

---

## 1、实验名称及目的

**硬件在环仿真实验：**实现两路串口通信硬件在环仿真。

## 2、实验原理

首先通过 `PX4MavCtrl.PX4MavCtrlr()` 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.PX4MavCtrlr()` 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。

### 3、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。



### 4、文件目录

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
windows	放于 windows 主机上运行的程序
HITL.bat	硬件在环仿真启动脚本

### 5、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 <sup>①</sup>	1
2	RflySim 平台免费版及以上		
3	Visual Studio Code		
4	Linux	虚拟机/NX	1

①：推荐配置请见：<https://doc.rflysim.com/1.1InstallMethod.html>

## 6、实验步骤

### Linux 下

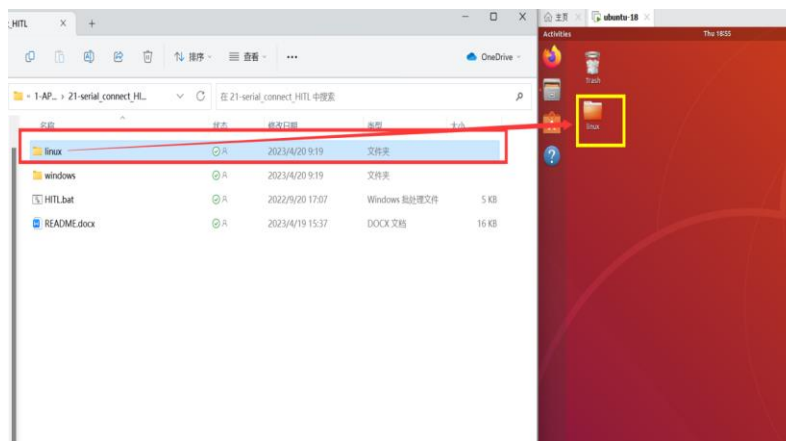
#### Step 1:

将设备连接起来，如图



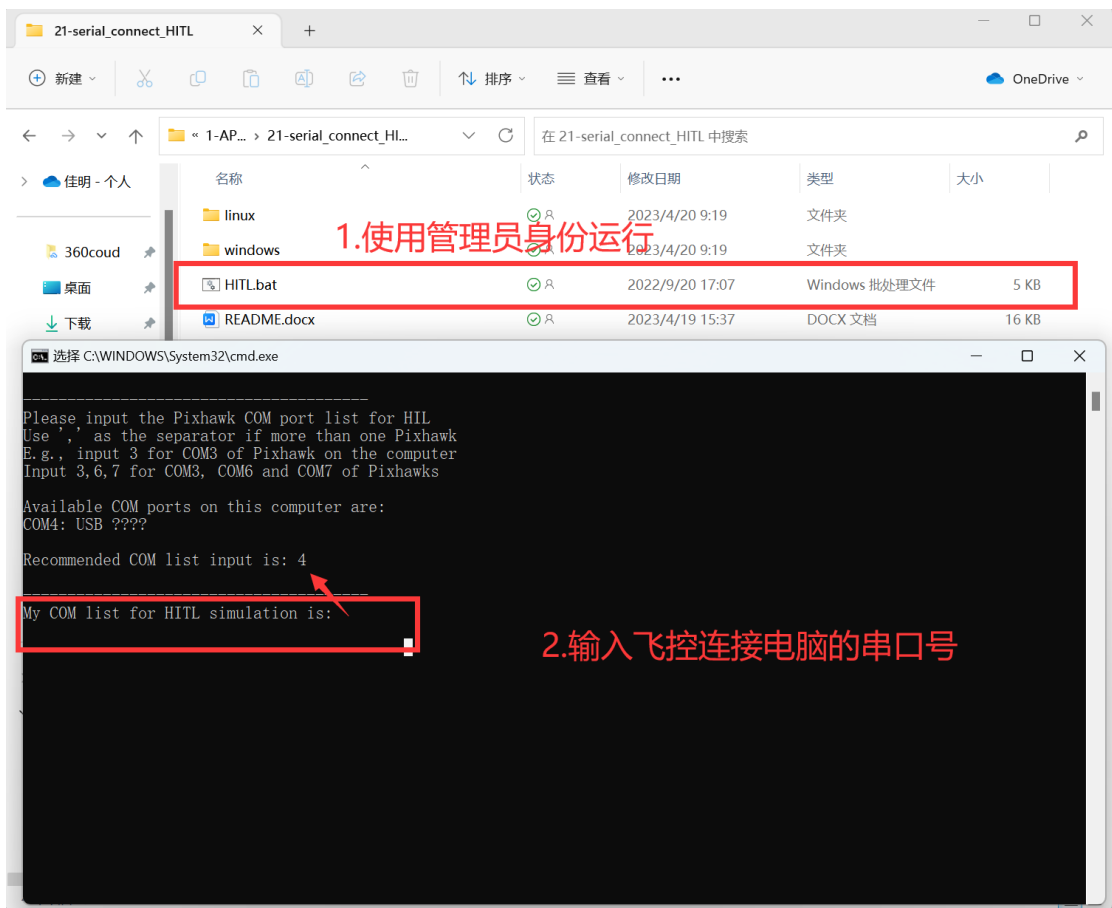
#### Step 2:

