

| simulink配置读取与写入uORB消息实验

| 1. 实验目的

自定义uORB消息，PX4的uORB消息系统是提供了非常强大且方便的内部模块间数据交互能力，本例程是通过创建一个自定义的uORB消息实现读写功能，以此熟悉并掌握PX4的uORB消息系统。

| 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],遥控器和遥控器接收机；数据线和杜邦线等^[3]。

| 3. 实验地址

例程目录：[\[安装目录\]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\6.uORB-Read-Write](#)

- Readme.pdf:uORB消息读取与写入实验步骤
- Exp1_uORB_read.slx:uORB消息读取程序
- Exp2_uORB_read_func.slx:uORB消息读取触发程序
- Exp3_uORB_write.slx:uORB消息发送程序
- Exp4_uORB_DifID.slx:uORB消息ID与文件名不一致处理程序
- PX4ExtMsgSender.slx:飞控uORB消息监听程序

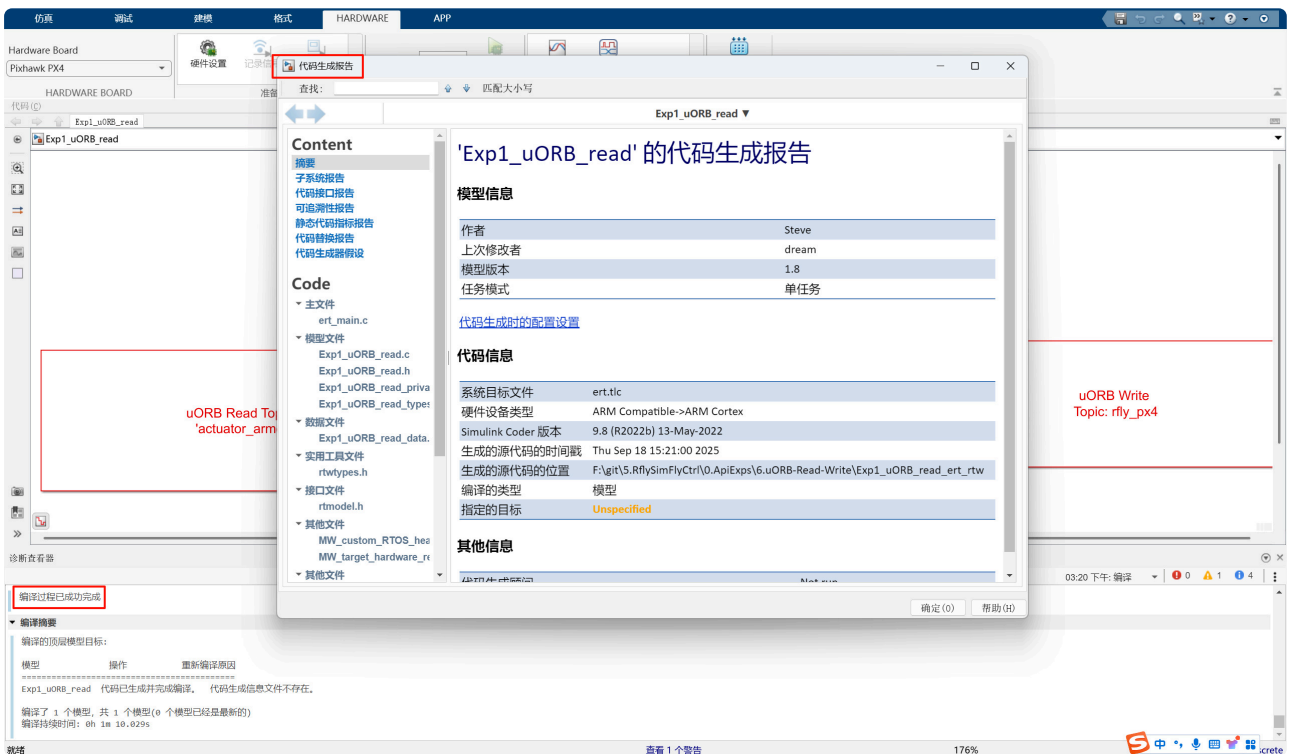
4. 实验内容或步骤

5.1 步骤1：uORB消息读取实验

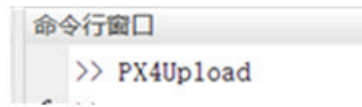
打开MATLAB软件，在MATLAB中打开Exp1_uORB_read.slx文件，在Simulink中，点击编译命令。



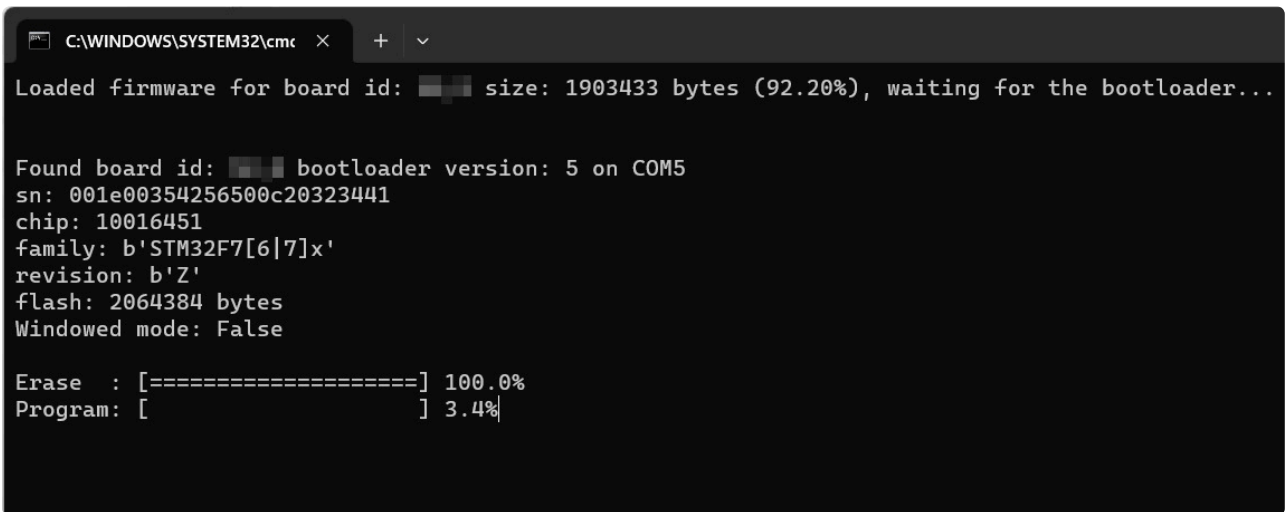
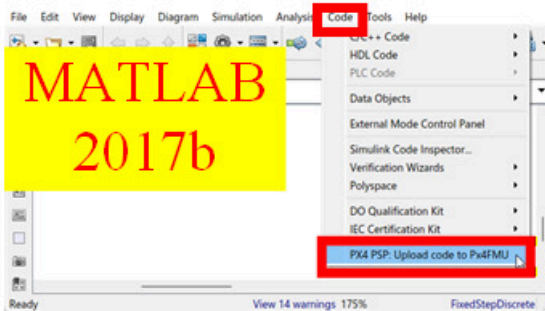
在Simulink的下方点击查看诊断，即可弹出诊断对话框，可查看编译过程。在诊断框中弹出编译过程已成功完成，即可表示编译成功，也会弹出代码生成报告。



