Pyfylo Software Quick Reference Guide

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Author** |
| 1.0.0 | 2020-03-05 | Initial commit | Fathi,  |
| 1.0.1 | 2020-04-03 | Operating cmd and activating need administor user. | Fathi,  |

# 1 개요

Pyfylo소프트웨어는 실시간으로 비행기를 연결하고 제어하는 데 사용되며, 또한 프로그램에 의한 드론 대형형성에도 사용된다.

이 문서에서는 이 소프트웨어를 사용하기 위한 Quick guide와 Pyfylo 설치를 제공합니다.

이 소프트웨어 키트에는 다음과 같은 4개의 폴더가 포함되어 있습니다.

-Demo

-doc

-software

-whl

구조는 다음과 같습니다.



html 폴더에는 index.html라는 파일이 있는데, 이 파일은 인터넷 브라우저에서 열 수 있다. 여기에는 "userapi 모듈", 업데이트 문 등이 포함되어 있습니다. userapi 모듈의 모든 기능은 자세한 설명과 함께 표시됩니다.

**Warnning: 6번의 중요 문항은 비행기를 조종할 때마다 명심해야 한다!!**

# 2 SOFWARES 설치

## 1. 지원되는 운영체제

- windows 7 (x86/x86\_64)

- windows 10 (x86/x86\_64)

## 2. Python installer

Python 설치 프로그램이 소프트웨어 폴더에 있습니다. "amd64"로 명명된 것은 x86\_64 시스템에 적용되며, 다른 하나는 x86 시스템에 적용된다.

파이썬 공식 웹사이트(<https://www.python.org/downloads/windows/>) 에서도 동일한 설치 프로그램이 제공됩니다.

**Warning: 현재까지 python 3.6만 지원됩니다**.

## 3. Installing Python and Configure

A. python 설치 프로그램을 더블클릭합니다. 아래 그림과 같은 대화 창이 나타납니다. "Add python 3.6 to PATH"를 선택한 후 " Customize installation"을 선택합니다.



B. 다음 대화는 아래 그림과 같습니다. 모든 옵션을 선택하고 "다음" 단추를 클릭합니다.



C. 아래 그림으로 4개의 옵션을 선택하고 대상 경로로 "찾아보기"를 선택합니다. 또는 "설치" 단추를 클릭합니다.



D. 이 그림에는 설치 상태가 표시됩니다.



E. 아래 그림은 설치가 완료되었음을 의미합니다.



F. 파이썬 버전을 확인합니다. 윈도우키나 시작버튼을 누른후 검색창에 "CMD"를 입력합니다. 그리고 난 후 명령 프롬프트를 오른쪽 마우스로 클릭한 후 “관리자 권한으로 실행”합니다. 명령 프롬프트가 실행되면 “**python –V**”를 입력하여 다음 그림과 같이 설치 버전을 확인합니다. 설치한 버전과 동일하면 pyfylo 설치를 시작합니다.



## 4. Installing pyfylo

A. python의 "pip" 명령을 확인합니다. cmd를 상위 설명으로 열고 pip --version에 입력합니다. 반환된 결과가 아래 그림과 같으면 다음 단계로 이동하십시오.

B. Install whl.

 CMD창을 열고 다음과 같이 입력합니다.

 **pip install pyfylo**

C. whl 설치를 확인합니다. cmd창에 **pip list** 를 입력합니다. 그림과 같이 pyfylo, pyserial두가지가 확인되면 whl설치가 완료되었습니다.



## 5.Activating pyfylo

FyloTools.exe를 찾아 마우스 오른쪽 단추로 클릭합니다. 팝업 목록에서 "관리자로 실행"을 클릭합니다. 아래와 같이 창이 나타납니다. 그런 다음 'Activate'를 클릭하여 활성화합니다.

# 3 Updating

## updating pyfylo

cmd창에 다음과 같이 입력합니다. **pip install --upgrade pyfylo**.

## updating documents

# FyloTools.exe를 연 다음 'Download'를 클릭하여 최신 문서를 다운로드합니다.

# 4 Demo Folder

## 1. File Descriptions

**Demo.py** Fylo를 실시간으로 컨트롤하는 진입프로그램

**createdance.py** 댄스 프로그램 제작의 예

 **demo\_dance.py** 8개 기체에 대한 랜덤 댄스 프로그램 자동 생성 예제

 **dance\_lib.py demo\_dance.py의 보조파일**

 **cf.py demo\_dance.py 의 보조파일**

**diy\_dance.py** DIY 댄스 프로그램을 위한 기본 인터페이스

**check\_version.py** 기체펌웨어의 새 버전 감지를 위한 인터페이스

## 2. Entry Point Demo.py

이 파일은 실시간 제어의 모든 명령을 키보드의 키에 바인딩합니다.

