1.实验名称及目的

1.1实验名称

基于最大模板的GPS模块故障注入。

1.2实验目的

了解最大模板的GPS故障注入模块。

1.3关键知识点

原理：FaultInjectAPITest.py通过PX4MavCtrlV4库的sendInDoubCtrls/sendSILIntDouble/sendSILIntFloat 系列接口，将故障信号inSIL28d（或inSILInts/inSILFloats）在UDP端口中传输到simulink最大模块中的GPS故障注入模块，以此实现GPS故障注入功能。编译simulink最大模块生成DLL文件，将DLL文件导入CopterSim后，成功注入GPS故障并完成仿真。

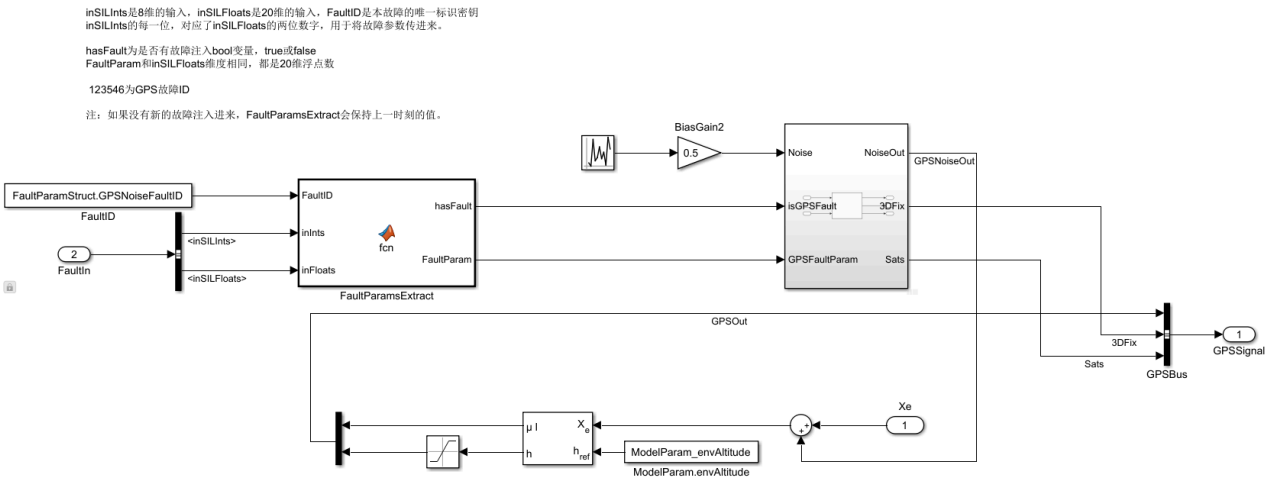
GPS故障在故障信号inSIL28d的识别ID为：123546；其故障输入参数按照排列顺序依次为：噪声增益，3D方式，星数。若未注入GPS故障，噪声增益为属于[-1,1]区间内的随机信号，3D方式默认为3，星数默认为10。

注释： 噪声增益（Noise Gain）： 噪声增益是指在GPS信号上人为添加的噪声强度，用来模拟GPS信号受到干扰或噪声影响的情况。实际应用中信号都是用电磁波传输，其中包含随机白噪声，而增益的值越大，模拟的白噪声强度越高，对GPS信号的准确性影响也越大。 3D方式（3D Mode）：这通常指的是GPS接收器的工作模式，3D方式意味着接收器能够提供三维空间的位置信息，即经度、纬度和高度。其中3D方式的参数是指用于定位的卫星数量。三颗卫星可以确定三维定位信号，两颗卫星可以确定二维定位信号，一颗卫星可以确定一维定位信号。 星数（Satellites Count 或 Satellites Number）：星数是指GPS接收器能够锁定并用于定位的卫星数量。在正常情况下，为了获得准确的定位信息，需要接收到至少4颗GPS卫星的信号。在故障注入的情况下，减少星数可以模拟信号接收不良，比如由于遮挡或干扰导致无法接收到足够数量的卫星信号。

