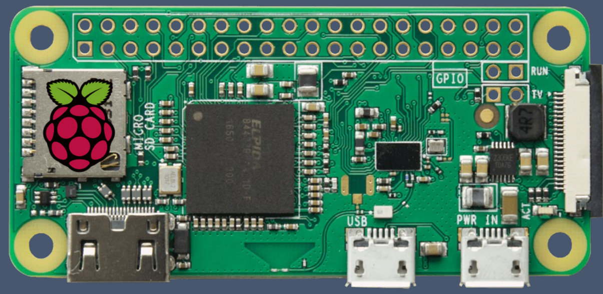
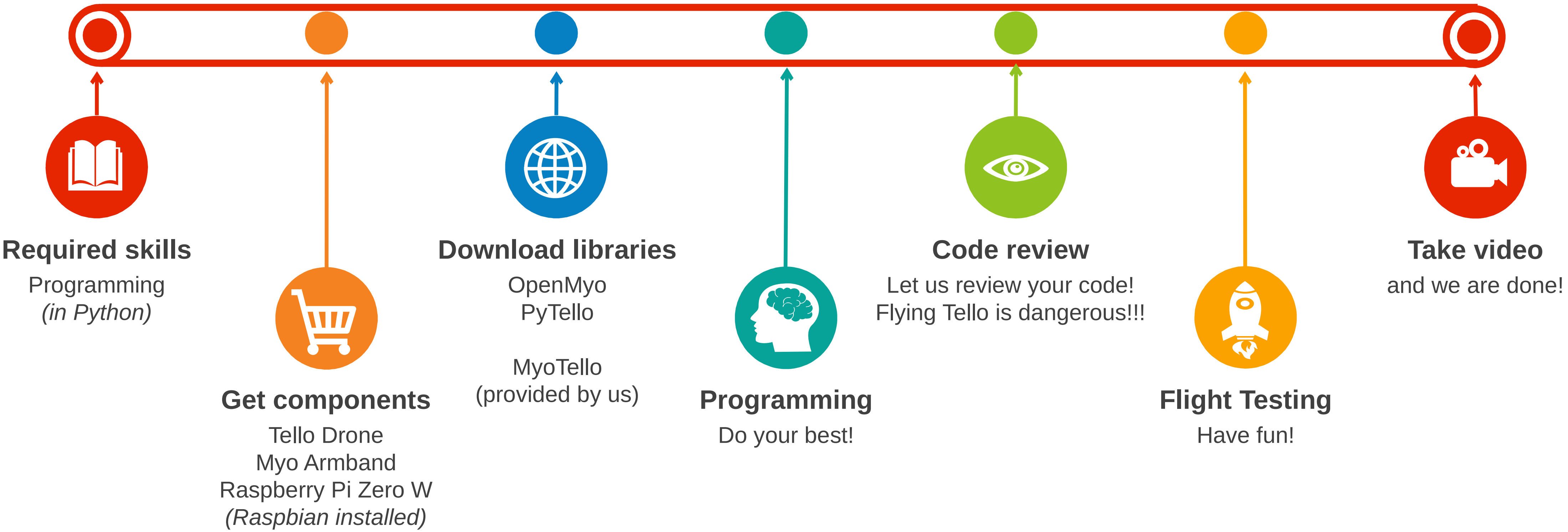


Control Tello drone with Myo armband and Raspberry Pi

Hoang Chu

Neuro-Robotics lab, Tohoku University

Let’s do it!



Required skills
Programming
(in Python)

Get components
Tello Drone
Myo Armband
Raspberry Pi Zero W
(Raspbian installed)

Download libraries
OpenMyo
PyTello

MyoTello
(provided by us)

Programming
Do your best!

Code review
Let us review your code!
Flying Tello is dangerous!!!

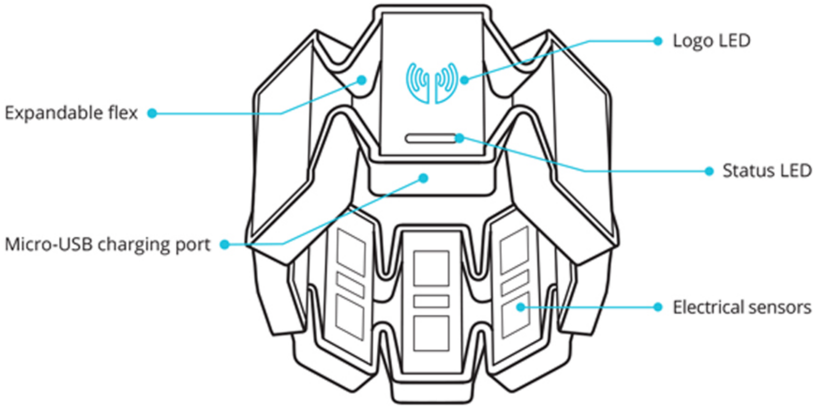
Flight Testing
Have fun!

