

1. 实验名称及目的

1.1 实验名称

视觉盒子硬件在环仿真（基于网口穿环）

1.2 实验目的

实现基于网口方式的视觉盒子硬件在环仿真，控制飞机进行穿环。

1.3 关键知识点

本实验主要是实现通过Python接口ReqCopterSim.py使用（见RflySimAPIs\RflySimSDK\ctrl目录）自动获取的IP，去建立远端电脑与本机RflySim3D（发送图片）与CopterSim（接收控制指令）的联机仿真，同时通过ROS订阅得到图像数据, 使用mavros 去控制飞机，C++实现穿环。关键代码解析如下：

本例子和其他分布式例子的区别，主要在于进行了更加复杂的控制，结合opencv库，实现飞机穿环功能，整个流程主要是启动mavros，开启图像读取，从图像中选点作为飞机穿环的位置，从而最终实现穿环。

关键知识点1: SendProtocol[0]决定了图像的传出模式。SendProtocol[0]=0：共享内存（仅限Windows下获取图像），1：UDP直传png压缩，2：UDP直传图片不压缩（只适用图片类传感器），3：UDP直传jpg压缩（只适用图片类传感器）。如果是激光雷达数据只有0或1（共享内存和UDP网络传输）。

关键知识点2: 通过ReqCopterSim可以自动从局域网获取到仿真电脑的IP地址，从而自动建立连接，不再需要手动指定IP地址。不过，此种连接方式，可能在局域网中产生干扰（多台电脑同时打开多个CopterSim会产生误识别），不适合多个实验同时进行的场景。

1) 视觉接口使用

```
1 vis.jsonLoad(3) # 使用jpeg压缩方式加载Config.json中的传感器配置文件
2 isSuss = vis.sendReqToUE4() # 向RflySim3D发送取图请求, 发给ip为TargetIP的地址
3 vis.startImgCap()# 启取图
4 vis.hasData[i] # 图片i数据是否更新
5 vis.Img[i] # 图片i数 (像素矩阵)
6 cv2.imshow('Img'+str(i),vis.Img[i]) # 显示图片i图像
```

2) ReqCopterSim接口使用 (自动获取ip接口)

```
1 req = ReqCopterSim.ReqCopterSim() \# 获取局域网内所有CopterSim程序的电脑IP列表
2 req.sendReSimIP(1) \# 请求mavlink数据到本电脑
3 req.sendReSimUdpMode(1,2) \# 强制切换MAVLINK_FULL
4 coptersim_ip = req.getSimIpID(1) \#自动获取CopterSim的1号程序所在电脑的IP, 作为目标IP。这里获
5 vis = VisionCaptureApi.VisionCaptureApi(coptersim_ip) \#创建一个视觉传感器实例, 这个实例对
```

3) 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:

```
1 "SeqID":0 : 使用自动更新ID的方式, 创建了SeqID为0的视觉传感器
2 "TypeID":1 : 传感器类型为RGB彩色图像
3 "TargetCopter":1 : 相机绑定在1号飞机上
4 "SendProtocol": [1,0,0,0,0,0,0] : 传输模式为1 : UDP网络传输模式 (图片使用jpeg压缩, 点云直传)。
5 "SensorPosXYZ": [0.3,0,0] : 相机分布位置。
```

2.实验效果

无

3.文件目录

例程目录:

[\[安装目录\]\RflySimAPIs\8.RflySimVision\0.ApiExps\0.Preparation\3.NXwithPX4Config\2.Pixhawk6xNetSim](#)

文件夹/文件名称	说明
PX4+NX_SITL	启动软件仿真配置文件

文件夹/文件名称	说明
PX4+NX_HITL	启动硬件仿真配置文件
Config.json	视觉传感器配置文件
server_ue4_Serial	Python实验程序

