

# 1. 实验名称及目的

## 1.1 实验名称

硬件在环仿真实验

## 1.2 实验目的

实现两路串口通信硬件在环仿真。

## 1.3 关键知识点

首先通过PX4MavCtrl.PX4MavCtrler()创建一个通信接口实例命名为mav，其中在硬件在环仿真中，使用的主机不同需要在PX4MavCtrl.PX4MavCtrler()中设定不同的端口号与通信波特率，其输入格式也有所不同具体如下：

- ID:如果ID $\leq$ 10000则表示飞机的CopterID号。如果ID $>$ 10000，例如20100这种，则表示通信端口号port。按平台规则，`port=20100+CopterID*2-2`（为了兼容旧接口的过渡定义，将来ID只表示CopterID）。
- ip:数据向外发送的IP地址。默认是发往本机的127.0.0.1的IP；在分布式仿真时，也可以指定192.168打头的局域网电脑IP；也可以使用255.255.255.255的广播地址（会干扰网络其他电脑）。
- Com: 与Pixhawk的连接模式。`Com='udp'`，表示使用默认的udp模式接收数据，这种模式下，是接收CopterSim转发的PX4的MAVLink消息（或UDP\_full,simple）消息包。使用port+1端口收和port端口发（例如，1号飞机是20101端口收，20100端口发，与CopterSim对应）。`Com='COM3'`（Windows下）或`Com='/dev/ttyUSB0'`（Linux系统，也可能是ttyS0、ttyAMA0等），表示通过USB线（或者数传）连接飞控，使用默认57600的波特率。注意：波特率使用port口设置，默认port=0，会重映射为57600。`Com='Direct'`，表示UDP直连模式（对应旧版接口的真机模式），这种模式下使用同一端口收发（端口号有port设置），例如`Com='Direct', port=15551`，表示通过15551这一个端口来收发数据。

注意：COM模式和Direct模式下，ID只表示飞机的ID号，而不表示端口号。

`Com='redis'`:使用Redis模式通信，服务器地址为ip，服务器端口为port。

- port:UDP模式下默认情况下设为0，会自动根据IP填充，按平台规则，`port=20100+CopterID*2-2`。如果这里赋值大于0，则会强制使用port定义的端口。COM模式下，Port默认表示波特率`self.baud=port`。如果port=0，则会设置`self.baud=57600`。Direct模式下，Port默认表示收发端口号（使用相同端口）。redis模式下，Port对应服务器端口号`self.redisPort =port`。如果port=0，则`self.redisPort=6379`为平台默认值。然后通过SendPosNED函数发送位置指令，飞机即可完成飞行。

## 2.实验效果

本实验通过NX传输控制命令可以看到当NX的控制程序启动后，位于主机的仿真界面的飞机会根据控制命令飞行。



## 3.文件目录

例程目录：

[\[安装目录\]\RflySimAPIs\8.RflySimVision\0.ApiExps\1-UsageAPI\9.serial\\_connect\\_HITL](#)

