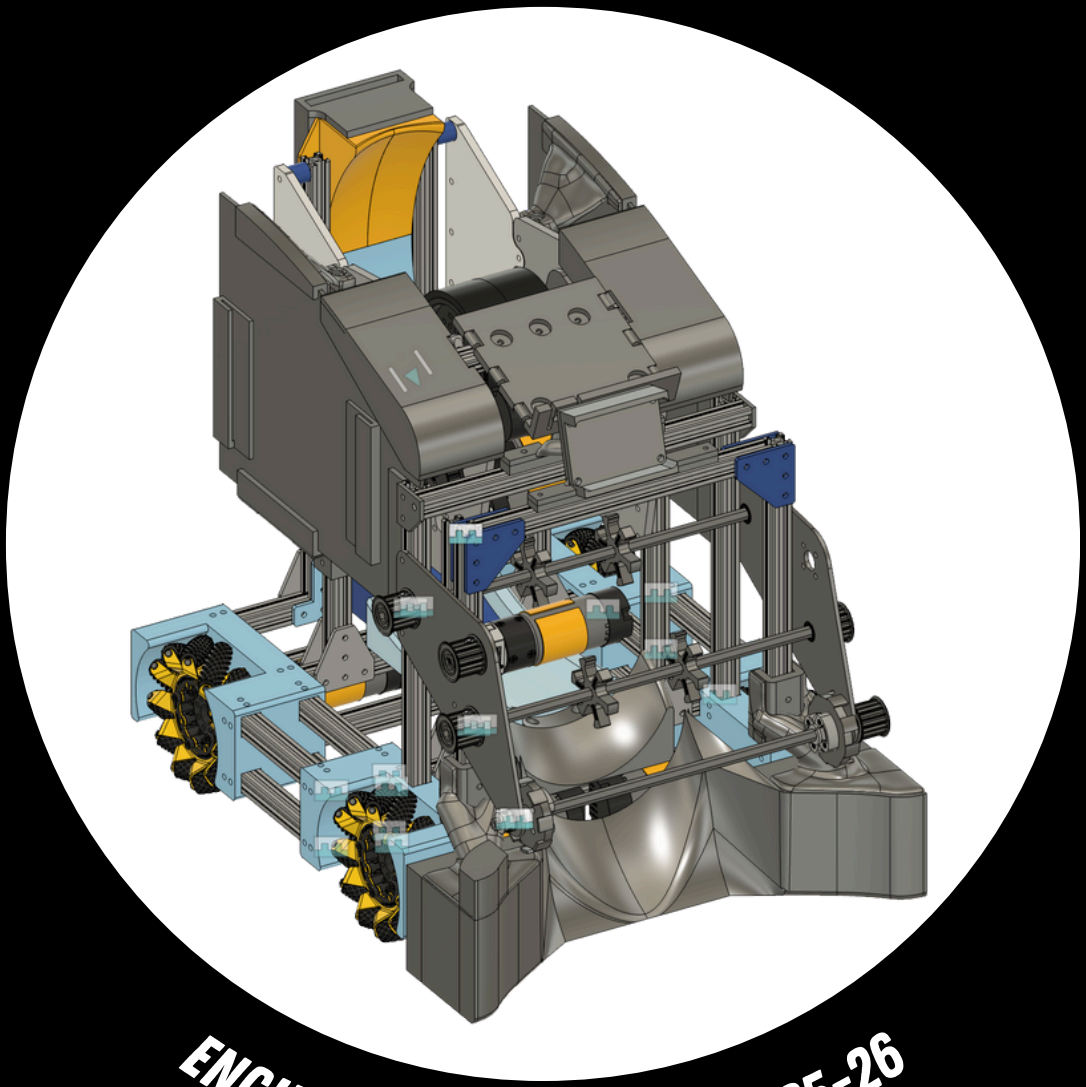
#27628



ENGINEERING PORTFOLIO 2025-26

BEARGINEERS

"Building Robots with a roa-a-ar!"

BEARGINEERS TEAM

Operations

Yasha

Grade 11 (Junior)

- *H*elps out in all teams
- Helps organise and delegate tasks.

Software

Nina

Grade 12 (Senior)

- *S*econd year as Software Lead
- Can program in Python, Kotlin, HTML, PHP, SQL.

Yalin

Grade 10 (Junior)

Nicolas

Grade 11 (Senior)

Hardware & Design

Mirgali

Grade 12 (Senior)

- *S*econd year as Hardware &
- Design Lead
- Specialises in mechanical and electrical engineering

Josh

Grade 12 (Senior)

- Second year as Hardware &
- Design Lead
- Specialises in mechanical engineering and 3D design.

Terentiy

Grade 12 (Senior)

Celine

Grade 11 (Junior)

Leo

Grade 10 (Sophomore)

Marketing

Team lead: Konrad

Grade 12 (Senior)

- *S*econd year as Marketing Lead

Tiffany

Grade 10 (Sophomore)

Helena

Grade 10 (Sophomore)

Freddy

Grade 12 (Senior)

Mentors

Ford Watson

One of our mentors is a mathematics teacher at the International School of Amsterdam, who supports the team with problem-solving strategies and logical reasoning.

Maxim Shafirov

Our second mentor is a programmer who teaches us software development, sensor integration, and key hardware concepts, helping us connect code with real-world robot behavior.

