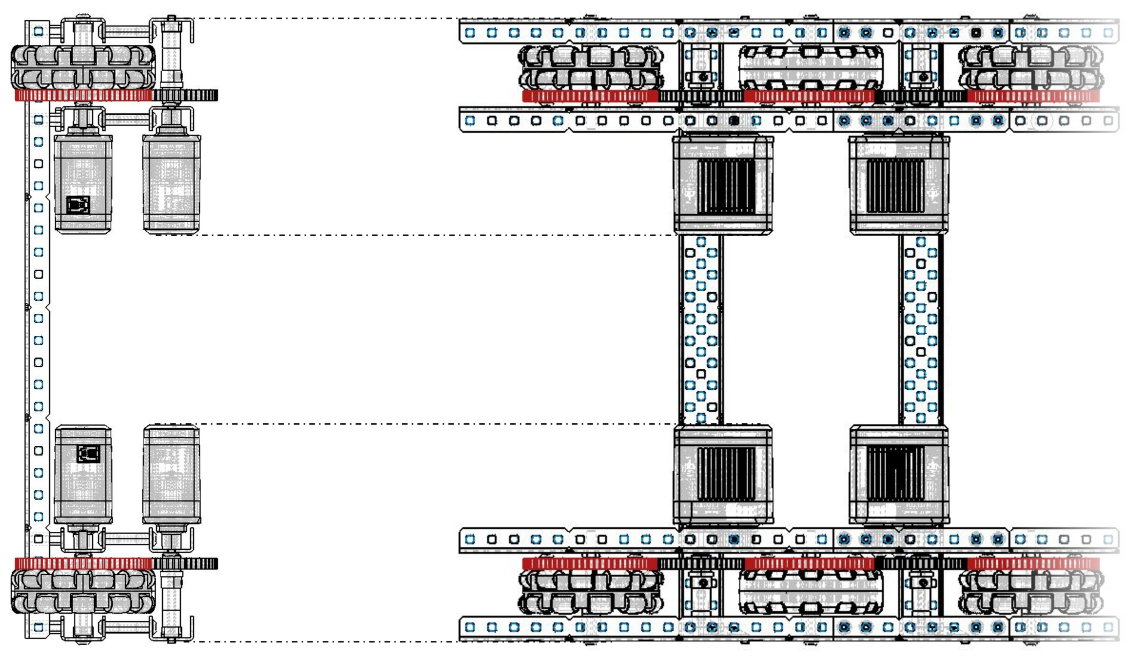
How To Build A 6627X Chassis

Build Summary:

The chassis build involves assembling the core structure of the robot, starting with precise measurements and ensuring proper alignment of C-channels, screws, and standoffs for stability. The traction and omni wheels are installed for optimal movement, while the 48T gear assemblies are integrated to support the drivetrain. Tracking wheels are added for accurate navigation, and sensors are incorporated for enhanced control. The chassis is then programmed to ensure smooth operation, and all components are carefully tested for reliability and performance. This step lays the foundation for the robot's functionality and maneuverability.



Assembly Steps:

1. **Assemble the Structure**
   The structure of a robot refers to the physical framework or "skeleton" that supports all its components, including motors, sensors, and any mechanical subsystems. The structure is essential for providing stability, durability, and proper alignment of parts. This structural framework often includes materials like metal or plastic (e.g., C-channels and spacers) and various fasteners (screws, nuts, and standoffs) that are used to assemble it. A well-designed structure ensures that the robot maintains its shape and functionality under stress, making it foundational to both the robot's performance and its longevity.
2. **Lathe 48 and 72 Tooth Gears** (Optional)
   This step involves using a lathe to modify the gears, reducing their weight and size. This step is optional and intended for teams seeking to enhance efficiency through precision machining. However, small modifications are needed if this step is skipped.
3. **Assemble the Wheels and Gears**
   A wheel or gear assembly refers to the complete setup that includes the wheel or gear itself, along with all additional components needed to attach it to the robot and ensure proper operation. This includes the wheel and/or gear, axle or screw, bearings, amd hardware.
   a. Omni Wheel Assembly (Repeat 4 Times)
   Four of these assemblies are required.
   b. Traction Wheel Assembly (Repeat 2 Times)
   Two of these assemblies are required.
   c. 48T Gear Assembly (Repeat 6 Times)
   Six of these assemblies are required.
   d. Attach Previous Wheel and Gear Assemblies to Structure
4. **Implement Electronics**
   To implement electronics in robotics means to integrate and set up the electronic components necessary for the robot's functionality. This involves installing, connecting, and configuring various components.
5. **Assemble and Incorporate Tracking Wheel**
   This step involves constructing the tracking wheel and integrating it into the robot's design. This wheel helps the robot maintain accurate positioning and alignment, improving navigation and overall performance.

Programming:

6. **Program the chassis**
   This step involves developing and implementing code to control this chassis. This includes configuring motors, setting up the rotational sensor of the tracking wheel, and fine-tuning drivetrain functions to ensure precise movement and responsiveness during operation.

What Makes our Design Unique?

Benefits of Screw Joints

In our design, we chose to use screw joints for moving components that aren't directly powered by a motor, rather than the more common choice of shafts. This decision was made to enhance the reliability, efficiency, and durability of our design.

Precision and Stability
Screw joints provide a secure and rigid connection that prevents slippage, ensuring consistent, controlled movement. This added stability allows for reliable operation, even in high-stress scenarios.

Low-Friction Pivots
By incorporating washers or bushings, screw joints significantly reduce friction at pivot points. This not only creates smoother motion but also minimizes wear over time, extending the lifespan of the components.

Durability
Unlike traditional shafts, which are prone to bending under stress, screw joints maintain their shape and functionality over time. This makes them a more robust option, especially in high-load or repetitive-motion applications.

By using screw joints, we achieve a more durable, efficient, and reliable design, making this approach a key feature of our robot.

Tracking Wheels

A key feature that sets our design apart is the integration of a tracking wheel equipped with odometry, positioned beneath the chassis. This design choice enhances the accuracy and precision of our movements during both the autonomous and driver-controlled periods.

Improved Positional Accuracy
The tracking wheel continuously measures back-and-forth movements, providing real-time data on the robot's position. This ensures that our robot can execute complex tasks and return to precise locations with minimal error.

Enhanced Autonomous Performance
By utilizing odometry, the tracking wheel allows us to create more reliable and consistent autonomous routines. This precision is crucial for scoring points efficiently during the autonomous period.

Driver-Controlled Benefits
Even during the driver period, the tracking wheel contributes to smoother and more controlled movement.

By incorporating a tracking wheel beneath our chassis, we enhance the overall functionality of our robot, making it more versatile, efficient, and competitive in both autonomous and driver-controlled phases.

