game manual 0

a guide for FTC teams

enjoy!

table of contents

Introduction

Hello! Welcome, and thank you for taking the time to read Game Manual 0, the premier resource for new and upcoming FIRST Tech Challenge teams! If you are part of a rookie or new team, we would especially like to welcome you to the FTC community. We want to make your beginning steps in FTC as seamless as possible, and offer some of our own experience and advice. Our guide was written by members in FTC teams all across the nation, many of whom have competed at the worlds level.

The goal of Game Manual 0 is to create the most extensive guide out there for FTC, as online and physical resources for FTC are few and far between. A major inhibitor for newer teams is the lack of a knowledge base, as well as not knowing any experienced teams who might offer advice and support throughout the build and competition season. Thus, they are left in the dark to figure things out, seemingly with no way out. While the journey of learning does not have shortcuts, GM0 seeks to address these shortcomings by providing a starter’s guide to the hardware and software in FIRST Tech Challenge.

When perusing this guide, it is important to keep in mind the authors' perspective. Many, if not most, of the teams who contributed in the writing of this guide are worlds level teams in the upper echelon of FTC. This means that our recommendations are almost solely from the competitive advantage standpoint. For example, it is possible for a team to learn just as much from using a Tetrix kit compared to an Actobotics kit. Just because there is a competitive advantage to one does not translate into a learning disadvantage for the other. Please take our recommendations with a large grain of salt if your team is primarily based around learning (which we totally encourage and support!) instead of the competitive robotics scene. There is so much to learn, and we would love to share our knowledge with you. We want to help all teams in FTC, but as our experience has been on the competitive side of the fence, certain parts of the guide may not be very applicable to some teams.

Before diving in, we would like to add a short disclaimer: this guide is *not* about how to build a specific drivetrain, linear slide, intake, etc. The purpose of GM0 is simply to provide knowledge and advice to teams about the possible options, as well as including some tips on how to get started. GM0 has knowledge and advice, but not instructions. **It is of the utmost importance that all teams learn the right way (by trial-and-error), not by reading a step-by-step guide or instruction manual. Thus, while GM0 has plenty of advice, we do not have specific steps included. Good luck, and have fun in FTC!**

The Game Manual 0 Writers

Contributors

Game Manual 0 would not have been possible without the many contributors who have sacrificed their precious summer freetime (or merely found another excuse to procrastinate their work). GM0 had been my dream for quite some time, and as an alum tremendously grateful for all that FIRST has taught me, I wanted to create GM0 to preserve and spread the knowledge base that might have been lost with the FTC class of 2019 and beyond. For this reason, I wanted to recognize all the contributors who have helped to make GM0 a great resource for FIRST Tech Challenge teams. Feel free to contact any of the names listed below via the email (gamemanual0@gmail.com) or through the other contact emails - I'm sure that they would be more than willing to help you out!

Kleptomaniac

Managing Editors

Tom - FTC Alum - 3736 Serious Business- seminole3736@gmail.com
Frank - FTC 8581 Aedificatores - fgportman00@gmail.com
Justin - FTC 9656 Omega - evhsomega@gmail.com
Kleptomaniac - FTC Alum - 13075 Coram Deo Robotics - REV Robotics - nathanchu@utexas.edu

Hardware & Electrical

David - FTC 2753 Team Overdrive
Sam - FTC 2753 Team Overdrive
Tyler - FRC 3184 Blaze Robotics
Tom - FTC Alum - 3736 Serious Business- seminole3736@gmail.com
Karter - FTC Alum - 5975 Cybots
Neo - FTC 6710 Summation - ftcsigmas@gmail.com
Ethan - FTC Alum - 7236 Recharged Green - goBILDA
Coal - FTC 7548 SPAREPARTS
Ian - FTC 7842 Browncoats
Andrew - FTC 8417 'Lectric Legends - team@lectriclegends.org
Eric - FTC Alum - 8417 'Lectric Legends - team@lectriclegends.org
Frank - FTC 8581 Aedificatores - fgportman00@gmail.com
Justin - FTC 9656 Omega - evhsomega@gmail.com
Baylor - FTC 10641 Atomic Gears
Peter - FTC 12533 Inception - team@ftc12533.org
Nate - FTC 12897 Newton's Law of Mass
Adham - FTC 14875 LightSpeed
Dom - FTC 15692 Rust in Pieces
Ben - FTC Alum - Copperforge/WPI
John V-Neun - FRC Mentor - JVN for FTC Calculator

Software

Dan - FTC 7026 JDroids
David - FTC 7236 Recharged Green - pellarobotics@gmail.com
Frank - FTC 8581 Aedificatores - fgportman00@gmail.com
James - FTC 14298 Lincoln Robotics - ftc14298@gmail.com

Credits

We would also like to recognize the teams who have provided pictures of their robot(s) for this guide. We hope you find the pictures to be informative as real examples of what teams have achieved in the past seasons.

724 RedNek Robotics Wun
731 Wannabee Strange
2818 G-Force
2844 Valley X Robotics
2856 Tesseract
3736 Serious Business
3846 Maelstrom
4042 Nonstandard Deviation
4545 ViperBots Ouroboros
5040 Nuts and Bolts
5064 Aperture Science
5975 Cybots
6022 To Be Determined
6299 ViperBots QuadX
6929 Data Force
7161 ViperBots Hydra
7182 Mechanical Paradox Cubed
7203 KNO3
7209 Tech Hogs Robotics
7236 Recharged Green
7244 OUT of the BOX Robotics
8103 Null Robotics
8148 Aleph Bots

FRC 694 StuyPulse

8221 Cubix
8375 Vulcan Robotics
8393 BrainSTEM Robotics
8417 ‘Lectric Legends
8640 Trojan Robotics
8644 Brainstormers
8680 Kraken-Pinion
8686 Height Differential
9794 Wizards.exe
9804 Bomb Squad
9829 MakBots
9889 Cruise Control
9971 LanBros
10030 7Sigma
10641 Atomic Gears
11115 Gluten Free
11190 Mechadojos
11260 Up-A-Creek Robotics
12599 Overcharged
12670 Eclipse
12897 Newton’s Law of Mass
13075 Coram Deo Robotics

Getting Started in FTC

Welcome to FIRST Tech Challenge! As a new team, you may be slightly overwhelmed and disoriented, wondering where to start. No fear! This guide is meant to help navigate you through your beginning year in FTC.

After registration and going through the STIMS process in the FIRST website, you'll likely have to purchase a starter kit (**which we highly recommend for new teams**) and electronics kit in order to build your robot. The electronics kit is standard for all teams, but please read our kit guide to find out which kit is right for your team.

The physical components of a robot are split into the hardware and electronics section. On the hardware side, you have your drivetrain, linear extension mechanisms, a claw or intake to pick up game elements, as well as any other mechanisms specific to the game. On the electronics side, there are the two REV expansion hubs and the Android phone, along with extras such as the Servo Power Module, if necessary.

The kit guide provides information on selecting the right kit for your team. Along with this, we have included a list of tools that may be necessary or beneficial for your team to purchase. The building guides for drivetrain, linear extension, and intake go over the different routes teams can take, as well as providing advice on which to choose. However, we want your team to have the full learning experience, so we don't provide specific instructions for building mechanisms.

Here's an overview of how FTC works. Each season, Game Manual 1 will be updated sometime in the summer. GM1 contains the general rules of FTC, competition and advancement, as well as hardware and software restrictions. Game Manual 2 is released on kickoff day. It provides details about the specific game that your team will be participating in this year. Every year, teams are required to build a robot according to rules in GM1 and document what they have done in an Engineering Notebook. Each team will have at least one chance to compete, either at Qualifier Events or League Meets. At Qualifying tournaments, every team will play 5 matches. The top 4 teams are captains and form four alliances of three teams each. More information on the selection process is in GM1. The top teams will advance to the next level, generally the State or Regional Championship. Leagues are different, having a few meets where teams participate in matches. At the end, the top teams from the League Championship will also advance to the State or Regional Competition.

The Game Manual, released by FIRST every year, has two parts. Game Manual 1 is released earlier in the year, and covers general rules such as robot restrictions and parts limitations, tournament structure, and advancement. Game Manual 2 is published when the season officially starts, and provides game specific rules that change from year to year. While we cannot provide any help for GM2, here are some rules in GM1 for teams to be aware of, as many new teams run into problems during their tournament because of it.

- All teams MUST bring the Consent and Release forms to every competition. Teams will not be allowed to compete if the forms are not signed.
- The maximum **starting** size up the robot is an 18 inch cube. The robot may expand after the match starts. This sizing limit has been in place since the start of FTC.
- For the 2018-2019 season, the maximum weight of the robot was 42 lb. This may change in later games, but as of the time of writing this remains the restriction.
- Sharp corners, liquids, and certain wheels are not permitted. We advise you to use the wheels that are listed in this guide only.
- Hydraulic and pneumatic devices are prohibited.
- For the 2019
- Commercial Off the Shelf (COTS) parts are allowed as long as they only have one degree of freedom.
- **Each team must have a robot power button next to the robot power switch.**
- Each team must have a secure battery and phone mount.
- Only 8 DC motors and 12 servos are allowed. VEX 393 EDR counts as a servo.
- Modifying servos and replacing fuses are not allowed.

Principles of FIRST Tech Challenge

In this section, we have compiled some important and valuable lessons that we have learned over the years. We encourage you to read through this section in order to avoid some of the mistakes we made in the past as young teams. The principles are generally in order of importance from top down.

Here's a TL;DR of this section, followed by the full version.

- FIRST Tech Challenge is all about reliability and efficiency.
- Execution is the key to success. A bad design implemented well nearly always beats a good design with bad implementation.
- Simpler = Less points of potential failure = Less likely to break = More reliable.
- No team exists in a vacuum, take inspiration from other team's efforts.
- Doing one task consistently trumps doing every task poorly.
- Driver practice, after reliability and efficiency, is the third most important element of your robot.
- Always assume your mechanisms will break during competition! Always build with redundancy and easy repair in mind.
- The hardware of a successful team nearly always includes a simple and reliable drivetrain, an active passthrough intake, and a quick linear extension.
- Set a realistic schedule and follow it to the best of your ability.
- CAD does not replace actual prototyping.
- Overbuild, rather than underbuild.
- Linear extensions > arms.
- Attach mechanisms with redundancy.
- Servos =/= motors.
- A poor center of gravity can cause your robot to tip or turn off-center.
- Always aim for at least 90° of chain and belt wrap.
- Use the JVN for FTC calculator to determine feasibility and optimization.

- **FIRST Tech Challenge is all about reliability and efficiency**. This cannot be stressed enough! Your robot will win only if it is both reliable and efficient. Reliability should be your top priority when building - focus on building repeatable mechanisms that always act how you want them to. Leave plenty of time to test, diagnose, and improve your mechanisms before competition. Reliable means that the robot can complete a task well over 90% of the time. A reliable mechanism is only half of the equation, though. If it cannot score quickly, then the robot will not win matches. Efficiency is key to maximizing a robot's scoring potential. Short cycle times will lead to greater efficiency. When designing, always keep in mind that the shortest distance between A and B is a straight line. This may seem silly, but it is worth knowing that many top teams have optimized the pathway from collection to deposit to nearly a straight line, as it is simple and efficient to do so. The best way to do this is generally by using a passthrough intake. What this means is that the game elements should pass through the robot, allowing the robot to collect and deposit the elements without having to turn 180 degrees, minimizing cycle time. However, not every team can build a successful intake, and the simplicity of a well-executed claw trumps a poorly-designed intake.

- **Simple but effective should be the top priority for all teams. Execution is the key to success. A bad design implemented well NEARLY ALWAYS beats a good design with bad implementation.** This is probably the second-most important principle in FTC. Many teams at kickoff think of great designs that they have no way or no time to properly execute. There are designs that require machining and precision tools, and there are ones that don't - but that doesn't mean the simple design is necessarily inferior. In Relic Recovery, teams with a simple claw design could compete at a high level - for example, a clawbot could complete a cipher + relic + balance, slightly under or on par with many of the top teams who used intakes. This is because it takes a lot more time to test and perfect an intake and a deposit mechanism, as opposed to the vastly simpler claw, which achieves both collection and deposit in one. A good claw driver with a serviceable could consistently win matches against mediocre intake drivers with average intakes. This goes to show that a simple claw can be much more effective than an intake, which a lot of teams struggled with reliability and consistency issues.

- **Simpler = Less points of potential failure = Less likely to break = More reliable** (usually). The simpler the mechanism, the less likely it will be to break. This doesn't mean that well-designed complex mechanisms don't work - it just means there needs to be more thought put into them as there are many more things that could go wrong. If you're designing a complex mechanism, **make sure you have a justification for why it needs to be so complicated.** Simple mechanisms can be very effective. In Rover Ruckus, a robot won a state championship using a plastic cup as its primary scoring mechanism. In short: **Don't underestimate the power of simplicity, and don't reinvent the wheel!**

- **No team exists in a vacuum, take inspiration from other team's efforts.** A simple and easy way to jump ahead in efficiency is observing other teams. Their iteration also becomes your iteration. FTC has thousands of teams, and robotics challenges all have similar challenges. Chances are, another team has had the same idea that you have and already executed it. For example, both the RES-Q (2015-2016) and Rover Ruckus (2018-2019) seasons necessitated the pickup of a mix of both cubes and balls (in fact, the exact same cubes and balls). Therefore, teams playing Rover Ruckus were able to take inspiration from robots from RES-Q. Additionally, many teams will share their ideas/efforts online throughout the season, whether it be informally through platforms like Discord, or more formally through robot reveals, blog posts, or other videos. An important early season event is the Ri3D (Robot in 3 Days) videos, where mentors or experience teams attempt to build a competition robot in a very short timespan. These

Ri3D efforts serve as a great inspiration early season, and some even predict the most competitive designs.
Disclaimer: We're not trying to encourage copying designs, but there's a lot to be learned from more experienced teams.

- **Doing one task consistently trumps doing every task poorly.** Jack-of-all-trades robots exist, but it is very difficult in general to accomplish, much more so as an inexperienced team. This is not only a robotics principle, but can be applied to many areas of life as well. Obviously, top teams seek to complete every task, but inexperienced teams are better off choosing one objective and sticking to it. Generally, there are two tasks in driver-controlled mode - one during the initial 1:30, and one that can only be completed in the last :30 endgame.
    - Velocity Vortex - shoot balls; raise and place exercise ball (cap ball)
    - Relic Recovery - pick up and deposit glyphs; pick up and place the two relics
    - Rover Ruckus - pick up and deposit minerals; hang onto lander

  Our suggestion is to devote time to autonomous, but select one objective in tele-op and **focus on that for the whole season**. Don't overextend yourself and your resources trying to complete every objective, because what will happen is that you do everything very poorly, if at all. A robot in Rover Ruckus that does minerals well or can consistently hang will beat a robot which can do a couple of minerals and hang 50% of the time. Strive for consistency - do one thing 90% of the time and you will win more matches than robots that do everything 50% of the time.
  Disclaimer: This tip is not meant to discourage teams from aiming for the stars. Exceeding expectations is something that we all aspire to. However, we have seen far too many young teams fall into the trap of thinking that they have to complete everything to be a successful team, and fall short in both time and execution. On the other hand, an experienced team who specializes in one objective has a better chance of winning rounds and consequently, the competition. Setting realistic goals is an important priority for all teams.

- **Driver practice, after reliability and efficiency, is the third most important element of your robot.** Just like a sports team, a robotics team requires practice. Even if individual players have great talent, only practice and communication can guarantee success as a team. This principle also applies to the robot. Even if your robot is the best in the world, and inferior robot with a competent drive team more than likely will beat your robot. Drive teams need to communicate with one another and practice for the unexpected. A high school sports team practices about 15-20 hours weekly. To be successful, that is the same amount of time that you need to devote to your robotics team. By April's world championships, many top teams had 150+ hours of practice, not including building or coding meetings. This means that 1) the teams were dedicated to weekly (and sometimes even daily) drive practices and 2) their robot was reliable enough to survive 150+ hours of operation.

- **Always assume your mechanisms will break during competition. Always build with redundancy and easy repair in mind**. When building mechanisms, take into account that motors or servos may fail during operation. Will the mechanism continue to work if a motor blows? This is a key reason behind 4 motor drivetrains - often, it is possible to keep moving if a motor dies during a match. It is important to take into account points of failure and address them. Furthermore, build with easy repair in mind. For example, do not seclude your drivetrain motors in the deepest corners of your robot. You don't want to take apart your entire drive base to replace a motor in between matches. For new teams, modular design is not recommended, but teams should consider how easy it is to change motors or swap out a broken mechanism.

- **The hardware of a successful team nearly always includes a simple and reliable drivetrain, an active passthrough intake, and a quick linear extension.** Not many of the top teams are lacking in any of these mechanisms. Teams should focus on these three mechanisms in the off-season, and Game Manual 0 has guides specifically devoted to each of these essential components.

- **Set a realistic schedule and follow it to the best of your ability.** Ideally, your robot will be mostly functioning one full week before competition *at the latest* - planning for two full weeks before competition gives you plenty of time and creates flexibility that you'll thank yourself for later. Make sure you know how your progress compares to the schedule you set, and adjust your work accordingly to meet your deadlines. If you think that can't finish a key mechanism in time, simplify it until you can. Ensure that the team is optimized for efficient and that you're utilizing your team to its maximum ability - everyone should have a goal at each meeting, even if it's something really simple. Not only will this make your robot come together faster, it'll make everyone feel more engaged in the process. To reiterate: **do not try to build or code something the night before a competition. It won't work.**
    - How to set a realistic schedule - keep in mind that a simple design takes much less time to build, test, and perfect than a complicated design. Complex designs often take 3-5x the time it would take to produce a final product of a simple mechanism. Teams also forget that prototyping takes time. a mechanism rarely works the first or second time you build it, and it takes time in order to improve it. This is generally a great stumbling block for new teams - **they vastly underestimate the difficulty and time needed to construct a reliable and repeatable mechanism.**

- **Try not to cantilever shafts, especially on drivetrains.** Axles and shafts should **always** be supported on both sides, with the exception of motor shafts (because they are too short to be supported). It is far too common that shafts bend or gears come out of alignment due to the fact that there is only one support for an axle. This is fundamentally unsound and should be remediated as soon as possible. Gears, chain, and belt especially require good support and alignment to reduce efficiency loss and prolong its lifetime.

- **CAD does not replace actual prototyping.** While CAD can be a great tool to plan out the robot, no team can jump from CAD → final design. Prototyping is important to understand how your robot will actually work. It is a quick way to determine feasibility and to determine how in general the mechanism will work. It is possible to prototype with many different materials - cardboard, for example, makes an easy box or funnel to direct objects along a certain path. Wood is more sturdy and should be used for mechanisms that might be under some load. Finally, scrap extrusion or channel can also be a good option, if you have extra.

- **Overbuild, rather than underbuild.** Too often, teams seem to have underbuilt mechanisms. By that, we mean their mechanisms either aren't strong enough to withstand competition usage, or have not been attached properly.

- Linear extensions should be prioritized over arms. Many newer teams rush into a challenge by trying to build multi-axis arms. **Arms are *hard to build* and almost never worth it for new teams**. This is because arms firstly require high torque. None of the motors provided in any of the base kit will be able to withstand the load of multi-axis arms. In order to get the torque you need, you'll need to buy a planetary gearbox with a very high ratio, install that gearbox, and beef up every moving part of your robot. In the end, it's almost always easier to build a linear extension. In the latter part of the guide, we have provided documentation and many pictures of linear slides to help you get started with them.

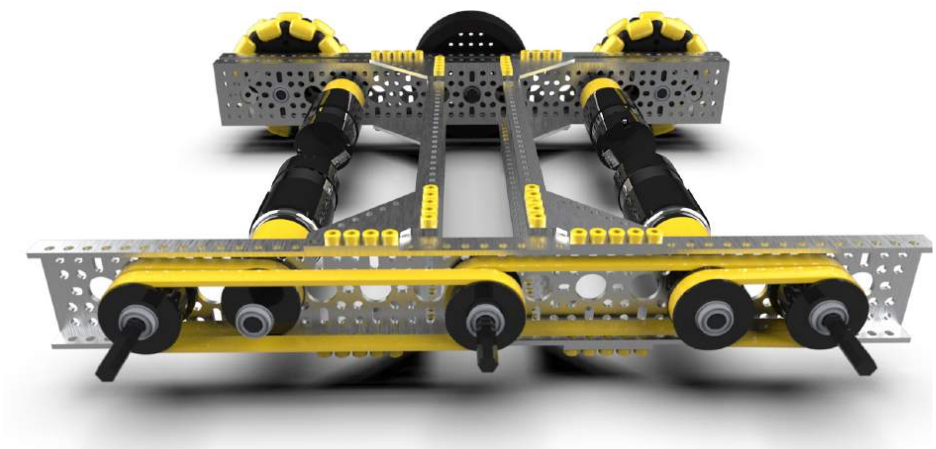
- **Redundancy in supporting and attaching objects** is a very important topic that many teams will struggle with in their first season. You want to **attach objects with 4+ bolts, especially if they are structurally important.** There isn't a reason to use less than 4 bolts. Many times, we have seen mechanisms bend, break, or wheels plain fall off because they weren't properly attached. It isn't rocket science, and it could cost you a match because you saved 30 seconds and two bolts. Using nyloc nuts (locknuts) is highly recommended. Use loctite in places like tapped holes.

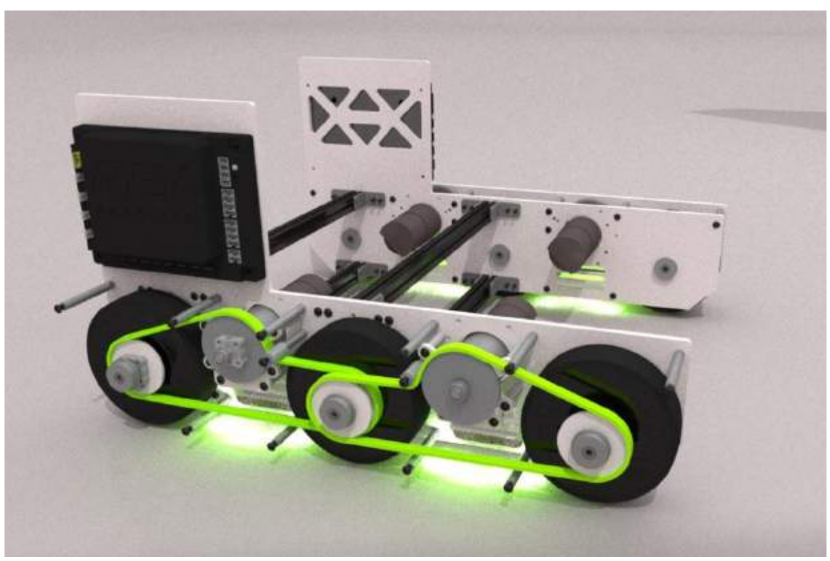
- **Servos and motors are not the same**. Do not treat them like the same! Motors are used for medium-high load applications, and are the primary method to power drivetrains and intakes. Servos are made for low-load precision applications, **not for intakes**. Far too many times, we have seen teams try to actuate hooks or load-bearing parts using servos, and unless you know how to properly support it, this ends up being a very bad idea.

- **Keep in mind your robot's center of gravity** when designing and building it. For mecanum wheels especially, center of gravity makes a big difference in terms of strafing ability. CoG is affected by two factors: the weight of the robot, and where the weight is distributed. For example, mounting an arm high above the ground causes the CoG to rise up and potentially allowing the robot to tip over when the arm is actuated. Keep in mind that mechanisms (especially arms) have momentum and don't stop instantly. CoG affects skid-steer drivetrains less, but can increase wheel scrub and decrease turning ability.

- **Chain and belt should, at the very least, have 90° of contact with the sprocket or pulley. The best practice is to have 180° or more of contact**, as it is very unlikely to fall off with proper tensioning. Chain skipping, especially on drivetrains or arms, is very possible without proper chain wrap or tensioning.

Best practices for wrap





Acceptable chain wrap, not recommended

- **Use the JVN for FTC calculator to determine feasibility.** Refer to the JVN section for more information on how to use the calculator itself. JVN is a useful tool so that teams can plan ahead and know what they're getting into, and see if their planned design is realistic or not. For example, if a team is limited to 40:1 gearboxes for an arm, JVN could assist in finding a suitable ratio - such as 80:1 - that would be achievable. JVN also can optimize spool size in linear mechanisms and maximum speed in drivetrains. Teams can plug different spool sizes in to see which would attain the optimized speed.

Kit and Hardware Guide

Before beginning your season, your team is presented with a crucial decision - one that will undoubtedly heavily influence the team's direction of hardware design in at least the first year. The FIRST storefront currently offers two kits for new teams: the TETRIX kit from Pitsco, and the REV Starter kit from REV Robotics. However, teams should carefully consider other options, such as Actobotics from Servocity and goBILDA, before selecting a kit. Every build system has advantages and disadvantages, which we have highlighted in our detailed build system guides below. While the guide may be slightly biased, we purposefully chose to do so, stemming from our own experience with the different kits, so the slant is there with reason. As with pretty much anything in FTC, there is no *one* right answer - but of course there are better answers than others. We hope this guide gives some solid advice on which kit might be the best for you.

Why use a kit?

After all, plenty of successful robots have not followed this framework. However, we still recommend new teams purchase a starter kit for one big reason. Established teams are all but guaranteed to have spare parts lying around to use to build their next bot. However, rookie teams, as is obvious, don't have the plethora of parts from previous seasons to use. Thus, new teams should purchase these parts themselves in order to have something to build from once the season starts, and starter kits offer these parts for less money than they would be if bought individually. As will be discussed, options that don't involve kits exist and are certainly very useful, but sticking to parts designed for FTC is recommended for new teams as a starting point.

TETRIX

The TETRIX build system, in a few words, is subpar, considered by many a difficult way to get started in FTC. After all, one has to start somewhere, right? The TETRIX Kit was the only available option from the FTC storefront for many years, giving rookie teams no other choice than investing into the TETRIX ecosystem. Because of this, you'll see plenty of TETRIX parts at competition, but that isn't indicative of quality.
The TETRIX build system revolves around 32 mm aluminum c-channels. The TETRIX flower is patterned along these channels to create plenty of mounting options for building structures. These channels have a tendency to flex and bend under load, often requiring reinforcement using multiple channels to maintain their shape. While TETRIX channel is measured in metric units, the system employs Imperial (SAE) bolts and Imperial chain. This mix of units means that things often don't quite line up how they're supposed to. This means that additional tensioners are required when using chain on TETRIX. Beyond this, TETRIX gears and sprockets offer very limited options for creating ratios, and the aluminum gears tend to grind away very quickly, especially if they are not supported correctly on both sides. As a young team, you may be considering TETRIX as a starter kit, but we encourage you to explore your options and consider your club limitations before selecting a kit. While Tetrix does pick up quite a lot of flak from the community, it isn't a bad choice in terms of physical qualities such as strength, but we feel that there are better and more well thought-out options to choose from which limit teams less. For example, TETRIX channel is similar to any other channel, and its 6mm D-shaft is quite sturdy. However, the mounting options and hole patterns leave more to be desired and restrict teams in terms of flexibility. **If you go through the FIRST storefront, we would recommend picking up the REV starter kit instead. Not only is it much cheaper, it is also a better build system. Additionally, TETRIX offers 25% off for all FTC teams, but this does not make this build system cheaper than any of the others.**

Kit Parts
https://www.pitsco.com/sharedimages/resources/FTC_COMPETITIONSET_18-19.pdf
(Note: this is for the 2018-2019 season)

- 32x32mm Aluminum C-Channel, various lengths
- Assorted plates and brackets, flat and 90 degree
- Flat plates
- Axles (6mm d-shaft), hubs, spacers
- Aluminum gears
- Standoffs
- **HS-485 HB servo (4x)**, assorted accessories
- **TETRIX Torquenado motor (4x)**
- Bars and angles
- omni and traction wheels
- Nuts, screws, fasteners
- Battery

- Tools (screwdrivers, etc.)

Advantages

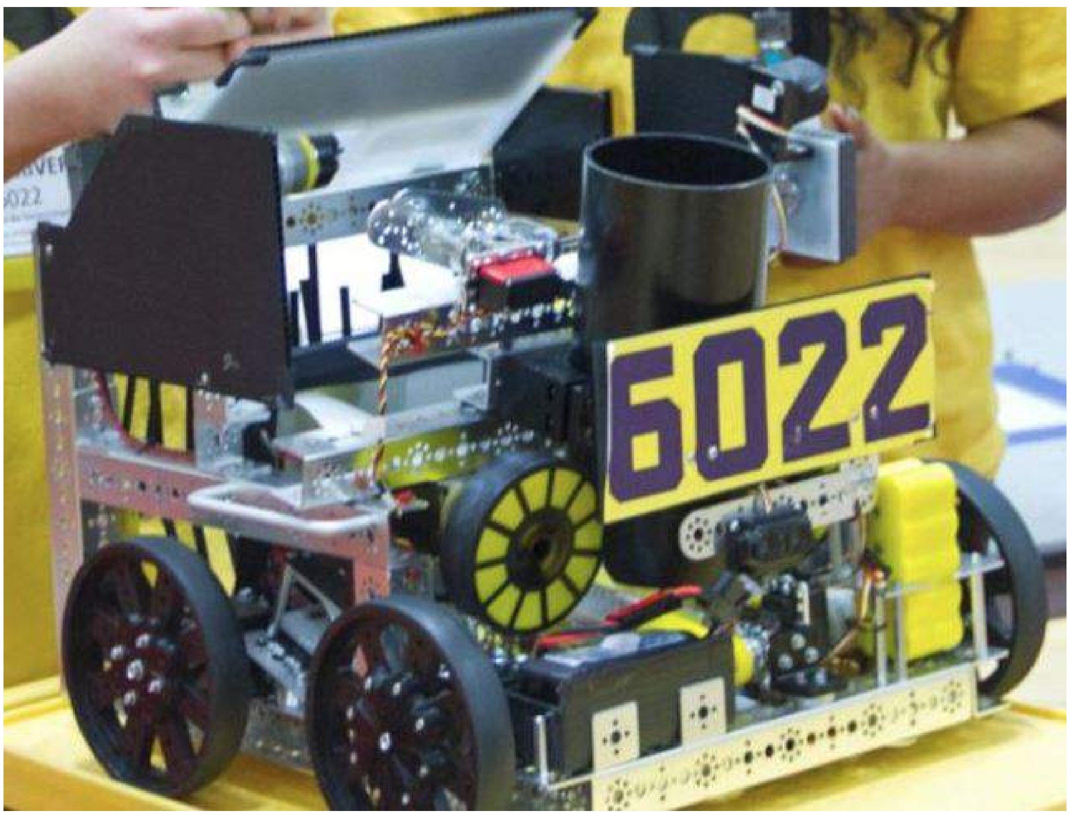
- The TETRIX kit, being the most basic of all kit options, is easy to learn and provides a variety of options in building. The kit itself comes with c-channel, which is aluminum shaped in a C. It has pre-cut holes so motors, gears, or drive shafts can be seamlessly integrated into the channel. For a beginner team with little to no experience, a TETRIX kit allows you to assemble a working drivetrain in a couple of hours (Note that most other kits allow you to do the same thing, but with more customization options).
- Furthermore, TETRIX is actually decent in terms of structural integrity, as long as the channels are loaded in the correct orientation.Typically, the orientation should be like an "n", with the top face upward. Connecting the channels with locknut instead of regular nuts aids in longevity. However, TETRIX is definitely the weakest of all the contemporary build systems and is very prone to bending, especially with multi-axis arms and heavy mechanisms.
- It is also simple to build basic mechanisms such as an arm using the gears and d-shaft. However, as will be explained later, there are many potential drawbacks to doing so.
- Due to its outdatedness, many teams will be willing to donate or sell their TETRIX parts cheaply.

Disadvantages

- The motors (Torquenado) are comparable with Neverest 60:1. They are usable, albeit relatively fragile compared to contemporary motors. Due to their gear ratio, they have high torque but rotate relatively slowly. However, due to their spur gearbox, they will not handle as much shock load until breaking, especially compared to Andymark 20 Orbitals, REV HD Planetary, or the goBILDA Yellowjacket Planetaries.
- The aluminum gears, in our experience, have seemed to grind against each other often, even with correct spacing, and do not last very long, especially under high torque situations.
- TETRIX hubs are 6mm round and are based on set screws, so the set screws are torque transferring. These set screws are notorious for coming loose under load, so special care must be taken to continuously tighten these screws. Additionally, it is highly recommended that teams use some sort of threadlocker on set screws, whether it be Loctite Blue (removable) or Loctite Red (nonremovable). A possible workaround is purchasing 6mm D clamping hubs from goBILDA to use on TETRIX shafts, however, this also necessitates the use of a pattern adapter from the goBILDA pattern to TETRIX.
- Due to using a metric based pattern but using imperial based holes and chain standards, no chain will have perfect tension, so an external tensioner will be required to implement chain systems.
- TETRIX is also the most expensive kit on average while providing the most limited build options. Finally, the tight spacing and mix of units limits teams if they would want to integrate custom parts with TETRIX.

- Set screws on TETRIX hubs can destroy motor shafts, and the bore of the hub itself can enlarge, leading to a wobbly or misaligned hub.

Tips for use:
In any build system, it is important to properly support the structure of your robot. This includes supporting axles at two or more points of contact, or having multiple points of support for a piece of channel. Due to TETRIX's relatively fragile nature, this advice goes double with TETRIX. Instead of just supporting a shaft at two points, it is recommended to support the shaft at 4 or more (if you are using a 4.7mm Standard). When mounting motors, it is recommended that two clamping motor mounts are used to mount one motor. Using standoffs inside channel to prevent the channel from bending inward or outward is also highly recommended.



TETRIX based robots have succeeded in competition, albeit it has become rarer in recent years,
6022 To Be Determined
Worlds Semifinalist (St. Louis), Velocity Vortex

TETRIX has a tendency to bend when not properly supported. Here, a piece of TETRIX channel bends extremely under load during a collision.

Actobotics

The Actobotics kit has long been regarded as one of the premier kits for FTC teams for many years. Its robust 1.5” c-channel and ball-bearing based motion system allows teams to quickly iterate and build reliable mechanisms entirely from kit parts. One advantage of Actobotics is that their channel hole pattern has many more mounting holes than a Tetrix channel, so it is less restrictive in terms of mount placement. Actobotics uses imperial units across the board, allowing for cleaner spacing and better fitting. A variety of mounts and brackets make strong, compact joints between channels. Actobotics also sells mini-channel in addition to the standard size channel. In addition to the channel, the main structural component in Actobotics kits, the X-rail extrusion system offers immense adjustability in mechanisms. To complement the structural offerings, Actobotics offers a series of high-quality clamping hubs that rarely come loose and a very strong ¼” steel D-shaft that doesn’t bend easily. Finally, Actobotics has an off-the-shelf Linear Motion Kit, using their X-rail extrusion, to easily build extending mechanisms like those seen on many world-class robots. The patented ServoBlock, which prolongs the life of a servo, is highly recommended for all teams, regardless of kit; however, ServoBlocks seamlessly interface with the Actobotics ecosystem. Actobotics furthermore offers options such as bevel gears and Linear Actuator kits (using lead screws) for specific use cases such as a hang. Actobotics can interface with other kits such as REV through a variety of Pattern Adapters. **Servocity (the Actobotics vendor) offers a 25% off discount for all FIRST teams, making pricing very competitive. Actobotics is a great choice and value for new teams to seriously consider, offering a solid base kit with many options to expand upon.**

Kit Parts
https://www.servocity.com/ftc-competition-kit

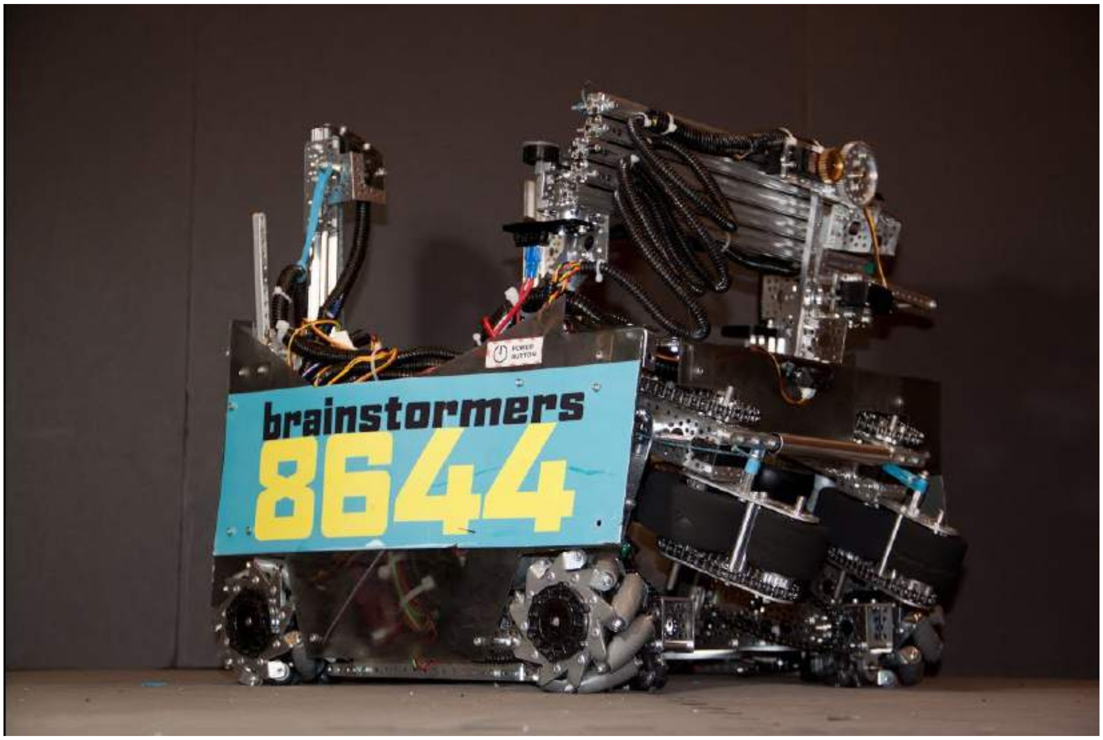
- Aluminum c-channel and mini channel, various lengths
- X-rail (a special type of extruded rail similar to 8020)
- Assorted plates and brackets, flat and 90 degree
- Flat plates
- Clamping hubs, spacers
- Gears and sprockets
- ¼” d-shaft
- Standoffs
- **HS-488 HB servo (4x)**, assorted accessories
- **5201 Series, 53:1 Ratio, 105 RPM Spur Gear Motor**
- **5201 Series, 26.9:1 Ratio, 210 RPM Spur Gear Motor**
- Bars and angles
- Omni and traction wheels
- Nuts, screws, fasteners
- Battery
- Tools (screwdrivers, etc.)

Advantages

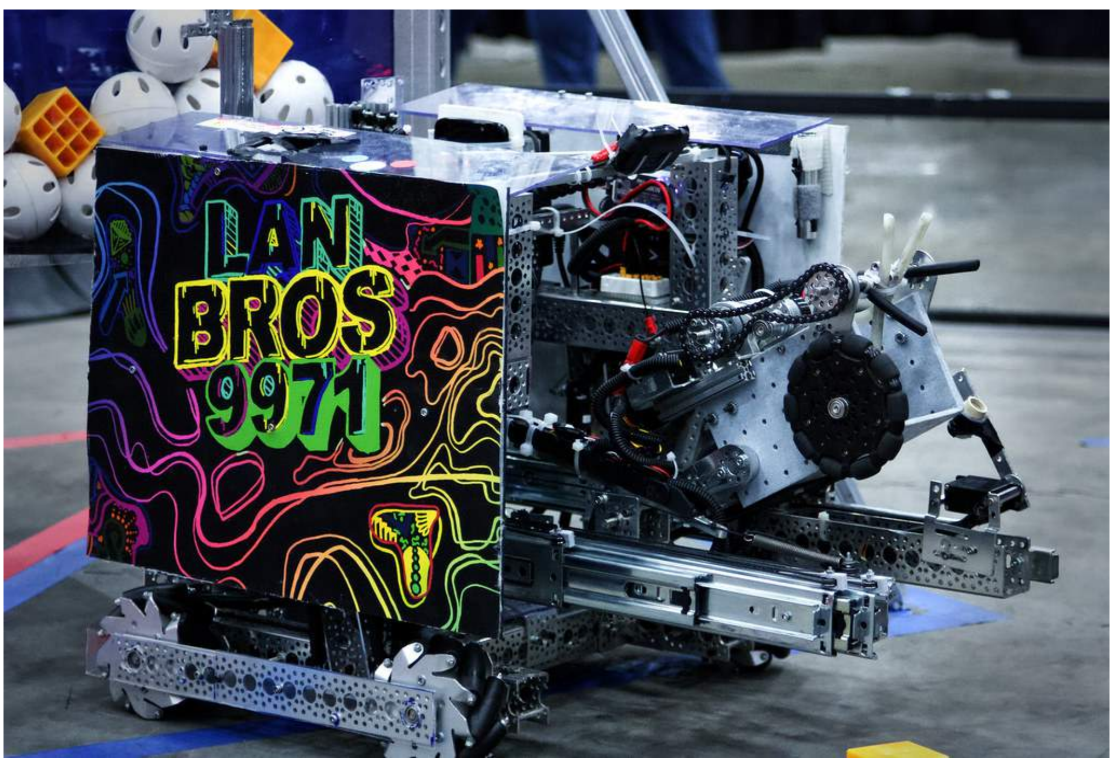
- Actobotics provides a great value build system that is generally easier to assemble than REV's kit, which requires cutting extrusion to length.
- Actobotics has the patented ServoBlock, which helps drastically increase servo life by protecting it from shock loads. More information may be found in the glossary.
- Actobotics is easily compatible with other build systems such as REV's kit using adapters which can be found on the Servocity website.
- Actobotics is more sturdy than TETRIX in terms of drivetrain flex and has more support options to prevent structural bending.
- One great advantage to Actobotics is the clamping hubs that are offered. As discussed in the disadvantages to TETRIX section, set screws in particular are vulnerable to loosening, especially without application of Loctite.
- Actobotics' motion system is **very** robust and relies on ball bearings, which have a lower coefficient of friction than bushings. Actobotics is compatible with the 5mm hex used by REV with their adaptable hubs.
- Actobotics also allows for face mounting of motors as opposed to clamp mounting. This is explained in the glossary section.

Disadvantages

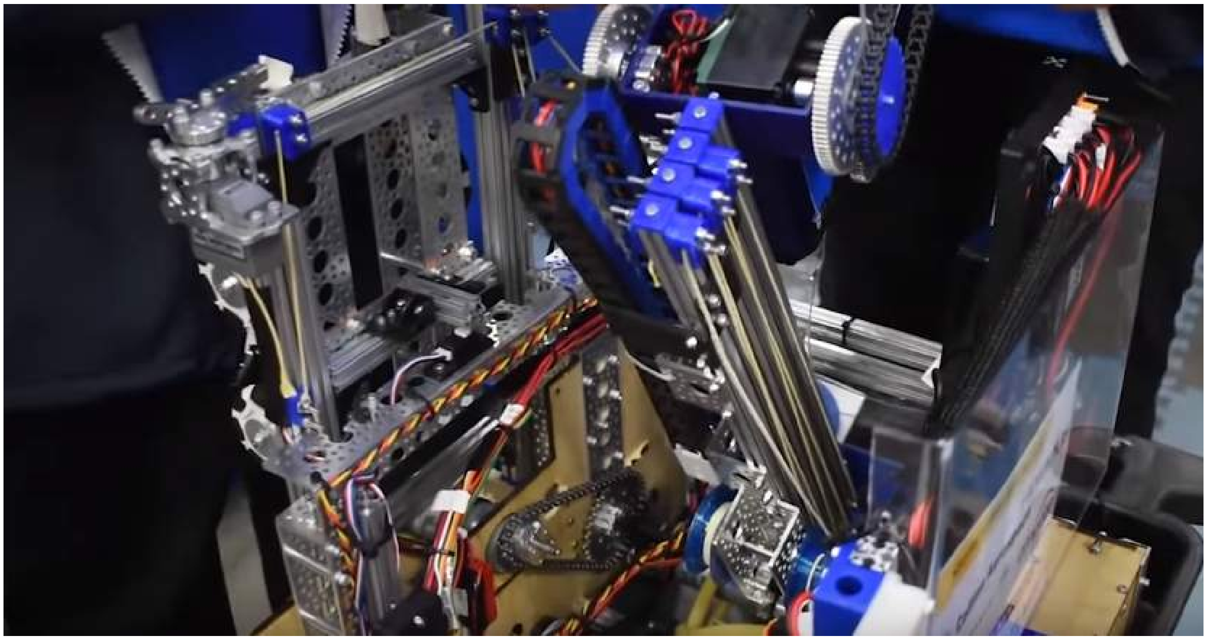
- Actobotics is not very cheap, so its cost may be prohibitive for some teams with a low budget. Note that with the 25% off FTC team discount, Actobotics can be cheaper than TETRIX. The only cheaper build system is REV.
- The belts in the Actobotics system is not great (it is recommended that belts are purchased from vendors other than ServoCity), which may necessitate 3D printed belt pulleys and other parts, which is more difficult for newer teams. There also isn't a large amount of space inside the channel for a belt or chain run, vastly limiting the size of the pulleys.
- Channel takes up more space than extrusion, so mechanisms are a bit larger with the Actobotics kit. To remediate this issue, Servocity sells mini-channel which is a similar size as extrusion.



An example of a successful Actobotics based robot, 8644 Brainstormers
2018 Winning Alliance Captain (Detroit), Relic Recovery



Another successful Actobotics based robot, 9971 LanBros
2019 Winning Alliance Captain (Detroit), Rover Ruckus



9794 Wizards.exe used both Actobotics and REV extrusion to build their Rover Ruckus robot

goBILDA

goBILDA by Robotzone, the makers of Actobotics, is the newest build system on the market, being fully released in the 2018-2019 Rover Ruckus season. Even with its relative novelty, it has quickly rose through the ranks and has become one of the premier build systems in FTC. Designed for the international market, it uses a metric hole pattern, M4 screws, 8mm pitch chain, HTD 3 belt, and metric shafts (6mm D). Much like Actobotics, it is channel based with an additional extrusion system. Unlike Actobotics, however, low side channel is the foundational building material to supplement full size U-channel. Even though this may initially seem like a stepback, the use of low sized u channel opens up many new possibilities such as a kit only parallel plate drivetrain. Additionally, goBILDA is still being developed at an incredible rate (4-5 new products a month), their customer service is top notch, and they are very active in the FTC community, taking suggestions and rolling out new products. **Just like Actobotics, goBILDA offers a 25% discount for FIRST teams. For these reasons, goBILDA may soon become a popular robot kit in FTC.**

Kit of Parts

https://www.gobilda.com/master-ftc-kit-2018-2019-season/

- 5202 Series, 26.9:1 Ratio, 223 RPM Planetary Gear Motor w/Encoder **x2**
- 5202 Series, 13.7:1 Ratio, 435 RPM Planetary Gear Motor w/Encoder **x2**
- 2000 Series Dual Mode Servo (25-1) **x4**
- Servo mount plates, attachments, accessories
- Matrix 12V 3000mAh battery with XT30 to Tamiya Adaptor
- U-channel, assorted lengths
- goRAIL extrusion, assorted lengths
- Pattern and grid plates
- Square beams, u-beams, l-beams
- Assorted brackets, standoffs, spacers
- Gears, sprockets, chain (8mm plastic chain)
- Bearings, hubs, collars
- Shafting (6mm D-shaft)

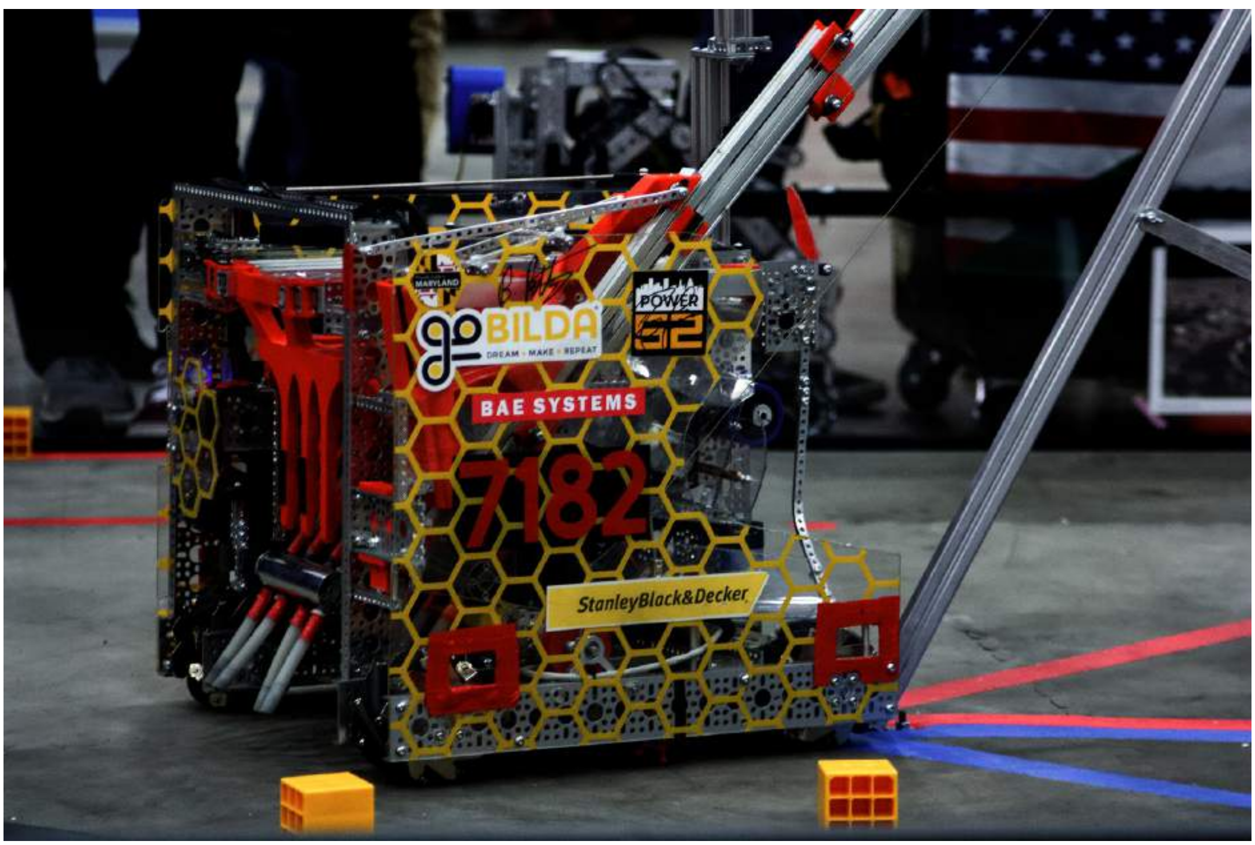
Advantages

- goBILDA's low sized U-channel opens up many new possibilities due to its flexibility and compactness. For example, one can now make a parallel plate drivetrain without custom machining, or make custom width U channel.
- goBILDA also has ServoBlocks (identical to Actobotics other than hole pattern) which drastically increase servo life.
- goBILDA has native large bore hex shaft support (12mm Rex is comparable to ⅜ Thunderhex) which is one of the main advantages of custom fabrication brought to a kit based system.
- goBILDA also is able to interface with TETRIX channel because they share some holes.

- goBILDA has all of the advantages of Actobotics, it has a well thought out ball bearing based motion system with smart motion transfer (easy to do chain or belt in channel. 8mm chain and HTD 3 belt can do perfect C2C (center to center) on the goBILDA Pattern. It also has plenty of shaft, bearing and pillow block options. It also has Servoblocks.
- goBILDA can be incredibly strong and robust.
- goBILDA motors can face mount natively into channel, eliminating the need for motor mounts and providing a robust, reliable way to mount motors.
- goBILDA has some special parts are unavailable in other systems, like square beam shafts.

Disadvantages

- goBILDA is not the cheapest build system; it is relatively equivalent in pricing to Actobotics, but REV is cheaper.
- To achieve perfect center-to-center (C2C), goBILDA chose to adopt the HTD 3 belt standard. While this is a good choice, note that under extremely high torque loads (like an arm) the belt may skip. An easy way to remedy this would be to 3D print and use the HTD 5 belt standard, which will have perfect C2C on goBILDA channel every 5 bearing holes.
- Because of the Metric Pattern Spacing goBILDA utilizes 8mm pitch chain, as opposed to the FTC standard #25 Imperial chain. This means that other kits' chain and sprockets won’t work with goBILDA chain and sprockets. This of course has the advantage of having perfect C2C on channel.



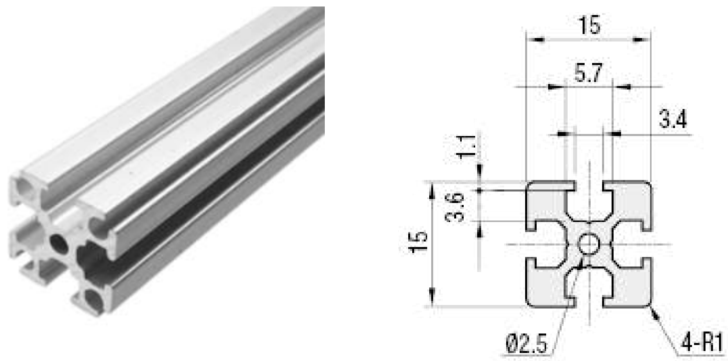
An example of a successful goBILDA based robot, 7182 Mechanical Paradox Cubed
Finalist Alliance 1st Pick (Detroit), Rover Ruckus

REV Robotics Starter Kit

The REV kit is very different from the other kits mentioned above, primarily because it is an extrusion-based system. The REV kit uses 15mm aluminum extrusion instead of channels commonly found in other kits. Thus, there is no fixed pitch, which means having fixed holes and mounting points. Extrusion systems are all based upon adjustability and flexibility - for example, tensioning chain is simple when sliding the mount or bracket increases tension. Extrusion systems are one step towards a custom bot, as it needs to be cut to length using a saw or bandsaw. The REV system, unlike the other kits, uses metric dimensions (15 mm extrusion, M3 hardware). REV also uses 5 mm hex for their shafts, which is something that other vendors have added compatibility for in recent months, a great benefit and time-saver when trying to adapt to different systems. REV also has adapters built into many parts such as the aluminum brackets. Another advantage to REV is the delrin (a kind of low-friction plastic) and aluminum components. In the base starter kit, many parts will be delrin in order to reduce the kit cost. While these are suitable for a new team, REV does sell many components in a metal version, which is recommended if your budget allows for it. **In general, REV has a steeper learning curve than other build systems** (owing to the fact that in extrusion the builder must figure out the optimal mounting position, etc.), but this is not necessarily a large disadvantage. **The REV kit is a great choice for teams willing to invest the time into an extrusion building system. It has the ability to upgrade parts for those wanting a further step and investment into an even more complete building system.**

Another advantage to the REV kit is the compatibility of 15 mm Misumi extrusion. REV extrusion isn’t as structurally sturdy as Misumi, because the REV extrusion can twist more easily under load. This is because the REV extrusion has corner holes for tapped M3 bolts. Misumi offers greater strength at a lower bulk cost. Additionally, Misumi will cut to the half millimetre for free, making it a great option for teams needing an exact cut. The drawback to Misumi is that it is quite a bit heavier than the REV extrusion, given that is is more solid. **It is encouraged that teams use Misumi for drivetrain and structural support, and REV for mechanisms that will be under low to medium load.** REV also sells punch tubing, which is 15 mm aluminum square tubing that is compatible with ⅛” rivets. This will prevent bolts from loosening or sliding around, as punch tubing requires teams to drill through the tubing in order to fasten parts to it.

As a tip, while it is certainly possible to construct drivetrains out of 15 mm extrusion (and there have been teams who have done so successfully), keep in mind that 15 mm extrusions' torsional rigidity is inferior to channel. A REV drivetrain needs more support and crossbeams than a typical channel based drivetrain. Additionally, the drivetrain may flex in the middle of the robot is heavy, thus requiring more supports.



REV extrusion (left); Misumi extrusion (right)

Kit of Parts
http://www.revrobotics.com/rev-45-1270/

- 2 40:1 HD Hex Motors
- 2 Core Hex Motors
- 4 Smart Robot Servos with other attachments
- Smart Robot Servo Programmer
- 15 mm aluminum extrusion, various lengths
- Omni and traction wheels
- 32 gears in 7 different sizes
- 10 sprockets in 3 different sizes
- Delrin brackets
- Delrin bearings and pillow blocks
- Motor and servo power/data cables
- Slim Robot Battery and charger
- 5.5 mm nut driver and 5.5 mm hex wrench

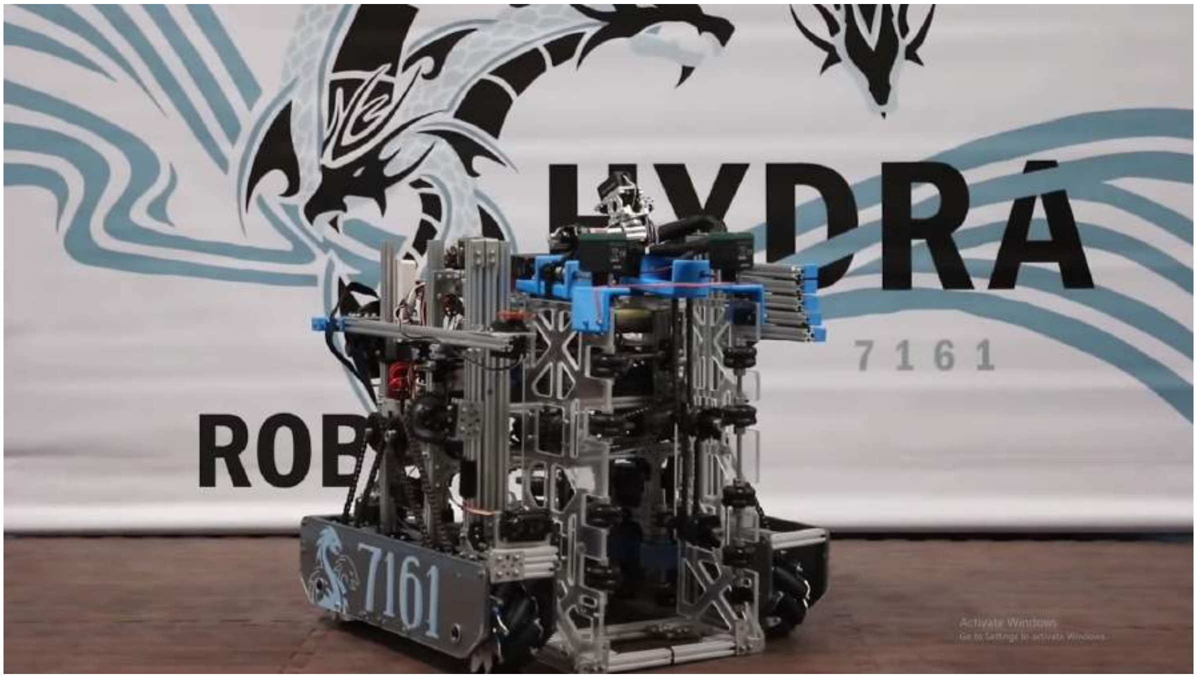
Advantages

- Extrusion systems don't need to worry about tensioning as mounts are adjustable to your needs.
- Extrusion allows teams to save space as opposed to channel, and is lighter than aluminum channel.
- Extrusion allows infinite positioning options instead of being locked in to a specific distance - useful for fine-tuning a mechanism.
- Delrin products are inexpensive yet durable for most use cases.
- REV has the option to upgrade to aluminum parts if need be - something that no other build system offers. (not for all parts)
- 5 mm hex is a robust shaft and motion system and is easily adaptable to Ultrahex ½" hex shaft. Other companies have adapting options with 5 mm hex.
- Punch tubing is a great final iteration option if you are sure about placement.

Disadvantages

- Steeper learning curve, more time consuming to prepare extrusion.

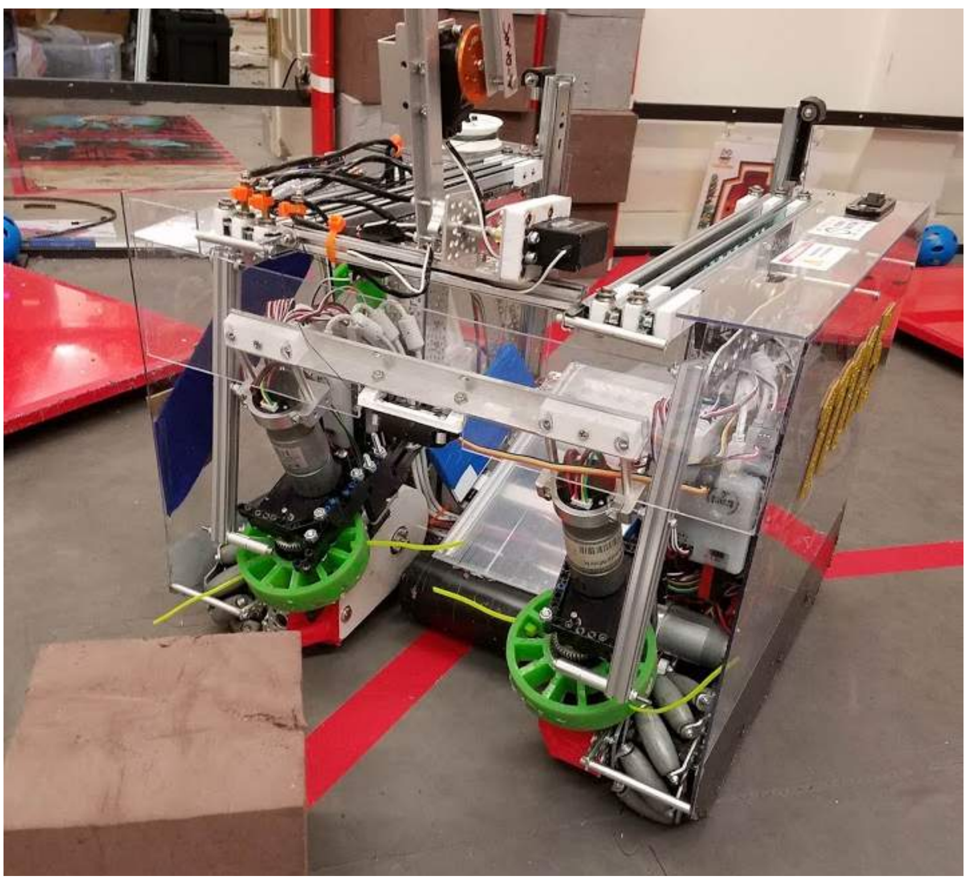
- Requires tools such as a saw and bandsaw
- Requires forethought and planning of extrusion length and placement
- Parts loosen over time (to remediate: use punch tubing)
- Metric makes adapting to other systems slightly harder
- M3 bolts, especially those sold directly by REV robotics, are prone to bending under higher load (such as when used as an axle for a pulley)
- 5 mm hex shaft is also prone to bending, especially if the shaft is long.



7161 ViperBots Hydra, Finalist Alliance 1st Pick (Houston), Relic Recovery

9889 Cruise Control, Rover Ruckus

6299 ViperBots QuadX, Velocity Vortex



11115 Gluten Free used both REV and Actobotics for the 2017-2018 season.

Finalist Captain (Detroit), Relic Recovery

Custom

Many teams elect to ignore kits entirely and instead make their own mechanisms completely from scratch. While this **allows for much more design freedom**, keep in mind it **has many caveats and drawbacks**, especially for a new team.

In discussing custom options for teams, there are two options for teams - one encouraged and one discouraged.

- Custom using 3D printing - new teams should learn to 3D print parts for their robot, though there will be much trial and error. Make sure to print spares of *everything 3d printed* that is on your robot.
- Custom using machining - this is highly discouraged for new teams **unless you have a mentor with experience in this field.**

Advantages of 3D Printing

- 3D printing allows for customizable sizing; for example, teams can print a spool of the exact diameter needed for optimal speed.
- 3D printing allows teams to adapt between kits and individual parts easily, as not all kits have adaptable mounts or brackets. A good example of this are the Nexus mecanum adapters that teams 3D print.
- 3D printing allows teams to fabricate parts that would otherwise be impossible with materials such as aluminum due to equipment restrictions.

Disadvantages of 3D Printing

- 3D printed components are generally weaker than other materials such as aluminum.
- The size of 3D printed parts are limited by the size of your print bed.

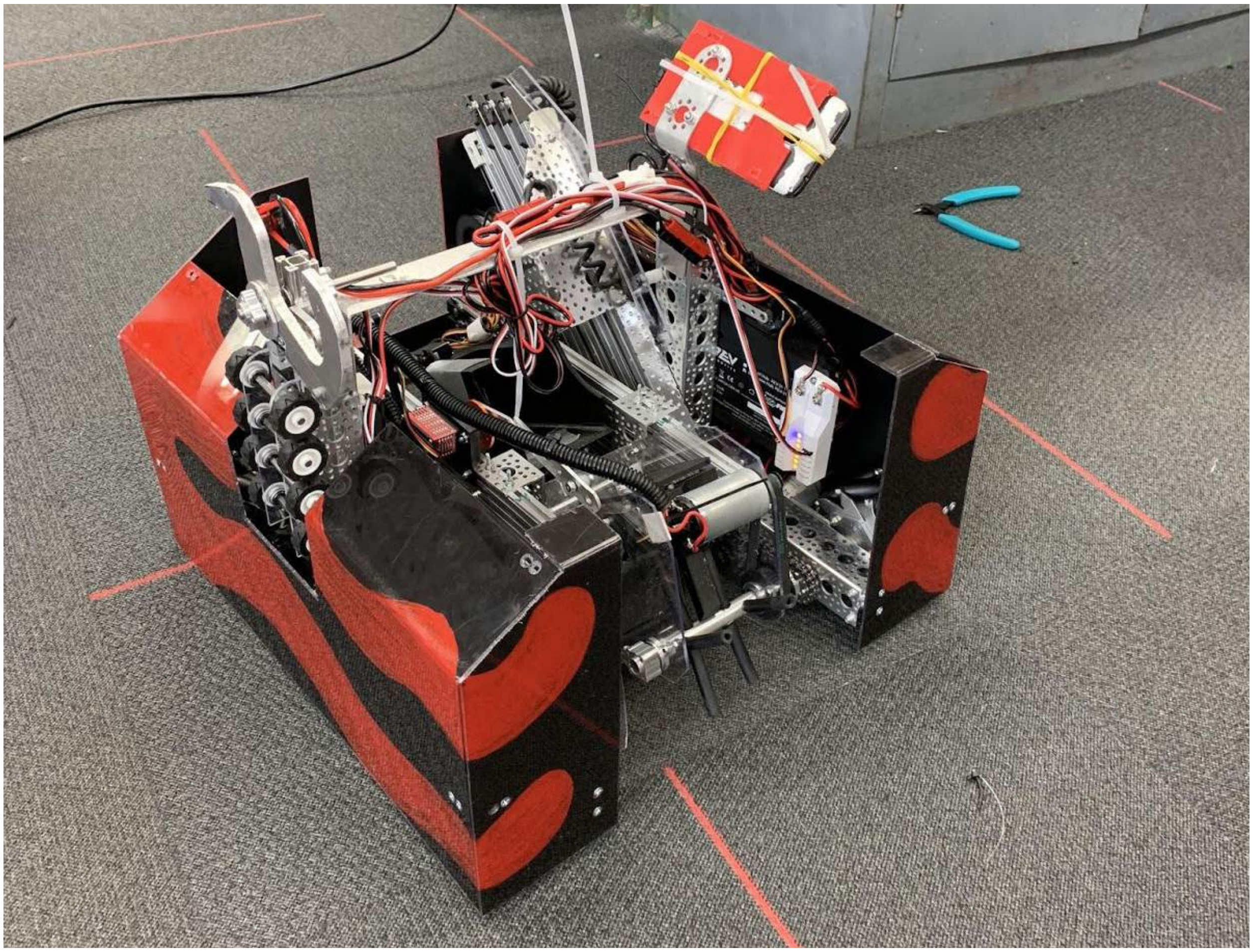
Advantages of Machining

- Machining allows teams to create practically any part without limitations.

Disadvantages of Machining

- Perhaps the biggest barrier when making a fully custom robot is that it requires expensive manufacturing equipment. While 3d printers have become more and more common in FTC and allow teams to create or customize small parts, many teams do not have access to equipment such as lathes, routers, waterjet cutters, or CNCs.
- Fully custom robots need to be fully designed in CAD.
- Another barrier is that prototyping with fully custom systems is much slower than using kit parts. This is because the full CAD process takes time, and fabrication/getting parts fabricated by a shop or sponsor is another lengthy process. Notwithstanding those two factors, if something goes drastically wrong then it will require a repeat process of CAD and fabrication.

However, we do not discourage using custom parts. Many successful teams are actually a hybrid of both kit and custom parts.



8680 Kraken Pinion, Division Semifinalist (Detroit Worlds), Rover Ruckus
Kit-based drivetrain (Actobotics) and horizontal linear slides (REV) with a custom intake, housing, and mineral container. Kit linear actuator (Actobotics) with custom hook.

FTC 8393 Giant Diasymphalic BrainSTEMs Robotics, Relic Recovery, fully custom

Tools List

Here are a selection of necessary and helpful tools for your team. You'll obviously need to purchase the basics such as the right screwdriver, hex keys, and drill bits. For a bit more precision and the ability to dabble in custom, opt for a drill press and bandsaw. A router might be useful for teams interested in working with wood, a great material for prototyping (and also for final iterations!)

Necessary

- **Safety glasses**. **Wear them when you're using power tools. Seriously.**
- Screwdrivers
    - Philips, assorted sizes
    - 3/32", ball head for set screws is discouraged (TETRIX)
    - 5.5 mm hex nut driver (REV)
- Drill and drill bits
- Hex keys (metric and US standard)
    - 1.5mm hex key (REV)
- Pliers, needle-nose and locking
- Metal file (sandpaper isn't recommended)
- Clamps (2+) or vise
- Hammer

- Centerpunch
- Wrench
- Hacksaw
- Wire stripper/wire cutter
- Zip ties/Velcro ties
- Electrical Tape

Helpful

- Bandsaw
- Chop saw
- Drill press
- Dremel
- Grip tape
- Caliper
- Soldering iron
- Heat gun
- Router or table saw
- Jigsaw

Design Process

Engineering Design Process

Define the problem

Specify requirements

Brainstorm/CAD design

Prototype/Experiment

Test

Iterate

Final implementation

In this section, the engineering process will be discussed. In general, the design process contains these steps.

- Define the problem - what are you trying to solve? what is the timeline?
- Specify requirements - what are necessities for your proposed solution?
- Brainstorm/CAD design - draw or sketch a prototype on paper or in CAD
- Prototype/Experiment - put a first design together using physical materials
- Test - ensure you thoroughly test every part of your prototype to detect flaws
- Iterate - identify and remediate problems through successive prototypes
- Final implementation - polish and solidify a final design that won't change

As an example, let's look at an imaginary team building a drivetrain for Rover Ruckus (RR2), which was 2018-2019's game. In RR2, there was a crater, which was about 3" tall, and could be traversed. In that year's game, there were two main options: traverse said crater, or not traverse it, and instead reach over with an arm. The team must first specify requirements for the drivetrain. One of the most important aspects of a drivetrain is maneuverability. Another requirement could be speed, traction, reliability, etc. From the team's specified requirements, they would now look at drivetrains that fit their requirements. If the team wanted to traverse the crater, then a 4WD or 6WD would be an optimal design. If they did not need to traverse the crater, many options are still on the table such as the holonomic drivetrains. The next step would be to brainstorm actual designs. It would be prudent to have more than one student to design one drivetrain prototype, so that the team could test more than one drivetrain. From there, a team could begin the prototyping and testing process. This could involve tests of the time taken to cross the crater, time from crater to lander, top speed, maneuverability, etc. Simple tests may be conducted instead of complicated ones. For example, if the robot is projected to weigh 30 pounds, a dumbbell placed on top of the drivetrain simulates the extra weight well to see if the drivetrain can get over the crater easily and reliably, so the robot will not get stuck on the crater edge. Furthermore, iteration is likely needed. If, for example, there was too little clearance, and the drivetrain got stuck on the crater part of the time, then it would be necessary to raise the drivebase in order to remediate this problem. However, don't change too many things at once - **you want to change ONLY ONE variable at a time**, or else you won't know what may cause a further problem should it arise. Always try to change one variable at a time. **It often takes multiple iterations to get things right, so don't be discouraged if your second or even third attempt doesn't perform as well as you think it will**. Many teams have 10+ iterations of intake designs in order to refine and hone down their design to be optimally efficient. While that many tries isn't recommended for new teams, don't be afraid to modify *one factor at a time* to isolate and solve problems. Your final implementation could be just an upgraded prototype, or if you had used subpar/scrap materials for a prototype, you could change them out for durable ones.

Materials Guide

In FTC, teams aren't limited in terms of what raw materials to use. However, there are definitely some important materials which you should be aware of. Here are the materials to use, listed in order of importance.

- Aluminum: a high strength, medium-high density material. Suitable for use in nearly every application; recommended in load-bearing applications.
  - Aluminum channel is used in many build system kits and is very popular among teams of various skill. Channel is strong yet relatively lightweight, and offers many mounting options for teams.
  - 15 mm extrusion is compatible with M3 hardware, allowing teams to slide in bolts to their desired location. REV extrusion is not as great structurally, but is lighter than Misumi extrusion, which is more resistant to flexing and/or twisting under load. Both REV and Misumi extrusion are sold in bulk quantities, and Misumi has the option to cut to the half millimetre. Keep in mind that a lot of extrusion can add up quickly in terms of weight.
  - 8020 extrusion (1") is regularly used as the primary building system in FRC. It is definitely overkill for FTC purposes and should not be considered as the primary building component by teams. It may be useful in certain applications (for example, linear motion using v-groove bearings).
  - Aluminum flat and aluminum angle is widely available at hardware stores. In certain applications, such as adapting from different build systems, it is possible to drill custom adapter plates to mount mechanisms to the drivetrain. Aluminum angle is also a very sturdy structural support piece that takes up relatively little space, and can adapt to any build system. We suggest using ⅛” alumimum with drivetrain or mounting applications, and 1/16” aluminum for low-load situations.
  - Aluminum drawer slides, often with ball bearing sliders, are recommended over steel drawer slides primarily for weight purposes. Refer to the Linear Motion section for more information.
  - Sheet aluminum is generally used for drivetrain plates on custom drivetrains. The recommended thickness is ⅛" or 3/16". Because it is a plate, sheet aluminum will bend if not supported correctly with standoffs or channel. Only load the sheet in the plane that it is in (if the sheet is vertical, then only put vertical load on it; do not load it horizontally). However, there may be some applications that would benefit from a slight bit of flex for adjustability - in those cases, use your judgment and test it out for yourself!

- Wood: a medium to high strength material suitable for use in a wide variety of applications. Keep in mind that wood cannot be exposed to water or excessive humidity, as lumber for use in FTC isn't treated and may warp or expand. Do not try to use laser cut wood for a drivetrain. This is especially important if you live in a humid region such as Florida - wood expanding can completely ruin a drivetrain.

  - Baltic Birch is the highest grade plywood, used in commercial applications and furniture. It contains 8+ layers (usually), is extremely dense, and is recommended for high-load applications or structure. It is quite an expensive material, so prototype and plan carefully before cutting.
  - Plywood is not recommended for final iterations, but can be a cheap prototyping material. It doesn't bear load especially well and can flex quite a bit.
  - MDF is generally discouraged as there are better options.

- Steel
  - Steel is unnecessarily heavy for FTC structure. Aluminum provides plenty of strength at a fraction of the weight, and doesn't require welding.
  - The proper uses of steel in FTC are in shafts (most are made out of precision-ground stainless steel) and gearboxes.
  - Steel drawer slides *can* be used, but aluminum slides are highly recommended.

- Lexan
  - Polycarbonate, also known as Lexan, is a material that is great for applications such as drivetrain plates or intake collector boxes. It can bear load and is very impact-resistant. Many teams use it in the intake and deposit mechanisms as it is clear, allowing the drive team to see into the intake itself, an advantage over wood. Thick lexan can be used for drivetrain plates, though this is not recommended for inexperienced teams. Thin lexan can be bent with a metal brake or sheet bender. It is recommended for teams to use bends instead of connecting with bolts - bending tends to be much stronger than bolting as it means the part stays in one continuous piece. If a sheet bender is out of the question, it is possible to use camping burners to heat up the lexan in order to bend it. This is not recommended as it can cause injury as well as bubbling (which results from overheating the lexan). Alternatively, "cold bending", bending along a straight edge without applying heat can work for thinner sheets. Lexan is one of the most expensive materials per square foot, so make sure you have carefully planned out what you are cutting before doing so.

- Acrylic
  - **Acrylic is not a load-bearing material. It will crack and possibly shatter under impact.** Do not use it on drivetrains by any means! It should only be used in mechanisms where there is no chance of shock impact. It can be used for aesthetic purposes or as shielding (to protect game elements from falling into the robot or from other robots tangling with wires, etc.) Treat acrylic as fancy clear cardboard - it cannot be loaded and should only be used in very specific circumstances.

- Delrin
    - Delrin is a low-friction plastic used in many REV robotics parts. It is a durable and strong plastic that is not easily cracked. Delrin can be used in drivetrain plates, but delrin sheet is also quite expensive.

- Carbon fiber
    - Carbon fiber is one of the strongest materials that any FTC team could use. For most teams it is totally overkill, but it can be used in some specific applications. Carbon fiber rods are used in custom linear slide extensions or multi-axis arms. It is probably the most expensive material to purchase.
    - **Machining carbon fiber, like any fibrous substance, is a significant SAFETY HAZARD. Carbon fiber dust especially can cause cancer and is incredibly dangerous. DO NOT MACHINE/CUT Carbon Fiber unless you know what you are doing. When you do, make sure to either use machinery that is designed to cut carbon fiber, or cut in a well ventilated area with sufficient respiratory protection and running water over the carbon fiber.**

- Cardboard
    - **Please do not use cardboard as a load-bearing material.** We have seen too many teams use cardboard in ways that it shouldn't be used. Treat cardboard as a sheet of paper: it has no structural rigidity and only should be used as guides to channel pieces from A to B.

Drivetrains

This section will cover the heart of any robot, the drivetrain. The purpose of the drivetrain is to facilitate the movement of the robot, and thus is a mechanism crucial to the overall function of the robot. If the drivetrain doesn't work, the rest of the robot won't work either. There are many possible types of drivetrains in FTC, which we have covered in the guide. Drivetrains are split into two main types: **tank** (skid-steer) and **holonomic**.

Tank

A tank drivetrain primarily utilizes traction wheels and cannot strafe (move sideways). To change directions, a tank drivetrain relies on either turning the wheels on the left and right side in the opposite direction (thus spinning the robot) or running one side faster than the other side (thus making the robot take the path of an arc). Tank drivetrains prioritize traction and acceleration over pure maneuverability, giving these drivetrains the potential to traverse obstacles and play defense. Tank drivetrains are relatively simple to build, yet are still competitive at the highest levels.

Holonomic

A holonomic drivetrain, in contrast to a tank drivetrain, can move sideways, due to using either mecanum or omni wheels. These kinds of wheels have special rollers that allow strafing movements. Thus, holonomic drivetrains prioritize movement over traction. Holonomic drivetrains eliminate the time it takes to turn for a tank drivetrain. However, holonomic is susceptible to defense and can suffer with a heavy robot. Holonomic has been proven to be competitive at the highest level for many years, and is common among world-class robots.

When building any mechanism, teams must list out some necessities or desired features. Here are some priorities for that we think are important for each drivetrain:

1. **Reliability**: The drivetrain must work every time, not break, and if there are flaws, these flaws must be reliably present so they can be corrected. One aspect of reliability to consider is the type of motor and gearbox that is used in the drivetrain. For example, spur gearboxes are much more likely to break under load than a planetary gearboxes. (see motor guide for details). Thus, spur gear motors are not the optimal choice for drivetrain, especially if the robot is projected to be on the heavy side (30+ pounds). Reliability, the key to success in FIRST Tech Challenge, starts with the drivetrain, the foundation to any robot Our advice for new teams is to stick to the simpler drivetrains such as 4 or 6 wheel drive and mecanum drive.

2. **Agility:** There are many factors to agility: top speed, acceleration, turning radius, turn speed, and ability to strafe. Note that turning radius is an often overlooked feature that is critical to the overall agility of the drivetrain. Generally, a solid drivetrain should have a

maximum speed of around 2-3.5 feet/second, and be able to turn fully in around two seconds. A good resource to determine the agility of a drivetrain is the JVN for FTC calculator - refer to that section for more information.

3. **Number of motors and gear ratio**: Generally, new teams may try to use only two motors on the drivetrain. While this is possible, it is not recommended, as all competitive teams use four motors on the drivetrain. Another issue stemming from experience is that most teams' drivetrains are too slow. More advanced teams may focus on the ability to play defense, but in general, maneuverability and speed are the main factors to a successful drivetrain. 60:1 and 40:1 drivetrains are ***almost always* too slow for FTC use cases**. A popular setup in FTC is **4 motor, 20:1 using 4 inch wheels**. This ratio gives a great balance, having near instant acceleration and a high top speed. Teams can slow the drivetrain down in code by providing less power to the motors if needed. **It is not recommended for teams to use spur gearboxes on their drivetrain. Instead, use planetary gearboxes, as they are less prone to shock loads and breakage.**

4. **Traction/Pushing Power:** While this feature is often overemphasized, it is still very important. Pushing power describes drivetrain's ability to endure defense/engage in defense. Many factors affect the pushing power of a drivetrain, including wheel type, motor gearing, and overall weight of the robot. Note that if you already have a very agile drivetrain with experienced drivers, a team can avoid defense instead of having to fend it off or engage in it.

5. **Powering the drivetrain:** Generally, there are four options for power transmission: direct drive, chain, gears, and belt. More detail about each option can be found in the power transmission section. Teams should *stay away from direct drive*, as gearboxes are prone to breaking, especially under shock loads (for example, if the wheel is hit by another wall, or the wheel slams into the wall). We recommend belted drivetrains, but realize that belt is a difficult option for new teams. Chain and gears are also great options - chain requires a bit more forethought, as 1+ tensioner per side is required to maintain correct tension in the chain. CAD or a detailed sketch is generally recommended with chain in order to visualize the chain run (where the chain will be placed). Gears are slightly simpler, and can be a fantastic and easy option, especially when using extrusion. We would advise to stay away from TETRIX gears, and use the gears from other kits such as the REV delrin gears (*with hex hub strengthener*) or REV aluminum gears.

An important step is determining what you want out of your drivetrain. Do you want speed? Pushing power? Ability to get over terrain? Do you need to strafe? All of these questions need to be answered before choosing a drivetrain.

S-tier

6 wheel drive
Mecanum

A-tier

4 wheel drive
8 wheel drive
H-drive

B-tier

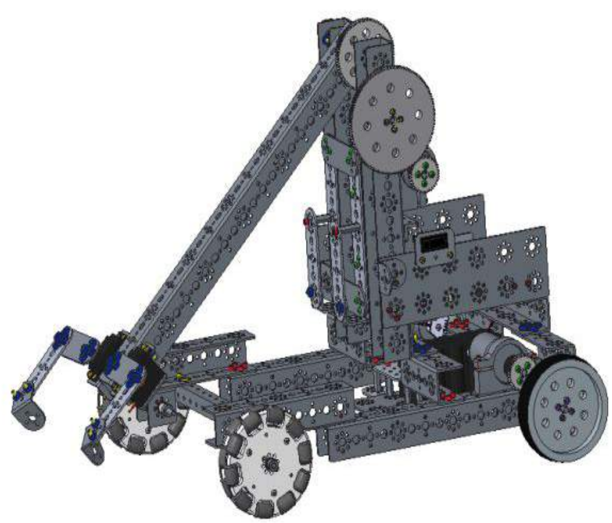
2 wheel drive
X-drive
Exotic drivetrains

Tank (Skid-Steer) Drivetrains

2 Wheel Drive (Pushbot Drive)

Recommended only for first drivetrain, not for competitions

This rookie drivetrain is considered one of the inferior drivetrains, though it is definitely usable at low to mid-competition levels. This is the introductory drivetrain type for many rookies, as it is promoted in official guides published by FIRST (giving it the name pushbot). It often has direct driven traction wheels with unpowered omni wheels. This type of drivetrain has poor turning as the center of turning is at the back of the robot between the two powered wheels. In comparison to other drivetrains, it has poor acceleration due to only using two motors. Even though it may not be an optimal drivetrain, it is still possible to be competitive as long as the drivetrain is reliable. As a consensus, we would advise every new team to build the pushbot for educational purposes only. The pushbot is a good starting point and helps the team get started with using kit parts; however, it is subpar to every other drivetrain in a competitive context. **While the pushbot is a good first drivetrain for new teams to get familiarized with the kit, it is recommended that teams move away from this drivetrain when building their competition robot.**



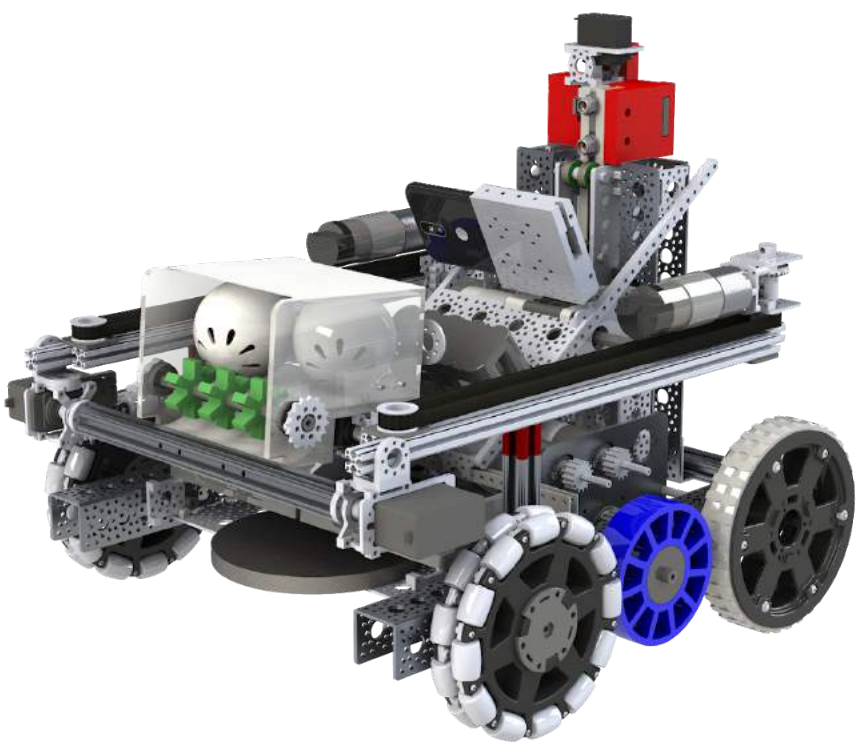
Example pushbot drivetrain included in the starter FTC guide

4 Wheel Drive

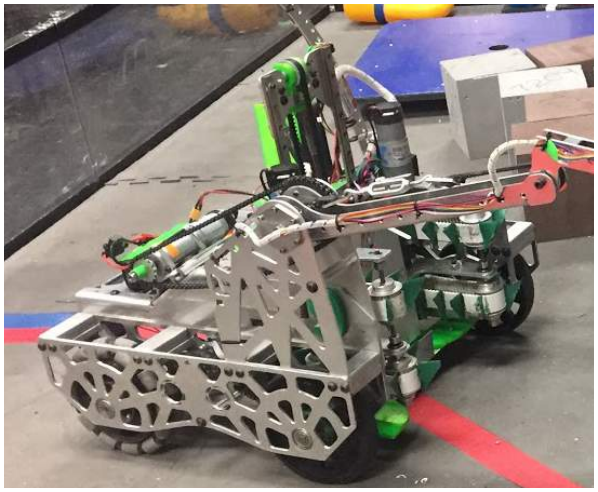
Recommended conditionally

In its most common form, the drivetrain uses the same wheel layout as the two wheel drivetrain (2 traction wheels in the back, 2 omnis in the front), but with one notable difference: all four wheels are powered. Some teams will put all four traction wheels or use all four omni wheels. It is not suggested to use all 4 traction wheels due to poor turning that results in this setup. This is caused by wheel scrub (friction between the side of the wheel and the floor tile)**.** Using omni wheels for all four wheels will result in incredible turning, with the robot rotating around its center. However, this advantage comes with a major loss of traction. For these reasons, many teams choose to use two traction wheels and two omni wheels for a balance between quick turning and traction. Even though 4 wheel drive is a huge step up, **unless it is needed to have a raised drivetrain to get over terrain, it is recommended for teams to move up to a 6 wheel drive due to the increased stability and pushing power of a 6WD.**

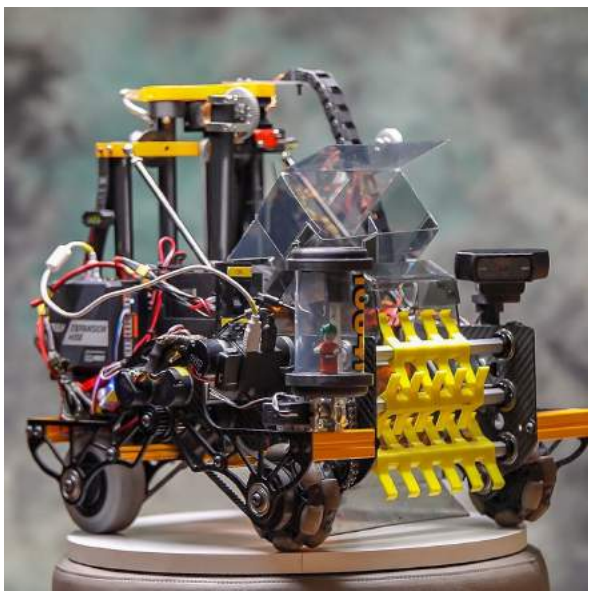
It is suggested that a four wheel drivetrain be close to, or exactly, a square. Otherwise, one may encounter problems turning. **Weight distribution is furthermore a large factor that should be considered: the more weight in the back, the better.** Off-center turning, which may or may not be a drawback, is nearly ubiquitous among 4 wheel drives. This may not be a problem for teams, but it is good to be aware of it. Off-center turning can be an advantage, but be warned that turning will be slightly slower on 4 wheel drives than their six or eight wheel counterparts.

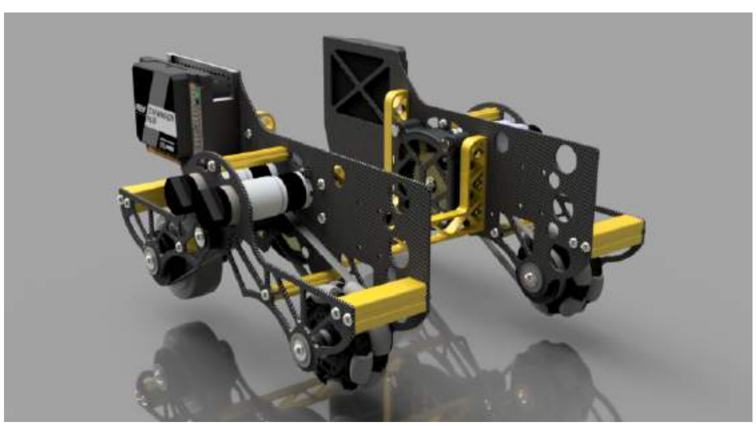


3736 Serious Business, Rover Ruckus; modified 4WD



7209 Tech Hogs, Relic Recovery





10641 Atomic Gears, Rover Ruckus

6 Wheel Drive (6wd)

Highly recommended

A 6 wheel drivetrain is one of the most common competitive drivetrains in FTC for multiple reasons: it has fantastic traction, great turning, and by having 6 wheels, the drivetrain has more contact with the ground, helping with stability and traction. There are two main types of 6 wheel drivetrains: ones with **corner omnis** and ones with a **drop center.**

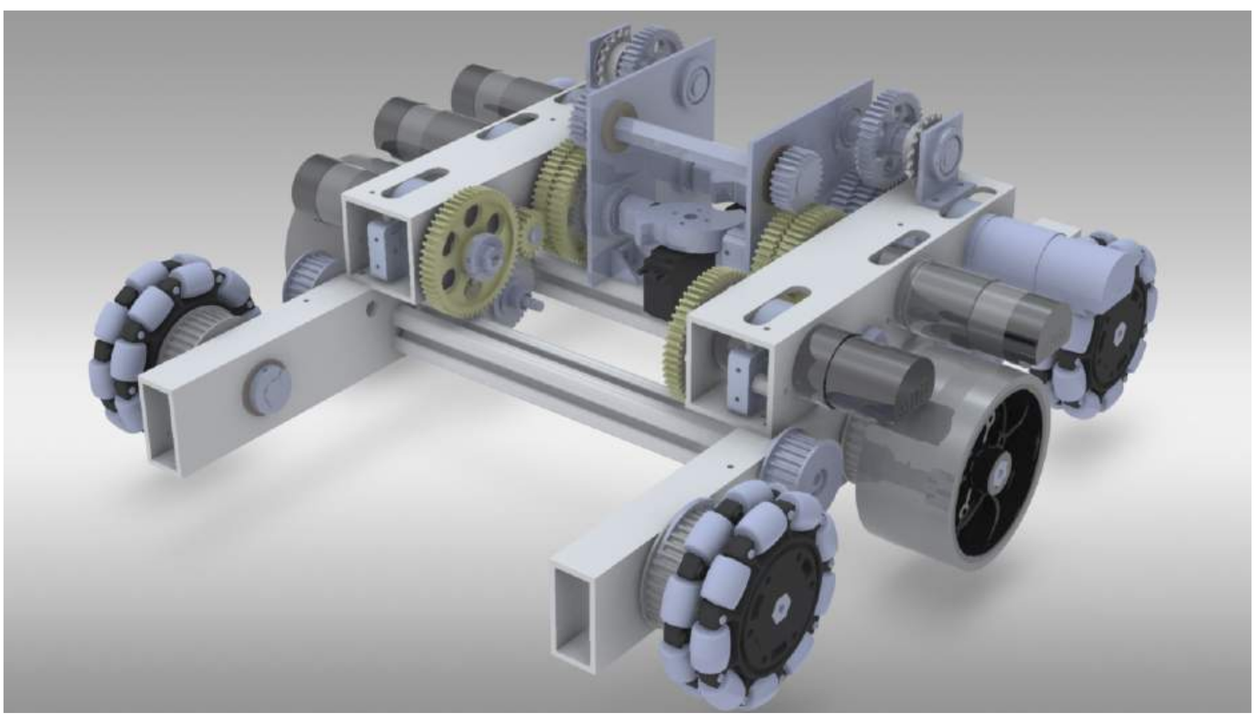
A drop center 6wd is a 6 wheel drive with the center wheel mounted slightly below the other two wheels. The drop should be anywhere from more than 1/16" up to ¼". However, the recommended drop is around ⅛". The purpose of dropping the middle wheel is to ensure that only 4 wheels are in contact with the ground at all times. This is because turning with 6 wheels on the ground is very difficult. Note that the **required drop may vary depending on both field condition and weight of the overall robot**. Turning can drastically degrade due to a difference in material underneath the field, leading to the robot sinking down further than usual. Moreover, **most drop center 6 wheel drives are made using custom drivetrains** because it is difficult to get the center wheel drop using a kit based build system (a notable exception being REV). It is possible to execute a drop center using goBILDA and Actobotics using pillow blocks, but it is a little bit more awkward.

6 wheel drive with corner omni wheels attempts to solve the issue of turning by replacing the corner traction wheels with omni wheels, allowing the drivetrain to achieve fantastic turning, albeit with slightly less traction than a center-drop. This is very easily buildable in kits, and is a great all-around drivetrain. Drop center and corner omnis can be combined for maximum turning reliability, although this comes with side effects like rocking and reduced traction.

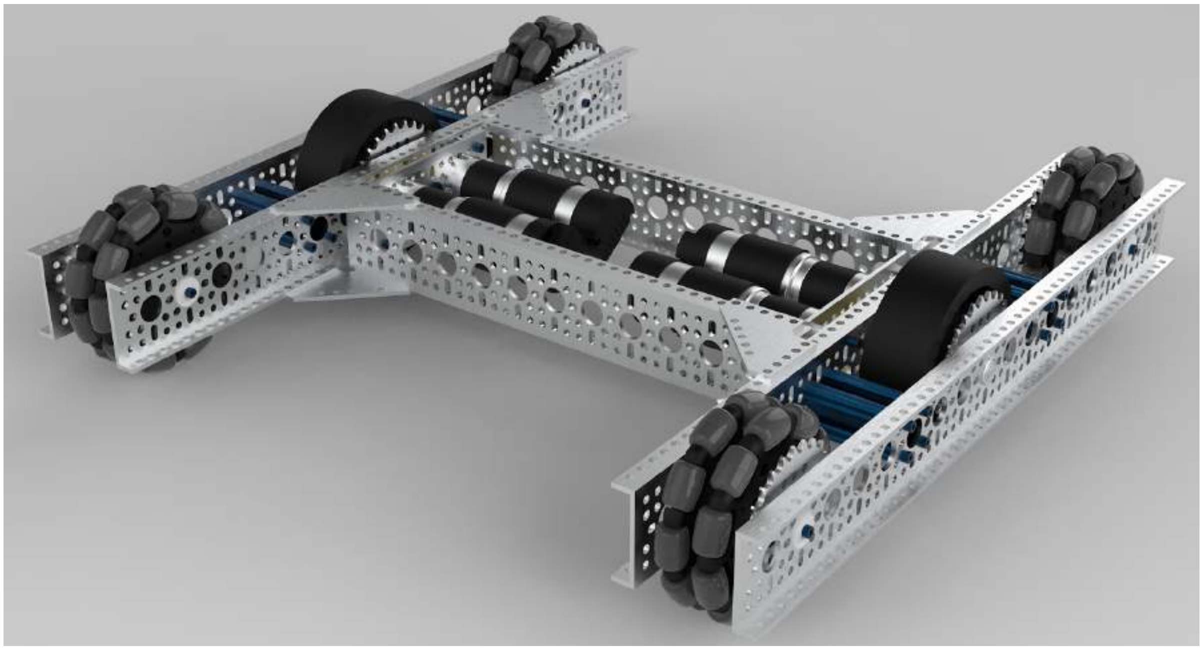
(Side note: tolerances in the size of the wheel may affect the traction of this drivetrain - many VEX Robotics Competition teams have built 6WDs like this only to find that their center traction wheel wasn't touching the ground due to variance in size. We don't have data on this phenomenon with FTC-class wheels, but it's something to be aware of.)

COTS Andymark Tilerunner Drivetrain with 0.05" center drop. **Not recommended due to price.**

7236 Recharged Green, Rover Ruckus; drop center 6WD



5975 Cybots, Velocity Vortex, drop center **and** corner omni 6WD



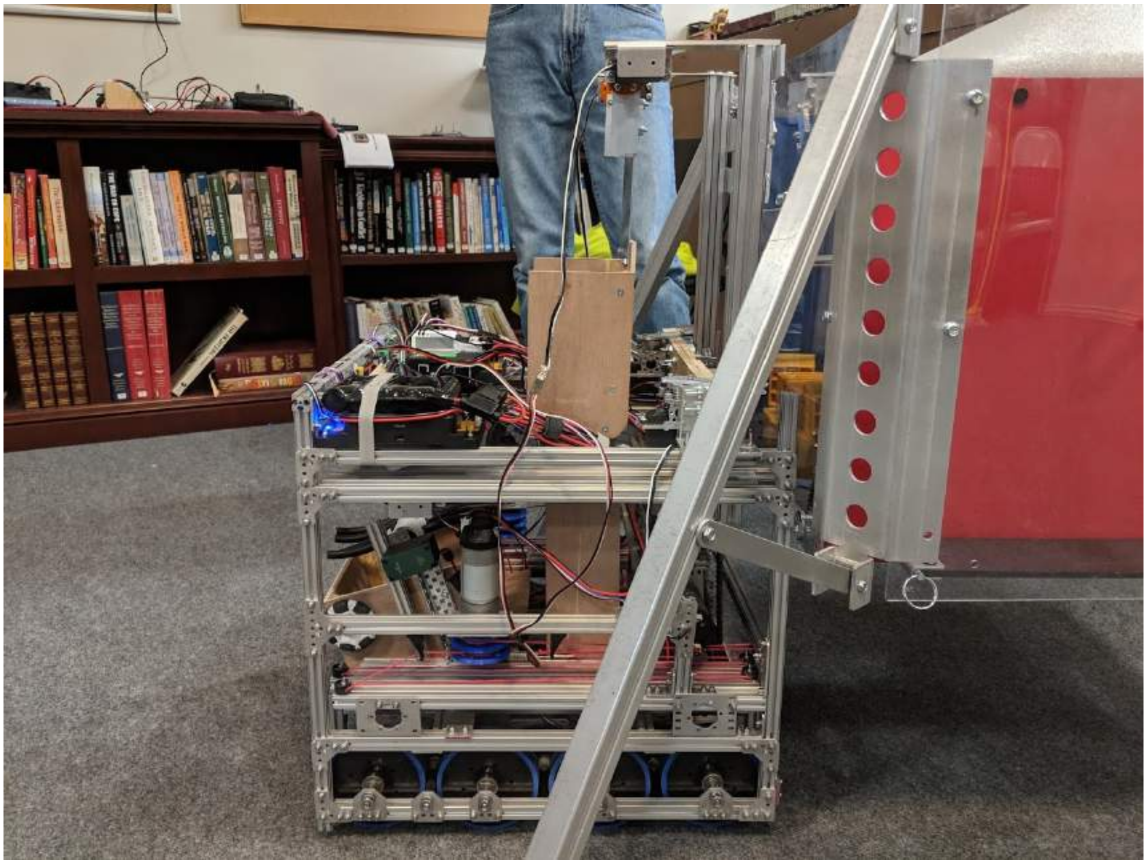
Ethan Doak, goBILDA 6 wheel drive with corner omni wheels

8 Wheel Drive

Recommended

An 8 wheel drivetrain is less common than its 6wd counterpart, combining elements found in both 4 wheel and 6 wheel drivetrains. For example, the 6 wheel drivetrain generally will have a dropped center wheel so that the robot turns on four wheels instead of six, reducing friction and increasing turning mobility. On an 8 wheel drive, the center four wheels are dropped. This means that when turning, only these middle four wheels are touching the ground. Thus, the 8 wheel drivetrain has slightly more stability while turning than a 6 wheel drive, whereas 6 wheel drives can turn more quickly. Furthermore, since the 8 wheel drive has wheels in the same place as a 4 wheel drive, it has the stability of a 4 wheel drive. It is suggested that all eight wheels should be powered, and planetary motors should be used over a spur gear motor. Teams have the option of using doubled omni wheels on the outer four wheels. Doing so will reduce traction/pushing power and increase mobility.

