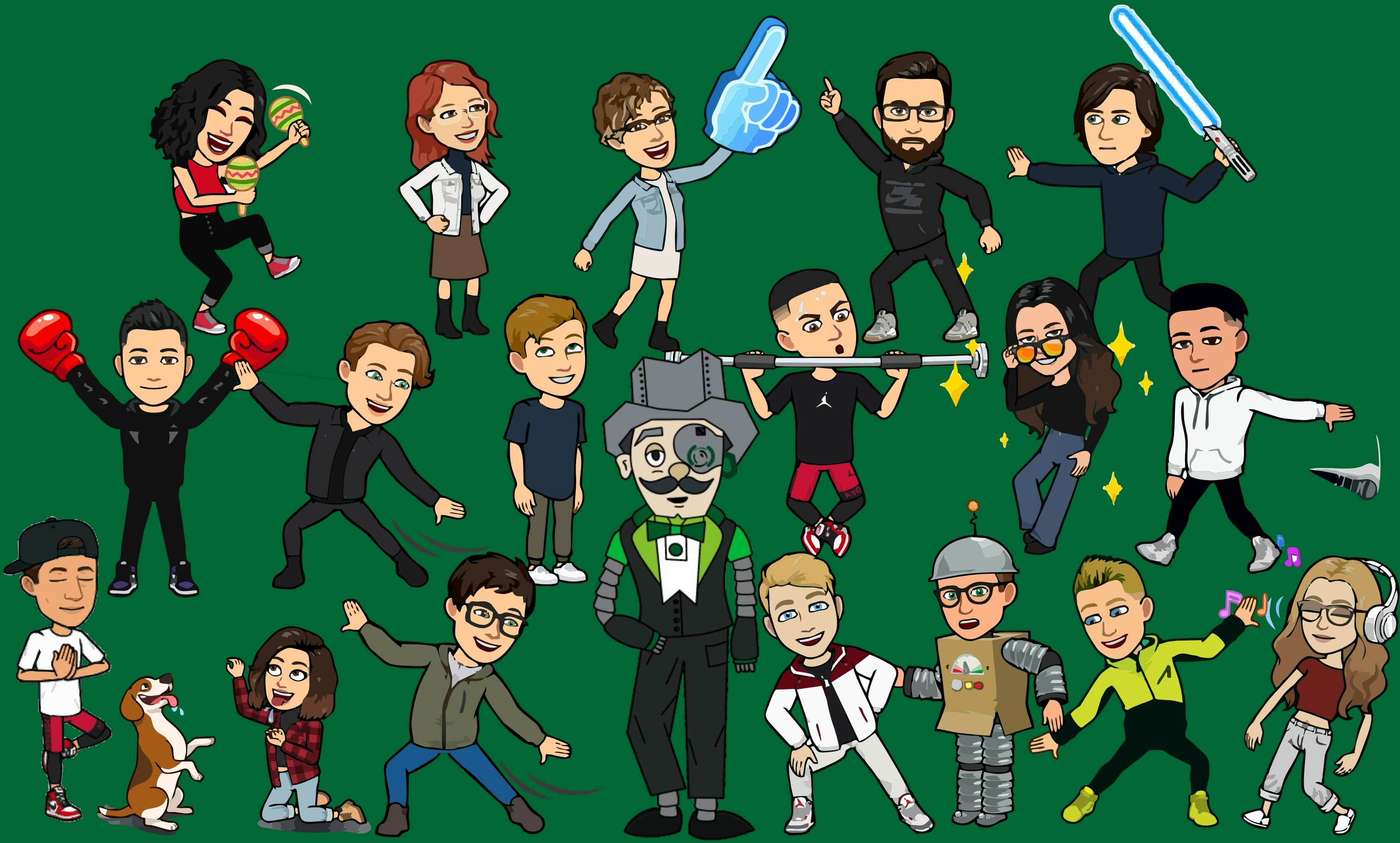
RO 140 15993

THE ENGINEERING PORTFOLIO



inf(1) Robotics

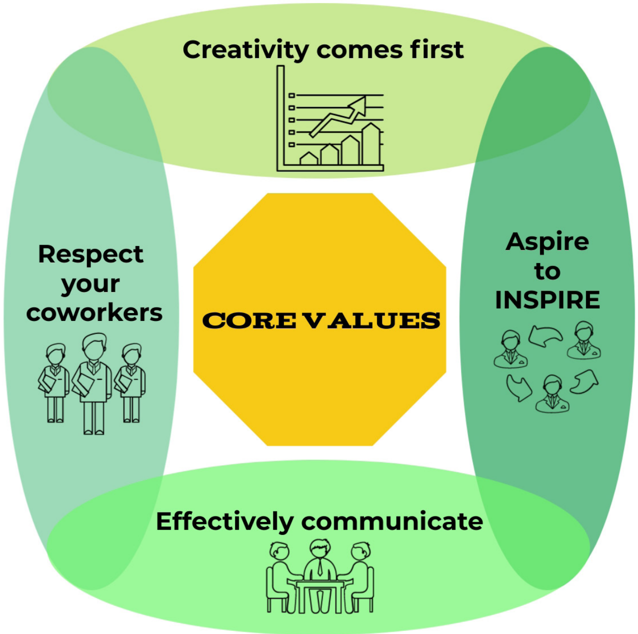
1.Team Section

infO(1)Robotics is a First Tech Challenge Team currently consisted of 15 members, pupils at National College "I.L.Caragiale" Ploiesti and 3 mentors. Our journey started in 2018, the team participating now in its third season of FTC.

The team is divided into 2 branches: non-technical and technical and 4 departments: PR, Programming, Building and 3D Design.

Our team is based on several principles that also define our working process and relationships, whether internal or with other external entities and teams. We chose to divide them into 3 major categories:

- ***Mission*** - using robotics and STEM Fields in order to help humanity overcome its difficulties.
- ***Vision*** - Having a vision means we have a clear sense of purpose, we see a much larger picture of our business, than simply setting and reaching short-term goals and tackling problems as they come along.
- ***Core Values*** - In order to make sure we would become a united and successful team, we established from the beginning that we would keep some ideas in mind every time we have to do or create something.



