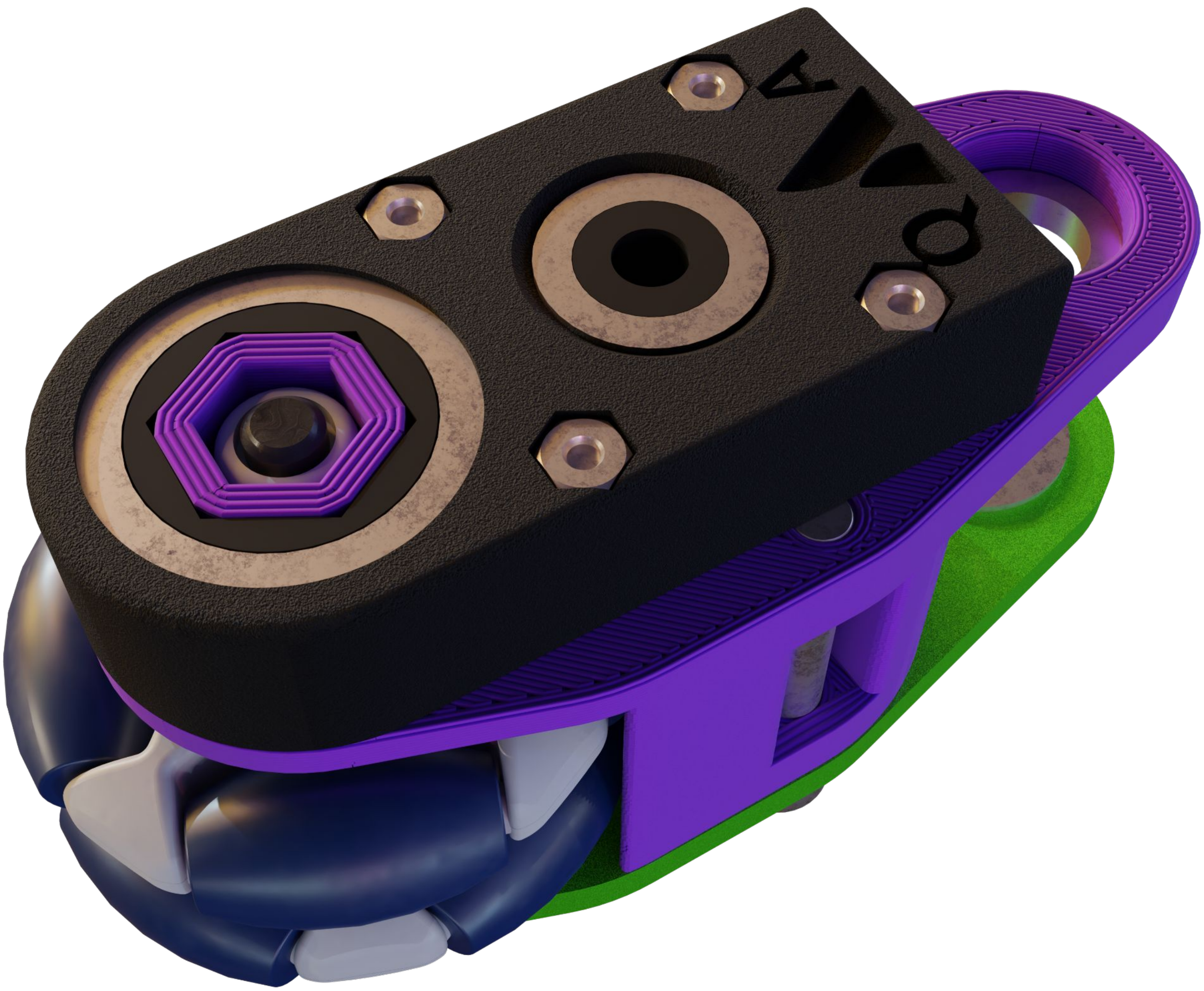
LoonyOdo Quickstart

AND TIPS FOR DESIGNING FTC ODOMETRY



LoonyOdo V0.1 BETA

Designed by Alex
Written by Alex & Jameson

TABLE OF CONTENTS

Check out our social media!

https://www.instagram.com/the_loony_squad/
https://twitter.com/the_loony_squad
https://www.youtube.com/channel/UCBaOt_nNzTs9vGdwq03uBAw

WHAT ARE ODOMETRY PODS?

Odometry pods, or dead wheels, are a method of determining a robot’s position on the field relative to where it started. Each pod has an encoder with an omni wheel that is rolled along the floor as the robot moves. As the omni wheel turns, the encoder measures the number of ticks it has turned, which can then be used to derive a distance moved. By using two or more odometry pods, one can derive the robot’s position in two axes.

Implementing odometry has two aspects: hardware and software. This guide is primarily focused on hardware and the design of odometry pods themselves. In order to utilize the pods, you will need to implement the proper programming to do so. [Learnroadrunner.com](https://learnroadrunner.com) is perhaps the most complete guide to programming with odometry. Road Runner will allow you to use odometers on your robot without having to do all the math it runs. If you are looking for a deeper dive into programming with odometry, see game manual 0.

RESOURCES FOR ODOMETRY

- https://learnroadrunner.com/ - a complete guide for implementing Road Runner, a motion profiling library for FTC that utilizes odometry
- https://gm0.org/en/latest/docs/software/concepts/odometry.html?# - game ganual 0’s documentation of odometry
- https://discord.gg/first-tech-challenge - the FTC Discord is a excellent community to consult for help with odometry

