向外回传uORB消息mavlink_log实验

1. 实验目的

在飞控中，我们时常需要向外发布一些文字消息，来反映系统当前的运行状态，这个功能可以通过发送“mavlink_log”的uORB消息来实现。

2. 实验要求

- 软件要求：Windows 10及以上版本；RflySim工具链[1]，MATLAB2022B以上版本，平台安装时的编译命令为：px4_fmu-v6x_default，推荐PX4固件版本为：1.14.3。
- 硬件要求：笔记本/台式电脑1台[2]，Pixhawk 6X/Pixhawk 6X；遥控器和遥控器接收机；数据线和杜邦线等。

3.实验地址

例程目录：[安装目录]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\8.Mavlink-Msg-Echo

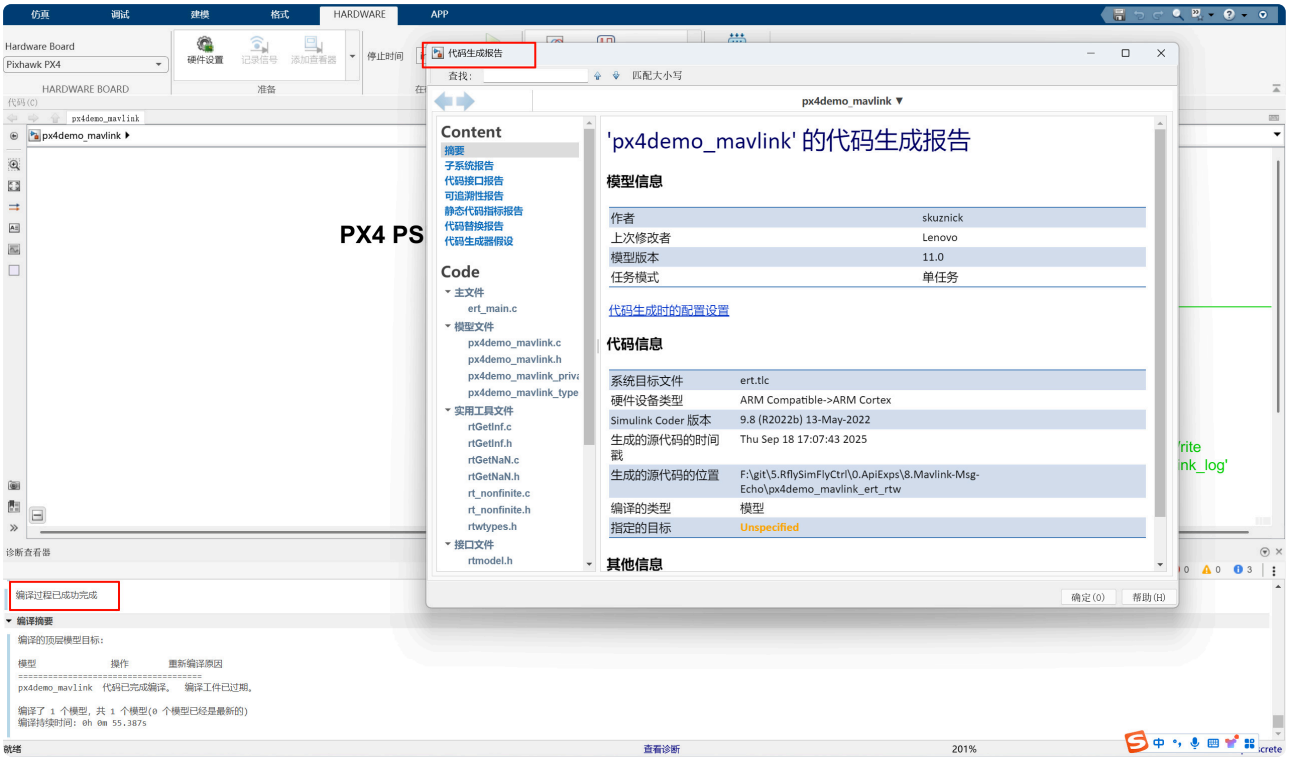
- px4demo_mavlink_rc.slx：回传遥控器状态量提示消息模型文件

