


#27518

# Meet The Team

## BUILDING

- Ashe
- Daxton
- ★ Destiny
- ★ Genesis
- Gael

Our building team works tirelessly to put together the robot and make continuous improvements within tight time frames.

## SOFTWARE

- Destiny
- Daxton
- ★ Nikki
- ★ Valeria

The software team works together to learn and write code that will automate our robot.

## CREATIVE

- ★ Andrea
- Ashe
- Austin
- ★ Genesis
- Josie

The creative team adds flare to our party. They decide the color schemes, themes and designs of our logos, banners, and posters.

## OUTREACH

- Camryn
- ★ Jennifer
- ★ Josie

Our outreach team strengthens our local presence; they reach out to sponsors, organize brochures, and manage social media.

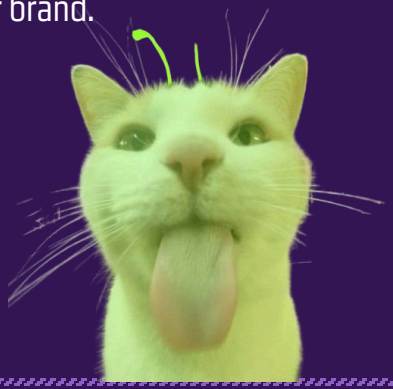


# SEASON GOALS

## 2026 FTC Season:

At S.T.A.R.S, we value everyone's quirks, hobbies, and skills. Our meetings don't just prioritize coding and building, but also the development of our social media, business outreach, and maximization of our design. Our authenticity (and silliness) is our brand.





## CHANGES

For simplicity: Block Code → JavaScript

This season, we decided to make more bold changes to our construction with 3d printing and concept drafts.
Due to our recruitment via brochures and social media, we dversified our team's skill set.

Nobody is truly set to one team; we all try to work together when it comes down to it. Our builders can be artists, and our drivers can be coders.

We budgeted better and double checked what would be useful for this season and the seasons ahead.

We put our funds to a new computer for programming-A Lenovo ThinkPad- and a second (prototyping) robot to try out the viperslides before we put them on our main robot.

## OUR SPONSORS:

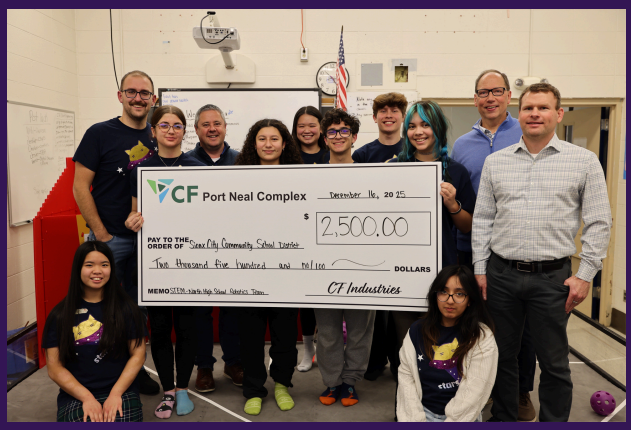
UnityPoint Health Sioux City

CF

Avery Brothers Sign Company

Gene Haas Foundation

# ----OUTREACH----



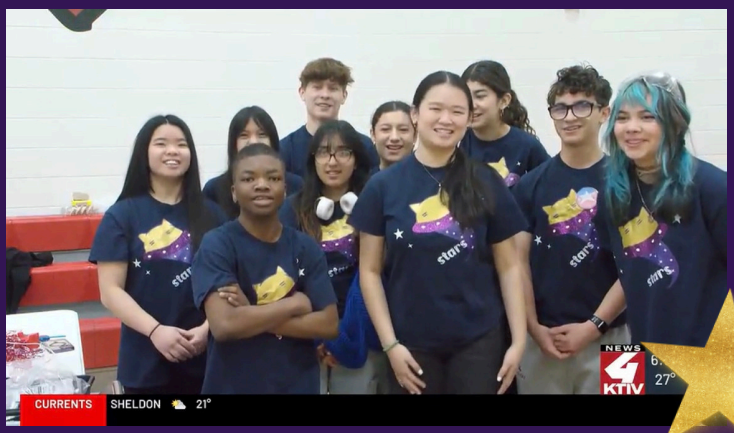

After reaching out to various companies, CF Industries, our local engineering manufacturer, decided to grant us a check for $2,500 to aid us in paying for GoBilda parts.




