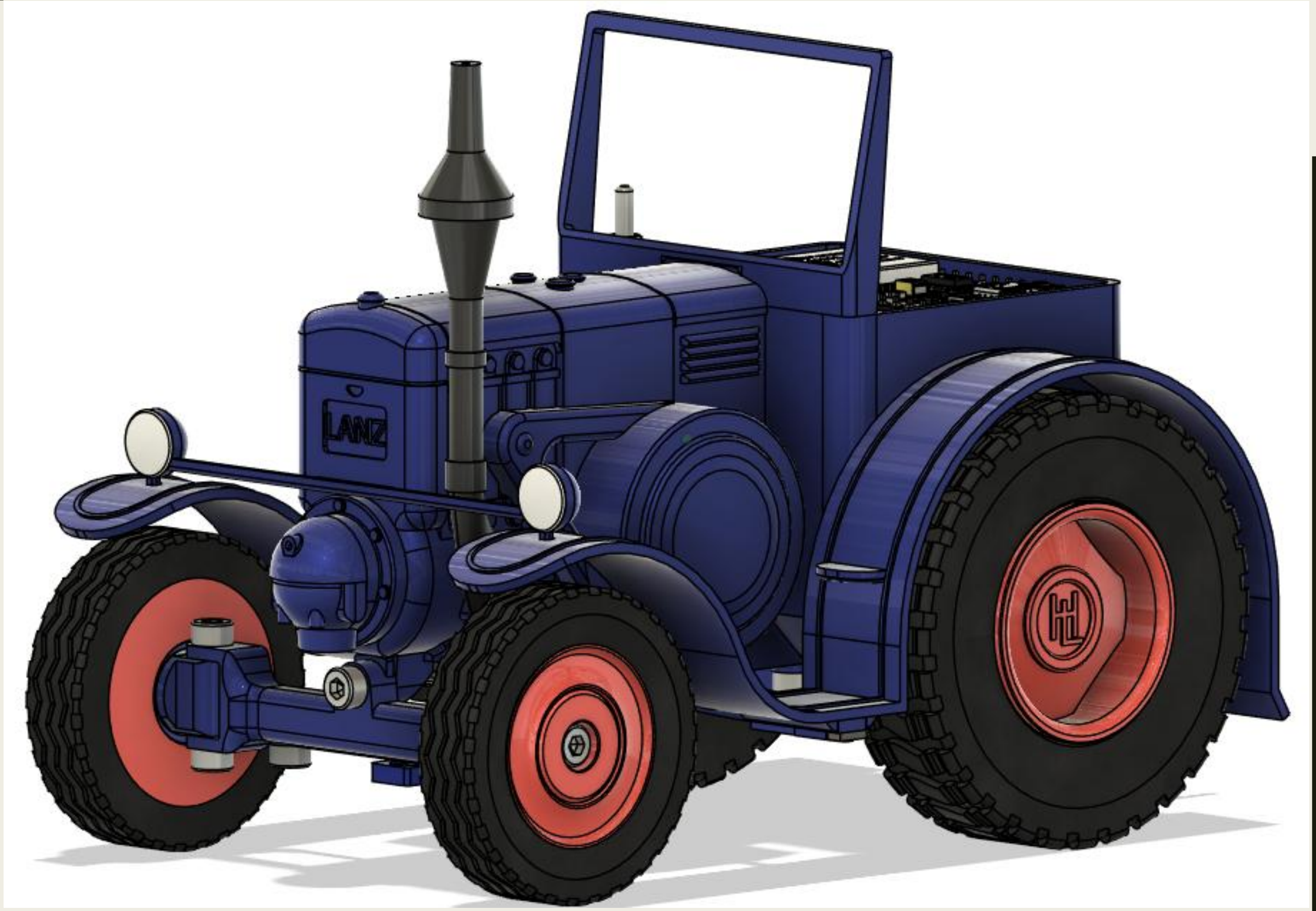
Assembly instructions for

Lanz D9506 „Ackerluft-Bulldog“

1/20 scale



Rev. 2

Contents

a) General info

This model is based on the rather famous Lanz D9506 produced by Heinrich Lanz AG in Mannheim, Germany.

Founded in 1859, they became famous for their 1 cylinder 10 liter engines capable of producing up to 55hp at speeds of typically around 300-450rpm. Early models didn‘t feature a reverse gear, instead the operator had to reverse the direction of the engine.

In the 1950s the Heinrich Lanz AG was sold to John Deere & Company, which took over the production plant in Mannheim. The last „Lanz Bulldog“, as they were called, was built in the early 1960s after which the production of John Deere tractors took over.

Source (German): https://de.wikipedia.org/wiki/Heinrich_Lanz_AG

General info

The D9506 was the luxury variant of the more basic Lanz D9500. Some of the added improvements were a 6 speed gearbox – instead of a 3 speed – and better tires compared to the D9500. Alone with these changes, the engineers managed to get 30% more pulling force from essentially the same engine.

The engine delivered 38hp at 630rpm and could at peak power output a maximum of 45hp for short durations.

With this power you could pull up to 1700kg at speeds up to 16km/h

From: https://de.wikibooks.org/wiki/Traktorenlexikon:_Lanz_D_9506

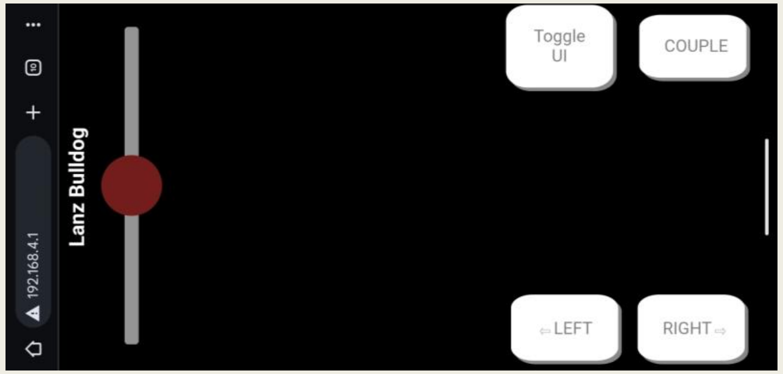
General info

You can controll this modell via two different methods from the same Webpage.

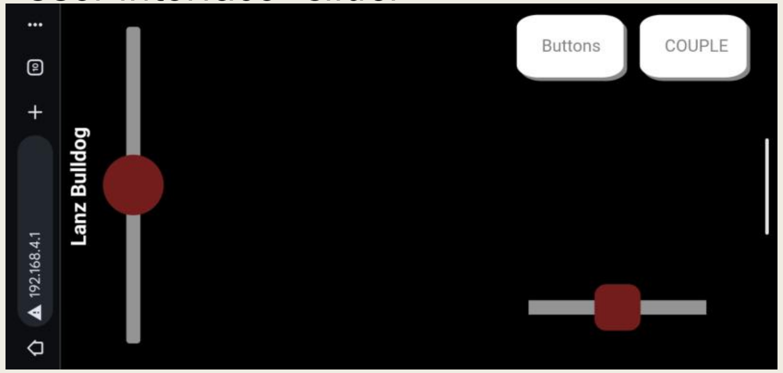
For one, there is the classic “button“ interface with just buttons for steering left or right and a slider for the throttle. Not the most accurate but works very well.

The other option is using a slider for steering as well by pressing the “Toggle UI“ button. You can cycle between both interfaces using this button. Please be aware that the slider for steering tends to “lag“ and feel unresponsive.
As of now, I haven‘t found a solution to this issue :(

User interface “*buttons*“

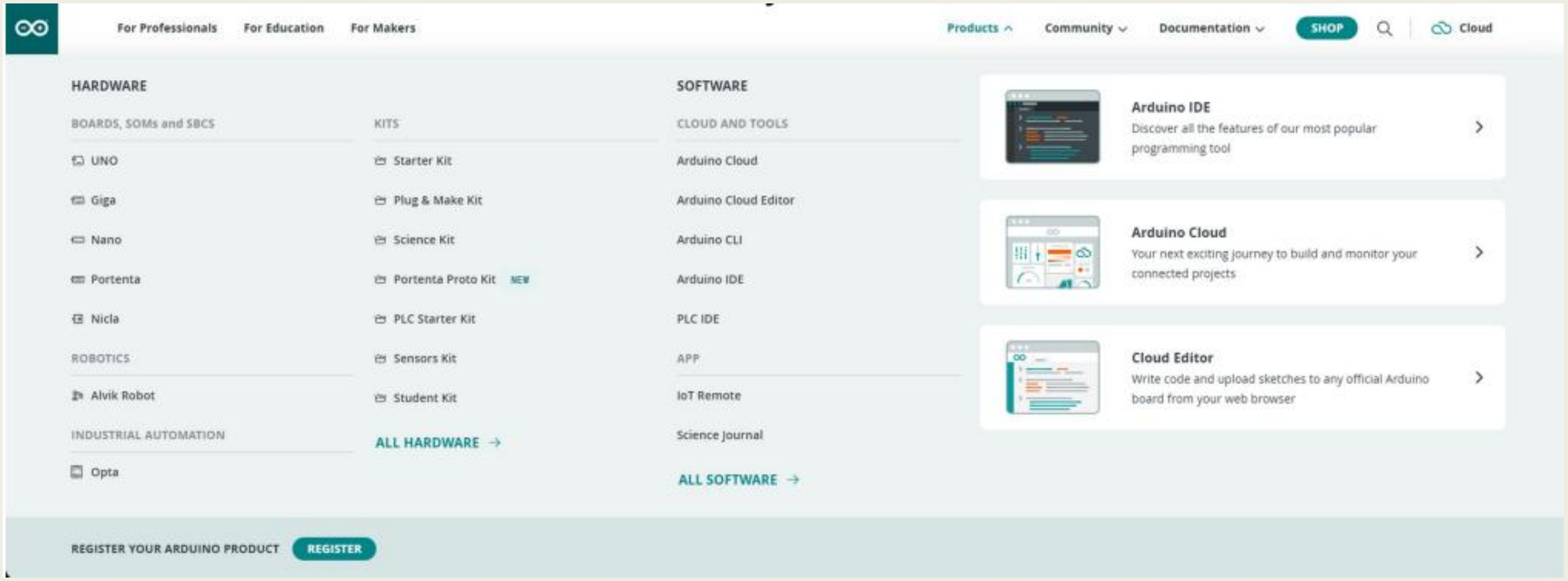


User interface “*slider*“



b) Uploading Code to the ESP32

- To upload the Code to the ESP32, you first need to download the Arduino IDE which you can get from arduino.cc. Under ‚Products‘ – ‚Arduino IDE‘ you find the download link.



Discover all the features of our most popular programming tool

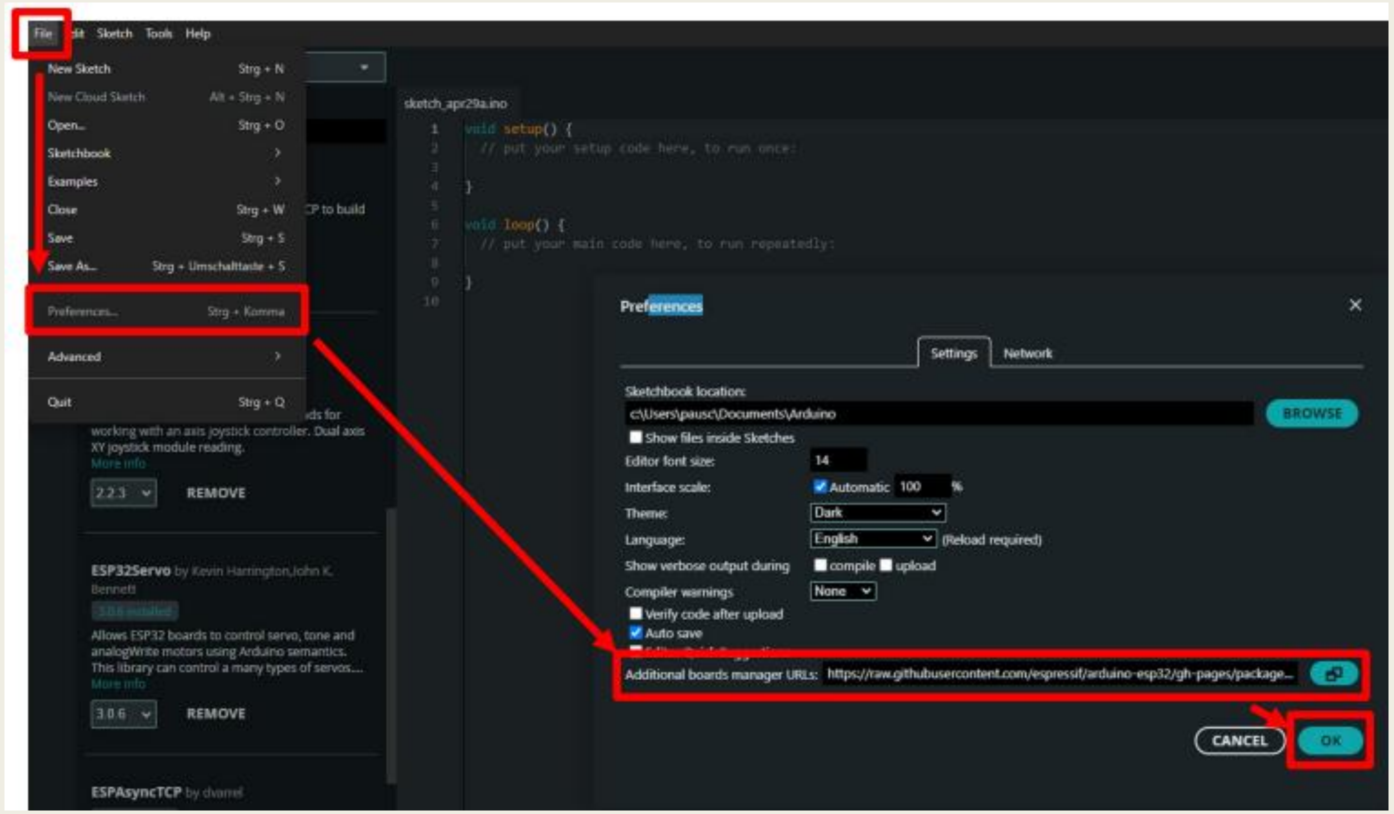
Your next exciting journey to build and monitor your connected projects

Write code and upload sketches to any official Arduino board from your web browser

- Official installation guide arduino.cc

Importing ESP32 Boards

- In the Arduino IDE under “File → Preferences“: Paste the link below in „Additional boards manager URL’s” -> Click “ok“



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

working with an axis joystick controller. Dual axis XY joystick module reading.
More info
2.2.3

Allows ESP32 boards to control servo, tone and analogWrite motors using Arduino semantics. This library can control a many types of servos....
More info
3.0.6

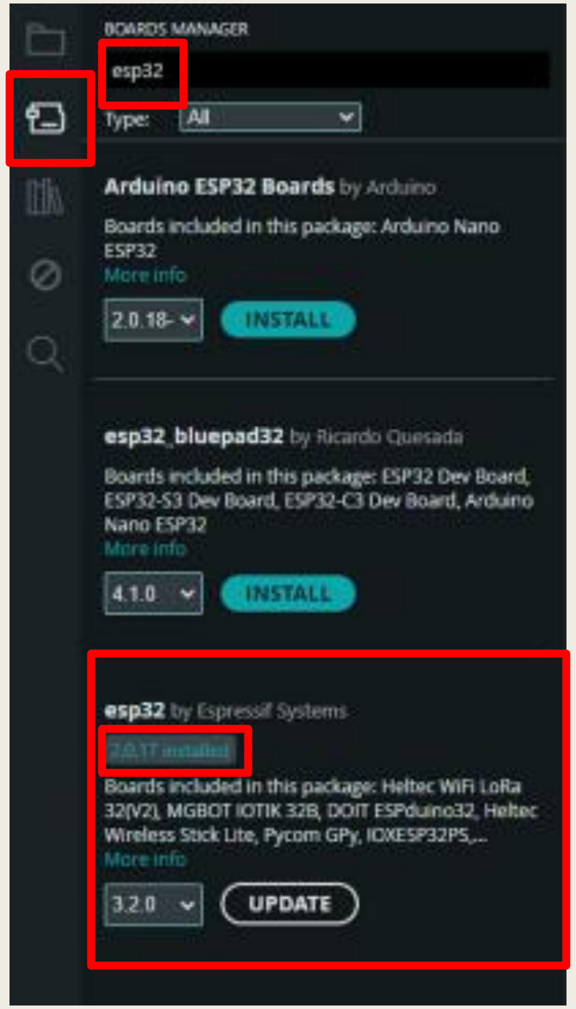
Sketchbook location: c:\Users\pausc\Documents\Arduino
[ ] Show files inside Sketches
Editor font size: 14
Interface scale: [✓] Automatic 100 %
Theme: Dark
Language: English (Reload required)
Show verbose output during [ ] compile [ ] upload
Compiler warnings: None
[ ] Verify code after upload
[✓] Auto save
Additional boards manager URLs: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package...

- https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json,https://raw.githubusercontent.com/ricardoquesada/esp32-arduino-libbuilder/master/bluepad32_files/package_esp32_bluepad32_index.json

Importing ESP32 Boards

- In the Arduino IDE locate the “boards manager” (2nd item on the side band)
- Search for “esp32“ by “Espressif Systems“
- Make sure you have Version 2.0.17 installed. Later versions of this board will not work as they break something in the servo library!

  If you have the wrong version just downgrade by selecting the correct version in the dropdown menu.



BOARDS MANAGER

esp32

Type: All

Arduino ESP32 Boards by Arduino

Boards included in this package: Arduino Nano ESP32

More info

2.0.18- INSTALL

esp32_bluepad32 by Ricardo Quesada

Boards included in this package: ESP32 Dev Board, ESP32-S3 Dev Board, ESP32-C3 Dev Board, Arduino Nano ESP32

More info

4.1.0 INSTALL

esp32 by Espressif Systems

2.0.17 installed

Boards included in this package: Heltec WiFi LoRa 32(V2), MGBOT IOTIK 32B, DOIT ESPduino32, Heltec Wireless Stick Lite, Pycom GPy, IOXESP32PS,...

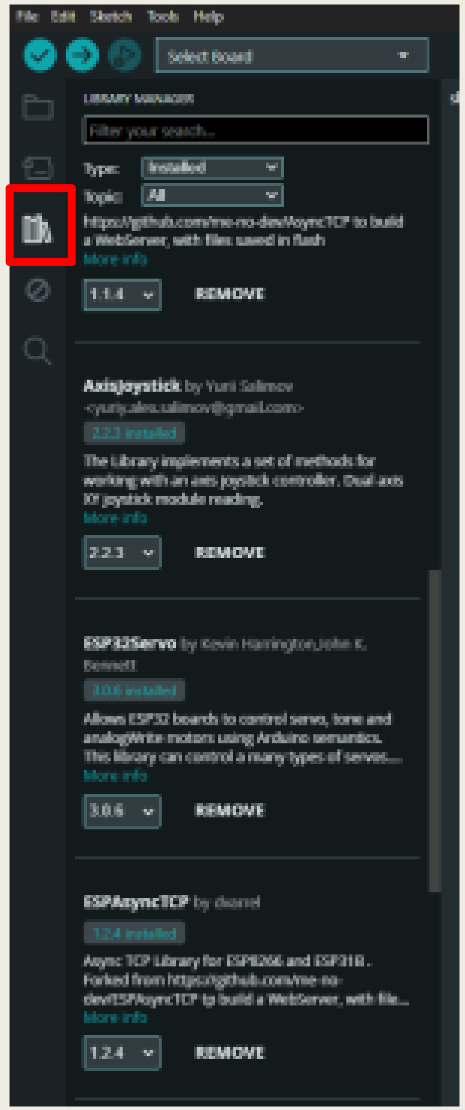
More info

3.2.0 UPDATE

Installing Libraries

- Open the library manager either by clicking the middle button on the left band or via ‚Sketch→Include library→Manage Libraries‘
- Search for and then install the following libraries:

  -“ESPAsyncWebSrv“ by “dvarrel“
  -“AsyncTCP.h“ by “dvarrel“
  -“ESPAsyncTCP.h“ by “dvarrel“



Type: Installed
Topic: All

https://github.com/me-no-dev/AsyncTCP to build a WebServer, with files saved in flash
More info
1.1.4 REMOVE

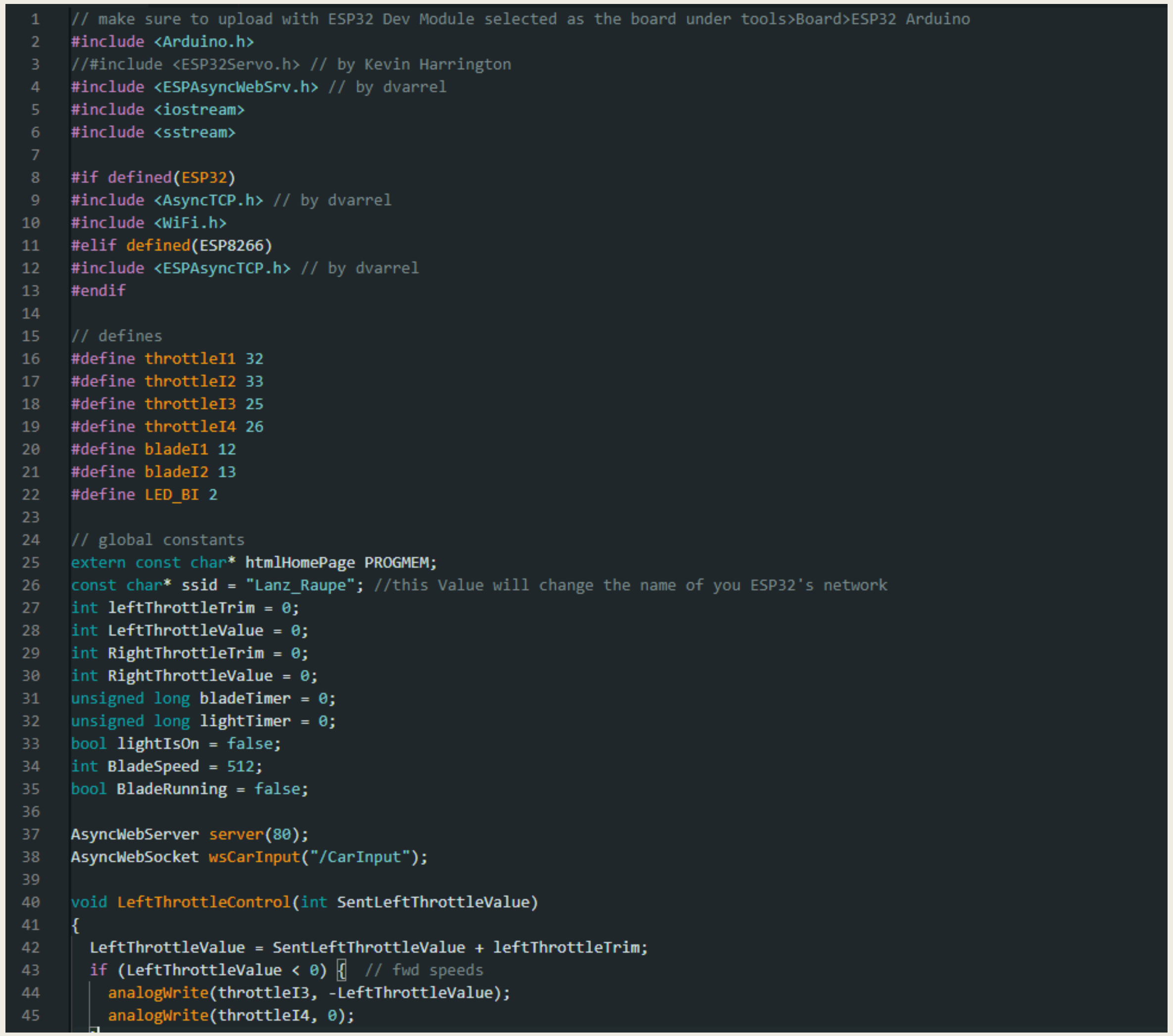
AxisJoystick by Yuri Salimov <yuriy.alex.salimov@gmail.com>
2.2.3 installed
The Library implements a set of methods for working with an axis joystick controller. Dual axis XY joystick module reading.
More info
2.2.3 REMOVE