4. 实验内容或步骤

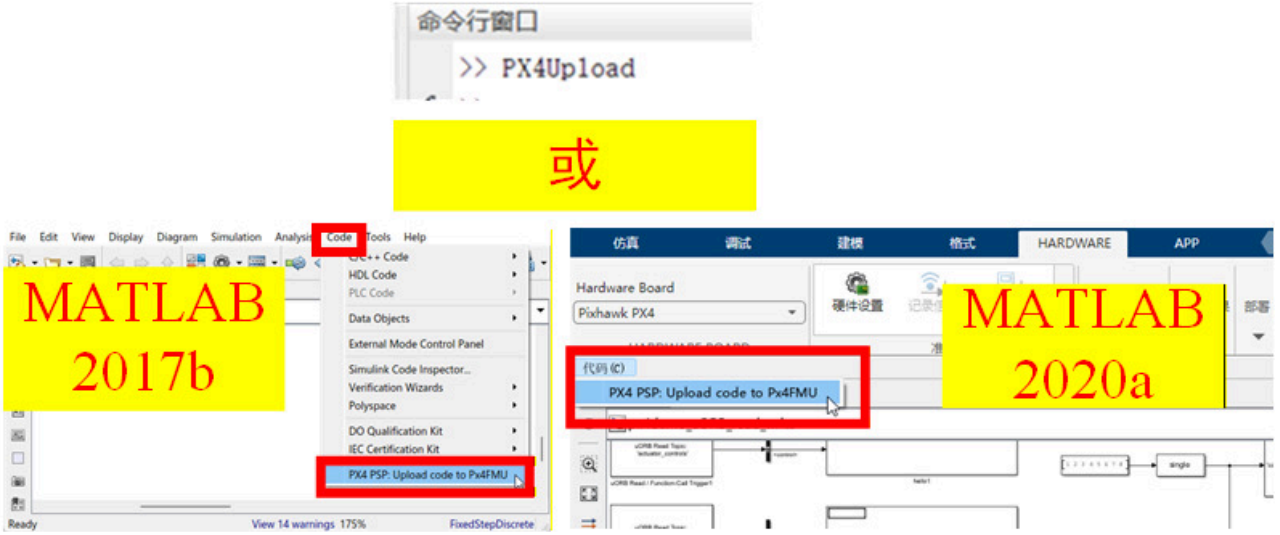
5.1 步骤1：回传提示消息实验

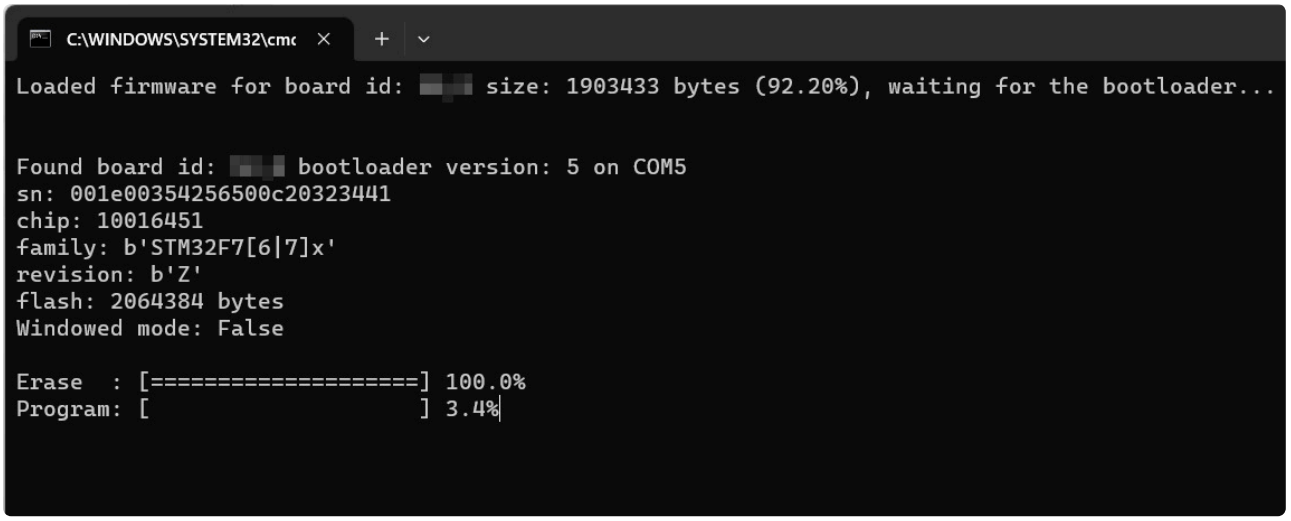
打开MATLAB软件，在MATLAB中打开px4demo_mavlink_rc.slx文件，在Simulink中，点击编译命令。

在Simulink的下方点击查看诊断，即可弹出诊断对话框，可查看编译过程。在诊断框中弹出编译过程已成功完成，即可表示编译成功，也会弹出代码生成报告。

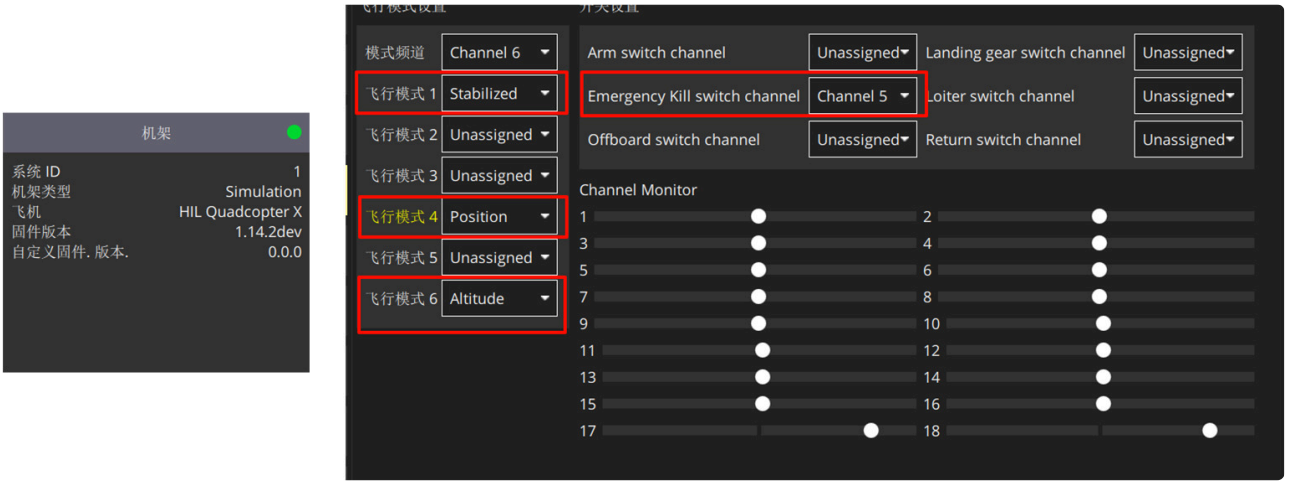


用USB数据线链接飞控与电脑。在MATLAB命令行窗口输入：PX4Upload并运行或点击PX4PSP：Upload code to Px4FMU，弹出CMD对话框，显示正在上传固件至飞控中，等待上传成功。

