2024/5 SEASON

MISHMASH

12016

Made by MISHMASH | Enhanced by AI

Introduction

In a world that too often suppresses individuality, we reject conformity. We are **MISHMASH 12016** emerging from the Jusidman Science Center for Youth in Beer Sheva resilient by nature, fueled by unwavering determination. We feed off our failures, using them as the fuel to climb higher, build better, and grow stronger.

MISHMASH represents more than just a name, it embodies a mindset. We view failure as a source of strength, understanding that each setback brings us closer to progress. Like an asymptote nearing infinity, we continuously push our limits refusing to settle, never content, and always aspiring for more.

YOU ARE YOUR ONLY LIMIT

Core Values: The DNA of MISHMASH

We're a synergy of people mentors, alumni, volunteers, and FIRST service-year members each bringing something different, but working as one team. We come together with one goal and one mindset. That's what makes us MISHMASH.

- **Resilience isn't a trait, it's a choice** - High-stake moments sharpen our thinking. They push us to extremes, challenge our limits, and force us to adapt under pressure. That's where real growth happens and where resilience is built, not taught.
- **Strategic Trial and Error** - Failure isn't a setback. It's part of the process. We feed off our mistakes, analyze what went wrong, and come back stronger.
- **You Are Your Only Limit** - External barriers are nothing compared to the ones we place on ourselves. The real work begins when we choose to confront those limits and keep going anyway. Limits don't disappear we just stop letting them decide for us.

This mindset follows us, even when we leave the workshop.

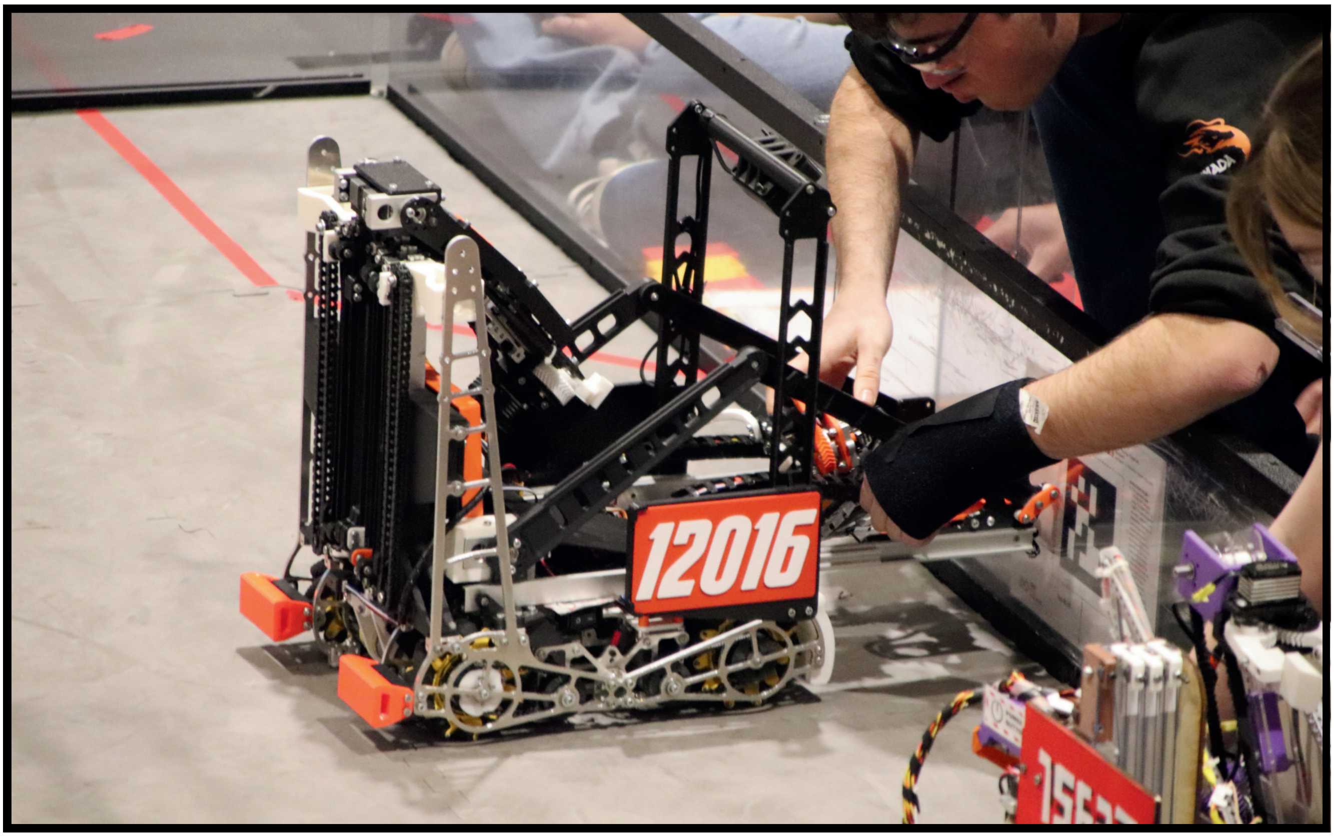
Season Goals

Our goal is simple - **To become a world-class team**.

