

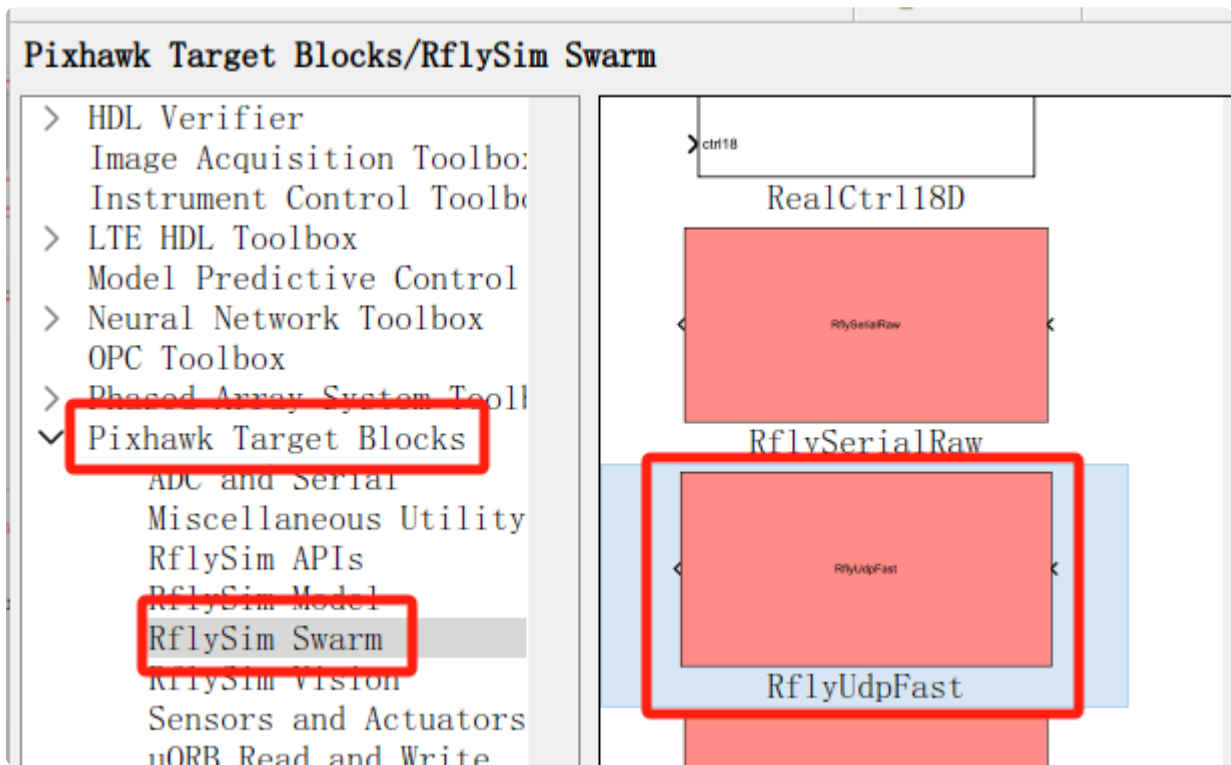
# 1. 实验名称及目的

## 1.1 实验名称

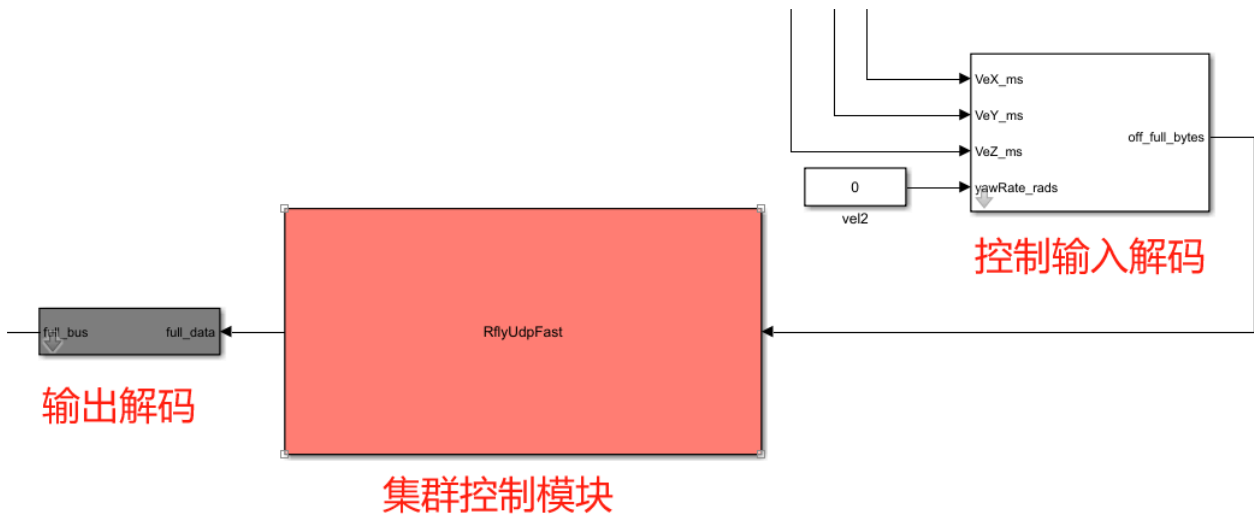
RflyUdpFast通信接口的FullData模式单机实验

## 1.2 实验目的

通过RflySim工具链的Simulink库 – Pixhawk Target Blocks – RflySim Swarm提供的RflyUdpFast传输模块



接收无人机的状态信息，然后对单个无人机的局部位置运动控制进行Simulink建模，并发送控制指令到该模块。



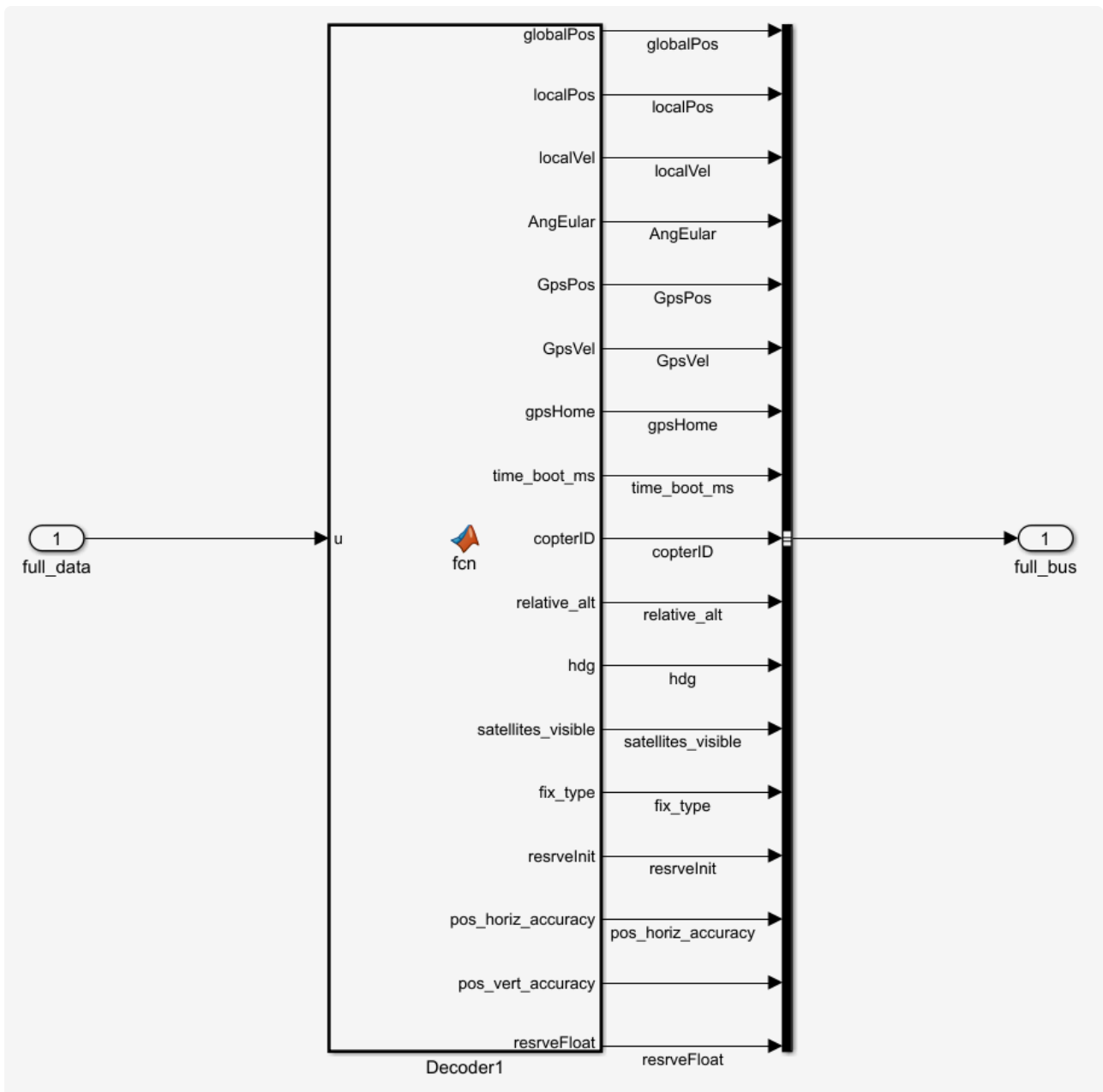
## 1.3 关键知识点

了解RflySim

Swarm的RflyUdpFast传输模块搭建Simulink模型的参数配置信息，关键点“设置成FullData Mode”；



