

- 1.实验名称及目的
  - 1.1 实验名称
  - 1.2 实验目的
  - 1.3 关键知识点
    - 关键知识点1: 如何配置PX4 与 ROS 2 进行集成
    - 关键知识点2: 配置uXRCE-DDS
- 2.实验效果
- 3.文件目录
  - 4.1 软件要求
  - 4.2 硬件要求
- 5.实验步骤
  - 5.1 ROS2订阅PX4飞控消息（环境预备）
    - Step 1: 切换ROS环境
    - Step 2: 修改连接方式
    - Step 3: 启动软件在环仿真
    - Step 4: 创建连接
    - Step 4: 查看消息列表
  - 5.2 自动配置基于ROS2的PX4官方offboard控制实验
    - Step 1: 自动配置基于ROS2的PX4官方offboard控制实验
    - Step 2: 配置与暴露飞控消息（选做）
    - Step 3: MicroXRCEAgent的连接方式（选做）
    - Step 4: 设置PX4端的uxrce\_dds\_client客户端（选做）
  - 5.3 手动配置基于ROS2的PX4官方offboard控制实验（选做）
    - Step 1: 创建工作空间目录
    - Step 2: 进入源代码目录
    - Step 3: 克隆PX4相关的Git仓库
    - Step 4: 进入工作空间根目录
    - Step 5: 编译工作空间
    - Step 6: 每次运行前的环境配置
    - Step 7: 执行Offboard控制脚本
- 6.参考资料
- 7.常见问题

# 1.实验名称及目的

## 1.1 实验名称

## 1.2 实验目的

PX4 与 ROS2 的集成，通过 uXRCE-DDS 进行直接通信。

注：本实验需要RflySim中PX4版本为1.14.4及以上版本

## 1.3 关键知识点

### 关键知识点1：如何配置PX4 与 ROS 2 进行集成

入门教程：

步骤：[\[安装目录\]\RflySimAPIs\2.RflySimUsage\0.ApiExps\12\\_PX4ROS2](#)

原理：<https://docs.px4.io/main/en/ros/ros2.html>

### 关键知识点2：配置uXRCE-DDS

配置教程：

[https://docs.px4.io/main/en/ros/ros2\\_comm.html#setup-micro-xrce-dds-agent-client](https://docs.px4.io/main/en/ros/ros2_comm.html#setup-micro-xrce-dds-agent-client)

核心配置代码如下（注：WSL平台已配置，不需要再配，注意：需要CMake2.20以上才能编译，如果编译报错可以更换克隆的包的版本）：

```
git clone https://github.com/eProxima/Micro-XRCE-DDS-Agent.git
cd Micro-XRCE-DDS-Agent
mkdir build
cd build
cmake ..
make
sudo make install
sudo ldconfig /usr/local/lib/
```

## 2.实验效果

通过在Ubuntu中发送指令，可以观察到windows端的软件在环仿真无人机解锁启动。

## 3.文件目录

例程目录：[\[安装目录\]\RflySimAPIs\6.RflySimExtCtrl\0.ApiExps\19\\_uXRCE-DDS\\_ROS2CtrlExps\1.NetPortComm](#)

文件夹/文件名称	说明
SITLRun.bat	软件在环仿真启动脚本
RosSwitch.bat	Ros环境切换脚本
extract_and_copy.bat	基于ROS2的PX4官方offboard控制一键脚本
setup_px4_ros.sh	配置px4例程包

## 4.1 软件要求

Windows 10及以上版本；RflySim工具链。

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。

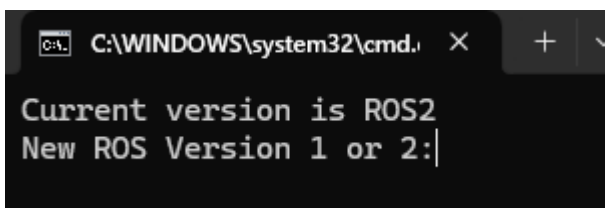
①：推荐配置请见：<https://rflysim.com/doc/zh/HowToInstall.pdf>

# 5. 实验步骤

## 5.1 ROS2订阅PX4飞控消息（环境预备）

### Step 1：切换ROS环境

双击运行“RosSwitch.bat”，确认处于ROS2状态，如果已经是ROS2则不操作，如果是ROS1则切回ROS2。



```
C:\WINDOWS\system32\cmd...  
Current version is ROS2  
New ROS Version 1 or 2:|
```