Brochures: our handout to give intrested young engineers an idea of our club.



Our builders took advantage of the local Career Academy tours, showing off the robot's mechanisms to the middle schoolers, even allowing some to assist in taking apart the robot for our second prototype.




We were not afraid to discuss our process with reporters and any teams who asked.

# MORE OUTREACH

Our CAD designer Daxton used TinkerCad to make a 3d version of our mascot; these were handed out to every alliance team we had during the Orange City Meet.

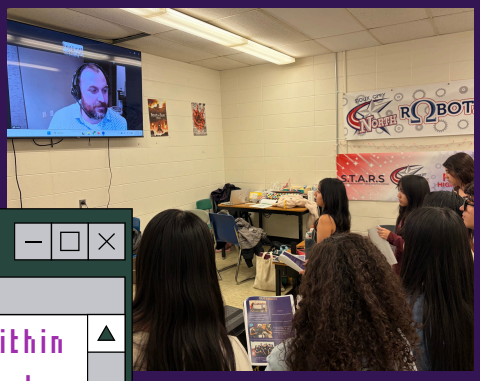







## Virtual Outreach

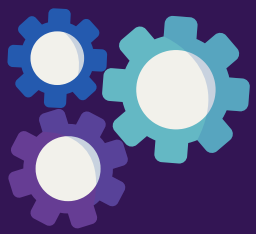
We presented our robot to Andy Neary, a manufacturer/ industrial engineer who works at JW Bell. He builds components for construction and mining equipment/ accessories.
He gave us notes and advice on our process, suggesting how to make educated costs, and telling us to emphasize our thought process behind decision making.



**ADVICE HE GAVE:**

Employers look for experience within the general field and willingness to learn more about that field like hands on experience. "It really boils down to your personality and how you work with others within in the field and who can deduct good thoughts.

Our team also attended a STEM activities fair during April, concluding the last season. We showed off our build to 8th graders and introduced them to robotics.



We also did a collaboration event with the Math CLub at North High, Sioux City, Iowa.

**MATH CLUB**

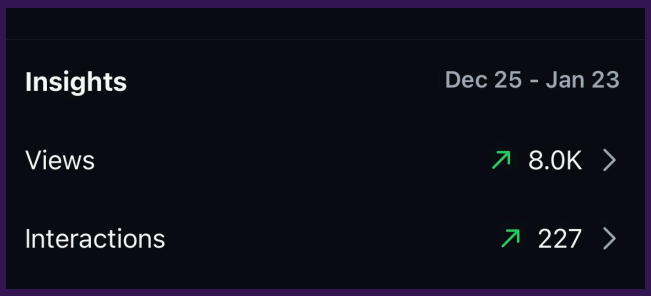
Tuesdays
3pm -4:10pm
@room 257



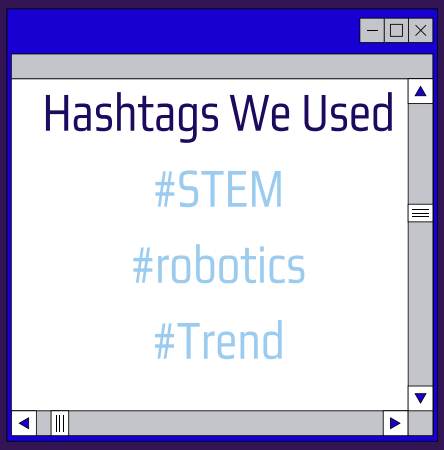
# SOCIAL MEDIA

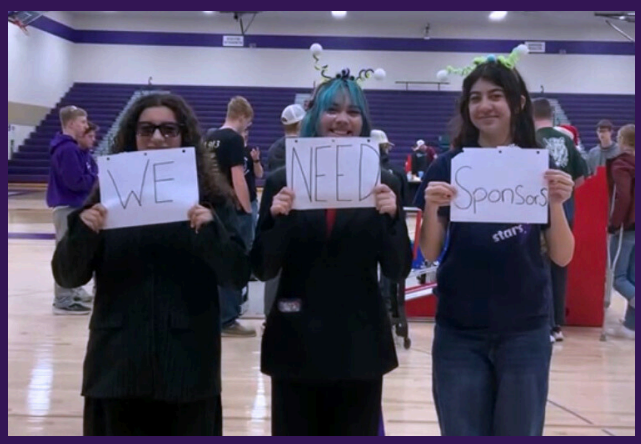
Since our last season, we decided it was best to expand our social platform to recruit more members, garner support, and connect with other Sioux City North pages.



