

Fiche Activité 1

Prise en main du drone LiteWing

Ce document te guide étape par étape pour mettre en place l'environnement python permettant de programmer le drone, se connecter au drone, faire le « hello world », lire les mesures gyroscopique du drone, piloter avec le clavier / une manette le drone, mettre en place un pilote automatique simple.

Une *fiche ressource* associé à cette fiche activité apporte des informations supplémentaire sur certains aspects et quelques aide pour déboguer.

Pour toute question / hésitation / difficulté ou si vous êtes perdu **n'hésitez pas à demander**.

Etape 1 : Mise en place de l'environnement

Le drone LiteWing utilise un client procuré par le constructeur pour communiquer avec un ordinateur. Ce client (« cfclient ») permet de piloter le drone, de lire ses données, ect. Nous utiliserons une library python (« cflib ») qui permet à un programme python d'accéder à ce client et donc de piloter le drone. Il faut donc installé le client. Ce dernier est intégré dans l'environnement que j'ai mis en place, ainsi que les scripts que nous utiliserons.

Ouvrez un terminal dans un dossier « LiteWing » et faites les commandes suivantes :

```
git clone https://github.com/Arthur1459/LiteWing-Drone-Project.git
pip install -r requirements.txt
cd crazyflie-clients-python
pip install -e .
```

[Si « git » ne fonctionne pas ou n'est pas installé uniquement] : installation manuelle

- visitez à <https://github.com/Arthur1459/LiteWing-Drone-Project>
- cliquez sur « code » en vert puis sur « download zip »
- placer le .zip télécharger dans un dossier adapté (ex: Document) et décompressez le.
- Ouvrez un terminal à dans ce dossier et exécutez :

```
pip install -r requirements.txt
cd crazyflie-clients-python
pip install -e .
```

Verifiez l'installation en testant cfclient :

```
cd crazyflie-clients-python
cfclient
```

Etape 2 : Connection au drone

Verifier la fixation du drone sur un support stable permettant de maintenir le drone dessus et de le manipuler en toute sécurité.

Brancher la batterie au drone ou un cable USB-C de votre ordinateur, basculer l'interrupteur du drone sur ON. Des lumières apparaissent, les moteurs font un petit tour.

Sur votre ordinateur, **connectez vous au réseaux wifi du drone**

« LiteWing_1020BA1EADD1 », **mot de passe « 12345678 »**. **Attention à choisir le bon réseaux !** Chaque drone créer un réseaux wifi différent, connecter-vous au bon (observer quels réseaux apparait lorsque vous passez le drone en ON) (**une fois trouvé le bon, noter le nom du réseaux** car vous pourrez avoir à vous y reconnecter). Une fois connecter au réseaux wifi du drone, tout est bon.

Etape 3 : Faites tourner les moteurs avec « hello_litewing.py »

Ouvrez le fichier « hello_litewing.py » dans votre editeur de code ou créer le avec le code suivant :

```
import time
import cflib.crtp
from cflib.crazyflie import Crazyflie

# URI for your LiteWing drone
DRONE_URI = "udp://192.168.43.42"

# Initialize CRTP drivers
cflib.crtp.init_drivers()

# Create Crazyflie instance
connection = Crazyflie(DRONE_URI)

# Connect to the drone
print("Connecting to drone...")
connection.open_link(DRONE_URI)

# First send zero setpoint to unlock safety and arm drone
print("Sending zero setpoint to unlock safety...")
connection.commander.send_setpoint(0, 0, 0, 0)
time.sleep(0.1)

# Flight parameters
roll = 0.0
pitch = 0.0
yaw = 0
thrust = 10_000 # Thrust value is 10_000 minimum and 60_000 maximum
# Start motors
print("Starting motors at minimum speed...")
connection.commander.send_setpoint(roll, pitch, yaw, thrust)
#print(connection.log.log_blocks)
time.sleep(1)

# Stop the motors
print("Stopping motors...")
connection.commander.send_setpoint(0, 0, 0, 0)
time.sleep(0.1)

# Close the connection
connection.close_link()
```

Essayer de modifier la consigne de test envoyé au drone :

```
# Flight parameters
roll = 35.0
pitch = 0.0
yaw = 0
thrust = 30_000
```

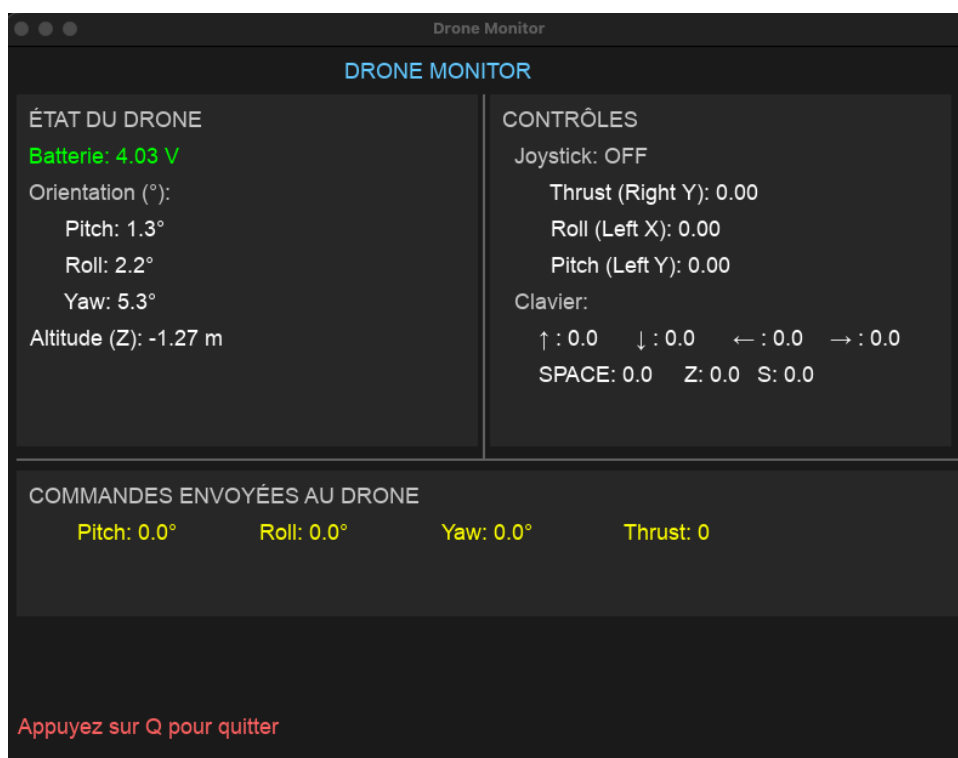
Etape 4 : Observation des mesures gyroscopiques

