

研究室ツール紹介

Drone編

岩見彰太

武市研究室 M2

2021/05/20

Agenda

1. ドローン関係の法律
 - 航空法, 民法, 条例, 電波法...
2. ドローン業界図
 1. PX4
 2. DJI
3. 研究室で使用しているツール
 1. SITL
 1. Drone Kit
 2. Gazebo
 3. QGround Control
 4. (ulog)
 5. (Flight Review)

航空法

- 人口密集地などの空域においては、あらかじめ航空局に申請を提出し、承認を受けなければならない
- 飛行方法
 - 日の出から日没まで（夜間飛行は別途申請が必要）
 - 催しが行われている上空での飛行禁止（別途申請が必要） etc.
- ※ しかし、200g以下の機体は、規制される無人航空機に含まれない
（2022年度に100gに制限が引き下げられる予定...）

民法

- 土地所有権
 - 民法207条「土地所有権の範囲」に基づき、他人の土地で無許可にドローンを飛行させてしまうと土地所有権の侵害
 - しかし、現状「飛行させた際の不利益」証明できなければ訴えることができない

小型無人気筒飛行禁止法（議員立法）

- 「重要な施設の周辺地域の上空における小型無人機等の飛行の禁止に関する法律」

道路交通法

- 公道上空での飛行は、一般交通に影響を及ぼすものとされるため、一般的には所轄警察署所長の許可が必要

条例

- 各地域で独自に無人航空機に関する条例が定められている
 - ex.) 東京都立公園条例

