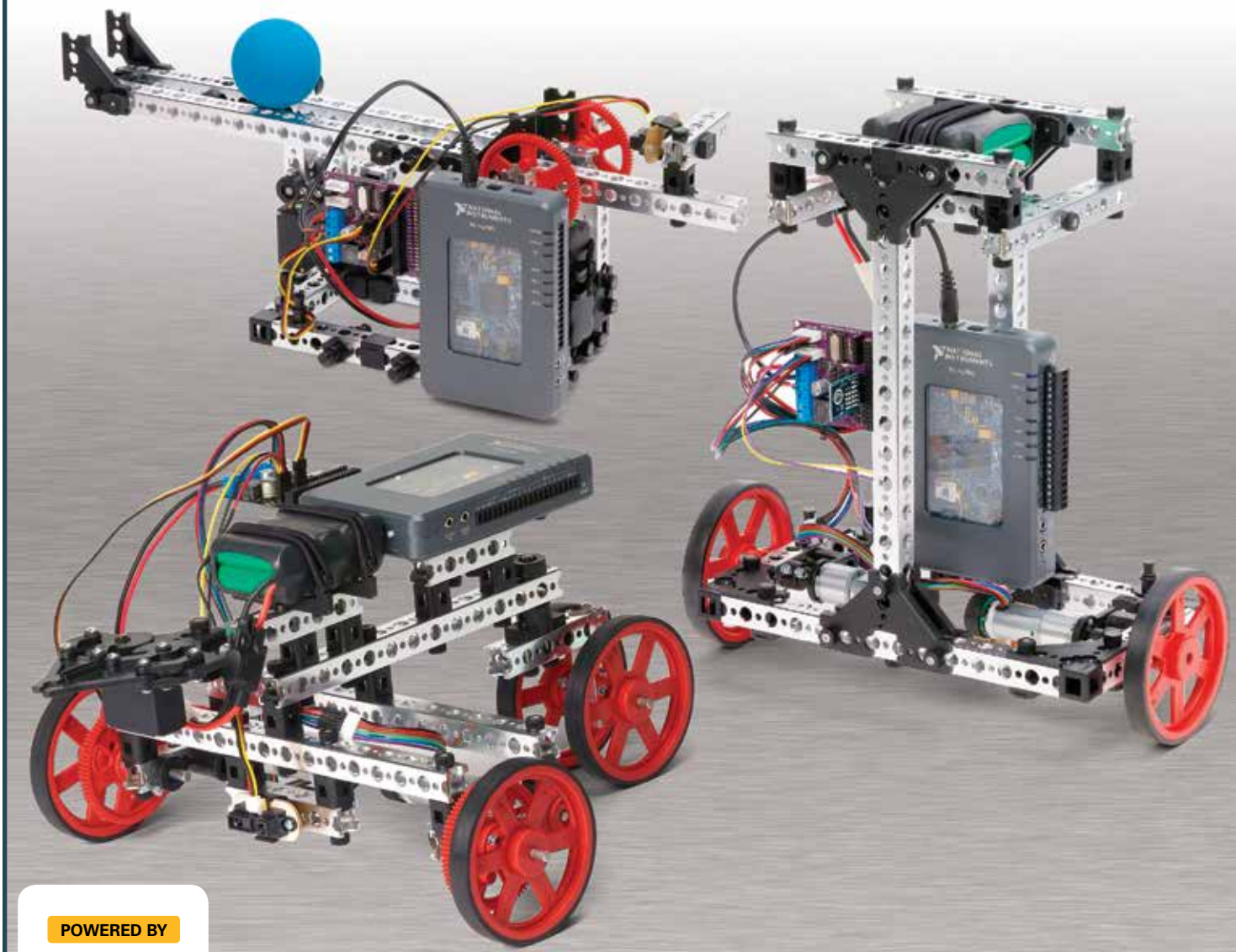
PITSCO

TETRIX® PRIME

Robot Builder's Guide

for NI myRIO



POWERED BY

NATIONAL INSTRUMENTS LabVIEW

PITSCO EDUCATION

41427

Contents

Content Advising by Paul Uttley, Professor R. H. Bishop of the University of South Florida, and National Instruments

Graphics by Todd McGeorge.

©2015 Pitsco, Inc., 915 E. Jefferson, Pittsburg, KS 66762

All rights reserved. This product and related documentation are protected by copyright and are distributed under licenses restricting their use, copying, and distribution. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Pitsco, Inc.

All other product names mentioned herein might be the trademarks of their respective owners.

Back to Contents page

Preface

This builder's guide has been developed to provide students with positive experiences in robotics engineering. With the assistance of this guide, students will learn to use the TETRIX PRIME parts and myRIO to design and construct three different robots. After using this guide, students should be able to use the TETRIX PRIME parts to construct a robot of their own design.

Safety Information

Mechanical

- Keep fingers, hair, and loose articles of clothing clear of gears and moving parts.
- Never pick up the robot while it is moving or the servo motors are running.
- Remove any burrs caused by cutting the metal beams.

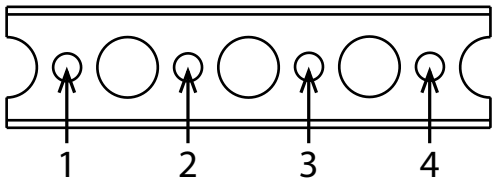
Electrical

- Make sure the power is turned off when the robot is not in operation.
- Do not operate the robot in a wet environment.
- Always power down the robot before making any changes.
- Use caution when working with bare wires to avoid creating a short circuit situation.
- Route wires carefully and secure them if necessary to avoid damage to the wire or its insulation.
- Mount the battery pack and all electronic components securely.

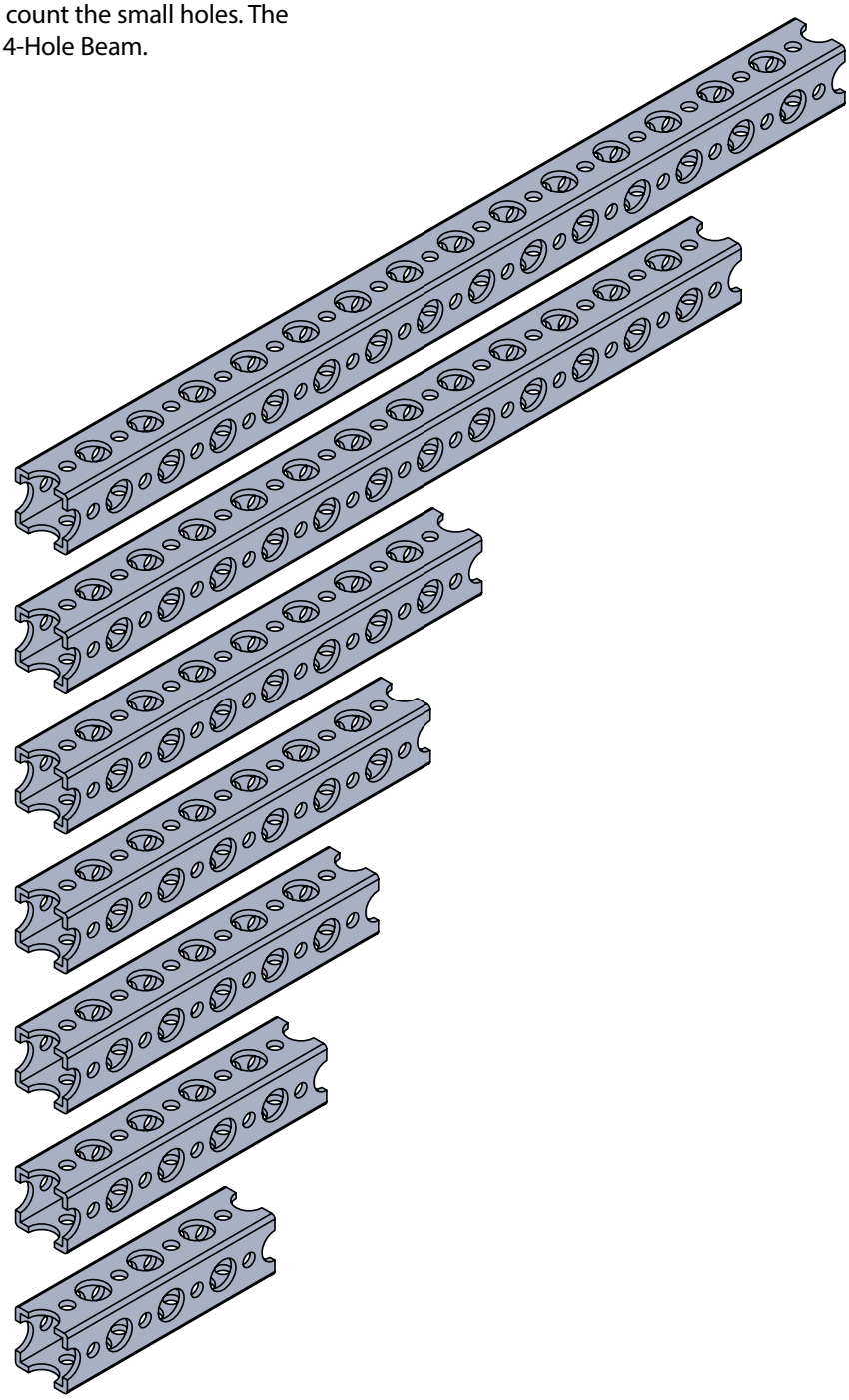
Set Components

This section documents all the parts included in the kit, including images, quantities, and part numbers. The part number of each part will match the part number noted on the label of each part bag. The part bags may contain one or more parts, and there may be more than one bag of a particular type of part. Instructions for purchasing additional parts are found at the end of this section.

The beams are named by the number of small holes on one side of the beam. Do not select beams by counting the larger holes.



To identify TETRIX PRIME beams, count the small holes. The example above is a 4-Hole Beam.



2x – 15-Hole Beam 40207

2x – 13-Hole Beam 40206

4x – 8-Hole Beam 40205

4x – 7-Hole Beam 40204

4x – 6-Hole Beam 40203

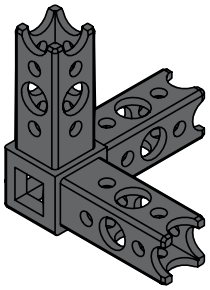
4x – 5-Hole Beam 40202

4x – 4-Hole Beam 40201

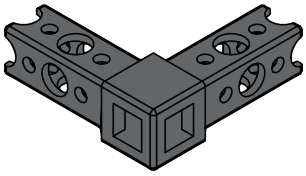
Back to Contents page

Mechanical Parts

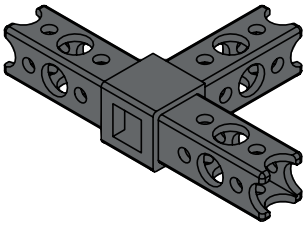
4x – 3-Way Beam Connector 40212



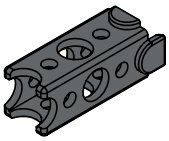
4x – 90-Degree Beam Connector 40211



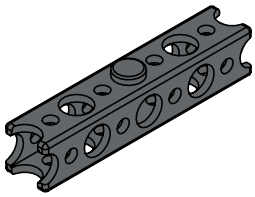
4x – Tee Beam Connector 40213



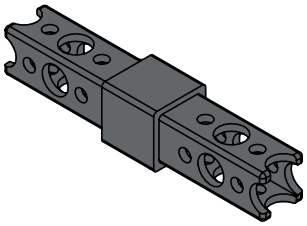
4x – Beam End Connector 40214



4x – Beam Extension 40322

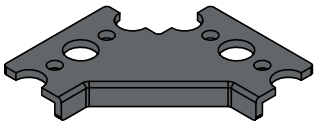


4x – Straight Beam Connector 40215

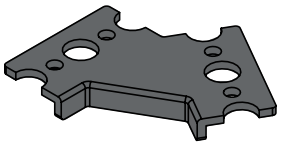


Mechanical Parts

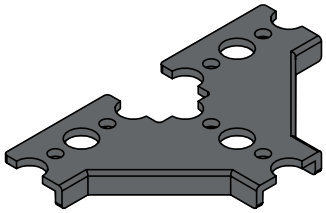
10x – 90-Degree Bracket 40208



10x – 60-Degree Bracket 40209



10x – Tee Bracket 40210



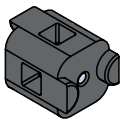
24x – Quick Rivet Peg 40220

24x – Quick Rivet Connector 40219

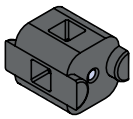
50x – Thumbscrew 40323

24x – Wing Nut 40221

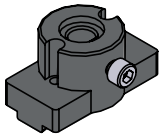
10x – 90-Degree Cross Block Connector 40217



10x – Straight Block Connector 40216

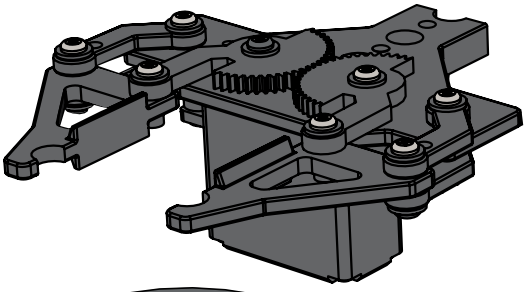


4x – Beam Attachment Hub 40228

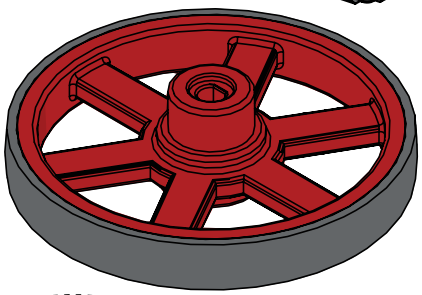


Mechanical Parts

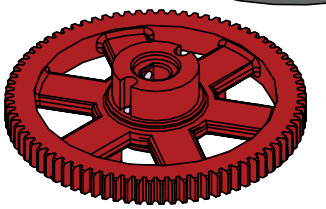
1x – PRIME Gripper Assembly 40234



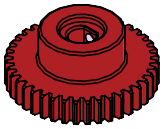
4x – 90 mm Wheel & Tire 40222



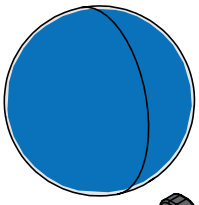
4x – 80-Tooth Gear 40224



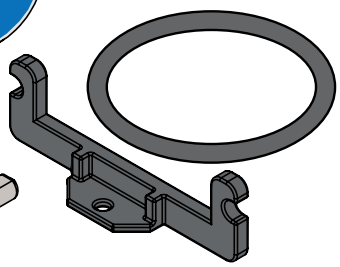
4x – 40-Tooth Gear 40223



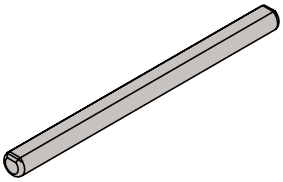
1x – Blue Racquetball 43121



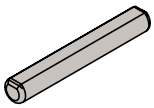
2x – Battery Holder and Bungee 40236



6x – 80 mm Axle 40225

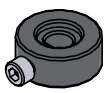


6x – 40 mm Axle 40226



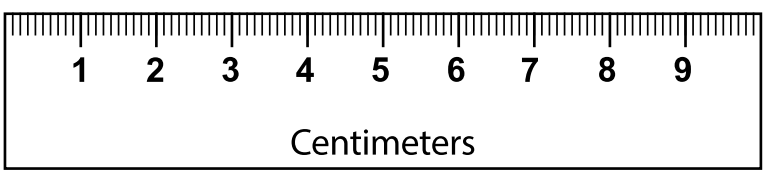
12x – Bronze Bushing 40227

8x – Shaft Collar 40229

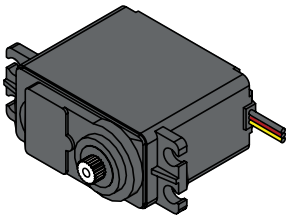


25x – Setscrew 40516

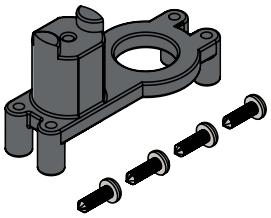
Measure 80 mm and 40 mm axles here.



Motors

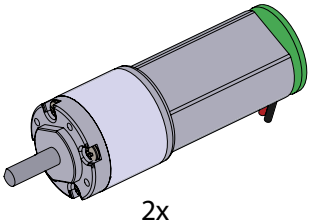


2x
Standard Servo 40538

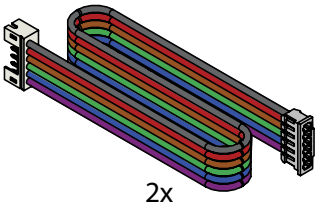


2x
Servo Mount 40232

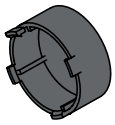
2x
Servo Shaft Hub 40230



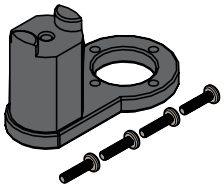
2x
Digilent DC
Gear Motor
41143



2x
DC Motor
Extension Cable
41147



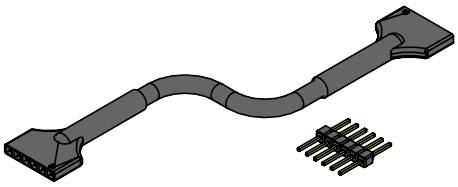
2x
DC Motor
Encoder Cap
41231



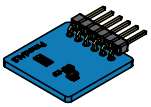
2x
Digilent DC
Motor
Mount 41138

2x
Digilent Motor
Shaft
Adapter 41139

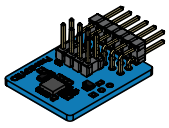
Sensors



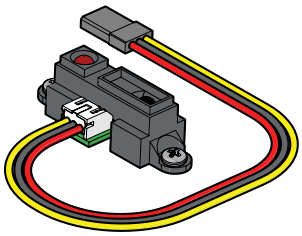
1x
Digilent Ambient
Light Sensor Kit 41197



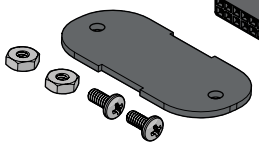
1x
Digilent
Ambient Light
Sensor 41146



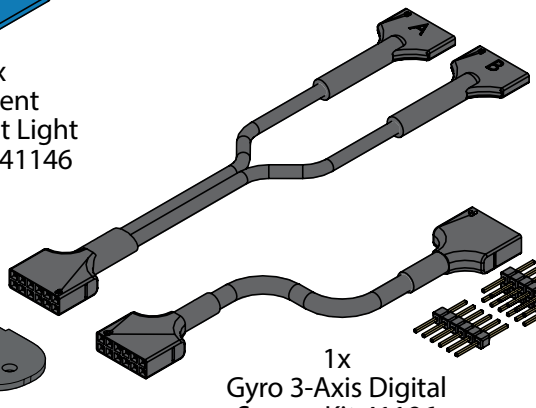
1x
Digilent Gyro
Sensor
41144



1x
Digilent Sensor
IR Rangefinder 41145

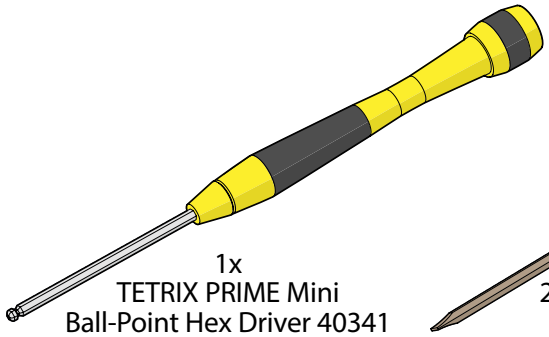


1x
Digilent Sensor
IR Rangefinder Mounting
Plate 41308

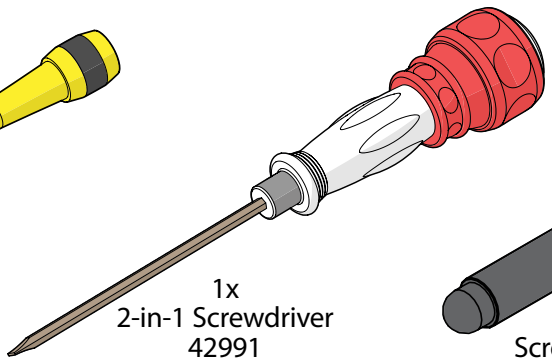


1x
Gyro 3-Axis Digital
Sensor Kit 41196

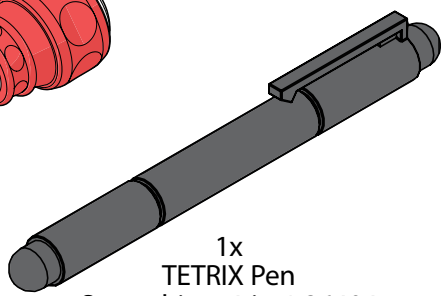
Additional Tools



1x
TETRIX PRIME Mini
Ball-Point Hex Driver 40341



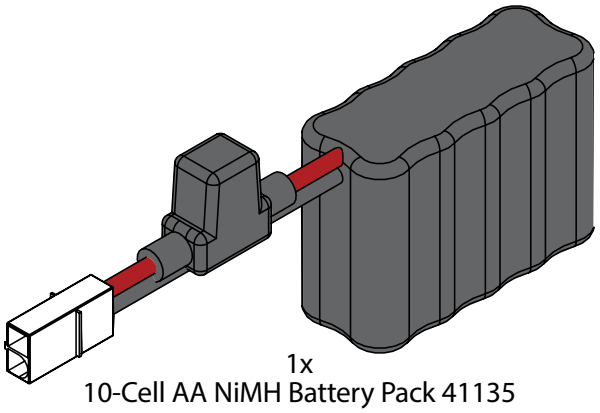
1x
2-in-1 Screwdriver
42991



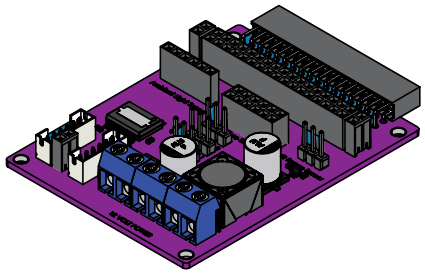
1x
TETRIX Pen
Screwdriver 4-in-1 36404

Back to Contents page

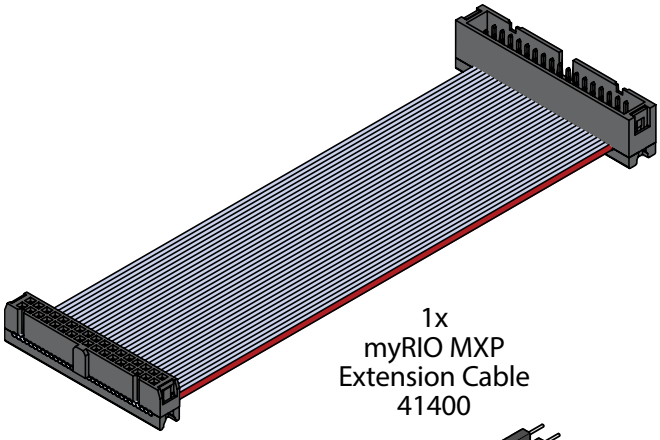
myRIO Accessories



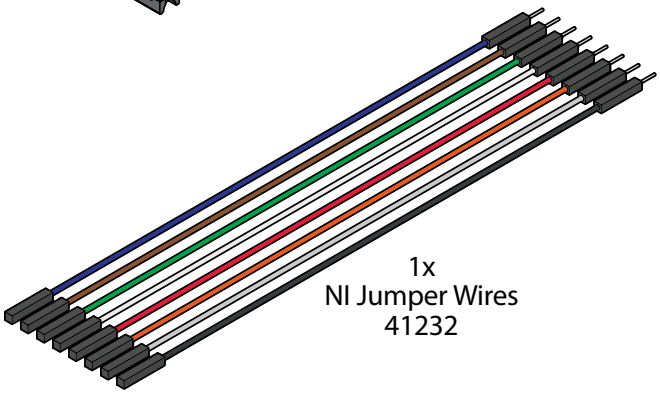
1x
10-Cell AA NiMH Battery Pack 41135



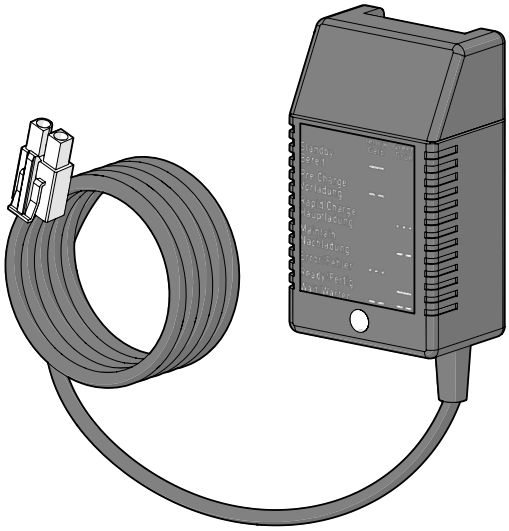
1x
myRIO Motor Controller Board 41136



1x
myRIO MXP
Extension Cable
41400

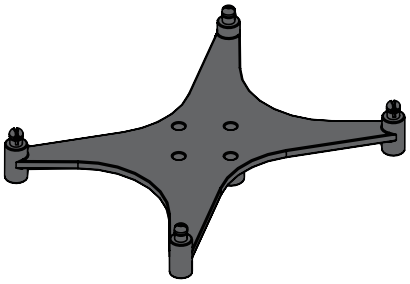


1x
NI Jumper Wires
41232

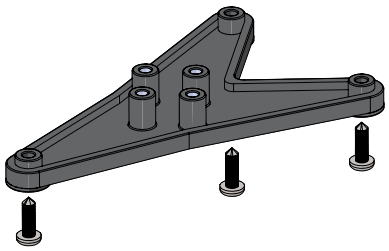


1x
10-Cell AA NiMH Battery Quick Charger 41142

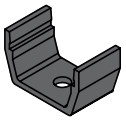
Brackets and Mounts



1x
myRIO Control Board
Mount Bracket 41140



1x
myRIO Controller Mount 41137



3x
Digilent Sensor
Mounting Clip 41148

*myRIO sold separately. For more information, visit ni.com/myrio.

Replacing and Supplementing Kit Parts

Individual kit parts can be purchased from either Pitsco Education or Digilent, Inc.

Purchase parts from Pitsco Education

All parts listed in this section, except those included in the Digilent part list below, can be purchased from Pitsco Education. Part numbers for proprietary PITSCO products are listed in under each part earlier in this section.

Visit Pitsco.com and search using the part number to find and purchase parts.

Purchase Parts from Digilent, Inc.

Digilent parts are listed below and are also pictured earlier in this section. Throughout this guide and on each individual packing label, they are identified by a PITSCO part number. Corresponding Digilent

part numbers are provided below.

Part Name	Pitsco Part Number	Digilent Part Number
DC Gear Motor	41143	290-008P
DC Motor Extension Cable	41147	250-084
Ambient Light Sensor	41146	410-286P
Ambient Light Sensor cable kit	41197	240-021-2 (Cable Kit 12 Pin)
Gyro cable kit	41196	240-021-1
Gyroscope	41144	410-215P
IR Rangefinder	41145	240-037P

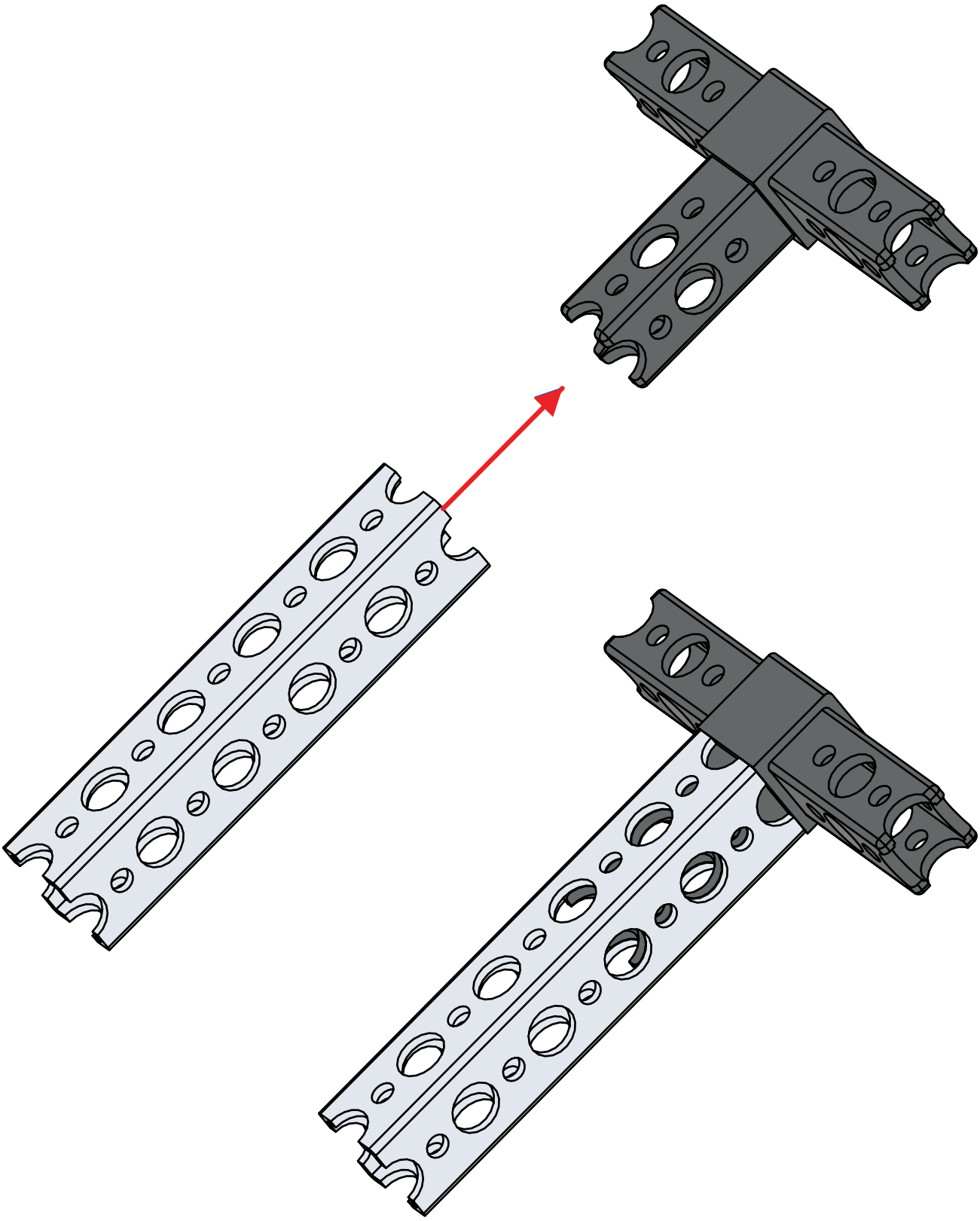
Visit Digilent.com and search using the Digilent part number to find and purchase parts.

Note: When purchasing outside of the United States, you may consider purchasing from Digilent distributers to avoid additional charges. Find a list of Digilent distributers here: http://store.digilentinc.com/our-distributors/

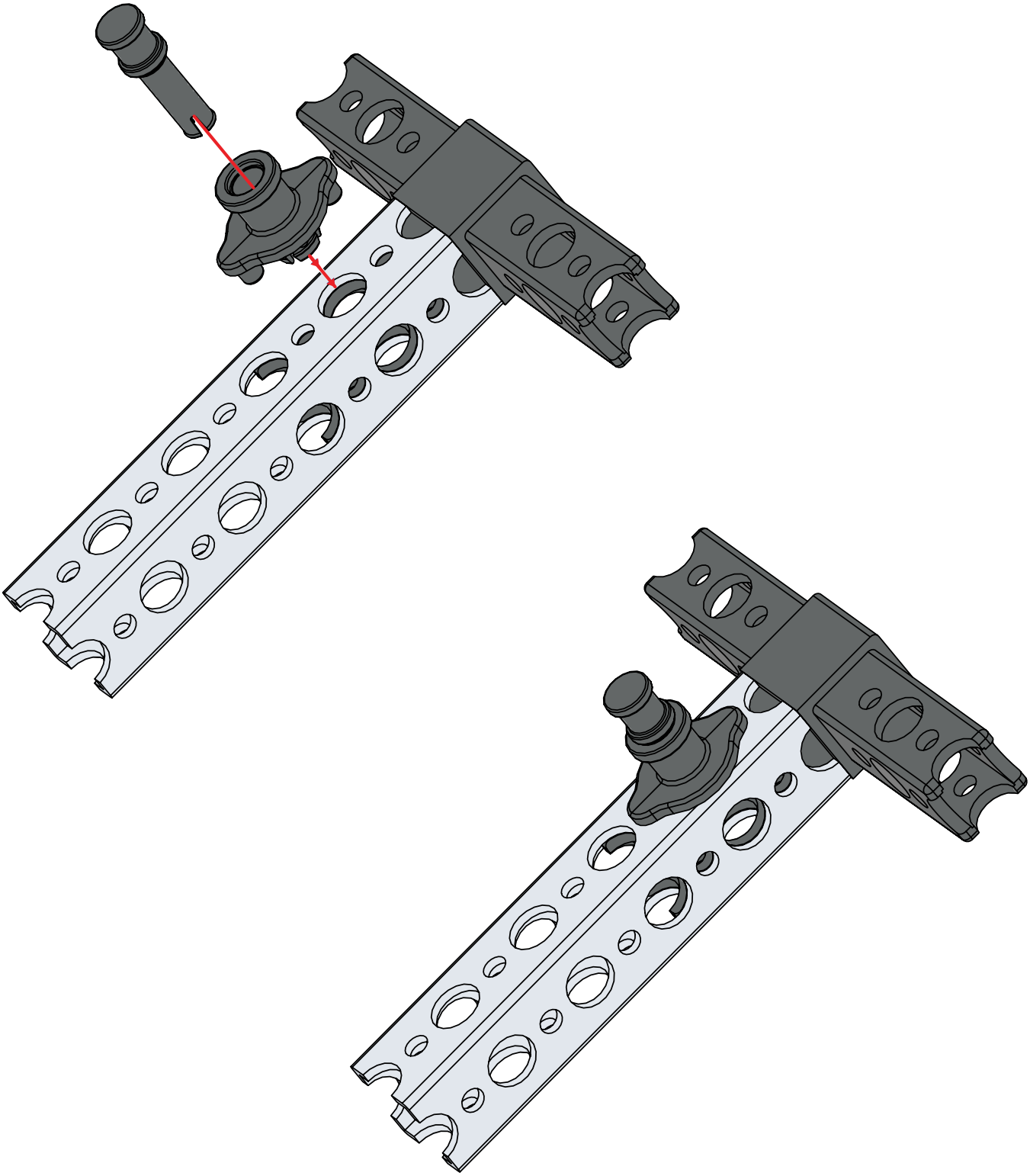
Back to Contents page

Introduction to Building with TETRIX® PRIME

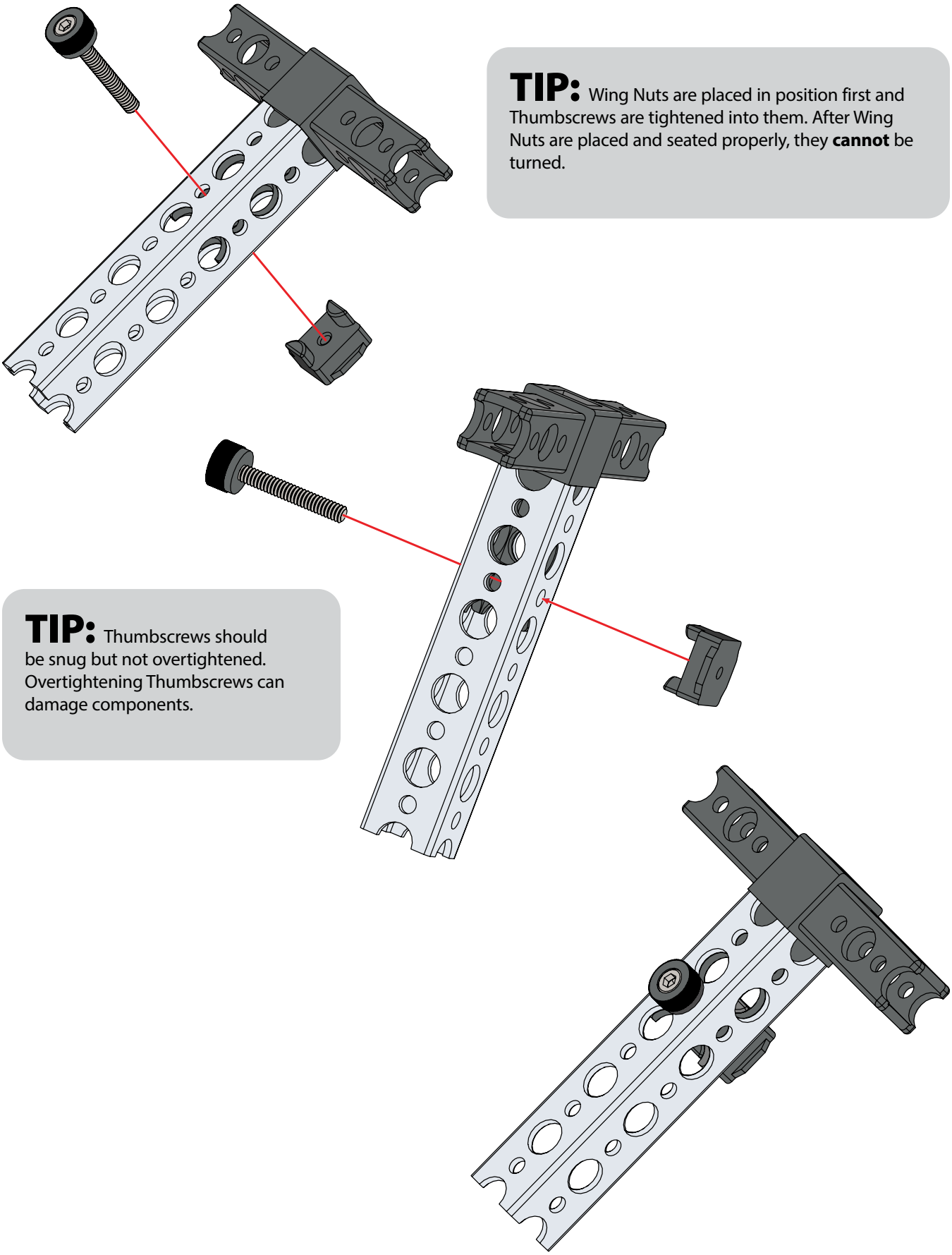
Connectors fit inside beams and come in straight, 90-degree, tee, and corner designs.



Quick rivets and pegs are a quick option for securing connectors. Press the rivet in place on the beam and use the peg to spread the rivet to secure the connection. Using quick rivets on two sides of the connection will make it more stable.



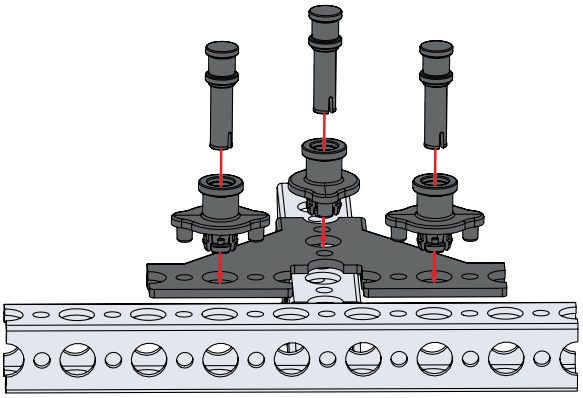
Joints can be made more permanent by using a Thumbscrew and Wing Nut to secure the beams and connectors.

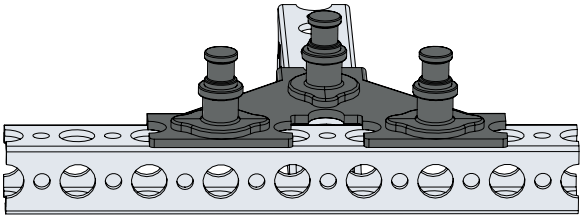


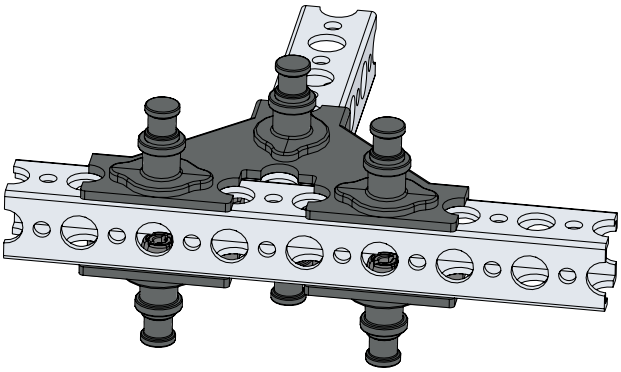
TIP: Wing Nuts are placed in position first and Thumbscrews are tightened into them. After Wing Nuts are placed and seated properly, they **cannot** be turned.

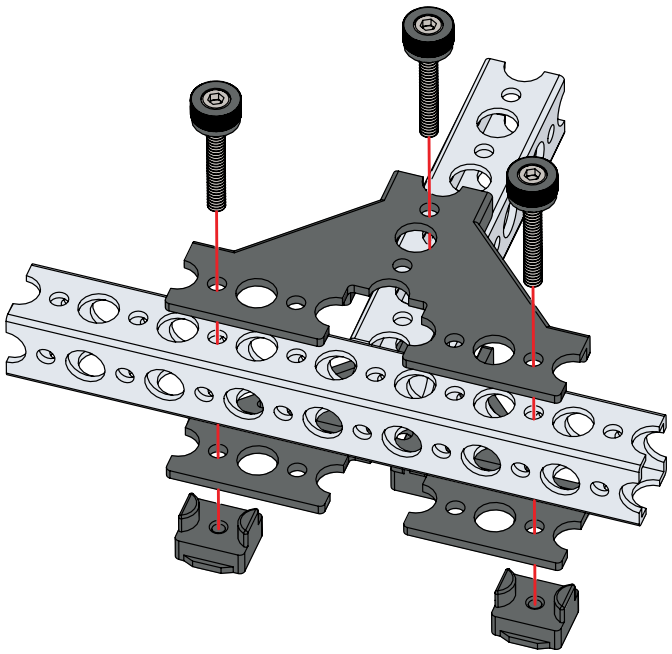
TIP: Thumbscrews should be snug but not overtightened. Overtightening Thumbscrews can damage components.

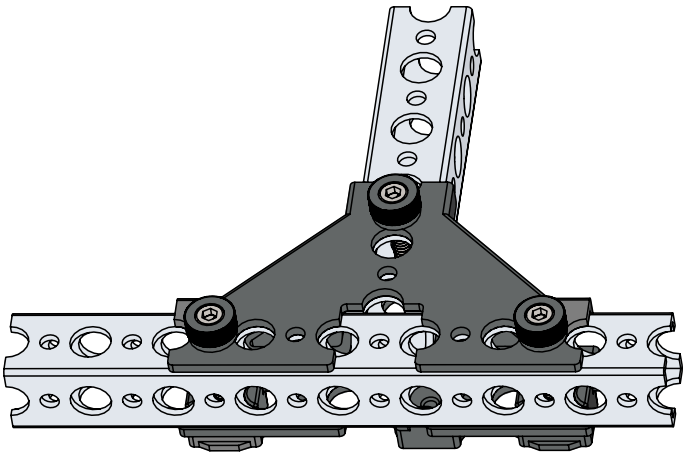
Brackets can also be used to connect beams. Brackets are available for a tee connection, 60-degree connection, or 90-degree connection. Brackets should be used in pairs, with two brackets on opposite sides of a beam. Brackets are secured using quick rivets and pegs or thumbscrews and wing nuts.



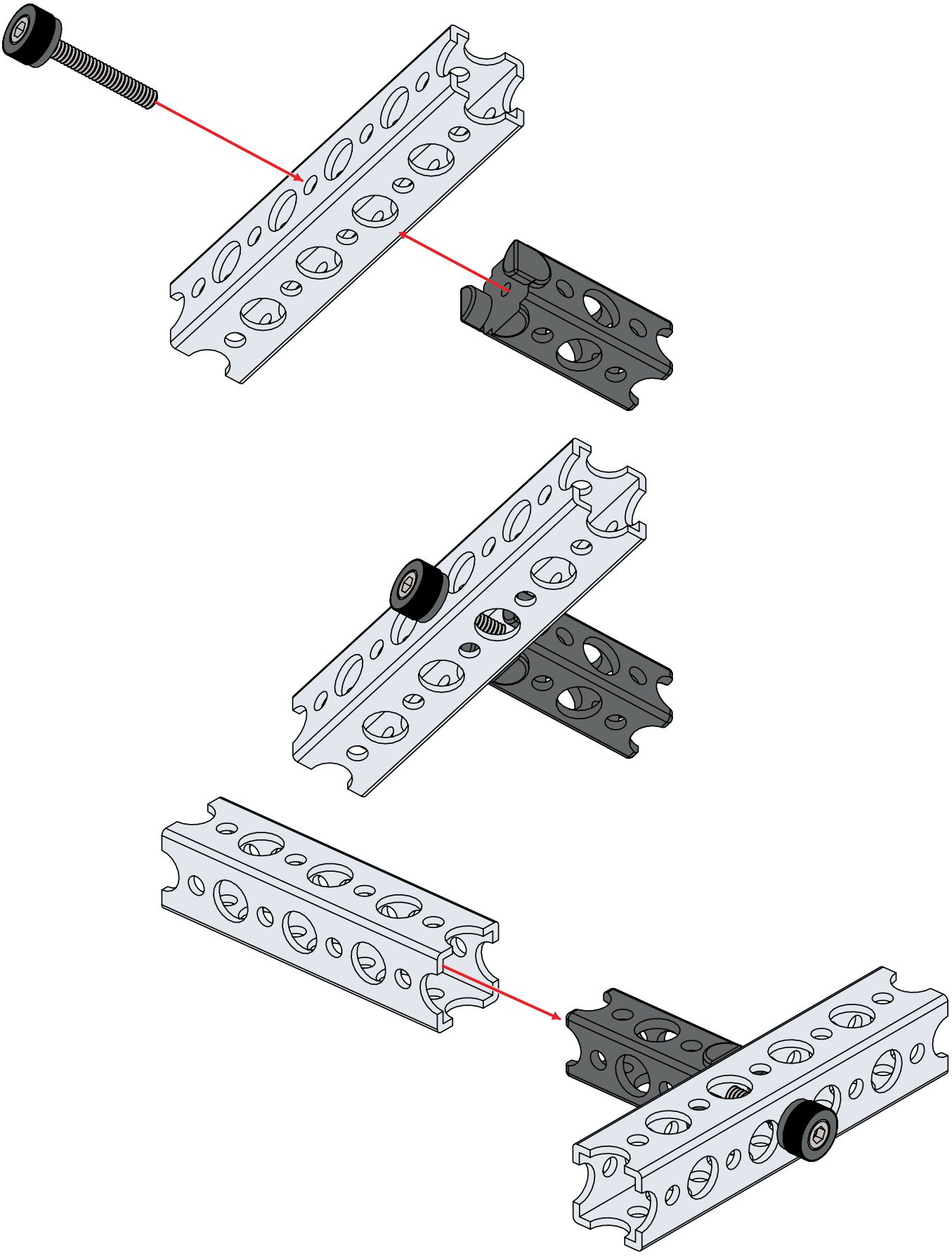




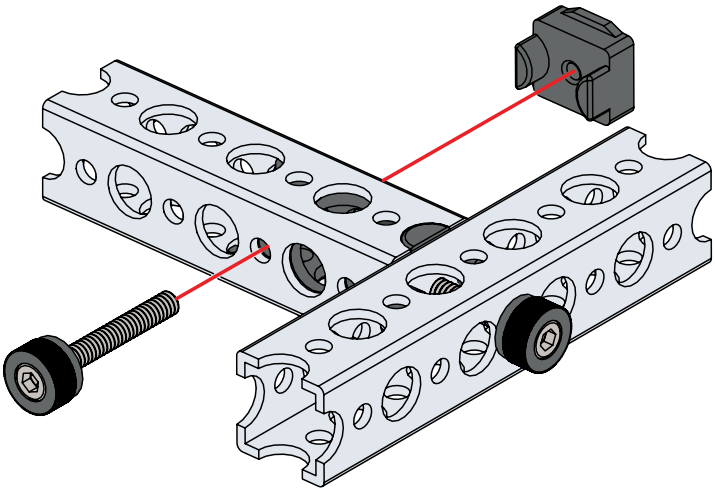


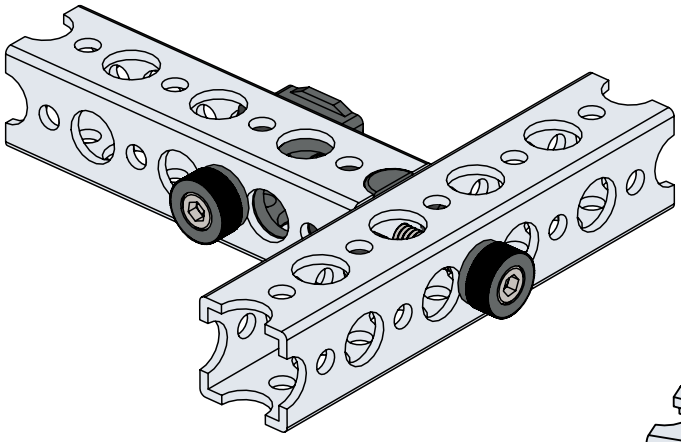


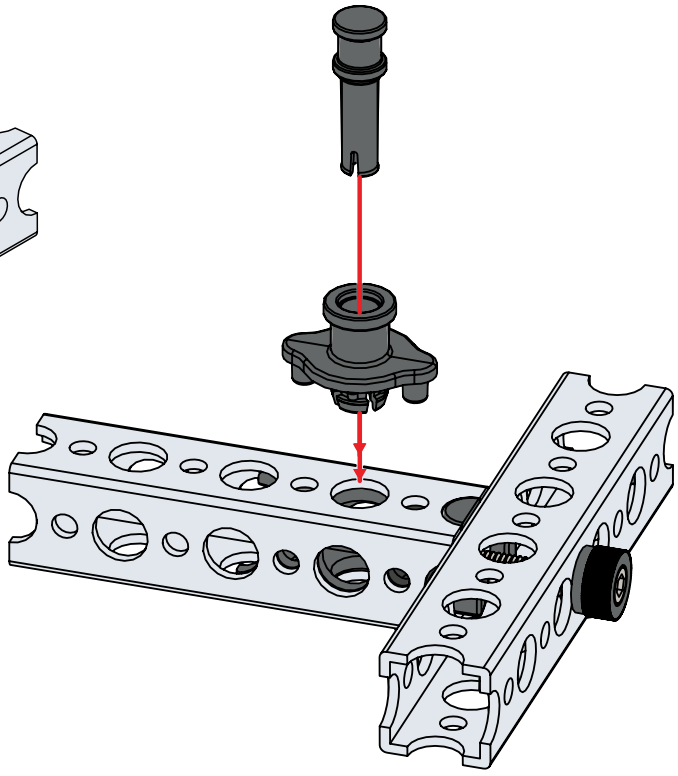
End connectors, straight block connectors, and 90-degree cross block connectors are secured using a thumbscrew through the beam and into the connector.

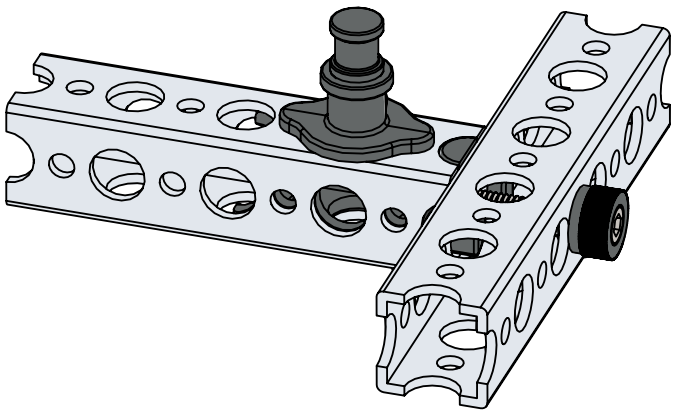


After the thumbscrew is used to secure the end of the connector, a quick rivet and peg or a thumbscrew and wing nut are used to secure the intersecting beam.

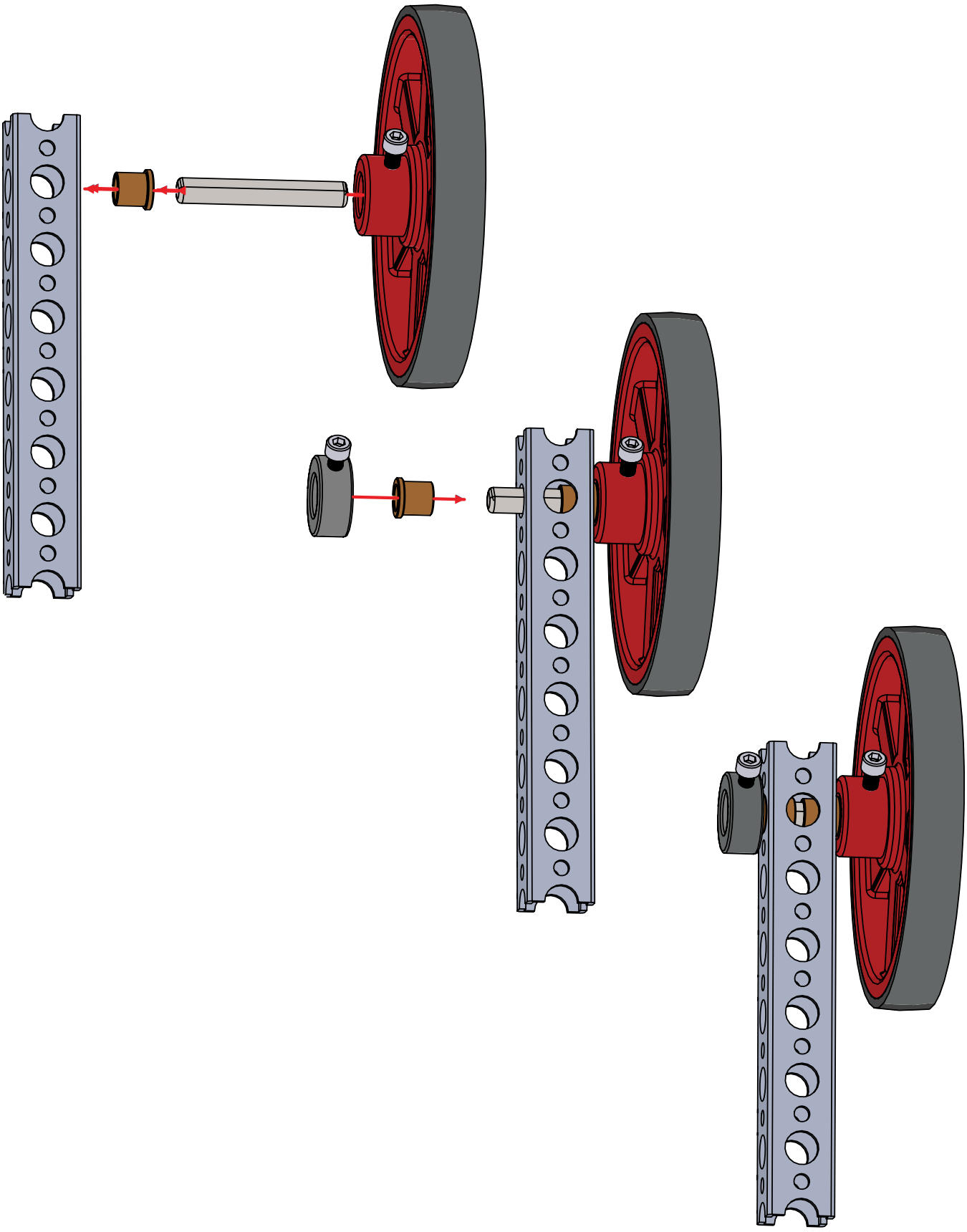






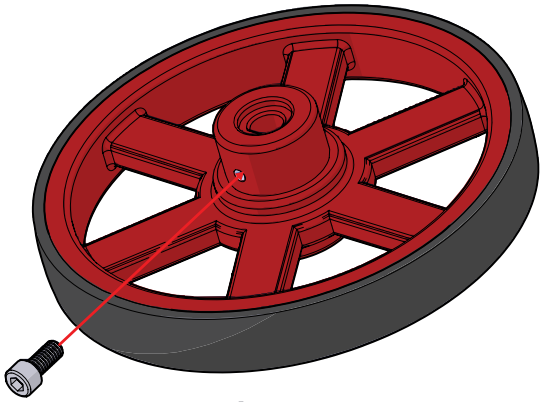


Anytime an axle is used, it should be supported at two points. Place a bronze bushing on opposite sides of a beam and place the axle through the bushings. Secure the axle to a stop collar, wheel, gear, or hub.

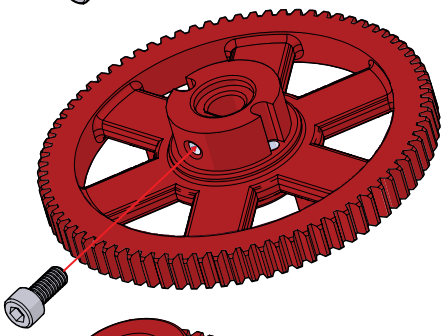


Setscrews are used to secure axles inside the hubs of wheels, collars, and connectors.

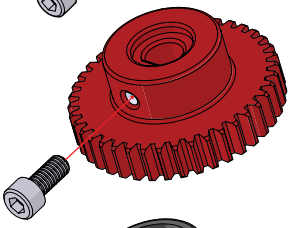
Wheel with Tire (90 mm) 40222



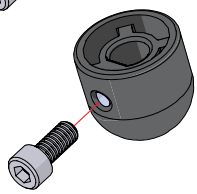
80-Tooth Plastic Gear 40224



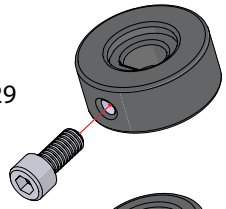
40-Tooth Plastic Gear 40223



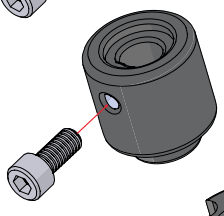
16 mm Ball Skid Point 41260



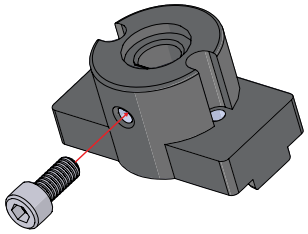
D-Shaft Set Collar 40229



Shaft Servo Hub 40230



Beam Attachment Hub 40228



TIP: Because some parts from the TETRIX PRIME R/C Starter Set might already have Socket Head Cap Screws installed, the directions for the expansion builds might or might not show installing Socket Head Cap Screws. Please keep in mind that if your parts already have Socket Head Cap Screws installed and the directions call for that operation, you can skip to the next step.

TIP: Directions will show these types of parts in proper position. Please remember after the parts are in place to tighten the Socket Head Cap Screws before moving to the next step.

TIP: Socket Head Cap Screws should be snug but not overtightened. Overtightening Socket Head Cap Screws can damage components.

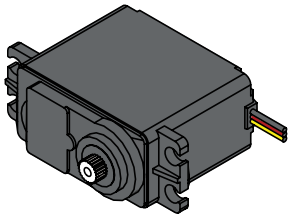
Subassembly Construction Instructions

This section provides instructions for building common subassemblies. Many of these subassemblies are referenced in the assembly build instructions sections. The subassembly instructions may also be helpful for building new designs.

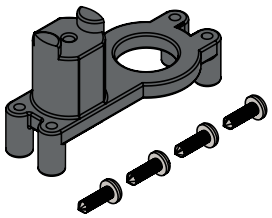
Standard Servo Mount Assembly

You will need the HS-322HD standard servo with screw, the servo mount with screws, a servo shaft hub, and a setscrew. You will also need the TETRIX Pen Screwdriver 4-in-1 and TETRIX PRIME Mini Ball-Point Hex Driver. Remove the white plastic servo horn attached to the servo. Retain the screw for future use, but discard the white plastic servo horn. Attach the Standard Servo label to the servo. Assemble one of the standard servos as indicated in the illustrations. The other standard servo will be used for the gripper assembly.

Standard servo motors are used for proportional rotation and for grippers, steering, and positioning.



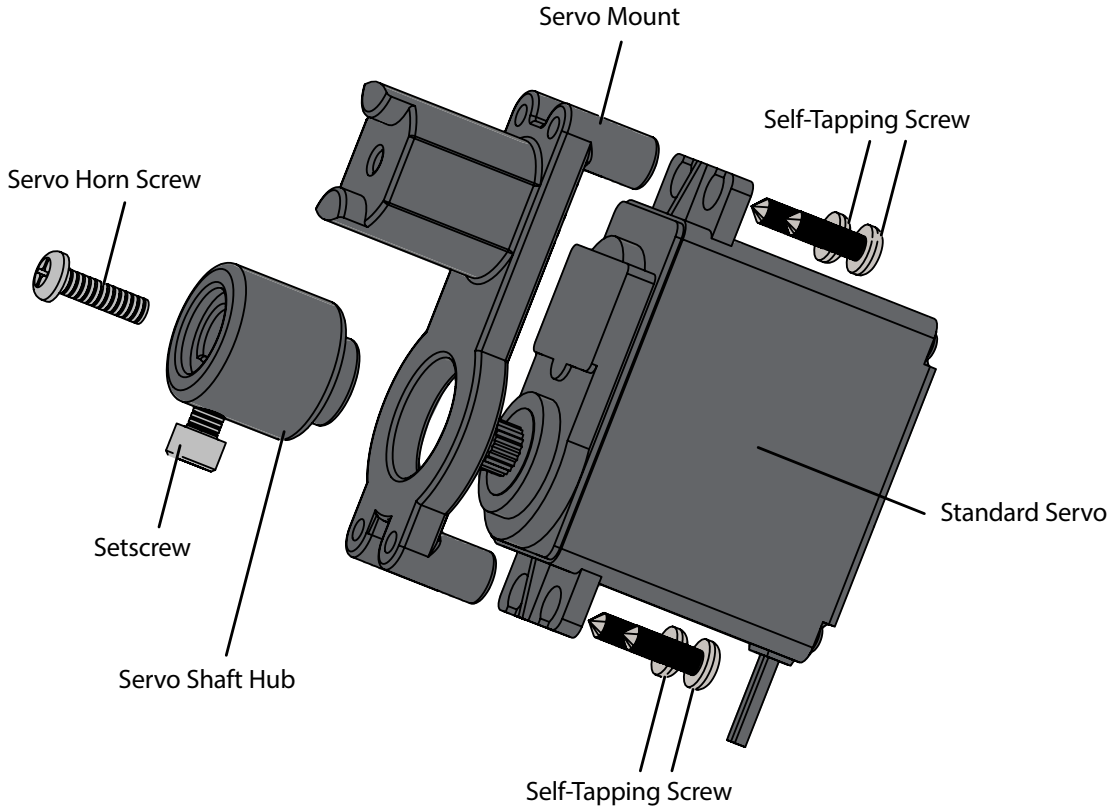
1x
Standard Servo 40538



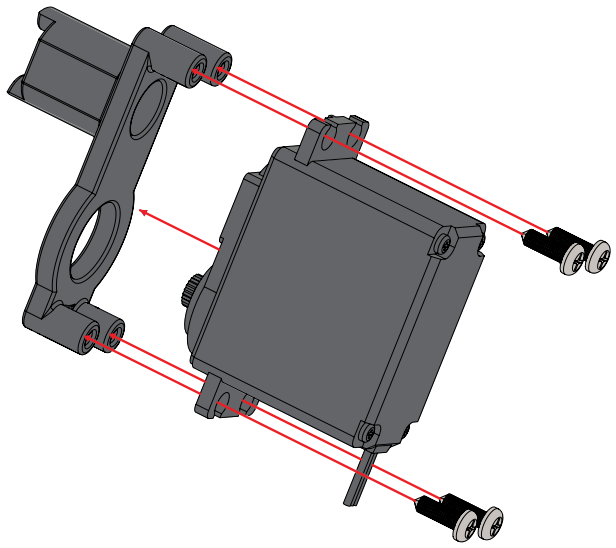
1x
Servo Mount 40232

1x
Servo Shaft Hub 40230

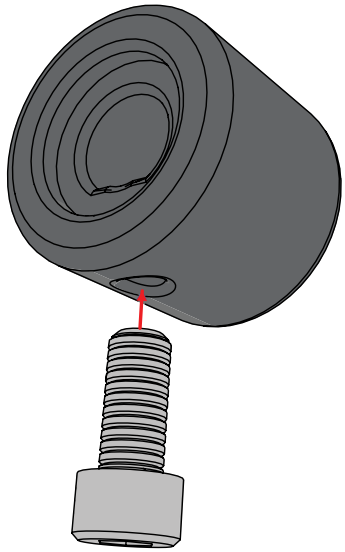
1x
Setscrew 40516



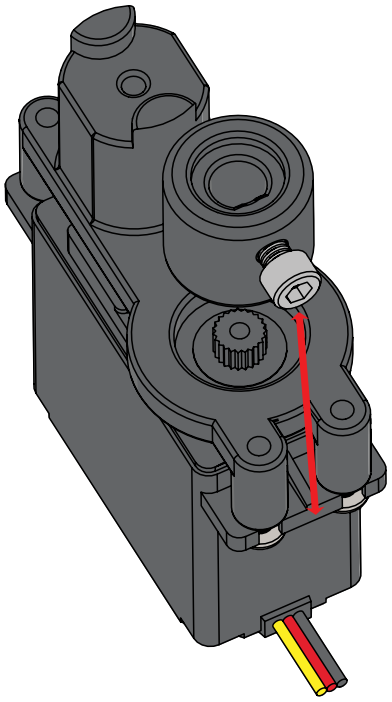
Step 1



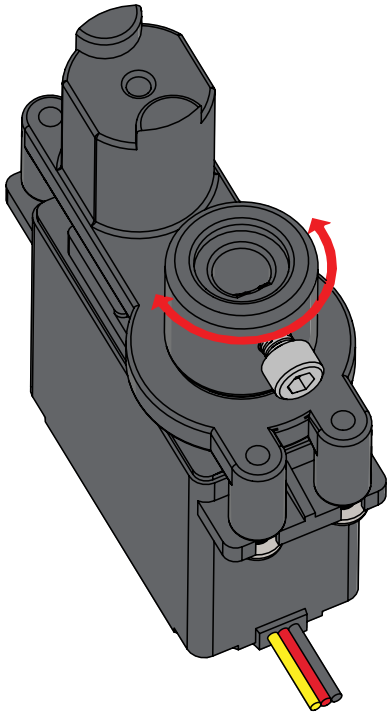
Step 2



Step 3

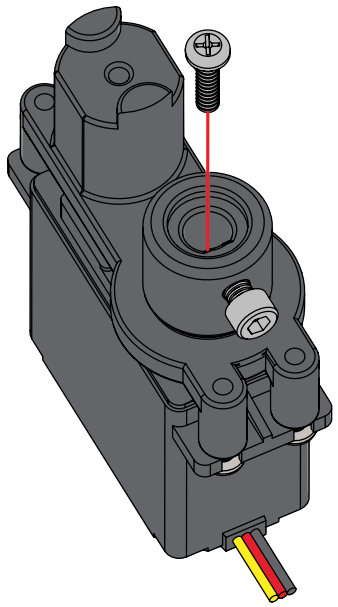


Step 4

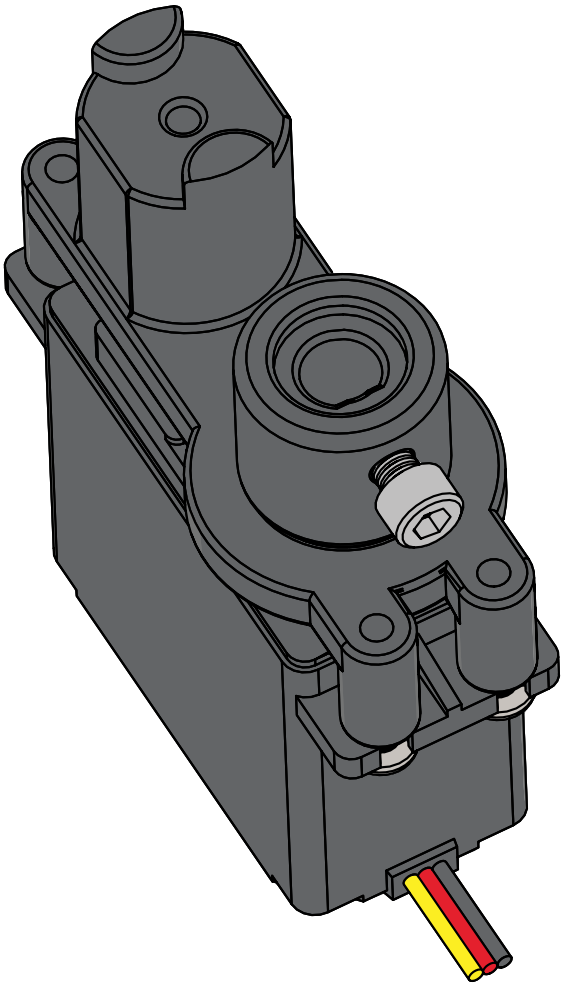


TIP: Ensure that when the servo hub is positioned as pictured, it can rotate both left and right.

Step 5

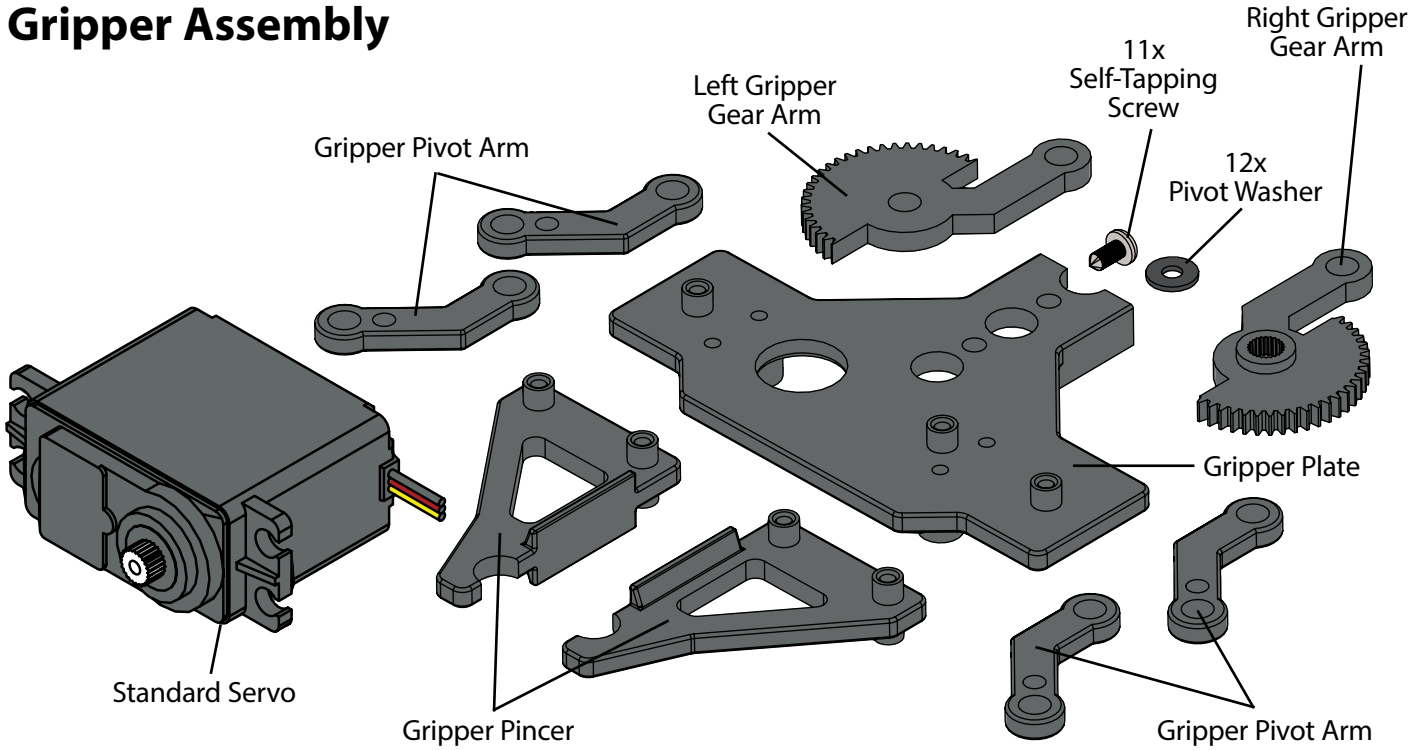


Finished Assembly

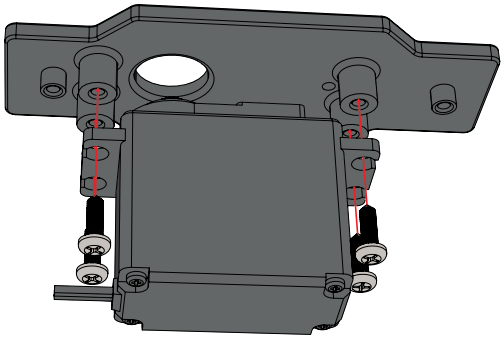


Back to Contents page

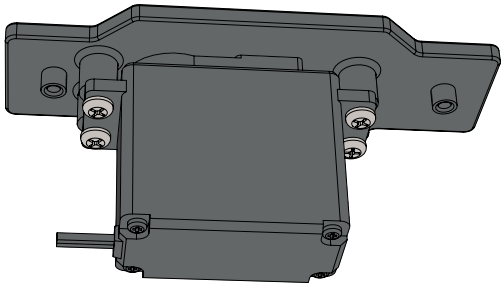
Gripper Assembly



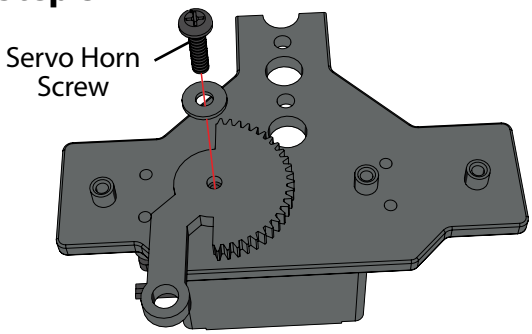
Step 1



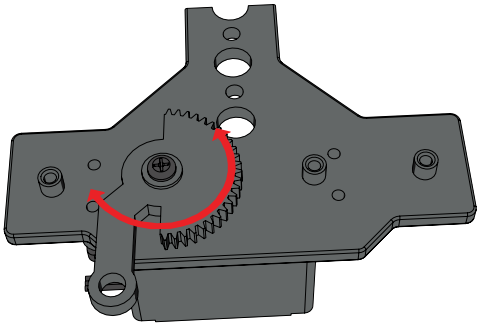
Step 2



Step 3

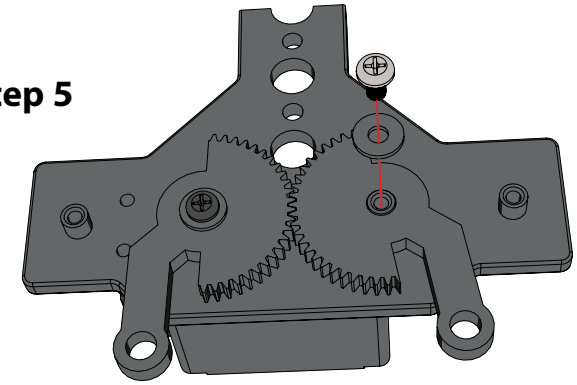


Step 4

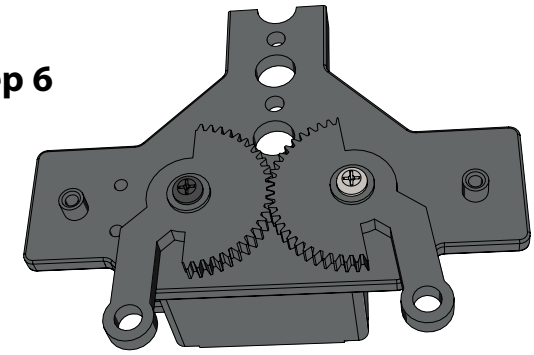


TIP: Before attaching the Left Gripper Gear Arm, you must make sure the servo is in the neutral position. Please make sure the when the gripper gear arm is positioned as pictured, it can rotate both left and right.

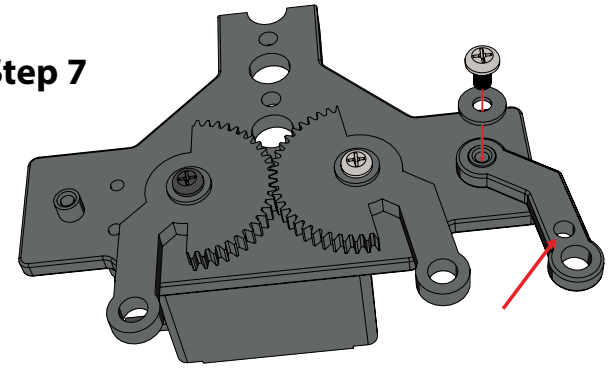
Step 5



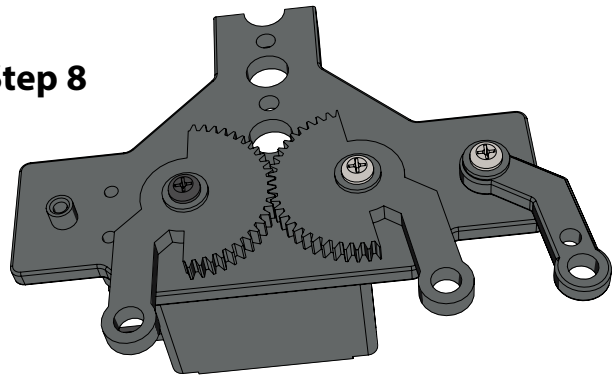
Step 6



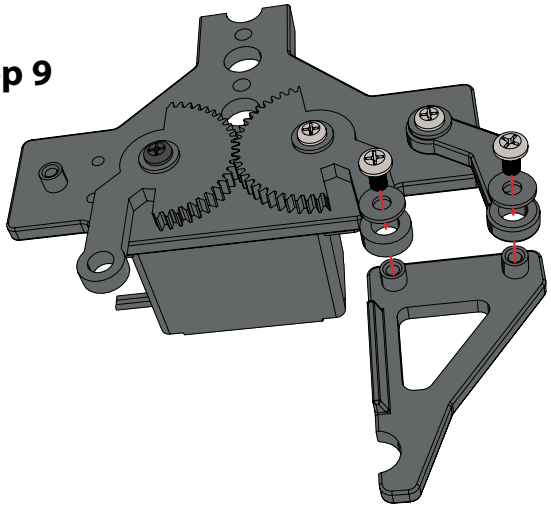
Step 7



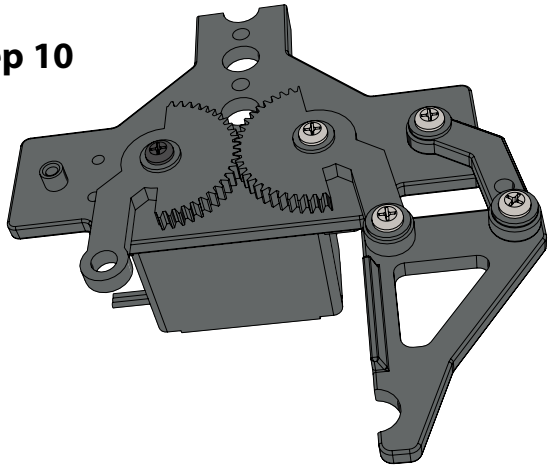
Step 8



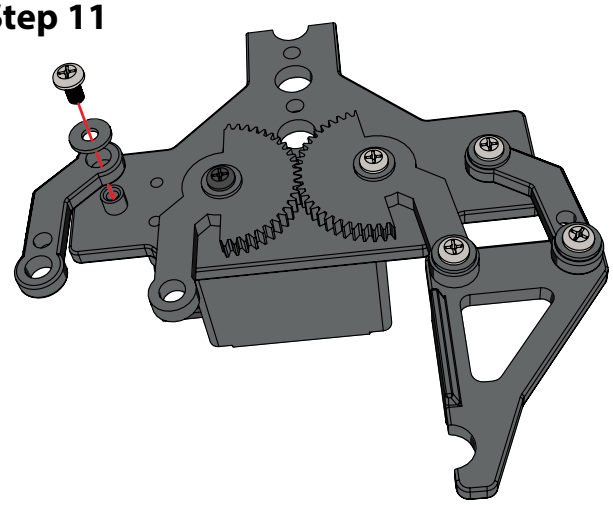
Step 9



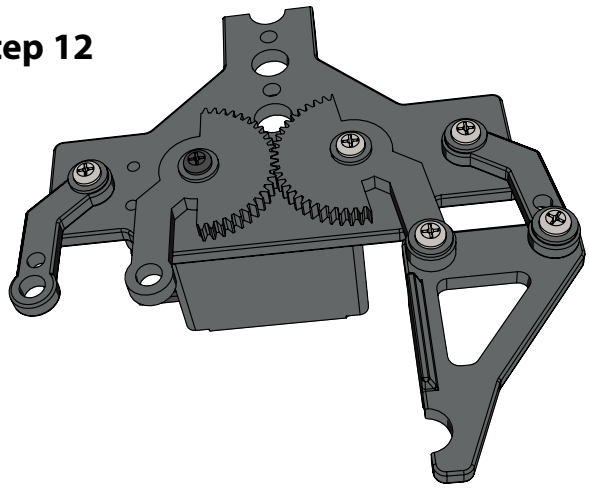
Step 10



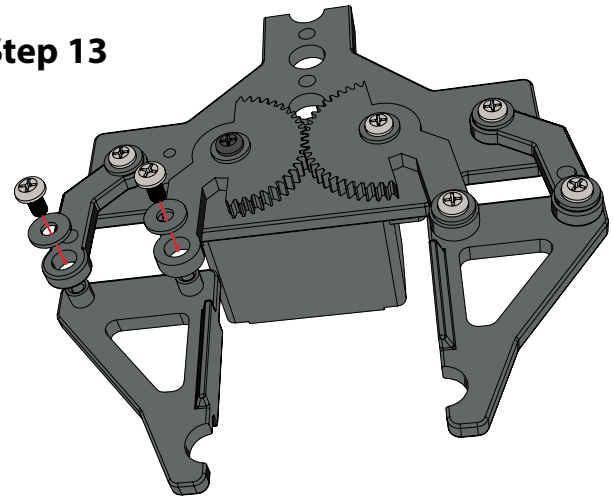
Step 11



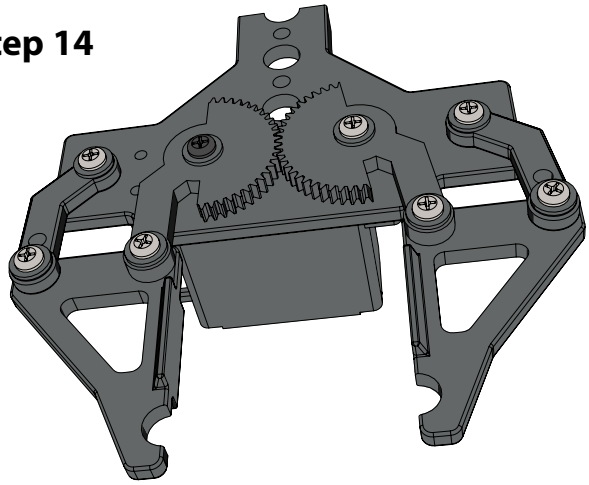
Step 12



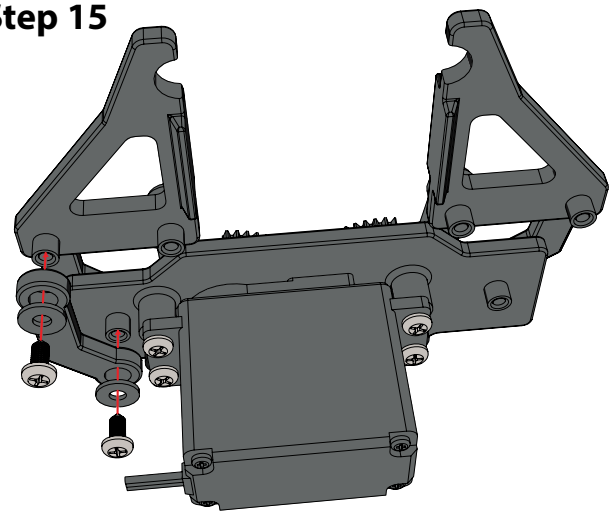
Step 13



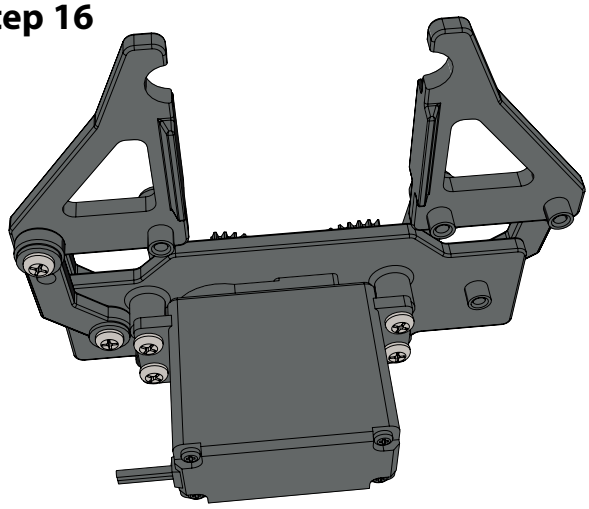
Step 14



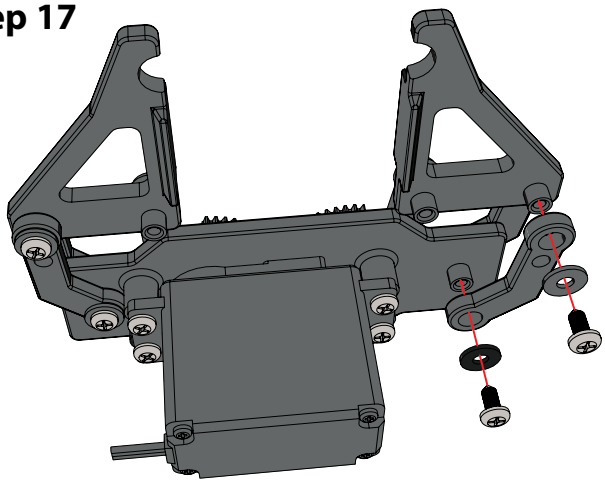
Step 15



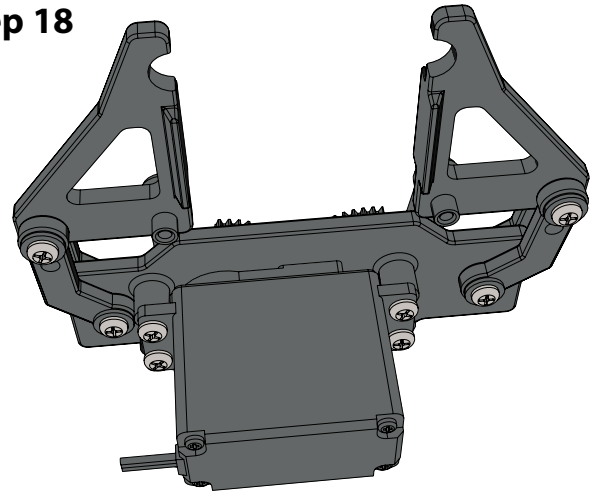
Step 16



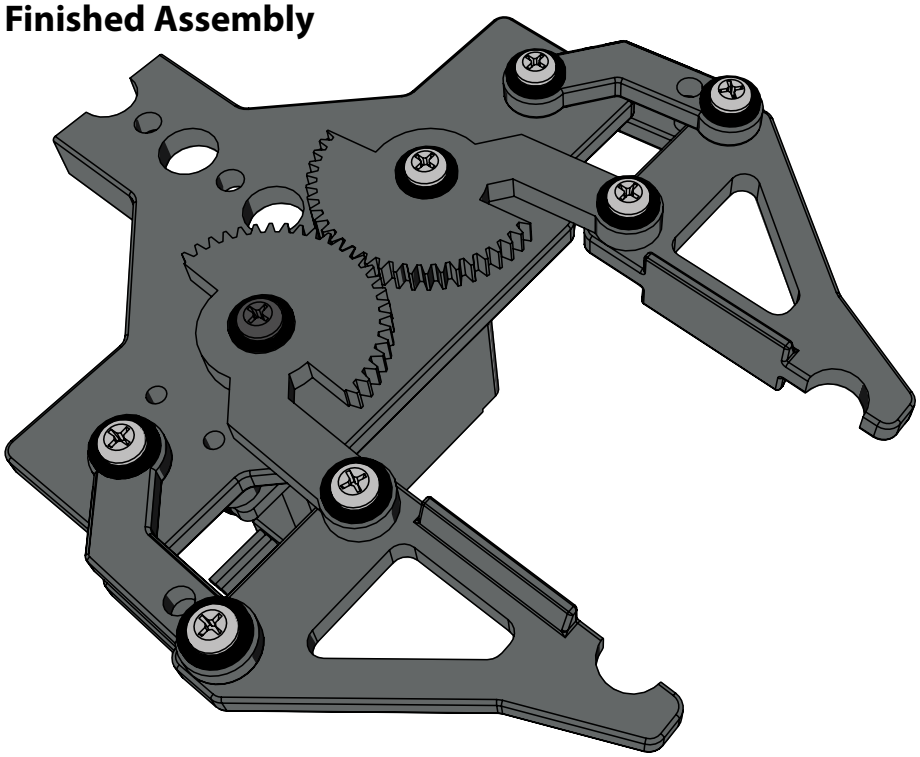
Step 17



Step 18



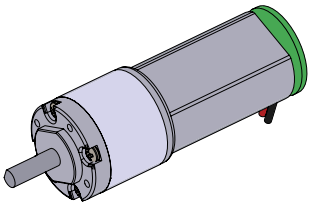
Finished Assembly



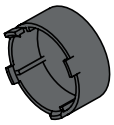
Back to Contents page

DC Motor Mount Assembly

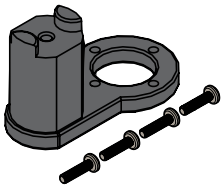
Parts Needed



1x
Digilent DC
Gear Motor
41143



1x
DC Motor
Encoder Cap
41231

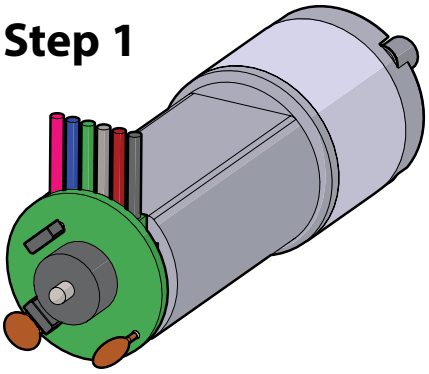


1x
Digilent DC Motor
Mount 41138

1x
Digilent
Motor Shaft
Adapter 41139

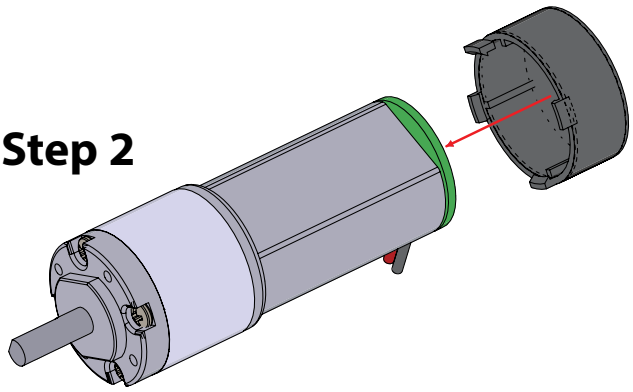
2x
Setscrew 40516

Step 1

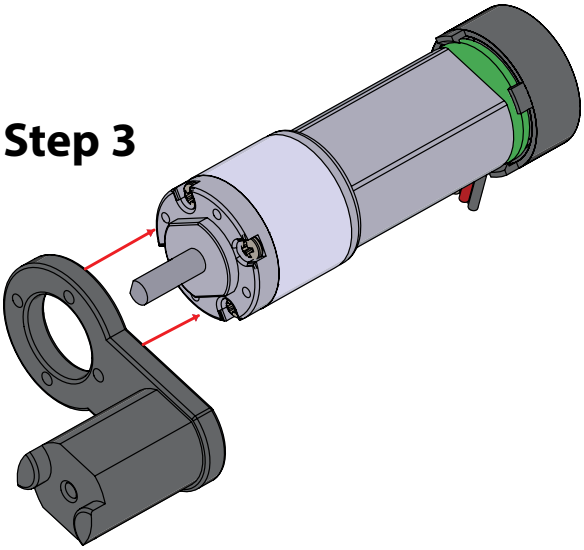


After unpacking the Digilent DC Gear Motor, inspect the electronic components on the encoder board before snapping the encoder cap in place. If components appear to be bent, carefully bend them back up straight to ensure that they are not touching the spinning center magnet.