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:

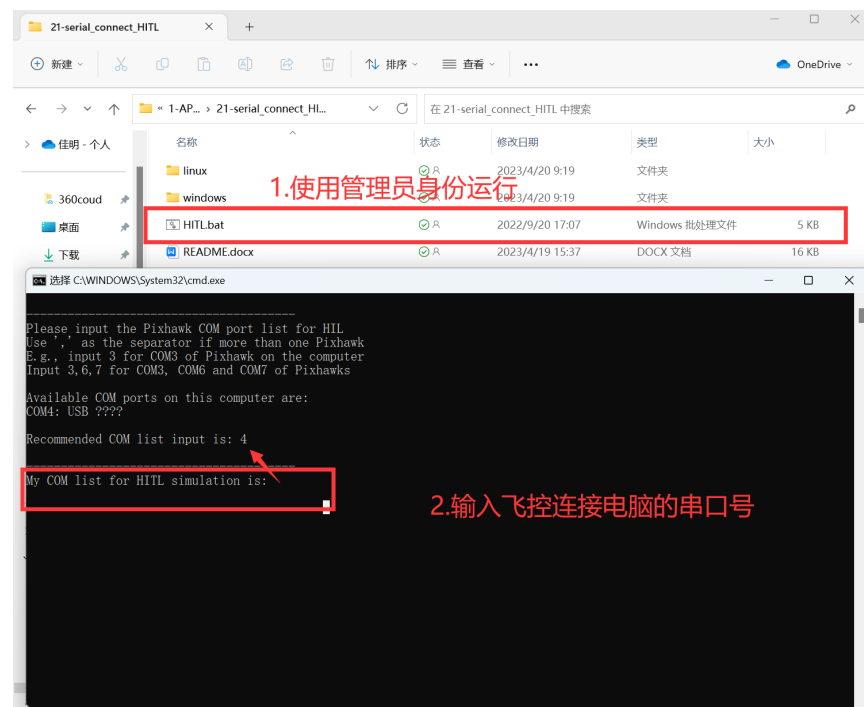
在 Linux 系统中打开 demo.py 文件，更改 demo.py 中使用的接口函数设定的端口号以及通信波特率

```
1 # import required libraries
2 # pip3 install pymavlink pyserial
3
4 import PX4MavCtrlV4 as PX4MavCtrl
5
6 # 启用ROS发布模式
7 mav = PX4MavCtrl.PX4MavCtrl(port="/dev/ttyACM0:57600")
8
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 # lastTime = time.time()
23 # num = 0
24 # lastClock = time.time()
25 # while True:
26 #     lastTime = lastTime + 1/30.0
27 #     sleepTime = lastTime - time.time()
28 #     if sleepTime > 0:
29 #         time.sleep(sleepTime)
30 #     else:
31 #         lastTime = time.time()
```