STEP 1 Overview

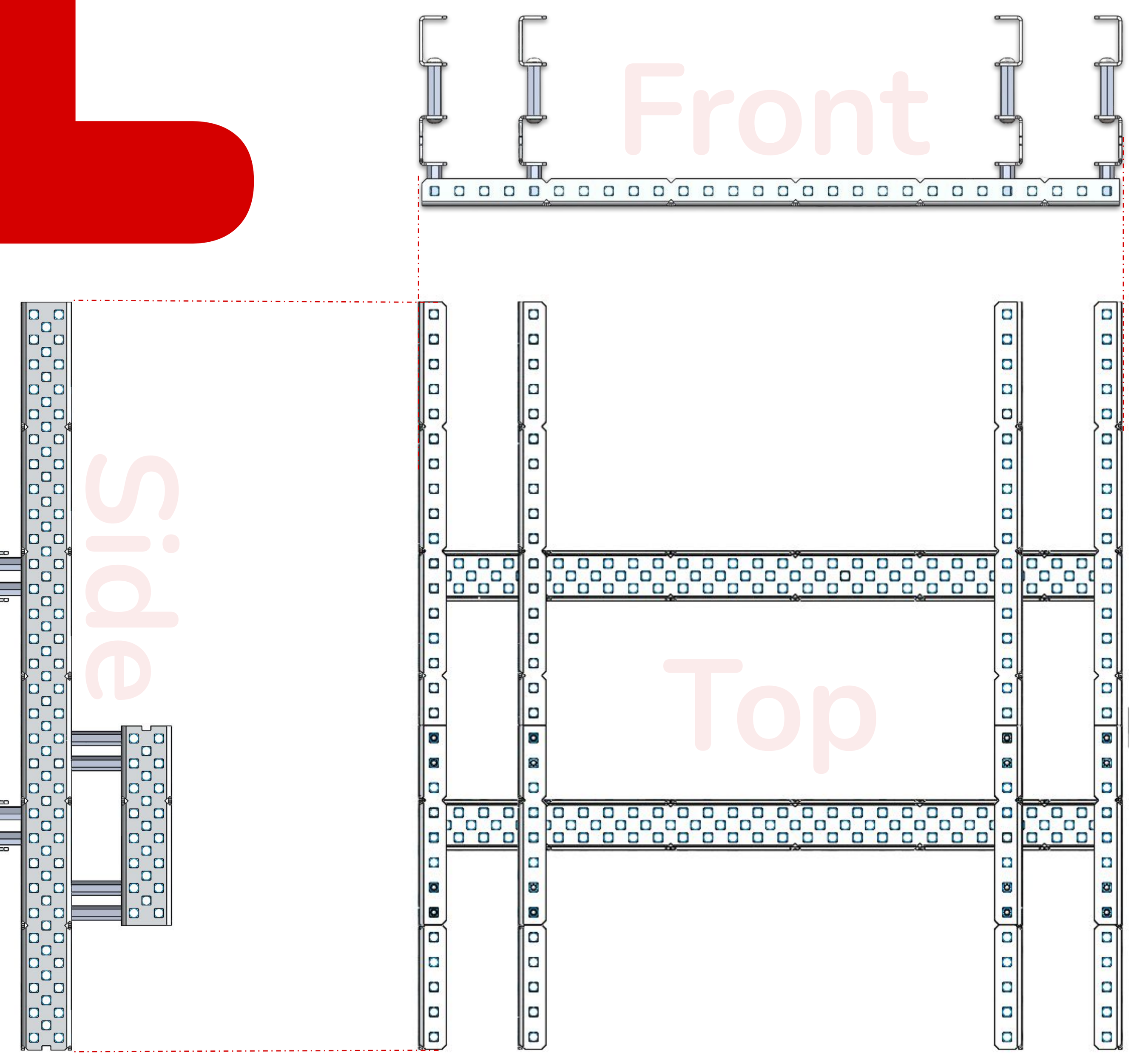
Structure

C-channels, screws, and standoffs are used throughout the chassis because they provide a sturdy and modular foundation, which is essential for the robot's stability and strength. C-channels allow for a versatile frame design, while screws and standoffs securely fasten and brace the structure, ensuring it can withstand the forces and impacts encountered during competition.

Tool Suggestion: 3/32" Screwdriver

COMPLETE BOM FOR STEP 1

QTY	PART
2	1x2x1x28 Aluminum C-Channel
4	1x2x1x8 Aluminum C-Channel
4	1x2x1x30 Aluminum C-Channel
32	0.750" Standoff
8	0.25" Screw
48	0.375" Screw



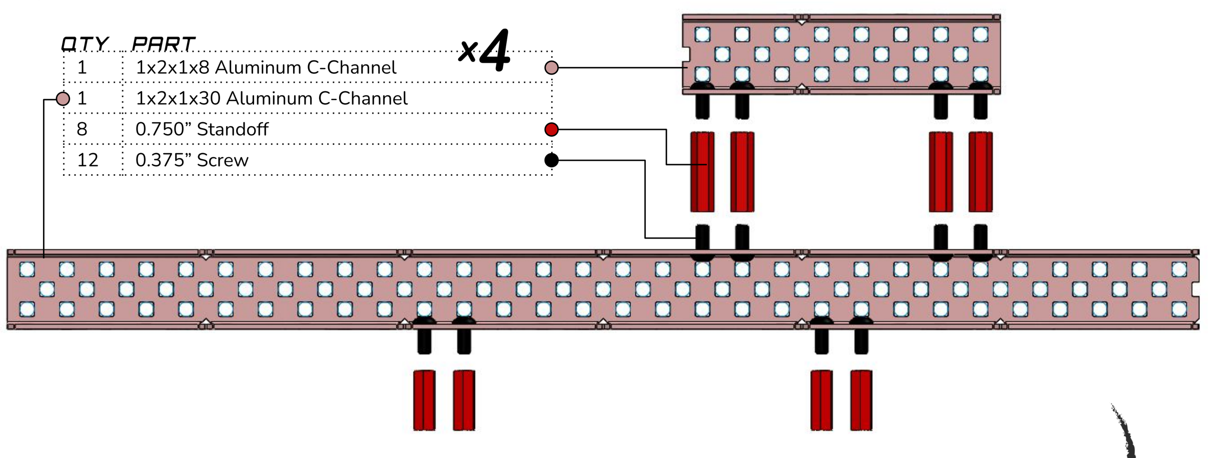
This overview may be referenced during parts A and B for further clarity.

PART A ×4

(Repeat this step 4 Times)