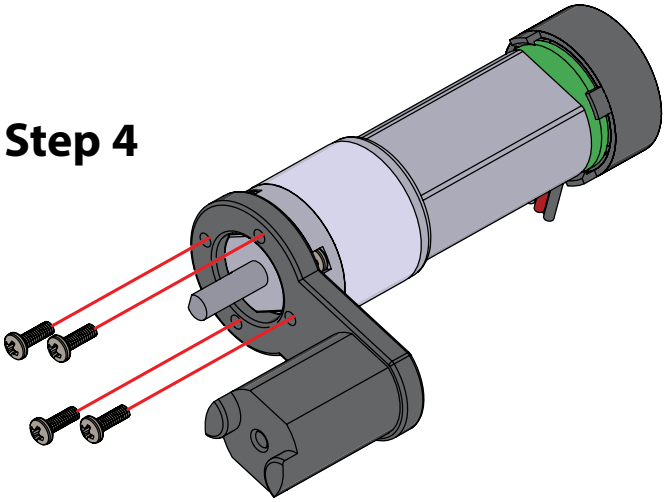
Step 2



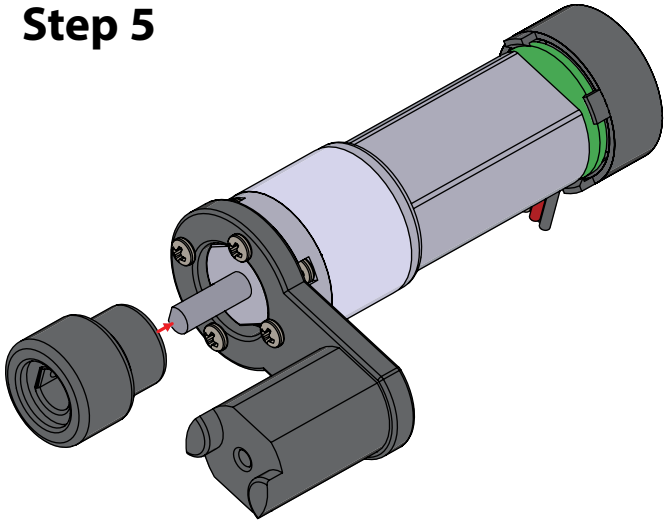
Step 3



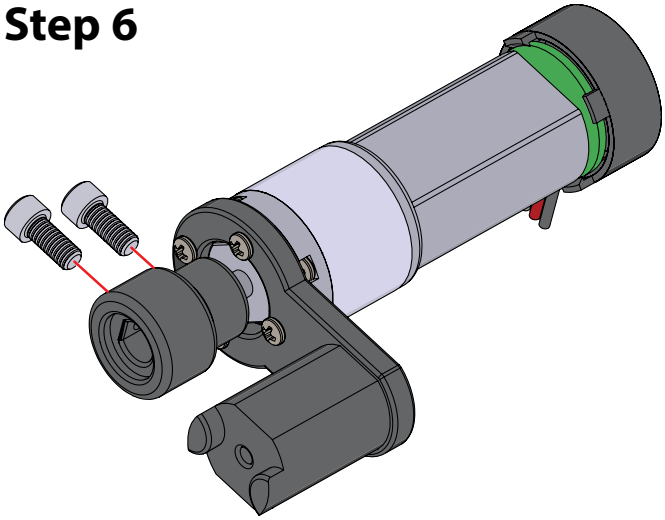
Step 4



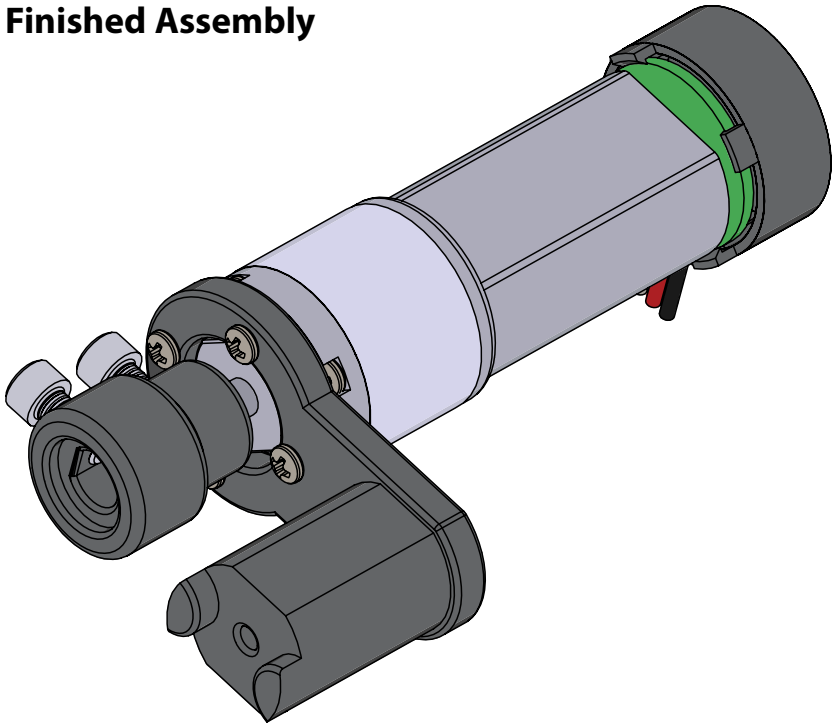
Step 5



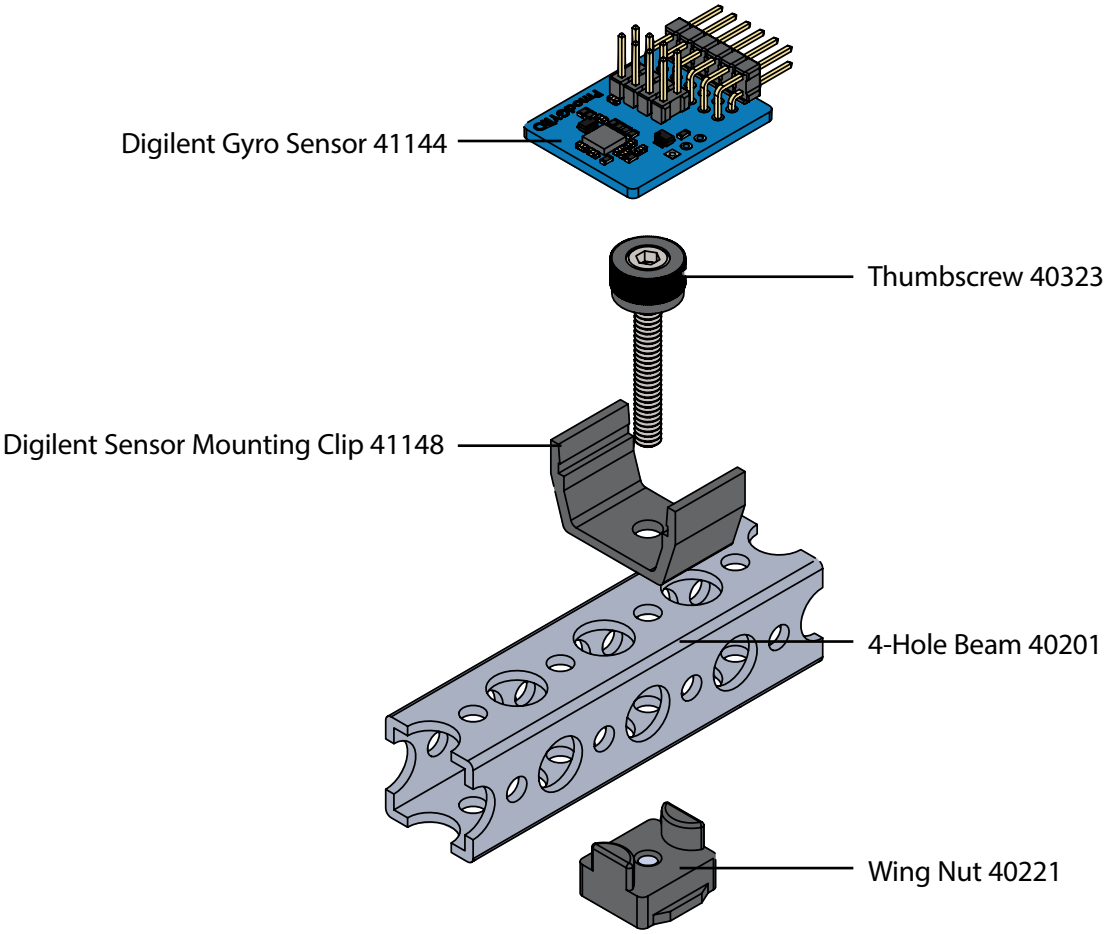
Step 6



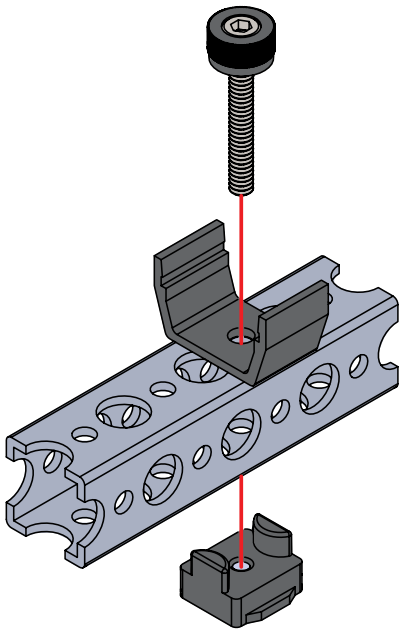
Finished Assembly



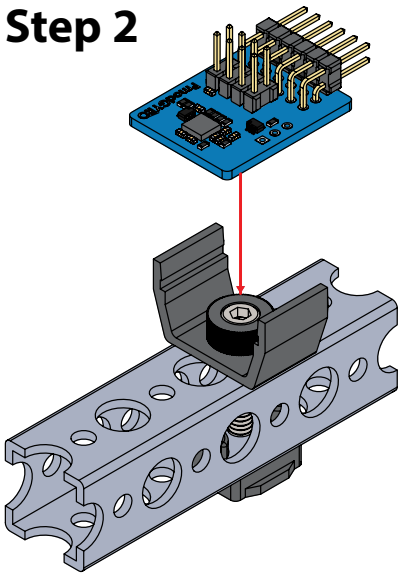
Gyroscope Sensor Mount Assembly



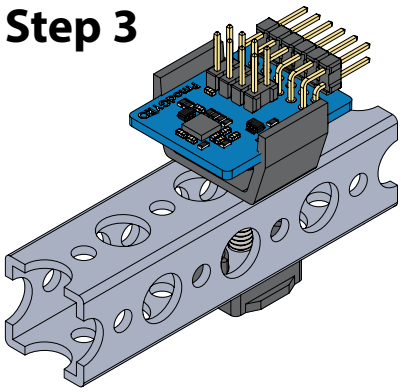
Step 1



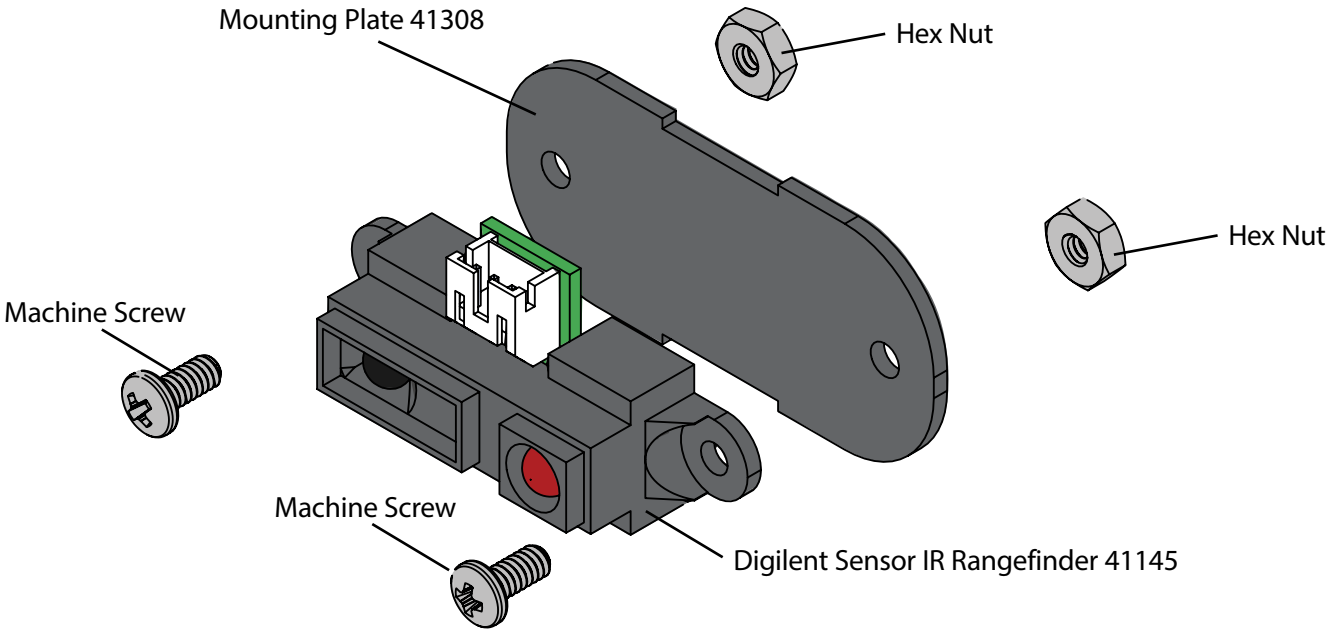
Step 2

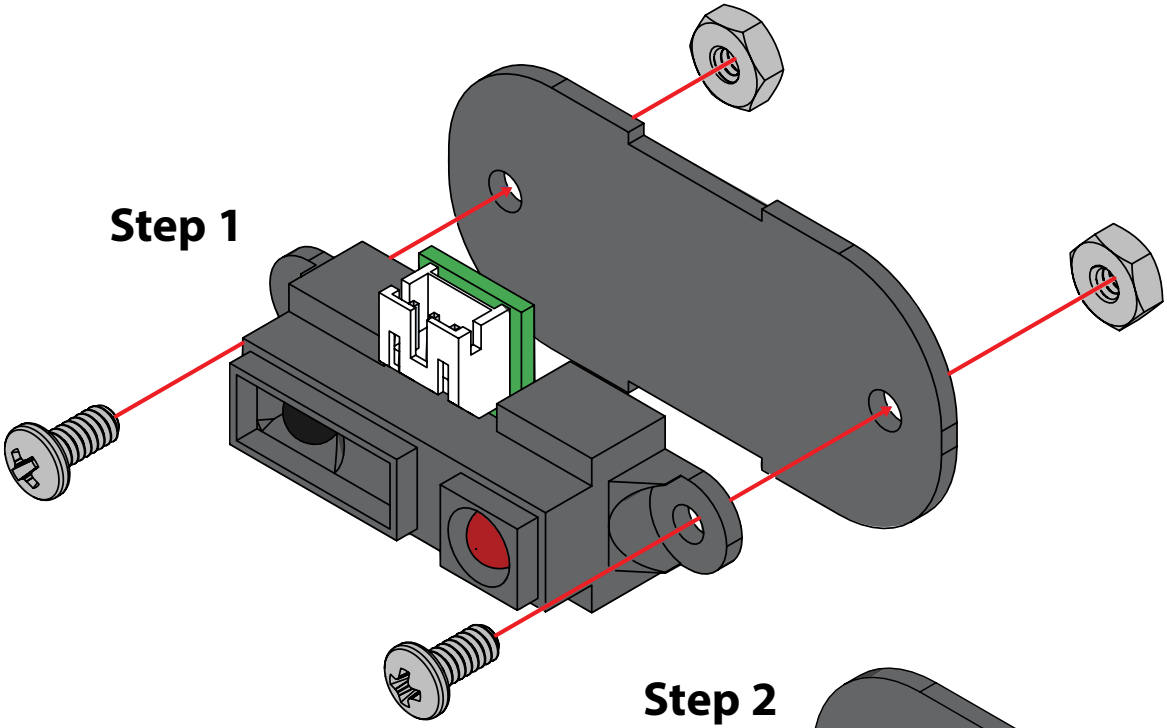


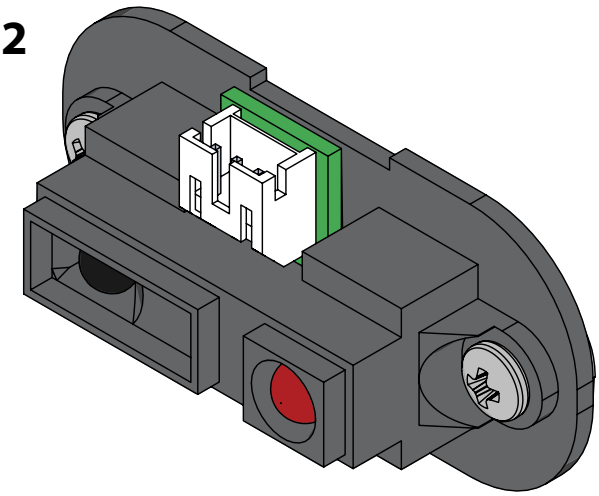
Step 3



IR Rangefinder Sensor Mount Assembly



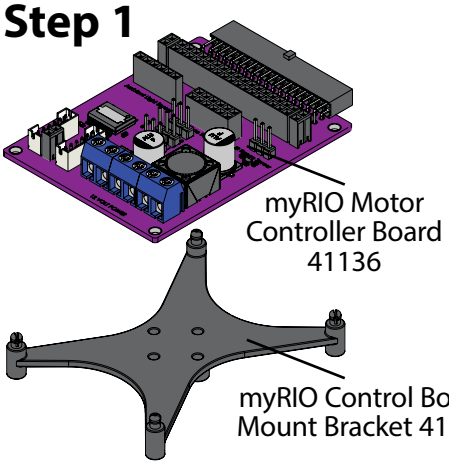




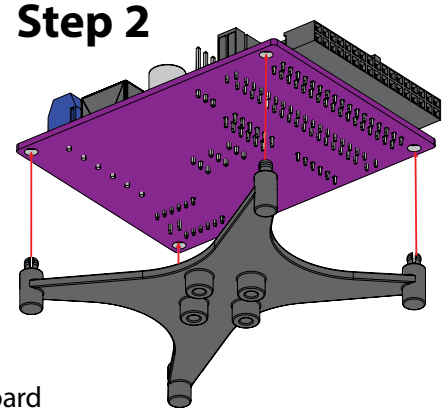
Motor Controller Board Mount Assembly

Attach myRIO Motor Controller Board to myRIO Control Board Mount Bracket

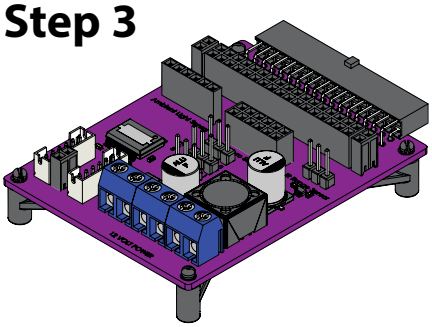
Step 1



Step 2

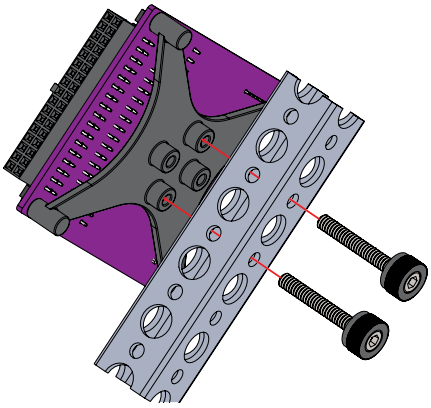


Step 3

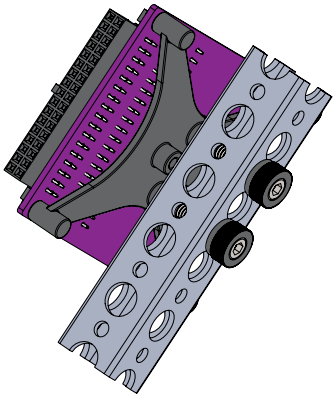


Attach myRIO Motor Controller Board to Beam

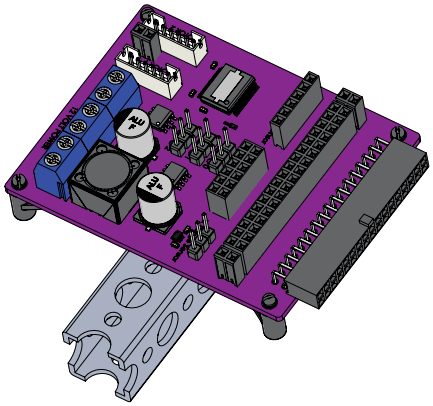
Step 1



Step 2

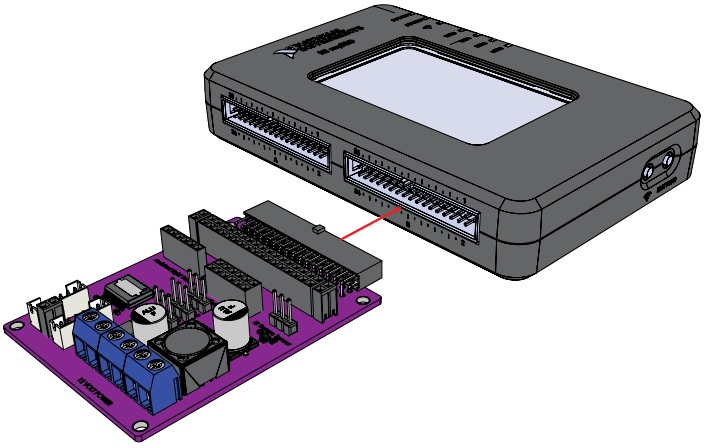


Step 3

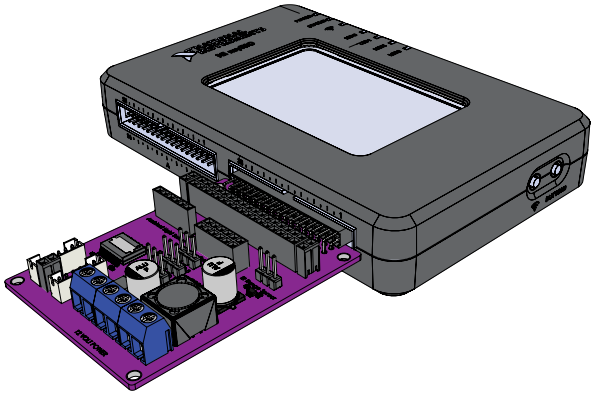


Attach myRIO Motor Controller Board to myRIO

Step 1



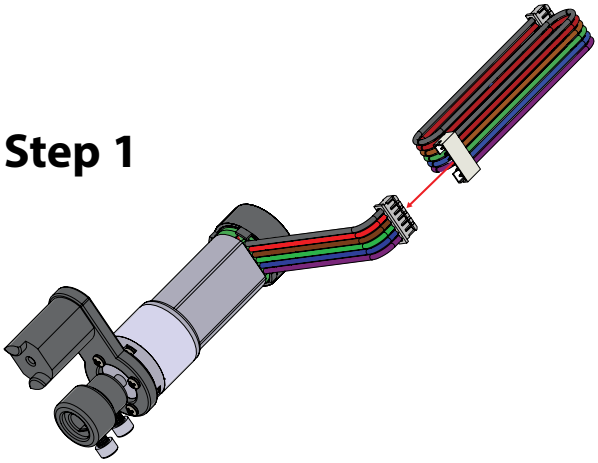
Step 2



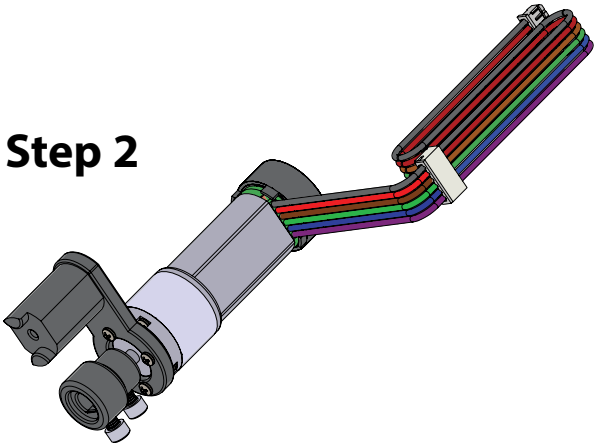
Cord Extension Assemblies for Electronics

Attach Digilent DC Motor and Extension Cable

Step 1

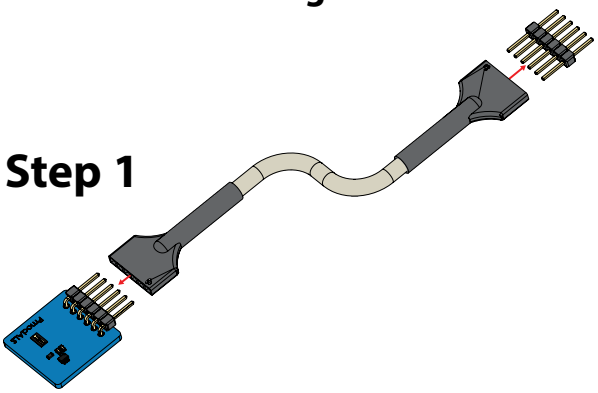


Step 2

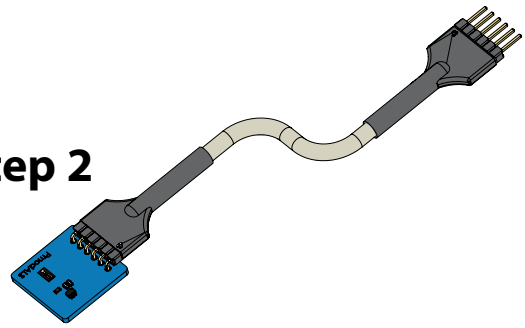


Attach Ambient Light Sensor and Extension Cable

Step 1

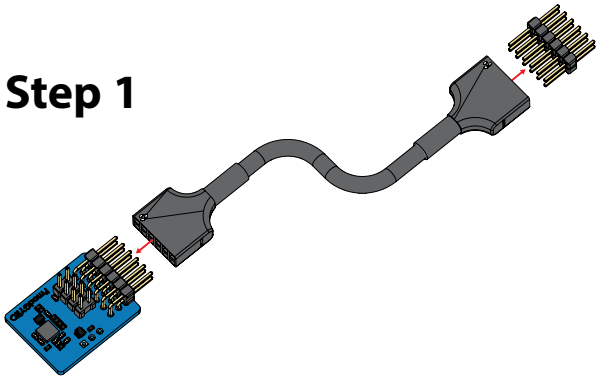


Step 2

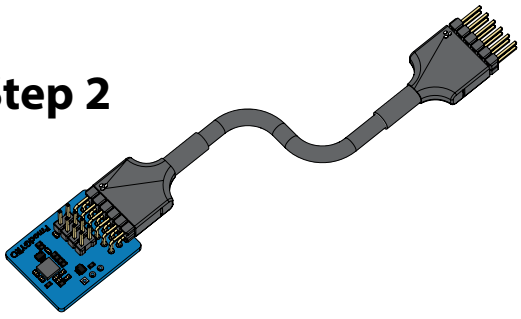


Attach Digilent Gyro Sensor and Extension Cable

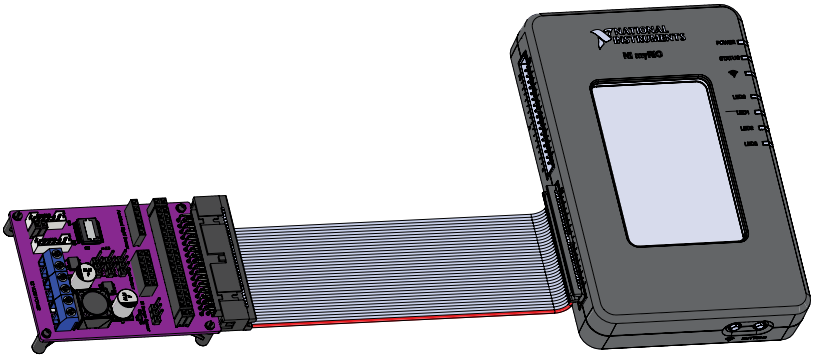
Step 1



Step 2



Attach motor controller board to myRIO via the MXP extension cable

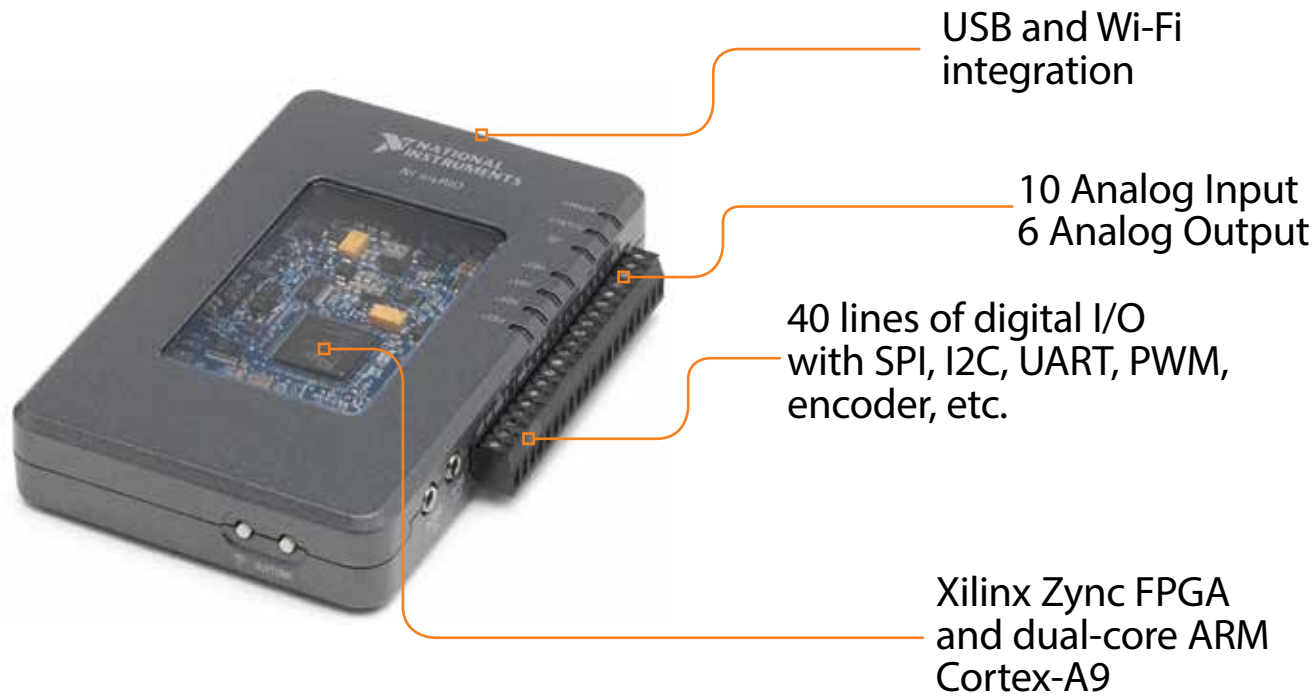


Back to Contents page

Introduction to myRIO and LabVIEW™ Graphical Programming

myRIO Overview

NI myRIO is an embedded hardware device designed for students and based on technology used in industry for controls and robotics. It has all the I/O and processing power you need to control the systems you create using this kit and more!



Using myRIO you can . . .

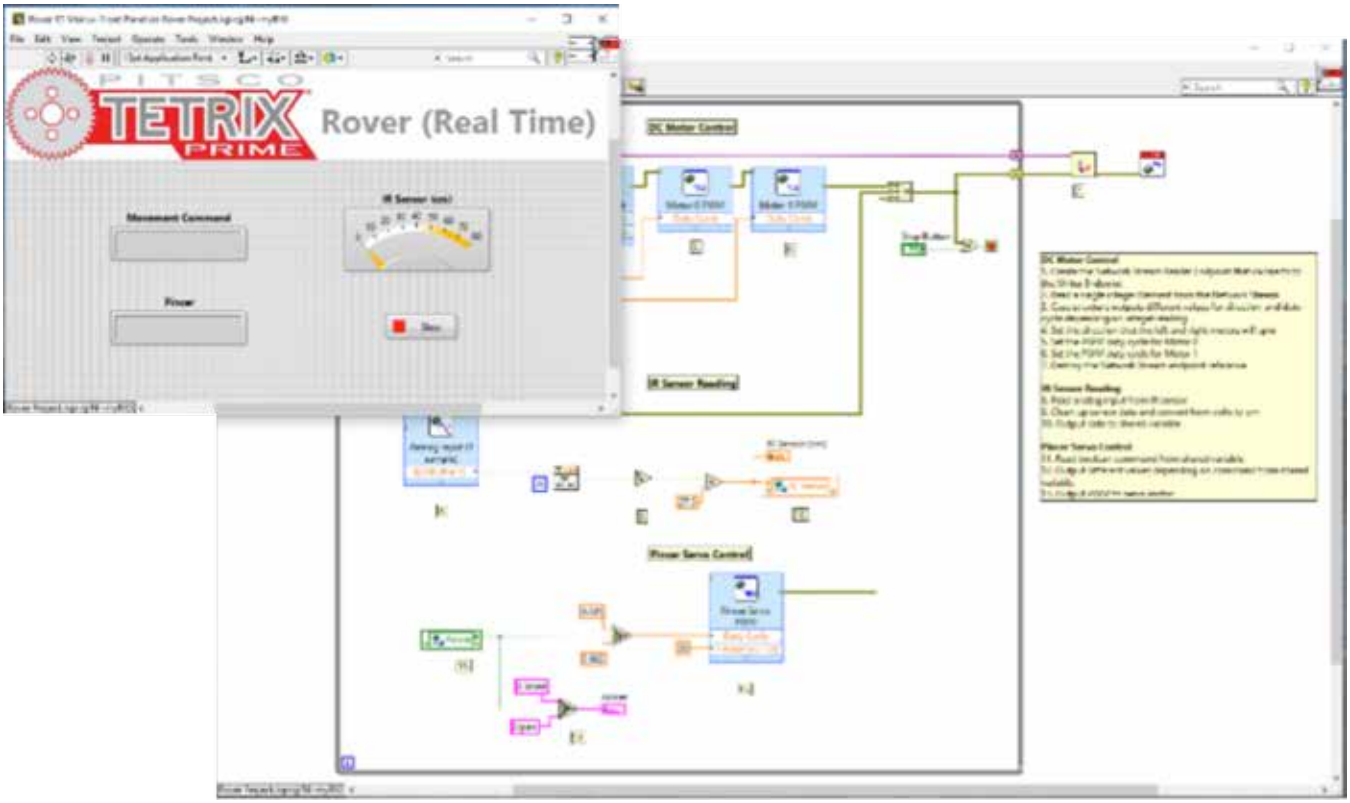
- Implement simple to complex controls algorithms such as PWM and PID
- Communicate over Wi-Fi to control and monitor your robot remotely
- Deploy programs to allow your robot to run autonomously
- Use the motor board to interface with all the sensors and actuators in this kit

. . . and much more! To learn more about myRIO, visit ni.com/myrio.

LabVIEW™ Overview

LabVIEW is a graphical programming language used all over the world in industry and academia. In education, it's considered a tool of discovery. LabVIEW gives students complete visibility into engineering systems, enabling them to learn and design systems of their own faster. For the same reason, it's considered a tool to accelerate the productivity of engineers and scientists. LabVIEW is a standard for test, instrumentation, and control applications in just about every industry. Though you may not realize it, if you can turn it on, drive it, fly it, changes are, NI and LabVIEW made it happen.

LabVIEW uses a dataflow model instead of sequential lines of text. This empowers coders to write functional code using a visual layout that resembles your thought process. When you code, you use function blocks and connect them using wire that transfer data from function to function. As you code, you can create a user interface on the Front Panel. This allows you to always have visibility of your data as you design your system, and also to allow end users to interact with the program. To learn more about LabVIEW and how it's used in industry, visit ni.com/LabVIEW.



Back to Contents page

myRIO and LabVIEW Learning Resources and Manuals

In this kit, you will use LabVIEW to program myRIO to run controls algorithms, interact with sensors and actuators, communicate with other devices, and implement other innovative functionality you'd like your robot to have. Though this guide has instructions for setting up LabVIEW on your computer and running the code that accompanies the base TETRIX PRIME assemblies, we recommend that you use the resources below to learn more about how to use LabVIEW and myRIO.

LabVIEW Learning Resources

- Learn LabVIEW Video Series: http://www.ni.com/academic/students/learn-labview/
- *Learning with LabVIEW*, by R. H. Bishop: https://decibel.ni.com/content/docs/DOC-39874
- Self-paced online training: http://sine.ni.com/tacs/app/fp/p/ap/ov/lang/en/pg/1/sn/n5:selfpacedonline/

myRIO Learning Resources

- Learn myRIO Video Series: http://www.ni.com/academic/students/learn-rio/applications/
- NI myRIO Project Essentials Guide: http://www.ni.com/tutorial/14621/en/
- NI myRIO Vision Essentials Guide: http://www.ni.com/white-paper/52475/en/

myRIO Manuals

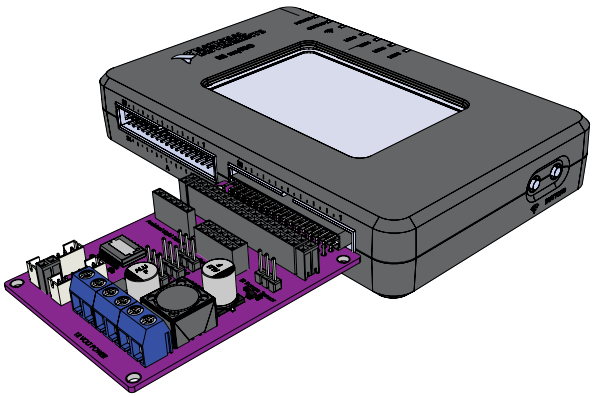
- NI myRIO-1900 User Guide and Specifications: http://www.ni.com/pdf/manuals/376047a.pdf
- myRIO Shipping Personality Reference: http://www.ni.com/product-documentation/14655/en/

Electronics Overview and Specifications

This kit provides sensors, motors, and a motor controller board for interfacing with myRIO and creating mechatronic systems. This section provides overview and specification information for each electronic component. For the motor controller board, it provides an overview and diagram all the ports contained on the board. For each sensor and motor, it provides a brief overview, links to specifications and resources, and specifications of the particular motor controller port.

To set up and test each sensor and motor, see the following section. To learn more about the provided sensor and motors, as well as other sensors, actuators, and interfaces you can use with myRIO, check out the myRIO Project Essentials Guide: http://www.ni.com/tutorial/14621/en/ .

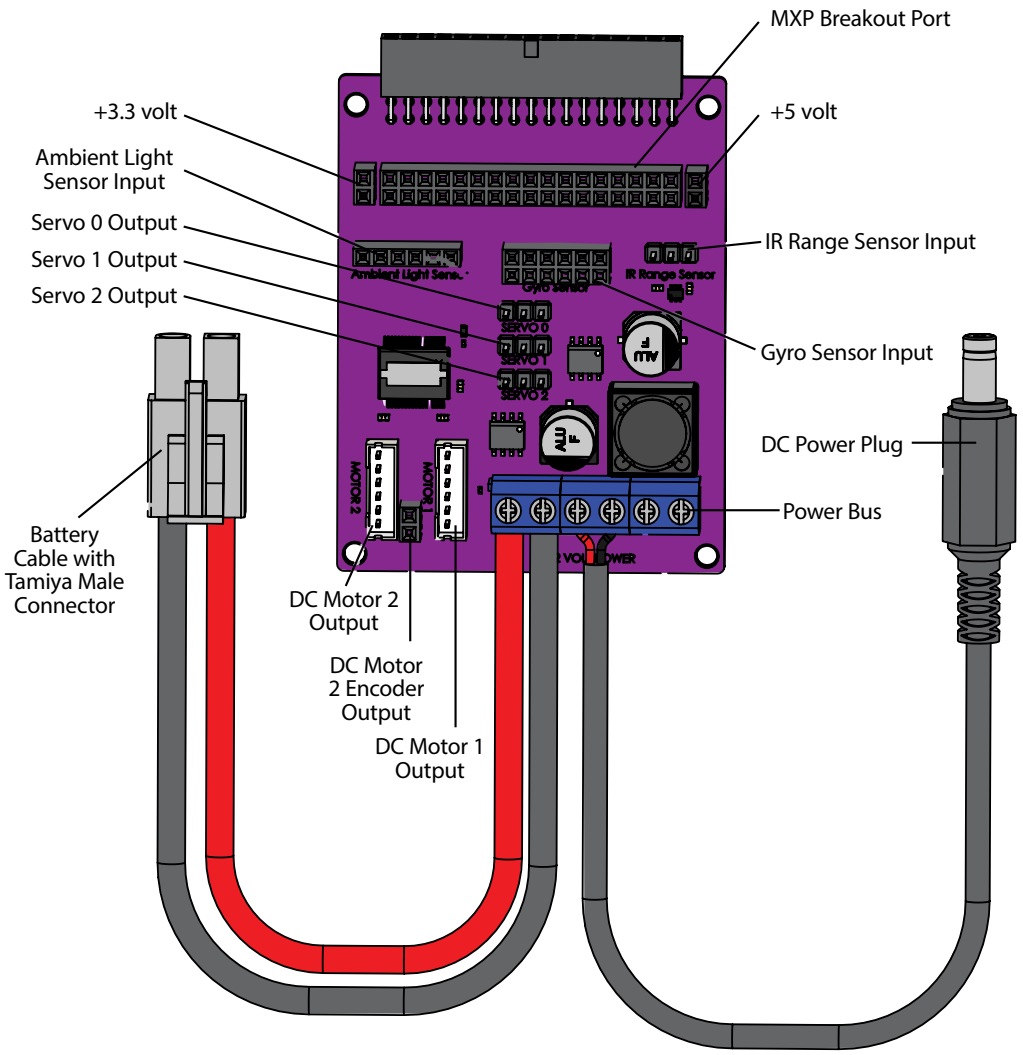
myRIO Motor Controller Board

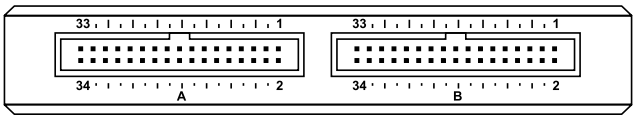


The purpose of the motor controller board is to provide extra power and signal conditioning to motor output signals, while also providing protection to myRIO. It also breaks out all the pins needed for the sensors in the kit so that they are easier to connect to. It connects directly to the 34 pin myRIO Extension Port (MXP). myRIO has two identical MXP ports, A and B. The motor controller board can be connected into either port. The signals are distinguished in software by the connecto r name, as in ConnectorA/DIO1 or ConnectorB/DIO1. Refer to the software documentation for information about configuring and using signals

The pin outs on the myRIO motor controller board are designed to be used with the sensors and actuators used in the kit, but can also be repurposed for other sensors and actuators depending on their specifications and connectors. See the figure below for port locations, and the next few pages for detailed port descriptions.

myRIO Robotics Motor Board Assembly





MXP Connector and Breakout Ports

The MXP connector port of the motor controller board connects directly to the myRIO MXP port. The MXP breakout port is a direct breakout of the myRIO MXP port. The following figure and table shows the signals on both MXP connectors A and B. Note that some pins carry secondary function as well as primary functions.

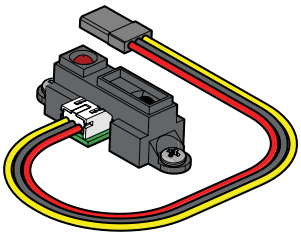
+3.3V	DIO10/PWM2	DIO9/PWM1	DIO8/PWM0	DIO7/SPI.MOSI	DIO6/SPI.MISO	DIO5/SPI.CLK	DIO4	DIO3	DIO2	DIO1	DIO0	A13	A12	A11	A10	+5V
33	31	29	27	25	23	21	19	17	15	13	11	9	7	5	3	1
34	32	30	28	26	24	22	20	18	16	14	12	10	8	6	4	2
DIO15/12C.SDA	DIO14/12C.SCL	DGND	DGND	DIO13	DGND	DIO12/ENC.B	DGND	DIO11/ENC.A	DGND	UART.TX	DGND	UART.RX	DGND	AGND	AO1	AO0

Power Port

The power port consists of three terminals, each including a +12V pin and a ground pin. Use one of the terminals to supply power to the motor board and myRIO by plugging in the supplied battery pack via the battery cable. Use another terminal to supply power to myRIO via the DC power plug.

Learn more about the supplied rechargeable battery pack: http://www.pitsco.com/TETRIX_MAX_12_V_Battery_Pack?SKU=W41135&tp=1

Infrared (IR) Range Sensor



1x
Digilent Sensor
IR Rangefinder 41145

The infrared range sensor outputs an infrared signal pulse of known frequency and detects the reflection of the signal off an object. The elapsed time from the signal output to detecting the reflected signal can be used to distance from the signal source to the object. The elapsed time can also be used to distinguish differences among materials or colors based on IR reflectivity.

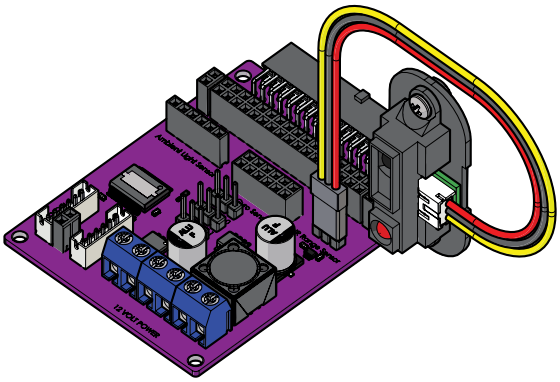
The range finder pinout includes power, ground, and an analog signal, which supplies a voltage proportional to the measured distance. Signal conditioning can be done in LabVIEW code to convert the analog signal to units of length.

Learn more:

- Specifications: http://store.digilentinc.com/ir-range-sensor/
- Project Essentials Guide, chapter 19: http://www.ni.com/tutorial/14621/en/

IR Rangefinder Input Port

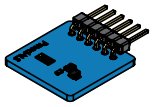
The IR range sensor input port of the motor controller board is a direct breakout of myRIO analog input pins.



Pin (from left to right based on the image above)	Wire Color (if using provided IR sensor)	MXP Pin Number	Name in Software (based on MXP A)
Signal	Yellow	3	A/AI0
Ground	Black	6	n/a
+5 VCC	Red	1	n/a

Back to Contents page

Ambient Light Sensor



1x
Digilent
Ambient Light
Sensor 41146

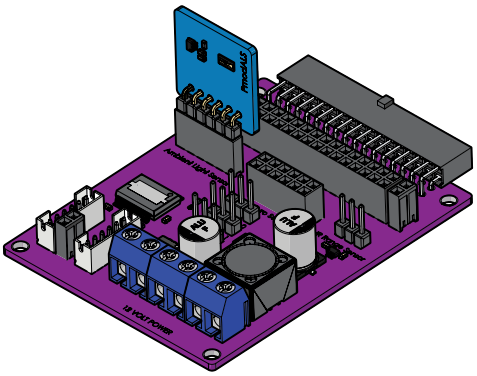
The ambient light sensor uses a phototransistor and amplifier circuit to detect intensity of light. The chip itself is considered a Peripheral Module interface, or PMOD, which is an open standard defined by Digilent Inc for pre-built circuits with a standard 6-pin interface. The sensor communicates light intensity data using a common communication protocol called the Serial Peripheral Interface (SPI). Per the SPI protocol, the pinout includes a chip select, data line, clock line power, and ground.

Learn more:

- Specifications: http://store.digilentinc.com/pmodals-ambient-light-sensor/
- Project Essentials Guide, chapter 24: http://www.ni.com/tutorial/14621/en/

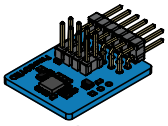
Ambient Light Sensor Input Port

The ambient light sensor input port of the motor controller board is a direct breakout of the myRIO SPI pins.



Pin (from left to right based on the image above)	MXP Pin Number	Name in Software (based on MXP A)
CS	11	A/DIO4
NC	Not connected	Not connected
SDA	23	A/SPI.MISO
SCL	21	A/SPI.CLK
GND	30	n/a
+3.3 VCC	33	n/a

Gyroscope Sensor



1x
Digilent Gyro
Sensor
41144

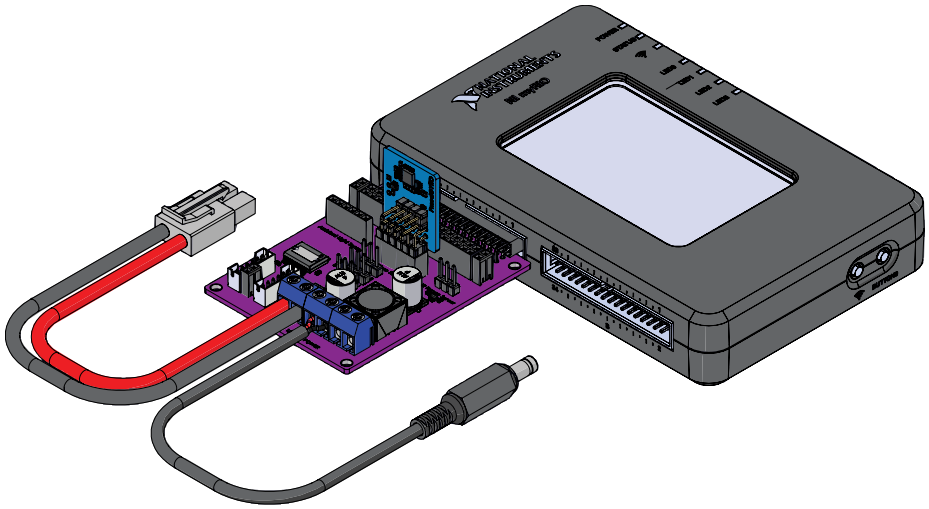
The gyroscope (gyro) sensor measures the angular velocity, or rate of rotation, about three axes. When mounted, it provides 3-D attitude (pitch, roll, and yaw). Similar to the ambient light sensor, the gyro is a PMOD. It can communicate angular velocity using either the SPI or I2C protocol, but in this kit we only use the I2C protocol. Per the I2C protocol, the gyro pinout includes a serial data (SDA), serial clock (SCL), two interrupts, power, and ground.

Learn More

- Specifications: http://store.digilentinc.com/pmodgyro-3-axis-digital-gyroscope/
- Project Essentials Guide, chapter 22: http://www.ni.com/tutorial/14621/en/

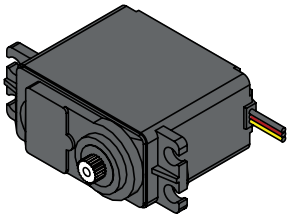
Gyroscope Sensor Input Port

The gyroscope sensor input port of the motor controller board is a direct breakout of the myRIO I2C and digital input pins.



Board Pin Number (Based on pin-out diagram above)	Pin Name	MXP Pin Number	Name in Software (based on MXP A)
1	NC	Not connected	Not connected
2	Serial data (SDA)	34	A/I2C.SDA
3	NC	Not connected	Not connected
4	Serial clock (SCL)	32	A/I2C.SCL
5	Ground	30	n/a
6	NC	Not connected	Not connected
7	Interrupt 2	13	A/DIO1
8	Interrupt 1	11	A/DIO0
9	NC	Not connected	Not connected
10	NC	Not connected	Not connected
11	Ground	30	n/a
12	+3.3 VCC	33	n/a

Servo Motor



2x
Standard Servo 40538

The servo motor provides motion control. Pulse-width modulation (PWM) is used to control the position of the motor shaft. Internally, the servo combines a DC motor, gearbox, potentiometer and controller electronics to provide relatively precise angular position. The servo included in the kit is a standard servo, which means it can rotate about 180 degrees. The pinout of the servo motor includes power, ground, and the PWM signal line.

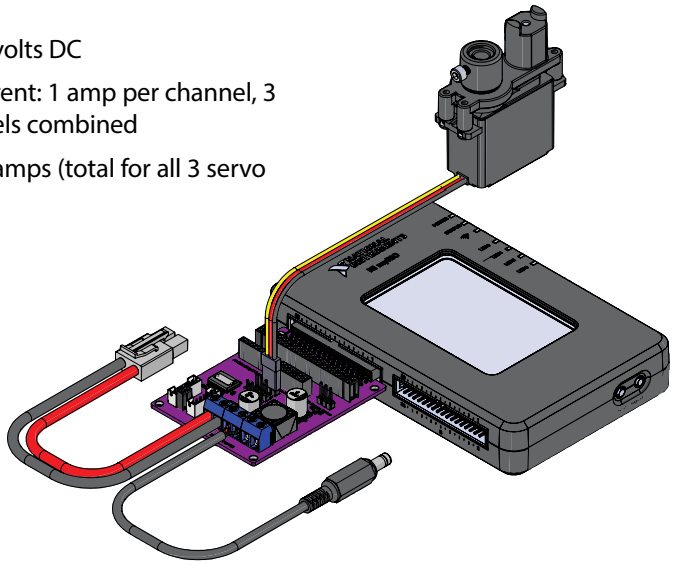
Learn More

- Specifications: https://www.pitsco.com/HS-322HD_Standard_Servo?ap=10118-8049
- Project Essentials Guide, chapter 17: http://www.ni.com/tutorial/14621/en/

Servo Motor Output Port

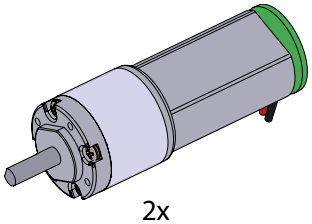
The motor controller board contains circuitry to provide more power and extra protection for the connected servos.

- Servo Channels: 3
- Servo Power Voltage: 6 volts DC
- Continuous Output Current: 1 amp per channel, 3 amps total servo channels combined
- Peak Output Current: 6 amps (total for all 3 servo channels combined)



Servo Number (noted on motor board)	Pin Name (from left to right based on the image above)	Wire color (if using provided servo motor)	MXP Pin number	Name in software (based on MXP A)
0	Signal	Yellow	27	A/PWM0
0	VCC	Red	n/a	n/a
0	Ground	Black	6	n/a
1	Signal	Yellow	29	A/PWM1
1	VCC	Red	n/a	n/a
1	Ground	Black	6	n/a
2	Signal	Yellow	31	A/PWM2
2	VCC	Red	n/a	n/a
2	Ground	Black	6	n/a

DC Motor and Encoder



2x
Digilent DC Gear Motor 41143

The geared DC motor provides speed control. It receives a differential analog voltage signal, which proportionally controls the angular velocity of the motor shaft. To maximize power efficiency and motor torque, we use PWM signals to control the speed of the DC motor, and a digital line to control the direction. The signals are converted to differential analog voltage by the motor controller board, which is described in more detail below.

The DC motor also includes a built-in quadrature encoder. The encoder provides feedback for the motor shaft's speed and position. The encoder pinout includes power, ground, and two digital signal lines.

Learn More

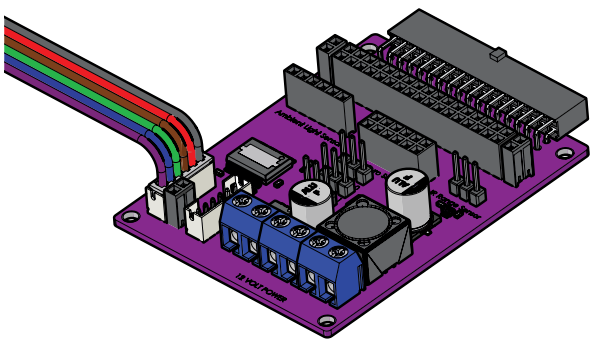
- Specifications: http://store.digilentinc.com/dc-motor-gearbox-1-53-gear-ratio-custom-6v-motor-designed-for-digilent-robot-kits/
- Project Essentials Guide, chapter 18: http://www.ni.com/tutorial/14621/en/

DC Motor and Encoder Output Port

For each DC motor, the motor controller board receives a PWM signal for speed control and a digital signal for direction control. It contains a full bridge motor driver (with protection circuitry) to convert the PWM and digital signals to a differential analog voltage signal, which is passed to the DC motor. Though it can be repurposed for other motors, the port is designed for the DC motor in the kit, which includes a built in encoder.

The myRIO MXP port only contains one set of quadrature encoder input pins. On the motor controller board, these pins are wired to the Motor 1 encoder port. For the Motor2 port, the encoder pins are instead connected to a header port. Jumper wires can be used to connect the pins of the header port to the set of quadrature encoder pins on the opposite MXP port (MXP B if the motor board is connected to MXP A and vice versa).

- Motor Channels: 2
- H-Bridge Chip: TI DRV8432 Dual Channel Full-Bridge PWM Motor Driver
- Motor Voltage Output: 6 volts DC
- Continuous Output Current: 6 amps (total for both motor channels combined)
- Peak Output Current: 9 amps (total for both motor channels combined)
- PWM Operating Frequency: Up to 500 kHz
- Duty Cycle: 0 – 99%
- Integrated Self-Protection Circuits: Under Voltage, Over Temperature, Overload and Short Circuit
- On-board Over Temperature LED Indicator (OT) – also pinned to myRIO DIO13



DC Motor and Encoder (continued)

Motor Number (noted on motor board)	Pin Name (from left to right based on the image above)	Wire color (if using provided DC motor)	MXP Pin number	Name in software (based on MXP A)
1	Encoder B	Purple	22	A/ENC.B
1	Encoder A	Blue	18	A/ENC.A
1	Encoder Ground	Green	n/a	n/a
1	Encoder VCC	Brown	n/a	n/a
1	Motor +	Red	27 – PWM speed control, 15 – DIO direction control	A/PWM0 A/DIO2
	Motor -	Black		
2	Encoder B	Purple	22 (on opposite MXP port)	B/ENC.B
2	Encoder A	Blue	18 (on opposite MXP port)	B/ENC.A
2	Encoder Ground	Green	n/a	n/a
2	Encoder VCC	Brown	n/a	n/a
2	Motor +	Red	29 – PWM speed control, 17 – DIO direction control	A/PWM1 A/DIO3
	Motor -	Black		

Motor Controller Board Resource Conflicts

Since multiple ports of the motor control board use the same MXP pins, some components cannot be used simultaneously on the same motor controller board. If this limits your design, you can use another motor controller in the second MXP port to avoid the conflict. The chart below shows the ports that can be used simultaneously (indicated with a ✓ and the ports that cannot be used simultaneously (indicated with an X).

	IR Sensor	Ambient Light Sensor	Gyroscope	Servo 0	Servo 1	Servo 2	DC Motor 1	DC Motor 2
IR Sensor		✓	✓	✓	✓	✓	✓	✓
Ambient Light Sensor			✓	✓	✓	✓	✓	✓
Gyroscope				✓	✓	✓	✓	✓
Servo 0					✓	✓	X	✓
Servo 1						✓	✓	X
Servo 2							✓	✓
DC Motor 1								✓
DC Motor 2								

Back to Contents page

Control System Set Up and Testing

The control system of this kit consists of myRIO, LabVIEW code, the PC used for code development and teleoperation where applicable, the motor controller board, sensors, and motors.

LabVIEW and myRIO Setup

1. Install the software required to program myRIO. You can find this software on the disks provided with myRIO as well as to http://www.ni.com/product-documentation/14603/en/. Disk 1 includes the minimum software required to use myRIO. Disk 2 includes additional software you can use to expand the functionality of myRIO.

   The minimum software required to use myRIO is:

   1. LabVIEW
   2. LabVIEW Real-Time Module
   3. LabVIEW myRIO Toolkit

   Activate the software using your software serial number. The myRIO software is only compatible with Windows operating systems.

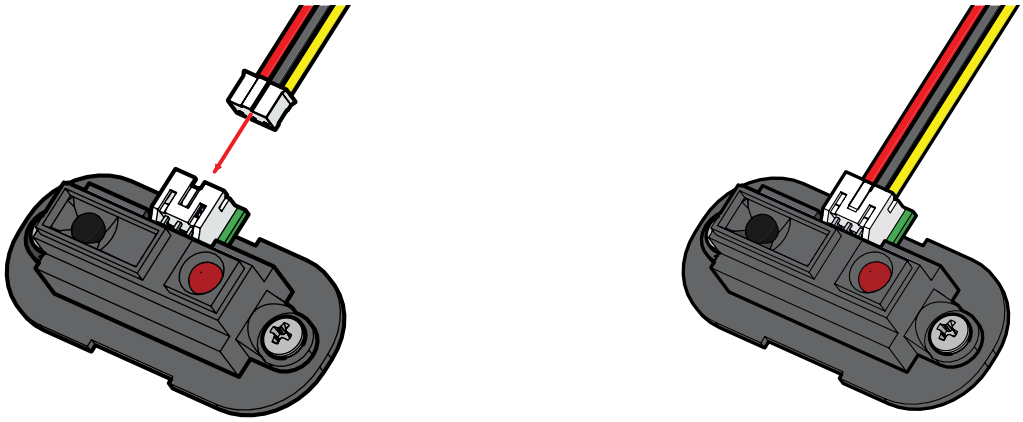
2. After connecting myRIO to power and to your PC using the included USB cable, the myRIO device monitor will appear and will provide some options. For your first time connecting myRIO to your computer, select "Launch the Getting Started Wizard." This will lead you through identifying myRIO, renaming myRIO, installing software onto myRIO, and testing the onboard devices.

3. At the end of the Getting Started Wizard, select "Start my first project now" to be guided through your first experience in LabVIEW using the locally hosted web tutorial.
4. After completing the tutorials in the tab named "Your First NI myRIO Program," move on to the tutorial in the tab "Configuring WiFi on the NI myRIO." Scroll down to the section named "Creating a Wireless Network." Follow the instructions in this section to use myRIO as an access point to create wireless networks. Take note of the SSID you chose and the IPv4 address myRIO was assigned. You will need this information to wirelessly connect to myRIO from your computer.
5. For more information about getting started with the myRIO, visit ni.com/myrio/setup/getting-started.
6. Download code to test your system and control the mechatronic systems described in this guide from ni.com/tetrix-prime. This will automatically open the VI Package Manager (VIPM), which will guide you through an installation process. Though VIPM should have installed in step 1 by default, if it is not installed, you can visit http://vipm.jki.net/ , click Get "VIPM," and download the free version
7. Find the code in a folder on your desktop in a folder named "National Instruments."

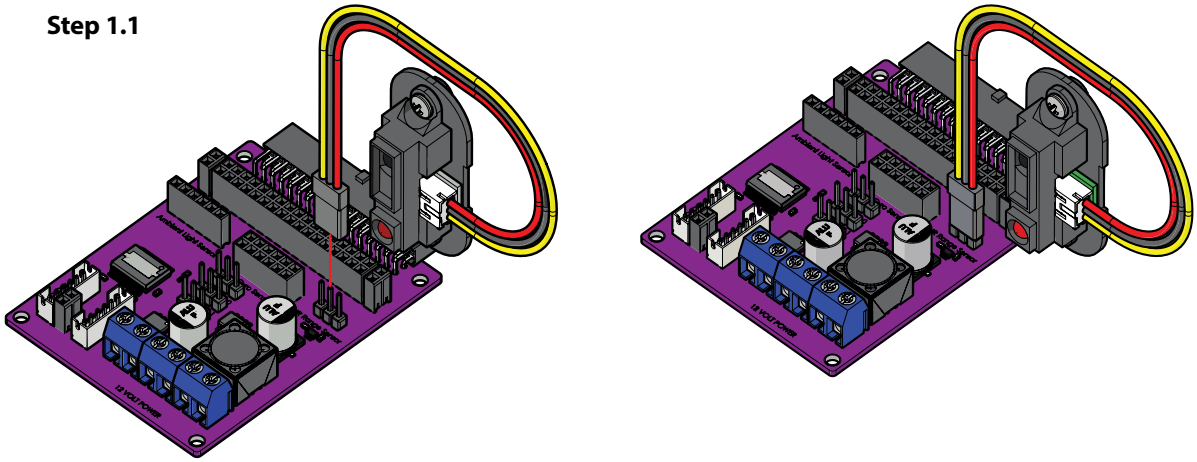
Set Up and Test the IR Sensor

Step 1: Connect the IR sensor to the motor controller board

Step 1.0



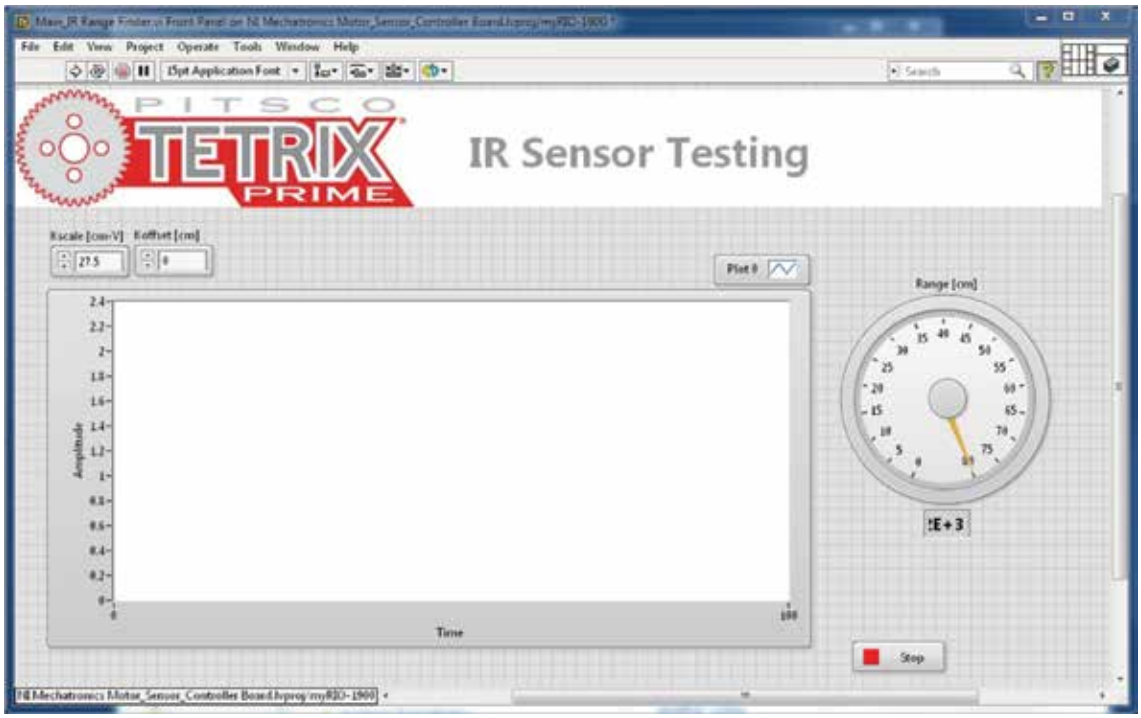
Step 1.1



Step 2: Insert the motor board into myRIO MXP A, plug the battery cable into the battery, and plug the DC connector into the myRIO power port to power it on. Allow myRIO about 30 seconds to boot.

Step 3: After completing the LabVIEW and myRIO Setup instructions, navigate to "…\Desktop\National Instruments\Pitsco Tetrix Prime for NI myRIO\Test Code" and open "Control System Test Project." Connect to myRIO by right clicking myRIO-1900 and clicking connect. Allow a few moments to connect. Expand the myRIO dropdown, and navigate to "Testing Your System > Main_IR Range Finder.vi." Feel free to explore the code on the block diagram by pressing Ctrl + E, and then press Ctrl + E again to return to the front panel.

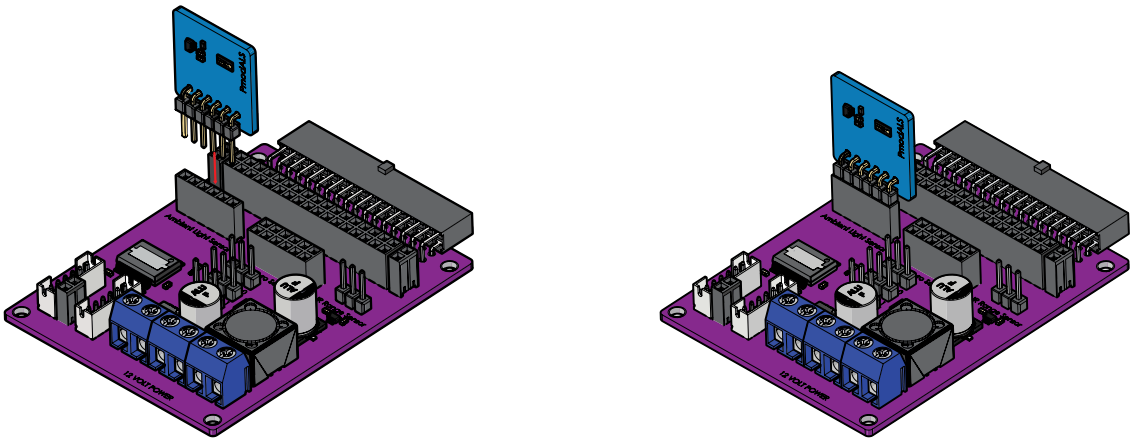
Set Up and Test the IR Sensor (continued)



Step 4: Run the VI by pressing the arrow in the left hand corner. Place your hand about 10 inches away from the sensor and move it back and forth. Confirm that the voltage decreases as your hand moves away from the sensor and that it increases as your hand moves closer. Stop the VI using the stop button, and close the VI.

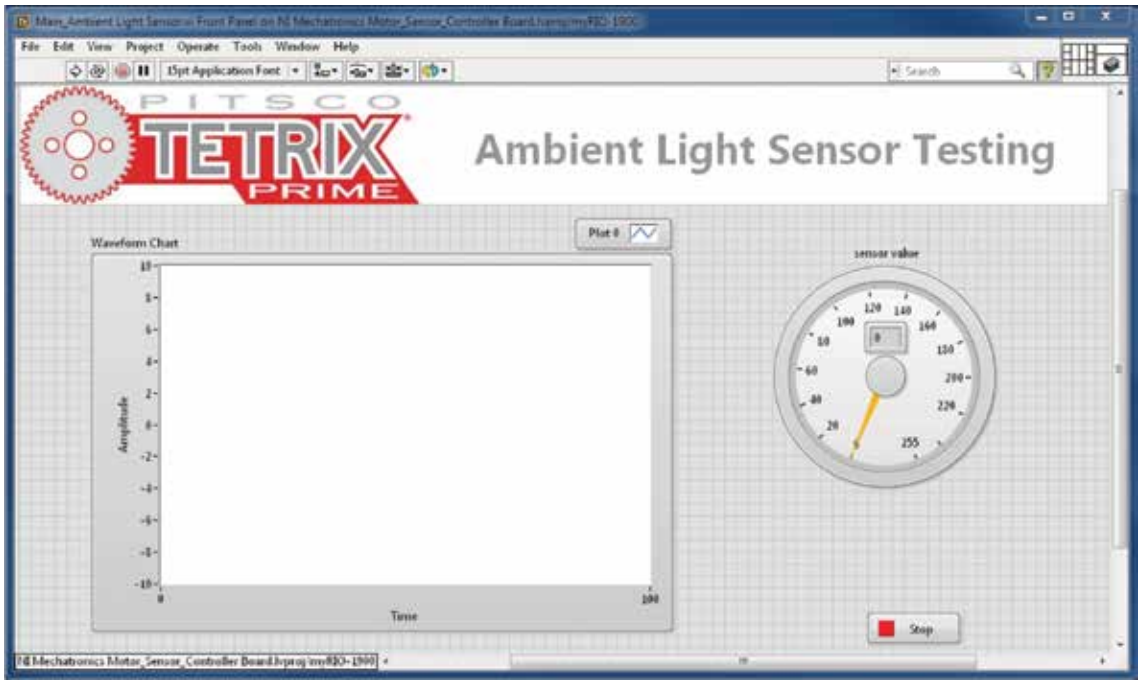
Set Up and Test the Ambient Light Sensor

Step 1: Connect the ambient light sensor to the motor controller board



Step 2: Insert the motor board into myRIO MXP A, plug the battery cable into the battery, and plug the DC connector into the myRIO power port to power it on. Allow myRIO about 30 seconds to boot.

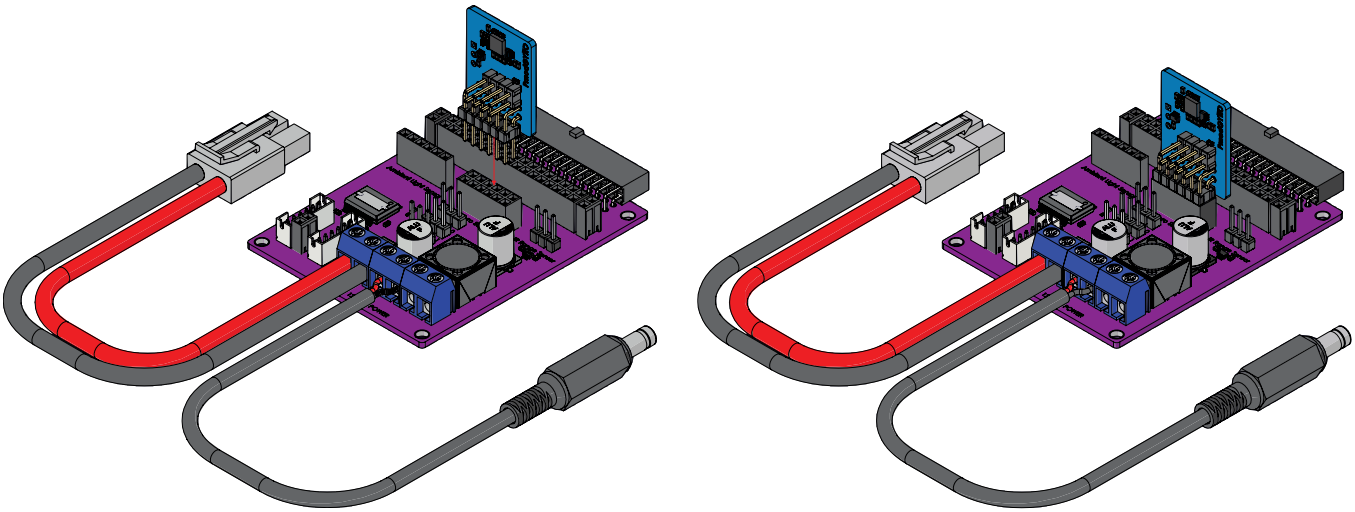
Step 3: After completing the LabVIEW and myRIO Setup instructions, navigate to "…\Desktop\National Instruments\Pitsco Tetrix Prime for NI myRIO\Test Code" and open "Control System Test Project." Connect to myRIO by right clicking myRIO-1900 and clicking connect. Allow a few moments to connect. Expand the myRIO dropdown, and navigate to "Testing Your System > Main_Ambient Light Sensor.vi." Feel free to explore the code on the block diagram by pressing Ctrl + E, and then press Ctrl + E again to return to the front panel.



Step 4: Run the VI by pressing the arrow in the left hand corner. Position the sensor to point towards a light source to confirm the data increases. Completely cover the sensor to confirm the data decreases. Stop the VI using the stop button, and close the VI.

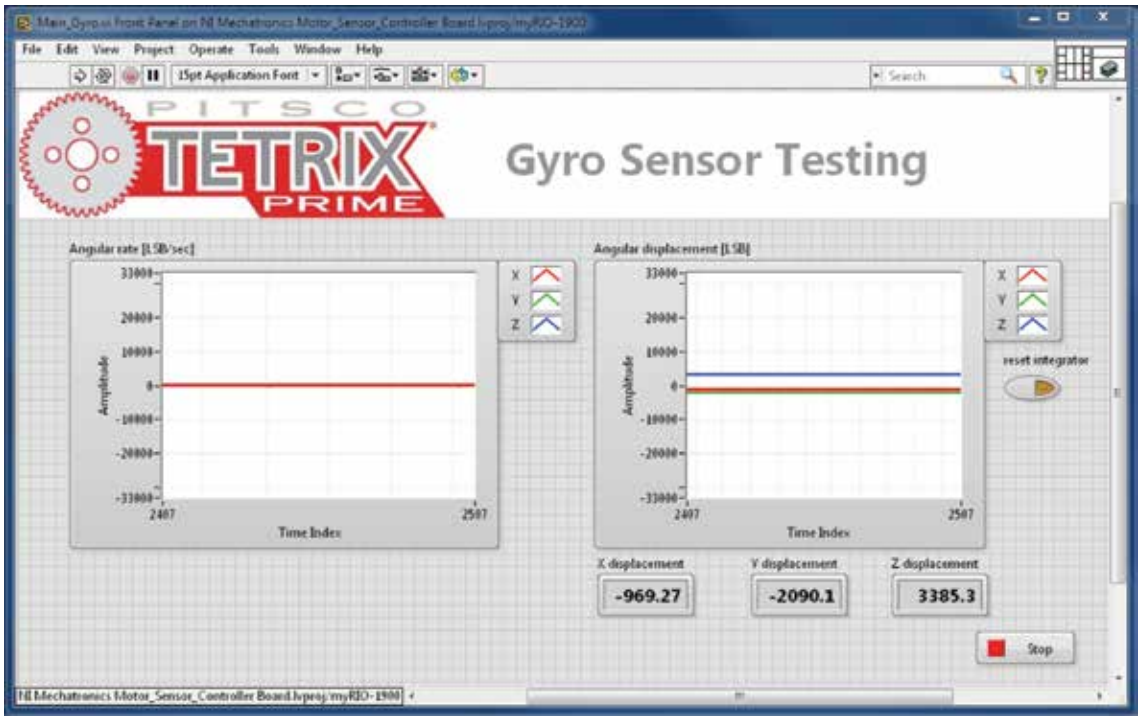
Set Up and Test the Gyroscope Sensor

Step 1: Connect the ambient light sensor to the motor controller board



Step 2: Insert the motor board into myRIO MXP A, plug the battery cable into the battery, and plug the DC connector into the myRIO power port to power it on. Allow myRIO about 30 seconds to boot.

