Stanford Pupper

Release 2020

Sep 18, 2021

CHAPTER 1

About Pupper

Stanford Pupper is a small quadruped robot that can hop, trot, and run around. We hope that its low cost and simple design will allow robot enthusiasts in K-12 and beyond to get their hands on fun, dynamic robots.

The robot's brain is a Raspberry Pi 4 computer, which receives commands from a wireless PS4 controller and controls the servo motors, three per leg, to move the feet and body to the right places.

The robot is designed to be "hacked" – we want you to be able to adjust and expand the robot's behaviors to your heart's content. While the robot can walk out-of-the-box, some of the features you could add include different gaits (bounding, galloping, etc), or high level behaviors like playing fetch or following you around. You can also simulate the robot's motion in PyBullet before touching the real robot.

To get started, check out the pages linked below on part sourcing and assembly. If you purchase the parts yourself, it'll run you about $900-$1000. However, you can purchase a kit to build the robot from either MangDang or Cypress Software for cheaper than what it would cost you to get the parts yourself. The two vendors sell different options so check both of them out to see what works for you. While we're not affiliated with either company, we've verified both of their kits.

1.1 Part Sourcing

1.1.1 Pre-Made Kits

Two small businesses are now selling Pupper kits. Both vendors sell a variety of types of kits, from partially complete to fully assembled. Buying a kit instead of purchasing the parts yourself will most likely save you money. Check out both websites to see which kit best suits your needs.

- Cypress Software
- MangDang

1.1.2 Bill of Materials

Link: Bill of Materials

If you'd like to source the parts yourself follow the instructions in the BOM. Most of the parts can be bought directly from a reseller like Amazon or McMaster-Carr, but for some you'll need to get them custom manufactured for you. The custom parts include the carbon fiber routed parts, the 3D printed parts, the power distribution printed circuit board, and the motors. The BOM spreadsheet goes into much more detail.

1.2 Assembly

Contents

1.2.1 Hip Assembly

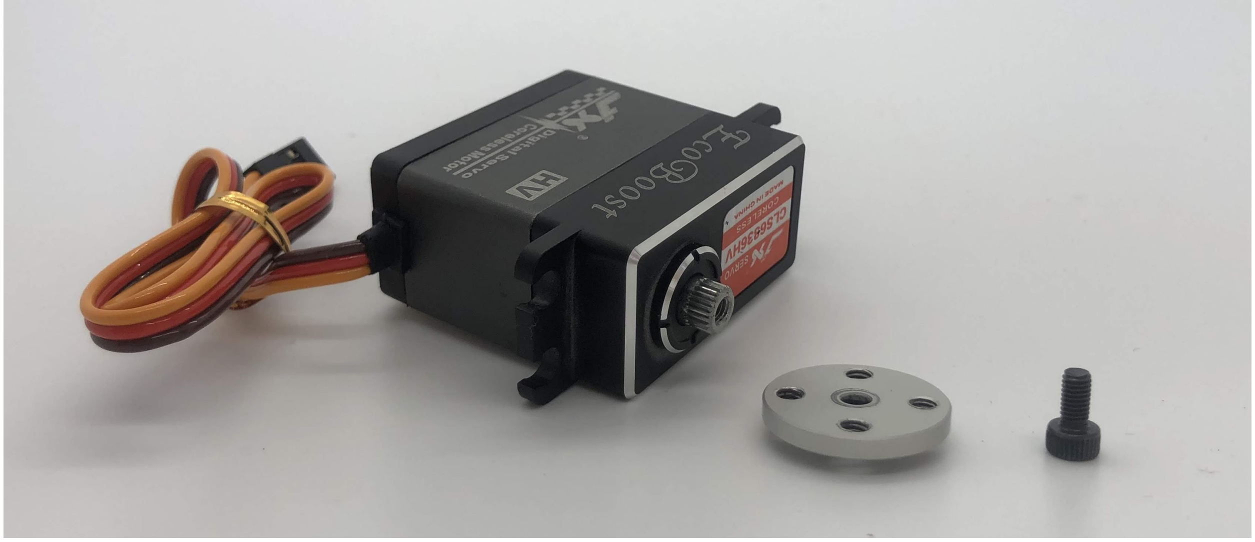
Step 1. Install disc

- Video Instructions: https://youtu.be/aUnK3Wvj8CU
- Materials: M3x6mm socket head screw, servo disc, loctite, servo arm (to adjust servo motor)
- Tools: 2.5mm hex driver

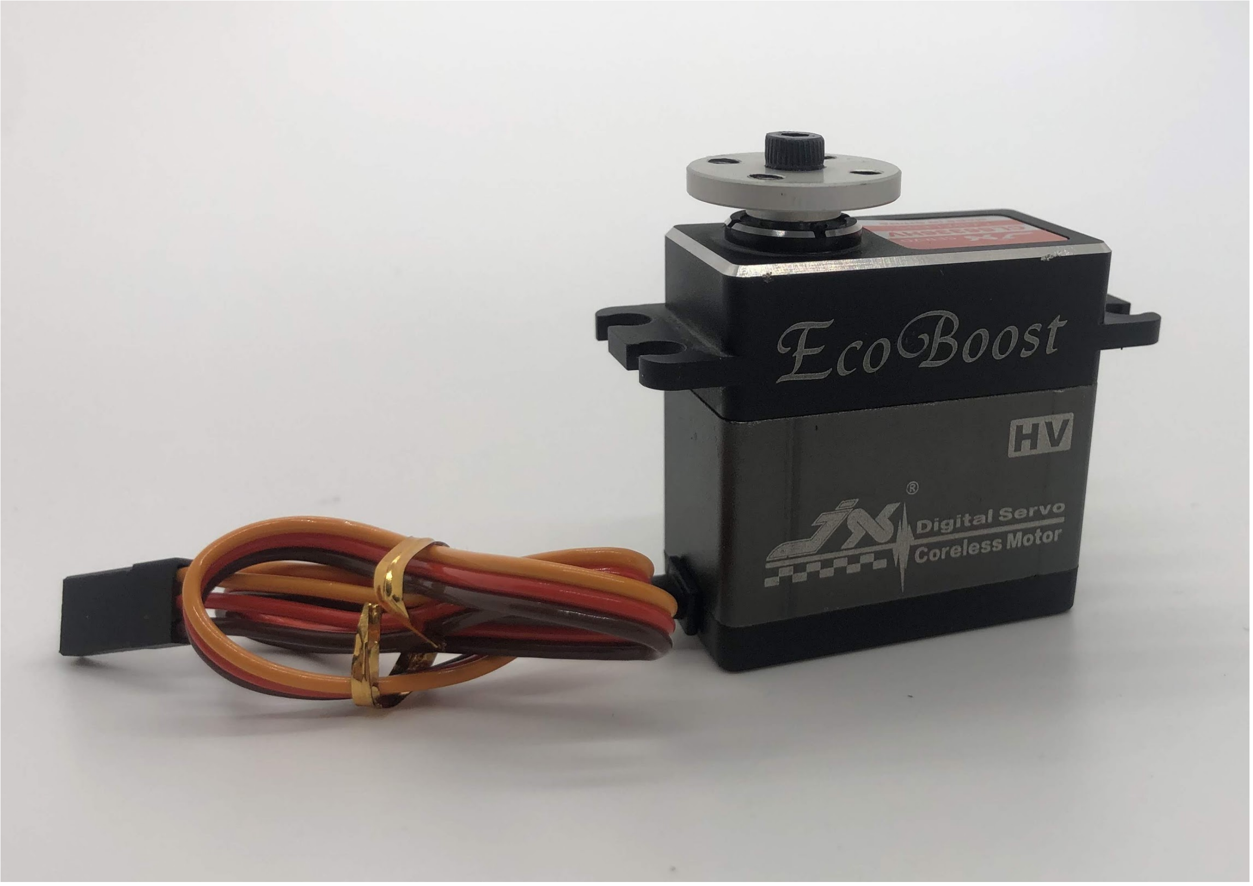
Instructions:

1. Attach a servo horn (not disc) to the servo and turn the servo into its neutral position and then remove the servo horn
2. Align the servo disc on the servo shaft so that the disc holes are roughly at 45 degree marks (see picture below)
3. Put a tiny dab of loctite on the screw
4. Gently start threading the screw in and then continue to screw it in, this will cause the disc to sink onto the servo spline (the output shaft)
5. Some of the servo discs are poorly manufactured, so if this is the case for you, it's expected that it'll take a lot of torque to start threading the screw. At some point however, the servo disc will finish deforming to fit the shaft and it'll become a lot easier to screw on. When the disc is fully seated, you'll again see an increase in torque and you should stop. Do not over-tighten the screw once the disc is fully seated or you risk breaking the servo motor.

Preparing for the assembly



Completed step



Step 2. Install the M3 threaded insert into the inner hip piece

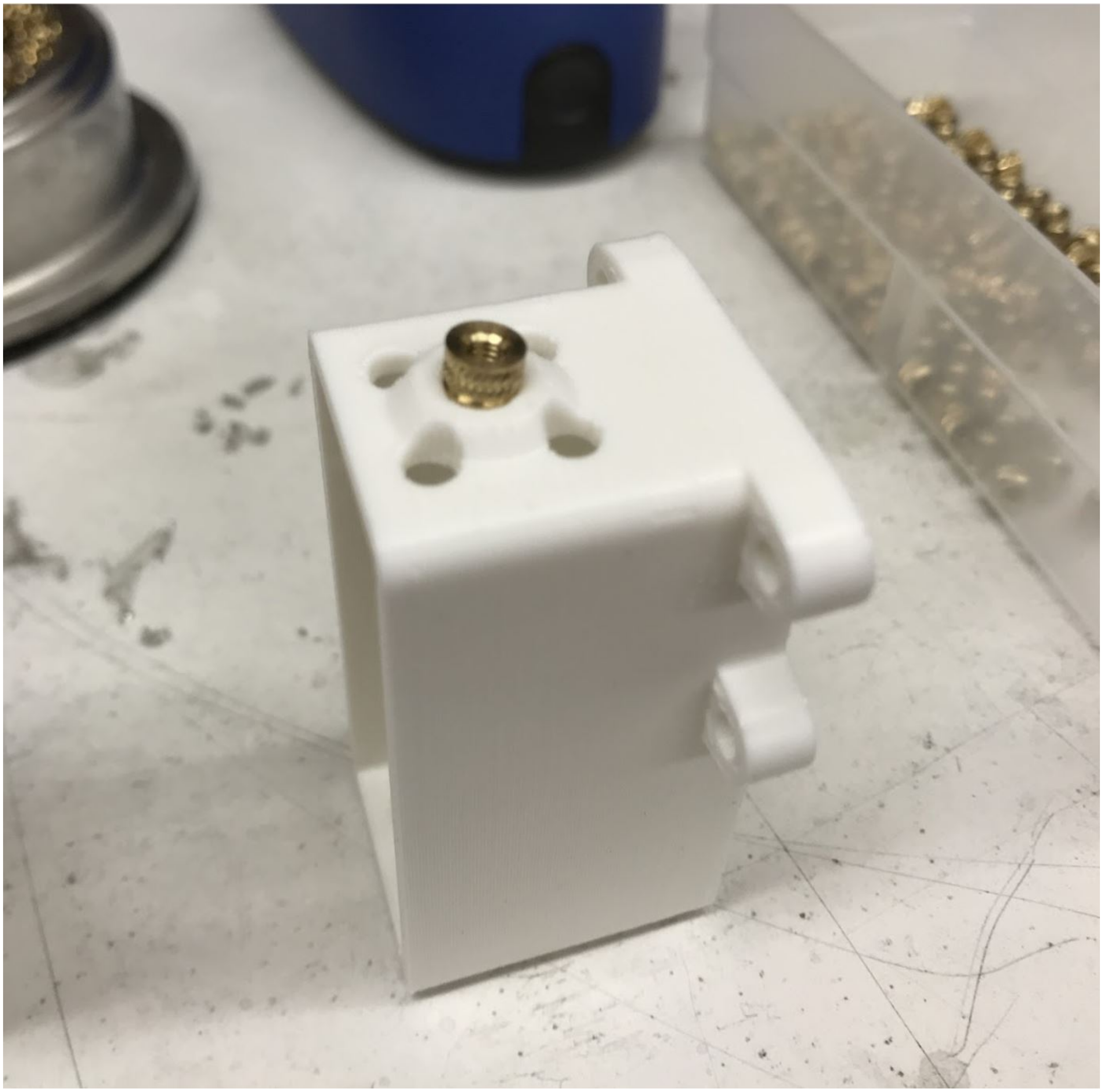
- No Video Available
- Materials: Assembly so far, M3 tapered threaded insert

- Tools: Soldering iron

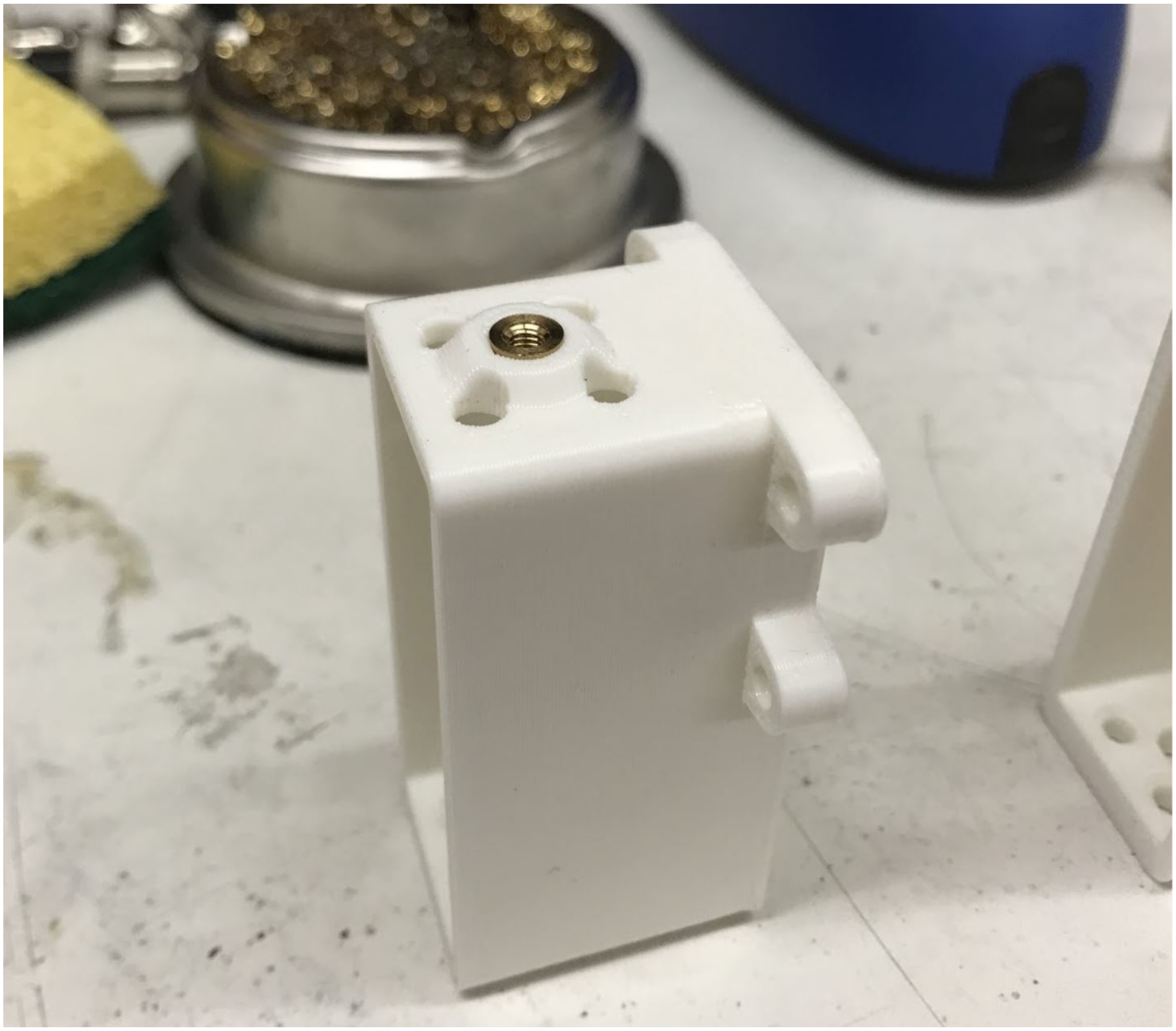
Instructions:

1. Place the insert into the hole with the tapered side down
2. Set the soldering iron to around 500f or 260c and then gently press the insert into the plastic. I recommend just using the weight of the iron to press the insert in, and I also suggest doing it in steps, ie pressing it in 1mm, taking the iron out, then pressing another 1mm etc into it is all the way in. This method prevents the soldering iron from getting stuck to the iron.

Setting up the insert for insertion. The actual 3D printed part in the picture is outdated, but the insertion is still the same.



Completed step



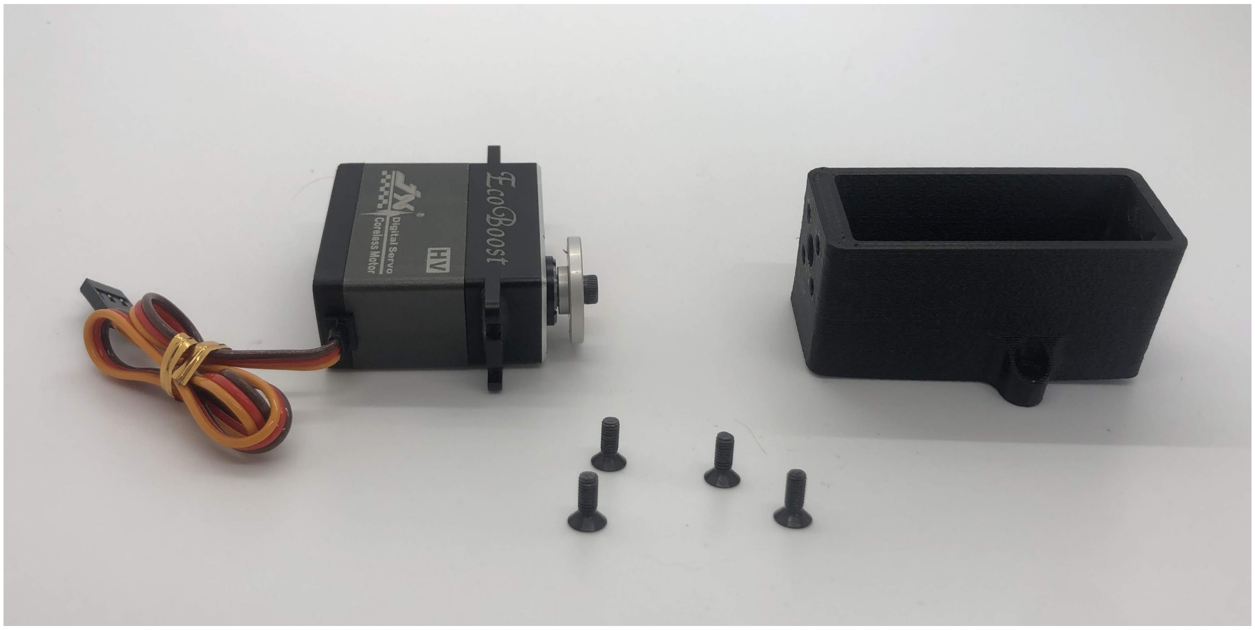
Step 3. Mount disc to inner hip part

- Video Instructions: https://youtu.be/qUoLoNEeEI8
- Materials: M3x8 flat head screw, Inner hip part, loctite
- Tools: 2mm hex driver

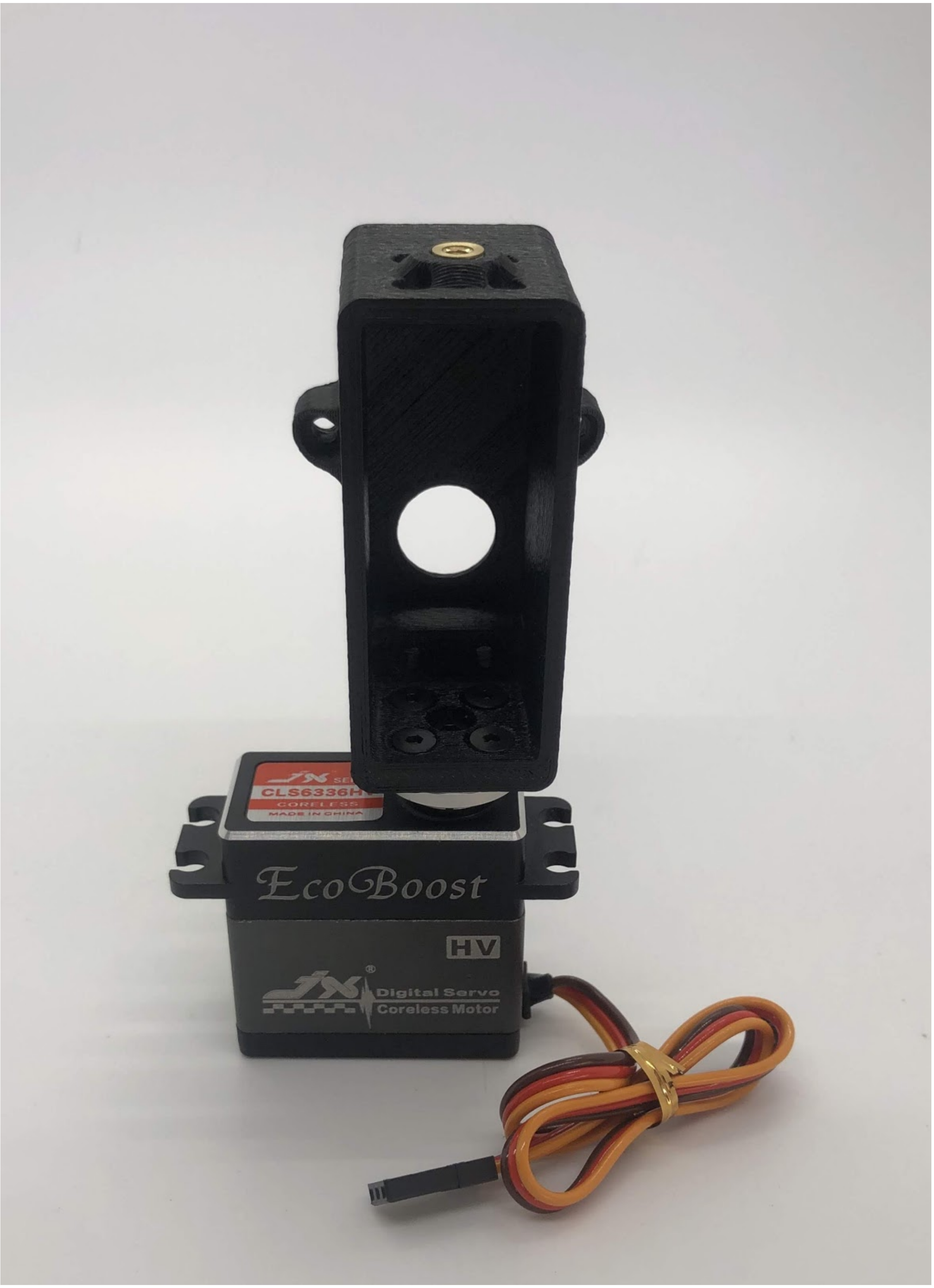
Instructions:

1. Install the inner hip part at a 90deg angle and then use the access holes on the other side to tighten down the M3x8 flat head screw attaching it to the servo disc.

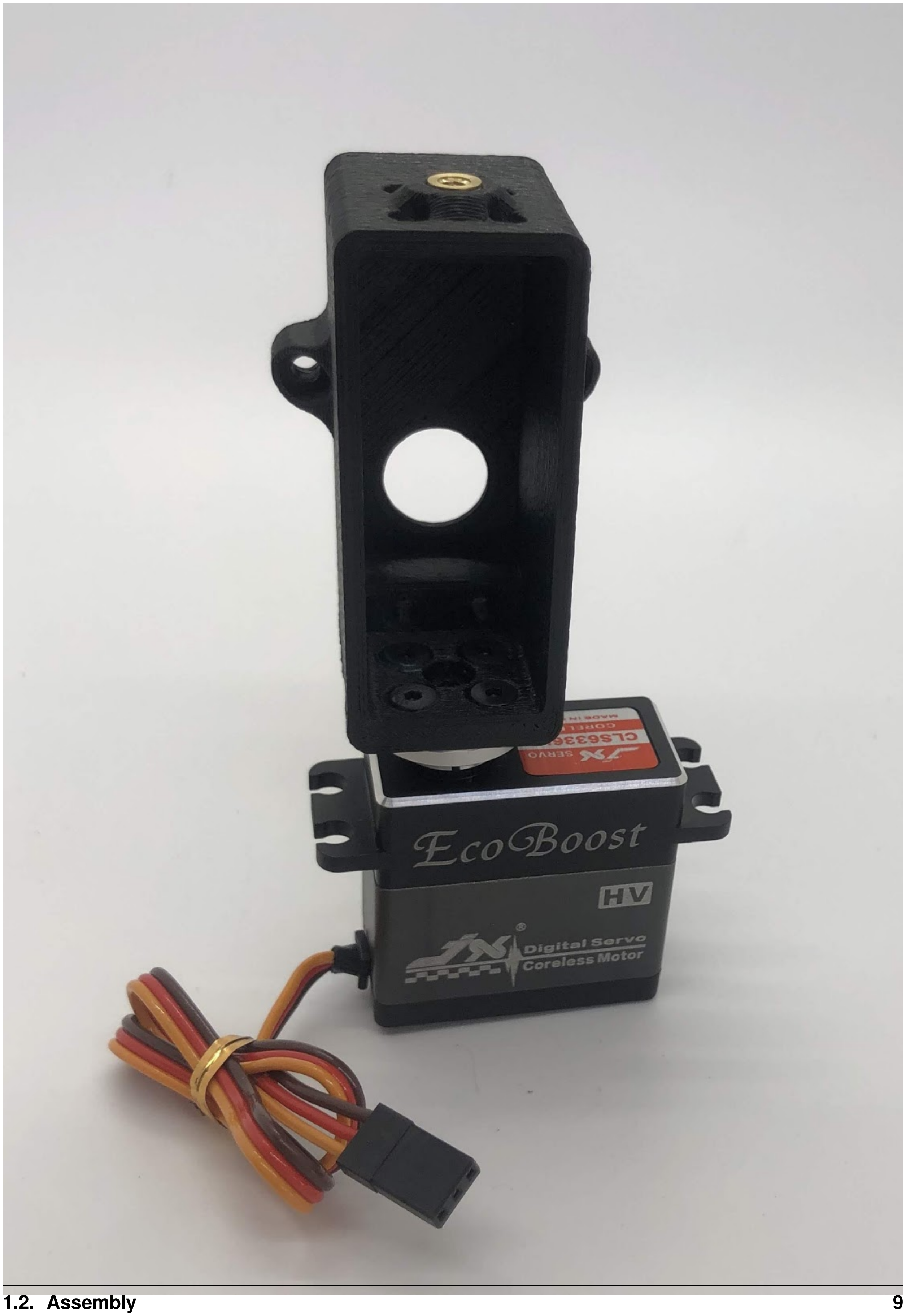
Preparing for the assembly



Completed step for left side


CLS6336H
CORELESS
MADE IN CHINA
EcoBoost
HV
Digital Servo
Coreless Motor

Completed step for right side


EcoBoost
HV
Digital Servo
Coreless Motor
SERVO

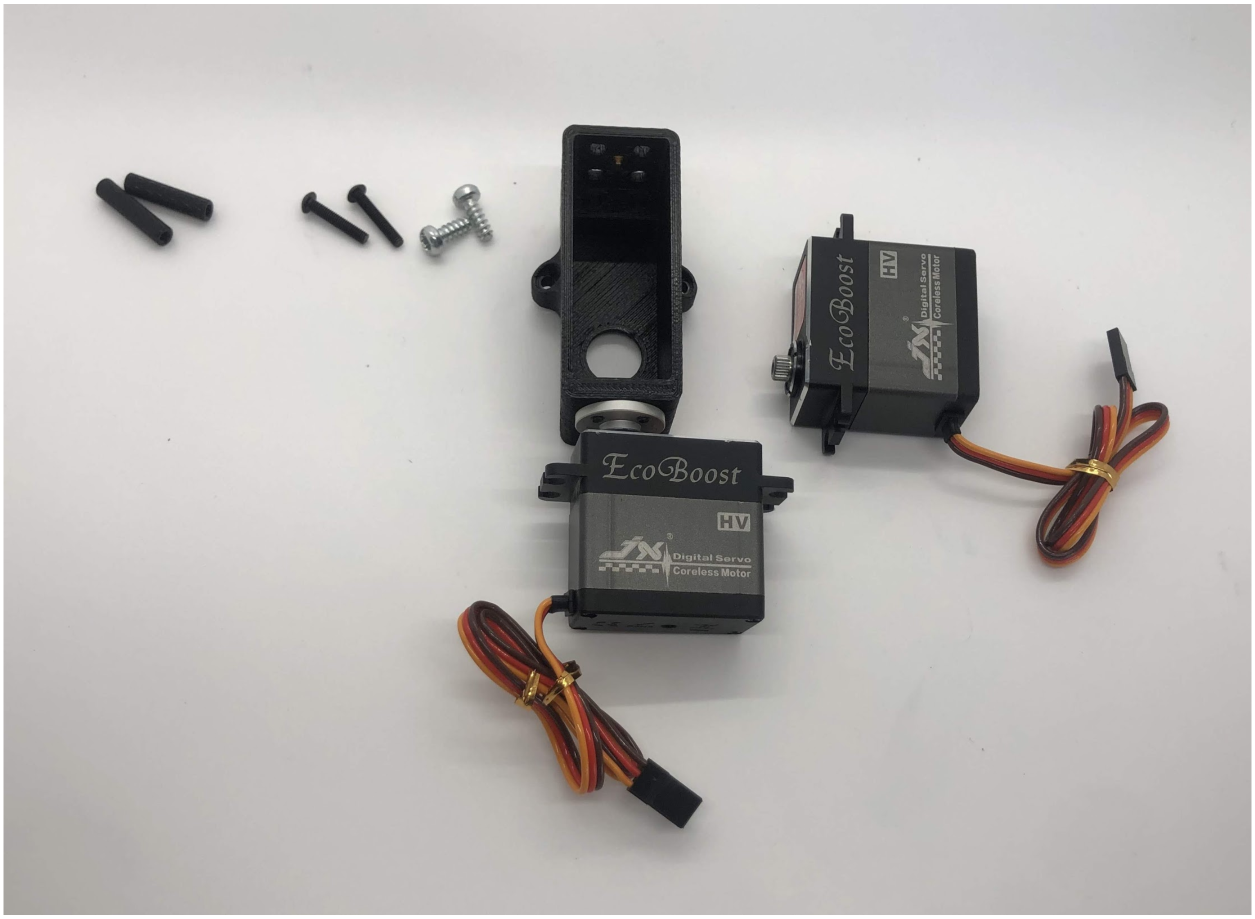
Step 4. Install the inner hip servo

- Video Instructions: https://youtu.be/6Rd2ZSjpYhM
- Materials: Inner Hip Assembly so far, servo, M4x10mm screws for plastic (silver), M3x16mm button head, 2x standoff
- Tools: T20H torx driver, 2mm hex driver

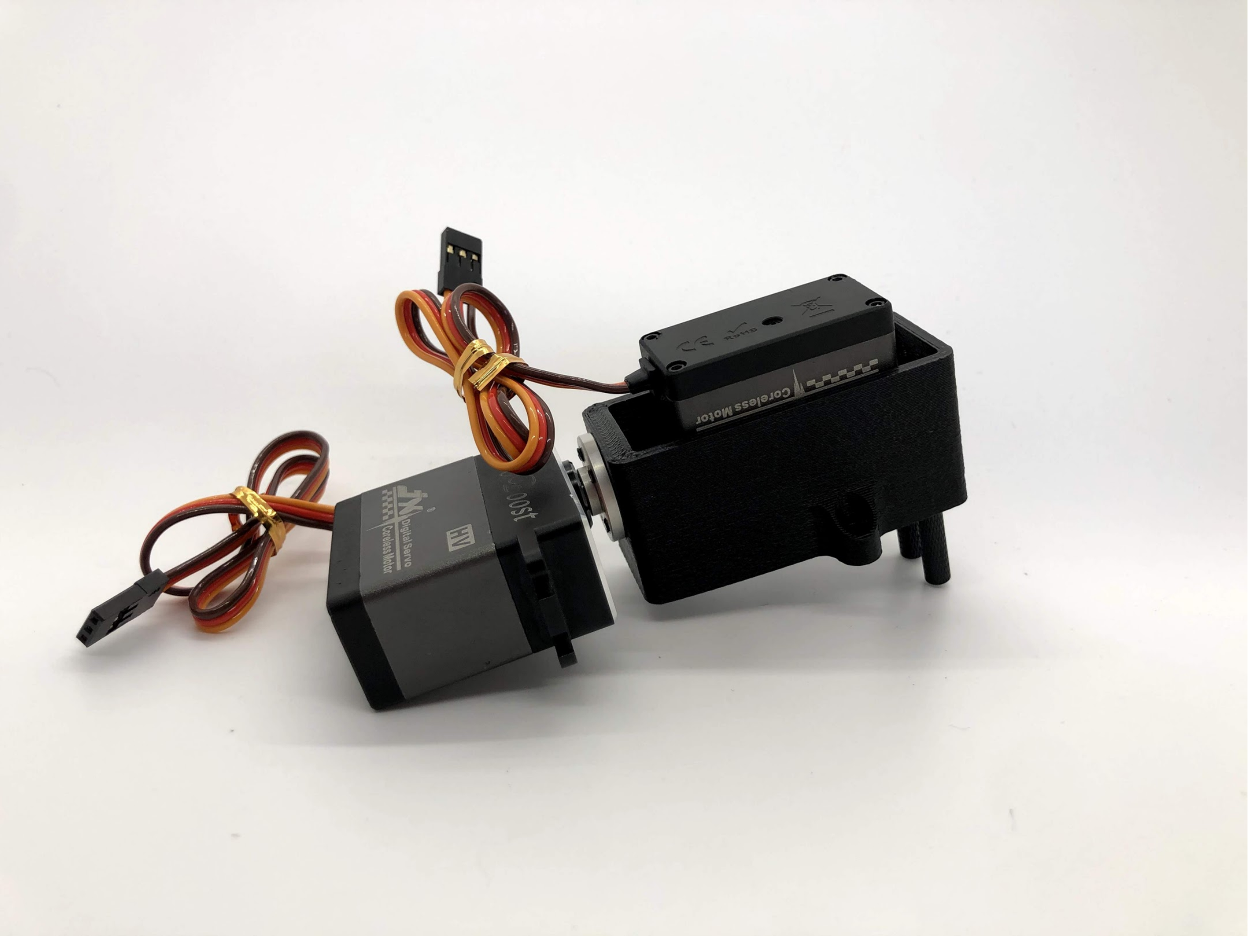
Instructions:

1. Place servo motor in the inner hip part and gently wiggle such that the servo shaft is sticking out of the big circular hole in the inner hip part
2. Screw the M4x10mm screws on the left side of the servo and the M3x16mm screws on the right side of the motor. Use locktite on M3x16mm screws
3. Turn over assembly and screw M3x16mm screws onto standoffs

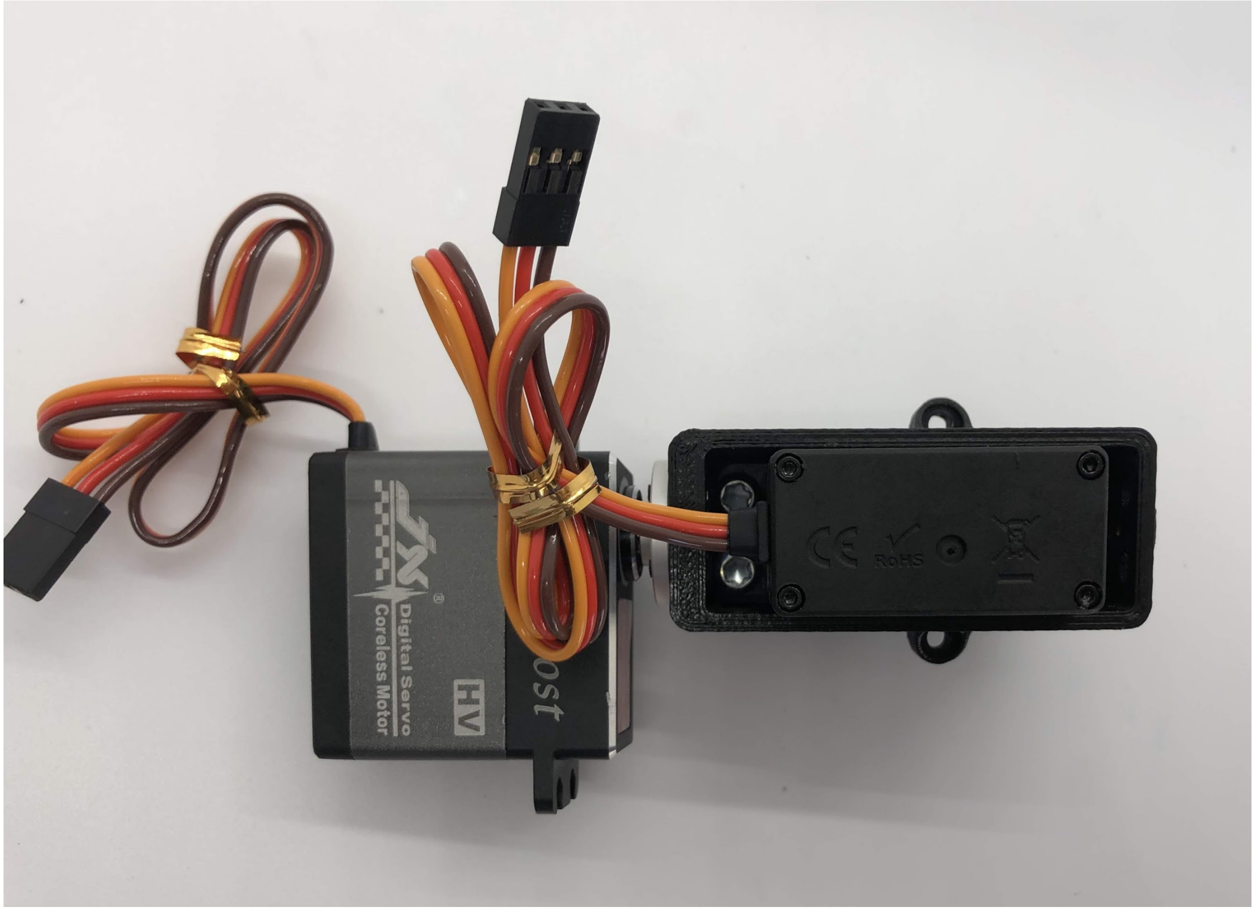
Preparing for the assembly



Completed step



Another look at the assembly. Note that the plastic screws are on the left, and the M3 screws are on the right



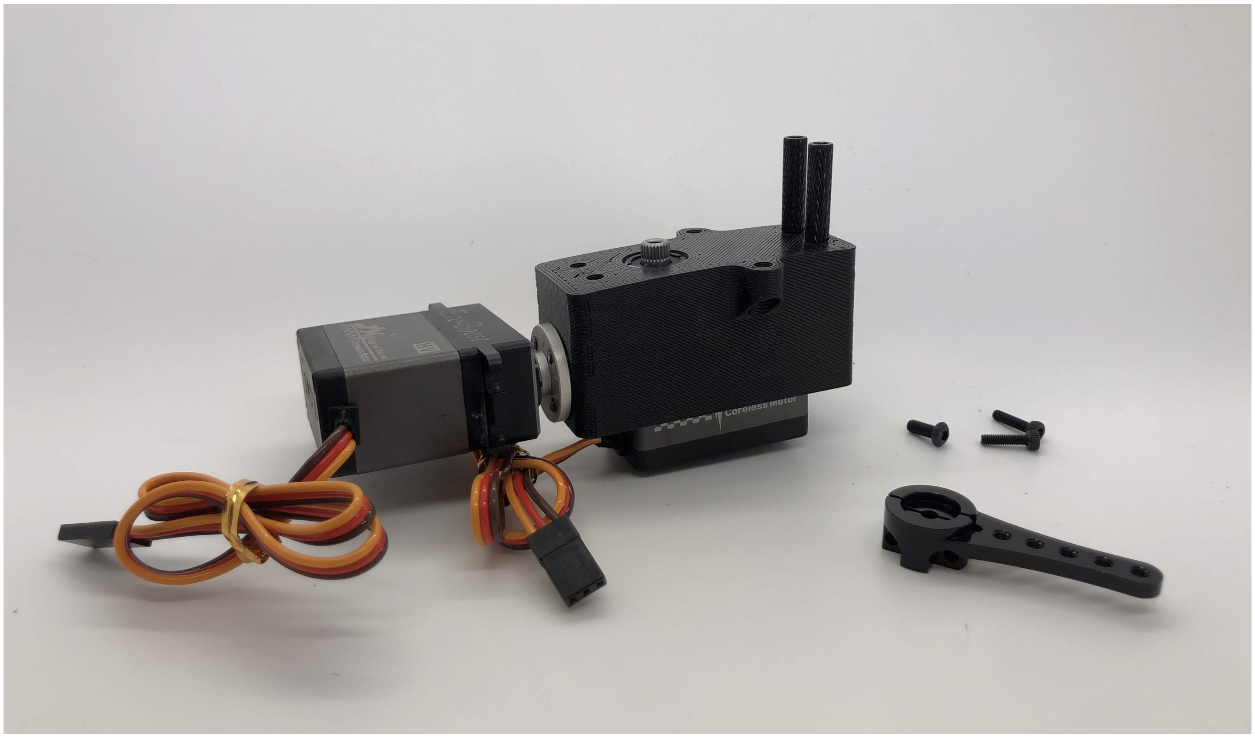
Step 5. Install servo horn on inner hip servo

- Video Instructions: https://youtu.be/wqRM8rbfDBM
- Materials: Inner Hip Assembly so far, M3x8mm button head screw, M2x8mm socket head screw, servo horn
- Tools: 2mm hex driver

Instructions:

1. Turn the servo into its neutral position and then slide the horn on at the angle shown (45 degrees downwards)
2. Screw the M3x8mm screw on to the top of the horn and screw the M2x8mm screws into the side of the horn
3. Don’t forget to use loctite!

Preparing for the assembly



Completed step for right side

EcoBoost
HV
JX
Digital Servo
Coreless Motor

Completed step for left side

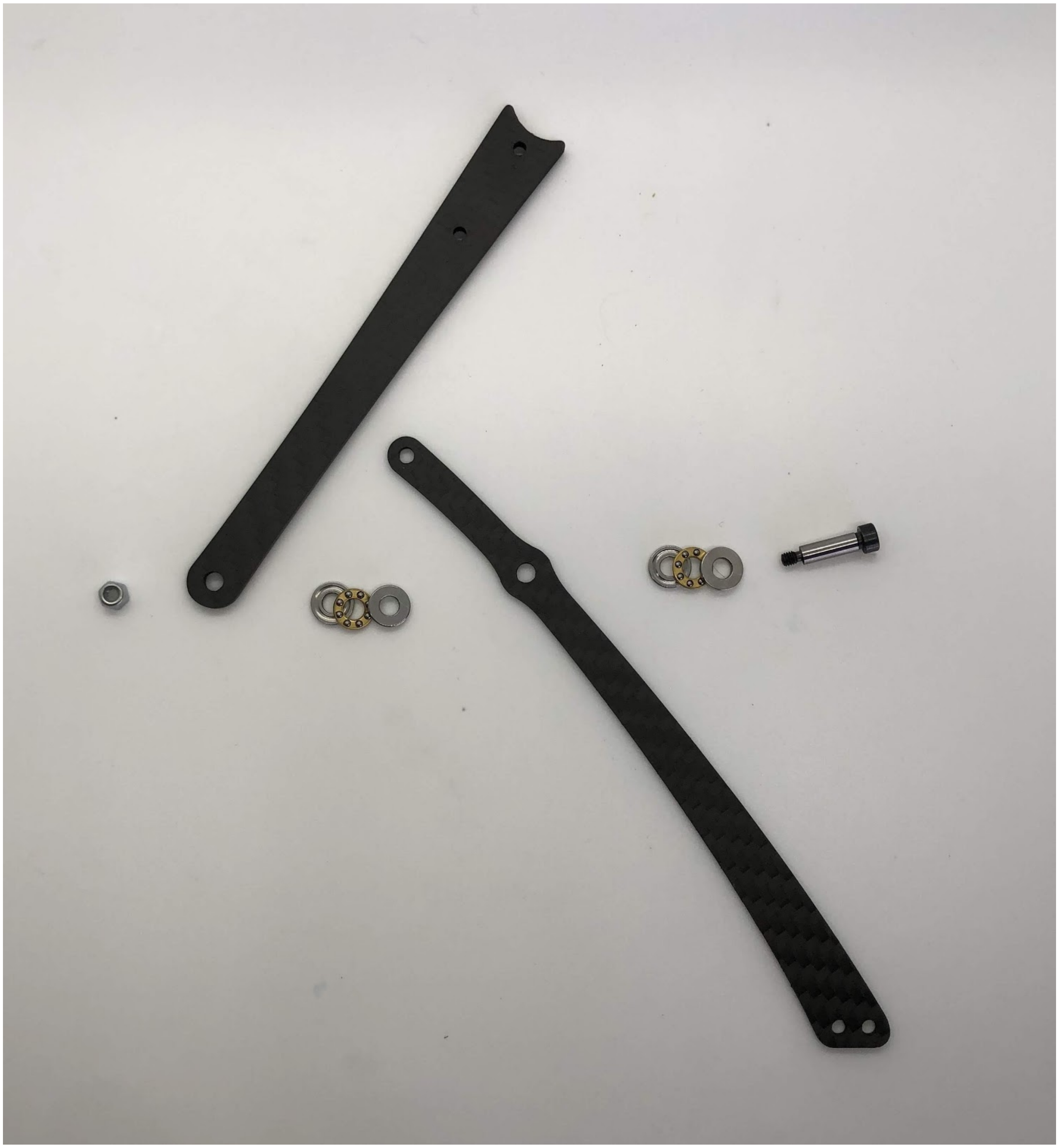
Step 6. Attach Leg

- Video Instructions: https://youtu.be/bMr0gCNQJxM
- Materials: Bottom Leg, Top Leg, 3-part Thrust Bearing x2, Shoulder Bolt, M3 Lock Nut
- Tools: 2mm driver, wrench for lock nut

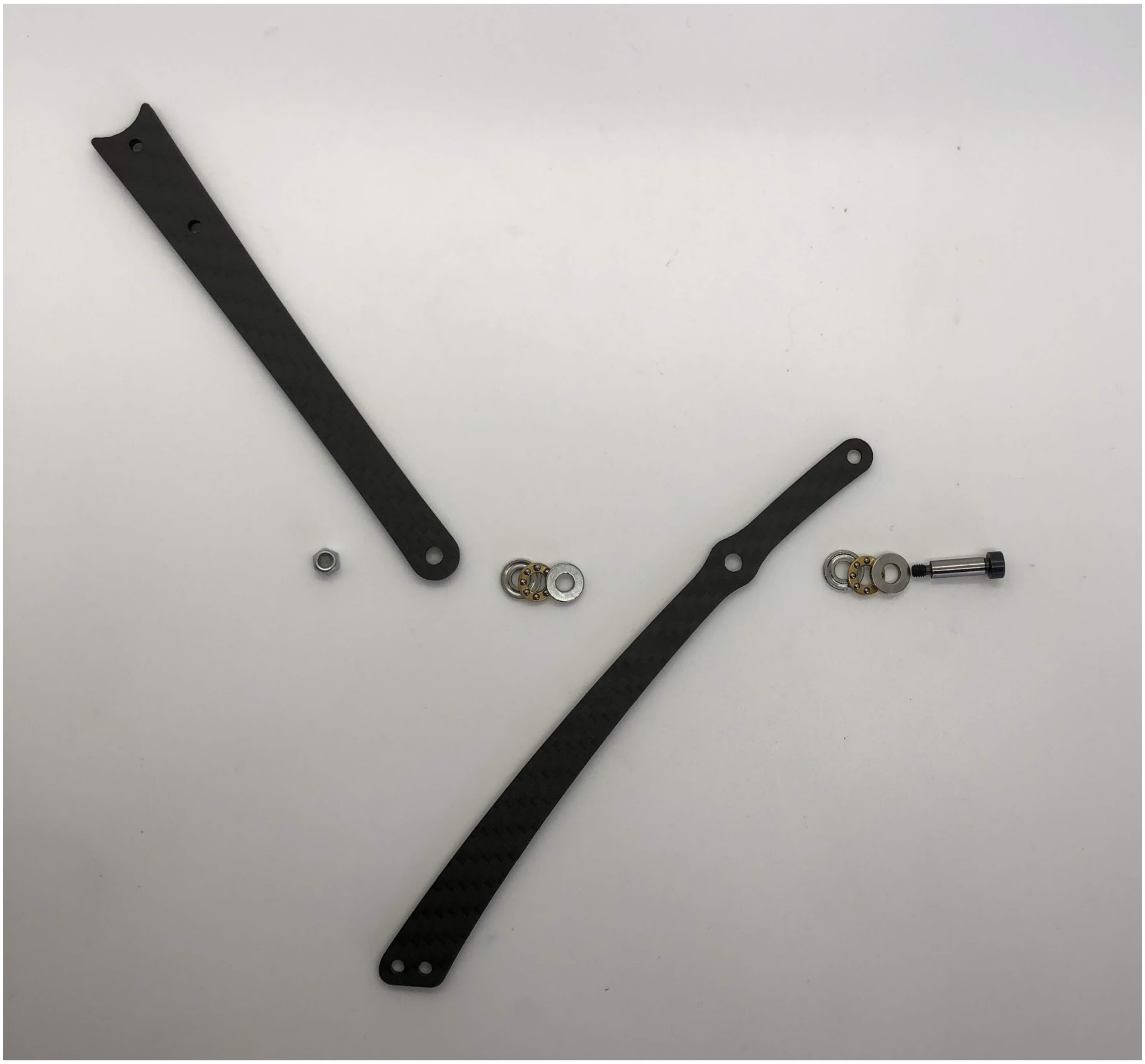
Instructions:

1. Add one 3-part thrust bearing on the shoulder bolt, then the Bottom leg, then another 3-part thrust bearing, then the Top leg then locking nut. Flip orientation of Bottom and Top leg accordingly for the left and right leg. See pictures for reference.

Preparing for the right assembly



Preparing for the left assembly



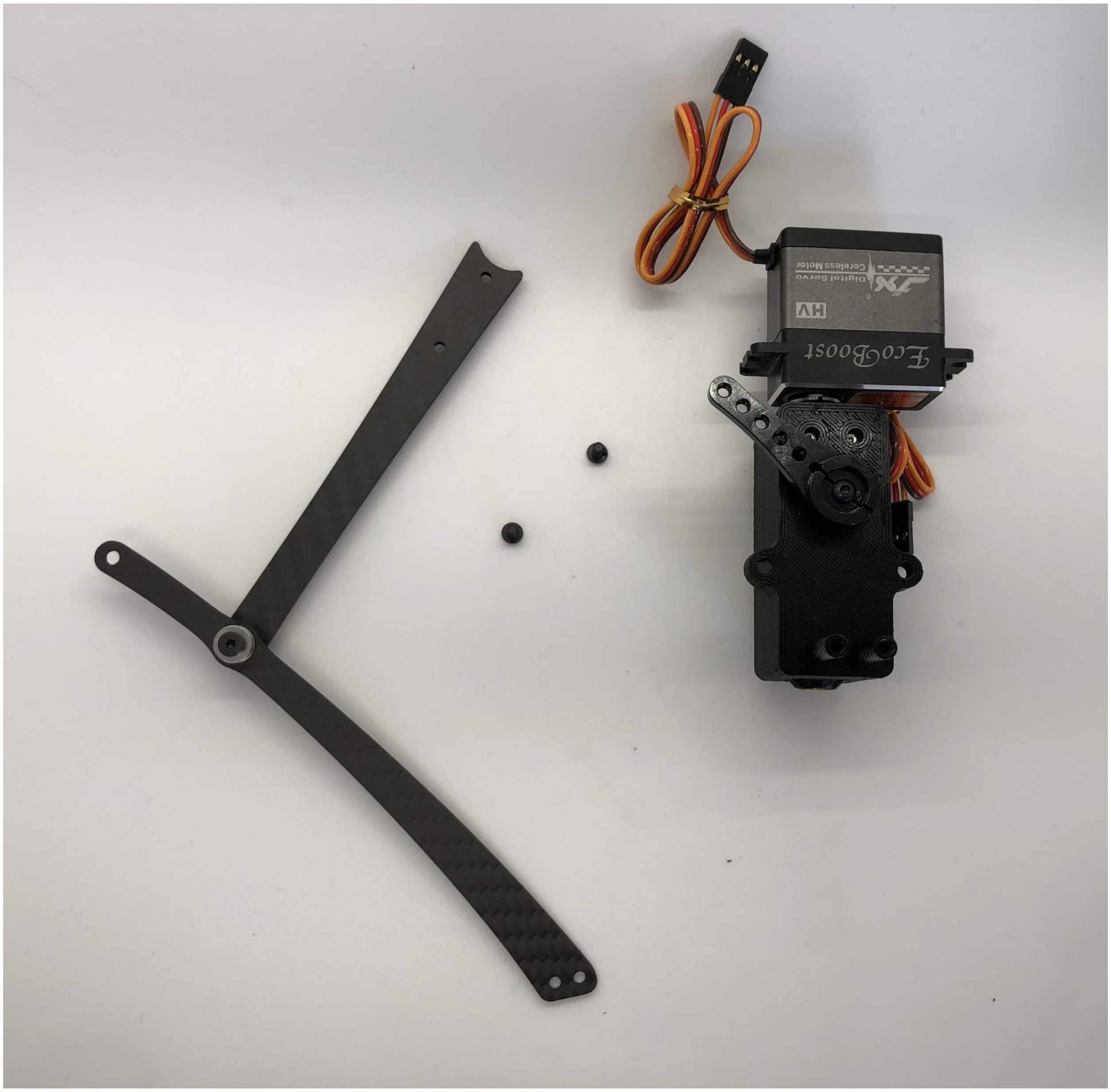
Step 7. Attach top carbon leg link to servo horn

- Video Instructions: https://youtu.be/Tp3HsjZY7qY
- Materials: Inner Hip Assembly, Leg Assembly, M3x6 Button Head x2.
- Tools: 2mm hex driver

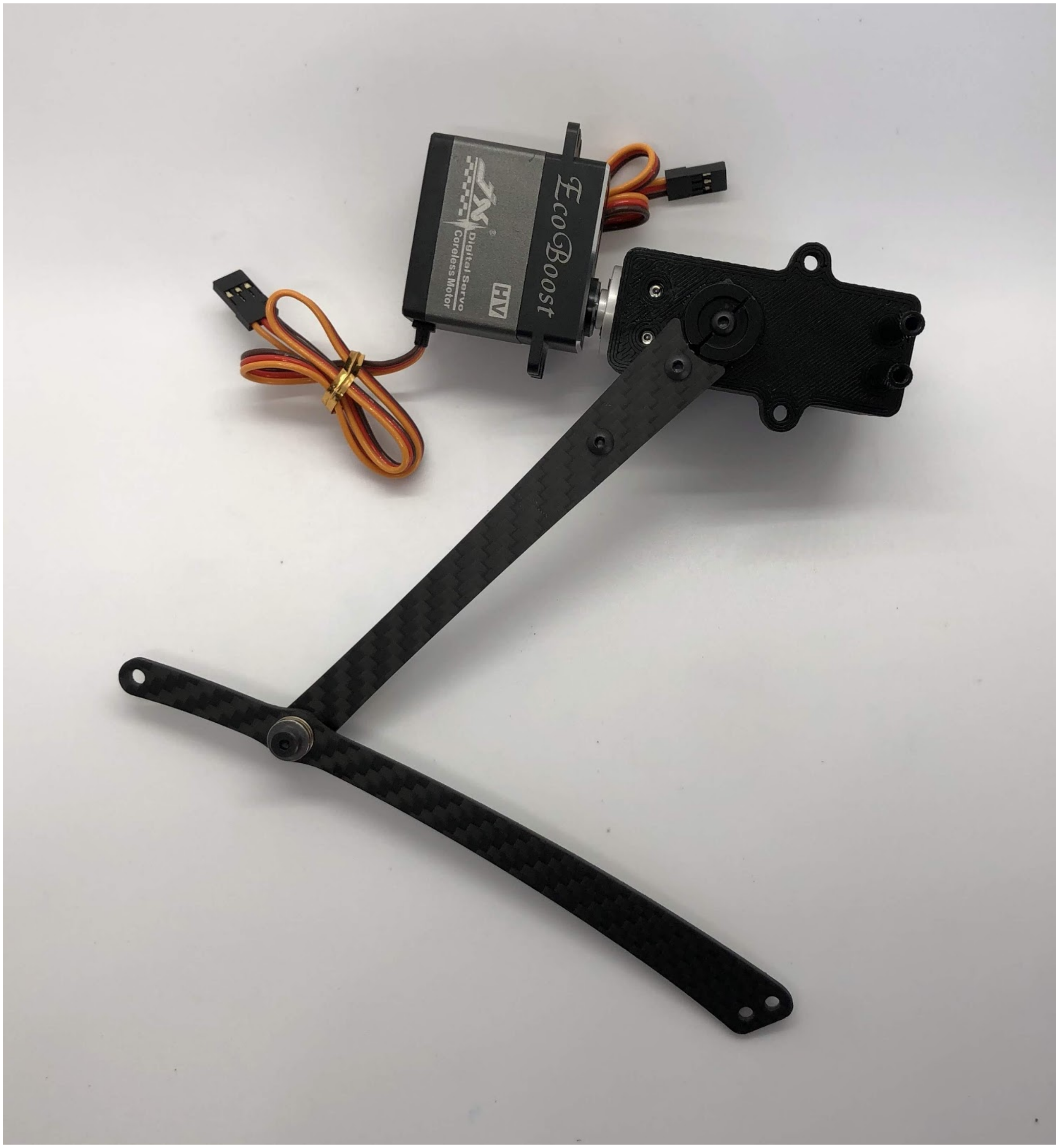
Instructions:

1. Align the curved edge of the left Top leg with the left Servo horn. Screw in the M3x6 button head screws through the carbon leg holes. Repeat for right side.
2. Be careful when seating the screw nearest to the servo to ensure it is vertical. It is necessary to hold the screw vertically to avoid cross threading.

Preparing for the assembly



Completed step for right side



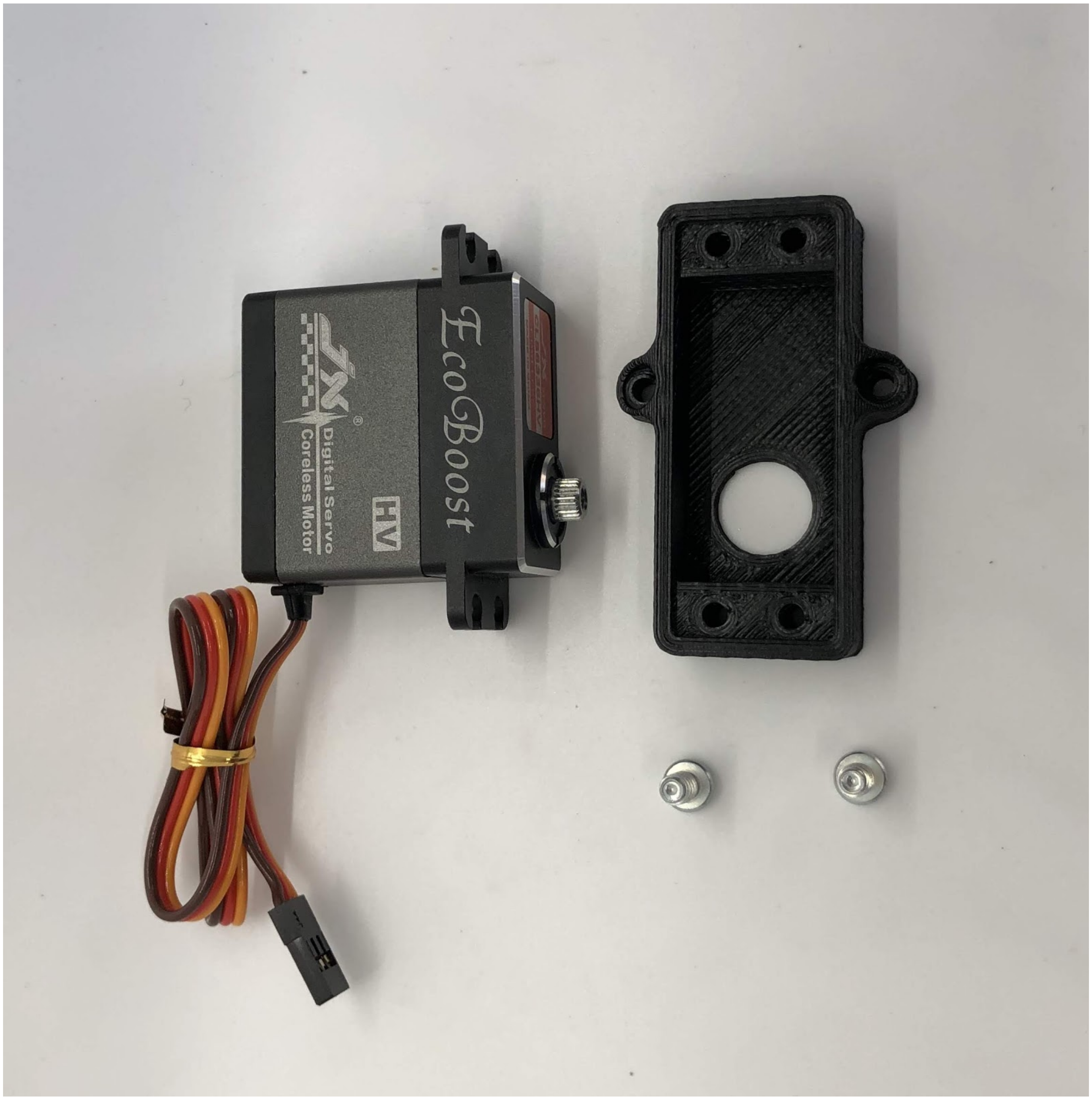
Step 8. Install outer hip assembly

- Video Instructions: https://youtu.be/iIqjgKaIPs8
- Materials: Servo, outer hip part, M4x10mm screw plastic
- Tools: T20H torx driver

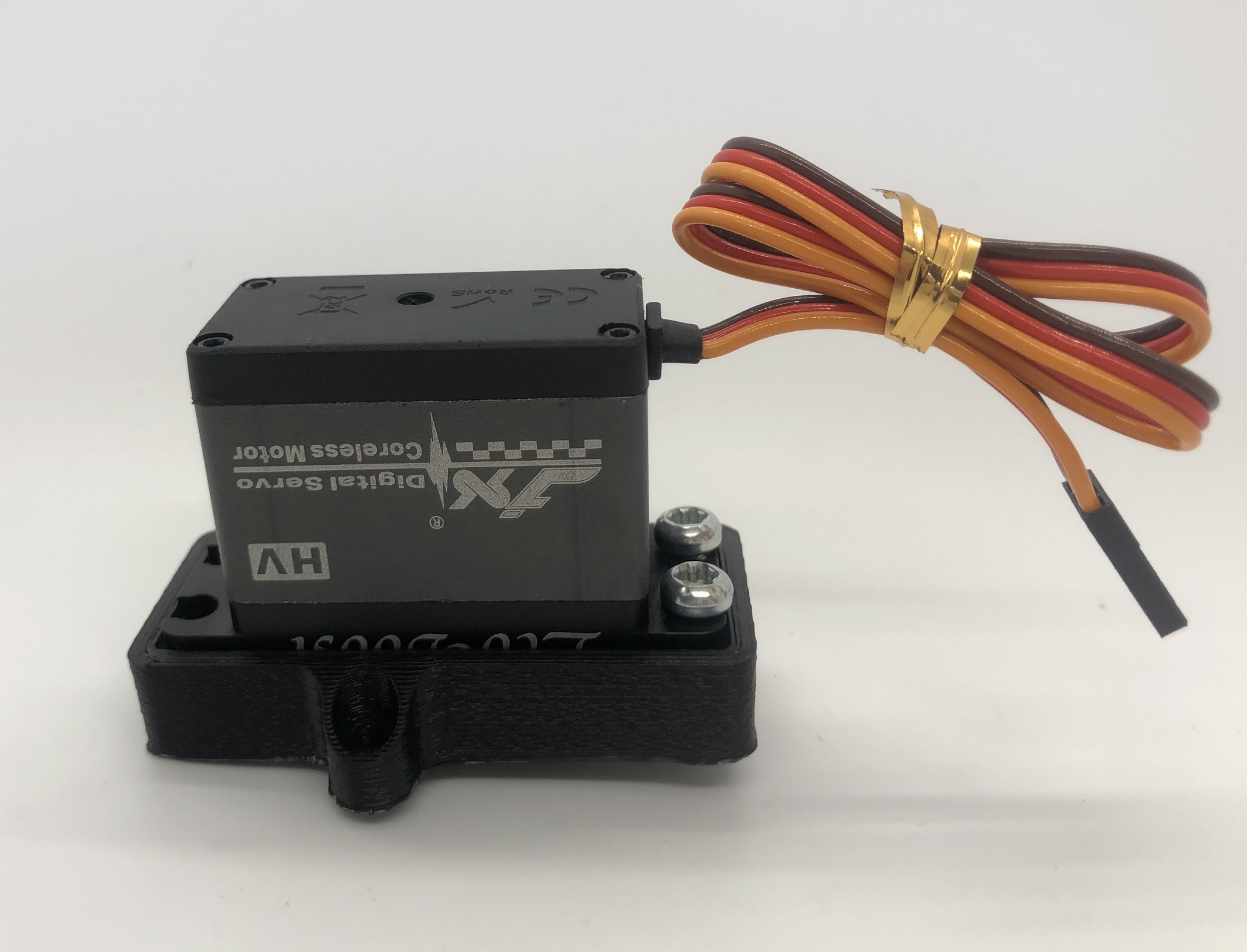
Instructions:

