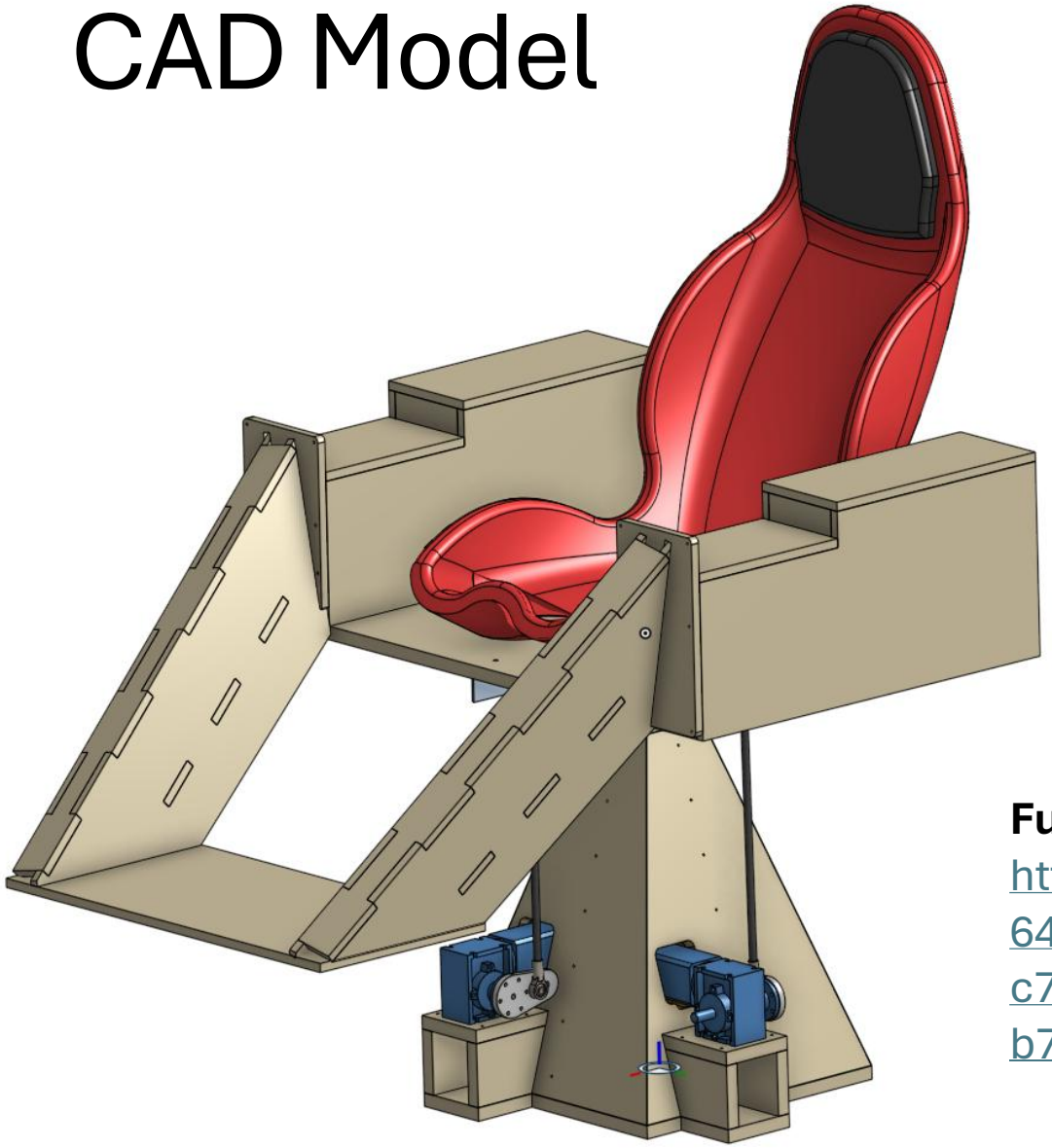
# DIY Motion Simulator

Michael Rechtin - 2025

# Project Overview

- This project is a great combination of woodworking, and mechatronics which allows you to build a motion simulator capable of flying both RC planes as well as simulator games at a very low price compared to commercial options.
- THIS IS A DIY PROJECT. I have documented my build in this document as well as released the CAD and Code associated to make replicating this build as easy as possible. However, part availability in your local area or different applications may warrant changes to the design for your use.

# CAD Model



**Full CAD:**
https://cad.onshape.com/documents/14465a5308115755a641bf91/w/473302792caf9fe5237c9eca/e/42b5f99c290e18c70798f158?renderMode=0&uiState=67888ffc05482c0eb0b74d11

# Tools Required

- A large portion of this project involves cutting shapes out of 0.5” (12.5mm) plywood and 0.75” (19mm) plywood. Many different saw types including circular saw, jigsaw, bandsaw can create the cuts needed in the project. A CNC router can also be an ideal choice if available.

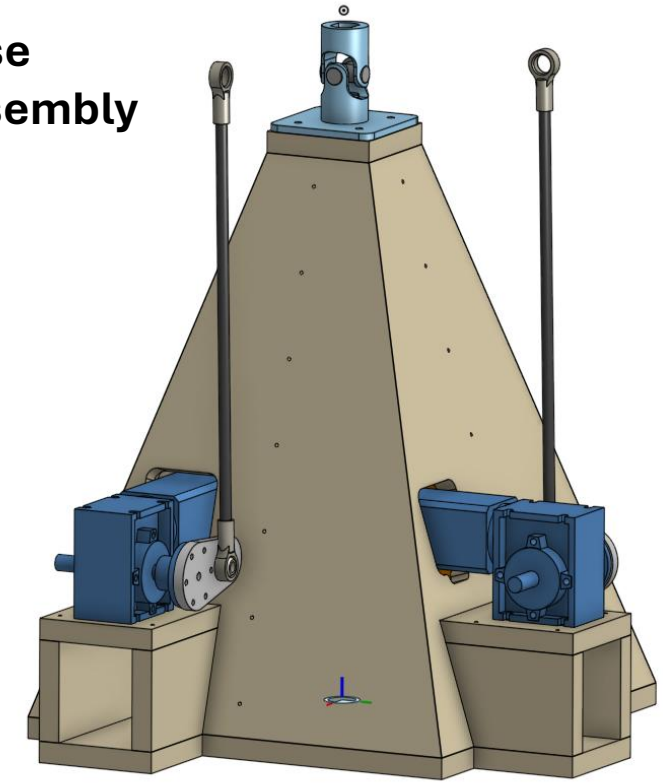
- Tools required:
  - Some type of saw (circular saw, jig saw, table saw, band saw or CNC router with 4’x2’ or greater cut area)
  - Power drill, impact driver
  - Angle grinder or hack saw
  - Soldering iron
  - 3D printer

# Mechanical Assembly

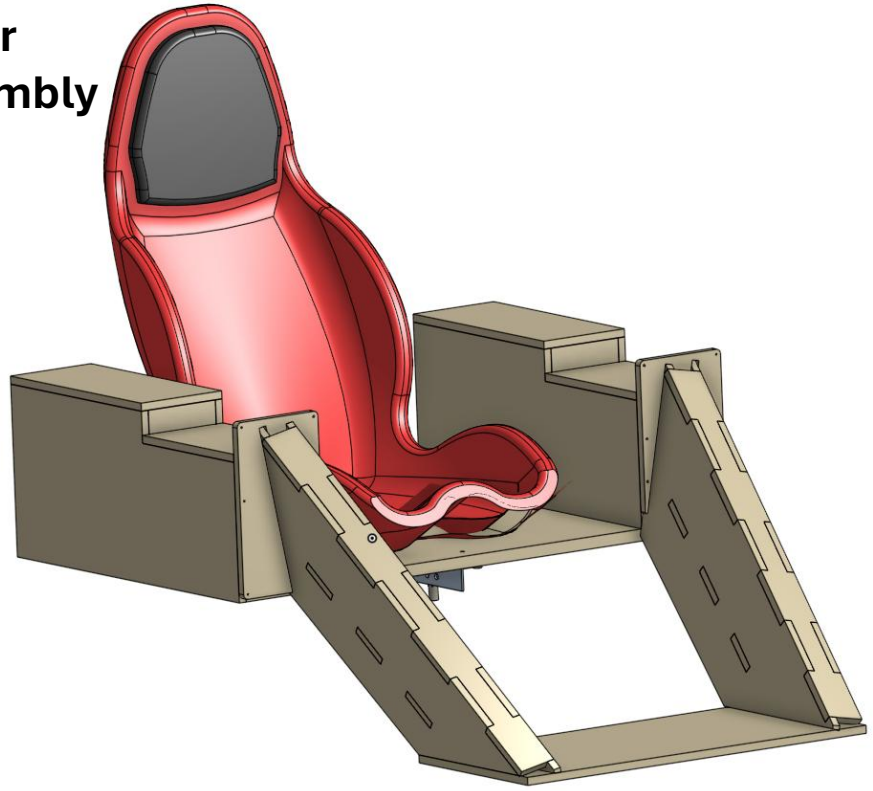
# Parts Overview

- This chair breaks down into two main parts for easier transport. These parts are built separately and then joined together using universal joint.