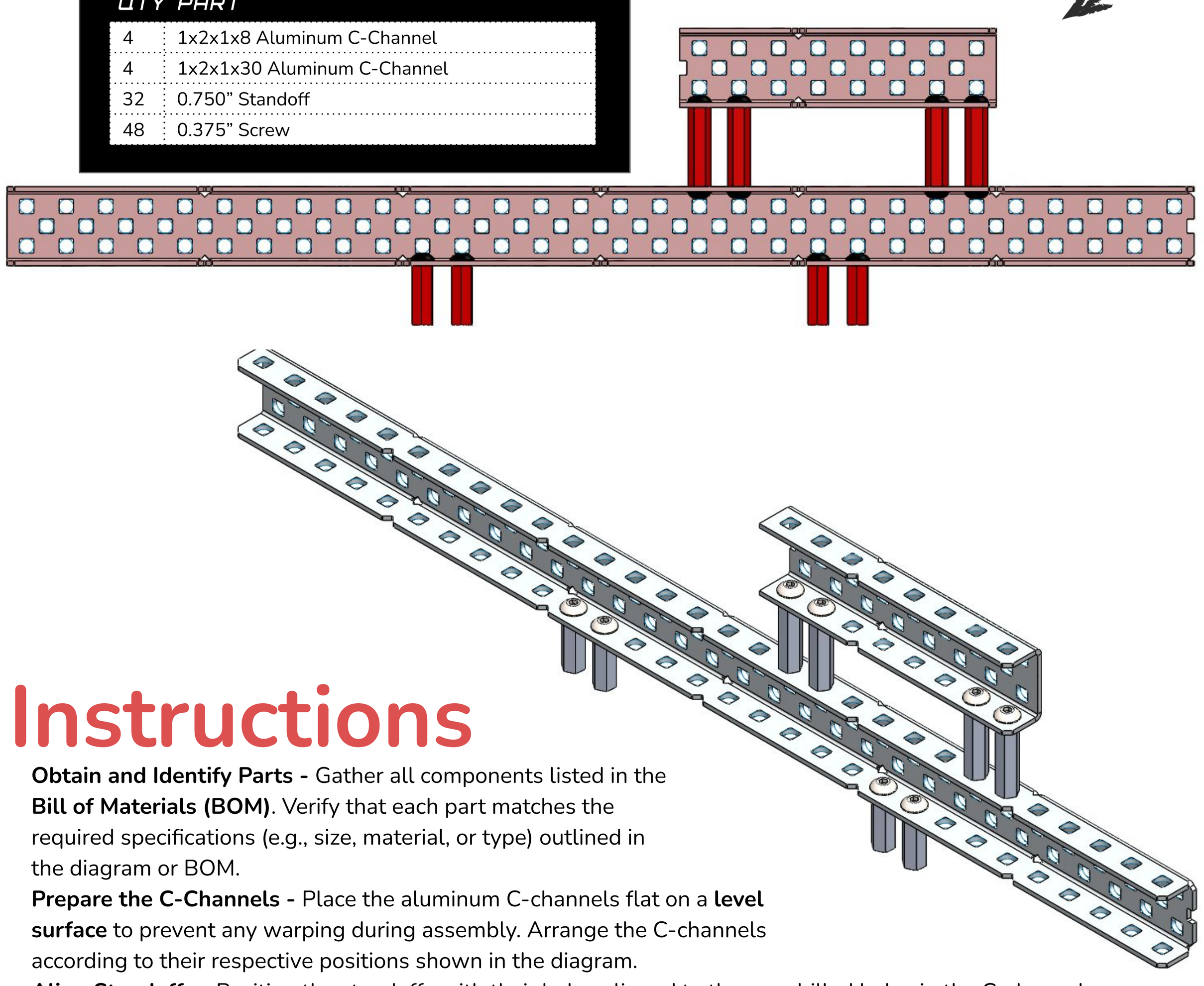
QTY	PART
1	1x2x1x8 Aluminum C-Channel
1	1x2x1x30 Aluminum C-Channel
8	0.750" Standoff
12	0.375" Screw

×4



COMPLETE BOM FOR PART A

QTY	PART
4	1x2x1x8 Aluminum C-Channel
4	1x2x1x30 Aluminum C-Channel
32	0.750" Standoff
48	0.375" Screw



Instructions

1. **Obtain and Identify Parts -** Gather all components listed in the **Bill of Materials (BOM)**. Verify that each part matches the required specifications (e.g., size, material, or type) outlined in the diagram or BOM.
2. **Prepare the C-Channels -** Place the aluminum C-channels flat on a **level surface** to prevent any warping during assembly. Arrange the C-channels according to their respective positions shown in the diagram.
3. **Align Standoffs -** Position the standoffs with their holes aligned to the pre-drilled holes in the C-channels. Ensure the standoffs match the indicated lengths in the diagram.
4. **Assemble the C-Channels -** Use screws to securely fasten the C-channels together, threading them through the aligned holes and standoffs. Tighten the screws evenly to avoid misalignment or unnecessary tension.

