

RflySim平台基础功能demo

1. 简介

使用平台免费版即可使用RGB相机，深度相机，灰度相机，以及激光雷达（包括传统机械式扫描雷达，以及固态雷达）。

仿真可以使用视觉定位方案，也可以使用激光雷达定位方案。避障可以用激光雷达，也可以用深度相机等。

Demo

仅展示配置mid360激光雷达，深度相机，前视以及下视单目RGB，如果需要其他传感器，参考[\[RflySim安装目录\]/RflySimAPIs/8.RflySimVision/API.pdf](#)

7.3部分内容，如果配置需要的传感器，免费版仅支持3个传感器。Demo

程序分为两部分，一部分在windows上运行，另一部在ubuntu上运行（默认系统里包含了ROS环境）。

2. Windows 系统上的程序

3. 配置SITLPosStr.bat脚本：

更改内容“SET IS_BROADCAST= 192.168.3.18”，这里IP为自己Ubuntu里IP，当然，前提是这两个IP能够ping通；

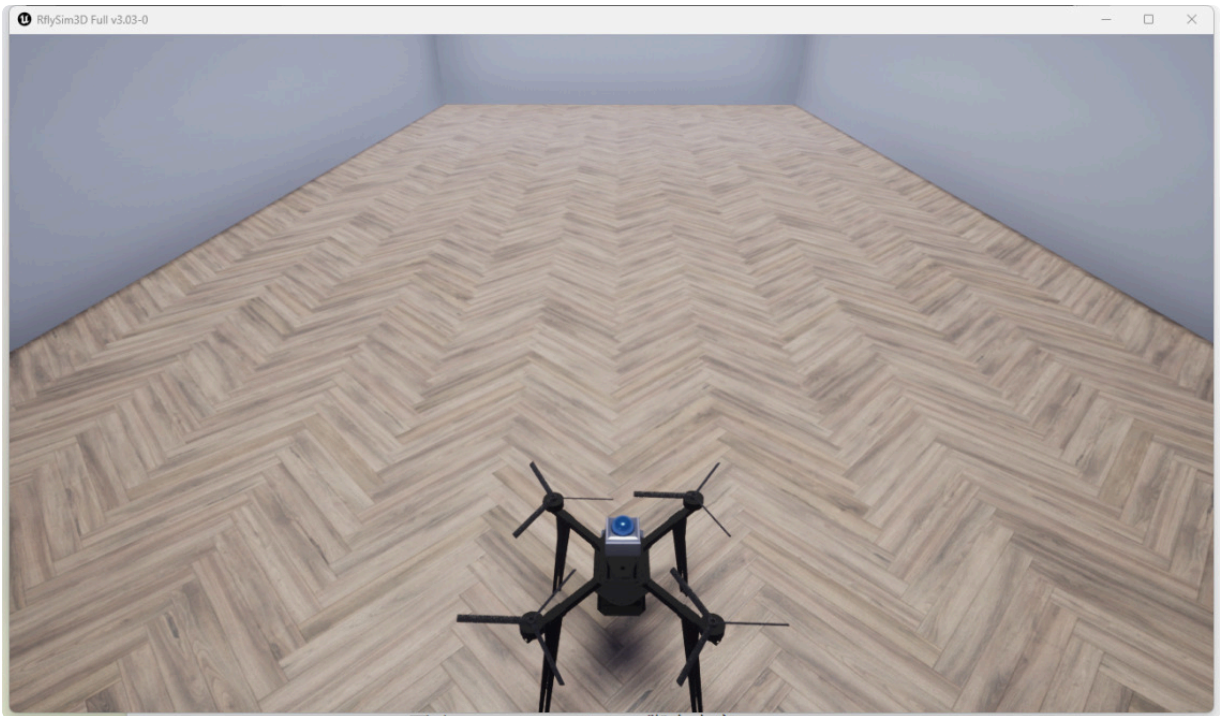
4. 配置传感器（可选）：

参考[\[RflySim安装目录\]/RflySimAPIs/8.RflySimVision/API.pdf](#)