4. 运行环境

4.1 软件要求

Windows 10及以上版本；RflySim工具链；Visual Studio Code；Linux（Ubuntu 20.04）。

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；视觉盒子（内置NX+飞控） 1台。

①：推荐配置请见：<https://rflysim.com/doc/zh/HowToInstall.pdf>

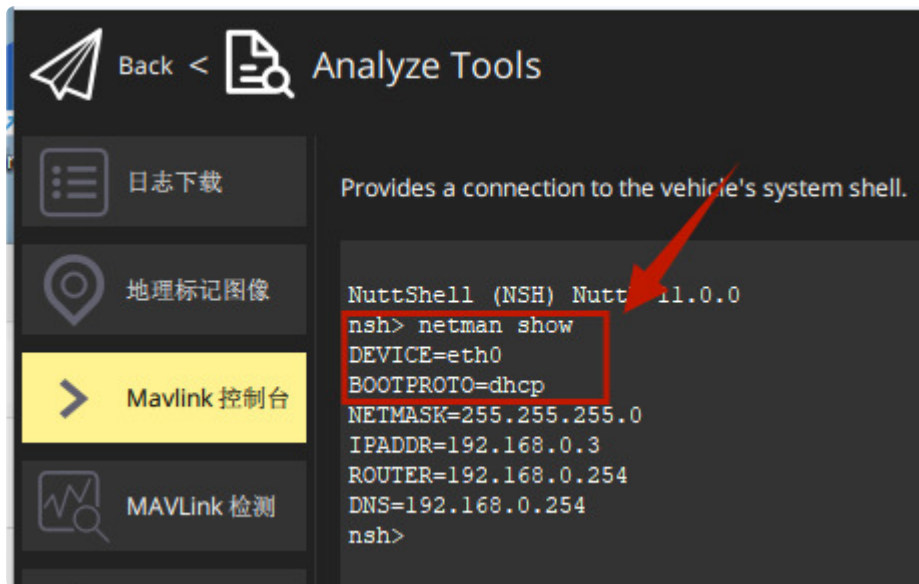
5. 实验步骤

5.1 必做实验：基于网口方式的硬件在环穿环实验（仅rflysim平台完整版可做）

1. 查看视觉盒子中的NX和飞控是否连接，如果未连接，按照5.1实验中将NX和飞控连接。
2. 首先插上视觉盒子的电源线，按下飞控开关和智能开关，使用usb数据线连接飞控USB和电脑。



打开QGC，按照5.1步骤完成固件烧录，机架选择，参数设置（如果这三个步骤已经做了可以跳过）。接着进入Analyze Tools – Mavlink 控制台页面，输入 netman show 命令，查看当前的网口配置。确认连接协议 BOOTPROTO 是否为dhcp。注意：后面的IP地址不重要，在 dhcp 模式下以路由器上显示为准。