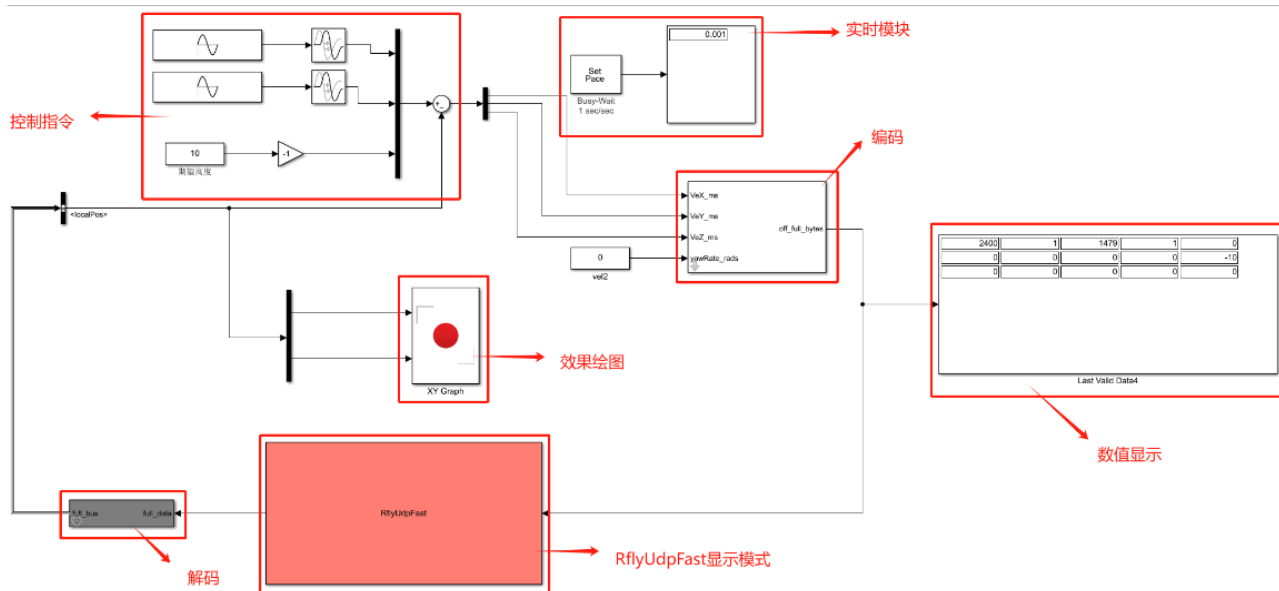
FullData完整模式输出bus支持的消息如下



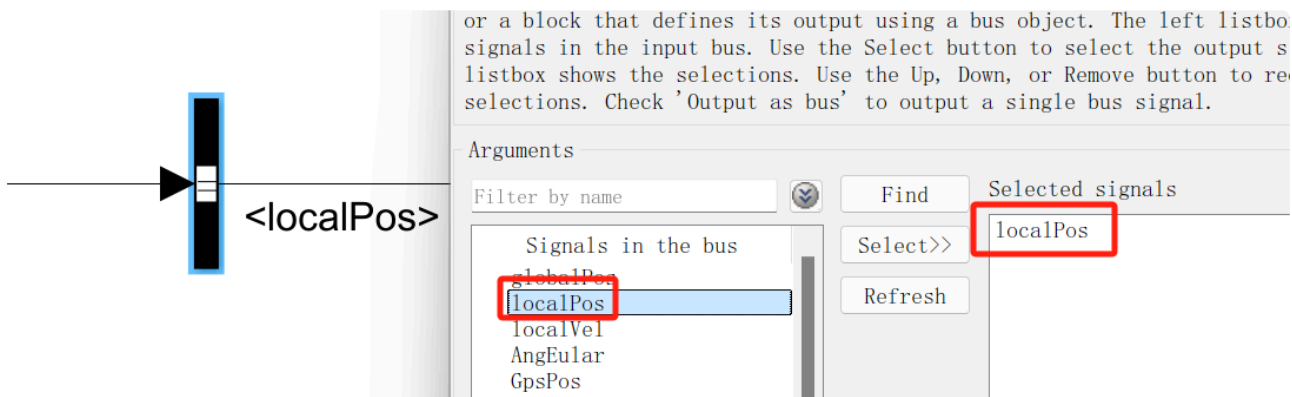
熟悉使用RflyUdpFast传输模块开展软件在环实验及硬件在环实验。

使用RflySimTool工具链提供的RflyUdpFast传输模块来接收四旋翼无人机的状态信息。这个传输模块可以通过UDP快速传输数据，以实现实时信息的接收。

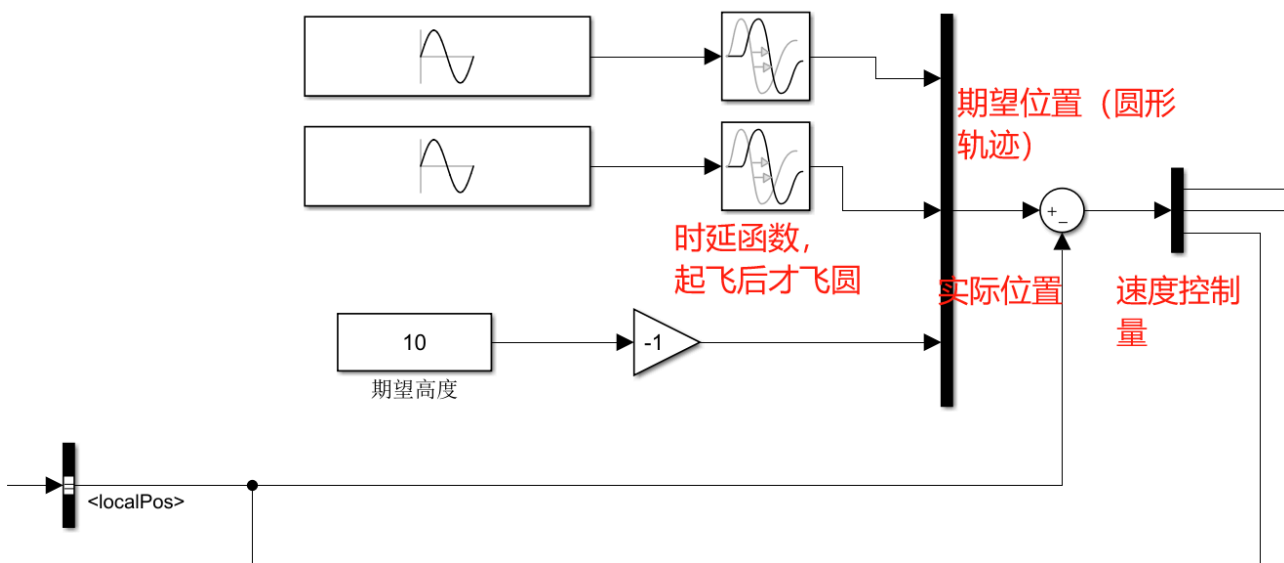
在Simulink中建立一个模型，包括RflyUdpFast传输模块和用户搭建的控制算法模块；



本例程订阅FullData总线输出的localPos位置数据；



和期望轨迹做差（一个圆形飞行轨迹），生成速度控制指令；



经过vel\_ned\_full模块，将速度vx vy vz yaw速度指令，转化为RflyUdpFast需要的信号维度并最终通过offboard的信号发给飞控；

运行模型进行仿真，观察控制指令的发送和四旋翼无人机的局部位置运动控制效果。

通过这个实验，您可以验证控制算法在接收无人机状态信息后的运行效果，并优化算法以确保其性能和可靠性。请注意调整模型参数和算法设计，以满足您的具体实验要求和目的。

## 2. 实验效果



## 3. 文件目录

文件夹/文件名称	说明
<HITLPosStrGPS.bat>	常规硬件在环仿真脚本（支持位置自定义）
<HITLPosStrGPSSysID.bat>	硬件在环仿真脚本，CopterID由飞控SysID确定
<HITLRunPX4Net.bat>	硬件在环视觉盒子、集群盒子或Pixhawk 6x的网口仿真模式（限完整版）
<RflyUdpFullOne.bat>	纯软件在环仿真
<GenerateSwarmExe.p>	.exe文件生成一键运行脚本（限完整版才能运行）
<RflyUdpFullOne.slx>	Simulink控制模型主程序

文件夹/文件名称	说明
<RflyUdpFullOne.exe>	生成好的exe版本程序，支持高性能运行
<Readme.docx>	用户指南

## 4. 运行环境

### 4.1 软件要求

Windows 10及以上版本；RflySim工具链；MATLAB R2022b及以上版本。

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；Pixhawk 6X 1台。

①：推荐配置请见：<https://rflysim.com/>

## 5. 实验步骤

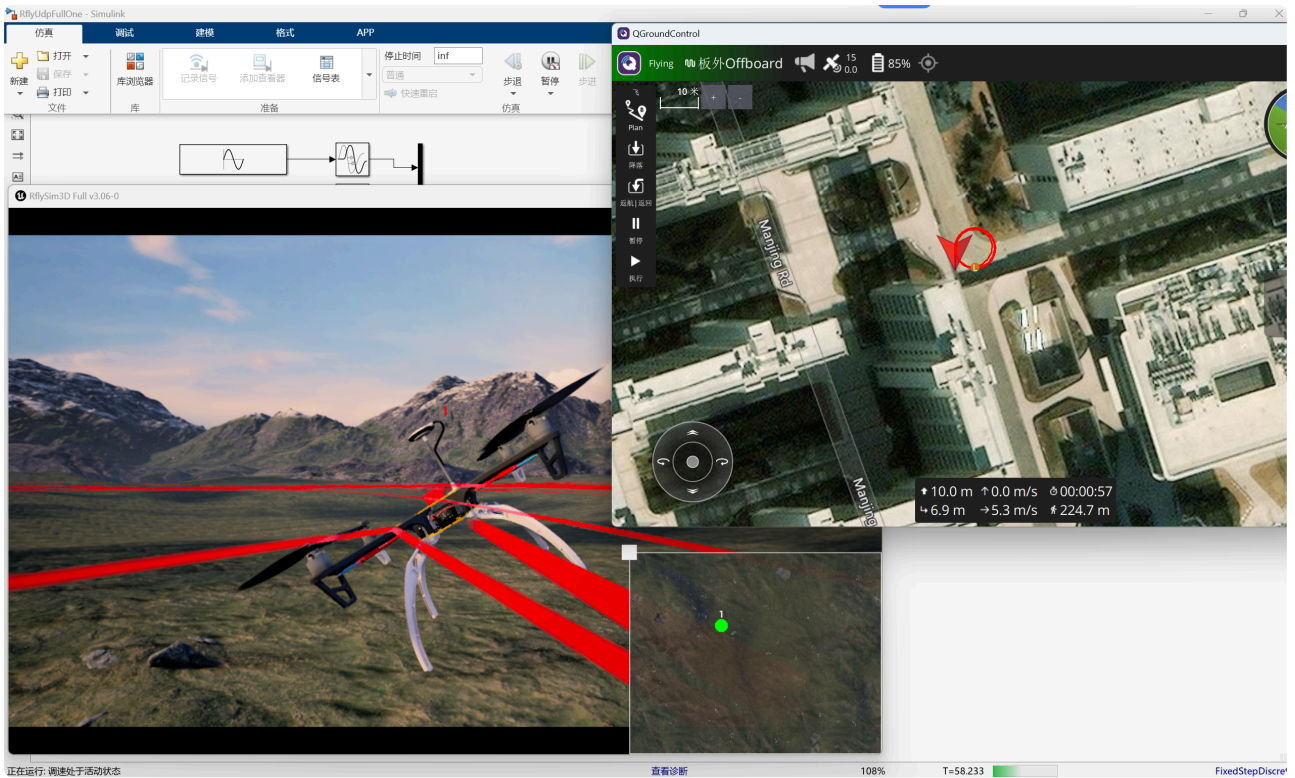
### 软件在环仿真实验步骤

(1) 双击打开<RflyUdpFullOne.bat>即可自动启动RflySim3D、CopterSim、QGroundControl软件，等待所有CopterSim的状态框中显示：PX4: GPS 3D fixed & EKF initialization finished。

