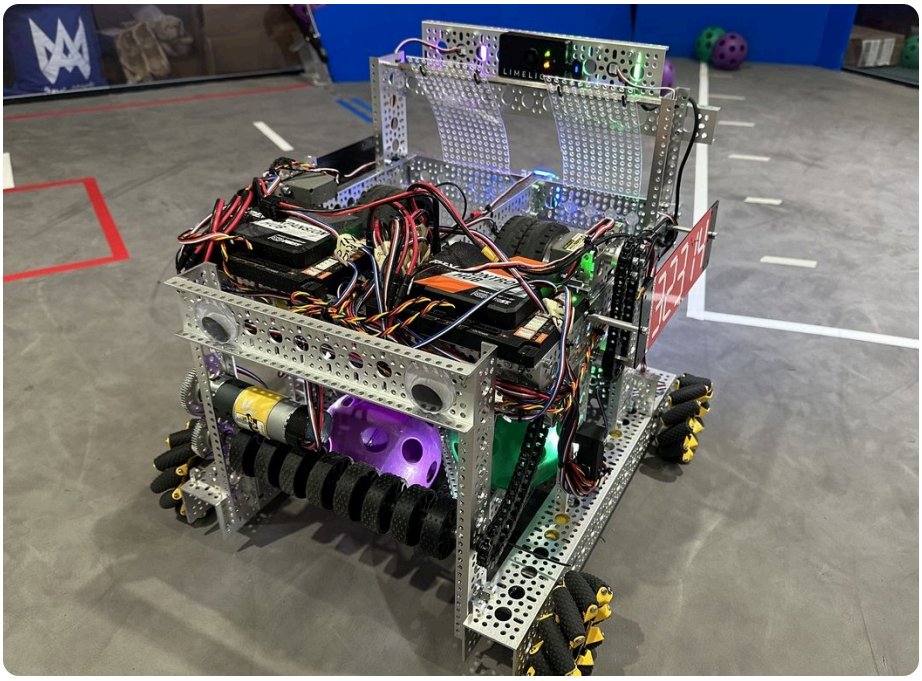
32314

Marcus Bartholomew the Third Senior

FIRST Tech Challenge — DECODE 2025–2026



Think Award / Story of Our Team and Season

Intro

We are a first-year team and as such we didn't know what we were doing. Some of our team members were on the FLL team Marcus Bartholomew the Third Junior, so we named our team Marcus Bartholomew the Third Senior. Because we are a first-year team our first robot was a GoBilda starter bot.

Starter Bot

We built the starter bot and began making improvements. One of the first things we noticed is that we could only shoot from right next to the goal which is not good. If you can only shoot from one spot you are very vulnerable to defense. To be able to shoot from further away we added a "flap" to the robot's shooter. The flap was a piece of flat plastic connected to a servo and increased the range in which we could shoot from by adjusting the angle of the shot.

We wanted our robot to be easier to drive and score with accurately. Also, from where the drivers stand it's difficult to judge the angle to the goal. One of the best ways to do this is to aim automatically. When the robot aims itself, we help remove driver error from the equation. To do this we added the Gobilda Pinpoint Odometry computer which tracks the robot's position using the data from two unpowered Gobilda 4 bar odometry pods. We also started working on having the robot automatically angle the flap and flywheel speed, but we stopped when we began building a new robot.