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。



或



如下图所示，打开QGroundControl软件，点击左上角Logo在弹出的对话框中，选中Analyze Tools，选择Mavlink控制台：

在本nsh页面，可以输入一些命令，来查看飞控内部的数据运行状态。参考资料见：

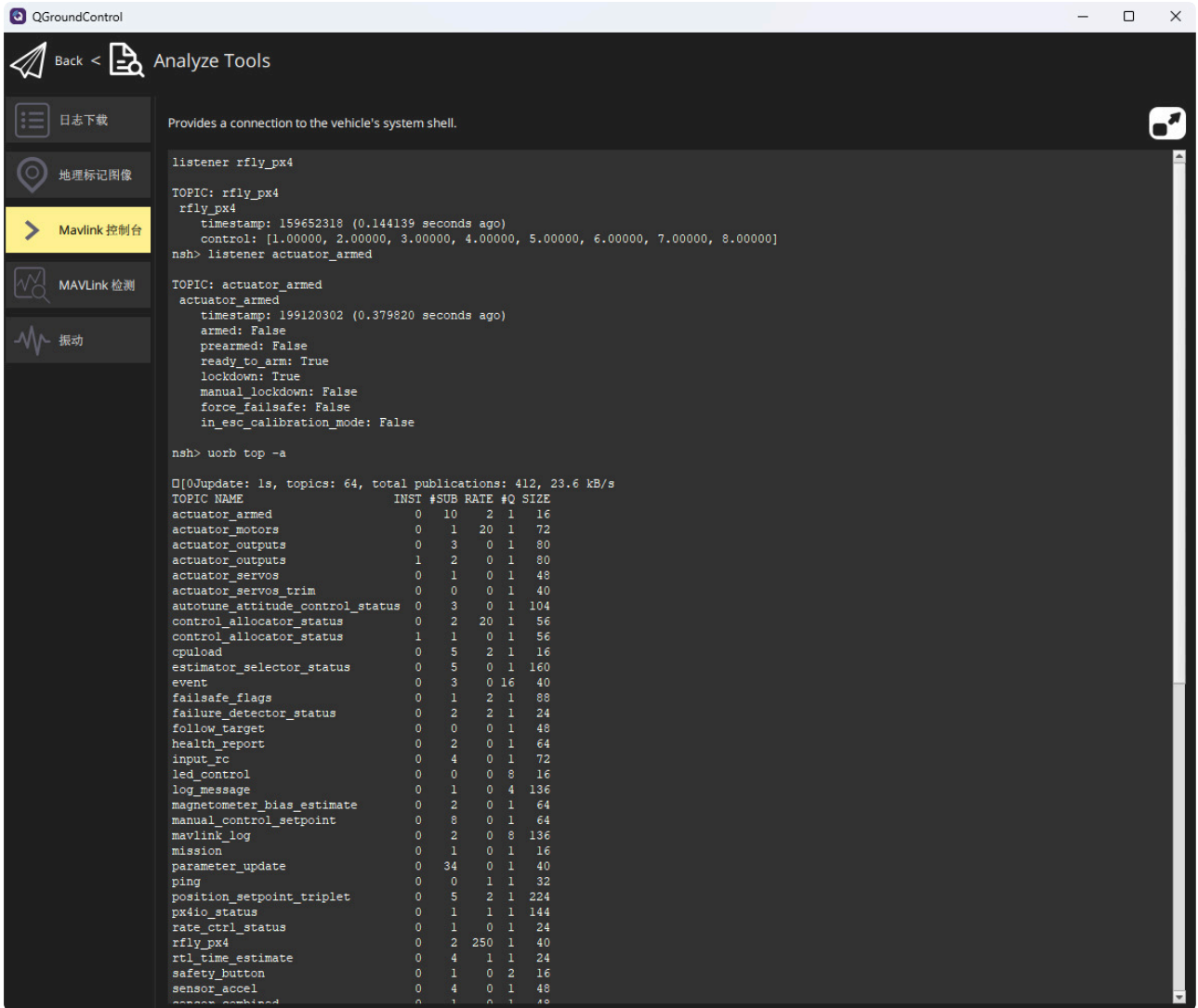
https://docs.qgroundcontrol.com/master/en/qgc-user-guide/analyze_view/mavlink_console.html

