

# 1.实验名称及目的

## 1.1实验名称

基于最大模板的全故障模块注入的原理。

## 1.2实验目的

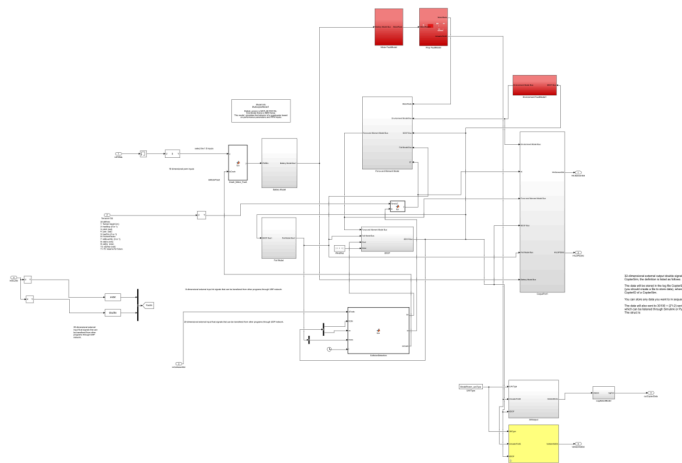
对最大模板的全故障模块故障建模，将故障建模的模型导出为DLL文件，再通过CopterSim加载DLL文件，最后通过udp模式（python/matlab形式）注入故障码进行故障注入仿真。

## 1.3关键知识点

无

# 2.实验效果

可输入以下任意故障或其组合。



故障ID	故障类型	故障参数
123450	电机执行效率故障	#1~#x号电机执行效率比（0~1）
123451	螺旋桨故障	#1~#x号螺旋桨执行效率比（0~1）

故障ID	故障类型	故障参数
123452	电池失效故障	无
123453	低电压故障	电压失效比 (0~1)
123454	低电量故障	电量失效比 (0~1)
123455	负载故障	重量泄露比 (0~1)
123456	负载漂移故障	重量泄露比+x,y,z的泄露因子 (0~1)
123457	负载泄露故障	重量泄露比+泄露因子(0~1)
123458	常风故障	X,y,z轴的风速
123459	阵风故障	阵风强度+风到达时间
123540	紊流风故障	紊流风强度
123541	切向风故障	切向风强度
123542	加速度计噪声干扰	噪声增益
123543	陀螺仪噪声干扰	噪声增益
123544	磁力计噪声干扰	噪声增益
123545	气压计噪声干扰	噪声增益
123546	GPS故障	噪声增益+3D方式+星数

说明文档	说明文档链接以及地址
simulink中模块合集	<a href="#">PX4PSP/RflySimSDK/html/md_phm_2md_2PHMmodels.html</a>
<a href="#">FaultInjectAPITest.py</a>	<a href="#">PX4PSP\RflySimAPIs\7.RflySimPHM\0.ApiExps\xe4_FaultInjectAPITest_py\readme.pdf</a>
PX4MavCtrlV4库	<a href="#">PX4PSP/RflySimSDK/html/PX4MavCtrlV4_8py.html</a>

主要要求掌握：熟悉最大模板的使用方法及其内部实现。

## 3. 文件目录

例程目录：

[安装目录]\RflySimAPIs\7.RflySimPHM\2.AdvExps\e1\_FullFaultModelTemplate

文件夹/文件名称	说明
MulticopterModel.slx	故障注入模块的最大模板模型文件。
MulticopterModelHITL.bat	硬件在环仿真批处理文件。
MulticopterModelSITL.bat	软件在环仿真批处理文件。
GenerateModelDLLFile.p	DLL格式转化文件。
FullFaultModelTemplate_Init.m	动力学模型相关参数。
readme.txt	全故障代码说明文档。
Python38Run.bat	Python程序执行脚本。
<a href="#">FaultInjectAPITest.py</a>	故障注入程序。

## 4. 运行环境

### 4.1 软件要求

Windows 10及以上版本；RflySim工具链；MATLAB 2022B及以上版本。

①：若使用Pixhawk 6X飞控，平台安装时的编译命令为：px4\_fmu-v6x\_default，推荐PX4固件版本为：1.12.3。其他配套飞控及编译命令请见：

