

1. 实验名称及目的

1.1 实验名称

视觉盒子硬件在环仿真（多盒子联合仿真）

1.2 实验目的

使用两台视觉盒子实现视觉盒子联合仿真，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
3 isSuss = vis.sendReqToUE4() \# 向RflySim3D发送取图请求, 发给ip为TargetIP的地址
4
5 vis.startImgCap() \# 开启取图
6
7 vis.hasData[i] \# 图片i数据是否更新
8
9 vis.Img[i] \# 图片i数据 (像素矩阵)
10
11 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
3 req.sendReSimIP(1) \# 请求mavlink数据到本电脑
4
5 req.sendReSimUdpMode(1,2) \# 强制切换MAVLINK_FULL
6
7 coptersim_ip = req.getSimIpID(1) \#
8 自动获取CopterSim的1号程序所在电脑的IP, 作为目标IP。这里获取CopterSim所在仿真电脑的IP。
9
10 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
3 "TypeID":1 : 传感器类型为RGB彩色图像
4
5 "TargetCopter":1 : 相机绑定在1号飞机上
6
7 "SendProtocol": [1,0,0,0,0,0,0,0] : 传输模式为1 : UDP网络传输模式 (图片使用jpeg压缩, 点云直传)。
8
9 "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
\4.MultiNxBox

文件夹/文件名称	说明
PX4+NX_SITL	启动软件仿真配置文件
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） 2台。

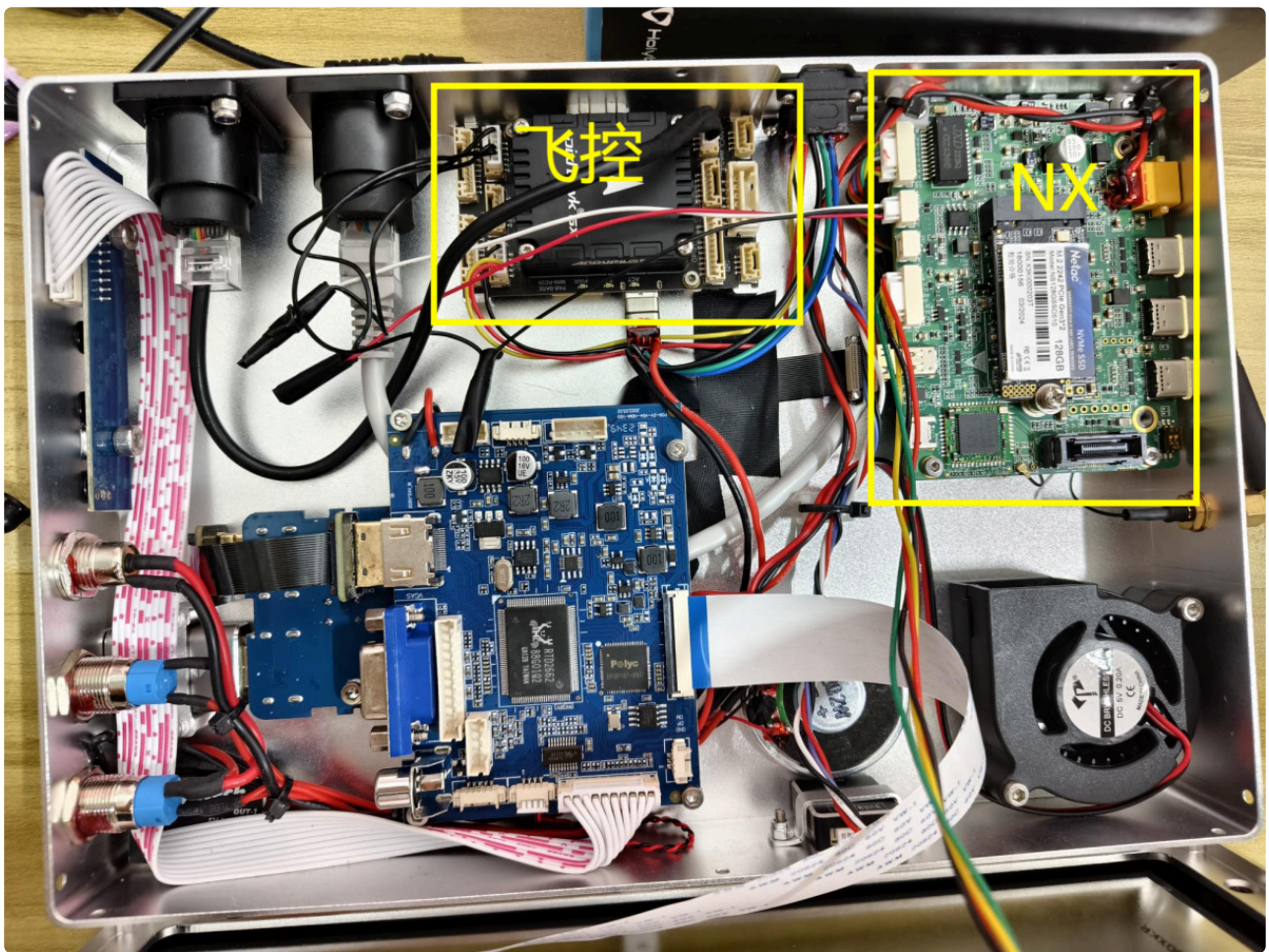
①：推荐配置请见：<https://rflysim.com/doc/zh/HowToInstall.pdf>

