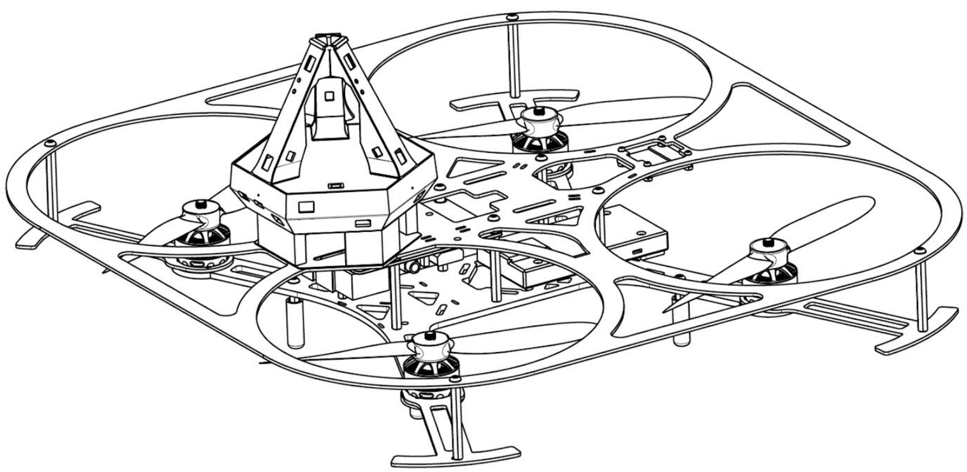
Otus Quadcopter

Manual V1.1



Date: 2019-06-06

Connection Information Guide

Router

Wifi Network name: **Robotics Network**
Password: **rcbenchmark**

To configure the router, enter 192.168.2.1 in your browser.
Administration of the router user name: **admin**
Password: **rcbenchmark**

Note: Networks are typically on 192.168.**1.**1. By changing to .2, you can connect to two networks at once on your Ground Computer. We recommend the Asus RT-N12. Other routers may cause issues difficult to troubleshoot (random lag spikes, difficulty to connect to Qgroundcontrol...)

Raspberry Pi

User name: **pi**
Password: **raspberry**

Once the router is powered, power the Raspberry Pi and wait to see the IP address on the OLED screen. The, type the ip address in your browser followed by :8080. You will load the Cloud9 editor, which allows you to access the files and the console on you drone.

Introduction

Thank you for purchasing the Otus Quadcopter! This development quadcopter is based on the open source PX4 and the Pixhawk platform. The PX4 code base is actively developed by a very strong and diverse team of developers from multiple companies and universities. It is field proven with hundreds of vehicles. This Otus Quadcopter has the advantage of being designed and tested for indoor flight, which can typically save hundreds of hours in setup time.

Many components have to come together for indoor and automated flight. Every effort has been made to simplify the setup and add automated checks. Please read the entire manual before installing your drone, it may save you significant troubleshooting time!

List of Component Required for Flight

Before installation, please check that you have all the following components:

Hardware

- Otus Quadcopter
- Otus Tracker, vibration damper, wireless receiver, USB extension, and Otus Tracker power wire.
- (2x) Vive Base station
- 3000mAh, 3S Lipo batteries
- Computer with Windows 7, 8 or 10 installed (ground computer)
- A dedicated router connected by ethernet to your ground computer
- SD card reader

Software

- RCbenchmark Tracking lab
- Qgroundcontrol
- For the simulation of scripted flight, install the PX4 toolchain and Dronecode, ideally on a Linux computer.
- To backup your software on the Raspberry Pi, use software Win32 disk imager

List of Useful References

The Otus quadcopter is leveraging powerful, well documented open source software. Here are the links to their documentation if you need additional information:

RCbenchmark documentation: https://docs.rcbenchmark.com/en/

PX4 user guide: https://docs.px4.io/en/

PX4 developer guide: https://dev.px4.io/en/

Qgroundcontrol manual: https://docs.qgroundcontrol.com/en/

Dronecode SDK manual (for automated control scripts): https://sdk.dronecode.org/en/

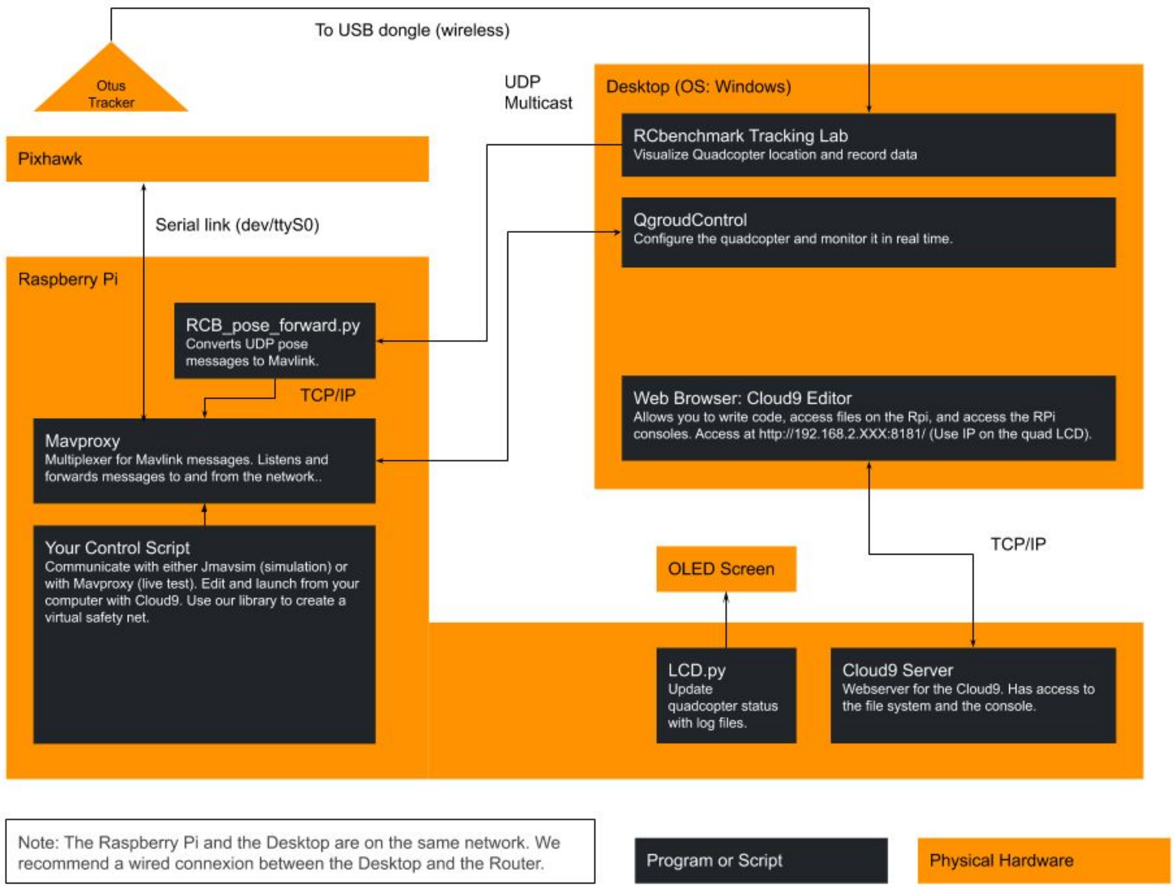
Win32 disk imager https://sourceforge.net/projects/win32diskimager/

SD memory card formatter https://www.sdcard.org/downloads/formatter/

Client code to receive data from RCbenchmark Tracking lab, and to setup the development environment of the Otus Quadcopter:
https://gitlab.com/TytoRobotics/Transport_Protocol/tree/master

System Architecture

Here is a diagram of all the software components running on your Otus Quadcopter and on your ground computer. Please take time to study this diagram and the following explanation. It will help you develop and write code for your system!



You Otus Quadcopter's brain is the flight controller, the Pixhawk. It controls the motors and does all the sensor processing for the flight. The flight controller communicates with the outside using a protocol named Mavlink. The ground control software and your control script, for example, communicate with the Pixhawk using Mavlink packets. Here is a summary of the functions performed by the **flight controller**:

- Sensor reading (gyroscope, accelerometer, barometer)
- Sensor fusion with an extended Kalman filter. The filter uses the position provided by the Otus Tracker and compensates for the 10 ms transmission delay.
- Control system
- Motor output

Your quadcopter is also equipped with a companion computer. The companion computer communicated with the Pixhawk through a serial link. The **companion computer** is responsible for the following tasks

- Forwarding the Mavlink messages from Qgroundcontrol, from the Otus tracker script, and from your control script to the Pixhawk through the serial port.
- Executing your control script written with Dronecode SDK
- Controlling the OLED screen on the Otus quadcopter and provide useful information
- Run a Cloud9 server. Cloud9 is an editor that you can access from a browser on the ground computer. This editor allows you to edit files and run scripts on the Raspberry Pi without having to connect an HDMI cable.

The **ground computer** is responsible for the following tasks:

- Run RCbenchmark tracking lab and SteamVR. The software that calculates the position of your drone.
- Run QgroundControl. The ground control station software monitors the flight and allows you to configure flight controller.
- Run Cloud9 editor in your browser. The editor that allows you to access and modify the files on the Raspberry Pi. You can use this editor to write control scripts on the Raspberry Pi.

To summarize, here is the sequence of events that allows flight in position control mode:

1. The quadcopter moves from its desired set point and the movement is detected by the Otus Tracker
2. The Otus tracker transmits the position wirelessly to RCbenchmark tracking lab on the ground computer.
3. RCbenchmark Tracking lab emits UDP packets on the local network.
4. The script RCB_Pose_forward on the Raspberry Pi reads the UDP packets and converts them to Mavlink position messages.
5. The Mavlink position messages are forwarded to the Pixhawk with the mavproxy script.
6. The Pixhawk uses its Extended Kalman filter to estimate the position of the drone with all the onboard sensors and the Otus tracker messages.
7. The Pixhawk computes the necessary corrections and sends them to the motors.

Raspberry Pi SD card software upgrade and backup.

The operating system of the Raspberry Pi and all the onboard scripts are contained on the SD card. You may want to upgrade your software with the latest changes from RCbenchmark, or you may want to backup all your code and configuration. It is possible to create an image of the SD card.

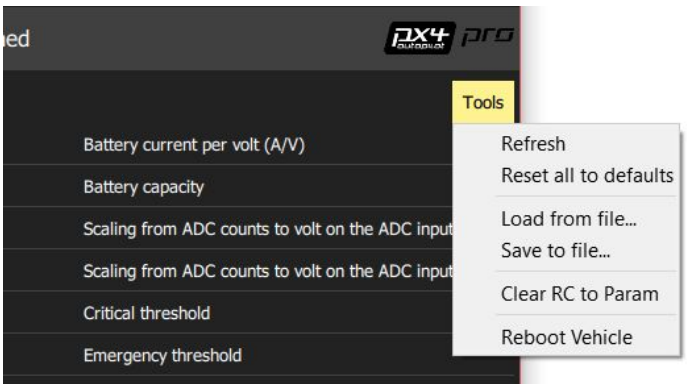
The latest image file from RCbenchmark is available at this link:

https://docs.rcbenchmark.com/en/download.html

To backup or write an image to your SD card, you will need the software Win32 disk imager. Don't forget to backup regularly your work! To write an image to a new SD card, you may need to format it beforehand. We recommend to use the SD Memory Card formatter software from the SD card association.

