
1、实验名称及目的

大疆 Livox 激光雷达点云数据 UDP 直传模式实验：通过平台接口 python 发送取图请求给 RflySim3D,后者通过大疆 Livox 激光雷达扫描直接通过 UDP 直传 10hz 频率传出点云数据。

2、实验原理

2.1 Windows 版本地回环地址通信测试

首先运行 LidarAPIDemo.bat 开启一个飞机的软件在环仿真，然后运行 LidarAPIDemo.py 文件，通过 VisionCaptureApi 接口创建一个取图接口，通过 sendUE4Cmd 接口使用 b'RflyChangeMapbyName SLAMScene'参数更换场景，并通过 Config.json 文件加载传感器配置，其中配置文件参数含义如下

“SeqID”代表第几个传感器。此处表示第 1 个传感器（免费版只支持 2 个图）。

“TypeID”代表传感器类型 ID，此处 6 代表 livox 激光雷达；

“TargetCopter”传感器装载的目标飞机的 ID，可改变。“TargetMountType”代表坐标类型，0：固定飞机上（相对几何中心），1：固定飞机上（相对底部中心），2：固定地面上（监控）也可变。

“DataWidth”为激光雷达一个 ring 内的点云个数此处为 250

“DataHeight”为激光雷达线束数量此处为 40。

“DataCheckFreq”点云发布频率此处为 10HZ。

“SendProtocol[8]”为传输方式与地址，SendProtocol[0]取值 0，表示使用共享内存进行数据传输（0：共享内存（免费版只支持共享内存），1：UDP 直传）

“CameraFOV”为相机视场角（仅限视觉类传感器），单位度也可改变。

“SensorPosXYZ[3]”为传感器安装位置，单位米也可改变。

“SensorAngEular[3]”为传感器安装角度，单位度°也可改变。

“otherParams[8]”：[激光最远距离(m),精度(m),水平扫描角度下限值(度),水平扫描角度上限值(度),垂直扫描角度下限值(度),垂直扫描角度上限值(度),预留,预留]

然后向 RflySim3D 发送取图请求，通过共享内存的方式得到点云数据，通过死循环将不断得到的点云数据输入到自定义的绘制点云图函数从而绘制点云图，使得点云图一直显示在屏幕上。

2.2 Windows 运行客户端与 Ubuntu 远程通信

首先更改 RunRflysim3DSITL.bat、client_ue4.py、server_ue4.py、Config.json 的 IP 地址，使用指定 IP 传输，使得传输更加稳定与快捷，通过 Config.json 文件配置传感器，然后运行 client_ue4.py 向 RflySim3D 发送取图请求，通过 UDP 直传 png 压缩的方式将图像数据传输到 Ubuntu 虚拟机，Ubuntu 虚拟机运行 server_ue4.py 通过 PX4MavCtrlr 创建控制接口，以此向 RflySim 发送控制指令，并将包含传感器数据的话题发布,启动 Rviz，添加话题，使得 R

viz 能够获得到 server_ue4.py 发布的话题消息，同时添加 lidar.rviz 配置文件，使得 Rviz 生成自定义 tf 坐标的结果。

3、实验效果

本实验通过平台接口进行 RflySim3D 直接 10hz 频率 UDP 直传点云数据。

4、文件目录

文件夹/文件名称	说明
LidarAPIDemo.bat	启动仿真配置文件
LidarAPIDemo.py	Python 实验脚本
client_ue4.py	Windows 下客户端文件
server_ue4.py	Ubuntu 下服务端文件
Config.json	视觉传感器配置文件
lidar.rviz	Rviz 配置文件
Livox_theory.pdf	Livox 激光雷达原理详解