(2) 在MATLAB打开<RflyUdpFullOne.slx>文件，点击上方运行按钮。

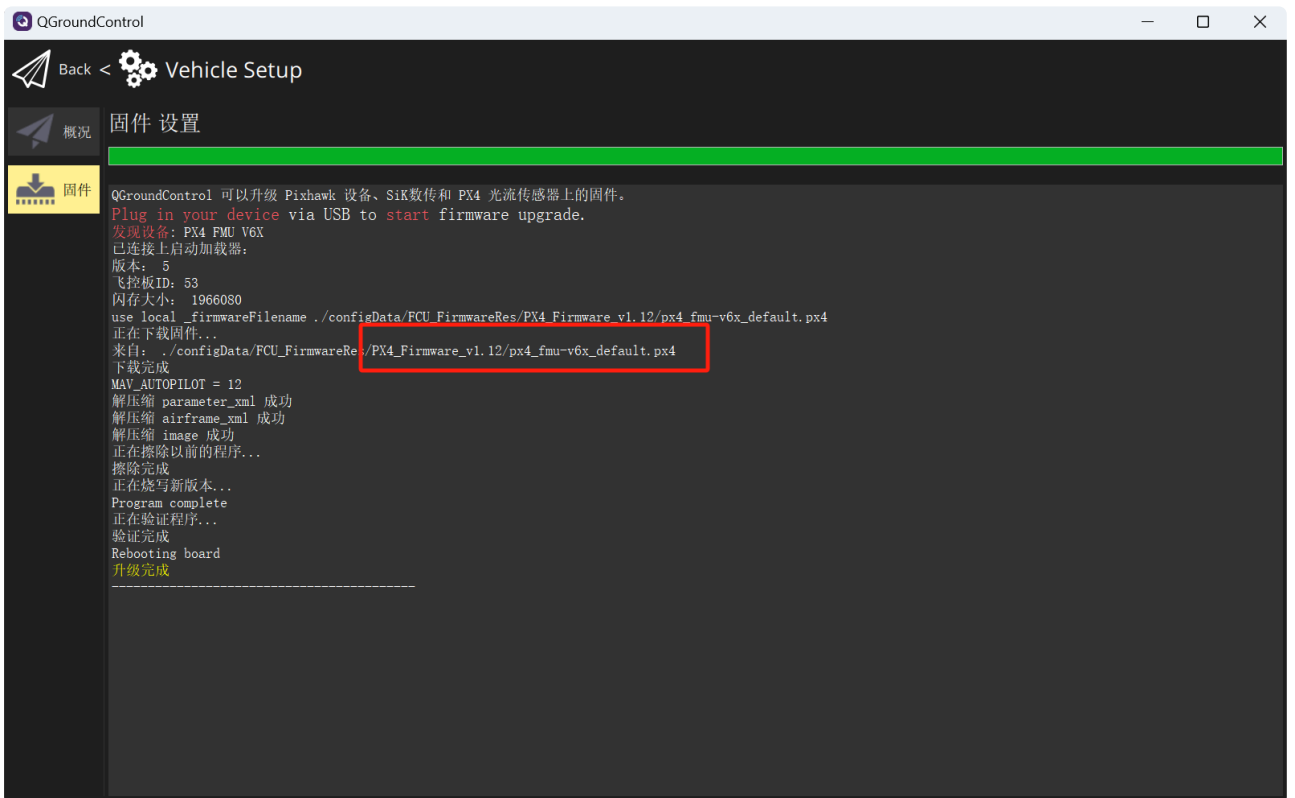


(3) 打开RflySim3D窗口，按S键打开飞机标号，T键开启飞机飞行轨迹，L键开启小地图。在QGC软件中可以看到飞机的飞行路线。



## 硬件在环仿真实验步骤

(1) 在做实验之前需要将飞控还原。



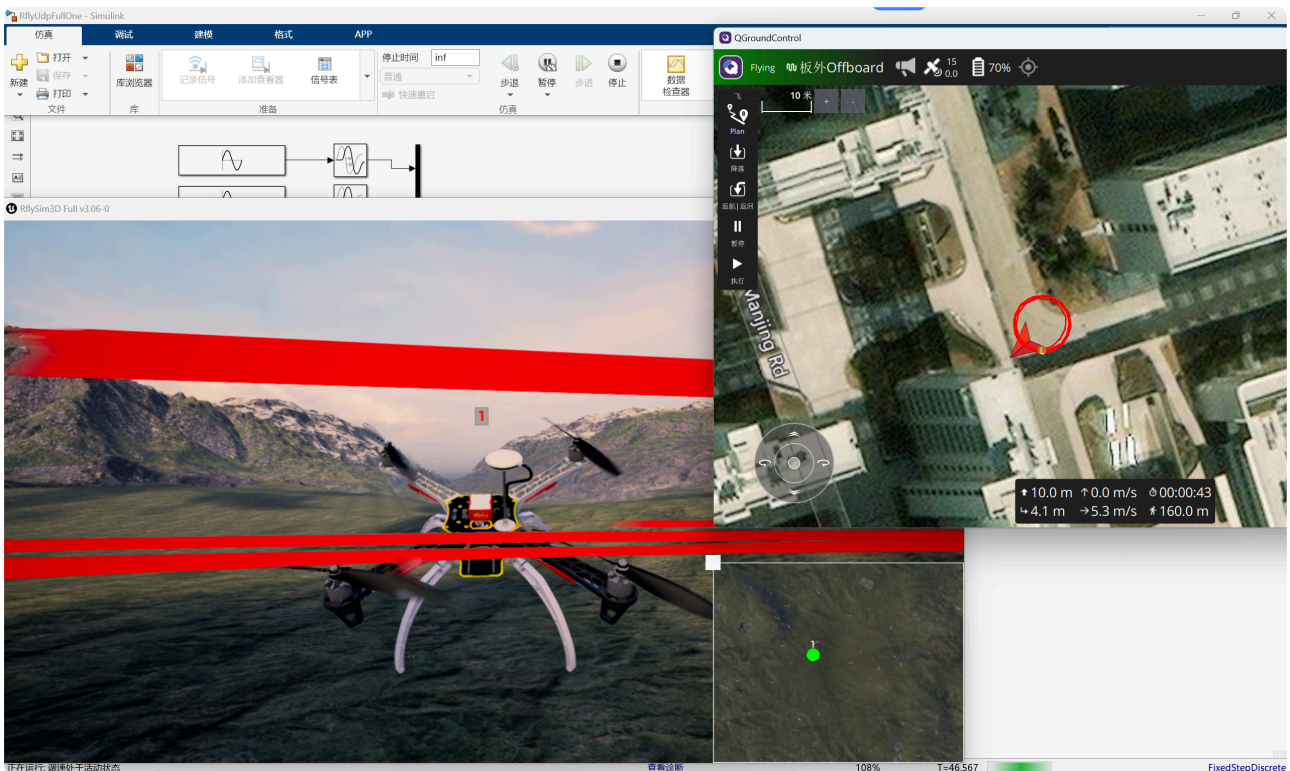
(2) 使用USB线连接飞控与电脑。

(3) 打开<HITLPosStrGPS.bat>一键运行脚本，即可自动启动RflySim3D、CopterSim、QGroundControl软件，等待所有CopterSim的状态框中显示：PX4: GPS 3D fixed & EKF initialization finished。