5. 实验步骤

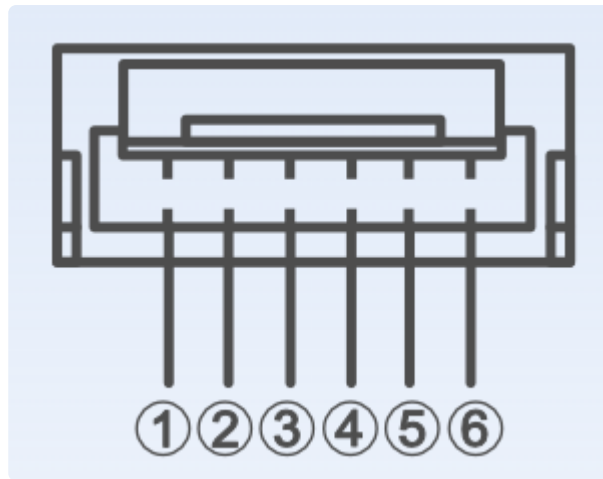
5.1 必做实验：基于串口方式的更改sys_id硬件在环穿环实验

Step 1：视觉盒子基本配置

视觉盒子配置了NX以及Pixhawk 6x，如图所示。



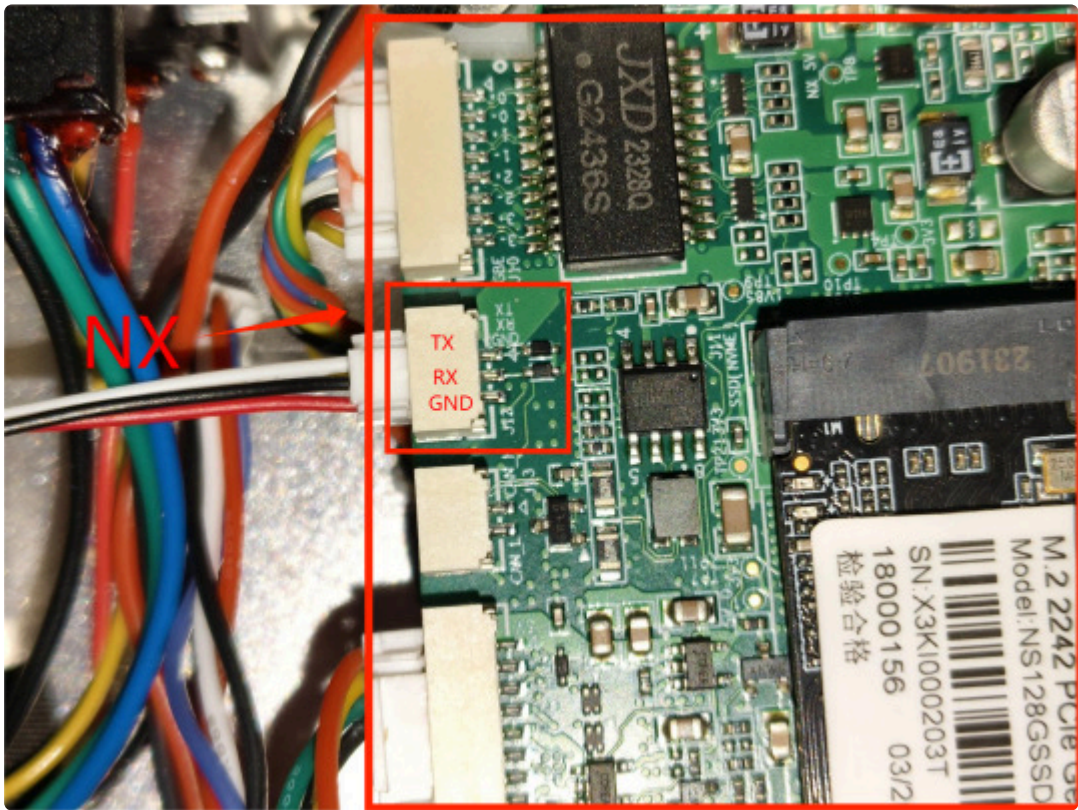
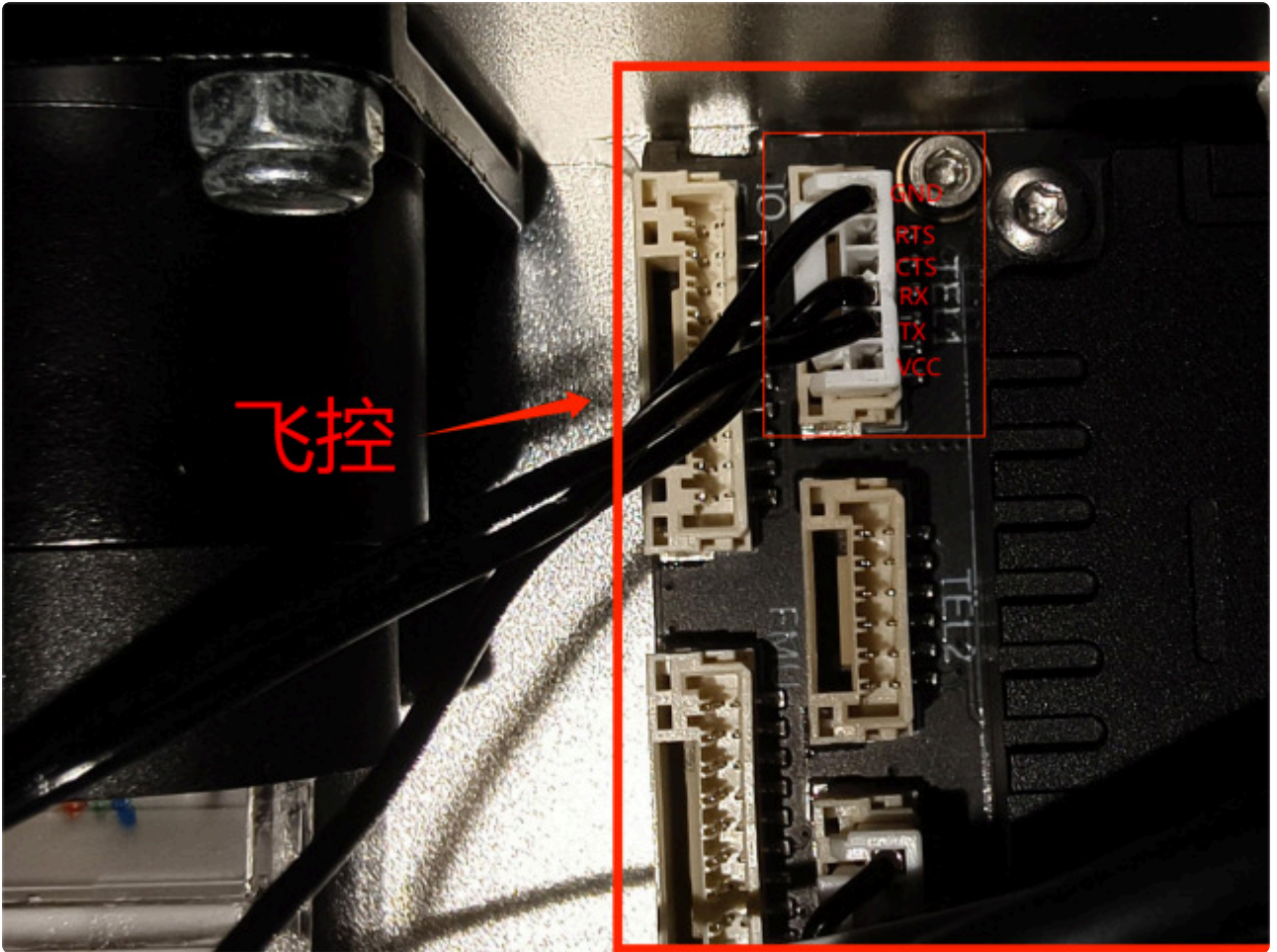
1. 首先将飞控与NX相连，如果没有则需要根据官方文档提供的接口线序自制线材，但是需要注意线序。