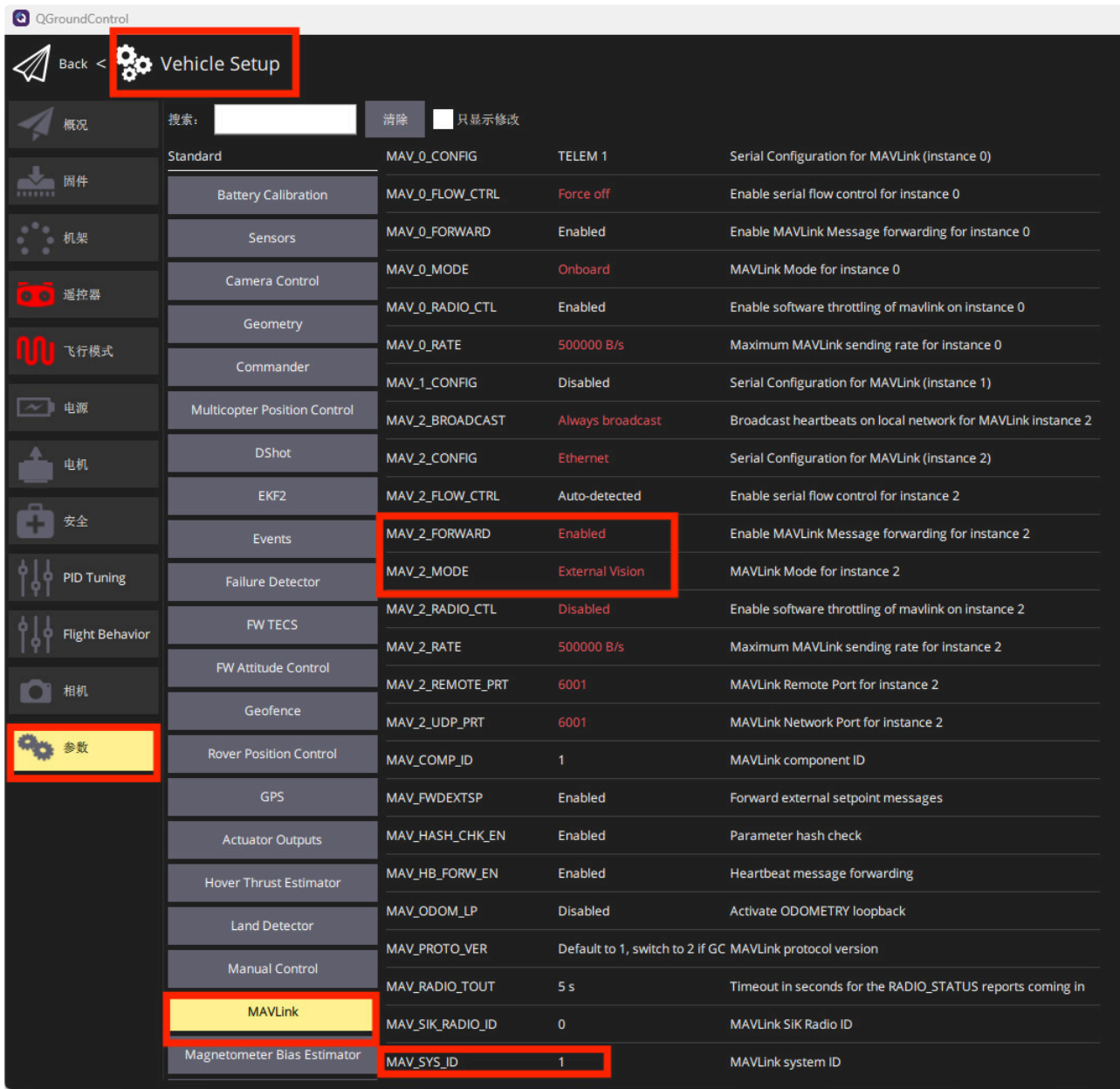
3. 如果不是 dhcp 模式，则执行下面命令，重启飞控，再次输入 netman show，看是否已经改过来。

```
echo BOOTPROTO=dhcp \>\> /fs/microsd/net.cfg
```

