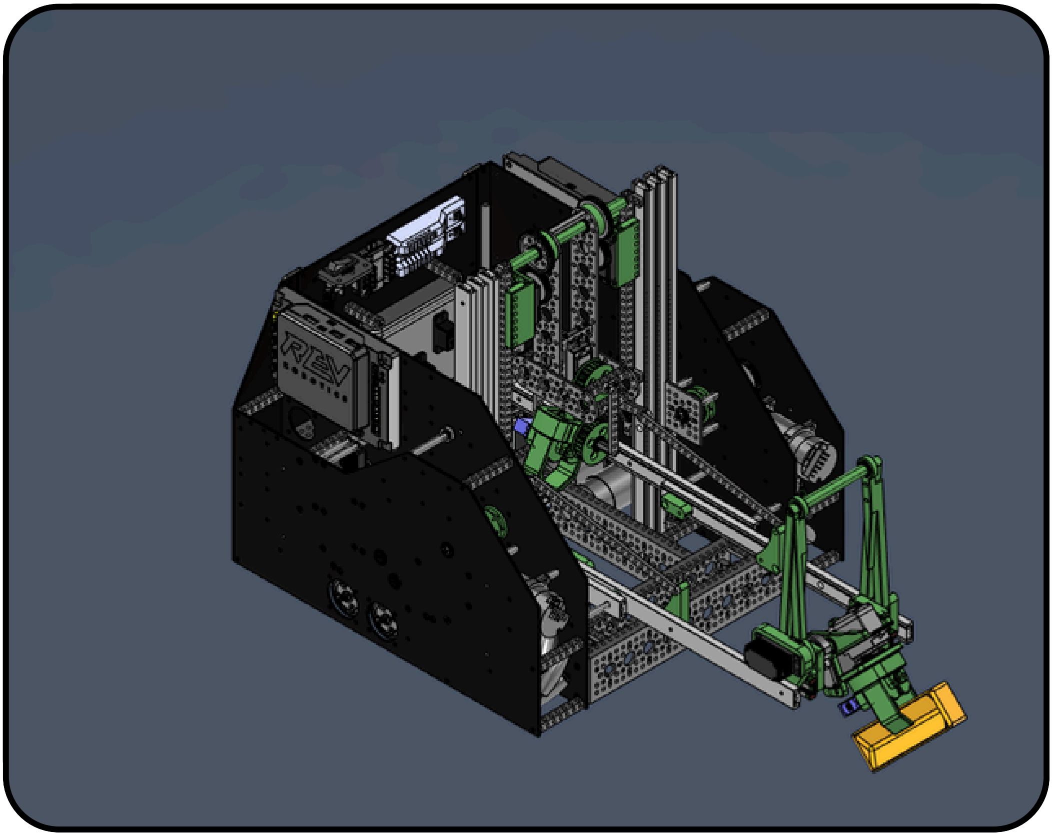
19892 DIGITAL DUKS



ENGINEERING PORTFOLIO 2024-2025

19892
ENGINEERING PORTFOLIO

THE TEAM

This is a photo of our team this season. We have competed in 3 seasons prior to this year. Our team has 3 members: Caden (driver/ portfolio manager), Lukas (CAD/head designer/driver), and Jeremiah (Programmer/driver coach/designer/CAD). All of us pitch in for the design process and building the robot. We have challenged ourselves with only having 3 teammates to work on the bot this season.

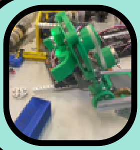
TABLE OF CONTENTS...

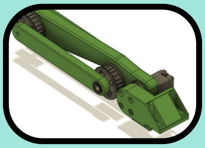
19892
ENGINEERING PORTFOLIO

TIMELINE

Preseason week 4:

We began our season experimenting with belt driven mecanum wheels and swerve modules.



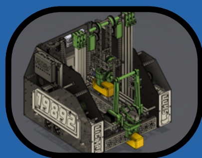


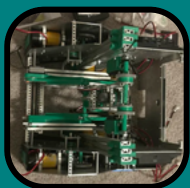
Week 1:

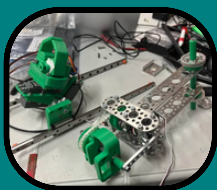
We then started prototyping claws and extendos.

Week 8:

With only finishing touches left for the CAD model, we began ordering parts and machining our side plates.







Week 12:

With everything in order, the building could commence. We worked hard to get our robot assembled and ready to compete.

Week 18:

We redesigned our intake claw, and 3D printed some wire management to hold our wires. We also redesigned the lift after our first qualifier a few weeks before.