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		
4	Ubunt 虚拟机		

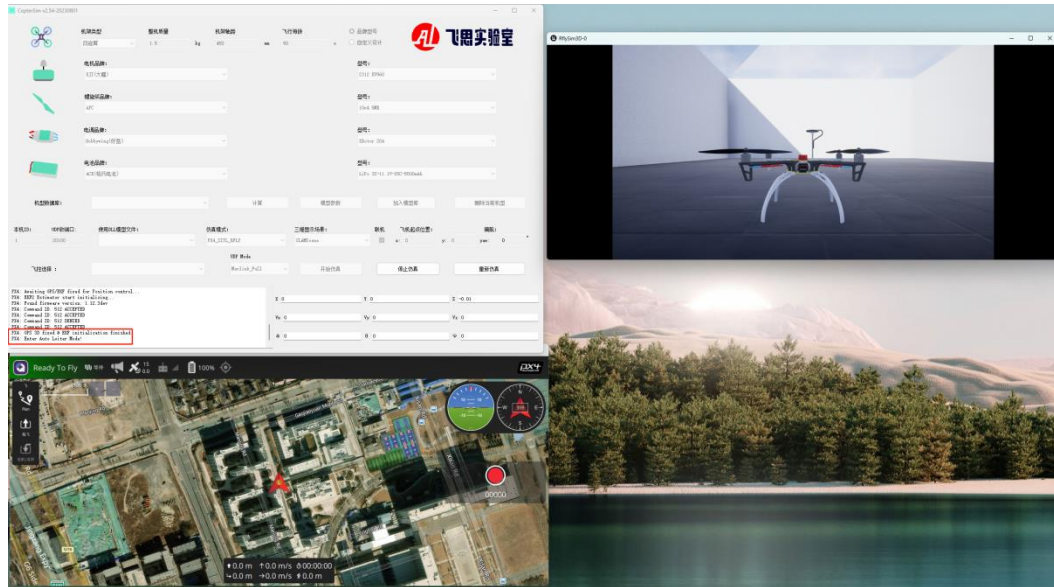
①：推荐配置请见：<https://doc.rflysim.com/1.1InstallMethod.html>

6、实验步骤

6.1 Windows 版本本地回环地址通信测试

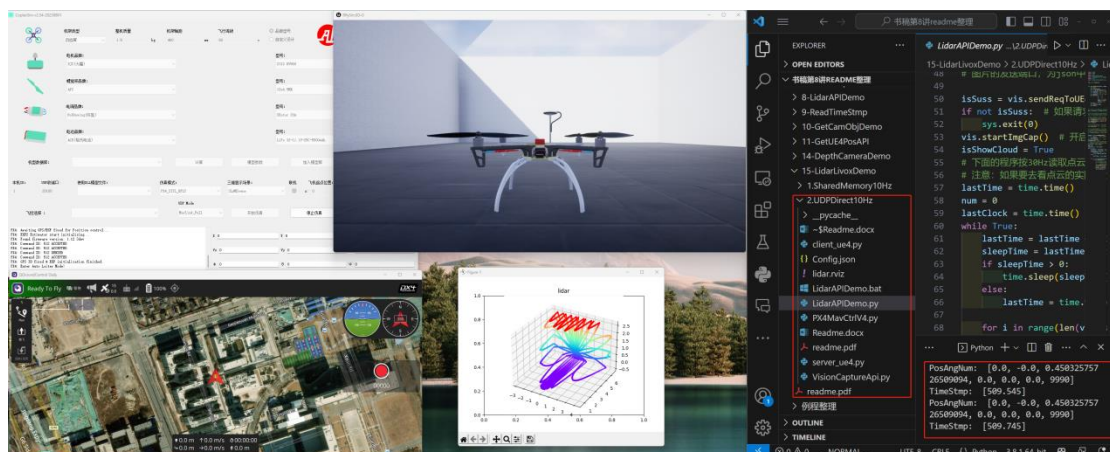
Step 1:

以管理员方式运行 LidarAPIDemo.bat 开启一个飞机的软件在环仿真。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 RflySim3D 软件内有 1 架无人机。



Step 2:

用 VScode 打开到本实验路径文件夹，运行 LidarAPIDemo.py 文件，可以看到一个三维点云窗口界面以及终端出现的提示。

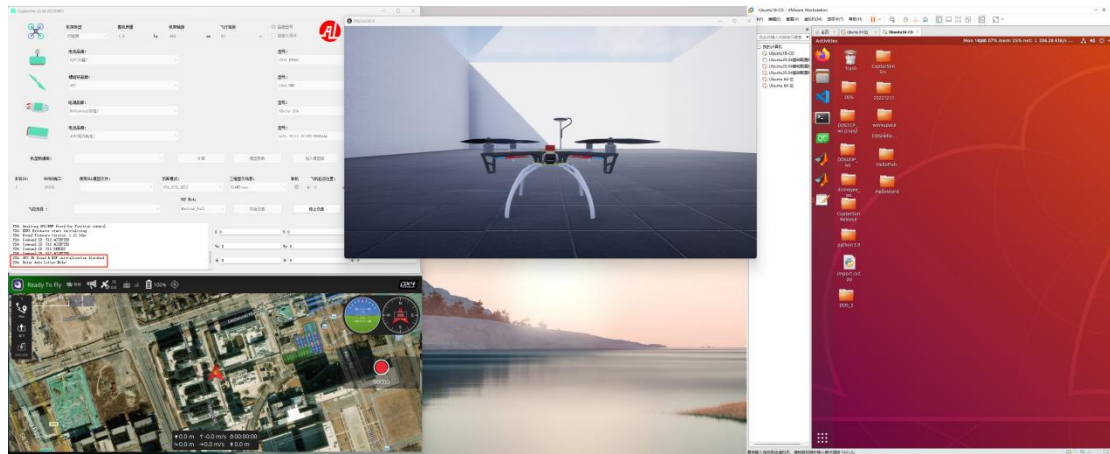


6.2 Windows 运行客户端与 Ubuntu 远程通信

Step 1:

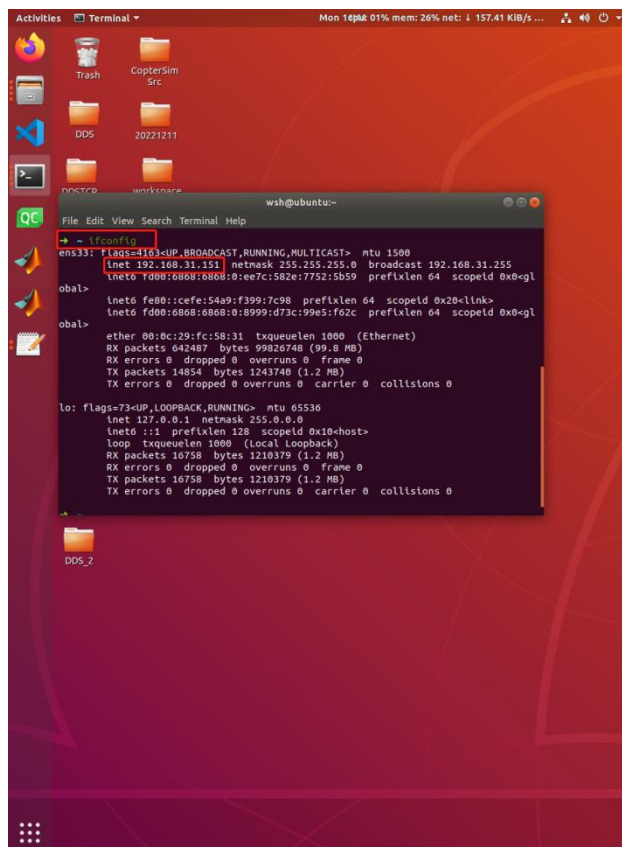
以管理员方式运行 LidarAPIDemo.bat 开启一个飞机的软件在环仿真。将会启动 1 个 GC 地面站，1 个 CopterSim 软件且其软件下侧日志栏必须打印出 GPS 3D fixed & EKF initialization finished 字样代表初始化完成，并且 RflySim3D 软件内有 1 架无人机。

并且启动一个已安装了 ros 的 Ubuntu 虚拟机。



Step 2:

在虚拟机终端中输入 `ifconfig` 命令，查找该虚拟机的 IP 地址，如下图：

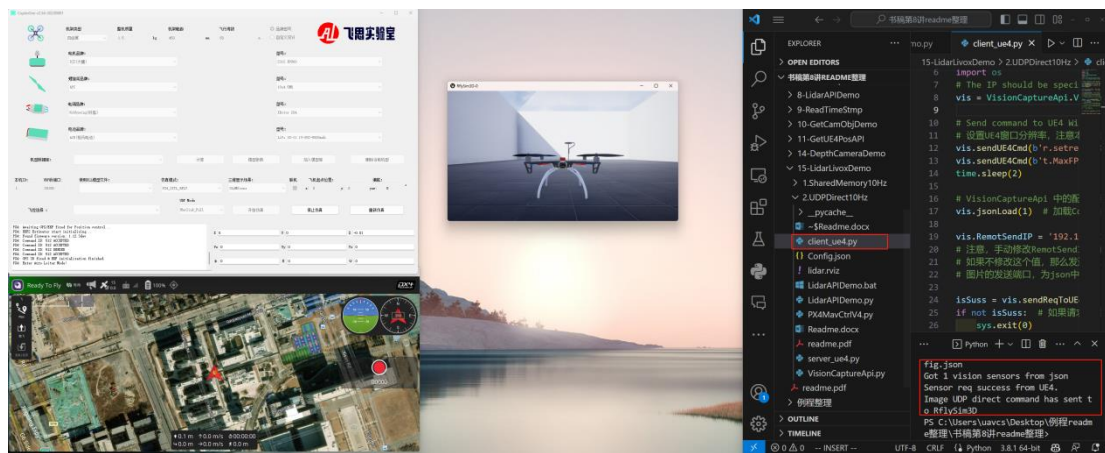
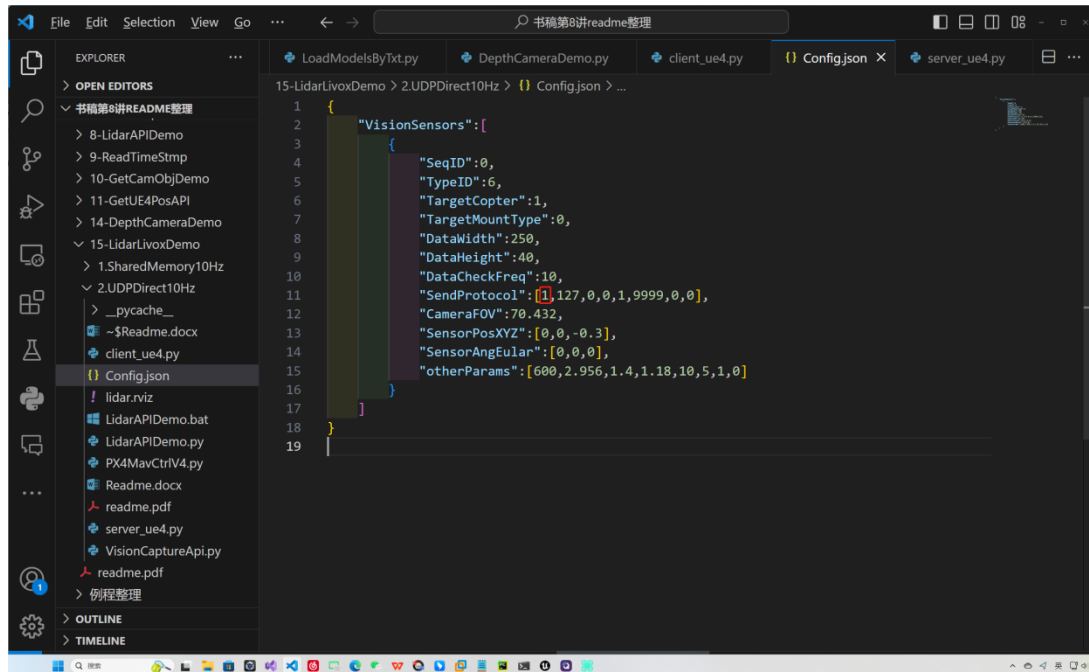


Step 3:

修改 `client_ue4.py` 与 `server_ue4.py` 中的代码 `vis.RemotSendIP` 值，改成自己的虚拟机地址。在 Windows 运行 `client_ue4.py`。并修改文件夹下 `Config.json` 配置文件下的 `SendProtocol` 的第一个数为 1，确保可以与虚拟机通信，如果是 0 则是 Windows 下本地通信。