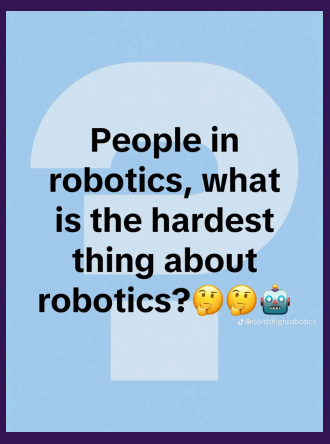

Our social medias are not just a side gig. We try to find ways to engage with engineers, coders, and kids interested in STEM.
We find opportunies to communicate to teams to have a bonding community through out the FTC season!










Our social media team also worked quikly to hop on new trends; surely they worked because we got more feedback from other north high students and club pages (East & West High).

# BUILD OVERVIEW

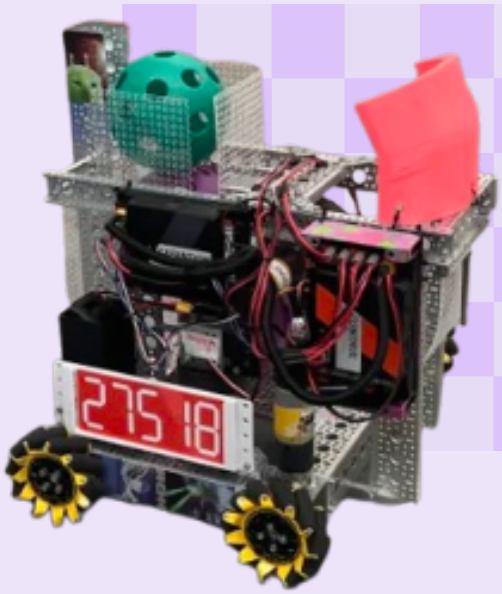


WIRING

Wires in protective wire wrap. Secured by zip ties and ribbons.

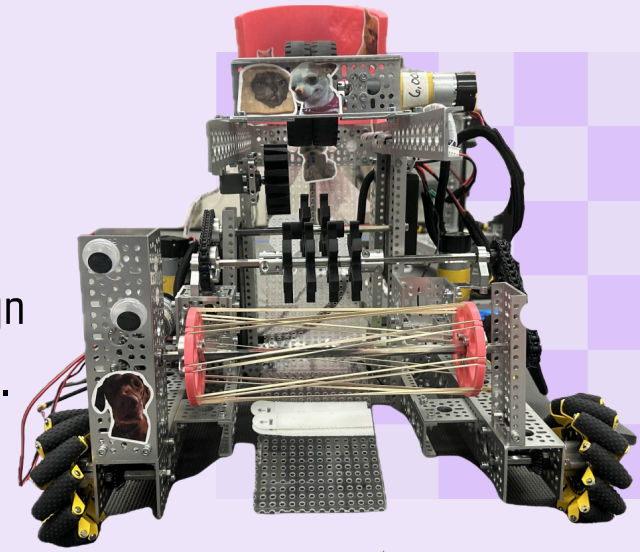
## 1 First Prototype

Rocking the starter bot! We added mechanum wheels, weights in the front, a shooting hood in the back for better consistency.



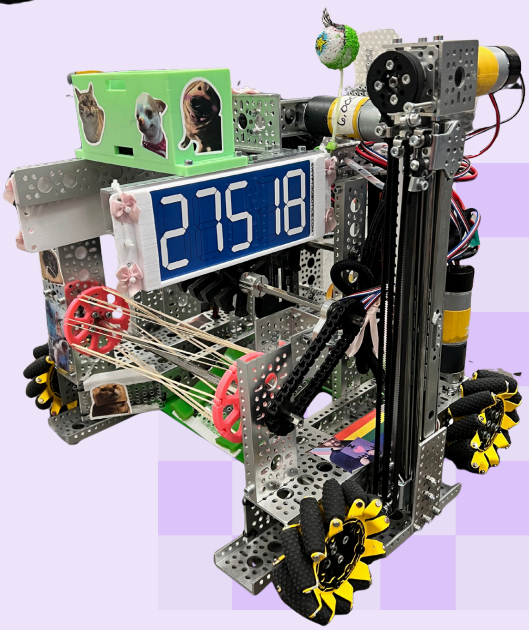
## 2 Second Prototype

We completely removed the front to make room for an intake. This was a huge turning point in our robot's design and it was the most challenging build.



## 3 Final Prototype

We refined the robot to intake more efficiently, shoot more consistently, and lift up using viperslides.



# FIRST PROTOTYPE

## MECANUM WHEELS

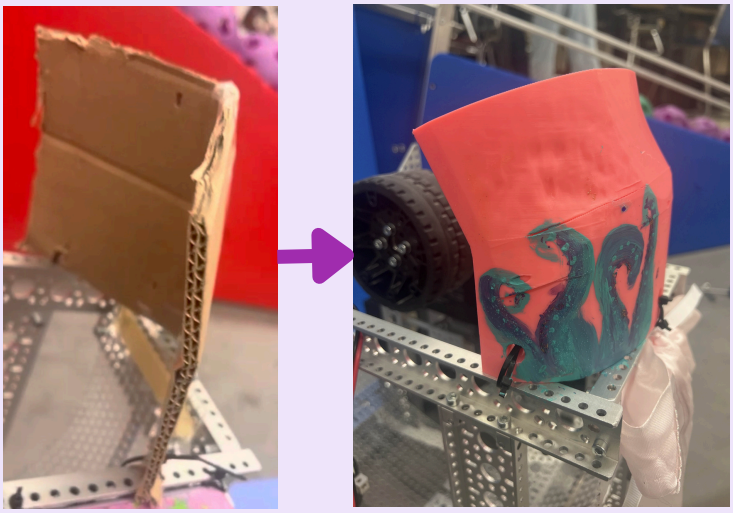
We knew we wanted mecanum wheels rather than the omni wheels that came in the kit. We preferred the motors to be lay horizontal, but four would not fit. So, two motors in the back stood vertically.



## SHOOTING HOOD

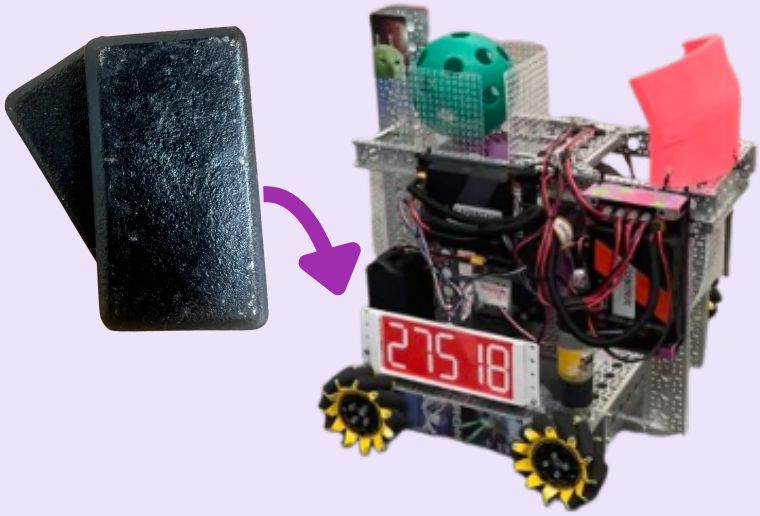
The most prominent issue was the inconsistency of our launching. The artifacts would frequently launch almost straight up and to the right.

We made a ramp using cardboard to find the right angle and how much cover the sides would need. Then, a 3d printed part was made to replace it.



## WEIGHTS

The back was too heavy, causing the robot to tilt and sometimes flip over backwards. So, we added these mini weights between the U-channel in the front. However, adding too many caused the back wheels to lose contact with the tiles. Two weights were perfect.

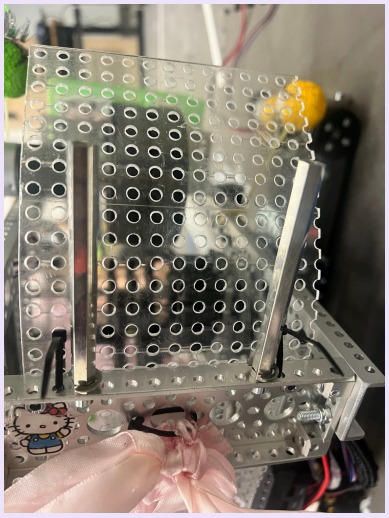



## Concerns:

While we were comfortable with our robot. The most obvious concern was that we didn't have an intake.