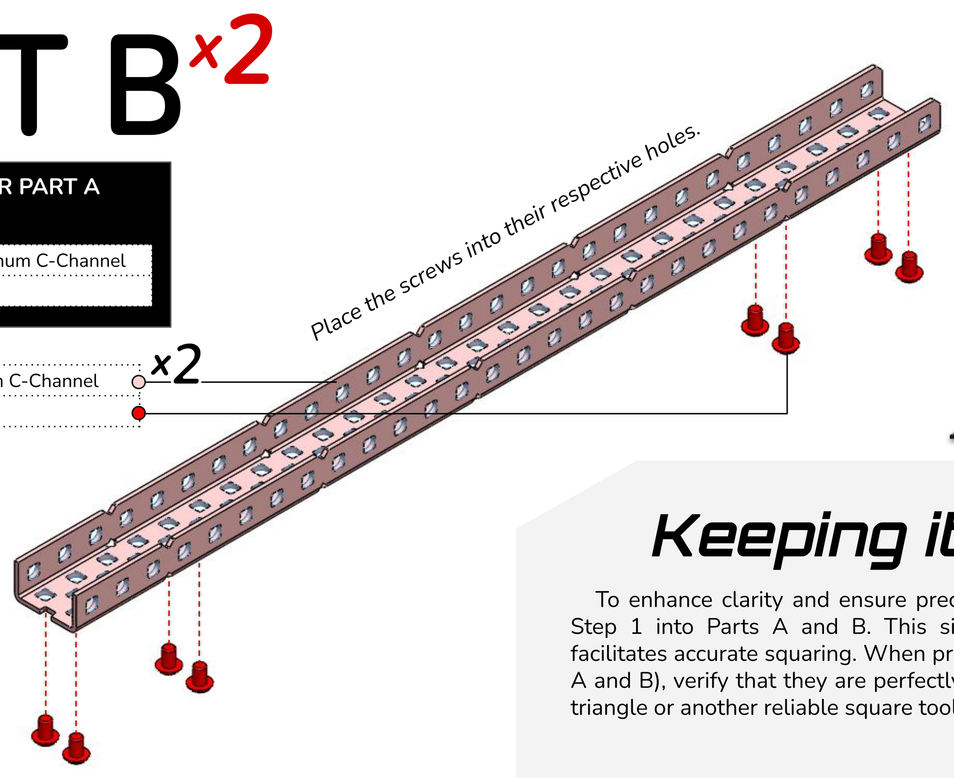
PART B ×2

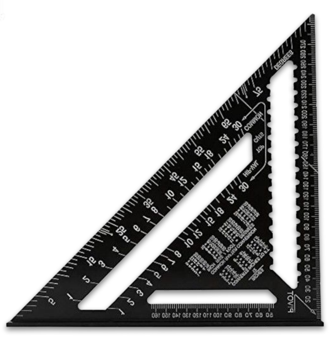
COMPLETE BOM FOR PART A

QTY	PART
2	1x2x1x28 Aluminum C-Channel
8	0.25" Screw

QTY	PART
2	1x2x1x28 Aluminum C-Channel
8	0.25" Screw

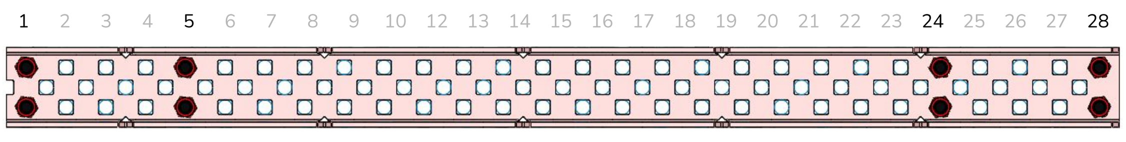
×2





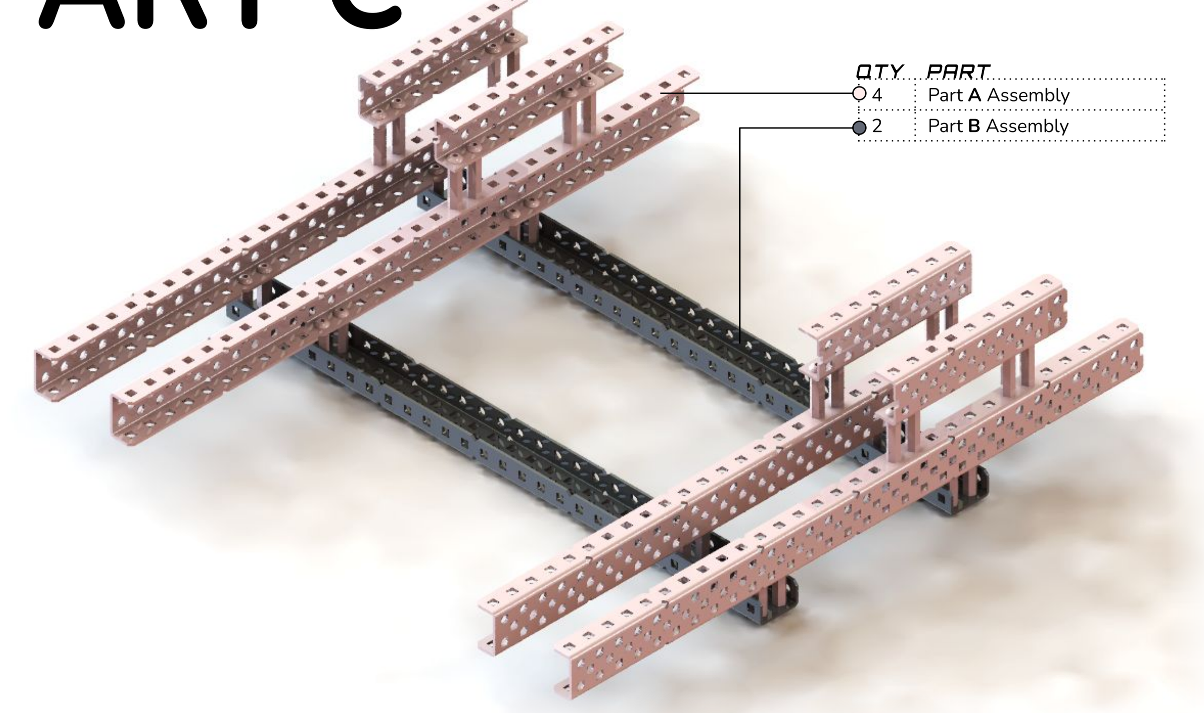
Keeping it Square

To enhance clarity and ensure precise assembly, we divided Step 1 into Parts A and B. This simplifies the process and facilitates accurate squaring. When proceeding to Part C (joining A and B), verify that they are perfectly perpendicular by using a triangle or another reliable square tool.

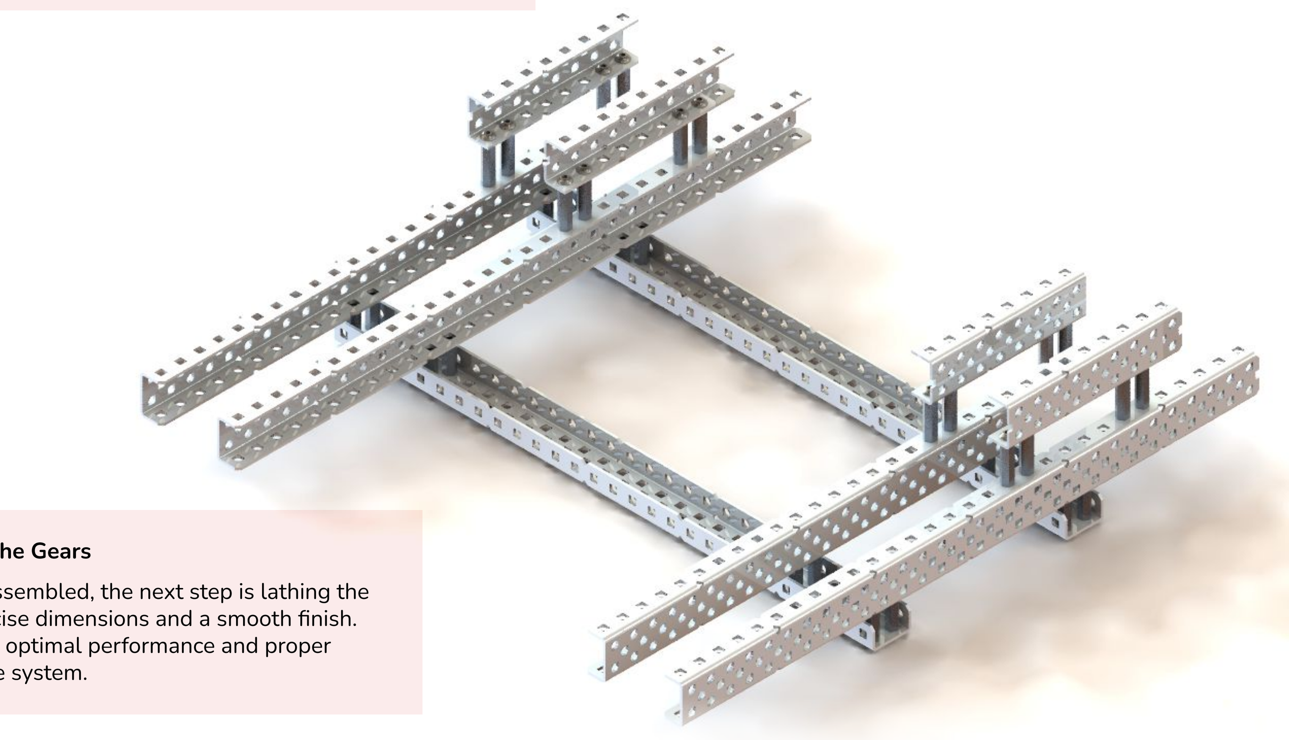


PART C

QTY	PART
4	Part **A** Assembly
2	Part **B** Assembly



Isometric View of the Subsystem Upon Completion of Step One



Next Step: Lathing the Gears

With the structure assembled, the next step is lathing the gears to achieve precise dimensions and a smooth finish. This process ensures optimal performance and proper integration within the system.