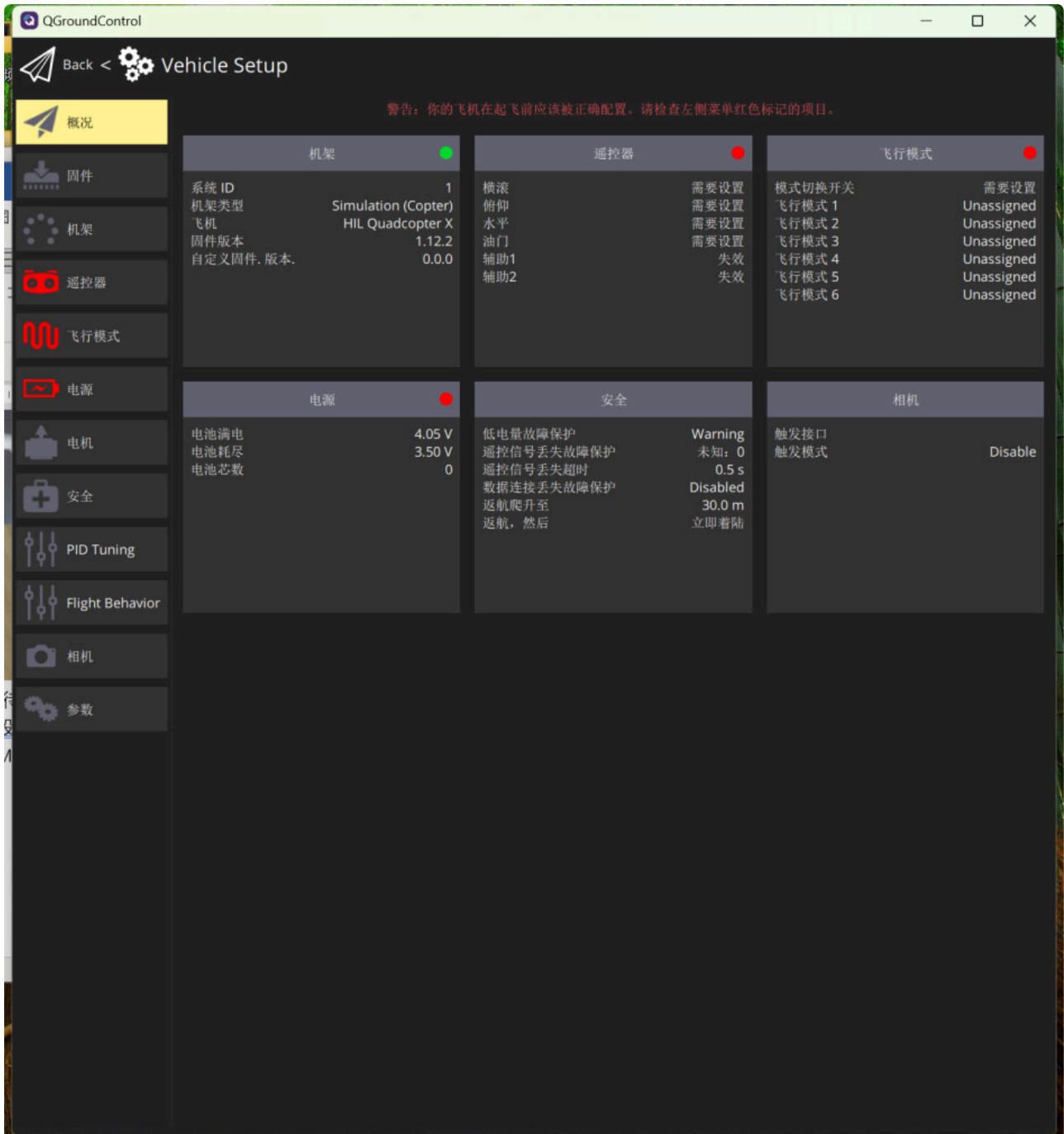
Telem1, Telem2, Telem3 ports

Pin	Signal	Volt
1(red)	VCC	+5V
2(black)	TX7/5/2 (out)	+3.3V
3(black)	RX7/5/2 (in)	+3.3V
4(black)	CTS7/5/2 (in)	+3.3V
5(black)	RTS7/5/2 (out)	+3.3V
6(black)	GND	GND

如上图所示，飞控的TELEM1口从左到右为VCC TX RX CTS RTS GND,本实验采用串口通信的方式将NX与飞控进行连接，因此我们需要将NX的TX接到TELEM1口的RX上，再将NX的RX接到TELEM1口的TX上，最后把GND连接到一起即可。（一定要注意接线顺序若接反将飞控TELEM1口的VCC接到NX的GND上会导致NX损坏）



