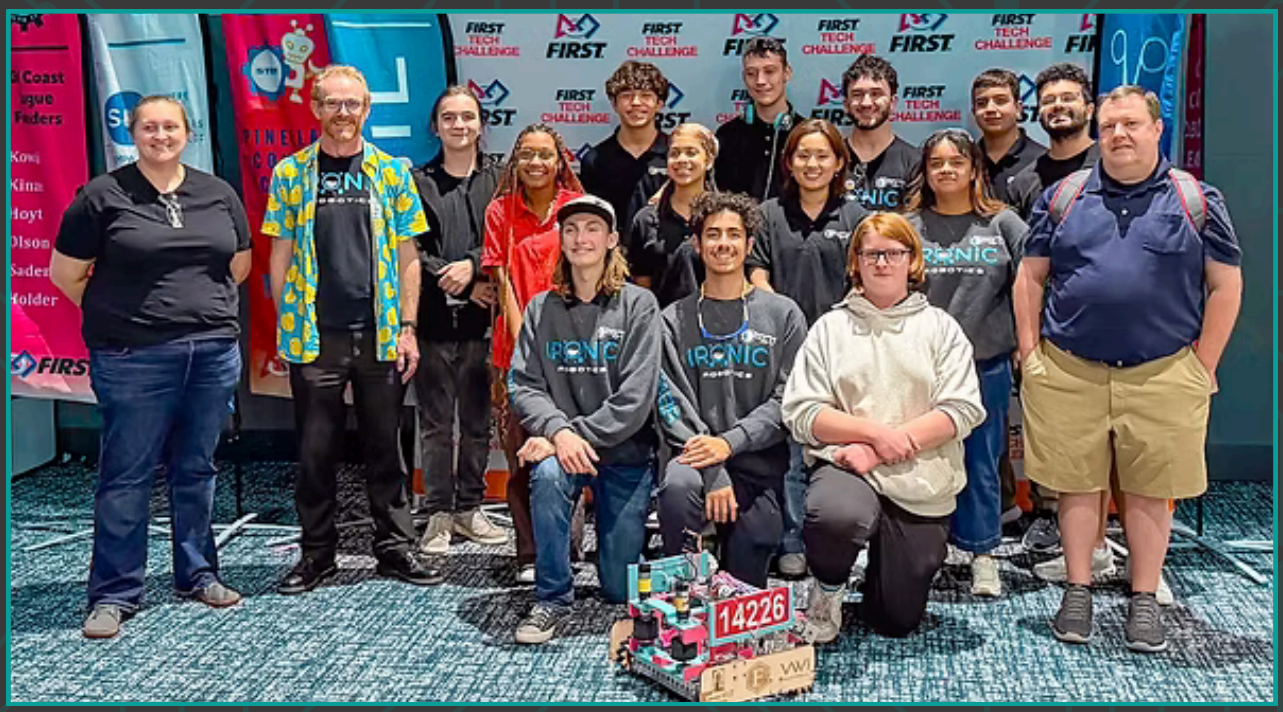
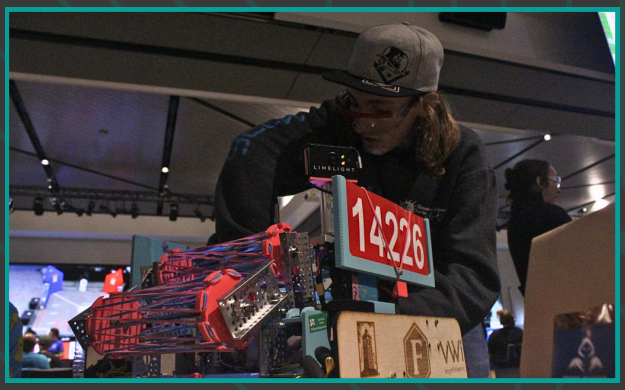
IRONIC
ROBOTICS







14226
TEAM IRONIC

2025-2026

FIRST

DECODE
PRESENTED BY RTX

TABLE OF CONTENTS

Meet The Team — 2

Team Plan — 3-4

- Finance — 3
- Sponsorship — 3
- Sustainability — 4

Design Process — 5

Design of Subsystems — 6-8

- Intake — 6
- Kickstand — 6
- Indexer — 7
- Shoot System — 8
- Index to Shoot Transfer — 8

Strategy — 9

- Quality of Life — 9

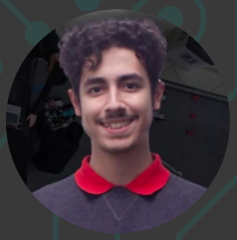
Programing — 10-13

- RoadRunner to PedroPathing — 10
- Queue Creator — 10
- PIDF — 11
- Indexing — 12
- Sensors — 13

Outreach — 14-15

MEET THE TEAM

Captains



Jayden E.
Coder
Drive Team



Ariana V.
Media
Management



Kollin C.
Builder/CAD
Drive Team

Team Members



Lucas B.
Builder/CAD



Nina Z.
Builder/CAD



Amberlynn P.
Coder
Drive Team



Logan M.
Builder/CAD



Joshua V.
Coder



Jackson P.
Coder



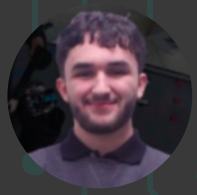
Imani C.
Media
Management



Chase M.
Builder/CAD
Drive Team



Elias C.
Media
Management



Sean W.
Media
Management

Coaches and Mentors



Mr. Foster
Programing Coach
7th season



Ms. McCollough
Mechanical Coach
4th season



Mr. Joe
Programing Mentor
1st season



Mr. Nick
Programing Mentor
2nd season mentor
4 seasons Alumni

TEAM PLAN

Finance

Financial planning of the team is primarily done through our team captains and with the guidance of our coaches and mentors. The funds of our team are held by the finance department of the school; and as a result we are responsible for accounting, as well as up-keeping and maintaining accurate books. Our team sets up budgets to maintain incoming and outgoing funds. This allows us to plan changes in robot design as well as the cost of upcoming events and team plans.