<https://rflysim.com/doc/zh/1/Hardware.html>

### 4.2 硬件要求

笔记本/台式电脑① 1台。

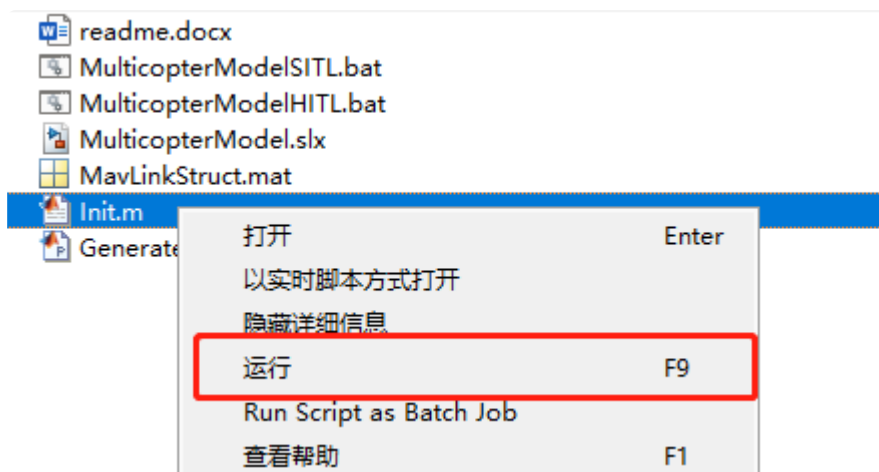
①：推荐配置请见：<https://rflysim.com/>

# 5. 实验步骤

## 5.1. 必做实验

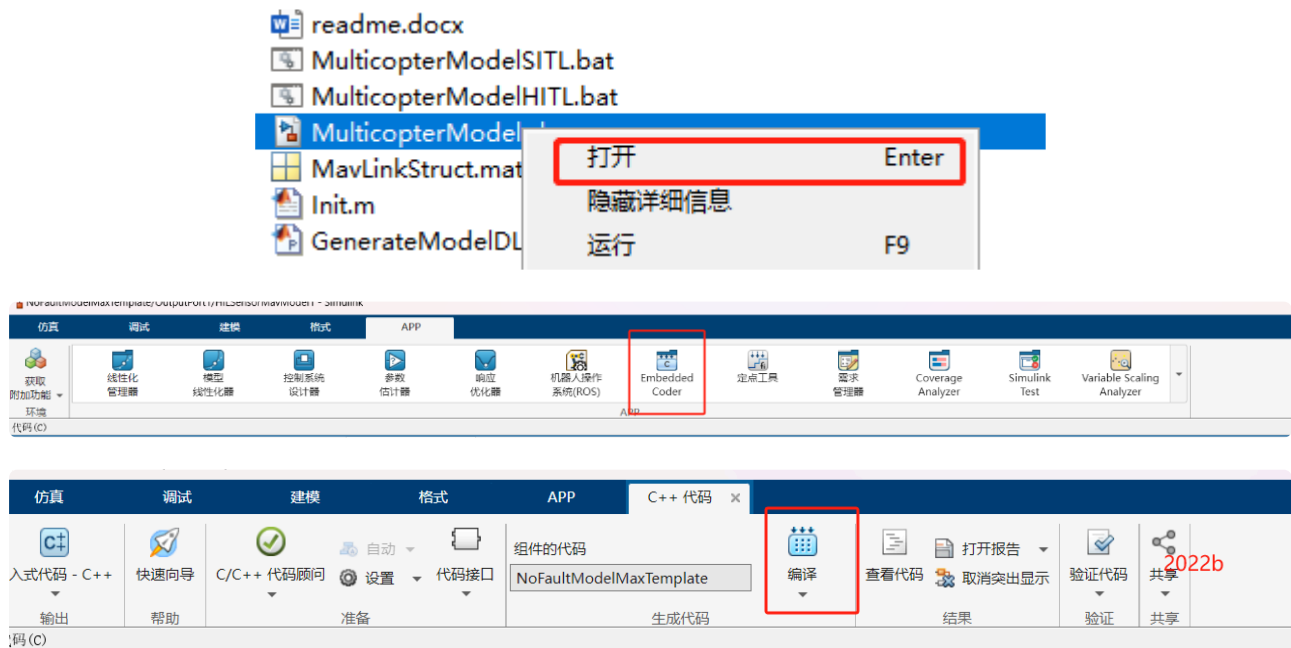
### Step 1: 初始化数据

打开“Init.m”文件并运行。



### Step 2: 编译文件

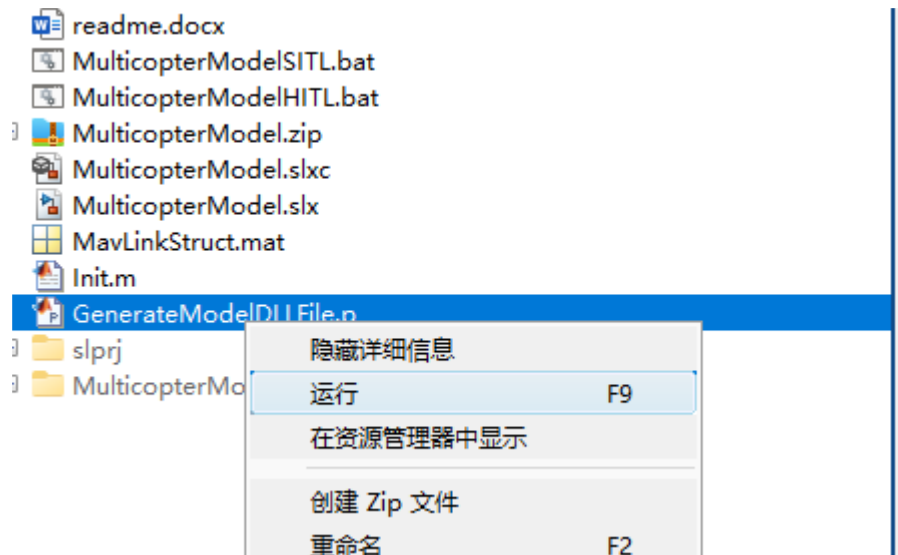
打开“MulticopterModel.slx” Simulink 文件，点击Build Model 按钮生成代码。



注：如果故障模块版本错误，无法编译，需要从故障模块库中选择对应的模块进行替换。

## Step 3: 生成dll文件

代码生成完毕后，在 matlab 中右键“GenerateModelDLLFile.p”文件，点击运行，生成 DLL 文件。



## Step 4: 运行在环脚本

以管理员身份运行软件在环脚本。



## Step 5: 打开python文件，修改故障参数

打开Visual Studio

Code，选择打开文件夹，打开文件夹

RflySimAPIs\7.RflySimPHM\0.ApiExps\e4\_FaultInjectAPITest\_py。



对FaultInjectAPITest.py其中的故障注入代码按照 RflySimAPIs\7.RflySimPHM\0.ApiExps\e4\_FaultInjectAPITest\_py中的 FaultInjectAPITest\_py中的故障注入代码进行更改（各个模块故障注入代码可以查看参考文献，具体更改内容可以查看故障代码说明文档），并对故障参数进行修改。（具体修改方法可以参考e4\_FaultInjectAPITest\_py文件夹中的readme）

本次对于故障的注入，与高级版中的各个故障分别进行注入不同。本次例程需要对多个故障同时进行注入，所以需要将多个故障代码ID和故障参数进行写入。

```
silInt[0:4]=[123540,123450,123451,123451]
silFloat[0:8]=[0,0,0,0,0,0,0,0]

# silInt[0:2]=[123451,123451]
# silFloat[0:4]=[1,2,3,4]

# silInt[0:1]=[123542]
# silFloat[0:2]=[15,0]
```