Not just in performance but in how we think, how we work, and how we lead.

We're building a robot that can compete at the highest level, with the precision and reliability it takes to become a world-class robot.

But along the way, we're also learning what it really means to think like MISHMASH to challenge limits, and push ourselves beyond what we thought possible.



MISHMASH12016

Sustainability Plan

Sustainability at MISHMASH isn't about maintaining what we have it's about making sure we never stop growing.

We're not here to survive seasons. We're here to build something that lasts.

That means building systems that outlive individuals, and habits that carry across generations of members.

This is how MISHMASH stays alive: through constant motion, shared knowledge, and a team culture that refuses to stand still.

Team Development & Growth

Unlike school-based teams with a built-in pool of students, we don't inherit teammates—we go out and find them.

Firstly, we recruit actively, not passively. Through school outreach, open houses, social media, and word-of-mouth, we search for individuals with potential and the mindset to thrive in a culture built on accountability, pressure, and growth.

Secondly, this season we expanded that culture by developing two parallel robots—opening new leadership roles and giving more members the chance to lead, build, and shape the team's technical direction.

Lastly, we strive to ensure that knowledge doesn't slip through the cracks. Our goal is to preserve every line of code, every CAD file, and every process so future generations of MISHMASH won't have to start from scratch, but from where we left off, and take it further.

AI-Driven Optimization

Artificial Intelligence has become essential to our operational strategy accelerating decision-making and improving efficiency across the board. From scouting automation to content development, AI allows us to analyze, adapt, and scale faster than ever. This portfolio itself is the result of iterative collaboration with AI, helping us refine our tone, sharpen messaging, and stay aligned with the MISHMASH mentality.

In the past, our scouting relied on data manually collected by team members. Over time, we found that this information wasn't always reliable or consistent, prompting us to search for more accurate and efficient ways to assess team performance. One solution we adopted was Estimated Points Added (EPA). In previous seasons, calculating EPA required significant manual setup and spreadsheet work. This year, we streamlined the process by leveraging AI to generate a more robust, data-driven system. This year, we streamlined the process by leveraging AI to generate a more robust, data-driven approach. The result: a scalable system that tracks performance trends, predicts alliance value, and supports real-time strategic decisions.

These tools don't replace our thinking they extend it. They reflect the way we work: always iterating, always refining. We're not just embedding knowledge we're building systems that evolve with us and turn experience into a competitive edge.

MISHMASH12016

Budget Management

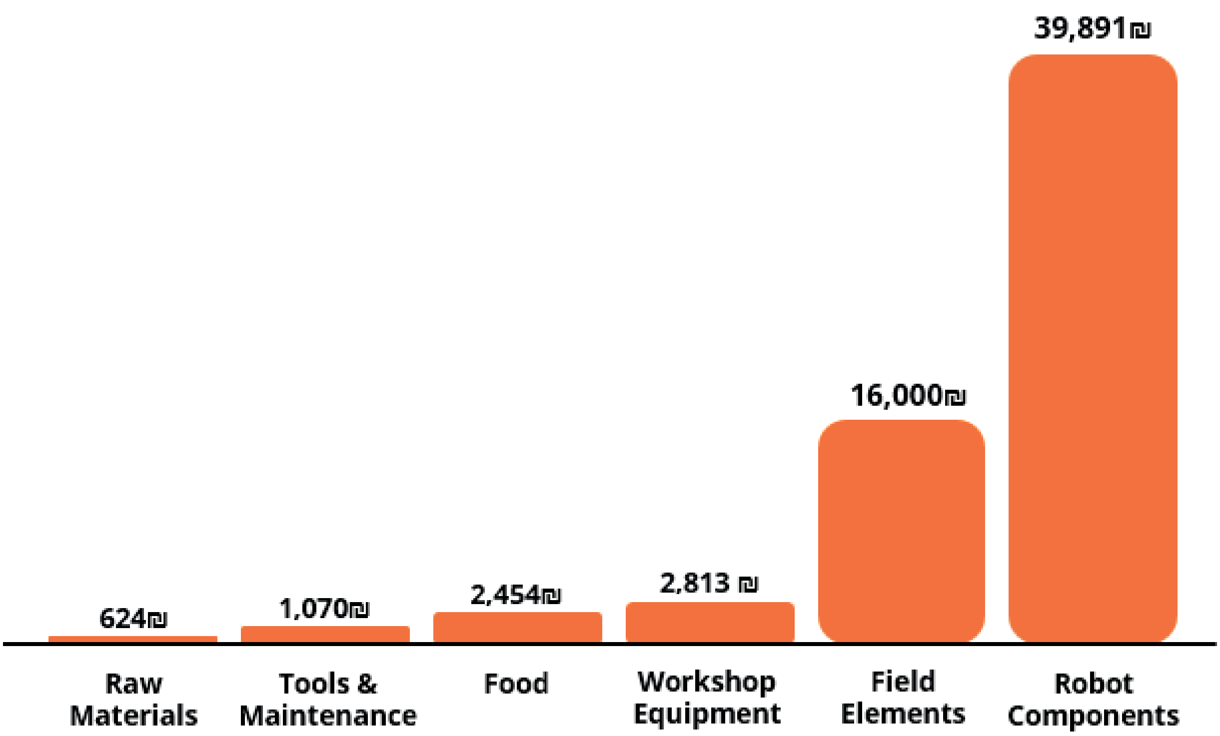
We manage our own budget fully and independently.
Every financial decision goes through us. We calculate the pros and cons, weigh impact vs. cost, and choose where every shekel goes. This budget is made up not only of a base allocation, but also of revenue we've earned—through our own efforts, as detailed later in this portfolio.