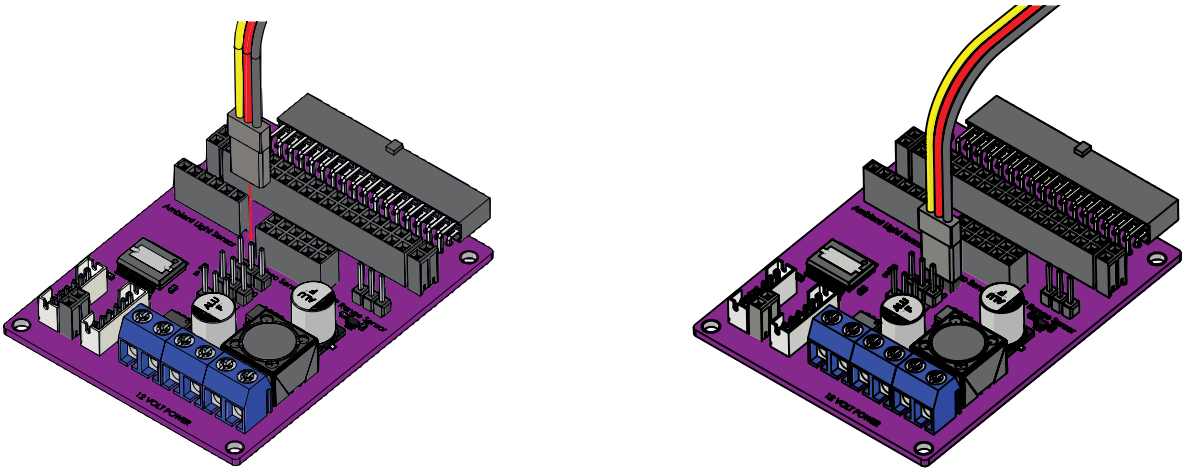
Step 3: After completing the LabVIEW and myRIO Setup instructions, navigate to "…\Desktop\National Instruments\Pitsco Tetrix Prime for NI myRIO\Test Code" and open "Control System Test Project." Connect to myRIO by right clicking myRIO-1900 and clicking connect. Allow a few moments to connect. Expand the myRIO dropdown, and navigate to "Testing Your System > Main_Gyro.vi." Feel free to explore the code on the block diagram by pressing Ctrl + E, and then press Ctrl + E again to return to the front panel.



Step 4: Run the VI by pressing the arrow in the left hand corner. Move the hardware in all different directions to confirm that the angular rates for each axis change depending on the direction of motion. Continue moving the hardware to confirm that the angular displacement for each axis changes depending on the orientation of the hardware. Stop moving the hardware to confirm that the angular rates return to zero and that the angular displacement values stop changing. Stop the VI using the stop button, and close the VI.

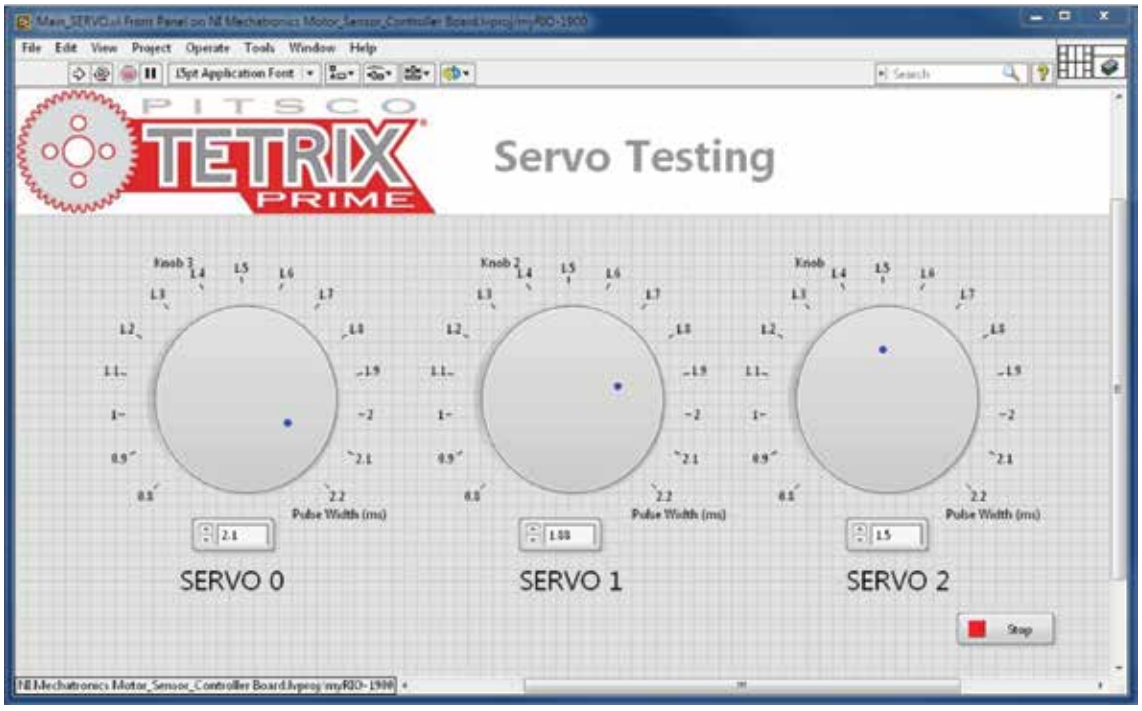
Set Up and Test the Servo Motor

Step 1: Connect the servo to the motor controller board using the port Servo0



Step 2: Insert the motor board into myRIO MXP A, plug the battery cable into the battery, and plug the DC connector into the myRIO power port to power it on. Allow myRIO about 30 seconds to boot.

Step 3: After completing the LabVIEW and myRIO Setup instructions, navigate to "…\Desktop\National Instruments\Pitsco Tetrix Prime for NI myRIO\Test Code" and open "Control System Test Project." Connect to myRIO by right clicking myRIO-1900 and clicking connect. Allow a few moments to connect. Expand the myRIO dropdown, and navigate to "Testing Your System > Main_SERVO.vi." Feel free to explore the code on the block diagram by pressing Ctrl + E, and then press Ctrl + E again to return to the front panel.



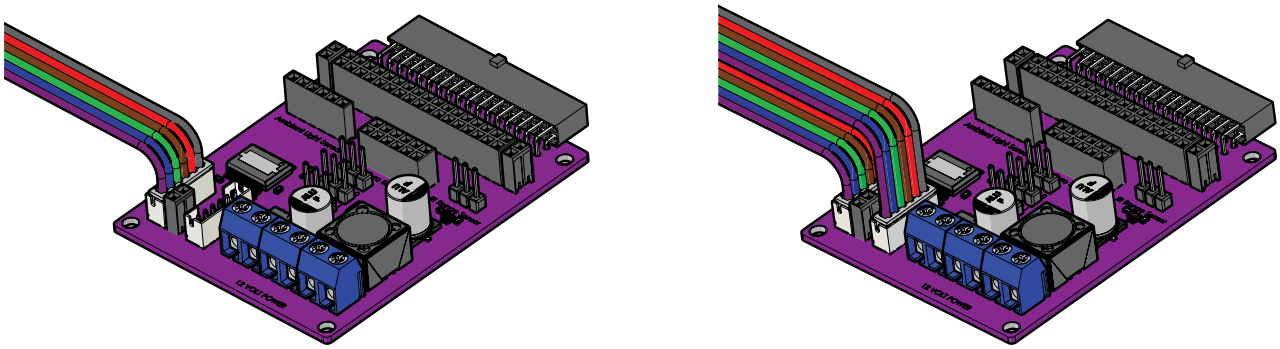
Step 4: Change the value of the dial labeled SERVO 0 to a few different locations. At each location, confirm that the servo shaft position turns and stays at a set position.

Step 5: Stop the VI by pressing the stop button. Unplug the servo motor from port Servo0 and move it to port Servo1. Repeat step 4 using the dial labeled SERVO 1.

Step 6: Repeat step 5 for port Servo2.

Set Up and Test the DC Motor and Encoder

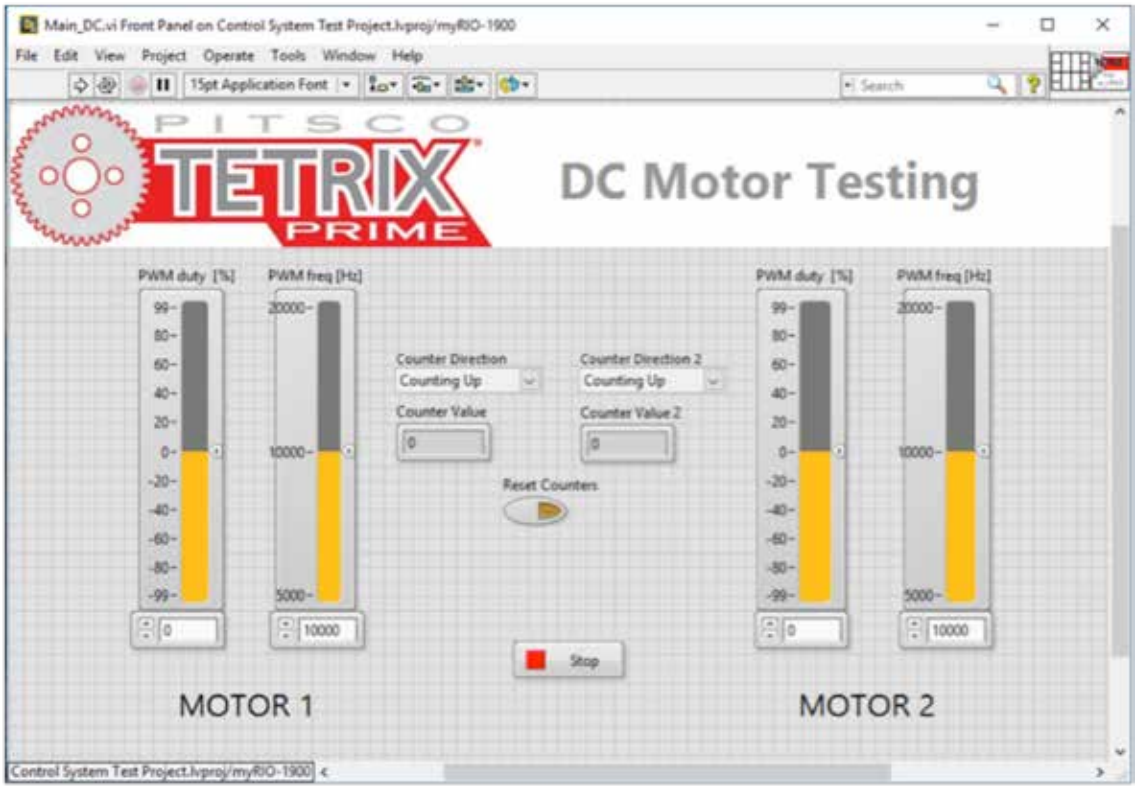
Step 1: Connect both DC Motors to the motor controller board using DC Motor ports 1 and 2.



Step 2: For the Motor2 port, the encoder pins are instead connected to a header port. To connect the Motor2 encoder, use jumper wires to connect pins of the header port to the set of quadrature encoder pins on the MXP port B. Pin A of the header port should connect to MXP B pin 18. Port B of the header port should connect to MXP B pin 22.

Step 3: Insert the motor board into myRIO MXP A, plug the battery cable into the battery, and plug the DC connector into the myRIO power port to power it on. Allow myRIO about 30 seconds to boot.

Step 4: After completing the LabVIEW and myRIO Setup instructions, navigate to "…\Desktop\National Instruments\Pitsco Tetrix Prime for NI myRIO\Test Code" and open "Control System Test Project." Connect to myRIO by right clicking myRIO-1900 and clicking connect. Allow a few moments to connect. Expand the myRIO dropdown, and navigate to "Testing Your System > Main_DC.vi." Feel free to explore the code on the block diagram by pressing Ctrl + E, and then press Ctrl + E again to return to the front panel.



Set Up and Test the DC Motor and Encoder (continued)

Step 4: Click the "Reset Counters" button two times to set the encoder couter value to zero. Maintain the frequency of motor1 at 1000, and set the motor1 PWM duty cycle to 10. Confirm that the motor starts spinning and that the encoder counter value is increasing.

Step 5: Set the motor1 PWM duty cycle to 0 and confirm that the motor stops spinning and that the encoder counter value stops changing. Next, set the motor1 PWM duty cycle to -10 and confirm that motor1 starts spinning in the opposite direction and that the encoder counter value is decreasing.

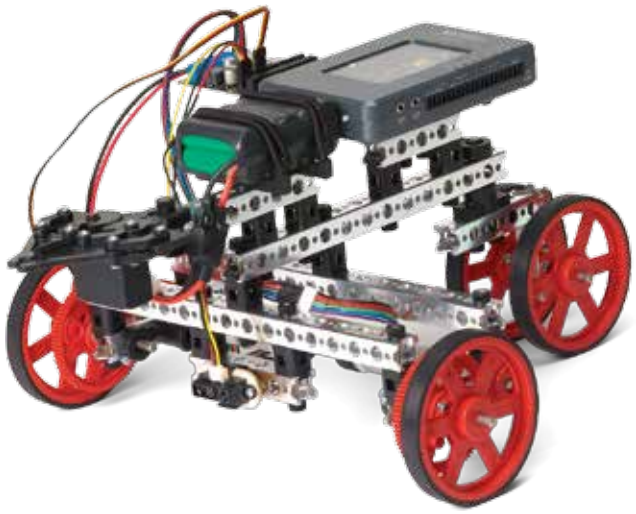
Step 6: Set the motor1 PWM frequency to 6000 and note any changes in the motor when you change its duty cycle. Set the motor1 PWM duty cycle back to zero and click the reset counters butty two times to reset the encoder counter value back to zero.

Step 7: Repeat steps 4 through 6 for motor2

Step 8: Stop the VI by pressing the stop button and close out the VI.

Rover Vehicle

Overview



The TETRIX PRIME Rover Vehicle is a versatile and fun starting point for a variety of mechatronics and robotics design projects. It is a ground vehicle controlled by myRIO and equipped with motors and sensors. You can start by following instructions to build the Rover and run the provided code. This will allow you to teleoperate the rover to travel and grasp objects with its pincer end effector. You can expand the Rover's functionality so that it can utilize controls algorithms and complete tasks.

Base Functionality

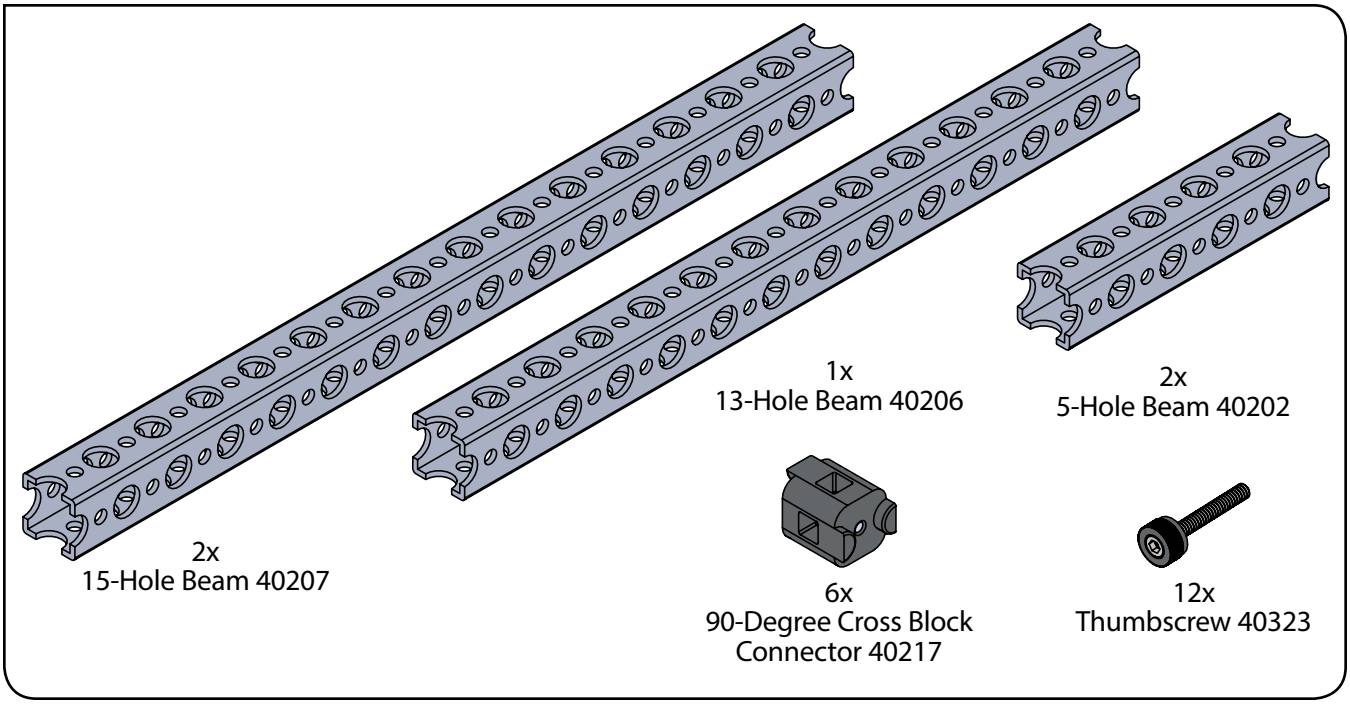
- The two front wheels are driven independently of one another by DC motors. The back wheel is used for balance and can rotate freely
- The DC motor speeds are controlled using PWM, and their directions are controlled using a digital line (this wiring is done for you via the motor board)
- The rover has differential steering, meaning that the direction of the rover can be changed by varying the relative rotational velocity of the DC motors
- The infrared (IR) range finder data is read through an analog line and converted to centimeters. It can be used to detect distance from other objects or distinguish color/material differences based on IR reflectivity
- The pincer end effector is controlled by a servo motor. The position of the servo motor is controlled by PWM
- A VI will be deployed to myRIO, enabling it to output motor signals, input sensor data, and transfer data to and from a host computer via Wi-Fi
- Another VI will run on a host computer for teleoperation. Here the user can input movement commands, open and close the pincers, and view the IR sensor data.

Expansion and Teaching Options

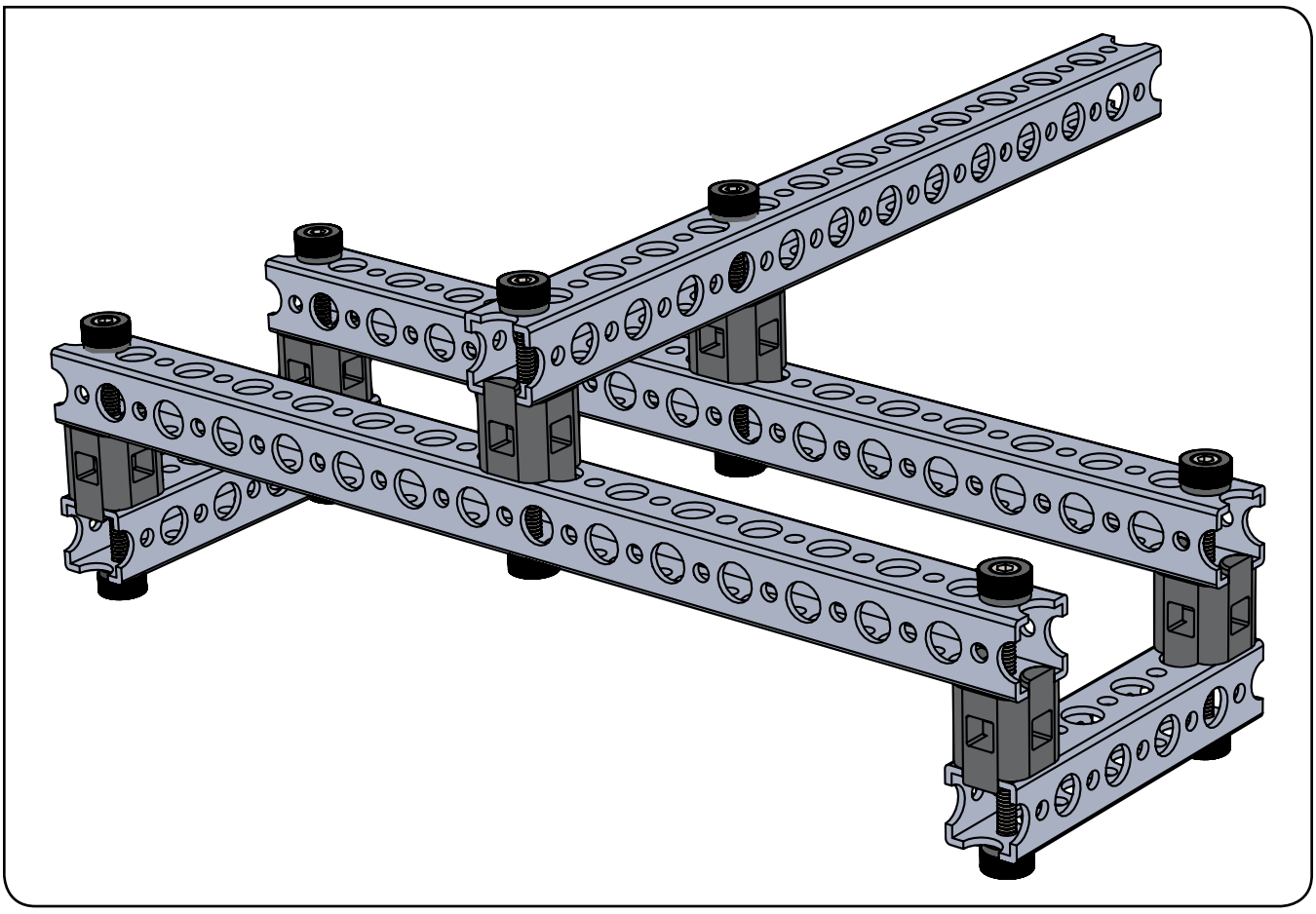
- Implement open-loop and closed-loop control algorithms in LabVIEW to precisely control the Rover's position and velocity (See the Connections to Controls Concepts at the end of this section)
- Use the IR sensor to detect objects. You can write LabVIEW code to avoid the objects or grasp them with the pincers
- Use the IR sensor to detect a line, and write LabVIEW code to follow it Program the Rover to operate autonomously so that it doesn't require user inputs from the host
- Add additional features to the rover. Example ideas: USB camera, ultrasonic sensor, custom 3-d printed parts

Step 1

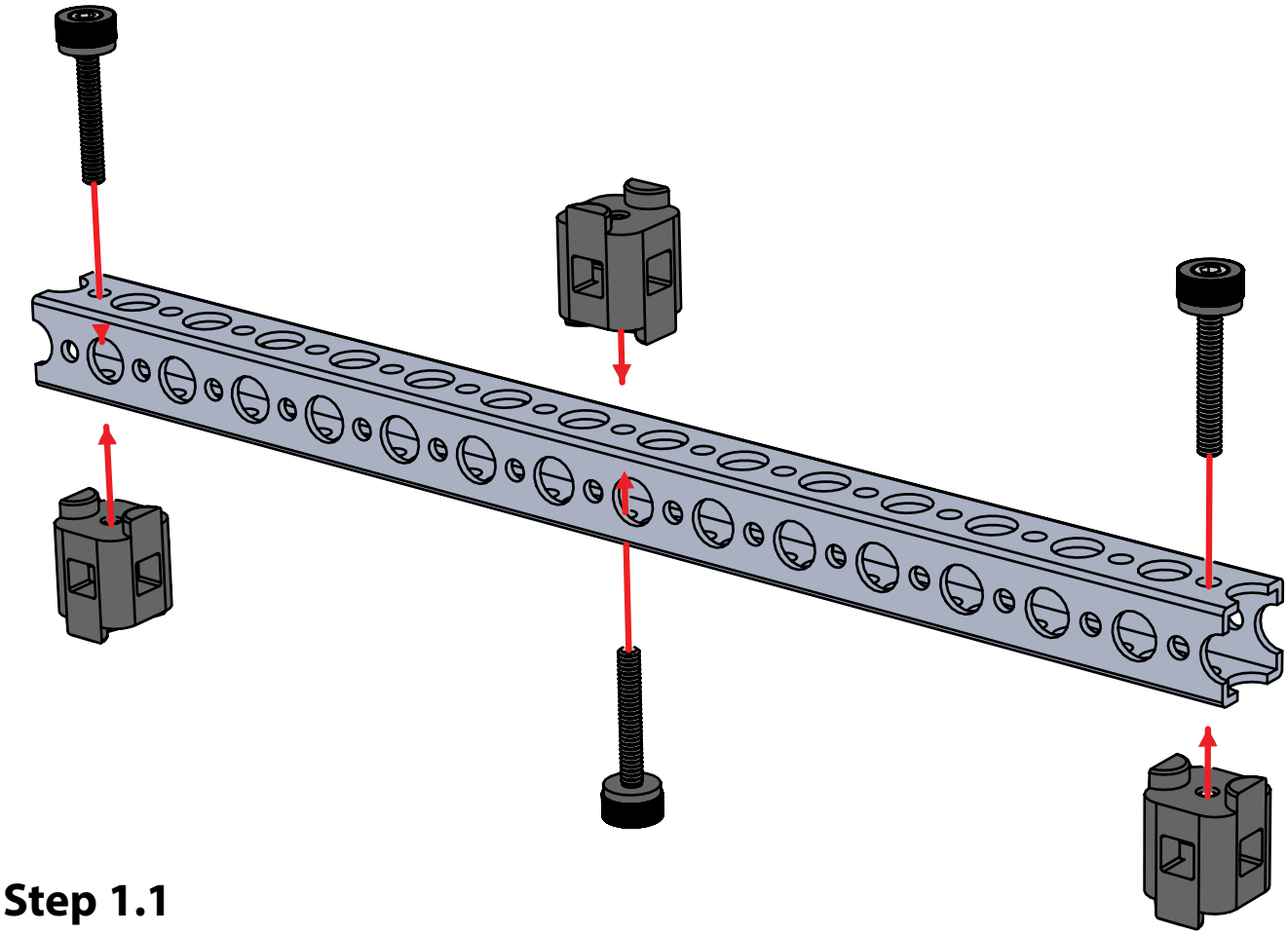
Parts Needed



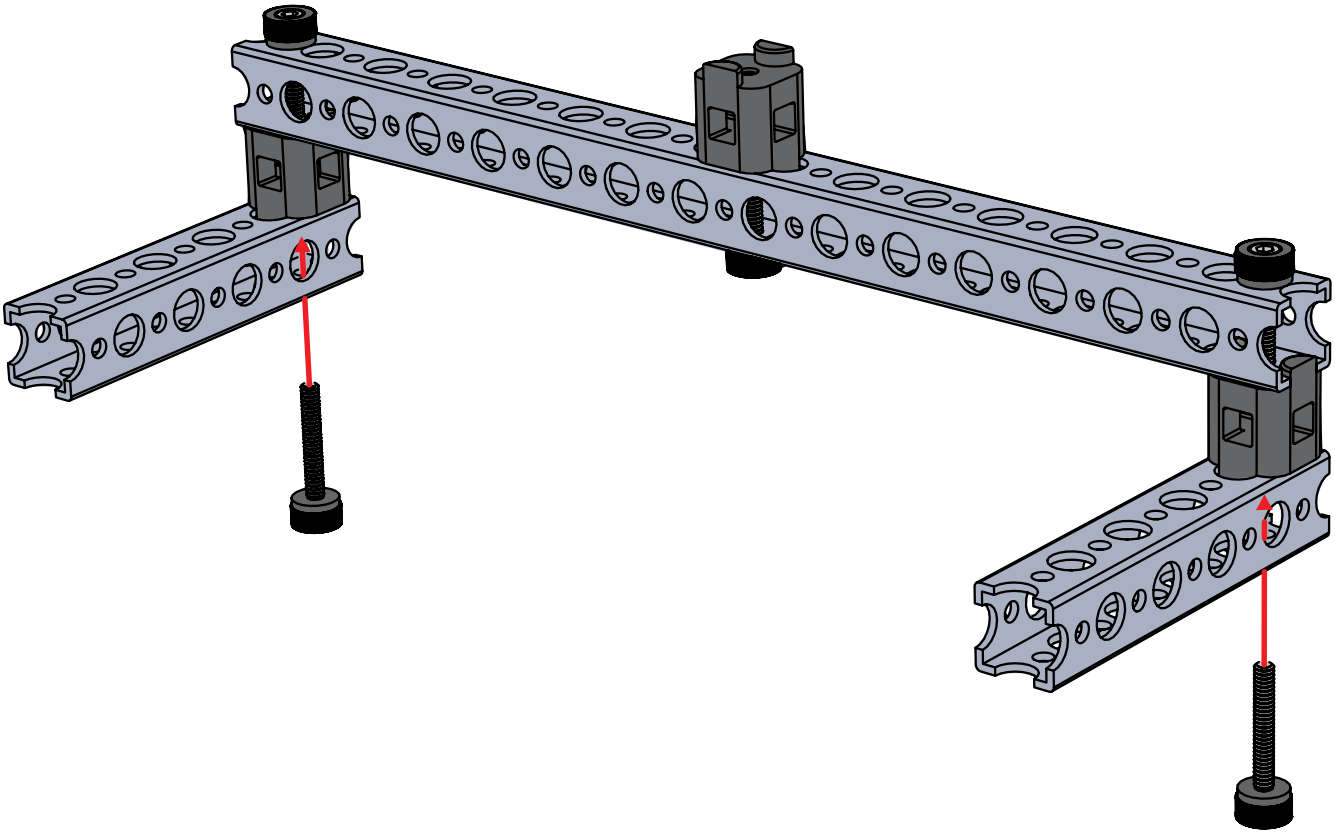
Partial assembly should look like this.



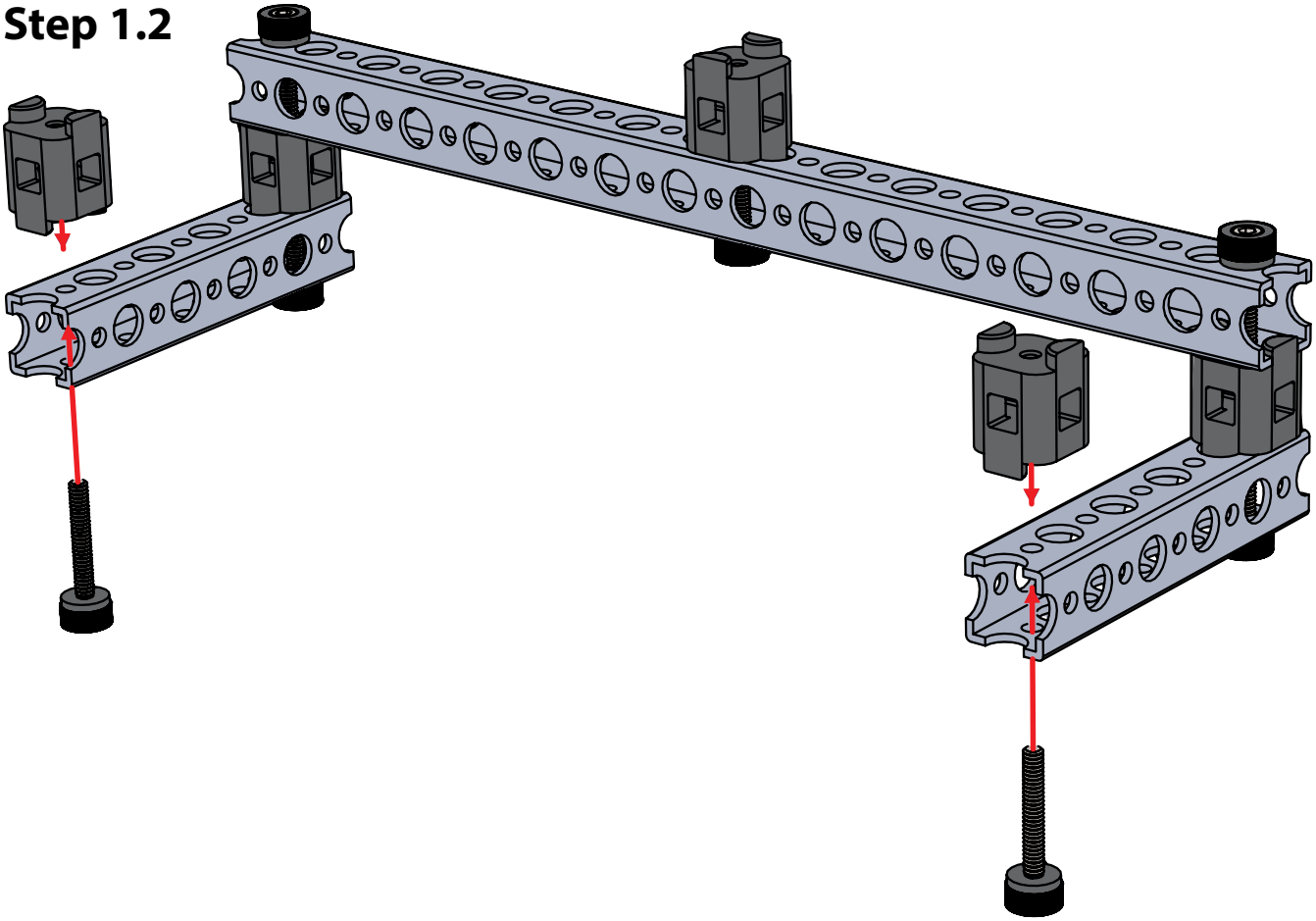
Step 1.0



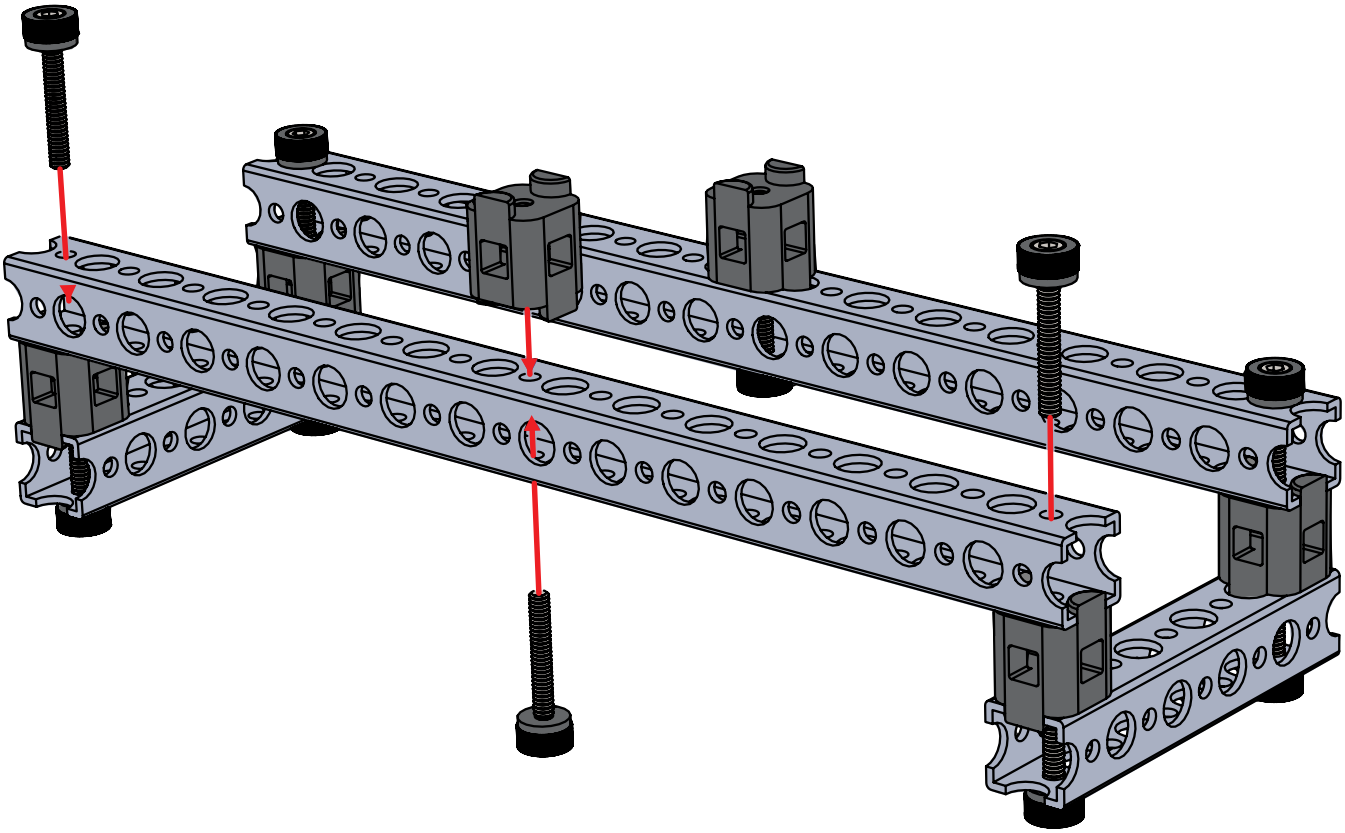
Step 1.1



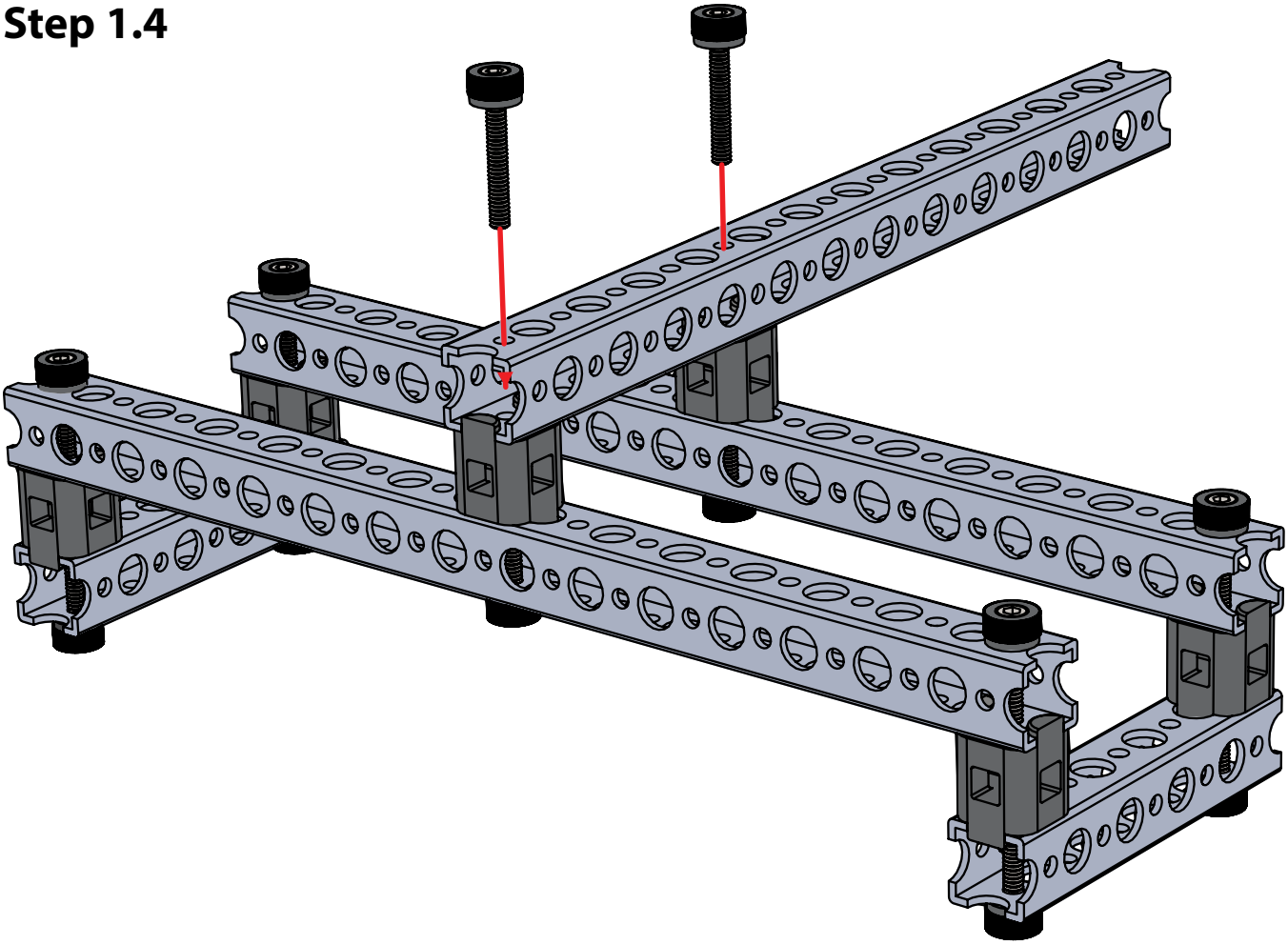
Step 1.2



Step 1.3

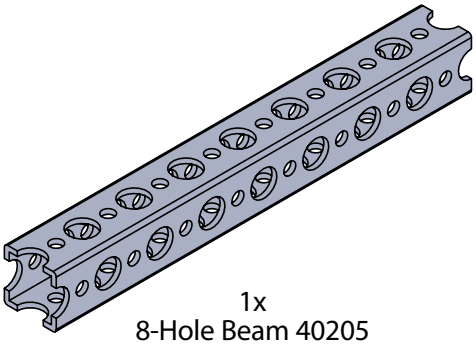


Step 1.4

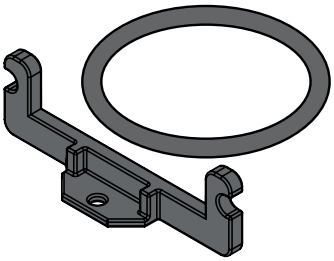


Step 2

Parts Needed

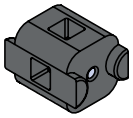


1x
8-Hole Beam 40205



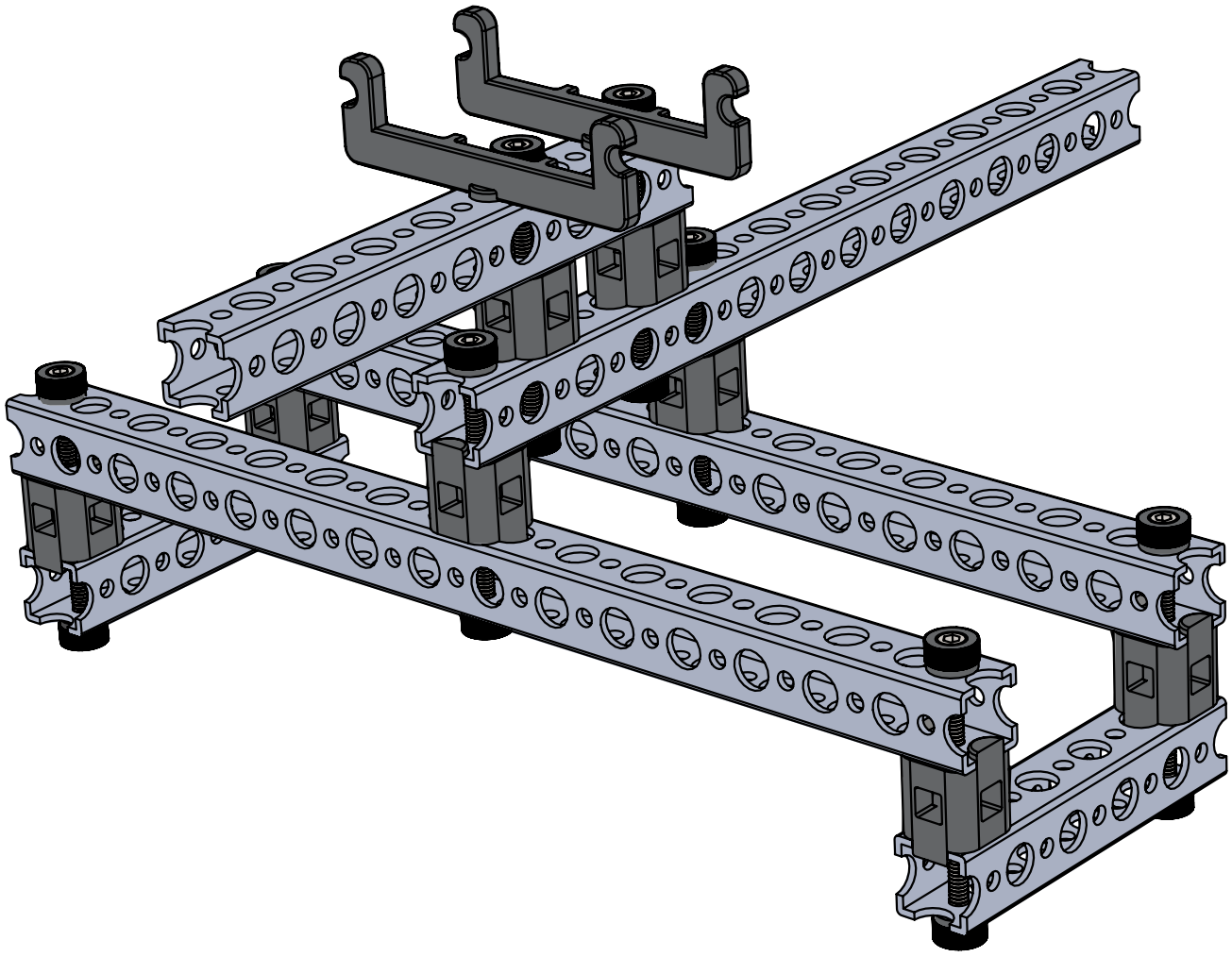
2x
Battery Holder 40236

4x
Thumbscrew 40323

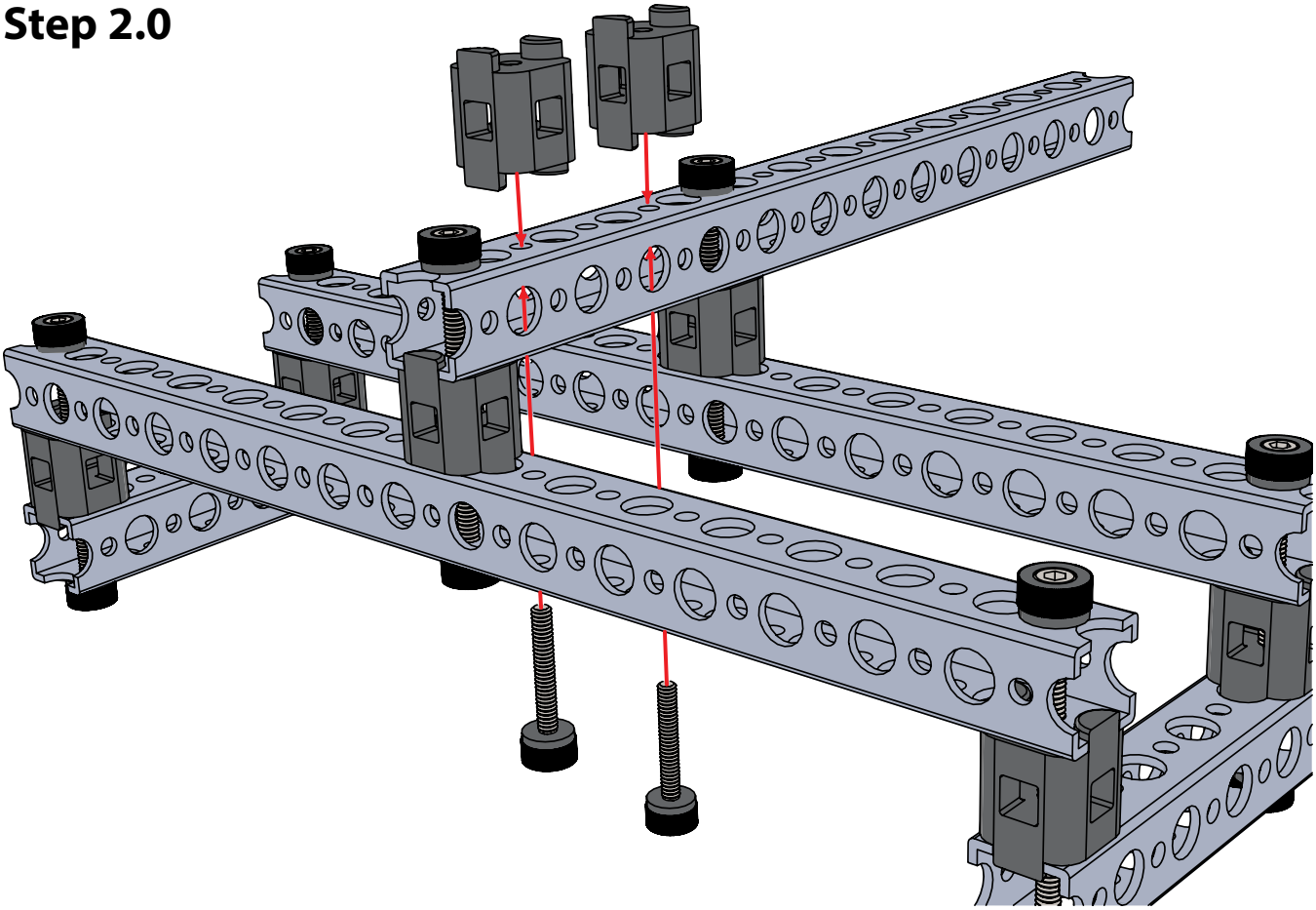


2x
Straight Block
Connector 40216

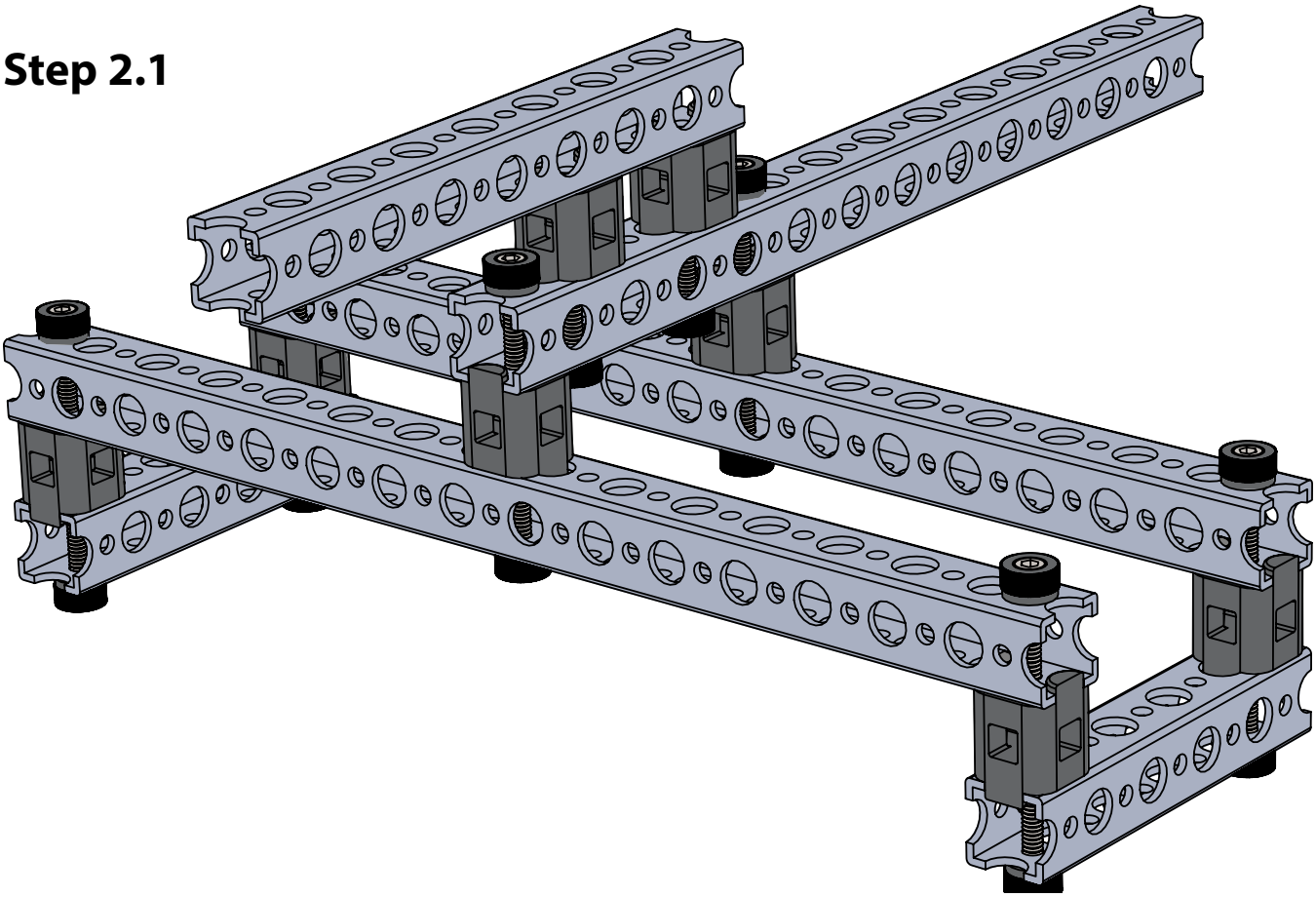
Partial assembly should look like this.



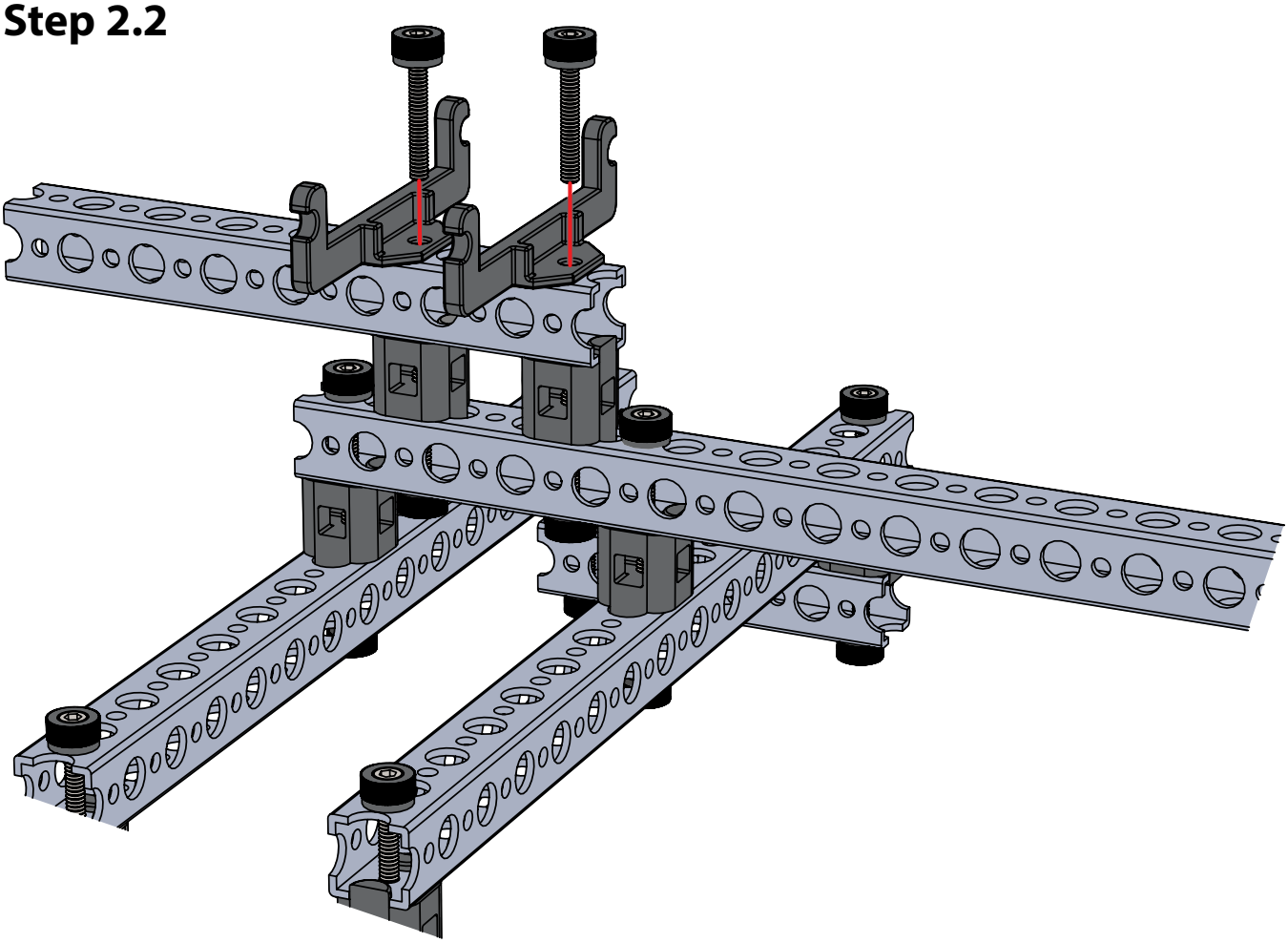
Step 2.0



Step 2.1

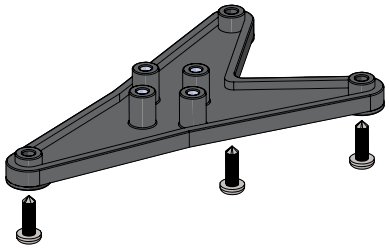


Step 2.2

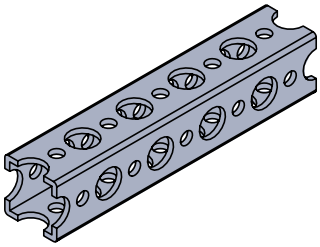


Step 3

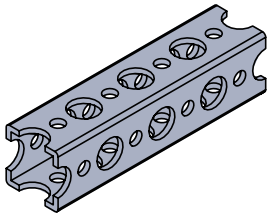
Parts Needed



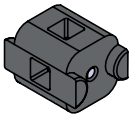
1x
myRIO Controller Mount
41137



1x
5-Hole Beam 40202



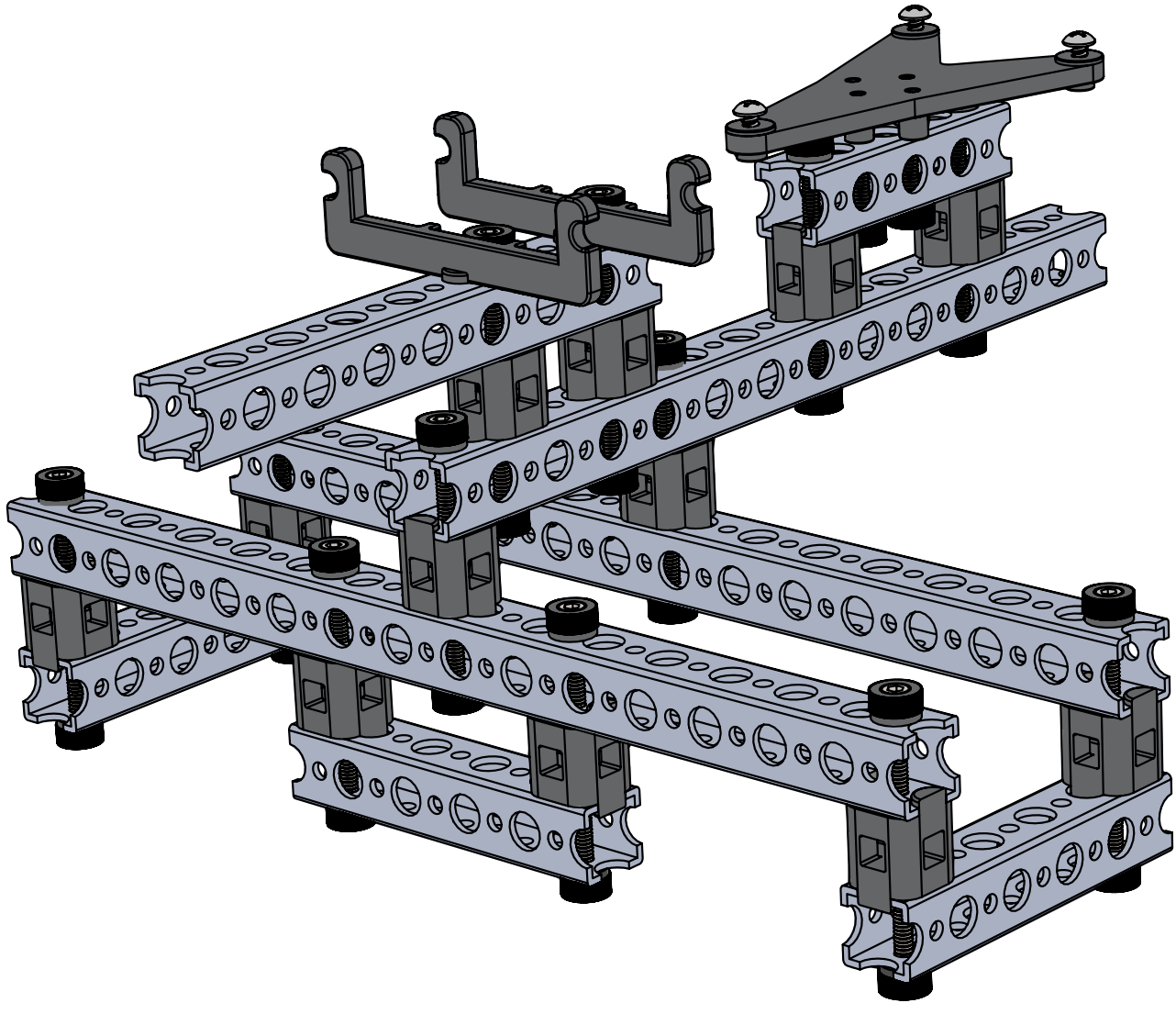
1x
4-Hole Beam 40201



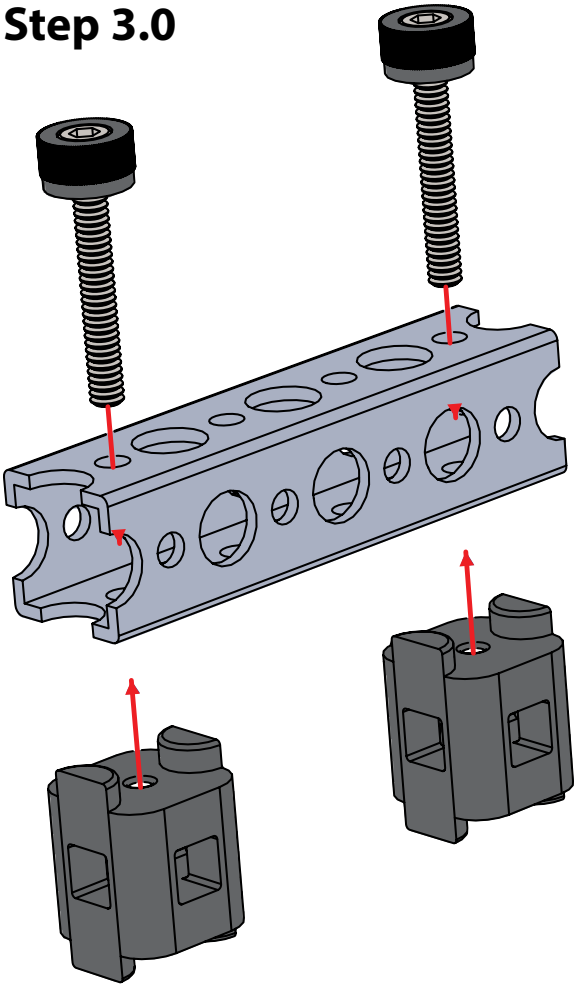
4x
Straight Block
Connector 40216

10x
Thumbscrew 40323

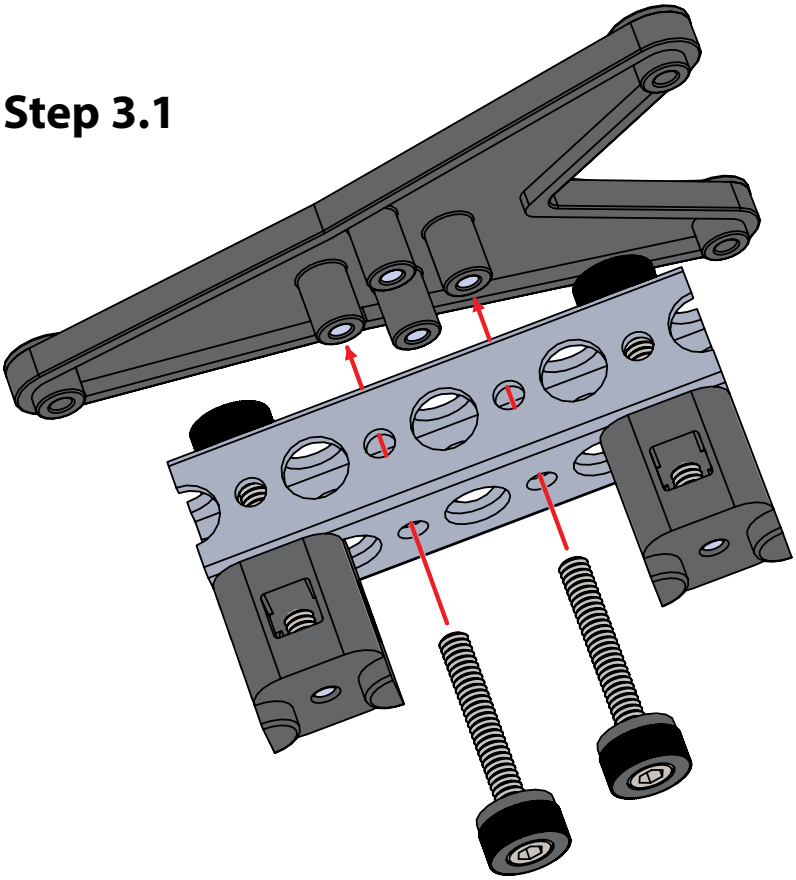
Partial assembly should look like this.



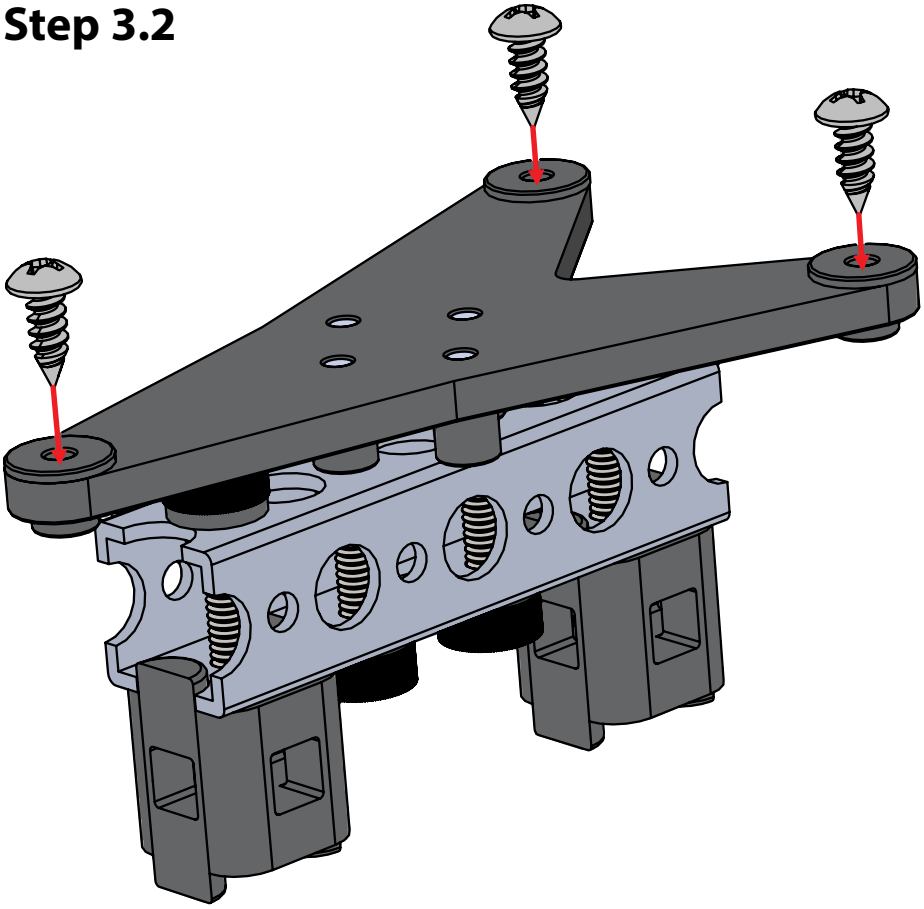
Step 3.0



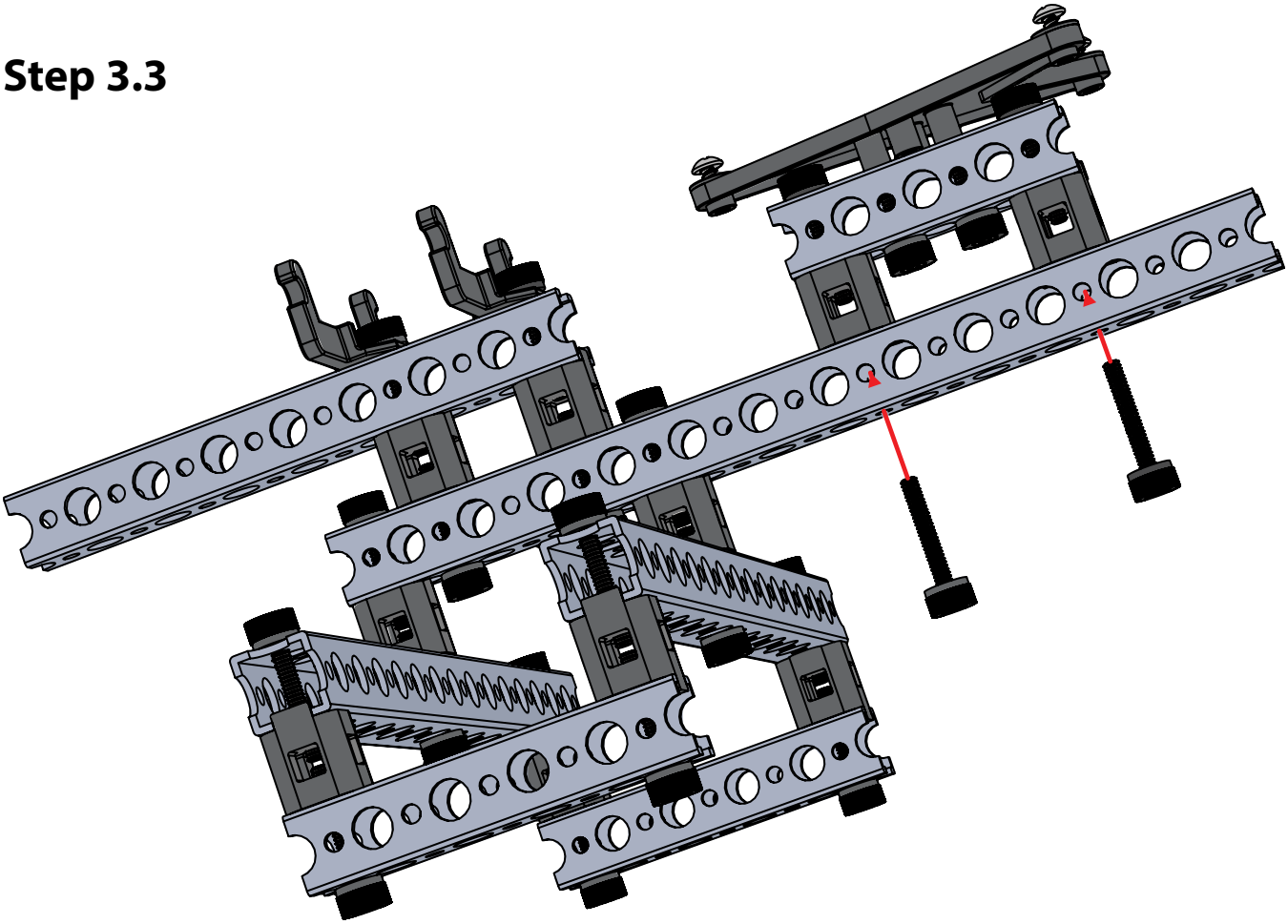
Step 3.1



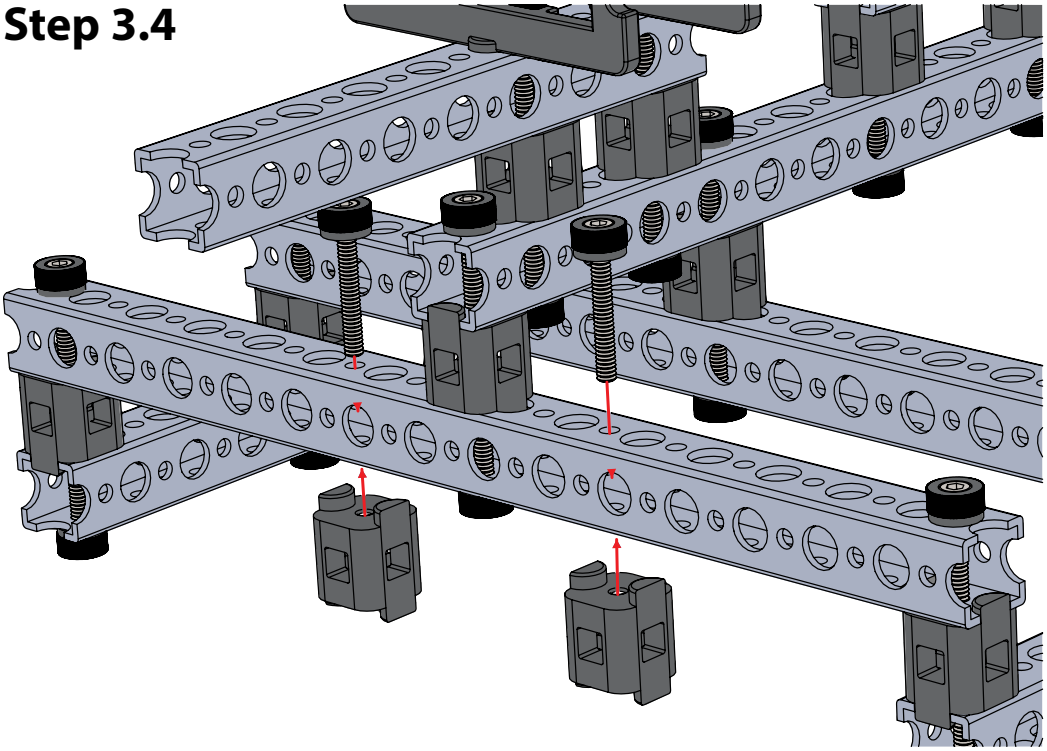
Step 3.2



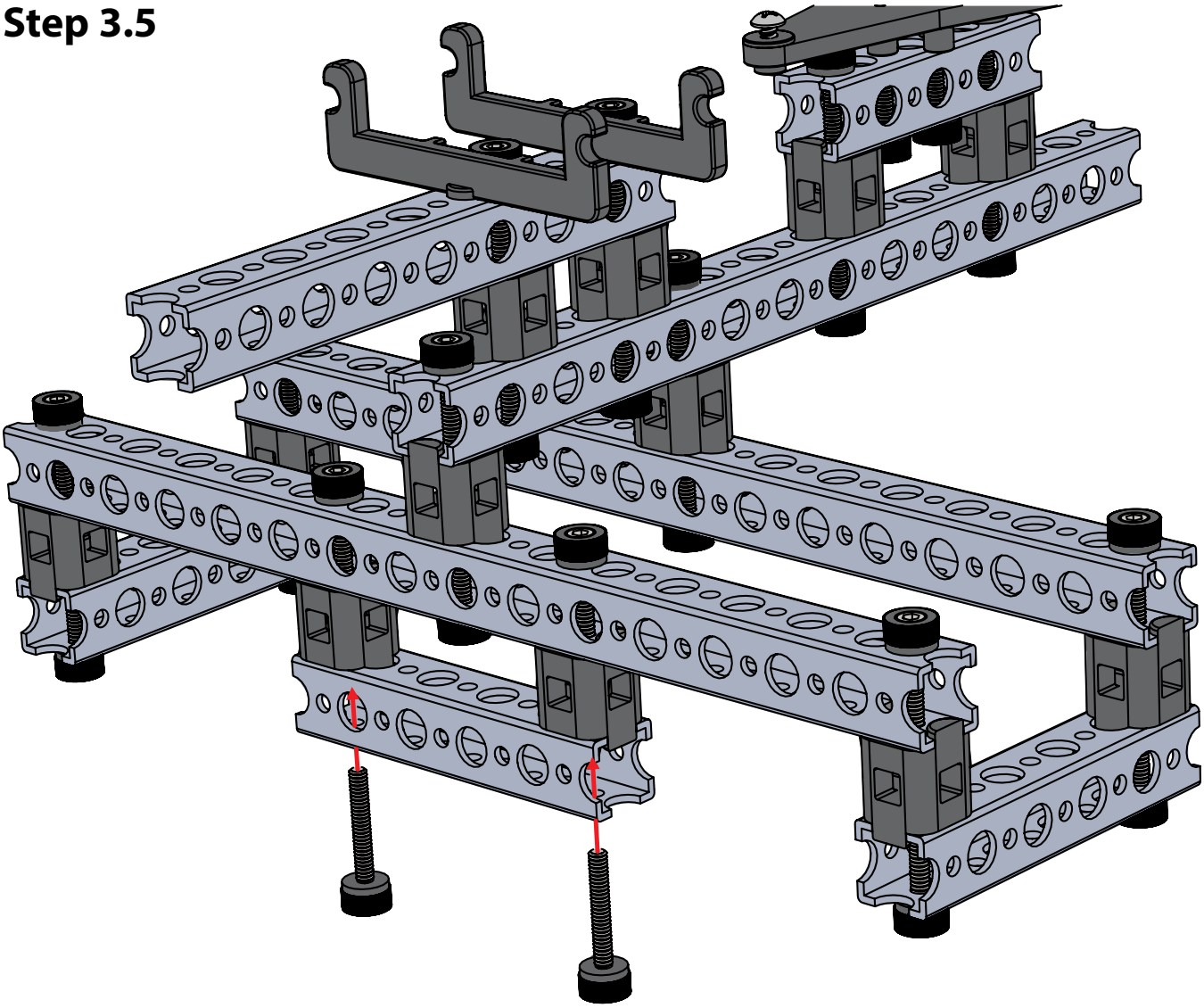
Step 3.3



Step 3.4

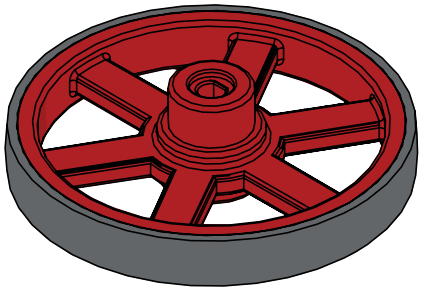


Step 3.5

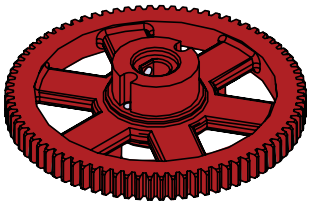


Step 4

Parts Needed

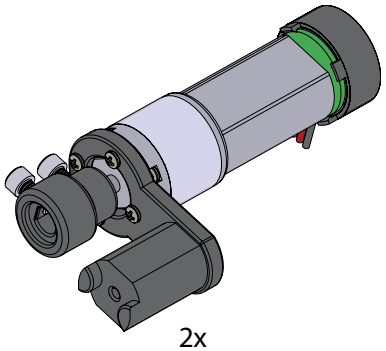


2x
90 mm Wheel & Tire 40222

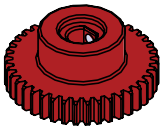


2x
80-Tooth Gear 40224

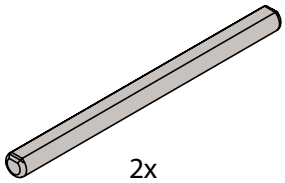
8x
Bronze Bushing 40227



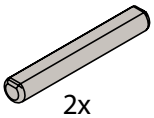
2x
Digilent DC Motor and Mount



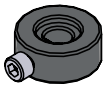
2x
40-Tooth Gear 40223



2x
80 mm Axle 40225



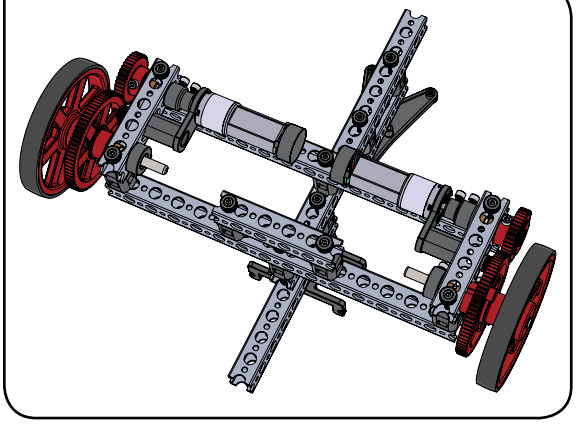
2x
40 mm Axle 40226



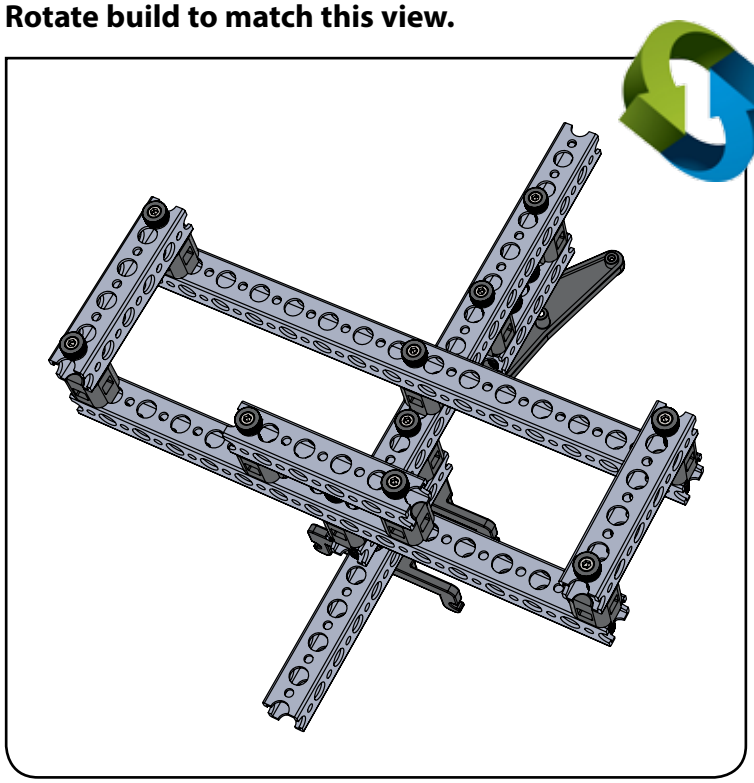
2x
Shaft Collar 40229

2x
Thumbscrew 40323

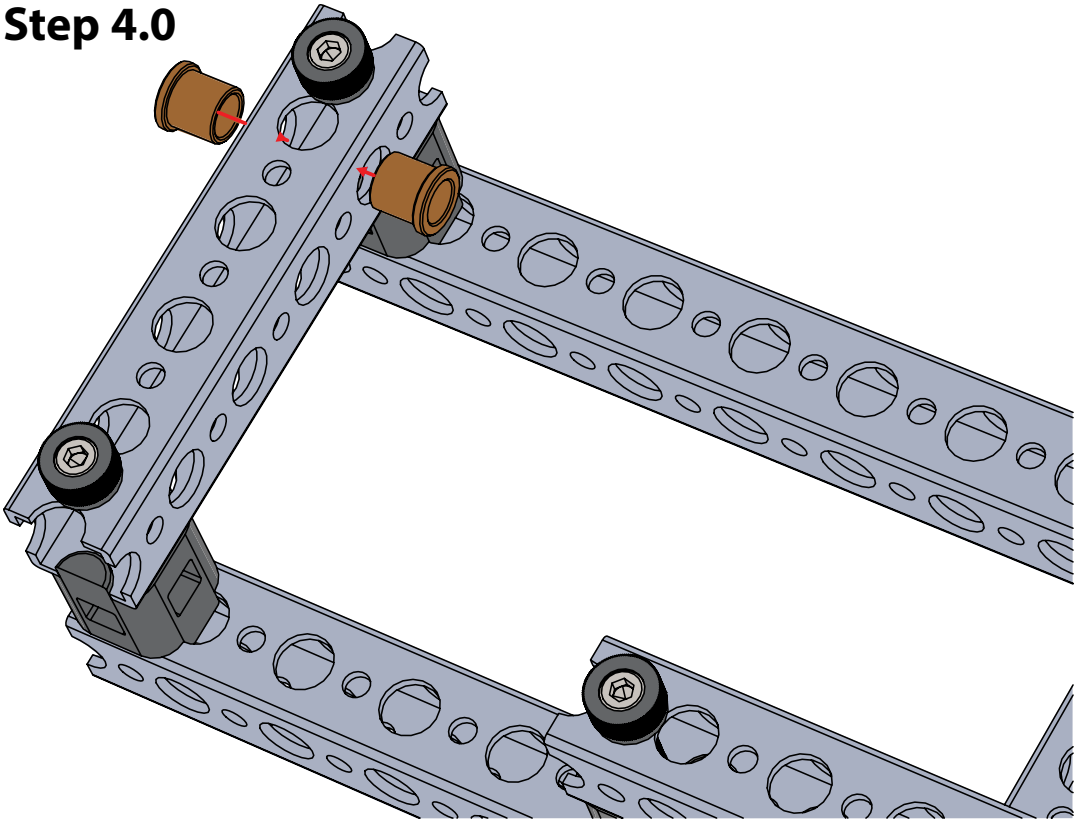
Partial assembly should look like this.



