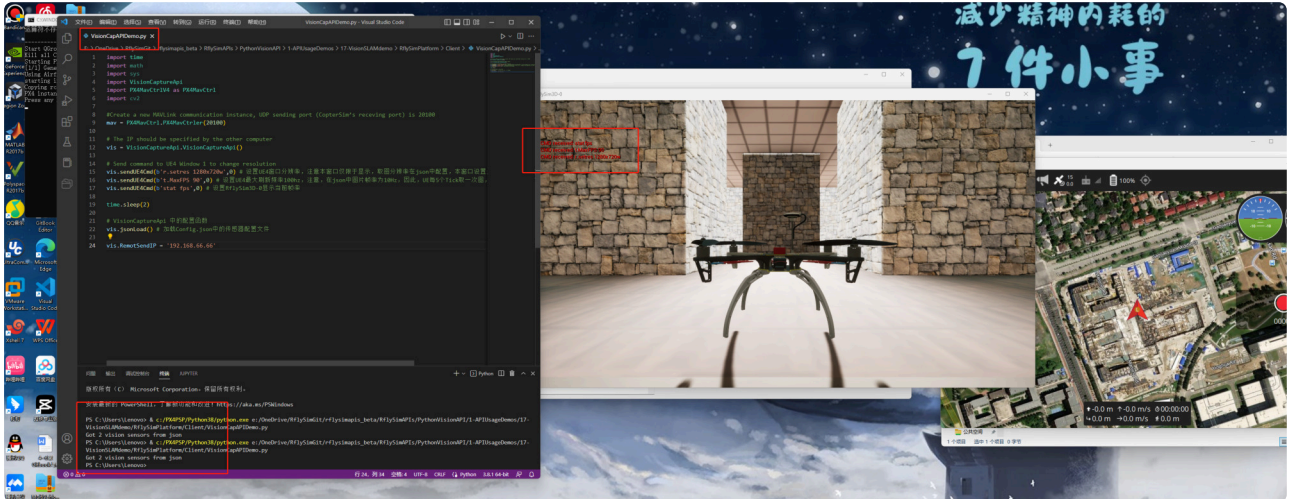


1.打开RflySimAPIs\PythonVisionAPI\1-APIUsageDemos\17-VisionSLAMdemo\RflySimPlatform\Client已管理员方式运行VisionCapAPIDemo.bat，等待3DFixed之后，再用Python运行VisionCapAPIDemo.py，待起飞之后，在SITL的黑窗口按下任意键，关闭CopterSim和RflySim3D，此时Python内数据会停止更新，将数据拷贝到一个txt文件里面，进行后续的分析。运行效果如下图所示：



注意事项：

### 1.修改VisionCaptureApi.py，打印图片和IMU的时间戳

- 1) img\_mem\_thrd函数，“self.timeStmp[idx]=”语句后，添加时间戳打印语句
- 2) getIMUDataLoop函数，“self.imu.timestamp=”语句后，添加时间戳打印语句

2.修改VisionCapAPIDemo.py文件，“vis.sendUE4Cmd(b't.MaxFPS90',0)”语句，其中的90可以替换成希望UE4运行的帧率。

注意：从UE取图都会有一个Tick（例如90Hz就是1/90.0秒）的延迟，也就是在下一帧才能拿到图片并发送。因此，UE的运行帧率决定了图像取图延迟的最小值（还需要加入数据传输等延迟）

3.修改Config.json文件，DataCheckFreq的条目对应了图片的取图与发送频率。

- 1) 在本例中，UE的MaxFPS设置为90Hz，而DataCheckFreq取图30Hz，则UE会每3个Tick发送一次图片，最小延迟是1/90每秒。
- 2) 如果设置UE的MaxFPS设置为30hz，而取图DataCheckFreq取图30Hz，则最小延迟是1/30Hz。
- 3) 如果设置UE的MaxFPS设置为100hz，而取图DataCheckFreq取图10Hz，则最小延迟是1/100Hz。

4) 如果设置UE的MaxFPS设置为10hz，而取图DataCheckFreq取图10Hz，则最小延迟是1/10Hz。

4.从上面分析可见

1) MaxFPS尽量取更大值，能显著减小取图延迟，但是对显卡的考验会越来越大。

2) DataCheckFreq的取图频率应该尽量是MaxFPS的公因数，可以整除，这样能够得到稳定的取图频率。

5.VisionCapAPIDemo.py中“vis.sendImuReqCopterSim(1)”语句的意思是向CopterSim请求IMU的数据，采用的默认频率为200Hz。

1) IMU的时间戳和取图时间戳都是用的CopterSim开始仿真后的时间。IMU的时间戳是从CopterSim直接通过UDP读取，几乎没有延迟；

2) Python的取图时间戳，是CopterSim先发送给RflySim3D（约50Hz或100Hz），RflySim3D经过一个Tick的延迟，再随着图片一起转发到本Python程序。

3) 通过图片时间戳与IMU时间戳的对比，可以大致判断取图环境的延迟情况。

6.log\UE90Hz-Capture30Hz-IMU200hz.txt对应了本demo的记录数据，通过Capture：的图像时间戳与最近的IMU时间戳对比，可知本取图频率稳定在30Hz，且取图延迟在0.01~0.015s左右。这个时间延迟已经非常小了（小于人眼的反应时间），因此是可以用于控制无人机的高机动飞行的。