GAME STRATEGY

As we have analysed the game as the season went on, we noticed that there were two important things to get a balance between, and that there would be trade-offs between them:

Game points

Ranking points

1) Pattern ranking point:

As the pattern points are only awarded at the end of autonomous and at the end of the game, a spindexer that would limit our ability to have a high throughput for the rest of the game, so we have decided to compromise on those ranking points in favor of more game points.

2) Movement ranking point:

For the reason that to get a ranking point in the qualifier, it is enough for both robots to leave the shooting zone in auto, and to have one be partially parked, and for the other to be fully parked. And the lifting mechanism would take a lot of time to rise, because we have a heavy robot, so we have decided to leave it for now, and maybe do something for the Benelux championship.

3) Amount of ball ranking points:

As we have made a sacrifice on not having any indexing, we are hoping to get a very low cycling time, allowing us to score a lot of artifacts.

Being flexible

As a general rule, it is good to be flexible in any FTC game, as you get to play with a variety of different teams, all capable of doing different things.

1) Being able to shoot from far and close:

As some teams may not be able to shoot from far or from close, which is less common, but still happens, it is a great advantage to be able to compensate for that and shoot anywhere

2) Being a tiny bit smaller than the parking zone

As our robot is a little narrower than 18 inches, it allows us to fully park, while the other robot can partially park, giving us both more game points, as well as making it easier to get a ranking point

Playing defense

Our robot is built in a way to be able to handle defense with a durable structure, allowing it to play defense itself

DESIGN BRAINSTORM

Idea

A front-mounted ramp intake designed to quickly guide artifacts into the robot while minimizing jams and driver precision. The angled ramp directs balls upward into a short roller system, allowing a smooth transition from the floor to the feeder.

Design Rationale:

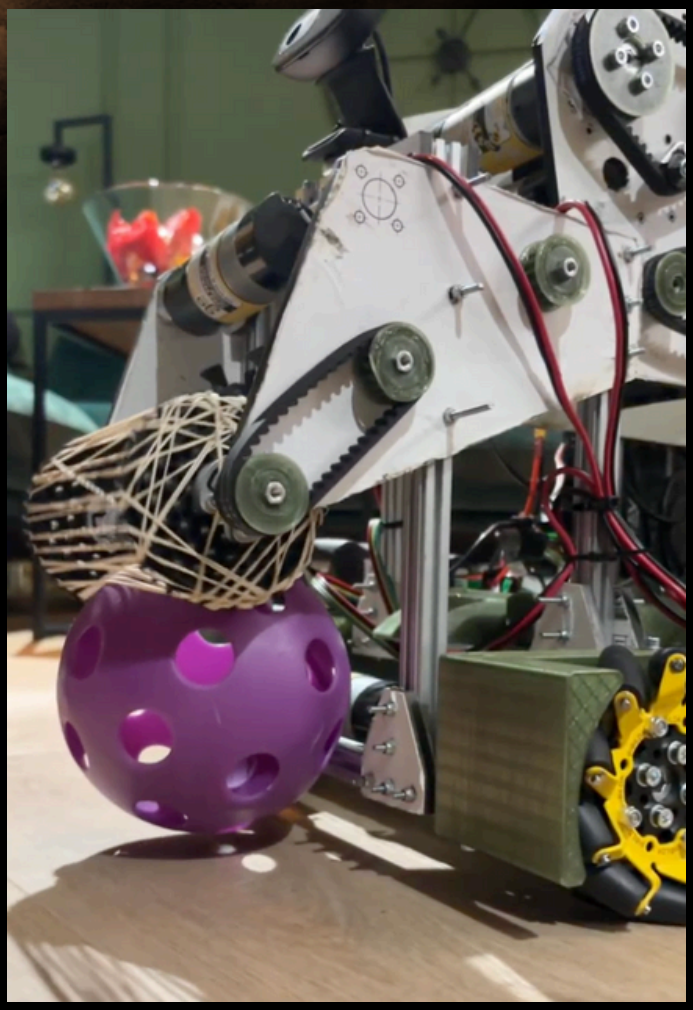
We considered fixed rollers and dual-stage conveyors, but a ramp intake provided the simplest, most reliable path for fast cycles. By using a gentle incline and slightly compressive roller, artifacts are guided into the robot automatically, even if approached at an angle.

Advantages:

- Fast, forgiving ground pickup, easy to control
- Reduced jam risk due to smooth ramp path
- Easy integration with feeder and shooter systems

Outcome:

Chosen for its reliability and seamless integration with the shooting strategy.



Previous robot without ramp

Idea:

Optimize for rapid intake from quick shoot cycles, relying on speed and software automation.

Advantages:

- Very fast cycle time
- Simple, reliable ball path
- Better compatibility with automated aiming
- Lower mechanical failure risk

Limitations:

- Less control over artifact order
- Pattern ranking point is less consistent

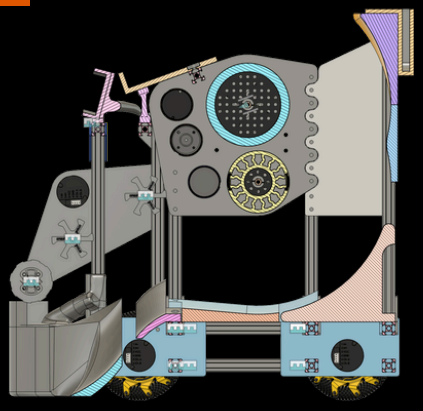
Outcome:

Selected. This concept best aligns with our goal of maximizing game points while remaining flexible with alliance partners.

