RflySim工具链WSL与图形界面使用说明

1. 实验目的

1. 理解 RflySim 工具链内置 WinWSL 环境的作用与进入方法
2. 掌握在 Ubuntu 环境下执行 Python 或 shell 脚本的基本流程
3. 学习 WslGUI 图形界面的启动、退出和例程运行方法
4. 掌握 ROS1/ROS2、mavros、Simulink 与 RflySim 工具链的基础互联方式

2. 实验要求

- 软件要求：Windows 10/11；RflySim工具链[1]；工具链内置 WinWSL（RflySim-20.04）环境。
- 硬件要求：笔记本/台式电脑1台[2]。
- 扩展要求：如需运行 ROS 与 Simulink 互联实验，建议使用 MATLAB 2022a 或以上版本。

3.实验地址

例程目录：[安装目录]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e7.WslGUI

本例程文件夹下的文件目录结构说明如下：

```
e7.WslGUI
├── GuiTest                          # WslGUI图形界面与图像传输测试
│   ├── client_ue4_SITL.bat          # 打开SITL仿真飞机
│   ├── Config.json                  # UE4图像传输配置文件
│   ├── server_ue4.py                # Ubuntu环境下运行的图像接收显示脚本
│   ├── WinWSL.bat                   # 快速进入当前目录的WinWSL脚本
│   └── WslGUI.bat                   # 快速启动WslGUI图形界面脚本
├── MavTest                          # Ubuntu环境下运行MAVLink控制测试
│   ├── MavTest.bat                  # 打开SITL仿真飞机
│   ├── MavTest.py                   # Ubuntu环境下运行的MAVLink控制脚本
│   └── WinWSL.bat                   # 快速进入当前目录的WinWSL脚本
├── RosTest                          # ROS、mavros与Simulink互联测试
│   ├── RosTest.slx                  # Simulink ROS测试模型
│   └── SITLRunROS.bat               # 启动SITL并连接mavros的脚本
```

4. 实验内容或步骤

本实验包含 WinWSL 命令行使用、WslGUI 图形界面使用、ROS 开发接口验证，以及 ROS 与 Simulink 互联互通四部分内容。

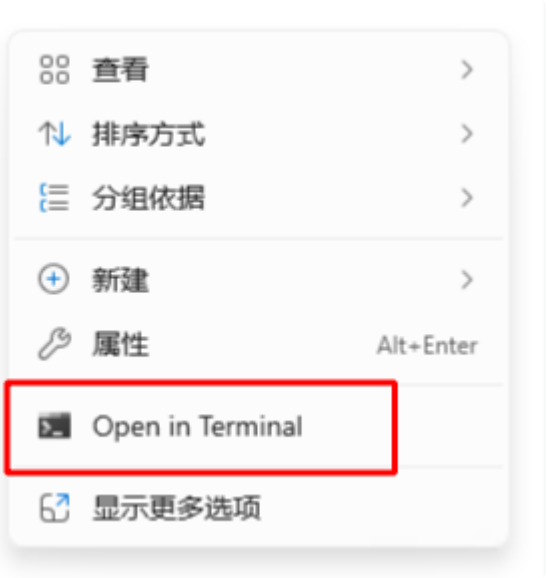
4.1 WinWSL命令行的使用方法

RflySim 工具链提供了一个基于 WSL 的 Ubuntu 20.04 虚拟机子系统，名称为 `RflySim-20.04`。一方面，它用于编译 PX4 固件、开启 SITL 仿真并连接 CopterSim 仿真器；另一方面，它也是一个完整的 Ubuntu 系统，可支撑视觉算法开发与调试环境，工具链中已配置 ROS1/ROS2、OpenCV 等环境。

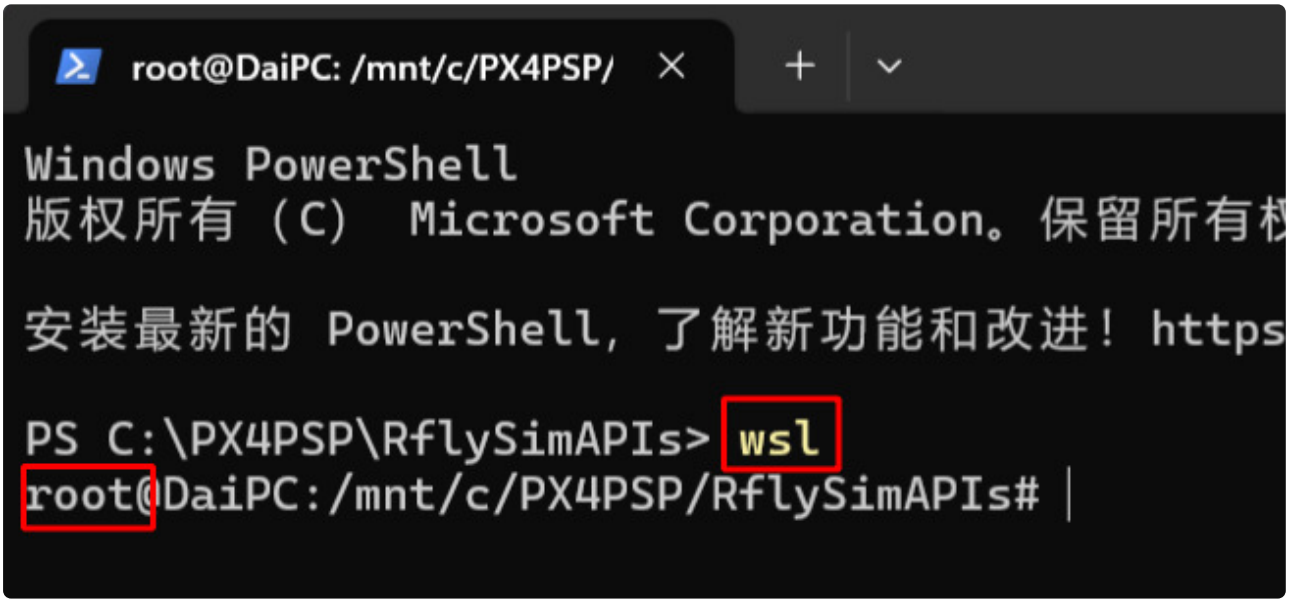
进入 RflySim 工具链环境并进入指定例程目录后，即可在 Ubuntu 环境下执行 Python 或 shell 脚本。常用方法如下。

4.1.1 鼠标右键输入命令进入WinWSL

进入任意例程文件夹，鼠标右键，点击“Open in Terminal”（在终端打开）。



在弹出窗口中输入 `wsl` 或 `bash`，即可进入类似 `root@用户名:文件目录` 的 Ubuntu 目录下。

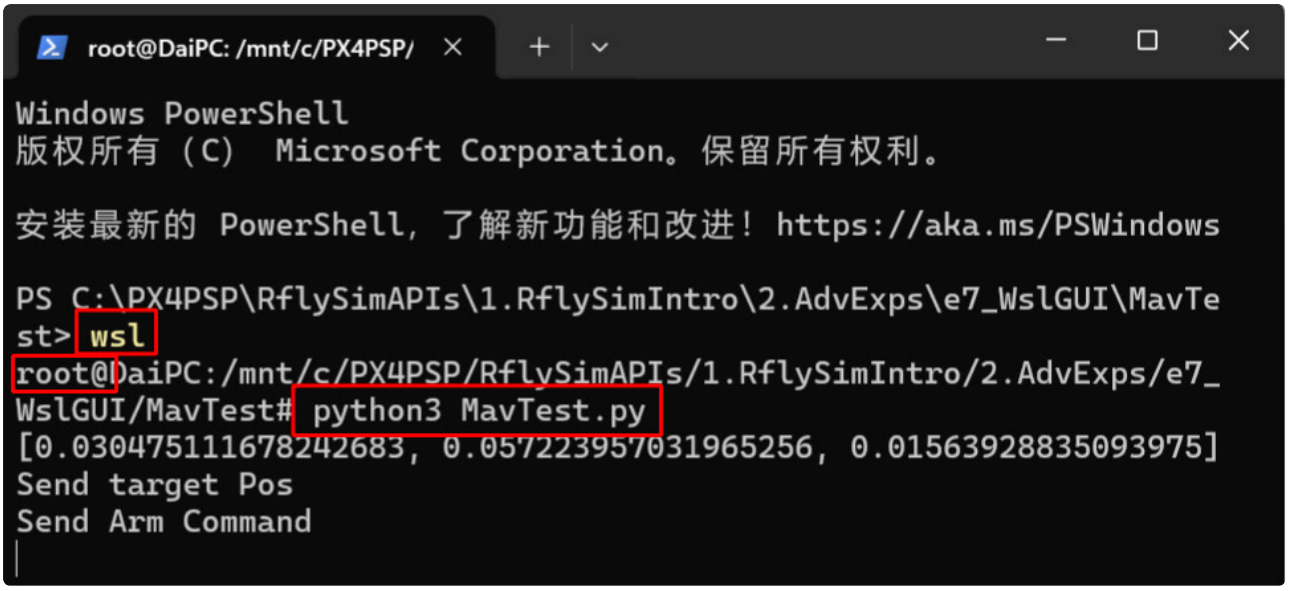


此时可以正常输入 Ubuntu 命令，例如 `ls`、`python3 ***` 等。

例如进入 `e7.WslGUI\MavTest` 目录，先双击 `MavTest.bat` 打开一个飞机的 SITL 仿真；然后在本例程目录中鼠标右键用终端打开，输入 `wsl` 进入 WinWSL 环境，再输入：

```
python3 MavTest.py
```

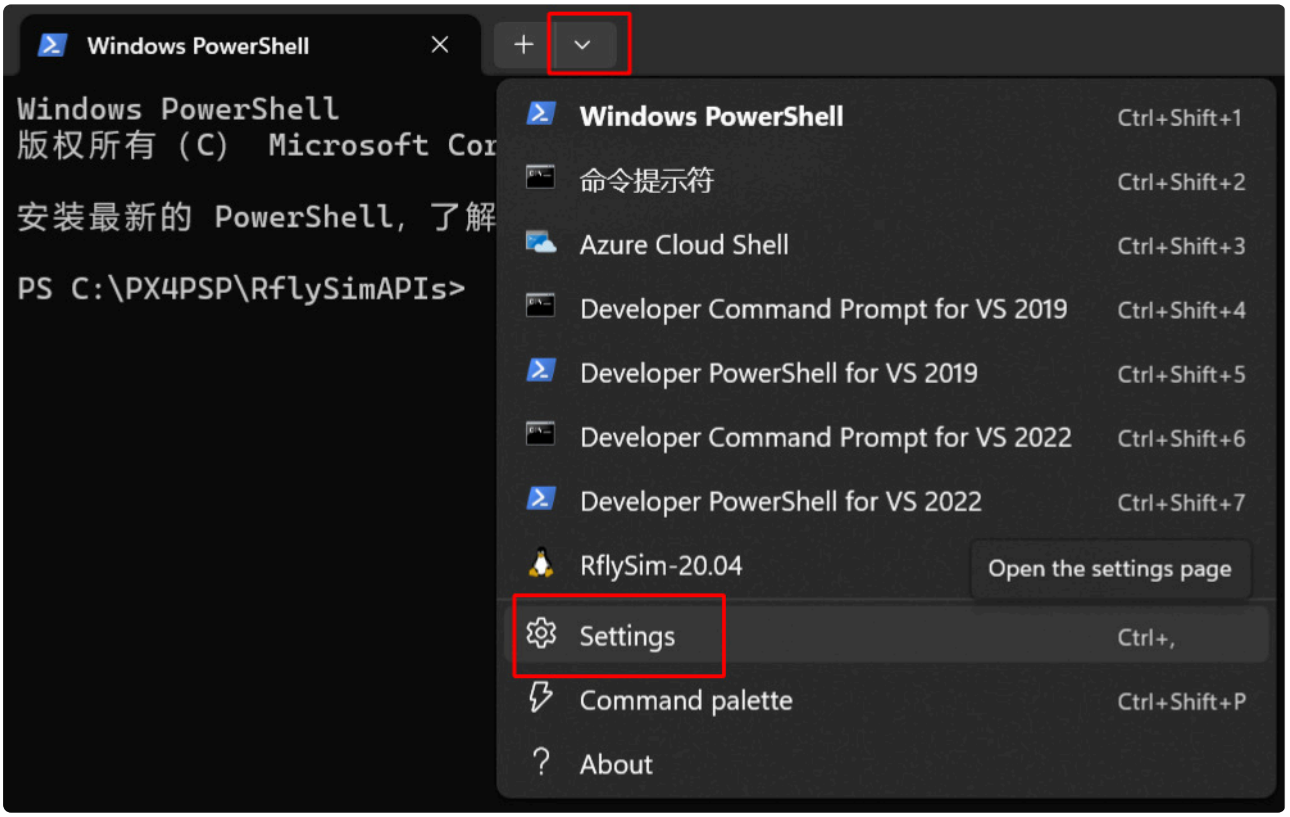
即可在 Ubuntu 环境下运行 `MavTest.py` 脚本，实现飞机的飞行控制。



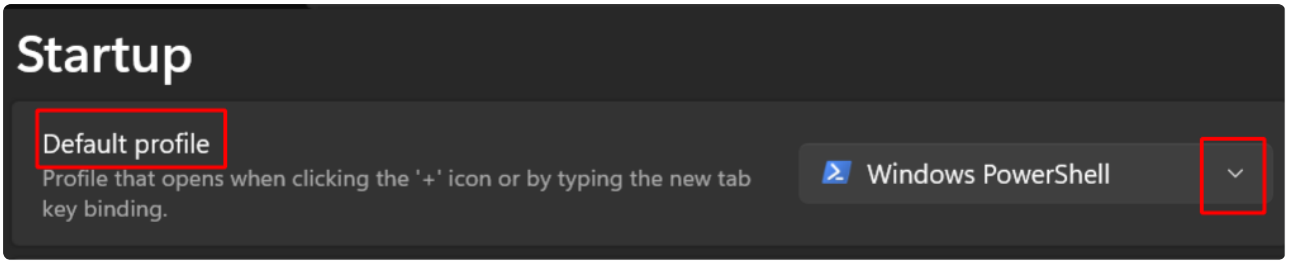
4.1.2 鼠标右键直接进入WinWSL（推荐）

在任意文件夹空白位置右键，点击“Open in Terminal”（在终端打开）。

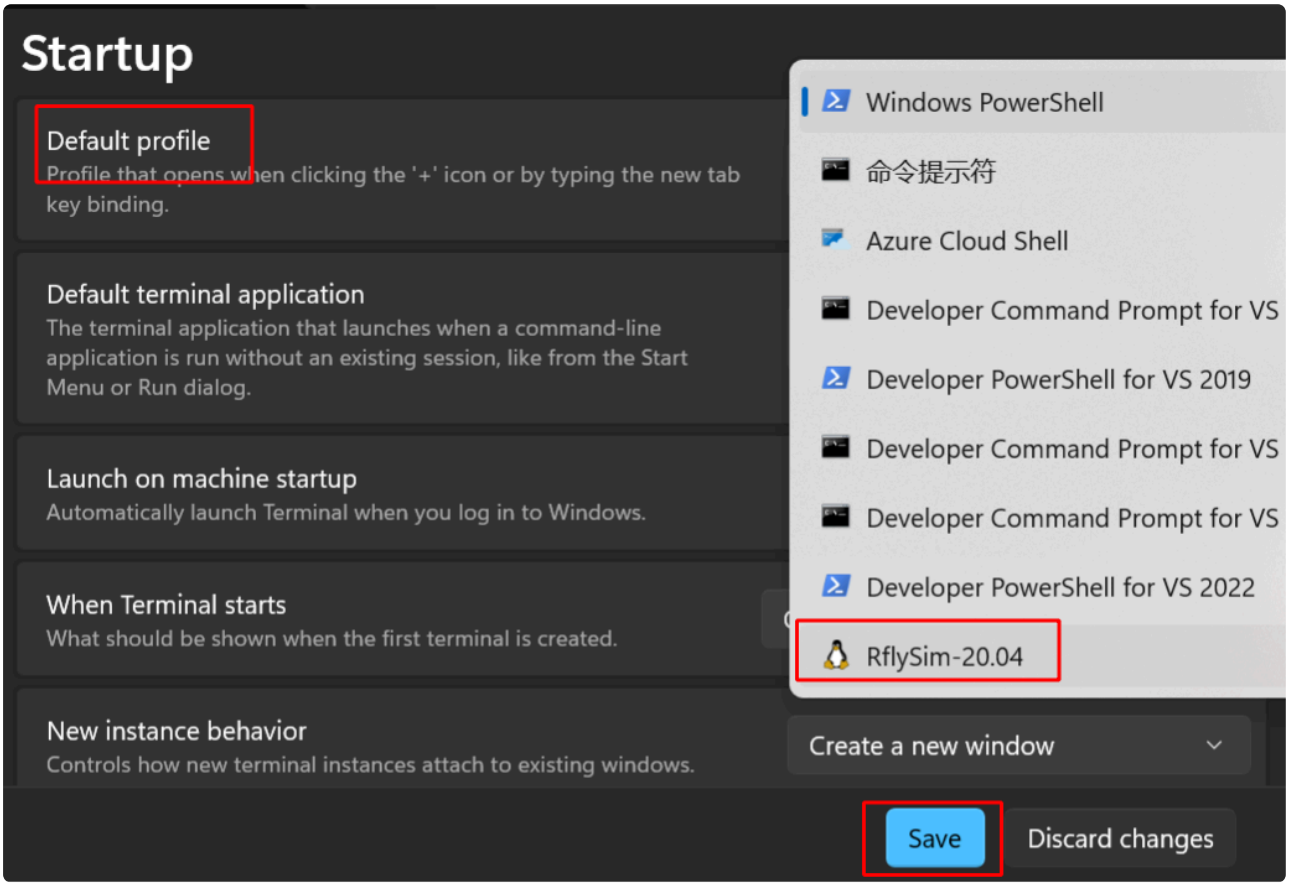
在弹出终端中，点击“下箭头” -> “Settings”，进入设置页面。



在“Startup” -> “Default profile”中点击下箭头。



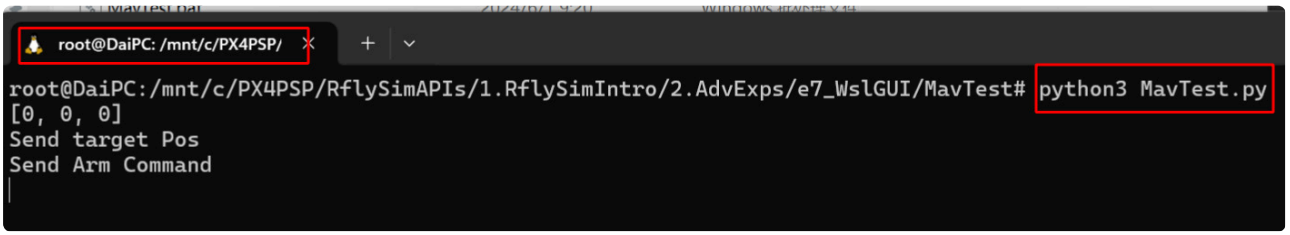
选择 `RflySim-20.04`，再点击“Save”保存，即可让 WinWSL 作为默认右键打开的终端。