### Step 2：修改连接方式

1. 修改wsl.localhost\RflySim-20.04\root\.bashrc：

- 将 `export ROS_LOCALHOST_ONLY=1` 改为 `export ROS_LOCALHOST_ONLY=0`，启用广播支持

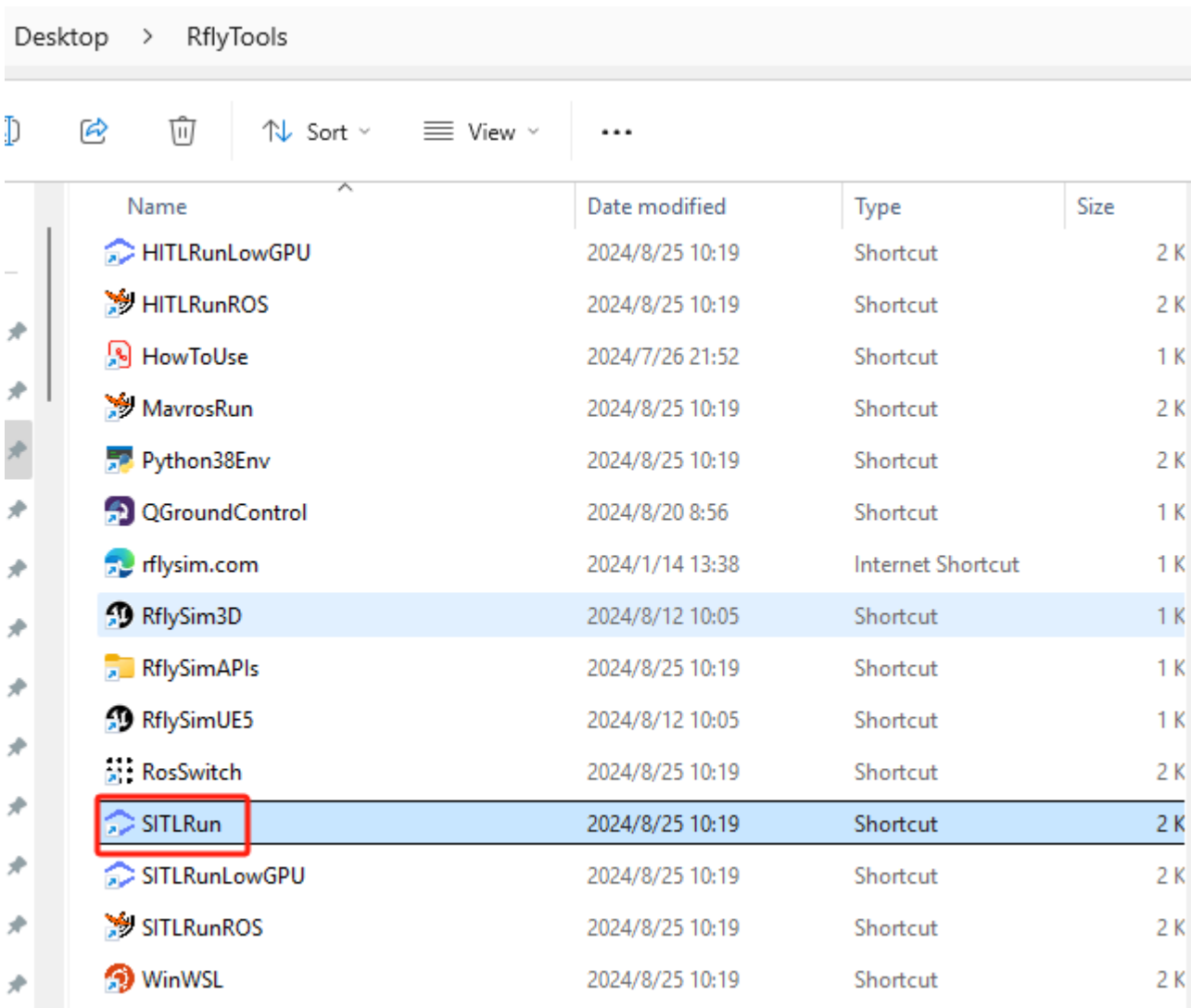
- 设置export RMW\_IMPLEMENTATION=rmw\_fastrtps\_cpp, 启用fast-dds, 来和MicroXRCEAgent匹配。注意注释掉cyclonedds的内容。