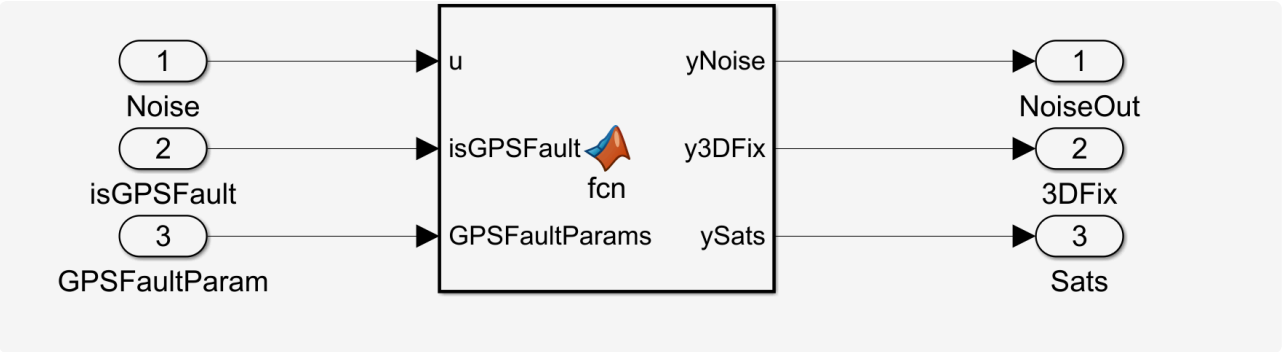
注入故障所用到的simulink模块GPSFault结构图 及其 输入输出参数解释：

该模块用于提取故障参数和模拟真实的GPS定位系统。模块输入FaultIn，即inSIL28d，为包含8维的inSlLInts，20维的inSILFloats的故障参数。输出为带噪声的坐标，噪声增益，3D方式，星数的集合。其中带噪声的坐标是指无人机真实坐标叠加随机噪声，目的是更贴合真实情况。

功能	
	123546为GPS故障ID
输入	
Xe	平面地球框架中的位置，指定为3 × 1向量。
FaultIn	故障输入，包括故障类型和故障参数。
输出	
GPSSignal	将3DFix，GPSOut，Sats封装在总线中



注入故障所用到的simulink模块GPSFault内层模块结构图：



Init.m | FaultParamsExtract | MATLAB Function

```
1   function [yNoise,y3DFix,ySats] = fcn(u,isGPSFault,GPSFaultParams)
2
3 -     yNoise = u;
4 -     y3DFix = 3;
5 -     ySats = 10;
6
7 -     if isGPSFault
8 -         yNoise = (GPSFaultParams(1))*u;
9 -         y3DFix = GPSFaultParams(2);
10 -        ySats = GPSFaultParams(3);
11      end
12
```

功能	通过检查GPS故障标志来调整输入的GPS数据，并返回三个输出参数。
输入	
Noise	均匀分布的随机噪声信号
isGPSFault	指示是否存在GPS故障。如果为 true，表示存在GPS故障。
GPSFaultParam	包含GPS故障参数的向量。 GPSFaultParams(1) 用于调整噪声。 GPSFaultParams(2) 设置3D固定值。 GPSFaultParams(3) 设置卫星数量。
输出	
NoiseOut	调整后的GPS数据。默认情况下等于输入数据 Noise。如果存在GPS故障，则 yNoise 为 GPSFaultParams(1) 乘以 Noise。
3DFix	3D固定值。默认情况下为3。如果存在GPS故障，则 3DFix 为 GPSFaultParams(2)。
Sats	：卫星数量。默认情况下为10。如果存在GPS故障，则 Sats 为 GPSFaultParams(3)。

主要要求掌握：

GPS故障注入模块的功能及其实现。

说明文档	说明文档链接以及地址
simulink中GPSFault模块	PX4PSP/RflySimSDK/html/md_phm_2md_2HILGPSModle.html
FaultInjectAPITest.py	PX4PSP\RflySimAPIs\7.RflySimPHM\0.ApiExps\e4_FaultInjectAPITest_py\readme.pdf