This year, we made two major decisions:

- **3D Printing Expansion -** We purchased three additional 3D printers not as a luxury, but as a step forward. They power rapid prototyping, support our Tech Research program, and help us iterate faster than ever.
- **Workshop Upgrades -** Upgrading our tools wasn't a bonus it was essential. With better fabrication capabilities, we're able to build more precisely and work more efficiently. We also collaborate with other teams to reduce costs and share resources—maximizing what we have and making every shekel count.

Total Revenue vs. Expenses (2024/5)

Base Budget:	60,000₪
Revenue:	19,250₪
Expenses:	62,852₪



Education & Knowledge Sharing

We're working towards making knowledge something that doesn't just get passed down—but built to last. We're developing a system to ensure that expertise doesn't leave with graduating members. Instead, it's preserved, organized, and expanded so future generations can build on what came before, not start from scratch. It's still a work in progress, but one we're committed to shaping and improving each season.

Alumni and FIRST Shinshin have played a key role in reinforcing our programming subteam, bridging gaps left by departing veterans. But mentorship alone wasn't enough. We needed lasting infrastructure to safeguard what we've learned.
Last year taught us a lot. We realized that having one person carry most of the programming workload even if effective in the short term left the team vulnerable in the long run. That insight shaped how we approached this season not only in programming, but across the entire team.
We didn't just aim to avoid repeating mistakes. We rebuilt the system.

We're using our programming subteam as a case study because it reflects how we think about sustainability: we analyze past gaps, build systems around them, and grow stronger with every cycle. At the heart of this shift were two key elements: **MMlib**, and a new workflow method we call **The Splitting**.

To prepare the programming subteam for independence, our senior programmer led an off-season training program aimed at onboarding new members and reducing dependency. But the more he supported them, the more they leaned on him. Instead of building confidence, the system unintentionally reinforced reliance something that clashed with our team's core: learning through doing.

MMlib
In parallel, he began building **MMlib -** a custom software library designed to be both a launchpad and a safety net.
Built on top of FTCLib and CuttlefishFTCBridge by Team **Roboctopi**, MMlib includes essential systems like PID control, drivetrain code, and utilities designed for clarity, speed, and adaptability.
It shortens the time needed to get a robot moving, freeing programmers to focus on innovation instead of boilerplate. But its real power lies in how it teaches: structure, readability, and embedded logic turn the codebase into a hands-on learning environment.
MMlib wasn't a patch. It was a plan for independence, scalability, and long-term growth.

The Splitting
At the start of the season, we introduced a new workflow method we now call The Splitting. Tasks were deliberately divided and assigned to new members with minimal guidance, encouraging them to problem-solve independently using available resources. This shift transformed our programming culture from reliant to resilient.

Open-Source Contributions
We believe that knowledge grows when it's shared. We actively release our software tools, automation algorithms, and engineering solutions to the wider robotics community. Through documentation, mentorship, and open discussion, we help others implement our systems and we learn from them in return

Collaboration with Other Teams

We believe in raising the bar not just for ourselves, but for the entire community. When the level of competition rises, we rise with it. That’s why we’re committed to supporting other teams sharing knowledge, offering help, and building strong partnerships.

Collaboration, for us, is a strategic strength. A great robot is only part of the equation—working alongside capable, motivated teams pushes all of us further.

We mentor **Fata Morgana** (#15637), providing ongoing support in strategy, engineering, and team operations. When the team was on the verge of shutting down this season, we stepped in to help revive them offering guidance, encouragement, and hands-on assistance. Throughout the season, they became a weekly presence in our workspace, working closely with us as they rebuilt their momentum.

As the ISR Championship approached, it became clear that preparation had to extend beyond our own team. With alliance synergy and high-level gameplay becoming increasingly decisive during playoffs, we focused not only on building a competitive robot but on cultivating strong, field-ready partnerships.

