# **Installation:**

Clone into this repository

> git clone <https://github.com/arkalim/openquad.git>

Inside the folder openquad, you will find a folder named installation\_files. Open it and you will find:

Common\_dependencies.sh

sos\_mavros.sh

GeographicLib

Then make Common\_dependencies.sh executable using

> chmod a+x Common\_dependencies.sh

Then run it using

> sudo ./Common\_dependencies.sh

Install Q ground control using

https://docs.qgroundcontrol.com/en/getting\_started/download\_and\_install.html

Then make it executable using

> chmod a+x ros\_mavros.sh

Then run it using

> ./ros\_mavros.sh

Then run

> cd ~/catkin\_ws/src/mavros/mavros/scripts

> ./install\_geographiclib\_datasets.sh

Now go the the given directory and see if there are 3 non empty folders

> cd /usr/share/GeographicLib

If they are empty, replace the original GeographicLib folder in usr/share with GeographicLib from openquad.

> sudo cp ~/openquad/installation\_files/GeographicLib /usr/share/GeographicLib

If you have lock symbols on src folder and catkin\_ws folder then this means you don’t have ownership of those folders and cannot build.

Do the following

> cd ~

> sudo su

You will get root access using this command

> sudo chown -R sarthak catkin\_ws

> sudo chown -R sarthak src

This is the username

Install gstreamer

> sudo apt-get install libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev

Install xterm terminal

> sudo apt install xterm

Install pygame

> sudo pip2 install pygame

Now, Build

> cd ~/catkin\_ws

> catkin build

Modify the bashrc file to have these at the end

source /opt/ros/melodic/setup.bash

source ~/catkin\_ws/devel/setup.bash

# Local use

export ROS\_HOSTNAME=localhost

export ROS\_MASTER\_URI=http://localhost:11311

# Connecting multiple computers. My computer is the master

# export ROS\_HOSTNAME=192.168.43.161

# export ROS\_MASTER\_URI=http://192.168.43.161:11311

Go to home directory and press CTRL+H you will see a .ignition folder and open it. Go inside fuel folder and open config.yaml.

Change url: <https://api.ignitionfuel.org> to url: [https://api.ignitionrobotics.org](https://api.ignitionfuel.org)

# **Running MAVROS with Gazebo in Simulation:**

### **Running individual commands:**

Run this in terminal. This will activate the mavros

> roslaunch mavros px4.launch fcu\_url:="udp://:14540@localhost:14557"

This command will simulate the PX4

> cd ~/src/Firmware

> no\_sim=1 make px4\_sitl\_default gazebo

This command will export the required packages for running Gazebo

> cd ~/src/Firmware

> source Tools/setup\_gazebo.bash $(pwd) $(pwd)/build/px4\_sitl\_default

> export ROS\_PACKAGE\_PATH=$ROS\_PACKAGE\_PATH:$(pwd)

> export ROS\_PACKAGE\_PATH=$ROS\_PACKAGE\_PATH:$(pwd)/Tools/sitl\_gazebo

**Run any one of the below commands to launch the required world in Gazebo:**

Empty world in Gazebo

> roslaunch gazebo\_ros empty\_world.launch world\_name:=$(pwd)/Tools/sitl\_gazebo/worlds/iris.world

This command will run an empty world in Gazebo with iris drone having front facing FPV camera

> roslaunch gazebo\_ros empty\_world.launch world\_name:=$(pwd)/Tools/sitl\_gazebo/worlds/iris\_fpv\_cam.world

This command will run an empty world in Gazebo with iris having optical flow sensor

> roslaunch gazebo\_ros empty\_world.launch world\_name:=$(pwd)/Tools/sitl\_gazebo/worlds/iris\_opt\_flow.world

Obstacle Avoidance world in Gazebo

> roslaunch gazebo\_ros empty\_world.launch world\_name:=$(pwd)/Tools/sitl\_gazebo/worlds/iris\_obs\_avoid.world

Lidar world in Gazebo

> roslaunch gazebo\_ros empty\_world.launch world\_name:=$(pwd)/Tools/sitl\_gazebo/worlds/iris\_rplidar.world

Vision world in Gazebo

> roslaunch gazebo\_ros empty\_world.launch world\_name:=$(pwd)/Tools/sitl\_gazebo/worlds/iris\_vision.world

IR Lock world in Gazebo

> roslaunch gazebo\_ros empty\_world.launch world\_name:=$(pwd)/Tools/sitl\_gazebo/worlds/iris\_irlock.world

After executing one of the above instructions, run Q Ground Control

To kill all the simulation, run these commands in terminal

> killall gzserver

> killall gzclient

### Running using sh files (recommended):

Clone the repository for sh files

> git clone https://github.com/arkalim/openquad.git

Run these commands in terminal

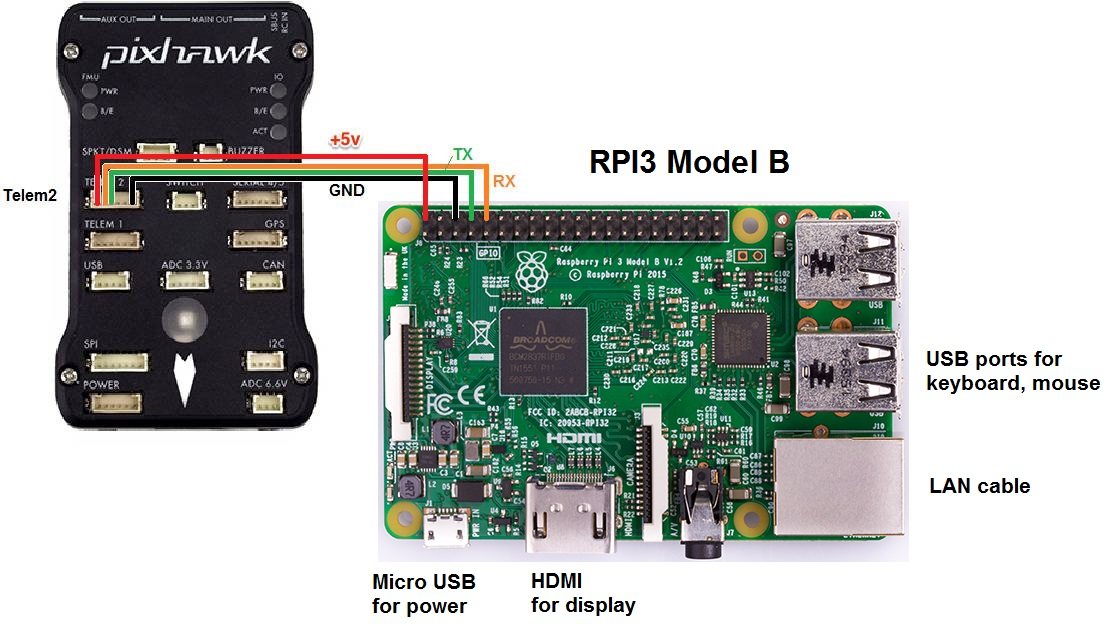
> cd ~/catkin\_ws/src/openquad/sh\_files

> ./simulation.sh

# **MAVROS on PC and communicating with Pixhawk using Raspberry Pi3:**

## **Hardware Setup:**

### **Connecting Raspberry Pi to Pixhawk using Serial Port:**



**Note: Do not connect the 5V pin as shown in the diagram, if you are powering the pixhawk and Raspberry Pi separately (recommended)**

### **Connecting Raspberry Pi to Pixhawk using USB Port:**

Use a USB cable to connect the Raspberry Pi to Pixhawk

## **Software Setup:**

Run these to install required packages in the Raspberry Pi

> sudo apt-get update

> sudo apt-get install screen python-matplotlib python-opencv python-pip python-numpy python-dev libxml2-dev libxslt-dev python-lxml

> sudo pip install future

> sudo pip install pymavlink==2.2.21

> sudo pip install mavproxy==1.7.7

SSH into Raspberry Pi and run

> sudo raspi-config

Go to Interfacing Options -> Serial

Select No for the first screen and Yes for the second screen.

Press OK and reboot the Raspberry Pi

Run this command in terminal

> sudo nano /boot/config.txt

Add this line to the bottom of this file

enable\_uart=1

## **Running:**

### **When Raspberry Pi and Pixhawk are connected using Serial Port.**

SSH into Raspberry Pi and this command in a terminal in Raspberry Pi. This command will connect the Serial Port S0 to the IP address 192.168.43.203 on port 14557.

> sudo mavproxy.py --master=/dev/ttyS0 --baudrate 57600 --out "udp:192.168.43.203:14557"

This is the IP address of the computer running the mavros (PC running Ubuntu 18.04).

Run this command in the terminal in your PC. This will run Mavros and connect to IP address 192.168.43.203 (localhost) on port 14557.

> roslaunch mavros px4.launch fcu\_url:="udp://:14557@localhost"

This is the address of the Serial port. To find this port address run:

> ls /dev/tty\*

This is the IP address of the computer running the mavros (PC running Ubuntu 18.04).

This the port number for communication. This must be the same in both the commands.This is the local IP address of the PC. In this case it is 192.168.43.203

### 

### **When Raspberry Pi and Pixhawk are connected using USB Port.**

SSH into Raspberry Pi and this command in a terminal in Raspberry Pi. This command will connect the USB Port AMA0 to the IP address 192.168.43.203 on port 14557.

> sudo mavproxy.py --master=/dev/ttyAMA0 --baudrate 57600 --out "udp:192.168.43.203:14557"

This is the IP address of the computer running the mavros (PC running Ubuntu 18.04).

Run this command in the terminal in your PC. This will run Mavros and connect to IP address 192.168.43.203 (localhost) on port 14557.

> roslaunch mavros px4.launch fcu\_url:="udp://:14557@localhost"

This is the address of the USB port. To find this port address run:

> ls /dev/tty\*

This is the IP address of the computer running the mavros (PC running Ubuntu 18.04).

This the port number for communication. This must be the same in both the commands.This is the local IP address of the PC. In this case it is 192.168.43.203

### Now you can run MavROS codes on PC and your Pixhawk will respond accordingly.