Parameter File Update and Backup

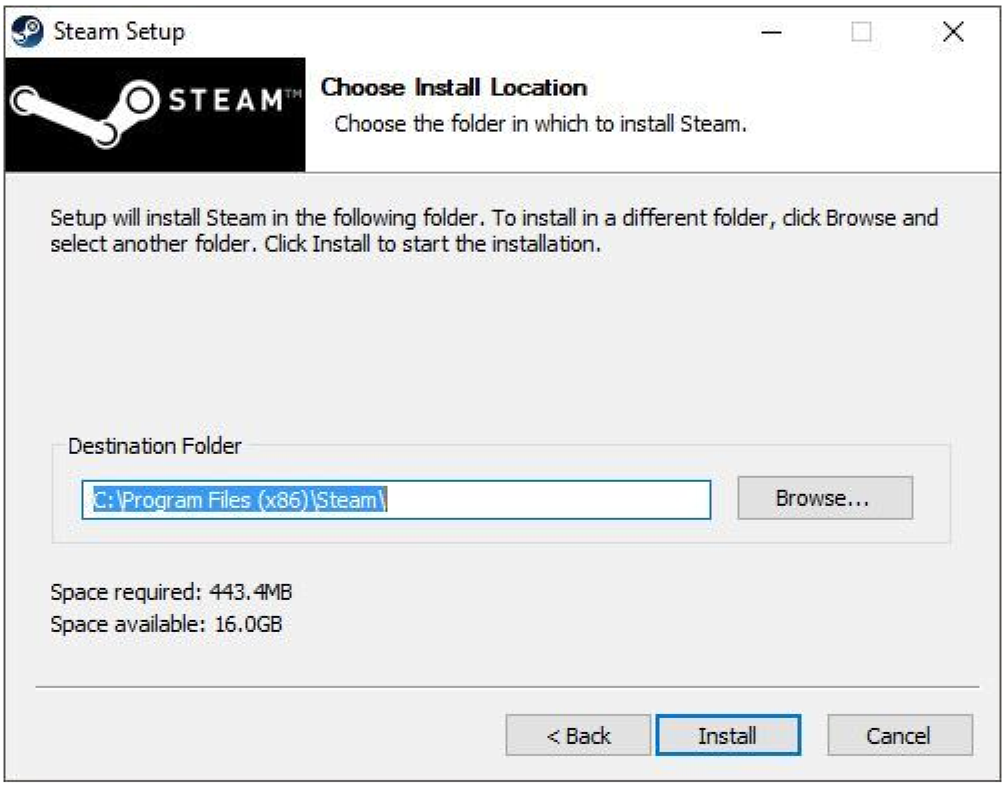
The software running on the PX4 is entirely defined by parameters. For example, the number of motors on the vehicle, the PID constants of the controller, and the calibration coefficient of the gyroscope are all parameters. You can access those parameters from Qgroundcontrol once the ground station is connected to the Pixhawk. The Otus quadcopter comes with the all the parameters configured for indoor flight, and the quadcopter is pre-calibrated. You may want to change the controller parameters, or any other parameter for your development. You can backup and restore the parameter from Qgroundcontrol. Please be aware that if you use the parameter file available from our website, or if you transfer the parameter file between quadcopter, **you will have to redo the calibration** of your quadcopter. To perform the calibration, check the PX4 User guide.



Software Setup

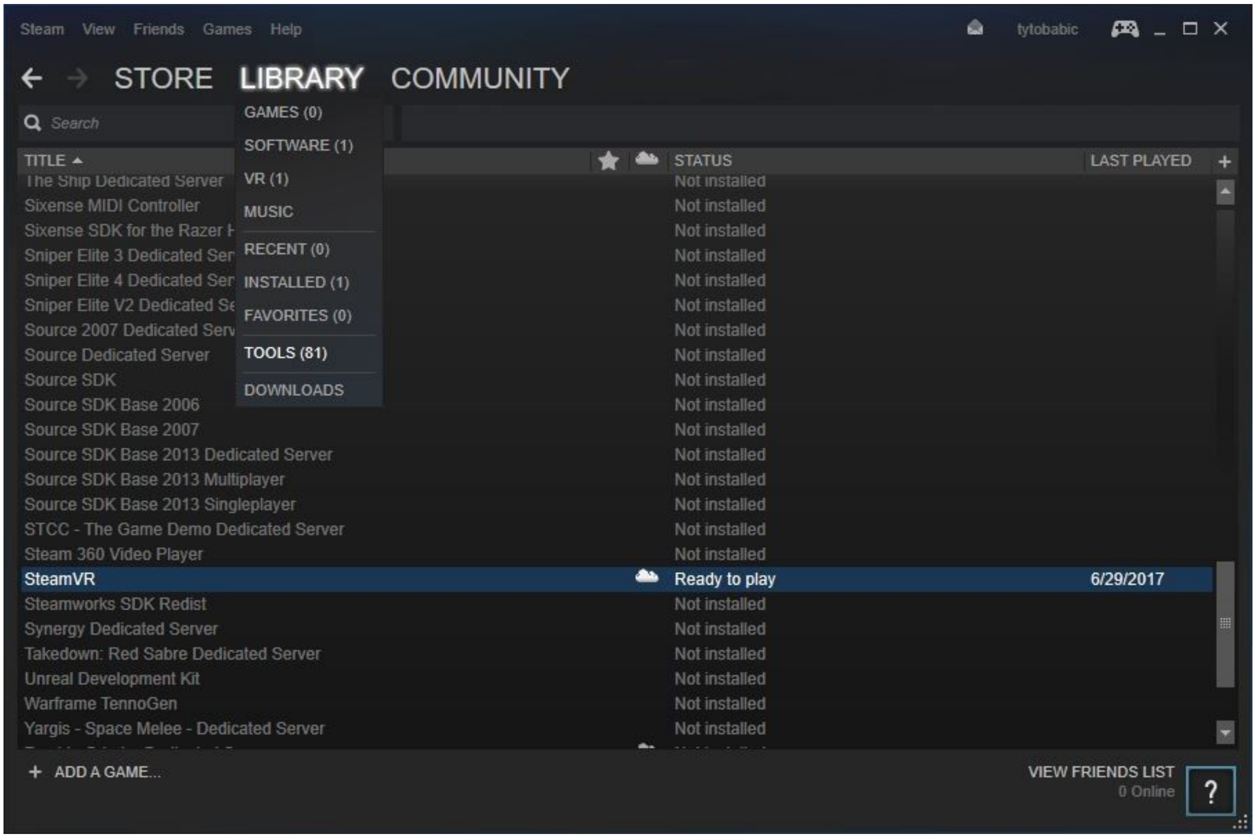
Install Steam

The RCbenchmark Tracking Lab and the Otus tracker are compatible with and require SteamVR. Install Steam by downloading the executable (.exe) from the [official website](http://store.steampowered.com/about/) (http://store.steampowered.com/about/). You will have to create an account. Please note that it is preferable to install Steam on the default destination folder as shown on the Steam Setup image.



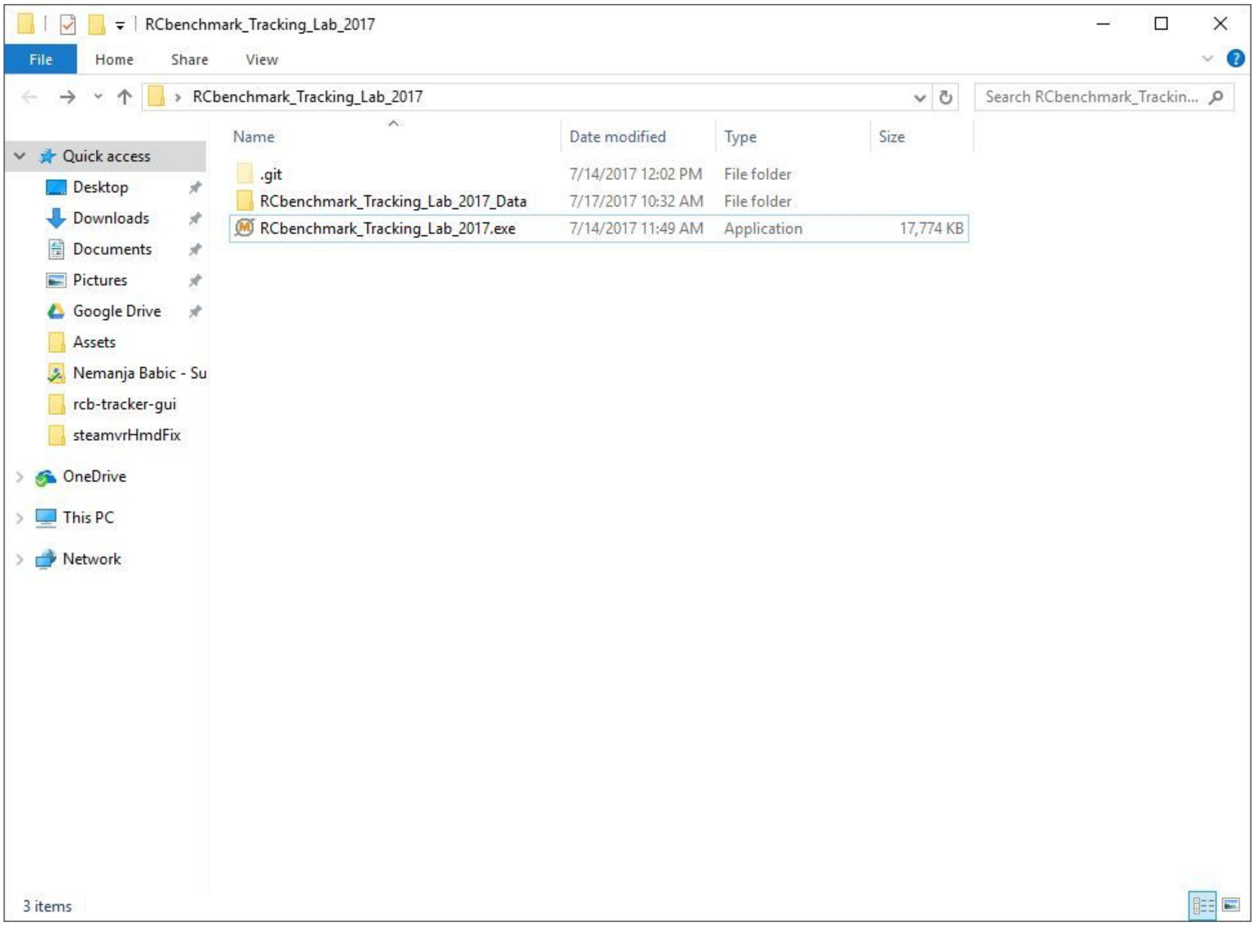
Install SteamVR

SteamVR can be installed inside the Steam software. Open Steam, go to ‘Library’ and click on Tools. Search for SteamVR and double click on it to start the installation.



