

# | 基于binary\_logger模块的日志写入和读取验证实验

## | 1. 实验目的

使用二进制日志记录模块：binary\_logger，完成飞行数据写入与读取，log数据记录，以RflySim平台设定了20s的四维随机数据，数据存储位置飞控板内的片上外设存储卡内（路径为/fs/microsd/log/pixhawk），熟悉PX4飞控的底层运行逻辑。

## | 2. 实验要求

此编写实验目的

- 软件要求：Windows 10及以上版本；RflySim工具链<sup>[1]</sup>，MATLAB2022B以上版本，平台安装时的编译命令为：px4\_fm-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<sup>[2]</sup>，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\5.Log-Write-Read](#)

- <pixhawk\_A.bin>: \*.bin格式日志文
- <log\_0\_2024-9-23-19-34-50.ulg>:\*.ulg格式日志文件
- <px4demo\_log.slx>:Simulink飞行日志写入\*.bin格式日志模型
- px4demo\_log\_qgc.slx: Simulink飞行日志写入\*.ulg格式日志模型。

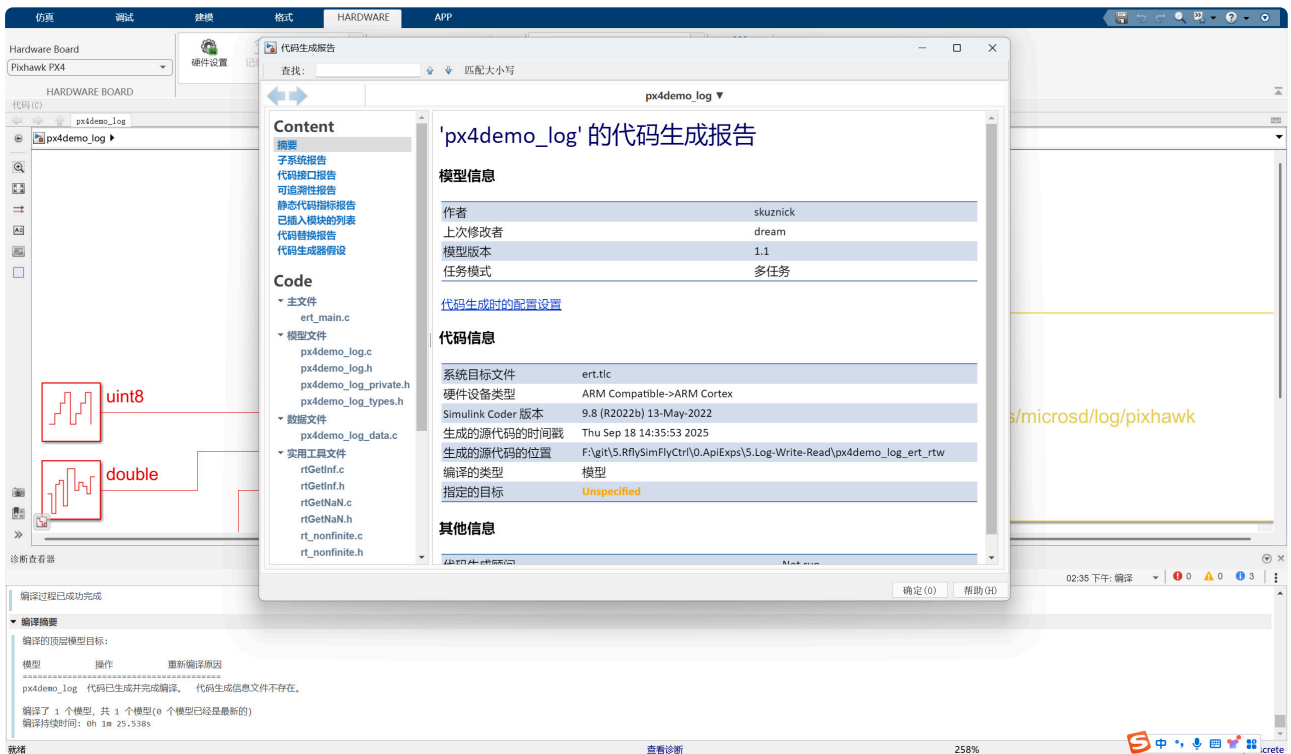
## 4. 实验内容或步骤

### 5.1 步骤1:\*.bin格式日志数据记录与读取实验

打开MATLAB软件，在MATLAB中打开px4demo\_log.slx文件，在Simulink中，点击编译命令。



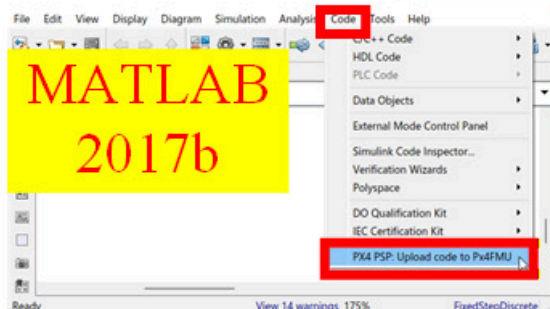
在Simulink的下方点击查看诊断，即可弹出诊断对话框，可查看编译过程。在诊断框中弹出编译过程已成功完成，即可表示编译成功，也会弹出代码生成报告。



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PX4 PSP: 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。

```
命令行窗口
>> PX4Upload
```

或

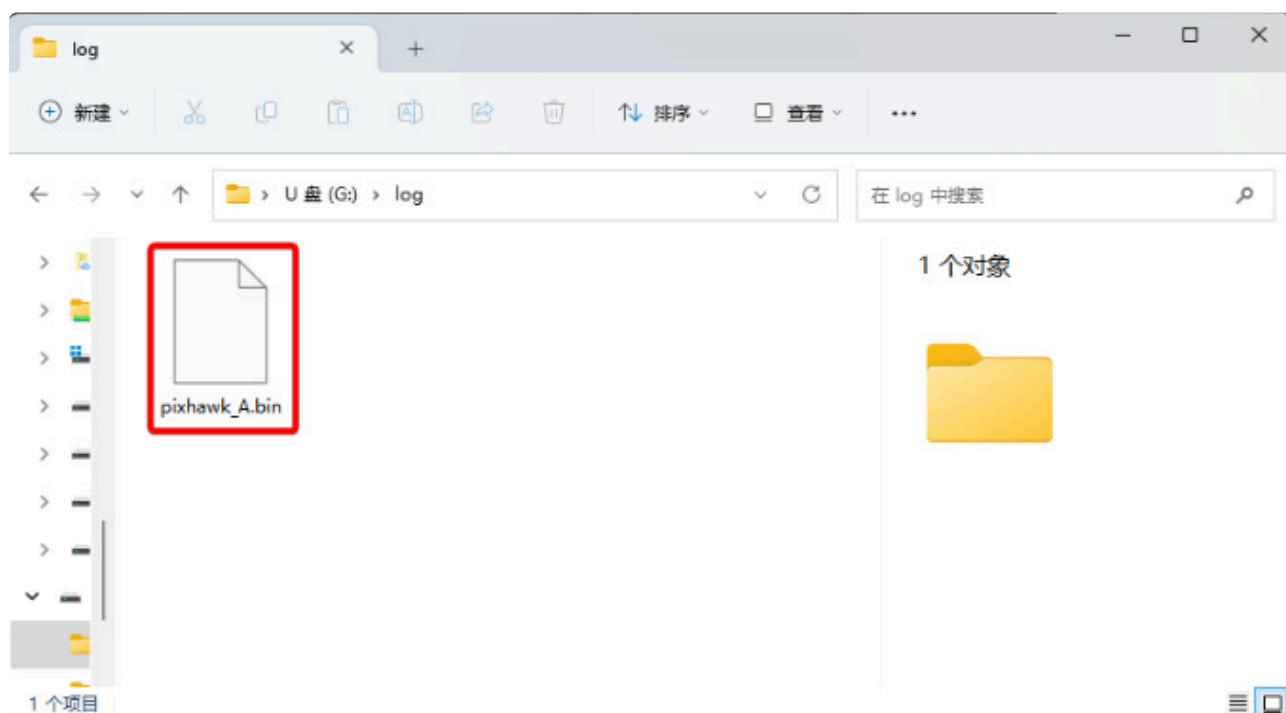


```
C:\WINDOWS\SYSTEM32\cmd
Loaded firmware for board id: [redacted] size: 1903433 bytes (92.20%), waiting for the bootloader...

Found board id: [redacted] bootloader version: 5 on COM5
sn: 001e00354256500c20323441
chip: 10016451
family: b'STM32F7[6|7]x'
revision: b'Z'
flash: 2064384 bytes
Windowed mode: False

Erase : [=====] 100.0%
Program: [ ] 3.4%
```