Ball storage we deemed most efficient

DESIGN PROCESS

Design Process: Iteration After Scrimmage



After our first scrimmage, the robot performed well and validated our fast cycling and shooting strategy. However, match conditions revealed reliability issues, particularly with intake consistency, artifact containment, and positioning accuracy. These observations guided focused improvements.

Design Improvements & Iteration

Ramp Intake Redesign

- Our initial ramp was a simple triangular shape and not specifically fitted to the artifact, which caused occasional bouncing and loss during fast intakes.
- We redesigned the ramp to match the artifact's dimensions, adjusting width, angle, and side walls to guide artifacts naturally into the robot.
- This improved intake reliability, especially at angled approaches.

Widened Intake Area:

- Increased the width of the intake zone to contain artifacts better.
- Reduced artifact loss during aggressive driving and fast transitions.

Manufacturing Improvements

- Transitioned from hand-cut parts to laser-cut components.
- Improved precision, alignment, and ensures better structural integrity

Sensor Upgrades

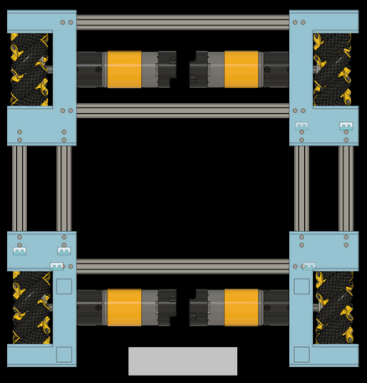
- Added dead wheels for accurate odometry independent of wheel slip.
- Integrated a camera for automated alignment and distance-based shooter control.

CHASSIS DESIGN

Drivetrain – 4-Motor Mecanum

Goal:

Precise alignment and short cycle time, holonomic motion for repositioning without turning.



Findings After Test Tank Drive

Metric	Tank	Mecanum
Time to align to goal	Requires turn correction	Strafe reduces correction time
Defense resistance	Higher	Lower
Shot consistency	Good once aligned	Higher due to micro-adjustments

Design:

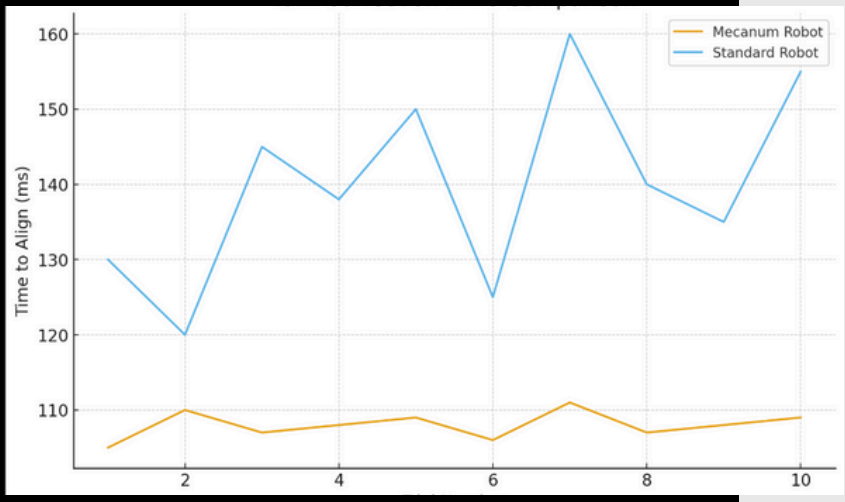
- 96 mm mecanum wheels (normal X-pattern).
- 1:1 drive with a close wheelbase to reduce rotational inertia.
- Center mass kept low and central to minimize pitch during acceleration/shooting.

Rationale & Tradeoffs:

- Chosen over traction/tank to reduce aim time, strafing reduces lateral correction cycles.
- Accepted cost: lower pushing power, mitigated with driver practice.

Drivetrain motors:

We chose to use four motors for the drivetrain to enable full mecanum movement, including side-to-side strafing. Although this utilized a large portion of our available motors, the significant improvement in driving efficiency and alignment justified the decision.



The graph shows Turn Correction Time (ms) across 10 trials, comparing:

- **Mecanum drivetrain** → lower and more consistent alignment times
- **Standard drivetrain** → higher and more variable alignment times

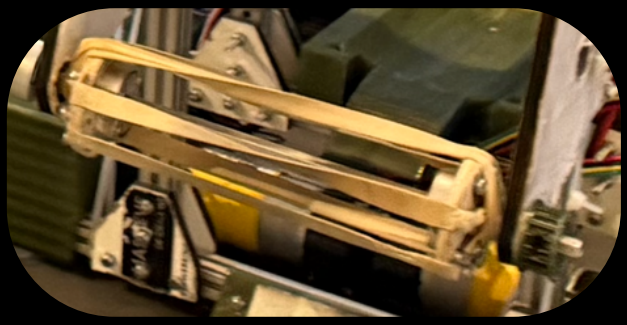
INTAKE SYSTEM

Goal

Fast ground pickup with minimal driver precision; positive ball control into the robot.