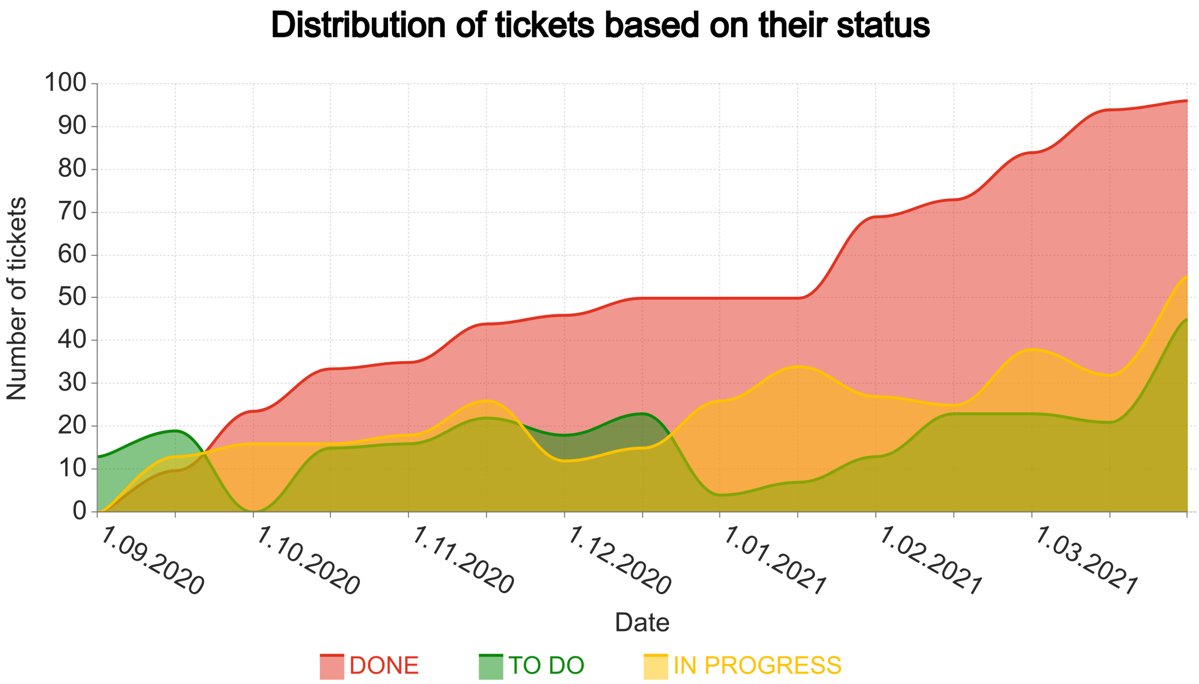
Last season, we started to implement Agile Management in the way we worked. Seeing how many benefits it brought us, we decided to keep using it. We combined the two forms of Agile: Scrum and Kanban and obtained the Scrumban Method. We used the prescriptive nature of Scrum to be Agile and the process improvement of Kanban to allow the team to continually improve its process.

In order to achieve the best results we could, we concluded that implementing a week or two-week sprint workflow will allow us to complete a significant number of assigned tasks while having a high productivity rate.

We started using Trello, an app where we could organize and prioritized our tasks by using a color-coded label system.

At the end of every sprint, we had a meeting in which we discussed our then-finished sprint and assigned new tasks to team members.

Below, you will see a graph showing the evolution of the tickets of our team, taking into account the number of solved, planned and ongoing activities, from all the team departments.



1.1. Objectives & Key Results (OKRs)

For this season, we decided to use the OKR framework to organise our workflow, since it helps us build simple and measurable actions. Moreover, it provides the means to continuously measure our progress.

Objectives define *where we want to go.* They are short and inspirational.

Each objective has a set of *measurable deliverables*, known as ***key results.*** They are used to measure our progress towards each goal.

In the diagram below you can see our OKRs for this season:

2. Business Plan

SWOT

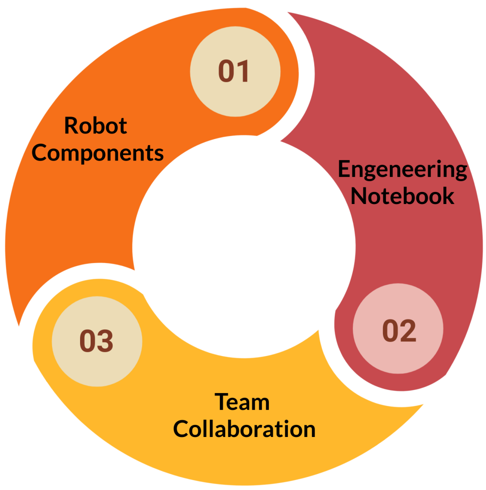
S – Strenghts
- Efficient project management (Agile)
- Young members to carry the team further

O – Opportunities
- Sponsorships
- Better robot code

W – Weaknesses
- Social media stagnation
- Slower robot

T – Threats
- Increased competition
- The influence of Covid-19

2.1. Strategic Plan



Short-term strategy

We began by structuring an initial blueprint of the robot and decided which parts of the team influence it. The departments that help the construction of the robot are ***Building, 3D Design*** and ***Programming.***

Afterwards, we merged all of our departments in order to create the ***Engineering Notebook***, another important asset to our team in the competition, besides the robot.

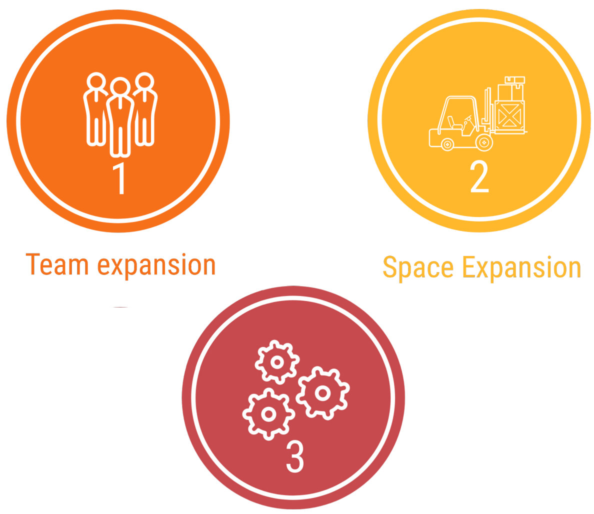
Finally, we developed the fundamental part of our strategy which was ***collaborating within the departments as well as with each other.***

Long-term strategy

Team expansion is in reference to extending the infO(1) Robotics community in several areas. This means cresting a new juniors team to compete in FLL and acting as mentors for them.

Space Expansion describes the fact that we wish to move to a larger location and also use more innovative technology to simplify our work (Smart Office model).

Robotics Development means working on more advanced types of robots that could prove useful in other domains than FTC such as an ecological robot.



Team expansion

Space Expansion

Robotics Development

2.2. Curent actions

Since last year we have made several changes in how we function as a team, this being best reflected in the tasks we have done until this point.

We focused on improving our approach towards various fields in order to obtain the best results possible and to enhance the number of people that know about our team's activity.

No.	Action	Tasks	Completion Date
1.	Workshops	Opportunities to teach our colleagues about what we do	All around the current season
2.	Donation of visors	Donations of visors made with the 3D printer	March 2020
3.	InfO(1) Talks	The second edition of our interactive conference aiming to spread STEM and other topics related knowledge	May 2021
4.	PO1Cast	Discussions with experts in the STEM field	April 2021

2.3. Future actions

In order for our team to spread important information about robotics and to enlarge its popularity, we have decided to partner up in the future with businesses or companies that work in other fields.