说明文档	说明文档链接以及地址
PX4MavCtrlV4库	PX4PSP/RflySimSDK/html/PX4MavCtrlV4_8py.html

2.实验效果

3.文件目录

例程目录：[安装目录]\RflySimAPIs\7.RflySimPHM\1.BasicExps\e2_GPSFault

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。
Init.m	动力学模型相关参数。
Python38Run.bat	Python程序执行脚本。
FaultInjectAPITest.py	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。

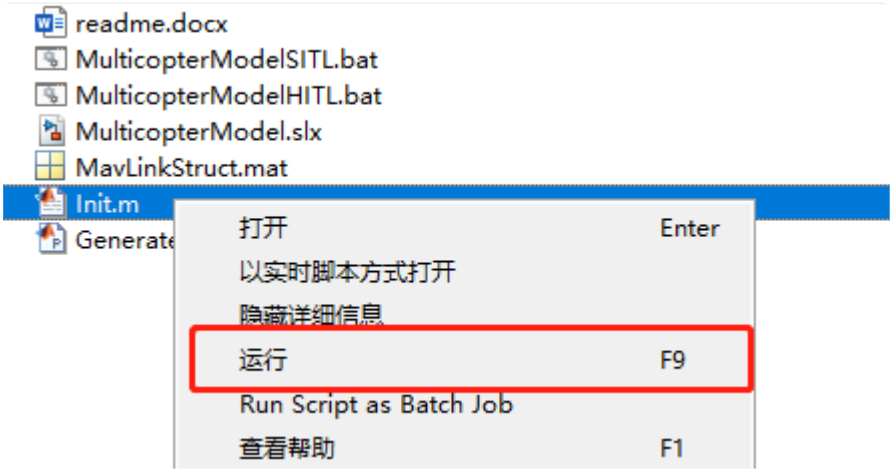
> ①：推荐配置请见： https://rflysim.com/

5.实验步骤

5.1. 必做实验

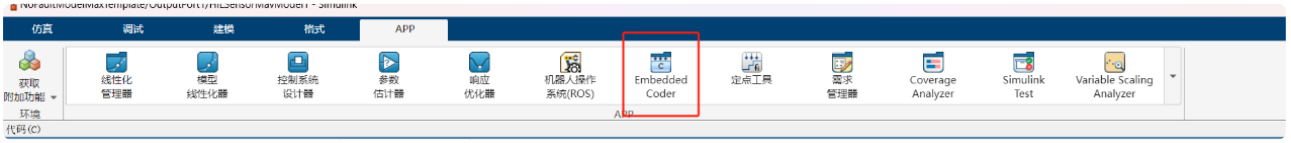
Step 1：初始化数据

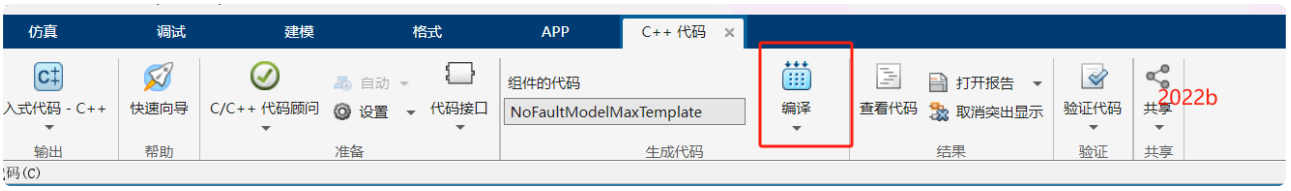
打开“Init.m” 文件并运行。



Step 2：编译文件

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

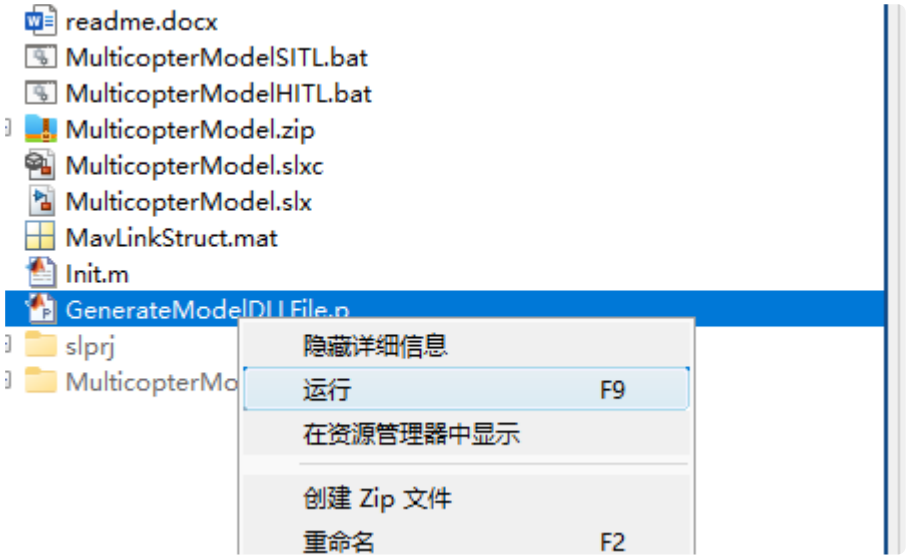




注：如果故障模块版本错误，无法编译，需要从故障模块库中选择对应的模块进行替换。

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：运行软件在环仿真

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，选择打开当前文件夹。

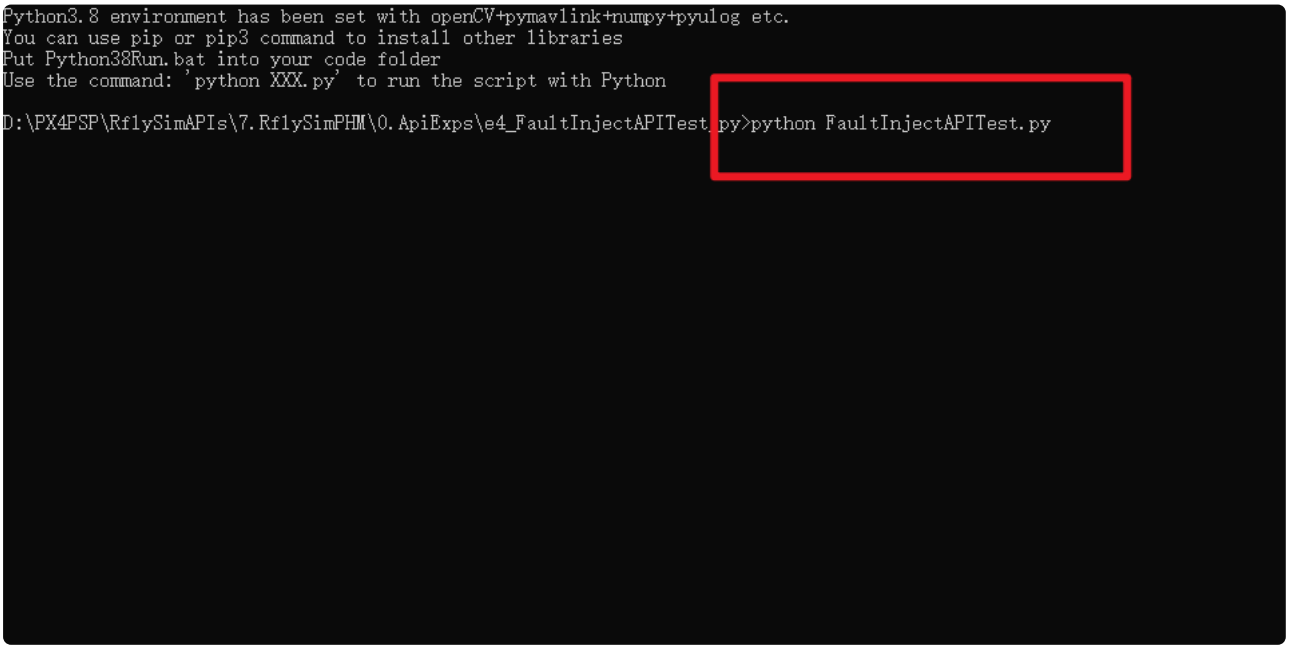
对FaultInjectAPITest.py其中的故障注入代码按照RflySimAPIs/7.RflySimPHM/0.ApiExps/e4_FaultInjectAPITest_py中的FaultInjectAPITest_py中的故障注入代码更改为GPS故障，并对故障参数进行修改。如下图所示GPS故障的三个参数分别为50，0，0（具体修改方法可以参考7.RflySimPHM/0.ApiExps/e4_FaultInjectAPITest_py文件夹中的readme）