ESP32Servo by Kevin Harrington,John K. Bennett
3.0.6 installed
Allows ESP32 boards to control servo, tone and analogWrite motors using Arduino semantics. This library can control a many types of servos....
More info
3.0.6 REMOVE

ESPAsyncTCP by dvarrel
1.2.4 installed
Async TCP Library for ESP8266 and ESP31B . Forked from https://github.com/me-no-dev/ESPAsyncTCP tp build a WebServer, with file...
More info
1.2.4 REMOVE

Opening the Code

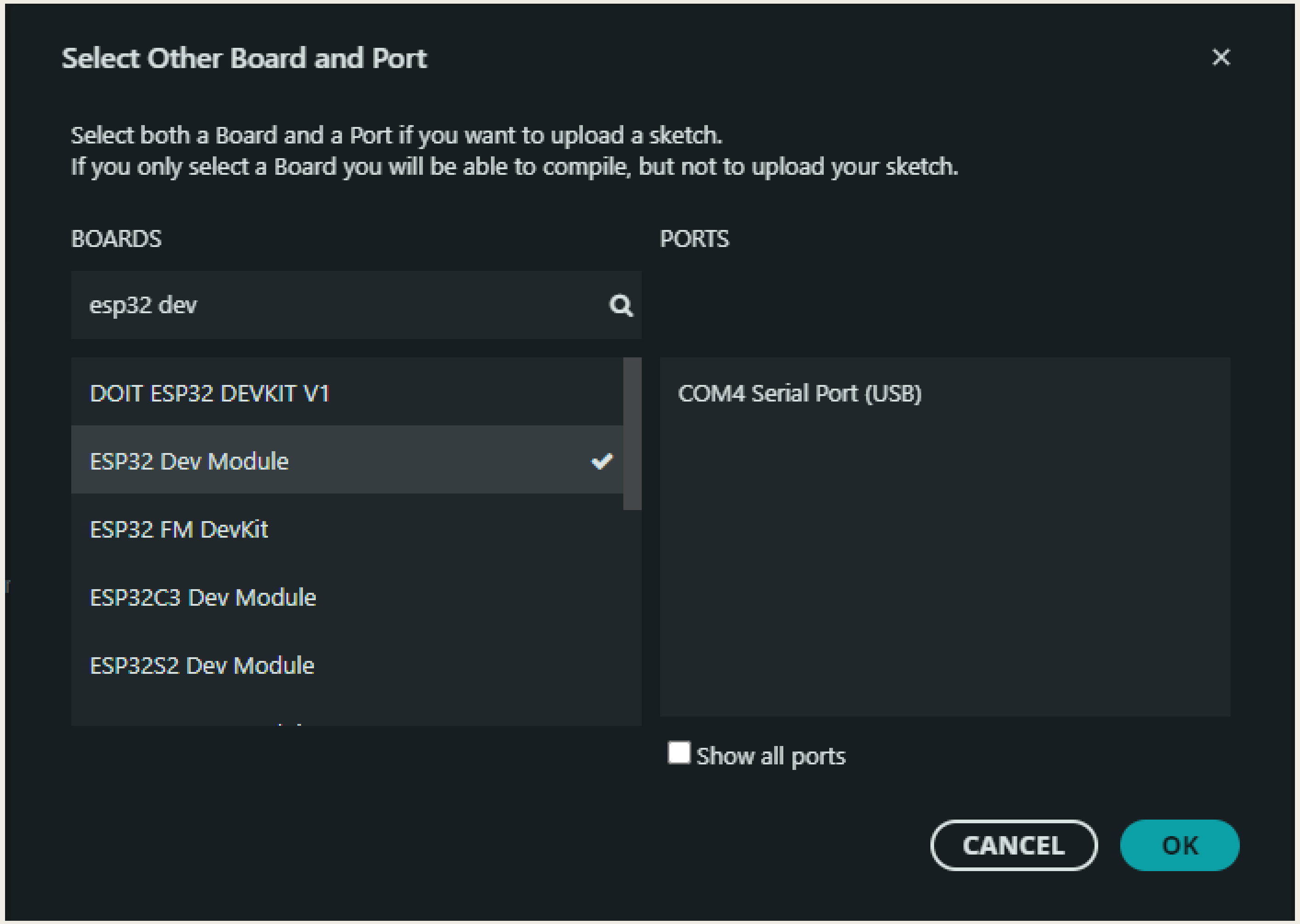
- After downloading and extraction the code.zip file you can open “SUB_LanzBulldog.ino”
  with the Arduino IDE.



```
// make sure to upload with ESP32 Dev Module selected as the board under tools>Board>ESP32 Arduino
#include <Arduino.h>
//#include <ESP32Servo.h> // by Kevin Harrington
#include <ESPAsyncWebSrv.h> // by dvarrel
#include <iostream>
#include <sstream>

#if defined(ESP32)
#include <AsyncTCP.h> // by dvarrel
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESPAsyncTCP.h> // by dvarrel
#endif

// defines
#define throttleI1 32
#define throttleI2 33
#define throttleI3 25
#define throttleI4 26
#define bladeI1 12
#define bladeI2 13
#define LED_BI 2

// global constants
extern const char* htmlHomePage PROGMEM;
const char* ssid = "Lanz_Raupe"; //this Value will change the name of you ESP32's network
int leftThrottleTrim = 0;
int LeftThrottleValue = 0;
int RightThrottleTrim = 0;
int RightThrottleValue = 0;
unsigned long bladeTimer = 0;
unsigned long lightTimer = 0;
bool lightIsOn = false;
int BladeSpeed = 512;
bool BladeRunning = false;

AsyncWebServer server(80);
AsyncWebSocket wsCarInput("/CarInput");

void LeftThrottleControl(int SentLeftThrottleValue)
{
  LeftThrottleValue = SentLeftThrottleValue + leftThrottleTrim;
  if (LeftThrottleValue < 0) {  // fwd speeds
    analogWrite(throttleI3, -LeftThrottleValue);
    analogWrite(throttleI4, 0);
```

Selecting the board and port

- From “Select Board“ search for and select “ESP32 Dev Module“
- For the port you can plug the ESP into your computer via USB and check which port pops up.
- Select that port (in my case COM4, however this will most likely differ for your install)
- Click “ok“


Select Other Board and Port

Select both a Board and a Port if you want to upload a sketch.
If you only select a Board you will be able to compile, but not to upload your sketch.

BOARDS

esp32 dev

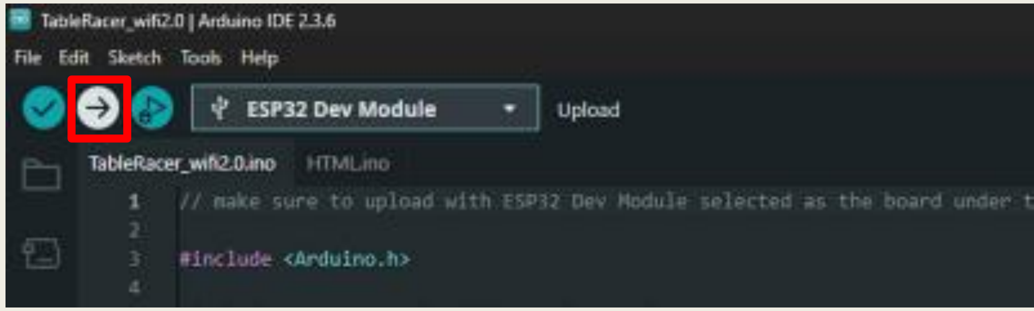
DOIT ESP32 DEVKIT V1
ESP32 Dev Module ✓
ESP32 FM DevKit
ESP32C3 Dev Module
ESP32S2 Dev Module

PORTS

COM4 Serial Port (USB)

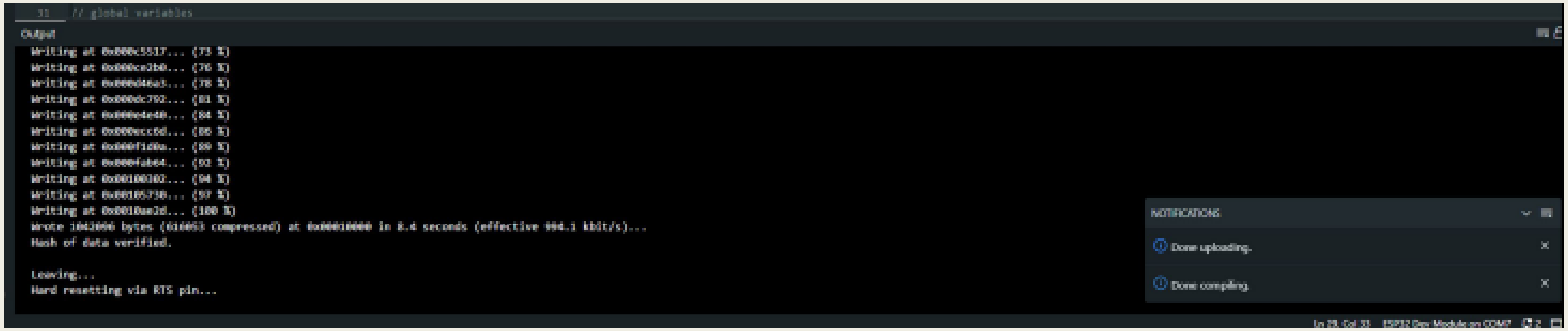
Uploading the code

- With the correct board and port selected and the ESP32 plugged in via USB you now can upload the code.


TableRacer_wifi2.0 | Arduino IDE 2.3.6
File Edit Sketch Tools Help
ESP32 Dev Module
Upload
TableRacer_wifi2.0.ino
HTML.ino

```
// make sure to upload with ESP32 Dev Module selected as the board under t

#include <Arduino.h>
```

- Just click the right arrow. The code will be uploaded, you can check the process in the console, which will appear on the bottom of your screen.



```
Writing at 0x000c5517... (73 %)
Writing at 0x000ce2b0... (76 %)
Writing at 0x000d46a3... (78 %)
Writing at 0x000dc792... (81 %)
Writing at 0x000e4e40... (84 %)
Writing at 0x000ecc6d... (86 %)
Writing at 0x000f1d0a... (89 %)
Writing at 0x000fab64... (92 %)
Writing at 0x00100302... (94 %)
Writing at 0x00105730... (97 %)
Writing at 0x0010ae2d... (100 %)
Wrote 1042096 bytes (616053 compressed) at 0x00010000 in 8.4 seconds (effective 994.1 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Verifying the upload

- To check wether the code was correctly uploaded and is working, leave the ESP32 plugged in and check your wifi-networks.
- After a few seconds there should be a network with the name specified in Code (or “Lanz Bulldog“ by default).
- Connect to the network and via a browser access http://192.168.4.1

  If there are any notifications like “No internet access“ just ignore/dismiss them. We don’t want to access the internet with our ESP. :)

  Now the interface should pop up.
  Note: this interface is made for mobile devices and does not work in desktop mode!

- When everything shows up – congratulations – you’ve successfully uploaded the code!

c) Ordering the PCB

Ordering the PCB is really simple on any site. I usually order mine from JLCPCB – the process is pretty much the same for any other suppliers.

I‘m not sponsored in any way by JLC, but I used EasyEDA to design the PCB, which makes it easy for me to order from them.

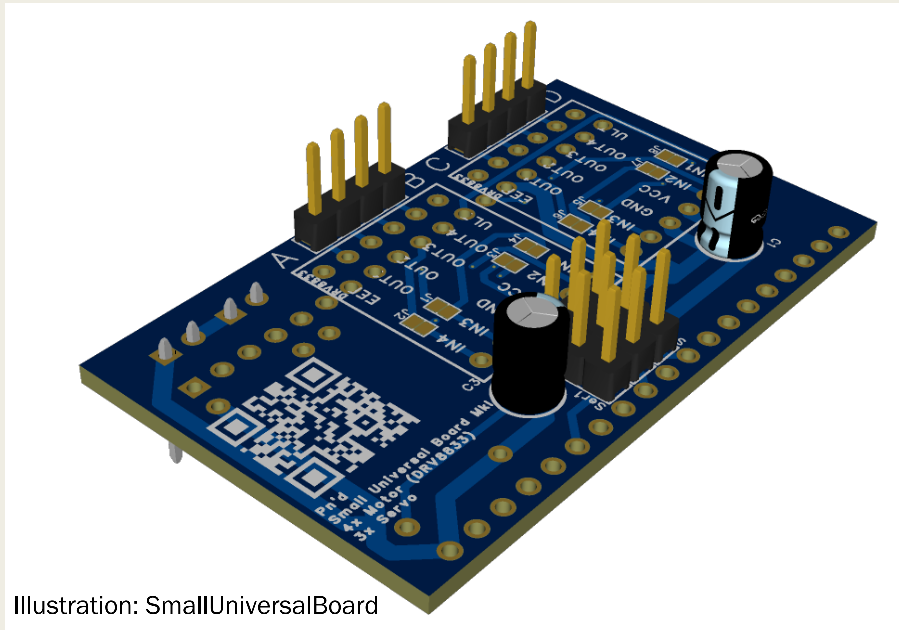
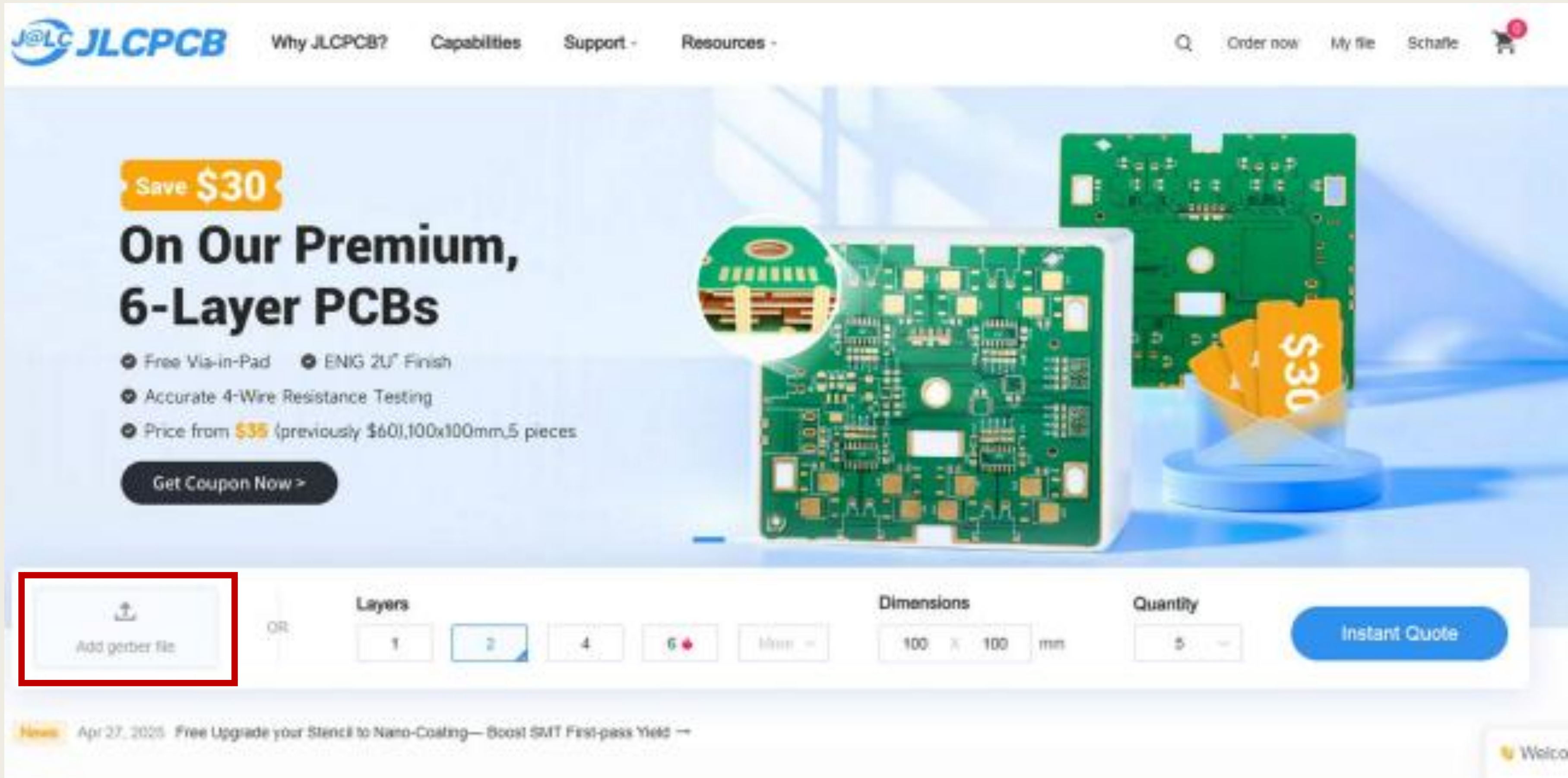


Illustration: SmallUniversalBoard

Uploading the Gerber Files

Drag the zipped PCB.zip folder to ‚Add gerber file'

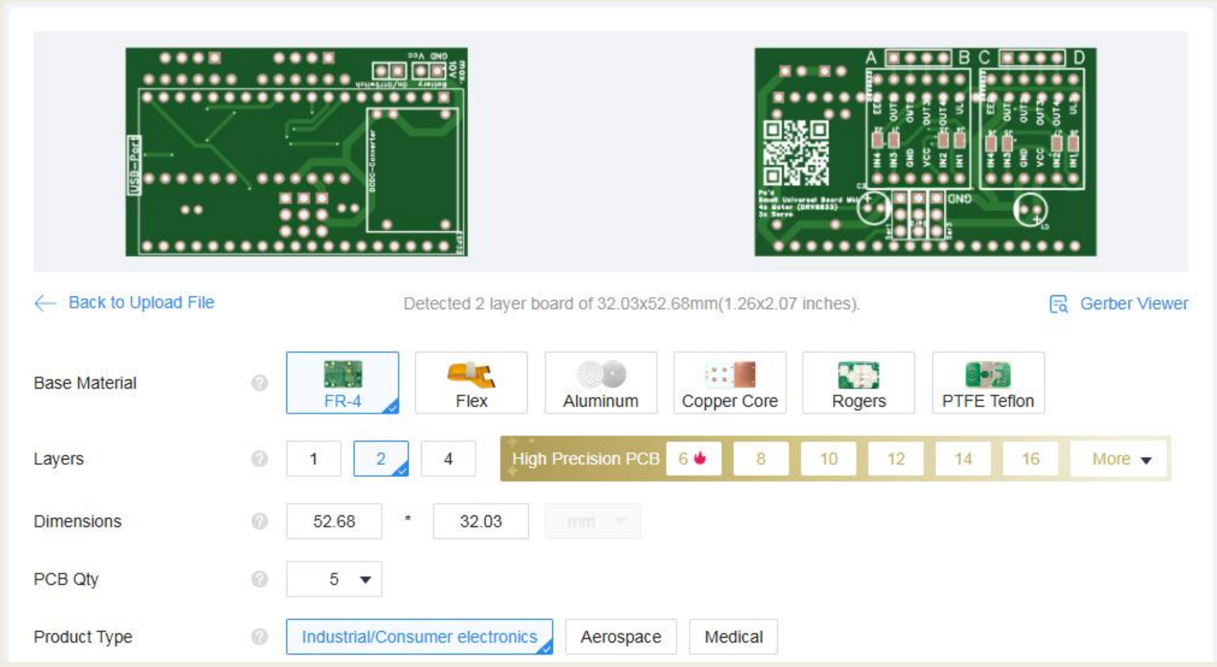

JLCPCB
Why JLCPCB?
Capabilities
Support
Resources
Order now
My file
Save $30
On Our Premium, 6-Layer PCBs
Free Via-in-Pad
ENIG 2U" Finish
Accurate 4-Wire Resistance Testing
Price from $35 (previously $60),100x100mm,5 pieces
Get Coupon Now >
Add gerber file
OR
Layers
1
2
4
6
Dimensions
100
100
mm
Quantity
5
Instant Quote
Apr 27, 2025 Free Upgrade your Stencil to Nano-Coating— Boost SMT First-pass Yield →