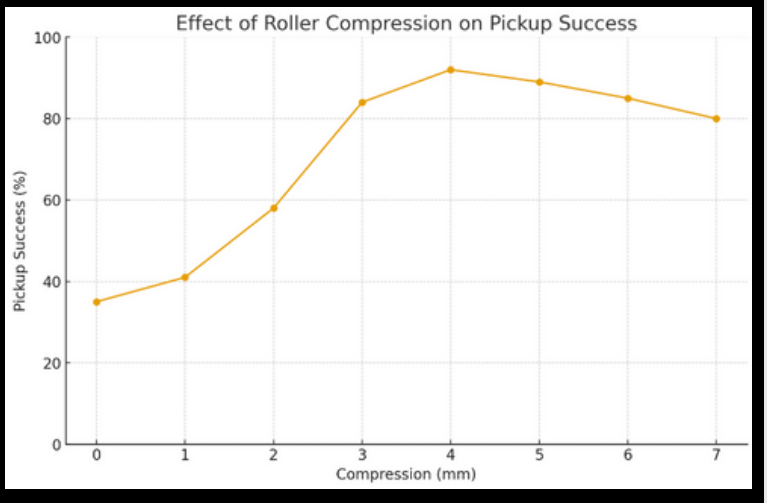
Design

Rubber bands are rotating and moving the balls using friction. Because of them moving so fast, we were able to make the artifacts go in efficiently, with rubber bands having some pull, and not get them stuck because of over compression. A feeder wheel in the middle to push balls to the shooter, having a sensors to detect when we have artifacts in the robot, to shoot more efficient use of aoutonomous time.



Rubber inatke

The intake relies on the friction between the rubber bands and the ball. A 3D-printed part is used to hold the rubber bands and maintain tension, ensuring consistent compression and contact. Below is a graph showing the testing of compression VS. pickup success.



Ramp

A 3D-printed ramp was added to the front of the robot to guide balls smoothly into the intake. The ramp reduces sensitivity to pickup angle, allowing the intake to engage balls reliably even when they are misaligned.

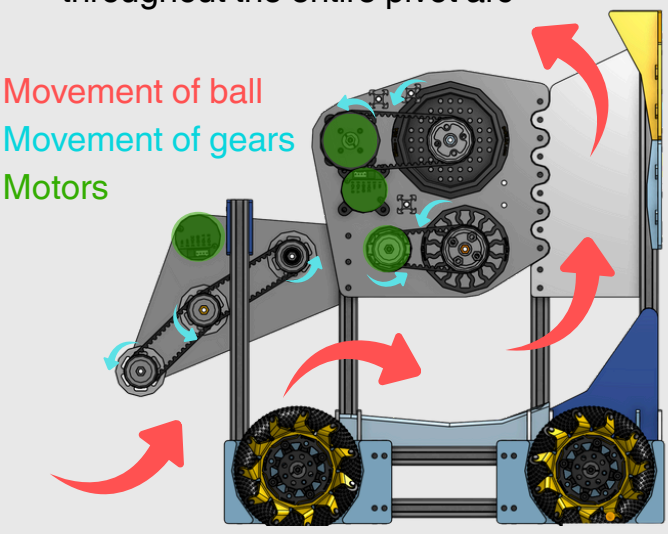
This design improves ease of pickup during tele-op and increases robustness in autonomous mode, where precise alignment is harder to achieve. By passively correcting ball position, the ramp saves cycle time and reduces failed pickup attempts without adding mechanical complexity.

Belt Path

The belt path was designed to provide smooth power transmission while minimizing frictional losses and belt wear.

Key considerations:

- Maintaining a consistent belt wrap around pulleys to prevent skipping
- Avoiding sharp bends that increase belt fatigue
- Ensuring the belt remains fully tensioned throughout the entire pivot arc

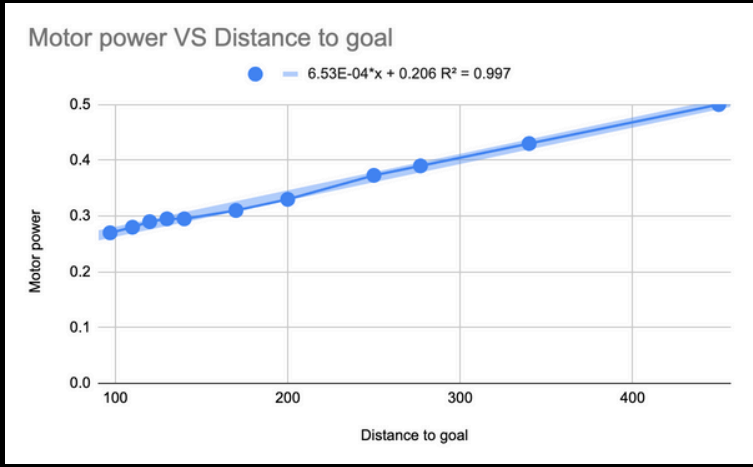


SHOOTER

Code:

Optimal flywheel speed tuning:

To calculate the speed for the flywheel on which the flywheel will shoot into the goal, we first made a table where we had 10 points of distance, and figured out the best flywheel speed using trial and error, and then, in an Excel spreadsheet, made a linear formula to approximate it.