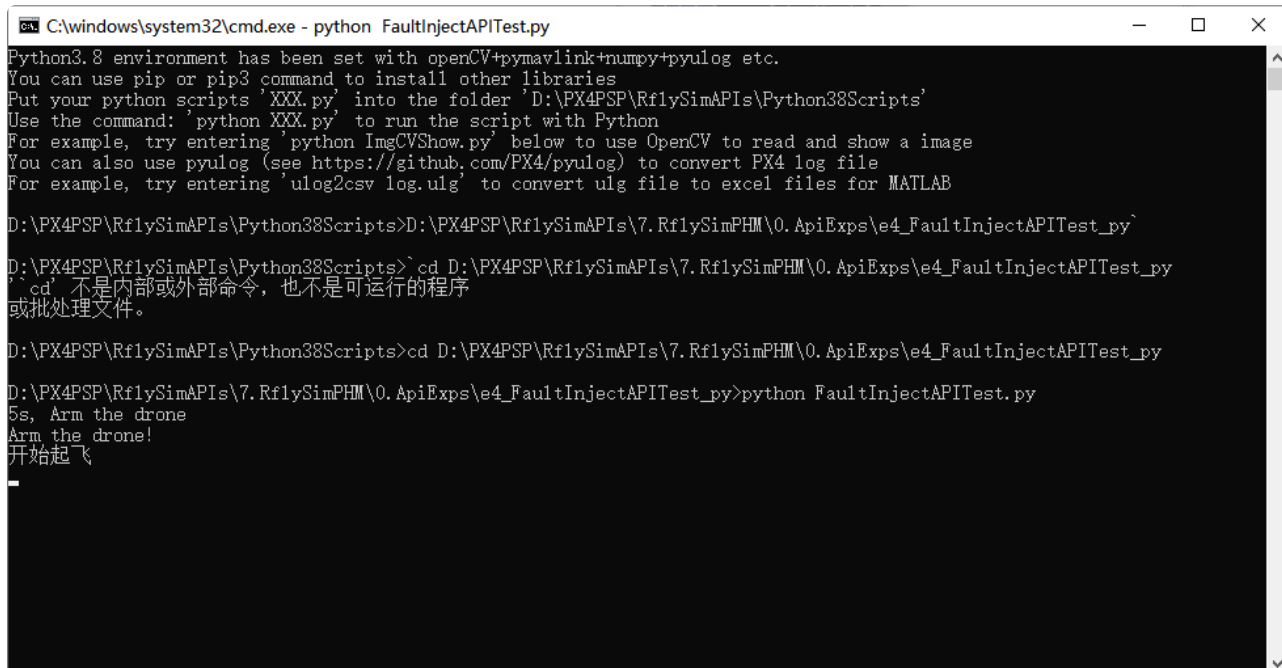
注：本例程可以同时注入多种故障，如上图中同时注入电机执行效率故障以及螺旋桨故障，两种故障都分别有四种参数，所以下方的八位参数都能够发挥作用，可以自己进行设置。