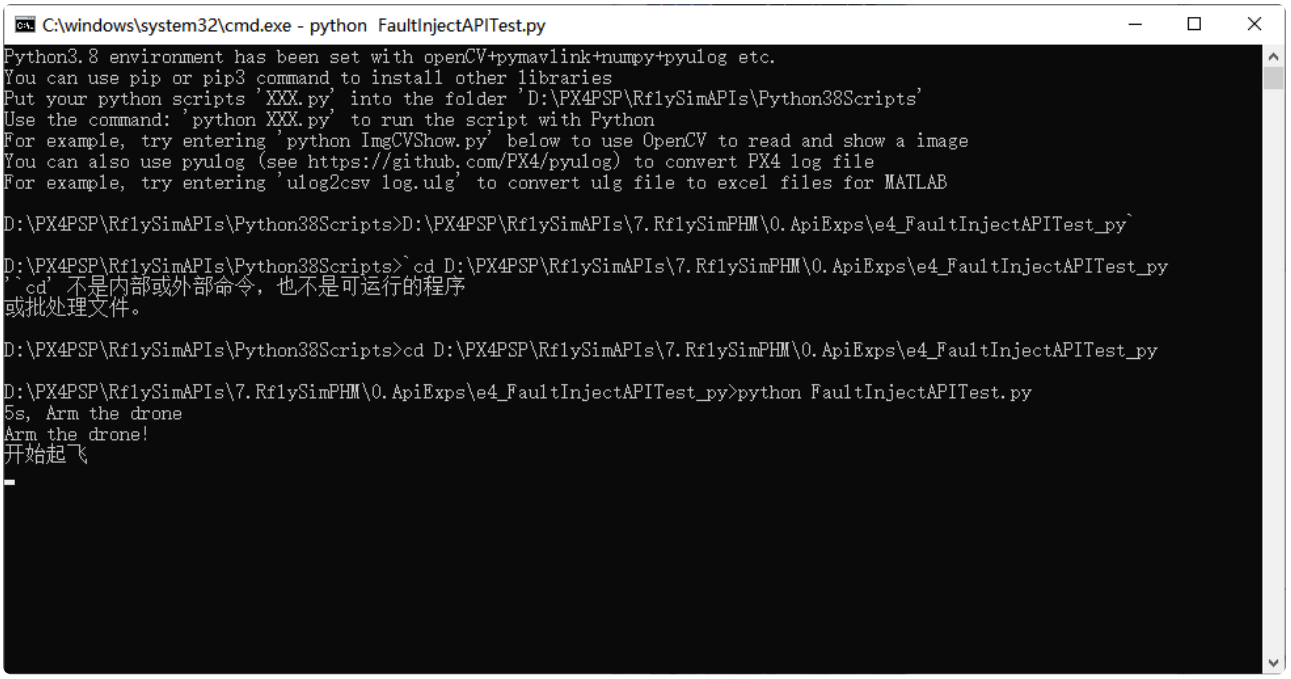
```
58
59      if time.time() - startTime > 20 and flag==1:
60          #np.zeros()
61          silInt=np.zeros(8).astype(int).tolist()
62          silFloat=np.zeros(20).astype(float).tolist()
63          silInt[0:2]=[123546,123546]
64          silFloat[0:4]=[50,0,0,0]
65          # silInt[0:1]=[123542]
66          # silFloat[0:2]=[15,0]
67          mav1.sendSILIntFloat(silInt,silFloat)
68          print('Inject a fault, and start loging')
69          # 调用 sendRflyShowTextTime 发送命令
70          ue.sendRflyShowTextTime("Inject_a_fault", 20)
71          flag=2
```

