

---

## 1. 实验名称及目的

**固定翼速度/高度/偏航接口验证实验(Python):** 该例程以 Python 的形式，通过平台固定翼接口，实现在软硬件在环仿真过程中固定翼按期望指令飞行。

## 2. 实验原理

### 2.1. 软/硬件在环仿真（SIL/HIL）的实现[1][2]

从实现机制的角度分析，可将 RflySim 平台分为运动仿真模型、底层控制器、三维引擎、外部控制四部分。

- **运动仿真模型：**这是模拟飞行器运动的核心部分。在 RflySim 平台中，运动仿真模型是通过 MATLAB/Simulink 开发的，然后通过自动生成的 C++ 代码转化成 DLL（动态链接库）文件。在使用 RflySim 平台进行软硬件在环仿真时，会将 DLL 模型导入到 CopterSim，形成运动仿真模型。这个模型在仿真中负责生成飞行器的运动响应，它拥有多个输入输出接口与底层控制器、三维引擎、地面控制站和外部控制进行数据交互，具体数据链路、通信协议及通信端口号见 [API.pdf 中的通信接口部分](#)。
- **底层控制器：**在软/硬件在环仿真（SIL/HIL）中，真实的飞行控制硬件（如 PX4 飞行控制器）被集成到一个虚拟的飞行环境中。在软件在环仿真（SIL）中，底层控制器（通过 wsl 上的 PX4 仿真环境运行）通过网络通信与运动仿真模型交互数据。在硬件在环仿真（HIL）中，它（将 PX4 固件在真实的飞行控制器（即飞控）硬件上运行）则通过串口通信与运动仿真模型进行数据交互。飞控与 CopterSim 通过串口（硬件在环 HITL）或网络 TCP/UDP（软件在环 SITL）进行连接，使用 MAVLink 进行数据传输，实现控制闭环。
- **三维引擎：**这部分负责生成和处理仿真的视觉效果，提供仿真环境和模型的三维视图，使用户能够视觉上跟踪和分析飞行器的运动。CopterSim 发送飞机位姿、电机数据到三维引擎，实现可视化展示。
- **外部控制（offboard）：**从仿真系统外部对飞行器进行的控制，包括自动飞行路径规划、远程控制指令等。在平台例程中主要通过地面控制站（QGC）、MATLAB 和 Python 调用对应接口实现。

### 2.2. 通过外部控制接口（python）进行速度/高度/偏航控制

单机控制脚本 AircraftMathworksController.py 中依次调用了 RflySim 平台飞机控制接口协议文件 PX4MavCtrlV4.py 中定义的以下接口函数

#### 创建通信示例

```
mav1 = PX4MavCtrl.PX4MavCtrl(1)
```

创建一架飞机的通信示例

---

## 启用 Mavlink 消息监听循环

```
mav1.InitMavLoop()
```

配置 CopterSim 通信模式，该函数的参数定义如下：

```
def InitMavLoop(self,UDPMode=2):
    """ Initialize MAVLink listen loop from CopterSim
        0 and 1 for UDP_Full and UDP_Simple Modes, 2 and 3 for MAVLink_Full and
        MAVLink_Simple modes, 4 for MAVLink_NoSend
        The default mode is MAVLink_Full
    """
```

默认通信模式为 **Mavlink\_Full**：Python 直接发送 MAVLink 消息给 CopterSim，再转发给 PX4，数据量较大适合单机控制；适合单机或少量飞机仿真，无人机数量小于 4；

## 设定航路点

```
n = 30
r = 400
missionPoints=[]
for i in range(n):
    angle = 2*math.pi*i/n
    x=r*math.sin(angle)
    y=r*math.cos(angle)
    missionPoints.append([x,y,-100])
```

用一组离散的点模拟圆形运动轨迹，并在循环中通过 `append` 方法逐个将相应的轨迹点存入目标点列表（`missionPoints`）。`missionPoints.append([x,y,-100])`表示在 `missionPoints` 列表的末尾添加一个新的列表`[x,y,-100]`。

根据欧拉公式：

$$e^{ix} = \cos x + i\sin x$$

这些点将在 x-y 平面上形成一个圆形轨迹。

## 飞行阶段

完成上述设置后，程序会通过检查一个 `flag` 变量的值来决定无人机应该执行哪些动作。

**当 `flag == 0` 时，解锁飞机**

解锁飞机

```
mav1.SendMavArm(True)
```

设定起飞目标点

```
targetPos=[200, 0, -100]
mav1.sendMavTakeOff(targetPos[0],targetPos[1],targetPos[2])
```

发送绝对的 GPS 坐标作为起飞目标点，使用 `sendMavTakeOffGPS` 命令，最后三位分别是经度、维度、和高度，会先从 `uavPosGPSHome` 向量中提取解锁 GPS 坐标，在此基础上用绝对坐标

**当 `flag == 1` 时，无人机起飞和进入航路寻迹模式**

位置检测

```
curPos=mav1.uavPosNED
```

```
dis = math.sqrt((curPos[0]-targetPos[0])**2+(curPos[1]-targetPos[1])**2)
```

计算飞机当前位置和起飞目标位置的水平距离，用于判断是否到达目标位置，以开始下一阶段任务。

#### 启动外部控制（offboard）

```
mav1.initOffboard()
```

使 px4 控制器进入外部控制模式，且以 30HZ 的频率发送 offboard 指令

#### 航路寻迹模式

```
targetPos=missionPoints[flagI]  
mav1.SendPosNED(targetPos[0], targetPos[1], targetPos[2])
```

会通过航路点索引 flagI 的值从 missionPoints 列表中读取相应的航点，并通过 SendPosNED 函数更新为下一个目标点。

当 flag == 2 时，发送期望速度、偏航和高度

```
mav1.SendVelYawAlt(10,math.pi/2,-150)
```

发送期望速度、偏航和高度，测试命令是否能正确运行，可以观测速度是否为 10，高度是否为 150

### 3. 实验效果

固定翼在软硬件在环仿真中，能按照期望指令飞行。

### 4. 文件目录

文件夹/文件名称	说明
AircraftMathworks.dll	固定翼 DLL 模型文件。
AircraftMathworksVelYawAltCtrl.py	固定翼速度、高度、偏航例程文件。
AircraftMathworksHITLRun.bat	硬件在环仿真批处理文件。
AircraftMathworksSITLRun.bat	软件在环仿真批处理文件。
PX4MavCtrlV4.py	RflySim 平台视觉/集群控制接口文件。

### 5. 运行环境

序号	软件要求	硬件要求	
		名称	数量
1	Windows 10 及以上版本	笔记本/台式电脑 <sup>①</sup>	1
2	RflySim 平台免费版	PX4 飞控 <sup>②</sup>	1
3	Python3.11	数据线	1

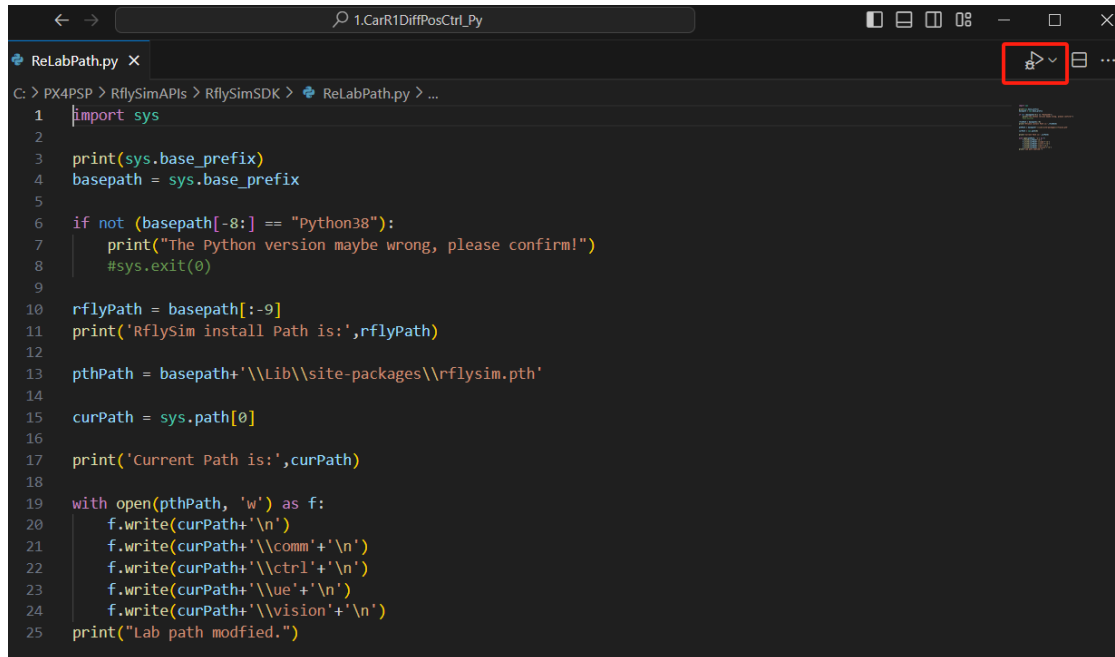
① 推荐配置请见：<https://doc.rflysim.com/1.1InstallMethod.html>

