JDHS FTC ROBOTICS

Team The Hydraulic Hydras #9384

2023-2024

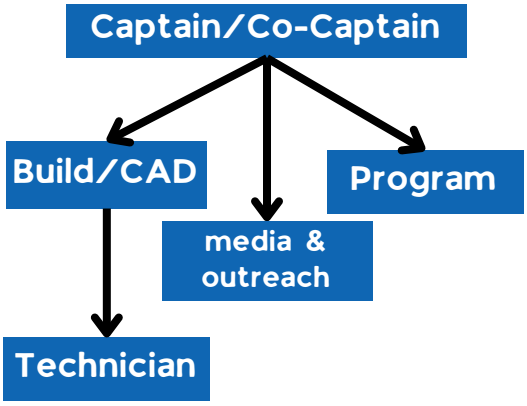
Introduction

Table of Contents

Robot Based Goals

Our team goals for the center stage season is to create a robot that can preform all ***3 major tasks*** this season. They include scoring the game element known as the pixel efficiently, climbing quickly and shooting the drone accurately into the 30 point zone. With these tasks being ***completed efficiently*** we can be a very high preforming robot.

Role Distribution



Who Are We

The Hydraulic Hydras are a FIRST Tech Challenge team from John Dewey High School from Brooklyn, New York. Currently the team is made up of 6 students ranging from 10th-12th grade and 3 mentors. All of our students are from John Dewey High School aside from Aiden who is from Midwood high school located in Brooklyn New York. Our rookie year was in 2014. This team has evolved throughout this time leaving legacies behind allowing new members to learn from past mistakes and adapting older ideas from previous
robots to newer robots.

As a team we embody FIRST core values in our everyday lives.
Whether it be being gracious throughout our day or being professional while helping someone who needs it. Our team has set these core values to create not only a better work environment but to also have a safe space for all.

The Hydraulic Hydras may be an old team however all of the students on the team have only 1 year of FTC experience. Due to this this season was a big learning process for our team.

Team Philosophy

Our team philosophy is screw it we ball. This phrase may sound silly but it has been displayed through our whole design, build and programming process through the idea that we need to disregard the current situation we are in and take control of the elements that are within our control. On top of this Through this we have spread gracious professionalism throughout our school due to our professional, respectful and helpful work environment.

Team Legacy

Our team has left many legacies from previous season within John Dewey High school. Whether it be having the ***first custom made chassis and mechanisms*** without the use of Rev Robotics based systems or building a simple battery powered robot cart. We have done a lot for our schools robotics teams. However we have never left a ***global impact*** until this year. Our Lead Programmer Tea created his own ***Custom Library "Hydraulic Lib"***. This library optimizes tuning, programming and robot movement on the field. There is also a programming notebook Tea wrote to ***help programmers everywhere*** learn to code both in blocks and java. This Library (once complete) will be similar to road runner but will ***provide new programmers a way to learn programming for FTC***.

About Us

Our Team

Tauha "Tea"

Captain, Lead Programmer, Diver, Lead Strategist

"I program the robot and do the PID moment."

(programmed the whole robot)

Aloe

Co-Captain, Lead Cadder, Technician, Coach, Engineering portfolio & notebook, Outreach

"CUSTOM CHASSIS WILL BE THE DEATH OF ME"

(built the custom chassis , outtake, and lift)

Aiana

Lead Media, Lead Engineering portfolio &, notebook, human player builder

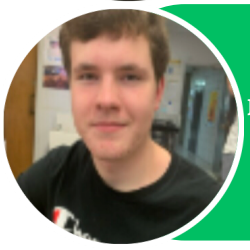
" I do everything on the team besides bulid the robot"

Aidan

Builder, Strategist, Technician

I am from Midwood High School. I go here because my Dad mentors our team and my sister graduated from Dewey also in robotics.

(bulit the intake on the robot)



Peter

Lead Technician, Cadder, Builder, Electrician

PETER MOMENT

(built the shooter and wired the whole robot)

Abdullah

Co-Driver & bulider

"Hard work beats talẻǹt when talent fails to put in the work"

Ailin

Designer

Mentors

Mr. Dispenza

Teacher, Mentor

A teacher at John Dewey High School for the last 15 years. He started the 4 FTC teams here at John Dewey in 2013 and 2014. He is a huge Star Wars fan and droid aficionado!

Mr. Tang

Mentor

Failure is the key to success; Don't give up easily

Anthony

Mentor

"What Tang said..."