1. Place servo into joint and add affix with two screws closest to the servo spline

Preparing for the assembly



Completed step



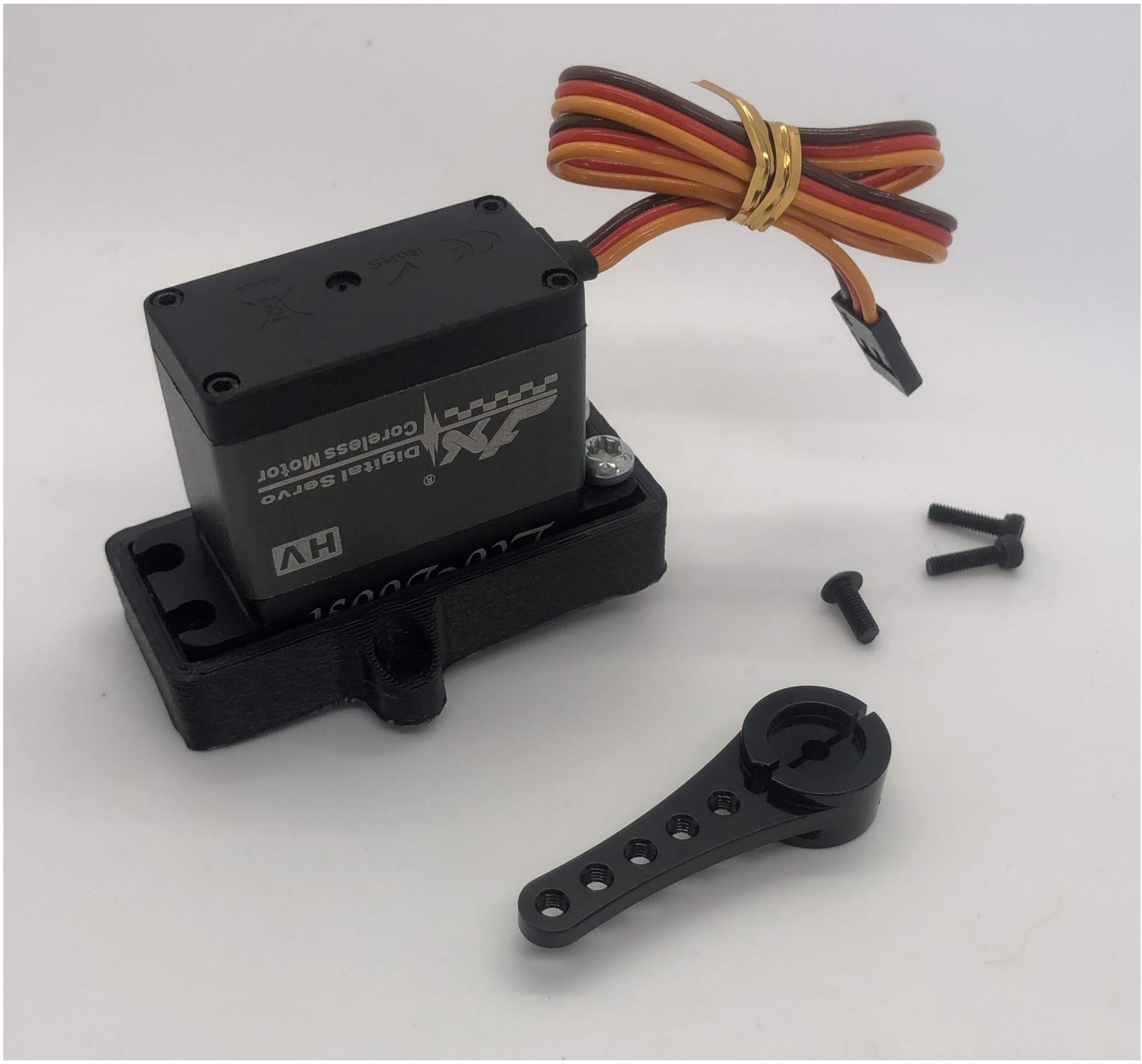
Step 9. Install servo horn on outside servo

- Video Instructions: https://youtu.be/Tj7zx2M6xas
- Materials: Servo horn, Outer Hip assembly, M3x8 button head, M2x8 socket head
- Tools: 2mm hex driver

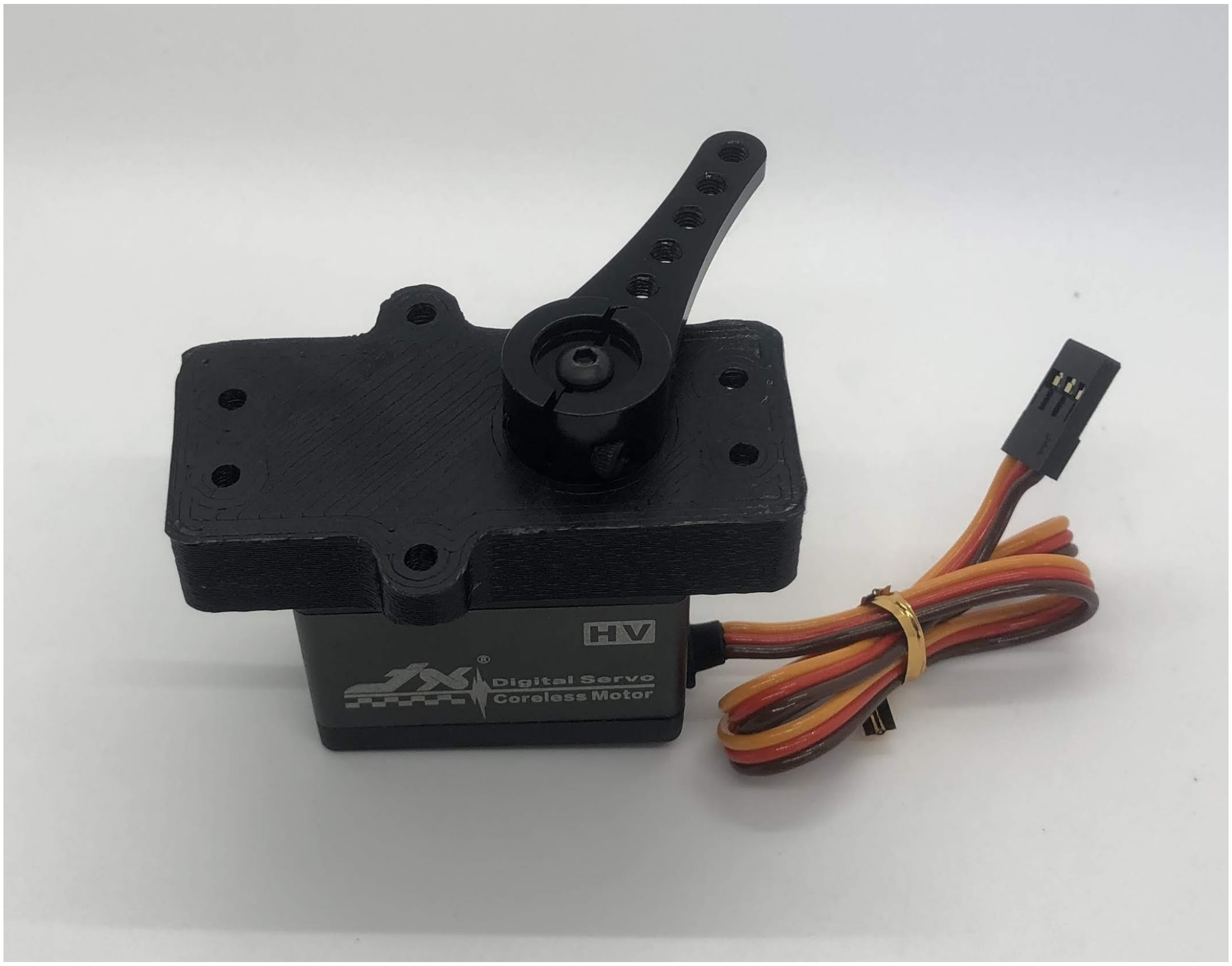
Instructions:

1. Turn the servo horn to its neutral position and then attach the horn at a 45 degree angle as shown.
2. First tighten the servo horn down with the M3x8, then add the M2x8 screws to tension the servo horn. Similar to Step 5.

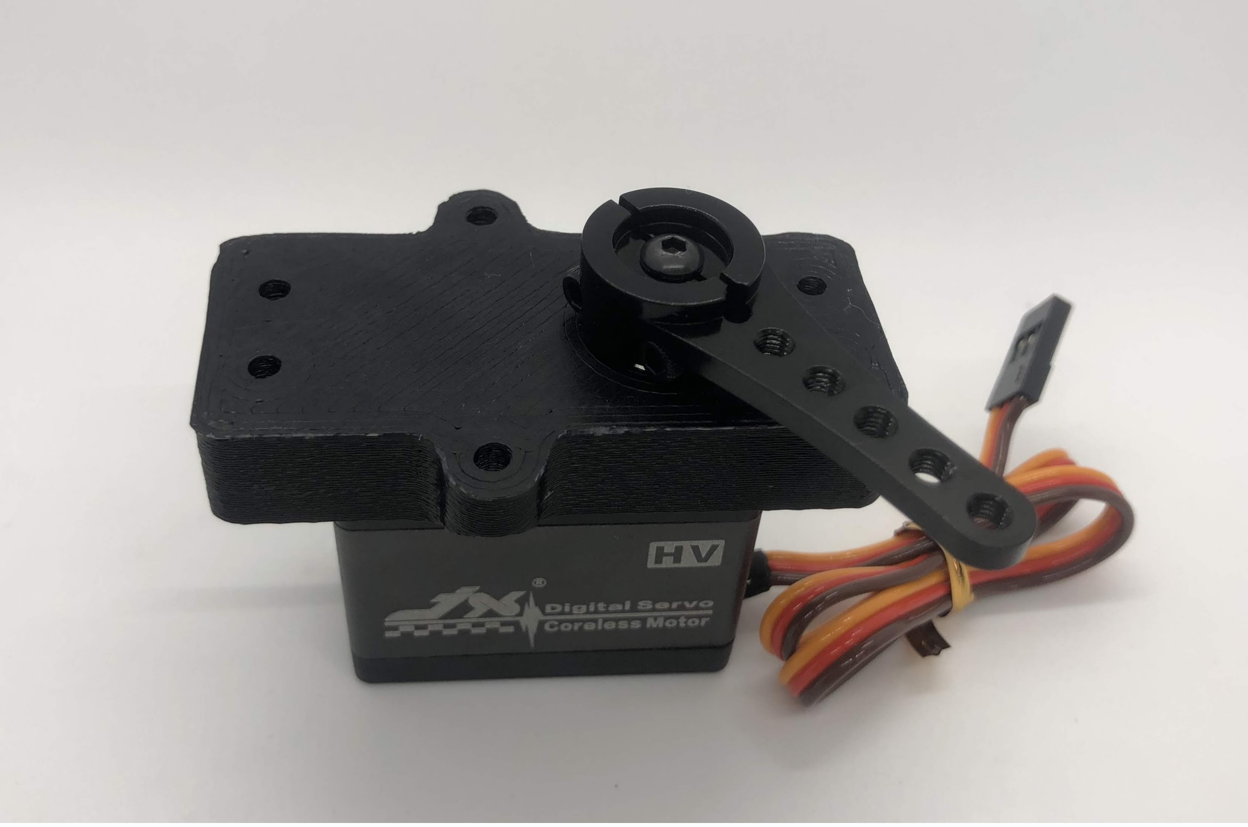
Preparing for the assembly



Completed step for right side



Completed step for left side



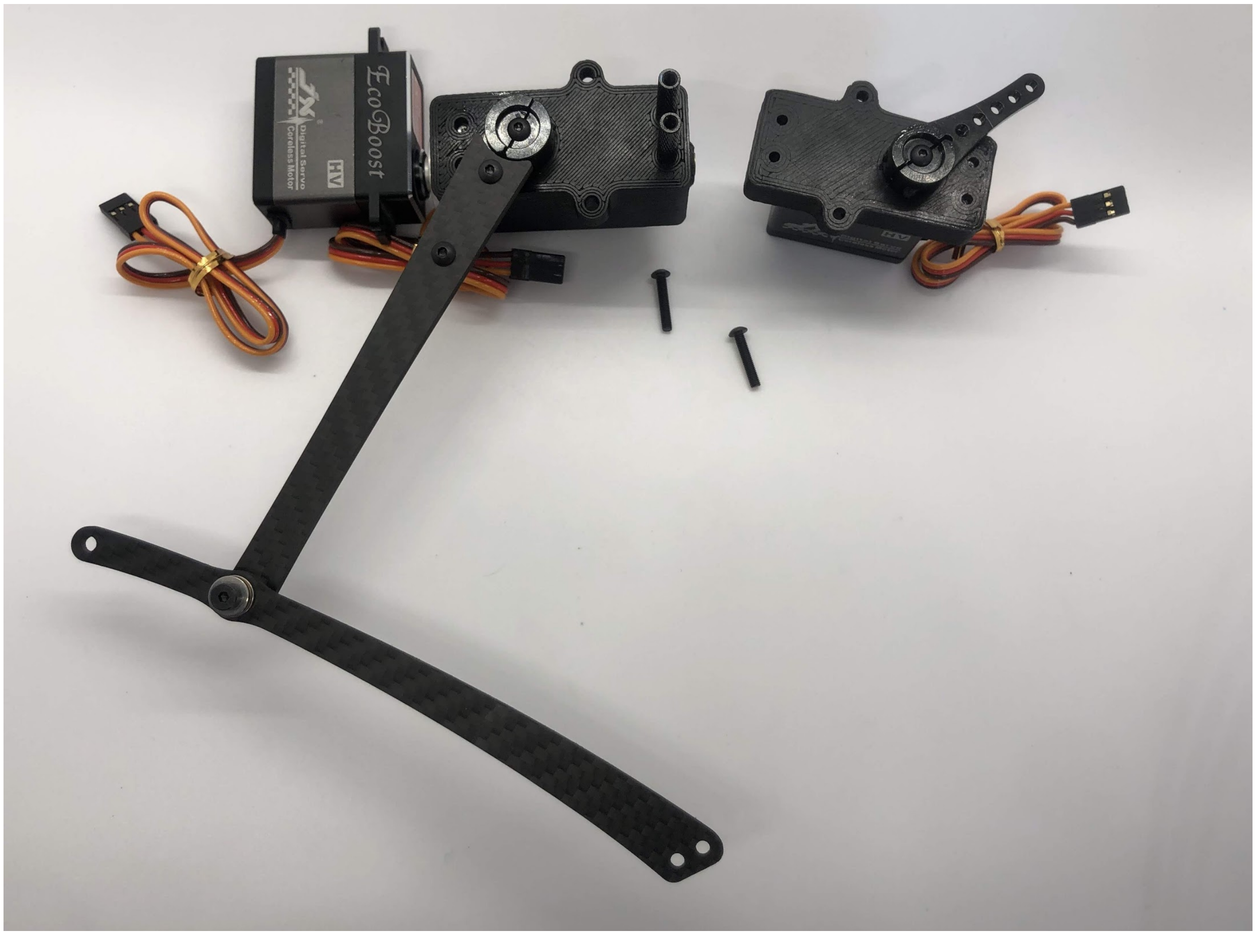
Step 10. Assemble the two sides

- Video Instructions: https://youtu.be/dKv7VrdE290
- Materials: Inner and Outer Hip assembly, M3x16 button head screws for screwing into standoffs, loctite
- Tools: 2mm hex driver

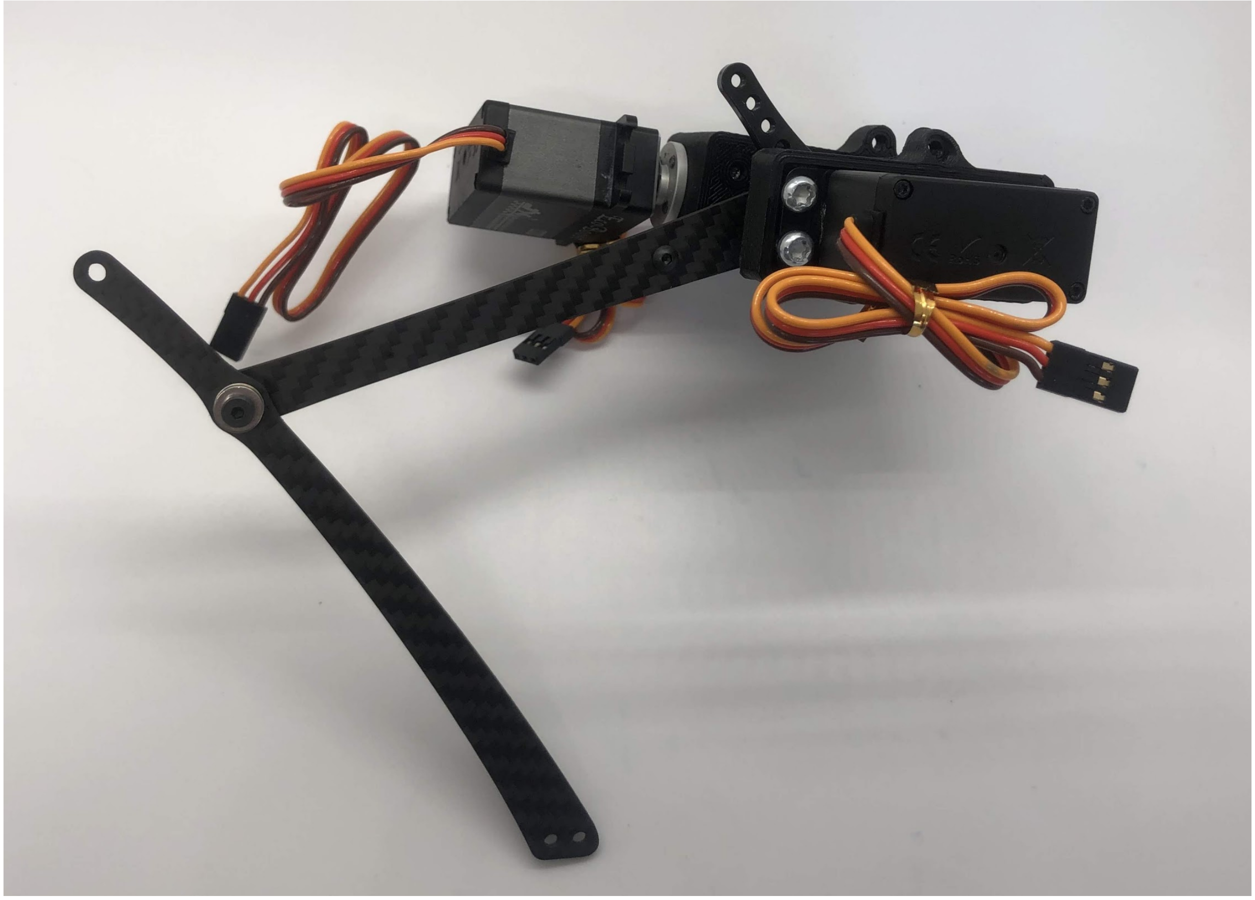
Instructions:

1. Align Inner and Outer Hip assembly, M4x10mm plastic screws should be on the same side and servo horns should be at a 90degree angle.
2. Connect assemblies with M3x16 screws through Outer Hip assembly to standoffs. Add loctite on screws. Don't tighten the screws down all the way yet.
3. At this point, your legs might start to move, feel free to mark your left and right side so you don't get confused. If you don't know which side is which, compare with the 3D model: https://stanford195.autodesk360.com/g/shares/SH919a0QTf3c32634dcfedf61e031f673710

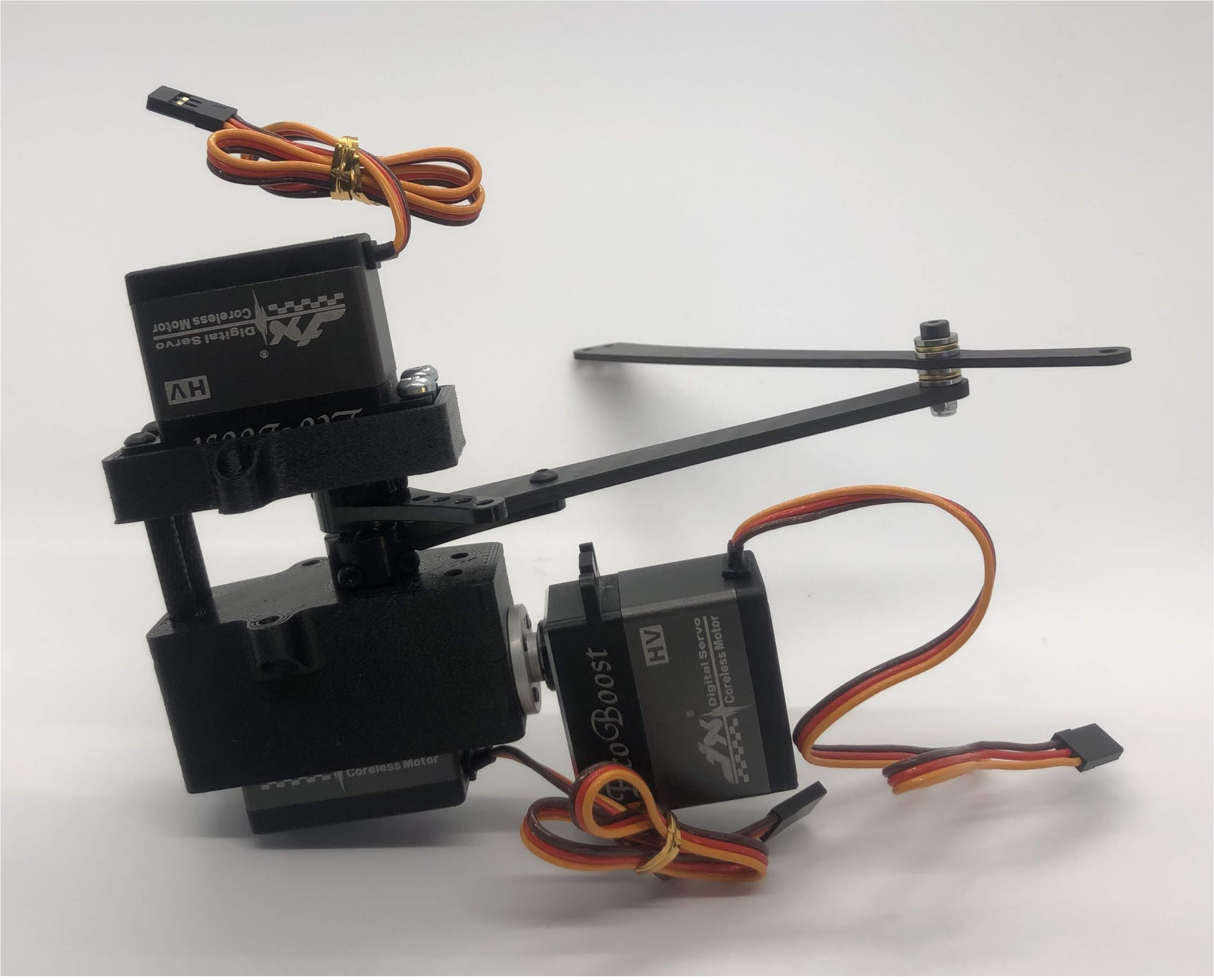
Preparing for the assembly



Completed step



Another look at the assembly



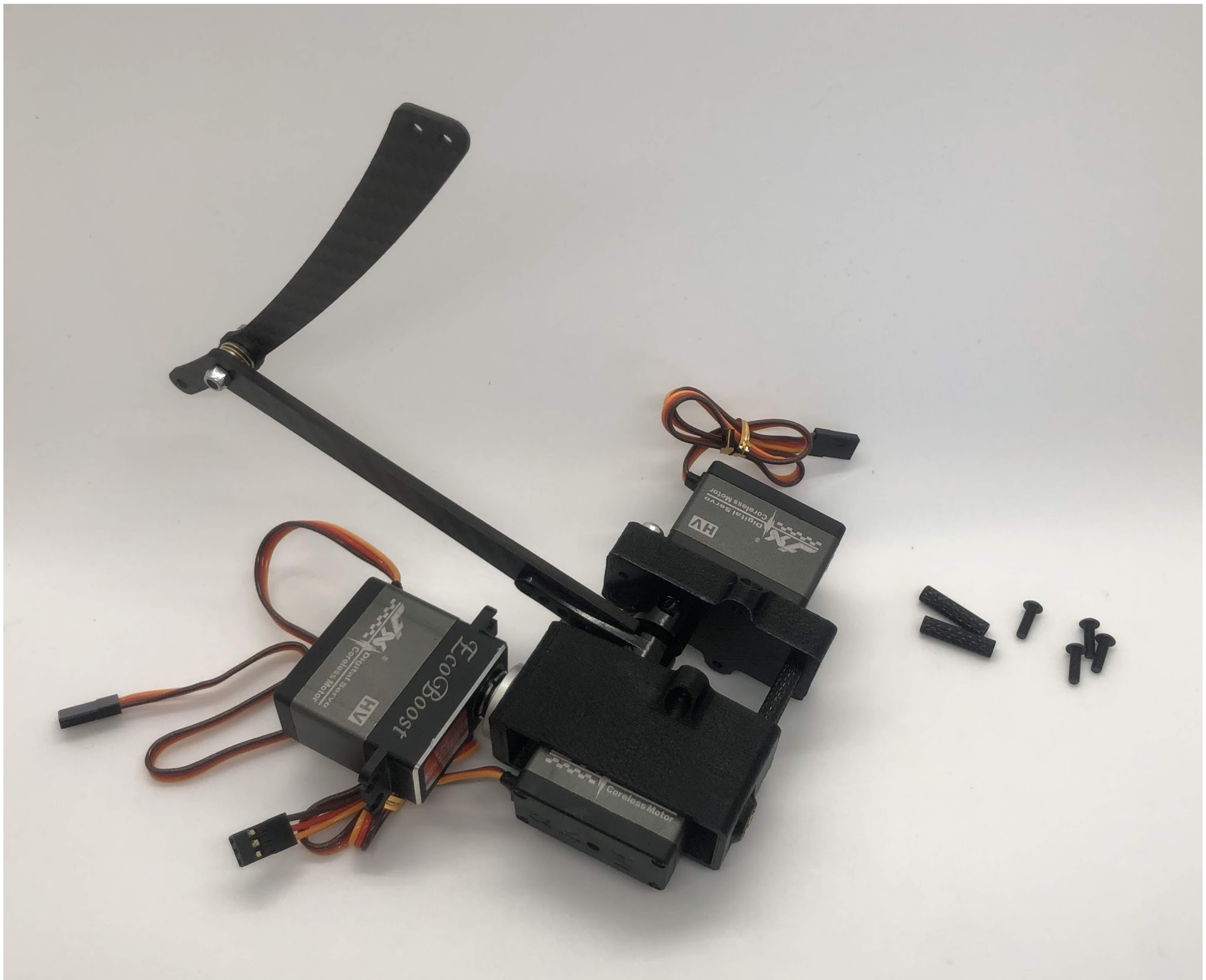
Step 11. Assemble the other 2 standoffs

- Video Instructions: https://youtu.be/nD_yWAIB70c
- Materials: Assembly, 4 M3x10 button head screws, 2 standoffs
- Tools: 2mm hex driver

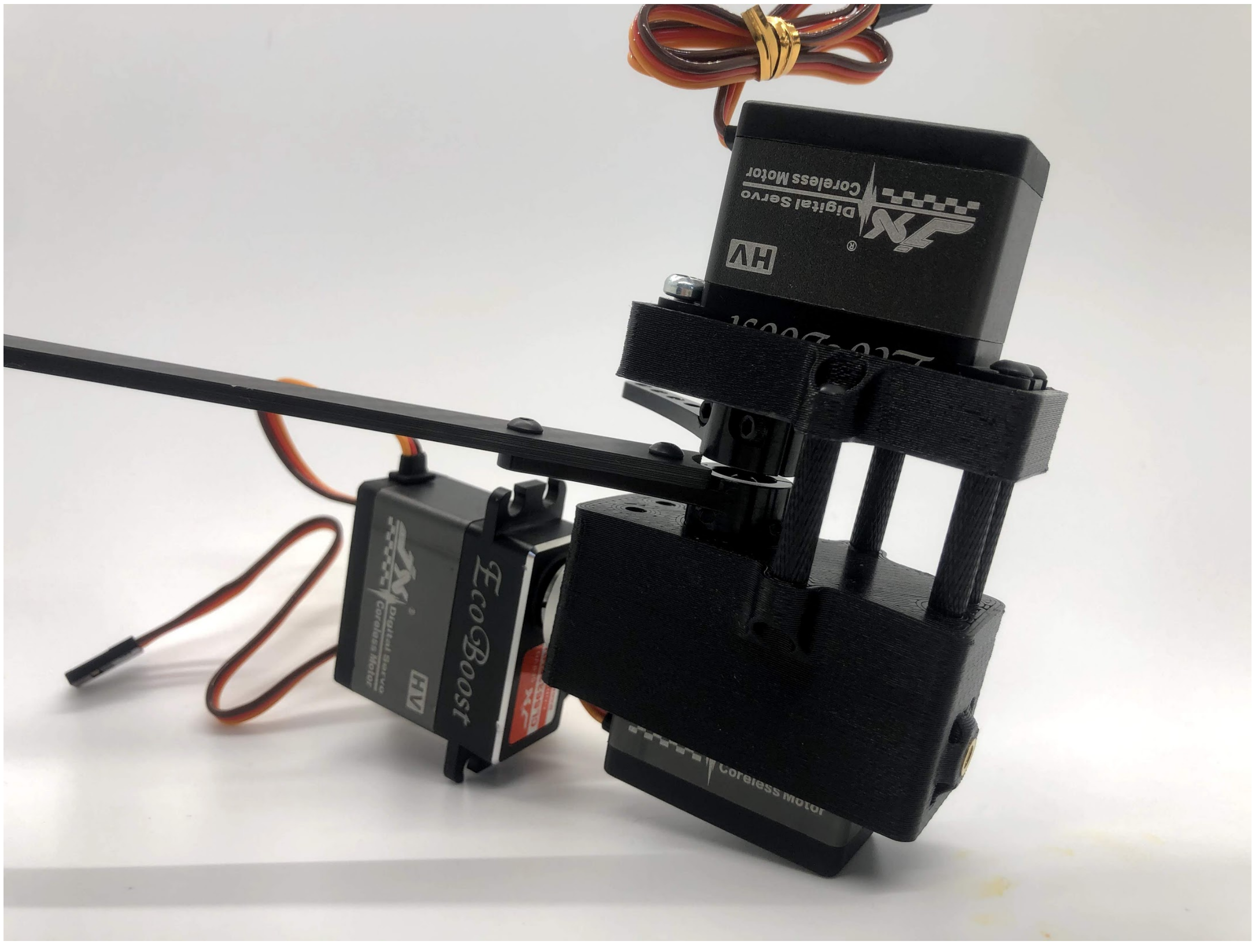
Instructions:

1. Install the other 2 standoffs and fasten with M3x10 button head screws

Preparing for the assembly



Completed step



Step 12. Test the full range of motion for each servo

- Video Instructions: https://youtu.be/gvaUp9pQ-W4

Instructions:

1. Hip should go the fully flat on either side
2. The horn nearest the body should go from 45 degrees upward to fully touching the lower standoff
3. The horn away from the body should go from touching the standoff upwards to going 45 degrees downward

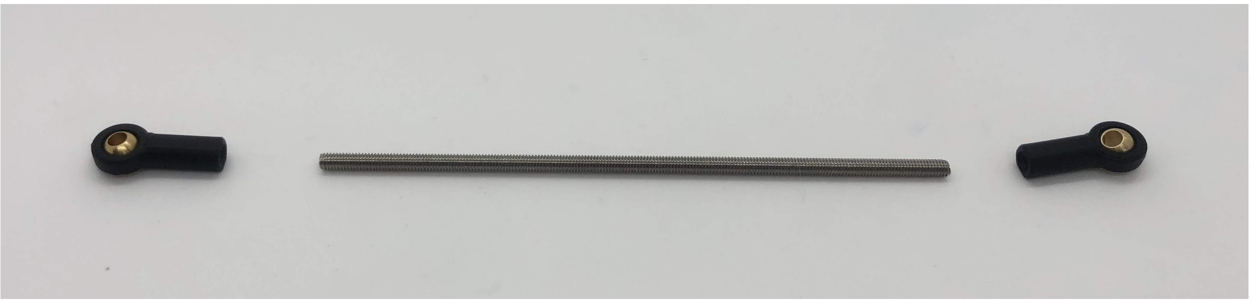
Step 13. Assemble the upper leg extension rod

- Video Instructions: https://youtu.be/4e2r8jGPv5Q
- Materials: Threaded rod, rod end x 2
- Tools: None

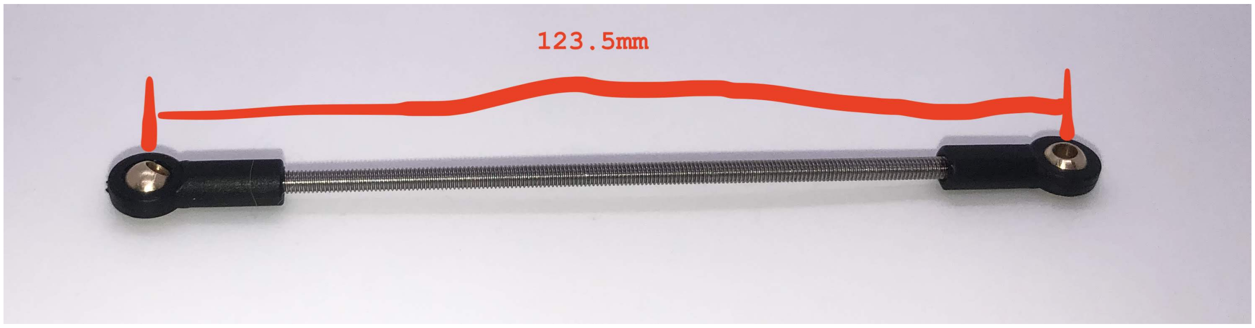
Instructions:

1. Screw the rod ends on equally until the distance between the center of the holes in the rod ends is 123.5mm. The goal here is to make the center-to-center hole distance in the extension rod match that of the upper leg link.

