
1. 实验名称及目的

1.1. 实验名称

时间戳获取实验

1.2. 实验目的

通过 python 接口获取时间戳数据。

注意：本实验只支持 Windows 下 Python 环境运行。不支持在 WSL 以及虚拟机下运行。

1.3. 关键知识点

在控制任务中通过使用 `mav.StartTimeStmplisten` 接口监听飞机时间戳，该函数绑定本机的 20005 端口，在子线程中接收 CopterSim 回传的心跳和时间戳数据，确保终端和 CopterSim 的连接正常，并不阻塞主线程。其中 `cpID` 表示 CopterSim 的 ID。

在数据传输任务中通过使用 `vis.StartTimeStmplisten` 接口监听飞机时间戳，该函数绑定本机的 20005 端口，在子线程中接收 CopterSim 回传的心跳和时间戳数据，确保终端和 CopterSim 的连接正常，并不阻塞主线程。其中 `cpID` 表示 CopterSim 的 ID。

本实验主要是实现通过 Python 接口 `VisionCaptureApi.py`（见 `RflySimAPIs\RflySimSDK\vision` 目录）获取时间戳数据。关键代码解析如下：

1）视觉接口使用

```
vis = VisionCaptureApi.VisionCaptureApi() # 创建一个视觉传感器实例
vis.jsonLoad() # 加载 Config.json 中的传感器配置文件
isSuss = vis.sendReqToUE4() # 向 RflySim3D 发送取图请求
vis.sendImuReqCopterSim() # 发送请求，从目标飞机 CopterSim 读取 IMU 数据，
会自动开启数据监听，已经可以通过 vis.imu 读取到 IMU 数据了。
vis.startImgCap() # 开启取图
```

2）相机数量和参数配置

其中，视觉传感器的初始状态由本文件夹下的 `Config.json` 决定，主要包含以下配置项：

`"SeqID":0`：使用自动更新 ID 的方式，创建了 SeqID 为 0 和 1 的两个视觉传感器

`"TypeID":1`：传感器类型为 RGB 彩色图像

`"TargetCopter":1`：相机绑定在 1 号飞机上

`"SendProtocol":[0,0,0,0,0,0,0,0]`：传输模式为 0 共享内存机制，因此本例程只能运行在 Windows 环境下。

`"SensorPosXYZ":[0.3,0,0]`：相机的分布。

3) UE 控制

接口详细使用方法见：[UE4CtrlAPI.py](#)

```
ue = UE4CtrlAPI.UE4CtrlAPI() # 创建 UE 控制实例
ue.sendUE4Cmd('r.setres 720x405w',0) # 发送指令，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。
ue.sendUE4Cmd('t.MaxFPS 200',0) # 发送指令，设置 UE4 最大刷新频率 200Hz，
同时也是取图频率
ue.sendUE4Cmd('stat fps',0) # 在 UE4 窗口显示当前刷新频率
```

2. 实验效果

本实验通过 python 接口获取时间戳数据。时间戳内包含校验位、当前飞机 ID、仿真起始时间戳、当前时间戳、心跳计数等。

3. 文件目录

例程目录：[\[安装目录\]\RflySimAPIs\8.RflySimVision\0.ApiExps\1-UsageAPI\6.ReadTimeStmp\](#)

文件夹/文件名称	说明
ReadTimeStmp.bat	一键仿真启动脚本
ReadTimeStmp.py	Python 实验代码
Python38Run.bat	Python 环境启动脚本