By up-keeping our accounting books, we are able to track not only our current financial standing but also the predicted funds that we need in the future. In addition, our financial plan allows us to budget before jumping into new robot builds. We are also able to account for future needs such as next year's registration and the funds that will be needed as we advance through the current season.

Additionally, these financial books allow us to account for the amount we need to fundraise. A great example of this is the expenses we had at last season's state championship.

Since it was our first time going, we did not know the full scope of the event. We had only accounted for expenses for further robot design, food, and lodging. It was not until we got to the championship that we understood how big the pit environment and culture was.

Moving into this season, we now have allocated additional funds into our budget in hopes that if we are able to attend this season that we can participate in all aspects of the state championship to our highest ability.

Robotics Club Book Keeping - MSA

LEGEND

D = Member Dues Dep WE = Event Payment Withdrawl
= Member Donation D WB = Material/Parts Payment Withdrawl
DS = Sponsor Depos DO = Other Deposit

CURRENT BALANCE
$3,279.73

	Transaction	Withdrawal	Deposits	Balance
DO	End of Season Funds Remaining		$1,170.94	$1,170.94
DS	4C Concreat Construction INC.		$250.00	$1,420.94
DS	B Man Computer Service		$250.00	$1,670.94
DS	FPL Grant		$750.00	$2,420.94
WB	GoBilda	$170.74		$2,250.20
WB	Amazon	$57.56		$2,192.64
WB	GoBilda	$81.91		$2,110.73
DS	Homeschool STEAM		$250.00	$2,360.73
WB	GoBilda	$257.35		$2,103.38
WB	Amazon	$56.47		$2,046.91

ABOVE: Simplified screen shot (excludes direct sponsor purchases) of team books showing income and expenses of the team through the 2025-2026 season.

Sponsorship

In financing our robot and our season, the majority of our funds comes from sponsorships, grants secured from local businesses and companies in our area. Running the program primarily off of sponsor donations places the importance into two areas: securing new sponsors and retaining relationships with current sponsors.

In securing a new sponsor, we use a sponsorship form. This form allows us to articulate the levels of sponsorship as well as what we offer to our sponsors at different levels. We primarily reach out to potential sponsors in the off-season. To do this, we either go to the location in person or send an email an executive. Typically, we check back in with the business after two weeks.

BRONZE $250	• Sponsor's name on the back of the IRONIC team jersey and Robotics Club t-shirt • Advertised as being a Sponsor on IRONIC's website
SILVER $500	• Bronze level sponsorship opportunities • Logo on all marketing materials • Logo on Team Banner
GOLD $1000	• Bronze level sponsorship opportunities • Silver level sponsorship opportunities • Sponsors advertisement in our brand new MSA Robotics room. • Your Logo on the IRONIC Competition Robot

ABOVE: Simplified screen shot of teams sponsorship form showing the different levels and offers for each level.

We find that maintaining sponsor relationships is just as important as finding them. Initially when the team started we found that if the sponsor was not updated on the team and our success, they would be less likely to join us for the following season. Knowing this, we try to update our sponsors monthly with newsletters and social media shoutouts. At the end of the season, we take the time to formally thank our sponsors. We update them on the results of the season, and how their contribution helped make it happen.

TEAM PLAN

Sustainability

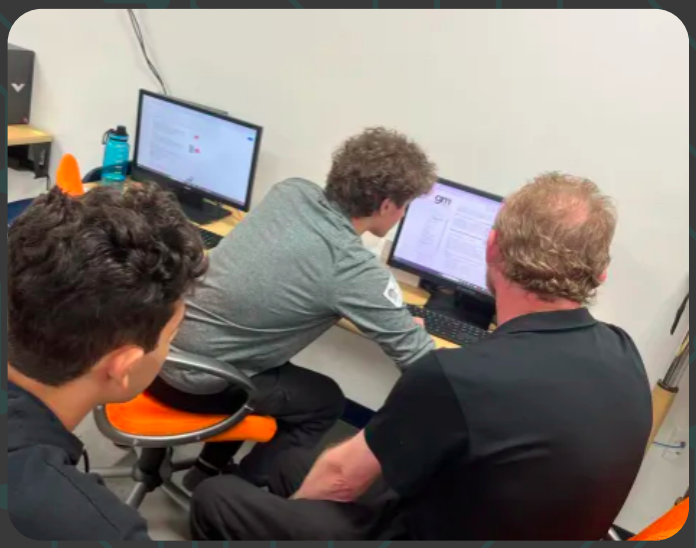
To us, sustainability is a key value. We heavily prioritize retaining student interest in our community and school to allow robotics programs to thrive.

Last year, we made an impact on our school's robotics program. Before the Into The Deep season, we were the only form of robotics program in our school. This made it really hard for students interested in robotics to pursue robotics. After our success last season, our school's overall interest in robotics expanded; allowing the program to grow into something larger. Now, our school hosts 3 classes with a total of 6 sessions. Many of our team members are integrated with these newly made robotics programs at the school, with some even becoming teacher's assistants and helping to run the program themselves.

Every season, we run stations during our school's open houses to get incoming and current students interested in robotics. This season, 60 middle school students came to our station and expressed their interest in robotics. With a heightened interest in robotics from middle school students, our school was able to create a FIRST Lego League team which we helped to support and mentor.



ABOVE: Team members showcasing the DECODE season challenge.



ABOVE: Students going over and practicing portions of the team tryout with Mr. Foster.

This season, we relied on a tryout process to select our team members. Over 20 students came to attempt the tryouts, many more visited the lab to show interest and learn what we are about. The purpose of introducing the tryouts to our selection process was to assess whether students could succeed in tasks required for the team as well as if they were open to learning more. Our tryout process also helped students truly understand what our competition was about, allowing them to get experience before the real competition.

Our team is taking charge in our community to be STEAM activists, advocates and ambassadors. We participate in community outreach every single year. We provide opportunities to students and others alike, which open them to a whole new community. This helps keep STEAM culture alive in our community and also preserves the legacy of our team. However, in doing this, we allow our newer members to step up. This helps them build confidence and leadership skills, allowing us to sustain our team. Our goal is to not only inspire others outside of the team, but to also help our own members grow into innovators and leaders who will guide the next generation of our team.