Preparing for the assembly



Completed step



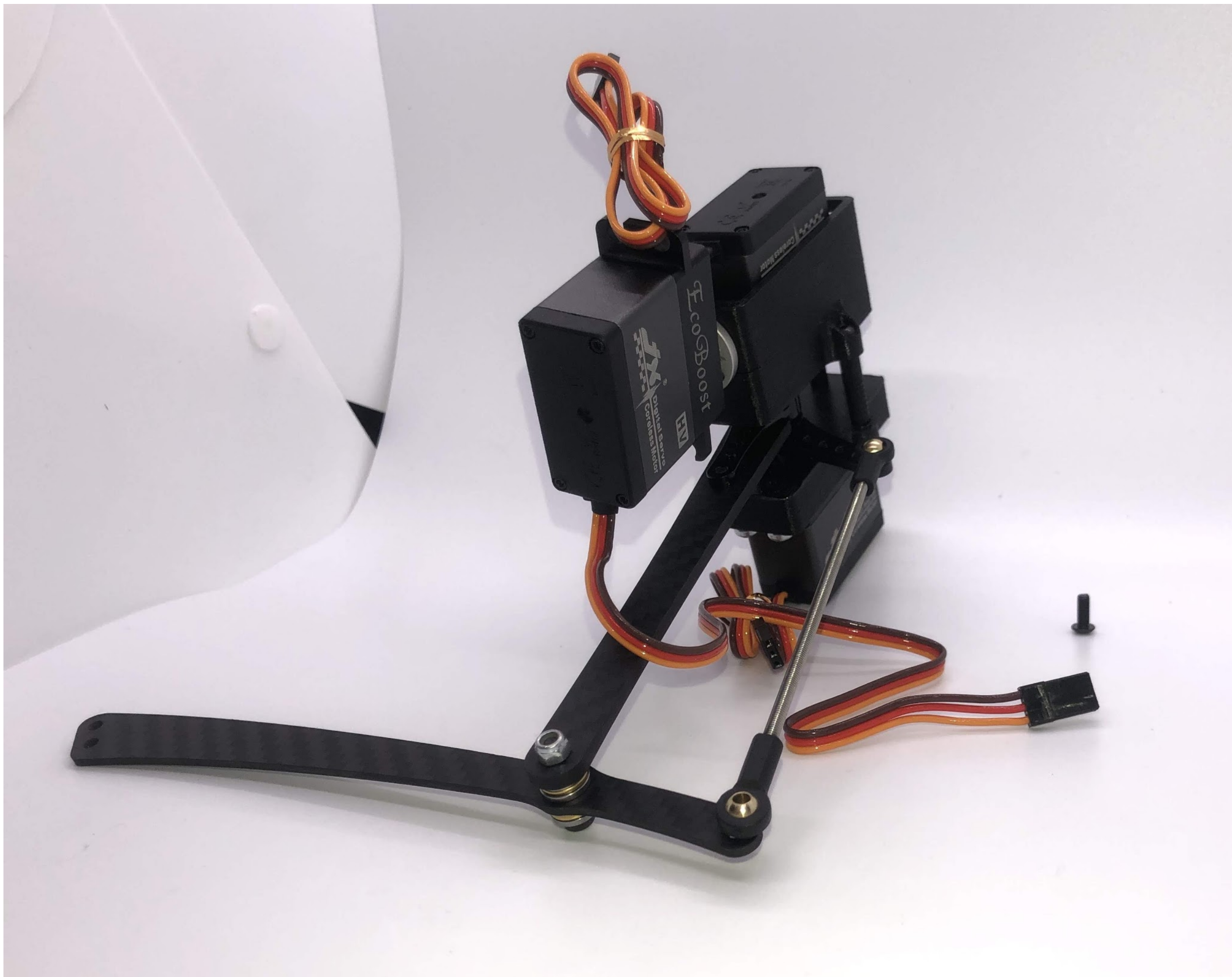
Step 14. Attach Upper Leg Extension Rod to Servo Horn

- Video Instructions: https://youtu.be/c0DC35XpYTk
- Materials: M3x8 button head screw
- Tools: 2mm driver

Instructions:

1. From the inside, screw the extension rod to the servo horn with the M3x8 button head screw.

Preparing for the assembly



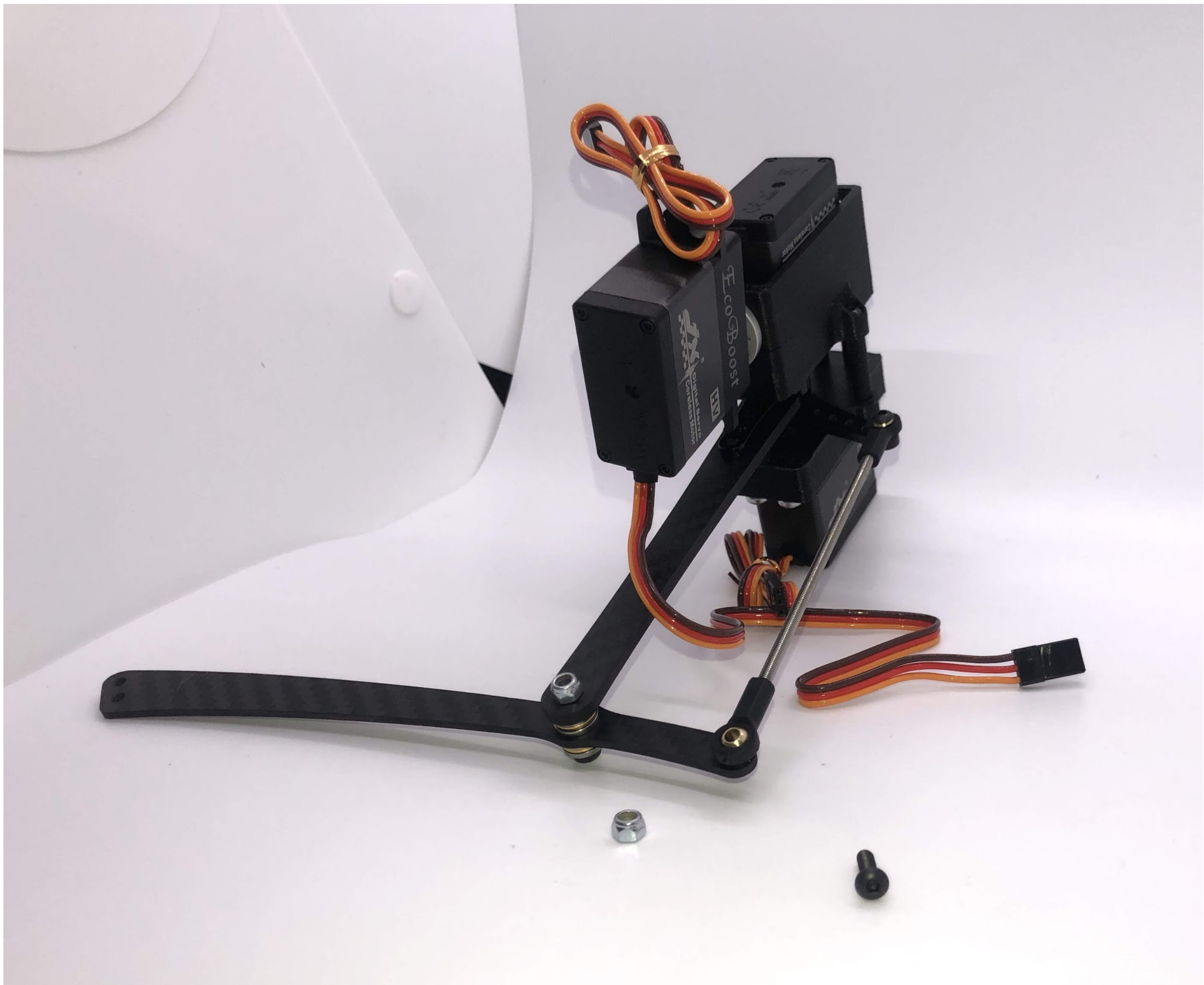
Step 15. Attach Upper Leg Extension Rod to Lower Leg Carbon Linkage

- Video Instructions: https://youtu.be/uQt9EFQzu2w
- Materials: M3x10 button head screw, M3 Locking Nut
- Tools: 2mm driver, wrench

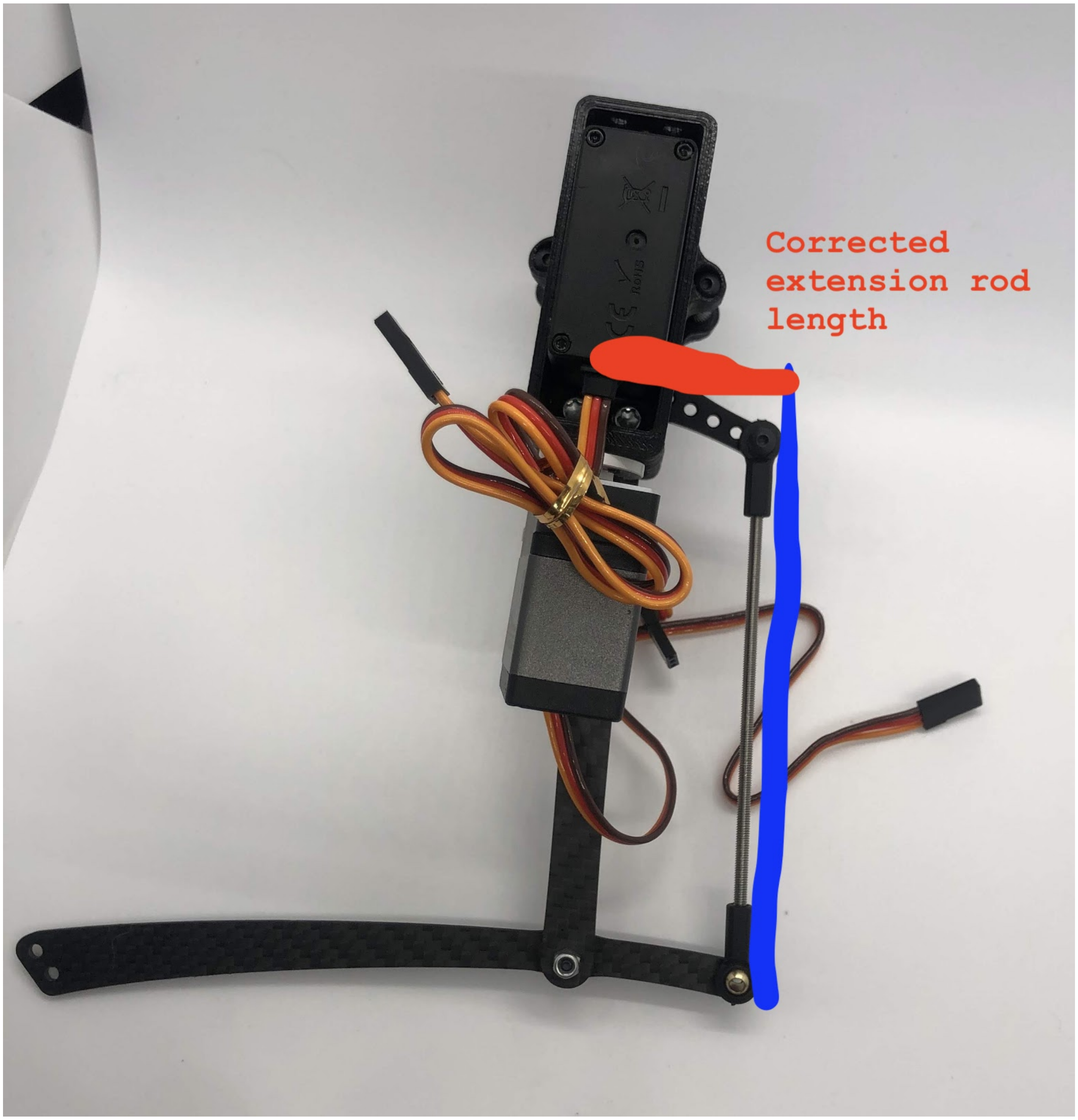
Instructions:

1. Slide a M3x10 button head screw through the carbon fiber piece and then the rod end. Then fasten the screw with a M3 locknut, using a wrench to keep it in place while you use an allen key to tighten.

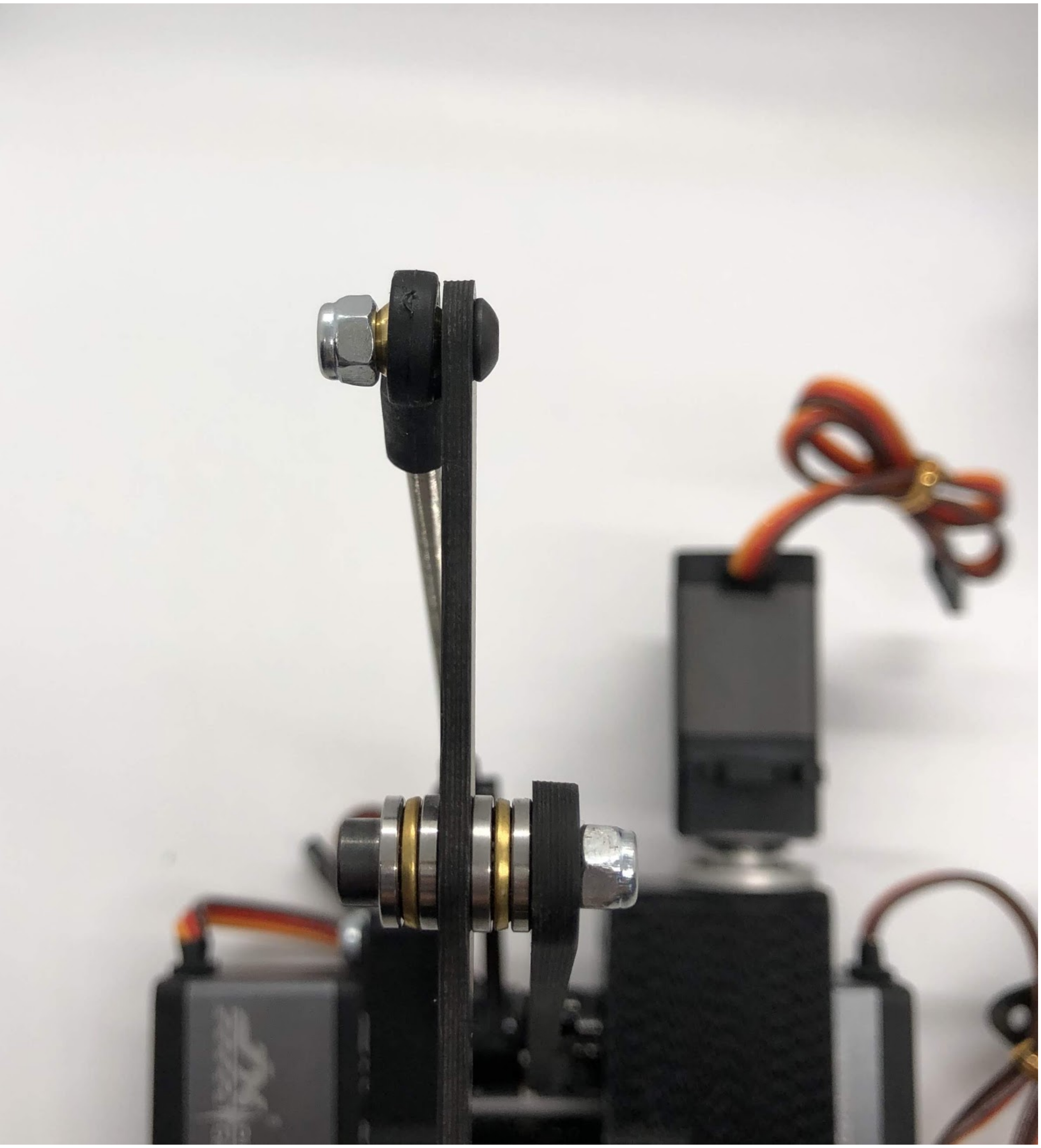
Preparing for the assembly



Completed step. CORRECTION: The extension rod in this picture is actually too short. See the red and blue annotation for the correct assembly. The servo horn, extension rod, upper leg link, and upper part of the lower leg link should form a perfect parallelogram.

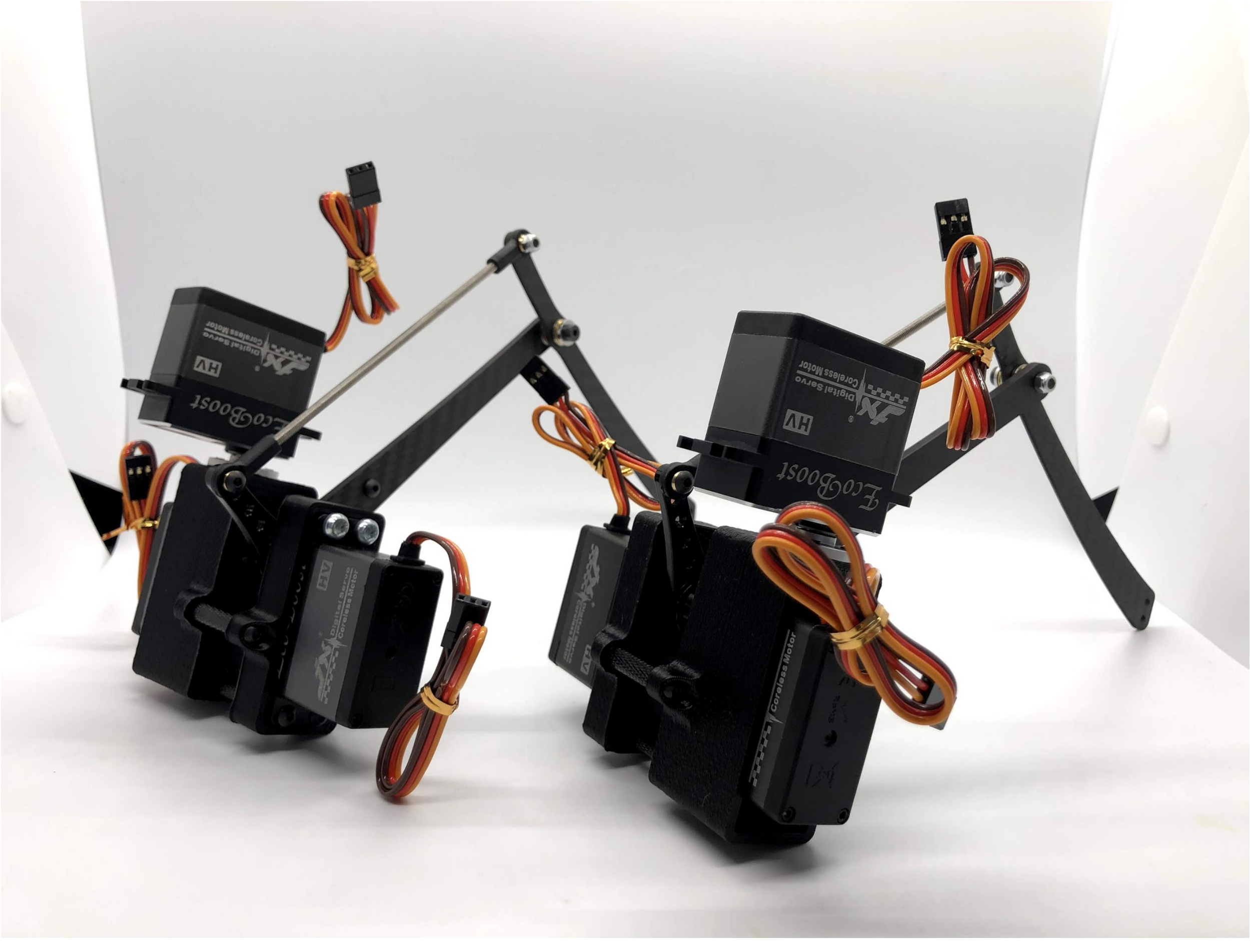


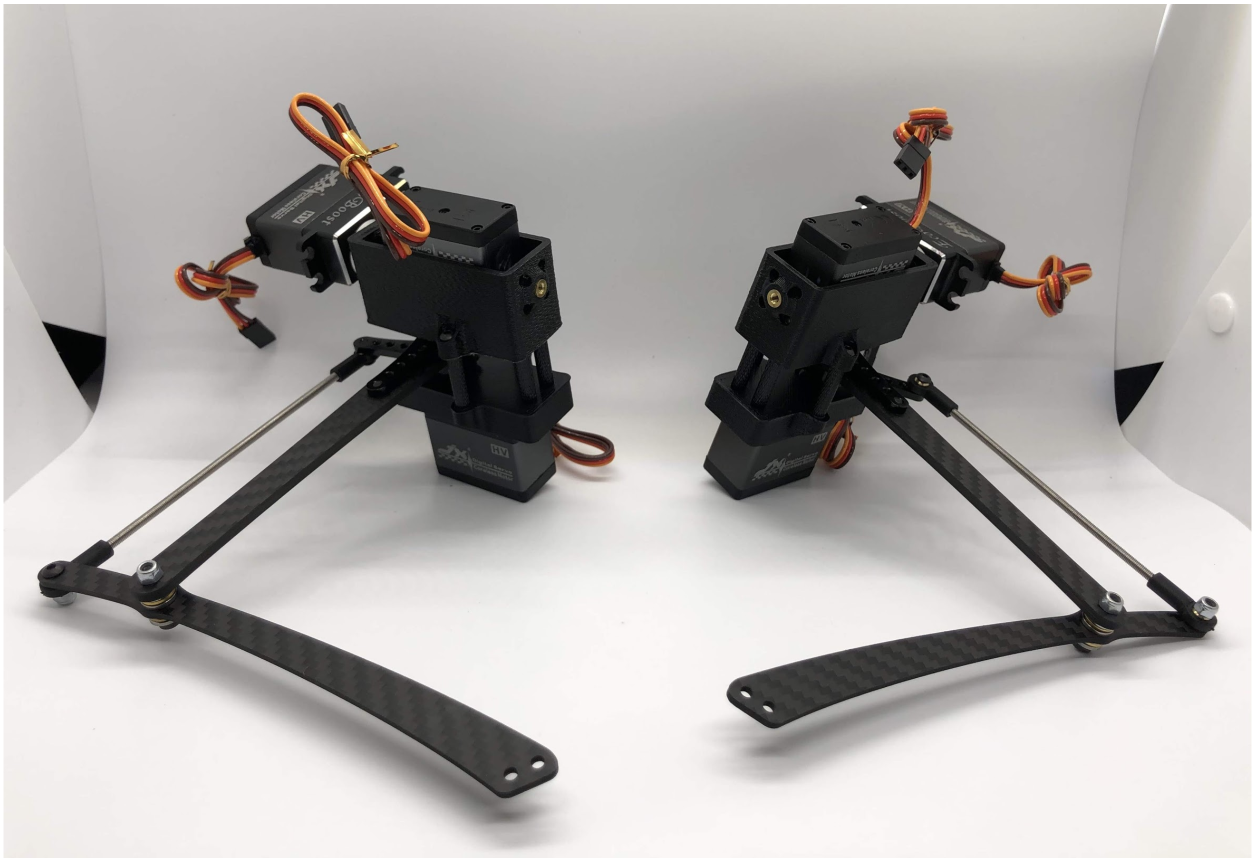
Another look at the assembly



Done!

Left and Right side


EcoBoost
HV
Digital Servo
Coreless Motor



Continue on to Body Assembly

1.2.2 Body Assembly

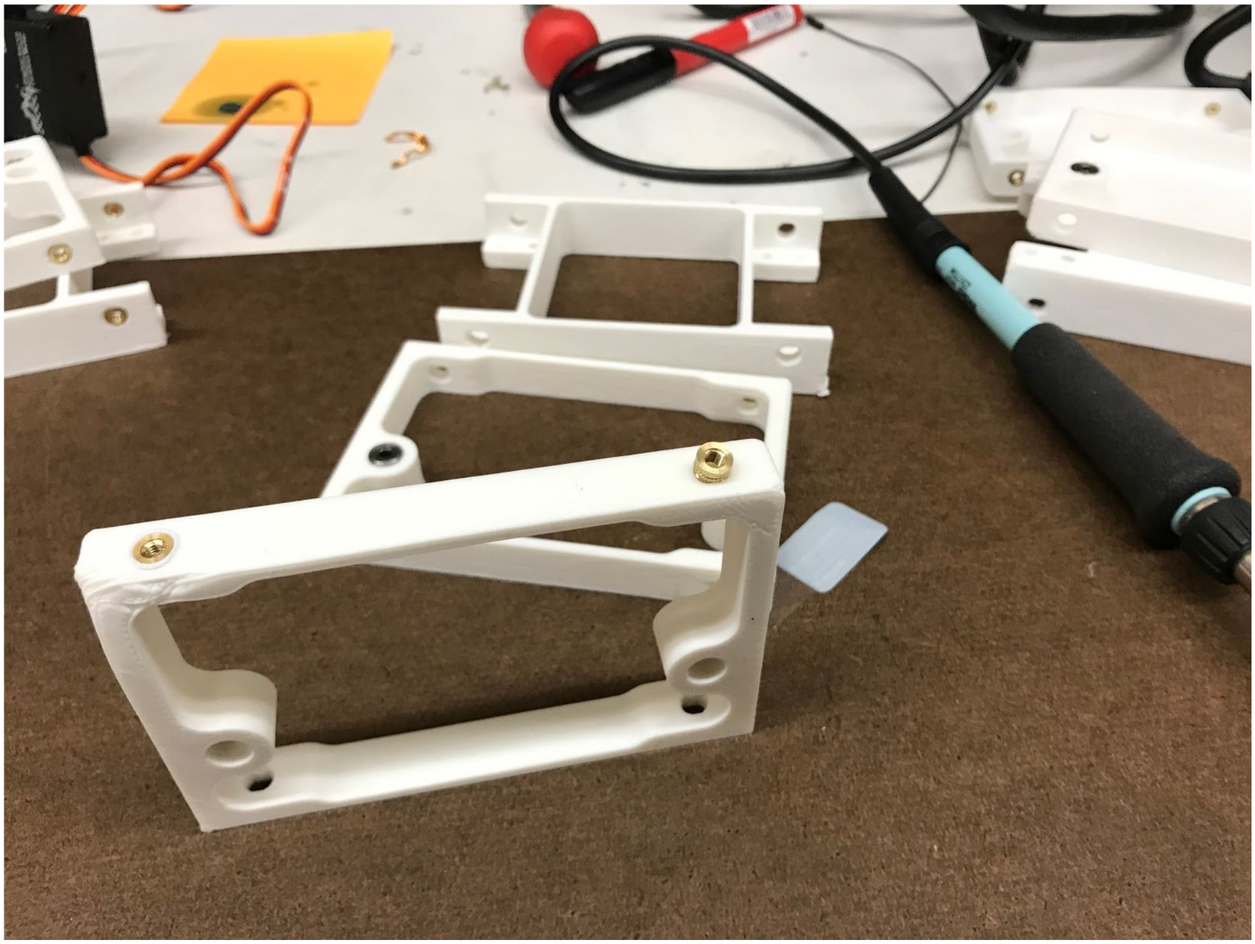
Step 1. Install tapered threaded heat inserts into 3D printed parts

- No Video Available
- Materials: M3 tapered heat-set inserts for plastic x16, 4 body pieces
- Tools: Soldering iron set to around 500f / 260c

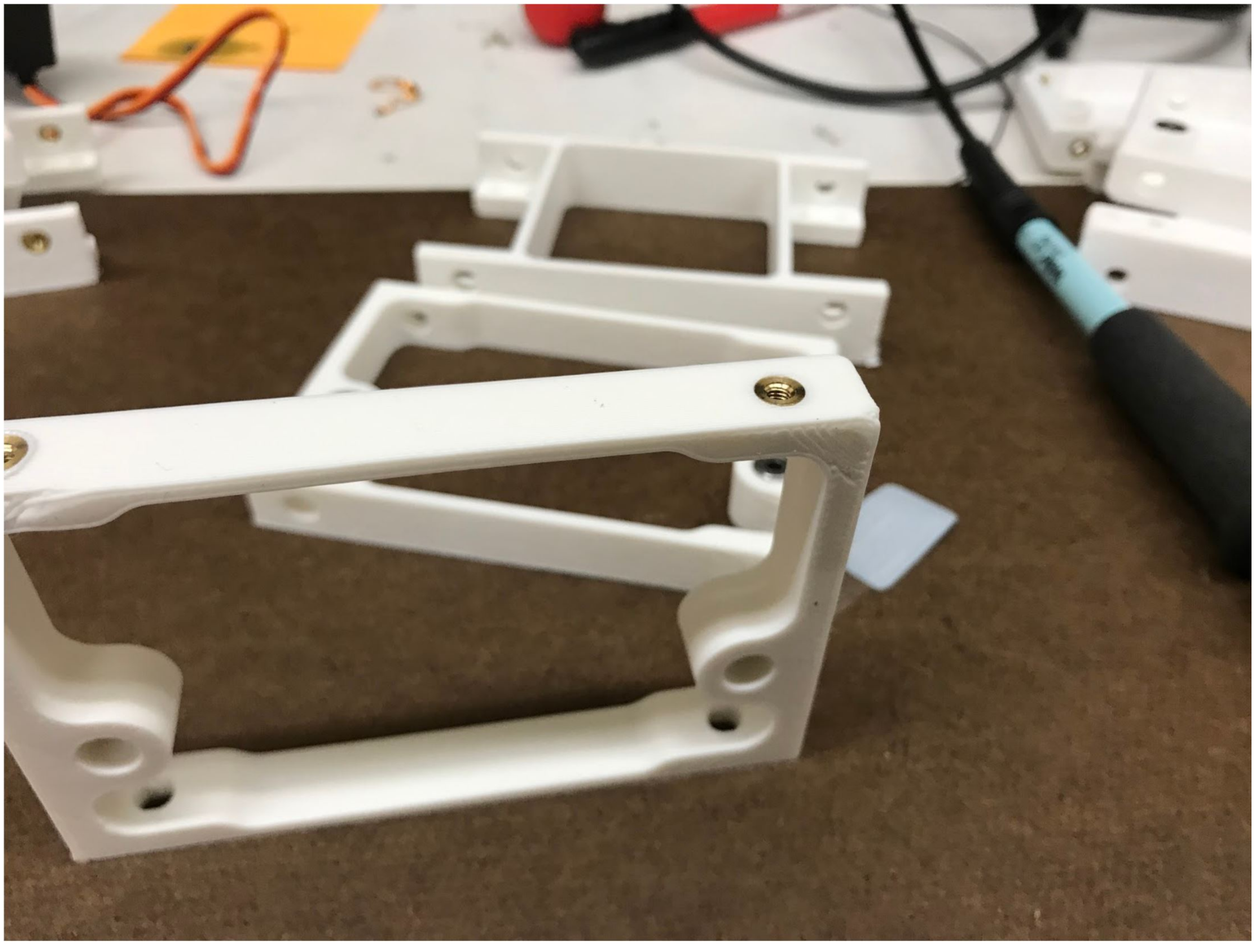
Instructions:

1. Each of the 3D printed body pieces have four holes — two on top and two on bottom that hold the tapered heat-set inserts for plastic
2. Place the insert into the hole with the tapered side down
3. Use a soldering iron set to around 500f or 260c to gently press the insert into the plastic. I recommend just using the weight of the iron to press the insert in, and I also suggest doing it in steps, ie pressing it in 1mm, taking the iron out, then pressing another 1mm etc into it is all the way in. This method prevents the soldering iron from getting stuck to the iron.

