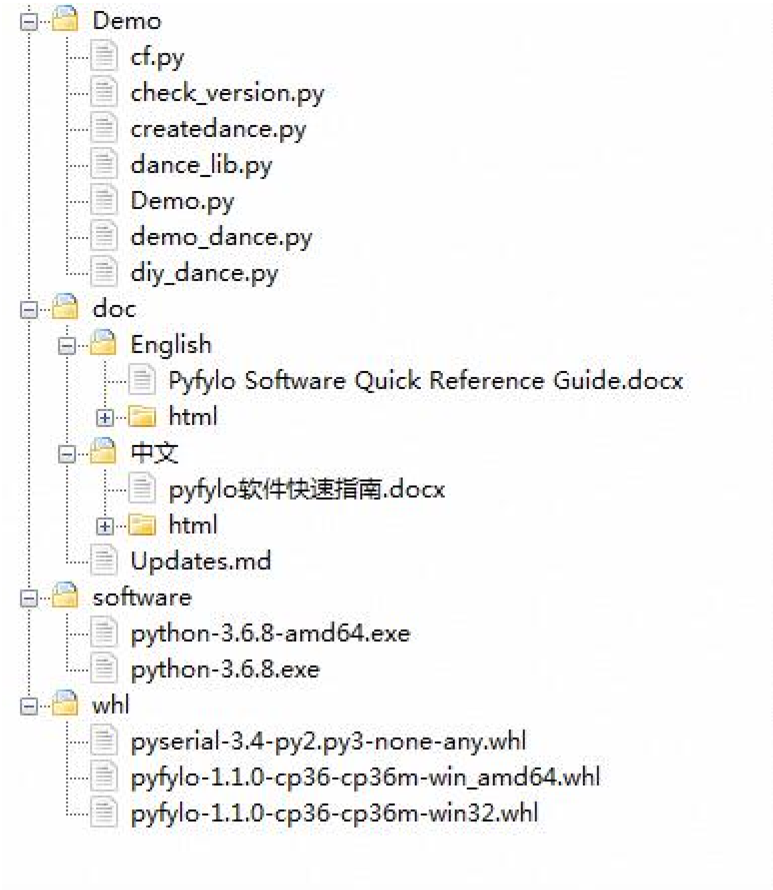
Pyfylo Software Quick Reference Guide

Version No.	Date	Description	Principal
1.0.0	2020-03-05	First time submit	Fathi
1.0.1	2020-04-03	Need to use cmd with administrator privileges and implement activation	Fathi
1.0.2	2022-01-24	Added FYLO EDU+ operation instructions	Fathi,Faidy

1 OVERVIEW

Pyfylo software is used to connect and control drones in real-time, also used to make drone formation by programs. This document provides the installation of pyfylo and a quick guide to use this software.

This software kit contains 4 folders: Demo, doc, software and whl. The structure is illustrated below:



```
Demo
    cf.py
    check_version.py
    createdance.py
    dance_lib.py
    Demo.py
    demo_dance.py
    diy_dance.py
doc
    English
        Pyfylo Software Quick Reference Guide.docx
        html
    中文
        pyfylo软件快速指南.docx
        html
    Updates.md
software
    python-3.6.8-amd64.exe
    python-3.6.8.exe
whl
    pyserial-3.4-py2.py3-none-any.whl
    pyfylo-1.1.0-cp36-cp36m-win_amd64.whl
    pyfylo-1.1.0-cp36-cp36m-win32.whl
```

There is a file named index.html in html folder, which can be opened in internet browser. It contains “userapi module”, update statement and so on. All the functions of userapi module are displayed there with detail comments.

Warnning: The critical statements in 6th section should be keep in mind everytime controlling drones!

2 INSTALLING SOFWARES

1. Supported Operating Systems

- windows 7 (x86/x86_64)
- windows 10 (x86/x86_64)

2. Python installer

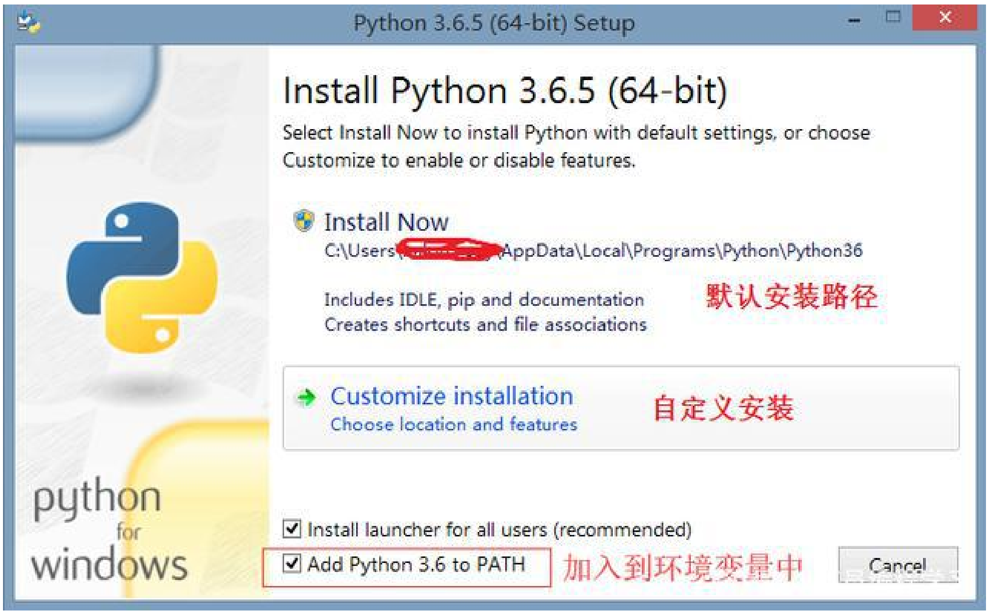
Python installers are in the software folder. The one named with “amd64” is applied for x86_64 system, the other is for x86 system.

