



Engineering Notebook

2026
SEASON

ARTEMIS

CONTENTS

Chapter 1: Game Analysis

- Game overview
- Field layout
- Scoring system
- Match structure
- Game challenges

Chapter 2: Strategy

- Match strategy
- Robot objectives
- Priority list
- Strategic decisions and trade-offs

Chapter 3: Engineering Process

- Brainstorming
- Concept development
- Design decision process
- Prototyping
- Testing
- Iterations and improvements

Chapter 4: Robot Architecture

- Overall robot architecture
- Main robot components
- Design decisions

Chapter 5: Robot Subsystems

- Drivebase
- Intake
- Feeder (gravity indexer)
- Shooter
- Extendable hopper
- Post-test improvements

Chapter 6: Programming

- Software architecture
- Autonomous mode
- Teleoperated mode
- Sensors
- Robot localization
- Testing and debugging

Chapter 7: Testing & Results

- Test procedures
- Results and performance analysis
- Robot improvements

Chapter 8: Lessons Learned

- Main challenges
- Engineering lessons
- Future improvements

GAME ANALYSIS

01



GAME ANALYSIS

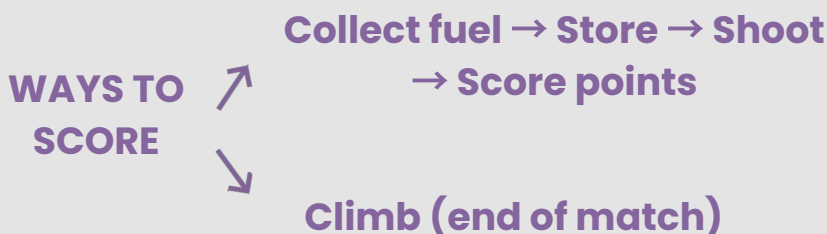
Game overview

The 2026 challenge, REBUILT presented by Haas, asks teams to design and build a robot that maximizes points through different strategic actions.

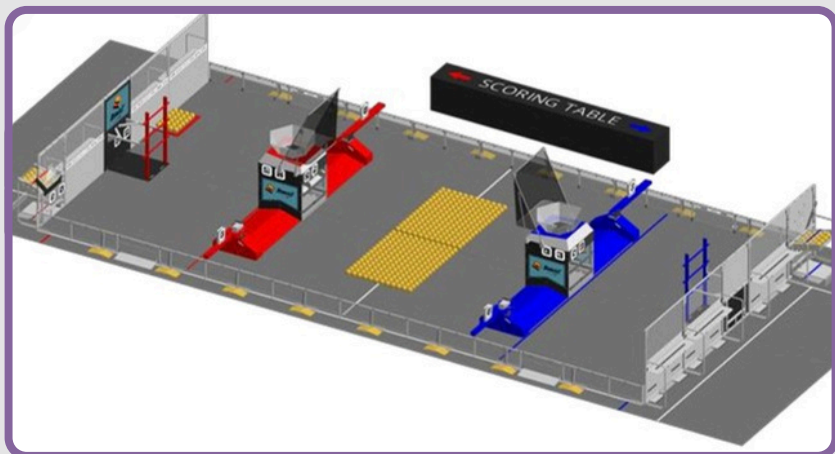
We identified two main ways to score: shooting balls (fuel) into the alliance hub, and climbing the tower at the end of the match. Each action has a different value, so from day one we analyzed which offered the most scoring potential relative to the time available in a match.

The match has an autonomous period and a teleoperated period, so the robot must perform actions both automatically and under driver control. We focused our analysis on how to use each stage of the match to maximize the efficiency of the system.

Instead of designing a robot that could do every possible action, we chose to prioritize the actions that generate points consistently throughout the match. That became the foundation of our strategy.



Field layout



The field lets several robots work at the same time, creating a dynamic environment where efficient movement is key.

During our field analysis we mapped the fuel collection zones and the scoring targets. One of the most relevant elements was the bump: an obstacle the robot must cross without getting stuck.

This led us to keep the tank-style (KitBot) drivetrain, which offers stability and the ability to cross this kind of obstacle without losing control. That decision let us focus on increasing fuel storage capacity without risking mobility on the field.

Scoring system

Points come mainly from shooting fuel into the hub, where every shot that goes in adds to the alliance score. We identified shooting speed and consistency as the key factors for maximizing points in a match.

We also analyzed the end-of-match climb, which awards additional points. However, it requires dedicated time and an additional mechanism. Based on this analysis, we decided to focus our design on optimizing the shooter, prioritizing the ability to score continuously instead of depending on a single end-game action.

Match structure

A match is divided into stages in which the robot must perform different actions. First comes the autonomous period, where the robot runs only on its programming. Then the teleoperated period begins, where drivers control the robot to move around the field, collect balls and shoot at the target. The robot's performance also depends on coordination with its alliance partners, so from the start we considered how important it is to contribute steadily to the alliance's total score. Based on this, we focused the robot's design on actions it could perform continuously during both phases of the match.

Game challenges

One of the main challenges was building a system that could shoot balls quickly and consistently. Our analysis showed that poor fuel alignment inside the robot could cause jams or inaccurate shots, significantly reducing the system's efficiency. Another major challenge was mobility: the robot had to move quickly and cross obstacles like the bump without losing stability. From these challenges, the team defined three main design criteria:

Speed, to collect and shoot balls in the shortest possible time. Storage capacity, to keep a continuous flow of fuel. Shooting consistency, to reduce errors and maximize points.

These criteria guided the team's engineering decisions and became the foundation of the robot's strategy.

“WE DECIDED TO DESIGN A ROBOT THAT COULD CONTRIBUTE POINTS CONTINUOUSLY THROUGHOUT THE ENTIRE MATCH.”

**“FROM THE GAME
ANALYSIS, WE CHOSE
TO PRIORITIZE ACTIONS
THAT GENERATE POINTS
CONSISTENTLY
THROUGHOUT THE
MATCH.”**

STRATEGY

02



STRATEGY

Match strategy

After analyzing the game and understanding how points are earned, the team decided to focus the robot's design on maximizing point generation throughout the whole match.

To do this, we compared the two main strategies: shooting fuel and climbing.

Strategy	Estimated points	Time required / Level	Consistency	Notes
Fuel shooting	~10 points	50 seconds	High	Generates points continuously throughout the match.
Climb	~10-20-30 points	Level 1 Level 2 Level 3	Medium	Only possible at the end of the match.

“Although climbing awards points, we decided not to build this mechanism: it required dedicated time at the end of the match and design resources that were better invested in the shooting system.”

Robot objectives

Based on this strategy, we defined the following design objectives:
An intake that collects balls quickly and continuously without jamming
A storage system that keeps a constant flow of fuel
A consistent shooter that maximizes the probability of scoring
These objectives guided every engineering decision during the robot's development.