② 须保证平台安装时的编译命令为：px4\_fmu-v5\_default，固件版本为：1.12.3。其他配套飞控请见：<http://doc.rflysim.com/hardware.html>

## 6. 实验步骤

### 6.1. Python 库文件部署

以 VsCode 打开 “C:\PX4PSP\RflySimAPIs\RflySimSDK\ReLabPath.py”，并运行。



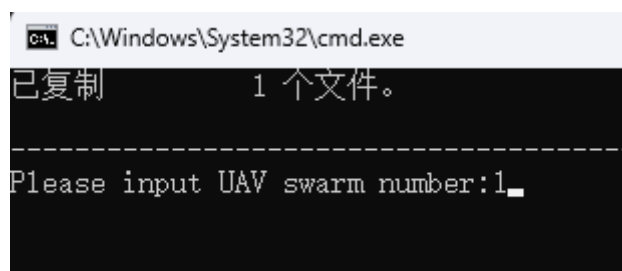
```
1 import sys
2
3 print(sys.base_prefix)
4 basepath = sys.base_prefix
5
6 if not (basepath[-8:] == "Python38"):
7     print("The Python version maybe wrong, please confirm!")
8     #sys.exit(0)
9
10 rflyPath = basepath[:-9]
11 print('RflySim install Path is:', rflyPath)
12
13 pthPath = basepath+'\\Lib\\site-packages\\rflsim.pth'
14
15 curPath = sys.path[0]
16
17 print('Current Path is:', curPath)
18
19 with open(pthPath, 'w') as f:
20     f.write(curPath+'\n')
21     f.write(curPath+'\\comm'+'\n')
22     f.write(curPath+'\\ctrl'+'\n')
23     f.write(curPath+'\\ue'+'\n')
24     f.write(curPath+'\\vision'+'\n')
25 print("Lab path modified.")
```

完成 Python 公共库环境部署。

### 6.2. 软件在环仿真

#### Step 1:

右键以管理员身份运行 “AircraftMathworksSITLRun.bat” 批处理文件，在弹出的终端窗口中输入 1，启动 1 架飞机的软件在环仿真。



#### Step 2:

完成初始化。



### Step 3:

打开 AircraftMathworksVelYawAltCtrl.py 文件并运行，运行一段时间后可看到固定翼按期望指令(速度、高度、偏航角)飞行。

```

AircraftMathworksVelYawAltCtrl.py 9+ x
D: > OneDrive > 桌面 > VelAltYawCtrlAPI > PythonCtrlAPI > AircraftMathworksVelYawAltCtrl.py > ...
64
65     if flag==1:
66         curPos=mav1.uavPosNED
67         dis = math.sqrt((curPos[0]-targetPos[0])**2+(curPos[1]-targetPos[1])**2)
68         if dis < 50:
69             print("到达起飞位置")
70             flag = 2
71             flagTime=time.time()
72             flagI=0
73             mav1.initOffboard()
74             print("开始进入Offboard模式")
75
76             print("开始进入航路寻迹模式")
77             targetPos=missionPoints[flagI]
78             #mav1.SendPosNEExt(targetPos[0], targetPos[1], targetPos[2], 3) # For PX4 1.22
79             mav1.SendPosNED(targetPos[0], targetPos[1], targetPos[2])
80             #设置盘旋半径为20m
81             mav1.SendCruiseRadius(20)
82
83     if flag == 2:
84         #print("发送期望速度、高度和偏航")
85         mav1.SendVelYawAlt(10,math.pi/2,-150) # 测试命令是否能正确运行，看下速度是否为10，高度是否为100