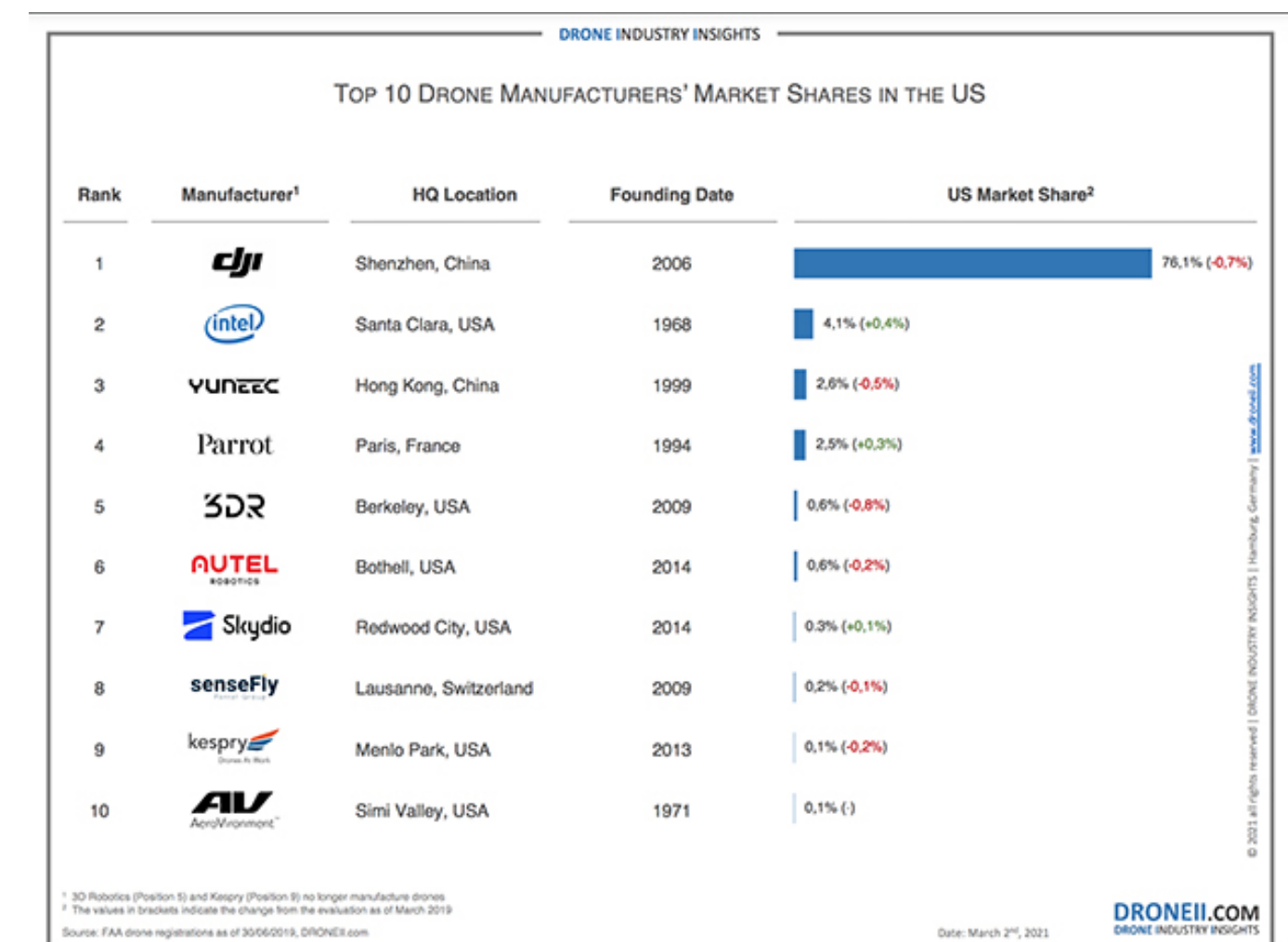
電波法

- セルラードローンの規制
 - 携帯キャリアの通信網を利用して通信を行うドローンのこと
 - 現在は実用化試験局の免許手続きを行った場合に限り、実用化試験可能

ドローン業界図

業界シェア

- 中国の大手メーカー DJI がトップを独走
- しかし外交関係から、アメリカなどではDJIの機体を政府の利用から完全に除外する制裁を発動
- 今後、DJI以外のメーカーからこの辺りの需要を突くような製品の開発が進んでいくと思われる
 - と、ずっと言われているがDJIの技術力が高すぎるためどこも追いつけない
- ENRIでもDJIのドローンの使用は禁止されたりしたので、日本でも政府系の利用は今後厳しいかもしれない
- 民間ではDJI機体は積極的に用いられている



2021年3月現在アメリカのドローン市場

DRONEII、ドローンメーカーマーケットシェア2020 TOP10発表！中米紛争後の米国ドローン市場シェアはどうなった？

ドローン業界図

PX4

- 世界で最も大きなOSS ドローンソフトウェア
- 各社製品で、これをベースに自社開発しているプロダクトも多い（最近だとSONYとか）
- かつて3D Robotics社が管理しており、その際に作られたラッパーライブラリも多い（以前使用した`drone-kit`はこれ）



DJI

- 世界最大手中国ドローンメーカー
- テレビで空撮の映像が流れたら、100% DJI製
- 実は元々ジンバルを作っていた会社
- DJI SDK を利用可能な機体は、自分でカスタマイズできる（後述）