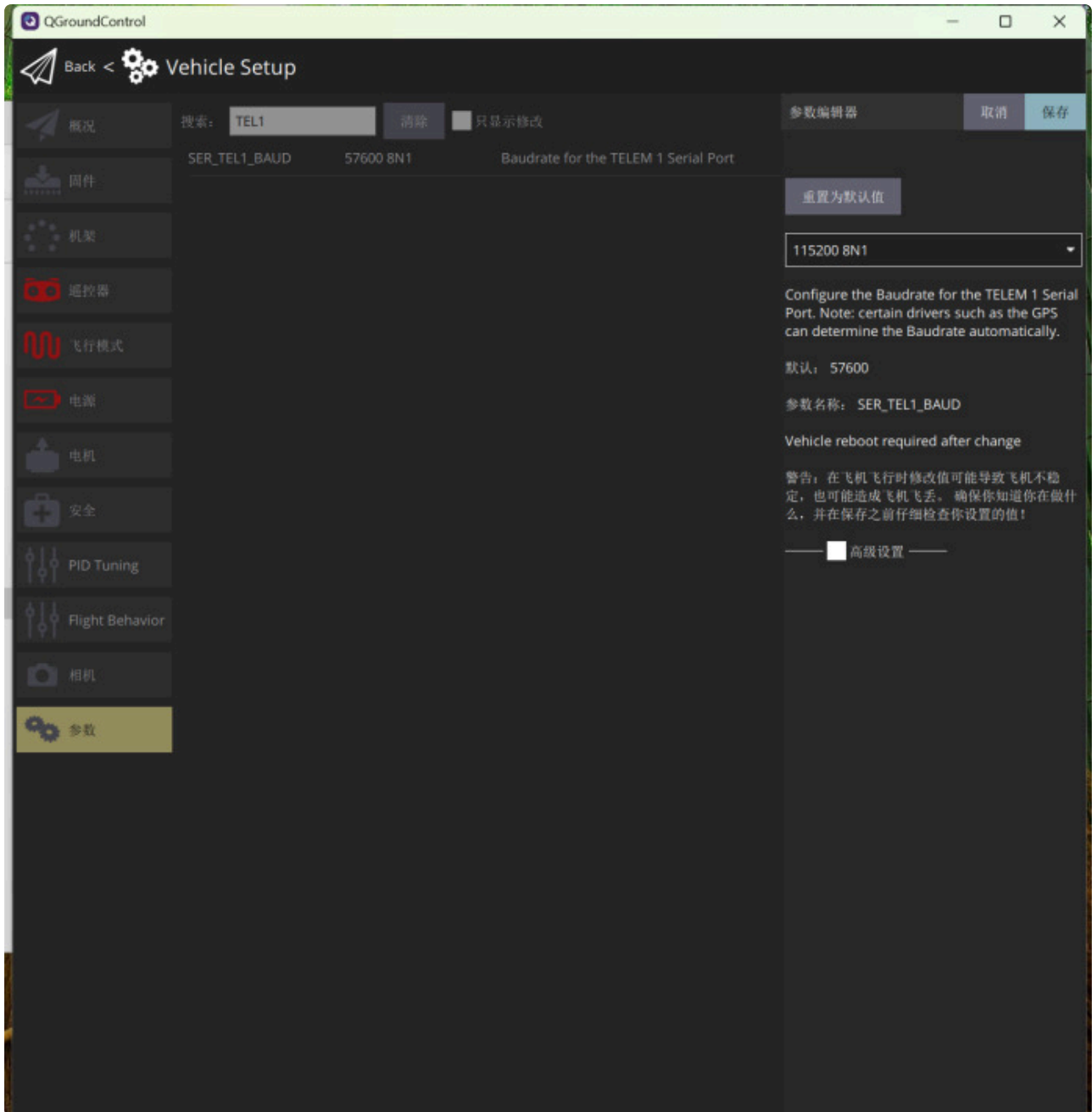
4. 进入Vehicle Setup-MAVLINK 页面，按下图修改MAV_2_MODE=External_Vision，REMOTE_PRT和 UDP_PRT=6001 几个参数。

注意：REMOTE_PRT、UDP_PRT 和 MAV_SYS_ID 是匹配的，规则是 6000+MAV_SYS_ID 的格式。如果是 2 号飞机，需要设MAV_SYS_ID 为 2，同时设置 REMOTE_PRT、UDP_PRT为 6002