```

PROBLEMS 41 OUTPUT TERMINAL DEBUG CONSOLE

```

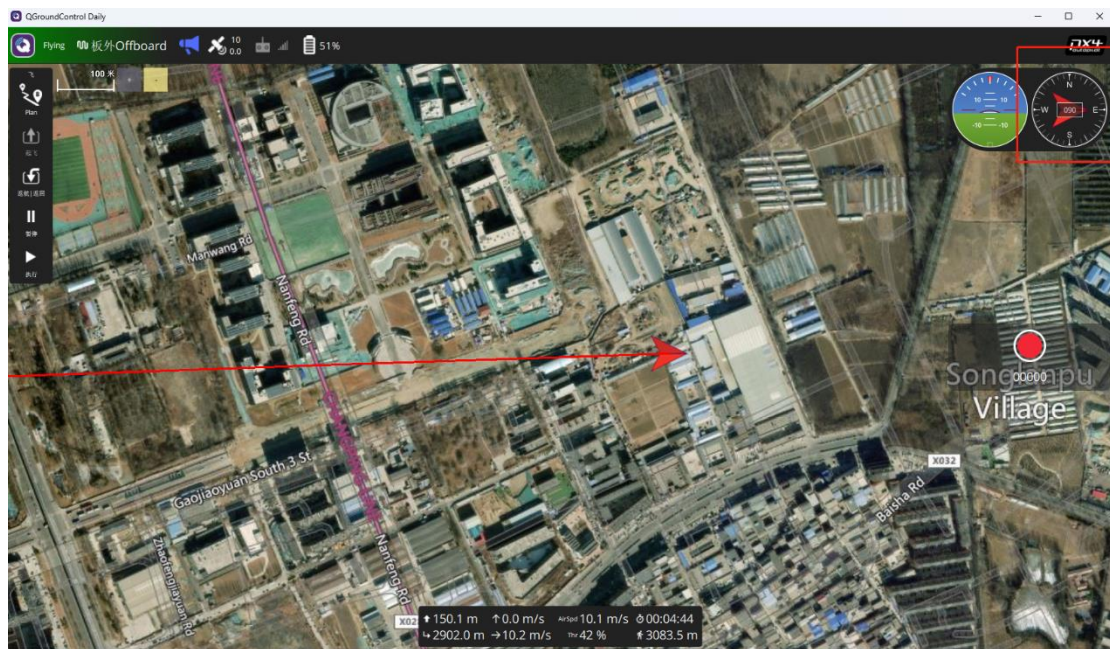
PS Z:\工程师\2022-叶颖鑫-模型组\4-项目文件\5-例程整理\童俊豪\PythonCtrlAPI\固定翼航向角控制> d.; cd 'd:\OneDrive\桌面\VelAltYawCtrlAPI\PythonCtrlAPI'
extensions\ms-python.python-2023.12.0\pythonFiles\Lib\python\debugpy\adapter\..\..\debugpy\launcher' '55464' '--' 'D:\OneDrive\桌面\VelAltYawCtrlAPI\PythonCtrlAPI'
5s, Arm the drone!
Arm the drone!
开始起飞

```





在 QGC 右上角罗盘处可以看到偏航角为  $90^{\circ}$  。



在 CopterSim 右下角可以看到飞行合速度即空速约为 10m/s，飞行高度约为 100m。

X 1515.43	Y 650.669	Z 100.079
Vx 0.111	Vy 10.295	Vz 0
$\phi$ 0.653	$\theta$ 2.716	$\psi$ 88.879