或 https://docs.px4.io/main/en/debug/consoles.html#console_vs_shell

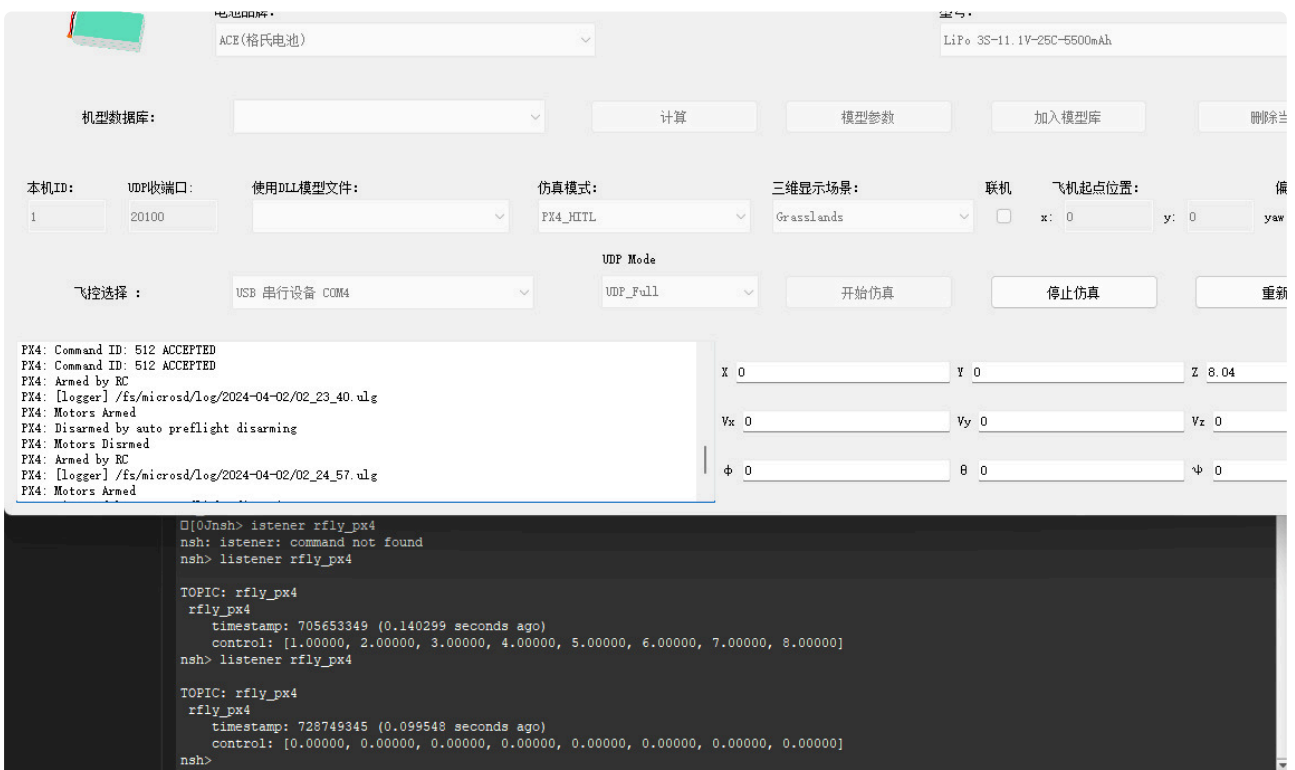
输入“?”可以查看支持的命令，输入“命令?”可以查看命令使用方法。

分别输入“listener rfly_px4”再输入“listener actuator_armed”，查看两条消息，发现符合逻辑。

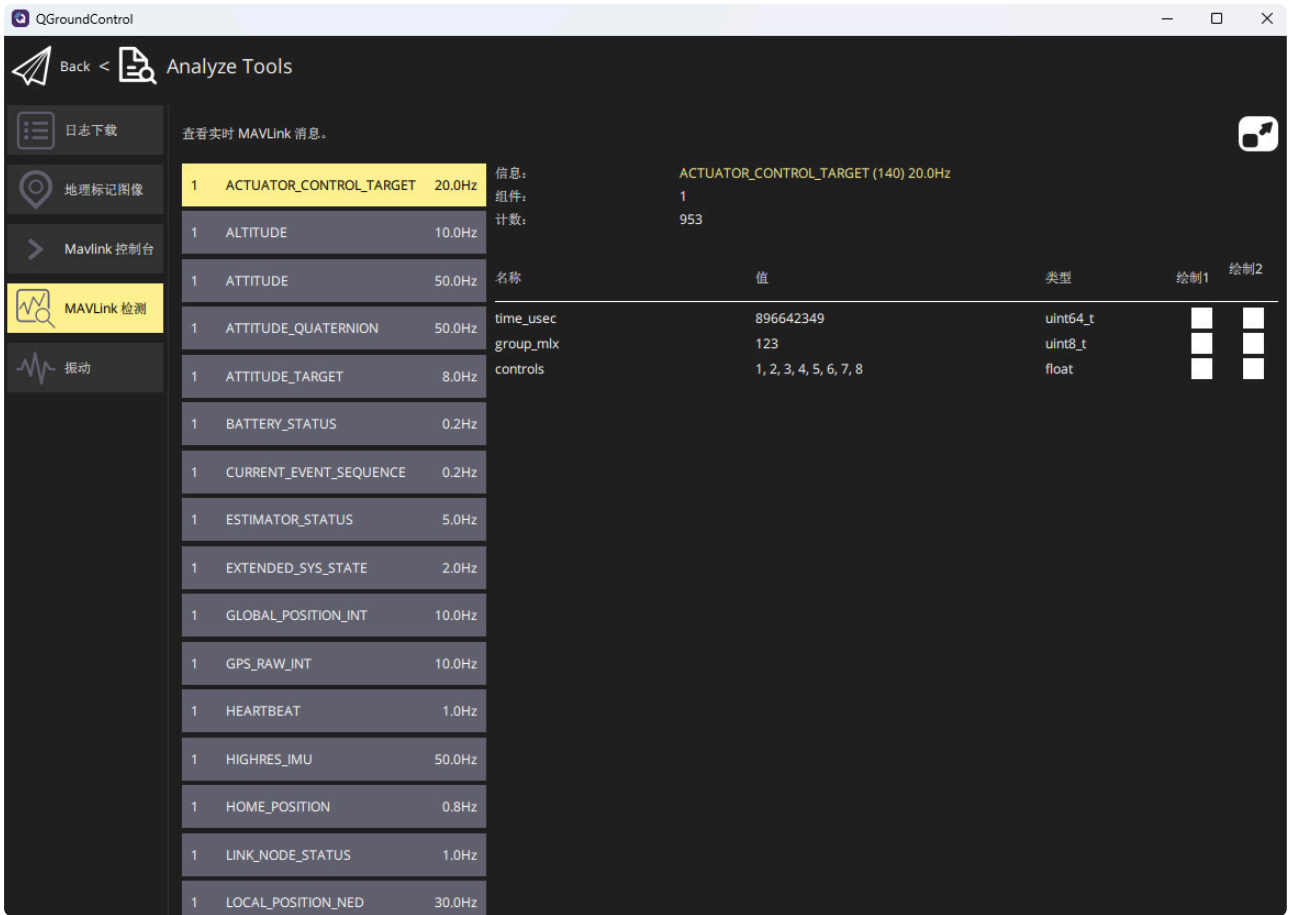
输入命令“uorb top -a”查看所有消息的发布频率。发现rfly_px4是以250Hz频率发布的，满足实验设定。



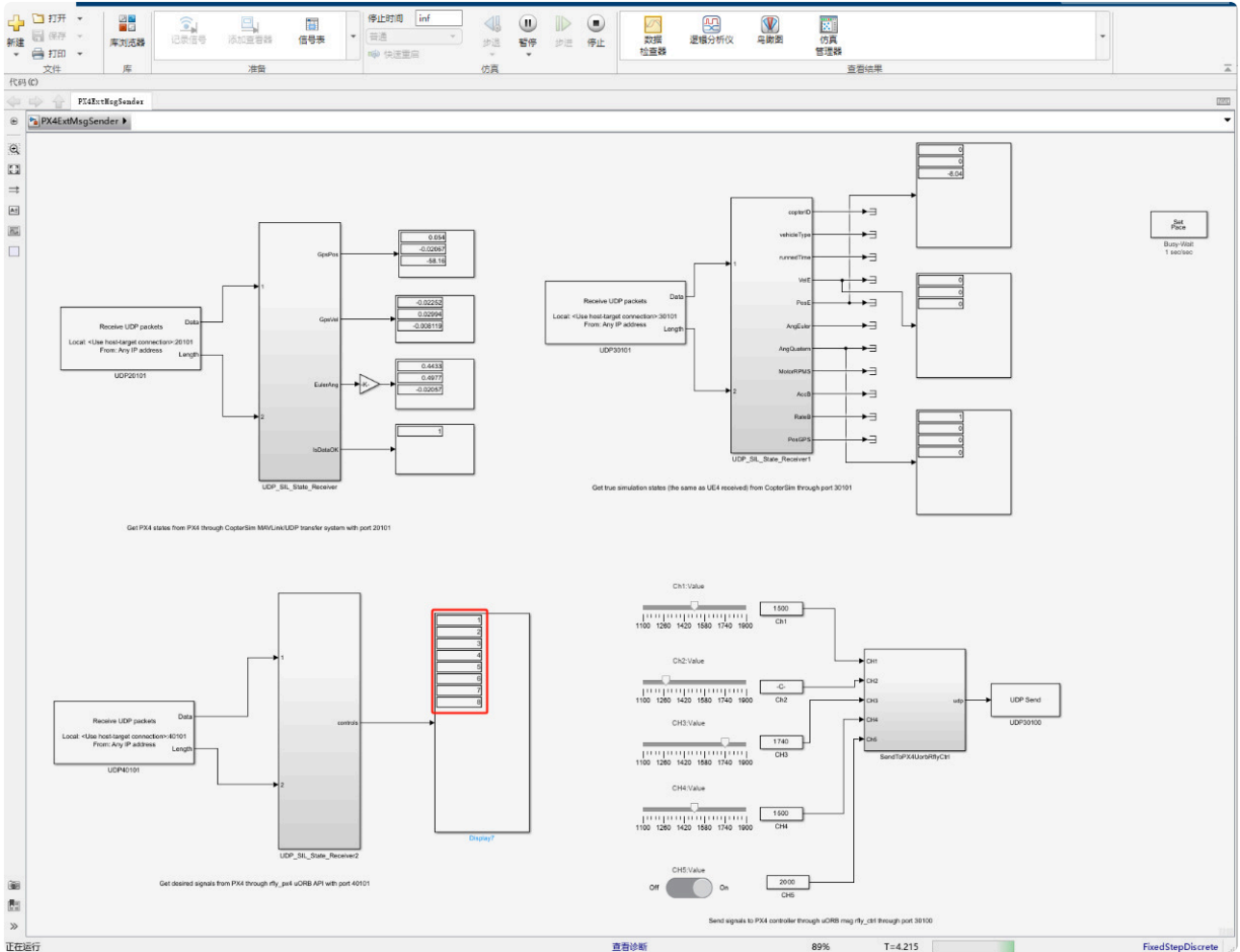
HITLRun开启硬件在环仿真，并用QGC或遥控器解锁飞控，输入“listener rfly_px4”，效果如下图。输出为全0，满足程序设定。



扩展实验1。待飞控重新上锁，在“MAVLink检测”页面，可以观察ACTUATOR_CONTROL_TARGET消息，观察到controls为12345678，满足设定值。



扩展实验2。在Simulink中直接运行5.RflySimFlyCtrl\0.ApiExps\9.PX4CtrlExternalTune\PX4ExtMsgSender.slx程序（注意，在PC上运行，而不是烧入飞控），通过其中的“UDP40101”模块，可以观察到12345678的输出。注：CopterSim在收到ACTUATOR_CONTROL_TARGET的消息（包含rfly_px4）会转发到40100系列端口，被其他程序接收并显示。