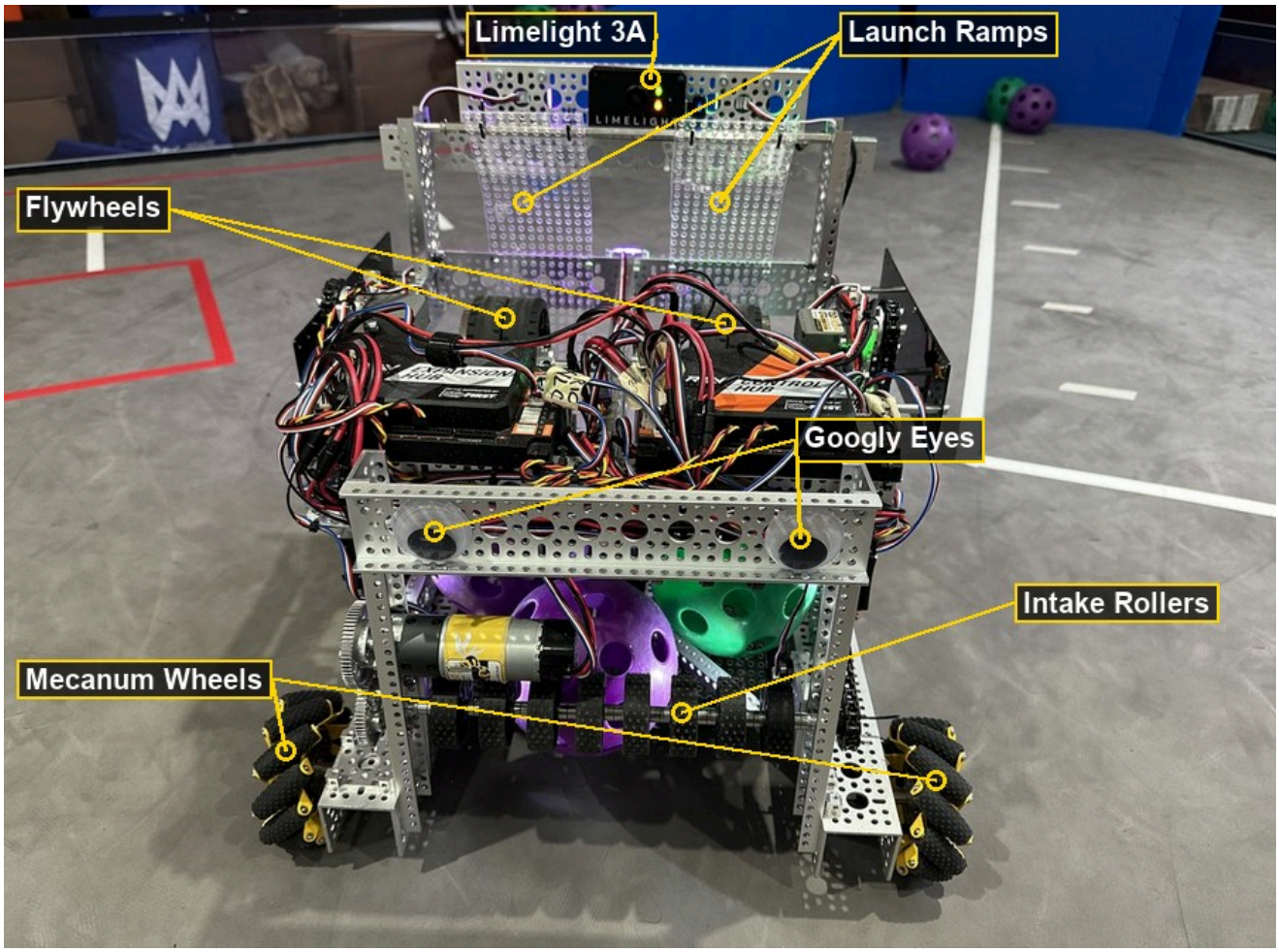
Another problem was that no matter where we put our power switch, artifacts could hit the power switch and turn the robot off. To solve this, we covered our power switch with a short U channel.

The starter bot was a great help in getting us experience with building a robot and beginning to understand what we wanted to do, However it had some limitations that we didn't like.

- It didn't have an intake
- It had a tank drive and not a mechanum drive

Because of this we decided to build the Gobilda Robot in 3 Days, or RI3D.

Robot in ~~3 Days~~ 6 Weeks



The Robot in 3 Days is what our current robot is based on. We wanted a few things from our next robot

- Intake so that we could have faster cycle times
- Mechanum Drive for maneuverability
- Far and Near Shooting so it's hard to play defense on us
- Be an existing design because we're a new team and don't have experience to do our own thing.

We decided that the RI3D met these conditions well and would be good for us. Based on the CAD model we built the RI3D. It was NOT build in three days for us.

The GoBilda RI3D features a dual flywheel setup with a diverter to chose which launcher the balls go to. This was designed so that it could do the pattern well. To intake three balls we have to intake one and then switch the diverter. We think that the diverter was not beneficial to us but overall, we believe the RI3D is perfect for us this year.

When we finished, we noticed a few issues with the design.

- We had built a few things wrong and needed to fix them
- Sometimes artifacts ended up in a position where the feeders couldn't get them

- Sometimes the diverter got stuck

To fix both the wobbly intake and Diverter getting stuck we added supports to the intake structure and plates to stop the diverter.

Another improvement was that we added a few guides so that the feeder could always reach the balls.

We then began thinking of other improvements we could make. One of the first things we did was taking the odometry from the starter bot and putting it in the RI3D and adding autoaim.

We wanted a few things

- More automation in the code
    - Automatically switching diverter
    - Automatically turning intake on and off
    - Automatically parking
    - AN AUTO
        - remembering where we are after the auto
- color/distance sensors so we know where the balls are and what they are
- Generally improve accuracy
- Lights on the back of the Robot so the driver could know where the artifacts are and how many we have

We also wanted PedroPathing to make our auto programming easier. PedroPathing is a robot control library that uses PID controllers to drive the robot better and allow you to drive to a specific point on the mat which makes autos a lot easier. We installed PedroPathing and began tuning the robot, we had a few issues understanding how tuning worked but we researched it and figured it out.

We planned on going to a scrimmage on March 1st so we had a deadline to get as many improvements done as possible.

One of the first things we did was use the color/ distance sensors to automatically switch the diverter. Besides Auto Aim this is the most impactful automation for the drivers.

We also began working on an auto program. We used a command system to program our autos as this makes it easier and quicker to make changes. We only have a single auto program, but we have it so that you can select different versions like a near or far in the init_loop().

Scrimmage

While we are based in Florida, we wanted to have practice in actual matches. Our driver and our coach went to the North Carolina scrimmage. The scrimmage was great for us! Our driver go a ton of experience and we were able to know what we had to do to improve our robot. During the scrimmage we ended up in second during the Qualification matches. Team 26520, the NC Avengers chose us for the playoffs and we won!