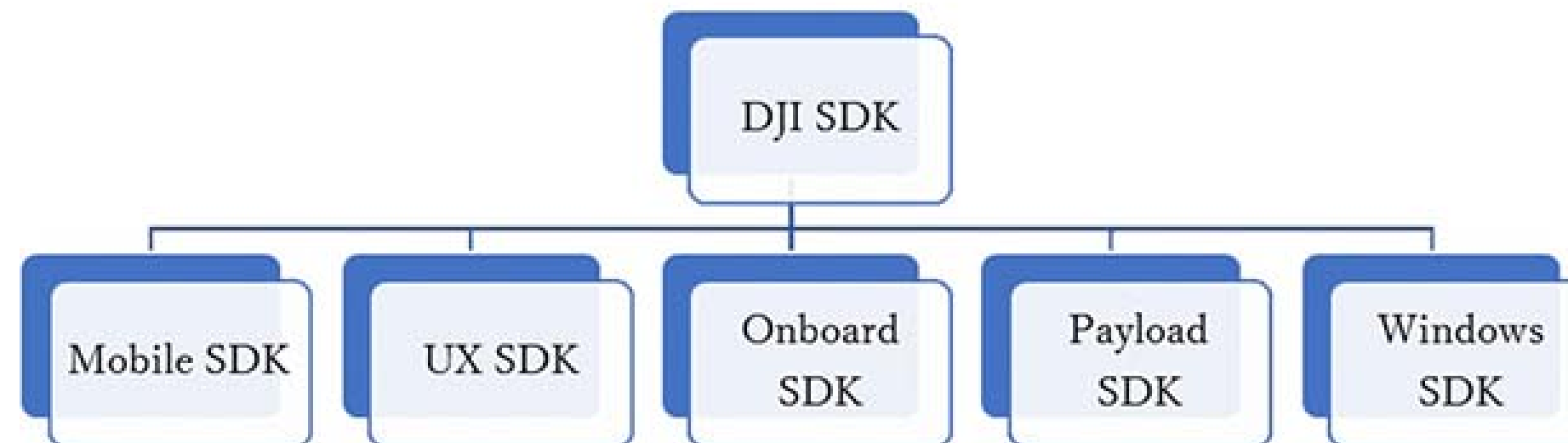
Matrice 600

研究室で買った機体

- いくつかある DJI SDK の内 **Onbord SDK** が使える
 - シリアル接続により、直接フライトコントローラーと通信ができる
 - センサーのデータを自由に取りれる
 - それを利用して飛行軌道の制御ができる
- 高い

そもそも DJI SDK とは

- DJI製の機体を開発するためのキット
- それぞれ用途により分かれている
- 研究において（もし使うとしたら）関係あるのは Mobile SDK と Onbord SDK
 - Mobile SDK
 - RC で操作するAndroid or iOS アプリ開発用のSDK（Swift, Kotlin）
 - サポートされているセンサーデータの取得や飛行指示などが利用できるはず
 - Onbord SDK
 - 機体に相乗りさせて利用する際に使用するSDK
 - Raspberry piなどのマイコンから何か操作する際に利用



PX4

今まで研究室（自分）が利用してきたのはこっち

最終実装目標はこのSITLが回せること



四次元航法/dronekit/px4-gazebo-sim

Distance to waypoint (1): 49.243883279330555
現在時刻: 2021-01-07 20:39:57.286121
開始時刻: 2022-01-07 20:48:15
設定速度: 0.62463138476841373
開始まで残り: 2

Distance to waypoint (1): 49.243883279330555
現在時刻: 2021-01-07 20:39:57.391916
開始時刻: 2022-01-07 20:48:15
設定速度: 0.625619162246594952
開始まで残り: 2

Distance to waypoint (1): 49.243883279330555
現在時刻: 2021-01-07 20:39:57.498591
開始時刻: 2022-01-07 20:48:15
設定速度: 0.626484798717627525
開始まで残り: 2

Distance to waypoint (1): 49.243883279330555
現在時刻: 2021-01-07 20:39:57.605738
開始時刻: 2022-01-07 20:48:15
設定速度: 0.638157631263136864
開始まで残り: 2

Distance to waypoint (1): 49.23190884345997
現在時刻: 2021-01-07 20:39:57.711182
開始時刻: 2022-01-07 20:48:15
設定速度: 0.6314176429543972
開始まで残り: 2

Distance to waypoint (1): 49.23190884345997
現在時刻: 2021-01-07 20:39:57.817448
開始時刻: 2022-01-07 20:48:15
設定速度: 0.63242553779542145
開始まで残り: 2

Distance to waypoint (1): 49.23190884345997
現在時刻: 2021-01-07 20:39:57.924108
開始時刻: 2022-01-07 20:48:15
設定速度: 0.63279193118214687
開始まで残り: 2

Distance to waypoint (1): 49.23190884345997
現在時刻: 2021-01-07 20:39:58.030812
開始時刻: 2022-01-07 20:48:15
設定速度: 0.63446565591273988
開始まで残り: 1

Distance to waypoint (1): 49.22677267087846
現在時刻: 2021-01-07 20:39:58.139142
開始時刻: 2022-01-07 20:48:15
設定速度: 0.63646256534218788
開始まで残り: 1

Distance to waypoint (1): 49.22677267087846
現在時刻: 2021-01-07 20:39:58.245962
開始時刻: 2022-01-07 20:48:15
設定速度: 0.637243131548166275
開始まで残り: 1

Distance to waypoint (1): 49.22677267087846
現在時刻: 2021-01-07 20:39:58.352168
開始時刻: 2022-01-07 20:48:15
設定速度: 0.648173955261797386
開始まで残り: 1

Distance to waypoint (1): 49.22677267087846
現在時刻: 2021-01-07 20:39:58.459294
開始時刻: 2022-01-07 20:48:15
設定速度: 0.644367884686246955
開始まで残り: 1

Distance to waypoint (1): 49.226778964184355
現在時刻: 2021-01-07 20:39:58.566598
開始時刻: 2022-01-07 20:48:15
設定速度: 0.644362417209945145
開始まで残り: 1