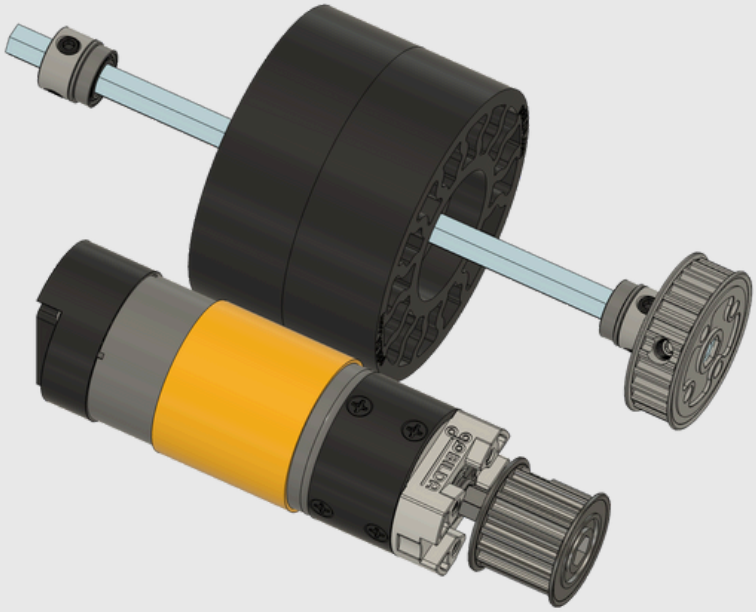
PIDF control:

To have the flywheel accelerate to the optimal speed in the shortest time possible, we have used the PDF parts of the PIDF control, where P stands for proportional, I stands for integral, D stands for derivative, and stands for fixed. We have not included integral, as it accumulates over time, and as our desired flywheel speed always changes, depending on the distance to the goal, it would accumulate on multiple goals, and would not be accurate.

Robot design:

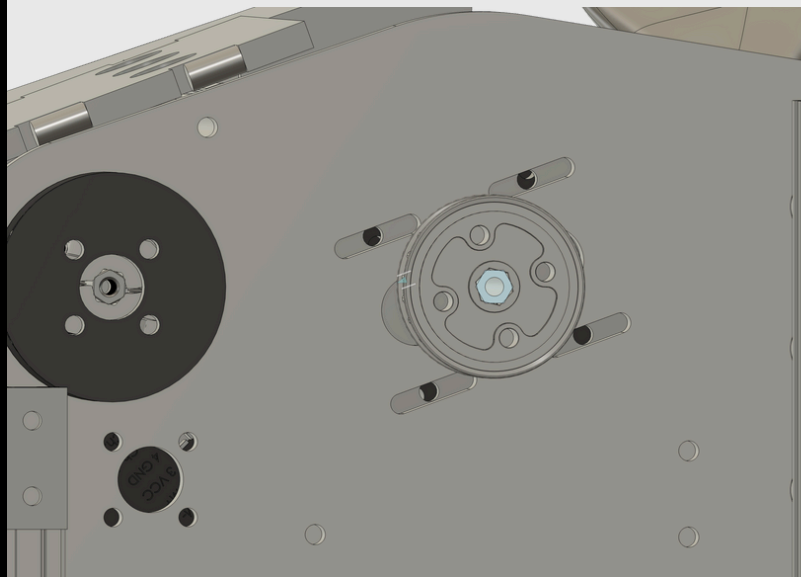
Feeder:

For the 3 artifacts to be able to reliably and fast into the shooter, we have used a grippy wheel driven by a 1000 RPM motor, for very rapid shooting.



Adjustable pressure:

With the previous generation of our robot, we have encountered that it has a lot of backspin, and a lot of the time, bounces back from the goal. We have implemented a part, so we can adjust how much pressure the flywheel puts on the ball, to get the balance between speed and spin.



CODE

Why did we not use existing libraries for programming?

- We have gained more inside on how the software works, and what is possible and what is not
- Easier to implement custom features, as we better know the structure and how to use the code
- We have gained very important experience and practice in how to code
- Custom things that pedropathing has not made!

Managing different robot interations

PROBLEM:

As we had a few iterations of the robot, including a straffer from Gobilda as our test stand for testing functions such as driving to target, we needed a way to not write the same code every time, and for changes that were made while developing one robot to be usable by another robot.

SOLUTION:

We made an entity for each robot that stored all the robot's parameters, such as its dimensions or tuning parameters, as well as real-time data, such as its position on the field. Then the main code, which is the same, manages all the actions the robot should do, and reads the control and sensor inputs.

AUTONOMOUS:

AUTO PHASES

All stages are executed in an autonomous manner and are organized into phases. These phases can be written in code as a sequence and are overseen by the main loop, which determines the necessary actions and executes them accordingly. That lets the robot do several things in parallel, like for example going to the shooting zone, and adjusting the flywheel speed to the distance to the goal.

FOLLOW PATH

The robot uses a function we made called "followPath" that takes in a list of positions on the field, and drives to them in that order. To achieve this, the code calculates the vector of movement, and calculates the relative powers of the motors for the robot to move there, and then the PD (proportional derivative) control is applied to make the robot reach its destination smoothly and without overshoot

TELEOP:

This season, our main objective was to make Teleop very automated, having automated goal aiming, shooter speed adjustment, and going to the shooting zones.

Using the auto phases, we were able to do multiple actions at once during auto, as well as teleop. Examples are: adjusting the flywheel speed, and controlling the robot, as well as using automated features outlined on the left ------->

Using the follow path function in teleop, we were able to mount it to a button, which, when pressed, would move the robot to the position we want, for example, the closest point in the shooting zone, and then shoot automatically when the flywheel speed, the heading, and the positions were correct.