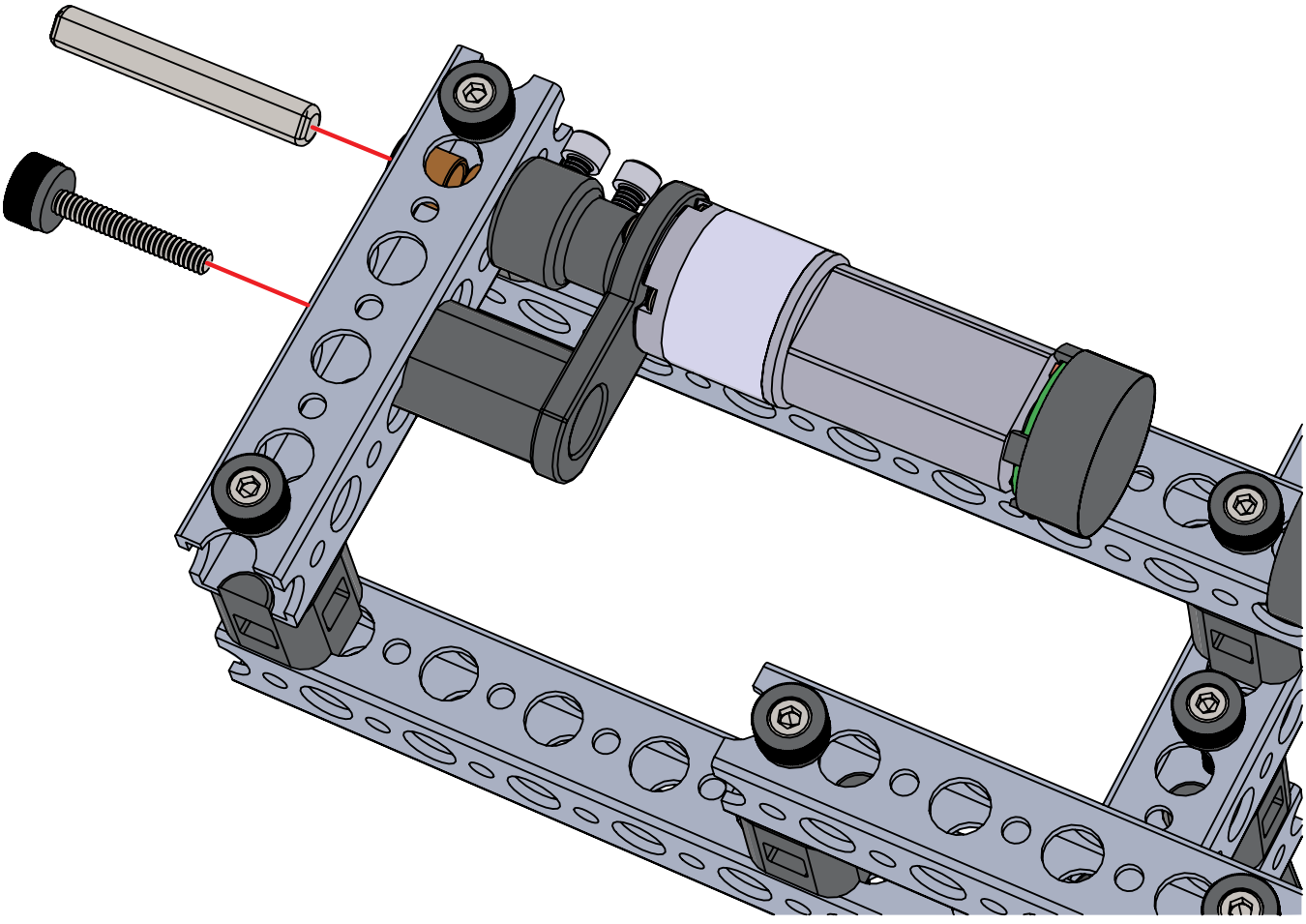
Rotate build to match this view.



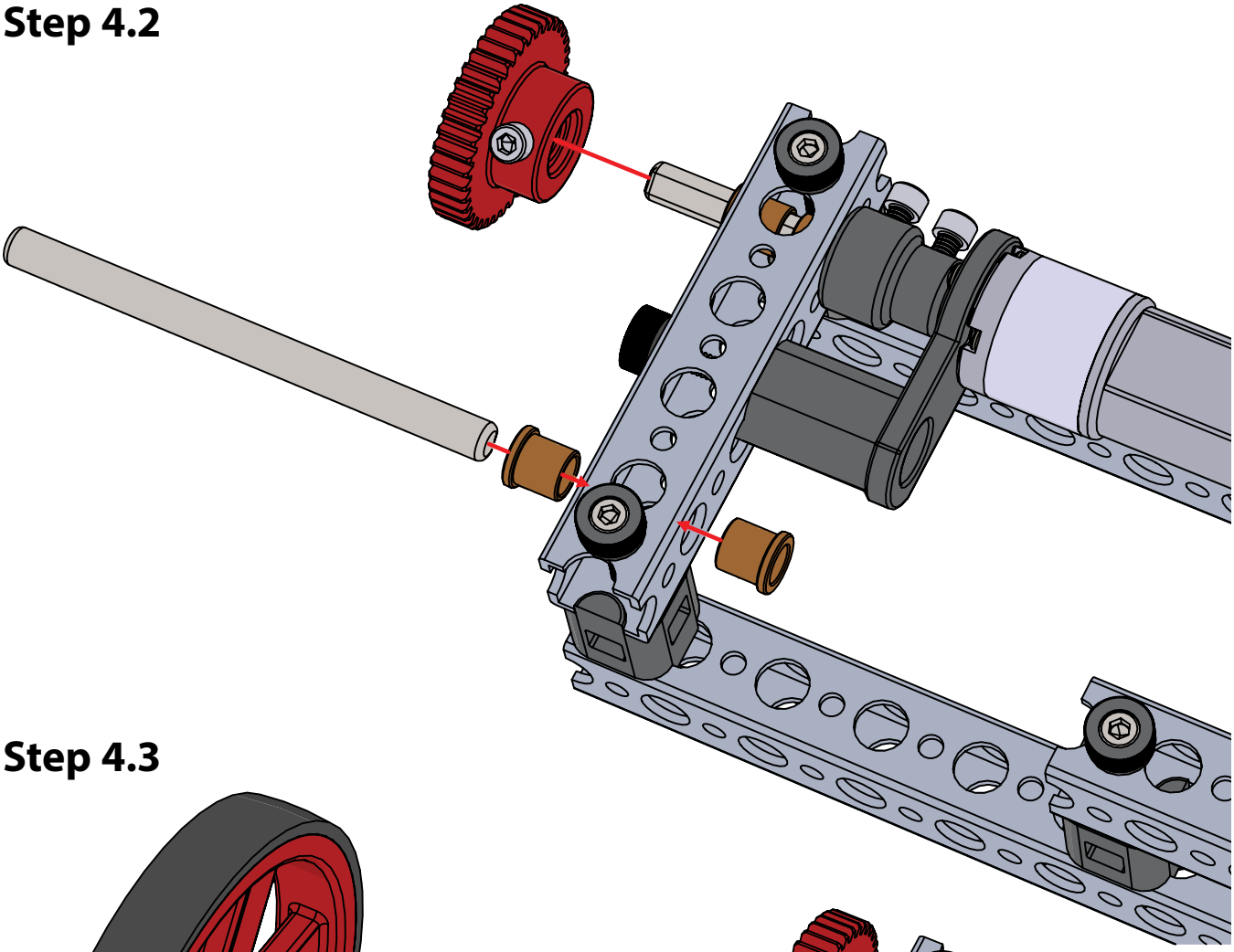
Step 4.0



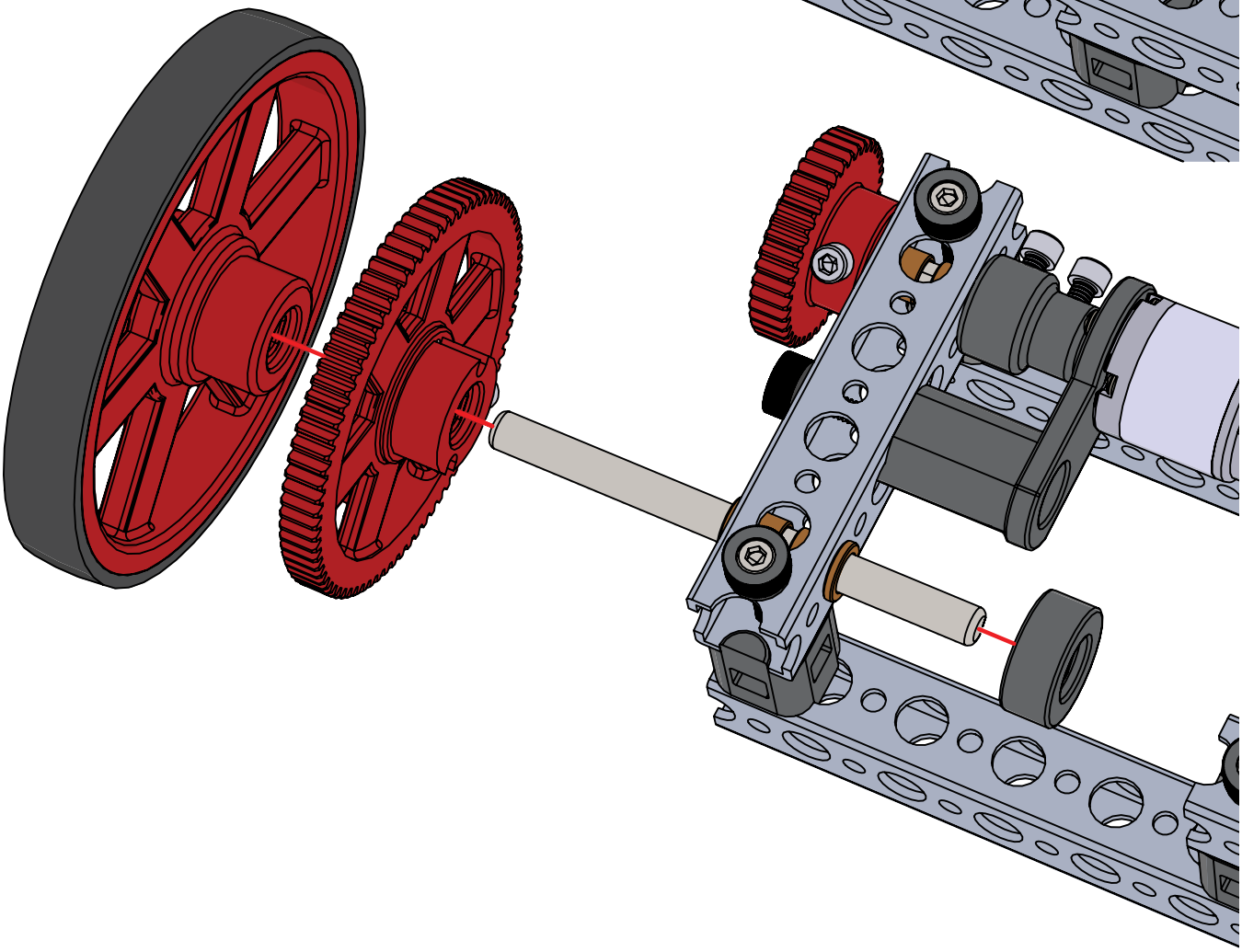
Step 4.1



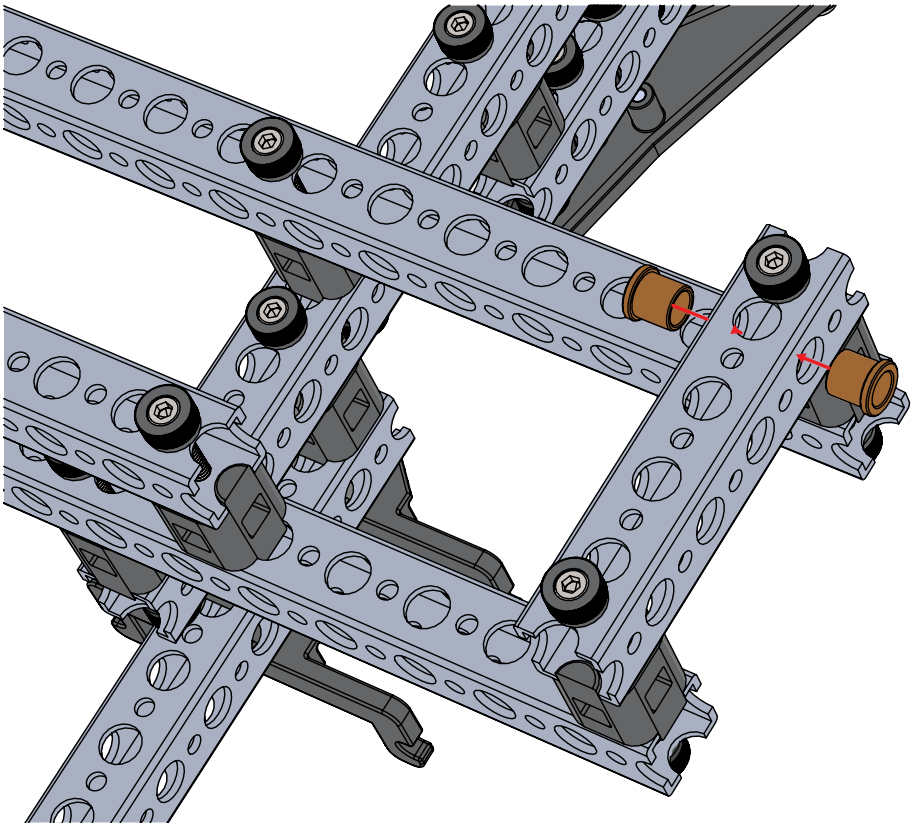
Step 4.2



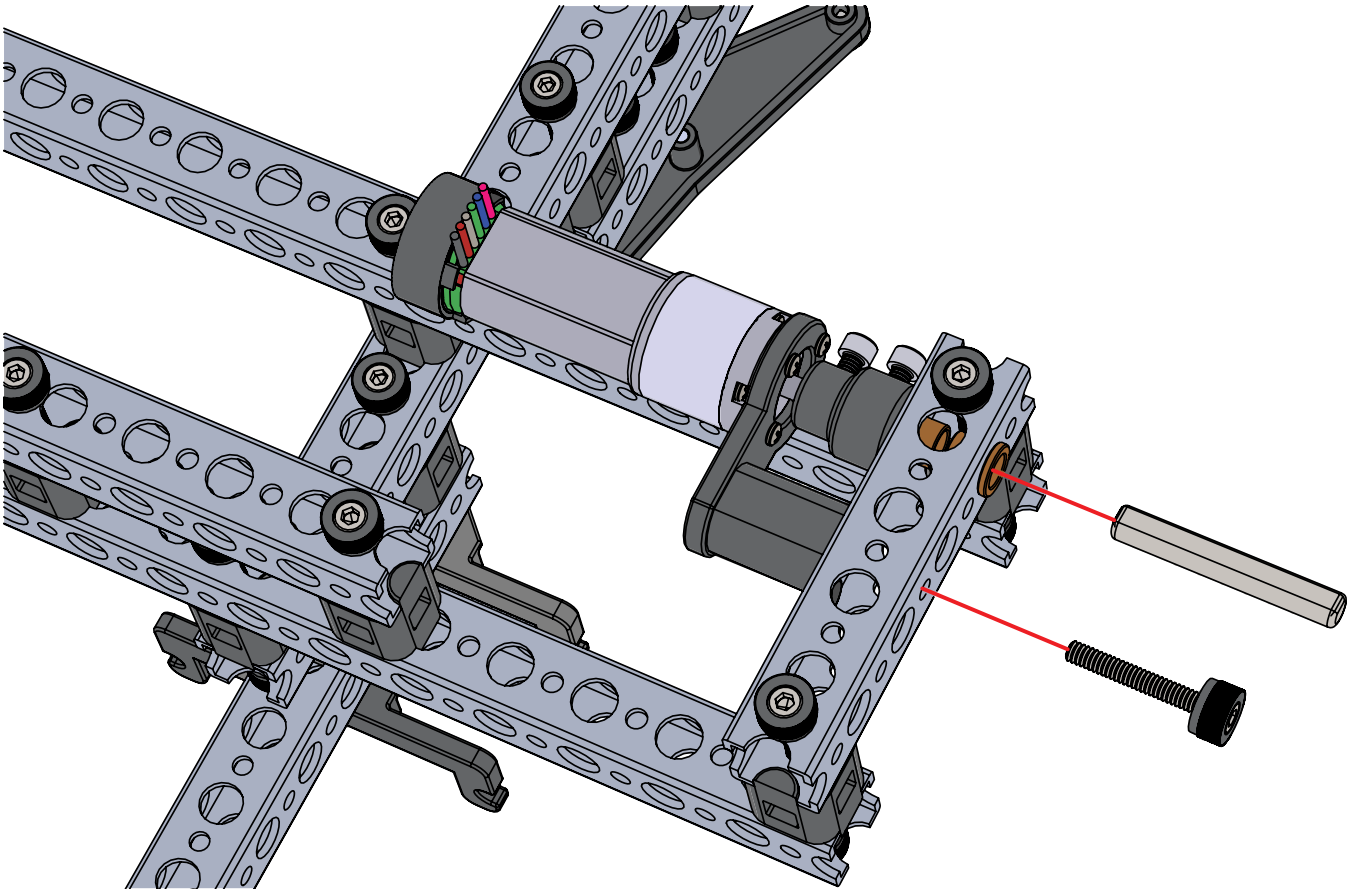
Step 4.3



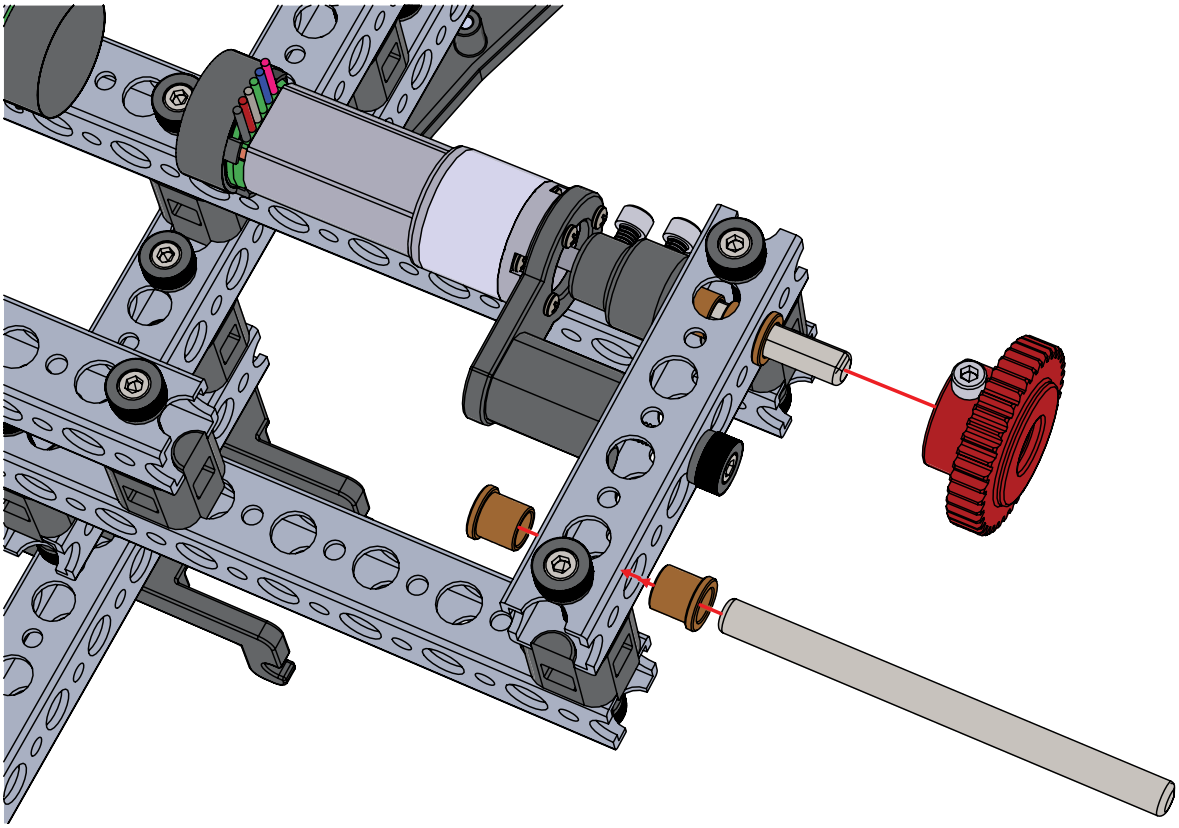
Step 4.4



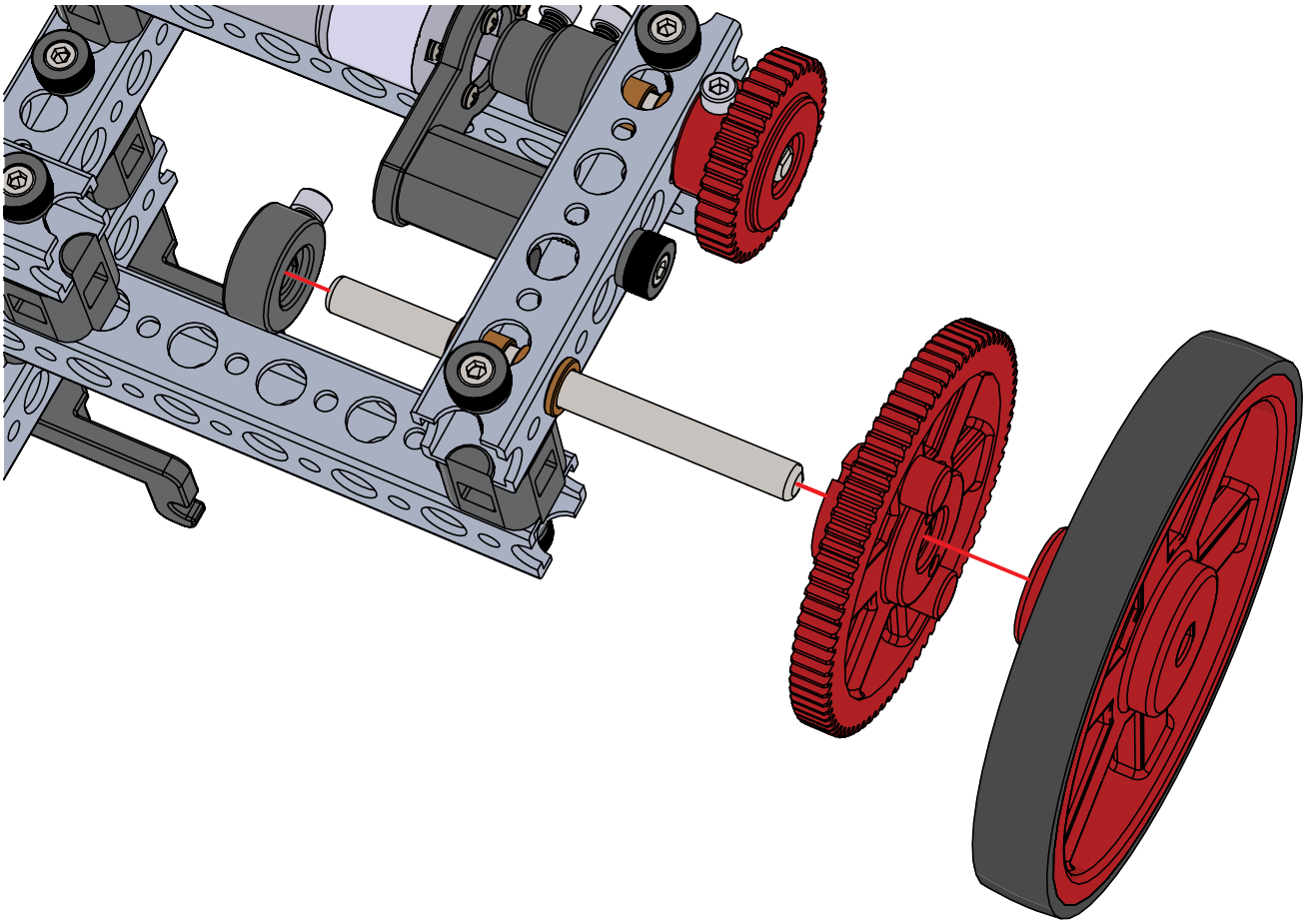
Step 4.5



Step 4.6

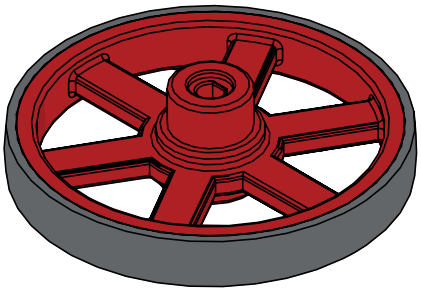


Step 4.7

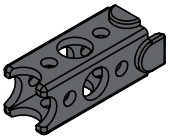


Step 5

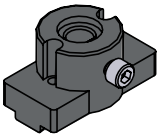
Parts Needed



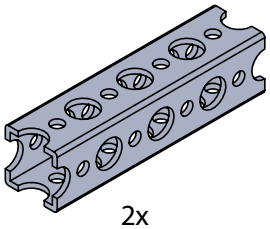
2x
90 mm Wheel & Tire 40222



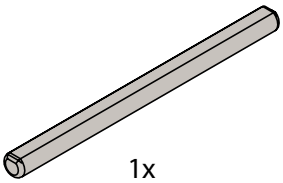
1x
Beam End
Connector 40214



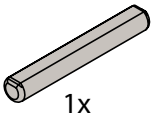
1x
Beam Attachment
Hub 40228



2x
4-Hole Beam 40201

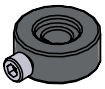


1x
80 mm Axle 40225



1x
40 mm Axle 40226

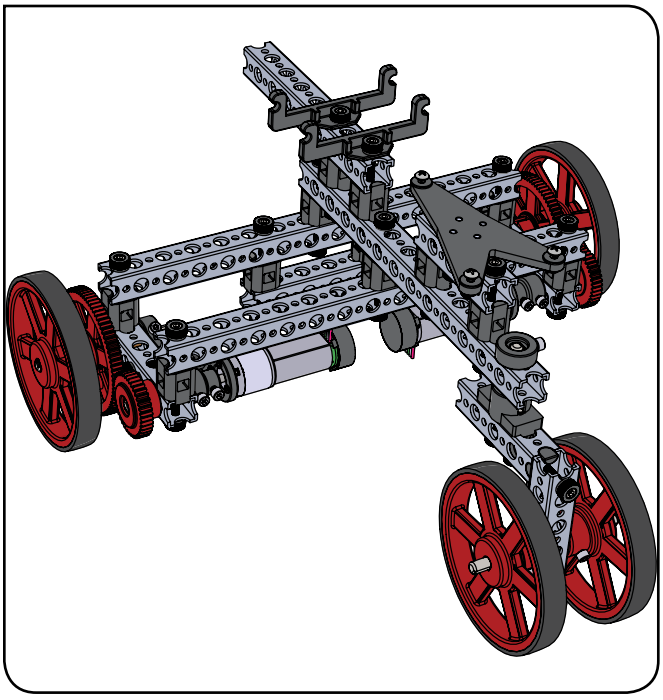
4x
Bronze Bushing
40227



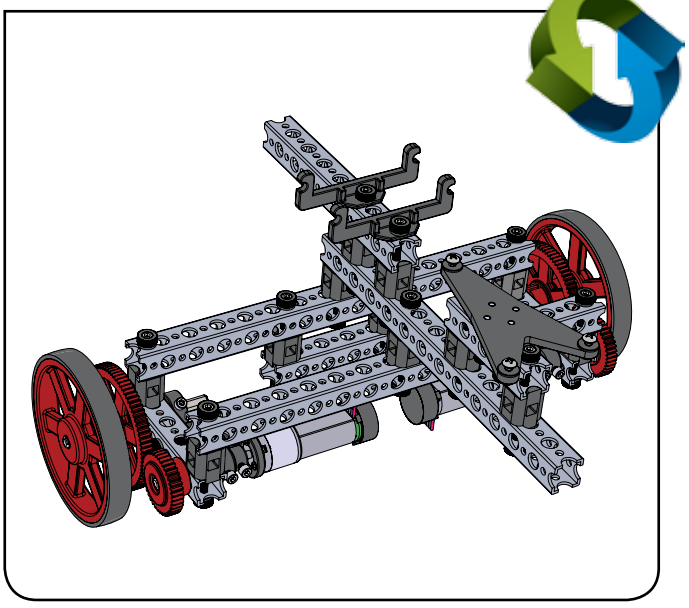
1x
Shaft Collar 40229

3x
Thumbscrew 40323

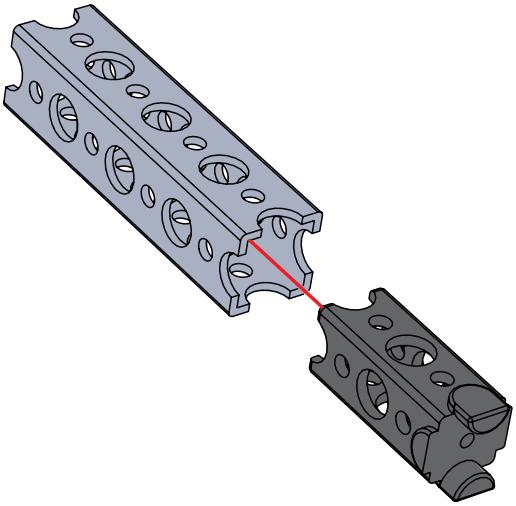
Partial assembly should look like this.



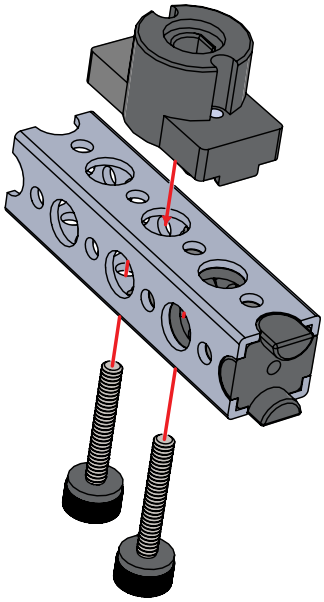
Rotate build to match this view.



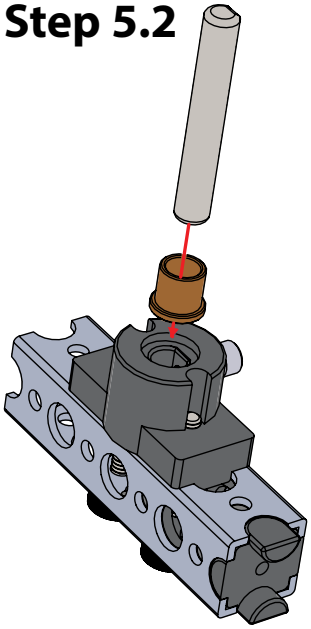
Step 5.0



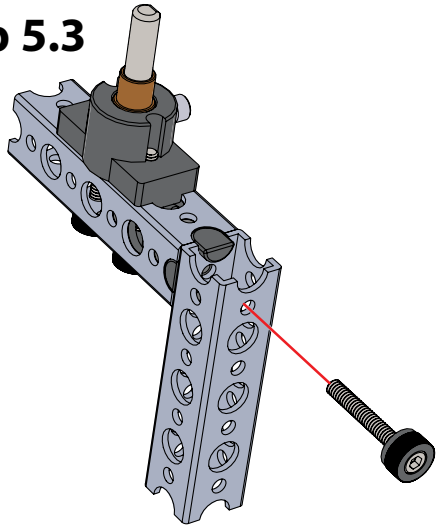
Step 5.1



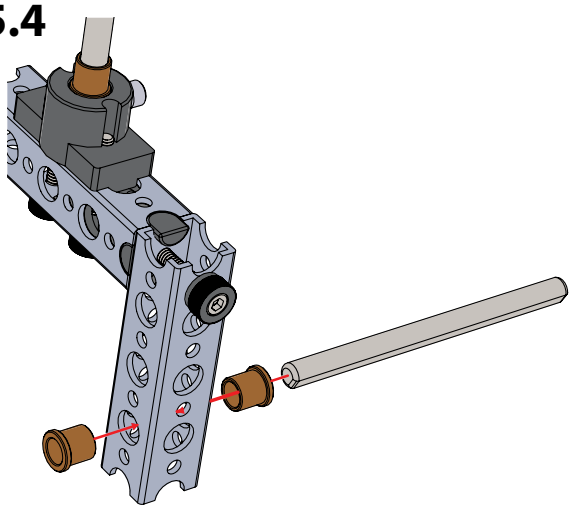
Step 5.2



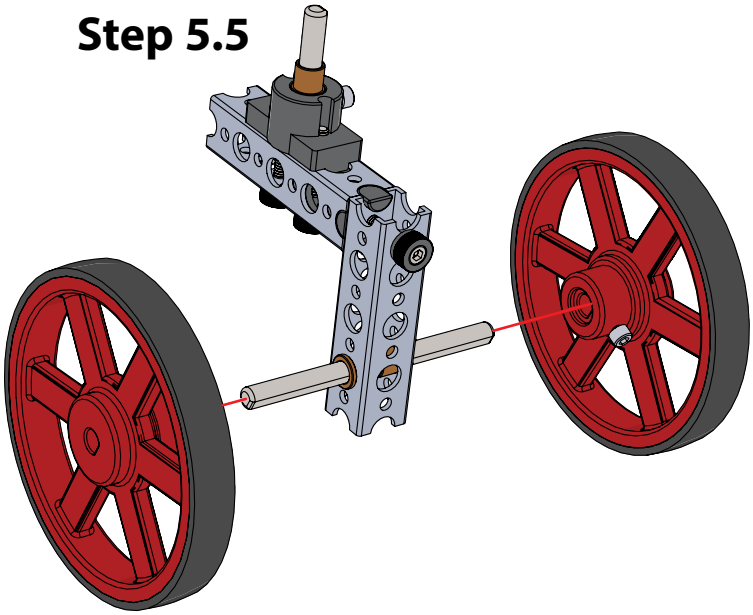
Step 5.3



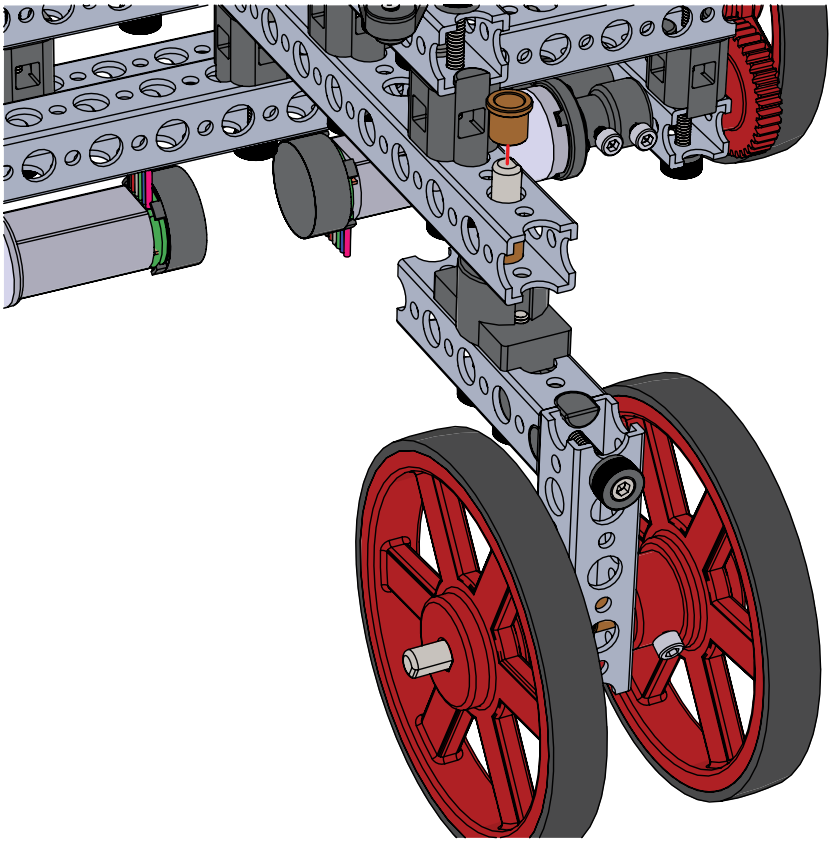
Step 5.4



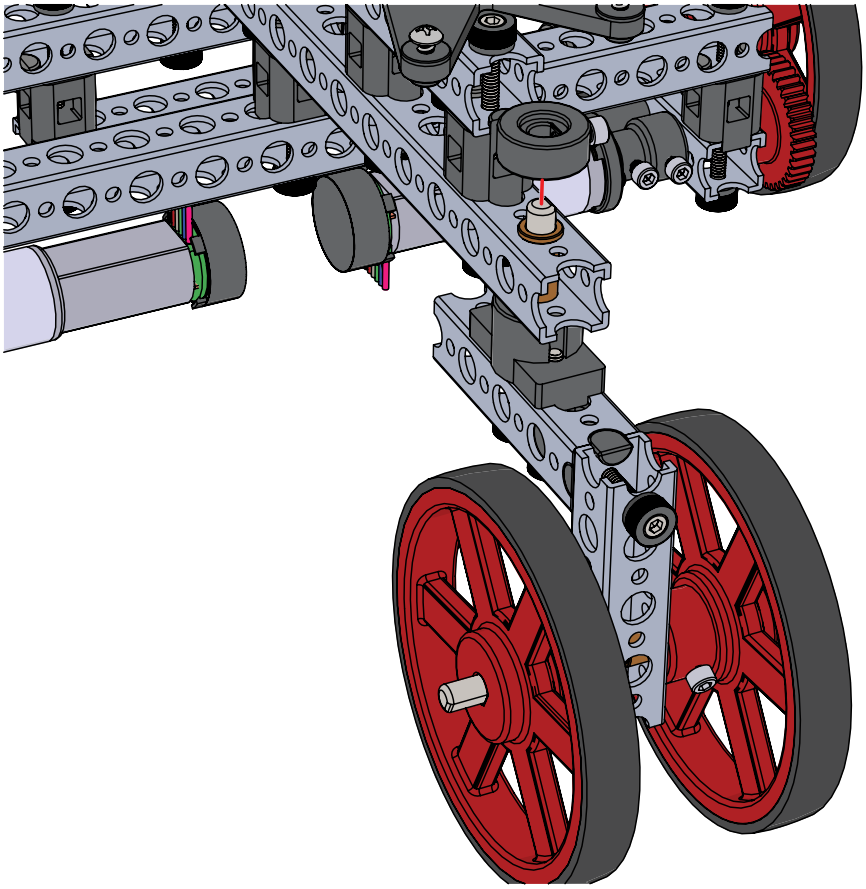
Step 5.5



Step 5.6

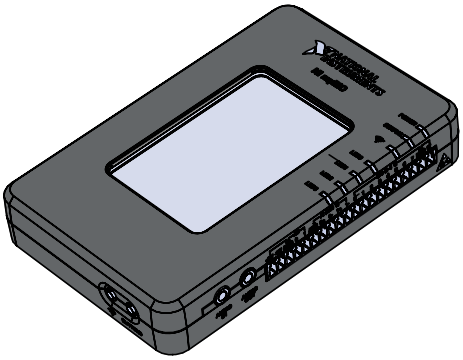


Step 5.7

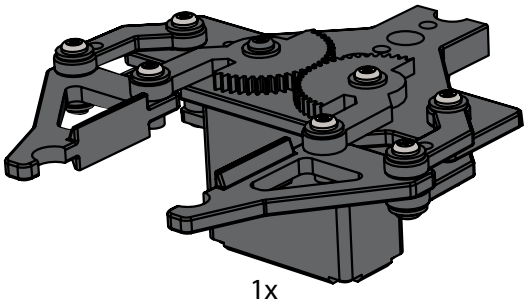


Step 6

Parts Needed



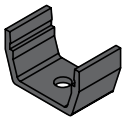
1x
myRIO Controller



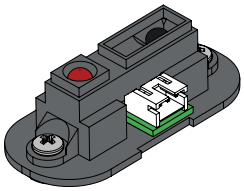
1x
Gripper Kit 40234

2x
Thumbscrew 40323

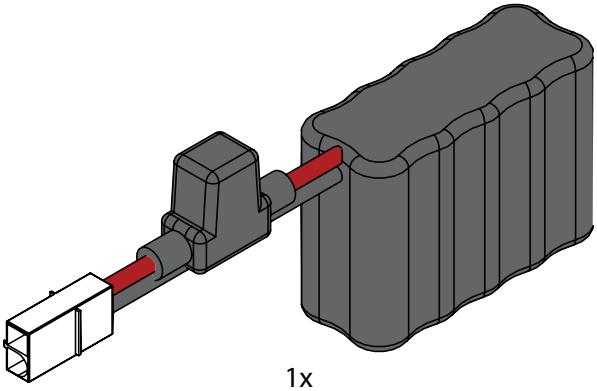
2x
Wing Nut 40221



1x
Digilent Sensor
Mounting Clip 41148

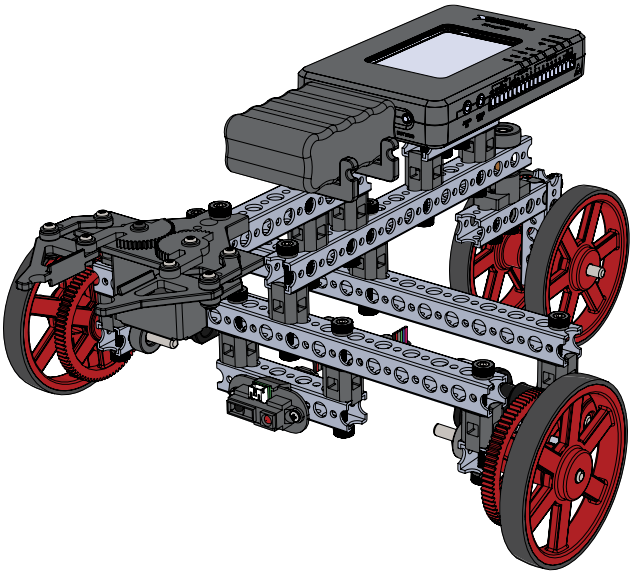


1x
Digilent Sensor
IR Rangefinder with
Mounting Plate

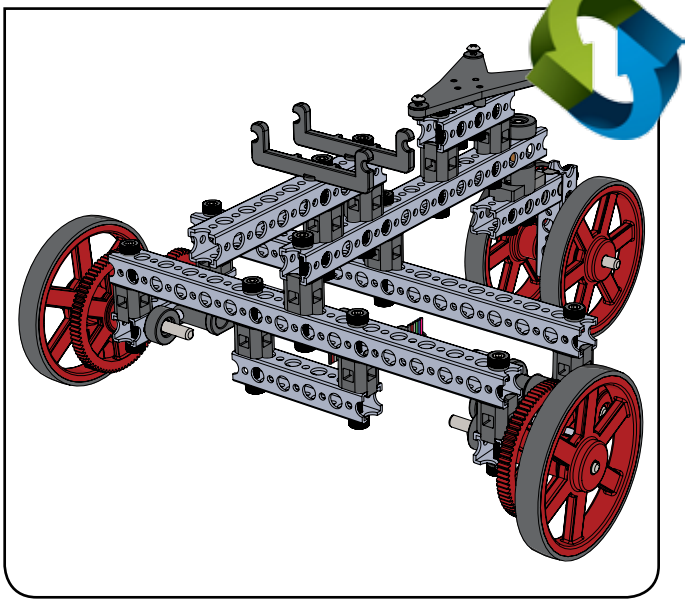


1x
10-Cell AA NiMH
Battery Pack 41135

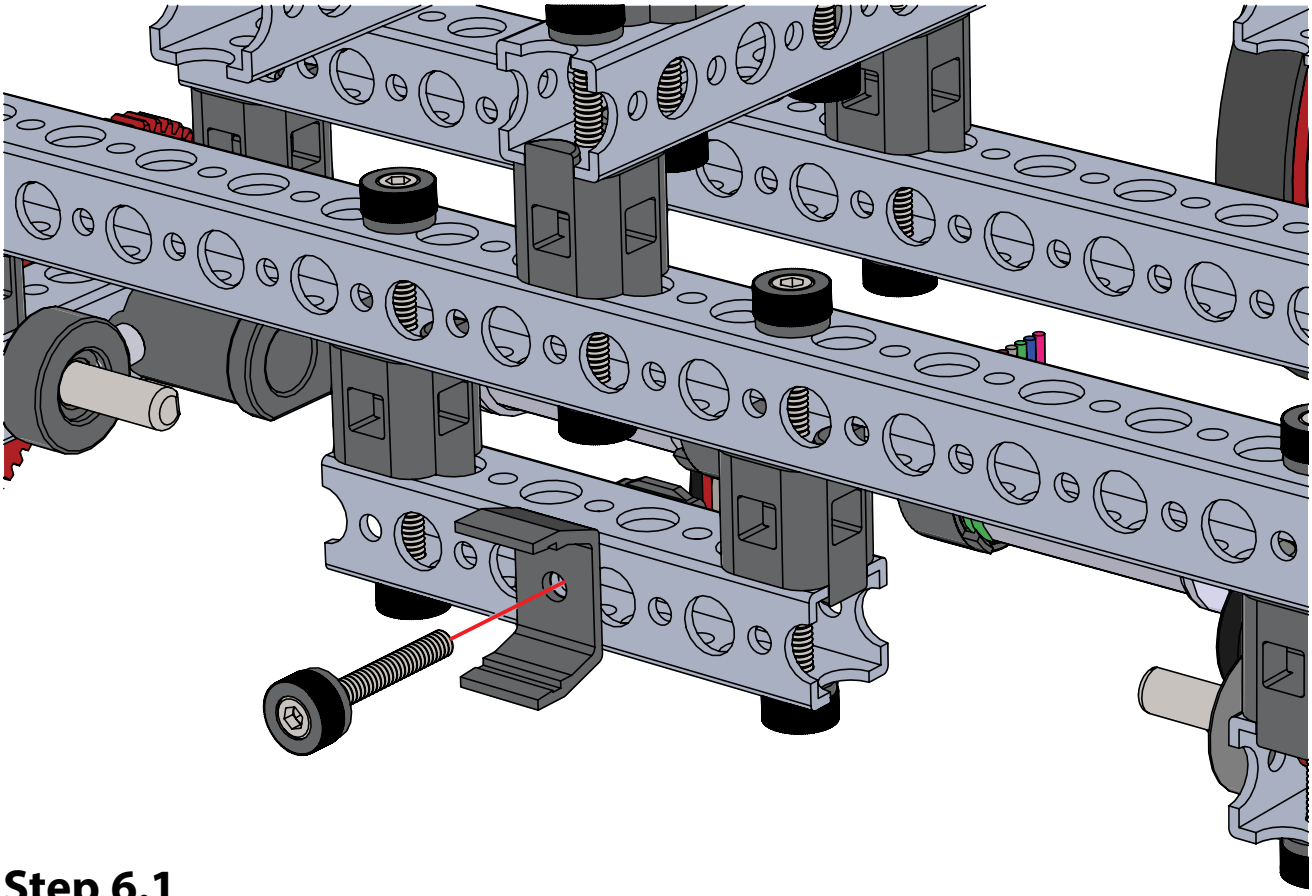
Finished assembly should look like this.



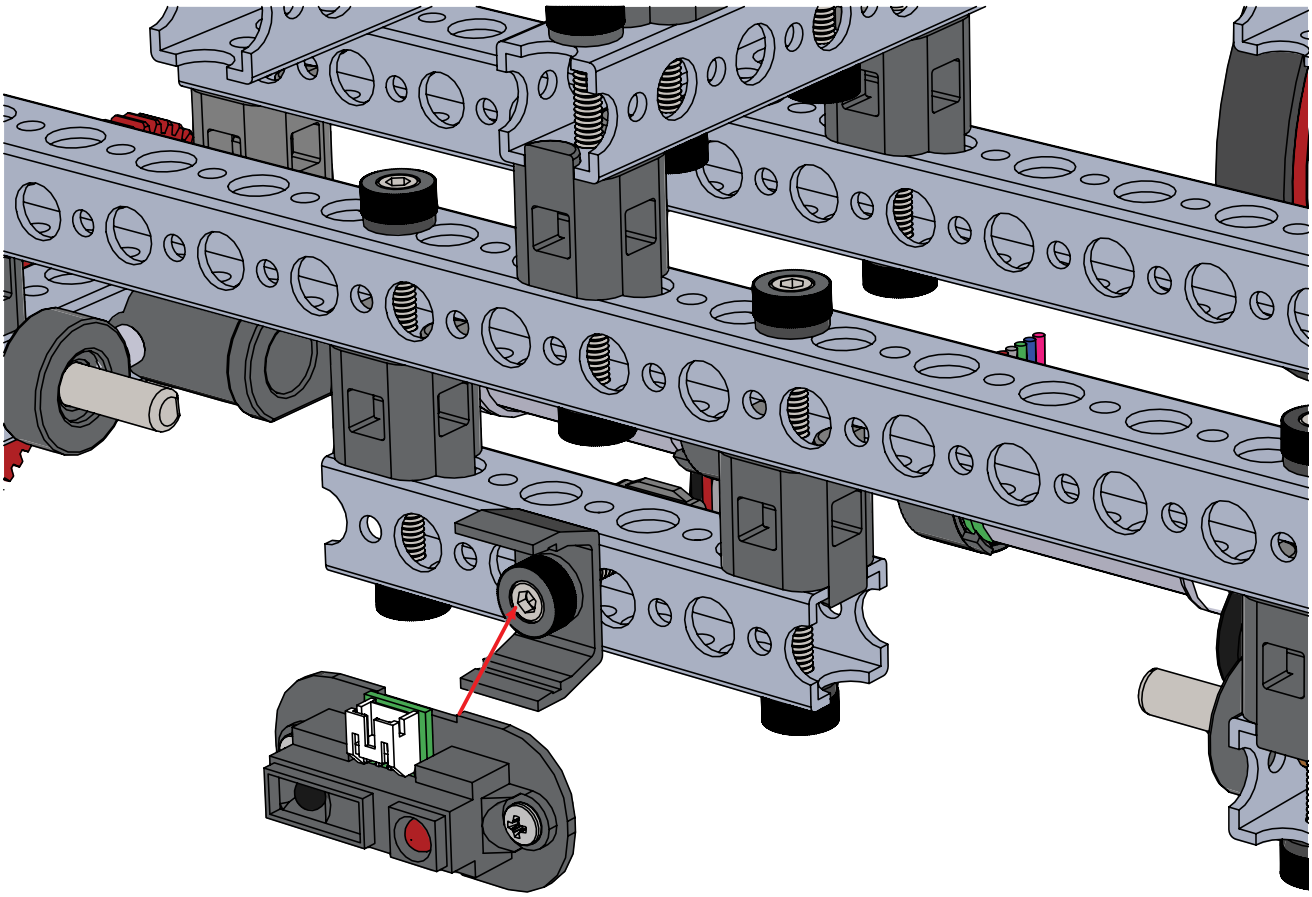
Rotate build to match this view.



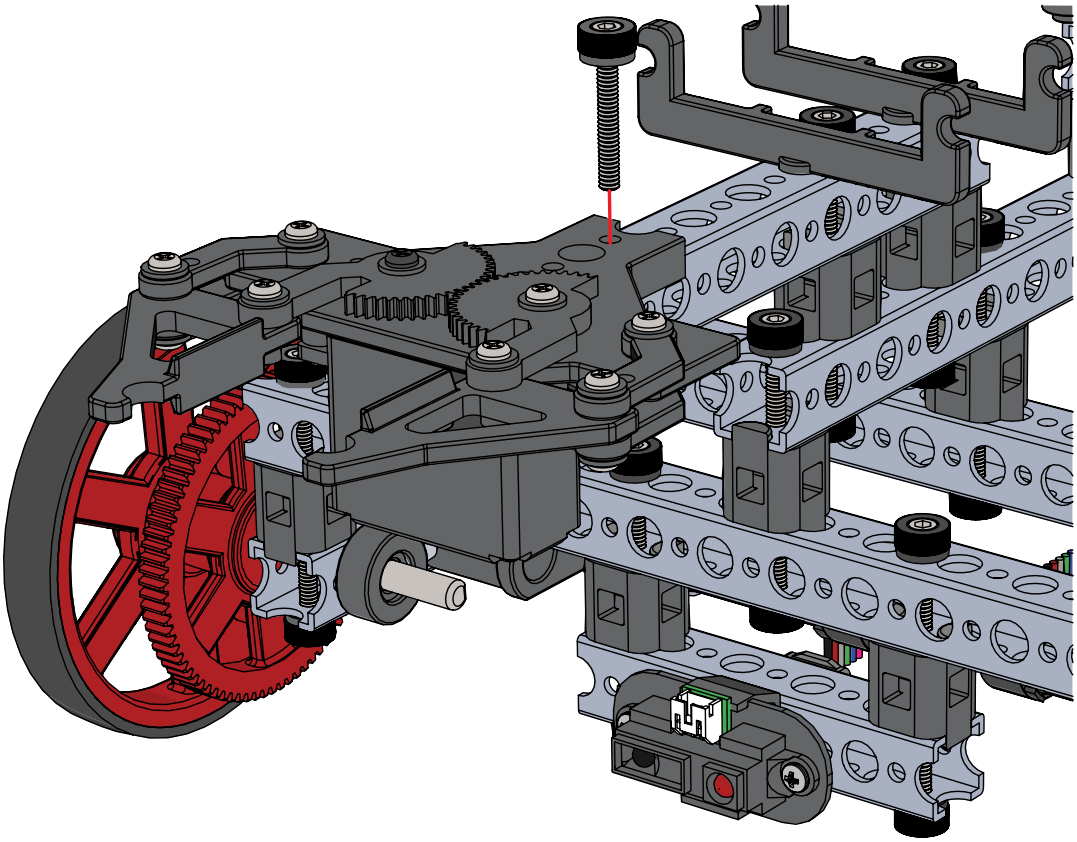
Step 6.0



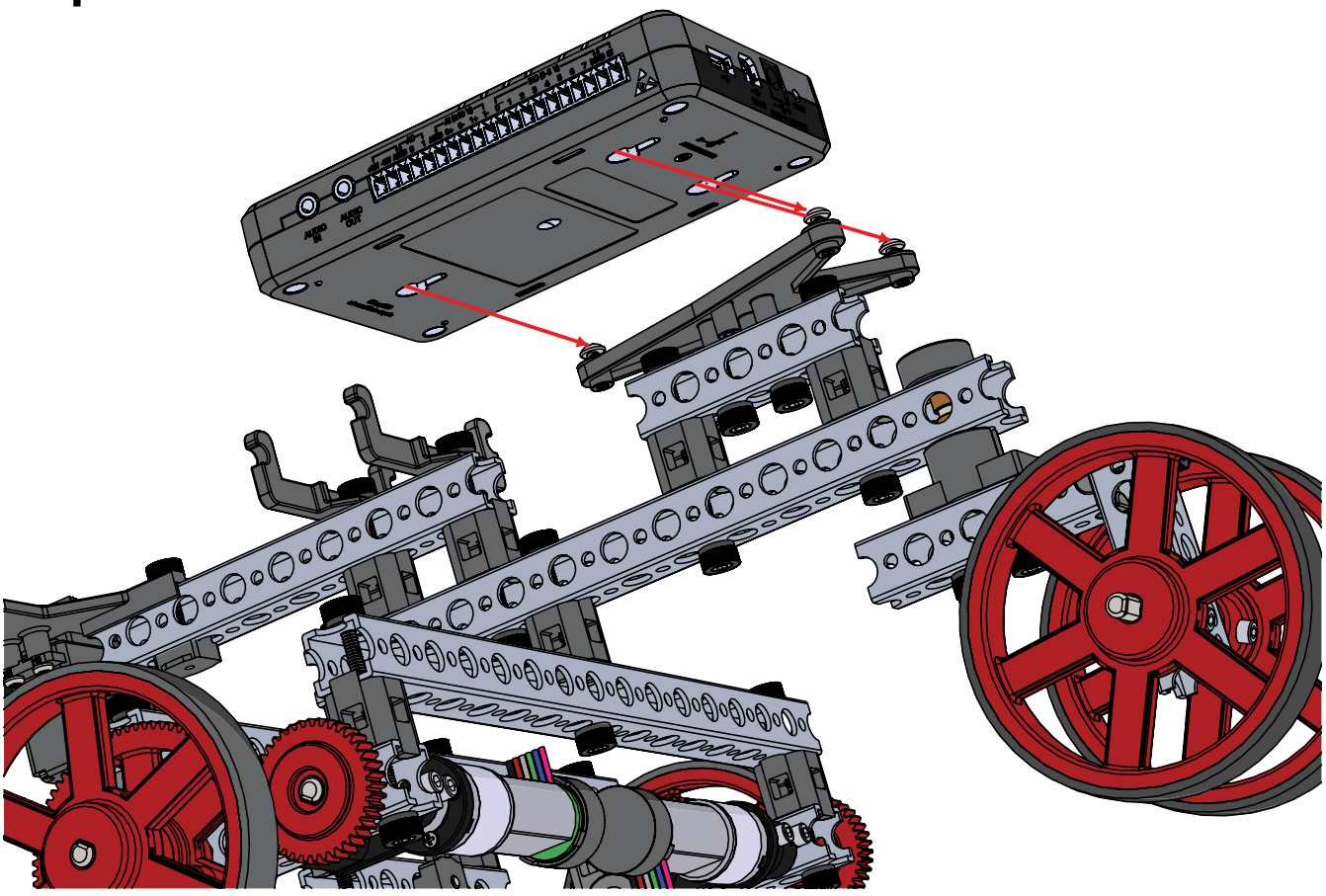
Step 6.1



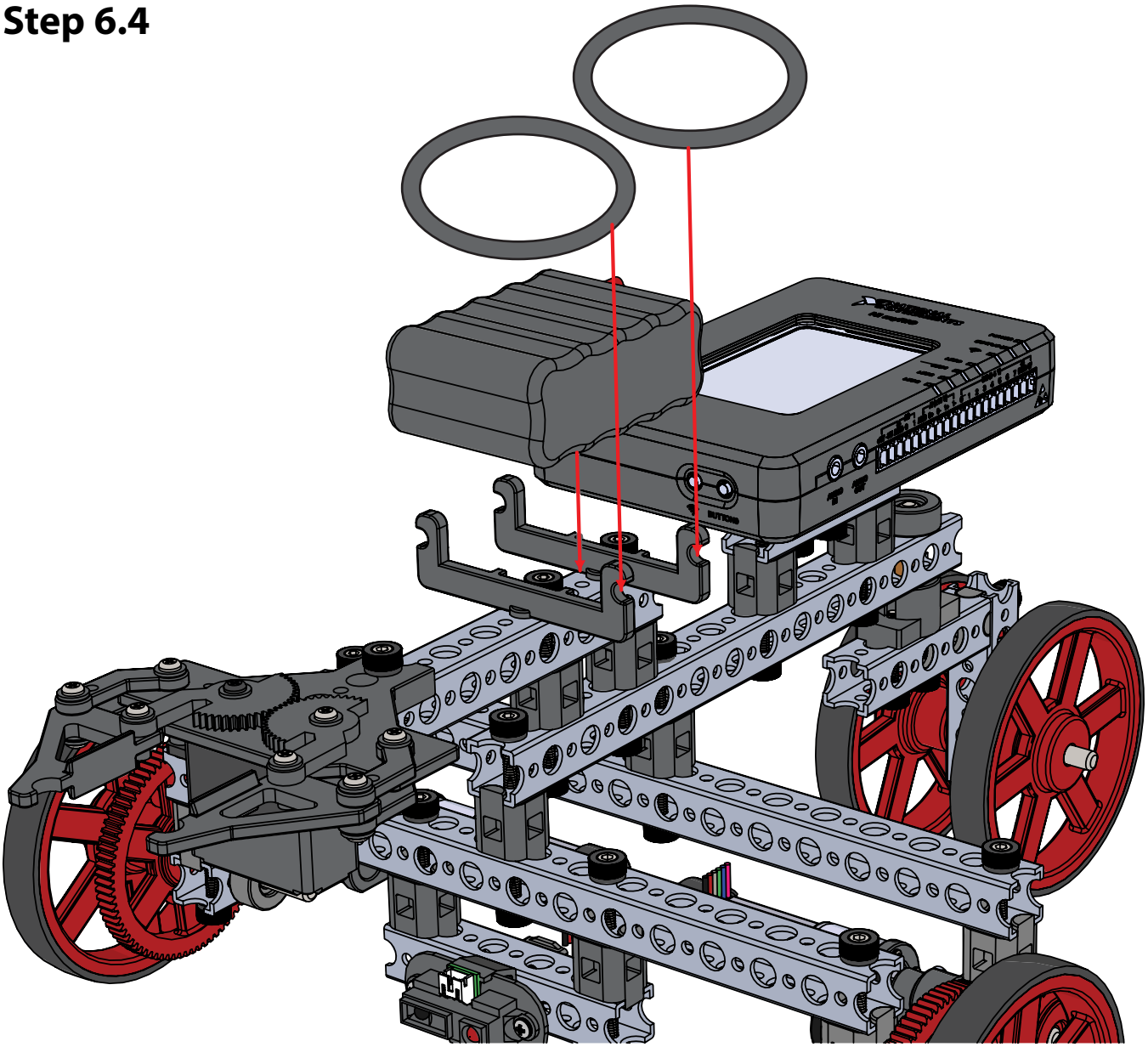
Step 6.2



Step 6.3

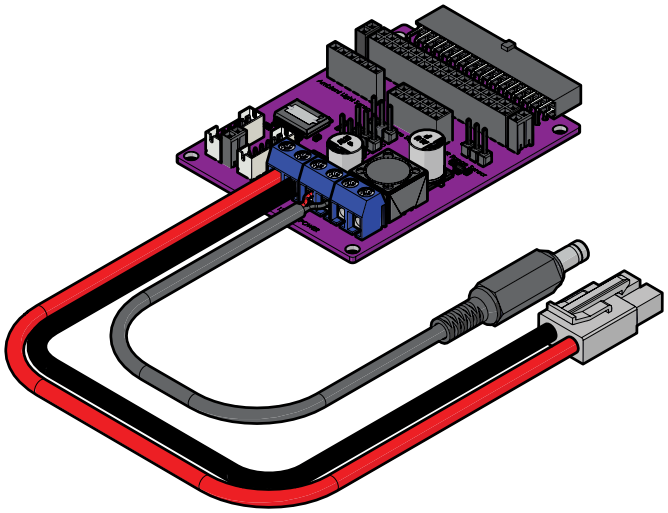


Step 6.4



Step 7

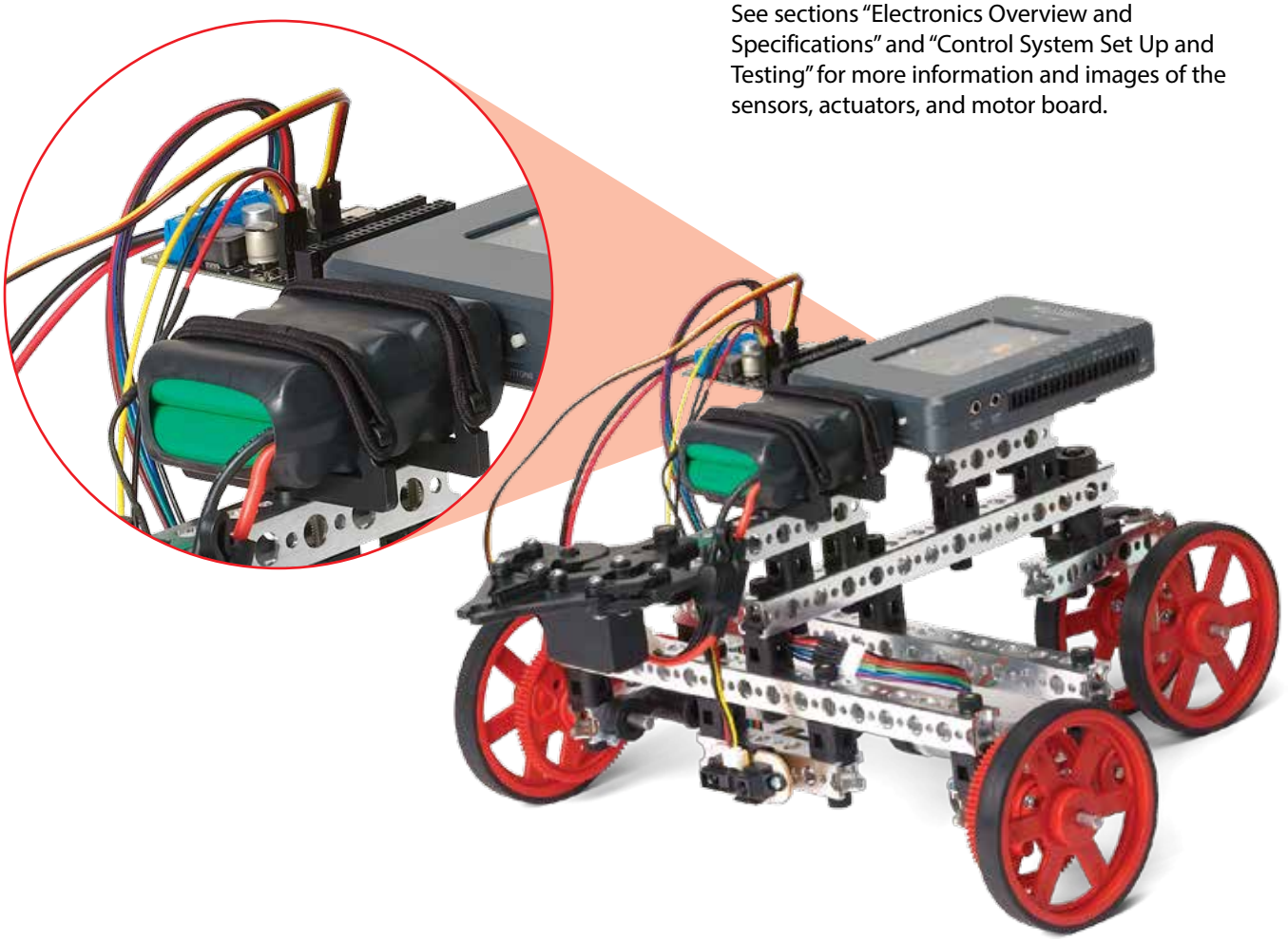
Parts Needed



1x
myRIO Motor Controller Board 41136
with DC Power Plug and Battery Cable attached

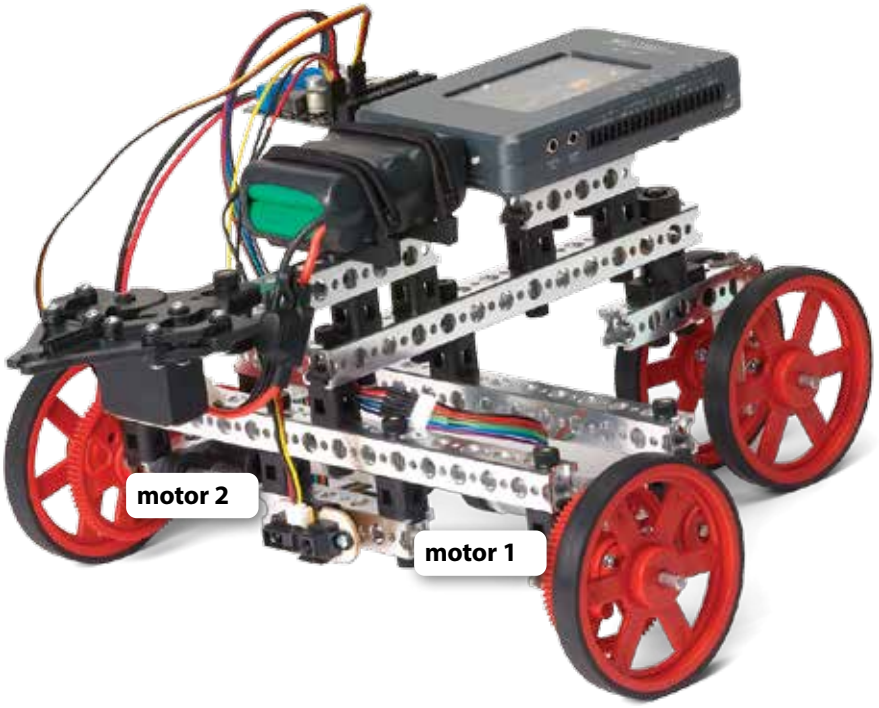
Finished Assembly

See sections "Electronics Overview and Specifications" and "Control System Set Up and Testing" for more information and images of the sensors, actuators, and motor board.



Step 7.1 Connect the motor board to myRIO MXP B

Step 7.2 Connect the Motor 1 and Motor 2 cables (note image below) to the motor board ports labeled"Motor 1" and "Motor 2"



Step 7.3 Connect the IR Sensor cable to the motor board port labeled "IR Range Sensor." Note that the cable should be oriented so that the yellow cable is next to the label "Sig" and the black cable is next to the label "-"

Step 7.4 Connect the servo cable to the motor board port labeled "Servo 2." Note that the cable should be oriented so that the red wire is lined up with the "+" and the black wire is lined up with the "-"

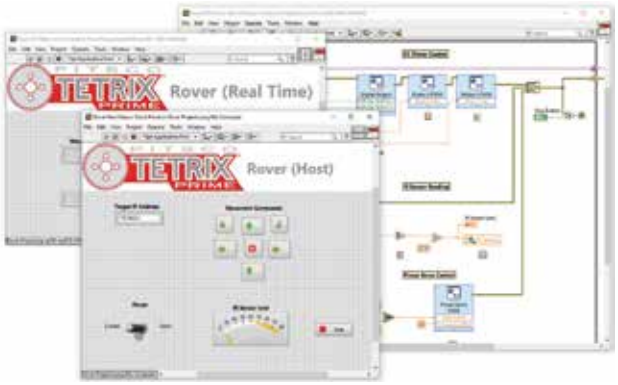
Step 7.5 Use the battery cable to connect from the T1 port of the motor board (red cable to +, black cable to -) to the battery cord

Step 7.6 Use the DC power cable to connect from the T2 port of the motor board (red cable to +, black cable to -) to the myRIO power port

See the Setup/Hookup Tips and Getting Started with myRIO sections for more details and images of sensor, motor, and battery connections.

Step 8

Set up myRIO and run LabVIEW code



8.1. Follow the hardware and software setup instructions in the "Set Up Your Control System" section. Make sure you:

- Connect the motor board to MXP B and connect the sensors and actuators to the motor board
- Connect the system to power
- Install software to your PC (via disk or web installer) and myRIO (via the Getting Started Wizard)
- Configure myRIO for WiFi

8.2. Connect to the myRIO WiFi signal (remember the SSID you chose) using the network icon on the bottom right corner of your screen, similar to how you connect to any other WiFi network

8.3. Navigate to Rover code using the following path: C:\Users\...\Desktop\National Instruments\Pitsco Tetrix Prime for NI myRIO\Rover Vehicle and open "Rover Project"

8.4. Make sure the IP Address in the parenthesis next to the "NI-myRIO" line item matches the myRIO wireless adapter IPv4 address in the Network Adapters window of the myRIO Configuration menu. If it doesn't: Right click the NI myRIO line item, and select "Properties" Update the IP address and click OK

8.5. Right click the NI myRIO line item and select "Connect."

8.6. Open "Rover RT Main.vi" by double clicking the line item in the project and press the run arrow in the upper left corner or press ctrl + r to run

8.7. Open "Rover Host Main.vi" under My Computer

8.8. Make sure that the IP address on the Front Panel matches the myRIO wireless IP Address

8.9. Press the run arrow in the upper left corner or press ctrl + r to run

8.10. Use the buttons to move the rover and the switch to open and close the pincer. View the meter to detect objects near the rover

Step 9: Troubleshooting

9.1. If you are having trouble sending messages from your host computer to myRIO, make sure that the IP address on the Front Panel of the Host VI matches the myRIO wireless IP address

9.2. If the rover drives in the wrong direction, make sure that the motors are connected as described in step 7.1.

Next Steps

- Learn about how the Rover Vehicle relates to control theory in the next section
- Get some ideas for expanding the Rover on the first page of this section

Connections to Control Concepts: Rover Vehicle

Prof. R. H. Bishop

University of South Florida

The rover vehicle assembly provides insight into two feedback control system concepts: (i) openloop control to a reference input and (ii) closed-loop feedback control to track a predetermined path.

A. Demonstrating open-loop control

Plant	Rover vehicle
Sensor	None
Actuator	DC servo motors
Performance	Command following
Design objectives	Tune the control system by adjusting PID gain constants
Reference inputs	External commands to turn right, left, back, forward

Open-loop control utilizes an actuator and a controller to command the plant directly without using feedback. In this case, the rover vehicle and two geared, DC motor-driven wheels serve as the plant and actuators, respectively. The commands enter the control system via the host computer through front panel commands sent over the network connection to the myRIO which hosts the motor control code. Students may add PID code to control the plant's position or velocity. The main benefit of openloop control is reduced overall complexity—meaning that the additional challenges associated with introducing sensors and feedback loops are avoided. However, the open-loop control system is highly sensitive to external disturbances, represented by Td(s), and to both knowledge of and changes in the parameters of the plant, represented by G(s). Therefore, open-loop control does not address key design issues in feedback control of disturbance reduction and robustness. For example, when commanded to 'turn right', the open-loop control system may indeed turn the rover, but in the presence of external disturbances, the rover may not turn at an acceptable rate and the final direction may not be acceptable. The open-loop control system performance can be tuned by adjusting the PID controller gains.

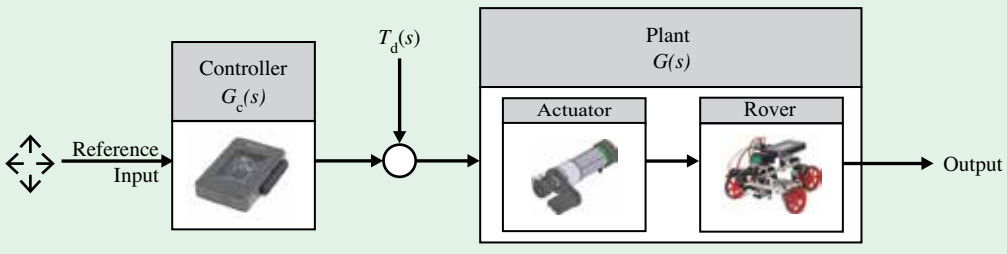


Figure 1 Open loop control with external disturbances.

B. Demonstrating closed-loop control to track a predetermined path

Plant	Rover vehicle
Sensor	Infrared (IR) range finder
Actuator	DC servo motors
Performance	Command following in the presence of disturbances
Design objectives	Tune the control system by adjusting PID gain constants & track the predetermined path
Reference inputs	Predetermined path with obstacles

A closed-loop feedback control system utilizes a negative feedback loop with a sensor to measure the output and compare the measured output, represented by Y(s), with the reference input, represented by R(s), to generate an error signal, represented by Ea(s), that drives the controller. If we can utilize the error signal appropriately, then we will achieve our tracking objective, that is, minimize E(s)=Y(s)-R(s), in the presence of external disturbances and plant uncertainties and parameter changes. This is the key objective of closed-loop feedback controller design. The IR sensor can used to distinguish between colors based on differences in IR reflectivity to track a prescribed path marked on the ground. It can also measure the distance to obstacles allowing us to miss obstacles or track a prescribed path through an obstructed area. With the introduction of the sensor, we have additional unwanted sensor noise, represented by N(s). The main benefits of closed-loop control include (i) increased robustness of the closed-loop performance to changes in the parameters of the plant, (ii) improved external disturbance rejection, measurement noise attenuation, and reduction of the steady-state error of the system, and (iii) ready control and adjustment of the transient response of the system by skillful design of the controller. However, these advantages come with cost. The main cost of closed-loop feedback control is additional complexity that means higher monetary costs and greater likelihood of component failures. Since the benefits far outweigh the disadvantages, we find closed-loop feedback control is widely employed in modern control systems. The key to closed-loop feedback control is the use of the tracking error signal to improve the transient response (settling time, percent overshoot, etc.) as well as reduce steady-state errors tracking errors.

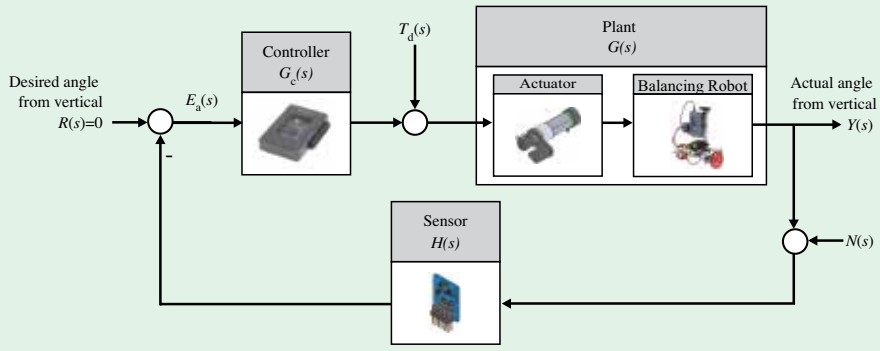


Figure 2 Closed-loop control of the rover with IR sensor feedback.

Note: Control concepts described in detail in Modern Control Systems, by R. C. Dorf & R. H. Bishop, 13th Ed., Pearson Education, Inc., 2017.

Balancing Arm

Overview

The TETRIX Prime Balancing Arm is an assembly that demonstrates how control concepts taught in engineering classes can be applied. The arm itself is rotated by a servo motor, and balances a ball in a position specified by the user (i.e. setpoint). The servo is controlled by a closed-loop PID (proportional-integralderivative) algorithm implemented in LabVIEW. The feedback data is ball position, which is collected from an infrared (IR) sensor. If the ball is out of position, the difference between the set point and position data from the IR sensor (i.e. error) will be calculated, and the PID loop will correct it over time.

Base Functionality

- The balance arm is rotated by a servo motor, which receives PWM position data
- The position of the ball is measured by an infrared (IR) sensor, which inputs data via an analog line
- The user specifies a position setpoint on the front panel of the LabVIEW VI
- The system gradually moves the ball to the position using PID control
- After building the assembly the user must calibrate the arm by indicating the servo position at which the arm is parallel to the ground
- Before every use, the user must also calibrate the controller to recognize the minimum and maximum edges of the arm
- The PID gains are auto-tuned, requiring no input by the user

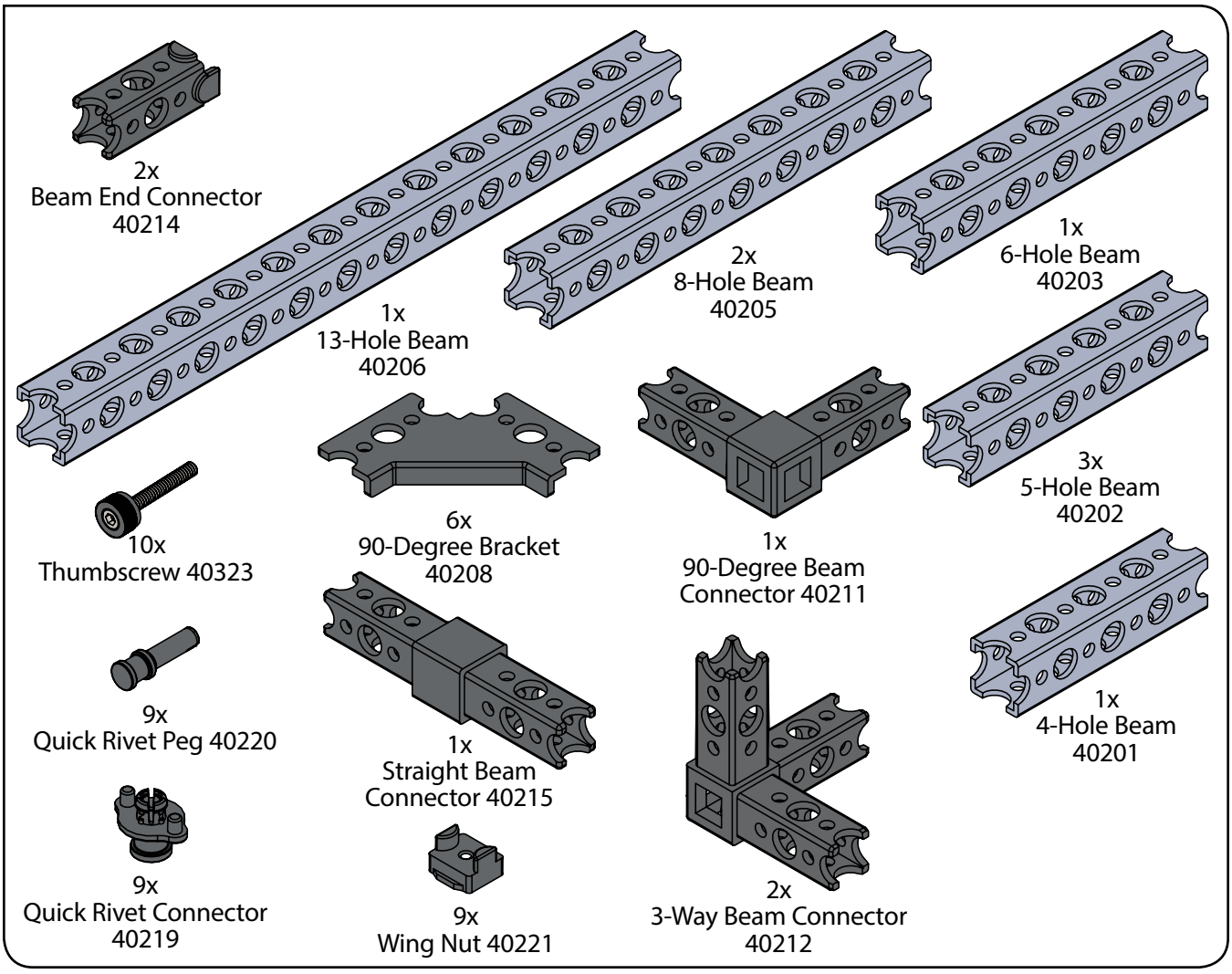
Expansion and Teaching Options

- Learn how the Balance Arm relates to controls concepts like closed-loop stability, disturbance rejection and performance, and PID controller design (See the Connections to Controls Concepts at the end of this section)
- Program the balance arm to alternate between two different position setpoints

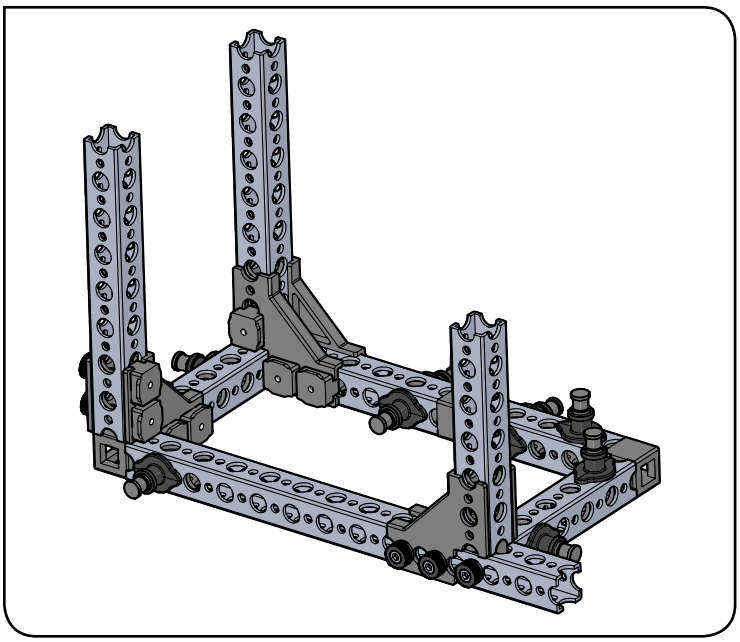
Note: It's recommended that you only balance a standard racquetball on the Balancing Arm due to its size, weight, consistency, and other properties. The control system has also been optimized for use with a racquetball.