对于只对应一种或是三种参数的故障类型，则需要参考e4\_FaultInjectAPITest\_py文件夹中的readme。

## Step 6: 运行python文件实现故障注入

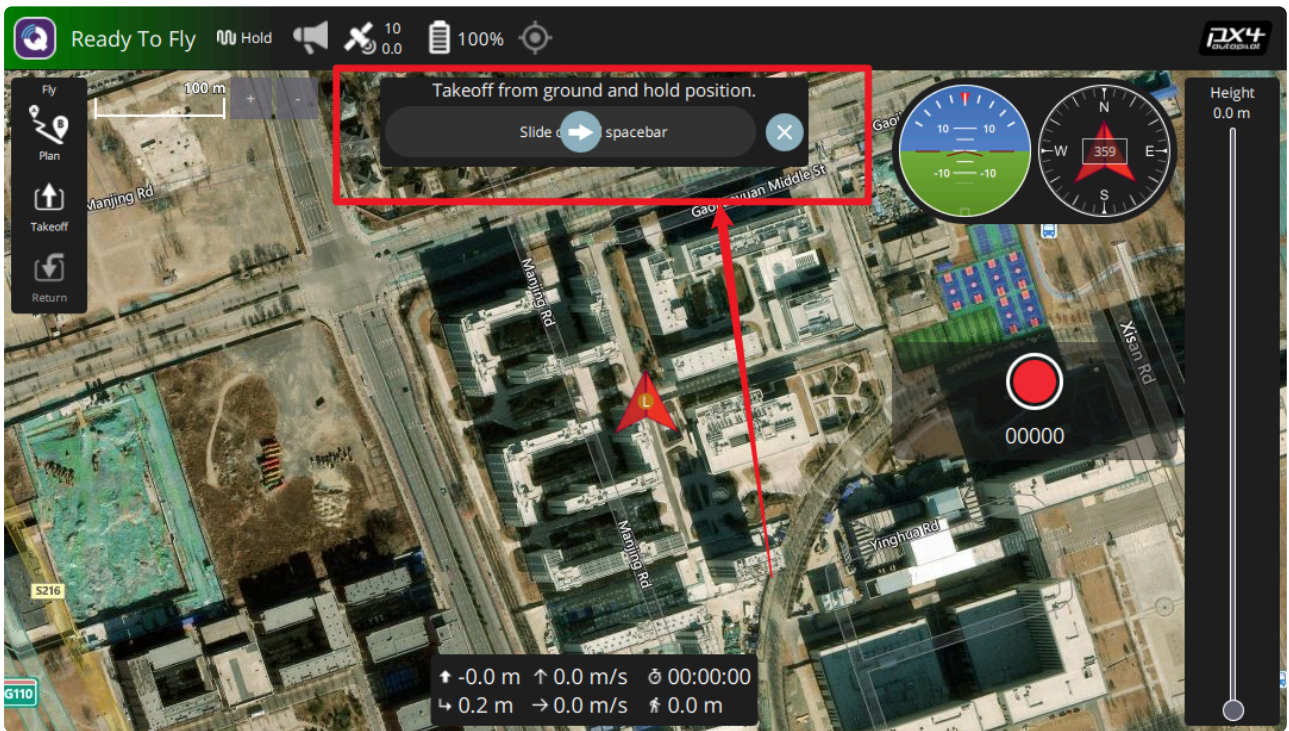
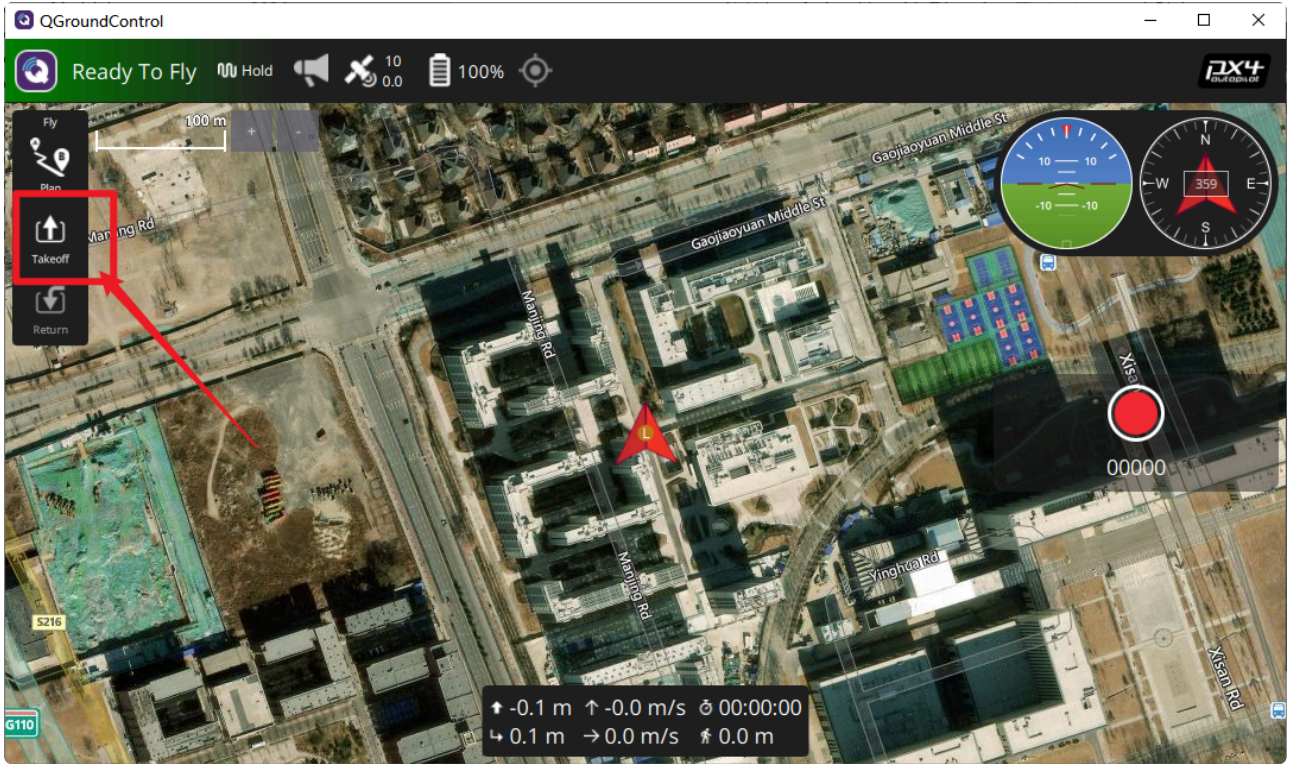
在文件夹下，双击Python38Run.bat，打开集成好的python环境，在该环境下运行 FaultInjectAPITest.py文件，输入 `python FaultInjectAPITest.py`，运行FaultInjectAPITest.py。

```
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.  
You can use pip or pip3 command to install other libraries  
Put Python38Run.bat into your code folder  
Use the command: 'python XXX.py' to run the script with Python  
D:\PX4PSP\RflySimAPIs\7.RflySimPHM\0.ApiExps\e4_FaultInjectAPITest.py>python FaultInjectAPITest.py
```