AUTONOMOUS STRATEGY

As autonomous is half the opportunity to get pattern ranking points, it is important to utilize it. Here is the logic we have used:

1) Preloaded artifacts:

We always load our artifacts in the same order, so if the order matches the motif by chance, the robot will know it, and will be able to shoot the initial load, and not open the ramp, so it counts towards the pattern ranking point

2) Order of picking up spikes:

If the initial load happed to be not the same as the motief, the spikes are always the same, and there is at least one spike that matches the motief, this way we first shoot the initial load, open the ramp, and then firstly pick up the right spike, to have the highest chance of it landing when the amount of artifacts in the ramp is divisible by 3(so the spike is right).

3) Ending the autonomous:

At the end of the autonomous, we always make sure that there is enough time for the robot to leave the shooting zone, as well as drive close to opening the ramp, so that when the teleop starts, it is easy for the driver to open the ramp, without taking too much time.

Being flexible

1) Autonomous maker

We have implemented a special autonomous mode that allows you to compose a new autonomous mode from the initialising menu of the robot, consisting of building blocks, like “open the gate”, “shoot, and pick up 3rd spike, which lets us be flexible when playing qualification matches with other teams, and make the auto specifically for playing with this team.

2) Presets autonomous

As well as being able to compose autos on the fly, we also have some default ones, which are a little more advanced, and have tactics as outlined at the top of the page, which custom ones lack. There are presets for both far field play, as well as ones for close shooting, and ofcourse they are mirrored for both of the teams.

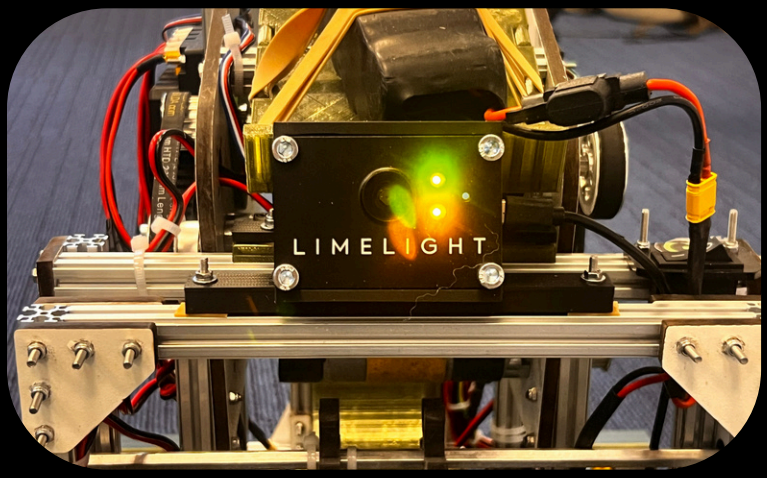
SENSORS

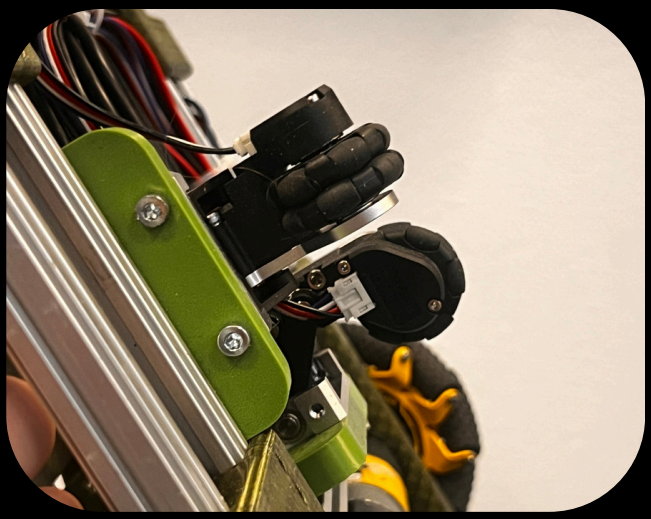
While the hardware and code are important, sensors are the tool of fine-tuning, elevating our design to the next level

Visual Sensor / Camera

In the first iteration of our robot, we have used a webcam, but realised that it took a lot of processing power from the control hub, and was not accurate enough.

We have decided to switch to a limelight, which processes everything on the device, and is more accurate





Deadwheels

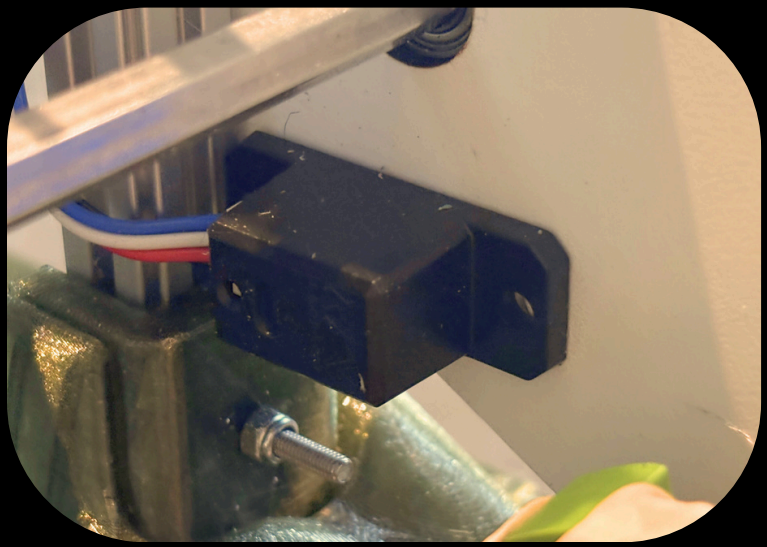
When we do not see the camera, we use the Gobilda PinPoint localizer, which consists of the dead wheels and an IMU to know our relative location.