Priority list

Priority	Element	Justification
1	Shooting speed	Increases the number of balls launched during the match.
2	Consistency	Reduces errors and prevents jams inside the robot.
3	Storage	Sustains a continuous shooting flow.

These priorities were defined from the game analysis and the robot's first tests, with the goal of maximizing efficiency during the match.

Strategic decisions and trade-offs

During the design process, the team made key decisions to optimize the robot's performance.

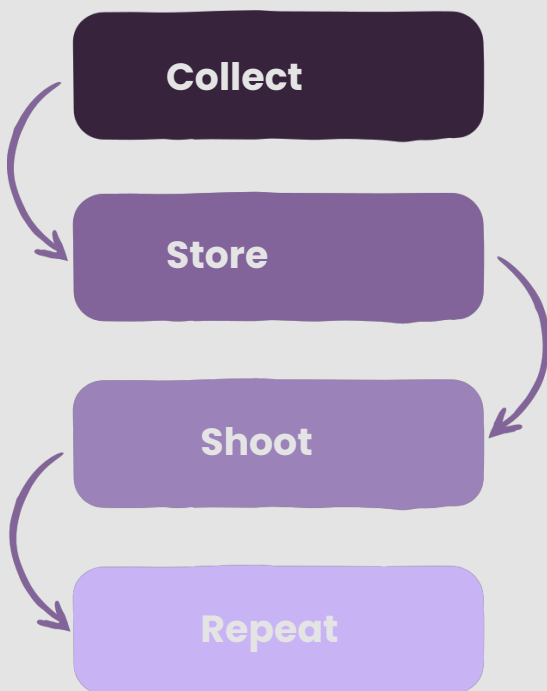
One of the most important was not to prioritize a climbing mechanism. Although climbing awards points, it requires dedicated time at the end of the match and an additional mechanism.

Instead, we invested our resources in improving the speed and consistency of the shooter, because that would let us generate more points throughout the match.

This approach meant an important trade-off: giving up a single high-value action (the climb) in exchange for a strategy of constant point generation.

We also prioritized efficient fuel handling, adjusting the intake and shooter to prevent jams and improve ball alignment.

Overall, our strategy centered on a robot that can collect, store and shoot balls continuously, maximizing its impact throughout the match.



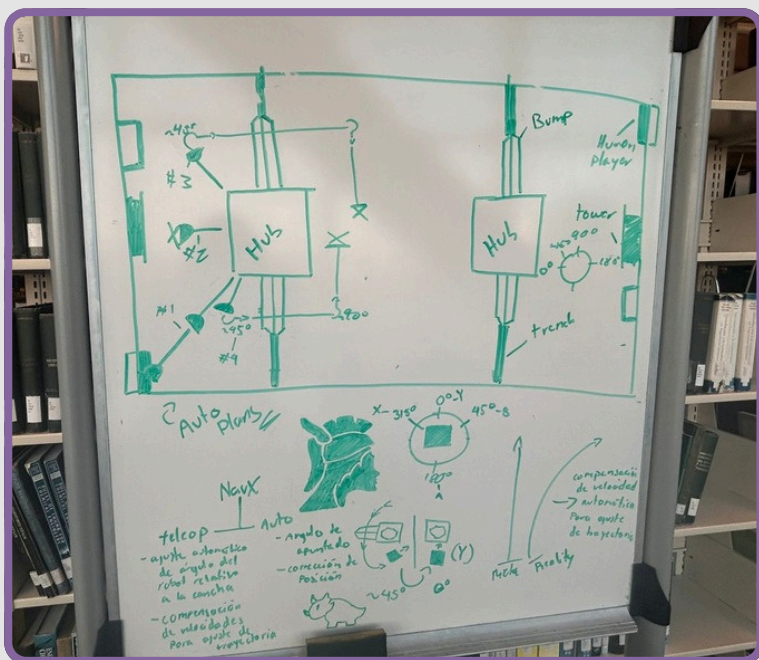
ENGINEERING PROCESS

10

03



ENGINEERING PROCESS



Brainstorming

The design process began with an analysis of the season challenge and the KitBot's starting capabilities. From there, we evaluated what the base robot could do and where it fell short before proposing new solutions. During brainstorming, the team identified three key areas for improvement: shooting speed, ball storage capacity and fuel alignment inside the robot.

These ideas focused the robot's development on improving existing systems instead of designing entirely new mechanisms from scratch.

Concept development

Building on our first ideas, we decided to focus on optimizing ball handling. Our analysis of the KitBot showed that its shooter worked but was inconsistent in speed and direction, so we decided to improve both the intake and the shooter. We also identified the need for more storage capacity, which led to the concept of an extendable hopper.

Design decision process

Every design decision was made based on its direct impact on the robot's performance on the field.

We kept the KitBot tank drivetrain, because it crosses the bump without compromising stability

We optimized the intake and shooter to improve shot consistency

We added an extendable hopper to increase storage capacity

These decisions prioritized efficiency, simplicity and on-field performance.

Iteration V1: Base system (KitBot)

Decision: use the original KitBot as the starting point

Problem found: low shooting consistency and poor fuel alignment

Result: the robot worked, but was not competitive in speed or accuracy

Iteration V2: Better alignment and flow

Decision: improve ball flow inside the robot

Changes made:

added cone wheels to align the fuel

redistributed the compliant wheels

Problem solved: fewer jams and better shot direction

Result: more consistent launches

Iteration V3: Shooter optimization

Decision: optimize fuel compression and exit speed

Changes made:

compliant wheels of different durometers

shooter configuration adjustments

Result:

higher shooting speed

higher scoring success rate

significantly fewer errors

Prototyping

Before building the final design, we validated our ideas with simple prototypes.

The most important was a cardboard prototype of the extendable hopper, used to study how it worked and how the fuel aligned inside it.

This let us adjust the design before building the final mechanism.

Testing

To validate the robot's performance, our tests focused on:

shooting speed

fuel alignment

scoring success rate

These tests showed that the main problem was poor ball alignment, which caused jams and inaccurate shots.

Iterations and improvements

Based on the results, we made targeted improvements to the system:

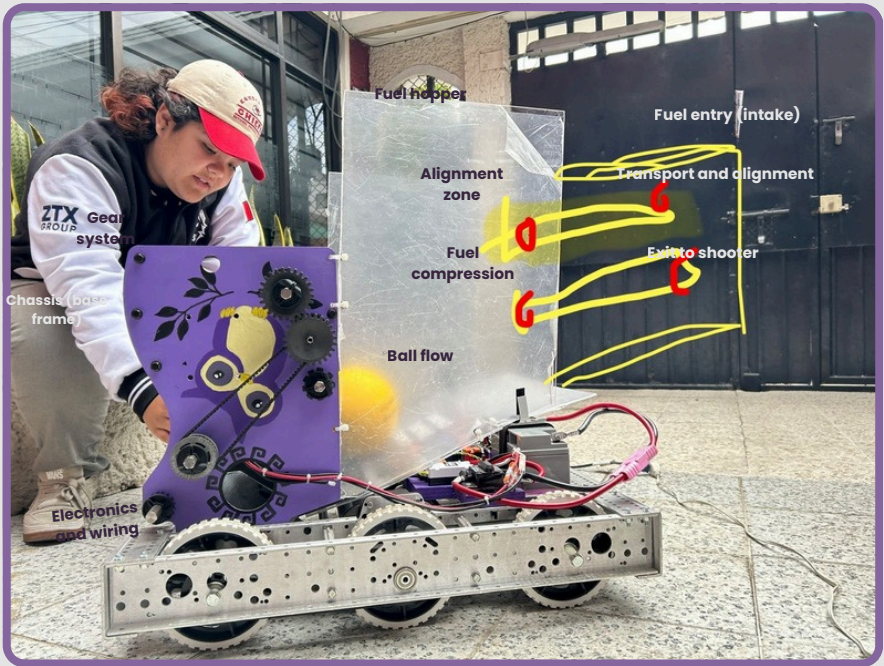
cone wheels to improve alignment

fuel compression tuned with different wheel types

optimized internal ball flow

These improvements increased the robot's speed, consistency and efficiency during the match.

FUEL HANDLING SYSTEM FLOW



**Intake → Base → Rollers →
Shooter → Exit**

We designed the robot around a continuous ball flow, in which each component has a specific role in ensuring alignment, compression and shot consistency.

**“OUR ROBOT WAS NOT
DESIGNED IN A SINGLE
VERSION, BUT THROUGH
ITERATIONS THAT
IMPROVED ITS
PERFORMANCE BASED
ON REAL TESTING.”**

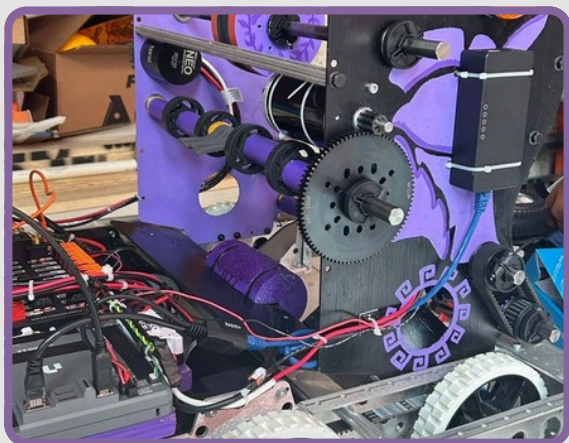
ROBOT ARCHITECTURE

16

04



ROBOT ARCHITECTURE



Overall robot architecture

After defining the team's strategy, the next step was to design the robot's overall structure and decide how to integrate every mechanism needed to meet the game objectives.

We divided the robot into three main systems:

the drivetrain, which moves the robot

the intake and shooter, which handle the fuel

the extendable hopper, which stores more balls

The goal of this architecture was a robot that collects, stores and shoots balls quickly and constantly during a match.

To achieve it, each component was designed with a specific role in the overall system.

Robot flow

We designed the robot around a continuous ball flow, in which each subsystem has a specific role in the process:

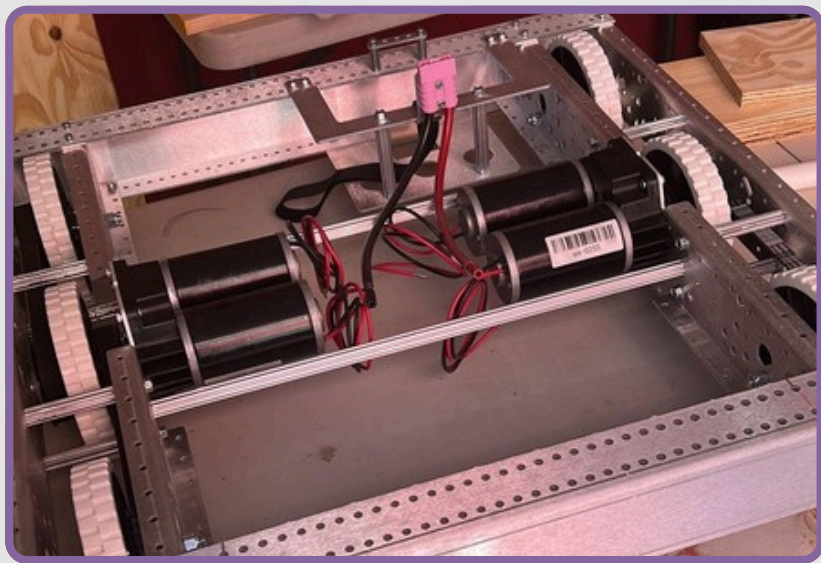
Intake → Intake base → Transport → Shooter → Exit

This flow reduced jams, improved fuel alignment and increased shooting consistency.

Main robot components

The robot is made of several systems that work together on the field. Each was designed to fit into a continuous flow: the drivetrain positions the robot, the intake and shooter move the fuel, and the hopper keeps a constant supply of balls for shooting.

Robot chassis



The chassis is the main structure that holds and supports every component of the robot. It lets the robot move around the field and carry the weight of all the mechanisms.

For the drivetrain we used the tank drive included in the KitBot, powered by four CIM motors. This drivetrain proved very effective at crossing the field's bump without getting stuck, which let the robot carry more fuel without compromising mobility.

Fuel mechanism



The fuel mechanism is one of the most important systems on the robot, since it collects and launches the balls.

It includes:

the intake shafts

the shooter shaft

the shooter hood

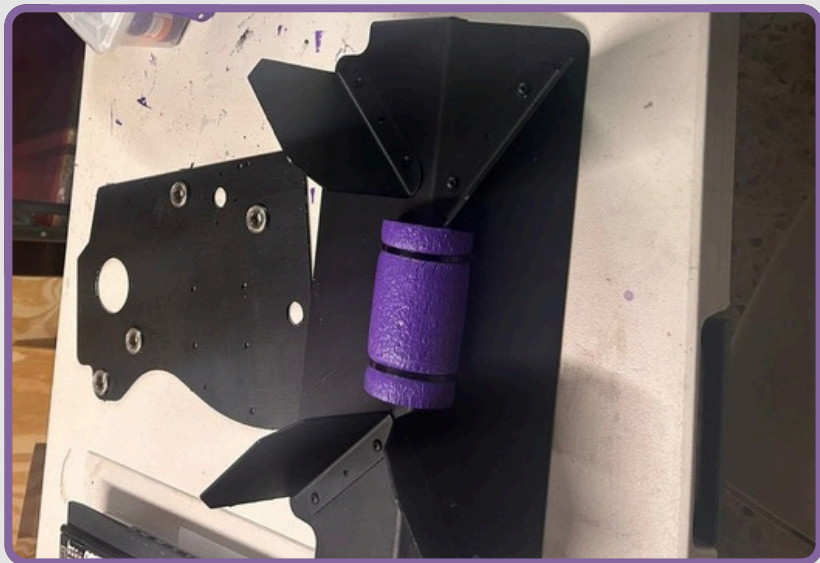
motors

gears and pulleys

All of these components sit between structural plates that keep the shafts aligned and let the system work correctly.

