competitive Record Though District Championship Play:
39-13-1
Greater Boston District - Deans List Semi-Finalist: Sean Tao
WPI District Event – Winners
WPI District Event – Enginnering Inspiration Award Winnerd
NE Championship – Wilson Divison – Winners
NE Championship – Wilson Division – Excelence in Engineering
New England District Championship - Winners

EPA Record

TABLE OF CONTENTS

GAME ANALYSIS

Every FRC season starts the same way; we gather together as a team, watch the kickoff stream, then hunker down and break down the game in back-to-back 8-hour build sessions. The hope is that by the time we walk out the door on Sunday night, we understand the game and know what we are doing.

After carefully considering the different ways you can score points, we concluded that placing GAME PIECES on the NODES was the most critical ability in this game, with it having the highest potential points available. Without the ability to DOCK and ENGAGE, however, it will be virtually impossible to remain competitive due to the lack of ranking points.

After two days of deliberation, these are the design Requirements we settled on.

DRIVE

- Need to be a Small Bot – The smaller the bot, the easier it is for 3 robots to balance on CHARGE STATION
- Need a low center of gravity
- Need to be able to drive and balance on the CHARGE STATION.
- Preferably autonomous balancing on CHARGE STATION
- Use of vison (April Tags) to provide feedback to the onboard odometry system
- Use of vision to identify and seek out game pieces on the field.

ARM

- Arm needs to be strong and durable
- Use Encoders on the input and output of gearboxes to monitor and minimize backlash.
- Either 2 or 3 Degrees of Freedom Further testing will be needed.
- Needs to score at all 3 levels BOTTOM, MIDDLE and TOP Nodes.

INTAKE

- Quickly acquire GAME PIECES (Touch It – Own It)
- MUST pick up CONES and CUBES from the LOADING STATION
- MUST pick up CUBES and upright CONES from the ground.
- Would like to be able to pick tipped-over CONES from the ground.

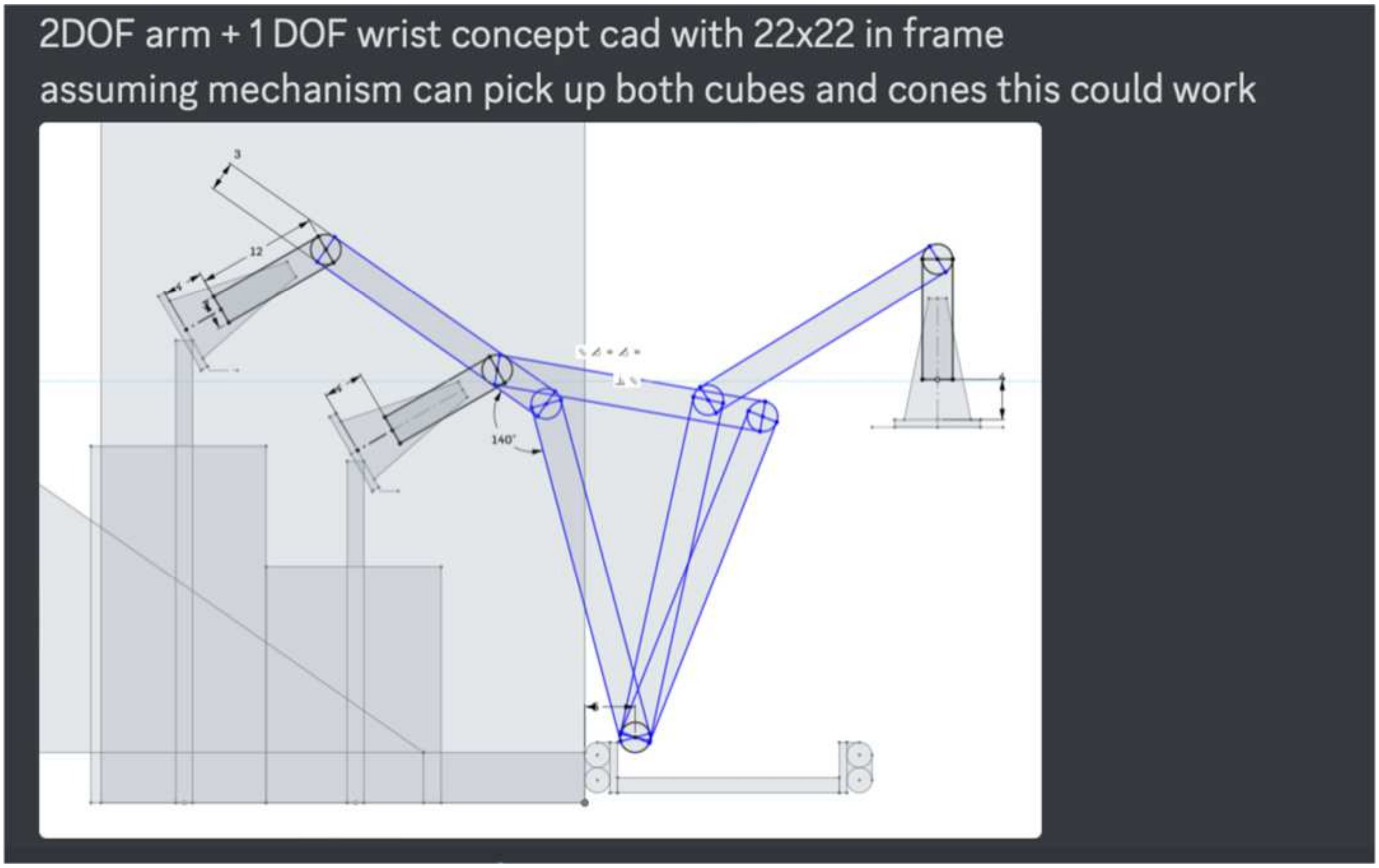
GENERAL DESIGN CONCLUSIONS

Our overall goal for the season was to be a competitive bot in district-level play and qualify for New England Championship. To accomplish this, we need to, at the bare minimum, make it to Elimination at both our district events, hopefully as an Alliance captain or 1st pick.

We approached our design as trying to build a highly reliable jack-of-all-trades bot, focusing on gaining one of the two performance-based ranking points in either match.

Inspired by the cost-effective production strategies of the Hass Formula 1 racing team and our limited team members and design resources, we prefer to use pre-engineered solutions wherever possible to focus our design resources on critical complex components.

IDENTIFYING DESIGN CONSTRAINTS

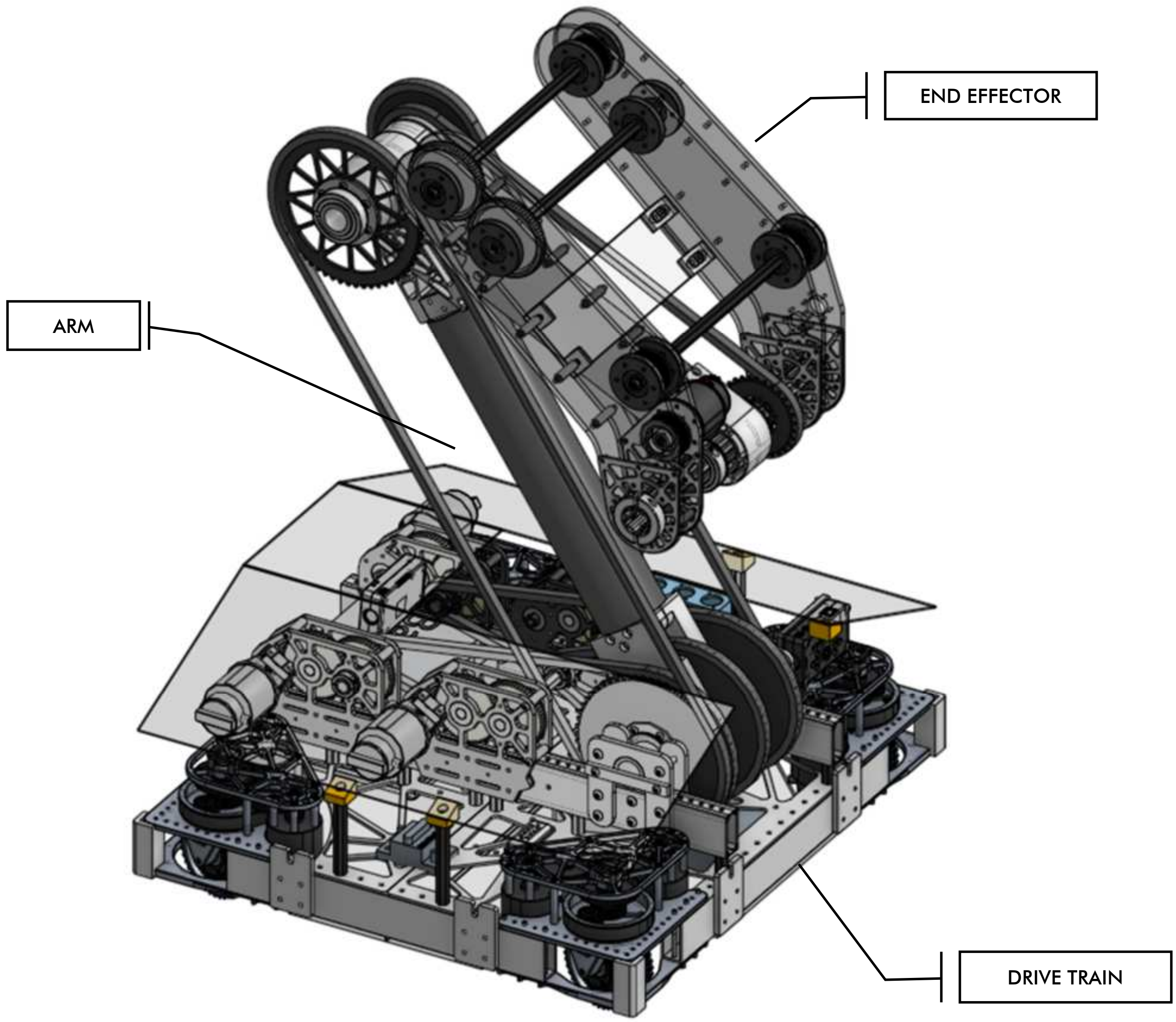


We are thinking about using an arm as a manipulation mechanism. We potentially envision a 2DOF arm + 1 DOF wrist that can pick up both cubes and cones, with a high range of motion on the wrist joint. As we can utilize the bot's movement, we do not need the Arm to move from side to side. An important note is that with an arm the starting configuration poses a good challenge, as it will need to fit inside of the robot's frame before activation. We have found that the shoulder joint only needs to move 90 degrees max, the elbow joint 210 degrees, and the wrist joint somewhere like 270 (at least in the configuration, lots to play with) to achieve all necessary motion.

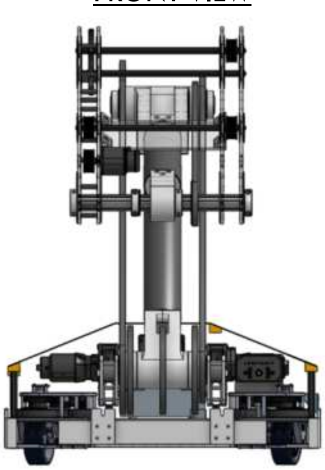
THE 1757 RAPID DEVELOPMENT MODEL

- Clearly Identify the design requirements of the system
- Design and Build a prototype that can be used to test design assumptions and Test
- Use what we learned from testing to develop a final design
- Fabricate final version and intergate into overall robot systems

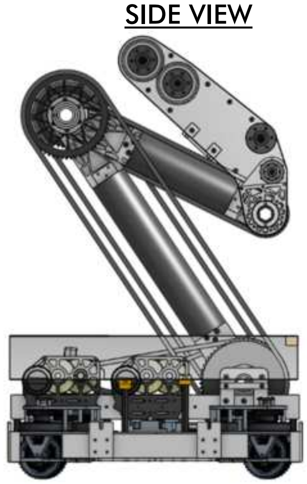
Final Robot Design



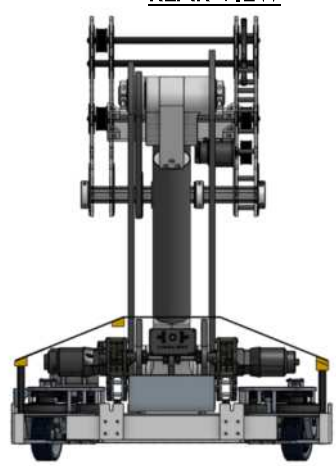
FRONT VIEW

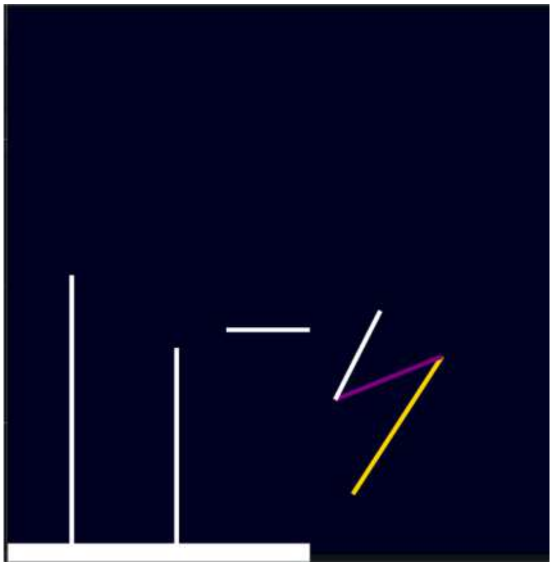


SIDE VIEW

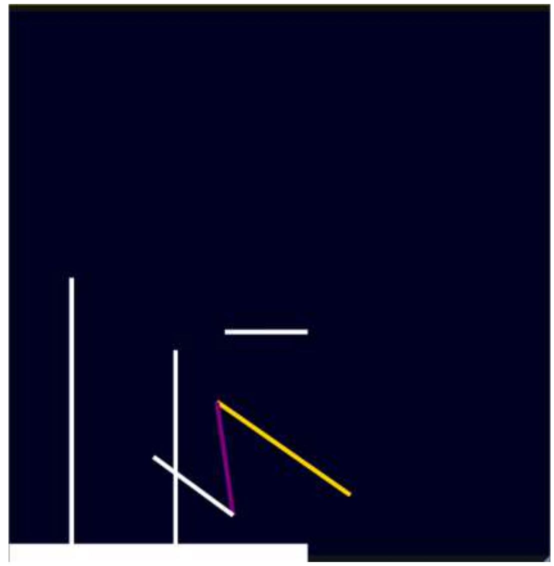


REAR VIEW

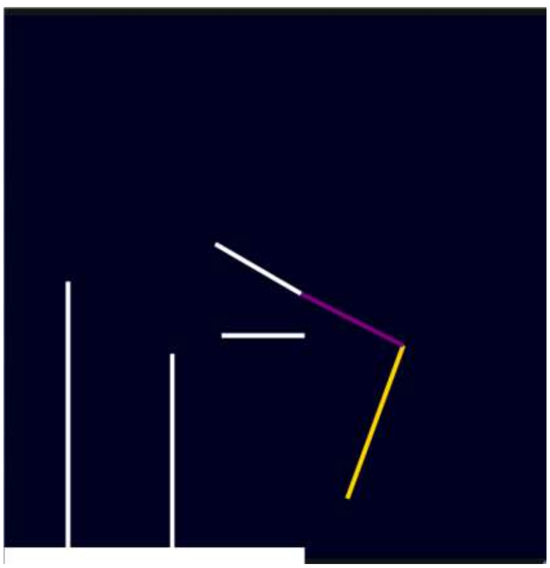




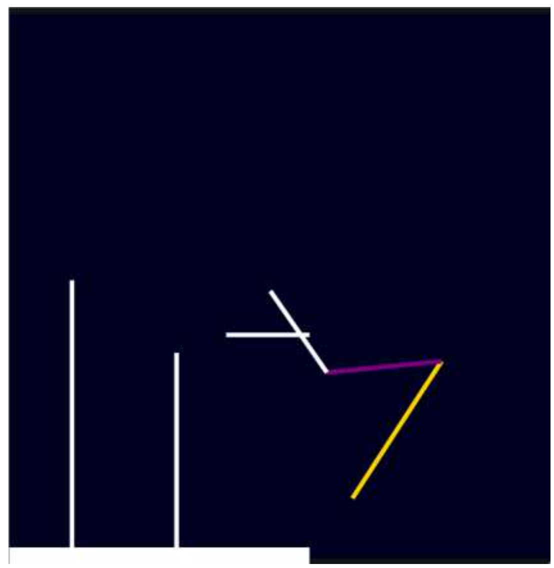
DEFAULT CONFIGURATION



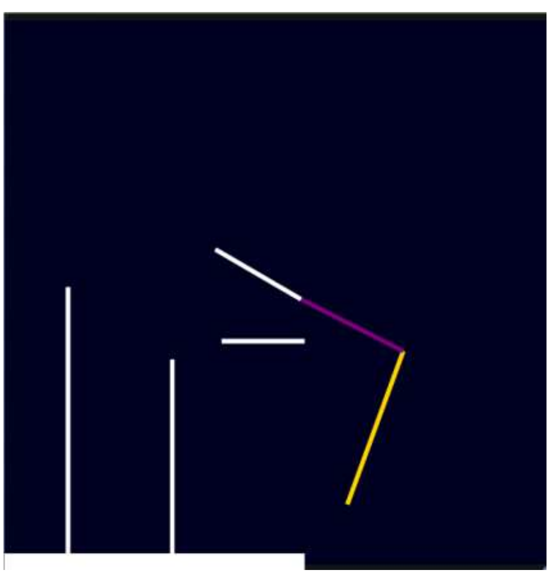
FLOOR PICKUP



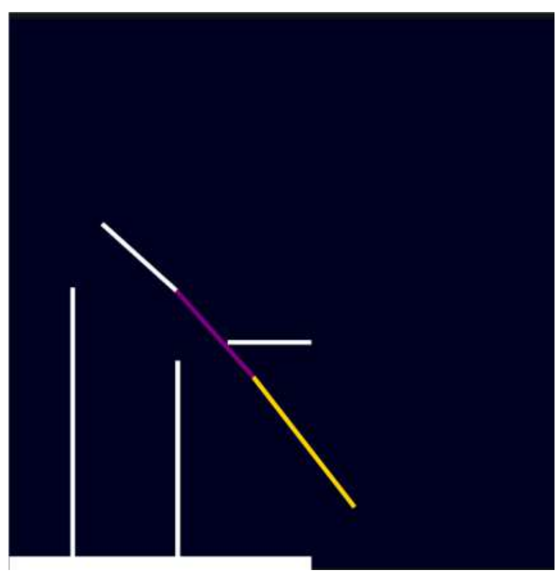
SUBSTATION PICKUP



SCORE – BOTTOM



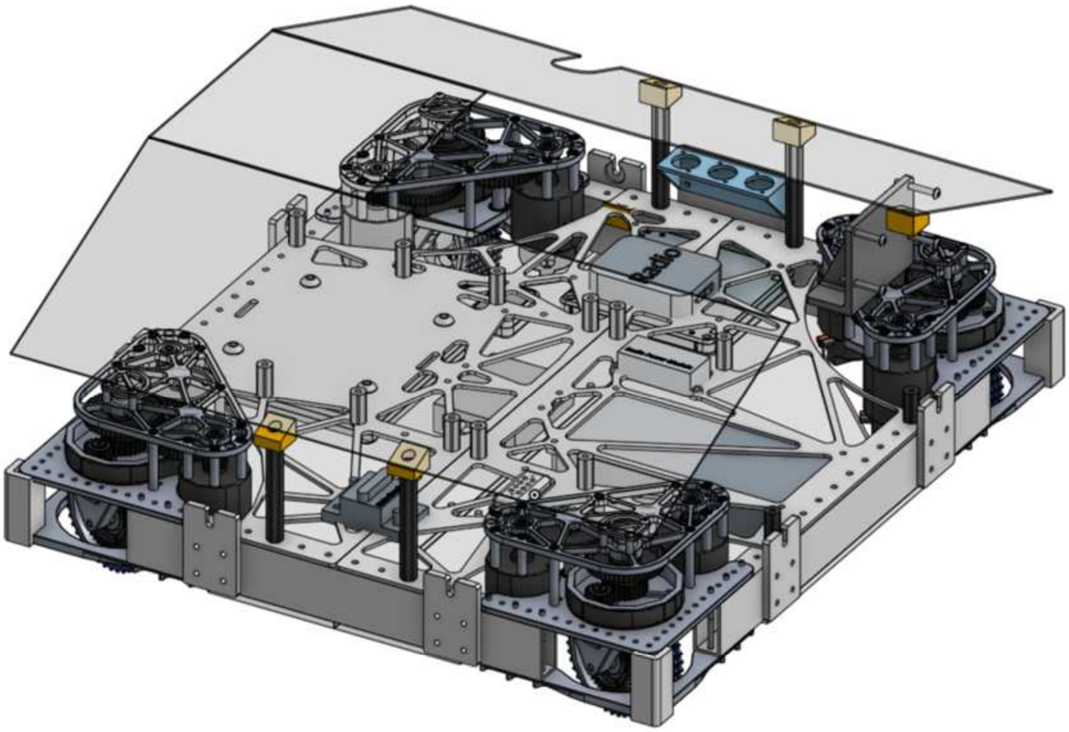
SCORE – MIDDLE

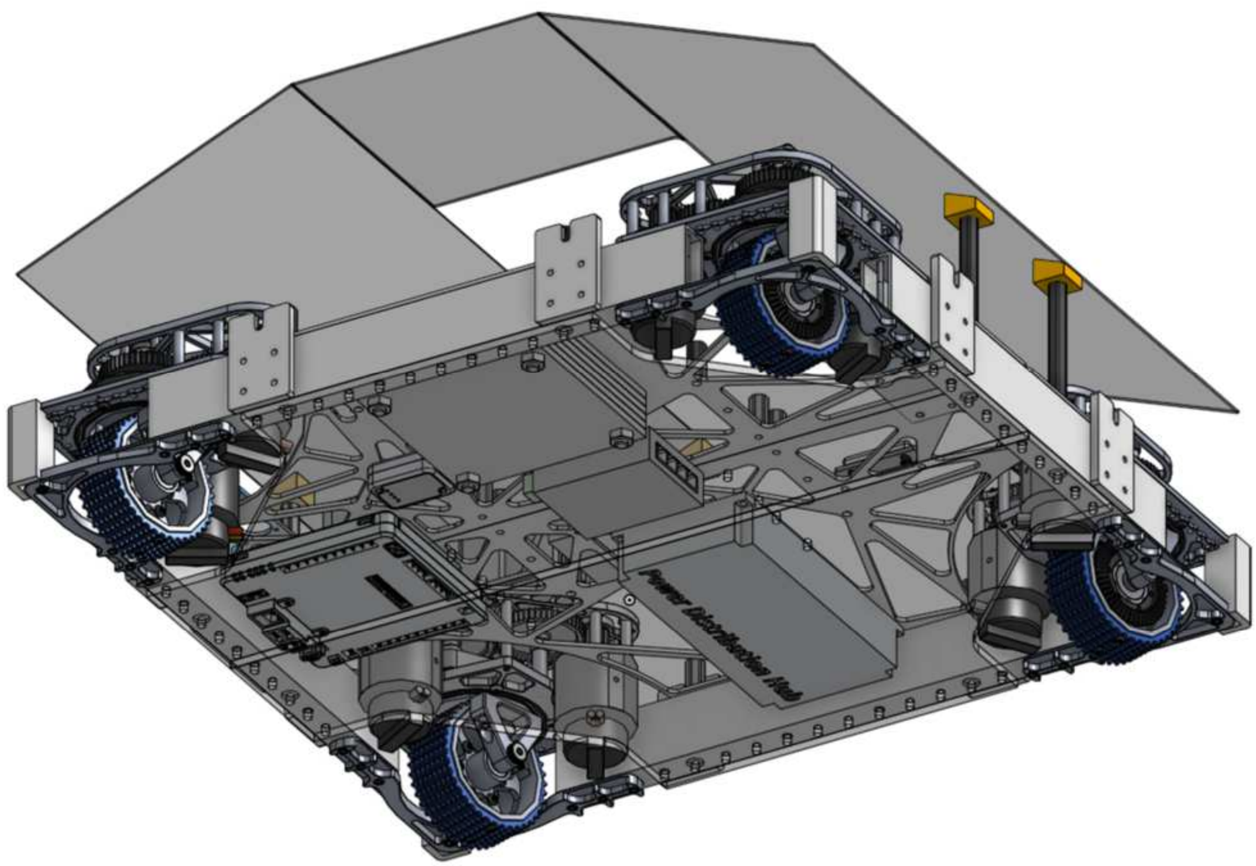


SCORE - TOP

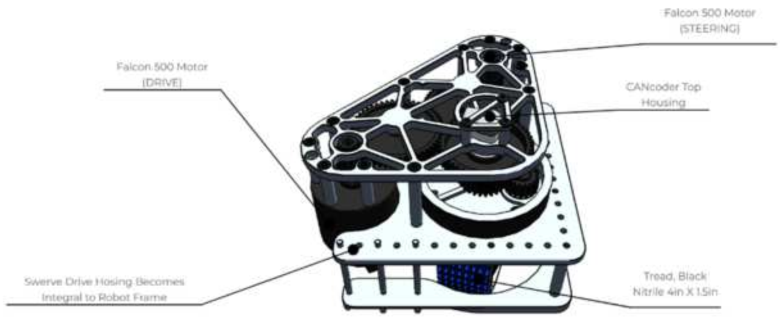
Above – 3D Simulation of Arm Joints in all of its various Arm Configurations

MAJOR SYSTEM #1: DRIVE TRAIN





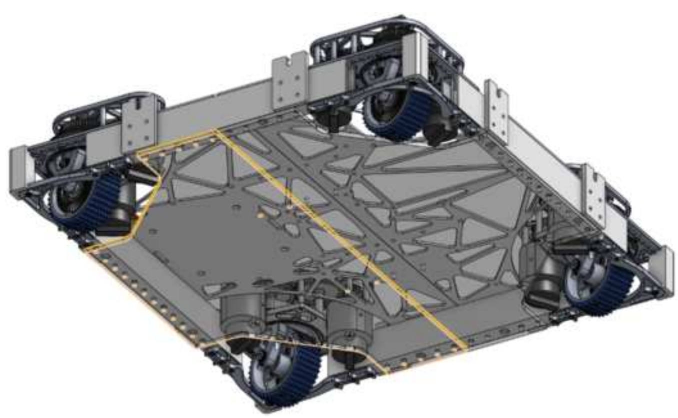
1.1 - SWERVE DRIVE MODULES



Last season was our first competitive season using swerve drive and we could not be happier with the results, because of the equal power between steering and driving there are none of the performance trade offs inherent in other drive systems. We can still run circles around the field when we need to and we can still push another robot across the field when they are in our way. One of our favorite features exclusive to swerve drive is what we call the park feature, by turning all 4 wheels to a 45° angle relative to the corners of the robot the robot effectively parks itself in place and wont move, another robot can push against us all match long and we wont move. Last year we used Swerve Drive Specialties Mk4 units, and this season we upgraded to the newly released Mk4i units. This revised design points the motors downward into the bot instead of mounting above the module. This allowed us to eliminate ≈2" of vertical space in our robot between the drive frame and the major systems.

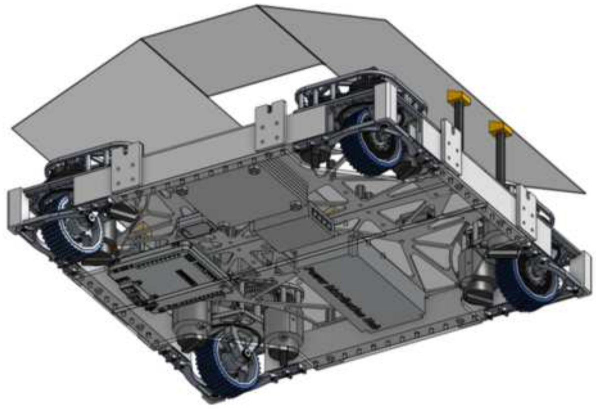
SDS MK4i Swerve Modules	
Powerplant	Falcon 500
Gearbox Configuration	L2
Overall Gearbox Ratio	6.75 : 1
Unadjusted Free Speed	16.3 ft/sec

1.2 - ROBOT FRAME



Left – CAD – Isometric Top View of robot frame
Right – CAD – Isometric Bottom View of Robot frame

1.3 - ELECTRONICS SUBSYSTEM



One thing we struggled with on our 2022 robot was how inaccessible most electrical components were. On our 2022 robot, all the electronics sat in a belly pan at the base of the bot, and the only way to access most of the components required you to remove the majority of the Robot Systems. Inspired by another team's design from 2022, we decided to hang all of our electronic components upside down and face the ground. Now to access the electronics, we tip the bot on its side, remove the ¼" protective polycarbonate plate, and you have full easy access to all the electronics components. (Credit to Team 125 for the Idea, they have the thanks of a grateful drive team and Pit Crew)

ELECTRONICS SYSTEM MAJOR COMPONENTS

- (1 ea) National Instruments - RoboRio 2
- (1 ea) REV Robotics - Power Distribution Hub
- (1 ea) Navex 2 – RoboRio MXP expansion Board
- (1 ea) CTRE CANivore
- (1 ea) CTRE CANdel
- (1 ea) BrainBoxes – SW-015 5 Port Gigabit Switch
- (1 ea) Generic Passive POE Injector
- (1 ea) Limelight 2 Camera
- (12 ea) Falcon 500 Motors
- (1 ea) REV Robotics Sparkmax brushless motor controller
- (1 ea) REV Robotics NEO 550 Brushless DC Motor
- (1 ea) Open Mesh Access Point [Insert Model Number]

1.4 - TESTING PORTS

We added a convenient patch panel to the upper side of the robot to allow for quick access to essential data ports when we don't want to access the underslung electronics.

PATCH PANEL SLOT 1 – USB TYPE A

This slot connects to one of the USB Type A ports on the RoboRio. This typically has a USB flash drive plugged in. During a match all the system logs are copied to the USB drive. After a Match, the USB drive can be pulled and opened up in AdvantageScope on the debug machine for post-game analysis. It's our version of a Blackbox on an airplane.

PATCH PANEL SLOT 2 – USB TYPE B

This connects to the USB Type B Port on the RoboRio—a redundant method for tethering the robot for control and debugging at events.

PATCH PANEL SLOT 3 – RJ45 CONNECTOR

This connects to the Ethernet Switch Via CAT5e for network access. Used for tethered connections to the bot during testing. Ethernet tethering is preferred, but we have encountered software reliability issues in the past.

DCMP Update

At the Revere District event we ran into serious problems tethering to the robot via ethernet and via USB B. we traced the ethernet tethering problem to a problem with the network configuration issue on the driver station laptop. We were unable to determine a definite cause of the USB-B connection issue, but, we think it most likely to to poor quality of the 90° usb connector used on the robot. From that point on we connect a USB-B cable directly into the port on thr

At the same time we realized we needed a button to manually put the arm motors in coast mode for serviceability when the bot is not connected to the driver station. Since we are no longer using the USB-B testing port we replaced it with a momentary push button switch.

1.5 – CAMERA/VISON SYSTEMS

LIMELIGHT 2 – CAMERA

We are utilizing a Limelight 2 Camera for a variety of tasks on the robot mostly devoted to sensor fusion and automation of systems using computer vision. The Limelight's field of vision (FOV) is essentially parallel to the floor and at the height of the April Tags.

Please refer to the Software section of this document for more information on how we use the limelight and April Tags to improve the Onboard odometry of the robot.

1.6 – COUNTERWEIGHT

Not originally intended as part of the design, upon testing of the robot with the Arm fully extended in the scoring position, we realized that robot was prone to falling forward. To resolve this, we looked to add ballast to the bot. First we thought of lead but didn't want to deal with the potential health risks of improperly encapsulated lead. We investigated tungsten; however, a review of the current price of tungsten plate (≈$40/kg) quickly ruled it out as a potential candidate. We settled on 6" x6" x1/4" steel plates mounted directly under the robot battery. After testing with different #'s of the plate, we decided on 6 Plates with a total weight of ≈25 lbs. Now the bot is highly stable even when the Arm is fully extended.

1.7 – PROTECTIVE COVERS

We added protective covers that Slope away from the central arm structure down to the bumpers. Not only do these plates provide a valuable location to display all of our great sponsors they serve to prevent errant game pieces from getting stuck inside the robot during a match.

DCMP Update

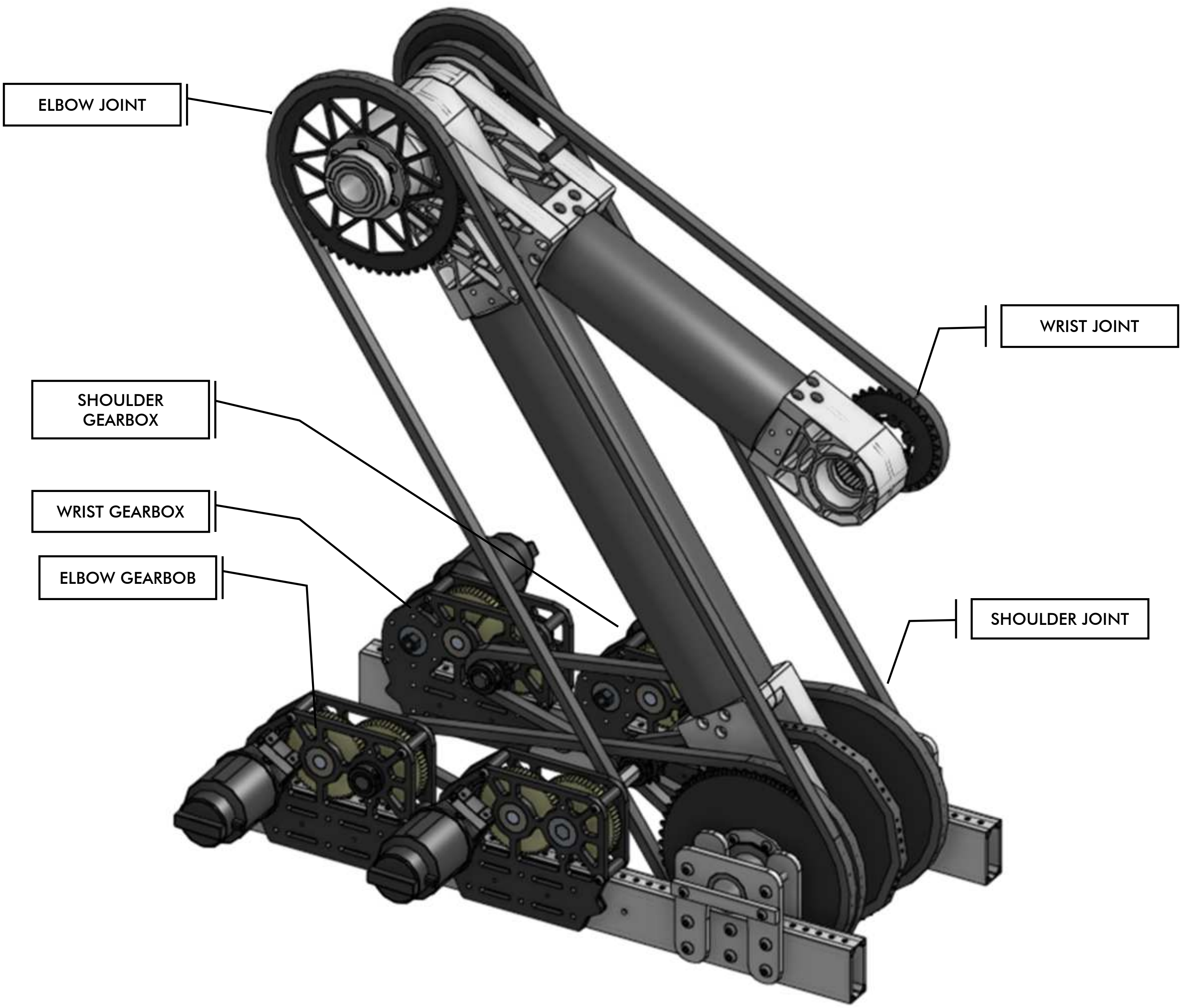
Originally the protective covers were only hled on with 3M™ Dual Lock™ SJ3560, this material is nice because it very strong but eaisily removable. Durring qualifying matches in Revere however, these panels kept falling off and dragging around the fieild. The Dual Lock strips were reinforced with zip ties and these held through all of playoffs in revere, and all of qualifications at WPI. Then in Playoff Matches we shered off 3 of the metal standoffs holding up the protective covers. We made quick repairs to keep going however prior to DCMP we will be swaping out all the ½ thunderhex standoffs with 1" 80/20 extrusion with hardened bolts for strength.

1.8 – GAMEPIECE INDICATORS

One of the hardest things to do in a match is how to signal between the driver station and the human player what game piece you want them to load into the robot. People use Hand Signals, Colored pieces of paper or guess. We wanted to take the guesswork out of the equation, so we mounted 2 LED Strip lights along the top of the protective covers. The driver controls what color these strips are so he can communicate to the human player which game piece to feed to the bot – Yellow for Cones and Purple for Cubes.

Left – Robot Displaying "I Want a Cone"
Right – Robot displaying "I Want a Cube"

MAJOR SYSTEM #2: ARM



2.1- MOTORS & GEARBOXES

To keep the robot's center of gravity low and keep the Arm as simple as possible we decided to locate all of the heavy motors and gearboxes at the base of the superstructure.

GEARBOX # 1 – SHOULDER GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into Custom Gearbox based on WCP Gearbox design.

GEARBOX # 2 – ELBOW GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into COTS WCP Single Speed Gearbox.

GEARBOX # 2 – WRIST GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into COTS WCP Single Speed Gearbox.

2.2 - CHAIN DRIVE

Using a combination of dead and live axels we transfer the power of the gearboxes up though the Arm to power each of the individual joints. For Reliability and durability, we chose to use #35 roller chain rated for 11,000 lbs of force.

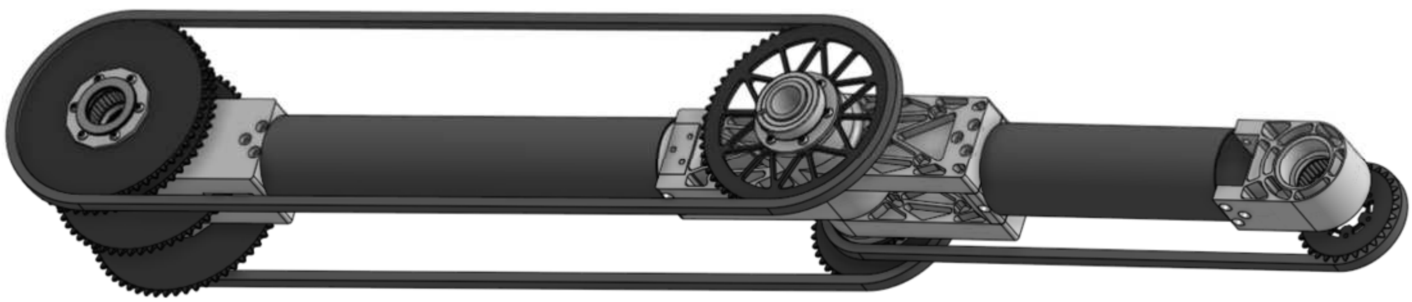
Below is a summary of the different chain runs on the Arm

CHAIN DRIVE 1 – SHOULDER
Shoulder Gearbox Output 12t Sprocket → 60t sprocket on Shoulder (Dead Axel)

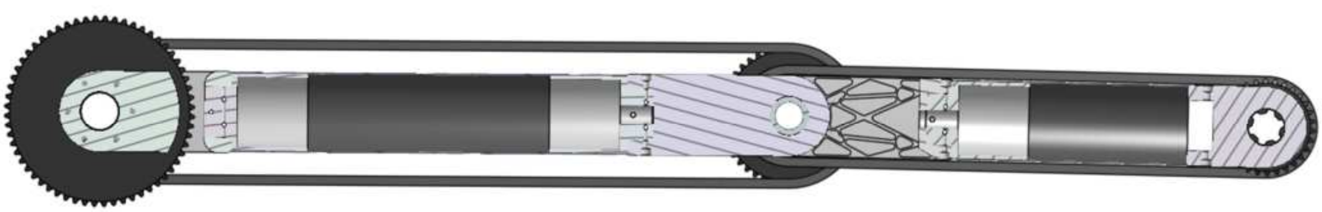
CHAIN DRIVE 2 – ELBOW
Elbow Gearbox Output Shaft 12t Sprocket → 60t Sprocket on Shoulder (Live Axel) → 70t Sprocket on Elbow (Dead Axel)

CHAIN DRIVE 3 – WRIST
Wrist Gearbox Output Shaft 12t Sprocket → 60t Sprocket on Shoulder (Live Axel) → 60t Sprocket on Elbow (Live Axel) → 40t Sprocket on Elbow (Live Axel) → 32t Sprocket on the Wrist (Dead Axel)

2.3 - ARM STRUCTURE



Above - 3-D View of Outstretched Arm



Above - Section View Though Center of Carbon Fiber Arm

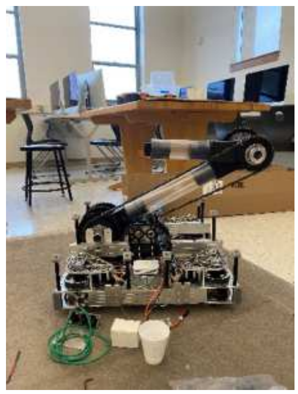
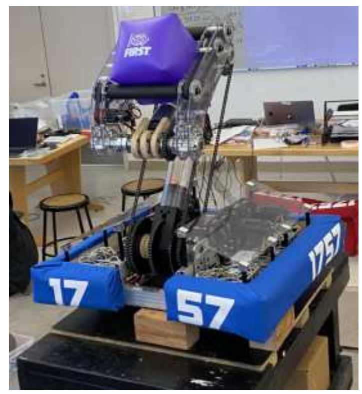
CARBON FIBER ARMS

We chose to use carbon fiber tubes as the main structure of the Arm due to its strength and lightweight, the more weight we could save on the Arm the lower we could push the robot Center of Gravity. Carbon Fiber tubes are a stock McMaster item 3" Ø. Carbon Fiber is Epoxy bonded to 3" hollow aluminum plugs bolted to the aluminum joints.


Left – Final carbon fiber arm links after final glue up
Right – Final aluminum plugs used in the ends of the carbon fiber tubes

3D Printed and Polycarbonate Prototypes

Because we knew the carbon fiber and machined aluminum would take time and money to manufacture, we heavily used 3D-printing to make prototypes of the Arm and test and confirm critical geometry before placing final fabrication orders. These prototypes are two fragile to be used on a competition bot but worked well for their intended purposes. We learned very Important lessons about where the concentrations of forces were along the axels and what parts needed reinforcement.

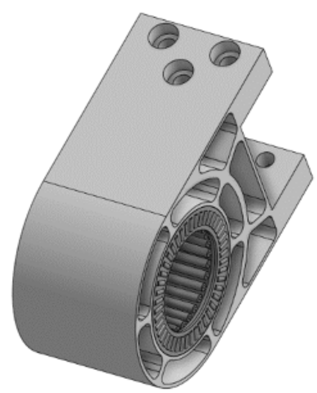




Left – 3D printed Prototype of the wrist joint, printed on a FormLabs 2 SLA Printer
Center – Polycabonate Prototype arm Mounted on bot for the First Time
Right – Fully Assembled "Alpha" Robot build

2.4 – JOINT STRUCTURE

CAD - Shoulder Joint

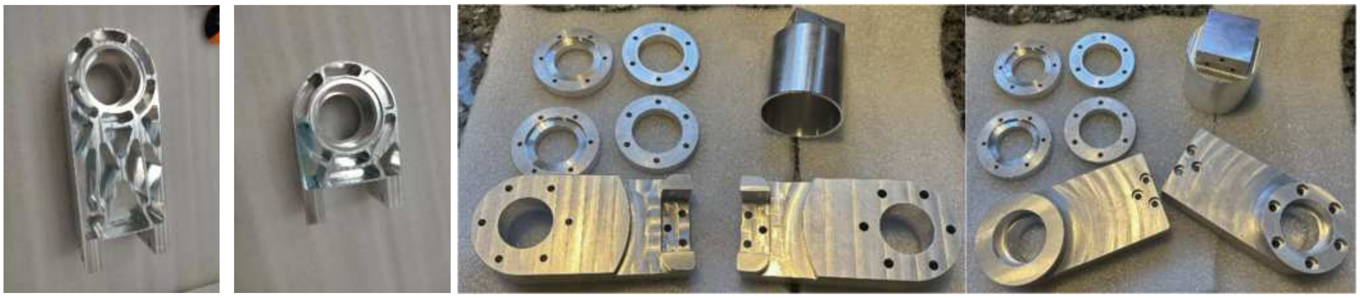
CAD - Elbow Joint

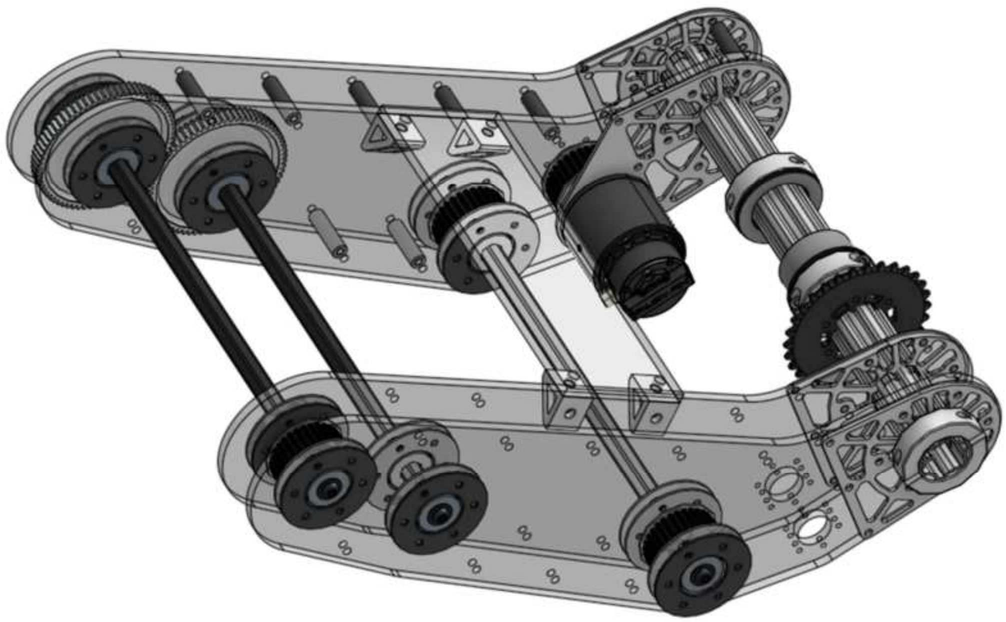


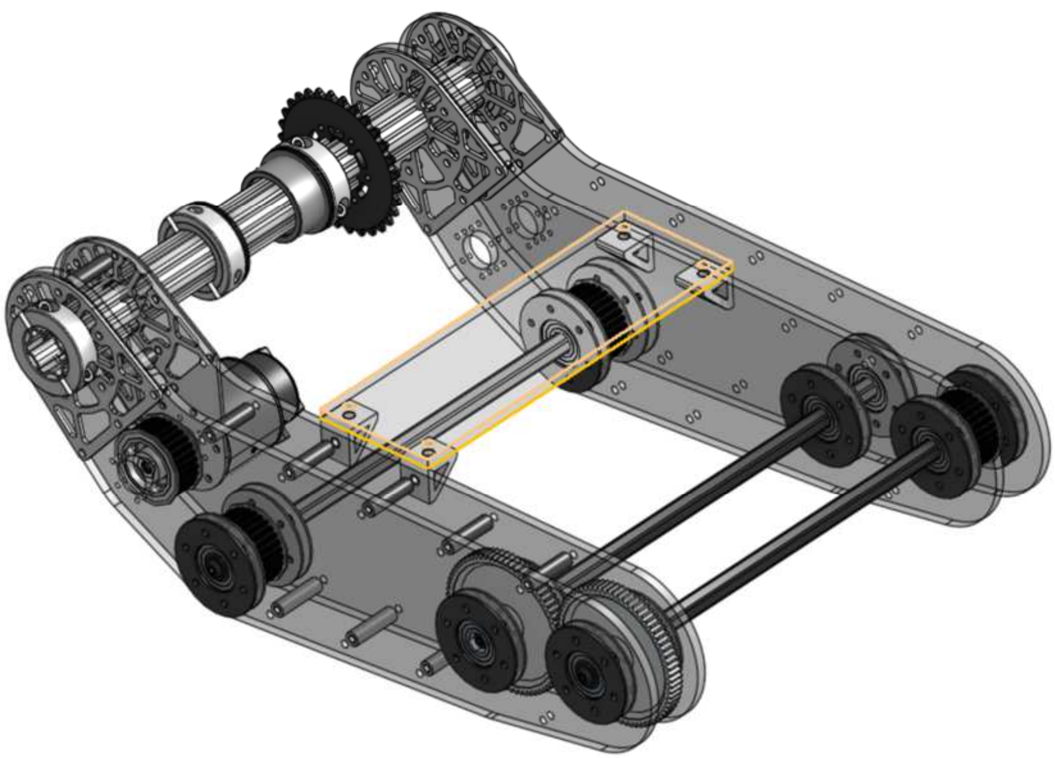
CAD - Wrist Joint

Needle and Thrust Bearings

Used in All three joints to allow for smooth rotary motion in each joint.

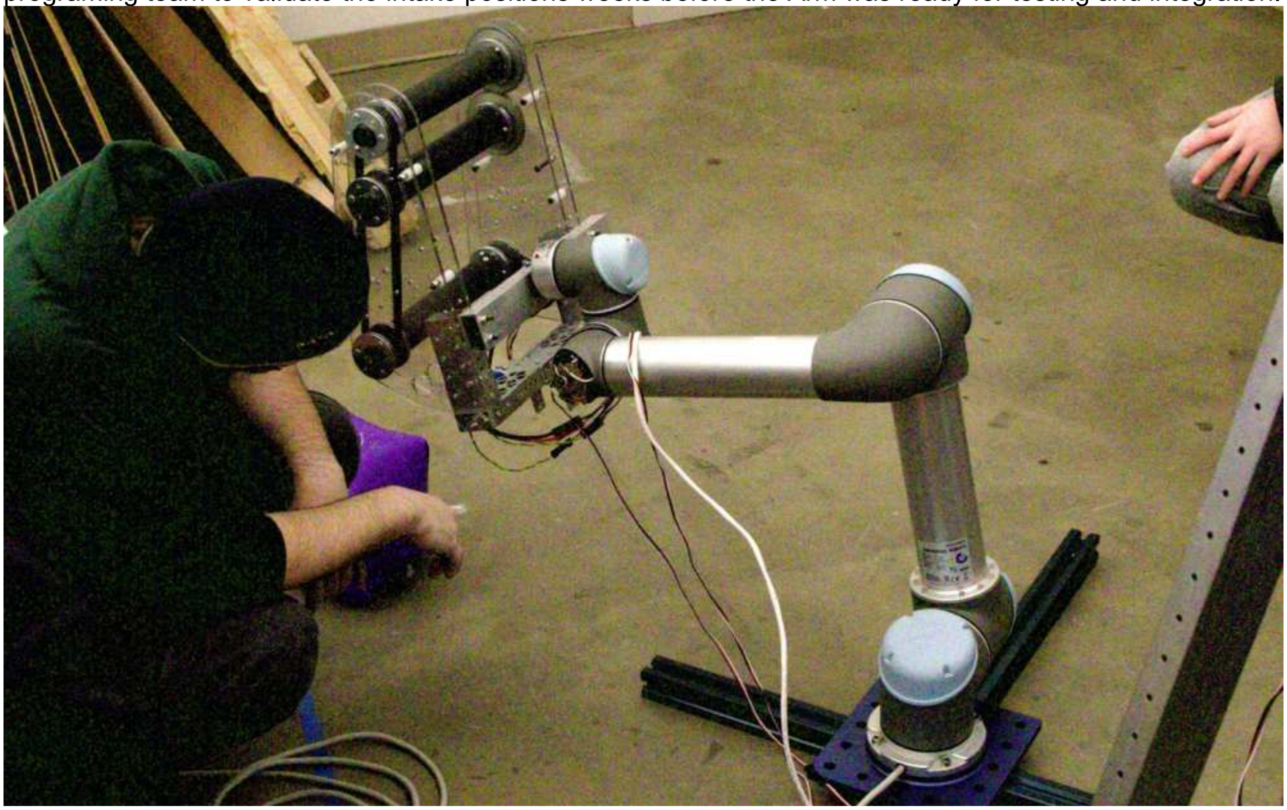

Photos - Final Machined Parts





Intake Prototyping – Fun With Industrial Robots

We had a lot of fun prototyping this mechanism, since it was the first major mechanism that we finished we had lots of time to put it trough its paces. One of our mentors has access to a Universal Robots UR3 robot and brought into our lab during one of our weekend build sessions (see images below). This actually proved to be essential because it allowed our programing team to validate the intake positions weeks before the Arm was ready for testing and integration.



3.1 – ROLLERS

The rollers we are using are Vex VersaHub Rollers with ¼" neoprene tubing stretched to cover them, they are very grippy and hold the cube verry securely. We started with the dimensions of the everybody roller for our prototype then made some modifications before settling on final separation differences. The neoprene tubing is undersized for the OD of the polycarbonate roller. We learned a fun trick to clamp off one end of the neoprene tube and inflate it with an air compressor to stretch it over the polycarbonate tube, when the air is released, it makes a perfect friction fit between the Neoprene and the polycarbonate. We have had no detectable slippage after weeks of testing with the rollers.

DCMP Update

After 33 competitive matches one thing is clear, we have problems picking CUBES up off the floor and in order to maintain our competitive edge at DCMP we know we need to be able to get CUBES up off the floor. We thing the majority of the problem is related to how narroe our end effector is, the original design was only 1” wider than the width of the CUBE. To aliviate this issue we are planning to widen the end effector by 3”.

3.2 – MOTORS & GEARBOXES

The intake is powered by a REV Robotics NEO550 Brushless motor into a REV Robotics Ultraplanetary gearbox. The motors small size is nice however because we mounted the Sparkmax Motor controller on the intake as well there is no significant weight savings compared to using a Falcon 500 with an integrated Talon SRX. We may end up swaping this out for simplicity sake in the future.

REV Ultraplanetrary	
Powerplant	NEO550
Gearbox Configuration	4:1, 5:1
Overall Gearbox Ratio	20:1

SOFTWARE

SOFTWARE: OUR DEVELOPMENT ENVIRONMENT

WPILib

The perineal stalwart, we still rely on core elements of WPIlib for robot communications and debugging. WPILib's new Logging features have greatly enhanced our Debugging capabilities

RobotPy

We have found that students have a lot easier time learning python then they do Java or C++ so with the growing support for RobotPy we migrated our Codebase from Java to Python in 2020. As of this March we are an official contributor to the RobotPy project

GitHub

Without Github our level of remote work and collaboration just wouldn't be possible.

SOFTWARE: NEW AND UPDATED TOOLS THIS YEAR

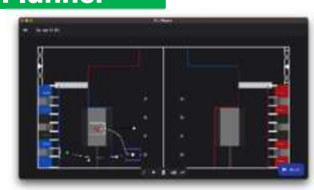
AdvantageScope

WE LOVE ADVANTAGE SCOPE! Not only does it log *everything* but does it in away that is intuitive and easy to review. No more searching though 10000 lines of log files to find the one piece of information we need. Huge thanks to team 6328 for building such a great tool.

PhotonVision

We are using PhotoVision as our native development framework for Computer vision due to its growing wide support inside the FRC community. It does not include native support for RobotPy however so as an offseason project our lead programmer wrote a custom wrapper for PhotonVivion so it can work inside our RobotPy environment

PathPlanner



Last year we used PathWeaver, but we were disappointed in the lack of native support and increased complexity in the development stack so starting with the off season we transitioned all of our Autonomous path planning to PathPlanner. We had much fewer issues with this system.

SOFTWARE: DRIVE

Taking off of last year, the drivetrain codebase has stayed the same. We are running field oriented drive with robot relative rotation to allow for quick maneuverability. A button to align to the nearest 90 degree angle was added to help with driver alignment. This state slightly reduces the speed and snaps the angle of the robot in order to have perfect alignment to the double substation, single substation, and grid every time. For our automated balance sequence, we work in robot relative space on the robot relative gyro.

A Brief Tangent - Absolute Relative Drive

Last year our lead programmer had a new idea for drive control, an absolute relative drive. The common swerve drive control method was to have a field relative translation for the bot, and a robot relative rotation. What this meant is a left input on the rotation axis would result in the robot rotating to the left at a constant speed. A translation action was not affected by rotation but instead was in "field relative" space. The difference of absolute drive is that the rotation is also field relative. A left input on the rotation stick will yield the robot turning to face left. This year we expect this type of robot control to be very important for drivers when they have to be able to turn to specific positions for collection and scoring on swerve drives. You can see this in action in any one of our videos from last year. Having fixed controlled rotation will allow for precise driver input and less fiddling with controls when cycle time is very important.

The drivers have also experimented with alternate driving methods on swerve to get used to interesting control schemes such as a curvature drive, standard tank drive, standard field relative drive, and full robot relative drive.

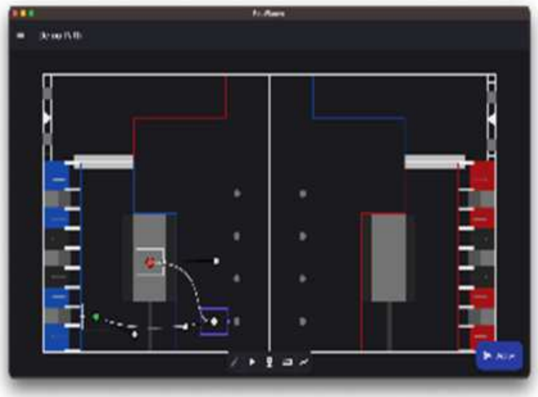
SOFTWARE: INTAKE

The intake is using a state machine in order to regulate its expected behavior. There are 2 enumerable values: one for the gamepiece intended direction (intake, outtake, hold) and one for the desired gamepiece type (cube, cone). The transition between each state is dictated by a user input to any given category. If no input is given, the system holds its position and keeps the desired gamepiece remembered. The state value of the desired gamepiece is displayed to the driver and to the human player through pulsing leds of the respective color.