DESIGN PROCESS

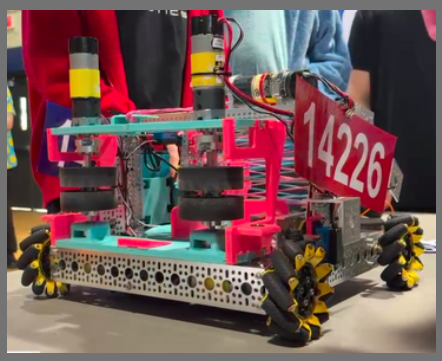
Intro To Design

Learning from our experience at the state championship last season, we decided to view robot iterations and improvements differently. In previous seasons, we opted for full bot redesigns between competitions. This design process made us feel like we were always trying to catch up to the competition. Coming into this season, we looked at iterations as fine tuning changes to individual systems. In doing this we were also able to preserve our base code in addition to the fluidity between system transfers.

To ensure we use our time efficiently, we now follow a set design process. When deciding on a potential design change, we first render a functioning model in CAD. After the CAD model is complete, we begin building physical prototypes. Once there is a functional and heavily tested prototype, we implement the change to our robot.

Robot Iterations

Scrimmage and Meet 1



Going into this season's game we wanted to include a few things. We knew we wanted an intake system, this would allow us to pick up artifacts efficiently. We also wanted an indexing system in order to complete the motif.

The last thing we knew we needed was a shoot system. We landed on a horizontal flywheel design that could move up and down to utilize more space. We included a transfer system to push the artifacts from the indexer directly to the launcher. Our chassis design itself was formed like a U to allow us full access to the inside of the robot.

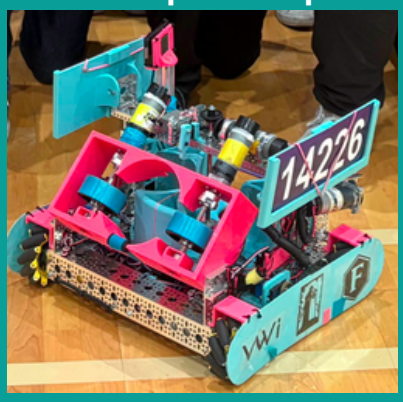
Meet 2



When designing the second iteration of our robot, we made multiple changes to improve our efficiency while in a match. We changed the design of the indexer to have walls which flared out. We also changed the actuator on the intake from a servo to a motor to improve speed and efficiency.

As for the shooter, we released compression from the top portion of the ball by raising the top of the assembly, creating an arch. We also made the shooter itself stationary and no longer able to lift up and down to make our cycle time quicker.

Meet 3 & League Championships



When designing our third iteration we thought of the indexer and shooter. We needed a better hold on the artifacts while the indexer was rotating. To do this, we made built-in walls to the indexer that wrapped around the artifact. This allowed us to rotate the indexer faster compared to before.

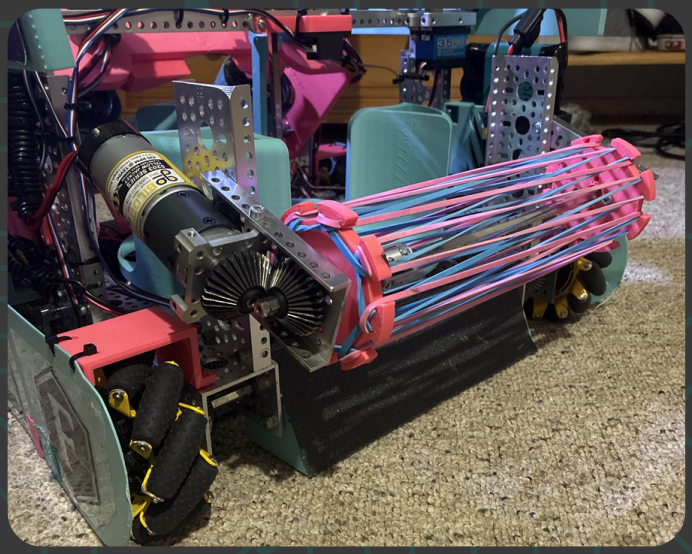
To solve a lot of problems with shooting, we needed to minimize variation in the shooting trajectory of the artifact. To fix this, we moved the transfer system to be above the indexer. We also made a chute inside the shooter to better fit to the artifacts.

As we move into league championship, we aim to improve the accuracy more by moving the wheels further from each other in order to lower the amount of compression of the artifacts.

DESIGN OF SUBSYSTEMS

Intake

For this season's robot we wanted to use an intake system. We decided this over other methods because of the opportunity to pick up artifacts anywhere in a timely manner. We used rubber bands on our intake because they form around the artifact. We also allowed the intake to move slightly up and down to help guide the artifact. On top of that, we had to make a ramp for our intake, which we applied grip tape to in order assist in gaining friction. To help with the transfer from the intake to the index system we placed a second intake row to help push the artifact further up the system. However, we decided to remove this later to give our robot the ability to be human player friendly.



ABOVE: The intake system powered by a 412 RPM motor and bevel gears. Two spindles with rubber bands act as the grabbing mechanism to pull artifacts into the robot.

Our choice of actuator at first was a speed servo, however we quickly realized that we needed more speed on our intake. We ended up going through a super speed servo, a 312 RPM motor, and then a 435 RPM motor, which finally ended up giving us the speed that we needed. Since we increased the speed on our intake, we also realized we could slant the ramp on our intake to be steeper.

Kickstand

When we were making changes for iteration two of the bot, we decided to add a kickstand for endgame play. With our robot using all 8 motor slots we decided that the kick stand, only requiring a servo, was the best option without having to redesign our other systems.

We opted for two kickstands mounted on the outer perimeter of the chassis. By distributing the weight of the robot across both, we could reduce strain to any singular servo.

Each kickstand consists of a large gear with a flat beam, and a small wheel placed on the bottom to allow it to roll This also allows for the drivers to adjust the park easier. After Meet 2, we decided to flip the placement of the small wheel from facing out, to facing in to allow us to fit better within our 18 by 18 inch guidelines. We also placed a rod to prevent the servo from zeroing to the wrong location to ensure that the system always starts in the correct position.