注：文件中的silFloat已经修改好，可以直接运行注入例程对应故障。

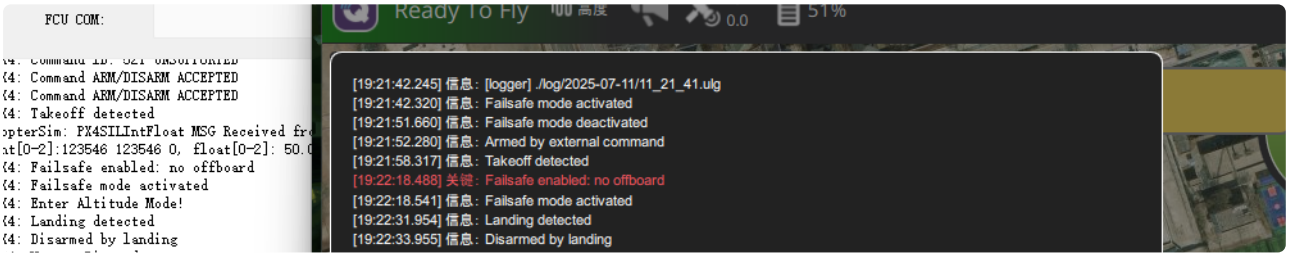
Step 6：运行python文件实现故障注入

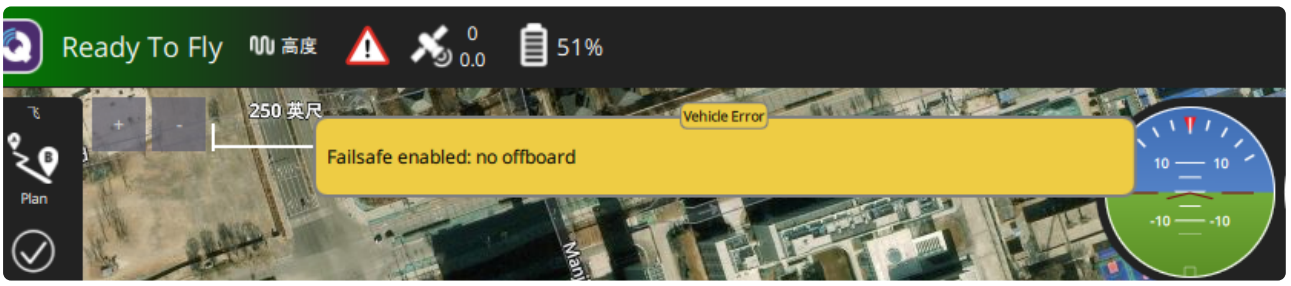
在文件夹下，双击Python38Run.bat，打开集成好的python环境，在该环境下运行FaultInjectAPITest.py文件，输入
python FaultInjectAPITest.py，运行FaultInjectAPITest.py。

