**Hướng dẫn phát triển dự án Swarm Done**

**Nền tẳng: ubuntu 22.04 ( Ram 8-16Gb, Rom 60Gb)+ card đồ họa rời.**

**Tổng quan về các gói cài đặt cần thiết:  
Gói 1: PX4-Autopilot và Gazebo Ros2 ( mục đích sử dụng để mô phỏng cho các drone thật, nghiên cứu các hàm, thư viện MAVSDK)  
Gói 2: MAVSDK( Mục đích sử dụng là thư viện chính để sử dụng các hàm kết nối, điều khiển,…)**

**Gói 3: mavsdk\_server và Mavlink- router ( Mục đích sử dụng là chia cổng uart( Px4-1..6) thành 2 cổng với khả năng sử dụng dữ liệu song song giữa phần mềm QGC và App xây dựng bằng Python)**

**Gói 4: PyQt5 ( Mục đích sử dụng là xây dựng giao diện trên App trực quan, đơn giản)**

**Gói 5: Các gói phụ trợ khác:( tùy theo cấu hình máy cần thêm gói nào sẽ cài đặt thêm gói đó để cho các gói có khả năng kết hợp và cài đặt trên ubuntu 22.04)**

**\*\*\*\*\*  
Cài đặt gói 1 ( Thực hiện trình tự các lệnh sau trên bash)( hoặc xem tại đường dẫn:** [**https://docs.px4.io/v1.12/en/dev\_setup/dev\_env\_linux\_ubuntu.html**](https://docs.px4.io/v1.12/en/dev_setup/dev_env_linux_ubuntu.html) **)**

**a) PX4-Autopilot**

**git clone https://github.com/PX4/PX4-Autopilot.git --recursive**

**git clone https://github.com/PX4/PX4-Autopilot.git –recursive**

**bash ./PX4-Autopilot/Tools/setup/ubuntu.sh**

**bash ./PX4-Autopilot/Tools/setup/ubuntu.sh**

**git submodule update --init --recursive**

**b)Gazebo Ros2 ( xem đường dẫn :** [**https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html**](https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html)**)( kiểm tra phiên bản ros: rosversion -d)**

**cài gazebo9:**

**sudo apt update**

**sudo apt upgrade**

**sudo apt install gazebo9**

**sudo apt install ros-<YOUR-ROS-DISTRO>-gazebo-ros-pkgs**

**Thay <YOUR-ROS-DISTRO> bằng phiên bản ROS mà bạn đang sử dụng, ví dụ: noetic hoặc melodic**

**(sudo apt install ros-humble-gazebo-ros-pkgs)**

**Cài đặt gói 2&3: MAVSDK(mavsdk\_server) & MAV-router**

**asdada asd121325457**

**a) cài đặt MAVSDK(mavsdk\_server)**

**pip install mavsdk**

***pip install asyncio***

***b) cài đặt mavlink -router***

git clone https://github.com/intel/mavlink-router.git

cd mavlink-router

git submodule update --init --recursive

**sudo apt install git meson ninja-build pkg-config gcc g++ systemd**

**cd**

sudo pip3 install meson

meson setup build

ninja -C build

**sudo ninja -C build install**

**Cài đặt gói 4 : QT5**

Cài qt designer 5:

sudo apt-get install python3-pyqt5

sudo apt-get install qttools5-dev-tools

sudo apt-get install qttools5-dev

**Hướng dẫn mô phỏng với Gazebo và Jmavsim**

1. **Với Mô phỏng Gazebo ( cd PX4-Autopilot hoặc truy cập file chạy terminal)**

**Hướng dẫn (https://docs.px4.io/main/en/sim\_gazebo\_classic/multi\_vehicle\_simulation.html)**

**[https://docs.px4.io/main/en/dev\_setup/dev\_env\_linux\_ubuntu.html#simulation-and-nuttx-pixhawk-targets](https://docs.px4.io/main/en/dev_setup/dev_env_linux_ubuntu.html" \l "simulation-and-nuttx-pixhawk-targets)**

**fix**

**(Make clean)**

**(make clean**

**make distclean)  
(sudo apt-get install libgstreamer-plugins-base1.0-dev gstreamer1.0-plugins-bad gstreamer1.0-plugins-base gstreamer1.0-plugins-good gstreamer1.0-plugins-ugly -y) gỡ lỗi**

**Bước 1: lấy môi trường mô phỏng là Gazebo :**

**make px4\_sitl\_default gazebo-classic ( với lần đầu tiên)**

**Bước 2 Chạy mô phỏng mới N drone quard:( chạy cái này là ok) ./Tools/simulation/gazebo-classic/sitl\_multiple\_run.sh -n 6 -m iris ( Trong đó 5 là số lượng drone cần mô phỏng, iris là loại máy bay cần mô phỏng)**