测试方法：进入 `e7.WslGUI\MavTest` 目录，先双击 `MavTest.bat` 打开一个飞机的 SITL 仿真。然后在本例程目录中鼠标右键打开终端，此时会默认进入 WinWSL 环境，直接输入：

```
python3 MavTest.py
```

即可在 Ubuntu 环境下运行 `MavTest.py` 脚本，实现飞机的飞行控制。



4.1.3 桌面快捷方式进入WinWSL

双击桌面 `RflyTools` 文件夹下的 `WinWSL` 快捷方式。

默认会进入 `PX4PSP/Firmware` 目录。

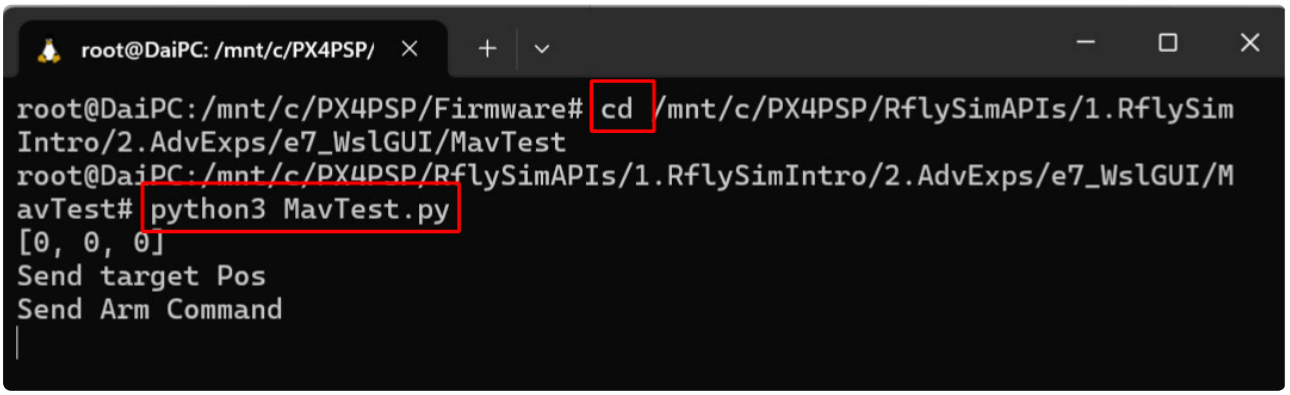
准备好要访问目录的全局路径，例如：

```
C:\PX4PSP\RflySimAPIs\1.RflySimIntro\2.AdvExps\e7.WslGUI\MavTest
```

将盘符 `C:` 去掉冒号并换成小写 `c`，加上 `/mnt/` 前缀，反斜杠换成正斜杠，得到 Linux 下访问 Windows 文件的路径：

```
/mnt/c/PX4PSP/RflySimAPIs/1.RflySimIntro/2.AdvExps/e7.WslGUI/MavTest
```

使用 `cd ***` 进入对应路径，再运行 `python3 MavTest.py` 即可运行程序。



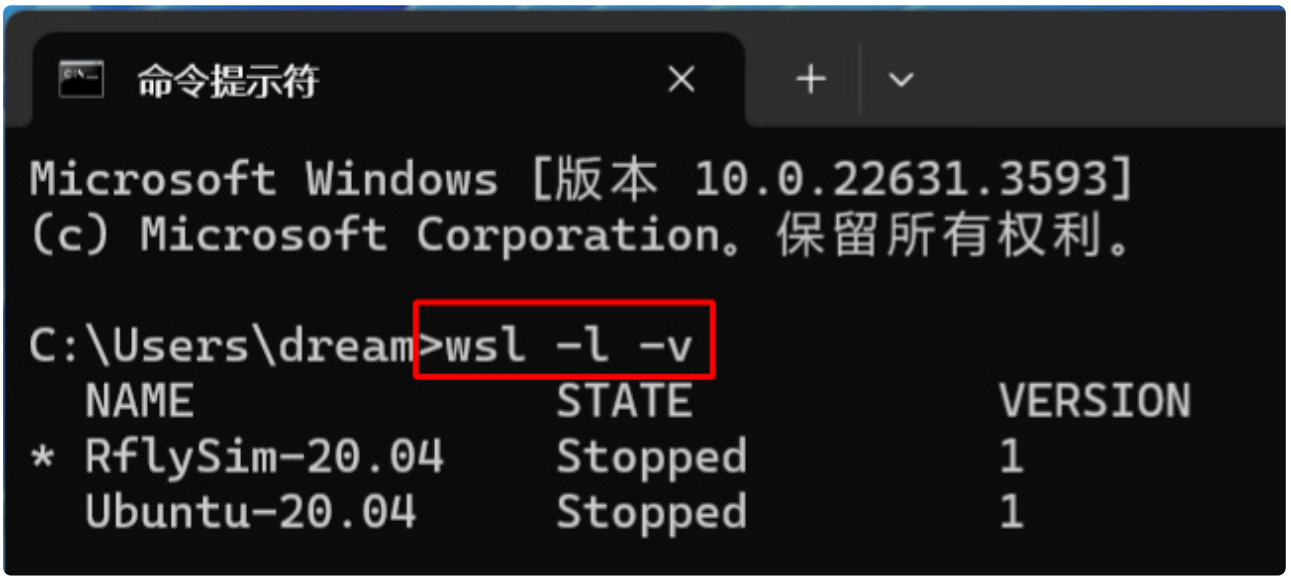
注意：`/mnt/` 是 Linux 下访问 Windows 文件的前缀。上述几种方法中，推荐使用 4.1.2 节的方法。

4.1.4 多WSL并存时启动WinWSL方法

如果电脑的 WSL 上部署了多个 Linux 虚拟机环境，那么输入 `wsl` 或 `bash` 可能会进入默认的 Ubuntu 系统。打开一个 cmd 命令提示符窗口，输入：

```
wsl -l -v
```

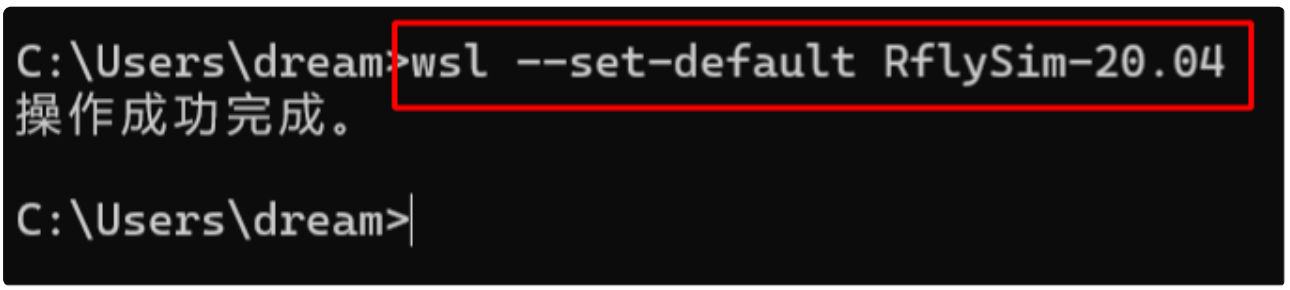
即可看到电脑上安装的所有 WSL 系统，其中 `*` 开头的是默认 Linux 系统。正常情况下，RflySim 工具链安装完成后，会导入 `RflySim-20.04` 虚拟机工具链，并将其作为默认 WSL。



切换 `RflySim-20.04` 为默认 WSL 系统的方法如下：

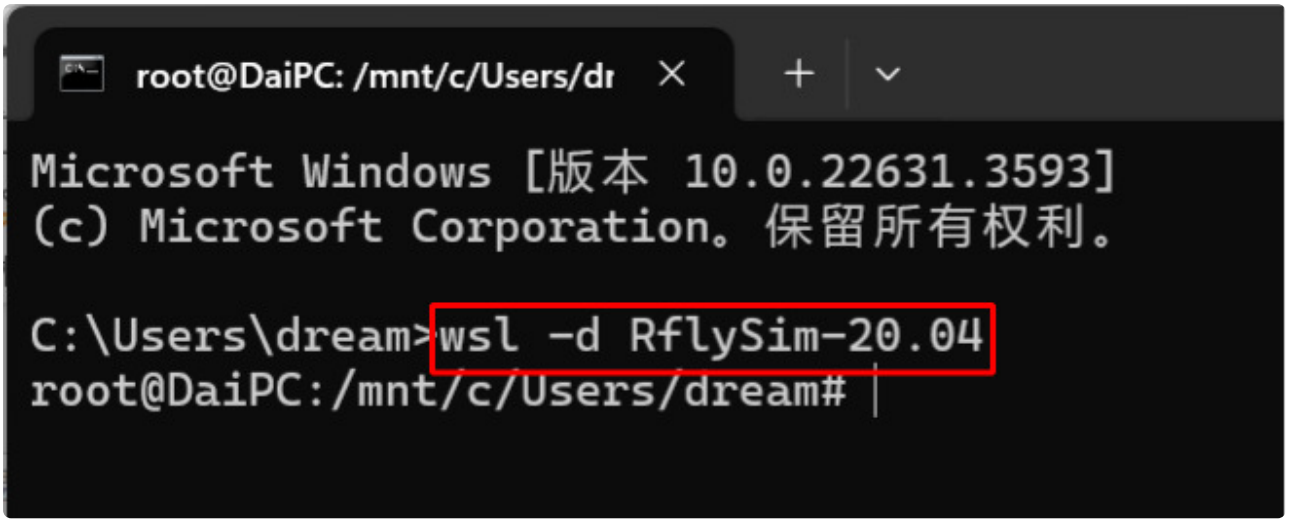
```
wsl --set-default RflySim-20.04
```

设置后，通过 `wsl` 或 `bash` 可以从终端或 cmd 窗口快速进入 RflySim 的 WinWSL 环境。



如果不切换默认 WSL 系统，但仍想使用 WinWSL 环境，则需要使用如下命令：

```
wsl -d RflySim-20.04
```



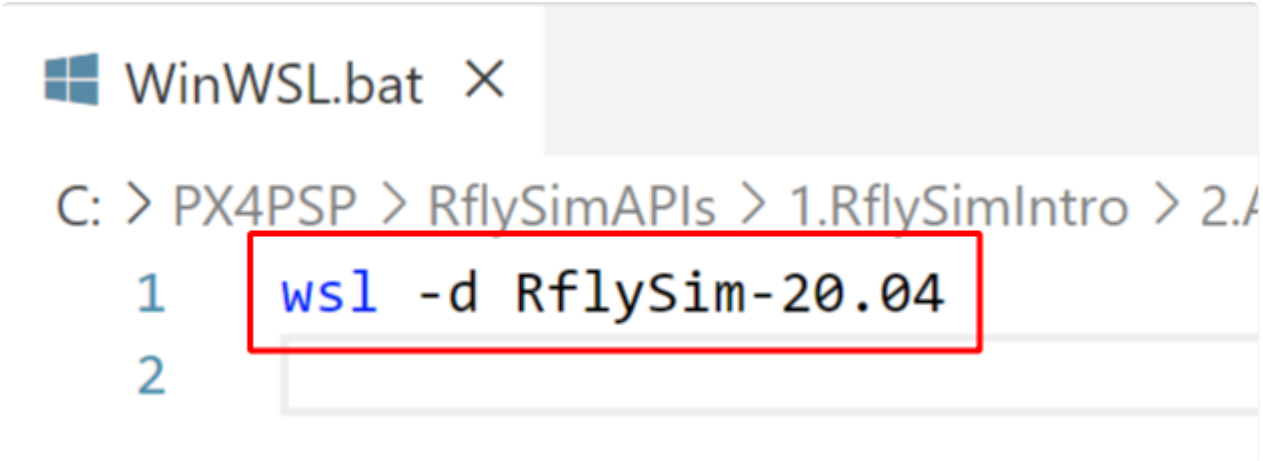
工具链所有 bat 脚本在调用 WSL 时都加入了上述指定发行版的语句，确保在多 WSL 并存时，也能正确访问 RflySim 的 WinWSL 编译器环境。右键用 VS Code 打开 `RflySimAPIs\SITLRun.bat`，搜索 `wsl`，可看到相关语句。

```
153
154    SET WINDOWSPATH=%PATH%
155    if %ToolChainType% EQU 1 (
156        wsl -d RflySim-20.04 echo Starting PX4 Build;
157    ) else (
158        REM CYGPath
159        cd /d %PSP_PATH%\CygwinToolchain
160        CALL setPX4Env.bat
```

4.1.5 双击WinWSL.bat进入WinWSL环境

某些例程文件夹中自带 `WinWSL.bat` 脚本，双击它即可快速进入工具链的 `RflySim-20.04` 环境，并进入当前文件夹。在此窗口中可以直接输入 Ubuntu 命令，等同于在 Ubuntu 虚拟机上运行。

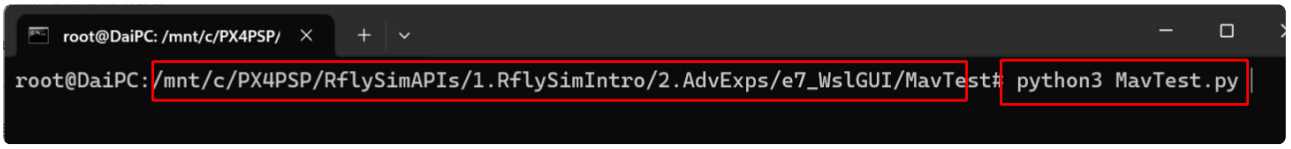
可拷贝或自行创建一个 `WinWSL.bat` 脚本文件，并在其中加入指定 WSL 版本和当前目录跳转的语句。



该脚本指定了 WSL 启动的 Ubuntu 版本，确保多 WSL 系统并存时也能跳转到 `RflySim-20.04`。

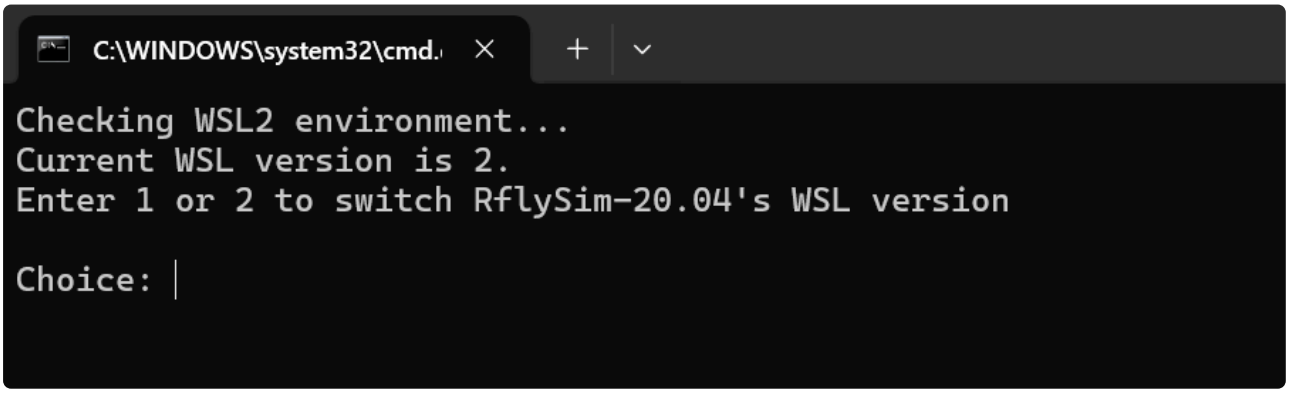
双击 `WinWSL.bat` 后即可快速进入工具链指定的 WinWSL 环境。例如在上文中的例程中，可以输入：

```
python3 MavTest.py
```