(4) 在MATLAB打开<RflyUdpFullOne.slx>文件，点击上方运行按钮。



(5) 查看RflySim 3D实验效果。

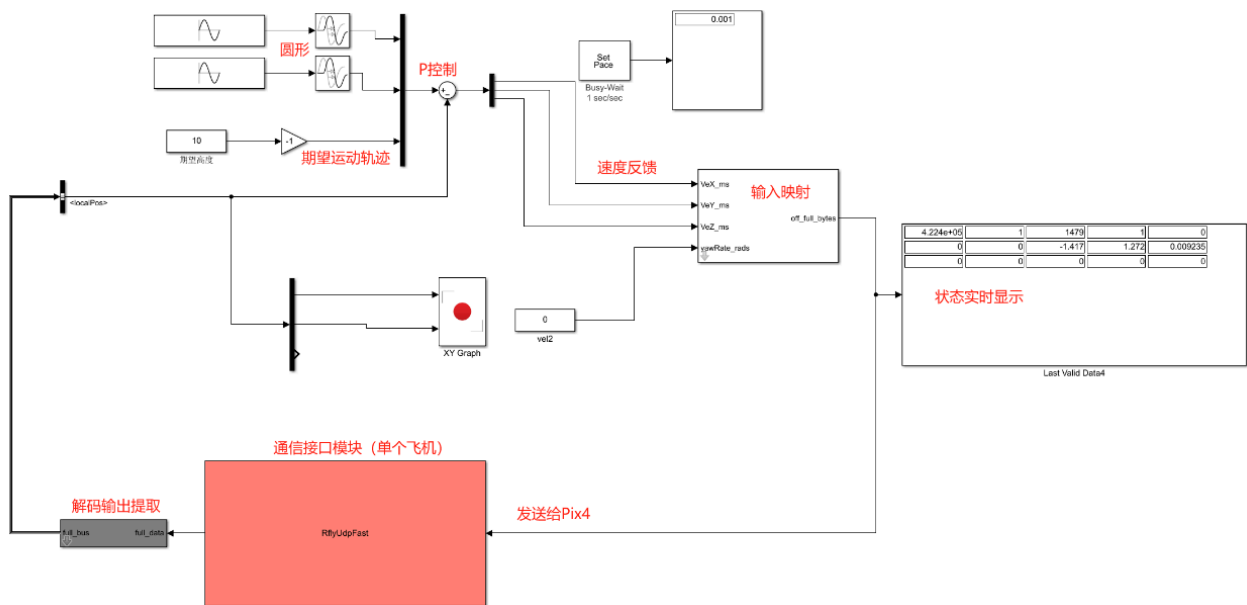


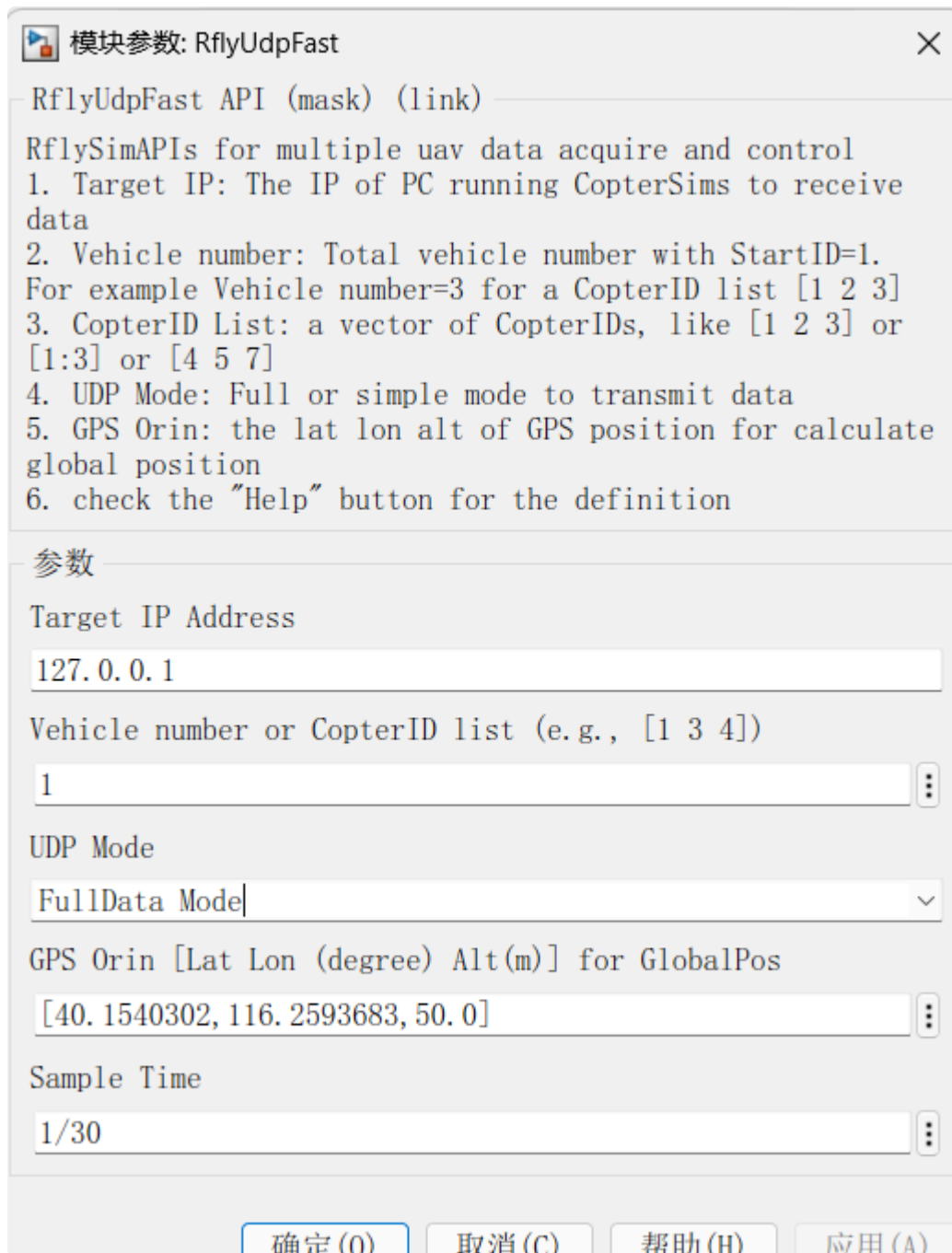
## 6. 参考资料

### 1. 通信接口的FullData模式数据协议

此实验的案例模型中Full\_data\_decoder模块以总线的方式输出为31维的double型向量（全部转发自Pixhawk内部滤波值），具体定义如下：第1~3维：gpsHome[3]; %Home点（上电之后不会变）的经纬高坐标，经纬度需要除以1e7才能得到度为单位的经纬度，高需要除以1e3才能得到m为单位的高（向上为正）；第4~6维：AngEular[3]; %Pixhawk估计得到的姿态欧拉角，单位弧度第7~9维：localPos[3]; %Pixhawk估计得到的以gpsHome为原点的相对北东地位置向量，单位m，z轴向下为正；第10~12维：localVel[3]; %北东地的运动速度向量，单位m/s；第13~15维：GpsPos[3];