3846 Maelstrom, Relic Recovery



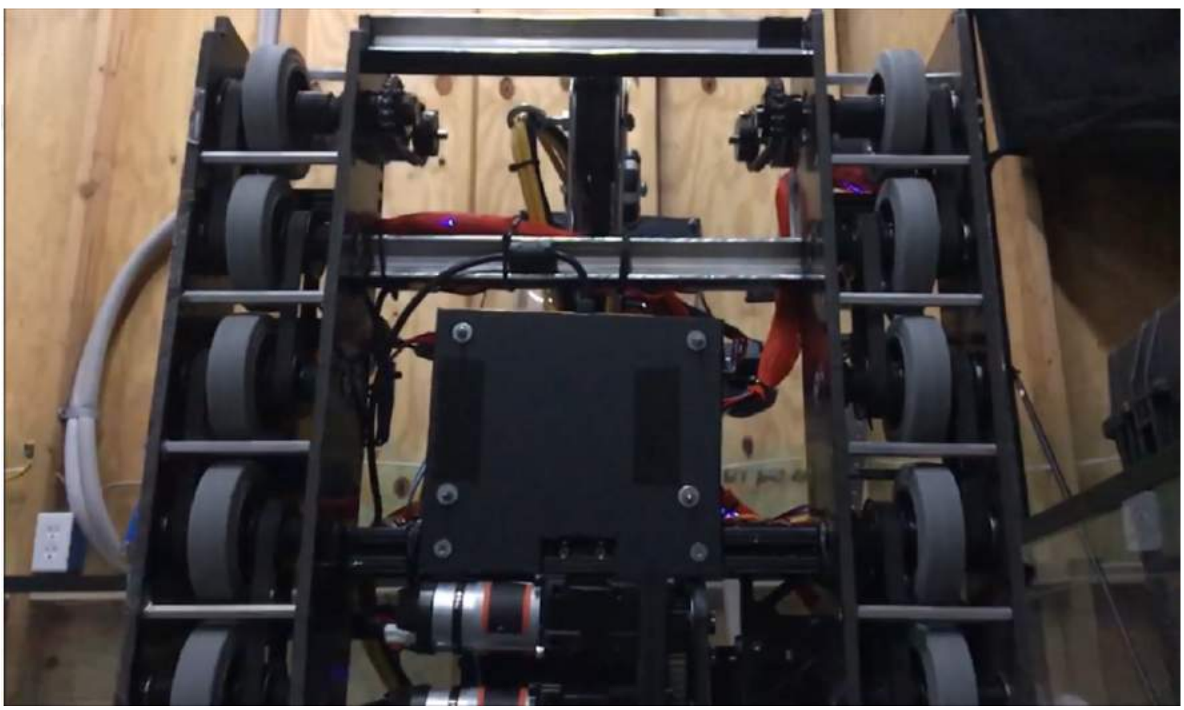
13075 Coram Deo Academy Robotics, Rover Ruckus

10 Wheel Drive

Not Recommended

A 10 wheel drivetrain is similar to an 8 wheel drivetrain in form and function. It provides slightly more stability and traction at the cost of turning mobility. Generally, there isn't a reason to pursue 10 wheel drive.

3736 Serious Business 10WD prototype



12670 Eclipse, Rover Ruckus

Tread Drive

Highly discouraged

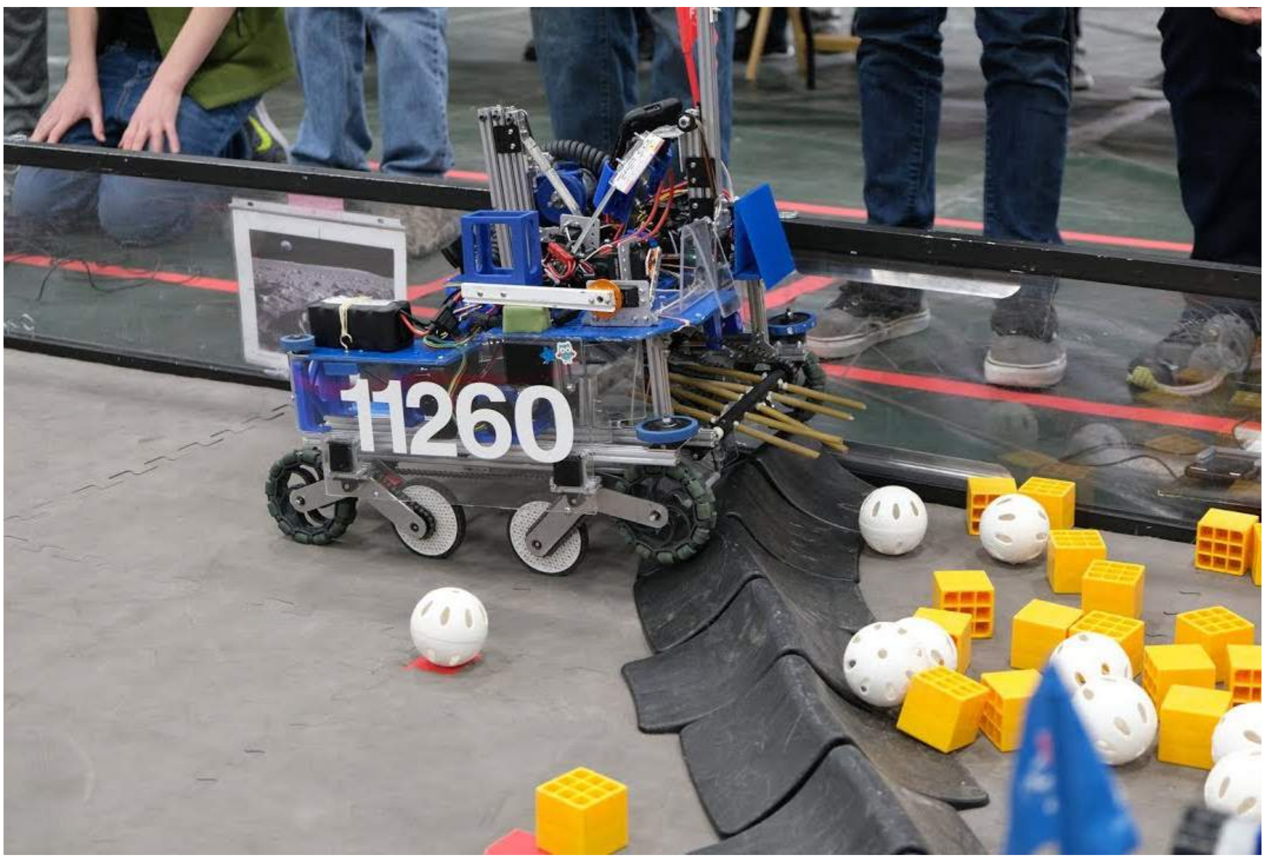
Tread drive is the use of tank treads or wide belts to power movement, much like a real life tank. Unfortunately, in FTC, it is not a competitive drivetrain for a number of reasons. Tread is complex, and has many points of failure. Treads are also very prone to defense, and a slight hit from another robot is enough to misalign the treads. Commercial Off-The-Shelf (COTS) tread options aren't great either - TETRIX tracks have a tendency to snap and derail when used on robot drivetrains, making them not suitable for competition use. To develop a competitive track drive, team 5975 CYBOTS collaborated with their sponsor Brecoflex to develop a custom application-specific tread and pulley system, a huge investment of funds that could be used elsewhere. Even with custom treads designed in collaboration with professionals, 5975's tread drive saw negligible traction improvements at the cost of maneuverability. As a general rule, **please do not do or even consider tread drive and instead build a 6 wheel drive, 8 wheel drive, or a 4 wheel drive.**

5975 Cybots, Res-Q

Rocker-Bogie

Highly discouraged

This type of drive is one of the unique, exotic drives that are designed specifically to traverse terrain. Compared to tri-wheel, it is a more viable type of drive, however, it is relatively mechanically complex. Inspired by space rover designs, rocker bogie relies on allowing the wheels to move up and down to conform to the terrain. Only two teams have used this type of drive and had competitive success: 11260 Up-A-Creek Robotics (Rover Ruckus) and 4140 Fish in the Boat (Ring it Up). **Because of rocker-bogie's complex design and the requirement for machining, even though it possesses some advantages while going over terrain, it does not possess enough to make it recommended to newer/inexperienced teams.**

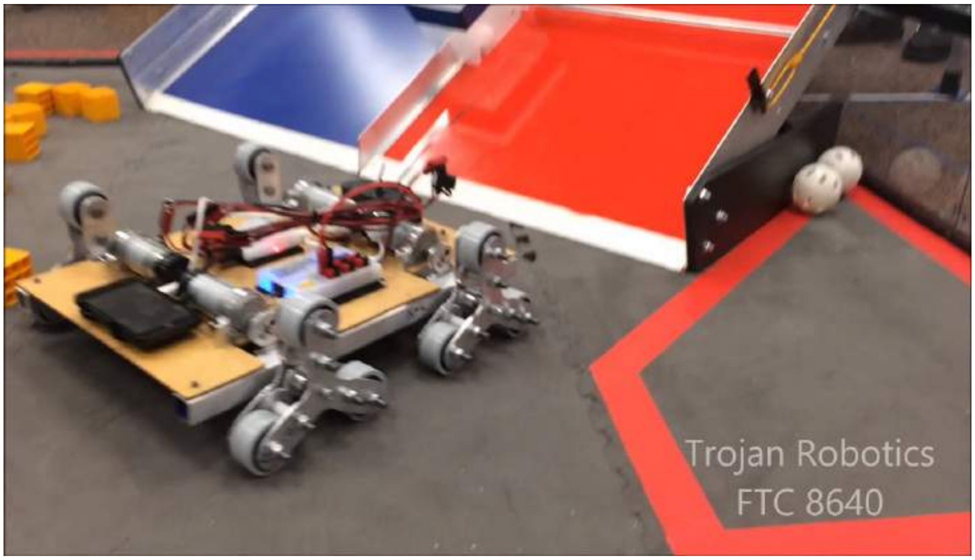


11260 Up-A-Creek Robotics, Rover Ruckus

Tri-wheel

Very highly discouraged

Tri-wheel is another exotic drivetrain that is specifically designed to get over terrain. This type of drive gained great popularity in Res-Q. **It's hard to build, has a lot of flaws, and doesn't even effectively climb terrain compared to a raised drivetrain and even worse than a well designed rocker-bogie. There is no reason to consider or use a tri-wheel drivetrain.**



8640 Trojan Robotics, Res-Q

7209 Tech Hogs Robotics, Res-Q

Holonomic Drivetrains

Mecanum Drive

Highly recommended

Mecanum drivetrains consist of four mecanum wheels. Each wheel is powered independently by one motor. This configuration angles the velocity of each wheel, allowing the robot to strafe. The rollers on mecanum wheels form a 45 degree angle with the wheel's axis of rotation. However, contrary to intuition, mecanum drivetrains can't strafe as fast as they can drive forward. This can be explained by discussing the forces involved. When each wheel rotates, it applies a friction force to the ground, which moves the robot. When moving forward, both sets of left wheels rotate in the same direction at the same speed, and both sets of right wheels rotate in the same direction at the same speed, meaning that the forces do not oppose each other. However, when strafing, neither the two left wheels nor the two right wheels are rotating at the same speed. In many cases, they even rotate in opposite directions. These two opposing forces cause the rollers to slip more and more, which angles the robot's velocity at the expense of traction (more slipping results in a loss of speed). However, the wheels do still slip when moving forward but not as drastically as they do when strafing. This is the primary disadvantage to mecanum drivetrains: they tend not to have much pushing power and thus, are vulnerable to defense by a sturdy tank drive. Due to the fact that mecanum wheels are more likely to slip, an optional addition to mecanum drives is a separate odometry mechanism in order to track the robot's location, particularly during autonomous. For more information, see the Odometry section of the programming guide. Despite their complexity, **mecanum drivetrains are both extremely effective and easy to build, making them a great choice for both new and experienced teams looking for a drivetrain with holonomic movement.**

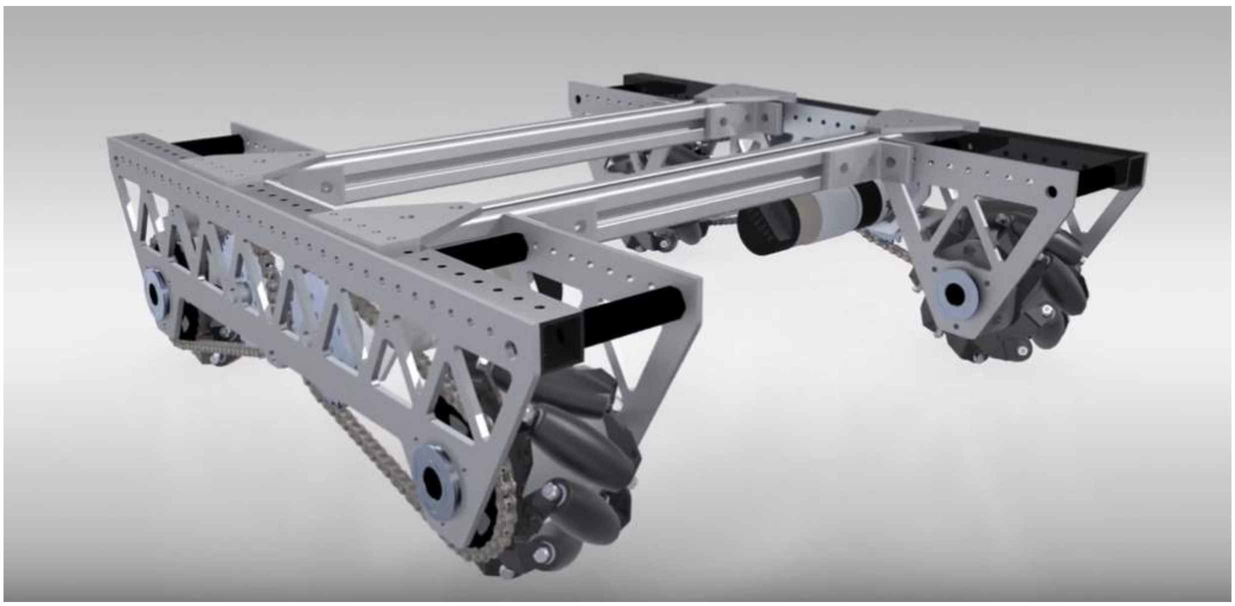
It is important to note that in order to maximize the efficiency and stability of mecanum drives, when viewed from above, the rollers of each wheel should point towards the center of the robot, forming an X shape, rather than a rhombus. The primary reason for this is that it allows the drivetrain to turn significantly faster than it would otherwise be able to. When using the suggested setup, when viewed from the robot's underside, the rollers form a rhombus. This allows the force applied by the wheels on the ground to act tangent to the turn radius, leading to faster turning.

Mecanum Wheels Miniguide:

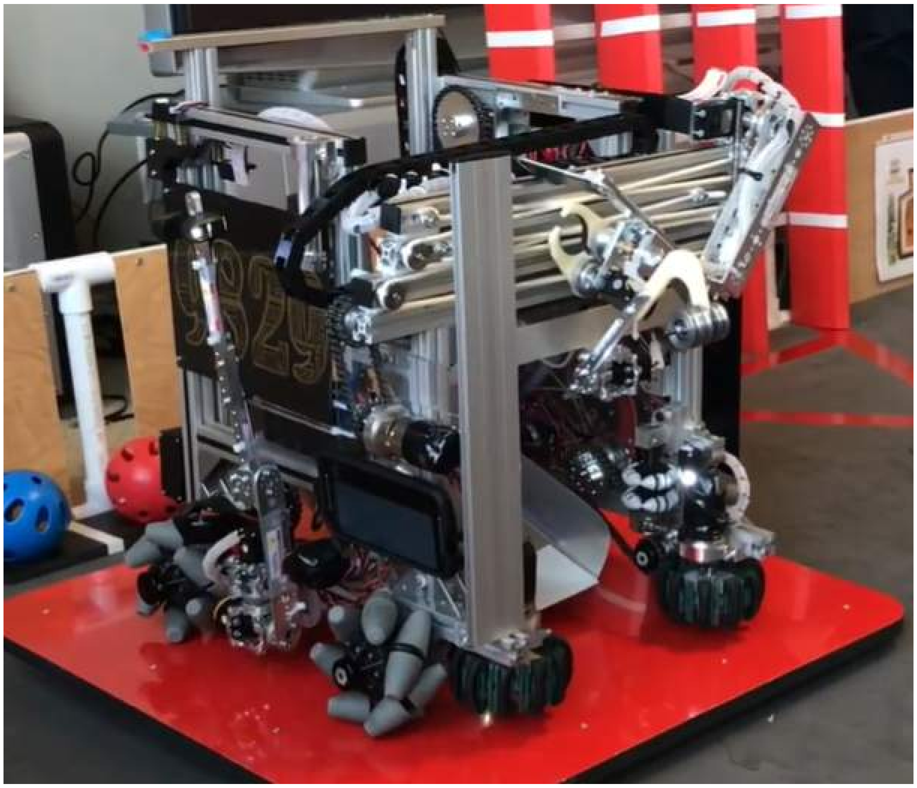
There are plenty of mecanum wheels on the market, and it can be very daunting to choose between the many vendors. An important feature is the type of mechanism that facilitates the motion, either **bushing** or **bearing**. Bearing based mecanum wheels often have superior strafing because there is less resistance for the rollers to overcome. Another important note is that some FTC teams invest in 6 inch mecanum wheels instead of 4 inch mecanum wheels, often at a much higher price. **It is highly recommended that teams stick with 4 inch mecanum wheels.** Here is a general list of the mecanum options ranked in order of recommendation.

1. **goBILDA Mecanum Wheels ($105 with team discount) (link):**This recent addition to the lineup has become one of the strongest options for its variety of positive attributes. goBILDA Mecanum wheels are based on the tried and tested Nexus bearing mecanum wheels, which means it has fantastic strafing. It is also very robustly built, and is significantly more convenient to mount to FTC standard build systems. It has a built in goBILDA 16mm and 32mm hole pattern, and has easy support for dead axle. You can mount hubs in wheels and goBILDA mecanum wheels can easily mount to ½ Hex, ⅜ Hex, 12 mm REX, 6 mm D, ¼ D, and many other shafts. Hubs can also be mounted inside the wheel for very low profile mounting.It is also the cheapest bearing mecanum on the market. Due to its convenient mounting and fantastic strafing performance, we recommend all teams consider goBILDA mecanum wheels.
2. **Nexus Bearing Mecanum Wheel ($134) (link):** This was the old gold standard, and still has fantastic performance for the price. This has identical performance with goBILDA mecanum wheels, however is slightly less convenient to mount to. However, these wheels feature the 1.875" bolt pattern commonly used in FRC motion products. It is also slightly heavier than goBILDA Mecanum wheels. Many teams will 3D print adapters or build new cores for Nexus Mecanum wheels. Even though the goBILDA mecanum offer advantages and very few disadvantages over nexus bearing wheels, these wheels remain a solid option.
3. **AndyMark Heavy Duty 4" Mecanum Wheel ($225) (link):** These are easily the most expensive mecanum wheels on the list. These are bushing based mecanum, so they have decent strafing, albeit not as good as the goBILDA and the Nexus bearing mecanum wheels. What sets these mecanum wheels apart is the 80A roller material. AndyMark HD mecanum wheels have higher traction than all other mecanum wheels, which make them desirable for climbing terrain. For example, during the Relic Recovery season, teams had to climb a "balancing stone", and many teams chose to use the AndyMark HD mecanum wheels to be easily able to climb the balancing stone. However, in most cases, being able to more effectively strafe is more important than having good traction. For this reason**, teams are recommended to buy bearing based mecanum wheels like the Nexus or the goBILDA mecanum wheels instead of the AndyMark HD wheels due to the major price difference.**
4. **Nexus Bushing Mecanum Wheel ($84) (link):** This is the Nexus Bearing Mecanum wheel with bushings instead of bearings. Before the introduction of the goBILDA mecanum wheels, these were the best budget option, however, it is now more sensible to spend the $21 premium to get the bearing goBILDA mecanum wheels.
5. **VexPro Mecanum Wheels ($119.96) (link):** These mecanum wheels are most suited for vectored intakes on FRC robots. They are relatively tough, but have somewhat poor strafing and are not quite as durable as the other wheels higher on the list. They are a decent choice if you already have them, but otherwise, there is no reason to consider them.

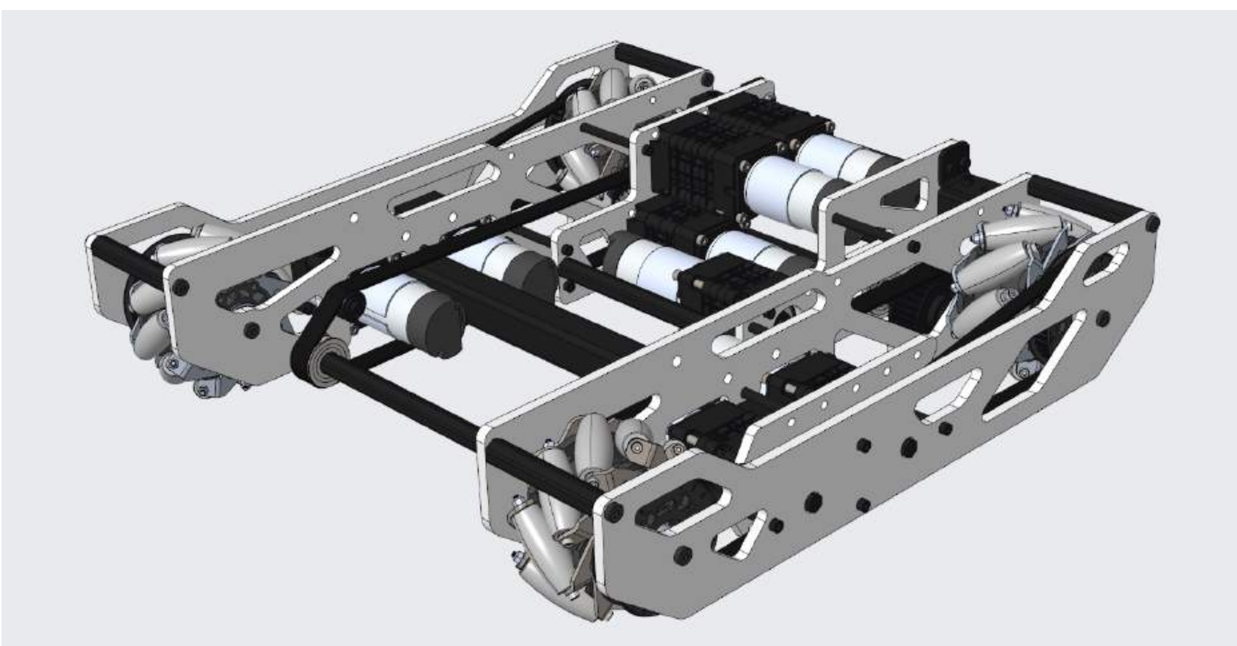
6. **TETRIX Mecanum Wheels** (**$113) (link**): At the time of writing this guide, these haven't been released so no verdict can be reached. However, they have a built in hub so they can be easily mounted on 6mm D shaft. They are bushing based, and due to no testing and evaluation outside of TETRIX, and its higher price than the goBILDA mecanum wheels, **we cannot recommend the TETRIX Mecanum Wheels.**
7. **VEX EDR Mecanum Wheels ($59.99) (link):** These are the cheapest mecanum wheels, but have a strange shaft standard (⅛ square) which require the use of 3D printed adapters. **There are not many reasons to purchase these wheels.**
8. **AndyMark Standard Duty Mecanum Wheels ($92) (link): DO NOT PURCHASE THESE WHEELS.** These are terrible mecanum wheels. They barely strafe and are super fragile. Just buy goBILDA mecanum wheels for $13 more.



8103 Null Robotics, Rover Ruckus, using Nexus mecanum



9829 MakBots, Relic Recovery, using **VexPro mecanum**

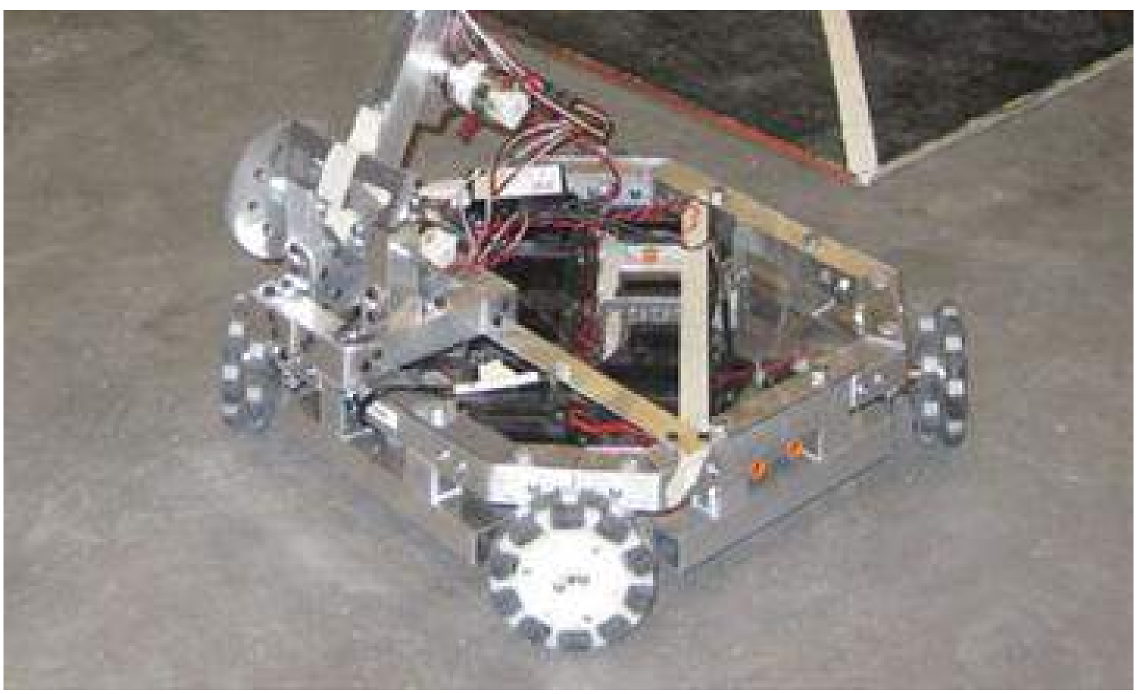


731 Wannabee Strange, Rover Ruckus, using AndyMark HD mecanum wheels

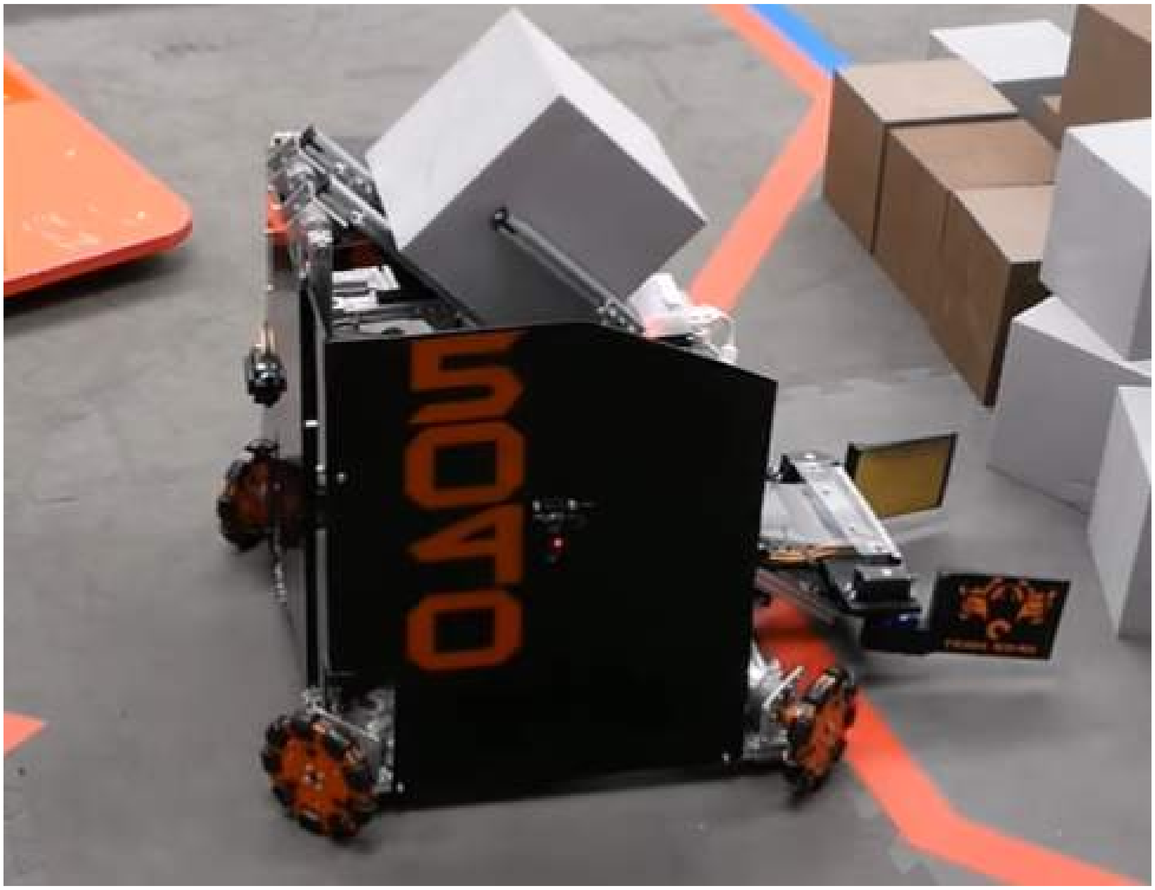
X-Drive

Recommended

X-Drive is probably the most popular of the holonomic omni-wheel based drivetrains. This type of drive involves mounting 4 omni wheels at the corner of the robot at a 45 degree angle. One notable difference between X-Drive and Mecanum is strafe speed. While, as mentioned in the mecanum section, the ratio of strafe speed to forward speed is noticeably less than 1, the ratio on an X-Drive is exactly 1 due to the rotational symmetry of the wheel placement. This means that an X-Drive bot's strafe speed and forward speed are equivalent. The drivetrains are slower, however, when strafing at 45° (approximately $\frac{\sqrt{2}}{2}$ of its forward speed). Even though X-drive has good turning and acceleration, the main downside to the drive is packaging/form factor. Packaging refers to how easy/convenient the drivetrain fits into the overall design of the robot. Ideally, the drivetrain should take up as little space as possible to make it easier to design mechanisms around. Because the omni wheels are offset, packaging a X-Drive is more difficult than other types of holonomic drive like mecanum or H-Drive. Also because of strange packaging, it is relatively hard to cleanly incorporate power transfer from the motors, meaning that most X-Drives end up being direct-driven, which is bad on the lifespan of the motor gearbox. **Due to packaging challenges, it is recommended that teams move away from X-drive and into using Mecanum. However, it is still one of the more viable omni holonomic drivetrains (better than kiwi) and plenty of successful teams have used this type of drivetrain.**



2844 Valley X Robotics

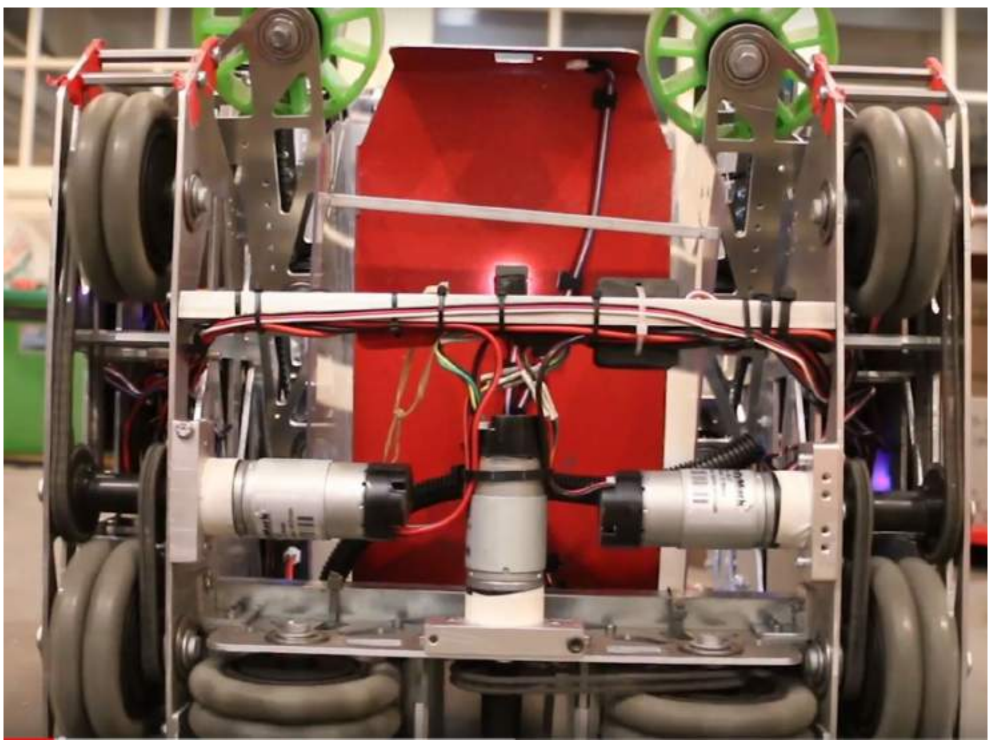


5040 Nuts and Bolts, Relic Recovery

H-Drive

Recommended conditionally

H-Drive (also known as U drive depending on the configuration) is a holonomic type drive that uses all omni wheels. H-Drive relies on a set of “strafer wheels” that are perpendicular to the forward/backward wheels to achieve strafing. H-Drive is one of the more viable and competitive holonomic drivetrains, providing reliable strafing and driving to teams. H-Drive is theoretically very easy to code, but most teams employ some sort of gyro correction to strafe straight, although it is not necessary with proper weight distribution. H-Drive has a number of possible motor configurations - 1 or 2 motors can be put on each forward drive pod, and one or two motors can be put on the strafe wheels. In this configuration, H-Drive has slightly reduced acceleration compared to mecanum drive. For the highest possible reliability, many FRC teams will suspend their strafe wheels on a rocker system to ensure that all wheels are in contact with the ground while the robot is not strafing. **While H-Drive is a viable option, mecanum drive has simpler packaging and better acceleration, making it the better choice in most cases.** However, **H-drive can be more practical depending on motor distribution.** For instance, if you want to dedicate only 3 motors to your mechanisms and you have a motor left over, using a 1 strafe motor, 4 drive motor configuration is absolutely viable. Or if you dedicate 5 motors for your mechanisms, H-drive with 2 drive motors and 1 strafe motor is definitely optimal.



9804 Bomb Squad, Relic Recovery

Kiwi Drive

Not recommended

Kiwi drive is probably one of the least optimal of the holonomic omni wheel drivetrains. It involves mounting three omni wheels on a triangle frame. It is incredibly painstaking as far as packaging goes, requires a little bit more complex code to drive, and basically forces your robot frame to be a triangle. **There is no reason to use kiwi drive. If you need to use an omni wheel based holonomic drivetrain, consider h-drive or x-drive.**

FRC scale kiwi drive

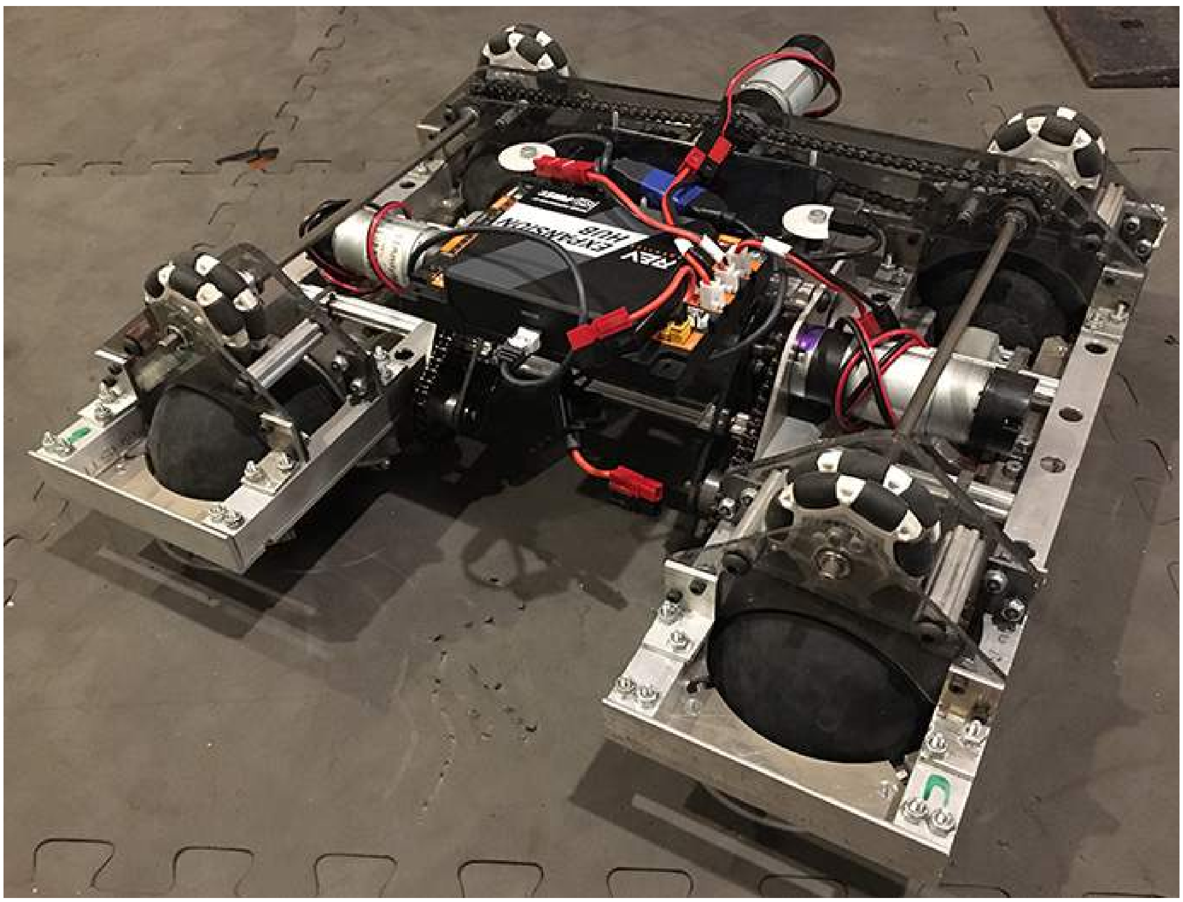
2818 G-Force, Velocity Vortex

Ball Drive

Highly discouraged

If you are not team 4042, Nonstandard Deviation, please do not attempt this type of drive. Thank you. :)

Instead driving using wheels, Ball drive instead uses balls powered by wheels to move. Small traction wheels touch the center of the balls, allowing for the balls to push the robot forward. Omniwheels positioned at the top allow for the balls to push the robot holonomicly as well. Ball drive shares many advantages with H-drive, but with the additional advantage of higher traction. However, ball drive is an incredibly advanced type of drivetrain. It has great turning and great traction and can strafe. Unfortunately, only one team has ever built that drive and they are a team with fantastic design and machining capability and they have a history of trying inventive things. **Ball drive is not recommended for any team due to its incredible mechanical complexity and not having enough advantages over a traditional holonomic chassis like mecanum.**

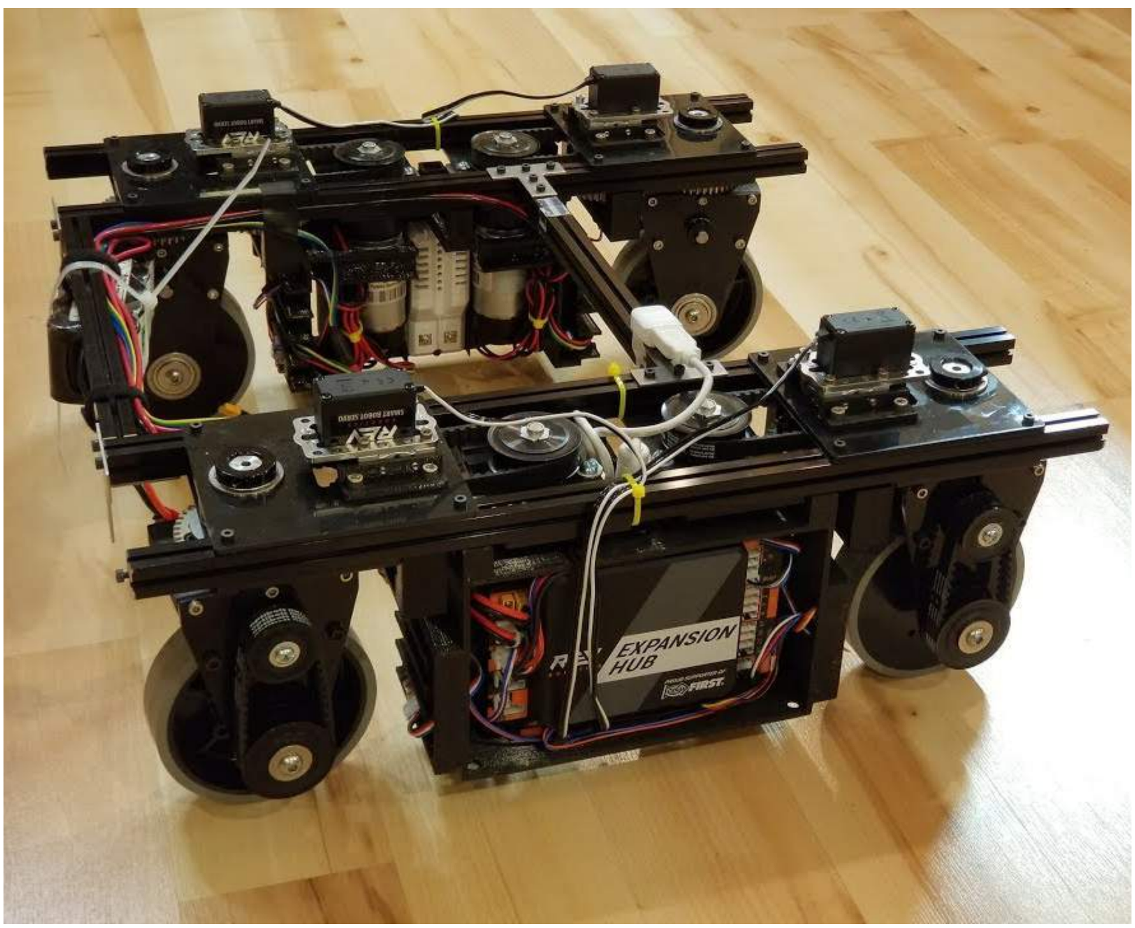


4042 Nonstandard Deviation, Rover Ruckus

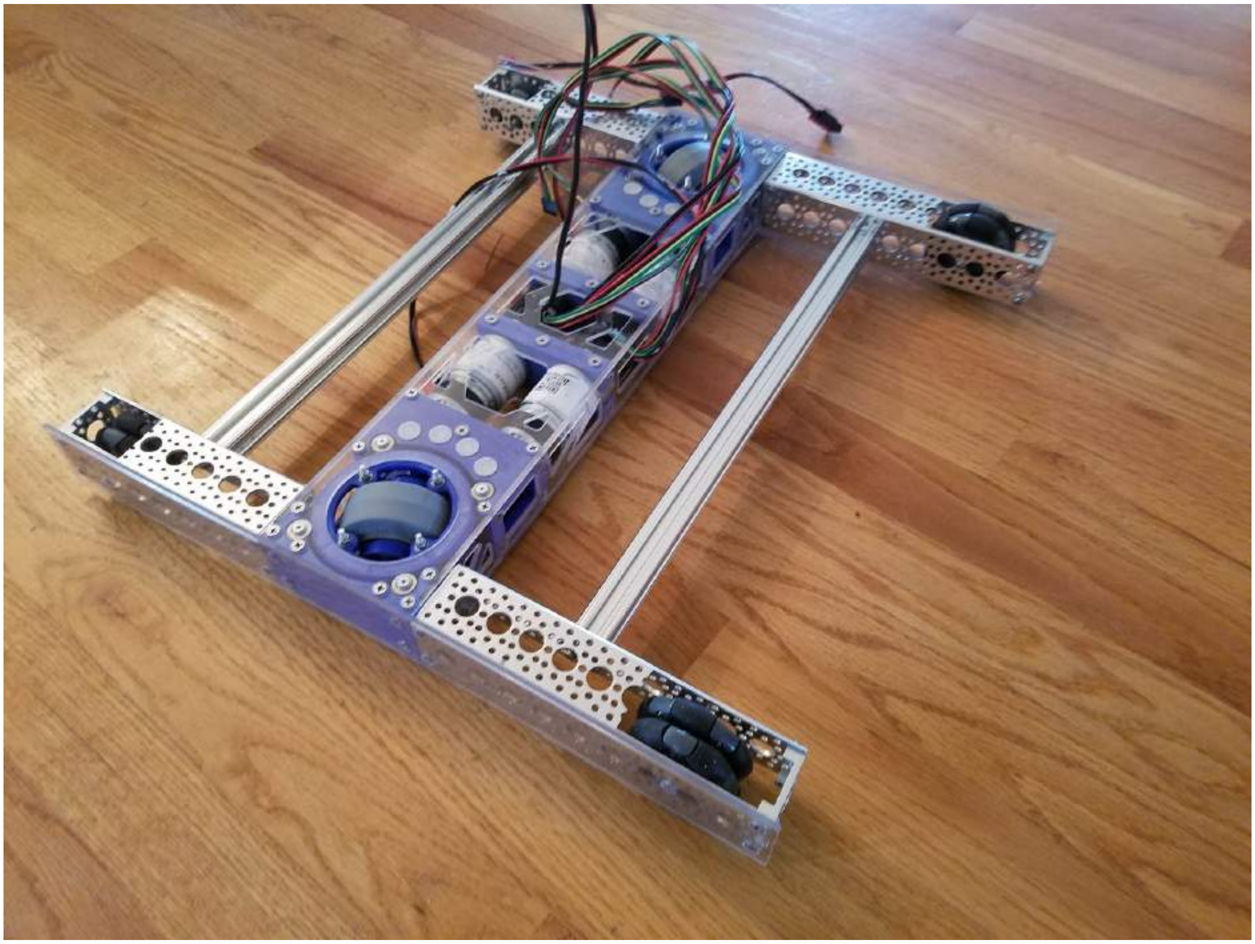
Swerve

Highly discouraged

Swerve drivetrains are, effectively, motorized cart wheels. They are typically composed of 4 wheels, each powered by one motor each. However, instead of being angled straight forward like the standard 4wd, or being angled at 45 degree angles like X-drives, the orientation of swerve wheels is not fixed. Instead, servos (or swervos, as the cool kids call them) are used to change the orientation of each wheel, allowing for holonomic movement. Theoretically, swerve has all the benefits of tank and mecanum, strafing, great traction, great turning. You can also strafe just as fast as you can drive forward. In fact, driving in any orientation is just as fast as moving forward. However, a swerve drivetrain is the most difficult type of drivetrain to build in FTC. It is highly discouraged for new teams, and poses an extreme challenge even for experienced teams. There are many types of swerve: Hotwired, coaxial, crab, and differential. However nearly all require very complex machining and often **do not have enough advantages advantages over mecanum to be worth the time and effort**. Due to limitations in FTC hardware (specifically the underpowered servos, number of motors, and current draw), it is seemingly impossible to make a feasible swerve drivetrain.



5975 Cybots, Relic Recovery



11115 Gluten Free Differential swerve prototype

Drivetrain in an Hour - CAD Tutorial

Baylor Edwards, 10641 Atomic Gears

Introduction

Disclaimer: CAD is not *necessary* in FTC to build a successful robot. Many successful teams didn't CAD their robot at all. Other successful teams only made parts of their robot in CAD. Still other teams fully designed their robot in CAD. It is encouraged that new teams try their hand at CAD, especially if you have a mentor or parent who is experienced in such an area. CAD is beneficial for two reasons. First, CAD solves a lot of preventable headaches, such as spacing issues. Thus, it will save time when you discover problems in CAD that you can remediate before you build your robot. Second, CAD is a professional tool that is used in many STEM fields, and having knowledge of CAD will be beneficial in your future career, should you study and work in those fields.
However, CAD is not the magic genie that will guarantee you success in FTC. When used properly, it is a great tool to aid teams in building their robot. Keep in mind, though, that many teams have had success without CAD.

My process of designing a drivetrain for FTC may be completely different from others, and is by no means the only way or even the best. However, it works extremely well for me and saves time when compared with other methods.

Starting off, choose a CAD program and learn it as well as you possibly can. Perhaps you can spend a few weeks just finding stuff in your house, sit down with a ruler or even a set of calipers, and make a quick cad model. You can also pick stuff from your favorite TV show, video game, comic, book, etc. It really doesn't matter - just find something you want to make and make it to the best of your abilities. This is the way I learned and, in my opinion, it was the fastest way for me to learn. There are also tons of videos on YouTube that help a lot. My favorite is TFI who makes Autodesk Inventor tutorials and walks you through what all of those buttons do and how to use them.