Step 1

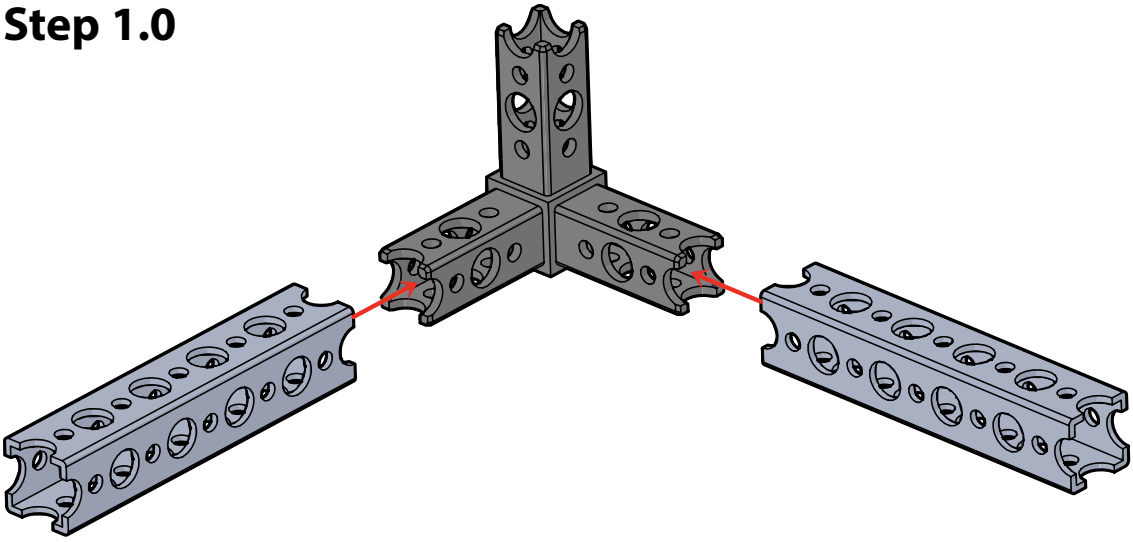
Parts Needed



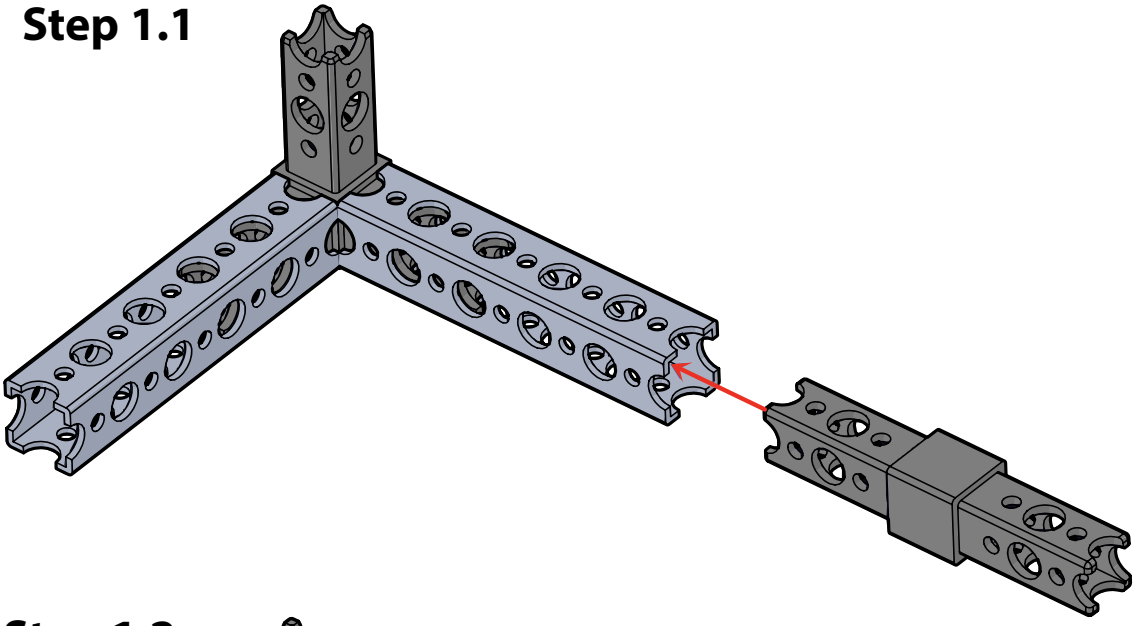
Partial assembly should look like this.



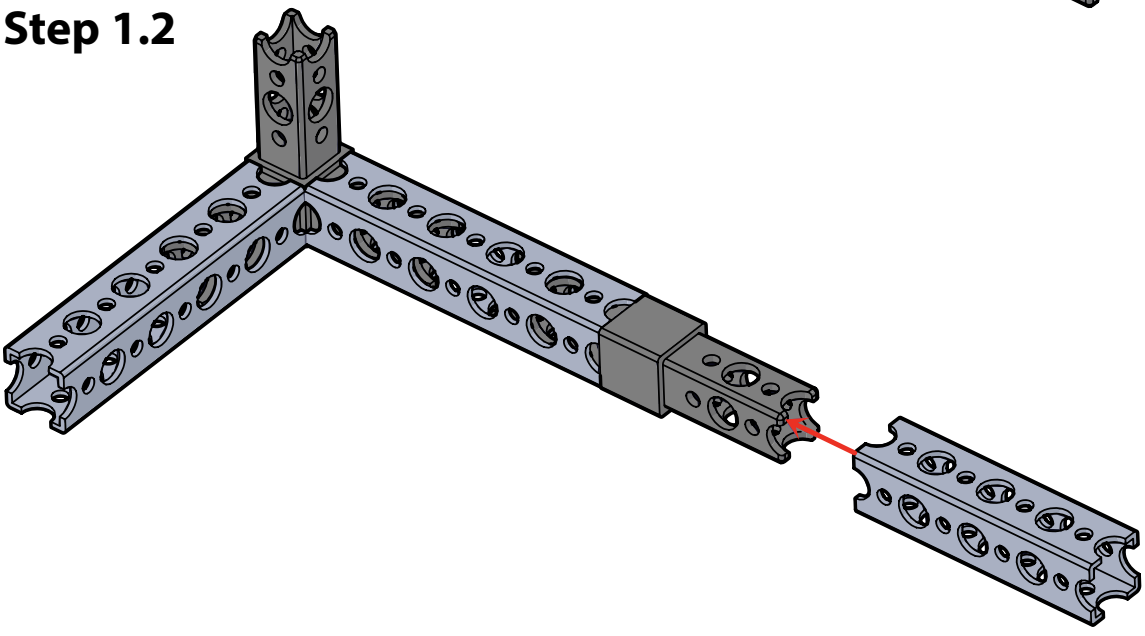
Step 1.0



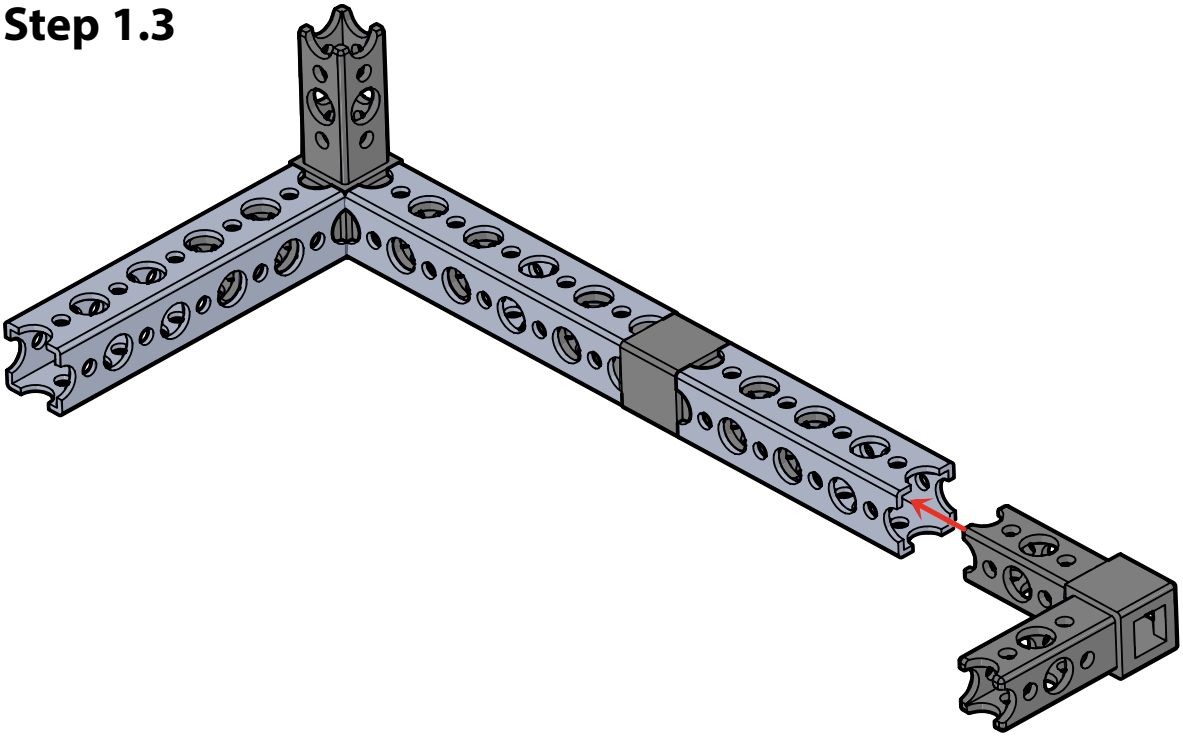
Step 1.1



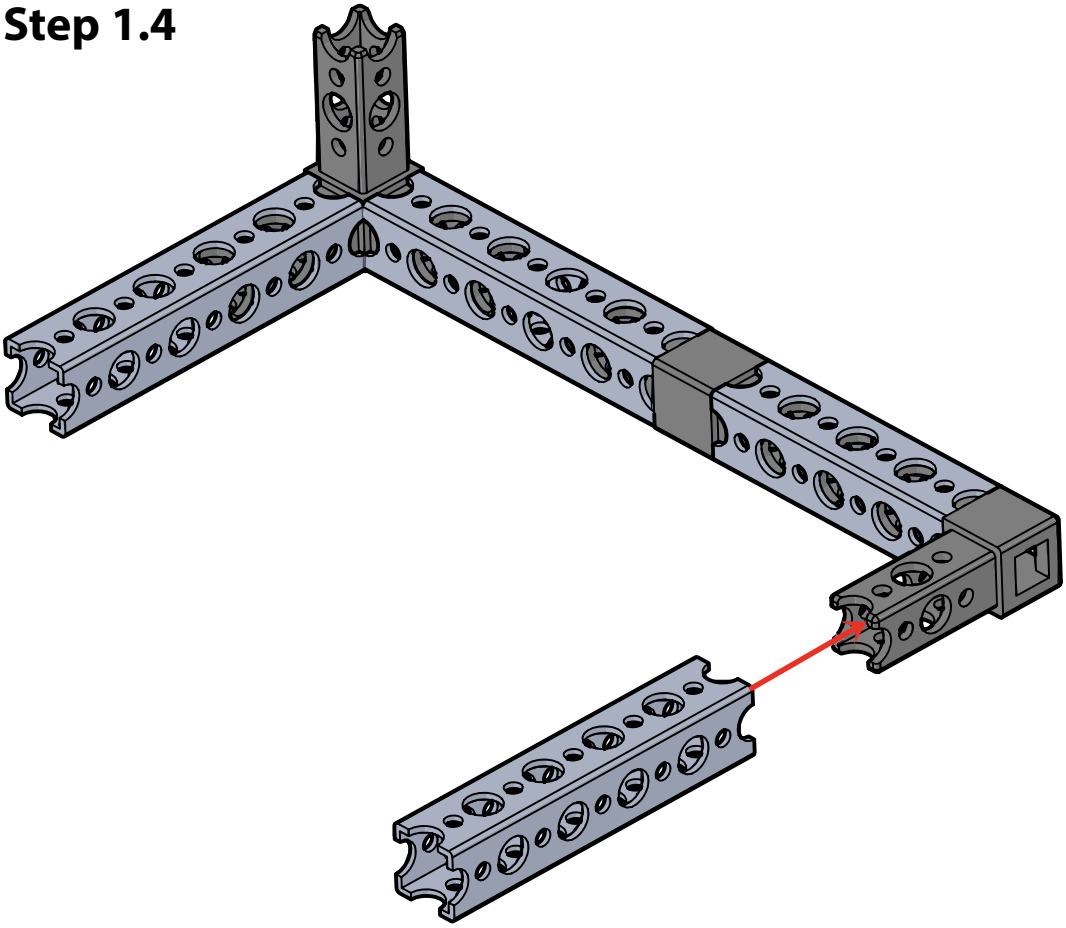
Step 1.2



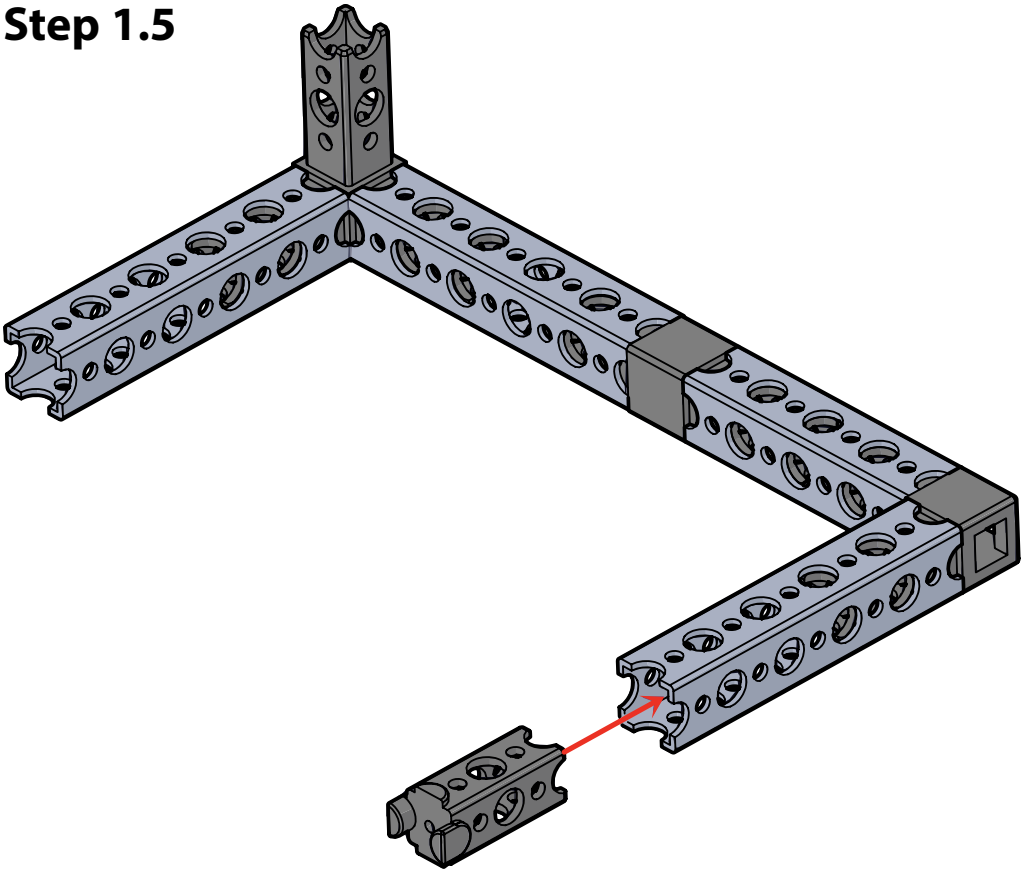
Step 1.3



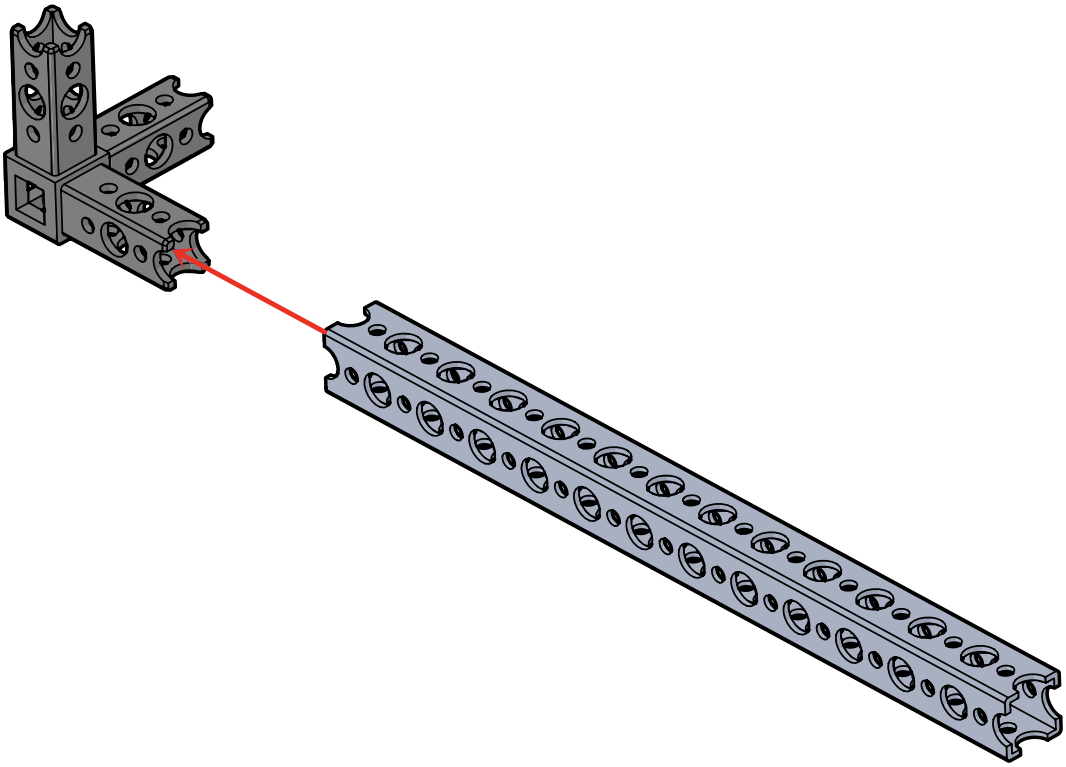
Step 1.4



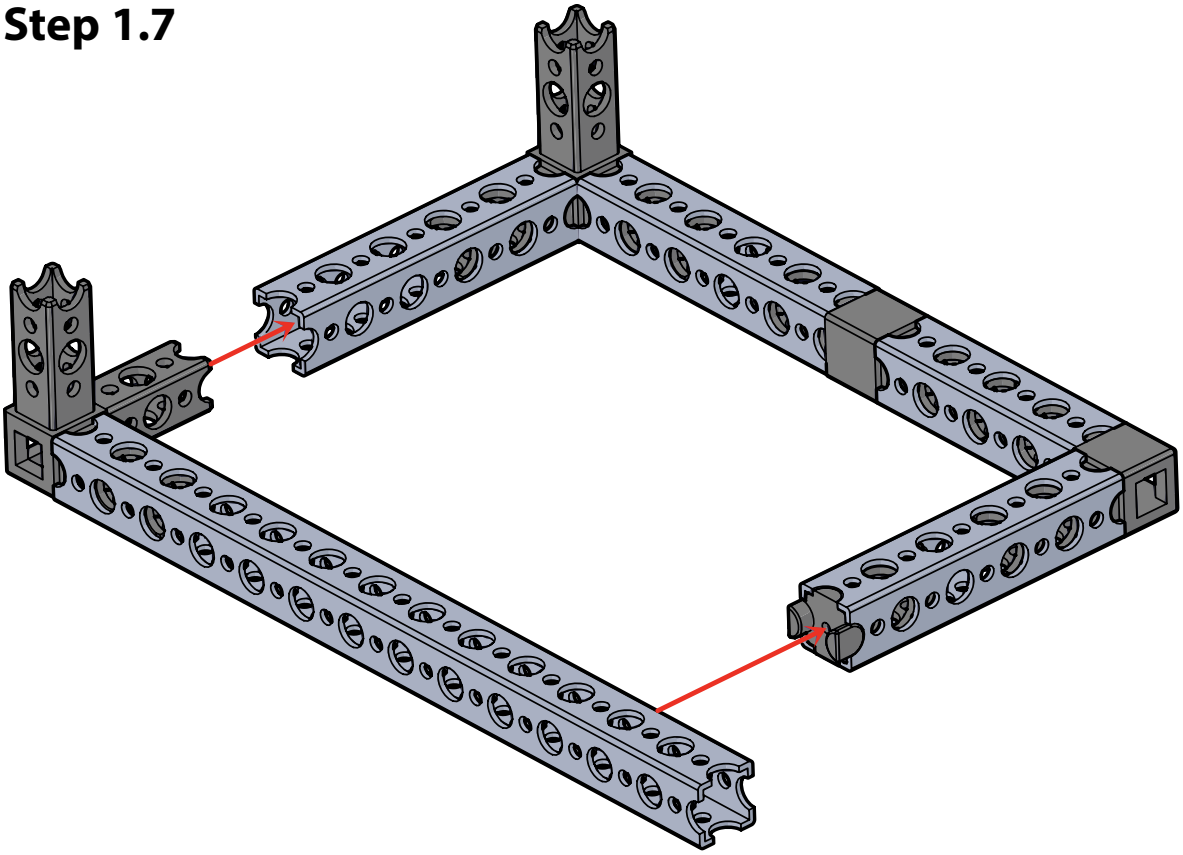
Step 1.5



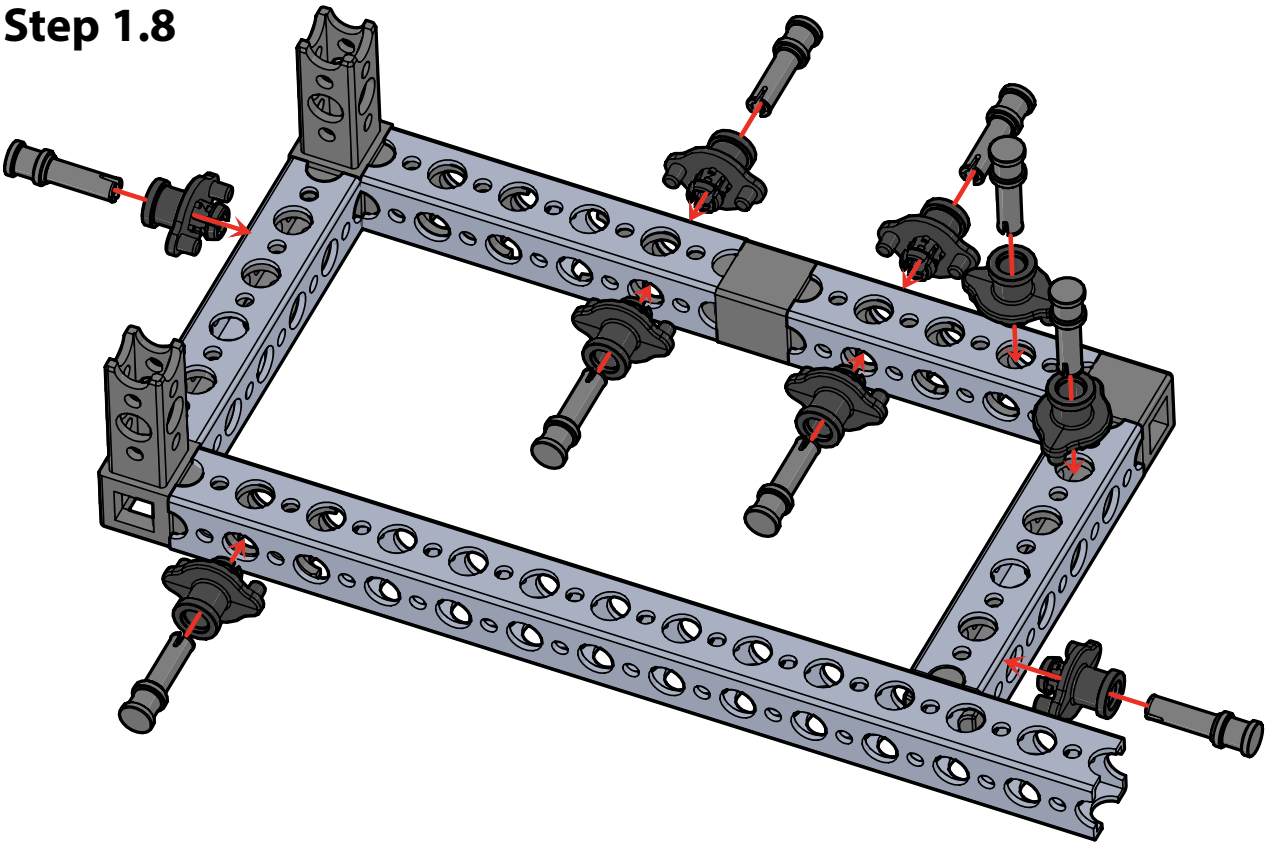
Step 1.6



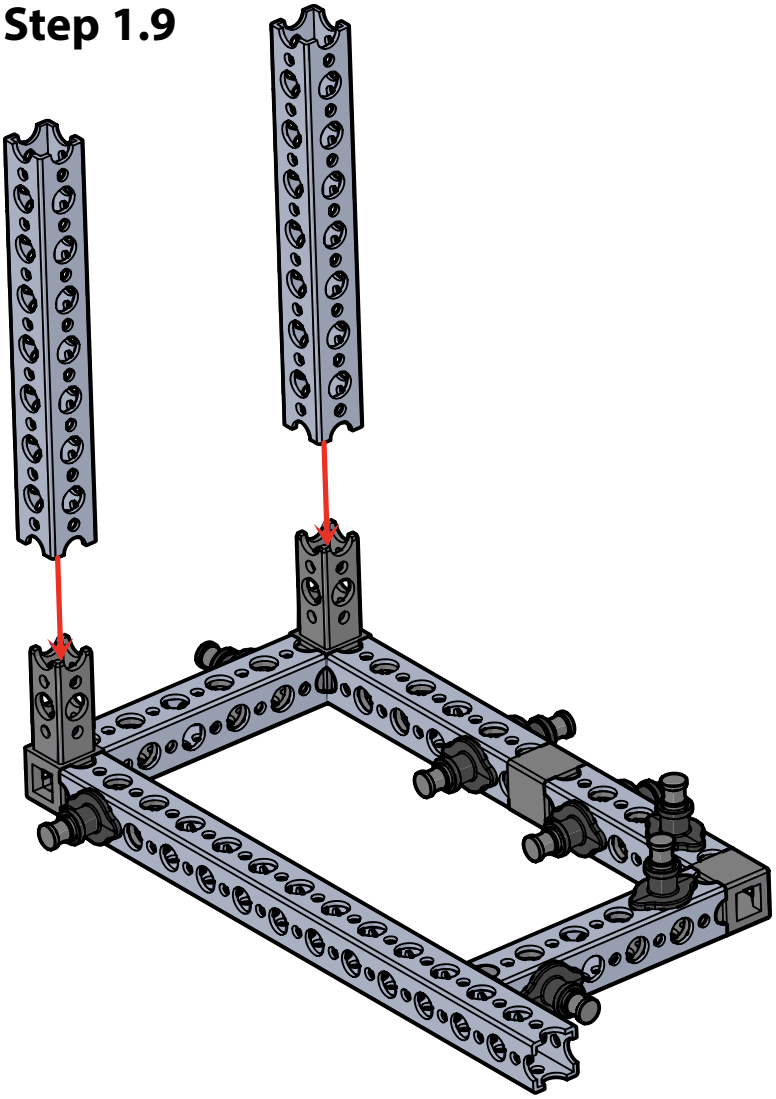
Step 1.7



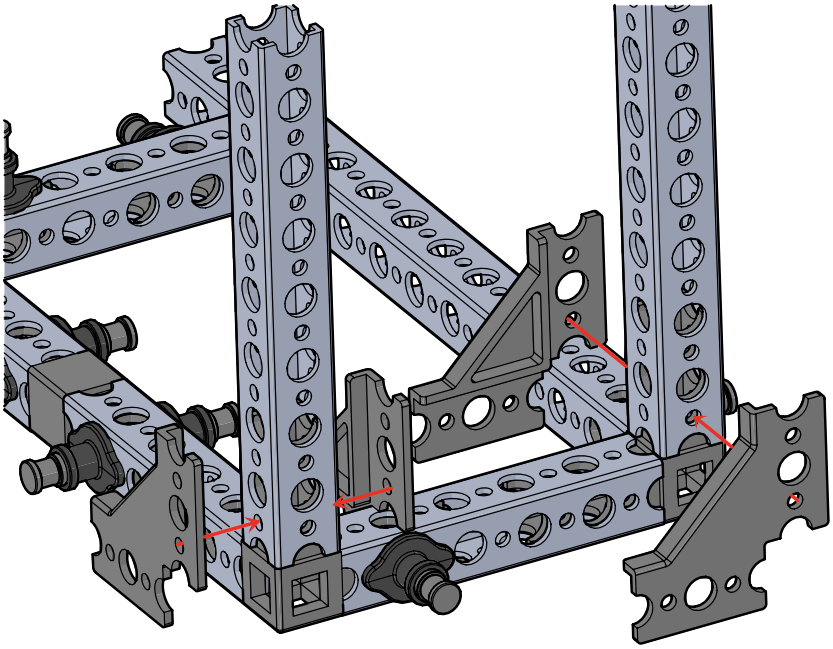
Step 1.8



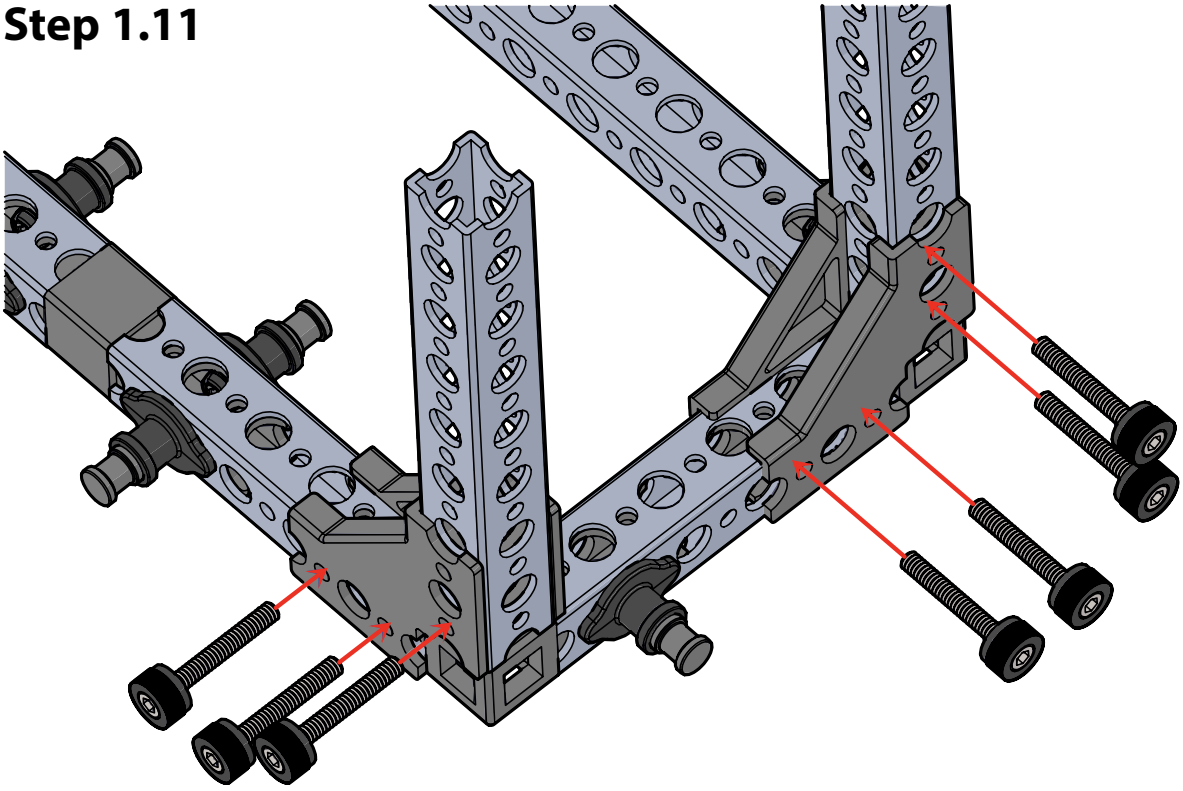
Step 1.9



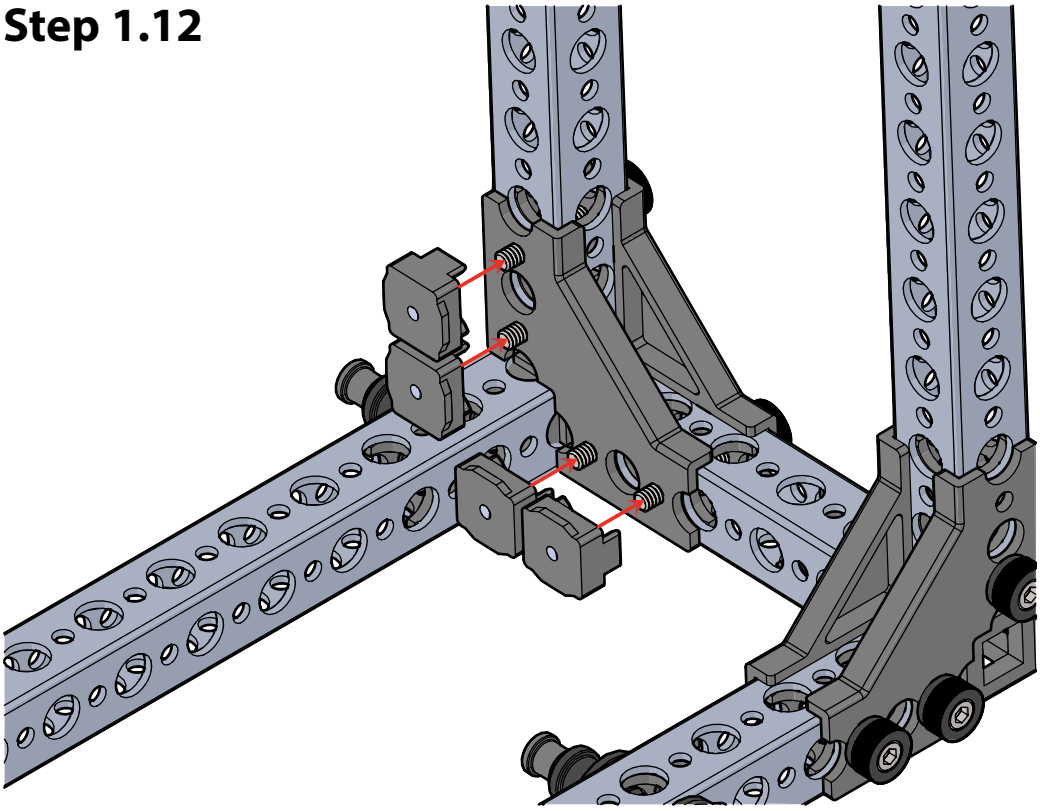
Step 1.10



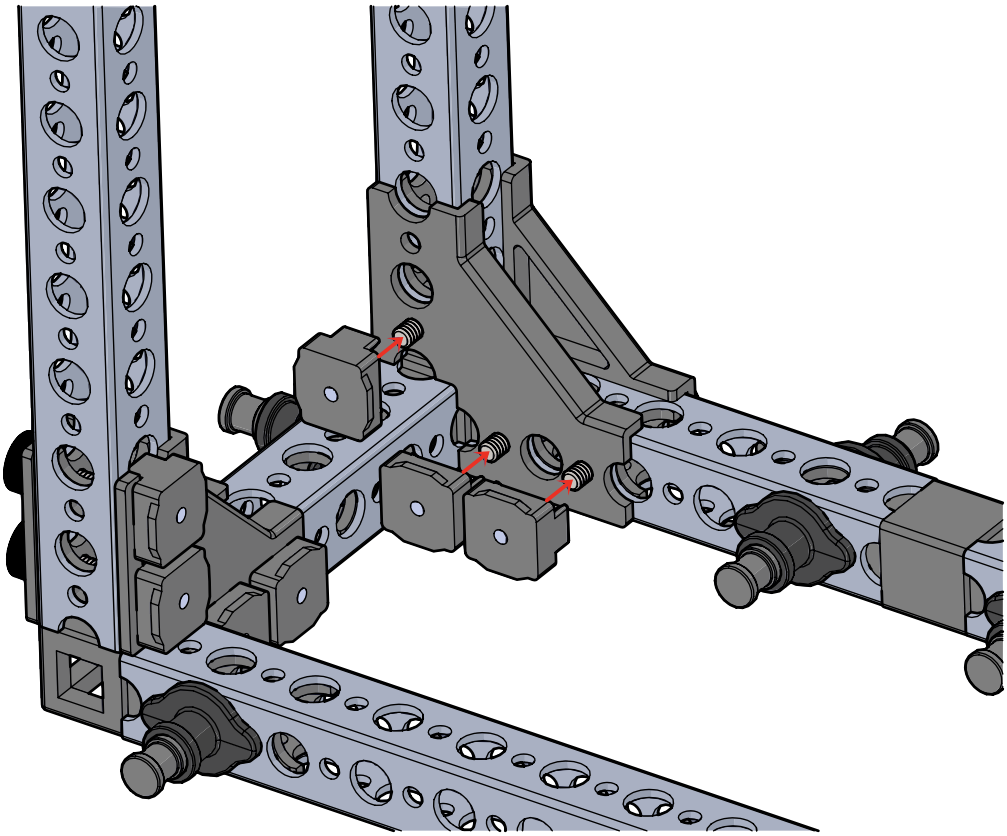
Step 1.11



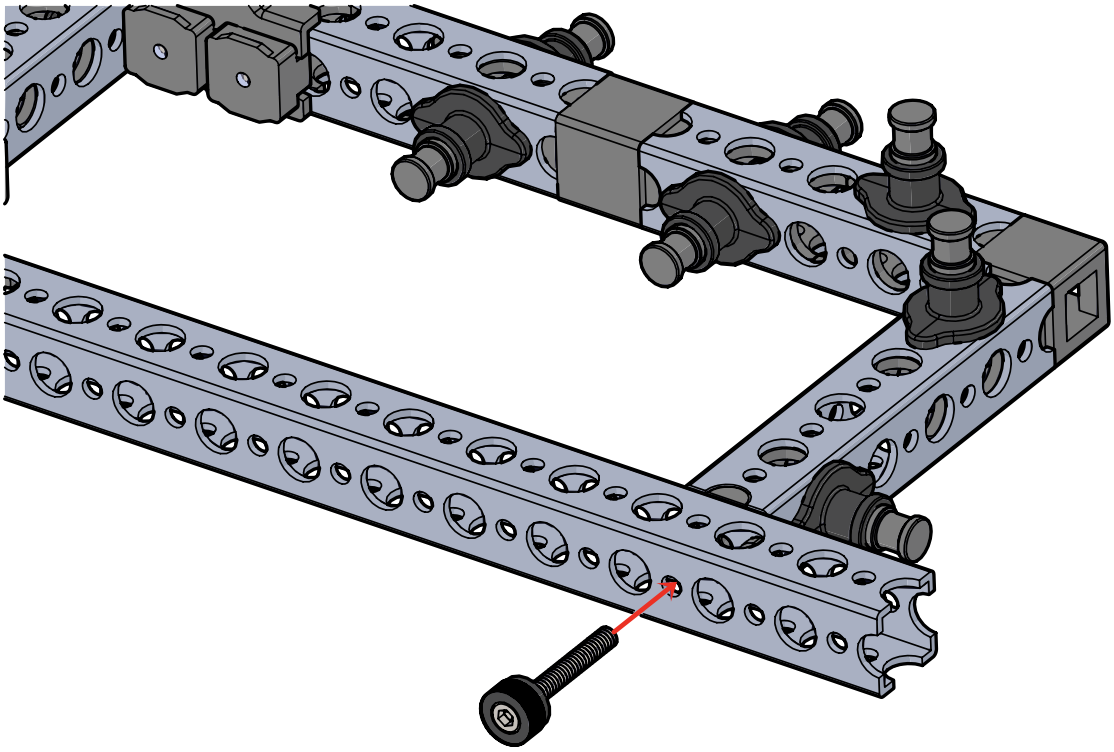
Step 1.12



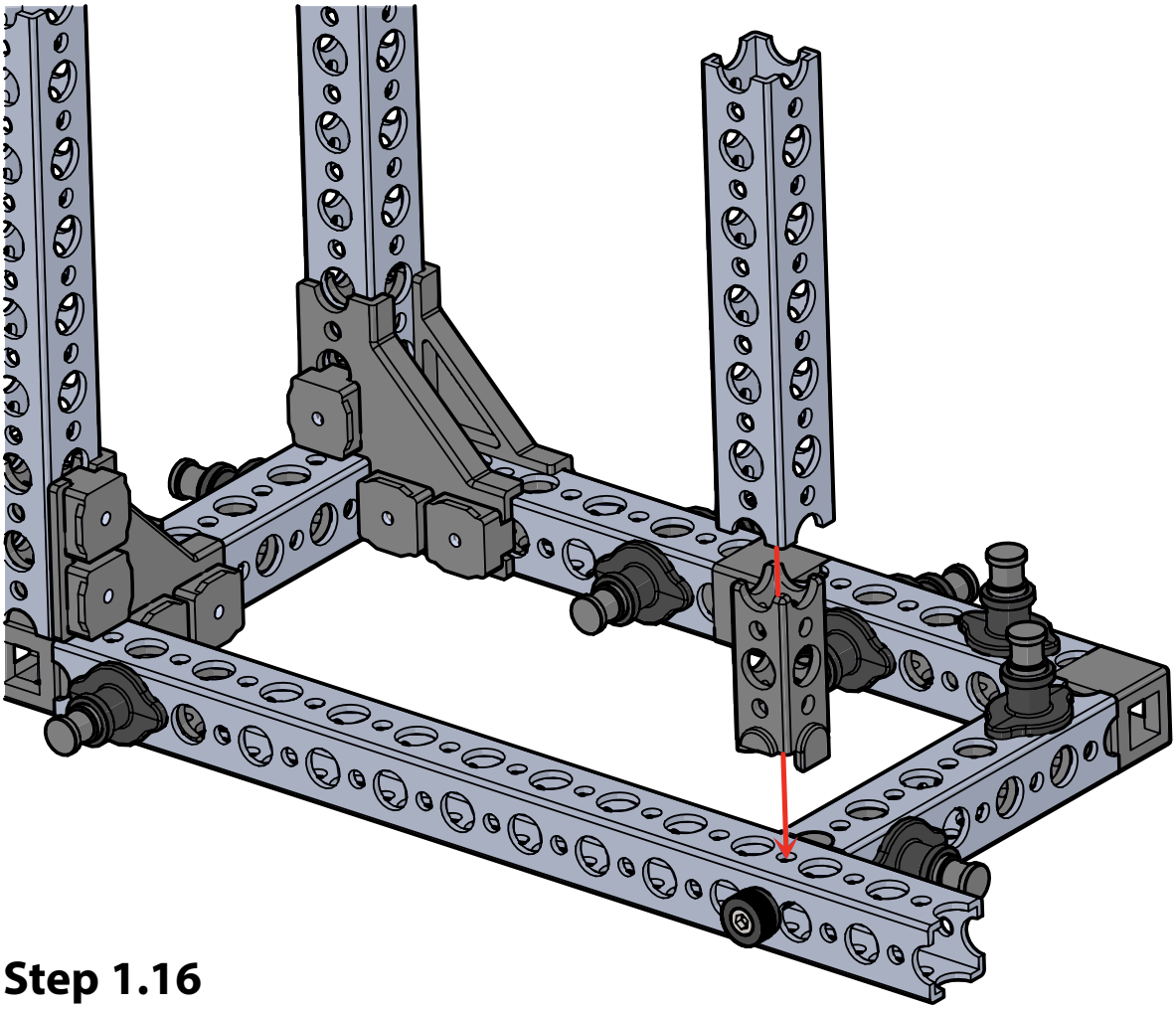
Step 1.13



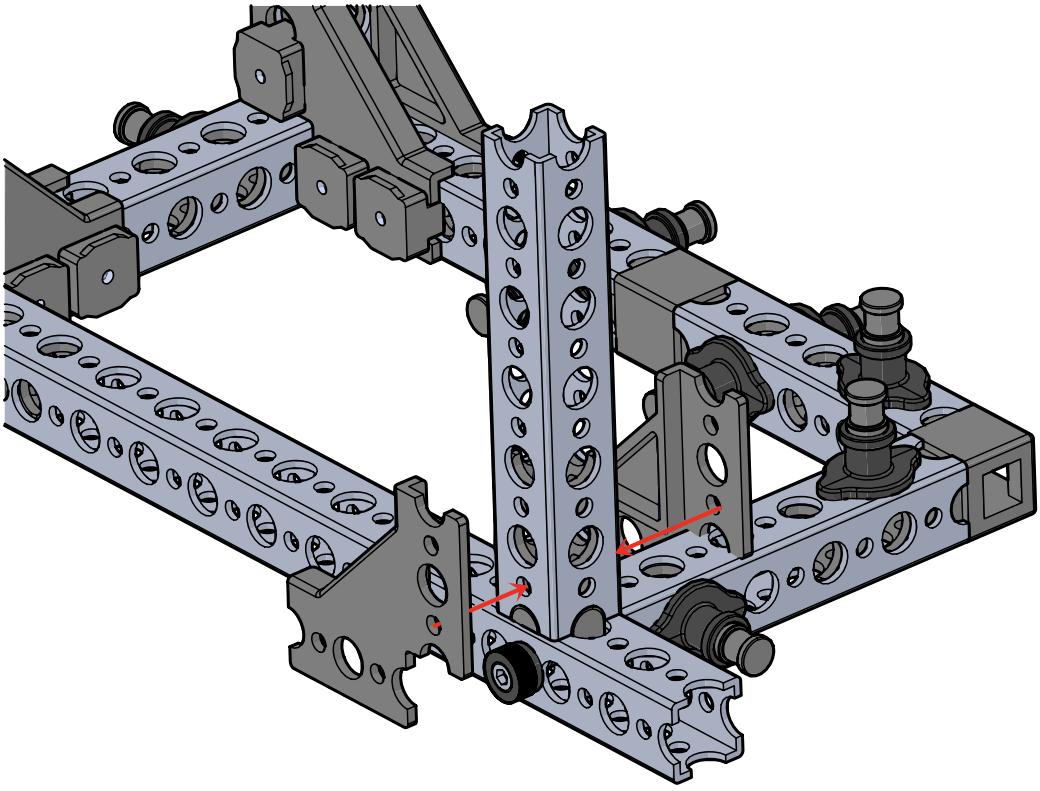
Step 1.14



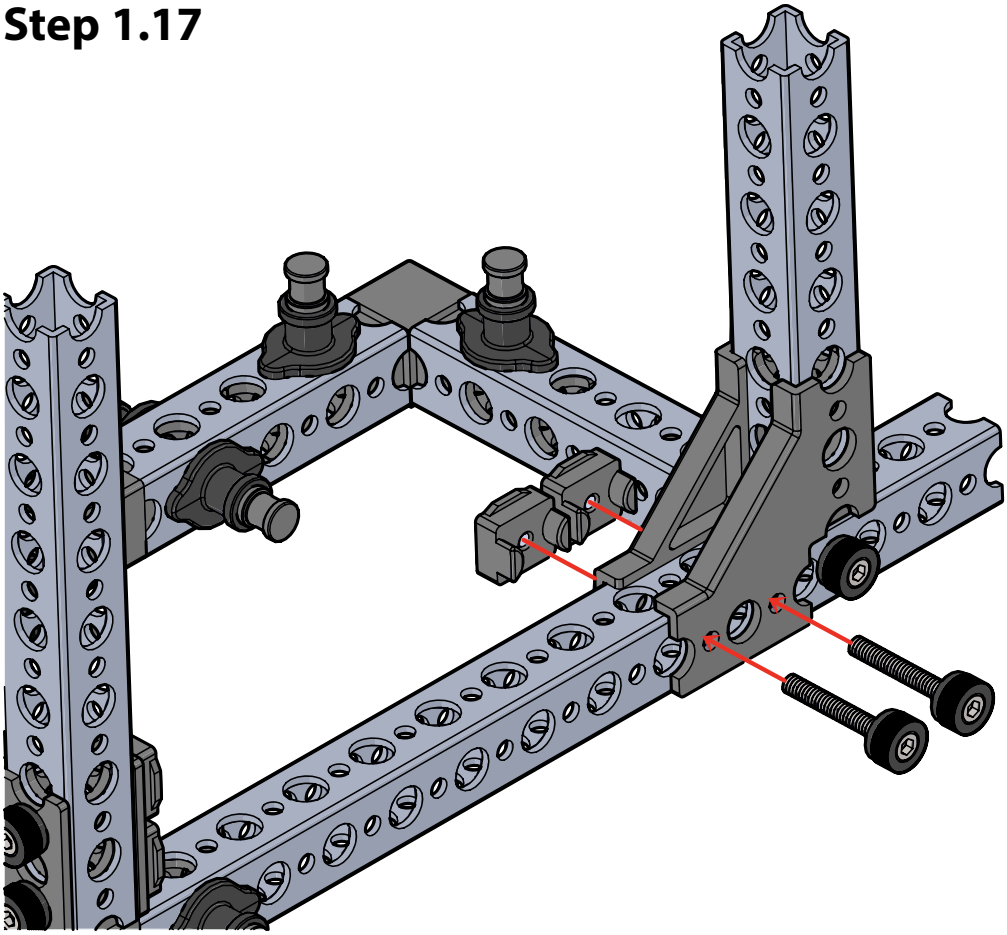
Step 1.15



Step 1.16

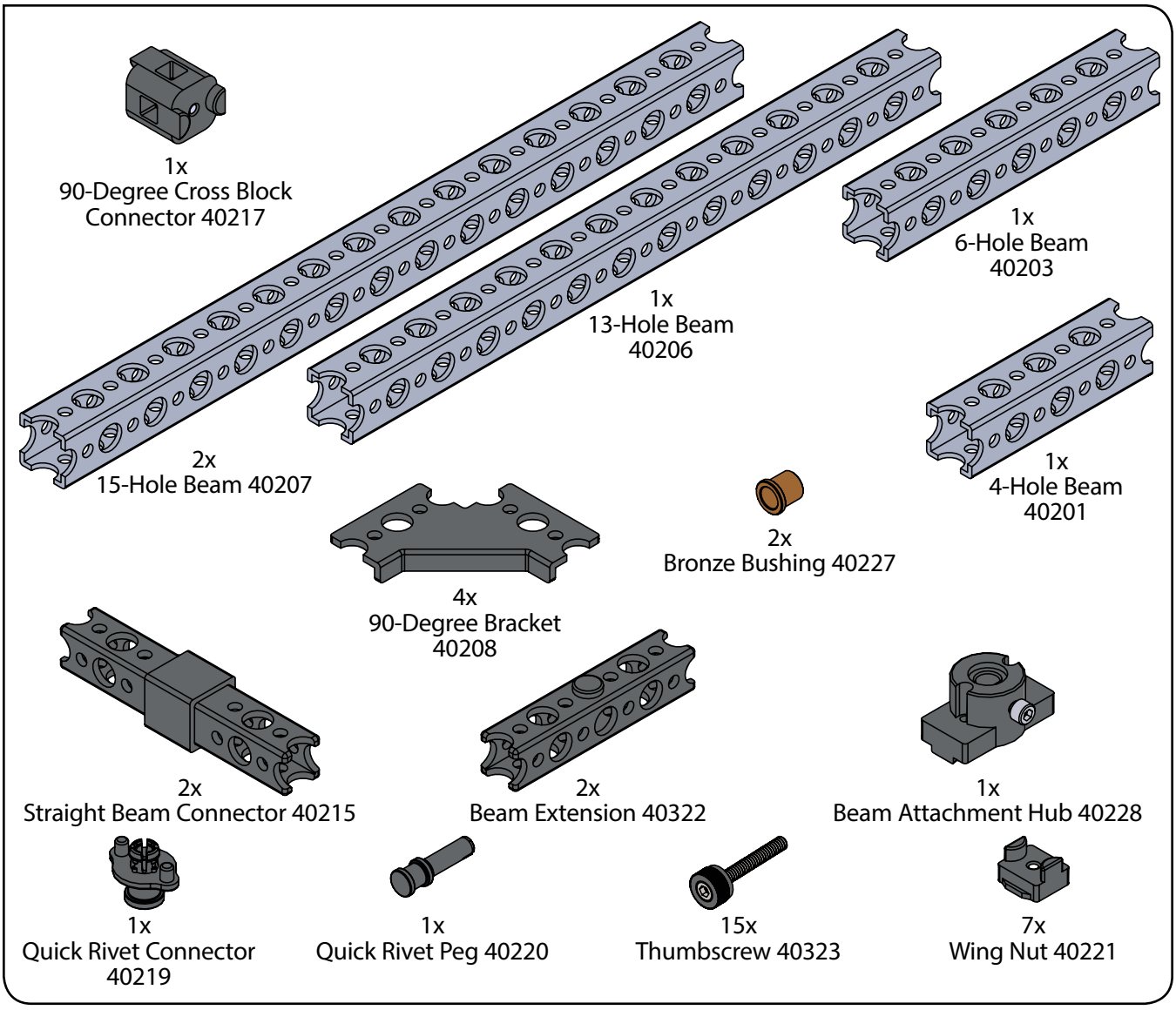


Step 1.17

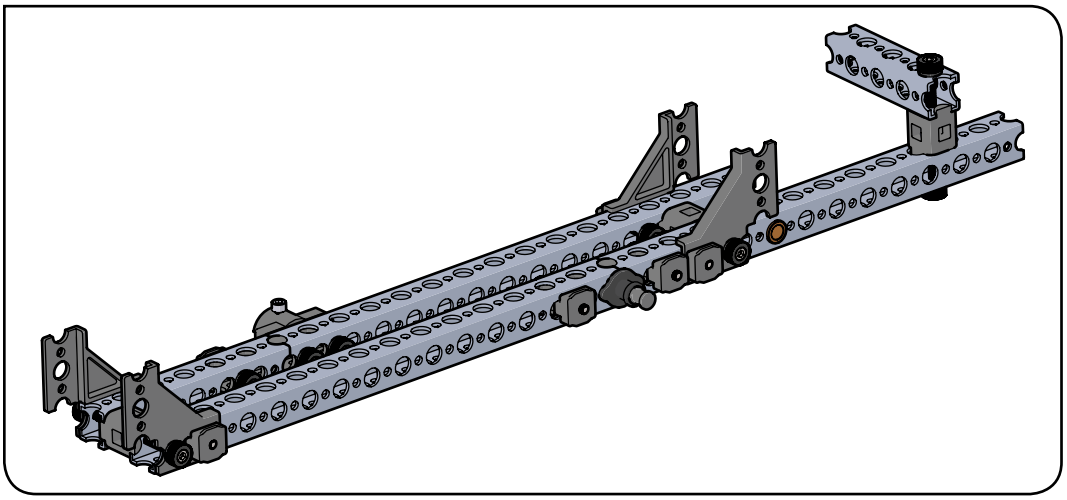


Step 2

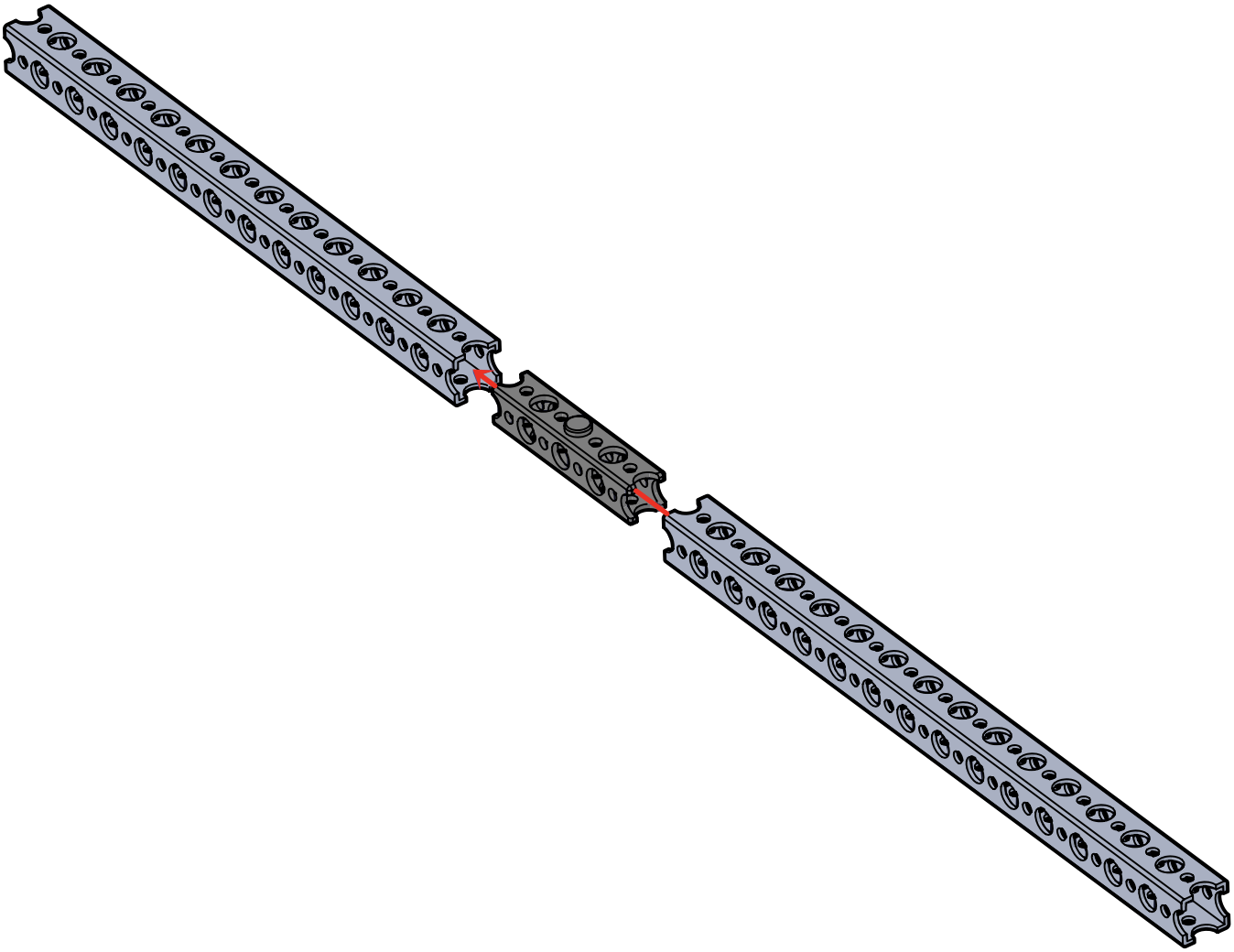
Parts Needed



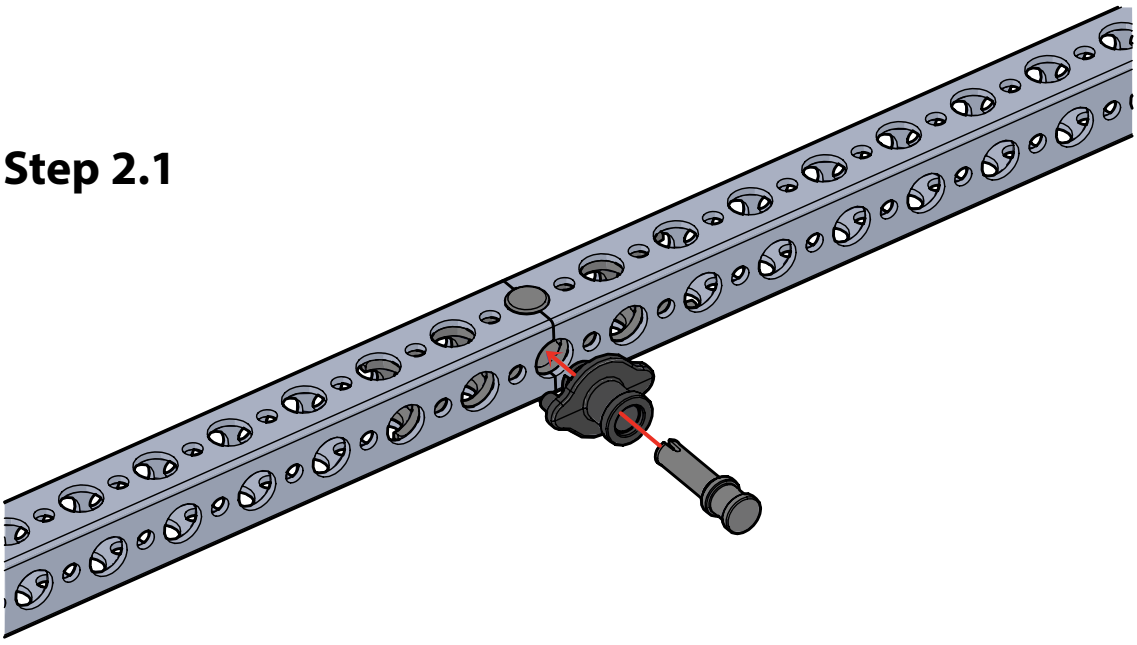
Partial assembly should look like this.