We learned some things from the scrimmage

- Our flywheels need to be off at the end of the match

- our intake needed to be sturdier, during the scrimmage the gears powering the intake disconnected multiple times.
- our intake was on too much. We took in to many artifacts and we got fouls for carrying more than three artifacts and sometimes shooting outside the launch zone.
- we should know when to stop feeders based on distance sensors
- we should have a single button to shoot all as we had to press a bunch of buttons to shoot three balls
- we should have a button to auto park the robot as parking is hard
- we should add a far auto because many times our alliance partners couldn't do their auto because of ours.
- Odometry is very susceptible to small angle changes in starting position. During the first Playoff our auto aim was completely off because of this and we did not score much.

After Scrimmage

- We rebuilt the intake frame to have it be sturdier by making the frame taller and adding a cross beam and we added Loctite to the gear screws. This helped the intake be more stable and not fall apart. The crossbeam is also perfect for the googly eyes which are extremely important!
- To solve the flywheels being on and the intake being on we added some logic to our code to turn them off when not needed.
- We made it so that when the color sensors detected that a ball left then the feeders would stop. When we tell the robot to shoot it will shoot.
- We added some logic to the code so that when Y is pressed all balls are launched.

Lessons learned/ if we had more time

- We learned that automation in teleop is important
- We learned that critical screws should be locktited.
- Odometry is great, but it is susceptible to small changes in starting position.
- The diverter is not the greatest as it increases our cycle times because we have to wait for it to flip when intaking and shooting.
- If we had a whole season to redo, we would likely choose a different robot design the RI3D was great for our first year but has some limitations like the diverter.

Goal Tracking:

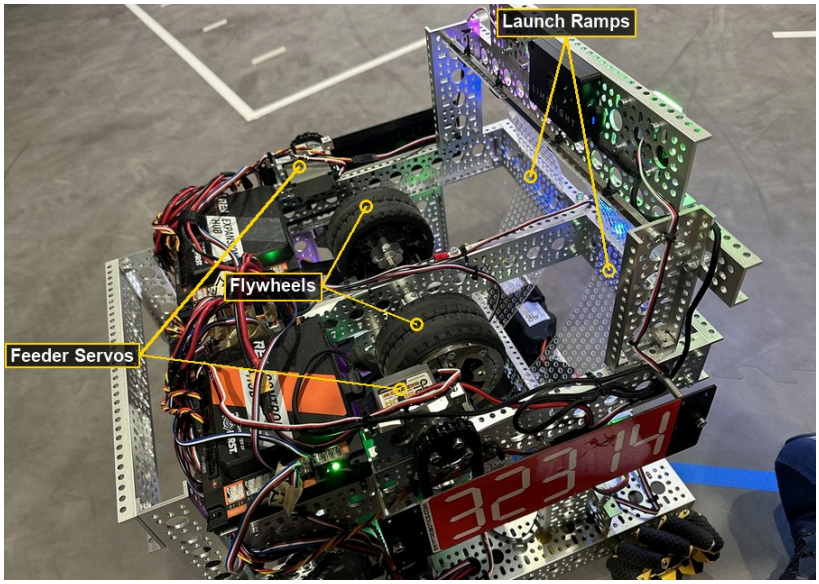
To track goals, we had checkboxes on our engineering log. We also used GitHub issues a little.

See Also

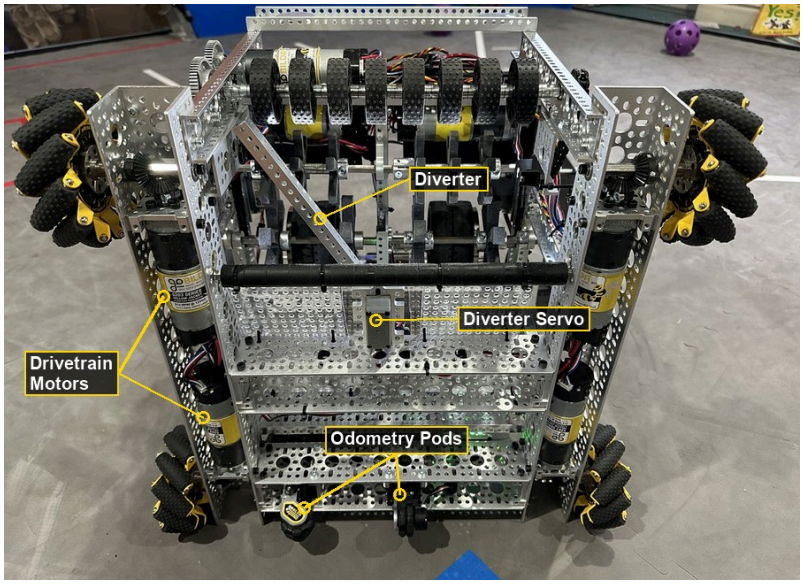
- For information on how we develop skills see the Connect Award
- For more detail on robot systems see the Control Award
- For how we recruit people into FIRST look at look at Reach Award

Control Award

We have many control systems on our robot with many of them are interlinked and can be split in many different ways.



Mechanum drive / PedroPathing / Odometry

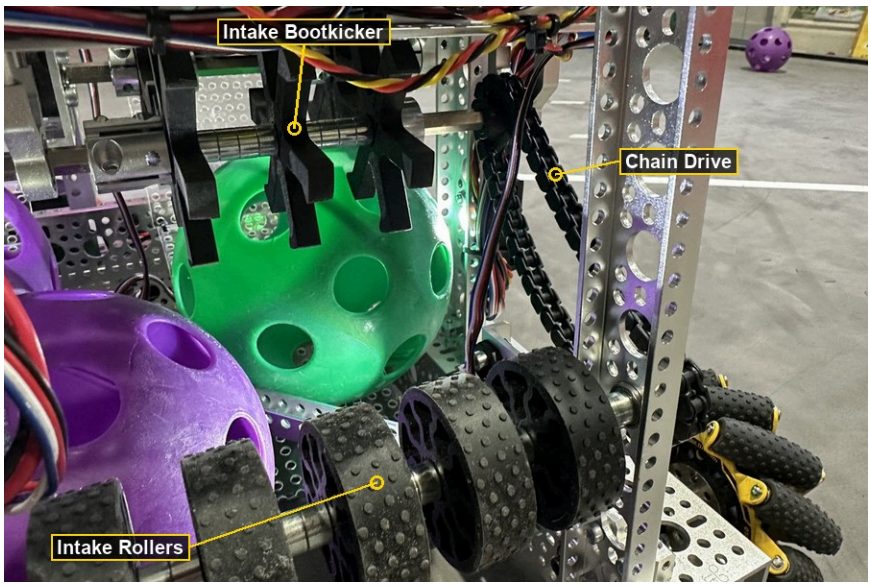


Challenge: A tank drive is not as maneuverable as a mechanum drive. In a tank drive it's a lot harder to correct small errors and you are more susceptible to defense. A tank drive also is harder to position for parking or hitting the gate. Additionally knowing where your robot is is helpful for many system. Additionally mechanum drive allows us to use field relative control for driving which is easier on the driver

Solution: We used a mechanum drive with PedroPathing. Pedropathing is a library that lets you drive your robot better using PID controllers

How it Works: Pedro pathing is a pathing library that lets you tell the robot where you want it to go and it will drive there quickly and accurately. The odometry uses data from an IMU and two unpowered wheels to figure out where the robot is. Field relative drive uses polar coordinates and the IMU to transform the control inputs to the right direction.

Intake

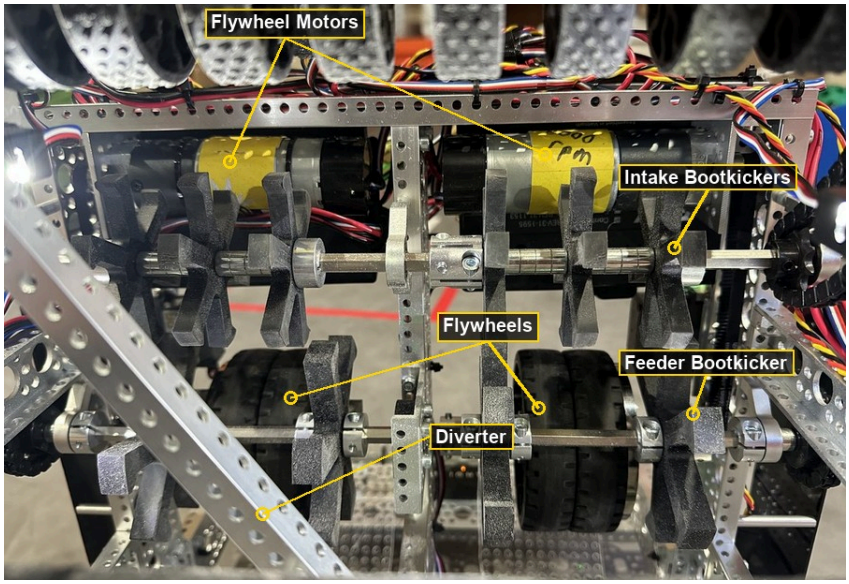


Challenge: The Starter Bot couldn't intake artifacts. This led to slow cycle times

Solution: The RI3D has an intake which helps us have faster cycle times. It also allows us to score better while playing defense.

How it Works: Our intake is a two-stage system. The first stage is what initially grabs the artifacts and gets them inside the robot. Then the second stage with boot wheels takes the artifacts to the feeder larger boot wheels. Both stages are powered by a single motor. There is also a beam on a servo called the diverter that sends the balls to either the left or right launcher.