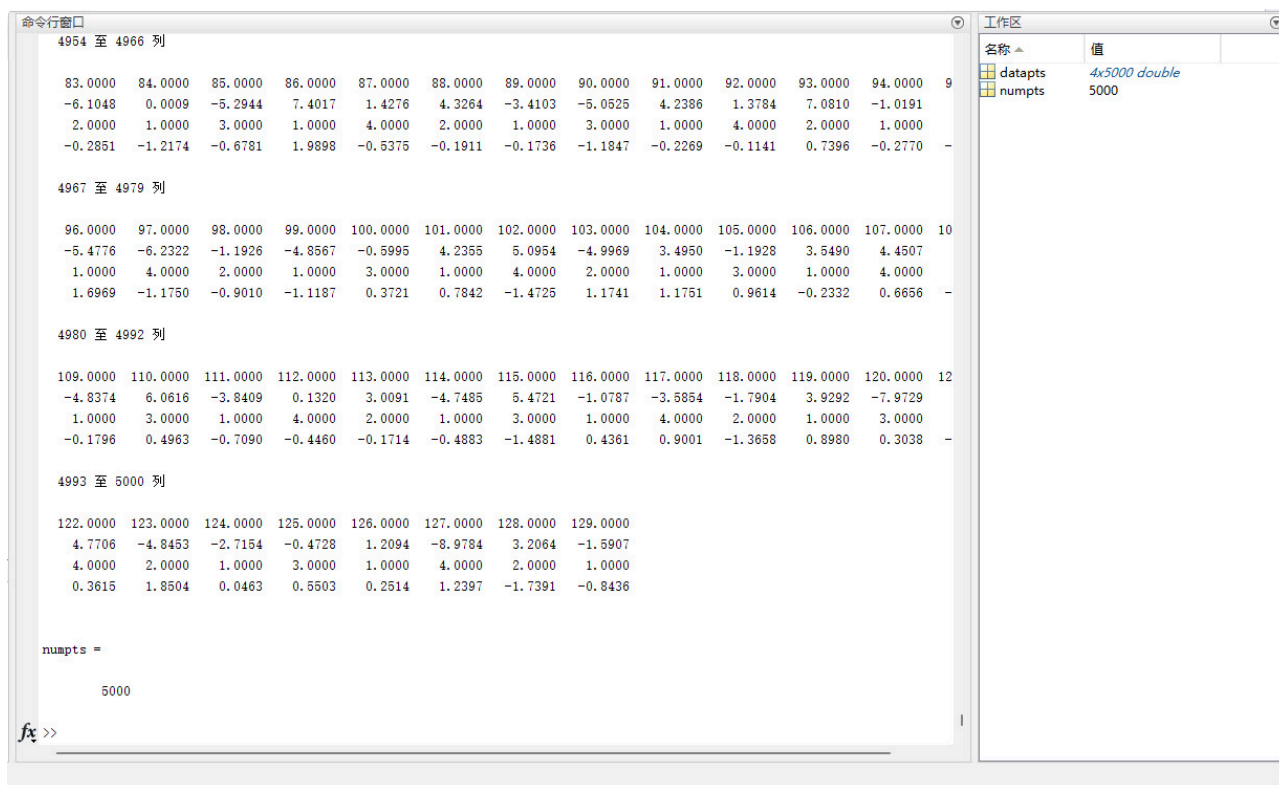
上传成功后，等待20s之后取出飞控中的SD卡，插入电脑后，在SD卡文件中找到log文件夹，复制pixhawk\_A.bin文件，到实验的文件夹下，可修改文件名避免冲突！



在MATLAB命令行中输入如下程序：

```
1 PX4ReadLog('pixhawk_A.bin')
2 或
3 PX4ReadLog 'pixhawk_A.bin'
```

运行后，datapts为记录的4\*5000矩阵数据，同时也将全部打印命令行中，numpts为采集到的数据点数量。

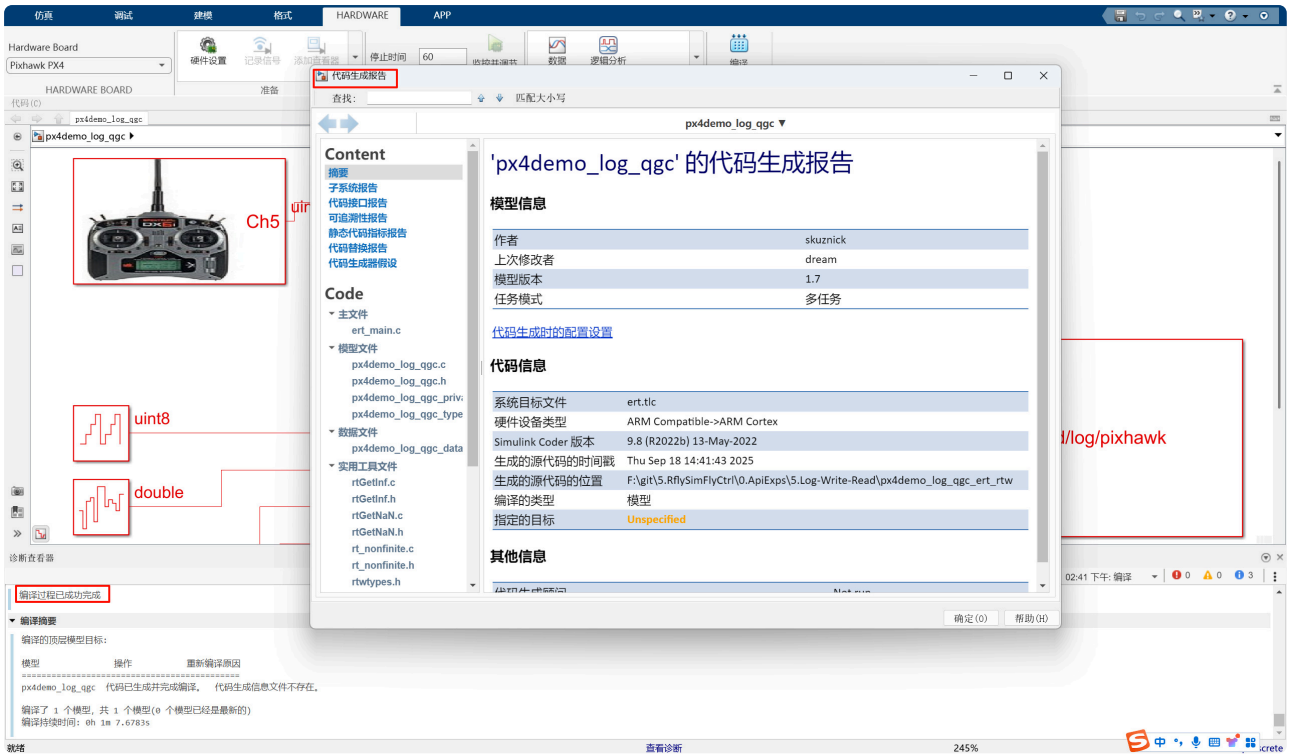


## 5.2 步骤2：\*.ulg格式日志数据记录与读取实验

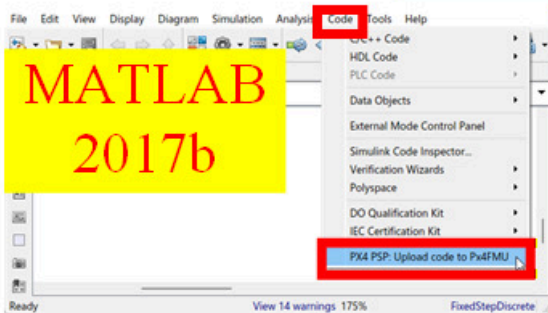
打开MATLAB软件，在MATLAB中打开<px4demo\_log\_qgc.slx>文件，在Simulink中，点击编译命令。