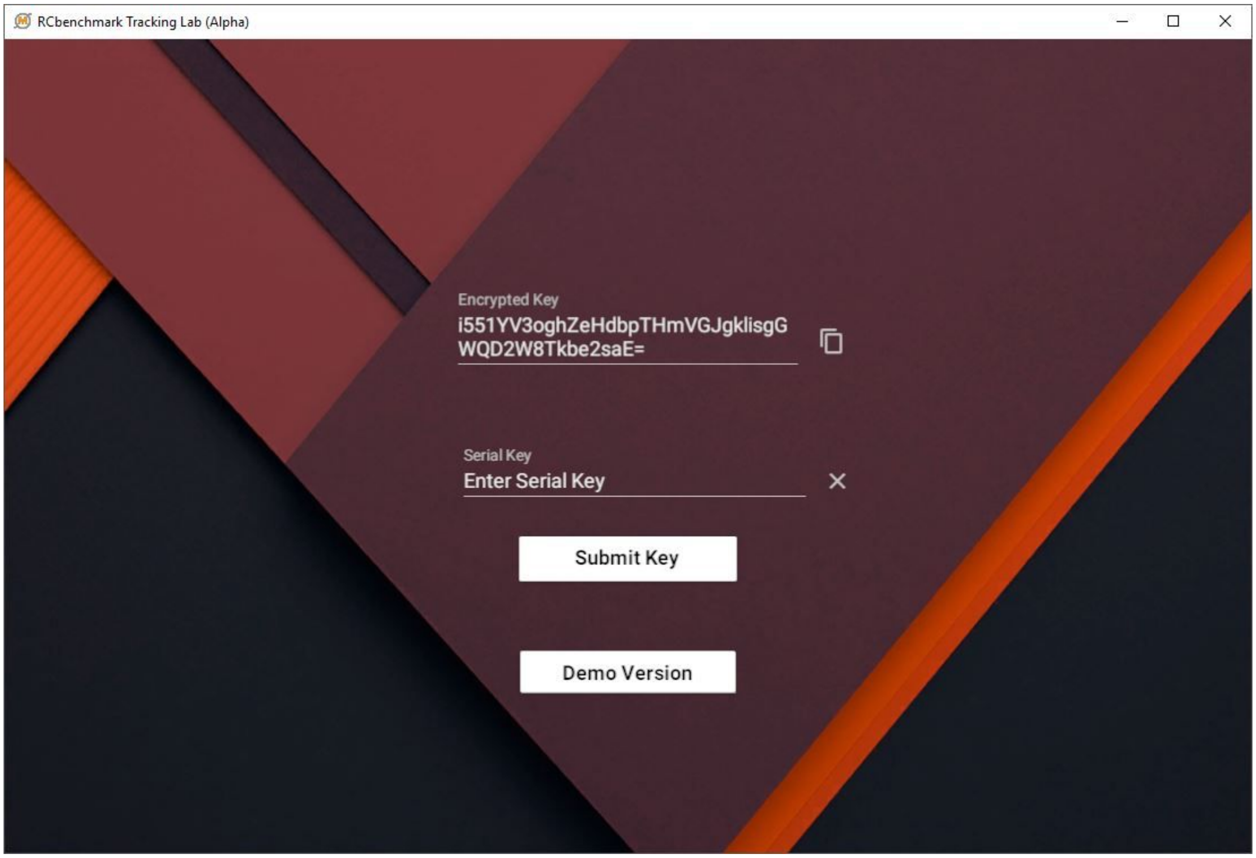
Download RCbenchmark Tracking Lab

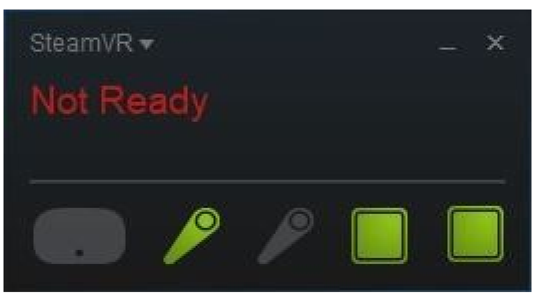
Please download the RCbenchmark Tracking lab from [this page](https://docs.rcbenchmark.com/en/motion-capture/rcbenchmark-tracking-lab.html) (https://docs.rcbenchmark.com/en/motion-capture/rcbenchmark-tracking-lab.html).



Run RCbenchmark Tracking Lab

Double click on RCbenchmark_Tracking_Lab_2017.exe. The first time you run the software, some background configuration will be made. The software will prompt you to manually restart the software and SteamVR. You will have to do this only once after installation. While you wait a maximum of 24 hours to receive the license key by email, you can stream data and connect controllers. The only limitation is that recording and data streaming is limited to 1 Hz.
Note: If after restarting the program, you get the error “Headset not connected properly” on the SteamVR tools panel, you will have to do the configuration manually as explained in the expandable section below.
Note: “Not Ready” text on the SteamVR tools panel is normal and does not impact the Otus Tracker.





[Otus Tracker Manual Configuration (Optional)]

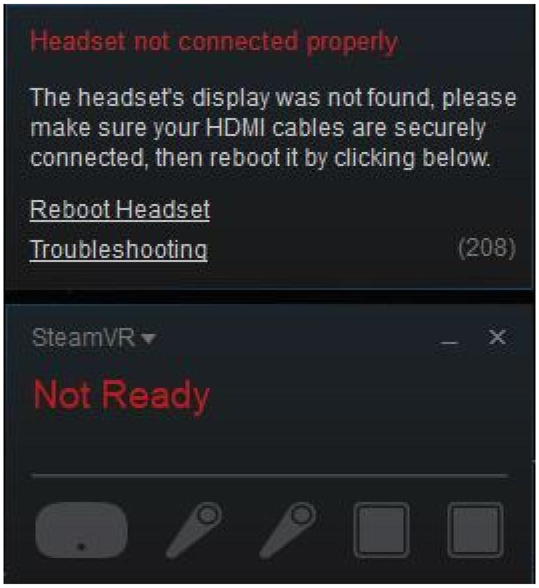
This section should be followed if the automatic configuration of the RCbenchmark Tracking Lab did not work.

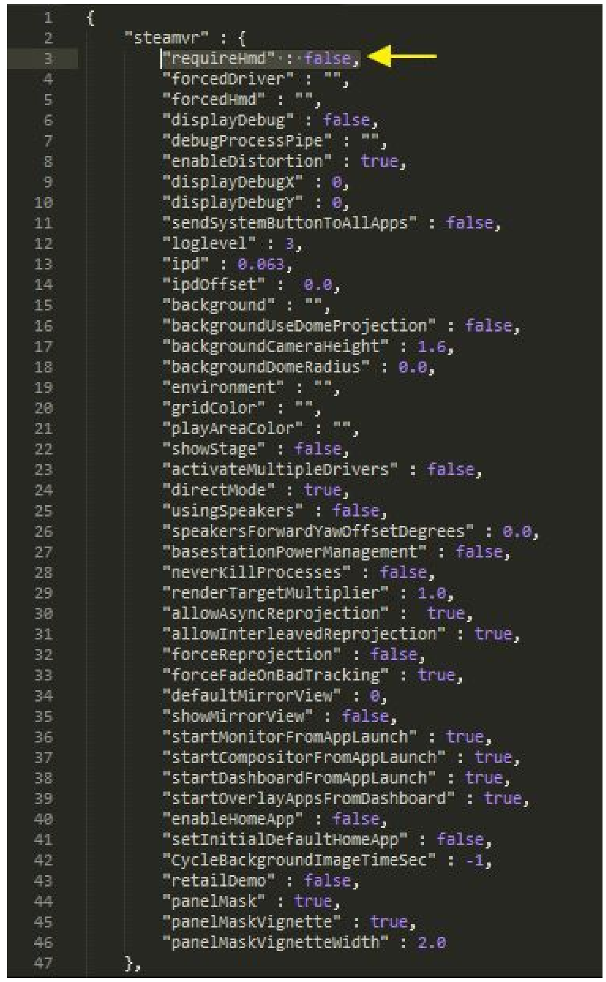
If you are getting the error shown on the right "Headset not connected properly" after running the RCbenchmark Tracking Lab Software, it means that the Otus tracker has not been configured to function with SteamVR properly. The RCbenchmark Tracking Lab Software is supposed to fix this problem the first time you run it. If the automatic modification failed, follow these steps:

1. Locate the following configuration file and open it with a text editor: C:\Program Files (x86)\Steam\steamapps\common\SteamVR\resources\settings\default.vrsettings
2. If you have not installed Steam on the default path " C:\Program Files (x86)\Steam", then Step 1 changes accordingly.

3. Search for the “requireHmd” key under “steamvr”, set the value of this key to “false” and save the file.
4. If SteamVR is running, close and restart it.
5. When SteamVR restarts, you will see that it is now possible to connect the Otus tracker without the HMD.

Note: “Not Ready” text is normal and does not impact the Otus Tracker.





```
{
    "steamvr" : {
        "requireHmd" : false,
        "forcedDriver" : "",
        "forcedHmd" : "",
        "displayDebug" : false,
        "debugProcessPipe" : "",
        "enableDistortion" : true,
        "displayDebugX" : 0,
        "displayDebugY" : 0,
        "sendSystemButtonToAllApps" : false,
        "loglevel" : 3,
        "ipd" : 0.063,
        "ipdOffset" :  0.0,
        "background" : "",
        "backgroundUseDomeProjection" : false,
        "backgroundCameraHeight" : 1.6,
        "backgroundDomeRadius" : 0.0,
        "environment" : "",
        "gridColor" : "",
        "playAreaColor" : "",
        "showStage" : false,
        "activateMultipleDrivers" : false,
        "directMode" : true,
        "usingSpeakers" : false,
        "speakersForwardYawOffsetDegrees" : 0.0,
        "basestationPowerManagement" : false,
        "neverKillProcesses" : false,
        "renderTargetMultiplier" : 1.0,
        "allowAsyncReprojection" :  true,
        "allowInterleavedReprojection" : true,
        "forceReprojection" : false,
        "forceFadeOnBadTracking" : true,
        "defaultMirrorView" : 0,
        "showMirrorView" : false,
        "startMonitorFromAppLaunch" : true,
        "startCompositorFromAppLaunch" : true,
        "startDashboardFromAppLaunch" : true,
        "startOverlayAppsFromDashboard" : true,
        "enableHomeApp" : false,
        "setInitialDefaultHomeApp" : false,
        "CycleBackgroundImageTimeSec" : -1,
        "retailDemo" : false,
        "panelMask" : true,
        "panelMaskVignette" : true,
        "panelMaskVignetteWidth" : 2.0
    },
```

Install the Base Stations

- The base stations should be installed at a maximum of 5 meters from each other. More than 5 meters may work, but it is outside the recommended specification.
- One base stations has to be in **b mode** and the other base station has to be in **c mode**. You can change the mode of operation by pressing the button at the back of the base station.
- We recommend installing the base stations at opposite corners of the room.

- To reduce occlusion, we recommend installing the base stations at a height of around 2 meters. They should be pointing slightly downward: they have a cone of vision of 120 degrees vertically and horizontally.
- More than two base stations is not supported.
- The base stations are synced optically. They have to see each other to work.

Install the OTUS Tracker

Unbox your Otus tracker kit. Make sure you have all of the following:

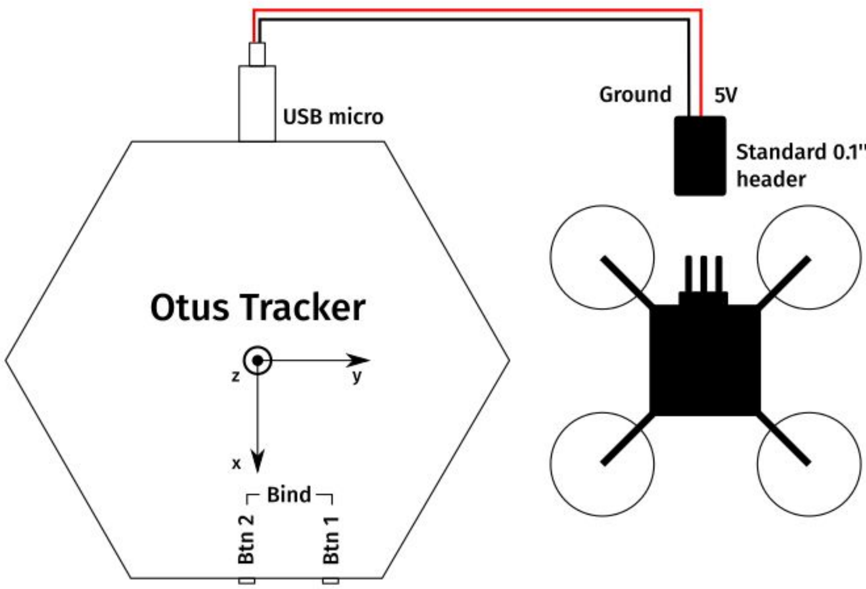
- Otus Tracker
- USB Wireless Receiver
- USB to 3-pin Connector
- USB Extension
- USB A to USB mini
- Vibration Isolator

To test tracking, connect the Otus tracker to a USB port on your local machine via the USB A to USB mini cable. Launch the RCbenchmark Tracking Lab software. The SteamVR control panel should launch automatically. The tracker has to see at least one base station to start tracking. Two base stations in sight are required for best tracking performances.
You can download the CAD model of the Otus tracker here (https://www.rcbenchmark.com/wp-content/uploads/2017/07/Otus_tracker_cleaned_published.zip) (FBX, OBJ and STL). It can be used for visualisation in your software or for 3D modeling. The model has been simplified to reduce the polygon count and increase performance when used in real time visualisation.

Power and Wireless Connections

The Otus tracker is powered from a 5V power source from your UAV or robot when used wirelessly. This is to reduce weight.

Follow the connection diagram on the following image. You Otus Quadcopter has a female connector specifically for the standard Otus Tracker power cable.

