#28080 VegaTech
Engineering Portfolio

"Luka"

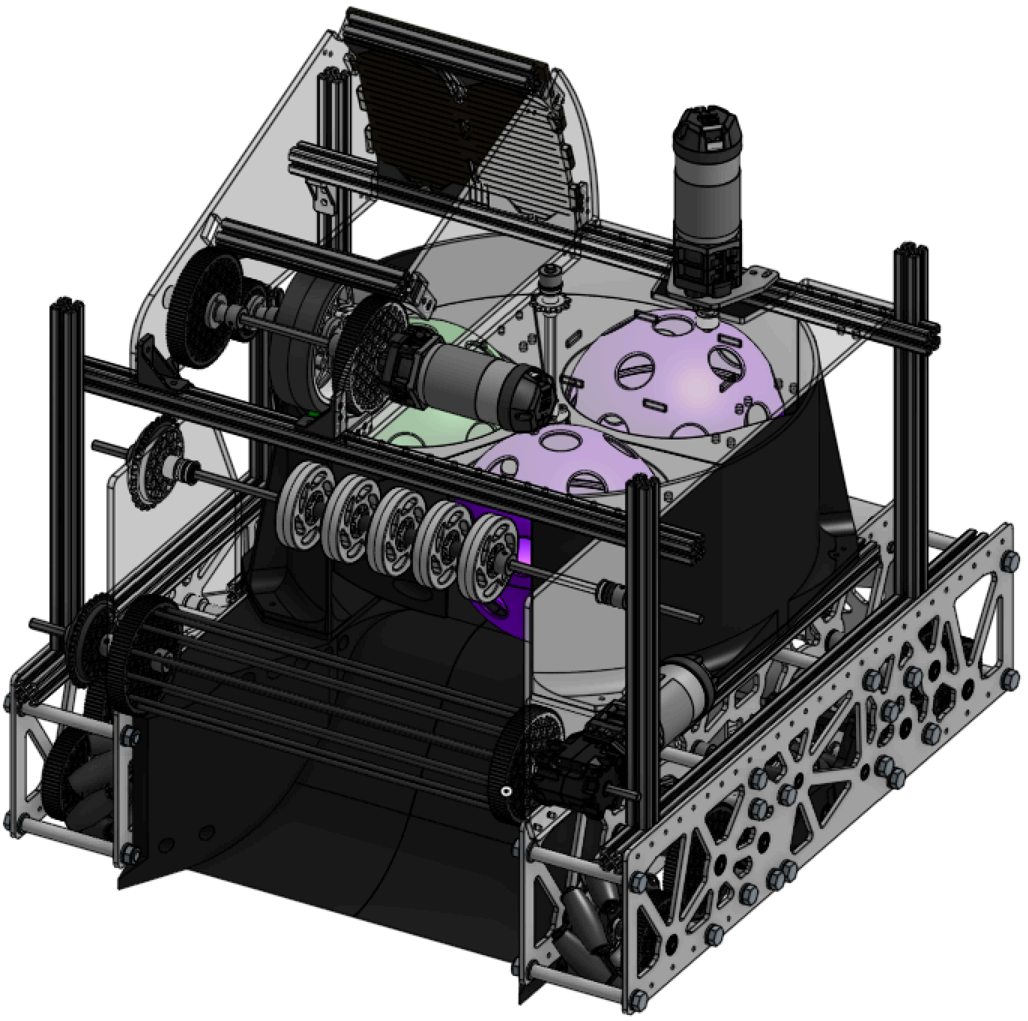


Table of Contents

Our Story

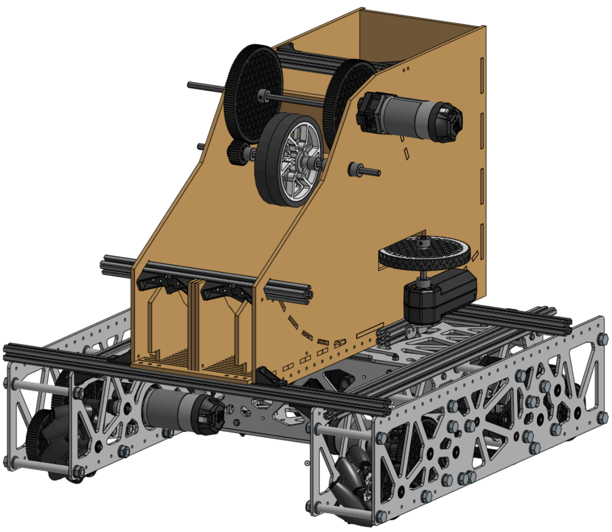
About Us

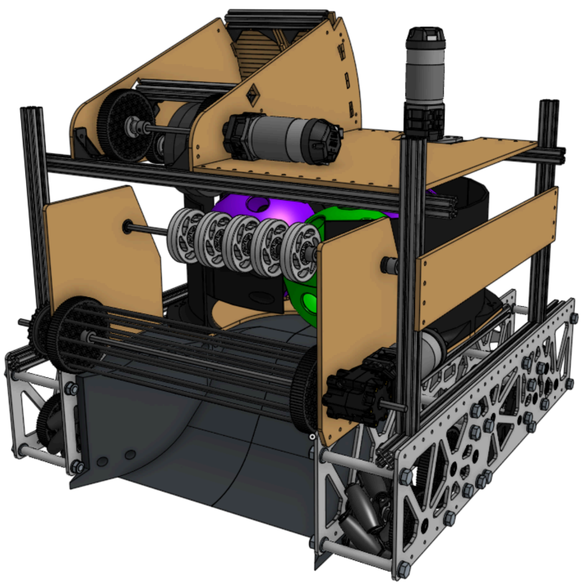
We are a team of three students from the Electrotechnical and Computer Technical School and Gymnasium Ljubljana. We are located in Ljubljana, Slovenia. We are sister team to #22903 VegaMind, and together we operate under the umbrella of *FIRST* Slovenia, where we compete in the summer at the FIRST Global Challenge.

Oberkochen Scrimmage

On November 15, 2025, we went to a friendly competition in **Oberkochen**, Germany. There we tested our first robot **prototype** with a new drivetrain. We also encountered the game elements, which until then we had 3D printed from PLA.

The scrimmage was successful for us, as we tested the shooter and the drivetrain, and on top of that we also placed **5th**. There we also saw many robots that gave us new ideas.





Germany Championship (Stuttgart)

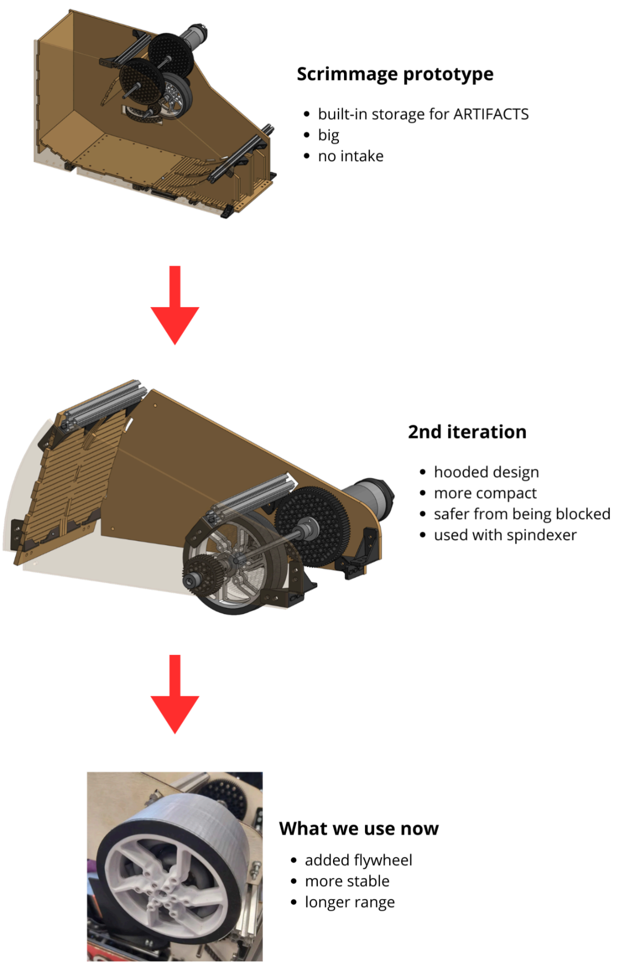
On January 31st, we went to the Germany Championship in **Stuttgart**. There, we **tested** a **new robot design** that is completely different from the one we used in Oberkochen.

For us, the competition was a success because we **almost reached** the **playoffs**, and the judges showed interest in FTCTui. Our sister team VegaMind also **won** the **Control Award** there.

Design

Shooter

The first version of the shooter was made for the scrimmage in Oberkochen. Back then, **it didn't have a hooded design yet**, but it did feature built-in **storage for 3 ARTIFACTs**. You could say we got lucky; we wanted to use REV 3" ION Compliant Wheels, anticipating that **centripetal force** would cause the wheels to expand. This worked with the **3D-printed ARTIFACTs**, but it failed when we tested it with actual ARTIFACTs. Fortunately, we brought REV 90 mm Grip Wheels with us. Since they are larger, we tested them and discovered they **worked perfectly**. Afterwards, we decided to use non-compliant wheels.