SPONSORS

Rivers Edge Engineering & Design:

We have some sponsors that sponsor our team, they are all listed below. For example, we were able to machine parts with the help of Rivers Edge Engineering & Design, letting us use their machinery. Without their help we would have been behind on our side plates.

Heggie's Pizza:

Our team sells Heggie's pizzas to help raise money for expenses on our team. Each of us raises around $200 per person, which is around 13-15 pizzas per person. We appreciate Heggie's for allowing us to fundraise money for our program.

TEAM SPONSORS

ABETECH® Bar Code & RFID Solutions

DAIKIN APPLIED

Stonebrook Fence Inc.
11071 225th St East
Lakeville, MN 55044
Phone: 952-469-8401
Fax: 952-469-8402

GENE HAAS FOUNDATION

RALLY GEAR

Rivers Edge Engineering & Design
John Ibeling
jibeling@ree-dinc.com

PROTOLABS

State of Fun

lucid

NORTHWEST METRO ROBOTICS

LIONS INTERNATIONAL

EMERSON

Steinwall PLASTIC INJECTION MOLDING

NEXTERA communications

HEGGIES PIZZA FUNDRAISING

Denny's

Reliable Parts. Reliable People. MECHATRONIC SOLUTIONS

RAMAR Precision Sheet Metal Inc.

rms

BANNER®

VOLUNTEERING

League Meet 0:

During the preseason league meet our team volunteered as referees to help operate the scoring for the matches. We helped set up the fields and pack up when the event was over.

League Meet 1:

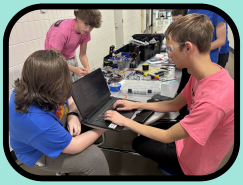
During the first league meet we helped referee the game as well as checked for field inspection. We helped set up the fields, making sure they had enough samples, and the bars were set correctly.

League Meet 2:

During the second league meet we volunteered as referees, scoring matches and inspecting the robot to see if it passes field inspection.

Outreach:

When we were volunteering at the league meets, we had some time before the matches started. We were able to help one of the teams "Exploding Microwaves" with some coding issues they were having. We helped them lower their robot's arm without hitting the ground.



We mentored some of the other teams at Rogers. Rogers has 6 teams and 4 of them are brand new to FTC. We have helped the other teams with questions for designs and programing issues.

Fun Fridays:

We volunteered at Fun Friday events. The premise was to teach elementary school children about the value of STEM by allowing them to construct straw rockets.

DESIGN PROCESS

Designing the robot:

When we designed the robot we started with making 3d printed pieces or simple assemblies. The pictures on the right shows parts we printed. We than machined or bought them if they needed more strength. We did this because buying parts and machining is expensive. We made sure the final bot utilizes machining and vendor parts to maximize strength.

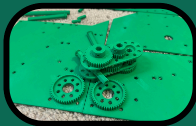
Lift pieces:

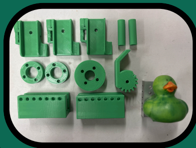
The spool holders on our lift where 3d printed. They are the top left parts in the lower picture. We changed the lift from cascade to a continuous rigging because of problems with the wire not staying tensioned. This is an example of how we used 3D printing to quickly iterate the robot.

Design Process:

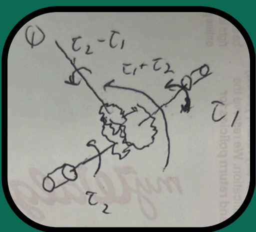
We also used simple drawings to help communicate with each other what we were talking about. Though Lukas is the lead designer, we made sure that everyone's ideas where heard and considered, including the coaches, and other teams.