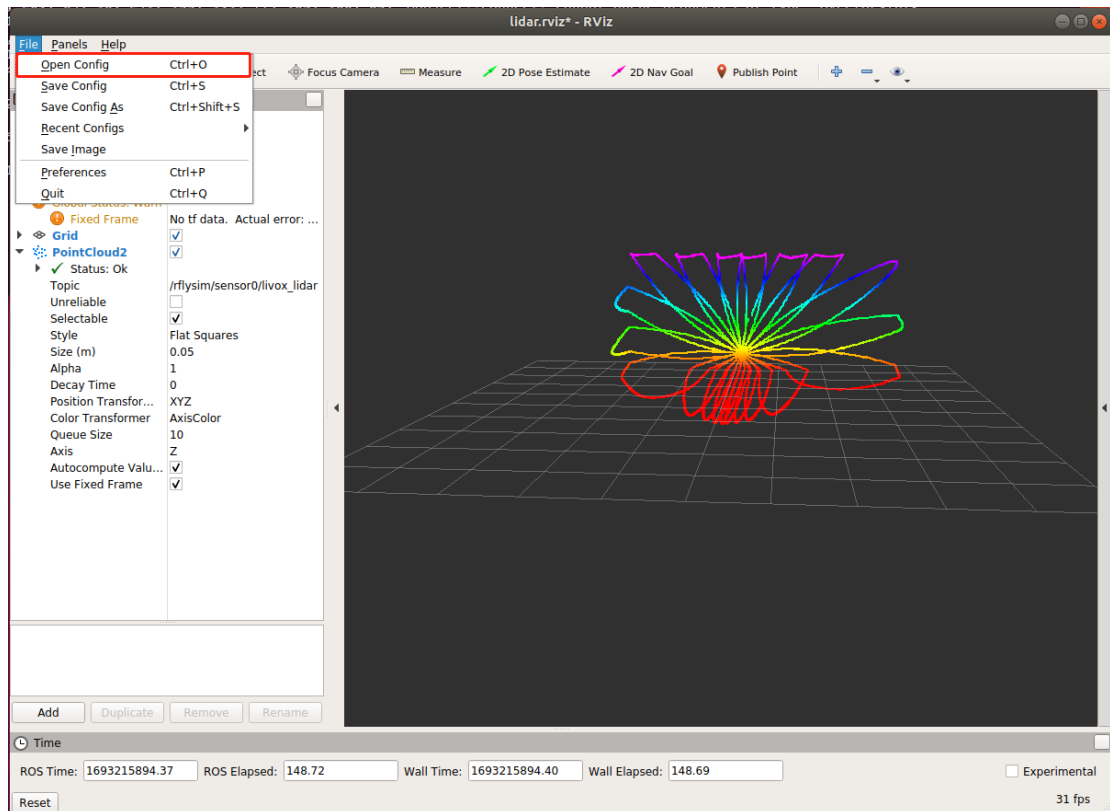
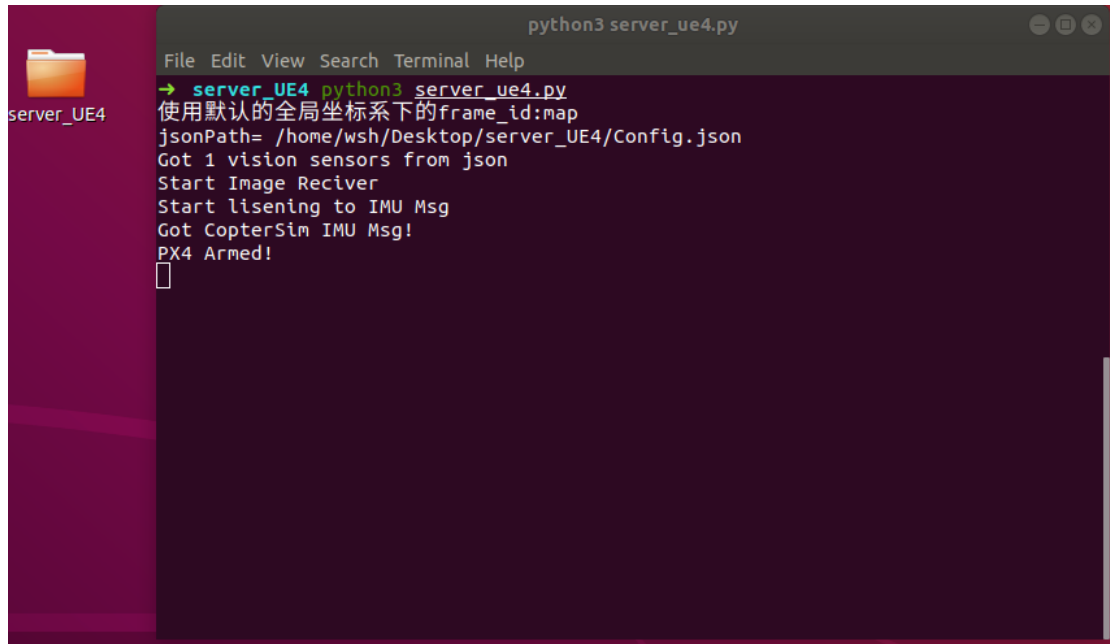
```
9 import PX4MavCtrlV4 as PX4MavCtrl
10 import math
11
12 # 启用ROS发布模式
13 VisionCaptureApi.isEnableRosTrans = True
14 vis = VisionCaptureApi.VisionCaptureApi()
15
16 # VisionCaptureApi 中的配置函数
17 vis.jsonLoad() # 加载Config.json中的传感器配置文件
18 vis.startImgCap() # 开启取图循环, 执行本语句之后, 已经可以通过vis.Img[i]读取到图片了
19 print('Start Image Receiver')
20
21 vis.RemotSendIP = "192.168.31.151"
22
23 vis.sendImuReqCopterSim()
24 # vis.sendImuReqServe()
25 vis.sendUE4Pos(1, 3, 0, [5.5, 0, -0.5])
26
27 # 控制飞机起飞到一定高度
28 VehicNum = 1
29 MavList = []
30 # Create MAV instance
31 for i in range(VehicNum):
32     MavList = MavList+[PX4MavCtrl.PX4MavCtrl(20100+i*2, '255.255.255.255')]
33
34 time.sleep(2)
35 # Start MAV loop with UDP mode: MAVLINK_FULL
36 for i in range(VehicNum):
37     MavList[i].InitMavLoop()
```

```
6 import os
7 # The IP should be specified by the other computer
8 vis = VisionCaptureApi.VisionCaptureApi()
9
10 # Send command to UE4 Window 1 to change resolution
11 # 设置UE4窗口分辨率, 注意本窗口仅限于显示, 取图分辨率在json中配置, 本窗口设置越小, 资源需求越少
12 vis.sendUE4Cmd(b'r.setres 720x405w', 0)