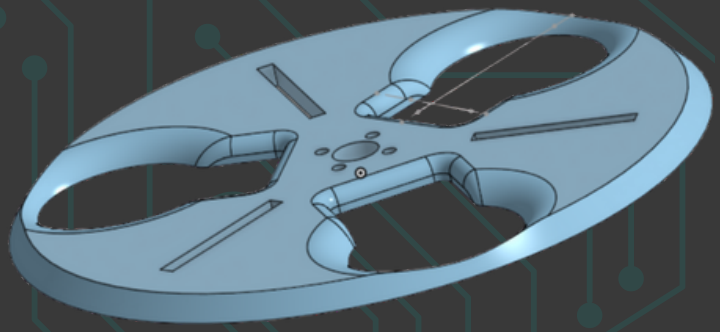
ABOVE: The right side kickstand consisting of gear and beam powered by a 45kg servo.

DESIGN OF SUBSYSTEMS

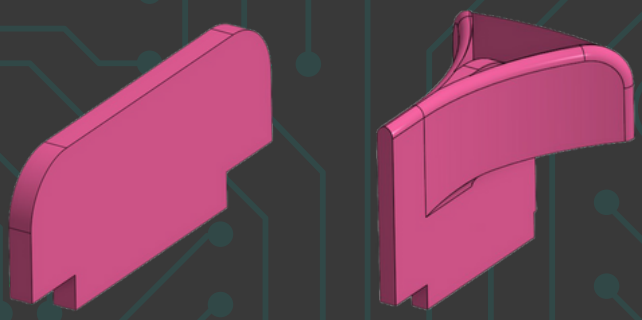
Indexer

Coming into this season, we aim to have our strategy revolve around the motif points. We felt that the best way to do this would be to allow the bot to pick up and choose which artifact we wanted before launching. Solidifying this strategy, we added a horizontal spinning indexer to our design that we kept through the whole season. As we moved through the season, some changes to the indexer were made to optimize the design.

Our first design of the indexer was a mix between a fidget spinner and a spinning plate. To allow for better control over the artifacts, indentations were added in addition to interchangeable divider walls. This allowed for the artifacts to stay stationary and minimized the risk of them falling out, or rolling into another slot.
As we moved to using a magnetic limit switch for calibration, a receded hole for a magnet was added to the underside of the indexer.



ABOVE: The first iteration of the indexer was a simple rotating plate with indentions for the balls to sit as they rotated.

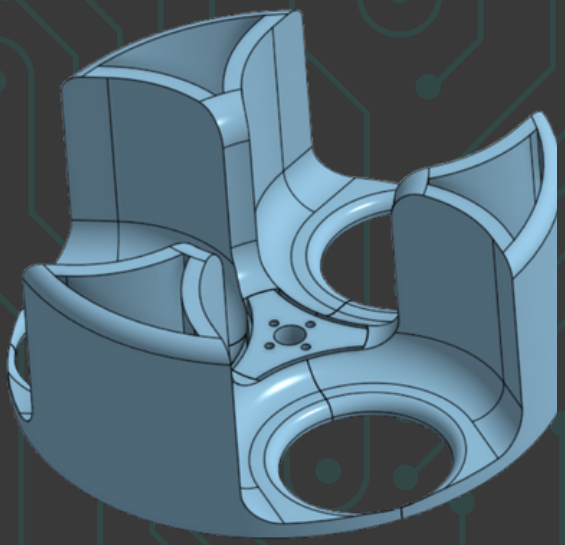


On the Left: First iteration of the indexer walls.

On the Right: Second iteration of the indexer walls with winged shape to better contain the artifacts while spinning.

In Meet 2, we faced jamming issues caused by interference with the artifacts against the portion holding the axle and motor for the indexer. We also faced artifact detection issues because of the holes on the artifact. Our solution to these issues was to implement bigger walls which flared out to better contain the artifacts in their respected areas. In addition, we made the outer walls curve around the indexer to prevent the artifacts from getting loose. We changed the positioning of the color sensors to have more surface area to read the artifacts in which helped with the detection issues.

For Meet 3, we continued to face jamming issues. In addition, we realized the slower speed of the indexer was hurting our cycle time. To resolve this, we removed the slotted walls and instead added built-in indexer walls with much more height and width to fully encompass the artifacts. These changes allowed for the artifacts to be transferred at much higher speeds without getting jammed.



ABOVE: Iteration 3 of the indexer with built in walls to better hold the artifacts during rotation and deeper indentations for the artifacts to sit better.

DESIGN OF SUBSYSTEMS

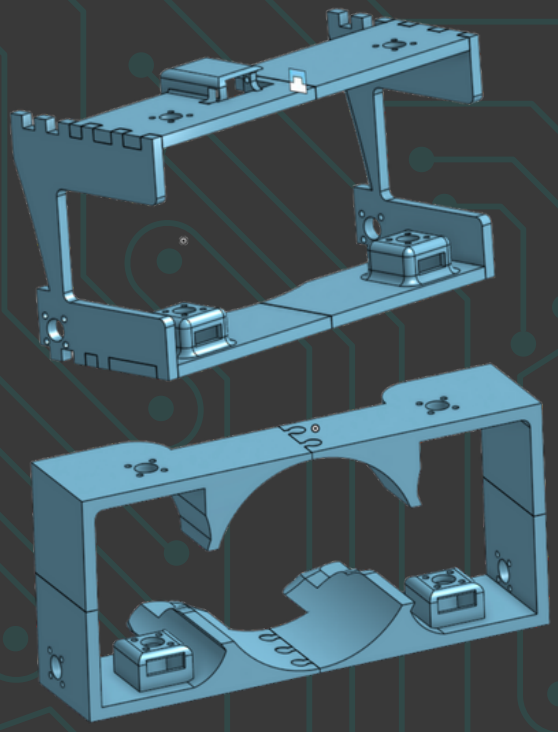
Shoot System

The shooter system consists of two fly wheels arranged horizontally that grip onto the artifact and launch it into the goal. In the first iteration, we had it so the shooter could tilt forward and back via 2 servos on either side of the shoot. We found that this caused variability and slowed cycle time because of constantly having to readjust; so we changed it to be static in the rest of our iterations.

