1.实验名称及目的

1.1实验名称

自定义控制序列配制与使用。

1.2实验目的

掌握自定义控制序列的基本使用与配置。

1.3关键知识点

这个Python脚本是一个用于控制和测试无人机（或其他飞行平台）的自动化系统，它通过发送预定义的控制命令序列来管理和记录无人机的行为。文件包含一个db.json文件和CmdCtrl.py文件。其中CmdCtrl.py会解析db.json文件的控制序列并映射为载具的控制指令。

1. 创建无人机控制API实例：PX4MavCtrl.PX4MavCtrler(1)：创建无人机控制器实例，编号为1，用于后续的命令发送和数据接收。
2. 初始化MAVLink数据接收循环：mav1.InitMavLoop()：启动无人机的数据接收循环，这是接收无人机状态信息的第一步。
3. 初始化真实数据循环：mav1.InitTrueDataLoop()：启动接收无人机真实飞行数据的循环，用于获取无人机的实际飞行状态。
4. 初始化非板载控制模式：mav1.initOffboard()：将无人机设置为非板载控制模式，允许外部系统发送控制指令。
5. 设置时间间隔和循环标志：timeInterval：设置为30Hz，即每0.0333秒执行一次循环体内的代码，用于控制主循环的执行频率。
6. 主循环：while
   True：一个无限循环，用于持续发送命令和接收数据，直到满足结束条件。
7. 执行起飞任务：在仿真时间超过5秒后，通过发送特定命令，触发无人机解锁并起飞至10米高度。
8. 注入故障并开始记录：当仿真时间超过20秒后，发送一组特定的数据（整型和浮点型）模拟故障注入，并开始记录相关数据。

9. 打印无人机位置信息：如果已经注入故障（通过flag标志判断），则打印出无人机的仿真位置和真实位置，用于监控无人机状态。
10. 结束条件：当仿真时间超过50秒后，满足退出条件，结束主循环。
11. 清理和结束：mav1.endMavLoop() 和 mav1.EndTrueDataLoop()：结束数据接收循环，并进行必要的清理工作。
    主要要求掌握：
    程序控制无人机方法：
12. 创建无人机控制API实例:使用PX4MavCtrl.PX4MavCtrler(1)创建无人机控制器的实例，这是与无人机进行通信的接口。
13. 初始化MAVLink数据接收循环:调用mav.InitMavLoop()来初始化无人机的数据接收循环，这是持续接收无人机状态信息的基础。
14. 初始化非板载控制模式:使用mav.initOffboard()将无人机设置为非板载控制模式，这是执行外部控制命令的前提。
15. 发送控制命令:通过TRIGGERMAVCMD函数来处理和发送具体的控制命令。这个函数解析控制命令字符串，并将命令发送给无人机。

2.实验效果

运行CmdCtrl.py,可以看见程序正确解析了控制序列并完成了控制。

3.文件目录

例程目录：[安装目录]\RflySimAPIs\7.RflySimPHM\0.ApiExps\e15_CmdExp

文件夹/文件名称	说明
MulticopterModelHITL.bat	硬件在环仿真脚本
MulticopterModelSITL.bat	软件在环仿真脚本
CmdCtrl.py	解析db.json文件的控制序列并映射为载具的控制指令
db.json	数据库文件