There are many possible CAD programs that teams can learn. All of these programs can be downloaded for free under a student or FTC team license. You'll have to do a bit of research here, as the requirements for free copies vary based on the program. Here are a couple suggestions to consider:

- **SolidWorks** is an industry standard CAD package made by Dassault Systemes. It's as fully featured as CAD software gets, including great simulation features and a very robust assembly environment. It's used widely in industry, but is also the program of choice for most college-level engineering classes. However, **it isn't available for Mac users**, and you'll need a pretty beefy computer to run it. If you have mentors or team members with previous experience in SolidWorks or an engineering class at your school that teaches SolidWorks, it will be your best choice.

- **Inventor** is Autodesk's industrial CAD offering. It offers many similar features to SolidWorks, but has a different UI and two distinct assembly modes. While it's used by many companies in the industry, it doesn't appear in very many college curriculums. **Inventor is also not available for Mac,** but it may run better on lower-spec PCs. If you have mentors or team members with previous experience in Inventor or an Engineering class at your school that teaches Inventor, it will be your best choice.
- **OnShape** is a fully-featured CAD package, but it runs entirely in the cloud. It can run on any computer (even Chromebooks!) and has iOS, iPadOS, and Android apps as well. It has all of the same core features as SolidWorks and Inventor, as well as the best collaboration workflow in the industry - many people call it "the Google Docs of CAD." Onshape allows multiple people to work on the same document at the same time, and allows users to "follow" each other and see what's on another user's screen. OnShape also has FeatureScript, a programming language where you can write custom features. The community has created a lot of very useful FeatureScripts already which you can use completely for free. Onshape also has a comprehensive tutorial system (https://learn.onshape.com) that will not only teach you how to use their software, but how to approach design problems. If you're just starting out with CAD, or you don't have access to powerful computers, or you want to have many people working on CAD; Onshape is the software for you.
- **Fusion 360** is a cloud-based all-in-one CAD/CAM package, also made by Autodesk. Fusion 360 is cross platform compatible, although it doesn't run very well on low-spec computers. It has a powerful CAM environment for machining your parts, and it has intuitive and easy cloud rendering that gives the heavy lifting to Autodesk's servers. To maintain a simpler UI, Fusion skips out on a lot of the more advanced features found in SolidWorks and Inventor, although this isn't really much of a problem. However, a more noteworthy difference is that Fusion ignores every single industry standard, creating its own structure and organizational system. Beyond simple sketches and extrude features, Fusion's modeling and assembly system is entirely foreign and not compatible with any other CAD software, making it very hard to switch away from Fusion. Because of this, if you aren't careful, Fusion's file hierarchy can actively encourage bad design habits and discourages reusability by allowing users to create new parts without designing them individually first. If one is careful to follow good design practices, Fusion is a solid option.
- Creo

After learning your CAD program of choice, determine the necessary requirements for your drivetrain based on the current game. You will definitely want to shoot for the wheelbase that works the best in that specific field's layout. For instance, in Relic Recovery (2017-2018) a drivetrain required precision to not only grab glyphs from the center pit, but also to line up against the cryptobox. Thus, mecanum wheels and a wide center section of the robot proved an advantage over a 6 wheel tank drive. (However, it should be noted that with sufficient practice and competent drivers, any drive base can be competitive up to a certain extent).

After selecting a drivebase, determine the number of motors. Keep in mind the eight motor limit is a pain that shouldn't be ignored. A good rule of thumb is four motors for driving and four motors for the other mechanisms (e.g. intakes, linear slides, arm, etc.) For most modern FTC games, you need minimum 7 motors to be highly competitive, although 8 is a good rule of thumb.

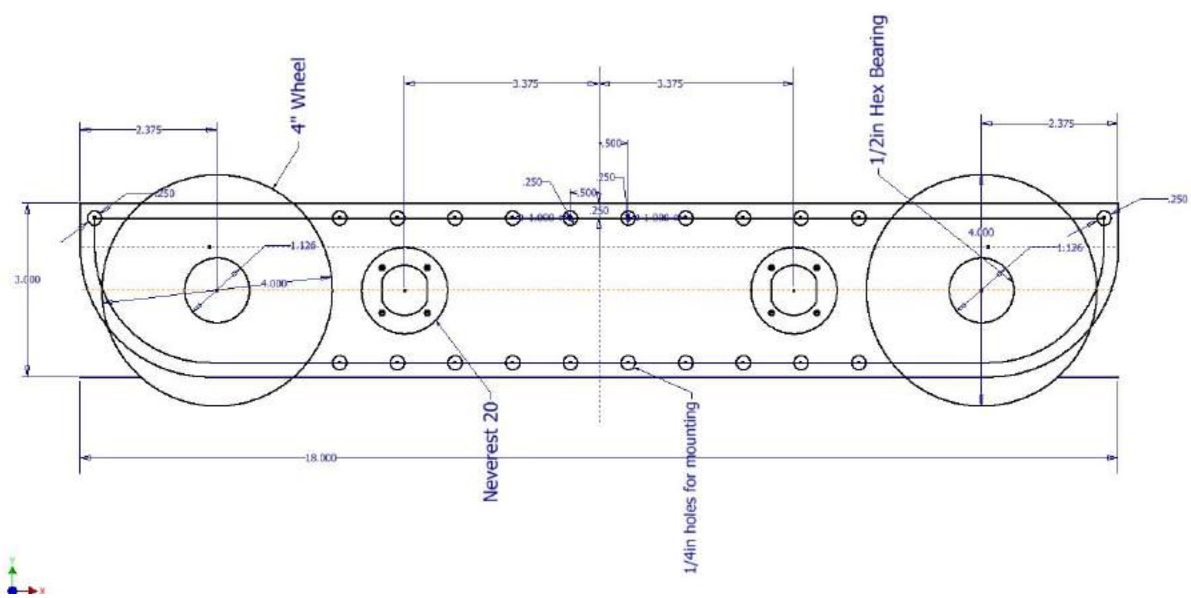
CAD and Design

After learning the CAD software, it's time to start the actual design. Here are some things you need to figure out before starting:

- Drive Type (mecanum, 6wd, 8wd, etc.)
- Number of Motors (four motors in most cases)
- Type of wheels (Colson, omni, etc.)
- Drive Power (belt, chain, gear)

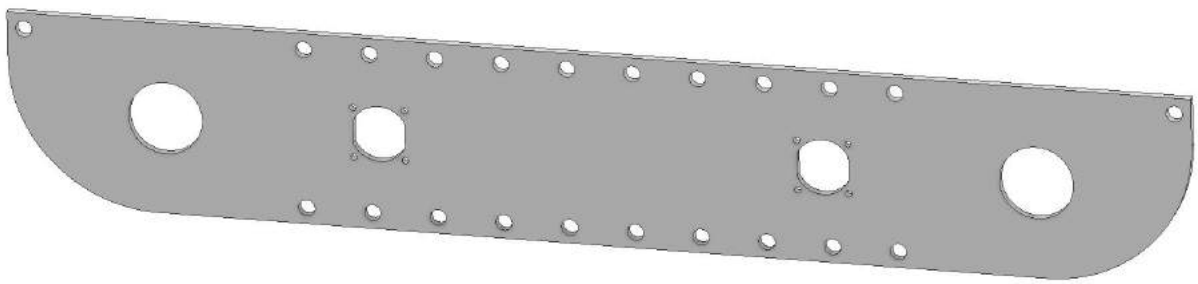
For this example, to keep it relatively simple, I'll be making a 4 wheel tank drive using four motors. The wheels I selected are 2 Colson wheels for traction, and 2 omni wheels to aid in turning.

First we will be making the left side of the drivebase. After completing it, all you have to do is mirror the left side to the right, so you don't have to do each side individually. Start with a 2D sketch of everything before trying to extrude and make actual 3D objects.



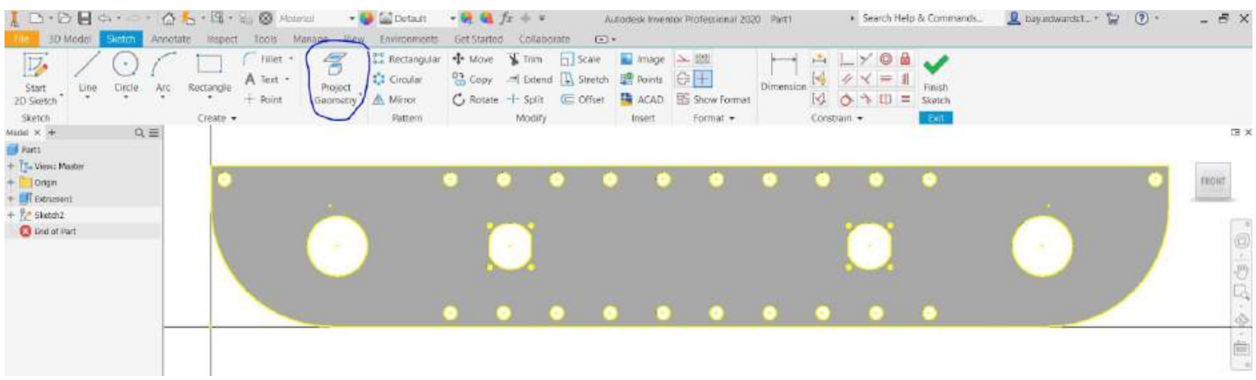
This is a sketch of the inner plate of the drive base. I laid everything out where they would go in a 2D sketch before anything else. 2D sketches are extremely helpful and I would highly recommend doing this before any project. After doing this, everything else honestly just falls into place and becomes pretty simple.

After this we can go ahead and extrude that sketch into our first plate of the drivetrain. Typically we use 1/8th in aluminum plate for our plates so extrude it to that thickness.

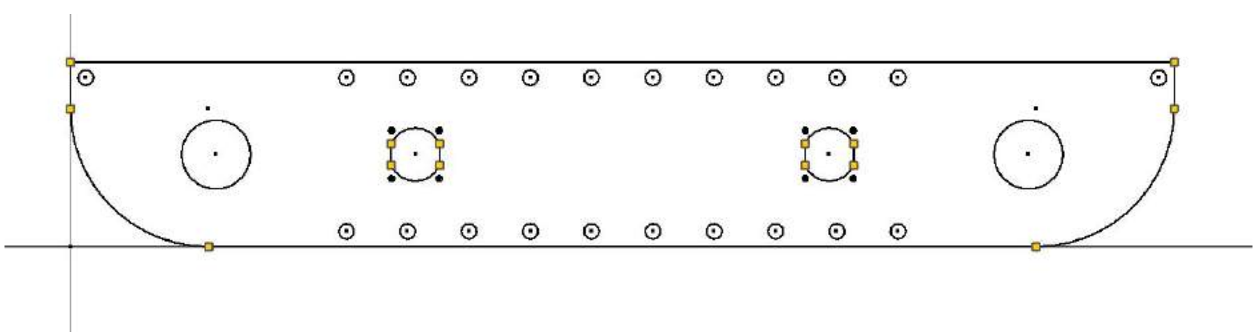


So this is what mine looks like after laying out and extruding the part.

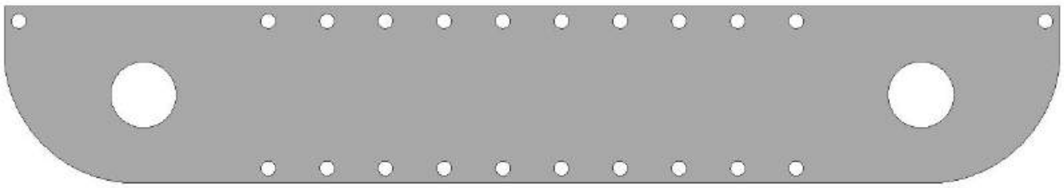
The next step will be making the outer plate for the drivebase. It is even faster to do than the inner. So to do this simply create a new part, after that go back to your inner plate and start a 2D sketch.



After starting the new sketch on the inner plate hit project geometry and just click anywhere on the part. It should highlight every outline of the part. (Mine is a yellow line; yours might be red, blue or some other color.) Now click and drag across the part selecting every line on the screen. Now go hit CTRL + C, then go to your new part you are making, and hit create 2D Sketch then hit CTRL + V.



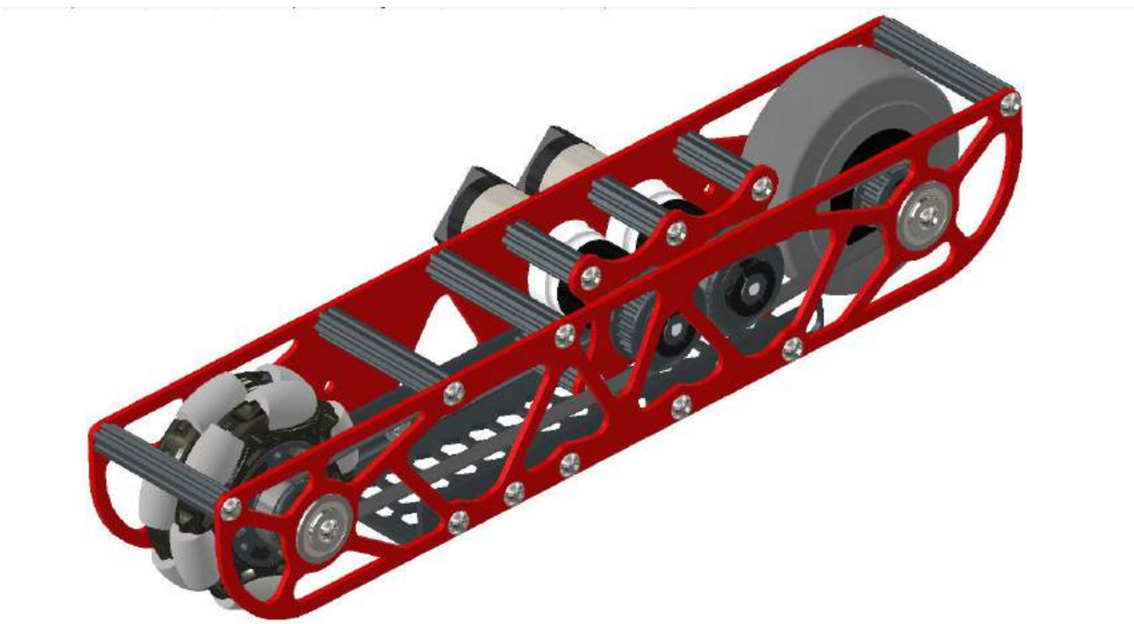
It should look like an exact copy of your inner plate but now as a sketch. Delete your motor mounts out of the middle, then extrude the outer plate.



This is what my outer plate looks like, an almost exact copy of the inner one without the holes for the motors. Now with those two plates made, it’s really just time to assemble the rest of the drivetrain, which is by far the most time consuming. Now, for some info on what to use to attach the two plates together, I would highly recommend churro. To attach the two halves of the drivetrain, use a custom u-brace. We found that this is a perfect way to stiffen up the drivetrain and required very little maintenance over the season. You can also use peanut extrusion or kit channel, which alternatively works just as well.

Note that when using a custom drivetrain as I did, you can cut out material from your drivetrain plates. This process is called **pocketing**. While not a vital step, pocketing helps you save weight. However, be careful not to remove too much material; if you do, it becomes less sturdy.

This is what I like to call a drive pod. This particular one is the left side, but to make the right side all you have to do is create an offset plane, select the mirror tool, then hit mirror.



After mirroring the drive pod to make your opposite side, you just have to connect those two halves together and you’re done with the drivetrain. This is what mine looked like after completing the CAD and rendering it. (Rendering is not a vital step; it just makes a 3D model of your design.)

Power Transmission

Direct Drive

Direct drive refers to powering a mechanical part (most commonly a drive wheel) directly from the motor axis. Many new and inexperienced teams will use this method to power their drivetrain as it is the simplest way to do so. However, there are significant drawbacks to this method, and it is not recommended unless absolutely necessary.

First, direct drive puts unnecessary load on the drive motor. This is because shock loads can destroy the gearbox, even planetary gearboxes. Gearboxes are able to withstand load along the axis of rotation, such as what occurs when the wheel changes direction. This is a normal situation of load. However, in direct drive, the gearbox shaft can be exposed to other shock loads outside of the normal axis. This happens when the wheel comes into contact with another robot or the field wall, which honestly happens more than you'd think. This can bend the motor shaft or permanently damage the gearbox.

Second, direct drive limits the gear ratio to whatever ratio the motor gearbox is at. One advantage of all the other three transmission systems is the ability to gear up or down, based on a team's needs. Direct drive cannot do so, and if your drivetrain uses 40:1 gearboxes, there is no way to reduce this ratio to a faster 20:1, for example.

Third, motor shafts are not built to carry large amounts of downward load. This can lead to bending of the motor shaft, since in direct drive the shaft is only supported by one side, the gearbox. Generally, a principle among all engineers is to support the shaft on both sides, which isn't possible in this case. This leads to cantilevering the shaft, something that should be avoided in general. Thus, with a heavy robot the motor shafts can easily be bent due to a lack of support.

For these reasons, **we advise that you do not pursue direct drive**.

Disclaimer: direct drive is acceptable for mechanisms such as an intake, linear slides, etc. - it is discouraged for mechanisms under high load, especially drivetrains and arms.

Advantages

- Saves space
- Easy to build; most simple form of transmission

Disadvantages

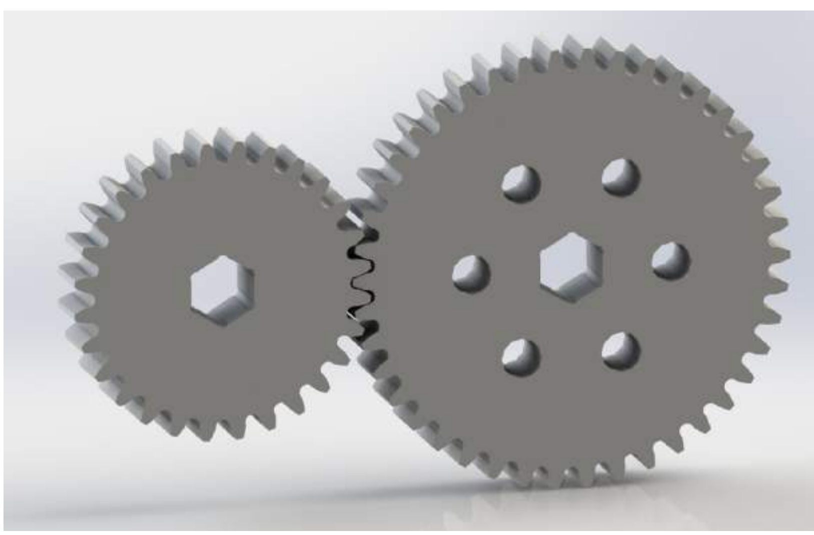
- Prone to shock loads which destroy the gearbox
- Limited gear ratio to the motor itself
- Wears the gearbox of the motor faster
- Can bend the motor shaft

Gears

Gears are a less common transmission option than chain, but are still very viable for most use cases. Gears are equally as reliable as chain, but can't be used for transferring power over long distances. Many teams dislike gears when using kit-based channels because the hole pattern limits which combinations of gears can be used. Consequently, it may be difficult to achieve a desired specific gear ratio. However, with extrusion systems, it's very easy to use different ratios, since the gears can be moved around in extrusion.
Gears are made in different materials, with the most common being 7075 aluminum. However, we advise you to stay away from TETRIX aluminum gears as they wear down very easily. Some REV gears are made out of Delrin, a self-lubricating plastic. It is a durable material, but keep in mind that it is very possible to strip the bore using a plastic gear. Thus, we advise using the REV Hex Hub Strengthener to avoid stripping the bore on Delrin gears.

Be sure to never mesh gears that are not of the same material or same diametral pitch.

Tips
When meshing gears, it is important that the gears are not too loose nor too tight. If the gears are too loose, the teeth will easily wear out, decreasing its longevity. If the gears are too tight, however, they will have too much friction and possibly grind or bind up. The ideal way to mesh gears are to make sure the teeth interlock and just touch the base of the gear. **If possible, it's best to avoid meshing gears with a clamping motor mount -- due to the sensitivity of the mesh, even the slightest movement of the motor inside the clamping mount can cause the gears to slip or damage each other.**



Correct gear mesh

As with sprockets, it is important to line up the gears so that they do not accidentally slip. Especially when using extrusion, it is possible that the gear may not be parallel to the extrusion, as the two supporting ends may not be perfectly in line with one another. It is imperative that the gear be lined up as straight as possible to prevent damage or gear binding.

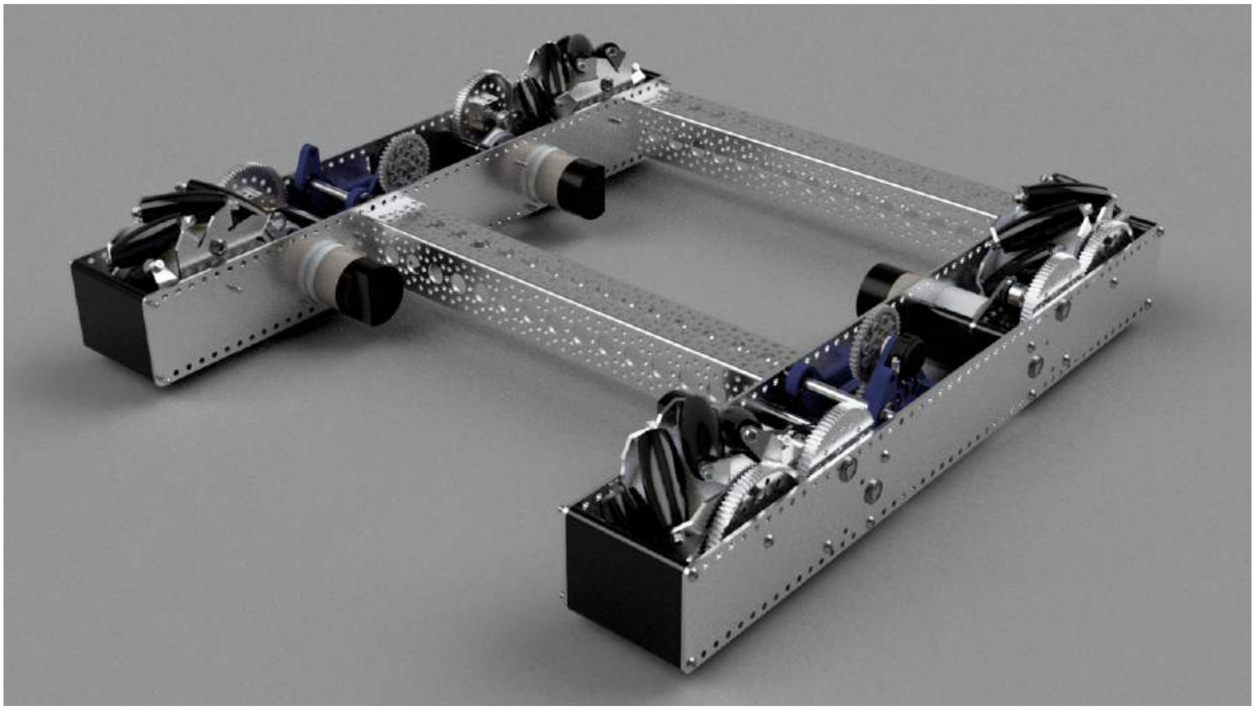
Another tip is to use white lithium grease between the gears to reduce friction and possible binding.

Advantages

- **Gears are a solid and proven power transmission method.** Early examples of gears date back to the 4th century BC, so you're using technology with millennia of development behind it. When it comes to gears, there's not much that we haven't figured out.
- **Gears are simple to use with both channel and extrusion.** On channel, your gears are already spaced correctly - you just need to choose the right pair of gears. Extrusion gives you even more flexibility - just slide your gears into mesh, and you can have whatever ratio you want.
- **Gears can give you big reductions in small areas:** Depending on the gear combination, one can achieve big ratios in reduction in very small spaces (for example, a 10 tooth gear and a 100 tooth gear will take much less space than a 10 tooth sprocket and a 100 tooth sprocket).
- **Gears require no tensioning. Once the spacing is correct, the gears will operate quickly.** Unlike chain or belt, there is nothing further transferring the power, which cuts out the need to properly tension chain or belt. This of course has the drawback of not being able to transfer power far distance.

Disadvantages

- **Sometimes, the ratio you want might not be easy to build.** Channel spacing limits gear ratios, but this can be circumvented with compound ratios and a bit of creativity.
- **Long distance power transfer is impractical with gears.** If you need to transfer power long distances, gear combinations can become complicated very quickly, so belt/chain is preferrable.
- **Meshing gears can be tricky.** It's only made worse by the sensitivity of a gear mesh. However, channels do solve this problem, providing pre-spaced holes to easily mesh your gears. Do keep in mind that gear mesh may not be perfect, even with channel.
- **Gears usually wear faster than sprockets** if there is too much friction between the gears**.** Teams can use white lithium grease or similar lubricant to help remedy this problem.



11115 Gluten Free gear-based drivetrain

13075 Coram Deo Robotics, Rover Ruckus gear-based drivetrain

Roller Chain

When your shafts aren't right next to each other, roller chain and sprockets will allow you to transmit power securely between your shafts. Roller chain is made up of a series of links joined by pins. Each link can rotate around its pins, creating a dynamic loop that can conform to any shape. The pins in the chain engage the gaps between teeth on each sprocket. If you've ridden a bike, chances are that you've already seen roller chain - the chain on your FTC robot is similar, but it's probably a different **pitch** (different size). Chains most commonly used in FTC are #25 or 8mm pitch.

When using chain, often there is a master link. This is a special type of link that has a removable end capsule in order to shorten the chain. However, as it is removable, it is not a very reliable chain link and can loosen and fall off under prolonged usage. There have been teams who have had master chain links fail during competition, costing them a match in the elimination rounds. A chain breaker eliminates the need for master links because it can break and join chain at any point. Chain breakers do so by pushing out the pins in the link and re-inserting them back. **It is highly recommended that teams purchase a chain breaker (we recommend the DarkSoul chain breaker) instead of using master links, which are prone to failure.**

The removable master link is shown on the very right.

Chain should, at the very least, have 90° of contact with the sprocket or pulley. The best practice is to have 180° or more of contact, as it is very unlikely to fall off with proper tensioning. Chain skipping, especially on drivetrains or arms, is very possible without proper chain wrap or tensioning. When tensioning chain, be sure to not undertension *or* overtension chain. Undertensioning chain can result in the chain falling off the sprocket or chain skipping, where the chain can skip along the sprocket. Overtensioning the chain often results in the motor burning out, or less seriously, a loss of efficiency. Push along the chain run, and if the chain moves slightly without significant resistance, chances are you've done it correctly. If it's too tight, then the chain will barely move under a gentle press.

Advantages

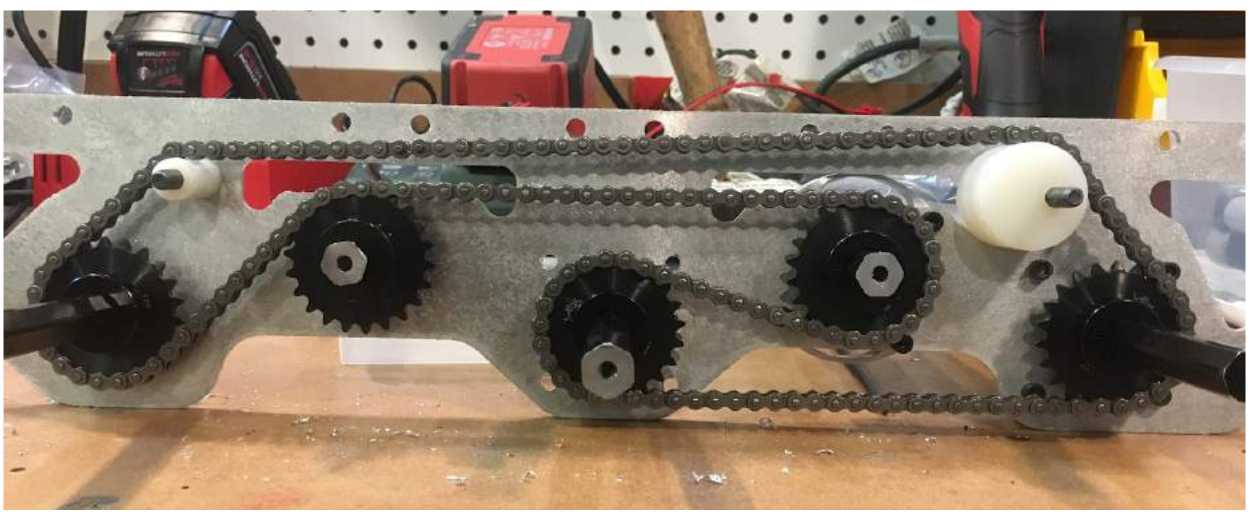
- **Chain can take a beating.** No matter what your application is, metal chain is usually up for the challenge. #25 Chain can hold up to 930lbs before breaking, and there's nothing you'll do in FTC that will exert that force. *(If your chain does break, it's most likely due to a faulty Master Link or sprockets that are not correctly aligned.)*

- **Chain can be however long or short as you wish.** If your ratio changes or your shafts move, it's easy to adapt your chain run - just break the chain and put it back together at its new length. You can often do this without even removing the chain from your robot.
- **Chain can be pretty precise.** When properly tensioned, roller chain doesn't have very much slop. However, you really need to get your chain tension right to reduce slop, and you'll probably want an adjustable tensioner for when the chain stretches. This can be done easily if using extrusion systems, as the sprocket can be adjusted for tension.

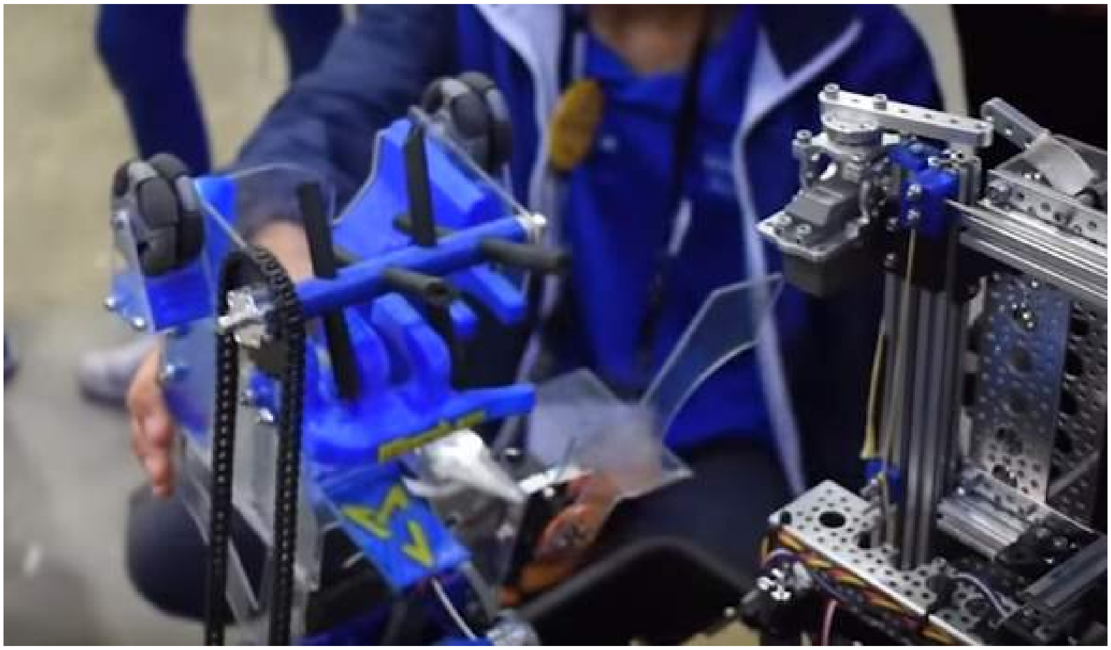
Disadvantages

- **Chain stretches over time.** As it's used, the connections between the links and rollers can stretch a bit. While it doesn't look like much, this stretching can introduce lots of slop into your chain run and even derail it in some cases. You'll most likely need an adjustable tensioner to keep your tension over time - some teams have used spring-loaded dynamic tensioners to automatically compensate for any changes.
- **Sprockets are really big.** If you want a really high reduction using chain, you'll pay for it in the space that it takes up. Sprocket teeth are much larger than gear or pulley teeth, so your reductions are going to be much larger.

Bottom Line: Chain is a great general-purpose power transmission method, but it can fail easily if you don't take care of it, and large reductions can become unwieldy.



7244 OUT of the BOX Robotics, Relic Recovery



9794 Wizards.exe, Rover Ruckus, plastic chain on intake

724 Rednek Robotics Wun, Winning Alliance 1st Pick (Houston), Relic Recovery

8103 Null Robotics, Rover Ruckus

Timing Belt

When you think of a belt, you're probably thinking of a very important men's fashion accessory. However, there's another type of belt, and it's way more relevant to robotics - the Timing Belt. If you've ever tinkered with the insides of a car before, you probably recognize timing belts as an important component designed to keep everything under the hood in sync. While a timing belt may complete a similar objective to chain, its characteristics and strengths are very different. Timing belts use a series of small, wide teeth to engage a Pulley with a number of matching grooves. They earn their name because they can be very precise, transmitting power with absolutely no slop and ensuring a snug connection between shafts. They're lighter and more compact than chains, but they lack the customizability of their bulkier brother - belts come in a closed loop of predetermined length, and there's no changing that length on the fly. Like chain, belt is identified by its **pitch** - common pitches found on FTC robots include HTD 5mm, HTD 3mm, and XL 0.2".

When using timing belts, correct tension is very important. There are two main ways to get your tension right. The first is easy - goBILDA and Actobotics already have belts integrated into their hole patterns. You can buy correctly sized belt directly from each vendor, and your tension will be perfect as soon as the belt is installed. As your designs gain complexity, so will your belt runs - maybe there are more than 2 pulleys, and maybe your pulleys are all different sizes. To compensate for this, the second way to ensure tension is to use a dynamic tensioner, similar to those found in complex chain runs. To design for these tensioners, we recommend planning more complex belt runs in CAD before building them in real life.

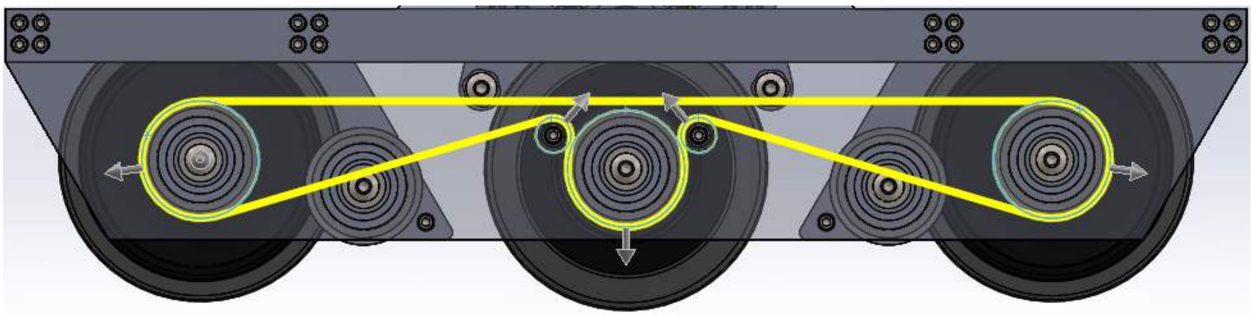
Advantages:

- **Pulleys can be made at home.** Pulleys can be 3D printed for most situations, allowing you to cut costs and create unique tooth counts easily.
- **Belts are very strong.** They're reinforced with fiberglass cords that are incredibly hard to break, giving belts immense strength. *(If you break a belt, ~~give yourself a round of applause~~ it's most likely because it was out of alignment or tensioned far too tightly.)*
- **When tensioned correctly, there is *absolutely* no slop.** Engines use timing belt for a reason - because it's the best possible solution for them to perfectly synchronize their shafts. There's nothing that matches the rotational accuracy of a properly tensioned belt.
- **Belts are efficient and quiet.** Compared to the loud shredding sound of a chain run, belt runs are dead silent, and they're more efficient than chains (although this makes zero practical impact in the robotics use case.

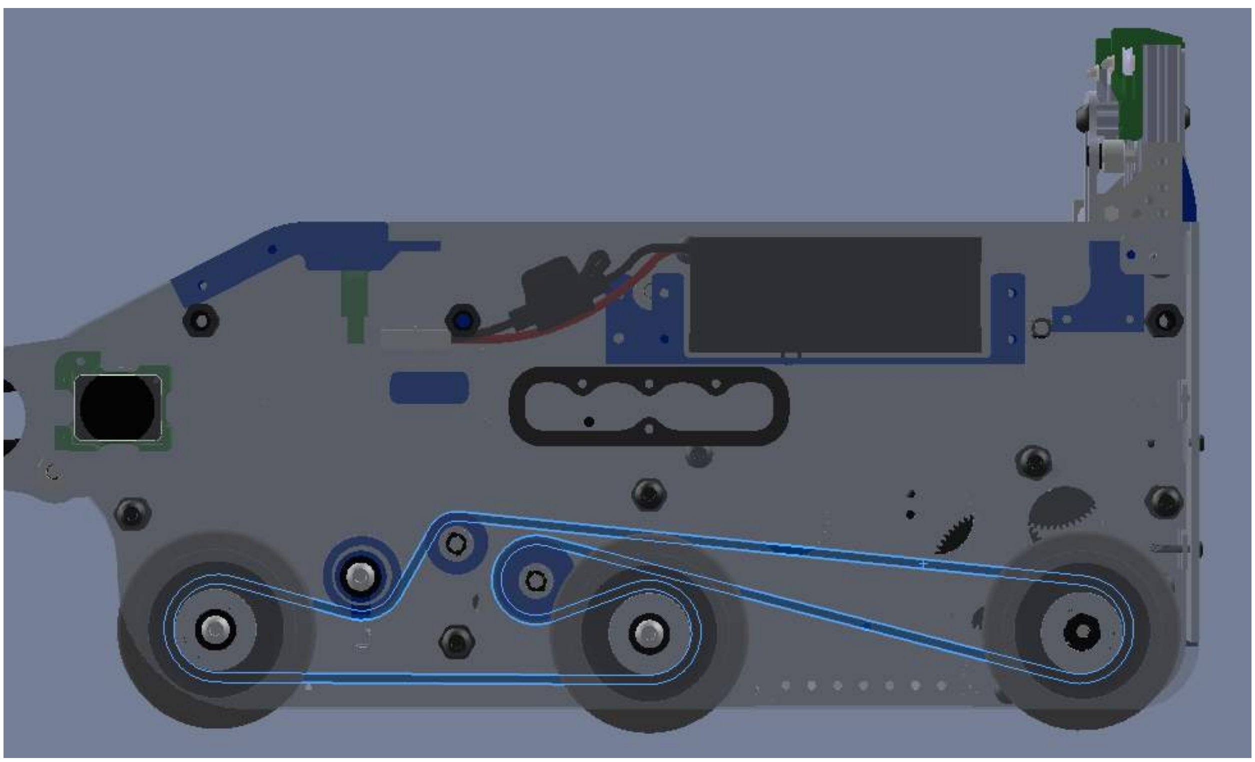
Disadvantages

- **Belts aren't customizable.** You buy a belt of a specific length and you're stuck with that length until you buy another one. This isn't too bad if you're planning out your robot properly, but chain will work better for prototypes where the chain length will be changing often.

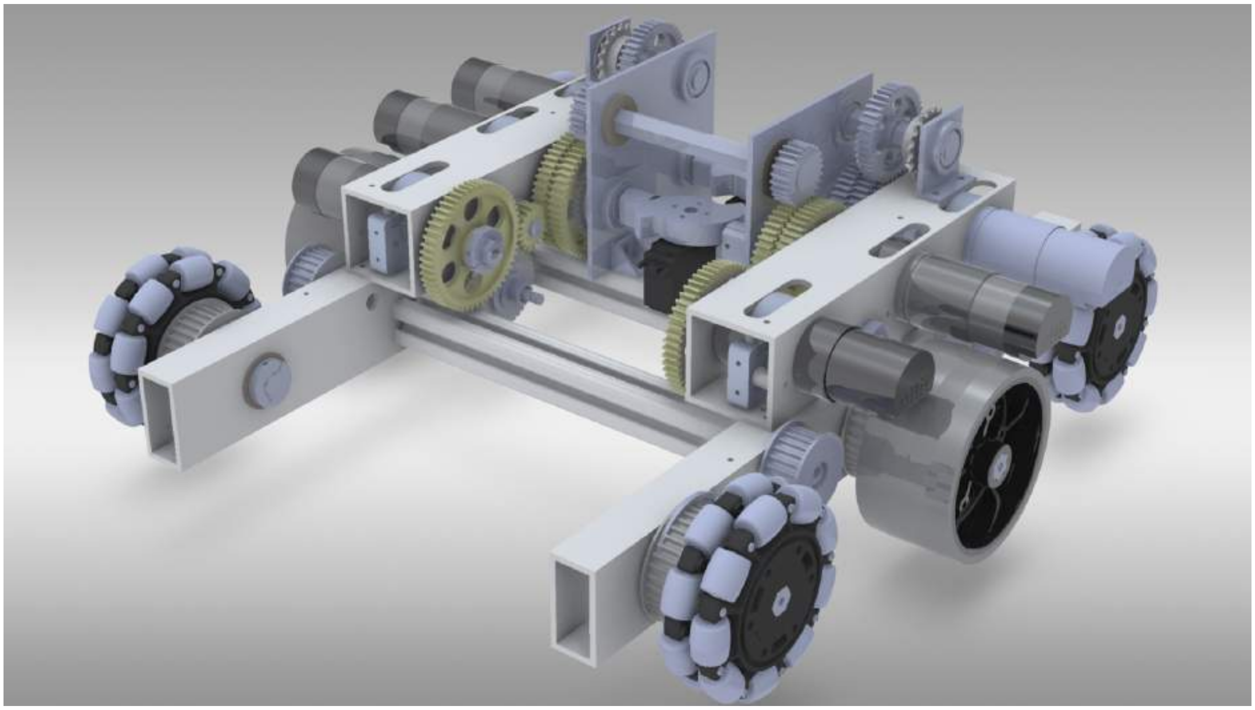
- **Belts can be wider than alternatives (especially chain).** This probably won't have much of an impact, but belt can often be wider than other power transmission methods, so it may not always fit.
- **Belts can be expensive (but you'll save money with pulleys).** While you can buy chain 10 feet at a time, you'll most likely be buying each belt brand new. While this can get expensive, you'll be saving money every time that you print a pulley instead of buying a sprocket - it's a trade-off.



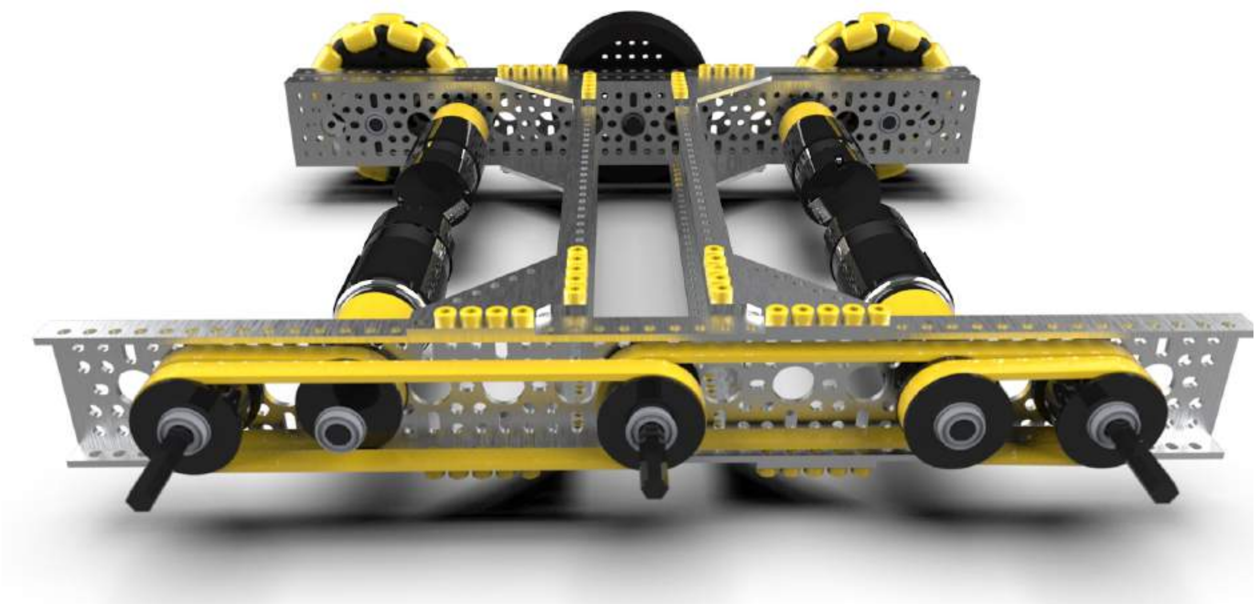
12897 Newton's Law of Mass, Rover Ruckus prototype



8417 Lectric Legends, Rover Ruckus



5975 Cybots, Velocity Vortex



Ethan Doak, goBILDA

Linear Motion Guide

Linear motion is one of the most important components of a successful robot. In most games, teams are required to reach into an area that the drivetrain cannot access in order to pick up or deposit game elements. For example, in 2017-2018, Relic Recovery, teams needed to extend an arm of some sort to grab the Relic that was placed in the corner of the playing field. This area of the game field was nearly impossible for a robot to drive to, so a linear extension was necessary.

Access of > 18” is necessary for nearly all games; >24” is preferred. In some games, an extension of 36" or more may be needed.

*Note - it is possible to achieve extension with an arm, but since this guide is geared toward newer teams, linear extensions should be prioritized over arms. Refer to the Arm Guide for more information.

Extrusion Slides

Extrusion slides are made up of a stack of extrusions that extend by sliding along each other. There are two ways to make this happen: bushings or V-wheels. Bushing slides connect the two slides with two self-lubricating plastic pieces that slide smoothly along the slots in the extrusion. V-wheel slides have V-shaped groove bearings on both sides of the extrusion that bite intro grooves on the extrusion, allowing the stages to slide smoothly. REV, Actobotics, and goBILDA all sell extrusion slide kits that integrate nicely with existing FTC kit parts. Beyond this, MakerBeam sells components for a V-wheel extrusion slide, and Misumi offers a few different sizes of bushing-based extrusion slides. REV furthermore sells a 8020 v-groove bearing slide kit for FRC, so it is not recommended for FTC use cases. **We recommend that newer teams stick to slide kits designed for FTC - they're easier to integrate into a robot and often cheaper than a DIY alternative.**

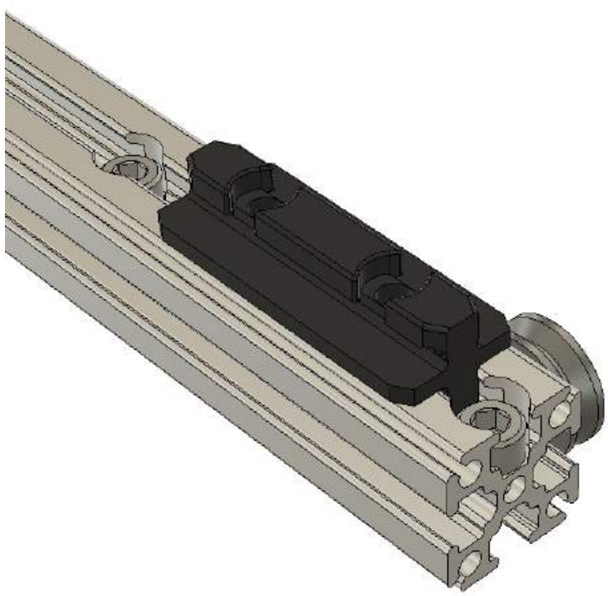
Listed below are the recommended extrusion slide kits:

- REV Robotics 15mm Linear Motion Kit
    - http://www.revrobotics.com/rev-45-1507/
    - The REV 15mm Linear Motion Kit is based off of the 15mm extrusion system. **This extrusion kit does not perform spectacularly without modification.** This has been partially remedied by REV as they have developed a second iteration of their slide kit, which has better tolerances on the Delrin sliders. Still, you'll see a lot of competitive teams use this kit with multiple modifications, such as adding lots of lubricant and mounting the sliders differently. Teams have also 3D printed their own sliders, though this is not a great idea for teams without much 3D printing experience. One of the biggest issues with the stock REV kit is the tendency of the slides to bind. Additionally, since the only thing attaching one extrusion to another is the plastic slider, the REV slides are not particularly sturdy, and require crossbeams to keep alignment. Overall, this kit is lightweight, simple, and cheap. It can be a good start for teams using REV and needing a linear extension, and is generally usable out of the box. However, it is not very smooth and only achieves its maximum potential when modified and tweaked.