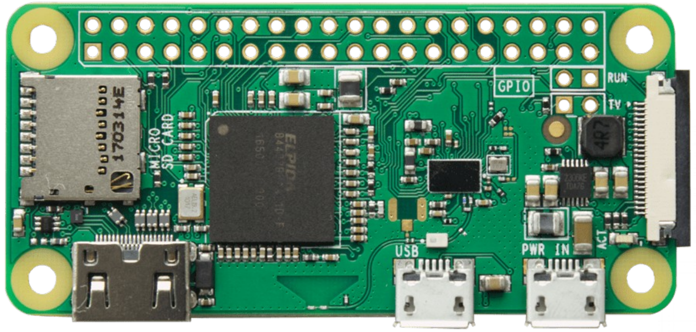
Take video
and we are done!

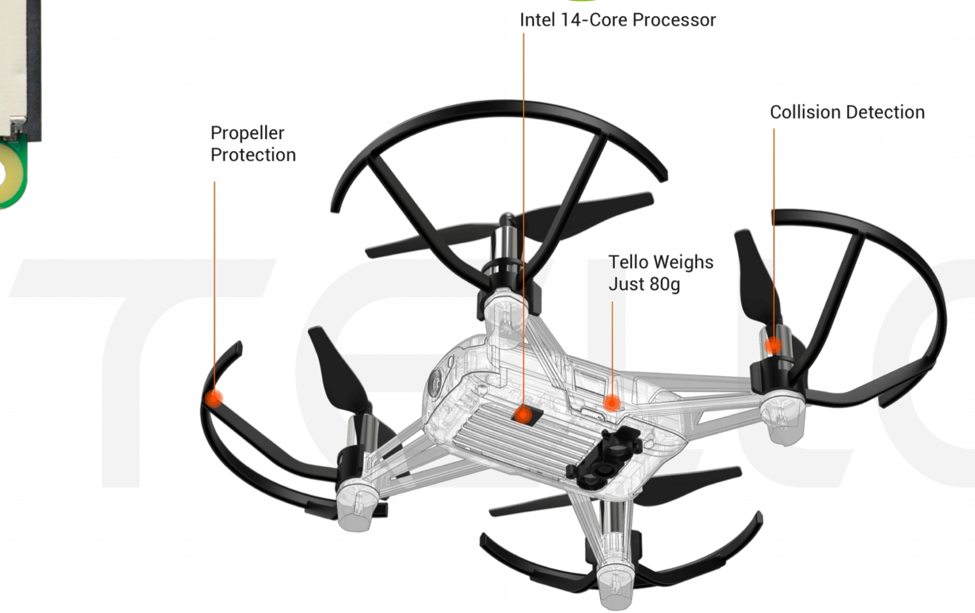
How stuffs work

Raspberry Pi

Obtains signals from Myo device and geneate control signal, then transmit to Tello Drone wirelessly







MyoTello = OpenMyo + PyTello

MyoTello.takeoff()
Simple takeoff
MUST take off from horizontal plane

MyoTello.land()
Simple landing
Tello can land on a flat surface or on hand

MyoTello.noMove()
Very important!
Has to be called to make sure the drone is not out of control. Tello keeps moving according to the LAST command sent to it

MyoTello.moveBackward()

MyoTello.moveForward()

MyoTello.moveLeft()

MyoTello.moveRight()

Class MyoTello

from myotello import MyoTello

MyoTello.moveUp()

MyoTello.moveDown()

MyoTello.turnLeft()

MyoTello.turnRight()

MyoTello.stop()
Has to be called to stop the connections between Tello drone and Raspberry Pi
Python program will keep running if this function is not called

Your creative ideas are welcomed!
Please think about new functions to control the drone smoothly
Investigate myotello.py if you want to add your functions

Initialize drone

```
drone=MyoTello(process_imu,process_sync,process_classifier,mac_address)
```