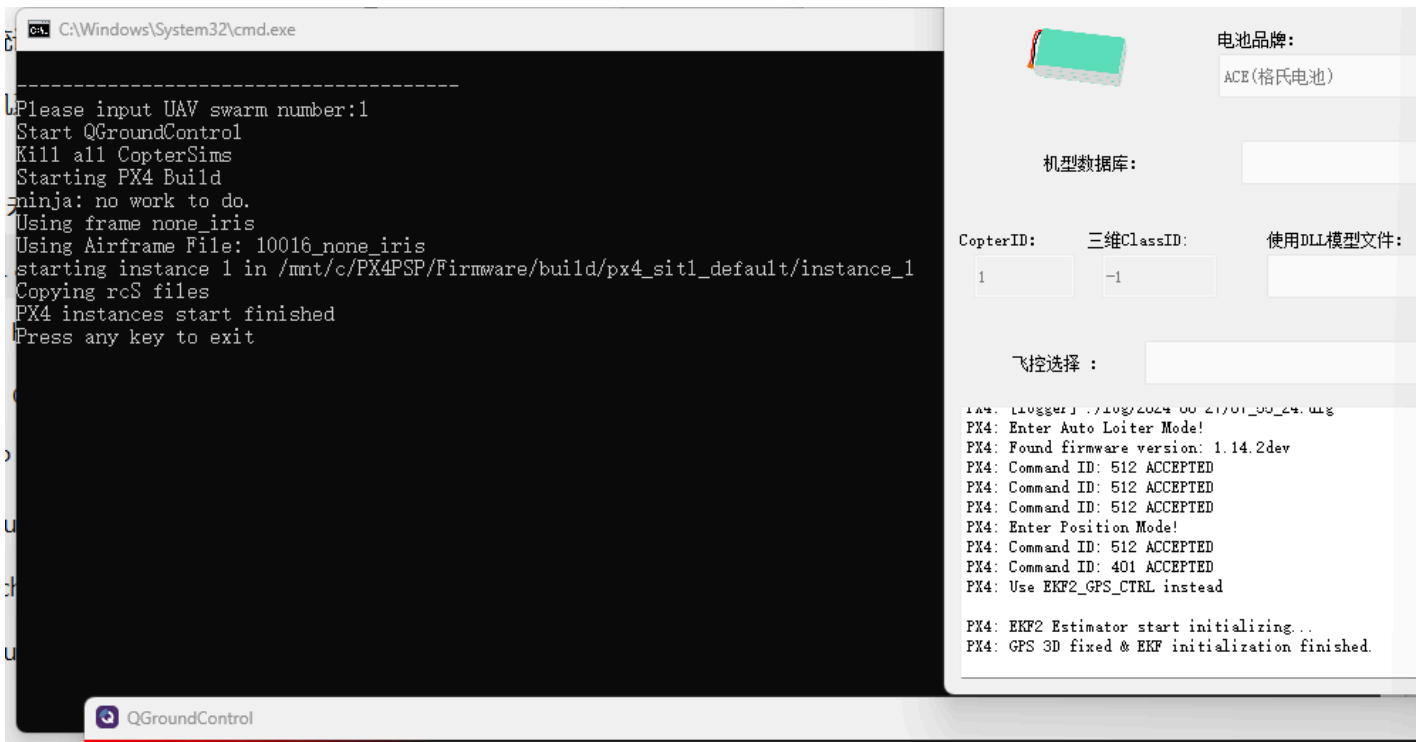
MicroXRCEAgent似乎必须这样才能发送消息。ROS\_LOCALHOST\_ONLY=1是为了防止局域网内电脑相互干扰, 在真机情况下ROS\_LOCALHOST\_ONLY=0是更可靠的方法。