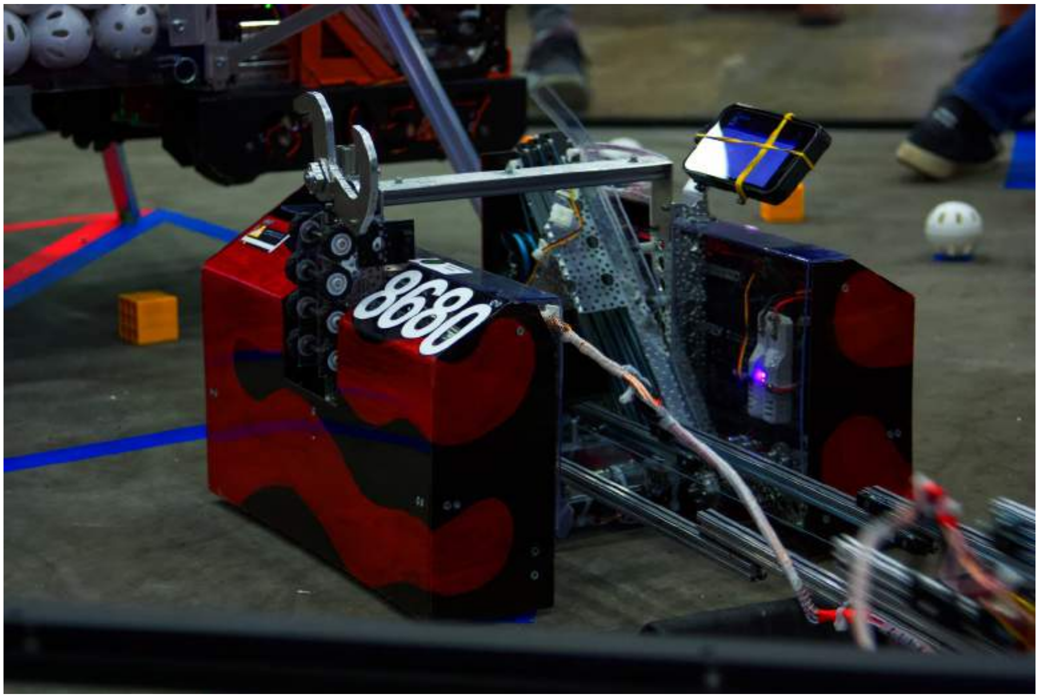
- Actobotics X-rail Slide Kit
    - https://www.servocity.com/cascading-x-rail-slide-kit
    - The X-rail Slide Kit works well out of the box. However, the main caveat is that the kit has a **very low maximum load (2lb. at maximum extension).** Teams will have to keep their designs on this kit particularly lightweight. This slide uses elastic retraction through the use of surgical tubing, which means that instead of having both an extend and return string, there is a retraction force applied at all times. This helps simplify tensioning and spooling, however, limits how fast the slide can be run. Additionally, the plastic end caps have a reputation of breaking regularly because it endures shock load every time the slide extends to

maximum. 3D printed alternatives may be more sturdy than the stock end caps. **It is highly recommended that teams add an additional set of v-groove bearings at the end of each piece of extrusion to give each stage an additional point of support. This will increase load capacity and possibly smoothness.**

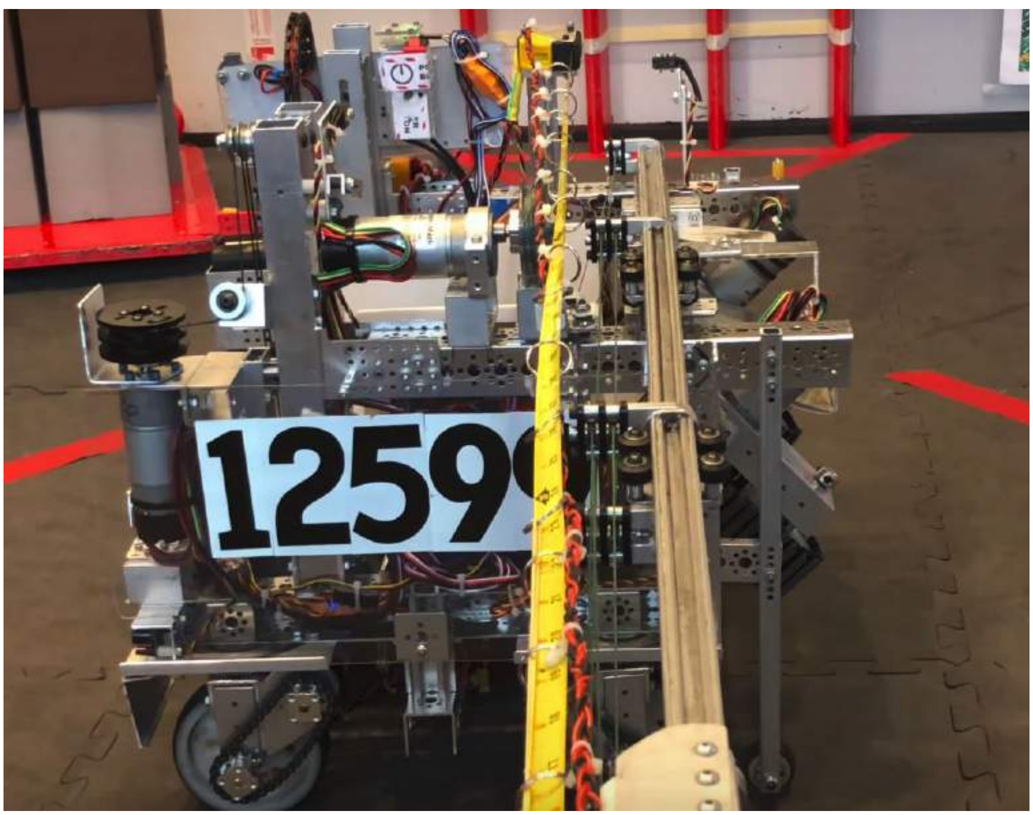
- goBILDA goRAIL
  - https://www.gobilda.com/linear-motion-guides/
  - **Note that there is no preconfigured kit for goRAIL, instead, teams must configure their own kits.** If teams are confused, it is recommended that they contact goBILDA for help in setting up a goRAIL based linear motion system.



11115 Gluten Free, Rover Ruckus, custom REV slide kit



8680 Kraken-Pinion, Rover Ruckus, REV



12599 Overcharged, Relic Recovery, Actobotics X-Rail

Drawer Slides

You've *definitely* used a drawer slide before - at least two of them are mounted to almost any drawer that you've opened. Teams use these drawer slides for linear motion, often stacking them using 3d printed parts to achieve plenty of extension. These slides are available from a number of different vendors, and come in many varieties, so choosing the right slide can seem overwhelming. Steel drawer slides are common, but can be finicky with mounting, as they aren't made so that another slide may be attached easily. Aluminum drawer slides, such as the MiSUMI slides or the Hafele slides listed below, are generally the best option for teams. The Hafele slides are less than $3.50 for a pair, but the MiSUMI rails offer much smoother motion and have a better ball bearing system. Igus is a brand known and trusted by robotics teams and manufacturers, but their parts are certainly not cheap, and can reach into the $100+ range. **If linear slide kits aren't for you, we recommend the MiSUMI slides for newer teams**, especially if you are using REV, as MiSUMIi uses the M3 standard. ***Drawer slides should be mounted oriented vertically, like in an actual drawer. They can be mounted horizontally, but this is not recommended as they will sag much more.***
Listed below are the recommended drawer slides.

- ### Steel-rolled cabinet drawer slides
  - Available from your local hardware store, steel slides aren't a bad option for FTC teams, as they are heavy-duty enough for most use cases. However, these kinds of slides are much heavier than other aluminum slide options. Furthermore, these slides are not designed to have bearings or a second slide attached to them, because they only contain mounting for a standard drawer. Thus, these slides require drilling holes in order to mount the necessary parts for linear extension. While the drawer slides are usable (and have been used by top teams in the past), there are better options for teams looking for linear extensions.

- ### MiSUMI Telescopic Slide Rails
  - https://us.misumi-ec.com/vona2/detail/110300072130/?HissuCode=SAR240
  - The MiSUMI slide rails are arguably the number one option for linear extension in FTC. Used by many top-tier teams, these rails are sturdy and very reliable, ridiculously smooth due to the ball bearing system, and have almost no flex. **MiSUMI slides are able to withstand a significant amount of load with little flex.** They are also low-profile, and have a M3 mounting pattern, making it easy to attach to REV extrusion. However, MiSUMI slides have a higher price point, and it is often difficult to attach one slide to the next. An easy solution is to attach the end of one slide piece to REV extrusion, and do the same with the next slide. Then attach the REV pulley bearing on the top of the extrusion piece for the string to run through. Some teams have 3D printed an insert that goes between each slide instead of using the 15mm extrusion piece, as it takes up quite a bit of

space. In order to attach the slides to anything, teams will need to purchase **countersunk M3 screws** from McMaster-Carr. For attaching to REV extrusion, buy 6mm M3 screws with the M3 nut (***not* locknut**) to insert inside the extrusion. As a tip, try to protect chips or sawdust from falling into the slides, as the sliding will have a noticeable difference. Generally, the MiSUMI slides, even with their disadvantages, are one of the best price-performance options for FTC extensions.

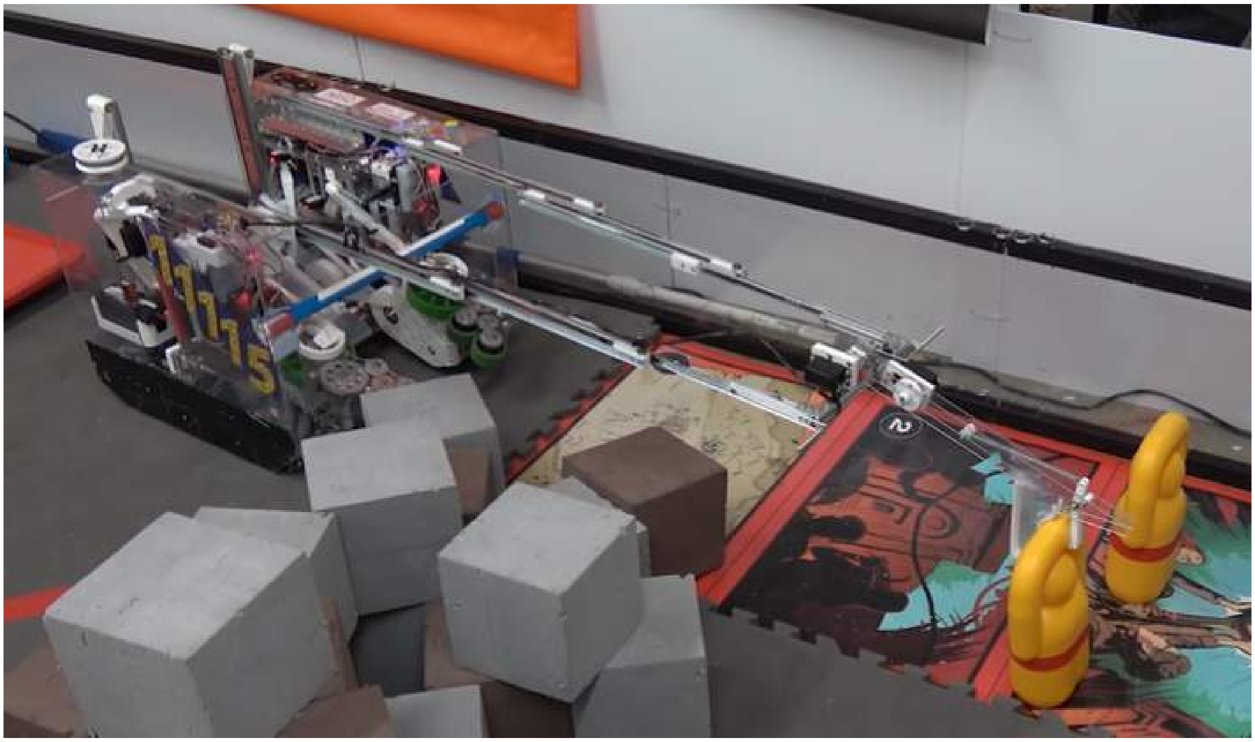
- Rail length: 400 mm, part number SAR 240

- Hafele Drawer Slides
  - https://www.cabinetparts.com/p/hafele-drawer-slides-side-mount-drawer-slides-HAF42058373
  - Hafele drawer slides are a low-cost, low-weight option for teams not wanting to invest in MiSUMI slides. Hafele slides are less smooth than MiSUMI, which is to be expected given their low price point. They are unable to sustain a large amount of load, making it good for low/medium-load use cases, such as to pick up and deposit the relic in Relic Recovery. Hafele slides are a great option for teams looking for a cheap alternative to the other more expensive options at the cost of robustness.

- Igus
  - Igus slides used to be a popular option among top teams for linear extension back in the earlier days of FTC. However, with the arrival of MiSUMI slides, they have decreased in popularity due to its prohibitive price point. While Igus slides are absolutely great in terms of performance, the MiSUMI slides offer a better value for teams in general (great performance - high price vs. good/great performance - good price). However, if your team is looking for a high-load or special use case, by all means explore this option.

7236 Recharged Green, Rover Ruckus, Misumi SAR-3*
*Incorrect mounting orientation; SARs should be mounted vertically, not horizontally

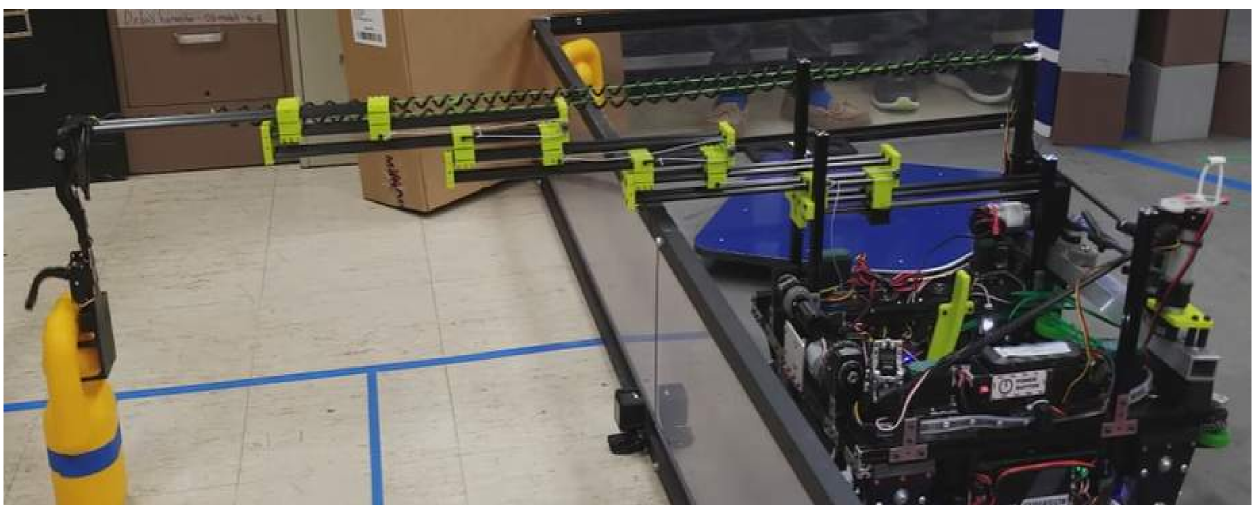
6929 Data Force, Rover Ruckus, Misumi SAR-3



11115 Gluten Free, Relic Recovery, Hafele slides

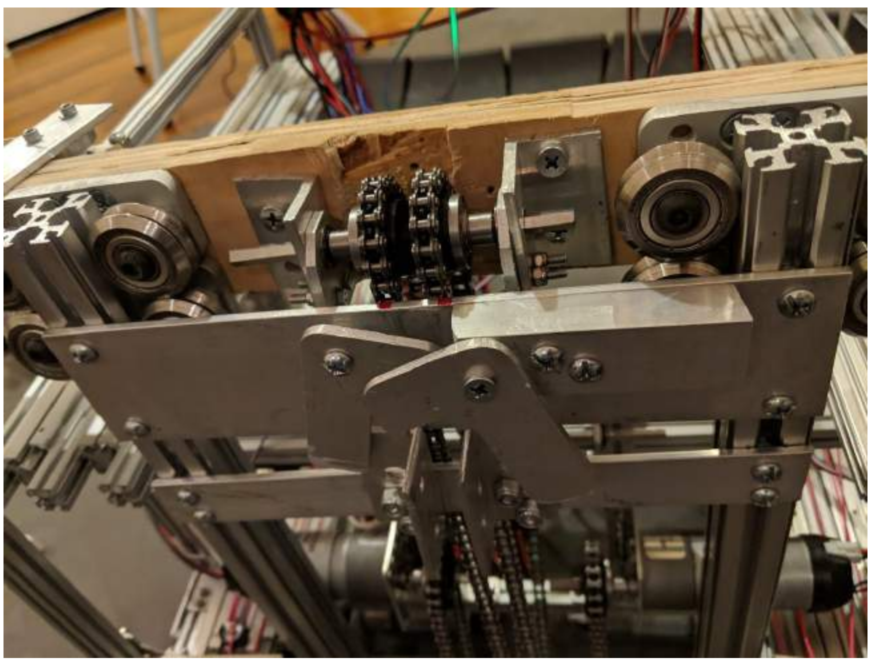
Custom Options

Some teams have opted to build custom solutions for linear motion. Many of these teams borrow concepts from FRC teams, modeling their elevators off of the tall systems found on the larger robots. There's a reason that many competitive FRC teams build the same type of elevator - at their scale, the box tube elevator has proven to be the most efficient way to get game pieces off the ground at blisteringly fast speeds. When built correctly, an elevator of this type can withstand hundreds of pounds of load on any axis while barely weighing anything. However, these solutions don't scale very well to FTC robots, and existing off-the-shelf options already fill the linear motion needs of almost every FTC team. Beyond that, custom elevators require tons of work in CAD, hours upon hours of print time, and may need multiple iterations before they work correctly. Due to their complexity and how challenging they are to design, we do not recommended that less experienced teams try to build custom elevators. Additionally some teams will opt to use custom carbon fiber rod based slides. These slides are primarily designed to withstand loads perpendicular to the direction of the slide, so they are a light strong option for slides that need to extend far and quickly. However, note that lightweight drawer slides (MiSUMI aluminum) can offer similar performance at a fraction of the complexity.



5975 Cybots, Relic Recovery, custom carbon fiber and linear bearing slide

7236 Recharged Green, Rover Ruckus, custom belt driven box tube linear elevator



13075 Coram Deo Academy Robotics, Rover Ruckus, custom 8020 chain driven hang

Lead Screws

Lead screws use a threaded rod to create high-torque linear motion. They are useful for hanging mechanisms or other high load situations, but most mechanisms would benefit from a much higher speed extension because lead screws are designed for slow and precise movement. If your robot needs to do a pull-up or hang, such as in Rover Ruckus, a lead screw might be the right choice. Another use case for lead screws is to change the angle of an arm platform, though this is more common in FRC. However, for most linear motion use cases, you'll want to stay away from them.

11115 Gluten Free, Rover Ruckus, Actobotics leadscrew used for hang

Rack and Pinion

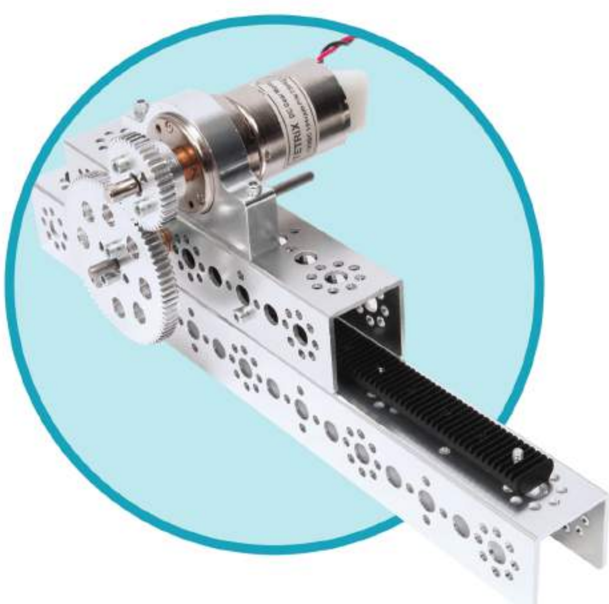
Rack and pinion refers to a circular gear (the *pinion* gear) attached to a shaft and meshed to a rack, which is a toothed linear gear. When the pinion gear is driven, it will drive the pinion gear upwards or downwards, depending on how the rack and pinion is mounted. Generally, rack and pinion is a good light-use option for FTC teams in terms of linear extension. However, there are some disadvantages to rack and pinion compared to the othe roptions of linear extension. Therefore, rack and pinion is generally not recommended for teams.

Advantages

- Easy way to extend upwards.

Disadvantages

- Rack and pinion generally is only used in one stage, because it would require multiple motors to drive multiple stages. Therefore, rack and pinion cannot extend as far as most linear extension mechanisms.
- Rack and pinion is slower than the other options.
- Rack and pinion cannot sustain heavy load, or else the mesh will fail. Therefore, **it is not advisable to use rack and pinion as a hanging method.** Instead, use a lead screw.



The Tetrix rack and pinion has a tendency to fail under medium to high load.

Rigging

Rigging refers to the way that string, belt, or chain is set up to extend and retract a linear extension.

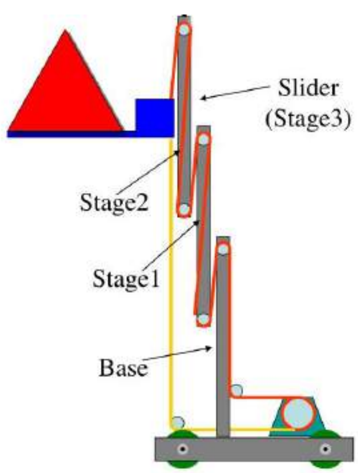
Example rigging setup from team 7236 Recharged Green. This setup utilizes a continuous stringing setup for the first stage and a cascaded belt-driven second stage

Continuous rigging

Generally recommended
Continuous rigging entails rigging one long **extension string**, originating from a motor-powered spool, to the top of the base stage, then to the bottom of the first stage, then to the top of the first stage, then to the bottom of the second stage, etc. A **retraction string**, originating from a second spool on the same axis as the extension spool, is then anchored to the top stage. When the motor rotates one direction, the extension spool reels in the extension string, so it becomes shorter. In doing so, the distance between the top of one stage and the bottom of the next stage decreases, causing the system to extend. Note that **the last stage always extends and retracts before the other stages (this can be either an advantage or a disadvantage, depending on the application).** Once the last stage hits its limit, the next to last stage extends outwards, and so on; the pattern repeats until every stage is fully extended. When the motor spins in the opposite direction, the retraction string is reeled in, pulling the top stage closer to its starting position until the system is back where it started. For the retraction string, it is often necessary to add an extra pulley near the back of the extension. This is because the retraction will only retract to the farthest point, which generally is the spool. However, the spool may not be mounted at the very back of the robot - thus an extra pulley is needed. Note that for this to work, the extension string should be wrapped around the spool **in the opposite direction** of the retraction string. Thus, if the extension is wrapped clockwise, the retraction must be wrapped counterclockwise.

Here are some additional considerations when rigging a continuous system.

- As a general rule, continuous spools can be powered by a system with a relatively low gear ratio.
- The extension string and retraction string **do not need to be separate strings, but it is much easier to tension the system if they are separate**.
- The extension spool and retraction spool should be the same diameter.



Cascade rigging

Generally recommended
Cascade rigging is a bit more complicated than continuous rigging. Much like continuous rigging, an extension string originating from a spool is rigged to the top of the base, running down to the bottom of the first stage. However, instead of being rigged to the top of the stage, the extension string is anchored to the bottom of the first stage. A second extension string, anchored to the top of the base, is rigged to the top of the first stage and anchored at the bottom of the second stage. The pattern continues until all stages have been rigged. Note that **the number of strings required to extend is equal to the number of stages in the system.** When the motor rotates one direction, the extension spool reels in the first string, decreasing the distance between the base and the bottom of the first stage. This pushes the second string forward, decreasing the distance between the top of the first stage and the bottom of the second stage, and so on. Note that unlike continuous rigging, **every stage moves at the same time.** The second stage moves 2 times as fast as the first stage relative to the base, the third 3 times as fast, and so on.

A cascaded system can be retracted in three ways: using continuous retraction, elastic retraction, or reverse-cascade retraction.

Retraction Options

Continuous retraction

Recommended

As the name suggests, continuous retraction utilizes the same retraction method as continuous rigging with one notable difference: if the variable N is the number of stages in the system, the diameter of the extension spool must be **N times smaller** than the retraction spool.

Advantages

- Requires less string
- More space efficient than cascade retraction

Disadvantages

- None

Elastic retraction

Not recommended

Instead of retracting using a retraction spool, one common way to retract is to attach a piece of elastic (commonly surgical tubing) to the last stage. The elastic applies a force on the last stage that is counteracted by the motor when extending. However, when retracting, the motor reels the last slide back in. While this is the retraction method recommended by many kit slide manuals, we do not recommend this method.

Advantages

- There is only one string to tension, instead of multiple, so tensioning is simpler.
- The elastic automatically tensions the extension string.

Disadvantages

- Since the elastic applies a force to the slide at all times, this force opposes the force applied by the motor when extending the slides. Thus, **elastic retraction decreases extension speed considerably.**
- The elastic does not apply a constant force at all times. It applies force proportional to the amount that the slide is extended, so retraction may not be smooth and controlled, like other rigging methods.

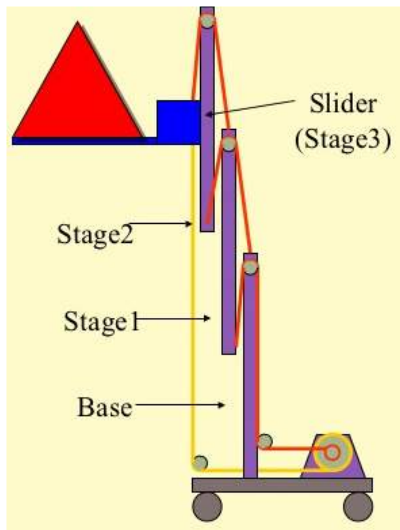
Cascade retraction

Not recommended

There is virtually no advantage to using cascade retraction on a traditional string-powered system. It's just more strings to tighten.

Additional Considerations

- If the system has only one stage, cascade rigging is **identical to continuous rigging**
- Unlike continuous rigging, each time a new stage is added to a cascaded system, the gear ratio required to maintain the same extension speed increases. For instance, if 2 stages are added to a 1 stage cascaded lift that is geared at a ratio of 3:1, the ratio must increase by a factor of $\frac{2+1}{1}$ to maintain the same speed, changing the ratio to 9:1.
- One disadvantage of cascade rigging is that each string must be kept tensioned. This is still the case with continuous rigging, but you have many more strings to keep track of, as tension must be maintained on all of them.



Belt-driven slides

One increasingly popular alternative to traditional string-based rigging is belt-driven slides. This can be done continuously or using cascade rigging.

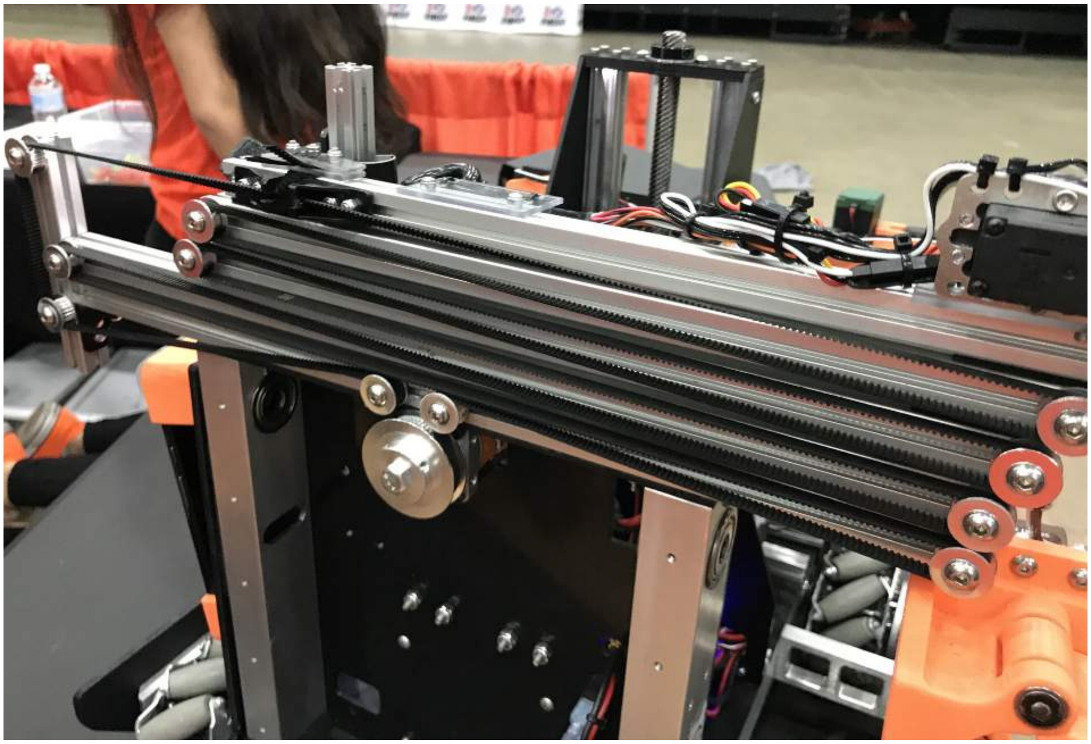
Advantages over string

Unlike string, belts used on slides never need to be tensioned. As will be discussed later in the linear motion section, in order for string-driven slides to remain efficient, string tension must be maintained. Naturally, string loosens over time, so you either need a mechanism that can provide extra tension (a spring) or manually tighten string, which can get a bit tedious (especially for cascade rigging). However, belts do not have this issue. They tend not to stretch over time, meaning complex external tensioners are rarely needed.

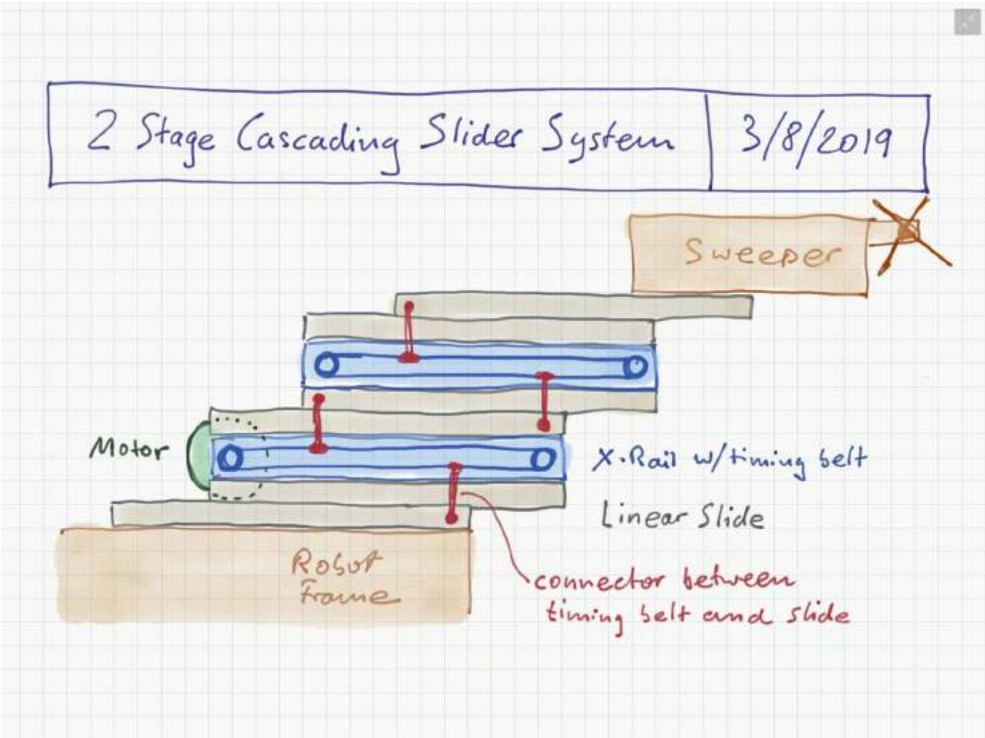
Disadvantages over string

The main disadvantage of belt-driven slides are the size restriction. Simply put, belt pulleys take up much more room than the 4mm thick, 12mm diameter pulley bearings REV sells. When using belt-driven slides, pulleys are at least double that thickness and have a considerably larger diameter, meaning each stage must be thicker. Other than pure packaging, this size restriction can limit the number of stages on a linear motion system, as each stage takes up more room than they would otherwise.

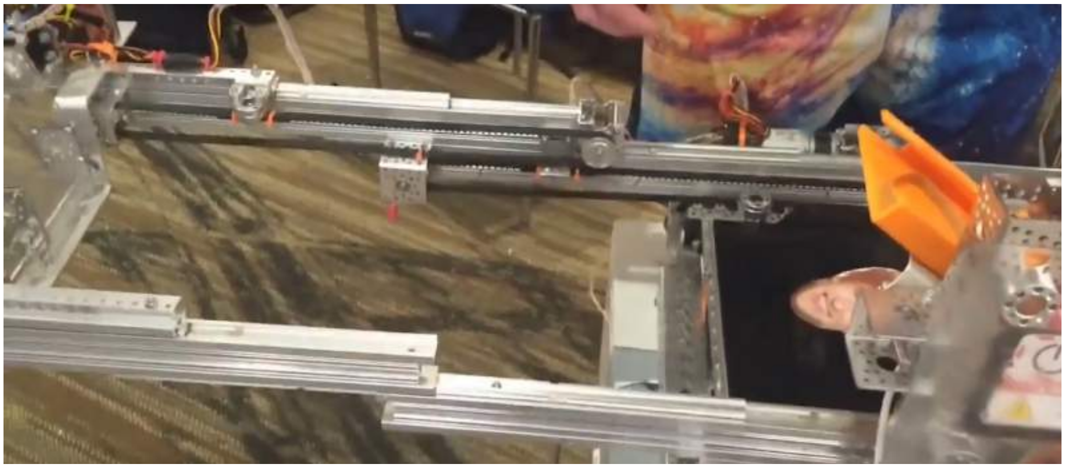
7236 Recharged Green, Rover Ruckus, **continuous** rig



11190 Mechadojos, Relic Recovery, **continuous** belt rig



5064 Aperture Science Rover Ruckus **cascade** belt-driven proof of concept drawing



5064 Aperture Science Rover Ruckus **cascade** belt-driven final version

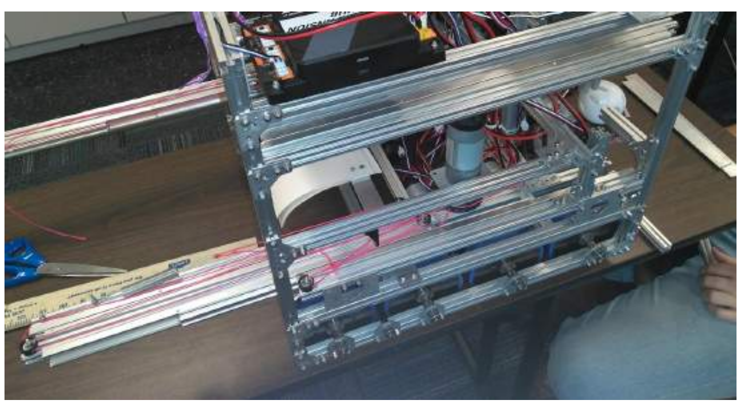
Belt or chain driven elevators

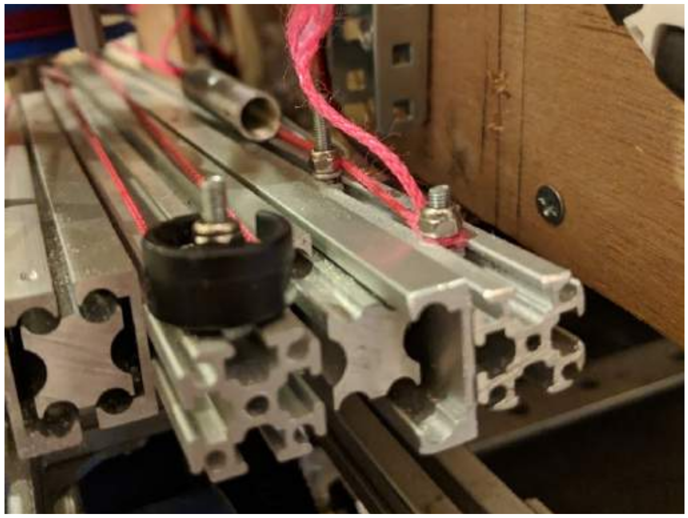
Almost ubiquitous in FRC, belt and chain elevators have existed seemingly forever. The basic premise is to have sprockets or pulleys mounted at the top and bottom of the elevator to the robot superstructure. Then, the belt or chain is hard bolted to the elevator near the bottom sprocket or pulley. When the chain or belt is driven, the elevator will move up and down. It is possible for elevators to have multi stage designs, but powering them will be more complex.

7236 Recharged Green, Rover Ruckus

Tensioners

Tensioning string is one of the most painstaking tasks for a builder in FTC. Ensuring that both sets of slides are tensioned evenly can be an arduous and annoying job. However, adding tensioners to your strings can help solve the uneven tension and ensure that both sides of slides run together. The most common type of tensioner is a spring that can be purchased at a hardware store. It generally will be placed at the end of the string run, near the part that extends farthest out from the robot's center. By doing so, the string will stretch out when the spool extends the arm, keeping tension so that the string does not detach from the pulleys in the linear slide extension. Another form of tensioner can be a spring-loaded pulley. Since the pulley is spring-loaded, it will take up the slack in the string. Alternatively, it is possible to mount a pulley on a piece of extrusion, and slide it so the string is taut. **However, keep in mind that spool size changes as string is added/removed from the spool.** If the spool radius increases, the speed of the extension will also increase, and torque, which opposes speed, will decrease. Consequently, tension will change as well. Therefore, a bit of slack is inherent in all linear extension designs using string. It is highly encouraged that teams have at least one spring tensioner per linear slide set.





13075 Coram Deo Robotics, Rover Ruckus, string tensioner

Picking the right spool size

Spools have a special property that isn't often discussed, but is extremely useful when creating linear slide systems. Just as the system's speed and torque can be changed by changing its gear ratio, **speed and torque can also be changed by changing the spool size**. Think about it: the motor rotates the spool at a constant angular speed. Thus, the translational speed (the speed of the slide) is proportional to the radius of the spool, and since torque is inversely proportional to speed, changing the spool size changes torque as well. This is important to recognize, as **changing spool size is often more convenient than changing gear ratio** to get the desired combination of speed and torque. To illustrate this, say you have a linear extension system with a 3.7:1 gear ratio. You then decide that a 5:1 gear ratio would provide a more desirable combination of speed and torque than your current 3.7:1 ratio. In many cases, instead of swapping gearboxes, it makes more sense to swap out spools to a smaller one. If your spool is currently 2 inches, your new size should be $\frac{2*3.7}{5}$ inches to achieve the same result.

Cable management

When extending outwards, wire management becomes increasingly important. Obviously, it is a necessity to use wires slightly longer than the extension length. However, it is not recommended that these wires are left unprotected, as they can get tangled or caught in the slides much more easily than with protection. In general, teams should ensure that wires never protrude outside the structural parts of the robot, because they can get caught on other robots or game pieces. This can be accomplished by cable ties or Velcro ties, or be using acrylic plate to keep wires inside. However, for linear extensions, other forms of cable management are needed. The two types of cable management recommended are cable carrier and retractable coil cord. Refer to the Electronics and Wiring section for more information.

Cable Carrier (aka drag chain)

Cable carrier, the standard wire management method within the manufacturing industry, is plastic chain with a hollow center. Cables are placed inside the chain, allowing the system to extend indefinitely. The links are somewhat stiff yet flexible, allowing cable chain to bend when the extension is retracted and straighten when extended.

Advantages
- Nearly impossible to get tangled
- Very sturdy

Disadvantages
- Takes up a lot of space
- New links must be added to the chain every time new stages are added to the slide

FTC 7236 Recharged Green, Rover Ruckus used a cable carrier on the left side of their horizontal slides and the right side of their vertical slides

Retractable Coil Cord

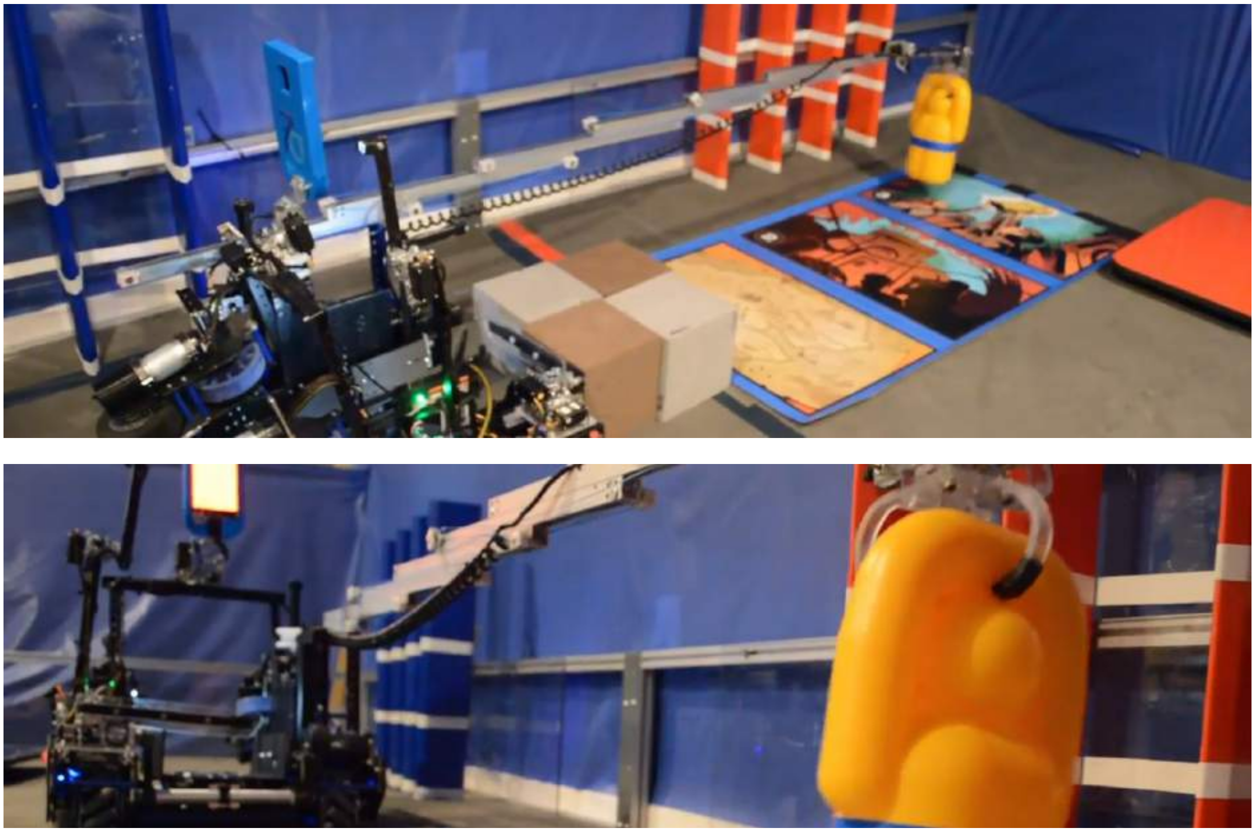
While not used within industry, coil cords are still very common (coil cord is a nearly ubiquitous staple of older telephones). Retractable coil cord is more flexible than cable carriers, stretching when extended.

Advantages:

- Very space-efficient
- Can usually extend to any length you would need in FTC (unlike cable carriers, no new links ever need to be added)

Disadvantages

- Can get tangled easily



FTC 10030 7 Sigma Robotics, Rover Ruckus used coil-cord on their horizontal extension mechanism

Arms

Arms are another way to achieve extension past the 18” x 18” dimension of the robot. However, due to their complexity and multiple things that can go wrong, newer teams are discouraged from trying a multi-axis arm. For one, arms require lots of torque - a standard 40:1 or even 60:1 gearbox will not be suitable in this application. A high-end gearbox such as the Andymark Neverest Sport or VersaPlanetary is nearly always required - and this is a significant investment in team budget. With arms, the higher the ratio the better. For example, many teams will run a 256:1 gearbox for their rotation motor. A budget option is the goBILDA planetary 188:1 which gives a high gear ratio at a relatively low price of $30. Note that these will have to be very well supported, more so than the VersaPlanetary or Sport options, and the arm should never be directly mounted to the driving motor. Instead, torque should be transferred via gear, chain, or belt. Another problem is backlash and controllability. This is because motors are unlike servos - they are built to provide lots of rotational movement and not necessarily very precise rotation. Thus, large arms are hard to control (with the addition of momentum adding load to the gearbox, it is hard to stop a three-foot arm that weighs five pounds quickly without breaking a gearbox). With many of these factors in mind, it is difficult and generally not worth the time for an inexperienced team.

The different types of arms in FTC include single arm, chain bar and multi-axis arms.

Advantages

- Single bar arms can be relatively simple to build and an easy way to make an extension.
- Arms can be useful in low-load applications; however, most mechanisms in FTC are not very light.

Disadvantages

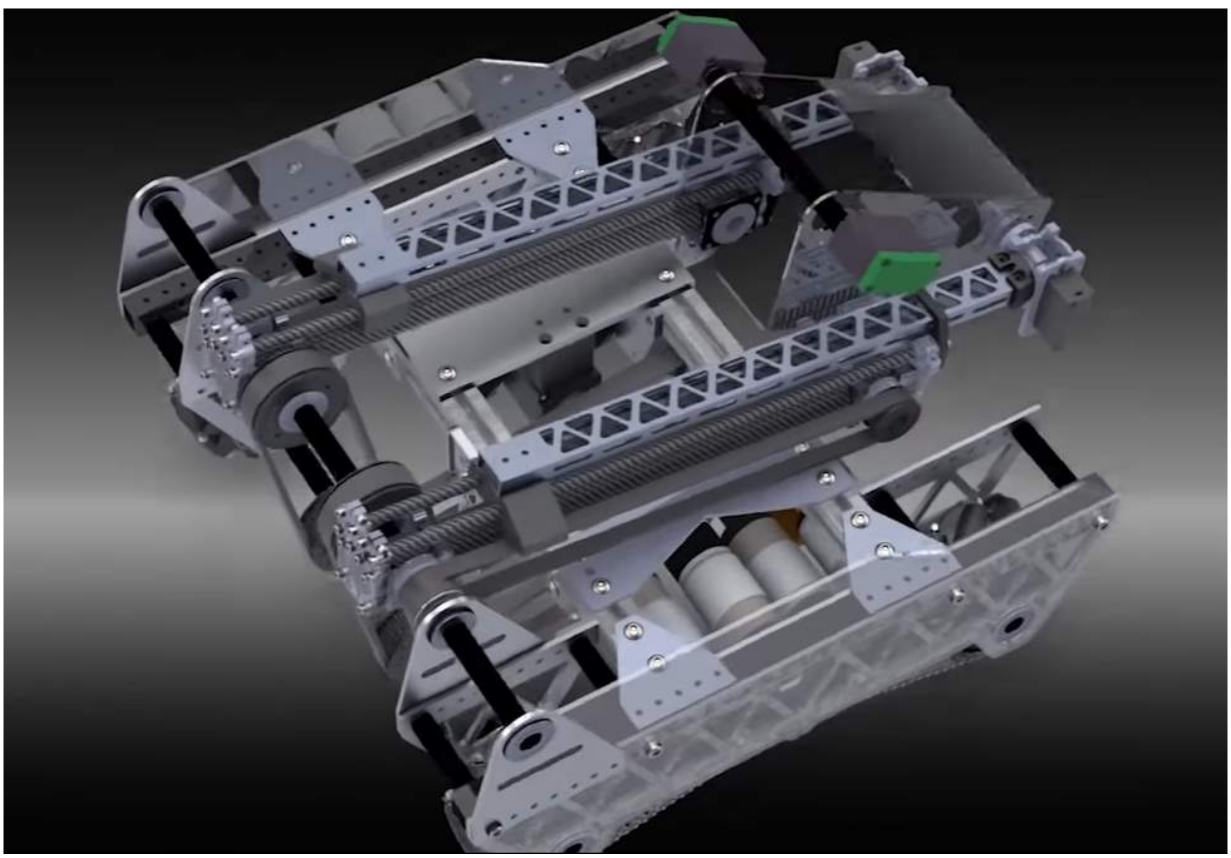
- Arms require an insane amount of torque, and in order to do so teams must purchase special gearboxes not found in kits, such as the Andymark Neverest Sport or VersaPlanetary.
- While single arm arms may be more simple, they cannot provide enough extension for most games.

Single Arm

The most simple type of arm in FTC, a single arm refers to an arm on one axis of rotation. While it is possible to successfully build this kind of arm, generally a single axis arm will only afford around 15-16” of extension, which is inadequate for nearly every game. The reason for this is that the longest a channel can be is 18” (technically you could have a longer channel by placing it diagonally, but this complicates matters). Thus, with a maximum of 18” of extension, a couple inches must be subtracted, since the point of rotation is inside the 18” sizing cube; therefore the extension is around 15-16”. Therefore, a single arm with further linear extension is needed, which complicates complexity even further. For example, 8103 Null Robotics built a single arm with an added linear extension mechanism to reach the desired extension length, which is generally >24”.

The advantages of a single arm are that it is relatively easy to build, and can be a quick way to gain some form of extension outside of the robot cube.

However, there are many disadvantages such as having a high gear ratio, requiring much more support than a linear slide, and controllability.

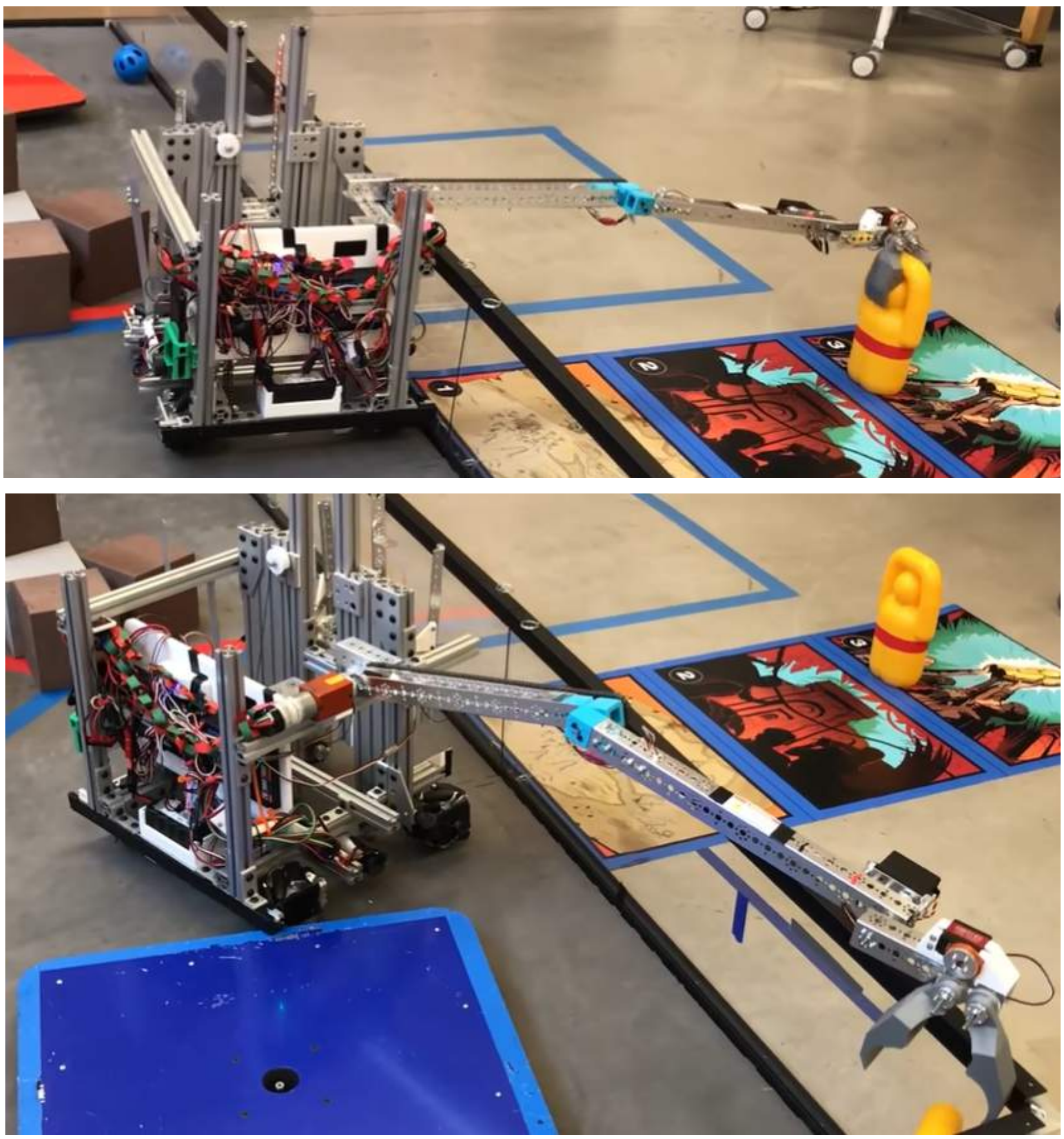


8103 Null Robotics, Rover Ruckus, single arm + custom belt driven linear extension

Chain Bar Arm

A chain bar arm is a special type of multi-axis arm that involves chain hard-bolted to a bar. The purpose of a chain bar arm is to make an extension such that the outer bar always keeps the same angle in reference to the ground. This means that if the outer bar is parallel to the ground in the stow position, it will remain parallel to the ground *at all points in the extension of the arm*. This is especially important if, for example, you have a claw to pick up an object, and you want the claw to remain parallel to the ground.