To meet this goal, we launched a three-day intensive training bootcamp—designed as a shared platform for technical growth and alliance development.

We invited top-performing teams from our division, including **Barnyard** (#22993), **BTJ** (#13452), and **Orange Fox** (#12363), and structured the camp around three core focus areas:

- **Robot Upgrades** – Hands-on sessions to test, iterate, and refine subsystems and autonomous routines, with real-time feedback and joint problem solving.
- **Match Strategy** – Simulated matches, daily drive practice, and mock alliance reviews to improve communication, coordination, and in-game adaptability.
- **Cross-Team Collaboration** – Open sharing of tools, code, and insights turned competition into true cooperation.

The momentum didn’t stop there. In the days that followed, we hosted a focused demo competition at our workshop with **Atomix** (#11443), **SciFighters** (#16473), **Desert Machinery** (#21850), and **Fata Morgana** (#15637). Structured practice matches simulated real-event pressure, giving teams space to test autonomous paths, refine coordination, and experiment with strategy in a high-intensity, low-stakes environment.

Establishing MISHMASH as the Engineering Authority

We may have inherited our physical space at the Jusidman Science Center, but what we've built inside it is unmistakably ours. Room 308 began as an overlooked lab in a building filled with constant movement: visiting schools, educational programs, and a stream of untapped potential. Where others saw a quiet room, we saw leverage.

We developed a focused, long-term plan to convert this idle space into a fully operational base for robotics, engineering, and technical leadership. Our objective was never to participate—it was to become integral. We started by designing and teaching our engineering courses, transforming Room 308 into a high-functioning, student-driven workspace. Then we scaled—launching an FLL team, mentoring younger students, and hosting robotics sessions that injected consistent, high-value activity into the space.

To embed ourselves deeper, we aligned with two of Jusidman's external school programs:

- **Science Days** – A series of hands-on activities introducing students to core engineering concepts through robotics challenges.
- **Tech Research** – A project-based program built around 3D-printed components and experimental lesson plans.

For both, MISHMASH supports the entire process: preparing materials, maintaining equipment, and ensuring technical readiness. In Tech Research, we serve as beta testers, stress-testing new activities, improving lesson plans, and fine-tuning each module before it reaches a classroom.

We are not an extension of the system—we are part of its foundation. Room 308 no longer accommodates MISHMASH—it depends on us. Every organized kit, calibrated printer, and outreach-ready activity begins in our hands.

Our presence doesn't supplement robotics education at Jusidman. It drives it.

ROBORUMBLE

This summer, during the Jusidman Center's summer enrichment program, we developed and launched ROBORUMBLE—a hands-on robotics course for middle schoolers, designed and taught entirely by our team. Inspired by BattleBots, the program focused on building from scratch, solving real problems, and learning through failure—not just following instructions.

Following its success, we brought ROBORUMBLE to the Da Vinci after-school program as well, expanding its reach and impact.

Over 115 students participated, learning to prototype, troubleshoot, and compete with robots they built themselves. Every shekel—almost ₪20,000—went straight back into our team: tools, upgrades, and the resources we need to keep growing.

APRIBOT

We launched APRIBOT this season—our FLL team, entirely created and actively mentored by us. From building their robot to reviewing their code, we guide the team through every stage of the process. This isn't a one-time outreach effort. APRIBOT is our way of building continuity. It gives us a direct path to developing future MISHMASH members—students who already understand how we think, work, and build.

Evolution of Our Season

The Sukkot Sprint - Early Prototyping

This year, we broke the pattern. Instead of our usual post-kickoff prototyping, we set a self-imposed seven-week deadline—ending at Sukkot—to finalize our first full robot. We launched development early, prioritizing speed over reliability. It wasn't about building the final robot—it was about learning faster, and building smarter with every iteration. This head start gave us early insight into the game's core dynamics and strategic demands.