7.3部分，注意：激光雷达只能使用1模式通信方式，因此 SendProtocol[0]只能设置为1

运行：

使用管理员身份运行SITLPosStr.bat脚本即可运行平台，如图：

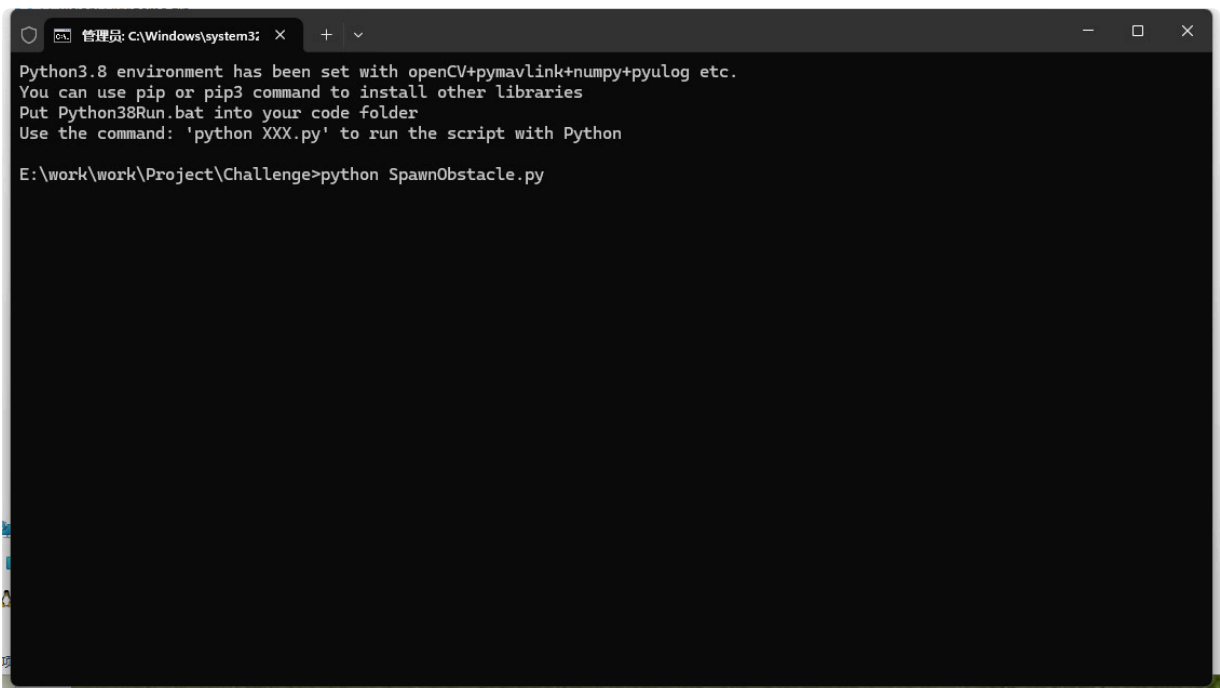


这时候RflySim3D里面仅有一架飞机，没有其他障碍物。等到QGC上提示Ready To Fly即可，如图：

接下来就可以运行Windows/SpawnObstacle.py python

脚本，鼠标右键运行Python38Run.bat(不需要使用管理员权限)，脚本，打开命令行终端，输入

python SpawnObstacle.py, 如图：



此时RflySim平台内会创建障碍物，以及移动小车，且创建Config.json里面的传感器。如图：



3. Ubuntu 系统里的程序

使用NX或者ubuntu 虚拟机，把程代码序拷贝进去。

更改内容main.py脚本内容

`vis.jsonLoad(jsonPath = "/home/nvidia/catkin_ws/src/sensor_pkg/Config.json")`

指定Config.json 文件路径

`vis.RemotSendIP = "192.168.2.5"` 改成自己Ubuntu IP

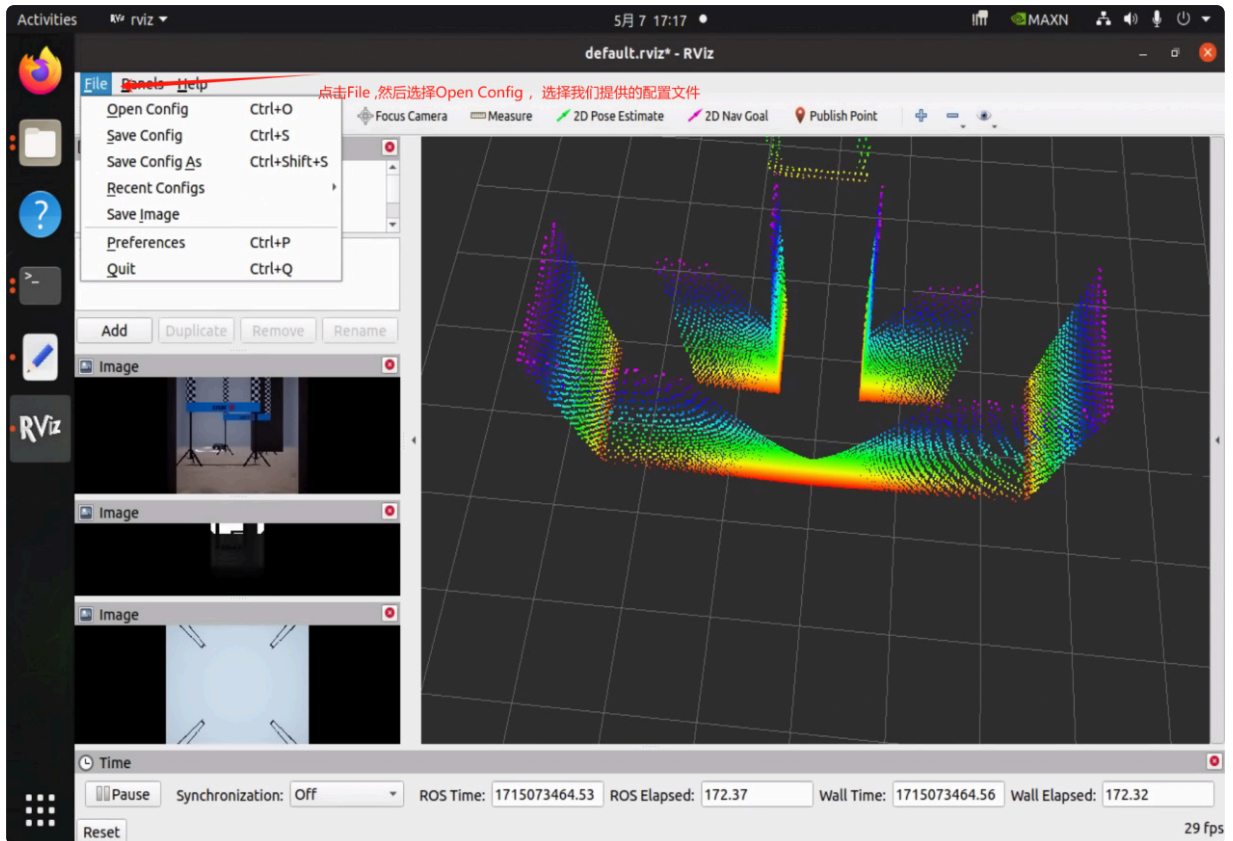