Justin

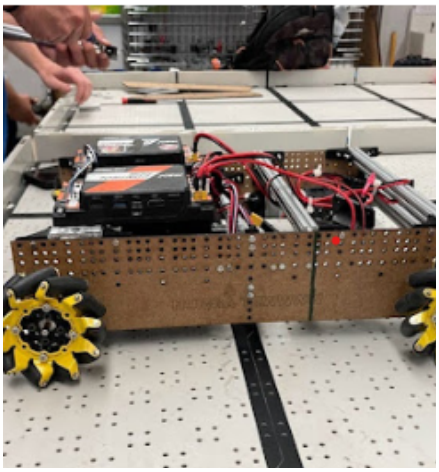
Mentor

The most inefficient systems produces the most efficient outcomes, the most efficient systems produce the most inefficient outcomes"

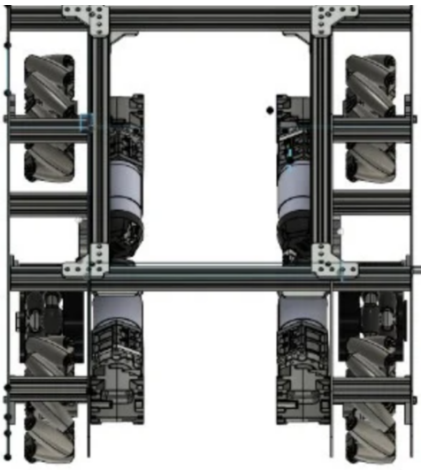
Design

Chassis

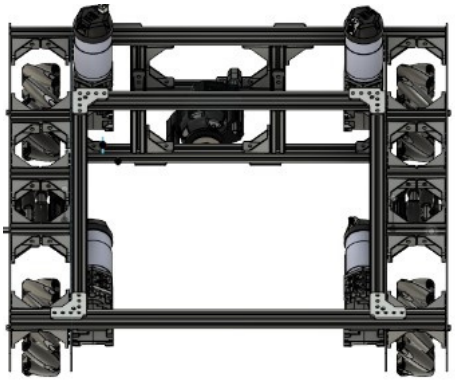
We initially designed the chassis with modular features, using small plastic L brackets for extrusion mounting. However, this led to structural challenges, causing inadequate rigidity and chassis flex under stress. To address this, we introduced three full-span extrusions and used metal L brackets for bolting, significantly improving rigidity and resolving flexing issues. Custom odometry pods were strategically placed for guidance. As the season progressed, we refined the chassis, incorporating dedicated mounting solutions for lift, intake, shooter, and climber systems, while maintaining the core extrusion-based structure.



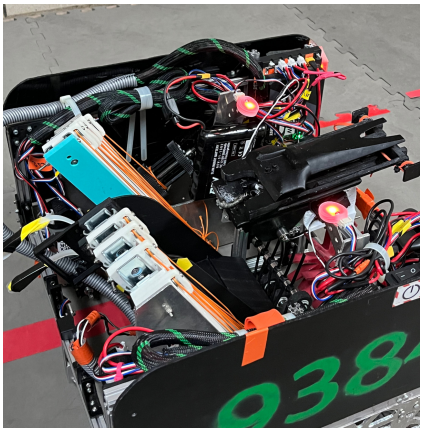
Original Prototype

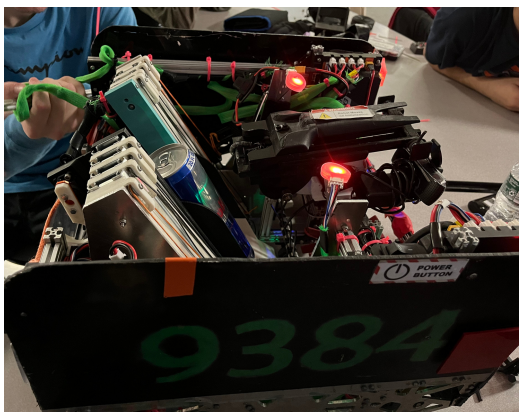


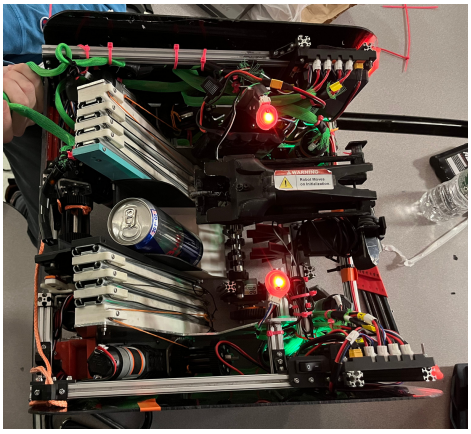
Final prototype



Final prototype with Odometry







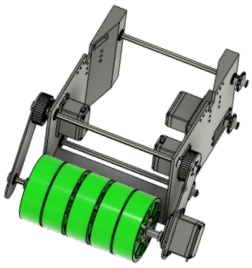
Final Chassis with final mechanisms

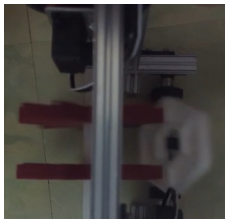
Design

Intake

When developing our intake system, our main goals were to efficiently pick up pixels from both the ground and a stack. The initial design utilizing two servos to lift a linkage, encountered challenges due to underestimating the torque of the servo. This impacted both the outtake and pixel transferring mechanism. Due to this we undertook a last-minute redesign just a day before our initial competition. The updated intake introduced two distinct wheel sets. One set are Roller wheels positioned closest to the floor to push the pixel into the robot. A separate set of star wheels are used to grab and push pixels into the robot. This dual-wheel approach improved our intaking efficiency. The final design uses zip ties to guide the pixel inside of our outtake.