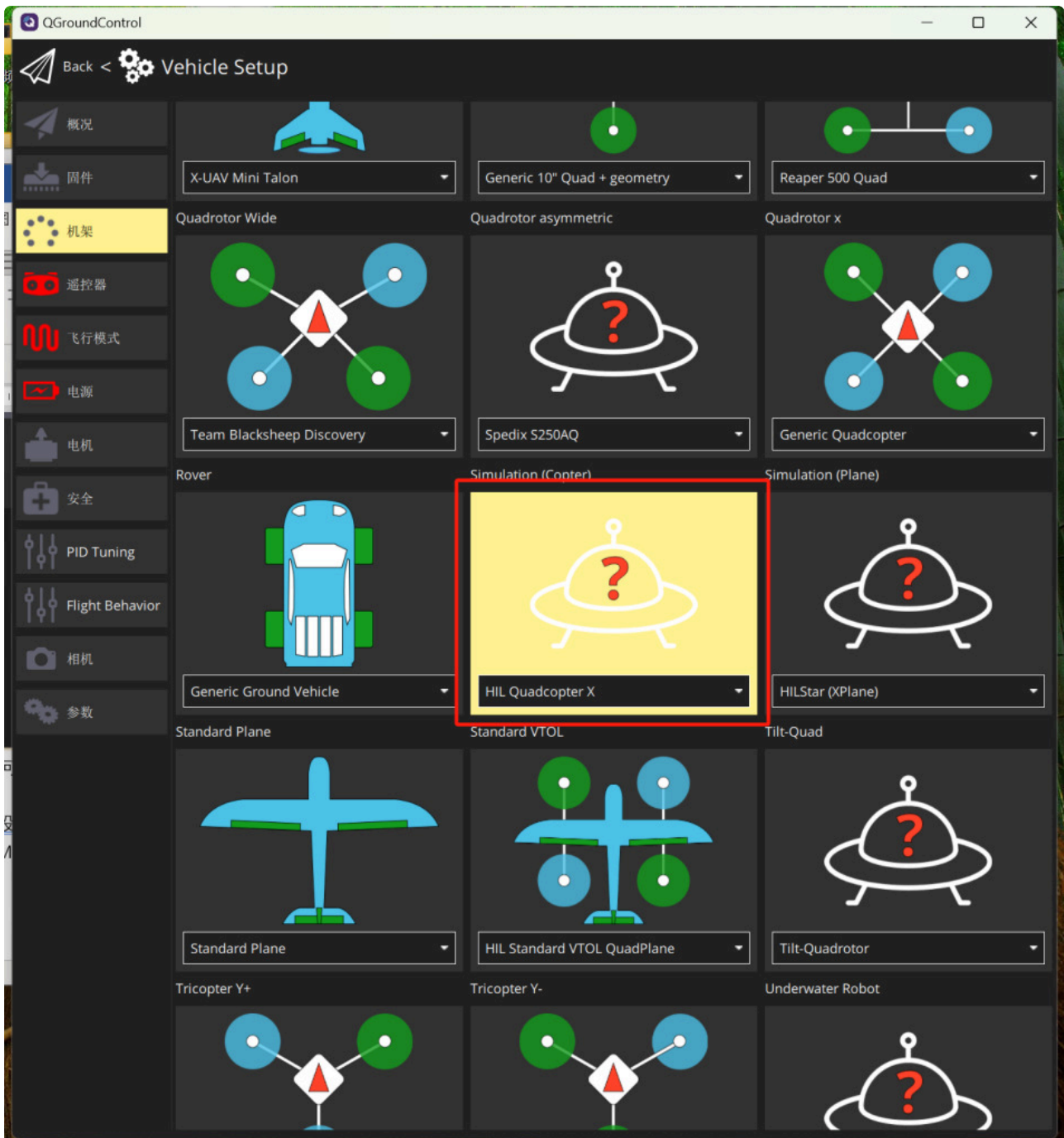
2. 使用USB线连接电脑与飞控的Type-c口并打开QGC会自动跳转到下面界面（如果出现未出现跳转，多次插拔重新连接），注意需要对两台视觉盒子进行同样的设置。