Distance to waypoint (1): 49.226778964184355
現在時刻: 2021-01-07 20:39:58.672293
開始時刻: 2022-01-07 20:48:15

```
INFO [tone_alarm] notify negative
INFO [logger] closed logfile, bytes written: 24484773
INFO [tone_alarm] notify negative
INFO [tone_alarm] notify neutral
INFO [commander] Armed by external command
INFO [navigator] RTL: landing at home position.
INFO [logger] [logger] ./log/2021-01-07/11_29_11.ulg
INFO [logger] [logger] ./log/2021-01-07/11_29_11.ulg
INFO [commander] Disarmed by auto preflight disarming
INFO [logger] closed logfile, bytes written: 754696
INFO [tone_alarm] notify neutral
INFO [commander] Armed by external command
INFO [logger] Start file log (type: full)
INFO [logger] [logger] ./log/2021-01-07/11_29_29.ulg
INFO [navigator] Takeoff to 9.9 meters above home.
INFO [logger] [logger] ./log/2021-01-07/11_29_29.ulg
INFO [commander] Takeoff detected
INFO [tone_alarm] notify neutral
INFO [commander] Landing detected
INFO [navigator] Mission finished, landed.
INFO [commander] Disarmed by landing
INFO [tone_alarm] notify neutral
INFO [logger] closed logfile, bytes written: 6966745
INFO [tone_alarm] notify negative
INFO [tone_alarm] notify negative
INFO [tone_alarm] notify neutral
INFO [commander] Armed by external command
INFO [logger] Start file log (type: full)
INFO [logger] [logger] ./log/2021-01-07/11_32_03.ulg
INFO [logger] [logger] ./log/2021-01-07/11_32_03.ulg
INFO [navigator] Takeoff to 10.1 meters above home.
INFO [commander] Takeoff detected
INFO [navigator] RTL HOME activated
INFO [navigator] RTL: landing at home position.
INFO [navigator] RTL: climb to 499 m (10 m above destination)
INFO [navigator] RTL: return at 499 m (10 m above destination)
INFO [navigator] RTL: land at destination
INFO [commander] Landing detected
INFO [commander] Disarmed by landing
INFO [tone_alarm] notify neutral
INFO [logger] closed logfile, bytes written: 3120236
INFO [tone_alarm] notify negative
INFO [tone_alarm] notify neutral
INFO [commander] Armed by external command
INFO [navigator] Takeoff to 9.8 meters above home.
INFO [logger] Start file log (type: full)
INFO [logger] [logger] ./log/2021-01-07/11_38_19.ulg
INFO [logger] [logger] ./log/2021-01-07/11_38_19.ulg
INFO [commander] Takeoff detected
```