SOFTWARE: ARM

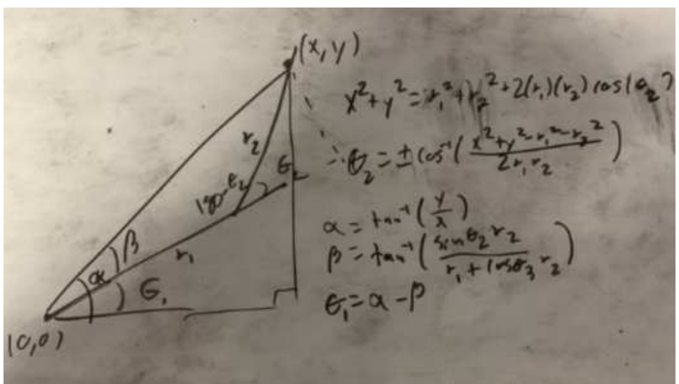
A triple jointed arm is no easy feat in order to program smoothly. From cad, states are given about the end effector's desired position and rotation relative to the floor. From there we use inverse kinematics to determine each per-joint relative rotation at any given position. A cartesian control on the wrist joint's position is added using a trapezoidal PID profile to lay out a path for the Arm to follow. For the wrist it has another trapezoidal PID profile controller. When going state per state on the Arm's motion, we check for if the relative angle goes over software end limits in order to prevent running the Arm into itself. These are done in joint relative space. Since the Arm is controlled from the base through chain and sprockets there is a virtual 4 bar created in which the rotation of any given joint is given relative to the ground. These are converted into motor space and passed onto each motor where they have a position PID controller onboard. For simulating the system we are using mechanism2d to view the expected values of the Arm and be able to run through positions. This simulation first approach has allowed minimal revision and a solid foundational codebase that is mostly complete before the bot is finished. Furthermore the position of the Arm is logged in 3d with advantagescope based on the position it believes the bot is in. Logging was also a priority for a complex system and therefore we log the instructed position and actual position per joint and each end position in cartesian space.

SOFTWARE: AUTONOMUS



We experimented in the offseason with pathplanner and use it extensively for our autonomous. Each necessery command is placed into a potential pool of events for pathplanner to fire. At the beginning a sequence determined solely in pathplanner is fired. Going off of last year we had a large time gap in order to make slight changes so instead for this year we are using the waypoint system and opting to have any given autonomous contained solely in pathplanner. This has increased our ability to construct autos and tweak any given aspect as needed. For the library itself of parsing, the lack of first class robotpy support meant we had the freedom to reimplement autonomous however we pleased based on the path. We follow a changing trajectory and the swerve drive using onboard odometry and a weighted vision estimate determines its bot position relative to the global field and follow through it between each section.

SOFTWARE: SIMULATION



armsubsystem.py

```
    def setEndEffectorPosition(self, pose: Pose2d):

        twoLinkPosition = Translation2d(
            pose.X() - constants.kArmwristLength * pose.rotation().cos(),
            pose.Y() - constants.kArmwristLength * pose.rotation().sin(),
        )

        endAngle = math.acos(
            twoLinkPosition.X() * twoLinkPosition.X()
            + twoLinkPosition.Y() * twoLinkPosition.Y()
            - constants.kArmtopLength * constants.kArmtopLength
            - constants.kArmbottomLength
            * constants.kArmbottomLength
            / (2 * constants.kArmtopLength * constants.kArmbottomLength)
        )

        startAngle = math.atan2(twoLinkPosition.Y(), twoLinkPosition.X()) -
math.atan2(
            math.sin(endAngle) * constants.kArmtopLength,
            constants.kArmbottomLength + math.cos(endAngle) *
constants.kArmtopLength,
        )
        wristAngle = pose.rotation().radians() - startAngle - endAngle

        bottomArmEncoderPulses = (
            startAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kBottomArmGearRatio
        )
        topArmEncoderPulses = (
            endAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kTopArmGearRatio
        )
        wristArmEncoderPulses = (
            wristAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kWristPivotArmGearRatio
        )

        self.topArm.set(Falcon.ControlMode.Position, topArmEncoderPulses)
        self.bottomArm.set(Falcon.ControlMode.Position,
bottomArmEncoderPulses)
        self.wristArm.set(Falcon.ControlMode.Position, wristArmEncoderPulses)
```

Due to our team's resources, virtual simulation is a huge part of our ability to quickly and reliably construct the bot's codebase. Some key examples of simulation are a wrapped onto a simulated falcon motor. Given our team's extensive use of falcons on the robot, a wrapper that provides simulation support allows for the programming team to iterate much easier and creates a cleaner codebase. Each falcon is logging the values of the motor % and the encoder position, as well as an override value to allow the user to manually in simulation change the value for sensor readings. Entire robot configuration is done on a single call and the getting of velocity, position, and percent and the setting of velocity, position, and percent are easy to access functions to allow interfacing with the motors more accessible than the CTRE library. Given this robot also has a NEO550, the simulation system was adopted to have a similar interface for ease of replacement from a falcon to a motor on the intake. We geometrically derived the inverse kinematics for 3 links with a fixed Pose endjoint. Each of these poses actually allows for two configurations of the proximal 2 arm joints (they can simply be mirrored over the line created from the wrist joint to the shoulder joint, however by forcing the sign on the elbow joint they can all be consistent.

SOFTWARE: VISION

NOW WITH APRILTAGS AND PHOTONLIB

We have a vision system complete with sensor fusion for complete robot localization. Last year,we worked with our first complete vision system as a team that resulted in significantly enhanced system performance, and using apriltags will be very important to account for combined sensor error as well as for being able to reliably use sensor data for automated alignment to various points on the field such as the double substation and the grid.

THE HOW

Photonvision generates camera-relative 3d transforms of each apriltag. Since the position of the camera is known and the position of the apriltag is known, the position of the robot can be determined from a single apriltag datapoint. These transforms are fed into a RobotPoseEstimator in order to create a sense of where the robot could be at a given time, this is combined with the gyro and wheel encoder information to get an accurate sense of where the robot is on the field at any given time. This is used in other subsystems when needed, as well as results being logged to AdavantageScope through the usage of each known pose and ghost poseseepaste

GOING FURTHER

We plan on using this odometry data to have automated alignment in complete robot space for important precision actions such as placement of gamepieces on the grid and collection of those gamepieces. Autonomous will also use this data. Perhaps an automatic engagement on the charge station by using the rotation gained from the apriltags will be possible. Overall having a sense of where the robot is on the field is beneficial to aid in other systems.

ENGINEERING TEAM

WHS Faculty Advisors

James Looney & Raul Madera

Team Captain

Charley Marsland*

Team Business Lead

Sean Tao

Team Technical Lead

Luke Maxwell

Senior Mentors

Dwight Meglan | **Chris Aloisio°** | **Steve Harrington°**

Mentors

Anthony Gelsomini | **Manny Barros°** | **Sean Lendrum°** | **Mark Holthouse** *Mentor Emeritus* | **Amber Maxwell**

STUDENT MEMBERS

Safety Captain

Jacob Kaplan

Graphic Design Lead

Ivan Cai

CAD Lead

Landon Bayer

John Santasuoso*	**David Girard***	**Declan MacDonald**	**Liam McWeeney**
Alex Theofilou	**Claire Peng**	**Andrew Gong**	**Rachel Qu**
Baili Jiang	**Nolan folmar**	**Jeffery Pan**	**Melissa Yang**
Divya Siddi	**Vinny Milinazzo**	**David Confoey**	**Adrianna Cirillo**
Elizabeth Lowney	**Jacob Liu**	**Anthony Yang**	**Constantina F.**
Luke Li	**Owen Monahan**	**Erik Curlli**	**Kevin Bai**
Henry Marsland	**Nina Pappas**	**Rachel Lee**	**Hanya Yang**
Sophia Patrick	**Lauren Buza**	**Kaylee Phu**	**Elias Mukdissi**

Westwood Robotics, Inc
Board Of Directors

Steve Harrington° *President* | **Catherine Tseng** *Treasurer* | **Manny Barros°** | **Cheten Gopal** *Clerk* | **Jack Tseng**

**Denotes High School Seniors*

°Denotes FRC Alumni

SPONSORS

Heartlander Surgical	Westwood High School	Medtronic	Act Fasteners
Gene Haas Foundation	TE Connectivity	General Dynamics	IRA Motor Group
Bayer Familt	Capital Group	Yamaguchi Family	Plymouth Rock Assurance
Deadham Savings	Maxwell Family	Schneider Electric	Marsland Family
Porier Apliances	Franklin Templeton Investments	Jake and Joes Resturant	McGolf
Roche Bros		Capers Catering	

VENDORS

Andymark	West Coast Products	Vex Pro	Swerve Drive Specialties
Discord	Google Work	Onshape	Robo Promos
McMaster-Carr	The Home Depot	Cross the Road Electronics	MS Visual Studio
Python	GitHub	RobotPy	Java

1757

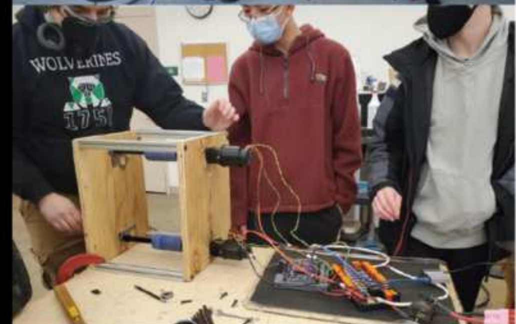
Special thanks to everyone who makes this team possible Including

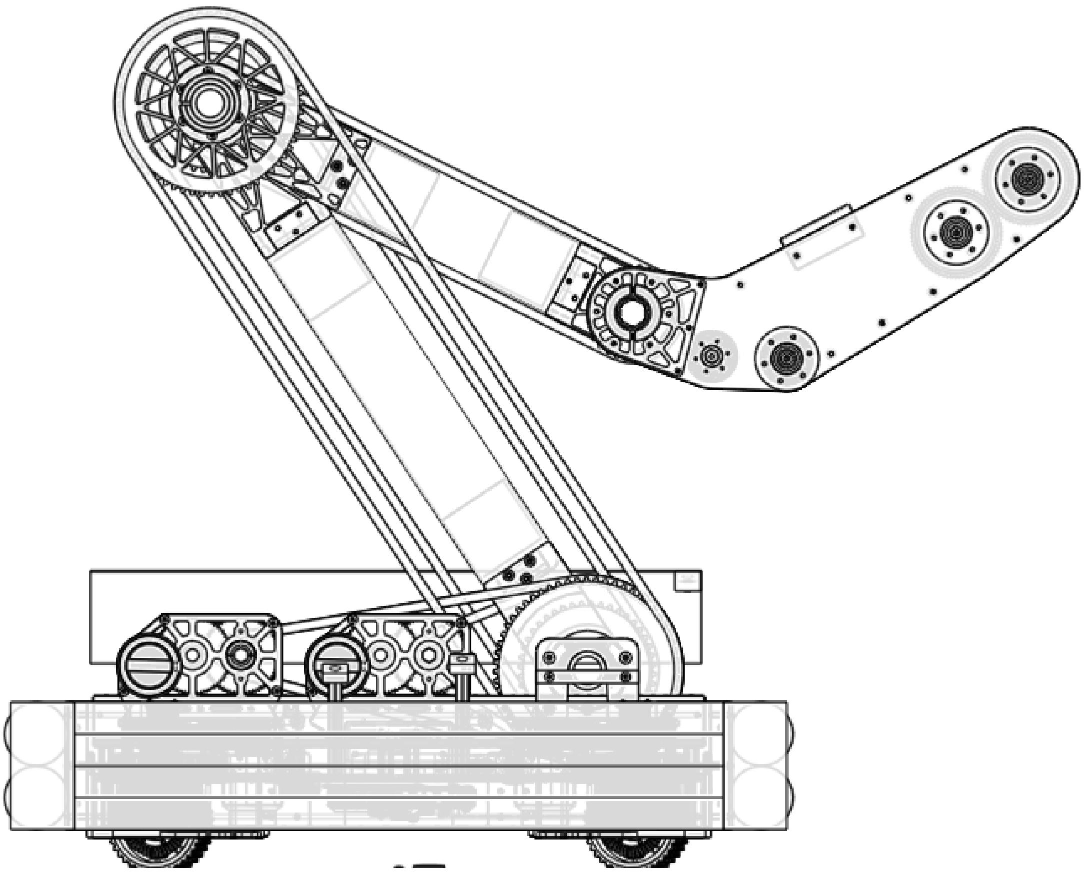
Our Mentors
Our Students
Our Teachers
Our Parents

Our Amazing Sponsors

Westwood Public Schools
Emily Parks - Superintendent
Allison Borchers - Asst. Superintendent
Deb Nolan - Building Use Coordinator
Tom Carey - Director of Facilities
Ron Smith - Assistant Director of Facilities
Amy Davenport - Principal
Aishleen Flanagan - Assistant Principal
Tom Millett - Assistant Principal
Andrew Miller - Science Dept. Chair
WHS Janitorial/Facilities Staff
WHS Library Staff
The Open Alliance

#FRC1757
@WWRobotics1757





FRC WORLD CHAMPIONSHIP EDITION

LUXO

WOLVERINES

1757

2023 TECHNICAL BINDER

FORWARD

Hello, and let us welcome you to FRC Team 1757's 2022-2023 Season. This season has continued the tremendous growth in our robot's design and technical ability that started last year as our team emerged from the hibernation of COVID-19 to become a surprising contender in the New England Region. Continuing to recruit rookie students to supplement our now more veteran team members and Senior Mentors, we have pushed our collective talents to their limits to deliver the competition-worthy robot contained within the pages of this binder.

Our season started in the fall of 2022, introducing a new class of over 10 freshmen, sophomores, and juniors to the world of FRC. We showed off the robot at local town events, built a T-Shirt Cannon to raise school spirit at the prep rally, and hosted weekly technical seminars on everything from the engineering process to CAD, Electronics, Pneumatics, Mechanics, and everything in-between. Over the Summer we got a new OMIO X8 bed router and practiced our CAD and fabrication skills by designing and building an enclosure for the machine. We traveled to Billerica, MA in October to compete in the first-ever New England Robotics Derby. We finished in Second Place, losing in the Finals (The best competitive finish in team history). We piled into our classroom on a cold Saturday morning in January, eagerly anticipating this year's game. 4 CAD models, 8 shared Google Drives, ten weeks, 20 Weekend Build Sessions, 50 Zoom calls, 5799 lines of code, 170 git commits, 19,129 discord messages, and many, many cups of coffee later, we are proud to unveil our robot "LUXO" for the 2023 FRC Season.

Why did we name the robot LUXO? Is it because of the shining lights on its frame that illuminate what game piece we are looking for on the field…no. Is it the bright shining future of the team…no. Is it a reference to solar power and how that ties into the theme of this year's FIRST season…good guess, but no. In truth, we are a bunch of animation nerds, and we thought the robot looked like the lamp in the Pixar Animation title sequence named Luxo. Not every robot name has a deep prophetic meaning…sometimes it's just about the memes.

One very exciting thing about this year is that Team 1757 joined the Open Alliance. We found the Open Alliance teams and their open and timely build season updates so helpful to our team last season that we decided to join so we could help other teams the same way the alliance has already helped us. In addition to frequent updates on our build thread, we also made two appearances on the Open Alliance Show Streamed on twitch. If you want to learn even more about our robot and the design process, beyond what is contained in this manual, please visit our Chief Delphi Build Thread at https://www.chiefdelphi.com/t/frc-1757-wolverines-2022-2023-build-thread/416564

We hope you enjoy this brief look at the design process and technical details that went into this robot, and if you have any questions, look for one of our team members in the stands, in the pits, or on the field. We are always ready to share the knowledge we have gained and share a few hard-learned lessons we learned along the way.

DCMP Update

So it has been a whirlwind of a season so far, after meddeling performance at Greater Boston district we went on win the WPI District Event. Not only were we Alliance captian of the the #2 alliance, we also won the Engineering Inspiration award at WPI. Though out this document you will find various updated information featuring design changes/Repair/modifications that were made during the competition season.

Competitive Record Though District Play:

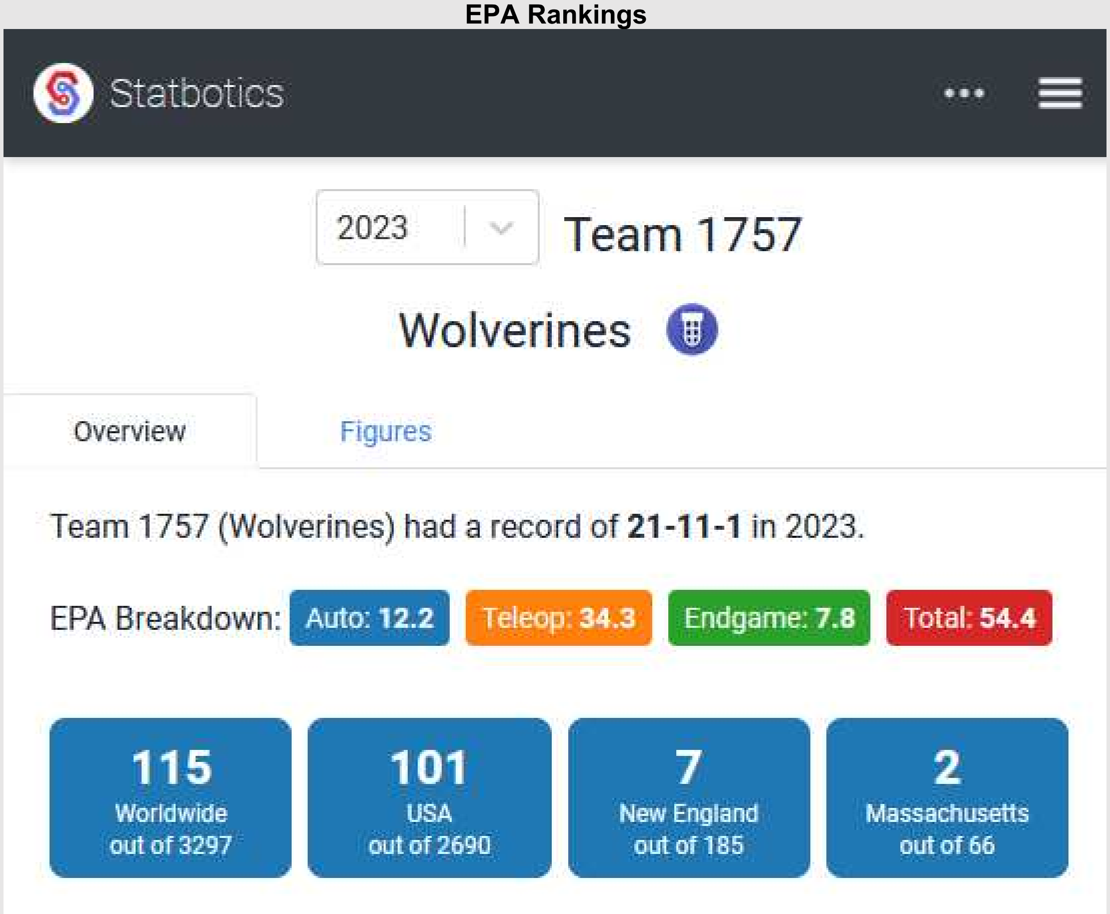
22-11-1

Greater Boston District - Deans List Semi-Finalist: Sean Tao

WPI District Event – Winners

WPI District Event – Enginnering Inspiration Award Winnerd

EPA Rankings



World Championship Update
We thought our seson couldn't get any better then taking home the teams first ever blue banner at WPI. We were wrong. We came into Wilson Division at New England Championship a solid middle of the pack Contender, however we quicky proved why we were there, our robots consistent and Reliable play led us to take #1 overall at the end of Qualifications, after picking the highest rated offensive bot on the field 176 Aces High, we picked up 1699 Robocats to round out a great alliance. We went undefeated in the Wilson Divison playoffs, taking home another blue banner before taking on the Mier Divison winners for the New England District championship. With the Championship Tied 1-1, we went into a nail-bitting sudden death match where we came out on top.

Please review our OA thread on Chief Delphi for more details.

Competitive Record Though District Championship Play:
39-13-1
Greater Boston District - Deans List Semi-Finalist: Sean Tao
WPI District Event – Winners
WPI District Event – Enginnering Inspiration Award Winnerd
NE Championship – Wilson Divison – Winners
NE Championship – Wilson Division – Excelence in Engineering
New England District Championship - Winners

EPA Record

TABLE OF CONTENTS

GAME ANALYSIS

Every FRC season starts the same way; we gather together as a team, watch the kickoff stream, then hunker down and break down the game in back-to-back 8-hour build sessions. The hope is that by the time we walk out the door on Sunday night, we understand the game and know what we are doing.

After carefully considering the different ways you can score points, we concluded that placing GAME PIECES on the NODES was the most critical ability in this game, with it having the highest potential points available. Without the ability to DOCK and ENGAGE, however, it will be virtually impossible to remain competitive due to the lack of ranking points.

After two days of deliberation, these are the design Requirements we settled on.

DRIVE

- Need to be a Small Bot – The smaller the bot, the easier it is for 3 robots to balance on CHARGE STATION
- Need a low center of gravity
- Need to be able to drive and balance on the CHARGE STATION.
- Preferably autonomous balancing on CHARGE STATION
- Use of vison (April Tags) to provide feedback to the onboard odometry system
- Use of vision to identify and seek out game pieces on the field.

ARM

- Arm needs to be strong and durable
- Use Encoders on the input and output of gearboxes to monitor and minimize backlash.
- Either 2 or 3 Degrees of Freedom Further testing will be needed.
- Needs to score at all 3 levels BOTTOM, MIDDLE and TOP Nodes.

INTAKE

- Quickly acquire GAME PIECES (Touch It – Own It)
- MUST pick up CONES and CUBES from the LOADING STATION
- MUST pick up CUBES and upright CONES from the ground.
- Would like to be able to pick tipped-over CONES from the ground.

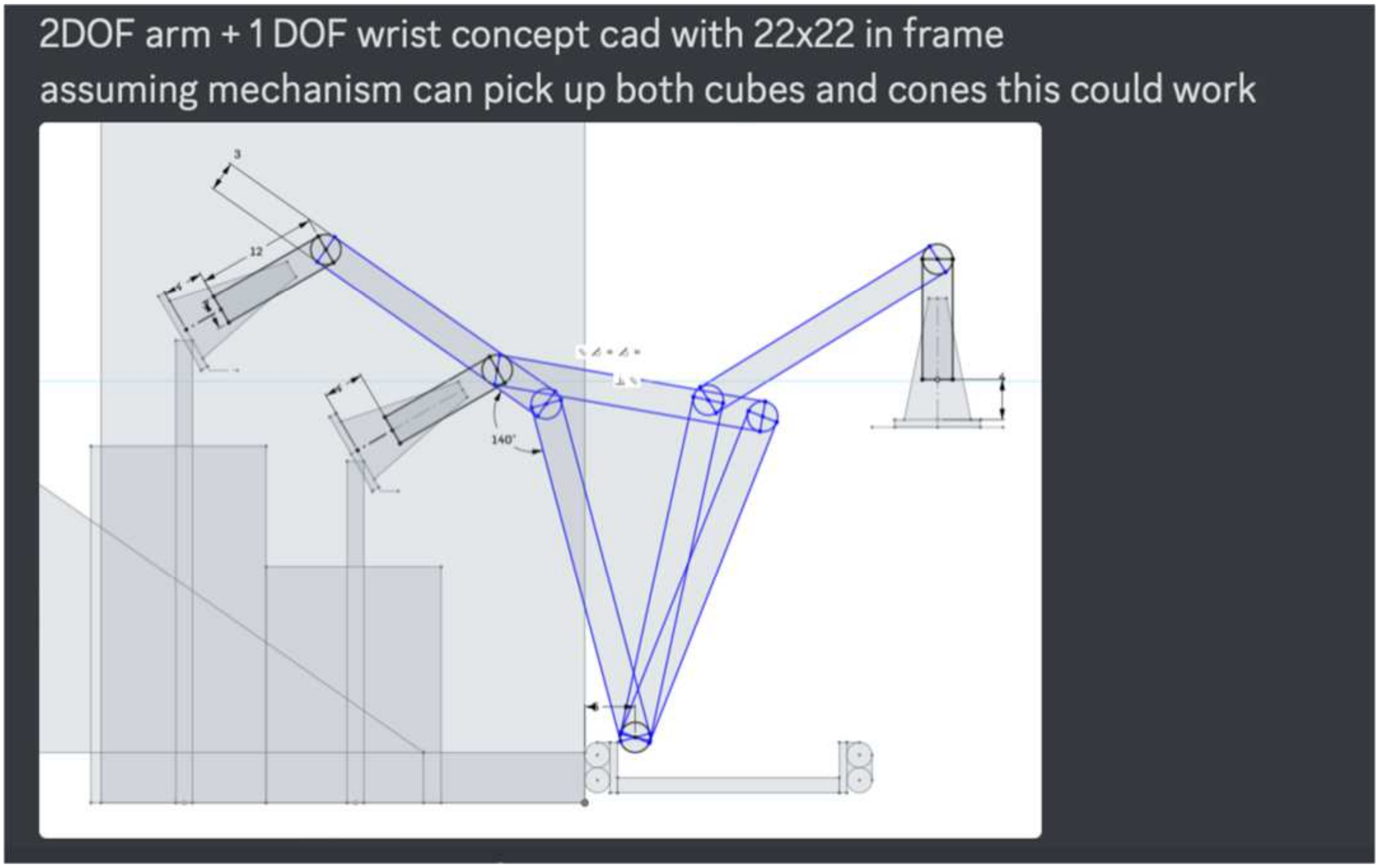
GENERAL DESIGN CONCLUSIONS

Our overall goal for the season was to be a competitive bot in district-level play and qualify for New England Championship. To accomplish this, we need to, at the bare minimum, make it to Elimination at both our district events, hopefully as an Alliance captain or 1st pick.

We approached our design as trying to build a highly reliable jack-of-all-trades bot, focusing on gaining one of the two performance-based ranking points in either match.

Inspired by the cost-effective production strategies of the Hass Formula 1 racing team and our limited team members and design resources, we prefer to use pre-engineered solutions wherever possible to focus our design resources on critical complex components.

IDENTIFYING DESIGN CONSTRAINTS

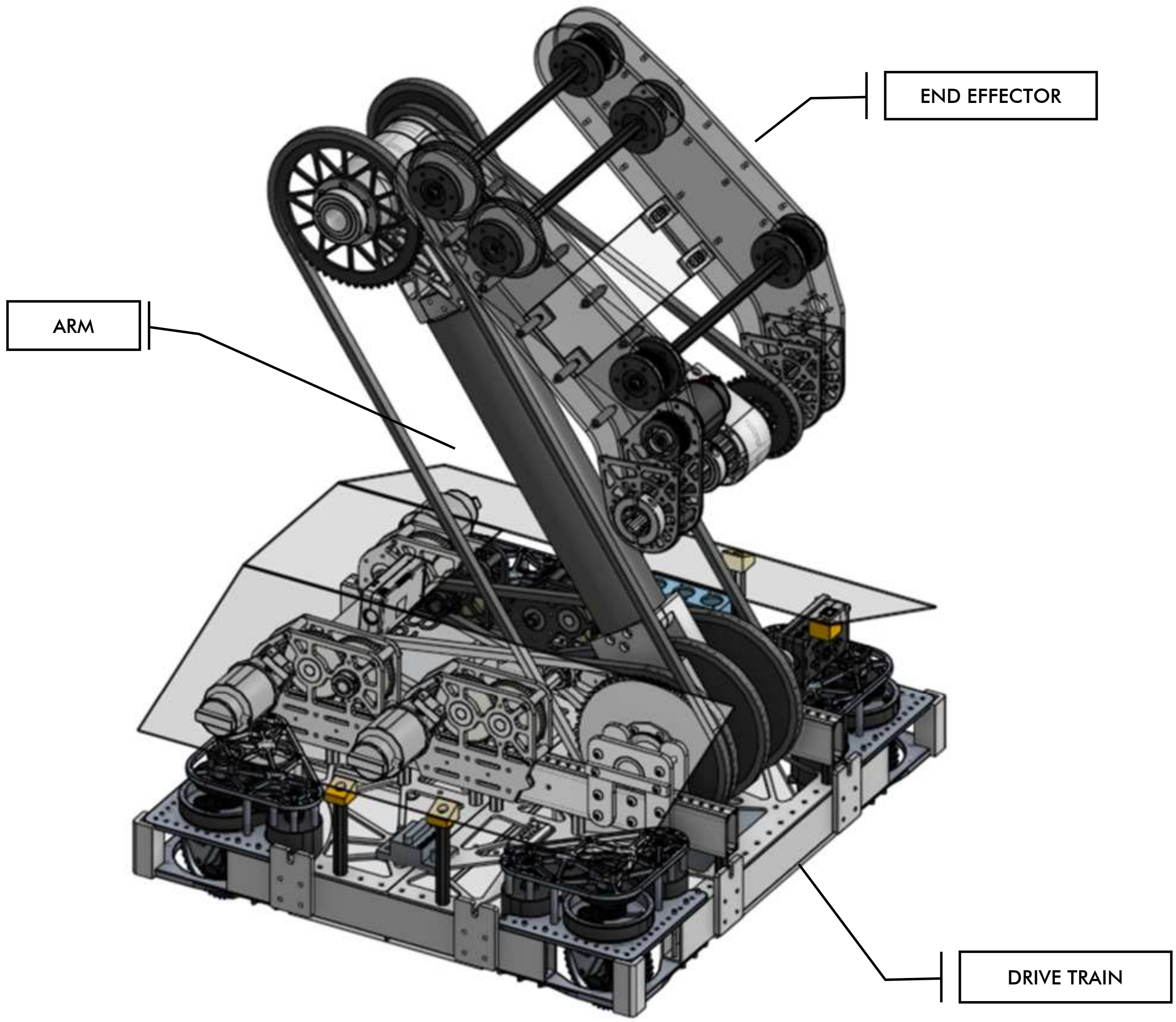


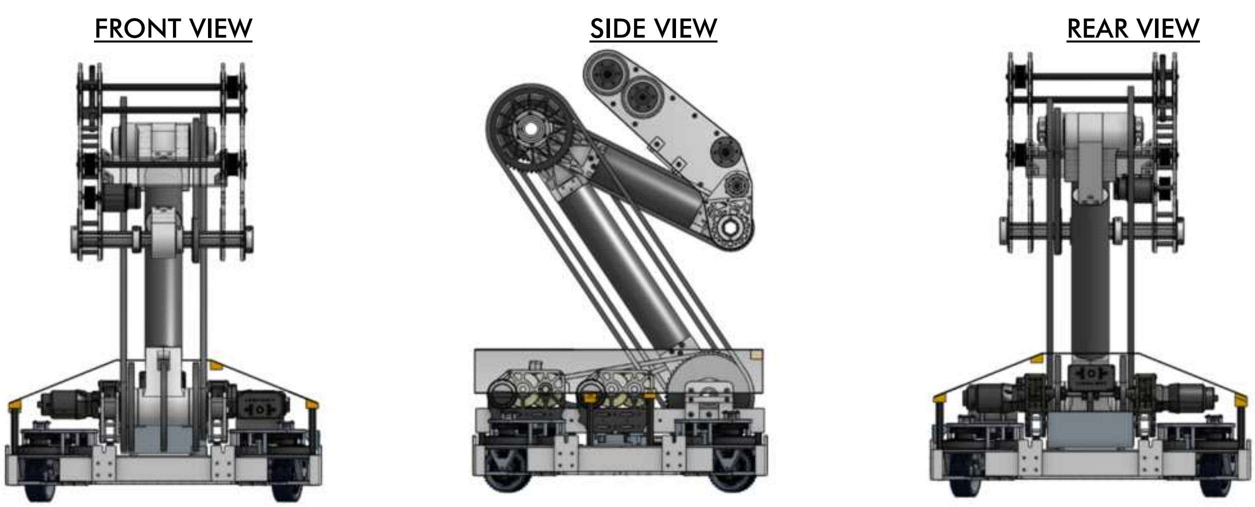
We are thinking about using an arm as a manipulation mechanism. We potentially envision a 2DOF arm + 1 DOF wrist that can pick up both cubes and cones, with a high range of motion on the wrist joint. As we can utilize the bot's movement, we do not need the Arm to move from side to side. An important note is that with an arm the starting configuration poses a good challenge, as it will need to fit inside of the robot's frame before activation. We have found that the shoulder joint only needs to move 90 degrees max, the elbow joint 210 degrees, and the wrist joint somewhere like 270 (at least in the configuration, lots to play with) to achieve all necessary motion.

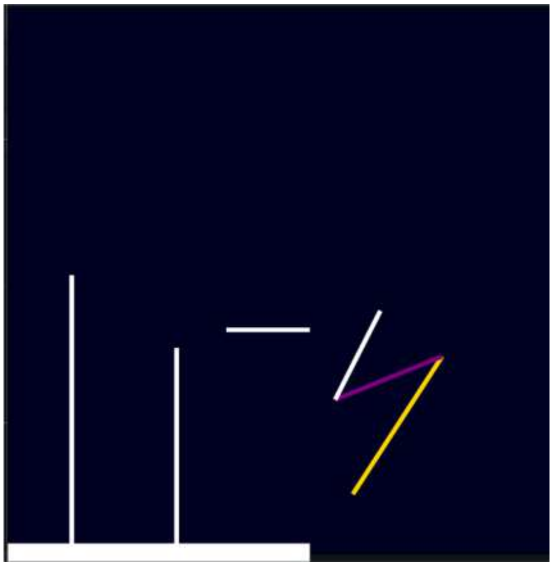
THE 1757 RAPID DEVELOPMENT MODEL

- Clearly Identify the design requirements of the system
- Design and Build a prototype that can be used to test design assumptions and Test
- Use what we learned from testing to develop a final design
- Fabricate final version and intergate into overall robot systems

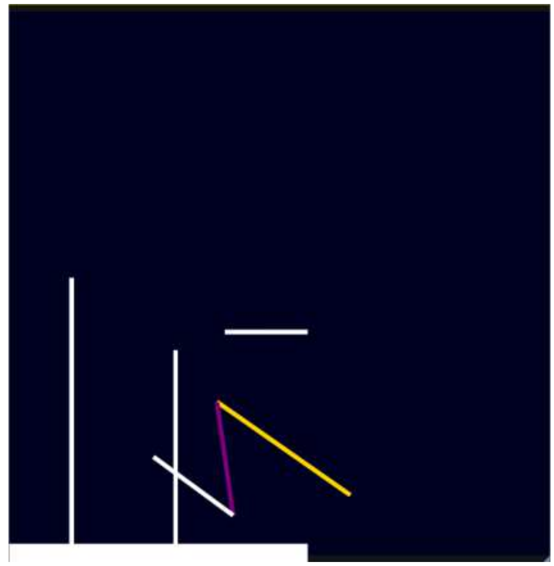
Final Robot Design



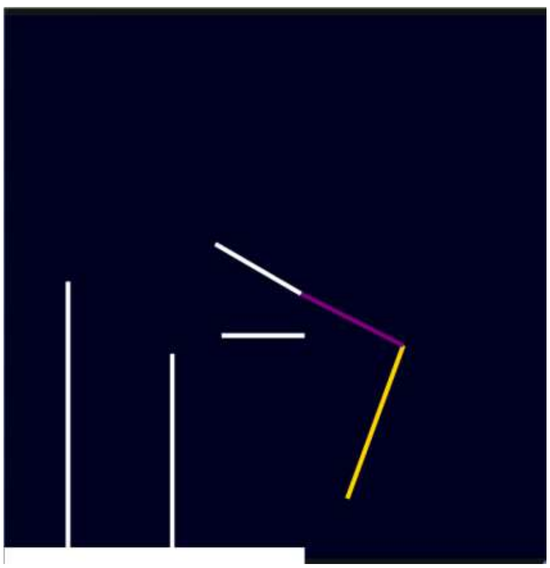




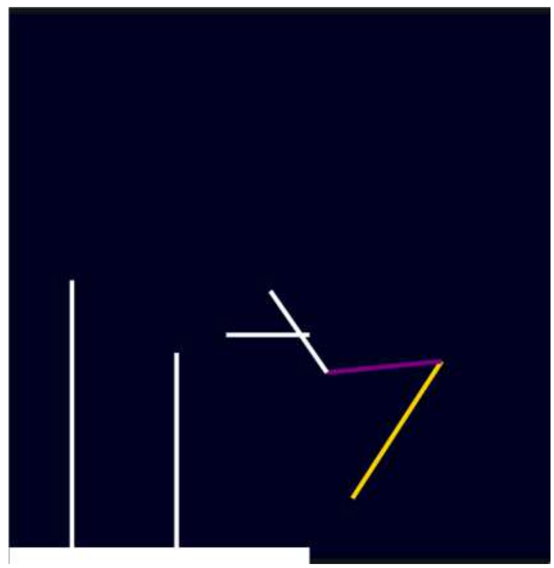
DEFAULT CONFIGURATION



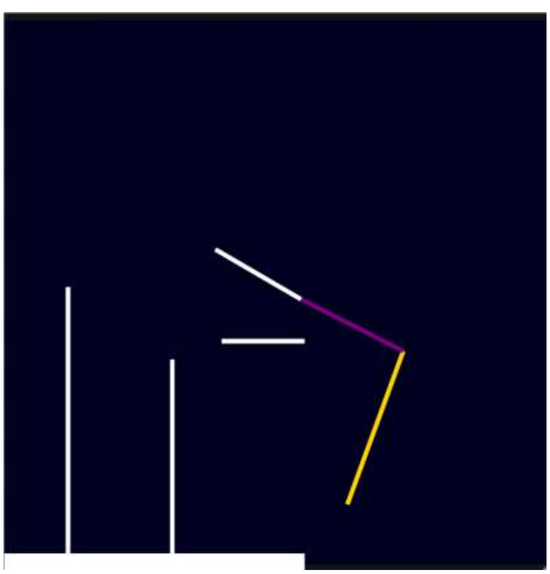
FLOOR PICKUP



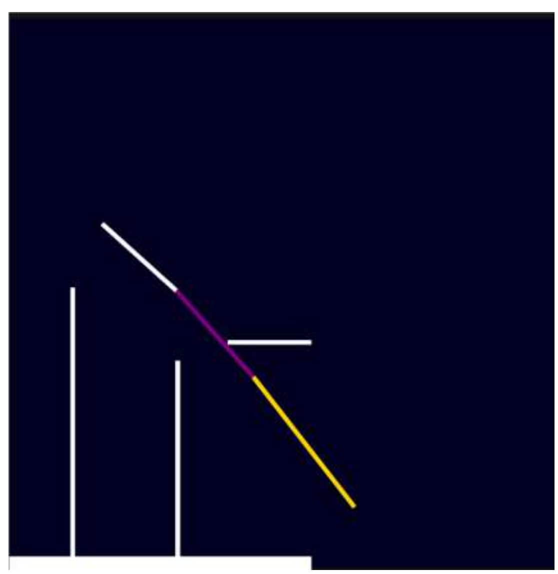
SUBSTATION PICKUP



SCORE – BOTTOM



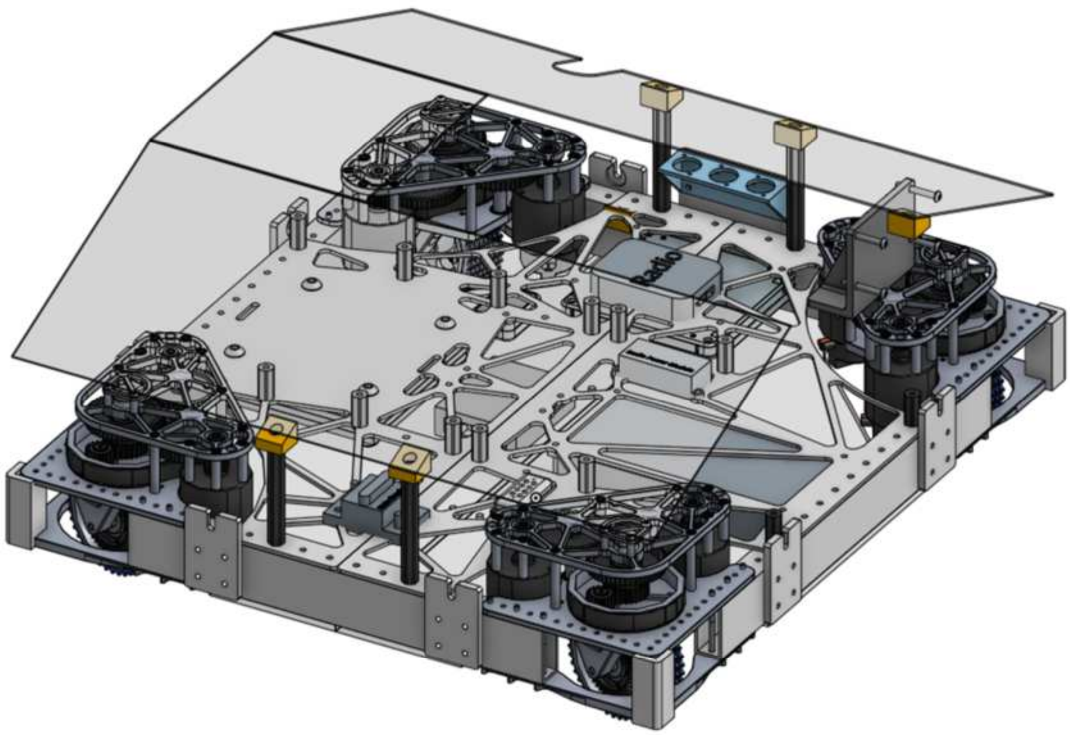
SCORE – MIDDLE

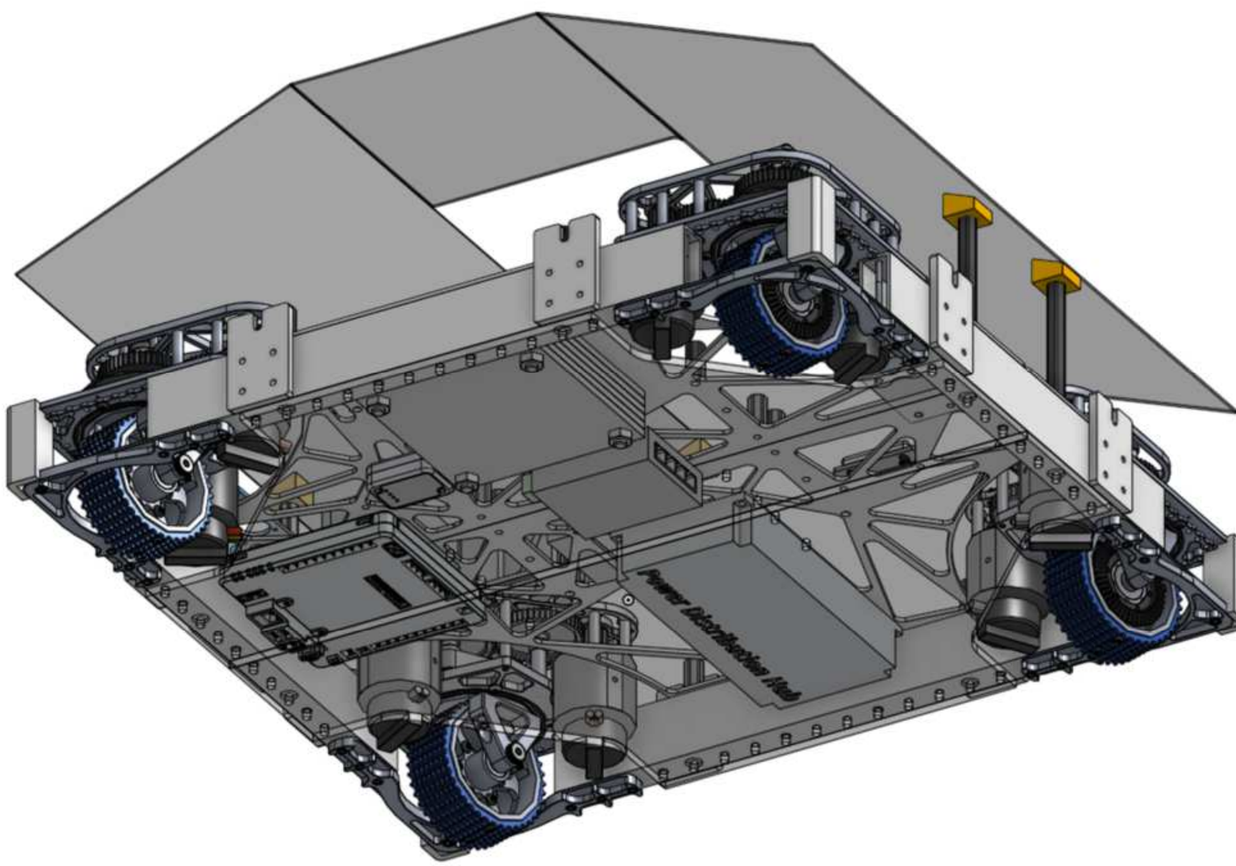


SCORE - TOP

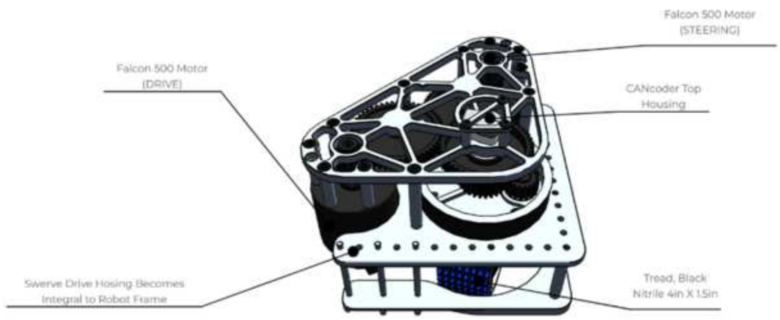
Above – 3D Simulation of Arm Joints in all of its various Arm Configurations

MAJOR SYSTEM #1: DRIVE TRAIN





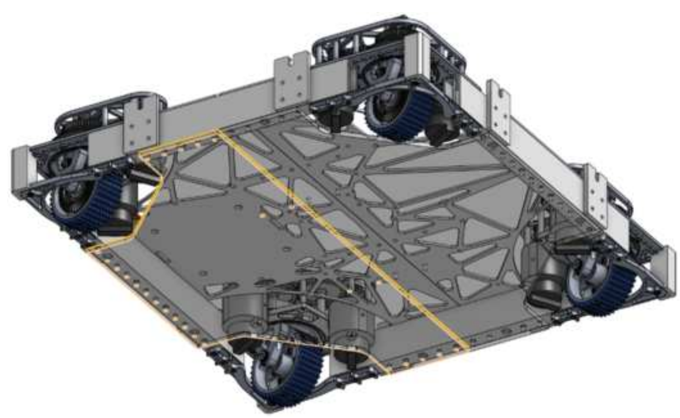
1.1 - SWERVE DRIVE MODULES



Last season was our first competitive season using swerve drive and we could not be happier with the results, because of the equal power between steering and driving there are none of the performance trade offs inherent in other drive systems. We can still run circles around the field when we need to and we can still push another robot across the field when they are in our way. One of our favorite features exclusive to swerve drive is what we call the park feature, by turning all 4 wheels to a 45° angle relative to the corners of the robot the robot effectively parks itself in place and wont move, another robot can push against us all match long and we wont move. Last year we used Swerve Drive Specialties Mk4 units, and this season we upgraded to the newly released Mk4i units. This revised design points the motors downward into the bot instead of mounting above the module. This allowed us to eliminate ≈2" of vertical space in our robot between the drive frame and the major systems.

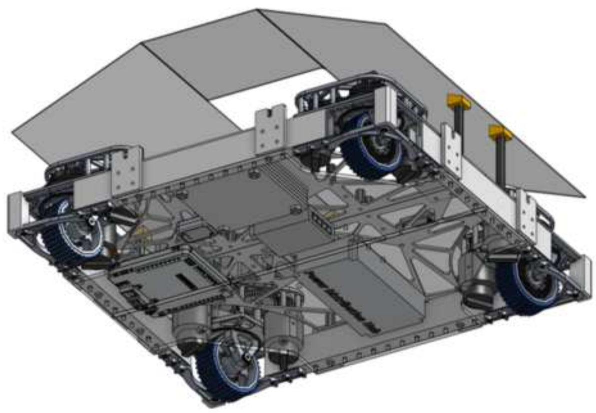
SDS MK4i Swerve Modules	
Powerplant	Falcon 500
Gearbox Configuration	L2
Overall Gearbox Ratio	6.75 : 1
Unadjusted Free Speed	16.3 ft/sec

1.2 - ROBOT FRAME



Left – CAD – Isometric Top View of robot frame
Right – CAD – Isometric Bottom View of Robot frame

1.3 - ELECTRONICS SUBSYSTEM



One thing we struggled with on our 2022 robot was how inaccessible most electrical components were. On our 2022 robot, all the electronics sat in a belly pan at the base of the bot, and the only way to access most of the components required you to remove the majority of the Robot Systems. Inspired by another team's design from 2022, we decided to hang all of our electronic components upside down and face the ground. Now to access the electronics, we tip the bot on its side, remove the ¼" protective polycarbonate plate, and you have full easy access to all the electronics components. (Credit to Team 125 for the Idea, they have the thanks of a grateful drive team and Pit Crew)

ELECTRONICS SYSTEM MAJOR COMPONENTS

- (1 ea) National Instruments - RoboRio 2
- (1 ea) REV Robotics - Power Distribution Hub
- (1 ea) Navex 2 – RoboRio MXP expansion Board
- (1 ea) CTRE CANivore
- (1 ea) CTRE CANdel
- (1 ea) BrainBoxes – SW-015 5 Port Gigabit Switch
- (1 ea) Generic Passive POE Injector
- (1 ea) Limelight 2 Camera
- (12 ea) Falcon 500 Motors
- (1 ea) REV Robotics Sparkmax brushless motor controller
- (1 ea) REV Robotics NEO 550 Brushless DC Motor
- (1 ea) Open Mesh Access Point [Insert Model Number]

1.4 - TESTING PORTS

We added a convenient patch panel to the upper side of the robot to allow for quick access to essential data ports when we don't want to access the underslung electronics.

PATCH PANEL SLOT 1 – USB TYPE A

This slot connects to one of the USB Type A ports on the RoboRio. This typically has a USB flash drive plugged in. During a match all the system logs are copied to the USB drive. After a Match, the USB drive can be pulled and opened up in AdvantageScope on the debug machine for post-game analysis. It's our version of a Blackbox on an airplane.

PATCH PANEL SLOT 2 – USB TYPE B

This connects to the USB Type B Port on the RoboRio—a redundant method for tethering the robot for control and debugging at events.

PATCH PANEL SLOT 3 – RJ45 CONNECTOR

This connects to the Ethernet Switch Via CAT5e for network access. Used for tethered connections to the bot during testing. Ethernet tethering is preferred, but we have encountered software reliability issues in the past.

DCMP Update

At the Revere District event we ran into serious problems tethering to the robot via ethernet and via USB B. we traced the ethernet tethering problem to a problem with the network configuration issue on the driver station laptop. We were unable to determine a definite cause of the USB-B connection issue, but, we think it most likely to to poor quality of the 90° usb connector used on the robot. From that point on we connect a USB-B cable directly into the port on thr

At the same time we realized we needed a button to manually put the arm motors in coast mode for serviceability when the bot is not connected to the driver station. Since we are no longer using the USB-B testing port we replaced it with a momentary push button switch.

1.5 – CAMERA/VISON SYSTEMS

LIMELIGHT 2 – CAMERA

We are utilizing a Limelight 2 Camera for a variety of tasks on the robot mostly devoted to sensor fusion and automation of systems using computer vision. The Limelight's field of vision (FOV) is essentially parallel to the floor and at the height of the April Tags.

Please refer to the Software section of this document for more information on how we use the limelight and April Tags to improve the Onboard odometry of the robot.

1.6 – COUNTERWEIGHT

Not originally intended as part of the design, upon testing of the robot with the Arm fully extended in the scoring position, we realized that robot was prone to falling forward. To resolve this, we looked to add ballast to the bot. First we thought of lead but didn't want to deal with the potential health risks of improperly encapsulated lead. We investigated tungsten; however, a review of the current price of tungsten plate (≈$40/kg) quickly ruled it out as a potential candidate. We settled on 6" x6" x1/4" steel plates mounted directly under the robot battery. After testing with different #'s of the plate, we decided on 6 Plates with a total weight of ≈25 lbs. Now the bot is highly stable even when the Arm is fully extended.

1.7 – PROTECTIVE COVERS

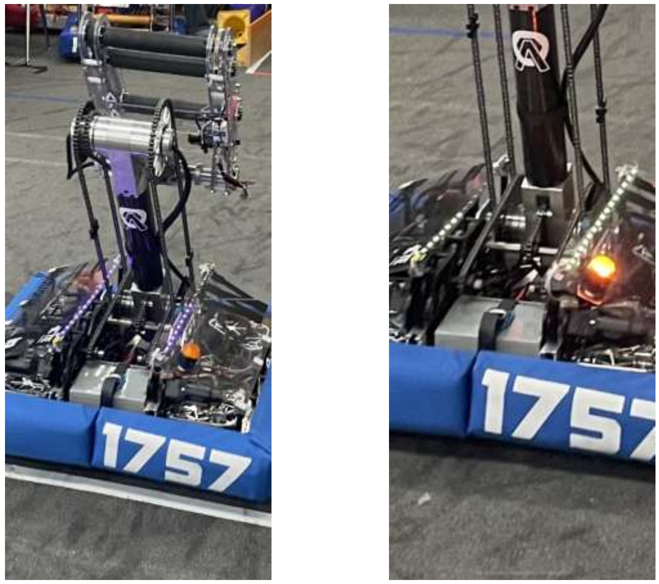
We added protective covers that Slope away from the central arm structure down to the bumpers. Not only do these plates provide a valuable location to display all of our great sponsors they serve to prevent errant game pieces from getting stuck inside the robot during a match.

DCMP Update

Originally the protective covers were only hled on with 3M™ Dual Lock™ SJ3560, this material is nice because it very strong but eaisily removable. Durring qualifying matches in Revere however, these panels kept falling off and dragging around the fieild. The Dual Lock strips were reinforced with zip ties and these held through all of playoffs in revere, and all of qualifications at WPI. Then in Playoff Matches we shered off 3 of the metal standoffs holding up the protective covers. We made quick repairs to keep going however prior to DCMP we will be swaping out all the ½ thunderhex standoffs with 1" 80/20 extrusion with hardened bolts for strength.

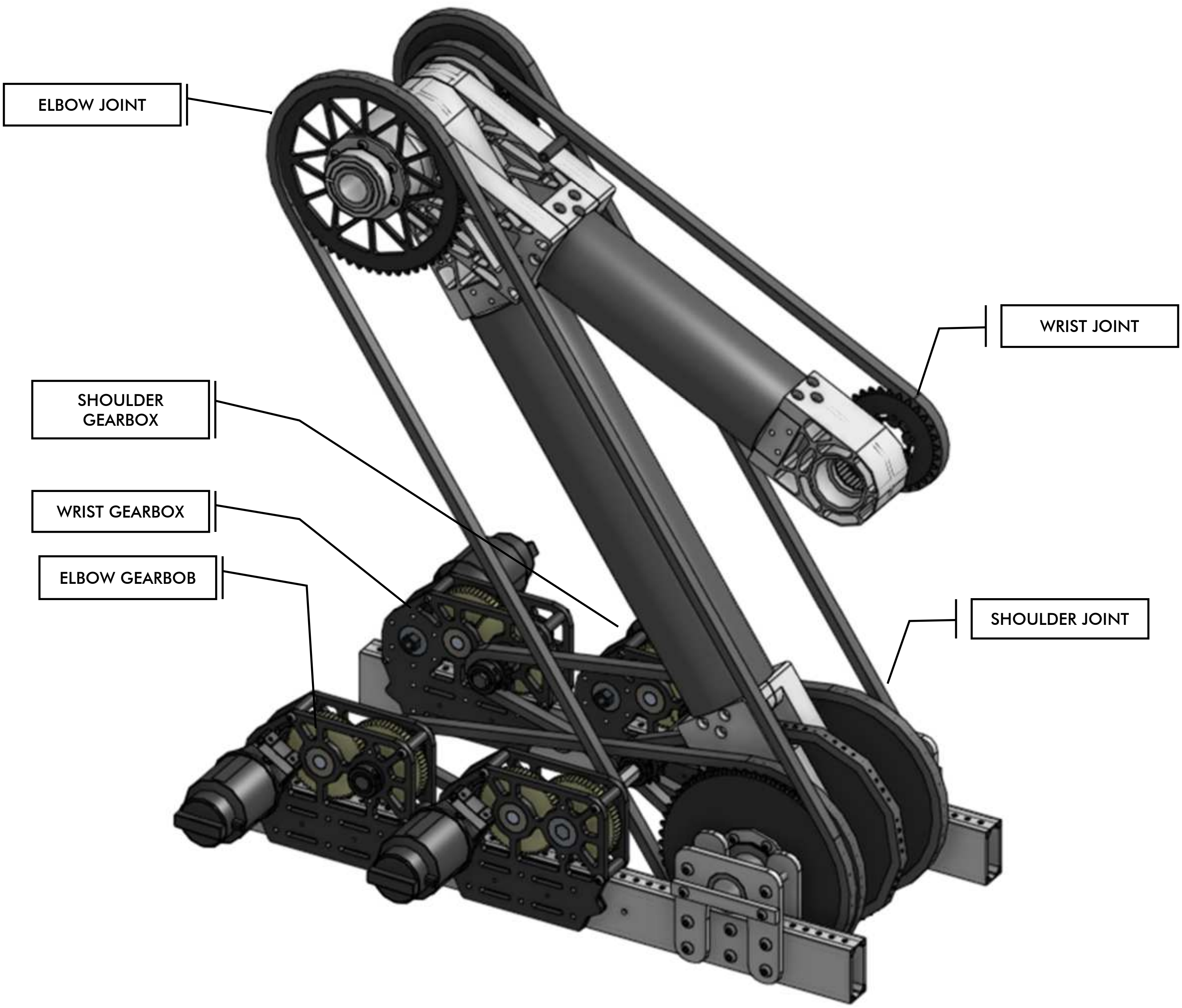
1.8 – GAMEPIECE INDICATORS

One of the hardest things to do in a match is how to signal between the driver station and the human player what game piece you want them to load into the robot. People use Hand Signals, Colored pieces of paper or guess. We wanted to take the guesswork out of the equation, so we mounted 2 LED Strip lights along the top of the protective covers. The driver controls what color these strips are so he can communicate to the human player which game piece to feed to the bot – Yellow for Cones and Purple for Cubes.