### 6.3. 硬件在环仿真





#### Step 1:

按下图所示将飞控与计算机链接。



## Step 2:

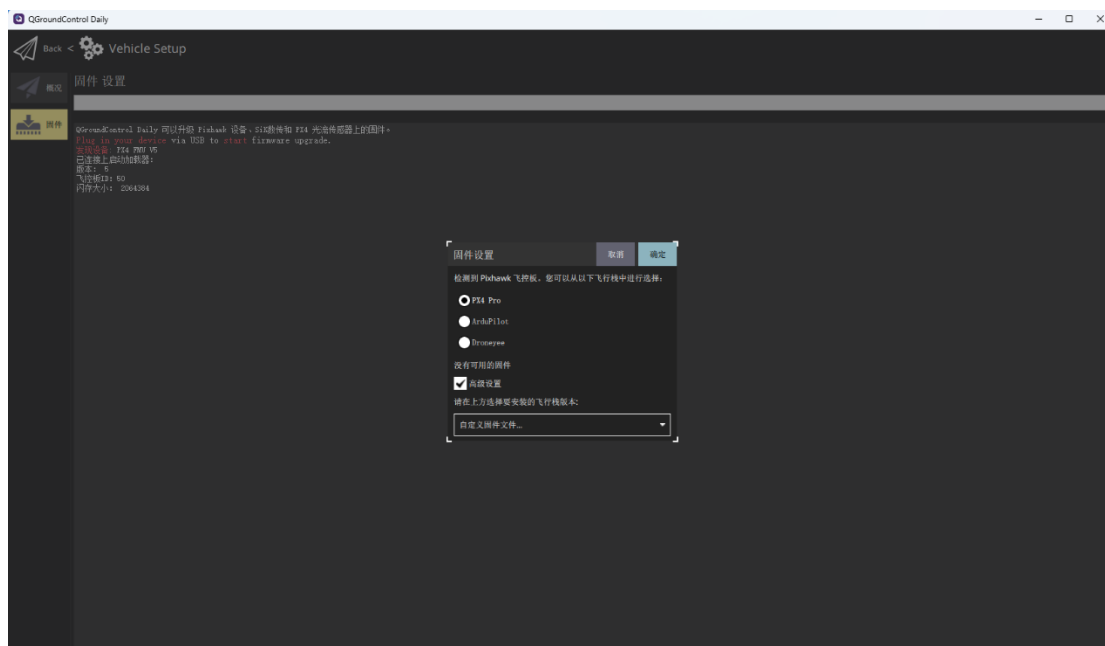
在 Rflytools 文件夹中打开 QGC 地面站。

 3DDisplay	2023/7/27 15:02	快捷方式	1 KB
 CopterSim	2023/7/27 15:02	快捷方式	1 KB
 FlightGear-F450	2023/7/27 15:02	快捷方式	2 KB
 HITLRun	2023/7/27 15:02	快捷方式	2 KB
 Python38Env	2023/7/27 15:02	快捷方式	2 KB
 QGroundControl	2023/7/27 15:02	快捷方式	1 KB
 RflySim3D	2023/7/27 15:02	快捷方式	1 KB
 RflySimAPIs	2023/7/27 15:02	快捷方式	1 KB
 RflySimUE5	2023/7/27 15:02	快捷方式	1 KB
 SITLRun	2023/7/27 15:02	快捷方式	2 KB
 Win10WSL	2023/7/27 15:02	快捷方式	2 KB