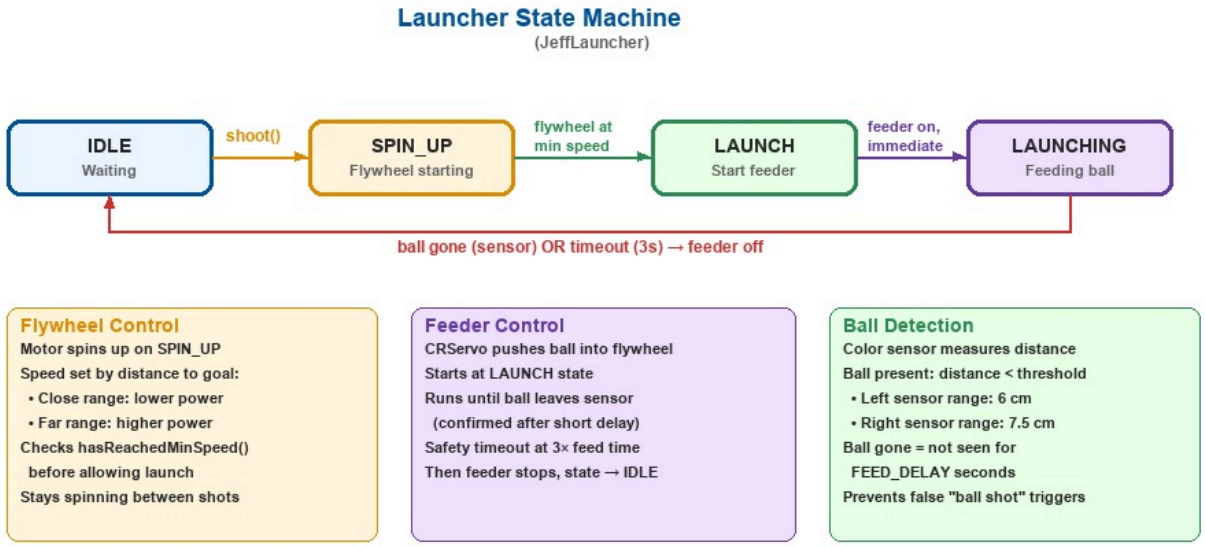
Dual Shooter



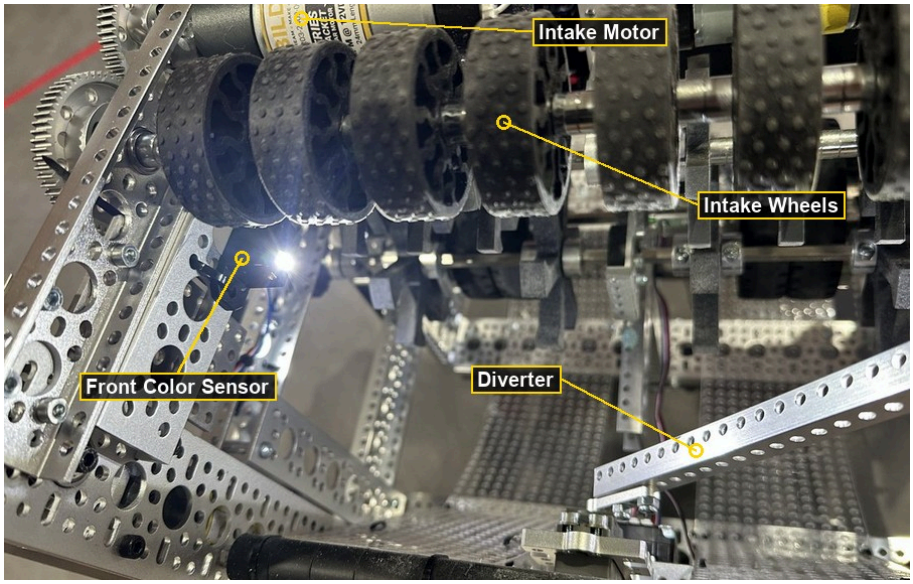
Challenge: The starter bot does not have the option to sort the artifacts by color.

Solution: The RI3D solves this by having two flywheels to be able to sort the artifacts.

How it Works: The previously mentioned diverter can choose which shooter the artifact goes in. Then when we want to shoot the feeder servos for each side power some boot wheels to feed the artifact into the flywheel. There is a state machine that keeps track of what state of shooting we're in. First it spins up the flywheel. Then it starts the feeders once the flywheel is up to speed. Once the color sensors detect the bal has been gone or the 3 second timer runs out it stops the feeders



Color / Distance Sensors

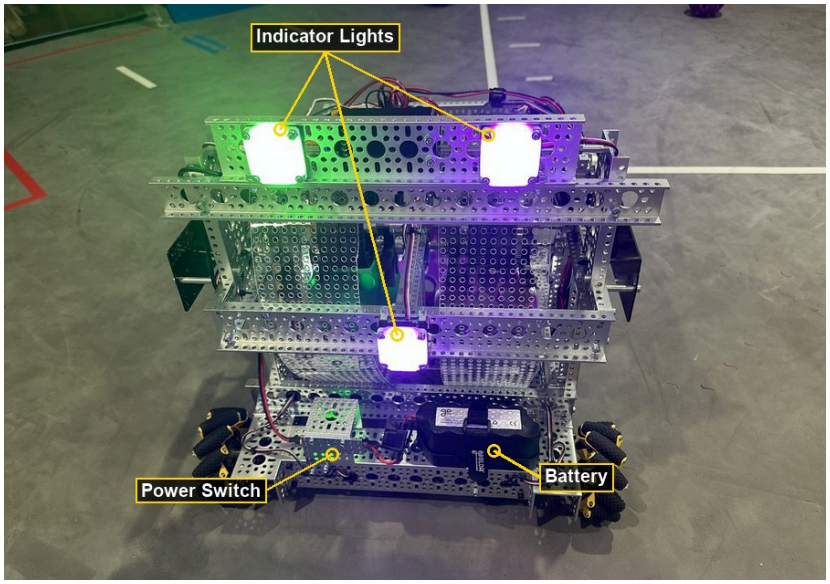


Challenge: We want to know where the artifacts are as its useful for many things. This helps solve lots of other challenges, for example it can help make the launcher more reliable by detecting when an artifact has been fed into the launcher.