Before pressing the tapered threaded heat insert



After pressing the tapered threaded heat insert



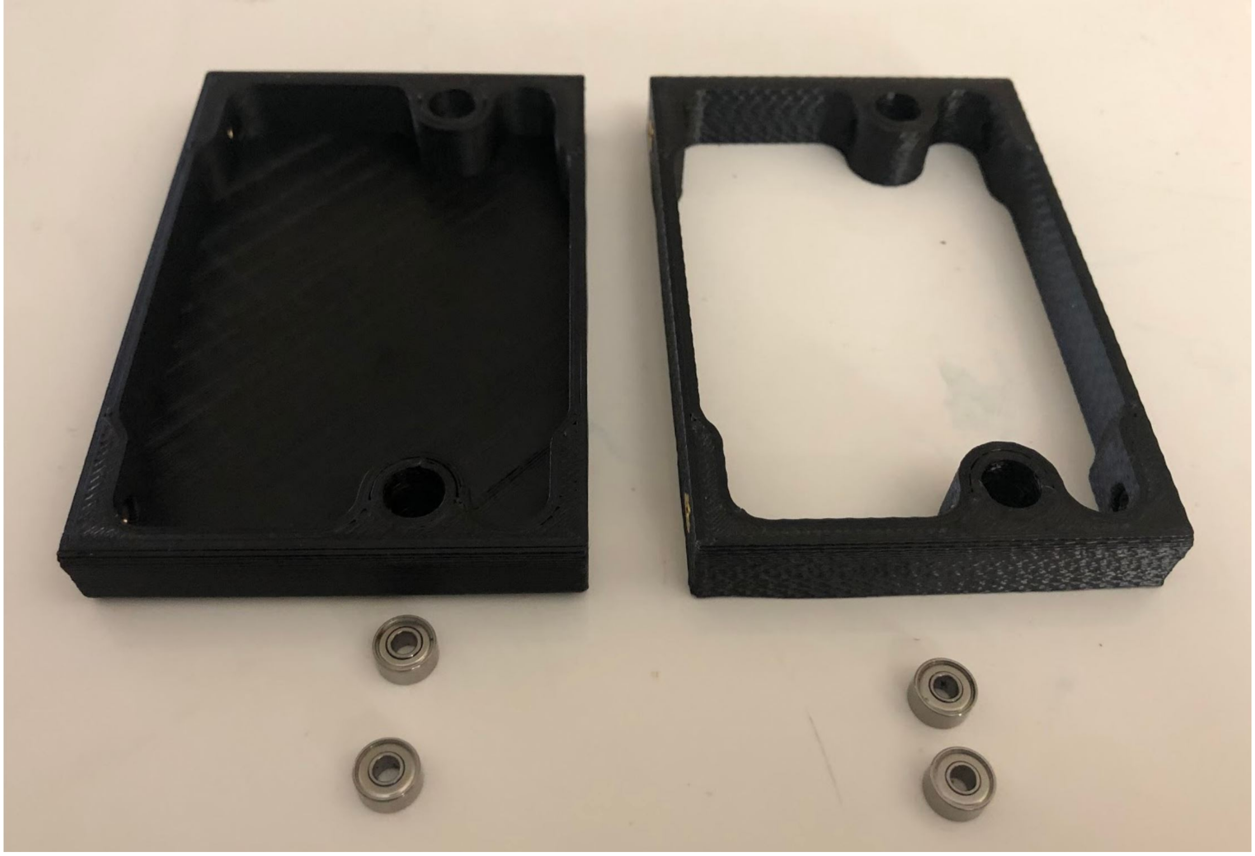
Step 2: Press the radial bearings into the body pieces

- No Video Available
- Materials: 4 bearings (3mm x 8mm x 4mm Bearing MR693-zz), Front Front body part, Back Front body part
- Tools: Your hands, arbor press, or vice

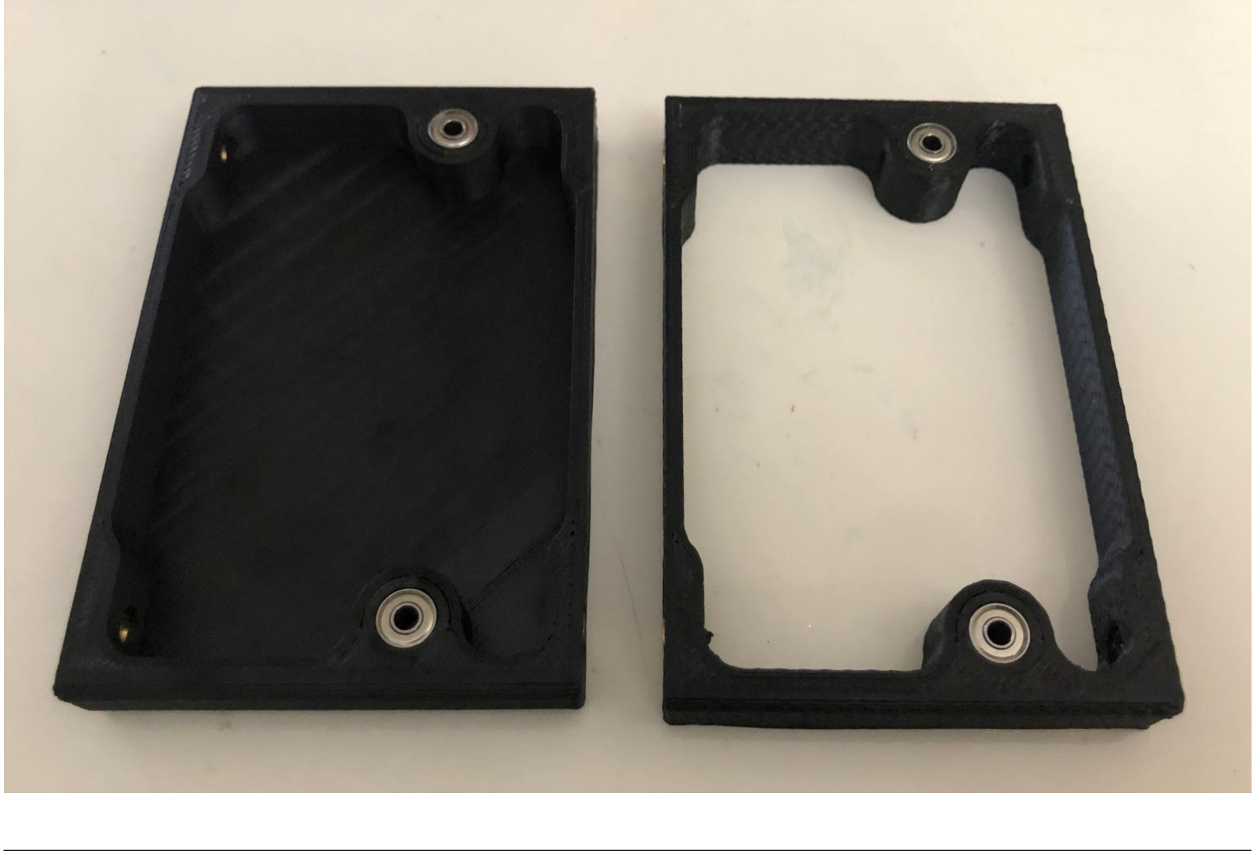
Instructions:

1. Press two bearings into the two holes in the frontmost piece (called Front Front), and two bearings into the two holes in the back piece (called Back Front).

Preparing for the assembly



Completed Step



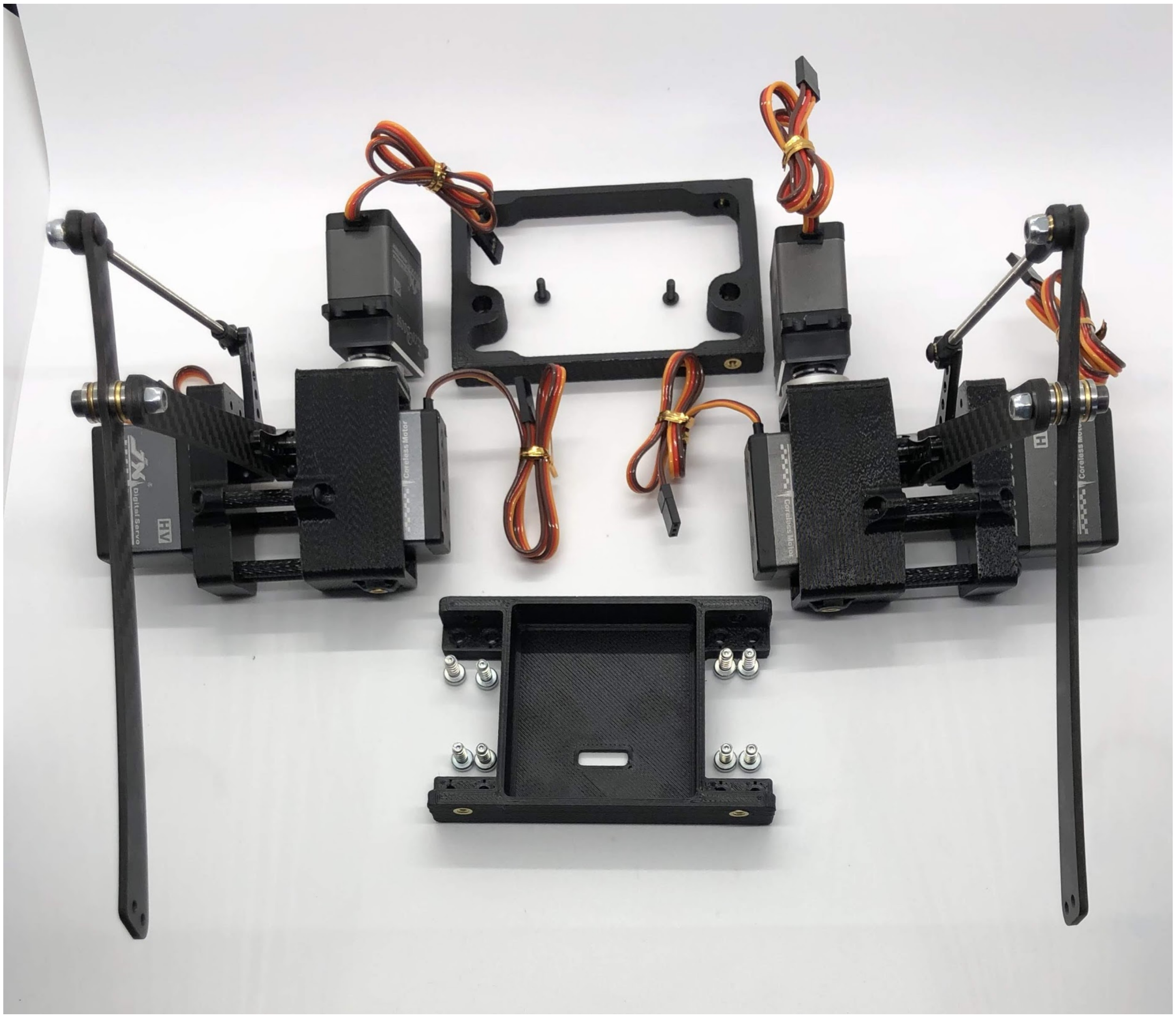
Step 3. Fasten the hip assemblies

- Video Instructions: https://youtu.be/Av9e2HzpbBo
- Materials: 16x M4x8 screws (plastic), 4x M3x8 button head screw, four hip assemblies, four body parts
- Tools: Torx T20 + 2mm driver

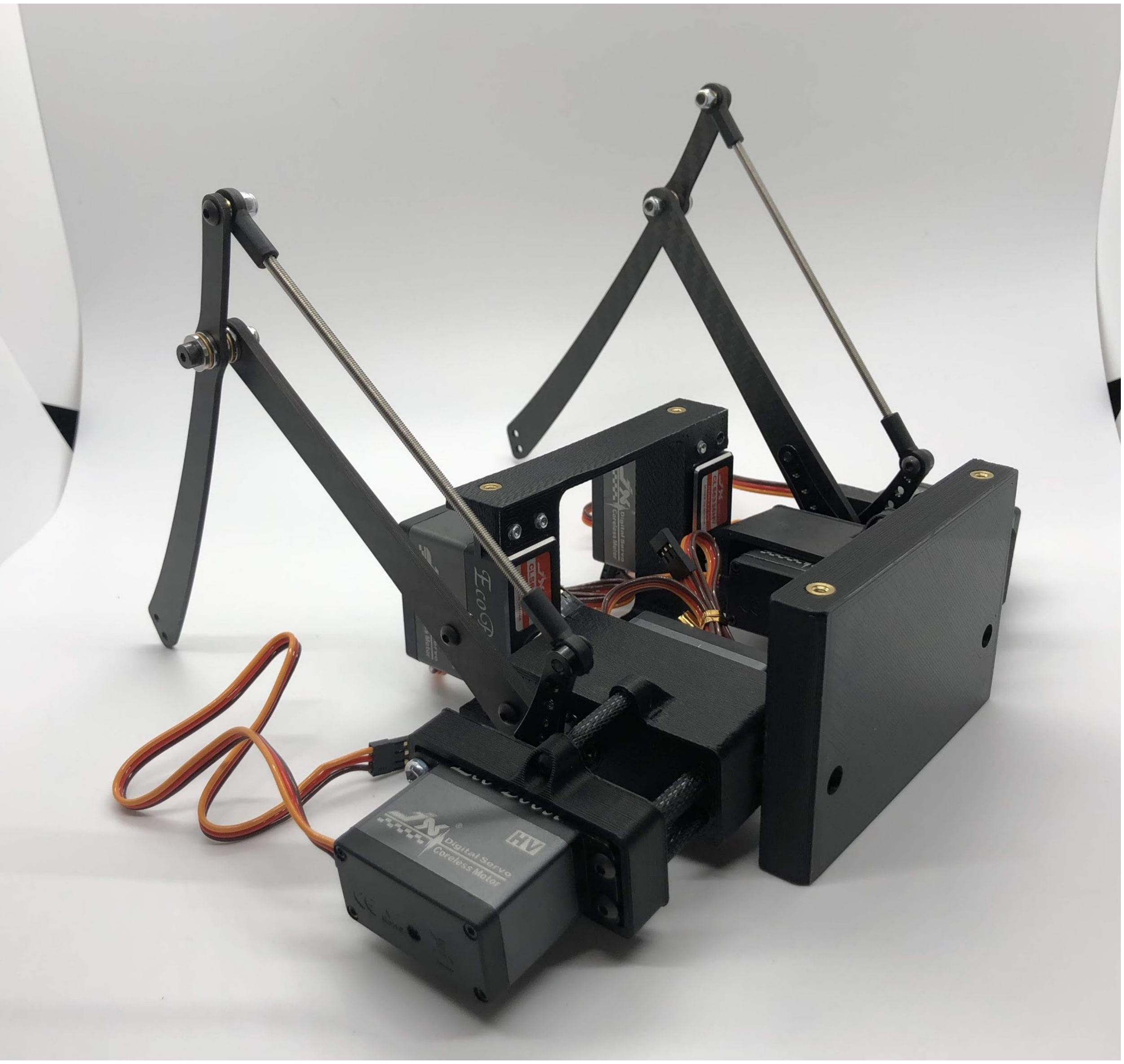
Instructions:

1. Use the M4x8 screws for plastic to fasten two hip assemblies to the Back Back body part and another two hip assemblies to the Front Back body part
2. Then screw the M3x8 button head screws through the bearings you pressed into the Front Front and Back Front parts and thread them into the threaded inserts in the hip assembly

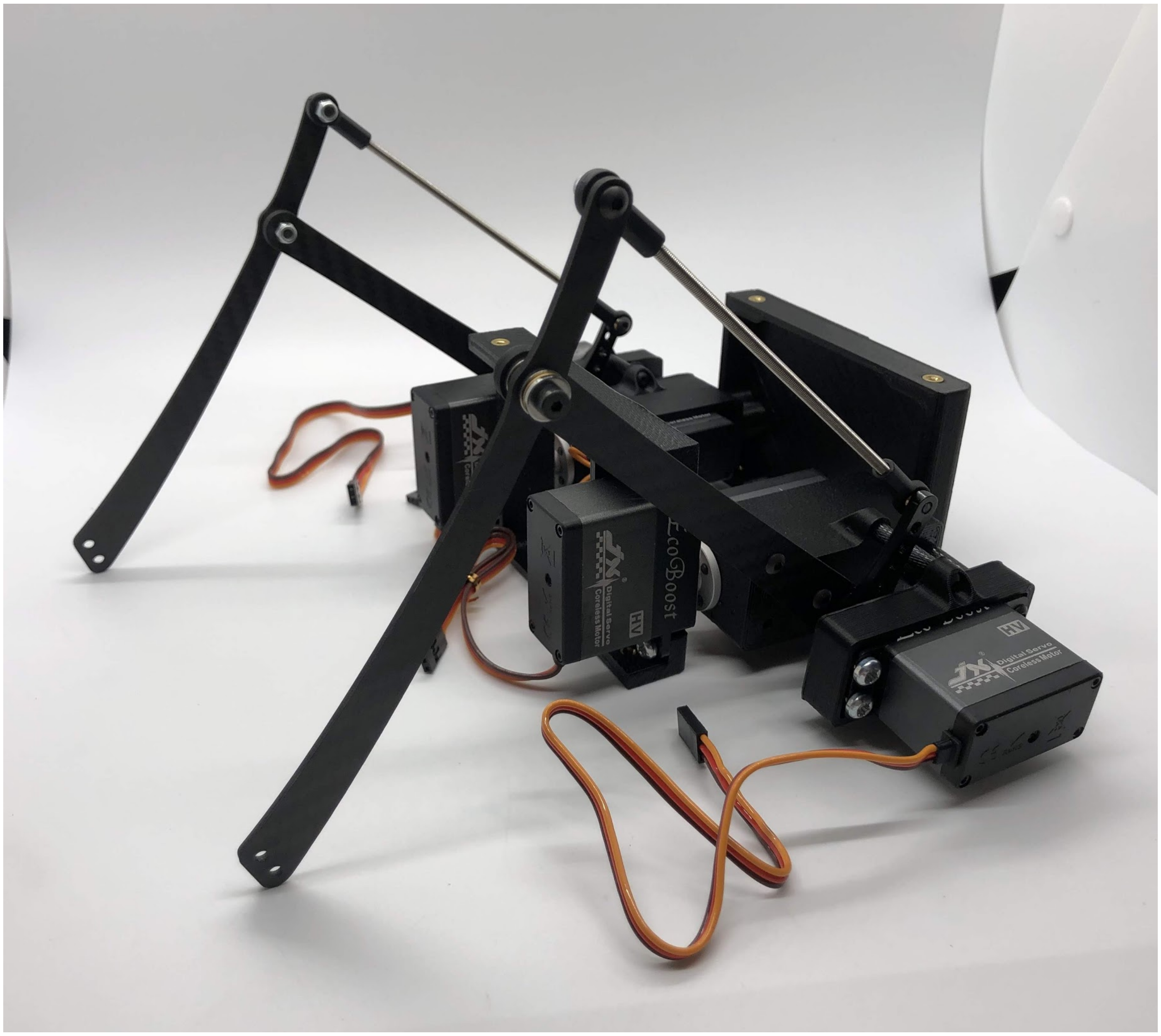
Preparing for the assembly



Completed Step



Another look at the assembly



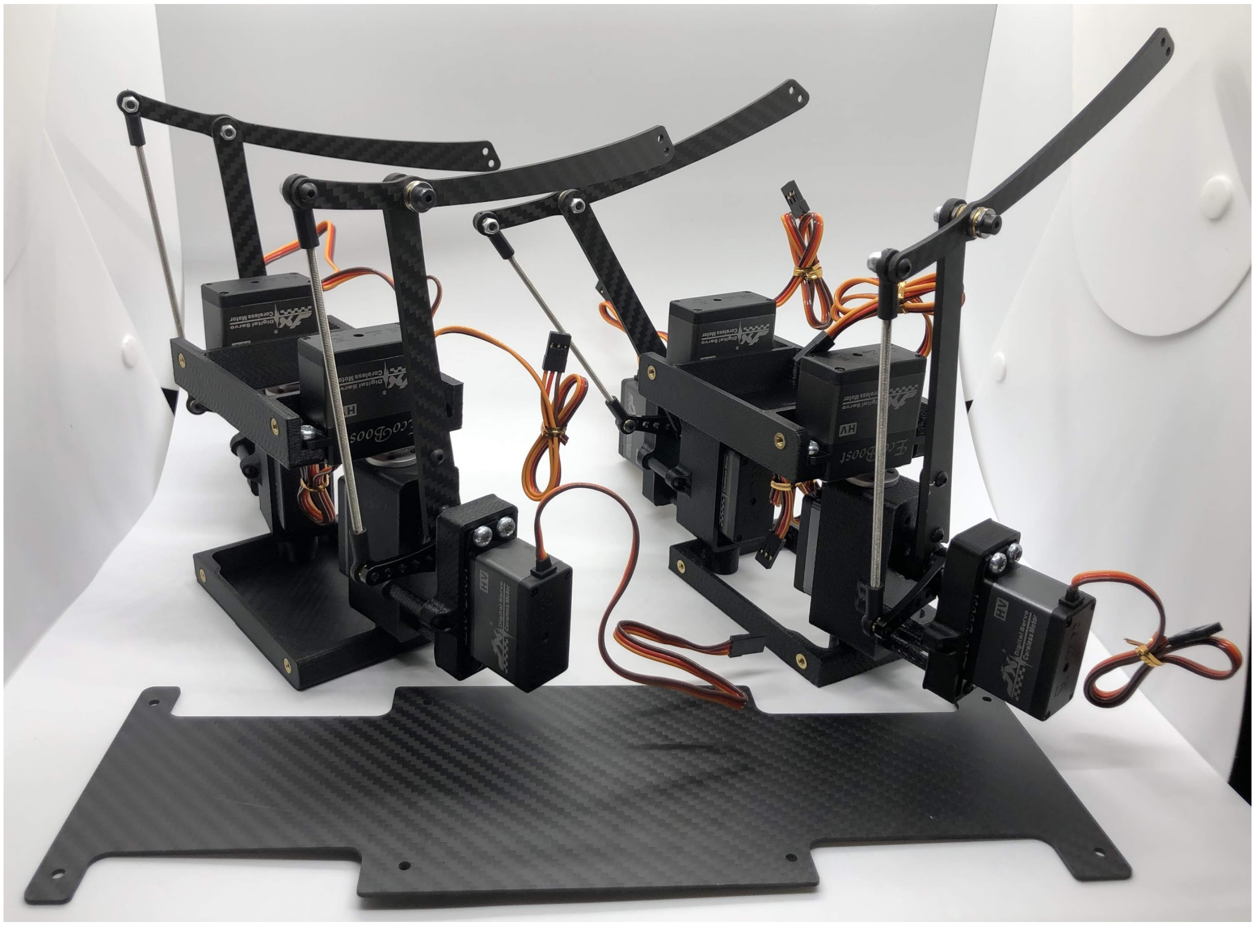
Step 4. Attach the two leg/body assemblies to the bottom carbon fiber plate

- Video Instructions: https://youtu.be/f4iDKkfCkIs
- Materials: 16x M3x6 button head screws, 2 leg/body assemblies, Botton carbon fiber plate
- Tools: 2mm hex driver

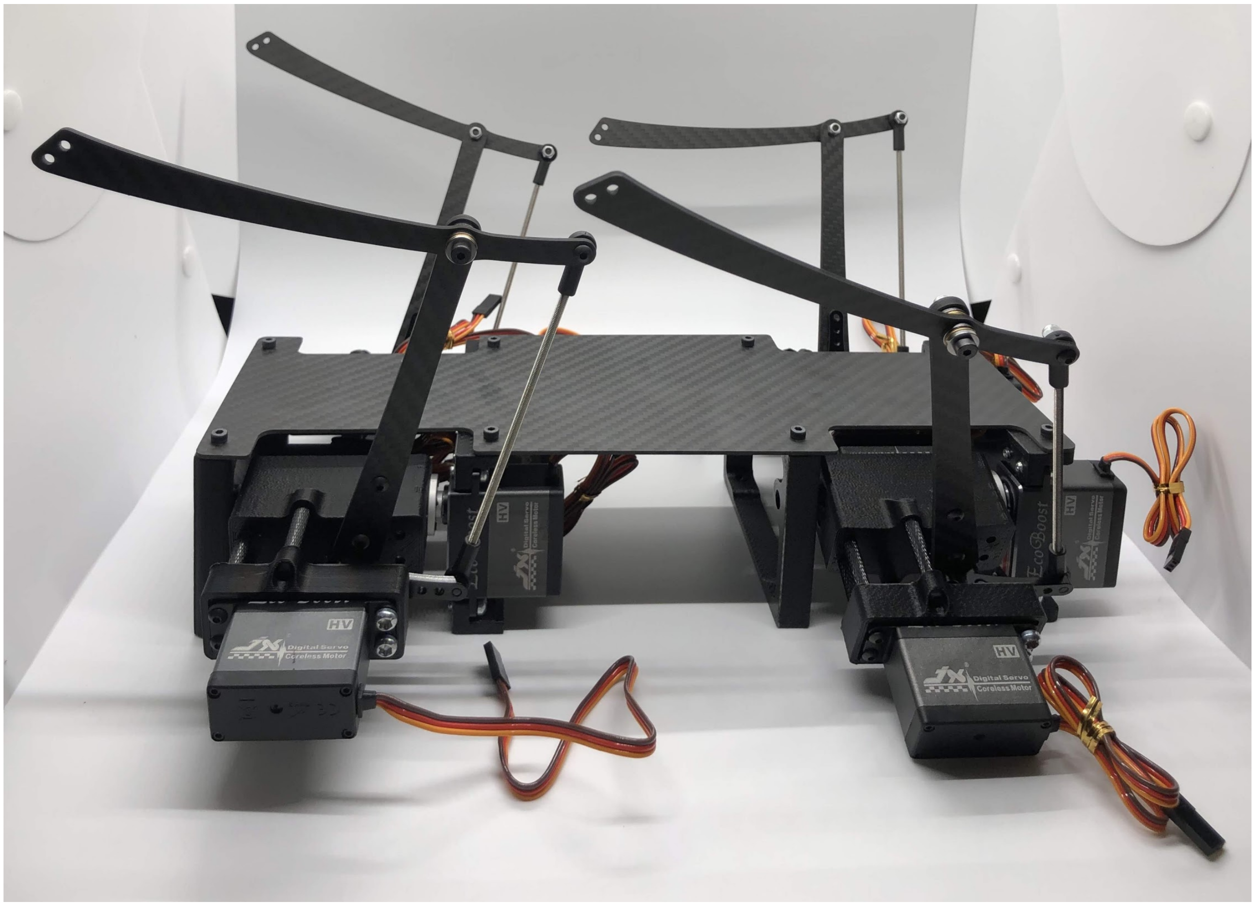
Instructions:

1. Use the M3x6 button head screws to fasten the two leg/body assemblies you built to the bottom carbon fiber plate.

Preparing for the assembly



Completed step



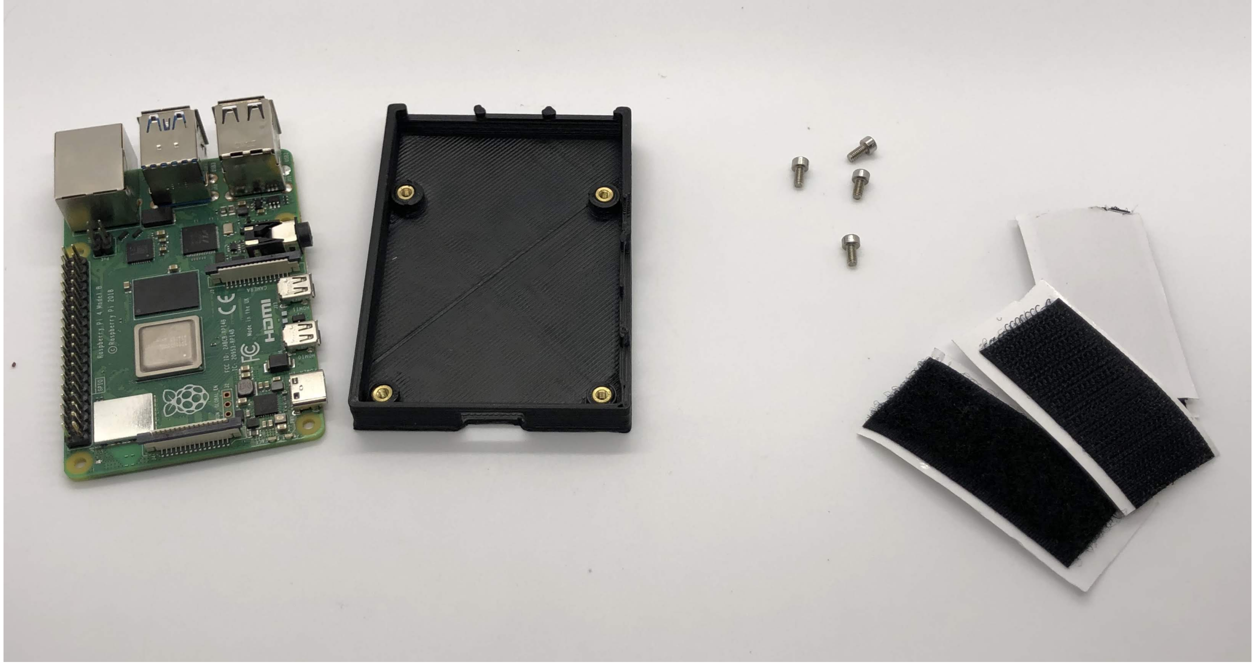
Step 5. Prepare and mount the Raspberry Pi case

- Video Instructions: https://youtu.be/ZlbkTc2Jxu8
- Materials: Raspberry Pi case (picase.stl), 4x M2.5 tapered heat-set inserts, 4x M2.5x6 socket head screws, Dual Lock
- Tools: Soldering iron, 2mm driver

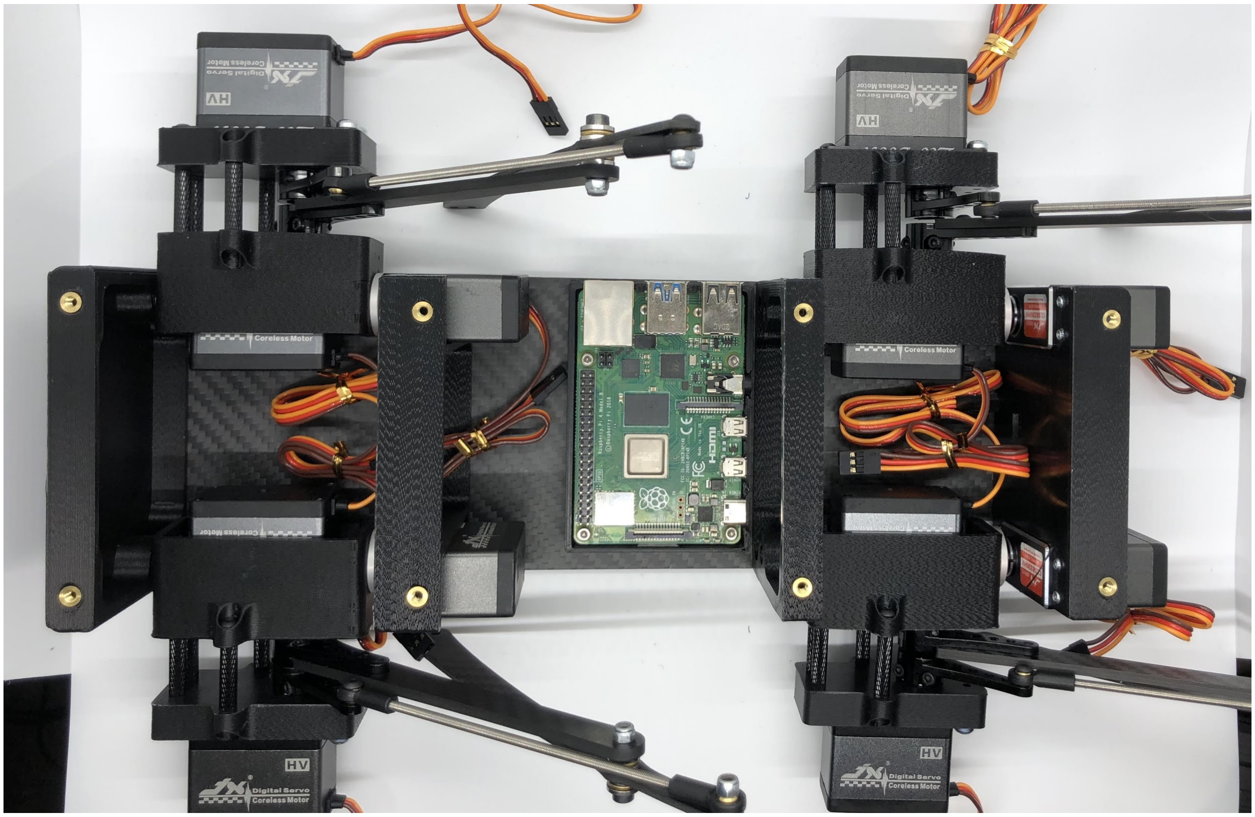
Instructions:

1. In the same way you installed the previous inserts, press the M2.5 inserts into the holes in the raspberry pi case. Then, use the M2.5x6 socket head screws to screw the raspberry pi to the case
2. Finally, add Dual-Lock to the case to mount it to the bottom carbon fiber plate

Preparing for the assembly



Completed Step



Step 6. Assemble the PCB (if not done so already)

Navigate to PCB Assembly Instructions

Step 7. Plug in servo motors to Raspberry Pi

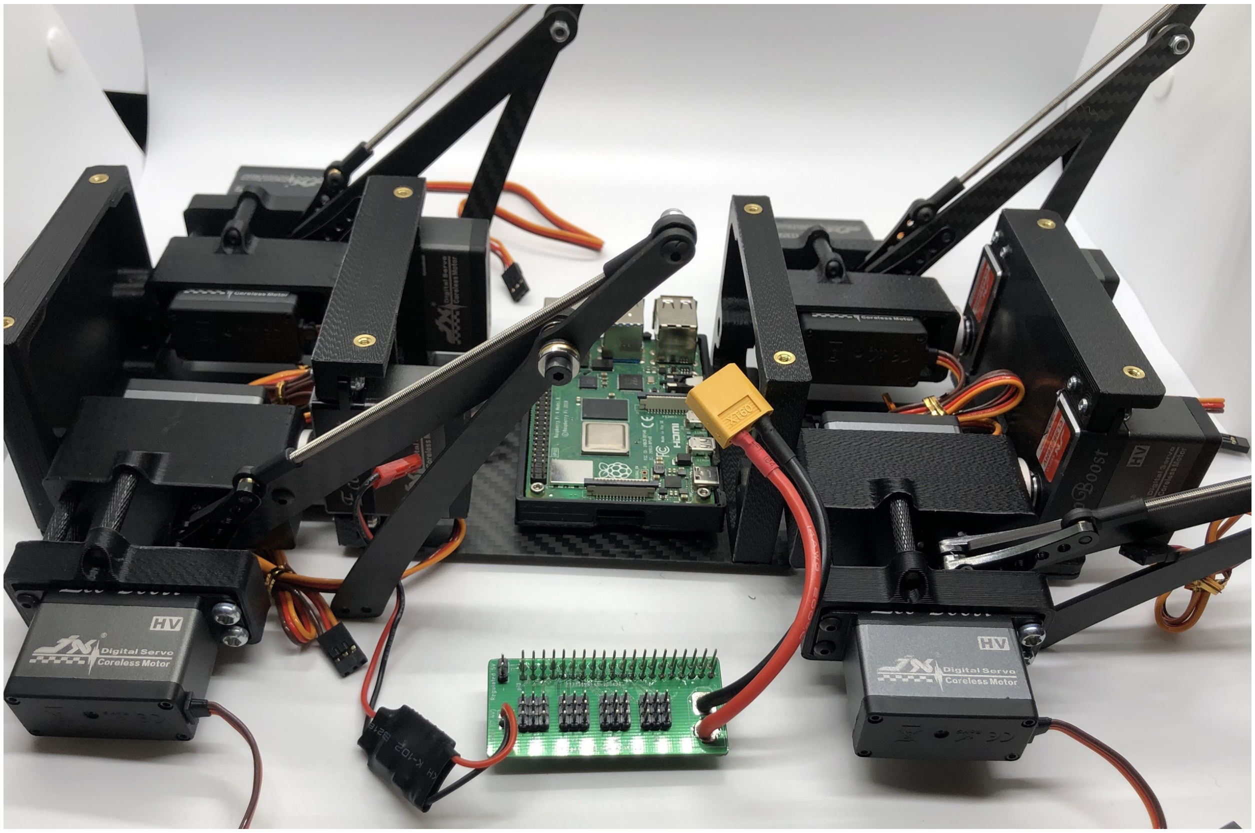
- Video Instructions: https://youtu.be/ToJtlmDO4AY
- Materials: Four hip assemblies mounted to the bottom plate, mounted Raspberry Pi with servo power distribution hat
- Tools: None