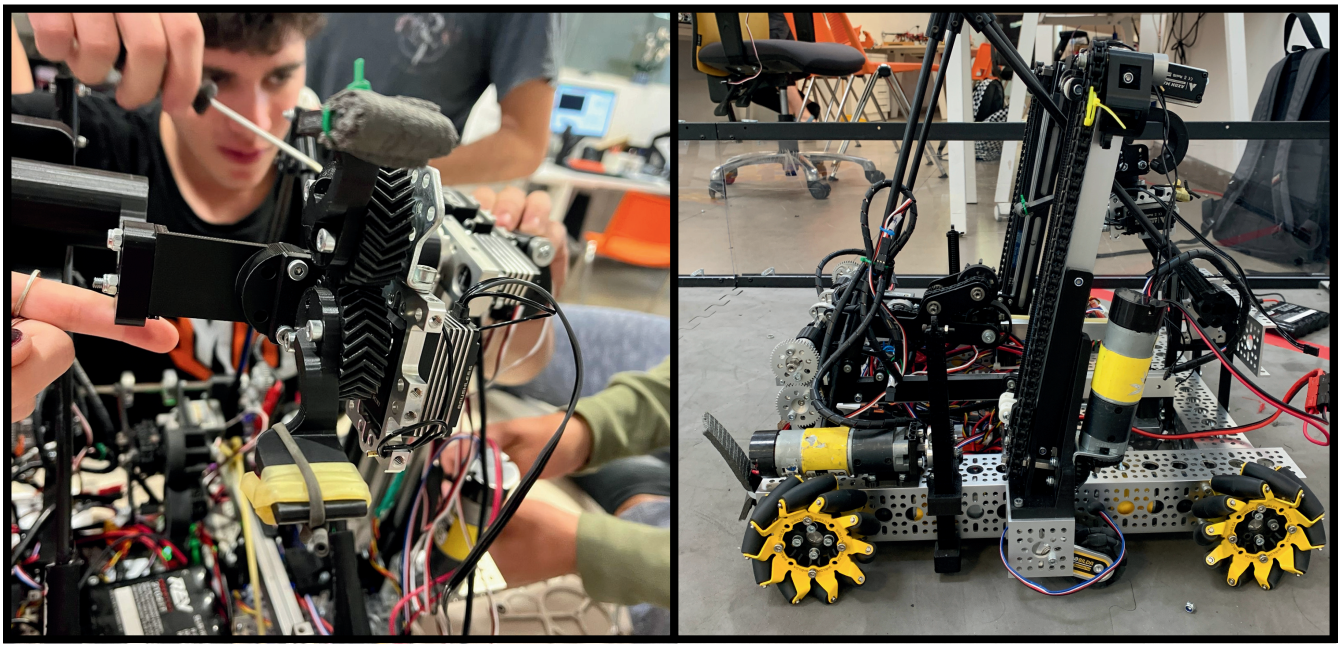
Our initial targets were clear:

- Score the preload and one additional game element during autonomous.
- Execute a consistent seven-cycle Teleop routine.

To support these objectives, we designed a dedicated sprint robot featuring six key systems:

- **Roller Intake System** – A servo-powered pinching roller mechanism gripped samples against a flexible surface for rapid collection.
- **Custom Misumi Elevator**—Built from precision extrusion, this elevator uses dual motors driving 90-degree wheels along the sprocket axis, optimized for kit-style adjustability.
- **Scoring Arm Unit**—A servo claw on an arm designed for stable pinching and controlled release. Its early iteration struggled with gear skipping and hard assembly and disassembly.
- **Linear Intake** – Built from dual Misumi sliders driven by carbon rods. Unstable, difficult to handle, and challenging to assemble.
- **Intake Arm** – An early design with gear-skipping issues and overall instability.
- **Climbing Mechanism** – Elevator-based system with a hook; struggled with power and stability, leading to an overly high gear ratio.

Early momentum created space for controlled iteration — allowing us to surface critical issues before they become season-defining problems.



The Splitting – Challenges & Team Adaptation

We brought in new members to support a full-season build and opened more paths for people to become meaningfully involved. This shift surfaced real challenges: limited resources, uneven experience levels, and breakdowns in communication. We adjusted our workflow, clarified responsibilities, and worked through the friction as it came.

The split between Kanna and Hakoda began immediately after the sprint. Each robot followed a different path with its own goals, team, and workflow.

Kanna was developed by newer team members, continuing to build on the sprint robot from Sukkot and gaining hands-on experience through iteration and refinement.

Hakoda, on the other hand, was assigned to a team of experienced members who took the key lessons learned from the sprint and applied them to a brand-new robot—designed from scratch with a fresh strategic and mechanical approach.

We set a clear deadline: the Early Season Competition (ESC) in late December. Both robots would compete head-to-head. The stronger version would define our direction for the rest of the season.

Teams & Robots Breakdown

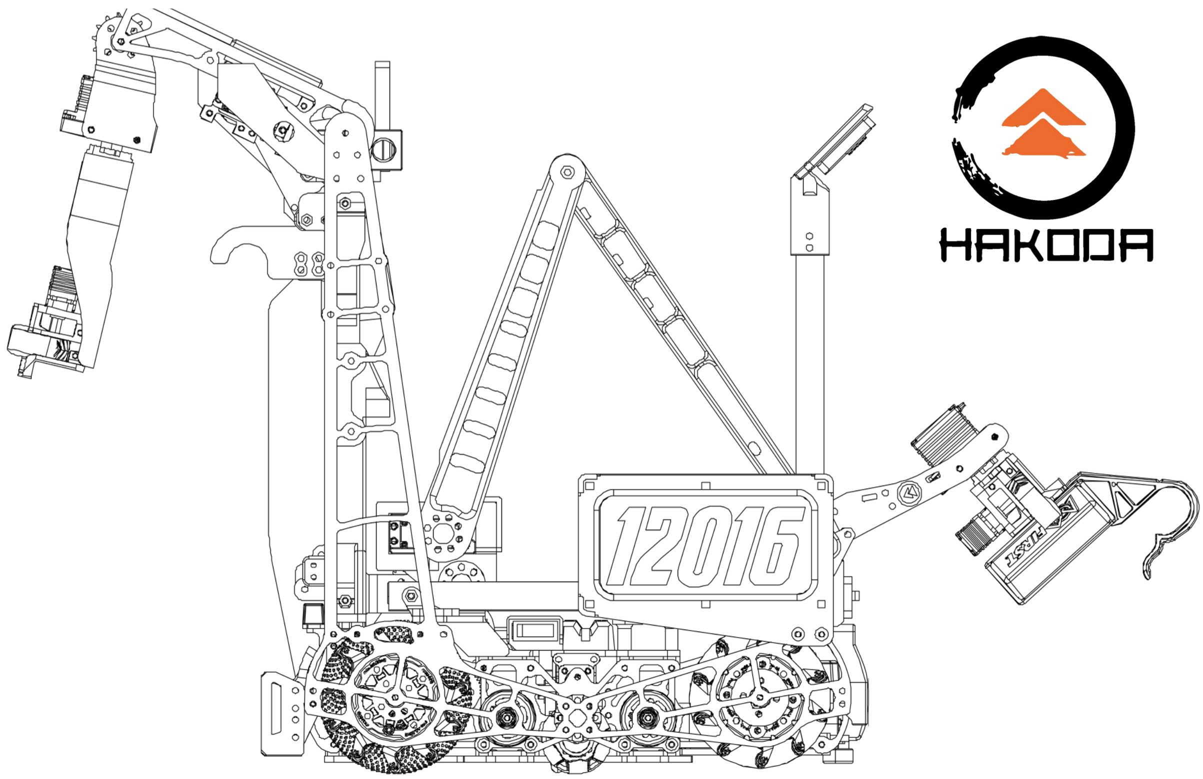
	Kanna	Hakoda
Starting Point	Continued sprint robot framework	Built entirely from scratch
Team Composition	New members or new to subteam roles	Experienced members with subsystem expertise
Focus Areas	Programming and mechanical improvements	Mechanical design, strategy, and full engineering development
Design Philosophy	Improve and stabilize existing concepts	Explore, iterate, and test high-ceiling ideas

The Reunion – Refining Our Strategy

When the competition came, only Kanna was functional. Hakoda stayed behind. But even before the event ended, we knew Hakoda was the robot we wanted to move forward with—it had a stronger engineering base, cleaner systems, and more long-term potential.

Merging the teams wasn't easy. Each side had its codebase, subsystems, and workflow habits. Communication lagged. Integration took effort — but it forced alignment. We re-evaluated our geometry, modeling practices, programming libraries, and code structure.

Meet the Robot: Hakoda



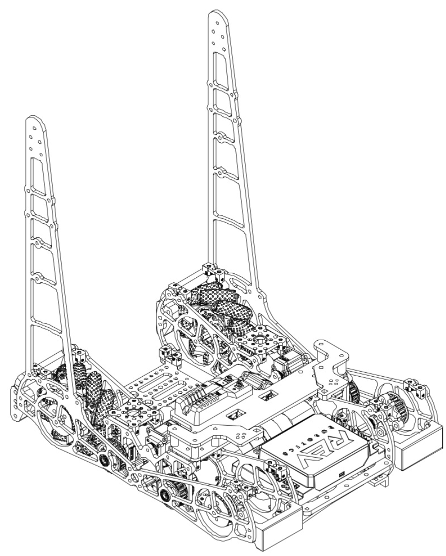
HAKODA

Hakoda is the result of focused iteration and hard decisions.. Our goal was to create a versatile and reliable robot, capable of handling various tasks efficiently and precisely. Every mechanism reflects how we think: sharp, intentional, and never settling. Below is a breakdown of our key mechanisms, explaining why we built them the way we did and how each system evolved:

Drive Train

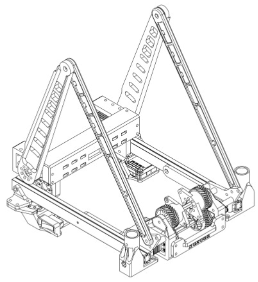
Our drivetrain is a custom mecanum drive in its fourth iteration, powered by four GoBilda motors and utilizing timing belts to drive the wheels. The chassis is constructed from CNC-cut aluminum plates, ensuring precision and durability. This year, we focused on structural integrity and modularity, transitioning to GoBilda components exclusively for all connections.

Additionally, a bottom plate now connects both sides of the drivetrain, enhancing rigidity while keeping the attachment points low to maintain a stable center of gravity. We adjusted the gear ratio to 14:1 to optimize performance, balancing speed and control. This enables us to navigate swiftly while maintaining defensive resilience when necessary.



