
1. 实验名称及目的

1.1. 实验名称

python 外部控制库 PX4MavCtrl 验证实验

1.2. 实验目的

熟悉无人机 offboard 模式控制、状态数据获取和 RflySim3D 的控制接口，了解 SITL 通信框架

1.3. 关键知识点

首先打开 MAVLink 以监控 CopterSim 数据并实时更新。然后发送指令让飞控中初始化为 Offboard 模式，并在 Python 中开始发送数据循环。然后首先要一直发送控制指令给飞控，然后才能让飞控解锁进行下面的相应控制，最后，发送指令让飞控退出 Offboard 模式，并且停止监听 MAVLink 数据。关键代码解析如下：

关键知识点 1：飞机控制指令

```
mav = PX4MavCtrl.PX4MavCtrl(1) # 创建飞机控制实例
mav.InitMavLoop() # 初始化 Mavlink 监听程序，读取飞机数据
mav.initOffboard() # 进入 Offboard 模式
mav.SendPosNED(0, 0, -1.7, 0) # 发送 1.7 米高的位置控制指令
mav.SendMavArm(True) # 解锁飞控
mav.SendVelNED(0, 0, 0.2, 0) # 发送速度指令
mav.endOffboard()# 结束 Offboard 模式
mav.stopRun()# 停止仿真
```

关键知识点 2：UE 控制

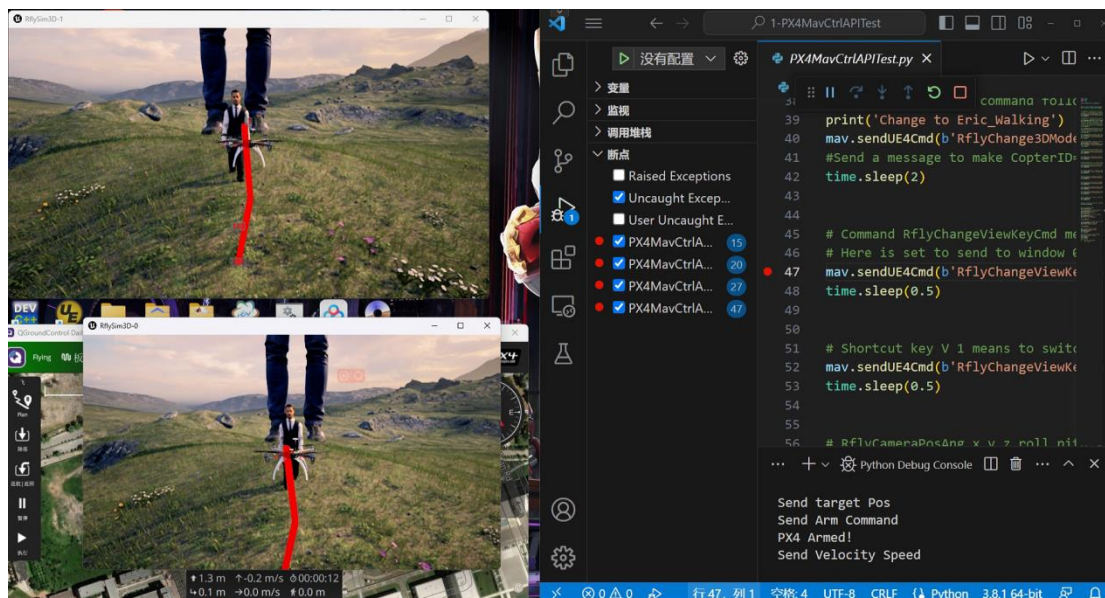
接口详细使用方法见：[UE4CtrlAPI.py](#)

```
ue = UE4CtrlAPI.UE4CtrlAPI() # 创建 UE 控制实例
ue.sendUE4Cmd('RflyChangeMapbyName Grasslands') # 切换指定地图
ue.sendUE4Pos(100,30,0,[2.5,0,-8.086],[0,0,math.pi])# 生成一个 3D 对象。
对象的 ID 为 100，类型为 30（代表一个人物），转子速度为 0，位置坐标为 [2.5,
0, -8.086]
ue.sendUE4PosScale(101,2030,0,[10.5,0,-
8.086],[0,0,math.pi],[10,10,10])# 生成另一个 3D 对象，其 ID 为 101，类型
为 2030（代表一种人的模型）。位置为[10.5,0,-8.086]米，欧拉角度为[0,0,
math.pi] 弧度，尺寸比例为[10,10,10]
ue.sendUE4Cmd('RflyChange3DModel 100 12') #
ue.sendUE4Cmd('RflyChange3DModel 100 Eric_Walking') # 改变 ID 为 100
的对象模型样式 Eric_Walking，代表一个行走的人
ue.sendUE4Cmd('RflyChangeViewKeyCmd B 1',0)# 模拟按下快捷键 B1，切换视
角到 ID 为 1 的对象
```

```
ue.sendUE4Cmd('RflyChangeViewKeyCmd V 1',0)# 模拟按下快捷键 V1, 切换到第一个机载摄像头
ue.sendUE4Cmd('RflyCameraPosAng 0.1 0 0',0)# 设置前置摄像头位置
```

2. 实验效果

python 程序发送一系列指令，在 RflySim3D 程序中新建了一个走动的人的目标，设置了视角形式、尺寸、位置，向仿真的无人机发送控制指令使其起飞与降落。



3. 文件目录

例程目录：[\[安装目录\]\RflySimAPIs\6.RflySimExtCtrl\0.ApiExps\e1_PX4MavCtrlAPITest](#)

文件夹/文件名称	说明
PX4MavCtrlAPITest.bat	启动仿真配置文件
PX4MavCtrlAPITest.py	实现功能主文件
Python38Run.bat	Python 环境启动脚本

4. 运行环境

序号	软件要求	硬件要求	
		名称	数量(个)
1	Windows 10 及以上版本	笔记本/台式电脑 ^①	1
2	RflySim 工具链		
3	Visual Studio Code		

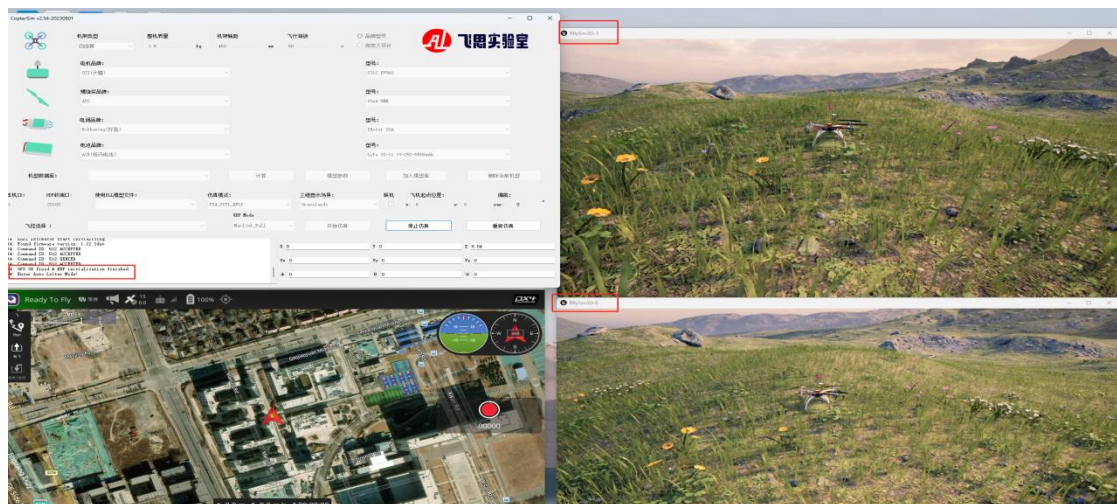
①：推荐配置请见：<https://rflysim.com/doc/zh/HowToInstall.pdf>

5. 实验步骤

5.1. 必做实验:

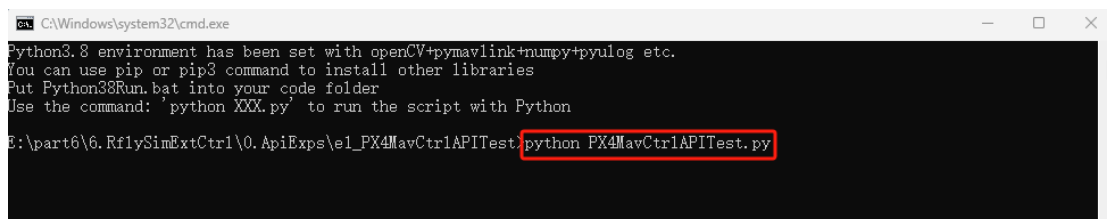
Step 1: 开启仿真

启动 PX4MavCtrlAPITes.bat 脚本将会启动 1 个 QGC 地面站, 1 个 CopterSim 软件且其软件下侧日志栏必须打印出 GPS 3D fixed & EKF initialization finished 字样代表初始化完成, 并且有 2 个 RflySim3D 软件各有 1 架无人机。如下图所示:



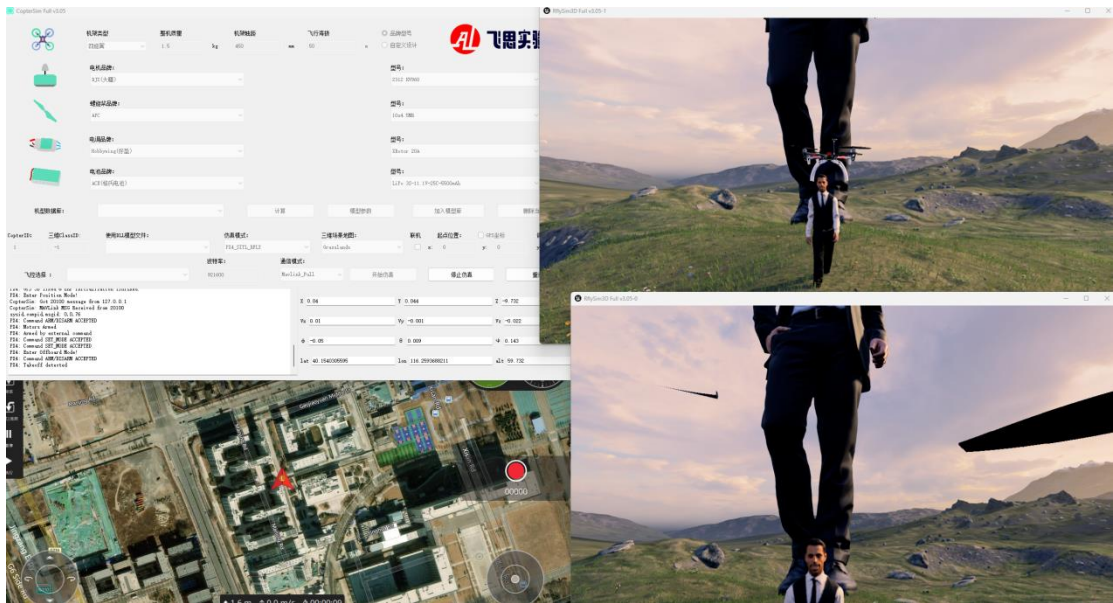
Step 2: 运行控制程序

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 PX4MavCtrlAPITest.py 文件, 输入 python PX4MavCtrlAPITest.py



Step 3: 观察结果

会打开两个 RflySim3D 窗口, 一个是前置摄像头, 一个是上帝视角观测, 可以生产两个人物模型, 看到飞机成功起飞。注意: 在 RflySim3D 窗口按 T 键开启或关闭飞机轨迹记录功能, T+数字*开启/更改轨迹粗细为*号。



Step 4: 结束仿真

在“PX4MavCtrlAPITest.bat”脚本开启的命令提示符 CMD 窗口中，按下回车键（任意键）就能快速关闭 CopterSim、QGC、RflySim3D 等所有程序。

5.2. 选作实验（VS Code 调试运行）

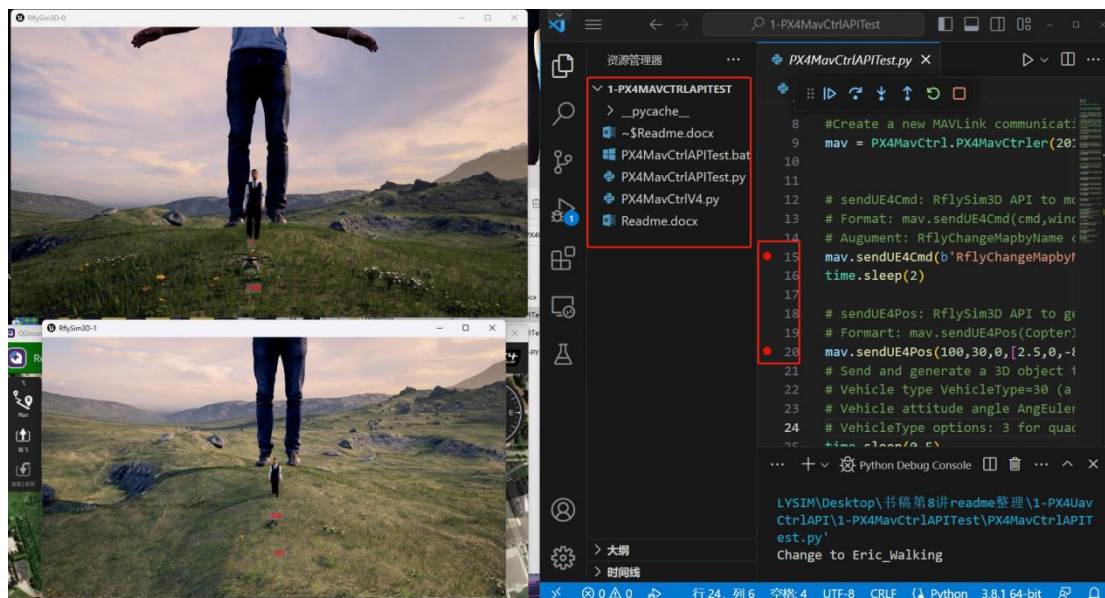
准备工作：

- 先确保已经按 [RflySimAPIs\1.RflySimIntro\2.AdvExps\3_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 Step2 运行 PX4MavCtrlAPITest.py 时，可使用 VS Code（或 Pycharm 等工具）来打开 PX4MavCtrlAPITest.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 PX4MavCtrlAPITest.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。如下图在每条关键语句前面点上断点（红点）。

Step 1:



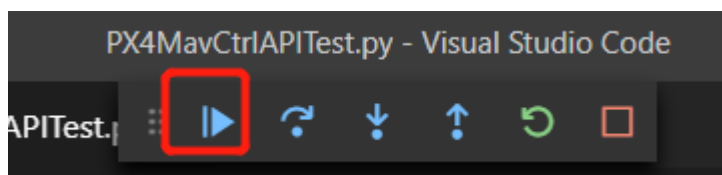
Step 2:

按下图所示开启调试模式。



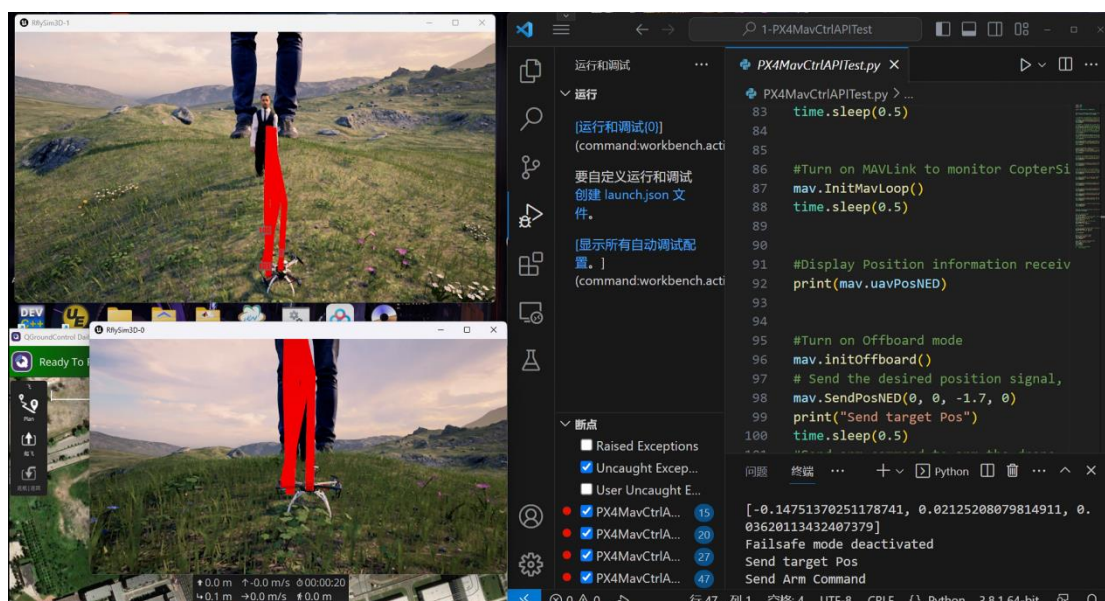
Step 3:

点击下图箭头按钮，依次执行语句。



Step 4:

如下图所示，本例子会打开两个 RflySim3D 窗口，一个是前置摄像头，一个是上帝视角观测。并且按 T 键开启或关闭飞机轨迹记录功能，T+数字*开启/更改轨迹粗细为*号。然后在 VScode 终端上就会出现无人机的仿真状态数据。



Step 5:

在下图“PX4MavCtrlAPITest.bat”脚本开启的命令提示符 CMD 窗口中，按下回车键（任意键）就能快速关闭 CopterSim、QGC、RflySim3D 等所有程序。

```

C:\WINDOWS\system32\cmd.exe

-----
Start QGroundControl
Kill all CopterSims
Starting PX4 Build
[1/1] Generating ../logs
killing running instances
starting instance 1 in /mnt/c/PX4PSPFull/Firmware/build/px4_sitl_default/instance_1
PX4 instances start finished
Press any key to exit
  
```

按下回车键，快速关闭所有仿真窗口

Step 6:

在下图 VSCode 中，点击“终止终端”，可以彻底退出脚本运行。



6. 参考资料

[1]. 无

7. 常见问题

Q1: ***

A1: ***