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。

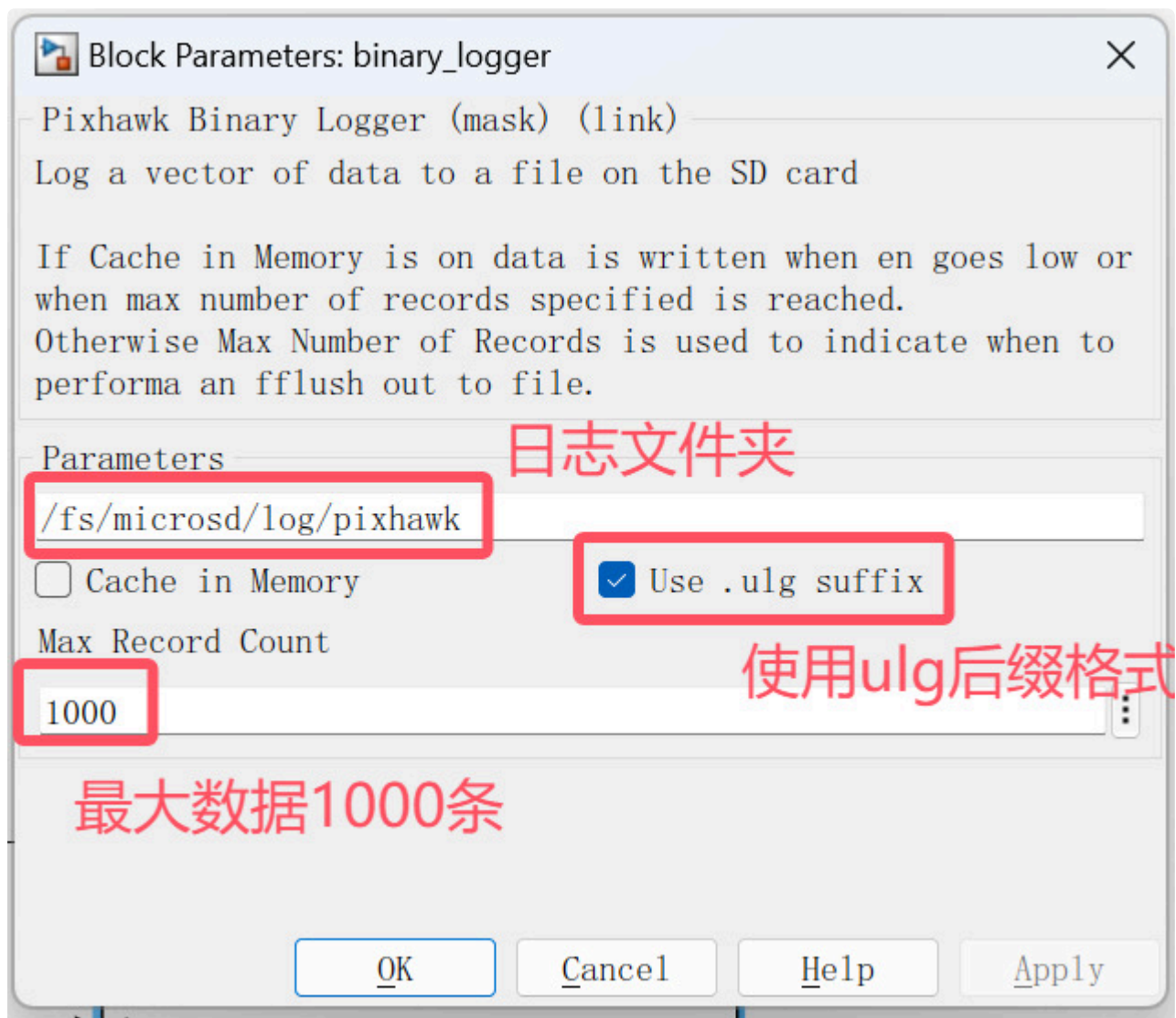


```
C:\WINDOWS\SYSTEM32\cmd.exe
Loaded firmware for board id: [redacted] size: 1903433 bytes (92.20%), waiting for the bootloader...

Found board id: [redacted] bootloader version: 5 on COM5
sn: 001e00354256500c20323441
chip: 10016451
family: b'STM32F7[6|7]x'
revision: b'Z'
flash: 2064384 bytes
Windowed mode: False

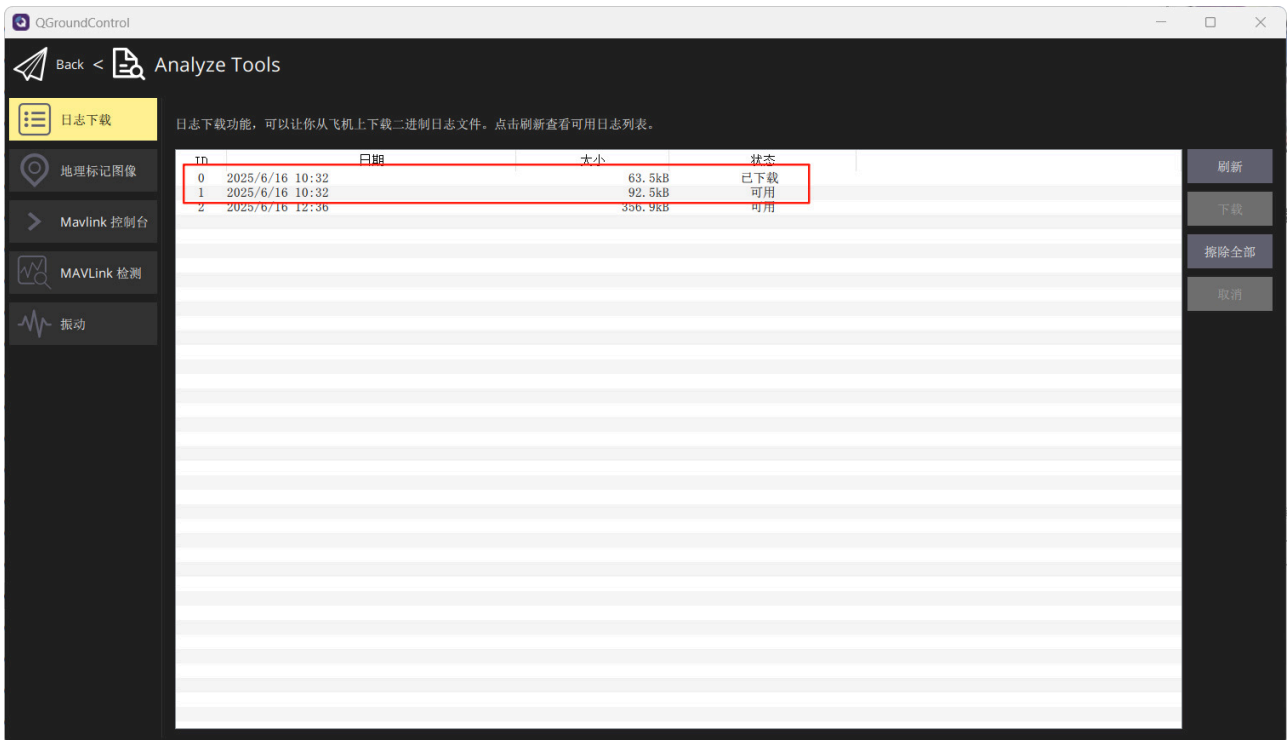
Erase : [=====] 100.0%
Program: [ ] 3.4%
```