4.运行环境

4.1 软件要求

Windows 10及以上版本；RflySim工具链；Visual Studio Code。

> 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。

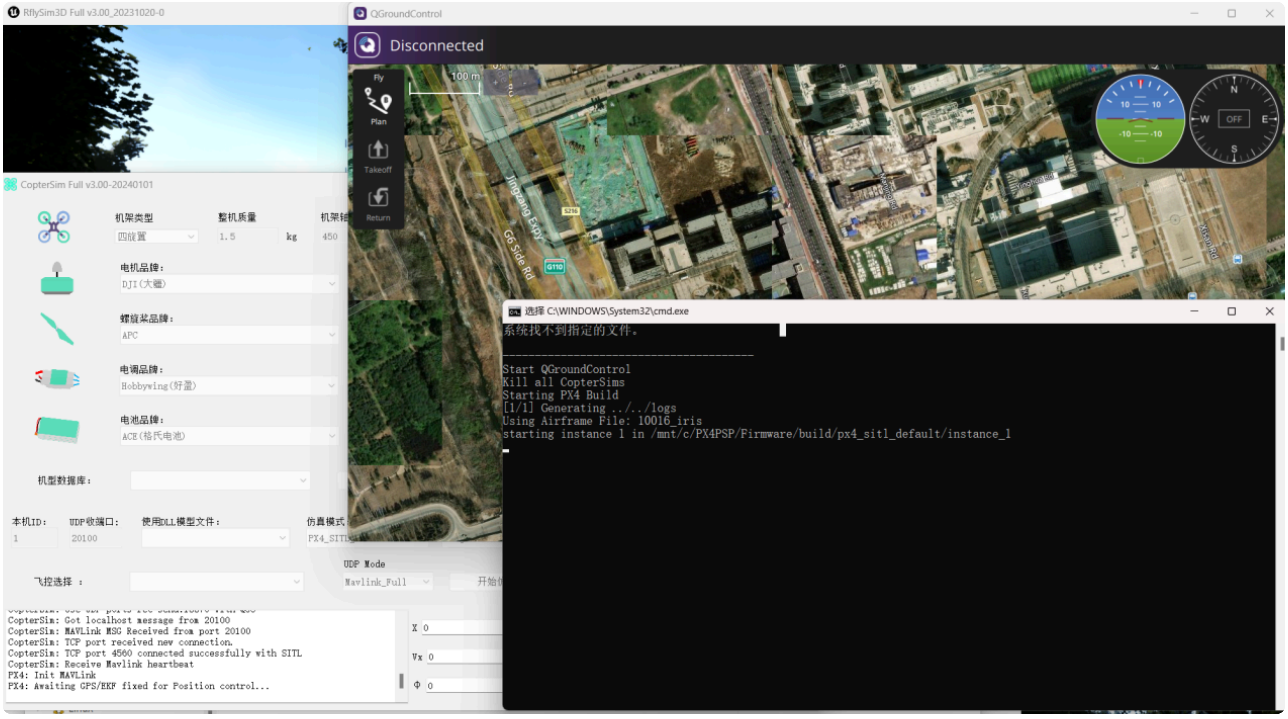
> ①：推荐配置请见：https://rflysim.com/

5.实验步骤

5.1. 必做实验

Step 1：运行.bat文件

管理员方式打开MulticopterModelSITL.bat脚本，等待所有软件的连接。



Step 2：运行python文件

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CmdCtrl.py文件，输入
python CmdCtrl.py，运行CmdCtrl.py。

可以看见程序正确解析了控制序列并完成了控制。

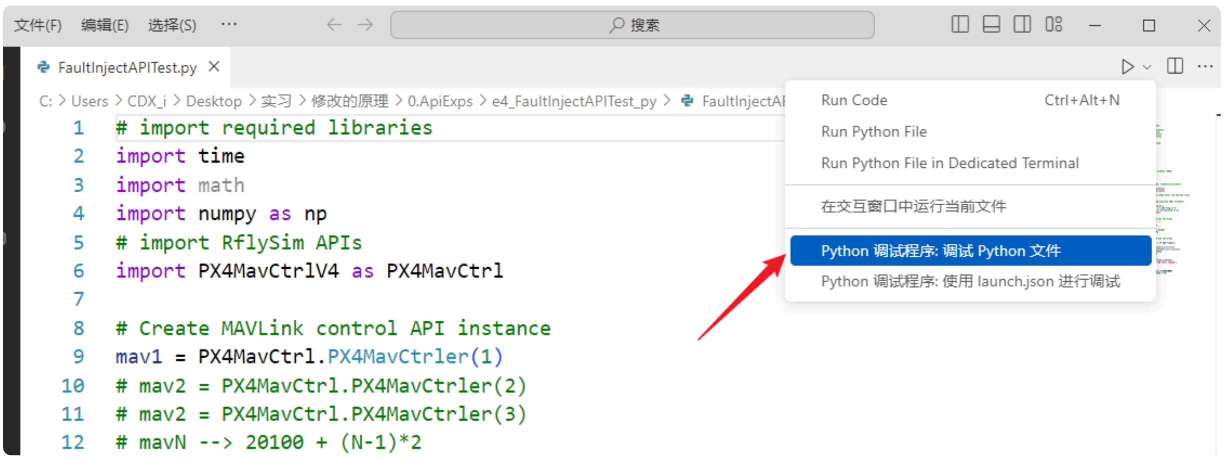
5.2. 选作实验（VS Code调试运行）

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CmdCtrl.py时，可使用VS Code（或Pycharm等工具）来打开CmdCtrl.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CmdCtrl.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

1. 无

7.常见问题

Q1：无

A1：无