**ANACONDA AND DRONE SIMULATION**

**Anaconda Setup**

1.Setup by following the steps here: <https://docs.anaconda.com/anaconda/install/linux/>

***Skip step 2 in the doc link above***

When installing Debian package as the very step in the document, if they aren’t installing, individually install all the packages mentioned in the given command in step 1 of the documentation.

When installing pycharm from the above link, install the “exe” version and not the zip file and make sure to run: **source /home/meghna/anaconda3/bin/activate (to get /home/meghna/anaconda3 – type pwd after opening terminal in anaconda3 folder)**

**Environment Setup**

After Installing Anaconda

Note:Execute the following command: conda config --set auto\_activate\_base False

Whenever we install Anaconda there is a 'base' environment which is always active by default when you open any new Terminal Session. To stop this from happening we execute the above command.

Close and reopen the terminal to see the change.

1. Open up terminal

2. Create a new virtual environment with the following command:

“conda create -n mydroneenv python=2.7”

#To deactivate environment: conda deactivate

To activate environment: conda activate mydroneenv //or whatever is the name of your environment

Now install the following dependencies to setup the required environment for our work

ii. pip intall opencv-python

iii. sudo apt-get install python-dev

iv. pip install matplotlib

v. pip install pygame

vi. pip install pygtk ***#might give error, just ignore and move ahead***

vii. pip install lxml

viii. pip install pyyaml

ix. conda install -c anaconda wxpython

x. pip install mavproxy

xi. pip install dronekit

xii. pip install dronekit-sitl

3. Create a new directory named courseRoot(the name does not matter) for you to store all files we will be using in the future

4. Create a subdirectory inside courseRoot name 'apm' and execute the following in the apm directory

5. Install ardupilot with the following command:

git clone -b Copter-3.5.5 https://github.com/ardupilot/ardupilot

6. Now we will need to initialise the submodules within the ardupilot directory we just cloned. Follow the below steps:

cd ardupilot //change into the ardupilot directory

git submodule update --init –recursive

**QGroundControl Installation**

1. Go to the following link:

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

Go To the Ubuntu Linux Section and download the "QGroundControl.AppImage"

2. Now follow the following commands:

1. Go to the directory courseRoot

2. Create a subdirectory 'src'

3. Copy the App Image from the Downloads folder to this Folder

4. Now to make this file executable do:

chmod +x QGroundControl.AppImage

5. Adding the program into our path environment variable so that we can execute it from any location:

sudo cp QGroundControl.AppImage /usr/local/bin/QGC.AppImage

6. Now we can type QGC.AppImage and start QGroundControl station from anywhere

3. Connecting QGroundControl to the SITL Drone

1. Click on the 'Q' Button on the top left corner

2. Go To Comm Links

3. Click the add button

4. Under type change to TCP from the drop down box

5. Under Host Address Enter in our local host i.e. 127.0.0.1

6. In TCP Port Enter 5763/5762 (add whichever works – usually it is 5762)

7. Now click on that port and hit connect

**Bash Script Code**

We will create a bash script to launch our python code and QGround Control station together in one command instead of having to execute them one by one manually.

Note: Make sure you are in the courseRoot directory (or whatever other name you might have given to this folder)

1. Enter "whereis dronekit-sitl" in the terminal and get the path location of dronekit sitl: /home/meghna/anaconda3/envs/mydroneenv/bin/dronekit-sitl

2. Enter "which QGC.AppImage: in terminal and get the location of QGC App Image: /usr/local/bin/QGC.AppImage

3. Enter "which python" and get the location: /home/meghna/anaconda3/envs/mydroneenv/bin/python

4. Do the following steps:

1. gedit launchSitl

2. Enter the following code into the file and replace all highlighted portions with the paths obtained in steps 1, 2 and 3 respectively.

***Code***

#!/bin/bash

kill -9 $(ps -eF | grep QG | awk -F' ' '{print $2}') > /dev/null 2>&1

kill -9 $(ps -eF | grep ardu | awk -F' ' '{print $2}') > /dev/null 2>&1

kill -9 $(ps -eF | grep mav | awk -F' ' '{print $2}') > /dev/null 2>&1

kill -9 $(ps -eF | grep apm | awk -F' ' '{print $2}') > /dev/null 2>&1

##Launch a SITL Instance

/usr/local/bin/dronekit-sitl copter --home=44.5013,-88.0622,0,180&

sleep 5

##Launch QGroundControl

/usr/local/bin/QGC.AppImage 2>/dev/null&

sleep 5

##Start MavProxy

screen -dm mavproxy.py --master=tcp:127.0.0.1:5760 --out=127.0.0.1:14550 --out=127.0.0.1:5762

##Launch the dronekit-python script

/usr/bin/python "$1" --connect 127.0.0.1:5762

function finish {

kill -9 $(ps -eF | grep QG | awk -F' ' '{print $2}') > /dev/null 2>&1

kill -9 $(ps -eF | grep ardu | awk -F' ' '{print $2}') > /dev/null 2>&1

kill -9 $(ps -eF | grep mav | awk -F' ' '{print $2}') > /dev/null 2>&1

kill -9 $(ps -eF | grep apm | awk -F' ' '{print $2}') > /dev/null 2>&1

}

trap finish EXIT

5. Go to the courseRoot directory in the terminal

6. Execute the following command to make the "launchSitl" file executale:

chmod +x launchSitl

7. Now we place it in the the user path:

sudo mv launchSitl /usr/local/bin

**Executing python codes**

1. Open the terminal in “courseroot” directory and activate environment:

Conda activate mydroneenv

2. To execute any of the python files in the folder, type the command:

launchSitl filename.py