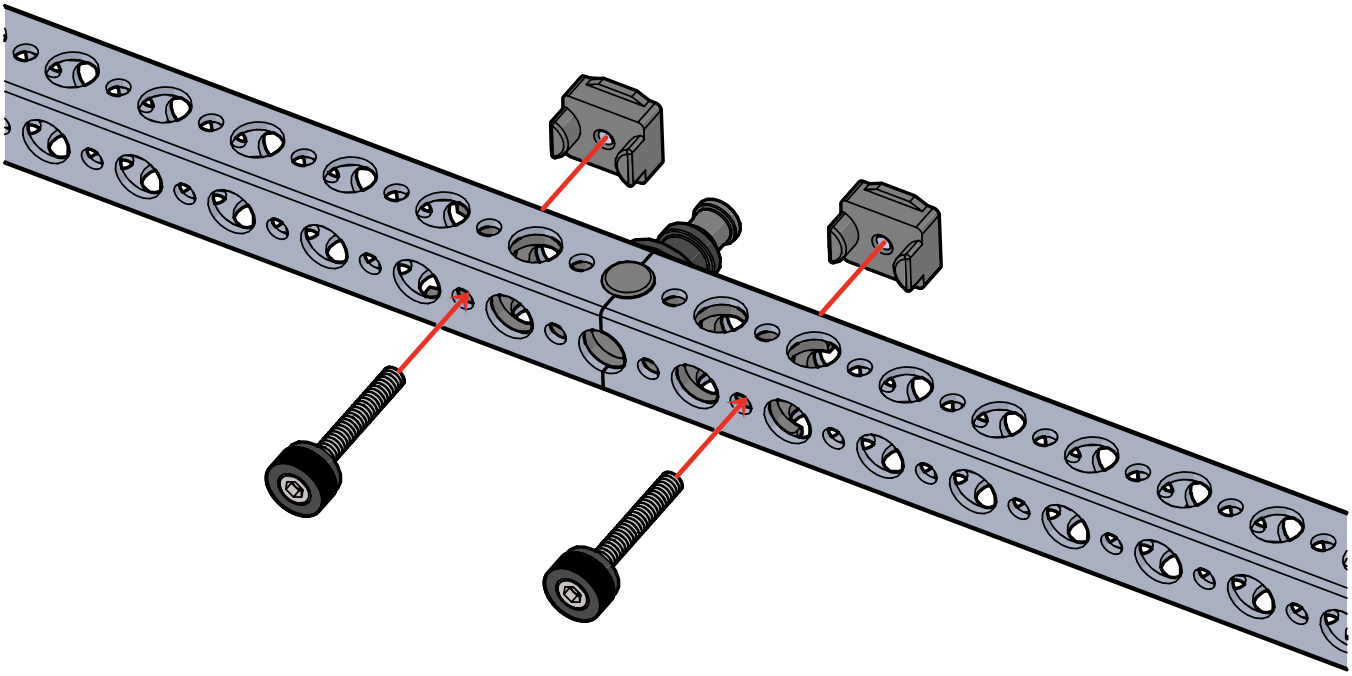
Step 2.0



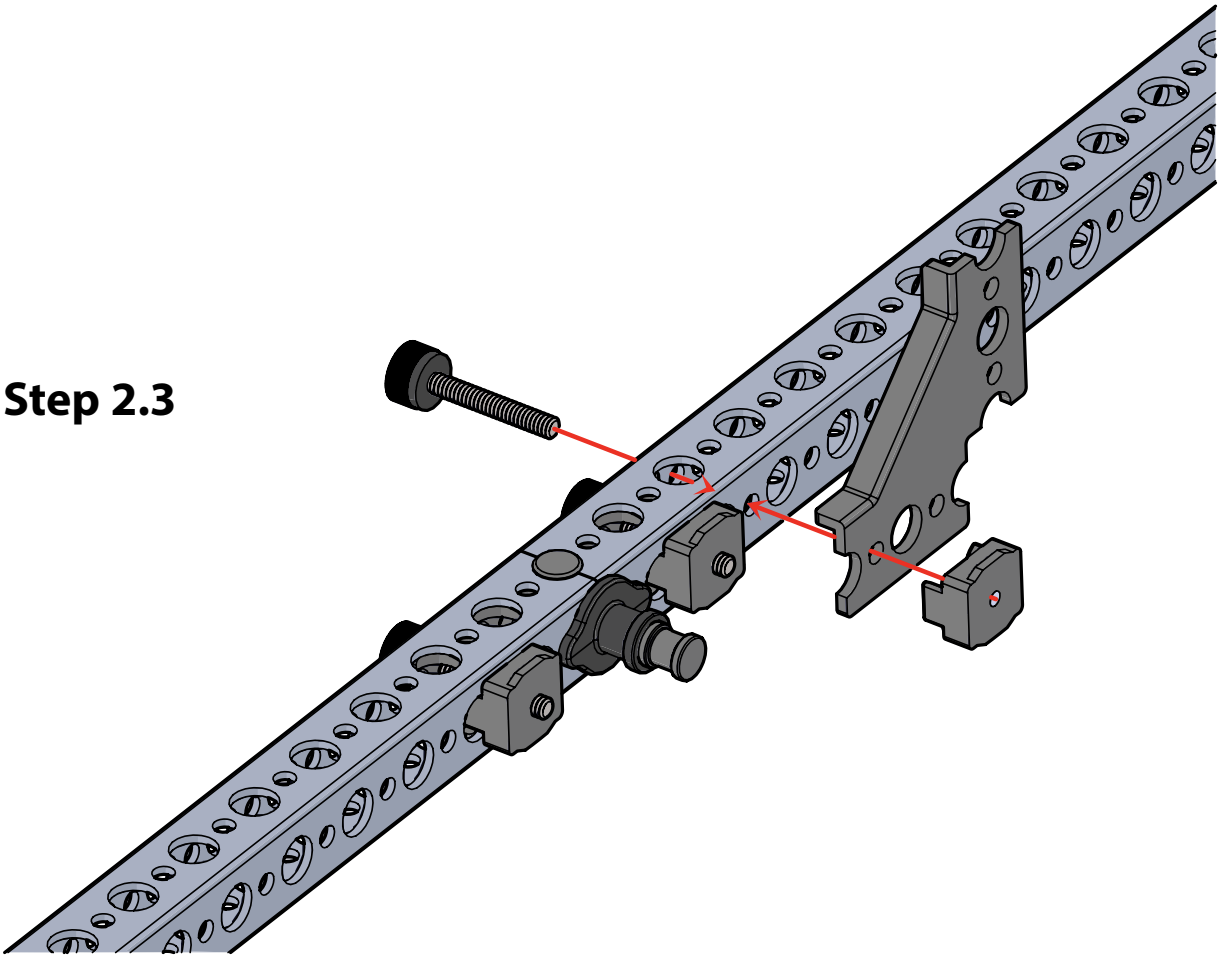
Step 2.1



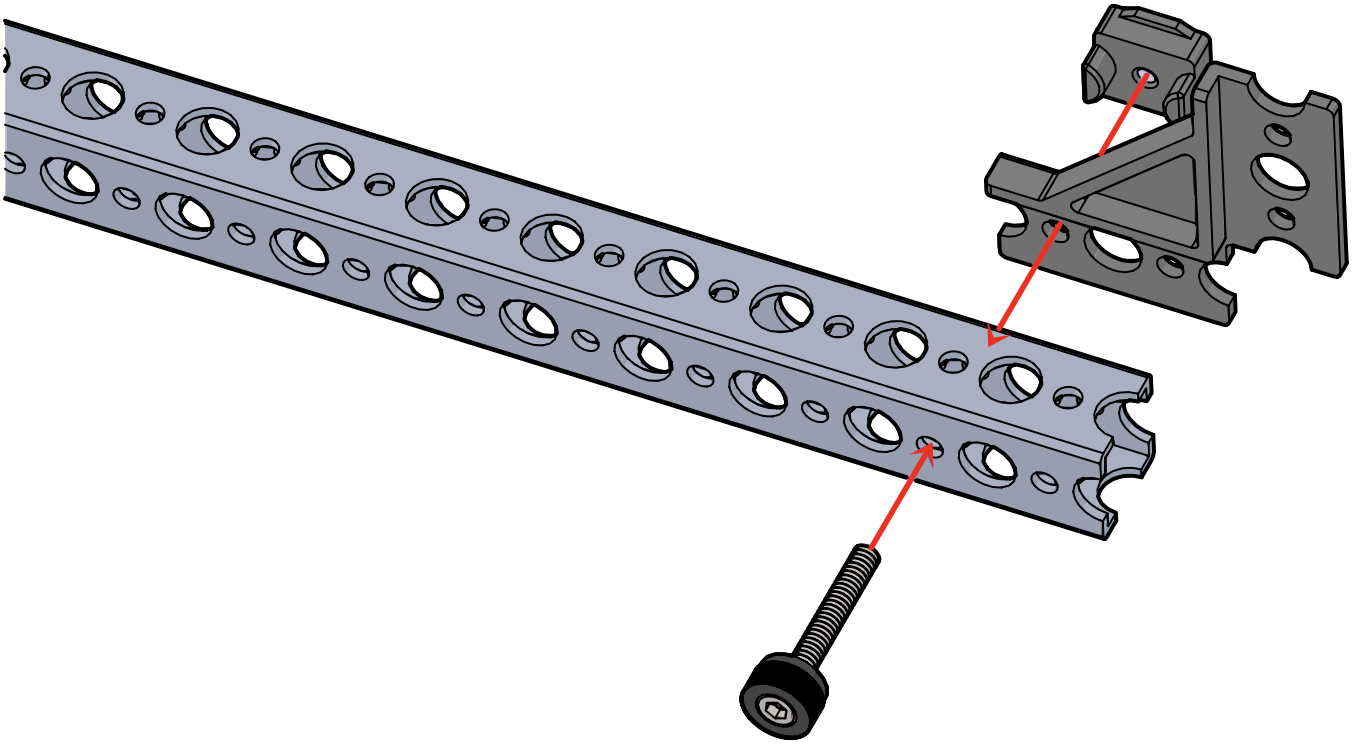
Step 2.2



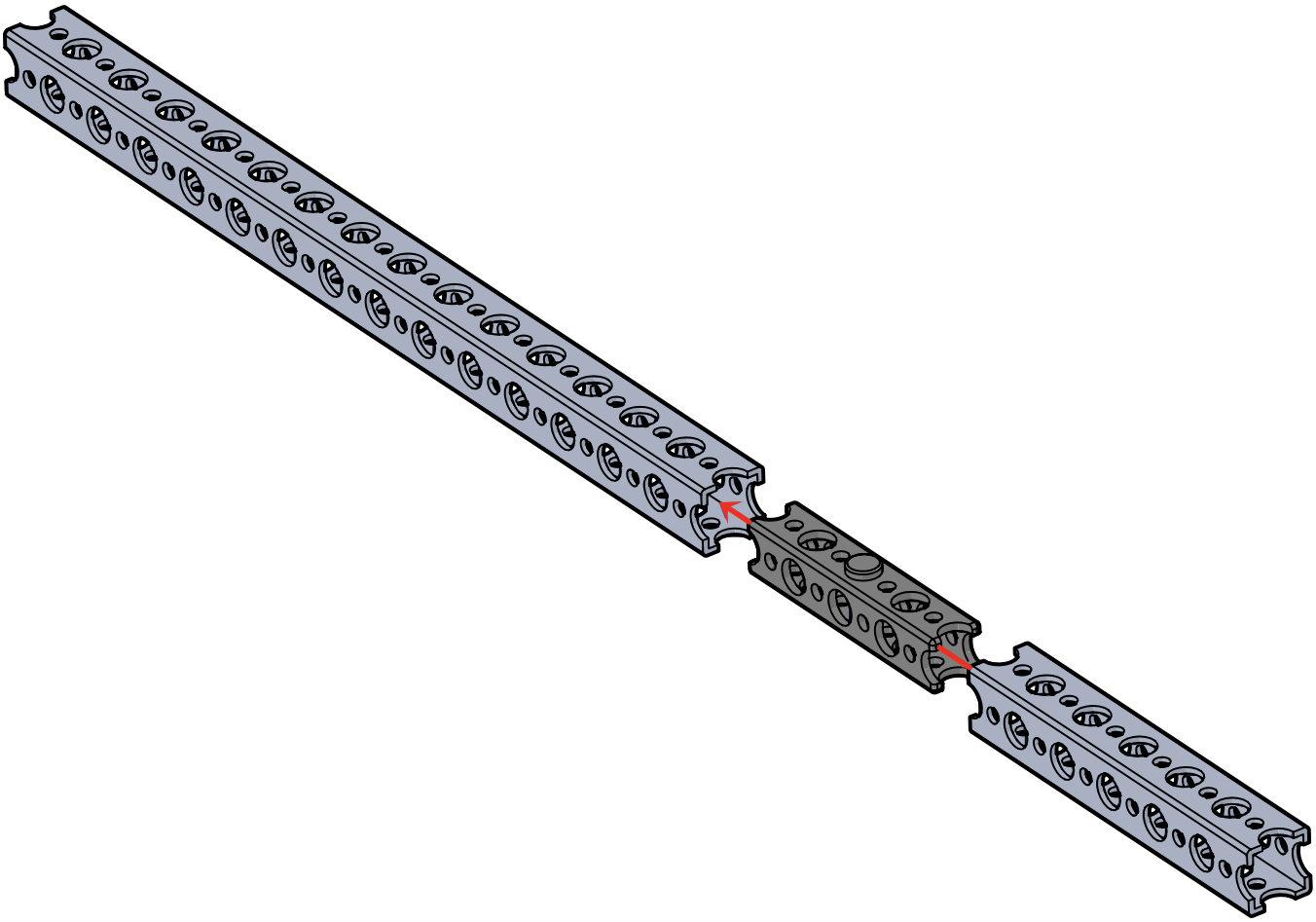
Step 2.3



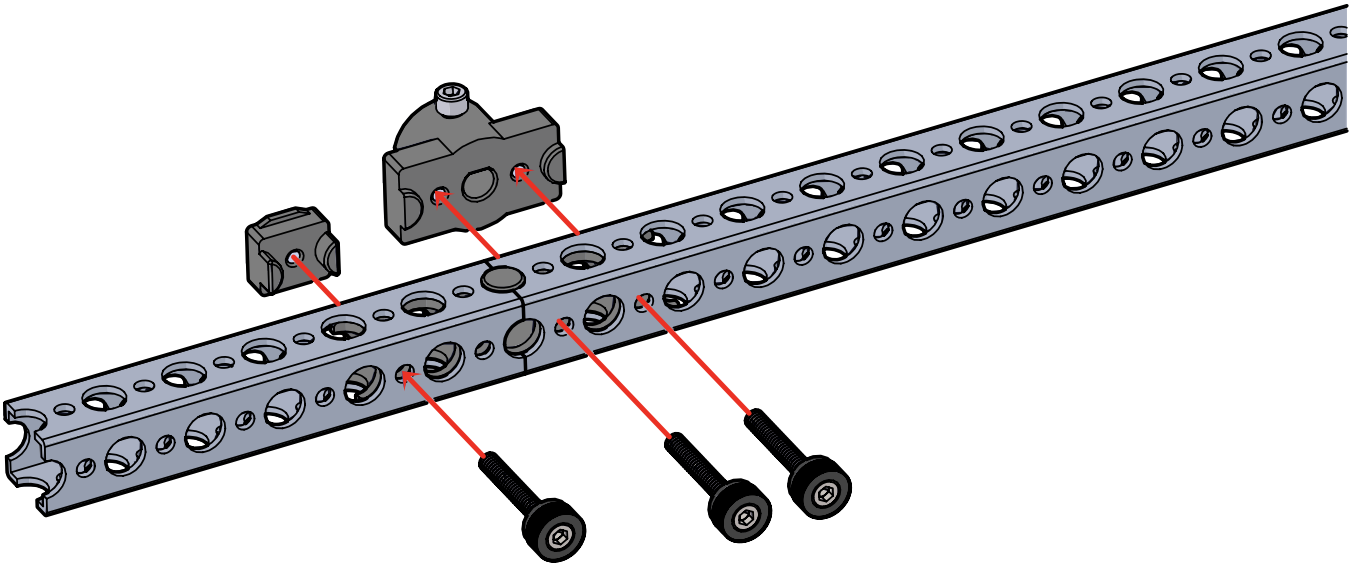
Step 2.4



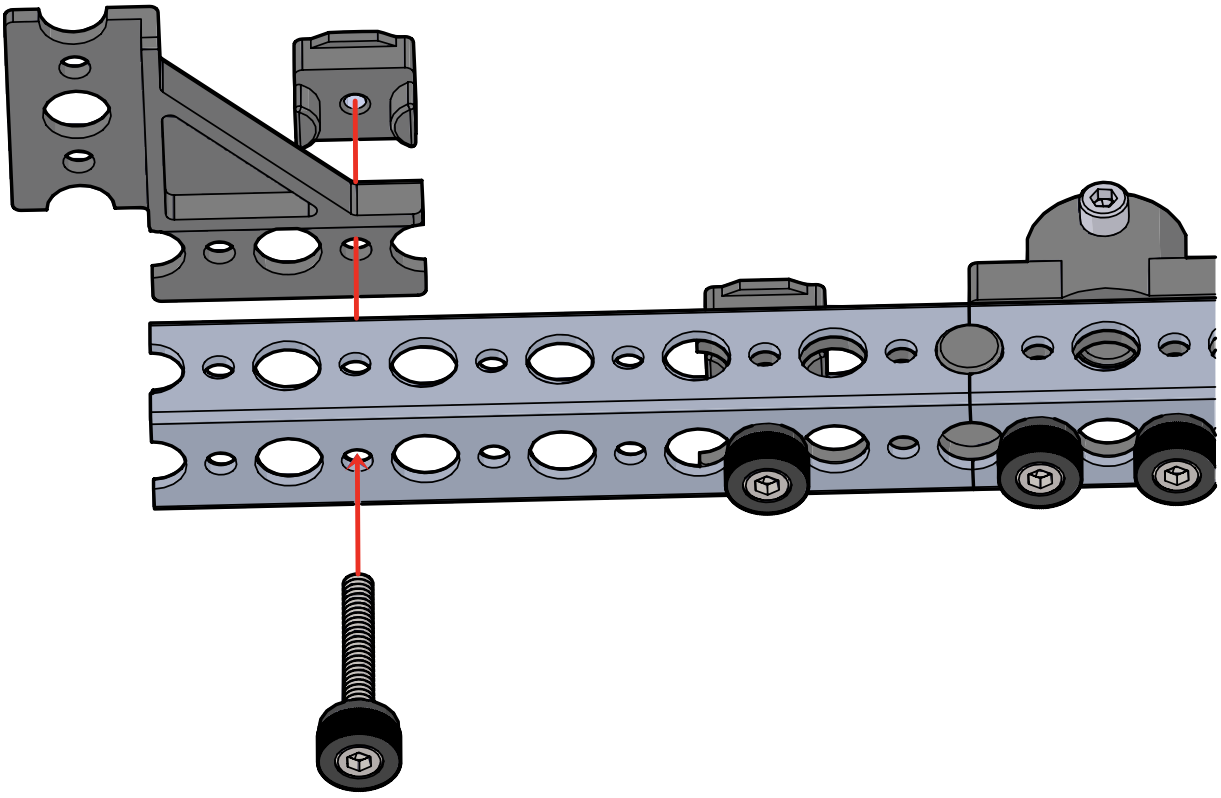
Step 2.5



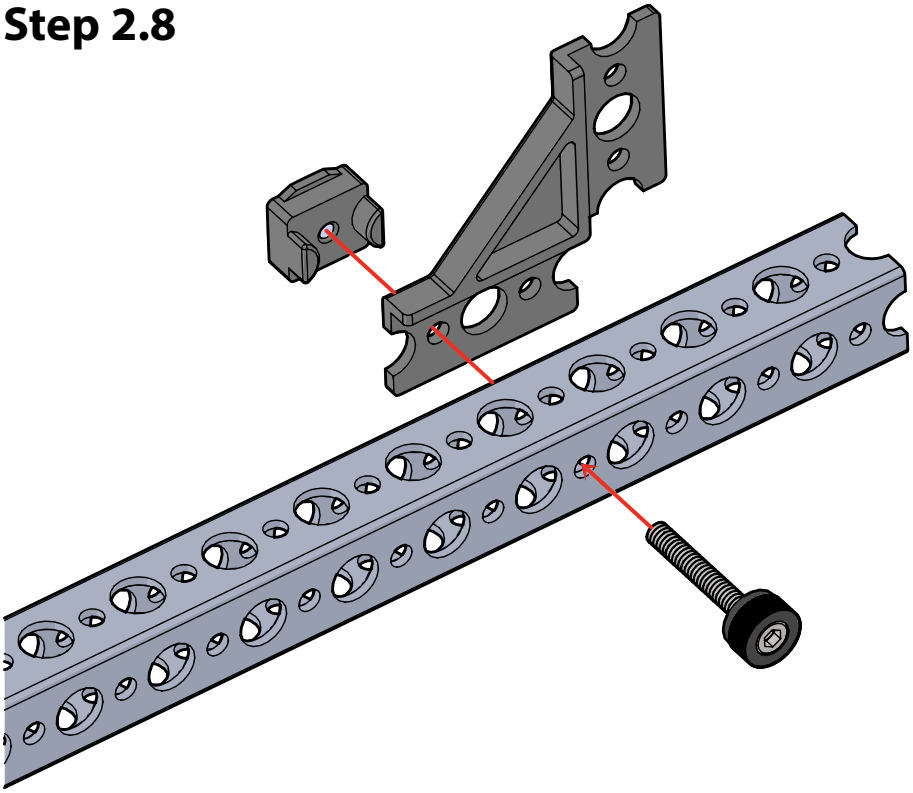
Step 2.6



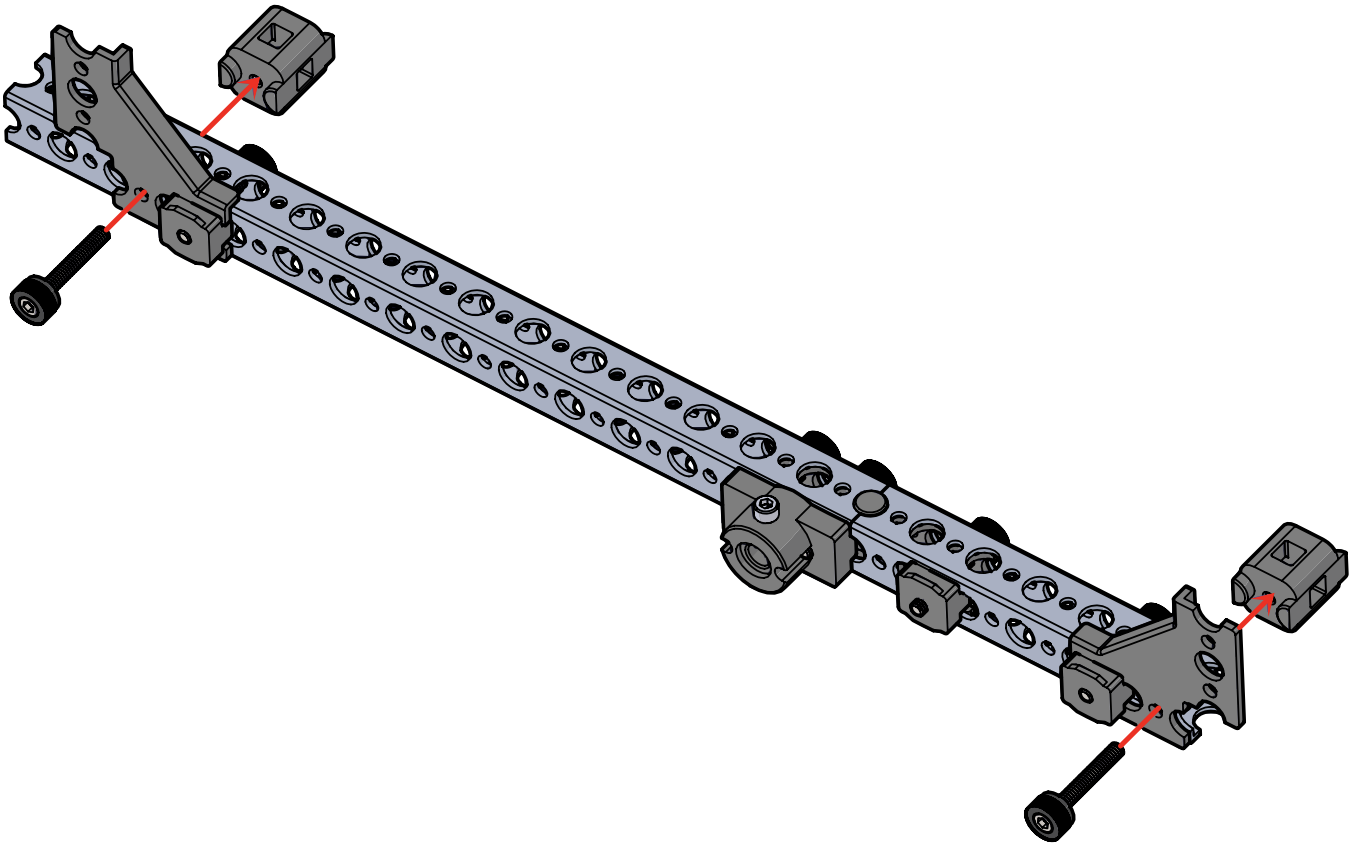
Step 2.7



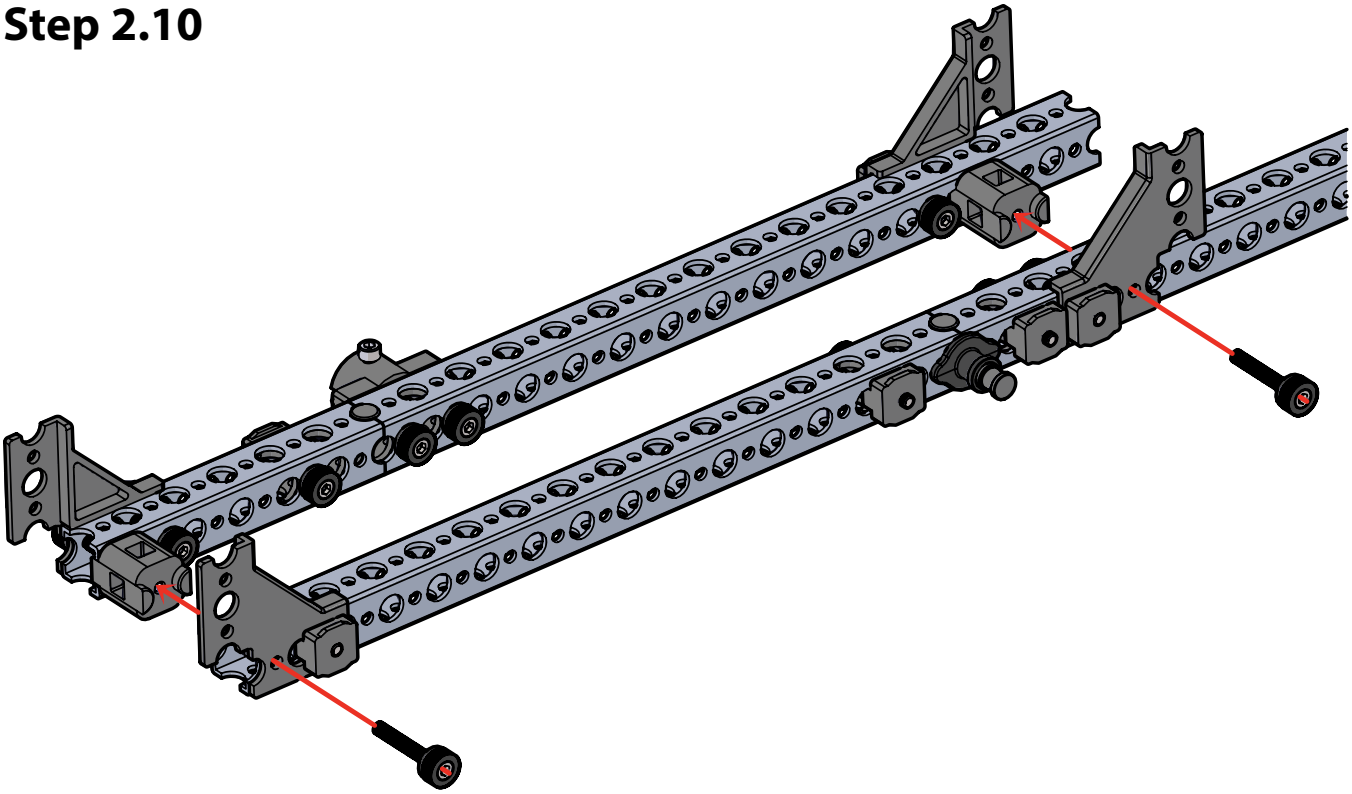
Step 2.8



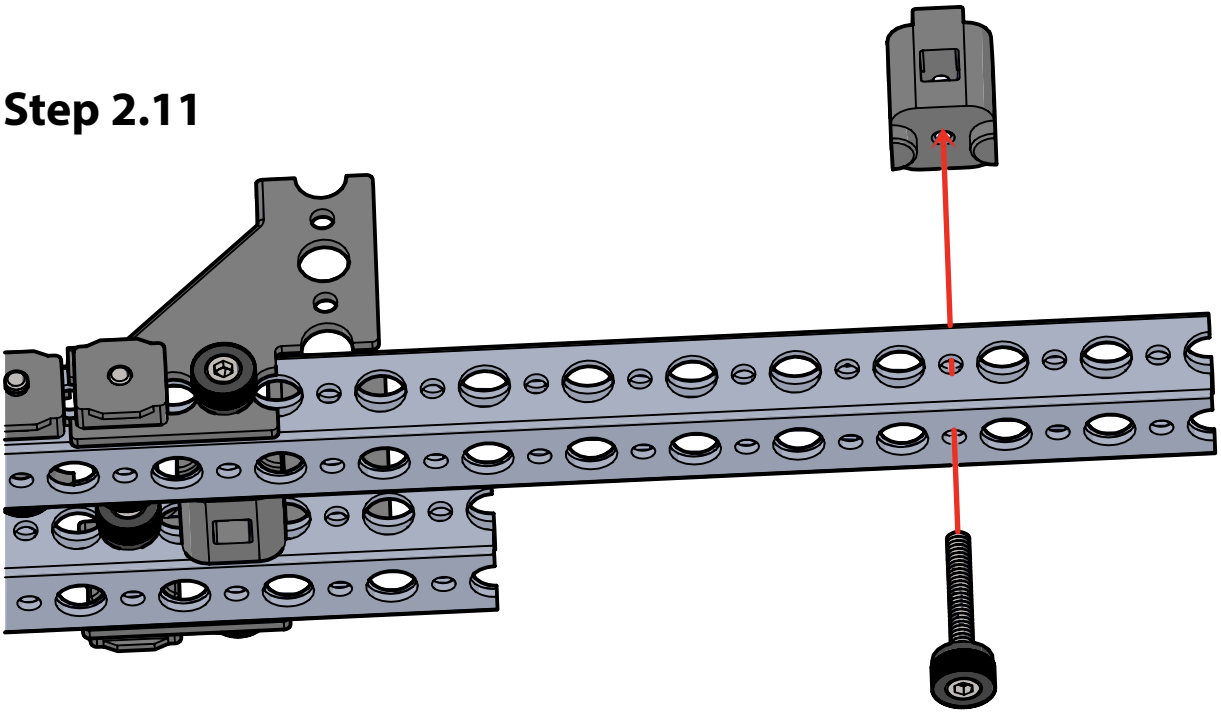
Step 2.9



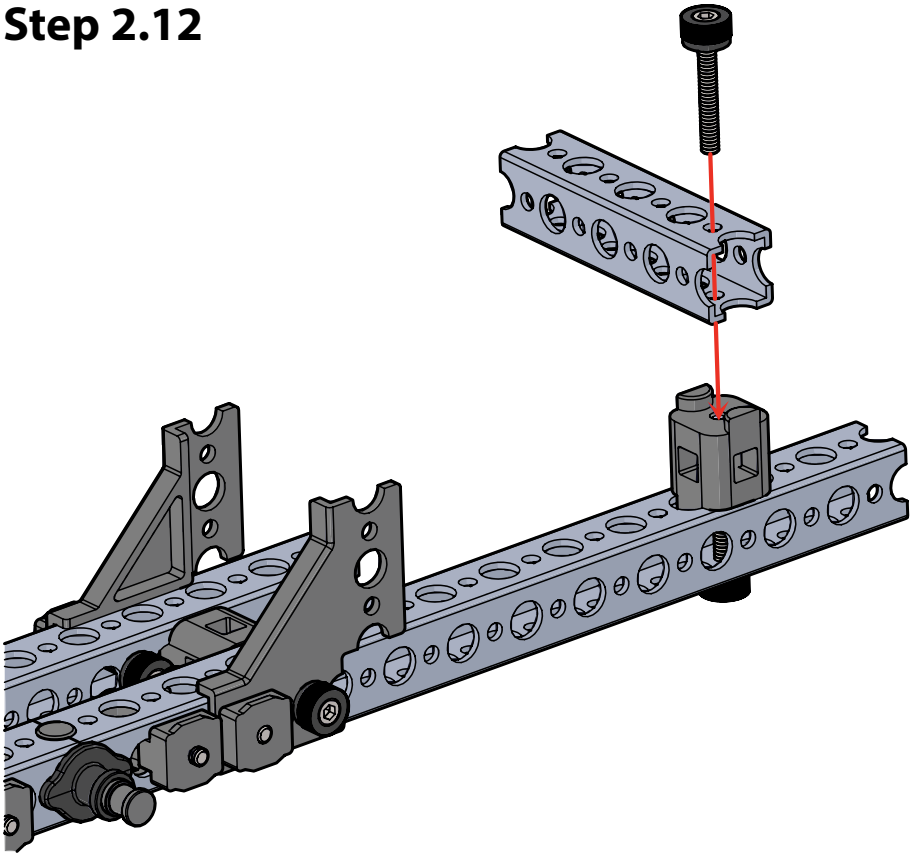
Step 2.10



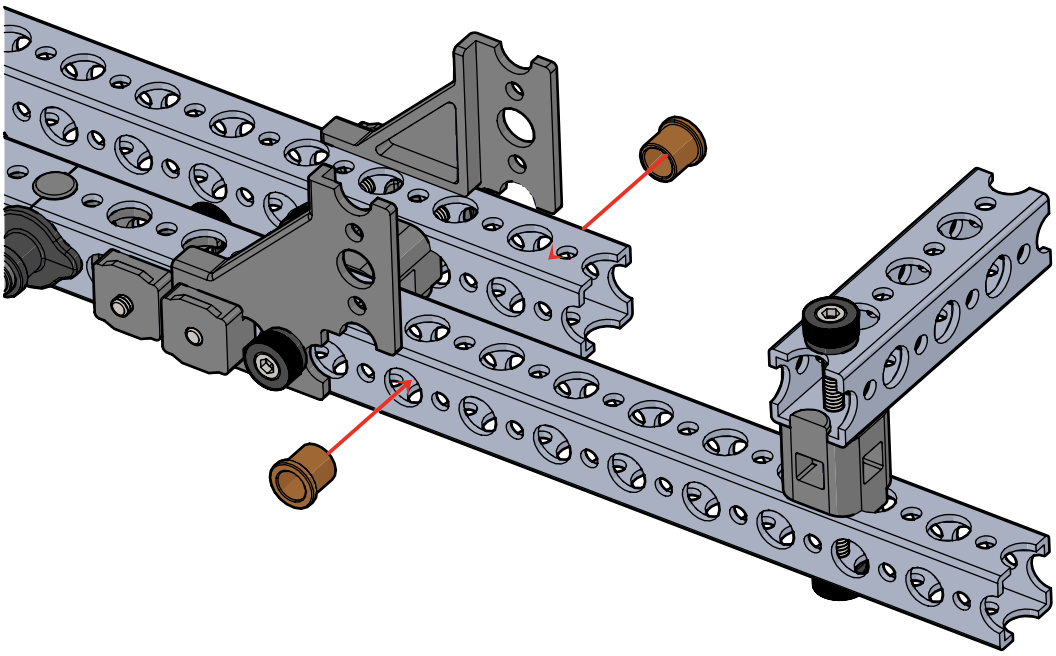
Step 2.11



Step 2.12

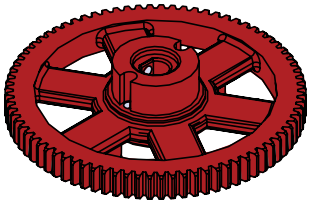


Step 2.13

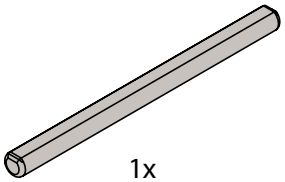


Step 3

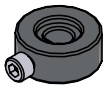
Parts Needed



2x
80-Tooth Gear 40224



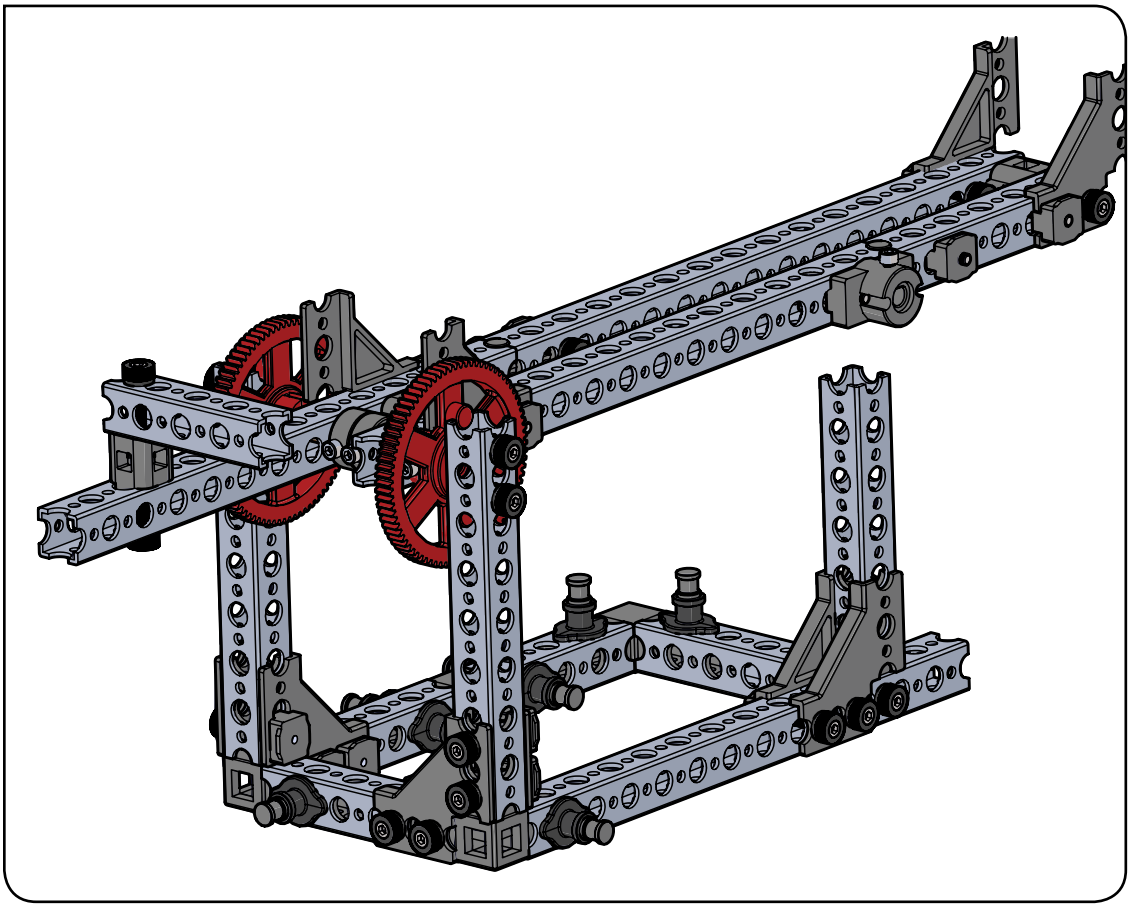
1x
80 mm Axle 40225



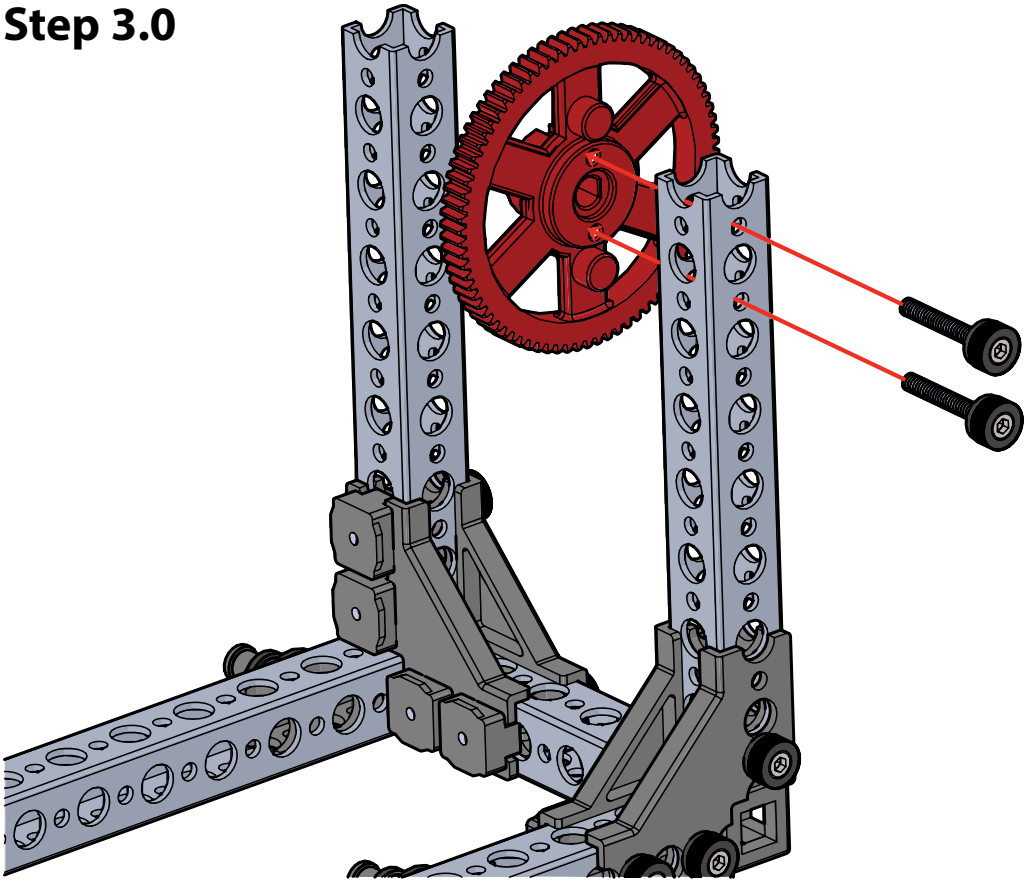
2x
Shaft Collar 40229

4x
Thumbscrew 40323

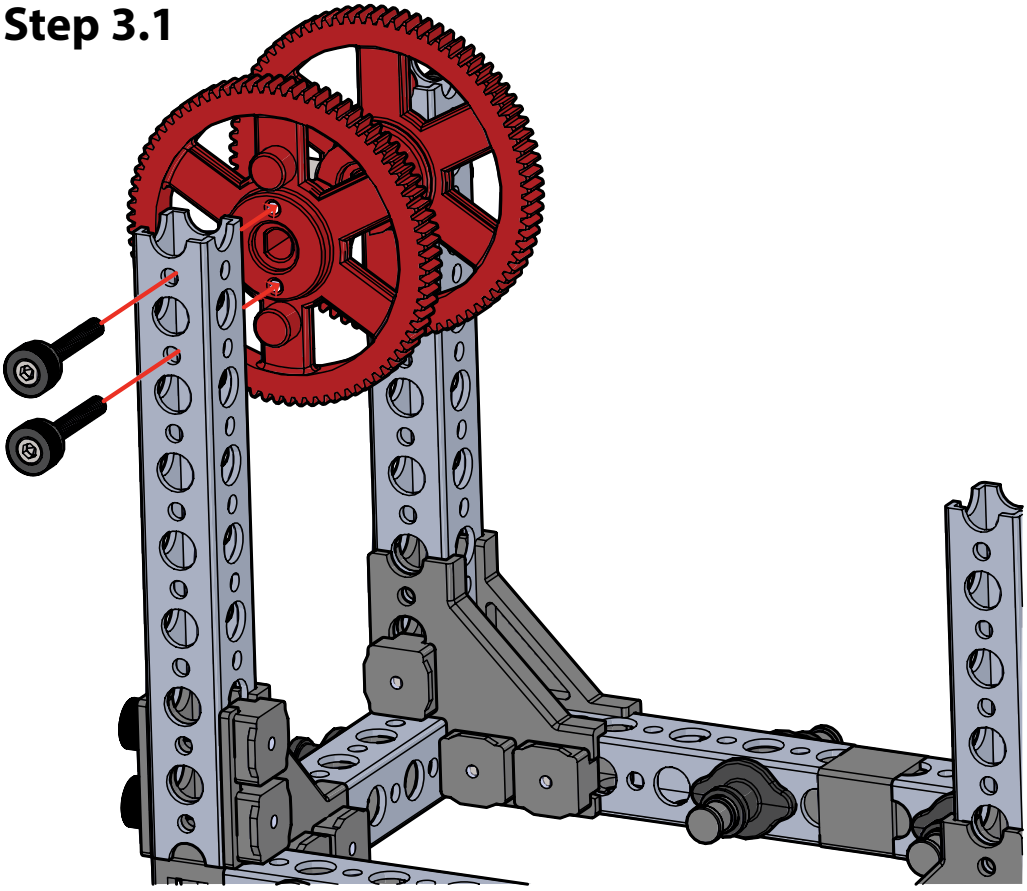
Partial assembly should look like this.



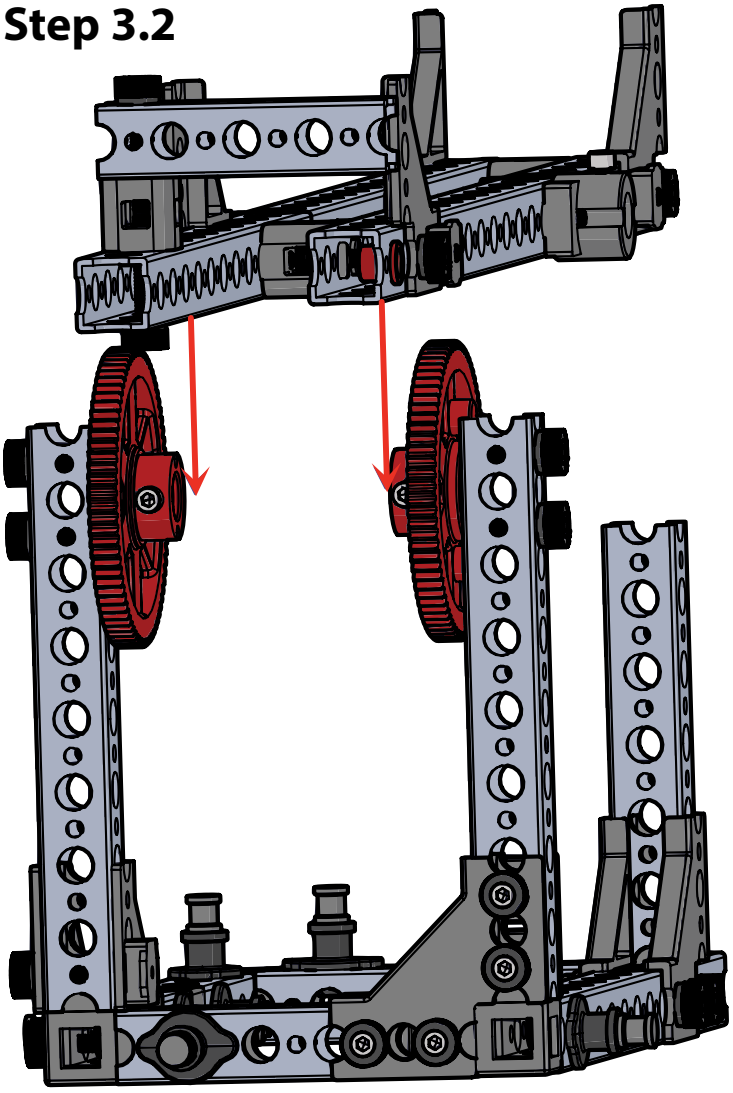
Step 3.0



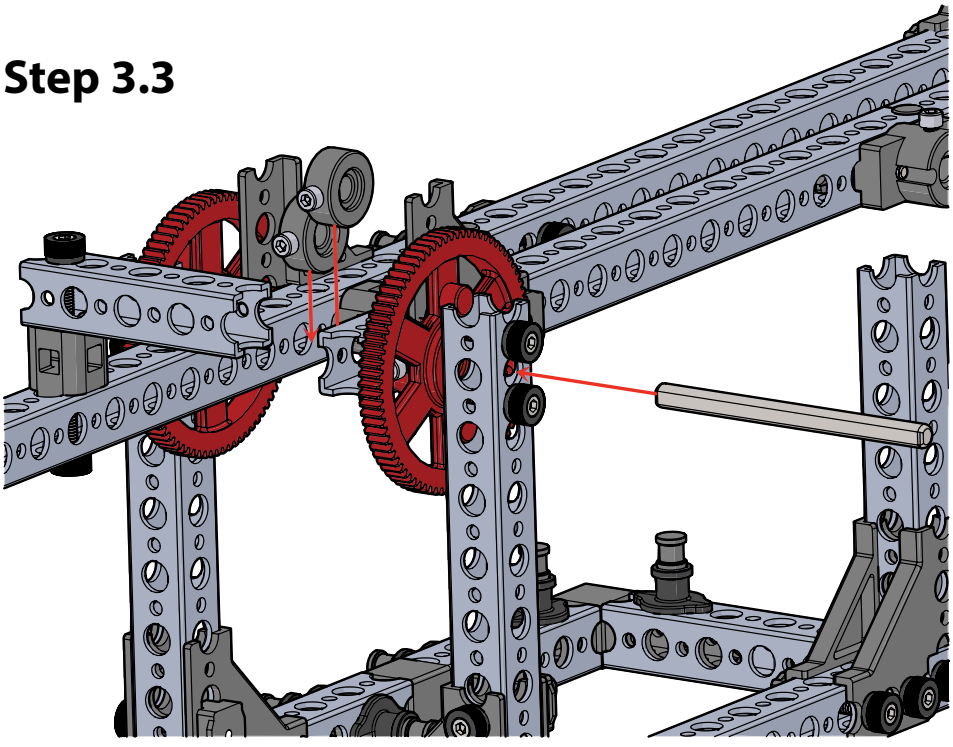
Step 3.1



Step 3.2

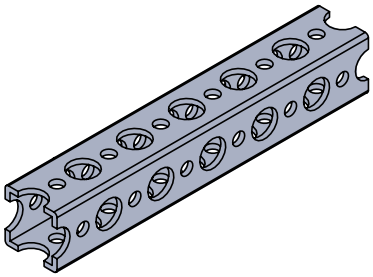


Step 3.3

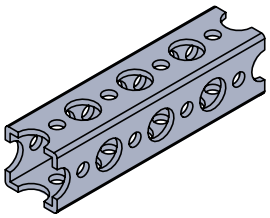


Step 4

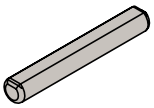
Parts Needed



1x
6-Hole Beam 40203



1x
4-Hole Beam 40201

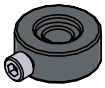


3x
40 mm Axle 40226

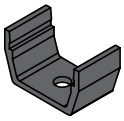
6x
Bronze Bushing 40227

5x
Thumbscrew 40323

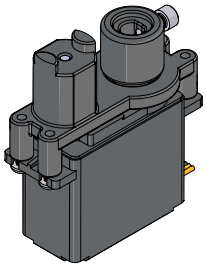
1x
Wing Nut 40221



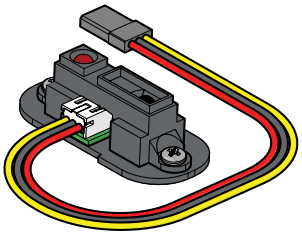
2x
Shaft Collar 40229



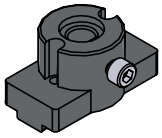
1x
Digilent Sensor
Mounting Clip 41148



1x
Standard Servo with
Servo Mount

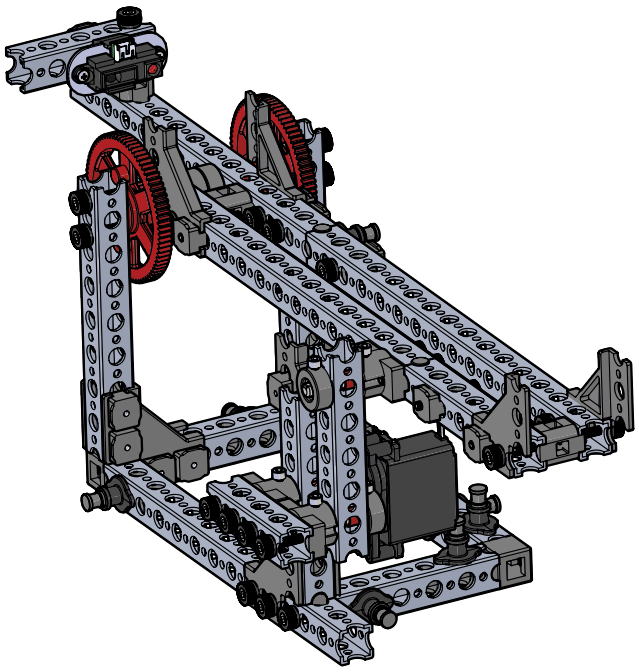


1x
Digilent Sensor
IR Rangefinder with
Mounting Plate

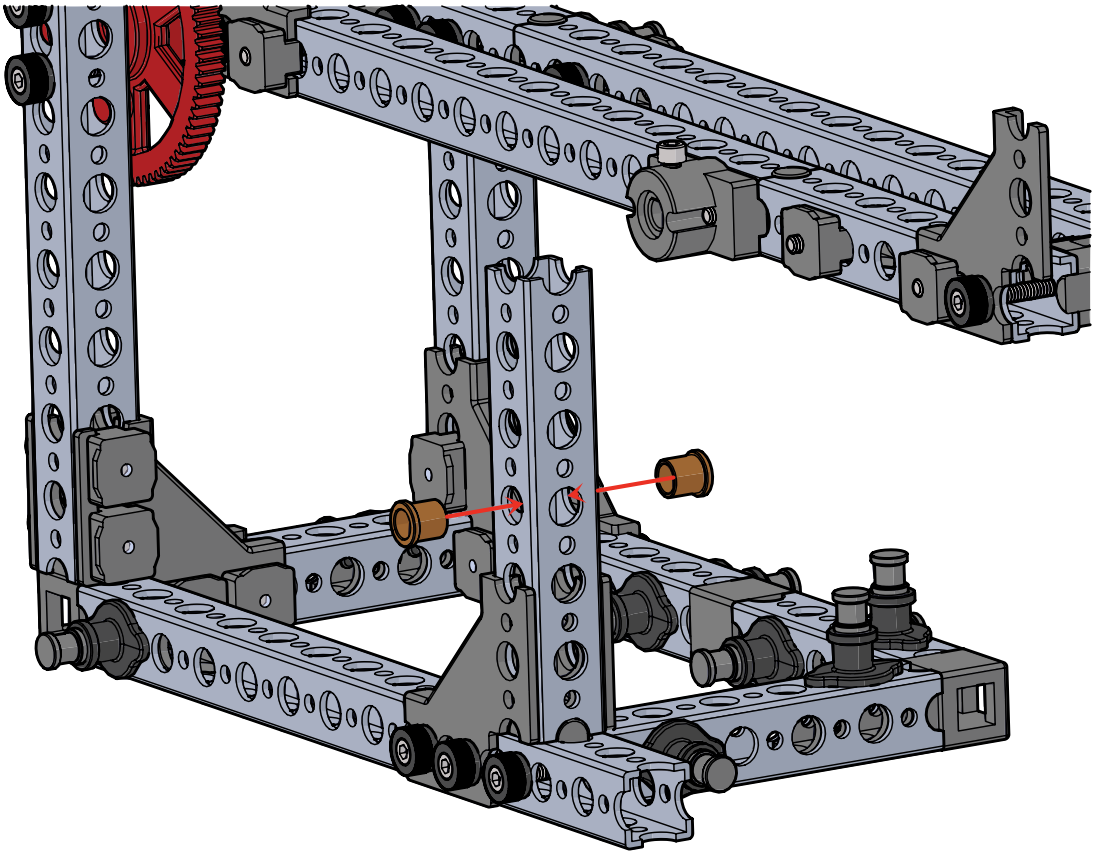


2x
Beam Attachment
Hub 40228

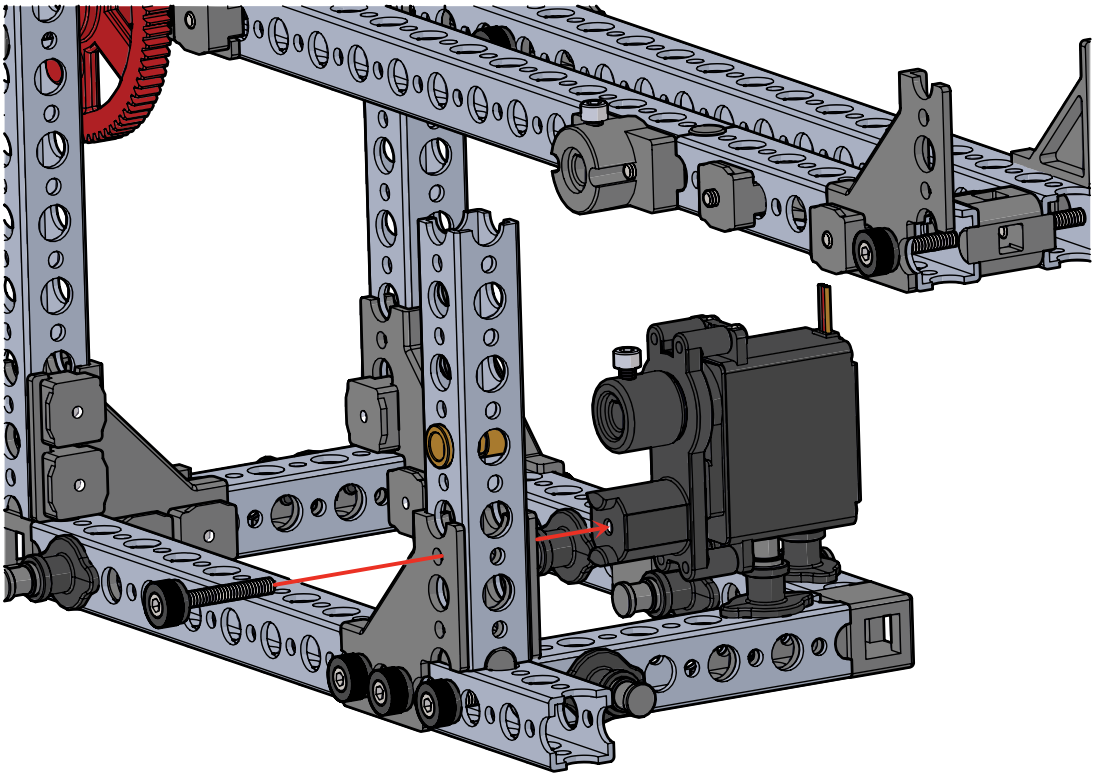
Partial assembly should look like this.



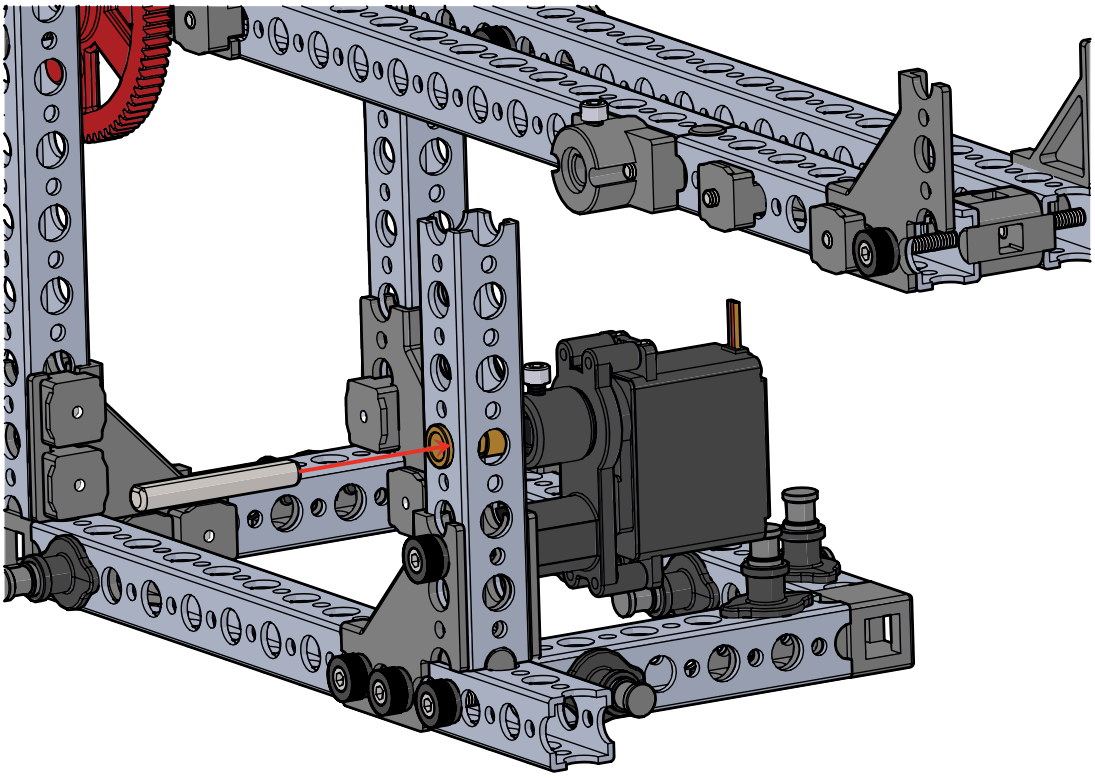
Step 4.0



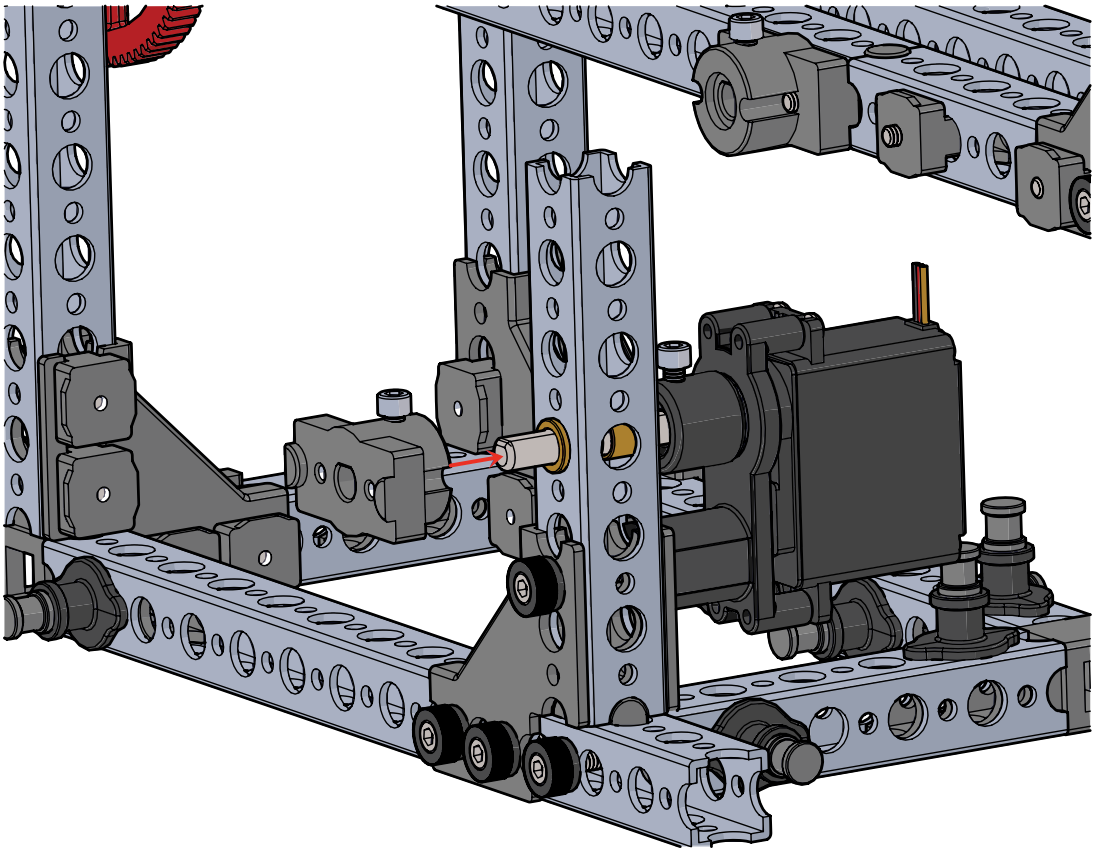
Step 4.1



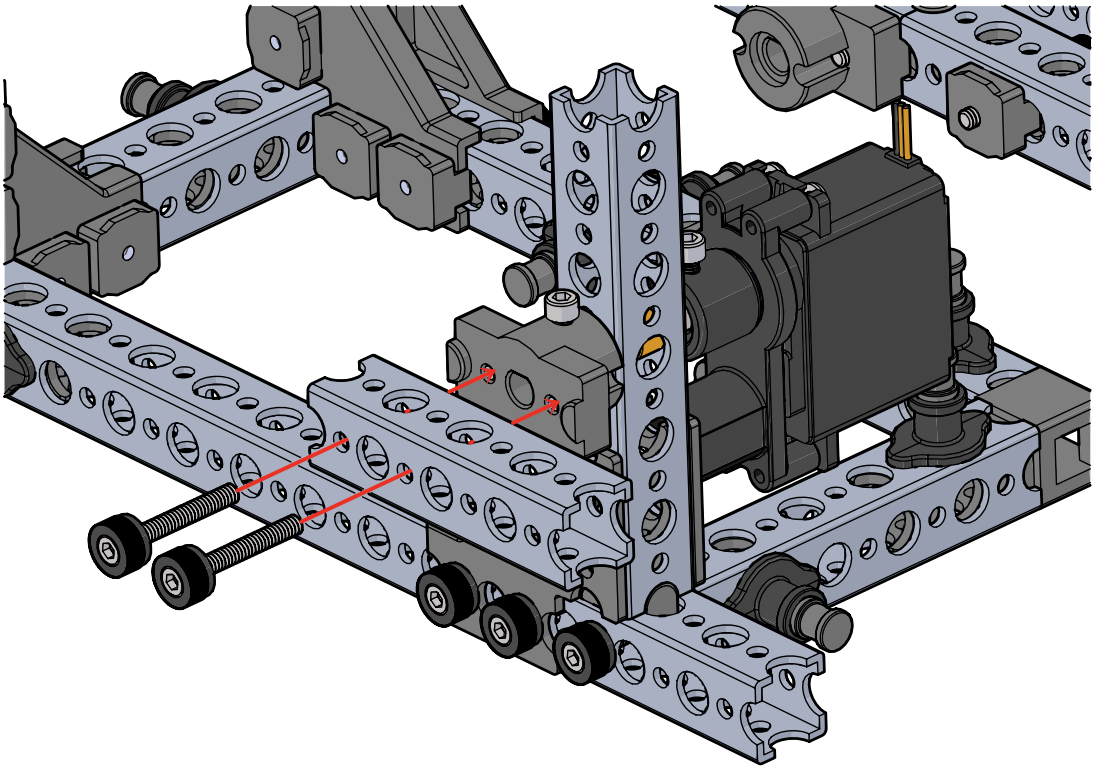
Step 4.2



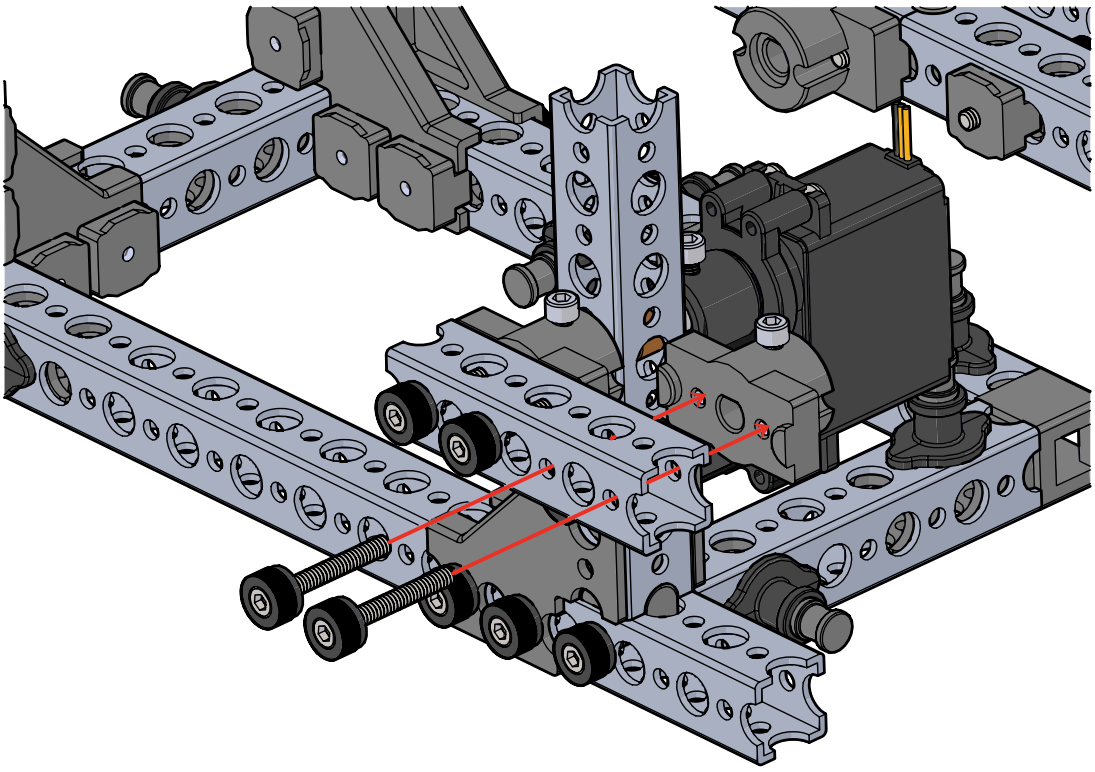
Step 4.3



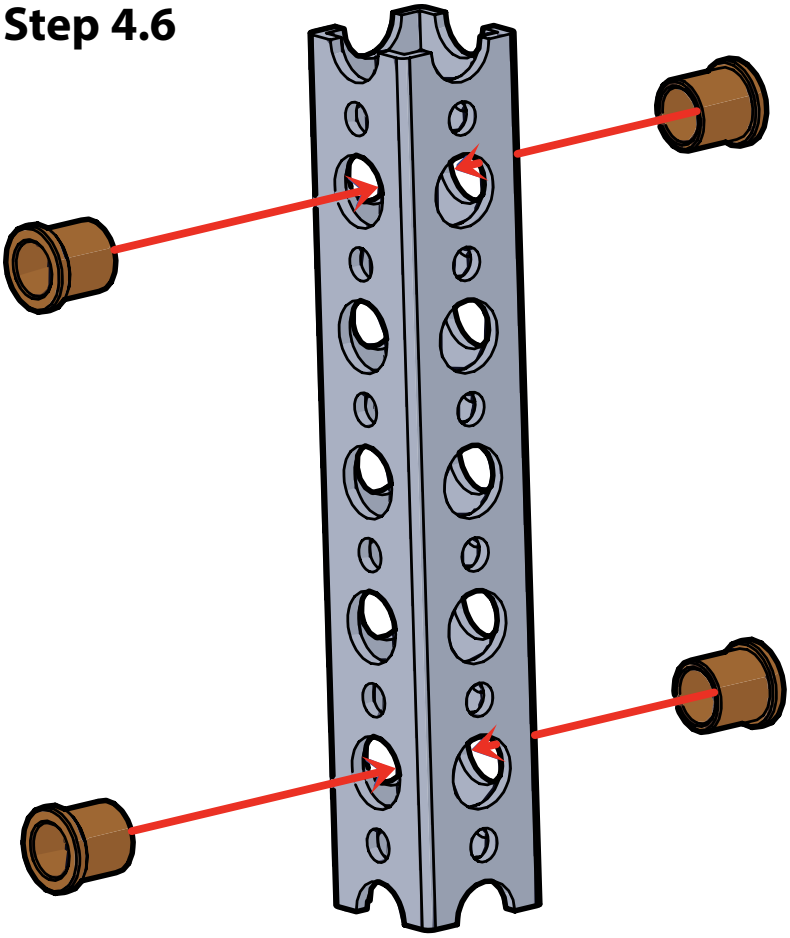
Step 4.4



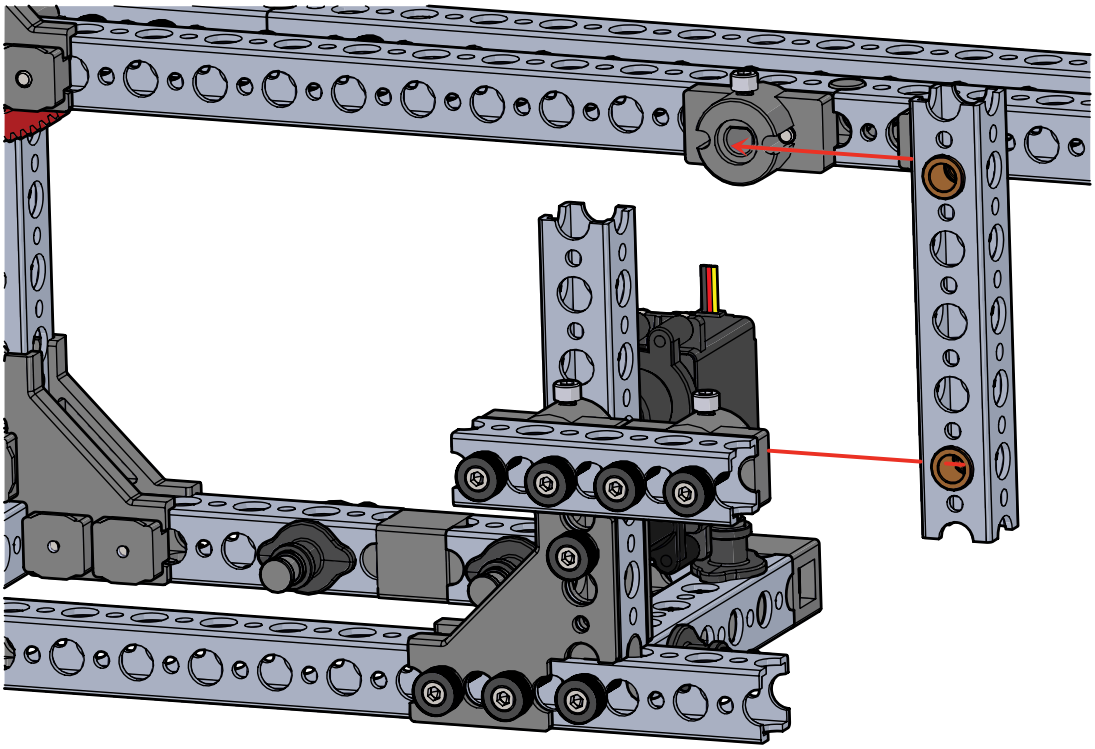
Step 4.5



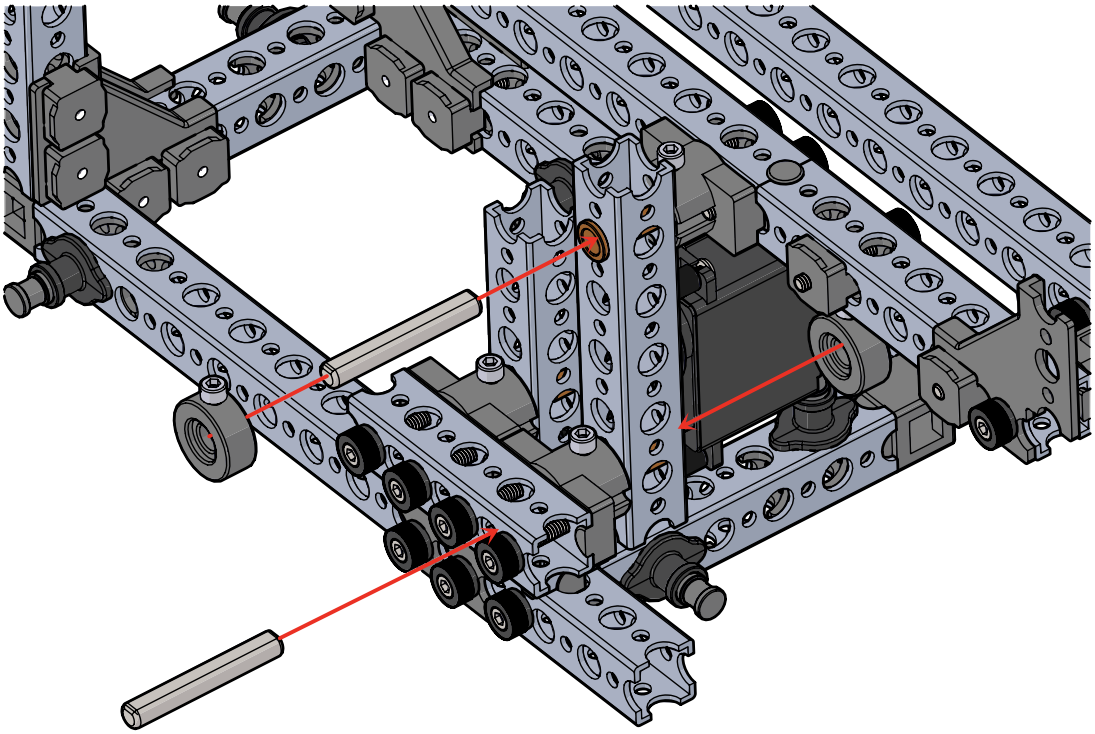
Step 4.6



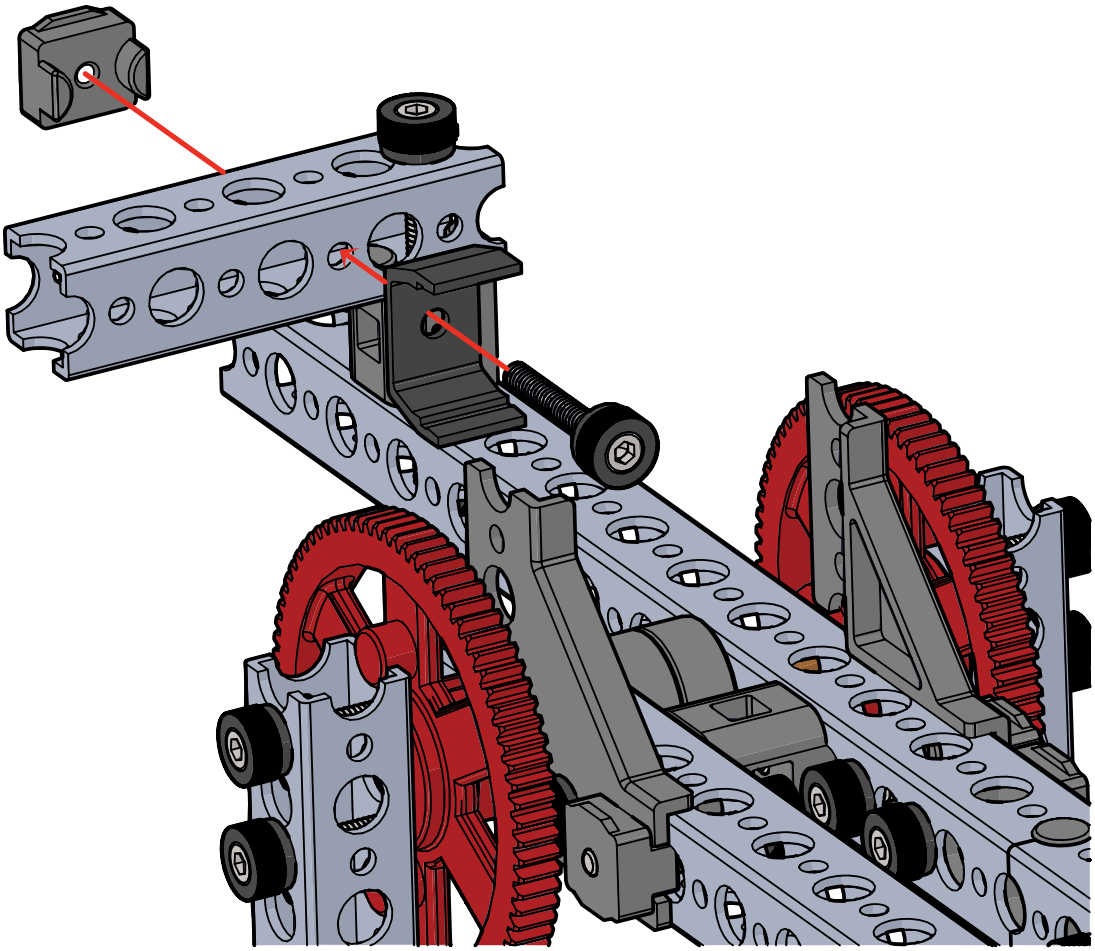
Step 4.7



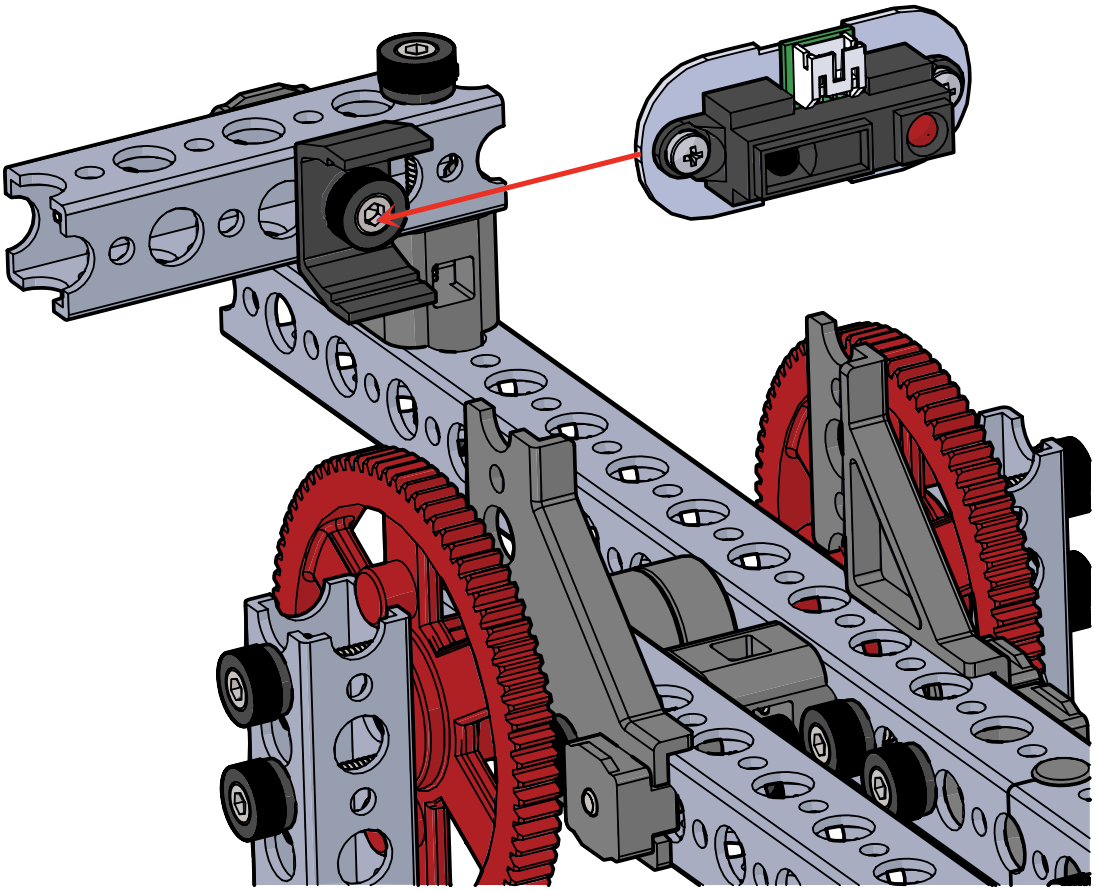
Step 4.8



Step 4.9

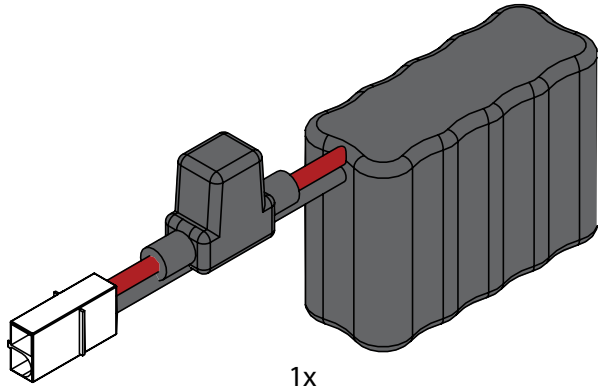


Step 4.10

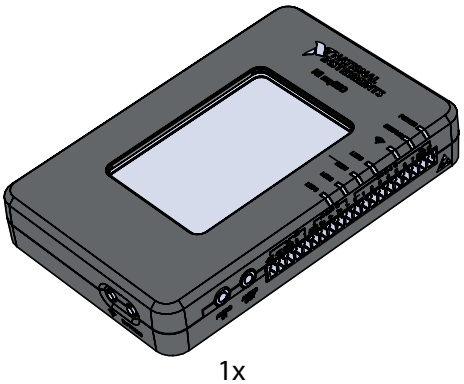


Step 5

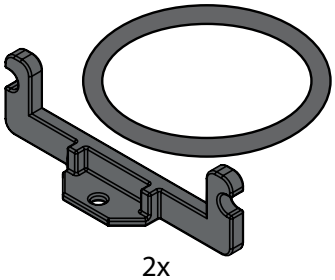
Parts Needed



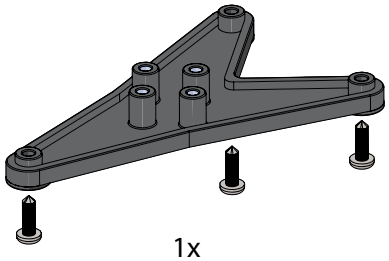
1x
10-Cell AA NiMH Battery Pack 41135



1x
myRIO Controller



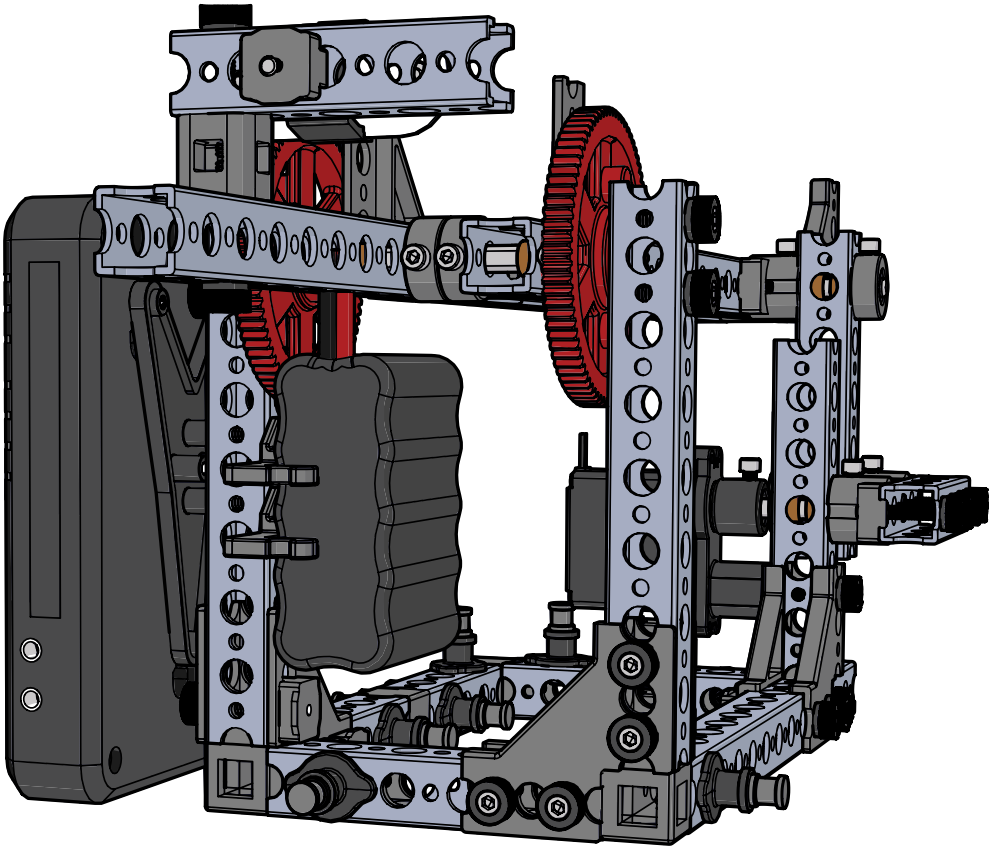
2x
Battery Holder and Bungee 40236



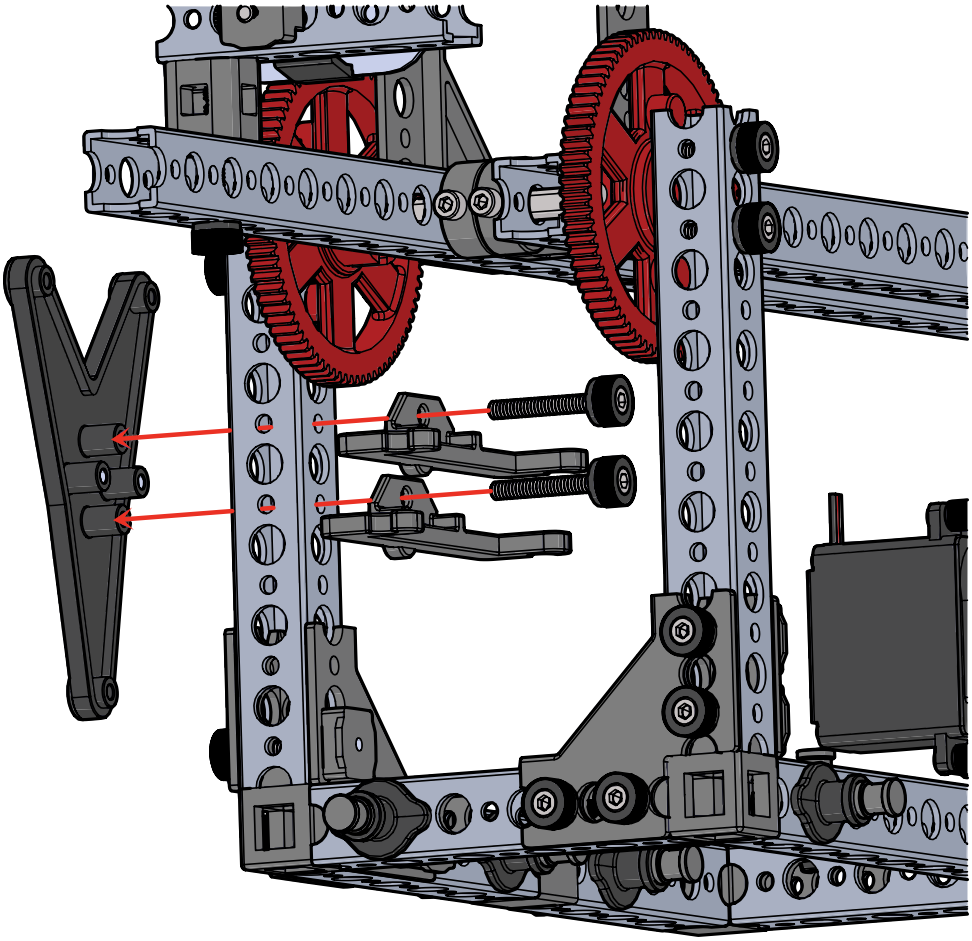
1x
myRIO Controller Mount 41137

2x
Thumbscrew 40323

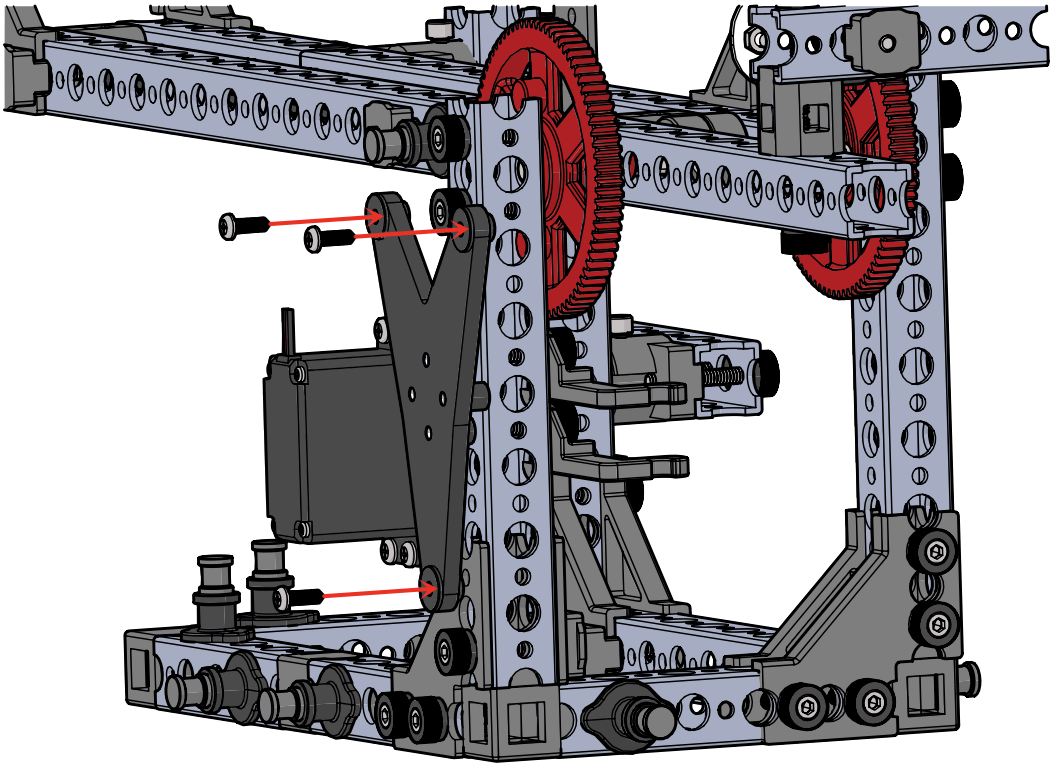
Finished assembly should look like this.



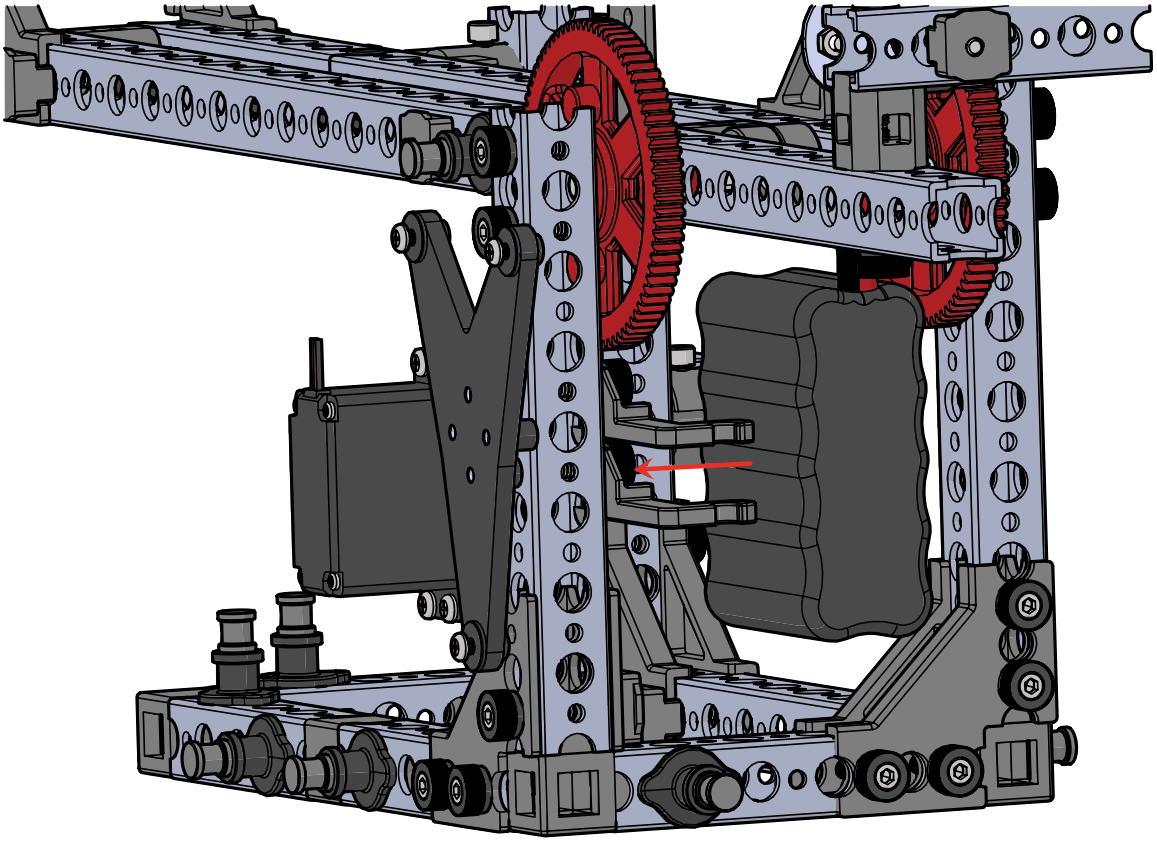
Step 5.0



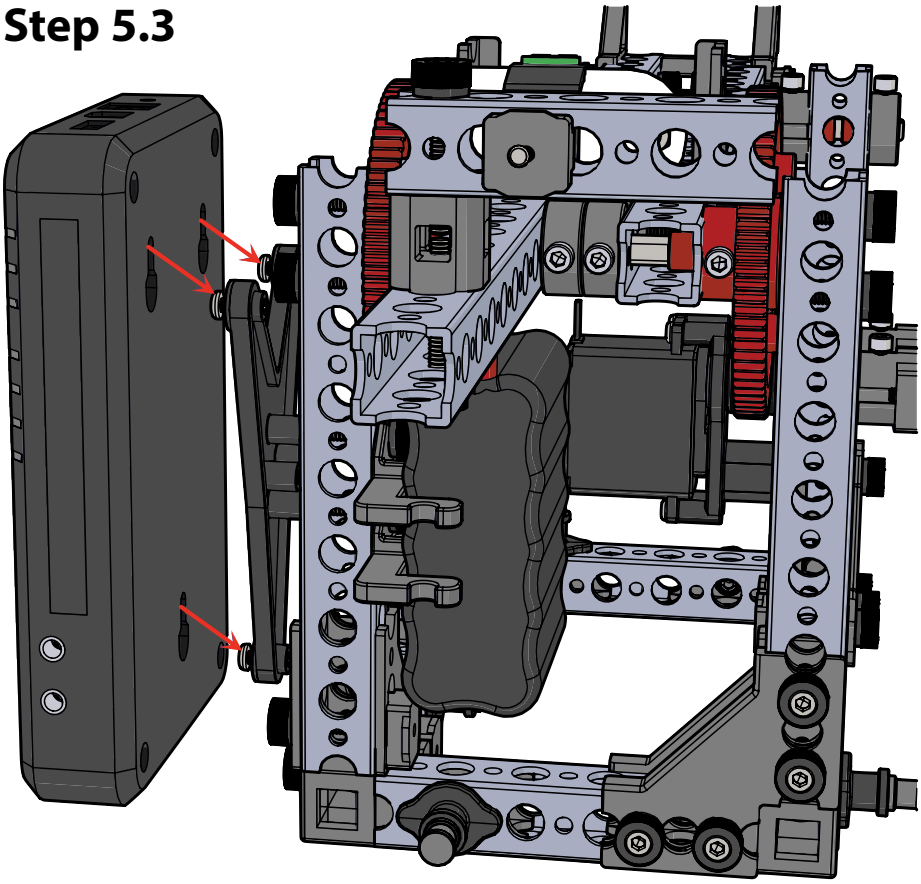
Step 5.1



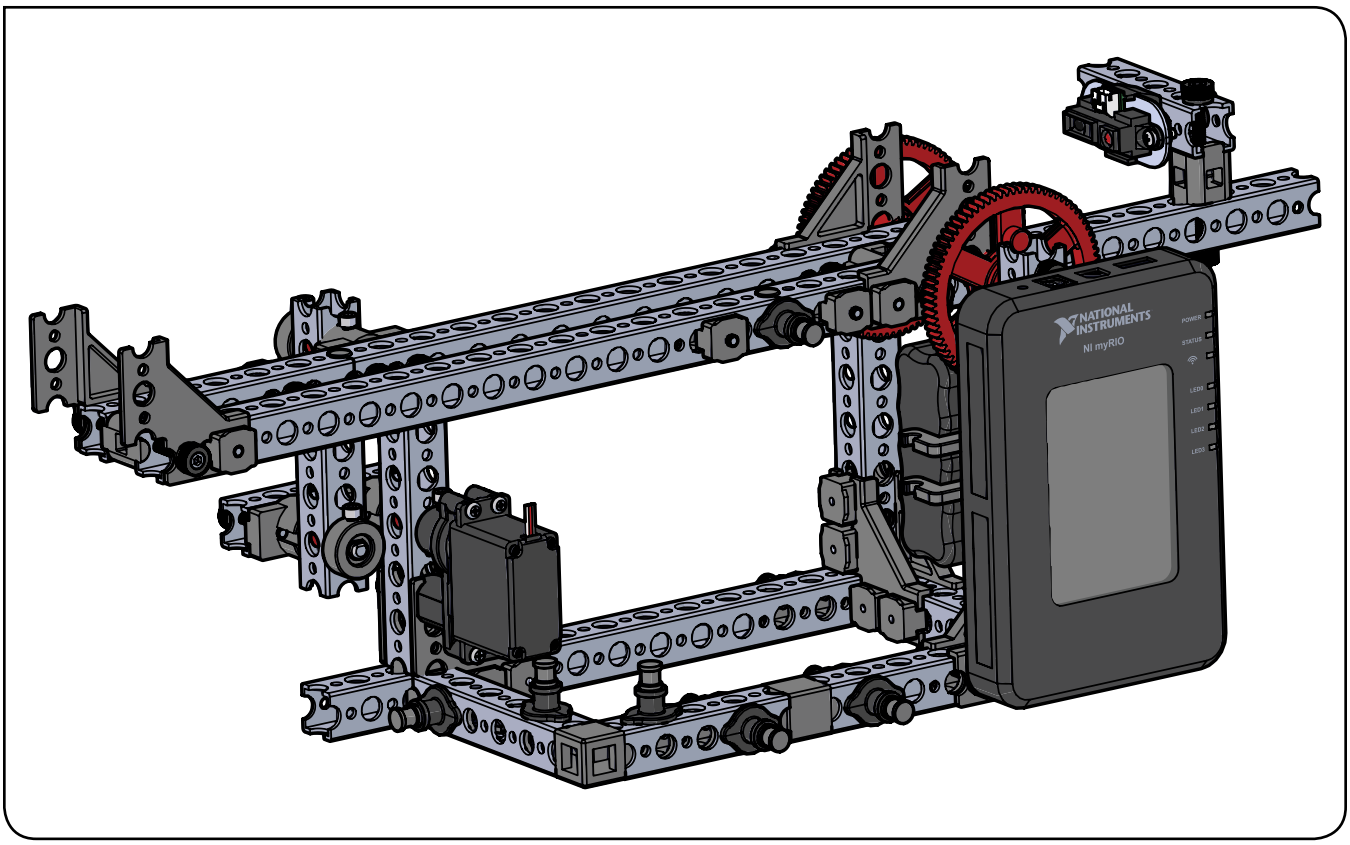
Step 5.2



Step 5.3

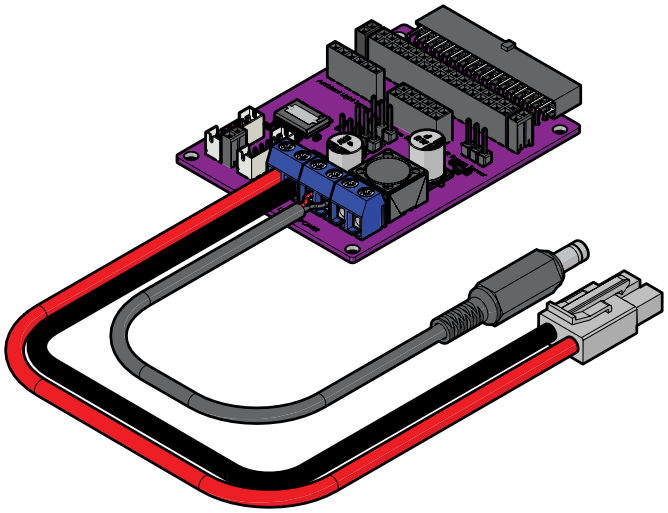


Finished assembly should look like this.

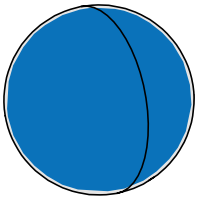


Step 6

Parts Needed

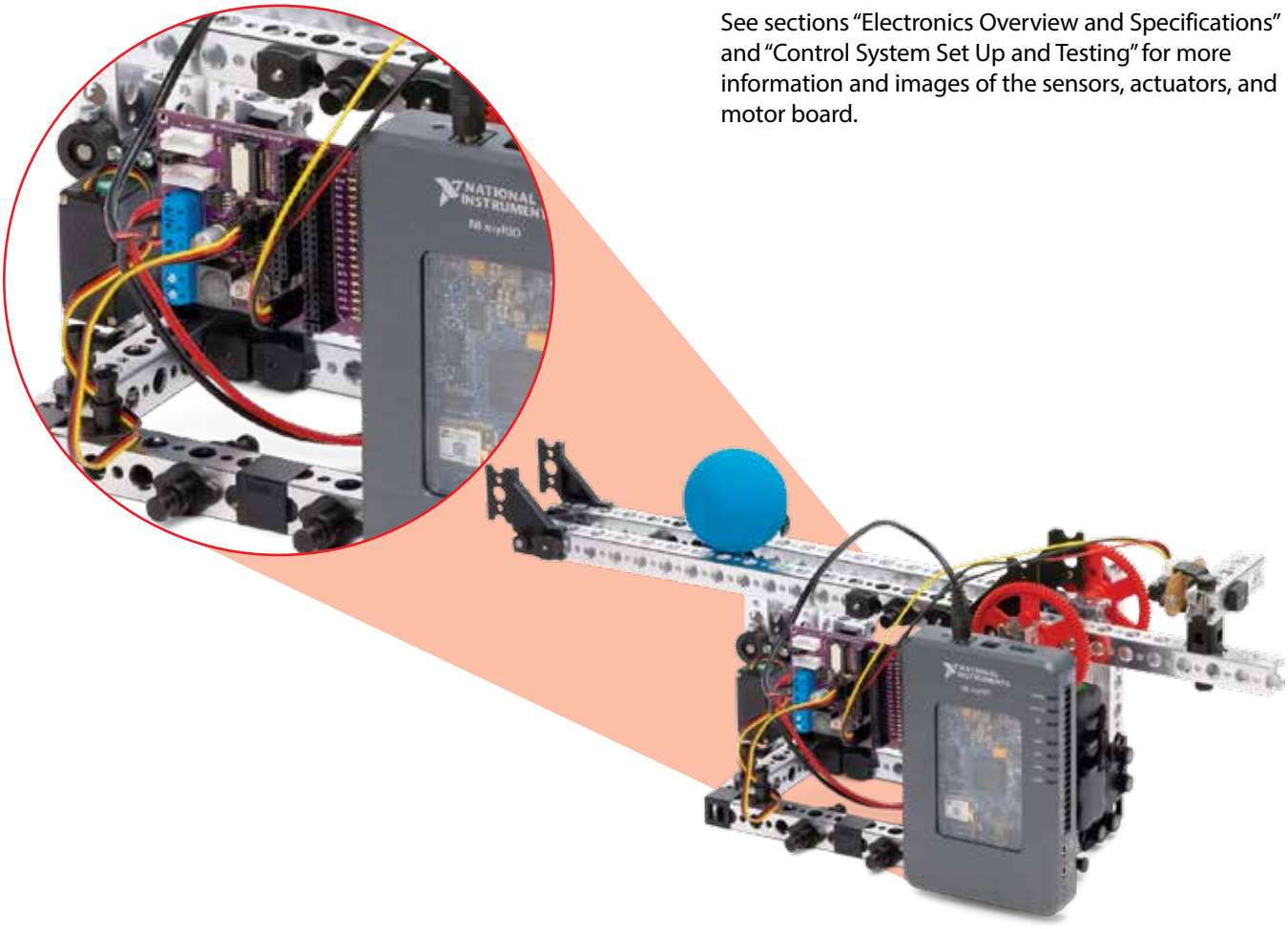


1x
myRIO Motor Controller Board 41136
with DC Power Plug and Battery Cable attached



1x
Blue Racquetball 43121

Finished Assembly



See sections "Electronics Overview and Specifications" and "Control System Set Up and Testing" for more information and images of the sensors, actuators, and motor board.