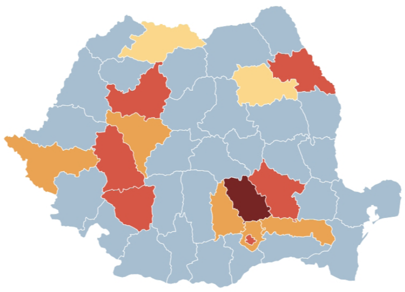
We decided to expand our activity in the following domains:

- Charitable events
- Health workshops
- Physical activities

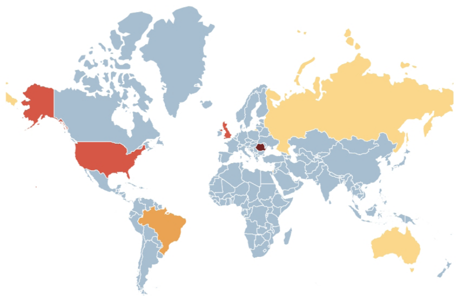
No.	Target	Actions	Completion Date
1.	Events' Topics Enhancement	Charitable, heath and sports	Next season
2.	Ploiesti Tech Talk	Debate competition taking 3 days	Next season
3.	Social Platforms	Social media development	Next season
4.	Junior team	A team to participate in "First Lego League"	1-2 years
5.	Robotics hub	A place for teams to co-work easily	4 years
6.	The Eco-Bot	Robot that cleans the environment	7 years
7.	Smart office	Completely technology friendly working space	10 years

2.4. Marketing analysis

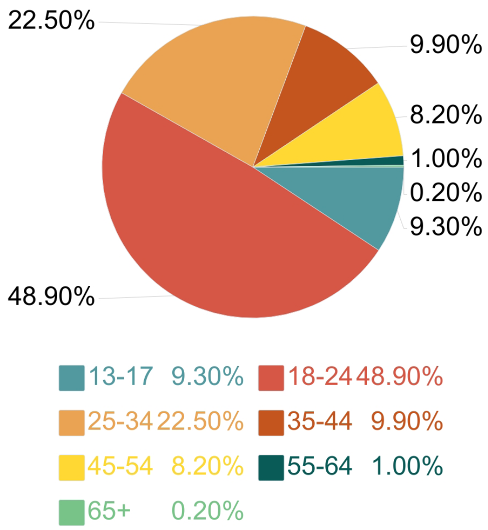
National audience



World audience



Age distribution



Social media

"InfO(1)Robotics" site
Directly documents our overall activity

~1000 followers
62 posts
Over 1200 stories

217 subscribers
Max views 1.7 K
P01Cast - our podcast

~1500 followers
Max views 3.1 K
8 events

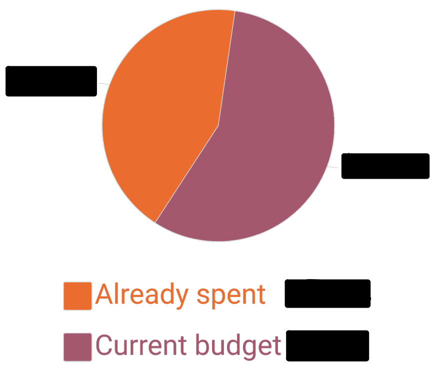
79 karma points
Better comunication with international teams

2.5. Financial Statement

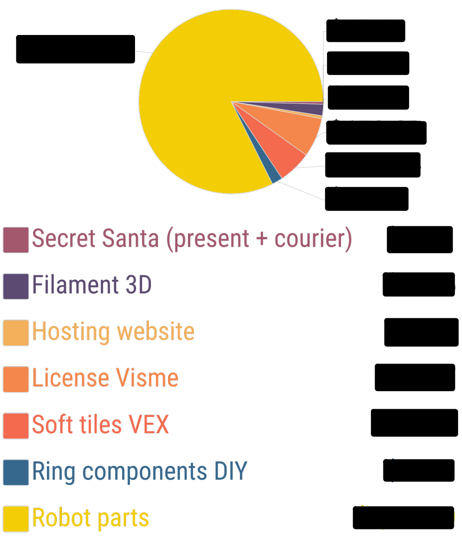
Source of funds

[redacted]
The complete distribution of funds for this season

Sponsor
PION Impex [redacted]



Expenses



3. Outreach

Our outreach philosophy is to raise awareness about STEM and robotics and to connect as many social categories as possible. Not only that the community we created is now up to date to the impact that technology has on our everyday life, but they now unconsciously connect STEM with the name of ***infO(1)Robotics***.

>1350

PEOPLE REACHED

Part of our events, the ones that we directly organized or taken part in, there were more than 1350 people involved, that contributed to or benefited from the whole project.

3.1. Mentoring

This year we are mentoring info(1) Technologies, a team that just started its way into the FTC community. Non-technically, we've helped them with their organization and shared some aspects from our FTC experience.

Furthermore, regarding the technical part, we've helped them with all the necessary resources to build a robot and offered our insight when it comes to a proper development of the building and programming process. We love seeing other people, most of them colleagues from our school, involved in the FTC community, understanding what robotics and FIRST promotes and represents, thus providing our complete experience and background in their benefit.

3.2. Events

This year has been a real challenge for us to organize events due to the pandemic. It was complicated to connect with people remotely while we were on the other side of the screen. Although, we managed to organize 3 major projects: the PO(1)Cast, the infO(1)Scrimmage (Demo) and the 3D Printing Campaign for March 1st that we ran in our school for the teachers and other employees.

Especially because of the pandemic situation and current context, we decided to focus on three major events, each of them emphasising an important part of FTC: the PO(1)Cast reveales our appetite for learning and sharing our STEM values in the community, the Scrimmage demonstrates our involvement in the FTC community and passion for robotics, while the March 1st campaign showcases our devotion and gratitude to the ones that constantly help and support us.

3

MAJOR PROJECTS ORGANIZED

We organized 3 major projects: the PO(1)Cast, the Demo and the March 1st Campaign.

Throughout the Ultimate Goal season, infO(1)Robotics spent numerous hours carefully planning the events and deciding the best solutions for our problems regarding the outreach activities.

200

HOURS SPENT ORGANIZING

The team members spent more than 200 hours organizing these events and making sure every detail is perfect.

PO(1)Cast

One of the most important events we held is the PO(1)cast. The podcast consists of online conversations with certain guests. In the first edition, our mentors participated as guests and we discussed their experiences throughout personal and professional lives, as well as their direct input regarding the FTC mentorship: how this changed their perspective on STEM and teamwork and what other challenges they encountered during the process.