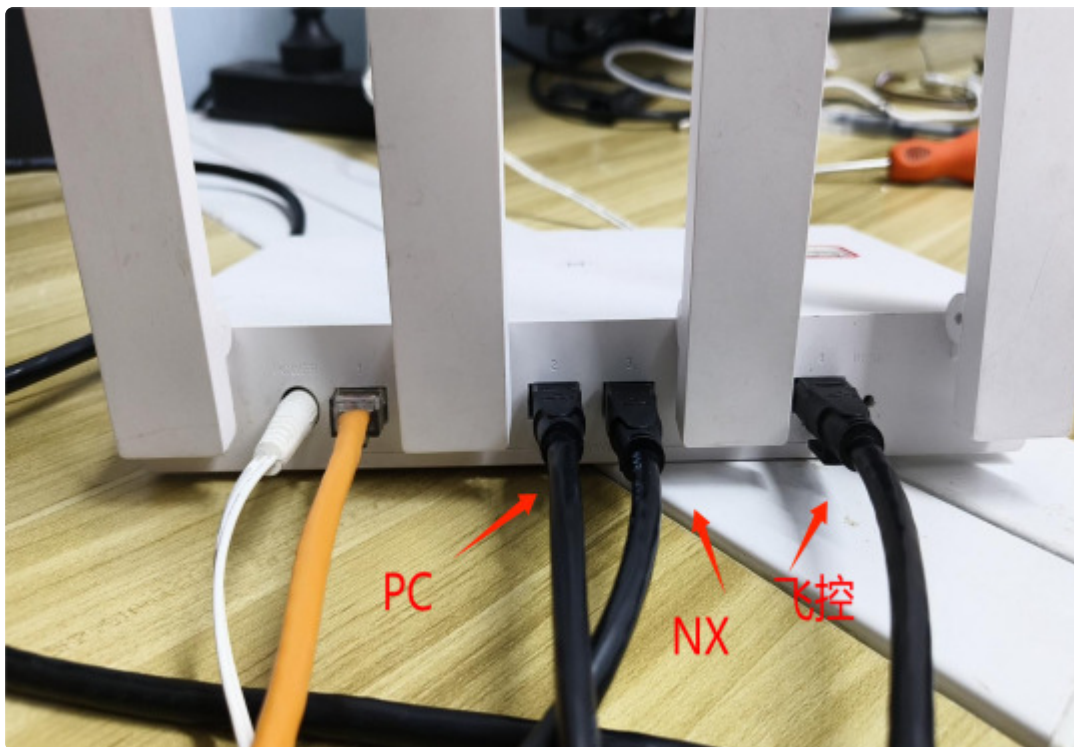
注意：MAV_2_MODE 设置为 Onboard 时，没法连接 QGC，因此使用 External_Vision。



5. 选择参数界面，搜索TEL1，将波特率更改为115200



6. 单独准备一个路由器（便于观察设备 IP 地址，尽量不要用公司或学校的大路由器不便于观察，且可能存在网络阻塞），让电脑和飞控上的网口，插在同一个路由器下，飞控重新上电，确保网口开始连接网络。如下图所示。