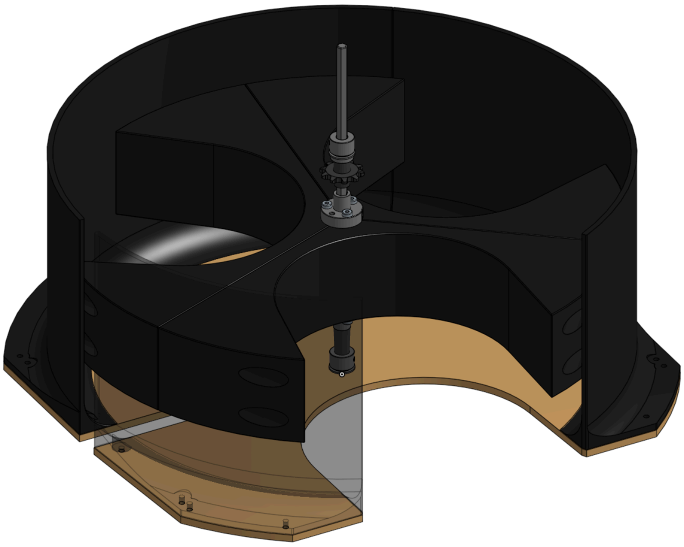
Upon our return, we began fabricating a **hooded** design. With this shooter, we also utilize a spindexer for indexing, so the lack of storage capacity is not an issue. Initially, we attempted to construct it using REV 60 mm Traction Wheels, which were selected due to their lower profile and the resulting compactness of the robot. These proved to be unusable, as they failed to transfer sufficient velocity to the ARTIFACTs. We then reverted to REV 90 mm Grip Wheels, which proved to be very successful due to their **greater mass** and **consequently higher inertia**. We also tested numerous gear ratios. Ultimately, we decided on a 1.5:1 ratio, which provides the maximum power at the desired RPM.

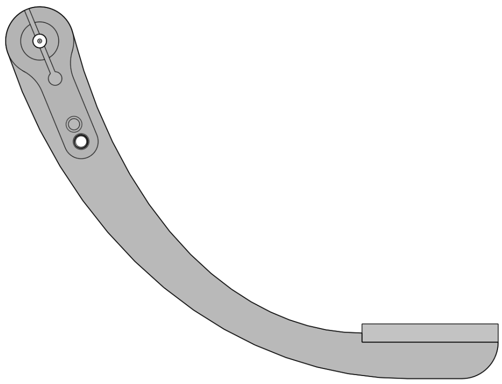
With this gear ratio, we can **theoretically achieve** a maximum of 4000 RPM using a REV HD Hex Motor, providing us with enough power to launch the ARTIFACT into the GOAL from the far zone. After experiencing some **issues** with the code (**PID**), we rebuilt the structural framework, which is now stronger and doesn't vibrate as much; consequently, the PID control functions more effectively.

Following the Germany Championship, we determined that the robot would shoot better if we added a **flywheel**. We were resourceful in this regard and utilized **e-waste—old electrical cables**—which we embedded between two REV 90mm Traction Wheels. This modification simultaneously increased the velocity, range and stability of our shots.

Spindexer

Ever since Oberkochen, we knew that we wanted to mount the **spindexer horizontally**, positioned above the chassis and below the shooter. The first version was created solely in CAD, intended to be constructed from wood. After observing that VegaMind's 3D printed version functioned well, we decided to 3D print ours as well.

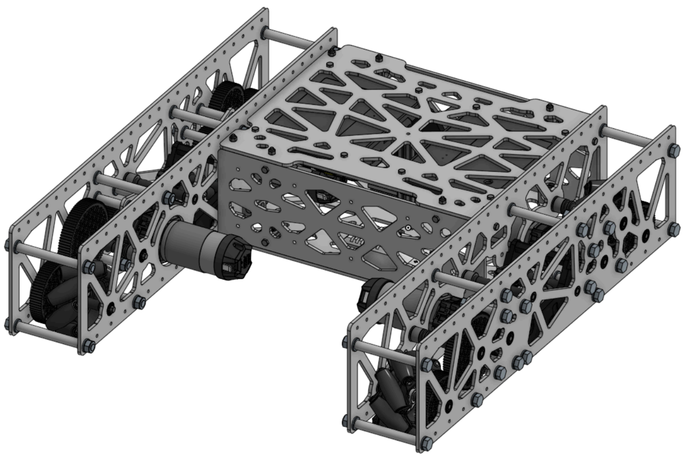




To transfer the ball from the spindexer into the shooter, we utilize a servo with a custom laser-cut part mounted to it. We calculated the dimensions based on the positions of the servo and the spindexer. In doing so, we **accounted** for the **path of the specific point contacting the ARTIFACT**; we also had to be mindful of the shooter and spindexer walls to ensure the ball would not jam or be forced against a wall.