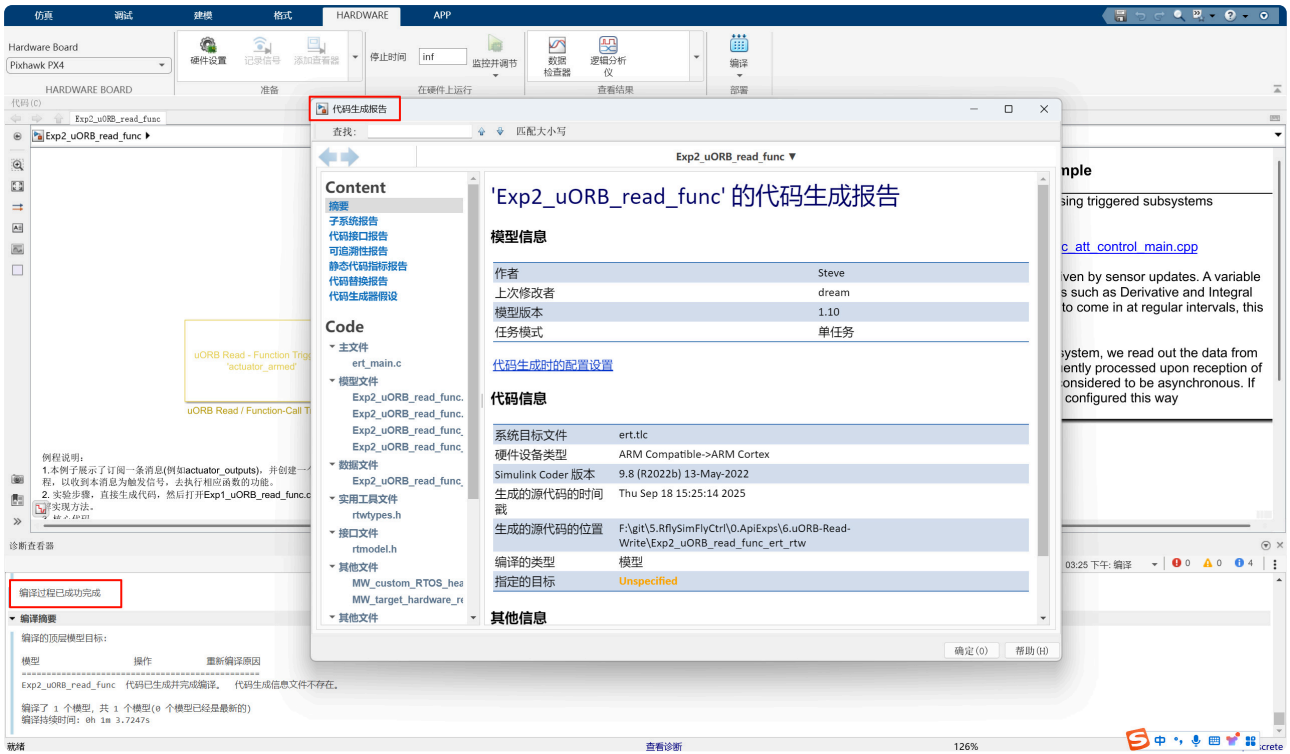


5.2 步骤2：uORB消息读取触发实验

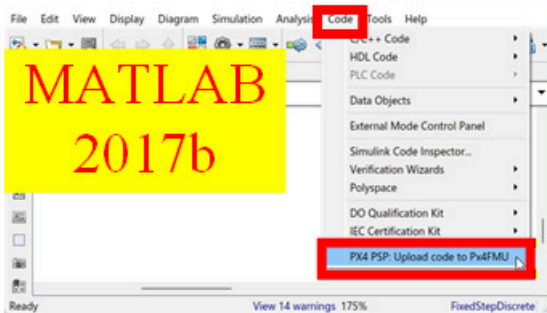
打开MATLAB软件，在MATLAB中打开Exp2_uORB_read_func.slx文件，在Simulink中，点击编译命令。



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  x  +  v
Loaded firmware for board id: [REDACTED] size: 1903433 bytes (92.20%), waiting for the bootloader...

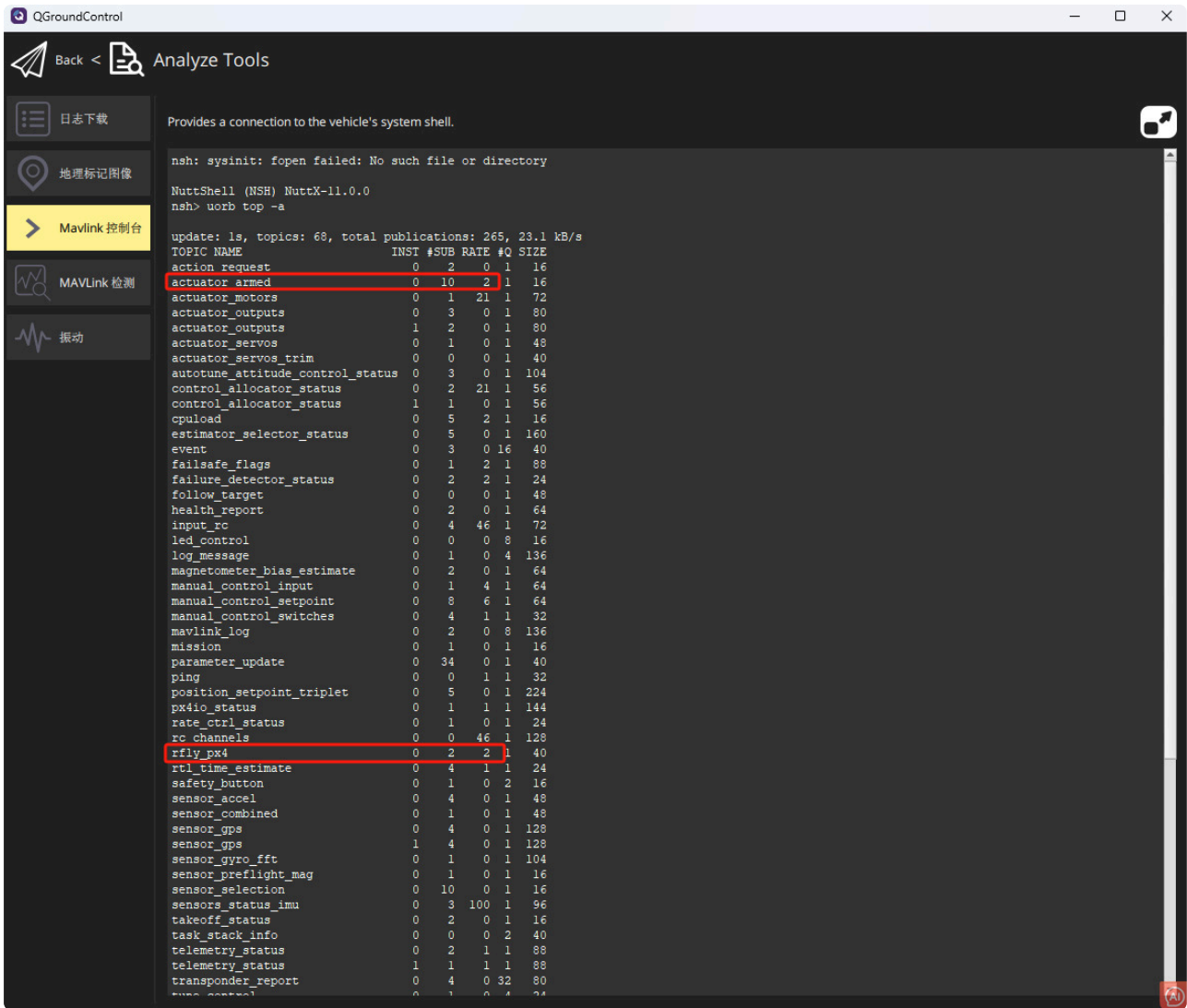
Found board id: [REDACTED] bootloader version: 5 on COM5
sn: 001e00354256500c20323441
chip: 10016451
family: b'STM32F7[6|7]x'
revision: b'Z'
flash: 2064384 bytes
Windowed mode: False

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

如下图所示，打开QGroundControl软件，点击左上角Logo在弹出的对话框中，选中Analyze Tools，选择Mavlink控制台：

输入命令“uorb top -a”查看所有消息的发布频率。和“Exp1_uORB_read.slx”实现相比，rfly_px4的发布频率没有跟随设置中的250Hz，而是和飞控actuator_armed消息一致，满足2Hz频率。