Team 7236

Team 11115

VARIONS OF ODOMETRY

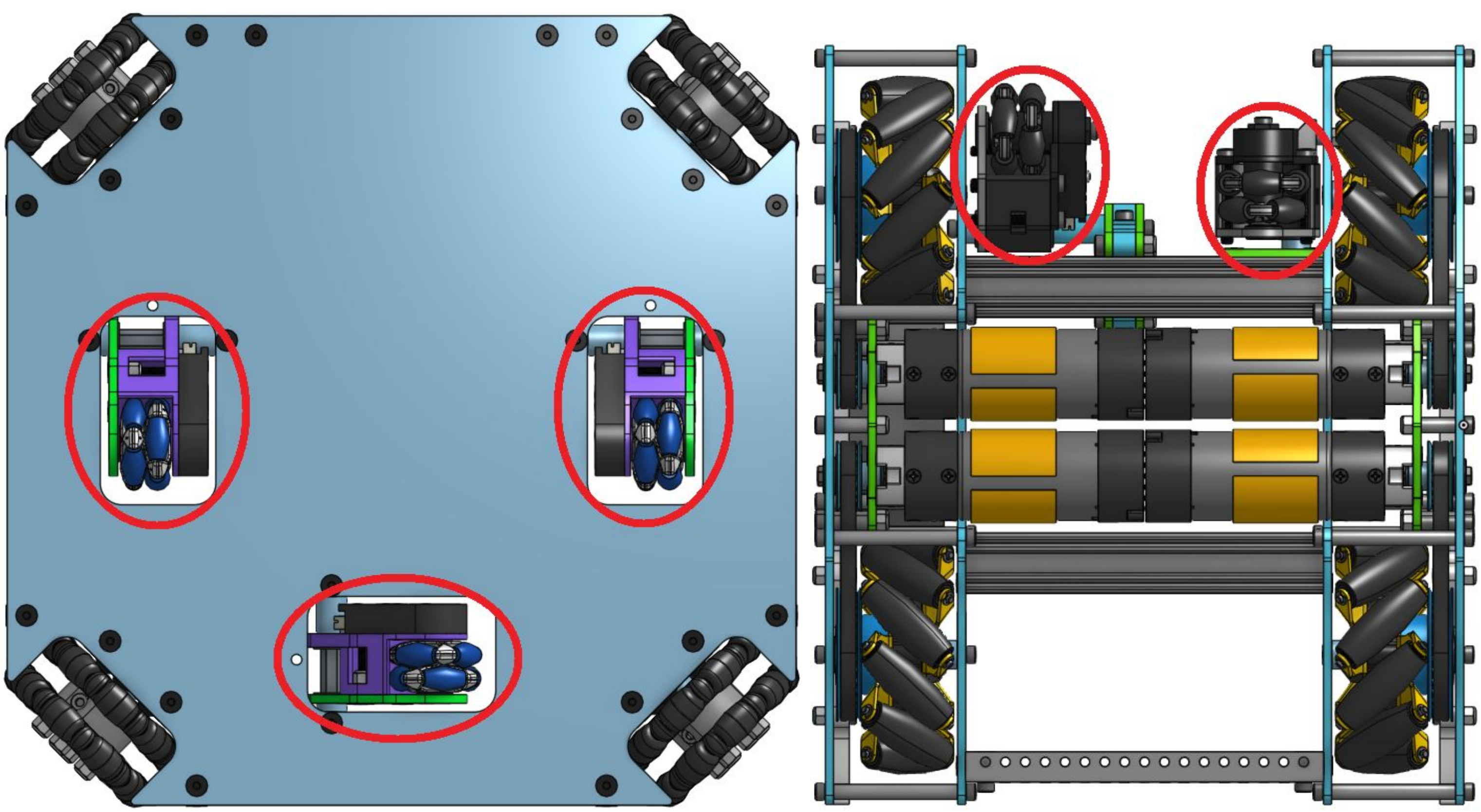
VARIATIONS OF ODOMETRY

Odometry typical comes in two forms in FTC: 3 wheel and 2 wheel. The primary difference between these two setups is how robot heading is calculated.

In a 3 wheel setup, two of the odometers are run parallel to measure the x-axis (forward/backward) movement of the robot, and the third wheel is set perpendicular to them to measure the y-axis (side/side) movement. Robot heading is calculated by measuring the difference in readings of the two x-axis pods.

In a 2 wheel setup, the robot heading cannot be derived from the odometers because the two wheels are perpendicular to each other. Instead, the IMU in the control hub (or an external IMU) must be used to calculate the robot heading. Using the IMU in the expansion hubs is <u>not recommended</u> because the very slow polling time drastically increases drift (see Road Runner documentation). However, using the control hub allows this system to be very accurate and save the additional cost and space required by a third wheel.

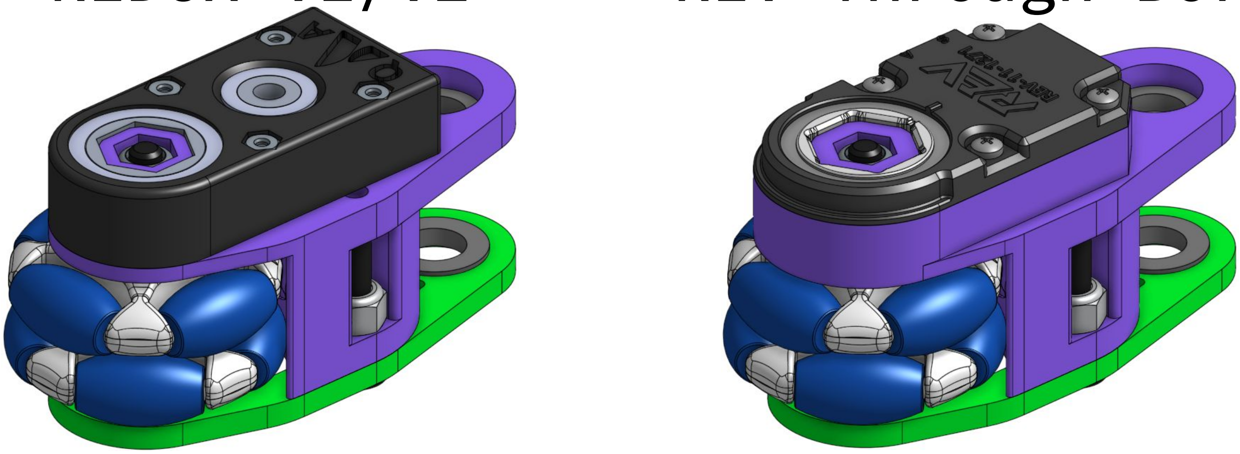
3 WHEEL VERSUS 2 WHEEL SETUP



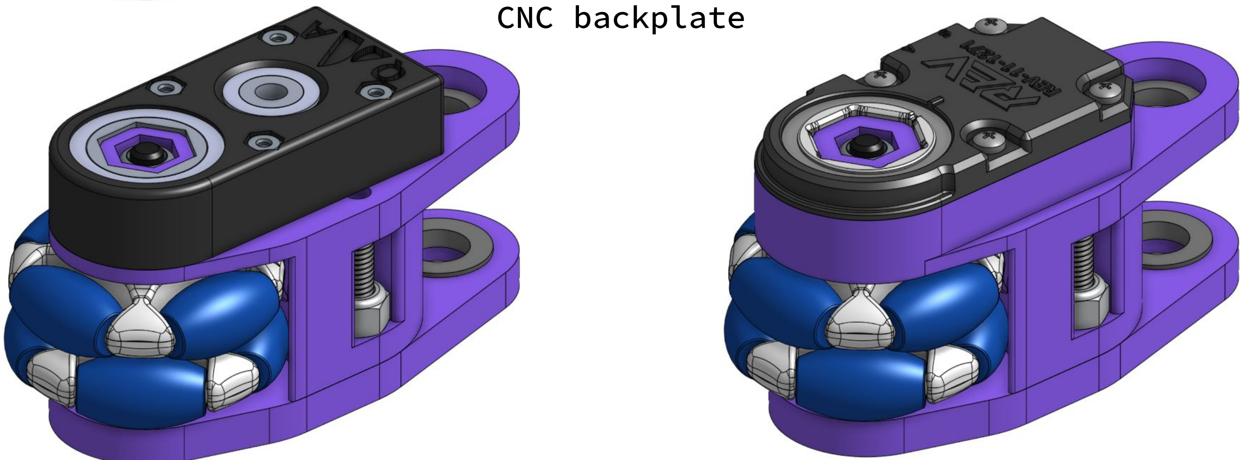
LOONYODO OVERVIEW

LoonyOdo is designed to be configurable and customizable. There are 3 official versions, each with 4 combinations of screws and backplates:

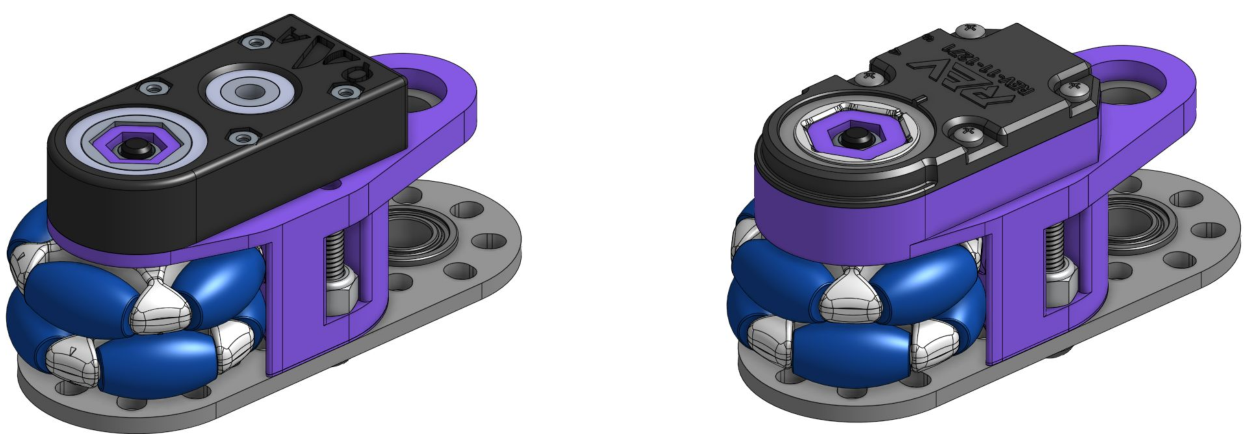
REDUX V1/V2 — REV Through Bore



CNC backplate



3Dp backplate



goBILDA backplate

Each version can be assembled with the stock M4x25 screws that come with the Codex odometry bundle or with custom M4 screws and nuts. The specific instructions for each will clarify length and head shape.

HOW THE ENCODER WORKS

LoonyOdo (and all other FTC-legal odometry pods) is based on a quadrature (quad) encoder output. Quad encoders work by outputting two out of phase channels, which convey both position and direction. Direction is derived by determining which channel is “leading” and which is “following” For more information about quad encoders, see dynapar.com.

Both the Axon Redux and REV Through Bore encoders also support an absolute output mode. This mode is not relevant for odometry because the count resets after each revolution. Keeping track of distances longer than 1 encoder rotation would require additional software to count the number of rotations. Therefore, using quad mode results in less programming and easier implementation.

SETTING ENCODER MODE

To switch between quad and absolute modes on the Axon Redux encoder, simply plug the cable into the corresponding port on the back of the encoder. The port labeled Q is the quad output, and the port labeled A is the absolute output.

The REV Through Bore Encoder also supports absolute outputs, but it is not compatible with the FTC control system

WIRING ODOMETRY

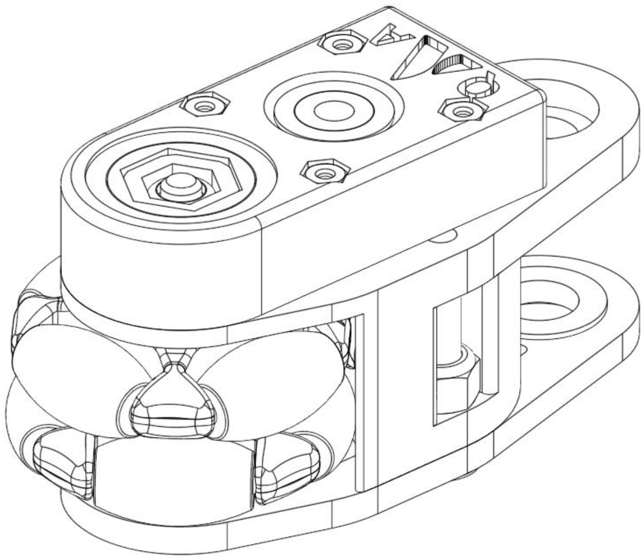
To run odometry, the motor encoder ports in the control hub must be used. While it is possible to use the expansion hub ports, best practice is to use the ports on the control hub. Unfortunately, this means that a robot with odometry and 8 motor encoders is not possible. However, using the encoders on the drive motors is not necessary if running odometry, which opens up 4 motor encoder ports.

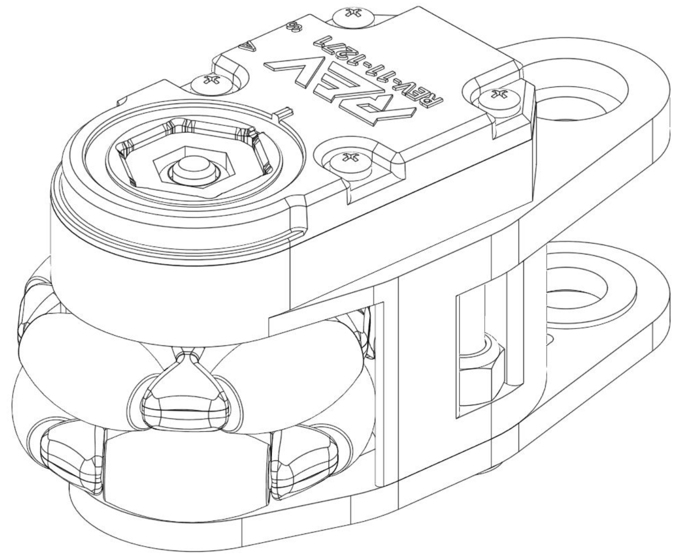
For the Redux encoder, you will need the included cable. For the REV encoder, use the included 4 wire cable. The large end plugs into the encoder and the small end into the control hub.

Several examples of encoders

CHOOSING AN ENCODER

LoonyOdo supports both the Axon Redux encoder and the REV Through Bore encoder. Each system has its own advantages and drawbacks, some of which are listed here. The assembly for each encoder can be found on pages 9 and 10





Axon Redux V1/V2

- $33.99
- Disassembles face-down, much easier to keep parts contained when taken apart
- 10.9mm height
- Quadrature or analog output
- Mounting screws come with nuts
- Resolution: 4096 CPR
- 11mm hex bore

REV Through Bore

- $48.00
- Disassembles back-down
- 11.5mm height
- Quadrature output
- Screws thread into plastic of Base
- Resolution: 8192 CPR
- 1/2in (12.7mm) hex bore

REDUX V1/V2 DIFFERENCES

Depending on when you ordered your REDUX encoder, you might get shipped a V1 or a V2 encoder. Functionally, they are identical, but the internal encoder mounting is changed slightly. You can know what REDUX version you got, as the V2 has an A etched between the two encoder ports and the V1 does not. Be sure to print the correct Base (either RX1 or RX2 for V1 and V2 respectively) to ensure that everything lines up properly

MANUFACTURING

To create the parts for LoonyOdo, and many other custom parts both in FTC and in the real world, many manufacturing options are available. This page will cover manufacturing options that are relevant to LoonyOdo, but many more can be found here: [Custom Manufacturing - Game Manual 0].

3D PRINTING

3D printing is a core manufacturing technique behind any odometry design. It allows complex and precise geometries to be created in a matter of hours for a price many can afford. Most of the parts in LoonyOdo are printed. 3D printing handles small scale well, as the initial cost to get started fabricating a part is near 0.

Fun Fact: The Redux Encoder case is also 3D printed, just with advanced SLS technology.

CNC MACHINING

CNC machining is another relatively common FTC manufacturing method. When we say CNC machining we mean CNC routing, which uses a system of stepper motors to cut through sheet material with a router bit. This method of machining can quickly produce custom plates for all sorts of robot mechanisms out of many materials.

Other methods of CNC machining include water jet cutting and laser cutting. Both of these methods have the advantage of being able to cut square inner corners, which a CNC router cannot do because of its round bit.

INJECTION MOLDING

Many COTS plastic parts are injection molded. Injection molding forces molten plastic into metal molds to create a solid part. Custom injection molded solutions are all but nonexistent for FTC because molds are expensive and require other advanced methods of manufacturing. However, nearly every commercial plastic part you can buy is injection molded.

DESIGNING FOR A MATERIAL

In order to take models from CAD to robot, a material must be chosen for a part based on three factors: precision, strength, and cost. While you can always make a part stronger than it has to be, it will be more expensive.

DESIGNING 3D PRINTED PARTS

3D printed parts are very precise, but lack in terms of strength. In order to maximize the strength of 3D printed parts, follow these general rules:

1. **Orientation:** 3D printed objects are not isotropic (same strength in all directions) due to layer adhesion. Print parts such that loads are not spread across layers.
2. **Overhangs and Supports:** 3D printers are capable of much more overhang than you might expect without using supports. Minimizing supports results in smoother and less wasteful parts.
3. **Printability:** Make sure that the part can get a large contact patch with the print bed to minimize warping

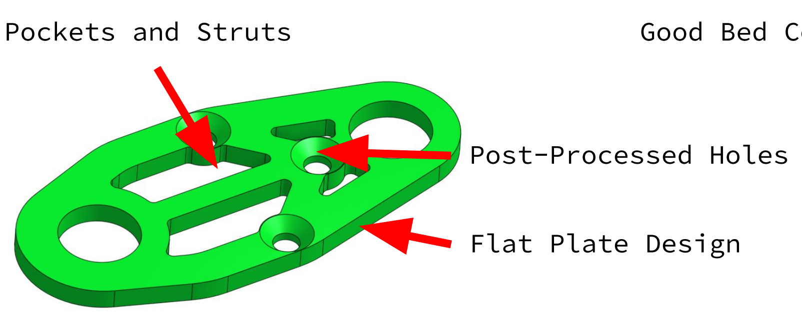
Further reading here: ALL3DP GUIDE.