Screenshot from the first edition of PO(1)cast

infO(1)Scrimmage

During the weekend of April 3-4, infO(1) Robotics team organized the second edition of our demo, following its first edition that took place in the Skystone season. A number of 19 teams have participated, from both Romania and the Netherlands. This included some international teams which we reached to beforehands due to their experience. We were really happy to collaborate with and learn from them.

Screenshot from the InfO(1)Scrimmage Closing Ceremony

25

EVENTS PARTICIPATED AT

During the Ultimate Goal season, infO(1)Robotics participated through its members at 25 events, workshops and conferences.

Furthermore, our team was actively involved in several events organized by Natie Prin Educatie Association, or the other FTC Romanian teams, thus developing our knowledge, creating new long-lasting bonds and supporting the organization of events during the current situation.

3.3. collaborations

We focused on collaborating with other teams, either from Romania or other countries, as we can exchange ideas and perceptions on critical points during the season.

Quantum Robotics*(#14270)* is one of the few teams that we can always rely on. They have a lot of experience and they are always happy to help us as we are to help them. This year we decided to invite them to our second episode from our team's podcast series.

Another really important team that we collaborated with this season is Pink to the Future *(#3954)*, from the Netherlands, with prolonged experience in FTC. We were glad to have them participate at our demo, especially because we discussted with them some robot ideas, teamwork skills.

20

TEAMS COLLABORATED WITH

We collaborated with more than 20 teams for our podcast, demo and workshops.

4. Building

The philosophy that we adopted when building our robot, mainly relates to the core values of our team and of FIRST. However, we thought of a series of aspects that are important to effectively build this season's robot:

- ***Efficiency:*** Participating at a competition where the complexity of the proposed themes appear to get more difficult with every season, the mechanisms that we use must also raise up to the expectations. Moreover, during the 2 min and 30 sec of the match, the robot must complete a large number of actions in both the Autonomous and Driver Controlled Periods. Having that in mind, our building department, in collaboration with the 3D Design and the Programming Departments, have tried to create a functioning robot, that performs all the required tasks to gain as many points as possible.

- ***Constant improvement:*** The progress is a must in anything we perform, if we want to gain knowledge and learn from what we do. Even if there are moments where the team seems to be going in a wrong direction or if we do not win a certain prize, we must continue working, as improving is maybe more important than everything.

- ***Simplicity over complication:*** Our team is the adept of neat and smart working, rather than complicate our tasks for no reason. Thus, we applied this mindset to the design of the robot and its parts.

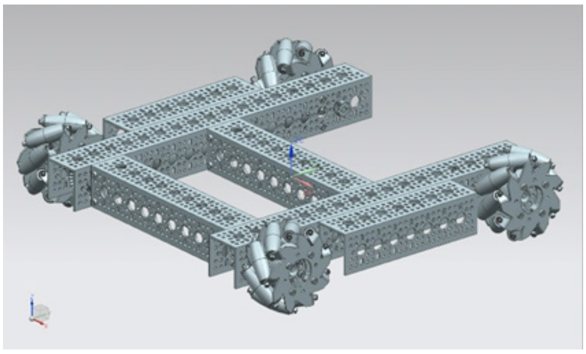
Our robot consists of three main systems:

1. The drivetrain: We are using the goBilda Strafer Chassis Kit because of its advantageous mounting spaces and 90 degree transmission which can be extremely useful in our building journey during the season.
2. The intake: We tried to create a robust mechanism which can gather the rings fastly and consistently, without worrying about the possibility of the system's damaging.
3. The outtake: Given the fact that this season our robot needs to constantly shoot the rings with great precision and rapidity we adopted a strategy which would maximize our scoring capability in autonomous and driver-controlled periods.

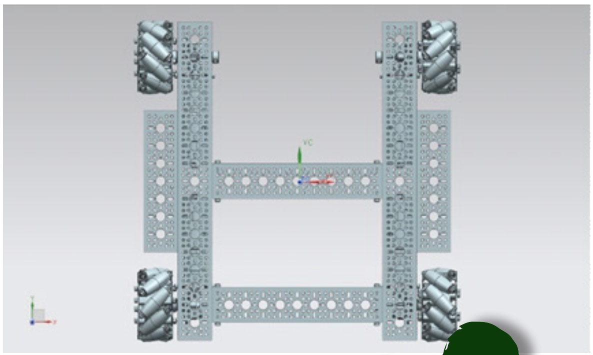
4.1. Drivetrain

- ***The chassis***

As seen in the pictures below, we have 6 U-Channels from goBilda (432mm/216mm/192mm Length), which gives us plenty of mounting space for our mechanisms. Moreover, the two 216mm U-Channels that merge with the other big two Channels gives us the possibility of running the wheels' motor cables through them, in order to have a more simplist look of the robot.



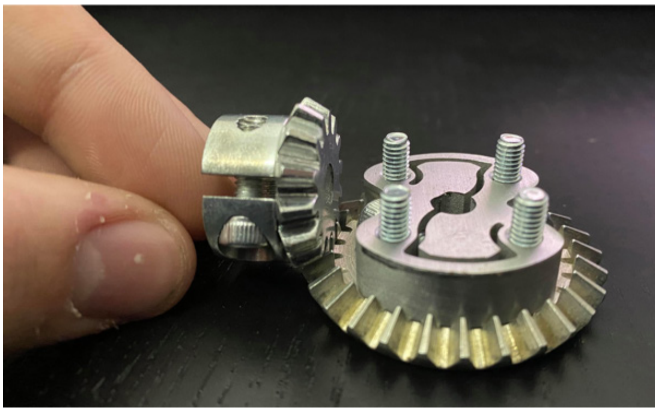
Chassis side-view & top-view



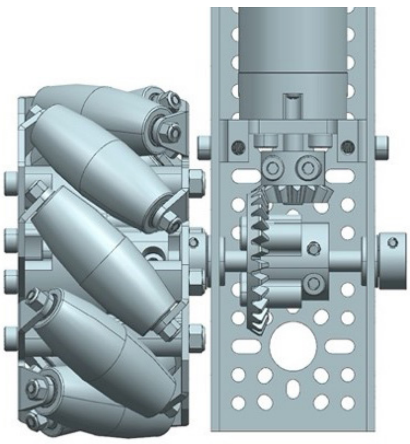
- ***The power transmission***

Having the possibility of saving space with our transmission the Strafer Kit came to our help. Using two bevel gears with different diameters (one is two times bigger than the other), we managed to build a robust system that has a low possibility of failing.

One downside that this system has is the fact that our total ratio of the motors went up to ***27:4:1*** and the total rpm halved at ***217.5*** because of the different diameters of the bevel gears.