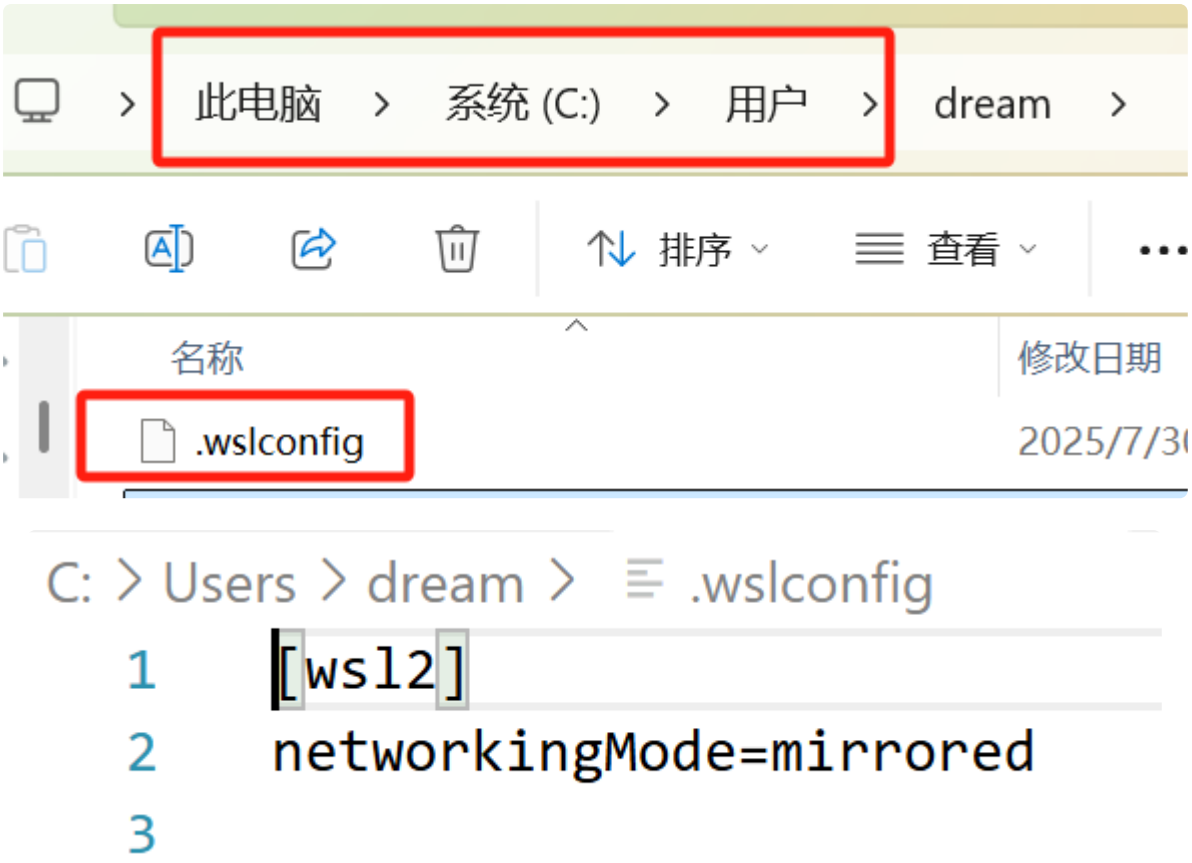
4.1.6 WinWSL在WSL1和WSL2之间切换

双击 `[桌面]\RflyTools\WslSwith2` 快捷方式，或 `[安装目录]\RflyTools\WslSwith2.bat`，即可根据提示在 WSL1 和 WSL2 之间切换。



在弹出的窗口中输入 `1` 或 `2`，即可将 WinWSL 环境切换为 WSL1 或 WSL2。

切换为 WSL2 时，会调用 Windows 官方命令 `wsl --set-version **** 2` 进行转换；同时在 `C:\Users\用户名` 目录下生成 `.wslconfig` 文件，并启用镜像网络 `mirror` 模式，确保 WSL2 和 Windows 共用网络，从而保证 PX4_SITL 和 CopterSim 能互联互通。



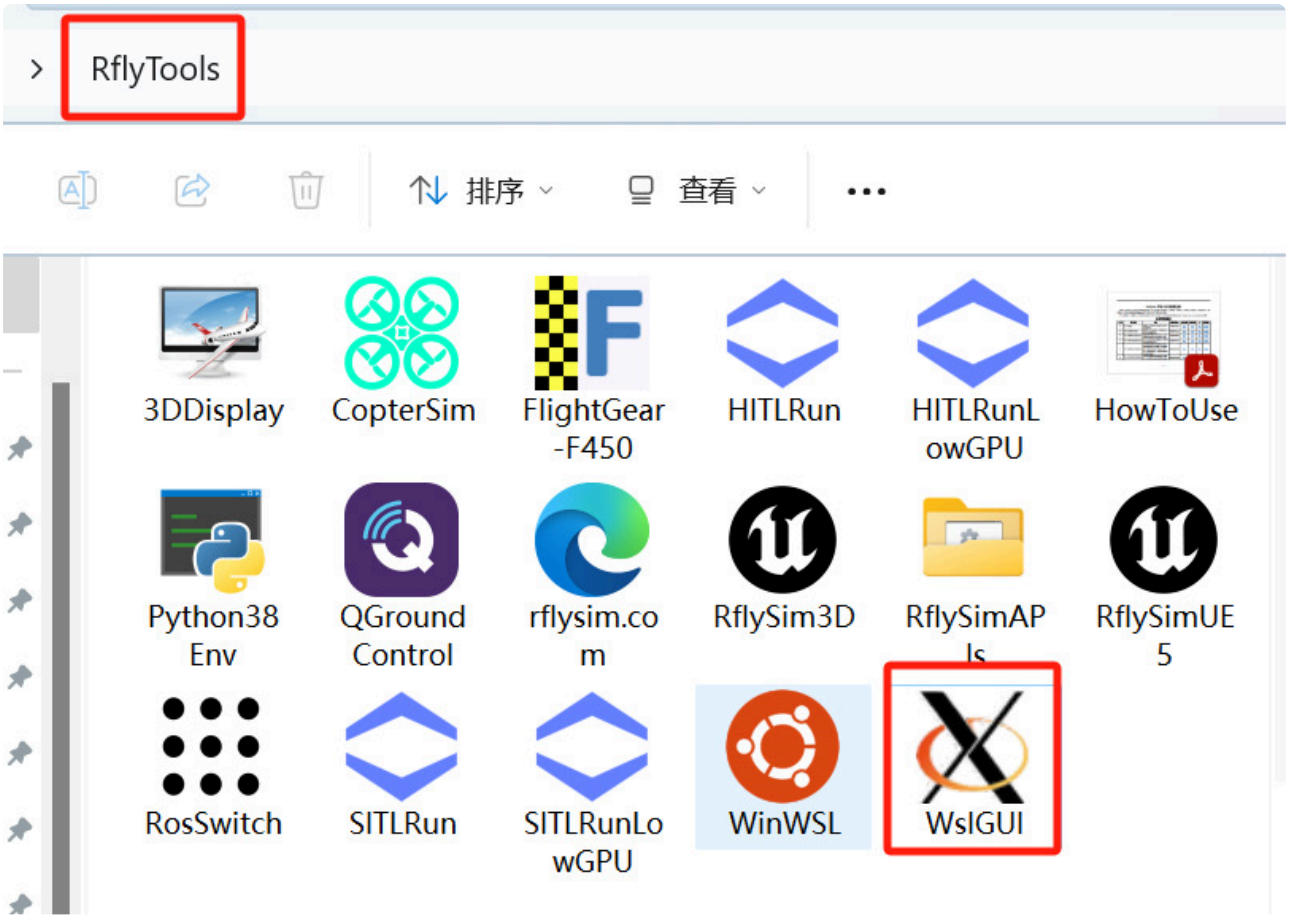
注意：该配置会对所有 WSL2 全局适用。如果电脑中存在多个 WSL2 版本，需要特别注意，避免影响本机其他 WSL 配置。

WSL1 和 WSL2 的主要区别在于架构、性能、文件系统和网络支持。WSL1 基于兼容层直接在 Windows 内核上运行 Linux 系统调用，启动速度快、资源占用较少，但文件系统 I/O 性能较弱；WSL2 通过虚拟化技术使用完整 Linux 内核，系统调用兼容性和 I/O 性能更好，也原生支持 Docker，但网络结构和资源占用更复杂。自 Windows 11 22H2 起，WSL2 支持镜像网络模式，可让 Linux 环境更好地与 Windows 网络栈交互。

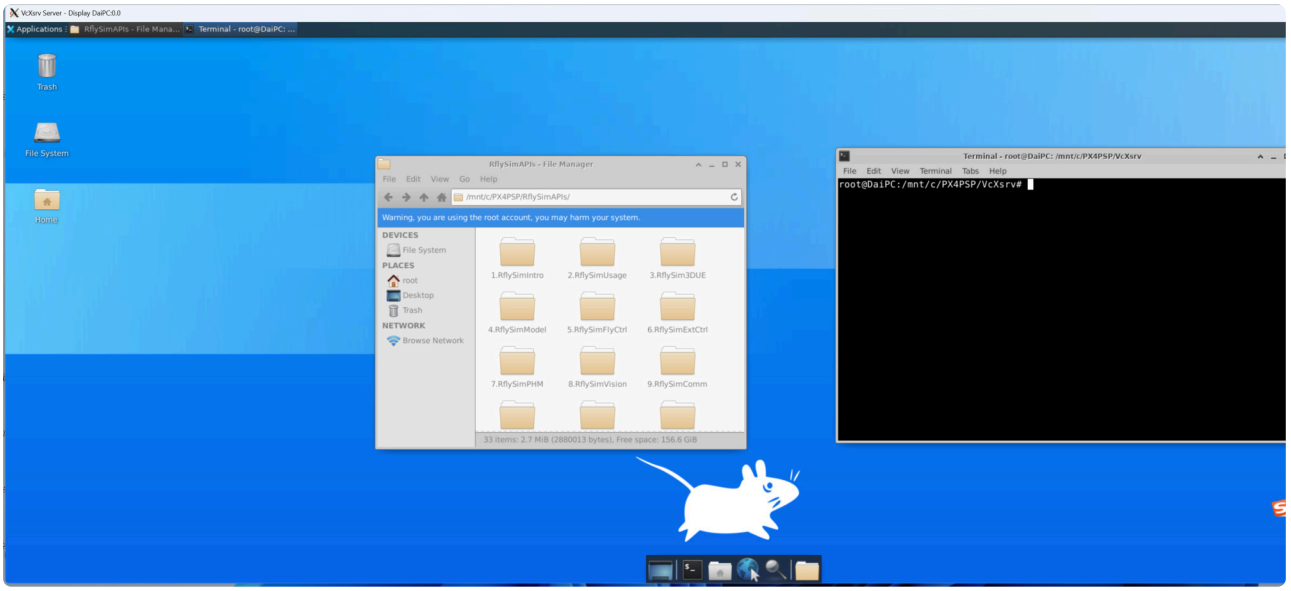
4.2 WslGUI的使用方法

4.2.1 基本使用方法

运行桌面 `WinWSL` 快捷方式，开启一个 Ubuntu 虚拟机终端。然后双击并运行桌面 `WslGUI` 快捷方式（对应 `C:\PX4PSP\VcXsrv\WslGUI.bat`）。

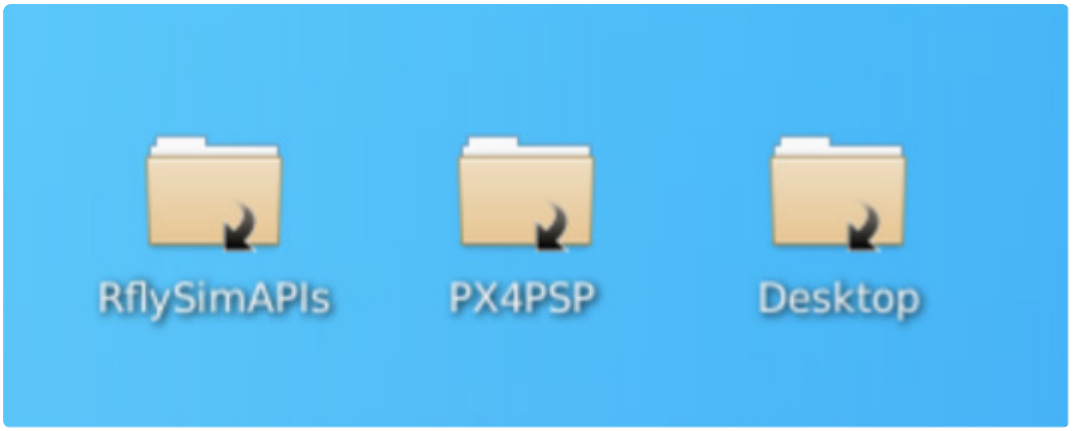


此时会弹出全屏 Linux 图形界面，可在其中进行界面操作，但运行可能略有卡顿。

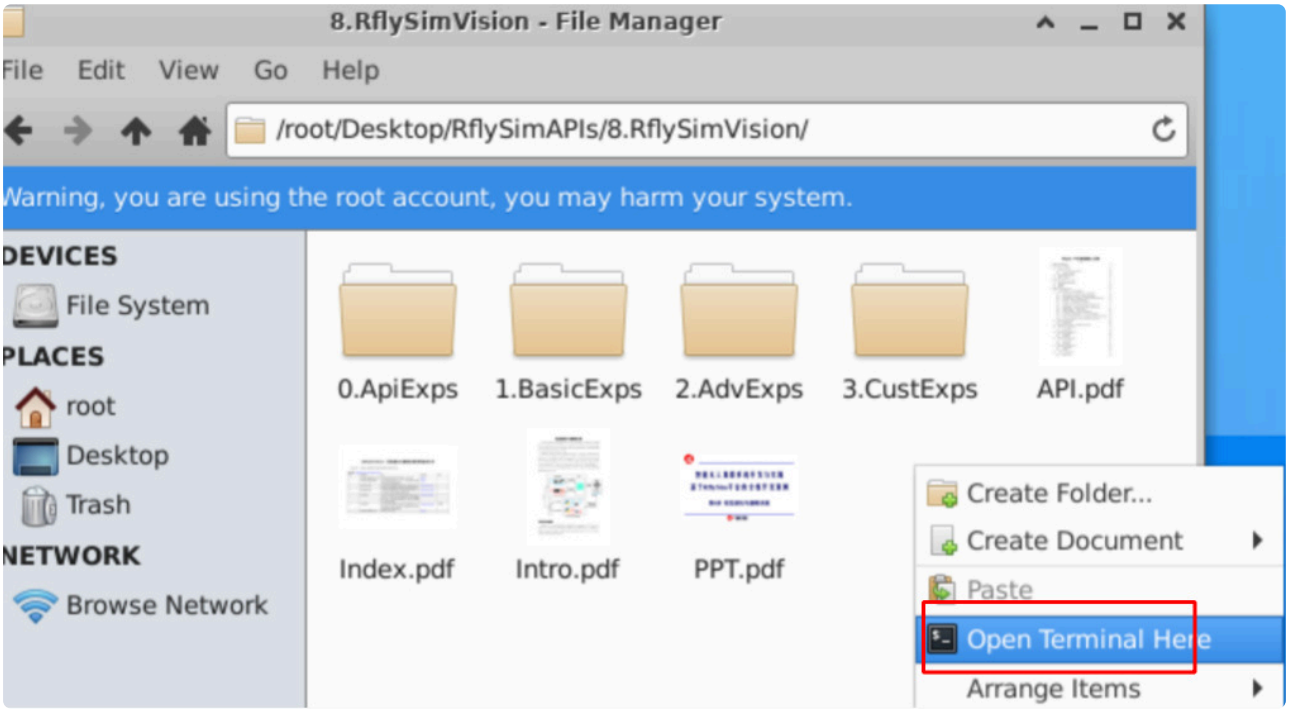


按 `Ctrl+Alt+T` 可以打开一个终端，之后按正常 Linux 命令行方式操作即可。

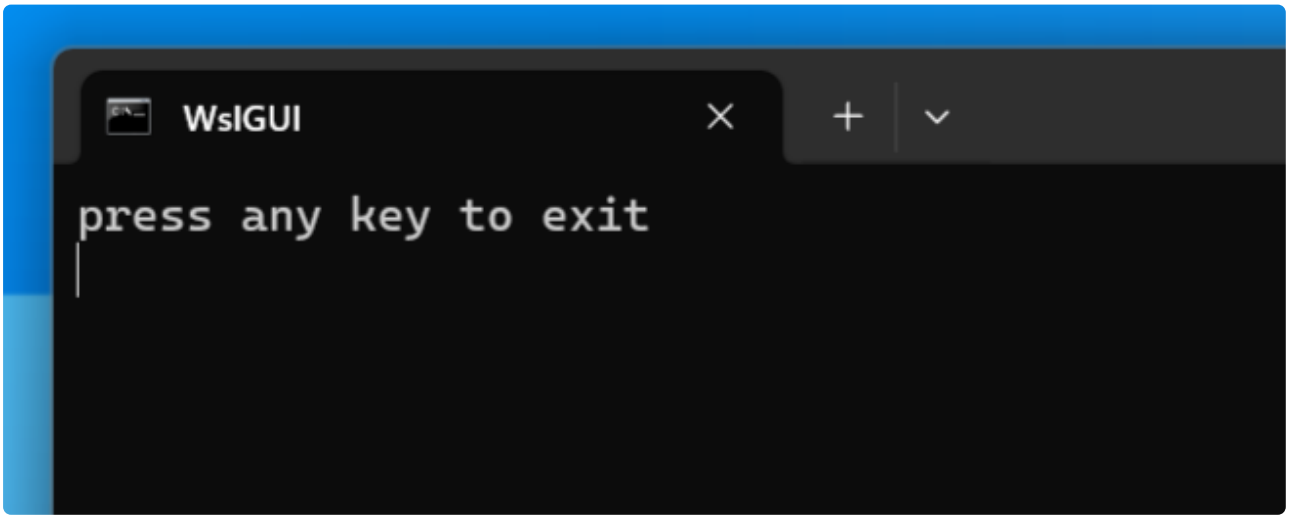
桌面上分别有 `RflySimAPIs`、`PX4PSP` 和 `Desktop` 三个文件夹，分别对应 RflySim 工具链例程目录、安装目录和用户桌面。