3. 进入固件烧录模式，选择Rflysim 1.13.2 Stable版本。



4. 选择机架，将机架设置为HIL Quadcopter X，设置完成后选择应用并重启，等待QGC重新加载。



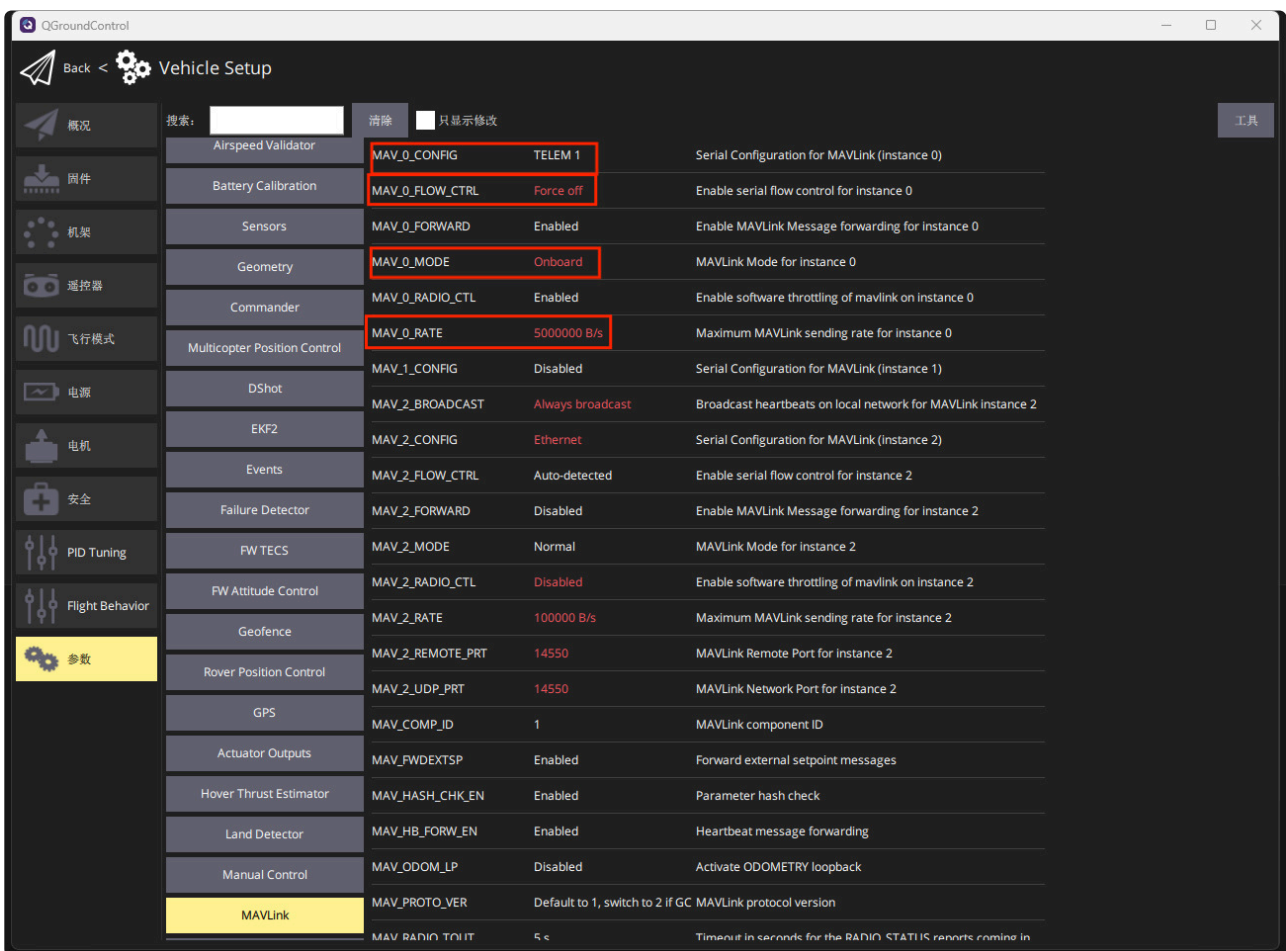
5. 进入参数设置界面的MAVLink控制台设置MAV_0_CONFIG为TELEM1该参数为选择mavlink消息输出的接口。设置MAV_1_FLOW_CTRL: force off否则可能会导致无法连接到机载电脑, 并将MAV_0_MODE设置为Onboard用于和板载计算机通信。