13 vis.sendUE4Cmd(b't.MaxFPS 30', 0) # 设置UE4最大刷新频率, 同时也是取图频率
14 time.sleep(2)
15
16 # VisionCaptureApi 中的配置函数
17 vis.jsonLoad(1) # 加载Config.json中的传感器配置文件
18
19 vis.RemotSendIP = '192.168.31.151'
20
21 # 注意, 手动修改RemotSendIP的值, 可以将图片发送到远端Linux电脑的IP地址
22 # 如果不修改这个值, 那么发送的IP地址为json文件中SendProtocol[1:4]定义的IP
23 # 图片的发送端口, 为json中SendProtocol[5]定义好的。
24
25 isSuss = vis.sendReqToUE4() # 向RflySim3D发送取图请求, 并验证
26 if not isSuss: # 如果请求取图失败, 则退出
27     sys.exit(0)
28
29 # isSuss = vis.sendImuReqCopterSim(
30     # copterID=1, IP='192.168.31.88', port=31000, freq=200)
31 # isSuss = vis.sendImuReqClient()
32 # if not isSuss:
33     # sys.exit(0)
34
35 # 注意: 这里不需要调用startImgCap()来去图, 本程序只是发送数据请求
36
37 print('!Teage UDP direct command has sent to RflySim3D')
```

Step 4:

在虚拟机中新建一个文件夹，并将该实验文件全拷贝过去，然后在终端中运行 `roscore` 指令，将起新终端路径选择到新建的文件夹路径，再运行命令 `python3 server_ue4.py` 运行脚本 `server_ue4.py`。然后另起一个终端运行 `rviz` 命令，打开 `rviz` 可视化工具，点击 **Add** 键添加 `PointCloud2`，然后在 **Topic** 选择话题/`rflysim/sensor3/livox_lidar` 数据，然后选择文件夹下的 `lidar.rviz` 配置文件可看到如下效果：



Step 5:

在下图“LidarAPIDemo.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。



7、参考文献

[1]. Livox 激光雷达原理详解 Livox_theory.pdf