Linear Intake

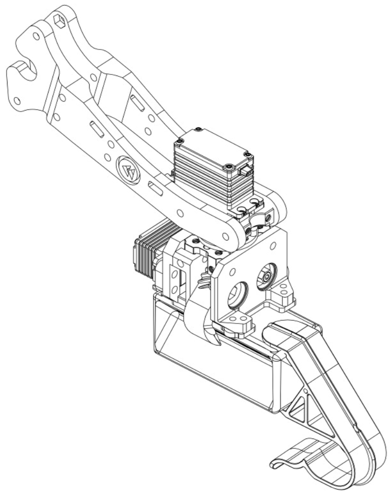
The intake system features two dual Misumi sliders that extend via a linkage mechanism made of 3D-printed rods, powered by two Axon servos. This design has proven reliable, with no significant modifications needed this season. We integrated a **REV Servo Hub to improve efficiency**, enabling faster deployment by operating the servos at a higher voltage.



Intake Mechanism

Our intake mechanism is divided into two parts:

1. **Lifter Mechanism**: Powered by two servos and a herringbone gear-based system connected to linear slides, ensuring precise positioning to accommodate various sample positions.
2. **Intake System**: This system features a rotator and a claw controlled by an additional servo for seamless sample collection; its geometry is explicitly built to access grouped samples.



Gear Slippage: High-speed movements caused the gears to slip. To address this, we replaced them with double helical gears and reinforced the arm's structure by switching to a dead axle. The result is a more robust arm, improving stability and durability.

Ease of Assembly and Maintenance: The original intake system was challenging to assemble. After refining the design, the central axle is now secured with two screws, making disassembly and maintenance easier.

Climbing Mechanism

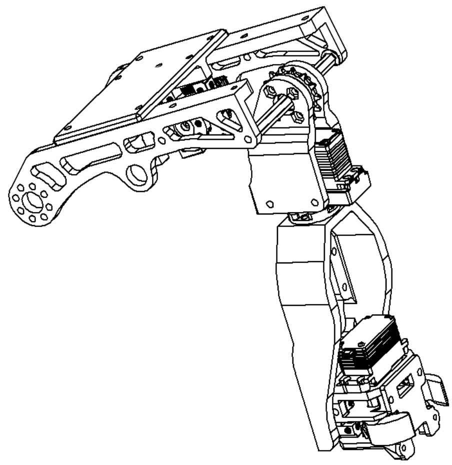
Currently, our climbing mechanism allows us to reach the first stage. However, it is designed for future expansion, with plans for a two-stage upgrade to maximize hanging points. The system includes:

- A hook mounted on the elevator.
- A triangular aluminum structure attached to the drivetrain for added support.
- Two rotating hooks with a ratchet mechanism positioned at the upper edge of the allowed 18-inch frame.

Scoring Unit

Our redesigned scoring system prioritizes clarity, precision, and performance. It addresses three key goals:

1. **Speed & Reliability**- The new “touch-and-go” scoring style allows for faster, more forgiving placements—reducing alignment demands and minimizing handling. This improves both reliability and cycle times under match pressure.
2. **Forward-Facing Scoring**- Scoring from the front enables seamless transitions between scoring and intake. Especially during autonomous, this allows us to complete efficient 1+5 specimen routines without repositioning.



1. **Advanced Articulation-** Mounted on the elevator, the dual-jointed scoring arm—powered by five Axon servos offers precise angle control and adaptability during scoring.
2. **Human Player Integration**- A longer intake arm now pulls specimens directly from the wall into the outtake system, cutting out transfer stages and reducing points of failure.

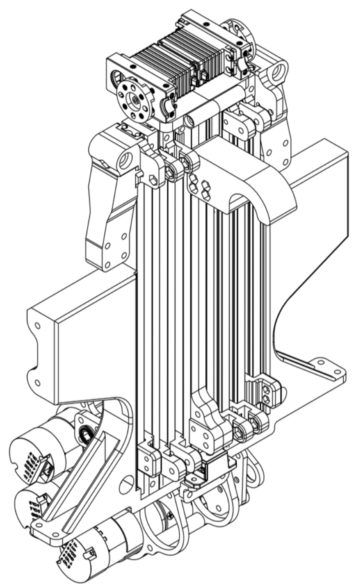
Elevator

Our elevator went through several design cycles focused on power transmission, modularity, and mechanical efficiency.

The final version features a cascade-style lift with G2 belts and **Misumi** sliders, allowing synchronized extension across all stages. It's powered by four 19.7:1 motors mounted at the base to keep the center of gravity low—an intentional choice applied across all eight motors on the robot to maximize stability.

Key Improvements:

- **Power Transmission-** We replaced the original plastic chain mechanism with a gear-driven system, eliminating tension issues and power loss, significantly improving reliability and responsiveness.
- **Climbing Integration-** To support a Level 2 climb, we added a forth motor and adjusted gear ratios for increased torque. We also reinforced the lift structure to handle climbing stress without compromising scoring functionality.
- **Modular Design-** The elevator houses the scoring unit, expansion hub, and all related wiring in a compact, serviceable layout. It mounts to the drivetrain via eight easily accessible screws, enabling fast removal for repairs or iteration.