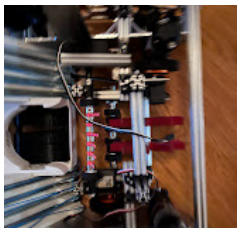
Original prototype





Star Wheel Prototype

Final intake



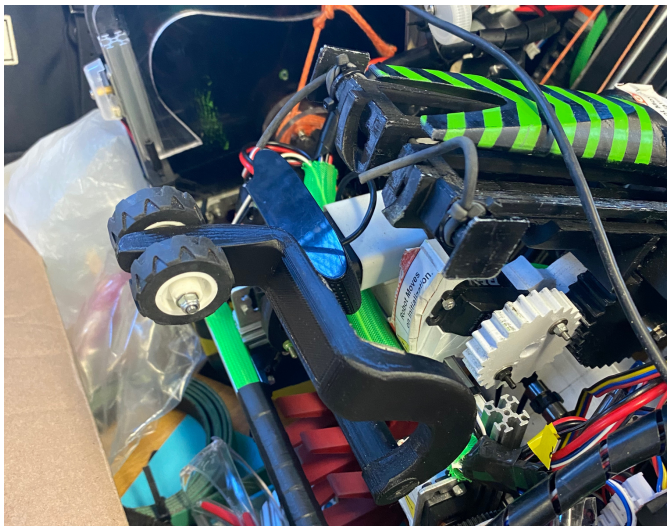
Original prototype

Pixel Claw

Due to limited space for additional mechanisms, we added a servo hook to assist the intake pushing the stack inwards. In version 1, the initial design showed potential but had limitations in angles. Version 2 improved by relocating the servo mount but faced issues with limited reach and brittleness. In version 3 we added toy car wheels to the front of the hook and a brass rod through the center to hold the wheels in place to our final design because to increase durability and griff. We added the wheels to the hook to help with traction and push down the stack easier.

Second prototype

Final assembled design

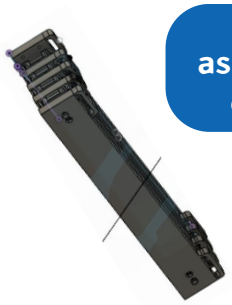


Design

Linear Slide

Original prototype

Second prototype

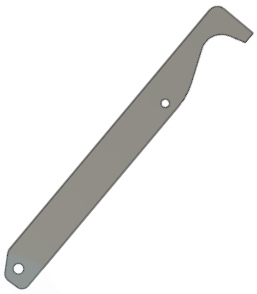


Final assembled design

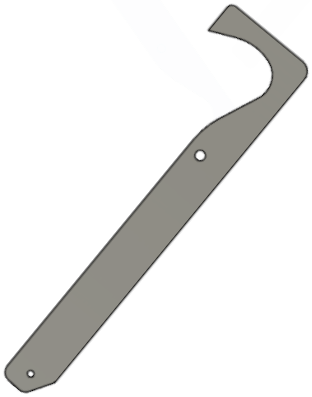
We found a problem with the design of the Rev Robotics linear slides: the small plastic guiding pieces caused friction and made the slider snap. After talking to other teams, we realized that ball-bearing-based slides would work better. We created a CAD design for these slides, but during manufacturing, we found that the weight caused the extended part to drop. To fix this, we added 3D-printed 95A TPU braces under the components, which absorbed the shock and made the slides nearly indestructible in this application

Climber

Original Hook

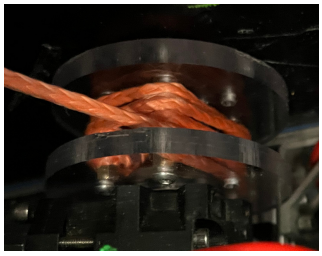


Final Hook

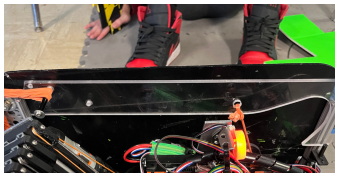


While brainstorming designs of climbing mechanisms we noticed that there wasn't much space on the robots chassis. Due to this, we decided to attach hooks onto our Lexan plates. This will allow us to manufacture hooks that are very long to make it easier for our drivers to line up and instantly climb once the hook collides with the truss. Our first hook allowed us to climb but the tip of it was too short. This prevented us from accurately and consistently climb every time. On our final hook design we extended the tip of it to allow for it to easily attach to the truss. After creating these hooks we attached them to string and the string to separate drums. These drums are made of Lexan and use brass standoffs for added stability. After rigging the drums up we created a suspension to provide constant force to the hook allowing it to be in its upward position when the string is loose.

