**Installation (ROS and Mavros)**

1. **On Raspberry pi 3B+**
2. **On Ubuntu Laptop\*** 
   1. Install ROS (kinetic, binary installation) on your ubuntu 16.04 laptop.
      1. Melodic not supported on Ubuntu 16.04
      2. <http://wiki.ros.org/kinetic/Installation/Source>
   2. Install Mavros package using binary installation
   3. Install Gazebo on Ubuntu

Note: You can first create all the nodes on laptop and then copy that on raspberry pi.

**Installation SSH (RPi and Laptop)**

**Creating a package and node on RPi**

1. Create a new catkin workspace

mkdir -p px4\_rpi\_ws

mkdir -p px4\_rpi\_ws/src

cd px4\_rpi\_ws

catkin init // initialize catkin

wstool init src // register source file with ros

1. build your catkin workspace

catkin build

1. source it

source devel/setup.bash

1. Move to src folder
2. Create a ros package

catkin\_create\_pkg [name of package] [dependencies]

Ex: catkin\_create\_pkg mav\_offb\_control std\_msgs rospy roscpp

1. You can add dependencies later in the cmake file of the package
2. Move to package -> src
3. Create a cpp node

Ex: nano offb\_node.ccp

1. Write your code here
2. You can create multiple node/code in one package
3. Go back to package folder and open cmake file
4. Add dependencies if needed
5. Add executable for each node, give a command name (ex. offboard\_command).
6. Go back to your workspace folder and build it and source it
7. Repeat the above step every time you change anything the workspace

**Establishing Connection with Pixhawk**

1. Connect Pixhawk with Raspberry Pi via USB
2. Open a terminal and type and run: roscore (it starts ROS master)
3. Open new terminal
4. Connect with Pixhawk:

rosrun mavros mavros\_node \_fcu\_url:=/dev/tty/ACM0:921600

1. Use rostopic to public/see messages
2. Use rosrun (package) (nodes) to run nodes

Ex: rosrun mav\_offb\_control savedata\_commands

**Test your script in Gazebo with ROS**

1. Go to px4 firmware folder and start gazebo simulation:

make px4\_sitl gazebo\_solo (or gazebo\_plane)

1. Run ‘roscore’ in one terminal
2. Link ros with gazebo (using local host)

roslaunch mavros px4.launch fcu\_url:="udp://:14540@127.0.0.1:14557"

1. Build and source your package (go to workspace: source devel/setup.bash)
2. Run your node

rosrun mav\_offb\_control offb\_commands

1. Let your node run and see simulation
2. You can give command manually also using rostopic

**HILS**

1. run roscore

2. general setting of qGC: check only udp

3. mavros mavros\_node \_fcu\_url:=/dev/ttyACM0:921600 \_gcs\_url:=udp://@127.0.0.1

4. do everything req on qgc

5. select HILS model

6. enable HITL in safety qgc

7. firmware: gazebo Tools/sitl\_gazebo/world/plane.world

**MAVROS Topics**

References:

1. <http://wiki.ros.org/ROS/Tutorials>
2. <https://github.com/mavlink/mavros/tree/master/mavros#installation>
3. <http://wiki.ros.org/mavros>
4. <https://dev.px4.io/v1.9.0/en/ros/>
5. <http://wiki.ros.org/geometry_msgs>
6. <http://wiki.ros.org/sensor_msgs>
7. <http://wiki.ros.org/nav_msgs>
8. <http://wiki.ros.org/mavros_msgs>