Programming & Software

Hakoda's programming reflects our intentional, adaptive, and constantly evolving mindset. Every system is designed to be modular, reliable, and fast to iterate on. Here's how we approached the control components on Hakoda:

Foundations

We develop in **Android Studio** using **FTClib**, our **MMlib** for command-based structure, and manage version control through GitHub with Fork. Every mechanical system is reflected by a dedicated software subsystem, giving us a clean, one-to-one structure that's easy to test, update, and scale. A central configuration class handles all constants—motors, servos, directions—so changes are clean and fast.

Autonomous Mode

We build autonomous paths with **RoadRunner v1.0**, powered by dead wheel odometry and **LazyIMU** integration. A library of pre-built routes allows us to adapt each match based on starting position, alliance side, and partner strategy. We use **MeepMeep** to visualize and refine these paths without running the robot, cutting iteration time dramatically and reducing field-time dependency.

TeleOp Mode

TeleOp is built around a command-based architecture. Actions are composed in layered groups—parallel or sequential—allowing fast transitions and tight control. Our codebase separates all mechanisms into individual Subsystems:

- Drivetrain (field-oriented using IMU yaw)
- Scoring Claw & Arm & Elbow
- Vision (limelight camera)
- Intake Arm & Claw
- Linear Intake
- Elevator

Each has a standalone class and command set, making logic modular and easy to maintain. Field-oriented drive converts joystick input relative to the robot's heading, improving driver precision without extra thought.

Key Sensors & Feedback

Clean response comes from clean signals. Hakoda uses layered sensor input to stay reliable throughout the match:

- PID controllers handle the drivetrain and elevator for stability and responsiveness.
- In the Elevator, Motor encoders pair with a magnetic touch sensor for hard resets at full retraction—preventing drift and eliminating manual recalibration.
- Limelight Vision camera provides real-time vision-based targeting, assisting in autonomous navigation, intaking by detecting field elements and aligning the robot accordingly.

Vision – Limelight

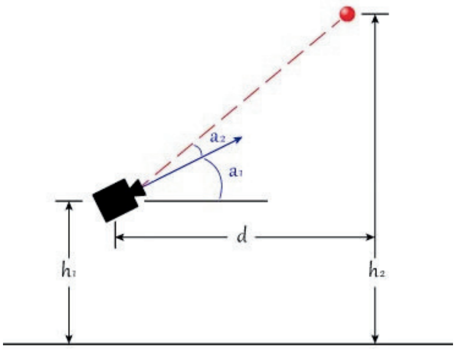
We use a **Limelight** camera running a dynamic **OpenCV** pipeline initially developed as an open-source tool. We've customized the configuration and tuning to meet our strategic needs.

The robot uses this system to detect samples, adjust to the exact position of samples, open the linear intake, and rotate the intake claw to the position of the sample. All while maintaining vision lock during the entire intake. The system supports both autonomous and driver-controlled modes with live data.

We use Python to program the OpenCV pipeline and send our results to the control hub as an array of results:

- **Angle to turn the claw to** - calculated with special OpenCV functions for accuracy.
- **The angle in the X and Y axes from the sample to the camera** -calculated automatically by the limelight. The X axis is used to command the robot to align with the sample.
- **Area of the sample** - calculated automatically by the limelight. Used to determine if sample is visible to camera.
- **Distance in the y axis** - calculated with the y axis angle with this formula:

d = (h2-h1) / tan(a1+a2)
d -> distance from limelight, ground level
h1 -> height of the camera from ground
h2 -> height of the the target
a1 -> fixed mounting anfgle
a2 -> angle sent from limelight from the y axis



- **There are Designated Pipelines for each color,** each with its own detection settings, so switching between colors is easy, fast, and accurate.
- **Refined Contour Filtering** removes irrelevant shapes based on area, height, and shape data. Will target rectangle shapes like samples.
- **Vector and high mathematical principles** for finding the lowest distance to the robot center
- **Brightness Thresholding** ignores underexposed regions.
- **HSV Color Filtering** isolates elements of specific color across lighting conditions, so lighting has a much smaller effect on the detection.
- **Edge and aspect ratio validation** ensures shape accuracy.
- **Intelligent Contour separation** untangles overlapping samples, unlike classic color detection, which cannot separate overlapping samples. Thus, it expands the intake capabilities.
- **Frame-to-frame tracking locks onto targets even when in motion,** ensuring targeting only the sample when in fast movement.
- **Screen splitting** ensures the limelight will not return samples that the linear intake cannot open to or in unreachable places for a claw, like the submersible edges.
- **Draw info function** overlays debugging data—angle, color, area—onto the image for quick debugging and live reference.