Its main job is to pick balls up from the field and launch them in the right direction toward the target.

Intake base



The intake base sits under the fuel mechanism. It guides the balls as the robot picks them up, preventing jams during intake or shooting, and directs them into the robot and the hopper. During development we adjusted this base to improve ball flow and prevent buildup inside the system.

Hopper



We designed the hopper where the robot stores its balls. It is built from acrylic panels joined into a box, which keeps the balls visible and contained while they travel to the shooter. During the design process we widened the hopper horizontally with additional acrylic panels, increasing the robot's storage capacity. We also developed a mechanism that extends the hopper to hold even more balls during the match.

Bumpers

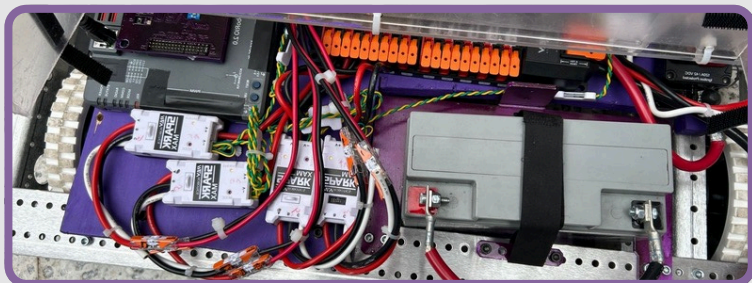


The bumpers protect the robot by absorbing impacts with other robots and field elements.

They are built from sturdy wood boards covered in foam and wrapped in fabric to meet game rules.

This component is essential to protect the robot's structure during matches.

Electronics and wiring

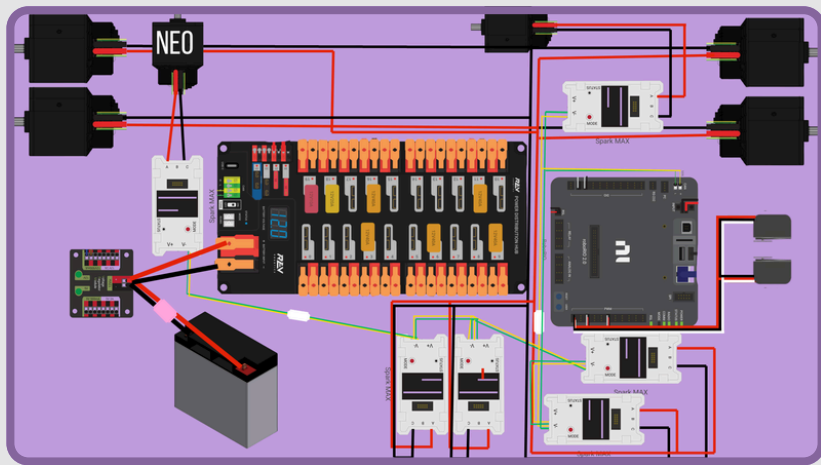


The electrical system controls how the robot operates. It is made up of circuits, sensors, wiring and electronic components that drive the motors and report data from the robot during the match. The main electronic components are:

- roboRIO (main controller)
- Power Distribution Hub (power distribution)
- Radio
- Voltage Regulator Module
- Robot Signal Light
- Spark MAX and Victor SPX motor controllers
- 12 V batteries
- sensors including encoders, a gyroscope and a camera

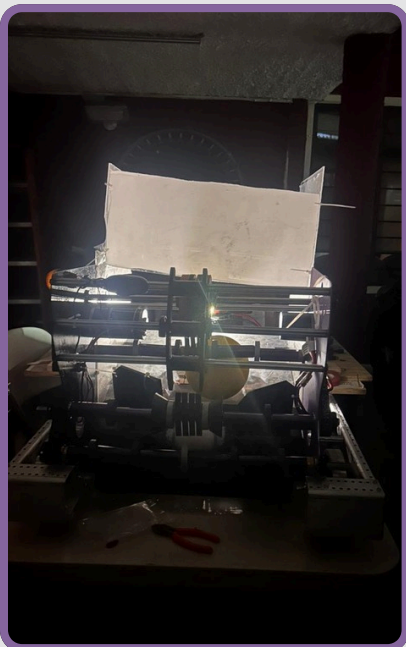
All of these components are mounted on a dedicated board and carefully connected, keeping the system organized and making faults easy to find.

Network topology and electrical architecture



We designed an organized electrical architecture for efficient power distribution and communication between the robot's components. The battery feeds the system through the main breaker, which supplies the Power Distribution Hub (PDH). From there, power goes to the motor controllers and the rest of the electronics. For communication we use CAN bus, connecting the roboRIO with the motor controllers and sensors for coordinated control of every subsystem. This organization kept the system tidy, made diagnostics easier and ensured reliable operation during testing and competition.

Design decisions



Most of the robot's main mechanisms were required to complete the game tasks.

We also developed additional mechanisms to improve performance, such as the hopper extension that lets the robot store more balls during the match.

Every component was included for a specific purpose in the team's strategy, so the robot could collect, store and shoot fuel as efficiently as possible.

Throughout development, the team made key decisions based on expected performance and feasibility.

One of the most important was keeping the KitBot tank drivetrain.

We chose it over more complex alternatives because it offered greater reliability, stability and ease of implementation without compromising performance.

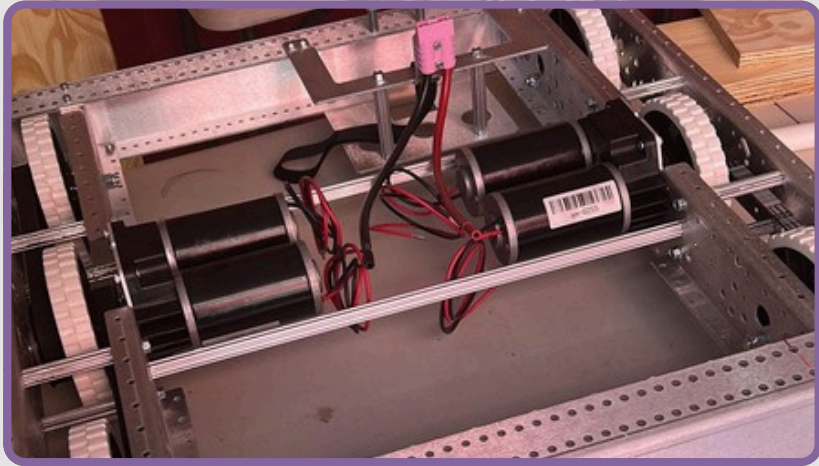
**“THE ROBOT
ARCHITECTURE WAS
DESIGNED TO MAXIMIZE
THE EFFICIENCY OF THE
BALL FLOW, ENSURING
EVERY SUBSYSTEM
CONTRIBUTES DURING
THE MATCH.”**

ROBOT SUBSYSTEMS

05



ROBOT SUBSYSTEMS



We divided the robot into independent subsystems to optimize the design, simplify iteration and make sure each component had a specific role in the fuel flow. This let us design, build and improve each mechanism more clearly, understanding how it interacts with the others.