MAV_0_MODE有以下几种模式可选，各模式用途如下：

Normal	用于和地面站通信
Custom	自定义模式，一般是开发者使用

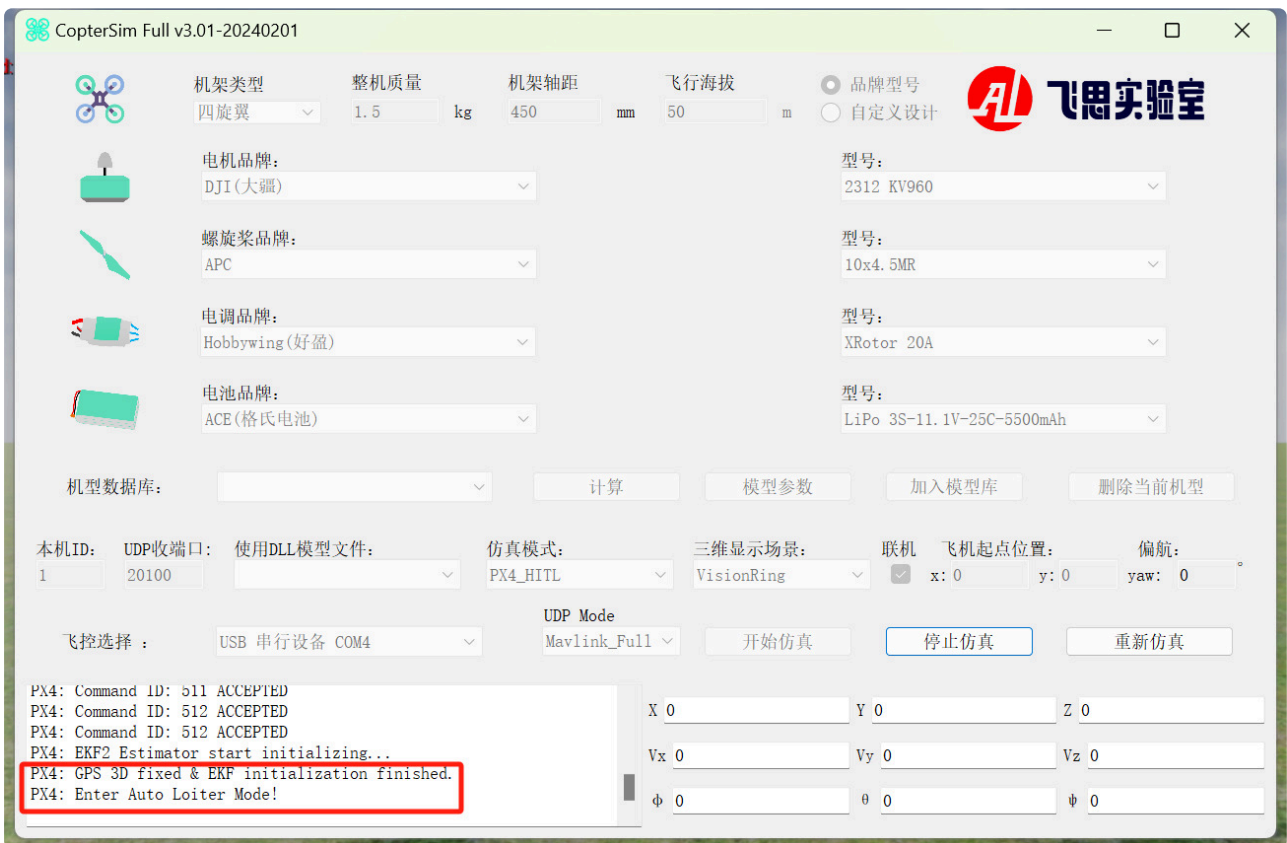
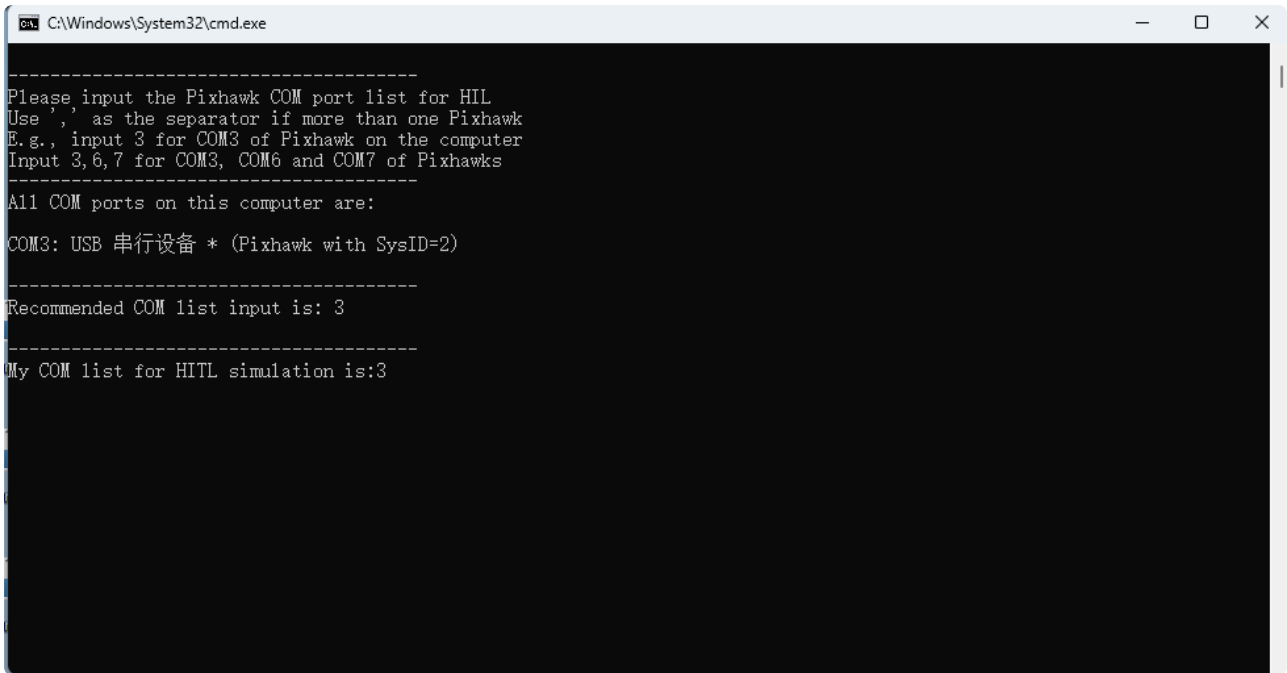
Normal	用于和地面站通信
Onboard	用于和板载计算机通信
OSD	用于OSD显示
Config	用于快速连接（如USB）
Minimal	与地面站通信，通行量最小，适用于高延时的链路
ExtVision or ExtVisionMin	用于板外的视觉里程计数据通信
Iridium	用于铱卫星通信系统