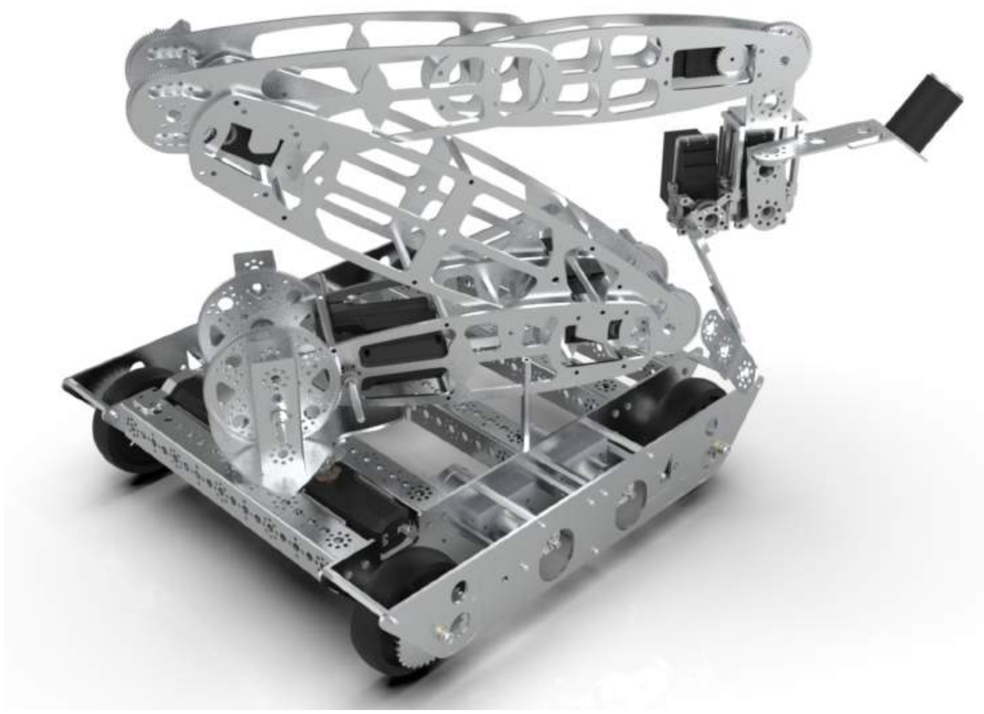
Chain bar arms can provide the >24” optimal extension needed to be competitive in games, but it also has disadvantages. It is much harder to control than a linear slide, since a chain bar arm has momentum. Thus, it is very difficult to be precise with a chain bar.



8103 Null Robotics, Relic Recovery

Multi-Axis Arm

A multi-axis arm is an arm which has multiple points of rotation. It is highly discouraged for inexperienced FTC teams due to the difficulty of building as well as the need for machine tools.



8148 Aleph Bots, Relic Recovery

Claws

Claws are probably one of the first mechanisms that jump to mind when teams brainstorm how their robot will control game elements. Many new teams use claws for their simplicity, compared to an intake, which requires a period of refinement and iteration. However, it is advised that teams move away from claws and focus on intake for competitive advantage. This is not to say that claws cannot be competitive - in Relic Recovery, for example, many teams used claws to great success (States/Super-Regional level). A claw robot was still able to complete all the robot objectives in RR1. However, every season, the best teams all used intakes, which goes to illustrate - practically every winning team from years past have had intakes and not claws.
However, a proper use for claws would be for picking up specific elements that cannot be controlled by intakes. For example, the relic in Relic Recovery would be impossible to control using a passthrough intake; thus, teams used a claw. Our recommendation is that if the game element is able to be controlled by intake, by all means pursue intake.

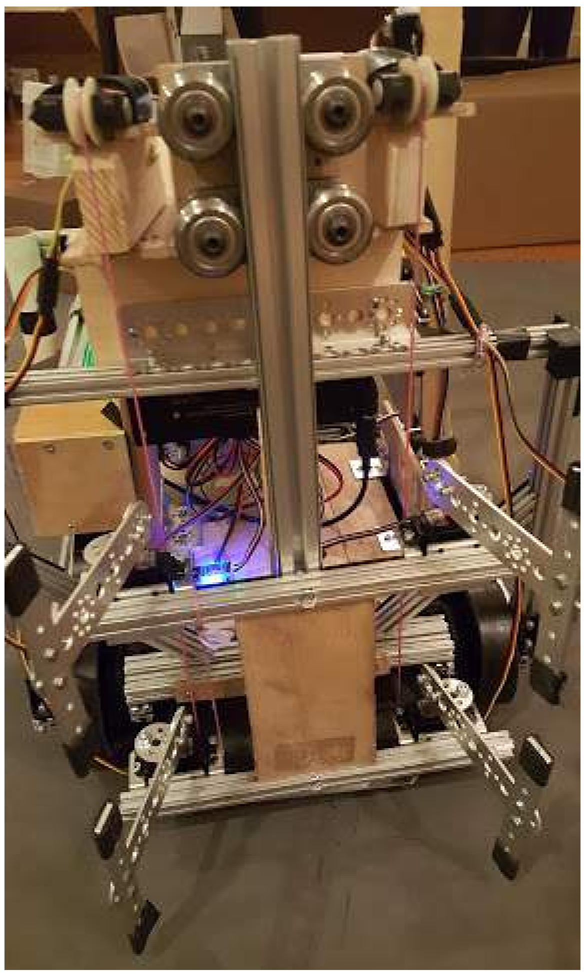
Advantages

- Much simpler than an intake
- Takes a lot less time to build than a reliable intake + transfer
- Less moving parts = less things that can go wrong

Disadvantages

- Claws are inherently slower than intakes.
    - This is because the driver must angle the robot, position the claw, and grab the game element. With an intake, the driver needs only to position the intake and grab the game elements.
- Claws can only grab one game element at a time, as opposed to an intake, which can grab multiple elements (depends on the game)
- Claws are fragile and break easily.
- Claws are prone to defensive maneuvers.

11115 Gluten Free, Finalist Alliance Captain (Detroit), Relic Recovery **(proper use case)**



13075 Coram Deo Robotics, Relic Recovery **(suboptimal use case)**

Intakes

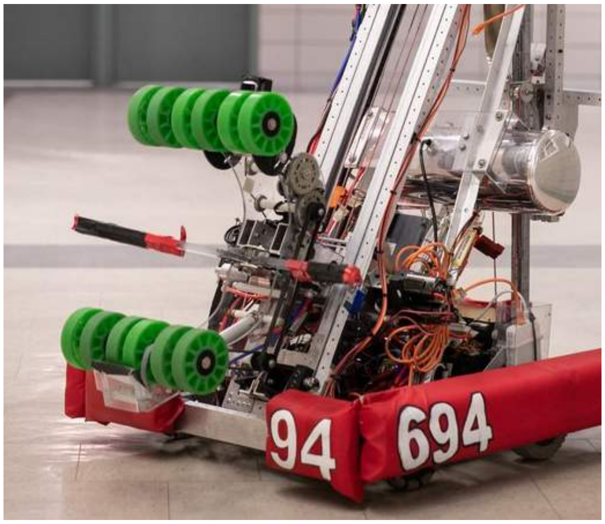
There's one main rule when it comes to intakes in FTC: you'll always need one. Claws are most almost always outperformed by robots using active intakes simply because a claw can't compete in ease of use, speed, and versatility. Looking back at the highest-performing robots in the past 5 seasons, the one thing that is consistent across all of them is that they use some sort of wheeled intake or roller intake. These robots were successful largely because of how easy it was for them to acquire and control game elements.

There are multiple parts of an effective intake: intake geometry, intake type and material, and speed. However, a common rule of them is the faster the intake, often the more effective it is. For example, near the end of Rover Ruckus, many of the top performing teams used a 3.7:1 (1600 RPM) motor or a 5.2:1 (1000 RPM) motor on their intake. **It is highly recommended that teams dedicate at least one motor to intake because of its necessity.**

When building an intake, keep in mind that the intake must funnel the elements into a collection area for deposit. This is because the materials cannot stay in the intake as they will get spit out.

"Touch it, own it": the JVN school of picking stuff up

When your intake comes in contact with game elements, you want it to instantly be controlling them. You should design with a margin for error, both in the orientation of the game piece and in the alignment of the robot. For example, look at FRC team 694's ball intake on their 2019 robot:



When this robot drives up to a ball, it barely has to point itself at the ball and it's instantly locked in between the two rows of wheels. What makes it so effective is the immense amount of testing that went into its design. The team tried a number of intake shapes, initially building with wood and rubber bands, and recorded which shapes were most effective. By testing their intake design out *before* competition, they didn't have to guess whether it would work as intended: they put it on the field with confidence.

Principles of an intake

Reliability

- An intake must be reliable, as picking up and scoring game elements is the primary method to gain points in FTC.
- Intakes have many moving parts and are susceptible to breaking, especially at high RPM. This, the intake material must be durable and withstand long periods of operation.
- Intakes often stick outside of the robot frame perimeter, so must be built so that they can withstand impacts/collisions with other robots or parts of the field. There are two ways to accomplish this - either by building the intake very robustly (lots of support so it doesn't break), or making the intake flexible (for example out of lexan or spring loading it) so that even though it may bend due to impacts, it will always spring back into place. Another way to prevent intake breaking is to make a fully retractable intake that won't protrude outside the 18" cube, though this is seldom needed.

Consistency

- The intake **must** be able to pick up game elements consistently. For example, in Rover Ruckus each robot could only possess 2 minerals at a time. Therefore, the intake should only pick up two pieces at a time, not zero, one, or three. This is a common mistake that many inexperienced teams fail to take into account.
- Another component is the varying angles that the game objects can be located in. This was especially apparent in the Relic Recovery season, where the "glyphs" (foam cubes) could be oriented in many directions. Even though it was relatively easy to make a compliant wheel intake that could effectively intake glyphs in one direction, it was hard to make an intake that could deal with angled glyphs. Being able to intake glyphs in all orientations was especially important for multi-glyph autonomous modes.

Controllability

- The intake must be able to consistently control the game elements. For example, if the intake is too fast and the collection box is not well designed, then game pieces might fly out. It is possible for pieces to get jammed at an unreachable angle, especially when using wheeled intakes. Optimally, the minerals should follow a certain path each time if funneling is done correctly.

Efficiency

- The key to any successful robot is cycle time. Reducing cycle time by having an efficient intake will lead to major improvements in score. A good intake should take no more than a few seconds to successfully collect the needed elements. For example, in Relic Recovery, the best intakes often had a <3 second collection time for two game elements, and in Rover Ruckus a 1 second collection time was desired.
- A key rule to remember in FTC is the shortest distance rule: how can you get minerals from A to B in the shortest distance? The answer is usually a couple straight lines. The

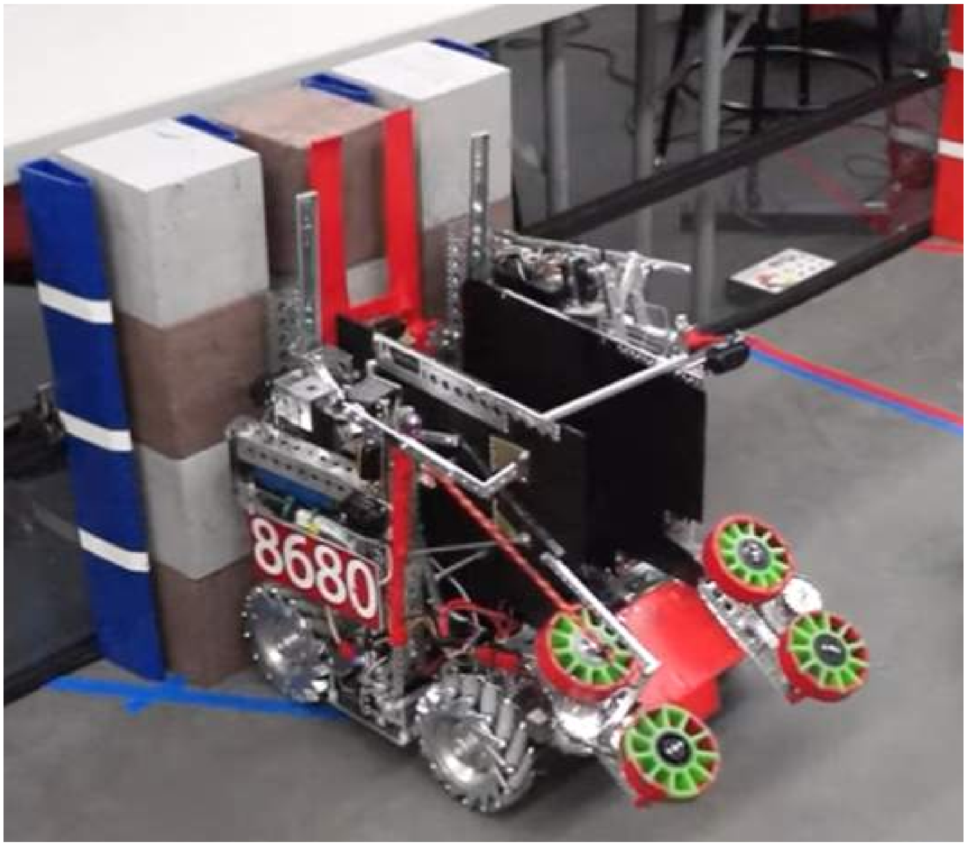
closer the minerals follow this path, the faster they will go from collection to deposit. Don't make overly long routes unless needed.

Types of Intake

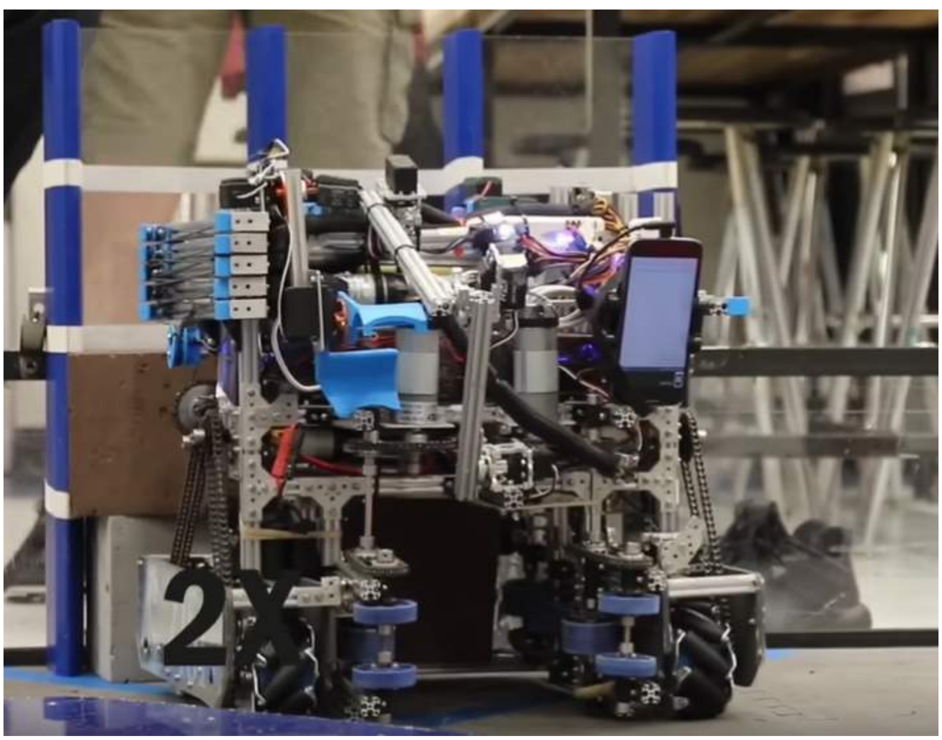
Compliant Wheel Intake

- The compliant wheel intake is most commonly used with large game elements such as the glyphs in the 2017-2018 season, Relic Recovery. In this game, robots had to pick up glyphs, which were 6 inch foam cubes, from the center pit and place them in the cryptobox. This game had many wheeled intakes primarily because the wheels had consistent and controllable contact with the glyphs. Wheeled intakes were able to propel the glyphs in a consistent fashion from the point of contact to the deposit plate, which would flip up to deposit the glyphs. A wheeled intake can use different size wheels (generally 2 in. and 4 in.) to prevent jamming.
- Wheeled intakes can be spring-loaded or locked into place. A spring-loaded intake is able to pivot in order to accommodate a game element when it is passing through the intake, but will snap back when the element has gone through. A spring-loaded intake requires more thought, but guarantees that the intake wheel will always be in contact with the desired element. A locked intake simply means that the wheels are locked into place and cannot pivot. Teams could choose one or mix; in Relic Recovery, some teams spring-loaded the set of wheels that made contact first, and then had a fixed set in the back. This is up to the design team's choosing.
- Wheeled intakes operate at much slower RPM than surgical tubing intakes, as wheeled intakes are meant to pick up **one** element at a time. They generally require more torque than a surgical tubing intake, which is geared for speed.

- Advantages
    - Very controllable
    - Propels elements to desired location
    - Great at picking up large elements
- Disadvantages
    - Picks up only one element at a time
    - Elements can get stuck in a bad position
    - Not generally used for picking up small elements
    - Can generally only pick up one type of element (not balls *and* cubes)



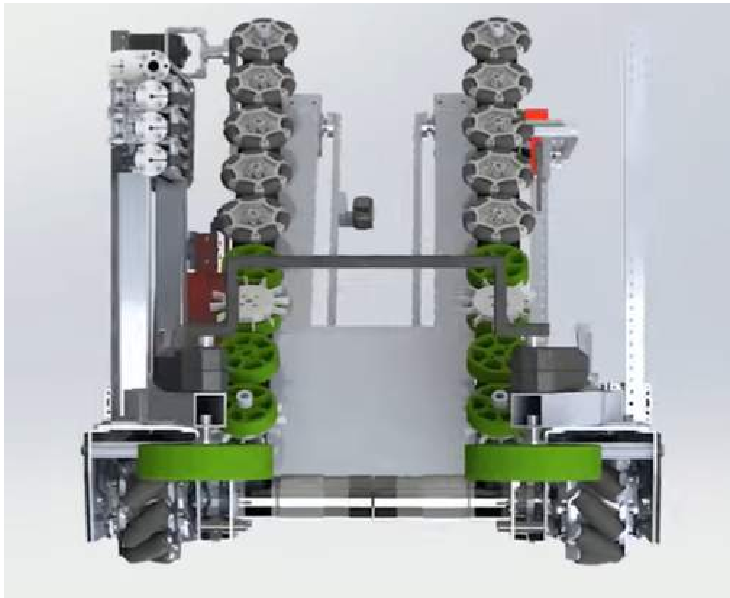
8680 Kraken-Pinion, Relic Recovery



7161 ViperBots Hydra, Relic Recovery

9971 LanBros, Relic Recovery, springloaded

11115 Gluten Free, Relic Recovery, springloaded



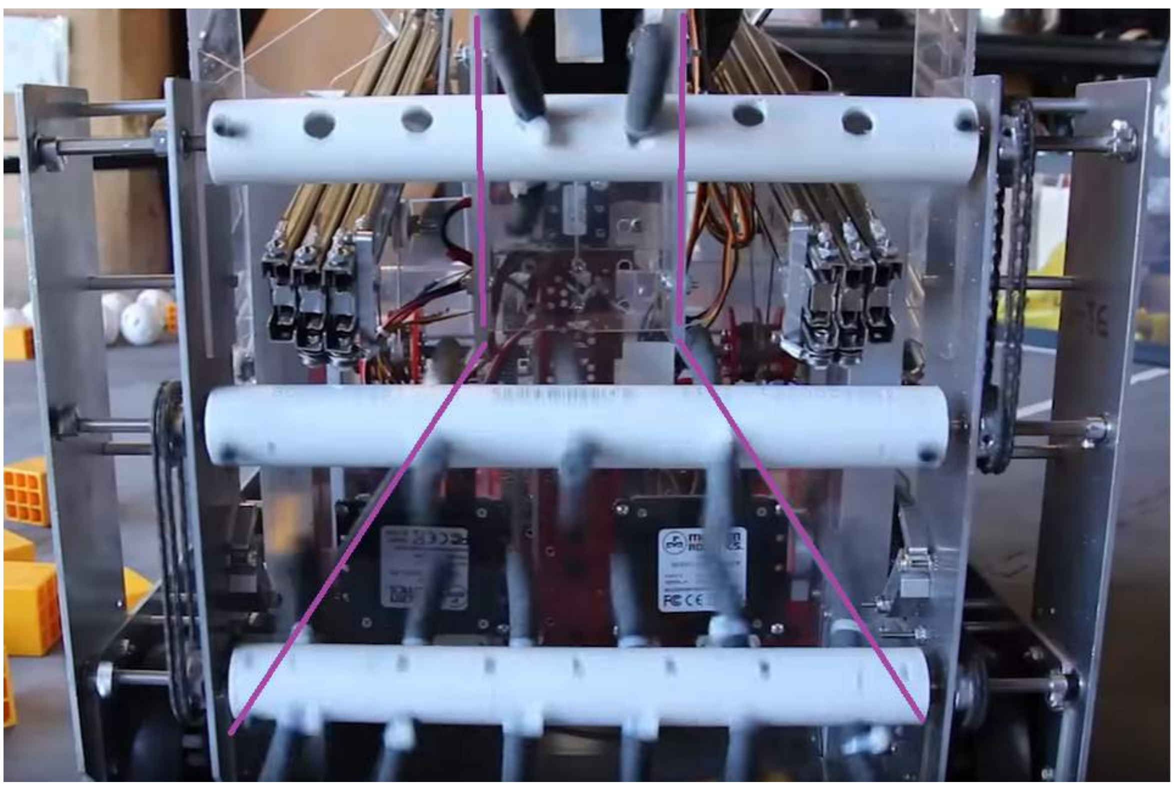
2856 Tesseract, Relic Recovery, 2 & 4 in. compliant wheels, 2 in. omni wheels

Surgical Tubing Intake

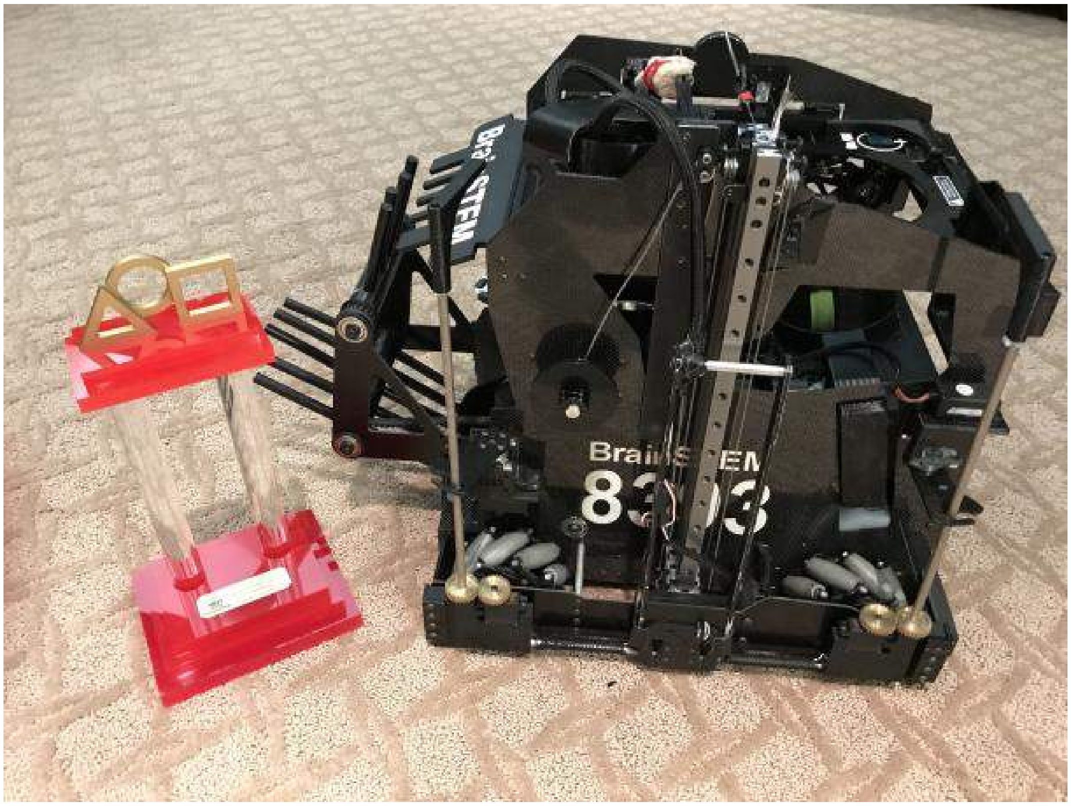
- Surgical tubing, sold by many different manufacturers, is a great option for picking up small game elements such as the minerals in Res-Q or Rover Ruckus. Surgical tubing intakes can, and often, have multiple sets of tubing in order to move minerals from the collection point to the holding box. This was most often seen in games such as Res-Q and Velocity Vortex, when robots had to transfer minerals from collection to a holding box that would dump the minerals into the goal, or to a flywheel to shoot the balls. However, having multiple tubes requires that they be driven, generally by chain, which can cause some complications for inexperienced teams. Unlike wheeled and rubber band intakes, surgical tubing intakes are practically always fixed at a certain height and angle, as there is no reason to have an adjustable intake.
- Surgical tubing by itself is often very soft and pliable. Teams have two options: 1) increase the RPM to 800-1000+ RPM, or use polyurethane tubing at a lower RPM (100-250 RPM). Polyurethane tubing can be purchased at a local hardware store and is a clear tubing that is quite stiff. Using some lubricant, insert the clear tubing into the surgical tubing for added stiffness. It is encouraged that teams test different RPMs and stiffness to develop the optimal intake.
- Surgical tubing intakes are especially good at picking up **multiple elements** at a time, due to the high RPM (sometimes >1000 RPM) of the rollers. However, it suffers from a lack of controllability, as sometimes the driver may accidentally pick up more than needed, and have to spit it out.

- Advantages
  - Able to collect multiple elements at a time
  - Generally more efficient than wheeled intakes
  - Specializes in small elements
- Disadvantages
  - Requires high RPM
  - Less controllable
  - Not able to pick up large elements easily

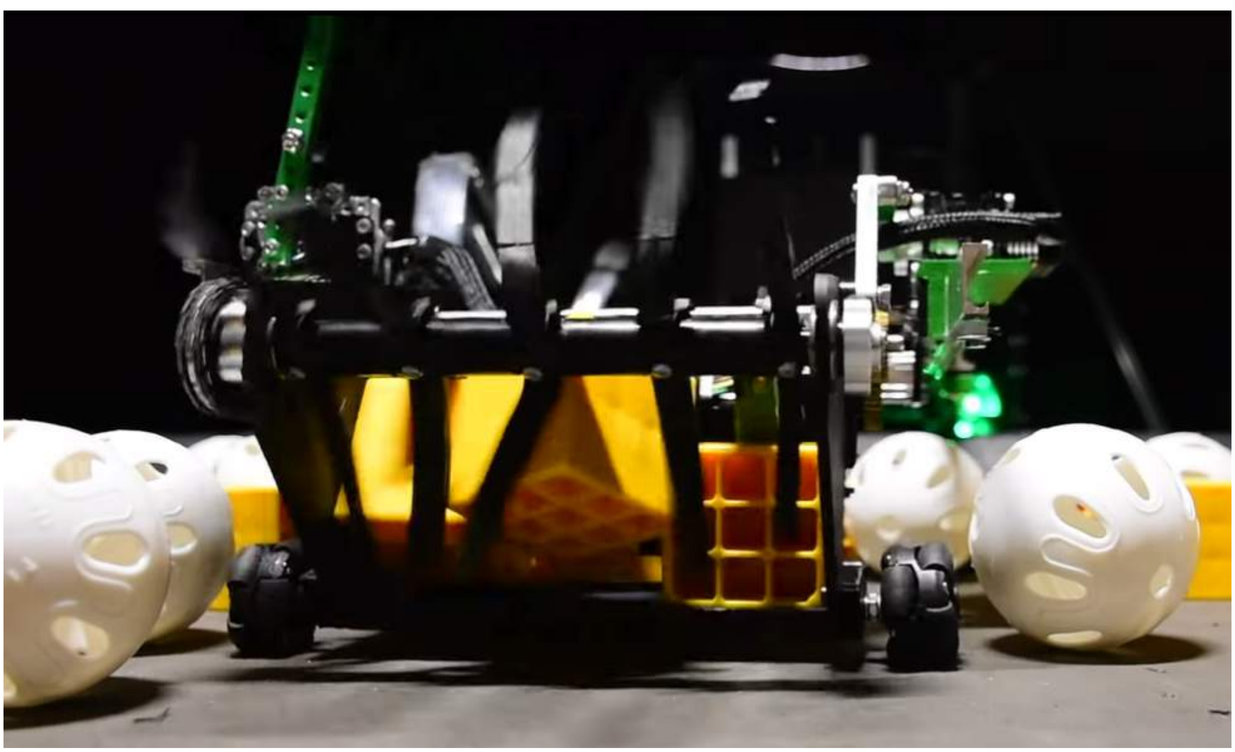
8644 Brainstormers, Velocity Vortex



8375 Vulcan Robotics, Res-Q - great example of funneling



8393 Giant Diencephalic BrainSTEM Robotics Team,
Winning Alliance 1st Pick (St. Louis), Relic Recovery



7203 KNO3, Rover Ruckus

Foam Roller Intake

- A foam roller intake uses foam rollers commonly found in the paint department of most hardware stores. It is generally used in games where the elements are balls. For example, Velocity Vortex saw quite a few teams use foam rollers to great success. Generally, foam roller intakes are the first point of contact, and then teams can use other means to transfer the element to the desired location. Usually, robots with foam rollers drive over the balls that they want to intake, however, foam rollers allow for very high throughput (it can pick up a large volume of balls very quickly).

6299 QuadX, Velocity Vortex

8686 Height Differential, Velocity Vortex

Other Intake Types

Rubber Band Intake

- Rubber band intakes, commonly used by VEX teams, generally feature sprockets, wheels, or gears at two ends, with rubber bands interlaced in between to form a pliable and bendable roller. Generally, it can be actuated or adjustable with a servo, although this is not necessary. Rubber band intakes are great with intaking balls, but not so great with other types of game elements such as cubes. It generally is slower than a surgical tubing intake, and requires multiple stages to transfer elements from collection to deposit. Zip ties can be added to increase the intake's range to accommodate for smaller balls.

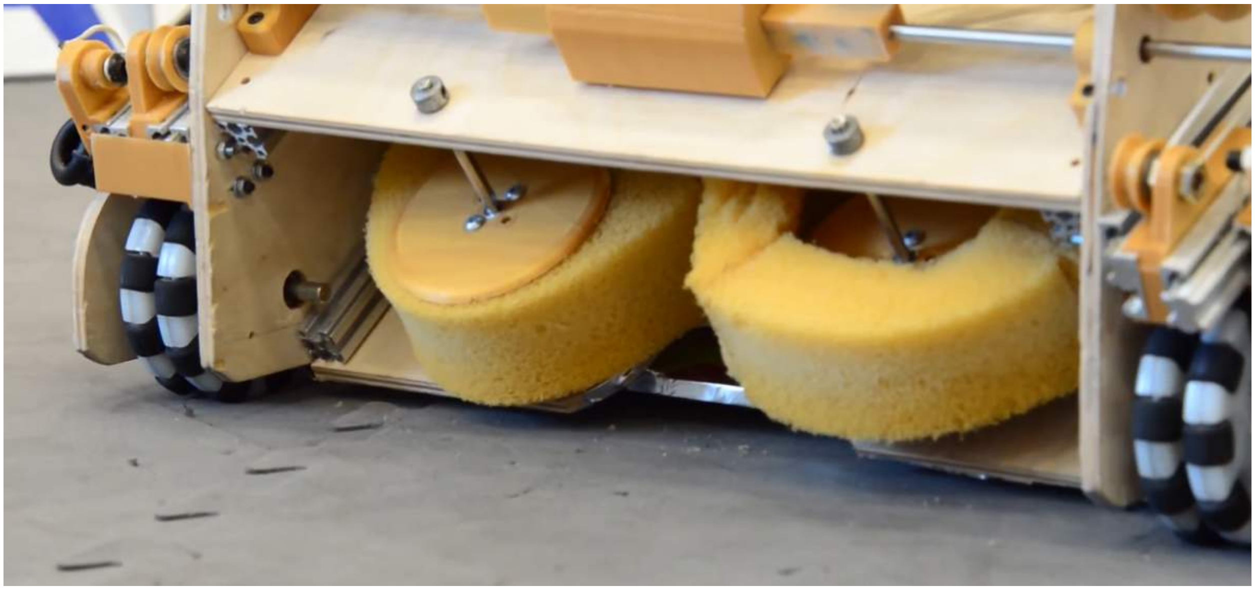
Ball intake for VRC Game Turning Point

Zip Tie Intake

- Instead of using surgical tubing, some teams opt for heavy zip ties instead. This can work, but we recommend surgical tubing as it is one of the most tried and tested methods for picking up nearly any game element.

Foam Wheel Intake

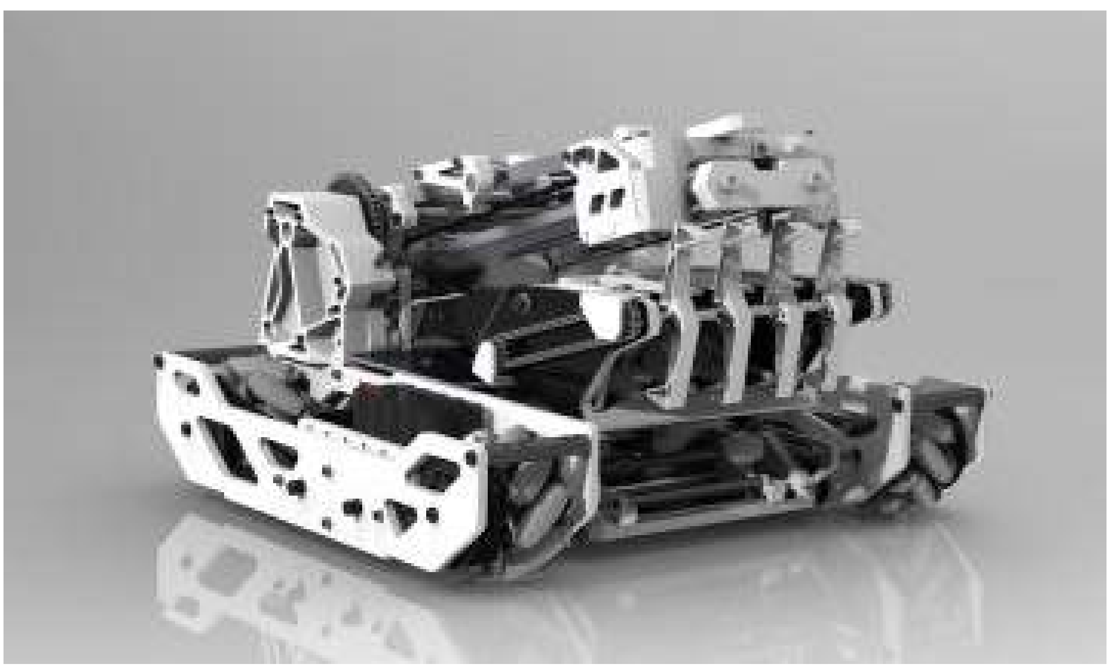
- A foam wheel intake has the same principles as a compliant wheel intake, except that it uses foam wheels. It is generally recommended that teams stick to compliant wheels as they are grippier and easier to control.



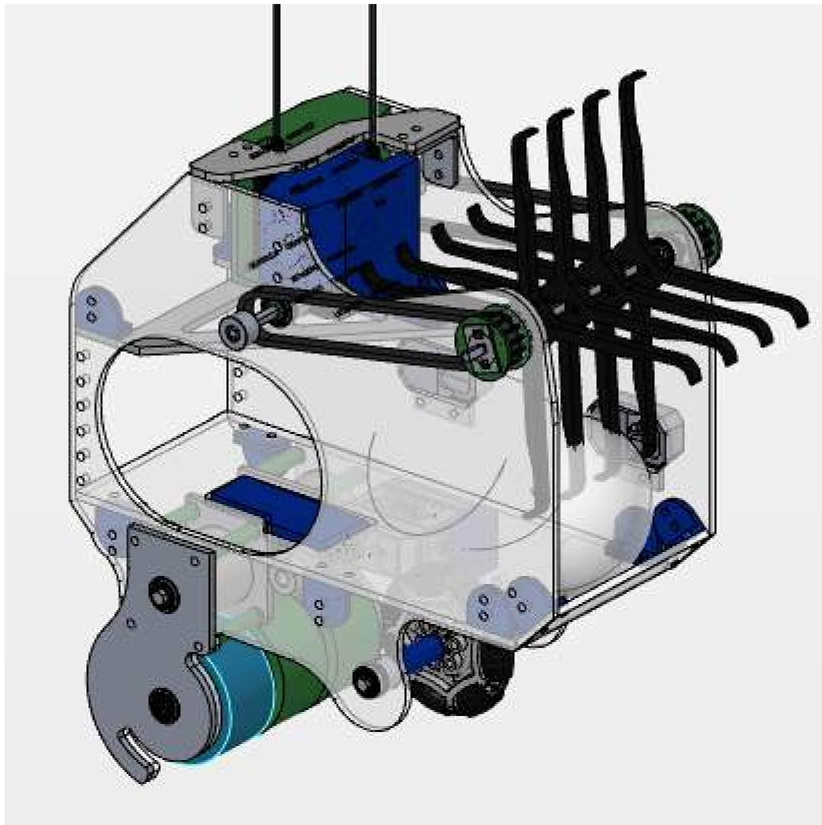
6299 QuadX Res-Q

3D printed intake (NinjaFlex/TPU Filament)

- TPU/NinjaFlex 3D printer filament is a great low-RPM intake flap option, if designed right they work very well with VEX 393 motors (considered servos in FTC), and several teams have successfully used this configuration in competition.
- We recommend 3D printed intakes only if your team has had experience in 3D printing parts.



731 Wannabee Strange, Rover Ruckus



8417 Lectric Legends, Rover Ruckus - TPU intake flaps, powered by two VEX 393 motors

Motor Guide

Motor Hardware

Shafts

In FTC, shafts are the way to transfer power from the motor to the mechanism it is powering. Keep in mind that some shafts such as 8 mm round shaft cannot use set screws (although set screws are generally discouraged). Adapting from different shaft types can be very inelegant and inefficient, with the exception of 5 mm to/from ½” hex. The different types of shafts in FTC are as follows.

- Hex
    - 5 mm hex
    - ⅜” hex (Thunderhex)
    - ½” hex (Ultrahex, Thunderhex)
- D-shaft
    - 4.7 mm D-shaft
    - 6 mm D-shaft
    - ¼” D-shaft
- Round
    - 6 mm round shaft
    - 8 mm round shaft
- Square
    - ⅛” square shaft
- REX (round and hex profile shaft)
    - 12 mm diameter

Connectors

There are different types of connections from various manufacturers, which is another aspect of motors to consider. Teams need to adapt to the JST system used by REV if they are using the REV Expansion Hub. Here are the different types of connectors for FTC motors.

- JST connector (standard REV)
- Anderson PowerPole
- Bullet connector

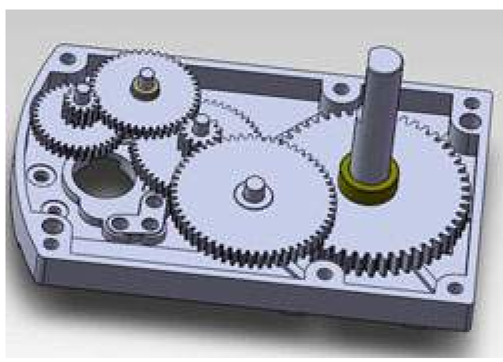
Gearboxes

Fundamentally, a gearbox is just a collection of gears and an enclosure that causes those gears to connect. Gearboxes have an output ratio, the final gear ratio between the motor input and the final output shaft. Gear ratios are further explained in the glossary.
In FTC, gearboxes may be more common than you think - every motor has a gearbox attached to it. These gearboxes are one of the following two types:

Spur Gearboxes

Spur gearboxes are an arrangement of gear ratios, often stacked to achieve a large compound ratio (e.g. 40:1). Each individual ratio only has two gears- one may be 8:1, another may be 5:1, but the final ratio will be 40:1. These gearboxes are used in the Andymark NeveRest Classic series and goBILDA's 5201 series motors, as well as some REV HD Hex Motors (gearbox types are denoted on REV's website very clearly). Due to the nature of how these gearboxes are built, each reduction only has a few teeth from each gear engaged, and those teeth carry the entire load of the gearbox. It's easy to damage a spur gearbox from shock load, and if anything happens to one of the gears, the gearbox will stop functioning. **We do not recommend using spur gearboxes on high-load applications such as drivetrains or arms. Instead, use planetary gearboxes.**

Example of a spur gearbox. Note how all gears mesh with only one other gear.



Reasons to use a spur gearbox

Cost. Generally, spur gearboxes are cheaper than planetary gearboxes. However, in FTC that price change is often minimal. A 20:1 planetary gearbox from Andymark is only $2 more than a spur 40:1 (used 40:1 for cost comparison because there is a lack of overlap in gear ratios between planetary and spur gearboxes from vendors).

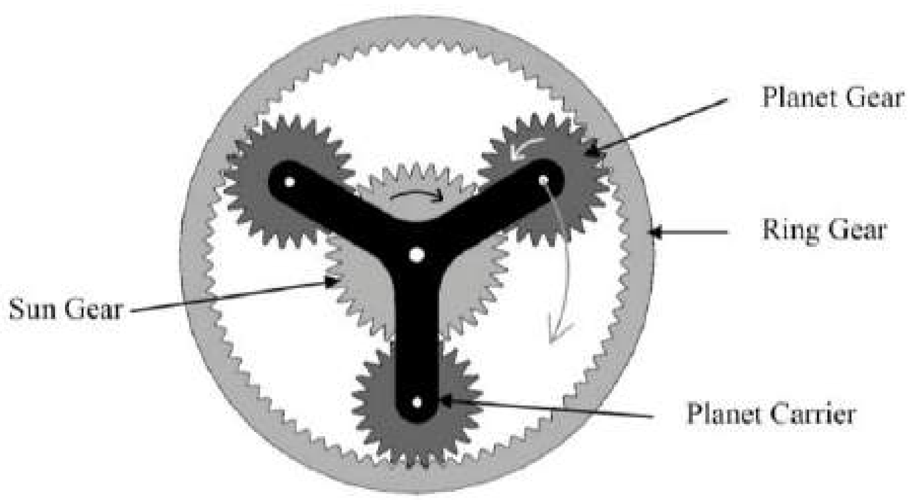
- Spur gearboxes from Andymark NeveRest Classic motors, the REV HD Hex 40:1 Spur motor, and goBilda 5201 Series Yellow Jacket Spur Gear Motors. The gears in these gearboxes, while not interchangeable, are very comparable and practically indistinguishable in performance (when they are the same reduction).

The main thing to consider here is your **desired reduction, your desired motor connections, and your desired output shaft type.**

- goBilda's 5201 series spur gearboxes are much cheaper than either REV's or Andymark's; whether that's a good or bad thing is up to you. They utilize the rather uncommon (in the FTC world) bullet connection for power. The output shaft is a 6mm D-shaft.
- REV HD Hex Planetary 40:1 motor - This motor comes only in a 40:1 ratio, but does use the same connections (JST VH) as the REV Expansion and Control hub for power which means no adapter cables. The output shaft is a 5mm hex shaft. REV UltraHex has a 5mm hex bore running through the middle of a ½" hex shaft, which makes adapting this motor to any length of Ultrahex, and by extension, ½" hex shaft, very easy.
- Andymark NeveRest Classic motors come in a few different ratios, which are 40:1 and 60:1. The output shaft is a 6mm D-shaft, and like all NeveRest motors use the Anderson Power Pole to connect to power. This connector is perhaps the most robust of the ones listed here.

Planetary Gearboxes

Planetary gearboxes use a more complex system of gears to achieve a robust reduction in a compact space. In automotive engineering, planetary gear sets can achieve a few different ratios without changing gear size, but all planetary gearboxes that you will see in FTC only achieve one gear ratio. Planetary gearboxes are used in the Andymark Orbital series, some REV HD Hex Motors (gearbox types are denoted on REV's website very clearly), and goBilda's wide selection of planetary gear motors. Additionally, AndyMark sells a few aftermarket planetary gearboxes called NeveRest Sport and 57 Sport, and VEXpro sells the customizable VersaPlanetary gearbox. As you can see from the graphic below, there are more teeth meshing per stage than in the spur gearbox.



Example of a planetary gearbox stage. Note how the sun gear meshes with >1 gear.

Reasons to use a planetary gearbox

- Backlash is lower than spur gearbox equivalents. Backlash is defined as the clearance or lost motion caused by gaps between parts. This can be easily be explained through putting a wheel or gear on a motor shaft and lightly rotating it. The part should be able to wiggle around a little without having considerable force imparted on it. This is caused because it is impossible for the gear teeth inside the gearbox to mesh perfectly, and is the same for chain and sprockets, or any other form of power transmission. However, planetary gearboxes have less backlash as they have less stages of gears.
- Efficiency is better than spur gearboxes. A typical two-stage spur gearbox is about 85% efficient, whereas most two stage planetary gearboxes are 94% efficient.
- Load capacity is higher for planetary gearboxes. This is due to having multiple teeth engaged per stage, which spreads the load. **This means planetary gearboxes will not break as easily when used in high-load applications such as drivetrains.**

Available Planetary Gearboxes and Motors

Low-end planetary gearboxes from Andymark NeveRest Orbital motors, REV 20:1 Planetary motor, and goBilda 5202 Series Yellow Jacket Planetary Gear Motors.
These aren't all that low end, but are when compared to other planetary gearboxes available. Any of these "low-end" gearboxes are more robust than the "regular" spur gearboxes. Like the spur gearboxes, the gears in these gearboxes, while not interchangeable, are very comparable in terms of robustness. Once again, the main thing to consider here is your desired reduction, your desired motor connections, and your desired output shaft type.

- goBILDA has, by far, the most varied selection of gearbox ratios with too many to list here, but utilize the rather uncommon (in the FTC world) bullet connection for power. The output shaft is a 6mm D-shaft.
- REV HD Hex Planetary motor - This motor comes only in a 20:1 ratio, but uses the same connections (JST-VH) as the REV Expansion and Control hub for power which means no adapter cables. The output shaft is a 5mm hex shaft. REV Ultrahex has a 5mm hex bore running through the middle of a ½" hex shaft, which makes adapting this motor to any length of Ultrahex, and by extension, ½" hex shaft very easy. The ratio of HD Hex is 20:1.
- Andymark NeveRest Orbital motors come in two ratios, which are 3.7:1 and 20:1. The output shaft is a 6mm D-shaft, and like all NeveRest motors use the Anderson Power Pole to connect to power. This connector is perhaps the most robust of the ones listed here.

High-end planetary gearboxes include Andymark NeveRest Sport and VersaPlanetary gearboxes, which can set your team back quite a bit.

- Andymark NeveRest Sport gearboxes - These gearboxes were designed to be more robust than the gearboxes found on Andymark NeveRest motors. The NeveRest sport gearboxes can handle larger shock loads, and can provide a larger reduction with options up to 256:1. These gearboxes are relatively expensive, but still not the most expensive gearbox you can use in FTC, with prices ranging from $46.00 for the 4:1 reduction to $77.50 for the 256:1 reduction. There are no motors included with the gearbox, you will need a bare NeveRest motor, REV HD Hex motor, or goBilda motor to attach to the gearbox. The selection should be based only on which connection type you like the most (powerpole vs. JST VH vs. 3.5mm bullet) for power.
- VersaPlanetary gearboxes - These planetary gearboxes were designed for FRC use and can be very expensive (upwards of $80 for a 40:1 ratio) but because they were designed for such heavy use the VersaPlanetary is arguably the most durable gearbox you can use in FTC. If you somehow manage you break one of the components of the gearbox, they are very easy to repair. Unlike other gearboxes, almost every part of the gearbox is replaceable and able to be bought by itself. Every other gearbox on this list would require you to buy a whole new gearbox if one part of it breaks. In addition to that, you can disassemble the gearbox from the motor-side of the gearbox, meaning in the event that something breaks, you don't have to remove the gearbox from the plate it's attached to to be able to switch that part out. Another nice thing is that attaching the pinion to a bare motor is super easy since it clamps onto the motor shaft instead of being a press fit like the NeveRest sport pinion gear is. This means you don't need any special tools like a press and can switch motors out very quickly, even during a competition. VersaPlanetarys also allow you to choose your own ratio with swappable 3:1, 4:1, 5:1, 7:1, 9:1, and 10:1 stages. When using VersaPlanetary gearboxes there are a few important factors for you to consider.
    - The higher the reduction, the weaker the stage is. So for instance, a gearbox with 4:1, 4:1, 5:1 stages would be stronger than a 10:1, 10:1 gearbox though they end up with the same output ratio of 100:1.
    - Always place the highest reduction as close as you can to the motor. There is less load placed on it here, and is less likely to break than if you placed it higher in the reduction. Then after your weakest stage, you should use your next weakest stage, until you reach the output stage.
    - Vex Pro also sells a VersaPlanetary Lite option, which replaces the outer aluminum ring gear and housing with a 30% glass filled nylon. VersaPlanetary Lite gearboxes use the same gear sets as the regular VersaPlanetary, so if you ever feel the need for a stronger gearbox you can mix and match parts to fill your requirement. While still expensive (compared to other planetary gearboxes from Andymark, REV, and goBilda) at about $50 for one gearbox, it's still one of the strongest gearboxes available for FTC use. It should be noted that the lite version comes with a ⅜" output shaft only, whereas the more expensive VersaPlanetary comes with various options including ½" hex, ⅜" hex, ½" round with ⅛" keyway, a CIM Output option, and a Universal Female Output Shaft. However, the shaft can

be swapped out for a different one. The VersaPlanetary lite also loses side mounting holes on the gearbox.
- Like the NeveRest Sport gearboxes, you will need to supply your own bare motor to attach to the gearbox.