A tout moment, si le niveau de batterie descend trop bas (< 3.4V), arrêter le drone et mettre la batterie à charger.

Ouvrez le fichier « DroneMonitor.py » dans votre éditeur de code et mettez les paramètres suivants comme variable de configuration :

```
# --- Configuration ---
KEYBOARD = True
CONTROLLER = False
ACTIVATE_CONTROL = False
MANUAL_CONTROL = False
```

Puis, en étant connecté au drone, lancez le programme. Une fenêtre pygame s'ouvre :



Observer l'interface et les données affichées.

Prenez le support du drone dans votre main et orientez le drone dans différentes directions. Observer comment varie le pitch, roll et yaw en fonction des orientations du drone.

Essayer d'appuyer sur les FLECHES du clavier, Z, S et ESPACE et observer si les touches sont bien reconnues.

Une fois terminé, appuyez sur Q ou la croix pour quitter. Cela clos la connection avec le drone en même temps.

Etape 5 : Piloter avec le clavier / une manette

A présent on va essayer de piloter le drone avec le clavier.

Dans « DroneMonitor.py », trouve la fonction « send_manual_drone_commands ». Observe son fonctionnement pour savoir comment controller le drone avec le clavier.

Mets à jour la configuration pour piloter le drone avec le clavier :

```
# --- Configuration ---  
KEYBOARD = True  
CONTROLLER = False  
ACTIVATE_CONTROL = True  
MANUAL_CONTROL = True
```

**Attention, pour éviter de casser le drone lors de son utilisation, assure que quelqu'un maintienne le support du drone.
En cas de problème, quitte directement le programme en appuyant sur Q ou en fermant la fenêtre.**

Connectes toi au drone et lance le programme. Envois différentes consignes avec le clavier.

Observe sur l'interface comment évolue les commandes envoyé en fonctions des touches appuyé.

Comment réagis le drone au consignes envoyées ?

Etape 6 : Piloter automatiquement

A présent on va essayer de piloter de manière automatique. C'est à dire que la consigne sera envoyé directement par le programme et non par un humain.

Dans « DroneMonitor.py », trouve la fonction « send_auto_drone_commands ». Observe son fonctionnement pour savoir comment elle marche.

Mets à jour la configuration pour piloter le drone avec le clavier :

```
# --- Configuration ---  
KEYBOARD = True  
CONTROLLER = False  
ACTIVATE_CONTROL = True  
MANUAL_CONTROL = False
```

Attention, pour éviter de casser le drone lors de son utilisation, assure que quelqu'un maintienne le support du drone.

En cas de problème, quitte directement le programme en appuyant sur Q ou en fermant la fenêtre.

Connectes toi au drone et lance le programme.

Observe le drone tout en le maintenant. Observe les commandes envoyées sur l'interface.

Comment réagit le drone aux consignes envoyées ?

Etape 7 : Programmer un système de pilotage

Maintenant que tu as pris en main le drone et tester différentes manières de le contrôler, il est intéressant d'essayer de le programmer soi-même.

Objectif :

- 1 : Faire en sorte que les moteurs s'arrêtent si le pitch > 20° ou le roll > 20°.
2. Mettre en place plusieurs régimes de puissance contrôlés par des entrées clavier (A->25% de la poussée, B->50%, C->75%)
- 3 : Essayer de piloter le yaw. (Faire tourner le drone sur lui-même)

Pour cela, **complète la fonction « my_pilot »** (ligne 144) dans « DroneMonitor.py » en t'inspirant des fonctions « send_manual_drone_commands » et « send_auto_drone_commands » qui sont les fonctions utilisées pour le pilote automatique et par le clavier que tu as essayé dans la partie 5 et 6.

Puis mets la configuration pour utiliser ton pilote :

```
# --- Configuration ---  
KEYBOARD = True  
CONTROLLER = False  
ACTIVATE_CONTROL = True  
MANUAL_CONTROL = False  
MY_PILOT = True
```

Attention, les moteurs peuvent recevoir une puissance entre 0 et 60 000 maximum ! Eviter de dépasser 40 000 sinon le drone risque de s'envoler.

Attention, pour éviter de casser le drone lors de son utilisation, assure que quelqu'un maintienne le support du drone.

En cas de problème, quitte directement le programme en appuyant sur Q ou en fermant la fenêtre.

N'hésite pas à demander de l'aide ou des conseils !

indice :

pour piloter uniquement la puissance utilise :

cf.commander.send_setpoint(roll, pitch, yaw, <la puissance que tu veux mettre entre 0 et 40 000>)