In the official python website: https://www.python.org/downloads/windows/ , same installers are supplied for downloading.

Warning: only python 3.6 is supported so far.

3. Installing Python and Configure

A. Double click the python installer. A dialogue window as below figure will pop up. Select “Add Python 3.6 to PATH”, and then choose “Customize installation”.



Python 3.6.5 (64-bit) Setup

Install Python 3.6.5 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\[redacted]AppData\Local\Programs\Python\Python36

Includes IDLE, pip and documentation
Creates shortcuts and file associations

默认安装路径

Customize installation
Choose location and features

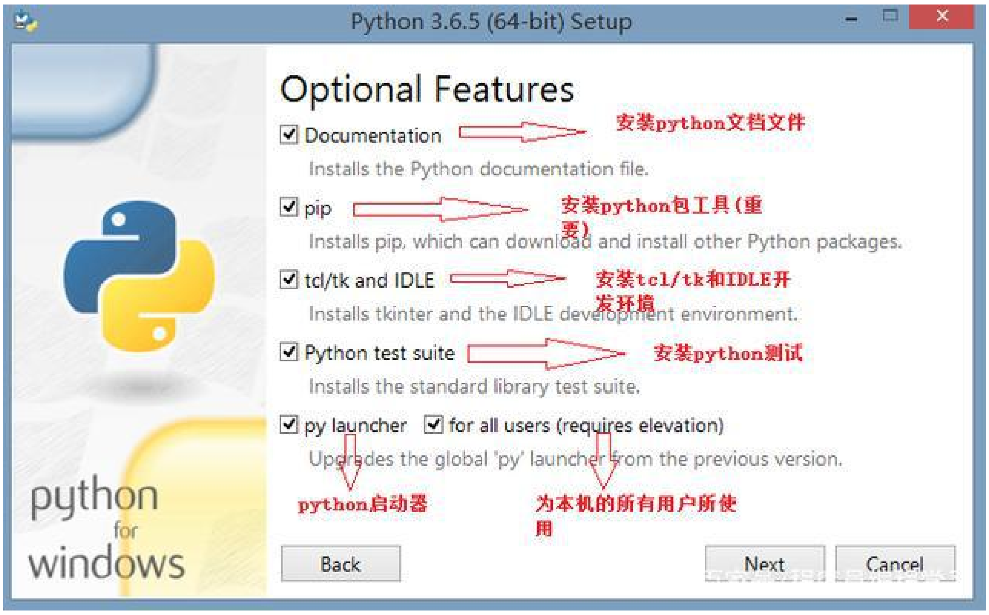
自定义安装

☑ Install launcher for all users (recommended)

☑ Add Python 3.6 to PATH

加入到环境变量中

B. The next dialogue is as the below figure. Select all the opetions and click the “Next” button.



Optional Features

☑ Documentation 安装python文档文件

Installs the Python documentation file.

☑ pip 安装python包工具(重要)

Installs pip, which can download and install other Python packages.

☑ tcl/tk and IDLE 安装tcl/tk和IDLE开发环境

Installs tkinter and the IDLE development environment.

☑ Python test suite 安装python测试

Installs the standard library test suite.

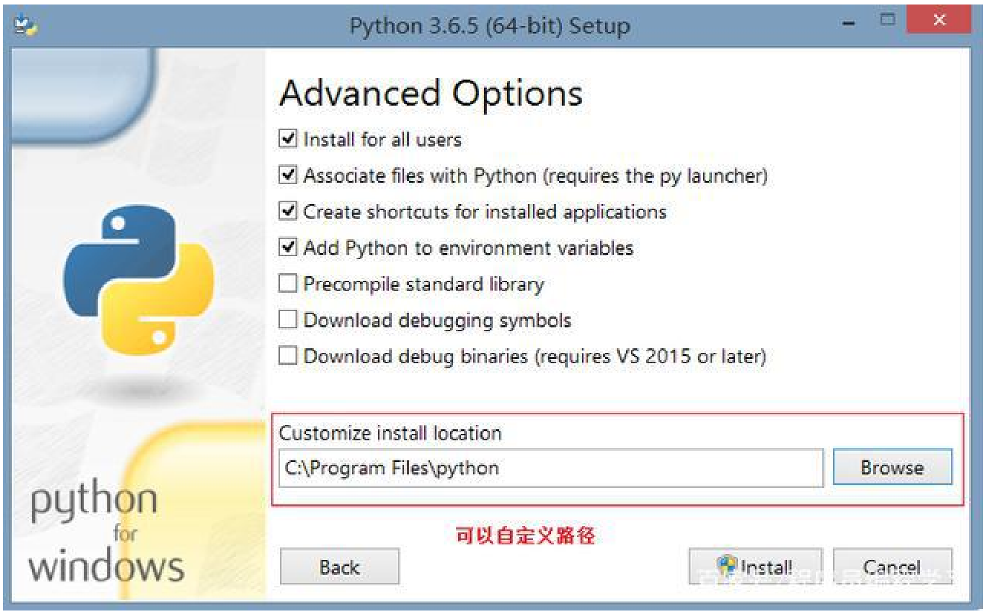
☑ py launcher ☑ for all users (requires elevation)

Upgrades the global 'py' launcher from the previous version.