文件夹/文件名称	说明
linux	放于NX主机上运行的仿真控制程序

文件夹/文件名称	说明
windows	放于windows主机上运行的程序
<a href="#">HITL.bat</a>	硬件在环仿真启动脚本

## 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。

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；虚拟机/NX 1台。

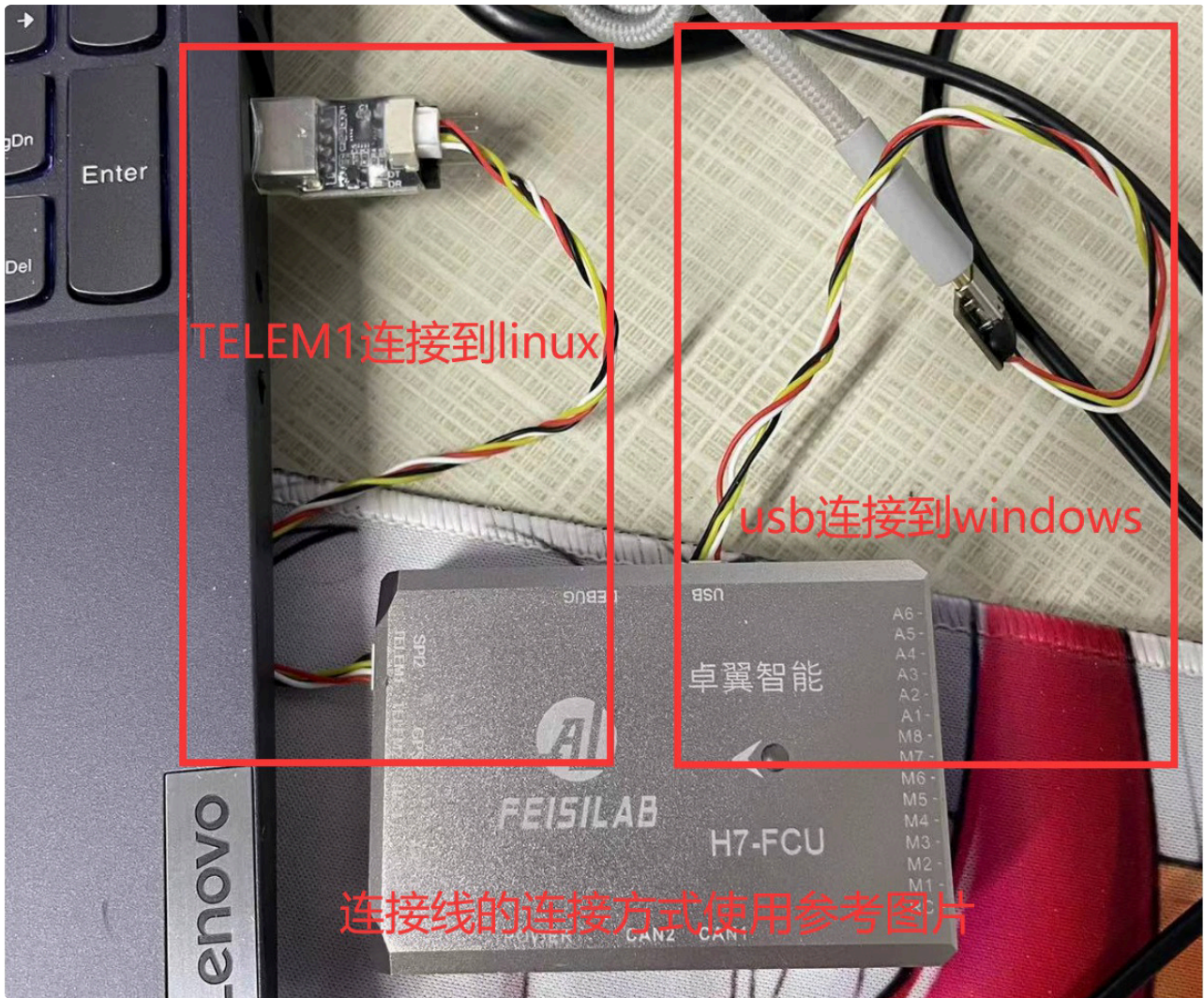
①：推荐配置请见：<https://rflysim.com/doc/zh/HowToInstall.pdf>

## 5. 实验步骤

### Linux下

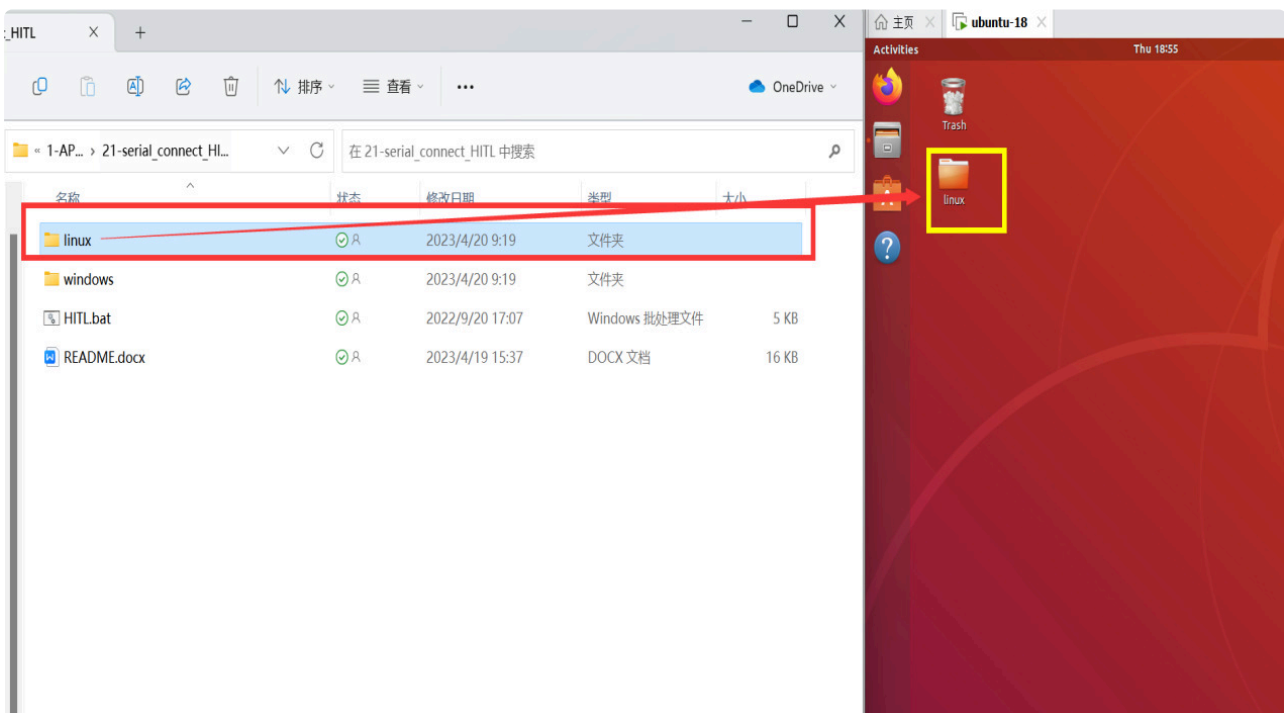
#### Step 1

将设备连接起来，如图。



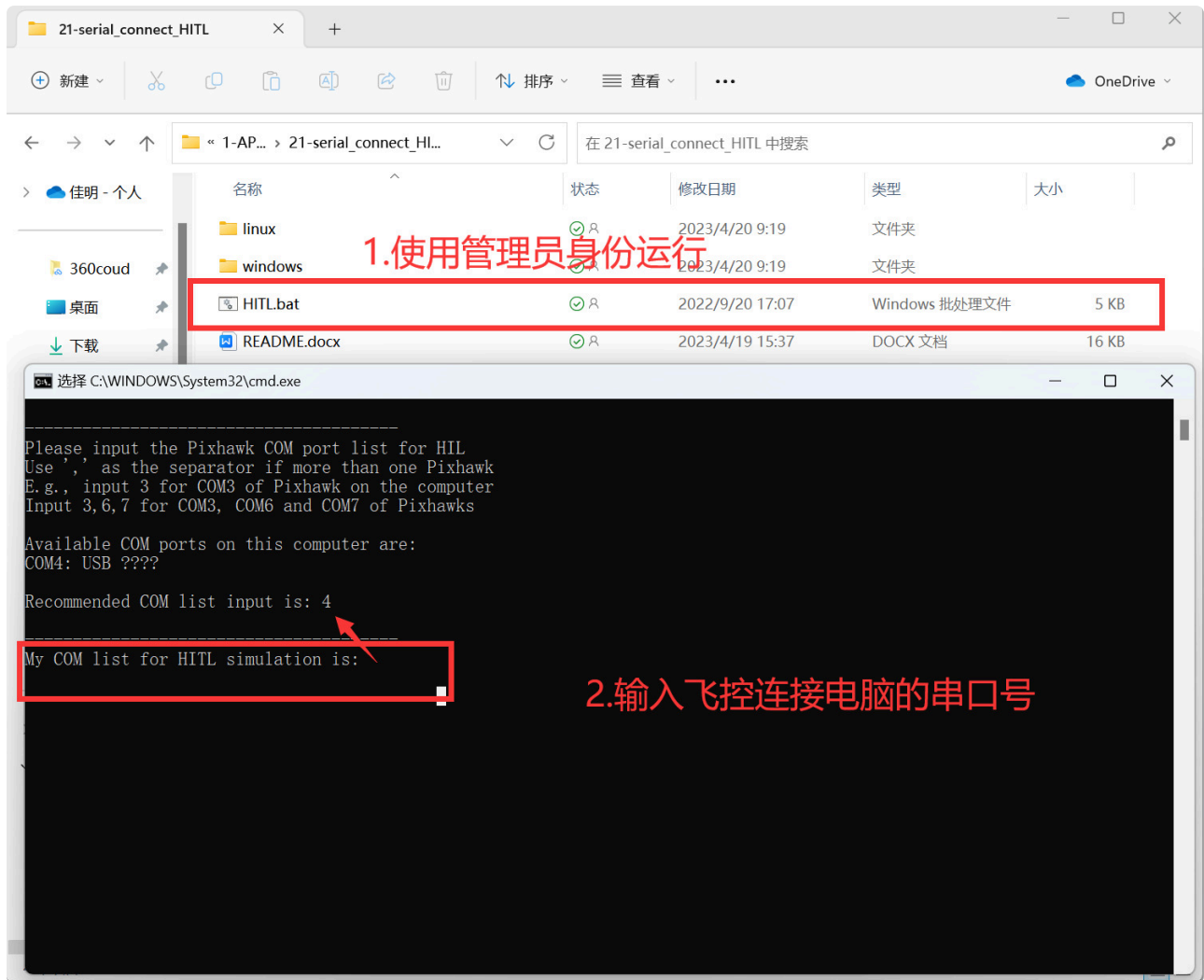
## Step 2

将linux文件夹复制到Linux系统中。



## Step 3

在Windows主机中，右键点击HTIL.bat以管理员身份运行，并在命令行界面输入端口号。



## Step 4

在Linux系统中打开demo.py文件，更改demo.py中使用的接口函数设定的端口号以及通信波特率，用 `python3 demo.py` 运行demo.py文件。

```
1
2 # import required libraries
3 # pip3 install pymavlink pyserial
4
5 import PX4MavCtrlV4 as PX4MavCtrl
6
7 💡 启用ROS发布模式
8 mav = PX4MavCtrl.PX4MavCtrlr(port="/dev/ttyACM0:57600")
9
10 mav.InitMavLoop()
11 # mav.InitTrueDataLoop()
12
13 # Enter Offboard mode to start vehicle control
14 mav.initOffboard()
15
16 # Get the takeoff position of each vehicle to the UE4 Map
17
18 # fly to 10m high above its takeoff position
19 mav.SendPosNED(1, 0, -1, 0)
20 # mav.SendCopterSpeed(3)
21
22
23 # lastTime = time.time()
24 # num = 0
25 # lastClock = time.time()
26 # while True:
27 #     lastTime = lastTime + 1/30.0
28 #     sleepTime = lastTime - time.time()
29 #     if sleepTime > 0:
30 #         time.sleep(sleepTime)
31 #     else:
32 #         lastTime = time.time()
```

## Step 5

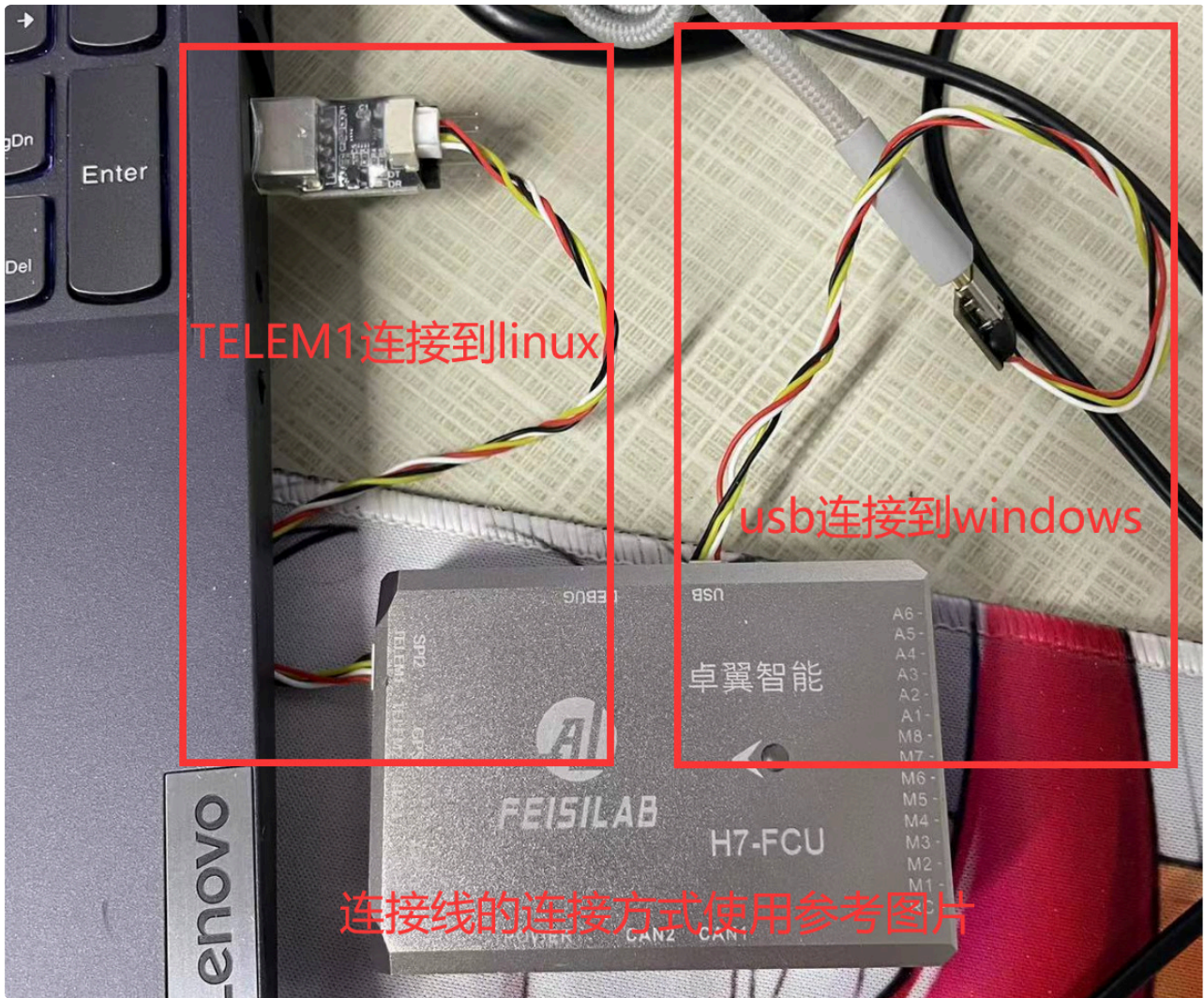
回到Windows主机上观察RflySim3D查看飞机。



## Windows下

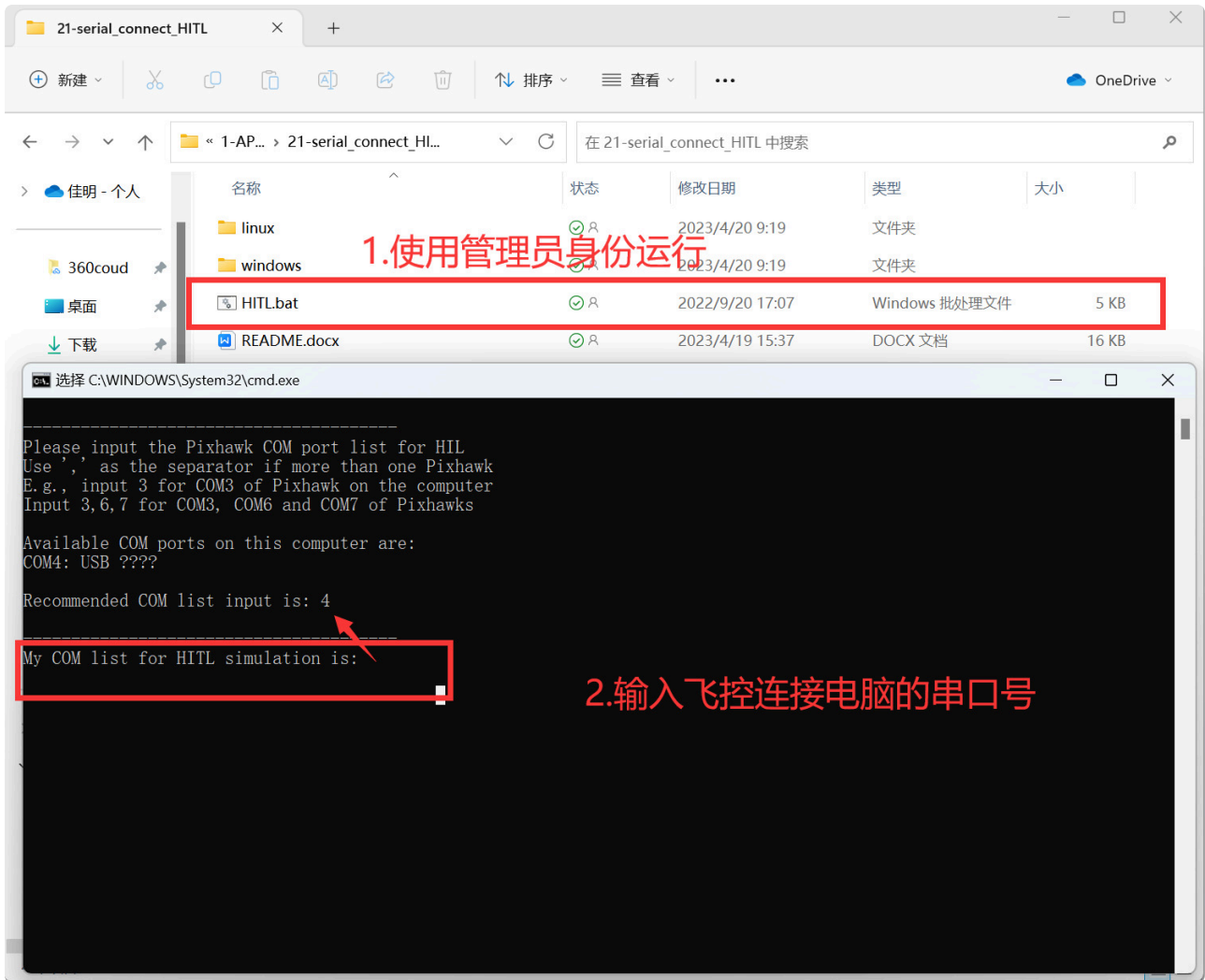
### Step 1

将设备连接起来，如图。



## Step 2

在Windows主机中，右键点击HTIL.bat以管理员身份运行，并在命令行界面输入端口号。



### Step 3

打开window文件夹下的demo.py文件，将接口函数的端口号与通信波特率更改为自己设定的值，并运行。

```
# import required libraries
# pip3 install pymavlink pyserial

import PX4MavCtrlV4 as PX4MavCtrl

# 启用ROS发布模式
mav = PX4MavCtrl.PX4MavCtrlr(port="COM3:11520")

mav.InitMavLoop()
# mav.InitTrueDataLoop()

# Enter Offboard mode to start vehicle control
mav.initOffboard()
mav.SendMavArm(True)
# Get the takeoff position of each vehicle to the UE4 Map

# fly to 10m high above its takeoff position
mav.SendPosNED(1, 0, -2, 0)
# mav.SendCopterSpeed(3)

# lastTime = time.time()
# num = 0
# lastClock = time.time()
# while True:
#     lastTime = lastTime + 1/30.0
```

## Step 4

在Rflysim3D仿真界面中可以看到飞机起飞。



## | 6.参考资料

无

## | 7.常见问题

Q1: 无

A1: 无