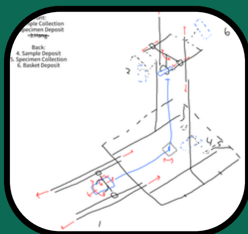
3D PARTS





EARLY DESIGNS

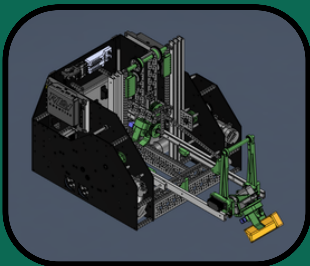


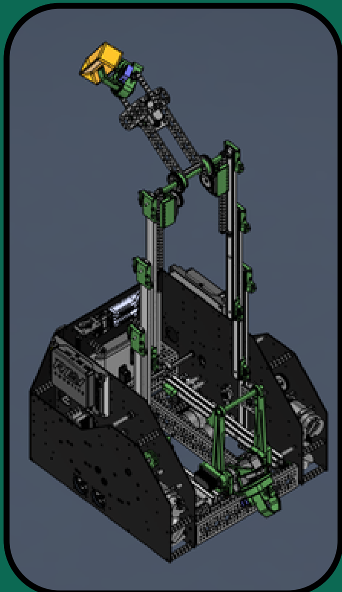


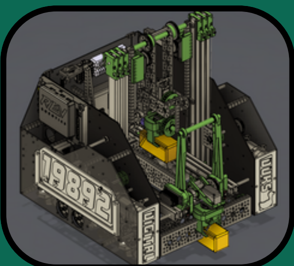
CAD

We made sure to use CAD to the fullest extent while designing. We had made a complete CAD model of the robot before we assembled anything other then early prototypes. We mainly used Fusion 360 but also occasionally used FreeCAD. During the design process we created a philosophy ,“Tolerance is a Boolean.” We did this initially as a challenge but quickly implemented it on the whole robot to maximize space for components, while minimizing the footprint. The following few pages will have pictures from the CAD with short descriptions.

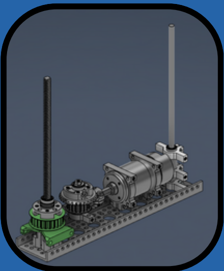
THE ENTIRE ROBOT CAD



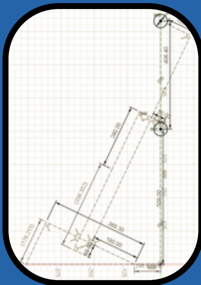


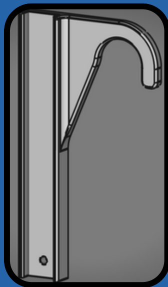


DESIGN



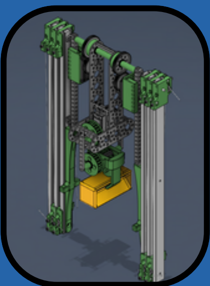
This is the stilt we use to tilt the bot during endgame to hang. It is powered by a lead screw on a motor.





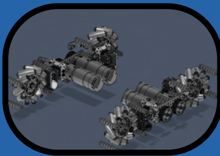
This is the clutch drive we use to let the drive motors power the lift during the hang. This drastically increases the power and allows us to not worry about weight during the hang.

This is the extendo. it reaches into the submersible and uses its claw to pick up samples. Its claw has 3 DOF and can slide back and forth independent of the slides. it does this so it can reach the back of the bot to pass the samples to the lift claw.

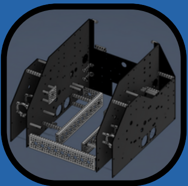


This is the lift. It has a 3 DOF claw as well and uses it to grab the specimens and samples from the extendo claw. It can then lift to predetermined height settings for any scoring area. It originally used cascade rigging but was swapped to continuous rigging

DESIGN



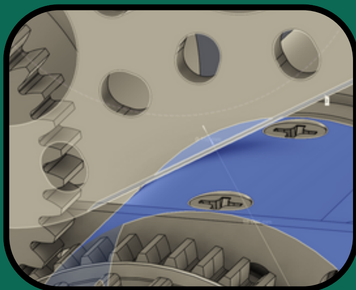
This is the drivetrain. It uses belts and mecanum wheels. It also features three wheel odometry for field concentric drive and autonomous.

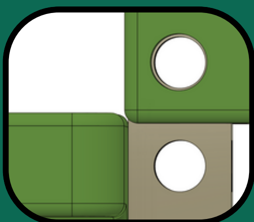
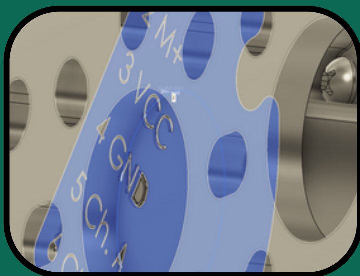


This is the structure of the robot. In previous years we made the components fit the structure. This year though, the structure was made to fit the components. This makes it much more sturdy and durable.