2. 打开一个CMD窗口, 输入 `wsl --shutdown` 关闭所有WSL, 确保下次启动生效

## Step 3: 启动软件在环仿真

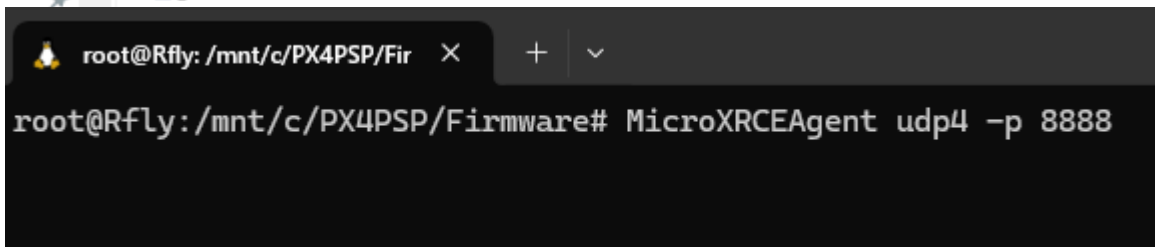
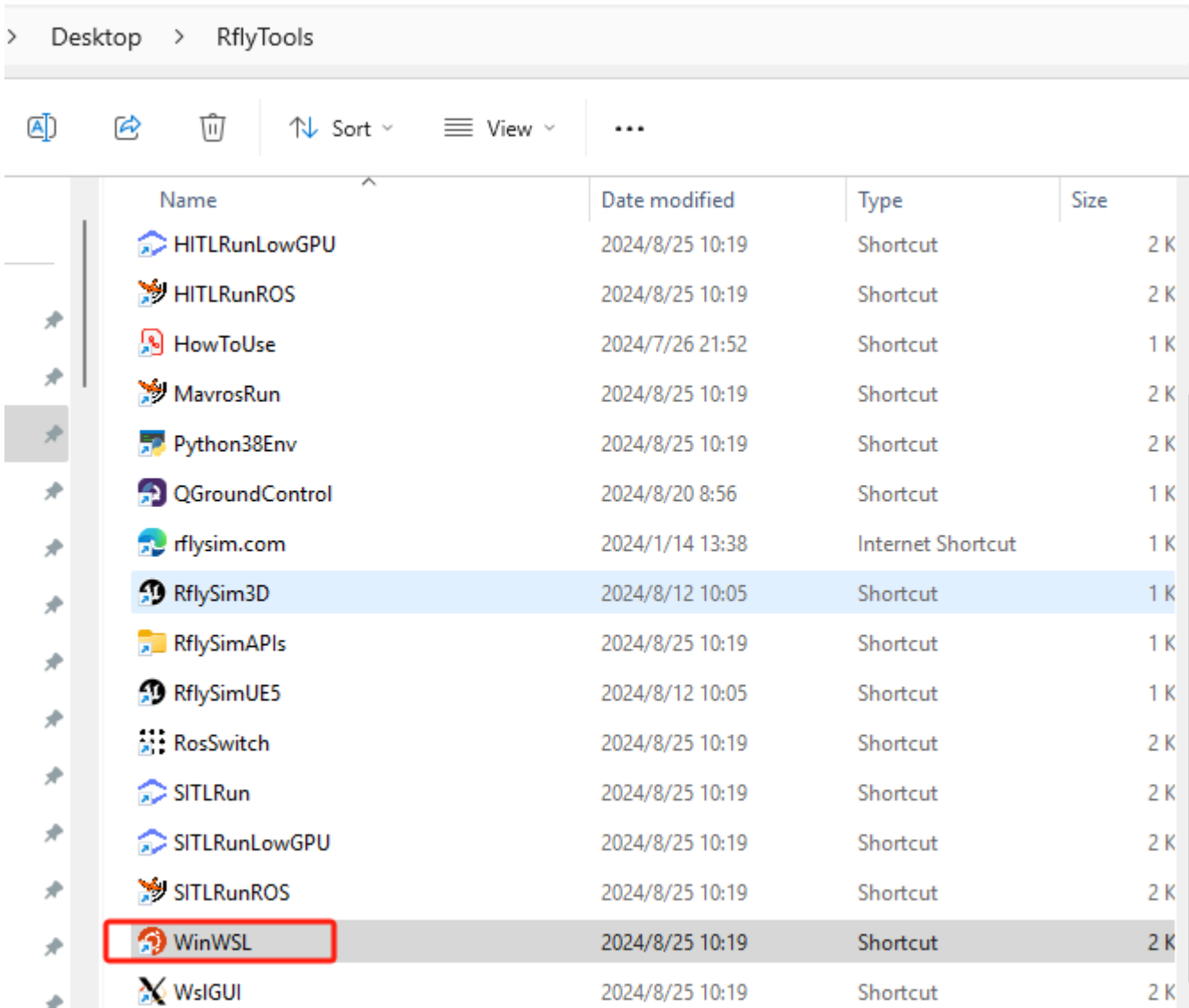
双击运行桌面SITLRun快捷方式, 启动一架飞机的软件在环仿真