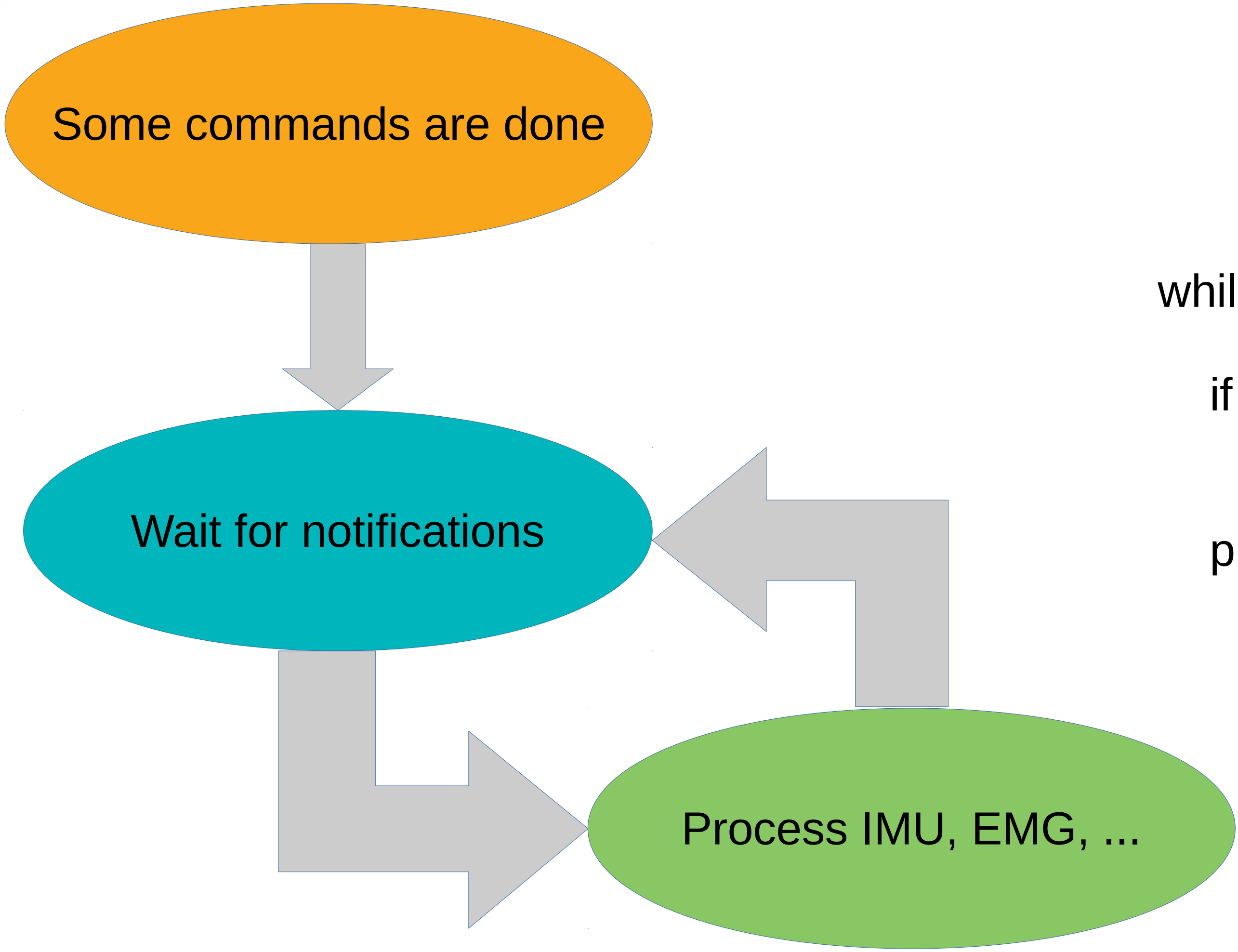
mac_address of Myo device is optional. If mac_address is not specified,
the program will automatically search for a nearby Myo and connect to it

```
def process_imu(quaternion, acceleration, gyroscope):
        #Quaternion can tell orientation of the myo
        #Enter your code
def process_sync(arm, x_position):
        #Enter your code
def process_classifier(pose):
        #Pose can be fist, double tap, rest, finger spread,
        # wave in and wave out
        #Enter your code
```

```
class Pose(Enum):
    REST = 0x00
    FIST = 0x01
    WAVE_IN = 0x02
    WAVE_OUT = 0x03
    FINGERS_SPREAD = 0x04
    DOUBLE_TAP = 0x05
    UNKNOWN = 0xff
```

These functions are handlers for Myo “events”
They are called when data is transmitted to Raspberry Pi.
You need to combine functions in the previous slide to control the drone.

