

# | ROS2环境下C++测试实验

## | 1. 实验目的

在WSL设置出的Ubuntu环境下，通过C++程序，观察ROS2数据交换。

## | 2. 实验要求

- 软件要求：Windows 10及以上版本；RflySim工具链<sup>[1]</sup>，MATLAB 2022b及以上，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。
- 硬件要求：笔记本/台式电脑1台<sup>[2]</sup>，平台安装时的编译命令为：px4\_fmu-v6x\_default，固件版本为：1.13.2。

## | 3. 实验地址

例程目录：

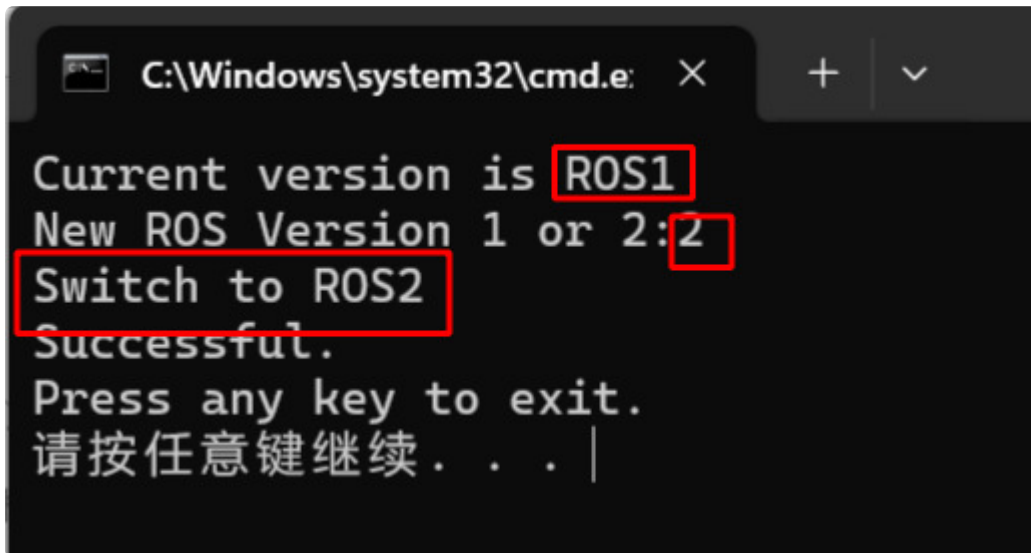
[\[安装目录\]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\18.MavrosCommAPI\ROS2TestC++](#)

- Ros2\_node：ROS2文件
- HITLRun.bat：硬件在环启动脚本 [mavros\\_run.sh](#)
- RosSwitch.bat：ROS环境切换脚本
- WinWSL.bat：Ubuntu终端环境启动脚本
- [mavros\\_run.sh](#)：Mavros启动脚本
- SITLRunDebug.bat：软件在环启动脚本

## 4. 实验内容或步骤

### ROS2环境下C++测试实验

双击"[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.e: X + v
Current version is ROS1
New ROS Version 1 or 2: 2
Switch to ROS2
Successful.
Press any key to exit.
请按任意键继续 . . . |
```

双击"[WinWSL.bat](#)"进入平台Ubuntu环境（或使用自己的Ubuntu虚拟机/NX板卡），并输入如下指令，将ros2\_node文件夹源码拷贝到~/colcon\_ws/src中，并编译

```
mkdir -p ~/colcon_ws/src
```

```
chmod 777 -R ros2_node
```

```
cp -rf ros2_node ~/colcon_ws/src
```

```
cd ~/colcon_ws
```

```
colcon build
```

如果提示Finished，说明我们的ROS2程序编译为可执行文件。

运行[SITLRunDebug.bat](#)开启一个飞机的SITL仿真，或者运行[HITLRun.bat](#)并输入飞控串口号，开启一个飞机的硬件在环仿真。

双击"[WinWSL.bat](#)"进入Ubuntu终端环境，再输入"./mavros\_run.sh 1"，启动一个飞机的mavros节点。

再次双击"[WinWSL.bat](#)"进入Ubuntu终端环境，再依次输入如下命令，开启ros2\_node节点