**Base Assembly**

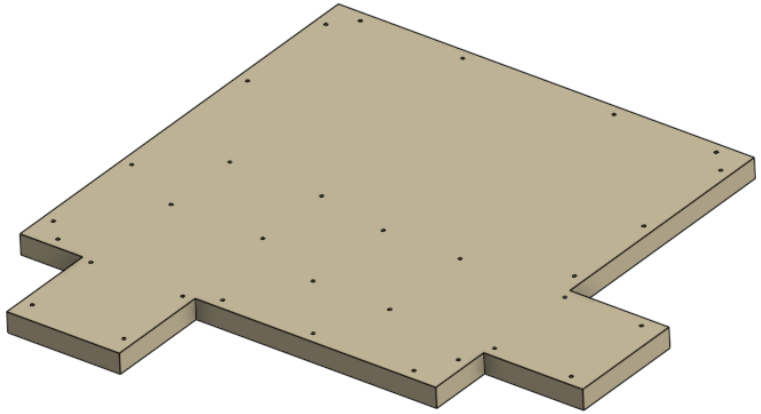


**Upper Assembly**

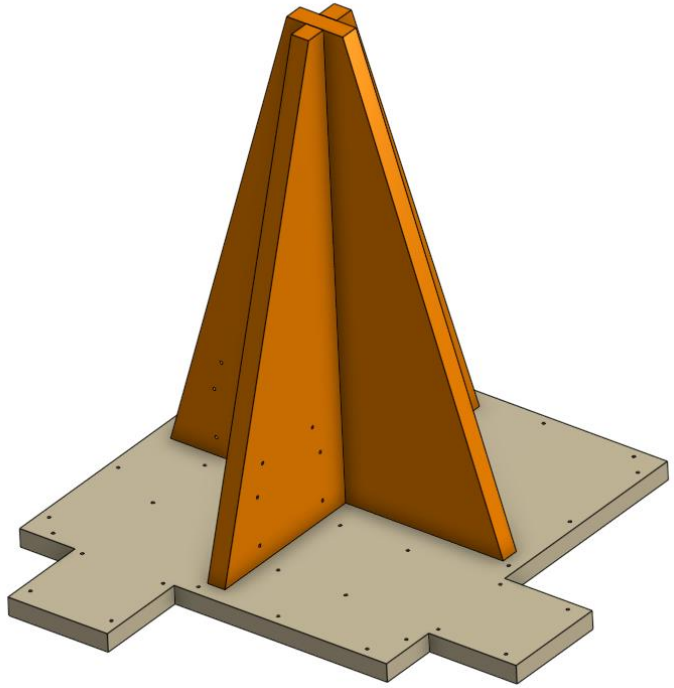


# Base Assembly Structure

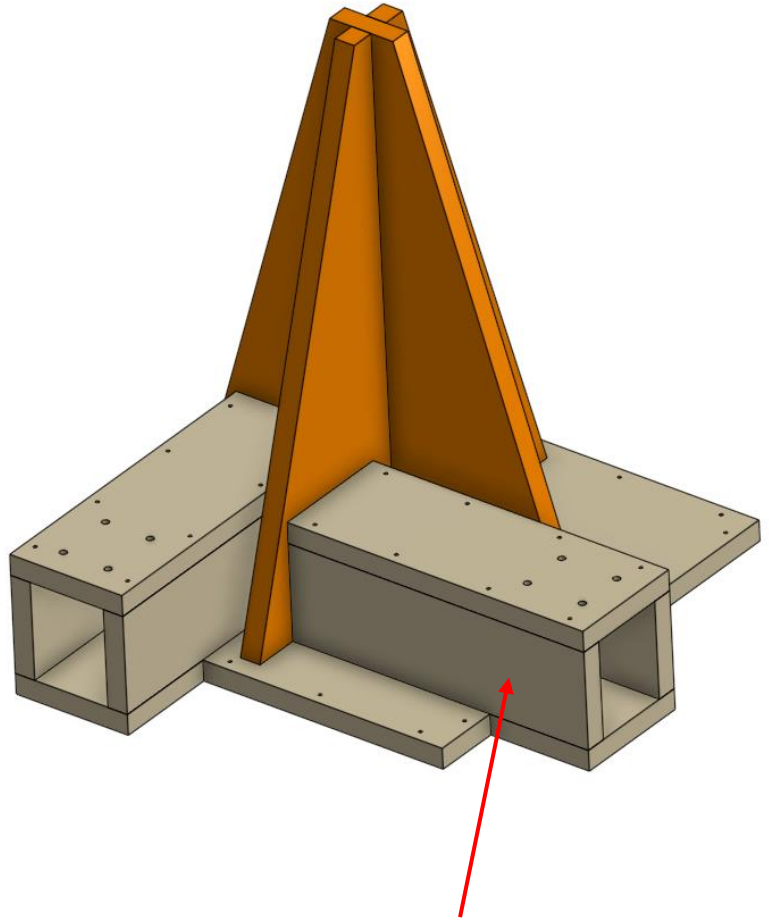
1. Start with base plate



2. Attach middle support structure using slot it interlock. Attach to base using screws and glue

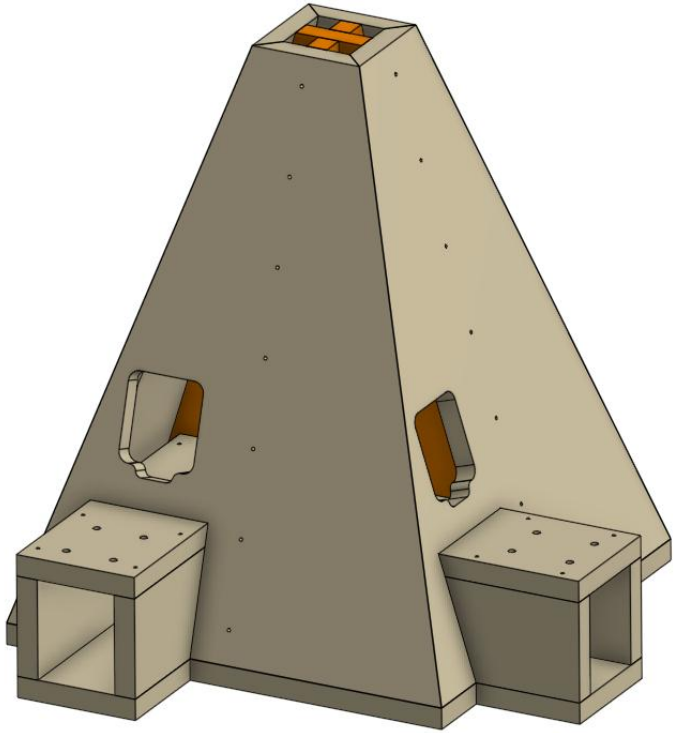


3. Attach motor mounts to the base plates and internal structure using screws and glue

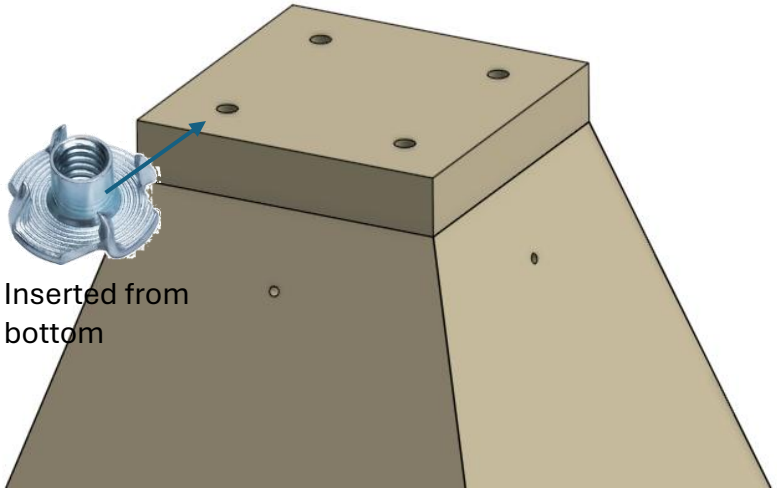


# Base Assembly Structure

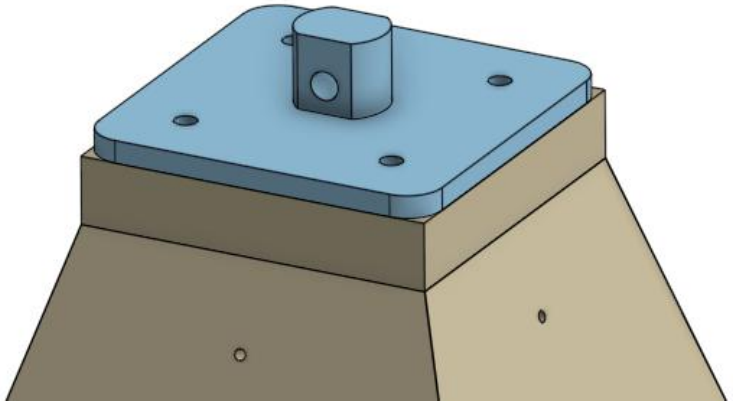
4. Attach outer panels. Perform multiple dry fits since this piece has multiple compound angles. Attach using screws and glue



5. Attach top plate. **Ensure that the holes for the joint mounting plate are already drilled and 0.25”x20 threaded inserts are already installed**. Attach with screws and glue

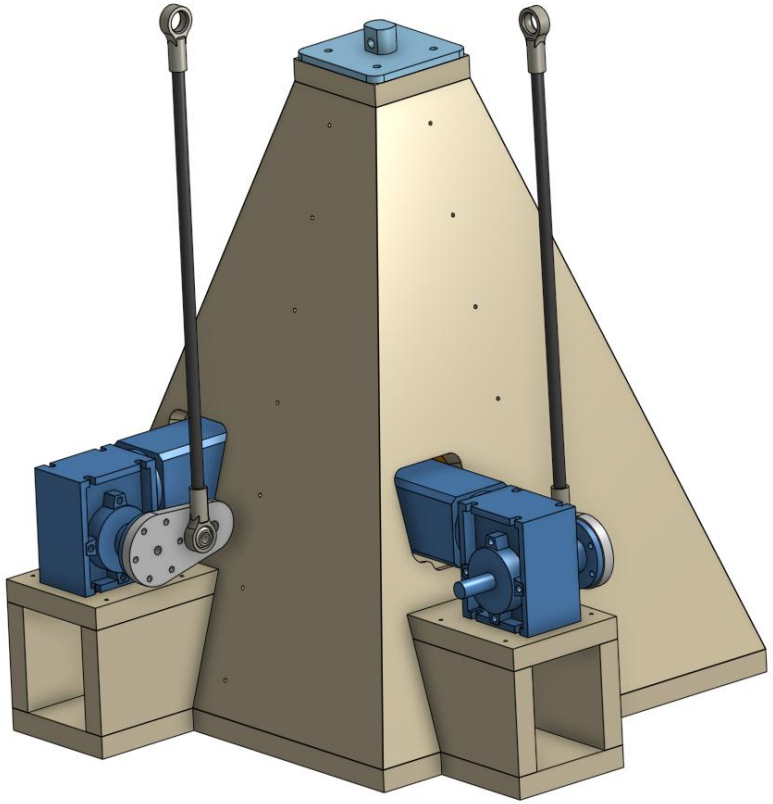



6. Attach joint mounting plate using 0.25”x20 bolts. Joint mounting plate should ideally be machined from aluminum or steel

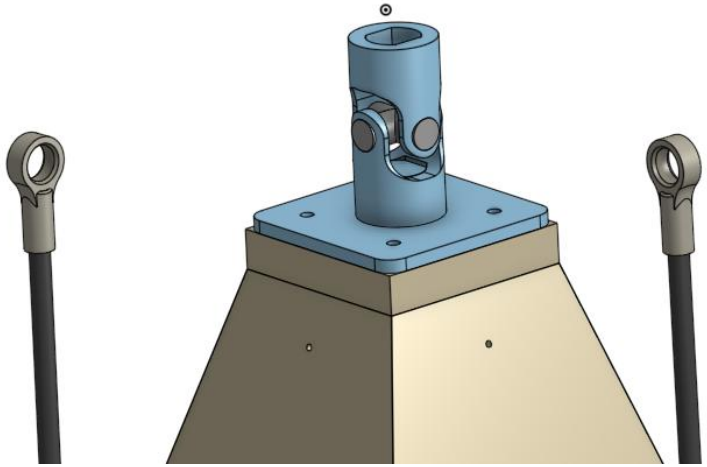


# Base Assembly Structure

7. Attach motors, motor arms, threaded rod, heim joints

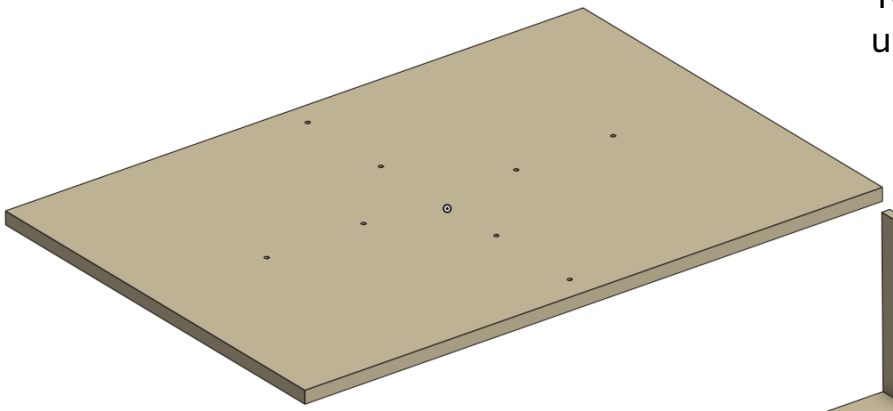


8. Attach universal joint

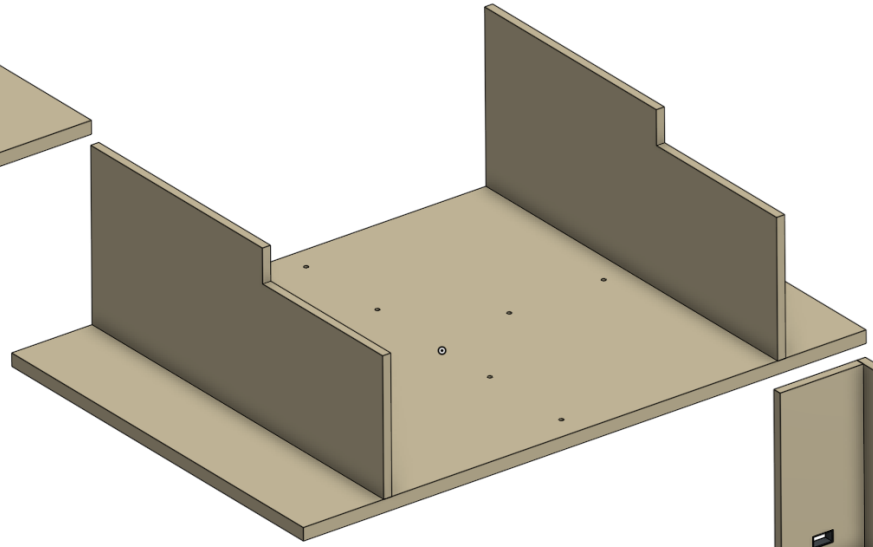


# Chair Assembly Structure

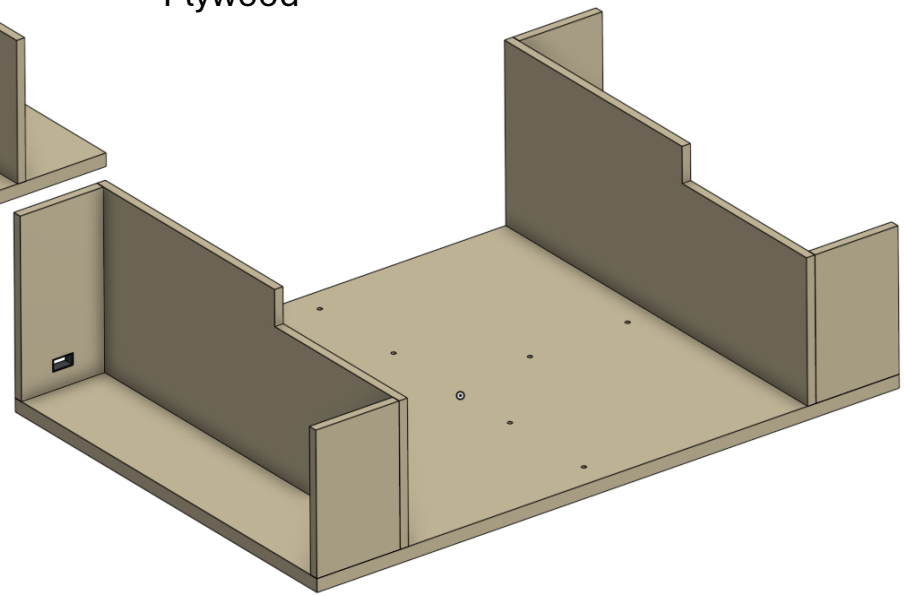
9. Start with baseplate. 0.75” Plywood



10. Add internal side panels and attach using screws and glue. 0.5” Plywood

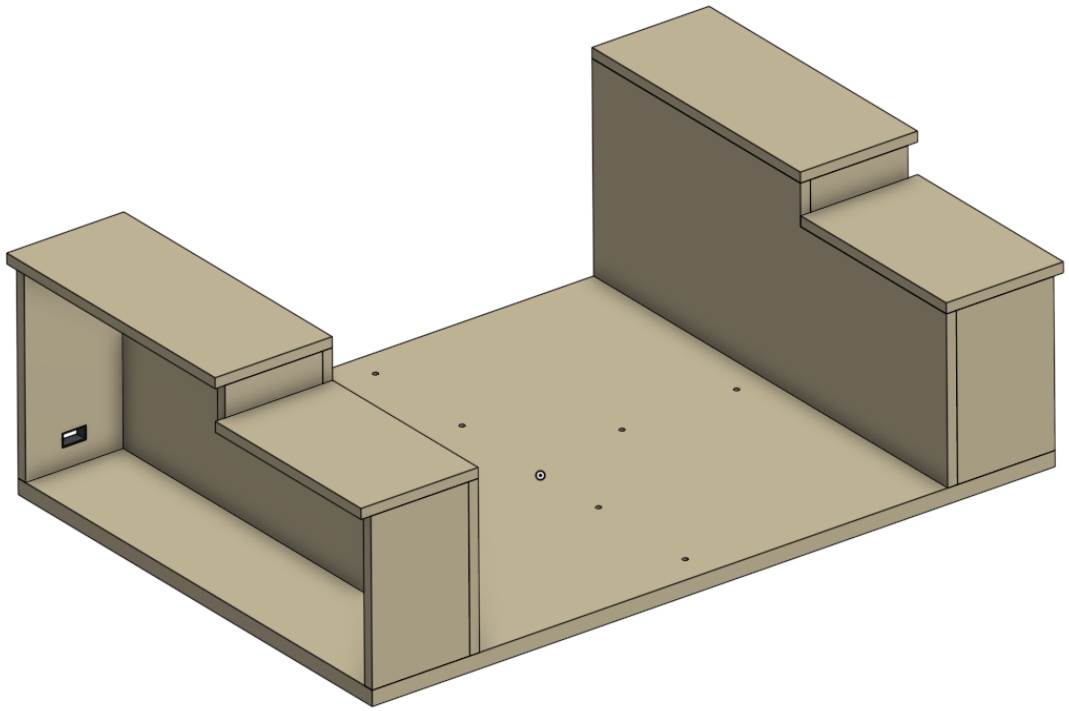


11. Add front and back panels and attach using screws and glue. 0.5” Plywood

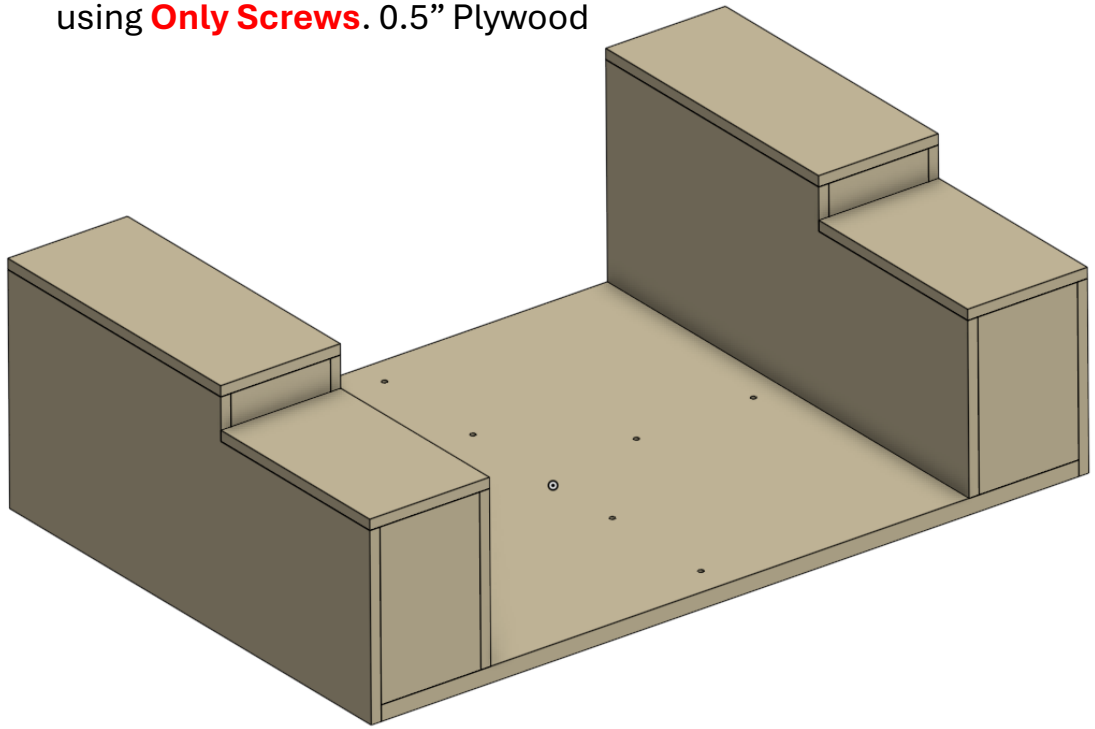


# Chair Assembly Structure

12. Add top panels and attach using screws and glue. 0.5” Plywood

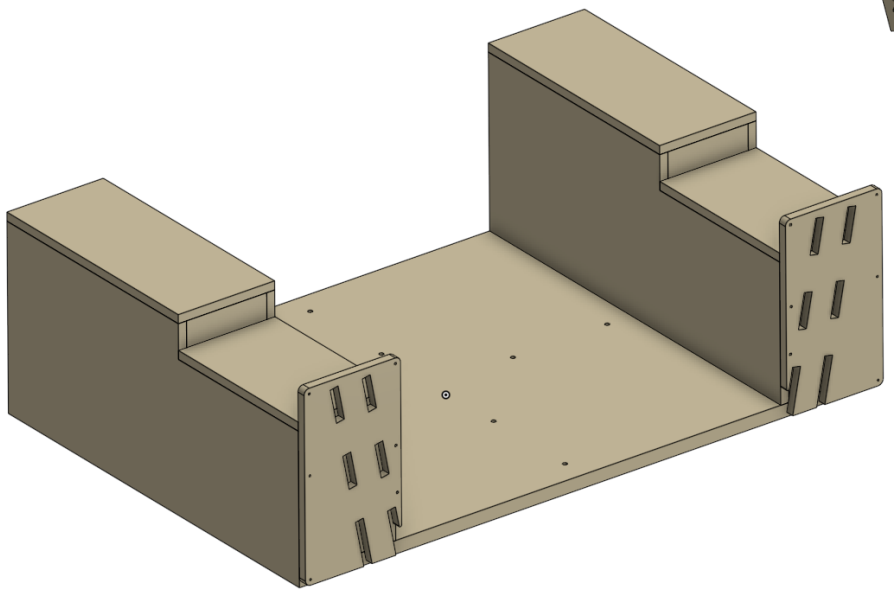


13. Add outer side panels and attach using **Only Screws**. 0.5” Plywood

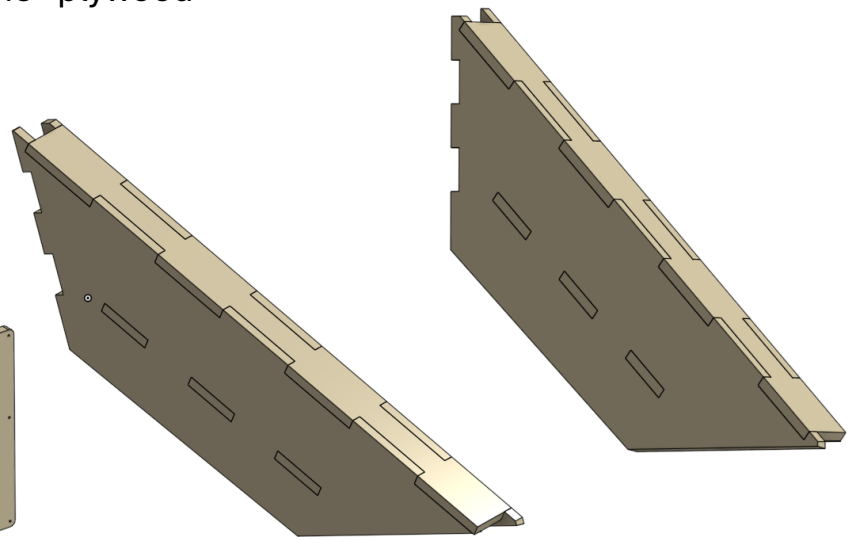


# Chair Assembly Structure

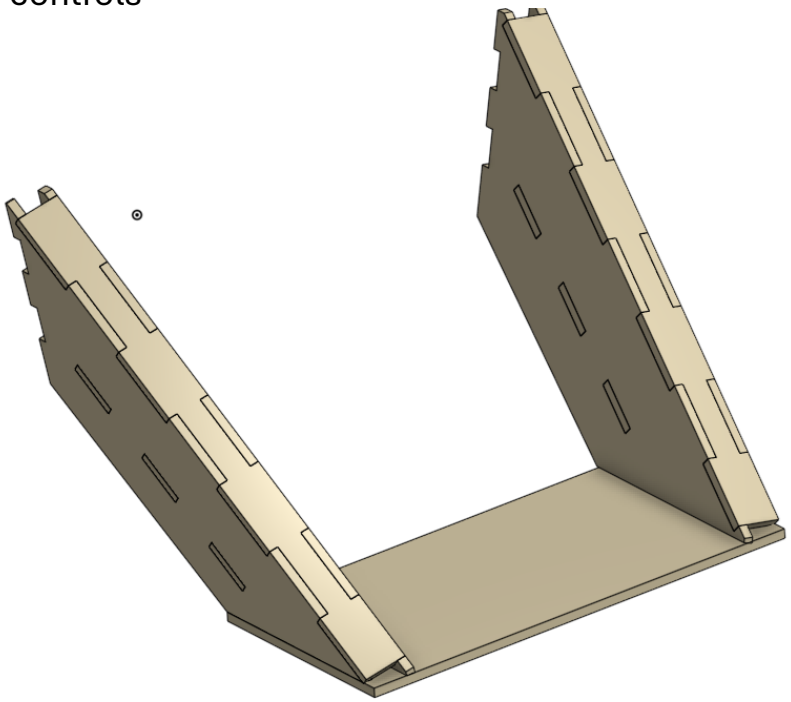
14. Add foot support attachment plates to assembly using **Only Screws**



15. Assemble foot support side structures as shown in CAD assembly. 0.5” plywood

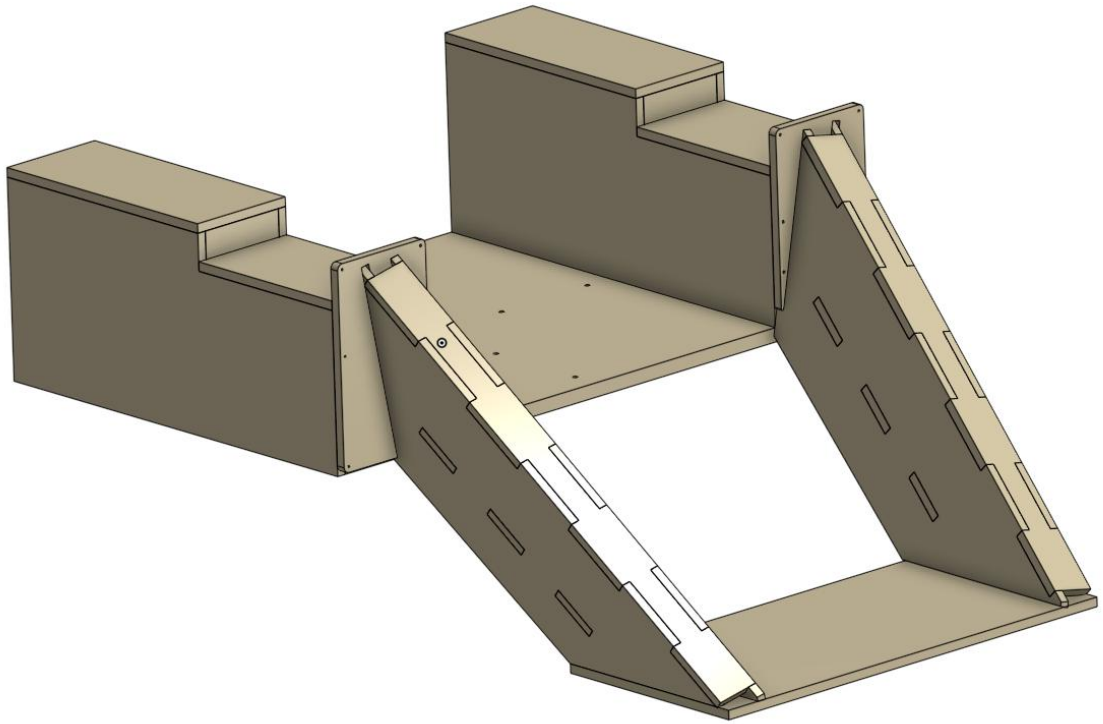


16. Add foot support plate. May need to be customized for your specific controls

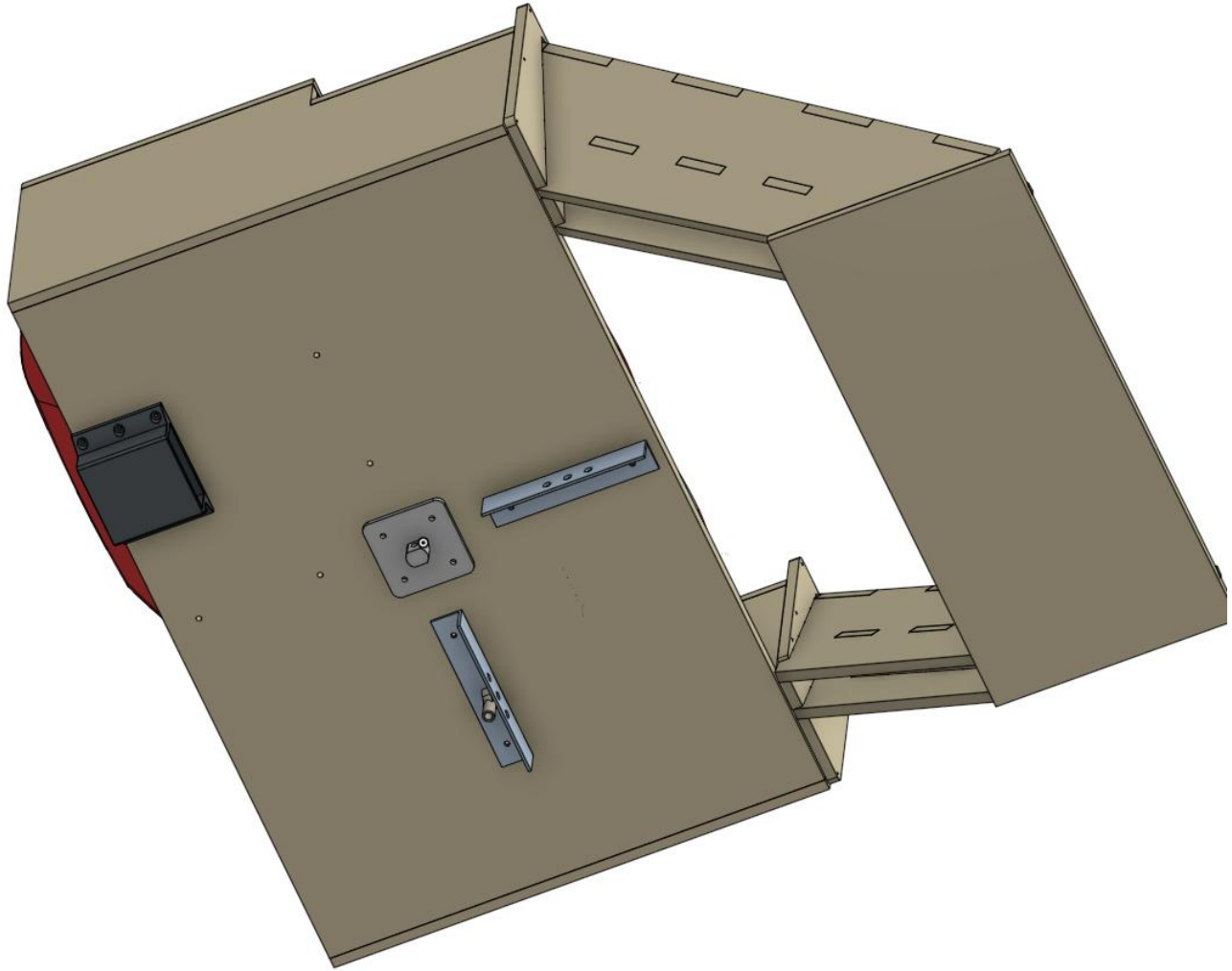


# Chair Assembly Structure

17. Attach to foot support attachment place using glue

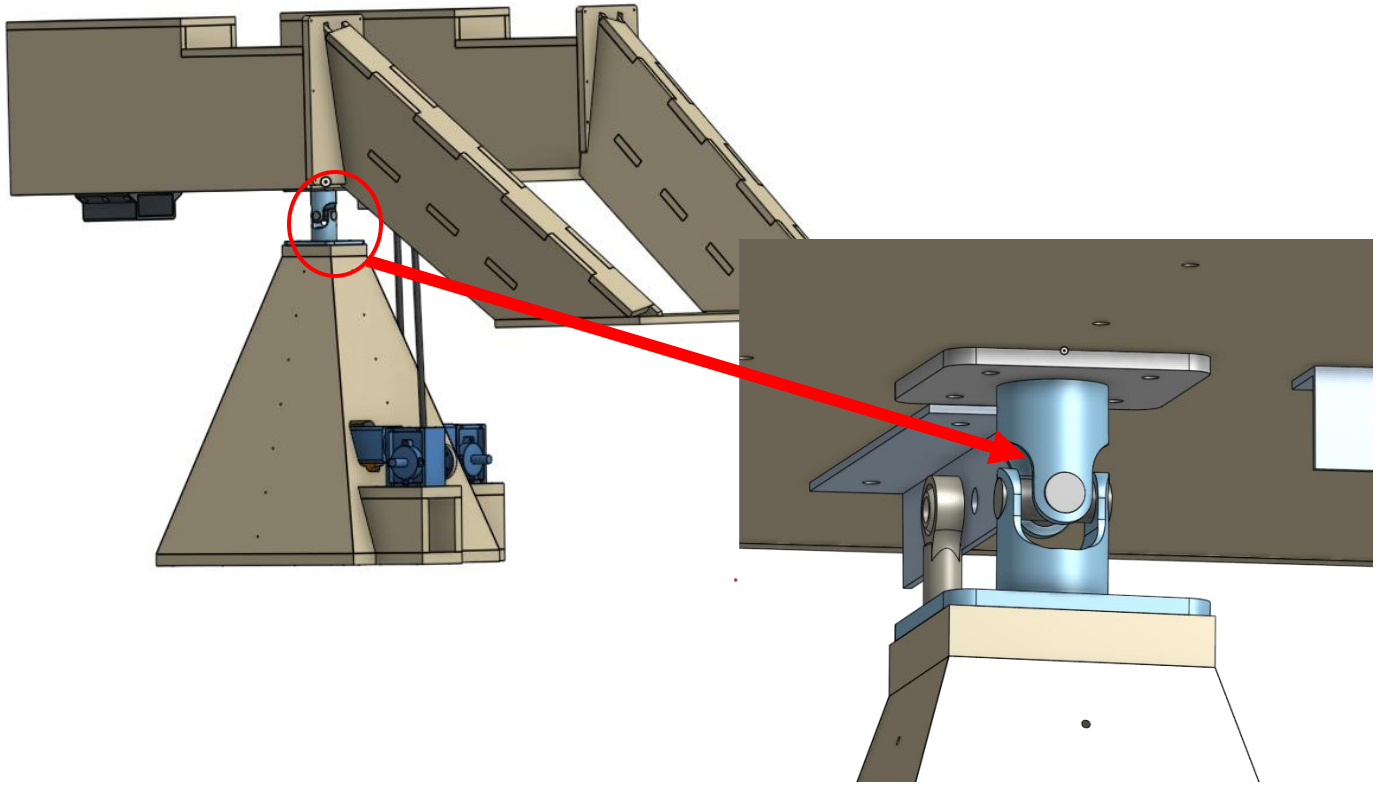


18. Add bottom auxiliary parts: Angle brackets and universal joint mounts

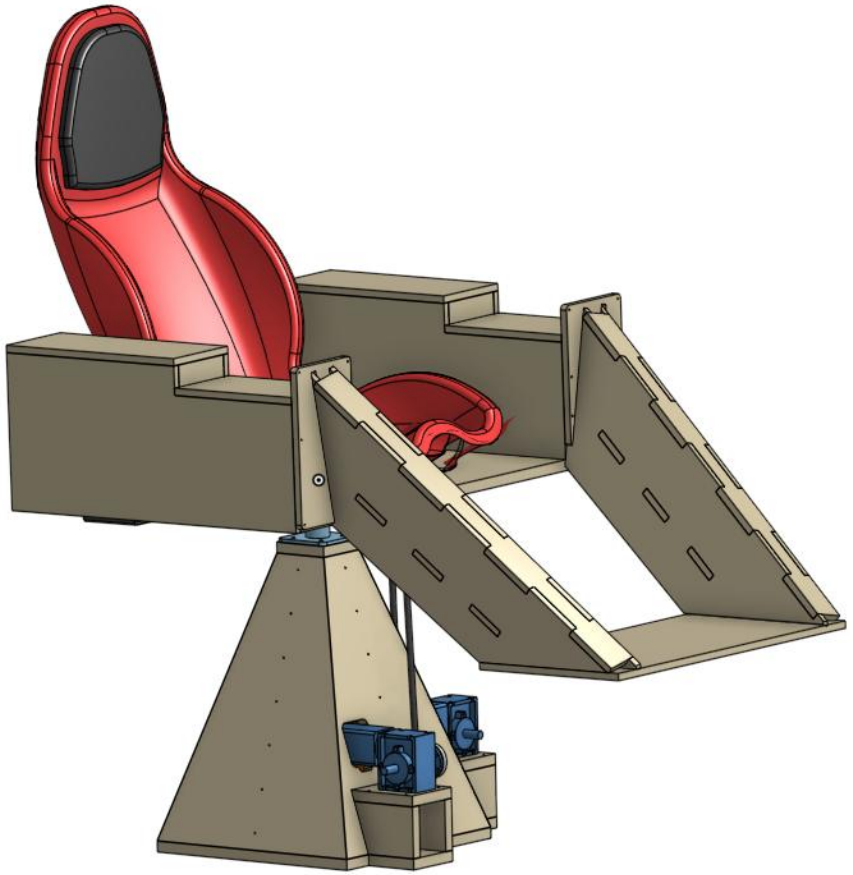


# Base / Chair Attachment

19. Attach upper and lower assemblies using universal joint

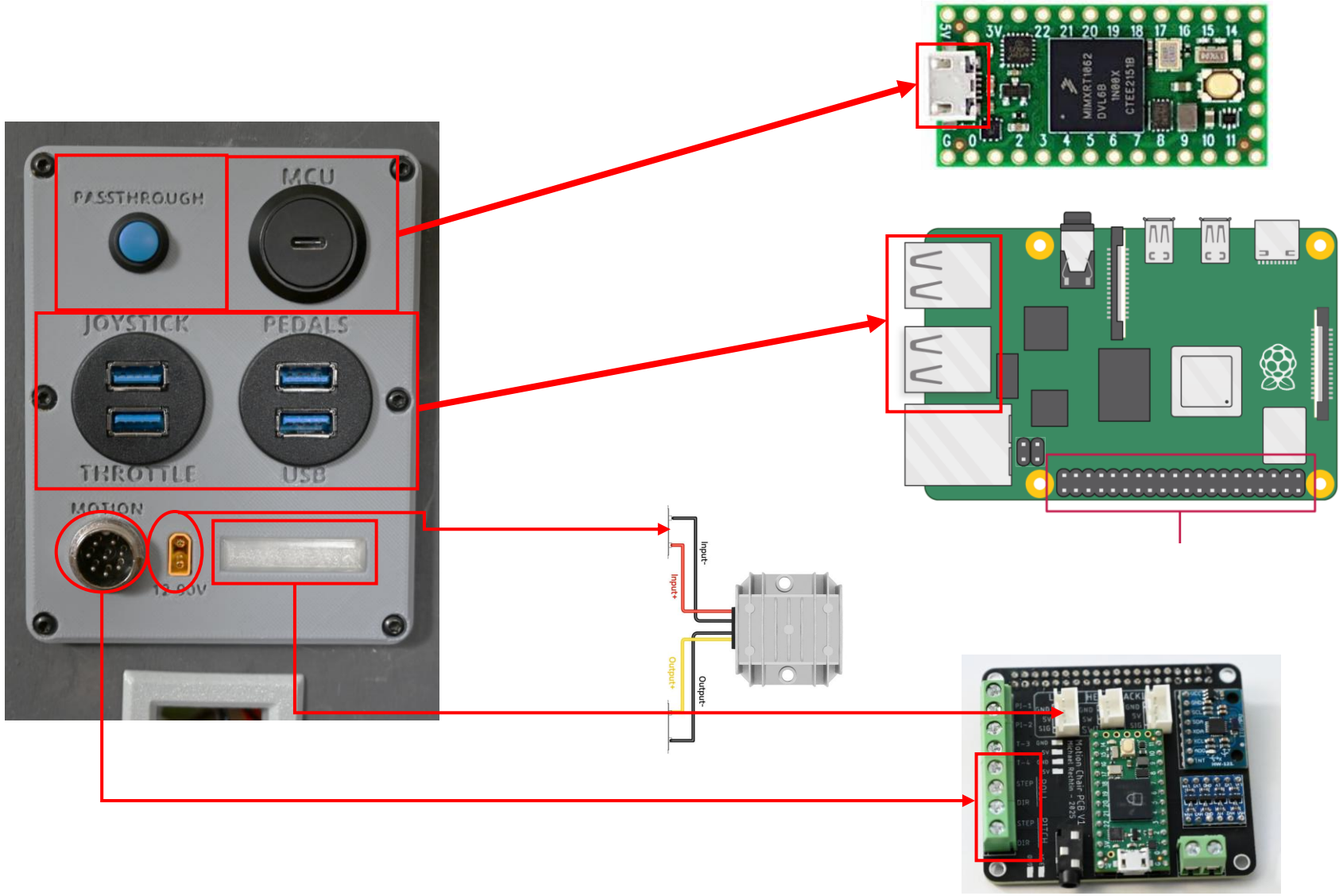


20. Add chair



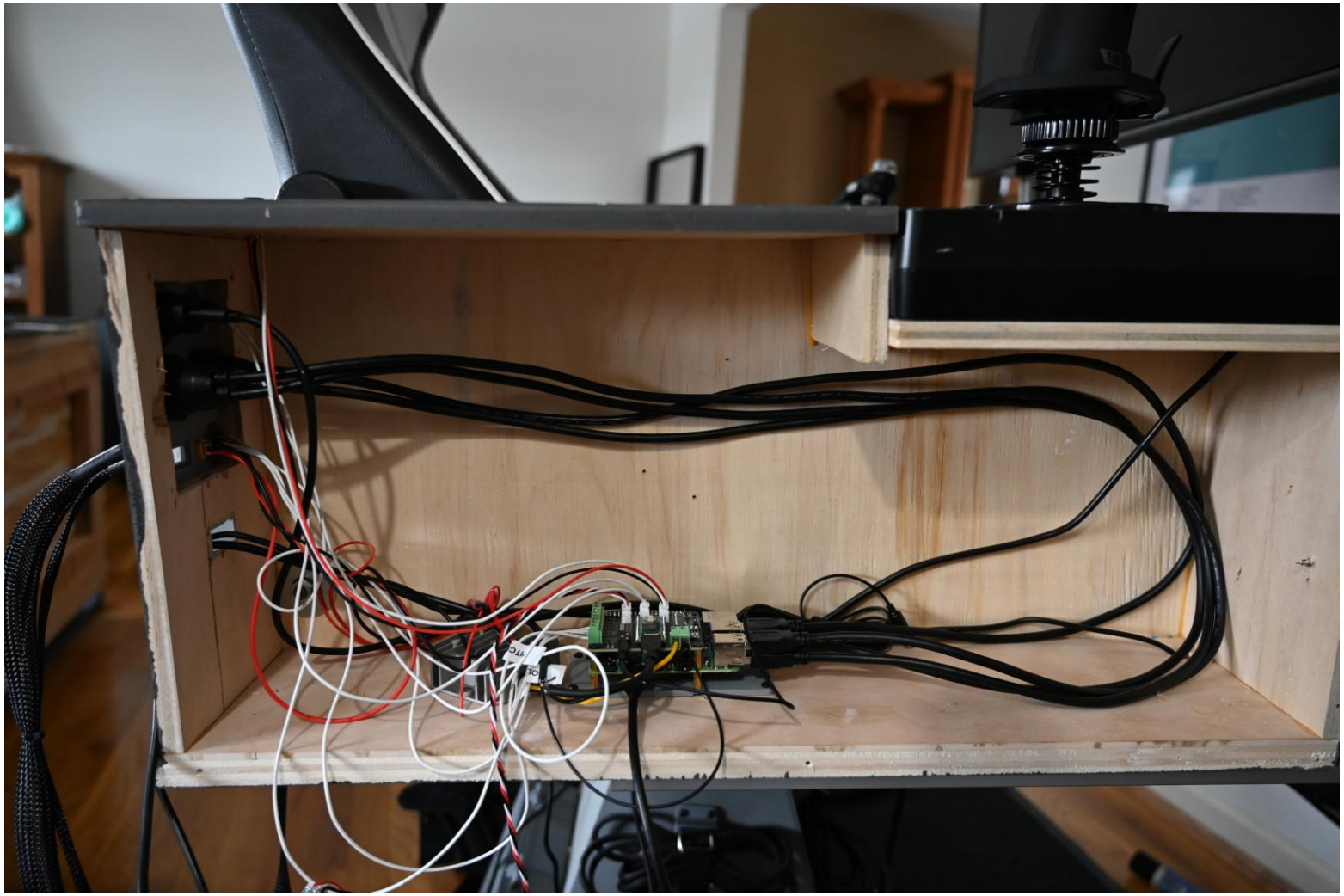
# Electronics

# Back Panel Wiring



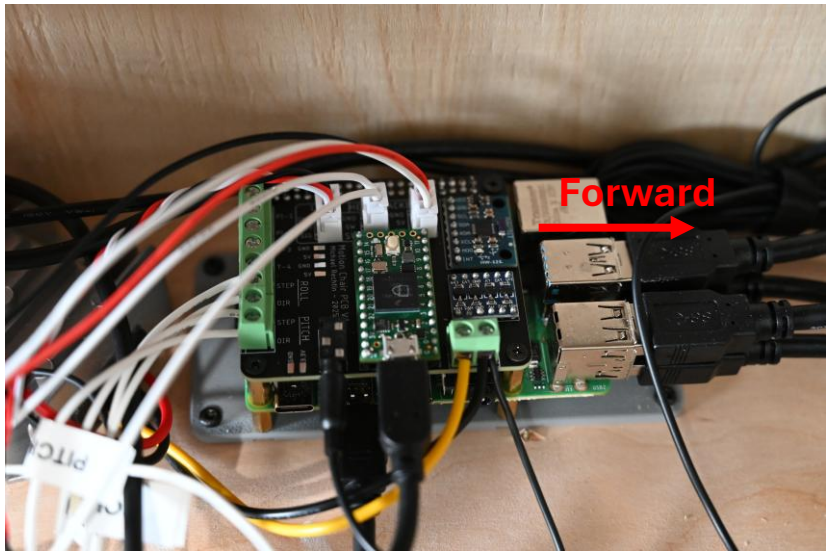
# Side Compartment Wiring

- All wiring is housed in the right-side armrest. Wire management is optional!



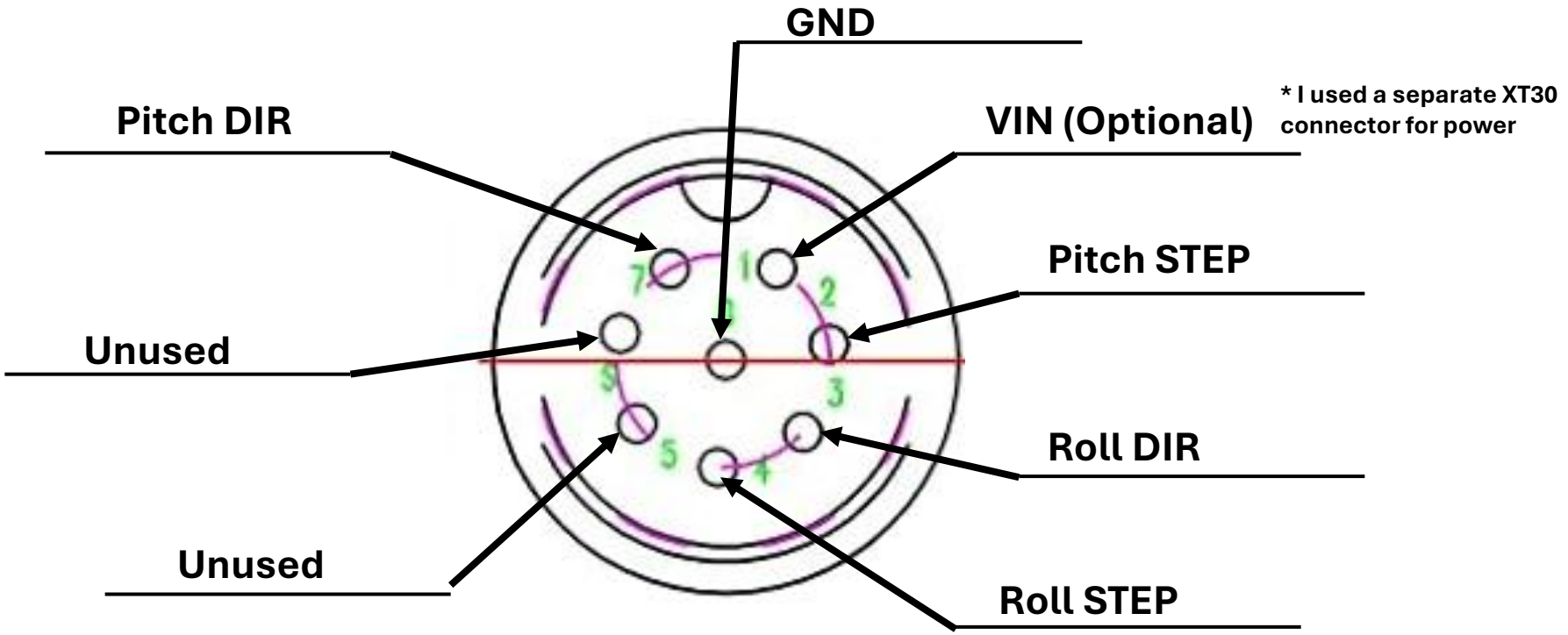
**IMPORTANT**

**MPU 6050 must be in the orientation shown below for IMU based chair calibration.**

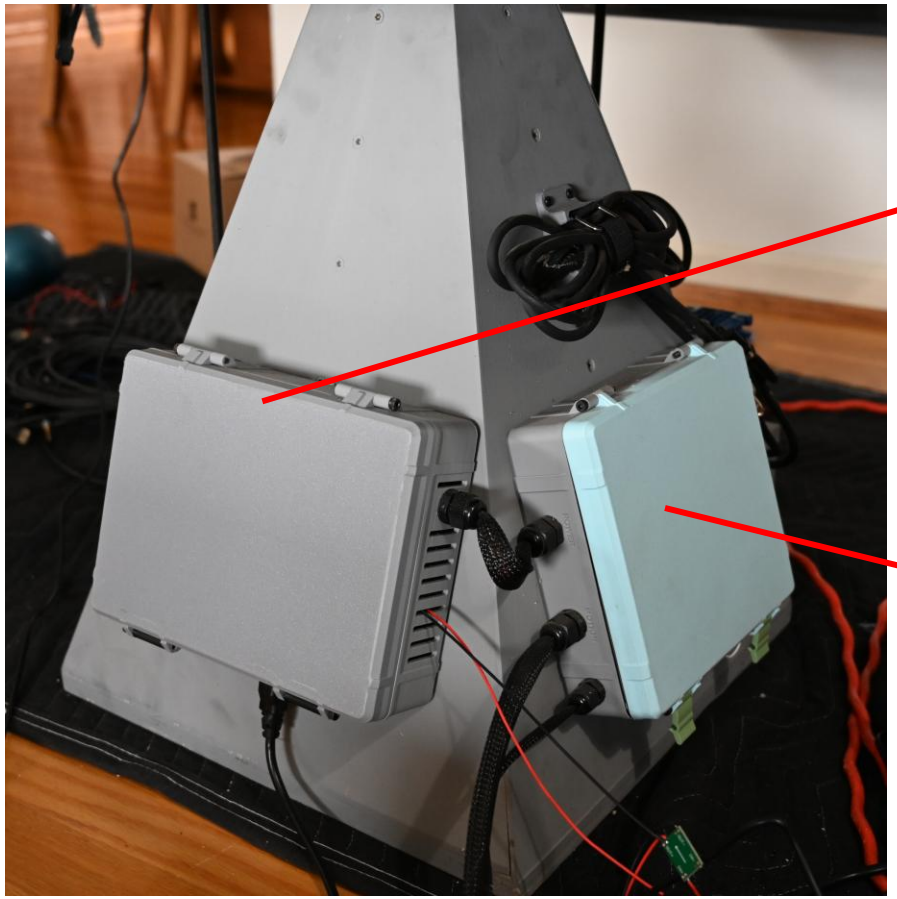


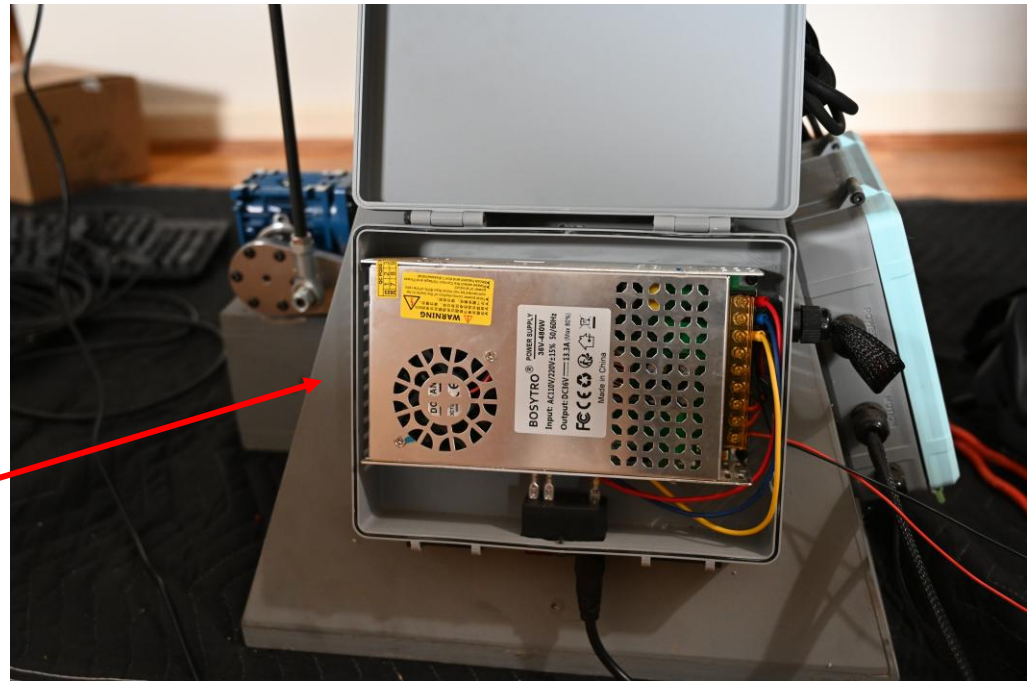
# Back Panel Connector Pinout

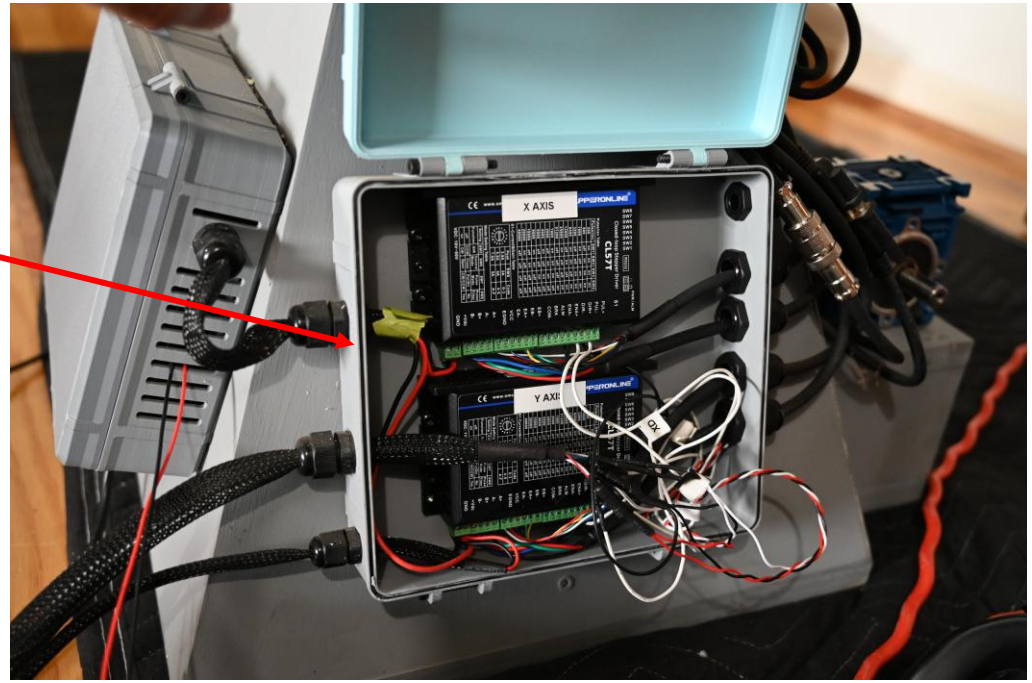
- Adding connectors to the harness from the base to the chair is helpful if the chair needs to be disassembled for moving.
- Pinout can be customized but MUST be consistent between the base and chair to create proper connections

# Motion Base Wiring

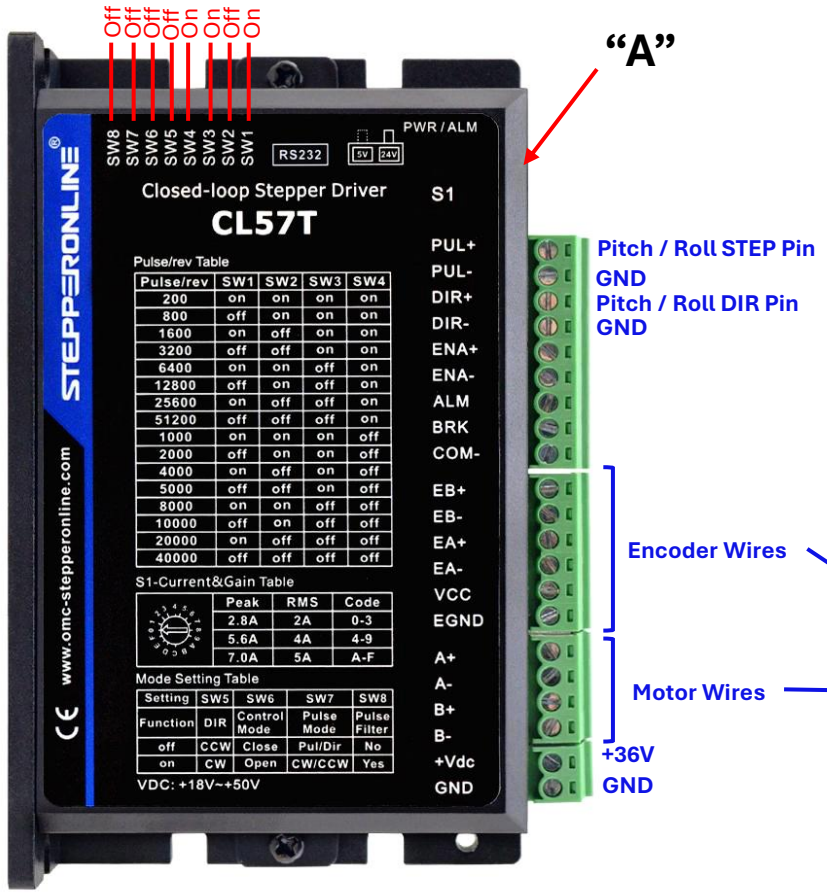




# Motion Base Wiring

CL57T Settings using Switches:
Current: “A” 7.0 Amp
Pulse/Rev: 1600



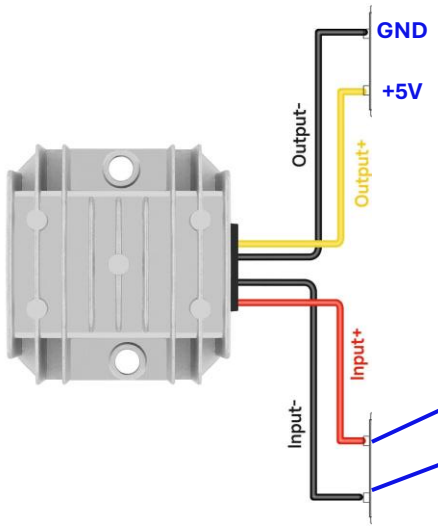

“A”

Pitch / Roll STEP Pin
GND
Pitch / Roll DIR Pin
GND

Encoder Wires

Motor Wires

+36V
GND





Stepper Motor / Driver wiring details: https://www.omc-stepperonline.com/closed-loop-stepper-driver-v4-1-0-8-0a-24-48vdc-for-nema-17-23-24-stepper-motor-cl57t-v41

# Wiring Pinouts

Passthrough Switch

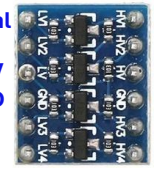
SDA: Connect to MPU SDA
SCL: Connect to MPU SCL

TX: Connect to PI GPIO 15
RX: Connect to PI GPIO 14
LED 3.3V Signal
Roll Axis: DIR
Roll Axis: STEP
Pitch Axis: DIR
Pitch Axis: STEP

5V VIN

GND


3.3V
GND
SCL: Connect to Teensy Pin 19
SDA: Connect to Teensy Pin 18

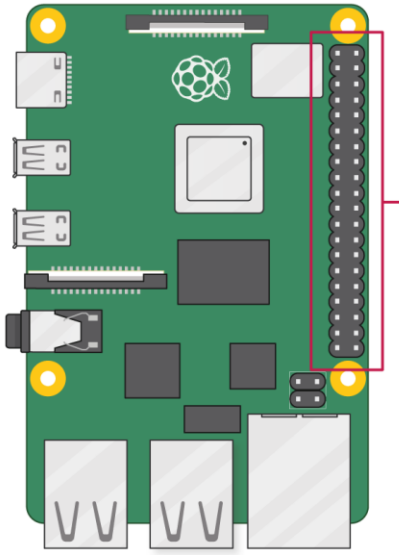


LED 3.3V Signal

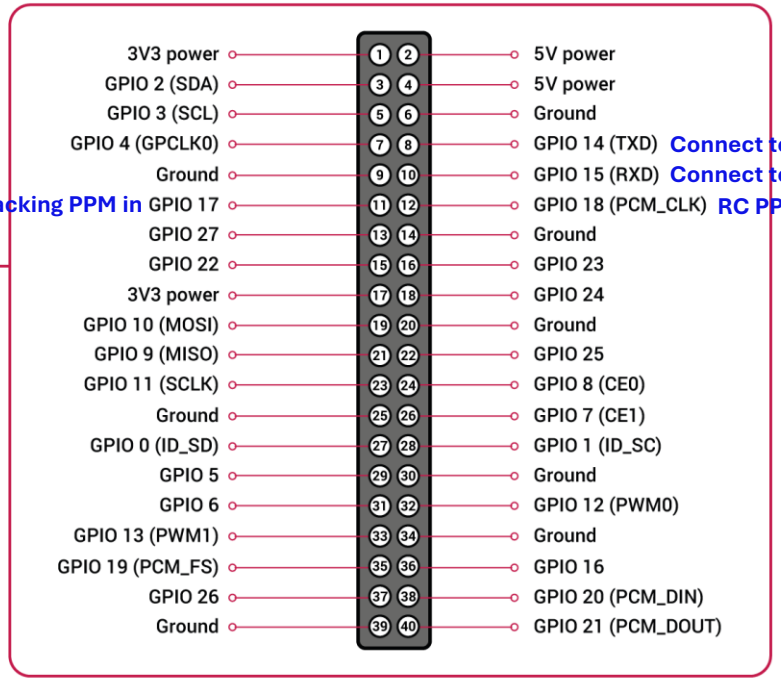
LED 5V Signal

3.3V
GND

5V
GND

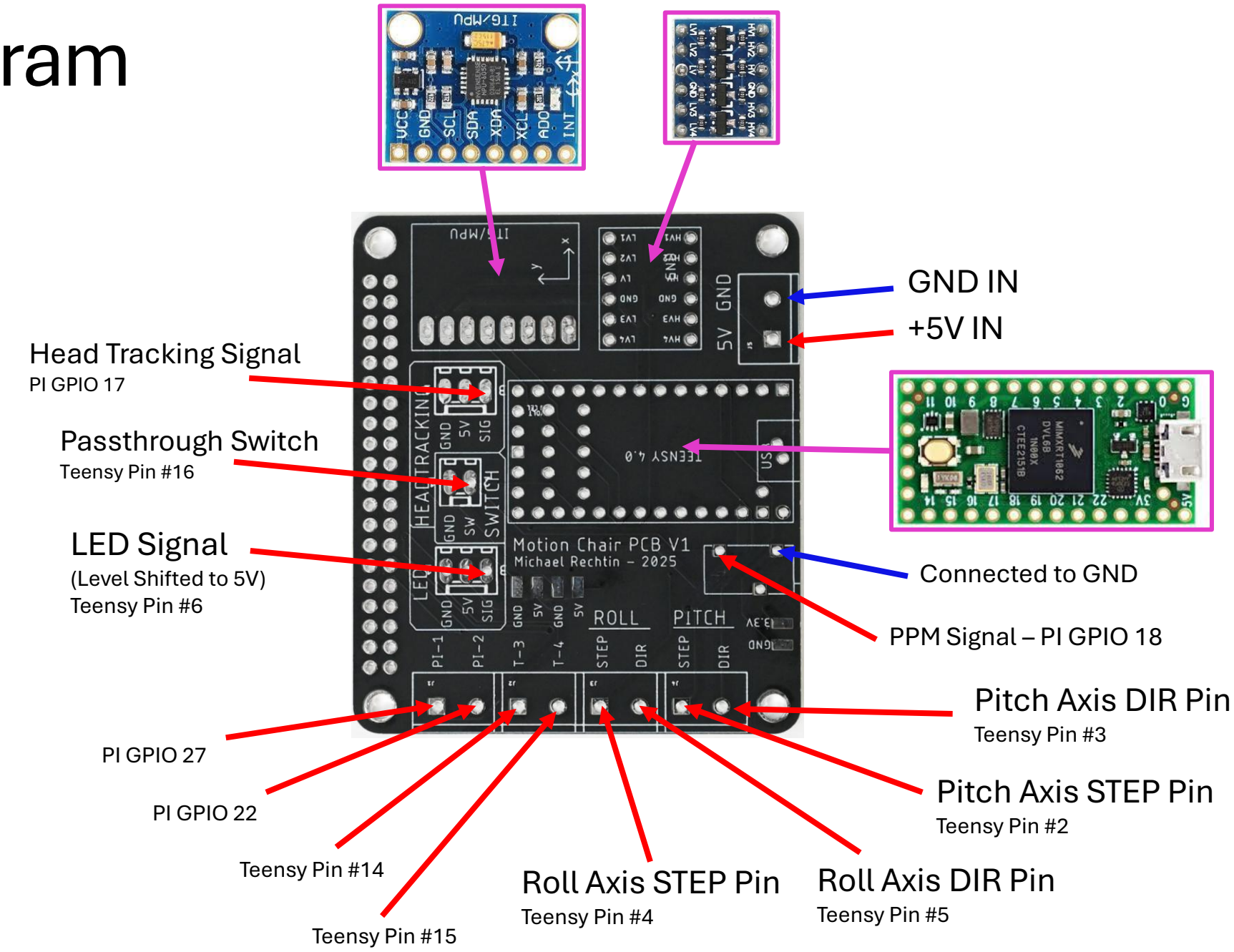


Headtracking PPM in




Connect to Teensy Pin 7
Connect to Teensy Pin 8
RC PPM Out

# PCB Diagram


GND IN
+5V IN
Head Tracking Signal
PI GPIO 17
Passthrough Switch
Teensy Pin #16
LED Signal
(Level Shifted to 5V)
Teensy Pin #6
Connected to GND
PPM Signal – PI GPIO 18
Pitch Axis DIR Pin
Teensy Pin #3
Pitch Axis STEP Pin
Teensy Pin #2
Roll Axis DIR Pin
Teensy Pin #5
Roll Axis STEP Pin
Teensy Pin #4
PI GPIO 27
PI GPIO 22
Teensy Pin #14
Teensy Pin #15
Motion Chair PCB V1
Michael Rechtin - 2025
ROLL
PITCH
5V GND
HEAD TRACKING
SWITCH
LED

# Flight Controls

- Most USB based controls should work but the ones listed below are options I have tested

Logitech G X56 H.O.T.A.S

https://amzn.to/4g3T0MK

Thrustmaster TFRP Rudder Pedals


https://amzn.to/4hnNeq2

# Software Setup - Chair

All code available at: https://github.com/MichaelRechtin/MotionSim

# Teensy 4.0 Code

C++ code running on the Teensy must be uploaded using the Arduino IDE. The teensy takes serial inputs from either the the Pi (passthrough mode) or an external computer to control the stepper motors for pitch / roll

IMU_Motion_Control.ino

```
#include <AccelStepper.h>
#include <Adafruit_NeoPixel.h>

// Define the stepper motor connections
#define XDIR_PIN 3
#define XSTEP_PIN 2
#define YDIR_PIN 5
#define YSTEP_PIN 4
#define PASSTHROUGH_PIN 16
#define LED_PIN 6
#define NUM_LEDS 5
```

Adjustable parameters depending on pinout. Default should work with PCB

```
// Define the maximum speed and acceleration
#define MAX_SPEED 10000.0 // Steps per second
#define ACCELERATION 10000.0 // Steps per second^2
```

Adjustable motion parameters. If motors skip steps these values may need to be lowered

```
int x_zero=0;
int y_zero=-4;

int x_pos=0;
int y_pos=0;
int steps_per_rev=3200;
```