Checking the files

After uploading you can see the actual PCB. You can leave all defaults. If you wish, you can change the PCB color, which might increase lead time but usually doesn‘t cost extra.

This PCB should have 2 layers and be just under 53mm x 33 mm

After this, you can save to cart and check out.



Back to Upload File

Detected 2 layer board of 32.03x52.68mm(1.26x2.07 inches).

Gerber Viewer

Base Material	[✓ FR-4] Flex Aluminum Copper Core Rogers PTFE Teflon
Layers	1 [✓ 2] 4 High Precision PCB 6 8 10 12 14 16 More
Dimensions	52.68 * 32.03 mm
PCB Qty	5
Product Type	[✓ Industrial/Consumer electronics] Aerospace Medical

d) Electronics assembly

Required parts from BOM:

1x PCB
1x ESP32
1x 5V DC-DC converter
2x Capacitor 100uF min. 16V
1x DRV8833 motor driver module
1x 4 pin header female
2x 6 pin header female
2x 19 pin header female
1x 4 pin header male
1x 2 pin JST

optional:
(1x 4 pin header male + Dupond connectors, if you want your motors to be crimped with Dupond connectors)

Caution:

Connecting the components wrong, be it polarity or missmatching pins, can and probably will lead to damaged components.

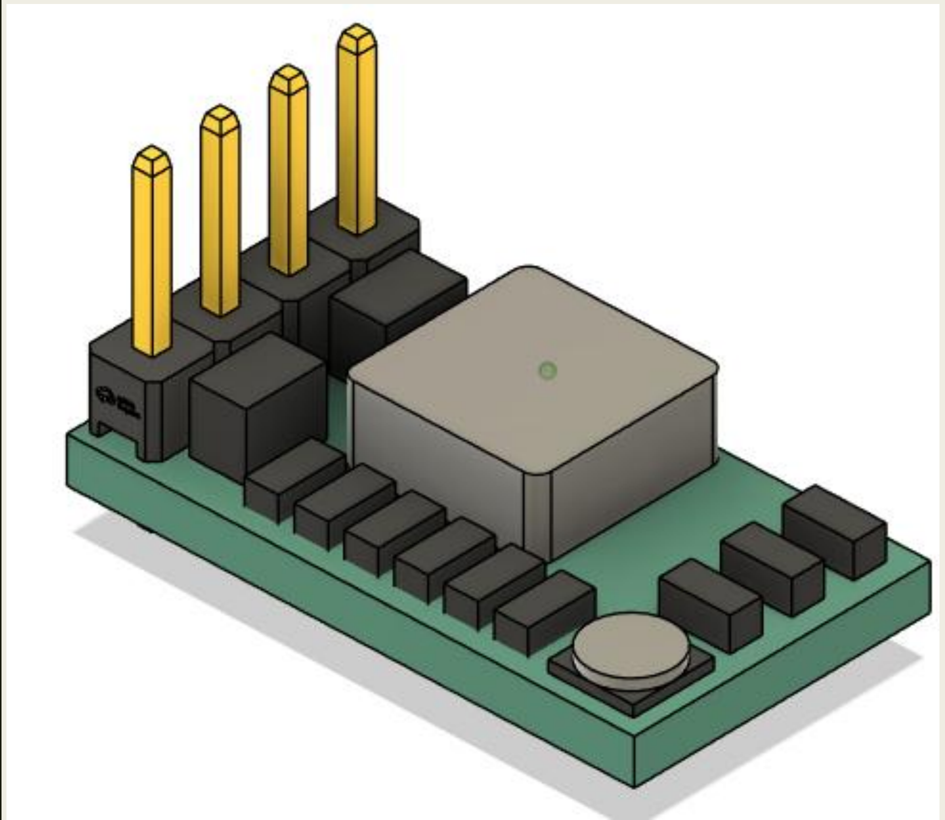
Take your time during those steps :)

Be carefull when working on any sort of electronics – always make sure there is no power to it when you handle the electronics.

[DC-DC-Converter]

[Step 1]

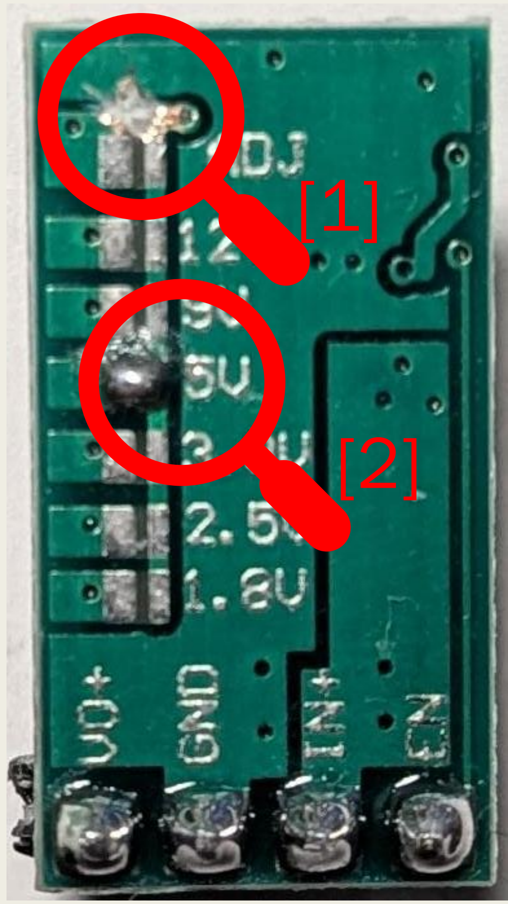
Prepare the *DCDC-converter* by soldering a 4 pin male header to the component side.



To set the desired output voltage according to the manufacturers instructions you have to follow two steps:

Step 1: “Use a knife to cut off the circuit in the red frame (along the black line)“. Just use a small sharp knife to cut the black line, until there is no electrical connection between the ends of the black line.

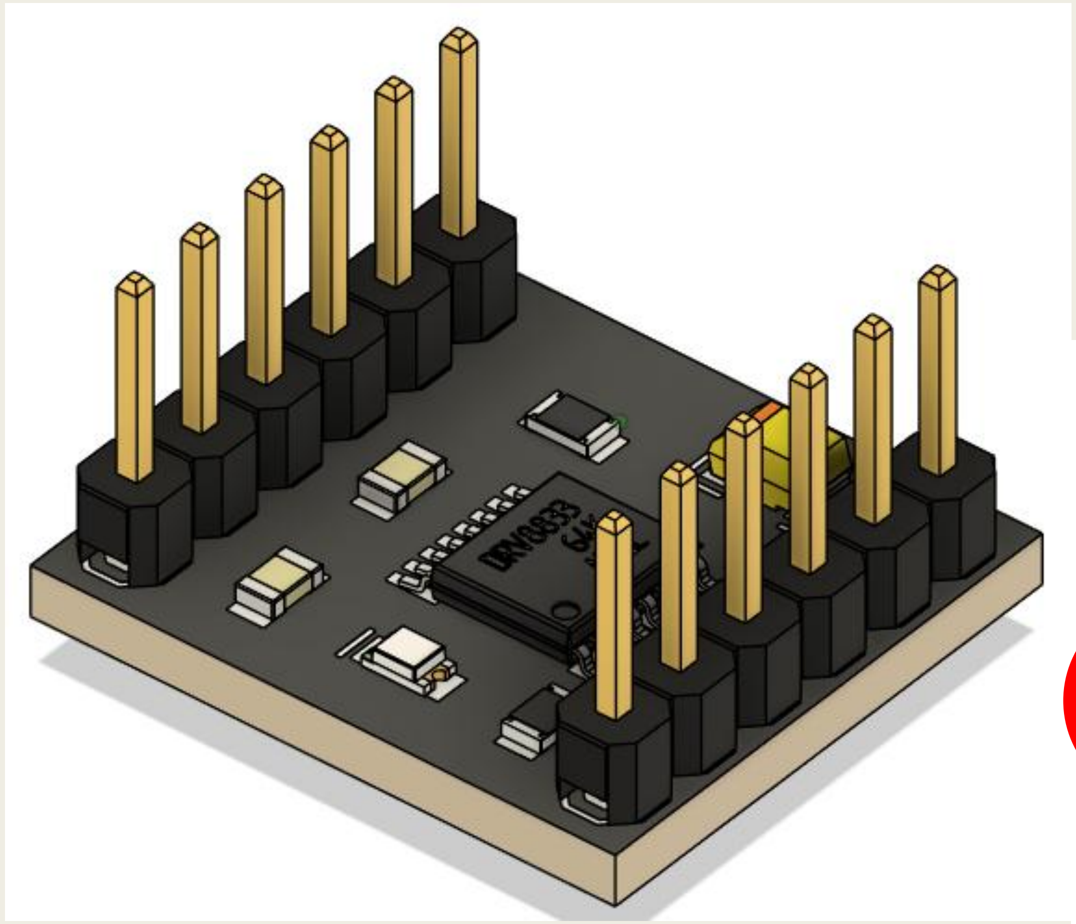
Step 2: “Solder the pad with solder at the voltage you need“. In our case we need to solder at the 5V mark

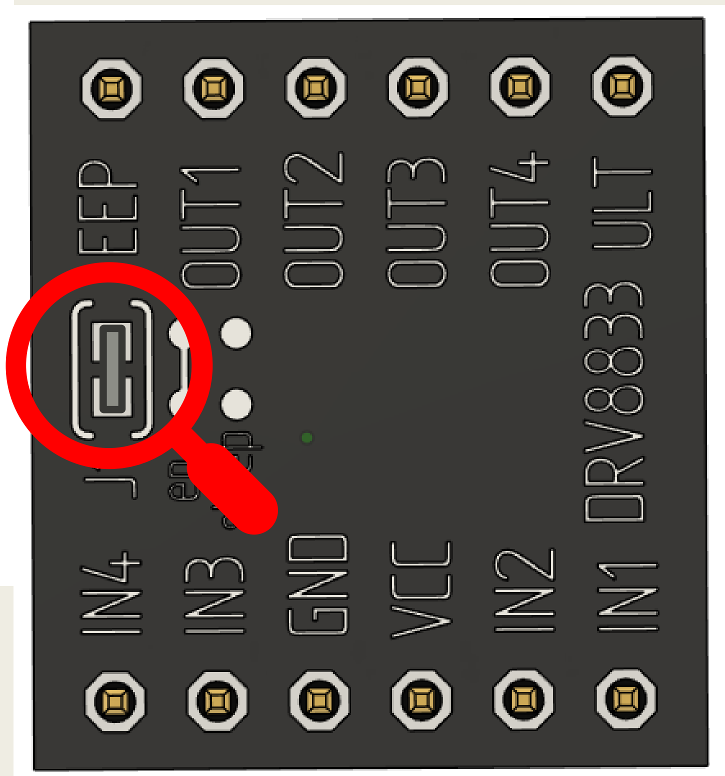


[DRV8833]

Prepare the *DRV8833* by soldering the included 6 pin headers facing the component side.

On the underside jump the solder jumper.



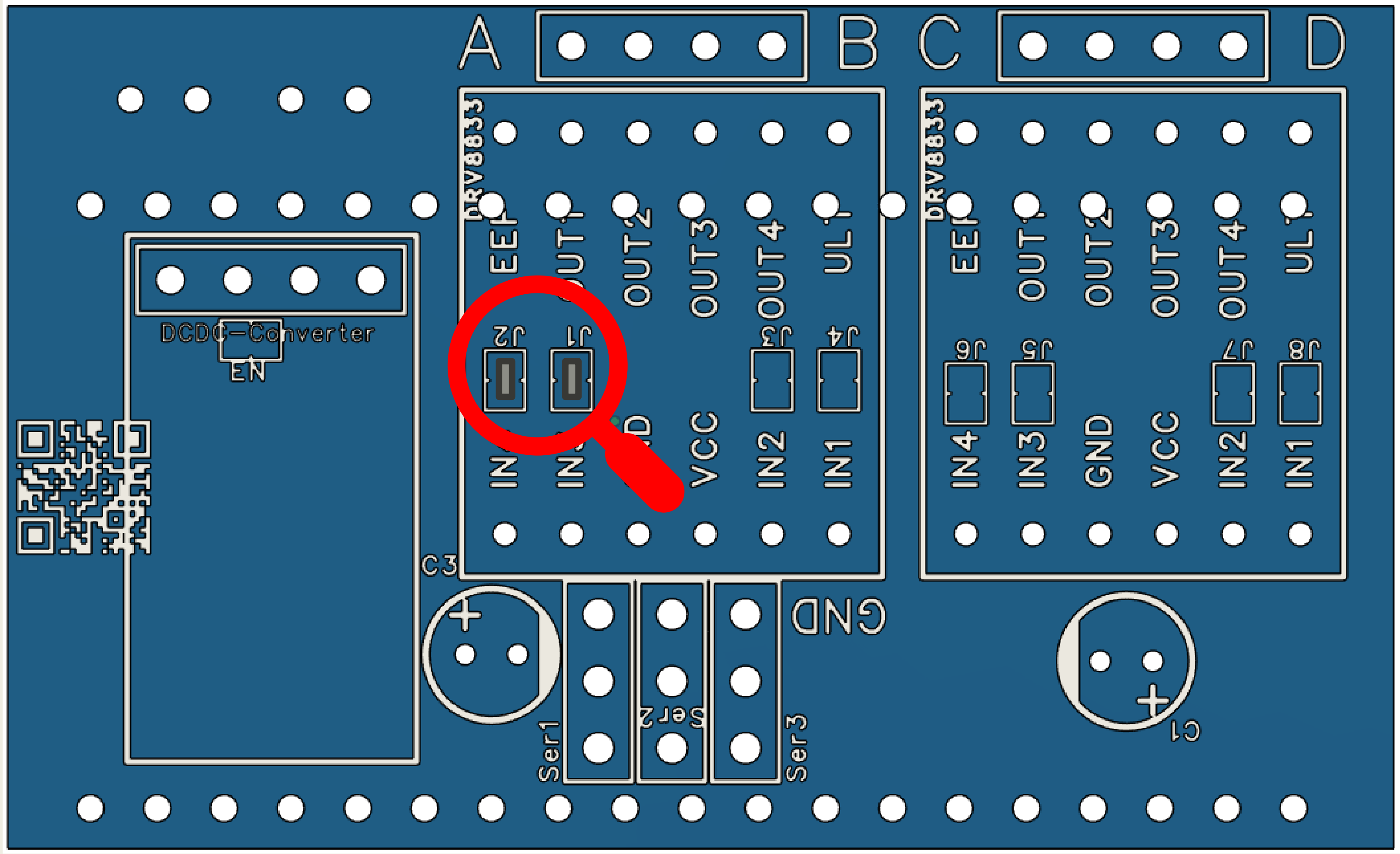


[Main PCB]

[Step 3]