Bevel gears ensemble



Power transmission final design

It is a system that drives all the speed and power of the 5202 Series Yellow Jacket Planetary Gear Motor (13.7:1 Ratio, 435 RPM, 3.3 - 5V Encoder) to 2 more sets of wheels, one of the set has Compliant Wheels and the other has Compliant Wheel 8 mm Bore 35 Durometer Green.

The motor is fixed on the chassis, using a 90 degree transmission. Afterwards, the power is directed to a 2101 Series Stainless Steel D-Shaft (6mm Diameter, 100mm Length) where a 3310 Series 8mm Pitch Aluminum Hub Mount Sprocket (14mm Bore, 14 Tooth) with a 1309 Series Sonic Hub (6mm D-Bore) is situated. The sets of wheels are moving in the same direction with the owing to the 3309 Series 8mm Pitch Plastic Chain Link mounted on the Pitch Aluminum Hub Mount Sprocket which is situated on the first shaft.On the shafts lateral parts of both sets's there is a 1611 Series Flanged Ball Bearing (6mm ID x 14mm OD, 5mm Thickness), which is 3D printed and a Pitch Aluminum Hub Mount Sprocket with a Sonic hub.

4.2. Intake

The final design for intake system is composed of two big pieces. One of them is a ramp fixed on the intake mechanism and the other is another ramp which transports the rings into the outtake system.

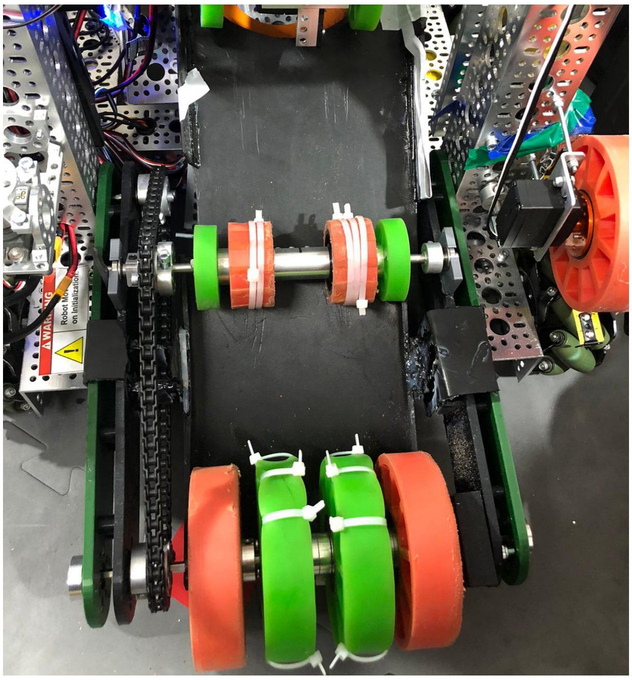
The first part of ramp, the one that it's mounted on the actual intake mechanism is used to "guide" the rings into the next ramp that has four compliant wheels, which have the role of transporting the rings into the outtake. The first ramp is mounted directly on the intake because we needed to fit the dimensions specified in the game manual.

Moreover, we thought that it would more effective if we would have a longer system because we can fit more wheels that will drive the rings in the outtake. Also, we have a Hex motor fixed on the upper part of the intake systems which helps with the transfers of the rings.

Some problems that we have encountered during the building process of this system are:

- Sometimes the rings would get stuck between the first ramp and the big compliant wheels because there was too little space for the rings to go pass (we solved this problem by making the ramp thinner at its end using a box-cutter)

- When we first mounted the falling part of the intake the compliant wheels would be too high to catch the rings (the solution was quite simple, because the only thing that needed to be done was to lower the system)



Intake top view

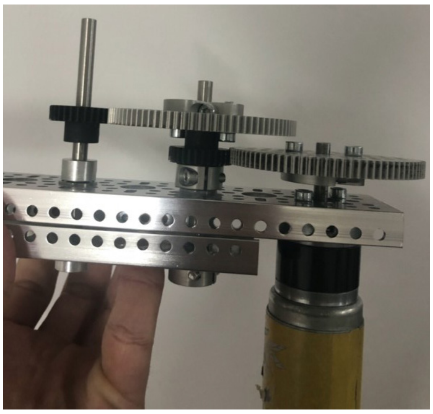
4.3. Outtake

- ***The 9:1 gear ratio***

We figured out that our 5202 Series Yellow Jacket Planetary Gear Motor (5.2:1 Ratio, 1150 RPM) powering the outtake would not be enough to launch the rings into the high goal, or even in hit the power shots with it. Thus, we came up with a system that can multiply our RPM 9 times, which means that we can get a total of 10350 RPM.

How does it work:

Basically we have a total of 4 gears, two with 60 teeth and two with 20 teeth. Mounting one of the big gears to the motors means that the particular gear will spin with an equal speed as the motor’s shaft. But, if we add another small gear to the system, the RPM will triple because the diameter of the small one is three times lower. We repeated this process again and this is how we managed to get a 9:1 gear ratio.



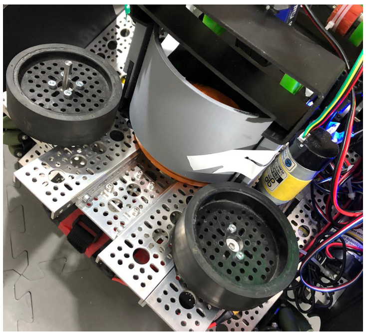
Outtake gear ensemble

- ***The launcher***

The one wheel launcher: this system is almost identical to what we have now, but its only problem was the fact that it did not shoot with precision. At higher speeds it is quite difficult to approximate where the ring will go, especially when you have only one wheel propelling it.

The two wheel launcher: this system has another wheel mounted symmetrical to the other powered by the motor, but it spins freely. This helps with the precision and consistency of the mechanism. Moreover, it also helps with the speed of the rings, due to the fact that the two wheels tighten them before shooting them into the power shots or high goal.

In order to launch them properly we needed something to push them into the outtake, so we used a goBilda servo for this job. We have printed two pieces that hold the rings together (the black and the grey one). Under them there is a servo that sends the rings into the outtake. The process is quite simple, because the only thing that needed to be done was to mount a metallic piece on the servo and set its limits in order to move the same every time.