Solution: We installed 4 Rev Color Sensor V3s on our robot, two in the back and two by the intake so we can detect every possible state. We also realized that if the color sensors are too close to an artifact, they can't tell the color. We 3D printed some mounts to space the color sensors properly.

How it Works: The color sensors can also detect distance so we can know if an artifact is there. This is then used in the code to detect when an artifact has been successfully fed. This also gives data to many other systems like the lights and many driver assistance functions.

Indicator Lights



Challenge: The robot is not transparent. It is helpful for the driver to know where and how many artifacts there are on the robot so they know if they should shoot or intake more artifacts.

Solution: We have three lights on the back of the robot that show what artifacts are where increasing the driver's situational awareness using the outputs from the color sensors.

How it Works: The lights work like this: When there is no artifact it will display the alliance color. When the color sensors detect an artifact then it will show the color of the artifact.

Limelight Camera

Challenge: Having odometry on the robot is great but it drifts and is susceptible to small changes in the starting angle.

Solution: Using a camera to track April Tags is more accurate as the April Tags don't move

How it Works: The limelight processes its video feed and outputs data about the April Tag's relative angle in the code.

Auto Command System

Challenge: It is hard to write autos quickly and easily.

Solution: We use a command system which makes programming autos easier

How it Works: We have different command types that are based on a base command. Each command type has something to do each loop. There is also a condition for it to be done. We have commands for shooting all the balls, running any function we want, among others. We also have a CommandCommand base class that is a command that runs commands. We use CommandCommand to avoid repeating a lot of code in the command system.

This is all we need to write a near auto code:

```
if (auto == Auto.NEAR) {
    scheduler.schedule(new SequentialCommand(
        new LineToCommand(follower, AimAt(nearScorePose, goalTarget)),
        new ShootAllCommand(scoring, vision),
        new YummyArtifacts(scoring, follower,  spike: 1.0),
        new LineToCommand(follower, AimAt(nearScorePose, goalTarget)),
        new ShootAllCommand(scoring, vision),
        new YummyArtifacts(scoring, follower,  spike: 2.0),
        new LineToCommand(follower, AimAt(nearScorePose, goalTarget)),
        new ShootAllCommand(scoring, vision),
        new LineToCommand(follower, endPose)
    ));
}
```

This is a CommandCommand for intaking artifacts. It avoids repetition in the code.

```
// YummyArtifact Command code: Intake all artifacts from a spike mark
addCommand( new InstantCommand(() -> scoring.intakeOn()));
addCommand(new LineToCommand(follower, getSpikePose(spike,  ball: -1)));
addCommand(new LineToCommand(follower, getSpikePose(spike,  ball: 1),
        maxPower: 0.5,  holdEnd: true));
addCommand( new WaitCommand( seconds: .25));
addCommand( new InstantCommand(() -> scoring.switchDiverter()));
addCommand( new WaitCommand( seconds: .25));
addCommand( new LineToCommand(follower, getSpikePose(spike,  ball: 3),
        maxPower: 0.25,  holdEnd: true));
addCommand( new InstantCommand(() -> scoring.intakeOff()));
```

I2C Read Rotation

Challenge: Our robot had really slow loop times which negatively affected PedroPathing by making it overshoot and they made our robot not drive so well. We used bulk Reads to help but they don't read I2C for some reason.

Solution: We have a queue for I2C reads so we only do one per loop.