python启动器

为本机的所有用户所使用

C. Select the 4 options as the below picture, and "Browse" to the target path. Than click the "Install" button.



Advanced Options

☑ Install for all users

☑ Associate files with Python (requires the py launcher)

☑ Create shortcuts for installed applications

☑ Add Python to environment variables

☐ Precompile standard library

☐ Download debugging symbols

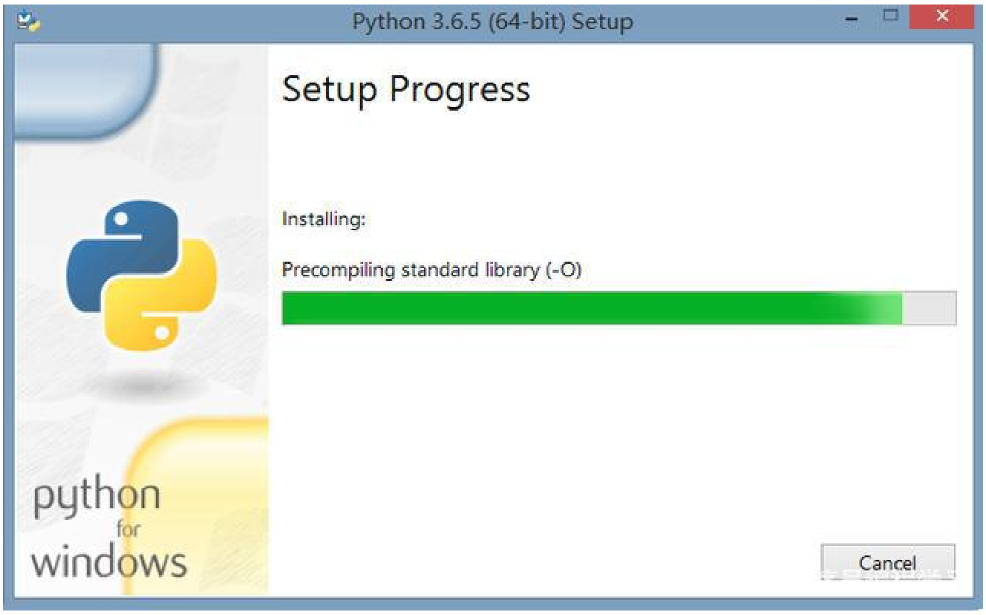
☐ Download debug binaries (requires VS 2015 or later)

Customize install location

C:\Program Files\python

可以自定义路径

D. This picture shows status of installing.



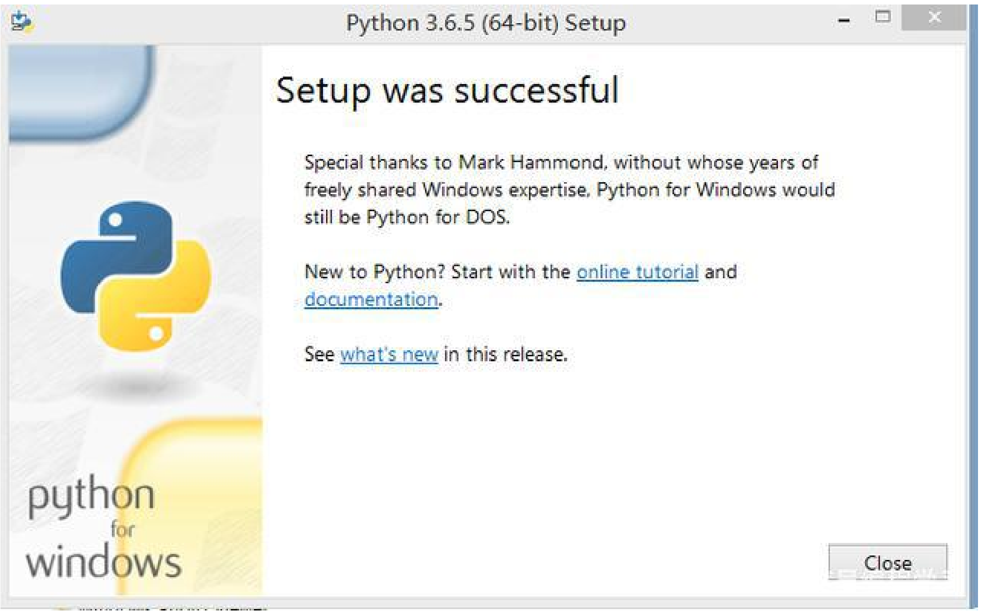
Python 3.6.5 (64-bit) Setup

Setup Progress

Installing:

Precompiling standard library (-O)