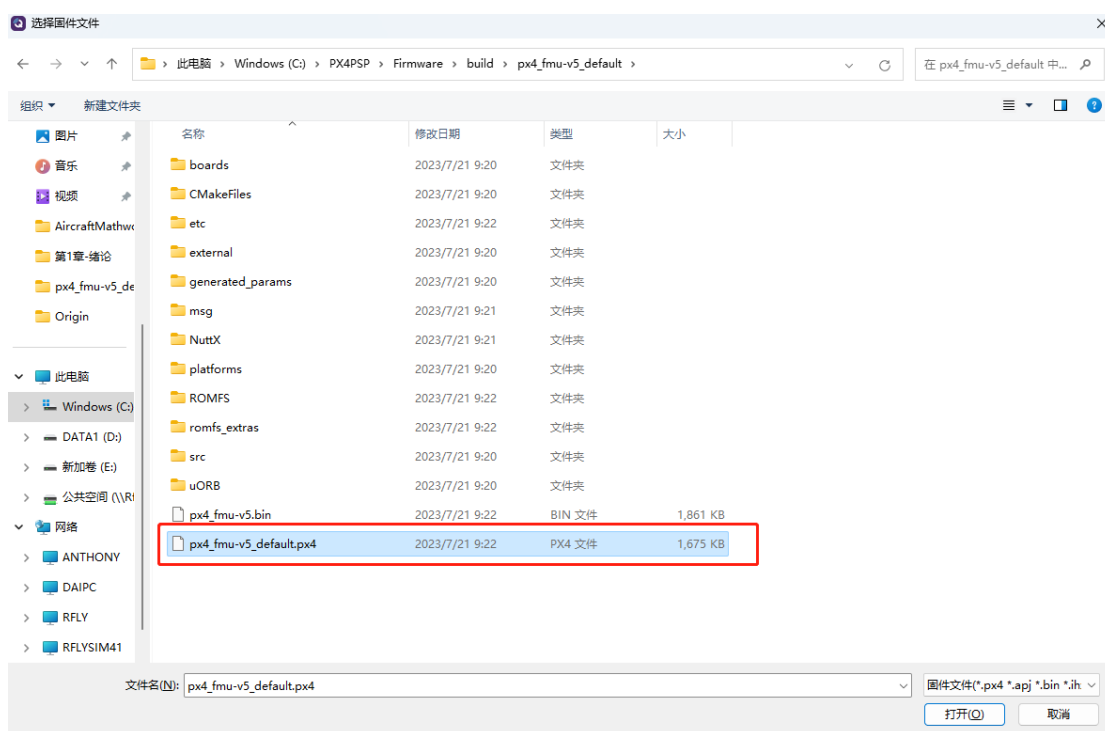
## Step 3:

点击进入左侧“固件”界面后，勾选下方“高级设置”选择自定义固件文件。





在 C:\PX4PSP\Firmware\build\px4\_fmu-v5\_default 这个路径下选择确认 px4\_fmu-v5\_default.px4 文件。

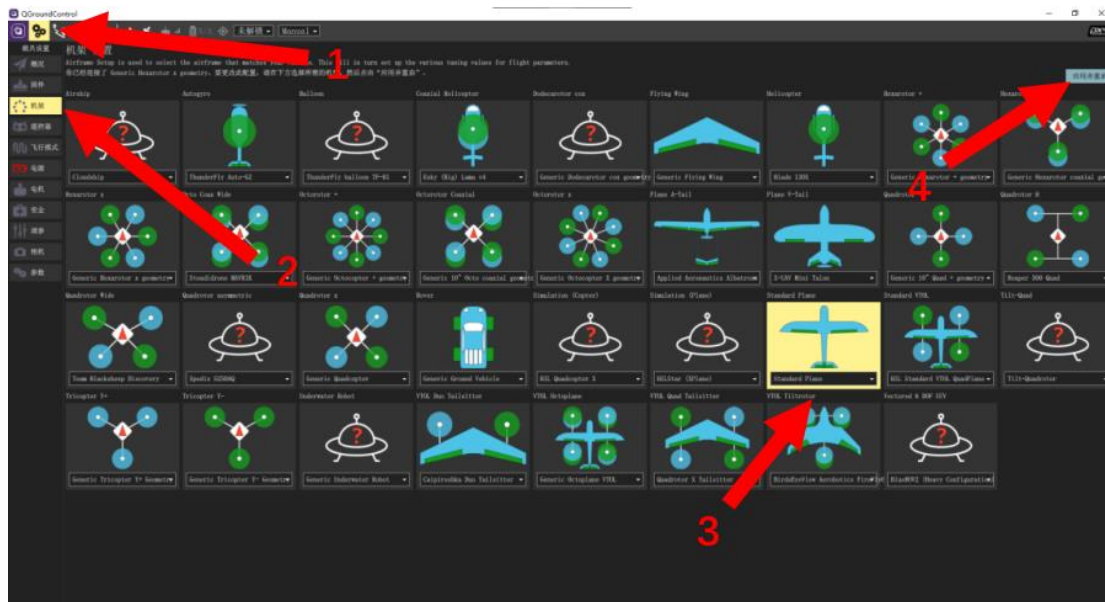


如果选择卓翼 H7 飞控的话，则在 C:\PX4PSP\Firmware\build\droneyee\_zyfc-h7\_default 这个路径下确认 droneyee\_zyfc-h7\_default.px4 文件。