Choosing the right gearbox

For regular use, any of the “low-end” (see above) type of planetary gearboxes will fit your needs. Planetary gearboxes are just a tiny bit more expensive, but boast better backlash and efficiency, higher load capacity, and better capacity for shock loads than spur gearboxes. The tradeoffs, cost and mechanical noise, are almost never a factor. **Because both gearbox types are so similar in price for similar ratios, it’s impossible to recommend the use of a spur gearbox over a planetary gearbox for any use case in FTC.** If you already own spur gearboxes, try to use them in lower-load situations and try to put planetary motors at least on your drivetrain.

For larger reductions, the 57 Sport and VersaPlanetary gearboxes are a viable but expensive option. In some cases, it may be more economical to choose a motor you already own and build an external reduction using gears, chain, or belts. It should again come down to your desired output shaft, desired gear ratio, and for the VersaPlanetary, whether you want the ability to swap parts out on the fly. You also may be able to buy 57 Sport gearboxes off of local FRC teams at a solid discount - some may be willing to sell their older gearboxes.

The motor itself, meaning the part that actually has power running to it and makes the shaft spin, not the gearbox that is often mounted on the motor, doesn’t matter all that much in FTC. The motors from Andymark, REV, and goBilda might have different specifications on their websites, but they’re all very similar. The VEX motor testing (link in JVN Calculator for FTC) shows the accurate specifications of a bare RS-555 series motor. When choosing between companies, always go with what works best with your motion system, and which is easiest to work with your existing electronics system.

JVN Calculator for FTC

With the JVN calculator for FTC, it is possible to calculate the gear ratio that is necessary for an arm. The JVN Design Calculator is a very valuable resource for picking appropriate gear ratios for almost application you need. It was developed by John V-Neun, coach of FRC 148, and was initially designed for FRC calculations.
It is fairly intuitive, but can save a lot of time doing calculations on your own, especially when developing designs like arms, linear slides, and drivetrains. This is as the balance between power/torque and speed is critical to the end design. For example, arms require much more torque than speed, while linear slides need very little torque to maximize speed.

To use the calculator for rotary mechanisms (arms), select the Rotary Arm sheet. A good practice is to "pencil in" VEX's dynamometer testing for the bare NeveRest motor. **The VEX dyno test is a trusted and accurate resource for teams, even though its results are significantly higher than the specifications posted on vendor pages**. The test results may be found by searching VEX Dyno Testing, or by clicking the link https://motors.vex.com/other-motors/am-NeveRest. Then change the gear ratio in the sheet itself. The bottom left of the sheet has a gear reduction page. Simply put the overall gear ratio of your gearbox in the right side, keeping the one on the left. For example for an 20:1 gearbox, write in "1" in the driving gear column and "20" in the "driven gear" column. Then, add the additional gearing. For example, if the total ratio is 40:1, there should be a 1:2 ratio in the driving:driven gear ratio below the 1:20. It doesn't matter whether it is 10:20, or 42:84 - only the proportion matters. This gives you a lot of flexibility when looking for the right ratio. Additional gear reductions can be added if needed.

The same goes for linear mechanisms, with one additional note. The calculator will ask for a pulley diameter, so provide the diameter of the spool that the string is wound on. For a horizontal linear mechanism, neglect stall load. For vertical linear mechanisms, only pay attention to stall load if the mechanism will hang the robot. It is recommended that the load is 60+ pounds for a 40 pound robot for safety. A good starting point is around 6 in/sec, but most teams will increase the speed over time.

For 1-speed drivetrains, just enter in the correct motor specifications and add in the total weight and wheel diameter. Don't worrk about the efficiency and speed loss, just keep it where it is. The adjusted speed will account for sag into field tiles, loss of efficiency through belts, gears, etc., and loss of grip as the wheels eventually wear out. A good starting point should be around 1.5-2 ft/s, though many teams have faster drivetrains.

General Tips

- Always use the VEX dyno test results for free speed, stall torque, stall current, and free current.

- Ensure that the total current draw does not exceed the stall current! Generally, this should not be a problem, but it is someting to be aware of.
- Stall load refers to how heavy a weight the mechanism can lift at the end before it stalls. **It is a good principle for the stall load to be 1.5x-2x, or even more, of the actual load.**
- For linear slides, maximize linear speed by using a low gear ratio. For arms, maximize torque by using a high gear ratio.

Download Link (may be broken - if so, search JVN for FTC):
https://www.chiefdelphi.com/uploads/default/original/3X/1/6/16e019399060799a45f54f4d75a8aa5fee1f394f.xlsx

Servo Guide

First, a disclaimer. **Servos are NOT replacements for DC motors, and should not be used as such. Servos are made for fine-tuned and accurate movement, not high-load or quick rotation applications.**

Servos are an interesting beast in FTC. They are their own animal and vary widely in price, performance and value. There's no way I'll be able to cover everything about this in this paper, but I'll try to hit the important points.

Picking the right servo for your application is a question that's almost impossible to give a blanket answer for. There are two subcategories of servo specifications that you should use to decide what servo you want to use. These are **output power** (measured in speed and torque) and **durability** (normally best represented by the gear material inside the servo).

Let's start with durability, as this is probably the easiest to summarize. The two things that threaten a servo's longevity are the motor inside burning out and more commonly, the gears stripping inside the servo. A motor burning out is pretty uncommon, but it can happen under large loads for a prolonged amount of time. This means you should never stall a servo against an immovable object. When selecting a servo, pay attention to the material the gears are made from. This ranges from plastic to titanium, so let's go down the list, starting from the weakest.

- **Plastic**: with low power servos, these are normally okay. Generally used for applications in model airplanes such as ailerons.
- **Karbonite**: Hitec's gear plastic is a very durable and long lasting plastic and is very good under long use and low load. Be aware that it can strip easily under the shock loads found in FTC.
- **Brass:** Brass gears are stronger than plastics but also suffer greatly when faced with shock loads in FTC, like intake wrists and deposit buckets. It's found on slightly higher end servos in the $25-30 range, such as the REV smart servo.
- **Steel:** This is where we start getting big. Steel gears are very durable and you'll have a tough time stripping these. Expect to pay $40+.
- **Titanium**: Titanium is where you get into really high end servos, they're virtually unbreakable. Savox servos start at $75 and reach into $100+.

The next variable is output power, measured in both **speed** and **torque**. Speed (normally in seconds per 60°), refers to how fast the servo turns 60 degrees in Standard Rotation mode. Torque (measured in oz-in), refers to the amount of force the servo can apply to a lever. Ounce-inch means if you put a 1” bar on a servo, then put a force gauge on the end, the torque rating of the servo (in oz/in) will be measured. As you may know, speed and torque have an inverse relationship. Generally you can find some insanely powerful servos that are pretty slow (slower than 0.20 s/60°) or some less powerful ones with faster ratios (anything faster than 0.12 s/60° is considered very fast). Finding the right

servo for your application can be tough, but a good way is trying to decide if you need more speed or torque, and if your servo will experience shock loads or not.
The last proposition is cost. Servos range from cheap $7 servos for light applications, all the way up to some Hitec or Savox servos for close to $200. By far the best bang for your buck servos out there are the Feetech dual mode servos, which is a programmable type of servo. This includes both the **REV SRS** (Smart Robot Servo) and **both goBILDA Dual Mode servos**. The biggest downside to the REV SRS and the old goBILDA servos are their brass gears. Coupled with high output power, this meant that stripping gears with any shock load was commonplace. The new goBILDA Dual Mode servos have steel gears now, but are new and aren't as competition tested as other servos. The next big name in FTC Servos is Hitec, who are a huge name in hobby servos for decades and are very well trusted. Their low end servos are inexpensive but easily broken (though it's hard to compare to a Feetech Dual mode).
A mid-priced Hitec servo is the HS 485-HB servo, with Karbonite gears. While it shouldn't be used in high load applications, it is fine for general use such as claws or trapdoors. Where Hitec really shines is the high end market. If your budget is over $100, you can get into some very powerful Hitec servos. Most have titanium gears and are programmable, so you can dial in the performance and range to exactly what you need. The last big player in the servo market in FTC is Savox, which produces great mid-high range servos (think $60-$100+). They are made with titanium gears (close to bulletproof) and are **fast**. Savox servos are mostly brushless and coreless, so they do tend to scream a little under load, but they're definitely worth it if your budget allows for it.
Another question we get is "what is the extended and continuous rotation options on the Servocity website/other sites?" You don't have to worry about those - as FTC ruled those modifications were illegal, just save a buck and buy stock servos. The only legal continuous rotation servos are servos that are continuous rotation from the factory. This includes the REV SRS servo, programmed to continuous rotation mode. In other words, if you have a seller like ServoCity modify a servo to continuous rotation, that servo is not legal.

Best Value
- Low Priced (~$18)
  - Hitec 485HB
- Medium Priced (~$30)
  - goBILDA Dual Mode Servo
  - REV Smart Servo

Best Performance
- Savox titanium servos
- Hitec titanium servos

-Ethan, Servocity and goBILDA Tech/R&D (tech@servocity.com)

Electronics

Introduction

The FTC control system is based on using Android phones as a “Robot Controller” and a “Driver Station.” The two major control systems of FTC are Modern Robotics and REV; however, nearly all teams are shifting over to REV because it is cheaper, simpler, and more reliable. More information about the FTC Control system can be found below.

Official control system Wiki on GitHub

REV Expansion Hub Documentation

FIRST troubleshooting guide

Tips and Tricks

In addition to what is written in the official resources, there are a couple of additional tips.

1. The traditional XT30 connector that is used to REV is prone to breaking. It is highly recommended that teams replace XT30 connectors with anderson powerpole, or put adapters on their current wires. An example of an adapter is this: (link). In lieu of this, teams can also 3D print strain relief connectors on the Expansion Hub to prevent XT30 disconnects. The file can be found on Thingiverse or through the link (https://www.thingiverse.com/thing:2887045).

2. The Tamiya connectors found on many of the FTC legal batteries are very weak and prone to becoming unreliable after 10-20 repeated plug/unplug cycles. It is recommended that teams crimp new Anderson Powerpole connectors onto the battery.

3. It is highly recommended for teams to use the [REV grounding strap] and the [REV USB strain relief] to help prevent disconnections.
4. To protect wires, teams often use [wire loom], and to help wires extend far, teams often use cable chain (also known as cable carrier.)

Wiring

Wiring is extremely important in FTC, as wires are crucial to the operation of the robot. It is highly discouraged for teams to overlook wiring, but many new teams seems to disregard it or throw it to the very last moment. While tedious and sometimes no fun, wiring can mean the difference between a win and loss.

Connectors and Wires

- There are many types of connectors for use in FTC. Here are the most common connectors you will find on an FTC robot.
    - PowerPoles or XT30 connectors for module power

    - PowerPoles or JST-VH connectors for motor power

    - Dupont 0.1”, JST-PH or JST-XH connectors for sensor wires

    - Dupont 0.1” connectors for servo wires

- The connectors you **SHOULD NOT** use:
    - Tamiya Plugs (the metal connectors are fragile and will lead to random disconnects)

FTC Wiring Guide

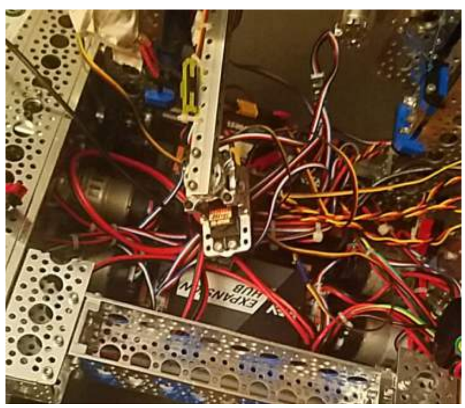
FIRST has created a wiring guide to help teams with tasks like crimping cables, soldering connections, and ESD mitigation that won't be covered in this guide. Once you read up on electronics and wiring here, look at the FTC Wiring Guide for the best practices and more tips & tricks. In addition, *FIRST* has written an ESD mitigation whitepaper that is worth taking a look at.
Robot Wiring Guide
ESD Mitigation Whitepaper

General Advice

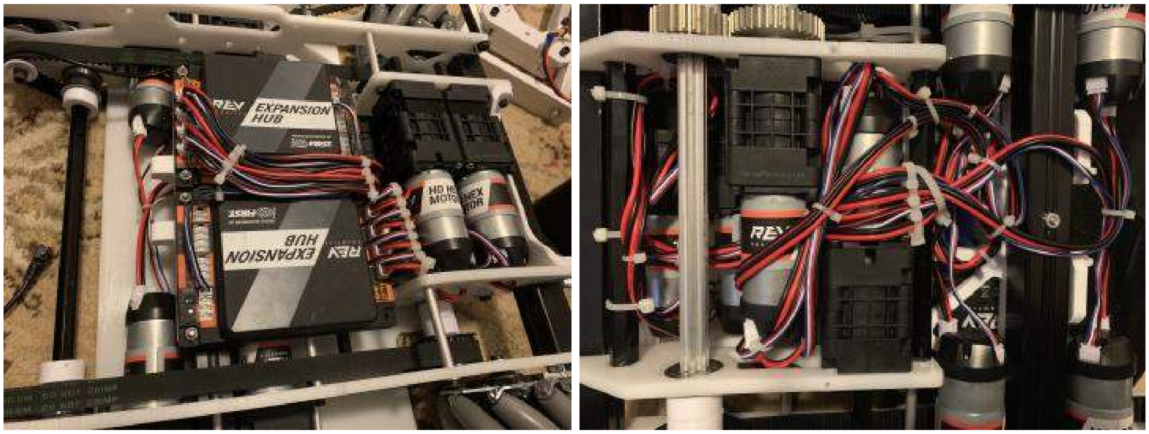
- **Always label wires! When bunched up, you may not know which wire goes into which port.**
- **Treat every wire connection as a point of failure**. Therefore, use electrical tape to tape up and insulate connections and utilize strain relief as much as possible.
- **Strain relief** should be used everywhere possible. It is highly recommended for teams to use products like the REV USB Retention Mount, as well as 3D printing strain relief methods for devices such as the Expansion Hub and robot controller phones.
- DO NOT solder a wire before crimping it. Solder can "creep" and losing connection is possible, possibly leading to fire.
- Keep all wire runs as short as possible to prevent entanglement and improve wire management.
- When using data/sensor cables, keep them away from motors. This will reduce electromagnetic interference (EMI). Add a ferrite bead if possible.

- Crimped connectors are generally better than soldered connectors, as solder joints can break easier than a crimped connection.
- Keep wires tucked away from moving mechanisms, and ensure that you will not be at risk of a mechanism snagging a wire. This is a proper application of materials such as acrylic, which allow drivers to see inside the robot while keeping wires out of the way of other robots/game pieces. It is advised for teams to purchase removable velcro ties or cable ties to aid with cable management.
- For power wires, lower gauge (larger size) wires are preferable. This means a lower resistance across the wire and higher power throughput. This is negligible for data wires, however.
- Small wires and cables are fragile. Treat them as such, and don't put them in an area where they will constantly be hit by another object. Larger power cables can take much more abuse.
- Ensure that your wires are kept out of pinch points where another mechanism could sandwich the wire. This is especially important in arms or mechanisms that are hinged.

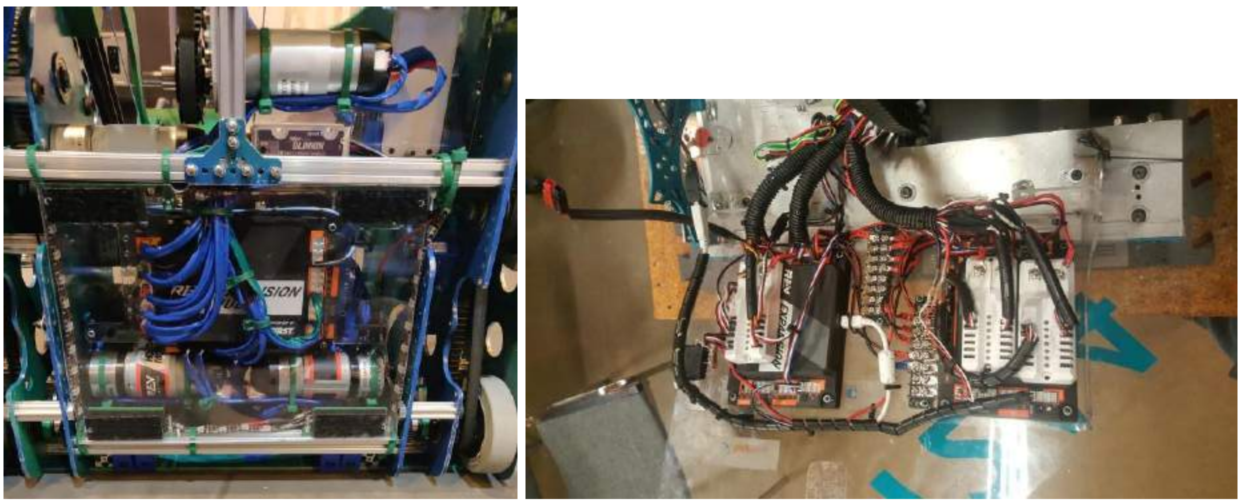
When wiring, take time to properly route wires and ziptie/velcro them together. It is important that all wires are labeled accordingly before starting. Here are some examples of good wiring practices that facilitate ease of repair and replacement, and will not have wires interfering during a match, as well as a bad example.



Bad example of wiring!

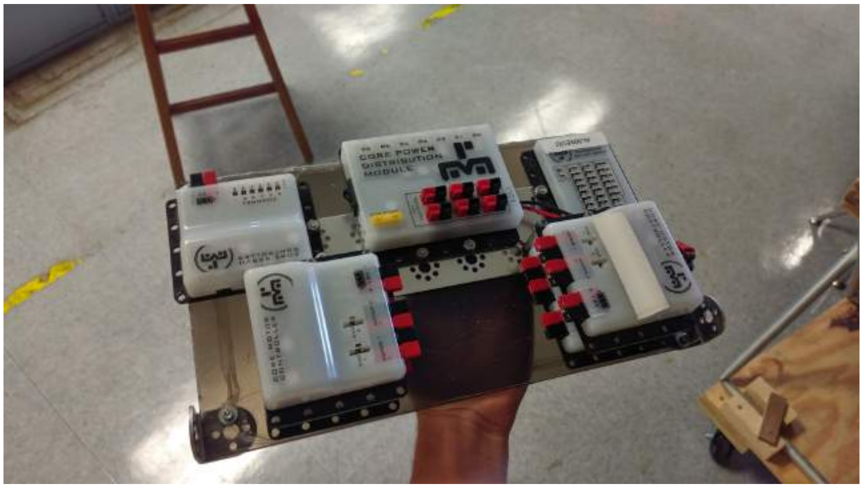


Good example: 731 Wannabee Strange, Rover Ruckus



Good examples: 8417 ‘Lectric Legends and 7244 Out of the Box Robotics

When wiring, also take the time to plan out a specific layout for your wires and how they will run throughout the robot. Take the time to lay out something like an electronics panel if necessary! When building the robot, *devote adequate space to wiring*. This could mean mounting a PVC pipe and running wires from the back end of the robot through it, or simply using velcro or zipties. **It is also recommended for electronics to be mounted on a nonconductive material such as wood to prevent ESD.**



3736 Serious Business

Specific Recommendations

Module Power

- Be aware, XT30 connectors can wear out significantly faster than PowerPole connectors. Additionally, because XT30 connectors are soldered, they can break much easier than PowerPole connections.
- Cabling for module power should be at least 14awg, if not 12awg. Keep in mind that it must be stranded, not solid, wire.

Motor Power

- Cabling for motor power should be between 16awg and 12awg. Again, stranded, not solid, wire.
- Some motors (like the REV HD Hex and Core Hex) will have removable power connectors on the back, while other motors (like the Andymark NeveRests) will have a cable permanently soldered on the back. It is much more convenient to have a connector on the back, or failing that, have a very short plug on the back of the motor. Once your wires are run and secured, taking them out won't be fun.

Servo Wires

- Using heavy-duty extension wires are recommended.
- Tape the connections between extension wires and servo wires with electrical tape. This is as the connections can become loose over time and are easy to pull out.
- The VEX Motor Controller 29 has the wrong gender connector on the 3-pin end. You are required to either use an adapter cable, or add the right connector to the wires

(recommended). **Be sure to protect the MC29, as it is fragile and prone to failure if it takes impact from another object.**

USB

- USB is generally a strong connector, but is prone to wearing out over time. Refrain from plugging/unplugging these cables more than necessary, especially on the RC/DS phones.
- USB loves strain relief. To keep disconnects low, tie down cables to leave as little loose cabling on the robot as possible.

Sensor Wires/Encoder Wires

- Sensor wires and their connectors can be incredibly fragile. Use caution when routing, and keep slack on the connector end when adding strain relief to the cable.
- The JST data connectors on the REV Expansion Hub and Control Hub have +5v, GND, and two data pins. If you are using a digital or analog sensor that does not use I²C, you can use a Y cable that gives two sensors off of one port.

Miscellaneous

- REV Grounding Strap
    - The REV Grounding Strap is currently the only legal way to ground your robot. Attach the end to the metal part of your robot frame, and plug the XT30 connector end into a free XT30 port on your robot.
- Power distribution blocks/panels
    - The REV Power Distribution Block allows teams to have more than four XT30 connectors (2 on each Expansion Hub). The block can be connected to the Servo Power Module to boost voltage for servos or provide continuous rotation for the VEX 393 EDR.
- Dryer sheet
    - Dryer sheets can be used to wipe down the robot after every match in order to reduce static buildup. This is not directly recommended by FIRST or any vendor, but our empirical evidence throughout the years suggests that it helps, or at the very least, can’t hurt to do so.
- Staticide/static spray
    - Staticide is a spray that helps to keep static off of the robot. Please be sure to spray your robot before an event and not during the event.
- Keeping things insulated,
- how to use level shifters properly and
- Common causes of static

- Every single contact point of your robot to the floor will increase the amount of static buildup. Additionally,
- Too much turning scrub (or traction when wheels try to turn). This is possible if a 4WD or 6WD (no center drop) with all traction wheels is used.
- A conductive part dragging along the ground. For example, try not to have an intake touch the ground when the robot is moving as much as possible. Foam wheels and foam rollers are a common culprit.

Sensor Glossary

Encoders/Potentiometers

- **Rotational**
  - **Absolute (+ pots)**
    MA3 (am-2899) or Potentiometer (REV-31-1155)
  - **Relative**
    E4T (am-3132) or Generic (Sparkfun)
  - **Both**
    CTRE (VEXpro) Mag Encoder (217-5049)
- **Positional**
  - **Linear Potentiometers**
    Slide Pot (Sparkfun)

Contact

- **Physical**
  - **Endstops**
    Generic (Sparkfun)
  - **Touch Sensor**
    REV (REV-31-1425)
- **Magnetic**
  - **Hall Effect Sensor**
    REV (REV-31-1462)

Optical

- **Color**
  - **Adafruit RGB**
  - **REV Color**
  - **MR Color**
- **Computer Vision**
  - **PixyCMU**
  - **OpenCV/DogeCV/EnderCV**
  - **Vuforia**
  - **TFLite**

Distance

- **ToF**
- **Ultrasonic**

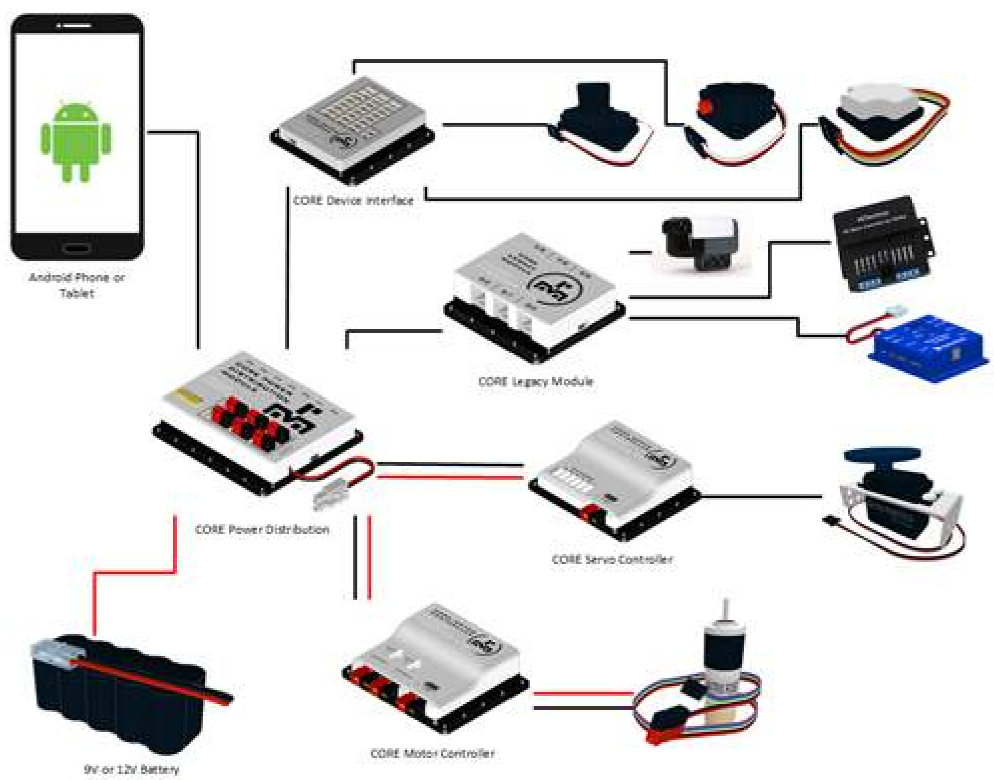
Other

- **IMU**
  - Accelerometer
  - Gyroscope
  - Compass
  - Magnetometer

Control Systems

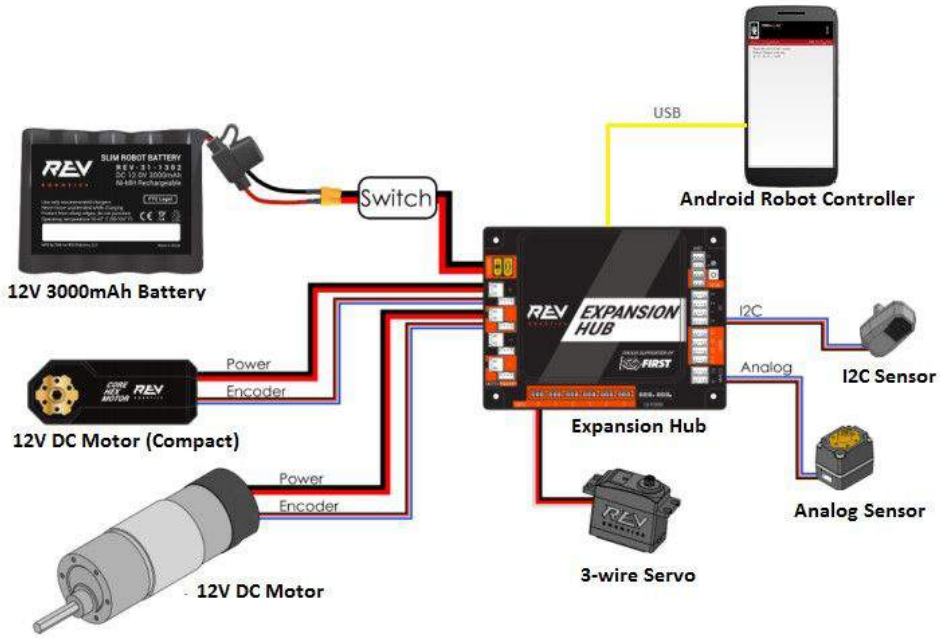
There are three possible control systems that can be run on an FTC robot legally:

- **RC Phone + Modern Robotics** (not included in the starter Electronics Kit). Modern Robotics is quickly being replaced by REV's control system, as it is less expensive and more reliable than the older Modern Robotics products. It is very likely you can ignore anything Modern Robotics related, especially if your team is starting from scratch.

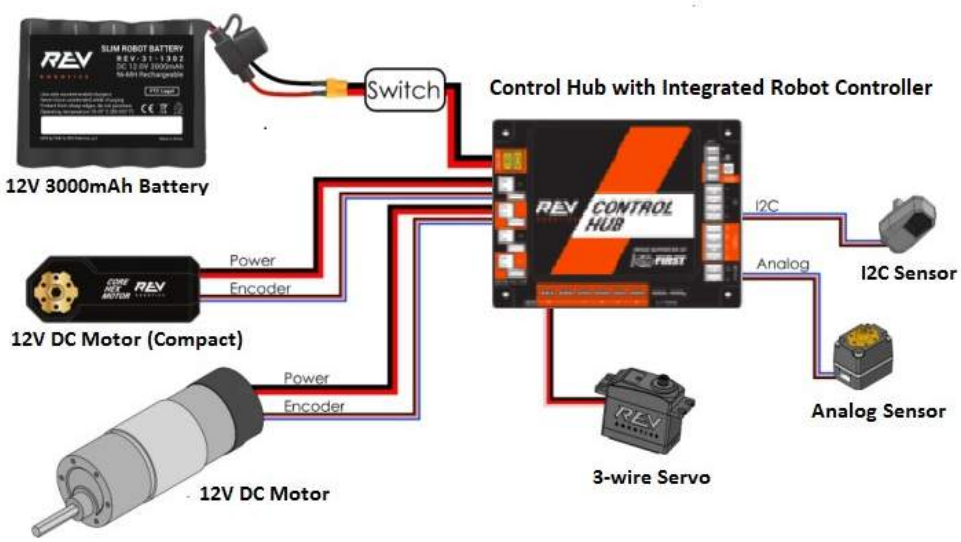


- **RC Phone + REV Expansion Hub(s)**. This is the standard control system for teams starting out in FTC. The REV Expansion Hub is reliable, as long as proper strain relief and wiring is carried out. This includes the USB Retention Mount, as well as 3D printing XT30 stress relief mounts. The Expansion Hub connects to the Robot Controller phone through the USBmini port, and the RC phone is linked to the DS (Driver Station) phone through WiFi Direct. For more information on setting up the Expansion Hub and configuring the robot, head to REV Robotics' Technical Resources page on their website.
  - [USB Retention Mount]
  - [XT30 Stress Relief]

- REV Robotics Technical Resources



- REV Control Hub + REV Expansion Hub
    - Note: The Control Hub is currently undergoing a pilot program in specific regions throughout the US. It is not legal for all FTC teams for the 2019-2020 season.



Software

Fundamental Concepts of Programming

For almost any programming language, whether Java, Python, or Blocks, there are concepts in coding that transfer across languages. These ideas are foundational when learning to program and should be applicable in FTC and beyond.

This section is primarily for people with limited Java experience. However, even if you are more experienced, it may still be helpful to skim through the section, as you might find concepts that have not yet been introduced to you.

Examples will mostly be in Java, where // indicates a comment which the program ignores and is used for people to read.

Variables

- Stores/represents information in a name
    a. Imagine a box that stores your stuff.
- You can do stuff with them:

```java
int number;  // Declaring that number will contain an integer.
number = 5;  // Setting a value so that the variable holds something.
int secondNumber = 6;  // Doing both above.
int total = number + secondNumber;  // Math.
System.out.println(total);  // Printing, it will show up as 11.
```

Java-Specific Exploratory Questions:

- If I didn't set a value for number and then I printed it, what would it print?
- What other operations can I do with number and secondNumber?
- Can I set a decimal to number? If not, what happens?
- What the heck is System.out.println(); ?
- Delete one character in the code. Remember the error (if any), and then undo it. Delete another part. How many different errors can you get?

There are different **types** of variables

- Numbers (Integers, Floats, Doubles)
- Strings (Text) or characters
- And a lot more depending on the language (Ex: Arrays)
- They help tell the program know the basis of what it should do with a variable.

```java
String coolName = "Gluten Free";
String restOfSentence = " is epic.";

// Prints out the sentence by combining the strings, unlike adding if they
// were integers
System.out.println(coolName + restOfSentence);

// Fun fact: Using + to add strings is called String Concatenation
```

Java-Specific Exploratory Questions:

- Replace the text in coolName to something else. Your name, a phone number, your favorite anime. What about emotes and copypastas? What about characters in other languages?
- Try adding a number and a string, what happens?
- Is it possible to add multiple strings and numbers together?

Important Control Structures

Be sure to familiarize yourself with basic control structures (if/else statements, for loops, while loops, and for-each loops). These control structures are by far the most commonly encountered, and thus, familiarizing yourself with these principles is extremely important (not just for FTC, but programming in general). However, there are a few control structures that are far less common that are extremely useful in FTC.

Finite State Machines and Enums

A Finite State Machine (FSM) is a control structure that executes one of several code segments depending on its state. The logic can be thought of using if-statements:

```java
if (state == 0) {
      doThing1();
}
else if (state == 1) {
      doThing2();
}
else if (state == 2) {
      doThing3();
}
```

In this example, the state variable can have 1 of 3 values: 0, 1, and 2. Depending on what value state has, different code segments are executed. However, instead of using primitive types or strings to store this information, the data type known as an Enum can be used instead.

Enums are a user-defined data structure that allow you to easily create and keep track of states in a state machine. **Each state should represent one unique, independent action of the robot.** Here's one example when a state machine is applied to a Rover Ruckus autonomous program:

This is how the state machine is defined:

```
public enum AutonomousState {
     LAND,
     DETECT_MINERAL,
     SAMPLE_LEFT,
     SAMPLE_CENTER,
     SAMPLE_RIGHT,
//... more states follow
     FINISHED
}
```

Then, you can initialize the an AutonomousState enum like this:

```
AutonomousState autoState = AutonomousState.LAND;
```

And here is how use the enum AutonomousState to create a state machine:

```
//anything inside the brackets after switch(autoState) defines the robot's
action at each state
switch (autoState) {
//case LAND is equivalent to if(autoState == AutonomousState.LAND)
     case LAND:
          land();
          if(hasLanded)
               autoState = AutonomousState.DETECT_MINERAL;
          //break tells the program it should no longer execute the code
within the LAND state and can move on to the next state
          break;
     case DETECT_MINERAL:
          detectMineral();
          if(sampleIsLeft)
               autoState = AutonomousState.SAMPLE_LEFT;
         else  if(sampleIsCenter)
               autoState = AutonomousState.SAMPLE_CENTER;
          else if(sampleIsRight)
               autoState = AutonomousState.SAMPLE_RIGHT;
```

```
            break;
        case SAMPLE_LEFT:
            driveToLeftSample();
            if (hasFinishedSampling)
                autoState = AutonomousState.TURN_TO_DEPOT;
            break;
        case SAMPLE_LEFT:
            driveToLeftSample();
            if (hasFinishedSampling)
                autoState = AutonomousState.TURN_TO_DEPOT;
            break;
//and so on
}
```

Data Structures (Arrays)

Object-Oriented Programming

Options for Programming

There are three recommended options for teams to use when programming a robot to do various tasks: Blocks, OnBot Java, and Android Studio.

The Block Programming Tool is a simple way to code simple actions for the robot. It has a colorful and lego-like design and includes sounds of block pieces of snapping together.

Using a device and a robot controller phone, connect the device to the phone's Wi-Fi (the same one that connects it to the Driver Station). Go to on your browser (preferably Chrome) and enter the address.

Advantages:

- Good for beginners: one of the easiest and simplest options to use.
- Easy to set up: requires a device (laptop, chromebook, tablet) and a phone in the Program & Manage screen
- Programs can be saved directly to the phone.
    - Changes can be made quickly
- Doesn't require an external Wi-Fi connection

Disadvantages:

- Not recommended if you're already familiar with some type of programming

    - Primarily a teaching tool for people with no programming experience
- Connecting to the robot controller's Wi-Fi network will prevent you from using anything on the internet including video tutorials and online communication
- Sacrifices flexibility and application for simplicity
- Don't ever use it on a phone, unless you're out of options. (Not phone-friendly)
    - You may make a bigger mess if you try to.

OnBot Java uses a similar method of using a browser to code and save directly to the phone. The difference being that a programming language called Java is used instead.

Advantages:
- Recommended if you're learning or have learned some programming, even better if you know a little Java.
- Greater flexibility than Block
- There are a lot more resources available in case you need help
- More applicable to the real-world than Block
- Maintains most of the advantages of Block

Disadvantages:
- Connecting to the robot controller's Wi-Fi network will prevent you from using anything on the internet including video tutorials and online communication
- Using external libraries is difficult and borderline impossible.
- Steeper learning curve than Block

Android Studio is a comprehensive Integrated Development Environment (IDE) that uses Java to program the phones. Instead of using a browser to upload code, Android Studio will compile your Robot Controller code into a .apk file (an app installer), and install that on the phone.

Advantages:
- Recommended if you're learning or have learned some programming, even better if you know a little Java.
- Much greater flexibility than Block
- Much easier to integrate libraries like OpenCV, DogeCV, Road Runner, OpenFTC, etc.
- Can use plugins like Road Runner (https://github.com/acmerobotics/road-runner)
- Can use either a USB connection to the RC phone, or a wireless connection to upload code
- Can debug in real-time
- Many resources for Java, Android Studio, and IDEA

Disadvantages:
- Connecting to the robot controller's Wi-Fi network will prevent you from using anything on the internet including video tutorials and online communication, unless you have a second Wifi adapter (cheap and easy)

- Relatively easy setup process, but time consuming and is a hefty install (3GB of files between Android Studio, ftc_app, and other libraries)
- Issues can be difficult to diagnose and solve

Kotlin is a relatively new and rapidly growing programming language from Jetbrains, the creator of the IntelliJ IDE, which Android Studio is based off. Kotlin was made to be completely compatible with Java but be easier to work with. Google recently announced it as an official android language, then announced they are going "Kotlin first" but still keeping Java support.

Advantages:
- Concise, readable, easy to edit code
- Easy to write as it takes much less code to do the same thing
- Both optional type inference and a stronger type system than Java
- Null safety
- Thread safety
- Functional programming
- Seamless integration with Java code and libraries
- Very easy to transition from Java

Disadvantages
- Not widely used in FTC yet
- New and has fewer community resources for training
- Not recommended for programmers who need large amounts of help from other teams

C and C++ are native programming languages compatible with Android. Very few teams have used C++. This is typically used for only part of the code with the majority being Java or Kotlin.

Advantages
- Fast execution for extremely resource-intensive applications.
- Supports more libraries

Disadvantages
- Rarely needed
- Very difficult to set up
- Difficult to debug code
- Very few teams can help you
- Very few online resources

LinearOpMode vs OpMode

There are two OpMode classes within the FTC SDK: OpMode and LinearOpMode. There is no difference in performance between the two, but the one you use affects how you write the program. For examples of how to use OpMode and LinearOpMode, [refer to the example OpModes in the sdk]().

LinearOpMode Methods
`runOpMode()`: Code inside the runOpMode() method will run exactly once after you press the INIT button. This is where you should put all code for the OpMode.
`waitForStart()`: This method pauses the Op-Mode until you press the START button on the driver station
`isStarted()`: returns "true" if the START button has been pressed, otherwise it returns false
`isStopRequested()`: returns "true" if the STOP button has been pressed, otherwise it returns false
`idle()`: puts the thread to sleep.
`opModeIsActive()`: returns isStopRequested() and calls idle()

OpMode Methods
`init()`: Code inside the init() method will run exactly once after you press the INIT button on the driver station
`init_loop()`: Once the code in init() has been run, code inside init_loop() will run continuously until the START button is pressed on the driver station
`start()`: Code inside the start() method will run exactly once after you press the START button on the driver station
`loop()`: Once the code in start() has been run, code inside loop() will run continuously until the STOP button is pressed on the driver station
`stop()`: Code inside the stop() method will run exactly once after you press the STOP button on the driver station

Note that unlike LinearOpMode, all methods in OpMode must be overwritten to be used.

Control Loops

Control loops are software used to operate power transmission systems (such as a drivetrain or linear slide) in a fast and controlled fashion. Not only do control loops let you run mechanisms quickly without fear of losing control, in many cases, they help preserve the longevity of mechanisms by reducing backlash on the power transmission components.

What is Error?

The first thing that must be defined when discussing control loops is the concept of error. Error is defined as the difference between where you are and where you want to be. For instance, say you tell your drivetrain to drive at 30 inches per second, but in actuality, at a time T, the drivetrain is driving at 28 inches per second. Since $30 - 28 = 2$, the error of the drivetrain's speed at this time T is 2 inches per second. In other words, at a time $t = T$, $e(t) = 2$.

PID

A PID controller (or Proportional Integral Derivative controller) is a control loop that solely uses error to control the system. PID is a form of a **feedback control loop,** or **closed loop control**. This means that data about the variable you are controlling is required in order for the loop to control that variable. In this case, information about the **error** of the system is required to control the system with a PID controller.

The Optional Calculus

The following equation represents the rigorous mathematical definition of the output of a PID controller f at any given time t:

$$f(t) = K_p e(t) + K_i \int_o^t e(t)\mathrm{d}t + K_d \frac{\mathrm{d}e(t)}{\mathrm{d}t},$$

where K_p, K_i, and K_d are constants and $e(t)$, as previously mentioned, is the error in the system. If you have no experience with calculus, don't worry; while PID is fundamentally rooted in calculus, **you do not need any calculus experience to be able to understand it, only basic algebra**. However, you are still urged to **read the rest of the section regardless of calculus experience**, as the formula alone doesn't tell you *why* it works.

Simplification of the PID formula

Here is a simplified version of the PID formula:

$$f(t) = K_p P(t) + K_i I(t) + K_d D(t)$$

All we have done is simply take the full formula and replace some terms with functions ($P(t)$, $I(t)$, and $D(t)$).

The Proportional Term

The first component of the function, $K_p P(t)$, is by far the most simple and easy to understand, as $P(t) = e(t)$. For the sake of example, let's pretend that $K_i = 0$ and $K_d = 0$ (a PID controller with only a proportional constant is known as a **P controller**). How will the system behave? Well, if the error is large, the output will be large. Likewise, if the error is small, the output will be small. Also, ideally, given enough time, the system always approaches its destination, assuming K_p is of the correct sign.

Say we apply this to a drivetrain. You want to drive a distance D, and you decide to set your motor powers using a P controller to accomplish this. In this case, your error is how far away the robot is from the desired location. As you start to drive forward, your error is large, so you drive forward quickly, which is desirable. After all, you aren't concerned with overshooting the target yet if you are far away from it. But as the robot's distance to the target approaches 0, you will start to slow down, gaining more control over the robot. Once the error is zero, ideally, the robot will stop, and you have reached your destination. If you happen to overshoot, the error will become negative, and the robot will backtrack, repeating the process.

The Derivative Term

This term, $K_d D(t)$, is intended to dampen the rate of change of the error. In other words, it tries to keep the error constant. How is this done? Well, for those of you with calculus under your belt, $D(t) = \frac{\mathrm{d}e(t)}{\mathrm{d}t}$. For those without calculus experience, it represents how fast the error is changing. Graphically, $D(t)$ is simply the slope of the error at any given time t. This slope can be calculated by keeping track of the error over successive iterations of the control loop. One iteration occurs at time t_n with an error of $e(t_n)$. At the next iteration, the time is t_{n+1} with an error of $e(t_{n+1})$. Thus, to find $D(t)$, simply find the slope of $e(t)$ given these two points.

You may be asking "why is this useful? Don't we want to minimize error, not make it constant?" And you are right to ask this question. However, a different effect is achieved when paired with the proportional term. When these two terms are used in tandem (one trying to decrease error and one trying to stop the error from changing), the system is less likely to overshoot the target. In our driving example from earlier, error changes more quickly as the robot approaches its destination. Adding a derivative term causes the robot to slow down even more than it would otherwise, thus decreasing the chance that it will overshoot the target.

Of course, in order for this to work, you need the integral term.

The Integral Term

Admittedly, the integral term is the least important term for FTC PID control loops. With a properly tuned K_p and K_d, you often can just set K_i to 0 and call it a day. However, it can still be useful in some cases. Just like the derivative term, the integral term intends to correct for overshoot. if the system thinks it reached its destination, it will stop, even when, in fact, the error is not yet 0. Perhaps the motor is no longer being supplied enough power to move. Well, given enough time, the integral term will increase the output (in this case, motor power), causing movement towards the destination. To explain without calculus, the integral term essentially sums the error over a specific interval of time. To do this, error in each loop iteration is added to a variable (in this case, $I(t)$). However, summing error this way has an unfortunate side effect: the longer the loop takes to complete one iteration, the more slowly this sum increases, which is obviously not desirable, as we don't want lag to affect how the robot moves. To compensate for this, before the error is added to $I(t)$, it is multiplied by how long the previous loop took to-complete, or $t_{n+1} - t_n$, preventing lag from making the system sum more slowly.

So say the robot stops short of the target. The P and D combination aren't strong enough to move it forward to the destination. You can either tune K_p and K_d to compensate **(this is recommended)**, or you can add the integral term to increase output **(this works too, but requires more attention and tuning to achieve the same result)**.

PID Pseudocode

```
loop:
        currentTime = getCurrentTime()
        currentError = desiredPosition-currentPosition

        P = currentError
        output = K_P * P

        I = I + currentError * (currentTime - previousTime)
        if I > maxI
                I = maxI
        if I < -maxI
                I = -maxI
        output = output + K_I * I

        D = (currentError - previousError) / (currentTime - previousTime)
        output = output + K_D * D

        previousError = currentError
        previousTime = currentTime
```

Tuning a PID Loop

Feedforward control

One less popular but equally useful control loop is the feedforward controller (sometimes unofficially referred to in FTC as the PVA controller, or Position-Velocity-Acceleration controller). For those without a physics background, velocity is the speed and direction something is moving and acceleration is how fast velocity is increasing or decreasing. Unlike PID, feedforward controllers require you to input not only where you want to go and where you are, but how fast you want to be moving at all times. Unlike feedback control loops such as PID, feedforward control loops don't require information about the variable you want to control. Instead of controlling a variable directly, it controls how fast that variable changes.

Conceptually, the controller is made up of 2 separate P controllers (remember, a P controller is made up of just the proportional term of a PID loop). Each of these P controllers are added together to create a feedforward controller.

Just like we did with the PID formula, we can define the function like this:

$$f(t) = K_v * V(t) + K_A * A(t)$$

In most FTC applications, f(t) controls the position of the output. As the name PVA suggests, the first term relates to velocity, and the second relates to acceleration. Just like in a P controller, each term contains a constant multiplied by an error term (in this case, V(t) and A(t)). However, unlike a PID controller, each term has their own setpoints and endpoints, meaning error is calculated differently for each term.

Unlike desired position, your desired velocity is likely to change throughout the control loop. After all, the entire point of using control loops are to try to create a balance of speed and control of a system. Remember, in most situations, you want to approach your destination quickly if you are far away and slow down if you are close for more control. For the sake of example, let's say $v(t)$ is a magic function that could tell you exactly how fast you should be going at any point. To calculate velocity error, subtract your current velocity from the magic function $v(t)$. This magic function can also be used to create another magic function: $a(t)$. This magic function tells you how exactly how fast the velocity should change in order to get to the next magic velocity at any specified time.

The last step is finding the magic functions $v(t)$ and $a(t)$, **which can be obtained using motion profiles** (discussed next).

Feedforward Pseudocode

```
loop:
        currentTime = getCurrentTime()

        currentVelocity = (currentPosition - previousPosition) / (currentTime - previousTime)
        currentVelocityError = desiredVelocity - currentVelocity
        output = K_V * currentVelocityError

        currentAcceleration = (currentVelocity - previousVelocity) / (currentTime - previousTime)
        currentAccelerationError = desiredAcceleration - currentAcceleration
        output = F + K_A * currentAccelerationError

        previousVelocity = currentVelocity

        ##end of feedforward code##

        previousError = currentError
        previousTime = currentTime
```

Motion Profiles

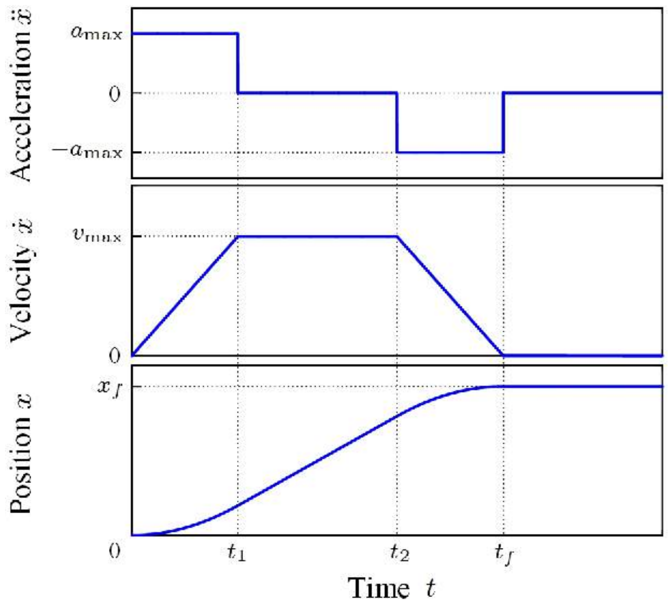
Motion profiling is a technique popularized in FRC that is starting to find its way to FTC. A motion profile is a function used to change the speed of a power transmission system in a controlled and consistent way by changing desired speed gradually rather than instantaneously. Let's illustrate this with an example: say you want your drivetrain, which is initially unmoving, to drive forward at full speed. Ordinarily, you would set all drivetrain motors to full power in the code. However, this can be problematic because even though you tell the motors to move at full speed instantaneously, the drivetrain takes time to get to full speed. This can lead to uncontrolled movements which have the potential to make autonomous less consistent and, perhaps more importantly, damage mechanisms. Motion profiling attempts to solve this issue.

Advantages:
- More controlled and predictable movements
- Reduces backlash

Disadvantages
- Can be slower

There are two main types of motion profiles: **Trapezoidal** profiles and **S-Curves**. Trapezoidal profiles accelerate the system at a constant rate, and S-Curves assume jerk (the speed acceleration changes) is constant. Given that S-Curves are not optimal for controlling 2d trajectories (such as driving) and exist to reduce slippage (which usually only occurs when driving in FTC), **Trapezoidal profiles are recommended for most FTC applications**.

Trapezoidal profiles get their name from the shape of the graph of velocity over time:



These are the “magic functions” for velocity and acceleration over time alluded to in the feedforward section.