E. The below picture means setup is finished.



Python 3.6.5 (64-bit) Setup

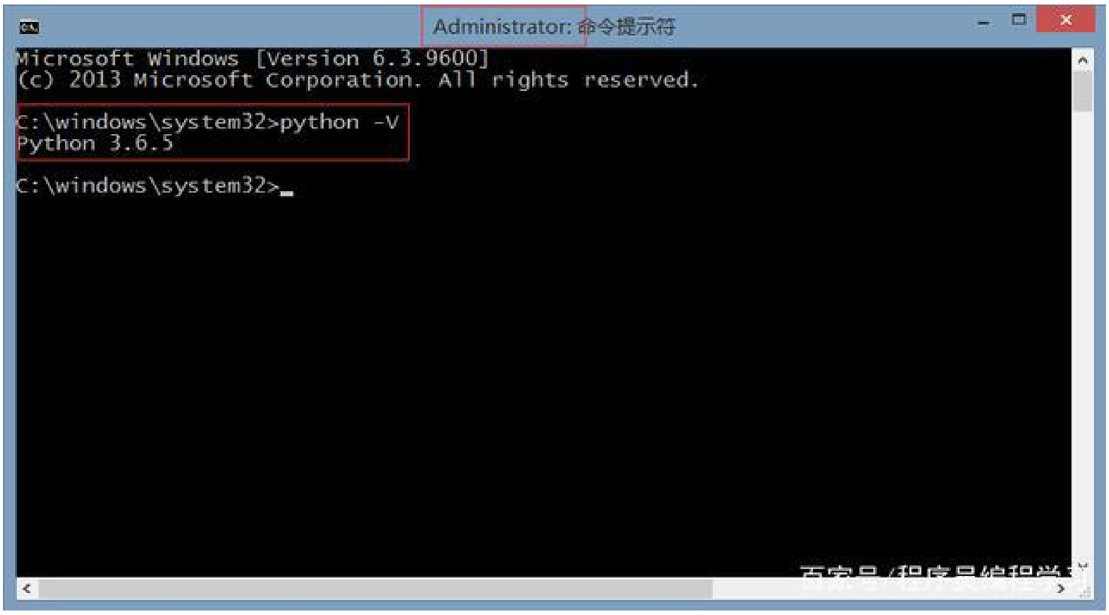
Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

F. Check python version. Click on Start, and input “cmd” in the search bar. Then right click the cmd.exe, and click on “Run as Administrator”. After the cmd window poped，type in **python –V** as shown below, and enter. Python version will be printed on the screen. If the version is same as the installation’s, start to install pyfylo.



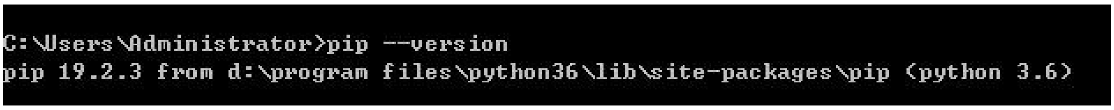
```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\windows\system32>python -V
Python 3.6.5

C:\windows\system32>_
```

4. Installing pyfylo

A. Check python’s “pip” command. Open cmd as upper description and type in **pip --version**. If the returned result is like the below picture, go to the next step.



```
C:\Users\Administrator>pip --version
pip 19.2.3 from d:\program files\python36\lib\site-packages\pip (python 3.6)
```

B. Install whl.

Open the cmd and type in:

pip install pyfylo

If you can’t connect to the Internet, open the cmd, direct to the path of whl, and type in command to install pyfylo and pyserial.

For x86 system:

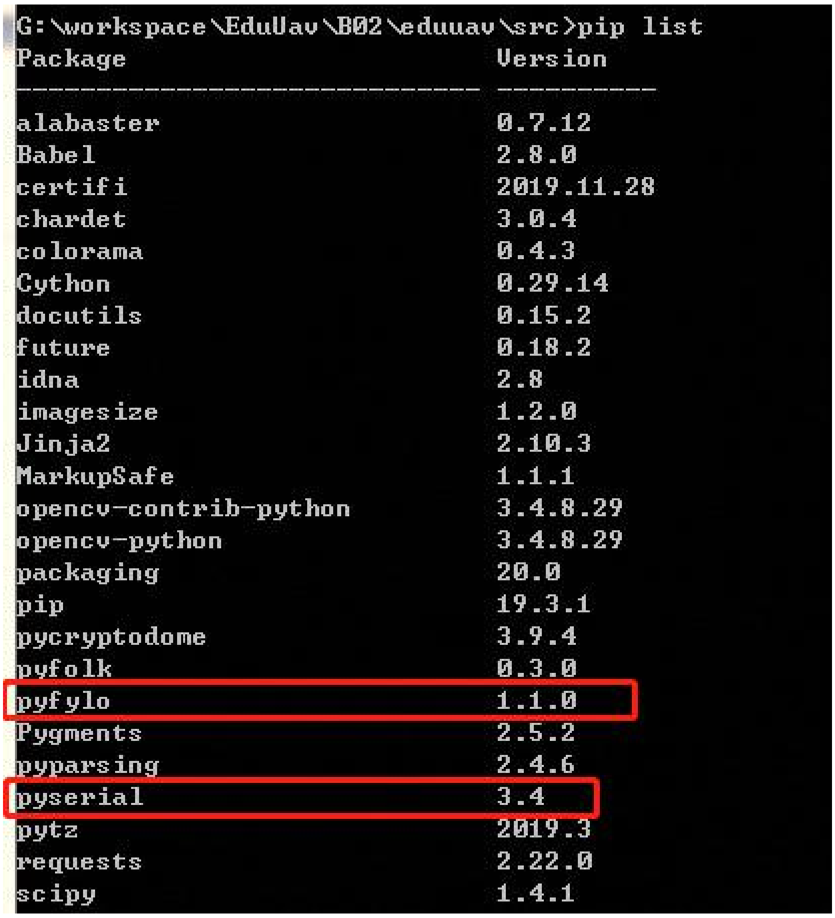
pip install pyfylo-*-win32.whl pyserial-3.4-py2.py3-none-any.whl

For x86_64 system:

pip install pyfylo-*-win_amd64.whl pyserial-3.4-py2.py3-none-any.whl

Note: the “*” shoulde be replaced to make sure the string is same as the name of the pyfyo file.