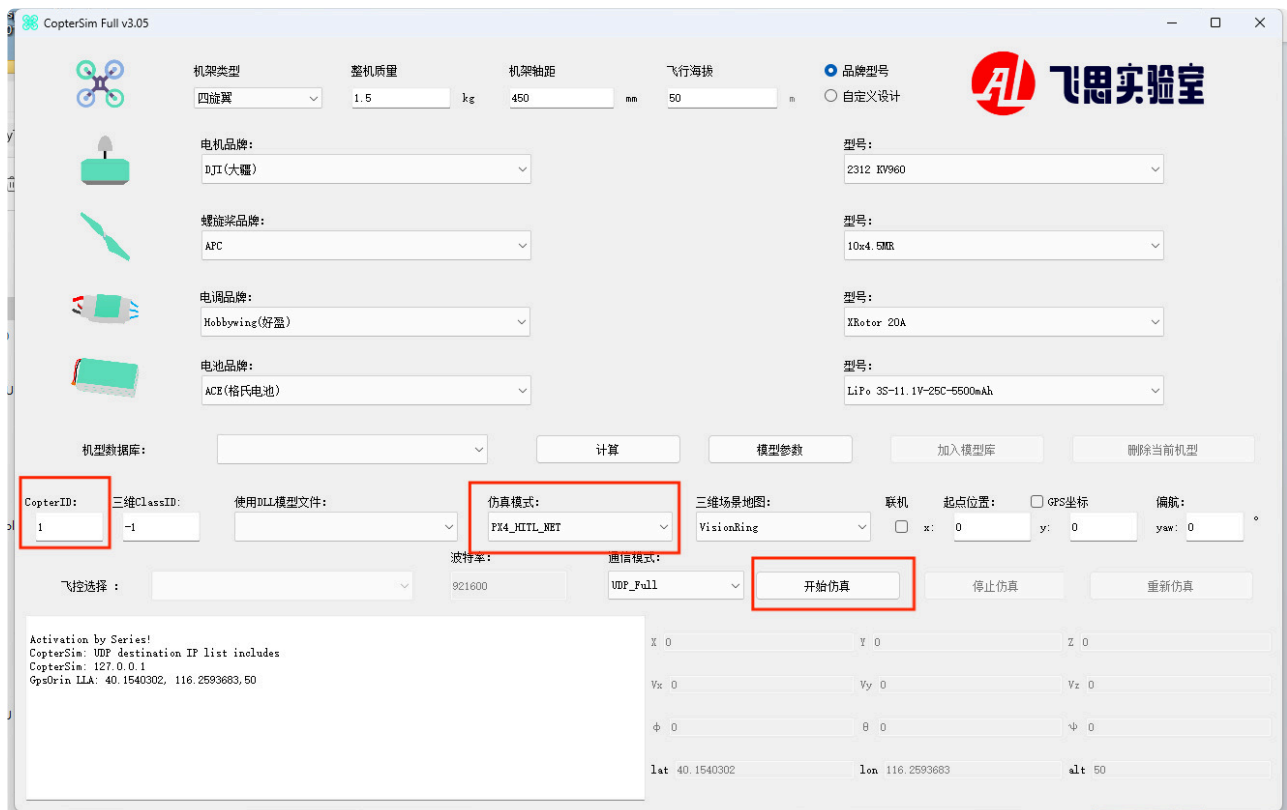
7. 登录路由器后台（密码可以在路由器上找到或咨询管理员），确认是否可以看到飞控的 IP 地址，如下图所示。

有线连接设备	允许上网	网络限速	限速值
 nvidia-desktop <input checked="" type="checkbox"/> ↑ 0 Kbps ↓ 0 Kbps IP 地址: 192.168.4.6 MAC 地址: 48:B0:2D:F9:2FA5 在线: 刚刚	NX	<input type="checkbox"/>	↑ 不限速 ↓ 不限速
 DESKTOP-2SR1JH9 <input checked="" type="checkbox"/> ↑ 0 Kbps ↓ 0 Kbps IP 地址: 192.168.4.3 MAC 地址: 08:26:AE:35:87:DA 在线: 0天5小时45分钟	PC (自己)	<input type="checkbox"/>	↑ 不限速 ↓ 不限速
 nuttx <input checked="" type="checkbox"/> ↑ 0 Kbps ↓ 0 Kbps IP 地址: 192.168.4.7 MAC 地址: 1E:B0:D7:83:79:DB 在线: 刚刚	飞控	<input type="checkbox"/>	↑ 不限速 ↓ 不限速