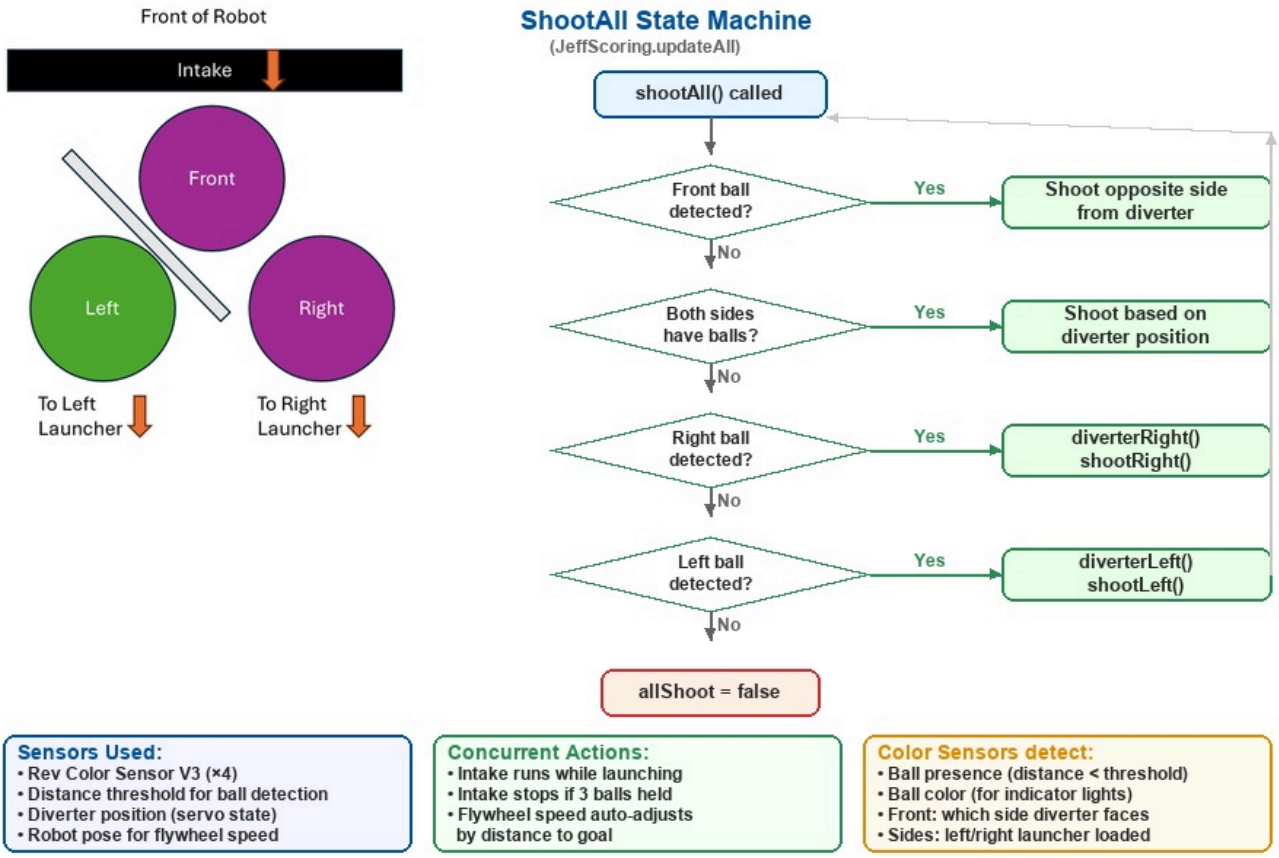
How it Works: Every time a read is requested it returns the last know value. It cycles through the list of sensors it has for reading a value. On the first time a value is requested it won't be put on the queue and will be read immediately so we don't return null values.

Driver assistance

We automated many driver tasks during teleop. Automating these task lets the driver focus on shooting and driving instead of minor tasks and reduces cycle time. It also reduces stress on the driver and reduces human error and fouls. The computer is also faster and quicker

- **Auto Aim** Auto Aim is essential as it's hard to align the robot manually to the goal. Using the data from the odometry we can use the atan2 function to find which direction to point the robot in the right direction. We tested out many small changes to where it aims during the season.
- **Auto Diverter** We automatically switch the diverter using the distance sensors because when intaking balls you need to intake one and the switch the diverter if you want to carry three balls.
- **Location Based Flywheels** To conserve battery and save time we automatically spin up the flywheels when in the launch zones and turn it off when we leave using data from odometry.
- **Automated Intake** We were having many fouls by intaking more artifacts than we should as the rules state you can only carry three artifacts but sometimes our robot took in four. Intaking extra artifacts also could lead to artifacts launching outside the launch zone resulting in more fouls. We made it so that when we detected three artifacts the intake would automatically turn off. It then turns on when the robot can carry more artifacts.
- **Shoot All** To shoot reliably our robot needs an artifact behind the artifact being shot or the diverter behind the artifact. Instead of having to pay attention where the balls and diverter are

we have a *shoot all* button that detects where the artifacts and diverter are and then shoots them and switches the diverter when needed using the data from the color/distance sensors which saves a lot of thinking and button presses.



Connect Award

Lesly

Skills focused on:

- Programming
- Building

Her goal is to improve these skills. She practices regularly, works with our coach, and follows instructions. She am steadily improving and becoming a better team member.

Lucas

Skills focused on:

- Programming
- Building

- Driving
- Design 3D

He practices regularly and contributes to improving the robot's performance. He's also our main programmer and takes 3D design elective in school. He designed the color sensor mounts and also 3D printed and launched model rockets for fun

Matthew

Skills focused on:

- Video editing
- creating content

  He is improving through practice and feedback

Josephine:

Skills focused on:

- Driving
- Building
- Programming
- Outreach

Her goal is to improve these skills. She comes to meetings regularly and works closely with our coach and my teammates.

Caleb:

Skills focused on:

- Public speaking
- Driving
- Being the human player in the robot game

He regularly attends meeting and practices with our team.

Reach Award

Our team's outreach focuses on growing the FIRST community by raising awareness, recruiting new members, and making robotics visible in our local area.