특히, 숫자키“5”의 핵심은 실시간(realtime)컨트롤에서 **createdance.py**에서 설계된 프로그램으로 변경하는 것입니다.

이 파일을 사용하기 전에 파일에서 파이썬 코드를 읽고 모든 명령의 성과를 학습하는 것이 좋습니다. 다음은 몇 가지 기본 설정입니다.

|  |  |  |
| --- | --- | --- |
| 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 |  | Plane arm |
| C |  | Plane disarm |
| P | 0,1 | Set No. 0 plane’s id as No. 1 |
| V | 2,10 | Set No. 2 plane’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 plane enters compass calibrating |
| ] | 0 | No. 0 plane exits from compass calibrating |
| 1 | 0, 100, 1 | In realime No. 0 plane takes off and flies to 100cm height, with compass on |
| 2 | 0 | In realtime No. 0 plane touches down |
| 3 | plane\_id\_list, 1, 1000 | 1000ms later, planes take off and start dance programs with compass on |
| 4 | plane\_id\_list | Planes stop dance programs and touch down |
| 5 |  | Generate dance |
| 6 |  | Planes are prepared for takeoff |
| 7 |  | Dance time sync |
| 8 |  | Positioning inits |
| 9 | plane\_id\_list | Send programs to planes in list |
| 0 |  | stations demarcates |
| - | plane\_id\_list, yaw | Planes auto sync |
| = |  | Show dance in screen |
| , |  | Show demarcation |
| . |  | Show planes’ status |
| ’ |  | Show status of planes’ sensors |
| / |  | Print help information |
| CTRL+C |  | Exit pyfylo |

## 3. Createdance.py

이 예제 파일에서 create\_dance는 댄스 프로그램을 만드는 시작점이다. 그리고 4가지 종류의 댄스 프로그램에 따라 "flag"라는 파라미터가 바뀝니다.

* flag = 0, 0번 드론이 보여주는 userapi 모듈을 이용한 단일 평면의 간단한 댄스 프로그램,
* flag = 1, userapi 모듈을 사용하여 만든 8개의 드론 편대 댄스 프로그램,
* flag = 2, 0번 드론이 보여주는 diy\_dance.py를 추가로 사용하여 단일 평면 댄스 프로그램의 모든 단계를 반자동으로 구현한다.
* flag = 3, dance\_lib.py 및 cf.py를 사용하여 8대의 드론을 위한 포메이션 댄스 프로그램의 자동 램덤 구현.

이 코드들은 우리 상품의 기초적인 설명이다.

사용자들은 더 많은 프로그래밍 기술로 더 다양한 댄스 프로그램을 구현할 수 있다.

## 4. check\_version.py

이 파일을 실행하여 pyfylo 및 비행기의 펌웨어 버전을 확인합니다.

실행하기 전에 다음을 구성하십시오.:

1. 사용자 환경에 따라 다운로드, download\_path,pland\_id\_list 구성합니다.
* download, 다운로드가 True일 때 이 프로그램은 업데이트가 필요할 경우 pyfylo를 다운로드합니다. 다운로드가 False이면 이 프로그램은 pyfylo를 다운로드하지 않습니다..
* download\_path, pyfylo의 다운로드 경로입니다.
* plane\_id\_list, 기체 id 체크.
1. 컴퓨터가 인터넷에 액세스할 수 있는지 확인합니다..
2. 확인하려는 드론과 기지국을 켭니다.
3. 업그레이드 과정에서 인터넷이 필요하므로 리피터를 사용하여 연결하십시오.

# 5 Quick Start

먼저 4개의 기지국과 0번 드론을 아래 그림과 같이 배치하십시오.

## Controlling in realtime

1. 0번 드론 자리에 서서 휴대폰으로 나침반 앱을 열고 휴대전화가 비행기와 같은 방향인지 확인합니다. 그런 다음 앱의 요 각도를 demo.py의 파라미터 yaw에 복사하고 파일을 저장합니다.
2. 스테이션과 드론을 상단 그림으로 배치합니다. 전원을 켜세요. 그리고 컴퓨터를 시스템에 연결합니다. Cmd를 열고 demo.py 경로로 이동합니다. **python demo.py**를 입력합니다
3. 기지국 설정 (Key “**0**”).
4. 기지국 상태 확인 (Key “**,**”). 4개의 측점의 좌표는 순서대로 위 그림과 같이 사각형의 꼭지점처럼 보정되어야 한다.
5. 위치시스템 inits(Key “**8**”).
6. 드론상태를 확인합니다(Key “**.**”). 반환된 결과에는 "aux setup: Yes" 및 "rangesafe: Yes"가 포함되어야 합니다.
7. 이륙(Key “**1**”).
8. 실시간 컨트롤(Keys “**W**”, “**A**”, “**D**”, “**S**”, “**Q**”, “**E**”, “**J**”, “**K**”, “**L**”, “**Z**”).
9. 착륙(Key “**2**”).