Left – Robot Displaying "I Want a Cone"
Right – Robot displaying "I Want a Cube"

MAJOR SYSTEM #2: ARM



2.1- MOTORS & GEARBOXES

To keep the robot's center of gravity low and keep the Arm as simple as possible we decided to locate all of the heavy motors and gearboxes at the base of the superstructure.

GEARBOX # 1 – SHOULDER GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into Custom Gearbox based on WCP Gearbox design.

GEARBOX # 2 – ELBOW GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into COTS WCP Single Speed Gearbox.

GEARBOX # 2 – WRIST GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into COTS WCP Single Speed Gearbox.

2.2 - CHAIN DRIVE

Using a combination of dead and live axels we transfer the power of the gearboxes up though the Arm to power each of the individual joints. For Reliability and durability, we chose to use #35 roller chain rated for 11,000 lbs of force.

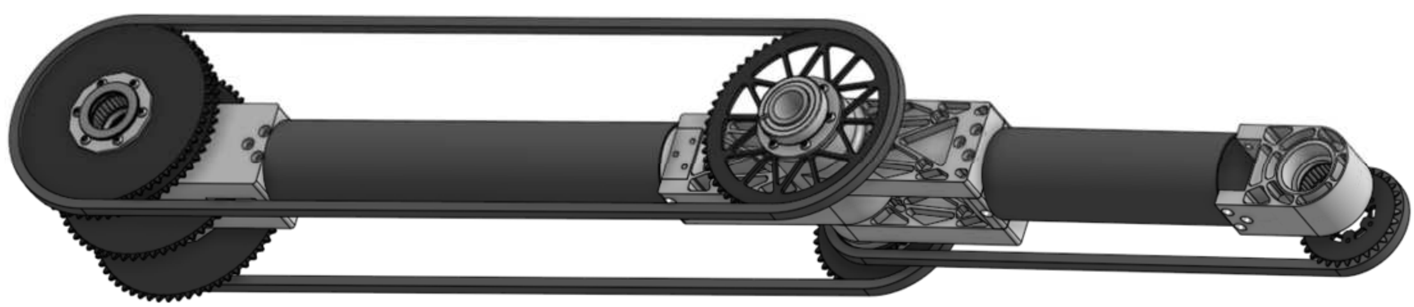
Below is a summary of the different chain runs on the Arm

CHAIN DRIVE 1 – SHOULDER
Shoulder Gearbox Output 12t Sprocket → 60t sprocket on Shoulder (Dead Axel)

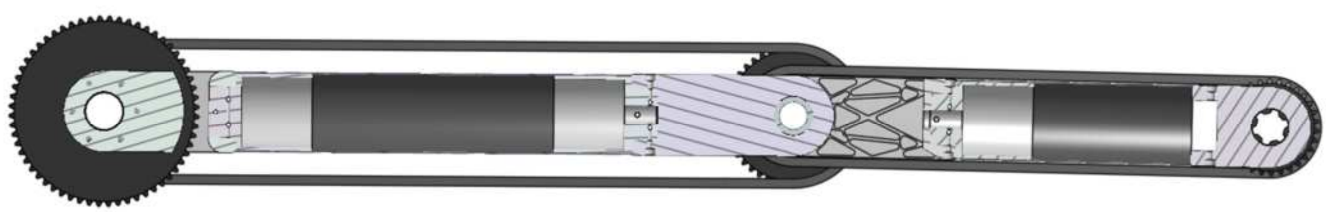
CHAIN DRIVE 2 – ELBOW
Elbow Gearbox Output Shaft 12t Sprocket → 60t Sprocket on Shoulder (Live Axel) → 70t Sprocket on Elbow (Dead Axel)

CHAIN DRIVE 3 – WRIST
Wrist Gearbox Output Shaft 12t Sprocket → 60t Sprocket on Shoulder (Live Axel) → 60t Sprocket on Elbow (Live Axel) → 40t Sprocket on Elbow (Live Axel) → 32t Sprocket on the Wrist (Dead Axel)

2.3 - ARM STRUCTURE



Above - 3-D View of Outstretched Arm



Above - Section View Though Center of Carbon Fiber Arm

CARBON FIBER ARMS

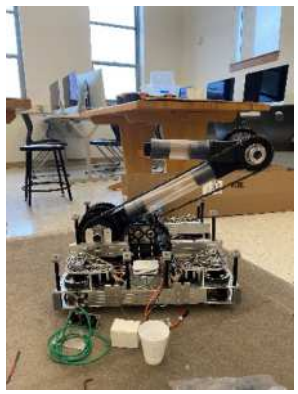
We chose to use carbon fiber tubes as the main structure of the Arm due to its strength and lightweight, the more weight we could save on the Arm the lower we could push the robot Center of Gravity. Carbon Fiber tubes are a stock McMaster item 3" Ø. Carbon Fiber is Epoxy bonded to 3" hollow aluminum plugs bolted to the aluminum joints.

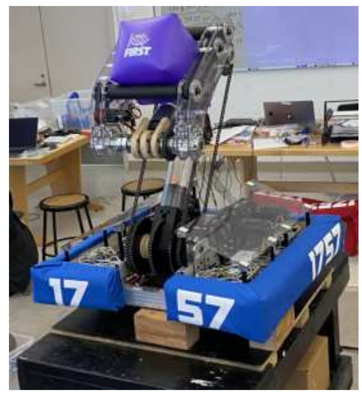
Left – Final carbon fiber arm links after final glue up

Right – Final aluminum plugs used in the ends of the carbon fiber tubes

3D PRINTED AND POLYCARBONATE PROTOTYPES

Because we knew the carbon fiber and machined aluminum would take time and money to manufacture, we heavily used 3D-printing to make prototypes of the Arm and test and confirm critical geometry before placing final fabrication orders. These prototypes are two fragile to be used on a competition bot but worked well for their intended purposes. We learned very Important lessons about where the concentrations of forces were along the axels and what parts needed reinforcement.





Left – 3D printed Prototype of the wrist joint, printed on a FormLabs 2 SLA Printer

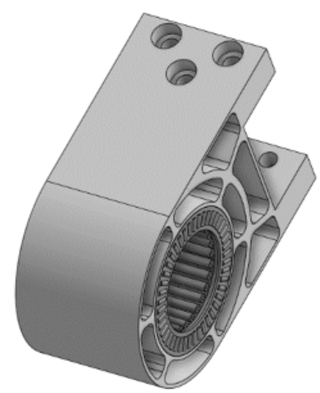
Center – Polycabonate Prototype arm Mounted on bot for the First Time

Right – Fully Assembled "Alpha" Robot build

2.4 – JOINT STRUCTURE

CAD - Shoulder Joint

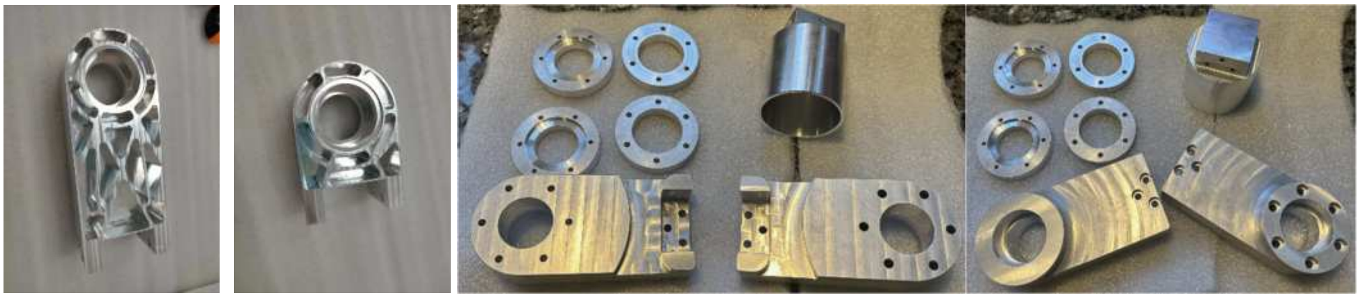
CAD - Elbow Joint



CAD - Wrist Joint

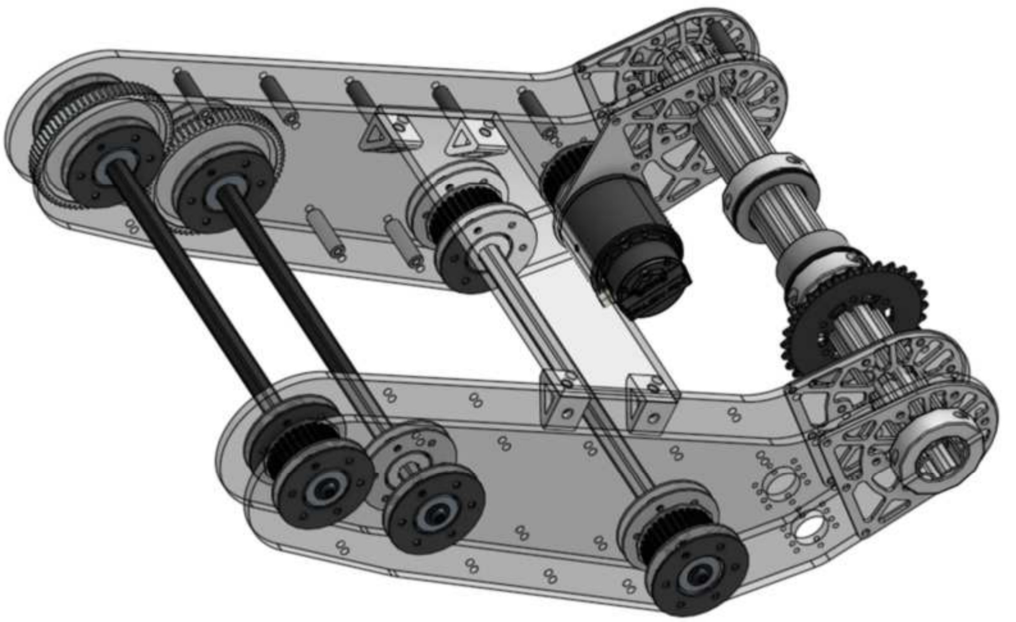
NEEDLE AND THRUST BEARINGS

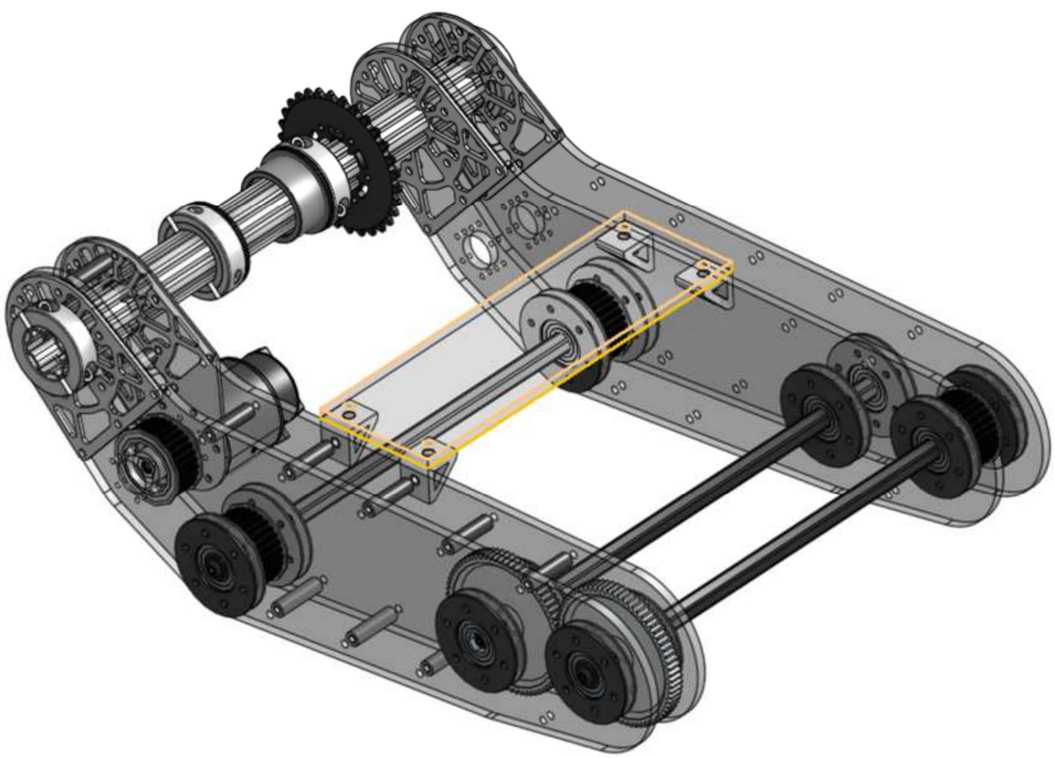
Used in All three joints to allow for smooth rotary motion in each joint.



Photos - Final Machined Parts

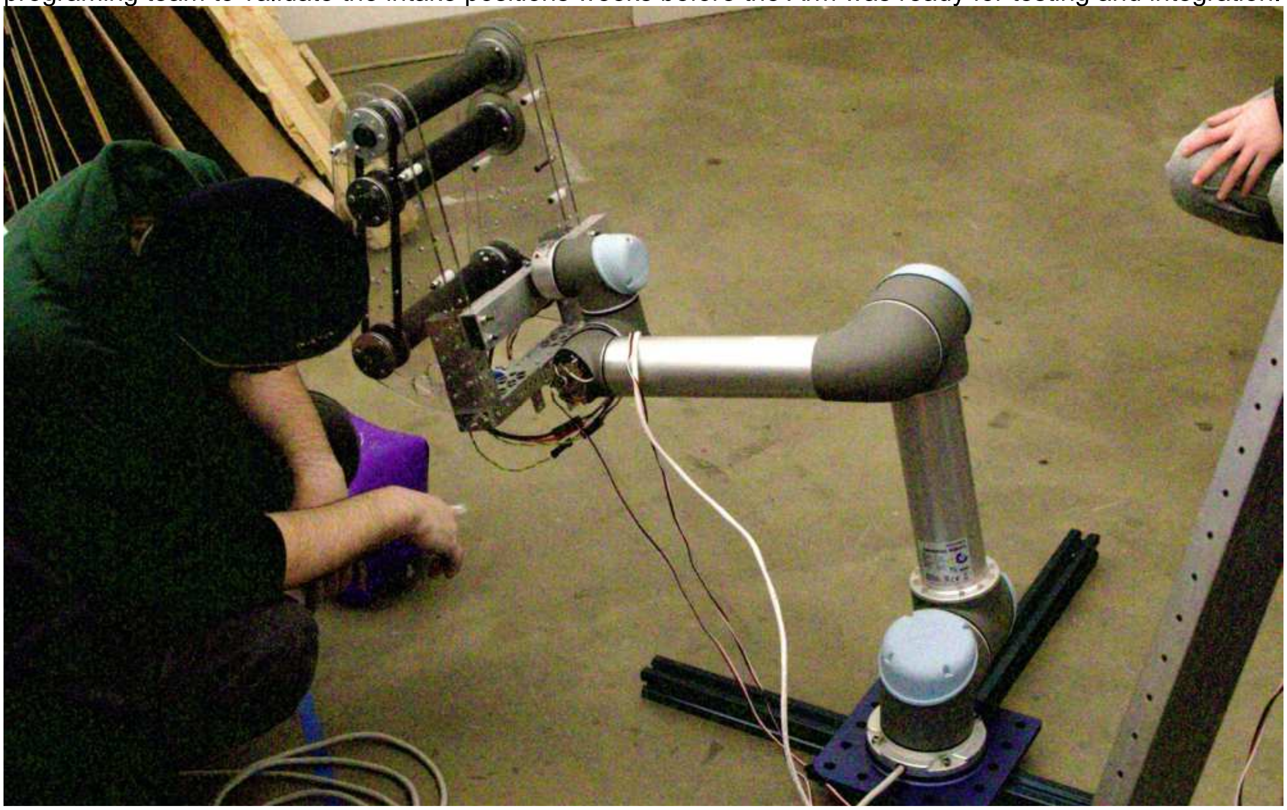
MAJOR SYSTEM # 3: END EFFECTOR





INTAKE PROTOTYPING – FUN WITH INDUSTRIAL ROBOTS

We had a lot of fun prototyping this mechanism, since it was the first major mechanism that we finished we had lots of time to put it trough its paces. One of our mentors has access to a Universal Robots UR3 robot and brought into our lab during one of our weekend build sessions (see images below). This actually proved to be essential because it allowed our programing team to validate the intake positions weeks before the Arm was ready for testing and integration.



3.1 – ROLLERS

The rollers we are using are Vex VersaHub Rollers with ¼" neoprene tubing stretched to cover them, they are very grippy and hold the cube verry securely. We started with the dimensions of the everybody roller for our prototype then made some modifications before settling on final separation differences. The neoprene tubing is undersized for the OD of the polycarbonate roller. We learned a fun trick to clamp off one end of the neoprene tube and inflate it with an air compressor to stretch it over the polycarbonate tube, when the air is released, it makes a perfect friction fit between the Neoprene and the polycarbonate. We have had no detectable slippage after weeks of testing with the rollers.

DCMP Update

After 33 competitive matches one thing is clear, we have problems picking CUBES up off the floor and in order to maintain our competitive edge at DCMP we know we need to be able to get CUBES up off the floor. We thing the majority of the problem is related to how narroe our end effector is, the original design was only 1” wider than the width of the CUBE. To aliviate this issue we are planning to widen the end effector by 3”.

3.2 – MOTORS & GEARBOXES

The intake is powered by a REV Robotics NEO550 Brushless motor into a REV Robotics Ultraplanetary gearbox. The motors small size is nice however because we mounted the Sparkmax Motor controller on the intake as well there is no significant weight savings compared to using a Falcon 500 with an integrated Talon SRX. We may end up swaping this out for simplicity sake in the future.

REV Ultraplanetrary	
Powerplant	NEO550
Gearbox Configuration	4:1, 5:1
Overall Gearbox Ratio	20:1

SOFTWARE

SOFTWARE: OUR DEVELOPMENT ENVIRONMENT

WPILib

The perineal stalwart, we still rely on core elements of WPIlib for robot communications and debugging. WPILib's new Logging features have greatly enhanced our Debugging capabilities

RobotPy

We have found that students have a lot easier time learning python then they do Java or C++ so with the growing support for RobotPy we migrated our Codebase from Java to Python in 2020. As of this March we are an official contributor to the RobotPy project

GitHub

Without Github our level of remote work and collaboration just wouldn't be possible.

SOFTWARE: NEW AND UPDATED TOOLS THIS YEAR

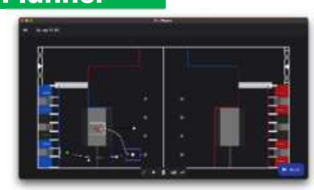
AdvantageScope

WE LOVE ADVANTAGE SCOPE! Not only does it log *everything* but does it in away that is intuitive and easy to review. No more searching though 10000 lines of log files to find the one piece of information we need. Huge thanks to team 6328 for building such a great tool.

PhotonVision

We are using PhotoVision as our native development framework for Computer vision due to its growing wide support inside the FRC community. It does not include native support for RobotPy however so as an offseason project our lead programmer wrote a custom wrapper for PhotonVivion so it can work inside our RobotPy environment

PathPlanner



Last year we used PathWeaver, but we were disappointed in the lack of native support and increased complexity in the development stack so starting with the off season we transitioned all of our Autonomous path planning to PathPlanner. We had much fewer issues with this system.

SOFTWARE: DRIVE

Taking off of last year, the drivetrain codebase has stayed the same. We are running field oriented drive with robot relative rotation to allow for quick maneuverability. A button to align to the nearest 90 degree angle was added to help with driver alignment. This state slightly reduces the speed and snaps the angle of the robot in order to have perfect alignment to the double substation, single substation, and grid every time. For our automated balance sequence, we work in robot relative space on the robot relative gyro.

A Brief Tangent - Absolute Relative Drive

Last year our lead programmer had a new idea for drive control, an absolute relative drive. The common swerve drive control method was to have a field relative translation for the bot, and a robot relative rotation. What this meant is a left input on the rotation axis would result in the robot rotating to the left at a constant speed. A translation action was not affected by rotation but instead was in "field relative" space. The difference of absolute drive is that the rotation is also field relative. A left input on the rotation stick will yield the robot turning to face left. This year we expect this type of robot control to be very important for drivers when they have to be able to turn to specific positions for collection and scoring on swerve drives. You can see this in action in any one of our videos from last year. Having fixed controlled rotation will allow for precise driver input and less fiddling with controls when cycle time is very important.

The drivers have also experimented with alternate driving methods on swerve to get used to interesting control schemes such as a curvature drive, standard tank drive, standard field relative drive, and full robot relative drive.

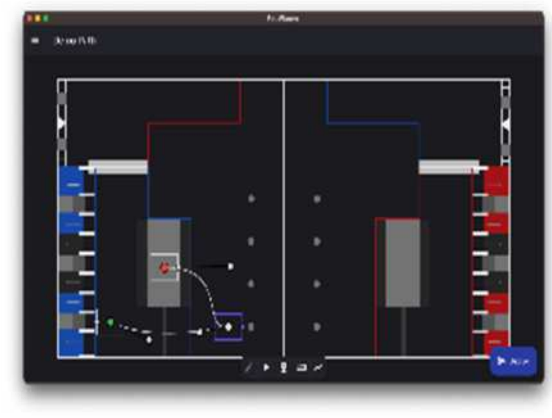
SOFTWARE: INTAKE

The intake is using a state machine in order to regulate its expected behavior. There are 2 enumerable values: one for the gamepiece intended direction (intake, outtake, hold) and one for the desired gamepiece type (cube, cone). The transition between each state is dictated by a user input to any given category. If no input is given, the system holds its position and keeps the desired gamepiece remembered. The state value of the desired gamepiece is displayed to the driver and to the human player through pulsing leds of the respective color.

SOFTWARE: ARM

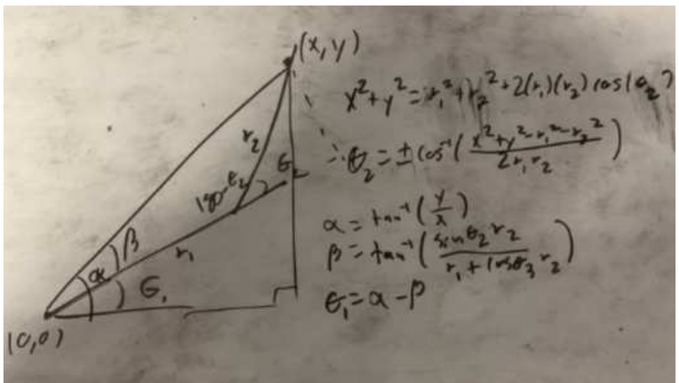
A triple jointed arm is no easy feat in order to program smoothly. From cad, states are given about the end effector's desired position and rotation relative to the floor. From there we use inverse kinematics to determine each per-joint relative rotation at any given position. A cartesian control on the wrist joint's position is added using a trapezoidal PID profile to lay out a path for the Arm to follow. For the wrist it has another trapezoidal PID profile controller. When going state per state on the Arm's motion, we check for if the relative angle goes over software end limits in order to prevent running the Arm into itself. These are done in joint relative space. Since the Arm is controlled from the base through chain and sprockets there is a virtual 4 bar created in which the rotation of any given joint is given relative to the ground. These are converted into motor space and passed onto each motor where they have a position PID controller onboard. For simulating the system we are using mechanism2d to view the expected values of the Arm and be able to run through positions. This simulation first approach has allowed minimal revision and a solid foundational codebase that is mostly complete before the bot is finished. Furthermore the position of the Arm is logged in 3d with advantagescope based on the position it believes the bot is in. Logging was also a priority for a complex system and therefore we log the instructed position and actual position per joint and each end position in cartesian space.

SOFTWARE: AUTONOMUS



We experimented in the offseason with pathplanner and use it extensively for our autonomous. Each necessery command is placed into a potential pool of events for pathplanner to fire. At the beginning a sequence determined solely in pathplanner is fired. Going off of last year we had a large time gap in order to make slight changes so instead for this year we are using the waypoint system and opting to have any given autonomous contained solely in pathplanner. This has increased our ability to construct autos and tweak any given aspect as needed. For the library itself of parsing, the lack of first class robotpy support meant we had the freedom to reimplement autonomous however we pleased based on the path. We follow a changing trajectory and the swerve drive using onboard odometry and a weighted vision estimate determines its bot position relative to the global field and follow through it between each section.

SOFTWARE: SIMULATION



armsubsystem.py

```
    def setEndEffectorPosition(self, pose: Pose2d):

        twoLinkPosition = Translation2d(
            pose.X() - constants.kArmwristLength * pose.rotation().cos(),
            pose.Y() - constants.kArmwristLength * pose.rotation().sin(),
        )

        endAngle = math.acos(
            twoLinkPosition.X() * twoLinkPosition.X()
            + twoLinkPosition.Y() * twoLinkPosition.Y()
            - constants.kArmtopLength * constants.kArmtopLength
            - constants.kArmbottomLength
            * constants.kArmbottomLength
            / (2 * constants.kArmtopLength * constants.kArmbottomLength)
        )

        startAngle = math.atan2(twoLinkPosition.Y(), twoLinkPosition.X()) -
math.atan2(
            math.sin(endAngle) * constants.kArmtopLength,
            constants.kArmbottomLength + math.cos(endAngle) *
constants.kArmtopLength,
        )
        wristAngle = pose.rotation().radians() - startAngle - endAngle

        bottomArmEncoderPulses = (
            startAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kBottomArmGearRatio
        )
        topArmEncoderPulses = (
            endAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kTopArmGearRatio
        )
        wristArmEncoderPulses = (
            wristAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kWristPivotArmGearRatio
        )

        self.topArm.set(Falcon.ControlMode.Position, topArmEncoderPulses)
        self.bottomArm.set(Falcon.ControlMode.Position,
bottomArmEncoderPulses)
        self.wristArm.set(Falcon.ControlMode.Position, wristArmEncoderPulses)
```

Due to our team's resources, virtual simulation is a huge part of our ability to quickly and reliably construct the bot's codebase. Some key examples of simulation are a wrapped onto a simulated falcon motor. Given our team's extensive use of falcons on the robot, a wrapper that provides simulation support allows for the programming team to iterate much easier and creates a cleaner codebase. Each falcon is logging the values of the motor % and the encoder position, as well as an override value to allow the user to manually in simulation change the value for sensor readings. Entire robot configuration is done on a single call and the getting of velocity, position, and percent and the setting of velocity, position, and percent are easy to access functions to allow interfacing with the motors more accessible than the CTRE library. Given this robot also has a NEO550, the simulation system was adopted to have a similar interface for ease of replacement from a falcon to a motor on the intake. We geometrically derived the inverse kinematics for 3 links with a fixed Pose endjoint. Each of these poses actually allows for two configurations of the proximal 2 arm joints (they can simply be mirrored over the line created from the wrist joint to the shoulder joint, however by forcing the sign on the elbow joint they can all be consistent.

SOFTWARE: VISION

NOW WITH APRILTAGS AND PHOTONLIB

We have a vision system complete with sensor fusion for complete robot localization. Last year,we worked with our first complete vision system as a team that resulted in significantly enhanced system performance, and using apriltags will be very important to account for combined sensor error as well as for being able to reliably use sensor data for automated alignment to various points on the field such as the double substation and the grid.

THE HOW

Photonvision generates camera-relative 3d transforms of each apriltag. Since the position of the camera is known and the position of the apriltag is known, the position of the robot can be determined from a single apriltag datapoint. These transforms are fed into a RobotPoseEstimator in order to create a sense of where the robot could be at a given time, this is combined with the gyro and wheel encoder information to get an accurate sense of where the robot is on the field at any given time. This is used in other subsystems when needed, as well as results being logged to AdavantageScope through the usage of each known pose and ghost poseseepaste

GOING FURTHER

We plan on using this odometry data to have automated alignment in complete robot space for important precision actions such as placement of gamepieces on the grid and collection of those gamepieces. Autonomous will also use this data. Perhaps an automatic engagement on the charge station by using the rotation gained from the apriltags will be possible. Overall having a sense of where the robot is on the field is beneficial to aid in other systems.

ENGINEERING TEAM

WHS Faculty Advisors

James Looney & Raul Madera

Team Captain

Charley Marsland*

Team Business Lead

Sean Tao

Team Technical Lead

Luke Maxwell

Senior Mentors

Dwight Meglan | **Chris Aloisio°** | **Steve Harrington°**

Mentors

Anthony Gelsomini | **Manny Barros°** | **Sean Lendrum°** | **Mark Holthouse** *Mentor Emeritus* | **Amber Maxwell**

STUDENT MEMBERS

Safety Captain

Jacob Kaplan

Graphic Design Lead

Ivan Cai

CAD Lead

Landon Bayer

John Santasuoso*	**David Girard***	**Declan MacDonald**	**Liam McWeeney**
Alex Theofilou	**Claire Peng**	**Andrew Gong**	**Rachel Qu**
Baili Jiang	**Nolan folmar**	**Jeffery Pan**	**Melissa Yang**
Divya Siddi	**Vinny Milinazzo**	**David Confoey**	**Adrianna Cirillo**
Elizabeth Lowney	**Jacob Liu**	**Anthony Yang**	**Constantina F.**
Luke Li	**Owen Monahan**	**Erik Curlli**	**Kevin Bai**
Henry Marsland	**Nina Pappas**	**Rachel Lee**	**Hanya Yang**
Sophia Patrick	**Lauren Buza**	**Kaylee Phu**	**Elias Mukdissi**

Westwood Robotics, Inc
Board Of Directors

Steve Harrington° *President*	**Catherine Tseng** *Treasurer*	**Manny Barros°**	**Cheten Gopal** *Clerk*	**Jack Tseng**

**Denotes High School Seniors*

°Denotes FRC Alumni

SPONSORS

Heartlander Surgical	Westwood High School	Medtronic	Act Fasteners
Gene Haas Foundation	TE Connectivity	General Dynamics	IRA Motor Group
Bayer Familt	Capital Group	Yamaguchi Family	Plymouth Rock Assurance
Deadham Savings	Maxwell Family	Schneider Electric	Marsland Family
Porier Apliances	Franklin Templeton Investments	Jake and Joes Resturant	McGolf
Roche Bros		Capers Catering	

VENDORS

Andymark	West Coast Products	Vex Pro	Swerve Drive Specialties
Discord	Google Work	Onshape	Robo Promos
McMaster-Carr	The Home Depot	Cross the Road Electronics	MS Visual Studio
Python	GitHub	RobotPy	Java

1757

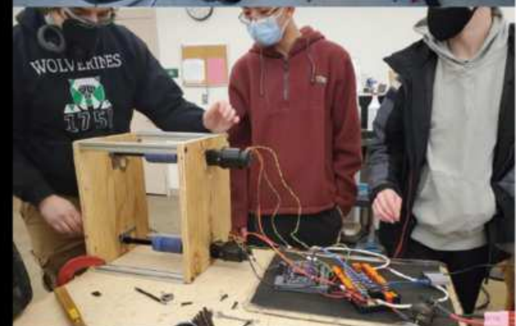
Special thanks to everyone who makes this team possible Including

Our Mentors
Our Students
Our Teachers
Our Parents

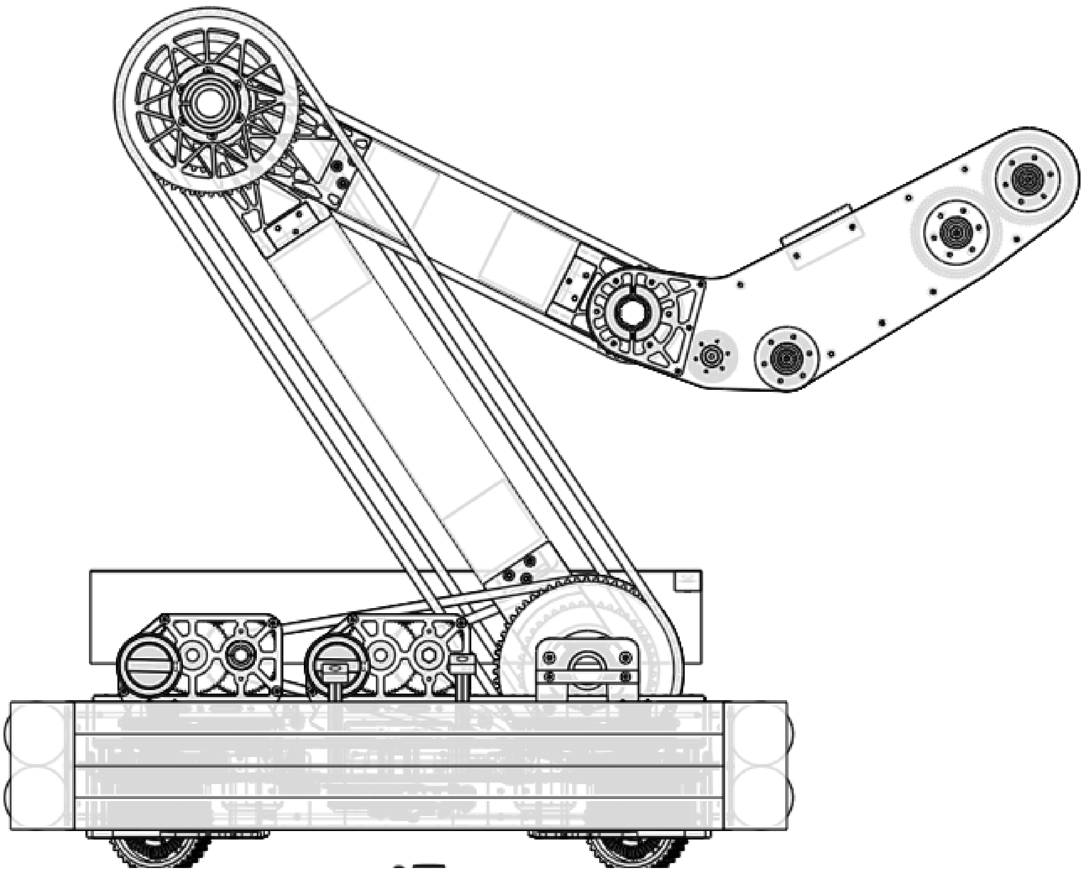
Our Amazing Sponsors

Westwood Public Schools
Emily Parks - Superintendent
Allison Borchers - Asst. Superintendent
Deb Nolan - Building Use Coordinator
Tom Carey - Director of Facilities
Ron Smith - Assistant Director of Facilities
Amy Davenport - Principal
Aishleen Flanagan - Assistant Principal
Tom Millett - Assistant Principal
Andrew Miller - Science Dept. Chair
WHS Janitorial/Facilities Staff
WHS Library Staff
The Open Alliance

#FRC1757
@WWRobotics1757





FRC WORLD CHAMPIONSHIP EDITION

LUXO

WOLVERINES

1757

2023 TECHNICAL BINDER

FORWARD

Hello, and let us welcome you to FRC Team 1757's 2022-2023 Season. This season has continued the tremendous growth in our robot's design and technical ability that started last year as our team emerged from the hibernation of COVID-19 to become a surprising contender in the New England Region. Continuing to recruit rookie students to supplement our now more veteran team members and Senior Mentors, we have pushed our collective talents to their limits to deliver the competition-worthy robot contained within the pages of this binder.

Our season started in the fall of 2022, introducing a new class of over 10 freshmen, sophomores, and juniors to the world of FRC. We showed off the robot at local town events, built a T-Shirt Cannon to raise school spirit at the prep rally, and hosted weekly technical seminars on everything from the engineering process to CAD, Electronics, Pneumatics, Mechanics, and everything in-between. Over the Summer we got a new OMIO X8 bed router and practiced our CAD and fabrication skills by designing and building an enclosure for the machine. We traveled to Billerica, MA in October to compete in the first-ever New England Robotics Derby. We finished in Second Place, losing in the Finals (The best competitive finish in team history). We piled into our classroom on a cold Saturday morning in January, eagerly anticipating this year's game. 4 CAD models, 8 shared Google Drives, ten weeks, 20 Weekend Build Sessions, 50 Zoom calls, 5799 lines of code, 170 git commits, 19,129 discord messages, and many, many cups of coffee later, we are proud to unveil our robot "LUXO" for the 2023 FRC Season.

Why did we name the robot LUXO? Is it because of the shining lights on its frame that illuminate what game piece we are looking for on the field…no. Is it the bright shining future of the team…no. Is it a reference to solar power and how that ties into the theme of this year's FIRST season…good guess, but no. In truth, we are a bunch of animation nerds, and we thought the robot looked like the lamp in the Pixar Animation title sequence named Luxo. Not every robot name has a deep prophetic meaning…sometimes it's just about the memes.

One very exciting thing about this year is that Team 1757 joined the Open Alliance. We found the Open Alliance teams and their open and timely build season updates so helpful to our team last season that we decided to join so we could help other teams the same way the alliance has already helped us. In addition to frequent updates on our build thread, we also made two appearances on the Open Alliance Show Streamed on twitch. If you want to learn even more about our robot and the design process, beyond what is contained in this manual, please visit our Chief Delphi Build Thread at https://www.chiefdelphi.com/t/frc-1757-wolverines-2022-2023-build-thread/416564

We hope you enjoy this brief look at the design process and technical details that went into this robot, and if you have any questions, look for one of our team members in the stands, in the pits, or on the field. We are always ready to share the knowledge we have gained and share a few hard-learned lessons we learned along the way.

DCMP Update

So it has been a whirlwind of a season so far, after meddeling performance at Greater Boston district we went on win the WPI District Event. Not only were we Alliance captian of the the #2 alliance, we also won the Engineering Inspiration award at WPI. Though out this document you will find various updated information featuring design changes/Repair/modifications that were made during the competition season.

Competitive Record Though District Play:

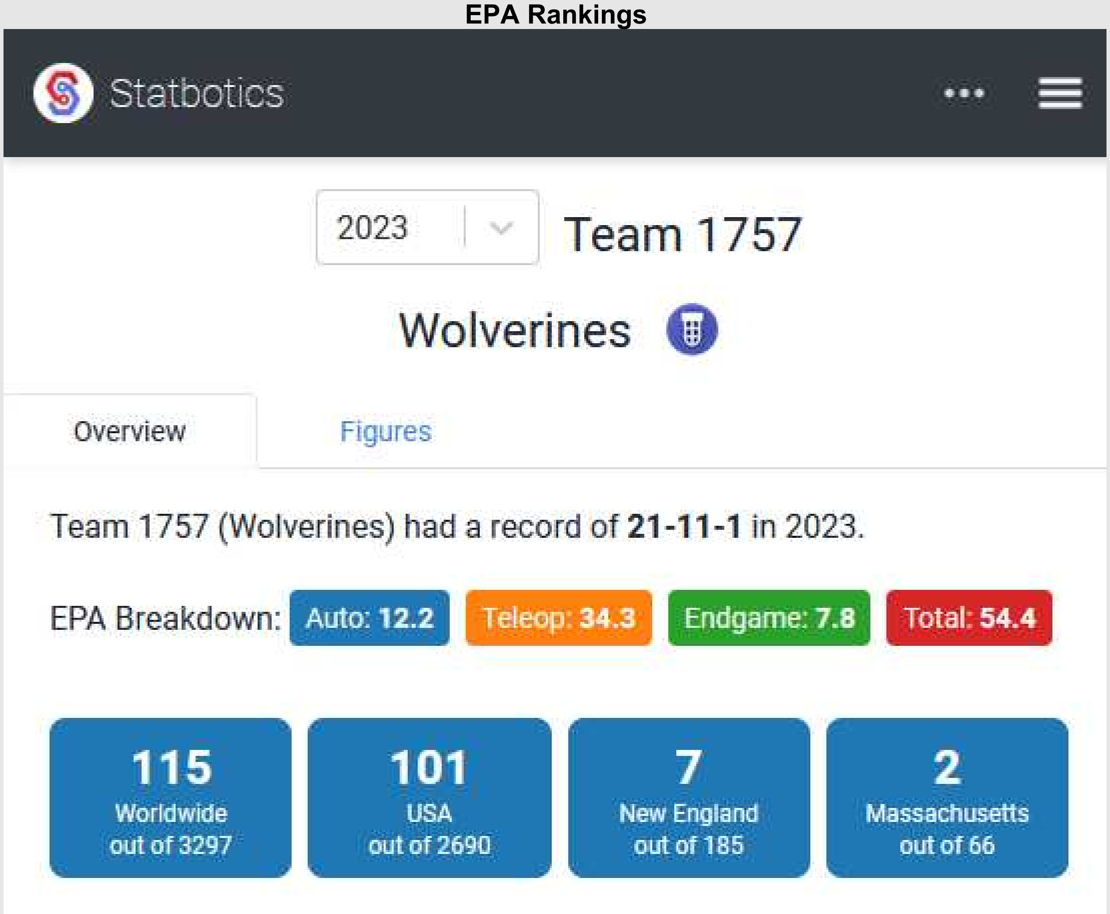
22-11-1

Greater Boston District - Deans List Semi-Finalist: Sean Tao

WPI District Event – Winners

WPI District Event – Enginnering Inspiration Award Winnerd

EPA Rankings



World Championship Update
We thought our seson couldn't get any better then taking home the teams first ever blue banner at WPI. We were wrong. We came into Wilson Division at New England Championship a solid middle of the pack Contender, however we quicky proved why we were there, our robots consistent and Reliable play led us to take #1 overall at the end of Qualifications, after picking the highest rated offensive bot on the field 176 Aces High, we picked up 1699 Robocats to round out a great alliance. We went undefeated in the Wilson Divison playoffs, taking home another blue banner before taking on the Mier Divison winners for the New England District championship. With the Championship Tied 1-1, we went into a nail-bitting sudden death match where we came out on top.

Please review our OA thread on Chief Delphi for more details.

Competitive Record Though District Championship Play:
39-13-1
Greater Boston District - Deans List Semi-Finalist: Sean Tao
WPI District Event – Winners
WPI District Event – Enginnering Inspiration Award Winnerd
NE Championship – Wilson Divison – Winners
NE Championship – Wilson Division – Excelence in Engineering
New England District Championship - Winners

EPA Record

TABLE OF CONTENTS

GAME ANALYSIS

Every FRC season starts the same way; we gather together as a team, watch the kickoff stream, then hunker down and break down the game in back-to-back 8-hour build sessions. The hope is that by the time we walk out the door on Sunday night, we understand the game and know what we are doing.

After carefully considering the different ways you can score points, we concluded that placing GAME PIECES on the NODES was the most critical ability in this game, with it having the highest potential points available. Without the ability to DOCK and ENGAGE, however, it will be virtually impossible to remain competitive due to the lack of ranking points.

After two days of deliberation, these are the design Requirements we settled on.

DRIVE

- Need to be a Small Bot – The smaller the bot, the easier it is for 3 robots to balance on CHARGE STATION
- Need a low center of gravity
- Need to be able to drive and balance on the CHARGE STATION.
- Preferably autonomous balancing on CHARGE STATION
- Use of vison (April Tags) to provide feedback to the onboard odometry system
- Use of vision to identify and seek out game pieces on the field.

ARM

- Arm needs to be strong and durable
- Use Encoders on the input and output of gearboxes to monitor and minimize backlash.
- Either 2 or 3 Degrees of Freedom Further testing will be needed.
- Needs to score at all 3 levels BOTTOM, MIDDLE and TOP Nodes.

INTAKE

- Quickly acquire GAME PIECES (Touch It – Own It)
- MUST pick up CONES and CUBES from the LOADING STATION
- MUST pick up CUBES and upright CONES from the ground.
- Would like to be able to pick tipped-over CONES from the ground.

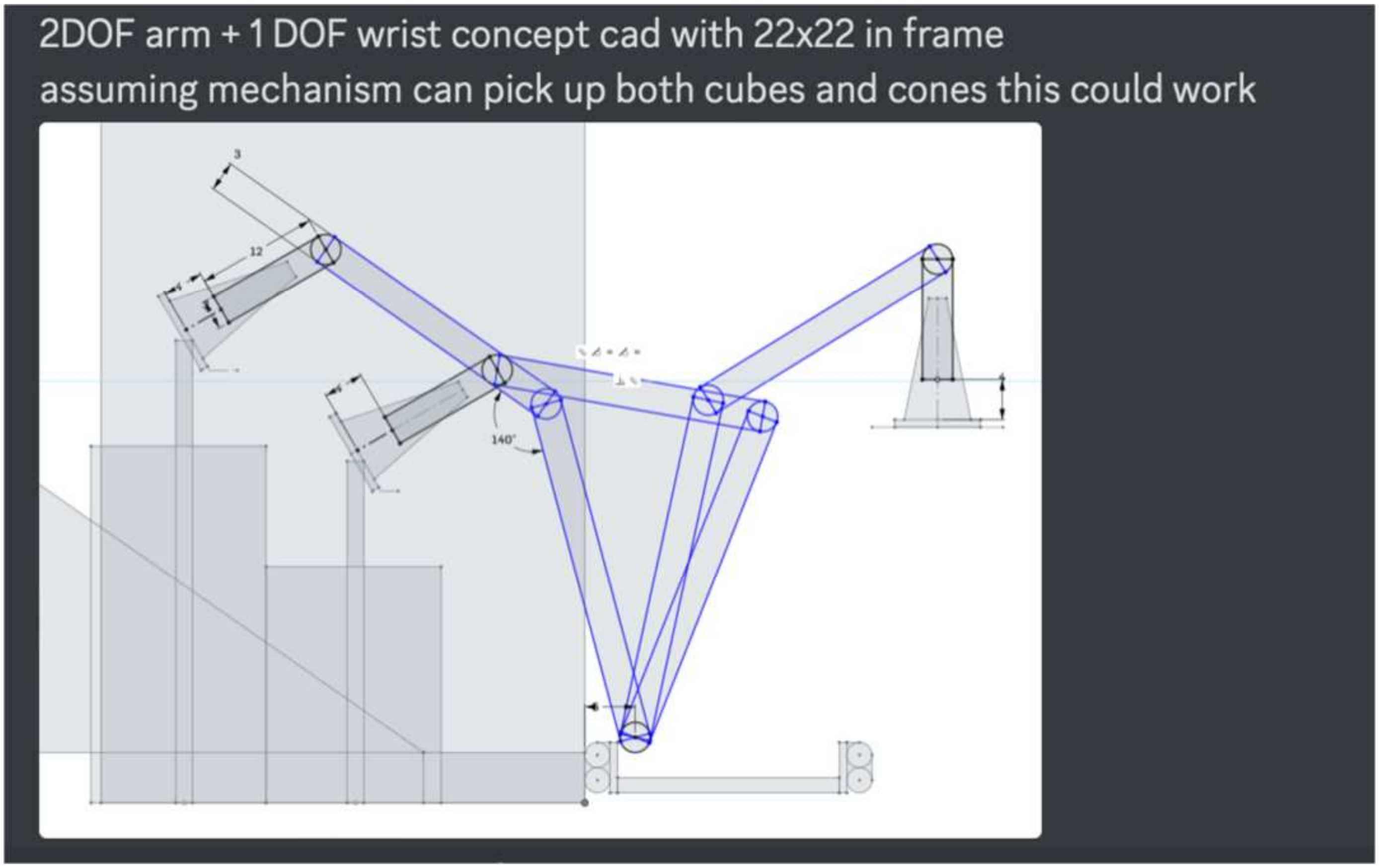
GENERAL DESIGN CONCLUSIONS

Our overall goal for the season was to be a competitive bot in district-level play and qualify for New England Championship. To accomplish this, we need to, at the bare minimum, make it to Elimination at both our district events, hopefully as an Alliance captain or 1st pick.

We approached our design as trying to build a highly reliable jack-of-all-trades bot, focusing on gaining one of the two performance-based ranking points in either match.

Inspired by the cost-effective production strategies of the Hass Formula 1 racing team and our limited team members and design resources, we prefer to use pre-engineered solutions wherever possible to focus our design resources on critical complex components.

IDENTIFYING DESIGN CONSTRAINTS

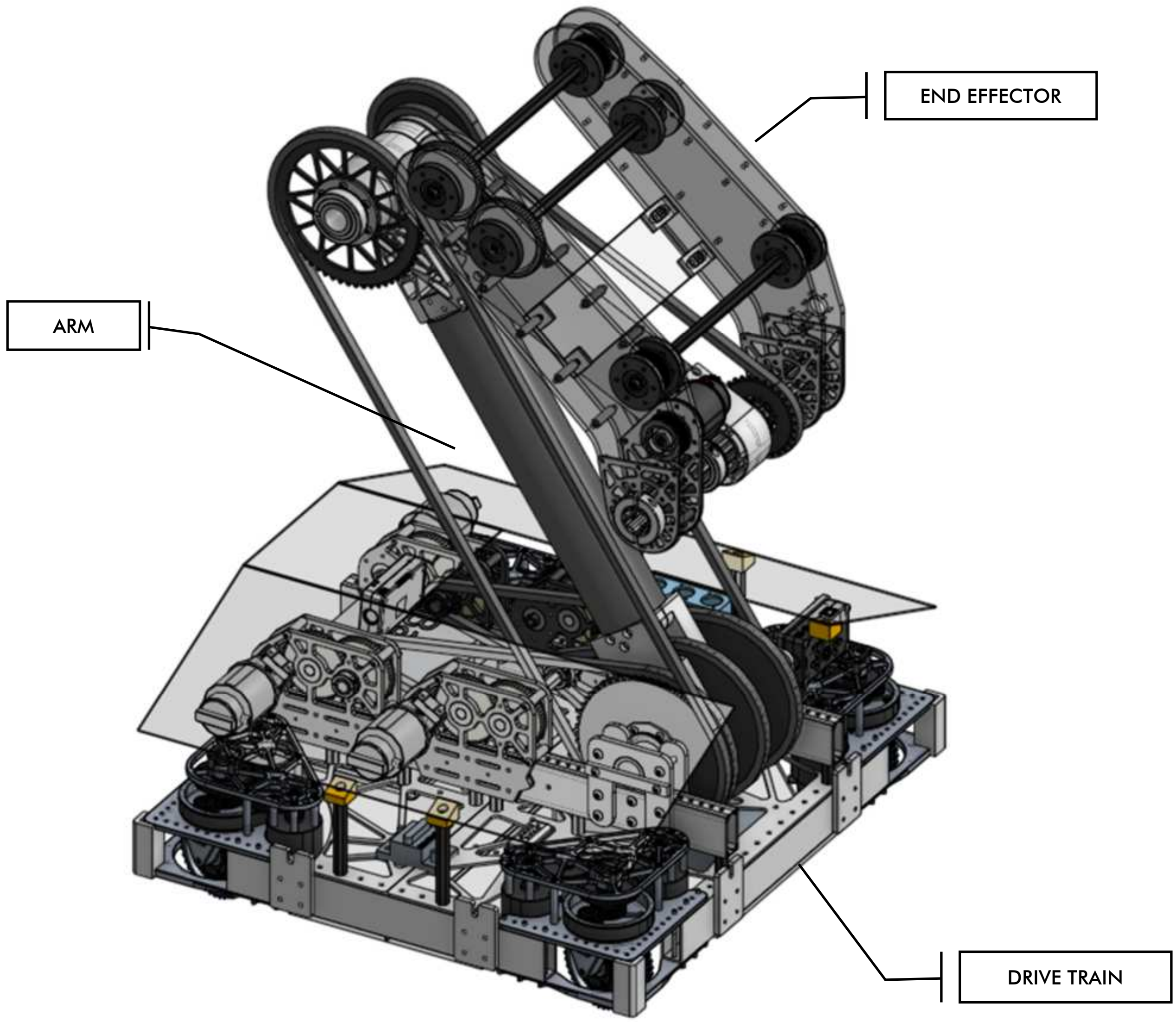


We are thinking about using an arm as a manipulation mechanism. We potentially envision a 2DOF arm + 1 DOF wrist that can pick up both cubes and cones, with a high range of motion on the wrist joint. As we can utilize the bot's movement, we do not need the Arm to move from side to side. An important note is that with an arm the starting configuration poses a good challenge, as it will need to fit inside of the robot's frame before activation. We have found that the shoulder joint only needs to move 90 degrees max, the elbow joint 210 degrees, and the wrist joint somewhere like 270 (at least in the configuration, lots to play with) to achieve all necessary motion.