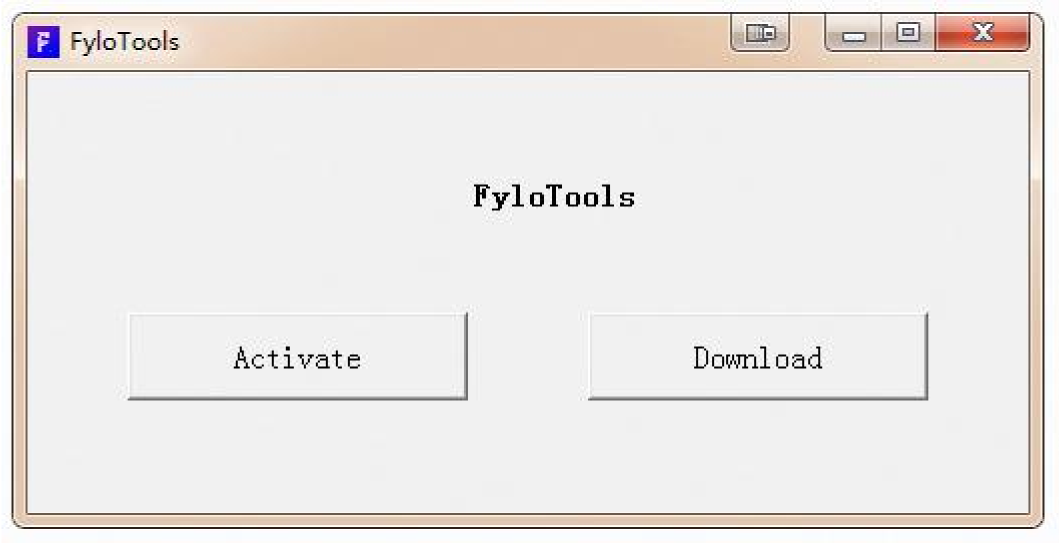
C. Check the whl’s installation. In the cmd, type in **pip list** and enter. If there are two strings like the below picture, the whl installation is finished.



```
G:\workspace\EduUav\B02\eduuav\src>pip list
Package                       Version
----------------------------- ----------
alabaster                     0.7.12
Babel                         2.8.0
certifi                       2019.11.28
chardet                       3.0.4
colorama                      0.4.3
Cython                        0.29.14
docutils                      0.15.2
future                        0.18.2
idna                          2.8
imagesize                     1.2.0
Jinja2                        2.10.3
MarkupSafe                    1.1.1
opencv-contrib-python         3.4.8.29
opencv-python                 3.4.8.29
packaging                     20.0
pip                           19.3.1
pycryptodome                  3.9.4
pyfolk                        0.3.0
pyfylo                        1.1.0
Pygments                      2.5.2
pyparsing                     2.4.6
pyserial                      3.4
pytz                          2019.3
requests                      2.22.0
scipy                         1.4.1
```

5.Activating pyfylo

Find FyloTools.exe and right click on it. In the poped list, click on “Run as Administrator”. A window will pop up as shown below. Then click on ‘Activate’ to activate pyfylo.



3 Updating

1. updating pyfylo

Open the cmd and type in **pip install --upgrade pyfylo**.

2. updating documents

Open FyloTools.exe, then click ‘Download’ to download the latest documentation.

4 Demo Folder

1. File Descriptions

Demo.py entry point of controlling drones in real-time or in program

createdance.py example of creating dance programs

demo_dance.py example of auto generating random dance programs for 8 drones

dance_lib.py dependency of **demo_dance.py**

cf.py dependency of **demo_dance.py**

diy_dance.py primary interfaces for diy dance programs

check_version.py intrfaces for new version detection of drone firmware

2. Entry Point Demo.py

This file binds all the commands of real-time controlling to the keys on the keyboard. Especially, the key “5” is to change the controlling from in realtime to in program that designed in **createdance.py**.

Before using this file, it is recommended that reading the python codes in file and learning the achievement of every command. Here is some primary settings

Key	Default Parameters	Descriptions
W	50	Fly forward 50cm in realtime
S	50	Fly back 50cm in realtime
A	50	Fly left 50cm in realtime
D	50	Fly right 50cm in realtime
Q	90	Fly turn left 90 degrees in realtime
E	90	Fly turn right 90 degrees in realtime
K	50	Fly up 50cm in realtime
L	50	Fly down 50cm in realtime
I		Led on
O		Led off
J	100	Fly bounce 100cm height in realtime
Z	(50, 50, 200)	Fly straight to (50,50,200) in realtime
X		drone arm
C		drone disarm
P	0,1	Set No. 0 drone's id as No. 1
V	2,10	Set No. 2 drone's low power threshold as 10%
B		Update firmware
N	1	Set connection mode to station as repeater mode
M		Disconnect from station
[	0	No. 0 drone enters compass calibrating
]	0	No. 0 drone exits from compass calibrating
1	0, 100, 1	In realime No. 0 drone takes off and flies to 100cm height, with compass on
2	0	In realtime No. 0 drone touches down

3	drone_id_list, 1, 1000	1000ms later, drones take off and start dance programs with compass on
4	drone_id_list	drones stop dance programs and touch down
5		Generate dance
6		drones are prepared for takeoff
7		Dance time sync
8		Positioning inits
9	drone_id_list	Send programs to drones in list
0		stations demarcates
-	drone_id_list, yaw	drones auto sync
=		Show dance in screen
,		Show demarcation
.		Show drones’ status
’		Show status of drones’ sensors
/		Print help information
CTRL+C		Exit pyfylo

3. Createdance.py

In this example file, create_dance is the entry point for creating dance programs. And the parameter ”flag” changes with 4 kinds of dance programs:

- flag = 0, single drone’s simple dance programs using userapi module, shown by No. 0 drone,
- flag = 1, 8 drones’ formation dance programs created using userapi module,
- flag = 2, semi-automatical implementation of every step of single drone’s dance programs, using diy_dance.py additionally, shown by No. 0 drone.
- flag = 3, automatical ramdom implementation of formation dance programs for 8

drones, using dance_lib.py and cf.py.