## Dance Programs

1. 0번 비행기 자리에 서서 휴대폰으로 나침반 앱을 열고 휴대전화가 비행기와 같은 방향인지 확인합니다. 그런 다음 앱의 요 각도를 demo.py의 파라미터 yaw에 복사하고 파일을 저장합니다.
2. 스테이션과 드론을 상단 그림으로 배치합니다. 전원을 켜세요. 그리고 컴퓨터를 시스템에 연결합니다. Cmd를 열고 demo.py 경로로 이동합니다. **python demo.py**를 입력합니다
3. 기지국 설정 (Key “**0**”).
4. 기지국 상태 확인 (Key “**,**”). 4개의 측점의 좌표는 순서대로 위 그림과 같이 사각형의 꼭지점처럼 보정되어야 한다.
5. 댄스 프로그램 생성 (Key “**5**”).
6. 댄스 스텝을 화면에 표시 (Key “**=**”).
7. 댄스프로그램 업로드(Key “**9**”).
8. 드론 상태확인(Key “**.**”). 반환된 결과에 "dance: Yes"가 포함되어 있는지 확인합니다.
9. 댄스 타임싱크(Key “**7**”).
10. 드론 상태확인(Key “**.**”). 반환된 결과에 “timesync: Yes”가 포함되어 있는지 확인합니다.
11. 위치확인 시스템 설정(Key “**8**”).
12. 드론 상태확인(Key “**.**”). 반환된 결과에 “aux setup: Yes” and “rangesafe: Yes” 가 포함되어 있는지 확인합니다.
13. 드론 이륙 준비 (Key “**6**”).
14. 드론 상태확인(Key “**.**”). 반환된 결과에 “takeoff allow: Yes” 가 포함되어야 합니다.
15. 군집드론dance show 시작(Key “**3**”)
16. 군집비행 중지하고 착륙(Key “**4**”). 필요한 경우 언제든지 착륙할 수 있습니다.

**Note:**

**자동 구성 기능(키 "-")은 시간 동기화 및 위치 지정 초기화를 구현할 수 있습니다. 호출 후 비행기 상태를 확인하고 반환된 결과에 "aux setup: Yes" 및 "rangesafe: Yes"가 포함되는지 확인합니다.**

## Global Variable Declaration (글로벌 변수 선언)

**plane\_id\_list**: 드론 목록

**connect\_type:** Wifi 연결 = 0, 리피터 연결 = 1, 연결해제 = 2

**yaw:** 드론의 yaw 각도

# 6 Warnings(경고)

|  |
| --- |
| 1. 댄스 쇼가 시작되기 전에 항상 비행기의 상태를 확인하고 반환된 결과에 "aux setup: Yes", "rangesafe: Yes", "timesync: Yes" 및 "takeoff allow: Yes"가 포함되는지 확인합니다.
 |
| 1. 공연 전에 사용할 기능의 파라미터를 구성합니다.
 |
| 1. 0번 드론은 모든 비행기의 기준점이 되어야 하므로 항상 존재해야 한다.
 |
| 1. 이륙 전 항상 드론이 초기 위치에 있는지 확인하고 위치 확인 시스템 초기화 및 시간 동기화를 완료한다.
 |
| 1. 드론의 이륙 전, 기지국설정 후 기지국이 이동되면 다시 설정해야하고, 위치 확인 시스템 초기화 후 드론이 이동하면 위치 확인 시스템 초기화를 다시 실시한다..
 |
| 1. 나침반을 사용하는 경우, 먼저 나침반을 보정하고 함수의 상대 파라미터를 1로 설정합니다..
 |
| 1. 이륙 전 항상 드론의 위치에서 비행기의 yaw 각도가 휴대폰의 나침반 앱의 판독값과 같은지 확인한다.
 |
| 1. 비행중 비상상황 발생시 “plane\_fly\_disarm” 나 “C”키를 사용하여 비상착륙 한다.
 |