

The following process describes the setup and execution of the SITL on a Ubuntu 20.04 OS, using ROS Noetic. Other versions are not covered in this tutorial.

Docker setup

Begin by installing Docker (<https://docs.docker.com/engine/install/ubuntu/>). Preferably, install using apt package manager.

Add your user to the "docker" group. To add your user to the Docker group run

```
sudo usermod -a -G docker <your_user_name_here>
```

Logout/login or restart. You should now be able to to run

```
docker run hello-world
```

without sudo.

Download and unzip the files. Then, make all scripts executable

```
chmod +x *.sh
```

Run the installation script, it will download and install the files needed.

```
./install.sh
```

Create two folders named host and catkin_ws in the same folder as the install.sh file,

```
mkdir -p host
```

```
mkdir -p catkin_ws
```

These folders are shared between host machine and Docker container, and either side can access and edit files in it. Find the **path to these folders** in the **host** machine (not inside Docker), as it will be relevant in the next step.

Copy the world created (my.world) to the *host* folder (shared folder). This could be done manually in the file explorer software in a terminal by:

```
cp /path/to/my.world /path/to/host/
```

Copy the offboard package folder to the correct place in the catkin_ws (shared folder).

```
cp -r offboard_package /path/to/catkin_ws/offboard_package/
```

Docker execution

Using 5 terminals, run the 5 sets of commands below

Terminal 1 – execute the Gazebo simulator

```
./start.sh
```

```
export PX4_SITL_WORLD=~/.host/danger_zones.world
```

```
cd ~/PX4-Autopilot/ && make px4_sitl_default gazebo_typhoon_h480  
param set COM_RCL_EXCEPT 4
```

Terminal 2 – execute the PX4 firmware

```
./terminal.sh  
roslaunch mavros px4.launch fcu_url:="udp://:14540@127.0.0.1:14557"
```

Terminal 3 – monitor position

```
./terminal.sh  
rqt_plot /mavros/setpoint_raw/local/position/x:y:z /mavros/local_position/pose/pose/position/x:y:z
```

Terminal 4 – monitor velocity

```
./terminal.sh  
rqt_plot /mavros/setpoint_raw/local/velocity/x:y:z  
/mavros/local_position/velocity_local/twist/linear/x:y:z
```

Terminal 5 – execute the FB controller

```
./terminal.sh  
cd catkin_ws/offboard_package/  
source devel/setup.bash  
roslaunch offb your_fb_controller.py
```

The terminal 5 is responsible for executing the offb controller implemented and plotting the top view figure with the trajectory.