该例程中相比于<px4demo\_log.slx>程序，自定义设置了binary\_logger模块记录的数据为.ulg格式文件，用户可通过QGC地面站下载该文件。



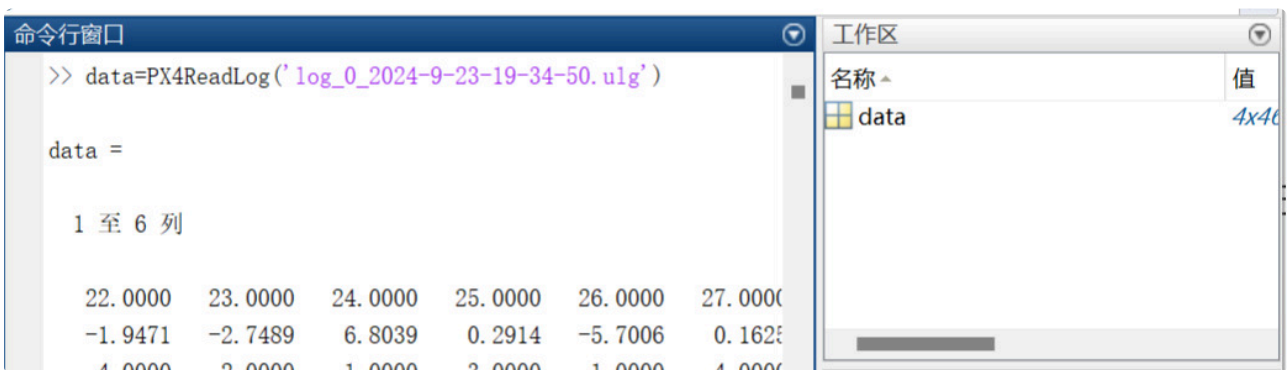
模型中定义了通过遥控器的CH5通道使能开始记录日志，在拨动CH5通道后等待20s之后，再次拨动CH5通道到低电位。即可完成数据的记录，由于该模块生的日志与PX4软件内部自动成的日志不是同一个文件，因此在进行本实验时，打开QGC地面站点击左上角“Q”字样

Logo—> “Analyze Tools” —> “日志下载” —> “刷新”，会看到一次性生成了两个日志文件，如下图所示，一般本实验生成的日志文件为第1个，但由于硬件和实验环境的差异性可能是第2个文件，因此需要将2个文件都下载（“下载”即可下载该日志文件），然后进行解析。



在MATLAB命令行中输入如下程序（将log\_0\_2024-9-23-19-34-50.ulg修改为自己的ulg文件的名称）：

```
1 PX4ReadLog('log_0_2024-9-23-19-34-50.ulg')  
2 或  
3 PX4ReadLog 'log_0_2024-9-23-19-34-50.ulg'
```



即可将记录的数据加载到MATLAB的工作区中。若解析数据如下图所示，则说明该\*.ulg日志文件为PX4内部生成的日志文件，可切换另一个日志文件进行解析。

```
F 文
G 文 >> PX4ReadLog('log_1_2025-6-16-10-32-24.ulg')
G 文
G 文 ans =
G 文
G 文     0
G 文
```

## 5. 关键知识点

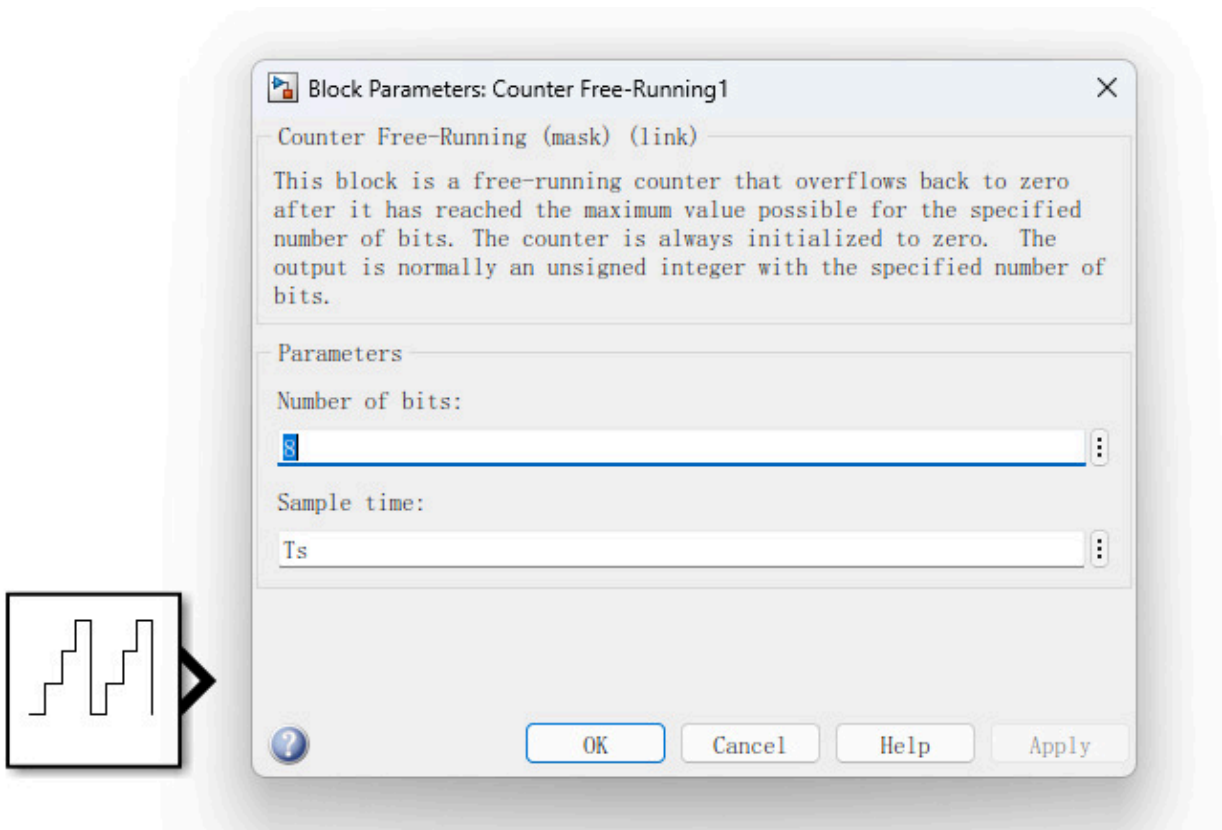
### 关键知识点1: Counter Free-Running—自由运行计数器

如下图所示，Counter Free-Running（自由运行计数器）模块是一个用于生成连续递增或递减计数序列的模块。它可以用于模拟时序信号、控制循环次数等应用场景。

#### 主要特点和用法:

- **计数方向:** 模块可以配置为递增或递减计数器，可以设置递增或递减的步长和初始值。
- **计数范围:** 可以指定计数器的范围，即计数器在何时回到初始值或停止计数。
- **输出信号:** 模块会生成一个连续的计数序列作为输出信号，可用于模拟时序信号、触发事件或者控制其他模块的行为。
- **触发方式:** 可以选择不同的触发方式来启动或停止计数器，例如基于外部触发信号或模拟时钟信号。
- **灵活性:** 模块提供了灵活的参数设置，可根据具体需求对计数器进行配置。