STEP 2

OBJECTIVE

We lathed the gears of our robot's chassis to reduce the overall width by 1 inch, improve maneuverability, and ensure the robot met competition size constraints. By slimming down the gear width, we were able to create more space within the chassis, allowing for a more compact design. This reduction in overall width helps the robot navigate tight spaces more easily, making it more agile during matches. Additionally, it helps us optimize the robot's weight distribution, which is crucial for maintaining balance and stability while performing various tasks.

The objective of this step is to reduce the thickness of six 72-tooth plastic gears from 0.5" to 0.25" using a lathe. Each gear is carefully secured in the chuck to ensure proper alignment along the axis of rotation and then a sharp cutting tool is used to shave off 0.25".

ALTERNATIVE

If one does not have access to a lathe, unmodified 72 Tooth and 48 Tooth High Strength gears may be used, however longer metal and screws must be used accordingly:

> (A 1x2x1x30 Aluminum C-Channel must be used instead of a 1x2x1x28 Aluminum C-Channel in Step 1, Part B)

> (A 2.75" Screw must be used instead of the 2.50" Screw in Step 3, Part A and B)

> (A 1.50" Screw must be used instead of the 1.25" Screw in Step 3, Part A and B)

> (A 3.25" Shaft must be used instead of the 3.0" Shaft in Step 3, Part C)

Steps to Reduce the Thickness of a Plastic Gear on an Automated Lathe

1. **Prepare the Lathe and Work Area**
Ensure the lathe is powered on and in manual mode. Clean the work area to remove any debris. Confirm that the cutting tool (suitable for plastic) is sharp and installed securely in the tool post.

2. **Secure the Gear to the Lathe via High Strength Shaft**
Insert a 4" high-strength shaft through the center of the gear to provide support and stability. Clamp the shaft securely into the lathe chuck, ensuring the gear is properly centered and aligned along the lathe axis. Double-check that the gear is seated flush against the chuck or a spacer to prevent wobbling.

3. **Set the Cutting Tool**
Mount a cutting tool designed for plastic machining (e.g., a high-speed steel or carbide tool with a sharp edge). Adjust the tool height to align with the centerline of the gear for precision cutting.

4. **Define Work Offsets**
Manually jog the cutting tool to the surface of the gear to set the Z-axis work offset. Zero the X-axis to the outer diameter of the gear as a reference.

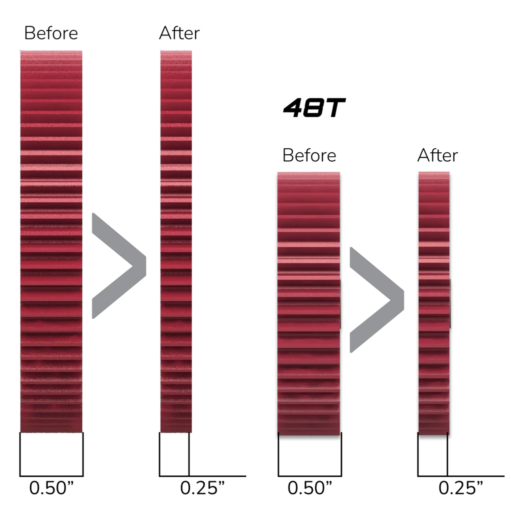
5. **Start the Lathe**
Set the spindle to an appropriate speed for machining plastic (moderate RPM to prevent melting). Ensure the feed rate is slow enough to avoid chatter but fast enough to maintain smooth cutting.

6. **Lath the Gear**
Jog the cutting tool towards the chuck by 0.25" Continue cutting the gear until the thickness is reduced from 0.5" to the desired 0.25".

7. **Inspect and Deburr**
Once the gear reaches the desired thickness, stop the lathe and remove the gear. Inspect the gear's surface for uniformity and check the thickness using calipers. If necessary, deburr the edges of the gear with a small file or abrasive tool to remove any sharp edges or plastic shavings.

8. **Clean the Machine**
Remove any plastic debris or chips from the lathe and surrounding area. Wipe down the machine to prepare it for the next operation.

Scan for a video tutorial of this step!



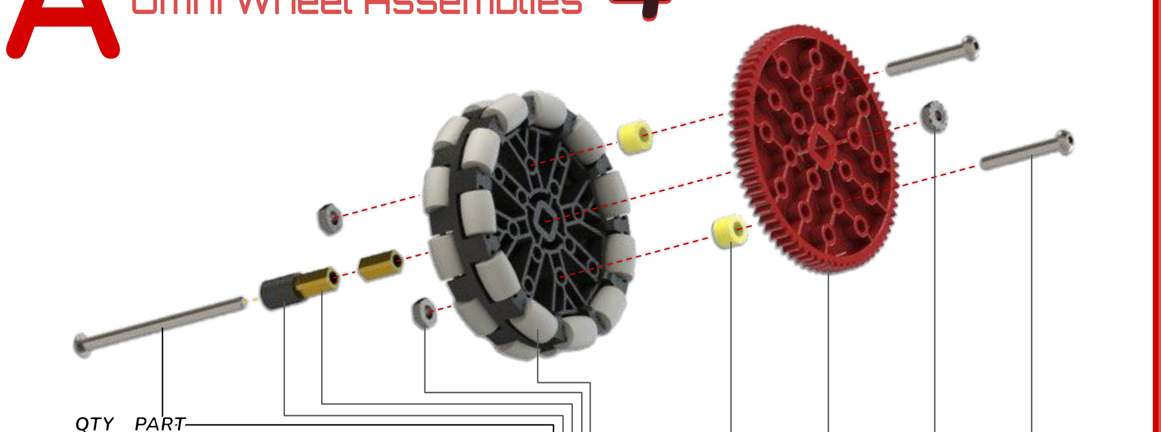
QTY	PART
1	4.0" Shaft
1	0.50" High Strength Spacer
2	High Strength Shaft Collar
2	0.50" Screw
2	Nylock Nut
The parts above do not translate into the next step, however, they are necessary to complete this step and yield the 12 lathed gears needed in the future.	
6	48T High Strength Gear
6	72T High Strength Gear

Lathe Set Up:

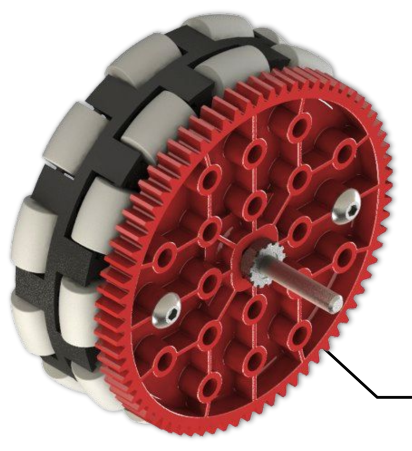
This side goes in the chuck of the lathe!