Those deadwheels turned out to be much better than the method we used on our last iteration, when we used the encoders on our wheel motors to know our position. It has not worked because the wheels slipped, undercounting some movement.

Artifact counter

As in autonomous, we are using a strategy of picking up artifacts directly from the ramp, and our robot is capable of picking up more than 3, so we needed a way to stop the intake when we have 3 artifacts:

We have mounted a distance sensor that detects an artifact coming past it. There are, of course, some limitations; for example, holes in the artifacts make it hard to tell if it is two or one artifact.



TEAM DEVELOPMENT AND COMMUNITY ENGAGEMENT

Building STEM passions through visibility, inclusion, and mentorship

School Lobby Demonstration

To reach students who would not normally encounter robotics, we built a full FTC field in the school lobby and demonstrated our robot throughout the day. Hundreds of students from multiple grade levels observed matches, asked questions, interacted with team members, and drove the robot. Hosting the event in a public space made engineering approachable and sparked interest beyond our existing club members.

Development of team skills

Step 1 — Skills Matrix

At the start of the season, we rate every member on key skills using a simple 0–3 scale:

- 0 watched it
- 1 did it with help
- 2 did it alone
- 3 can teach it

Step 2 — Weekly training block

Every week we run a short, focused training session (60 min) before build time:

- one software topic
- one mechanical/design topic
- one “how we document/test” topic

Step 3 — Biweekly skills checks

- Every 2 weeks, each member completes a small deliverable:
- Software: integrate a sensor, tune a controller, or fix a bug with a clear post-mortem
- Hardware: design/manufacture a part that fits first try OR revise a mechanism based on test data
- Ops: document a design decision with tradeoffs and a clear reason

Step 4 — Teach-back lesson where students teach students

Once per month, members teach what they learned to the team (or younger students). This forces real mastery of the skills that they have learned throughout the past weeks.

Step 5 — Review + improve the plan

At the end of each month:

- We review what training worked and what didn’t
- We adjust the plan (what we teach next and who needs support) to ensure everyone has a clear goal and plan

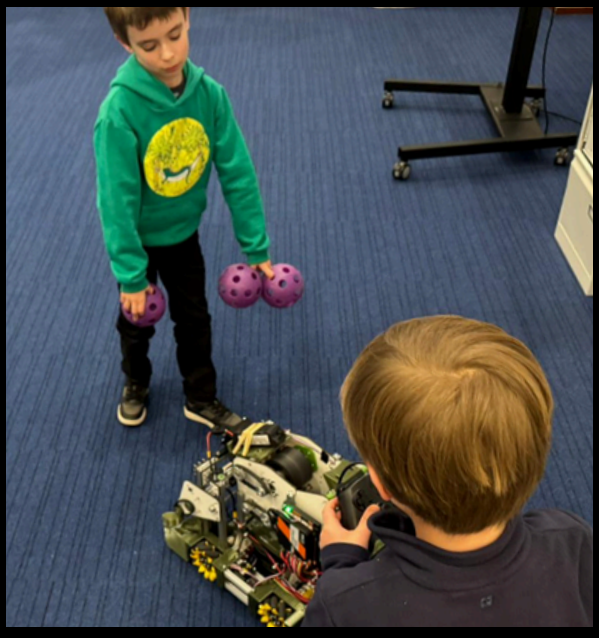
School-Wide STEM Promotion

We shared videos of our robot running and competing on school displays and announcements. These videos explained FIRST Tech Challenge and highlighted how students could join our team and engineering club, increasing awareness and lowering the barrier to entry for new members. When we demo the robot, we also explain the engineering decisions:

- why we chose certain mechanisms
- how sensors improve consistency

Outreach to Younger Students

We also introduce robotics to even younger students through simple robot demonstrations in classes. By focusing on visual explanations and basic concepts, we make robotics engaging and understandable, helping build interest in STEM.



REACH: BUILDING THE "BEAR-SYSTEM"

Strategic Objective: The Talent Pipeline

Our primary outreach objective this season was Sustainability through Vertical Integration. We identified a drop-off in engagement between Middle School and High School. To support the FIRST community, we shifted from passive demos to active integration.

Metric	Goal	Resuilt
Recruitment	Convert "observers" into "builders."	100% of our "Junior League" members have committed to joining FTC next year.
Mentorship	Involve non-technical adults in the team structure.	Recruited 1 Art Dept. faculty member to mentor our Portfolio design.
Awareness	Demystify robotics for the general student body.	Issued 125+ "Robot Driver Licenses" during lunch breaks.

Executing Requirement 2: The "Junior Bear-League"

We transformed our weekly club into the Junior Bear-League. Instead of random lessons, we divided our 10 younger members into two rookie teams using our old chassis and spare parts.

- **Simulated Season:** We (the senior team) acted as the 'Game Design Committee,' creating a simplified challenge for them.
- **The Outcome:** These two internal teams designed, built, and competed in a scrimmage refereed by us. This gave them the full FIRST experience before officially joining, effectively creating two new 'teams' within our school ecosystem.

Recruiting Non-Active Volunteers (Requirement 2)

We recognized that while our technical mentorship was strong, our visual identity and documentation needed professional guidance.

- **The Action:** We pitched the value of STEAM (Science, Technology, Engineering, Arts, Math) to the Arts Department.
- **The Recruit:** We successfully recruited Mr. Van den Hoven, who had no prior involvement with robotics. He now provides bi-weekly critiques of our Engineering Portfolio, teaching us layout principles that elevated our documentation quality.

