

# **Simulation d'un drone via dronekit-sitl**

Prepared by

**Bournousouzis Christos**

May 2021

## Objectif

L'objectif de ce rapport est tout simplement de pouvoir simuler entièrement un drone et de pouvoir observer ses actions via une station de contrôle.

Le code qui va suivre sera écrit suivant la librairie dronekit, n'hésitez pas à aller consulter le rapport concernant cette librairie pour saisir mieux la suite.

## Configuration du simulateur

Le simulateur que nous allons utiliser est dronekit-sitl qui est un simulateur configurable avec la librairie dronekit. Son installation est très simple :

```
pip install dronekit-sitl
```

Attention que dronekit-sitl est encore relativement expérimental et il n'y a que quelques véhicules préconstruits (dont certains sont assez vieux et/ou instables).

Pour lancer le simulateur, il suffit de lancer la commande suivante :

```
dronekit-sitl copter
```

Cela lance un simulateur de drone type « copter » étant à des coordonnées GPS par défaut. Pour mettre des coordonnées GPS que l'on souhaite, il suffit d'ajouter à la suite de la commande un paramètre « --home » comme :

```
dronekit-sitl copter --home=-35.363261,149.165230,584,353
```

Le paramètre « --home » prend comme valeurs (lat,lng,alt,yaw).

Le simulateur démarre et attend une connexion TCP sur 127.0.0.1:5760. Une fois que quelque chose se connecte au port 5760, le simulateur ouvrira les ports 5762 et 5763 et attendra des connexions supplémentaires sur le port 5763 pour rouvrir d'autres ports (5766, 5769, etc.).

## Connecter un script dronekit

Pour faire tourner un script Python sur notre simulateur, il suffit de se connecter via TCP. Pour cela, il faut utiliser la fonction dronekit suivante dans le script :

```
vehicle = connect('127.0.0.1:5760', wait_ready=True)
```

## Connecter une station de contrôle

Il est également possible de connecter une station de contrôle au simulateur pour voir en temps réel les actions du véhicule simulé. Pour cela, il faudra avoir installé MAVProxy sur son système pour établir une liaison MAVLink entre la station de contrôle et le drone simulé.

Pour se connecter à une station de contrôle, il suffit de lancer la commande suivante (il faut avoir initialisé un simulateur avant bien évidemment) :

```
mavproxy.py --master=tcp:127.0.0.1:5760 --out=127.0.0.1:14550 --out=127.0.0.1:14551
```

Cela initialise une connexion MAVLink entre le simulateur et le système. Cette connexion sort en sortie 2 connexions localhost sur les port 14550 et 14551. Ces via un de ces ports que va pouvoir venir se connecter la station de contrôle à la connexion MAVLink, elle-même connectée au simulateur.

Le port restant servira à initialiser la connexion entre le script python et le simulateur. La fonction Python dronekit devient donc la suivante :

```
vehicle = connect('127.0.0.1:14550', wait_ready=True)
```

## Démonstration

Pour la démonstration, nous allons utiliser [l'exemple](#) « simple goto » de dronekit se trouvant sur leur github.

Une fois que vous avez téléchargé l'exemple, les étapes pour faire l'exemple sont les suivantes :

1. Initialiser une simulation de drone
2. Initialiser la connexion MAVLink
3. Connecter la station de contrôle
4. Lancer le script python « simple goto »

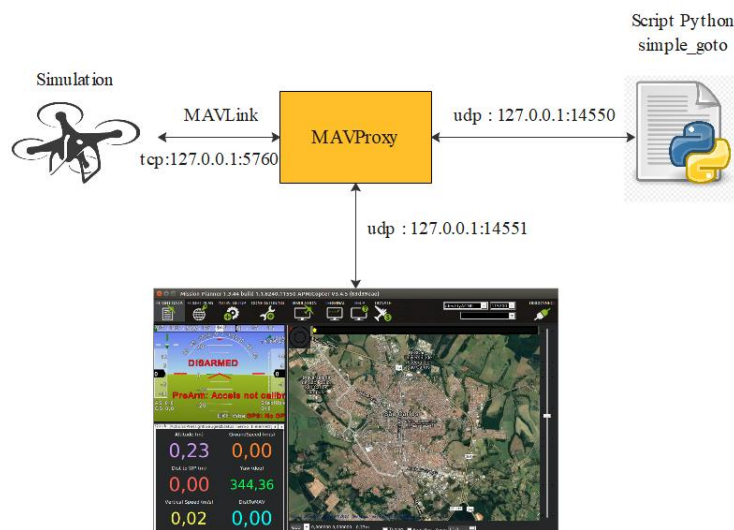


Figure 1 - Schéma de la Communication Simulateur-Script-Station de Contrôle

N'hésitez pas à vous référer à la vidéo d'exemple pour plus d'explications.

## Références

- [https://dronekit-python.readthedocs.io/en/latest/develop/sitl\\_setup.html](https://dronekit-python.readthedocs.io/en/latest/develop/sitl_setup.html)
- [https://dronekit-python.readthedocs.io/en/latest/guide/connecting\\_vehicle.html](https://dronekit-python.readthedocs.io/en/latest/guide/connecting_vehicle.html)
- [https://dronekit-python.readthedocs.io/en/latest/examples/simple\\_goto.html#example-simple-goto](https://dronekit-python.readthedocs.io/en/latest/examples/simple_goto.html#example-simple-goto)