本实验说明，通过uORB Read Function-Call Trigger模块的函数触发功能，能够控制数据的处理频率，节省计算量。

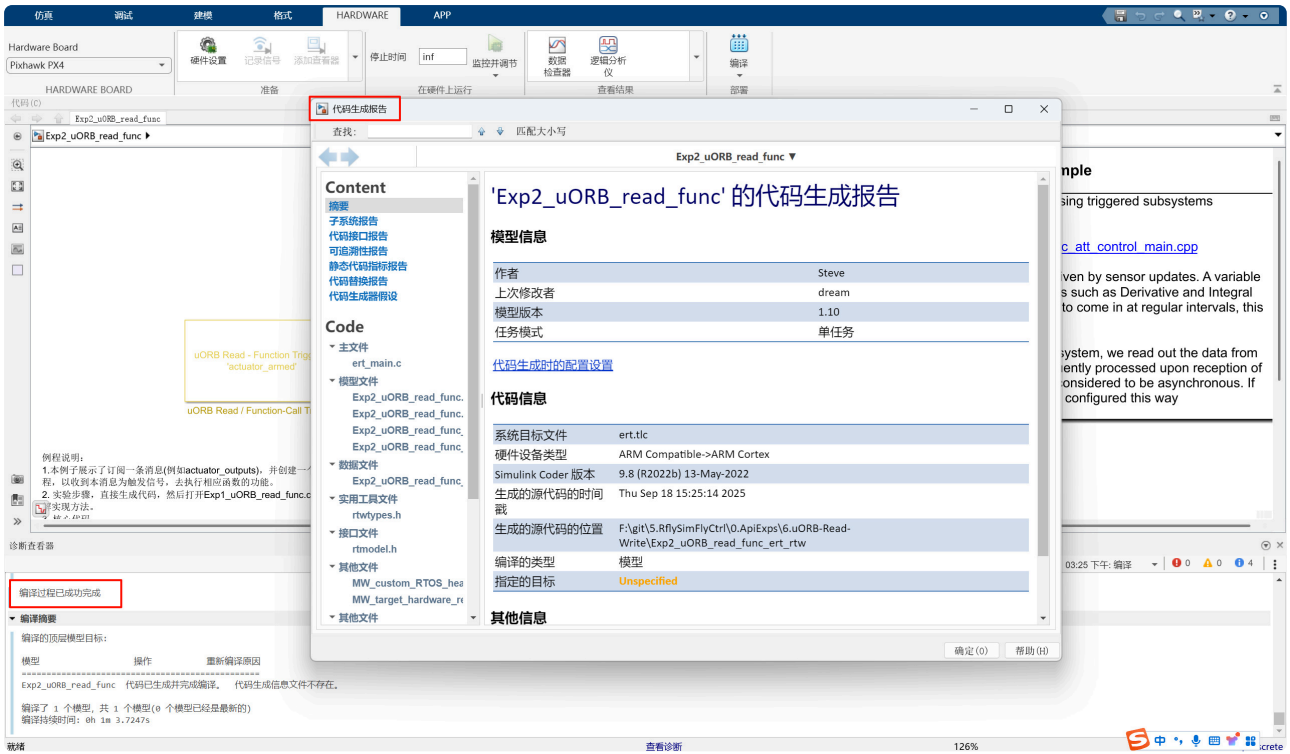


5.3 步骤3：uORB发送接口使用实验

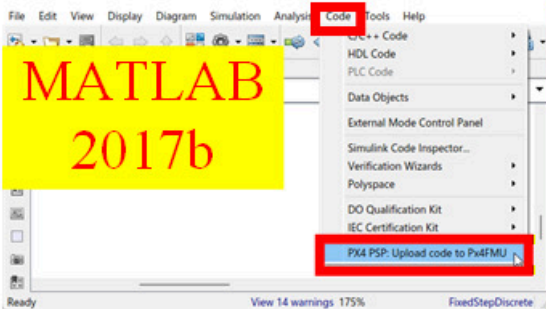
打开MATLAB软件，在MATLAB中打开Exp3_uORB_write.slx文件，在Simulink中，点击编译命令。



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%
```

如下图所示，打开 “*\Desktop\RflyTools\HITLRun.lnk”，在弹出的对话框中输入飞控 COM号如：4。

```
C:\WINDOWS\system32\cmd.exe

-----
Please input the Pixhawk COM port list for HITL
Use ',' as the separator if more than one Pixhawk
E.g., input 3 for COM3 of Pixhawk on the computer
Input 3,6,7 for COM3, COM6 and COM7 of Pixhawks
-----
All COM ports on this computer are:

COM3: Intel(R) Active Management Technology - SOL (unavailable or busy)
COM4: USB 串行设备 * (Pixhawk with SysID=1)

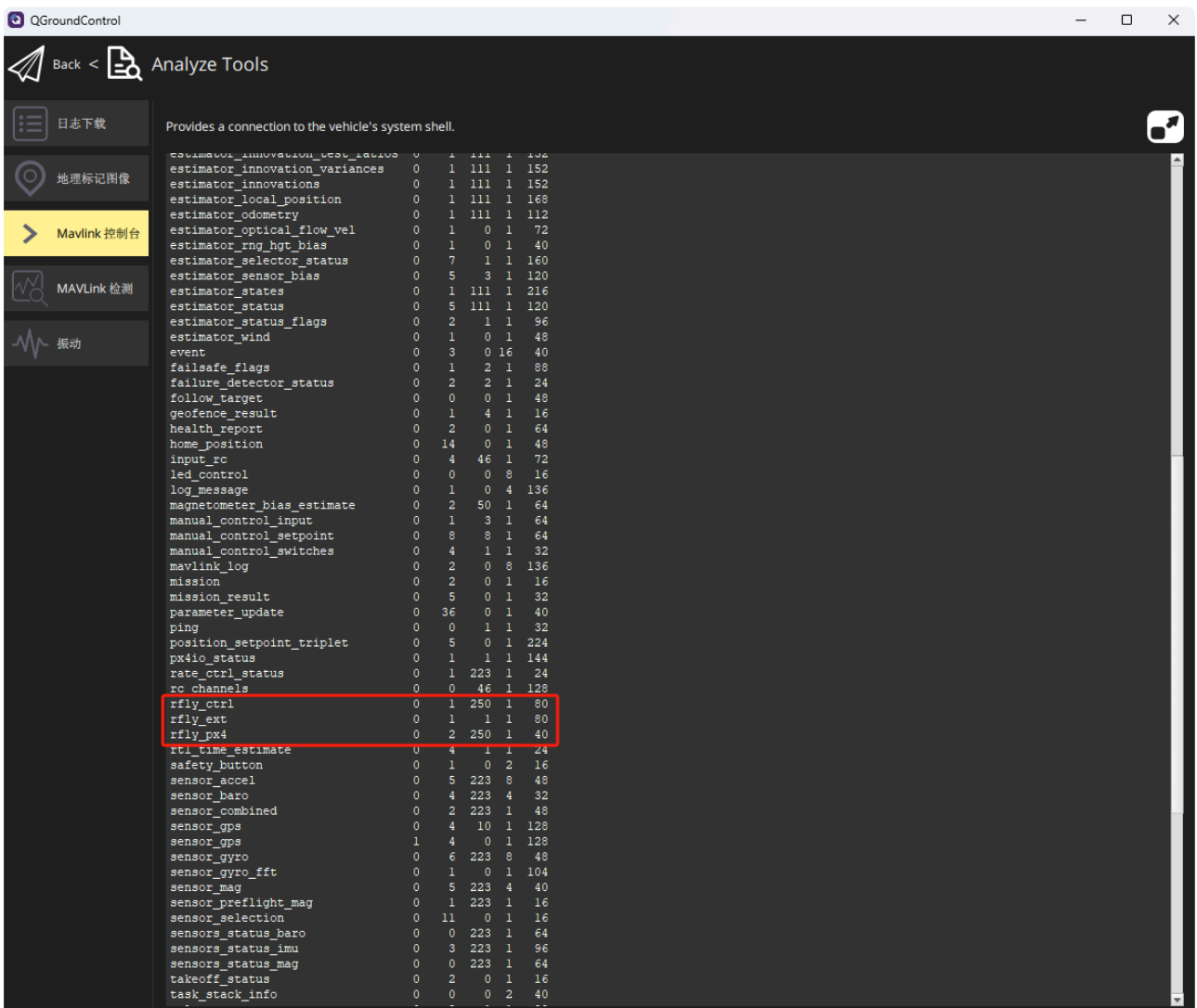
-----
Recommended COM list input is: 4
-----