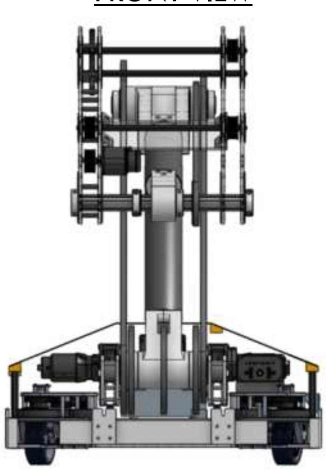
THE 1757 RAPID DEVELOPMENT MODEL

- Clearly Identify the design requirements of the system
- Design and Build a prototype that can be used to test design assumptions and Test
- Use what we learned from testing to develop a final design
- Fabricate final version and intergate into overall robot systems

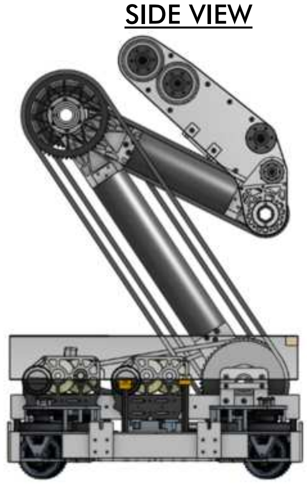
Final Robot Design



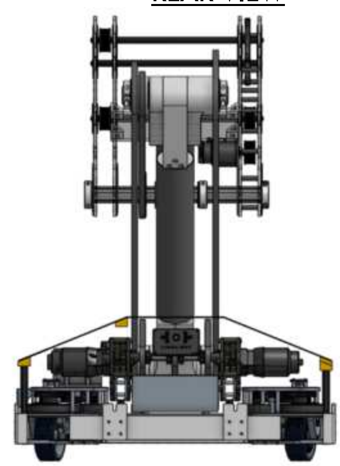
FRONT VIEW

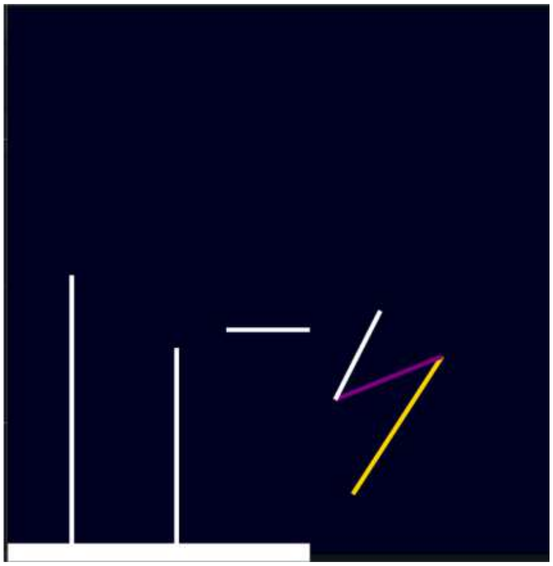


SIDE VIEW

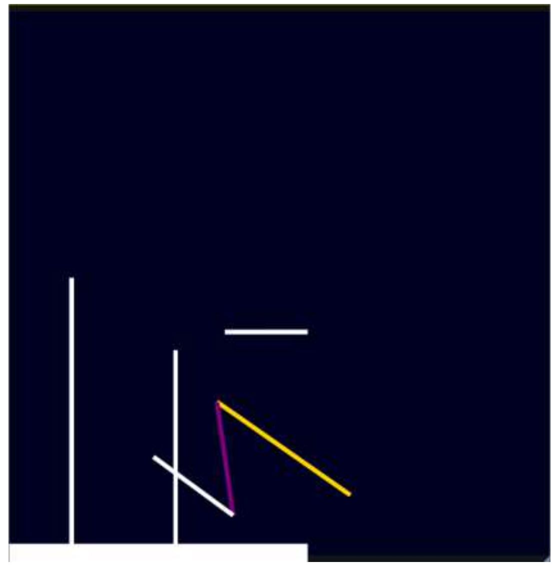


REAR VIEW

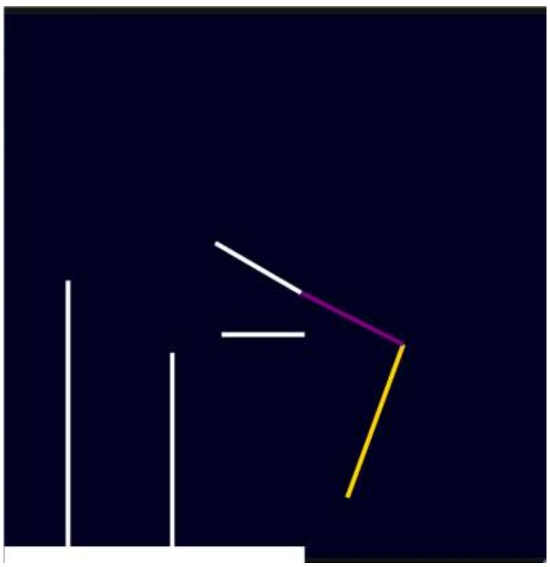




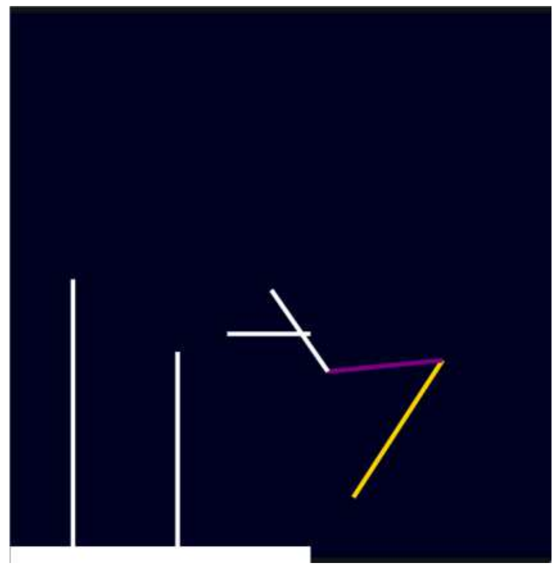
Default Configuration



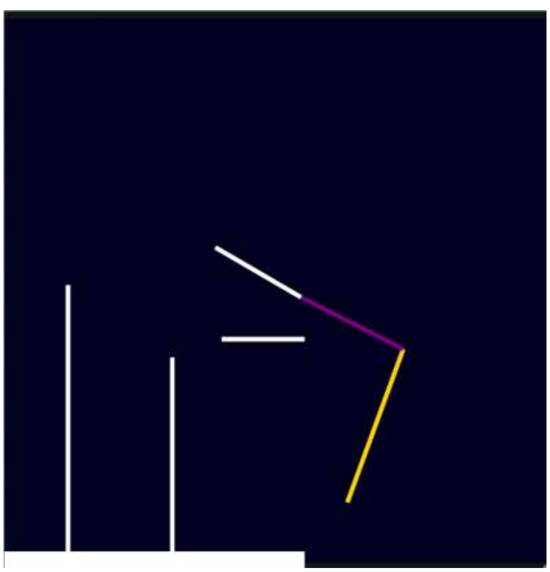
Floor Pickup



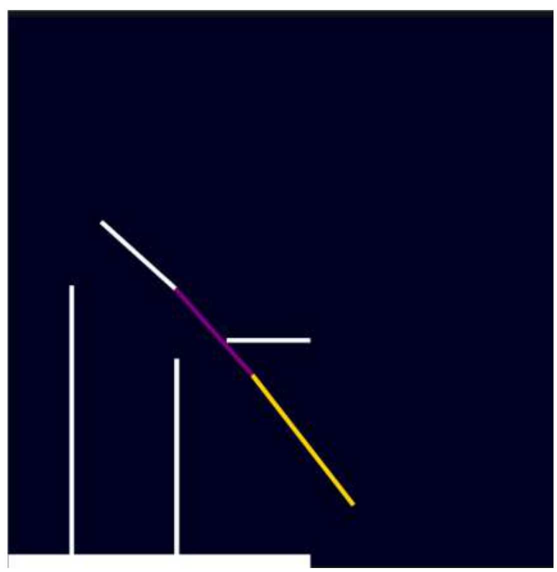
Substation Pickup



Score – Bottom



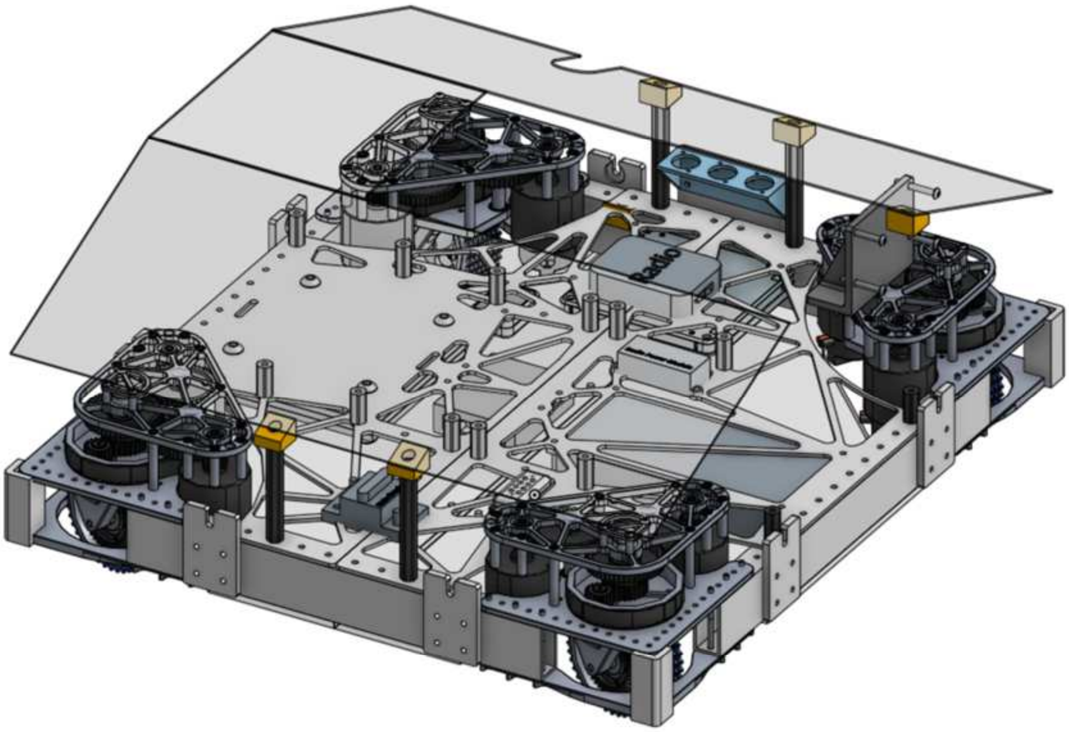
Score – Middle

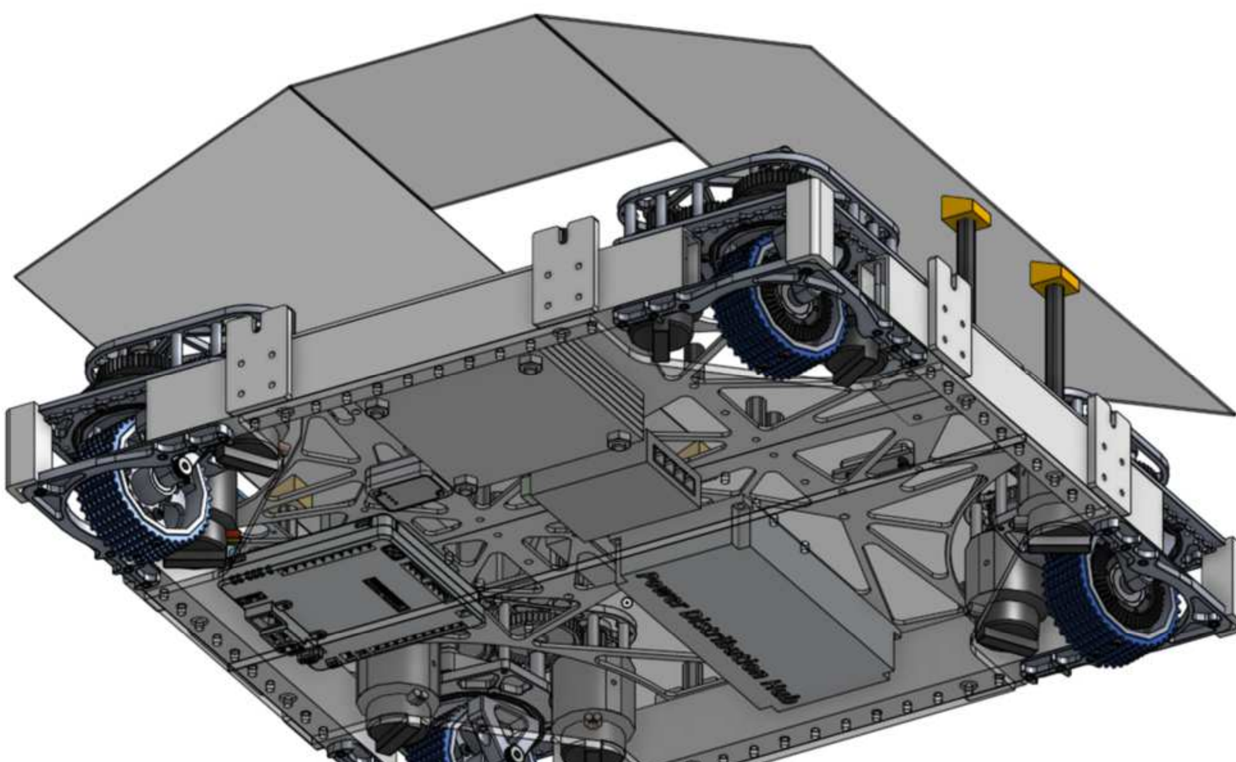


Score - Top

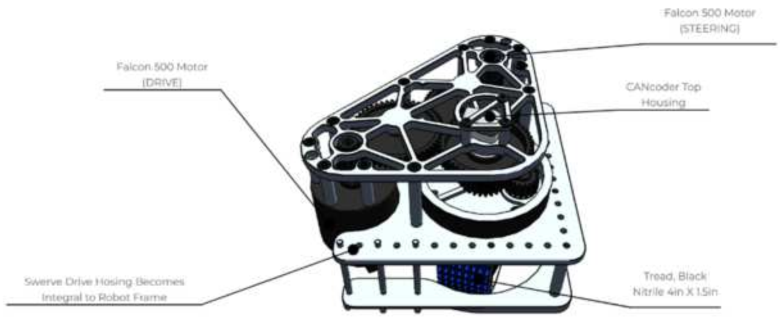
Above – 3D Simulation of Arm Joints in all of its various Arm Configurations

MAJOR SYSTEM #1: DRIVE TRAIN





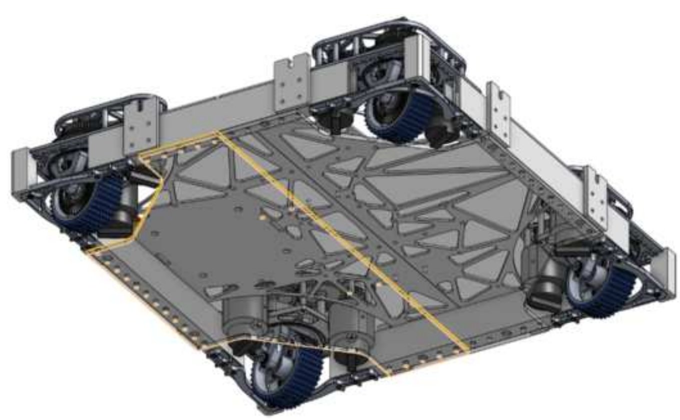
1.1 - SWERVE DRIVE MODULES



Last season was our first competitive season using swerve drive and we could not be happier with the results, because of the equal power between steering and driving there are none of the performance trade offs inherent in other drive systems. We can still run circles around the field when we need to and we can still push another robot across the field when they are in our way. One of our favorite features exclusive to swerve drive is what we call the park feature, by turning all 4 wheels to a 45° angle relative to the corners of the robot the robot effectively parks itself in place and wont move, another robot can push against us all match long and we wont move. Last year we used Swerve Drive Specialties Mk4 units, and this season we upgraded to the newly released Mk4i units. This revised design points the motors downward into the bot instead of mounting above the module. This allowed us to eliminate ≈2" of vertical space in our robot between the drive frame and the major systems.

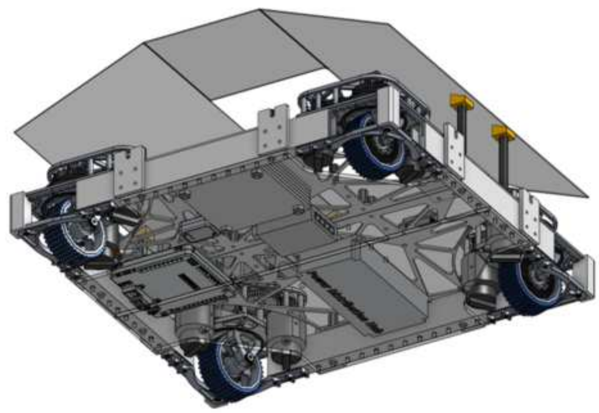
SDS MK4i Swerve Modules	
Powerplant	Falcon 500
Gearbox Configuration	L2
Overall Gearbox Ratio	6.75 : 1
Unadjusted Free Speed	16.3 ft/sec

1.2 - ROBOT FRAME



Left – CAD – Isometric Top View of robot frame
Right – CAD – Isometric Bottom View of Robot frame

1.3 - ELECTRONICS SUBSYSTEM



One thing we struggled with on our 2022 robot was how inaccessible most electrical components were. On our 2022 robot, all the electronics sat in a belly pan at the base of the bot, and the only way to access most of the components required you to remove the majority of the Robot Systems. Inspired by another team's design from 2022, we decided to hang all of our electronic components upside down and face the ground. Now to access the electronics, we tip the bot on its side, remove the ¼" protective polycarbonate plate, and you have full easy access to all the electronics components. (Credit to Team 125 for the Idea, they have the thanks of a grateful drive team and Pit Crew)

ELECTRONICS SYSTEM MAJOR COMPONENTS

- (1 ea) National Instruments - RoboRio 2
- (1 ea) REV Robotics - Power Distribution Hub
- (1 ea) Navex 2 – RoboRio MXP expansion Board
- (1 ea) CTRE CANivore
- (1 ea) CTRE CANdel
- (1 ea) BrainBoxes – SW-015 5 Port Gigabit Switch
- (1 ea) Generic Passive POE Injector
- (1 ea) Limelight 2 Camera
- (12 ea) Falcon 500 Motors
- (1 ea) REV Robotics Sparkmax brushless motor controller
- (1 ea) REV Robotics NEO 550 Brushless DC Motor
- (1 ea) Open Mesh Access Point [Insert Model Number]

1.4 - TESTING PORTS

We added a convenient patch panel to the upper side of the robot to allow for quick access to essential data ports when we don't want to access the underslung electronics.

PATCH PANEL SLOT 1 – USB TYPE A

This slot connects to one of the USB Type A ports on the RoboRio. This typically has a USB flash drive plugged in. During a match all the system logs are copied to the USB drive. After a Match, the USB drive can be pulled and opened up in AdvantageScope on the debug machine for post-game analysis. It's our version of a Blackbox on an airplane.

PATCH PANEL SLOT 2 – USB TYPE B

This connects to the USB Type B Port on the RoboRio—a redundant method for tethering the robot for control and debugging at events.

PATCH PANEL SLOT 3 – RJ45 CONNECTOR

This connects to the Ethernet Switch Via CAT5e for network access. Used for tethered connections to the bot during testing. Ethernet tethering is preferred, but we have encountered software reliability issues in the past.

DCMP Update

At the Revere District event we ran into serious problems tethering to the robot via ethernet and via USB B. we traced the ethernet tethering problem to a problem with the network configuration issue on the driver station laptop. We were unable to determine a definite cause of the USB-B connection issue, but, we think it most likely to to poor quality of the 90° usb connector used on the robot. From that point on we connect a USB-B cable directly into the port on thr

At the same time we realized we needed a button to manually put the arm motors in coast mode for serviceability when the bot is not connected to the driver station. Since we are no longer using the USB-B testing port we replaced it with a momentary push button switch.

1.5 – CAMERA/VISON SYSTEMS

LIMELIGHT 2 – CAMERA

We are utilizing a Limelight 2 Camera for a variety of tasks on the robot mostly devoted to sensor fusion and automation of systems using computer vision. The Limelight's field of vision (FOV) is essentially parallel to the floor and at the height of the April Tags.

Please refer to the Software section of this document for more information on how we use the limelight and April Tags to improve the Onboard odometry of the robot.

1.6 – COUNTERWEIGHT

Not originally intended as part of the design, upon testing of the robot with the Arm fully extended in the scoring position, we realized that robot was prone to falling forward. To resolve this, we looked to add ballast to the bot. First we thought of lead but didn't want to deal with the potential health risks of improperly encapsulated lead. We investigated tungsten; however, a review of the current price of tungsten plate (≈$40/kg) quickly ruled it out as a potential candidate. We settled on 6" x6" x1/4" steel plates mounted directly under the robot battery. After testing with different #'s of the plate, we decided on 6 Plates with a total weight of ≈25 lbs. Now the bot is highly stable even when the Arm is fully extended.

1.7 – PROTECTIVE COVERS

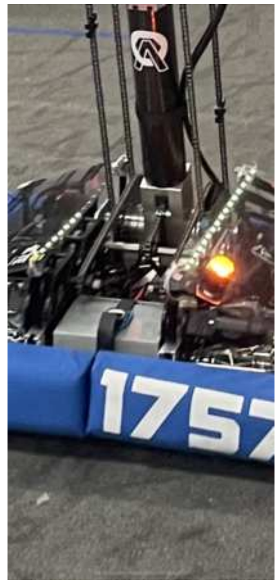
We added protective covers that Slope away from the central arm structure down to the bumpers. Not only do these plates provide a valuable location to display all of our great sponsors they serve to prevent errant game pieces from getting stuck inside the robot during a match.

DCMP Update

Originally the protective covers were only hled on with 3M™ Dual Lock™ SJ3560, this material is nice because it very strong but eaisily removable. Durring qualifying matches in Revere however, these panels kept falling off and dragging around the fieild. The Dual Lock strips were reinforced with zip ties and these held through all of playoffs in revere, and all of qualifications at WPI. Then in Playoff Matches we shered off 3 of the metal standoffs holding up the protective covers. We made quick repairs to keep going however prior to DCMP we will be swaping out all the ½ thunderhex standoffs with 1" 80/20 extrusion with hardened bolts for strength.

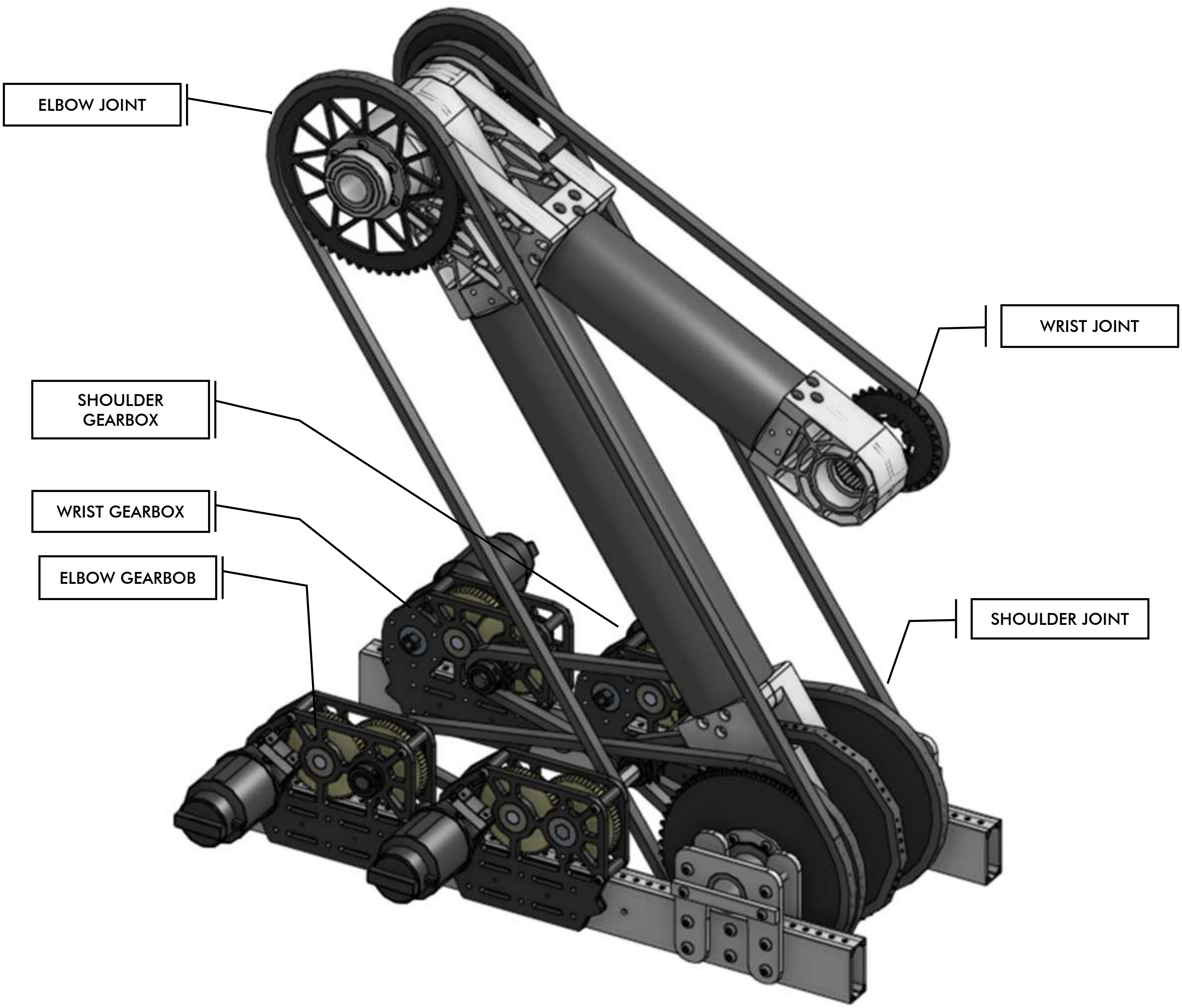
1.8 – GAMEPIECE INDICATORS

One of the hardest things to do in a match is how to signal between the driver station and the human player what game piece you want them to load into the robot. People use Hand Signals, Colored pieces of paper or guess. We wanted to take the guesswork out of the equation, so we mounted 2 LED Strip lights along the top of the protective covers. The driver controls what color these strips are so he can communicate to the human player which game piece to feed to the bot – Yellow for Cones and Purple for Cubes.



Left – Robot Displaying "I Want a Cone"
Right – Robot displaying "I Want a Cube"

MAJOR SYSTEM #2: ARM



2.1- MOTORS & GEARBOXES

To keep the robot's center of gravity low and keep the Arm as simple as possible we decided to locate all of the heavy motors and gearboxes at the base of the superstructure.

GEARBOX # 1 – SHOULDER GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into Custom Gearbox based on WCP Gearbox design.

GEARBOX # 2 – ELBOW GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into COTS WCP Single Speed Gearbox.

GEARBOX # 2 – WRIST GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into COTS WCP Single Speed Gearbox.

2.2 - CHAIN DRIVE

Using a combination of dead and live axels we transfer the power of the gearboxes up though the Arm to power each of the individual joints. For Reliability and durability, we chose to use #35 roller chain rated for 11,000 lbs of force.

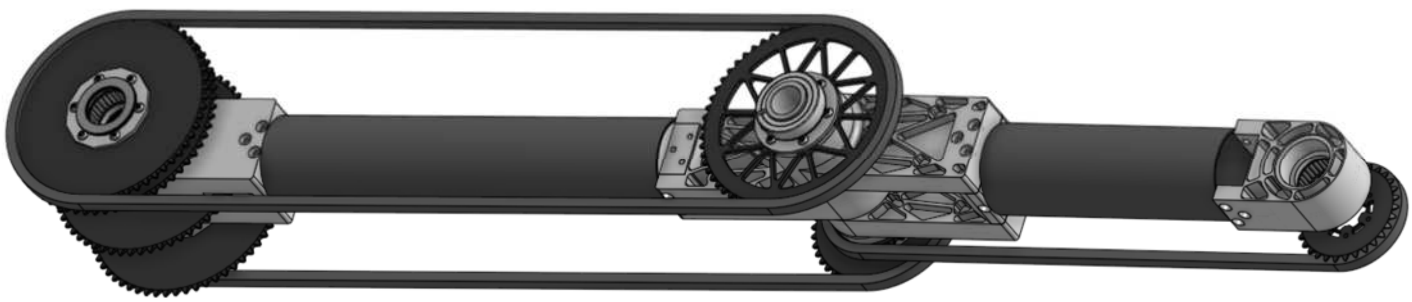
Below is a summary of the different chain runs on the Arm

CHAIN DRIVE 1 – SHOULDER
Shoulder Gearbox Output 12t Sprocket → 60t sprocket on Shoulder (Dead Axel)

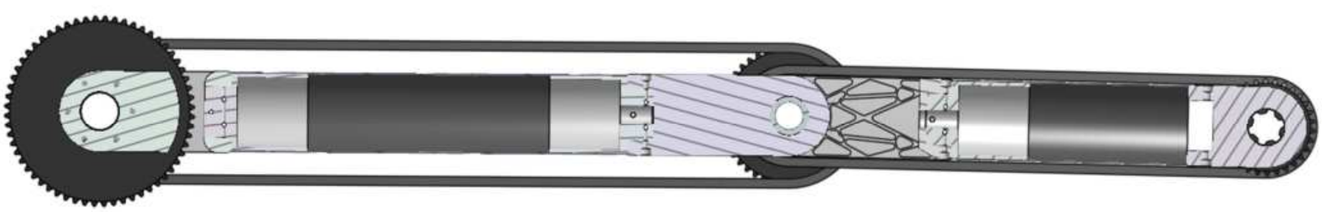
CHAIN DRIVE 2 – ELBOW
Elbow Gearbox Output Shaft 12t Sprocket → 60t Sprocket on Shoulder (Live Axel) → 70t Sprocket on Elbow (Dead Axel)

CHAIN DRIVE 3 – WRIST
Wrist Gearbox Output Shaft 12t Sprocket → 60t Sprocket on Shoulder (Live Axel) → 60t Sprocket on Elbow (Live Axel) → 40t Sprocket on Elbow (Live Axel) → 32t Sprocket on the Wrist (Dead Axel)

2.3 - ARM STRUCTURE



Above - 3-D View of Outstretched Arm



Above - Section View Though Center of Carbon Fiber Arm

CARBON FIBER ARMS

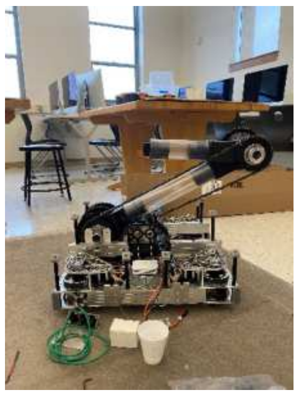
We chose to use carbon fiber tubes as the main structure of the Arm due to its strength and lightweight, the more weight we could save on the Arm the lower we could push the robot Center of Gravity. Carbon Fiber tubes are a stock McMaster item 3" Ø. Carbon Fiber is Epoxy bonded to 3" hollow aluminum plugs bolted to the aluminum joints.

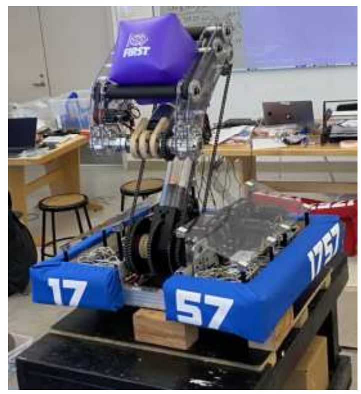
Left – Final carbon fiber arm links after final glue up

Right – Final aluminum plugs used in the ends of the carbon fiber tubes

3D PRINTED AND POLYCARBONATE PROTOTYPES

Because we knew the carbon fiber and machined aluminum would take time and money to manufacture, we heavily used 3D-printing to make prototypes of the Arm and test and confirm critical geometry before placing final fabrication orders. These prototypes are two fragile to be used on a competition bot but worked well for their intended purposes. We learned very Important lessons about where the concentrations of forces were along the axels and what parts needed reinforcement.





Left – 3D printed Prototype of the wrist joint, printed on a FormLabs 2 SLA Printer

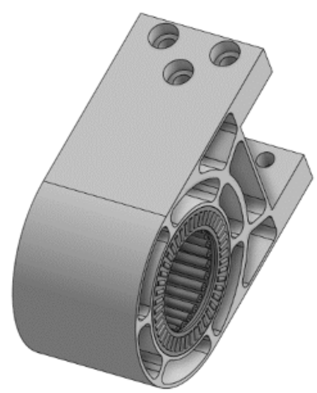
Center – Polycabonate Prototype arm Mounted on bot for the First Time

Right – Fully Assembled "Alpha" Robot build

2.4 – JOINT STRUCTURE

CAD - Shoulder Joint

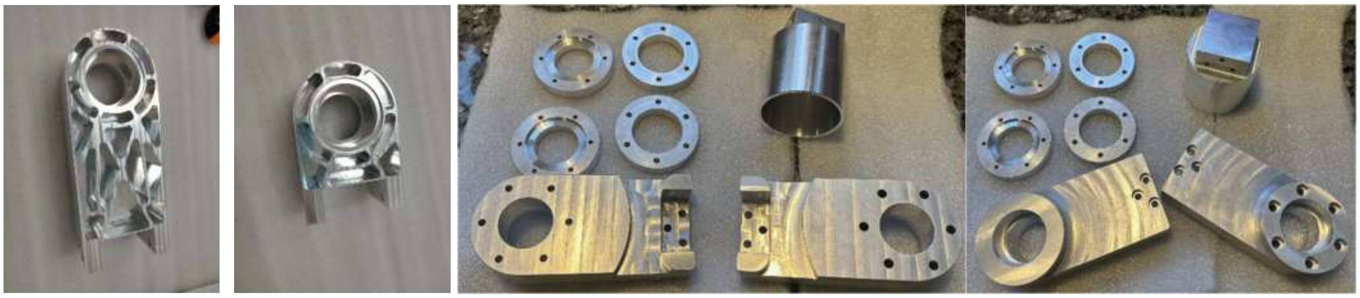
CAD - Elbow Joint



CAD - Wrist Joint

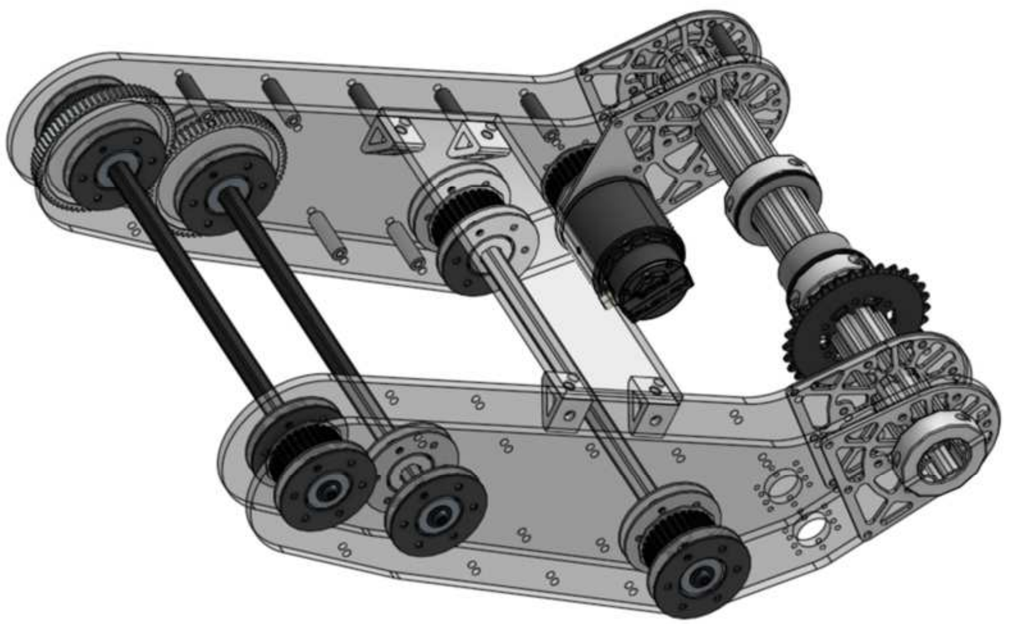
NEEDLE AND THRUST BEARINGS

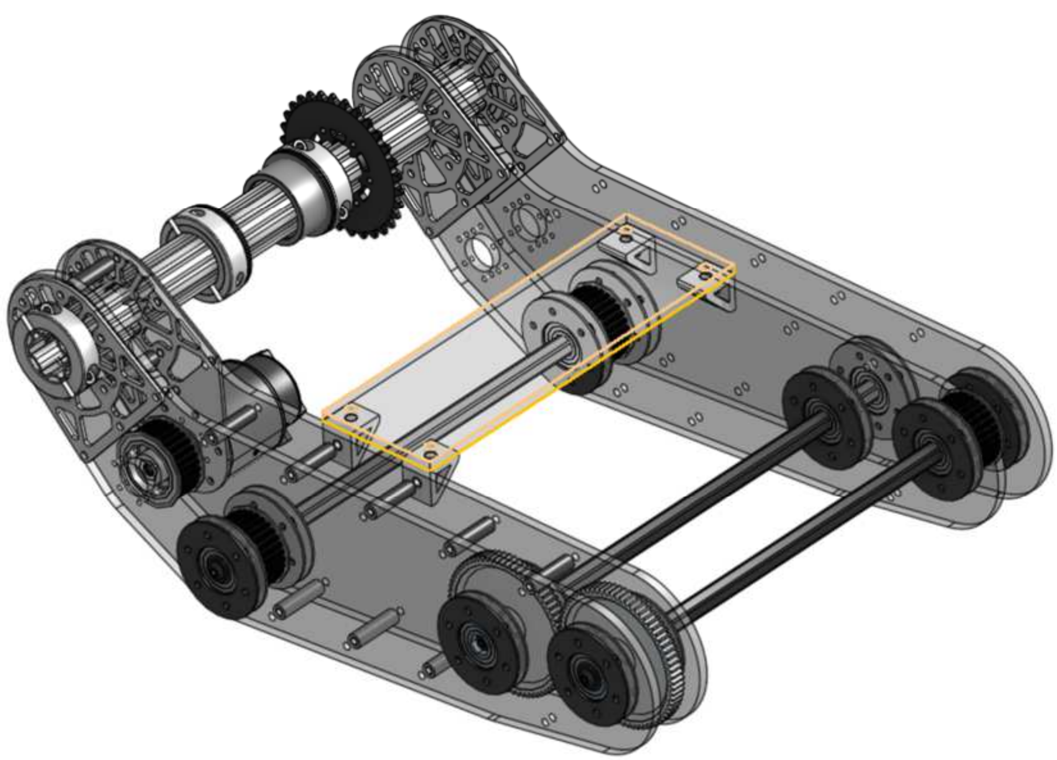
Used in All three joints to allow for smooth rotary motion in each joint.



Photos - Final Machined Parts

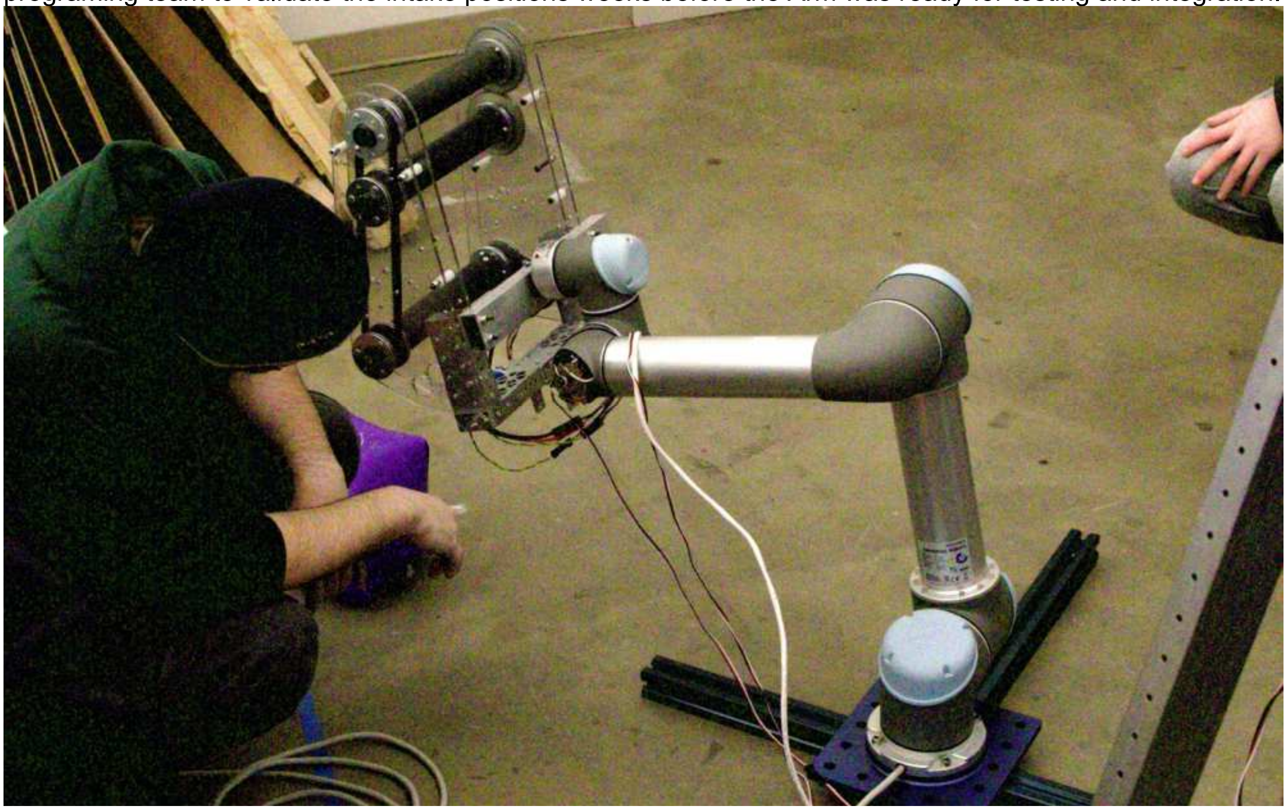
MAJOR SYSTEM # 3: END EFFECTOR





INTAKE PROTOTYPING – FUN WITH INDUSTRIAL ROBOTS

We had a lot of fun prototyping this mechanism, since it was the first major mechanism that we finished we had lots of time to put it trough its paces. One of our mentors has access to a Universal Robots UR3 robot and brought into our lab during one of our weekend build sessions (see images below). This actually proved to be essential because it allowed our programing team to validate the intake positions weeks before the Arm was ready for testing and integration.



3.1 – ROLLERS

The rollers we are using are Vex VersaHub Rollers with ¼" neoprene tubing stretched to cover them, they are very grippy and hold the cube verry securely. We started with the dimensions of the everybody roller for our prototype then made some modifications before settling on final separation differences. The neoprene tubing is undersized for the OD of the polycarbonate roller. We learned a fun trick to clamp off one end of the neoprene tube and inflate it with an air compressor to stretch it over the polycarbonate tube, when the air is released, it makes a perfect friction fit between the Neoprene and the polycarbonate. We have had no detectable slippage after weeks of testing with the rollers.

DCMP Update

After 33 competitive matches one thing is clear, we have problems picking CUBES up off the floor and in order to maintain our competitive edge at DCMP we know we need to be able to get CUBES up off the floor. We thing the majority of the problem is related to how narroe our end effector is, the original design was only 1” wider than the width of the CUBE. To aliviate this issue we are planning to widen the end effector by 3”.

3.2 – MOTORS & GEARBOXES

The intake is powered by a REV Robotics NEO550 Brushless motor into a REV Robotics Ultraplanetary gearbox. The motors small size is nice however because we mounted the Sparkmax Motor controller on the intake as well there is no significant weight savings compared to using a Falcon 500 with an integrated Talon SRX. We may end up swaping this out for simplicity sake in the future.

REV Ultraplanetrary	
Powerplant	NEO550
Gearbox Configuration	4:1, 5:1
Overall Gearbox Ratio	20:1

SOFTWARE

SOFTWARE: OUR DEVELOPMENT ENVIRONMENT

WPILib

The perineal stalwart, we still rely on core elements of WPIlib for robot communications and debugging. WPILib's new Logging features have greatly enhanced our Debugging capabilities

RobotPy

We have found that students have a lot easier time learning python then they do Java or C++ so with the growing support for RobotPy we migrated our Codebase from Java to Python in 2020. As of this March we are an official contributor to the RobotPy project

GitHub

Without Github our level of remote work and collaboration just wouldn't be possible.

SOFTWARE: NEW AND UPDATED TOOLS THIS YEAR

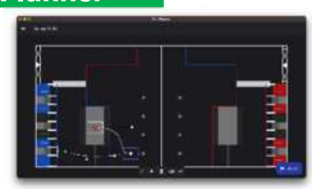
AdvantageScope

WE LOVE ADVANTAGE SCOPE! Not only does it log *everything* but does it in away that is intuitive and easy to review. No more searching though 10000 lines of log files to find the one piece of information we need. Huge thanks to team 6328 for building such a great tool.

PhotonVision

We are using PhotoVision as our native development framework for Computer vision due to its growing wide support inside the FRC community. It does not include native support for RobotPy however so as an offseason project our lead programmer wrote a custom wrapper for PhotonVivion so it can work inside our RobotPy environment

PathPlanner



Last year we used PathWeaver, but we were disappointed in the lack of native support and increased complexity in the development stack so starting with the off season we transitioned all of our Autonomous path planning to PathPlanner. We had much fewer issues with this system.

SOFTWARE: DRIVE

Taking off of last year, the drivetrain codebase has stayed the same. We are running field oriented drive with robot relative rotation to allow for quick maneuverability. A button to align to the nearest 90 degree angle was added to help with driver alignment. This state slightly reduces the speed and snaps the angle of the robot in order to have perfect alignment to the double substation, single substation, and grid every time. For our automated balance sequence, we work in robot relative space on the robot relative gyro.

A Brief Tangent - Absolute Relative Drive

Last year our lead programmer had a new idea for drive control, an absolute relative drive. The common swerve drive control method was to have a field relative translation for the bot, and a robot relative rotation. What this meant is a left input on the rotation axis would result in the robot rotating to the left at a constant speed. A translation action was not affected by rotation but instead was in "field relative" space. The difference of absolute drive is that the rotation is also field relative. A left input on the rotation stick will yield the robot turning to face left. This year we expect this type of robot control to be very important for drivers when they have to be able to turn to specific positions for collection and scoring on swerve drives. You can see this in action in any one of our videos from last year. Having fixed controlled rotation will allow for precise driver input and less fiddling with controls when cycle time is very important.

The drivers have also experimented with alternate driving methods on swerve to get used to interesting control schemes such as a curvature drive, standard tank drive, standard field relative drive, and full robot relative drive.

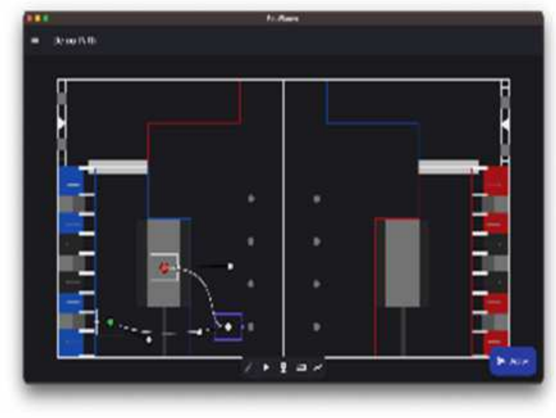
SOFTWARE: INTAKE

The intake is using a state machine in order to regulate its expected behavior. There are 2 enumerable values: one for the gamepiece intended direction (intake, outtake, hold) and one for the desired gamepiece type (cube, cone). The transition between each state is dictated by a user input to any given category. If no input is given, the system holds its position and keeps the desired gamepiece remembered. The state value of the desired gamepiece is displayed to the driver and to the human player through pulsing leds of the respective color.

SOFTWARE: ARM

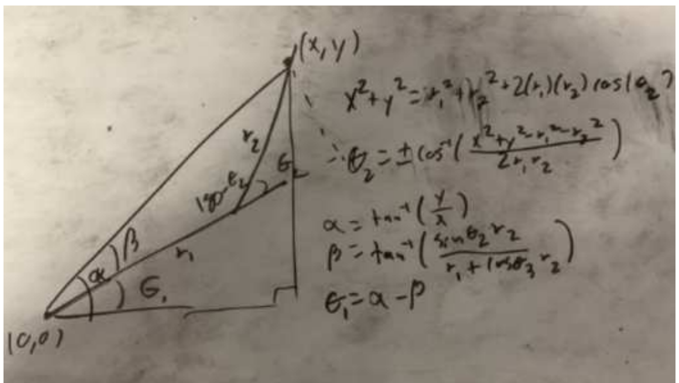
A triple jointed arm is no easy feat in order to program smoothly. From cad, states are given about the end effector's desired position and rotation relative to the floor. From there we use inverse kinematics to determine each per-joint relative rotation at any given position. A cartesian control on the wrist joint's position is added using a trapezoidal PID profile to lay out a path for the Arm to follow. For the wrist it has another trapezoidal PID profile controller. When going state per state on the Arm's motion, we check for if the relative angle goes over software end limits in order to prevent running the Arm into itself. These are done in joint relative space. Since the Arm is controlled from the base through chain and sprockets there is a virtual 4 bar created in which the rotation of any given joint is given relative to the ground. These are converted into motor space and passed onto each motor where they have a position PID controller onboard. For simulating the system we are using mechanism2d to view the expected values of the Arm and be able to run through positions. This simulation first approach has allowed minimal revision and a solid foundational codebase that is mostly complete before the bot is finished. Furthermore the position of the Arm is logged in 3d with advantagescope based on the position it believes the bot is in. Logging was also a priority for a complex system and therefore we log the instructed position and actual position per joint and each end position in cartesian space.

SOFTWARE: AUTONOMUS



We experimented in the offseason with pathplanner and use it extensively for our autonomous. Each necessery command is placed into a potential pool of events for pathplanner to fire. At the beginning a sequence determined solely in pathplanner is fired. Going off of last year we had a large time gap in order to make slight changes so instead for this year we are using the waypoint system and opting to have any given autonomous contained solely in pathplanner. This has increased our ability to construct autos and tweak any given aspect as needed. For the library itself of parsing, the lack of first class robotpy support meant we had the freedom to reimplement autonomous however we pleased based on the path. We follow a changing trajectory and the swerve drive using onboard odometry and a weighted vision estimate determines its bot position relative to the global field and follow through it between each section.

SOFTWARE: SIMULATION



armsubsystem.py

```
    def setEndEffectorPosition(self, pose: Pose2d):

        twoLinkPosition = Translation2d(
            pose.X() - constants.kArmwristLength * pose.rotation().cos(),
            pose.Y() - constants.kArmwristLength * pose.rotation().sin(),
        )

        endAngle = math.acos(
            twoLinkPosition.X() * twoLinkPosition.X()
            + twoLinkPosition.Y() * twoLinkPosition.Y()
            - constants.kArmtopLength * constants.kArmtopLength
            - constants.kArmbottomLength
            * constants.kArmbottomLength
            / (2 * constants.kArmtopLength * constants.kArmbottomLength)
        )

        startAngle = math.atan2(twoLinkPosition.Y(), twoLinkPosition.X()) -
math.atan2(
            math.sin(endAngle) * constants.kArmtopLength,
            constants.kArmbottomLength + math.cos(endAngle) *
constants.kArmtopLength,
        )
        wristAngle = pose.rotation().radians() - startAngle - endAngle

        bottomArmEncoderPulses = (
            startAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kBottomArmGearRatio
        )
        topArmEncoderPulses = (
            endAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kTopArmGearRatio
        )
        wristArmEncoderPulses = (
            wristAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kWristPivotArmGearRatio
        )

        self.topArm.set(Falcon.ControlMode.Position, topArmEncoderPulses)
        self.bottomArm.set(Falcon.ControlMode.Position,
bottomArmEncoderPulses)
        self.wristArm.set(Falcon.ControlMode.Position, wristArmEncoderPulses)
```

Due to our team's resources, virtual simulation is a huge part of our ability to quickly and reliably construct the bot's codebase. Some key examples of simulation are a wrapped onto a simulated falcon motor. Given our team's extensive use of falcons on the robot, a wrapper that provides simulation support allows for the programming team to iterate much easier and creates a cleaner codebase. Each falcon is logging the values of the motor % and the encoder position, as well as an override value to allow the user to manually in simulation change the value for sensor readings. Entire robot configuration is done on a single call and the getting of velocity, position, and percent and the setting of velocity, position, and percent are easy to access functions to allow interfacing with the motors more accessible than the CTRE library. Given this robot also has a NEO550, the simulation system was adopted to have a similar interface for ease of replacement from a falcon to a motor on the intake. We geometrically derived the inverse kinematics for 3 links with a fixed Pose endjoint. Each of these poses actually allows for two configurations of the proximal 2 arm joints (they can simply be mirrored over the line created from the wrist joint to the shoulder joint, however by forcing the sign on the elbow joint they can all be consistent.

SOFTWARE: VISION

NOW WITH APRILTAGS AND PHOTONLIB

We have a vision system complete with sensor fusion for complete robot localization. Last year,we worked with our first complete vision system as a team that resulted in significantly enhanced system performance, and using apriltags will be very important to account for combined sensor error as well as for being able to reliably use sensor data for automated alignment to various points on the field such as the double substation and the grid.

THE HOW

Photonvision generates camera-relative 3d transforms of each apriltag. Since the position of the camera is known and the position of the apriltag is known, the position of the robot can be determined from a single apriltag datapoint. These transforms are fed into a RobotPoseEstimator in order to create a sense of where the robot could be at a given time, this is combined with the gyro and wheel encoder information to get an accurate sense of where the robot is on the field at any given time. This is used in other subsystems when needed, as well as results being logged to AdavantageScope through the usage of each known pose and ghost poseseepaste

GOING FURTHER

We plan on using this odometry data to have automated alignment in complete robot space for important precision actions such as placement of gamepieces on the grid and collection of those gamepieces. Autonomous will also use this data. Perhaps an automatic engagement on the charge station by using the rotation gained from the apriltags will be possible. Overall having a sense of where the robot is on the field is beneficial to aid in other systems.

ENGINEERING TEAM

WHS Faculty Advisors

James Looney & Raul Madera

Team Captain

Charley Marsland*

Team Business Lead

Sean Tao

Team Technical Lead

Luke Maxwell

Senior Mentors

Dwight Meglan **Chris Aloisio°** **Steve Harrington°**

Mentors

Anthony Gelsomini **Manny Barros°** **Sean Lendrum°** **Mark Holthouse** *Mentor Emeritus* **Amber Maxwell**

STUDENT MEMBERS

Safety Captain		*Graphic Design Lead*	*CAD Lead*
Jacob Kaplan		**Ivan Cai**	**Landon Bayer**
John Santasuoso*	**David Girard***	**Declan MacDonald**	**Liam McWeeney**
Alex Theofilou	**Claire Peng**	**Andrew Gong**	**Rachel Qu**
Baili Jiang	**Nolan folmar**	**Jeffery Pan**	**Melissa Yang**
Divya Siddi	**Vinny Milinazzo**	**David Confoey**	**Adrianna Cirillo**
Elizabeth Lowney	**Jacob Liu**	**Anthony Yang**	**Constantina F.**
Luke Li	**Owen Monahan**	**Erik Curlli**	**Kevin Bai**
Henry Marsland	**Nina Pappas**	**Rachel Lee**	**Hanya Yang**
Sophia Patrick	**Lauren Buza**	**Kaylee Phu**	**Elias Mukdissi**

Westwood Robotics, Inc
Board Of Directors

Steve Harrington° *President*
Catherine Tseng *Treasurer*
Manny Barros°
Cheten Gopal *Clerk*
Jack Tseng

**Denotes High School Seniors* *°Denotes FRC Alumni*

SPONSORS

Heartlander Surgical	Westwood High School	Medtronic	Act Fasteners
Gene Haas Foundation	TE Connectivity	General Dynamics	IRA Motor Group
Bayer Familt	Capital Group	Yamaguchi Family	Plymouth Rock Assurance
Deadham Savings	Maxwell Family	Schneider Electric	Marsland Family
Porier Apliances	Franklin Templeton Investments	Jake and Joes Resturant	McGolf
Roche Bros		Capers Catering	

VENDORS

Andymark	West Coast Products	Vex Pro	Swerve Drive Specialties
Discord	Google Work	Onshape	Robo Promos
McMaster-Carr	The Home Depot	Cross the Road Electronics	MS Visual Studio
Python	GitHub	RobotPy	Java

1757

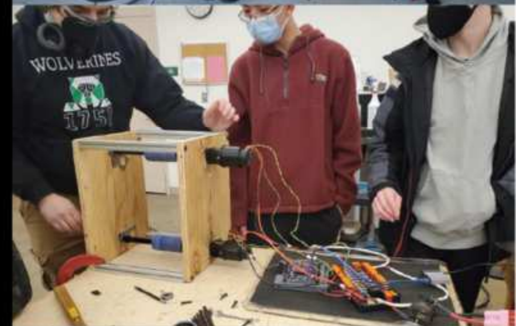
Special thanks to everyone who makes this team possible Including

Our Mentors
Our Students
Our Teachers
Our Parents

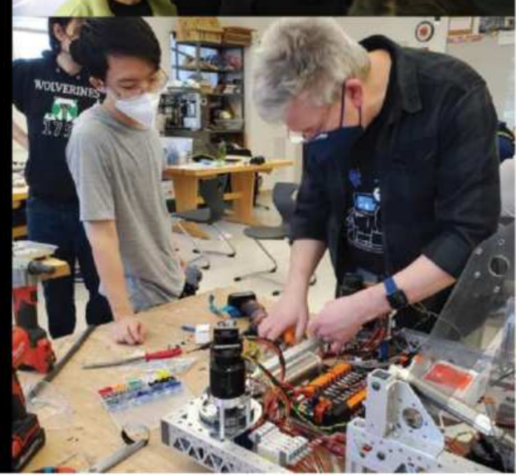
Our Amazing Sponsors

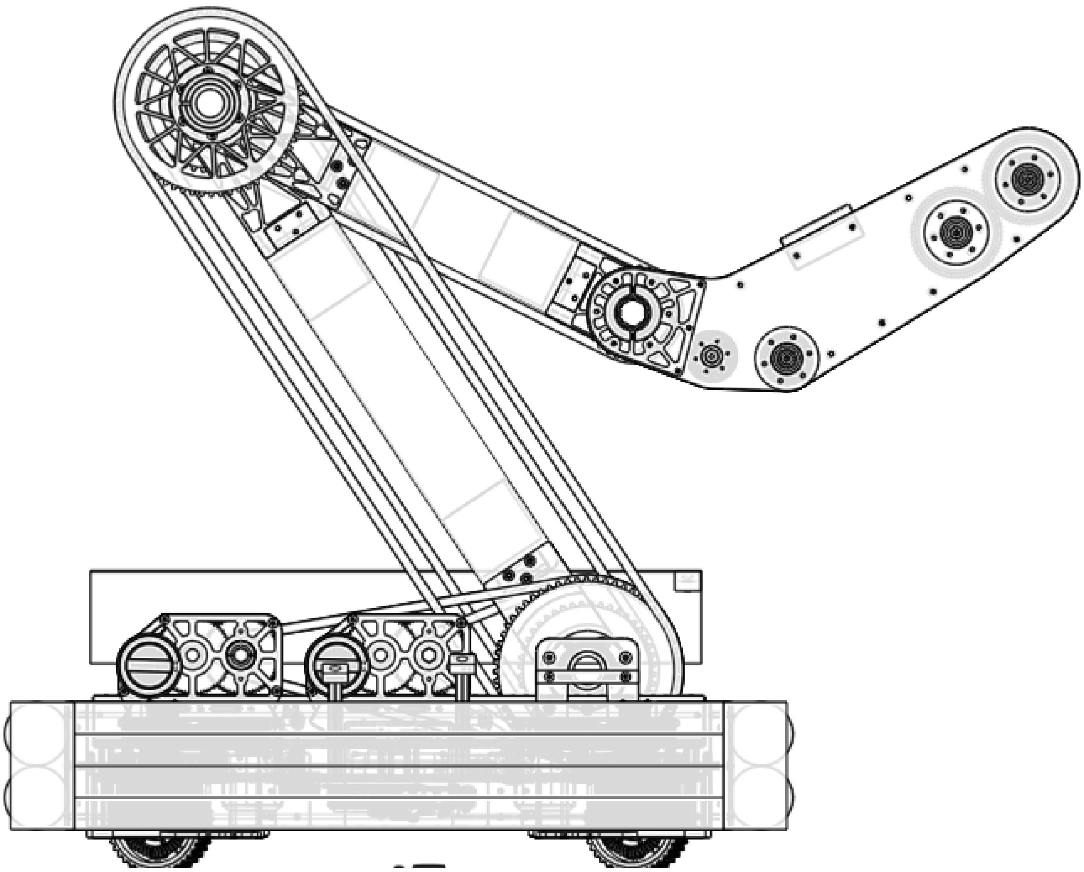
Westwood Public Schools
Emily Parks - Superintendent
Allison Borchers - Asst. Superintendent
Deb Nolan - Building Use Coordinator
Tom Carey - Director of Facilities
Ron Smith - Assistant Director of Facilities
Amy Davenport - Principal
Aishleen Flanagan - Assistant Principal
Tom Millett - Assistant Principal
Andrew Miller - Science Dept. Chair
WHS Janitorial/Facilities Staff
WHS Library Staff
The Open Alliance

#FRC1757
@WWRobotics1757







FRC WORLD CHAMPIONSHIP EDITION

LUXO

WOLVERINES

1757

2023 TECHNICAL BINDER

FORWARD

Hello, and let us welcome you to FRC Team 1757's 2022-2023 Season. This season has continued the tremendous growth in our robot's design and technical ability that started last year as our team emerged from the hibernation of COVID-19 to become a surprising contender in the New England Region. Continuing to recruit rookie students to supplement our now more veteran team members and Senior Mentors, we have pushed our collective talents to their limits to deliver the competition-worthy robot contained within the pages of this binder.

Our season started in the fall of 2022, introducing a new class of over 10 freshmen, sophomores, and juniors to the world of FRC. We showed off the robot at local town events, built a T-Shirt Cannon to raise school spirit at the prep rally, and hosted weekly technical seminars on everything from the engineering process to CAD, Electronics, Pneumatics, Mechanics, and everything in-between. Over the Summer we got a new OMIO X8 bed router and practiced our CAD and fabrication skills by designing and building an enclosure for the machine. We traveled to Billerica, MA in October to compete in the first-ever New England Robotics Derby. We finished in Second Place, losing in the Finals (The best competitive finish in team history). We piled into our classroom on a cold Saturday morning in January, eagerly anticipating this year's game. 4 CAD models, 8 shared Google Drives, ten weeks, 20 Weekend Build Sessions, 50 Zoom calls, 5799 lines of code, 170 git commits, 19,129 discord messages, and many, many cups of coffee later, we are proud to unveil our robot "LUXO" for the 2023 FRC Season.