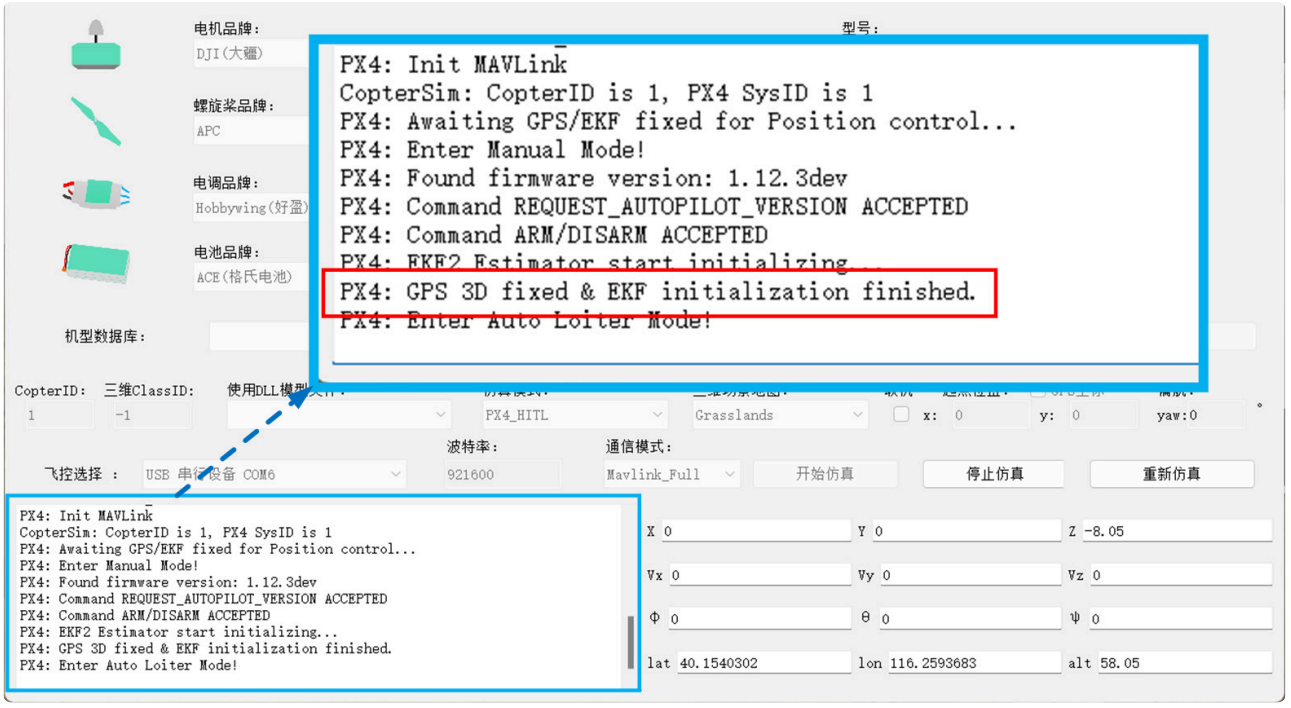




打开QGroundControl软件。确认无人机机架及遥控器通道设置如下：



上传成功后，双击打开"*\桌面\RflyTools\HITLRun.lnk"或"*\PX4PSP\RflySimAPIs\HITLRun.bat"文件，在弹出的CMD对话框中输入插入的飞控Com端口号，即可自动启动RflySim3D、CopterSim、QGroundControl软件，等待CopterSim的状态框中显示：PX4:GPS 3D fixed & EKF initialization finished。



在CopterSim的状态框中将实时输出数值变化

5. 关键知识点

关键知识点1：Mavlink_text_msg—Mavlink格式化消息文本

打开[[安装目录]]
(file:///C:/Users/admin/Desktop/55555/5.RflySimFlyCtrl/0.ApiExps/5.Log-Write-Read)

[5.RflySimFlyCtrl\0.ApiExps\8.Mavlink-Msg-Echo](file:///C:/Users/admin/Desktop/55555/5.RflySimFlyCtrl/0.ApiExps/8.Mavlink-Msg-Echo)中的px4demo_mavlink.slx文件，双击mavlink_text_msg该模块查阅函数代码。该函数主要用于生成格式化的消息文本，其中包含了计数器值、radio 和 chval 的值。在每次调用函数时，计数器会自动递增，并在达到 'Z' 后循环回到 'A'。

代码及解析如下：

一个名为 fcn 的函数，它接受两个输入参数 radio 和 chval。

```
function msgtxt = fcn(radio, chval)
% coder.cinclude('drivers/drv_hrt.h');
```

声明了一个持久变量 count，用于跟踪消息中的字符。

```
persistent count
```

检查计数器是否为空，如果是，则将其初始化为 'A' 对应的 ASCII 值。

```
if isempty(count)
count = uint8('A');
end
```

创建一个大小为 50 的 uint8 类型数组 msgtxt，用于存储消息文本。

```
msgtxt = zeros(1,50,'uint8');
```

使用 sprintf 函数将格式化后的消息写入到 msgtxt 数组中，包括计数器值、radio 和 chval 的值。

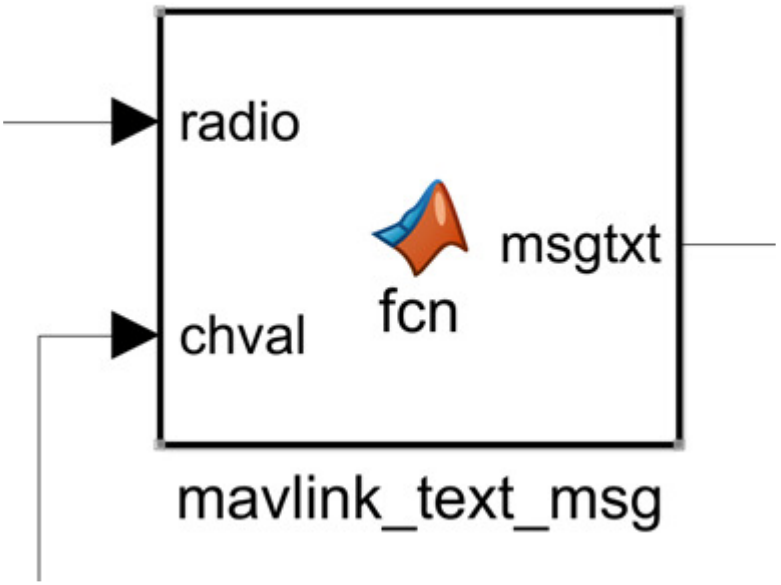
```
coder.ceval('sprintf', coder.wref(msgtxt), 'Simulink [%c] Radio %d Value %d',
count, int32(radio), int32(chval));
```

更新计数器的值，使其递增。

```
count = count + uint8(1);
```

检查计数器是否达到了 'Z' 对应的 ASCII 值，如果是，则将其重新设置为 'A' 对应的 ASCII 值，以实现循环计数。

```
if count == uint8('Z')
          count = uint8('A');
end
```



mavlink_text_msg

```
function msgtxt  = fcn(radio, chval)

% coder.cinclude('drivers/drv_hrt.h');

persistent count
if isempty(count)
    count = uint8('A');
end

msgtxt = zeros(1,50,'uint8');
coder.ceval('sprintf', coder.wref(msgtxt), 'Simulink [%c] Radio %d Value %d', count, int32(radio), int32(chval));

count = count + uint8(1);
if count == uint8('Z')
    count = uint8('A');
end
```

关键知识点2：Hrt_timestamp—获取时间戳

打开[[安装目录]](file:///C:/Users/admin/Desktop/55555/5.RflySimFlyCtrl/0.ApiExps/5.Log-Write-Read)[5.RflySimFlyCtrl\0.ApiExps\8.Mavlink-Msg-Echo](file:///C:/Users/admin/Desktop/55555/5.RflySimFlyCtrl/0.ApiExps/8.Mavlink-Msg-Echo)中的px4demo_mavlink.slx文件，双击hrt_timestamp该模块查阅函数代码。该函数主要用于返回当前的时间戳，如果在嵌入式硬件上运行，则获取绝对时间；否则，使用计数器来模拟时间戳的增加。

代码及解析如下：

这是一个名为 fcn 的函数，它没有输入参数。

```
function ts = fcn
% coder.cinclude('drivers/drv_hrt.h');
```

声明了一个持久变量 count，用于跟踪时间戳。

```
persistent count
```

检查计数器是否为空，如果是，则将其初始化为 0。

```
if isempty(count)
          count = 0;
end
```

初始化变量 hrt。

```
hrt = 0;
```

条件语句用于根据目标是否为嵌入式硬件来获取时间戳。在硬件目标情况下，使用 coder.ceval 调用获取绝对时间；否则，使用 count 作为时间戳。

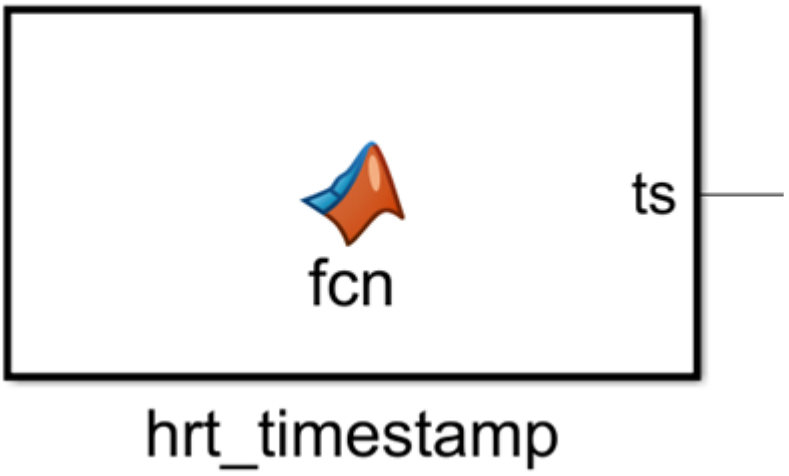
```
if coder.target('Rtw')
          hrt = coder.ceval('(double)hrt_absolute_time');
else
          hrt = count;
end
```