Outtake launcer

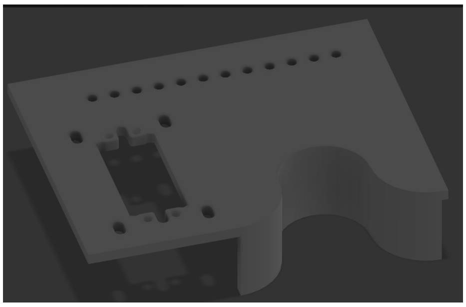
- ***Additional Parts***

Wobble Arm

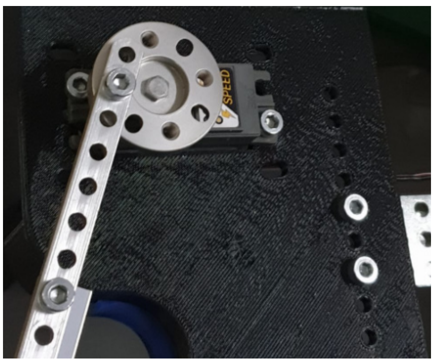
This system has 2 parts. The first one is the one that takes the wobble and holds it and the second one lifts the whole set so that the wobble is delivered in the dropzone. The first part is made for the secure delivery of the Wobble Goal in the target zones and then for the dropzone. After an analysis of the tasks that the robot must fulfill, we realized that it must complete a number of conditions:

1. Safe delivery
2. Large wobble pick-up area
3. Fast delivery of the wobble in both target zone and dropzone.

After establishing these conditions, we came up with the idea of creating a designated 3D part. This component has a specific area created in wobble size so that the game element can be pushed by a servo placed on the printed part.



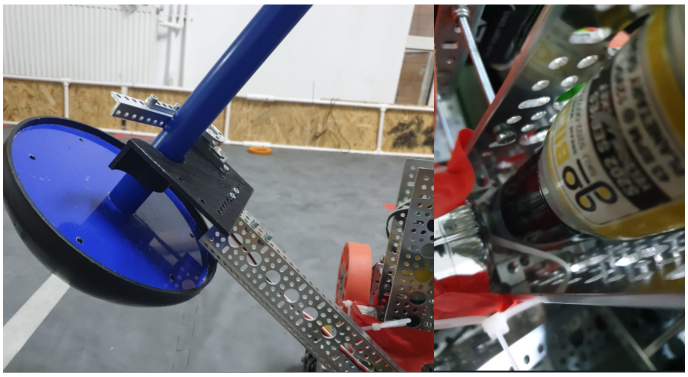
Wobble 3D-print



We encountered only one problem with this section. The area where the wobble was pushed by the servo being made of PETG does not provide the necessary adhesion. To solve this we covered the area with a few pieces of rubber for a better grip.

For the second part we also analyzed what task it has to perform. We knew that this part of the ensemble needs power as it has to lift the wobble, however it also needs to be placed in a high point to lift the wobble over the perimeter.

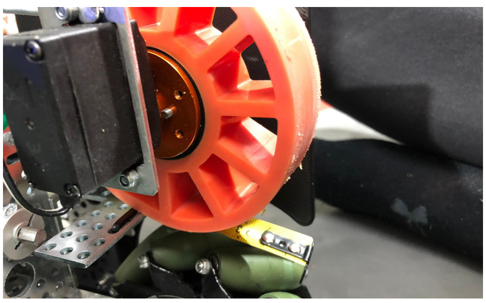
The 5202 Series Yellow Jacket Planetary Gear Motor 43RPM (139: 1 ratio) with a ratio we added of 2:1 fits this criteria. This motor spins a shaft attached to an 1121 Series Low-Side U-Channel, thus providing the right height for dropzone delivery. In this part we realized that we must be very careful with the Gear Motor assembly as it has a very high force and used carelessly can damage the auxiliary parts.



Wobble working process

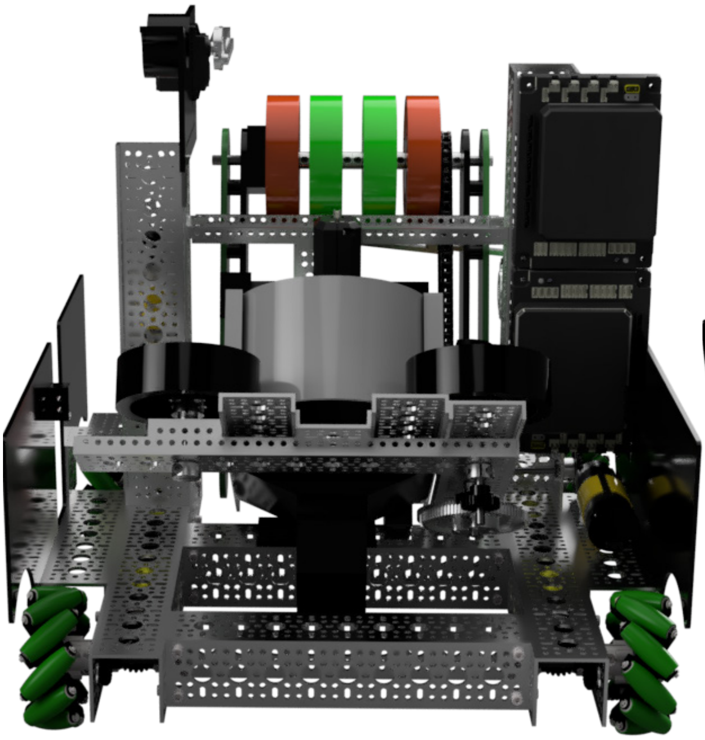
The measuring tape

We have encountered a problem during the 4 rings autonomy. Our robot did not have time to park because the other tasks were taking too much time. We figured out that the best solution would be to design a mechanism that can park the robot without actually moving it from the point where the time runs out. We took an old measuring tape that we had and mounted a servo that we set to move continuously in one direction until it is stopped in the code. This means that we can park the robot even if it is not above the white line, because the measuring tape is considered part of the robot.