%实时的GPS位置，单位和gpsHome相同，但是会实时变化；第16~18维： GpsVel[3];  
 %GPS速度，需要除以100得到m/s为单位的的速度；第19维： time\_boot\_ms； %上电时间；第20维： copterID;  
 %飞机ID（填1即可，目前没有使用）；第21维： relative\_alt=u(21); % GPS相对高度，需要除以1000得到m为单位的高度，向上为正；第22维： hdg=u(22); % GPS航向角，需要除以1000得到0~360度范围的角度；第23维： satellites\_visible=u(23);  
 %可见卫星数量；第24维： fix\_type=u(24); %定位精度；第25维： resrvelnit=u(25); % int类型的保留位；第26维： pos\_horiz\_accuracy=u(26); %水平定位精度，单位m；第27维： pos\_vert\_accuracy=u(27); %竖直定位精度，单位m；第28维： resrveFloat=u(28); % float型保留位，未被启用；第29~31维： globalPos=u(29:31); % CopterSim全局位置，单位m。根据飞机的GPS坐标解算得到，以Ue地图的坐标中心为原点，使用多机控制时，应该使用本接口。





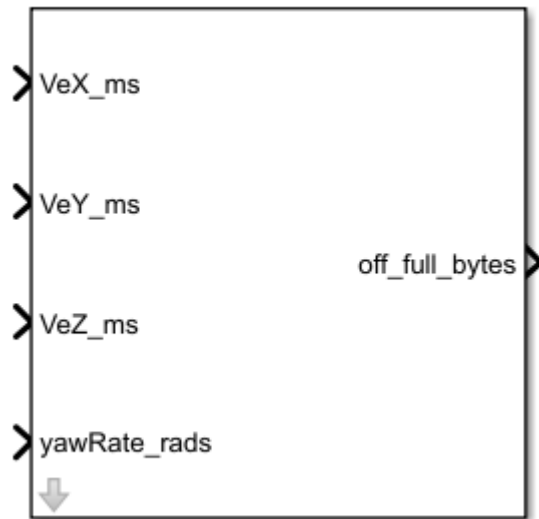
## 2. 通信接口Offboard\_full数据协议

此实验的案例模型中Full\_data\_decoder模块可使能载具进入Offboard模式，通过发送的指令(可以是：vx、vy、v z、yawrate)控制载具在Offboard模式下运动。

NED 坐标系下：Vx、Vy、Vz：速度控制 (m/s)，Yawrate：偏航速度控制 (rad/s)。

此实验的案例模型中Full\_data\_decoder模块可使能载具进入Offboard模式，通过发送的指令(可以是：vx、vy、v z、yawrate)控制载具在Offboard模式下运动。

NED 坐标系下：Vx、Vy、Vz：速度控制 (m/s)，Yawrate：偏航速度控制 (rad/s)。



### 3. 帮助页面

(1) 在RflyUdpFast模块点击帮助，可以弹出帮助页面。



## Block Parameters: RflyUdpFast



RflyUdpFast API (mask)

RflySimAPIs for multiple uav data acquire and control

1. Target IP: The IP of PC running CopterSims to receive data
2. Vehicle number: Total vehicle number with StartID=1. For example Vehicle number=3 for a CopterID list [1 2 3]
3. CopterID List: a vector of CopterIDs, like [1 2 3] or [1:3] or [4 5 7]
4. UDP Mode: Full or simple mode to transmit data
5. GPS Orin: the lat lon alt of GPS position for calculate global position
6. check the "Help" button for the definition

使用说明

Parameters

Target IP Address

127.0.0.1

目标电脑IP

Vehicle number or CopterID list (e.g., [1 3 4])

1

飞机数量或飞机ID列表

UDP Mode

FullData Mode

仿真模式

GPS Orin [Lat Lon (degree) Alt(m)] for GlobalPos

[40.1540302, 116.2593683, 50.0]

GPS原点

Sample Time

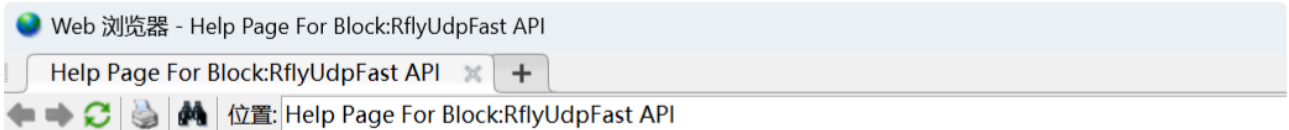
1/30

OK

Cancel

Help

Apply



# RflyUdpFast API

## 1. IP Address

需要控制飞机的CopterSim所在电脑的IP地址，通常情况下，取本机地址“127.0.0.1”即可。如果要在

## 2. Vehicle Number (飞机数量)

Vehicle number飞机数量表示需要连接的CopterSim数量，该模块的输入输出端口数量由该选项控制。

## 3. CopterID List (飞机ID列表)

用一个向量表示，本模块一共仿真多少个飞机（向量长度），每个飞机的序号是多少（向量数值）  
例如，[1:5] 等于 [1 2 3 4 5] 表示前五个飞机，或者[6:10]，或者[1 3 5 6]