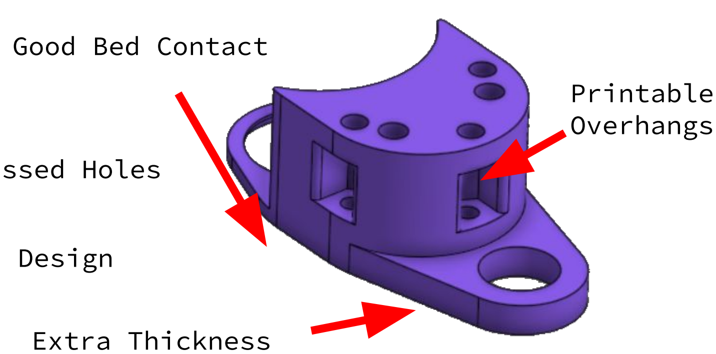
DESIGNING CNC'd PARTS

CNC-cut parts can be made from a variety of sheet materials. Aluminum and polycarbonate are by far the most common CNC-cut materials in FTC, but more can be used. Here are some general guidelines:

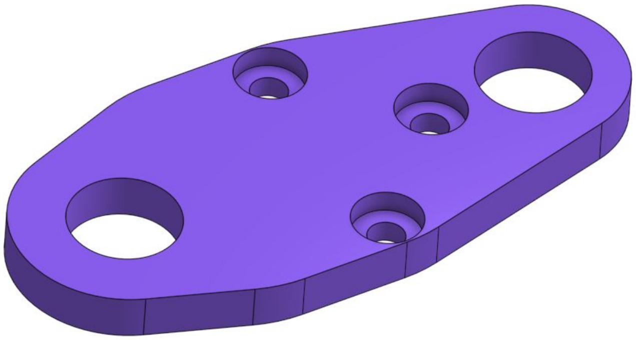
1. **Orientation:** CNC'd parts need to be laid completely flat to be cut. Plates can be bent after cutting.
2. **Tabs:** To keep parts attached to the sheet while cutting, use tabs on regular intervals. This isn't necessary for laser cutting.
3. **Pocketing and Struts:** For aluminum parts, it can be beneficial to add "pockets" in the part to reduce weight. Make sure the struts between pockets are more than 4-8mm thick depending on material.

EXAMPLES OF WELL DESIGNED PARTS



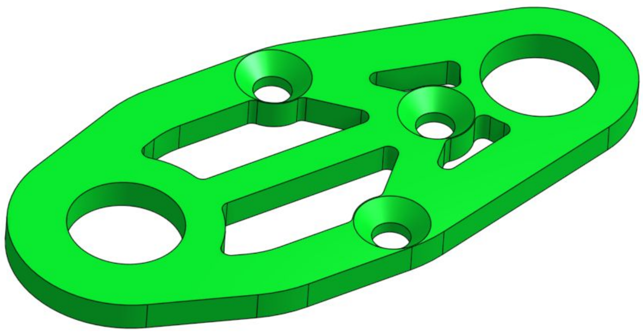


CHOOSING A BACKPLATE



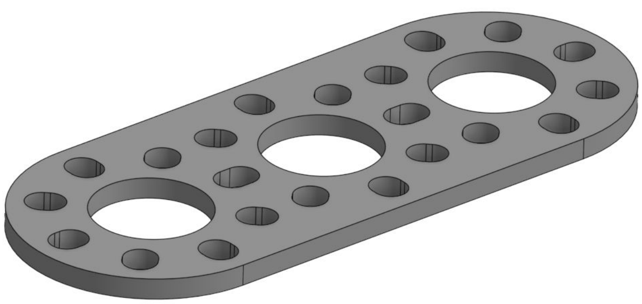
3Dp Backplate

- Weakest solution
- Least expensive
- Least clearance
- Least custom parts required



CNC Backplate

- Stronger than 3DP
- Allows for wide range of materials
- Best clearance due to countersunk screws
- Only custom parts required are screws and nuts



goBILDA Backplate

- Commercial solution
- Requires no custom manufacturing
- Requires 14mm OD bearings in addition to custom screws and nuts

SCREW SELECTION

All versions of LoonyOdo can be assembled with the stock screws from the Codex odometry bundle. Better clearance can be achieved by using custom screws. Here is a table with recommended custom screws for each type of backplate. Each set of custom screws also requires 3 M4 nuts.

BACKPLATE	**SCREW SIZE AND LENGTH**	**HEAD TYPE**	**QUANTITY**
3Dp	M4x25	Buttonhead	3
CNC	M4x22	Countersunk	3
goBILDA	M4x22	Buttonhead	3

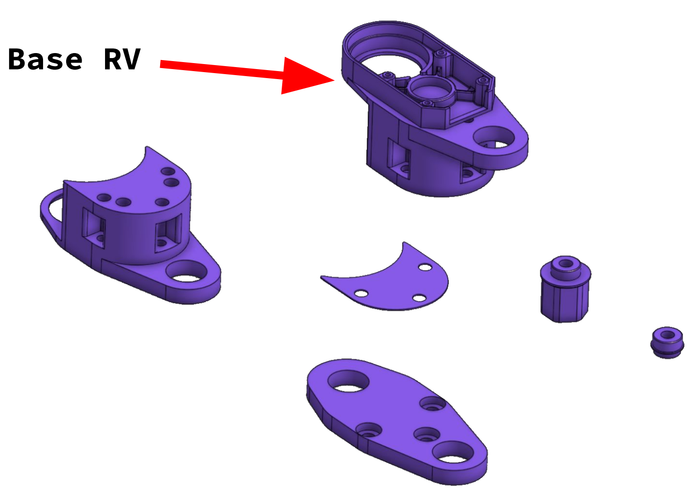
PRINTING PARAMETERS (PLA)

Here are the as-tested printing parameters for PLA. These were tested on a stock Ender 3. With MatterHackers MH Build PLA

NOTE: Your printer or filament may vary. This is intended only as a starting point.

PART NAME	INFILL (GYROID)	WALLS	TOP/BOTTOM
0.4mm nozzle 0.2mm layers			
Base	30	4	6/4
3Dp Backplate	50	4	6/4
Bearing Insert	100		
Encoder Insert	100		
goBILDA Spacer	30	4	6/4
0.6mm nozzle 0.3mm layers			
Base	25	3	3
3Dp Backplate	50	3	3
Bearing Insert	100		
Encoder Insert	100		
goBILDA Spacer	25	3	3

PART PRINTING ORIENTATION KEY



All printed parts except for Base RV do not need support material. For **Base RV**, print the part with the encoder case facing up, and set supports to only be on the build plate in the slicer. After the part finishes printing, take a drill and punch out the main holes. For teams without a drill, an allen key will do. **It is also recommended to print Base RV slower to ensure accuracy.**

NOTE: Settings for advanced manufacturing options are not provided.

BILL OF MATERIALS

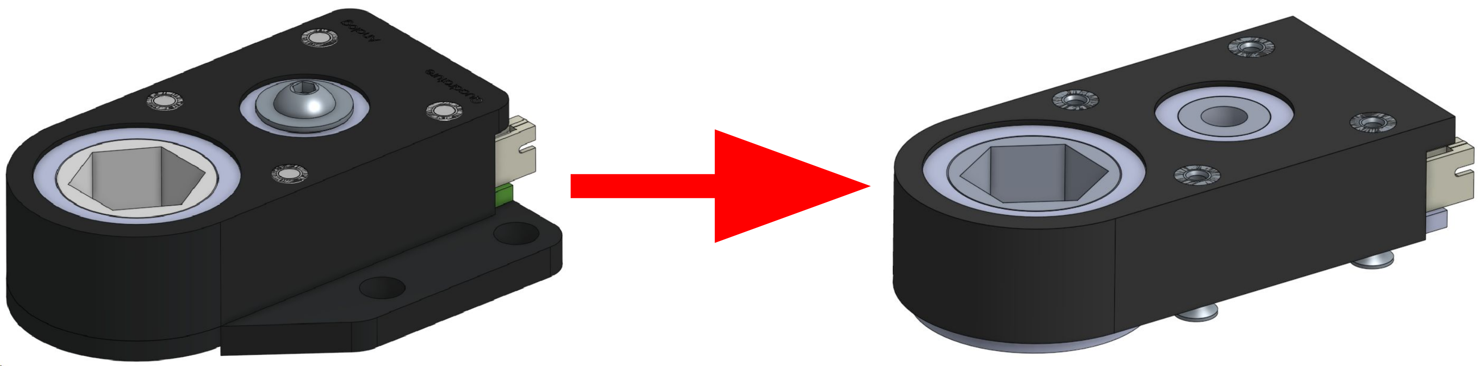
<table>
<tr><th colspan="4">BASE PARTS</th></tr>
<tr><th>PART NAME</th><th>SKU</th><th>QUANTITY</th><th>CHECK</th></tr>
<tr><td>35mm Rotacaster Omni (R or K) ^</td><td>R2-0354-5701</td><td>1</td><td></td></tr>
<tr><td>M4 Nut ^</td><td>2812-0004-0007</td><td>1</td><td></td></tr>
<tr><td>8mm ID 12mm OD Bearing ^</td><td>1601-0412-0006</td><td>3</td><td></td></tr>
<tr><td>M4x35 Countersunk Bolt ^</td><td>91294A202</td><td>1</td><td></td></tr>
<tr><td>M4 Nut (NOT NEEDED FOR M4x25 SCREW)</td><td>2812-0004-0007</td><td>0-3</td><td></td></tr>
<tr><th colspan="4">PLATE OPTIONS (ONE OF EACH, PAGE 3 FOR DETAILS)</th></tr>
<tr><td>M4x25 Screw ^</td><td>2800-0004-0025</td><td rowspan="3">3</td><td rowspan="3"></td></tr>
<tr><td>M4x22 Button-head Screw</td><td>2802-0004-0022</td></tr>
<tr><td>M4x22 Countersunk (FOR CNC ONLY)</td><td>91294A197</td></tr>
<tr><td>goBILDA Backplate</td><td>1105-0003-0080</td><td rowspan="3">1</td><td rowspan="3"></td></tr>
<tr><td>CNC Backplate ***</td><td>-</td></tr>
<tr><td>Printed Backplate *</td><td>-</td></tr>
<tr><th colspan="4">OTHER goBILDA VERSION-SPECIFIC PARTS</th></tr>
<tr><td>goBILDA Base Shim *</td><td>-</td><td>1</td><td></td></tr>
<tr><td>8mm ID 14mm OD Bearing</td><td>1611-0514-0008</td><td>1</td><td></td></tr>
</table>