Step 6.1 Connect the motor board to myRIO MXP A

Step 6.2 Connect the servo cable to the motor board port labeled "Servo 0." Note that the cable should be oriented so that the red wire is lined up with the "+" and the black wire is lined up with the "-"

Step 6.3 Connect the IR Sensor cable to the motor board port labeled "IR Range Sensor." Note that the cable should be oriented so that the yellow cable is next to the label "Sig" and the black cable is next to the label "-"

Step 6.4 Use the battery cable to connect from the T1 port of the motor board (red cable to +, black cable to -) to the battery cord

Step 6.5 Use the DC power cable to connect from the T2 port of the motor board (red cable to +, black cable to -) to the myRIO power port

See the Setup/Hookup Tips and Getting Started with myRIO sections for more details and images of sensor, motor, and battery connections.

Step 6.6 Place the racquetball on the beam.

Note: It's recommended that you only balance a standard racquetball on the Balancing Arm due to its size, weight, consistency, and other properties. The control system has also been optimized for use with a racquetball.

Step 7

Set up myRIO and run LabVIEW code

7.1. Follow the hardware and software setup instructions in the "Set Up Your Control System" section. Make sure you:

- Connect the motor board to MXP B and connect the sensors and actuators to the motor board
- Connect the system to power
- Install software to your PC (via disk or web installer) and myRIO (via the Getting Started Wizard)

7.2. Connect myRIO to your computer via the USB A to B cable

7.3. Navigate to Balance Arm code using the following path: C:\Users\...\Desktop\National Instruments\ Pitsco Tetrix Prime for NI myRIO\Balance Arm and open "Balance Arm Project"

7.4. Make sure the IP Address in the parenthesis next to the "NI-myRIO" line item is 172.22.11.2. If it is not:

- Right click the NI myRIO line item, and select "Properties"
- Update the IP address and click OK

7.5. Right click the NI myRIO line item and select "Connect."

7.6. Press the run arrow on the upper left corner of the Front Panel

7.7. Every time you run the Balance Arm code, you must calibrate the minimum and maximum edges of the beam to account for variations among IR sensors. To do so:

- Gently roll the ball from one edge of the beam to the other, and then let go
- The myRIO will automatically calibrate the edges as the minimum and maximum positions of the beam
- Click "Lock Min Max"
- If the Min and Max were set improperly, click "Lock Min Max" again to turn off the lock, click "Reset Min Max," and repeat the minimum and maximum calibration process

7.8. If you are running the Balance Arm for the first time, or if the balance arm is working unexpectedly, you may need to set a position offset to account for the variation in how the shaft connector can be placed on the servo shaft. To do so:

- Click the "Enable PID" button to turn PID off
- Manually increase and decrease the offset very slightly until the ball can balance at any position on the beam on its own
- Click the "Enable PID" button to turn the PID back on
- If the Balance Arm performance improved, set the new offset value to default:

Step 7 continued

- Stop the VI by pressing the "Stop" button below the Min/Max calibration menu
- Right click the edge of the "Offset" control, select "Data Operations", and select "Make Current Value default"
- Save the VI

7.9. Use the slider control on the left side to move the position setpoint to 20 percent of the arm, and notice:

- In the graph, the blue line indicates the setpoint, and the red line indicates the current position of the ball
- After the setpoint is set, the ball will oscillate back and forth across the set point until it finally settles

7.10. Gently roll the ball away from the set position, let go, and notice the system move the ball back to the original set point

7.11. Move the set point to other locations, allow the ball to settle, and notice:

- The setpoint gradually changes to improve stability
- As the ball moves further away from the IR range sensor, the control system is less stable due to IR range sensor noise

Step 8: Troubleshooting

If your Balance Arm is unstable:

8.1. Make sure the table you're working on is flat and stable

8.2. Make sure both myRIO and the battery are secured to the system, as their weight assist with stability

8.3. Check that the IR sensor is centered on the rail

8.4. Turn the ball so that its seam is parallel to the rails

8.5. Redo steps 7.7 and 7.8

Next Steps

- Learn about how the Balance Arm relates to control theory in the next section
- Get some ideas for expanding the Balance Arm on the first page of this section

Connections to Controls Concepts: Balance Arm

Prof. R. H. Bishop

University of South Florida

The balancing arm assembly provides insight into two feedback control system concepts: (i) closed-loop stability and (ii) transient and steady-state performance—the ability of closed-loop control to provide a quick response and to reduce steady-state tracking errors.

Plant	Balance arm
Sensor	Infrared (IR) range finder
Actuator	Servo motor
Performance	Command following in the presence of disturbances and stability
Design objectives	Tune the control system by adjusting PID gain constants & stabilize the ball at the given set-point
Reference inputs	Set-points

The balancing arm and one standard servomotor are the plant and actuator, respectively. The sensor is an IR range sensor and provides the feedback signal on the location of the ball relative to the sensor from which we can compute the offset from the set-point. The commands enter the control system via the host computer through front panel commands sent over the USB Ethernet connection to the myRIO, which hosts the PID controller code.

Closed-loop stability

A stable feedback control system displays a bounded response to a bounded input. For example, if the balancing arm feedback control system is stable, then when we place the ball at the setpoint and gently push it, the actuated arm will rotate according to commands from the controller to return the ball to the setpoint. If the system is unstable, when the ball is pushed, it will not return to the setpoint, but instead will continue to deviate farther and farther away from the setpoint until likely coming to rest against the stop. If the system is marginally stable, the ball will roll away and come rest at some distance from the setpoint in a controlled fashion. Stability is a key objective in the design of feedback control systems. Closed-loop feedback systems that are unstable are, in general, particularly undesirable.

Disturbance rejection and performance

The gentle push can be viewed as an external impulsive disturbance. The stable closed-loop balancing arm control system demonstrates an ability to reject these disturbances by returning the ball back to the setpoint. A key element of the control design process is not only to achieve closed-loop stability, but also to have the ball return to the desired setpoint with a specified performance. In other words, we want the feedback control system to meet our design specifications that typically include settling time, percent overshoot, time to peak, time to rise, and steady-state errors in response to a specific set of test input signals, such as impulses, steps,

and ramps. We definitely want to control the transient response and steady-state response of the ball motion. The transient response is the ball motion that occurs before steady-state is reached. The controller gains must be adjusted until the transient response is satisfactory—this is the design process. Note that a stable system can have a non-zero steady state error—this is different from marginal stability. With the PID controller, the ball will return to the setpoint with zero steady-state error after an impulsive external disturbance.

Design of the PID controller

The steady-state error of the closed-loop system is determined by the structure of the controller. The integral term of the PID controller increases the system type of the loop transfer function, represented by L(s)=G(s)Gc(s)H(s), by one which leads to improved steady-state error response. In fact, the PID controller is a particularly important structure in practice. The transient performance of the closed-loop system is determined by the PID controller gains. The PID controller has three gains: the proportional gain, denoted by Kp, the derivative gain, denoted by KD, and the integral gain, denoted by KI. The effect of each of these gains on the transient response to an external disturbance can be described conceptually as follows:

Increasing Kp leads to an increase in the percent overshoot, has minimal impact on the settling time, and reduces the steady-state errors;

Increasing KI leads to an increase in the percent overshoot and the settling time, but reduces the steady-state errors, and

Increasing KD leads to a decrease in the percent overshoot and the settling time.

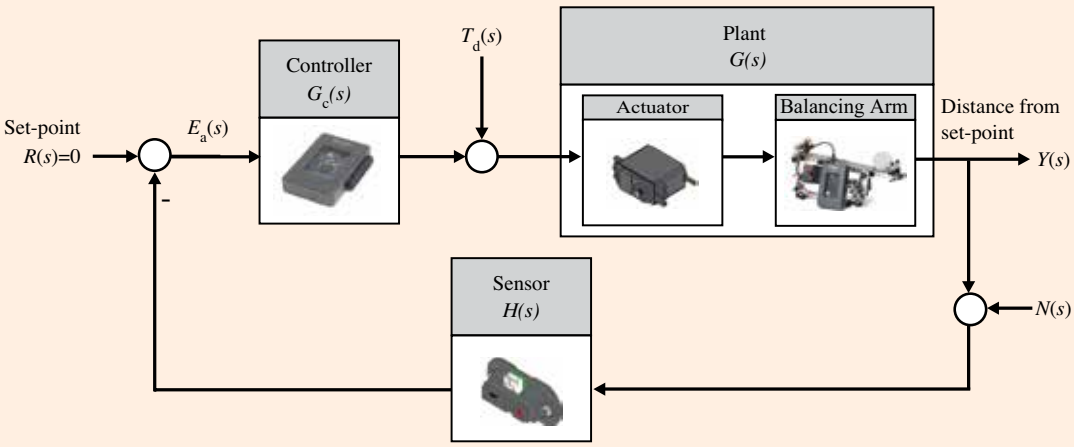
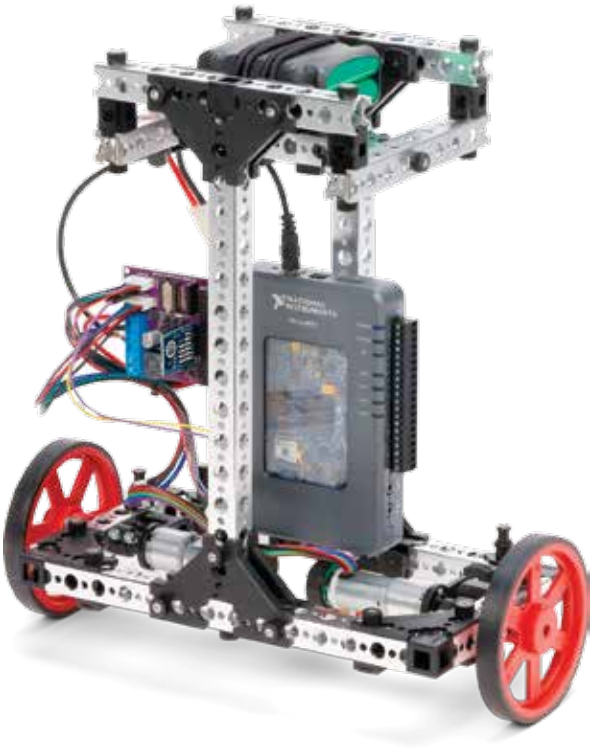


Figure 3 Closed-loop feedback control to a set-point.

Note: Control concepts described in detail in Modern Control Systems, by R. C. Dorf & R. H. Bishop, 13th Ed., Pearson Education, Inc., 2017.

Self Balancing Robot



Overview

The Self-Balancing Robot is a complex closed-loop control system that autonomously balances itself in place. It collects feedback from multiple sensors, including the onboard accelerometer of the myRIO, a gyroscope, and encoders built into both motors. It uses a complementary filter and a PD (proportionaldifferential) controller in LabVIEW to stand upright.

Base Functionality

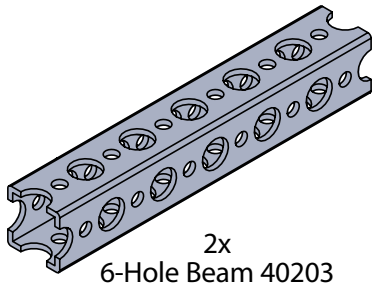
- The wheels are rotated by DC motors, which operate independently and receive PWM data to control their speeds
- The encoders built into each motor measure relative position, and communicate with myRIO using specialized encoder digital lines
- The onboard three-axis accelerometer of myRIO measures static and dynamic acceleration
- The gyroscope measures rotational velocity, and communicates with myRIO using the I2C communication protocol
- A complementary filter is used on the accelerometer and gyroscope data to eliminate the high frequency noise of the accelerometer and the low frequency noise of the gyroscope
- A PD controller is used to control the motor positions relative to one-another
- myRIO connects to the host PC via WiFi
- The LabVIEW code can either be run from the host PC or deployed as a start-up executable
- The robot is enabled by pushing the built-in button on myRIO

Expansion and Teaching Options

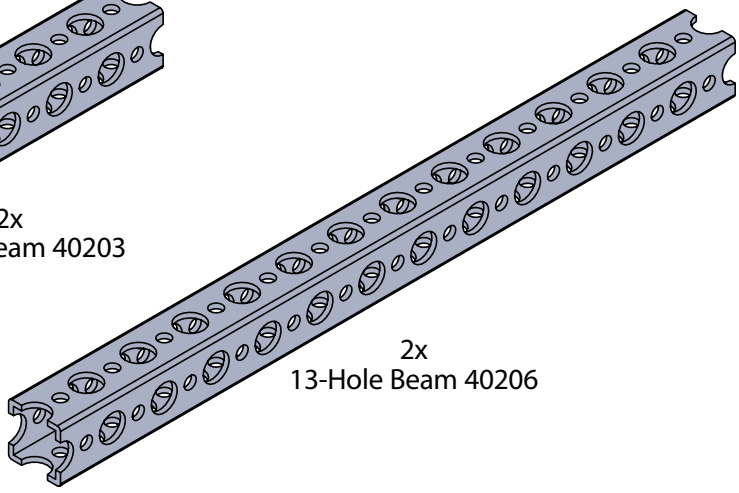
- Learn how the Self-Balancing Robot relates to controls concepts like relative stability, robust stability, and fundamental design tension (See the Connections to Controls Concepts at the end of this section)
- Advanced Challenge: Edit the LabVIEW code of the Self-Balancing Robot to enable it to travel on command

Step 1

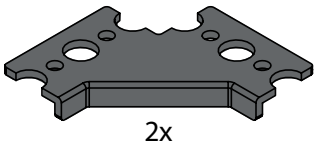
Parts Needed



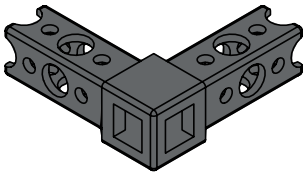
2x
6-Hole Beam 40203



2x
13-Hole Beam 40206



2x
90-Degree Bracket 40208



4x
90-Degree Beam
Connector 40211

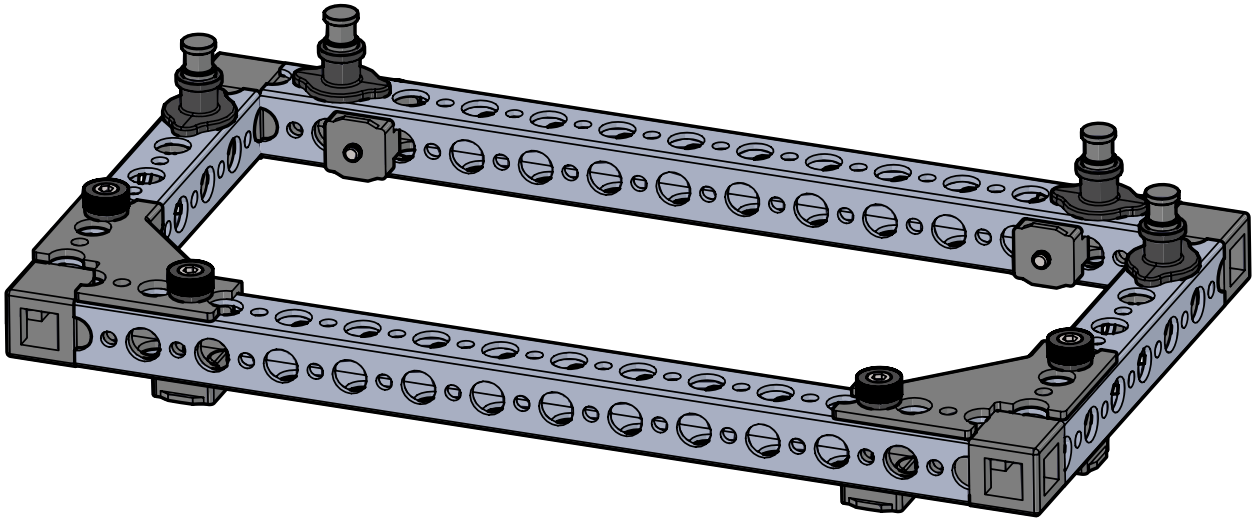
6x
Wing Nut 40221

6x
Thumbscrew 40323

4x
Quick Rivet Peg 40220

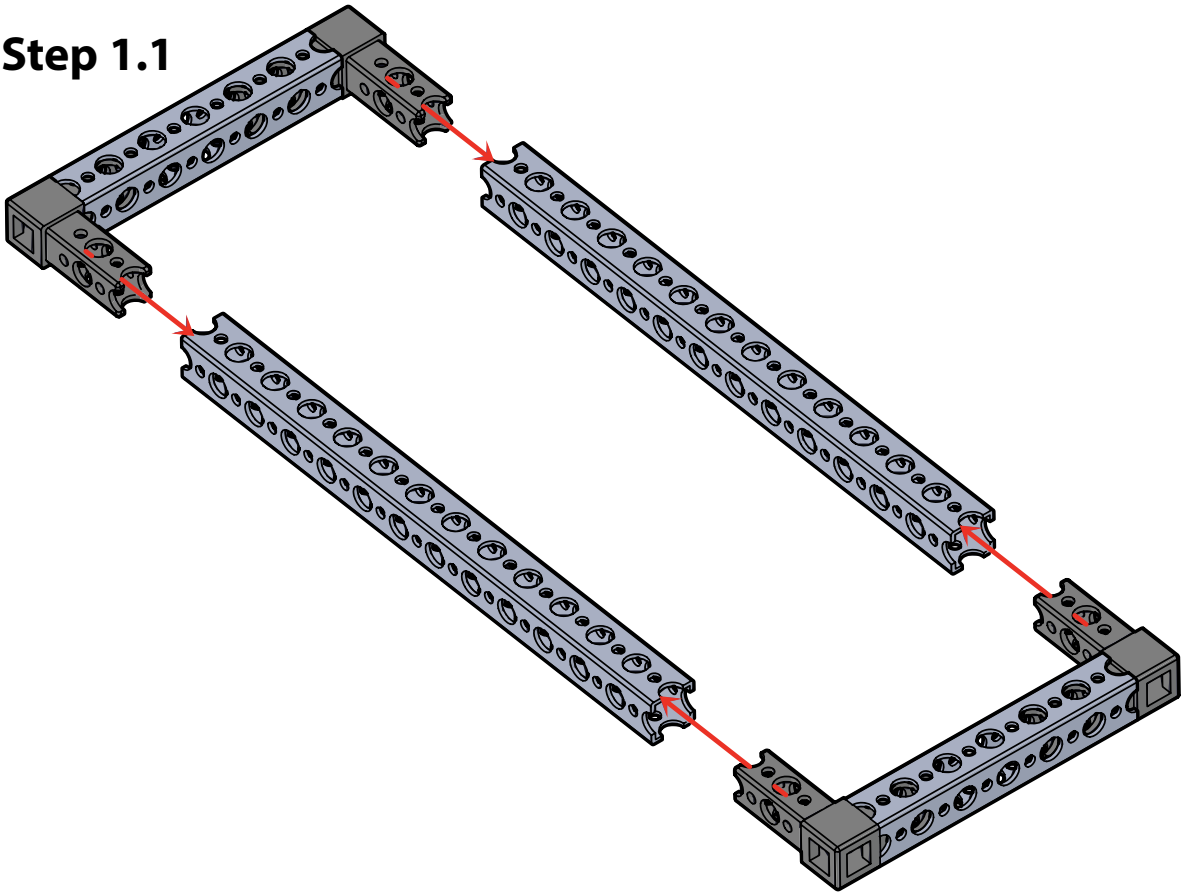
4x
Quick Rivet Connector
40219

Partial assembly should look like this.

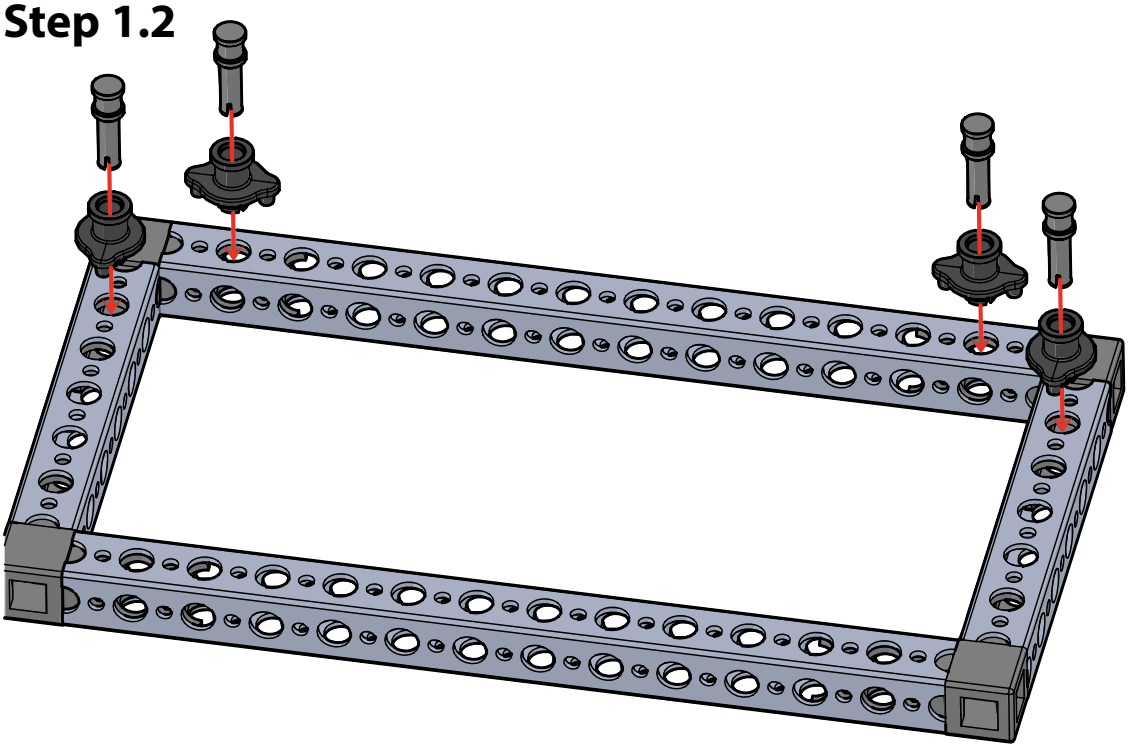


Step 1.0

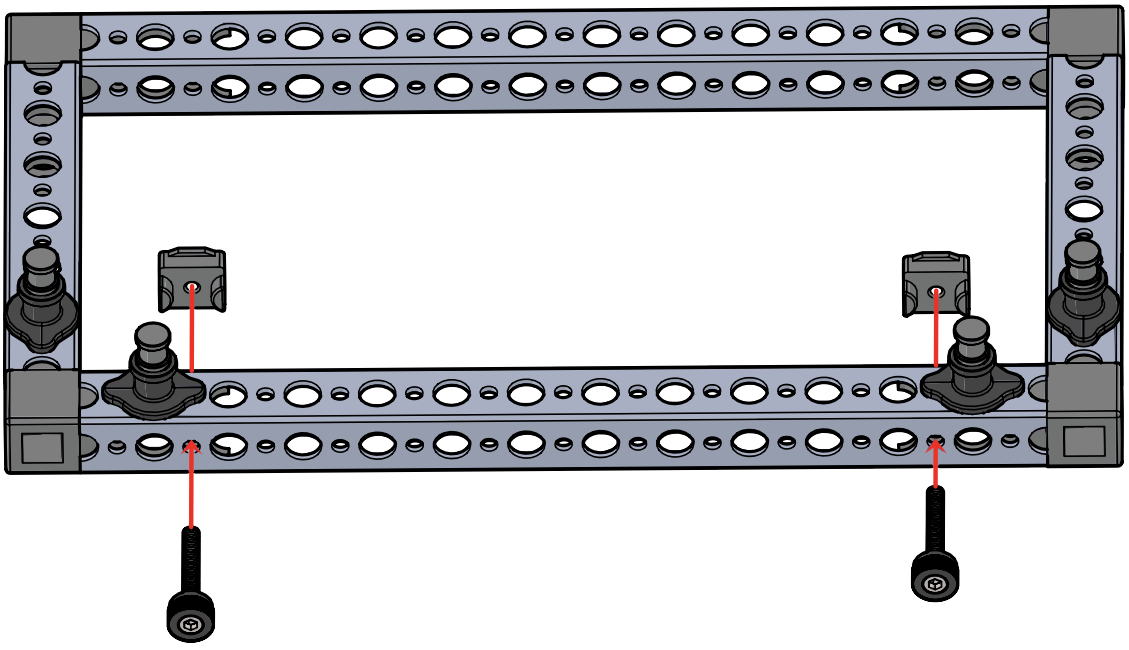
Step 1.1



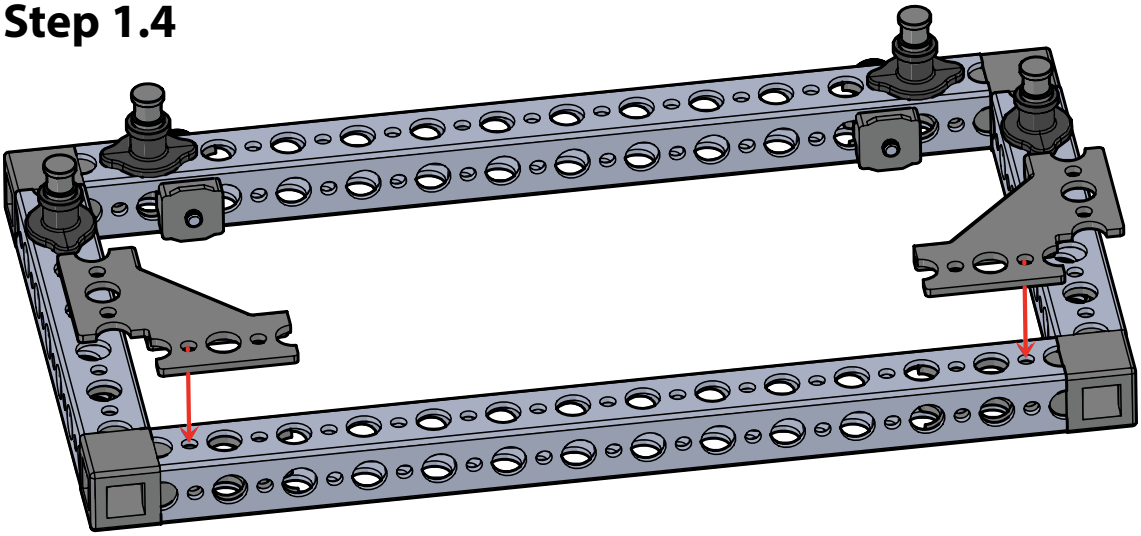
Step 1.2



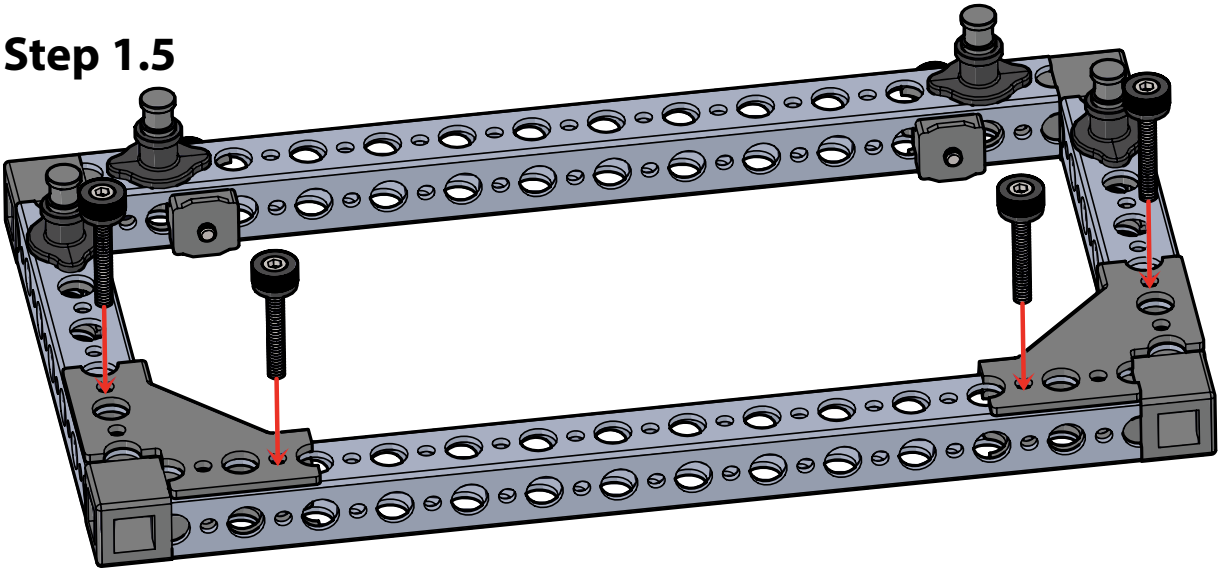
Step 1.3



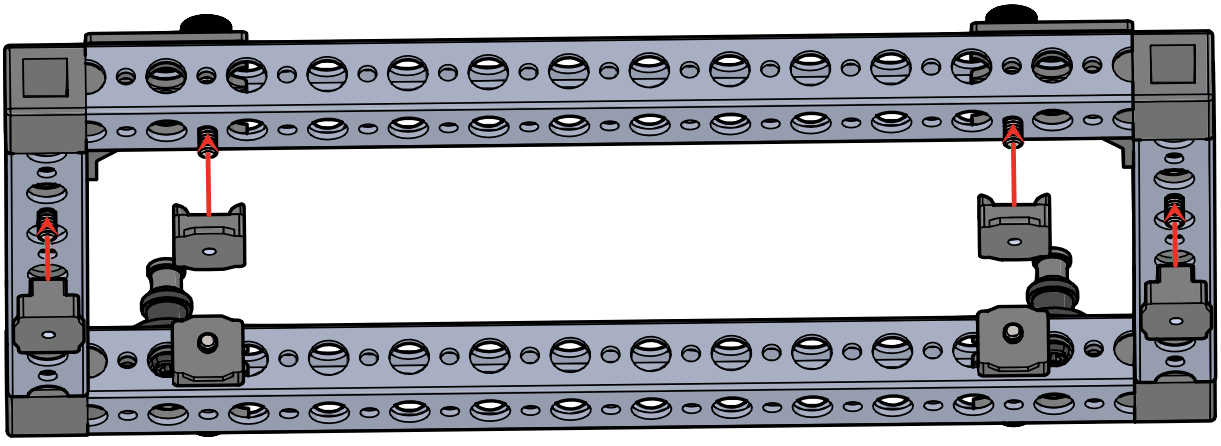
Step 1.4



Step 1.5

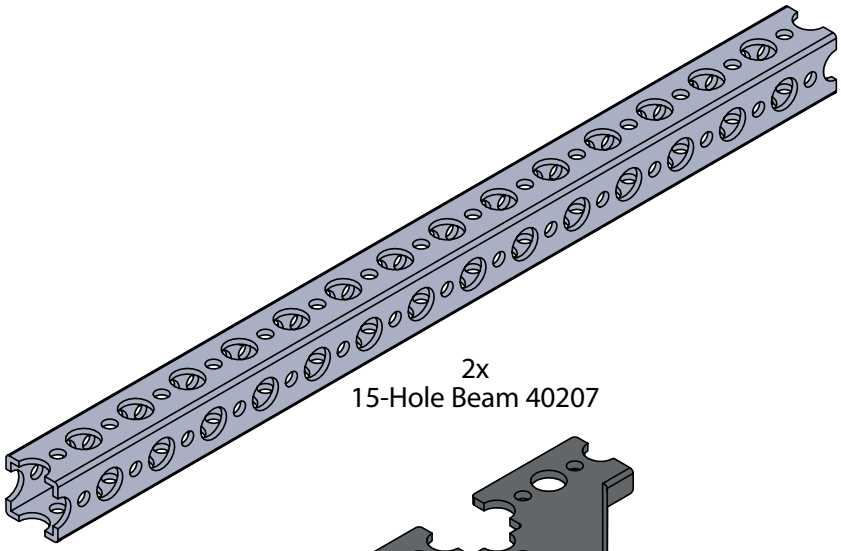


Step 1.6

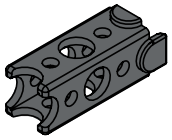


Step 2

Parts Needed



2x
15-Hole Beam 40207



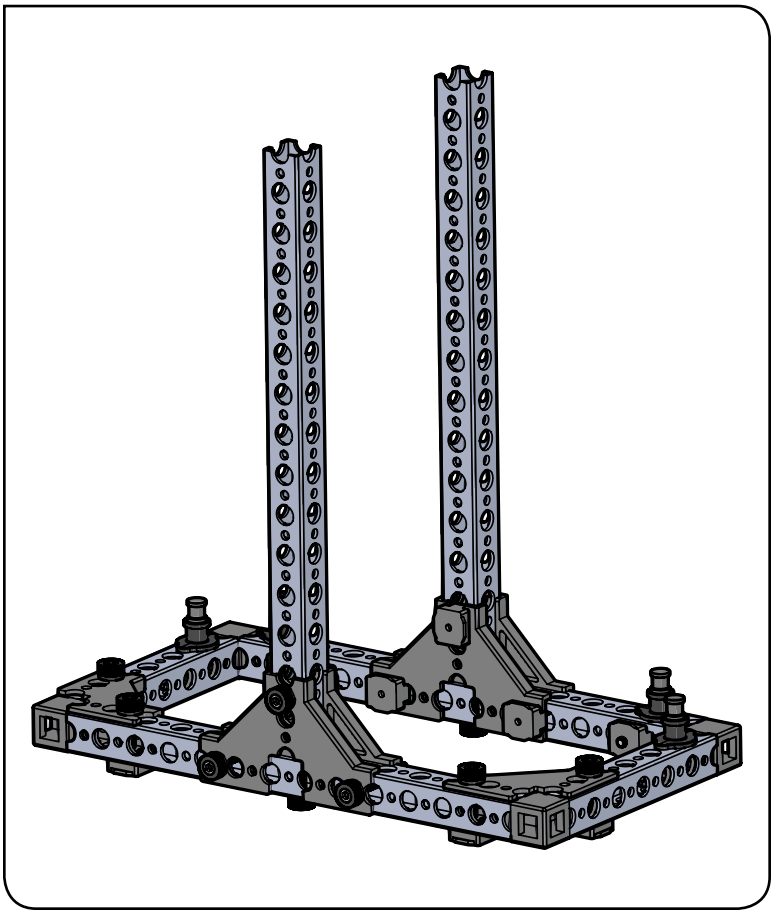
2x
Beam End Connector 40214

4x
Tee Bracket 40210

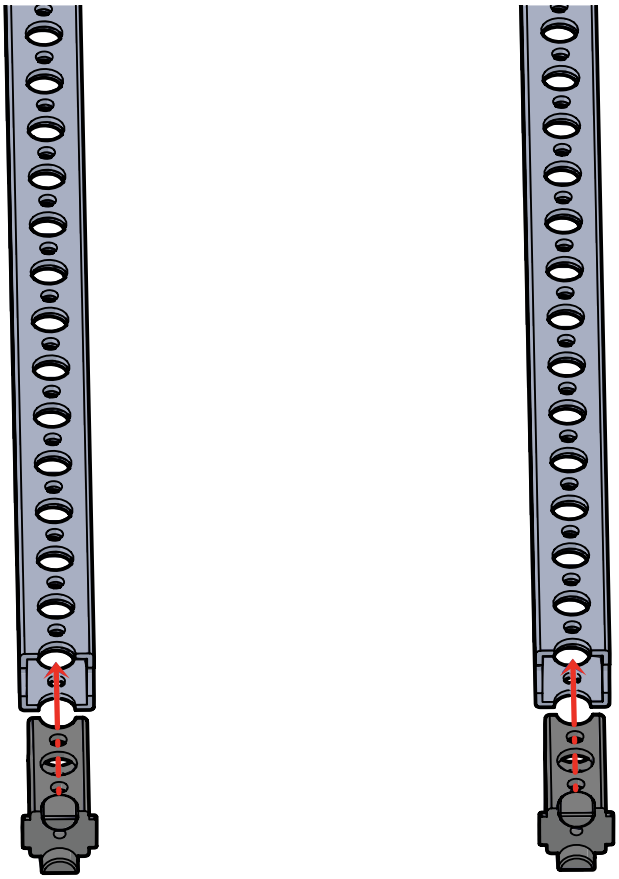
8x
Thumbscrew 40323

6x
Wing Nut 40221

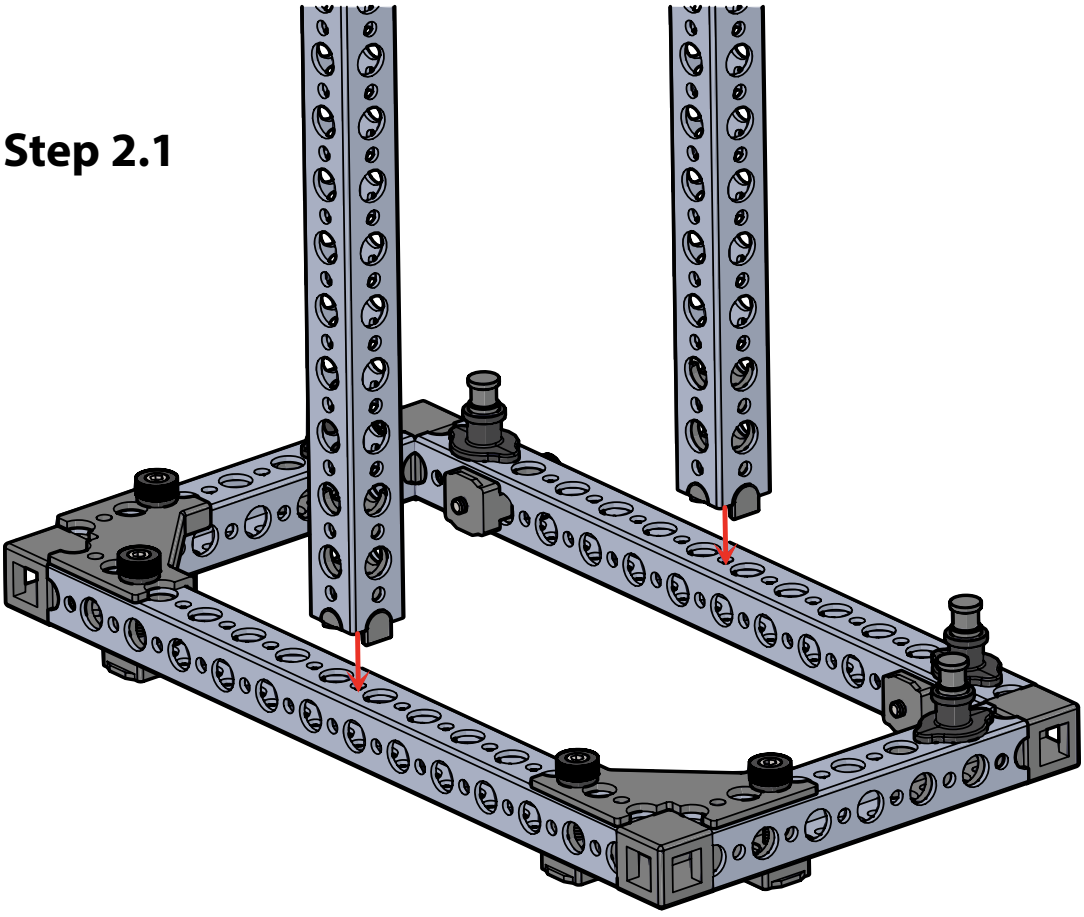
Partial assembly should look like this.



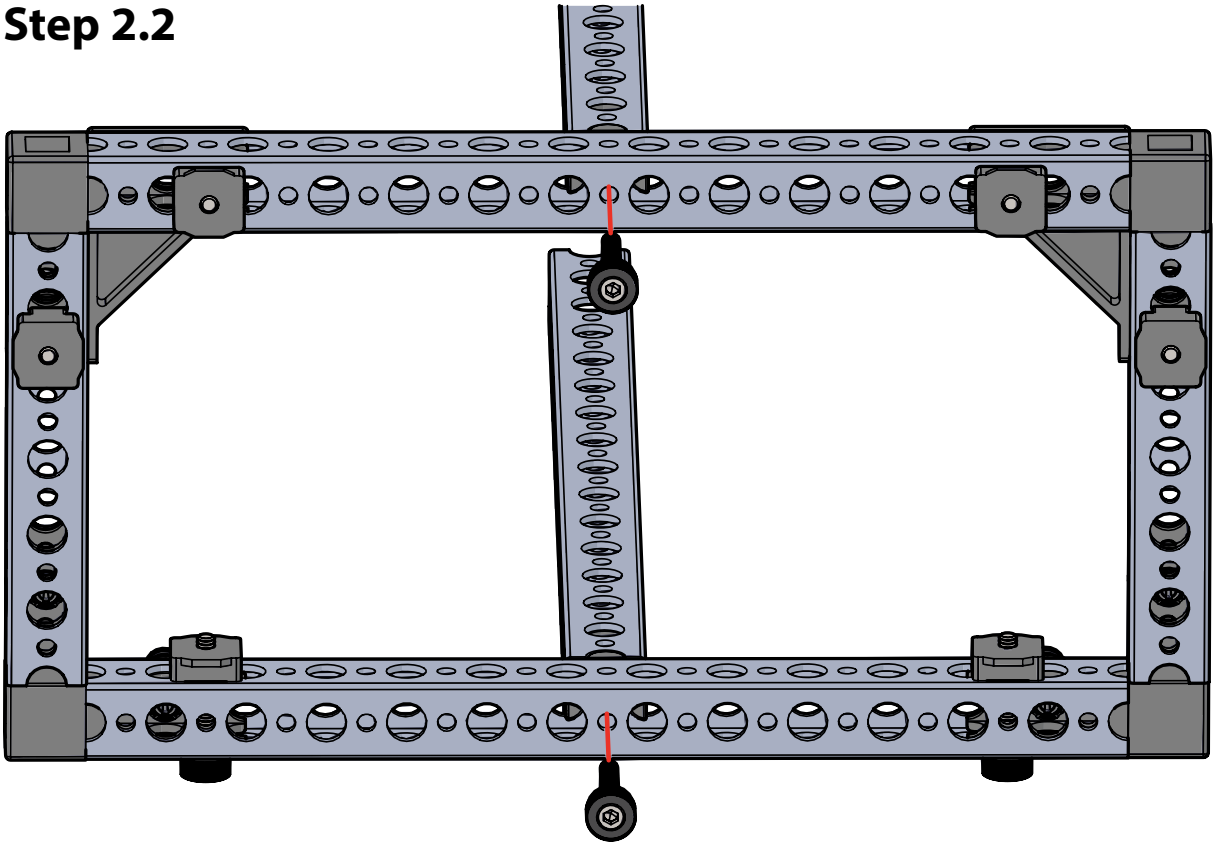
Step 2.0



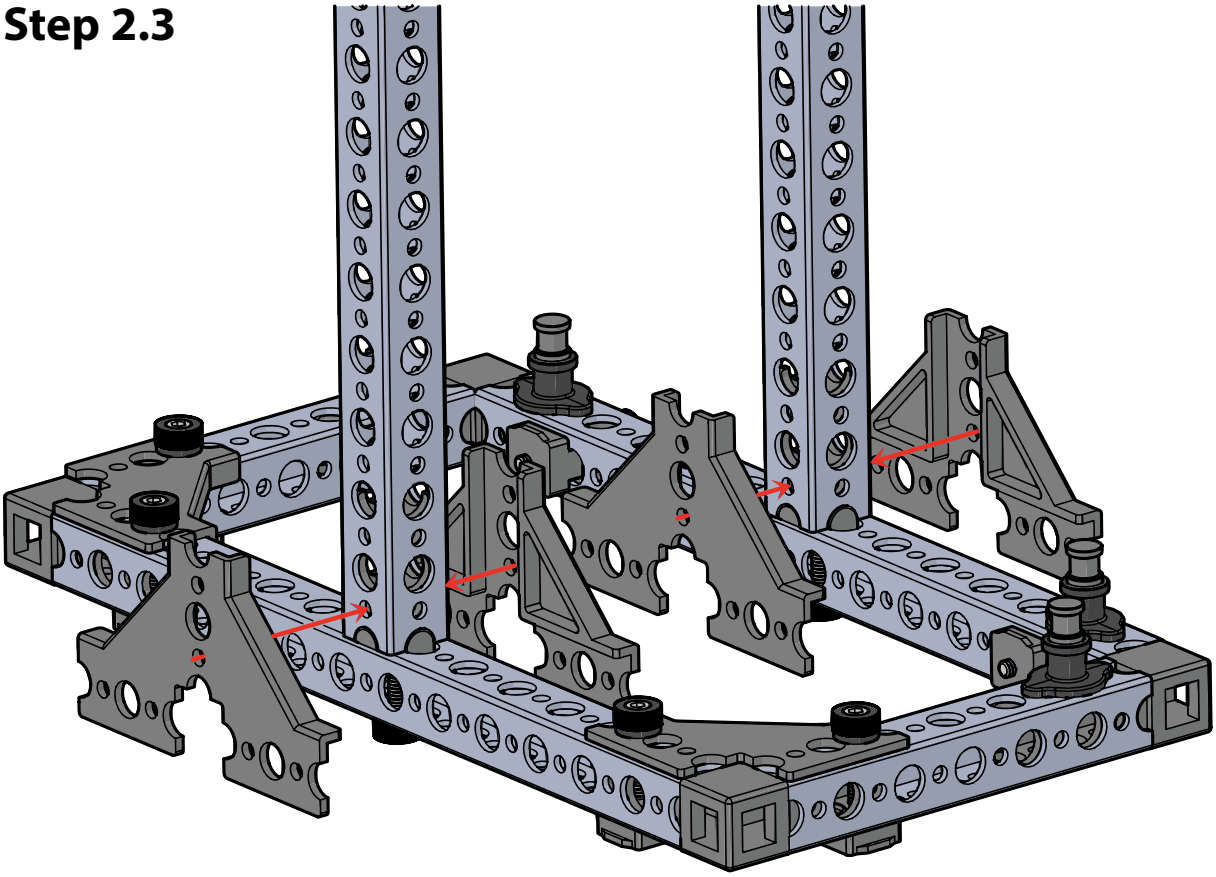
Step 2.1



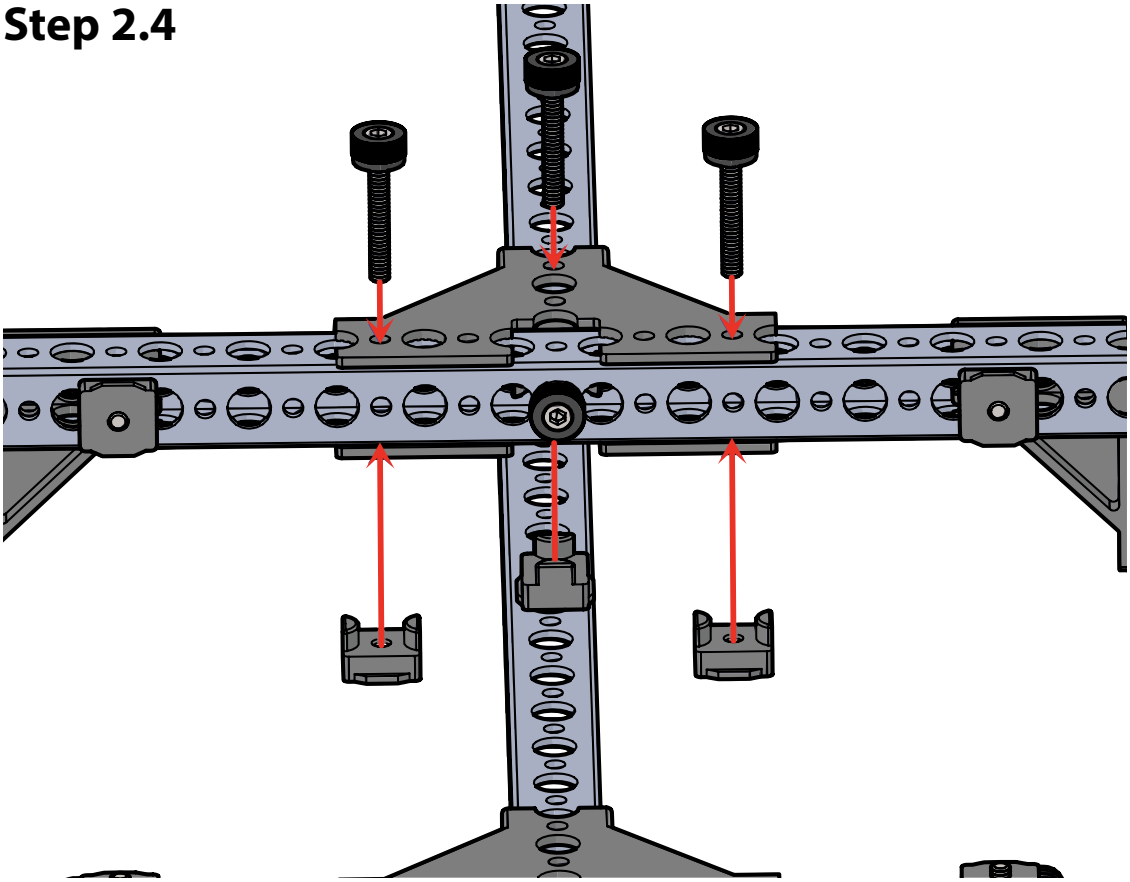
Step 2.2



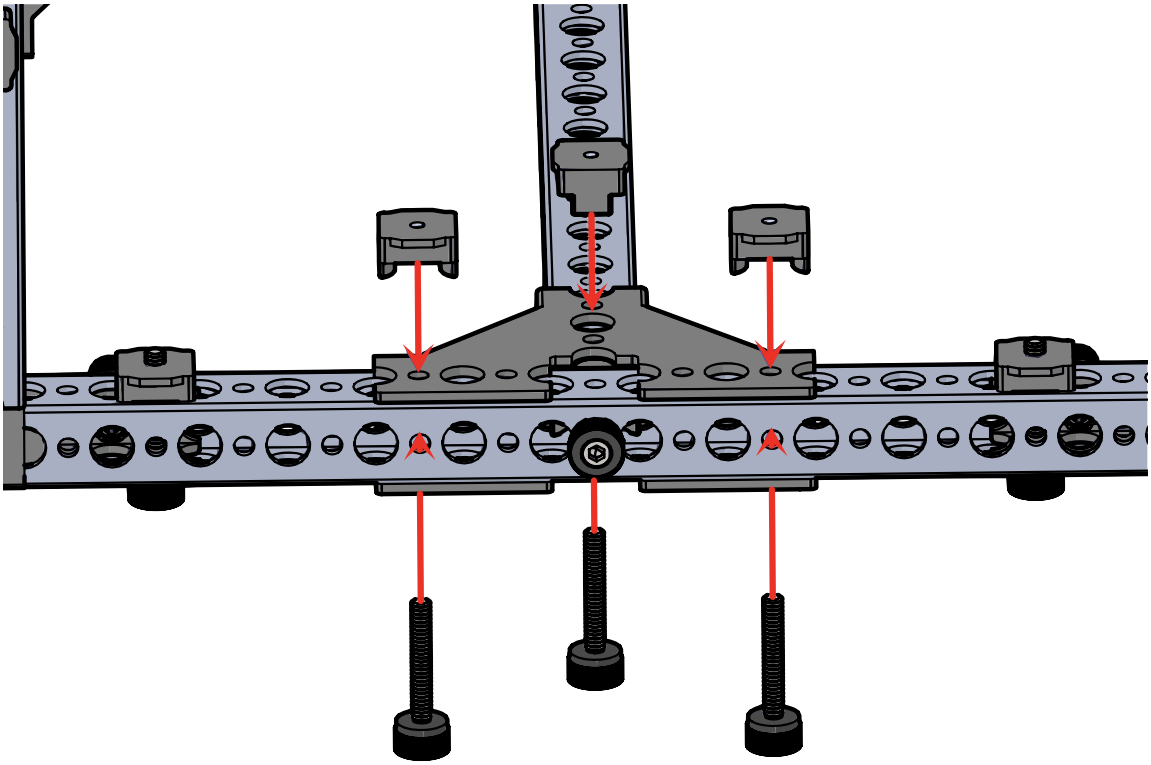
Step 2.3



Step 2.4

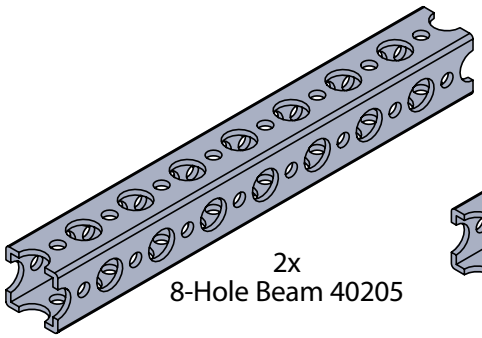


Step 2.5

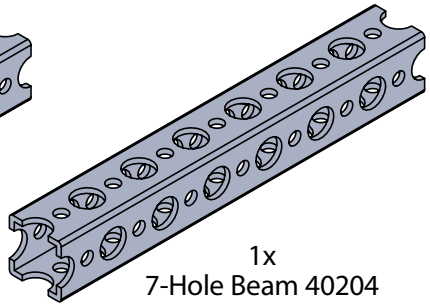


Step 3

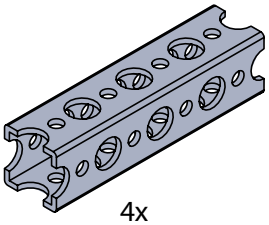
Parts Needed



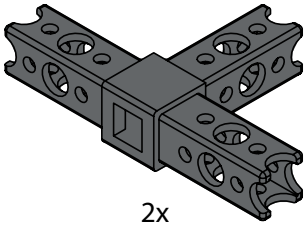
2x
8-Hole Beam 40205



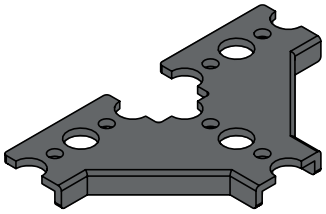
1x
7-Hole Beam 40204



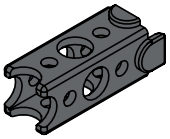
4x
4-Hole Beam 40201



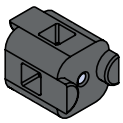
2x
Tee Beam Connector
40213



4x
Tee Bracket 40210



2x
Beam End Connector
40214



4x
90-Degree Cross Block
Connector 40217

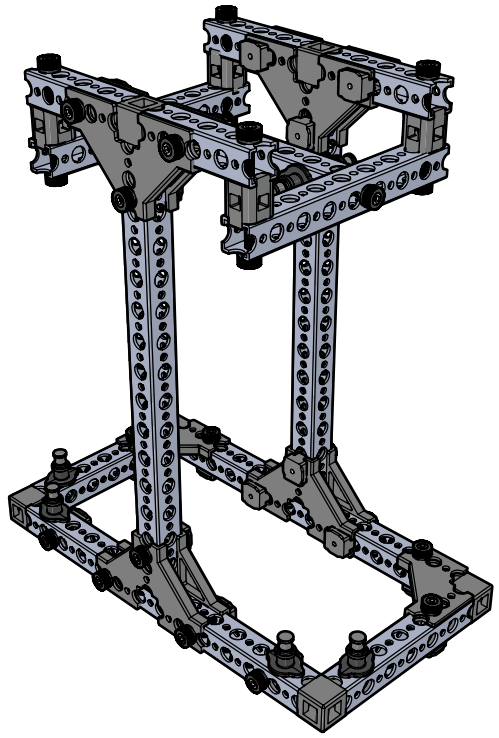
6x
Thumbscrew 40323

6x
Wing Nut 40221

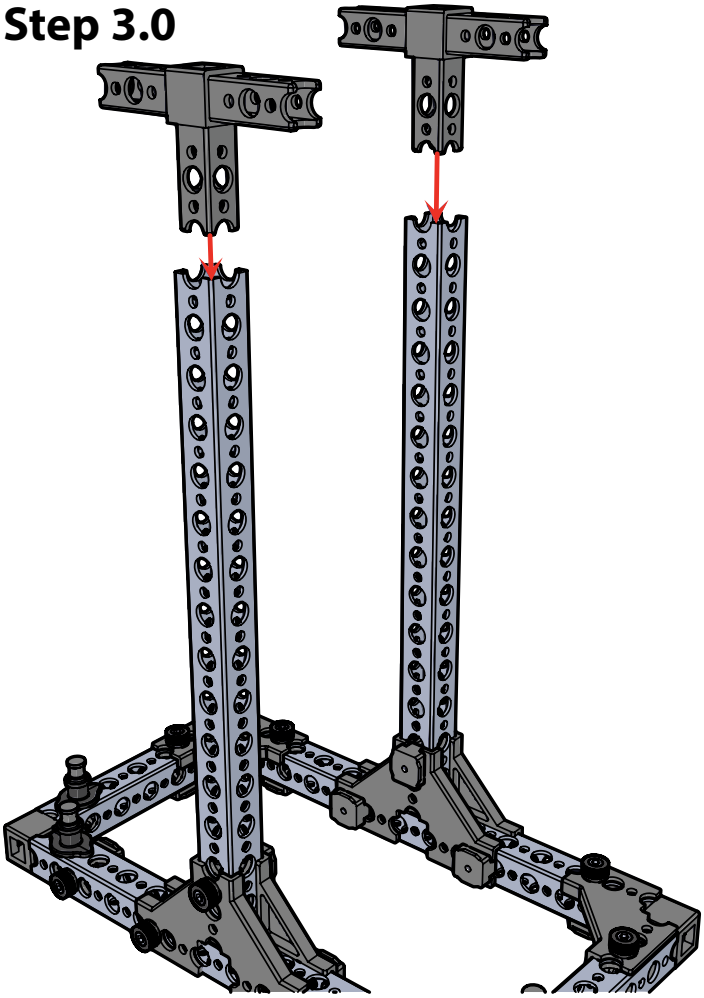
4x
Quick Rivet Peg 40220

4x
Quick Rivet Connector
40219

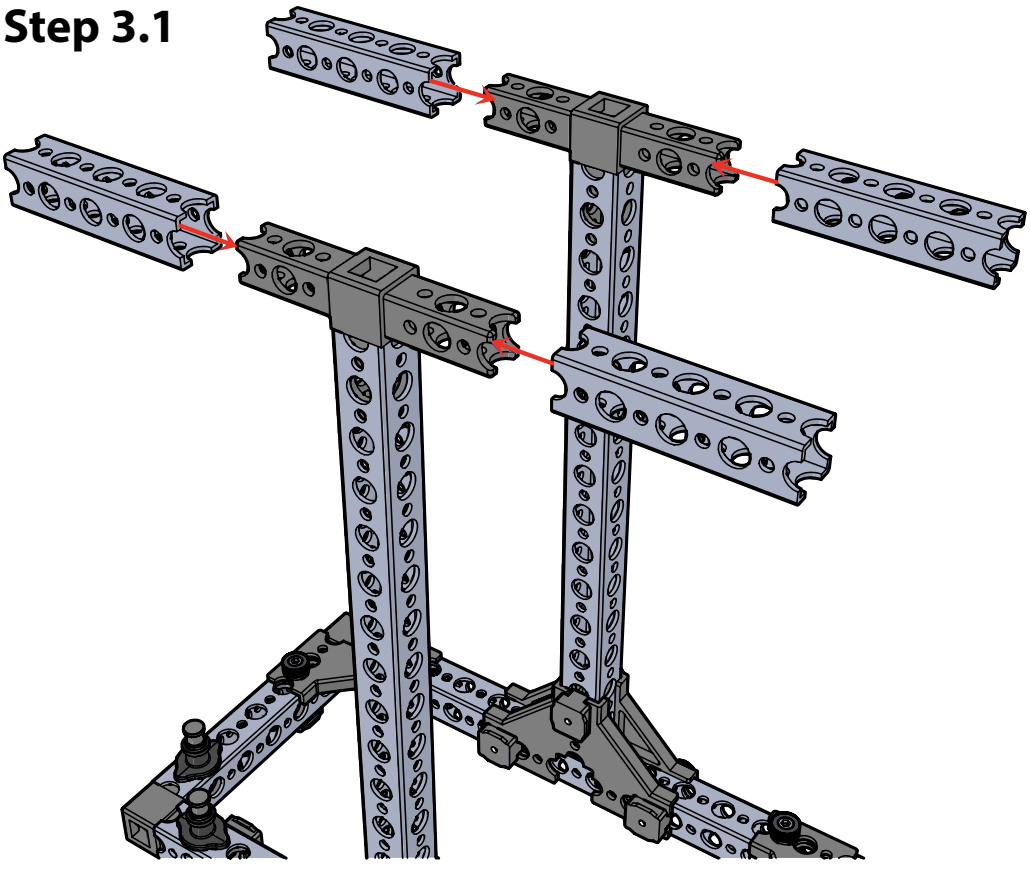
Partial assembly should look like this.