STEP 3

A Omni Wheel Assemblies x4

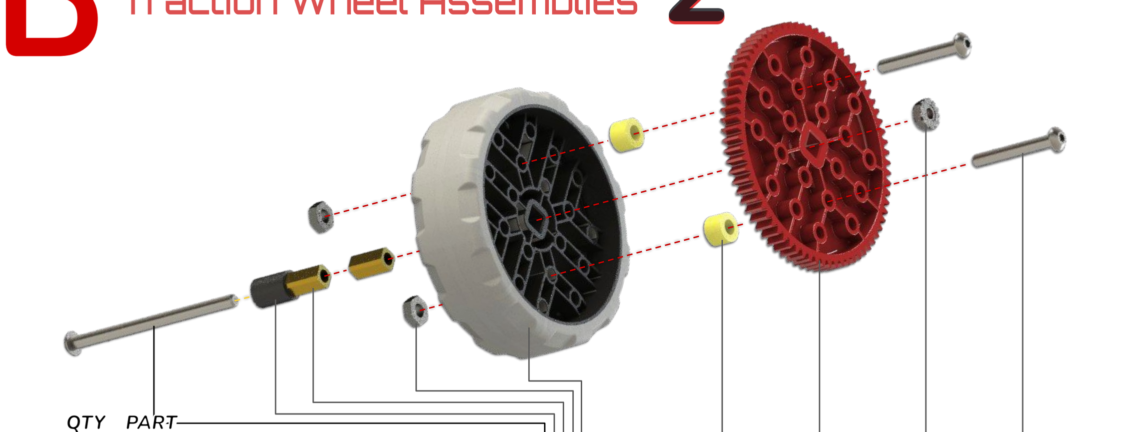


QTY	PART
1	2.50" Screw
1	0.50" 0.375 OD Nylon Spacer (Black)
2	High Strength Shaft Adapter Round
2	Low Profile Nylock
2	3.250" Omni-Directional Anti-Static Wheel
2	0.250" OD Nylon Spacer (Yellow)
1	72T High Strength Gear ***Lathed***
1	Nut Keps
2	1.250" Screw



The final assembly should resemble above visual.

B Traction Wheel Assemblies x2

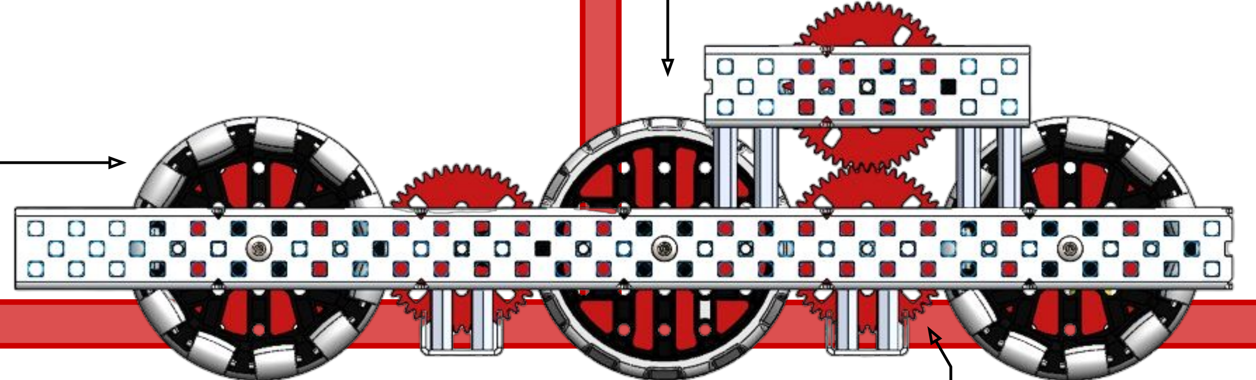


QTY	PART
1	2.50" Screw
1	0.50" 0.375 OD Nylon Spacer (Black)
2	High Strength Shaft Adapter Round
2	Low Profile Nylock
2	3.250" Omni-Directional Anti-Static Wheel
2	0.250" OD Nylon Spacer (Yellow)
1	72T High Strength Gear ***Lathed***
1	Nut Keps
2	1.250" Screw

The final assembly should resemble above visual.

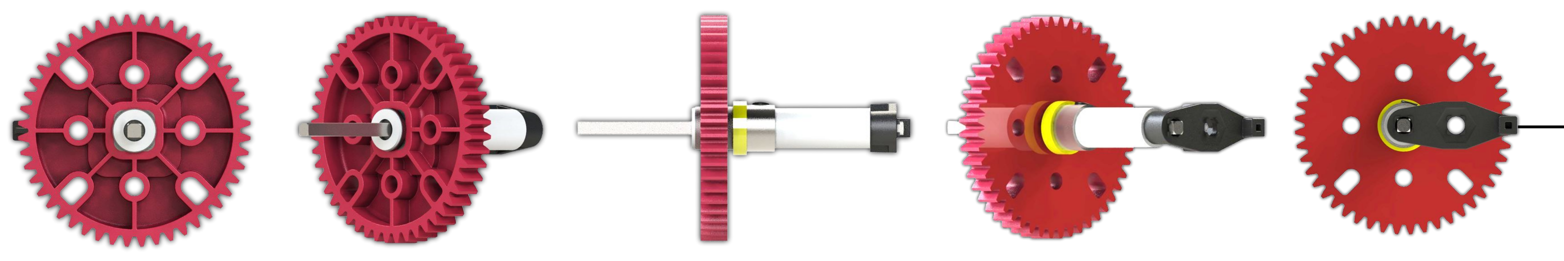
D

Position the omni wheel, traction wheel, and 48T gear assemblies in their designated locations, integrating these components into the robot's framework.

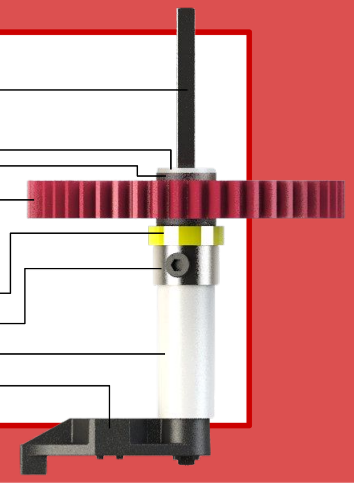


C 48T Gear Assemblies x6

Assorted Viewing Angles Below



QTY	PART
1	3.0" Shaft
1	1/32th Plastic Washer
2	High Strength Gear Insert
1	48T High Strength Gear ***Lathed***
1	0.125" OD Nylon Spacer (Yellow)
1	Shaft Collar
2	0.5" OD Nylon Spacer (White)
1	1 Post Hex-Nut Adapter