Outreach Goals

We designed our outreach around four main goals:

- Spread awareness about FIRST and robotics opportunities
- Encourage community support through social media and donations
- Recruit new students, mentors, and volunteers

- Inspire others to join or start FIRST teams

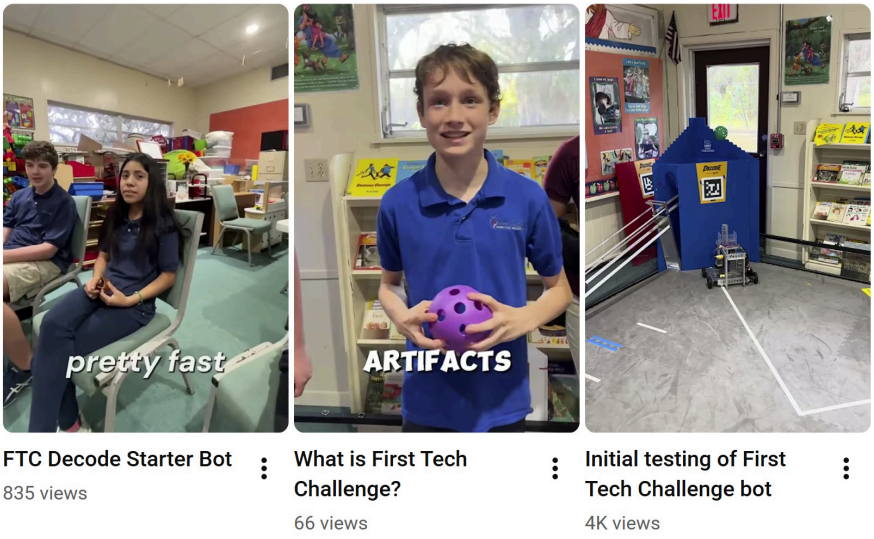
At our Fall Festival:

- Our Pathfinder club sold Pizza to raise money and visitors learned about our team
- We Engaged directly with families and students interested in joining
- This allowed us to connect with people who had never been exposed to FIRST before.

Community Presentations

- We promoted FIRST through church outreach, reaching a broad local audience:
- We created and distributed custom bulletin inserts with our team and FIRST information
- We Delivered live announcements at multiple locations
- These efforts helped us reach our entire congregation expanding awareness beyond typical STEM audiences.

Digital Outreach



FTC Decode Starter Bot
835 views

What is First Tech Challenge?
66 views

Initial testing of First Tech Challenge bot
4K views

We created and shared videos promoting our team and FIRST, which:

- Achieved close to 5,000 views with increased engagement
- We demonstrated our starter bot, showing each part about it and how it maneuvered and worked
- We demonstrated gameplay on the field and explained how the tournament operates
- We Made robotics more accessible and exciting to a wider audience
- And it provided a useful tool for outreach and recruitment

Online Presence

Our team supports outreach through the WCCA Robotics website,(http://wccarobotics.github.io/) which:

- Serves as a central hub for information about our teams

- Helps interested students and families learn how to get involved
- Reinforces and extends the impact of our in-person outreach

Recruitment & Growth Impact

Our outreach has directly contributed to bringing new people into FIRST:

- Our team demonstrates recruitment impact with a first-time co-coach and a student who had never participated in anything like FIRST before
- We continued growing by recruiting an additional team member mid-season
- We supported FIRST LEGO League teams, including one made up entirely of new participants
- We are actively working to recruit new mentors, including local educators
- These efforts show our commitment not just to participation—but to expanding the FIRST community.

Sustain Award

Financial Sustainability

The WCCA Robotics program operates under an "umbrella" model that ensures sustainability across all three teams (2 FLL and 1 FTC). By diversifying income streams and careful spending, we ensure that all teams have the tools to succeed.

Diverse Income Streams

Our financial wellbeing relies on a mix of corporate matching, community support, and student initiative.

Corporate Matching: A cornerstone of our funding is the Microsoft Volunteer Program. Microsoft contributes $25.00 per hour for every hour our coach volunteers.

Community Outreach: We maintain strong relationships with our "sister churches," whose donations provide additional funds when needed.

Student-Led Initiatives: Our teams take active ownership of their needs, organizing community fundraising events to bridge gaps for special opportunities. The Florida Sunshine Invitational was a good example of this. When the FLL team was invited to this tournament, they faced a significant financial hurdle.

The initial budget estimate was $15,000. Later, a more accurate estimate was $9,000.

Through church donations and local fundraising events, the students successfully met their target goal. A total of $14,000 in donations was given, but with careful spending they only spent $8,400.

Impact: This didn’t just fund a trip; it served as a lesson in teamwork, public relations, and financial responsibility. The surplus funds remain within the school to support future seasons.

Start up costs

Us being a first year team, a large sum of money was spent in getting supplies. This includes our playing field, two robot starter kits, and additional hardware for our robot. We spent around $5,000 on these startup costs, and around $1,000 for annual costs like registration, tournament costs, shirts, etc.

Going Forward: Our Plan For Success

Now that we are out of the startup phase, we can work on making our team better and refining our skills. Next years estimate is roughly $1000 in recurring annual costs and $1000 in new "fixed cost" investments.

Sustainability & Future Growth

How we are sure our team will last after the current members are gone?

Our FTC team started with students from FLL. Although some members dropped out becuase of school stress, we recruited our new team member Josephine, and our Jr. member Caleb. Caleb is in 8th grade and still participates in FLL, but is excited to be a full member of our FTC team next year.

We are committed to the long-term continuity of our team well beyond the graduation of current members. To ensure this, we maintain a direct recruitment program with our FLL teams.

By actively mentoring these students that are already a part of our robotics program, we guarantee a consistent influx of experienced members ready to start FTC roles each year.

Portfolio created by Team 32314 with assistance from GitHub Copilot (Claude) and Google Gemini. AI was used for portfolio layout, photo annotation, diagram generation, and writing assistance.