4. 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 工具链		
3	VS Code		可选

①：推荐配置请见：<https://rflysim.com>

5. 实验步骤

5.1. 必做实验：Windows 取图控制

Step 1: 开启仿真

双击运行 ReadTimeStmp.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: 运行控制程序

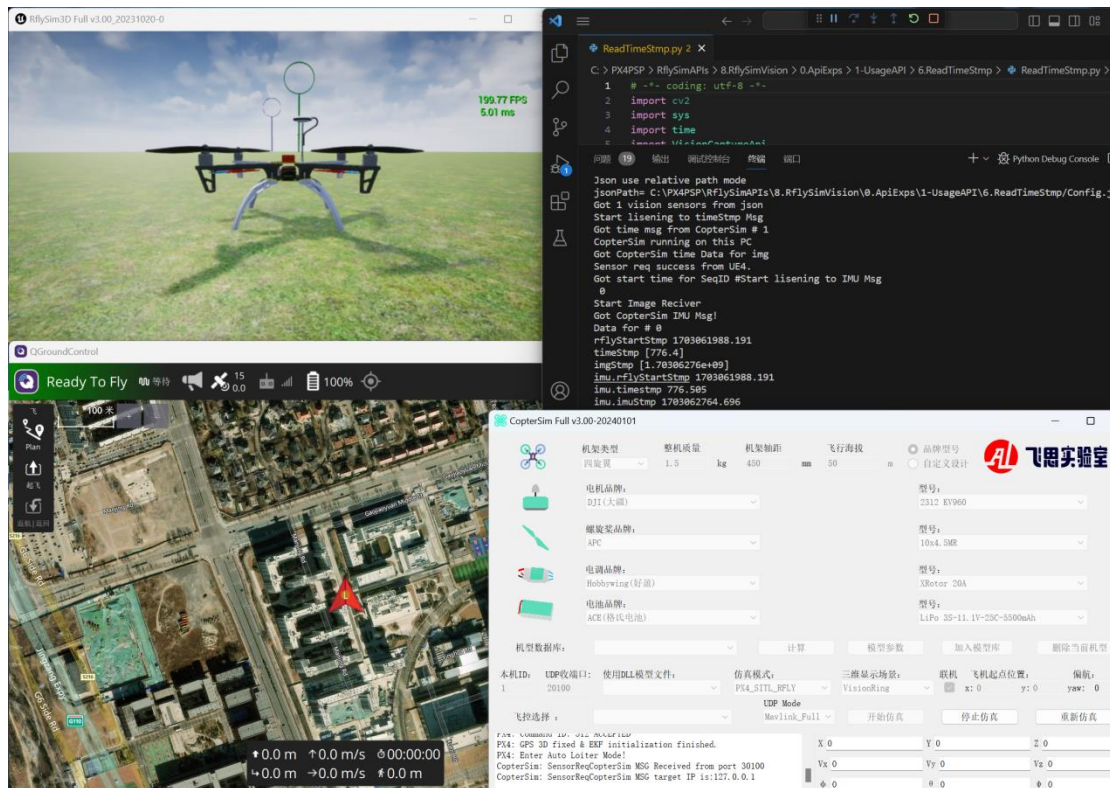
在文件夹下，双击 Python38Run.bat，打开集成好的 python 环境，在该环境下运行 ReadTimeStmp.py 文件，输入 python ReadTimeStmp.py

```
C:\Windows\system32\cmd.exe
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.
You can use pip or pip3 command to install other libraries
Put Python38Run.bat into your code folder
Use the command: 'python XXX.py' to run the script with Python

C:\Users\uavcs\Desktop\demo\8.RflySimVision\0.ApiExps\1-UsageAPI\6.ReadTimeStmp>python ReadTimeStmp.py
```

Step3: 观察结果

可以订阅得到时间戳，效果如下图所示：



Step 4: 结束仿真

在下图“ReadTimeStmp.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。

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\3 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 Step2 运行 ReadTimeStmp.py 时，可使用 VS Code（或 Pycharm 等工具）来打开 ReadTimeStmp.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 ReadTimeStmp.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。

```

VisionCapAPIDemo.py X
> RflySimAPIs > 8.RflySimVision > 0.ApiExps > 1-UsageAPI > 0.VisionSensorAPI > 1.CameraAPI
8   ue = UE4CtrlAPI.UE4CtrlAPI()
9
10  #Create a new MAVLink communication instance, UDP sending
11  mav = PX4MavCtrl.PX4MavCtrl(1)
12
13  # The IP should be specified by the other computer
14  vis = VisionCaptureApi.VisionCaptureApi()
15
16  # Send command to UE4 Window 1 to change resolution
17  ue.sendUE4Cmd('r.setres 1280x720w',0) # 设置UE4窗口分辨率，
18  ue.sendUE4Cmd('t.MaxFPS 30',0) # 设置UE4最大刷新频率，同时也
19  time.sleep(2)
20
21  # VisionCaptureApi 中的配置函数
22  vis.jsonLoad() # 加载Config.json中的传感器配置文件

```

- 请尝试修改代码，实现飞机位置改变、相机姿态角改变、相机参数改变等功能。

6. 参考资料

[1]. 无

7. 常见问题

Q1: 无

A1: 无