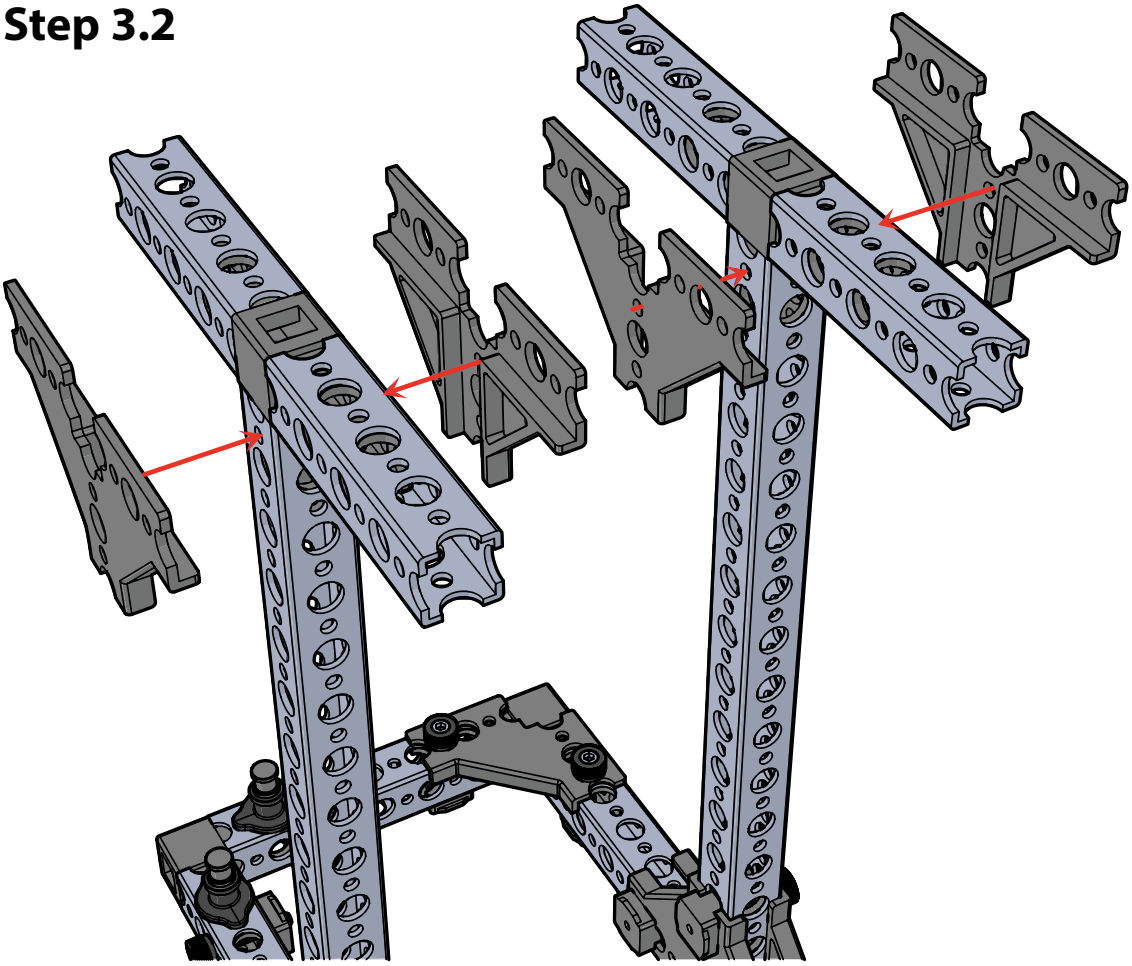
Step 3.0



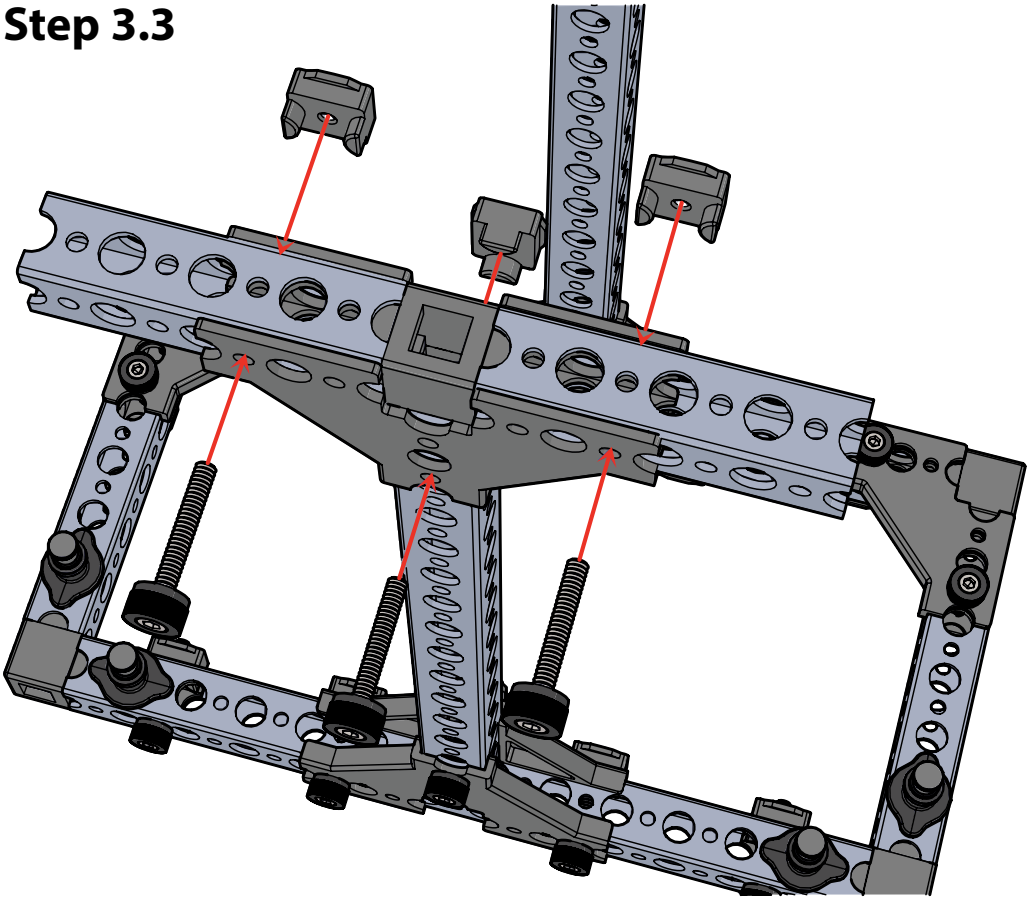
Step 3.1



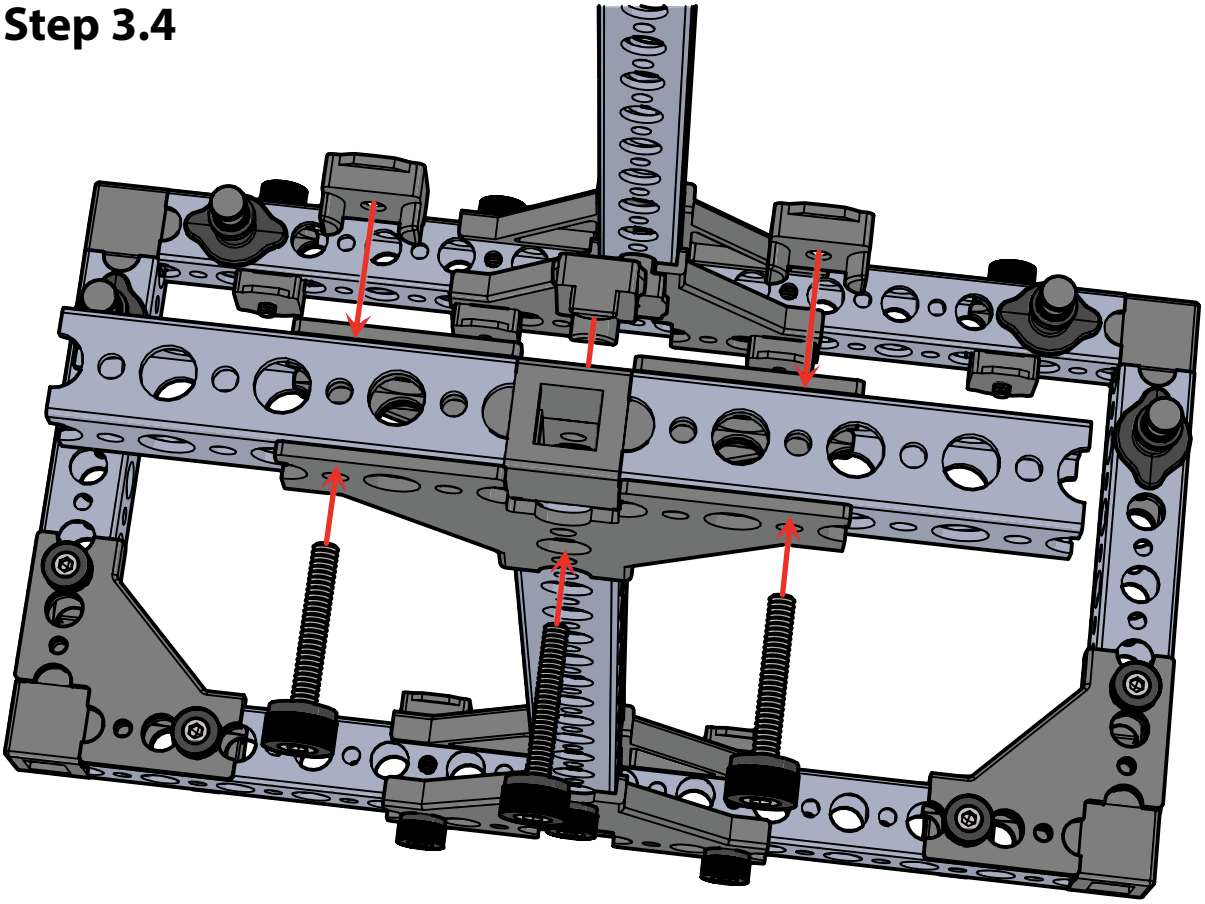
Step 3.2



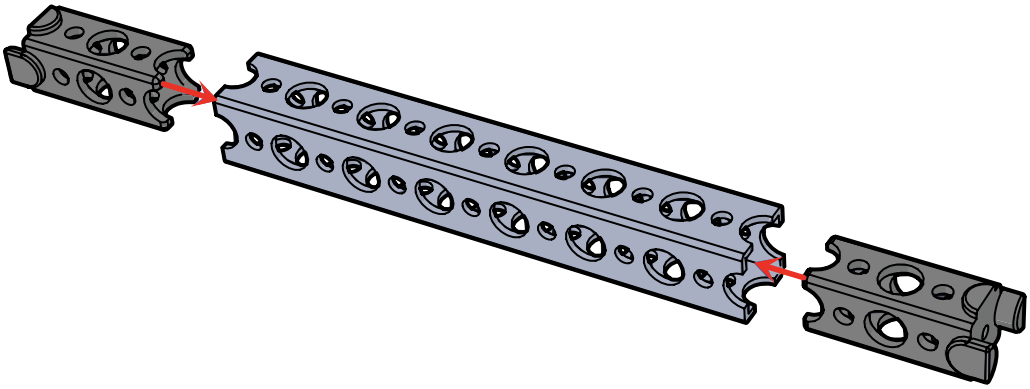
Step 3.3



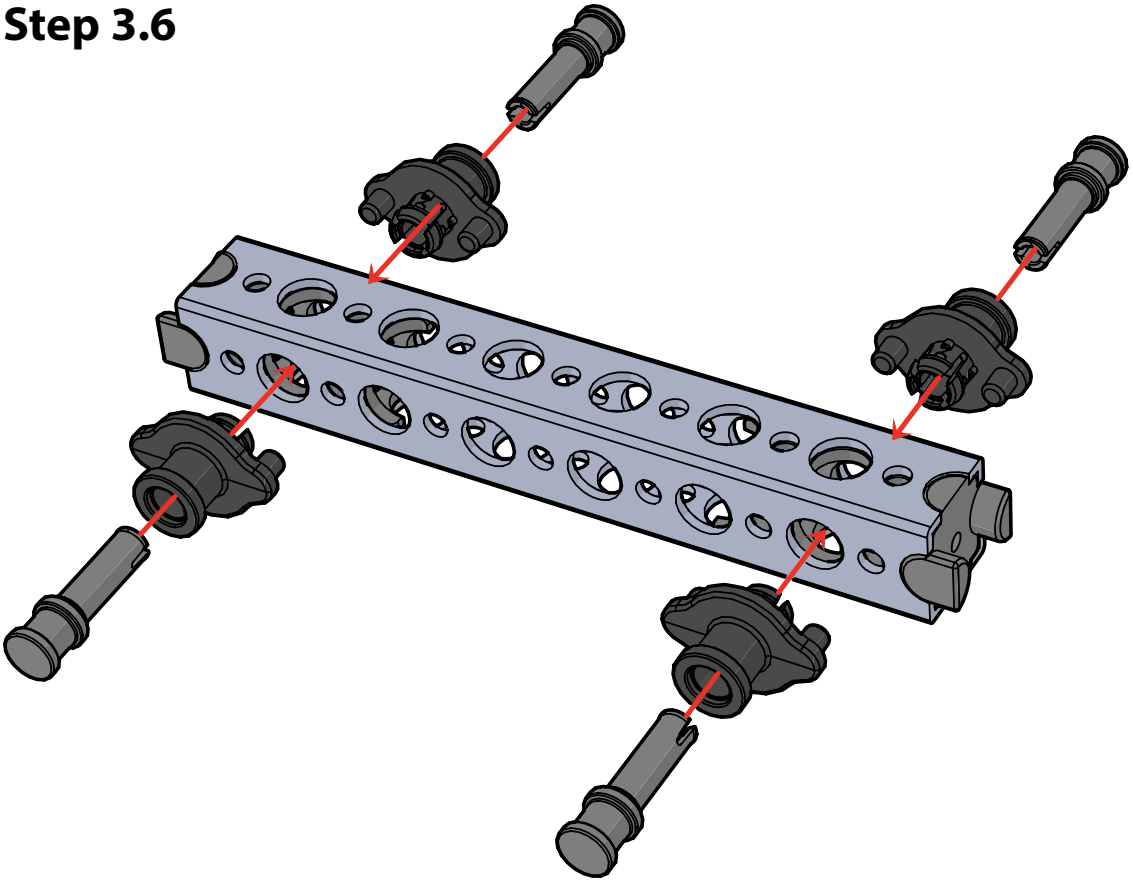
Step 3.4



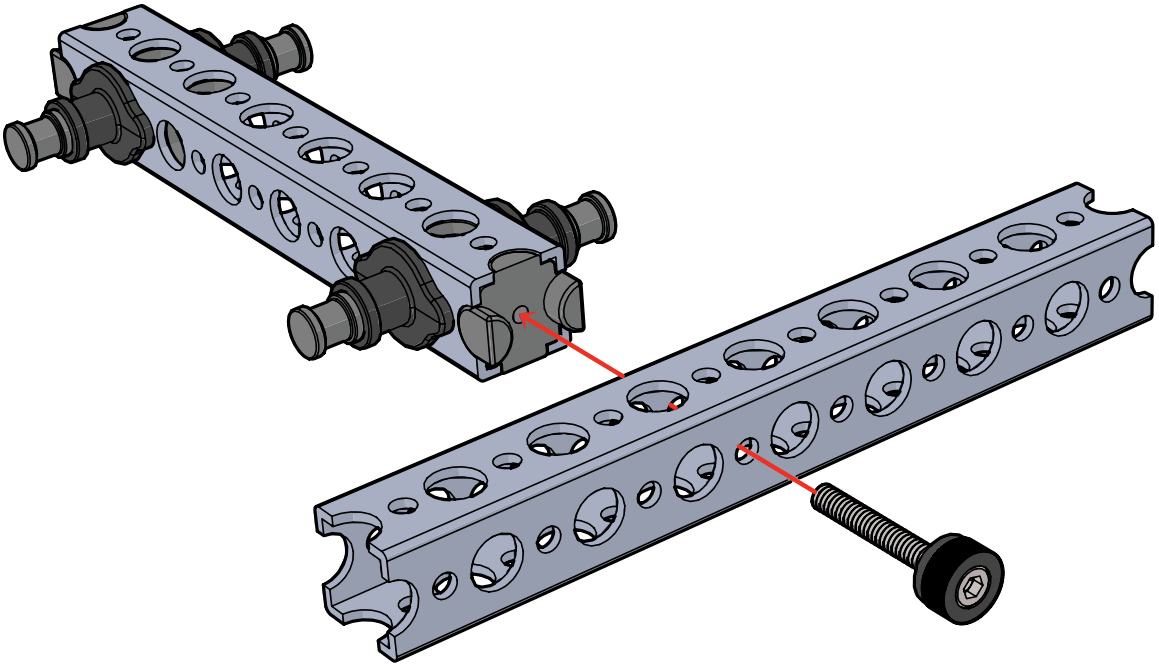
Step 3.5



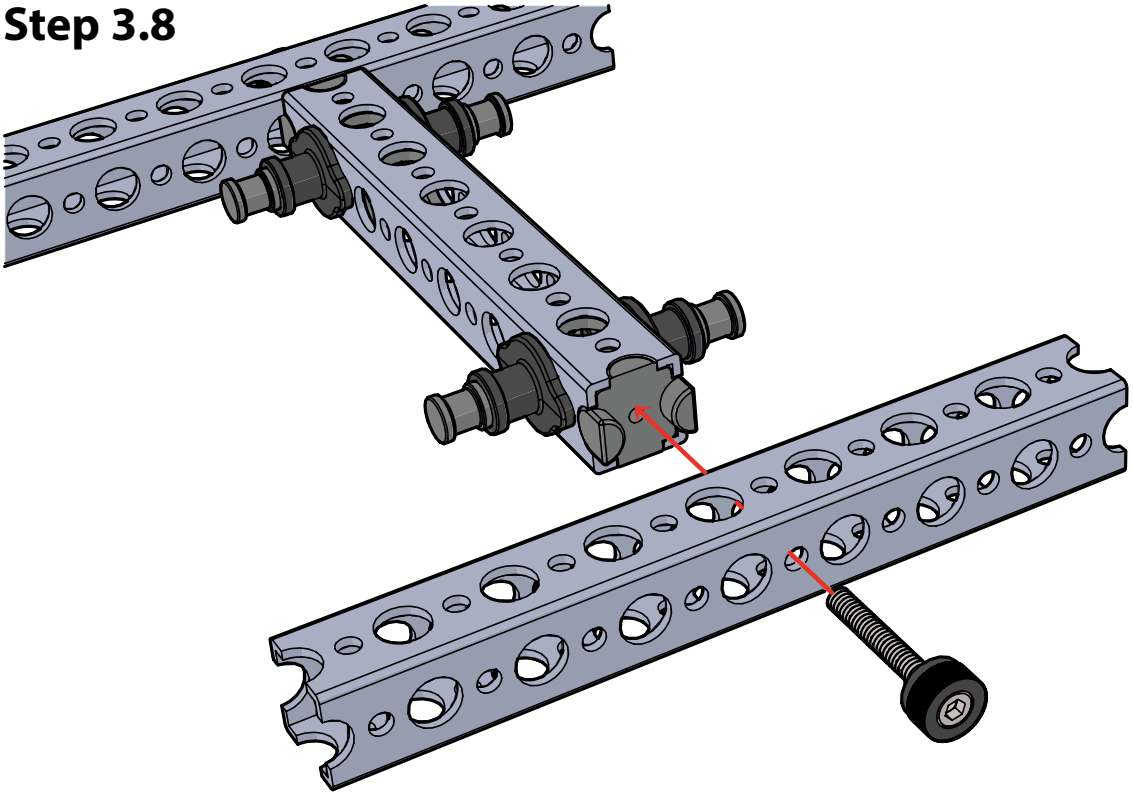
Step 3.6



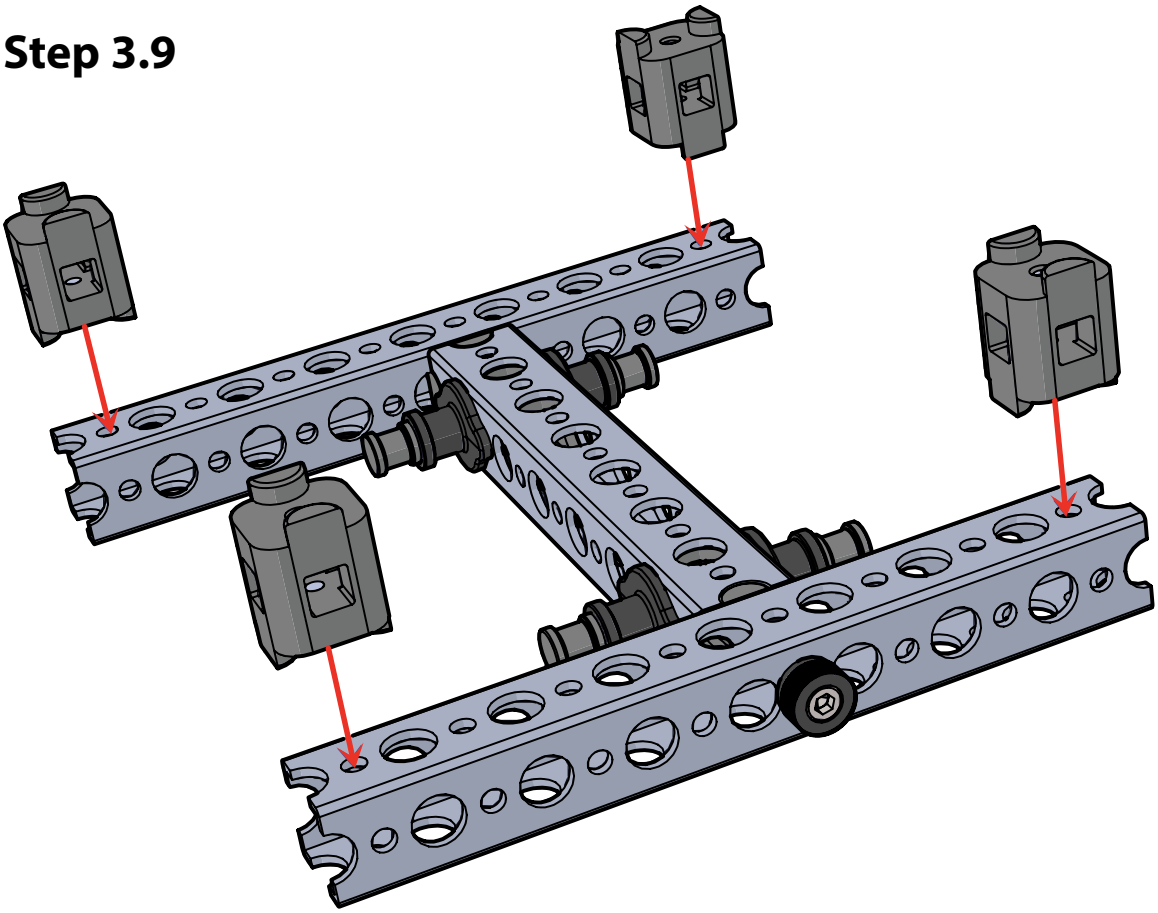
Step 3.7



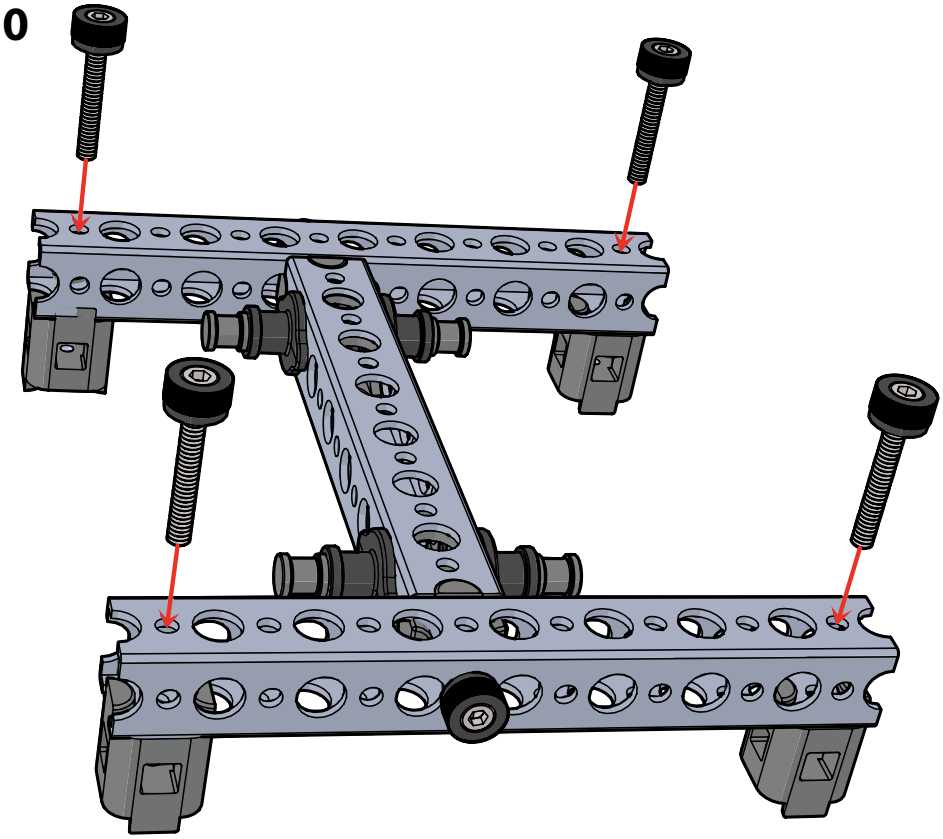
Step 3.8



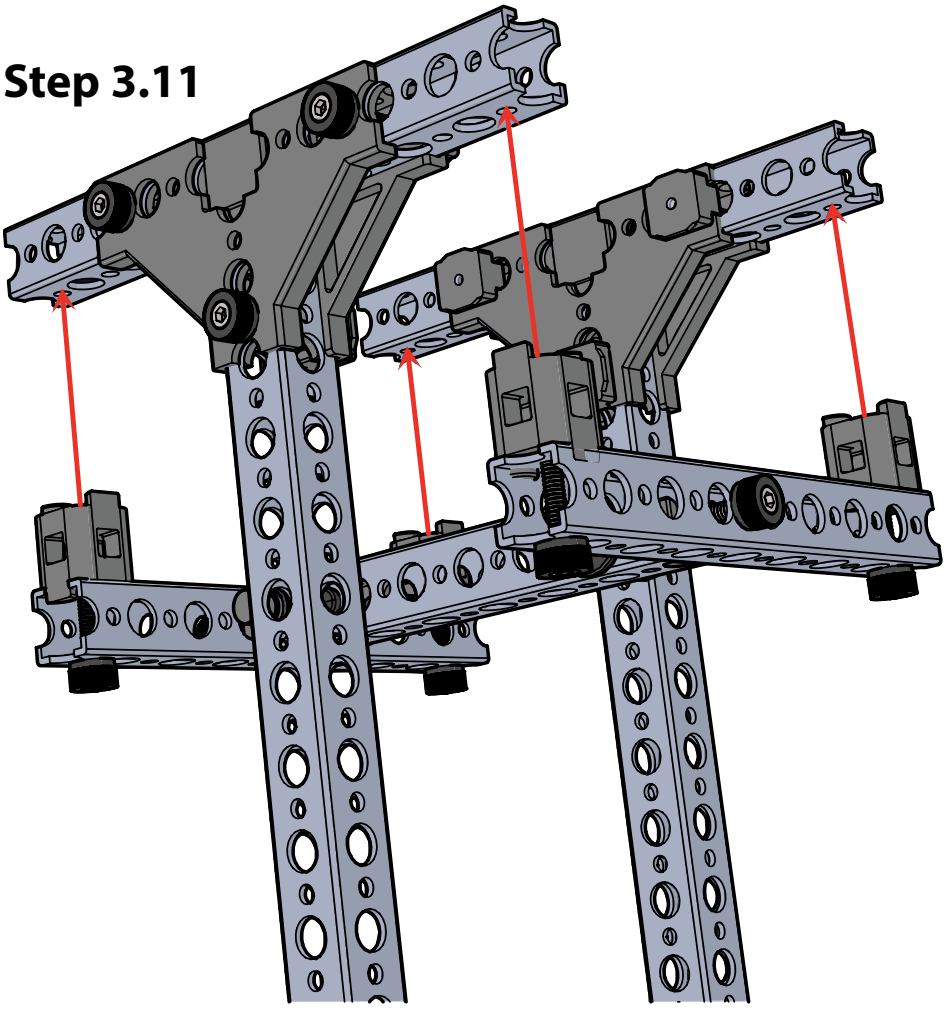
Step 3.9



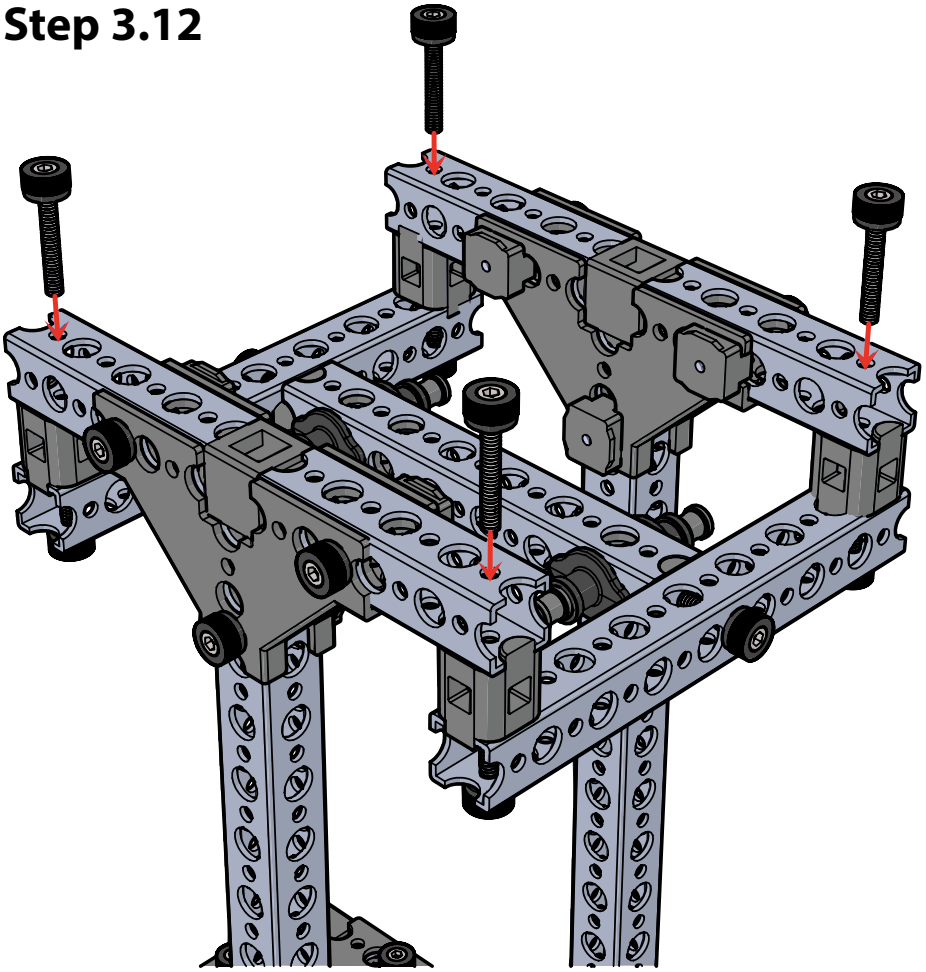
Step 3.10



Step 3.11

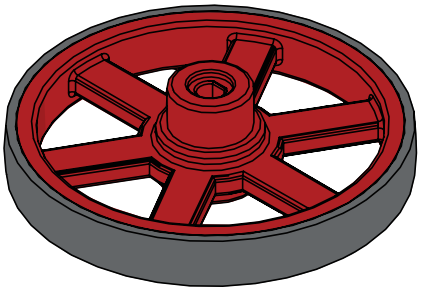


Step 3.12

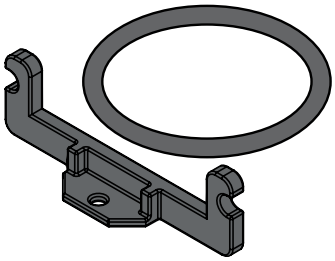


Step 4

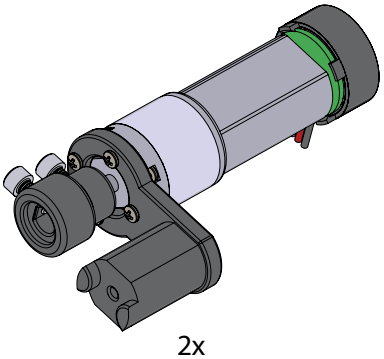
Parts Needed



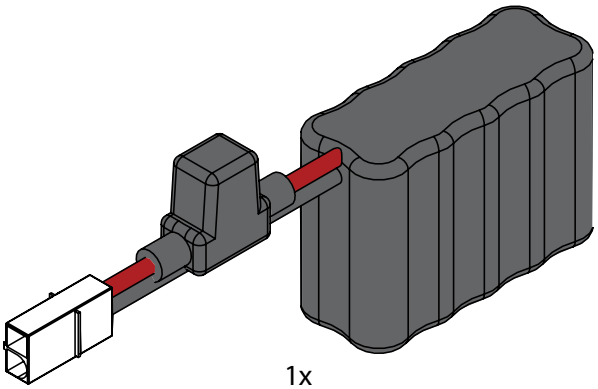
2x
90 mm Wheel & Tire 40222



2x
Battery Holder 40236



2x
Digilent DC Motor and Mount



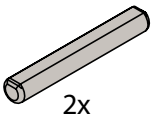
1x
10-Cell AA NiMH Battery Pack 41135

4x
Thumbscrew
40323

2x
Wing Nut 40221

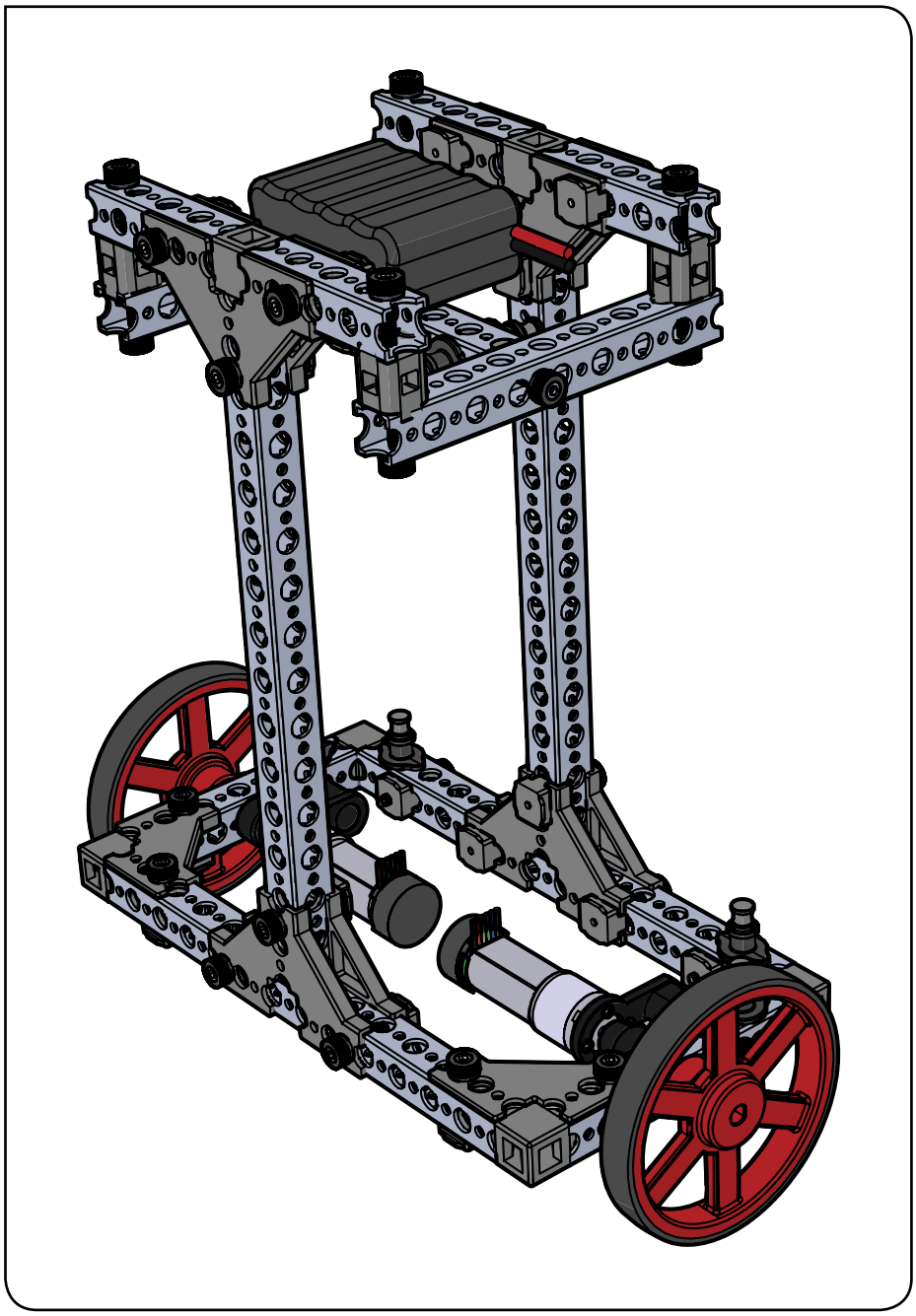
4x
Bronze Bushing
40227

6x
Setscrew 40516

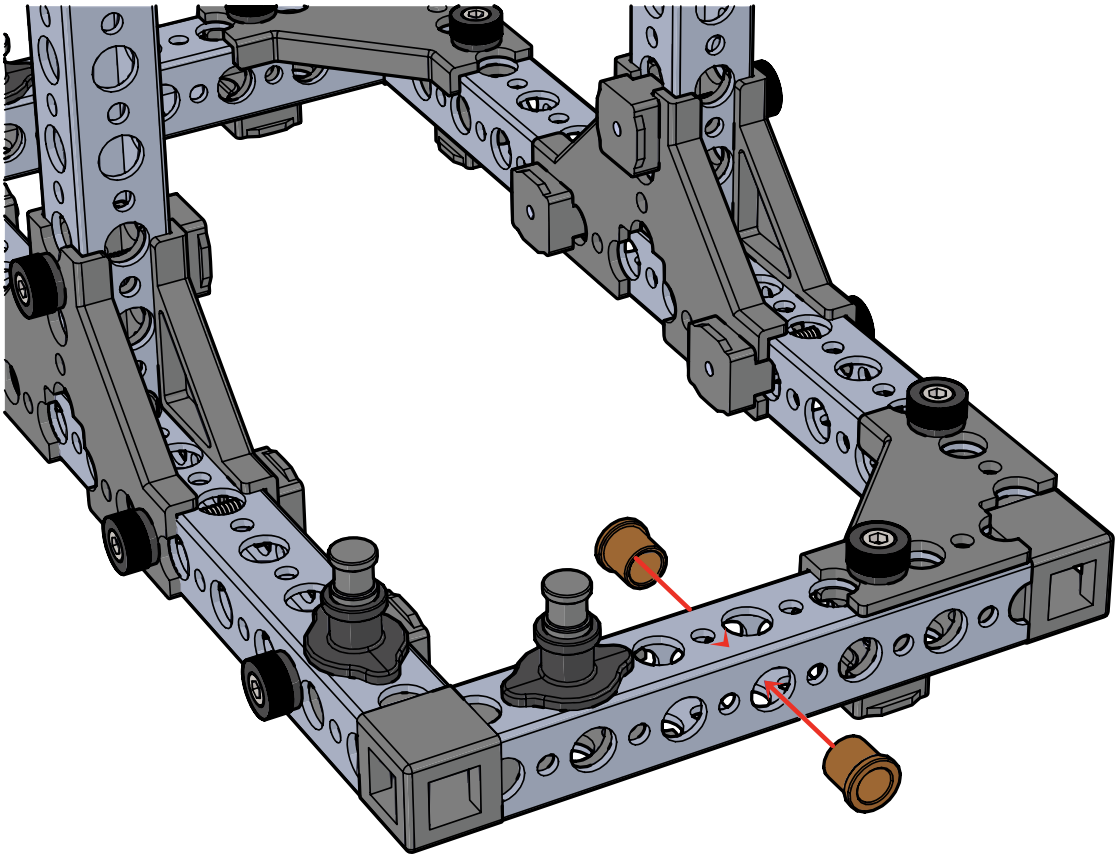


2x
40 mm Axle 40226

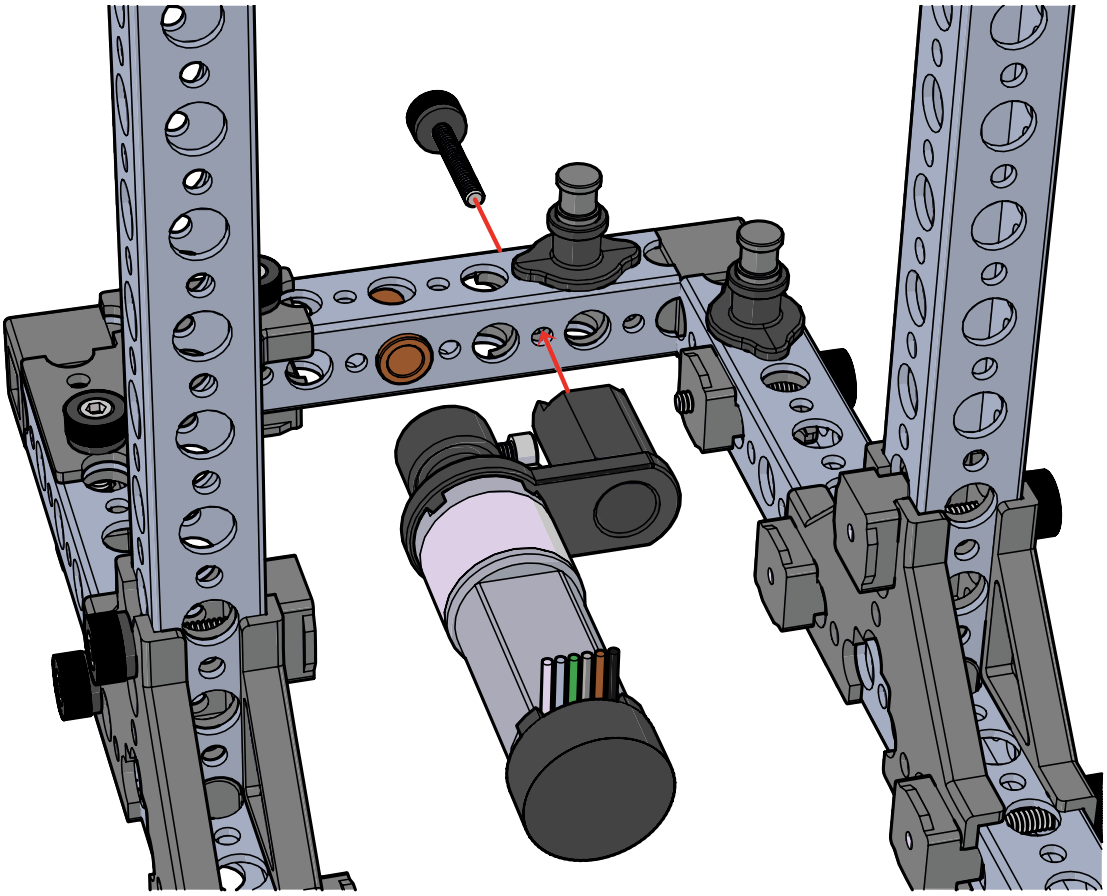
Partial assembly should look like this.



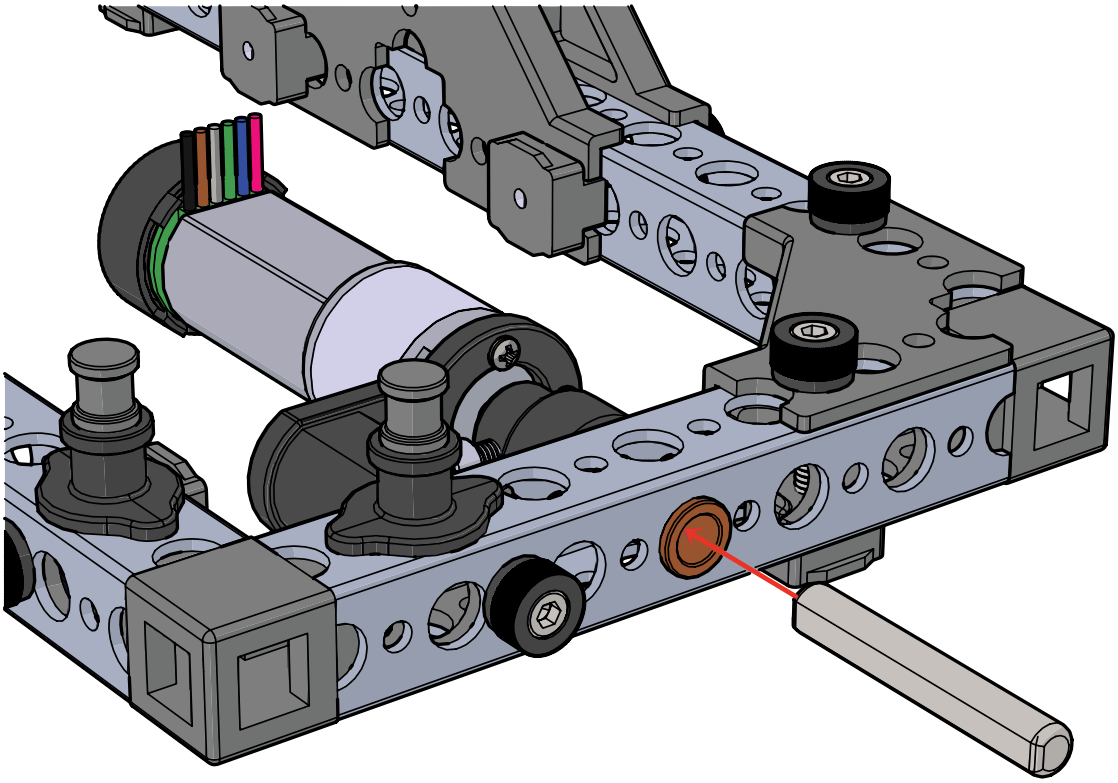
Step 4.0



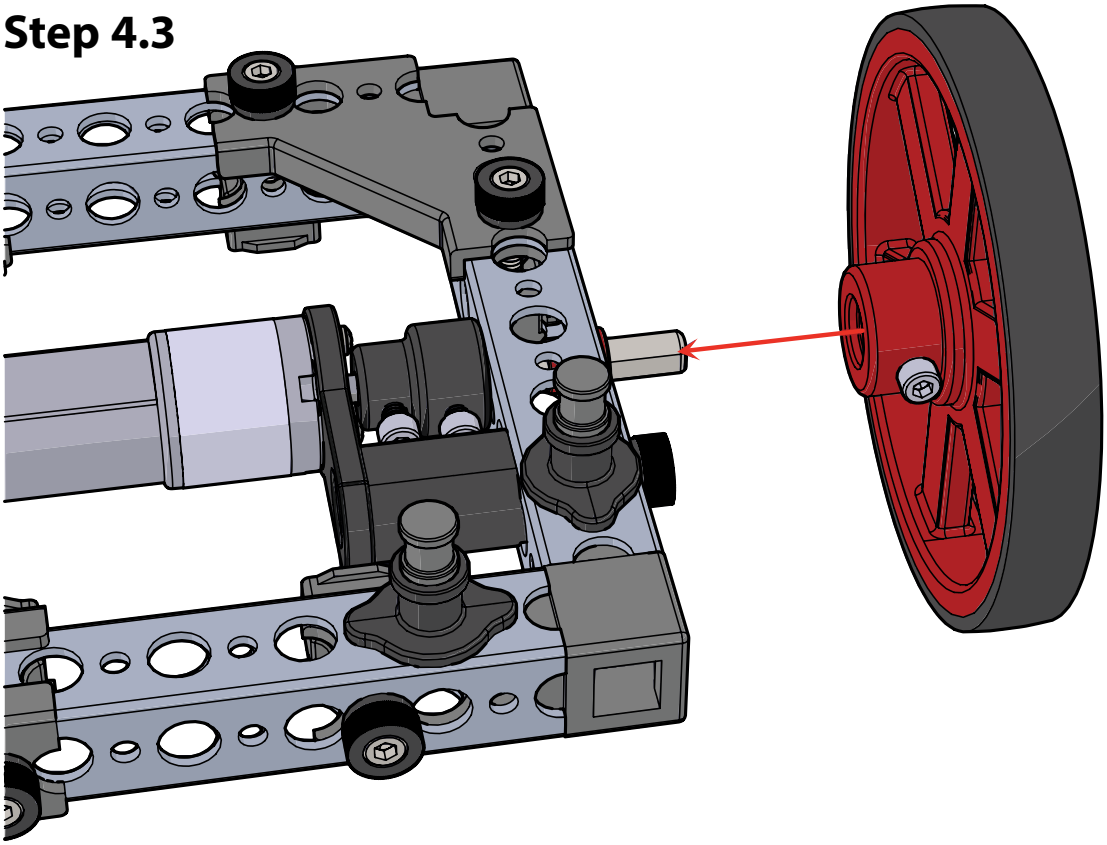
Step 4.1



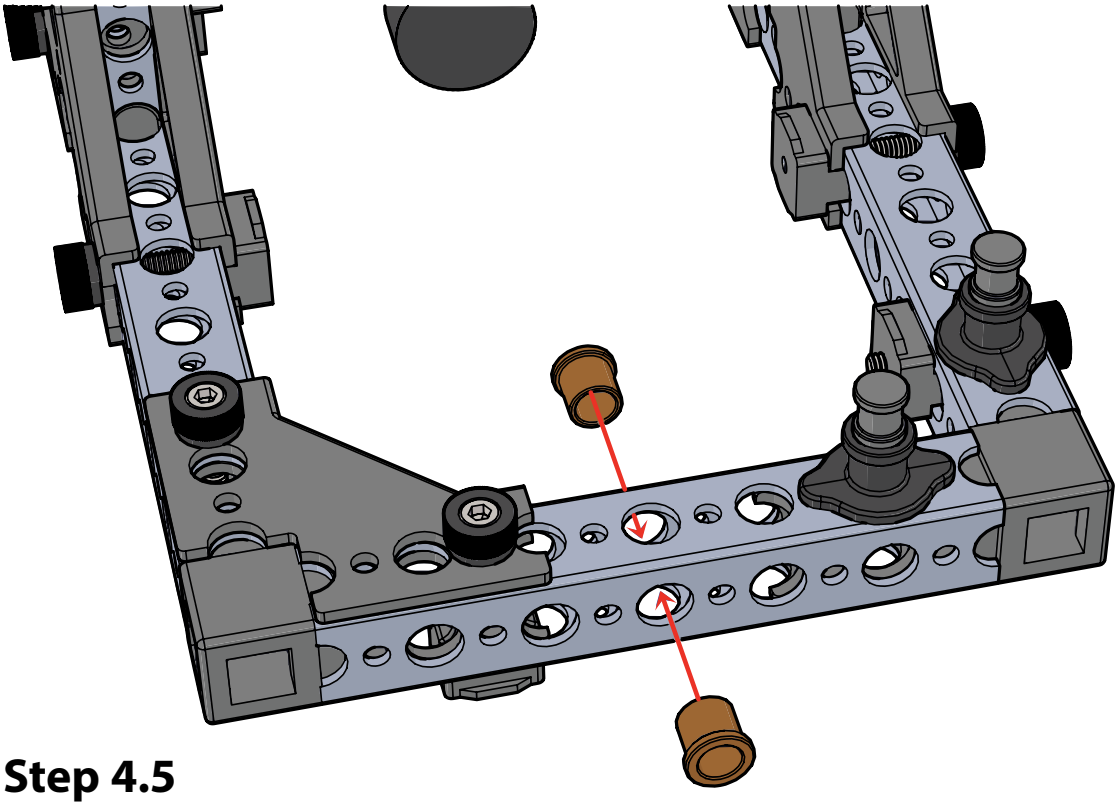
Step 4.2



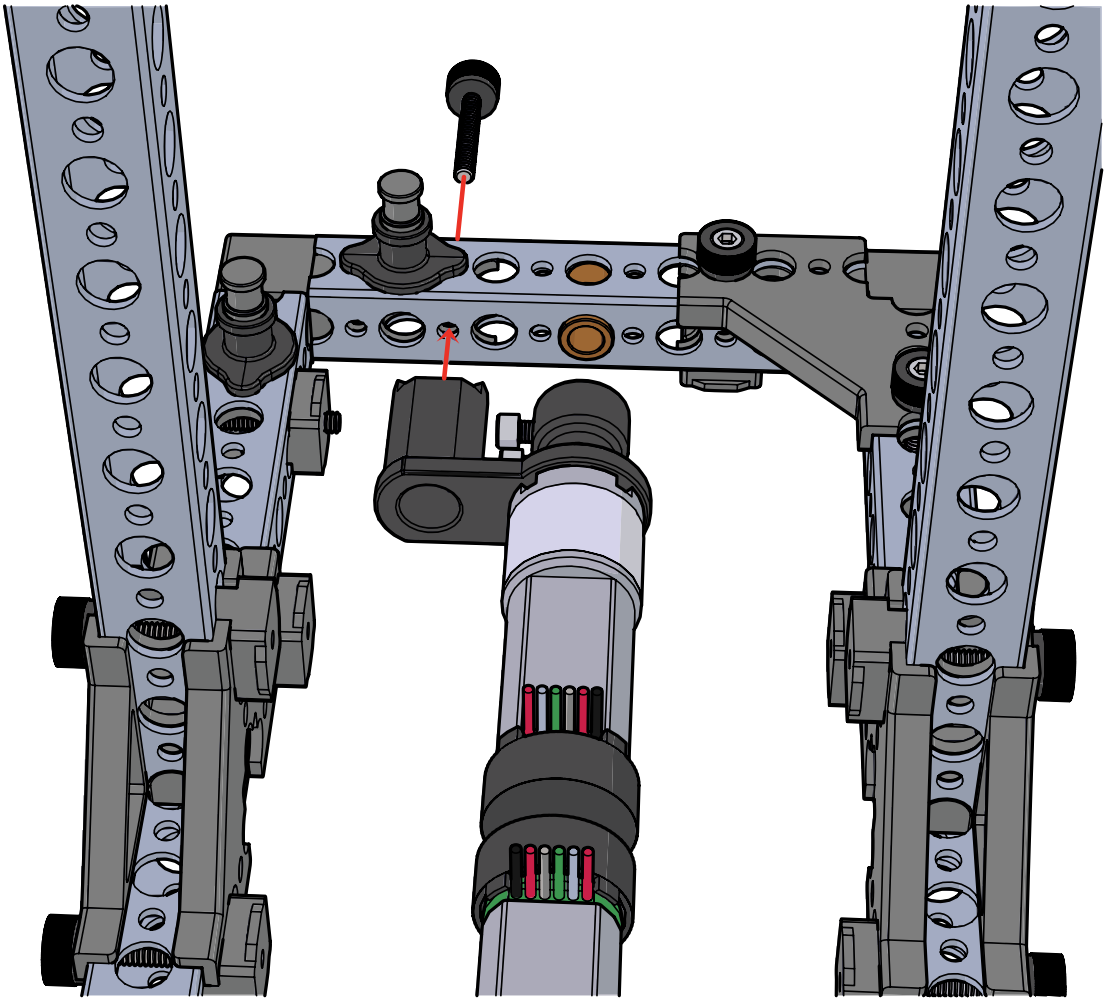
Step 4.3



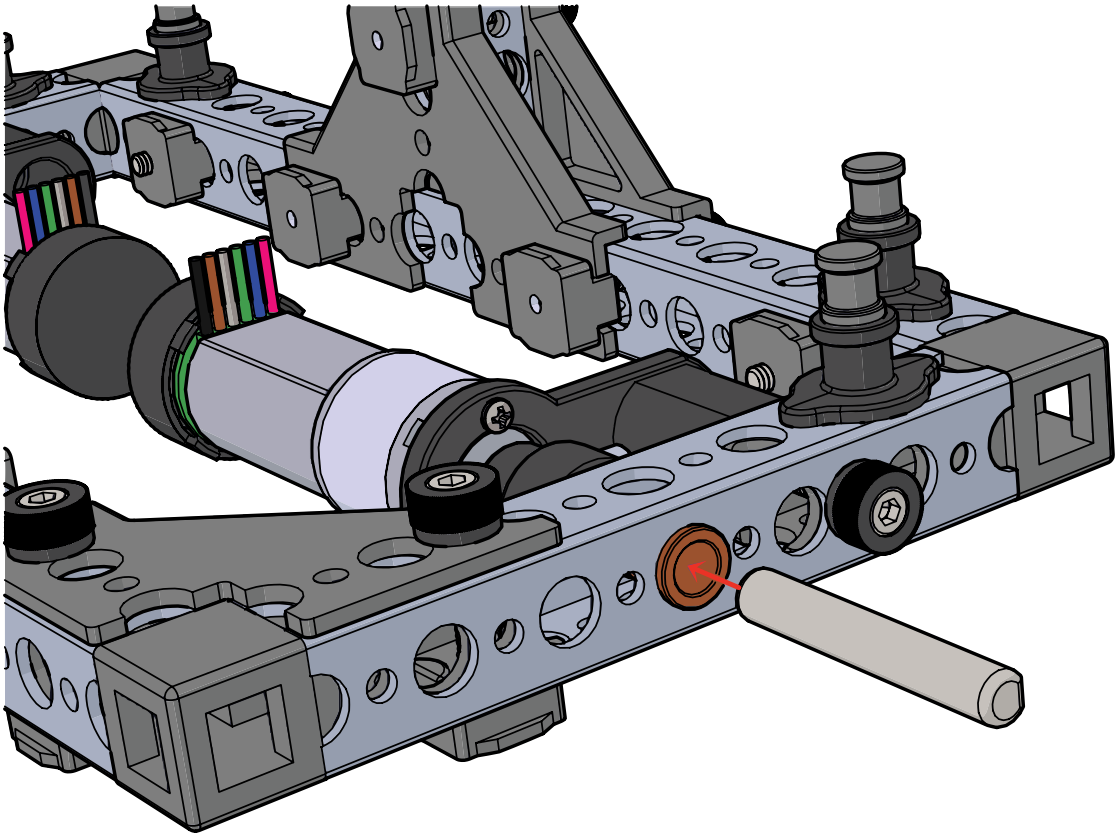
Step 4.4



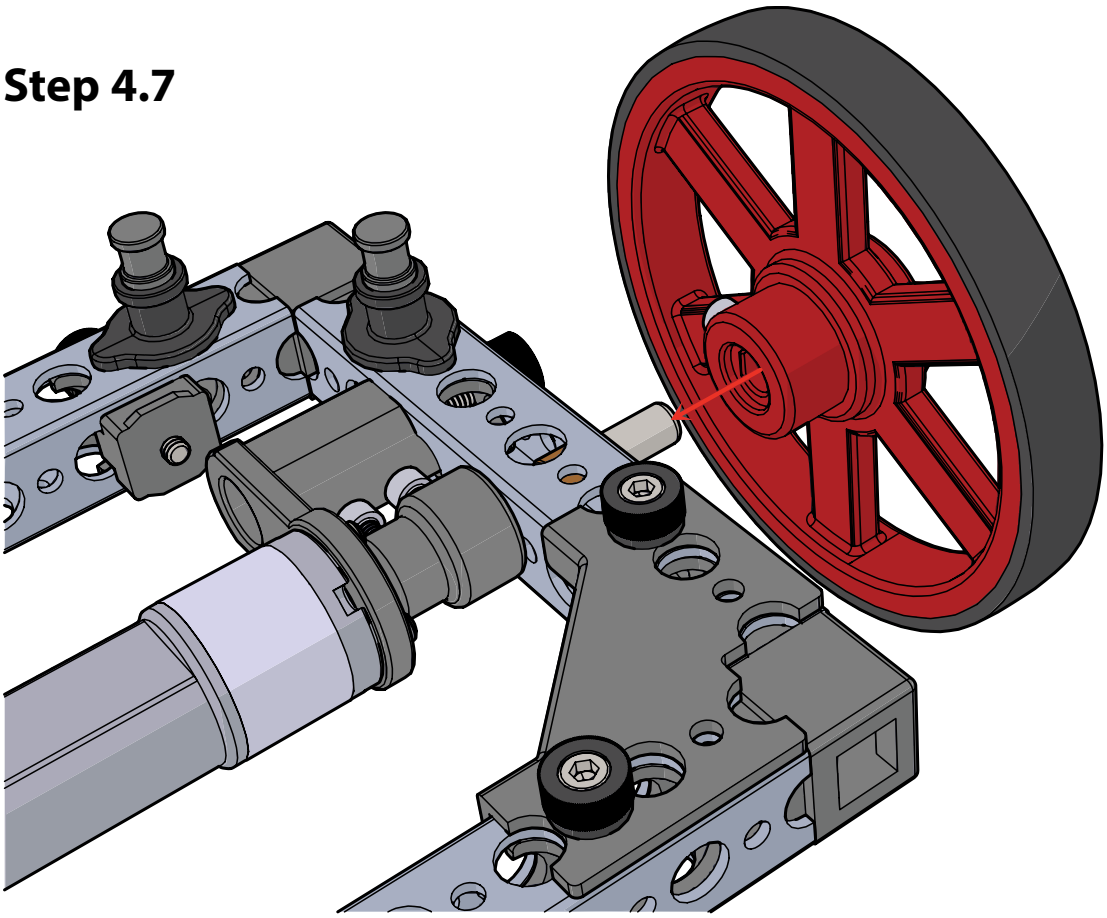
Step 4.5



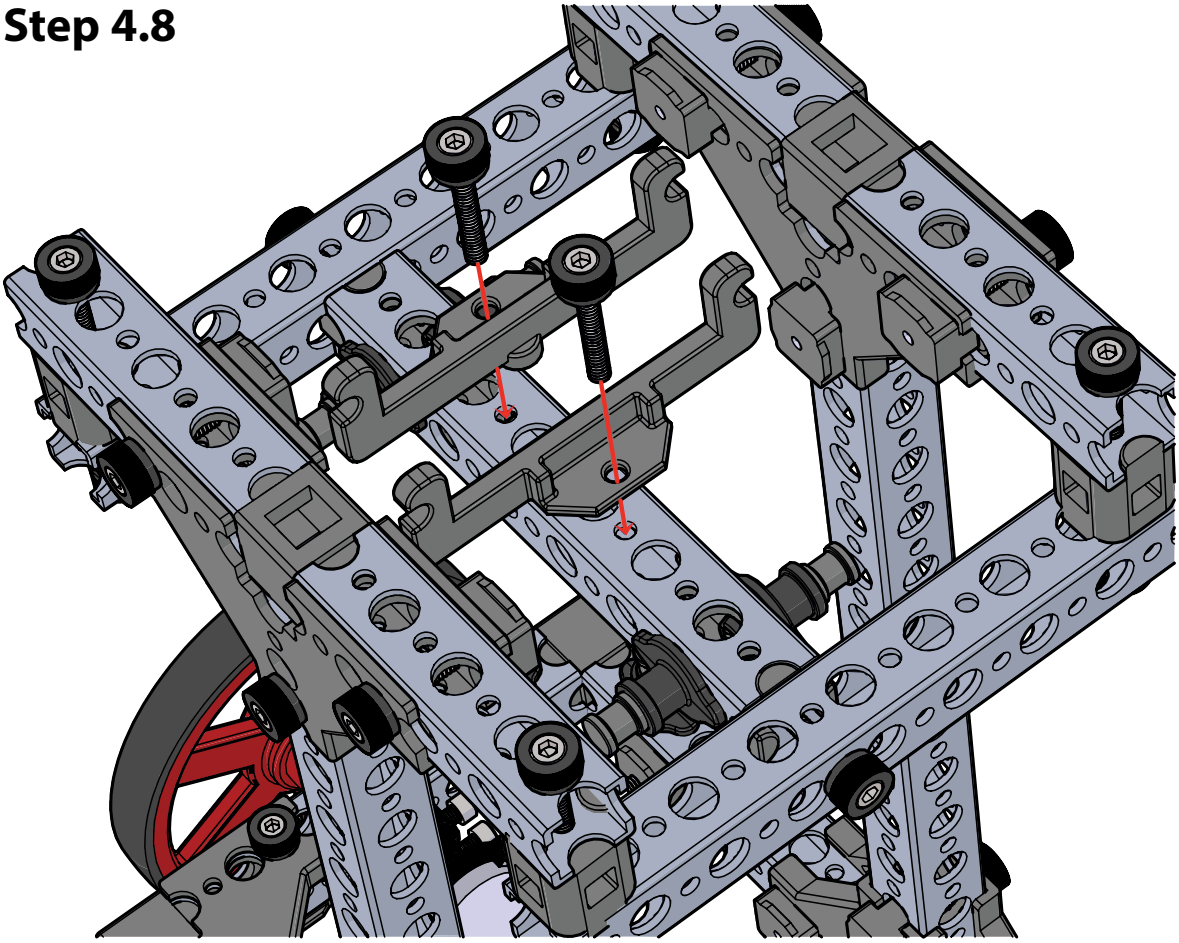
Step 4.6



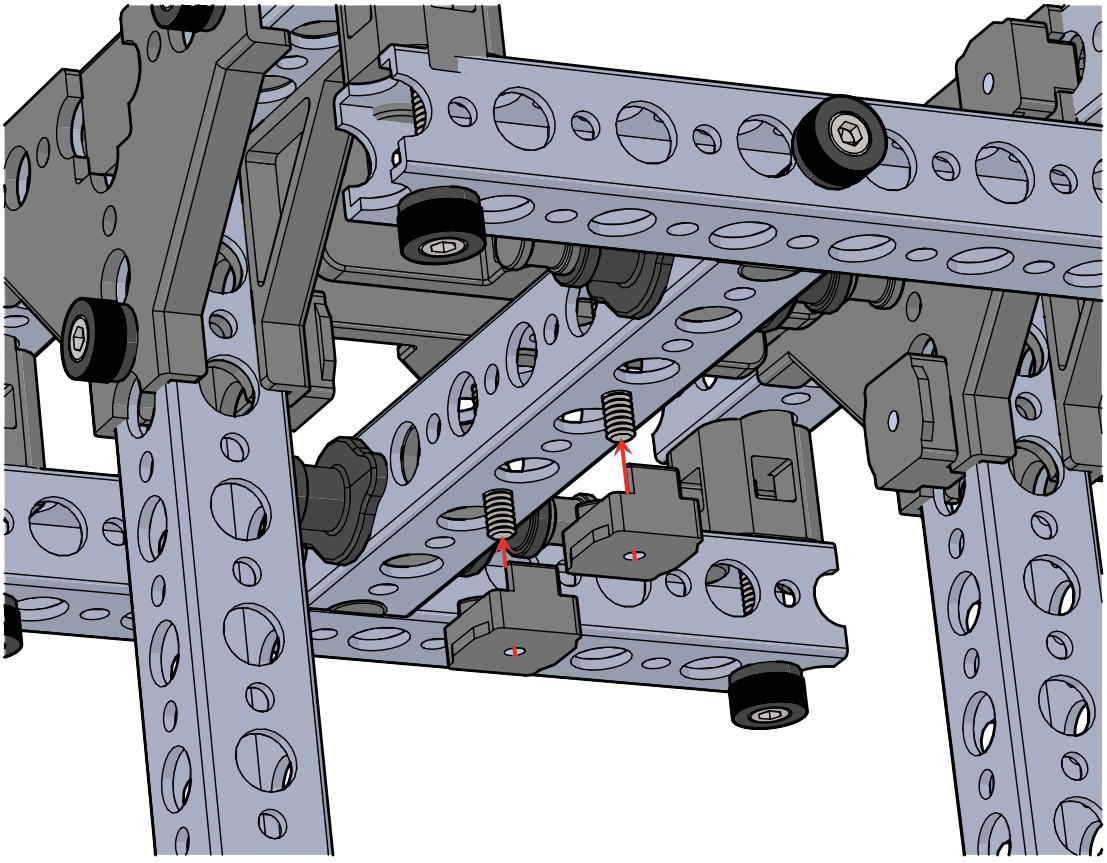
Step 4.7



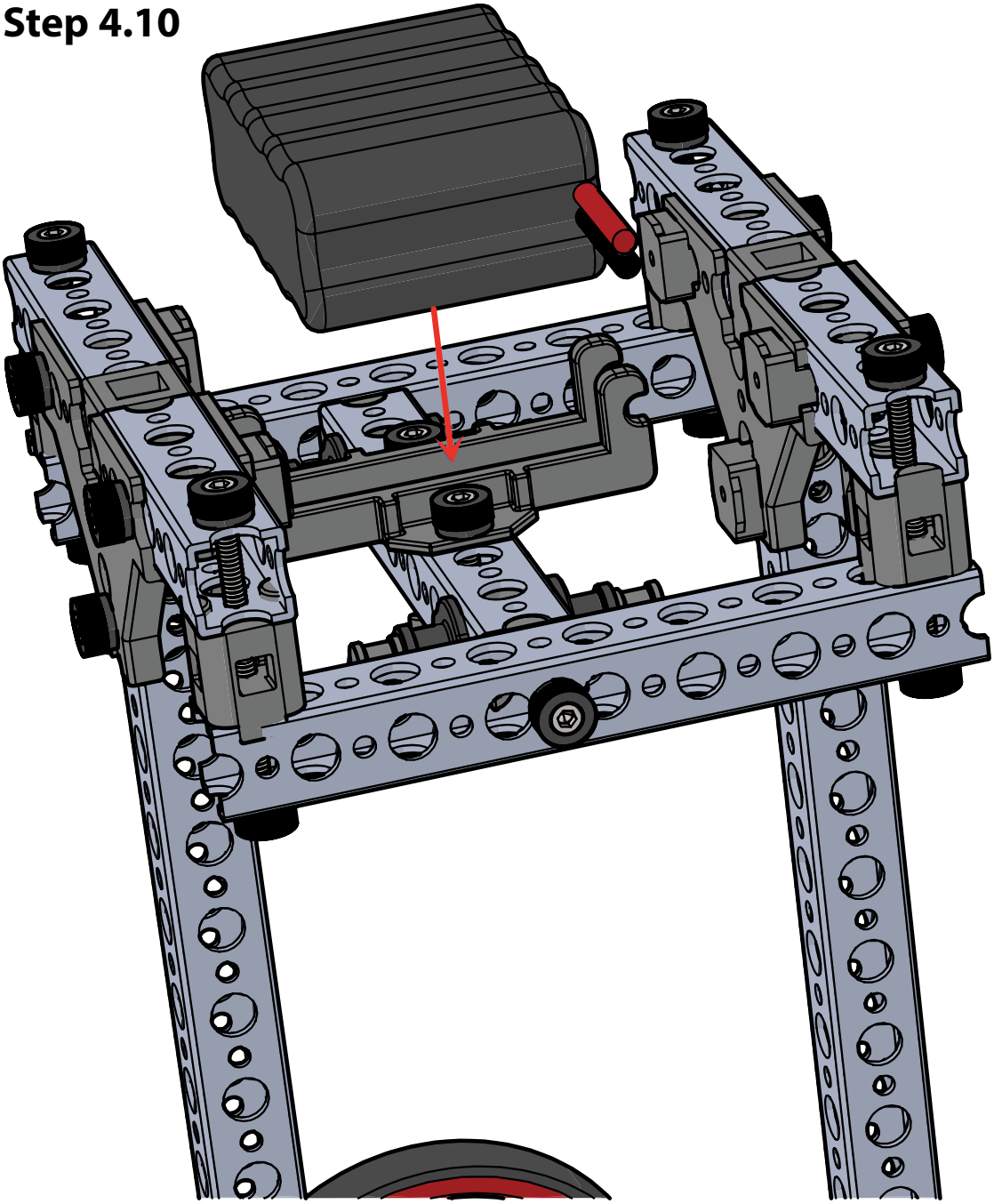
Step 4.8



Step 4.9

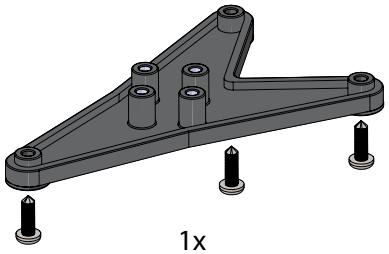


Step 4.10



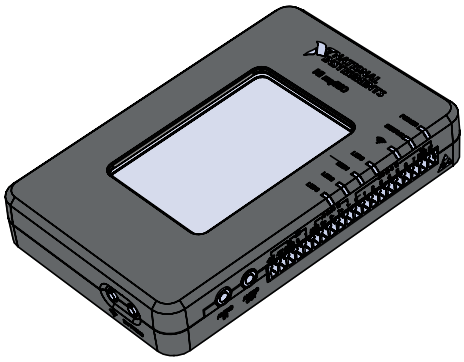
Step 5

Parts Needed



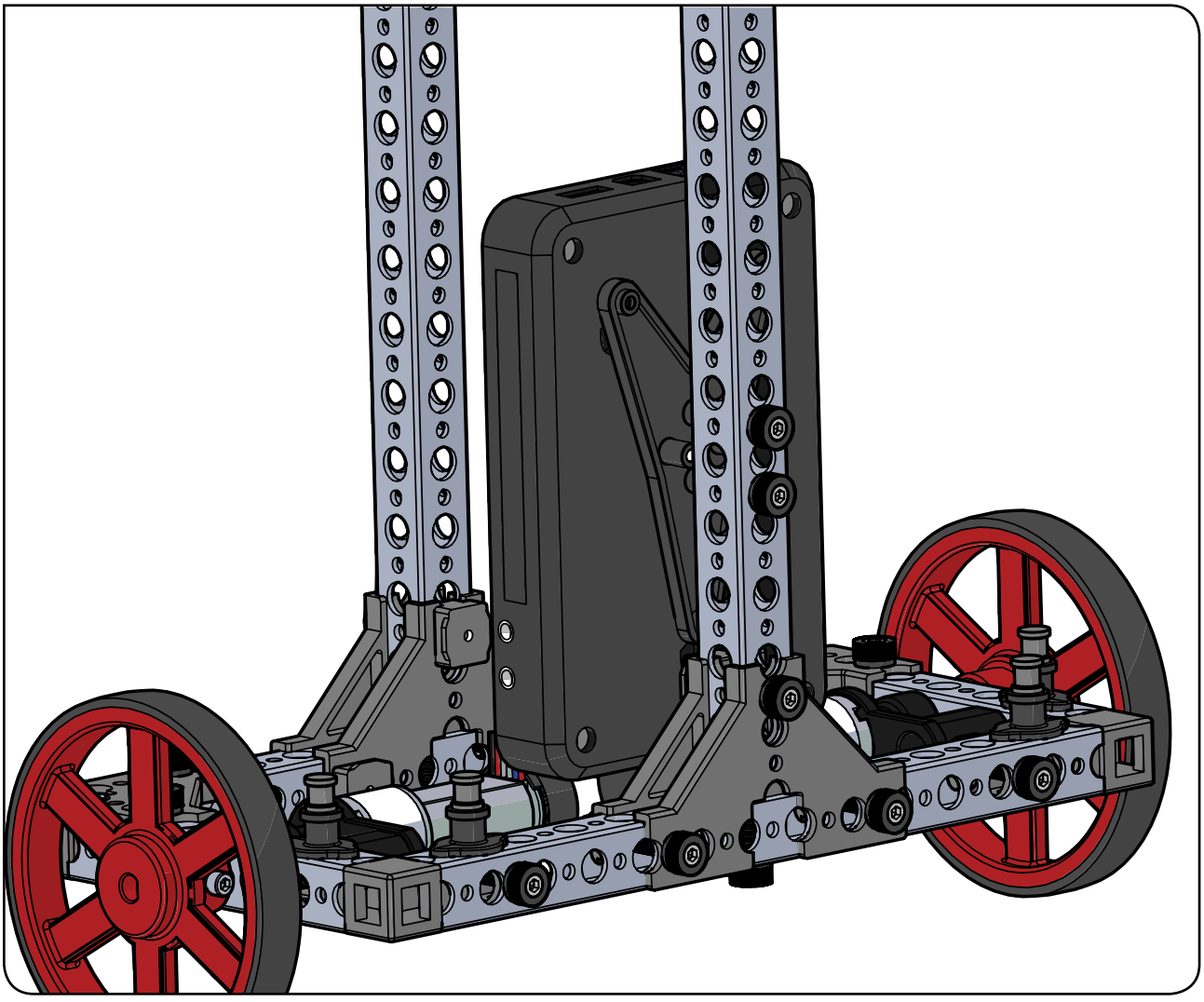
1x
myRIO Controller Mount 41137

6x
Thumbscrew 40323

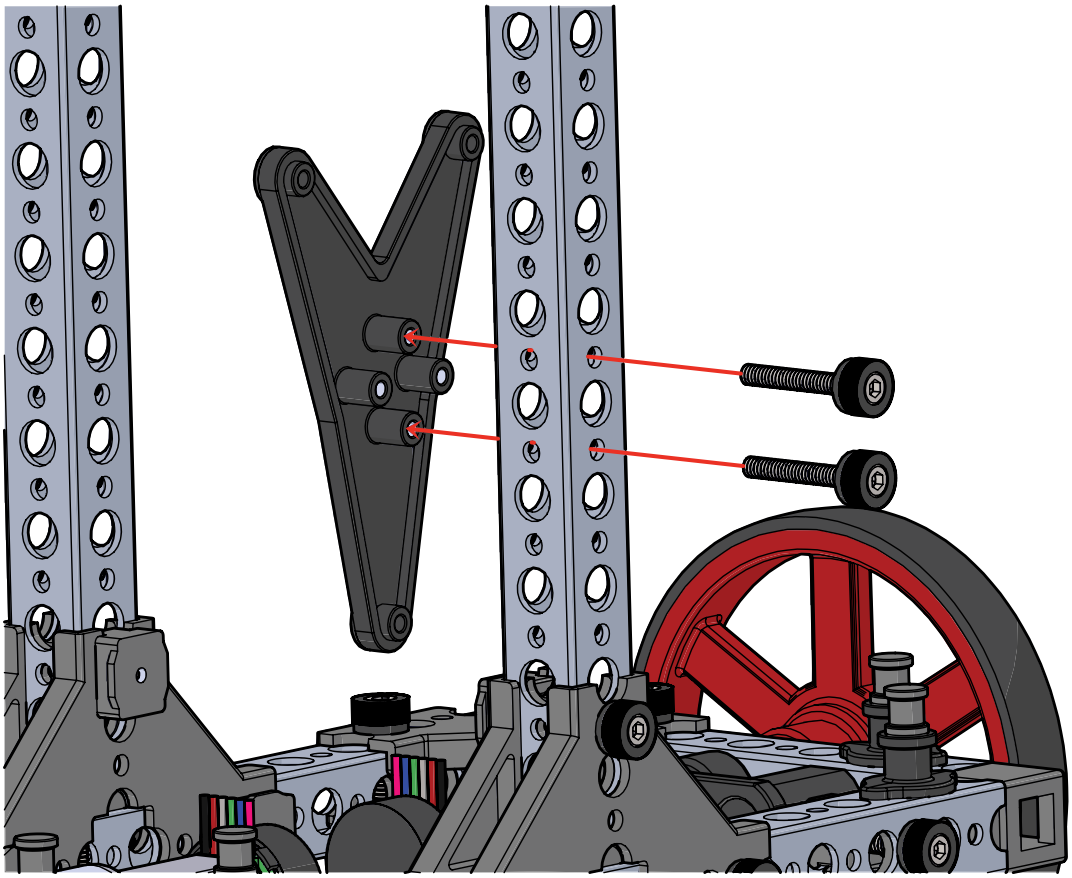


1x
myRIO Controller

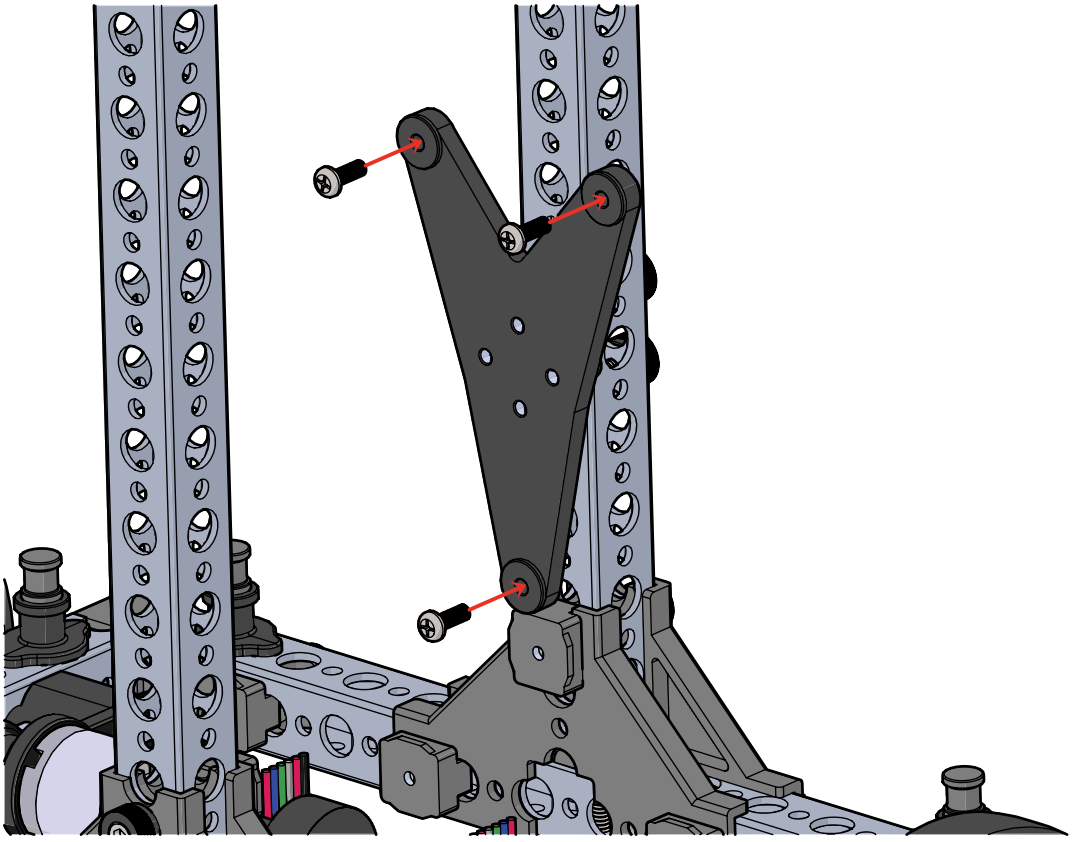
Finished assembly should look like this.



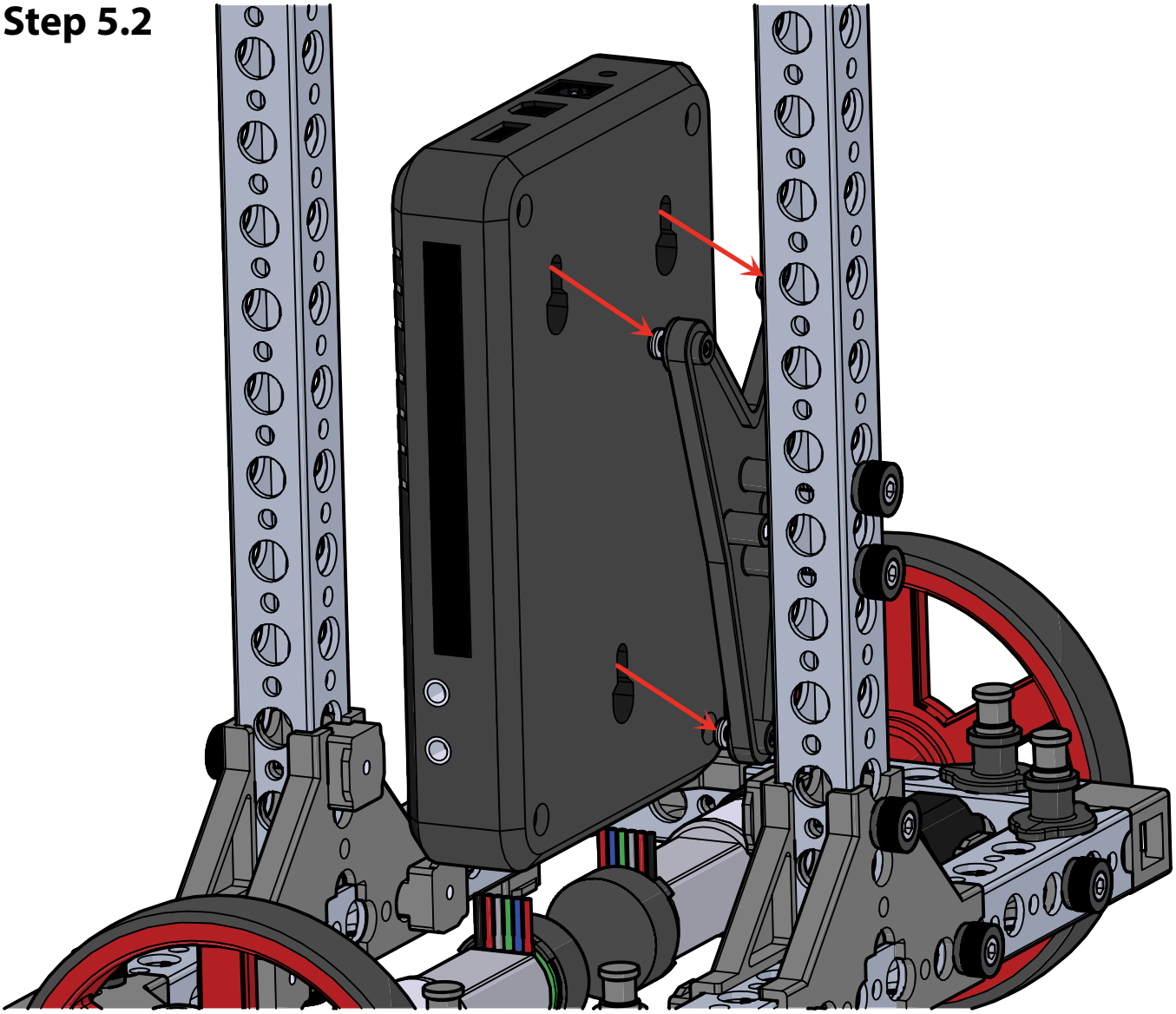
Step 5.0



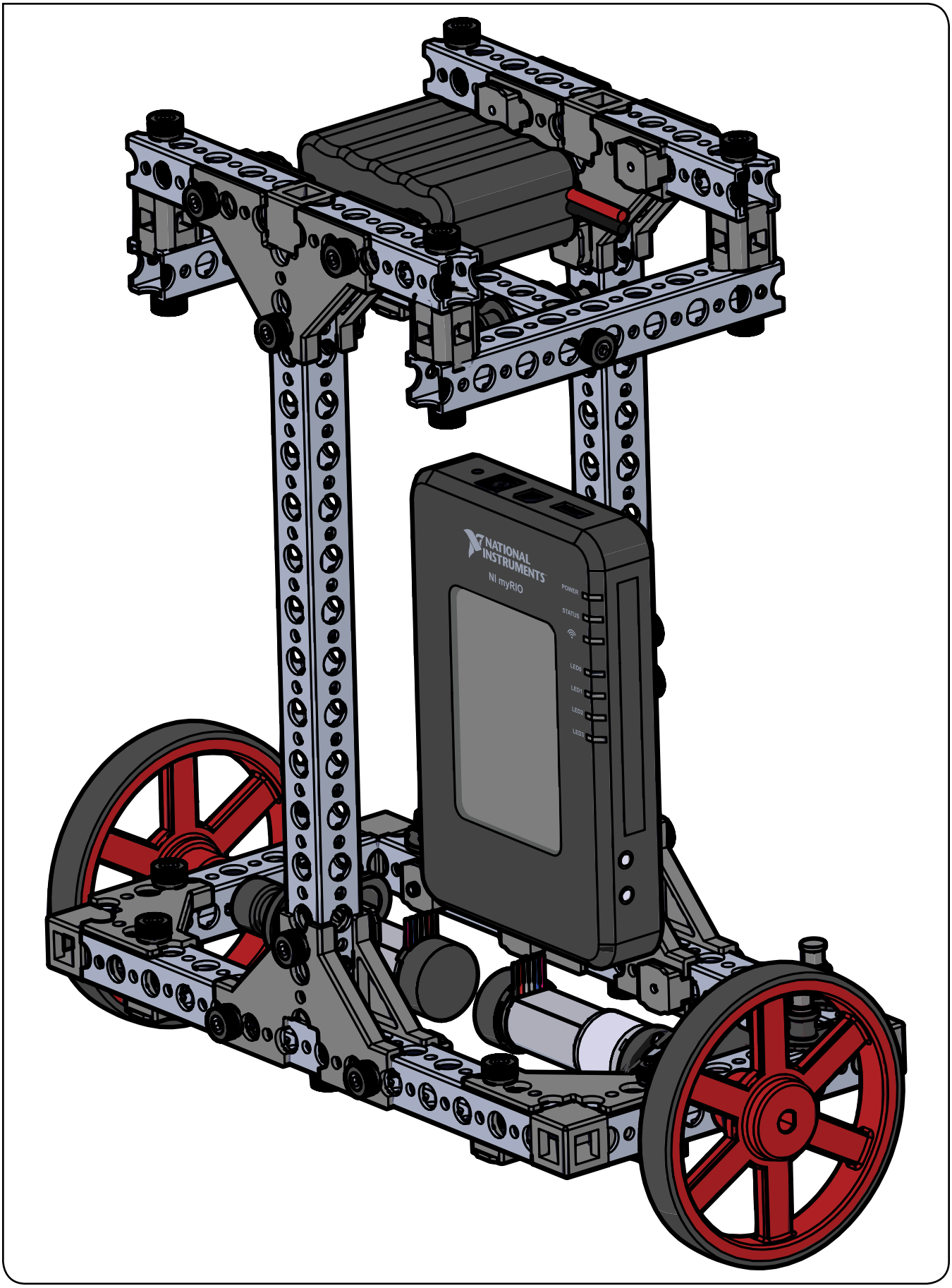
Step 5.1



Step 5.2

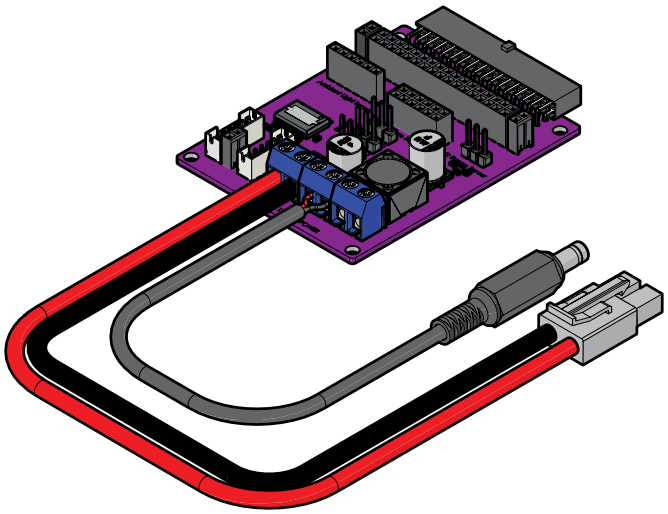


Finished assembly should look like this.

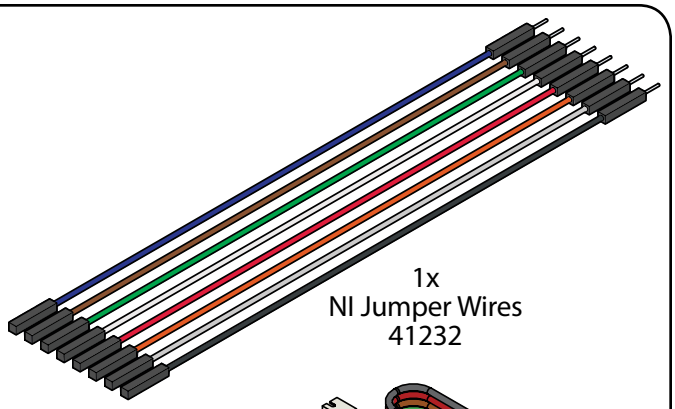


Step 6

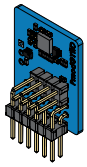
Parts Needed



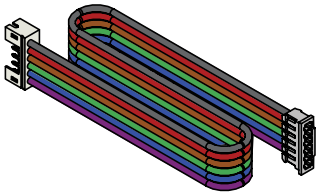
1x
myRIO Motor Controller Board 41136
with DC Power Plug and Battery Cable attached



1x
NI Jumper Wires
41232

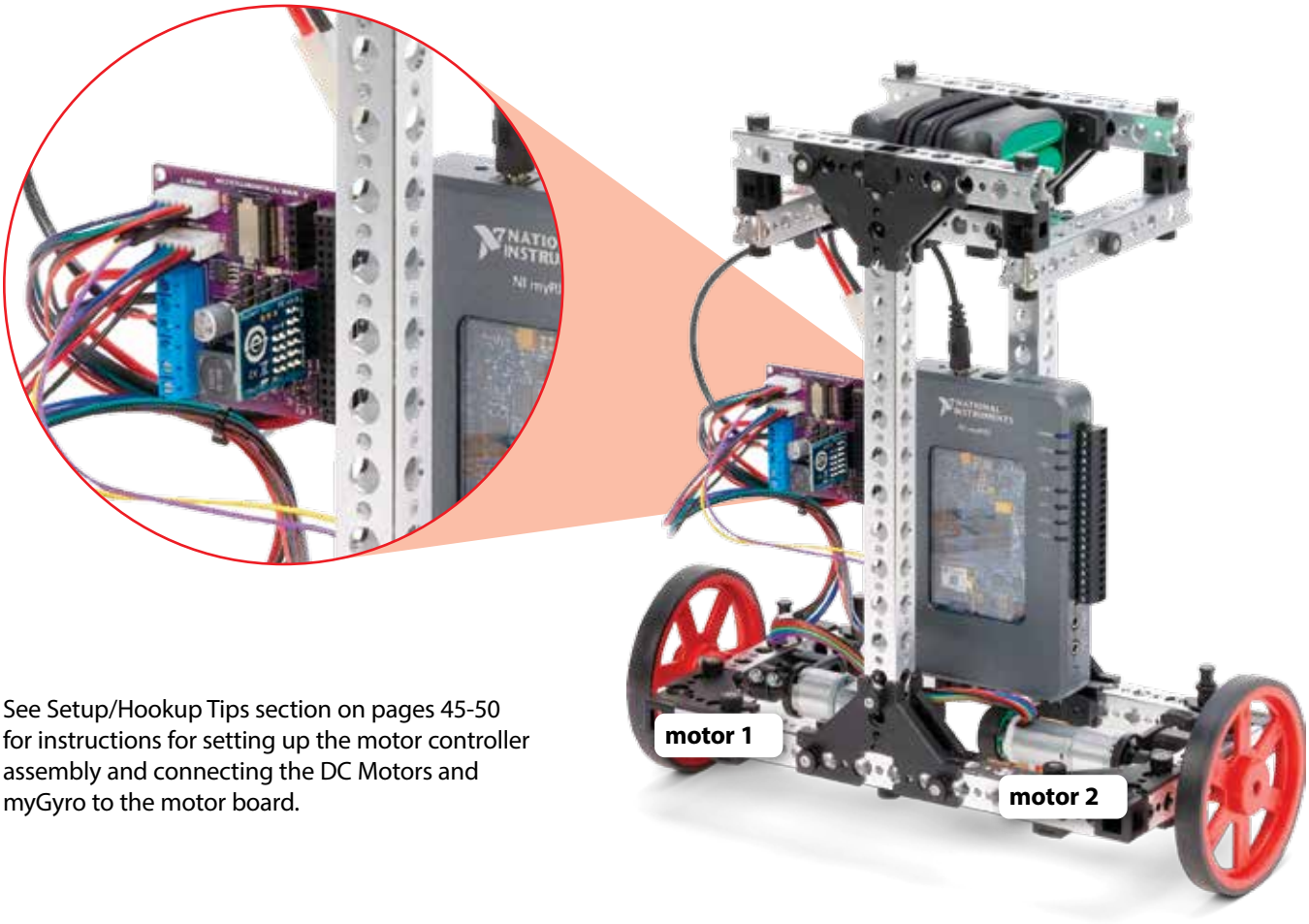


1x
Digilent Gyro
Sensor 41144



2x
DC Motor Extension
Cable 41147

Finished Assembly



See Setup/Hookup Tips section on pages 45-50
for instructions for setting up the motor controller
assembly and connecting the DC Motors and
myGyro to the motor board.

Step 6.1 Connect the motor board to myRIO MXP A

Step 6.1 Connect the Motor 1 and Motor 2 cables (as noted in the image at left) to the motor board ports labeled "Motor 1" and "Motor 2"

Step 6.2 Connect the Gyro sensor to the motor board port labeled "Gyro Sensor" and reference the image above for its orientation

Step 6.3 Find the encoder pins between the Motor 1 and Motor 2 ports on the motor board. Using the jumper wires, connect Pin A to MXP B pin 18 and Pin B to MXP B pin 22. Reference the myRIO MXP connector pinout in the "Electronics Overview and Specifications" section of this guide, or the myRIO user manual.

Step 6.4 Use the battery cable to connect from the T1 port of the motor board (red cable to +, black cable to -) to the battery cord

Step 6.5 Use the DC power cable to connect from the T2 port of the motor board (red cable to +, black cable to -) to the myRIO power port

Step 7

Set up myRIO and run LabVIEW code

7.1. Follow the hardware and software setup instructions in the "Set Up Your Control System" section. Make sure you:

- Connect the motor board to MXP B and connect the sensors and actuators to the motor board
- Connect the system to power
- Install software to your PC (via disk or web installer) and myRIO (via the Getting Started Wizard)
- Configure myRIO for WiFi

7.2. Connect to the myRIO WiFi signal (remember the SSID you chose) using the network icon on the bottom right corner of your screen, similar to how you connect to any other WiFi network

7.3. Navigate to Self-Balancing Robot code using the following path: C:\Users\...\Desktop\National Instruments\Pitsco Tetrix Prime for NI myRIO\Self Balancing Robot and open "Self Balancing Robot Project"

7.4. Make sure the IP Address in the parenthesis next to the "NI-myRIO" line item matches the myRIO wireless adapter IPv4 address in the Network Adapters window of the myRIO Configuration menu. If it doesn't:

- Right click the NI myRIO line item, and select "Properties"
- Update the IP address and click OK

7.5. Right click the NI myRIO line item and select "Connect."

7.6. Open "Self Balancing Robot Main.vi" by double clicking the line item in the project and press the run arrow in the upper left corner or press ctrl + r to run

7.7. Hold the robot upright so that it's center line is perpendicular to the ground

7.8. Press "Button 0" on the bottom of myRIO while still holding the robot upright and quickly let go

7.9. Notice the motor duty cycle changing on the front panel as the self-balancing robot sways

7.10. Gently tap the self-balancing robot and notice it return back to its original position

7.11. Try placing an object, such as a textbook, on top of the robot and see it continue to balance

7.12. Press Button0 on the bottom of myRIO to turn off the wheels and control system, and then repeat steps 7.7 and 7.8 to balance again

Step 8: Troubleshooting

If the Self-Balancing Robot is unstable:

8.1. Double check that the motors and encoder are wired properly – if they are swapped the robot will not balance

8.2. If you picked up the self-balancing robot or push it too hard, it will become uncalibrated. Press Button0 on the bottom of myRIO to turn off the wheels and control system, and then repeat steps 7.7 and 7.8 to balance again

Next Steps

- Learn about how the Self-Balancing Robot relates to control theory in the next section
- Get some ideas for expanding the Self-Balancing Robot on the first page of this section

Connections to Controls Concepts: Balance Arm

Prof. R. H. Bishop
University of South Florida

The self-balancing robot assembly provides insight into two key feedback control system concepts: (i) relative stability and robust stability of an unstable open-loop system and (ii) fundamental controller design tension in the presence of plant changes, external disturbances, and measurement noise.

Plant	Self-balancing robot
Sensor	Accelerometer, gyroscope, and encoder sensors
Actuator	DC servo motors
Performance	Robust stability, disturbance rejection, and measurement noise attenuation
Design objectives	Tune the control system by adjusting PD gain constants & stabilize the robot at the vertical
Reference inputs	Regulation to zero angle (vertical pose)

The self-balancing robot and two geared, DC motor-driven wheels are the plant and actuators, respectively. The sensor suite includes a gyroscope that measures angular velocity and an accelerometer that measures non-gravitational acceleration. Since the accelerometer and gyro are known to have systematic errors (bias, drifts, etc.) and are noisy, the signals from the sensors are combined to produce an accurate measurement of the orientation angle of the robot—more accurate than can be achieved with either sensor individually. The external reference commands enter the control system via the host computer either through keyboard entry or front panel commands sent over the network connection to the myRIO which hosts the PD controller code. The design goal is the balance the robot in an upright pose in the presence of external disturbances and to be robustly stable to plant variations (e.g., remain stable even when a large mass is placed on the robot). The controller is a PD controller.

Relative stability and robust stability

A closed-loop feedback system is either stable or not stable. We say that a feedback control system that displays a bounded response to a bounded input is stable. Note that this means that the closed-loop feedback system must have a bounded output to every bounded input. This is known as absolute stability. We might wonder about marginally stable closed-loop systems wherein the response remains bounded, but does not decay with time. Typically, our design specifications require the closed-loop feedback system tracking error response, represented in the frequency domain as E(s)=Y(s)-R(s), to decay to zero—not just remain bounded—and this is the focus of this discussion. Obviously, it is not practical to test the response to every bounded input; however, a necessary and sufficient condition for the closedloop feedback system to be stable is that all the poles of the closed-loop transfer function lie in the left-half s-plane—this is a primary requirement of the controller design process. Once we have expertly designed the controller so that the closed-loop feedback system is stable, we can further quantify the degree of stability, often referred to as relative stability. So, a closedloop feedback system can be more (or less stable) with a given controller design than the same system with a different controller design. This all relates to the concept of a robust feedback control system. That is, a robust feedback control system is one that maintains acceptable performance in the presence of plant uncertainty, external disturbances, and measurement noise. From the point of view of relative stability, a robust controller typically displays a greater relative stability than a controller with less ability to retain closed-loop stability in the presence of plant uncertainty, external disturbances, and measurement noise. Consider the PD controller for the self-balancing robot that allows the robot to remain stable even when a large load (such as a book) is placed on the top thereby significantly changing the plant or when an external gentle push is applied. A well-designed PID controller can be very robust and explains why they are utilized in industry to such a large extent. Note that the PD controller is a PID controller with the integral gain set to zero.

Fundamental design tension

Consider a system with a PD controller, actuator, sensor, and plant with three inputs, the reference input, the external disturbance, and the measurement noise. Our design objective is to determine the PD gains such that the system is robustly stable to plant uncertainties, rejects external disturbances, attenuates the measurement noise, and meets given performance specifications (such as settling time and percent overshoot). In the case of the self-balancing robot, we have significant measurement noise stemming from the noisy gyro and accelerometer. If we analyze the tracking error, E(s), we find that a fundamental design tension, that is, to reduce the sensitivity to plant changes (robustness) and to reject the external disturbances, we want to make the controller gains 'large'. On the other hand, to attenuate the measurement noise, we want to select the controller gains to be 'small'. This fundamental tension does not stem from using a PD controller, but is a fundamental design challenge no matter what controller structure is employed. Fortunately, there is often a separation of frequencies over which the external disturbances and measurement noise exist. Measurement noise is typically high frequency and external disturbances are typically low frequency. If this is not the case in a particular situation, the design problem becomes significantly more difficult. The classic control design approach is for the designer to design a controller with high gain at low frequencies and low gain at high frequencies! Consider the PD controller for the selfbalancing robot—increase the gains and observe the improved performance to an external gentle push. As you continue to increase the gains you should see the feedback control system response begin to deteriorate as it begins to amplify the measurement noise. The trade-off results in a set of controller gains that provide stable acceptable performance in the presence of plant changes (adding a mass at the top of the robot) and external disturbances (gentle pushes), while attenuating the measurement noise (from noisy gyro and accelerometer).

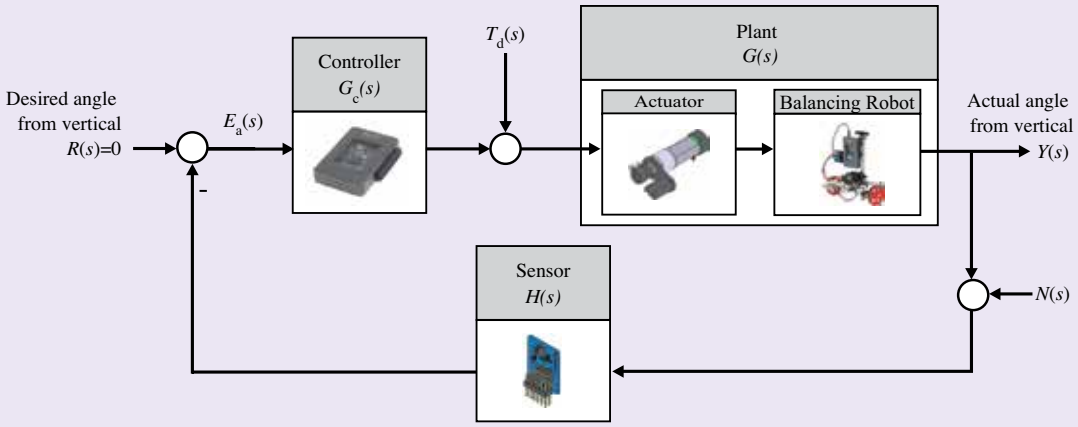


Figure 4 Closed-loop feedback control of an open-loop unstable plant.

Note: Control concepts described in detail in Modern Control Systems, by R. C. Dorf & R. H. Bishop, 13th Ed., Pearson Education, Inc., 2017.

Resources

For more resources for the PITSCO TETRIX PRIME for NI myRIO: ni.com/tetrix-prime

To learn more about myRIO: ni.com/myrio

To learn more about LabVIEW: ni.com/labview

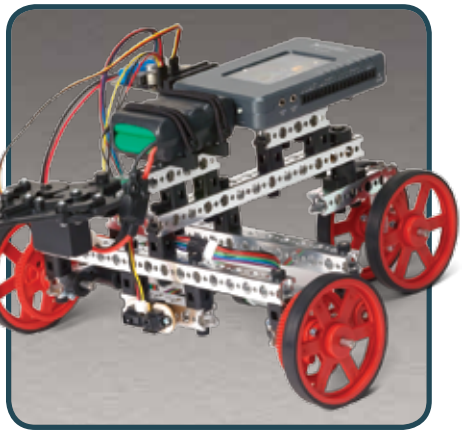
To learn more about the Pitsco TETRIX Building System: TETRIXrobotics.com

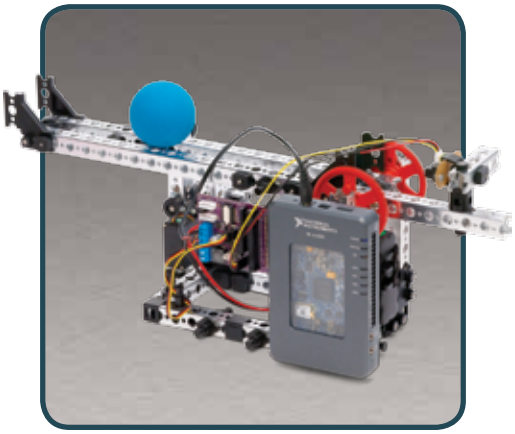
PITSCO

TETRIX® PRIME

Robot Builder's Guide

for NI myRIO





Call Toll-Free
800•835•0686

FreeFax
800•533•8104

Visit Us Online at
pitsco.com

CE

PITSCO EDUCATION