名称	修改日期	类型	大小
boards	2023/7/28 11:05	文件夹	
CMakeFiles	2023/7/28 11:05	文件夹	
etc	2023/7/28 11:07	文件夹	
external	2023/7/28 11:05	文件夹	
generated_params	2023/7/28 11:05	文件夹	
msg	2023/7/28 11:06	文件夹	
NuttX	2023/7/28 11:06	文件夹	
platforms	2023/7/28 11:05	文件夹	
ROMFS	2023/7/28 11:07	文件夹	
romfs_extras	2023/7/28 11:07	文件夹	
src	2023/7/28 11:05	文件夹	
uORB	2023/7/28 11:05	文件夹	
droneyee_zyfc-h7.bin	2023/7/28 11:07	BIN 文件	1,805 KB
droneyee_zyfc-h7_default.px4	2023/7/28 11:07	PX4 文件	1,617 KB

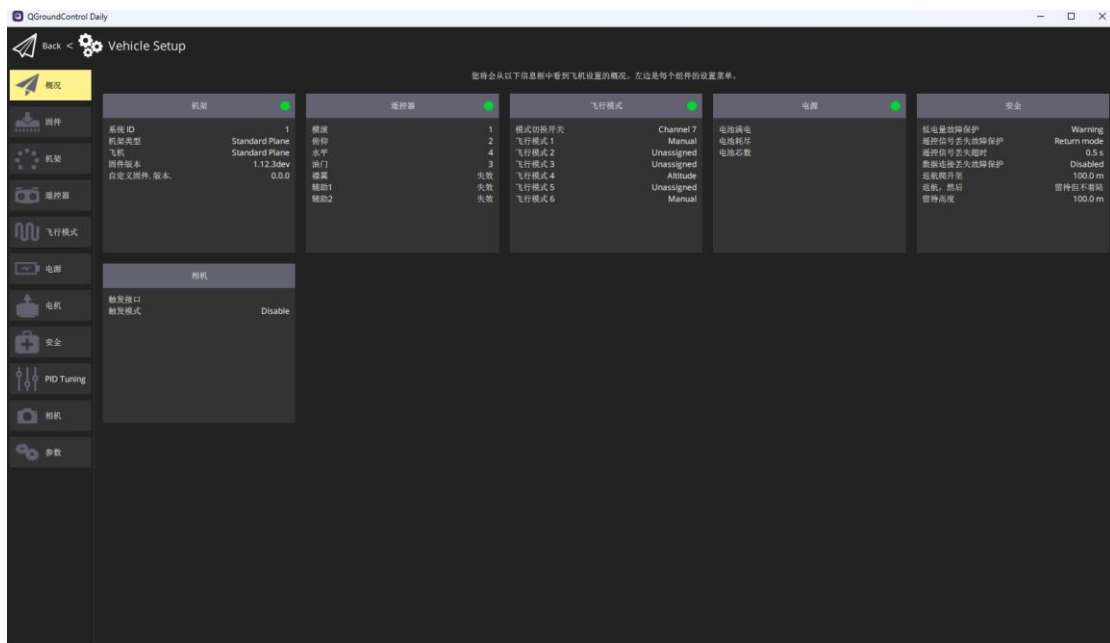
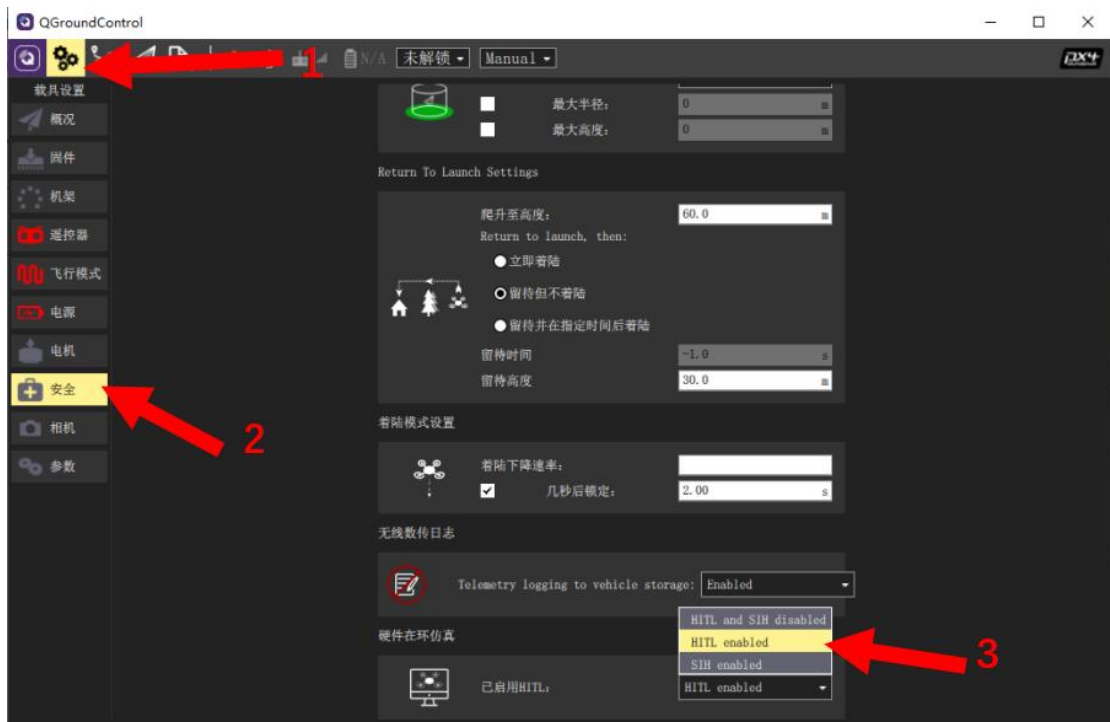
## Step 4:

在机架界面设置机架型号为“Standard Plane”，设置完毕后点击右侧“应用并重启”。



## Step 5:

在“安全”界面，选择“HITL enabled”启动硬件在环仿真，之后在概况界面中确认配置完成后，重新插拔飞控完成设置。



## Step 6:

右键以管理员身份运行“AircraftMathworksHITLRun.bat”批处理文件，在弹出的终端窗口中根据提示输入串口号 5，启动一架飞机的硬件在环仿真。

__pycache__	2023/7/24 11:00	文件夹	
AircraftMathworks.dll	2022/7/27 19:11	应用程序扩展	255 KB
AircraftMathworksMavlinkHITLRun	2022/9/20 17:07	Windows 批处理...	6 KB
AircraftMathworksMavlinkSITLRun	2022/9/20 17:07	Windows 批处理...	6 KB
AircraftMathworksVelYawAltCtrl	2023/7/21 15:06	PY 文件	3 KB
PX4MavCtrlV4	2023/6/7 10:43	PY 文件	137 KB

```
C:\Windows\system32\cmd.e: X + v
已复制 1 个文件。
-----
Please input the Pixhawk COM port list for HIL
Use ',' as the separator if more than one Pixhawk
E.g., input 3 for COM3 of Pixhawk on the computer
Input 3,6,7 for COM3, COM6 and COM7 of Pixhawks

Available COM ports on this computer are:
COM3: ??????????
COM4: ??????????
COM5: USB ????
```

Step 7:

之后测试步骤与软件在环的 Step2 到 Step3 相同，运行之后在 CopterSim、RflySim3D 和 QGC 中观察飞机是否按期望指令飞行。

注意事项：在固定翼的 offboard 控制中，用到如下控制接口：

- 1) SendMavTakeOff: 起飞指令。
- 2) SendVelYawAlt: 发送速度、偏航与高度指令。

7. 参考资料

[1]. DLL/SO 模型与通信接口 [..\..\API.pdf](#)  
[2]. 外部控制接口 [..\..\API.pdf](#)  
[3].

8. 常见问题

Q1.  
A1.