Instructions:

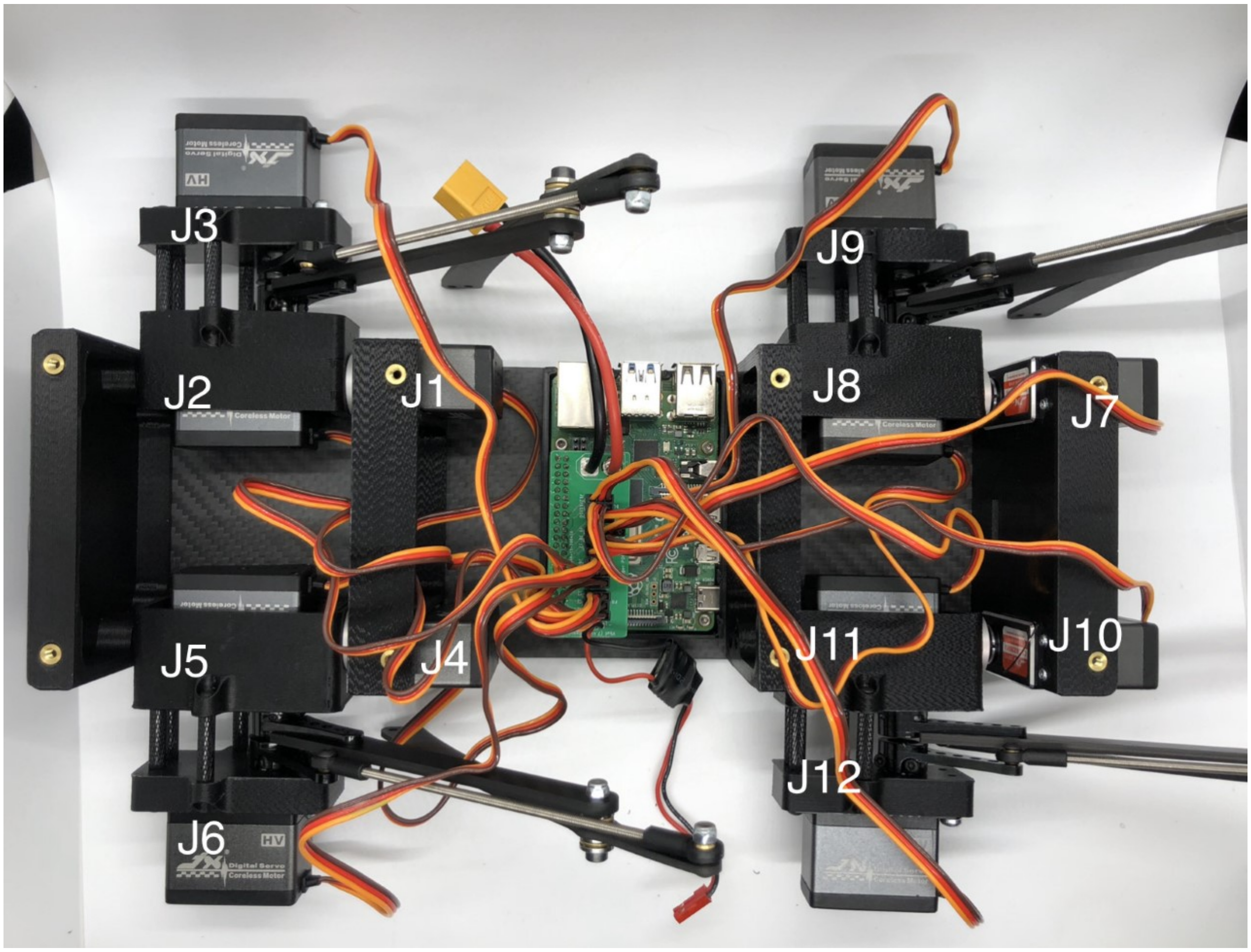
1. Connect PCB to Rasberry Pi
2. Plug the servo cables into the custom circuit board in this pattern shown below.

J1 through J12 correspond to one of the twelve sets of header pins soldered to the circuit board. The circuit board has indicators for how to align the signal, ground, and positive wires from the servo motors into the board, but in case they're too hard to see, you can know that the signal pins on the servo connectors always face towards the Raspberry Pi header.

Preparing for the assembly



Completed step



Done!

Complete PCB assembly if you haven't done so already.

1.2.3 PCB Assembly

Step 1: Solder the servo connector headers to the board

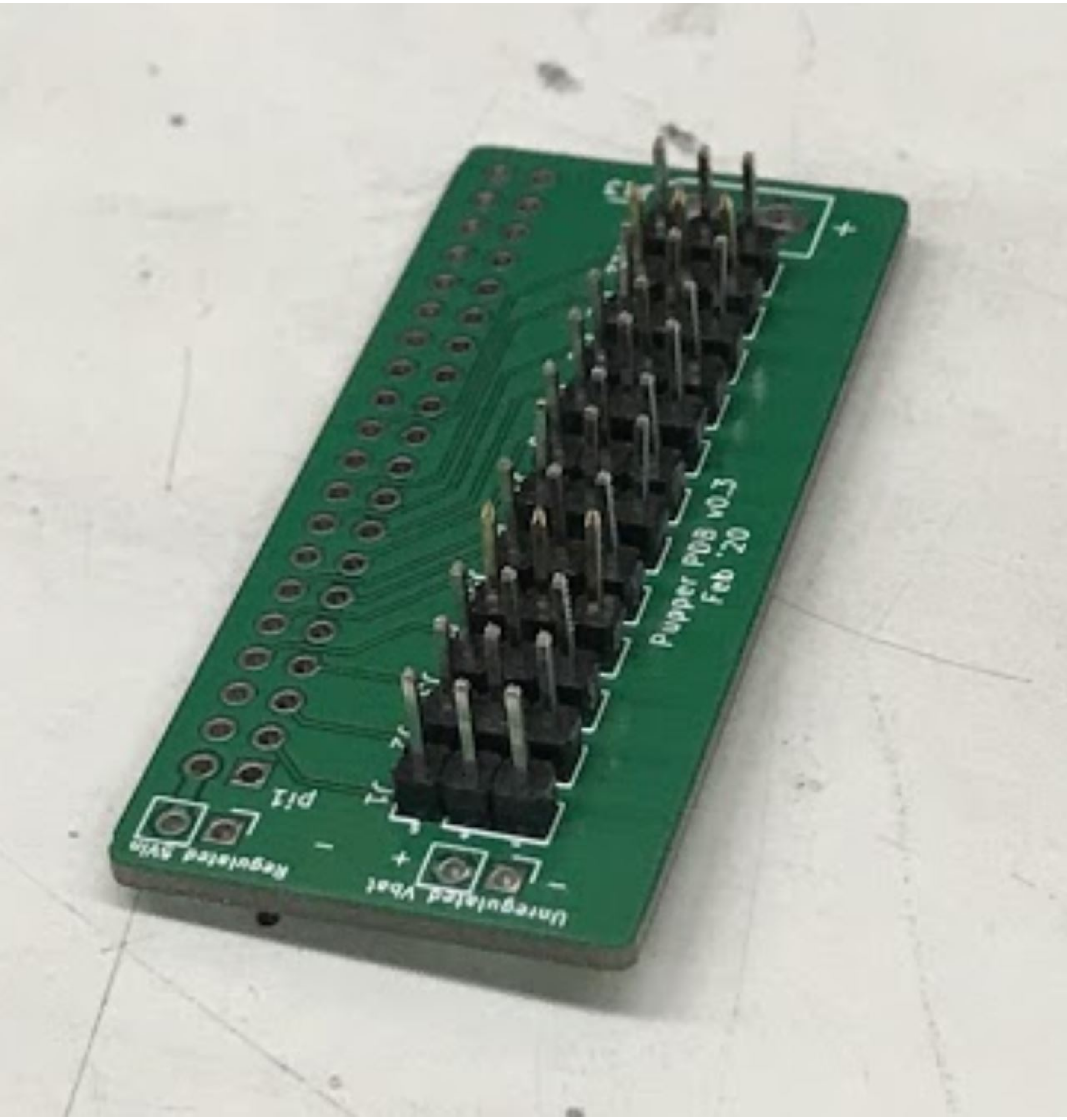
- No Video Available
- Materials: PCB, 12 male headers of 3 pins each
- Tools: Soldering iron, preferably a nice one with >=60W heat output.

Instructions:

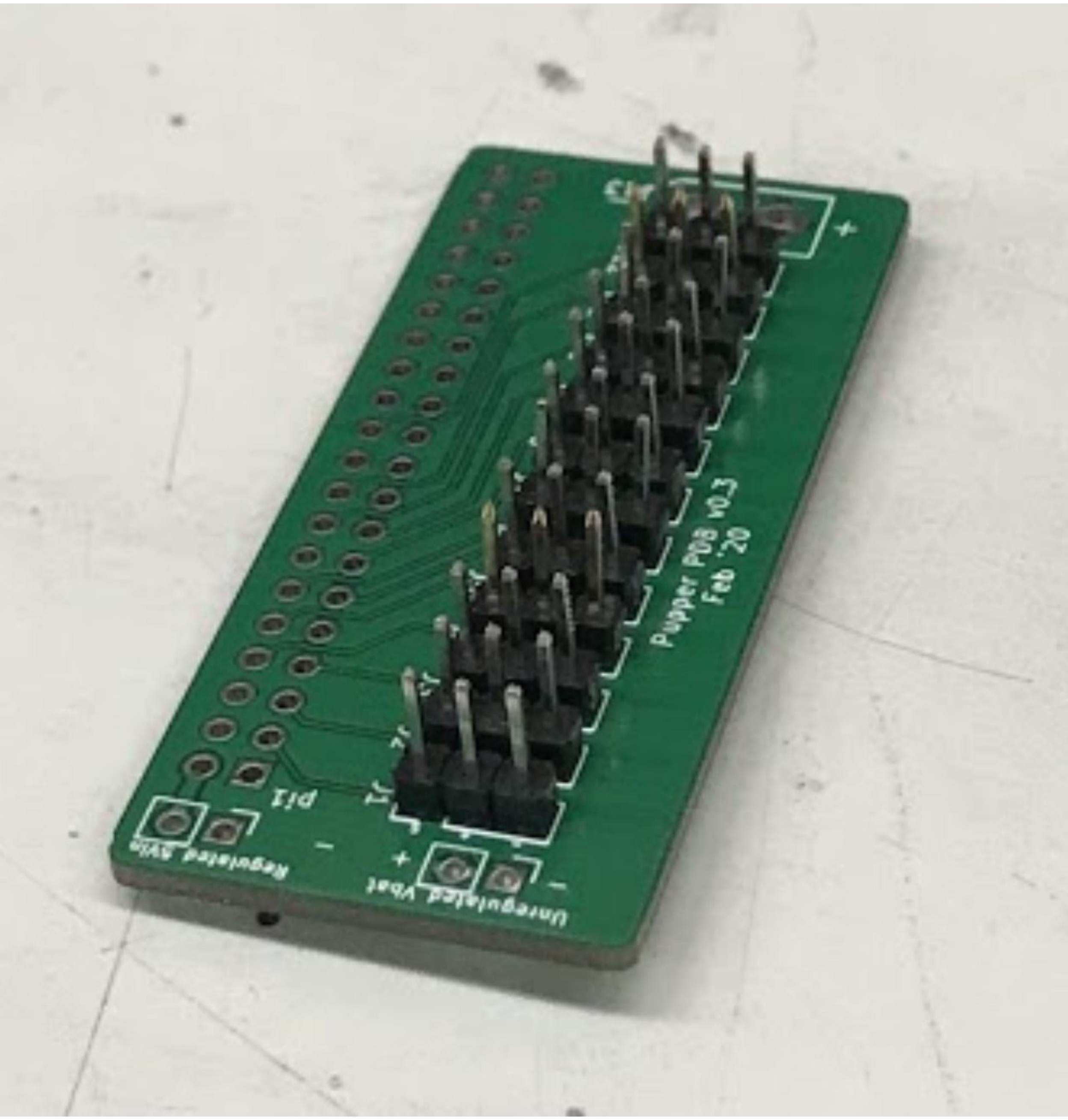
1. Place each of the 12 male header pins into their respective slots as shown in the photo.
2. Then, turn the board upside down so you have access to solder the underside. Be careful that the headers don't all fall out when you turn the board over. When I did this, I pressed a hard foam block up against the top side of the pins to make sure they didn't tilt or fall out when I turned the board over. You'll also want to check that the pins are mostly perpendicular to the board after you turn the board over.

3. Once the board is turned over, solder all of the signal pins to keep the headers in place. The signal pins are the pins closest to the Raspberry Pi header pin holes (the 2x20 array).
4. Once the headers are all tacked into place, solder the remaining ground and positive pins.

Placed all the pins into the board unsoldered



Completed Step, soldered pins



Step 2: Solder the Raspberry Pi header pin

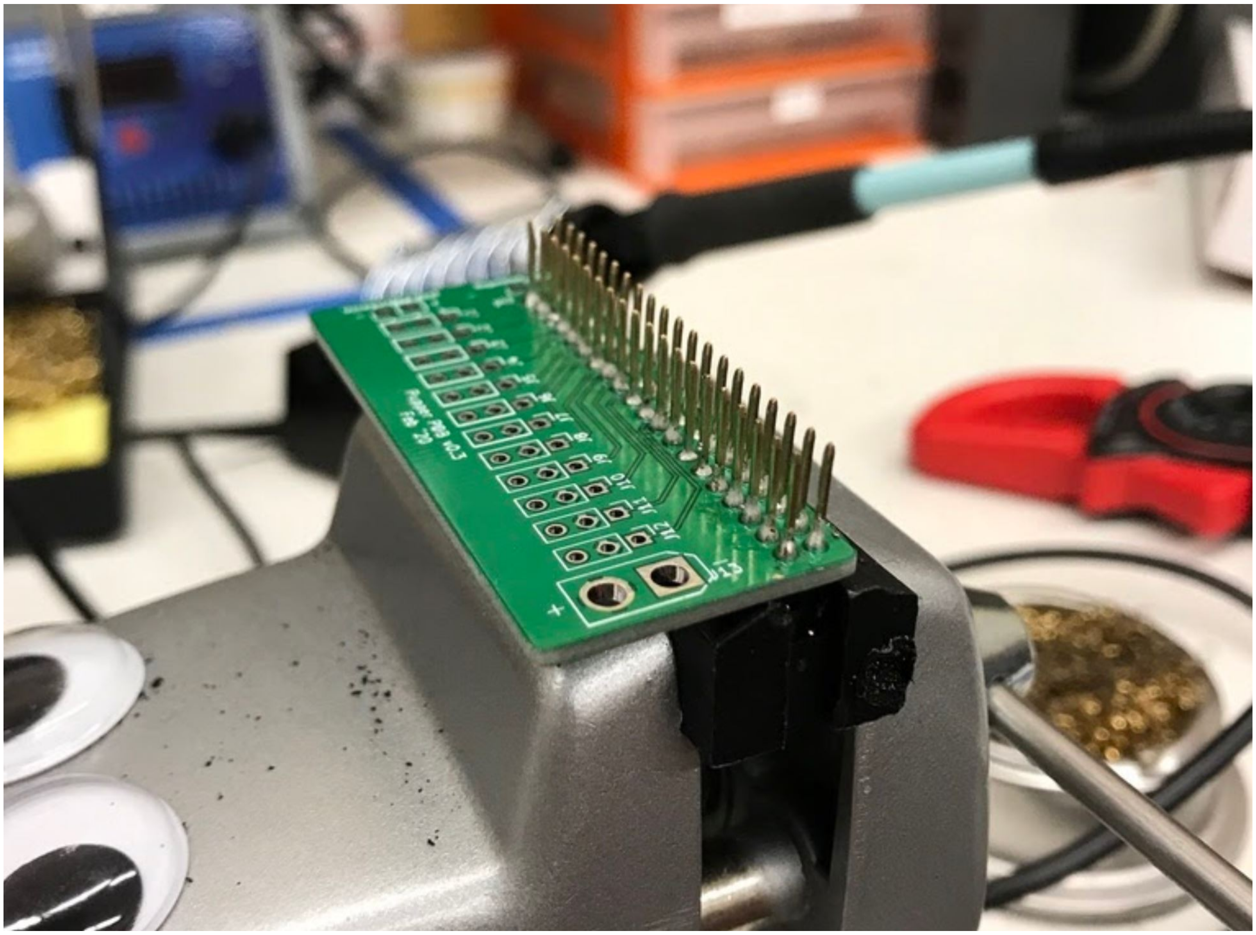
- No Video Available
- Materials: PCB, 2x20 raspberry pi header pin
- Tools: Soldering iron

Instructions:

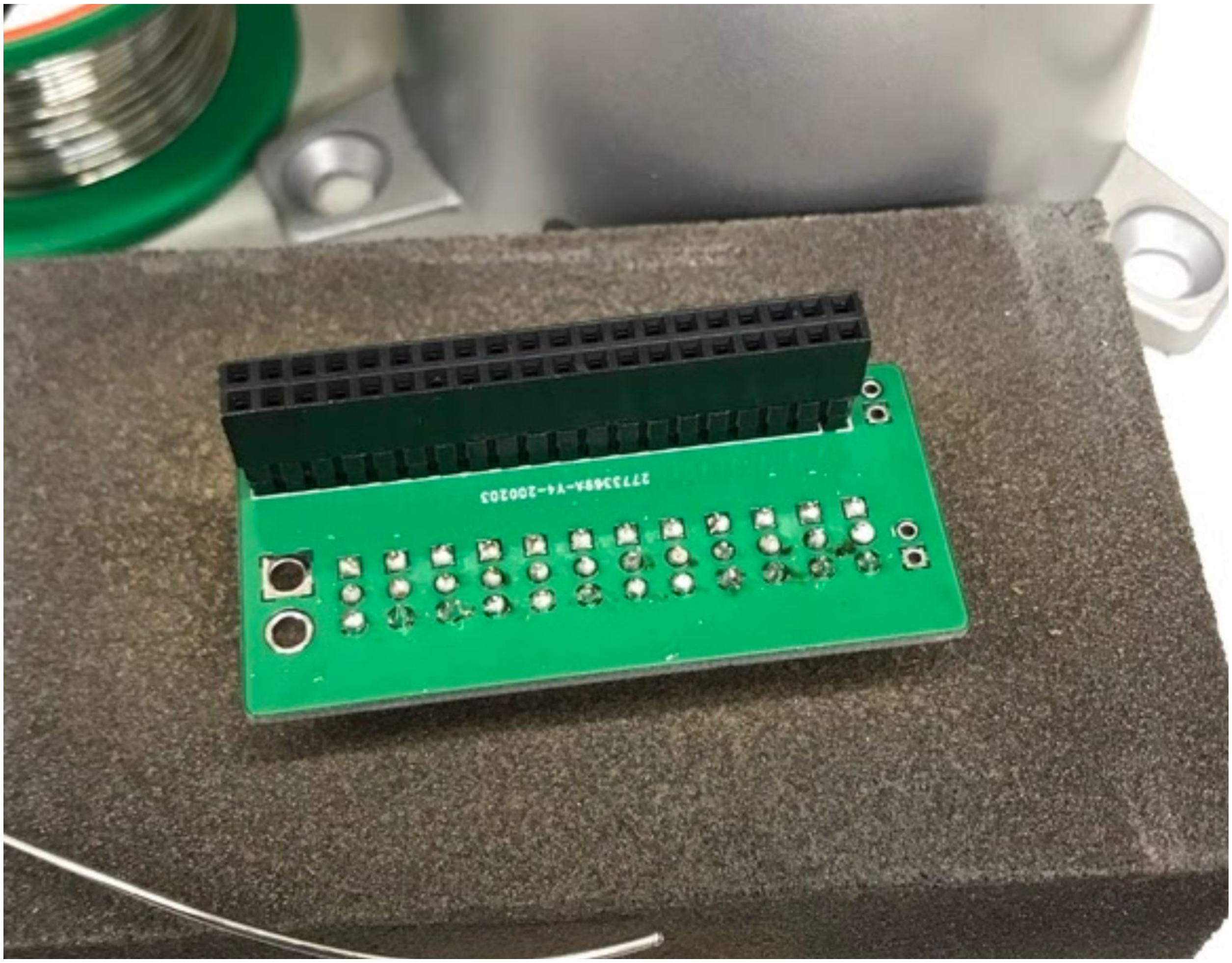
1. Insert the 2x20 header pin into the PCB. Make sure that you insert the header from the bottom so that the pins are coming out the top. This will allow the header to sit on top of the Raspberry Pi.
2. Secure the PCB and header pin in a vice

3. Solder the header pins in from the top.

After soldering the 2x20 header pin onto the PCB.



Underside of board after soldering the 2x20 header pin.



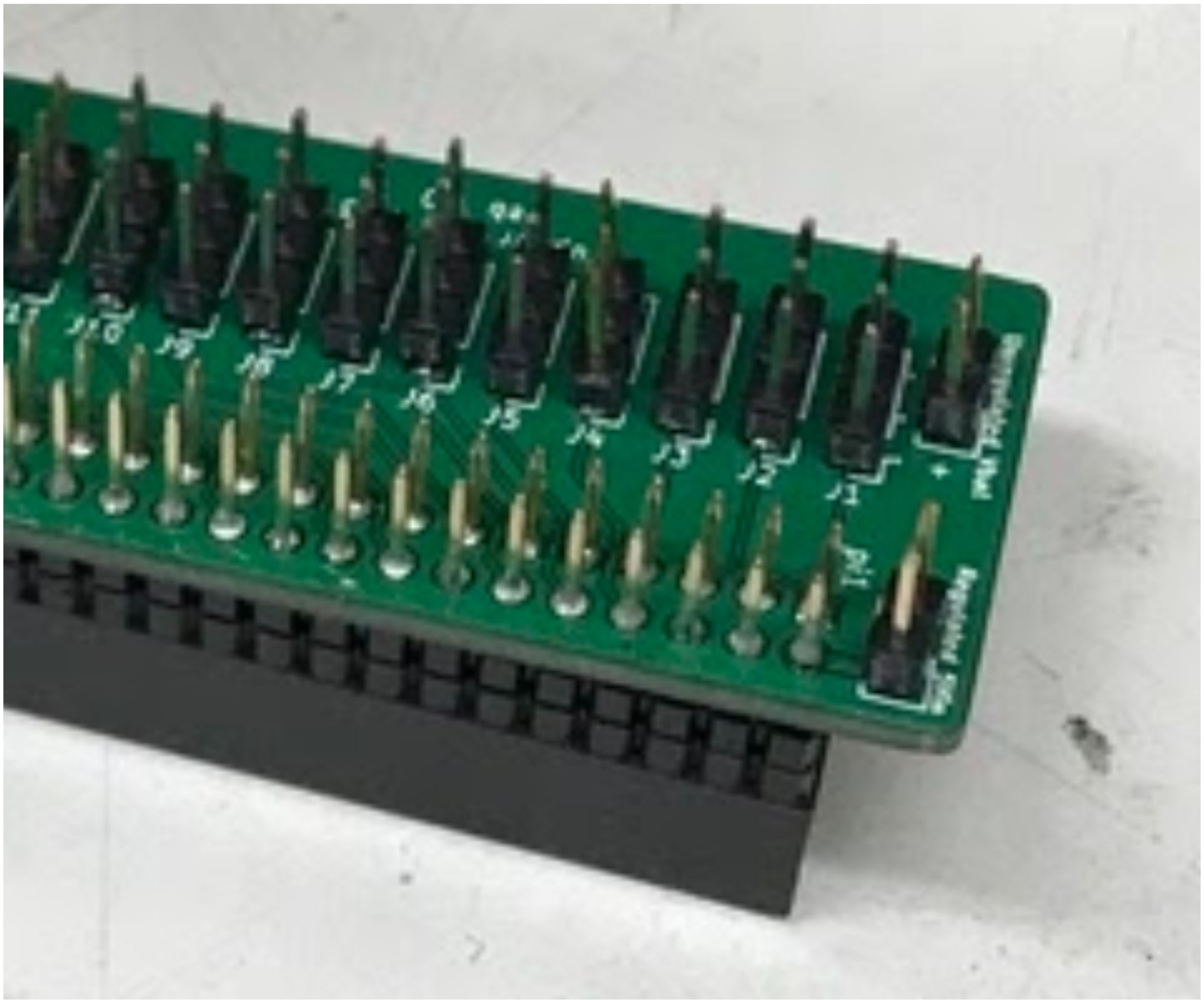
Step 3: Solder the bec and 5V in pins

- No Video Available
- Materials: PCB, header pins
- Tools: Soldering iron, vice

Instructions:

1. Snap off a pair of 1x2 header pins and solder them to the areas labelled Vbat and Regulated 5V. Important: If you do not have a dupont/jst crimps and crimper on hand, then do not solder pins to the Vbat holes.

BEC and 5V pins (four pins on the right) soldered to the PCB.



Step 4: Solder the XT 60 pigtail connector to the PCB

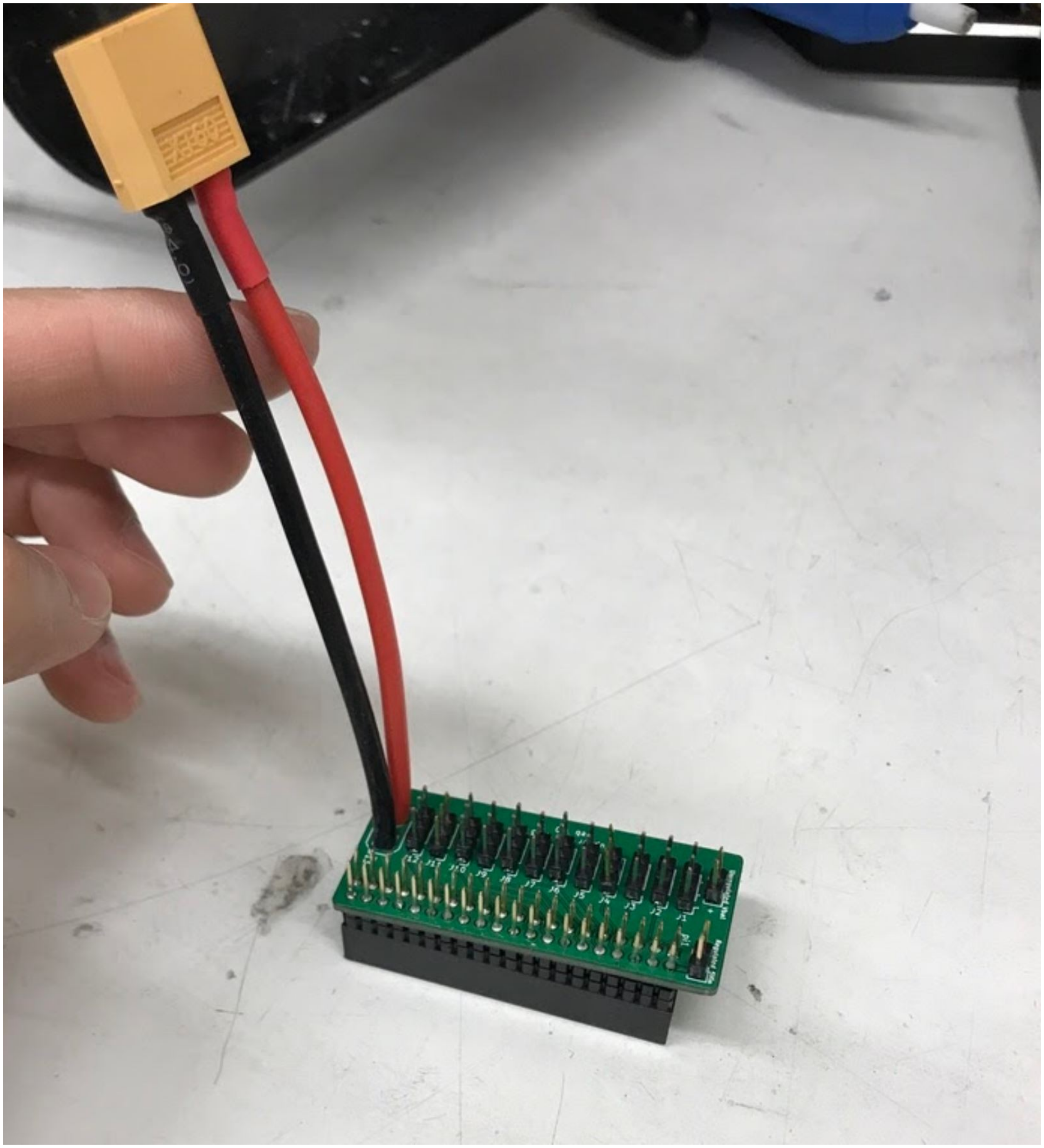
- No Video Available
- Materials: PCB, XT60 pigtail connector
- Tools: Soldering iron, vice

Instructions:

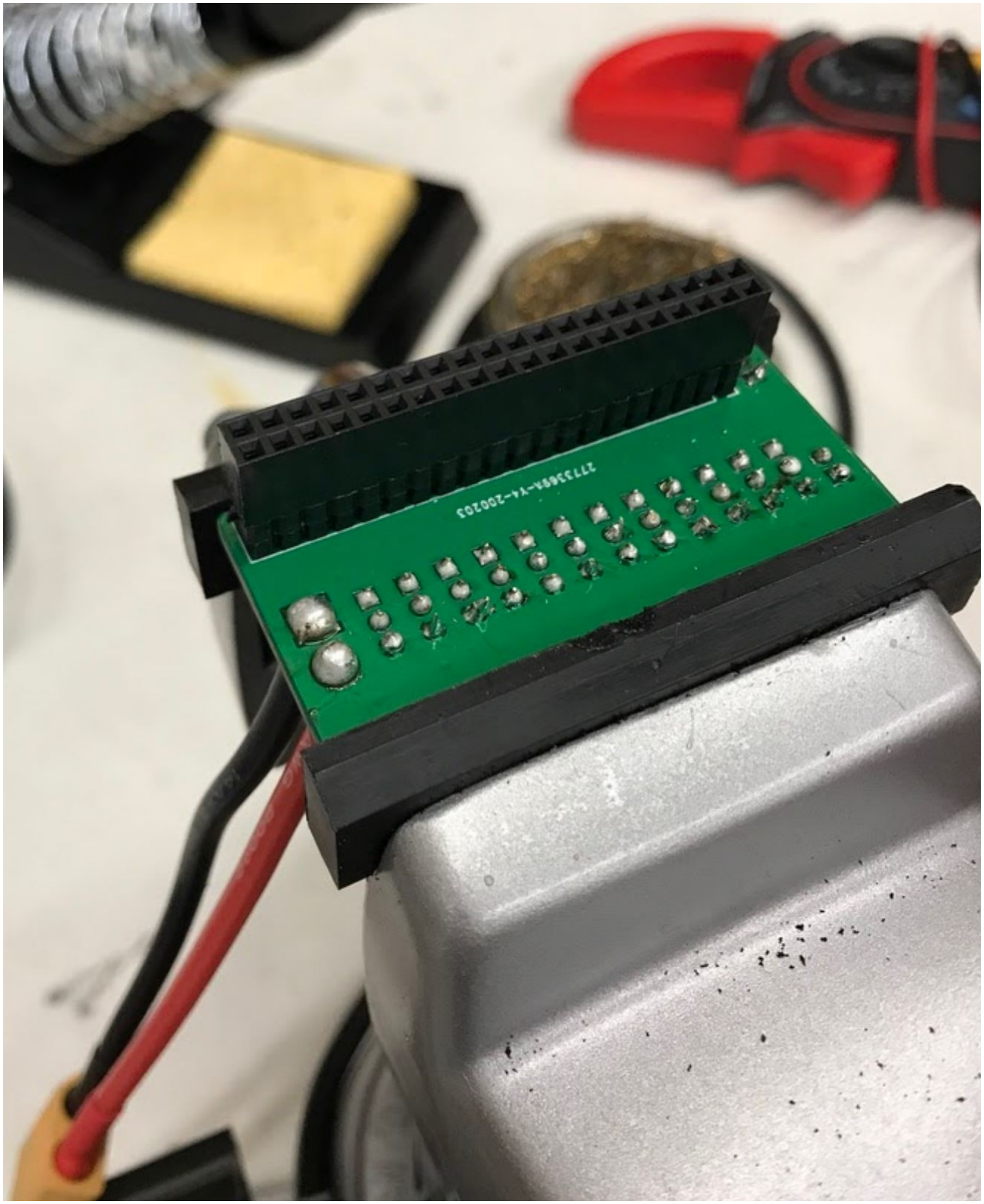
1. Insert the xt 60 pigtail from the top and solder from the bottom. Make sure you get the polarity correct! The PCB has little labels for the + and - wires of the xt 60 pigtail.