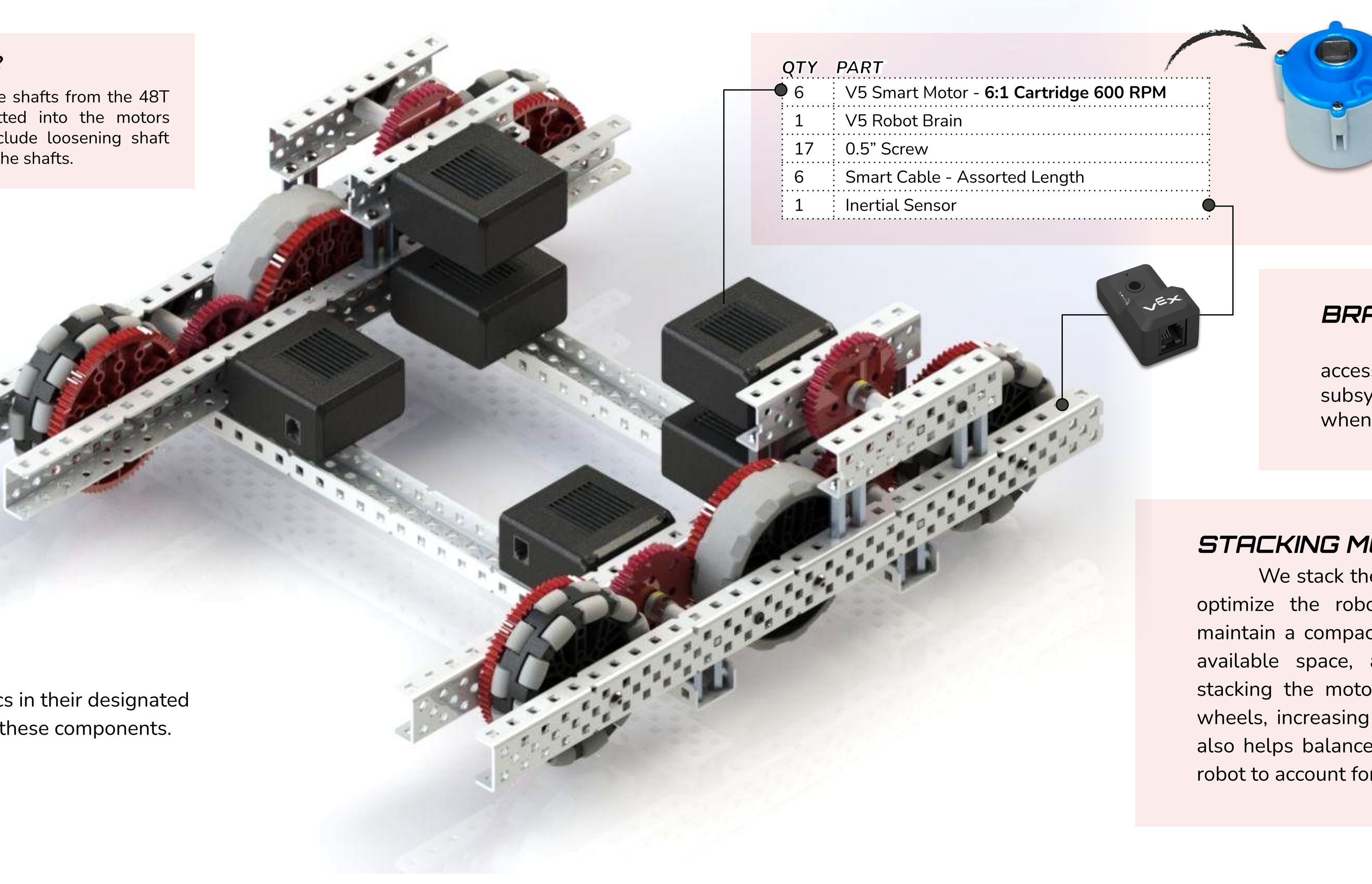


4 STEP Electronics

In the fourth step, we integrate the necessary electronics into the robot's chassis. This includes adding six V5 motors and the V5 Robot Brain, secured with 0.5" inch screws.

Shaft not in the motor?

Double check that the shafts from the 48T Gear Assemblies are fitted into the motors completely. This may include loosening shaft collars and repositioning the shafts.



Position the electronics in their designated locations, integrating these components.

QTY	PART
6	V5 Smart Motor - **6:1 Cartridge 600 RPM**
1	V5 Robot Brain
17	0.5" Screw
6	Smart Cable - Assorted Length
1	Inertial Sensor

BRAIN PLACEMENT

The Robot Brain was placed centrally for easy access and optimal wiring while there are no other subsystems. The brain placement is subject to change when additional mechanisms are added.

STACKING MOTORS

We stack the back motors to save space within the chassis and optimize the robot's weight distribution. By stacking them, we maintain a compact design that allows for more efficient use of the available space, accommodating other subsystems. Additionally, stacking the motors allows for more direct power transfer to the wheels, increasing torque and improving drive efficiency. This setup also helps balance the weight of the robot towards the back of the robot to account for the intake mechanism in the front.

LAGUNA STEP 5

ALTERNATIVE

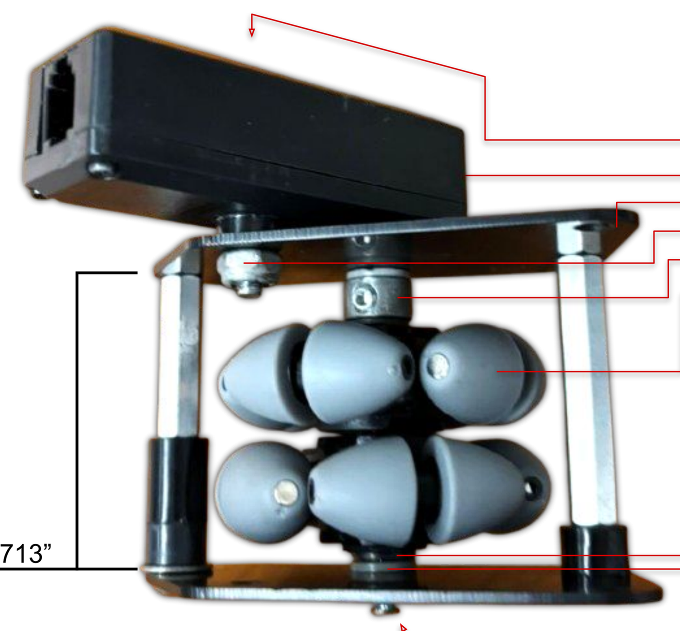
If one does not have access to a laser, the custom plastic pieces can be substituted with a 1x2x1x6 Aluminum C-Channel.

Scan this link for the Custom Plastic DXF !

Odometry pods, or tracking wheels, play a critical role in improving both driving and autonomous functions by providing precise positional data. These pods use encoders to measure the rotation and movement of independent wheels that are not influenced by slippage or friction from driving mechanisms.

By tracking the robot's movement along the y axis while preventing movement along the x axis via traction wheels, the odometry system calculates the robot's position and orientation relative to a starting point. This data enables more accurate path planning and course correction during autonomous routines, ensuring precise navigation and task execution.

Additionally, it enhances the responsiveness and reliability of driver control by compensating for any unintended drift or misalignment during operation. This precise tracking ensures that the robot can execute complex maneuvers and achieve consistent results on the field.



QTY	PART
2	1.500" Screw
1	**Rotational Sensor**
2	Custom Delrin Piece
1	Low Profile Nylock
1	Shaft Collar
2	**2.000" Omni-Directional Anti-Static Wheel**
2	High Strength Wheel Insert
2	1/16th" Teflon Washer
1	2.250" Cap Shaft
x	Standoff (Assorted Lengths)
x	Spacer (Assorted Lengths)
x	Screw (Assorted Lengths)

The amount of spacing needed is arbitrary and can be made with many different combinations of spacing, however the spacing must add up to **1.713"** in the specified region and use reasonably sized screws (<1.00").