In our first static design, we added a wedge at the bottom of the shooter to increase our launch angle. This addition also induced spin onto the ball. To allow for the ball to shoot in an arc, we raised the top part of our shooter to be arched.

We ended up finding variability in the path that the artifact took inside of the shooter prior to interacting with the fly wheels. To fix this, our last iteration incorporated a chute that fit the around the artifact. Additionally, we swapped the ramp for a rubber bumper to better induce spin onto the ball.

As we approach league championships, we plan on imparting 2 more changes to our shooter. Firstly, we plan on reducing the amount of compression we have on the artifact by increasing the space between our two fly wheels. After that, we would like to increase the amount of friction on the artifact through the shooter to further increase spin and therefore accuracy in our artifacts flight path.

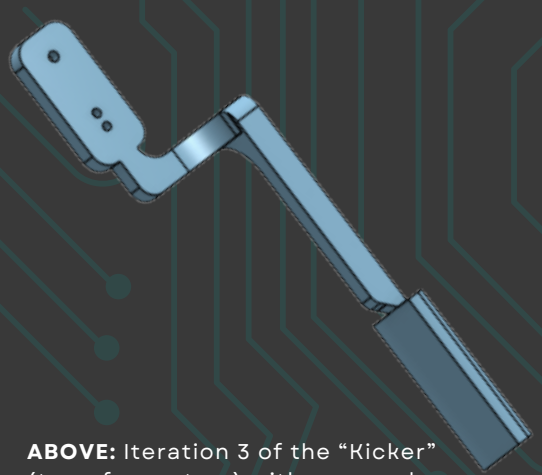


TOP: The first iteration of the shooter. It used 2 servos to rotate into the shooting position.

BOTTOM: The current shooter design that encompassed the form factor of the artifact to limit variation in launching.

Indexer to Shoot Transfer

The transfer system, or kicker, started out being mounted below the indexer. The first iteration of the transfer system rotated through premade cutouts on the indexer tray, to push the artifacts into the shooting mechanism. There were slight issues with the first iteration however; the kicker was too short to successfully push the artifacts into the shooter and as such was commonly inaccurate. To fix this problem, we revamped the indexer to have larger holes in the trays so that the transfer mechanism arm can be longer.



ABOVE: Iteration 3 of the "Kicker" (transfer system) with a curved base to wrap around the artifact.

This solved the majority of the problem, but this solution was still rather slow and inaccurate. To solve this, we had to take a new approach. We changed the mounting position of the kicker from the bottom to the top of the robot. We also extended the arm of the transfer mechanism. This solution works faster than the previous version as the paddle for the arm is much farther away than the source of rotation causing faster relative motion. Another change we made to the transfer system was to cup the end of the arm. On top of this, a limit switch was added in this phase to detect when retraction of the kicker was complete so we can start rotating immediately. This design works much more reliably to guide the artifact in and is a much intuitive solution to the transfer issue.

STRATEGY

We built our bot around being able to achieve multiple extra ranking points. Our indexer allows us to consistently score artifacts in a pre-determined order from the obelisk, helping our alliance gain the motif ranking point. When it comes to the parking ranking point, we utilize our kickstand to support half of our bot in the base, allowing us to gain a full park regardless of our alliance partner's size.

Our autonomous system is designed to be used within an adaptive environment. Working with our alliance partner, we determine which autonomous paths to use. Then, we use our queue creator to make the path sequence, and our Limelight camera to detect the motif from the obelisk. This allows our autonomous strategy to mesh well in any environment.

During the match, we work around our alliance's needs. Moreover, we heavily strategize with our alliance partner before matches. We talk about each other's strengths and weaknesses, we test if our autonomous programs work well together, ask how they prefer to score, and develop a strategy as a whole around that information. With recent changes, our cycle time increases tremendously when incorporating our human player. If possible, we always aim to be near the loading zone and take the artifacts in rapid succession before returning to the launch zone.



LEFT: Our drive team discussing with team 17015 KING TARP's drive team.

RIGHT: Our drive team discussing with team 24702 VOLTARIS' drive team.



Quality of Life

We added quality of life features to help the drivers and the coders with completing their tasks more efficiently. One example would be our toggle features on the controllers to make robot slower for the drivers. With this, we can park and aim with more accuracy. This also allows us to drive more carefully around protected zones.

We made most of our systems automatic so we could save time during matches. For example, our indexer turns automatically when the front color sensor detects an artifact. In addition, our transfer system has a limit switch that notifies us when the indexer is safe to move again. This avoids the problem of the indexer accidentally colliding with the transfer mechanism.

When it comes to coding, accessibility is key. When we make code, we always make sure to add documentation comments to help other programmers understand what specific parts of code are used for. We also make sure to review code for code smells, depriving the code of any unnecessary methods, classes, or counter-intuitive code. This makes reading the code simpler for others to review. We also made documents to each component's configuration in the Control Hub. This way engineers can replug components if any changes are made.

PROGRAMMING

RoadRunner to PedroPathing

This season, we decided to make a big change in which library we use for autonomous programming. Last year, we utilized RoadRunner 1.0. Now, we are using PedroPathing. We found PedroPathing to be easier to use and more customizable with our systems. With PedroPathing's coarse and fine correction, along with its correction for unexpected disturbances; it was the best choice for us. PedroPathing's inherent design was also easy to use with the code for our subsystems.



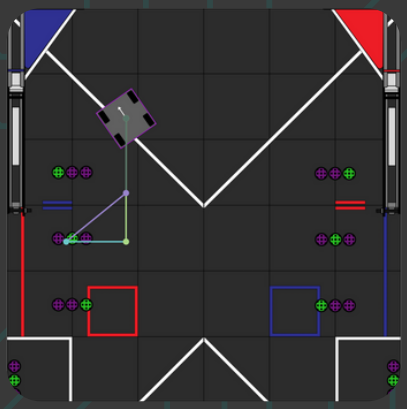
- Follows trajectories using feedforward & feedback control.
- Mostly automatic tuning system, having to work off scatter plots.
- Time-consistent trajectory following.
- Utilizes a marker based system that you can place throughout your paths to do other actions.