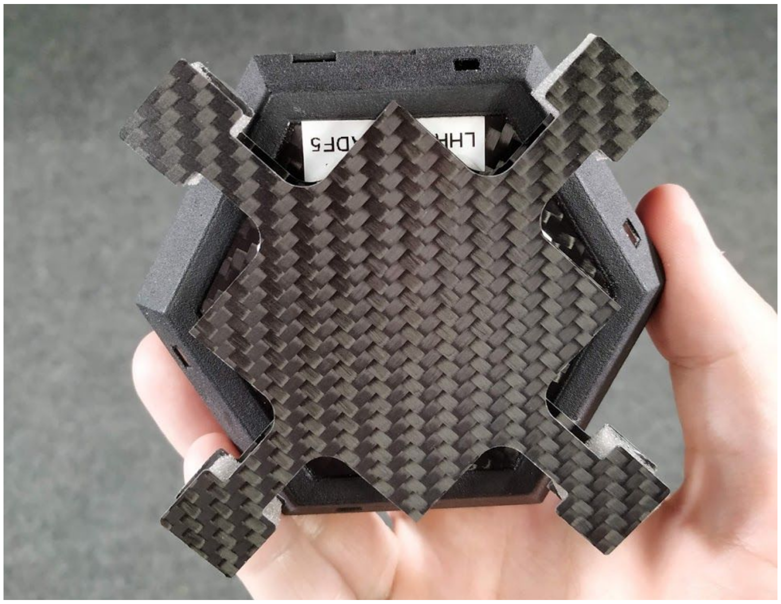


Connect the wireless receiver to the USB extension cable and the USB extension cable to your computer. The USB extension cable is not necessary, but it will reduce the risk of losing tracking due to electromagnetic interference.

Your Otus trackers come paired to their wireless receiver by default. If you want to pair the unit again, right click the controller icon in SteamVR and click “Pair Controller”. Then, press and hold the two buttons on the Otus tracker until the LED flashes blue. Once the LED is solid blue, the trackers are paired.

Vibration Damper and Quadcopter Installation

Please use the included double sided foam tape to install the Otus Tracker, the vibration damper, and the Quadcopter as indicated in the pictures below:

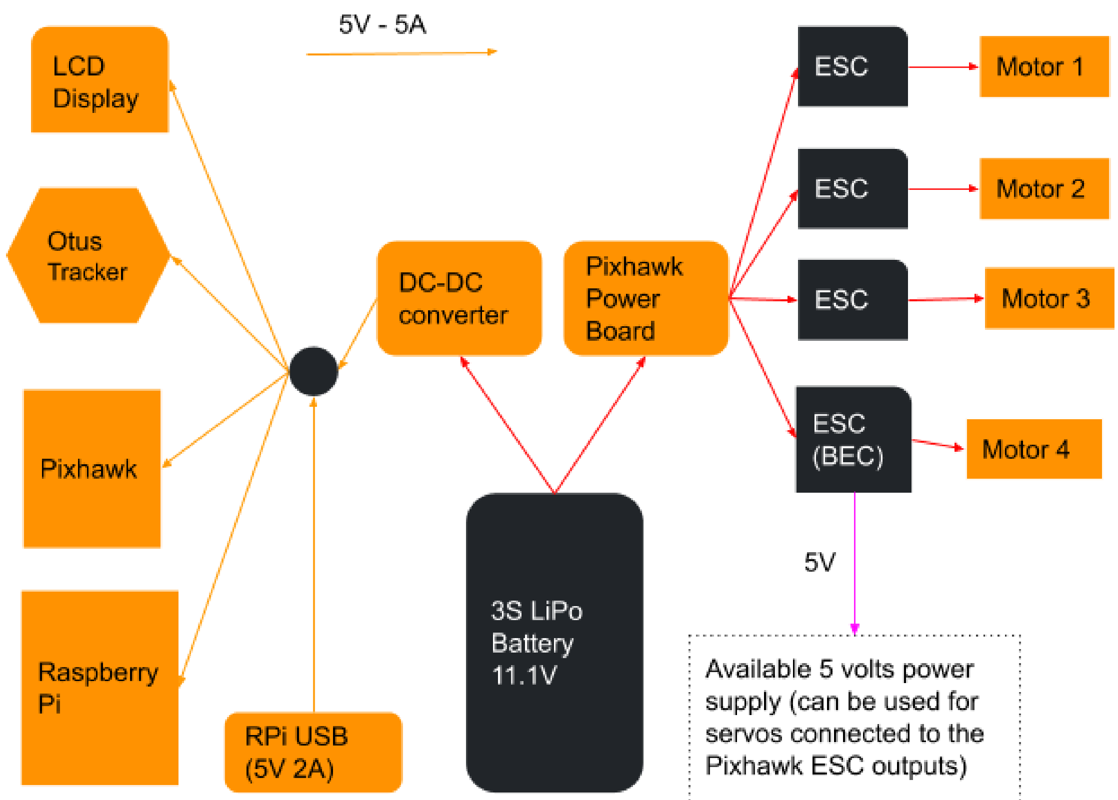


Note: The reference axis of the Otus Tracker is the North-West-Up (North - x axis ; West - y axis ; Up - z axis) and for the Pixhawk it is North-East-Down (North - x axis, East - y axis, Down - z axis).
The position forward script (pose_forward_rcb.py) on the Raspberry Pi transforms the two frame of references and provides vision estimates using the method, vision_position_estimate_encode().

Powering the Otus Quadcopter

The Otus Quadcopter can be powered with a power supply set at 15V or 16V, and by a 3S or 4S battery. The power supply can be useful during development. You can also power the Raspberry Pi through the USB micro port with a short USB cable and a power supply capable of providing 2.1A. Typical tablet charger can provide 2.1A. The cable has to be short, as the voltage may drop too much with a long USB cable, and the Raspberry Pi will run at reduced speed.

The image below shows the power diagram. You can use a 3S or 4S battery.



Battery safety instructions

We recommend to use a 3S, 3000 mAh battery. Never discharge your lithium polymer battery under 3.2V/cell (9.6V).

Warning Signs

1. Battery appears to be swollen or the battery case appears to be damaged.
2. Battery is heating up alot.
3. Battery was left discharged completely for more than 3 days.
4. Battery was overcharged.

If you notice any of these signs, please do not use the battery as it may not be safe

Charging

1. Always use the balance connector in order to charge each of the cells at the same rate.
2. Do not leave a charging battery unattended.
3. Do not charge the battery immediately after a discharge or when the battery is warm. Allow the battery to cool down before charging.
4. It is recommended that you charge the battery at 1C charge rate(1.3 Amps) and not go beyond 3.9 Amps(which is the maximum charge rate). Charging at the maximum rate will reduce the battery life.
5. The maximum charge the battery pack can hold is 16.8 Volts (at 4.2 volts per cell)

General Safety Instructions

1. Always store the battery in a fire-safe container or pouch.
2. Never overcharge or over-discharge the battery.
3. Never use a battery which appears damaged or swollen.
4. After usage allow the battery to cool down and store it in a storage mode, which is 15.2 Volts (3.8 Volts per cell)
5. Unplug the battery from the drone after use. Never leave the battery plugged into the drone after use.
6. Do not expose the battery to direct sunlight for long durations.

Connect the Ground Computer to the Network

Once powered, the Raspberry Pi will be automatically connected to the private network hosted by the router. A private network is preferred to reduce the risk of dropping datagrams in flight. Please use the recommended router, or contact us to test your router. Other routers have shown to cause random lag issues causing issues that are difficult to troubleshoot.

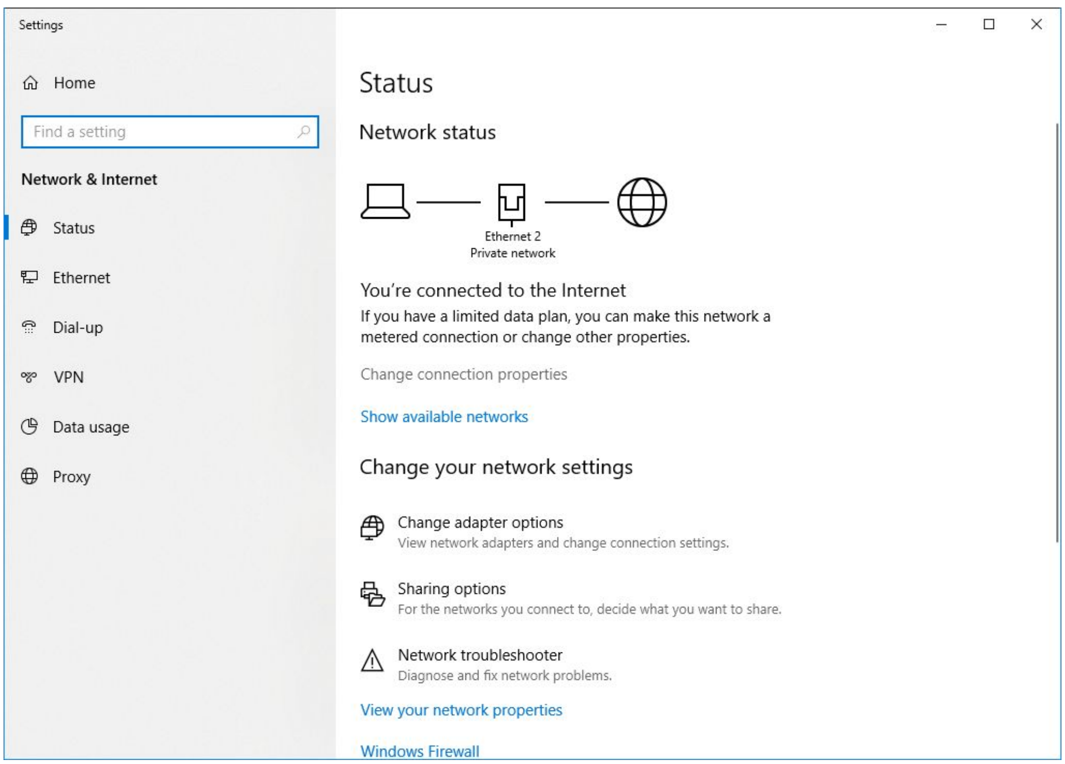
The router has been pre-configured to create a private sub-network between the main computer and the quadcopter which prevent conflicts with the default network (internet). This is the reason why the network address of the router is 198.162.2.xxx, where 2 in the host address indicates that the IP belongs to a subnet created by the router.

The ground computer (the server) which hosts RCbenchmark Tracking Lab should also be connected to the private network. Although, we can use wifi to connect the ground computer to the network, we recommend using a wired connection to minimize network latency.

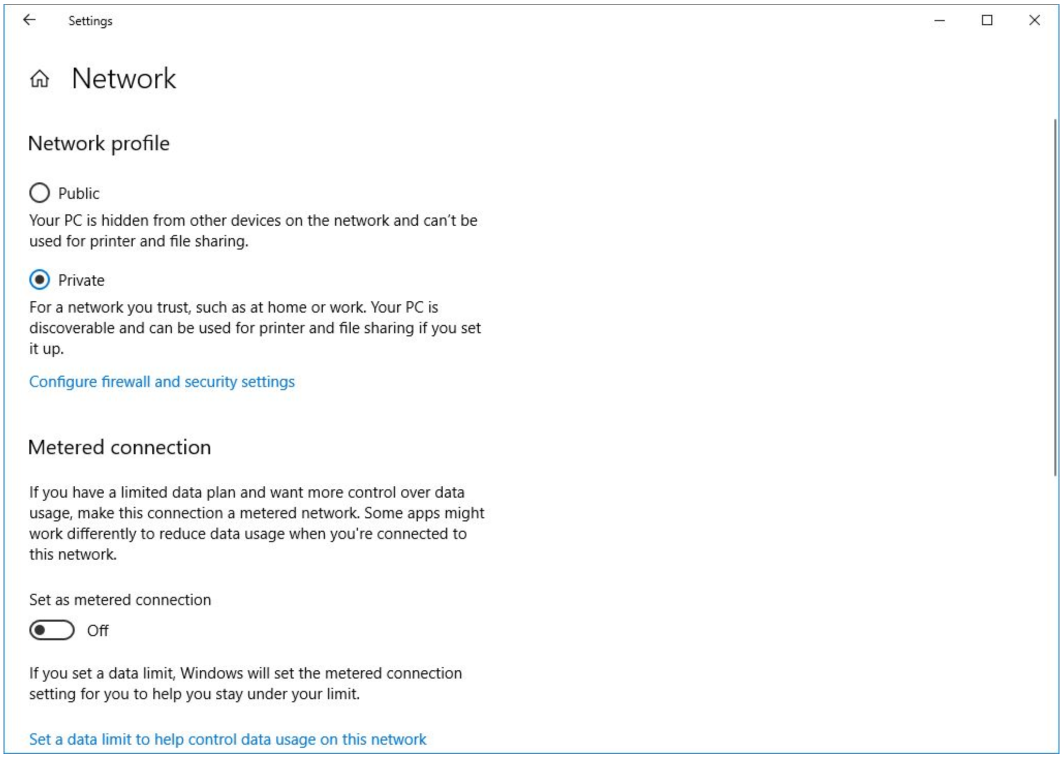
Setting up a Private Network

In order for QgroundControl to receive data from the Quadcopter, the server must be recognized in a private network.

In Windows, click on the network icon in the right corner of the taskbar. Then, click on Network and Internet Settings. Then, click on change connection properties.



Click on Private Network to switch to a private network.



Connect the Ground Computer to the Raspberry Pi

Powering Up the Drone and OLED screen

After powering your drone, check the OLED screen. Otus Quadcopter is equipped with an OLED screen to display real-time status of the flight control programs running on the Raspberry Pi. Once the Raspberry Pi is connected to the internet, the screen displays the IP address of the Pi.

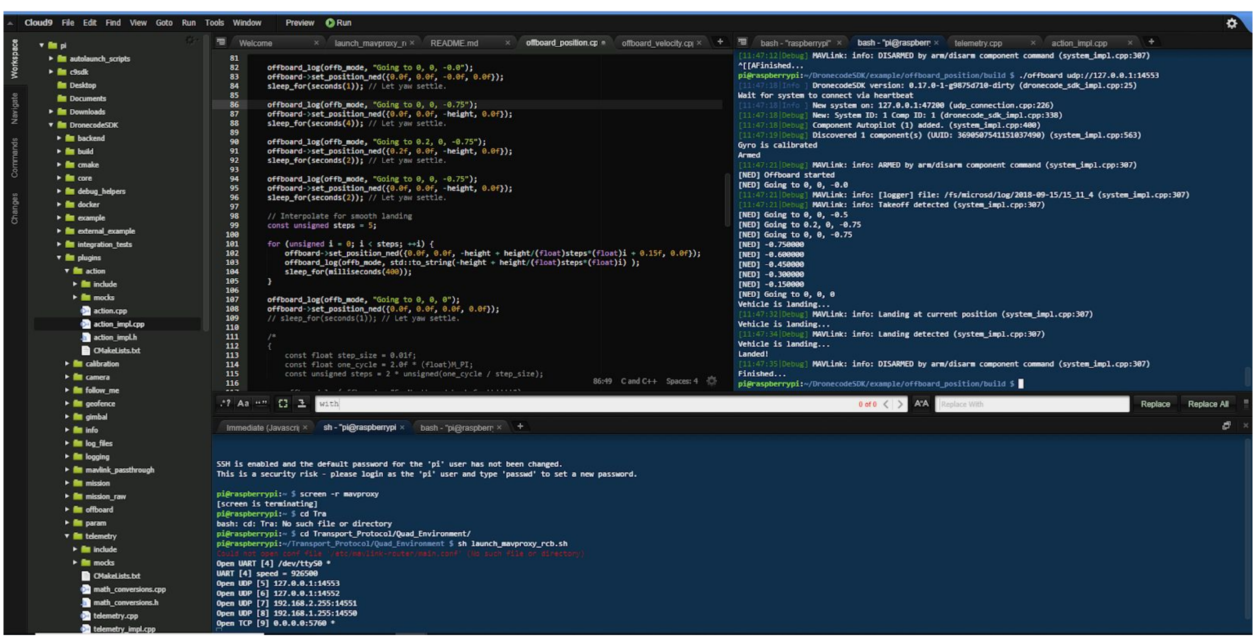
The OLED screen helps with quick discovery and debugging of the issues. Displayed below is a guide to understand what each error message means and how you can get resolved.

Error Message	Reason	Fix
Internet Offline	The Raspberry Pi is not connected to the internet	Connect the router present in the kit to the computer with an ethernet cable. Turn on the router and the pi will automatically connect to the router.

MavProxy Failed	This indicates that the MavProxy script did not launch.	Please reboot the Raspberry pi. Incase the problem still persists please download the ISO image file from here
Pose Forward Failed	This indicates that the position forwarding script has not been launched.	Please reboot the raspberry pi. This should fix the issue.
Tracking Lab Offline	This indicates that the pi is not receiving any messages from the RCBenchmark Tracking Lab	Please check the IP address and port number, then switch the streaming OFF and then toggle it back ON. Incase problem still persists, force close and restart the Tracking Lab.
EKF Yaw Error	This indicates that Local axis was not reset on the RCbenchmark Tracking Lab causing the position forward to process wrong estimates to compute the position information.	In order to fix this issue, you need to reset the local axis on the Rcbenchmark Tracking Lab and reboot the Pixhawk through QGroundControl.

Connect to the Raspberry Pi with the Cloud9 Editor

Open Firefox of Chrome. In the address bar, type the IP address of the drone followed by :8080. For example, type 192.168.2.39:8080. The IP address of the drone appears on the OLED screen.



Manually Connect to a New Network

If your drone was purchased with a router, the Raspberry Pi will connect automatically to the router. If you have your own router, you can connect the Raspberry Pi to the network using the graphical user interface of the Raspberry Pi by following those instructions.

1. Remove the propellers if you are going to power the drone with the XT60 connector.
2. Connect an HDMI cable and a display to the Raspberry Pi.
3. Connect a USB mouse and keyboard to the Raspberry Pi.
4. Power the drone and wait for the desktop to appear on your screen.
5. Click on the network icon on the top right of the Raspberry Pi desktop. Click on your wifi network SSID and enter your password.

Change the Default Wireless Network

If you have to alternate between two networks (for example, one with internet access and one without), you do not need to connect an HDMI cable every time. Connect to Cloud9 from your browser (192.168.X.XXX:8080, as shown on the quadcopter OLED) on a computer on the same network as the Raspberry Pi. Then, in the console, type:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

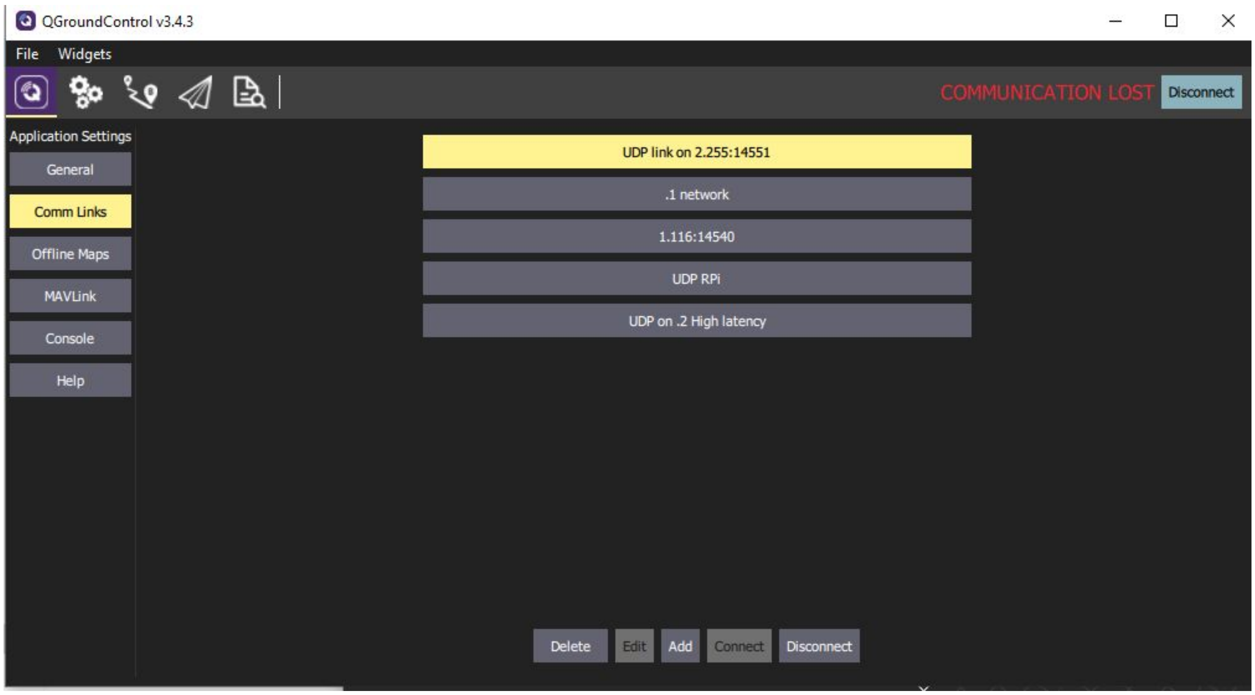
You will see this file

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev

update_config=1

country=US

network={

ssid="YOURSSID"

psk="YOURPASSWORD"