TOLERANCE IS A BOOLEAN



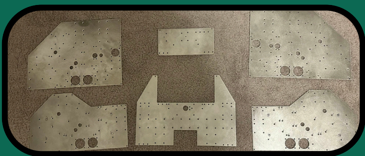


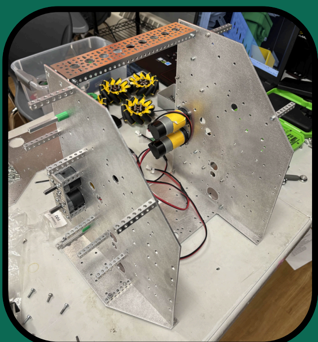



FABRICATION

While we made sure to utilize 3D printing and vendor parts, we also machined our side plates. John, the owner of Rivers Edge Engineering and Design was generous enough to machine our plates for free, as long as we showed up and learned how to do it. He showed us how to use many tools and machines, most notably his CNC router, and Haas mini mill.

MACHINING OUR SIDE PLATES





AUTONOMOUS/TELOP

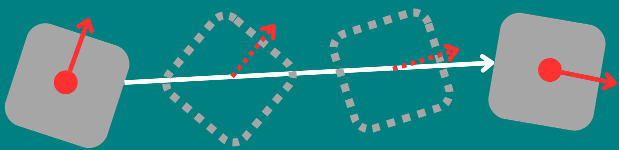
During both the teleop and autonomous period, our robot uses various sensors to make its own decisions and simplify driver controls. This includes distance sensors, encoders, cameras, and odometry wheels.

Teleop:

Every subsystem has its own state machine, allowing the drivers to control mechanisms through a couple of button presses easily. In fact, transferring samples from the intake claw to the deposit is entirely automatic; the robot knows when the appropriate conditions are met and can initiate a transfer. Furthermore, due to the design of the extendo linkage, the robot must automatically coordinate the position of these subsystems to avoid collisions. All motors are controlled using PIDF controllers - no motor or servo is entirely manual. We also use a field-centric control scheme with automatic heading correction.

Autonomous:

Using a custom localizer and navigation algorithm, the robot is able to autonomously move across the field to various waypoints and perform specific tasks.



PROGRAMMING

Our programming philosophy was to solve all problems ourselves without the use of external resources, which is why every algorithm and math equation associated with the robot's program is custom-made.

Odometry:

Beneath the robot are 3 dead wheel odometry pods that measure movement relative to the field tiles. These measurements are provided in encoder tick deltas, which accumulate to provide the exact position of the robot. Using calculus, we derived a formula for closely approximating the movement of the robot over each loop time (~30 ms). The equations below are derived from the graph on the following page. We also use this information to calculate the heading of the robot, as this decreases loop times since reading encoder ports is significantly faster than using the internal IMU (this is further assisted by our custom hardware caching system). However, data from the IMU is merged periodically to maintain absolute heading.

ODOMETRY EQUATIONS

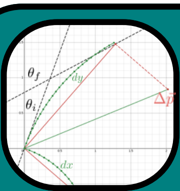
$$\Delta x = \int_0^{\Delta o_y} sin(\theta_i + \frac{y\Delta\theta}{\Delta o_y})dy + \int_0^{\Delta o_x} cos(\theta_i + \frac{y\Delta\theta}{\Delta o_x})dx$$

$$\Delta y = \int_0^{\Delta o_y} cos(\theta_i + \frac{y\Delta\theta}{\Delta o_y})dy + \int_0^{\Delta o_x} sin(\theta_i + \frac{y\Delta\theta}{\Delta o_x})dx$$

$$\vec{p}(t) = \langle \int \Delta x dt, \int \Delta y dt \rangle \approx \langle \sum \Delta x, \sum \Delta y \rangle$$

$$\Delta\theta = \theta_f - \theta_i = \frac{\Delta o_{y,r} - \Delta o_{y,l}}{2\pi T}$$

PROGRAMMING



From this diagram, assuming a relatively constant angular velocity for the duration of the loop cycle, the above equations come trivially as we integrate over the odometer reading with a varying angle.

Furthermore, using the difference between the measurements of the two parallel odometry pods, along with the odometer center $\vec{o}$, we can compute the robot's orientation and absolute position with any configuration of modules, making the hardware design process easier.

Custom Debugging Tools:

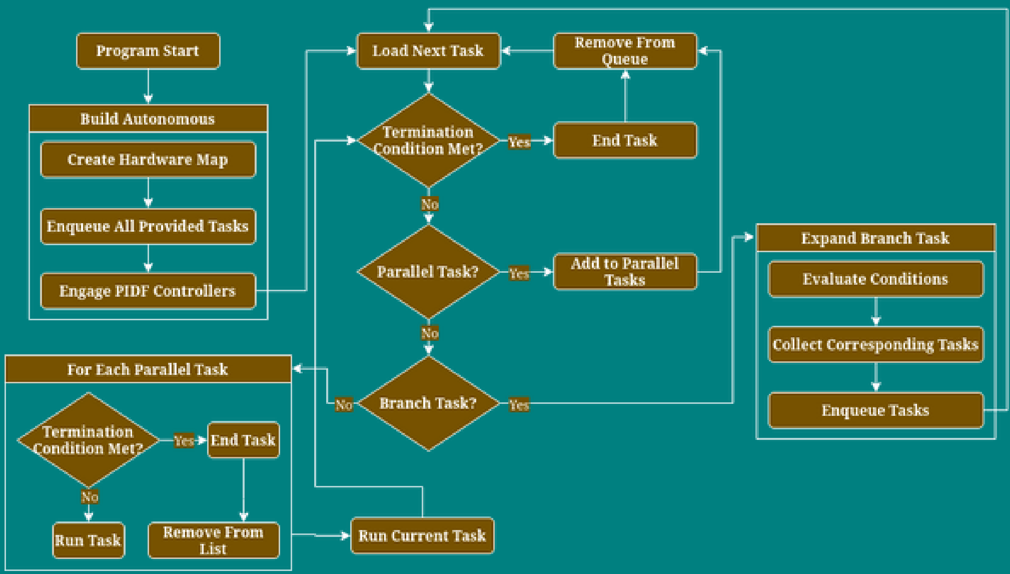
To aid in the development of software, we developed several debugging tools that ensured both safety and efficiency. Firstly, all hardware is wrapped with a custom class that manages interactions between software and hardware; this allows us not only to optimize our code, but also control access to individual motors and servos directly. If we wish to observe the value of any given peripheral without energizing it, the system allows this to occur as well. Furthermore, we can overwrite the perceived values from sensors, and, when combined with the supported dummy motors/servos,we can effectively simulate the entire robot through code without actually needing the hardware.

Disable Motor
↓
Motor cannot be powered
↓
Values being assigned can be read
↓
If desired, encoder readings can be artificially controlled for debugging

PROGRAMMING

Autonomous Task Scheduler:

From previous experience, writing new programs for every autonomous is extremely tedious, so we created our own task scheduler to resolve this issue. We define an opmode wrapper interface that conceals the standard execution functions and simply expects code to build the autonomous and nothing else. Thus, our autonomous programs read like a checklist and are really easy to create and modify.



The flowchart above describes the core of the autonomous executor, which is customized to suit our needs and overcome previous challenges we faced. To program an autonomous, we provide a sequence of tasks that we want the robot to run, and our algorithm will handle the underlying hardware interactions.

PROGRAMMING

TeleOP Enhancements:

To allow drivers to focus on strategy and gameplay, we attempted to minimize the number of buttons required to complete various tasks.

- Every controllable component has preset positions that are associated with various states, removing the need for drivers to align subsystems manually.
- The driver and robot share control over the states of subsystems; the driver indicates the desired action through the press of the button, and the robot automatically determines the best way to accomplish the desired action.
- Sensor data is consistently being used to aid driver control; this includes (but is not limited to):
  - Merging the IMU and odometry data to obtain the orientation of the robot, which allows for field-centric driving and controlled heading.
  - Motor encoders to maintain or approach desired angular positions.
  - Some buttons serve more than one purpose, and the robot can use position data to determine which action should occur when these buttons are pressed.
- The robot makes its own decisions on what the driver's intentions are. For instance, if both drivers set their subsystem to the transfer position, the robot can automatically facilitate a transfer with no driver input.

OVERVIEW

This year:

We had a member represent our team at the state capital during the off season. There, we advocated for STEM related legislation and demonstrated the FTC program.

Gameplan:

We were a three-person team this season. We tried our best to get a fully CAD'd robot before the season started. We built up a lot for programing that we were able to carry over into this season. Our programmer built up his own roadrunner that he has used for the past few seasons. We set out to get a robot that can score pixels on the bar or in the basket, and as soon as possible.

Volunteering:

We reached out to help at the league meets volunteering over 80 hours to help inspect and referee the league matches. We also helped take down the fields after the event finished.

What we value:

We value sportsmanship and being the best teammates we can. We congratulate our teammates whether we win or lose our game.

Team Challenges:

Our team was faced with a challenge to grab samples from the center cage. We decided to think outside of the box and create 2 arms. One to grab a sample and transfer to the other arm. This speeds up the process so we can score faster. We use the transfer to move the sample from one arm to the other, which saves time. Our team also got to machine our side plates to near perfect dimensions with the use of CNC machinery. We learned that we would need to slow down the intake claw speed to increase accuracy to grab the sample.