The main subsystems are the drivebase, which moves the robot; the intake, which picks balls up from the floor; the feeder, which carries fuel inside the robot; the shooter, which launches the balls at the target; and the extendable hopper, which stores more balls during the match.

Each subsystem was designed around the team's strategy, the speed the game demands and the need to handle many balls without jamming. During development, these systems were tested and tuned repeatedly to improve performance and make sure they worked as one integrated robot.

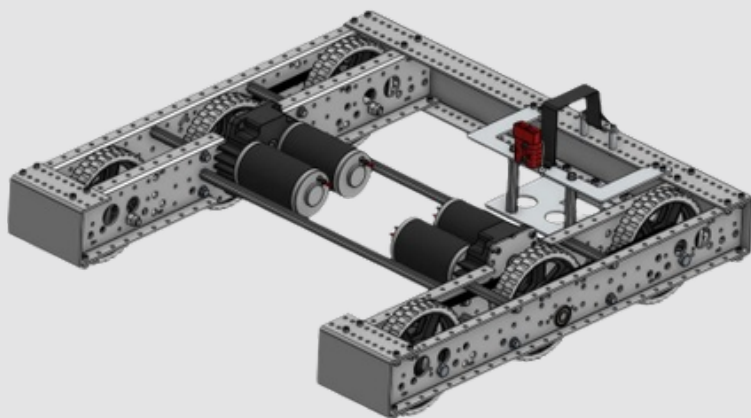
The main goal was a robot that could collect, store and shoot fuel quickly and consistently, a key part of our strategy for REBUILT.

Drivebase

Objective: stable, reliable movement across the field, including over the bump.

Trade-off: reliability over advanced maneuverability; more complex systems like swerve drive were ruled out.

Justification: this design fit our strategy best, letting us focus on fuel handling without compromising mobility.



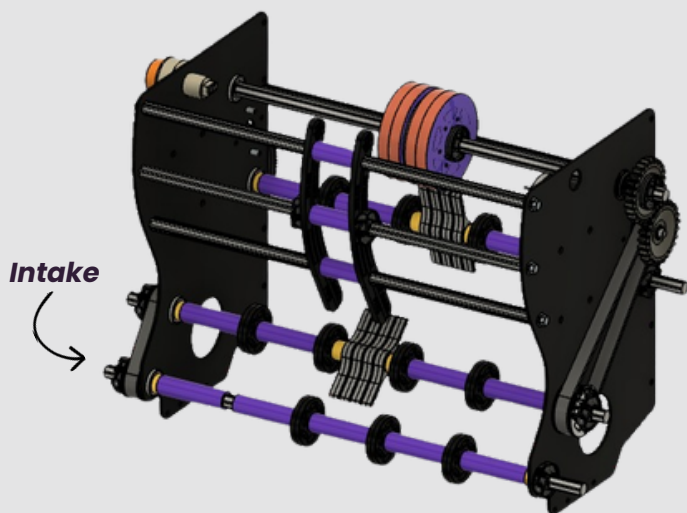
The drivebase uses the tank drivetrain included in the KitBot. Four CIM motors give the robot enough power to move around the field. We chose this drivetrain because it is reliable and effective at crossing the field's bump. That mattered for our strategy, since the robot needed to move freely while carrying a hopper full of balls. The tank drivetrain also gives drivers good control in teleop, making it easier to line up with the target before shooting.

Intake

Objective: collect balls quickly and without jams.

Trade-off: simplicity and continuous flow over more complex collection mechanisms.

Justification: compliant wheels handle fuel efficiently and reduce intake errors.



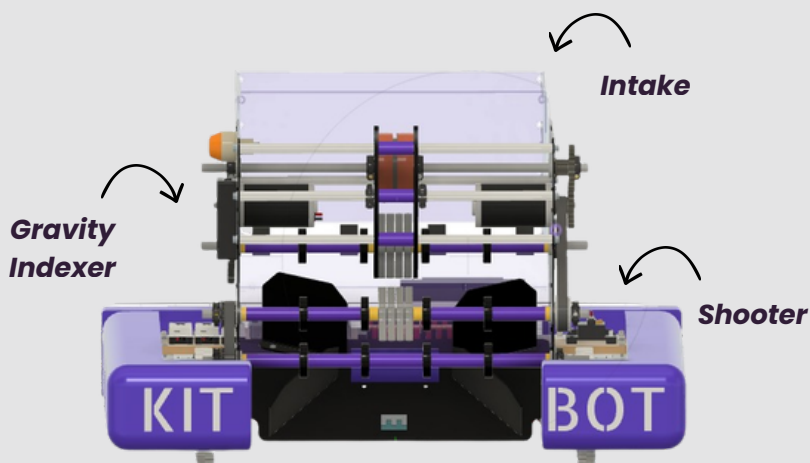
The intake picks balls up from the field and directs them into the robot. It uses several shafts with compliant wheels that grip the balls without damaging them and pull them inside. These wheels also keep balls from getting caught on the sides of the mechanism. During development we adjusted the wheel layout several times to improve ball flow. On the lower intake shaft, white compliant wheels were placed on the sides so balls would not catch on the edges and would be guided toward the center. This design let the robot collect balls much more efficiently.

Feeder (gravity indexer)

Objective: a continuous, natural flow of balls to the shooter using gravity.

Trade-off: we gave up full active control of the flow for a simpler, lighter and more reliable system.

Justification: lower mechanical complexity and fewer failures, with a more stable, jam-resistant flow during the match.



The robot's feeder is a gravity indexer that passively moves balls from the hopper to the shooter.

Unlike fully active systems with many rollers, this design uses gravity and the hopper's incline to guide the balls toward the shooter. The rollers mainly keep the balls moving and prevent jams; they are not the only means of transport.