Final Drums



Final Mechanism

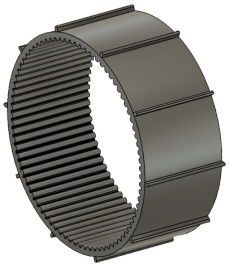


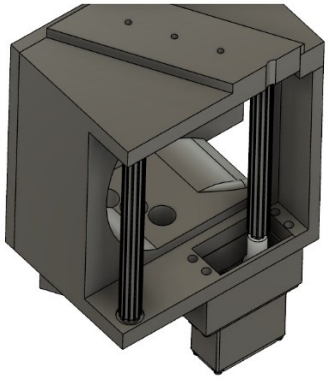
Design

Outtake

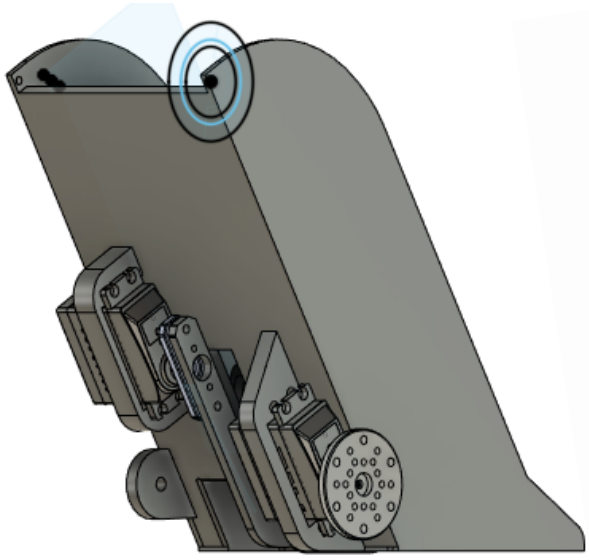
In the process of creating our outtake, our aim was to create a design that is simple and efficient. To achieve this, we crafted a design that used 3D-printed belts within a framed box. As we progressed the initial outtake design was inefficient and struggled with scoring on the backdrop. Issues arose, such as pixels falling from the outtake when not aligned with the backdrop. Drawing inspiration from our successful 2021 Freight Frenzy scoring mechanism, the new design used a rotatable box to control the pixel release. The initial implementation proved successful, ensuring consistent scoring. Another issue emerged when attempting to score multiple pixels together. The outtake holds pixels on top each other which resulted in both pixels falling out simultaneously which missed the backdrop.

To address this setback, we introduced a claw that allows us to independently score pixels. This allows the outtake to securely hold one pixel while facilitating the scoring of another.

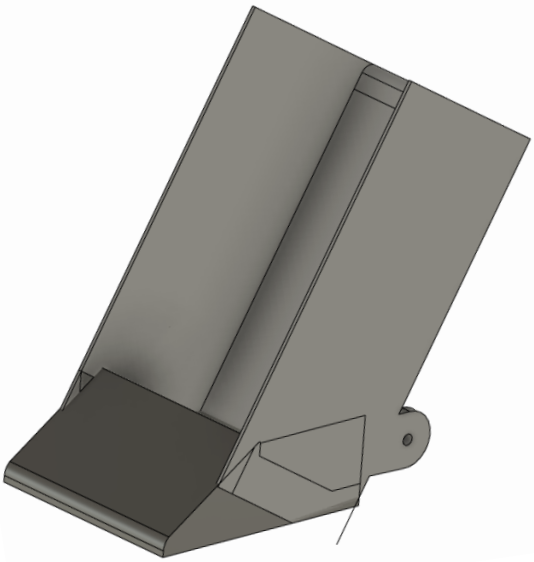




Original Box and Belt



Final Rotating Outtake with Hook

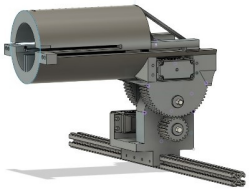


Updated Rotating Outtake

Design

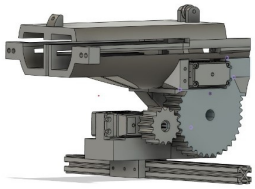
MK 1

MK 2



MK 3

MK 4



MK 6

Drone Launcher

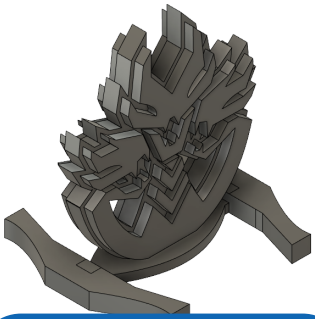
MK 5

When designing our launchers we noticed that we needed to use a lot of force to shoot our drone precisely. We did this by creating a mechanism that uses a rubber band to shoot the drone. The servo is located near the back of the launcher and releases a tensioned rubber band to shoot the drone. We also added another servo to alter the height of the launcher allowing it to change position depending where we are on the field. This launcher will be mounted on the intake.