My COM list for HITL simulation is: _
```

等待仿真环境初始化完成。脚本将会启动 1 个 QGC 地面站，1 个 CopterSim、1 个 RflySim3D 软件，等待 CopterSim 软件下侧日志栏必须打印出 GPS 3D fixed & EKF initialization finished 字样代表初始化完成。如下图所示：



在QGC左上角Logo在弹出的对话框中，选中Analyze Tools，选择Mavlink控制台：输入命令“uorb top -a”，可以发现收到了rfly_ctrl、rfly_px4、rfly_ext三条消息，且前两条是设定的250Hz，后两条是设定的1Hz。



在未解锁情况下，用listener接口打印三条消息，打印rfly_px4，数据从1到8，打印rfly_ctrl和rfly_ext，数据从1到16，符合程序编程原理。

```
nsh> listener rfly_px4
TOPIC: rfly_px4
rfly_px4
timestamp: 193462337 (0.010381 seconds ago)
control: [1.00000, 2.00000, 3.00000, 4.00000, 5.00000, 6.00000, 7.00000, 8.00000]
nsh> listener rfly_ctrl
TOPIC: rfly_ctrl
rfly_ctrl
timestamp: 230842154 (0.000774 seconds ago)
flags: 0
controls: [1.00000, 2.00000, 3.00000, 4.00000, 5.00000, 6.00000, 7.00000, 8.00000, 9.00000, 10.00000, 11.00000, 12.00000, 13.00000, 14.00000, 15.00000, 16.00000]
modes: 0
nsh> listener rfly_ext
TOPIC: rfly_ext
rfly_ext
timestamp: 235834155 (0.326696 seconds ago)
controls: [1.00000, 2.00000, 3.00000, 4.00000, 5.00000, 6.00000, 7.00000, 8.00000, 9.00000, 10.00000, 11.00000, 12.00000, 13.00000, 14.00000, 15.00000, 16.00000]
modes: 0
nsh>
```

先解锁飞控，打印rfly_px4，再锁定飞控，打印rfly_px4，发现解锁后，数据变为全0，未解锁时，为1到8，符合编程要求。

飞控选择: USB 串行设备 COM4 VNF_Full 开始仿真 停止仿真 重新仿真

FX4: Command ID: 512 ACCEPTED
FX4: GPS 3D fixed & EKF initialization finished.
FX4: Preflight Fail: ekf2 missing data