Drivetrain

For the drivetrain, we used the chassis from **Project Strand**. The entire drivetrain is laser-cut from aluminum. It uses a **sandwich design**, which allows us to mount mechanisms quickly and easily.

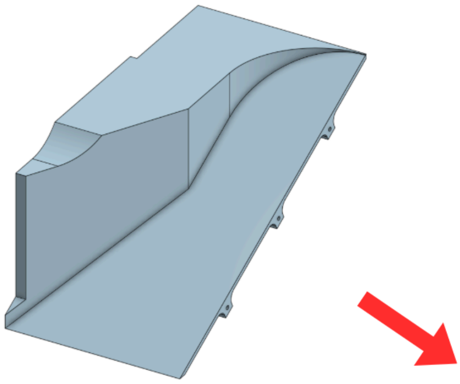


For propulsion, we use REV HD Hex Motor motors and REV 75 mm Mecanum Wheels, which are driven through a gear transmission. In the center, there is also space for the electronics.

We also use a **3-point odometry** system that **we 3D printed** ourselves. It works using a REV Through Bore Encoder, and we use it together with the IMU from the REV Control Hub.

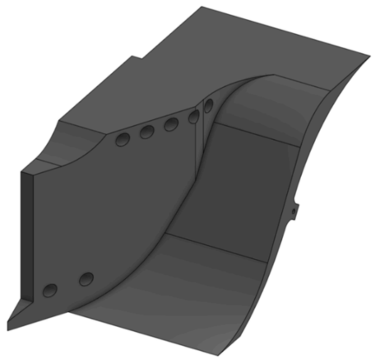
Intake

Since we intended to position the spindexer above the chassis, the first requirement for the intake was a ramp. We constructed **two versions**: the first had a linear profile, while the second featured a specific S-curve.



1st prototype

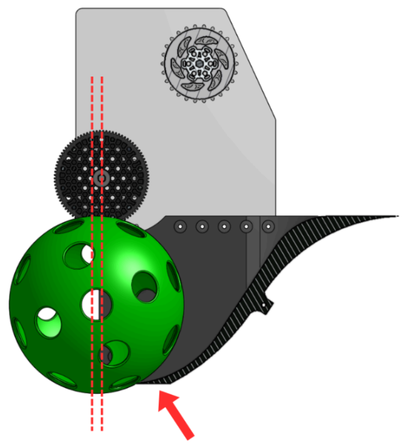
- linear ramp
- we had to push ARTIFACTS against a wall to intake



2nd iteration

- non-linear ramp
- moved start of the ramp more to the back for more leverage on the intake

The initial prototype was suboptimal, as we could not intake ARTIFACTs from a stationary position and **had to push them against a wall** to collect them. We resolved this in the final ramp design by **recessing the curve further into the robot's interior**. This improved the intake's performance by engaging the ARTIFACTs closer to their center. On this ramp, we also **integrated a funnel** to center the balls and guide them toward the spindexer. Initially, we encountered issues with traction, resulting in the balls merely spinning in place. We solved this simply by applying duct tape to increase friction.



The second crucial thing we had to create was the actual intake mechanism. We decided on a **rubber band** intake, as it offers a good **trade-off** between **grip** and **compression factor**, which is helpful for the ramp due to its specific shape. We utilized REV Surgical Tubing, which we wound around two 90-tooth gears. It is driven by a REV HD Hex motor via a 90-degree gearbox and a 1:3 reduction. Since the chassis is very high, we also needed a 2nd row of wheels. For this, we used REV 2” Compliant Wheels, which provide high grip. We connected both rows of the intake via a chain.