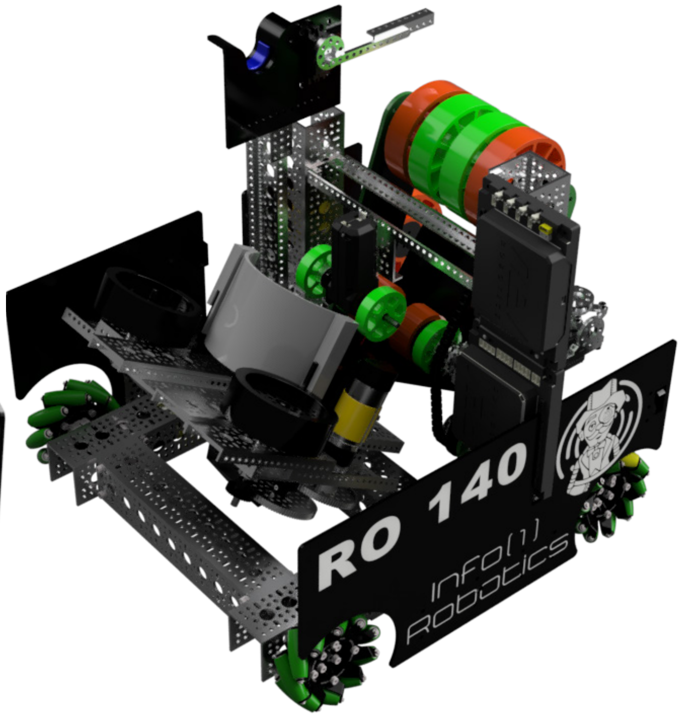


Measuring tape ensemble

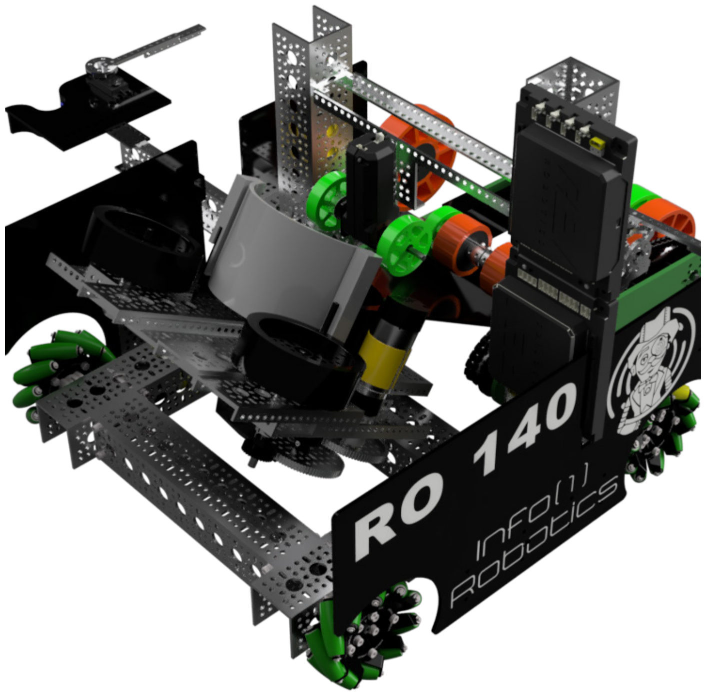
4.4. Final robot design



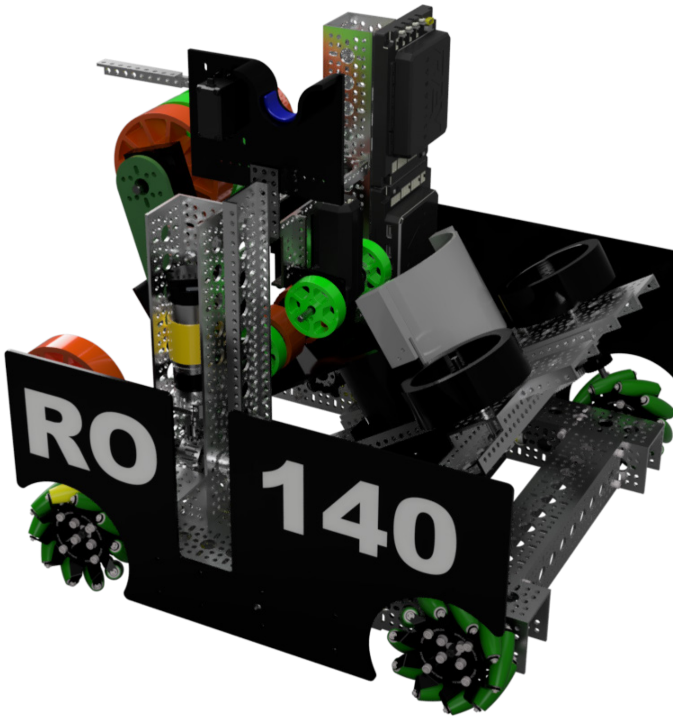
3D Design of the robot
FRONT VIEW



3D Design of the robot
FRONT RIGHT VIEW



3D Design of the robot
(open mechanisms)
FRONT RIGHT VIEW



3D Design of the robot
FRONT LEFT VIEW

5. Programming

This year we've experimented with multiple libraries and frameworks. Initially, we've tried to create everything from scratch, but we've soon realized that it would take a lot more time than we had.

After some research we've found a library named FTCLib. It includes wrappers for most of the classes of the official SDK, a solid framework architecture and a scheduler for different commands (see Section 5.2.). It worked great for teleop and programming the mechanisms of the robot, but its pure pursuit implementation is rather lackluster.

Although FTCLib supports Holonomic Odometry using dead wheels and Pure Pursuit, we did not manage to get it working the way we wanted. The kind people on the official FTC discord server recommended us Roadrunner (see Section 5.3.), since it is easy to use. After some trials and errors, with the help of *learnroadrunner.com* we managed to get really accurate movement.

To speed up our development process we've implemented FTCDashboard, a tool that allows us to monitor different values and properties of our robot and plot them over time. In addition to that, we can also change static variables without recompiling the code.

5.1. Version Control

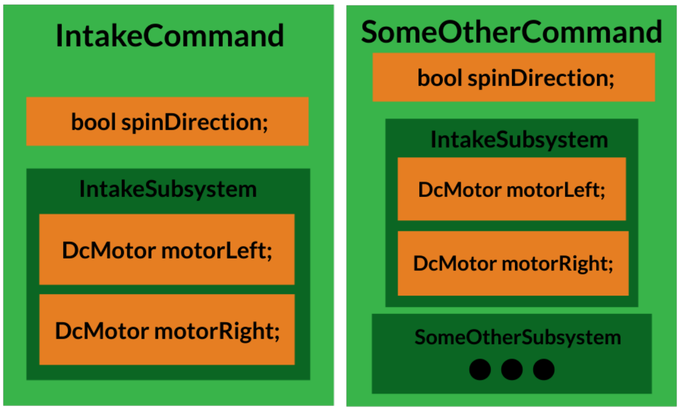
To make our lives easier we've used Git as our version control system and uploaded all our code on GitHub. Version control is a system that records changes to documents and large projects over time so that any specific versions can be recalled. Each version is represented as a „commit" with a clear description. Each commit is integrated into a branch. Branching allows us to use sandboxing when trying different ideas or features at any given time.

5.2. FTCLib

FTCLib is a library designed to include everything you will need for FTC programming. The project was initially meant to be a port of WPILib, a library used by FRC teams. The goal of FTCLib is to improve the initial programming experience for new members as well as greatly enhance the efficiency of code for veterans.

FTCLib uses a command-based system. The specific command system that FTCLib uses follows a declarative programming style. This allows for cleaner code where it is easy to bind commands to triggers or schedule them.

The Command system has two main elements: commands and subsystems.



Example of an FTCLib configuration

- ***Subsystems***

A subsystem is the basic unit of robot organization in the design-based paradigm. Subsystems encapsulate lower-level robot hardware (such as motor controllers, sensors, and/or pneumatic actuators), and define the interfaces through which that hardware can be accessed by the rest of the robot code. Subsystems allow users to "hide" the internal complexity of their actual hardware from the rest of their code - this both simplifies the rest of the robot code, and allows changes to the internal details of a subsystem without also changing the rest of the code.