```
source ~/colcon_ws/install/setup.bash
```

```
ros2 launch ros2_node ros2_node.launch
```

此时屏幕上打印"Message Send."，说明正在发送消息

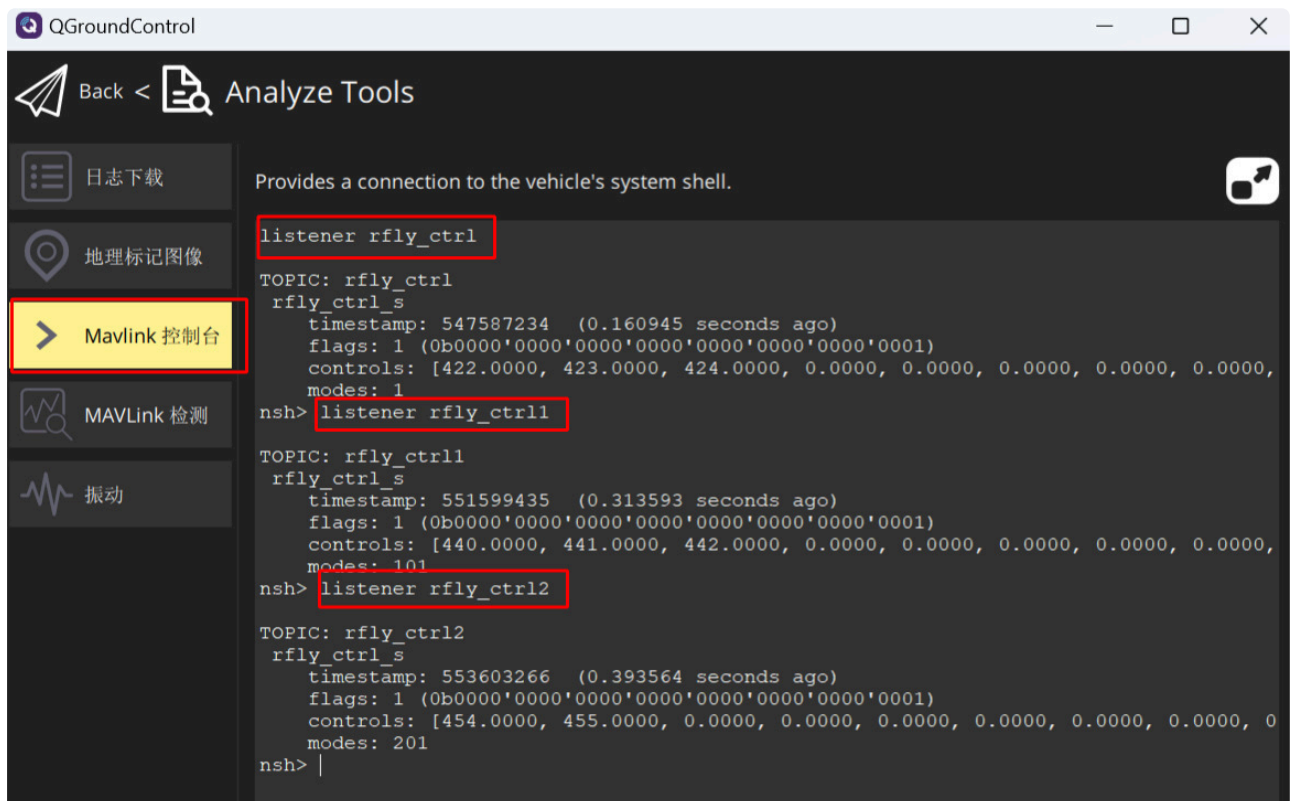
进入SITL的pxh或硬件在环的MAVLink控制台，依次输入如下指令并回车

```
listener rfly_ctrl
```

```
listener rfly_ctrl1
```

```
listener rfly_ctrl2
```

可以看到数据变化，说明成功通过ROS2将数据发给了飞控。



The screenshot shows the QGroundControl interface with the 'Mavlink 控制台' (Mavlink Console) tab selected. The console displays the following output:

```
Provides a connection to the vehicle's system shell.

listener rfly_ctrl
TOPIC: rfly_ctrl
rfly_ctrl_s
timestamp: 547587234 (0.160945 seconds ago)
flags: 1 (0b0000'0000'0000'0000'0000'0000'0000'0000'0001)
controls: [422.0000, 423.0000, 424.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000]
modes: 1
nsh> listener rfly_ctrl1
TOPIC: rfly_ctrl1
rfly_ctrl_s
timestamp: 551599435 (0.313593 seconds ago)
flags: 1 (0b0000'0000'0000'0000'0000'0000'0000'0000'0001)
controls: [440.0000, 441.0000, 442.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000]
modes: 101
nsh> listener rfly_ctrl2
TOPIC: rfly_ctrl2
rfly_ctrl_s
timestamp: 553603266 (0.393564 seconds ago)
flags: 1 (0b0000'0000'0000'0000'0000'0000'0000'0000'0001)
controls: [454.0000, 455.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000]
modes: 201
nsh> |
```

注意：本例程也支持在配置好ROS环境和RflySimSDK库的Ubuntu电脑、NX板卡上面运行，进行分布式控制实验。

## 5. 关键知识点

### 关键知识点1：PX4MavCtrlV4ROS库

基于ROS开发的控制接口程序库，专门用于与PX4飞控系统进行通信和控制。

## 关键知识点2：硬件在环实验

硬件设备（如传感器、执行器等）直接与计算机连接，通过ROS中的节点进行控制和数据交换，以模拟真实环境中的机器人行为

## 关键知识点3：PX4飞控系统

了解PX4飞控系统的基本工作原理和如何通过ROS接口与其进行通信和控制。

## 6.参考资料

1. [RflySim官方文档](#)
2. [ROS2官方文档](#)
3. [PX4飞控官方文档](#)

## 7.常见问题

### Q1：编译过程中出现依赖项缺失错误

A1：确保已正确安装所有依赖项，包括ROS2环境和相关的开发工具。可以尝试运行`sudo apt update && sudo apt upgrade`更新系统，然后重新安装缺失的依赖包。

### Q2：无法建立与飞控的MAVLink连接

A2：检查串口权限和连接状态，确保串口号正确。在Ubuntu中可以通过`ls /dev/tty*`命令查看可用串口，同时确保用户具有相应串口访问权限，可能需要将用户添加到`dialout`组：  
`sudo usermod -a -G dialout $USER`。

### Q3：ROS2节点无法正常通信或接收数据

A3：检查是否正确设置了ROS2环境变量，确认各节点在同一ROS域中。运行`source ~/colcon_ws/install/setup.bash`确保环境设置正确，并检查话题名称是否匹配。

---

1. <https://rflysim.com/> ↩

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