Why did we name the robot LUXO? Is it because of the shining lights on its frame that illuminate what game piece we are looking for on the field…no. Is it the bright shining future of the team…no. Is it a reference to solar power and how that ties into the theme of this year's FIRST season…good guess, but no. In truth, we are a bunch of animation nerds, and we thought the robot looked like the lamp in the Pixar Animation title sequence named Luxo. Not every robot name has a deep prophetic meaning…sometimes it's just about the memes.

One very exciting thing about this year is that Team 1757 joined the Open Alliance. We found the Open Alliance teams and their open and timely build season updates so helpful to our team last season that we decided to join so we could help other teams the same way the alliance has already helped us. In addition to frequent updates on our build thread, we also made two appearances on the Open Alliance Show Streamed on twitch. If you want to learn even more about our robot and the design process, beyond what is contained in this manual, please visit our Chief Delphi Build Thread at https://www.chiefdelphi.com/t/frc-1757-wolverines-2022-2023-build-thread/416564

We hope you enjoy this brief look at the design process and technical details that went into this robot, and if you have any questions, look for one of our team members in the stands, in the pits, or on the field. We are always ready to share the knowledge we have gained and share a few hard-learned lessons we learned along the way.

DCMP Update

So it has been a whirlwind of a season so far, after meddeling performance at Greater Boston district we went on win the WPI District Event. Not only were we Alliance captian of the the #2 alliance, we also won the Engineering Inspiration award at WPI. Though out this document you will find various updated information featuring design changes/Repair/modifications that were made during the competition season.

Competitive Record Though District Play:

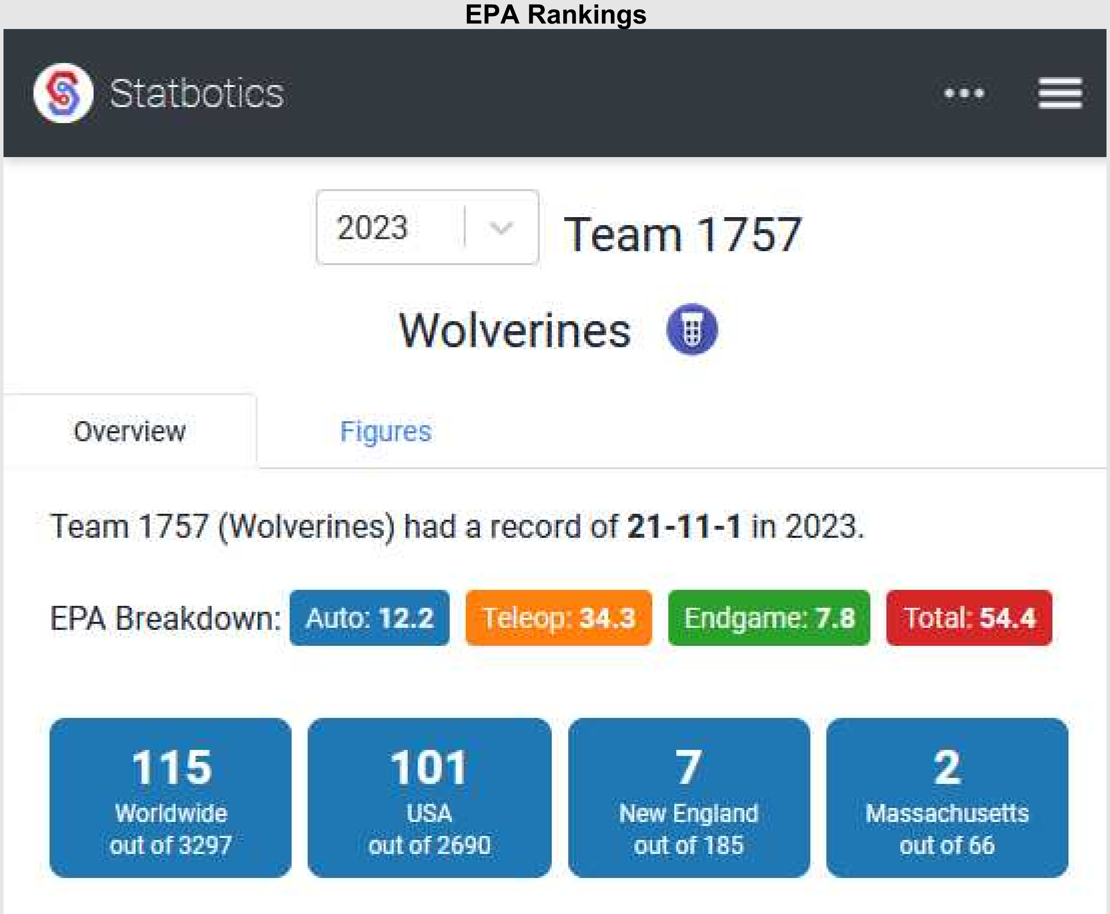
22-11-1

Greater Boston District - Deans List Semi-Finalist: Sean Tao

WPI District Event – Winners

WPI District Event – Enginnering Inspiration Award Winnerd

EPA Rankings



World Championship Update
We thought our seson couldn't get any better then taking home the teams first ever blue banner at WPI. We were wrong. We came into Wilson Division at New England Championship a solid middle of the pack Contender, however we quicky proved why we were there, our robots consistent and Reliable play led us to take #1 overall at the end of Qualifications, after picking the highest rated offensive bot on the field 176 Aces High, we picked up 1699 Robocats to round out a great alliance. We went undefeated in the Wilson Divison playoffs, taking home another blue banner before taking on the Mier Divison winners for the New England District championship. With the Championship Tied 1-1, we went into a nail-bitting sudden death match where we came out on top.

Please review our OA thread on Chief Delphi for more details.

Competitive Record Though District Championship Play:
39-13-1
Greater Boston District - Deans List Semi-Finalist: Sean Tao
WPI District Event – Winners
WPI District Event – Enginnering Inspiration Award Winnerd
NE Championship – Wilson Divison – Winners
NE Championship – Wilson Division – Excelence in Engineering
New England District Championship - Winners

EPA Record

TABLE OF CONTENTS

GAME ANALYSIS

Every FRC season starts the same way; we gather together as a team, watch the kickoff stream, then hunker down and break down the game in back-to-back 8-hour build sessions. The hope is that by the time we walk out the door on Sunday night, we understand the game and know what we are doing.

After carefully considering the different ways you can score points, we concluded that placing GAME PIECES on the NODES was the most critical ability in this game, with it having the highest potential points available. Without the ability to DOCK and ENGAGE, however, it will be virtually impossible to remain competitive due to the lack of ranking points.

After two days of deliberation, these are the design Requirements we settled on.

DRIVE

- Need to be a Small Bot – The smaller the bot, the easier it is for 3 robots to balance on CHARGE STATION
- Need a low center of gravity
- Need to be able to drive and balance on the CHARGE STATION.
- Preferably autonomous balancing on CHARGE STATION
- Use of vison (April Tags) to provide feedback to the onboard odometry system
- Use of vision to identify and seek out game pieces on the field.

ARM

- Arm needs to be strong and durable
- Use Encoders on the input and output of gearboxes to monitor and minimize backlash.
- Either 2 or 3 Degrees of Freedom Further testing will be needed.
- Needs to score at all 3 levels BOTTOM, MIDDLE and TOP Nodes.

INTAKE

- Quickly acquire GAME PIECES (Touch It – Own It)
- MUST pick up CONES and CUBES from the LOADING STATION
- MUST pick up CUBES and upright CONES from the ground.
- Would like to be able to pick tipped-over CONES from the ground.

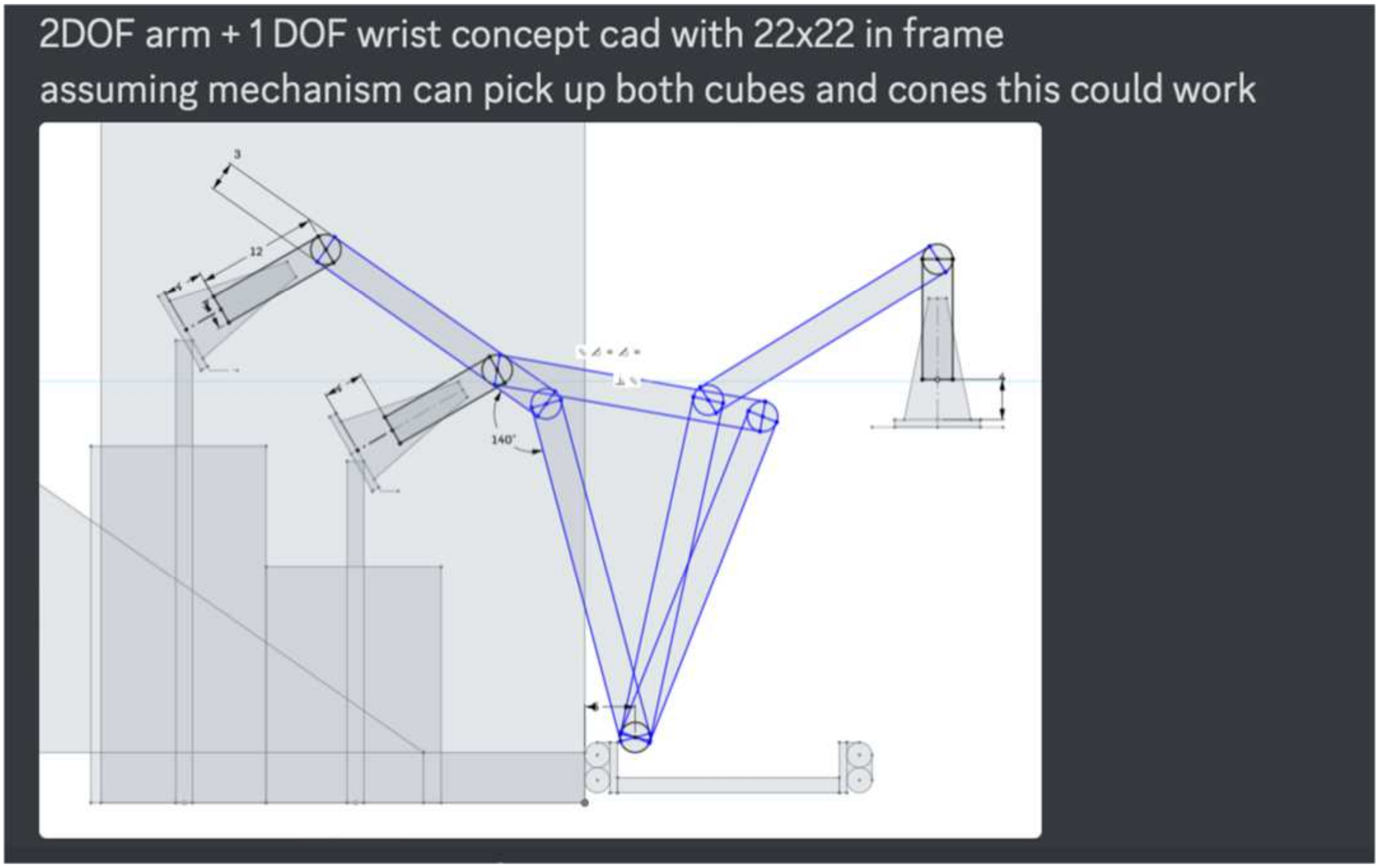
GENERAL DESIGN CONCLUSIONS

Our overall goal for the season was to be a competitive bot in district-level play and qualify for New England Championship. To accomplish this, we need to, at the bare minimum, make it to Elimination at both our district events, hopefully as an Alliance captain or 1st pick.

We approached our design as trying to build a highly reliable jack-of-all-trades bot, focusing on gaining one of the two performance-based ranking points in either match.

Inspired by the cost-effective production strategies of the Hass Formula 1 racing team and our limited team members and design resources, we prefer to use pre-engineered solutions wherever possible to focus our design resources on critical complex components.

IDENTIFYING DESIGN CONSTRAINTS



We are thinking about using an arm as a manipulation mechanism. We potentially envision a 2DOF arm + 1 DOF wrist that can pick up both cubes and cones, with a high range of motion on the wrist joint. As we can utilize the bot's movement, we do not need the Arm to move from side to side. An important note is that with an arm the starting configuration poses a good challenge, as it will need to fit inside of the robot's frame before activation. We have found that the shoulder joint only needs to move 90 degrees max, the elbow joint 210 degrees, and the wrist joint somewhere like 270 (at least in the configuration, lots to play with) to achieve all necessary motion.

THE 1757 RAPID DEVELOPMENT MODEL

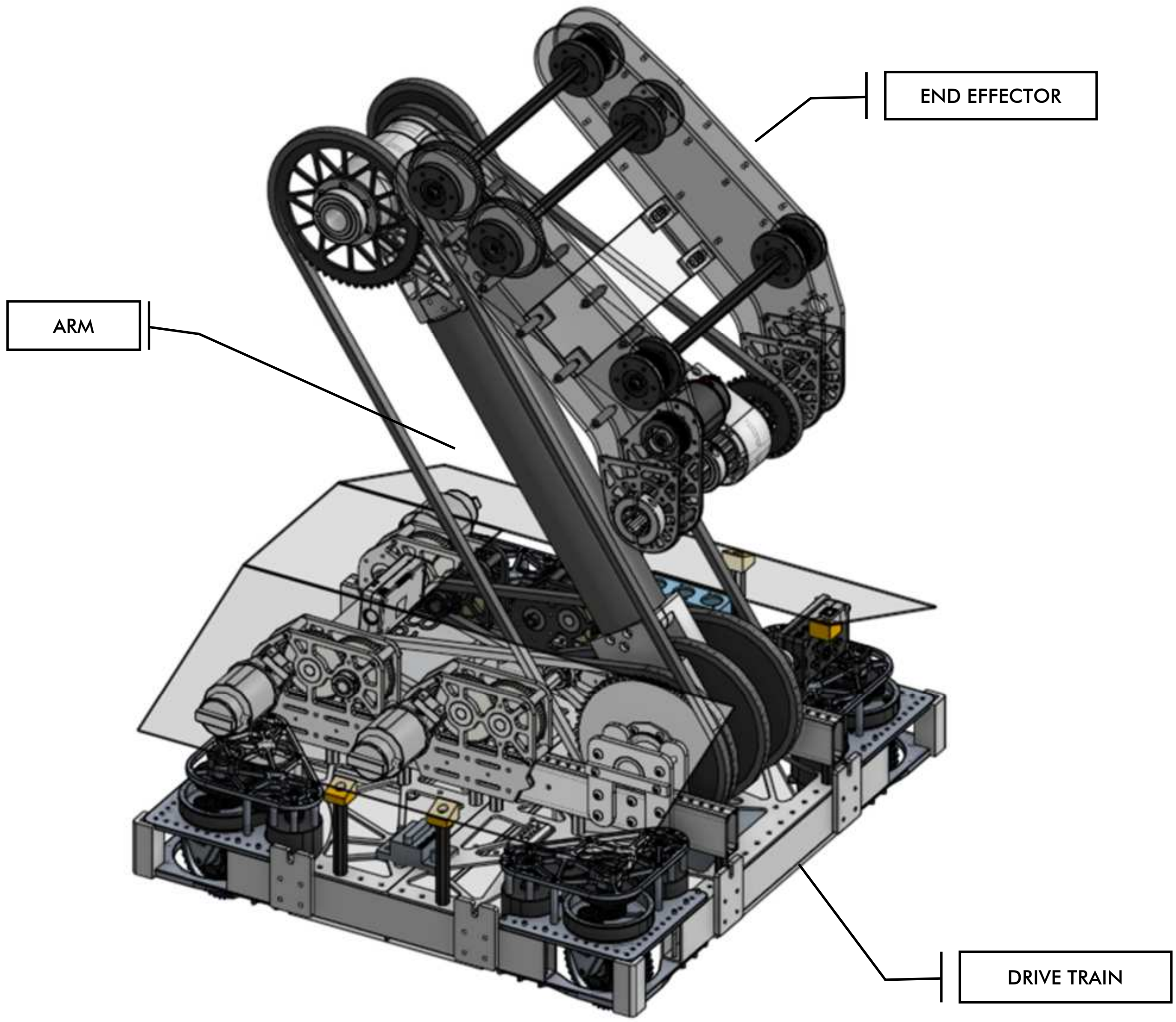
DEFINE
- Clearly Identify the design requirements of the system

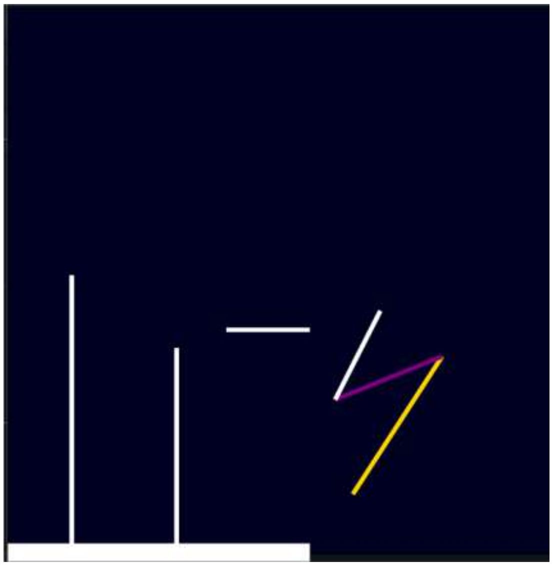
PROTOTYPE
- Design and Build a prototype that can be used to test design assumptions and Test

REFINE
- Use what we learned from testing to develop a final design

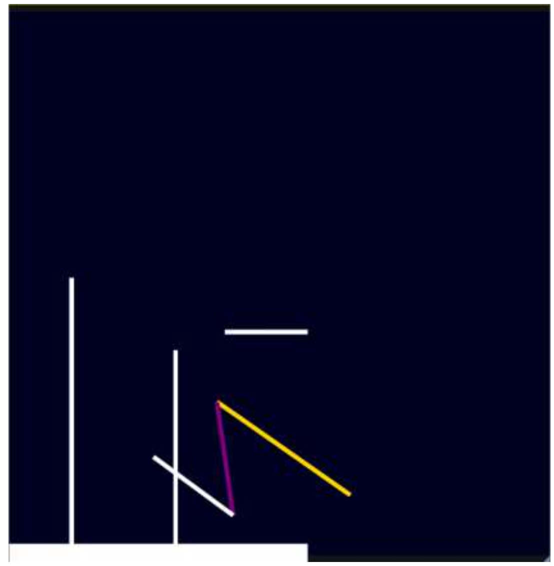
DEPLOY
- Fabricate final version and intergate into overall robot systems

FINAL ROBOT DESIGN

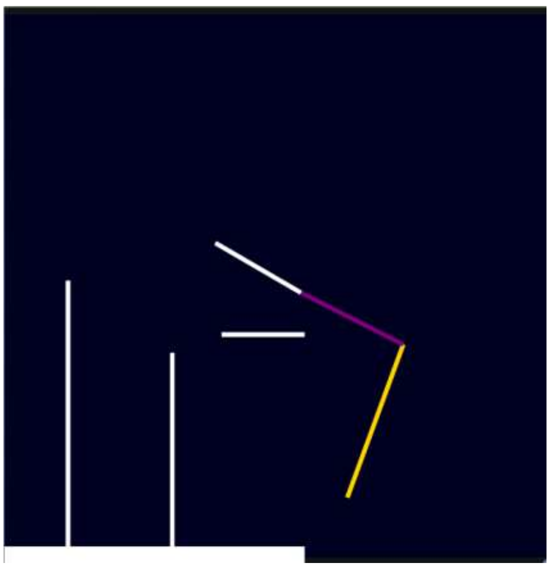




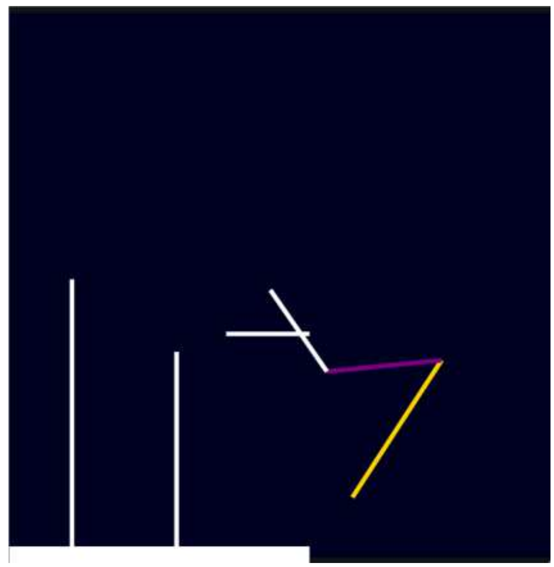
Default Configuration



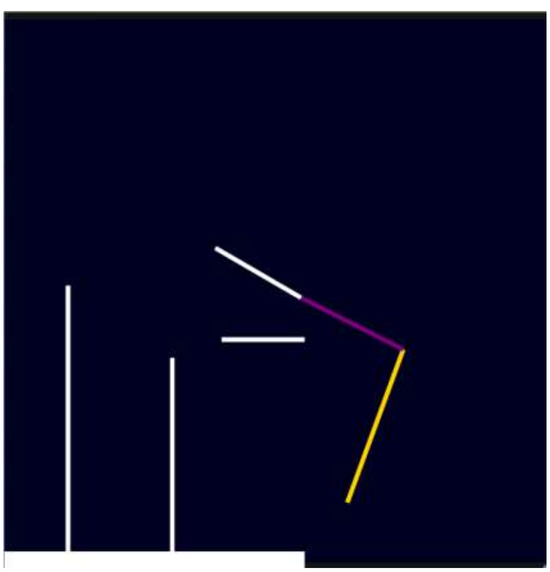
Floor Pickup



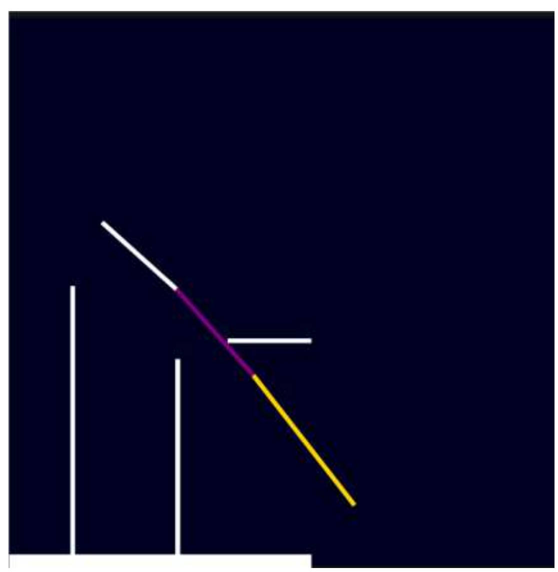
Substation Pickup



Score – Bottom



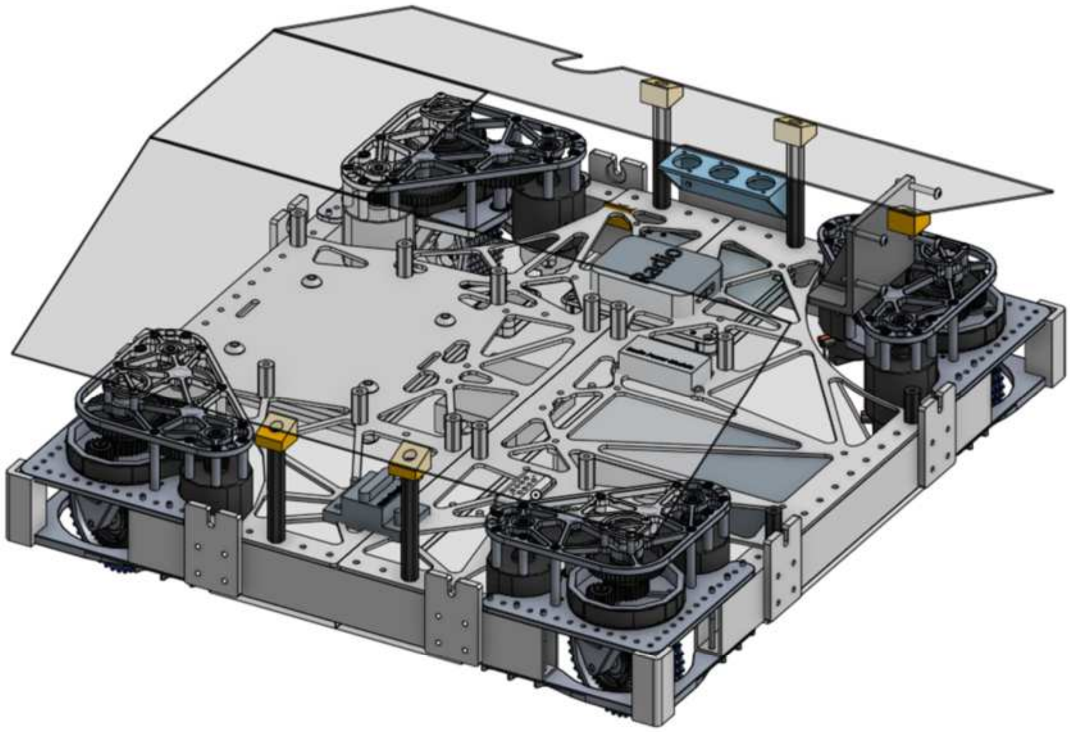
Score – Middle

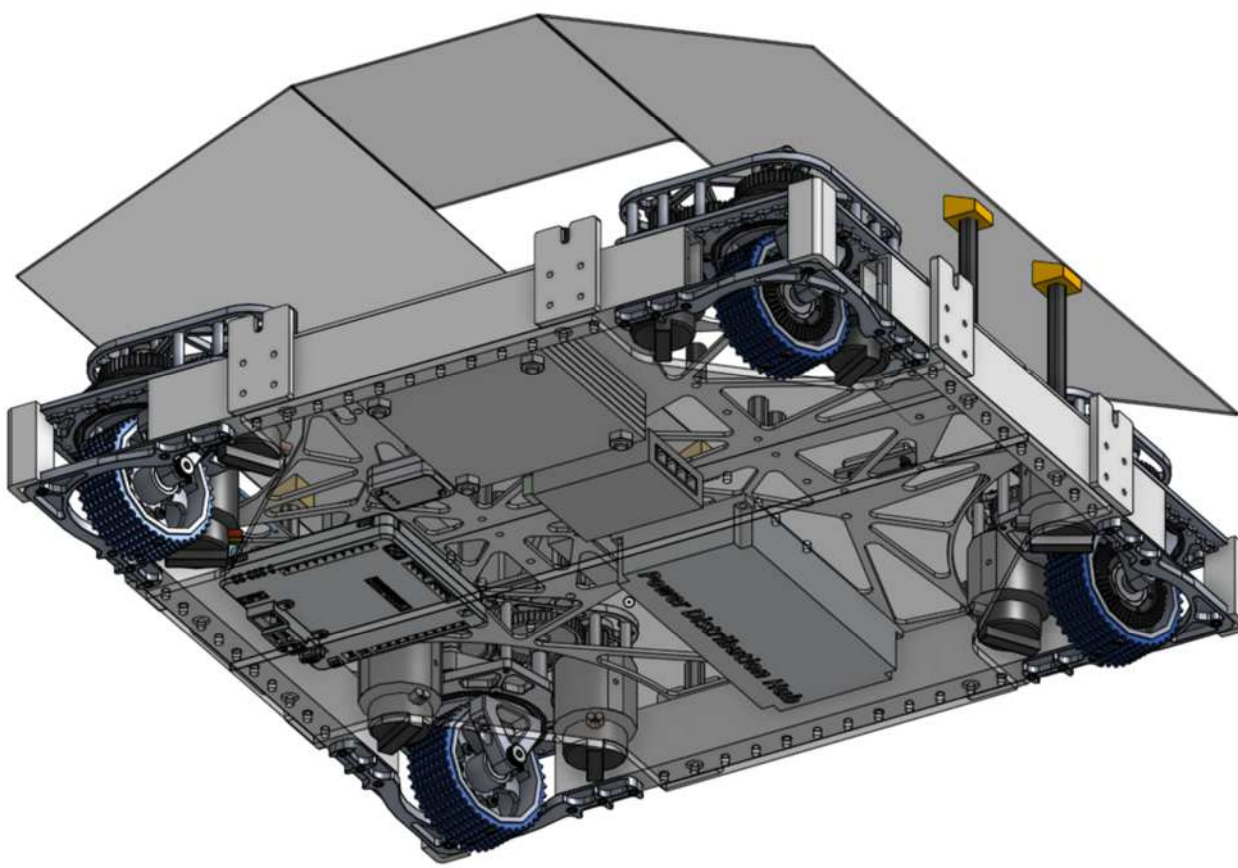


Score - Top

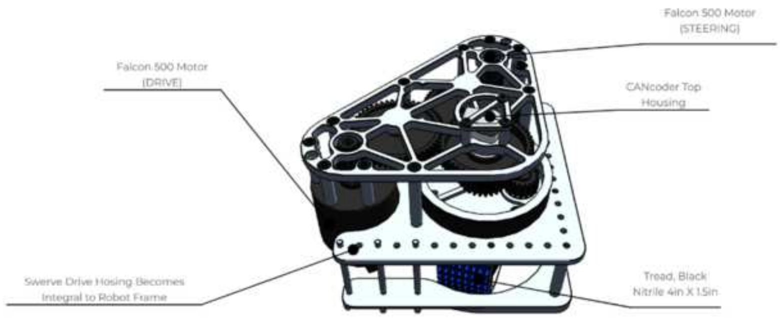
Above – 3D Simulation of Arm Joints in all of its various Arm Configurations

MAJOR SYSTEM #1: DRIVE TRAIN





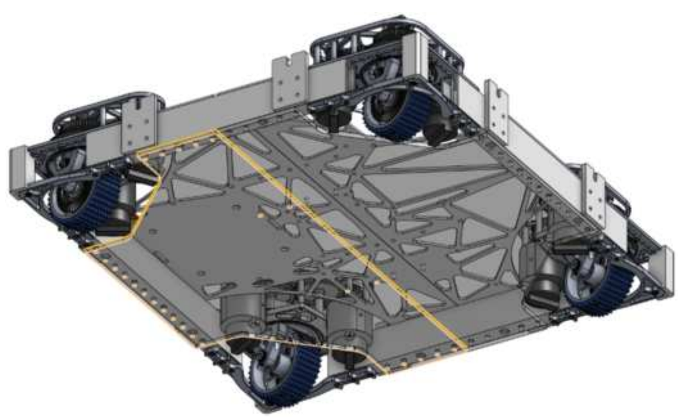
1.1 - SWERVE DRIVE MODULES



Last season was our first competitive season using swerve drive and we could not be happier with the results, because of the equal power between steering and driving there are none of the performance trade offs inherent in other drive systems. We can still run circles around the field when we need to and we can still push another robot across the field when they are in our way. One of our favorite features exclusive to swerve drive is what we call the park feature, by turning all 4 wheels to a 45° angle relative to the corners of the robot the robot effectively parks itself in place and wont move, another robot can push against us all match long and we wont move. Last year we used Swerve Drive Specialties Mk4 units, and this season we upgraded to the newly released Mk4i units. This revised design points the motors downward into the bot instead of mounting above the module. This allowed us to eliminate ≈2" of vertical space in our robot between the drive frame and the major systems.

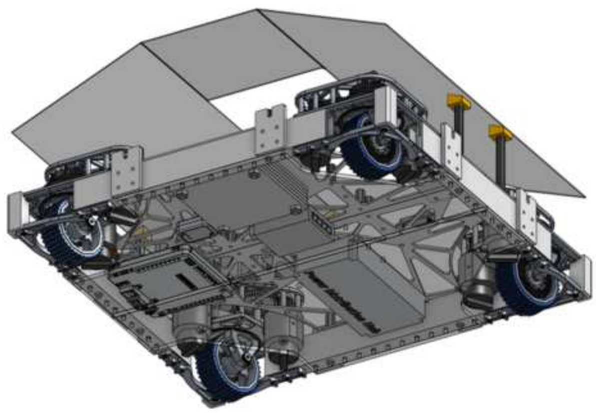
SDS MK4i Swerve Modules	
Powerplant	Falcon 500
Gearbox Configuration	L2
Overall Gearbox Ratio	6.75 : 1
Unadjusted Free Speed	16.3 ft/sec

1.2 - ROBOT FRAME



Left – CAD – Isometric Top View of robot frame
Right – CAD – Isometric Bottom View of Robot frame

1.3 - ELECTRONICS SUBSYSTEM



One thing we struggled with on our 2022 robot was how inaccessible most electrical components were. On our 2022 robot, all the electronics sat in a belly pan at the base of the bot, and the only way to access most of the components required you to remove the majority of the Robot Systems. Inspired by another team's design from 2022, we decided to hang all of our electronic components upside down and face the ground. Now to access the electronics, we tip the bot on its side, remove the ¼" protective polycarbonate plate, and you have full easy access to all the electronics components. (Credit to Team 125 for the Idea, they have the thanks of a grateful drive team and Pit Crew)

ELECTRONICS SYSTEM MAJOR COMPONENTS

- (1 ea) National Instruments - RoboRio 2
- (1 ea) REV Robotics - Power Distribution Hub
- (1 ea) Navex 2 – RoboRio MXP expansion Board
- (1 ea) CTRE CANivore
- (1 ea) CTRE CANdel
- (1 ea) BrainBoxes – SW-015 5 Port Gigabit Switch
- (1 ea) Generic Passive POE Injector
- (1 ea) Limelight 2 Camera
- (12 ea) Falcon 500 Motors
- (1 ea) REV Robotics Sparkmax brushless motor controller
- (1 ea) REV Robotics NEO 550 Brushless DC Motor
- (1 ea) Open Mesh Access Point [Insert Model Number]

1.4 - TESTING PORTS

We added a convenient patch panel to the upper side of the robot to allow for quick access to essential data ports when we don't want to access the underslung electronics.

PATCH PANEL SLOT 1 – USB TYPE A

This slot connects to one of the USB Type A ports on the RoboRio. This typically has a USB flash drive plugged in. During a match all the system logs are copied to the USB drive. After a Match, the USB drive can be pulled and opened up in AdvantageScope on the debug machine for post-game analysis. It's our version of a Blackbox on an airplane.

PATCH PANEL SLOT 2 – USB TYPE B

This connects to the USB Type B Port on the RoboRio—a redundant method for tethering the robot for control and debugging at events.

PATCH PANEL SLOT 3 – RJ45 CONNECTOR

This connects to the Ethernet Switch Via CAT5e for network access. Used for tethered connections to the bot during testing. Ethernet tethering is preferred, but we have encountered software reliability issues in the past.

DCMP Update

At the Revere District event we ran into serious problems tethering to the robot via ethernet and via USB B. we traced the ethernet tethering problem to a problem with the network configuration issue on the driver station laptop. We were unable to determine a definite cause of the USB-B connection issue, but, we think it most likely to to poor quality of the 90° usb connector used on the robot. From that point on we connect a USB-B cable directly into the port on thr

At the same time we realized we needed a button to manually put the arm motors in coast mode for serviceability when the bot is not connected to the driver station. Since we are no longer using the USB-B testing port we replaced it with a momentary push button switch.

1.5 – CAMERA/VISON SYSTEMS

LIMELIGHT 2 – CAMERA

We are utilizing a Limelight 2 Camera for a variety of tasks on the robot mostly devoted to sensor fusion and automation of systems using computer vision. The Limelight's field of vision (FOV) is essentially parallel to the floor and at the height of the April Tags.

Please refer to the Software section of this document for more information on how we use the limelight and April Tags to improve the Onboard odometry of the robot.

1.6 – COUNTERWEIGHT

Not originally intended as part of the design, upon testing of the robot with the Arm fully extended in the scoring position, we realized that robot was prone to falling forward. To resolve this, we looked to add ballast to the bot. First we thought of lead but didn't want to deal with the potential health risks of improperly encapsulated lead. We investigated tungsten; however, a review of the current price of tungsten plate (≈$40/kg) quickly ruled it out as a potential candidate. We settled on 6" x6" x1/4" steel plates mounted directly under the robot battery. After testing with different #'s of the plate, we decided on 6 Plates with a total weight of ≈25 lbs. Now the bot is highly stable even when the Arm is fully extended.

1.7 – PROTECTIVE COVERS

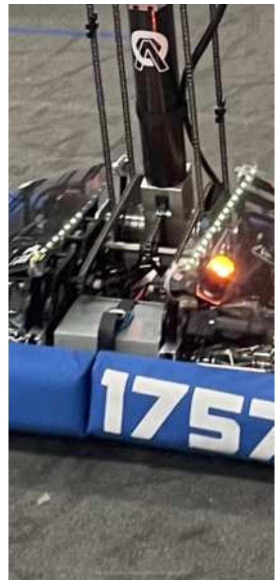
We added protective covers that Slope away from the central arm structure down to the bumpers. Not only do these plates provide a valuable location to display all of our great sponsors they serve to prevent errant game pieces from getting stuck inside the robot during a match.

DCMP Update

Originally the protective covers were only hled on with 3M™ Dual Lock™ SJ3560, this material is nice because it very strong but eaisily removable. Durring qualifying matches in Revere however, these panels kept falling off and dragging around the fieild. The Dual Lock strips were reinforced with zip ties and these held through all of playoffs in revere, and all of qualifications at WPI. Then in Playoff Matches we shered off 3 of the metal standoffs holding up the protective covers. We made quick repairs to keep going however prior to DCMP we will be swaping out all the ½ thunderhex standoffs with 1" 80/20 extrusion with hardened bolts for strength.

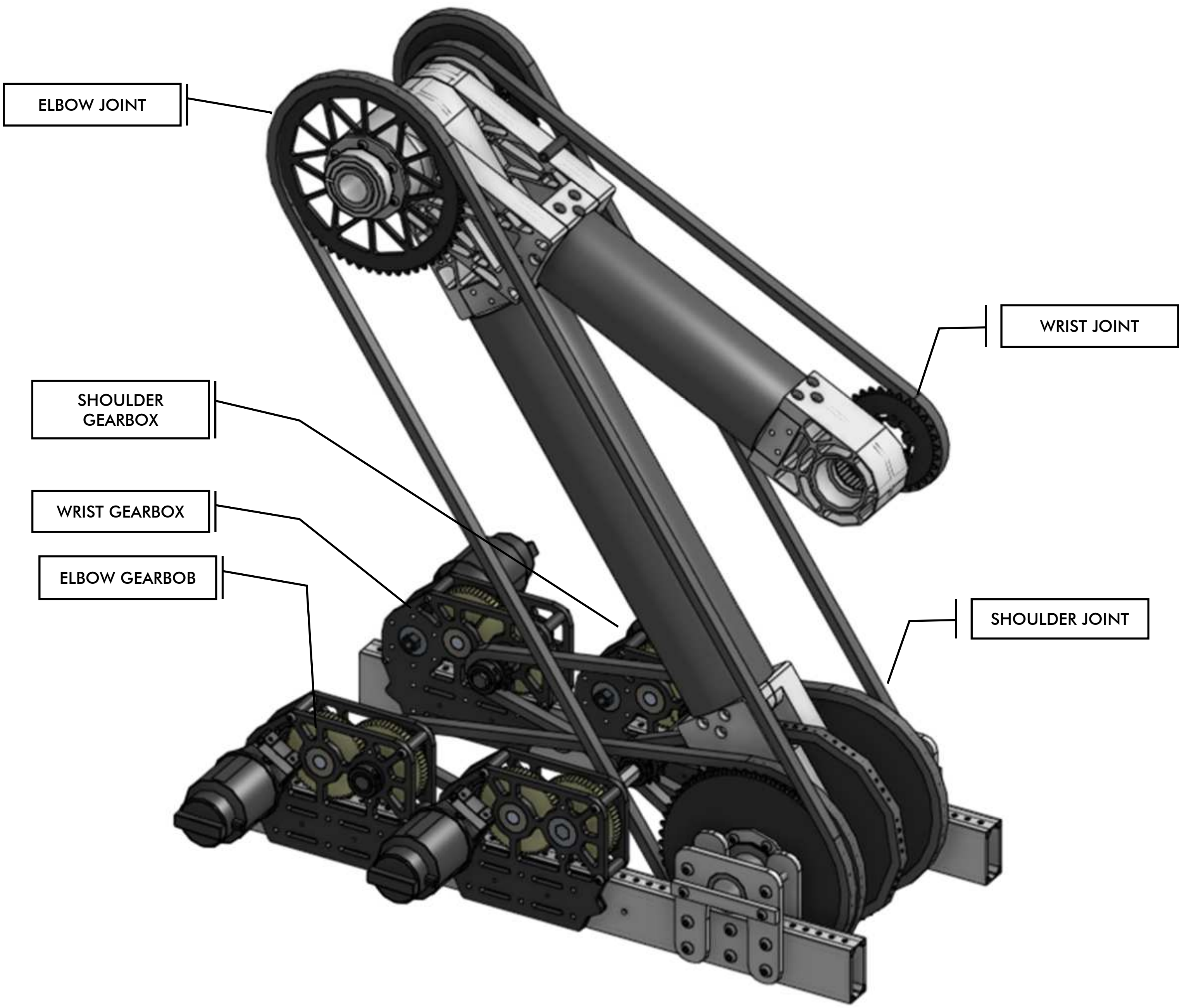
1.8 – GAMEPIECE INDICATORS

One of the hardest things to do in a match is how to signal between the driver station and the human player what game piece you want them to load into the robot. People use Hand Signals, Colored pieces of paper or guess. We wanted to take the guesswork out of the equation, so we mounted 2 LED Strip lights along the top of the protective covers. The driver controls what color these strips are so he can communicate to the human player which game piece to feed to the bot – Yellow for Cones and Purple for Cubes.



Left – Robot Displaying "I Want a Cone"
Right – Robot displaying "I Want a Cube"

MAJOR SYSTEM #2: ARM



2.1- MOTORS & GEARBOXES

To keep the robot's center of gravity low and keep the Arm as simple as possible we decided to locate all of the heavy motors and gearboxes at the base of the superstructure.

GEARBOX # 1 – SHOULDER GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into Custom Gearbox based on WCP Gearbox design.

GEARBOX # 2 – ELBOW GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into COTS WCP Single Speed Gearbox.

GEARBOX # 2 – WRIST GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into COTS WCP Single Speed Gearbox.

2.2 - CHAIN DRIVE

Using a combination of dead and live axels we transfer the power of the gearboxes up though the Arm to power each of the individual joints. For Reliability and durability, we chose to use #35 roller chain rated for 11,000 lbs of force.

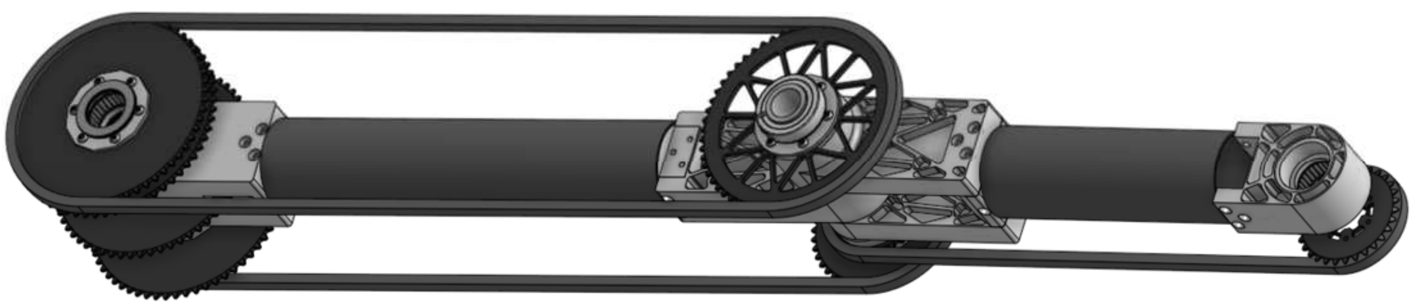
Below is a summary of the different chain runs on the Arm

CHAIN DRIVE 1 – SHOULDER
Shoulder Gearbox Output 12t Sprocket → 60t sprocket on Shoulder (Dead Axel)

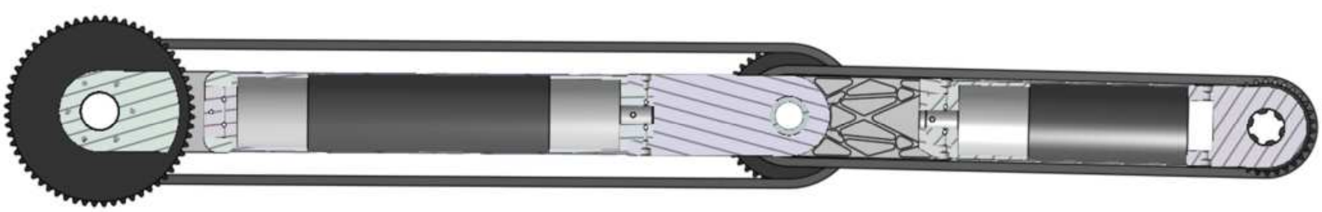
CHAIN DRIVE 2 – ELBOW
Elbow Gearbox Output Shaft 12t Sprocket → 60t Sprocket on Shoulder (Live Axel) → 70t Sprocket on Elbow (Dead Axel)

CHAIN DRIVE 3 – WRIST
Wrist Gearbox Output Shaft 12t Sprocket → 60t Sprocket on Shoulder (Live Axel) → 60t Sprocket on Elbow (Live Axel) → 40t Sprocket on Elbow (Live Axel) → 32t Sprocket on the Wrist (Dead Axel)

2.3 - ARM STRUCTURE



Above - 3-D View of Outstretched Arm



Above - Section View Though Center of Carbon Fiber Arm

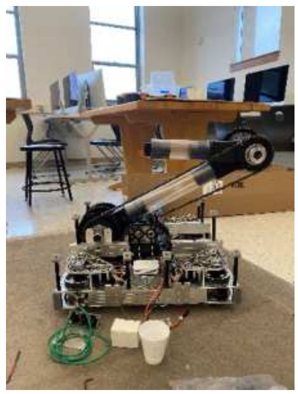
CARBON FIBER ARMS

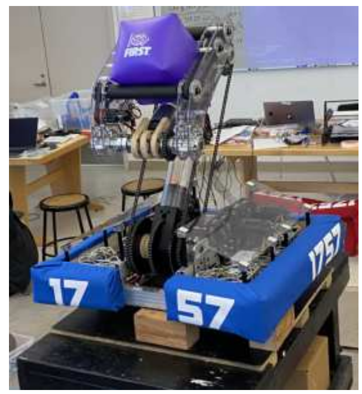
We chose to use carbon fiber tubes as the main structure of the Arm due to its strength and lightweight, the more weight we could save on the Arm the lower we could push the robot Center of Gravity. Carbon Fiber tubes are a stock McMaster item 3" Ø. Carbon Fiber is Epoxy bonded to 3" hollow aluminum plugs bolted to the aluminum joints.

Left – Final carbon fiber arm links after final glue up
Right – Final aluminum plugs used in the ends of the carbon fiber tubes

3D PRINTED AND POLYCARBONATE PROTOTYPES

Because we knew the carbon fiber and machined aluminum would take time and money to manufacture, we heavily used 3D-printing to make prototypes of the Arm and test and confirm critical geometry before placing final fabrication orders. These prototypes are two fragile to be used on a competition bot but worked well for their intended purposes. We learned very Important lessons about where the concentrations of forces were along the axels and what parts needed reinforcement.



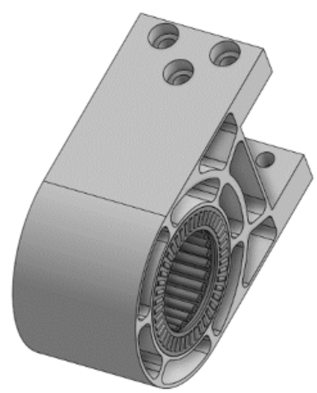


Left – 3D printed Prototype of the wrist joint, printed on a FormLabs 2 SLA Printer
Center – Polycabonate Prototype arm Mounted on bot for the First Time
Right – Fully Assembled "Alpha" Robot build

2.4 – JOINT STRUCTURE

CAD - Shoulder Joint

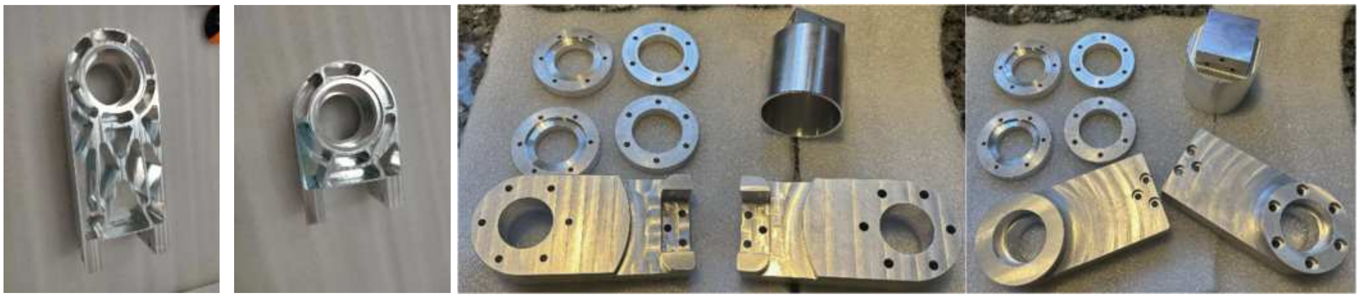
CAD - Elbow Joint



CAD - Wrist Joint

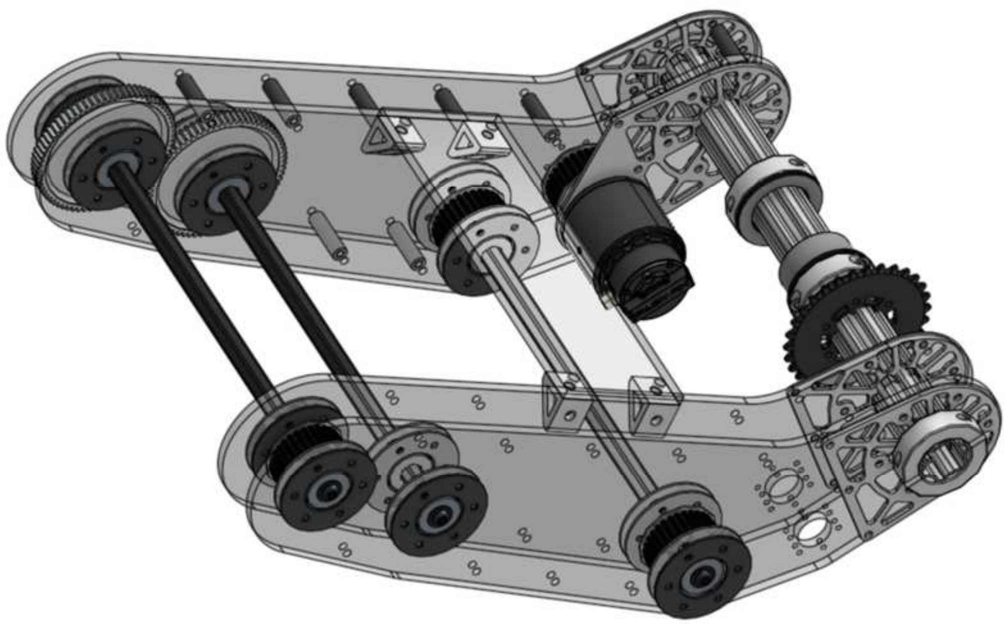
NEEDLE AND THRUST BEARINGS

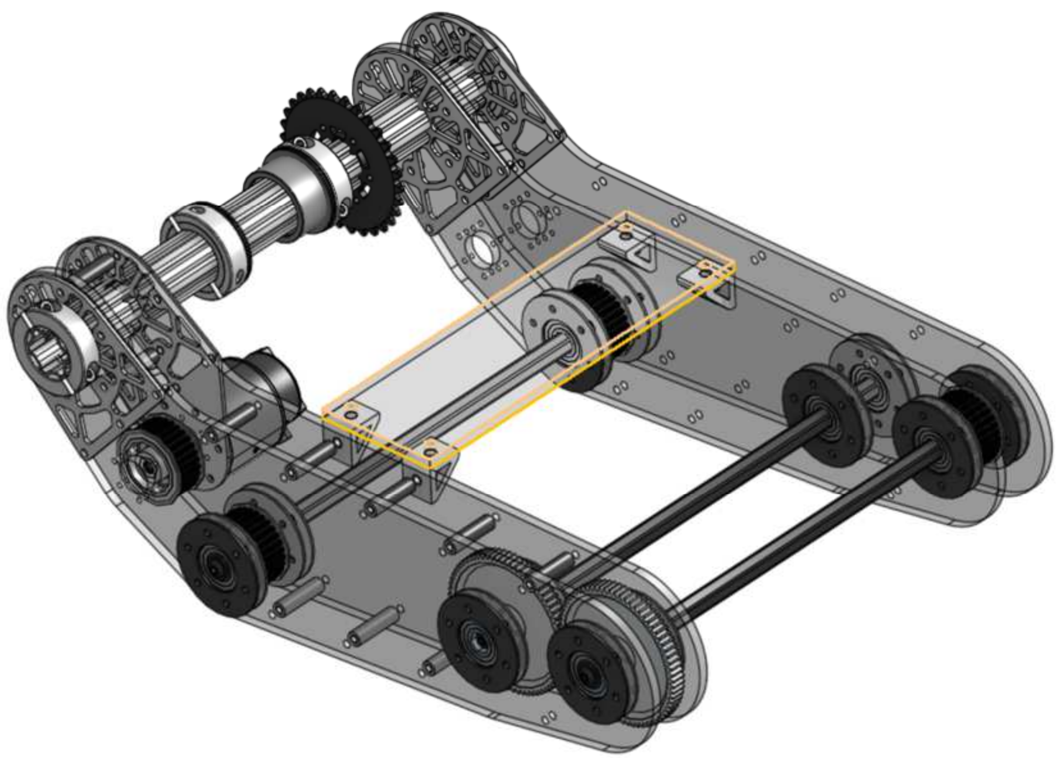
Used in All three joints to allow for smooth rotary motion in each joint.



Photos - Final Machined Parts

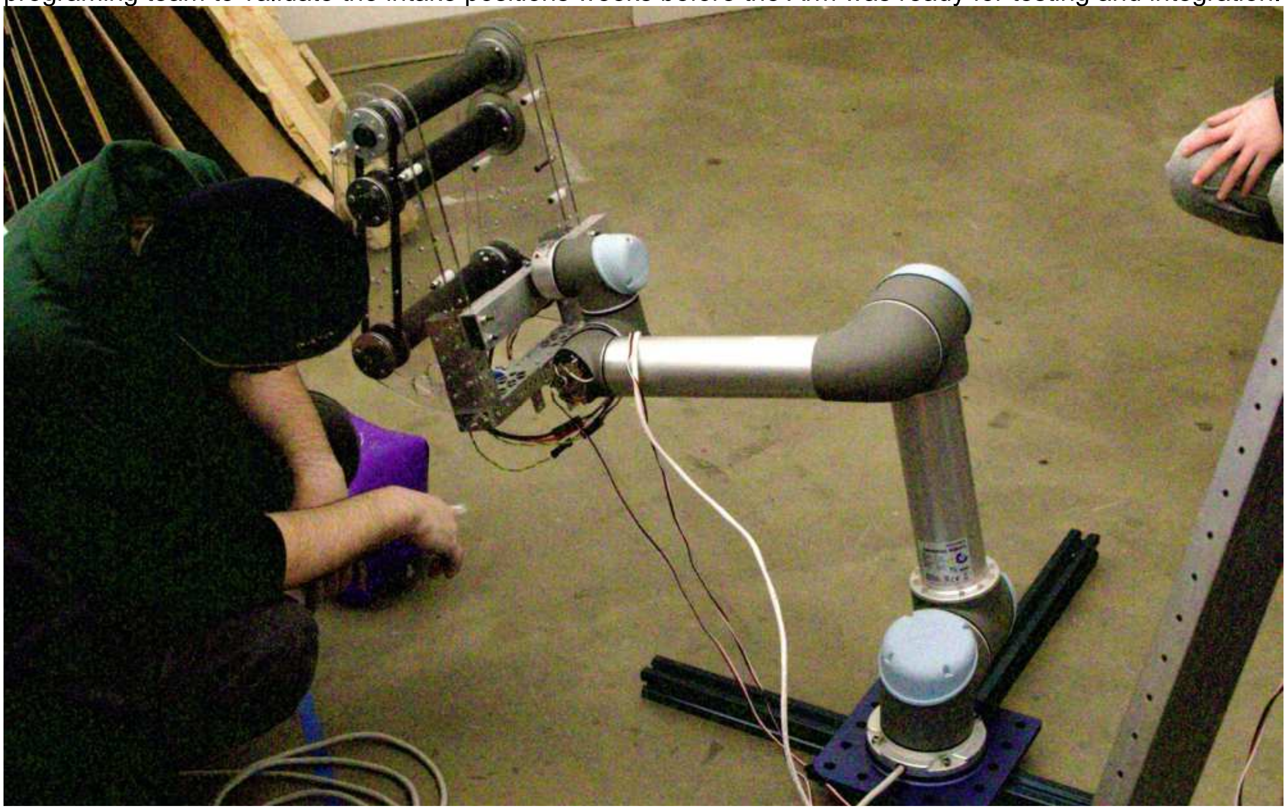
MAJOR SYSTEM # 3: END EFFECTOR





INTAKE PROTOTYPING – FUN WITH INDUSTRIAL ROBOTS

We had a lot of fun prototyping this mechanism, since it was the first major mechanism that we finished we had lots of time to put it trough its paces. One of our mentors has access to a Universal Robots UR3 robot and brought into our lab during one of our weekend build sessions (see images below). This actually proved to be essential because it allowed our programing team to validate the intake positions weeks before the Arm was ready for testing and integration.