The Robot Driver's License

To make FIRST 'Loud' in the hallways, we moved beyond staring at a static display. We launched the Robot Driver's License (RDL) initiative during lunch breaks.

- **The Hook:** Students were invited to complete a precision driving course with a slowed-down version of our robot.
- **The Reward:** Upon completion, they received a physical 'Beargineers Class C License' (a 3D printed token).
- **Impact:** This gamified approach removed the intimidation factor. We saw a 40% increase in non-STEM students asking about the club after RDL days.

FINANCE

From €2000 → €3000

Last season: the constraint

In 2024–25 our school budget was €2000. That sounds like a lot, but costs had added up fast. We couldn't buy everything we wanted, so we had to be strategic and improvise.

- Re-used parts from last season and repaired instead of replacing
- 3D-printed brackets/spacers and used off-the-shelf hardware when possible
- Ordered the minimum viable set of spares (to avoid one failure ending our season)
- Chose designs that reduce part count (simple belt paths, modular subassemblies)

Key Numbers

2024-25 school budget: €2000

2025-26 scool budget: €3000

Orders total: $2,134.20

$111.99 shipping + $57.38 tax (≈ 7.9%)

This season: advocating for an increase

To address the budget constraints, this year we advocated for a budget increase to the school administration. We collected evidence, justified our last year budget decisions, and planned for reuse. We documented what limited us last year and presented a clear purchasing plan focused on long-term assets (drivetrain + motors + sensors) and spares.

What We Bought (and why)

Across three major orders ($2,134.20 total) we prioritized parts that directly improve cycle time, accuracy, and reliability, then filled gaps with cost-effective components.

Core performance:

- Strafer® mecanum chassis kit → fast holonomic alignment
- 4-Bar odometry pack + hubs → accurate localization for auto/tele-op
- Yellow Jacket motors (1:1 / 5.2:1 / 13.7:1) - shooter, intake, mechanisms
- Distance sensor + wiring adapters → artifact counting & integration

Reliability & transmission:

- Timing belts (HTD 5mm) in multiple lengths → quick iteration
- Set-screw pulleys + idlers + tensioner brackets → stable belt path
- Flanged bearings + collars + standoffs → rigid, serviceable assemblies
- Rhino / boot wheels for odometry + rollers / intake contact

Where we compromised

Instead of buying every upgrade at once, we delayed optional mechanisms and limited spares to the parts most likely to fail. We reused electronics where possible and chose components that keep the design modular-so we can upgrade later without rebuying the base.

FIRST VALUES

1) Discovery - "*We explore new skills and ideas*"

Our team actively explores new skills and ideas through hands-on learning in engineering, programming, and problem-solving. By researching game challenges, experimenting with designs, and learning from mentors and peers, we continuously expand our technical and collaborative knowledge.

2) Innovation - "*We use creativity and persistence to solve problems*"

We apply creativity and persistence to develop original solutions to competition challenges. Through iterative design, testing, and refinement, our team embraces failure as part of the learning process and uses innovative thinking to improve robot performance and team efficiency.

3) Impact - "*We apply what we learn to improve our world*"

Our team applies the skills and knowledge gained through FIRST to make a positive impact beyond competition. By engaging in outreach, sharing STEM knowledge, and acting as role models within our school and community, we aim to inspire others to explore science, technology, and engineering.

4) Inclusion: "*We respect each other and embrace our differences*"

We foster a respectful and welcoming team environment where all members feel valued and heard. By embracing diverse perspectives, skills, and backgrounds, we ensure that every team member has the opportunity to contribute meaningfully to our success.

5) Teamwork: "*We are stronger when we work together*"

Our success is driven by collaboration and mutual support. Team members work together across engineering, programming, outreach, and strategy, recognizing that open communication and shared responsibility strengthen both our robot and our team.

6) Fun: "*We enjoy and celebrate what we do!*"

We celebrate learning and teamwork by maintaining a positive and enthusiastic team culture. Whether building, coding, or competing, we enjoy the process, support one another, and take pride in the experience of being part of FIRST.

FUTURE PLANS (SUSTAINABILITY)

Our goal is to ensure that our robotics club remains sustainable, inclusive, and continuously improving beyond a single season. We focus on long-term growth by developing members and strengthening our presence within the school community.

Team Development & Knowledge Transfer

- Maintain clear documentation of designs, code, and lessons learned to support future teams.
- Involve younger members directly in building, programming, and testing to ensure hands-on learning.
- Encourage mentorship within the team by pairing experienced members with newer students.

Recruitment & Growth

- Continue hosting school-wide demonstrations and outreach events to attract new members, as most of the team is made up of seniors
- Use visible projects and competition footage to spark interest in engineering and robotics.
- Lower the barrier to entry by welcoming students with no prior experience and providing structured learning sessions.

Education & Skill Building

- Run regular club meetings focused on teaching core engineering, programming, and design principles.
- Introduce CAD, manufacturing, and software concepts early to build strong technical foundations.
- This emphasizes design and problem-solving over simply building a robot.

Organization & Continuity

- Improve internal organization by clearly defining roles and responsibilities within the club, so no one is left doing nothing.
- Create season timelines and milestones to keep development on track.
- Ensure smooth leadership transitions by preparing underclassmen for future leadership roles, allowing the continuation of the club for the future.