在达到指定位数的最大值后溢出回零。计数器总是初始化为零，输出通常是一个具有指定位数的无符号整数。

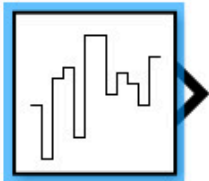


## ■ 关键知识点2: Band-Limited White Noise—带限白噪声

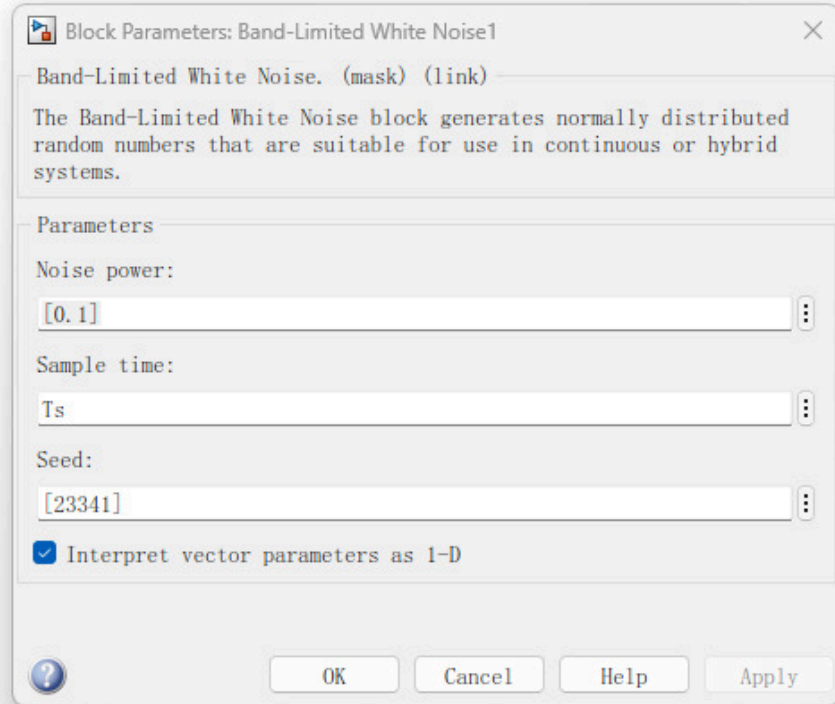
如下图所示，Band-Limited White Noise（带限白噪声）模块用于生成具有特定频率范围内的白噪声信号。白噪声是一种具有均匀功率谱密度的随机信号，在所有频率上具有相等的能量。

### 主要特点和用法：

- **频率范围**：可以通过设置模块参数来指定白噪声信号的频率范围（带宽）。
- **功率谱密度**：生成的信号具有均匀的功率谱密度，在指定的频率范围内每个频率上的功率相等。
- **采样时间**：可以设置模块的采样时间，以确定每个时间步长内生成的噪声样本数。
- **输出信号**：生成的信号可以作为模型中的输入信号，用于模拟系统的随机噪声成分。
- **参数配置**：提供了丰富的参数配置选项，包括频率范围、采样时间、随机种子等。



## Band-Limited White Noise1

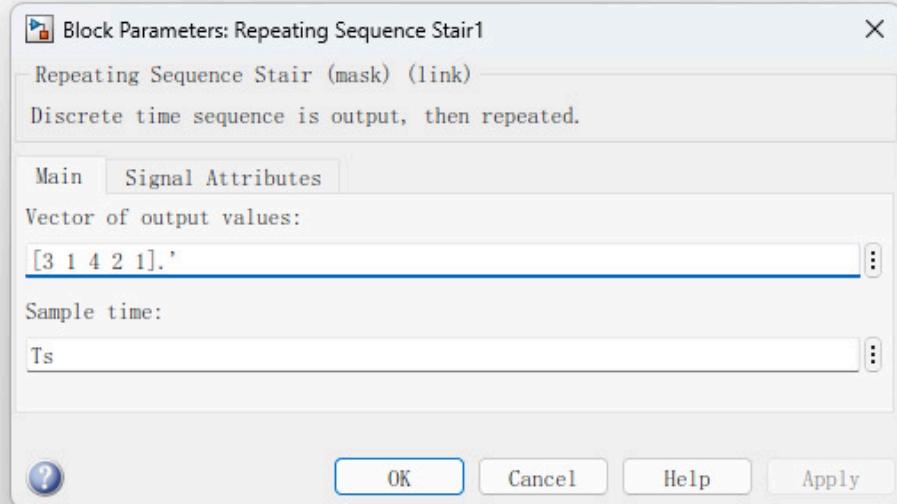
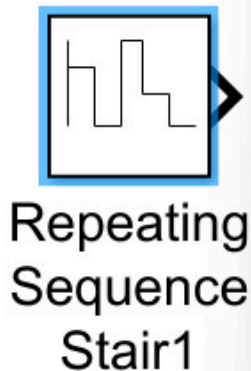


### 关键知识点3: Repeating Sequence Stair—重复序列阶梯

如下图所示，Repeating Sequence Stair（重复序列阶梯）模块用于生成重复的阶梯状序列信号。这种信号通常用于模拟周期性变化或周期性行为，例如周期性控制、测试或仿真应用中。

#### 主要特点和用法:

- **阶梯序列:** 生成一个具有阶梯状变化的序列信号，其中每个阶梯段的持续时间和幅度可以自定义。
- **重复周期:** 可以设置重复序列的周期，即序列在重复周期内重复出现。
- **阶梯数量:** 可以指定生成的阶梯数量，并为每个阶梯段指定持续时间和幅度。
- **输出信号:** 生成的信号可以作为模型中的输入信号，用于模拟系统的周期性行为或控制信号。
- **参数配置:** 提供了丰富的参数配置选项，包括重复周期、阶梯数量、每个阶梯段的持续时间和幅度等。



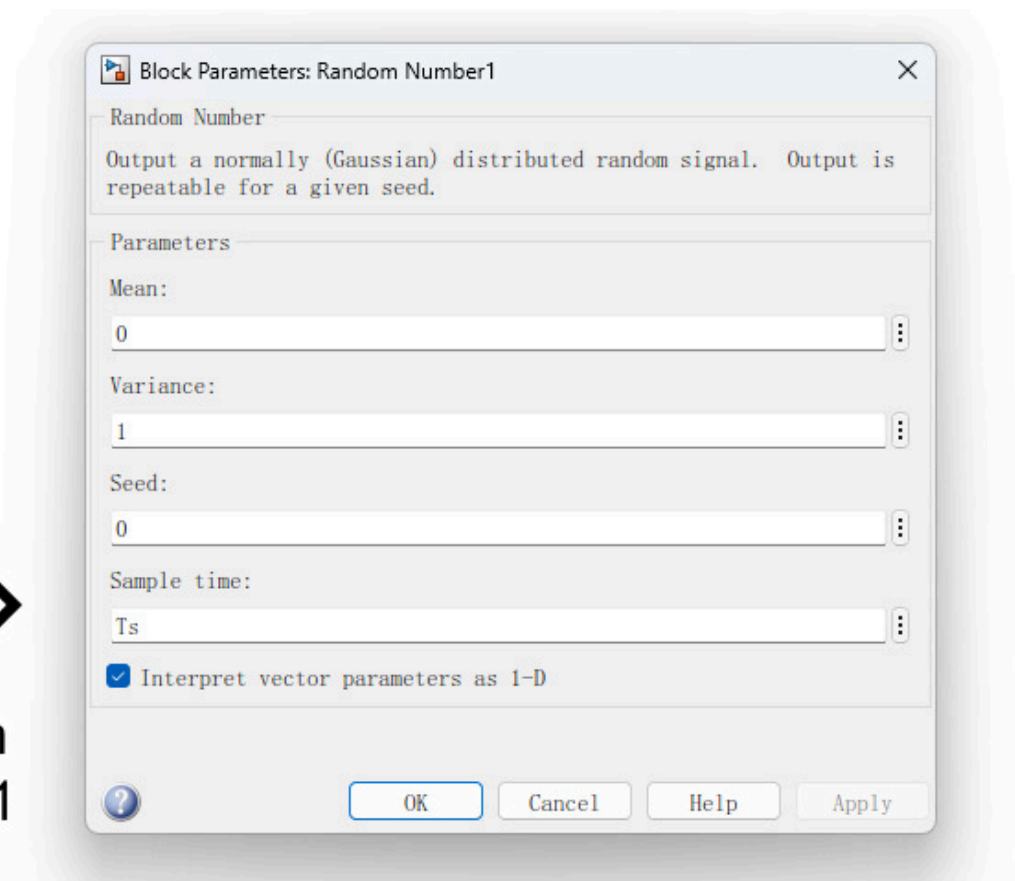
## 关键知识点4: Random Number—随机数

如下图所示，Random Number（随机数）模块用于生成各种类型的随机数信号，例如均匀分布随机数、高斯（正态）分布随机数等。这些随机数信号可以用于模拟系统中的随机性或噪声，进行概率分析，或者用于生成测试数据。

### 主要特点和用法:

- **随机数类型:** 支持生成多种类型的随机数，包括均匀分布、高斯分布、二项分布等。
- **参数设置:** 可以通过设置模块参数来调整随机数的生成方式，例如范围、分布参数、种子等。
- **输出信号:** 生成的随机数信号可以作为模型中的输入信号，用于模拟系统中的随机性或噪声。
- **统计分析:** 生成的随机数信号可以用于进行概率分析和统计分析。

通过 Random Number 模块，可以方便地在 Simulink 中生成各种类型的随机数信号，用于模拟系统中的随机性或噪声，进行概率分析，以及生成测试数据。

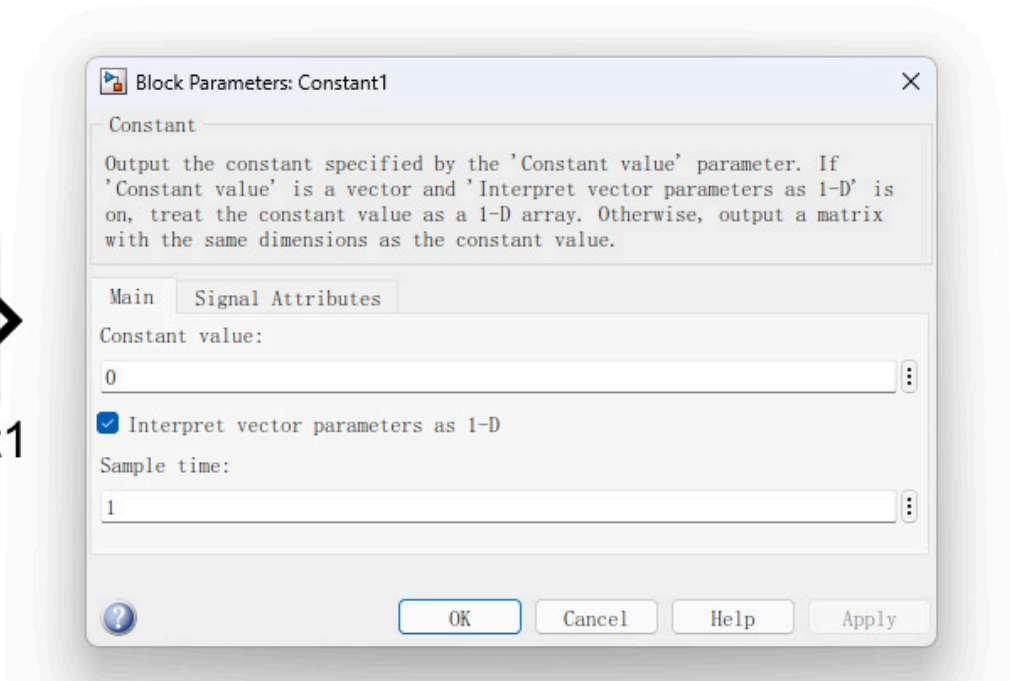
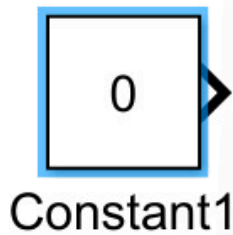


Constant—常数如下图所示，Constant（常数）模块用于生成一个固定值的信号。这个固定值可以是任何你想要的数值，可以是标量、向量或矩阵形式，用于提供模型中的常量输入或参数。

### 主要特点和用法：

- **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。
- **参数设置：**可以在模块参数设置中指定生成的固定数值。
- **输出信号：**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。
- **多维数组支持：**除了标量值外，还支持生成向量或矩阵形式的固定值信号。

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: Delay—延迟

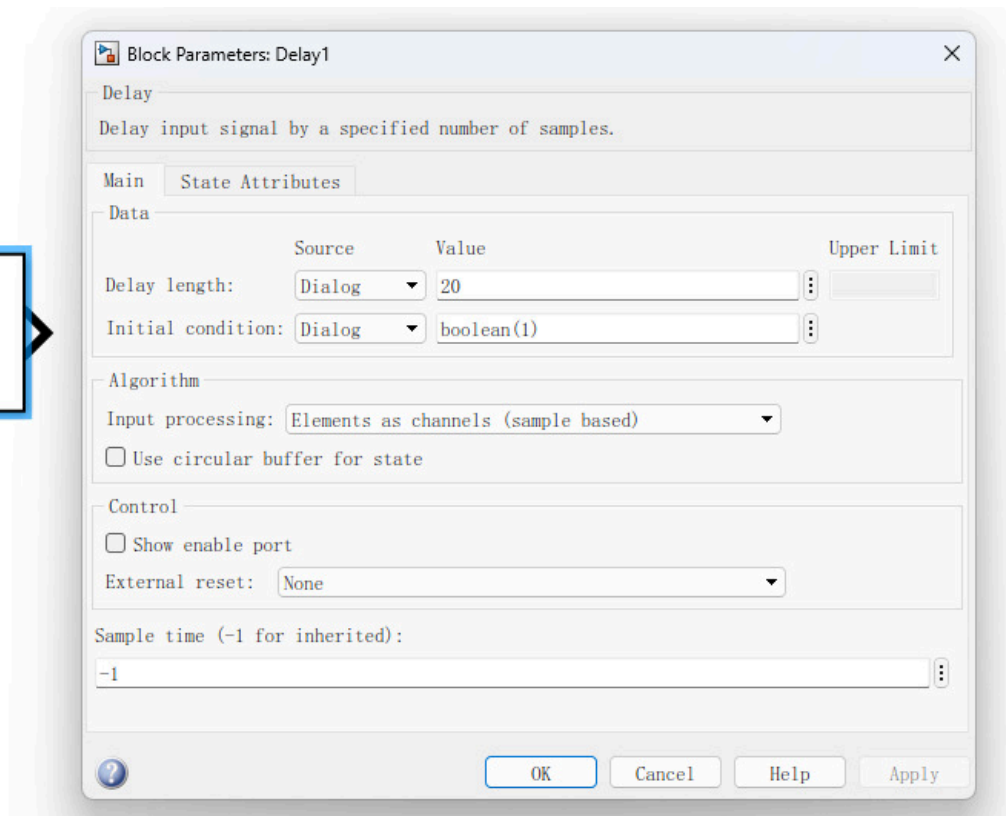
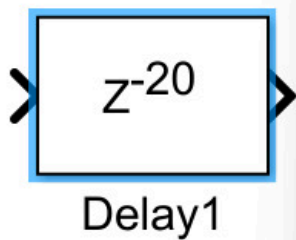
如下图所示，Delay（延迟）模块用于在信号路径中引入延迟。这个延迟可以是固定的时间延迟，也可以是可变的延迟，取决于如何配置模块。

### 主要特点和用法：

- **固定延迟**：可以在模块参数设置中指定一个固定的延迟时间。
- **可变延迟**：支持通过输入端口动态地指定延迟时间。
- **离散和连续延迟**：可以用于引入离散或连续的延迟，具体取决于信号的采样时间。
- **参数设置**：除了延迟时间外，还可以设置初始化延迟、延迟类型等参数。
- **缓冲**：在引入延迟的同时，模块会在内部使用缓冲来存储延迟的输入信号。

通过 Delay 模块，你可以在 Simulink 中方便地引入固定或可变的延迟，用于模拟系统中的信号传播延迟、处理器执行延迟等情况，从而更准确地建模和仿真系统的行为。