3.1 – ROLLERS

The rollers we are using are Vex VersaHub Rollers with ¼" neoprene tubing stretched to cover them, they are very grippy and hold the cube verry securely. We started with the dimensions of the everybody roller for our prototype then made some modifications before settling on final separation differences. The neoprene tubing is undersized for the OD of the polycarbonate roller. We learned a fun trick to clamp off one end of the neoprene tube and inflate it with an air compressor to stretch it over the polycarbonate tube, when the air is released, it makes a perfect friction fit between the Neoprene and the polycarbonate. We have had no detectable slippage after weeks of testing with the rollers.

DCMP Update

After 33 competitive matches one thing is clear, we have problems picking CUBES up off the floor and in order to maintain our competitive edge at DCMP we know we need to be able to get CUBES up off the floor. We thing the majority of the problem is related to how narroe our end effector is, the original design was only 1” wider than the width of the CUBE. To aliviate this issue we are planning to widen the end effector by 3”.

3.2 – MOTORS & GEARBOXES

The intake is powered by a REV Robotics NEO550 Brushless motor into a REV Robotics Ultraplanetary gearbox. The motors small size is nice however because we mounted the Sparkmax Motor controller on the intake as well there is no significant weight savings compared to using a Falcon 500 with an integrated Talon SRX. We may end up swaping this out for simplicity sake in the future.

REV Ultraplanetrary	
Powerplant	NEO550
Gearbox Configuration	4:1, 5:1
Overall Gearbox Ratio	20:1

SOFTWARE

SOFTWARE: OUR DEVELOPMENT ENVIRONMENT

WPILib

The perineal stalwart, we still rely on core elements of WPIlib for robot communications and debugging. WPILib's new Logging features have greatly enhanced our Debugging capabilities

RobotPy

We have found that students have a lot easier time learning python then they do Java or C++ so with the growing support for RobotPy we migrated our Codebase from Java to Python in 2020. As of this March we are an official contributor to the RobotPy project

GitHub

Without Github our level of remote work and collaboration just wouldn't be possible.

SOFTWARE: NEW AND UPDATED TOOLS THIS YEAR

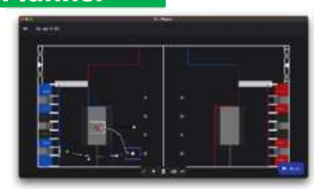
AdvantageScope

WE LOVE ADVANTAGE SCOPE! Not only does it log *everything* but does it in away that is intuitive and easy to review. No more searching though 10000 lines of log files to find the one piece of information we need. Huge thanks to team 6328 for building such a great tool.

PhotonVision

We are using PhotoVision as our native development framework for Computer vision due to its growing wide support inside the FRC community. It does not include native support for RobotPy however so as an offseason project our lead programmer wrote a custom wrapper for PhotonVivion so it can work inside our RobotPy environment

PathPlanner



Last year we used PathWeaver, but we were disappointed in the lack of native support and increased complexity in the development stack so starting with the off season we transitioned all of our Autonomous path planning to PathPlanner. We had much fewer issues with this system.

SOFTWARE: DRIVE

Taking off of last year, the drivetrain codebase has stayed the same. We are running field oriented drive with robot relative rotation to allow for quick maneuverability. A button to align to the nearest 90 degree angle was added to help with driver alignment. This state slightly reduces the speed and snaps the angle of the robot in order to have perfect alignment to the double substation, single substation, and grid every time. For our automated balance sequence, we work in robot relative space on the robot relative gyro.

A BRIEF TANGENT - ABSOLUTE RELATIVE DRIVE

Last year our lead programmer had a new idea for drive control, an absolute relative drive. The common swerve drive control method was to have a field relative translation for the bot, and a robot relative rotation. What this meant is a left input on the rotation axis would result in the robot rotating to the left at a constant speed. A translation action was not affected by rotation but instead was in "field relative" space. The difference of absolute drive is that the rotation is also field relative. A left input on the rotation stick will yield the robot turning to face left. This year we expect this type of robot control to be very important for drivers when they have to be able to turn to specific positions for collection and scoring on swerve drives. You can see this in action in any one of our videos from last year. Having fixed controlled rotation will allow for precise driver input and less fiddling with controls when cycle time is very important.

The drivers have also experimented with alternate driving methods on swerve to get used to interesting control schemes such as a curvature drive, standard tank drive, standard field relative drive, and full robot relative drive.

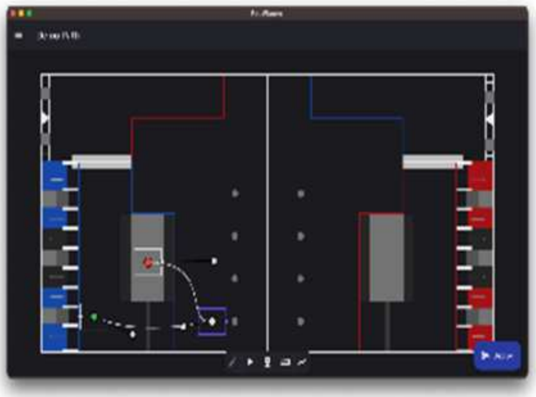
SOFTWARE: INTAKE

The intake is using a state machine in order to regulate its expected behavior. There are 2 enumerable values: one for the gamepiece intended direction (intake, outtake, hold) and one for the desired gamepiece type (cube, cone). The transition between each state is dictated by a user input to any given category. If no input is given, the system holds its position and keeps the desired gamepiece remembered. The state value of the desired gamepiece is displayed to the driver and to the human player through pulsing leds of the respective color.

SOFTWARE: ARM

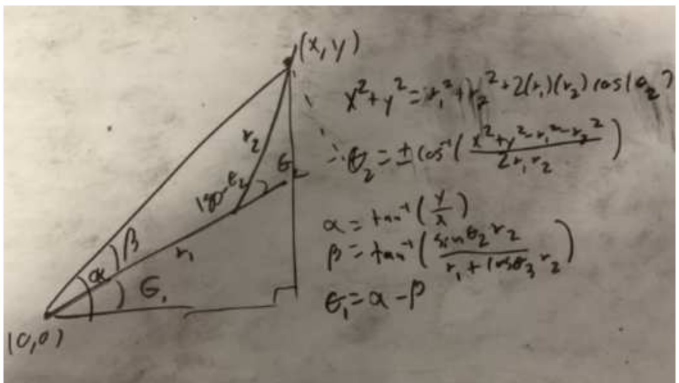
A triple jointed arm is no easy feat in order to program smoothly. From cad, states are given about the end effector's desired position and rotation relative to the floor. From there we use inverse kinematics to determine each per-joint relative rotation at any given position. A cartesian control on the wrist joint's position is added using a trapezoidal PID profile to lay out a path for the Arm to follow. For the wrist it has another trapezoidal PID profile controller. When going state per state on the Arm's motion, we check for if the relative angle goes over software end limits in order to prevent running the Arm into itself. These are done in joint relative space. Since the Arm is controlled from the base through chain and sprockets there is a virtual 4 bar created in which the rotation of any given joint is given relative to the ground. These are converted into motor space and passed onto each motor where they have a position PID controller onboard. For simulating the system we are using mechanism2d to view the expected values of the Arm and be able to run through positions. This simulation first approach has allowed minimal revision and a solid foundational codebase that is mostly complete before the bot is finished. Furthermore the position of the Arm is logged in 3d with advantagescope based on the position it believes the bot is in. Logging was also a priority for a complex system and therefore we log the instructed position and actual position per joint and each end position in cartesian space.

SOFTWARE: AUTONOMUS



We experimented in the offseason with pathplanner and use it extensively for our autonomous. Each necessery command is placed into a potential pool of events for pathplanner to fire. At the beginning a sequence determined solely in pathplanner is fired. Going off of last year we had a large time gap in order to make slight changes so instead for this year we are using the waypoint system and opting to have any given autonomous contained solely in pathplanner. This has increased our ability to construct autos and tweak any given aspect as needed. For the library itself of parsing, the lack of first class robotpy support meant we had the freedom to reimplement autonomous however we pleased based on the path. We follow a changing trajectory and the swerve drive using onboard odometry and a weighted vision estimate determines its bot position relative to the global field and follow through it between each section.

SOFTWARE: SIMULATION



armsubsystem.py

```
def setEndEffectorPosition(self, pose: Pose2d):

    twoLinkPosition = Translation2d(
        pose.X() - constants.kArmwristLength * pose.rotation().cos(),
        pose.Y() - constants.kArmwristLength * pose.rotation().sin(),
    )

    endAngle = math.acos(
        twoLinkPosition.X() * twoLinkPosition.X()
        + twoLinkPosition.Y() * twoLinkPosition.Y()
        - constants.kArmtopLength * constants.kArmtopLength
        - constants.kArmbottomLength
        * constants.kArmbottomLength
        / (2 * constants.kArmtopLength * constants.kArmbottomLength)
    )

    startAngle = math.atan2(twoLinkPosition.Y(), twoLinkPosition.X()) -
math.atan2(
        math.sin(endAngle) * constants.kArmtopLength,
        constants.kArmbottomLength + math.cos(endAngle) *
constants.kArmtopLength,
    )
    wristAngle = pose.rotation().radians() - startAngle - endAngle

    bottomArmEncoderPulses = (
        startAngle
        * constants.kTalonEncoderPulsesPerRadian
        * constants.kBottomArmGearRatio
    )
    topArmEncoderPulses = (
        endAngle
        * constants.kTalonEncoderPulsesPerRadian
        * constants.kTopArmGearRatio
    )
    wristArmEncoderPulses = (
        wristAngle
        * constants.kTalonEncoderPulsesPerRadian
        * constants.kWristPivotArmGearRatio
    )

    self.topArm.set(Falcon.ControlMode.Position, topArmEncoderPulses)
    self.bottomArm.set(Falcon.ControlMode.Position,
bottomArmEncoderPulses)
    self.wristArm.set(Falcon.ControlMode.Position, wristArmEncoderPulses)
```

Due to our team's resources, virtual simulation is a huge part of our ability to quickly and reliably construct the bot's codebase. Some key examples of simulation are a wrapped onto a simulated falcon motor. Given our team's extensive use of falcons on the robot, a wrapper that provides simulation support allows for the programming team to iterate much easier and creates a cleaner codebase. Each falcon is logging the values of the motor % and the encoder position, as well as an override value to allow the user to manually in simulation change the value for sensor readings. Entire robot configuration is done on a single call and the getting of velocity, position, and percent and the setting of velocity, position, and percent are easy to access functions to allow interfacing with the motors more accessible than the CTRE library. Given this robot also has a NEO550, the simulation system was adopted to have a similar interface for ease of replacement from a falcon to a motor on the intake. We geometrically derived the inverse kinematics for 3 links with a fixed Pose endjoint. Each of these poses actually allows for two configurations of the proximal 2 arm joints (they can simply be mirrored over the line created from the wrist joint to the shoulder joint, however by forcing the sign on the elbow joint they can all be consistent.

SOFTWARE: VISION

NOW WITH APRILTAGS AND PHOTONLIB

We have a vision system complete with sensor fusion for complete robot localization. Last year,we worked with our first complete vision system as a team that resulted in significantly enhanced system performance, and using apriltags will be very important to account for combined sensor error as well as for being able to reliably use sensor data for automated alignment to various points on the field such as the double substation and the grid.

THE HOW

Photonvision generates camera-relative 3d transforms of each apriltag. Since the position of the camera is known and the position of the apriltag is known, the position of the robot can be determined from a single apriltag datapoint. These transforms are fed into a RobotPoseEstimator in order to create a sense of where the robot could be at a given time, this is combined with the gyro and wheel encoder information to get an accurate sense of where the robot is on the field at any given time. This is used in other subsystems when needed, as well as results being logged to AdavantageScope through the usage of each known pose and ghost poseseepaste

GOING FURTHER

We plan on using this odometry data to have automated alignment in complete robot space for important precision actions such as placement of gamepieces on the grid and collection of those gamepieces. Autonomous will also use this data. Perhaps an automatic engagement on the charge station by using the rotation gained from the apriltags will be possible. Overall having a sense of where the robot is on the field is beneficial to aid in other systems.

ENGINEERING TEAM

WHS Faculty Advisors

James Looney & Raul Madera

Team Captain

Charley Marsland*

Team Business Lead

Sean Tao

Team Technical Lead

Luke Maxwell

Senior Mentors

Dwight Meglan | **Chris Aloisio°** | **Steve Harrington°**

Mentors

Anthony Gelsomini | **Manny Barros°** | **Sean Lendrum°** | **Mark Holthouse** *Mentor Emeritus* | **Amber Maxwell**

STUDENT MEMBERS

Safety Captain		Graphic Design Lead	CAD Lead
Jacob Kaplan		**Ivan Cai**	**Landon Bayer**
John Santasuoso*	**David Girard***	**Declan MacDonald**	**Liam McWeeney**
Alex Theofilou	**Claire Peng**	**Andrew Gong**	**Rachel Qu**
Baili Jiang	**Nolan folmar**	**Jeffery Pan**	**Melissa Yang**
Divya Siddi	**Vinny Milinazzo**	**David Confoey**	**Adrianna Cirillo**
Elizabeth Lowney	**Jacob Liu**	**Anthony Yang**	**Constantina F.**
Luke Li	**Owen Monahan**	**Erik Curlli**	**Kevin Bai**
Henry Marsland	**Nina Pappas**	**Rachel Lee**	**Hanya Yang**
Sophia Patrick	**Lauren Buza**	**Kaylee Phu**	**Elias Mukdissi**

Westwood Robotics, Inc
Board Of Directors

Steve Harrington° *President* | **Catherine Tseng** *Treasurer* | **Manny Barros°** | **Cheten Gopal** *Clerk* | **Jack Tseng**

**Denotes High School Seniors*

°Denotes FRC Alumni

SPONSORS

Heartlander Surgical	Westwood High School	Medtronic	Act Fasteners
Gene Haas Foundation	TE Connectivity	General Dynamics	IRA Motor Group
Bayer Familt	Capital Group	Yamaguchi Family	Plymouth Rock Assurance
Deadham Savings	Maxwell Family	Schneider Electric	Marsland Family
Porier Apliances	Franklin Templeton Investments	Jake and Joes Resturant	McGolf
Roche Bros		Capers Catering	

VENDORS

Andymark	West Coast Products	Vex Pro	Swerve Drive Specialties
Discord	Google Work	Onshape	Robo Promos
McMaster-Carr	The Home Depot	Cross the Road Electronics	MS Visual Studio
Python	GitHub	RobotPy	Java

1757

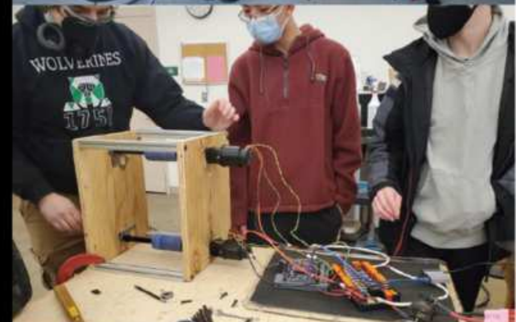
Special thanks to everyone who makes this team possible Including

Our Mentors
Our Students
Our Teachers
Our Parents

Our Amazing Sponsors

Westwood Public Schools
Emily Parks - Superintendent
Allison Borchers - Asst. Superintendent
Deb Nolan - Building Use Coordinator
Tom Carey - Director of Facilities
Ron Smith - Assistant Director of Facilities
Amy Davenport - Principal
Aishleen Flanagan - Assistant Principal
Tom Millett - Assistant Principal
Andrew Miller - Science Dept. Chair
WHS Janitorial/Facilities Staff
WHS Library Staff
The Open Alliance

#FRC1757
@WWRobotics1757

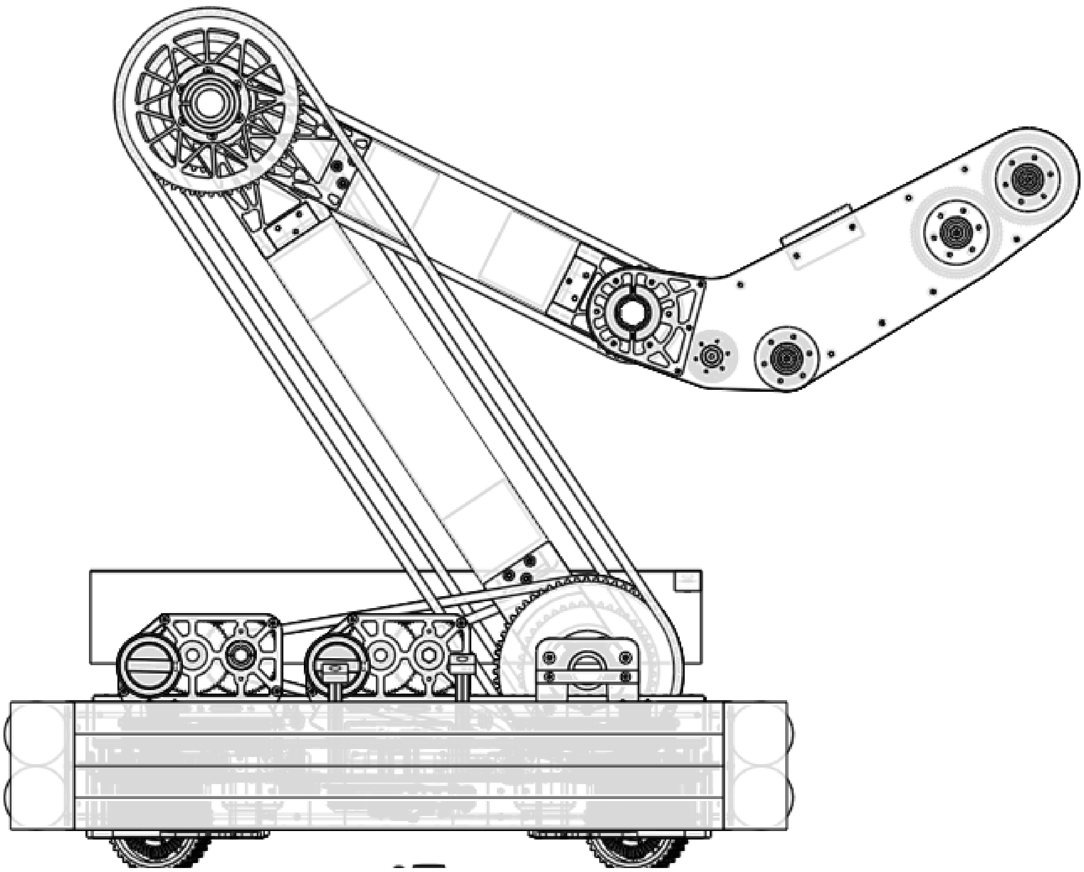






CHARGED UP

PRESENTED BY HAAS Gene Haas Foundation

PROUD MEMBER



FRC WORLD CHAMPIONSHIP EDITION

LUXO

WOLVERINES

1757

2023 TECHNICAL BINDER

FORWARD

Hello, and let us welcome you to FRC Team 1757's 2022-2023 Season. This season has continued the tremendous growth in our robot's design and technical ability that started last year as our team emerged from the hibernation of COVID-19 to become a surprising contender in the New England Region. Continuing to recruit rookie students to supplement our now more veteran team members and Senior Mentors, we have pushed our collective talents to their limits to deliver the competition-worthy robot contained within the pages of this binder.

Our season started in the fall of 2022, introducing a new class of over 10 freshmen, sophomores, and juniors to the world of FRC. We showed off the robot at local town events, built a T-Shirt Cannon to raise school spirit at the prep rally, and hosted weekly technical seminars on everything from the engineering process to CAD, Electronics, Pneumatics, Mechanics, and everything in-between. Over the Summer we got a new OMIO X8 bed router and practiced our CAD and fabrication skills by designing and building an enclosure for the machine. We traveled to Billerica, MA in October to compete in the first-ever New England Robotics Derby. We finished in Second Place, losing in the Finals (The best competitive finish in team history). We piled into our classroom on a cold Saturday morning in January, eagerly anticipating this year's game. 4 CAD models, 8 shared Google Drives, ten weeks, 20 Weekend Build Sessions, 50 Zoom calls, 5799 lines of code, 170 git commits, 19,129 discord messages, and many, many cups of coffee later, we are proud to unveil our robot "LUXO" for the 2023 FRC Season.

Why did we name the robot LUXO? Is it because of the shining lights on its frame that illuminate what game piece we are looking for on the field…no. Is it the bright shining future of the team…no. Is it a reference to solar power and how that ties into the theme of this year's FIRST season…good guess, but no. In truth, we are a bunch of animation nerds, and we thought the robot looked like the lamp in the Pixar Animation title sequence named Luxo. Not every robot name has a deep prophetic meaning…sometimes it's just about the memes.

One very exciting thing about this year is that Team 1757 joined the Open Alliance. We found the Open Alliance teams and their open and timely build season updates so helpful to our team last season that we decided to join so we could help other teams the same way the alliance has already helped us. In addition to frequent updates on our build thread, we also made two appearances on the Open Alliance Show Streamed on twitch. If you want to learn even more about our robot and the design process, beyond what is contained in this manual, please visit our Chief Delphi Build Thread at https://www.chiefdelphi.com/t/frc-1757-wolverines-2022-2023-build-thread/416564

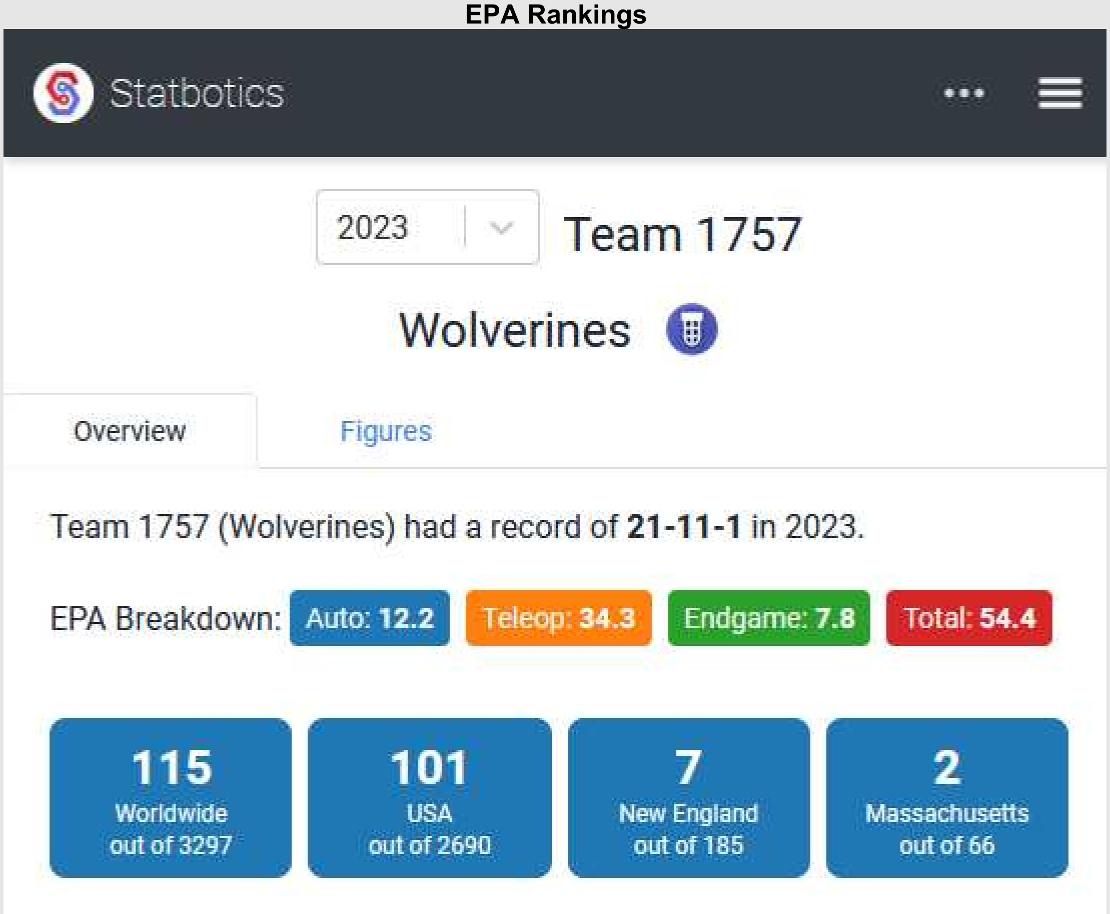
We hope you enjoy this brief look at the design process and technical details that went into this robot, and if you have any questions, look for one of our team members in the stands, in the pits, or on the field. We are always ready to share the knowledge we have gained and share a few hard-learned lessons we learned along the way.

DCMP Update

So it has been a whirlwind of a season so far, after meddeling performance at Greater Boston district we went on win the WPI District Event. Not only were we Alliance captian of the the #2 alliance, we also won the Engineering Inspiration award at WPI. Though out this document you will find various updated information featuring design changes/Repair/modifications that were made during the competition season.

Competitive Record Though District Play:
22-11-1
Greater Boston District - Deans List Semi-Finalist: Sean Tao
WPI District Event – Winners
WPI District Event – Enginnering Inspiration Award Winnerd

EPA Rankings



World Championship Update
We thought our seson couldn't get any better then taking home the teams first ever blue banner at WPI. We were wrong. We came into Wilson Division at New England Championship a solid middle of the pack Contender, however we quicky proved why we were there, our robots consistent and Reliable play led us to take #1 overall at the end of Qualifications, after picking the highest rated offensive bot on the field 176 Aces High, we picked up 1699 Robocats to round out a great alliance. We went undefeated in the Wilson Divison playoffs, taking home another blue banner before taking on the Mier Divison winners for the New England District championship. With the Championship Tied 1-1, we went into a nail-bitting sudden death match where we came out on top.

Please review our OA thread on Chief Delphi for more details.

Competitive Record Though District Championship Play:
39-13-1
Greater Boston District - Deans List Semi-Finalist: Sean Tao
WPI District Event – Winners
WPI District Event – Enginnering Inspiration Award Winnerd
NE Championship – Wilson Divison – Winners
NE Championship – Wilson Division – Excelence in Engineering
New England District Championship - Winners

EPA Record

TABLE OF CONTENTS

GAME ANALYSIS

Every FRC season starts the same way; we gather together as a team, watch the kickoff stream, then hunker down and break down the game in back-to-back 8-hour build sessions. The hope is that by the time we walk out the door on Sunday night, we understand the game and know what we are doing.

After carefully considering the different ways you can score points, we concluded that placing GAME PIECES on the NODES was the most critical ability in this game, with it having the highest potential points available. Without the ability to DOCK and ENGAGE, however, it will be virtually impossible to remain competitive due to the lack of ranking points.

After two days of deliberation, these are the design Requirements we settled on.

DRIVE

- Need to be a Small Bot – The smaller the bot, the easier it is for 3 robots to balance on CHARGE STATION
- Need a low center of gravity
- Need to be able to drive and balance on the CHARGE STATION.
- Preferably autonomous balancing on CHARGE STATION
- Use of vison (April Tags) to provide feedback to the onboard odometry system
- Use of vision to identify and seek out game pieces on the field.

ARM

- Arm needs to be strong and durable
- Use Encoders on the input and output of gearboxes to monitor and minimize backlash.
- Either 2 or 3 Degrees of Freedom Further testing will be needed.
- Needs to score at all 3 levels BOTTOM, MIDDLE and TOP Nodes.

INTAKE

- Quickly acquire GAME PIECES (Touch It – Own It)
- MUST pick up CONES and CUBES from the LOADING STATION
- MUST pick up CUBES and upright CONES from the ground.
- Would like to be able to pick tipped-over CONES from the ground.

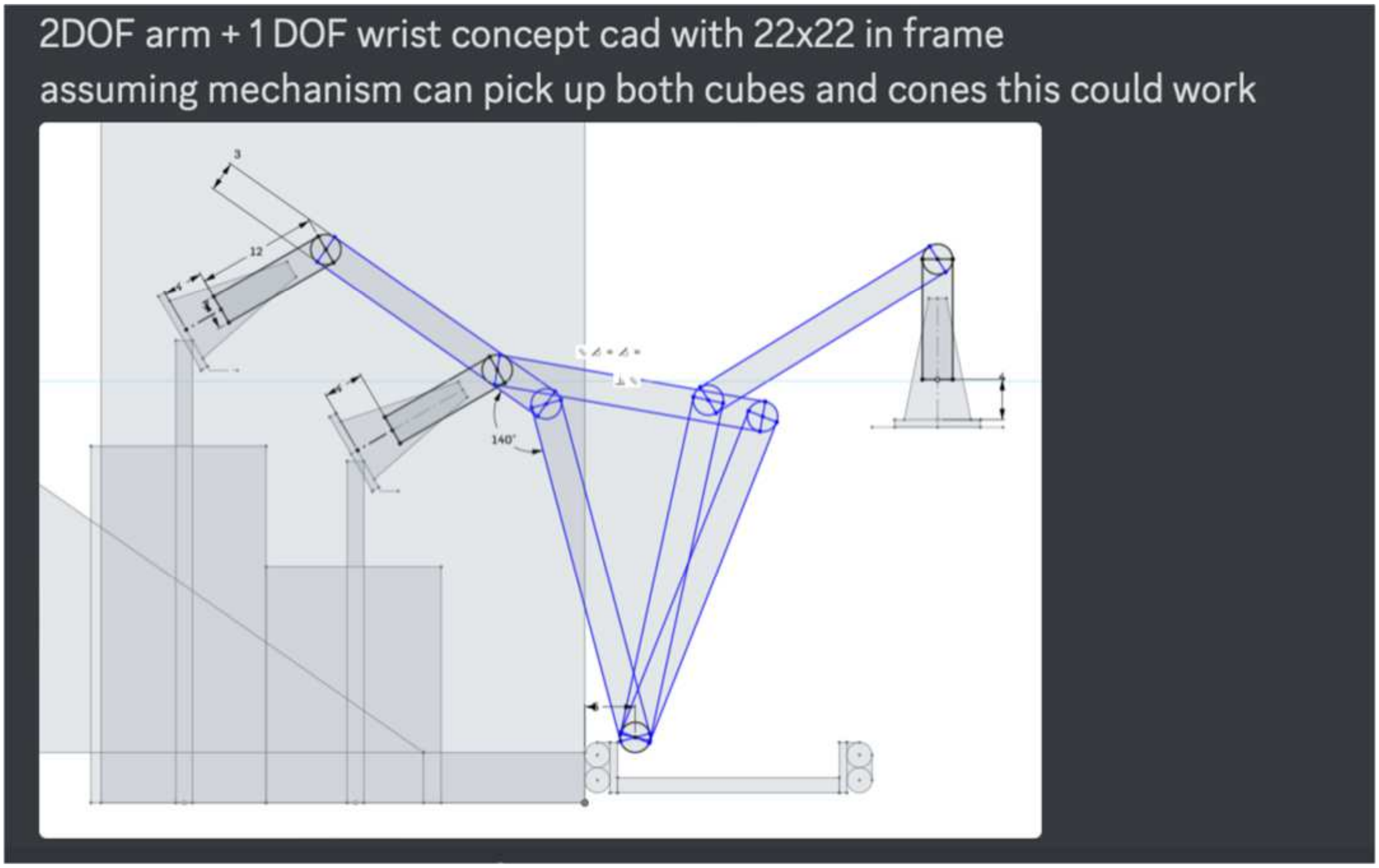
GENERAL DESIGN CONCLUSIONS

Our overall goal for the season was to be a competitive bot in district-level play and qualify for New England Championship. To accomplish this, we need to, at the bare minimum, make it to Elimination at both our district events, hopefully as an Alliance captain or 1st pick.

We approached our design as trying to build a highly reliable jack-of-all-trades bot, focusing on gaining one of the two performance-based ranking points in either match.

Inspired by the cost-effective production strategies of the Hass Formula 1 racing team and our limited team members and design resources, we prefer to use pre-engineered solutions wherever possible to focus our design resources on critical complex components.

IDENTIFYING DESIGN CONSTRAINTS

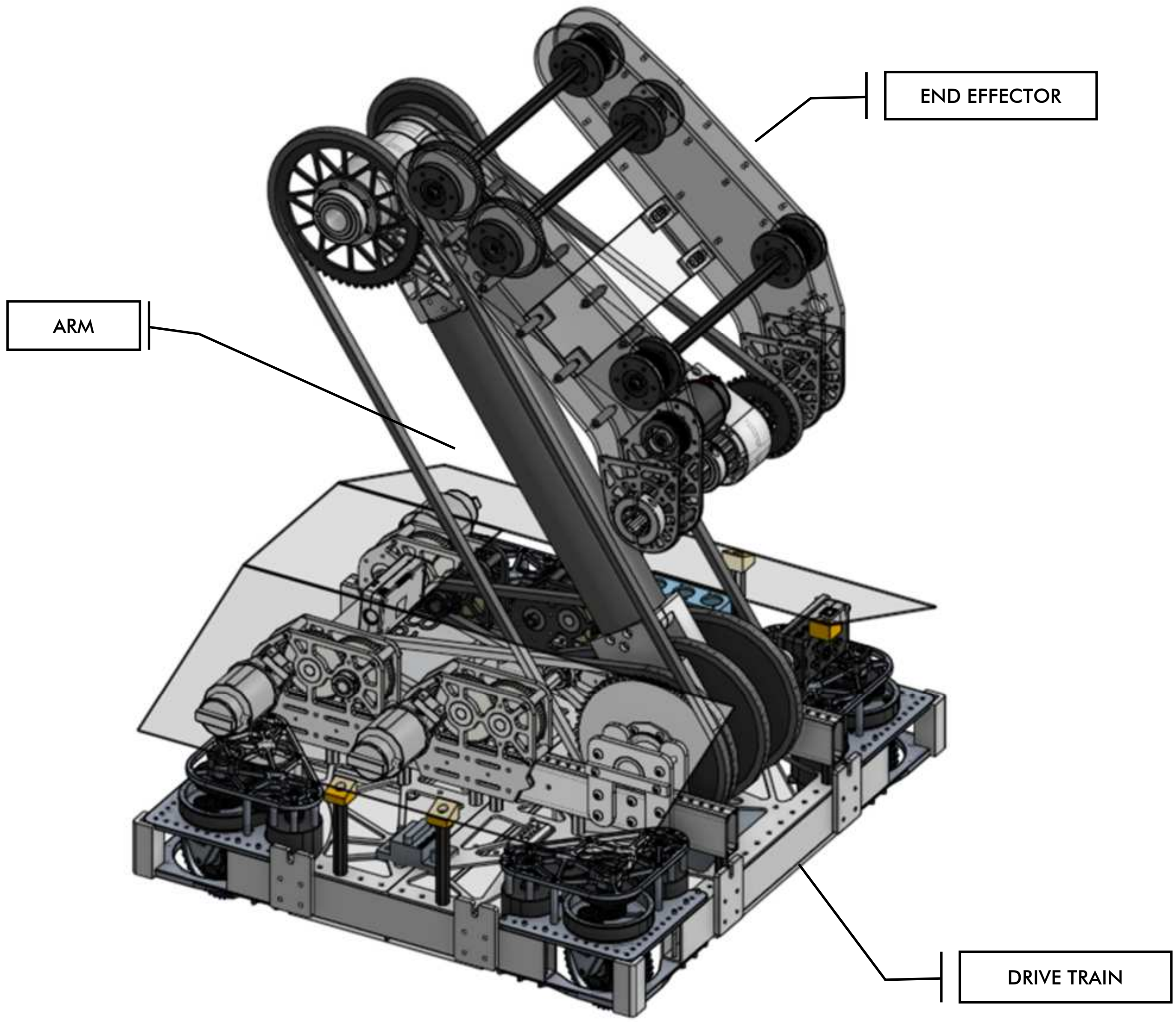


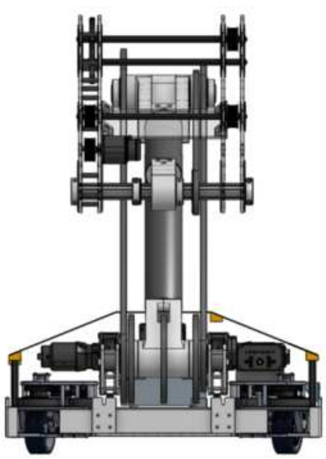
We are thinking about using an arm as a manipulation mechanism. We potentially envision a 2DOF arm + 1 DOF wrist that can pick up both cubes and cones, with a high range of motion on the wrist joint. As we can utilize the bot's movement, we do not need the Arm to move from side to side. An important note is that with an arm the starting configuration poses a good challenge, as it will need to fit inside of the robot's frame before activation. We have found that the shoulder joint only needs to move 90 degrees max, the elbow joint 210 degrees, and the wrist joint somewhere like 270 (at least in the configuration, lots to play with) to achieve all necessary motion.

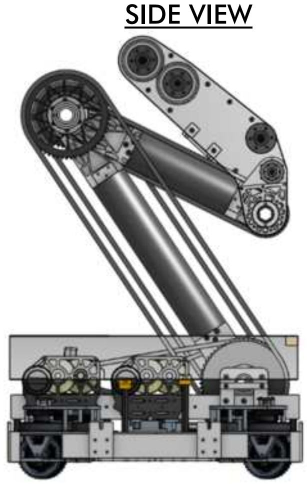
THE 1757 RAPID DEVELOPMENT MODEL

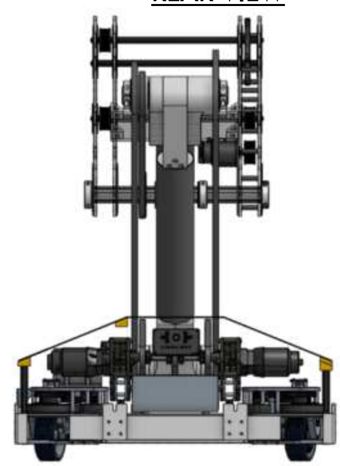
- Clearly Identify the design requirements of the system
- Design and Build a prototype that can be used to test design assumptions and Test
- Use what we learned from testing to develop a final design
- Fabricate final version and intergate into overall robot systems

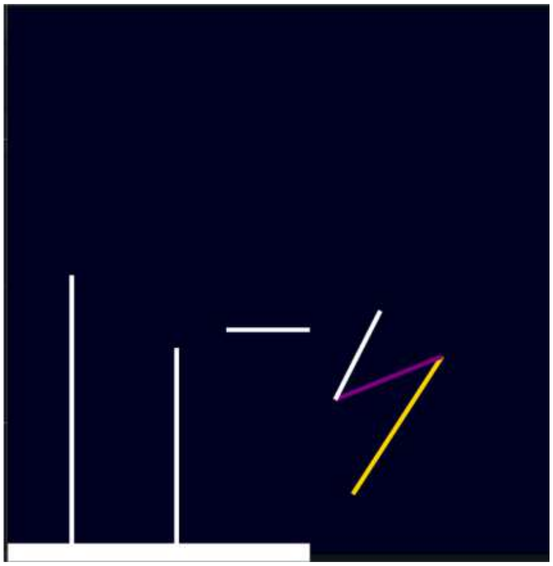
FINAL ROBOT DESIGN



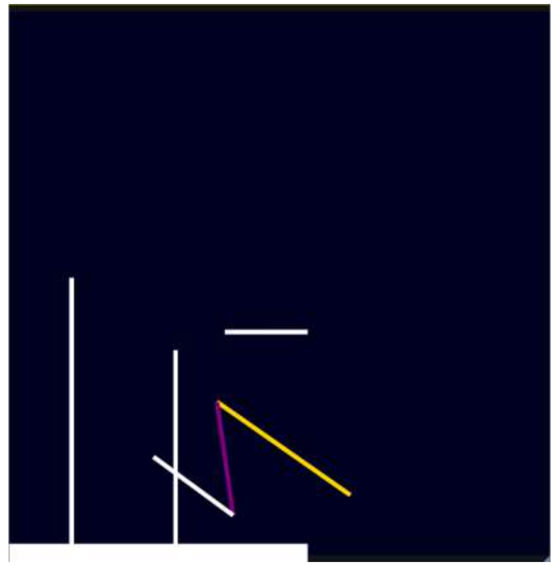




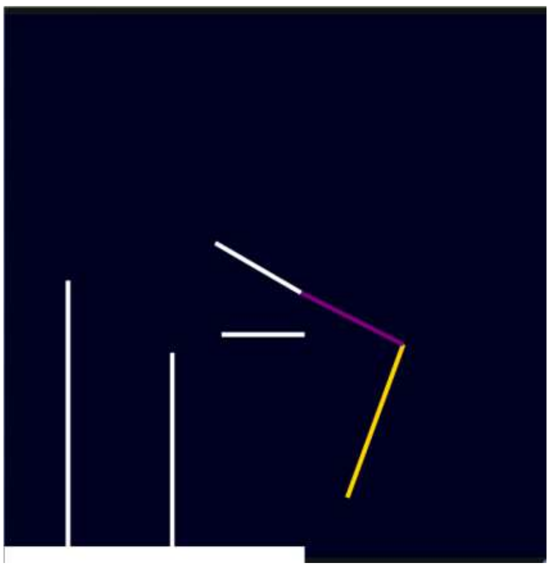




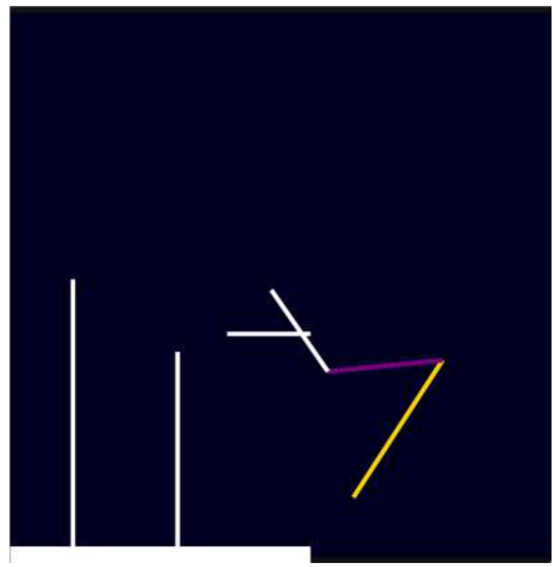
DEFAULT CONFIGURATION



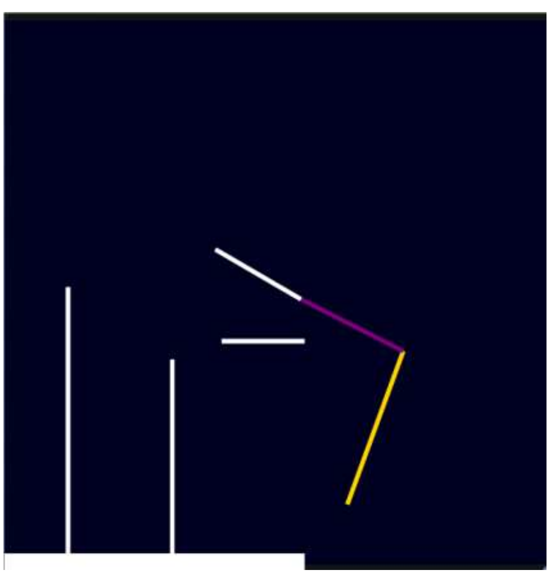
FLOOR PICKUP



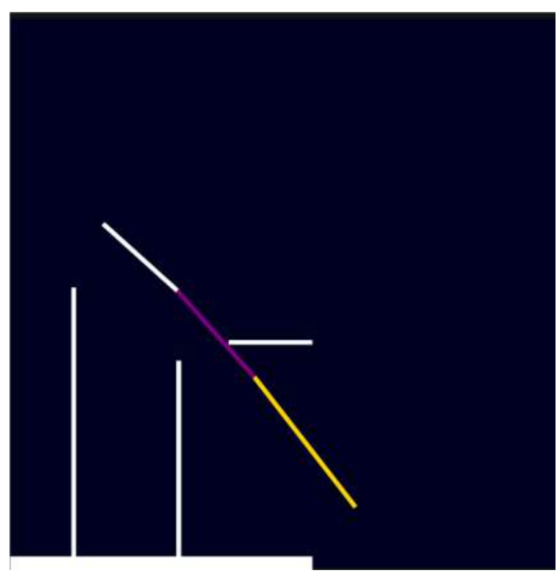
SUBSTATION PICKUP



SCORE – BOTTOM



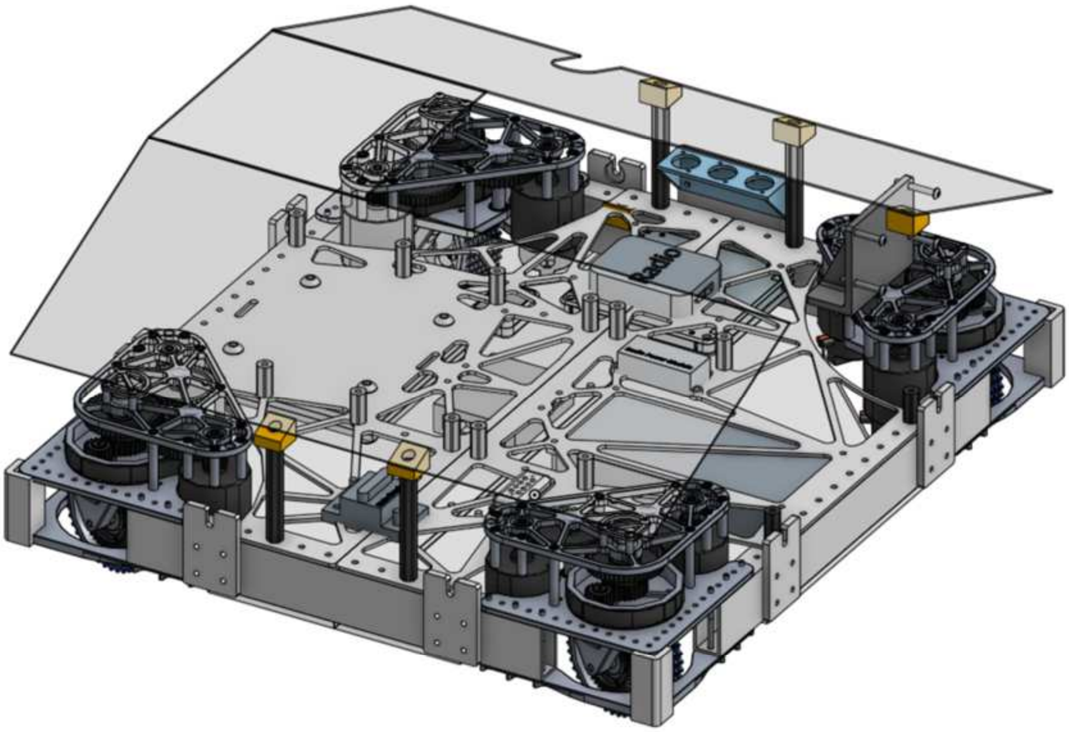
SCORE – MIDDLE

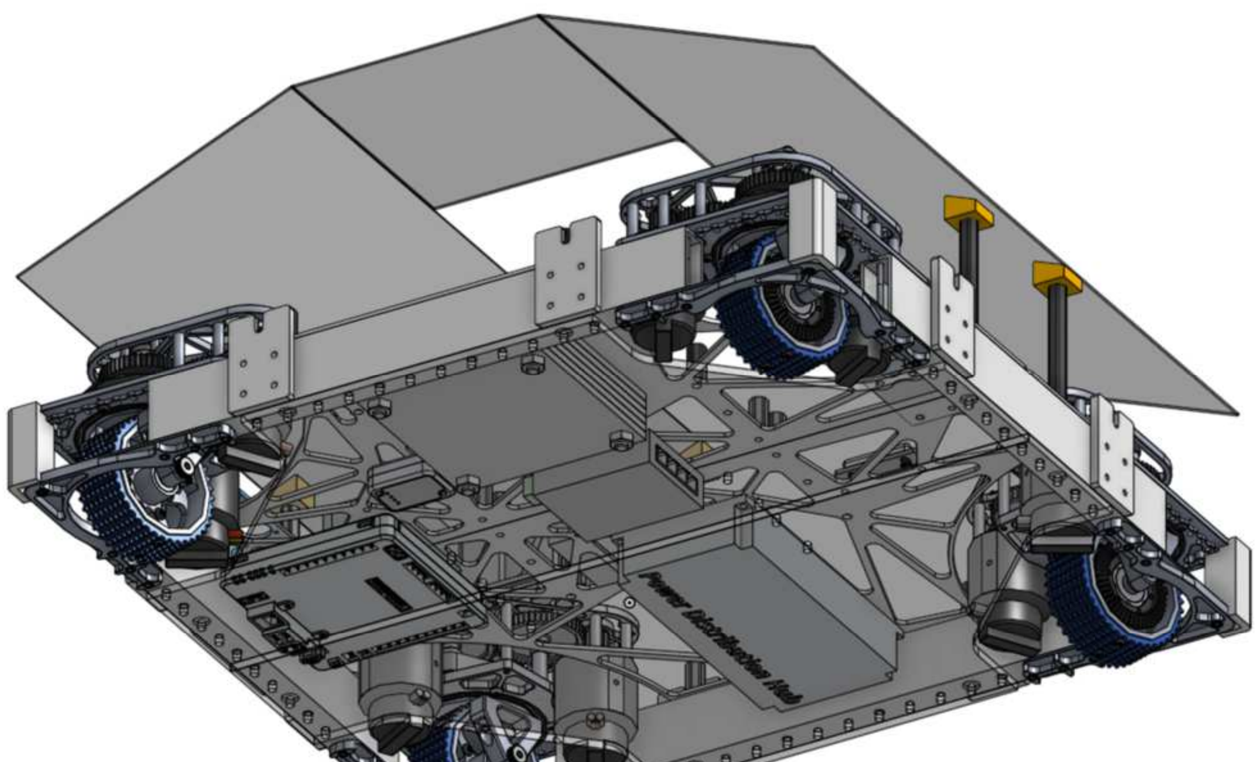


SCORE - TOP

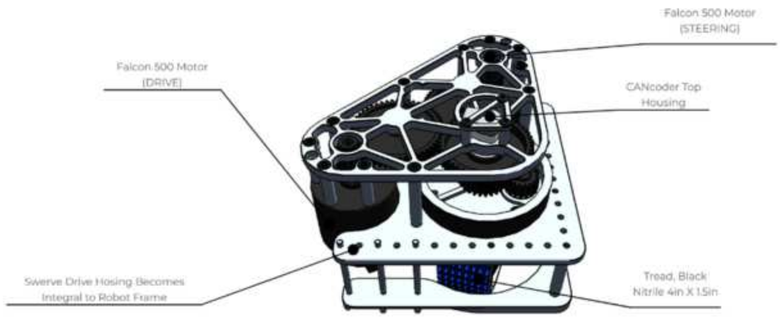
Above – 3D Simulation of Arm Joints in all of its various Arm Configurations

MAJOR SYSTEM #1: DRIVE TRAIN





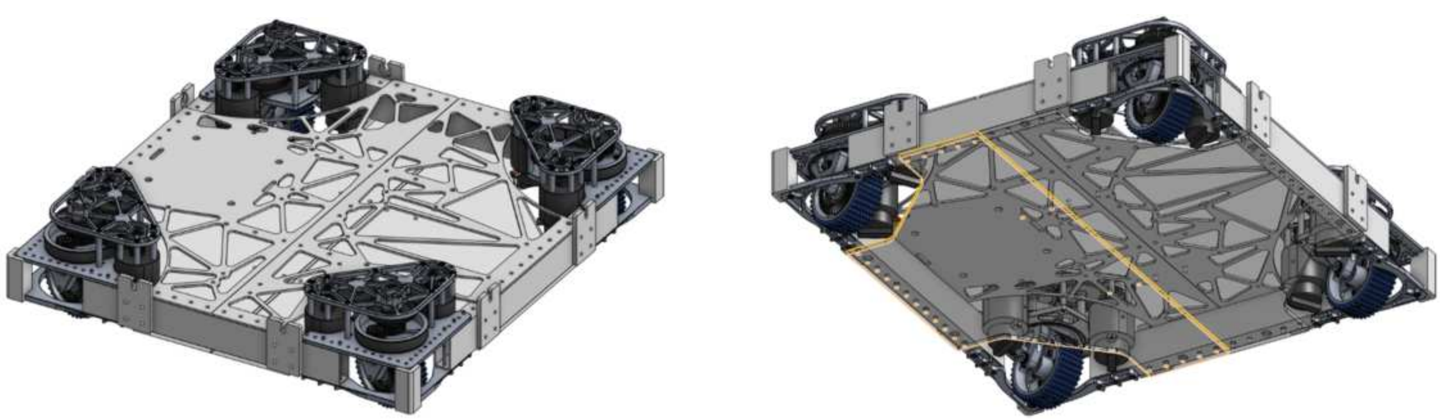
1.1 - SWERVE DRIVE MODULES



Last season was our first competitive season using swerve drive and we could not be happier with the results, because of the equal power between steering and driving there are none of the performance trade offs inherent in other drive systems. We can still run circles around the field when we need to and we can still push another robot across the field when they are in our way. One of our favorite features exclusive to swerve drive is what we call the park feature, by turning all 4 wheels to a 45° angle relative to the corners of the robot the robot effectively parks itself in place and wont move, another robot can push against us all match long and we wont move. Last year we used Swerve Drive Specialties Mk4 units, and this season we upgraded to the newly released Mk4i units. This revised design points the motors downward into the bot instead of mounting above the module. This allowed us to eliminate ≈2" of vertical space in our robot between the drive frame and the major systems.

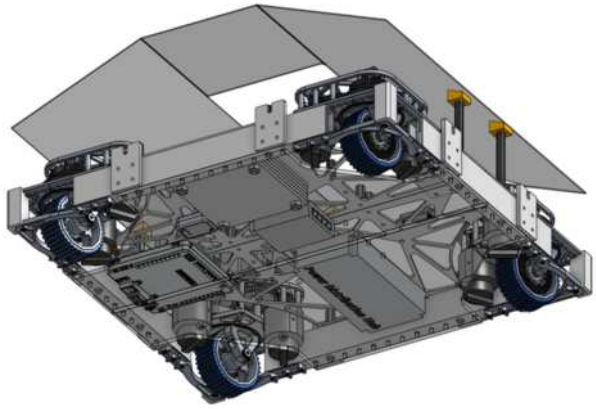
SDS MK4i Swerve Modules	
Powerplant	Falcon 500
Gearbox Configuration	L2
Overall Gearbox Ratio	6.75 : 1
Unadjusted Free Speed	16.3 ft/sec

1.2 - ROBOT FRAME



Left – CAD – Isometric Top View of robot frame
Right – CAD – Isometric Bottom View of Robot frame

1.3 - ELECTRONICS SUBSYSTEM



One thing we struggled with on our 2022 robot was how inaccessible most electrical components were. On our 2022 robot, all the electronics sat in a belly pan at the base of the bot, and the only way to access most of the components required you to remove the majority of the Robot Systems. Inspired by another team's design from 2022, we decided to hang all of our electronic components upside down and face the ground. Now to access the electronics, we tip the bot on its side, remove the ¼" protective polycarbonate plate, and you have full easy access to all the electronics components. (Credit to Team 125 for the Idea, they have the thanks of a grateful drive team and Pit Crew)

ELECTRONICS SYSTEM MAJOR COMPONENTS

- (1 ea) National Instruments - RoboRio 2
- (1 ea) REV Robotics - Power Distribution Hub
- (1 ea) Navex 2 – RoboRio MXP expansion Board
- (1 ea) CTRE CANivore
- (1 ea) CTRE CANdel
- (1 ea) BrainBoxes – SW-015 5 Port Gigabit Switch
- (1 ea) Generic Passive POE Injector
- (1 ea) Limelight 2 Camera
- (12 ea) Falcon 500 Motors
- (1 ea) REV Robotics Sparkmax brushless motor controller
- (1 ea) REV Robotics NEO 550 Brushless DC Motor
- (1 ea) Open Mesh Access Point [Insert Model Number]

1.4 - TESTING PORTS

We added a convenient patch panel to the upper side of the robot to allow for quick access to essential data ports when we don't want to access the underslung electronics.

PATCH PANEL SLOT 1 – USB TYPE A

This slot connects to one of the USB Type A ports on the RoboRio. This typically has a USB flash drive plugged in. During a match all the system logs are copied to the USB drive. After a Match, the USB drive can be pulled and opened up in AdvantageScope on the debug machine for post-game analysis. It's our version of a Blackbox on an airplane.

PATCH PANEL SLOT 2 – USB TYPE B

This connects to the USB Type B Port on the RoboRio—a redundant method for tethering the robot for control and debugging at events.

PATCH PANEL SLOT 3 – RJ45 CONNECTOR

This connects to the Ethernet Switch Via CAT5e for network access. Used for tethered connections to the bot during testing. Ethernet tethering is preferred, but we have encountered software reliability issues in the past.

DCMP Update

At the Revere District event we ran into serious problems tethering to the robot via ethernet and via USB B. we traced the ethernet tethering problem to a problem with the network configuration issue on the driver station laptop. We were unable to determine a definite cause of the USB-B connection issue, but, we think it most likely to to poor quality of the 90° usb connector used on the robot. From that point on we connect a USB-B cable directly into the port on thr

At the same time we realized we needed a button to manually put the arm motors in coast mode for serviceability when the bot is not connected to the driver station. Since we are no longer using the USB-B testing port we replaced it with a momentary push button switch.

1.5 – CAMERA/VISON SYSTEMS

Limelight 2 – Camera

We are utilizing a Limelight 2 Camera for a variety of tasks on the robot mostly devoted to sensor fusion and automation of systems using computer vision. The Limelight's field of vision (FOV) is essentially parallel to the floor and at the height of the April Tags.

Please refer to the Software section of this document for more information on how we use the limelight and April Tags to improve the Onboard odometry of the robot.

1.6 – COUNTERWEIGHT

Not originally intended as part of the design, upon testing of the robot with the Arm fully extended in the scoring position, we realized that robot was prone to falling forward. To resolve this, we looked to add ballast to the bot. First we thought of lead but didn't want to deal with the potential health risks of improperly encapsulated lead. We investigated tungsten; however, a review of the current price of tungsten plate (≈$40/kg) quickly ruled it out as a potential candidate. We settled on 6" x6" x1/4" steel plates mounted directly under the robot battery. After testing with different #'s of the plate, we decided on 6 Plates with a total weight of ≈25 lbs. Now the bot is highly stable even when the Arm is fully extended.

1.7 – PROTECTIVE COVERS

We added protective covers that Slope away from the central arm structure down to the bumpers. Not only do these plates provide a valuable location to display all of our great sponsors they serve to prevent errant game pieces from getting stuck inside the robot during a match.

DCMP Update

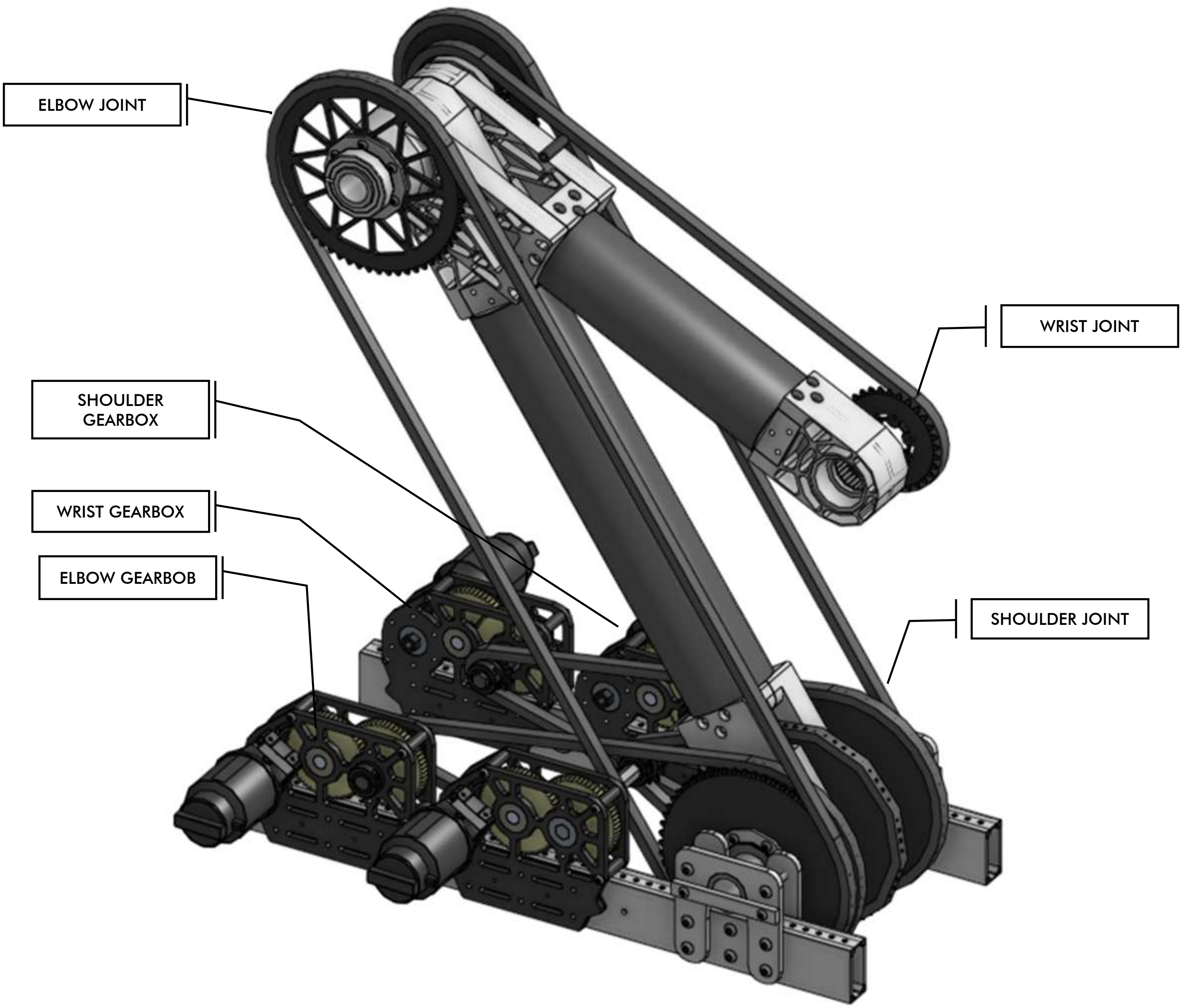
Originally the protective covers were only hled on with 3M™ Dual Lock™ SJ3560, this material is nice because it very strong but eaisily removable. Durring qualifying matches in Revere however, these panels kept falling off and dragging around the fieild. The Dual Lock strips were reinforced with zip ties and these held through all of playoffs in revere, and all of qualifications at WPI. Then in Playoff Matches we shered off 3 of the metal standoffs holding up the protective covers. We made quick repairs to keep going however prior to DCMP we will be swaping out all the ½ thunderhex standoffs with 1” 80/20 extrusion with hardened bolts for strength.

1.8 – GAMEPIECE INDICATORS

One of the hardest things to do in a match is how to signal between the driver station and the human player what game piece you want them to load into the robot. People use Hand Signals, Colored pieces of paper or guess. We wanted to take the guesswork out of the equation, so we mounted 2 LED Strip lights along the top of the protective covers. The driver controls what color these strips are so he can communicate to the human player which game piece to feed to the bot – Yellow for Cones and Purple for Cubes.

Left – Robot Displaying "I Want a Cone"
Right – Robot displaying "I Want a Cube"

MAJOR SYSTEM #2: ARM



2.1- MOTORS & GEARBOXES

To keep the robot's center of gravity low and keep the Arm as simple as possible we decided to locate all of the heavy motors and gearboxes at the base of the superstructure.

GEARBOX # 1 – SHOULDER GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into Custom Gearbox based on WCP Gearbox design.

GEARBOX # 2 – ELBOW GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into COTS WCP Single Speed Gearbox.

GEARBOX # 2 – WRIST GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into COTS WCP Single Speed Gearbox.

2.2 - CHAIN DRIVE

Using a combination of dead and live axels we transfer the power of the gearboxes up though the Arm to power each of the individual joints. For Reliability and durability, we chose to use #35 roller chain rated for 11,000 lbs of force.

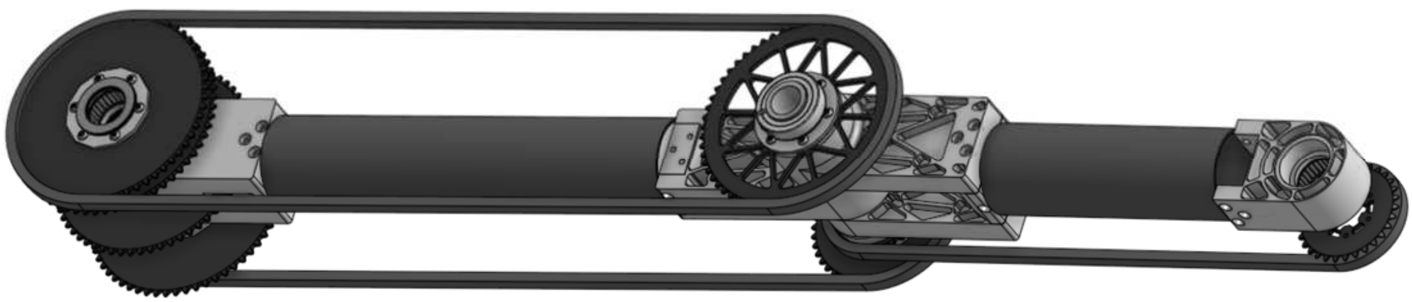
Below is a summary of the different chain runs on the Arm

CHAIN DRIVE 1 – SHOULDER
Shoulder Gearbox Output 12t Sprocket → 60t sprocket on Shoulder (Dead Axel)

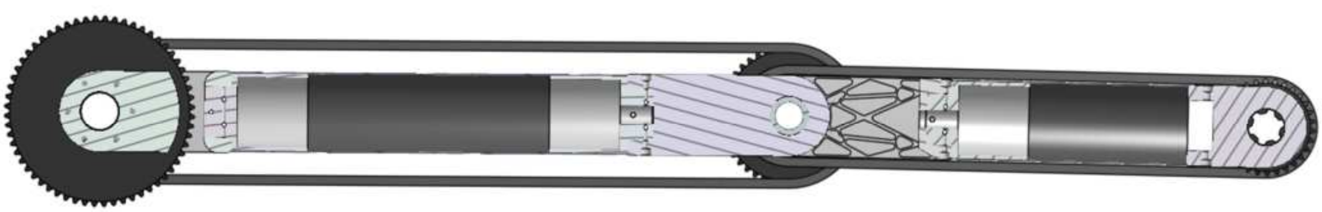
CHAIN DRIVE 2 – ELBOW
Elbow Gearbox Output Shaft 12t Sprocket → 60t Sprocket on Shoulder (Live Axel) → 70t Sprocket on Elbow (Dead Axel)

CHAIN DRIVE 3 – WRIST
Wrist Gearbox Output Shaft 12t Sprocket → 60t Sprocket on Shoulder (Live Axel) → 60t Sprocket on Elbow (Live Axel) → 40t Sprocket on Elbow (Live Axel) → 32t Sprocket on the Wrist (Dead Axel)

2.3 - ARM STRUCTURE



Above - 3-D View of Outstretched Arm



Above - Section View Though Center of Carbon Fiber Arm

CARBON FIBER ARMS

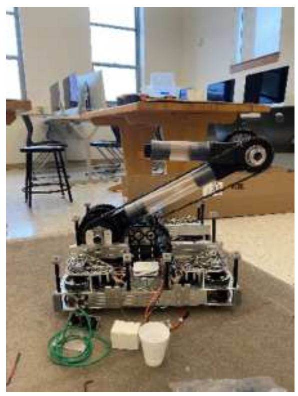
We chose to use carbon fiber tubes as the main structure of the Arm due to its strength and lightweight, the more weight we could save on the Arm the lower we could push the robot Center of Gravity. Carbon Fiber tubes are a stock McMaster item 3" Ø. Carbon Fiber is Epoxy bonded to 3" hollow aluminum plugs bolted to the aluminum joints.

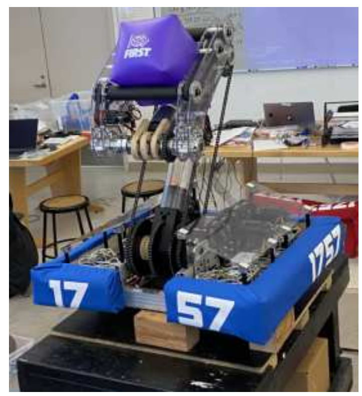
Left – Final carbon fiber arm links after final glue up

Right – Final aluminum plugs used in the ends of the carbon fiber tubes

3D Printed and Polycarbonate Prototypes

Because we knew the carbon fiber and machined aluminum would take time and money to manufacture, we heavily used 3D-printing to make prototypes of the Arm and test and confirm critical geometry before placing final fabrication orders. These prototypes are two fragile to be used on a competition bot but worked well for their intended purposes. We learned very Important lessons about where the concentrations of forces were along the axels and what parts needed reinforcement.





Left – 3D printed Prototype of the wrist joint, printed on a FormLabs 2 SLA Printer

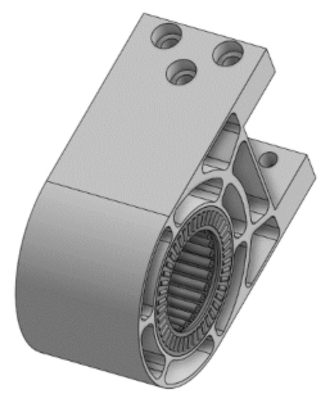
Center – Polycabonate Prototype arm Mounted on bot for the First Time

Right – Fully Assembled "Alpha" Robot build

2.4 – JOINT STRUCTURE

CAD - Shoulder Joint

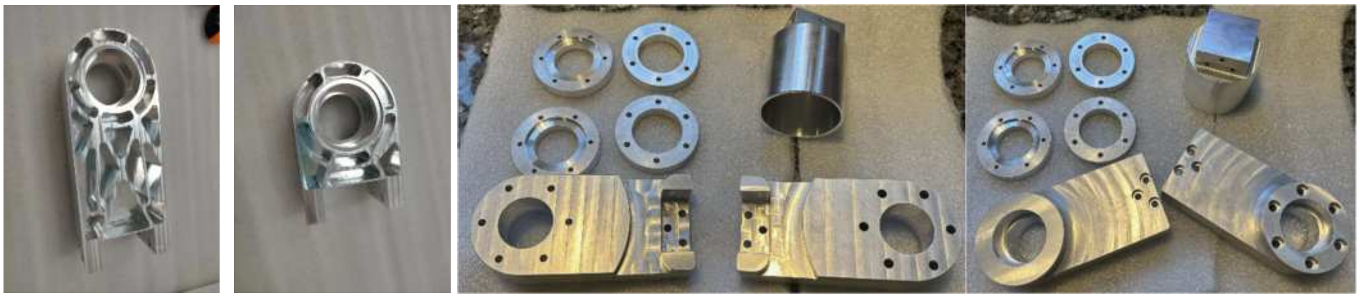
CAD - Elbow Joint



CAD - Wrist Joint

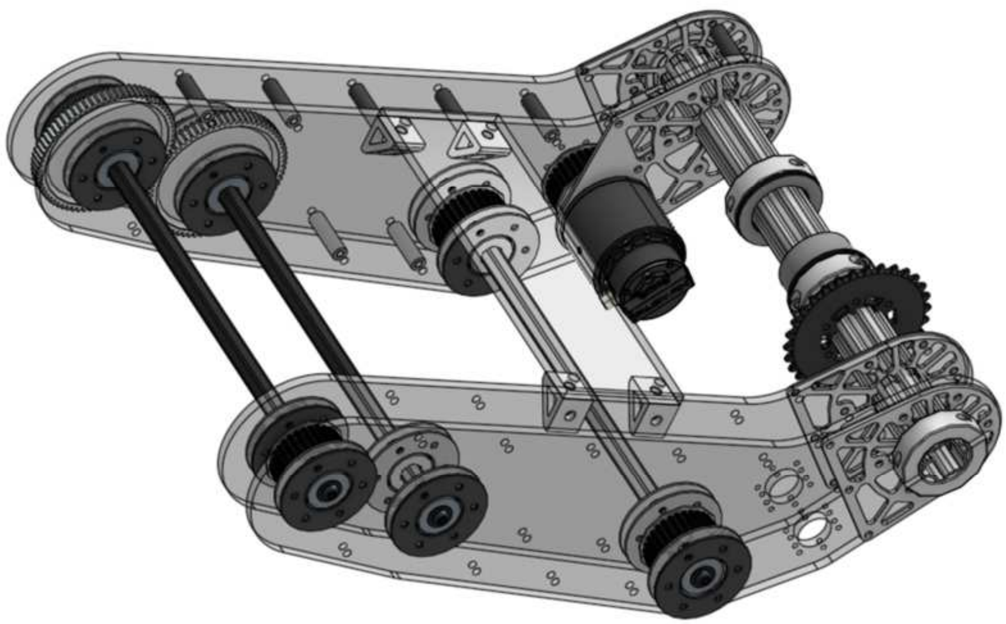
Needle and Thrust Bearings

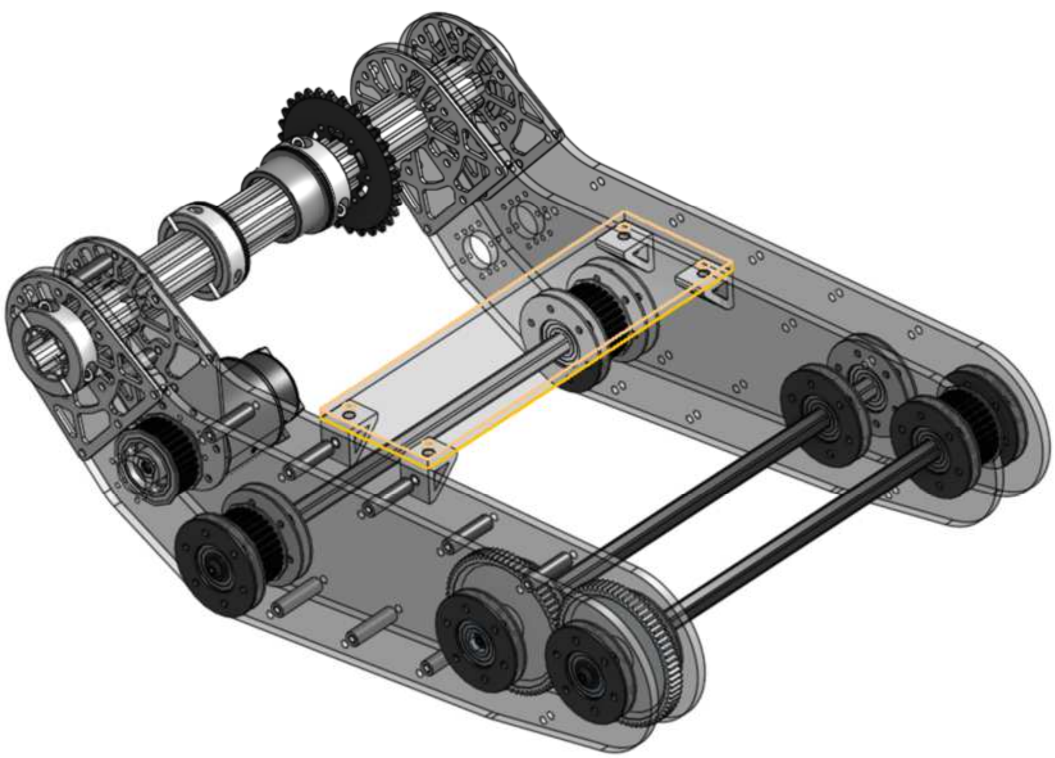
Used in All three joints to allow for smooth rotary motion in each joint.



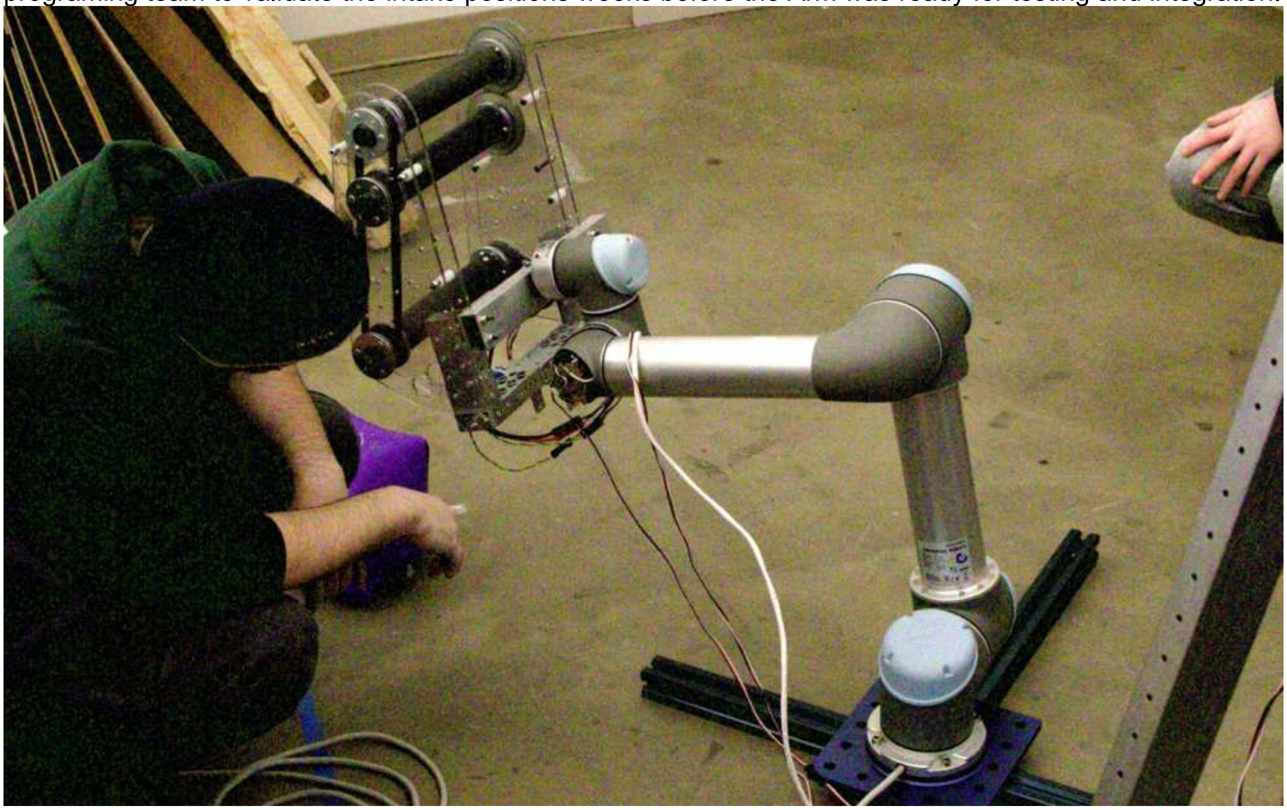
Photos - Final Machined Parts

MAJOR SYSTEM # 3: END EFFECTOR





INTAKE PROTOTYPING – FUN WITH INDUSTRIAL ROBOTS
We had a lot of fun prototyping this mechanism, since it was the first major mechanism that we finished we had lots of time to put it trough its paces. One of our mentors has access to a Universal Robots UR3 robot and brought into our lab during one of our weekend build sessions (see images below). This actually proved to be essential because it allowed our programing team to validate the intake positions weeks before the Arm was ready for testing and integration.



3.1 – ROLLERS

The rollers we are using are Vex VersaHub Rollers with ¼" neoprene tubing stretched to cover them, they are very grippy and hold the cube verry securely. We started with the dimensions of the everybody roller for our prototype then made some modifications before settling on final separation differences. The neoprene tubing is undersized for the OD of the polycarbonate roller. We learned a fun trick to clamp off one end of the neoprene tube and inflate it with an air compressor to stretch it over the polycarbonate tube, when the air is released, it makes a perfect friction fit between the Neoprene and the polycarbonate. We have had no detectable slippage after weeks of testing with the rollers.

DCMP Update
After 33 competitive matches one thing is clear, we have problems picking CUBES up off the floor and in order to maintain our competitive edge at DCMP we know we need to be able to get CUBES up off the floor. We thing the majority of the problem is related to how narroe our end effector is, the original design was only 1” wider than the width of the CUBE. To aliviate this issue we are planning to widen the end effector by 3”.

3.2 – MOTORS & GEARBOXES

The intake is powered by a REV Robotics NEO550 Brushless motor into a REV Robotics Ultraplanetary gearbox. The motors small size is nice however because we mounted the Sparkmax Motor controller on the intake as well there is no significant weight savings compared to using a Falcon 500 with an integrated Talon SRX. We may end up swaping this out for simplicity sake in the future.

REV Ultraplanetrary	
Powerplant	NEO550
Gearbox Configuration	4:1, 5:1
Overall Gearbox Ratio	20:1

SOFTWARE

SOFTWARE: OUR DEVELOPMENT ENVIRONMENT

WPILib

The perineal stalwart, we still rely on core elements of WPIlib for robot communications and debugging. WPILib's new Logging features have greatly enhanced our Debugging capabilities

RobotPy

We have found that students have a lot easier time learning python then they do Java or C++ so with the growing support for RobotPy we migrated our Codebase from Java to Python in 2020. As of this March we are an official contributor to the RobotPy project

GitHub

Without Github our level of remote work and collaboration just wouldn't be possible.

SOFTWARE: NEW AND UPDATED TOOLS THIS YEAR

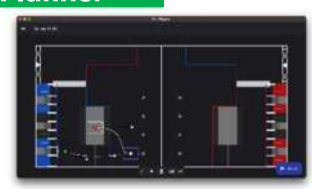
AdvantageScope

WE LOVE ADVANTAGE SCOPE! Not only does it log *everything* but does it in away that is intuitive and easy to review. No more searching though 10000 lines of log files to find the one piece of information we need. Huge thanks to team 6328 for building such a great tool.

PhotonVision

We are using PhotoVision as our native development framework for Computer vision due to its growing wide support inside the FRC community. It does not include native support for RobotPy however so as an offseason project our lead programmer wrote a custom wrapper for PhotonVivion so it can work inside our RobotPy environment

PathPlanner



Last year we used PathWeaver, but we were disappointed in the lack of native support and increased complexity in the development stack so starting with the off season we transitioned all of our Autonomous path planning to PathPlanner. We had much fewer issues with this system.

SOFTWARE: DRIVE

Taking off of last year, the drivetrain codebase has stayed the same. We are running field oriented drive with robot relative rotation to allow for quick maneuverability. A button to align to the nearest 90 degree angle was added to help with driver alignment. This state slightly reduces the speed and snaps the angle of the robot in order to have perfect alignment to the double substation, single substation, and grid every time. For our automated balance sequence, we work in robot relative space on the robot relative gyro.

A Brief Tangent - Absolute Relative Drive

Last year our lead programmer had a new idea for drive control, an absolute relative drive. The common swerve drive control method was to have a field relative translation for the bot, and a robot relative rotation. What this meant is a left input on the rotation axis would result in the robot rotating to the left at a constant speed. A translation action was not affected by rotation but instead was in "field relative" space. The difference of absolute drive is that the rotation is also field relative. A left input on the rotation stick will yield the robot turning to face left. This year we expect this type of robot control to be very important for drivers when they have to be able to turn to specific positions for collection and scoring on swerve drives. You can see this in action in any one of our videos from last year. Having fixed controlled rotation will allow for precise driver input and less fiddling with controls when cycle time is very important.

The drivers have also experimented with alternate driving methods on swerve to get used to interesting control schemes such as a curvature drive, standard tank drive, standard field relative drive, and full robot relative drive.

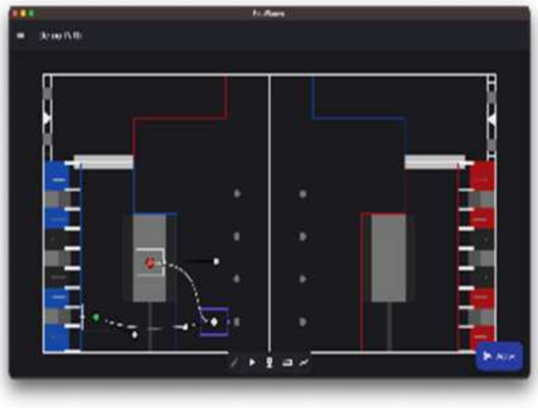
SOFTWARE: INTAKE

The intake is using a state machine in order to regulate its expected behavior. There are 2 enumerable values: one for the gamepiece intended direction (intake, outtake, hold) and one for the desired gamepiece type (cube, cone). The transition between each state is dictated by a user input to any given category. If no input is given, the system holds its position and keeps the desired gamepiece remembered. The state value of the desired gamepiece is displayed to the driver and to the human player through pulsing leds of the respective color.

SOFTWARE: ARM

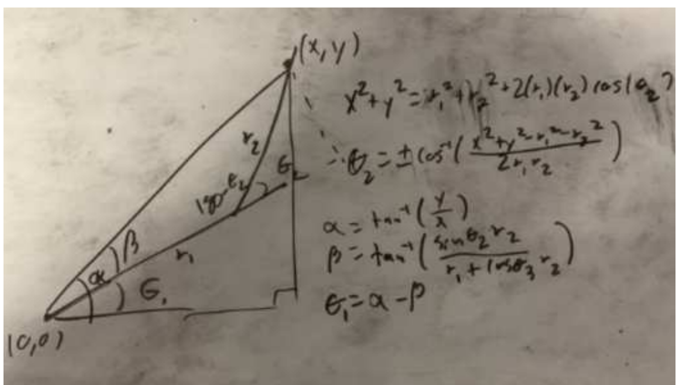
A triple jointed arm is no easy feat in order to program smoothly. From cad, states are given about the end effector's desired position and rotation relative to the floor. From there we use inverse kinematics to determine each per-joint relative rotation at any given position. A cartesian control on the wrist joint's position is added using a trapezoidal PID profile to lay out a path for the Arm to follow. For the wrist it has another trapezoidal PID profile controller. When going state per state on the Arm's motion, we check for if the relative angle goes over software end limits in order to prevent running the Arm into itself. These are done in joint relative space. Since the Arm is controlled from the base through chain and sprockets there is a virtual 4 bar created in which the rotation of any given joint is given relative to the ground. These are converted into motor space and passed onto each motor where they have a position PID controller onboard. For simulating the system we are using mechanism2d to view the expected values of the Arm and be able to run through positions. This simulation first approach has allowed minimal revision and a solid foundational codebase that is mostly complete before the bot is finished. Furthermore the position of the Arm is logged in 3d with advantagescope based on the position it believes the bot is in. Logging was also a priority for a complex system and therefore we log the instructed position and actual position per joint and each end position in cartesian space.

SOFTWARE: AUTONOMUS



We experimented in the offseason with pathplanner and use it extensively for our autonomous. Each necessery command is placed into a potential pool of events for pathplanner to fire. At the beginning a sequence determined solely in pathplanner is fired. Going off of last year we had a large time gap in order to make slight changes so instead for this year we are using the waypoint system and opting to have any given autonomous contained solely in pathplanner. This has increased our ability to construct autos and tweak any given aspect as needed. For the library itself of parsing, the lack of first class robotpy support meant we had the freedom to reimplement autonomous however we pleased based on the path. We follow a changing trajectory and the swerve drive using onboard odometry and a weighted vision estimate determines its bot position relative to the global field and follow through it between each section.

SOFTWARE: SIMULATION



armsubsystem.py

```
    def setEndEffectorPosition(self, pose: Pose2d):

        twoLinkPosition = Translation2d(
            pose.X() - constants.kArmwristLength * pose.rotation().cos(),
            pose.Y() - constants.kArmwristLength * pose.rotation().sin(),
        )

        endAngle = math.acos(
            twoLinkPosition.X() * twoLinkPosition.X()
            + twoLinkPosition.Y() * twoLinkPosition.Y()
            - constants.kArmtopLength * constants.kArmtopLength
            - constants.kArmbottomLength
            * constants.kArmbottomLength
            / (2 * constants.kArmtopLength * constants.kArmbottomLength)
        )

        startAngle = math.atan2(twoLinkPosition.Y(), twoLinkPosition.X()) -
math.atan2(
            math.sin(endAngle) * constants.kArmtopLength,
            constants.kArmbottomLength + math.cos(endAngle) *
constants.kArmtopLength,
        )
        wristAngle = pose.rotation().radians() - startAngle - endAngle

        bottomArmEncoderPulses = (
            startAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kBottomArmGearRatio
        )
        topArmEncoderPulses = (
            endAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kTopArmGearRatio
        )
        wristArmEncoderPulses = (
            wristAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kWristPivotArmGearRatio
        )

        self.topArm.set(Falcon.ControlMode.Position, topArmEncoderPulses)
        self.bottomArm.set(Falcon.ControlMode.Position,
bottomArmEncoderPulses)
        self.wristArm.set(Falcon.ControlMode.Position, wristArmEncoderPulses)
```

Due to our team's resources, virtual simulation is a huge part of our ability to quickly and reliably construct the bot's codebase. Some key examples of simulation are a wrapped onto a simulated falcon motor. Given our team's extensive use of falcons on the robot, a wrapper that provides simulation support allows for the programming team to iterate much easier and creates a cleaner codebase. Each falcon is logging the values of the motor % and the encoder position, as well as an override value to allow the user to manually in simulation change the value for sensor readings. Entire robot configuration is done on a single call and the getting of velocity, position, and percent and the setting of velocity, position, and percent are easy to access functions to allow interfacing with the motors more accessible than the CTRE library. Given this robot also has a NEO550, the simulation system was adopted to have a similar interface for ease of replacement from a falcon to a motor on the intake. We geometrically derived the inverse kinematics for 3 links with a fixed Pose endjoint. Each of these poses actually allows for two configurations of the proximal 2 arm joints (they can simply be mirrored over the line created from the wrist joint to the shoulder joint, however by forcing the sign on the elbow joint they can all be consistent.

SOFTWARE: VISION

NOW WITH APRILTAGS AND PHOTONLIB

We have a vision system complete with sensor fusion for complete robot localization. Last year,we worked with our first complete vision system as a team that resulted in significantly enhanced system performance, and using apriltags will be very important to account for combined sensor error as well as for being able to reliably use sensor data for automated alignment to various points on the field such as the double substation and the grid.

THE HOW

Photonvision generates camera-relative 3d transforms of each apriltag. Since the position of the camera is known and the position of the apriltag is known, the position of the robot can be determined from a single apriltag datapoint. These transforms are fed into a RobotPoseEstimator in order to create a sense of where the robot could be at a given time, this is combined with the gyro and wheel encoder information to get an accurate sense of where the robot is on the field at any given time. This is used in other subsystems when needed, as well as results being logged to AdavantageScope through the usage of each known pose and ghost poseseepaste

GOING FURTHER

We plan on using this odometry data to have automated alignment in complete robot space for important precision actions such as placement of gamepieces on the grid and collection of those gamepieces. Autonomous will also use this data. Perhaps an automatic engagement on the charge station by using the rotation gained from the apriltags will be possible. Overall having a sense of where the robot is on the field is beneficial to aid in other systems.

ENGINEERING TEAM

WHS Faculty Advisors

James Looney & Raul Madera

Team Captain

Charley Marsland*

Team Business Lead	*Team Technical Lead*
Sean Tao	**Luke Maxwell**

Senior Mentors

Dwight Meglan | **Chris Aloisio°** | **Steve Harrington°**

Mentors

Anthony Gelsomini | **Manny Barros°** | **Sean Lendrum°** | **Mark Holthouse** *Mentor Emeritus* | **Amber Maxwell**

STUDENT MEMBERS

Safety Captain		*Graphic Design Lead*	*CAD Lead*
Jacob Kaplan		**Ivan Cai**	**Landon Bayer**
John Santasuoso*	**David Girard***	**Declan MacDonald**	**Liam McWeeney**
Alex Theofilou	**Claire Peng**	**Andrew Gong**	**Rachel Qu**
Baili Jiang	**Nolan folmar**	**Jeffery Pan**	**Melissa Yang**
Divya Siddi	**Vinny Milinazzo**	**David Confoey**	**Adrianna Cirillo**
Elizabeth Lowney	**Jacob Liu**	**Anthony Yang**	**Constantina F.**
Luke Li	**Owen Monahan**	**Erik Curlli**	**Kevin Bai**
Henry Marsland	**Nina Pappas**	**Rachel Lee**	**Hanya Yang**
Sophia Patrick	**Lauren Buza**	**Kaylee Phu**	**Elias Mukdissi**

Westwood Robotics, Inc
Board Of Directors

Steve Harrington° *President*	**Catherine Tseng** *Treasurer*	**Manny Barros°**	**Cheten Gopal** *Clerk*	**Jack Tseng**

**Denotes High School Seniors* | *°Denotes FRC Alumni*

SPONSORS

Heartlander Surgical	Westwood High School	Medtronic	Act Fasteners
Gene Haas Foundation	TE Connectivity	General Dynamics	IRA Motor Group
Bayer Familt	Capital Group	Yamaguchi Family	Plymouth Rock Assurance
Deadham Savings	Maxwell Family	Schneider Electric	Marsland Family
Porier Apliances	Franklin Templeton Investments	Jake and Joes Resturant	McGolf
Roche Bros		Capers Catering	

VENDORS

Andymark	West Coast Products	Vex Pro	Swerve Drive Specialties
Discord	Google Work	Onshape	Robo Promos
McMaster-Carr	The Home Depot	Cross the Road Electronics	MS Visual Studio
Python	GitHub	RobotPy	Java

1757

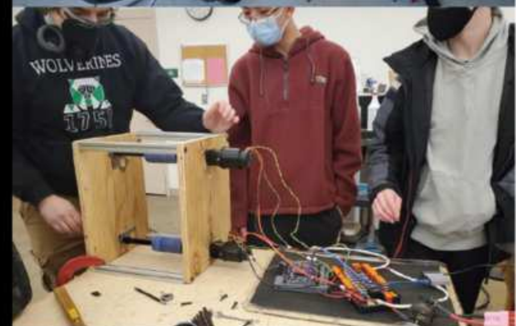
Special thanks to everyone who makes this team possible Including

Our Mentors
Our Students
Our Teachers
Our Parents

Our Amazing Sponsors

Westwood Public Schools
Emily Parks - Superintendent
Allison Borchers - Asst. Superintendent
Deb Nolan - Building Use Coordinator
Tom Carey - Director of Facilities
Ron Smith - Assistant Director of Facilities
Amy Davenport - Principal
Aishleen Flanagan - Assistant Principal
Tom Millett - Assistant Principal
Andrew Miller - Science Dept. Chair
WHS Janitorial/Facilities Staff
WHS Library Staff
The Open Alliance

#FRC1757
@WWRobotics1757

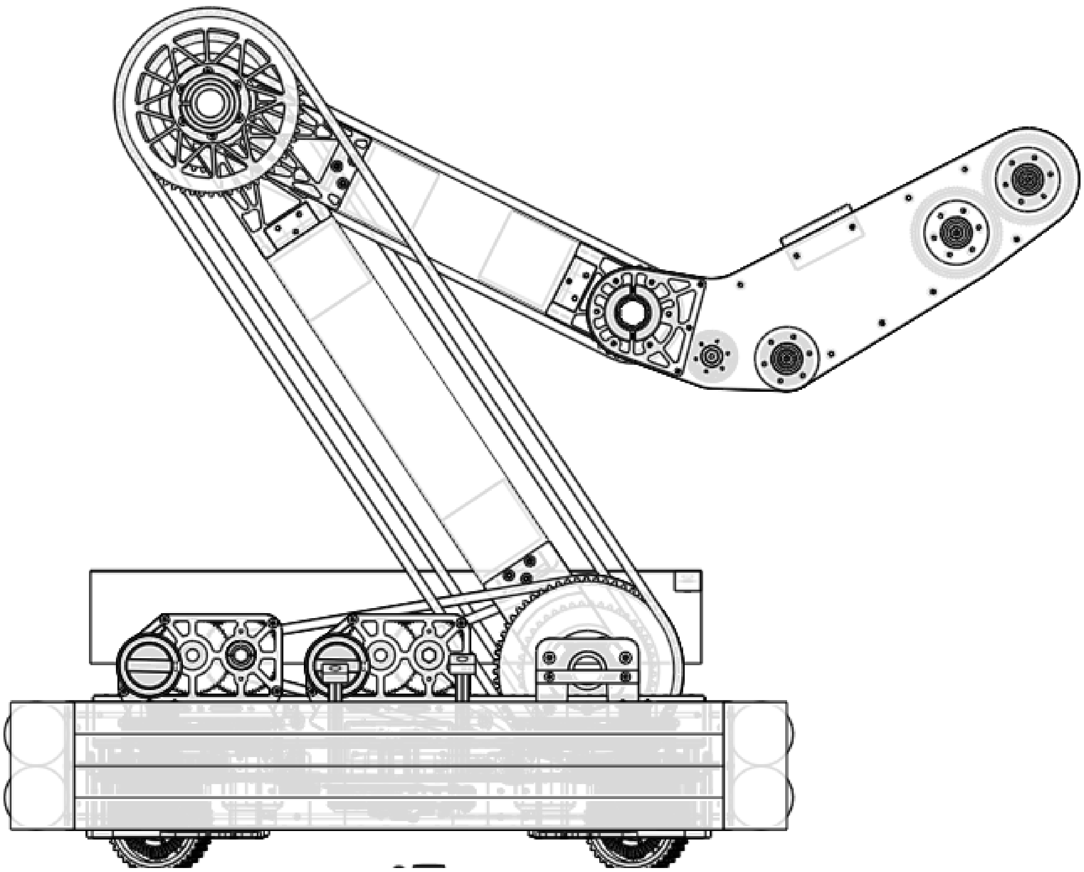





CHARGED UP

PRESENTED BY HAAS Gene Haas Foundation

PROUD MEMBER



FRC WORLD CHAMPIONSHIP EDITION

LUXO

WOLVERINES

1757

2023 TECHNICAL BINDER

FORWARD

Hello, and let us welcome you to FRC Team 1757's 2022-2023 Season. This season has continued the tremendous growth in our robot's design and technical ability that started last year as our team emerged from the hibernation of COVID-19 to become a surprising contender in the New England Region. Continuing to recruit rookie students to supplement our now more veteran team members and Senior Mentors, we have pushed our collective talents to their limits to deliver the competition-worthy robot contained within the pages of this binder.

Our season started in the fall of 2022, introducing a new class of over 10 freshmen, sophomores, and juniors to the world of FRC. We showed off the robot at local town events, built a T-Shirt Cannon to raise school spirit at the prep rally, and hosted weekly technical seminars on everything from the engineering process to CAD, Electronics, Pneumatics, Mechanics, and everything in-between. Over the Summer we got a new OMIO X8 bed router and practiced our CAD and fabrication skills by designing and building an enclosure for the machine. We traveled to Billerica, MA in October to compete in the first-ever New England Robotics Derby. We finished in Second Place, losing in the Finals (The best competitive finish in team history). We piled into our classroom on a cold Saturday morning in January, eagerly anticipating this year's game. 4 CAD models, 8 shared Google Drives, ten weeks, 20 Weekend Build Sessions, 50 Zoom calls, 5799 lines of code, 170 git commits, 19,129 discord messages, and many, many cups of coffee later, we are proud to unveil our robot "LUXO" for the 2023 FRC Season.

Why did we name the robot LUXO? Is it because of the shining lights on its frame that illuminate what game piece we are looking for on the field…no. Is it the bright shining future of the team…no. Is it a reference to solar power and how that ties into the theme of this year's FIRST season…good guess, but no. In truth, we are a bunch of animation nerds, and we thought the robot looked like the lamp in the Pixar Animation title sequence named Luxo. Not every robot name has a deep prophetic meaning…sometimes it's just about the memes.

One very exciting thing about this year is that Team 1757 joined the Open Alliance. We found the Open Alliance teams and their open and timely build season updates so helpful to our team last season that we decided to join so we could help other teams the same way the alliance has already helped us. In addition to frequent updates on our build thread, we also made two appearances on the Open Alliance Show Streamed on twitch. If you want to learn even more about our robot and the design process, beyond what is contained in this manual, please visit our Chief Delphi Build Thread at https://www.chiefdelphi.com/t/frc-1757-wolverines-2022-2023-build-thread/416564

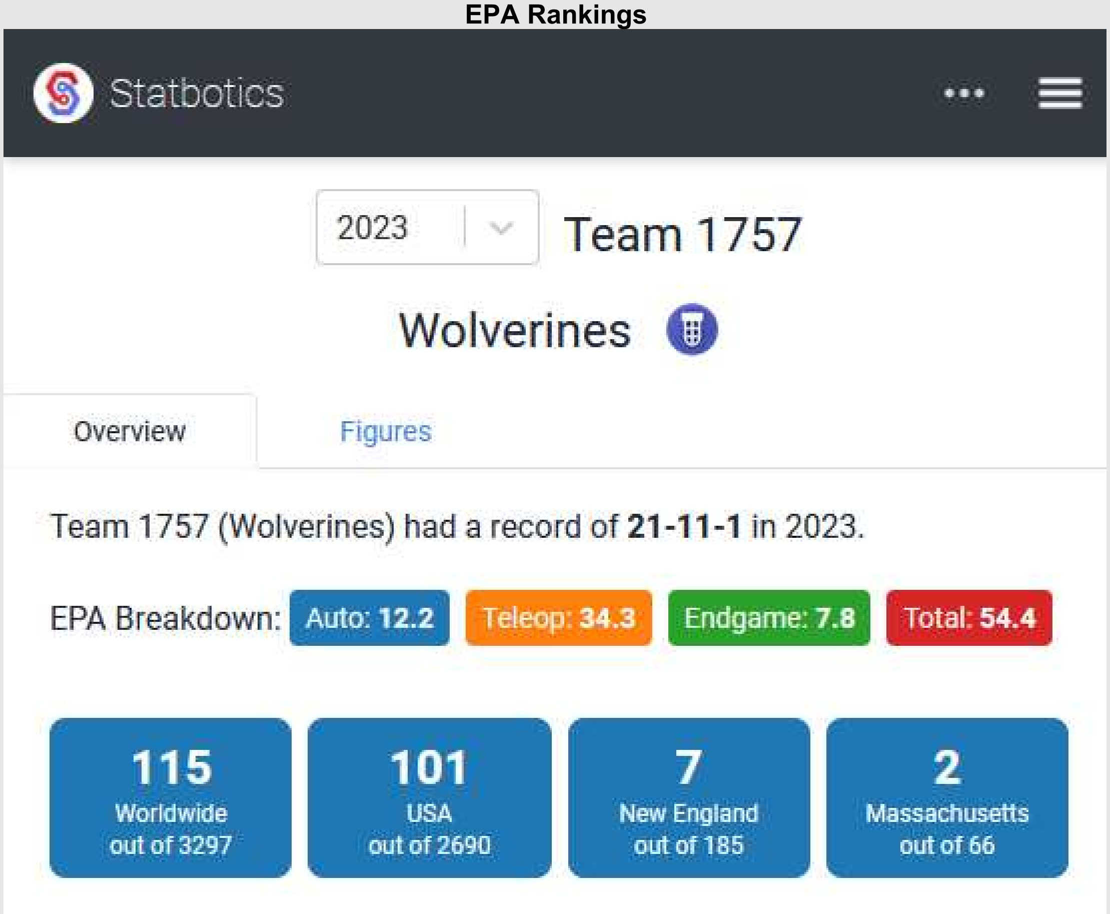
We hope you enjoy this brief look at the design process and technical details that went into this robot, and if you have any questions, look for one of our team members in the stands, in the pits, or on the field. We are always ready to share the knowledge we have gained and share a few hard-learned lessons we learned along the way.

DCMP Update

So it has been a whirlwind of a season so far, after meddeling performance at Greater Boston district we went on win the WPI District Event. Not only were we Alliance captian of the the #2 alliance, we also won the Engineering Inspiration award at WPI. Though out this document you will find various updated information featuring design changes/Repair/modifications that were made during the competition season.

Competitive Record Though District Play:
22-11-1
Greater Boston District - Deans List Semi-Finalist: Sean Tao
WPI District Event – Winners
WPI District Event – Enginnering Inspiration Award Winnerd

EPA Rankings



World Championship Update
We thought our seson couldn't get any better then taking home the teams first ever blue banner at WPI. We were wrong. We came into Wilson Division at New England Championship a solid middle of the pack Contender, however we quicky proved why we were there, our robots consistent and Reliable play led us to take #1 overall at the end of Qualifications, after picking the highest rated offensive bot on the field 176 Aces High, we picked up 1699 Robocats to round out a great alliance. We went undefeated in the Wilson Divison playoffs, taking home another blue banner before taking on the Mier Divison winners for the New England District championship. With the Championship Tied 1-1, we went into a nail-bitting sudden death match where we came out on top.

Please review our OA thread on Chief Delphi for more details.

Competitive Record Though District Championship Play:
39-13-1
Greater Boston District - Deans List Semi-Finalist: Sean Tao
WPI District Event – Winners
WPI District Event – Enginnering Inspiration Award Winnerd
NE Championship – Wilson Divison – Winners
NE Championship – Wilson Division – Excelence in Engineering
New England District Championship - Winners

EPA Record

TABLE OF CONTENTS

GAME ANALYSIS

Every FRC season starts the same way; we gather together as a team, watch the kickoff stream, then hunker down and break down the game in back-to-back 8-hour build sessions. The hope is that by the time we walk out the door on Sunday night, we understand the game and know what we are doing.

After carefully considering the different ways you can score points, we concluded that placing GAME PIECES on the NODES was the most critical ability in this game, with it having the highest potential points available. Without the ability to DOCK and ENGAGE, however, it will be virtually impossible to remain competitive due to the lack of ranking points.

After two days of deliberation, these are the design Requirements we settled on.

DRIVE

- Need to be a Small Bot – The smaller the bot, the easier it is for 3 robots to balance on CHARGE STATION
- Need a low center of gravity
- Need to be able to drive and balance on the CHARGE STATION.
- Preferably autonomous balancing on CHARGE STATION
- Use of vison (April Tags) to provide feedback to the onboard odometry system
- Use of vision to identify and seek out game pieces on the field.

ARM

- Arm needs to be strong and durable
- Use Encoders on the input and output of gearboxes to monitor and minimize backlash.
- Either 2 or 3 Degrees of Freedom Further testing will be needed.
- Needs to score at all 3 levels BOTTOM, MIDDLE and TOP Nodes.

INTAKE

- Quickly acquire GAME PIECES (Touch It – Own It)
- MUST pick up CONES and CUBES from the LOADING STATION
- MUST pick up CUBES and upright CONES from the ground.
- Would like to be able to pick tipped-over CONES from the ground.

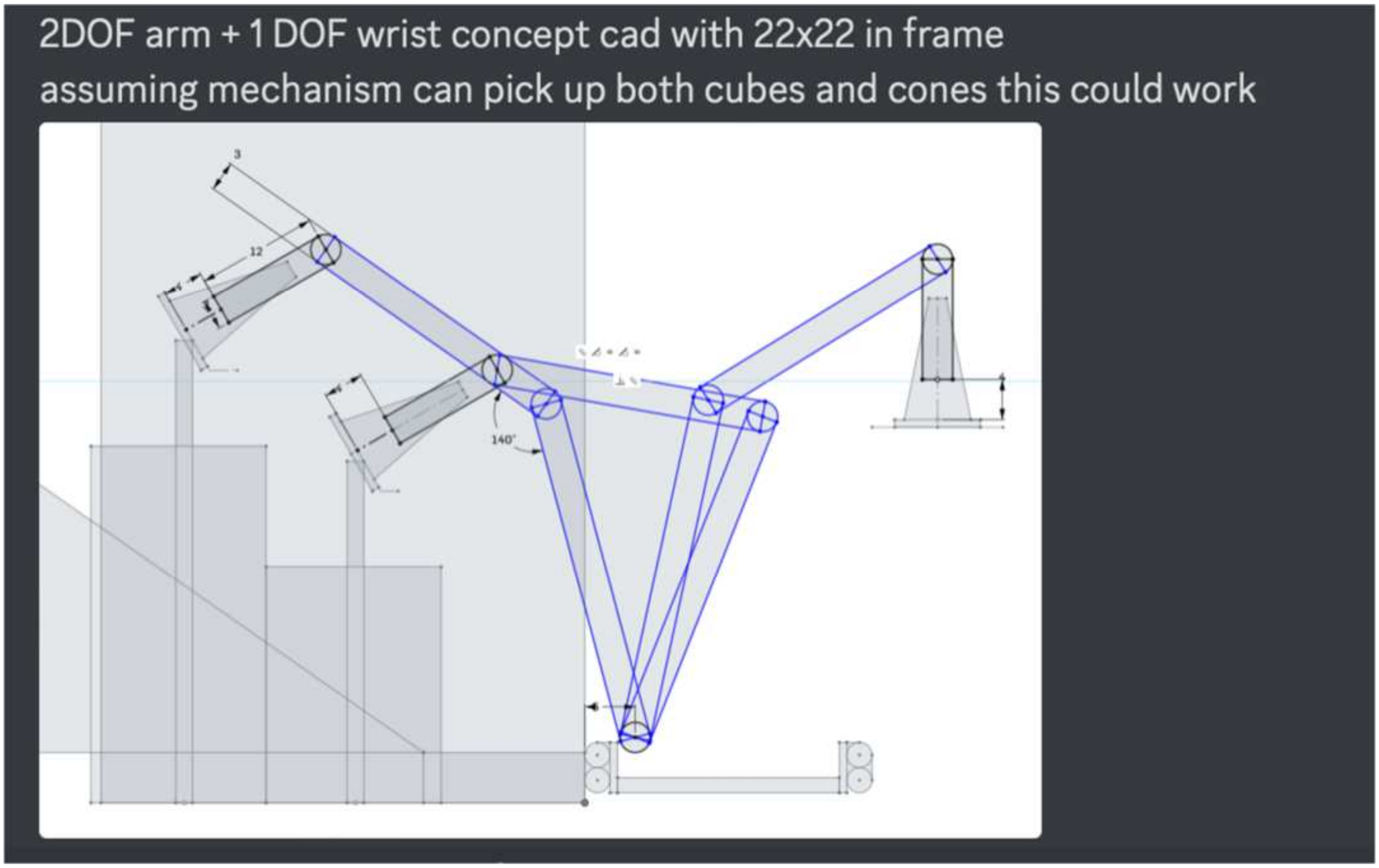
GENERAL DESIGN CONCLUSIONS

Our overall goal for the season was to be a competitive bot in district-level play and qualify for New England Championship. To accomplish this, we need to, at the bare minimum, make it to Elimination at both our district events, hopefully as an Alliance captain or 1st pick.

We approached our design as trying to build a highly reliable jack-of-all-trades bot, focusing on gaining one of the two performance-based ranking points in either match.

Inspired by the cost-effective production strategies of the Hass Formula 1 racing team and our limited team members and design resources, we prefer to use pre-engineered solutions wherever possible to focus our design resources on critical complex components.

IDENTIFYING DESIGN CONSTRAINTS

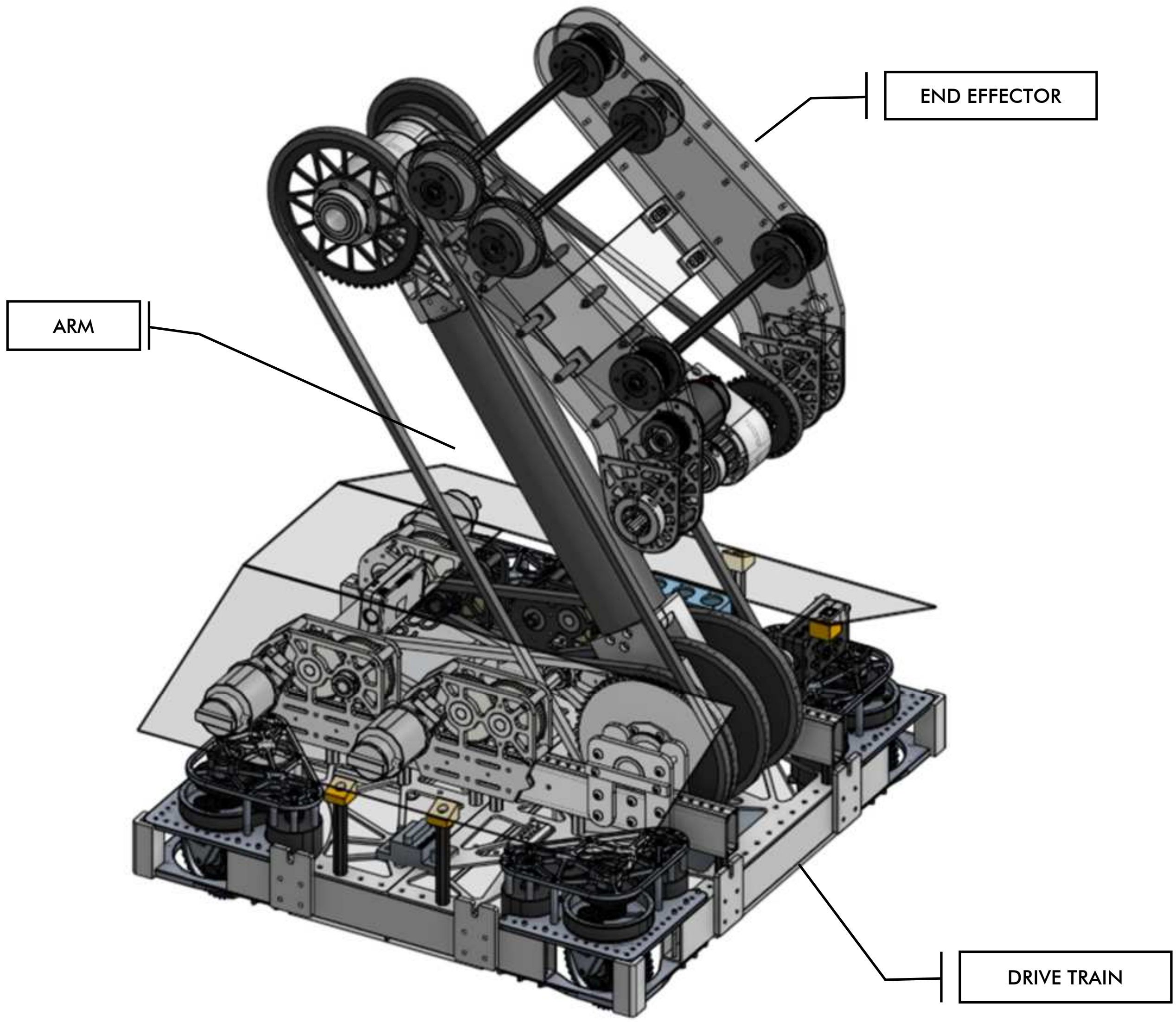


We are thinking about using an arm as a manipulation mechanism. We potentially envision a 2DOF arm + 1 DOF wrist that can pick up both cubes and cones, with a high range of motion on the wrist joint. As we can utilize the bot's movement, we do not need the Arm to move from side to side. An important note is that with an arm the starting configuration poses a good challenge, as it will need to fit inside of the robot's frame before activation. We have found that the shoulder joint only needs to move 90 degrees max, the elbow joint 210 degrees, and the wrist joint somewhere like 270 (at least in the configuration, lots to play with) to achieve all necessary motion.