将时间戳赋值给输出变量 ts。

```
ts = hrt;
```

在每次函数调用时，计数器会递增。

```
count = count + 1;
```



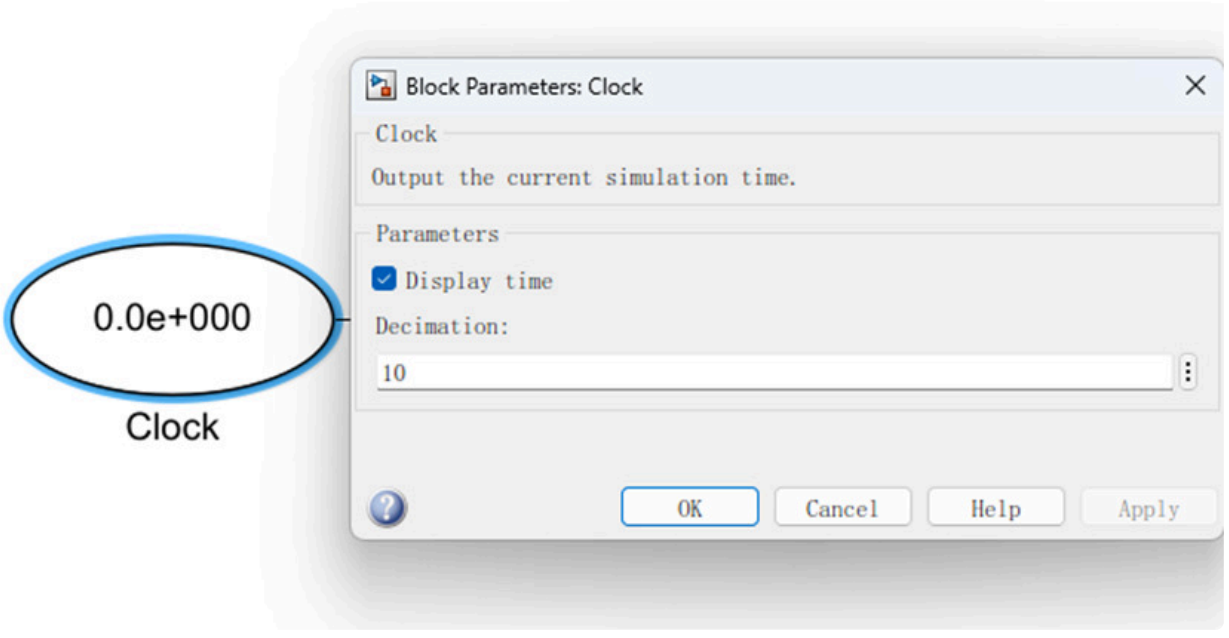
关键知识点3：1.1. Clock—时钟

，Clock（时钟）模块是一个常用的基础模块，用于生成周期性的信号或指定的时间戳。它可以用于各种模拟和仿真任务，例如：

时钟信号生成：Clock模块可以生成周期性的信号，如正弦波、方波或任何其他周期性信号。用户可以指定时钟的频率和相位。

时间戳生成：Clock模块也可以用于生成时间戳，这在某些实时系统仿真或数据记录任务中非常有用。

同步：在多模块系统中，Clock模块可以用于同步不同模块的操作，确保它们按照预定的时间间隔执行。



关键知识点4：Constant—常数

Constant（常数）模块用于生成一个固定值的信号。这个固定值可以是任何你想要的数值，可以是标量、向量或矩阵形式，用于提供模型中的常量输入或参数。

Constant 模块的主要特点和用法包括：

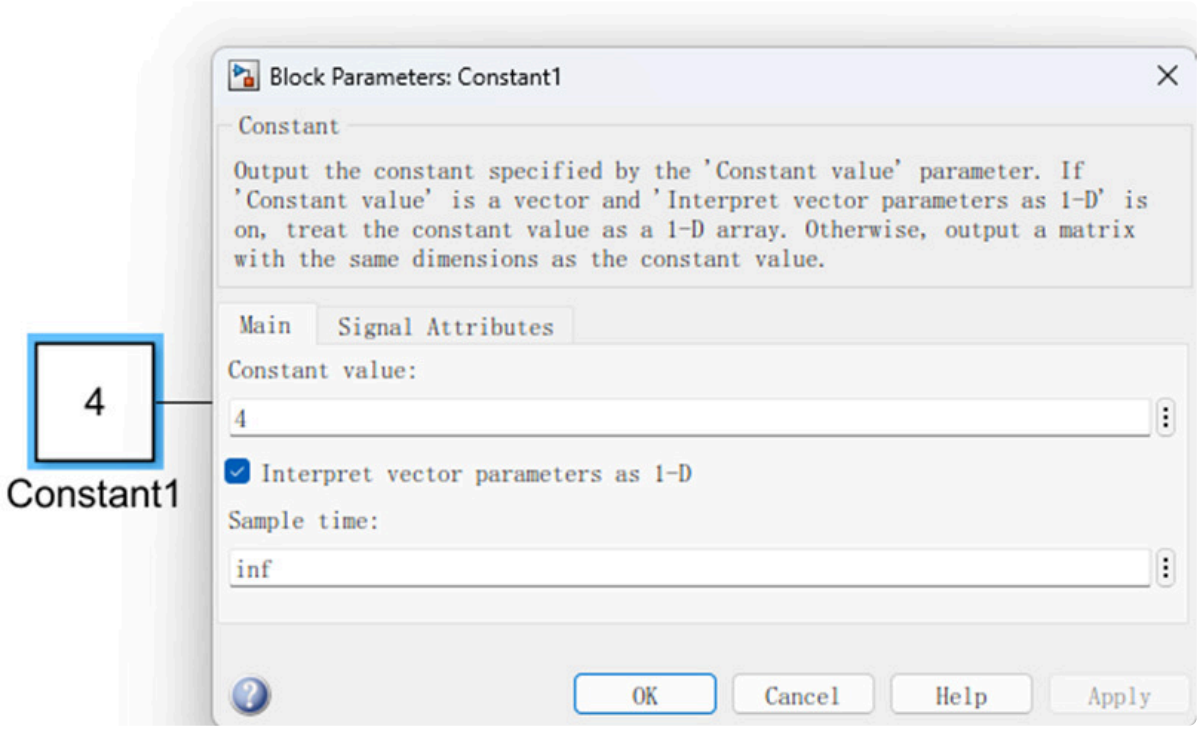
固定数值：该模块生成的信号具有固定的数值，不会随时间变化。可以在模块的参数设置中指定这个固定的数值。

参数设置：可以在 Constant 模块的参数设置中指定生成的固定数值。这使得能够轻松地在模型中引入常量值，用于初始化参数、提供固定输入信号等。

输出信号：Constant 模块生成的固定值信号可以作为模型中的输入信号，连接到其他模块的输入端口，用于模拟系统中的固定条件或参数。例如，可以将该信号连接到控制系统中的某些参数，以评估系统在固定条件下的行为。

多维数组支持：除了标量值外，Constant 模块还支持生成向量或矩阵形式的固定值信号。这使得能够在模型中引入多维数组的常量值，用于初始化参数或提供多维输入信号。

通过 Constant 模块，可以方便地在 Simulink 中生成固定的数值信号，用于提供常量输入或参数，初始化系统参数，或者用作模型中的固定条件。



关键知识点5：Data Type Conversion—数据类型转换

Data Type Conversion（数据类型转换）模块用于将输入信号的数据类型转换为所需的输出数据类型。这个模块在处理不同数据类型之间的转换时非常有用，例如将整数转换为浮点

数，或者将固定点数转换为双精度浮点数等。

主要特点和用法包括：

数据类型转换：Data Type Conversion 模块可以将输入信号的数据类型转换为指定的输出数据类型。常见的数据类型包括整数、浮点数、固定点数等。

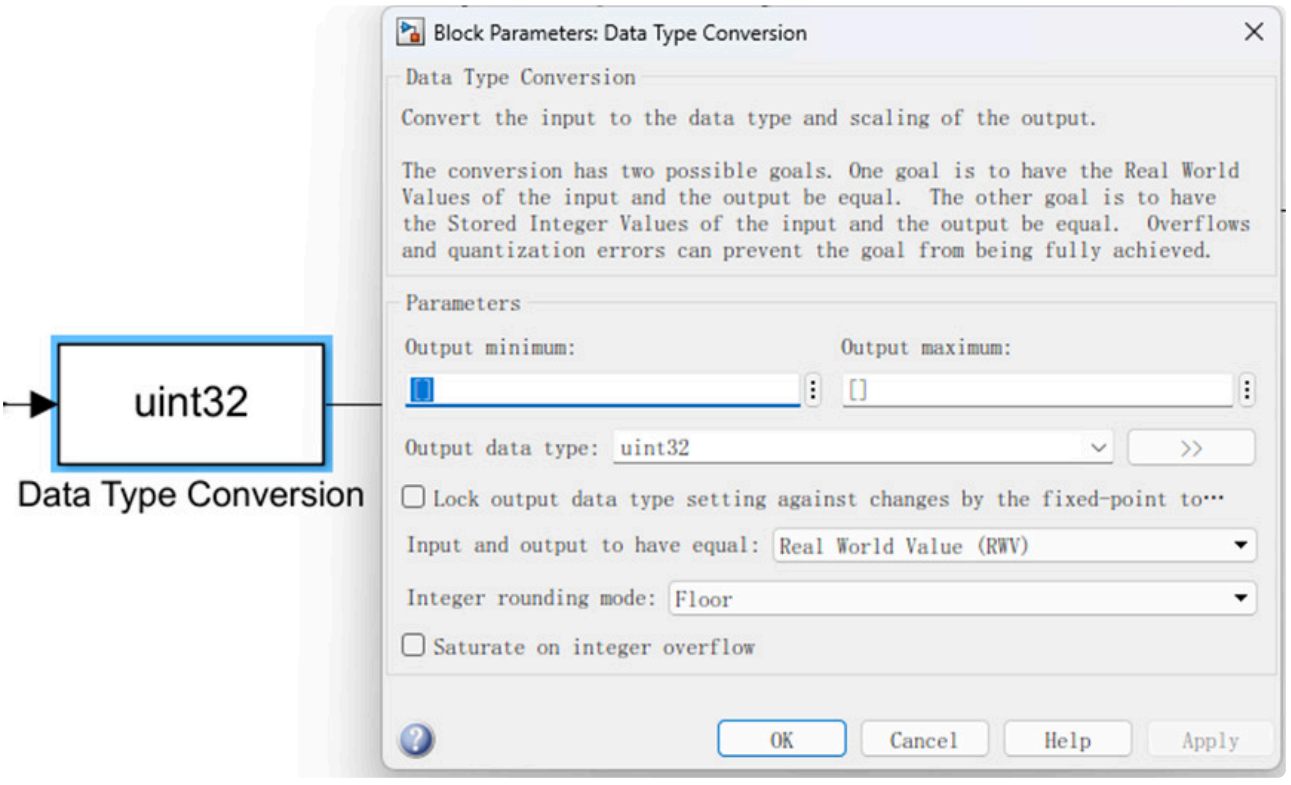
精度控制：通过 Data Type Conversion 模块，可以控制信号的精度，例如从高精度转换到低精度，或者从低精度转换到高精度。这对于在系统中平衡性能和计算成本非常有用。