本例程中，引用了20个单位延迟，表示信号将被延迟20s的时间后输出。



## 关键知识点6: Data Type Conversion—数据类型转换

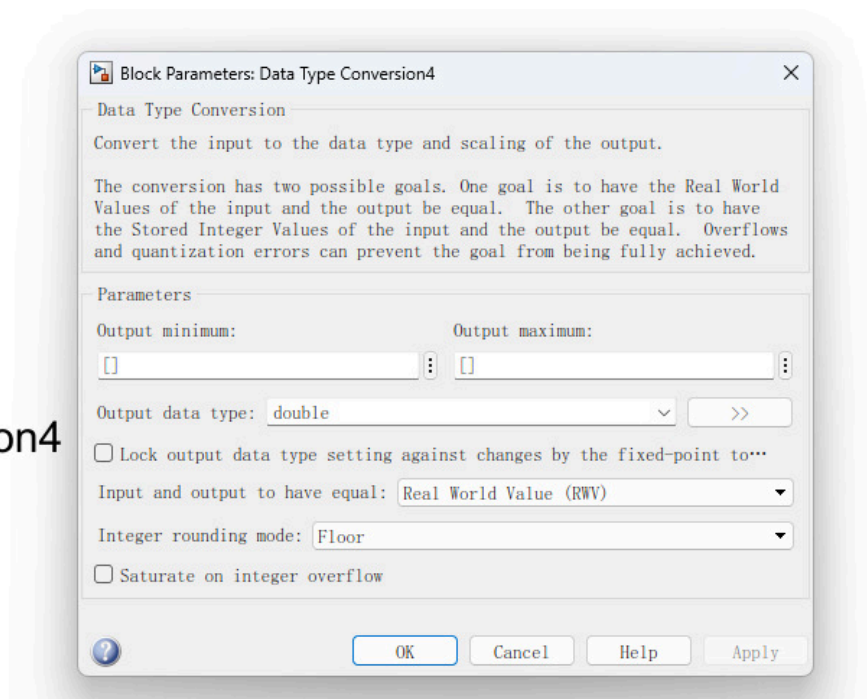
如下图所示，Data Type Conversion（数据类型转换）模块用于将输入信号的数据类型转换为所需的输出数据类型。这个模块在处理不同数据类型之间的转换时非常有用，例如将整数转换为浮点数，或者将固定点数转换为双精度浮点数等。

### 主要特点和用法:

- **数据类型转换:** 可以将输入信号的数据类型转换为指定的输出数据类型，如整数、浮点数、固定点数等。
- **精度控制:** 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。
- **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。
- **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。
- **输出端口数目:** 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。

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。

## double Data Type Conversion4

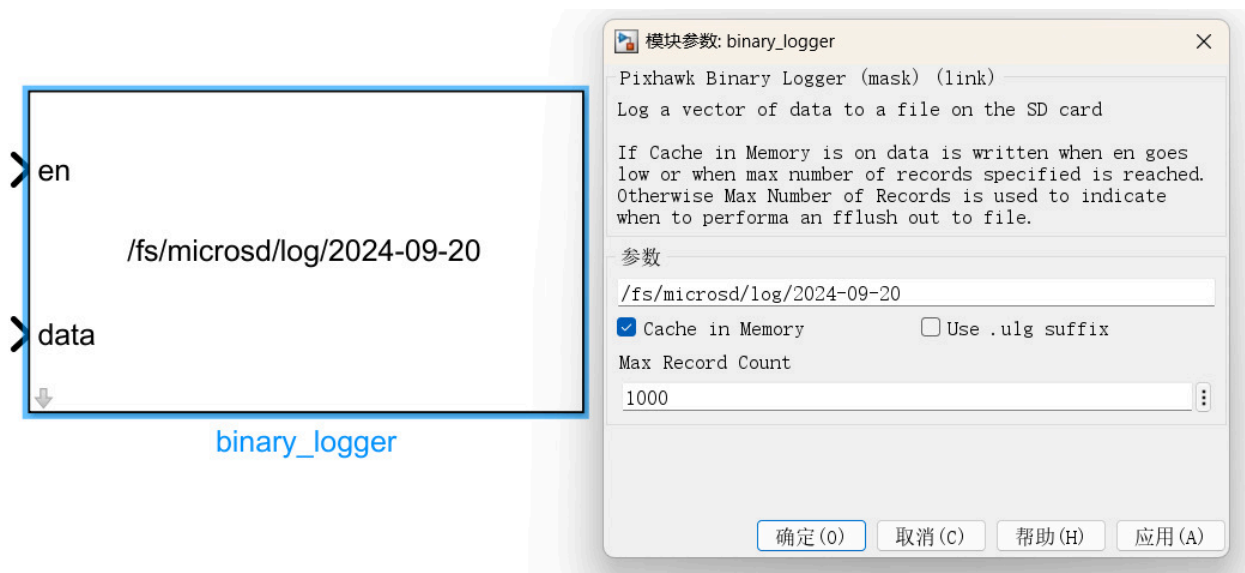


## 关键知识点7: Binary\_logger—数据记录模块

如下图所示，这个模块将数据记录为double型.bin或.ulg格式文件存储到SD卡中，如果在内存中缓存打开，则当en变低或达到指定的最大记录数时写入数据。在程序执行过程中第一个使能输入信号必须先触发高电平再降为低电平才能成功地记录数据，如果在代码结束运行之前没有降为低电平，那么记录文件将被认为处于打开状态，不可访问。此外，日志必须存储在目录"/fs/microsd/"下。并且可以设置日志存储路径及最大记录条数。

### 核心要点:

- 支持.bin和.ulg两种日志格式
- 数据先缓存后写入SD卡
- 使能信号en控制记录启停（首次需先高后低）
- 必须存储在"/fs/microsd/"目录下
- 可设置最大记录条数