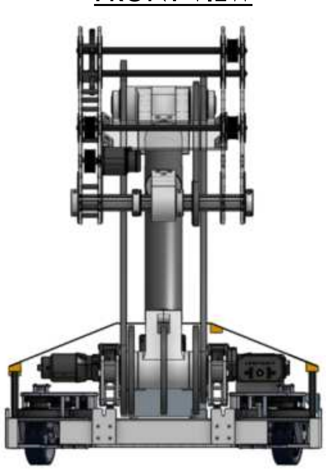
THE 1757 RAPID DEVELOPMENT MODEL

- Clearly Identify the design requirements of the system
- Design and Build a prototype that can be used to test design assumptions and Test
- Use what we learned from testing to develop a final design
- Fabricate final version and intergate into overall robot systems

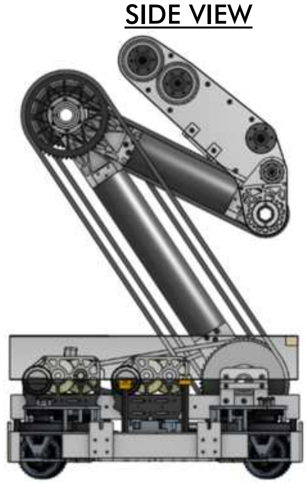
Final Robot Design



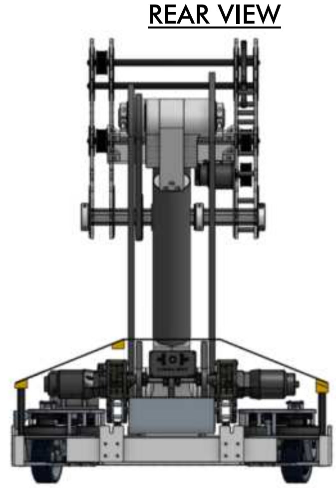
FRONT VIEW

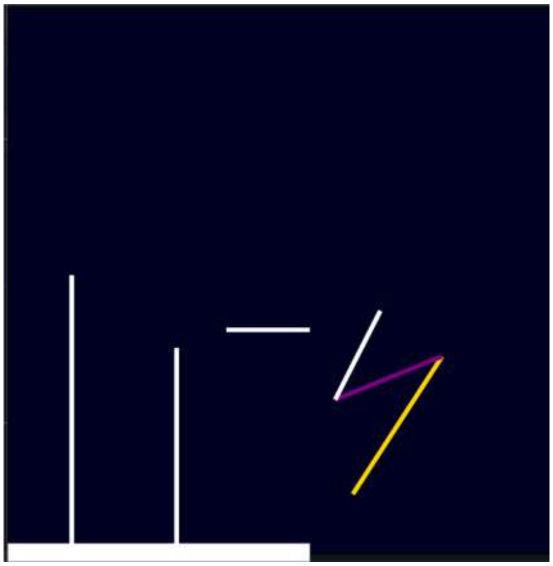


SIDE VIEW

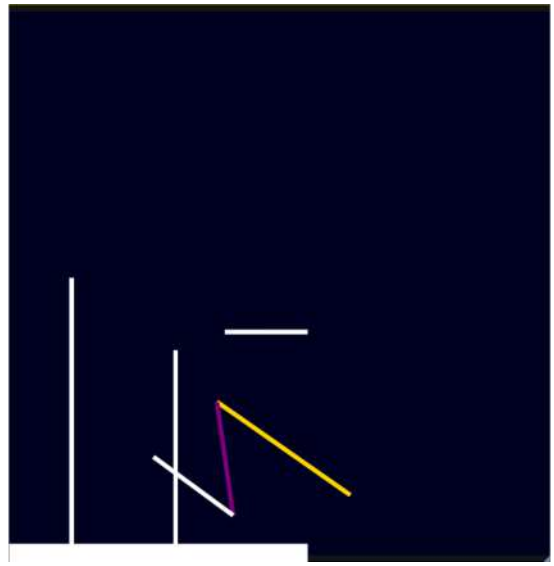


REAR VIEW

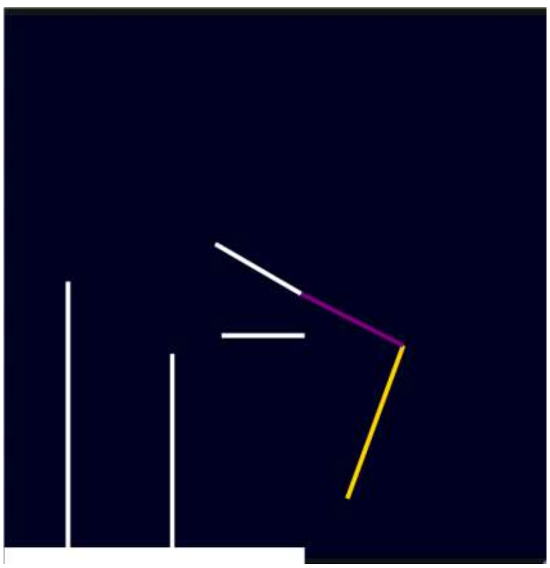




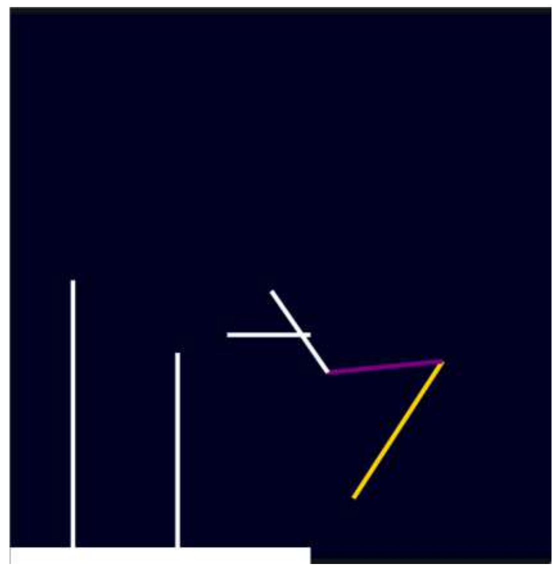
Default Configuration



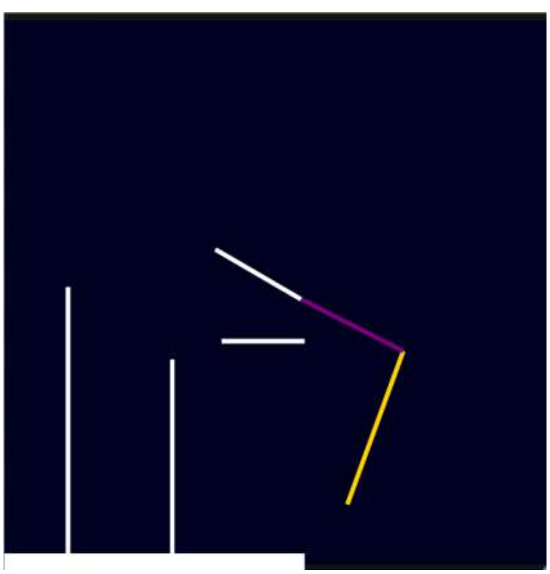
Floor Pickup



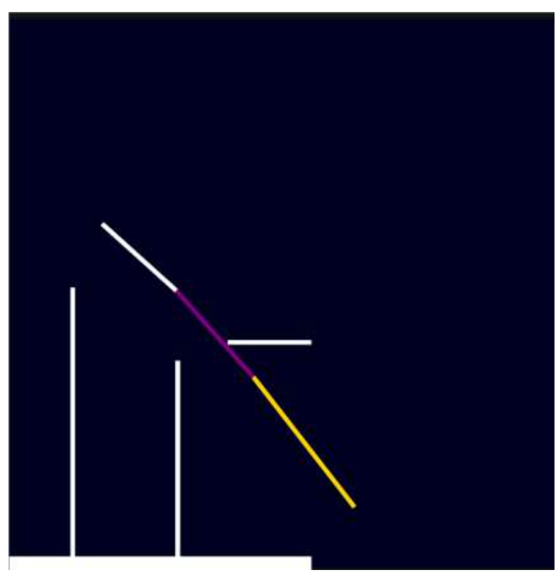
Substation Pickup



Score – Bottom



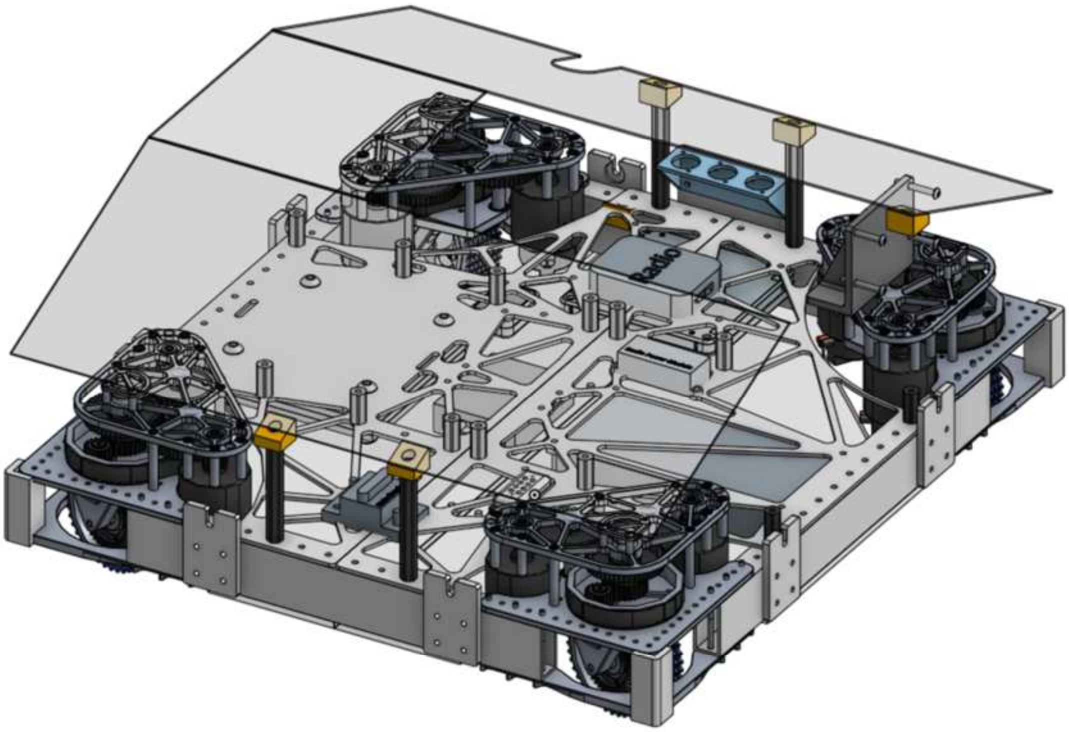
Score – Middle

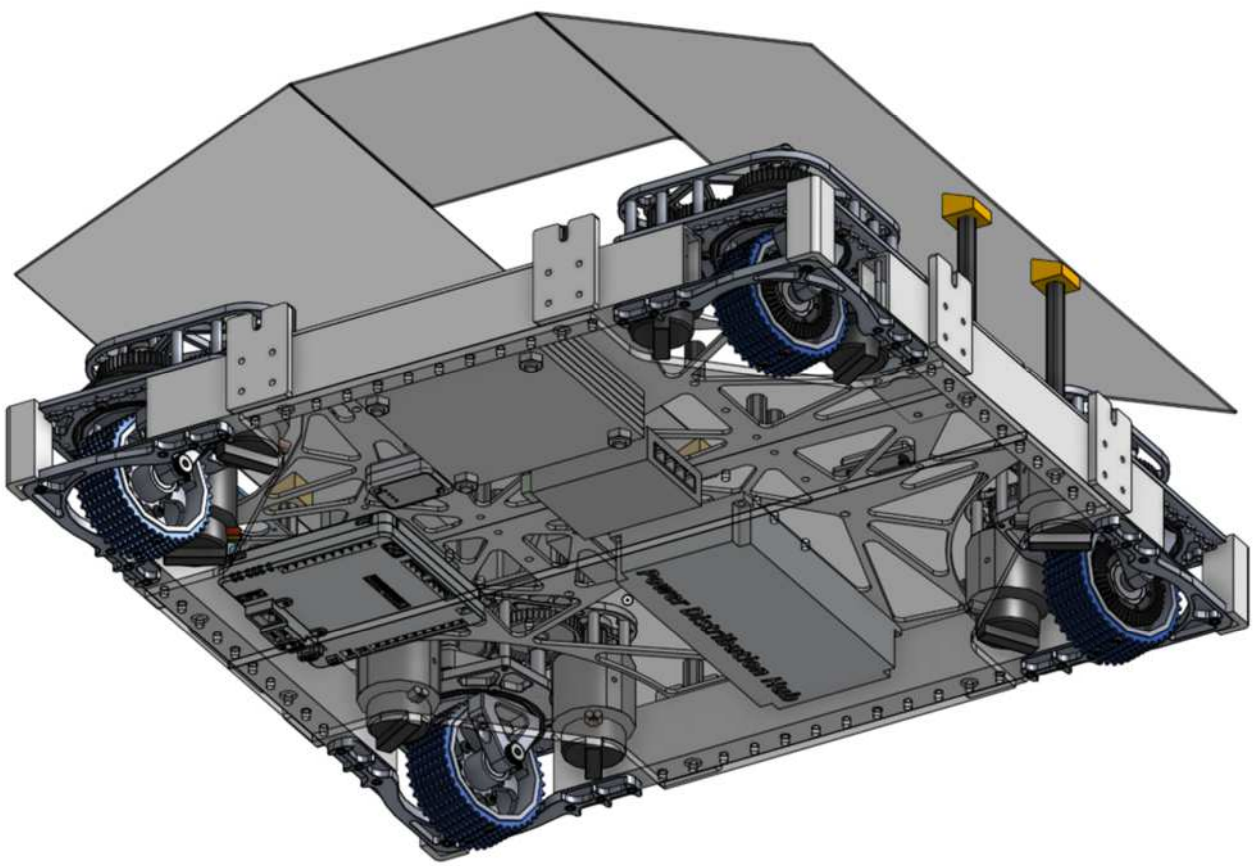


Score - Top

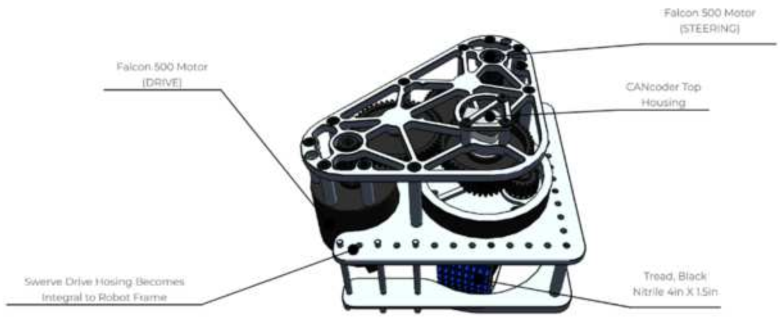
Above – 3D Simulation of Arm Joints in all of its various Arm Configurations

MAJOR SYSTEM #1: DRIVE TRAIN





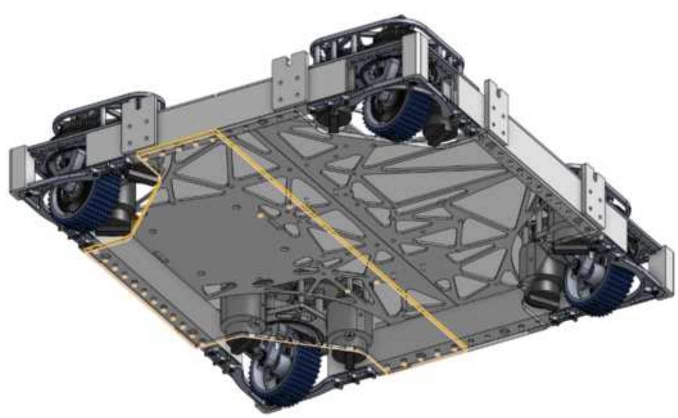
1.1 - SWERVE DRIVE MODULES



Last season was our first competitive season using swerve drive and we could not be happier with the results, because of the equal power between steering and driving there are none of the performance trade offs inherent in other drive systems. We can still run circles around the field when we need to and we can still push another robot across the field when they are in our way. One of our favorite features exclusive to swerve drive is what we call the park feature, by turning all 4 wheels to a 45° angle relative to the corners of the robot the robot effectively parks itself in place and wont move, another robot can push against us all match long and we wont move. Last year we used Swerve Drive Specialties Mk4 units, and this season we upgraded to the newly released Mk4i units. This revised design points the motors downward into the bot instead of mounting above the module. This allowed us to eliminate ≈2" of vertical space in our robot between the drive frame and the major systems.

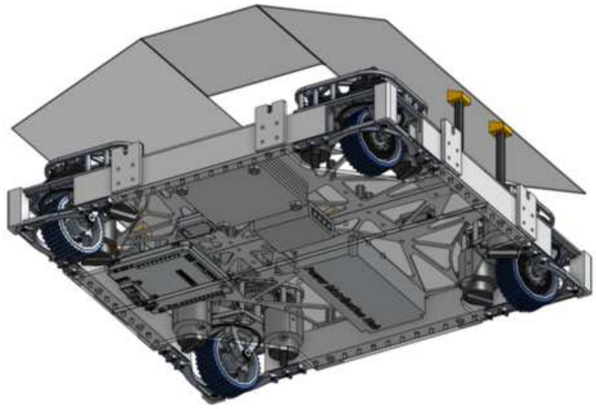
SDS MK4i Swerve Modules	
Powerplant	Falcon 500
Gearbox Configuration	L2
Overall Gearbox Ratio	6.75 : 1
Unadjusted Free Speed	16.3 ft/sec

1.2 - ROBOT FRAME



Left – CAD – Isometric Top View of robot frame
Right – CAD – Isometric Bottom View of Robot frame

1.3 - ELECTRONICS SUBSYSTEM



One thing we struggled with on our 2022 robot was how inaccessible most electrical components were. On our 2022 robot, all the electronics sat in a belly pan at the base of the bot, and the only way to access most of the components required you to remove the majority of the Robot Systems. Inspired by another team's design from 2022, we decided to hang all of our electronic components upside down and face the ground. Now to access the electronics, we tip the bot on its side, remove the ¼" protective polycarbonate plate, and you have full easy access to all the electronics components. (Credit to Team 125 for the Idea, they have the thanks of a grateful drive team and Pit Crew)

ELECTRONICS SYSTEM MAJOR COMPONENTS

- (1 ea) National Instruments - RoboRio 2
- (1 ea) REV Robotics - Power Distribution Hub
- (1 ea) Navex 2 – RoboRio MXP expansion Board
- (1 ea) CTRE CANivore
- (1 ea) CTRE CANdel
- (1 ea) BrainBoxes – SW-015 5 Port Gigabit Switch
- (1 ea) Generic Passive POE Injector
- (1 ea) Limelight 2 Camera
- (12 ea) Falcon 500 Motors
- (1 ea) REV Robotics Sparkmax brushless motor controller
- (1 ea) REV Robotics NEO 550 Brushless DC Motor
- (1 ea) Open Mesh Access Point [Insert Model Number]

1.4 - TESTING PORTS

We added a convenient patch panel to the upper side of the robot to allow for quick access to essential data ports when we don't want to access the underslung electronics.

PATCH PANEL SLOT 1 – USB TYPE A

This slot connects to one of the USB Type A ports on the RoboRio. This typically has a USB flash drive plugged in. During a match all the system logs are copied to the USB drive. After a Match, the USB drive can be pulled and opened up in AdvantageScope on the debug machine for post-game analysis. It's our version of a Blackbox on an airplane.

PATCH PANEL SLOT 2 – USB TYPE B

This connects to the USB Type B Port on the RoboRio—a redundant method for tethering the robot for control and debugging at events.

PATCH PANEL SLOT 3 – RJ45 CONNECTOR

This connects to the Ethernet Switch Via CAT5e for network access. Used for tethered connections to the bot during testing. Ethernet tethering is preferred, but we have encountered software reliability issues in the past.

DCMP Update

At the Revere District event we ran into serious problems tethering to the robot via ethernet and via USB B. we traced the ethernet tethering problem to a problem with the network configuration issue on the driver station laptop. We were unable to determine a definite cause of the USB-B connection issue, but, we think it most likely to to poor quality of the 90° usb connector used on the robot. From that point on we connect a USB-B cable directly into the port on thr

At the same time we realized we needed a button to manually put the arm motors in coast mode for serviceability when the bot is not connected to the driver station. Since we are no longer using the USB-B testing port we replaced it with a momentary push button switch.

1.5 – CAMERA/VISON SYSTEMS

Limelight 2 – Camera

We are utilizing a Limelight 2 Camera for a variety of tasks on the robot mostly devoted to sensor fusion and automation of systems using computer vision. The Limelight's field of vision (FOV) is essentially parallel to the floor and at the height of the April Tags.

Please refer to the Software section of this document for more information on how we use the limelight and April Tags to improve the Onboard odometry of the robot.

1.6 – COUNTERWEIGHT

Not originally intended as part of the design, upon testing of the robot with the Arm fully extended in the scoring position, we realized that robot was prone to falling forward. To resolve this, we looked to add ballast to the bot. First we thought of lead but didn't want to deal with the potential health risks of improperly encapsulated lead. We investigated tungsten; however, a review of the current price of tungsten plate (≈$40/kg) quickly ruled it out as a potential candidate. We settled on 6" x6" x1/4" steel plates mounted directly under the robot battery. After testing with different #'s of the plate, we decided on 6 Plates with a total weight of ≈25 lbs. Now the bot is highly stable even when the Arm is fully extended.

1.7 – PROTECTIVE COVERS

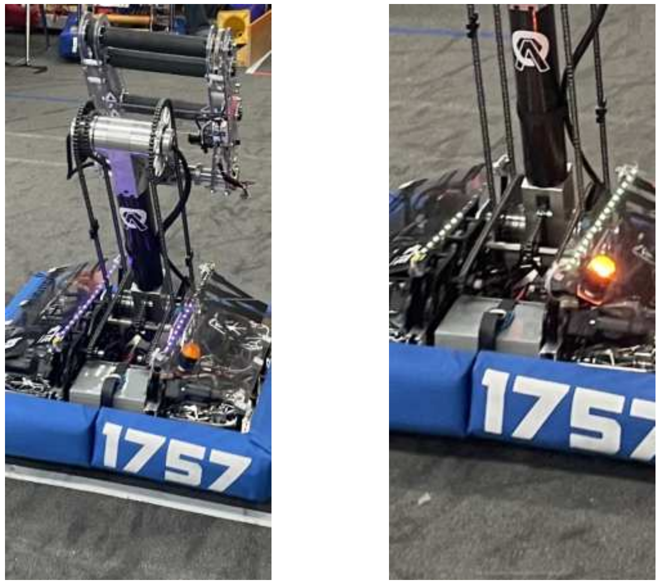
We added protective covers that Slope away from the central arm structure down to the bumpers. Not only do these plates provide a valuable location to display all of our great sponsors they serve to prevent errant game pieces from getting stuck inside the robot during a match.

DCMP Update

Originally the protective covers were only hled on with 3M™ Dual Lock™ SJ3560, this material is nice because it very strong but eaisily removable. Durring qualifying matches in Revere however, these panels kept falling off and dragging around the fieild. The Dual Lock strips were reinforced with zip ties and these held through all of playoffs in revere, and all of qualifications at WPI. Then in Playoff Matches we shered off 3 of the metal standoffs holding up the protective covers. We made quick repairs to keep going however prior to DCMP we will be swaping out all the ½ thunderhex standoffs with 1” 80/20 extrusion with hardened bolts for strength.

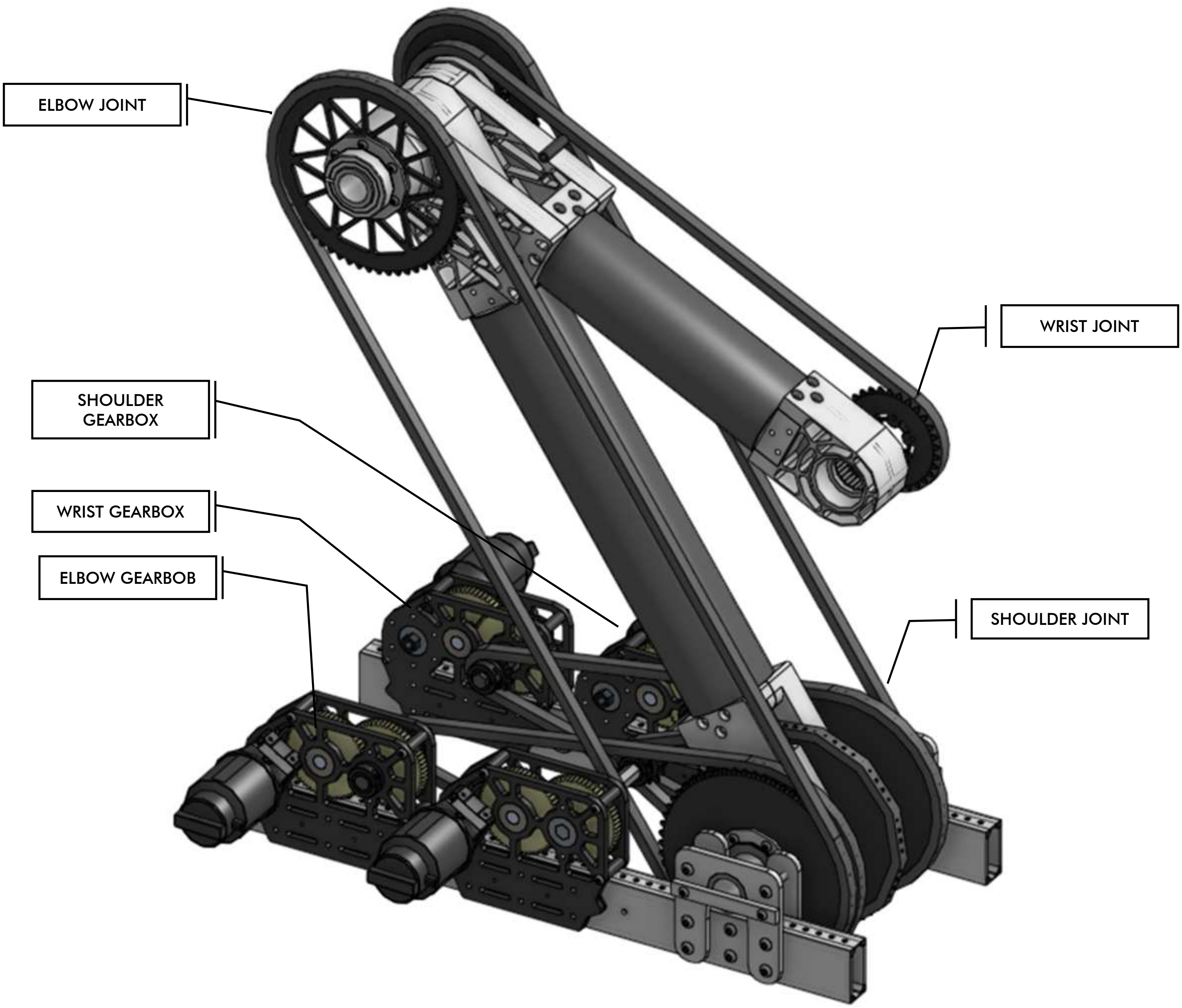
1.8 – GAMEPIECE INDICATORS

One of the hardest things to do in a match is how to signal between the driver station and the human player what game piece you want them to load into the robot. People use Hand Signals, Colored pieces of paper or guess. We wanted to take the guesswork out of the equation, so we mounted 2 LED Strip lights along the top of the protective covers. The driver controls what color these strips are so he can communicate to the human player which game piece to feed to the bot – Yellow for Cones and Purple for Cubes.



Left – Robot Displaying "I Want a Cone"
Right – Robot displaying "I Want a Cube"

MAJOR SYSTEM #2: ARM



2.1- MOTORS & GEARBOXES

To keep the robot's center of gravity low and keep the Arm as simple as possible we decided to locate all of the heavy motors and gearboxes at the base of the superstructure.

GEARBOX # 1 – SHOULDER GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into Custom Gearbox based on WCP Gearbox design.

GEARBOX # 2 – ELBOW GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into COTS WCP Single Speed Gearbox.

GEARBOX # 2 – WRIST GEARBOX

Falcon 500 Brushless Motor
Single Stage Versaplanetray into COTS WCP Single Speed Gearbox.

2.2 - CHAIN DRIVE

Using a combination of dead and live axels we transfer the power of the gearboxes up though the Arm to power each of the individual joints. For Reliability and durability, we chose to use #35 roller chain rated for 11,000 lbs of force.

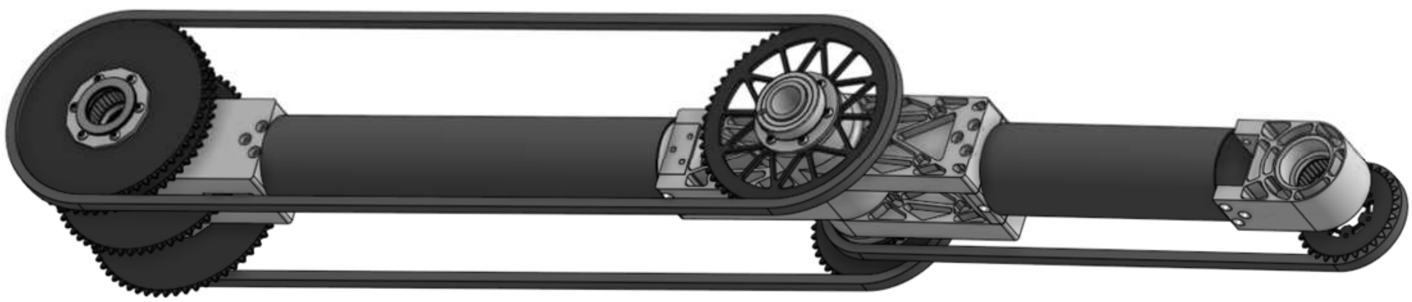
Below is a summary of the different chain runs on the Arm

CHAIN DRIVE 1 – SHOULDER
Shoulder Gearbox Output 12t Sprocket → 60t sprocket on Shoulder (Dead Axel)

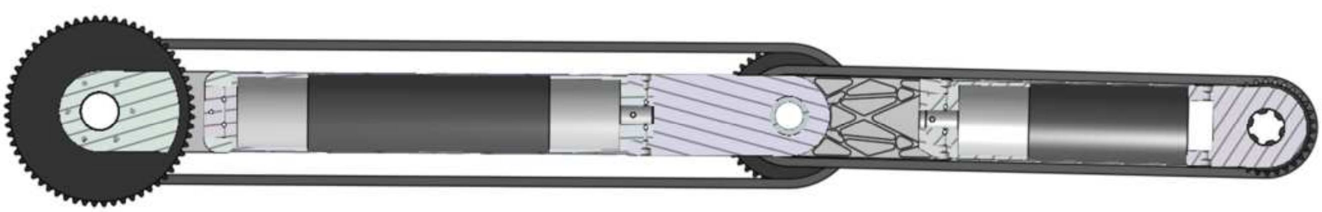
CHAIN DRIVE 2 – ELBOW
Elbow Gearbox Output Shaft 12t Sprocket → 60t Sprocket on Shoulder (Live Axel) → 70t Sprocket on Elbow (Dead Axel)

CHAIN DRIVE 3 – WRIST
Wrist Gearbox Output Shaft 12t Sprocket → 60t Sprocket on Shoulder (Live Axel) → 60t Sprocket on Elbow (Live Axel) → 40t Sprocket on Elbow (Live Axel) → 32t Sprocket on the Wrist (Dead Axel)

2.3 - ARM STRUCTURE



Above - 3-D View of Outstretched Arm



Above - Section View Though Center of Carbon Fiber Arm

CARBON FIBER ARMS

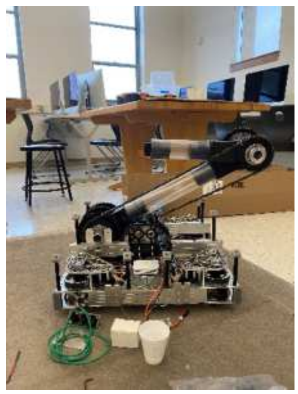
We chose to use carbon fiber tubes as the main structure of the Arm due to its strength and lightweight, the more weight we could save on the Arm the lower we could push the robot Center of Gravity. Carbon Fiber tubes are a stock McMaster item 3" Ø. Carbon Fiber is Epoxy bonded to 3" hollow aluminum plugs bolted to the aluminum joints.

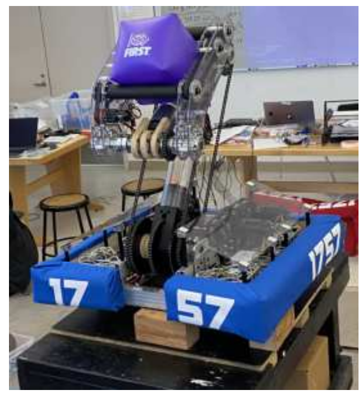
Left – Final carbon fiber arm links after final glue up

Right – Final aluminum plugs used in the ends of the carbon fiber tubes

3D Printed and Polycarbonate Prototypes

Because we knew the carbon fiber and machined aluminum would take time and money to manufacture, we heavily used 3D-printing to make prototypes of the Arm and test and confirm critical geometry before placing final fabrication orders. These prototypes are two fragile to be used on a competition bot but worked well for their intended purposes. We learned very Important lessons about where the concentrations of forces were along the axels and what parts needed reinforcement.





Left – 3D printed Prototype of the wrist joint, printed on a FormLabs 2 SLA Printer

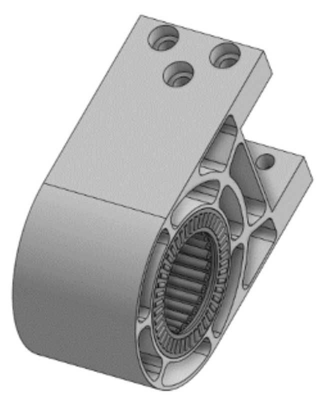
Center – Polycabonate Prototype arm Mounted on bot for the First Time

Right – Fully Assembled "Alpha" Robot build

2.4 – JOINT STRUCTURE

CAD - Shoulder Joint

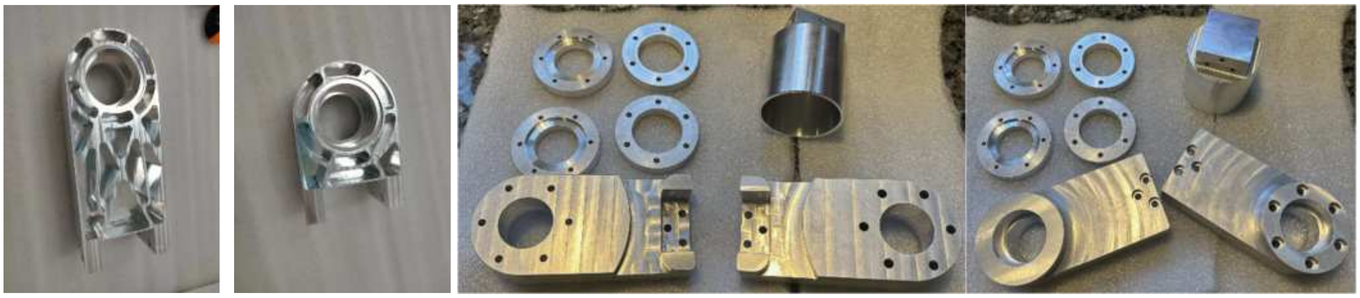
CAD - Elbow Joint



CAD - Wrist Joint

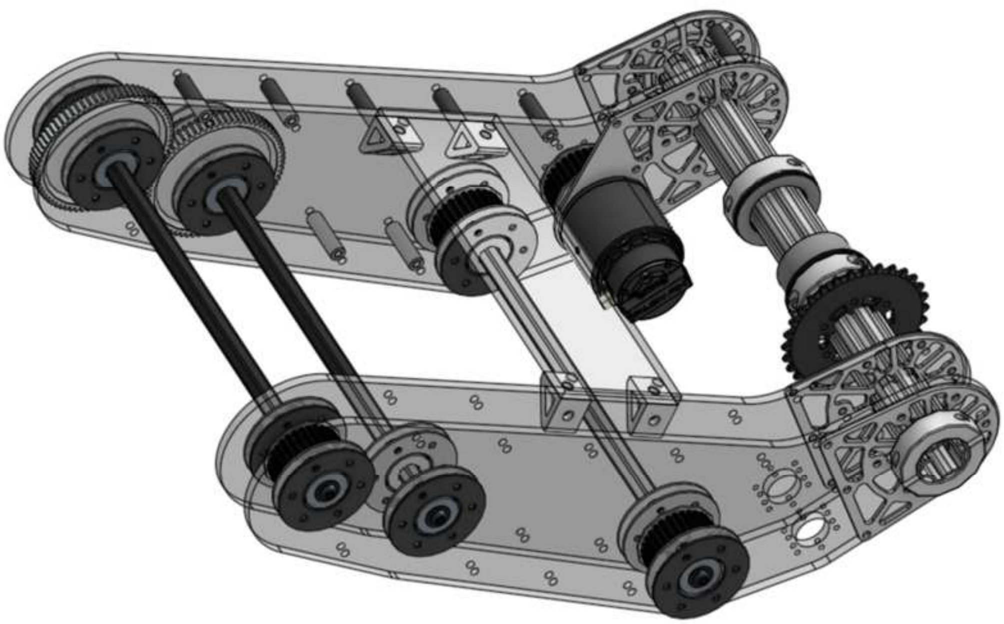
Needle and Thrust Bearings

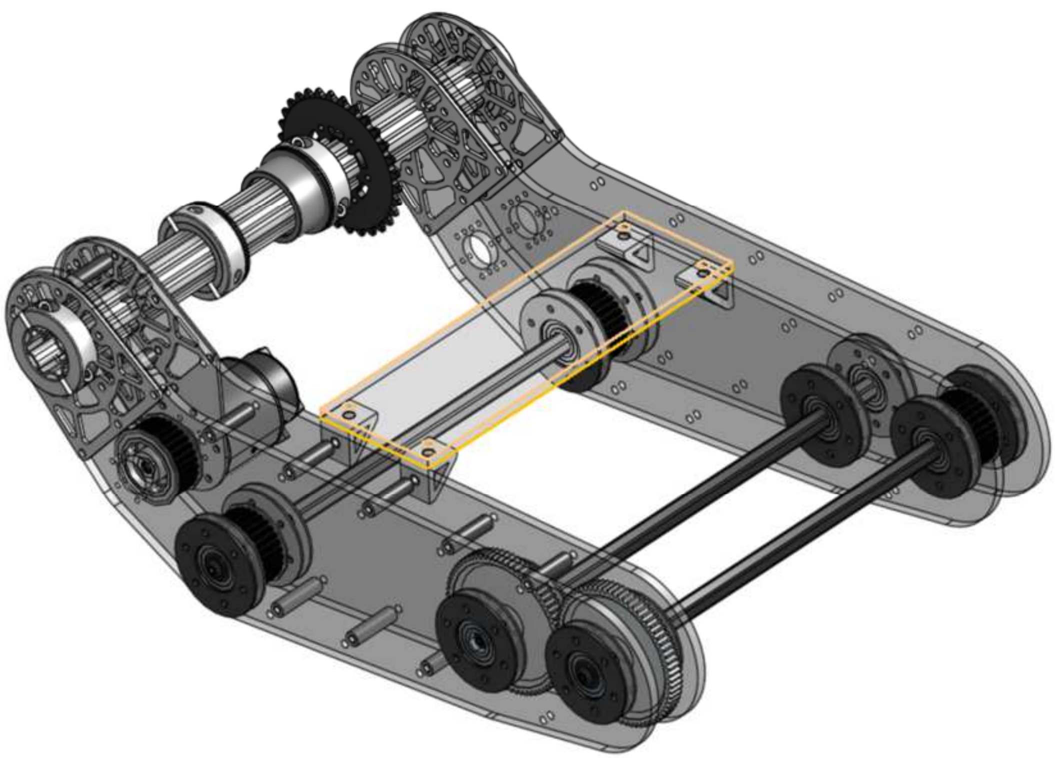
Used in All three joints to allow for smooth rotary motion in each joint.



Photos - Final Machined Parts

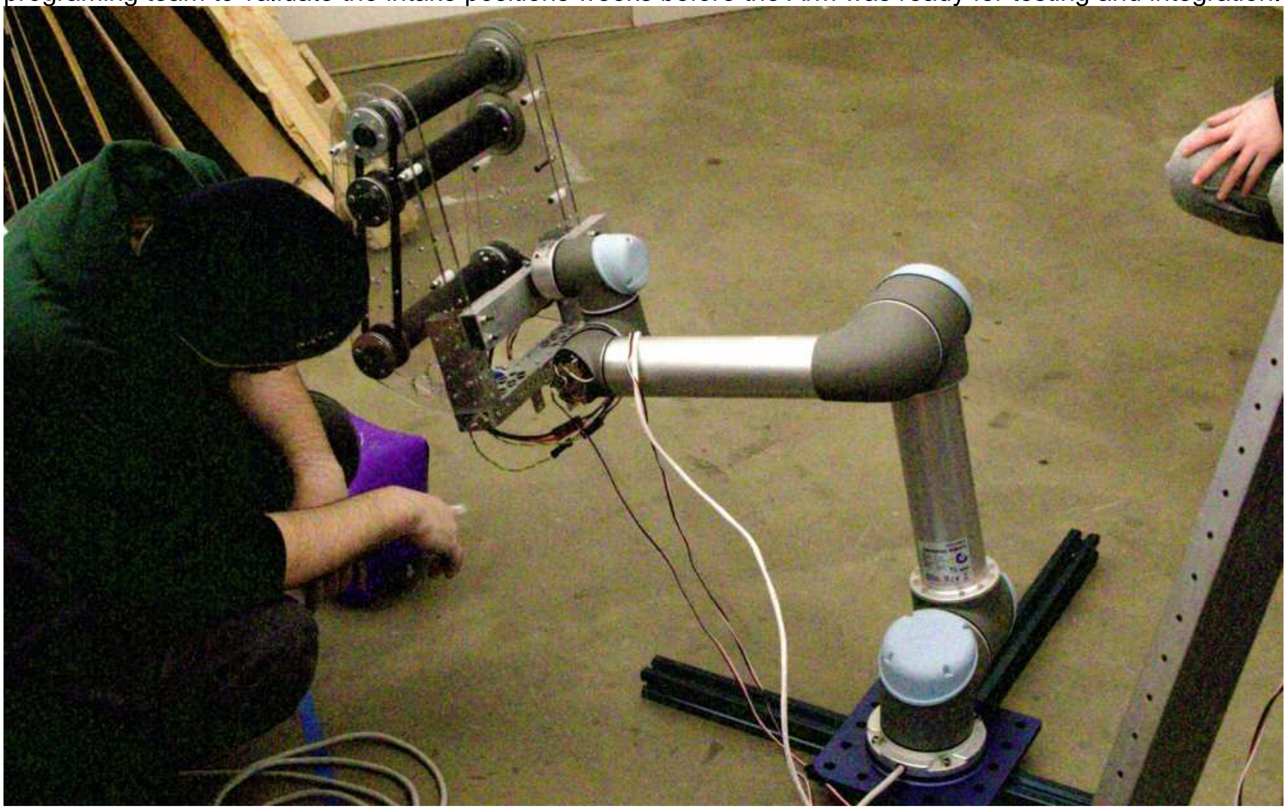
MAJOR SYSTEM # 3: END EFFECTOR





INTAKE PROTOTYPING – FUN WITH INDUSTRIAL ROBOTS

We had a lot of fun prototyping this mechanism, since it was the first major mechanism that we finished we had lots of time to put it trough its paces. One of our mentors has access to a Universal Robots UR3 robot and brought into our lab during one of our weekend build sessions (see images below). This actually proved to be essential because it allowed our programing team to validate the intake positions weeks before the Arm was ready for testing and integration.



3.1 – ROLLERS

The rollers we are using are Vex VersaHub Rollers with ¼" neoprene tubing stretched to cover them, they are very grippy and hold the cube verry securely. We started with the dimensions of the everybody roller for our prototype then made some modifications before settling on final separation differences. The neoprene tubing is undersized for the OD of the polycarbonate roller. We learned a fun trick to clamp off one end of the neoprene tube and inflate it with an air compressor to stretch it over the polycarbonate tube, when the air is released, it makes a perfect friction fit between the Neoprene and the polycarbonate. We have had no detectable slippage after weeks of testing with the rollers.

DCMP Update

After 33 competitive matches one thing is clear, we have problems picking CUBES up off the floor and in order to maintain our competitive edge at DCMP we know we need to be able to get CUBES up off the floor. We thing the majority of the problem is related to how narroe our end effector is, the original design was only 1” wider than the width of the CUBE. To aliviate this issue we are planning to widen the end effector by 3”.

3.2 – MOTORS & GEARBOXES

The intake is powered by a REV Robotics NEO550 Brushless motor into a REV Robotics Ultraplanetary gearbox. The motors small size is nice however because we mounted the Sparkmax Motor controller on the intake as well there is no significant weight savings compared to using a Falcon 500 with an integrated Talon SRX. We may end up swaping this out for simplicity sake in the future.

REV Ultraplanetrary	
Powerplant	NEO550
Gearbox Configuration	4:1, 5:1
Overall Gearbox Ratio	20:1

SOFTWARE

SOFTWARE: OUR DEVELOPMENT ENVIRONMENT

WPILib

The perineal stalwart, we still rely on core elements of WPIlib for robot communications and debugging. WPILib's new Logging features have greatly enhanced our Debugging capabilities

RobotPy

We have found that students have a lot easier time learning python then they do Java or C++ so with the growing support for RobotPy we migrated our Codebase from Java to Python in 2020. As of this March we are an official contributor to the RobotPy project

GitHub

Without Github our level of remote work and collaboration just wouldn't be possible.

SOFTWARE: NEW AND UPDATED TOOLS THIS YEAR

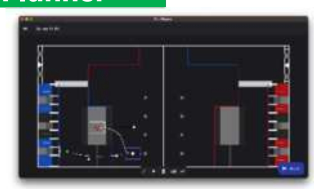
AdvantageScope

WE LOVE ADVANTAGE SCOPE! Not only does it log *everything* but does it in away that is intuitive and easy to review. No more searching though 10000 lines of log files to find the one piece of information we need. Huge thanks to team 6328 for building such a great tool.

PhotonVision

We are using PhotoVision as our native development framework for Computer vision due to its growing wide support inside the FRC community. It does not include native support for RobotPy however so as an offseason project our lead programmer wrote a custom wrapper for PhotonVivion so it can work inside our RobotPy environment

PathPlanner



Last year we used PathWeaver, but we were disappointed in the lack of native support and increased complexity in the development stack so starting with the off season we transitioned all of our Autonomous path planning to PathPlanner. We had much fewer issues with this system.

SOFTWARE: DRIVE

Taking off of last year, the drivetrain codebase has stayed the same. We are running field oriented drive with robot relative rotation to allow for quick maneuverability. A button to align to the nearest 90 degree angle was added to help with driver alignment. This state slightly reduces the speed and snaps the angle of the robot in order to have perfect alignment to the double substation, single substation, and grid every time. For our automated balance sequence, we work in robot relative space on the robot relative gyro.

A Brief Tangent - Absolute Relative Drive

Last year our lead programmer had a new idea for drive control, an absolute relative drive. The common swerve drive control method was to have a field relative translation for the bot, and a robot relative rotation. What this meant is a left input on the rotation axis would result in the robot rotating to the left at a constant speed. A translation action was not affected by rotation but instead was in "field relative" space. The difference of absolute drive is that the rotation is also field relative. A left input on the rotation stick will yield the robot turning to face left. This year we expect this type of robot control to be very important for drivers when they have to be able to turn to specific positions for collection and scoring on swerve drives. You can see this in action in any one of our videos from last year. Having fixed controlled rotation will allow for precise driver input and less fiddling with controls when cycle time is very important.

The drivers have also experimented with alternate driving methods on swerve to get used to interesting control schemes such as a curvature drive, standard tank drive, standard field relative drive, and full robot relative drive.

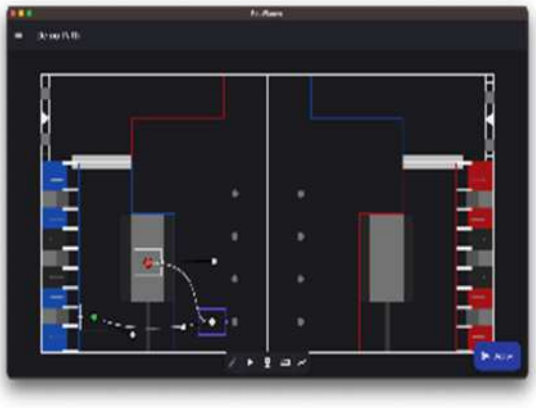
SOFTWARE: INTAKE

The intake is using a state machine in order to regulate its expected behavior. There are 2 enumerable values: one for the gamepiece intended direction (intake, outtake, hold) and one for the desired gamepiece type (cube, cone). The transition between each state is dictated by a user input to any given category. If no input is given, the system holds its position and keeps the desired gamepiece remembered. The state value of the desired gamepiece is displayed to the driver and to the human player through pulsing leds of the respective color.

SOFTWARE: ARM

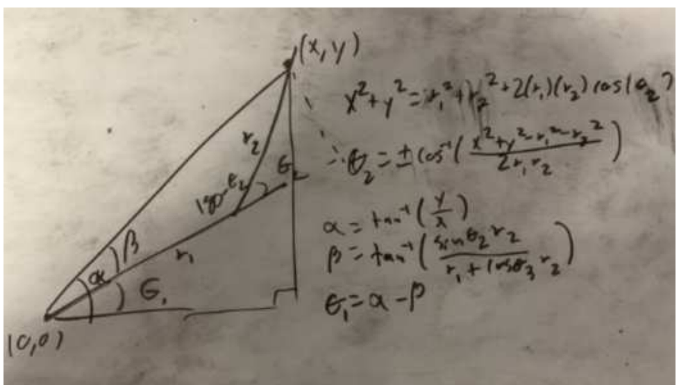
A triple jointed arm is no easy feat in order to program smoothly. From cad, states are given about the end effector's desired position and rotation relative to the floor. From there we use inverse kinematics to determine each per-joint relative rotation at any given position. A cartesian control on the wrist joint's position is added using a trapezoidal PID profile to lay out a path for the Arm to follow. For the wrist it has another trapezoidal PID profile controller. When going state per state on the Arm's motion, we check for if the relative angle goes over software end limits in order to prevent running the Arm into itself. These are done in joint relative space. Since the Arm is controlled from the base through chain and sprockets there is a virtual 4 bar created in which the rotation of any given joint is given relative to the ground. These are converted into motor space and passed onto each motor where they have a position PID controller onboard. For simulating the system we are using mechanism2d to view the expected values of the Arm and be able to run through positions. This simulation first approach has allowed minimal revision and a solid foundational codebase that is mostly complete before the bot is finished. Furthermore the position of the Arm is logged in 3d with advantagescope based on the position it believes the bot is in. Logging was also a priority for a complex system and therefore we log the instructed position and actual position per joint and each end position in cartesian space.

SOFTWARE: AUTONOMUS



We experimented in the offseason with pathplanner and use it extensively for our autonomous. Each necessery command is placed into a potential pool of events for pathplanner to fire. At the beginning a sequence determined solely in pathplanner is fired. Going off of last year we had a large time gap in order to make slight changes so instead for this year we are using the waypoint system and opting to have any given autonomous contained solely in pathplanner. This has increased our ability to construct autos and tweak any given aspect as needed. For the library itself of parsing, the lack of first class robotpy support meant we had the freedom to reimplement autonomous however we pleased based on the path. We follow a changing trajectory and the swerve drive using onboard odometry and a weighted vision estimate determines its bot position relative to the global field and follow through it between each section.

SOFTWARE: SIMULATION



armsubsystem.py

```
    def setEndEffectorPosition(self, pose: Pose2d):

        twoLinkPosition = Translation2d(
            pose.X() - constants.kArmwristLength * pose.rotation().cos(),
            pose.Y() - constants.kArmwristLength * pose.rotation().sin(),
        )

        endAngle = math.acos(
            twoLinkPosition.X() * twoLinkPosition.X()
            + twoLinkPosition.Y() * twoLinkPosition.Y()
            - constants.kArmtopLength * constants.kArmtopLength
            - constants.kArmbottomLength
            * constants.kArmbottomLength
            / (2 * constants.kArmtopLength * constants.kArmbottomLength)
        )

        startAngle = math.atan2(twoLinkPosition.Y(), twoLinkPosition.X()) -
math.atan2(
            math.sin(endAngle) * constants.kArmtopLength,
            constants.kArmbottomLength + math.cos(endAngle) *
constants.kArmtopLength,
        )
        wristAngle = pose.rotation().radians() - startAngle - endAngle

        bottomArmEncoderPulses = (
            startAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kBottomArmGearRatio
        )
        topArmEncoderPulses = (
            endAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kTopArmGearRatio
        )
        wristArmEncoderPulses = (
            wristAngle
            * constants.kTalonEncoderPulsesPerRadian
            * constants.kWristPivotArmGearRatio
        )

        self.topArm.set(Falcon.ControlMode.Position, topArmEncoderPulses)
        self.bottomArm.set(Falcon.ControlMode.Position,
bottomArmEncoderPulses)
        self.wristArm.set(Falcon.ControlMode.Position, wristArmEncoderPulses)
```

Due to our team's resources, virtual simulation is a huge part of our ability to quickly and reliably construct the bot's codebase. Some key examples of simulation are a wrapped onto a simulated falcon motor. Given our team's extensive use of falcons on the robot, a wrapper that provides simulation support allows for the programming team to iterate much easier and creates a cleaner codebase. Each falcon is logging the values of the motor % and the encoder position, as well as an override value to allow the user to manually in simulation change the value for sensor readings. Entire robot configuration is done on a single call and the getting of velocity, position, and percent and the setting of velocity, position, and percent are easy to access functions to allow interfacing with the motors more accessible than the CTRE library. Given this robot also has a NEO550, the simulation system was adopted to have a similar interface for ease of replacement from a falcon to a motor on the intake. We geometrically derived the inverse kinematics for 3 links with a fixed Pose endjoint. Each of these poses actually allows for two configurations of the proximal 2 arm joints (they can simply be mirrored over the line created from the wrist joint to the shoulder joint, however by forcing the sign on the elbow joint they can all be consistent.

SOFTWARE: VISION

NOW WITH APRILTAGS AND PHOTONLIB

We have a vision system complete with sensor fusion for complete robot localization. Last year,we worked with our first complete vision system as a team that resulted in significantly enhanced system performance, and using apriltags will be very important to account for combined sensor error as well as for being able to reliably use sensor data for automated alignment to various points on the field such as the double substation and the grid.

THE HOW

Photonvision generates camera-relative 3d transforms of each apriltag. Since the position of the camera is known and the position of the apriltag is known, the position of the robot can be determined from a single apriltag datapoint. These transforms are fed into a RobotPoseEstimator in order to create a sense of where the robot could be at a given time, this is combined with the gyro and wheel encoder information to get an accurate sense of where the robot is on the field at any given time. This is used in other subsystems when needed, as well as results being logged to AdavantageScope through the usage of each known pose and ghost poseseepaste

GOING FURTHER

We plan on using this odometry data to have automated alignment in complete robot space for important precision actions such as placement of gamepieces on the grid and collection of those gamepieces. Autonomous will also use this data. Perhaps an automatic engagement on the charge station by using the rotation gained from the apriltags will be possible. Overall having a sense of where the robot is on the field is beneficial to aid in other systems.

ENGINEERING TEAM

WHS Faculty Advisors

James Looney & Raul Madera

Team Captain

Charley Marsland*

Team Business Lead

Sean Tao

Team Technical Lead

Luke Maxwell

Senior Mentors

Dwight Meglan **Chris Aloisio°** **Steve Harrington°**

Mentors

Anthony Gelsomini **Manny Barros°** **Sean Lendrum°** **Mark Holthouse** *Mentor Emeritus* **Amber Maxwell**

STUDENT MEMBERS

Safety Captain
Jacob Kaplan

Graphic Design Lead
Ivan Cai

CAD Lead
Landon Bayer

John Santasuoso*	**David Girard***	**Declan MacDonald**	**Liam McWeeney**
Alex Theofilou	**Claire Peng**	**Andrew Gong**	**Rachel Qu**
Baili Jiang	**Nolan folmar**	**Jeffery Pan**	**Melissa Yang**
Divya Siddi	**Vinny Milinazzo**	**David Confoey**	**Adrianna Cirillo**
Elizabeth Lowney	**Jacob Liu**	**Anthony Yang**	**Constantina F.**
Luke Li	**Owen Monahan**	**Erik Curlli**	**Kevin Bai**
Henry Marsland	**Nina Pappas**	**Rachel Lee**	**Hanya Yang**
Sophia Patrick	**Lauren Buza**	**Kaylee Phu**	**Elias Mukdissi**

Westwood Robotics, Inc
Board Of Directors

Steve Harrington° *President* **Catherine Tseng** *Treasurer* **Manny Barros°** **Cheten Gopal** *Clerk* **Jack Tseng**

**Denotes High School Seniors* *°Denotes FRC Alumni*

SPONSORS

Heartlander Surgical	Westwood High School	Medtronic	Act Fasteners
Gene Haas Foundation	TE Connectivity	General Dynamics	IRA Motor Group
Bayer Familt	Capital Group	Yamaguchi Family	Plymouth Rock Assurance
Deadham Savings	Maxwell Family	Schneider Electric	Marsland Family
Porier Apliances	Franklin Templeton Investments	Jake and Joes Resturant	McGolf
Roche Bros		Capers Catering	

VENDORS

Andymark	West Coast Products	Vex Pro	Swerve Drive Specialties
Discord	Google Work	Onshape	Robo Promos
McMaster-Carr	The Home Depot	Cross the Road Electronics	MS Visual Studio
Python	GitHub	RobotPy	Java

1757

Special thanks to everyone who makes this team possible Including

Our Mentors
Our Students
Our Teachers
Our Parents

Our Amazing Sponsors

Westwood Public Schools
Emily Parks - Superintendent
Allison Borchers - Asst. Superintendent
Deb Nolan - Building Use Coordinator
Tom Carey - Director of Facilities
Ron Smith - Assistant Director of Facilities
Amy Davenport - Principal
Aishleen Flanagan - Assistant Principal
Tom Millett - Assistant Principal
Andrew Miller - Science Dept. Chair
WHS Janitorial/Facilities Staff
WHS Library Staff
The Open Alliance

#FRC1757
@WWRobotics1757