Basic Myo Loop



Some commands are done

Wait for notifications

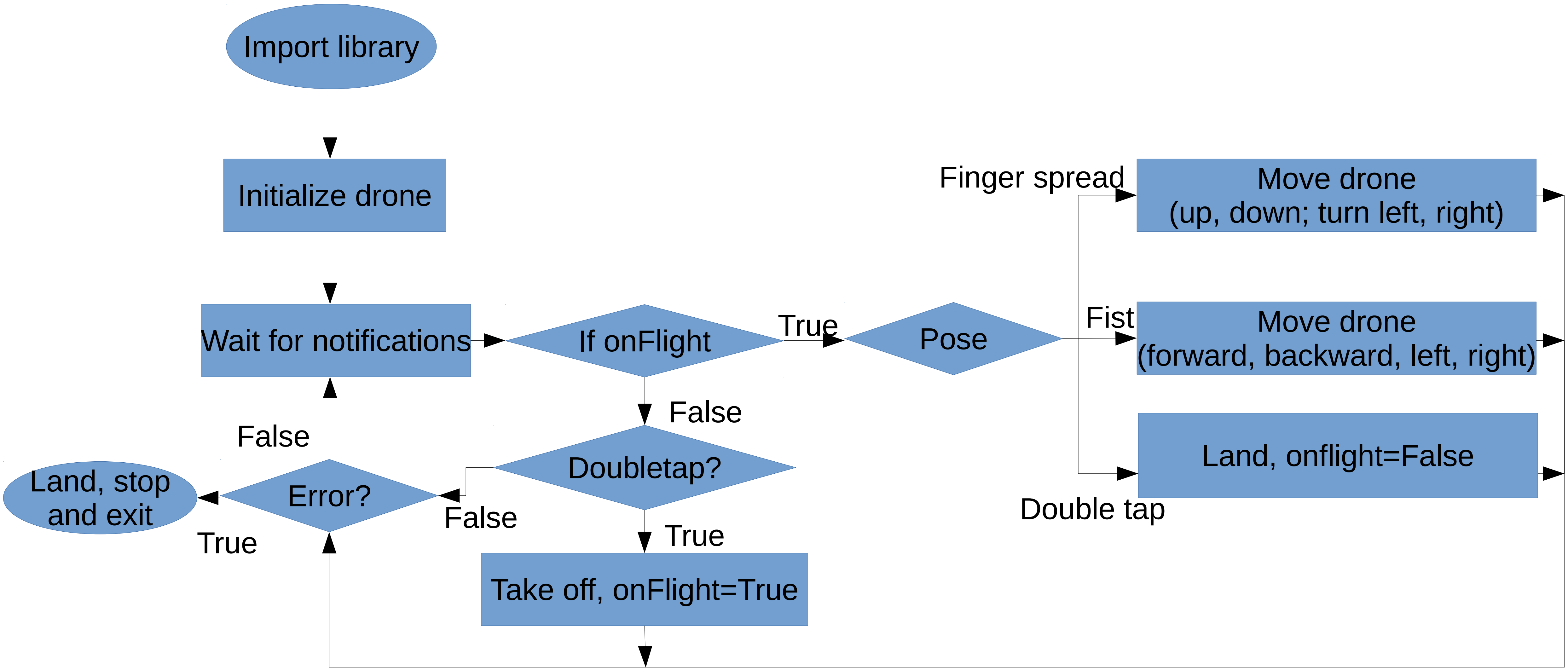
Process IMU, EMG, ...

```
while True:

    if drone.myo_device.services.waitForNotifications(1):
        continue

    print("Waiting...")
```

Sample diagram



Sample program

```
from myotello import MyoTello

def process_imu(quat,acc,gyro):
       print(quat)

def process_sync(arm, x_direction):
    print(arm, x_direction)

def process_classifier(pose):
    print(pose)

drone=MyoTello(process_imu,process_sync,process_classifier)

while True:
    try:
        if drone.myo_device.services.waitForNotifications(1):
            continue
        print("Waiting…")
    except:
        drone.stop()
```

Thank you for listening!