# FINAL PROTOTYPE

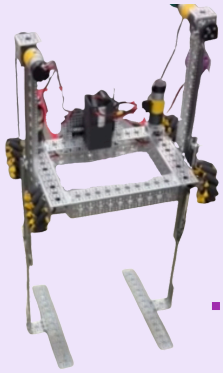
## ADJUSTMENTS



> Without a shooting hood, the artifacts did not have enough contact between the flywheel and a surface, causing inconsistencies. But a 3D shooting hood was too stiff. So, we simply extended the ramp with polycarbonate grids because it is flexible yet sturdy. The shafts make sure that they don't bend too much.

> The second-stage intake was moved forward by a single centimeter. This small adjustment improved the performance of our intake.

## VIPER SLIDES

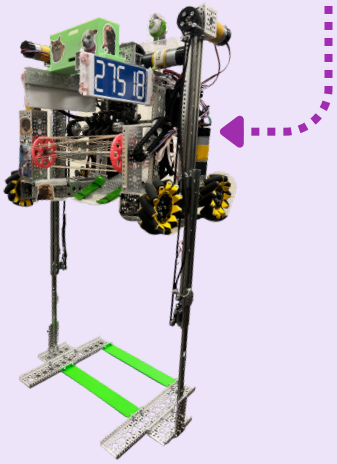


We worked on the viper slides that would lift our robot using two 117 RPM motors to achieve level-2 parking. They were attached to a second chassis to test.

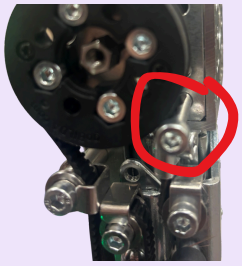
**Concerns:**

The lift may tilt. And because of gravity, the chassis would gradually drop down.

We attached the slides to our main chassis. And it lifted! We also added 3d printed channel bracket to keep the viper base channels aligned so that they wouldn't get stuck in the wheels and keep the robot stabled.

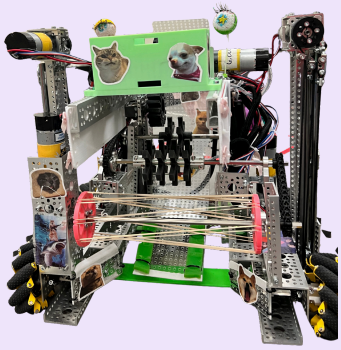


**Challenges:**

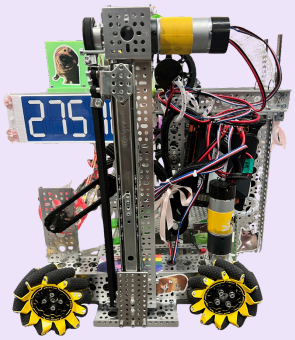


The viper slides would not work sometimes because the belt didn't have enough contact with the pulley. We put a screw to keep the belt from slipping.

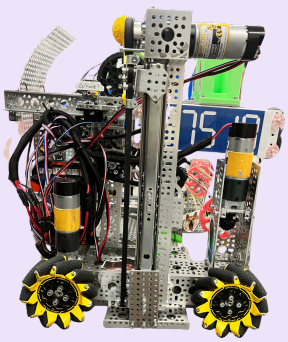
Front



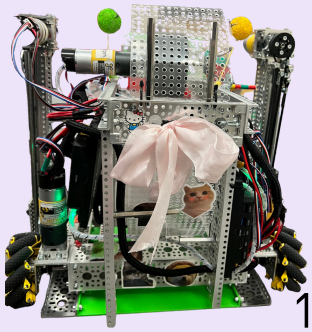
Right



Left



Back



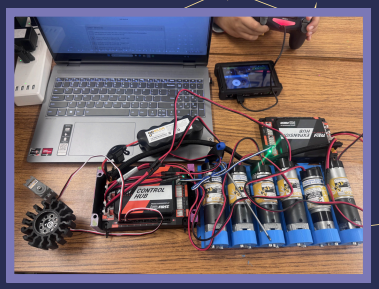
# TELEOP PROTOTYPE I:

File Edit Search Help

TeleOp code was originally taken from the GoBulida website. We tweaked the code to add two motors for mecanum wheels.

```
private DcMotor leftFrontDrive = null;
private DcMotor rightFrontDrive = null;
private DcMotor leftBackDrive = null;
private DcMotor rightBackDrive = null;
private DcMotorEx launcher = null;
private CRServo leftFeeder = null;
private CRServo rightFeeder = null;
```

While the builders worked on the robot, we built a testing platform with 3D printed parts to test our motors and servos.



We used the testing bench to make sure the controller buttons matched with the proper mechanisms, and to make sure all parts were functioning properly. In the process, we realized the servos were reversed.

We started the season with block coding. However, we switched to Java code to have easier use with AI as an assistance tool for resolving issues in our code.

# AUTONOMOUS PROTOTYPE I:

We had ChatGPT take our TeleOp code and give us a framework of a Autonomous code. We adjusted it to start an Auto with just a launcher. After tweaking the code for the launcher to work efficiently, we added movement to the code. Our first autonomous goal was to shoot three artifacts from the closer launch zone, and then move out of it.

**Issues of First Autonomous:**

1. The Autonomous code was showing up as a TeleOp code.
2. Artifacts kept bouncing out of the classifier.
3. Artifacts could not reach the goal.
4. Needed movement for robot to leave launch zone.

```
@Autonomous(name = "RedLauncherOnlyAuto", group = "StarterBot")
//@Disabled
public class RedLauncherOnlyAuto extends LinearOpMode {
```

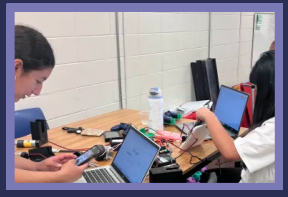
**Solution**

Fixing issues of the robot required a lot of trial and error. This consisted of switching parts of the code to test different angles and distances for launch accuracy, and to ensure the robot leaves the launch zone. To test these factors, we adjusted launch velocity and modified coordinates.

**Second Autonomous**

When in the farther launch zone, we had a simple autonomous that just moved forward to leave the launch zone.







# TELEOP PROTOTYPE II:

File Edit Search Help

We removed the two servos that fed the artifacts to launch them. Instead, we added a motor and servo for the intake. This would allow us to collect artifacts but also feed artifacts to launch them. Our goal was to have a button assigned to intake/shoot artifacts, but also another button to reverse the intake.

```
rivate DcMotor intakeMotor = null
rivate Servo intakeServo = null;
```

At first, we coded the intake separately from the TeleOp Code. When the motor and servo were functioning correctly with the controller, we transferred the code to our TeleOp.

When transferring the code, it had many errors but we went through each failed part and problem solved. If we could not find a fix, we ran it through ClaudeAI to give us an idea of the issue.

As we tested the code, the servo was continuously moving. The original code had the servo set to a continuous servo. We realized the problem was that the servo needed to be set as a standard servo.

# AUTONOMOUS PROTOTYPE II:

For our original Autonomous to function properly, we needed to add the mechanisms of the new intake motor and servo to the code. However, we did not create a code with collecting artifacts.




**First Autonomous**

With the new prototype, we kept most of the same code. However, we needed to remove the past feeders and replace them with the intake mechanisms to feed the artifacts to the launcher.

Once the code worked, we adjusted the coordinates and launch velocity again to ensure proper angles, movement, and launch accuracy.

**Second Autonomous**

During our most recent competition, we realized our robot moved too far out. We adjusted coordinates until the robot was set at the perfect distance.

# TELEOP FINAL PROTOTYPE:

Now that the building team made viper slides, we had to combine our viper slide code & tele Op code together.

```
if (gamepad2.y) {
    launcher.setVelocity(LAUNCHER_TARGET_VELOCITY);

} else if (gamepad2.b) { // stop flywheel
    launcher.setVelocity(STOP_SPEED);
}
/*
 * Now we call our "Launch" function.
 */
launch(gamepad2.right_bumper);

/*Servo*/
if (gamepad2.x) {
    intakeServo.setPosition(SERVO_FORWARD);     // X = Spin forward
} else if (gamepad2.a) {
    intakeServo.setPosition(SERVO_BACKWARD);    // A = Spin backward
} else {
    intakeServo.setPosition(SERVO_STOP);
}
```

StarterTeleop Code

```
int holdPosition = leftSlide.getCurrentPosition();

if (Math.abs(gamepad1.left_stick_y) > 0.1) {
    // Manual control
    leftSlide.setMode(DcMotor.RunMode.RUN_USING_ENCODER);
    rightSlide.setMode(DcMotor.RunMode.RUN_USING_ENCODER);

    double power = -gamepad1.left_stick_y * 0.6;
    leftSlide.setPower(power);
    rightSlide.setPower(power);

    holdPosition = leftSlide.getCurrentPosition();
} else {
    // Hold position
    leftSlide.setTargetPosition(holdPosition);
    rightSlide.setTargetPosition(holdPosition);
```

