

#ライブラリのインポート

```
import time
from codrone_edu.drone import *
from gdx import gdx
```

#ドローンのペアリング

```
drone = Drone()
drone.pair()
```

#GoDirect センサーの設定

```
gdx = gdx.gdx()
```

#Bluetooth の場合 : gdx.open(connection='ble')

センサ ID 指定の場合 : gdx.open(connection='usb/ble', device_to_open="GDX-FOR 071000U9 (センサの ID 【センサ本体に記載】))

```
gdx.open(connection='usb')
```

#ワイヤレス力/加速度センサのセンサ番号の指定

【センサ番号】 1:力 2:X 軸加速度 3:Y 軸加速度 4:Z 軸加速度 5:X 軸角速度 6:Y 軸角速度 7:Z 軸角速度

センサ番号を調べる際は GoDirect サンプル gdx_getting_started_device_info.py を実行してください。

```
gdx.select_sensors([1,2,3])
```

#GoDirect のサンプリングレート (ミリ秒)

```
gdx.start(100)
```

```
while True:
```

```
    measurements = gdx.read()
```

```
    if measurements == None:
```

```
        break # データがなければループを抜ける
```

#注意 : センサ番号と、指定したセンサの番号の若い順から [0] から始まる【リスト】に格納される。

#例として gdx.select_sensors([2,3]) とした場合

```
#acc_x = measurements[0]
```

```
#acc_y = measurements[1]
```

```
force = measurements[0]
```

```
acc_x = measurements[1]
```

```
acc_y = measurements[2]
```

gdx.select_sensors([3,2,1])と指定した場合でもリストの番号は変わらない。若いセンサ番号から順にリストに格納される。

```
force = measurements[2]
acc_x = measurements[1]
acc_y = measurements[0]
```



```
force = measurements[0]
acc_x = measurements[1]
acc_y = measurements[2]
```



#カセンサを押す動作

```
if force < -20: #強く押したら離陸
    drone.takeoff()
```

```
elif force < -5: #軽く押したら上昇
    drone.set_throttle(30)
    drone.move()
```

#カセンサを引く動作

```
elif force > 20: #強く引っ張ると着陸
    drone.land()
```

```
elif force > 5: #軽く引っ張ると下降
    drone.set_throttle(-30)
    drone.move()
```

順番を逆にしてしまうと数値の低い方の条件(上昇/下降)が優先されてしまうため、離陸/着陸ができなくなるので注意。

#X 軸加速度の動作

```
elif acc_x > 5: # X 軸方向右に傾けると右進
    drone.set_roll(30)
    drone.move()
```

```
elif acc_x < -5: # X 軸方向左に傾けると左進
    drone.set_roll(-30)
    drone.move()
```

#Y 軸加速度の動作

```
elif acc_y > 5: # Y 軸方向前に傾けると前進
    drone.set_pitch(30)
    drone.move()
```

```
elif acc_y < -5: # Y 軸方向後ろに傾けると後進
    drone.set_pitch(-30)
    drone.move()
```

#上記以外の動作(センサロゴ面が上を向いている間はホバリング)

```
else:
    drone.hover()
```