饱和和截断：在进行数据类型转换时，可以选择是否进行饱和或截断操作。饱和操作会将超出目标数据类型范围的值限制在目标范围内，而截断操作则会直接舍弃超出目标数据类型范围的部分。

舍入模式：在进行浮点数或固定点数转换时，可以选择不同的舍入模式，包括向上舍入、向下舍入、向零舍入等。这有助于控制在转换过程中的舍入误差。

输出端口数目：Data Type Conversion 模块可以有一个或多个输出端口，取决于输入信号的数量和转换类型。你可以根据需要选择合适的输出端口。

通过 Data Type Conversion 模块，可以灵活地控制信号的数据类型和精度，以满足系统建模和仿真的需求。这在处理不同数据类型的信号、接口以及在不同精度下进行计算时非常有用。



关键知识点6： Gain—增益

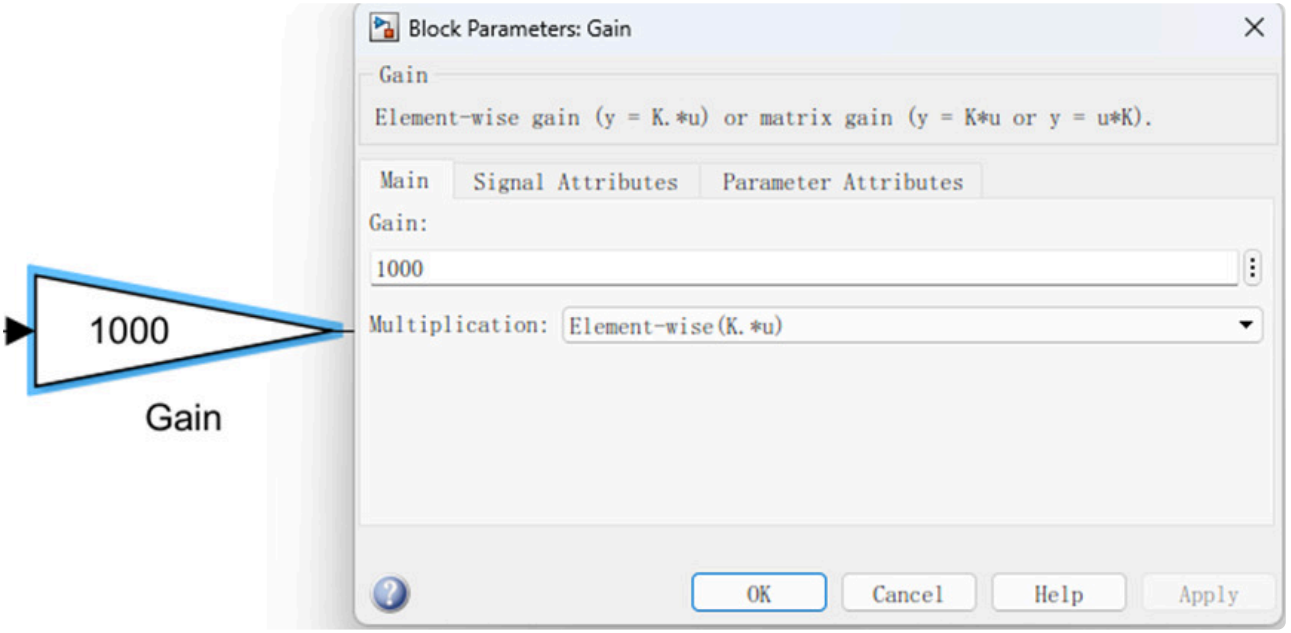
Gain（增益）模块是一种用于放大或缩小信号的基本模块。它允许通过乘以一个常数来调整输入信号的幅度。Gain模块在模拟和数字信号处理中非常常见，用于调整系统的增益或放大输入信号以满足特定要求。

功能特点：

增益调节：Gain模块允许以任意比例调整输入信号的幅度。可以通过设置模块参数中的增益值来实现所需的放大或缩小效果。

常数乘法：Gain模块实际上执行了一个简单的乘法运算，将输入信号乘以一个常数。这种乘法运算是模拟和数字信号处理中的基本操作之一。

实时调节：可以在Simulink模型中动态调整Gain模块的增益值，以便在仿真过程中实时改变系统的放大倍数。



关键知识点7：Simulation Pace—仿真节奏模块

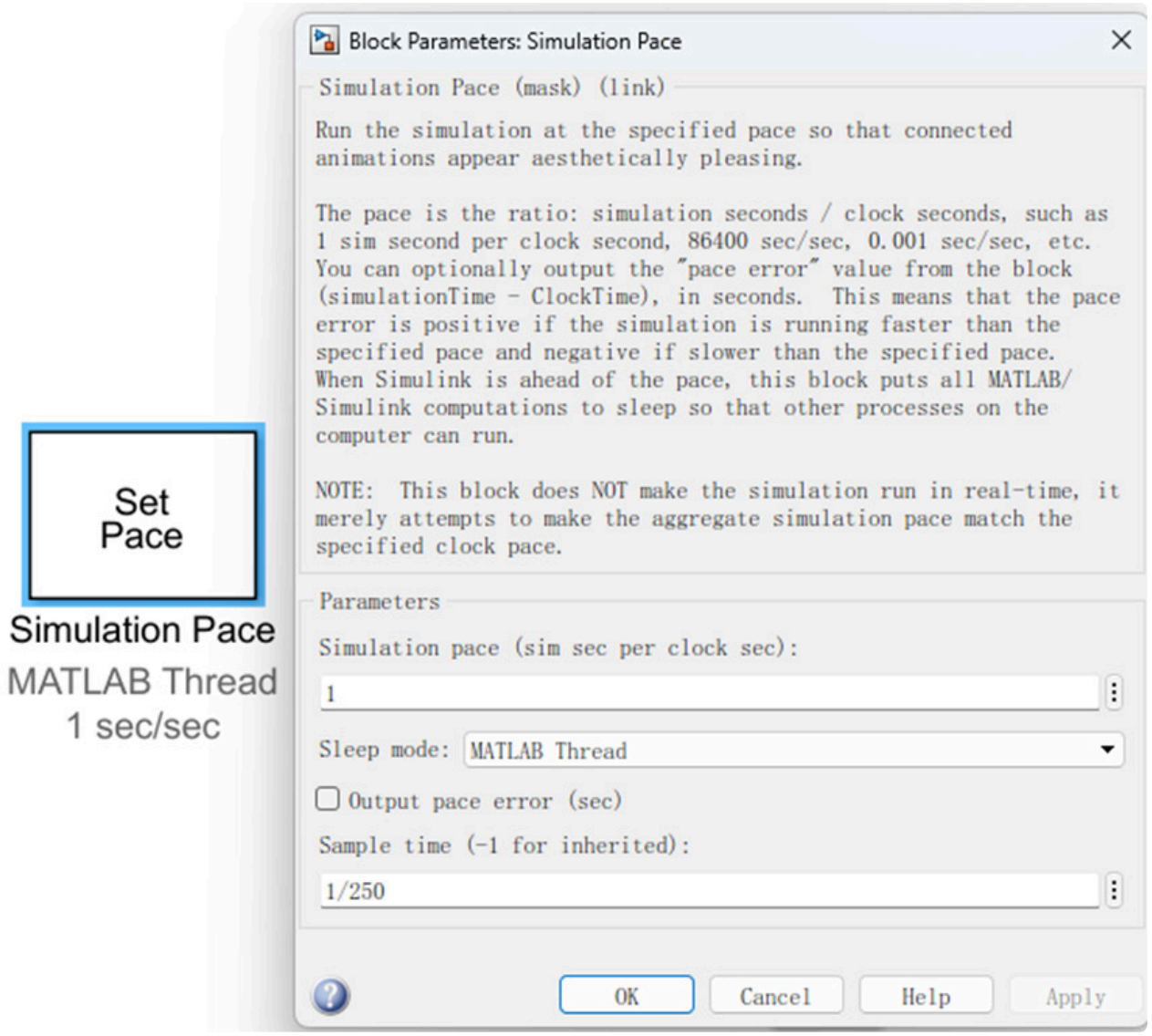
Simulation Pace（仿真节奏）模块是一种用于控制仿真速率的工具。它允许以特定的速率执行模型的仿真，从而可以控制仿真的时间步长和速度。这在需要控制仿真速率以匹配实际系统行为或加速仿真过程的情况下非常有用。