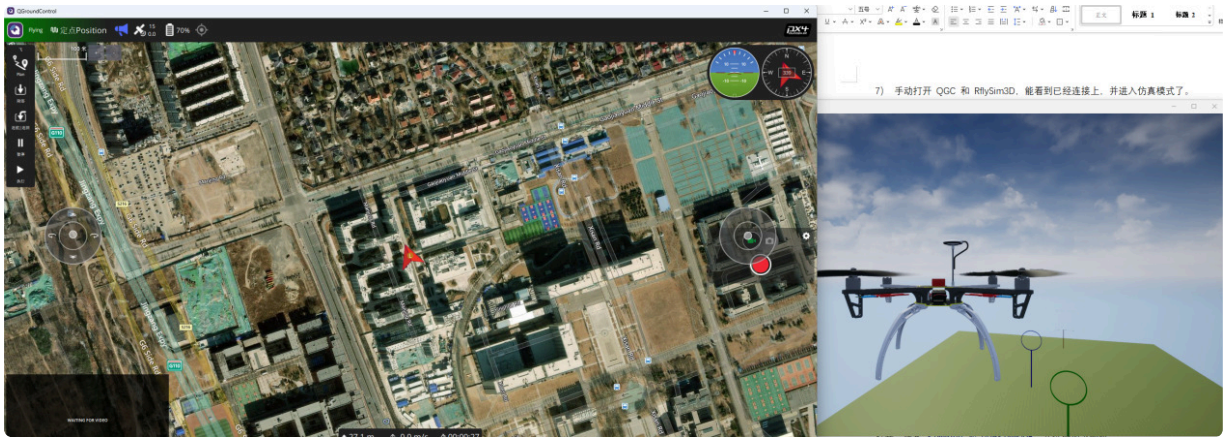
8. 现在需要验证网口方式的硬件仿真配置是否正确

第一种方式：手动方式

1. 首先断开飞控上的 USB 线，保持飞控网口连接状态，打开CopterSim，选择仿真模式“PX4_HITL_NET”模式，确认本机 ID 与上面配置的 Sys_ID相匹配（本例都选设置为 1），然后点击开始仿真。

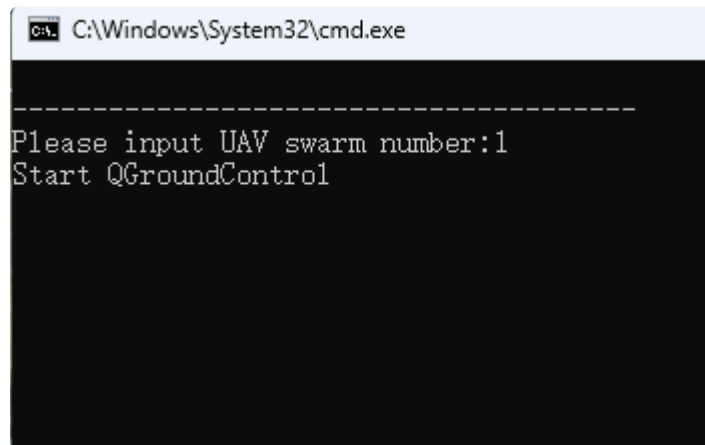


2. 手动打开 QGC 和 RflySim3D，能看到QGC已经连接上，并进入仿真模式了。在 QGC 中，可以控制飞机起飞，说明配置正确。

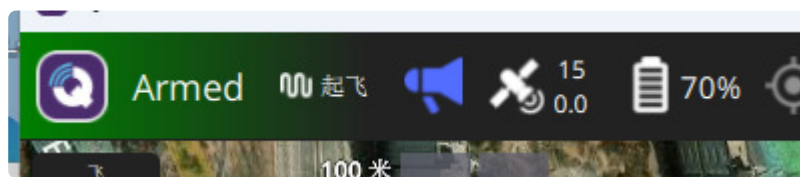


第二种方式：bat脚本的方式

1. 打开readme所在的目录下的 `HITL_NET_Run.bat` 脚本，并输入1, 1表示一台无人机，可以看到自动打开了QGCrflySim3D, CopterSim。



2. 可以看到QGC连接到了飞控，CopterSim也完成了仿真，QGC也能操控飞机，说明配置正确。



```

PX4: failsafe mode deactivated
PX4: Data link regained
PX4: Command ID: 512 ACCEPTED
PX4: Command ID: 512 ACCEPTED
PX4: Manual control regained after 526.4s
PX4: Enter Manual Mode!
PX4: Command ID: 512 ACCEPTED
PX4: Command ID: 401 UNSUPPORTED
PX4: GPS 3D fixed & EKF initialization finished.