Software

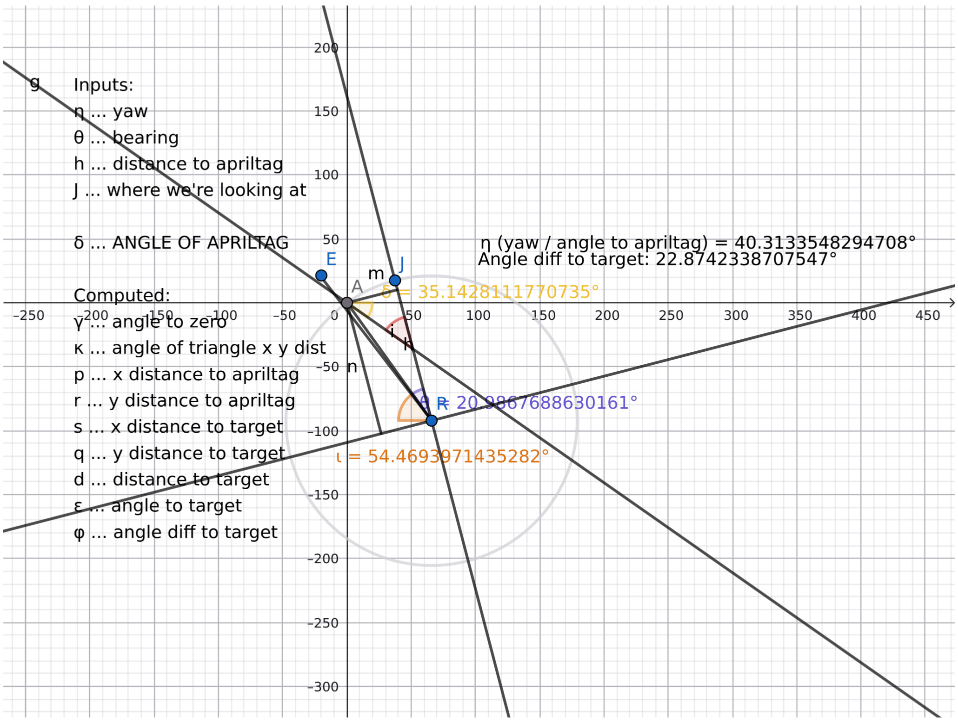
As in previous years, we developed the code in Android Studio using Java. Our programmer, Natan, is solely responsible for it. Currently, the codebase consists of approximately **5400 lines** of code across **44 files**.

Language	Files	Lines	Code	Comments	Blanks
Java	42	5315	3454	842	1019
Markdown	2	132	0	88	44
XML	3	176	11	150	15
Total	47	5623	3465	1080	1078

```
koza@fw16:~/code/FTC2026-Vegatech/TeamCode$
```

The first functional component was the **drivetrain**. Here, we can highlight our own **custom home-brewed PID** control system, which maintains the robot at the desired **heading**. The driver **separately** inputs the desired direction of **translation** and the desired **rotation**. Both are **field-centric** (or, more precisely, relative to the robot's orientation when the OpMode initializes). A **significant portion** of this code (and this driving concept) originates from last year's FGC in Panama; however, we had to **adapt** the code for FTC to function with Mecanum wheels.

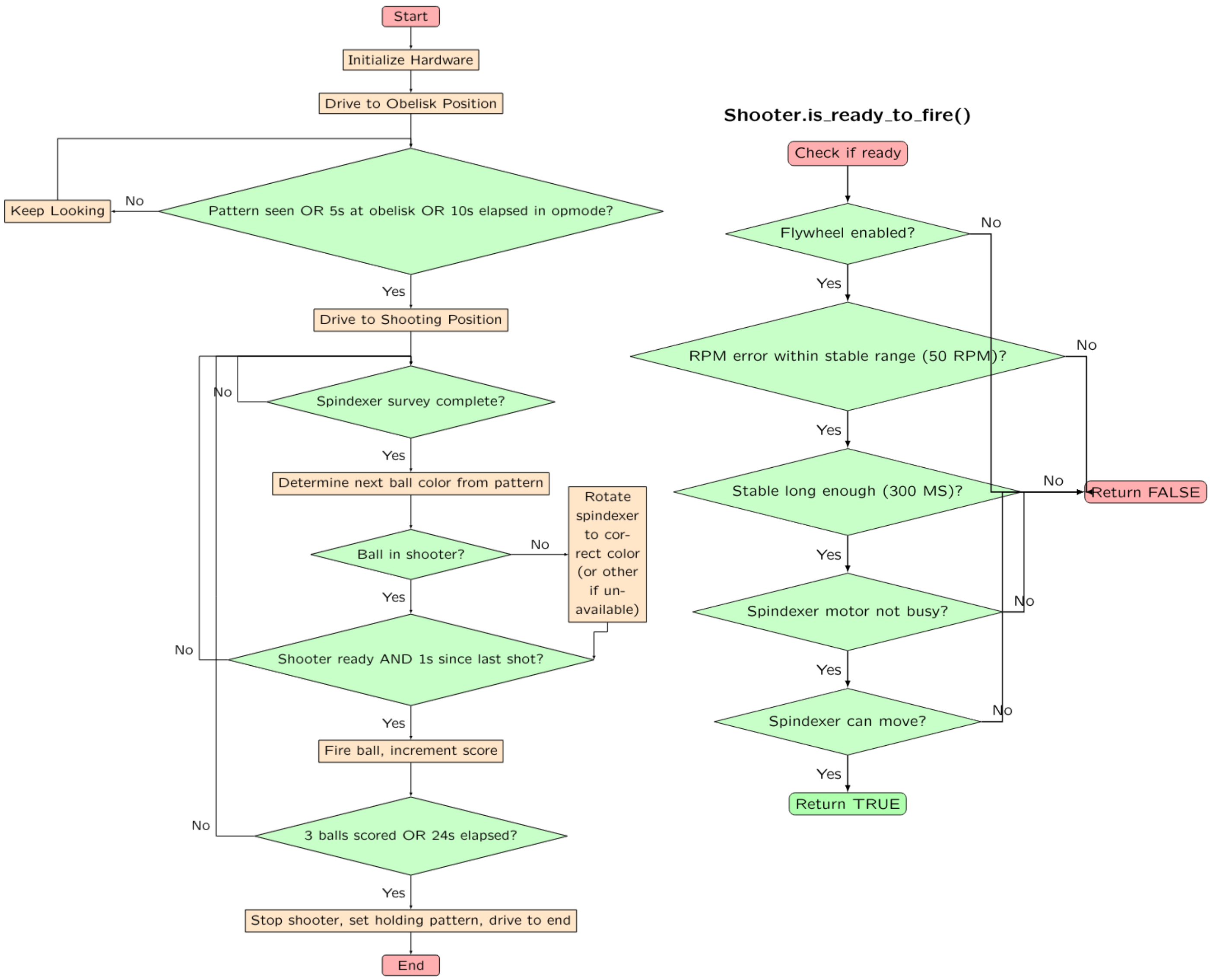
This **same system** also enables us to command the robot to rotate to the **optimal angle** for **shooting** ARTIFACTs. We derive this angle using a **camera** that detects the **AprilTag** on the basket, combined with some **geometry calculations**, which were likely the most challenging part of this year's code.