**Bước 3: Chạy MAVSDK\_server( mỗi lệnh 1 Bash riêng)**

.local/lib/python3.10/site-packages/mavsdk/bin/mavsdk\_server udp://:14541 -p 50062

.local/lib/python3.10/site-packages/mavsdk/bin/mavsdk\_server udp://:14542 -p 50063

.local/lib/python3.10/site-packages/mavsdk/bin/mavsdk\_server udp://:14543 -p 50064

.local/lib/python3.10/site-packages/mavsdk/bin/mavsdk\_server udp://:14544 -p 50065

.local/lib/python3.10/site-packages/mavsdk/bin/mavsdk\_server udp://:14545 -p 50066

.local/lib/python3.10/site-packages/mavsdk/bin/mavsdk\_server udp://:14546 -p 50067

**Bước 4: tương tự như bước 3 khi chạy với các drone thực tế**

**B Với Mô phỏng Jmavsim**

* *Lệnh chạy drone mô phỏng trên JMAVSim, cụ thể trong* [*link này*](https://docs.px4.io/main/en/simulation/multi_vehicle_jmavsim.html)

Chạy lệnh 1: **make px4\_sitl\_default**  ( ý nghĩa :Đây là một lệnh sử dụng trong môi trường Unix-like để tự động hóa quá trình biên dịch và xây dựng dự án từ mã nguồn, chỉ cần chạy 1 lần ."SITL" thường là viết tắt của "Software-in-the-Loop", là một phương pháp kiểm thử trong đó phần mềm của UAV được chạy trực tiếp trên máy tính mà không cần thiết bị vật lý.)

Chạy lệnh 2: **./Tools/simulation/sitl\_multiple\_run.sh 2 ( ý nghĩa :Đây là đường dẫn tới một tập lệnh hoặc tập kịch bản script (được viết bằng ngôn ngữ như Bash) để chạy mô phỏng PX4 SITL với các cấu hình và thiết lập cụ thể. Và 2 là số lượng UAV cần khởi tạo)**

**Chạy các lệnh sau để khởi tạo các UAV giả lập với các địa chỉ UDP tương tứng là 4560++**

**//( Lưu ý là mỗi 1 UAV giả lập cần khởi tạo 1 Terminal mới)**

./Tools/simulation/jmavsim/jmavsim\_run.sh -l ( UAV 1)

./Tools/simulation/jmavsim/jmavsim\_run.sh -p 4561 -l (UAV2)

./Tools/simulation/jmavsim/jmavsim\_run.sh -p 4562 -l(UAV3)

./Tools/simulation/jmavsim/jmavsim\_run.sh -p 4563 -l(UAV4)

./Tools/simulation/jmavsim/jmavsim\_run.sh -p 4564 -l(UAV5)

./Tools/simulation/jmavsim/jmavsim\_run.sh -p 4565 -l(UAV6)

* *Lệnh chạy mavsdk\_server ( Sử dụng với cả 2 trường hợp mô phỏng là Gazebo và Jmavsim)( Mục đích tạo thêm cổng để kết nối với app)*

( với ubuntu 22 thì sẽ là 3.10, cd chạy ở bash ngoài, mỗi 1 server là 1 Teminal)

.local/lib/python3.10/site-packages/mavsdk/bin/mavsdk\_server udp://:14540 -p 50060

.local/lib/python3.10/site-packages/mavsdk/bin/mavsdk\_server udp://:14541 -p 50061

.local/lib/python3.10/site-packages/mavsdk/bin/mavsdk\_server udp://:14542 -p 50062

.local/lib/python3.10/site-packages/mavsdk/bin/mavsdk\_server udp://:14543 -p 50063

.local/lib/python3.10/site-packages/mavsdk/bin/mavsdk\_server udp://:14544 -p 50064

.local/lib/python3.10/site-packages/mavsdk/bin/mavsdk\_server udp://:14545 -p 50065

.local/lib/python3.10/site-packages/mavsdk/bin/mavsdk\_server udp://:14546 -p 50066

**Cách Kết nối QGC&APP Với các Drone thực tế như sau:**

***( Hướng dẫn này làm với 1 drone, các drone sau tăng các giá trị lên 1)***

***ví dụ : mavlink-routerd -t 5761 /dev/ttyACM0:57600 -e 127.0.0.1:14541 -e 127.0.0.1:14551***

***> mavlink-routerd -t 5761 /dev/ttyACM1:57600 -e 127.0.0.1:14542 -e 127.0.0.1:14552***

***.local/lib/python3.10/site-packages/mavsdk/bin/mavsdk\_server udp://:14541 -p 50061***

***> .local/lib/python3.10/site-packages/mavsdk/bin/mavsdk\_server udp://:14542 -p 50062***

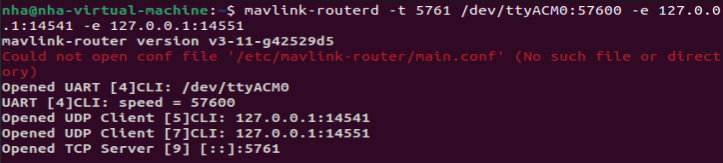
**+Kiểm tra port TCP đang mở: netstat -ltnp**