Here is some pseudocode for a trapezoidal profile:

```
currentVelocity = getCurrentVelocity()
currentTime = getCurrentTime()
outputVelocity

##if maximum speed has not been reached
if (MAXIMUM_SPEED > absoluteValue(currentVelocity)) {
##which direction should we move in?
      directionMultiplier = 1
      if(positionError < 0) {
            directionMultiplier = -1
      }

      outputVelocity = currentVelocity + directionMultiplier *
MAX_ACCELERATION * (currentTime - previousTime)
      outputAcceleration = MAX_ACCELERATION
}
##if maximum speed has been reached, stay there for now
```

```
else {
      outputVelocity = MAXIMUM_SPEED
      outputAcceleration = 0
}

##if we are close enough to the object to begin slowing down
if (positionError <= (outputVelocity * outputVelocity) / (2 *
ACCELERATION)) {
      outputVelocity = currentVelocity - directionMultiplier *
MAX_ACCELERATION * (currentTime - previousTime)
      outputAcceleration = -MAX_ACCELERATION
}

previousTime = currentTime
```

Computer Vision (CV)

Computer vision (or CV for short) is one of the most powerful tools in the FTC toolbox, utilizing one of its most powerful sensors: the camera. In FTC, CV involves obtaining an image from your camera, then analyzing that image to find the locations of game elements or vision targets throughout the field.

Webcam interfacing

Vuforia vs TensorFlow vs OpenCV

Setting up Vuforia

Setting up TensorFlow

OpenCV filtering process

How to find robot-relative coordinates of an object

Brief summary of EnderCV

Brief summary of DogeCV

Autonomous Path Planning

Autonomous Path-Planning refers to the pre-programmed movement of the robot's drivetrain during the autonomous period. There are 4 methods of autonomous navigation that will be discussed: distance-driven pathing, coordinate-driven pathing, linear pure pursuit pathing, and spline pathing. Note that **they are presented in the order of programming difficulty, not quality**. Their quality is completely dependent on the implementation of whatever method you choose. When properly implemented, **all of the methods described can lead to incredibly consistent and high-scoring autonomous programs**, as can be seen in the examples at the start of each section.

1) Distance-Driven Pathing (example (at 170% speed) from 10091 N.Y.A.N. (Not Your Average Nerds) Robotics, Rover Ruckus)

This is by far the most popular and easiest method of autonomous path planning. When using this method, the drivetrain can move in one of two ways: driving in a straight line and turning in place. Unlike the other methods, when using this method, **the drivetrain does not turn while moving forward**.

Each step of the process entails either driving in a straight line or turning in place. In both cases, the drivetrain should move in a consistent way until a certain condition is met. Once this condition is met, the robot knows that it is done with one movement and can move onto the next movement. Here's a fairly simple example of a code snippet from a LinearOpMode:

```
while(drivetrain.getEncoderDistance(); < 1000) {
      driveForward(0.5); //50% speed
}
drivetrain.resetEncoders();
//next step goes here
```

In this example, the drivetrain drives forward at 50% speed until the wheel encoder reads 1000 encoder ticks. Once it has traveled this distance, the encoder value is reset back to 0 and the next step is ready to begin.

Encoder ticks are just one way to set conditions. Other sensors, as well as time, can be used to track a condition. However, **sensor data (i.e. drivetrain encoders) tend to be the most reliable method**. Time-based autonomous programs have done well in competitions (notably, team 9971 LAN-Bros, the winning alliance captain at the Detroit World Championship in 2019, used a time-based autonomous), but more often than not, using time-based conditions (i.e. drive forward for 2 seconds) is far less consistent than using sensors. Encoders are the most popular sensor-based option, but they are not the only sensor option. Distance sensors can be incredibly used in situations where you don't know exactly how far you want to travel. Additionally, both encoders and gyroscopic sensors (such as the sensor inside every REV Expansion Hub or the sensor inside most FTC-legal phones) can be used to track the robot's angle. Using gyroscopic sensors will increase your loop time, but in some cases, they can be more accurate than encoders.

Note that this method can be improved by implementing PID control, Feedforward control, and motion profiling. Unlike the other methods (which all but require at least one of these control loops), distance-driven pathing can be successfully implemented without using any more advanced control loops. However, if one feels up to the challenge, it is highly recommended that teams who use a distance-driven system try out some of these control loops, as they almost always improve consistency when implemented properly.

----ADD BLURB ABOUT MECANUM STRAFE ANGLES

Intermediate step: Odometry

All methods discussed after this point require the program to know the robot's coordinates and angle on the field. The process of keeping track of this location is called odometry. In order to perform odometry calculations, you must specify the robot's starting coordinates and starting angle. Then, each time your control loop iterates, the angle and coordinates are updated based on sensor data. The section can be split up into two areas: tank-drive odometry and holonomic-drive odometry.

Tank-Drive Odometry

Tank-Drive odometry is fairly straightforward. The ----------

2) Coordinate-Driven Pathing (example from 8680 Kraken Pinion, Rover Ruckus)

Unlike the other 3 methods, coordinate-driven pathing doesn't define the robot's path. Instead, it defines a series of waypoints and uses a set of functions to help the robot get to each waypoint in a specific order. Instead of following a path, the program tries to direct the robot to the waypoint as quickly as possible, regardless of starting location. The easiest way to do this is to simply drive in straight lines while turning (as seen in the example linked above), which can be done with any holonomic drivetrain. The same method can be used for tank drives as well, but the robot will not travel in a straight line. **This method requires odometry. PID is highly recommended.**

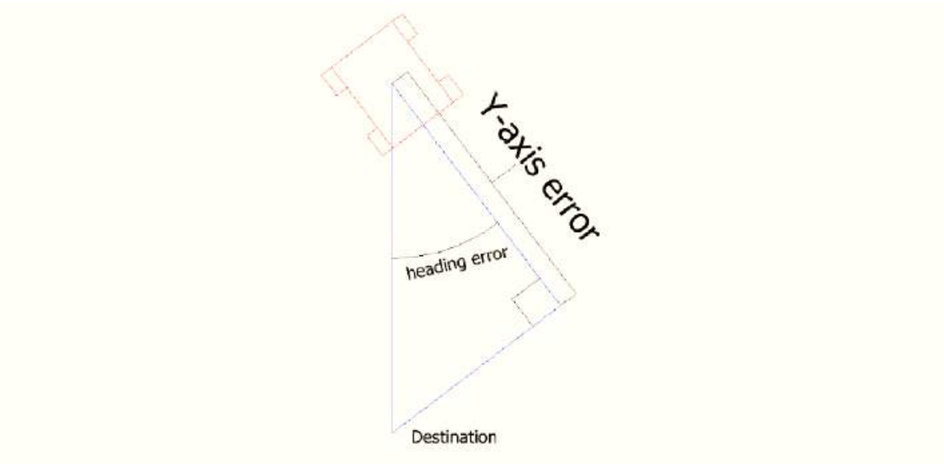
In order to travel to a waypoint, you must first calculate the distance between the robot and the waypoint. The code to do this is fairly simple:

```
distanceToWaypoint = Math.hypot(drivetrain.location.x - waypoint.x,
drivetrain.location.y - waypoint.y);
if(distanceToWaypoint < waypoint.MINIMUM_DISTANCE_AWAY)
     canMoveOn = true;
```

When this variable distanceToWaypoint is small enough, the robot knows that it has arrived at its destination and can move onto the next set of instructions. However, **if you want the robot to turn to a certain angle, another condition should be added**.

Now that the robot knows when it has arrived at its destination, its next step is to drive there. While there are many ways that this can be accomplished, when using this method, **it is highly recommended that teams use PID controllers to handle driving**. Feedforward control and motion profiling are helpful, but far from required.

Tank



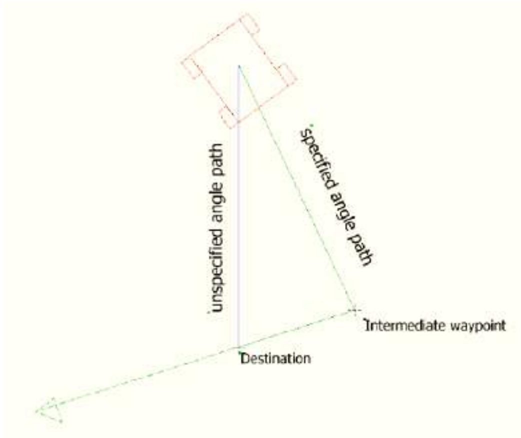

```
headingError = Math.atan2(drivetrain.location.x - waypoint.x,
drivetrain.location.y - waypoint.y)-drivetrain.location.angle;

yAxisError = distanceToWaypoint * Math.cos(headingError);
```

Using this method with a Tank-Drive requires two PID loops. The first controls the robot's forward movement and the second controls the robot's angle. The output of these two PID loops can be added together to create movement like this:

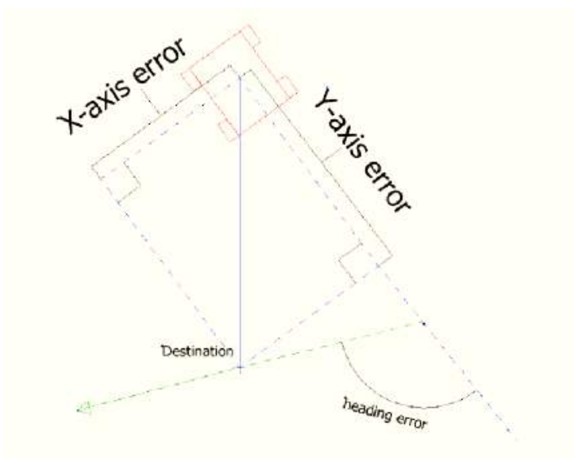
```
drivetrain.leftMotorPower = forwardLoopOutput - turningLoopOutput;
drivetrain.rightMotorPower = forwardLoopOutput + turningLoopOutput;
```

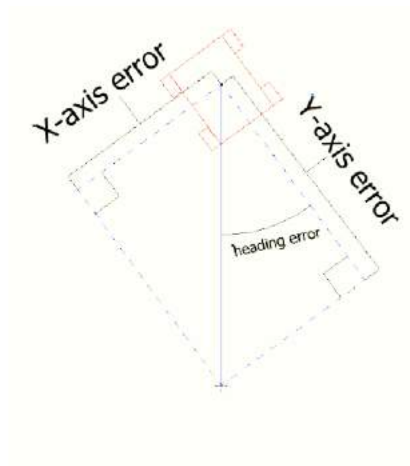
Note that when using a tank drive, you can't specify what angle the robot should be at when it reaches the destination. The best way to approach the destination from a certain angle is to travel to an intermediate waypoint before going straight to the destination.



specified angle

Holonomic





specified angle or unspecified angle

Holonomic drivetrains work a bit differently. Most notably, another PID loop is introduced: the x-axis loop:

```
xAxisError = distanceToWaypoint * -Math.sin(headingError);
```

Additionally, heading error can work a bit differently than it does when using a Tank-Drive.

Case 1: Unspecified angle
In some cases, you don't care about the angle; you just want to get to the destination as fast as possible. In that case, your heading error is defined in the exact same way as the heading error of a Tank-Drive.

Case 2: Specified angle

In some cases, you need to arrive at the destination facing a certain direction. The code to do this is actually much more simple than the code to approach the destination from an unspecified angle:

```
headingError = destination.specifiedAngle()-drivetrain.location.angle();
```

Unlike when using tank-drive, the code needed to combine the output of all 3 PID loops varies from drivetrain to drivetrain. Below is the code to combine the outputs for our 2 recommended holonomic drivetrains.

```
//mecanum
drivetrain.leftFrontPower = forwardLoopOutput + strafeLoopOutput -
turningLoopOutput;
drivetrain.rightFrontPower = forwardLoopOutput - strafeLoopOutput +
turningLoopOutput;
drivetrain.rightBackPower = forwardLoopOutput + strafeLoopOutput +
turningLoopOutput;
drivetrain.leftBackPower = forwardLoopOutput - strafeLoopOutput -
turningLoopOutput;

//H-drive
drivetrain.leftMotorPower = forwardLoopOutput - turningLoopOutput;
drivetrain.rightMotorPower = forwardLoopOutput + turningLoopOutput;
drivetrain.strafeMotorPower = strafeLoopOutput;
```

Linear Pure Pursuit (example from 11115 Gluten Free, Rover Ruckus)

Despite its popularity in FRC, in FTC, pure-pursuit is the least common method listen in the guide. However, it is very powerful and

Spline Pathing (example 8103 Null Robotics, Rover Ruckus)

Brief overview of Roadrunner

Brief overview of SpeedyDoge

Coming Soon.

Automating Tele-Op

Vendor List

- Actuonix (www.actuonix.com)
  - Actuonix sells linear actuators and linear motion components. Expensive, but robust.
  - Teams can apply for a FIRST sponsorship.
- AndyMark (andymark.com)
  - AndyMark sells the official game field and game sets, as well as individual game parts and the SoftTiles foam tiles.
  - AndyMark also sells NeveRest and NeveRest Sport motors, TileRunner, compliant, stealth, and mecanum wheels, as well as many other items.
- goBILDA (gobilda.com)
  - goBILDA sells its own build system, complete with Yellow Jacket motors, channel, motion components, and battery. Note that the cheapest batteries are found here, the MATRIX 12V Batteries are $39.99, with the team discount they are $29.99, nearly half the price of the $50 batteries sold elsewhere.
  - Teams can get a 25% Team Discount from goBILDA.
- McMaster-Carr (mcmaster.com)
  - McMaster-Carr sells hardware and raw materials in bulk quantities. They stock nearly every type of bolt, screw, and nut possible, as well as washers, bearings, springs, etc. Purchase from them for bulk quantities of hardware, as well as the times you need a very obscure part.
  - Don't be turned off by the hidden shipping. Generally, McMaster-Carr's shipping is around the same price as other vendors, and shipping is usually next-day.
- MiSUMI (us.misumi-ec.com)
  - MiSUMI is a Japanese company specializing in industrial and manufacturing components. They sell bulk 15mm anodized extrusion similar to the REV Robotics extrusion. The 15mm extrusion can be cut to length as well.
  - MiSUMI also sells aluminum drawer slides that are popular for linear extensions. They are available in different lengths, but the most common is 400mm.
- Pitsco (pitsco.com)
  - Pitsco sells the Tetrix kit with channels, TorqueNado motors, and their own motion system.
- REV Robotics (revrobotics.com)
  - REV Robotics sells the REV build system, which is an extrusion-based ecosystem complete with motors (HD Hex, HD Planetary, Core HEX), extrusion, servos (Smart Robot Servo), brackets, and battery.
  - REV also sells the control system for FTC (Expansion Hub and Control Hub).
  - REV offers various sensors (Magnetic Limit Switch, Color Sensor, Touch Sensor, Distance Sensor, Potentiometer, etc.)
  - Additionally, electronic components such as the Servo Power Module, SPARKmini, or Blinkin may be purchased.

- Servocity/Actobotics (servocity.com)
  - Servocity sells the channel-based Actobotics build system with robust motion and structure. They also offer the Servoblock, a highly recommended part.
  - Servocity sells X-rail extrusion and linear extension kits, as well as the lead screw actuator kit.
  - Servocity sells a wide range of servos, from Hitec to Futaba, at all price points.
  - Teams can get a 25% team discount from Servocity, making it an affordable vendor.
- VEX/VEXPro (vexrobotics.com/vexpro/ftc)
  - VEXPro sells parts angled toward FRC use, but many of their parts, such as Thunderhex, can be made compatible with FTC robots.
  - VEX sells the 393 Motor, as well as the Motor Controller 29 in order for it to adapt to the Expansion Hub.
  - VEX offers the VersaPlanetary motor which allows teams to customize a specific gear ratio for their needs.
- West Coast Products (wcproducts.net)
  - Also known as WCP, West Coast Products sells products exclusively aimed toward FRC use. However, much like VEXPro, many of their parts can be used in FTC, especially in custom robots. In fact, VEXPro and West Coast Products sell many of the same parts.

Vendor Identification

When you read some Bill of Materials (BOMs) or part numbers that are references in this guide, it might be confusing to find what vendor the part you needs comes from. The obvious thing to do is Google the part number, and generally it can identify what you want and where it comes from. But if that fails, here are some quick tips to identify the vendor selling what you need.

Prefix Identification:
AM- signifies **AndyMark** (i.e. AM-0447)
REV- signifies **REV Robotics** (i.e. REV-31-1155)
WCP- signifies **West Coast Products** (i.e. WCP-0117)
217- signifies **VEXpro** (i.e. 217-6194)
276- signifies **VEX EDR** (i.e. 276-2193)

SKU Identification:
Actuonix: 3-4 digit SKU, alphanumeric, in the form of XX00 or X00 (i.e. PQ12)
goBILDA: 12 digit SKU, numerical, in the form of 0000-0000-0000 (i.e. 3213-3606-0001)
Pitsco/Tetrix: 6 digit SKU, **W** + 5 numbers, in the form of **W**00000 (i.e. W44260)
Servocity/Actobotics: 6 digit SKU, numerical, in the form of 000000 (i.e. 615190)

Useful Resources

Here is a page dedicated to useful resources around the interwebs. Enjoy!

- Official *FIRST*Inspires Resources
  - Game and Season Information
  - 2019-2020 Game Manual Part 1
  - FTC Forum - USFIRST
  - Robot Inspection Checklist
  - Field Inspection Checklist
  - DIY Field Perimeter Build Guide
  - Robot Building Resources
  - Robot Programming Resources
  - Team Management Resources (Budget, Engineering Notebook, etc.)
  - Mentor Manual
- Building Resources
  - Simbotics FRC Drivetrain Design
  - FTCKey - FTC wiki (slightly outdated)
- Vendor Resources
  - REV Robotics Technical Resources
- YouTube
  - 8644 Brainstormers Tips and Tricks
  - goBILDA
  - Pitsco/Tetrix
  - REV Robotics
  - ServoCity/Actobotics
- Discord
  - https://discordapp.com/invite/8v3cbkj
  - The FIRST Tech Challenge Discord server is a discussion-based community server that has teams ranging from first-year rookies to Winning Alliance Captains at the World Championships. It also has a channel for direct access to vendors.

Glossary

Many terms used in GM0 that are unfamiliar to teams may be found here.

- Anderson PowerPole
    - Anderson PowerPole is a connector used by AndyMark on their NeveRest motors. PowerPole connectors are very reliable and recommended for teams. In addition, there are adapters available to other systems.

- Ball Bearing
    - Ball bearings refer to bearings with steel balls arranged in a circular fashion. This allows rotation of an element with less friction than a bushing, primarily because the surface area (or contact area) is much less than in a bushing. Bearings are definitely recommended for drivetrain and high speed usage. Bearings are used the Actobotics and REV kits, and are commonly sold by most robotics vendors.

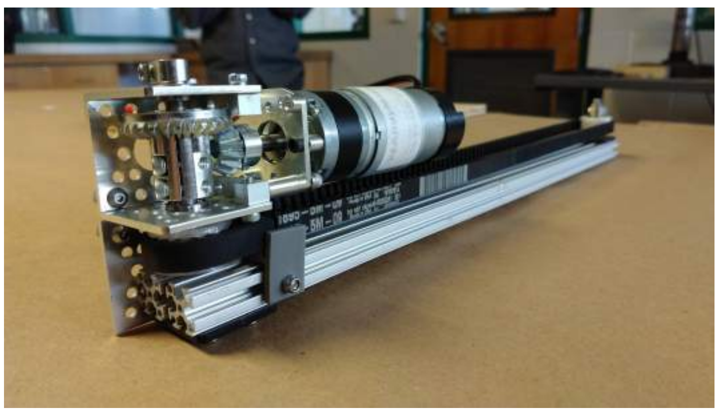
Actobotics dual ball bearing hub

- Banebots gearbox
    - The Banebots gearbox is a heavy-duty gearbox that can be attached to RS-555 series motors. It has high gearing options for teams to choose from if they wish to build a mechanism such as an rotating arm.

- Bare motor
  - A motor that does not have a gearbox attached to it.

A bare NeveRest Classic Motor (am-3104)

- Bevel gear
  - Bevel gears are gears that transfer power along different axes, which are perpendicular to each other. Bevel gears are generally considered more inefficient than regular gears. However, bevel gears can be very useful, especially in areas of limited space where the motor can be placed perpendicular to the element it is driving, and not in the same plane.



3736 Serious Business, Rover Ruckus

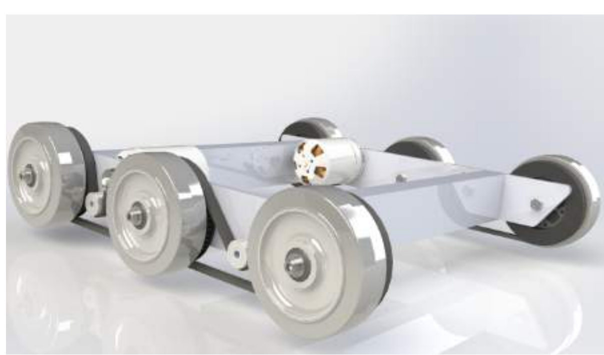
- Bore
  - The bore refers to the shape of the opening that the shaft is inserted into. For example, the bore for a 5 mm hex shaft is the hexagonal shape. “Stripping the bore” means that over time, the bore will lose its hexagonal shape, and become close to a circular shape, rendering the bore (and subsequently, the part) useless.

- Box tube
    - Box tube is aluminum shaped into hollow square or rectangular profiles. Commonly used in FRC, box tubing is seen less in FTC; however, small box tubing can be used for drivetrain or elevator purposes. Generally, we recommend new teams stick to kits unless they are prepared to tackle custom mechanisms.

- Bushing
    - A bushing is primarily mounted on the outside of a shaft. It rotates in a pillow block, which holds the bushing. Generally, both are made out of a low-friction material such as Delrin or bronze. Bushings are less efficient than ball bearings because it has a larger surface of contact, but are acceptable for low-load situations or low-budget teams.

REV Bushing (left) and pillow block (right)

- Cantilever
    - A cantilever refers to when an object (usually a shaft) is only supported on one side. This is not recommended in general, unless you know what you are doing. Using 2 or more points of support should be the rule, and cantilevering is the exception. However, if it is necessary, using a thicker shaft and plate can make this feasible.



8103 Null Robotics, Sanford's prototype

- Center to Center (C2C)
  - Center to center refers to the distance between the centers of a pair of sprockets, pulleys or gears. This will affect chain/belt tension and gear meshing, so calculating this correctly is essential.

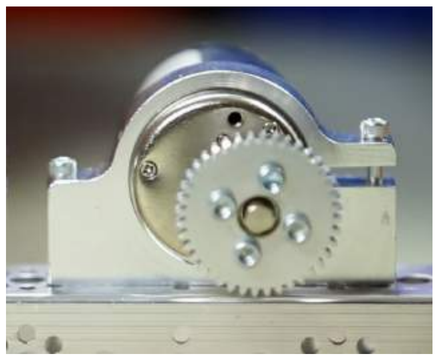
- Chain
  - Refer to "Sprocket" for more information.

- Chain breaker
  - A chain breaker is a tool used to "break" the chain by pushing out the pin in the chain link, and reconnects it by reversing the operation. **We highly recommend purchasing the DarkSoul chain breaker if you plan to use chain.**

- Channel (C-Channel)
  - Channel is aluminum that is in the profile of a C. Channel, along with extrusion, is the most common structural build element in FTC, and is found in Tetrix, Actobotics, and goBILDA kits. Channel is fixed pitch, which means that there are pre-drilled holes that limit mounting to finite locations. It can be used to easily construct drivetrains; however, be aware that gear and chain mesh may not be perfect with channel.

- Churro
  - Churro is a ½" or ⅜" hex product sold by AndyMark. It has a bore that is easily tapped to accommodate ¼-20 and ¼-28 bolts, and is commonly used as a large standoff. It is light and cheap compared to other hex products. A ⅜ hex version also exists. Using churro as shaft is highly discouraged, as it is slightly undersized as well as prone to twisting.

- Clamp mounting
  - Clamp mounting refers to securing a motor primarily by using friction instead of screws attached to the motor itself. This is generally discouraged as the motor can become loosened over time. One tip for clamp mounting is to use friction tape around the surface of the motor that is clamped down so that it will have less chance of moving around.



TETRIX clamp mount and v1 motor

- Clamping hub
  - A clamping hub is used to fixate part such as sprockets or gears on shafts. It is also used to prevent shafts from moving laterally. Clamping hubs are recommended over shaft collars because clamping hubs have more contact area than a set screw.

- Colson Performa Wheel
  - The Colson wheel, sold by VEX robotics and various sellers, is one of the premier traction wheels for FTC drivetrains. Offered in many different thicknesses and diameters, the Colson wheel can fit nearly any type of skid-steer drivetrain. The rubber on the Colsons provide great traction with impressive durability. It is sold in a ½” hex bore size, so teams will have to use Ultrahex or similar product in order to use Colsons.

- Commercial Off the Shelf (COTS)
    - Commercial Off the Shelf parts refer to parts that teams can purchase physically or through an online retailer. **FTC teams are limited to one degree of freedom (with some exceptions) to COTS parts.** Therefore, buying a drawer slide is an allowable part, as there is only one degree of motion, but purchasing a multi-axis arm isn't. However, teams can buy individual parts and assemble them together into a mechanism that has >1 degree of motion. This doesn't apply to the TileRunner or Actobotics' lead screw kit.

- Compliant Wheel
    - The compliant wheel, sold by Andymark, is a flexible rubber wheel that is primarily used for intakes. **It is not designed for use in a drivetrain**. The available bore options are ½" and ⅜" hex, as well as 8mm round with a TETRIX hole pattern (4 inch only). As with the compliant wheels, durometer (hardness of rubber) affects both traction and longevity, sacrificing one for the other. However, in the case of intakes, a lower durometer is recommended to have maximum grippiness for intaking game elements. Keep in mind that elements may get jammed at unfavorable angles in your robot. An alternative to the compliant wheel is the **West Coast Products Flex Wheel**. These wheels, while far less common, serve the same function as compliant wheels, but are generally considered more durable. However, for sizes greater than 2", you will need to design and manufacture a custom hub in order to create a mounting point.

2 inch (left) and 4 inch (right) compliant wheels

- Compound gearing

- ○ Compound gearing refers to multiple reductions in order to transmit power from A to B. This is used when a specific reduction might be needed, or for space issues. Compound gearing can be achieved by placing two gears or sprockets of different sizes on one shaft.

- Core Hex Motor
  - The Core Hex Motor, sold by REV, is different from the standard RS-555 series motors that are generally used by FTC teams. It features a 90 degree orientation and does not contain an output shaft. Thus, teams will have to cut 5 mm hex shaft to length as needed. The Core Hex motor has a slow gear ratio (72:1), and is not as powerful as the RS-555 series motor. **We advise teams to go *against* the Basic Bot Guide provided by FIRST, as Core Hex Motors should NOT be used to power drivetrains.**

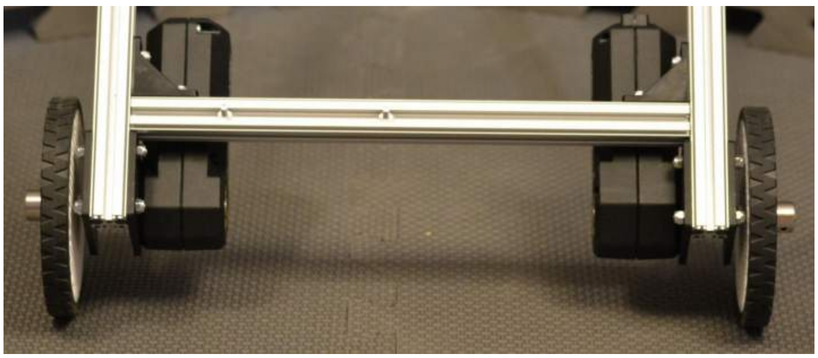
- DarkSoul
  - The DarkSoul chain breaker, designed for motorsport application such as BMX, is compatible with the FTC standard #25 metal chain. Analogs of this include the REV #25 chainbreaker and the VexPro #25 chain breaker. REV and VexPro claim that they have made FIRST specific improvements to this chainbreaker. **It is highly recommended that teams purchase this chain breaker if they are planning to use chain.**

DarkSoul #25 chain breaker

- Dead axle
  - A dead axle refers to an axle that **intentionally** does not spin. Instead, bearings are mounted directly to the moving part, such as a wheel in a drivetrain. Power is

transferred with a sprocket, pulley or gear that is also directly mounted to the moving part. This eliminates the need for the axle to transfer torque, and also eliminates the need for hubs. Additionally, the axle can be used for structural integrity, as it is rigidly mounted.

- Direct drive
    - Direct drive refers to mounting a wheel directly on the shaft of the drivetrain motor. This means that there cannot be any change of gear ratios between the motor and wheel. **Direct drive is not recommended because shock loads transfer easily between wheel and gearbox, and can break the gearbox, especially in drivetrain use.**



Basic Bot Guide - REV

- Disconnect (DC)
    - A disconnect is when, for any reason, the robot is not able to be controlled from the gamepad. This can happen for many reasons - static buildup on the robot, a loose cable, or an error in code. Generally, most DCs are caused by improper wiring, so wire stress relief is encouraged for all teams (USB retention mount). They can also be caused by WiFi disconnects, or an ESD (electrostatic discharge) shock to the electronics.

- Driver Station (DS)
    - The Driver Station phone refers to the phone that is used by the drive team and connects to the gamepad(s).

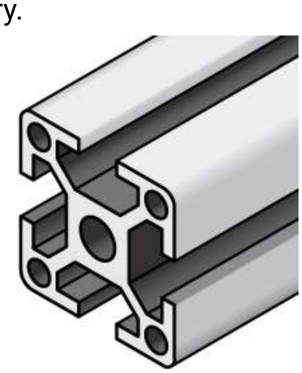
- Durometer
    - Durometer refers to the hardness of rubber. Having a high durometer translates to a harder rubber surface, more durability, but less traction. A low durometer means a softer rubber, worse durability, but improved traction.

- Encoder
    - An encoder refers to a device that tracks (generally) rotational movement around an axis. There are both absolute and relative encoders. An absolute encoder will

report at exactly what angle the shaft is compared to its absolute “zero”. A relative encoder will report how far the shaft has rotated since it started tracking (for example, when autonomous starts). Encoders are used to help find the position of where the robot, or one of its mechanisms, is.

- Expansion Hub
    - The REV Expansion Hub is a hardware controller that interfaces with the Android phone. It includes XT30 ports for power input and output, 4 motor ports with encoder, and 6 servo ports, as well as Mini USB for the Android phone.

- Extrusion
    - Extrusion is aluminum shaped into slotted profiles able to accept certain types of hardware. For FTC, the most common is the 15mm extrusion, used in the REV and Misumi products. 15mm extrusion accepts M3 bolts and nuts (note that only regular M3 nuts can fit inside the slot, not locknuts). Extrusion is not a fixed pitch system, allowing teams to adjust components as they wish. This makes it simple to achieve correct tension and put mechanisms where channel would limit mounting. The adjustability of extrusion is especially useful in precise situations, such as intake geometry.



- Face Mounting
    - Face mounting refers to mounting the motor by affixing the motor directly to the mount using bolts. This is the preferable way of mounting the motor because it is less likely to loosen over time, especially with the use of Loctite on the bolts. **It is advisable that 4-6 bolts be used to face mount for redundancy.** Additionally,

there is no way that the motor might rotate and cause a loss of tension in belts or chain.

REV v2 motor facemounted

- Gauge (wire)
    - Wire gauge refers to the diameter of wire. AWG stands for American Wire Gauge, the general system used in the US. The larger the gauge number, the smaller the wire diameter. Generally, servo wires are 22 AWG and motor wires are 18 AWG.

- Gear
    - A gear is a machine part that has cut teeth, usually written in the form “numberT” (e.g. 32T, 86T). Its purpose is to transfer power from the motor. Gears can be made in different materials. The most common is aluminum, while Delrin plastic may also be used.

56T REV aluminum gear

- Gearing up
    - Going from a higher gear ratio to a lower gear ratio. (i.e. 20:1 → 10:1).

- Gearing down
    - Going from a lower gear ratio to a higher gear ratio. (i.e. 10:1 → 20:1).

- Gearmotor
    - A component consisting of only one motor and one gearbox.

- Gear Ratio/Gear Reduction

- In any rotational power transmission system (typically involving motors and servos in FTC), a gear ratio defines both the number of rotations of the system's input and the number of rotations of the output. For instance, a Neverest 20 gearmotor consists of an unmodified Neverest motor and a planetary gearbox that has a gear ratio of 20:1 (or, when spoken, "20 to 1"). This means that in order for the output shaft of the gearbox to rotate 1 time, the input shaft of the motor must rotate 20 times. Gear ratios are one of the most important design considerations about a power transmission component. Any FTC motor or servo has two properties: speed and torque (or rotational force). These two properties are inversely proportional, meaning that increasing speed decreases torque, and vice versa. For instance, if one wishes to make a mechanism faster at the expense of torque by doubling the speed of that 20:1 gearbox, they would decrease the gear ratio by a factor of 2. Since 20/2 = 10, the new desired ratio would be 10:1 (this is referred to as **gearing up**). However, if one wishes to double torque instead, making the system more powerful and robust at the expense of speed, they would increase the gear ratio by a factor of 2, leaving them with a 40:1 ratio (this is referred to as **gearing down**). The most common ways of gearing up or down are using gearboxes, gears, sprockets and belt-driven pulleys, all of which exist in various sizes.

- Grounding Strap
  - The REV Grounding Strap is used to ground the metal frame of the robot to the XT30 port of the Expansion Hub. It is currently the only legal way to ground your robot.

- HD Hex Motor
  - The HD Hex motor, sold by REV Robotics, is a RS-555 series motor with spur gear and planetary gearbox options. The motor has a 5mm hex output shaft compatible with REV's motion system.

- High Strength Hex Hub
    - REV’s hex hub is a steel attachment whose purpose is to prevent the bore wearing out over time. The strengthener fits on the 5 mm hex shaft and into the gear, sprocket, or wheel. **It is highly recommended that all teams use strengtheners on all driven wheels, gears, or sprockets to prolong its longevity and prevent bore stripping.**

- HTD belt
    - HTD belt is a type of synchronous timing belt commonly used on drivetrains. It is available in different widths to accommodate different sized pulleys. The most common is 3mm and 5mm belt, which can be purchased from various online vendors.

- Idler gear/sprocket/pulley
    - An idler gear, sprocket, or pulley is one that is purposely not used for driving anything else on the shaft. The purpose of this idler is, in the case of gears, to transfer power to another direction. For chain and belt, idlers are more common, and are usually adjustable to maintain tension.

- JST-PH
    - JST-PH is a type of connector. For FTC, the 3-pin and 4-pin options will be used most often. For the 3-pin connector, it is used for RS-485 connections

- JST-VH

- JST-VH is a type of connector used by FTC motors to interface with the REV Expansion Hub. It is keyed and locks into place for improved reliability.

- Lead screw
    - A lead screw is very similar to a threaded rod. It is used for high load and high torque application such as hanging. However, due to the nature of the threaded rod, lead screws are generally quite slow compared to linear slides. The speed of a lead screw is determined by two factors. The first is how fast the motor outputs, and the second is the number of threads per inch (TPI).

- Linear actuator
    - Linear actuators are basically servos that translate its output into linear motion, instead of rotational motion. Linear actuators are rarely used in FTC due to its prohibitive cost, but they may have some uses in special applications.

- Locknut
    - A locknut is a nut that resists vibration by the nyloc inside. Nyloc is a type of plastic that holds the bolt securely on to the nut when it is screwed in. It is advised that teams purchase locknuts instead of regular nuts as FTC mechanisms often become loose over time.

- Loctite
    - Loctite is thread locking fluid used so that bolts do not come loose under use and vibration. Loctite should be applied to the threads of the bolts. There are two types of Loctite: blue, which is removable, and red, which is permanent (and we mean it). **It is highly recommended that teams use Loctite on all motor and servo mounts, as well as any mechanism prone to vibration.**

- ○ NOTE: THE BOTTLE COLOR AND THE FLUID COLOR ARE REVERSED. When we refer to the “color”, we mean the fluid color. Blue loctite usually comes in a red bottle.

Blue (removable, in red tube);
Red (permanent, in blue tube)

- Mecanum wheel
  - ○ Mecanum wheels are a special type of wheel that enable maneuverability and holonomic strafing as opposed to traditional wheels. They consist of a series of rubber rollers rotated 45 degrees to either the left or right. In a conventional mecanum drivetrain, running the wheels on one diagonal in the opposite direction to those on the other diagonal causes sideways movement. Combinations of these wheel motions allow for vehicle motion in any direction with any vehicle rotation (including no rotation at all).

- Micro USB On The Go (OTG) cable
  - ○ The Micro USB OTG cable connects the driver station phone with the Logitech controller that the driver uses in order to control the robot. **It is recommended that teams purchase a couple spares due to faulty OTG cable connections and its low price.**

- NeveRest motor
    - The NeveRest motor, sold by AndyMark, is a RS-555 series motor that is available in spur gear and planetary options. It has a 6mm D-shaft output compatible with Actobotics motion system.

- Omni[directional] wheel
    - Omni wheels, sold by many different vendors, are a special type of wheel that prioritizes mobility and strafing (moving laterally) over traction or front-back movement. It is similar to mecanum in that omni wheels have rubber rollers that rotate perpendicular to the plane of the wheel. Thus, the robot can move sideways (although the robot is not powered in the sideways direction). It is also utilized as a low-friction wheel in 4 wheel, 6 wheel, and 8 wheel drivetrains instead of having corner traction wheels. Furthermore, X-drive utilizes four omni wheels, though traction is at a minimum.
    - A mecanum wheel is technically an omnidirectional wheel, but when generally referred to, an “omni wheel” has rollers rotated 90 degrees to the rotation of the wheel, where a mecanum wheel is generally 45 degrees.

- Packaging
    - Packaging refers to the relative size and location of components on the robot. Generally, you want to design and locate (or package) components in the most space-efficient way you can
- Parallel plate drivetrain

  - A parallel plate drivetrain is a drivetrain that has drive pods that consist of 2 plates spread apart with wheels and drive transmission in between them. These plates can be anywhere from 1" to 5" apart, depending on the space requirements of the wheels and drive system. Generally, a pod width of 3" or less is desired to maximize the space between the drive pods for mechanisms such as an intake.

- Pitch diameter
  - Pitch Diameter (PD) is the imaginary circle that mates with any other gear's pitch diameter when the gears are properly spaced. The pitch diameter will always be smaller than the outside diameter of a gear.

- Planetary gear
  - Refer to the Motor and Servo guide for more information.

- Polyurethane tubing
  - Polyurethane tubing is a type of clear tubing that is stiffer than rubber or latex tubing. It is sold in different outer diameter sizes and thicknesses, and can be fitted inside of surgical tubing to make it stiffer.

- Punch tubing
  - The REV Punch Tube is 15 mm aluminum tubing that allows teams to use the 15 mm REV building system without having the disadvantages of extrusion, such as that parts come loose over time. With punch tubing, teams must pre-drill holes and attach, unlike extrusion, where teams can slide and adjust mechanisms. Thus, it is recommended that teams use extrusion in prototyping/iterative design, and use punch tubing on the final iteration of their robot to save money. Punch tubing is compatible with the Metric Step Drill and 1/8" or 3.2mm pop-rivets.

- Ring gear

    - Refer to the Motor and Servo guide for more information.

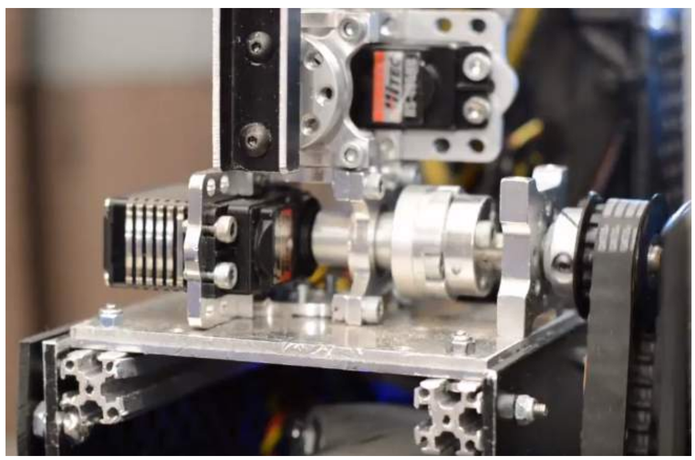
- Robot Controller (RC)
    - The Robot Controller phone refers to the phone that is on the robot and is connected to the Expansion/Control Hub via the Micro USB cable.

- RS 550 series motor
    - The RS-555 series motor is the standard motor in FTC. It forms the base for the Andymark NeveRest, REV HD Hex, and goBILDA Yellow Jacket motors.

- Servo
    - A servo is a small DC motor attached to servo gears that is very finely controllable. Servos are used in FTC for high-precision applications that are low-load - for example, opening a trapdoor for balls to fall through. The output has splines, which are the rigid teeth that are on top of the servo. Commonly, FTC uses 24 and 25 tooth splines, meaning there are 24/25 teeth around the circumference of the output shaft. There are many different types of servos legal for use in FTC - for information on how to choose the right servo, refer to the Motor and Servo section.

- Servoblocks

- ○ Servoblocks, sold by Servocity/Actobotics, are a way to mount servos to the Actobotics system. It is by far the best way to mount servos because it decreases the load on the servo spline, which is the weakest part of the servo. This is because under load, the servo spline teeth can easily become stripped, rendering the servo unusable. While Servoblocks are not cheap, they are **one of the best investments** for teams to pursue.



10030 7Sigma Robotics, Relic Recovery

- Servo Power Module (SPM)
  - ○ A Servo Power Module is a device made by REV Robotics that boosts the voltage that the Expansion Hub provides to a servo. The Hub's output is 5V @ 6amps, and the SPM boosts the voltage to 6V and up to 15amps. This is important for servos under high load conditions such as the Savox servo, as well as the VEX 393 motor.

- Set screw
  - A set screw is generally a hex socket screw that is used to fasten parts such as sprockets or gears to a shaft, or to fix a shaft in place so that it doesn't move around. Due to the hex socket, allen keys must be used to tighten and loosen set screws. **Set screws are not recommended for drivetrain and high-load applications** since there is very little surface area in contact with the shaft (only the tip of the screw). This makes the set screw likely to damage the shaft. Therefore, set screws can become loose very easily. If set screws must be used, then it is imperative to use Loctite to reduce the chance of it shaking loose. **Clamping hubs are much preferred to set screws, as clamping hubs apply pressure to the whole diameter of the shaft, as opposed to just one point.**

- Shaft
  - A shaft is a piece of shaped metal used in power transmission. Shafts are the primary method to transfer power from motor to wheel. Generally, shafts are made out of steel, so **do not use a bandsaw to cut a shaft**. Rather, use a hacksaw, as hacksaw blades can cut through steel. There are different kinds of bores in FTC, which are listed below.
    - Round shaft
    - D-shaft - has a flat part for set screws, otherwise round
    - Hex shaft - six sided shaft
    - Keyed shaft - round shaft which has a keyway (a slot) through the shaft
    - Square shaft - commonly used in VEX products

- Shaft collar

    - A shaft collar, which has a set screw, is fitted on to a shaft in order to secure parts. Refer to “Set screw” for more information.

- Spacer
    - A spacer is used for keeping parts aligned with each other in separate shafts. Generally, spacers are used because there isn’t space for a clamping hub or shaft collar, as those take up more space. However, spacers are very low-profile and hug the shaft closely. Spacers can be purchased in different configurations, from 1 mm to 15 mm. Custom spacers can also easily be 3D printed.

1.5 mm spacer (left) and 15 mm spacer (right)

- Sprocket
    - A sprocket is a mechanical part that transfers power through its cogs, which fit into chain. It is similar to a gear, except that instead of meshing with another gear, the sprocket meshes with chain. The cogs have the same system as gear teeth, using “numberT” (e.g. 32T or 86T). Chain is sold in both metal and plastic varieties. #25 roller chain is usually metal, while 8mm chain used in FTC is usually plastic but can be metal. Plastic #25 chain is not recommended for higher load applications, such as a drivetrain.

Delrin 20 Tooth #25 sprocket

- SRS Programmer
    - The REV SRS Programmer is a device that will send a special data signal to the REV Smart Robot Servos to control their electronic endstops, as well as the

continuous rotation mode of the servo. It can also be used as a servo tester for other servos.

- Standoff
    - A standoff is a fastener with two threaded ends and usually has a hex profile to be used with a wrench. These ends are usually female threaded, meaning that they can have a screw threaded into them. This is usually a more compact alternative to a long screw and spacers, and can be used to space things out as well as fasten them. Custom standoffs can be made out of hex stock, such as AndyMark Churro. Standoffs are usually used in drivetrain purposes, such as in parallel plate drivetrains, where the plates must be separated and supported by standoffs at equal distances.

- Stealth Wheel
    - The stealth wheel, sold by Andymark, is a typical traction wheel used by many FTC teams from new to experienced. Andymark sells the 2” diameter and 4” diameter, but most teams use the 4” diameter option for drivetrains. It is available in different durometers (hardness of rubber) so that teams may select the option that best suits them. A lower durometer (such as 35A) means more traction at the cost of longevity. For this reason, a medium durometer such as 50A (blue) or 60A (black) is recommended. Generally, 50A wheels can survive a year’s worth of driving and use, but it is recommended to swap them out mid-season unless they are cleaned regularly. Stealth wheels are available in different bore sizes, such as ½” hex, ⅜” hex, 8 mm round, and 5 mm hex. With the 5mm hex option, it is highly recommended to use the hex hub strengthener from REV in order to prevent the bore from stripping out.

4 inch stealth wheel

- Strafe

    - Strafing is the act of moving sideways or laterally (somewhat similar to drifting). It is possible with omni or mecanum wheels, and not possible with traction wheels.

- Surgical Tubing
    - Surgical tubing is generally latex or rubber tubing. Its most common use case is in active intakes, and has been popular among teams for many seasons. Surgical tubing has a hollow center and is sold in different diameters and wall thicknesses. Teams can experiment with different kinds of surgical tubing, as well as adding polyurethane tubing in order to make the tubing more stiff.

- Thunderhex
    - Thunderhex is aluminum hex shaft that comes in ⅜” and ½” hex sold by VEX robotics. It has a center bore that can be tapped. Its most notable feature is rounded corners, which allow it to fit inside 10.25 or 13.75mm bearings, respectively. Because of the nonstandard diameter, the cost advantage is negligible, but ease of assembly and better performance are its strong points.

- TileRunner
    - The TileRunner is an unassembled chassis kit sold by Andymark. Generally, while it is usable, it is not recommended due to its price point. It is cheaper to assemble a comparable drivetrain from kit parts.

- Torsional rigidity
    - Torsional rigidity refers to how difficult it is to twist an object due to an applied torque. This mainly refers to extrusion, as it is easier to twist extrusion than channel or an angle piece, for example. Torsional rigidity has consequences particularly in building drivetrains, as the drivetrain is the last mechanism on your robot that should flex or bend when weight or force is applied to it.

- UltraHex
    - UltraHex is ½” aluminum hex shafting sold by REV Robotics. There is an inner 5 mm hex bore in the middle, which allows compatibility with REV’s 5 mm hex shaft motion system. The 5mm hex bore also allows for a ¼-20 or M6 screw to be tapped into it. ½” hex is also compatible with many FRC vendors.

- USB Retention Mount
    - The USB Retention Mount, sold by REV, is a plastic part affixed to the expansion hub that relieves stress on the USB Mini port. This is especially important because if the USB cable is loose or disconnected, the robot phone cannot communicate with the Expansion Hub, causing a disconnect. **It is highly recommended for teams to purchase the USB retention mount.**

- VersaPlanetary gearbox

    - The VersaPlanetary gearbox is a customizable gearbox attachable to RS-555 series motors. It is a high-end gearbox option for teams looking to construct mechanisms that require a high gear ratio, such as arms.

- VEX Motor Controller 29
    - The VEX Motor Controller 29 is used specifically to convert the PWM signal used in three-wire servo cables to the two-wire cable that connects to the 393 motor. It is highly recommended to protect the motor controller from **any** sort of impact, as they easily break and/or wires become detached. Also, it is important to have the MC29 as close to the Servo Power Module as possible, so that the signal doesn't become overly noisy.

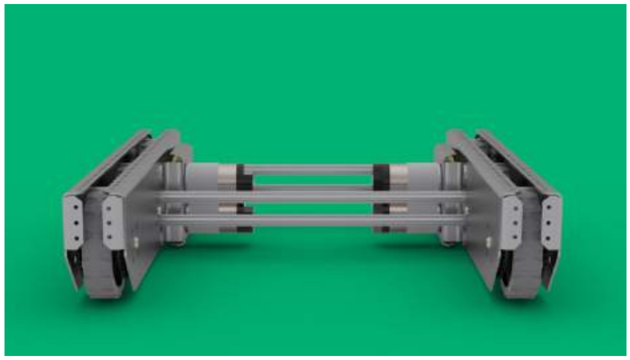
- VEX 393 EDR
    - The VEX 393 motor is a special type of motor that utilizes ⅛" square shaft. Therefore teams will have to fashion a custom motor mount and shaft adapter for the VEX 393. Under FTC rules, it is classified as a servo. However, in order to use the 393, teams must purchase an adapter from the 2-wire motor cable to the 3-wire servo cable, called the Motor Controller 29. These items are not usually in stock during the season, due to the demand from both VEX and FTC teams. Additionally, teams must purchase a Servo Power Module from REV Robotics to boost the output that the expansion hub provides. It is advised that only experienced teams use the 393 motor for this reason.


VEX 393 motor, 3736 Serious Business’ intake

- V-Groove bearing
    - V-groove bearing is a special type of bearing which has slanted “grooves” that allow for extrusion or rails to slide in between the bearing. V-groove bearings are often used in FRC for constructing linear elevators. In general, v-groove bearings are somewhat unnecessary in FTC unless a hang is involved, as the linear slide options presented in the Linear Slide guide are more than adequate for most use cases.

REV v-groove bearing with 1” extrusion

- West Coast Drivetrain (WCD)
    - This type of 6 wheel drive drivetrain was first pioneered in FRC by west coast teams like FRC 254 The Cheesy Poofs, giving it the nickname. This drivetrain was later adapted to FTC use due to its simplicity, durability, and great characteristics. The strict technical definition of a West Coast Drive is a drop center 6 wheel cantilevered drive where the center wheel is powered by a dual or triple motor input and the other wheels are chained/belted to the center wheel. Of course in FTC, this definition becomes much more lax, with most drop center 6 wheel drives being parallel plate.



12897 Nate's prototype

- XT30
  - The XT30 connector is used in the REV ecosystem through the Expansion or Control Hub. The XT30 through the REV Slim Battery provides power to the Expansion Hub, and teams will need an XT30 cable to transfer power from the main hub to a secondary hub.

- Yellow Jacket
  - Yellow Jacket is the RS-555 series motor and planetary gearbox sold by goBILDA. It has a 6mm D-shaft and is available in many different gear ratios up to 188:1.

Postscript

To all individuals and all teams who have taken the time to read this guide, on behalf of all of its contributors, we hope this resource has been helpful to you and wish you the best of luck for next season! If you have any questions, don't be afraid to direct them to gamemanual0@gmail.com.