Before soldering any components to the *PCB* make sure to jump the jumpers J1 and J2 using solder. The gap between the two pads is small enough for solder to flow over it.

Just be carefull not to accidentally short any two jumpers together!

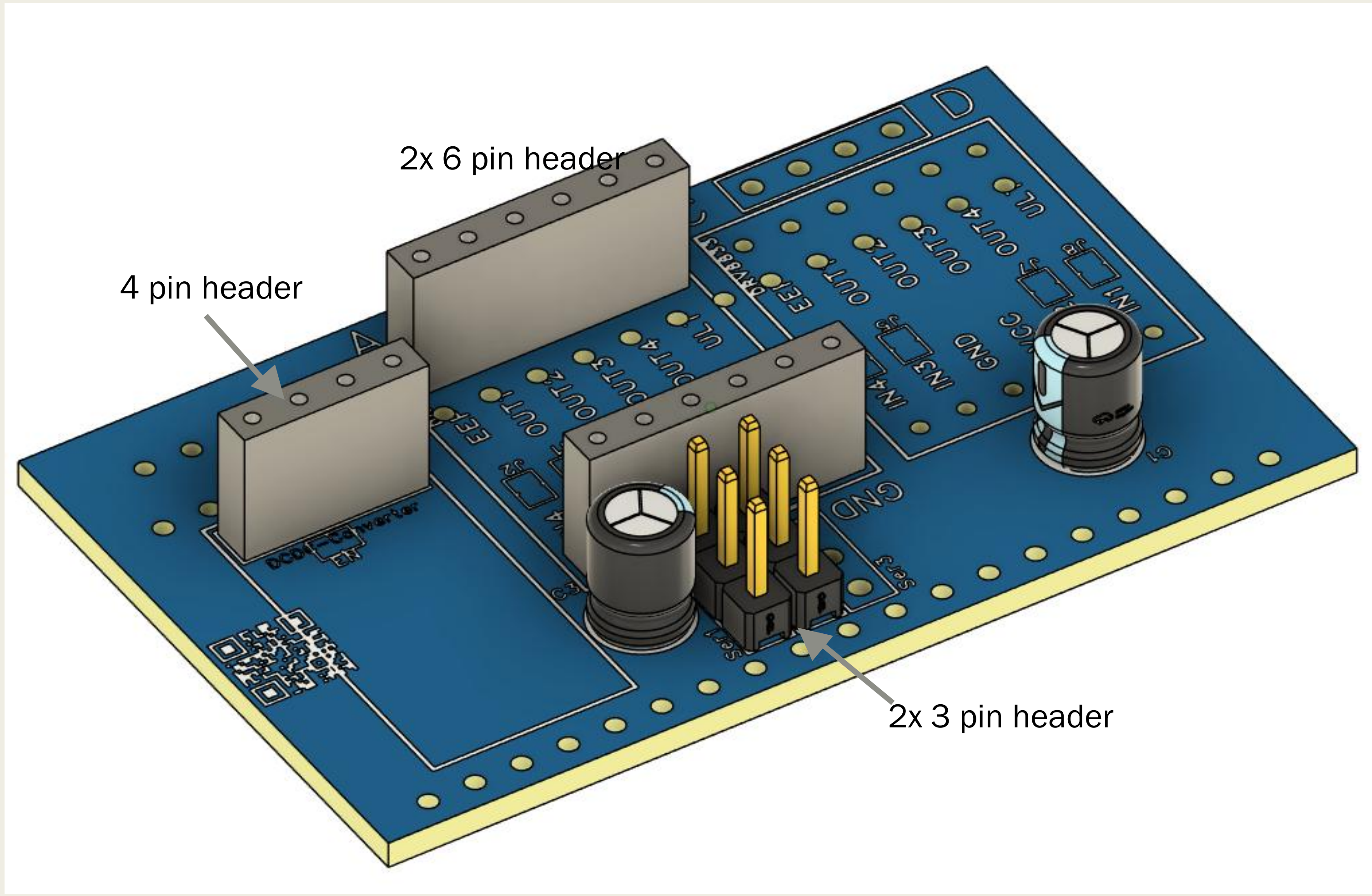


[Main PCB]

[Step 4]

Solder the *capacitors,* two 3 *pin male headers* and *two 6 pin female headers* to the underside of the *PCB*.

For the *capacitors* be extra carefull not to reverse the polarity!



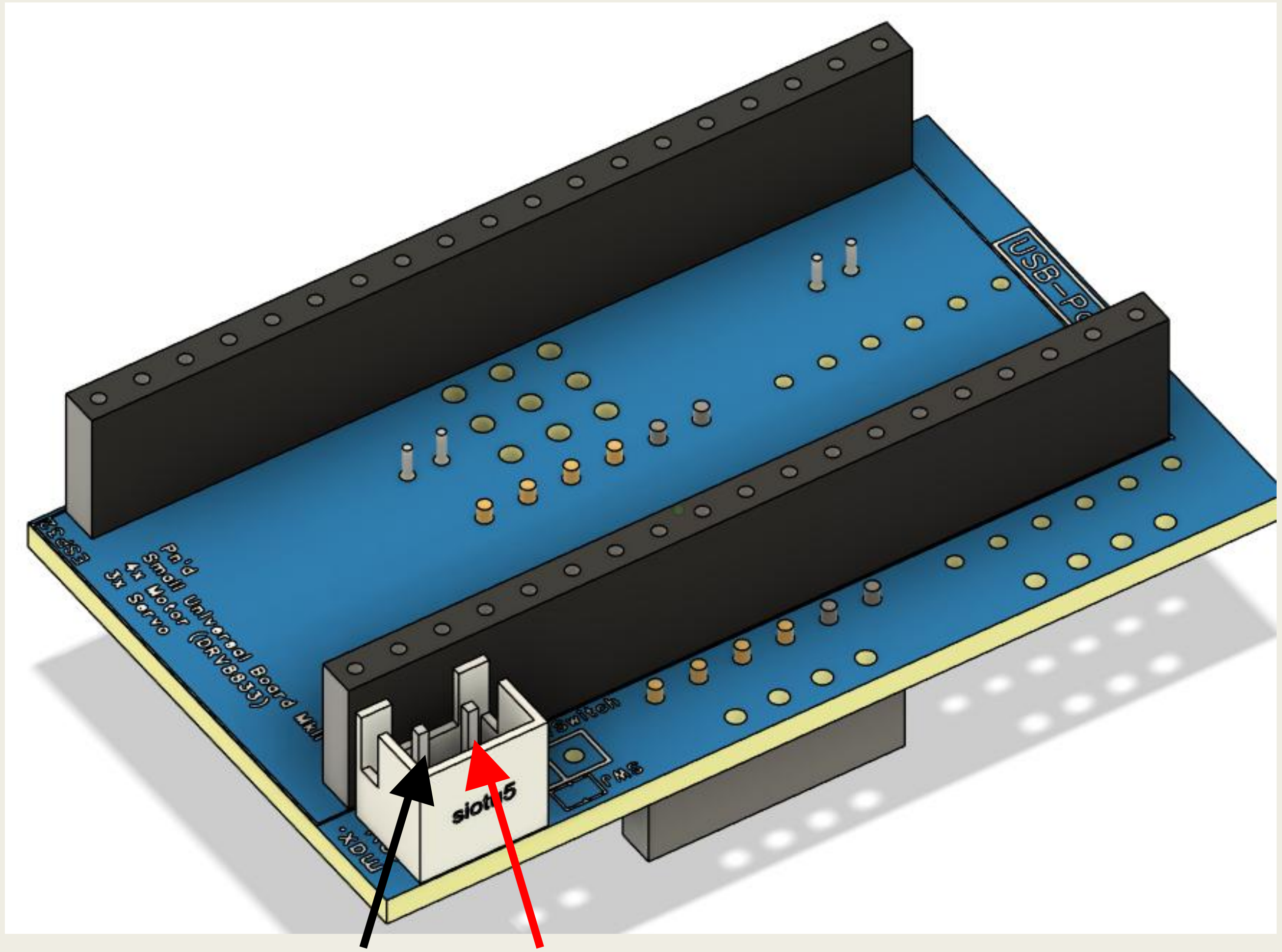
[Main PCB]

[Step 5]

Solder the *19 pin headers* to the top side of the *main PCB*.

Next, solder the *battery plug* (in my case a JST-Plug) to the *PCB*. Make sure to match the orientation to that marked on the *PCB*. A battery thats connected the wrong way will damage components, so take your time here!

If you choose not to use the switch later on, you can jump the solder jumper "SWJ" at this point



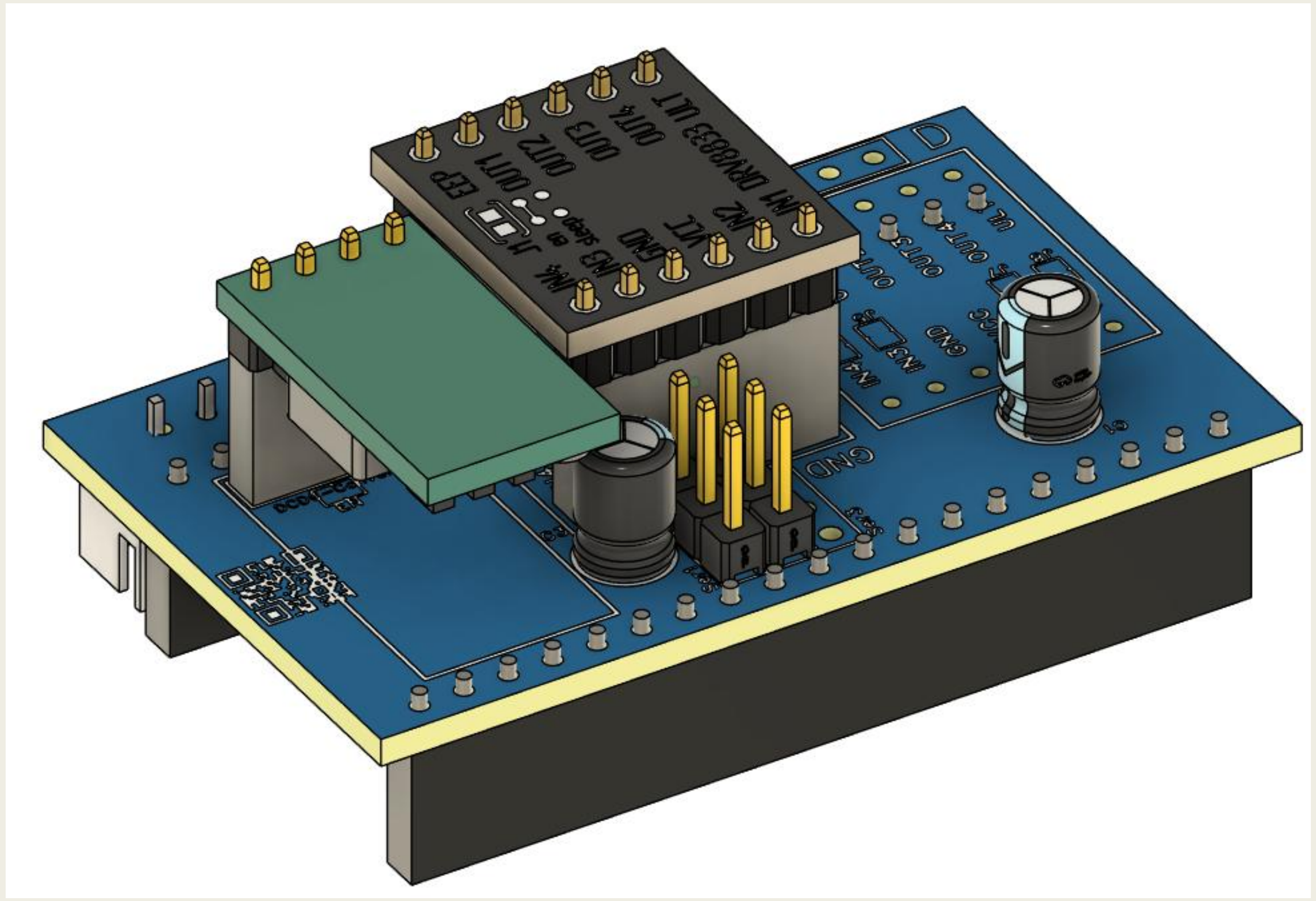
Battery has to match this polarity!

[Main PCB]

[Step 6]

Plug in the *DRV8833‘s* and *DCDC-converter* you prepared in the first steps.

For the *DRV8833‘s* make sure to match the orientation to that marked in the *PCB*.



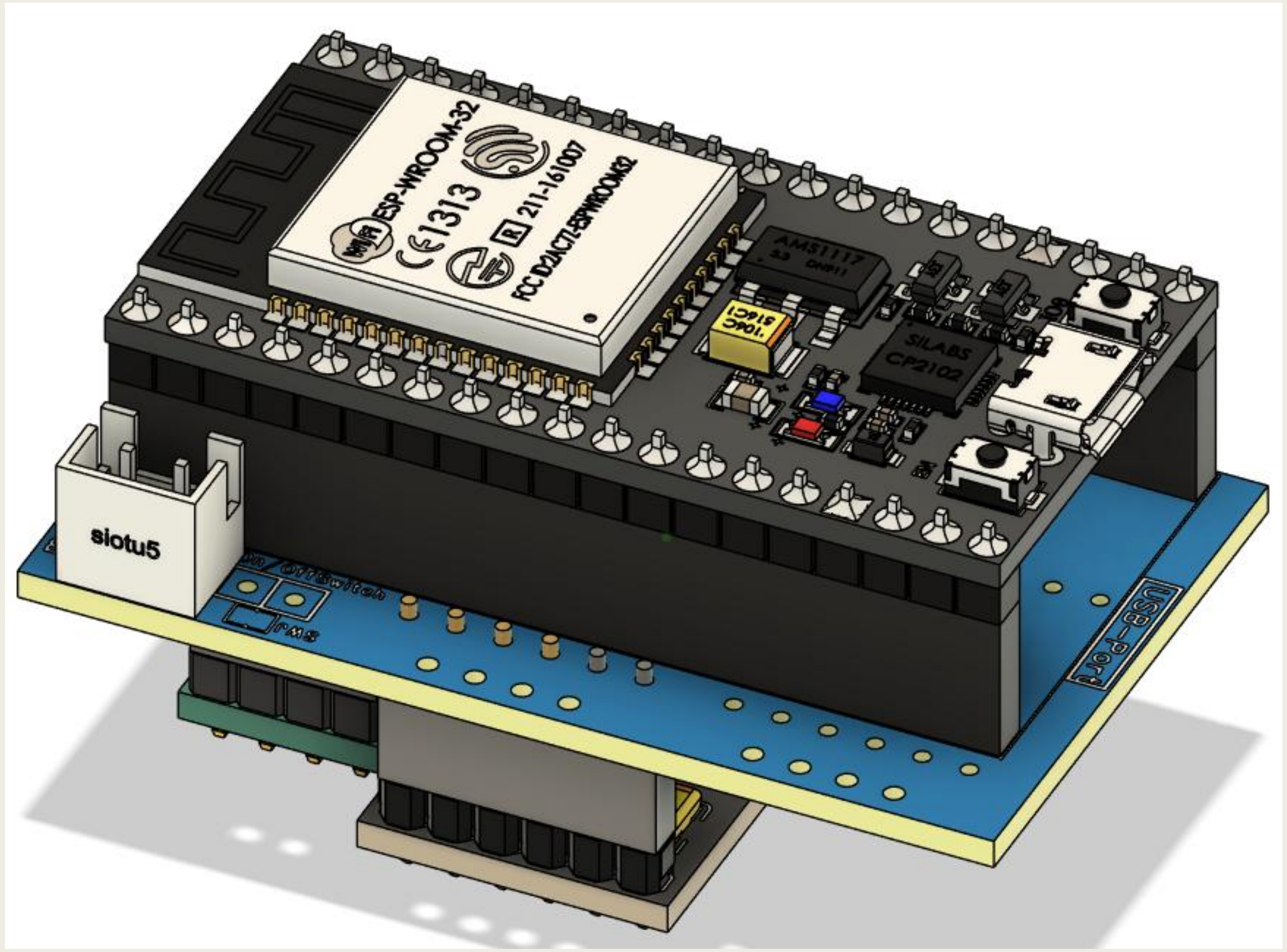
[Main PCB]

[Step 7]

Plug in the *ESP32*

The USB-port should be matching the mark on the *PCB*

If you want to upload code to the *ESP32* I recommend pulling it out of the *PCB* and uploading the new code, when it isn‘t connected to any other PCBs.

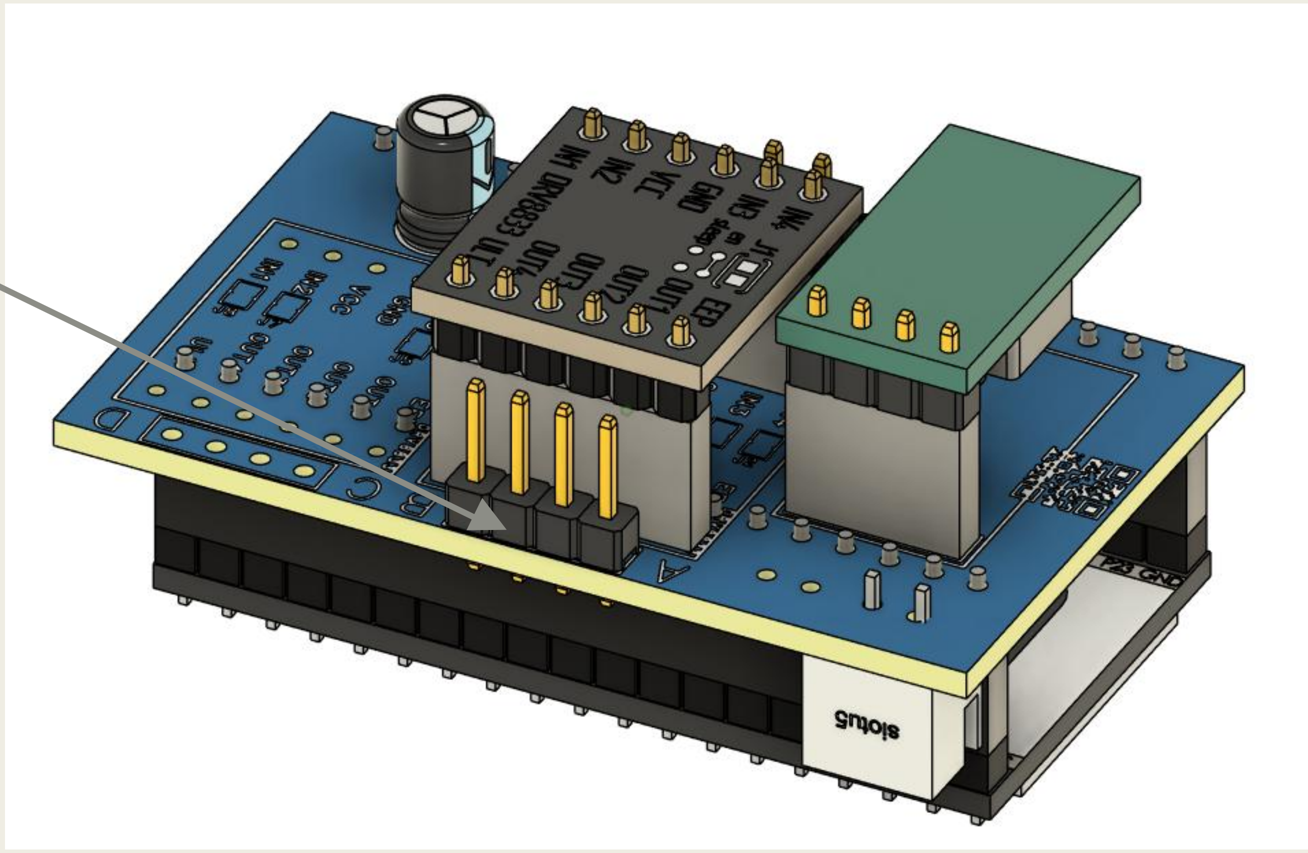


[Wires]

Optional:

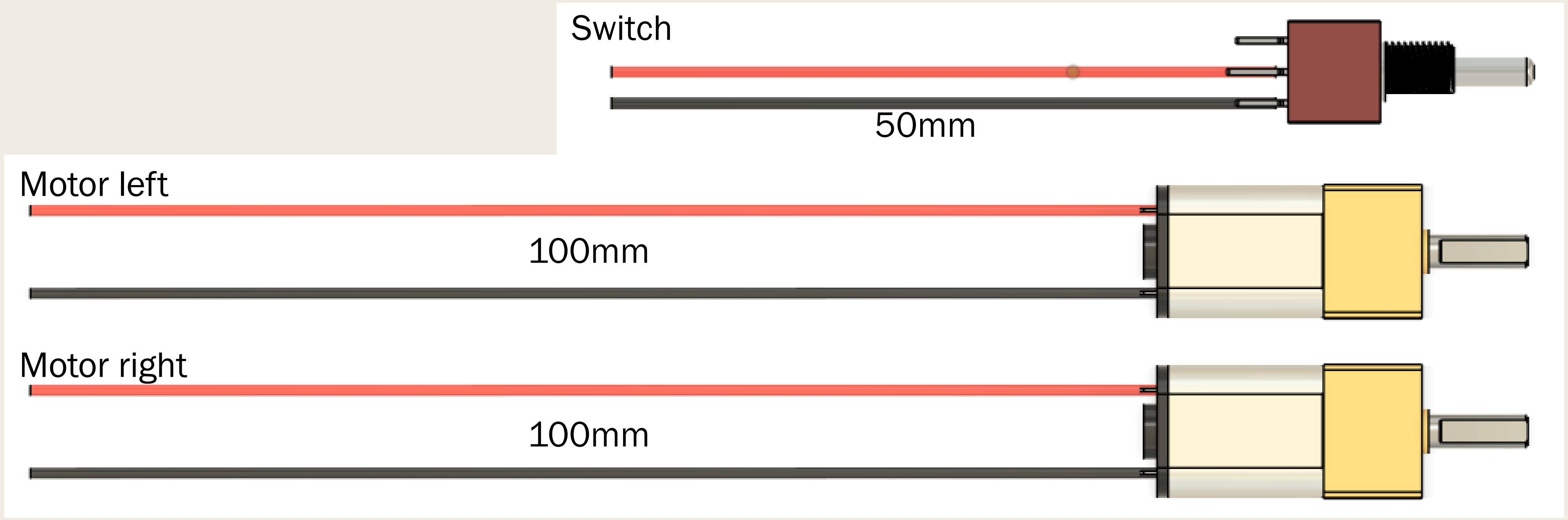
1x *4 pin header male*

The motor cables can either be soldered directly to the board or be crimped with Dupond connectors and use the pin headers to connect the plugs

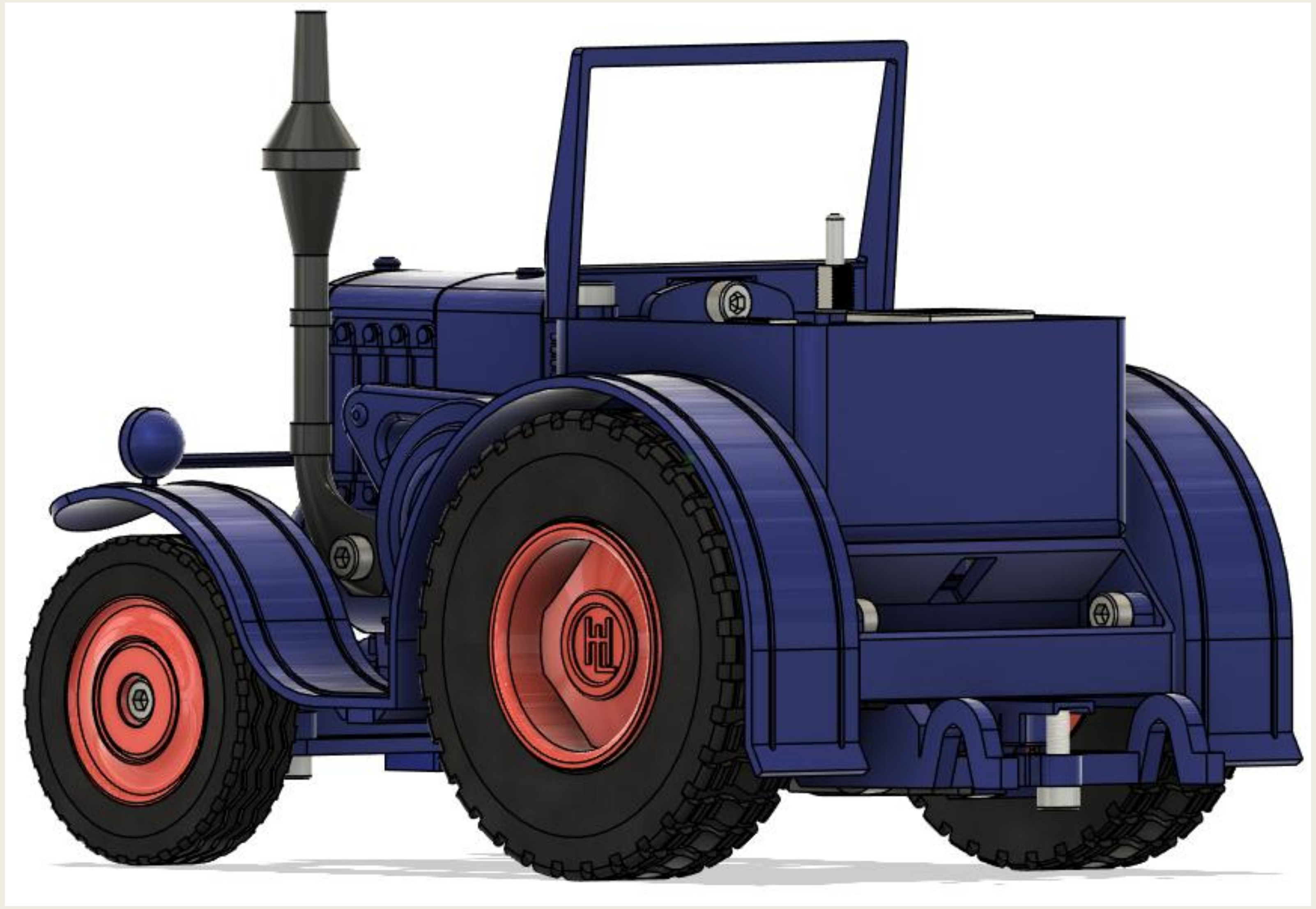


[Wires]

Solder the 100mm wires to the motor matching the polarity marked on the back of the motor to the colors of the wires (red – “+“; Black – “-“)

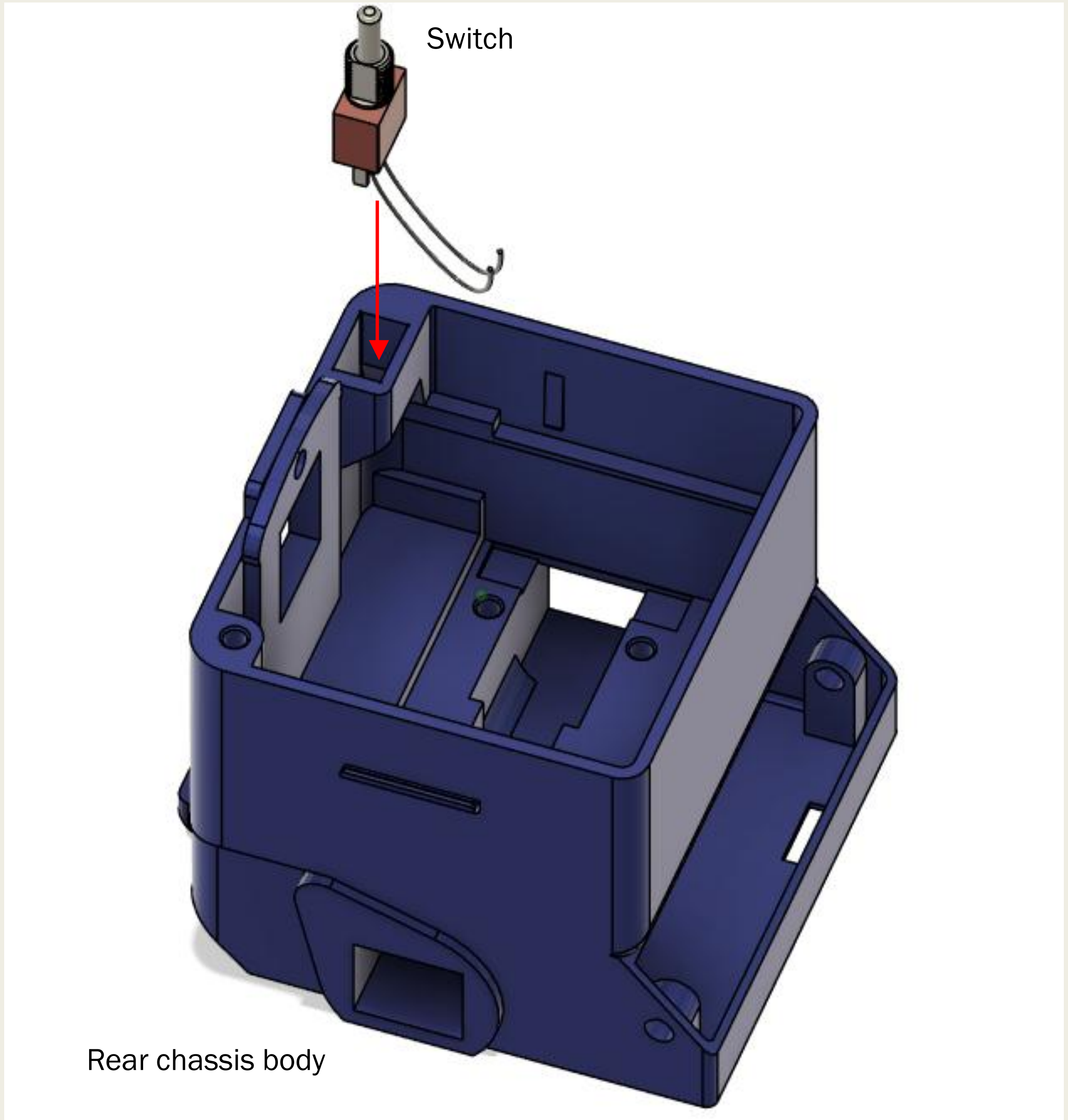


e) Mechanical assembly



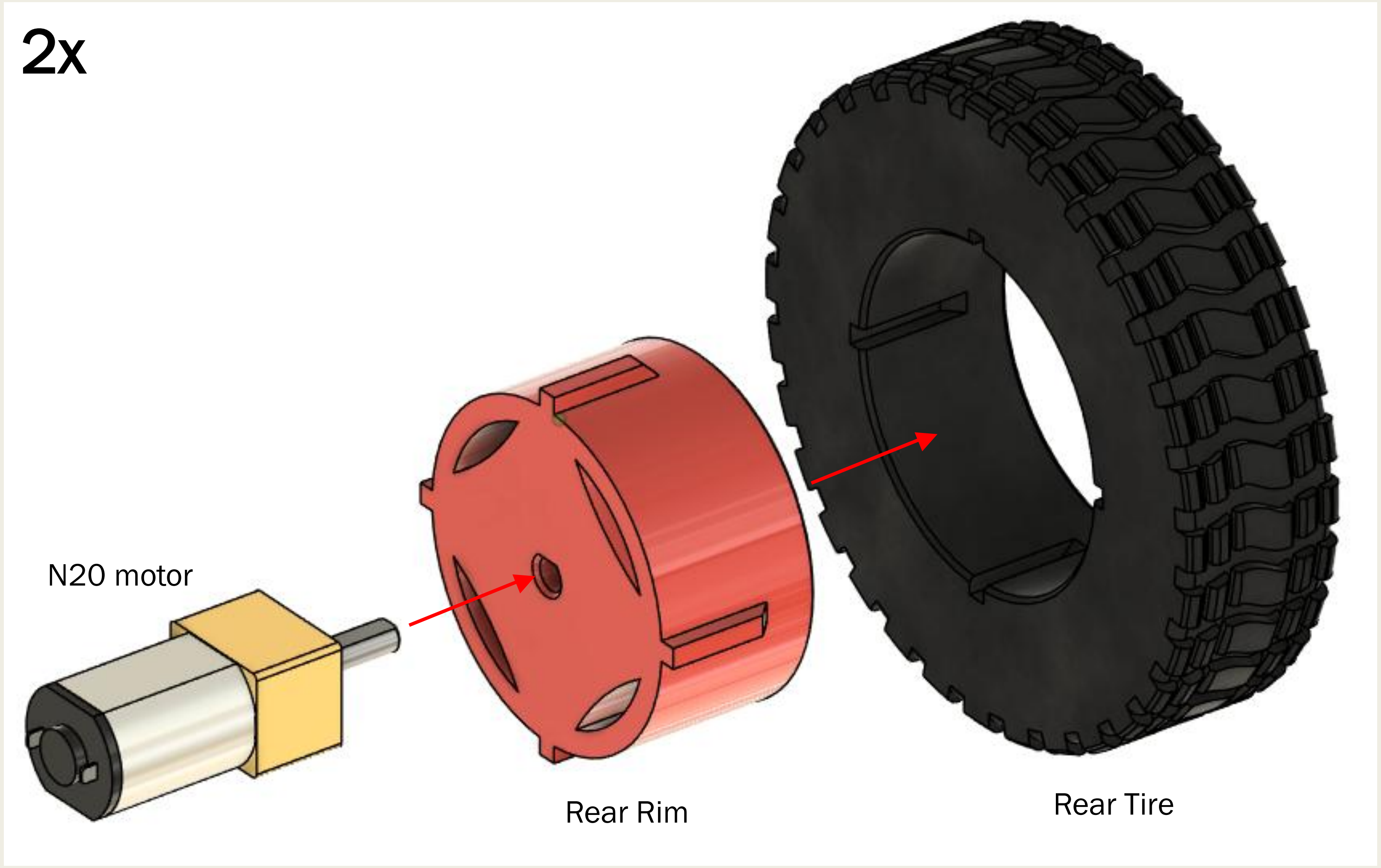
[Rear chassis]

[Step 1]



[Rear chassis]

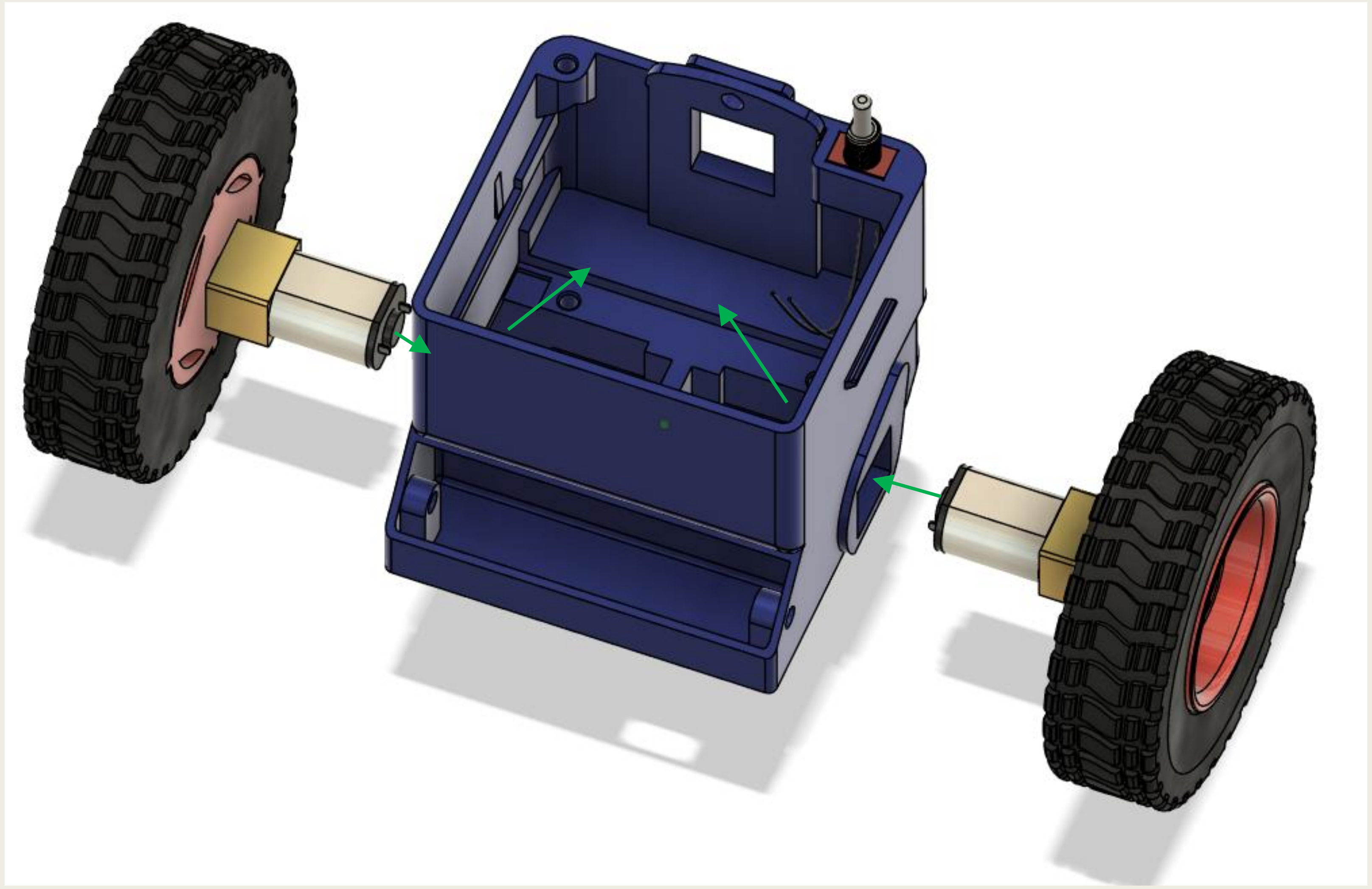
[Step 2]



The fit should be pretty tight. Use a flat surface to push the motor shaft into the Rim

[Rear chassis]