Team Prop

Our original team prop was made using our logo from previous years. The shape of it allowed Tea to tune it for our autonomous. We updated the team prop to our new logo to allow for a cleaner more recognizable look. On top of this, on our final design we added a new stand to allow the prop to sit onto the field better and to be printed in less parts. It looks exactly the same so it wont interfere with the code that is already made.

Version 1 (Old Logo)



Version 2 (New Logo)

Version 3 (New Stand)

Design

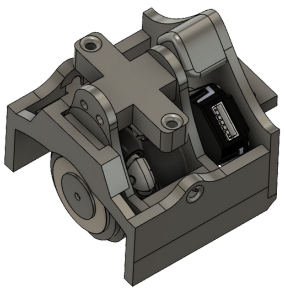
Odometry

During the design phase of our odometry pods we opted for a suspension-based pod to ensure consistent contact between the wheels and the field mats. This was achieved by integrating a spring mechanism on the side of the pod. This enables it to pivot downwards as needed. We utilized omni wheels with rubber rollers to enhance traction on the field surface. While the pods themselves functioned perfectly, placing them on the sides and back of the chassis presented an issue. During robot motion when driving over pixels, the setup resulted in the robot getting caught on the pixel. Our revised design retains the same wheel and absolute encoder configuration, with the only alteration being the suspension mechanism. We transitioned to an independent suspension system, where the spring facilitates vertical movement exclusively within the pod rather than employing a pivoting swing arm. This adjustment not only addresses the issue of the odometry getting stuck on pixels but also contributes to a more compact design.

Version 1

Final "Swing Arm"

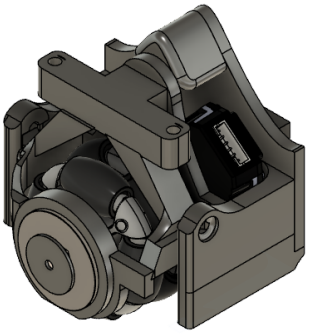
Version 3



Final Pod

Version 2

Version 4



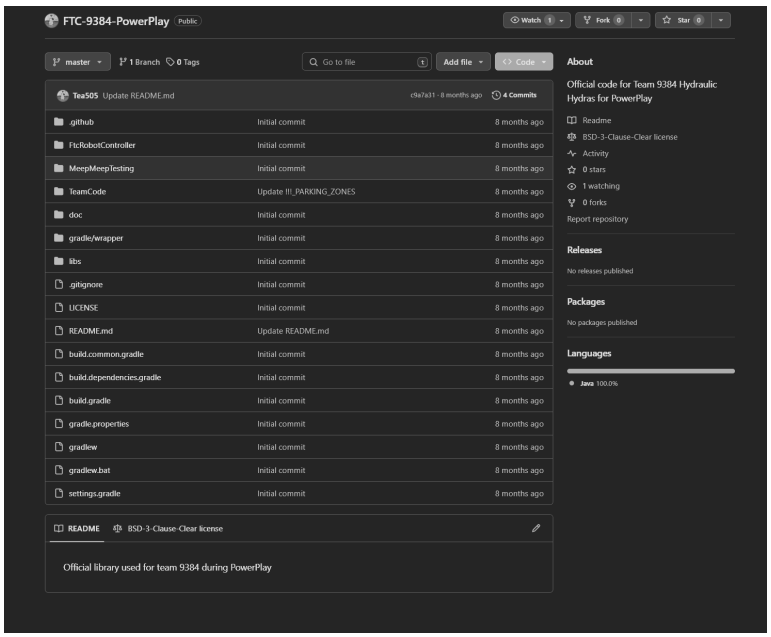
Programming

Introduction

We started the project by testing our drivetrain and through this he crafted a specialized programming library to enhance the intricacy of our code. Through this, our developmental custom library, dubbed “Hydraulic Lib” has been a long standing side project for Tea and will be released to the public once it is complete. This library extends far beyond our team and is intended to serve as a resource to everyone within FIRST. Due to this, the scope of our library will potentially contribute to the improvement of programming practices globally.

Hydraulic Lib

The “Hydraulic library” is a set of utility files designed to help programmers simplify the coding process. The project originated as a way to consolidate different projects into cohesive files, and it has undergone significant evolution over time. Our lead programmer has played a crucial role in creating custom libraries, and this journey has involved meticulous research and dedicated effort. The main focus of the library is to offer a versatile controller that can adapt to different drivetrains, such as X-drive, Mecanum, and Differential Swerve drive. This controller also encompasses mathematical equations, pathing sequences, and PIDF control systems to provide precise control across diverse robotic platforms. The library is still in beta stage and is an ongoing effort involving different team members who contribute enhancements and updates. This library is public to anyone on github.