`vis.sendImuReqCopterSim(IP="192.168.2.3")` 指定Windows平台IP

4. 传感器读取：

打开终端运行roscore 启动rosmaster；

另开终端：运行python3 [main.py](#)

另开终端运行rviz ,按照一下步骤加载rviz配置文件



这时候弹出一个文件对话框，选择rflsim.rviz 文件即可。看到上图的数据可视化

5. Mavros连接：

运行命令 `roslaunch mavros px4.launch`

`fcu_url:= "udp://:20101@192.168.2.3:20100"` 这里的IP 为Windows端的IP，端口都是默认值。

附录 使用RflySim GPU加速外挂环境运行实验

除了使用WinWSL编译器，还可使用RflySim基于WSL2/Ubuntu 22.04开发的外挂环境来运行本实验。本外挂环境，附带了docker、cuda、pytorch、tensorflow等额外功能，能够开发更复杂的无人系统感知决策算法。

1. 根据实验例程：`1.RflySimIntro\2.AdvExps\e13.WinWSL2-GPU\Readme.pdf` 来下载WinWSL2-GPU的外挂WSL2镜像（约50G），并部署到平台。
2. 运行 `\Windows\SITLPosStrOnekey.bat` 来启动无人机的SITL仿真。
3. 运行 `WinWSL2.bat` 来进入外挂环境，并和上文相同方法实验。