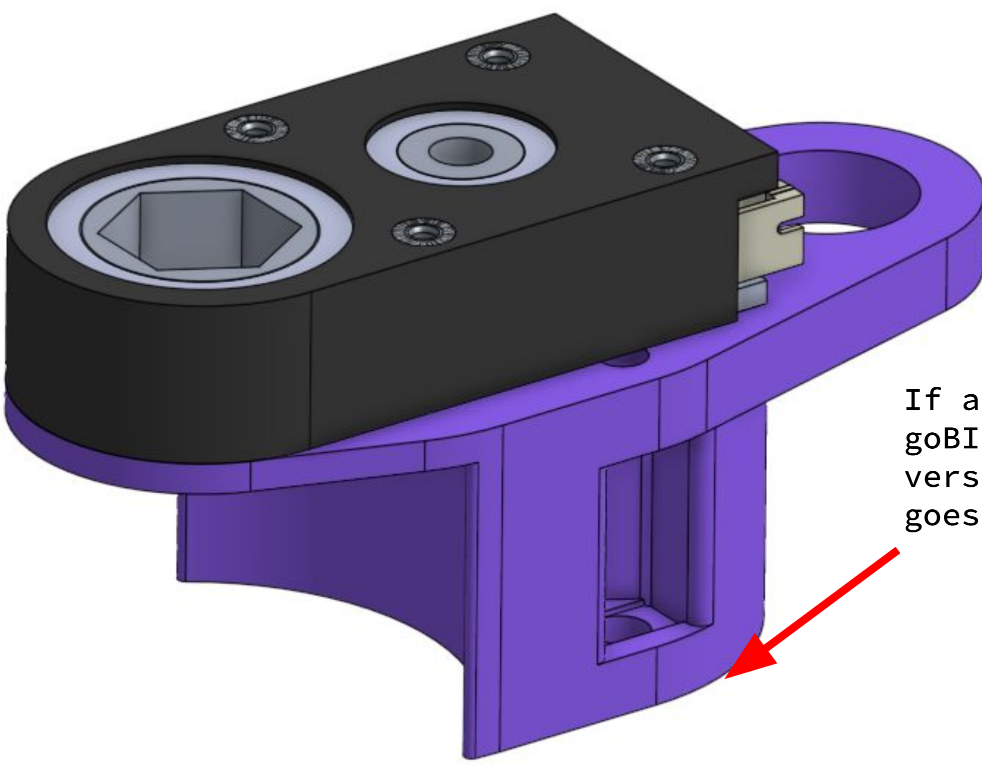
<table>
<tr><th colspan="6">ENCODER DEPENDENT PARTS</th></tr>
<tr><th>PART</th><th>REDUX V1</th><th>REDUX V2</th><th>REV THROUGH BORE</th><th>QUANTITY</th><th>CHECK</th></tr>
<tr><td>Base</td><td>Base RX1 *</td><td>Base RX2 *</td><td>Base RV *</td><td>1</td><td></td></tr>
<tr><th>BORE</th><th colspan="5">WHEEL INSERTS (ONE OF EACH)</th></tr>
<tr><td>Round</td><td colspan="2">Encoder Insert R11 *^</td><td>Encoder Insert R12 *</td><td rowspan="2">1</td><td rowspan="2"></td></tr>
<tr><td>Keyed</td><td colspan="2">Encoder Insert K11 *^</td><td>Encoder Insert K12 *</td></tr>
<tr><td>Round</td><td colspan="3">Bearing Insert R *^</td><td rowspan="2">1</td><td rowspan="2"></td></tr>
<tr><td>Keyed</td><td colspan="3">Bearing Insert K *^</td></tr>
</table>

^ Included in the CODEX Odometry Bundle, * Requires 3D Printing, * Advanced Machining**

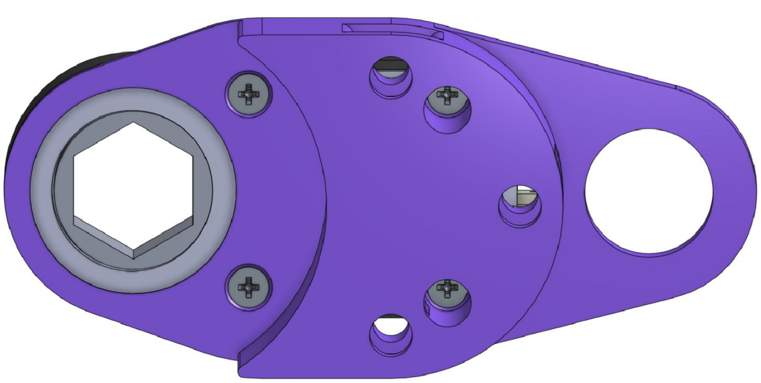
REDUX ASSEMBLY

The first step requires the disassembly of the encoder. The backplate must be removed. The stock M2 bolts are then used to mount the encoder to the 3D printed Base part.





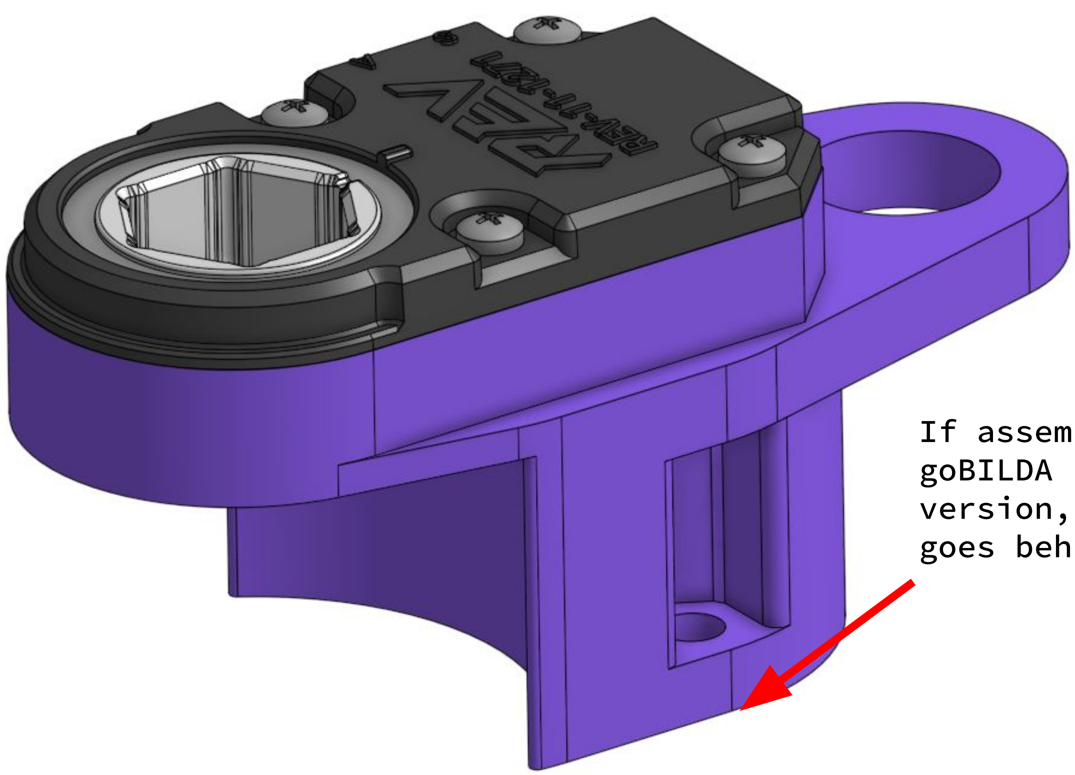
If assembling goBILDA backplate version, shim goes behind here