- ***Commands***

A command defines high-level robot actions or behaviors that utilize the methods defined by the subsystems. A command is a simple state machine that is either initializing, executing, ending, or idle. Users write code specifying which action should be taken in each state.

Command code must specify what the command will do in each of its possible states. This is done by overriding the `initialize()`, `execute()`, and `end()` methods. Additionally, a command must be able to tell the scheduler when (if ever) it has finished execution- this is done by overriding the `isFinished()` method.

- ***Scheduling commands***

A command can be scheduled by simply calling the `schedule()` function or by binding it to a trigger, such as a button from a driver's controller..

The scheduler also ensures that only one command will use a subsystem at a time using the `addRequirements()` function in the command to register the subsystem it uses.

For example we use the `toggleWhenActive` trigger to activate or deactivate the intake and outtake and we use the `whileActiveContinuous` method to move the arm up and down.

You can also combine triggers and create custom ones, although we didn't use this functionality.

- ***Custom Encoder Wrapper***

Since the Rev Control Hub/Expansion Hub has only 4 encoder slots per unit, we were limited to only 2x4=8 specialized encoder ports. Those slots were already occupied by the dead wheels and drive motors, and we needed one more encoder for the outtake, for reliability in the face of voltage variation.

Thankfully, we haven't used any digital channel ports and, in theory, the encoders were compatible with this port, although we would have to build a custom Encoder class which would interpret the input signal and resolve it to ticks. And it worked! We've set up a thread to monitor the changes in the digital input and fed those values in an Encoder wrapper made by FTCLib. This way, we were able to use PID Control on the outtake motor with one extra encoder on the robot than the hardware provides builtin.

5.3. Roadrunner

Roadrunner is the de-facto FTC library for robot motion. It provides extensible movement functions and easy-to-use configuration wizards. It is also built to work with FTCDashboard, resulting in an shorter time while debugging.

The library supports TeleOP and Autonomous workflows, and even provides a quick start project which implements motion profiles for common robot configurations, such as mecanum drive or 45-degree oriented Omni Wheels.

- ***PID Control***

A PID controller is a mechanism designed to catch errors caused by inaccuracies in the robot's motion and correct them in different manners using the constants P(proportional), I(integral), D(derivative). The REV Control Hub has such a controller built-in.

P - if the error is large and positive, the control output will be proportionately large and positive

I - if there is a residual error after the applying proportional error correction, it applies correction by taking into account the history of the error variation.

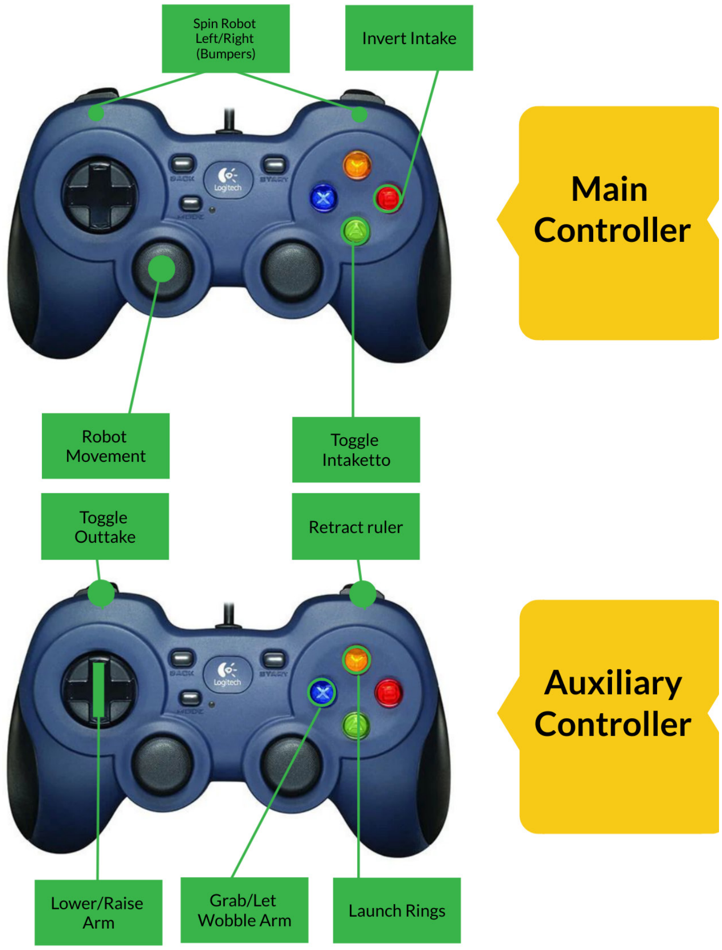
D - it basically predicts future errors and tries to minimize them

Roadrunner includes a PID tuner opmode that comes in handy.

5.4. TeleOP

We make extensive use of the commands system. For each action we want our robot to be able to do, we create a command and bind it to a gamepad button. These commands are then scheduled accordingly (either running all the time or by pressing on specific buttons) using the CommandOpMode provided by FTCLib. Using this custom opmode means less boilerplate , such as getting the scheduler instance by using `CommandScheduler.getInstance()`. We strive to make the controls intuitive and responsive for the drivers.

One notable thing we've tried is making the teleop controls driver-centric. The robot would move to the right/left/up/down of the driver, regardless of its heading. Our drivers couldn't adjust in time though, so we'll revisit the idea next year.



5.5. Autonomy

- ***Autonomy based on ticks***

This is the first and most basic approach that still yields acceptable results. Each motor has a built-in encoder that measures ticks, the smallest unit of measurement for an encoder. For example, the motors we use for movement have 384.5 ticks per revolution. Using this information we can tell the robot to move N ticks to a certain position. The problem with this method is that it does not take into account errors caused by wheel slippage and uneven friction.

- ***Autonomy based on ticks, but with Roadrunner***

We didn't iterate on this idea for too long because Roadrunner actually ran worse than the simple autonomy we've had (may have been because of inaccurate measurements; more testing required), and the dead wheels were almost ready to be used by the time we've implemented this idea.

- ***Autonomy with dead wheels***

After some hands-on experience, we've evaluated that the tick-based autonomy is too unreliable and we would need to come up with some other solution. After some unlucky scrimmages and some lucky moments, we've implemented the three-wheel localization class from Roadrunner in order to more accurately tell the position of the robot and make corrections in case of error.

- ***Vision implementation***

This year we're iterating on last year's implementation of computer vision, which was using Vuforia for object tracking and Tensorflow Lite for object detection. We've reached to the conclusion that having a smaller image size increases the detection accuracy, so we've cropped the camera stream to only the relevant portion, where the ring(s) are located. This resulted in a confidence rate of over 70%, which we're happy with, considering the lighting conditions in our lab aren't optimal.