```

3. 将文件夹 1.FeisiVisionBox 拷贝到NX中，并进入视觉盒子中NX的远程中，打开一个终端输入如下命令给相应的通信端口赋权限

```
sudo chmod 777 /dev/ttyTHS0
```

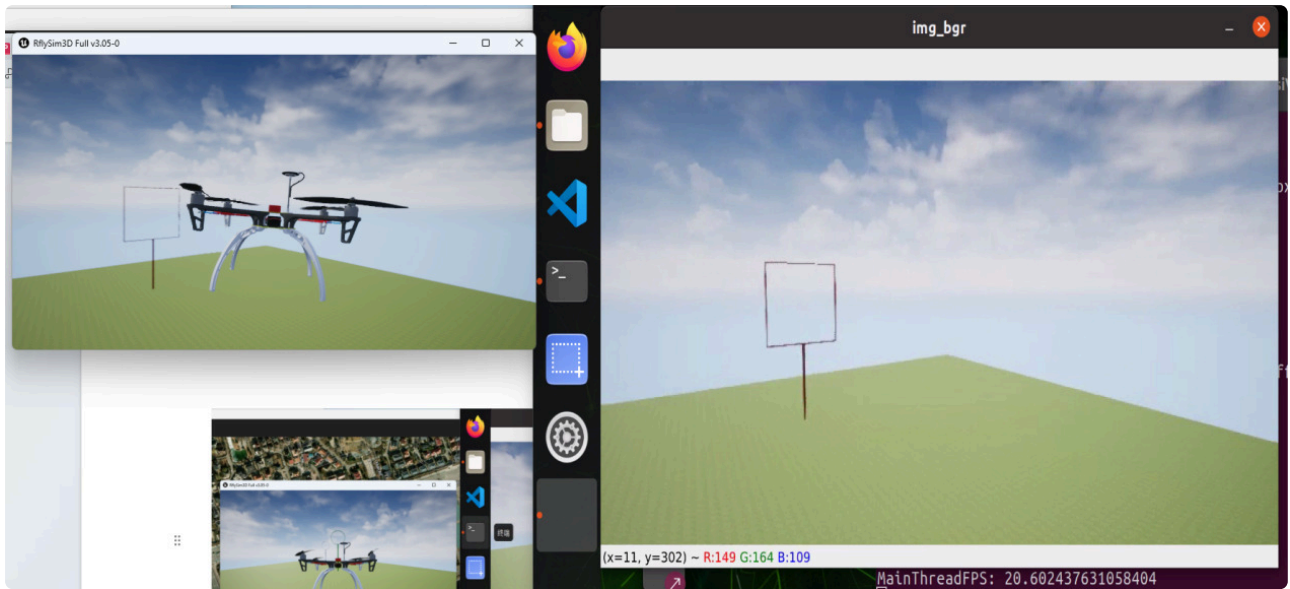
注：'/dev/ttyTHS0' 是用于飞控与 NX 通信的端口，其不一定是 THS0，但多数情况下为 THS0，但都需要对其进行赋权限。

另起一个终端输入如下命令

令 `roslaunch mavros px4.launch fcu_url:=/dev/ttyTHS0:115200`，再起一个终端输入如下命令 `rostopic echo /mavros/state`，查看数据是否在更新，若数据在更新，则正常，若未更新则需要检查端口是否正确，又或者视觉盒子内部接线是否正常。

```
secs: 1723106320
nsecs: 911943836
frame_id: ''
connected: True
armed: True
guided: False
manual_input: False
mode: "MANUAL"
system_status: 8
---
header:
  seq: 32
  stamp:
    secs: 1723106321
    nsecs: 901589070
  frame_id: ''
connected: True
armed: True
guided: False
manual_input: False
mode: "MANUAL"
system_status: 8
---
```

1. 在1.FeisiVisionBox目录下打开一个终端，运行 `python3 server_ue4.py`，可以看到飞机起飞并进行穿环。



6.参考资料

无

7.常见问题

Q1: 无

A1: 无