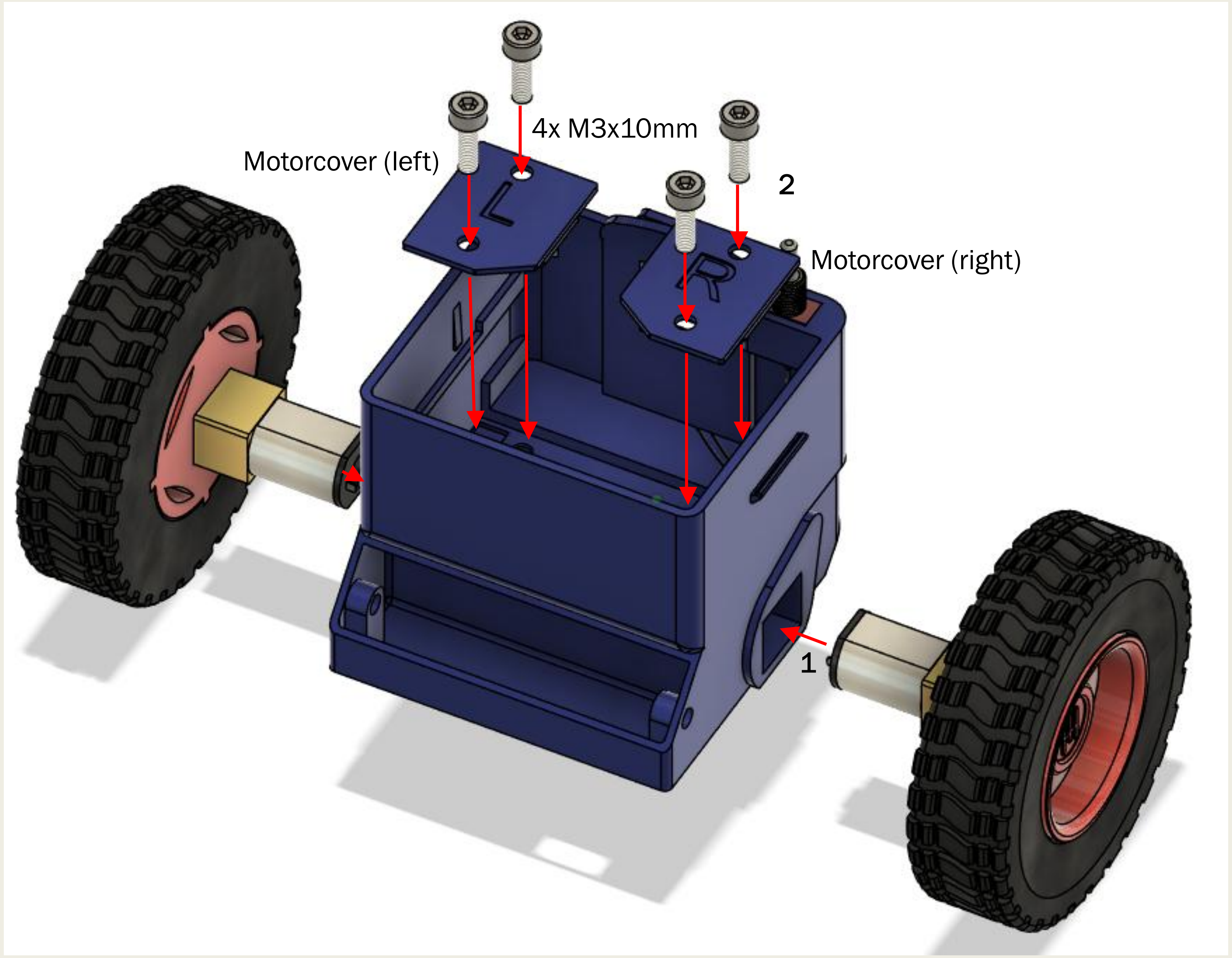
[Step 3]



Feed the prepared motor wires through the body.

[Rear chassis]

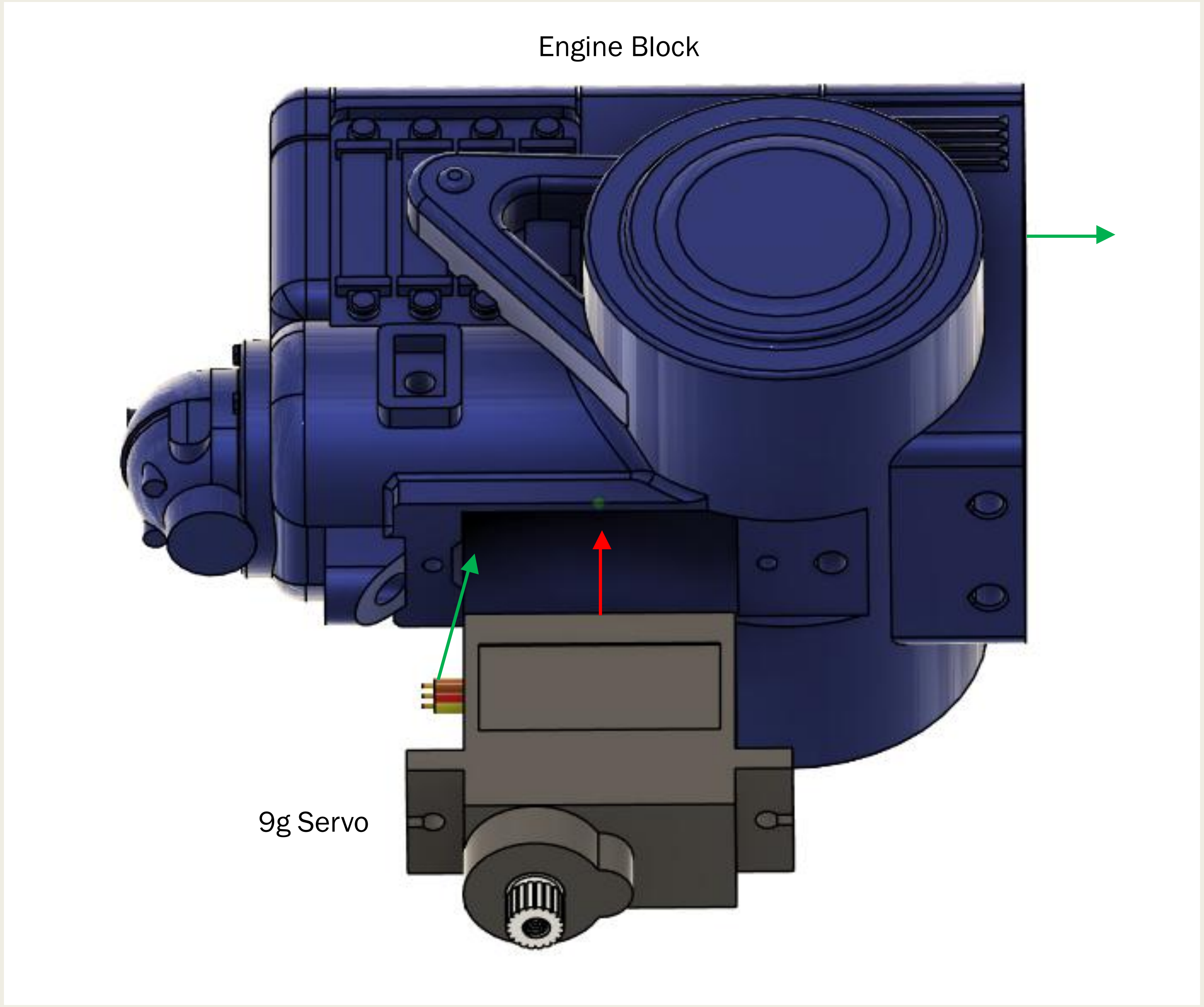
[Step 4]



Insert the motors into the *rear chassis body* and clamp them in place using the motorcovers and 4 M3x10mm screws

[Front chassis]

[Step 5]

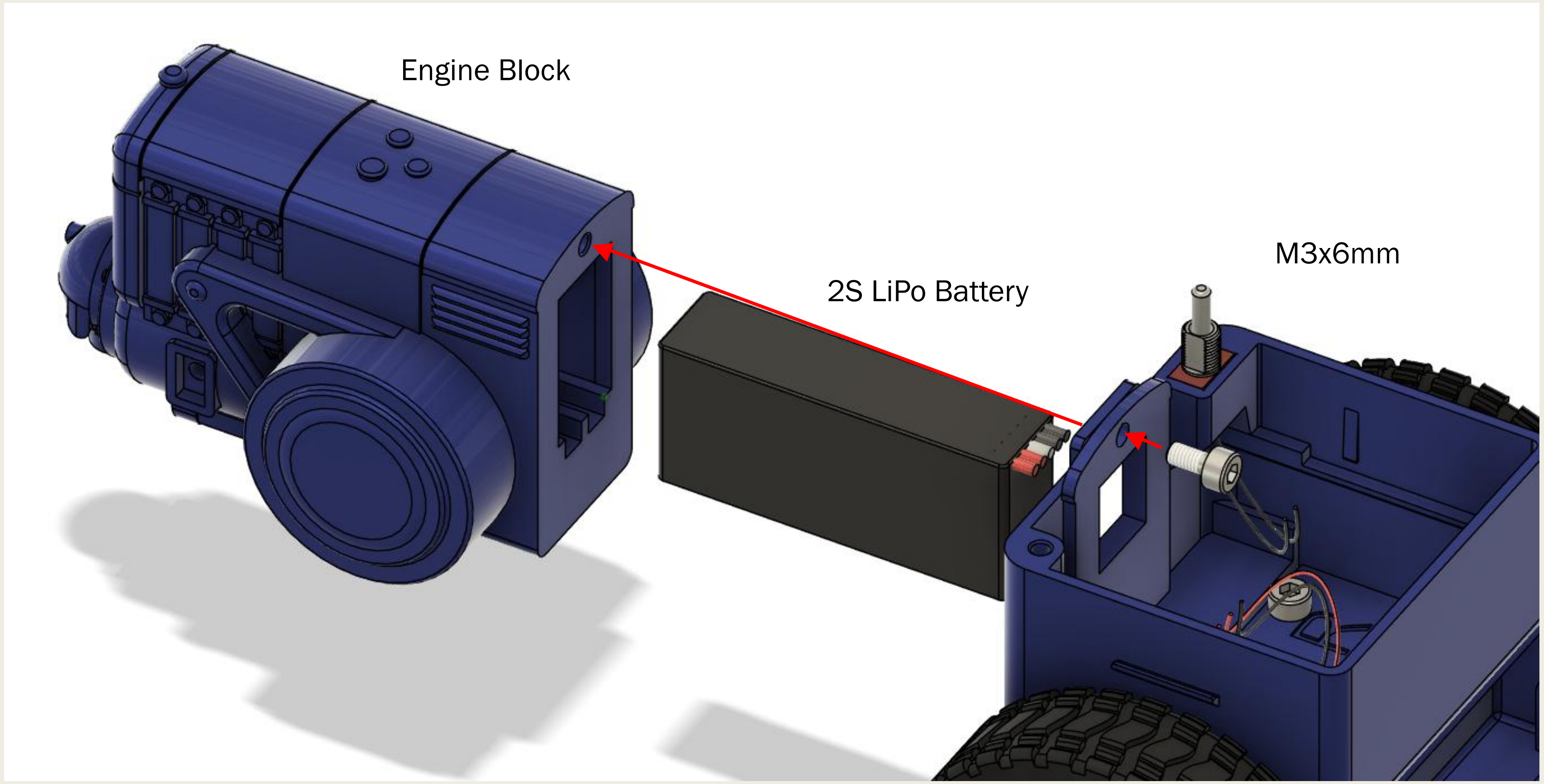


First, feed the Servo wire through the *Engine Block* towards the hole in the back.

For now just slide the 9g Servo into the matching cutout in the *Engine Block*.

[Front chassis]

[Step 6]



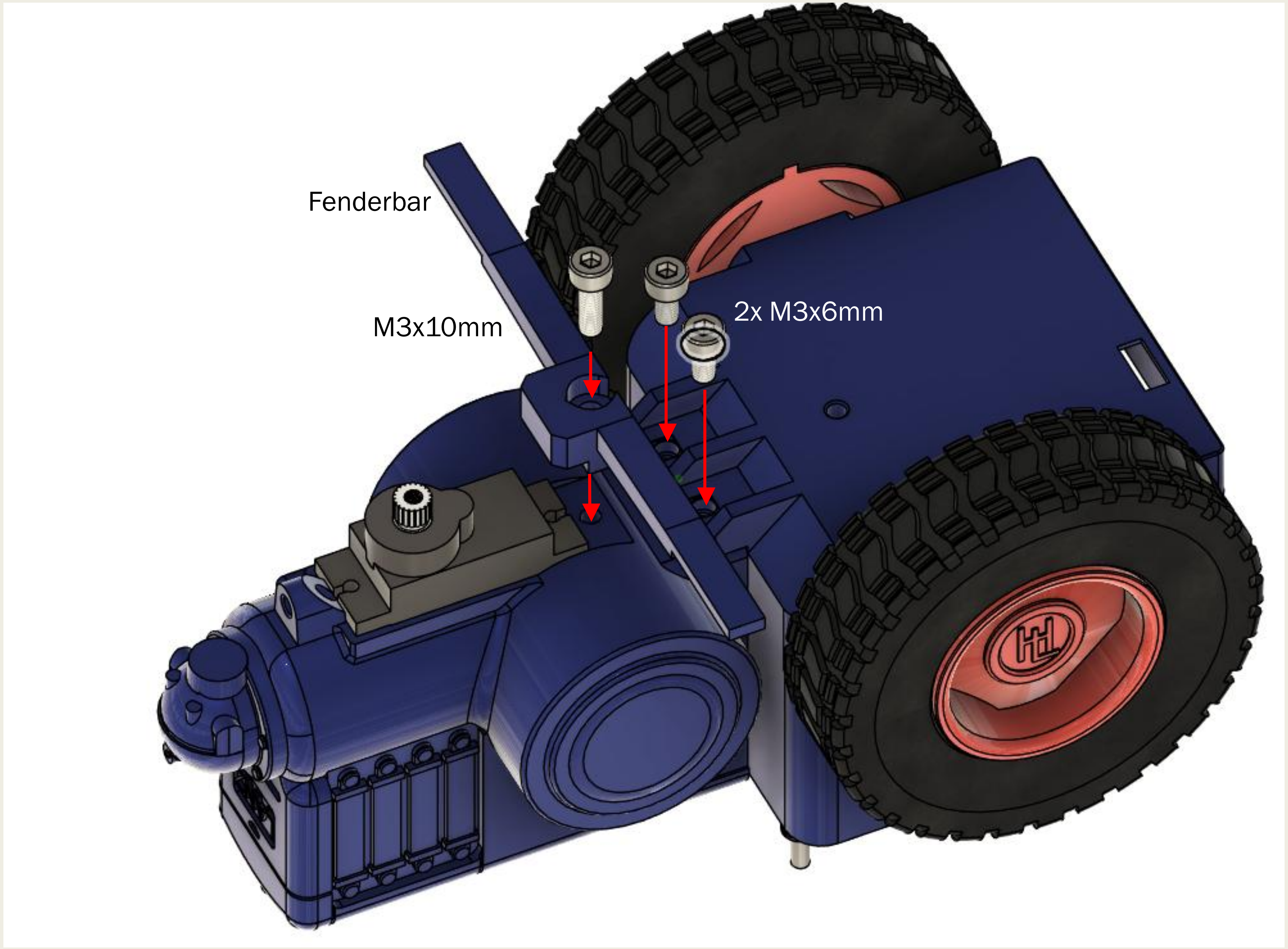
Slide the *2S LiPo battery* into the *Engine Block.*

Next, align the whole assembly to the *rear chassis assembly* and secure it using one M3x6mm screw.

Make sure you pull the battery and servo wires through the cutout in the bulkhead before tightening the screw.

[Front chassis]

[Step 7]

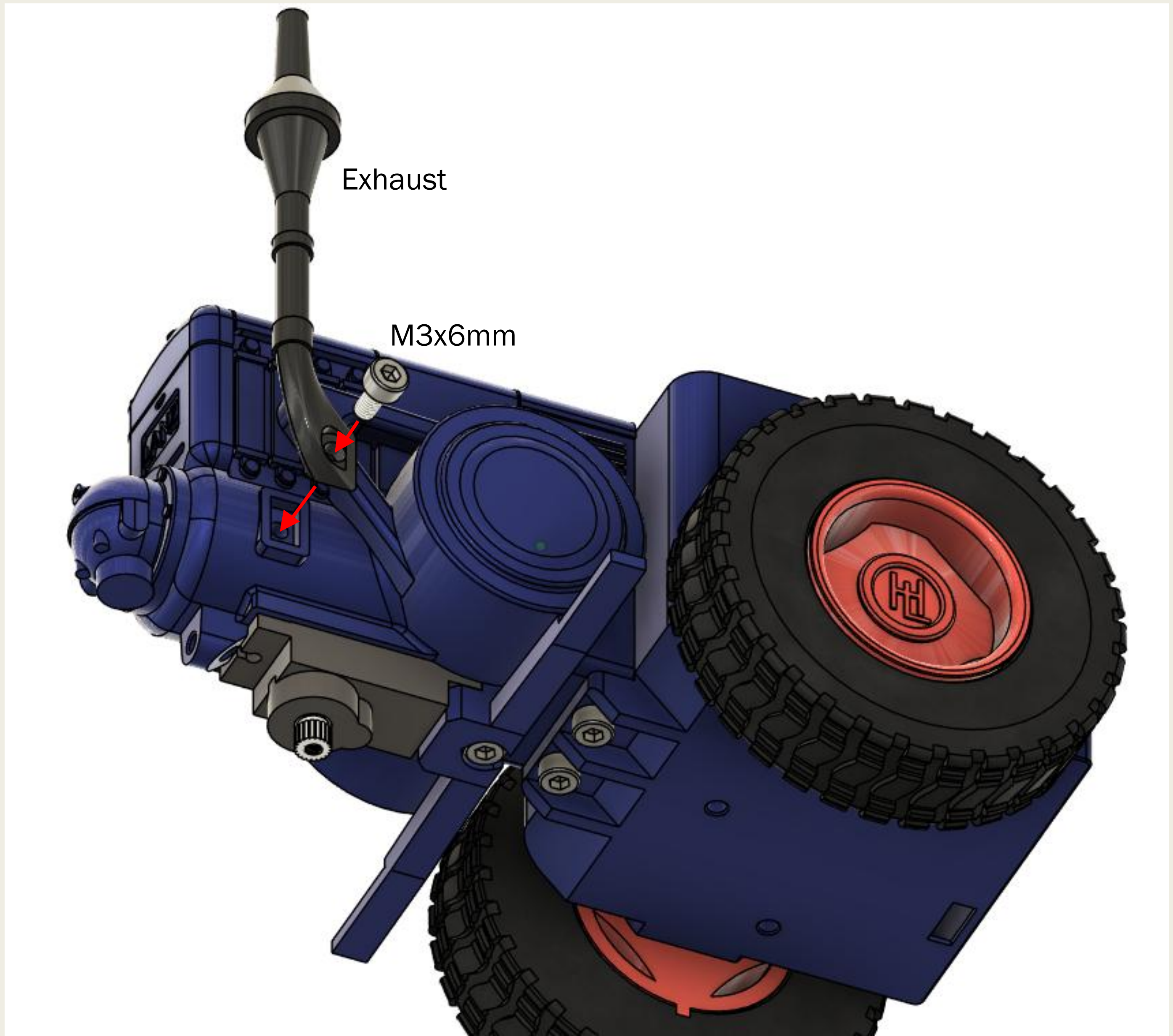


First, secure the *Engine Block* to the *Rear Chassis* using two M3x6mm screws.

At this point the servo from [Step 5] is secured using the *Fenderbar* and a M3x10mm screw.

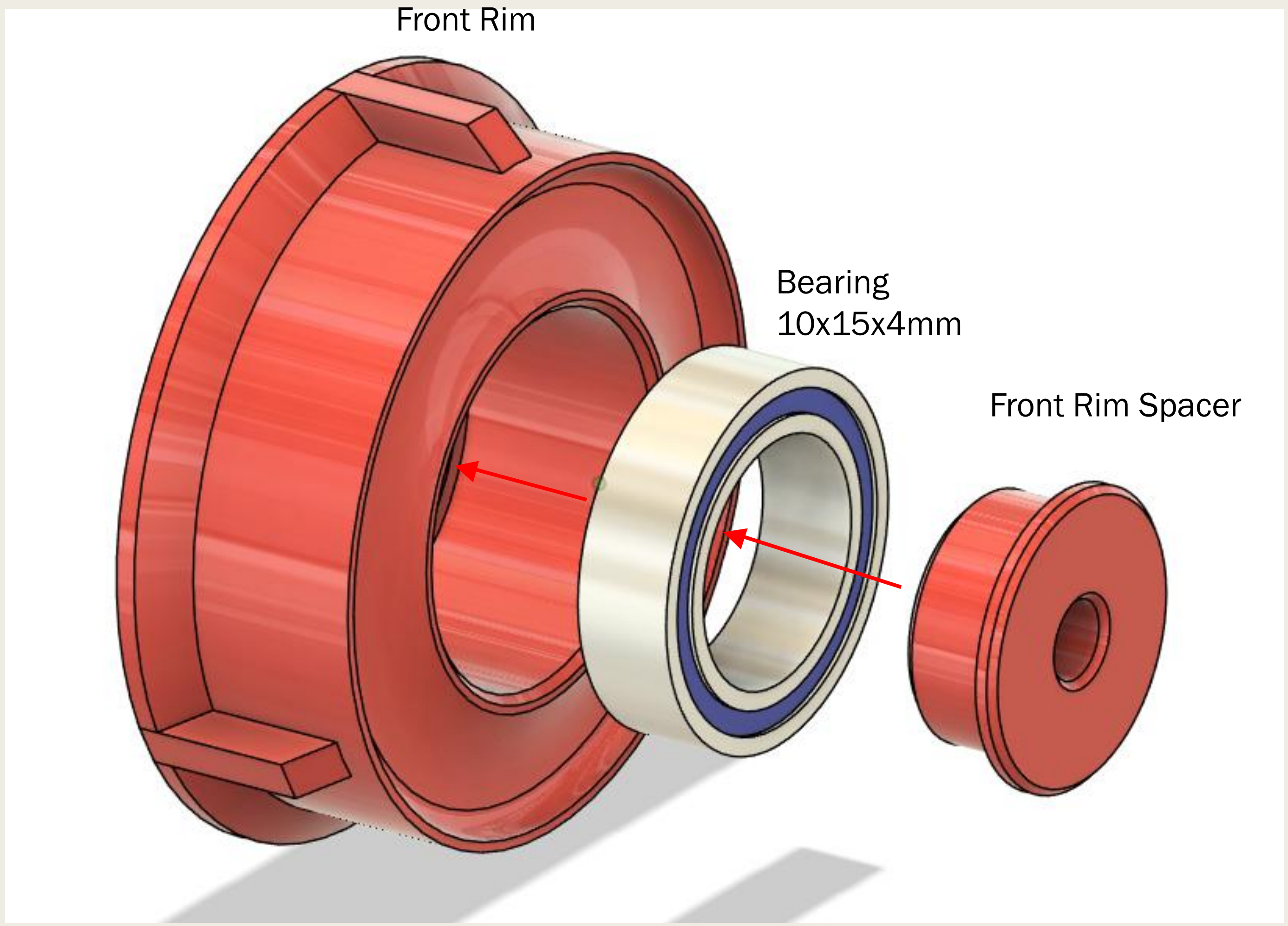
[Front chassis]

[Step 8]



[Front Axle]

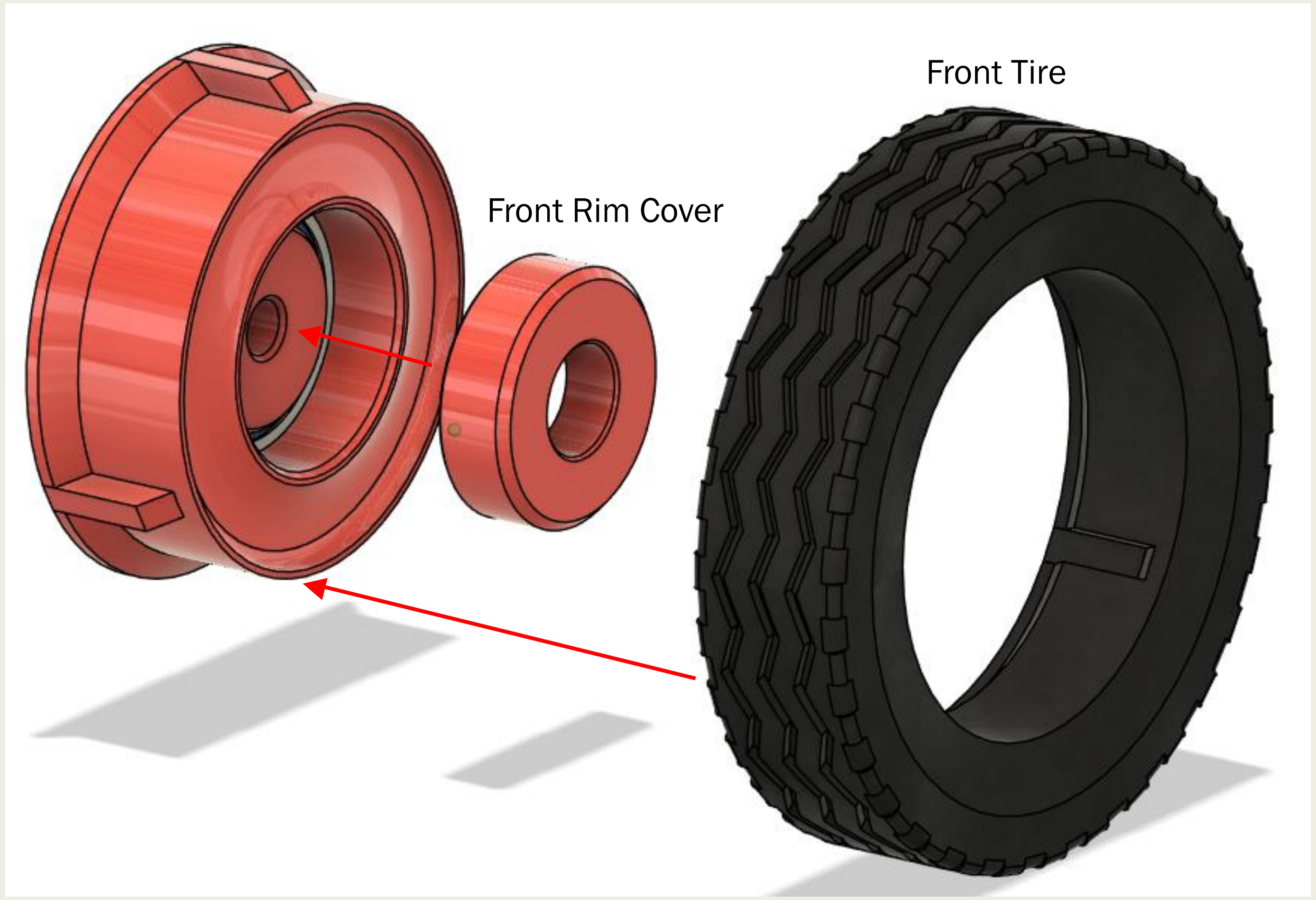
2x



[Front Axle]

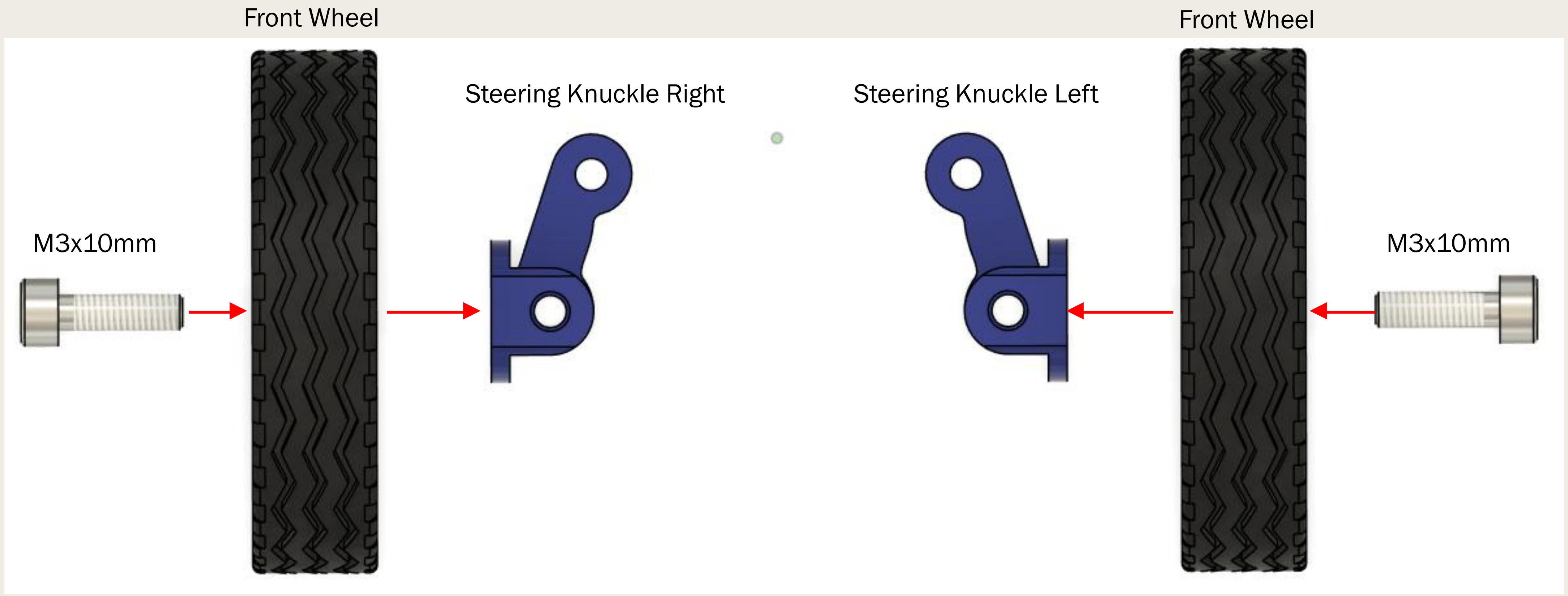
[Step 10]

2x



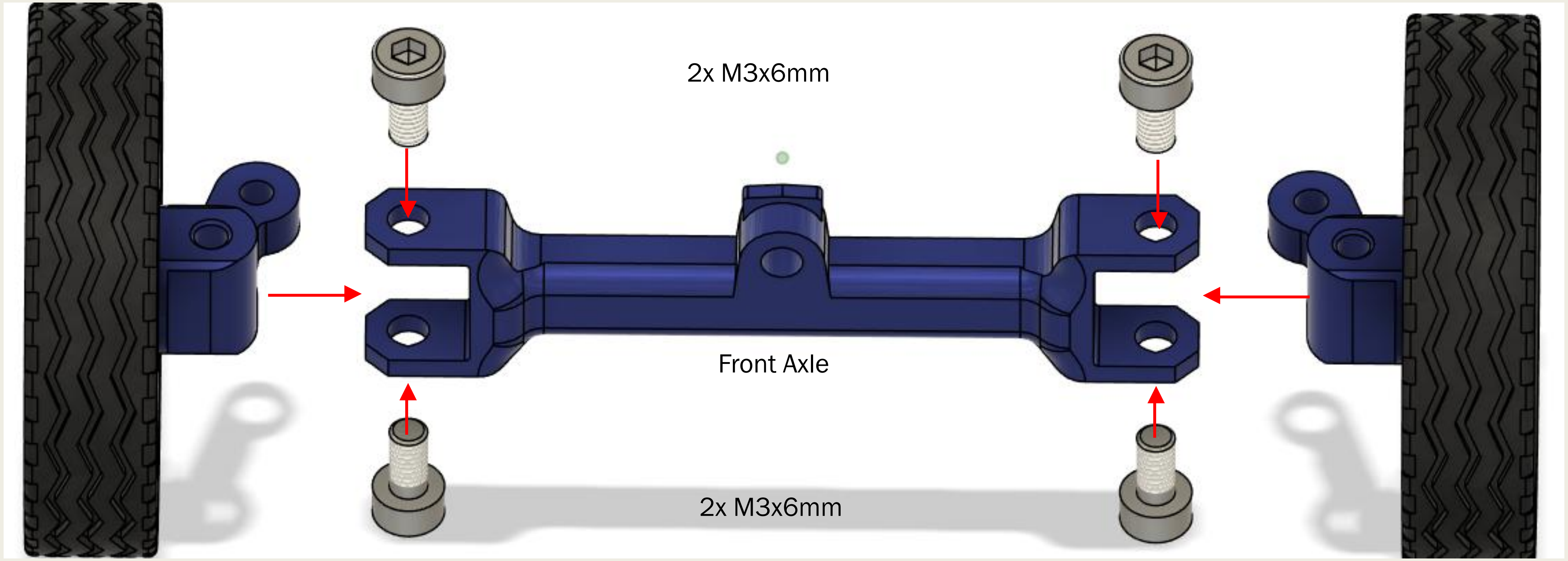
[Front Axle]

[Step 11]



[Front Axle]

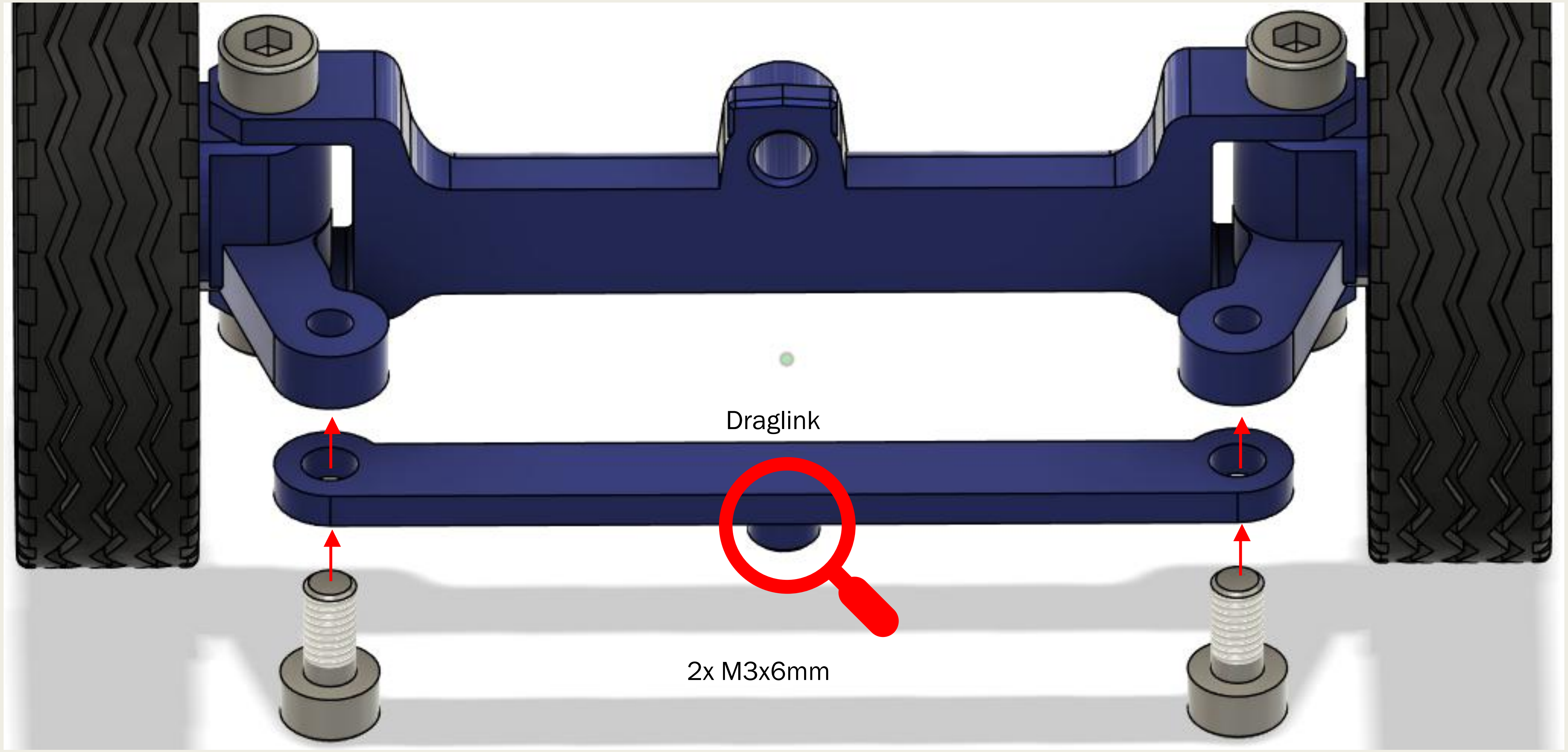
[Step 12]



The M3x6mm screws will bottom out before being fully seated. This is inteded so pleace make sure not to overtighten them and possibly strip the thread from the printed part :)

[Front Axle]

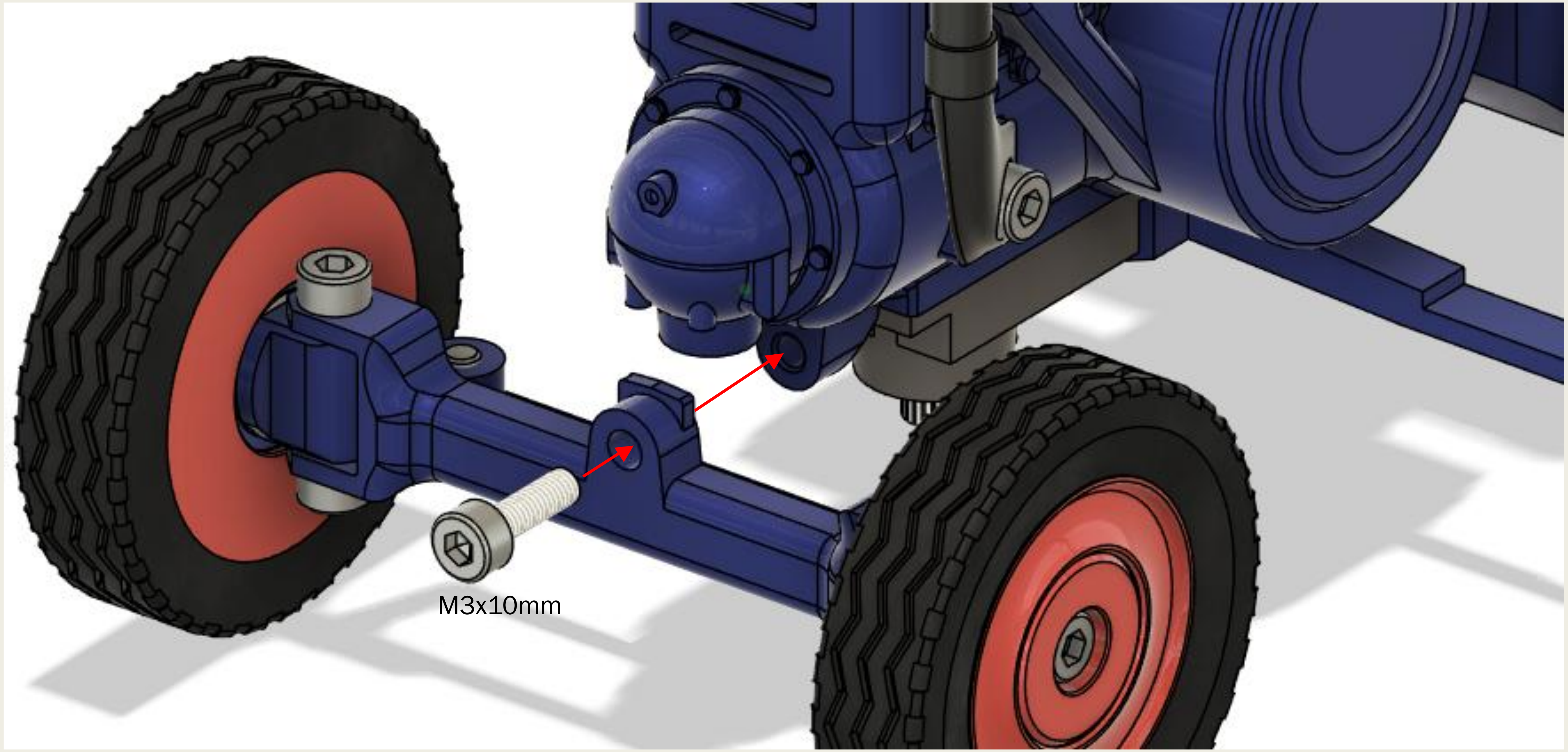
[Step 13]



Make sure the pin of *Draglink* is facing the bottom as pictured!
Only tighten these screws enough so that steering motion is still possible!

[Front Axle]

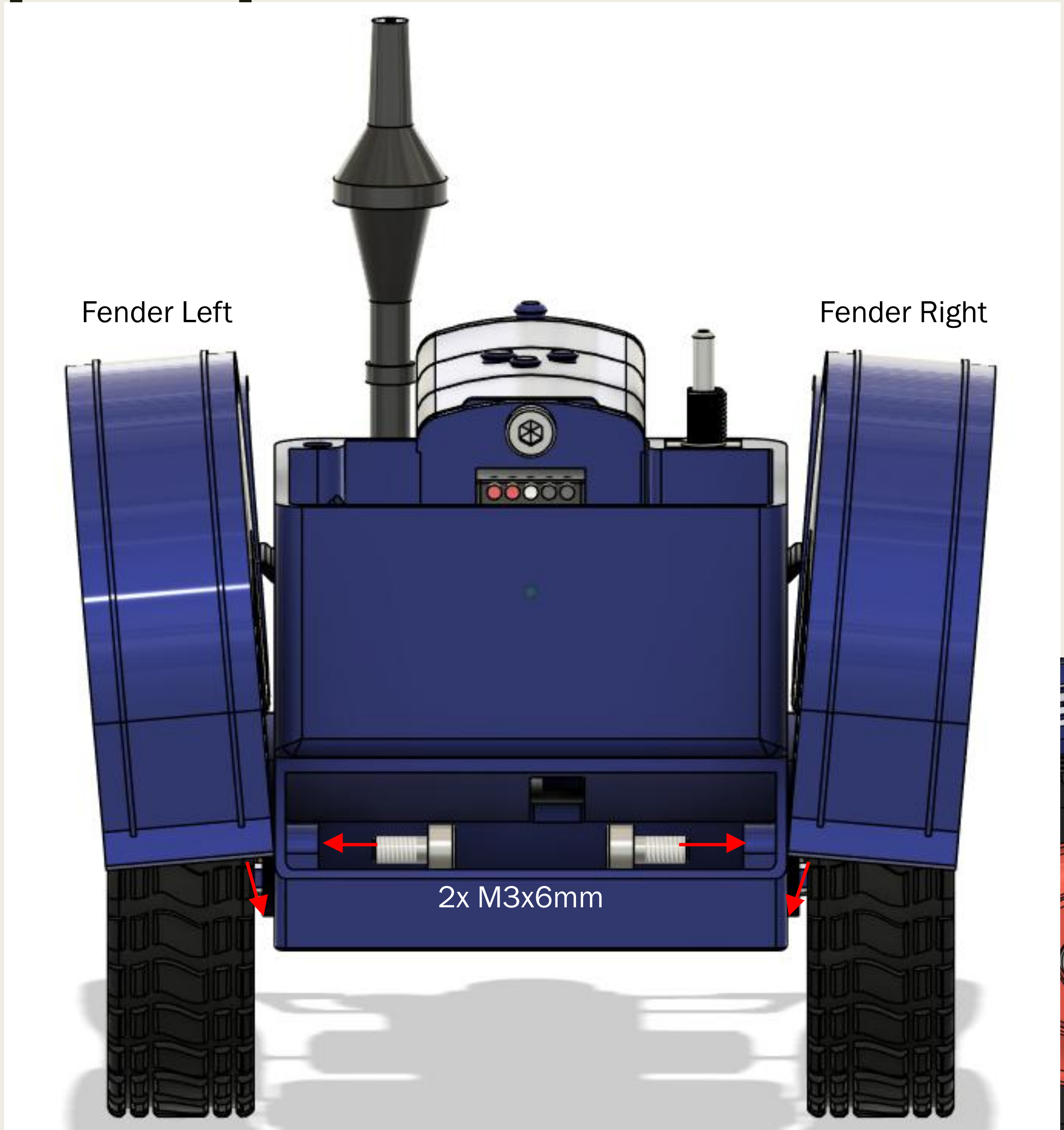
[Step 14]



Only tighten this screw enough so that side to side motion is still possible!

[Fender]

[Step 15]

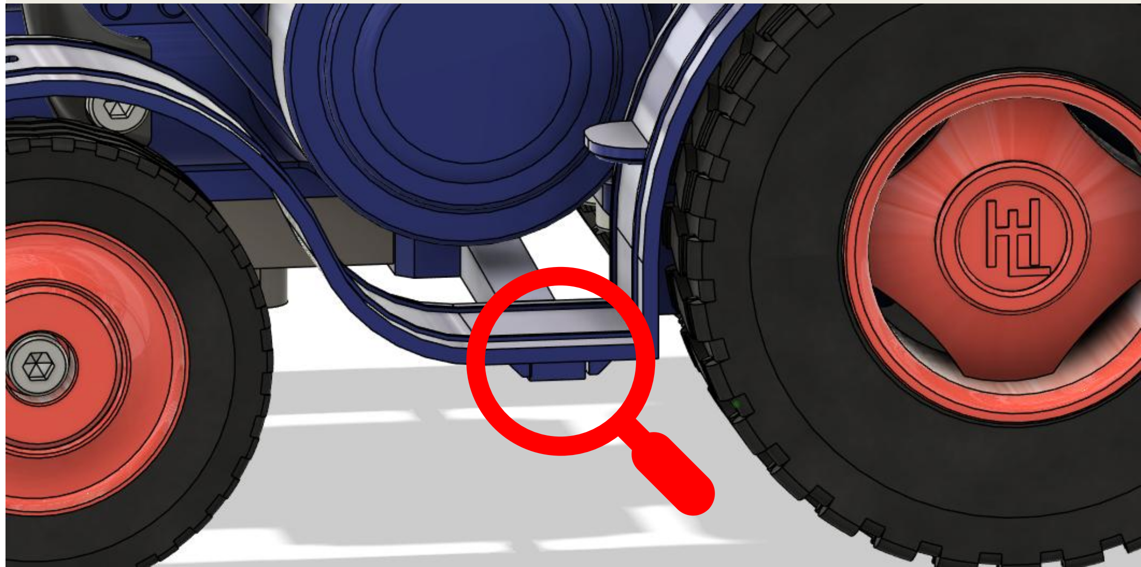


Slide the *Fenders* in at an angle so that the latch on the *Rear Chassis* can catch into the slot in the *Fender*.

Then secure the *Fender* using a M3x6mm screw.

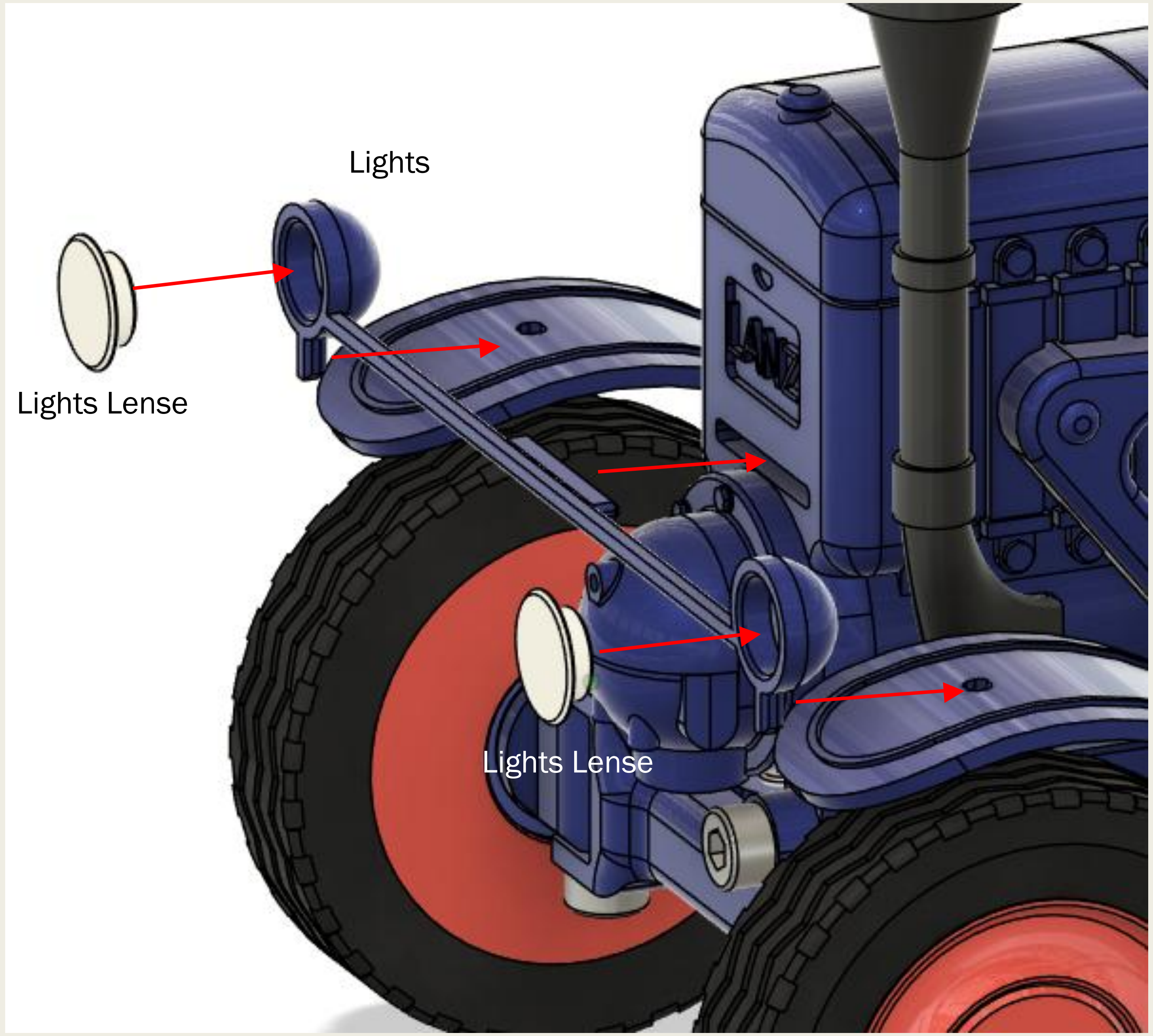
It helps to run the screw through the hole in the Fender to first cut the thread, then remove the screw and assemble as described above.

The front of the *Fender* should sit on the *Fenderbar* as can be seen in the picture below. If you find the fit to be too loose, you can add a drop of hotglue here.



[Fender]

[Step 16]



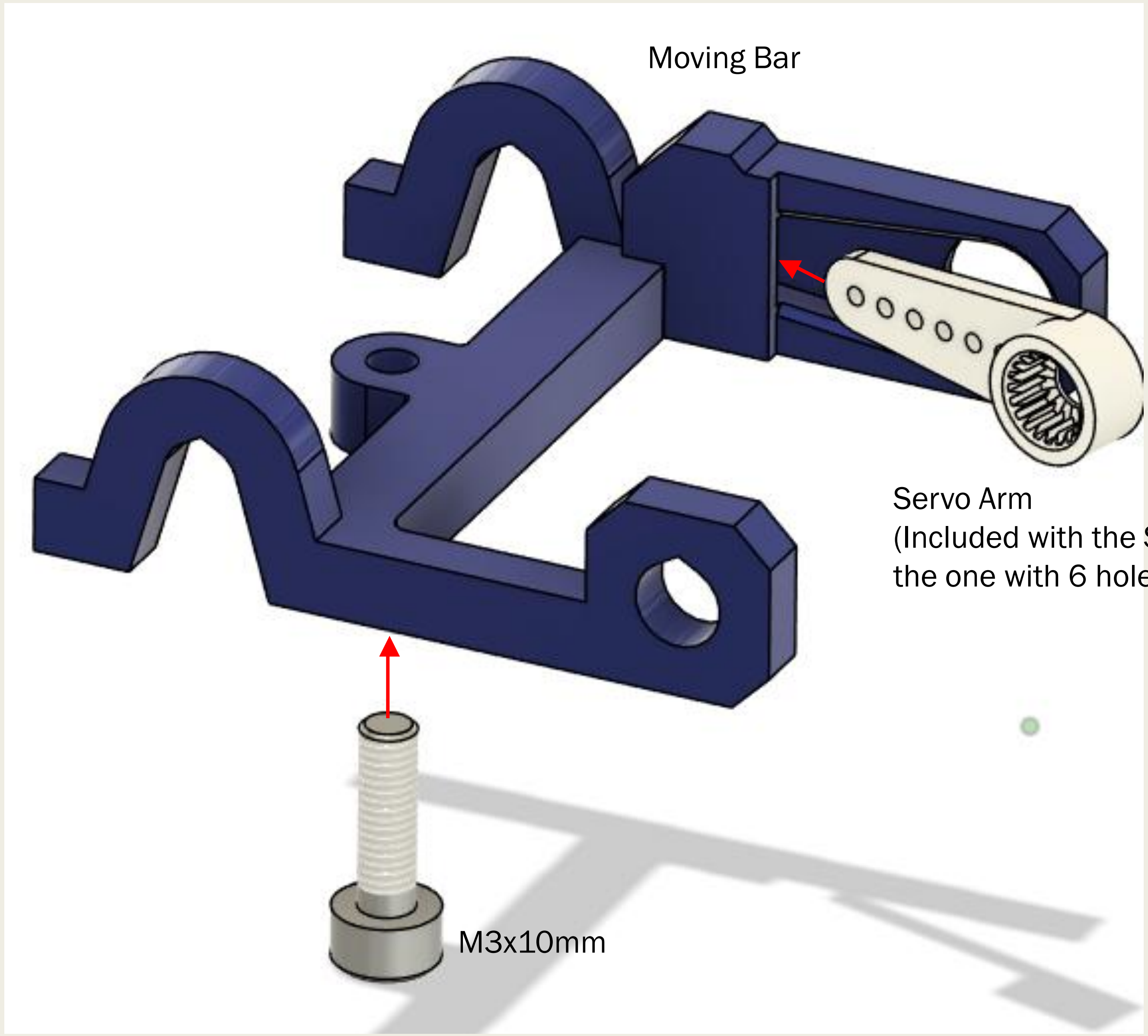
First, glue the *Lights Lenses* to the *Lights.*

Next, snap the Lights in place using the the three pinholes/slots.

Again, if you find the fit to be too loose, you can add a drop of glue to the center slot.

[Coupling]

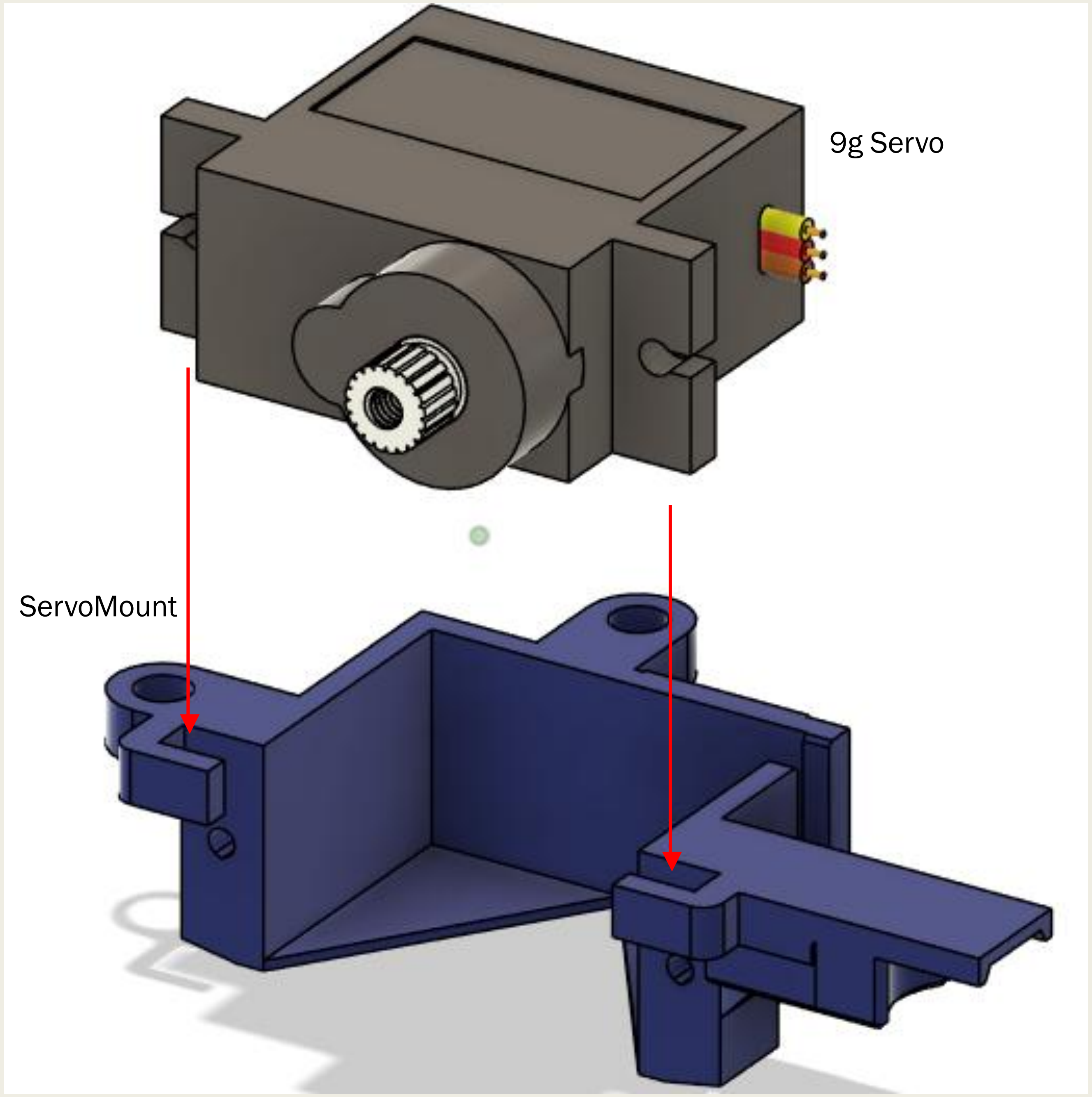
[Step 17]



Silde and rotate the *Servo Arm* into the *Moving Bar*. The fit should be fairly tight but the *Servo Arm* flexes enough to allow assembly.

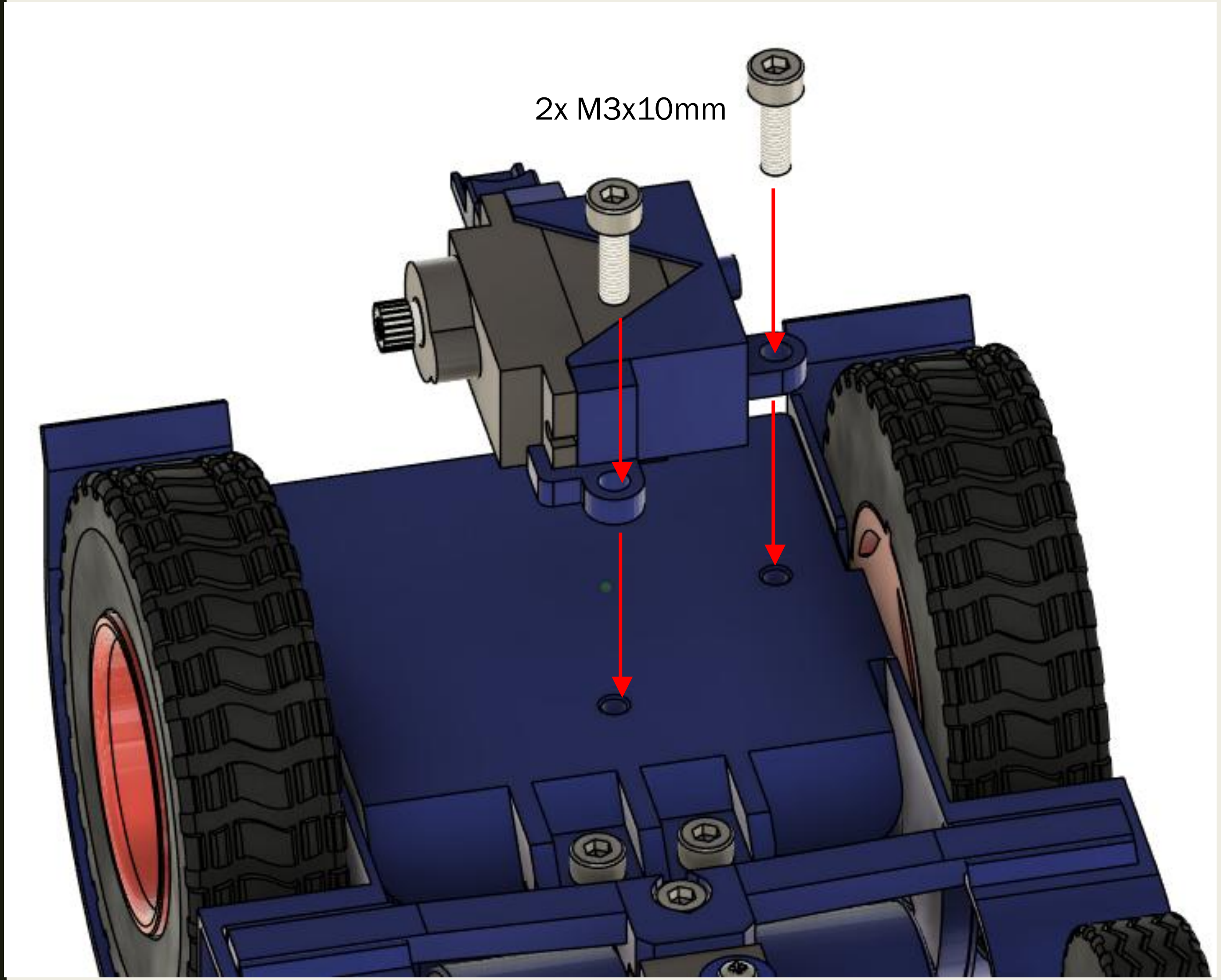
[Coupling]

[Step 18]

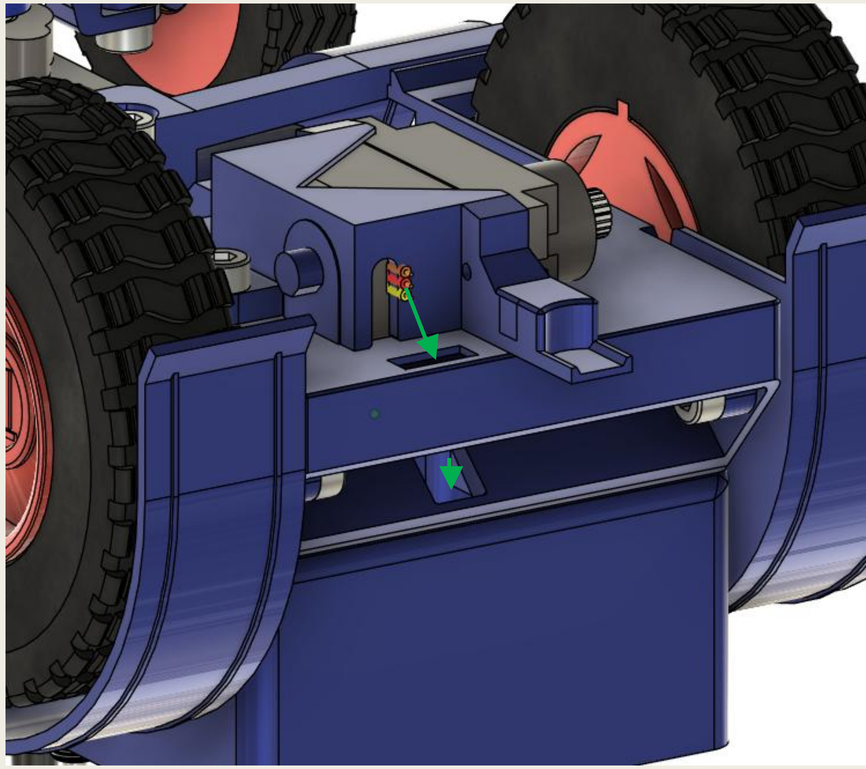


[Coupling]

[Step 19]

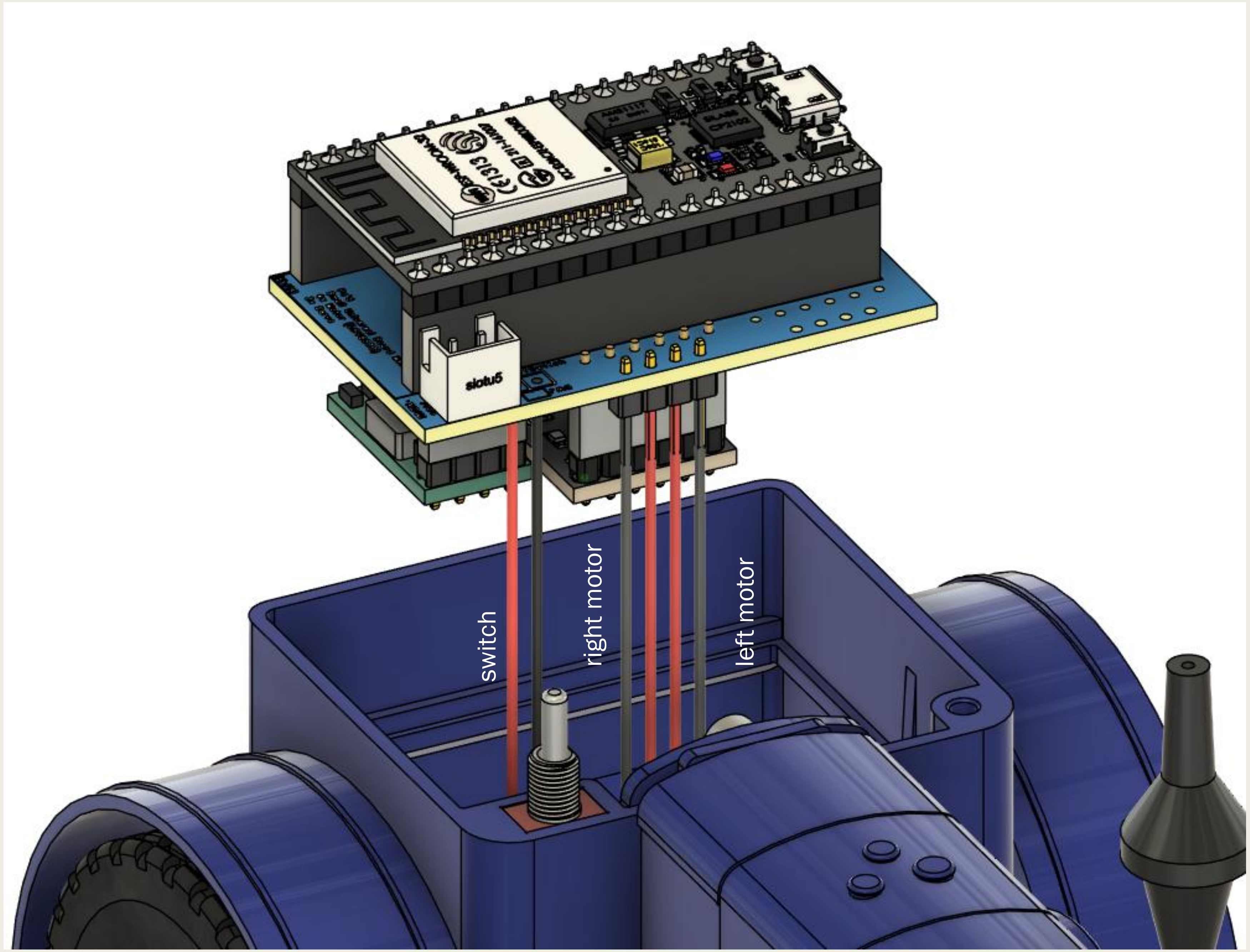


Route the servo wire into the *Rear Chassis* as shown in the picture below. You can later use the available pocket in the back to store the remaining wire length.



[Rear chassis]

[Step 20]

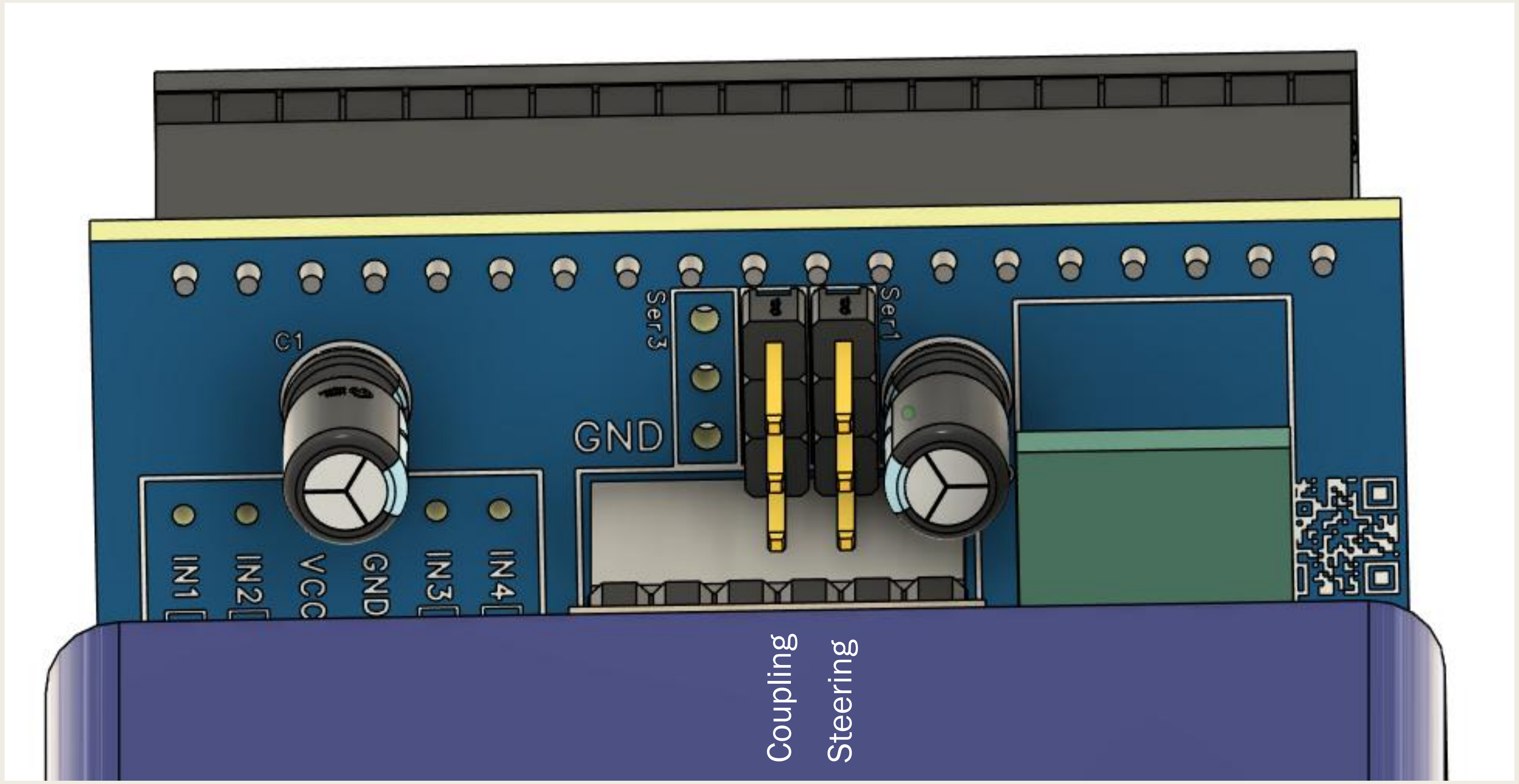


Solder the motor wires to the PCB according to the picture on the left. Note: pay attention to the colors of the wires. If you used the same RPM motors I used and made sure to match the color of the wire to the polarity marked on the motor, the wheels should spin in the correct direction. If not, you can always come back to this step and flip the polarity.

Next, solder the switch wires to the PCB using the “On/OffSwitch“-pins. Polarity doesn‘t matter here

[Rear chassis]

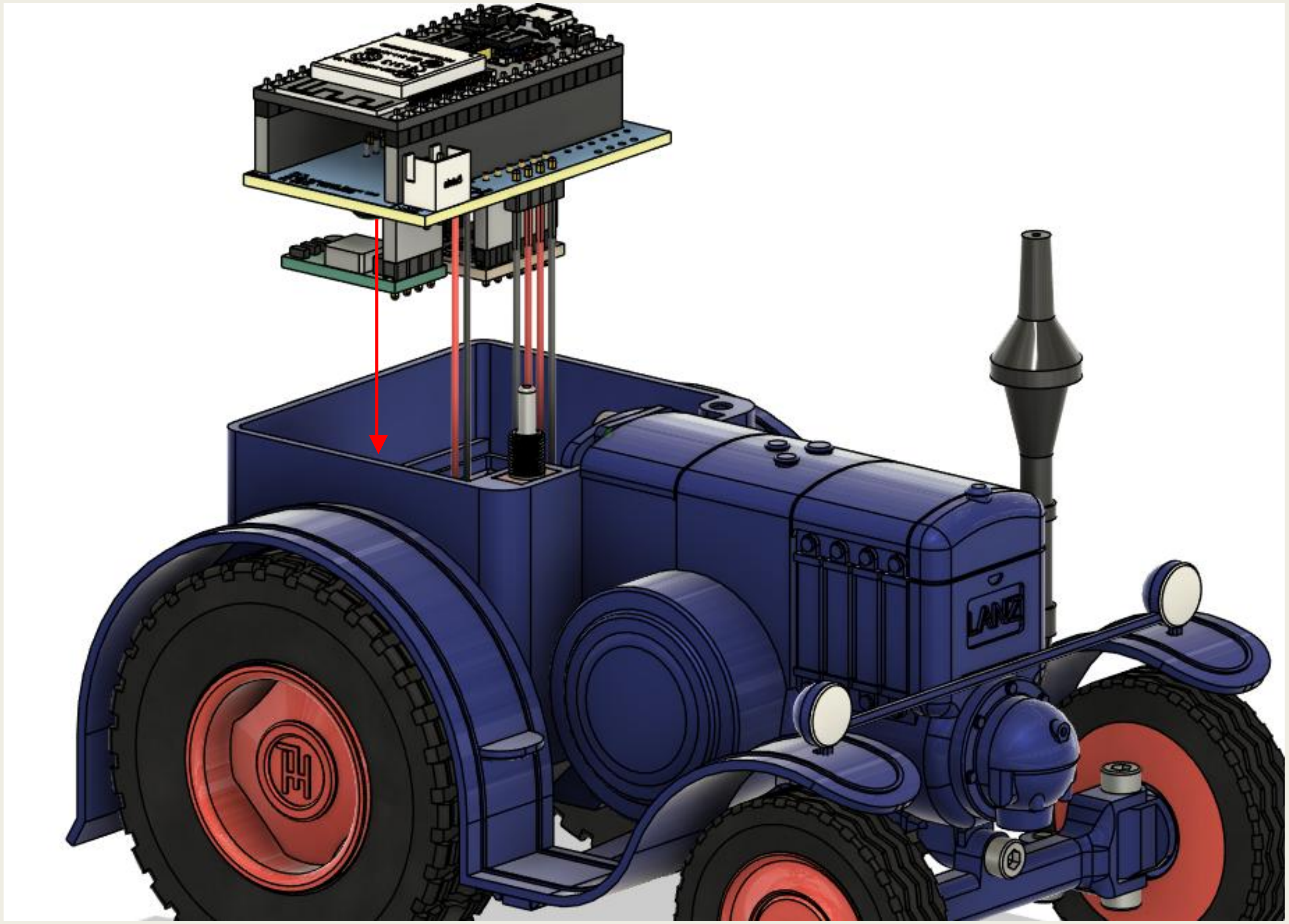
[Step 21]



Now you can plug in the servos into the two prepared servo ports. Make sure the plug is oriented the correct way (the black or brown wire going to GND).
A missmatch usually doesn‘t cause any damage but it won‘t work.

The “Steering“ servo goes in the *Ser1* port, the “Coupling“ servo in the middle (*Ser2*) port.

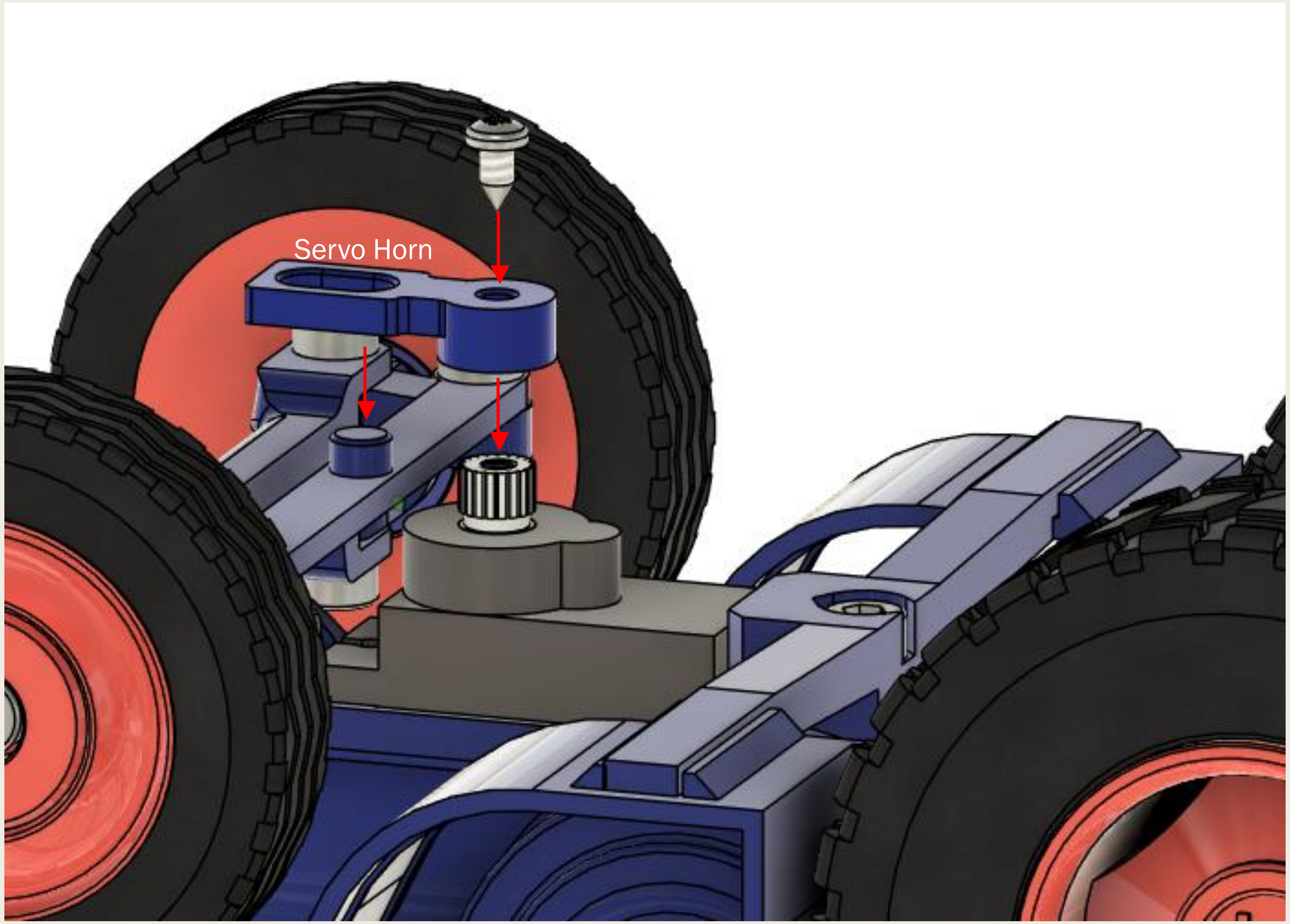
[Rear chassis]



Now clip the PCB into the *Rear Chassis*. Make sure you don‘t pinch or break any wires in the process!

[Servo calibration]

[Step 23]