NOTE: Your encoder may look slightly different depending on when it was ordered, but all versions are functionally the same (encoder pictured is Redux v1)

REV ASSEMBLY

The first step requires the disassembly of the encoder. The back piece must be removed, and all the internal components must be transferred to the Base. The bearing from the back piece of the encoder must also be transferred to the Base. These pictures show the process.

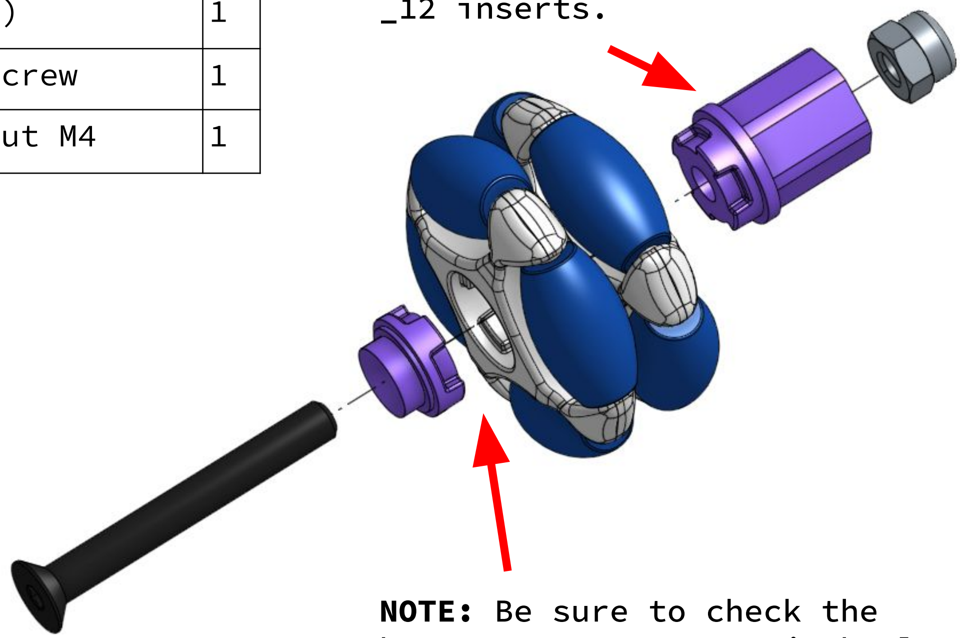


If assembling goBILDA backplate version, shim goes behind here

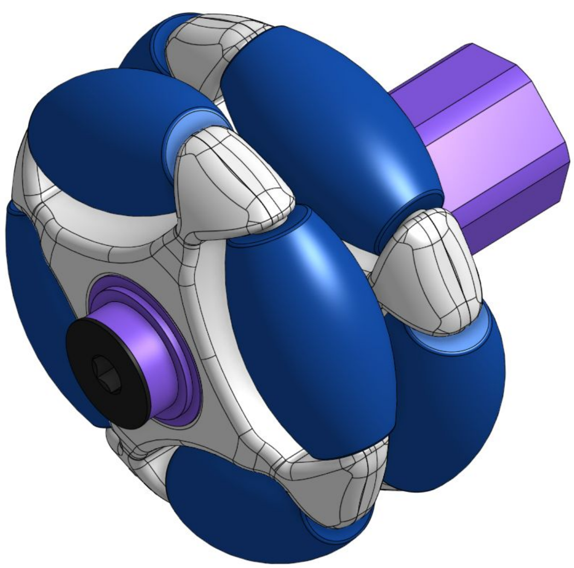
AXLE ASSEMBLY

Rotacaster 35mm Omni	1
Encoder Insert (ANY)	1
Bearing Insert (ANY)	1
M4x35 countersunk screw	1
Prevailing torque nut M4	1

NOTE: The Redux encoder requires _11 and the REV Through Bore uses _12 inserts.



NOTE: Be sure to check the bore on your 35mm omni wheel and determine the correct inserts (R for round, K for keyed)



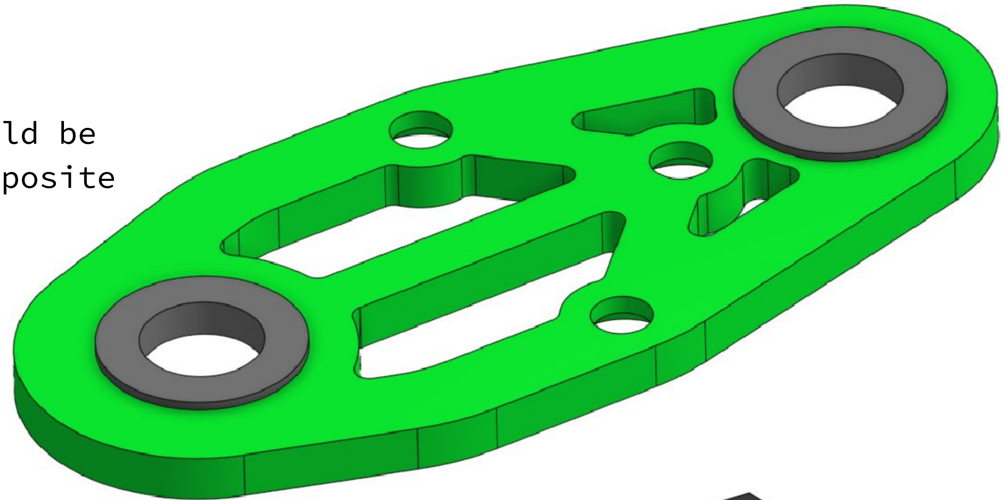
EXTRA LESSON: DESIGNING AXIAL ASSEMBLIES

The most important part of designing assemblies like this is to make sure they are axially constrained. This means ensuring that they cannot move side to side and come disassembled. In this assembly, the bearing in the encoder prevents the axle from sliding out in that direction, and the bearing in the backplate prevents it from sliding the other way. The screws through the assembly make sure the encoder and backplate do not separate.

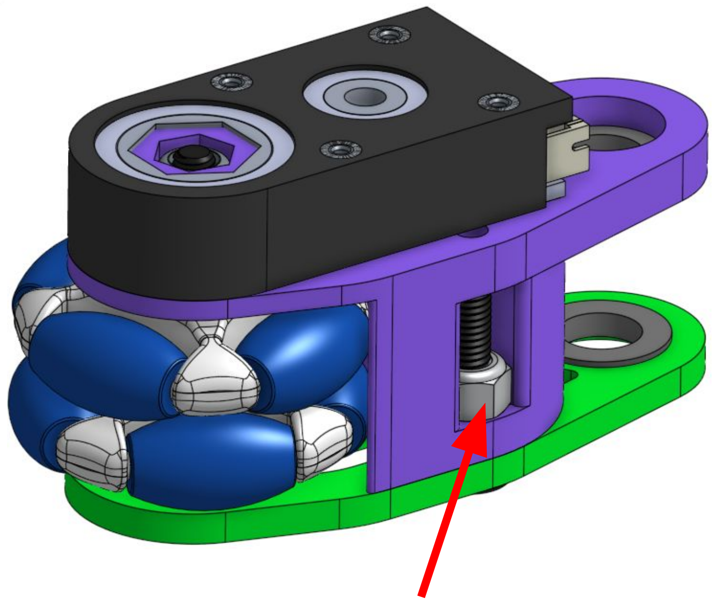
There are many tricks to constraining assemblies axially, but doing a mental check of all the ways a thing could fall apart is a good idea before manufacturing the design.