Naturally, shooting ARTIFACTs requires the **shooter** itself. Here, the code is responsible for measuring **RPM** and regulating it using **two additional home-brewed PID** controllers. We determine the desired **shooter velocity** based on the **distance** to the target—the center of the basket—(obtained via camera) and a **polynomial** that converts distance into velocity based on empirical **measurements**. We chose this approach because we find it easier to measure a few distances and **interpolate** between them rather than attempting to accurately approximate air resistance, elastic potential energy, and other physical factors influencing the shooter's range.

We should also mention the fully **autonomous spindexer**, which automatically detects which balls are loaded at the start of the match; it then **tracks the colors** of the balls inside it and manages the intaking process based on a **color sensor**. When we wish to launch a ball, we simply communicate the desired color to the spindexer.

For autonomous, we are utilizing the **Pedro Pathing** library for the first time this year. At the start of the OpMode, we identify the sequence displayed on the obelisk, then navigate toward the basket and launch **three balls** in the **correct order**. Finally, the robot moves to the optimal starting position for the TeleOp phase.



We have **several** different **OpModes** depending on the starting position and alliance color to ensure **maximum flexibility**. To avoid excessive code **duplication**, we utilize a **parent** class for each autonomous type (which e.g. navigates the robot to the basket and launches all stored balls), while each individual OpMode is an **inherited** class that simply overrides the specific target positions. Here, we **automatically calculate the optimal angles** for shooting balls and for viewing the obelisk with the correct sequence of balls.

A crucial aspect of our software development is **testing**, allowing us to quickly identify the root cause of issues. Therefore, we created a **diagnostic OpMode** that enables us to individually test all **hardware components** (motors, servos, etc.) as well as **specific subsystems** (drivetrain, shooter, etc.). For more complex systems, we have dedicated testing OpModes: for instance, one OpMode is dedicated solely to the shooter's PID controller, another to camera mathematics, and so on. To make this possible, all subsystems are encapsulated in their own classes and structured in accordance with SOLID principles.

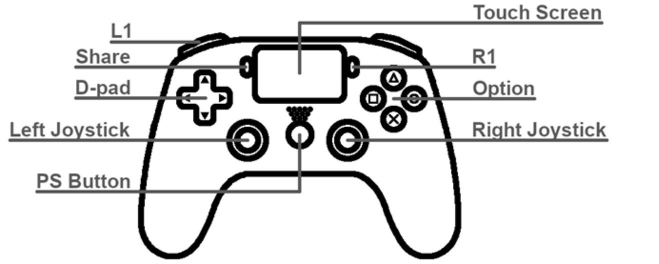
This year's robot relies heavily on **sensors** for the **intake**, **shooting**, and **spindexer** management. On one hand, this is beneficial; if all sensors function correctly, we **reduce driver workload and complexity**. However, we acknowledge that sensors are **not infallible**. As a contingency, the drivers can **manually override** and control all autonomous subsystems if necessary.