後で見る

共有

20:40:00開始まで待機





技術スタック

1. Drone Kit

- Pythonのラッパーライブラリ

2. Gazebo

- 3D シミュレーター（※ Windows では動きません）

3. QGroud Control

- PX4 を制御・監視モニタリングする地上局ソフト

4. ulog

- PX4 などのデバイスで利用される特殊な log ファイル

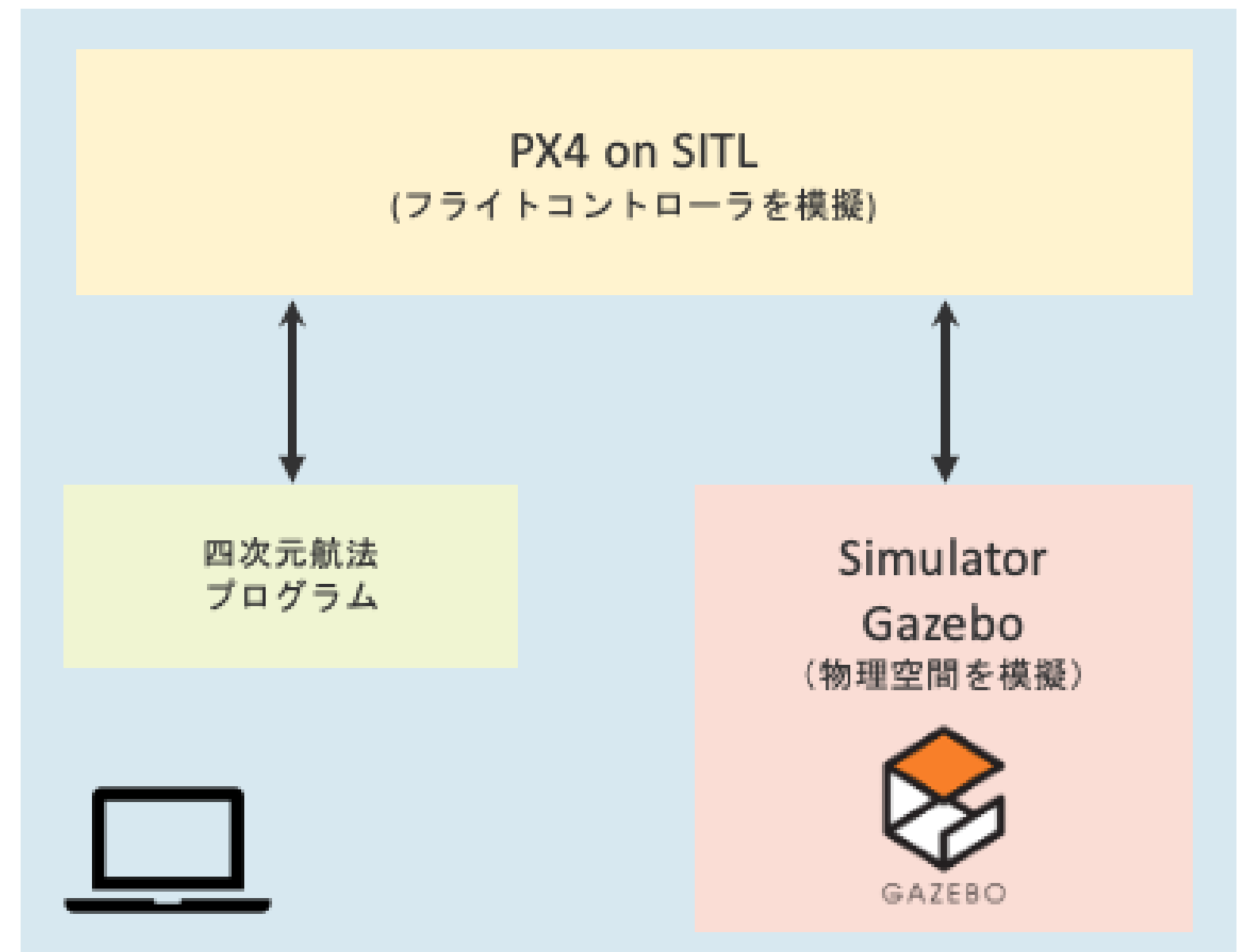
5. Flight Review

- 即座にフライトログを確認できるWEB App

SITLとは？

Software in the Loop

- ハードウェアなしで、機体の飛行をシミュレーションすること。
- 今回の場合は、シミュレーターでPX4を起動して、そこに対してGazeboで仮想環境（風や重力加速度等）を模擬して検証する



環境構築 (Mac OS の場合)

```
$ brew tap PX4/px4
$ brew install px4-dev

# install required packages using pip3
$ python3 -m pip install --user pyserial empy toml numpy pandas jinja2 pyyaml pyros-genmsg packaging
# if this fails with a permissions error, your Python install is in a system path - use this command instead:
$ sudo -H python3 -m pip install --user pyserial empy toml numpy pandas jinja2 pyyaml pyros-genmsg packaging

$ brew install --cask xquartz
$ brew install px4-sim-gazebo`
```

clone submodule が大量にあるので、`--recursive`を忘れない

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

シミュレーター実行

```
$ cd /path/to/PX4-Autopilot
$ make px4_sitl gazebo
```

drone-kit の install

(これで、PYTHONで機体を操作できるようになる)

作業ディレクトリで実行 自分の場合は、pyenvで仮想環境を作った

```
# $ python -m venv [newenvname]
# 名前はvenvにするのが無難
$ python -m venv venv

# activate
$ source [newenvname]/bin/activate

$ vi requirements.txt
# 右の記載をする

$ pip install -r requirements.txt
```

`requirements.txt`

```
flask
dronekit
dronekit-sitl
schedule
pyserial
pyulog
pandas
matplotlib
```


全体像

- あとは、localhost:14540 に対して **drone-kit** で接続すればいい
- 機体の操作は drone-kit のリファレンスを参照
- ここの通信は MavLink という特殊なプロトコルで行われているが、これをラッパーしたライブラリは実は Python 製以外も存在する（ex, Golang 製の Gobot など）