功能特点：

控制仿真速率：Simulation Pace模块允许指定仿真的速率，即每个仿真步骤的持续时间。这样可以确保仿真以特定的速率运行，与实际系统的行为相匹配。

调整仿真时间步长：通过设置Simulation Pace模块的参数，可以调整仿真的时间步长，使其更适合仿真需求。这对于需要更精细的仿真控制或更快的仿真速度都非常有用。

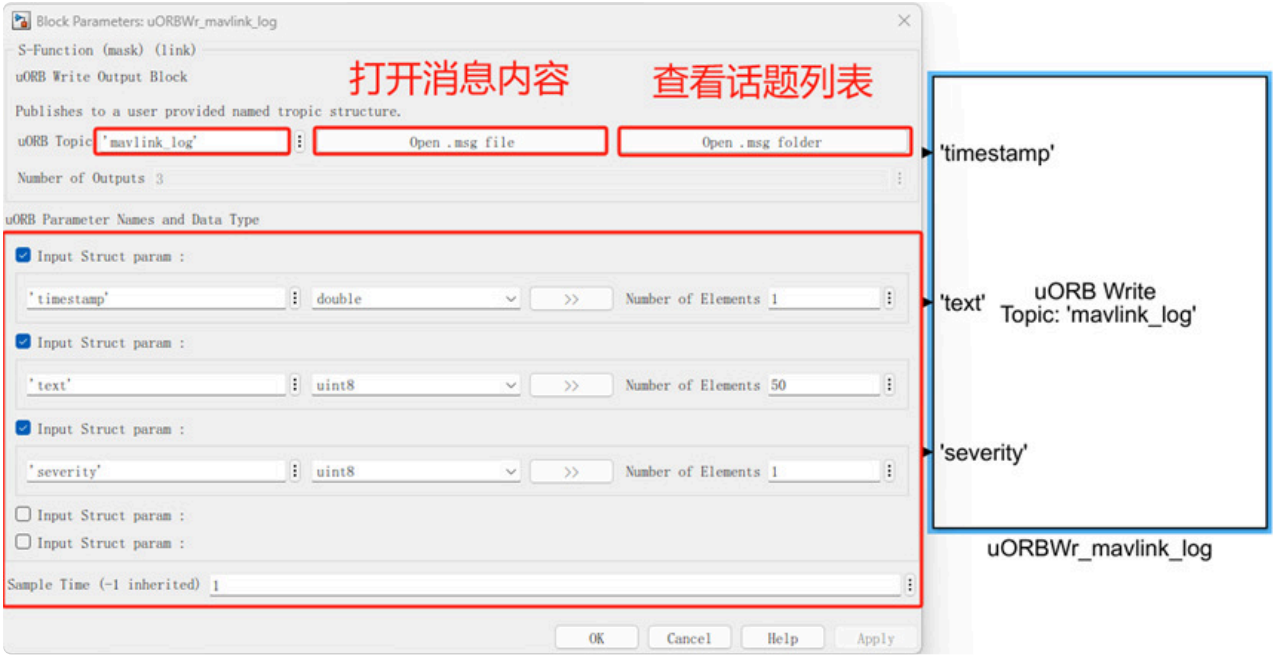
控制仿真速度：仿真节奏模块还允许控制仿真的速度，以便更快地进行模型验证或测试。通过加快仿真速度，可以更快地获取模型的输出结果。