RoadRunner

- Uses three PIDF controllers and centripetal force correction. Optional dual PIDF system, allows for coarse and fine corrections.
- Half automatic, half manual tuning system, can physically feel changes.
- Allows for easy correction of unexpected disturbances.

PedroPathing



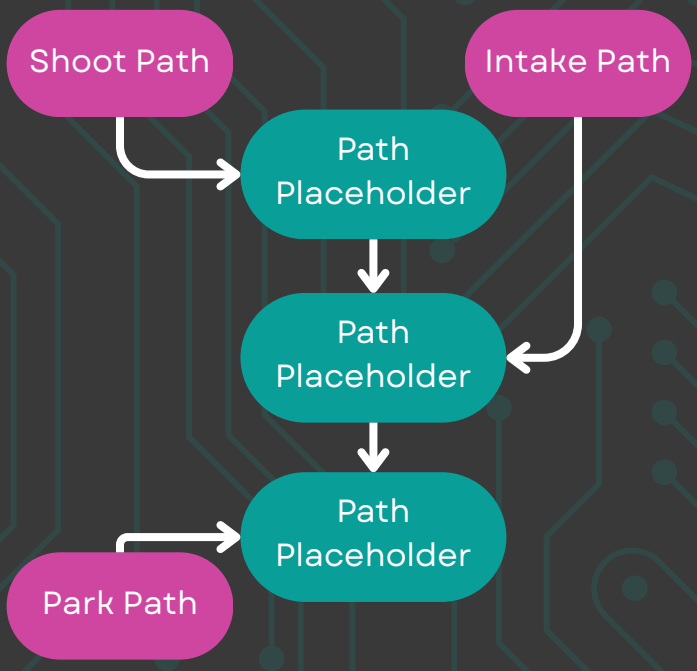


ABOVE: Visualization for one of our blue autonomous paths using PedroPathing Visualizer.

We used PedroPathing to create multiple paths, both for the red and blue side respectively, by using the online PedroPathing visualizer. Then, we crafted actions for the moving systems that are not the chassis: the intake, the indexer, and the shooter. We placed these actions inside of a state machine housed in the same class as the individual paths. After that, we hand the responsibility to our drivers to craft path sequences using our queuing system.

Queue Creator

Our queuing system allows us to combine and select paths to create unique autonomous sequences for certain scenarios. This is especially useful for working alongside alliance partners. We can build path chains around the alliance's abilities and weaknesses. The queue creator is very accessible and easy to use in a competition setting. Instead of having to go through the code to change paths, we can dynamically create path chains from the Driver Hub by selecting path segments in a preferred order. The ability to change and edit our autonomous program as we change alliance partners allows our strategy during autonomous mode to be very adaptable.



PROGRAMMING

PIDF

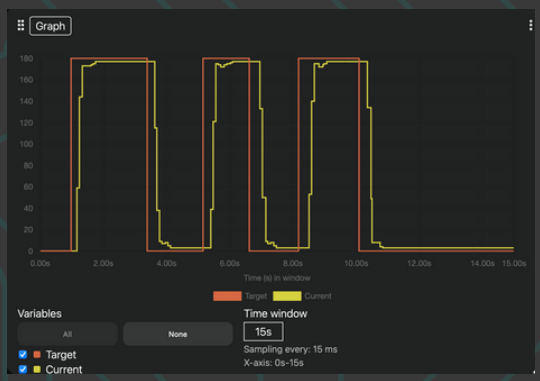
For a lot of our moving systems, we implement a proportional-intergral-derivative & feedforward (PIDF) controller on our motors.

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau)d\tau + K_d \cdot \frac{de(t)}{dt} + \mathrm{FF}(t).$$

This equation calculates corrections for past, present and future errors, and adjusts oscillations accordingly based on your needs. The PIDF equation states that t is time, $u(t)$ is the output and $e(t)$ is the calculated error factor. K_p is the proportional tuning gain and adjusts the output based on the current error. K_i is the integral tuning gain and sums up past errors to try to eliminate offsets. K_d is the derivative tuning gain and predicts future errors based on rate of change and dampens the output accordingly. ***FF*** stands for feedforward and is an additional term that gets added onto the standard PID equation; it helps compensate for real-world disturbances, like gravity or friction, to make the system more responsive.

Lots of different systems on our robot utilize PIDF controllers. For example, our chassis utilizes PIDF controllers on all four of our drive motors to move during the autonomous period and help with defensive driving. In addition, our shooter motors use PIDF controllers to help maintain a set velocity when launching artifacts. Lastly, we apply a PIDF controller to our indexer for precise and swift positioning.

The way in which we tuned each system varies slightly. For the chassis, we started by tuning the feedforward values first to compensate for friction as well as inertia from our robot's design. We also decided to tune the feedforward value first when it came to the shoot system's motors. Since the shooter motors are high RPM motors, they need to overcome more rotational inertia and can become unstable quickly. The feedforward value can help in these scenarios by providing a push in designated power; making the motors reach their target velocity quicker while also allowing the motors to stabilize more efficiently. Alternatively, for the indexer, we tuned the proportional gain first to get a quick response time. For league championships, we plan to test a dual PIDF system for the indexer to get a more accurate position. One PIDF controller will control coarse movement, and the other will control fine movement when the indexer gets close to its target position.





TOP: Panels graphing indexer PIDF tuning results.

BOTTOM: Panels graphing shooter PIDF tuning results. Each minor drop in the graph represents an artifact being launched.