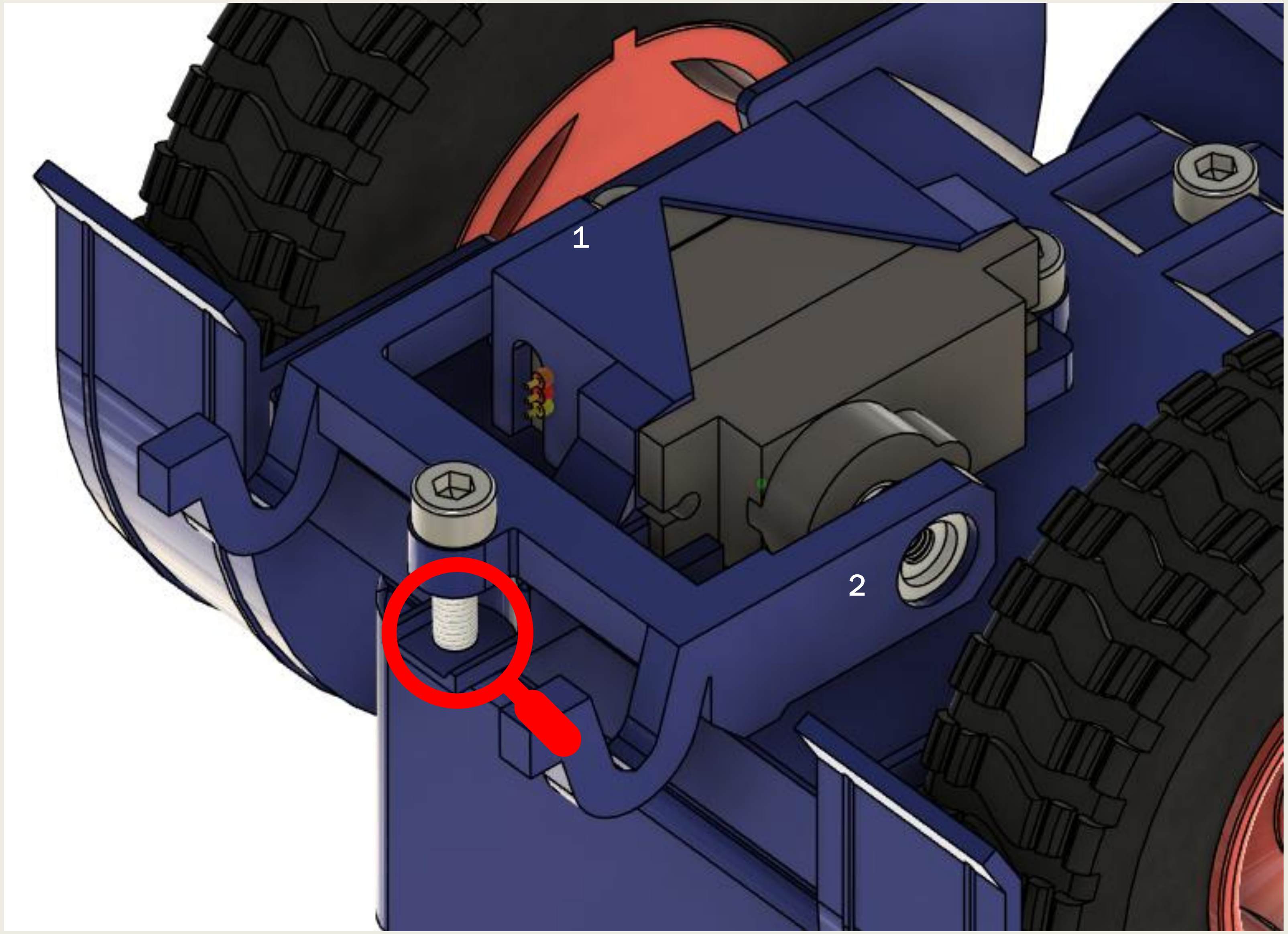
First, turn on the bulldog by plugging in the battery and fliping the switch into it‘s “ON“ position.

The servos will now move to their neutral position. To verify, connect to the ESP as mentioned in Verifying the upload.
By pressing the corresponding buttons in the UI you can test all servos. It‘s best to do this before securing anything to the servos.

Now slide on the Servo Horn and secure it using the screw included with the servo.

[Servo calibration]

[Step 24]

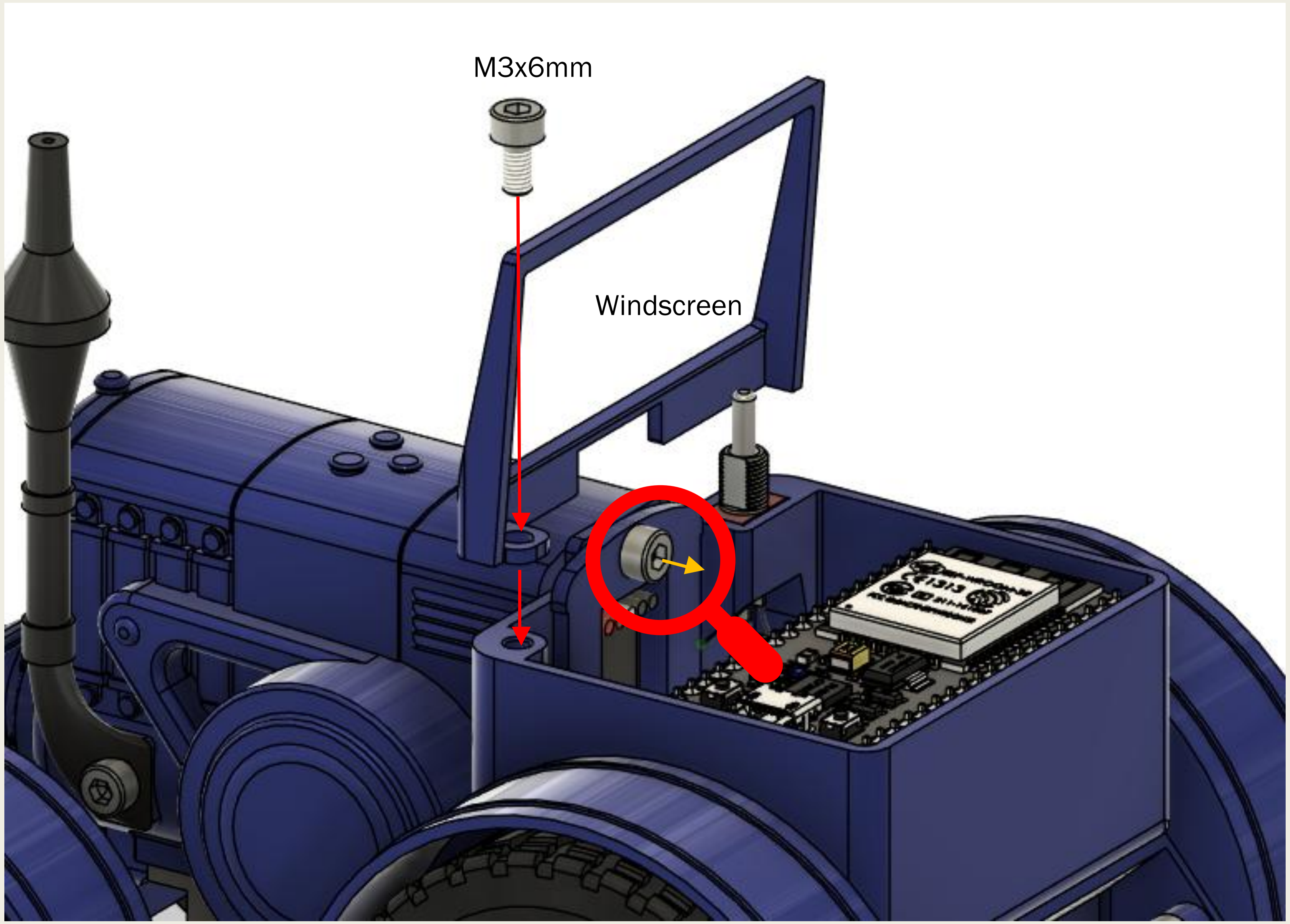


Whilst the coupling servo is in ist “CLOSED“ position, you can add the *Moving Bar* prepared in [Step 17].

To do this, first fit the hole in the *Moving Bar* to the pin on the *Servo Mount*. Then flex the *Moving Bar* open just enough, so it fits over the servo. Before pushing the *Servo Arm* onto the servo, make sure the screw in *Moving Bar* is just about touching the lip on the *Servomount*.

[Rear chassis]

[Step 25]

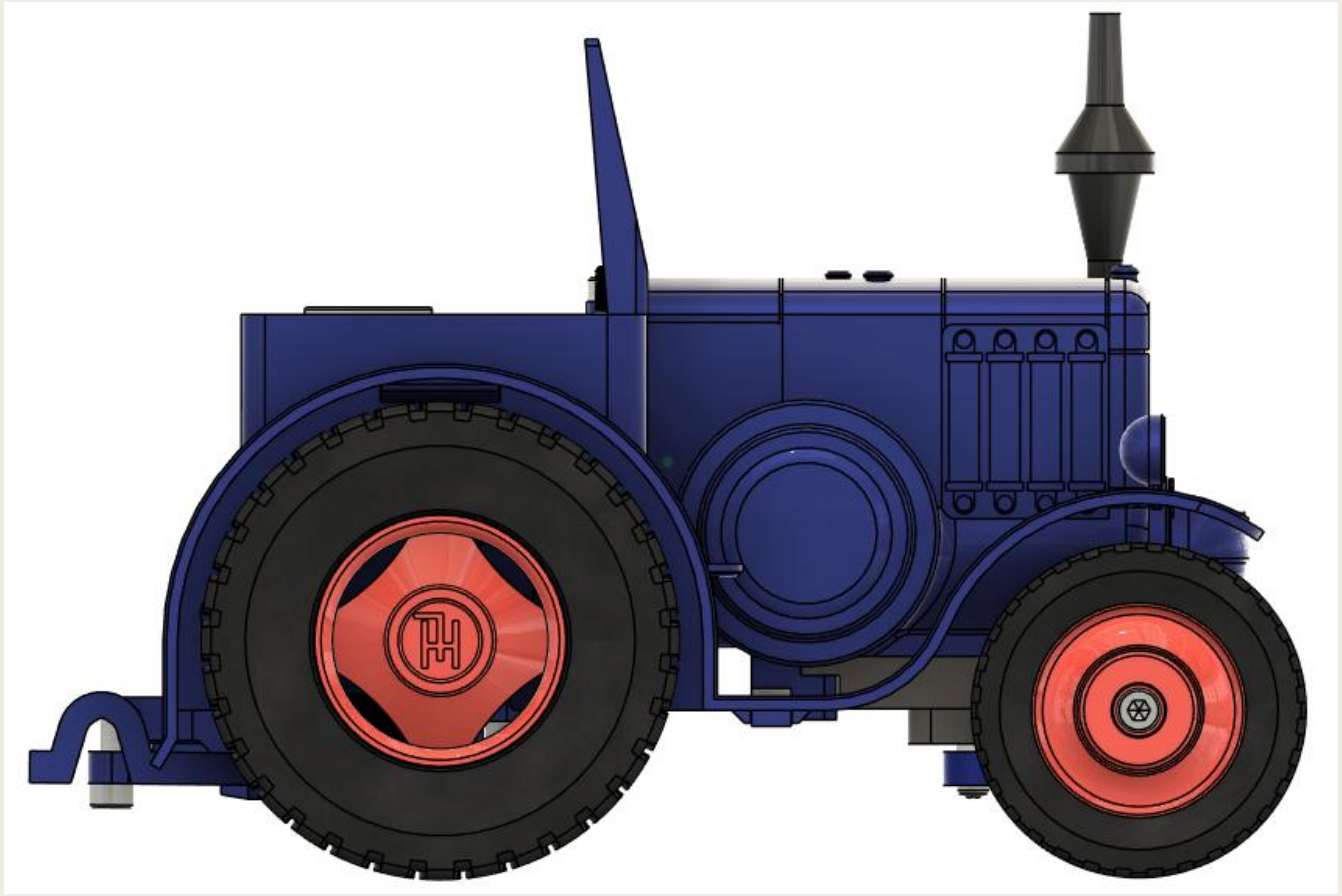


To mount the *Windscreen*, first unscrew the screw - holding the *Front Chassis* to the *Rear Chassis* - a few turns.

Then slide the *Windscreen* into the small gap between *Front Chassis* and *Rear Chassis* and secure it using a M3x6mm screw.

After this, tighten the first screw again.

[Done]

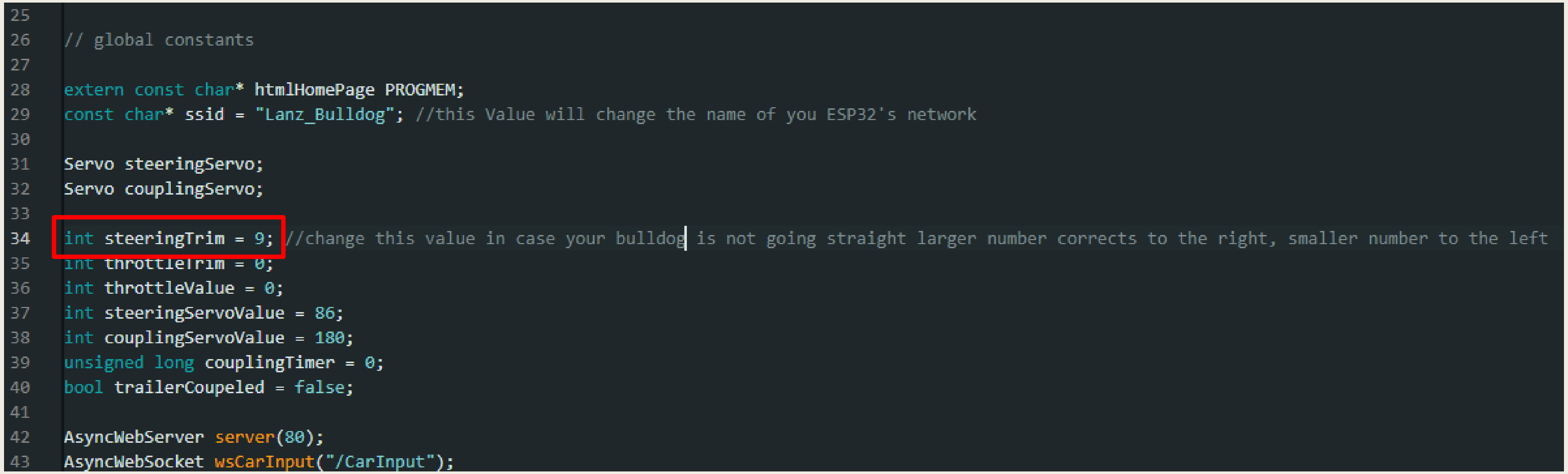


Congratulations – you have successfully assembeled the full bulldog! :)

[Toubleshooting]

[1]

In case the bulldog thends to steer in one direction:
-In code (line 34) change the *steeringTrim* value according to the description



```
25
26   // global constants
27
28   extern const char* htmlHomePage PROGMEM;
29   const char* ssid = "Lanz_Bulldog"; //this Value will change the name of you ESP32's network
30
31   Servo steeringServo;
32   Servo couplingServo;
33
34   int steeringTrim = 9; //change this value in case your bulldog is not going straight larger number corrects to the right, smaller number to the left
35   int throttleTrim = 0;
36   int throttleValue = 0;
37   int steeringServoValue = 86;
38   int couplingServoValue = 180;
39   unsigned long couplingTimer = 0;
40   bool trailerCoupeled = false;
41
42   AsyncWebServer server(80);
43   AsyncWebSocket wsCarInput("/CarInput");
```