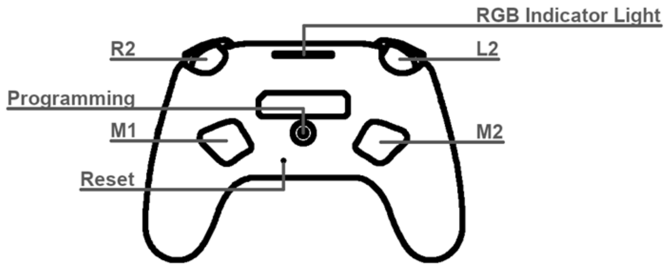
For the first time, we have integrated an **LED indicator** on the robot, allowing us to easily monitor the robot's status. The LEDs provide **visual feedback** on whether an AprilTag is detected, if the shooter has reached the target velocity, and which color ball is currently being queued for launch.

Throughout development, we adhered as closely as possible to the principles of clean, maintainable, and readable code. The entire repository can be accessed on our GitHub via the QR code below.

Controls

Driver 1





Driver 1 is primarily responsible for the drivetrain—specifically the movement and rotation of the robot—while also controlling the intake.

Left Joystick – Field-centric translation

Right Joystick – Field-centric target heading/angle for robot rotation

D-pad – Precision field-centric translation

A / X – Align to field (useful for endgame)

X / □ – Toggle intake wheels

Y / △ – Scan for AprilTags and rotate towards them at the ideal angle for the shooter

Guide / PS Button – Set current heading as “forward” for field-centric controls (reset IMU)

Driver 2

Driver 2 is responsible for the remaining subsystems on the robot: the shooter and the spindexer.

X / □ – Activate shooter

R2 (Right Trigger) – Automatically launch ball when the shooter reaches target velocity

L2 (Left Trigger) – Override for R2; launch ball regardless of velocity

L1 (Left Bumper) – Select green ball for launching

R1 (Right Bumper) – Select purple ball for launching

Y / △ – Reset spindexer from shooting state

A / X – Scan/Check which ball colors are currently in the spindexer

B / O – Move spindexer to position 0 (backup)

D-pad Up / Down – Manually rotate spindexer (backup)

FTC Tui

The story begins following the conclusion of the *Into The Deep* season. Having grown weary of the Driver Hub's various issues after just our rookie season, we conceived the idea of **controlling the robot directly via a computer**.

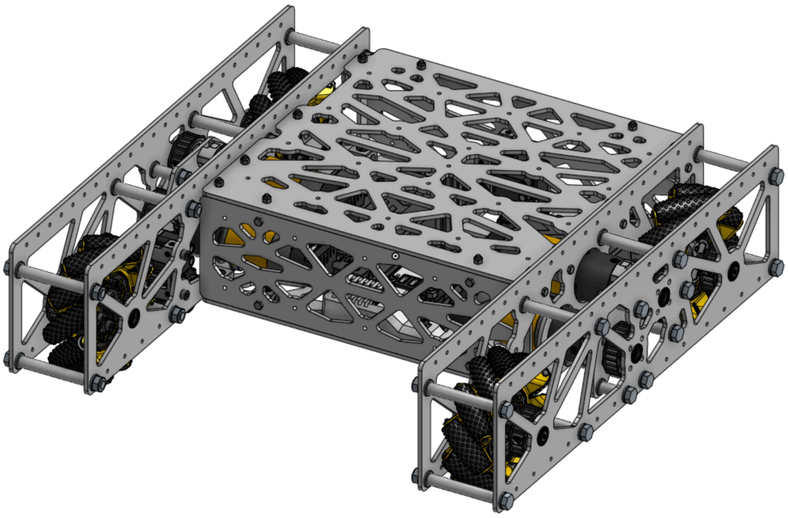
Initially, we attempted to emulate the Driver Hub within a virtual machine. This approach presented significant difficulties. Natan then proposed **reverse-engineering** the communication protocol and re-implementing it within a new standalone application. This would allow programmers to write, upload, and test code from a single device. (We would **no longer require a physical Driver Hub within the workshop at all**). After several months of protocol forensics, Natan **implemented** the solution **in Rust**, enabling robot control via a computer.

The advantage of this approach is that it requires **no additional application installation** on the **Control Hub**, meaning it can be used **easily** and without major complications. It allows **any computer** to control any robot without modifying the robot's code. Furthermore, it features additional functionality, such as **exporting telemetry** to a file, enabling us to **plot graphs** using Python and Matplotlib. This capability allowed us to calibrate our custom home-brewed PID controllers this year.

Currently, the application is in the beta phase. **Source code** and **release builds** can be accessed via the adjacent **QR code**. We anticipate a full release in the coming months, which will include the implementation of the robot's hardware configuration. In the future, we also intend (permission permitting, of course) to **publish** the **protocol documentation**, thereby allowing **other teams** to **benefit further** from our newly acquired understanding of the protocol.