All these codes are primary demonstration of our production. Users can implement more various of dance programs by more programming technologies.

4. check_version.py

Running this file to check the pyfylo and the drone’s firmware version.

Please configure the following before running:

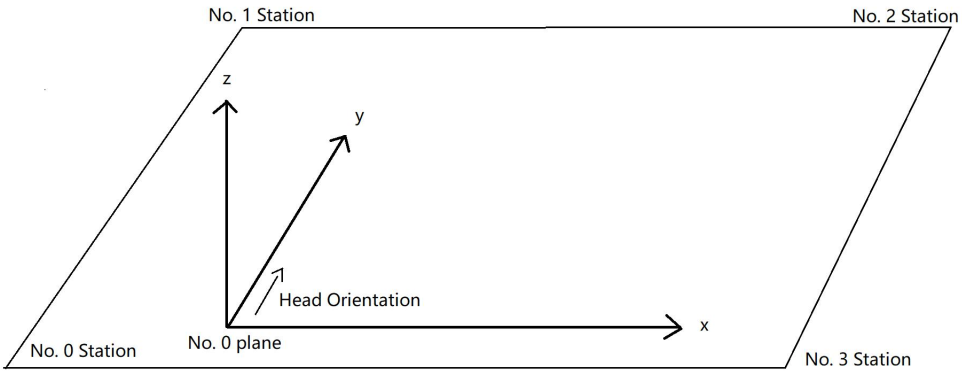
1. Configure download, download_path,pland_id_list according to you envionment

- download, when download is True, this program will download pyfylo if it need to update. When download is False, this program will not download pyfylo in any time.
- download_path, download path for pyfylo.
- drone_id_list, pland id you want to check.

2. Make sure your computer can access to the Internet.
3. Open the station and the drone you want to check.
4. please using repeater to connect, because the Internet is needed in the upgrade process

5 Quick Start

Firstly, place the 4 stations and No. 0 drone as the figure below.



Controlling in realtime

1. Stand at the No. 0 drone’s place, open the compass app in cellphone, and make sure the cellphone have the same orientation as the drone. Then copy the yaw angle in the app to *parameter yaw* in demo.py and save the file.
2. Place stations and drones as the upper figure. Turn their power on. And connect computer to the system. Open Cmd and direct to the path of demo.py. Type in **python demo.py.**
3. Stations demarcates (Key “**0**”).
4. Check the status of the demarcation (Key “**,**”). The coordinates of the 4 stations shoulded be corrected like 4 vertex of a rectangle and sequence as the upper figure.
5. Positioning system inits(Key “**8**”).
6. Check the drone’s status(Key “**.**”). Make sure that the returned results include “'aux setup: Yes” and “rangesafe: Yes”.
7. The drone takes off(Key “**1**”).
8. Move in realtime(Keys “**W**”, “**A**”, “**D**”, “**S**”, “**Q**”, “**E**”, “**J**”, “**K**”, “**L**”, “**Z**”).
9. Touch down(Key “**2**”).

Dance Programs

1. Stand at the No. 0 drone’s place, open the compass app in cellphone, and make sure the cellphone have the same orientation as the drone. Then copy the yaw angle in the app to *parameter yaw* in demo.py and save the file.
2. Place stations and drones as the upper figure. Turn their power on. And connect computer to the system. Open Cmd and direct to the path of demo.py. Type in **python demo.py.**
3. Stations demarcates (Key “**0**”).
4. Check the status of the demarcation (Key “**,**”). The coordinates of the 4 stations shoulded be corrected like 4 vertex of a rectangle and sequence as

the upper figure.

5. Generate dance programs (Key “**5**”).
6. Show dance steps on screen (Key “**=**”).
7. Upload dance programs (Key “**9**”).
8. Check the drone’s status(Key “**.**”). Make sure that the returned results include “dance: Yes”.
9. Dance time sync(Key “**7**”).
10. Check the drone’s status(Key “**.**”). Make sure that the returned results “timesync: Yes”.
11. Positioning system inits(Key “**8**”).
12. Check the drone’s status(Key “**.**”). Make sure that the returned results include “aux setup: Yes” and “rangesafe: Yes”.
13. drones are prepared for takeoff(Key “**6**”).
14. Check the drone’s status(Key “**.**”). Make sure that the returned results include “takeoff allow: Yes”.
15. Multiple drones’ dance show begin(Key “**3**”)
16. Multiple drones stop dance programs and touch down(Key “**4**”). Anytime can the drones touch down if needed.

Note:

Auto configure function(key “-”) is able to implement time synchronization and positioning initialization. After calling it, check the drone’s status and make sure that the returned results include “aux setup: Yes” and “rangesafe: Yes”.