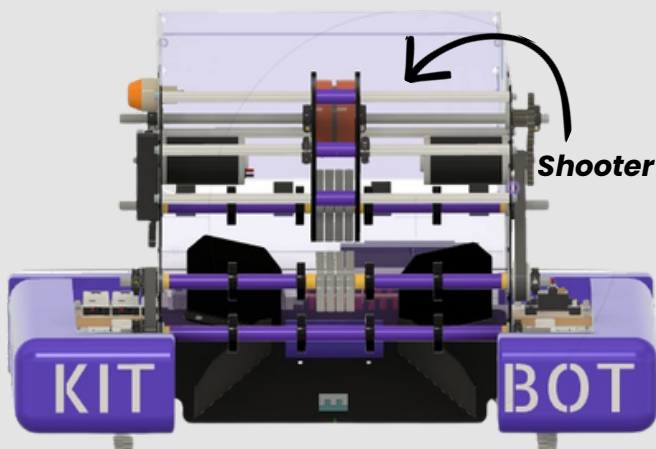
Early tests showed some balls reaching the shooter misaligned, causing jams or inconsistent shots. To fix this we adjusted the system's geometry, realigned the shafts and added elements to center the ball flow.

Shooter

Objective: fast, consistent shots.

Trade-off: consistency over extreme precision.

Justification: mixing compliant-wheel durometers and adding cone wheels improved fuel compression and alignment, increasing the success rate.



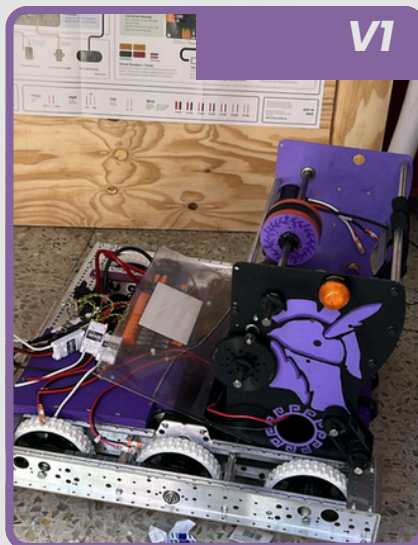
The shooter is one of the robot's most important mechanisms, since it launches the balls at the target.

A motor-driven shooter shaft gives the balls their exit speed, and the compression from the compliant wheels adds velocity to each shot. Testing showed that fuel did not always reach the shooter aligned. To fix this we added cone wheels, which steer the balls to the center of the shooter.

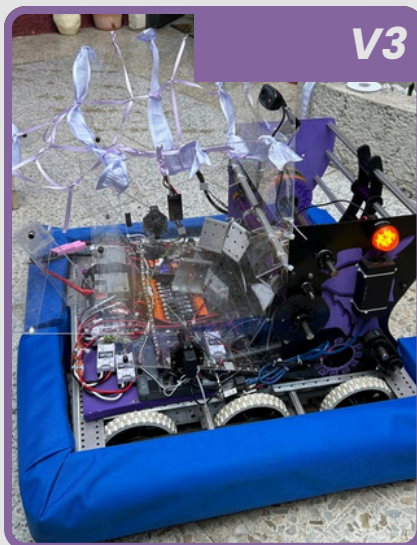
We also experimented with compliant wheels of different durometers: softer white wheels and firmer black wheels, placed in different positions to achieve the right compression on the balls.

These changes improved shot consistency and increased the robot's rate of fire.

Post-test improvements



V1



V3

The robot's first tests revealed several problems in the ball-handling system.

The main one was that balls jammed on the intake base or reached the shooter misaligned. We made several design changes to solve it.

We added a stop on one of the intake guides, made from leftover chassis material, which kept balls from building up inside the system.

We also added small cone-shaped wheels to steer the balls toward the shooter.

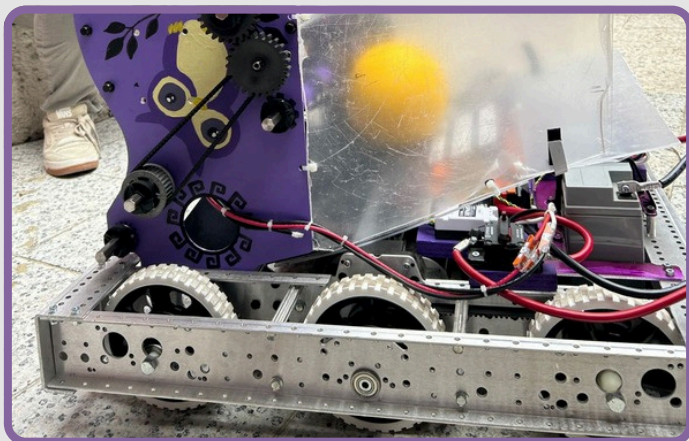
With these changes the robot shot with greater consistency and speed, jammed far less and performed better in testing.

PROGRAMMING

06



PROGRAMMING



Software architecture

We designed the software architecture to support a strategy built on speed and consistency in fuel handling, so every subsystem responds in coordination during the match.

The roboRIO is the robot's brain: it receives driver and sensor inputs and sends commands to the motor controllers (Spark MAX and Victor SPX) over CAN bus.

We assigned motors according to each subsystem's needs:

4 CIM motors for the drivetrain

1 NEO motor for intake and outtake

1 CIM motor for the feeder

2 NEO motors for the hopper extension

We chose CAN bus over simpler wiring because it gives better control, monitoring and synchronization between motors, which a continuous, error-free ball flow requires.

Our software doesn't just control the robot; it turns it into a coordinated system that executes the team's strategy.

Autonomous mode

We built an autonomous mode focused on simple but reliable actions, prioritizing consistency over complexity.

The robot has several autonomous routines for different starting positions: center, left and right, each collecting and shooting fuel according to the alliance strategy.

One of the right-side routines uses fuel supplied by the human player, increasing the number of balls available during this phase of the match.

More simplicity → lower risk of failure

Less complexity → fewer potential autonomous points

In autonomous, we chose to secure consistent points rather than risk everything for more.

Teleoperated mode

We split control of the robot between two operators to maximize efficiency:

Operator 1: driving (drivetrain)

Operator 2: intake, feeder and shooter

This let the robot collect and shoot at the same time, in line with our speed strategy.

During testing we tuned control sensitivity to improve:

movement precision

shot control

synchronization between mechanisms

If the shooter is active → the feeder runs automatically

If the intake detects balls → it stays on until the hopper is full

This kept a continuous flow without depending entirely on the operator.

The controls are designed to reduce human error and keep a steady pace of play.

Sensors

The robot uses several sensors to gather information about how it is running and to improve control of its mechanisms.



We integrated sensors not only to measure data, but to make real decisions while the robot runs.

Sensors used:

Encoders → motor speed

Gyroscope → robot heading

Camera → visual reference of the field

Real applications:

Encoders → hold shooter speed for consistency

Gyroscope → keep the robot stable while driving

Camera → help align with the target

Localization

We used encoder and gyroscope data to estimate the robot's movement on the field.