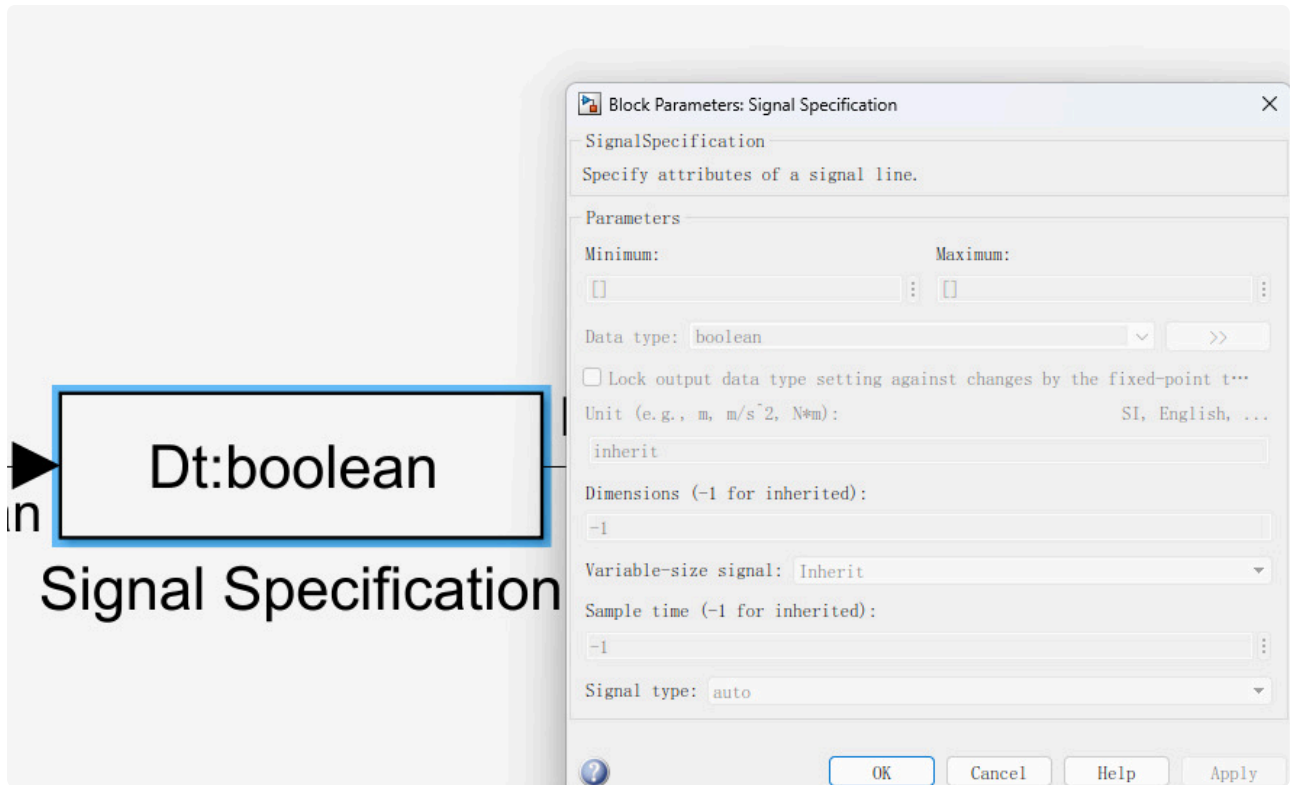
## 关键知识点8: Signal Specification—信号规范

如下图所示，在 Simulink 中，Signal Specification（信号规范）模块用于指定信号的特定属性，如数据类型、尺寸和采样时间等。这个模块通常用于定义输入信号的属性，以确保与其他模块的输入端口匹配。

### 主要特点和用法：

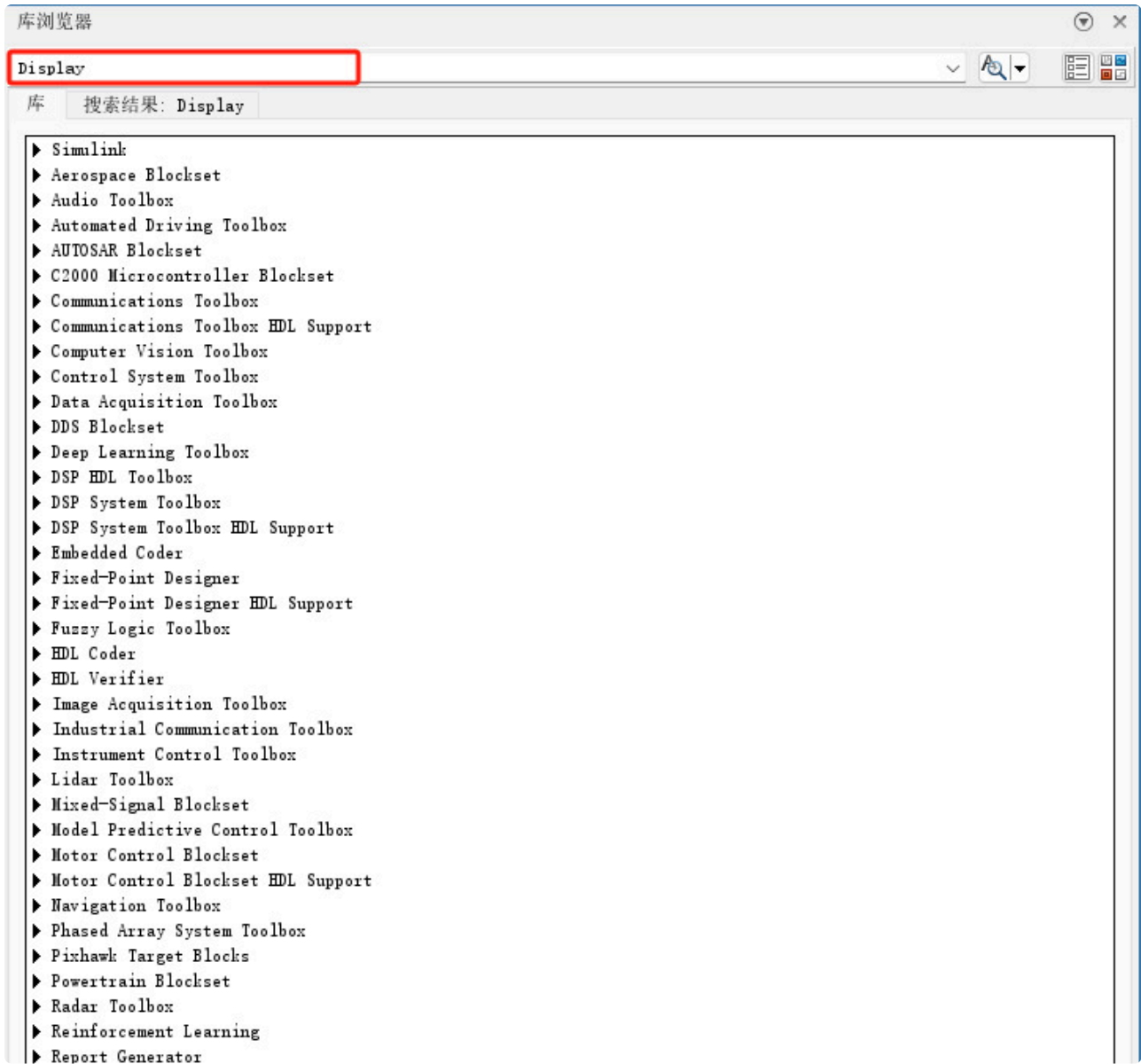
- **数据类型指定：**可以指定输入信号的数据类型，例如整数、浮点数、布尔值等。
- **信号尺寸指定：**可以定义输入信号的尺寸，包括向量或矩阵的维度。
- **采样时间指定：**可以设置输入信号的采样时间，以指定信号的更新频率。
- **单位指定：**可以指定信号的单位，例如长度单位、时间单位等。
- **数据范围指定：**可以指定输入信号的数据范围，例如最小值和最大值。

Signal Specification 模块通常用于模型的输入端口，以确保输入信号的属性与系统中其他模块的要求相匹配。通过指定信号的数据类型、尺寸、采样时间等属性，可以有效地管理模型中的信号流，并确保模型的准确性和一致性。



## 关键知识点9:其他模块

如下图所示，更多Simulink官方模块的详细用法和功能，请到Simulink界面的库浏览器中搜索，找到对应的模块，双击该模块，点"帮助"即可查看该模块的详细用法和功能。





## 6. 参考资料

1. [RflySim官方文档](#)
2. [PX4官方文档 - 日志分析](#)
3. [PX4 Logger模块设计](#)
4. [QGroundControl日志下载说明](#)
5. [pyulog - PX4日志解析工具](#)
6. [PX4 uORB消息传递机制](#)

## 7. 常见问题

### Q1: Binary Logger模块无法记录数据?

A1: 检查以下几点:

- 确保使能信号(en)的时序正确: 首次触发需要先高电平再降为低电平才能开始记录
- 检查SD卡是否正确挂载且有足够存储空间
- 确认日志存储路径是否正确设置为"/fs/microsd/log/"
- 查看飞控板上的SD卡指示灯状态是否正常

### Q2: 读取.ulg日志文件时出现数据不完整的情况?

A2: 可能原因及解决方法:

- 检查是否在代码结束运行前将使能信号置为低电平, 否则日志文件可能处于打开状态无法访问
- 确保记录完成后正确卸载SD卡再取出
- 检查SD卡是否存在坏块或者文件系统错误
- 如果使用QGroundControl下载, 尝试刷新并重新下载日志文件

### Q3: 生成的日志文件为空或数据异常?

A3: 检查以下配置:

- 确认模型中的信号连接是否正确
- 检查各模块参数设置是否合适
- 验证采样时间配置是否一致
- 确保飞控固件版本与RflySim工具链兼容 (推荐PX4固件版本1.14.3)

### Q4: 如何选择.bin和.ulg格式?

A4: 两种格式的区别和适用场景:

- .bin格式: 二进制原始格式, 文件较小, 适合对存储空间敏感的应用
- .ulg格式: PX4统一日志格式, 兼容性更好, 可直接使用QGroundControl和Flight Review等工具分析

- 根据实际需求选择合适的格式，一般推荐使用.ulg格式以便于后续分析
- 

1. <https://rflysim.com/> ↩

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