可以双击上述快捷方式进入自己的例程目录，在空白位置鼠标右键选择“Open Terminal Here”，即可在当前文件夹打开终端并执行自己的程序。

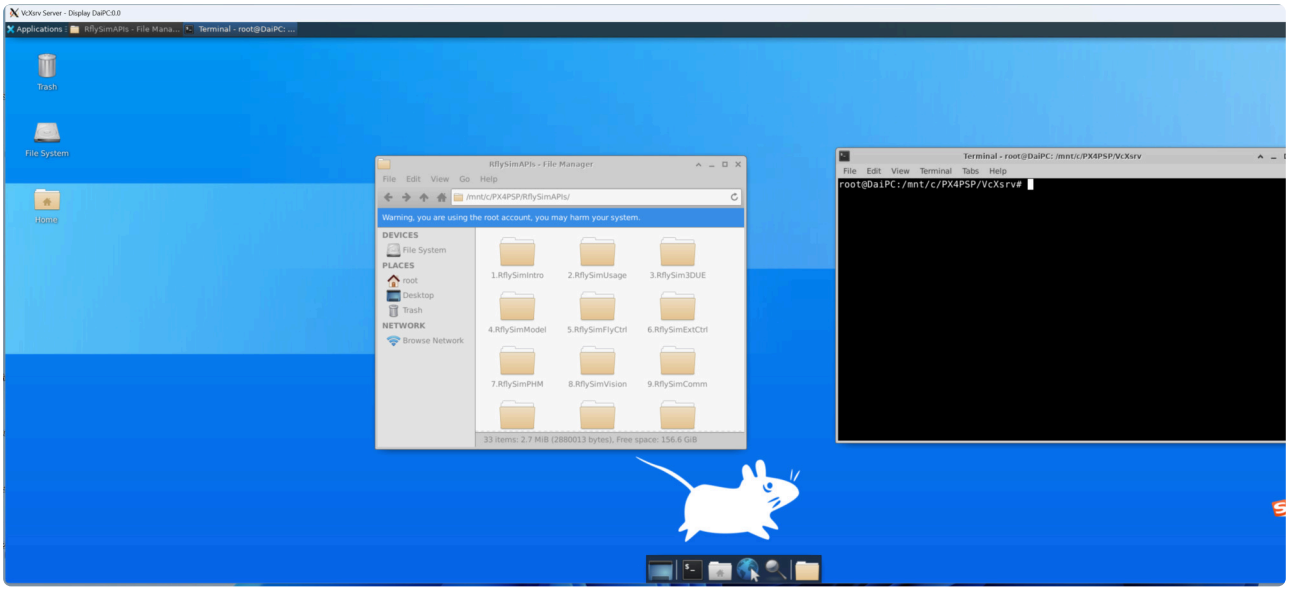


要退出 GUI，请打开 WslGUI 的黑色命令窗口，并按下任意键。



4.2.2 运行GuiTest例程

在 Windows 资源管理器中打开并进入 `e7.WslGUI\GuiTest` 目录，双击本文件夹的 `WslGUI.bat`，或双击桌面 `RflyTools\WslGUI` 快捷方式，即可打开 WinWSL 的 GUI 图形界面。



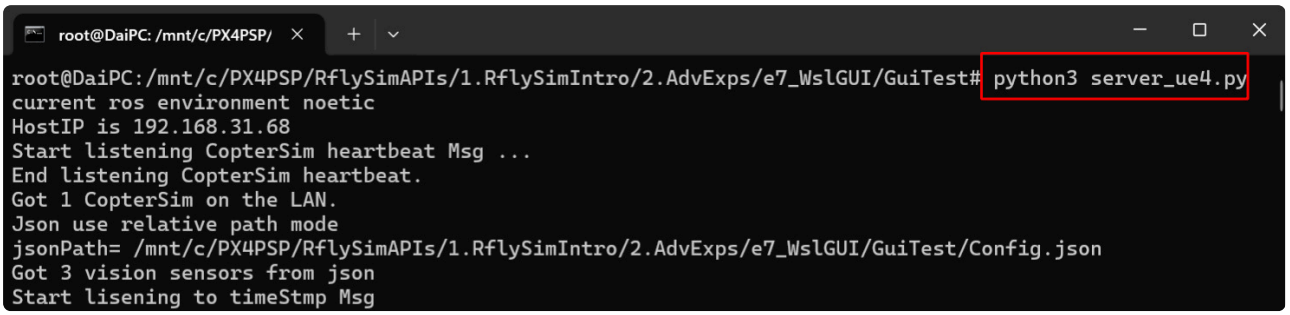
双击 `client_ue4_SITL.bat`，打开一个软件在环仿真飞机。

双击本文件夹的 `WinWSL.bat`，即可进入 `RflySim-20.04` 的 WinWSL 环境。也可以使用前文 4.1 节介绍的任意方式进入当前目录下的 WinWSL 环境。

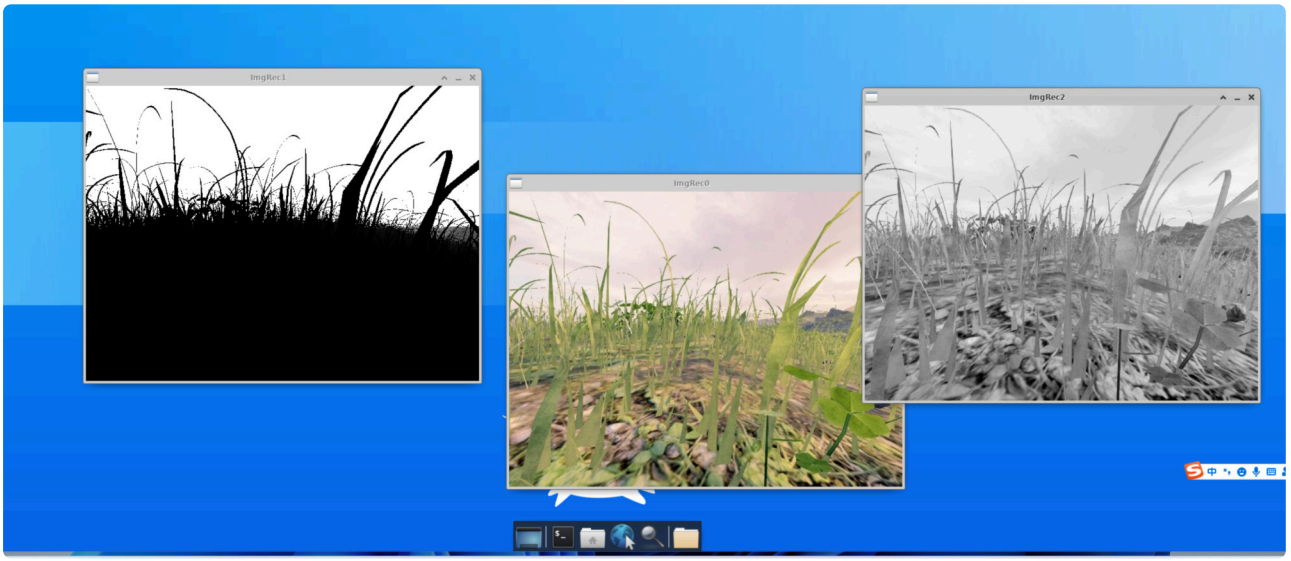
输入如下命令，即可在 Ubuntu 环境下运行 `server_ue4.py` 程序：

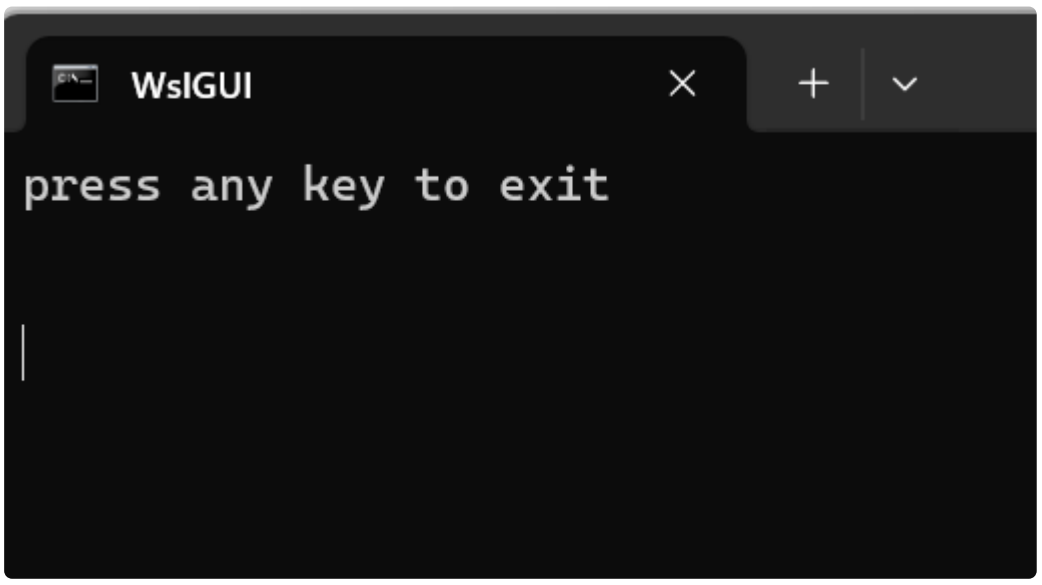
```
python3 server_ue4.py
```

该程序会将 Windows 下的图像通过模式3（UDP通信，JPEG压缩方式）发送给远端 Linux 系统，并进行显示。

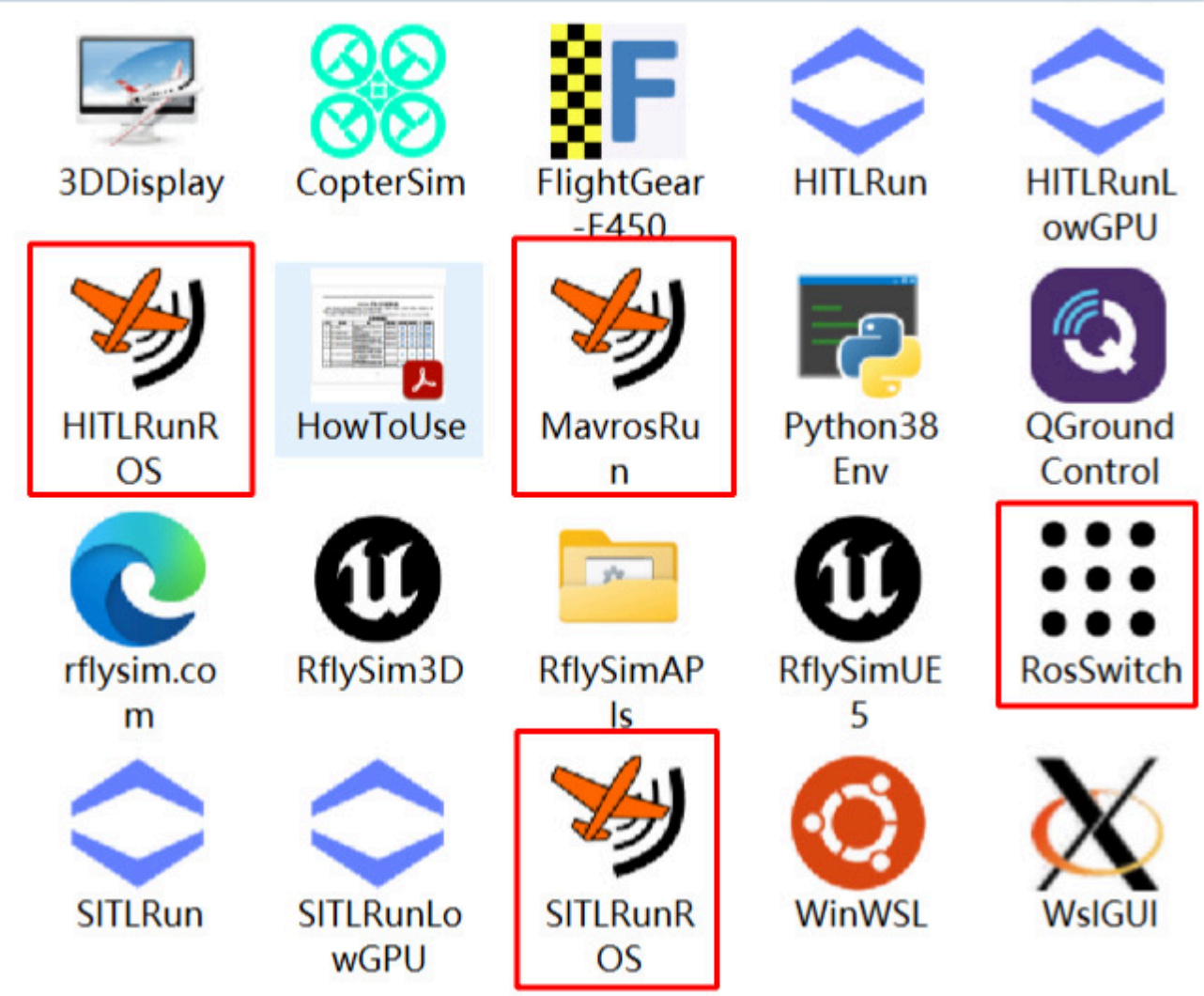


执行上述命令后等待一段时间，飞机会解锁起飞。几秒后，在 WinGUI 图形界面中的观察效果如下。





4.3 ROS开发相关功能



4.3.1 工具链ROS开发接口

工具链提供了 `RosSwitch`、`MavrosRun`、`SITLRunROS` 和 `HITLRunROS` 四个快捷方式，能够快速配置 ROS1/ROS2 环境，并启用带 ROS 的软件在环或硬件在环仿真。

快捷方式	简介	使用方法
`RosSwitch.bat`	WinWSL环境的ROS一键切换脚本	双击 `RosSwitch`，输入 `1` 或 `2` 来切换 ROS1 或 ROS2 环境。后续三个快捷方式使用的 ROS 版本会自动适配。
`MavrosRun.bat`	快速启动指定数量的 mavros 节点	先运行 `SITLRun` 或 `HITLRun` 开启指定数量的飞机（使用

快捷方式	简介	使用方法
		`MAVLink_Full` 通信模式），再运行 `MavrosRun.bat` 并输入飞机数量，即可开启对应数量的 mavros 节点。使用 ROS1 还是 ROS2 由 `RosSwitch` 指定。
`SITLRunROS.bat`	启动指定数量的 SITL 飞机，并开启对应 mavros 节点	是 `SITLRun + MavrosRun` 的组合脚本，效果相同，操作更便捷。
`HITLRunROS.bat`	启动指定数量的 HITL 飞机，并开启对应 mavros 节点	是 `HITLRun + MavrosRun` 的组合脚本，效果相同，操作更便捷。

4.3.2 多机仿真时mavros消息格式

工具链在只有一个飞机（`CopterID=1`）时，使用 `/mavros/` 前缀的消息名。当飞机数量大于等于 2 时，使用 `/mavros+飞机ID/*` 的消息格式，例如 `/mavros2/*`。

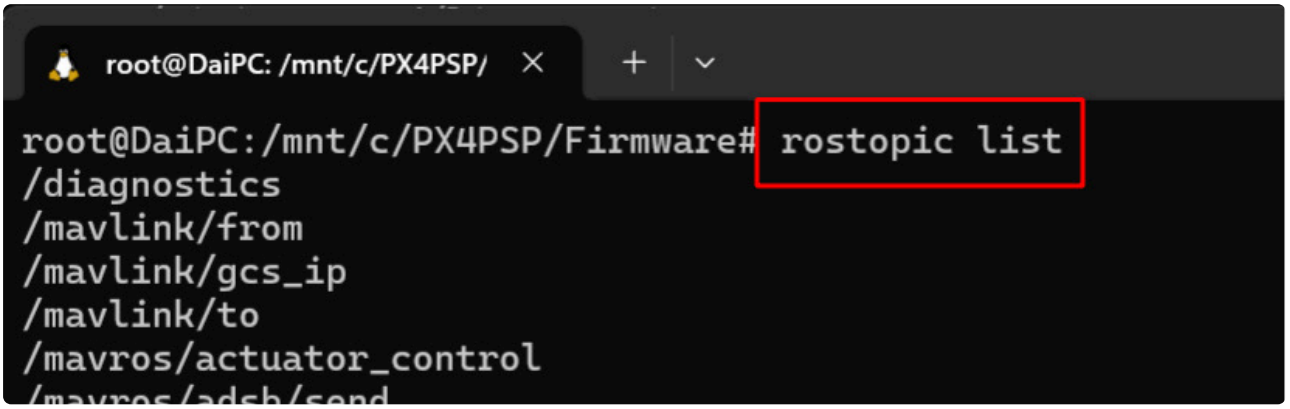
飞机ID	mavros消息
`CopterID=1`	`/mavros/****`
`CopterID=2`	`/mavros2/****`
`CopterID=* >=2`	`/mavros*/****`

运行桌面 `SITLRunROS` 快捷方式并输入 `2`，再双击 `WinWSL` 快捷方式，在其中输入：

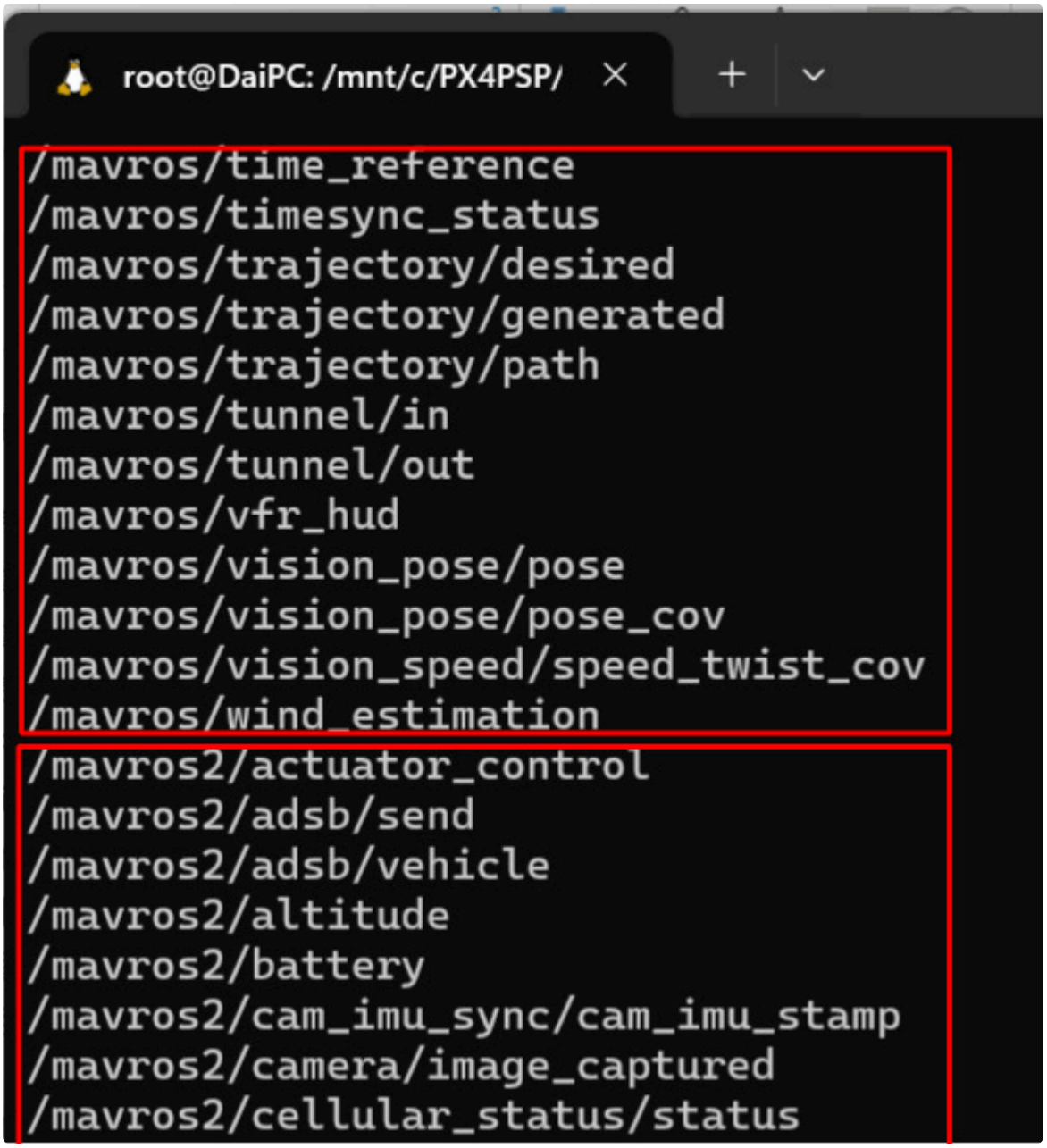
```
rostopic list
```

如果是 ROS2 环境，则输入：

```
ros2 topic list
```



输出结果如下图所示，1 号飞机的话题以 `mavros` 开头，2 号飞机以 `mavros2` 开头。在后续 ROS 开发中，只需订阅和发布正确的话题，即可实现飞机控制。



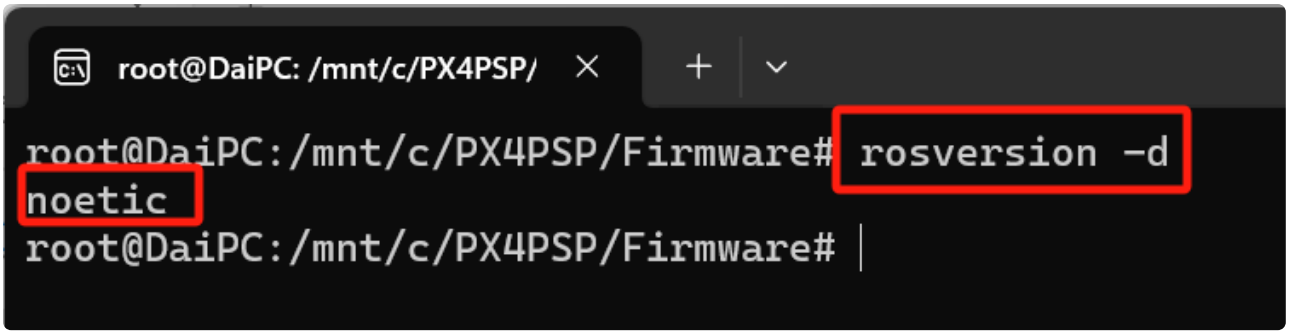
4.3.3 ROS1+ROS2环境与切换方法

工具链默认使用 ROS1 连接，且配备了 mavros 通信环境。双击打开桌面 `RflyTools\SITLRun.bat` 快捷方式，输入 `1` 并回车，快速启动一个基于 MAVLink 通信的软件在环仿真。

在桌面双击 `WinWSL` 快捷方式，并输入：

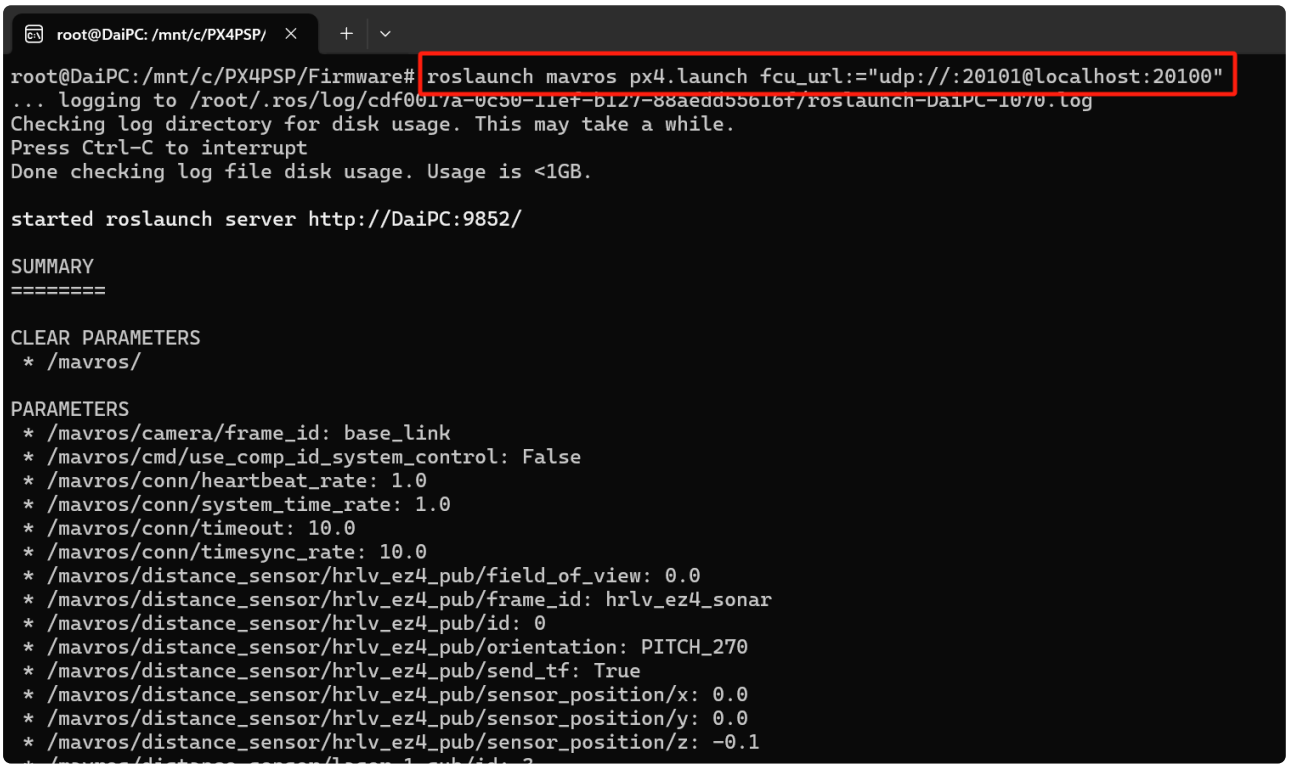
```
rosversion -d
```

如果输出 `noetic`，说明当前为 ROS1 环境。

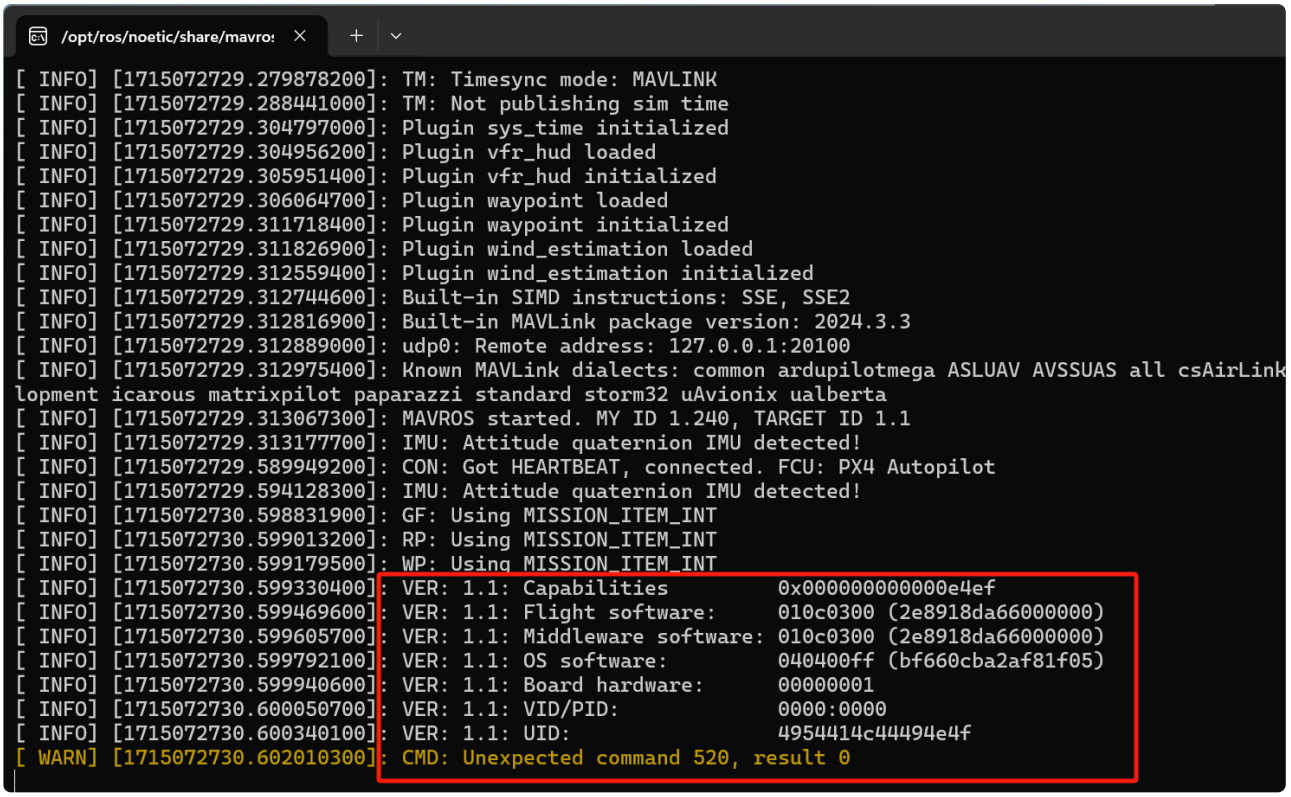


接着在 WinWSL 中输入如下命令，或运行 `MavrosRun` 快捷方式并输入 `1`：

```
roslaunch mavros px4.launch fcu_url:="udp://:20101@localhost:20100"
```



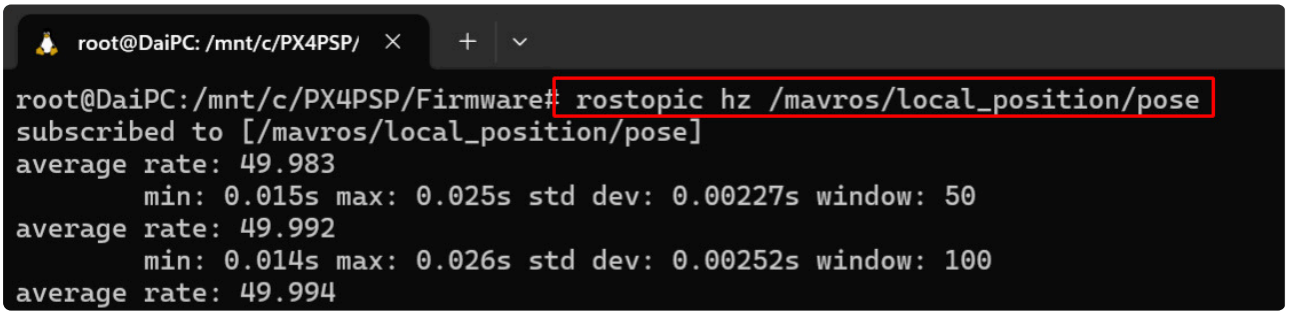
若能看到硬件版本等信息（包括 VID/PID 等），说明配置正确。



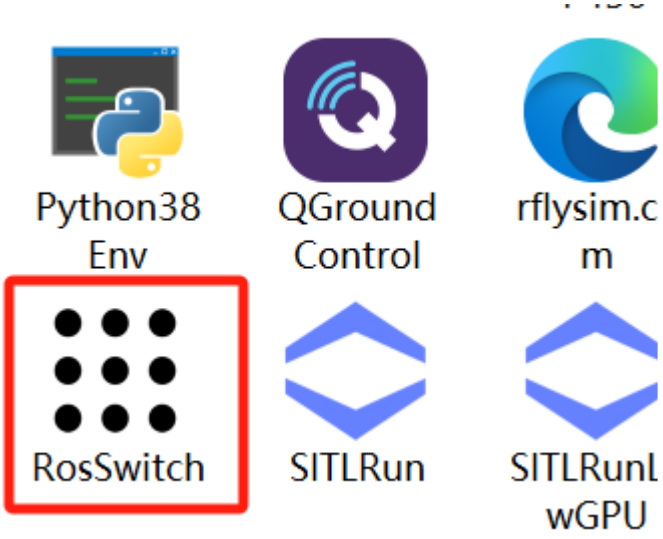
新打开一个 WinWSL 窗口，并输入：

```
rostopic list
rostopic hz /mavros/local_position/pose
```

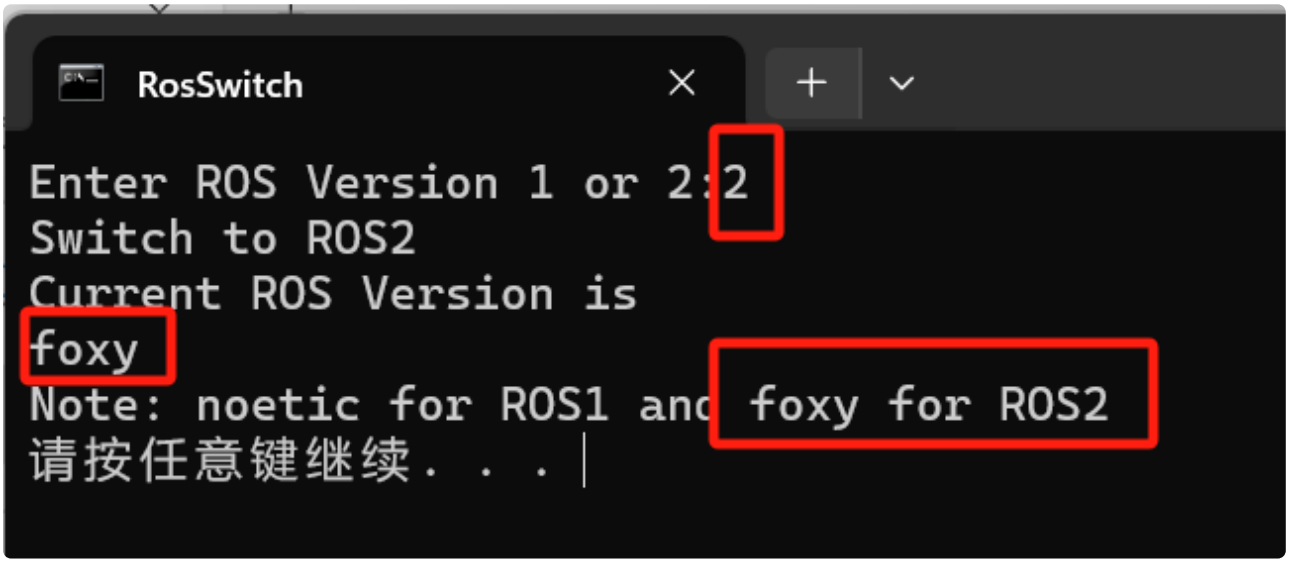
若能看到以 `mavros` 开头的消息，并且本地位置消息频率接近 50 Hz，说明连接正常。



下面切换 ROS2 环境。双击桌面快捷方式 `RosSwitch` 。



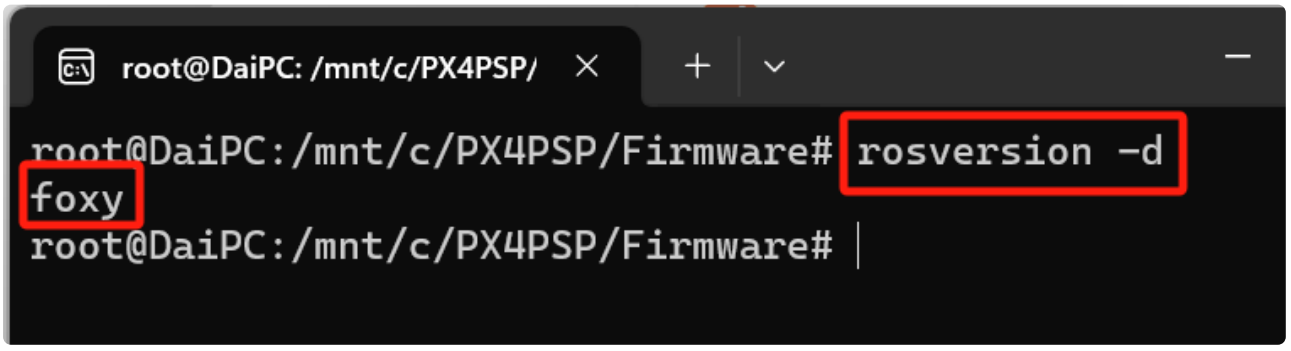
输入选项 `2` 完成切换。切换完成后，按任意键退出窗口。



重新打开 WinWSL，并输入：

```
rosversion -d
```

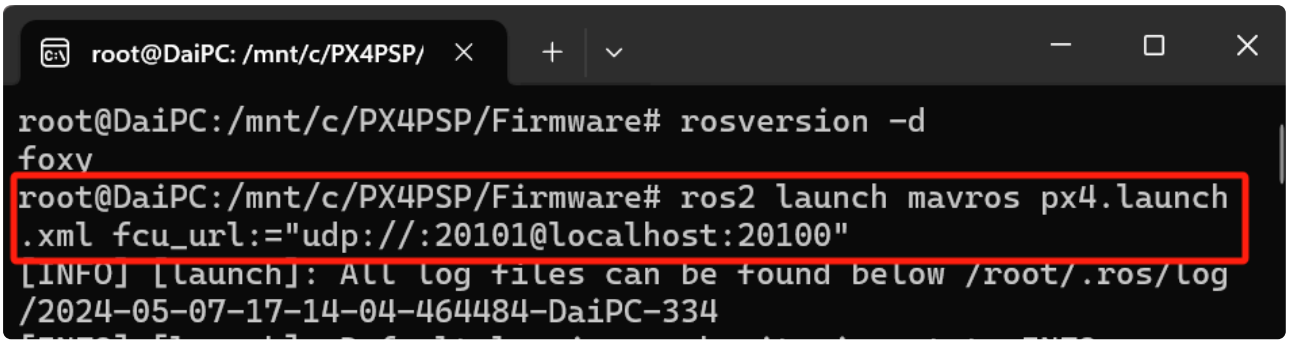
如果输出 `foxy`，说明当前为 ROS2 环境。



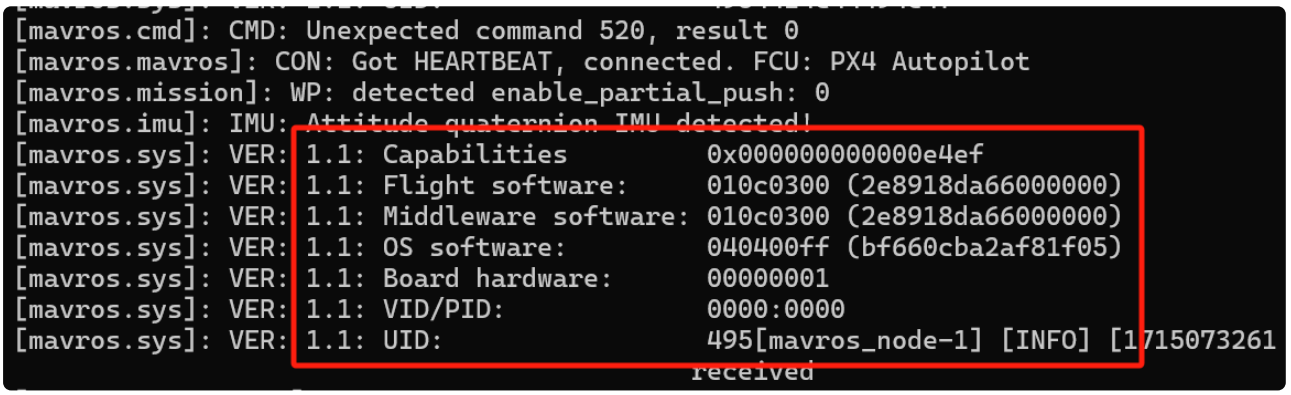
关闭前面的所有 SITL 和 WSL 窗口。运行桌面 `RflyTools\SITLRun.bat`，输入 `1` 并回车，重新启动一个基于 MAVLink 通信的软件在环仿真。

打开 WinWSL，再输入如下 ROS2 命令，或运行 `MavrosRun` 快捷方式并输入 `1`：

```
ros2 launch mavros px4.launch fcu_url:="udp://:20101@localhost:20100"
```



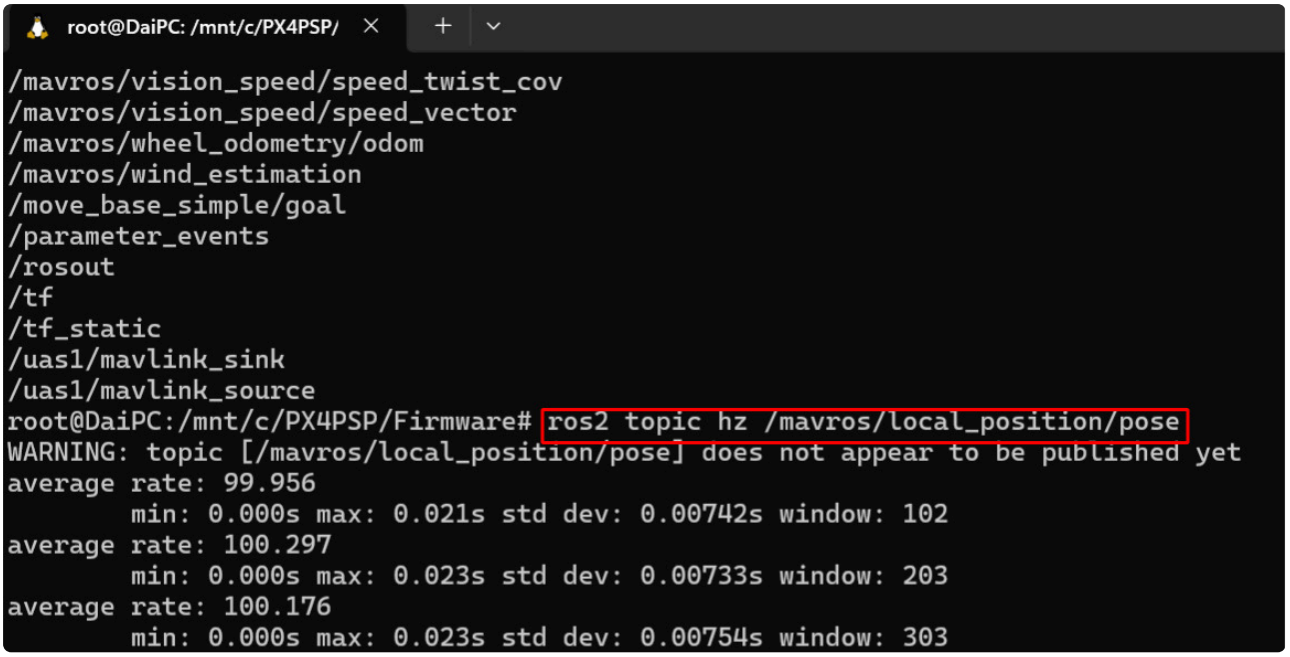
若能看到硬件版本等信息（包括 VID/PID 等），说明 ROS2 配置与连接正确。



新打开一个 WinWSL 窗口，并输入：

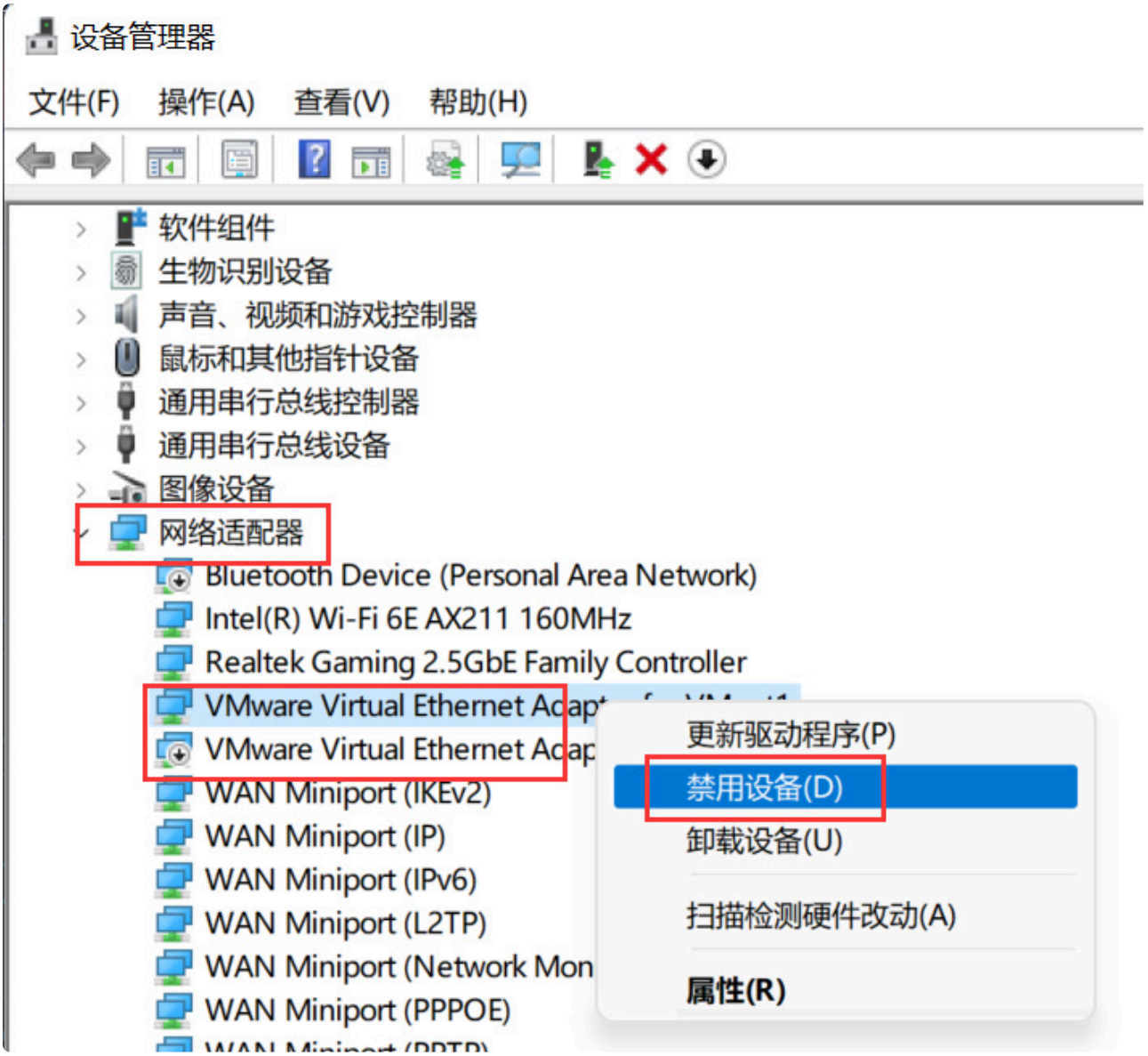
```
ros2 topic list
ros2 topic hz /mavros/local_position/pose
```

若能看到以 `mavros` 开头的消息，并能够订阅和打印本地位置消息，说明连接正常。



关闭所有程序，再次运行 `RosSwitch`，切回 ROS1，避免干扰后续实验运行。

注意：如果 ROS2 实验运行 `ros2 topic list` 命令时无法获取以 `mavros` 开头的消息，请尝试在设备管理器中禁用除当前使用网络适配器之外的其他适配器，例如 VMware 虚拟机适配器或其他有线/无线网卡适配器，然后重启电脑后再尝试运行本例程。

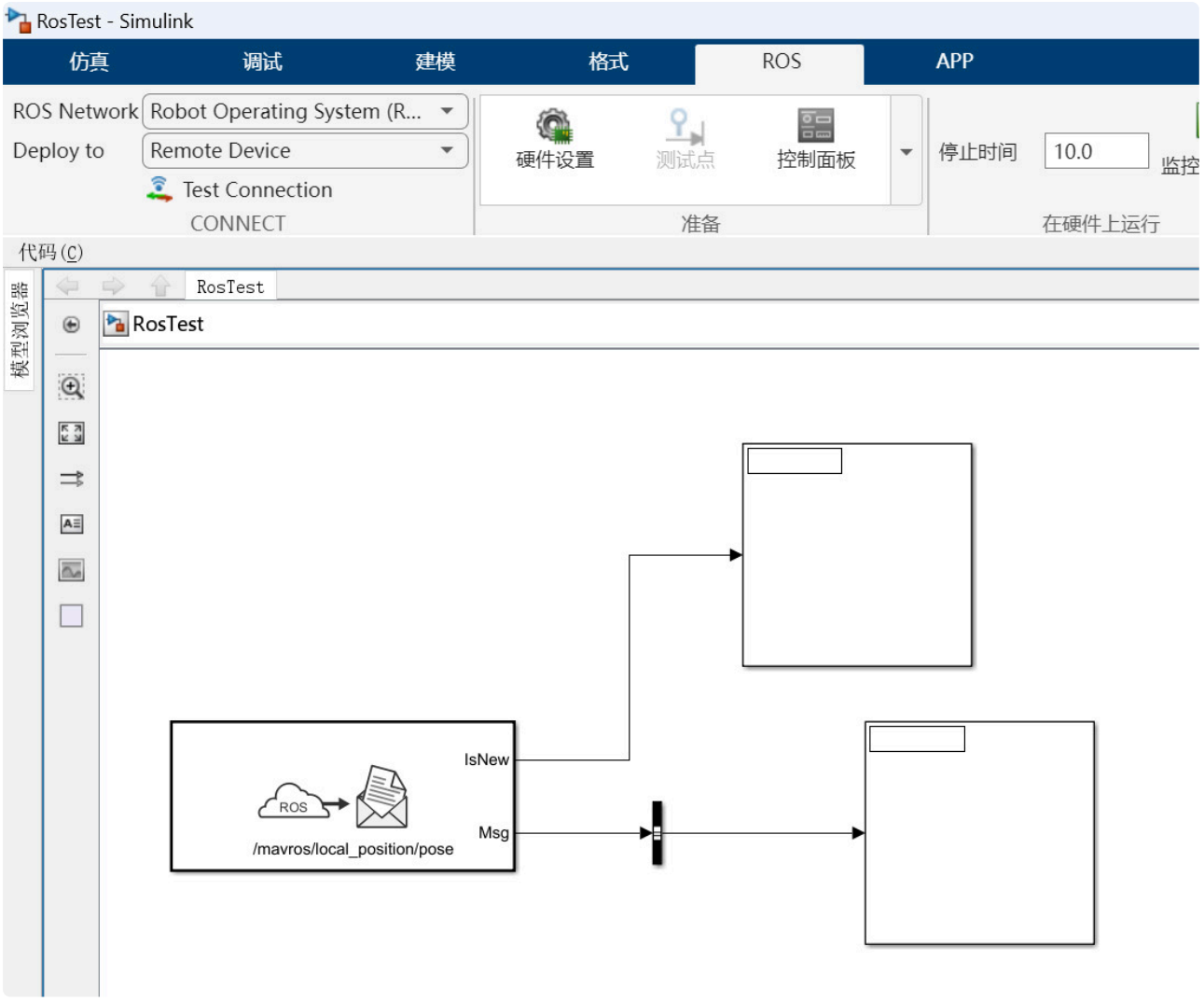


4.4 ROS和Simulink的互联互通

4.4.1 ROS与Simulink通信

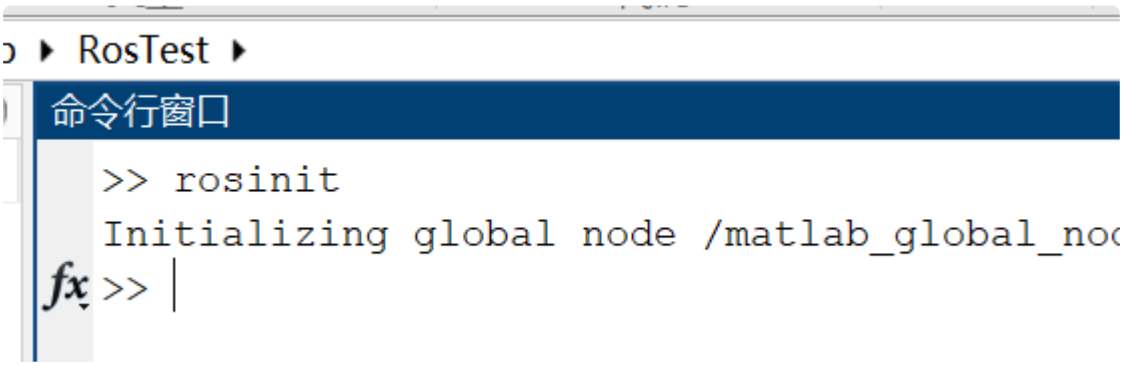
进入 `e7.WslGUI\RosTest` 例程目录，运行 `SITLRunROS.bat` 并输入 `1`，启动一个 SITL 四旋翼并连接 mavros。注意：本例程需要先切换回 ROS1 环境。

打开 `RosTest\RosTest.slx` 程序，注意需要使用 MATLAB 2022a 或以上版本。该程序调用 ROS 模块，订阅 `/mavros/local_position/pose` 消息。



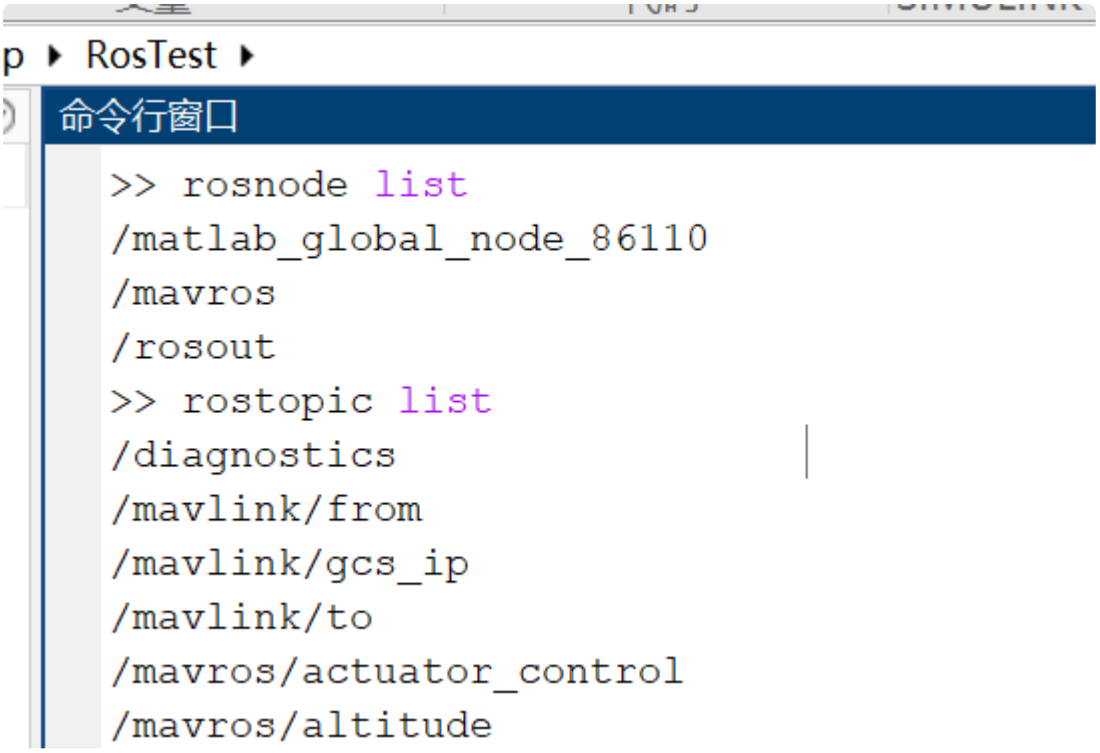
回到 MATLAB，输入如下命令。注意，仅 MATLAB 第一次运行 ROS 程序时需要执行；后续只要 master 节点 IP 没有变化，就不再需要运行。

```
rosinit
```

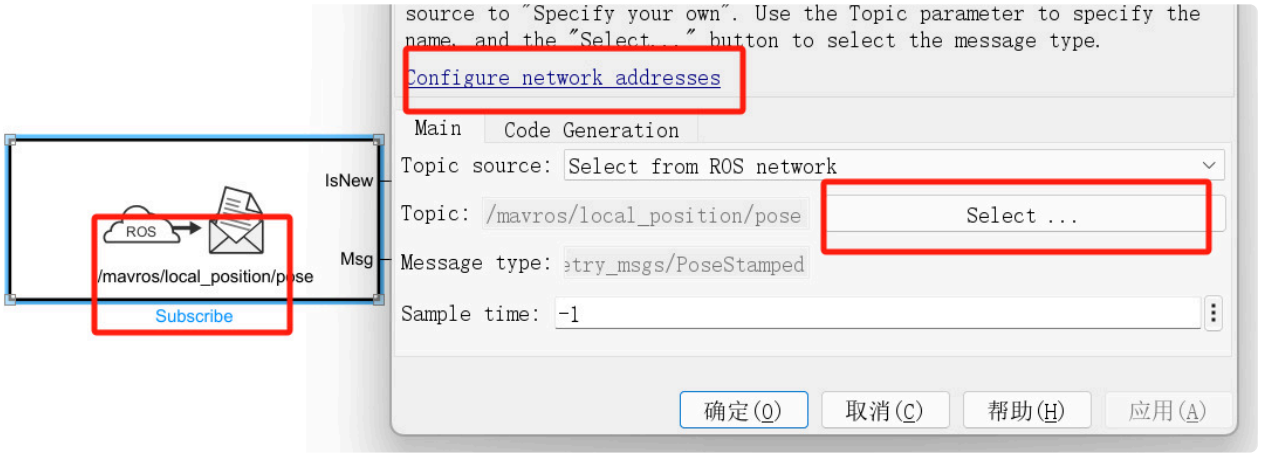


在 MATLAB 窗口输入如下命令，检查能否订阅到 mavros 消息。上述命令也可以在 WinWSL 中输入，效果相同。

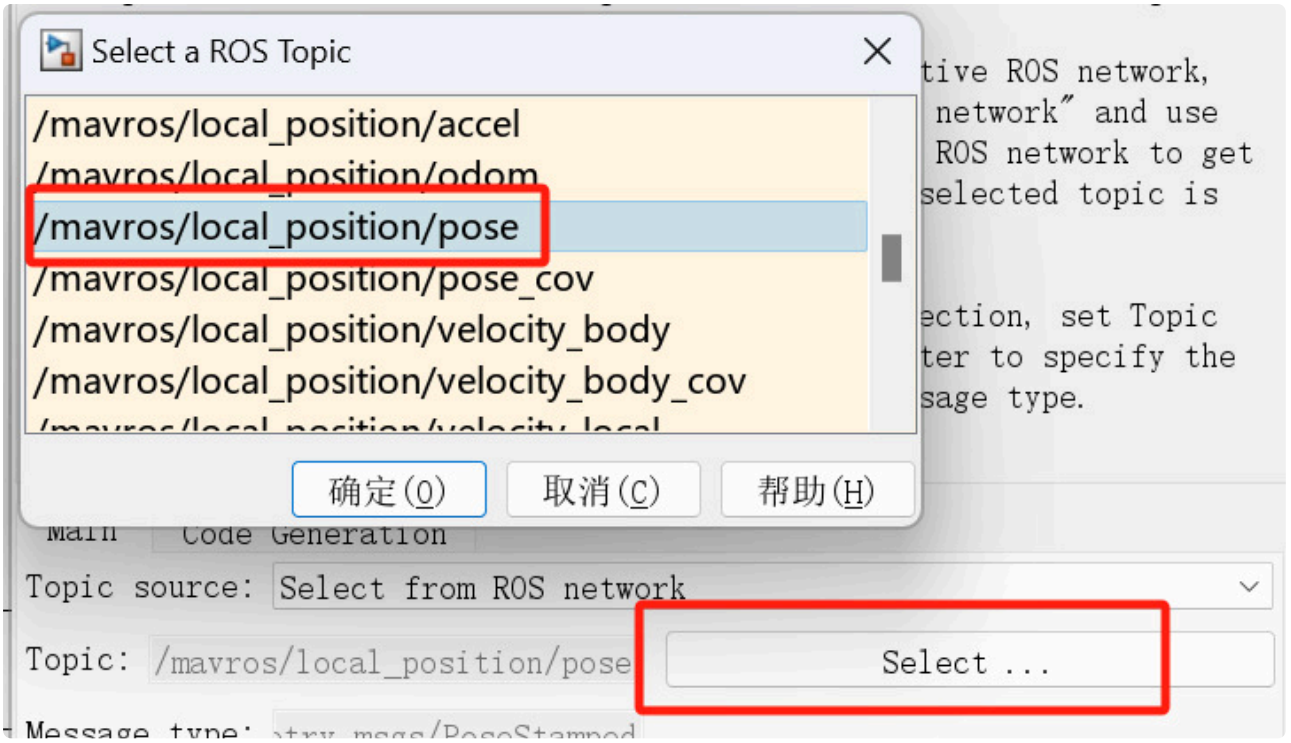
```
rosnode list
rostopic list
```



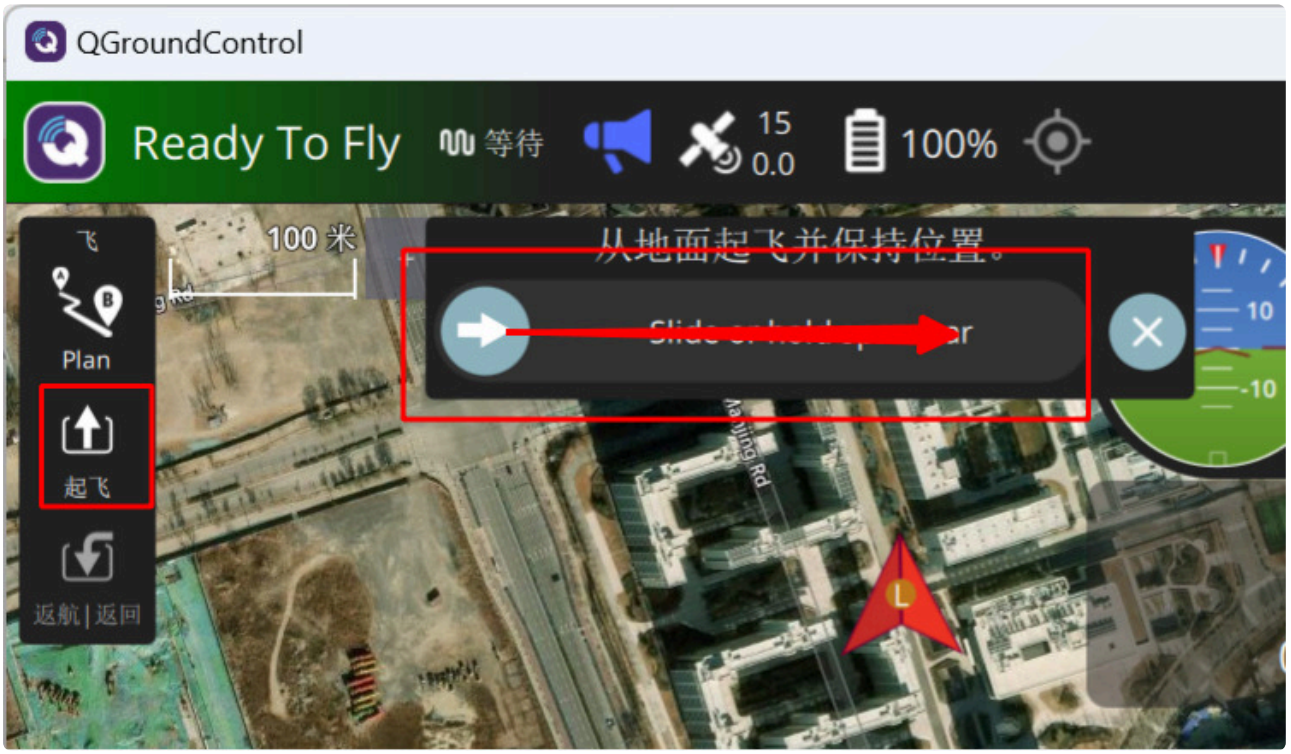
在 Simulink 中，双击 `Subscribe` 模块，可以进入 mask 封装页面。



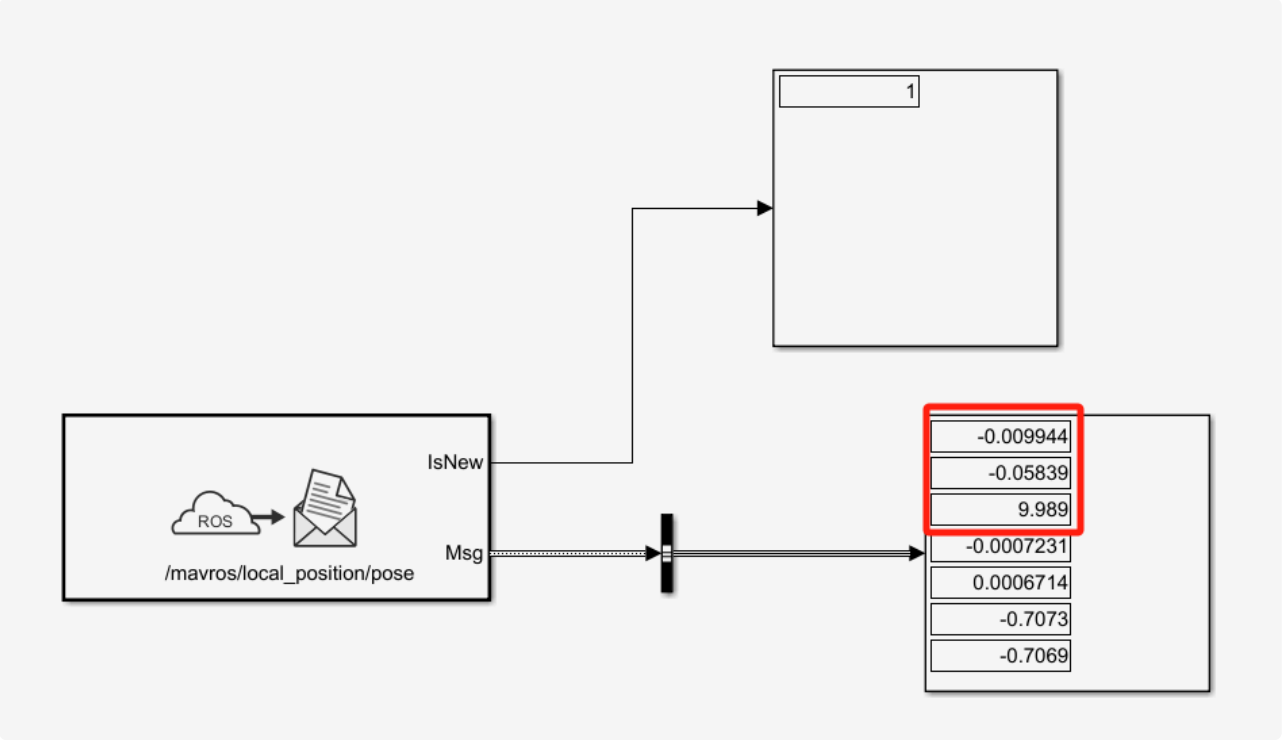
点击 `select...` 选项，可以直接选择需要订阅的 ROS 消息。本例程不需要修改，使用 `local_pose` 即可。实际开发时，可根据需要修改成感兴趣的数据。



在 QGC 中控制飞机起飞，默认会到 10 米高。如下图，点击“起飞”按钮，再滑动“箭头”从左到右确认操作。



然后运行 Simulink 程序，可以看到高度位置符合预期。注意：mavros 的坐标系不是 NED，而是 ENU，也就是向上为正，开发时需要特别注意。

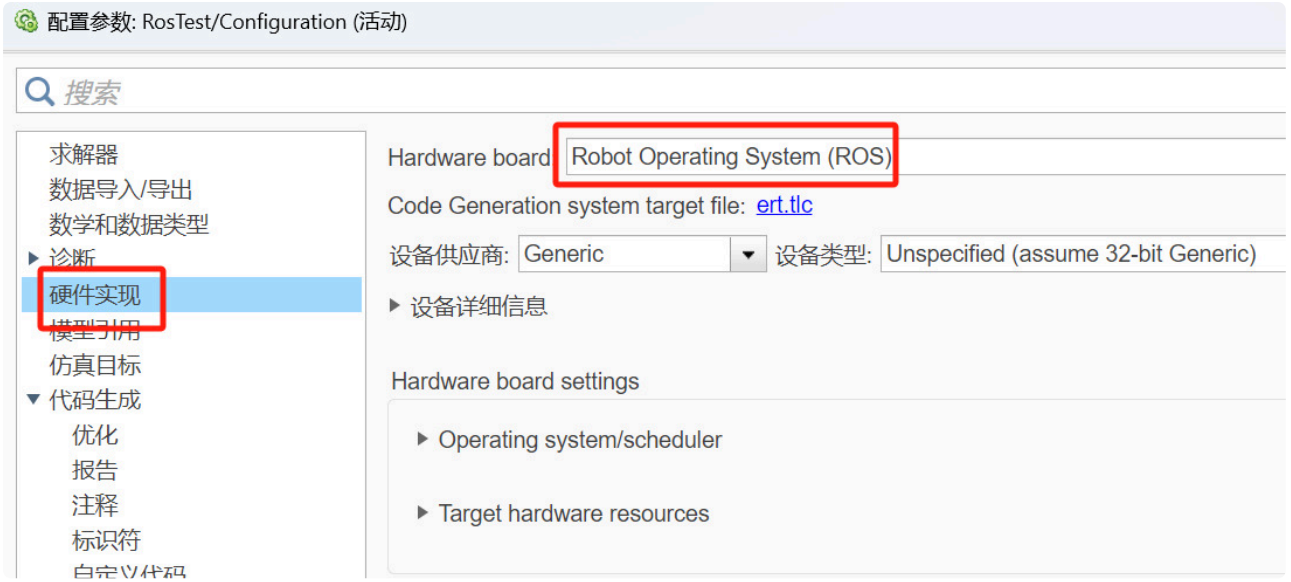


4.4.2 Simulink生成代码到ROS环境

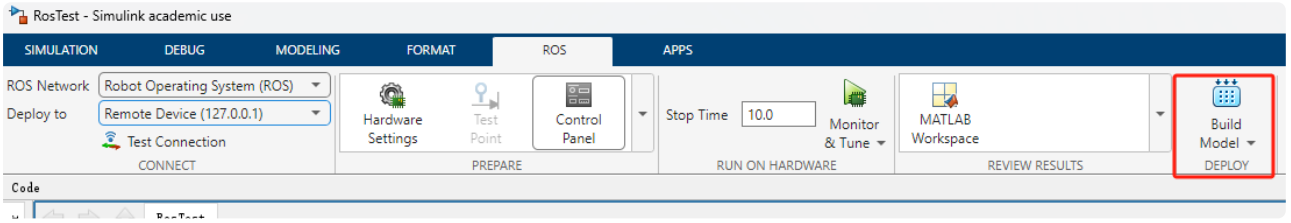
Simulink/ROS 开发的代码可以直接生成 C++，并在 ROS 环境下进行运行测试。

为了生成代码，需要进行如下配置：在“求解器”中选择“定步长”，并设置步长。

在“硬件实现”中选择 ROS 系统。

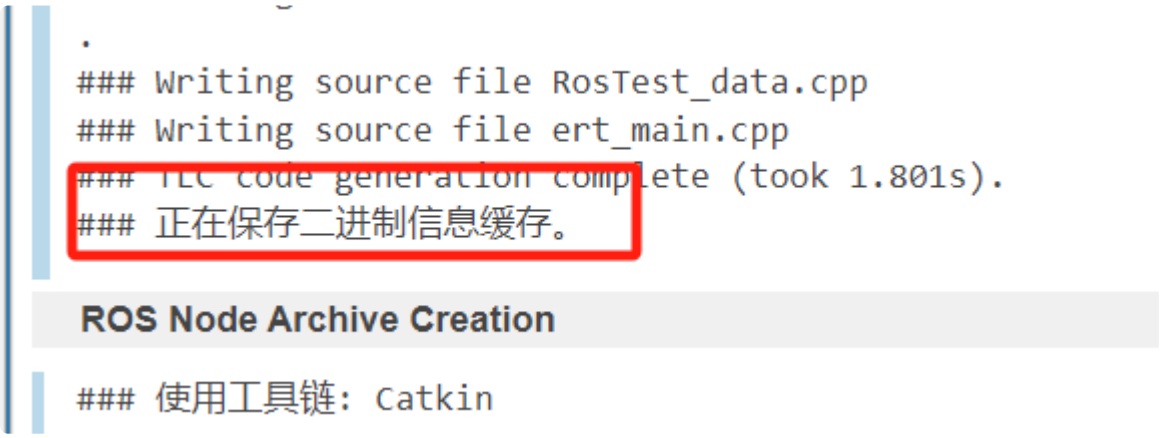


其他选项根据需求配置。在 ROS 页面会自动设置默认连接，然后点击 `Build` 编译按钮。

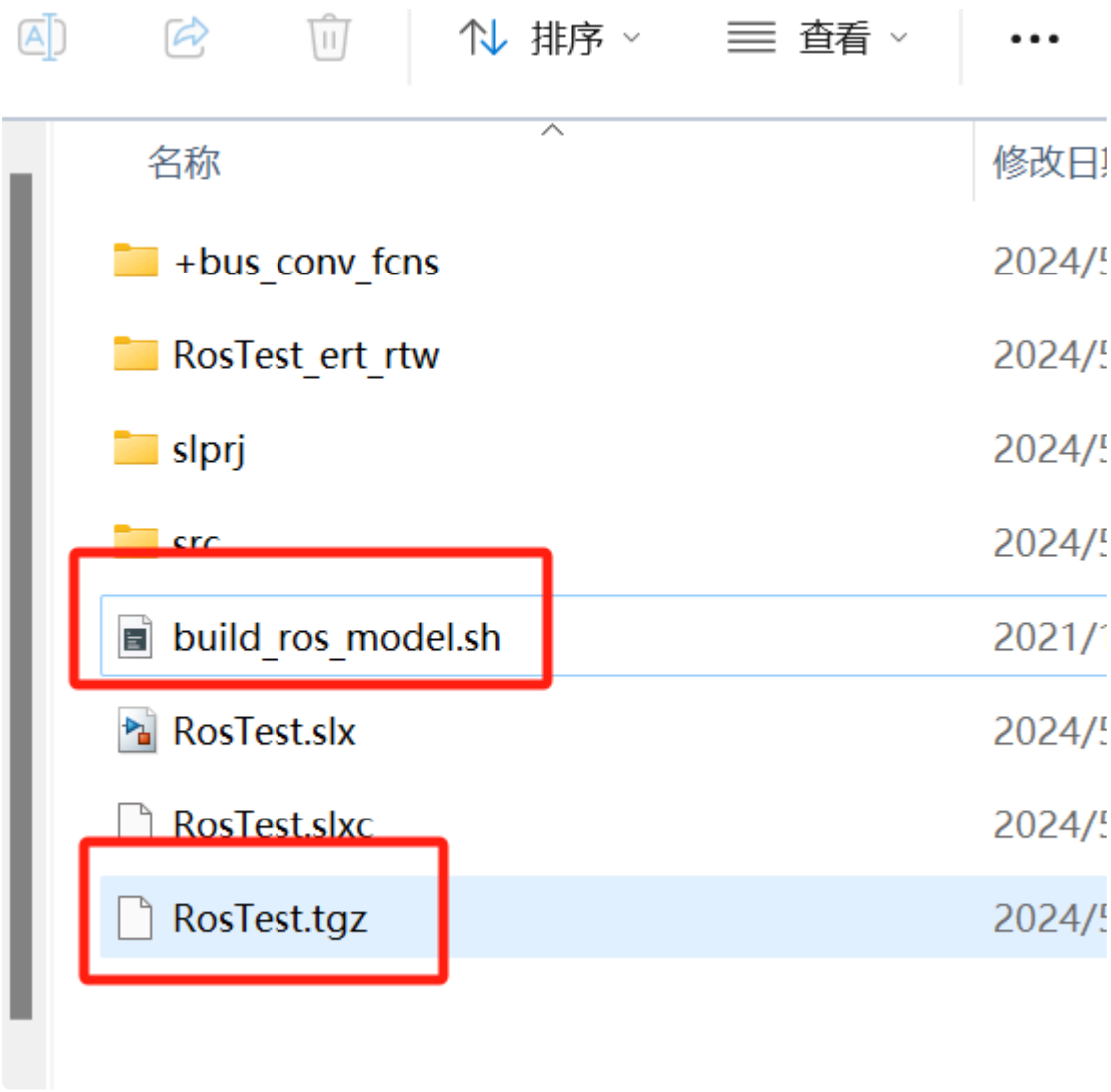


注意：自动代码生成时需要保证 rosmaster 正在运行。如果之前的 ROS 已关闭，可以打开 WinWSL 并输入 `roscore` 来启动 rosmaster。

若提示保留二进制，说明已完成代码生成。编译工作报错没关系，因为未安装 Windows 下的 Catkin 环境。

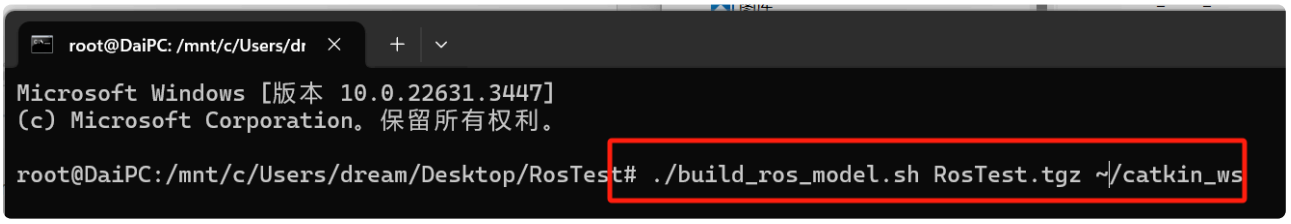


在目录下可得到如下关键文件。



打开 WinWSL，并进入当前目录。输入如下命令，在 Linux 下进行编译：

```
./build_ros_model.sh RosTest.tgz ~/catkin_ws
```



注意：对于其他例程，具体的编译命令提示可以运行 `build_ros_model.sh` 查看，或直接查看其内部注释。Linux 环境需要先在 `~/catkin_ws` 中用 ROS 进行初始化并预编译；工具链的 WSL 环境已经完成该配置。若在虚拟机或其他 Linux 系统部署，需要根据提示操作。

编译完成 100% 说明成功完成编译。

```
/libboost_thread.so.1.71.0 /usr/lib/x86_64-linux-gnu/libconsole_bridge.so.0
make[2]: Leaving directory '/root/catkin_ws/build'
[100%] Built target RosTest
make[1]: Leaving directory '/root/catkin_ws/build'
/usr/bin/cmake -E cmake_progress_start /root/catkin_ws/build/CMakeFiles 0
root@DaiPC:/mnt/c/Users/dream/Desktop/RosTest#
```

输入如下命令，进入编译的二进制文件夹，再运行编译的节点：

```
cd ~/catkin_ws/devel/lib/rostest/
./RosTest
```

```
root@DaiPC:/mnt/c/Users/dream/Desktop/RosTest# cd ~/catkin_ws/devel/lib/rostest/
root@DaiPC:~/catkin_ws/devel/lib/rostest# ls
RosTest
root@DaiPC:~/catkin_ws/devel/lib/rostest# ./RosTest
[ INFO] [1715076410.586341500]: ** Starting the model "RosTest" **
```

也可以先加载环境路径，再运行程序：

```
source ~/catkin_ws/devel/setup.bash
RosTest
```

编译生成的文件可以拷贝到其他 Ubuntu 系统上运行，也可以拷贝到其他系统重新编译后运行。之后，也可以按照 ROS 节点的启动方法正常启动节点。

5. 关键知识点

本实验以 RflySim 工具链内置的 `RflySim-20.04` WinWSL 环境为核心，串联 Windows 终端、Ubuntu 命令行、WslGUI 图形界面、mavros 通信、ROS1/ROS2 切换和 Simulink 代码生成等功能。整体思路是先保证 WinWSL 能进入正确目录并执行脚本，再验证图形界面和仿真数据传输，最后完成 ROS 与 Simulink 的联合开发链路。

关键知识点1：WinWSL环境的作用

WinWSL 是 RflySim 工具链内置的 Ubuntu 20.04 子系统，既可用于 PX4 固件编译和 SITL/HITL 仿真，也可用于 Python、OpenCV、ROS1/ROS2 等 Linux 开发任务。它的优势是能直接访问 Windows 下的 RflySim 例程目录，同时拥有 Linux 环境中的编译器、ROS 和 Python 运行能力。

关键知识点2：Windows路径与Linux路径映射

在 WinWSL 中访问 Windows 文件时，需要使用 `/mnt/<盘符小写>/...` 的路径格式。例如 Windows 路径 `C:\PX4PSP\RflySimAPIs` 对应 Linux 路径 `/mnt/c/PX4PSP/RflySimAPIs`。理解这一映射关系后，就能在 WinWSL 中进入任意 RflySim 例程目录并运行脚本。

关键知识点3：多WSL并存时的发行版选择

当电脑中存在多个 WSL 发行版时，直接输入 `wsl` 可能进入默认发行版，不一定是 RflySim 工具链环境。可通过 `wsl -l -v` 查看发行版列表，通过 `wsl --set-default RflySim-20.04` 设置默认环境，或通过 `wsl -d RflySim-20.04` 临时指定进入 RflySim 的 WinWSL 环境。

关键知识点4：WslGUI与命令行的关系

WslGUI 提供 Linux 图形桌面入口，方便用户在 Ubuntu 图形界面中打开终端、进入目录和运行程序；WinWSL 命令行则更适合快速执行脚本和调试命令。二者本质上都在访问同一个 RflySim-20.04 环境，只是交互方式不同。

关键知识点5：mavros消息命名规则

单机仿真时，mavros 消息通常使用 `/mavros/****` 话题前缀；多机仿真时，从 2 号飞机开始使用 `/mavros2/****`、`/mavros3/****` 等形式。后续 ROS 开发中，需要根据飞机 ID 订阅和发布正确的话题。

关键知识点6：ROS与Simulink互联

Simulink 可通过 ROS 模块订阅 mavros 话题，例如 `/mavros/local_position/pose`。在本例程中，Simulink 读取的是 mavros 的 ENU 坐标系数据，上方向为正；这与 PX4 常用的 NED 坐标系不同，开发控制算法时需要特别注意坐标系转换。

6.参考资料

1. RflySim官方文档：https://rflysim.com/doc/zh/

2. Microsoft WSL官方文档： https://learn.microsoft.com/windows/wsl/
3. ROS官方文档： https://www.ros.org/
4. MAVROS文档： https://wiki.ros.org/mavros
5. PX4 ROS/MAVROS说明：
   https://docs.px4.io/main/en/ros/mavros_installation.html

7.常见问题

Q1：右键终端输入 `wsl` 后没有进入 RflySim-20.04 怎么办?

A1：先在 cmd 中输入 `wsl -l -v` 查看当前默认 WSL 系统。如果默认系统不是 `RflySim-20.04`，可执行 `wsl --set-default RflySim-20.04`，或在需要使用时执行 `wsl -d RflySim-20.04`。

Q2：WinWSL 中如何进入 Windows 下的例程目录?

A2：将 Windows 路径转换为 `/mnt/<盘符小写>/...` 格式。例如 `C:\PX4PSP\RflySimAPIs\1.RflySimIntro\2.AdvExps\e7.WslGUI` 应转换为 `/mnt/c/PX4PSP/RflySimAPIs/1.RflySimIntro/2.AdvExps/e7.WslGUI`。

Q3：WslGUI 图形界面无法正常显示怎么办?

A3：先确认 `WinWSL` 能正常启动，再运行 `WslGUI.bat`。如果仍无法显示，检查工具链中的 VcXsrv/WslGUI 启动脚本是否完整，并优先在 `GuiTest` 目录运行最小化测试。

Q4：ROS2 下 `ros2 topic list` 看不到 mavros 话题怎么办?

A4：可尝试在设备管理器中禁用除当前使用网络适配器之外的其他适配器，例如 VMware 虚拟机适配器或其他有线/无线网卡适配器，然后重启电脑并重新运行实验。

Q5：Simulink 读取的位置方向和预期不一致怎么办?

A5：注意 mavros 使用 ENU 坐标系，上方向为正；PX4 常见说明中经常使用 NED 坐标系。进行控制算法开发时，需要确认坐标系定义并进行必要转换。

---

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