Global Variable Declaration

drone_id_list: list of drones

connect_type: connection mode: 0 for wifi, 1 for repeater and 2 for disconnecting

yaw: drone’s yaw angle

6 Warnings

1. **Everytime before dance show, Check the drone's status, and make sure that the returned results include "aux setup: Yes", "rangesafe: Yes", "timesync: Yes" and "takeoff allow: Yes".**
2. **Before performance, configure the parameters of functions to be used.**
3. **No. 0 drone should exist at its position as the upper figure for any drone's initialization.**
4. **Everytime before takeoff, make sure that drones stay at their initial positions, and finish positioning system initialization and time synchronization.**
5. **Before drones' takoff, if stations are moved after stations demarcation, redo demarcation; Before drones' takeoff, if drones are moved after positioning system initialization, redo positioning system initialization.**
6. **If using compass, firstly calibrate it and set its relative parameters in functions to 1.**
7. **Everytime before takeoff, make sure that at the drone's location, the drone's yaw angle is same as the reading of compass app in cellphone t .**
8. **While drones are in the air, be cautious to use the command "drone_fly_disarm" or key "C".**

7 Config Drone dance player's language

Open Lib\site-packages\pyfylo\DanceViewSoftware\config.ini in python install directory. Modifiy the value of language to config Drone dance player's language.

Chinese: Language=0

English: Language=1

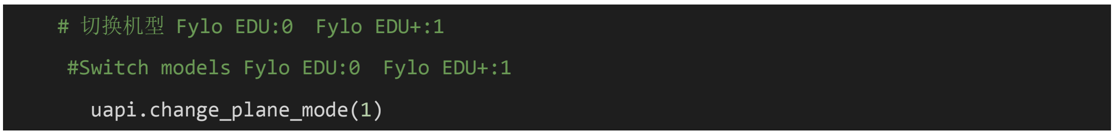
Traditional Chinese: Language=2

8 Fylo EDU+ Instructions for use

Remarks: The following functions are different from the original Fylo EDU, and other operations are the same as the Fylo EDU

1、Switch model

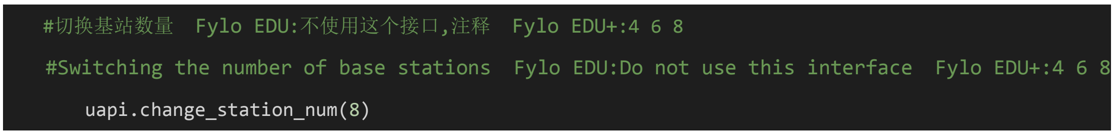
Call the uapi.change_plane_mode() function interface, 0 means the original Fylo EDU; 1 means the new Fylo EDU+ model



```
# 切换机型 Fylo EDU:0  Fylo EDU+:1
 #Switch models Fylo EDU:0  Fylo EDU+:1
    uapi.change_plane_mode(1)
```

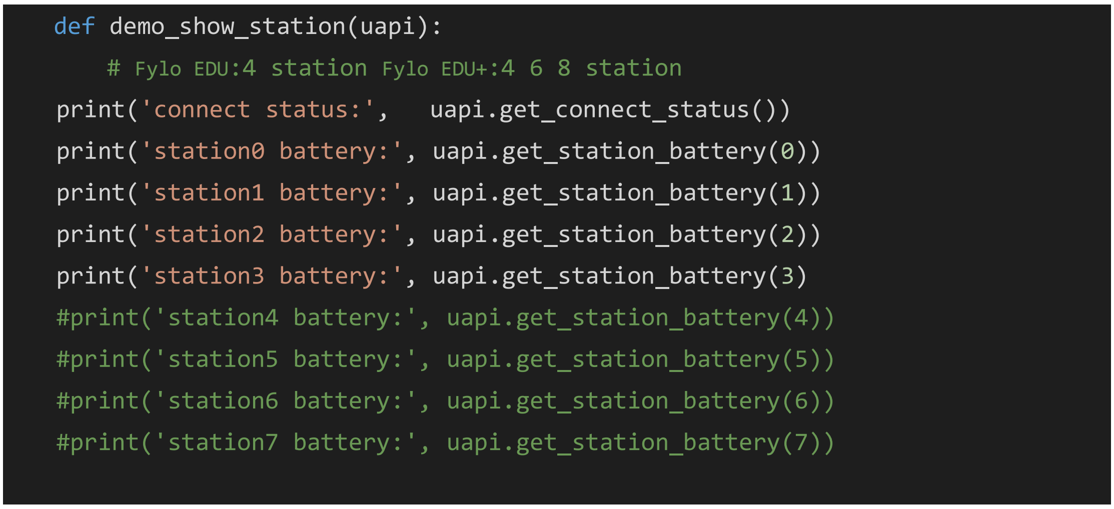
2、Switching quantity of base stations (only available in Fylo EDU+ model mode)

Call uapi.change_station_num() function interface, 4 is four base station mode; 6 is six base station mode; 8 is eight base station mode



```
#切换基站数量  Fylo EDU:不使用这个接口,注释  Fylo EDU+:4 6 8
#Switching the number of base stations  Fylo EDU:Do not use this interface  Fylo EDU+:4 6 8
    uapi.change_station_num(8)
```

Take Demo.py in the folder as an example：



```
def demo_show_station(uapi):
    # Fylo EDU:4 station Fylo EDU+:4 6 8 station
print('connect status:',   uapi.get_connect_status())
print('station0 battery:', uapi.get_station_battery(0))
print('station1 battery:', uapi.get_station_battery(1))
print('station2 battery:', uapi.get_station_battery(2))
print('station3 battery:', uapi.get_station_battery(3)
#print('station4 battery:', uapi.get_station_battery(4))
#print('station5 battery:', uapi.get_station_battery(5))
#print('station6 battery:', uapi.get_station_battery(6))
#print('station7 battery:', uapi.get_station_battery(7))
```

Please select the base station power information correctly in different base station quantity modes,4 base stations： print(uapi.get_station_battery(0--3)) base stations： print(uapi.get_station_battery(0--5))；