Programming

General Implementation

Through the use of “Hydraulic Lib” we are able to fine tune our robots chassis, lift, intake, outtake and launcher. The chassis code uses a bespoke code that is tailored to our Mecanum wheels. This code allows us to go in omnidirectional paths with precision and agility. We are also able to control the speed of the robot while driving. Using pre-set controls. Through our chassis Tea needed to program the trajectories of our robot using our Odometry Modules. We currently have 3 on our robot which allows us to convert the encoder movement known as ticks to inches. We can use this to format our robots point on the field through the x and y position. We can use this data to build a trajectory allowing us to form waypoints for our robot to go to. These waypoints will allow us to go in both linear and spline movements. These movements allow our robot to achieve precise movements during autonomous. Our cascade sliders underwent lots of thought when programming since the straightforward motion gets converted into a motion profiled motor. This profile delineates our lifts trajectory so it is difficult to get a precise position of our lifts without the use of magnetic limit switches. Due to this, we added magnetic limit switches at the minimum and maximum position of our lifts. The intake outtake and launcher are more simple then this since we need to set the angles of our servo motors from one position to another or just have the servo spin in 2 directions when it is activated.

Chassis Implementation

In configuring our drivetrain, we opted for a mecanum drivetrain design. To achieve omnidirectional movement, we developed a bespoke code specifically tailored for the drivetrain. This custom code employs linear algebraic equations to enable the drivetrain to navigate along omnidirectional paths with precision and agility. Using this custom code we could also control the speed of our robot while driving as the program contains a method that creates speed states for our robot.

```
double [] wheelspeeds = new double[4];

// Denominator is the largest motor power (absolute value) or 1
// This ensures all the powers maintain the same ratio, but only when
// at least one is out of the range [-1, 1]
double denominator = Math.max(Math.abs(y) + Math.abs(x) + Math.abs(rx), 1);
double leftFrontSpeed = (y + x + rx) / denominator;
double leftRearSpeed = (y - x + rx) / denominator;
double rightFrontSpeed = (y - x - rx) / denominator;
double rightRearSpeed = (y + x - rx) / denominator;

if (gamepad1.left_bumper) {
    powerMultiplier = 0.5;
}   else {
    powerMultiplier = 1;
}

wheelspeeds[0] = leftFrontSpeed * powerMultiplier;
wheelspeeds[1] = leftRearSpeed * powerMultiplier;
wheelspeeds[2] = rightFrontSpeed * powerMultiplier;
wheelspeeds[3] = rightRearSpeed * powerMultiplier;

ws[0] = wheelspeeds[0]; // left front
ws[1] = wheelspeeds[1]; // left rear
ws[2] = wheelspeeds[2]; // right front
ws[3] = wheelspeeds[3]; // right rear
```

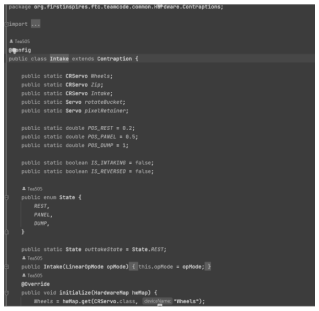
Programming

Intake Implementation

```
Tea505
public static void startIntaking() {
    IS_INTAKING = true;
    IS_REVERSED = false;
    Wheels.setPower(1);
    Zip.setPower(1);
    Intake.setPower(1);
}

Tea505
public static void stopIntaking() {
    IS_INTAKING = false;
    IS_REVERSED = false;
    Wheels.setPower(0);
    Intake.setPower(0);
    Zip.setPower(0);
}
```

Intake is compiled with three servos that use maximum power to take in the pixels, this three servo mechanism allows the pixel to be intaked extremely fast and the star wheel has a strong grip which allows it to instantly respond during autonomous.



Lift Implementation

```
new *
public void stopSlides() {
    LeftCascade.setPower(0);
    RightCascade.setPower(0);
}

Tea505
public void telemetry(Telemetry telemetry) {
    telemetry.addData( caption: "Left Position: ", LeftCascade.getCurrentPosition());
    telemetry.addData( caption: "Right Position: ", RightCascade.getCurrentPosition());
    telemetry.addLine();

    telemetry.addData( caption: "Direction of Left: ", LeftCascade.getDirection());
    telemetry.addData( caption: "Direction of Right: ", RightCascade.getDirection());
    telemetry.addLine();

    telemetry.update();
}
```

The mitsumi lift is powered by a Dc Motor with a gear ratio of 45:1 which provides enough torque for it to lift up. With that there's two magnet limit switches attached to the lift that limit the height of the lift so it doesn't exceed it maximum and minimum position.