This lets us:

understand displacement

improve control

adjust behavior in real time

Although we did not implement full field localization, this data was enough to improve the robot's stability.

Testing and debugging

The robot's first tests revealed some problems with the code and mechanism control.

One was that the upper intake and feeder shafts spun in opposite directions, so the system could not handle the balls correctly.

We fixed this by correcting the motor directions in code.

We also tuned shooting speed and power so the balls left faster and more consistently.

After these adjustments, the robot handled and shot balls much more efficiently during testing.

**“OUR CODE CONNECTS
EVERY SUBSYSTEM
TOWARD ONE GOAL:
PLAY FASTER, MORE
CONSISTENTLY AND
WITH FEWER ERRORS.”**

TESTING & RESULTS

07



TESTING & RESULTS



Test procedures

We structured our tests to validate the robot's real performance against our strategy: speed and consistency in fuel handling.

First we verified the electrical system on a 12 V battery and confirmed through their LED indicators that every controller (Spark MAX and Victor SPX) responded correctly. This confirmed proper communication with the roboRIO.

Next, we split the controls between two operators so we could test the robot in match-like conditions.

Testing covered three main areas:

Movement (drivebase): stability, control and crossing the bump

Ball flow: intake, feeder and hopper

Shooting: speed, direction and consistency

We also ran repeated shooting trials to measure:

success rate

exit velocity

rate of fire

**“WE DIDN'T JUST TEST
WHETHER THE ROBOT
WORKED. WE TESTED
HOW WELL IT EXECUTED
OUR STRATEGY.”**

Results and performance analysis



In the robot's first tests (V1), we found that the main problem was not a lack of power but a lack of control over ball flow. Fuel entered the shooter misaligned, causing frequent jams and inconsistent trajectories. We also saw balls building up on the intake base, interrupting the flow to the shooter. In the code, the upper intake and feeder shafts were spinning in opposite directions, directly hurting system synchronization. Finally, our analysis of the shots showed that wheel compression was too low, reducing both speed and consistency.

Based on these results, in the second iteration (V2) we redesigned internal fuel handling instead of simply adding shooter power. We added cone wheels to the upper intake and feeder to force the balls to the center, significantly reducing shooter entry errors.

We also redistributed the compliant wheels, combining soft (white) and firm (black) wheels for progressive compression. This transferred more energy to the ball without causing blockages.

In parallel, we added a stop on the intake base made from chassis material to prevent buildup and keep the flow continuous. Finally, we corrected the code to synchronize the feeder shafts, so every ball moved in the same direction.

In the third iteration (V3), we refined the complete system by integrating these changes, which clearly improved the robot's overall performance. Jams dropped significantly, shooting speed increased and ball trajectories became more consistent.

As a result, the robot went from an unstable system with constant flow losses to a mechanism that shoots continuously and under control, fully aligned with our strategy of maximizing points through volume and speed.

Intake → Feeder → Hopper → Shooter

Ball-handling system iterations (V1 → V3)

V1: Initial system (problems found)

In the first version, the ball-handling system had several faults that directly hurt performance.

Fuel entered the shooter misaligned, causing jams and inconsistent shots. Balls also tended to build up on the intake base, interrupting the flow inside the robot. On top of that, a code error made the upper intake and feeder shafts spin in opposite directions, breaking system synchronization.

As a result, the robot had low shooting consistency and constant flow losses.

V2: Alignment and flow adjustments

Based on these problems, we focused on controlling ball flow inside the robot instead of simply adding shooter power.

We added cone wheels to the upper intake and feeder to steer the balls to the center, and redesigned the compliant-wheel layout, combining soft and firm wheels for better compression.

We also added a stop on the intake base to prevent buildup and corrected the motor directions in code.

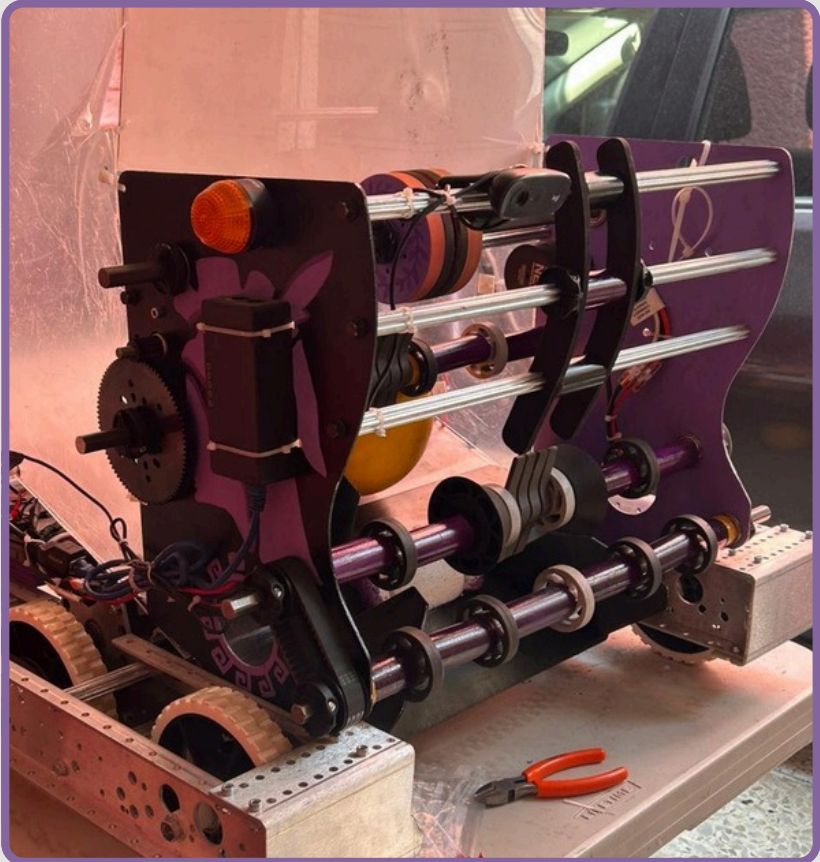
These changes improved fuel alignment and reduced jams.

V3: Optimized system

In the third iteration, we refined the integration of every change.

Ball flow inside the robot became continuous, with no buildup and cleaner entry to the shooter. The wheel compression produced more consistent, faster shots.

As a result, the robot shot balls continuously, with fewer errors and greater efficiency during testing.