scan\_ssid=1
}
```

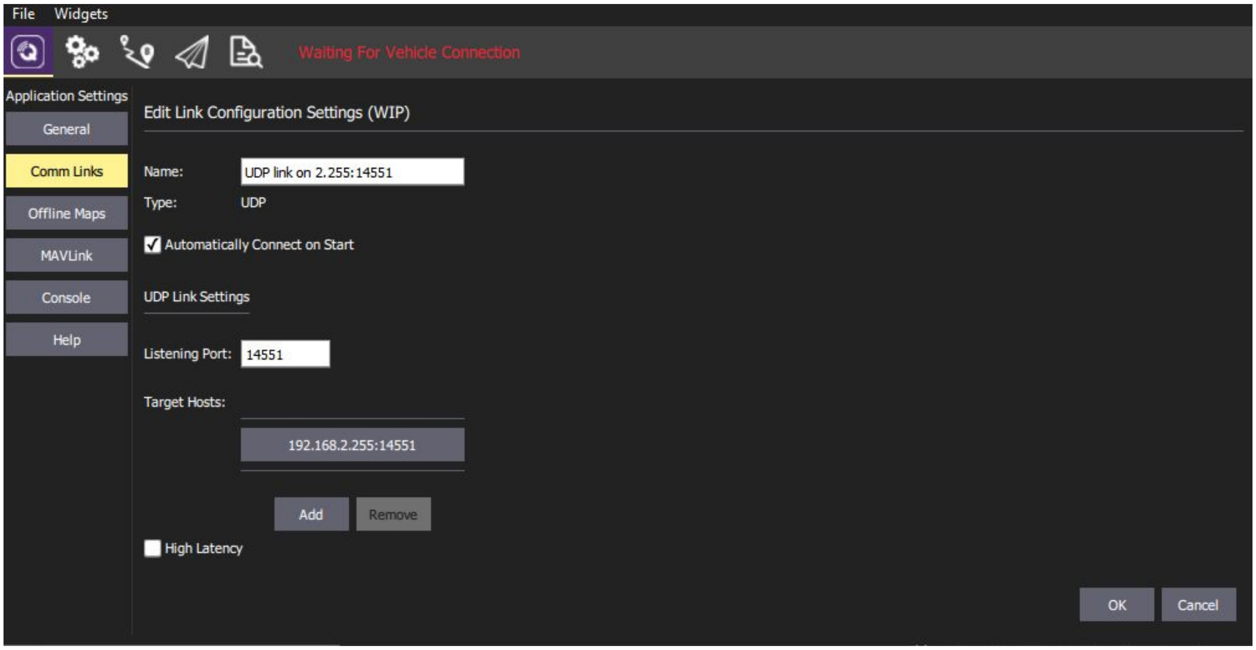
Edit YOURSSID and YOURPASSWORD to the new network. Do not add spaces. Press ctrl+O to save and ctrl+W to exit. Reboot the Raspberry Pi. The Raspberry Pi should now be connected to a new network. If you made a mistake in the network SSID or password, you will need to use an HDMI cable, a mouse and a keyboard to reconnect to the wifi.

Connect QGroundControl to the Raspberry Pi

QGoundControl should connect automatically to the quadcopter if the ground computer and the Raspberry Pi are on the same network. If the connection does not work, try to ping the Raspberry Pi from the ground computer to see if you obtain a response. Open the command window in Windows and type ‘ping quadcopterIPAdress’. For example, the command could be ‘ping 192.168.2.40’. If you receive a response, both computers are on the same network.



You may need to automatically add a connection link in Qgroundcontrol. Press the Q button on the top left and go the Comms link tab. Add a new comms link configured as in the image below. After, press the connect button.



Your First Flight

Checklist for your first flight

Here is a summary of the steps you have done up to now:

Step 1: Install the required tools

Install QGroundControl.
Install Steam and Steam VR.
Install RCbenchmark Tracking Lab and activate the license.
In case of any issues, follow the complete tools installation guide.

Step 2: Setup the network

Plug the ethernet cable from the router to the main computer and connect the router to the power supply.

Check the connection on the computer and set the connection to private network.

Step 3: Install the base stations

One of the base stations should be in mode b and the other in mode c.

Ensure that the LED lights on the base stations are green.
In case of any issues, check the base station section or the troubleshooting section.

Step 4: Install the Otus Tracker on the quadcopter.

Fix the vibration dampening base on the Quadcopter as explained, then fix the Otus Tracker on the base with double sided tape. Connect the Micro USB to the Otus Tracker.

Step 5: Power the Raspberry Pi with the battery.

Ensure that the XT60 connector is firmly connected to the battery.

Development Mode: In case you are developing a program to be run on the Raspberry Pi and do not need to power the motors, use the micro-usb slot on the Raspberry Pi to power only the Raspberry Pi and Pixhawk.

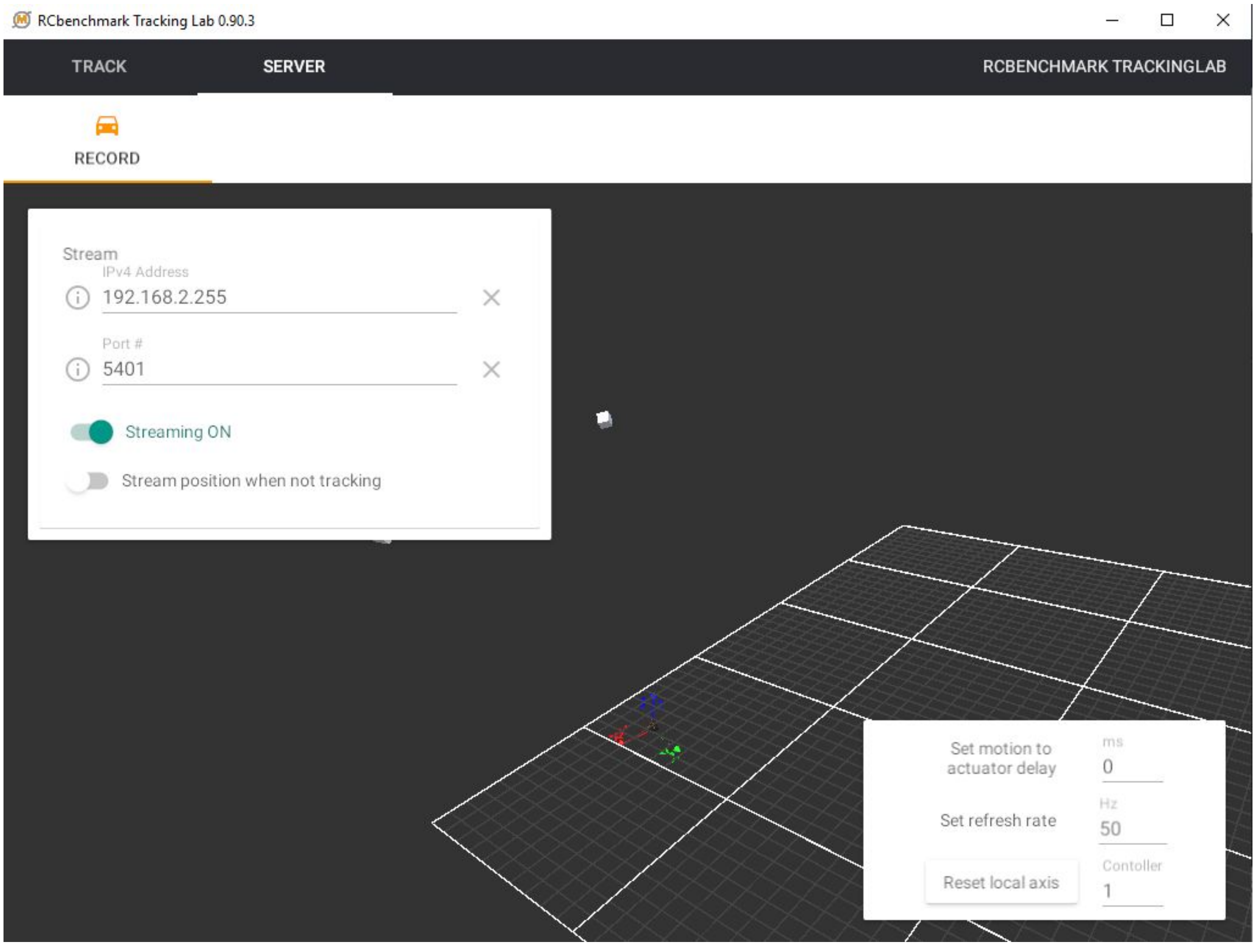
If your are using your own router, you will need to manually connect the Raspberry Pi to the network using a mouse, a keyboard and a screen connected to the Raspberry Pi.

Step 6: Check the LCD screen display for the following messages after powering the Raspberry Pi

RCbenchmark logo.
IP address of Quadcopter.
Tracking Lab Offline.

Step 7: Launch RCbenchmark Tracking Lab and Stream the position data.

Confirm that the Otus is tracked by the Tracking Lab software.
Go to the server tab and enter the IP address of the Quadcopter.
Enter the port number 5401.
Toggle streaming ON.
Set the refresh rate to 50 hz
The OLED screen on the quadcopter should display: Receiving Data!
In case of any issues with transmitting data, follow the RCbenchmark Tracking Lab instructions.



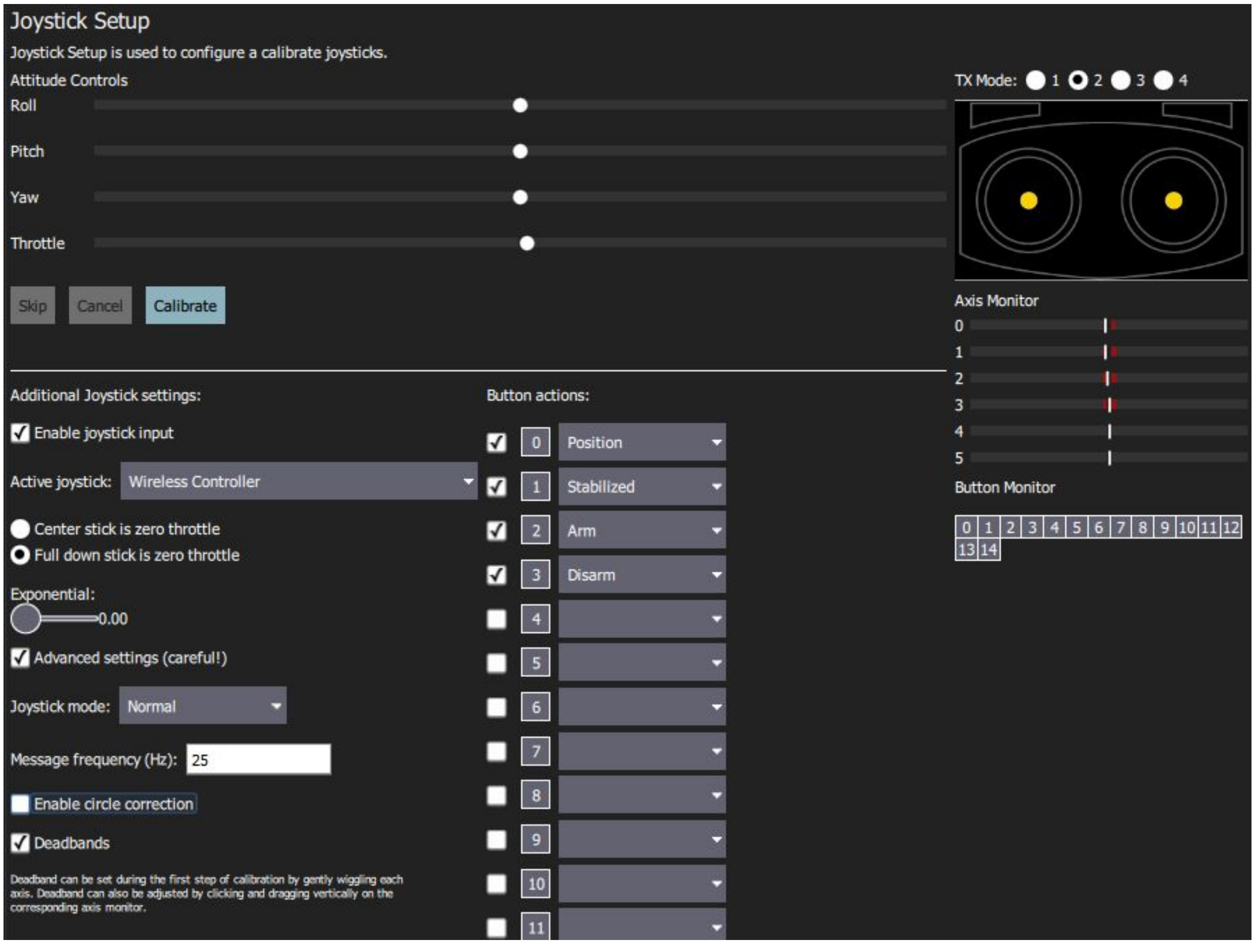
Step 8: Launch QGroundControl

Check that the QGroundControl is connected to the Quadcopter. If not, follow the connection instructions. Connect the PS4 controller and check that it works in the Joystick tab of Qgroundcontrol.

The quadcopter is now ready for flight.

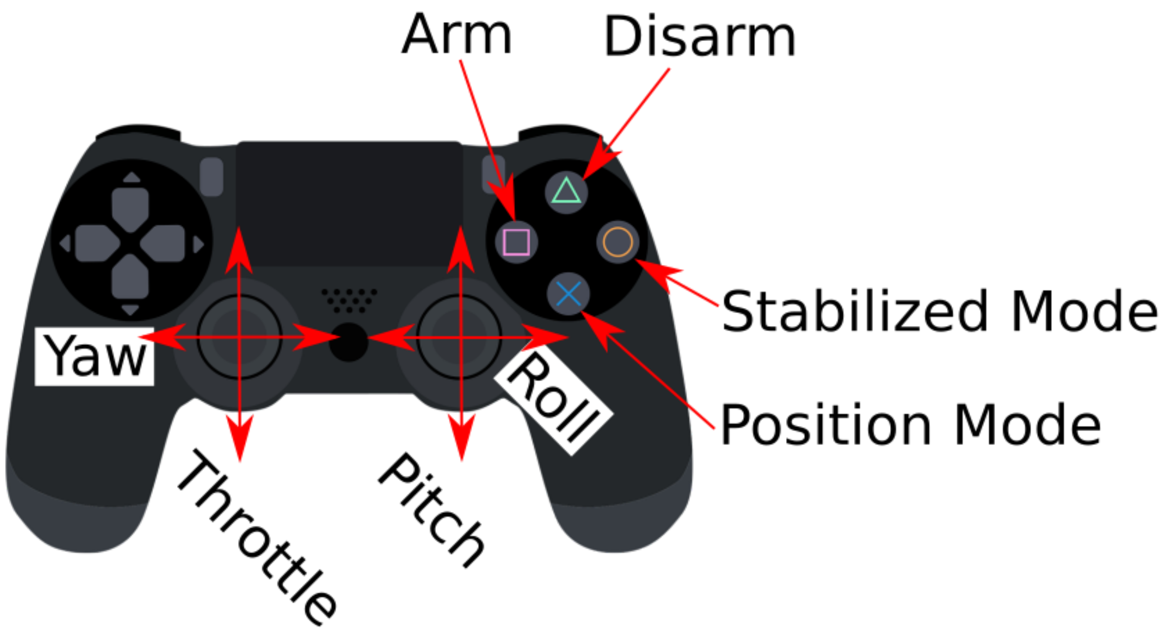
Manual Control Test Flight

You drone can fly in multiple flight modes. You will likely use three flight modes: Stabilized, Position Hold, and Offboard. In stabilized flight mode, you control the attitude of the drone with the joystick. When you release the pitch and roll stick, the quadcopter will return to level flight. Your quadcopter may still drift, as it does not control its positions. You first need to setup the joystick configuration in Qgroundcontrol. Please change the button configuration as in the image below.



The joystick has to be connected to the computer with Qgroundcontrol through USB. When done correctly, a new joystick tab will appear in Qgroundcontrol. Please activate "Advanced settings" and Deadbands. The deadband is configured during calibration. It will reduce wobble due to the joystick at hover. You have two options for throttle zero. We recommend to activate "Full down stick is zero throttle". With this option, it will not be possible to accidentally arm the quadcopter by pressing the arm button by accident. The throttle stick has to be pushed down for the quadcopter to arm.

To fly, arm the drone with the Throttle stick down and press the Arm button. The image below shows the shows the binding on the joystick.



To disarm the quadcopter, hold the throttle stick to the bottom left. Once airborne, the quadcopter should only need very light inputs from the right stick to correct for drift. Many beginner pilots tend to overcompensate. **If you feel you are losing control, the best course of action is to cut the throttle and disarm the drone**. Do not try to keep the drone in the air unless you feel you know how the drone is orientated and how much to compensate. Keep the drone in the same orientation relative to you to prevent disorientation.

We recommend that you learn to fly the drone manually with very gentle manoeuvers for 2 to 3 battery packs. This will give you a feeling of how responsive the drone is. It will also be easier for you to tell if the drone is behaving properly when you will work with automated flight mode.

Hover Throttle Calibration

When you fly in *Position Hold* or *Offboard mode*, the drone has to know approximately what throttle is necessary to hold a constant altitude. This hover throttle changes as a function of drone weight, battery voltage, motor, and propellers. If you do not set the hover throttle correctly, the quadcopter will have difficulty keeping the right altitude in position hold or offboard mode. The quadcopter will also change altitude rapidly if you change from position hold to stabilized mode in flight. You hover throttle should already be calibrated. If you need to change the hover throttle, go to the tuning tab in Qground control. Adjust the percentage so that your quadcopter holds altitude in stabilized mode when the throttle stick is at 50% (neutral position). Typically, the value of the throttle should be 52% with a battery, and 32% when flying with a tether and not battery.

Position Hold Flight

Once you quadcopter works well in stabilized mode, and you can confirm tracking during the flight, you are ready to fly in position hold. Land the quadcopter and switch to Position mode with the PS4 controller. Slowly increase the throttle until the quadcopter takes off. Position hold is actually a slightly confusing term. When the roll and pitch sticks are centred, the quadcopter is in position hold. When roll or pitch sticks are offcentered, the quadcopter is in velocity mode.

Automated (Offboard) Flight

Automated mode uses the power Dronecode API. Dronecode can be used on the ground computer, on a companion computer, or even on an Android or an Iphone phone. The library is actively developed in C++ with Python and Swift binding.

Once you are able to fly in position hold, you are ready to try an automated Flight! Your Raspberry Pi comes pre-installed and pre-compiled with Dronecode. To fly in automated mode, first make a test flight in position hold. After, type the following command in Cloud9:

```
cd
cd DronecodeSDK/example/offboard_position
```

The first time you compile the code, type:

```
mkdir build
cd build
```

to create a build directory. You only need to type mkdir once, and only if the folder build does not exits.

Then, build the code with

```
cmake ..
make
```

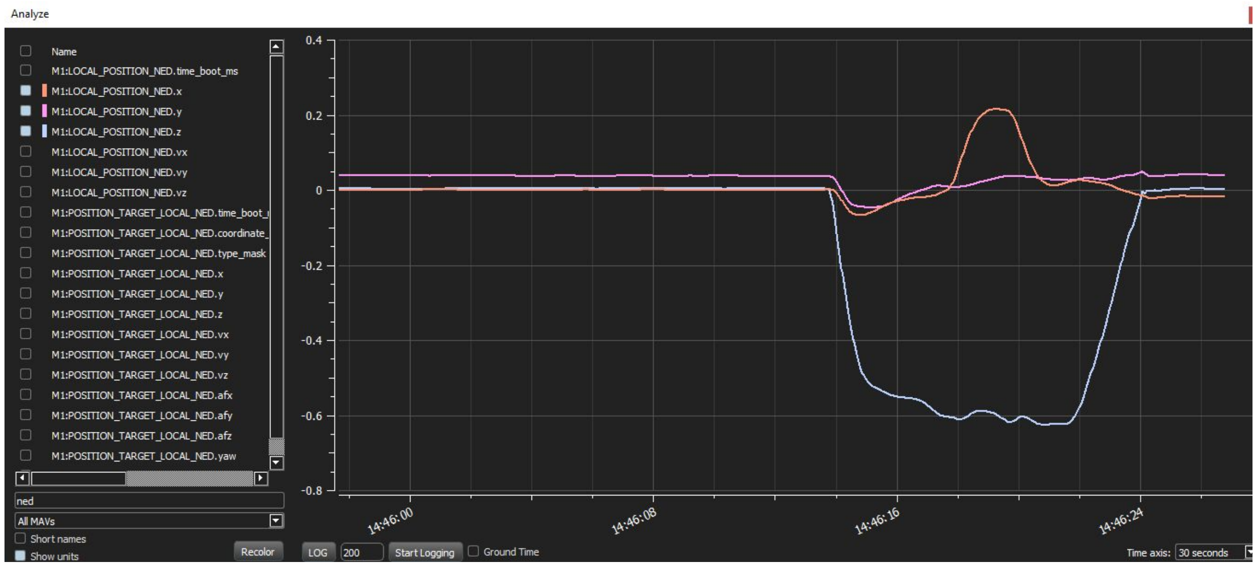
Finally, hold the thrust stick down on your joystick and run the code with