FX4: GCS connection regained
FX4: Command ID: 512 ACCEPTED
FX4: Command ID: 512 ACCEPTED
FX4: Armed by RC
FX4: [logger] /fs/microsd/log/2024-04-02/02_55_53_ulg
FX4: Motors Armed

modes: 0

nsh> listener rfly_ext
TOPIC: rfly_ext
rfly_ext
timestamp: 235834155 (0.326696 seconds ago)
controls: [1.00000, 2.00000, 3.00000, 4.00000, 5.00000, 6.00000, 7.00000, 8.00000, 9.00000, 10.00000, 11.00000, 12.00000, 13.00000, 14.00000, 15.00000, 16.00000]
modes: 0

nsh> listener rfly_px4
TOPIC: rfly_px4
rfly_px4
timestamp: 341364384 (0.348304 seconds ago)
control: [0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00000]

FX4: Preflight Fail: ekf2 missing data
FX4: GCS connection regained
FX4: Command ID: 512 ACCEPTED
FX4: Command ID: 512 ACCEPTED
FX4: Armed by RC
FX4: [logger] /fs/microsd/log/2024-04-02/02_55_53_ulg
FX4: Motors Armed
FX4: Disarmed by auto preflight disarming
FX4: Motors Disarmed

timestamp: 235834155 (0.326696 seconds ago)
controls: [1.00000, 2.00000, 3.00000, 4.00000, 5.00000, 6.00000, 7.00000, 8.00000, 9.00000, 10.00000, 11.00000, 12.00000, 13.00000, 14.00000, 15.00000, 16.00000]
modes: 0

nsh> listener rfly_px4
TOPIC: rfly_px4
rfly_px4
timestamp: 341364384 (0.348304 seconds ago)
control: [0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00000]

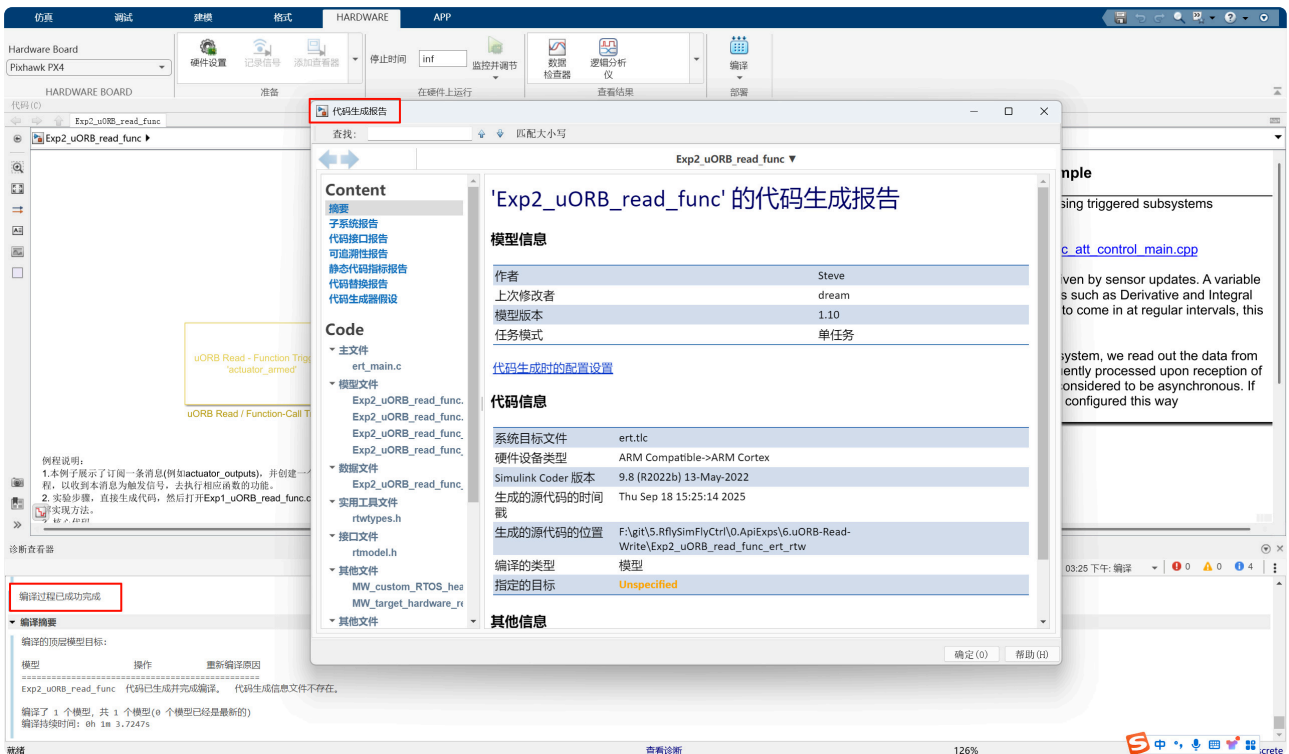
nsh> listener rfly_px4
TOPIC: rfly_px4
rfly_px4
timestamp: 373143308 (0.446772 seconds ago)
