

In [ ]:

```
# 必要なモジュールのインポート
import socket
import threading
import time
import sys

# telloのIPアドレスとポート
tello_address = ('192.168.10.1', 8889)

# ローカルコンピュータのIPアドレスとポート
local_address = ('', 9000)

# コマンドを送るためのUDP接続のの实体化
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# ローカルアドレスとポートのバインディング
sock.bind(local_address)

# telloへのコマンドの送信とディレイタイム（秒）生成関数
def send(message, delay):
    # 例外が表示されなかった場合, telloにエンコードされたコマンドを送信する
    try:
        sock.sendto(message.encode(), tello_address)
        print("Sending message: " + message)
    except Exception as e:
        print("Error sending: " + str(e))

    # ユーザーが定義した移動待ち時間（秒）
    time.sleep(delay)

# telloからのメッセージ受信関数
def receive():
    # telloからのメッセージ受信を常態化
    while True:
        # 例外が表示されなかった場合, 受信状態を保つ
        try:
            response, ip_address = sock.recvfrom(128)
            print("Received message: " + response.decode(encoding='utf-8'))
        except Exception as e:
            # もしエラー検出された場合, ソケットを閉じ, ループを終了する
            sock.close()
            print("Error receiving: " + str(e))
            break

# シークエンス関数
def sequence():
    # ここにシークエンスを書き込む. 引数は2つで第一引数は"コマンド 移動距離 (cm)", 第二引数はディレイ
    #####
    send("up 200", 2)
    send("down 150", 2)
    send("right 20", 2)
    send("down 20", 2)
    send("right 20", 2)
```

```

send("down 20", 2)
send("right 20", 2)
send("up 20", 2)
send("right 20", 2)
send("up 20", 2)
send("right 20", 2)
#####
# ここまでは追記・削除ok. あとはいじらないこと!

# ここからがメインの実行処理
# バックグラウンド実行されるスレッドを作り、スタートする
# これはreceive関数を利用して、telloから送信されたメッセージを常態的に受信する
receiveThread = threading.Thread(target=receive)
receiveThread.daemon = True
receiveThread.start()

# telloをコマンドモードにセット
send("command", 3)
# 離陸コマンドの送信
send("takeoff", 5)

# ユーザーのコマンド入力のヘルプを表示
print('Type in a Tello SDK command and press the enter key. Enter "start" to start sequence. Enter "quit" to quit the program.')

# ユーザーが"quit"もしくは"start"をタイプするか、ctrl+cを押すまで、コマンド入力待ちをループ実行する
while True:

    try:
        # ユーザーのキーボード入力を読み込み
        message = input("")
        # もしユーザーが"quit"を入力したら、ソケットを閉じアプリケーション終了
        if 'quit' in message:
            # Land
            send("land", 5)
            print("Program exited sucessfully")
            sock.close()
            break
        # もしユーザーが"start"を入力したら、シークエンス開始
        elif 'start' in message:
            sequence()
            message = ""

        # telloにコマンドを送信
        send(message, 3)

    # ctrl+cの入力があった場合の制御
    except KeyboardInterrupt as e:
        send("land", 5)
        sock.close()
        break

```

In [ ]:

In []:

In []:

In []:

In []: