

1、实验名称及目的

Mavros 版 PX4MavCtrlV4ROS.py 接口与 RflySim3D 和 Pixhawk 通信实验：通过平台 mavros 版本的 API 接口进行 RflySim3D 和 Pixhawk 的通信。

2、实验原理

首先通过 PX4MavCtrlV4 实例化一个控制接口，在本例程中，会使用到两种连接方式连接飞控与主机，由于连接方式与主机的系统的原因对于创建的控制接口有如下情况：

当为 Windows 主机与飞控之间的连接为 USB 连接时，接口函数应为 PX4MavCtrlr(1,'127.0.0.1','COM3',57600)，其中 57600 为数据传输的波特率；

当为 Windows 主机与飞控之间的连接方式为串口连接时，接口函数应为 PX4MavCtrlr(1,'127.0.0.1','COM4',57600)，其中 57600 为数据传输的波特率；

当为 Linux 主机与飞控之间的连接为 USB 连接时，接口函数应为 PX4MavCtrlr(1,'127.0.0.1','/dev/ttyS0',57600)，其中 57600 为数据传输的波特率；

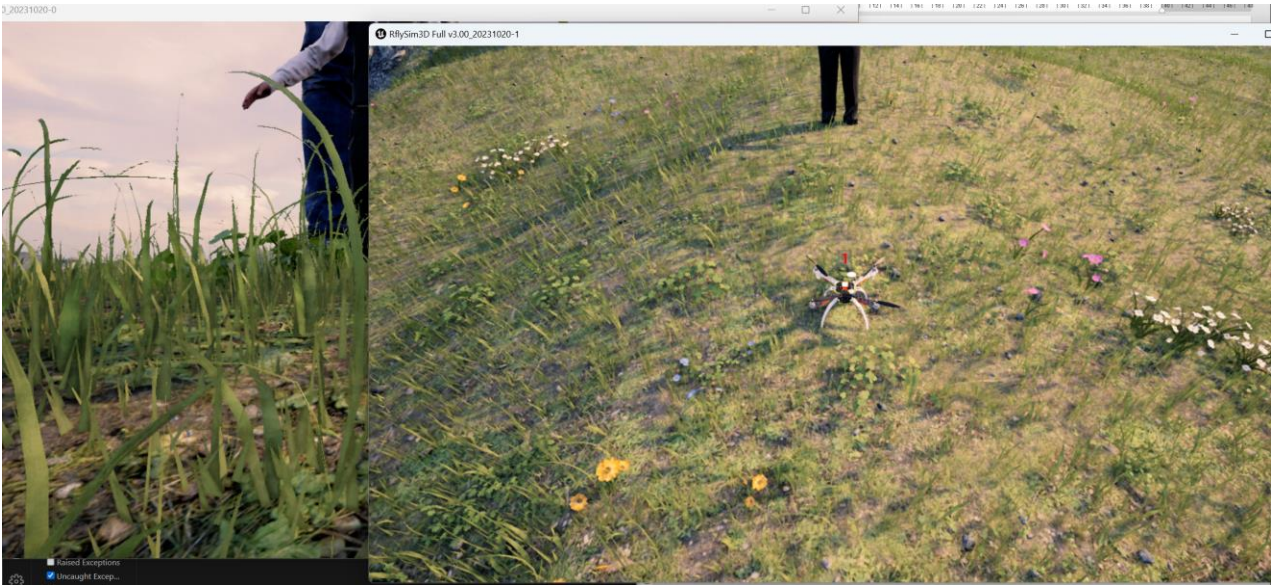
当为 Linux 主机与飞控之间的连接为串口连接时，接口函数应为PX4MavCtrlr(1,'127.0.0.1','/dev/ttyAMA0',57600)，其中 57600 为数据传输的波特率；

若不连接飞控，可直接使用 IP 地址建立通信

通过上述控制接口的使用方法正确建立主机与飞控之间的通信，即可对无人机进行仿真控制。

3、实验效果

运行会生成两个人，在两个仿真界面中所处视角不同



4、文件目录

文件夹/文件名称	说明
----------	----

client_ue4_SITL.bat	启动仿真配置文件
server_ue4.py	Python 实验脚本
server_ue4_serial.py	串口连接实验脚本
server_ue4ROS.py	ROS 下的 python 实验脚本
Server_ue4ROS_Serial	ROS 下使用串口连接的 python 实验脚本

5、运行环境

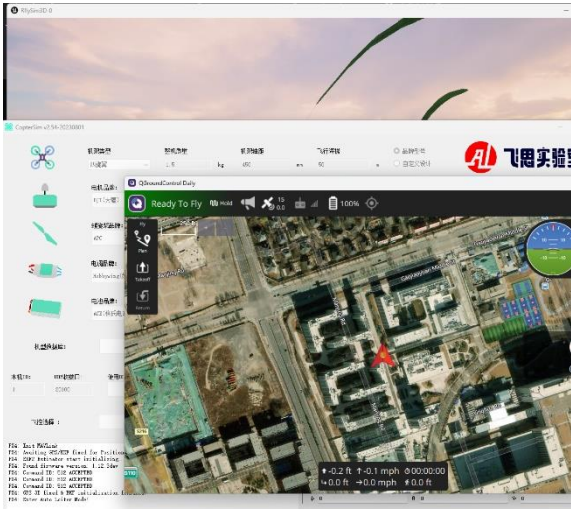
序号	软件要求	硬件要求	
		名称	数量(个)
1	Windows 10 及以上版本	笔记本/台式电脑 ^①	1
2	RflySim 平台高级版及以上		
3	Visual Studio Code		

- 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：droneyee_zyfc-h7_default，固件版本为：1.12.1。其他配套飞控请见：<http://doc.rflysim.com/hardware.html>

6、实验步骤

Step 1:

运行 client_ue4_SITL.bat 来启动 SITL 模拟。



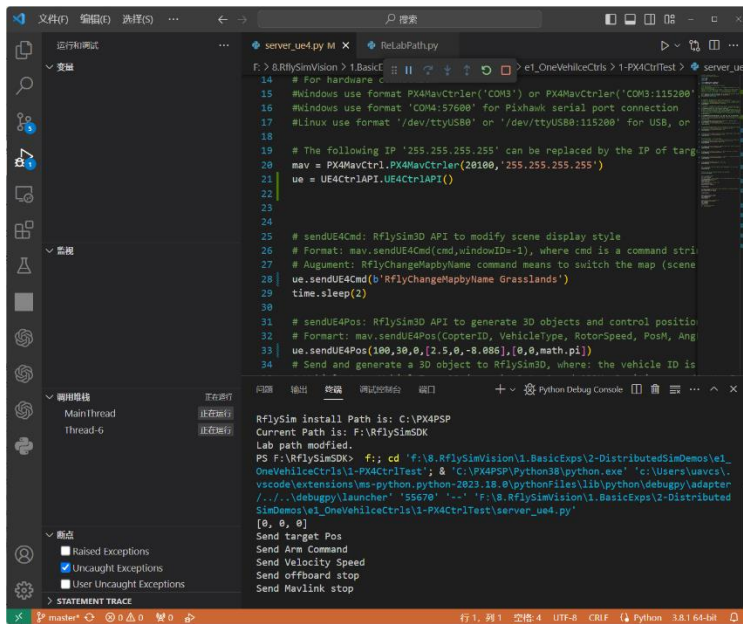
Step 2:

运行 PX4P\SPRfySimAPIs\RflySimSDK 目录下的 ReLabPath.py 文件。

Step 3:

使用 VS Code 打开 server_ue4.py 并在此计算机上运行它，以与 Pixhawk 和 UE4 进行通

信。



Step 4:

您可以将此文件夹中的所有文件复制到另一台计算机（树莓派、TX2 或任何具有 Linux 和 ROS 环境的计算机），使用命令“python3 server_ue4.py”通过 UDP 从前一台计算机与 Pixhawk 和 UE4 进行通信。

server_ue4_ROS.py 是“server_ue4.py”的 ROS 版本。它们之间唯一的区别是句子“import PX4MavCtrlV4 as PX4MavCtrl”和“import PX4MavCtrlV4ROS as PX4MavCtrl”，其中 PX4MavCtrlV4.py 是通过 pymavlink 进行的 Mavlink API，而 PX4MavCtrlV4ROS.py 是通过 mavros 进行的 Mavlink API。

PX4MavCtrlV4.py 是通过 Mavlink 与 Pixhawk 进行通信的 API（并通过 UDP 与 UE4 进行通信），而 PX4MavCtrlV4ROS.py 是 mavros 版本。

7、参考文献

[1]. 无

8、常见问题

Q1: 仿真启动后一直卡在初始地图且无飞机出现。

A1: 仿真 bat 脚本使用了广播通信方式，在 CopterSim 中等待如下语句出现后，切换到仿真界面按下按键 i