PROGRAMMING

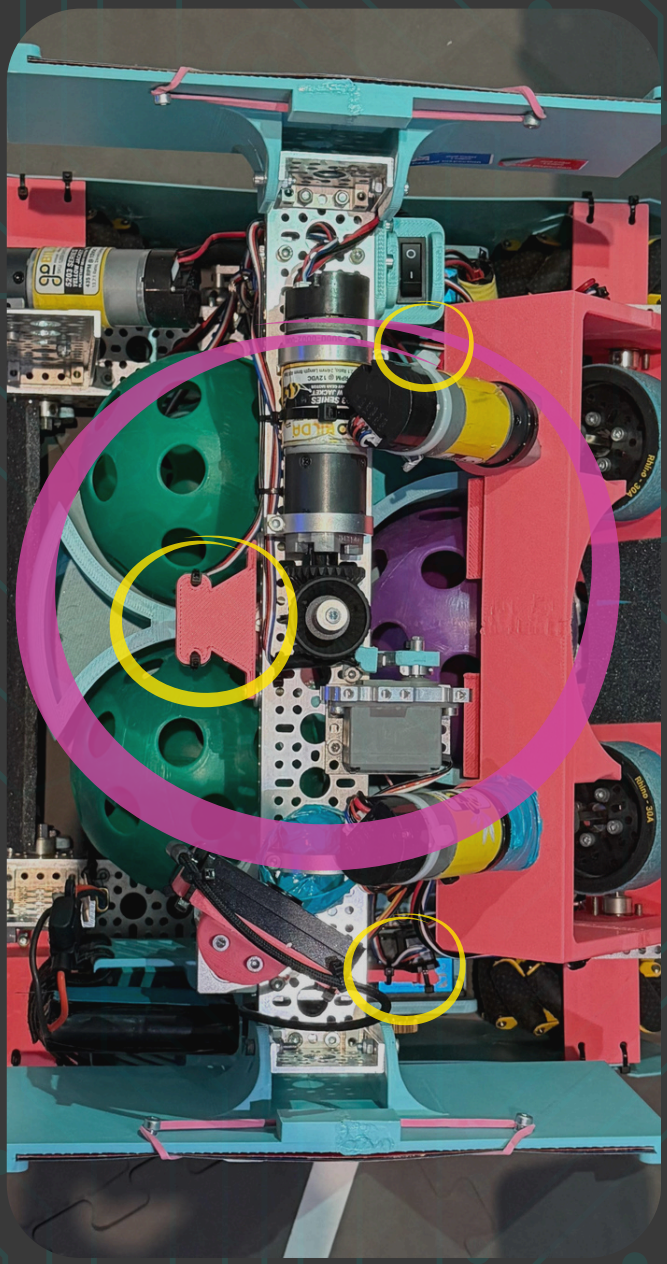
Indexing

The indexer has three slots for artifacts, and each slot has its own memory. The indexer detects artifacts through multiple color sensors and has the ability to sort them depending on the position of the indexer. The code saves the position and color of the artifacts stored inside of it. When the drivers are ready, they can easily swap through artifacts based on the color they need to complete a pattern. This information is also sent to the Driver Hub through telemetry and can be used to help the drivers know the color of the artifacts.

There were multiple issues we were able to solve with sensors. Originally, there was only one color sensor reviewing our indexer. To account for errors, now we have three. Additionally, the color sensors on the robot also act as distance sensors. When an artifact gets seen, the indexer knows that it is ready to move on. However, if the color of the artifact is not able to be read in time, the other color sensors would be able to correct it. To counter misalignment, we installed a magnet onto the indexer, and a magnet sensor underneath the indexer. When aligned with the magnet sensor, we know that its positioning is correct and that we are realigned.

Our indexer's sorting systems are useful during the autonomous section of the game as well. It is used to automatically sort through artifacts based on the pre-defined pattern. As the game begins, the Limelight camera stores the motif from the obelisk. The robot will use this information to shoot the artifacts needed to complete the motif.

The picture to the left showcases a top-down view of our indexer. The yellow circles indicate where the color sensors are located. These color sensors also act as distance sensors if necessary. The pink circle indicates our indexer. The indexer in the photo is indicative of the most recent design, one that is able to retain the ball while still moving swiftly.



ABOVE: The current indexer in shoot position with all 3 slots loaded with artifacts.

Yellow Circles: Color Sensors

Pink Circle: Indexer

PROGRAMMING

Sensors

This season we used many different types of sensors to increase accuracy, precision, and speed. Our current design encompasses sensors like motor encoders, color sensors, magnetic limit switches, odometry pods, limit switches, and a limelight 3A camera. Each sensor seen in our current build helps our robot navigate, sort, and shoot to its best abilities. This plays an important role in the robot's functionality.

Color Sensors

Our current design has 3 color sensors, one for each slot that detects whether the slots current artifact is green or purple, this can be used to sort the artifacts into the patterns needed.

The sensors can also detect if something is in their limited range. This helps if they sense something but aren't able to detect it's color in time.



Magnetic Limit Switch

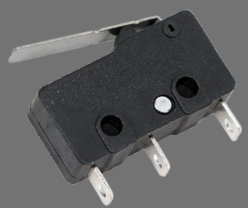
There is a magnetic sensor under our robot's indexer system. This allows us to detect if the indexer has completed a full rotation. This can be useful to help with the indexer's accuracy and precision. The magnetic sensor also helps calibrate the indexer if it gets misaligned.



Limit Switch

Our transfer system's limit switch is above the indexer. This limit switch indicates if the transfer system's arm is fully retracted. This makes cannot collide with the indexer.

Previously, an intake limit switch was added to our ramp, but later we removed it after learning that the color sensors could do the same job, more accurately.



Odometry Pods

We have a set of two odometry pods under the robot. The odometry pods track where the robot is. This is mainly used during the autonomous period.

The GoBilda Pinpoint Computer is where the gathered data of the odometry pods are processed to calculate the robot's position and movement. This information is then given to the control hub.



Limelight 3A Camera

Our robot also contains a Limelight 3A camera above the robot mounted on top of a tall extrusion. We use this to detect the AprilTag on the obelisk for the pattern, which is helpful for our autonomous program.