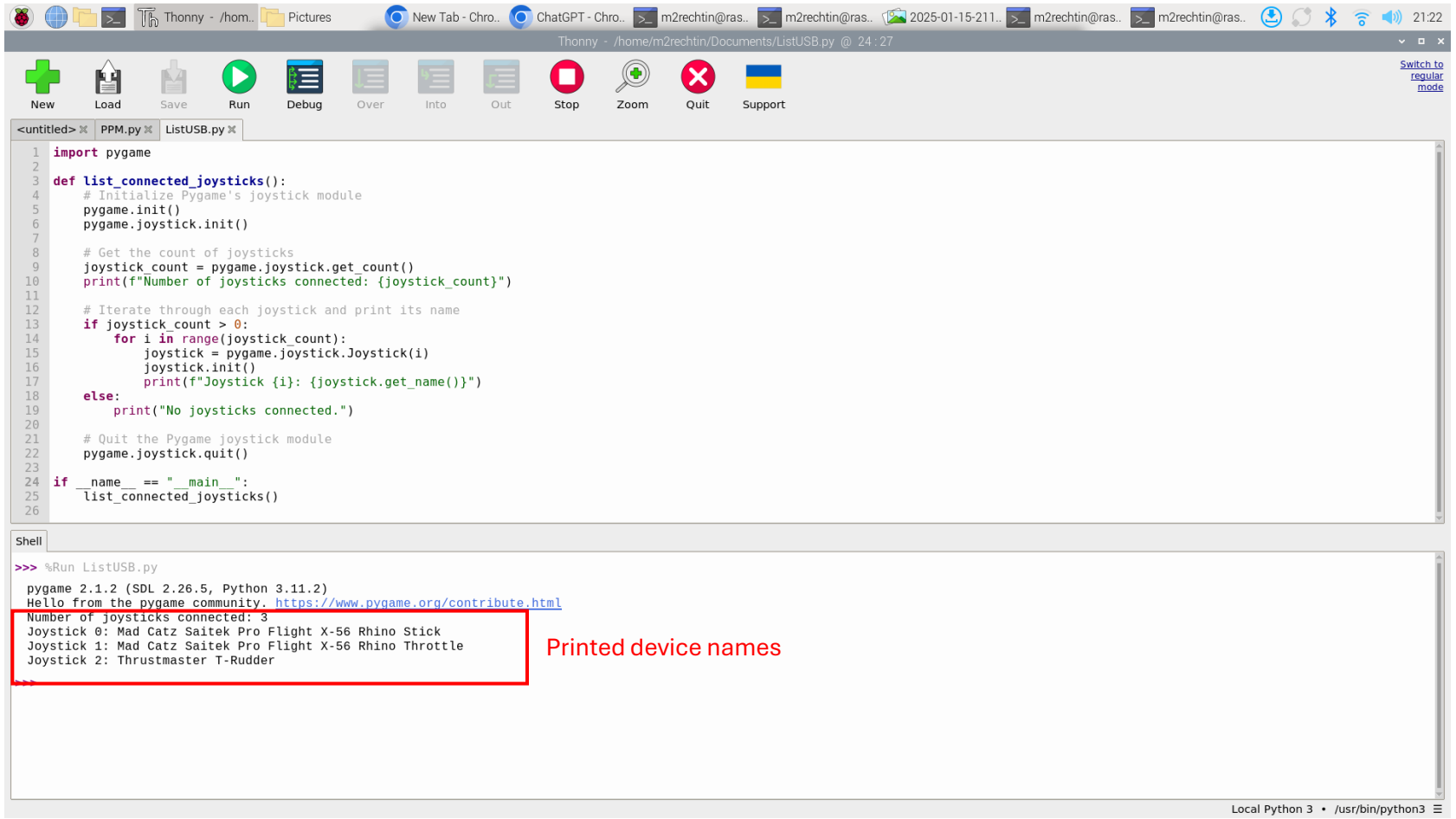
Adjust the “Zero” position that the chair calls home

Adjust is microstepping differently

# Raspberry Pi: Identify Flight Controls Names ListUSB.py

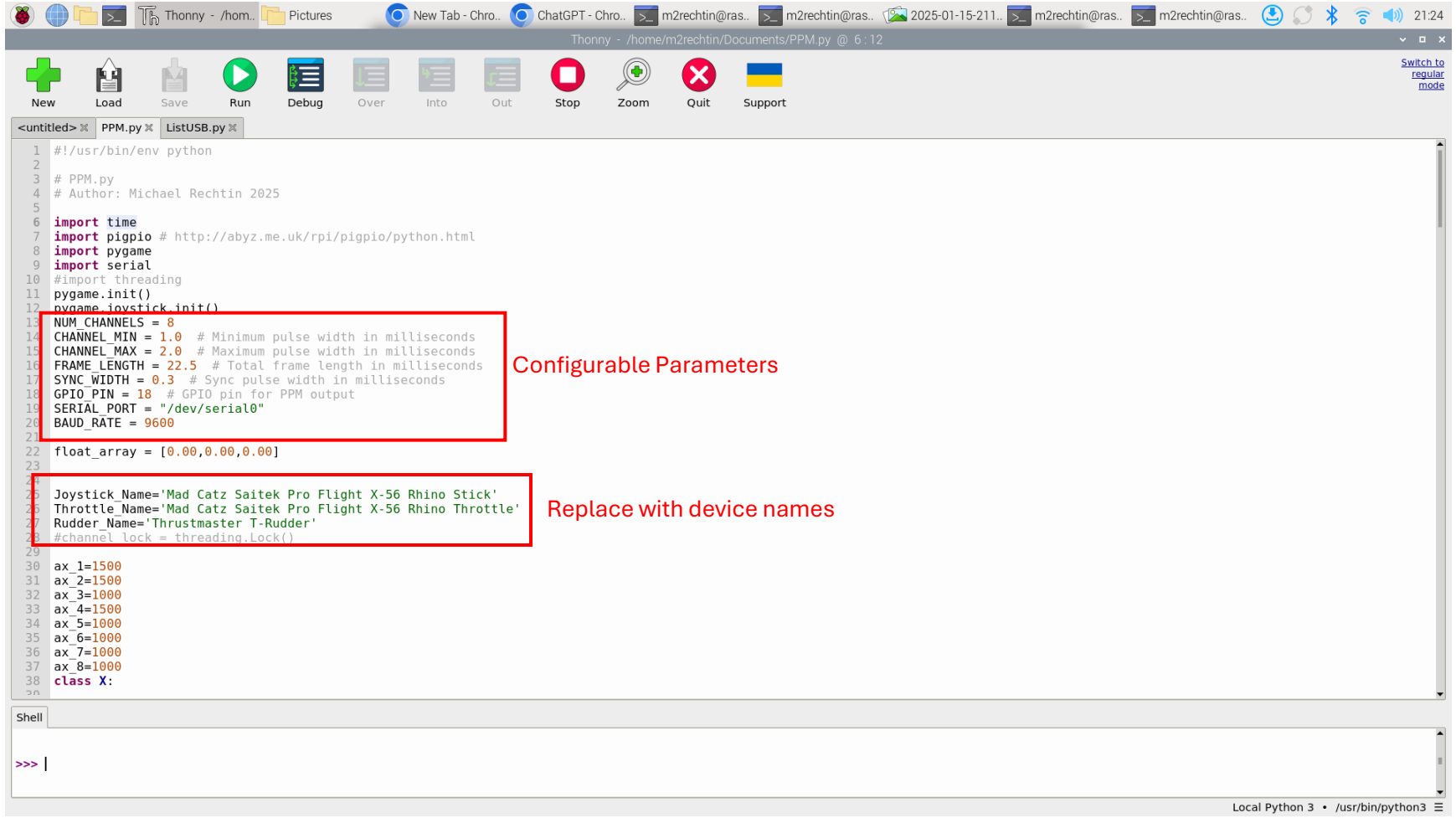
If you are using different USB controls you must first find the names of your USB devices.

1. Connect USB controls to the Raspberry Pi through the back panel or the USB ports directly.
2. Run ListUSB.py and it will print out the names of all detected devices



# Raspberry Pi: Insert Flight Controls Names – PPM.py

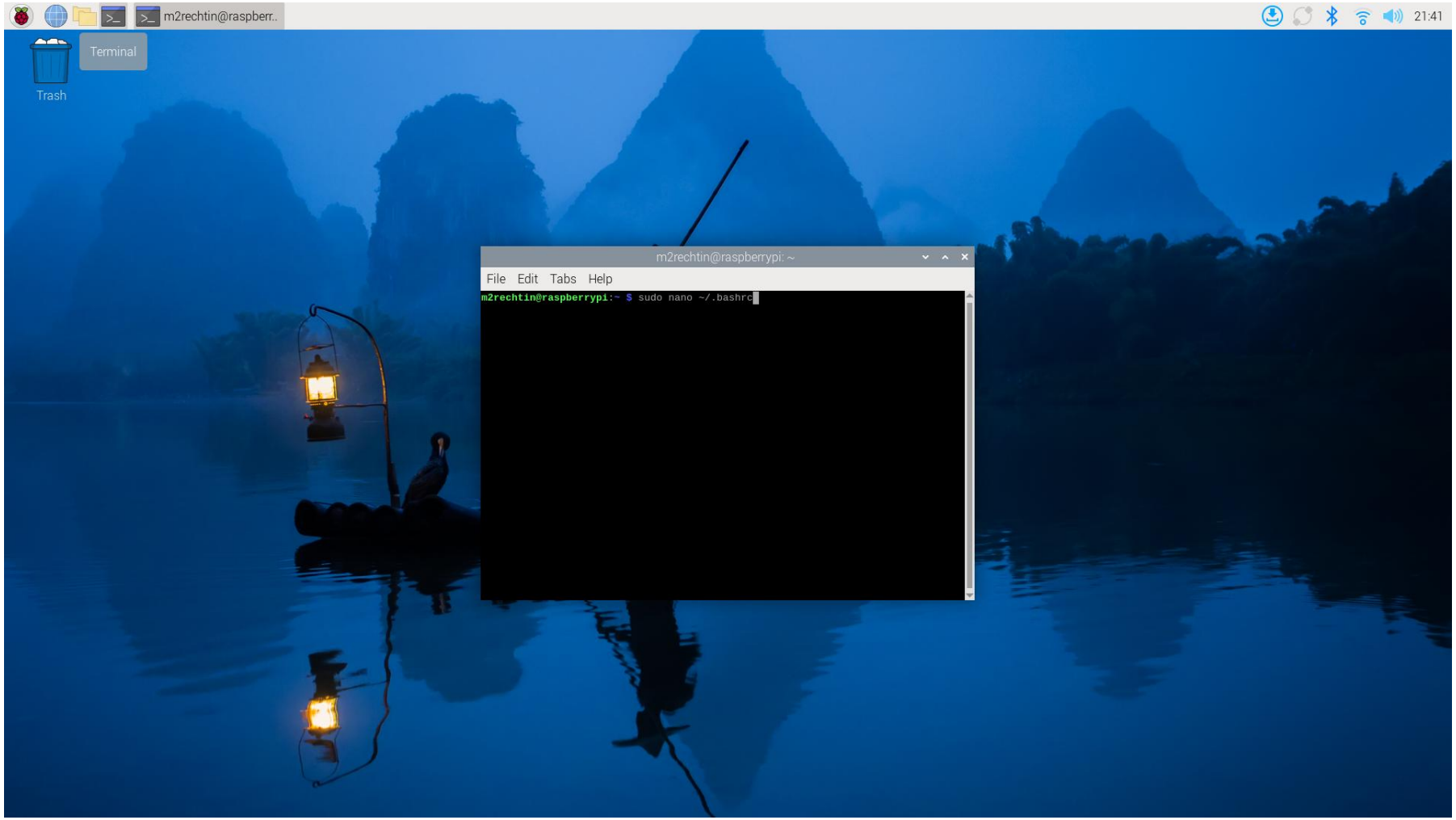
1. Open PPM.py
2. Replace device names with the outputs from ListUSB.py
3. Optionally configure any changes needed for RC applications



# Raspberry Pi: Auto-run script

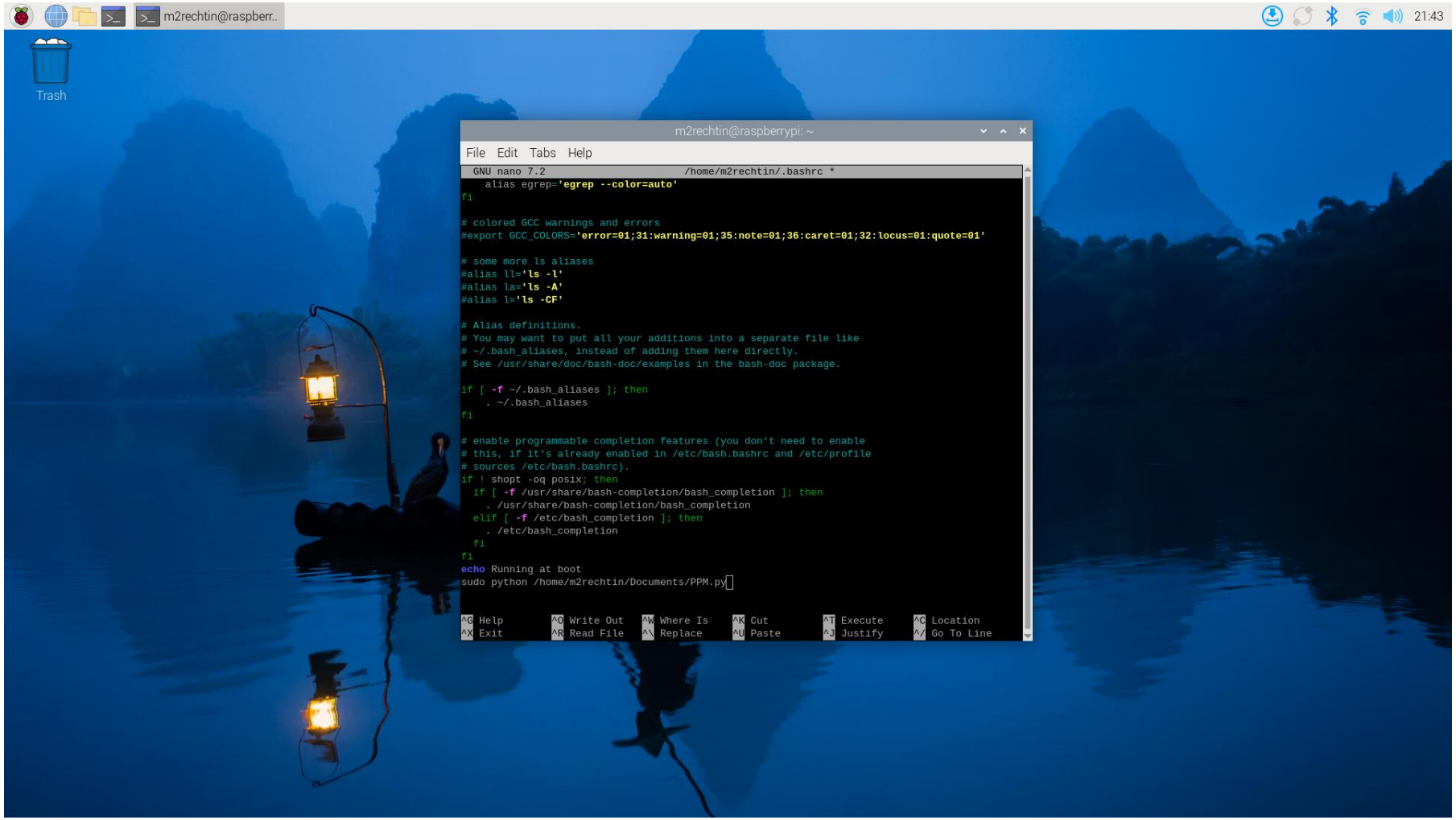
To operate properly, PPM.py must autorun when Pi is powered on.