Male XT60 Pigtail (Female housing, male pins)

After soldering on the XT60 pigtail.



Another view of XT60 solder connection.



Step 5: Test the power distribution board for shorts

- No Video Available
- Materials: PCB

- Tools: Multimeter

Instructions:

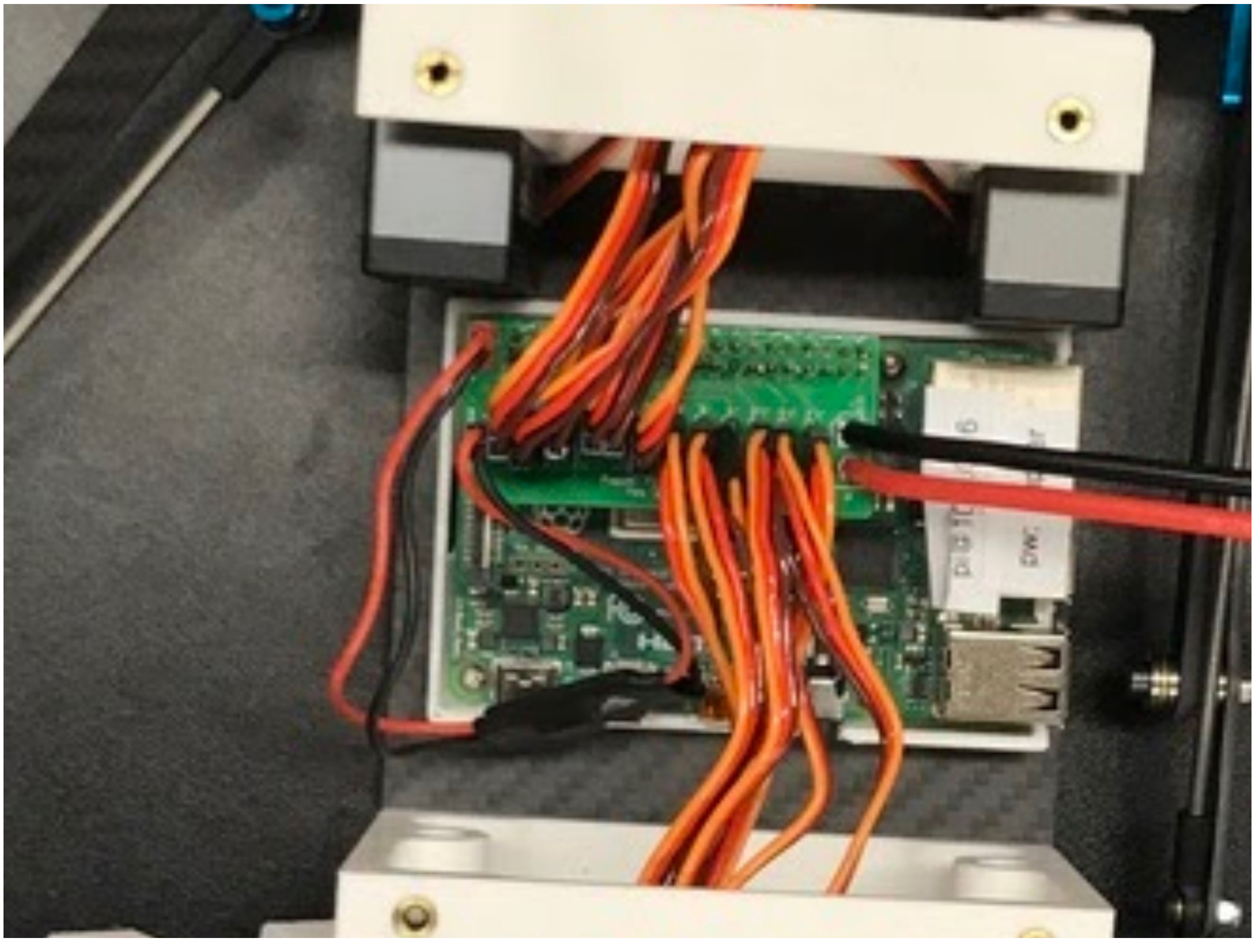
1. Inspect the board visually to make sure no solder blobs are shorting together
2. Turn the multimeter to the short detecting setting. This is usually indicated by a little speaker icon.
3. Test that the + and - pins of the xt 60 connector do not short together
4. Test that none of the signal wires short to + or - either.
5. Test that none of the signal wires short to each other.

Step 6: Test for servo power

- No Video Available
- Materials: PCB, servos
- Tools: none

Instructions:

1. Plug in your 2S lipo (never plug in anything more than 8.4V or it's very likely you'll burn out your servos)
2. Connect in a single servo into the board, noting the labels for the signal, -, and + wires. On servos, the signal wire is usually yellow or white.
3. Reference picture to determine correct wire orientation.
4. If the servo doesn't start smoking when you plug it in, good job!
5. Unplug the servo and battery for now.



Step 7: Plug in the 5V voltage regulator

- No Video Available
- Materials: PCB, 5V regulator (BEC)
- Tools: Soldering iron or crimper

Instructions:

1. We use a 5V BEC to reduce the 7.4-8.4V voltage from the battery to 5V for the Raspberry Pi. The 5V output of the BEC has a JST connector which mates nicely with the Regulated 5V in pins you soldered in step 4.
2. The input side of the BEC has a male JST connector which you should now snip off.
3. You can either strip these input wires and solder them to the Vbat holes directly, or you can crimp female dupont headers to the wires, put them in a 1x2 housing, and plug the wires into the Vbat pins.

Done!

Complete hip assembly and body assembly if you haven't done so already.

1.3 Software Installation

Contents

1.3.1 Setting up your Raspberry Pi

- Raspberry Pi 4
- SD Card (32GB recommended)
- Raspberry Pi 4 power supply (USB-C, 5V, >=3A)

- Ethernet cable

Preparing the Pi's SD card

From your desktop / laptop:

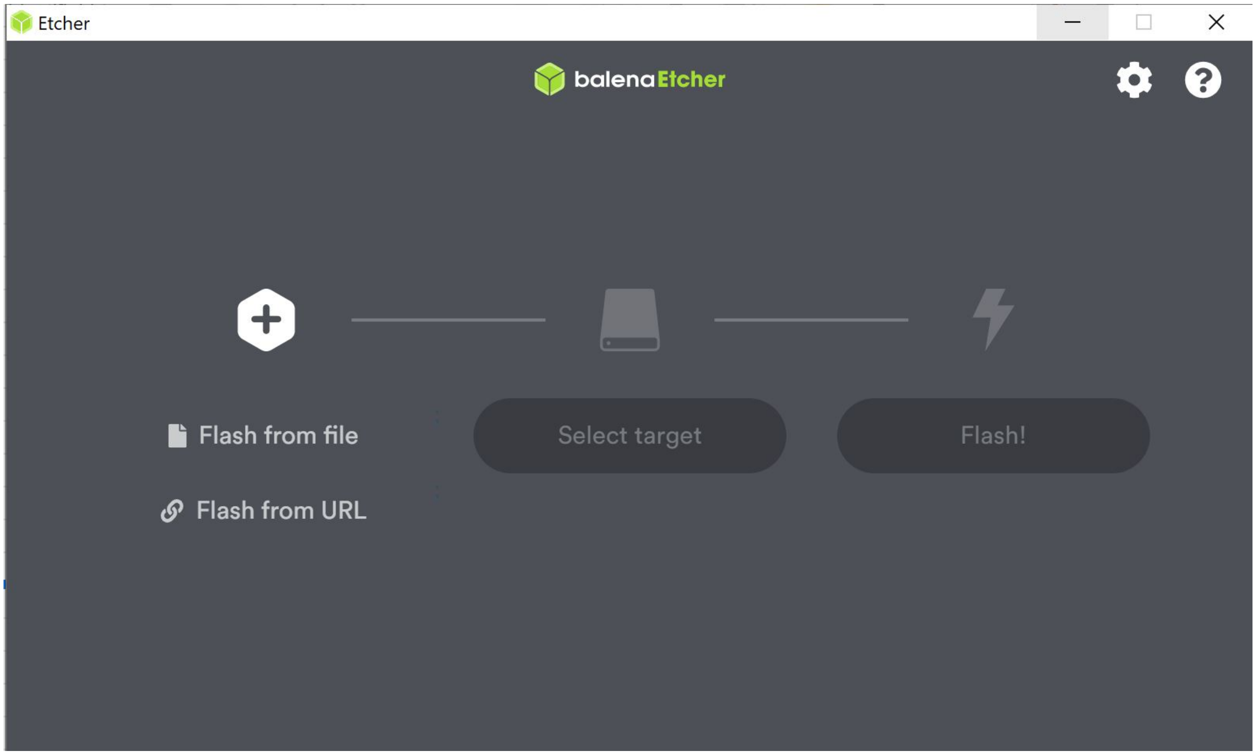
1. Put the SD card into your desktop / laptop.

2. Download this version of Raspbian

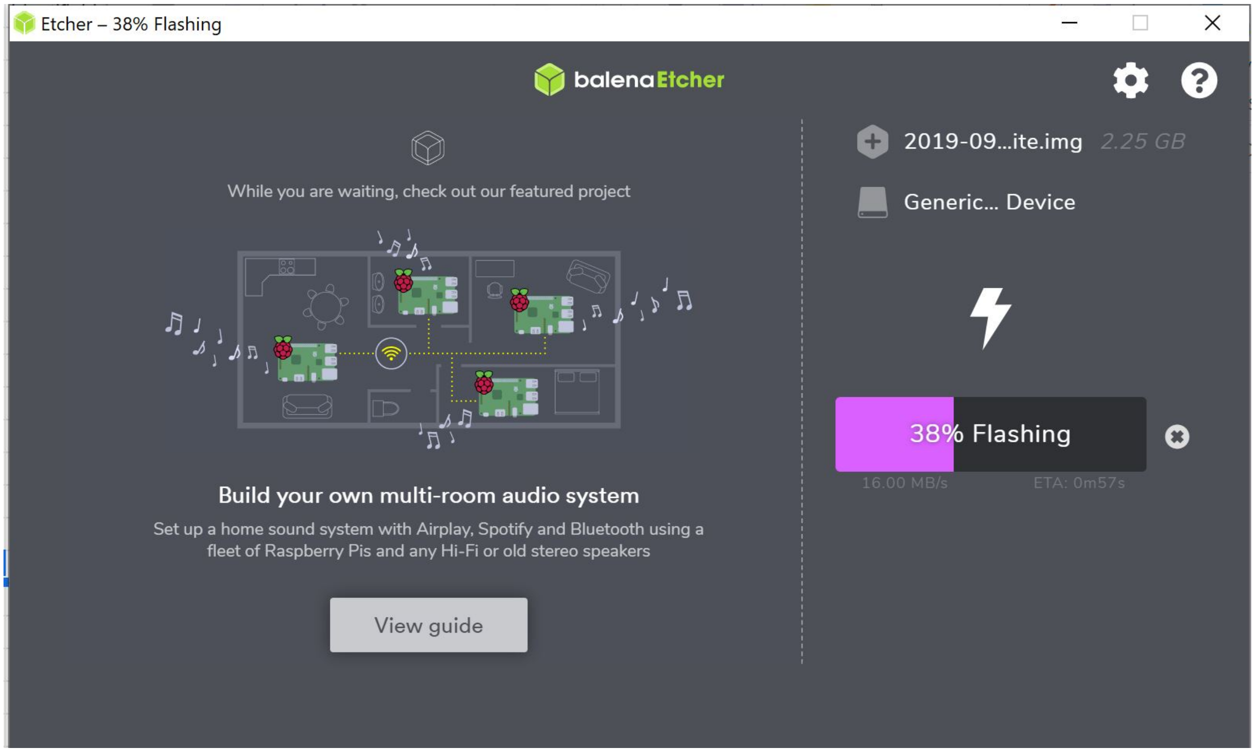
Use this version so everyone is using the same version. Unzip and extract the file.

3. Use etcher to flash the card.

- If you are using the recommended etcher, this is the start-up menu. Select 2019-09-26-raspbian-buster-lite.img (file inside zip)and the SD card.



- Image of SD card being flashed.

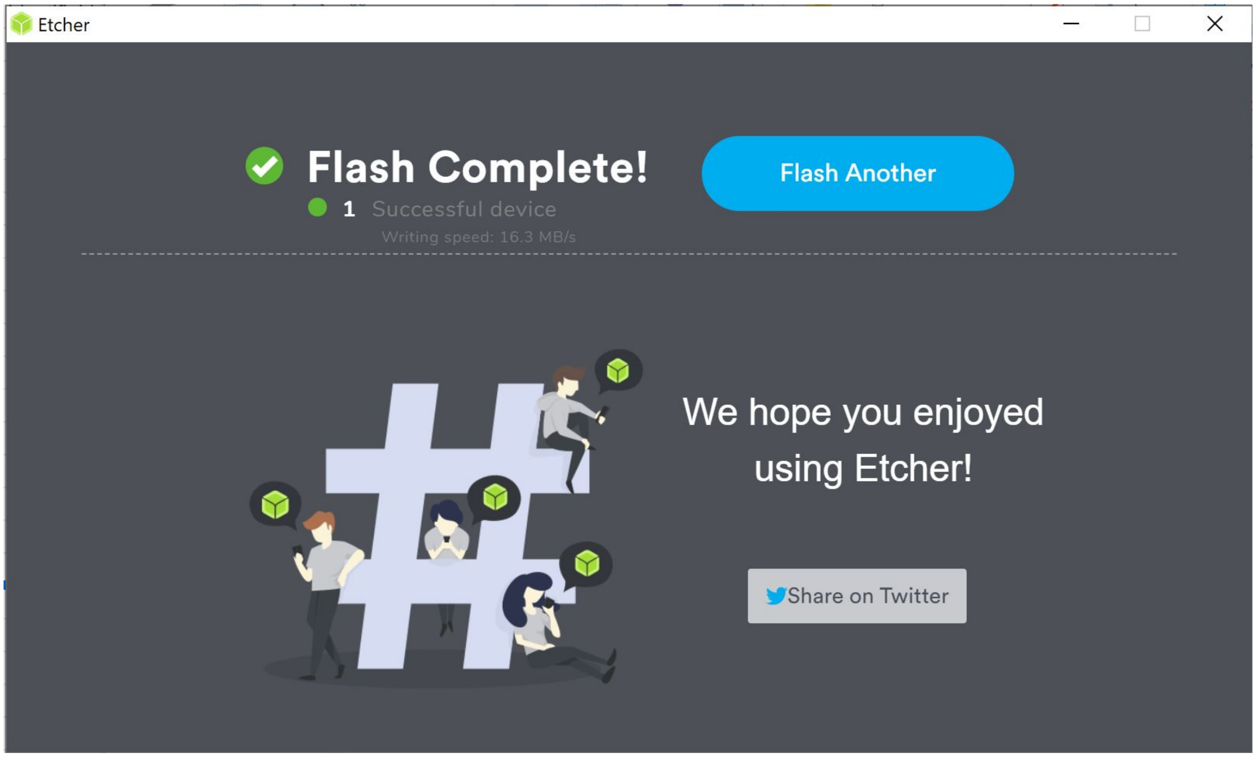


While you are waiting, check out our featured project

Build your own multi-room audio system

Set up a home sound system with Airplay, Spotify and Bluetooth using a fleet of Raspberry Pis and any Hi-Fi or old stereo speakers

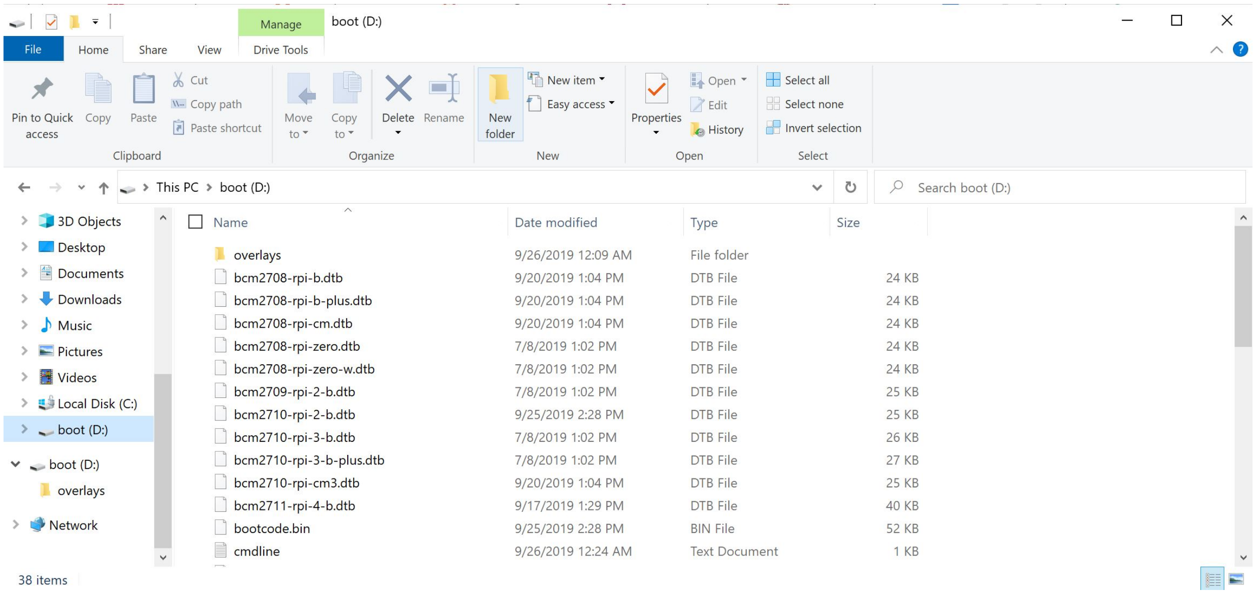
- Done!



We hope you enjoyed using Etcher!

4. Open up the SD card file system.

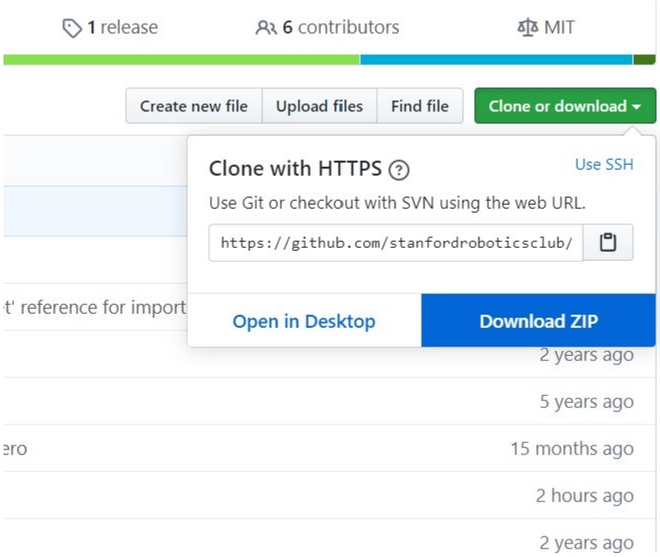
Sometimes it takes some time for your computer to read the SD card and show the boot folder. Try removing the SD card and putting it back in, if the problem persists.



Name	Date modified	Type	Size
overlays	9/26/2019 12:09 AM	File folder	
bcm2708-rpi-b.dtb	9/20/2019 1:04 PM	DTB File	24 KB
bcm2708-rpi-b-plus.dtb	9/20/2019 1:04 PM	DTB File	24 KB
bcm2708-rpi-cm.dtb	9/20/2019 1:04 PM	DTB File	24 KB
bcm2708-rpi-zero.dtb	7/8/2019 1:02 PM	DTB File	24 KB
bcm2708-rpi-zero-w.dtb	7/8/2019 1:02 PM	DTB File	24 KB
bcm2709-rpi-2-b.dtb	7/8/2019 1:02 PM	DTB File	25 KB
bcm2710-rpi-2-b.dtb	9/25/2019 2:28 PM	DTB File	25 KB
bcm2710-rpi-3-b.dtb	7/8/2019 1:02 PM	DTB File	26 KB
bcm2710-rpi-3-b-plus.dtb	7/8/2019 1:02 PM	DTB File	27 KB
bcm2710-rpi-cm3.dtb	9/20/2019 1:04 PM	DTB File	25 KB
bcm2711-rpi-4-b.dtb	9/17/2019 1:29 PM	DTB File	40 KB
bootcode.bin	9/25/2019 2:28 PM	BIN File	52 KB
cmdline	9/26/2019 12:24 AM	Text Document	1 KB

5. Download the latest release of the RPI-Setup repository.

- Unzip and extract all the files.



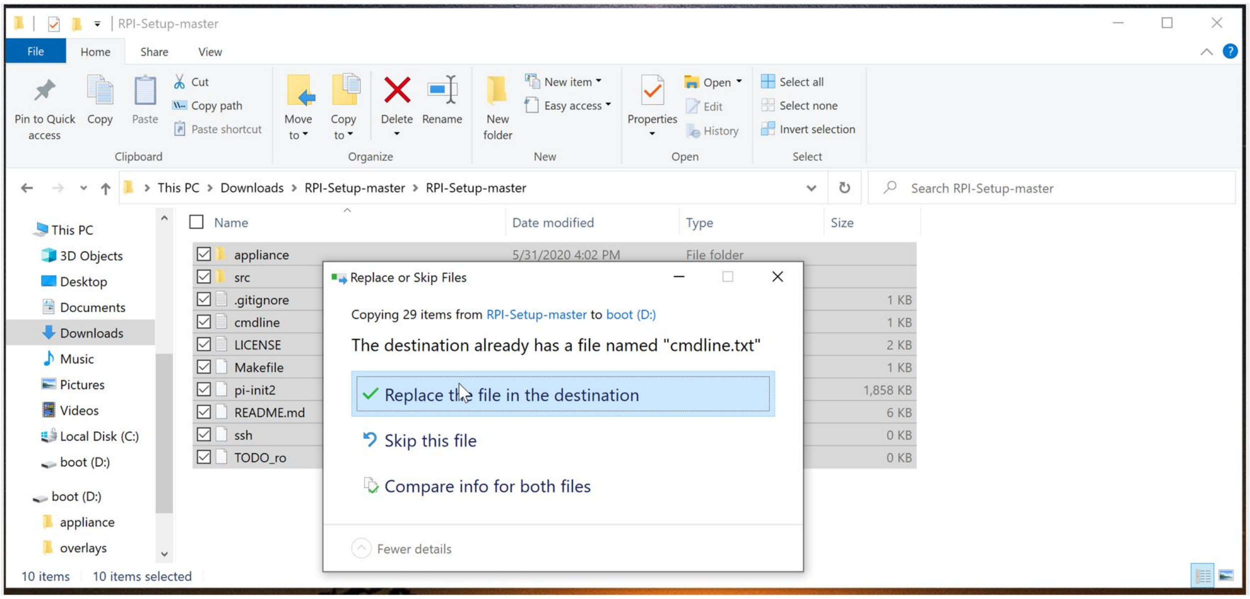
Clone with HTTPS

Use Git or checkout with SVN using the web URL.

```
https://github.com/stanfordroboticsclub/
```

6. Move all the files in the downloaded repository into the SD card.

- Replace any files that conflict so the repository's version overwrites the original version. You can now delete the zip file and the now empty folder.



	Name	Date modified	Type	Size
☑	appliance	5/31/2020 4:02 PM	File folder	
☑	src			
☑	.gitignore			1 KB
☑	cmdline			1 KB
☑	LICENSE			2 KB
☑	Makefile			1 KB
☑	pi-init2			1,858 KB
☑	README.md			6 KB
☑	ssh			0 KB
☑	TODO_ro			0 KB

Replace or Skip Files

Copying 29 items from RPI-Setup-master to boot (D:)

The destination already has a file named "cmdline.txt"

✓ Replace the file in the destination

Skip this file

Compare info for both files

Fewer details

Enabling Basic Functionality

1. Turn on your Raspberry Pi.

Remove SD card from computer and put it into your Raspberry Pi. Connect power to the Pi as well.

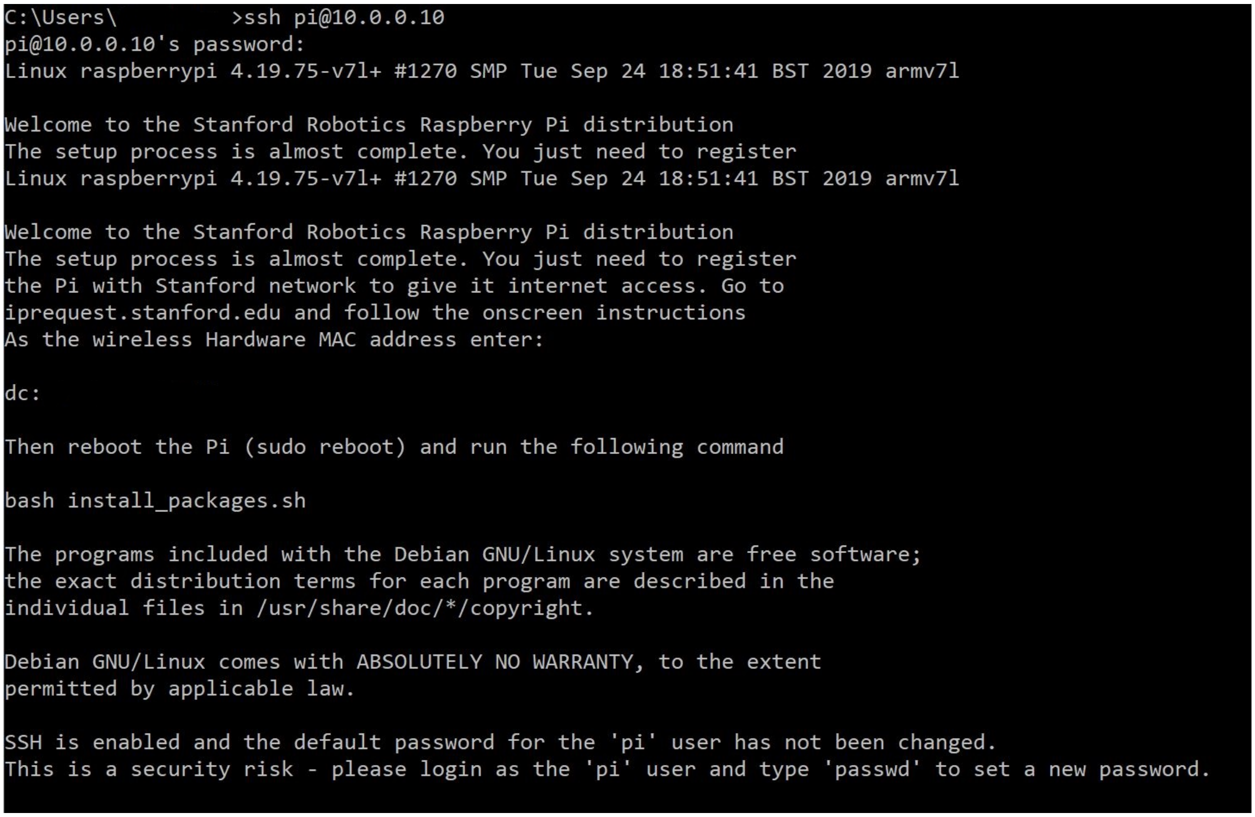
If your Pi does not boot, please try going back to step 3 "Use etcher to flash the card" and use this version of Rasbian instead: https://downloads.raspberrypi.org/raspbian_lite/images/raspbian_lite-2020-02-14/2020-02-13-raspbian-buster-lite.zip

2. Configure your computer to SSH into the robot

- To use ethernet for set up (recommended), connect the ethernet cable to your computer and the raspberry pi.
- Go to your network settings for the interface you wish to use (ethernet/wifi)
- Change your Configure IPv4: Manually
- Change your IP Address to something in range 10.0.0.X (If you are a part of Stanford Student Robotics pick something that doesn't colide with other systems from this document)
- Change your Subnet Mask: 255.255.255.0
- Leave the Router blank
- After disconnecting from the robot network remember to return those settings to what they orignially were, otherwise your internet on that interface won't work

2. SSH into the pi from your computer.

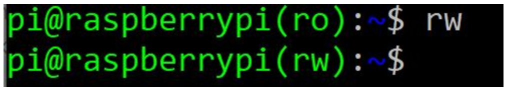
Run `ssh pi@10.0.0.10` (The default password is `raspberry`)



```
C:\Users\[redacted]>ssh pi@10.0.0.10
pi@10.0.0.10's password:
Linux raspberrypi 4.19.75-v7l+ #1270 SMP Tue Sep 24 18:51:41 BST 2019 armv7l

Welcome to the Stanford Robotics Raspberry Pi distribution
The setup process is almost complete. You just need to register
Linux raspberrypi 4.19.75-v7l+ #1270 SMP Tue Sep 24 18:51:41 BST 2019 armv7l

Welcome to the Stanford Robotics Raspberry Pi distribution
The setup process is almost complete. You just need to register
the Pi with Stanford network to give it internet access. Go to
iprequest.stanford.edu and follow the onscreen instructions
As the wireless Hardware MAC address enter:

dc:[redacted]

Then reboot the Pi (sudo reboot) and run the following command

bash install_packages.sh

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.
```

3. Enter read-write mode

Run `rw` in the robot shell. Confirm that the terminal prompt ends with (rw) instead of (ro).



```
pi@raspberrypi(ro):~$ rw
pi@raspberrypi(rw):~$
```

4. Get internet access

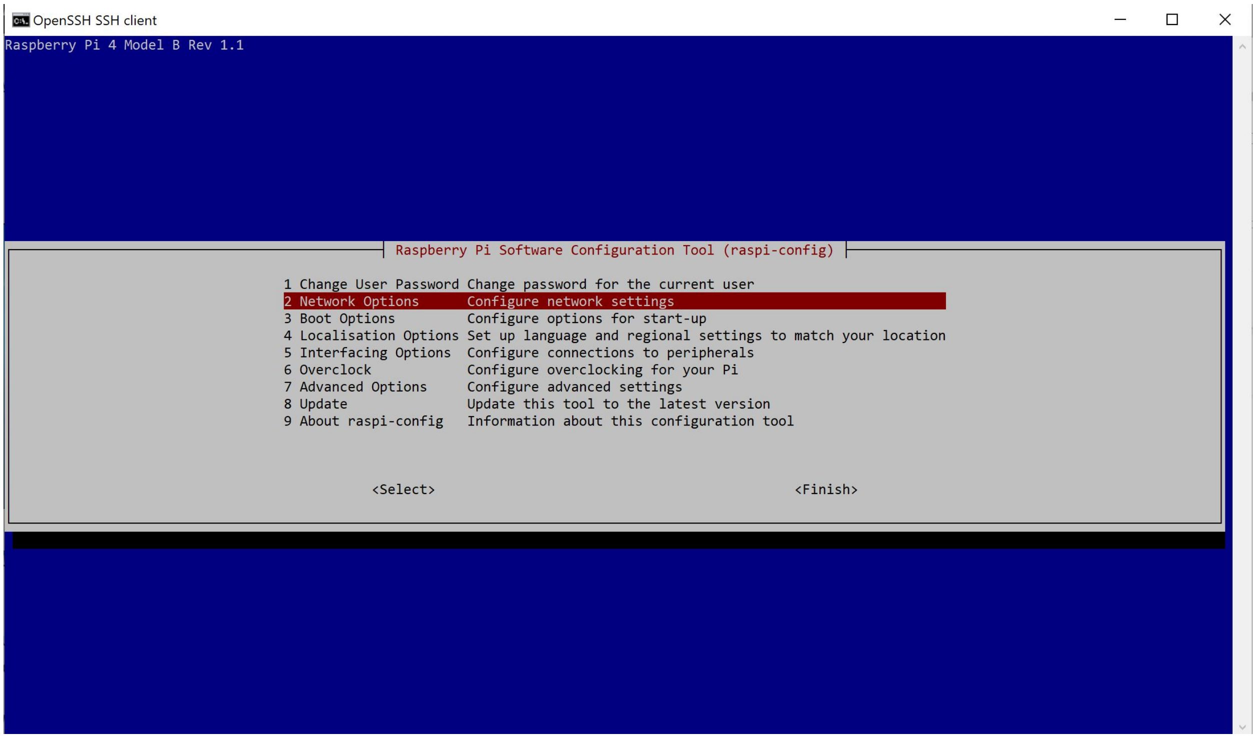
There are two methods for getting internet access: using the raspi-config tool on the Pi or changing the wpa_supplicant file on the SD card before inserting it into the Pi. If you're on Stanford campus, please follow the instructions in the next section instead since there are special requirements. If you're not on Stanford campus, using the raspi-config tool is simpler and recommended for beginners. However, modifying the wpa_supplicant file has the benefit that you can set the proper internet settings without SSHing into the Pi.

a. Raspi-config method

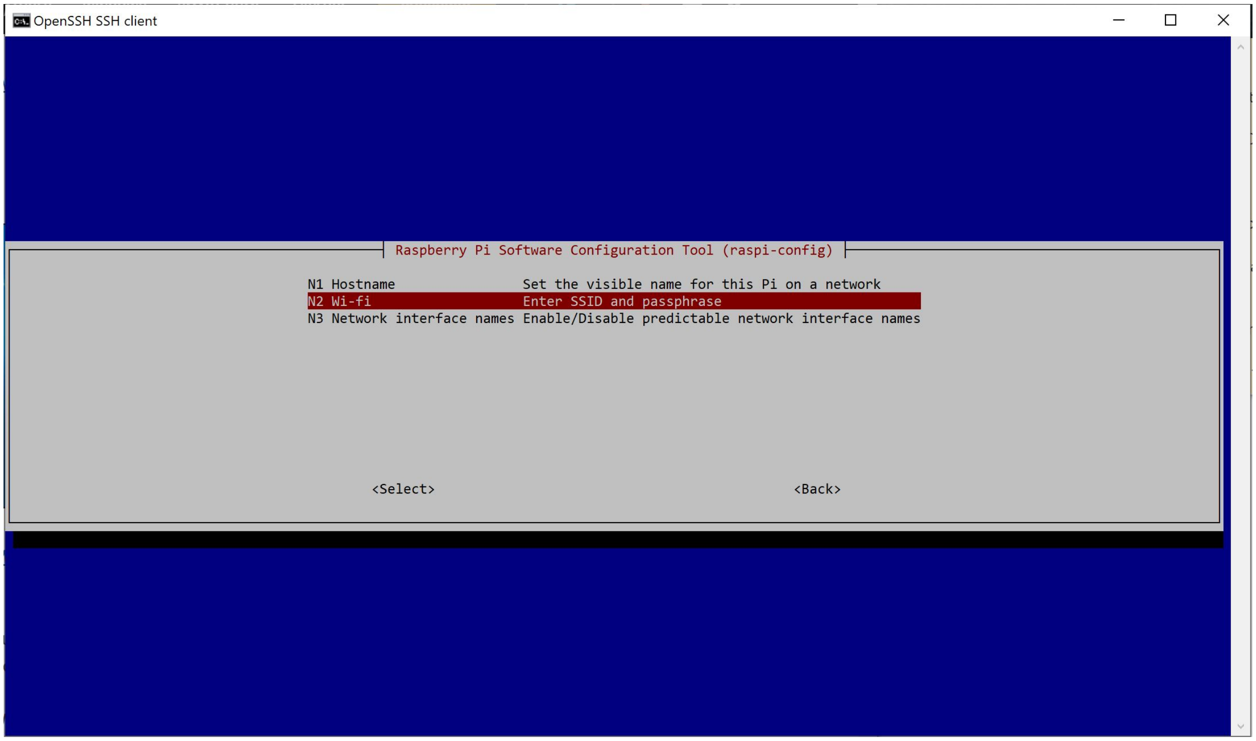
Once SSH'd into the Pi, run:

```
sudo raspi-config
```

This is the menu that will appear. Go to Network Options, then Wi-Fi and enter your SSID (Wi-Fi name, eg. Netgear, Linksys) and password.



```
Raspberry Pi 4 Model B Rev 1.1

            ┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

 1 Change User Password Change password for the current user
 2 Network Options      Configure network settings
 3 Boot Options         Configure options for start-up
 4 Localisation Options Set up language and regional settings to match your location
 5 Interfacing Options  Configure connections to peripherals
 6 Overclock            Configure overclocking for your Pi
 7 Advanced Options     Configure advanced settings
 8 Update               Update this tool to the latest version
 9 About raspi-config   Information about this configuration tool

           <Select>                                  <Finish>
```



```
            ┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

 N1 Hostname                Set the visible name for this Pi on a network
 N2 Wi-fi                   Enter SSID and passphrase
 N3 Network interface names Enable/Disable predictable network interface names

           <Select>                                  <Back>
```

b. Wpa_supplicant method

Edit **/etc/wpa_supplicant/wpa_supplicant.conf** as documented in this link , see “Adding the network details to the Raspberry Pi”. You can also see this link. Thanks to pi-init2 magic that file can be edited before the pi is ever turned on from **/boot/appliance/etc/wpa_supplicant/wpa_supplicant.conf**

4. [For Stanford Student Students] Get internet access at Stanford

- Plug your Pi in to power (over the onboard micro USB port). Log in to the Pi over SSH. In the welcome message that comes after the login line, look for the Pi's MAC address, which will appear under the line that says "wireless Hardware MAC address". Note that address down.
- Use another computer to navigate to iprequest.stanford.edu.
- Log in using your Stanford credentials.
- Follow the on-screen instructions to add another device:
  - **First page:** Device Type: Other, Operating System: Linux, Hardware Address: put Pi's MAC address
  - **Second page:** Make and model: Other PC, Hardware Addresses Wired: delete what's there, Hardware Addresses Wireless: put Pi's MAC address
- Confirm that the Pi is connected to the network:
  - Wait for an email (to your Stanford email) that the device has been accepted
  - **sudo reboot** on the Pi
  - After it's done rebooting, type ping www.google.com and make sure you are receiving packets over the network

5. Install prerequisites

- Run `sudo ./install_packages.sh`
- If the IP is still 10.0.0.10 you will be prompted to change it. The raspberry Pi IP should not be the same as your computer's IP, 10.0.0.Y.
- If the hostname is still raspberry you will be prompted to change it.
- You will be asked to enter the current time and date. You can skip to the next step if you'd like to automatically set the time and date.
- Run `sudo ./time_sync.sh` to automatically set the time and date.