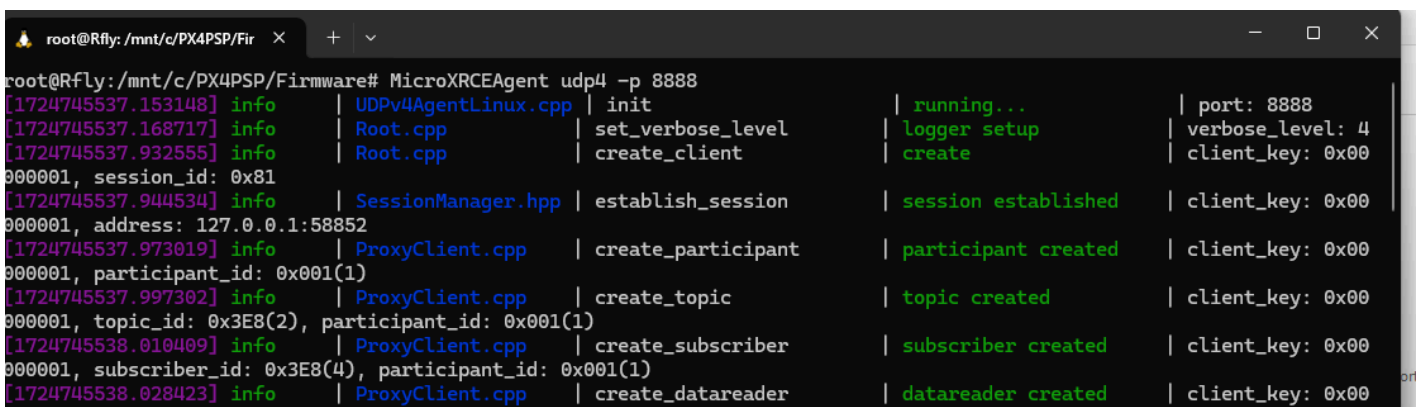


## Step 4: 创建连接

双击桌面WinWSL，开启一个终端，输入 `MicroXRCEAgent udp4 -p 8888`。MicroXRCEAgent是一个用于微控制器之间通信的代理程序。udp4表示使用IPv4的UDP协议，`-p 8888`表示监听8888端口。

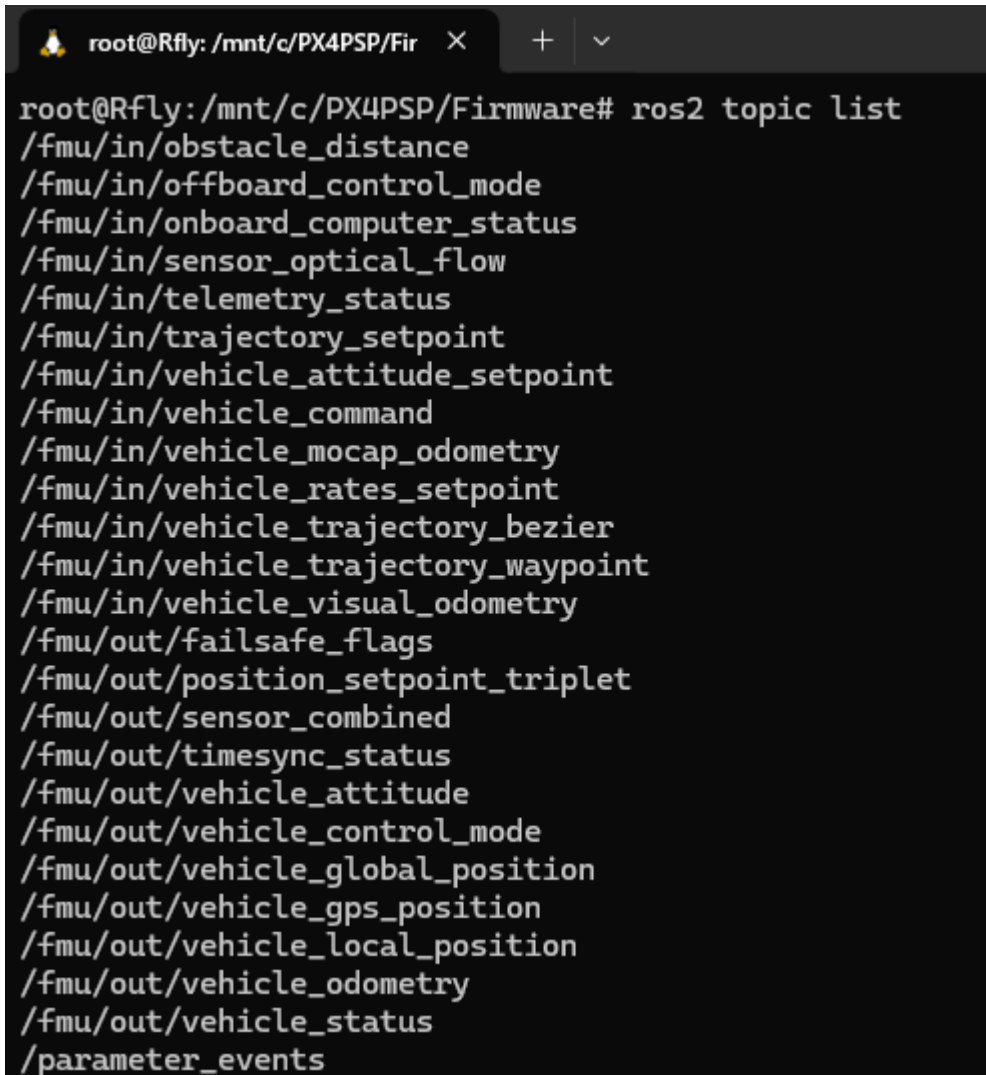


可以看到类似如下打印信息



## Step 4: 查看消息列表

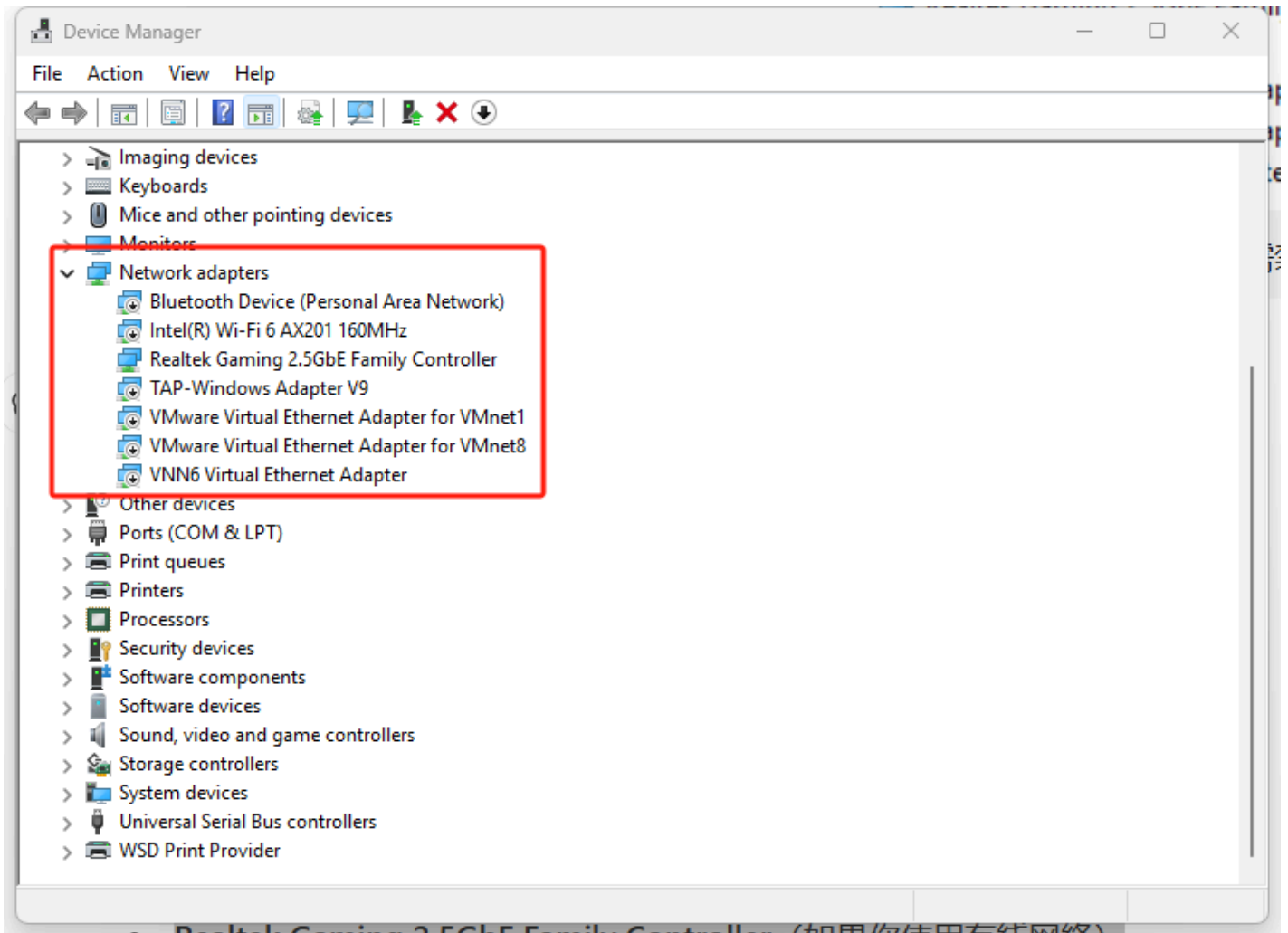
再次双击桌面WinWSL打开另一个终端，输入“ros2 topic list”就能看到现有的消息列表。一系列以/fmu/in/和/fmu/out/开头的消息主题。这些主题代表了与飞控单元（FMU）之间的数据通信。



```
root@Rfly: /mnt/c/PX4PSP/Fir x + v
root@Rfly: /mnt/c/PX4PSP/Firmware# ros2 topic list
/fmu/in/obstacle_distance
/fmu/in/offboard_control_mode
/fmu/in/onboard_computer_status
/fmu/in/sensor_optical_flow
/fmu/in/telemetry_status
/fmu/in/trajectory_setpoint
/fmu/in/vehicle_attitude_setpoint
/fmu/in/vehicle_command
/fmu/in/vehicle_mocap_odometry
/fmu/in/vehicle_rates_setpoint
/fmu/in/vehicle_trajectory_bezier
/fmu/in/vehicle_trajectory_waypoint
/fmu/in/vehicle_visual_odometry
/fmu/out/failsafe_flags
/fmu/out/position_setpoint_triplet
/fmu/out/sensor_combined
/fmu/out/timesync_status
/fmu/out/vehicle_attitude
/fmu/out/vehicle_control_mode
/fmu/out/vehicle_global_position
/fmu/out/vehicle_gps_position
/fmu/out/vehicle_local_position
/fmu/out/vehicle_odometry
/fmu/out/vehicle_status
/parameter_events
```

注意：

1.如果WSL中运行 `ros2 topic list` 命令时，无法获取上述消息（8888端口被占用），请尝试去设备管理器中，禁用除当前使用网络适配器之外的所有其他适配器（例如，VMware打头的虚拟机适配器，或有线/无线网卡适配器），然后重启电脑后，再尝试运行本例程。



可以尝试禁用以下适配器，然后重新启动计算机，再次运行相关的命令：

1. **VMware Virtual Ethernet Adapter for VMnet1**
2. **VMware Virtual Ethernet Adapter for VMnet8**
3. **TAP-Windows Adapter V9**
4. **VNN6 Virtual Ethernet Adapter**

保留：

- **Intel(R) Wi-Fi 6 AX201 160MHz**（如果使用无线网络）
- **Realtek Gaming 2.5GbE Family Controller**（如果使用有线网络）

## 5.2 自动配置基于ROS2的PX4官方offboard控制实验

### Step 1：自动配置基于ROS2的PX4官方offboard控制实验

再次双击桌面WinWSL打开一个终端，首先需要为ROS2工作空间创建一个新的目录，用于存放PX4相关的消息包和通信包：

```
mkdir -p ~/ws_sensor_combined/src/
```

双击启动本目录下的extract\_and\_copy.bat脚本，编译完成后进入offboard模式

```
colcon build [2/2 done] [0 ong × + v
px4_msgs already exists, skipping clone.
px4_ros_com already exists, skipping clone.
Starting >>> px4_msgs
Finished <<< px4_msgs [27.3s]
Starting >>> px4_ros_com
Finished <<< px4_ros_com [0.51s]

Summary: 2 packages finished [28.2s]
Starting offboard control node...
[INFO] [1724808461.808700200] [offboard_control]: Arm command send
```

此时RflySim3D中可以看见飞机起飞



## Step 2: 配置与暴露飞控消息（选做）

1.消息列表限制:

在运行命令 `ros2 topic list` 后，所看到的消息列表并不包括PX4飞控内的所有消息。实际上，哪些消息会被暴露（即可以在ROS2中看到），以及每个消息的发布频率，都是由一个名为dds\_topics.yaml的配置文件决定的。

2.dds\_topics.yaml 文件的配置:

dds\_topics.yaml 是一个配置文件，它定义了需要通过DDS（Data Distribution Service）通信协议暴露的消息，以及这些消息的频率。如果希望暴露额外的消息或调整频率，可以通过修改这个文件来实现。

可以参考 PX4文档 以了解如何配置和修改这个文件。

### 3.修改文件路径：

如果想要暴露额外的消息，需要编辑位于以下路径的 dds\_topics.yaml 文件：`Firmware/src/modules/uxrce_dds_client/dds_topics.yaml`

## Step 3: MicroXRCEAgent的连接方式（选做）

### 1. MicroXRCEAgent 网络连接：

之前的步骤已经提到，可以通过UDP网络连接来运行PX4模拟器并与之通信。例如，使用以下命令通过端口8888来启动代理：

```
MicroXRCEAgent udp4 -p 8888
```

### 2.有线连接（串行连接）：

除了通过网络连接，MicroXRCEAgent还可以通过串行接口直接与硬件通信。例如，如果在Raspberry Pi上使用UART0串口，可以使用以下命令连接：

```
sudo MicroXRCEAgent serial --dev /dev/ttyAMA0 -b 921600
```

这种连接方式通常用于实际硬件部署中，通过串行端口进行更低层次、更稳定的通信。

## Step 4: 设置PX4端的uxrce\_dds\_client客户端（选做）

### 1.默认启用：

PX4飞控端的 uxrce\_dds\_client客户端默认是启用的，这意味着在大多数情况下，不需要手动启动它。

### 2.设置和启动客户端的方法：

如果需要手动配置或启动 uxrce\_dds\_client，可以参考

[https://docs.px4.io/main/en/middleware/uxrce\\_dds.html#starting-the-client](https://docs.px4.io/main/en/middleware/uxrce_dds.html#starting-the-client)

了解详细操作步骤。这可能包括调整启动参数或者确保客户端在特定情况下正确运行。

## 5.3 手动配置基于ROS2的PX4官方offboard控制实验 (选做)

### Step 1: 创建工作空间目录

再次双击桌面WinWSL打开一个终端，首先需要为的ROS2工作空间创建一个新的目录，用于存放PX4相关的消息包和通信包：

```
mkdir -p ~/ws_sensor_combined/src/
```

`-p` 标志会自动创建目录结构（如果尚不存在），`~/ws_sensor_combined/` 是工作空间的根目录，`src/` 是存放源代码的子目录。

### Step 2: 进入源代码目录

然后，切换到刚创建的源代码目录：

```
cd ~/ws_sensor_combined/src/
```

### Step 3: 克隆PX4相关的Git仓库

通过Git命令克隆PX4的消息包和ROS通信包：

```
git clone https://github.com/PX4/px4_msgs.git  
git clone https://github.com/PX4/px4_ros_com.git
```

`px4_msgs`: 包含PX4使用的自定义消息定义。

`px4_ros_com`: 包含用于ROS 2和PX4之间通信的相关代码。

### Step 4: 进入工作空间根目录

切换回工作空间的根目录：

```
cd ~/ws_sensor_combined/
```

### Step 5: 编译工作空间

在工作空间目录下（例如 `~/ws_sensor_combined/`），执行以下命令：

```
colcon build
```

这将构建 `src/` 目录中的所有 ROS 2 包。

将生成的可执行文件、库文件、资源等安装到 `install/` 目录中。

## Step 6: 每次运行前的环境配置

为了确保ROS 2环境正确设置，运行以下命令来配置工作空间环境：

```
source ~/ws_sensor_combined/install/local_setup.bash
```

这条命令会加载工作空间的环境变量，使得可以使用在这个工作空间中构建的包和节点。

## Step 7: 执行Offboard控制脚本

运行以下Python脚本来实现Offboard控制功能：

```
ros2 run px4_ros_com offboard_control
```

这个脚本通常用于控制无人机或其他机器人在不依赖地面站的情况下进行自主飞行或操作。Offboard控制意味着可以通过自定义代码直接发送控制指令给飞控系统。

如果不确定路径，可以使用 `find` 命令来搜索文件的位置。例如：

```
find ~/ws_sensor_combined/ -name "offboard_control.py"
```

# 6. 参考资料

- 1.

# 7. 常见问题

Q1: \*\*\*

A1: \*\*\*