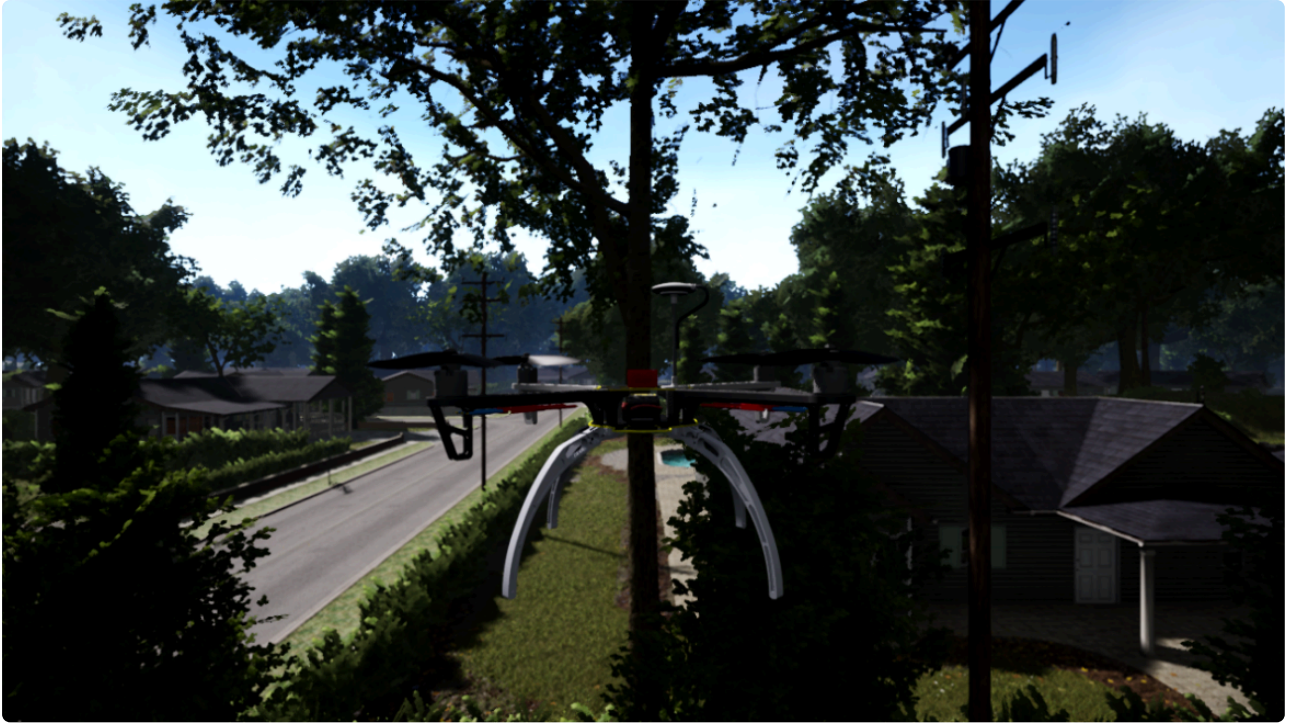


```
C:\windows\system32\cmd.exe - python FaultInjectAPITest.py  
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.  
You can use pip or pip3 command to install other libraries  
Put your python scripts 'XXX.py' into the folder 'D:\PX4PSP\RflySimAPIs\Python38Scripts'  
Use the command: 'python XXX.py' to run the script with Python  
For example, try entering 'python ImgCVShow.py' below to use OpenCV to read and show a image  
You can also use pyulog (see https://github.com/PX4/pyulog) to convert PX4 log file  
For example, try entering 'ulog2csv log.ulg' to convert ulg file to excel files for MATLAB  
D:\PX4PSP\RflySimAPIs\Python38Scripts>D:\PX4PSP\RflySimAPIs\7.RflySimPHM\0.ApiExps\e4_FaultInjectAPITest.py  
D:\PX4PSP\RflySimAPIs\Python38Scripts>`cd D:\PX4PSP\RflySimAPIs\7.RflySimPHM\0.ApiExps\e4_FaultInjectAPITest_py  
'cd' 不是内部或外部命令，也不是可运行的程序  
或批处理文件。  
D:\PX4PSP\RflySimAPIs\Python38Scripts>cd D:\PX4PSP\RflySimAPIs\7.RflySimPHM\0.ApiExps\e4_FaultInjectAPITest_py  
D:\PX4PSP\RflySimAPIs\7.RflySimPHM\0.ApiExps\e4_FaultInjectAPITest_py>python FaultInjectAPITest.py  
5s, Arm the drone  
Arm the drone!  
开始起飞  
-
```