1. Open terminal
2. Open bashrc file (using command below)



# Raspberry Pi: Auto-run script

1. Add lines to bottom of file. As shown below
2. Ctrl+X and then Y to save and exit



# Software Setup - PC

All code available at: https://github.com/MichaelRechtin/MotionSim

# Overview

The chair receives movement commands from an external PC. These movements can come from a simulation game (MSFS 2020, MSFS 2024, WarThunder currently supported) or from an RC plane (Ardupilot)

1. Connect PC to chair using USB Cable
2. Run python script


Connect to PC

**If using Chair for "Passthrough Mode" or flying RC planes, Connect USB joystick devices here**

**If using the chair for flying simulator games, connect USB joystick devices to the PC running the game**

# Microsoft Flight Sim 2020/2024

temp.py | MSFS_Connect.py | War_Thunder_Connect.py | Send_Data.py

```python
# -*- coding: utf-8 -*-
"""
Created on Sat Oct 26 18:48:44 2024

@author: micha
"""

from SimConnect import SimConnect, AircraftRequests
import time
from pprint import pprint
import serial
import serial.tools.list_ports
import math


port = "COM8"  # Update this to your Arduino's serial port
baud_rate = 9600
ser=serial.Serial(port, baud_rate, timeout=1)
time.sleep(2)

vector = [0, 0]

sm = SimConnect()
# Create a request object to pull data from MSFS
aq = AircraftRequests(sm, _time=10)

while True:
    try:
        # Retrieve pitch and roll data

        pitch = aq.get("PLANE_PITCH_DEGREES")
        roll = aq.get("PLANE_BANK_DEGREES")

        # Display pitch and roll
        #print(f"Pitch: {pitch:.2f} degrees, Roll: {roll:.2f} degrees")

        # Pause for a short interval to reduce load
        #time.sleep(0.1)
        vector[0]=roll*(-180/math.pi)
        vector[1]=pitch*(-180/math.pi)
        vector_str = f"{vector[0]},{vector[1]}\n"
        ser.write(vector_str.encode('utf-8'))
        print(vector)


    except KeyboardInterrupt:
        ser.close()
        print('Closing')
        break

    except:
        next
```

Download packages with Pip Installer

Replace with correct COM port

SimConnect:
https://pypi.org/project/SimConnect/
https://github.com/odwdinc/Python-SimConnect

# WarThunder

temp.py | MSFS_Connect.py | War_Thunder_Connect.py | Send_Data.py

```python
# -*- coding: utf-8 -*-
"""
Created on Wed Jul 31 00:17:30 2024

@author: micha
"""

from WarThunder import telemetry
from WarThunder import mapinfo
from pprint import pprint
import serial
import serial.tools.list_ports
import time

port = "COM14"  # Update this to your Arduino's serial port
baud_rate = 9600
ser=serial.Serial(port, baud_rate, timeout=1)
time.sleep(2)

vector = [0, 0]

while True:
    try:
        telem = telemetry.TelemInterface()

        while not telem.get_telemetry():
            pass

        dat=telem.basic_telemetry
        pitch_ang=dat['pitch']
        roll_ang=dat['roll']

        vector[0]=roll_ang
        vector[1]=pitch_ang
        vector_str = f"{vector[0]},{vector[1]}\n"
        ser.write(vector_str.encode('utf-8'))
        print(vector)


    except KeyboardInterrupt:
        print('Closing')
        break

```

Download packages with Pip Installer

Replace with correct COM port

WarThunder Python Package:
https://pypi.org/project/WarThunder/2.0.4/

# Ardupilot RC Plane

temp.py | MSFS_Connect.py | War_Thunder_Connect.py | Ardupilot_Connect.py

```python
# -*- coding: utf-8 -*-
"""
Created on Tue Feb  6 16:56:49 2024

@author: micha
"""

from pymavlink import mavutil
import struct
import math
import serial
import serial.tools.list_ports
ports = serial.tools.list_ports.comports()


import serial
import time
port = "COM5"  # Update this to your Arduino's serial port
baud_rate = 9600
ser=serial.Serial(port, baud_rate, timeout=1)
time.sleep(2)

vector = [0, 0]

def float_to_bytes(f):
    return struct.pack('f', f)

master = mavutil.mavlink_connection('udpin:localhost:14445')

timeout = 5 # maximum seconds to wait for a message
if not master.wait_heartbeat(timeout=timeout):
    ser.close()
    master.close()
    raise TimeoutError(f"No heartbeat within {timeout=}A seconds!")

while True:
    msg = master.recv_match(type='ATTITUDE', blocking=True).to_dict()
    #roll_bytes = float_to_bytes(msg['roll']*(180/math.pi))
    #pitch_bytes = float_to_bytes(msg['pitch']*(180/math.pi))
    #heave_bytes = float_to_bytes(0)
    #yaw_bytes = float_to_bytes(0)
    #sway_bytes = float_to_bytes(0)
    #surge_bytes = float_to_bytes(0)
    #data_send = roll_bytes + pitch_bytes+heave_bytes+yaw_bytes+sway_bytes+surge_bytes
    vector[0]=msg['roll']*(180/math.pi)
    vector[1]=msg['pitch']*(180/math.pi)
    vector_str = f"{vector[0]},{vector[1]}\n"
    ser.write(vector_str.encode('utf-8'))
    print(vector)
```

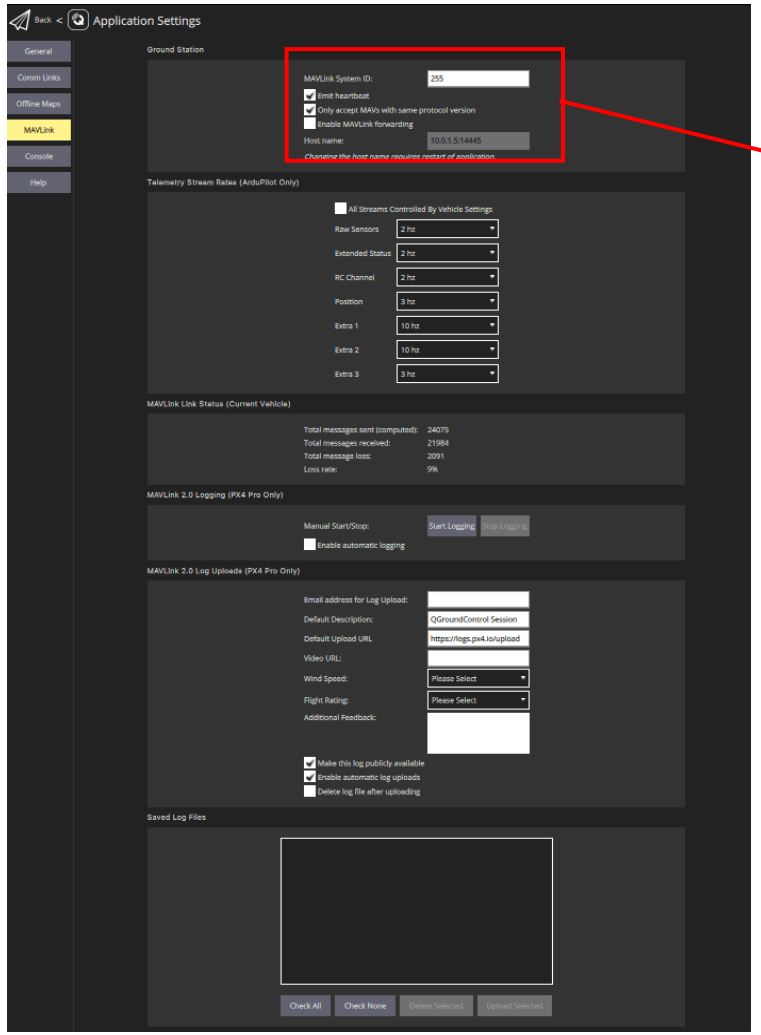
Download packages with Pip Installer

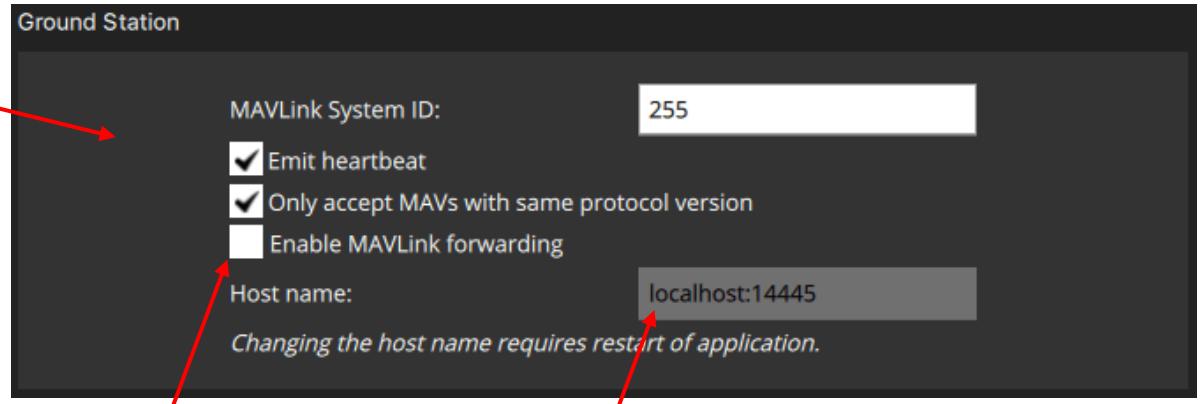
Replace with correct COM port

Replace with correct UDP port. See next Slide

# Ardupilot RC Plane

- QGroundControl must be configured to forward the MAVLINK packets to a UDP port
- Full Documentation: https://docs.qgroundcontrol.com/Stable_V4.3/en/qgc-user-guide/settings_view/mavlink.html




**Check This**

**Port Name**

# Buttkicker for Sim Games

# Bill of Materials

| Item | Qty | Link (May be Affiliate) | Notes |
|---|---|---|---|
| **Electronics** | | | |
| Raspberry Pi 4 | 1 | https://amzn.to/4gDeqRJ | |
| Teensy 4.0 | 1 | https://amzn.to/4h26auC | |
| MPU6050 | 1 | https://amzn.to/3DHh3n9 | |
| Level Shifter (Only if using LED) | 1 | https://amzn.to/3DGJHoi | |
| 5V LED Strip | 1 | https://amzn.to/4j0shDi | |
| JST Connectors (Optional but makes wiring easier) | 1 | https://amzn.to/4j26Rpf | |
| Terminal Blocks (Optional but makes wiring easier) | 1 | https://amzn.to/4gACqVC | |
| Custom PCB (Limited Stock Available) | 1 | | |
| USB Flight Stick / Throttle (Most should work) | 1 | https://amzn.to/3BJ2AXf | |
| USB Rudder Pedals (Most should work) | 1 | https://amzn.to/3BWKVLD | |
| 20:1 NEMA 21 Gearbox | 2 | https://amzn.to/4iXhycP | |
| NEMA 21 Closed Loop Stepper Motor Kit | 2 | https://www.omc-stepperonline.com/ts-series-3-0-nm-424-92oz-in-1-axis-closed-loop-stepper-cnc-kit-nema-23-motor-driver-1-cl57t-s30a-v41 | |
| NEMA 21 Motor Shaft Adapter | 2 | https://amzn.to/3PhDD8q | |
| Motor Shaft Hub | 2 | https://amzn.to/3W463Gx | Many versions exist of this hub. Steel or aluminum should would |
| Gearbox Double Shaft | 2 | https://amzn.to/40mkeJW | |
| | | | |
| | | | |
| **Chair Structure** | | | |
| 3/4" Plywood | | Local Hardware Store | |
| 1/2" Plywood | | Local Hardware Store | |
| Screws | | Local Hardware Store | |
| Heim Joints | | https://amzn.to/4jpR4Rm | 3/8-24 Female Thread Right Hand |
| Connecting Rod | | https://www.mcmaster.com/98842A031/ | 3/8 x 24 - 2 sections at least 400mm |
| Angle Iron | | Local Hardware Store | |
| T Nuts | | https://amzn.to/42f8U3v | |
| Pack of spacers for Heim Joints | | https://amzn.to/4g4bUCX | 1/2" OD x 3/8" ID 0.5" length |
| Pack of Bolts for Mounting Heim Joints: | | https://amzn.to/3WrSz7X | |
| 3/8 NyLock Nuts | | https://amzn.to/4ar7t3P | |
| | | | |
| | | | |
| **Other (Optional)** | | | |
| Buttkicker | | https://amzn.to/3Q2x4ah | |
| Amp for Buttkicker | | https://www.amazon.com/gp/product/B07MR6YCQ1/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1 | |