更改完成后运行 demo.py 文件

## Step 5:

回到 Windows 主机上观察 RflySim3D 查看飞机



## Linux 下

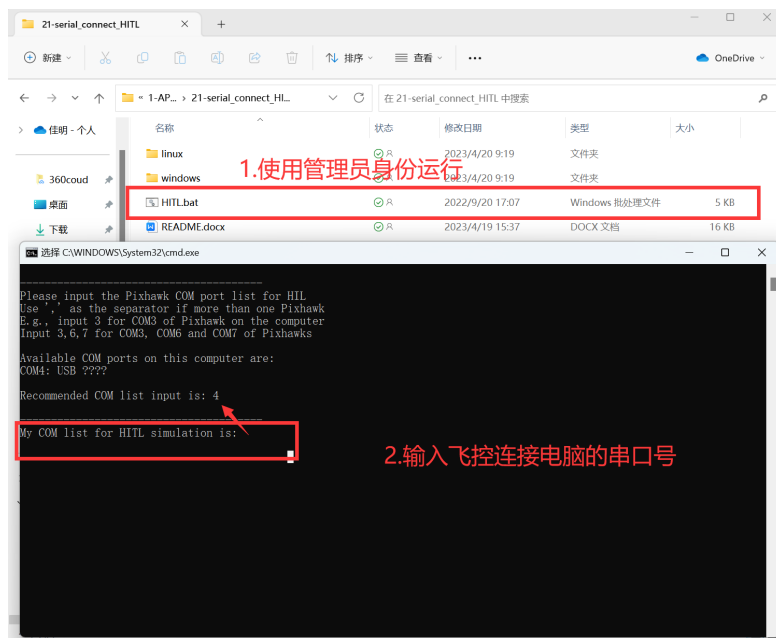
### Step 1:

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.PX4MavCtrl(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

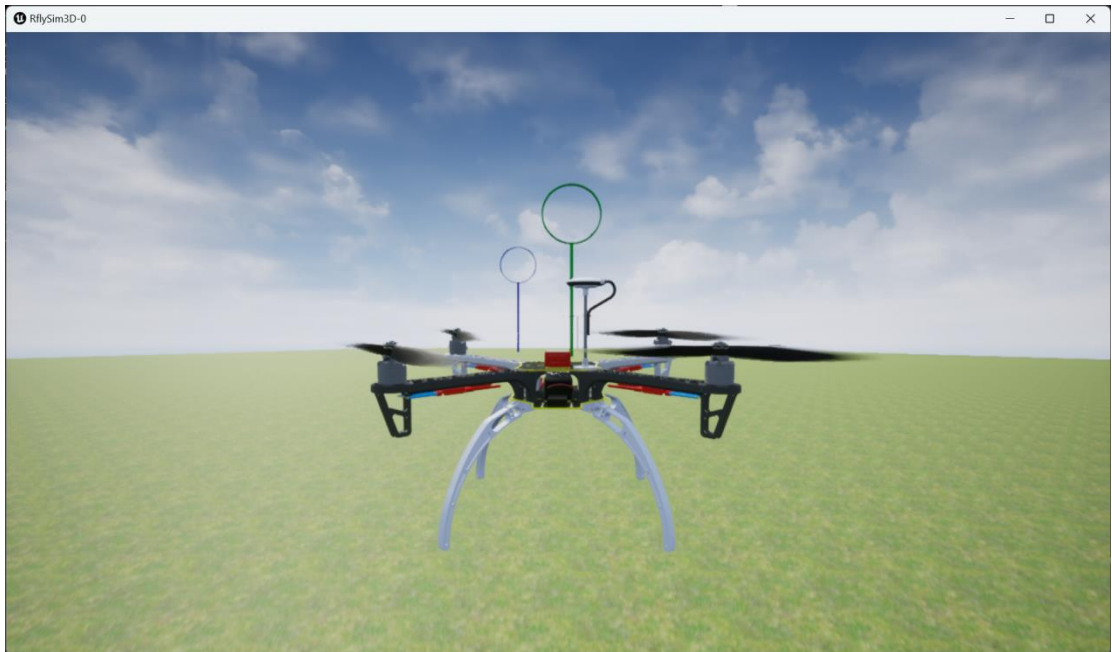
# fly to 10m high above its takeoff position
mav.SendPosNED(1, 0, -2, 0)
# mav.SendCopterSpeed(3)

# lastTime = time.time()
# num = 0
# lastClock = time.time()
# while True:
#     lastTime = lastTime + 1/20.0
```

---

## Step 4:

在 Rflysim3D 仿真界面中可以看到飞机起飞



## 7、参考文献

[1]. 无

## 8、常见问题

Q1: 无

A1: 无