设置MAV_0_RATE为500000, 该参数为通信速率, 如果总的通信速率超过此值, 则会降低单个消息的发送频率, 设置为0将数据速率设置为理论最大值的一半。



6. 打开文件夹

8.RflySimVision\0.ApiExps\0.Preparation\3.NXwithPX4Config\4.MultiNxBat
运行CrossRing3HITL.bat, 输入推荐的COM (注意有两台视觉盒子的串口) 回车打开硬件仿真, 等待CopterSim初始化完成。



7. 在NX系统上新开一个终端，输入如下命令给相应的通信端口赋权限

```
1 | 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。

另开一个终端输入如下命令

```
1 | roslaunch mavros px4.launch tgt_system:=2 fcu_url:=/dev/ttyTHS0:115200
```

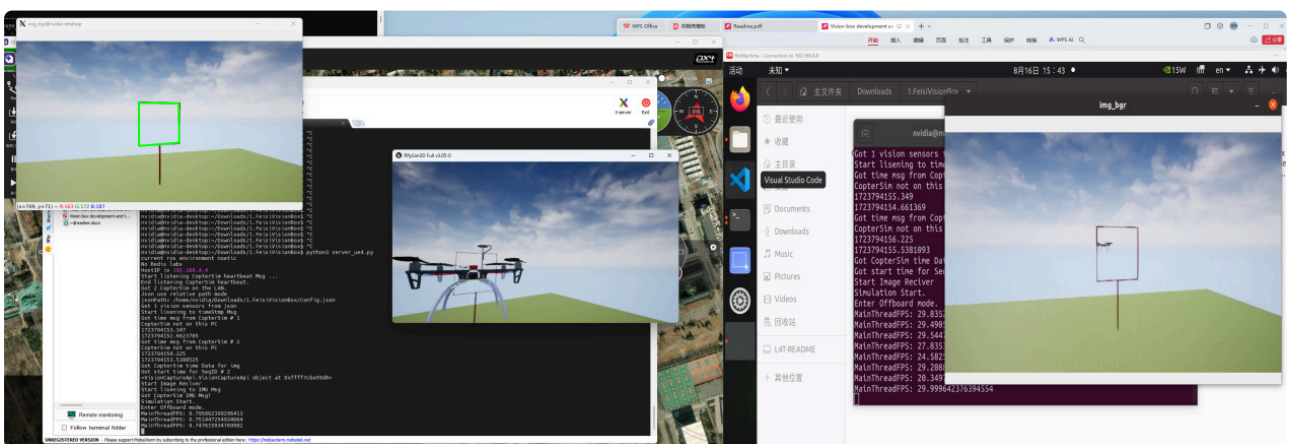
另起一个终端输入如下命令

```
1 | rostopic echo /mavros/state
```

```
header:
  seq: 50
  stamp:
    secs: 1722917074
    nsecs: 491893626
  frame_id: ''
connected: True
armed: False
guided: True
manual_input: False
mode: "AUTO.TAKEOFF"
system_status: 8
```

查看数据是否在更新，以及connected是否为True若数据在更新，则正常，若未更新且connected为False则需要检查端口是否正确。

8. 将 8.RflySimVision\0.ApiExps\0.Preparation\3.NXwithPX4Config 目录下的 4.MultiNxBox拷贝到两台视觉盒子的NX中去，并分别在两台视觉盒子上4.MultiNxBox目录下打开一个终端，一号机所对应的视觉盒子运行 `python3 server_ue4.py`，二号机所对应的视觉盒子运行 `server_ue4_2.py` 可以看到两架飞机起飞并进行穿环。



6.参考资料

无

7.常见问题

Q1: 无

A1: 无