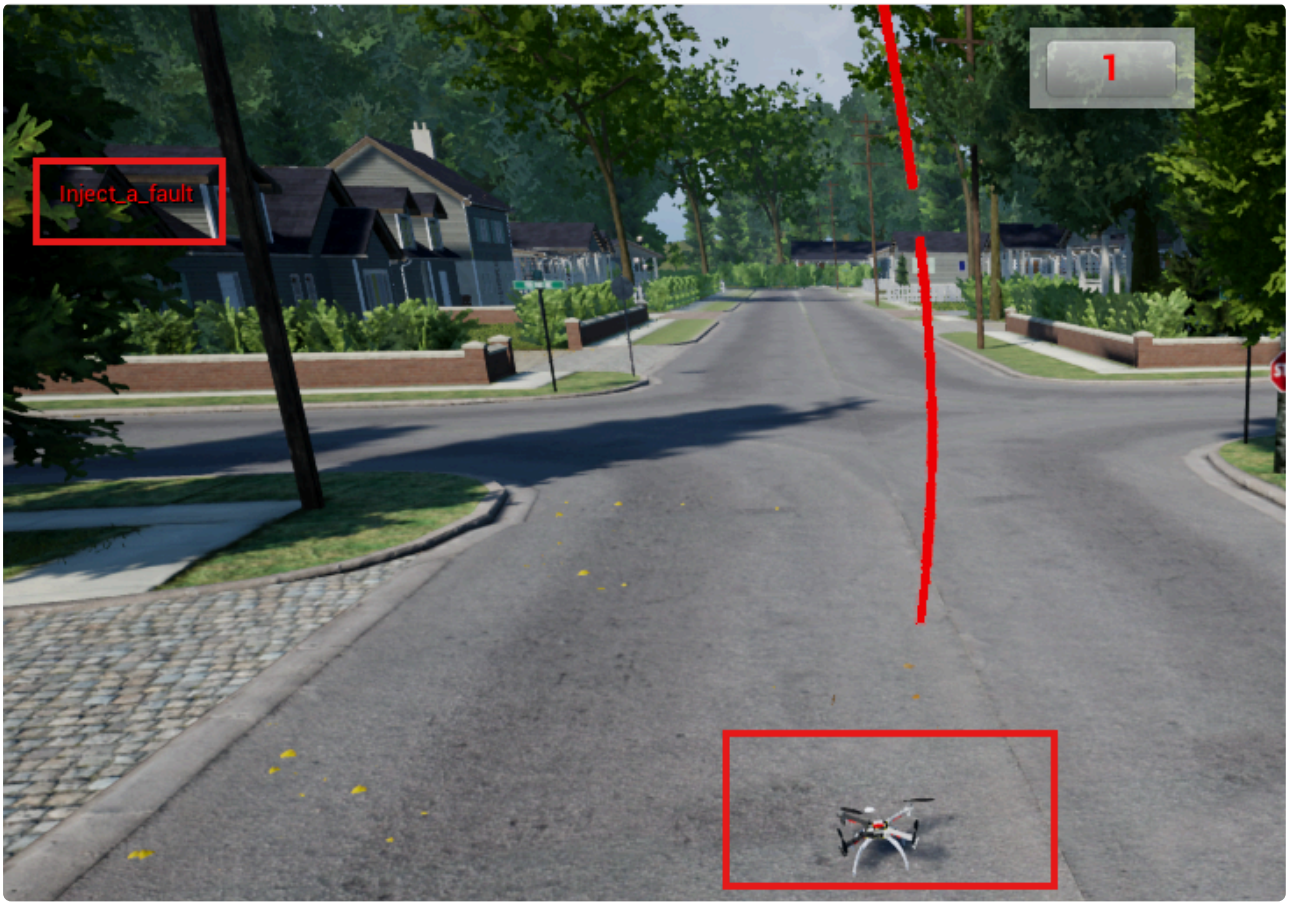




然后可以看到无人机已经注入GPS故障，飞机失去GPS信号后自动降落







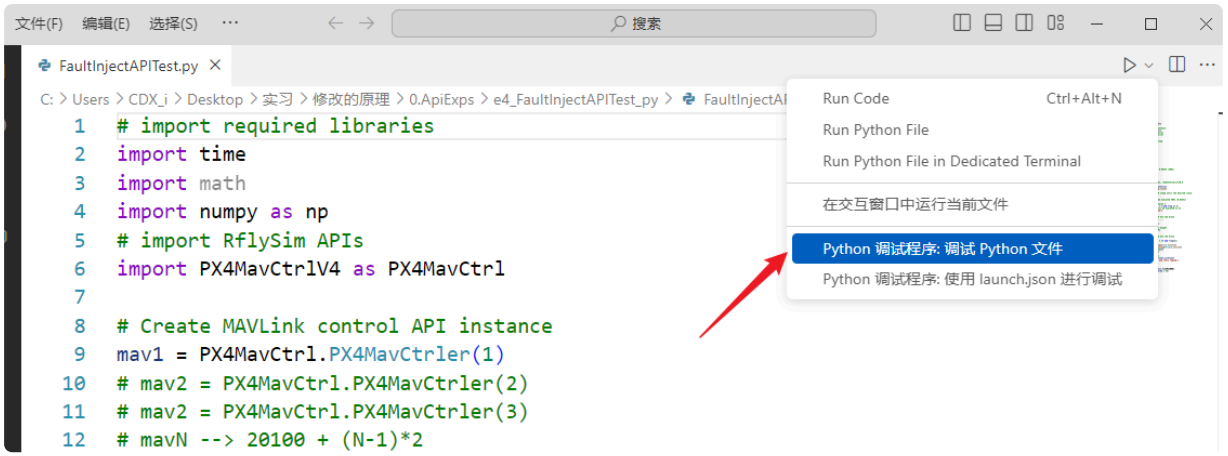
5.2. 选作实验（VS Code调试运行）

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。



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：***