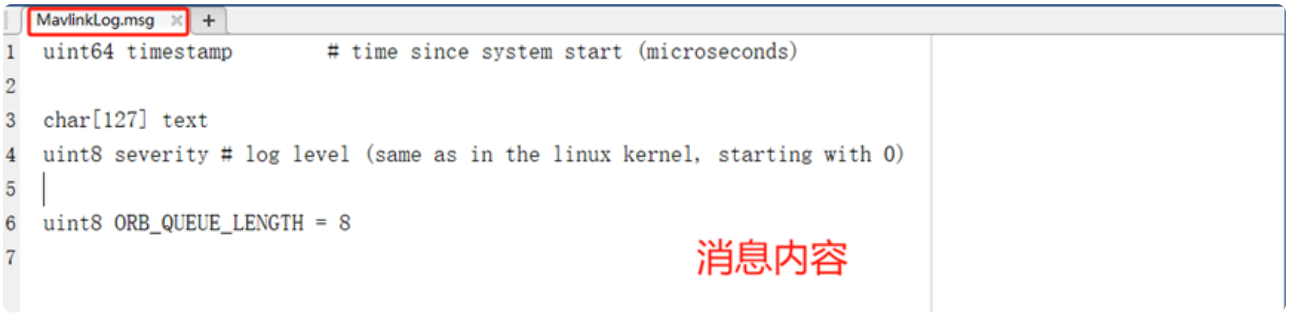


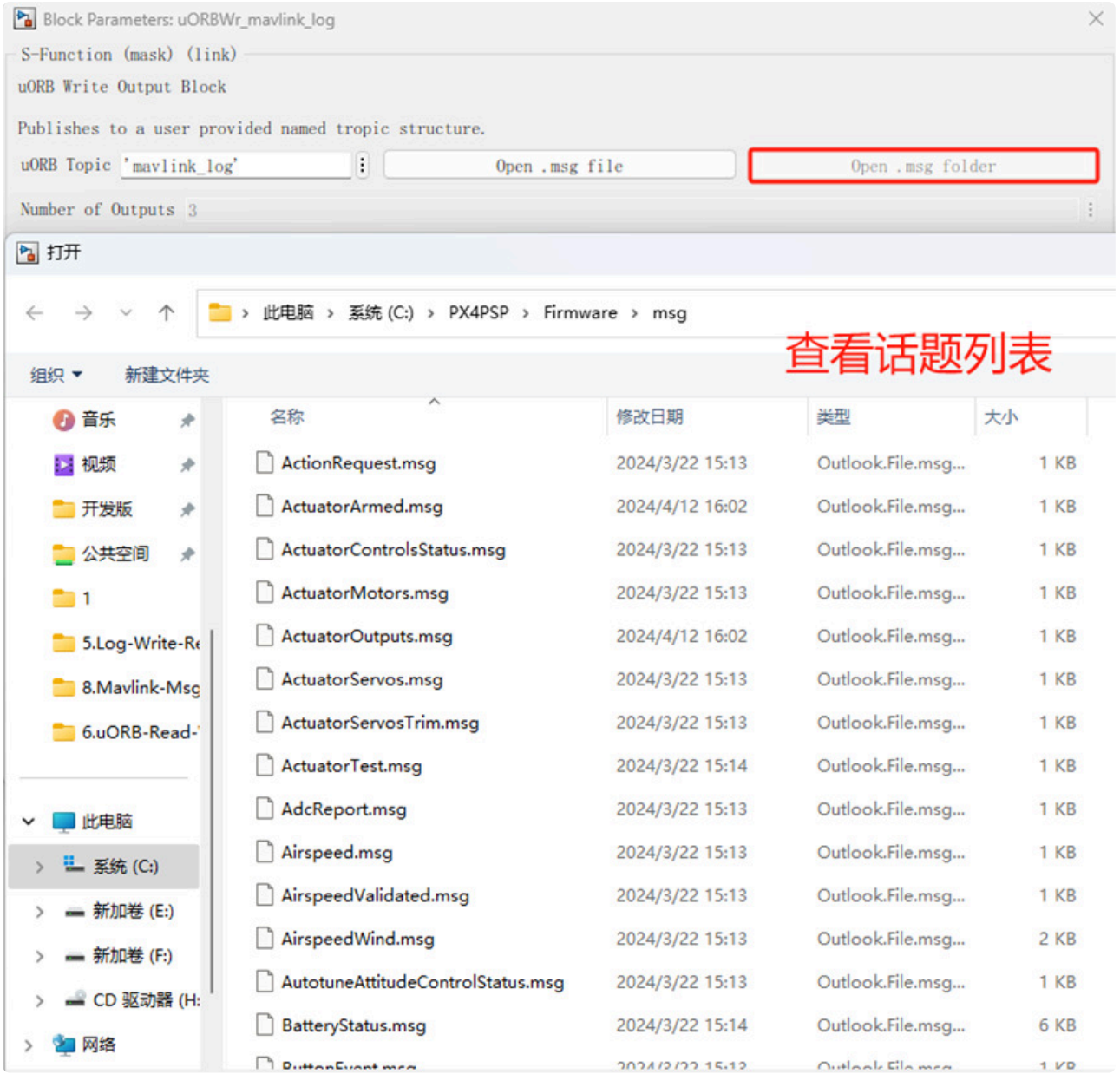
关键知识点8：uORB Write—uORB消息数据发布接口

该接口允许用户向uORB话题发布指定的值或结构体，通过这个模块可以向某个uORB话题发布对应的消息，话题必须经过正确的定义，一些已定义的话题放在目录C:\PX4PSP\Firmware\msg下，在生成的代码中会自动包含话题的定义文件。

如下图所示，可以输入话题名，点击按钮“Open.msgfile”打开对应的消息内容，点击按钮“Open.msgfolder”打开话题列表，设置输入端口名及数据类型以对应话题消息。


Block Parameters: uORBWr_mavlink_log
S-Function (mask) (link)
uORB Write Output Block
打开消息内容
查看话题列表
Publishes to a user provided named tropic structure.
uORB Topic 'mavlink_log'
Open .msg file
Open .msg folder
Number of Outputs 3
uORB Parameter Names and Data Type
Input Struct param :
'timestamp'
double
Number of Elements 1
Input Struct param :
'text'
uint8
Number of Elements 50
Input Struct param :
'severity'
uint8
Number of Elements 1
Input Struct param :
Input Struct param :
Sample Time (-1 inherited) 1
OK
Cancel
Help
Apply
'timestamp'
'text'
uORB Write
Topic: 'mavlink_log'
'severity'
uORBWr_mavlink_log


MavlinkLog.msg
1 uint64 timestamp # time since system start (microseconds)
2
3 char[127] text
4 uint8 severity # log level (same as in the linux kernel, starting with 0)
5
6 uint8 ORB_QUEUE_LENGTH = 8
7
消息内容



6.参考资料

1. RflySim官方文档
2. PX4 Developer Guide - https://docs.px4.io/main/en/
3. Simulink Documentation - https://www.mathworks.com/help/simulink/
4. MAVLink Protocol Documentation - https://mavlink.io/en/
5. uORB Messaging System - https://docs.px4.io/main/en/middleware/uorb.html

7.常见问题

Q1：编译过程中出现错误提示"无法找到PX4固件路径"?

A1：这个问题通常是因为PX4固件路径未正确配置导致的。请检查C:\PX4PSP\Firmware目录是否存在，如果不存在，请重新安装RflySim工具链或手动设置正确的固件路径。

Q2：上传固件到飞控时提示"连接失败"或"端口被占用"?

A2：这种情况通常是由于飞控未正确连接或端口被其他程序占用引起的。请检查USB数据线是否正常连接，尝试更换USB接口，关闭可能占用串口的程序如QGroundControl后再尝试上传。

Q3：在QGroundControl中没有看到从飞控发送的日志消息?

A3：如果日志消息未能显示，首先确认模型是否正确编译和上传，其次检查MAVLink消息的发送频率和格式是否符合规范。可以在QGroundControl的"分析工具"->"MAVLink Inspector"中查看是否有相应的mavlink_log消息接收。

---

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