Instructions for creating custom plastic pieces:

1. Save the Part File as a DXF
Open your CAD software (e.g., SolidWorks, AutoCAD, Fusion 360). Ensure the part is in 2D sketch mode or create a flat pattern if working with a sheet metal part. Click File > Save As. Choose DXF (*.dxf) as the file type. If prompted, adjust export settings to ensure curves are converted properly (e.g., AutoCAD 2007 DXF format is a safe choice). Click Save and note the file location.

2. Load the DXF File into LightBurn
Open LightBurn on the computer connected to the laser cutter. Click File > Import, then select your DXF file. Ensure the design appears correctly: If needed, resize or reposition it on the LightBurn workspace. Delete duplicate lines if they exist (use Edit > Select Duplicates). Use the Optimize Cut Path tool to minimize unnecessary movements.

3. Set Cutting Parameters for Acetal/Delrin
Select the cut layer in the right panel. Adjust the speed and power settings based on Delrin: 1/16" (1.5mm) Delrin. This is specific for individual lasers. Set air assist ON (this reduces charring and improves cut quality). Set Z offset if necessary (only for auto-focus lasers). Click Frame to check that the laser head moves within the correct work area.

4. Send the File to the Laser Cutter
Turn on the laser cutter and air assist (if external). In LightBurn, click Start (for direct control) or Send (to upload the file to the laser's internal memory). If using Send, go to the laser's control panel, select the file, and press Start.

5. Cutting the Part
Close the laser cutter lid. Press Start (if not already running). Monitor the cut for safety: Watch for excessive flames or smoke buildup. Pause or stop the job if needed. Once the cut is complete, wait a few seconds before opening the lid to let fumes clear. Carefully remove the part and inspect the edges. If the cut is incomplete, run a second pass at a slightly faster speed.

Final Notes
Ventilation is crucial when cutting Delrin, as it releases formaldehyde fumes. Use proper exhaust systems. Focus the laser properly before cutting for best results. Avoid very slow speeds at high power to prevent excessive melting.

PROGRAMMING

STEP 6

PROS

VS Code

EZ-Template

What is EZ Template?

EZ Template is a streamlined programming framework for VEX V5 robots using PROS (Purdue Robotics Operating System). It simplifies autonomous and driver control code implementation by providing a well-structured, pre-configured setup. This allows teams to focus on writing efficient and modular code without worrying about complex project setup.

Key Features:

- Pre-configured project structure for easier code organization
- Built-in asynchronous controls for smoother autonomous routines
- Simplified movement functions for chassis and subsystems
- Seamless integration with OkapiLib for advanced motion control
- Compatible with VEX V5 Brain and PROS CLI

EZ Template is a powerful tool that simplifies the coding process for VEX teams, making it easier to develop organized, modular, and efficient robot programs. By leveraging its pre-built structure and integration with PROS, teams can focus more on strategy and optimization rather than low-level setup issues.

How to Download EZ Template

Step 1: Install PROS
Before downloading EZ Template, ensure you have PROS installed on your computer:

1. Visit the PROS official website: https://pros.cs.purdue.edu/
2. Download and install the appropriate version for your operating system.
3. Open PROS Editor or use the Command Line Interface (CLI) to manage projects.

Step 2: Clone the EZ Template Repository

1. Open a terminal or command prompt.
2. Navigate to the directory where you want to store your project.
3. Run the following command to clone the EZ Template repository:
   git clone https://github.com/JacisNonsense/EZ-Template.git
4. Change into the project directory:
   cd EZ-Template

Step 3: Open the Project in PROS

1. Open PROS Editor.
2. Click **File** → **Open Folder** and select the **EZ-Template** directory.
3. You can now start writing code using the pre-configured functions!

Step 4: Build and Upload Code to VEX Brain

1. Connect the VEX Controller or Brain to your computer via USB.
2. Open the terminal in PROS and run:
   pros make
   pros upload

Your robot is now ready to run with EZ Template!

Odometry

Odometry (short for odometric positioning) is a method of tracking a robot's position and movement using sensors. In VEX robotics, it typically relies on encoders, tracking wheels, or inertial sensors to maintain precise localization throughout a match. EZ Template integrates odometry seamlessly, enabling more accurate autonomous movement and navigation.

How Odometry Works in EZ Template
EZ Template is designed to work with OkapiLib, a library within PROS that provides tools for motion control and odometry. The framework includes a pre-configured odometry system that can be customized based on the robot's drivetrain and sensor setup.

1. **Tracking Position with Encoders**
   - The robot uses one tracking wheel and drivetrain motor encoders to measure movement along the Y axes. *With this chassis design, movement in the X axis is restricted via traction wheels so that the programing is simpler. (XY Plane parallel to the floor)*
   - Each encoder logs how much distance the robot has traveled in both forward/backward (Y).
   - The system also calculates rotation using a third encoder or an inertial sensor (IMU).
2. **Pose Estimation**
   - The robot's position and heading (pose) are continuously updated using odometry equations.
   - By integrating small position changes over time, the system maintains an accurate (X, Y, θ) position on the field.
3. **Path Following and Autonomous Control**
   - EZ Template works with motion algorithms like PID, Pure Pursuit, and Ramsete to execute smooth and precise autonomous paths.
   - The robot adjusts its movements dynamically, correcting errors from wheel slippage or external forces.

Benefits of Using Odometry in EZ Template

High Accuracy*: Tracks position with minimal drift compared to time-based movements.*

Smooth Path Following*: Enables arc-based and curved movement instead of jerky turns*

Error Correction*: Adjusts for slippage and misalignment, improving autonomous reliability*

Simplified Integration*: EZ Template handles complex math, allowing teams to focus on strategy*

Setting Up Odometry in EZ Template

Modify the chassis controller settings in the **initialize()** function inside your PROS project.

Step 1: Configure Chassis and Odometry - In **src/main.cpp**, define the drivetrain and odometry settings:

```
#include "EZ-Template/api.hpp"

ez::Drive chassis;

// Initialize the robot
void initialize() {
    chassis.initialize();

    // Set drive motors (left, right)
    chassis.tank(10, 20);

    // Configure odometry tracking wheels (optional)
    chassis.set_odometry({{1, 2, 3}, 2.75, 4.0});
}
```

- {1, 2, 3} → The ports of left, right, and middle tracking wheels
- 3.25 → The wheel diameter (in inches)

Step 2: Using Odometry for Autonomous Movement - With odometry enabled, you can create more precise movements in the **autonomous()** function:

```
void autonomous() {
    // Drive forward 24 inches
    chassis.move_to_point(24, 0, 1000);

    // Turn 90 degrees
    chassis.turn_to_angle(90);

    // Drive to a specific field position (x = 30, y = 40)
    chassis.move_to_point(30, 40, 1500);
}
```

- move_to_point(x, y, timeout) → Moves the robot to an exact (x, y) position on the field
- turn_to_angle(degrees) → Rotates the robot to a specific heading

Hodge, Sydney	Foothill High School,
Karamally, Zain	19251 Dodge Ave,
Queeney, Samantha	Santa Ana,
Ruiz, Arianna	CA 92705,
Rustad, Andie	USA

VEX ROBOTICS

SOLIDWORKS

6627X

Knight Time Bots X

Xenoceratops