**“THE ROBOT
DIDN'T GET
BETTER BY
BECOMING
STRONGER, BUT
BY BECOMING
SMARTER.”**

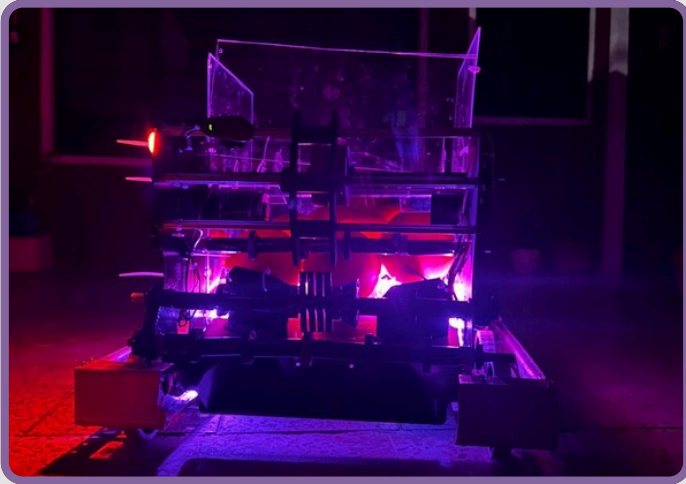
LESSONS LEARNED

47

08

A large, stylized white graphic on a purple background, resembling a bird's head or a wing in profile, facing right. The graphic is composed of several overlapping, curved shapes that create a sense of movement and form.

LESSONS LEARNED



Main challenges

Throughout the robot's development we faced challenges that forced us to make engineering decisions based on real testing, not assumptions.

One of the most important was achieving controlled ball flow inside the robot. In the first iteration (V1), fuel reached the shooter misaligned, causing jams and inconsistent shots. We chose to prioritize flow control over more power, which led us to redesign the internals with cone wheels and a better compliant-wheel layout.

Mechanically, assembling the fuel mechanism plates was a critical challenge. These plates had to keep several shafts (intake, feeder and shooter) aligned, so any measurement error affected the whole system. This forced us to iterate the design (V2) and adjust spacers and supports until the alignment worked.

We also faced material limits, such as PVC spacers that did not match the required dimensions. Instead of replacing them entirely, we adjusted and adapted them, prioritizing functional solutions with the resources we had.

In electronics and programming, we found errors that directly hurt performance, such as the reversed rotation of the upper intake and feeder shafts. This reinforced the importance of validating not only the physical design but also the system's logic.

Engineering lessons

Throughout the process we learned that designing a robot is not just building mechanisms, but making decisions with intent.

One of our most important lessons was that every component needs a clear purpose in the system. It was never about adding parts, but about understanding how each decision affected flow, speed and consistency. The cone wheels, for example, were not just a mechanical tweak; they were a deliberate decision to control fuel alignment.

We also learned the value of iteration. Instead of chasing a perfect solution from the start, we worked in versions (V1, V2, V3), and every test gave us data to improve. This took us from an unstable system to a consistent, competitive one.

Another key lesson was the importance of prototyping. Simple models, like the cardboard prototype of the extendable hopper, let us validate ideas before investing time and resources in the final build.

Finally, we learned that testing is a central part of design. Many of our most important decisions came from watching the robot fail in real tests.



The team also learned the importance of testing the robot constantly. Many of our improvements came after seeing problems in testing, such as ball jams or the need for faster shooting. This cycle of testing, analysis and improvement turned the robot into a more efficient and reliable version of itself.

Future improvements

If we continued developing this robot, we would focus on further optimizing the current system rather than changing the design completely.

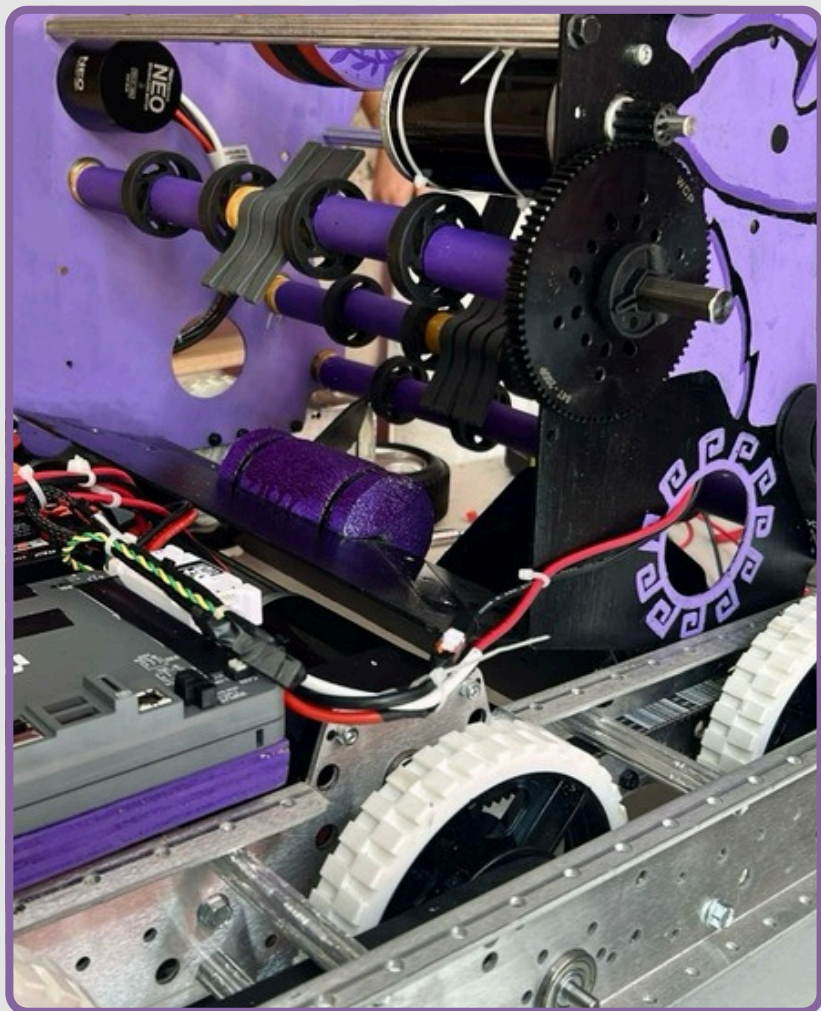
We would refine ball handling by tuning compression and alignment more precisely, to increase shot consistency in different game conditions. We would also improve the extendable hopper, aiming for a more space-efficient design with better internal flow control.

In programming, a key improvement would be more precise speed control and synchronization between subsystems, making better use of sensor data so the robot can adapt during the match.

These improvements would not change the robot's essence; they would strengthen decisions that have already proven effective.



Building this robot showed us that engineering does not depend on prior experience, but on the ability to analyze, iterate and keep improving. As a fully rookie team, we chose to build a functional robot while understanding every part of it and the reason behind every decision. Beyond the final result, this process proved that we can do real engineering, solve complex problems and compete with well-founded solutions.



“AS A ROOKIE, ALL-FEMALE TEAM, WE SHOWED THAT ENGINEERING DOESN'T DEPEND ON PRIOR EXPERIENCE, BUT ON THE ABILITY TO LEARN, ITERATE AND MAKE DECISIONS.”