在QGroundControl中点击起飞，拉动横条让无人机起飞。



然后可以看到无人机已经注入故障。



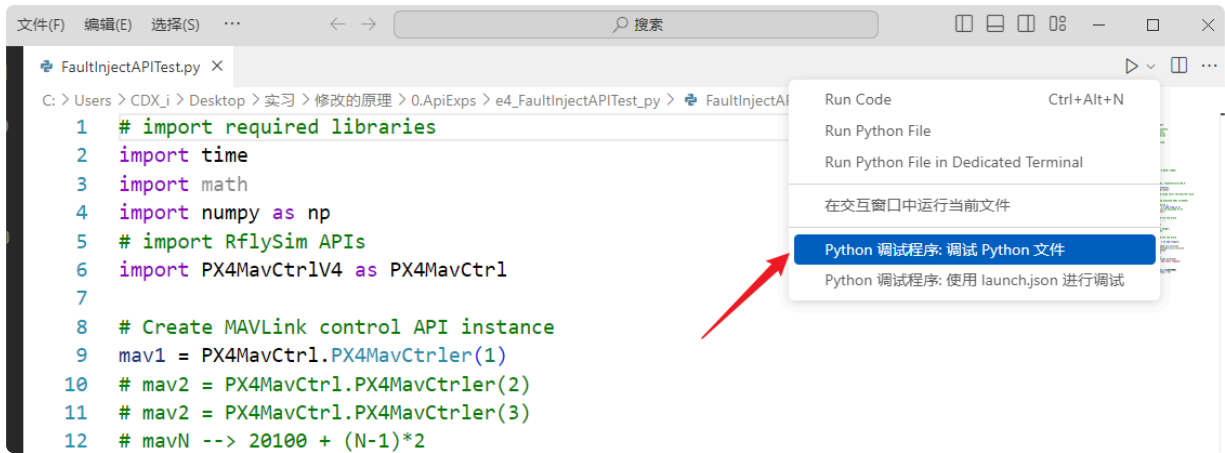
## 5.2. 选作实验（VS Code调试运行）

### 准备工作：

- 先确保已经按 [RflySimAPIs/1.RflySimIntro/2.AdvExps/e3.PythonConfig/Readme.pdf](#) 步骤，正确配置VS Code环境。或者配置了自己的Pycharm等自定义Python环境。
- 其他步骤与上文相同，在运行FaultInjectAPITest.py时，可使用VS Code（或Pycharm等工具）来打开FaultInjectAPITest.py文件，并阅读代码，修改代码，调试执行等。

### 扩展实验：

- 请自行使用VS Code阅读FaultInjectAPITest.py源码，通过程序跳转，了解每条代码的执行原理；再通过调试工具，验证每条指令的执行效果。



```
1 # import required libraries
2 import time
3 import math
4 import numpy as np
5 # import RflySim APIs
6 import PX4MavCtrlV4 as PX4MavCtrl
7
8 # Create MAVLink control API instance
9 mav1 = PX4MavCtrl.PX4MavCtrl(1)
10 # mav2 = PX4MavCtrl.PX4MavCtrl(2)
11 # mav2 = PX4MavCtrl.PX4MavCtrl(3)
12 # mavN --> 20100 + (N-1)*2
```

Run Code Ctrl+Alt+N

Run Python File

Run Python File in Dedicated Terminal

在交互窗口中运行当前文件

**Python 调试程序: 调试 Python 文件**

Python 调试程序: 使用 launchjson 进行调试

## 6. 参考资料

故障ID及对应参数见<../..../RflySimSDK/html/md\_phm\_2md\_2Faultinject.html>

## 7. 常见问题

Q1: \*\*\*

A1: \*\*\*