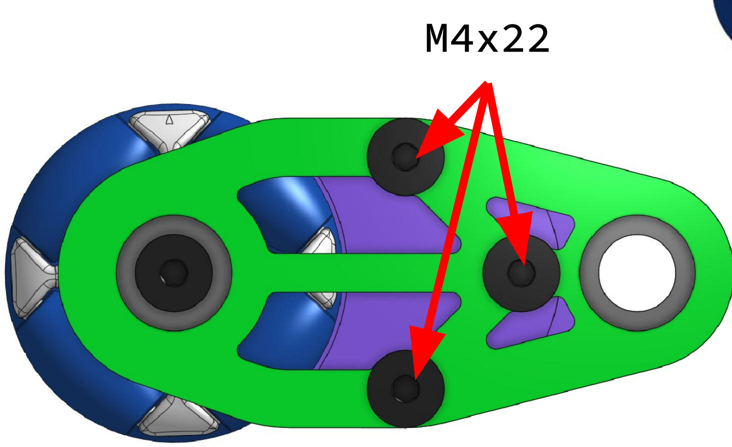
BACKPLATE ASSEMBLY

NOTE: Bearings should be inserted on side opposite countersinks

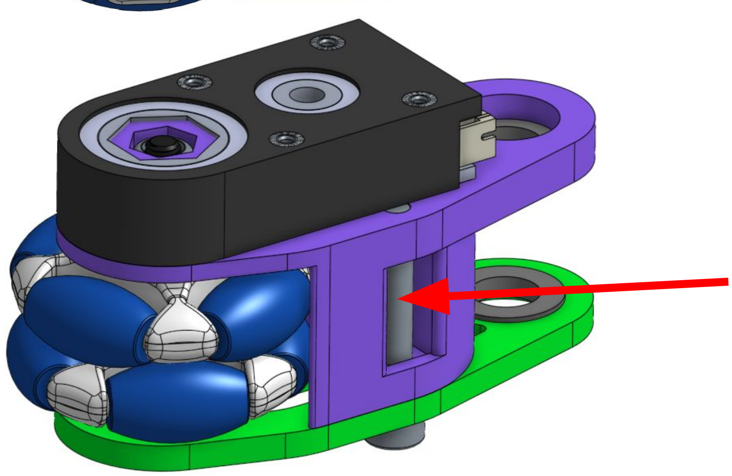


Countersunk M4x22	3
M4 Nut	3



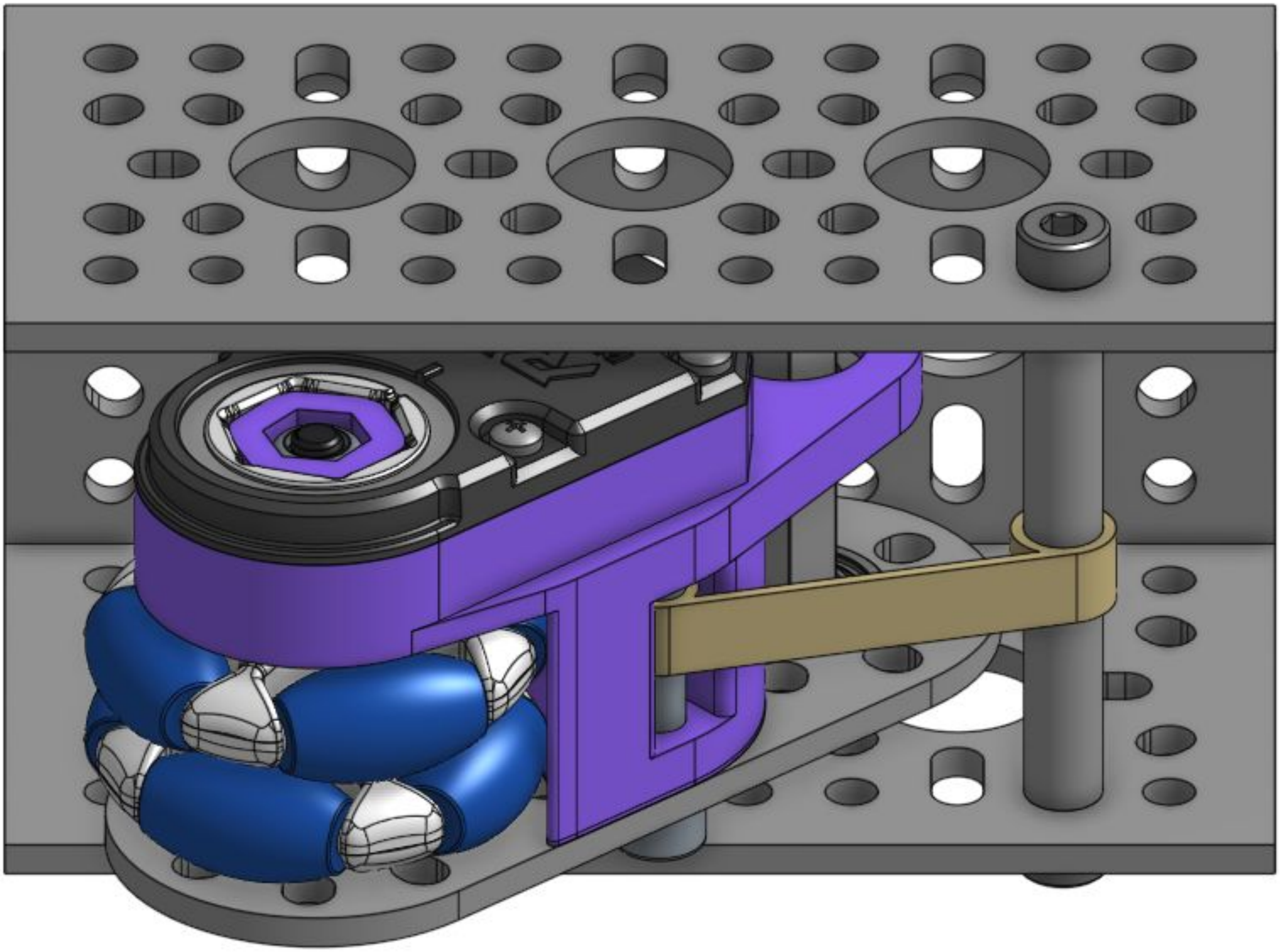


M4 nuts should be inserted into the nex-shaped pockets around the Base. The protruding screw can be used to spring the odometry pod later.



NOTE: This design can also be built using the stock M4x25 socket head bolts from the Codex odometry bundle. Using these does not require nuts in the hex-shaped pockets because the screws thread into the plastic.

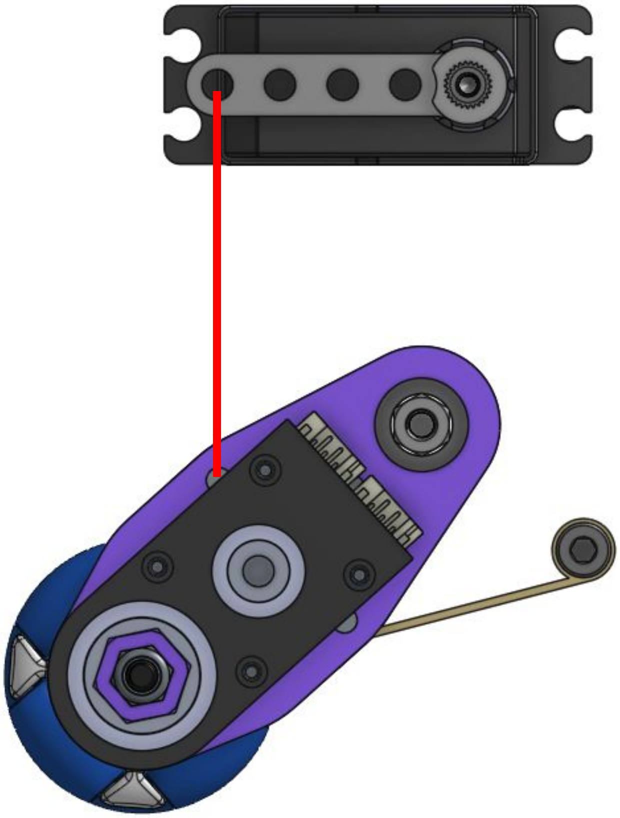
SPRINGING



Odometry only works if the dead wheels are in constant contact with the ground. For this reason, any of the three holes in the sides of LoonyOdo can be used for springing. The encoder is readily symmetric, so you can flip it around to use on either side of the robot without needing to reassemble. The simplest method of springing is attaching a rubber band to the bottom screw as shown. This method works with both custom and stock screws. A tension spring also works, but may require modification of the Base depending on the diameter.