Viper Slide Code

```
    //// SLIDE CONTROL////////
if (gamepad1.left_bumper){
leftSlide.setMode(DcMotor.RunMode.RUN_USING_ENCODER);
rightSlide.setMode(DcMotor.RunMode.RUN_USING_ENCODER);
leftSlide.setPower(0.6);
rightSlide.setPower(0.6);
holdPosition =leftSlide.getCurrentPosition();
    ////Manual Control////
} else if (gamepad1.left_trigger > 0.1){
leftSlide.setMode(DcMotor.RunMode.RUN_USING_ENCODER);
rightSlide.setMode(DcMotor.RunMode.RUN_USING_ENCODER);
leftSlide.setPower(-0.6);
rightSlide.setPower (-0.6);
holdPosition =leftSlide.getCurrentPosition();

     * Show the state and motor powers
     */
    telemetry.addData("State", launchState);
    telemetry.addData("motorSpeed", launcher.getVelocity())
    telemetry.addData("Intake Motor Power", "%.2f", intakeM
    telemetry.addData("Left Slide Encoder", leftSlide.getCu
    telemetry.addData("Right Slide Encoder", rightSlide.get
    telemetry.addData("Slide Difference", leftSlide.getCurr
}
```

Combined Code

# AUTONOMOUS FINAL PROTOTYPE:

**First Autonomous**

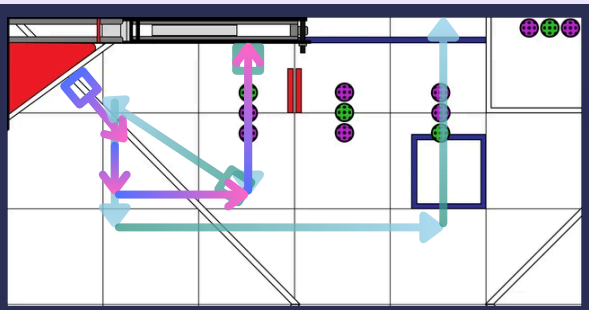
We coded an auto using timers that:

1. launch 3 loaded artifacts
2. intakes the nearest pre-staged artifacts
3. launch those artifacts
4. Strafe and collect the farthest pre-stage artifacts.

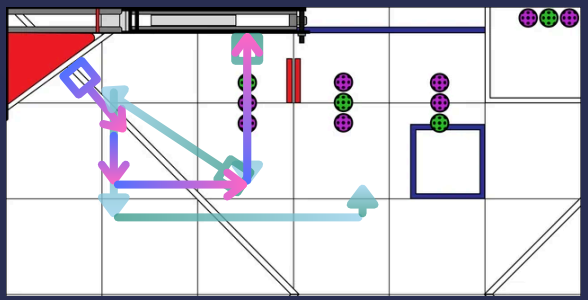
Problems:
Super inconsistent.
Movements changed each time.

**Final Autonomous**

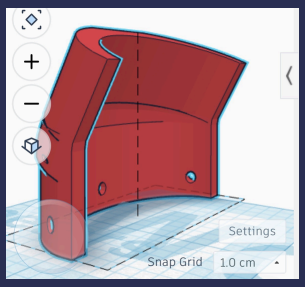
We learned that encoders would give us more accurate results. So, we converted and adjusted our code. Since it works best under 0.5 power, the autonomous was slower. Instead of step 4, the robot would stop and face the middle pre-stage artifacts. This works out because it lessens the risk of hitting the gate or the ally's bot.