```
./offboard udp://:14553
```

./offboard runs the executable file. The string after is an argument specifying how the script should connect to the drone. Note that this port is opened automatically at boot by the mavproxy script (Transport_Procotol/Quad_Environment/launch_mavproxy_rcb.py) on your Raspberry Pi

If everything goes well, you quadcopter should arm, rise to 75 cm, hold, move 20 cm forward and backward, and land. **You need to keep the thrust command on your joystick low at all time during the script to allow the drone to arm.** This is a safety feature.

You can see a video example [here](https://www.youtube.com/watch?v=YLWoQu2SE-M) (https://www.youtube.com/watch?v=YLWoQu2SE-M), and the position of the drone during the flight as shown in Qgroundcontrol in the image below.



Dronecode supports multiple vehicles, and you can extend it with the full power of C++ and Python.

For extensive development, we recommend that you follow the instructions in the Dronecode documentation. You will learn to run your code in simulation before flight testing it, which can reduce accidents.

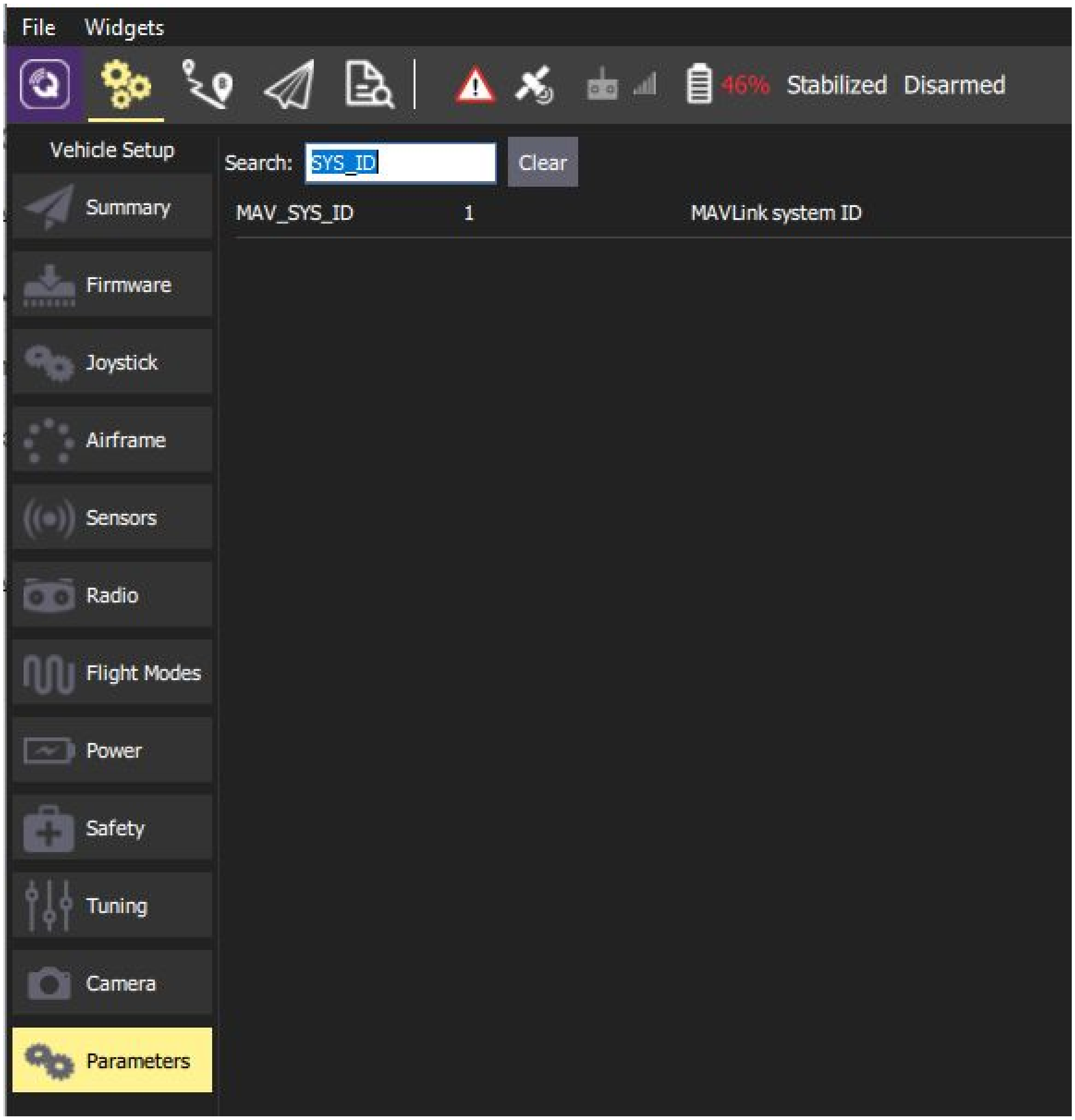
Note on Dronecode SDK:
Please note that your dronecode version is actually a fork: we added an example of position control in offboard mode, and we commented line 60 to 65 of the file /DronecodeSDK/plugins/action/action_impl.cpp to allow flight without GPS.

Multi-quadcopter flight

Warning: if you have multiple Otus trackers connected to your computer, you have to setup the flag *--controller_id* on your quadcopter as explained below, or the flight controller will receive conflicting position information.

Setup

A single QgroundControl instance can connect to multiple quadcopters. However, the system ID of each quadcopter has to be different, so Qgroundcontrol can differentiate between the messages received. To change the system ID, connect to a single quadcopter, and go in *Parameters* in Qground Control. Search SYS_ID in the parameter search box. Set a different ID for each quadcopter.



You also need to change the position forward script on each quadcopter so the flight controller receives position messages from a single quadcopter. Connect to your drone through the

Cloud9 editor (*ipaddress:8080* in a browser) and open the file *launch_pose_forward_rcb.sh* in the folder *pi/autolaunch_scripts*. As indicated in the script comments, add the following at the end of line 4:

--controller_id 4X4X4X4X

Where 4X4X4X4X is the controller id of the Otus tracker, found under the Otus tracker or in RCbenchmark tracking lab.

You can now connect normally to your drone with Qground control.

Fly with multiple drones

Flying with multiple drones can be done manually with multiple computers on which Qgroundcontrol is installed. In practice however, you will probably want to fly in an automated manner, as explained in the section above. The quadcopters all share a common frame of reference, so position commands will be consistent. The RCB_pose_forward script receives all position messages, but the flag --controller_id you setup above acts as a filter, so the PX4 software receives only the information from one Otus tracker.

The structure of your software depends heavily on your application. You can use a client-server interface to synchronize your Dronecode SDK control scripts from a single computer. Alternatively, a single quadcopter can be the master, and the others follow its lead. You can also use ROS to structure your program. Please contact RCbenchmark if you need help of feedback on your architecture.

Troubleshooting

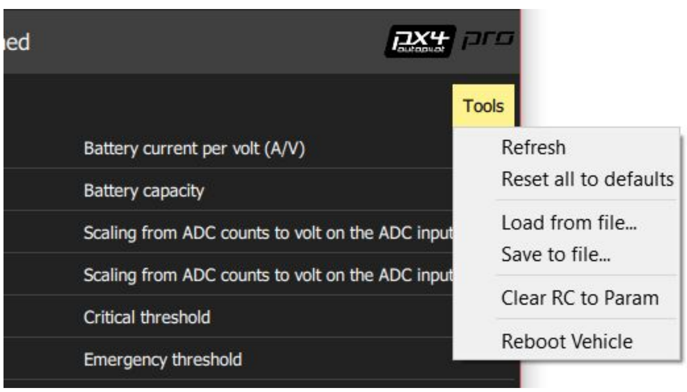
Loss of Tracking

Loss of tracking can happen due to base stations issues, vibrations, wireless interference, or power issue. First, check if you have a good tracking in RCbenchmark tracking lab. If the tracker loses position, it will default to the position of one of the base stations in the RCbenchmark tracking lab software.

- Base station or tracker issues
    - The base stations have to be view of each other (green LED) and at less than 5 m of each other. This is so the base stations are synced. The base stations should also be mounted firmly.
    - Large mirrors in the room must be covered
    - Base stations are too far. The maximum recommended distance between the base station is 5 m. The minimal distance between the base stations and the tracker is 75 cm. You may get a sync error message in SteamVR if the base stations are too far apart.
    - Strong direct sunlight can impact tracking.
    - Large mirrors or reflective surfaces in the room. Large windows could also act as mirrors.
    - Defective Otus tracker. Each Otus tracker is calibrated in the factory. Strong impacts may disconnect a sensor or change the geometry of the tracker, which reduces tracking reliability.
    - Defective base station: Unlikely. A defective base station will not show as solid green in StreamVR.
- Vibration issues: Check that you are using the included vibration damper and that no wires or components are touching the Otus tracker. Balance you propellers.
- Wireless interference: Environments with strong 2.4 Ghz interference (such as trade shows) can affect the downlink from the tracker, or the uplink to the Raspberry Pi.
    - Bring the wireless dongle and the router closer to the quadcopter.
    - Insure you are using a wired connection between the router and the ground computer.
    - Move to an environment with less 2.4 Ghz interference
    - Turn off 2.4 Ghz transmitters.
- Power issues: Those are rare. Check the wiring for damage.

Message relating to EKF yaw error or acceleration bias in Qgroundcontrol

In some cases, the initial orientation transmitted to the drone from the Otus tracker has an offset relative to the orientation of the quadcopter. Use the ‘’Reset axis’’ button in the RCbenchmark tracking lab software and wait for the EKF filter on the Pixhawk to converge. Make sure you are using the right controller ID to reset the local axis, especially if you have multiple controllers. In Qgroundcontrol, check the LOCAL_POSITION_NED yaw, pitch and roll. Look for differences of 1.5 or 3.1 rad (Pi or Pi/2 rad) which would indicate a bad local frame of reference of the tracking system. You can reset the Pixhawk in Qgroundcontrol in the ‘Parameter section’ by clicking the ‘Tool’ button and the ‘Reboot Vehicle’ option at the top right of the interface.



Error code from the Pixhawk (Red, yellow, green LED)

Please check the PX4 documentation here:
https://docs.px4.io/en/getting_started/led_meanings.html

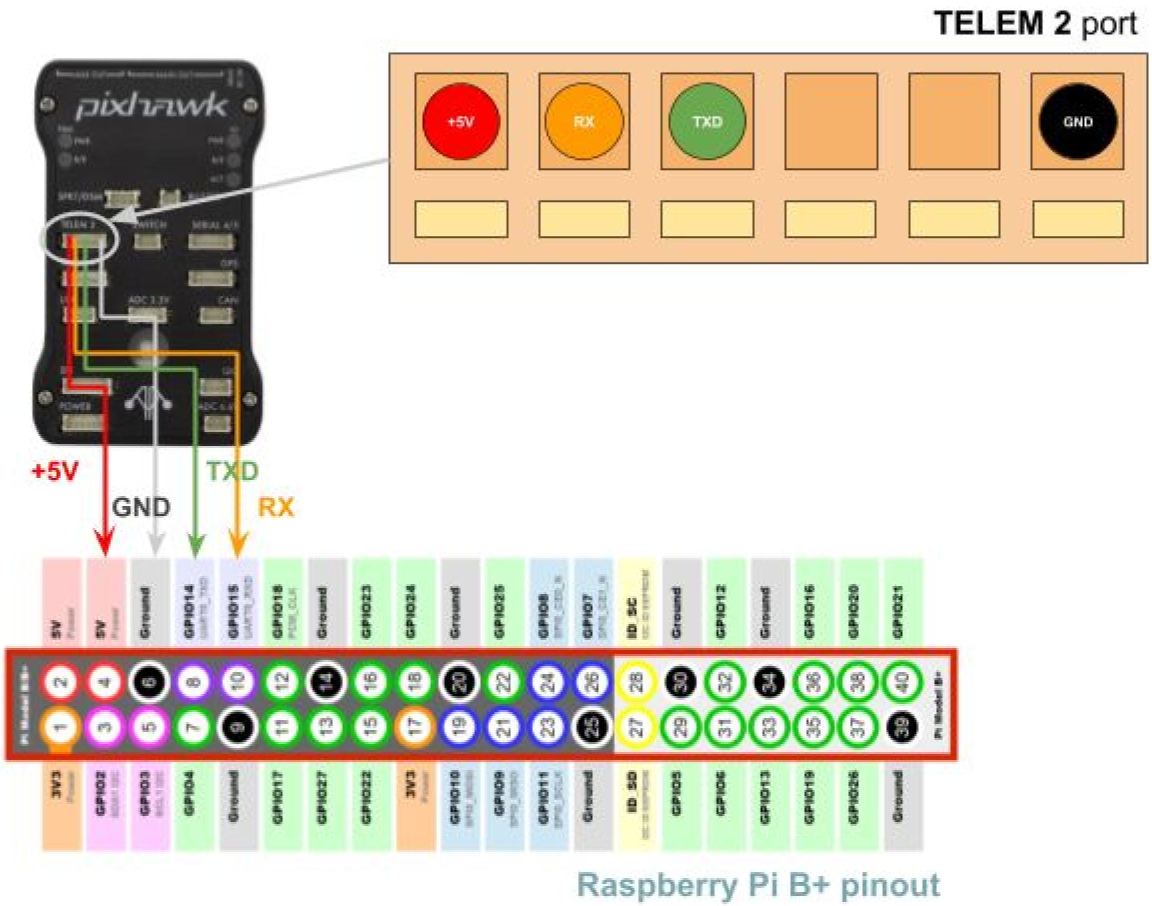
Reading Unstable in QgroundControl with Multiple Quadcopters

If the System ID of the Pixhawk is not set, Qground control will receive information from multiple vehicles at once. Please set a different System ID (SYS_ID) for each quadcopter by changing the SYS_ID parameter in Qgroundcontrol, as explained in the *Multi-quadcopter setup* section.

Annex A Electrical Connection Reference

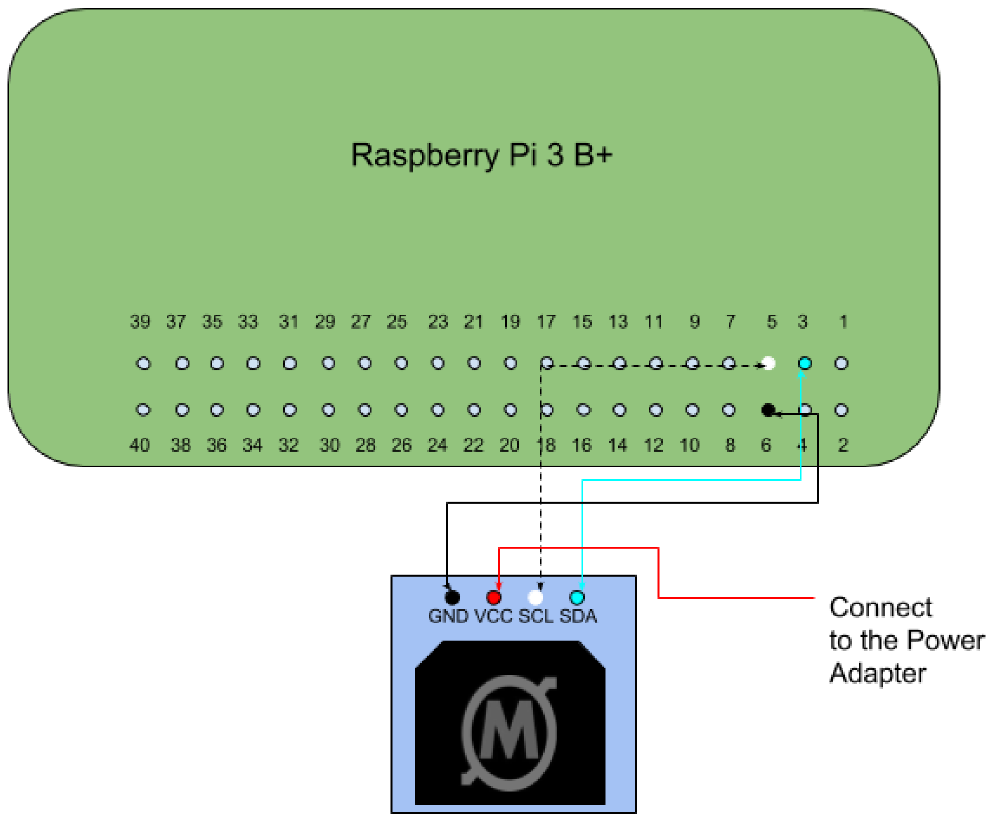
Serial connection between Pixhawk & Raspberry Pi

During flight, it is prefered to use serial connection as it is far more stable than USB. Use the following diagram to connect the Pixhawk flight controller to the Raspberry Pi B+. The Pixhawk should have come with a cable that you may cut and solder jump wire connectors (#26).



Connecting the LCD/OLED Display

The OLED display is driven by I2C. Take a 4 pin female to female Dupont connector (which has individual 1 pin connector on the other side) or four 1 pin dupont female to female connectors and connect to the the 4 pins on the OLED/LCD Display as depicted below.


Raspberry Pi 3 B+
39 37 35 33 31 29 27 25 23 21 19 17 15 13 11 9 7 5 3 1
40 38 36 34 32 30 28 26 24 22 20 18 16 14 12 10 8 6 4 2
GND VCC SCL SDA
Connect
to the Power
Adapter

Annex B: Adding Software to the Raspberry Pi

You may need to add software to the Raspberry Pi. For example, you may need to add a specific python libraries. We recommend that you start from a fresh copy of the operating system we provide, add your software, and make a new copy. This way, you always have a clean start for your developers. You can even automate the process by writing a script of the software setup. This is how we build the image for the Raspberry Pi. You can check the script installer_part1.sh and installer_part2 in the Quad_Environment folder of the Transport protocol github repository here:

https://gitlab.com/TytoRobotics/Transport_Protocol/tree/master/Quad_Environment

Those scripts can be useful if you want to build your own development environment.