Recommended rubber bands for springing (size #61):
https://www.amazon.com/Alliance-26649-Advantage-Contains-Approx/dp/B008X09P96/ref=sr_1_2?crid=2UWKNPBEJHKI8&keywords=rubber%2Bband%2B1%2F4%22&qid=1670262655&sprefix=rubber%2Bband%2B1%2F4%2B%2Caps%2C148&sr=8-2&th=1

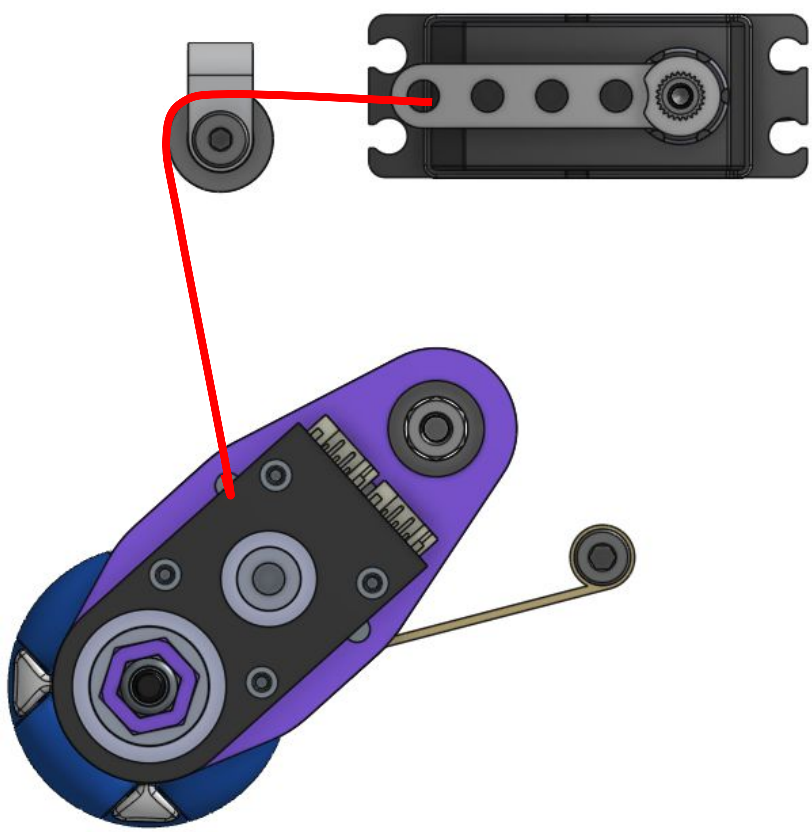
RETRACTABLE ODO

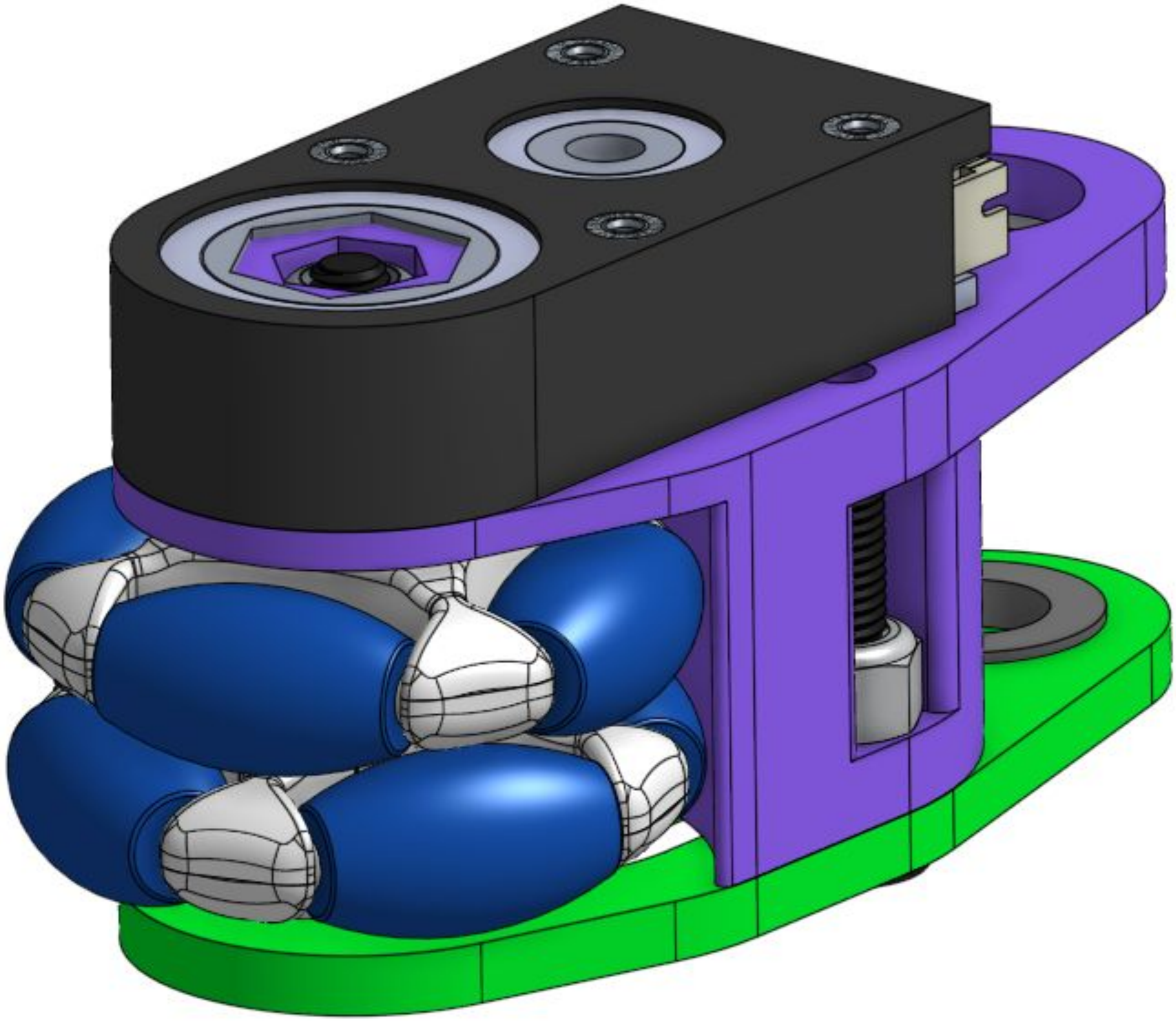


For games with terrain, such as Freight Frenzy, you may want to make your odometry retractable to prevent damage. LoonyOdo can be made retractable by attaching a string to the top screw, opposite the spring mount. The string can be run directly to a servo or through a system of pulleys. The latter allows the use of one servo to lift all odometry pods, which has the benefit of less power consumption and using 2 fewer servo ports at the cost of added complexity and having to deal with long cable runs.

Recommended retraction string:
https://www.amazon.com/gp/product/B07BKQLFRB/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&th=1

NOTE: You will still need a spring to run retractable odometry to keep them firmly planted on the ground when down. The servos will need to be strong enough to extend whatever spring you are using.





ADDITIONAL INFORMATION

CAD:
https://cad.onshape.com/documents/d3f71dd21473f1f7cb90f46a/w/c442869b72fff7747c68e8ef/e/66b8575e64647fa3550fca9c

CONTACT: b1nary#7727, G-Force#5900 on Discord

NEW PROJECTS: https://twitter.com/the_loony_squad
https://www.instagram.com/the_loony_squad/

FILE ACCESS: https://forms.gle/WupLkVazXi4h8t2j6