8base stations： print(uapi.get_station_battery(0--7))

```
    for id in range(4):
            coord = uapi.get_station_coordinate(id)
            if coord != None:
            # Fylo EDU: 'z': coord[2] 无效数据(invalid data)  Fylo EDU+: 'z': coord[2] 有
效数据(valid data )
                print('Station', id, ': x:', coord[0],
'y:',coord[1],'z:',coord[2])
                #print('Station', id, ': x:', coord[0], 'y:', coord[1])
```

After the base station is calibrated, obtain the status information of the base station, please correctly obtain the status of different numbers of base stations，for id in range(): 4,6,8 ； Please select the print content correctly to print the coordinate information. The original Fylo EDU model does not have it.'z': coord[2] Therefore choose print('Station', id, ': x:', coord[0], 'y:', coord[1]) ； The Fylo EDU+ model adds the function of setting the height of the base station， Therefore choose print('Station', id, ': x:', coord[0], 'y:', coord[1], 'z:', coord[2])

3、Set the height of the base station (only available in Fylo EDU+ model mode)

Call uapi.set_userheight() function interface, the height setting range is 300-2000mm

```
   #设置基站高度 Fylo EDU:不使用这个接口  Fylo EDU+: 300mm -- 2000mm
   #Setting base Station Height Fylo EDU:Do not use this interface   Fylo EDU+: 300mm -- 2000mm
      uapi.set_userheight(800)
```

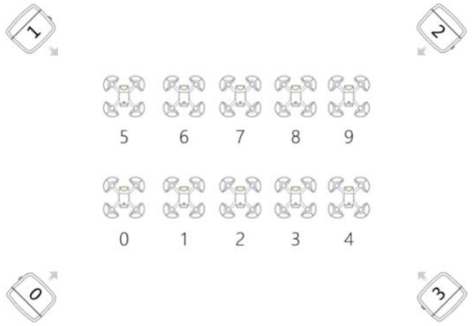
4、 Set the height positioning mode (only available in Fylo EDU+ model mode)

Call uapi.set_heightmode() function interface, tof height positioning: 0; tof+uwb mixed height positioning: 1; uwb height positioning: 2

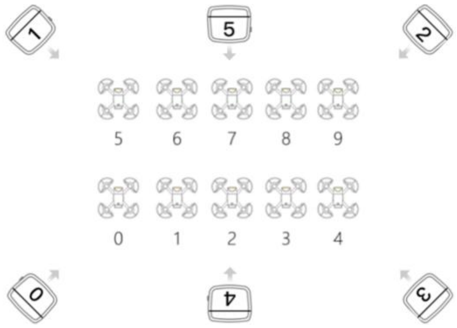
```
    # 切换定高模式  Fylo EDU:不使用这个接口 Fylo EDU+: tof:0 tof+uwb:1 uwb:2
   #Switch to fixed height mode  Fylo EDU:Do not use this interface  Fylo EDU+: tof:0 tof+uwb:1
uwb:2
      uapi.set_heightmode(1)
```

5、Base station placement

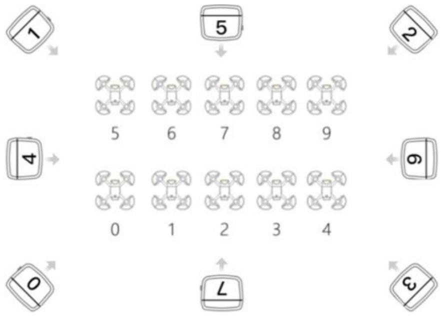
Fylo EDU+ 4 base stations placement



Fylo EDU+ 6 base stations placement



Fylo EDU+ 8 base stations placement



6、 Different calculation way of the base stations boundary range in different height positioning modes (to avoid the placement of base stations in different height positioning mode that can not meet the flight requirements)

Height Positioning Mode	4 base station	6 base station	8 base station
TOF +UWB	X≤15m Y≤15m z ≤ base station height + shortest side length*1.2	X≤30m Y≤15m z ≤ base station height + shortest side length*1.2	X≤30m y≤30m z ≤ base station height + shortest side length*1.2
TOF	X≤20m	X≤40m	X≤40m

	Y≤20m Z≤7m	Y≤20m Z≤7m	Y≤40m Z≤7m
UWB	X≤20m Y≤20m (base station height + longest interval*0.2) ≤ Z ≤ (base station height + shortest side length*1.2)	X≤40m Y≤20m (base station height + longest interval*0.2) ≤ Z ≤ (base station height + shortest side length*1.2)	X≤40m Y≤40m (base station height + longest interval*0.2) ≤ Z ≤ (base station height + shortest side length*1.2)

①X, Y and Z are the spatial coordinates of the base station respectively;

②Shortest side length: the minimum value among⓪-①, ①-②, ②-③and ⓪-③;

③Longest side length: the maximum value among ⓪-①, ①-②, ②-③and ⓪-③;

④ Longest interval: the maximum value among ⓪-④, ④-①, ①-⑤, ⑤-②, ②-⑥, ⑥-③, ③-⑦and⑦-⓪.

7、Upload firmware

Fylo EDU：

```
uapi.plane_sys_update_firmware('E:\\eduuav\\firmware\\PIL_01_20_531(beta4).bin')
```

Fylo EDU+:

```
uapi.plane_sys_update_firmware('E:\\eduuav\\firmware\\FYLO_01_20_732(beta3).fylo')
```

Note: The firmware version numbers of the two models are different, please select the corresponding firmware to upgrade correctly. The firmware path is the local firmware storage path.