Outtake Implementation

```
Tea505
public void outtakeLoop(Gamepad gamepad) {

    if (gamepad.a) {
        // Intake Position
        rotateBucket.setPosition(POS_REST);
        outtakeState = State.REST;
    } else if (gamepad.b && !Mitsumi.low_Limit.isPressed()) {
        if (rotateBucket.getPosition() == 1 || rotateBucket.getPosition() == 0) {
            // Parallel
            rotateBucket.setPosition(POS_PANEL);
            outtakeState = State.PANEL;
        } else if (rotateBucket.getPosition() == POS_PANEL) {
            // Drop
            rotateBucket.setPosition(POS_DUMP);
            outtakeState = State.DUMP;
        }
    }

    if (gamepad.x) {
        // open
        pixelRetainer.setPosition(0.46);
    } else if (gamepad.y) {
        // grab
        pixelRetainer.setPosition(0.42);
    }
}
```

Outtake is a bucket mechanism with a lock system compiled with two servos from which one servo rotates the bucket to position it and dump the pixel and then there's a servo attached at the bottom that holds one pixel still so we can align it for a Mosaic. This also allows us to have more control during auto as well so the pixel can be placed precisely.

```
Tea505
@Override
public void loop(Gamepad gamepad) {
    if (gamepad.right_trigger > 0) {
        // Intake
        Wheels.setPower(1);
        Zip.setPower(1);
        Intake.setPower(1);
    } else if (gamepad.left_trigger > 0) {
        // Outtake
        Intake.setPower(-1);
        Wheels.setPower(-1);
        Zip.setPower(-1);
    } else {
        Wheels.setPower(0);
        Intake.setPower(0);
        Zip.setPower(0);
    }
}
```

Shooter Implementation

Drone shooter consists of two servos, one that loads and shoots the servo and the other that manages its position. It is also linked to two LED lights that start blinking to indicate that the Launcher is in position and ready to shoot.

Programming

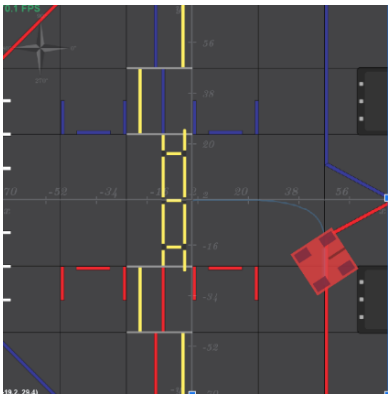
Sensor Implementation

Our robot is equipped with various sensors and tools to improve our performance in both tele-op and autonomous modes. We use two Magnetic Limit switches on our lifting mechanism to allow our robot to know the highest and lowest points it can go. The motor encoders on the lift allow us to get precise values for the lifts position. Additionally, our camera helps us detect our team's prop. During this detection process LEDs on the left, right and center of the robot visually show the detected region of the prop. Once detected our robot will move to certain scoring locations.

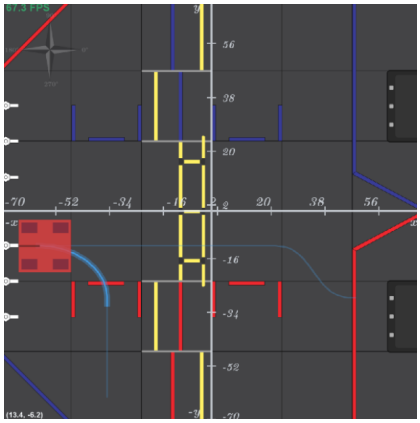
For autonomous movement, we rely on the motor encoders on the chassis to determine our robot's position on the field accurately. A distance sensor helps the robot stop when it reaches the backdrop, ensuring it doesn't go beyond the intended area. In manual control, the same distance sensor helps align the robot to the backdrop, assisting the drivers in moving it effectively.

Once endgame begins, our shooting mechanism initializes. Once our drive team wants to shoot the mechanism will pivot and 2 of our LEDs will blink. This allows other teams to know when we will shoot. Inspiration for the way our LED’s are used in end game come from CNC machines. The machines we use visually display information about sub-mechanisms within their enclosures.

Autonomous Path



Depending on randomization of the team prop the robot has multiple trajectories built into one autonomous. Depending on the location of the prop a trajectory is run. Each trajectory is different from the other. Our fastest has been 13 seconds. During our autonomous path we use a distance sensor on the back of the robot for a fail safe. Once a robot comes into our path, our robot will stop moving which prevent the robots from colliding. This failsafe method uses a finite state machine. The finite state machine uses multiple cases and switches the case based on the sensors reading and state of the autonomous.



Programming

Autonomous Path V2

After last qualifier we tuned our robot more and finished an auto for each side. After finishing the autos we decided to hit the white stack during auto for further points. We made two more autos for the far sides of the field and in one auto we only grab one extra white pixel and in the other auto we go for two cycles meaning we pick up one white pixel at first and then go to pick up two more white pixels after scoring.

```
.addTemporalMarker(Intake::fingerDown)
.waitSeconds(0.7)
.lineTo(new Vector2d( x: 63.1,  y: 12))
.addTemporalMarker(Intake::fingerReset)
.lineTo(new Vector2d( x: 63.1,  y: 21.5))
.addTemporalMarker(Intake::startIntaking)
.waitSeconds(1)
.lineTo(new Vector2d( x: 63.1,  y: -10))
.addTemporalMarker(Intake::stopIntaking)
.lineTo(new Vector2d( x: 63.1,  y: -67))

.addTemporalMarker(() -> mitsumi.autoMoveTo( pos: 1300,  power: 1))
.splineToConstantHeading(new Vector2d( x: 28.5,  y: -77.5), Math.toRadians(0))
.addTemporalMarker(() -> Intake.rotateBucket.setPosition(Intake.POS_PANEL))
.waitSeconds(0.6)
.addTemporalMarker(() -> Intake.rotateBucket.setPosition(Intake.POS_DOUBLE_DUMP) )
.waitSeconds(0.7)
// Remove these 2 markers if going for a 2 + 3
.addTemporalMarker(() -> Intake.rotateBucket.setPosition(Intake.POS_REST))
.addTemporalMarker(() -> mitsumi.autoMoveTo( pos: -200,  power: 0.75))
```

General Improvements

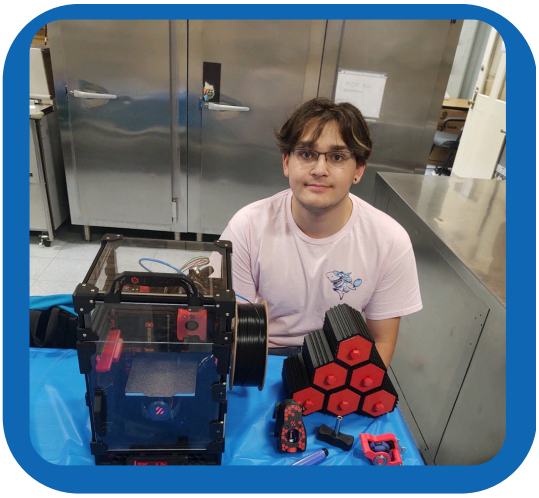
After Qualifier #4 we decided to switch a few things. First thing we switched is the camera model and we went from a logitech c270 to a logitech c920 pro. Using this new camera it provided us with better results as it could process more frames per second and had a higher resolution. This higher resolution drew clear images that defined more edges of our team prop making it easier and more accurate to detect.

C270

C920

Out Reach

3D Printing with PS/IS 121



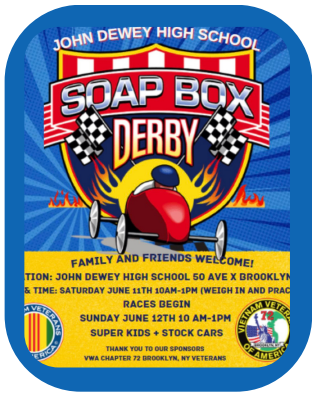
PS/IS 121 host a day for the stem and science activities expo every year. This year we decided to collaborate with ps 121 and work with the students and teach the basics of 3d printing and how it impacts the world. and showing how our robots uses 3d printing

Middle School Open House

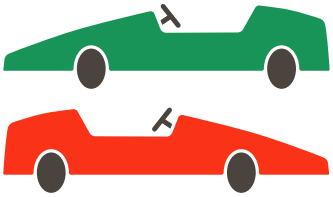
Every year John Dewey hosts open house events to welcome middle schoolers into high school and talk to them and introduce them to our schools resources and teams . During this event all 5 robotics teams talk about FIRST Robotics, our teams, other local FTC and FRC teams and how these students can get involved in FIRST within and outside of our school.

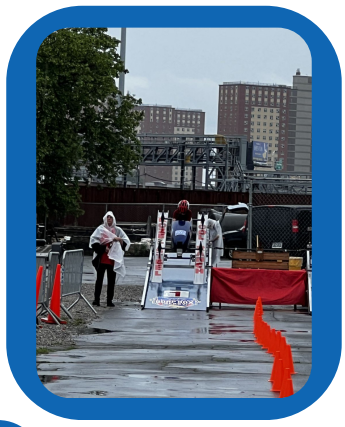
Out Reach

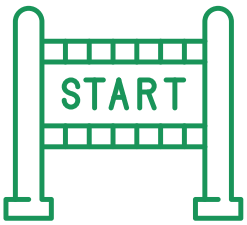
Soapbox Derby



A gravity powered racing event held by John Dewey. The cars were made by the John Dewey robotics team. The car we made was driven by team members during practice and by our ACES students during the qualifier. John Dewey has been hosting this event to our community, after the FIRST championship since 2022.





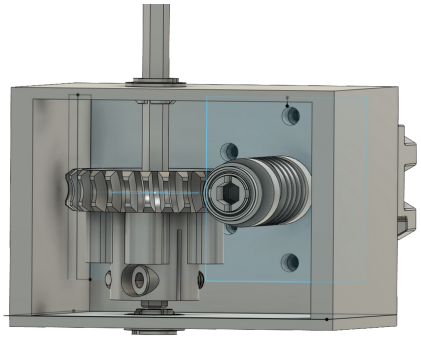


FIRST Team Chats

Lionotics

FTC Teams 5361 13475
Discussed robot designs

- Linear actuator climber
- Worm gear climber mechanism
- Shared CAD renders of ideas we have made



High Voltage Robotics

FRC team 369

- Fixed / did maintenance 3d printers