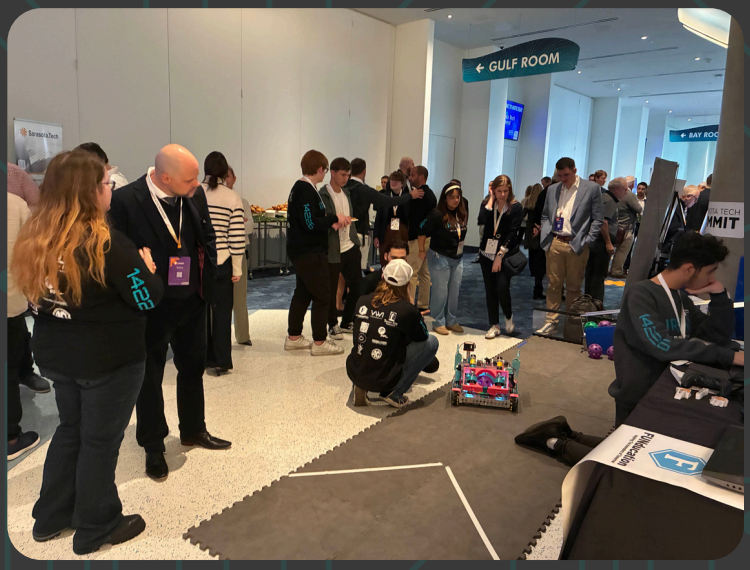
OUTREACH

Florida State Advocacy Conference

STEAM unites us, and we here at Team Ironic believe that STEM education should be available for everyone. The Florida State Advocacy Conference was a conference at the state capitol where we could get to express that. We were able to send a team representative to talk with legislators about positively influencing STEM education policies. This opportunity not only represented the team, but a vast community of people all around the nation. We are very thankful we were invited to go to this event.



ABOVE: Our team representative, Jayden, in front of the Florida State Capitol.



ABOVE: Team Ironic exhibiting to local tech businesses at the Sarasota Tech Summit.

Sarasota Tech Summit

Our team finds a lot of ways to share FIRST to the community and get more people interested in robotics. We recently went to the Sarasota Tech Summit, where we got to exhibit the experience of being on a FIRST Tech Challenge team. We shared our experiences in FIRST to over 500 individuals from the local tech community.

Suncoast Remake Learning Days

Through HOMESCHOOL STEAM, we were introduced to Suncoast Remake Learning Days. Suncoast Remake Learning Days is a community-wide learning celebration where learning happens every day. We participated in 5 Remake Learning Days at the very beginning of this season, introducing STEM to a younger community. Inaugurating STEM at a young age provides students opportunities, and brings a new way of learning to our local community.

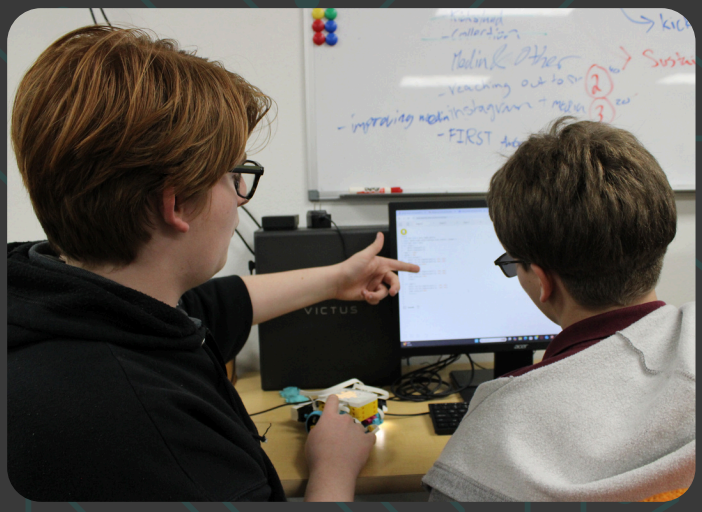


ABOVE: A group photo of all vendors from Homeschool STEAM Fest 2025 at the Lakewood Ranch Library.

OUTREACH

FIRST Lego League

After our school had a surplus of students interested in robotics, the school decided to create a FIRST Lego League team. We helped the FLL team through mentoring students to help them grow both mechanically and confidently. We guided them through the engineering design process, helped them understand robotic construction, and also shared strategies with them. As we continue to help them grow in robotics, we wish to grow their interest in engineering and coding. We hope to inspire these students to continue with STEM and FIRST.



ABOVE: A member of MSA's FLL team, Br@in Storm, receiving help from Chase on coding their team's LEGO robot.



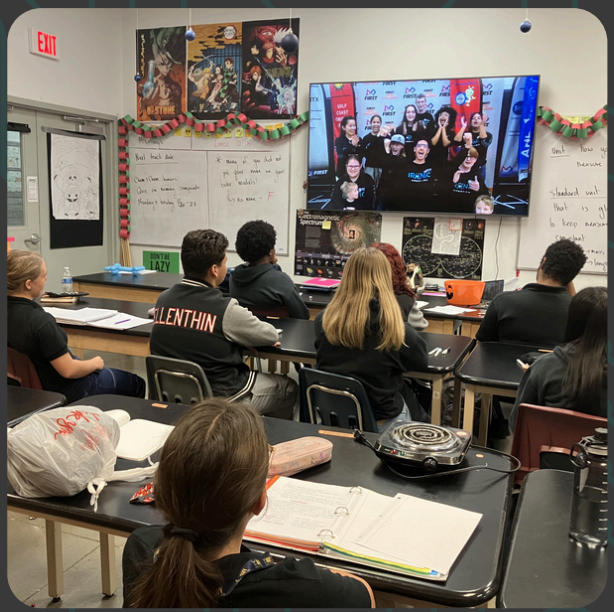
ABOVE: Kollin, Mr. Foster and a representative from team 27103 Lady Tarpons at FIRST Lego League Mix and Mingle.

FIRST Community

Being involved in the FIRST community is one of the largest contributions we as an FTC team can provide. One of the many ways we helped our community this season was by always being there to help set up for events when needed. We helped set up for multiple different events, such as our third meet and the FIRST Lego League Mix and Mingle. We have also collaborated with many other teams amongst us, including those just starting out. Being a FIRST ambassador means empowering others through teamwork, and creating an impact that lasts beyond competition.

School Representation

Within our school, we outreach in a lot of different ways, like our appearances on the announcements. We also participate in our school's open houses to engage students in robotics. We also recently expanded into our robotics classes in our schools, with some of our team members being teacher's assistants. We use these methods to keep our own school inspired and interested in robotics.



ABOVE: Students at our school watching our segment on the school announcements.