**+ Kiểm tra port UDP đang mở: netstat -lunp**

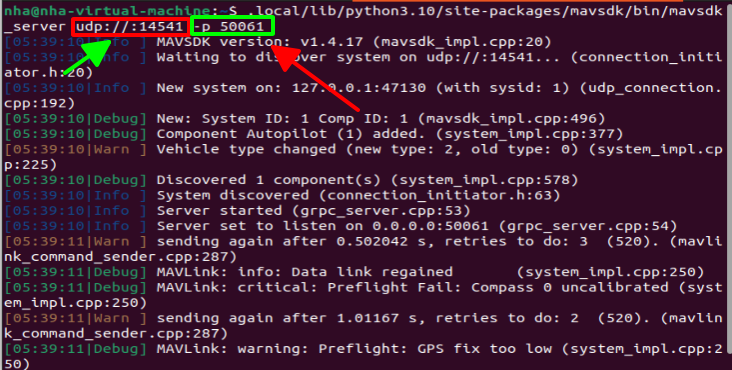
**+ls /dev/tty\* ( kiểm tra xem cổng uart nào đã được cắm vào Laptop)  
- Bước 1 : tùy vào số lượng drone kết nối để chạy chia luồng USB/ACM thành 2 cổng TCP ( Ý nghĩa : giúp việc đọc và gửi dữ liệu diễn ra song song với các hoạt động khác trong QGC và APP qua giao tiếp MAVLink bằng cách sử dụng mavsdk\_server) ( Trường hợp bị lỗi không thực hiện được lệnh 2 ngay thì open QGC kết nối Drone bằng Com ACM tương ứng tước rồi mới thực hiện lệnh 2)**

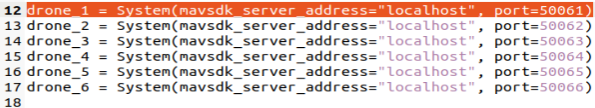
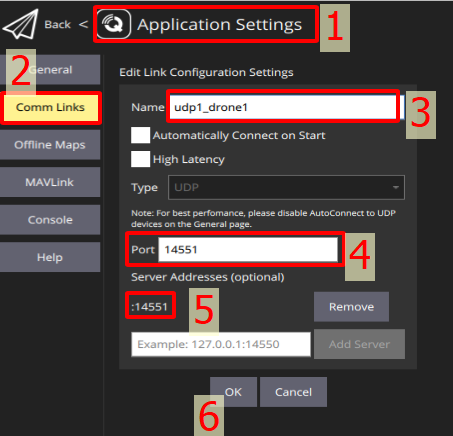
**Lệnh 1 trên bash)**

**mavlink-routerd -t 5761 /dev/ttyUSB0:57600 -e 127.0.0.1:14541 -e 127.0.0.1:14551**

**(Kết quả như vậy là thành công)**

**Lệnh 2 trên bash) .local/lib/python3.10/site-packages/mavsdk/bin/mavsdk\_server udp://:14546 -p 50061**

****

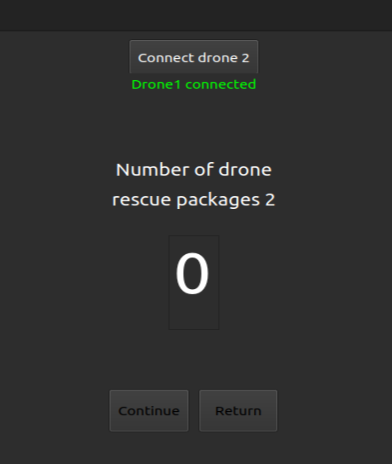
- Với APP đã khai báo Drone 1 được định danh Port 50061

- Với QGC hủy các kết nối tự động và kết nối bằng tay Application Settings Và chú ý phải điền đúng server và port.

Bước 2: Khởi động QGC và các thiết lập kết nối bằng tay với Application setting > Kết nối Drone1 vào QGC ( chú ý Drone 1 phải được cài đặt tham số MAV\_SYS\_ID =1 ( với 1 được định nghĩa là phương tiện bay số 1( Vehicle 1)

Bước 3:Khởi động APP đã được xây dựng từ trước để thực nghiệm hoặc phát triển các hàm ( tên thư mục : MultiDrone-6Drone, đường dẫn Project có thể download tại driver sau:<https://drive.google.com/drive/folders/1g7iezQF3GCz411pmrT7P3O4vfXcAVt62?usp=drive_link> )

Mở project và khởi chạy project : Python3 MyApp.py  
(



sudo apt install make

sudo apt install git

sudo apt install gazebo

**bash ./PX4-Autopilot/Tools/setup/ubuntu.sh**

**git submodule update --init --recursive**