What the RPI-Setup repo does

- Enables ssh. Because the password is kept unchanged (raspberry) ssh is only enabled on the ethernet interface. Comment out the ListenAddress lines from /boot/appliance/etc/ssh/sshd_config to enable it on all interfaces.
- Sets the Pi to connect to the robot network (10.0.0.X) over ethernet
- Expands the SD card file system
- Sets the file system up as read-only
- Prepares to connect to Stanford WiFi (see above for details)
- Gives the script to install tools and repos needed for development

1.3.2 Installing the StanfordQuadruped software on the Raspberry Pi

Steps

1. Connect to the Pi over SSH

Check that it has access to the internet. If you're having trouble SSH-ing into the Pi, please check the instructions for setting the Pi's ethernet settings linked in the previous step.

```
ssh pi@10.0.0.Y
```

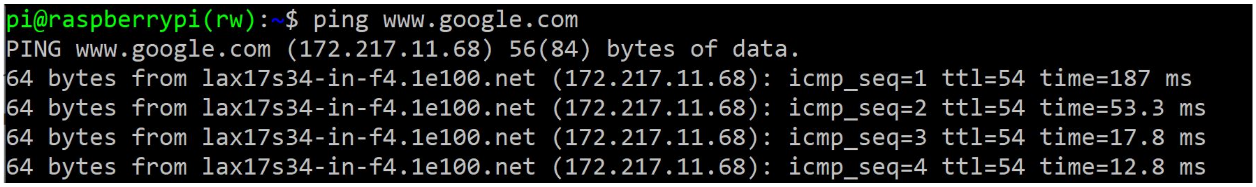
Here, "Y" is the IP address you chose for the Pi when running the install_packages.sh script. When prompted for the password, enter the default password "raspberry" or the one you set in the install_packages.sh script.

If you forgot what the Pi's IP address is, turn off the Pi, take out the SD card and put it in your computer. Then open the sd card folder and go to the folder: boot/appliance/etc/network/. Open the file called "interfaces" in a text editor. On line 19 it should show the IP address as "address 10.0.0.x".

2. Test for the internet connection.

```
ping www.google.com
```

This is what the output should look like:



```
pi@raspberrypi(rw):~$ ping www.google.com
PING www.google.com (172.217.11.68) 56(84) bytes of data.
64 bytes from lax17s34-in-f4.1e100.net (172.217.11.68): icmp_seq=1 ttl=54 time=187 ms
64 bytes from lax17s34-in-f4.1e100.net (172.217.11.68): icmp_seq=2 ttl=54 time=53.3 ms
64 bytes from lax17s34-in-f4.1e100.net (172.217.11.68): icmp_seq=3 ttl=54 time=17.8 ms
64 bytes from lax17s34-in-f4.1e100.net (172.217.11.68): icmp_seq=4 ttl=54 time=12.8 ms
```

If that doesn't work, do:

```
ifconfig
```

and check the wlan0 portion to check if you have an IP address and other debugging info.

3. Clone this repo (on the Pi)

```
git clone https://github.com/stanfordroboticsclub/StanfordQuadruped.git
```

4. Install requirements (on the Pi)

```
cd StanfordQuadruped
sudo bash install.sh
```

5. Power-cycle the robot

Unplug the battery, wait about 30 seconds, and then plug it back in.

6. Verify everything is working

1. If you just powered on the Pi, wait about 30 seconds until the green light stops blinking.

2. SSH into the robot

   - Run `ssh pi@10.0.0.xx (where xx is the IP address you chose for the robot)`

3. Check the status for the joystick service

   - Run `sudo systemctl status joystick`
   - If you haven't yet connected the PS4 controller, it should say something like

```
pi@pupper(rw):~/StanfordQuadruped$ sudo systemctl status joystick
 joystick.service - Pupper Joystick service
Loaded: loaded (/home/pi/PupperCommand/joystick.service; enabled; vendor␣
↪preset: enabled)
Active: active (running) since Sun 2020-03-01 06:57:20 GMT; 1s ago
Main PID: 5692 (python3)
    Tasks: 3 (limit: 4035)
Memory: 7.1M
CGroup: /system.slice/joystick.service
        ├─5692 /usr/bin/python3 /home/pi/PupperCommand/joystick.py
        └─5708 hcitool scan --flush

Mar 01 06:57:20 pupper systemd[1]: Started Pupper Joystick service.
Mar 01 06:57:21 pupper python3[5692]: [info][controller 1] Created devices /
↪dev/input/js0 (joystick) /dev/input/event0 (evdev)
Mar 01 06:57:21 pupper python3[5692]: [info][bluetooth] Scanning for devices
```

4. Connect the PS4 controller to the Pi by putting it pairing mode.

   - To put it into pairing mode, hold the share button and circular Playstation button at the same time until it starts making quick double flashes.
   - If it starts making slow single flashes, hold the Playstation button down until it stops blinking and try again.

5. Once the controller is connected, check the status again

   - Run `sudo systemctl status joystick`
   - It should now look something like:

```
pi@pupper(rw):~/StanfordQuadruped$ sudo systemctl status joystick
 joystick.service - Pupper Joystick service
Loaded: loaded (/home/pi/PupperCommand/joystick.service; enabled; vendor␣
↪preset: enabled)
Active: active (running) since Sun 2020-03-01 06:57:20 GMT; 55s ago
Main PID: 5692 (python3)
    Tasks: 2 (limit: 4035)
Memory: 7.3M
CGroup: /system.slice/joystick.service
        └─5692 /usr/bin/python3 /home/pi/PupperCommand/joystick.py

Mar 01 06:57:20 pupper systemd[1]: Started Pupper Joystick service.
Mar 01 06:57:21 pupper python3[5692]: [info][controller 1] Created devices /
↪dev/input/js0 (joystick) /dev/input/event0 (evdev)
Mar 01 06:57:21 pupper python3[5692]: [info][bluetooth] Scanning for devices
Mar 01 06:58:12 pupper python3[5692]: [info][bluetooth] Found device␣
↪A0:AB:51:33:B5:A0
Mar 01 06:58:13 pupper python3[5692]: [info][controller 1] Connected to␣
↪Bluetooth Controller (A0:AB:51:33:B5:A0)
Mar 01 06:58:14 pupper python3[5692]: running
```

(continues on next page)

(continued from previous page)

```
Mar 01 06:58:14 pupper python3[5692]: [info][controller 1] Battery: 50%
```

- If the pi can't find the joystick after a minute or two, it's possible that the pi's bluetooth controller was never turned on. Run `sudo hciconfig hci0 up` to turn the radio on. Then restart the pi.

6. Check the status of the robot service
   - Run `sudo systemctl status robot`
   - The output varies depending on the order of you running various programs, but just check that it doesn't have any red text saying that it failed.
   - If it did fail, usually this fixes it: `sudo systemctl restart robot`

7. Done!

Continue to Calibration.

1.4 Calibration

Calibration is a necessary step before running the robot because that we don't yet have a precise measurement of how the servos arms are fixed relative to the servo output shafts. Running the calibration script will help you determine this rotational offset by prompting you to align each of the 12 degrees of freedom with a known angle, such as the horizontal or the vertical.

1.4.1 Materials

1. Finished robot
2. Some sort of stand to hold the robot up so that its legs can extend without touching the ground/table.

1.4.2 Steps

1. MangDang produced a video illustrating the calibration steps outlined below. You can stop watching at 17:00 because the new code automatically writes the calibration numbers rather than requiring you to edit the calibration file manually.
2. Plug in your 2S Lipo battery
3. SSH into the robot as done in the installation section
4. Stop the robot script from taking over the PWM outputs:

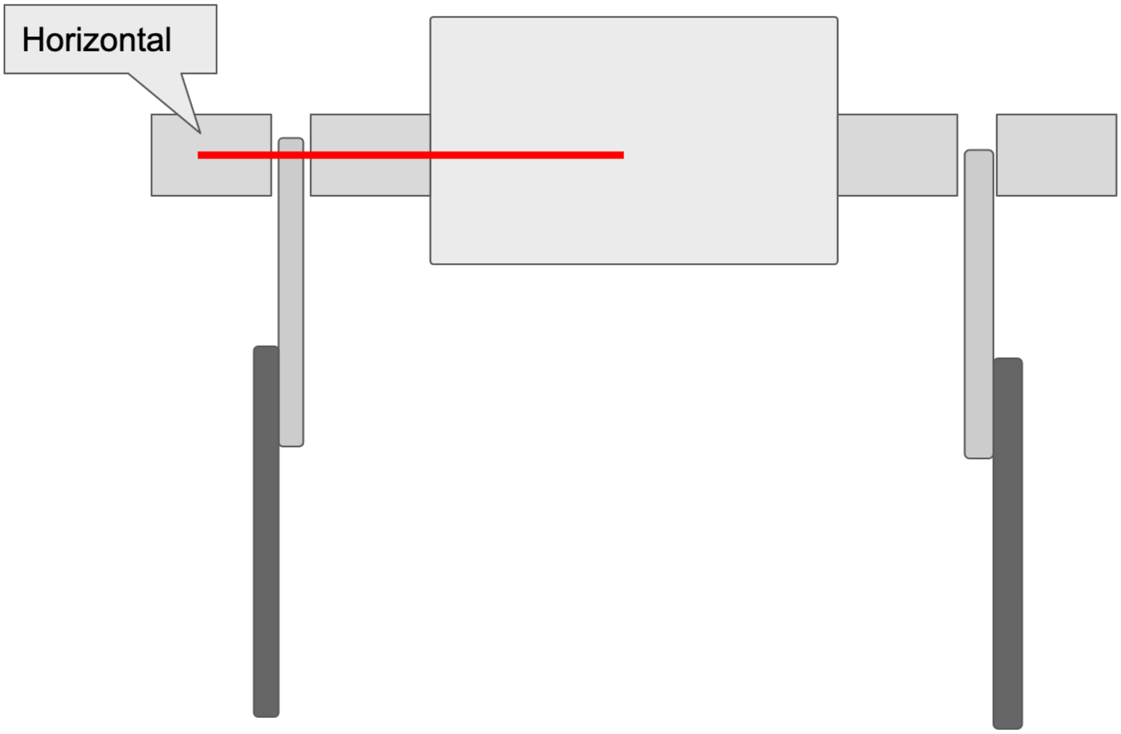
```
rw
sudo systemctl stop robot
```

5. Run the calibration script
   - The calibration script will prompt you through calibrating each of pupper's 12 servo motors. When it asks you to move a link to the horizontal position, you might be wondering what exactly counts as making the link horizontal. The answer is to align the *joint centers* of each link. For example, when aligning the upper link to the horizontal, you'll want to the line between the servo spline and bolt that connects the upper link to the lower link to be as horizontal as possible.:

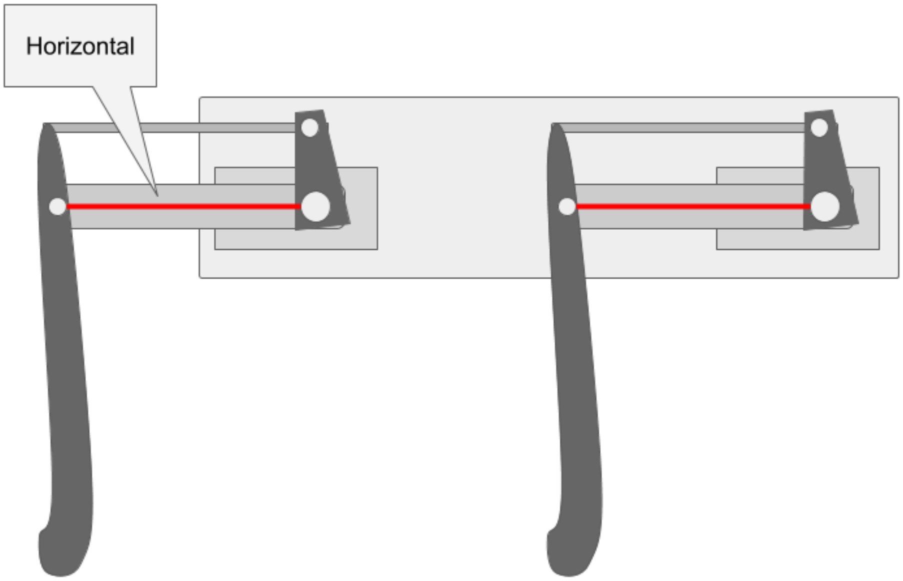
```
cd StanfordQudruped
sudo pigpiod
python3 calibrate_servos.py
```

- The images below illustate the horizontal and vertical positions mentioned in the calibration script.
- If your servos can't reach these positions, it's likely the servo discs and/or arms were assembled incorrectly.

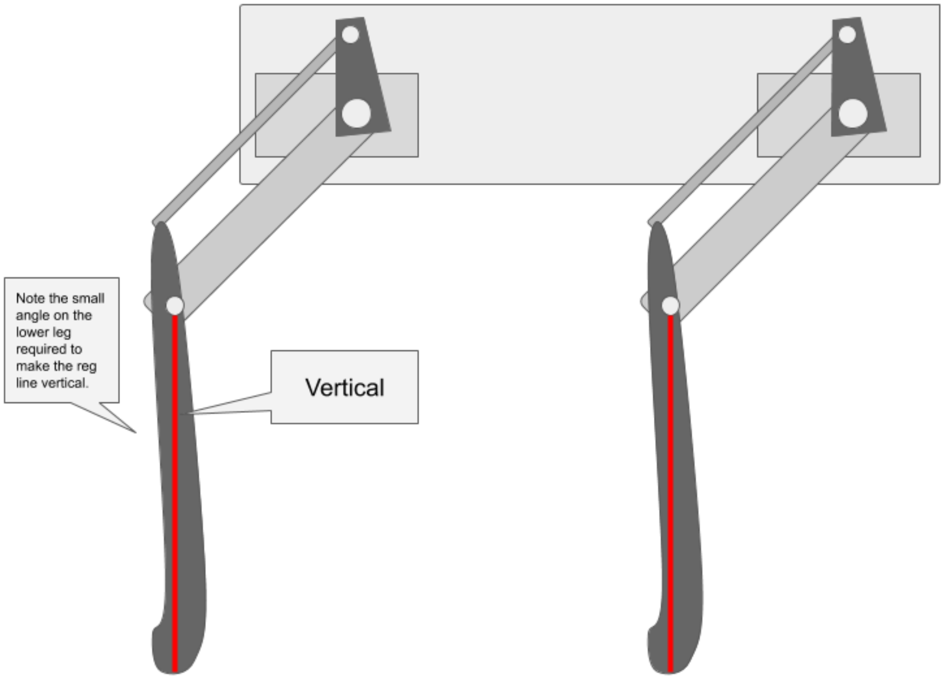
Correct alignment for the ab/adduction motors:



Correct alignment for the upper link:



Correct alignment for the lower link:


Note the small angle on the lower leg required to make the reg line vertical.
Vertical

1. Re-enable the robot script:

```
sudo systemctl start robot
```

1.5 Robot operation

1.5.1 Running the robot

1. Plug in your 2S Lipo battery.
   - If you followed the instructions above, the code will automatically start running on boot.
   - If you want to turn this feature off, ssh into the robot, go into rw mode, and then do:

     ```
     sudo systemctl disable robot
     ```

2. Connect the PS4 controller to the Pi by putting it pairing mode.
   - To put it into pairing mode, hold the share button and circular Playstation button at the same time until it starts making quick double flashes.
   - If it starts making slow single flashes, hold the Playstation button down until it stops blinking and try again.
3. Wait until the controller binds to the robot, at which point the controller should turn a dim green (or whatever color you chose in pupper/HardwareConfig.py for the deactivated color).
4. Press L1 on the controller to "activate" the robot. The controller should turn bright green (or again, whatever you chose in HardwareConfig).
5. You're good to go! Check out the controls section below for operating instructions.

1.5.2 Robot controls

- L1: Press to toggle active mode and deactivate mode.
  - Note: the PS4 controller's front light will change colors to indicate if the robot is deactivated or activated.
- R1: Press to transition between Rest mode and Trot mode
- Left joystick
  - Forward/back: moves the robot forward/backwards when in Trot mode
  - Left/right: moves the robot left/right when in Trot mode
- Right joystick
  - Forward/back: pitches the robot forward and backward
  - Left/right: turns the robot left and right
- D-Pad
  - Forward/back: raises and lowers the body
  - Left/rights: rolls the body left/right
- "X" button: Press it three times to complete a full hop

1.5.3 Important Notes

- PS4 controller pairing instructions (repeat of instructions above)
  - To put it into pairing mode, hold the share button and circular Playstation button at the same time until it starts making quick double flashes.

   - If it starts making slow single flashes, hold the Playstation button down until it stops blinking and try again.

- Battery voltage
  - If you power the robot with anything higher than 8.4V (aka >2S) you'll almost certainly fry all your expensive servos!
  - Also note that you should attach a lipo battery alarm to your battery when running the robot so that you are know when the battery is depleted. Discharging your battery too much runs the risk of starting a fire, especially if you try to charge it again after it's been completely discharged. A good rule-of-thumb for know when a lipo is discharged is checking whether the individual cell voltages are below 3.6V.
  - The robot will walk much more poorly when the battery is mostly discharged since a lower voltage is going to the motors.
- Feet!
  - Using the bare carbon fiber as feet works well for grippy surfaces, including carpet. If you want to use the robot on a more slippery surface, we recommend buying rubber grommets (McMaster #90131A101) and fastening them to the pre-drilled holes in the feet.

1.5.4 Tuning

- You can play around with different walking parameters by changing the config file `StanfordQuadruped/pupper/Config.py`
  - `self.max_x_velocity` [m/s]: The maximum forward/back trotting velocity
  - `self.max_y_velocity` [m/s]: Max left/right trotting velocity
  - `self.max_yaw_rate` [rad/s]: Max turning velocity
  - `self.z_clearance` [m]: How how the robot tries to lift each leg off the ground during swing. It's called z_clearance because it's the maximum distance in the z-axis between the foot and ground during swing. You can increase this value to make the robot step higher.
  - `self.overlap_time` [s]: Amount of time per step that the robot has all of its legs on the ground. Increase this value for more stable walking.
  - `self.swing_time` [s]: Amount of time the robot has each foot in the air for.

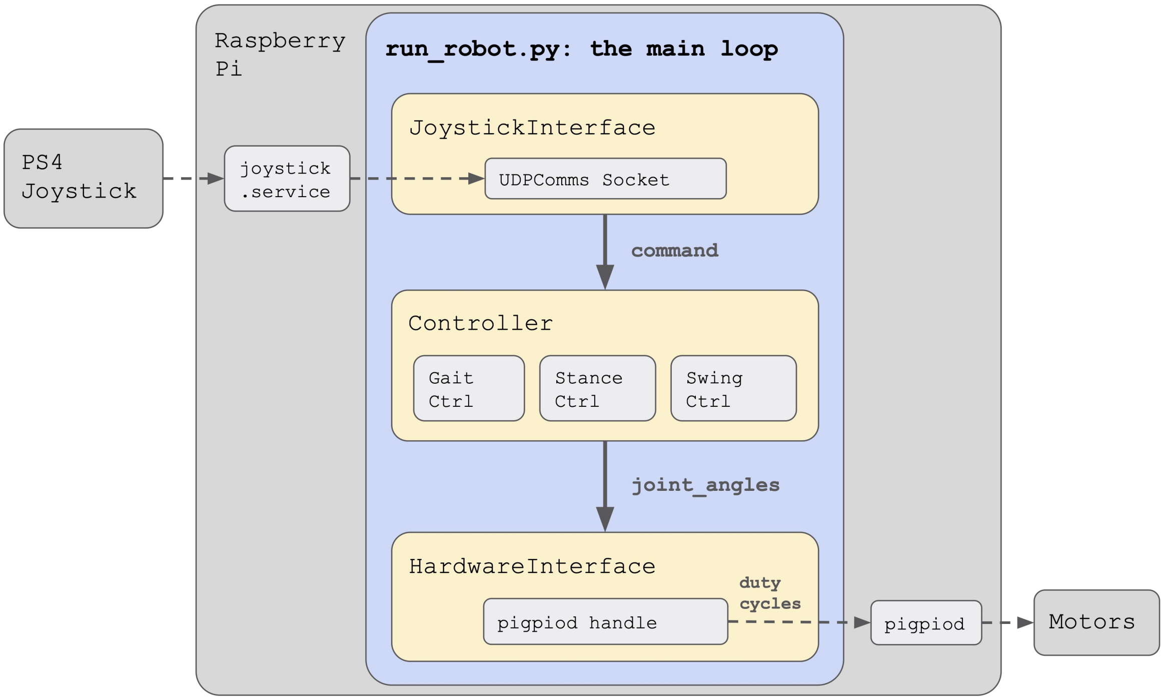
1.6 Mechanical Design

Fusion 360 CAD model: https://a360.co/2TEh4gQ

Power distribution pcb files: https://github.com/stanfordroboticsclub/Pupper-Raspi-PDB/

1.7 Controller Description

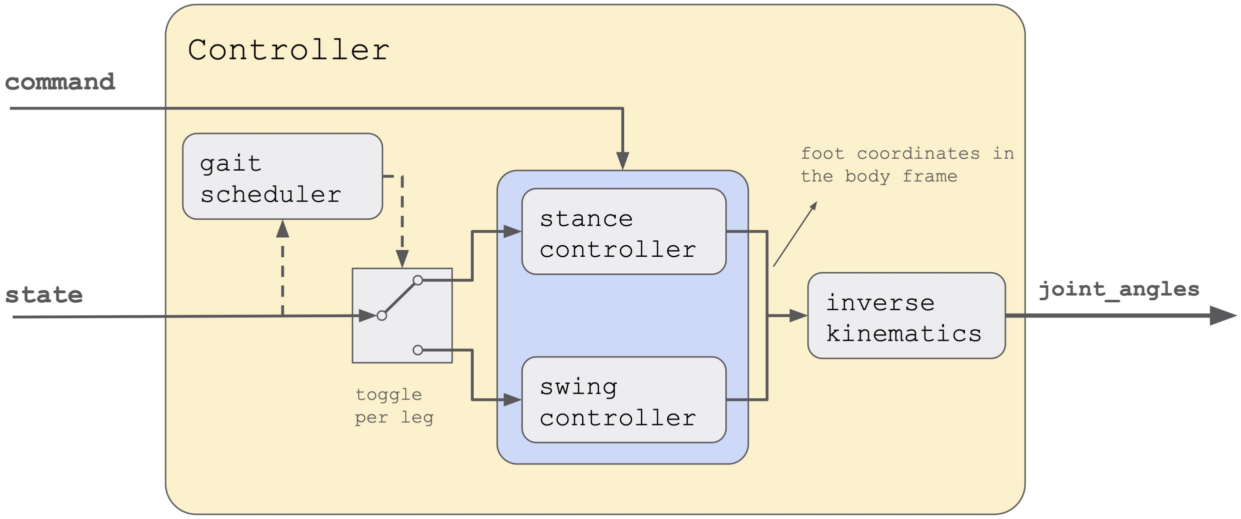
1.7.1 Main Loop



The main program is **run_robot.py** which is located in this directory. The robot code is run as a loop, with a joystick interface, a controller, and a hardware interface orchestrating the behavior.

The joystick interface is responsible for reading joystick inputs from a UDP socket and converting them into a generic robot **command** type. A separate program, **joystick.py**, publishes these UDP messages, and is responsible for reading inputs from the PS4 controller over bluetooth. The controller does the bulk of the work, switching between states (trot, walk, rest, etc) and generating servo position targets. A detailed model of the controller is shown below. The third component of the code, the hardware interface, converts the position targets from the controller into PWM duty cycles, which it then passes to a Python binding to **pigpiod**, which then generates PWM signals in software and sends these signals to the motors attached to the Raspberry Pi.

1.7.2 Controller Detail



This diagram shows a breakdown of the robot controller. Inside, you can see four primary components: a gait scheduler (also called gait controller), a stance controller, a swing controller, and an inverse kinematics model.

The gait scheduler is responsible for planning which feet should be on the ground (stance) and which should be moving forward to the next step (swing) at any given time. In a trot for example, the diagonal pairs of legs move in sync and take turns between stance and swing. As shown in the diagram, the gait scheduler can be thought of as a conductor for each leg, switching it between stance and swing as time progresses.

The stance controller controls the feet on the ground, and is actually quite simple. It looks at the desired robot velocity, and then generates a body-relative target velocity for these stance feet that is in the opposite direction as the desired velocity. It also incorporates turning, in which case it rotates the feet relative to the body in the opposite direction as the desired body rotation.

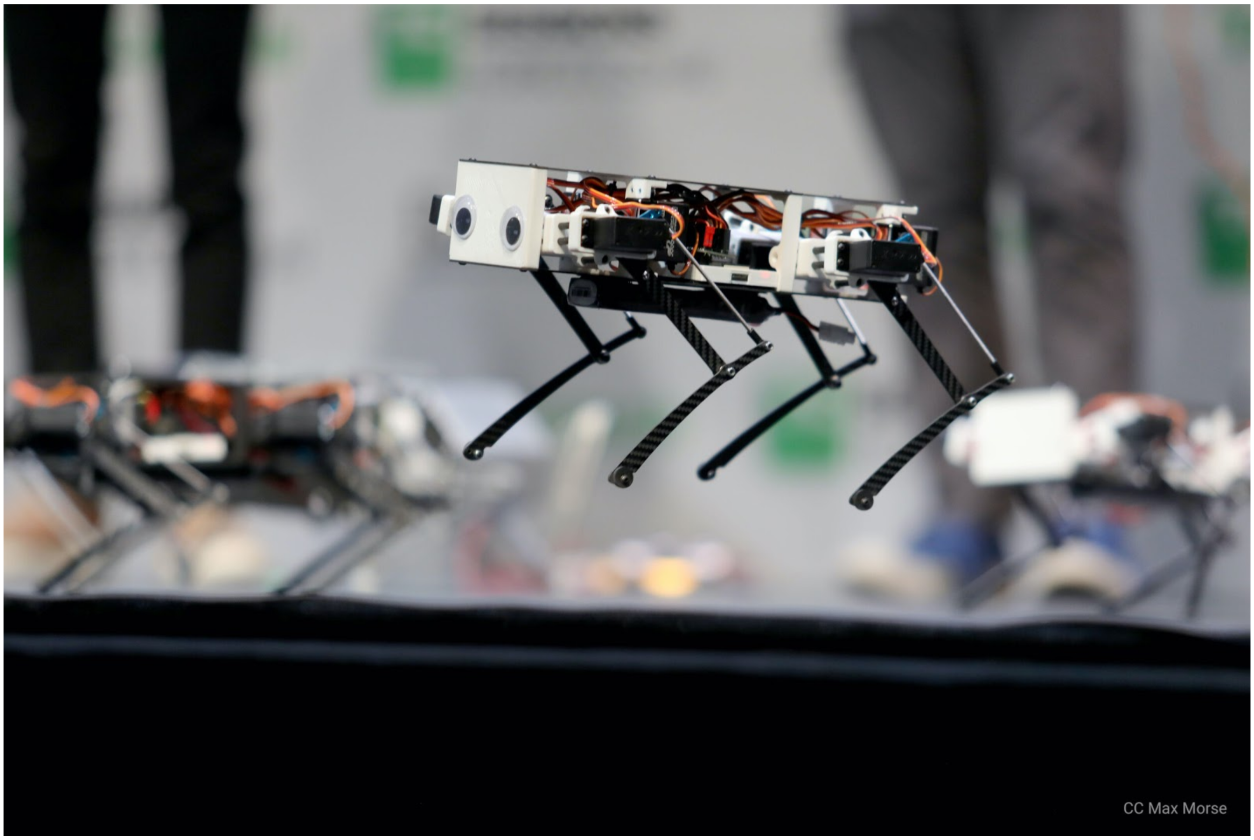
The swing controller picks up the feet that just finished their stance phase, and brings them to their next touchdown location. The touchdown locations are selected so that the foot moves the same distance forward in swing as it does backwards in stance. For example, if in stance phase the feet move backwards at -0.4m/s (to achieve a body velocity of +0.4m/s) and the stance phase is 0.5 seconds long, then we know the feet will have moved backwards -0.20m. The swing controller will then move the feet forwards 0.20m to put the foot back in its starting place. You can imagine that if the swing controller only put the leg forward 0.15m, then every step the foot would lag more and more behind the body by -0.05m.

Both the stance and swing controllers generate target positions for the feet in cartesian coordinates relative the body center of mass. It's convenient to work in cartesian coordinates for the stance and swing planning, but we now need to convert them to motor angles. This is done by using an inverse kinematics model, which maps between cartesian body coordinates and motor angles. These motor angles, also called joint angles, are then populated into the **state** variable and returned by the model.

1.8 Help

You can post any questions you might have to our Google Group: https://groups.google.com/forum/#!forum/stanford-quadrupeds

You can also email me at nathankau@stanford.edu.



CC Max Morse