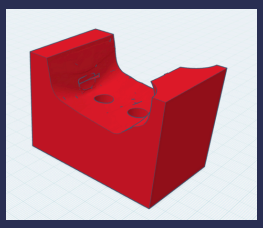
*mirrored on blue side



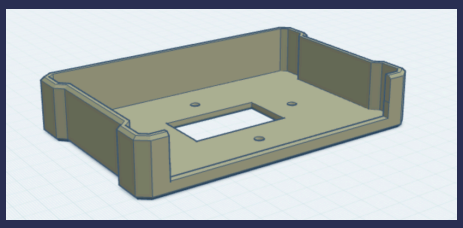
# CAD



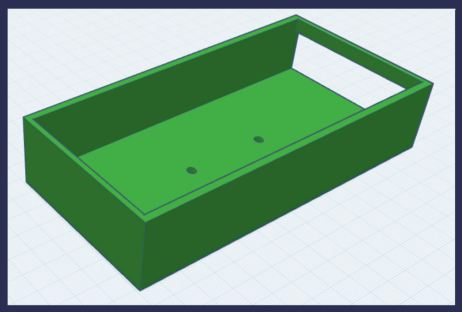

Item: Shooter Hood for First Prototype
Designed by: Destiny
Date: 11/18/25



Item: Controller Holder
Designed by: Daxton
Date: 11/17/25



Item: Driver Station Holder
Designed by: Daxton
Date: 12/08/25



Item: Power Bank Holder
Designed by: Daxton
Date: 11/9/25



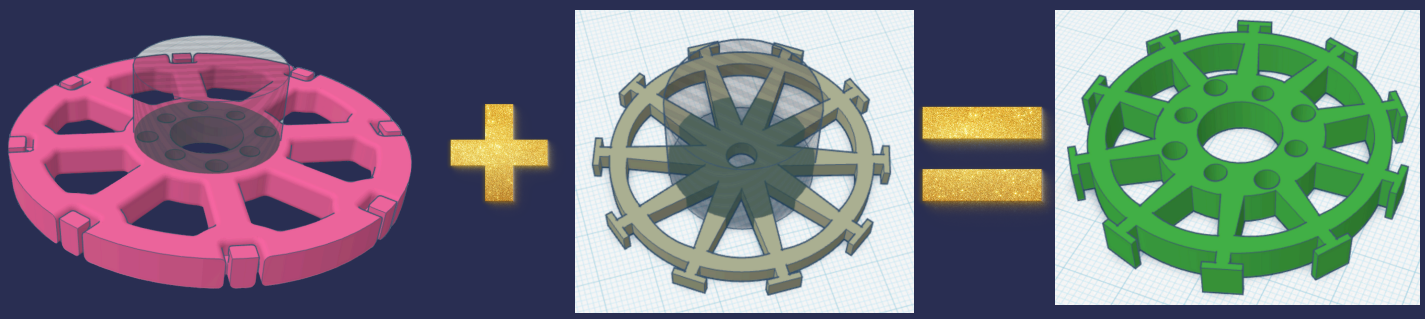
Item: Battery Holder
Designed by: Destiny
Date: 1/5/26

3D printing gave our team many opportunities to improve our robot and knowledge about CAD.

Our robot's uniqueness is in part due to its 3D components: Battery Holders, Shooter Hood, Rubber Band Intake holder. During our season we learned plenty of new ways to use it efficiently. Specifically, we learned how to add screw holes into models to allow easier mounting to our robot, tolerances for mechanical parts like gears, and how different shapes impact the strength of the model- depending on the different forces applied to it.

## 12/31/25 INTAKE BAND HOLDER

For the Second Prototype to assist in a cleaner intake system and easier installation.



(Made using TinkerCad)

# COMPETITION

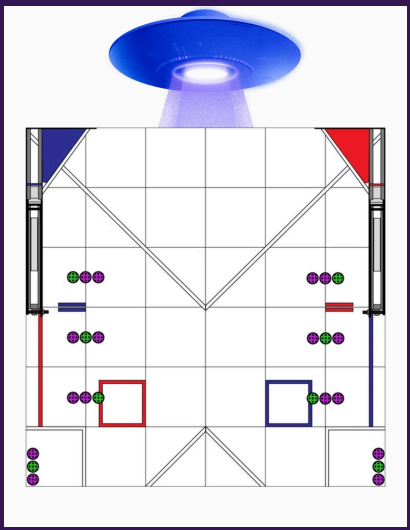
## Drive Team

- COACH: VALERIE
- 1 MOVEMENTS: DESTINY ⟷ AUSTIN
- 2 IN/OUTTAKE: GENESIS ⟷ DAXTON
- HUMAN: JOSIE

BACKUPS

We perform test runs with different members . Those who scored high scores and were most comfortable were chosen.

## Strategies






To plan with ally teams easily, we made a laminated sheet that can be marked over to demonstrate our auto movements along with useful questions.

WE REFLECT ON OUR MISTAKES AND FORMED PLANS BASED ON THEM.

- If a robot malfunctions but is still able to move. It shall resort to defense mode.
- Driver 2 will inform Driver 1 when there is three artifacts in the bot.
- When aligning at the depo, Driver 1 will inform Driver 2 when to launch.