Project Strand

This is an **open-source** project aimed at standardizing robot components, enabling faster, easier, and, crucially, more **cost-effective** robot design, while also facilitating the easy interchangeability of mechanisms. The project is led by FIRST Slovenia—specifically, us and Team VegaMind—and is targeted at **rookie teams** and teams with **limited financial resources**. Currently, the project includes various chassis components (compatible with both REV and GoBilda), linear slides, and structural elements. We also intend to add components incorporating the new control systems and motors for the 2026/27 season."



Outreach and social media

Social media

We are currently active exclusively on Instagram, where we have 140 followers. This season, we initiated a series of **informative videos** briefly explaining our robot's functionality. These have gaind us significant viewership, particularly from non-followers (averaging **around 1500 views per video**).

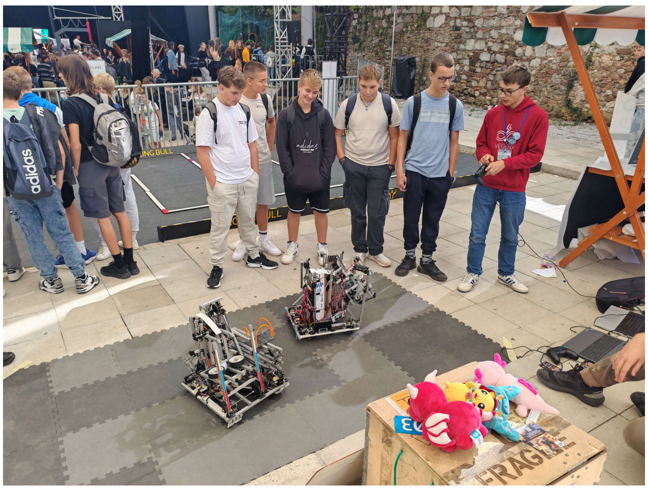
FIRST Lego League

This year, we presented robotics at the 2025 **National Championship** as well as at two **regional competitions**. At these events, we allowed children to drive our FGC robots, **explained** the operation of **FTC robots**, and outlined our plans for the current season.

MOS (International Trade Fair)

MOS (The International Trade and Business Fair) is a pan-Slovenian fair featuring exhibitors ranging from food and furniture to various technology companies. As student representatives from our high school, we presented the FGC and FTC core values and competitions. We were also visited by the **former Prime Minister**, to whom we introduced both **FGC and FTC**..

Informativa



Informativa is an education and career fair for high schools and companies. We represented our high school, allowing children to drive our FGC robot. During the event, we also **introduced** the **FIRST** program to **other schools**, inviting them to participate and **shared** our **contact**

information to assist in the formation of **new teams**.

Arena Tehnologij

Arena Tehnologij is an event showcasing numerous scientific and research projects. We presented both FGC and FTC there.

Information Days (Informativni dnevi)

Information Days are events held at our school, designed to introduce the school to prospective students. We presented FIRST as an extracurricular activity.

CAD Course

Following the conclusion of the *Into The Deep* season, we organized an extensive free 3D modeling course using Onshape. It was aimed at all **members of FIRST Slovenia** as well as **students from our school**. In the course, participants learned how to design custom parts and how to assemble a robot using pre-existing components.

Our Vision, Finances & Sponsors

Expansion of FIRST in Slovenia

Our goal for the future is to **expand FIRST throughout Slovenia**. We intend to achieve this by maximizing our presence at various events and creating resources for other teams (which we are already actively doing). Furthermore, last month we applied for a STEM grant. If funded, we will launch 10 new teams over the next two years at a lower cost to the teams than usual (via **co-financing**), prepare educational materials **in Slovenian**, and organize a scrimmage and a championship. We have already developed the **FIRST Slovenia FTC Starter kit**, costing teams about 1000 €. We would also provide rookie teams with an additional 2000 € in funding. The advantage of having more teams is **increased collaboration** and **development** within and between teams, as well as **increased recognition of FIRST in Slovenia**, which consequently leads to easier acquisition of sponsors, leading to even more teams, and so on.

Finances

We regularly seek new sponsors with the assistance of our mentors. We also look for sponsors who can assist us directly with their products, such as *Plastika Trček*, which donates filament for 3D printing. Additionally, we create projects for our sponsors, such as the 'Pajk Project,' which is a robotic arm designed to draw motifs on hats for the company *Pajk Klobuki*. ("pajk" in slovenian means "spider", btw.)
We also strive to be economical with materials and **reuse** them (such as reusing the chassis for future years); for **non-final** versions of the robot, we use **wood**, which is cheap and easily accessible.

We would also like to thank our biggest sponsors:

- Vegova ljubljana (our high school, providing us travel through ERASMUS)
- Zavod 404 (workspace)
- Mestna občina ljubljana (City Municipality of Ljubljana)
- Plastika trček (our 3D printer filament supplier)
- Delavska hranilnica
- Zavarovalnica triglav
- 1Home Solutions GmbH