## 4. UDP Mode (通信模式)

和CopterSim界面上的“通信模式”下的UDP\_Full和UDP\_Simple相对应。Full模式数据更全，但是数据

### A. FullData完整模式

模块输入为15维的double型向量，具体定义（实现MAVLink的Offboard消息）如下

第1维: time\_boot\_ms; %当前时间戳（填0即可，目前没有使用）

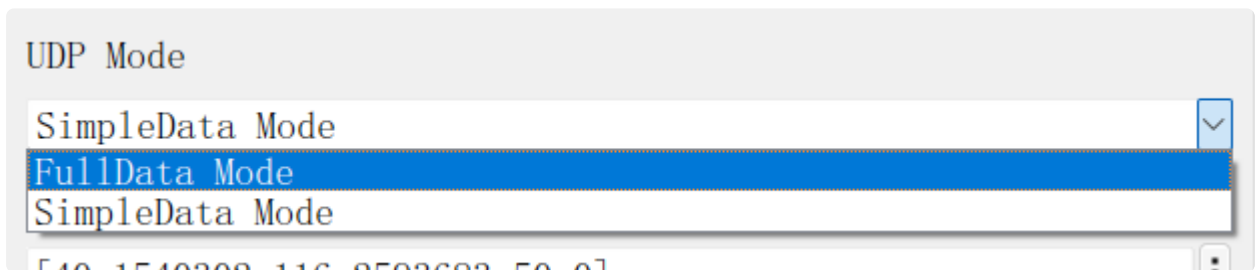
第2维: copterID; %飞机ID（填1即可，目前没有使用）

第3维: type\_mask; %输入控制模式（同Offboard定义）

第4维: coordinate\_frame; %坐标系模式（同Offboard定义）

第5~15维: ctrls[11]; %分别对应了3维的期望位置pos,3维的期望速度vel, 3维的期望加速度acc, 1维

(2) 增加了一个GPS Orin输入项，能够输入GPS原点的坐标，用于获取正确的GlobalPos



## 4. 关键参数讲解

### (1) 初始位置设置

```
23
24 REM Or comment the above code to use the following form
25 SET PosXStr=0,0,0,0,2,2,2,2,4,4,4,4,6,6,6,6
26 SET PosYStr=0,2,4,6,0,2,4,6,0,2,4,6,0,2,4,6
27 SET YawStr=0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
28
```

PosXStr

PosYStr

YawStr

三个逗号分隔的字符串，用于存储一个飞机布局列表。后续，进行软硬件在环仿真时，会依次从本列表中读取数值，并赋值给CopterSim的初始xy位置和偏航角的输入框。



注意：PosXStr、PosYStr、YawStr支持输入浮点数。

## (2) 修改飞机数量

主要针对RflyUdpSimpleFour.bat文件，里面的SET /A VehicleNum对应了启动的软件在环仿真的飞机数。

```
5  
6 SET /A VehicleNum=1
```

## (3) 修改三维样式

主要是使用SET /a

CLASS\_3D\_ID这个语句，默认值-1，会使用DLL或CopterSim里面内置的三维样式ID，如果这里指定了值，会使用本值发送给RflySim3D，从而显示指定的飞机样式。

注意：使用 样式ID+子样式ID\* 100000

的样式，可以设置子样式。例如，100003，表示3号样式（四旋翼）的低1个子样式（对应DJI机架）

## (4) 修改通信模式

```
SET UDPSIMMODE=UDP_FULL
```

或者

```
SET UDPSIMMODE=UDP_Simple
```

要根据当前的Simulink模块使用的数据协议，来对应修改bat里面的通信协议。总体来说，Full模式的数据更快、更全，需要占用较高带宽，适合小规模仿真，例如10个以下；simple模式的数据量小，适合大规模集群仿真，例如20个以上

## (5) 修改初始飞机ID号

```
SET /a START_INDEX=1
```

例如，我要仿真10到20号飞机，需要设置START\_INDEX=10，以及VehicleNum=11

#### (6) 修改地图和DLL模型等

修改DLL模型的名字，如果为0，则使用CopterSim的模型，如果是字符串，则搜索.dll名字  
的模型。

```
set DLLModel=0
```

使用指定的地图名字

```
SET UE4_MAP=Grasslands
```

#### (7) 通过开启多个bat脚本，来构建多种不同载具，不同ID的仿真场景

原则上，一个bat脚本只能启动一种类型的飞机，且需要连续数量。如果需要开启不同类型的载具，需要创建多个bat脚本，分别启动（或者再创建一个bat脚本，自动调用几个子bat脚本）

在修改脚本时，一定要避免后开的脚本，将一些软件关闭掉，或者重复开启QGC等软件。

例如：

```
110
111
112 REM QGCPath
113 tasklist|find /i "QGroundControl.exe" || start %PSP_PATH%\QGroundControl\QGroundControl.exe -noComPix
114 ECHO Start QGroundControl
115
116 REM UE4Path
117 cd /d %PSP_PATH%\RflySim3D
118 tasklist|find /i "RflySim3D.exe" || start %PSP_PATH%\RflySim3D\RflySim3D.exe
119 choice /t 5 /d y /n >nul
120
121
122 tasklist|find /i "CopterSim.exe" && taskkill /im "CopterSim.exe"
123 ECHO Kill all CopterSims
124
```

上面的脚本，就需要在后启动的脚本中删除掉。其余情况，请搜索taskkill来确认需要的软件，不会被自动关闭。

#### (8) 修改GPS原点坐标（对应QGC的GPS原点，或者UE Cesium全球大场景下的GPS原点）

```
SET isPosGps=0 # 如果设置为1，则PosXStr
```

、PosYStr、YawStr需要给GPS坐标值，而不是xyyaw的值。

```
REM set isBatLLAOrin to 1 if use LatLongAlt instead of LLA in model or map/txt
```

```
SET isBatLLAOrin=0 #
```

如果设置为1，则使用后文LatLongAlt的值，作为GPS原点；否则使用CopterSim自行决定的GPS原点值（png  
txt里面定义的，或者模型内部定义的）

REM use LatLongAlt to the GPS origin if isBatLLAOrin is set to 1

SET LatLongAlt=40.1,116.2,50 # GPS坐标原点的值

一组典型设置如下

SET PosXStr=40.1233

SET PosYStr=112.321

SET YawStr=0

SET isPosGps=1 # 让PosXStr使用GPS坐标来初始赋值

SET isBatLLAOrin=1 # 使用bat脚本内LatLongAlt的值设置GPS原点

SET LatLongAlt=40.1,116.2,50 # GPS原点坐标，会发送给QGC和UE

SET

ComNum=0表示1个飞机，串口号为0。CopterSim内部如果串口号设置为0，会遍历所有串口，找到可连接的飞控。

SET

ComNum=0,0,0,0表示4个飞机，串口号都为0，会自动连接串口。按时序的话，会依次连四个飞控。

## 7. 常见问题

Q1: \*\*\*

A1: \*\*\*