control: [1.00000, 2.00000, 3.00000, 4.00000, 5.00000, 6.00000, 7.00000, 8.00000]

5.4 步骤4：uORB ID与文件名不一致处理实验

打开MATLAB软件，在MATLAB中打开Exp4_uORB_DifID.slx文件，在Simulink中，点击编译命令。



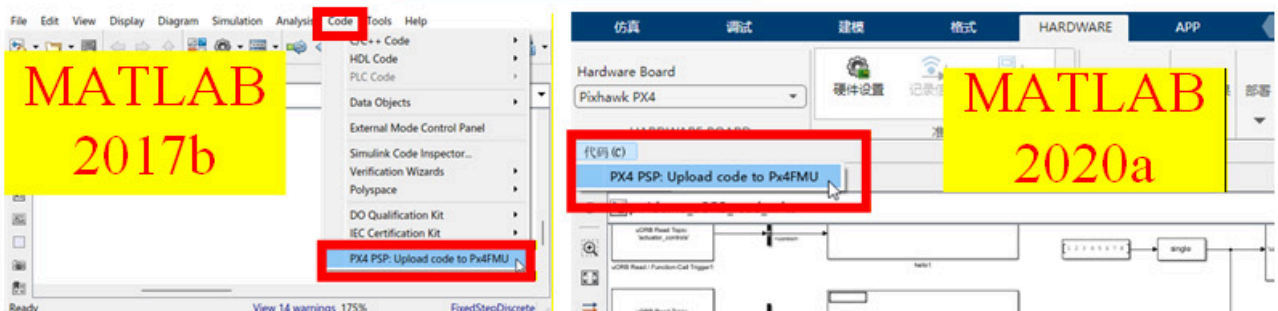
在Simulink的下方点击查看诊断，即可弹出诊断对话框，可查看编译过程。在诊断框中弹出编译过程已成功完成，即可表示编译成功，也会弹出代码生成报告。



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。

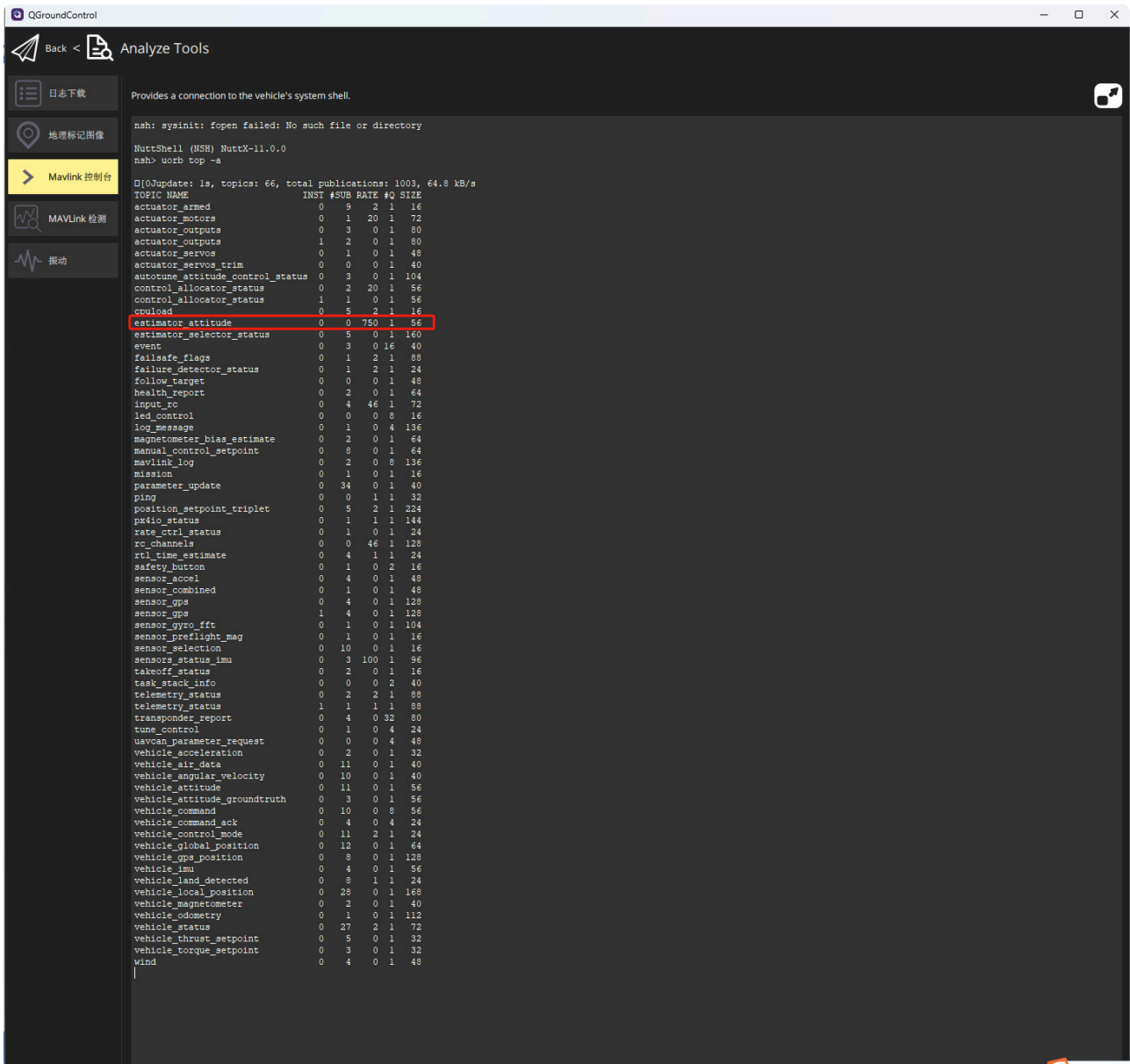


或



如下图所示，打开QGroundControl软件，点击左上角Logo在弹出的对话框中，选中Analyze Tools，选择Mavlink控制台：

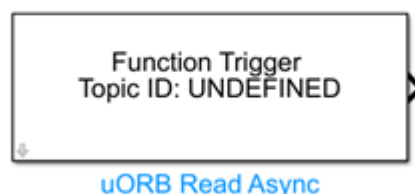
输入命令“uorb top -a”，观察消息发布情况。

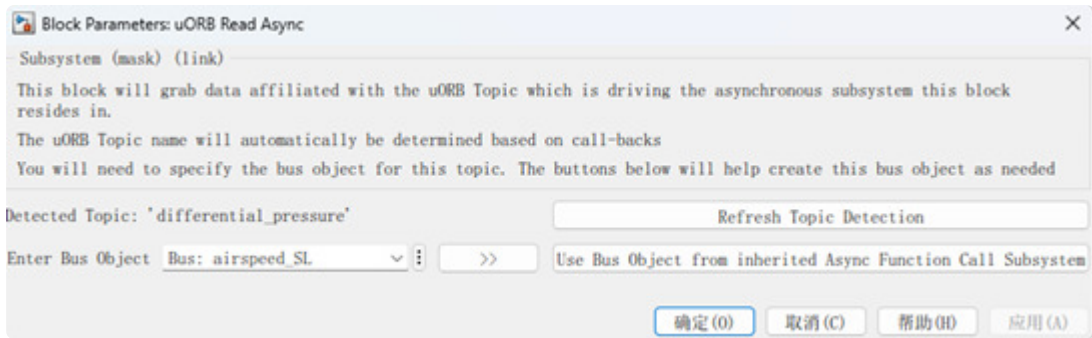


5. 关键知识点

关键知识点1: uORB Read Async—获取与uORB Topic相关的数据

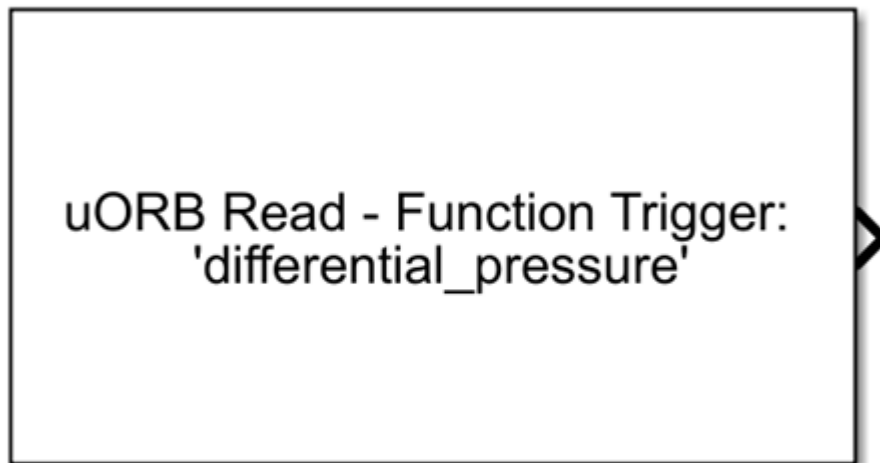
该块将获取与uORB主题相关的数据，该主题驱动该块所在的异步子系统。主题名称将根据回调自动确定，并且需要为该主题指定总线对象。更多信息可点击对话框的“帮助”按钮查看。





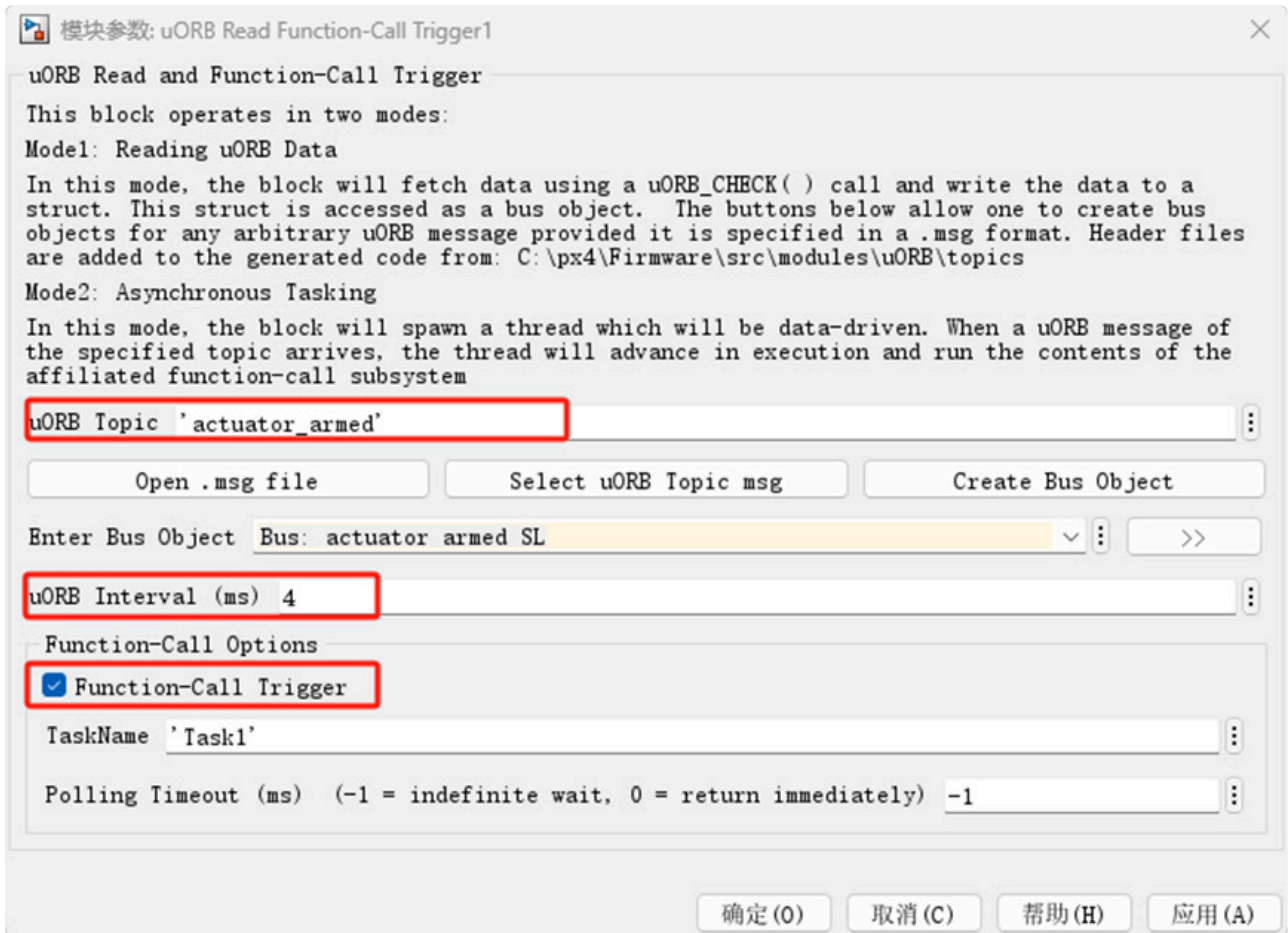
关键知识点2：uORB Read Function-CallTrigger—uORB 消息读取回调函数触发模块

这个模块提供了两个功能，第一个是从某个uORB话题订阅对应的消息。第二个是对异步事件，采用触发函数调用信号的方法订阅话题上的消息数据。



uORB Read Function-Call Trigger

第一个功能的使用步骤为：



选择一个已定义的话题

点击按钮“Select uORB Topic msg”可以打开话题列表以供选择，只支持非C++对象的话题。

创建总线（bus）对象

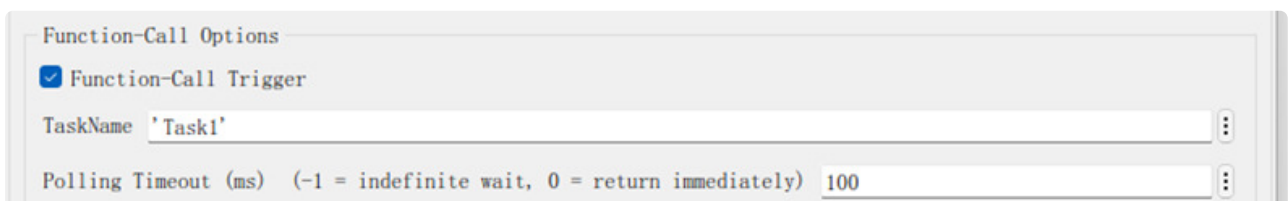
Simulink的总线对象用来接收uORB的消息，点击按钮“Create Bus Object”，Simulink将从.msg文件夹中找到对应的消息文件，并将其映射到MATLAB工作空间生成总线对象。

设置uORB读间隔

非异步模式下，需要设置查询频率，单位是毫秒，某些话题设置了最高数据更新速率，设置的频率不要超过这个最大值。

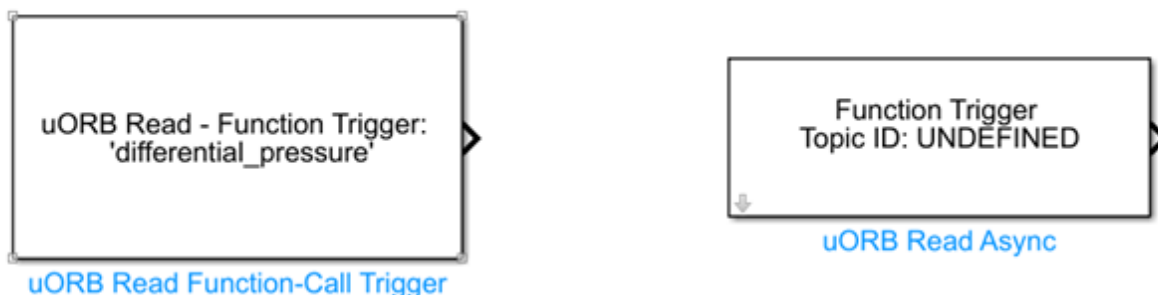
第二个功能的使用步骤为：

选择函数调用触发器



设置查询超时和任务名

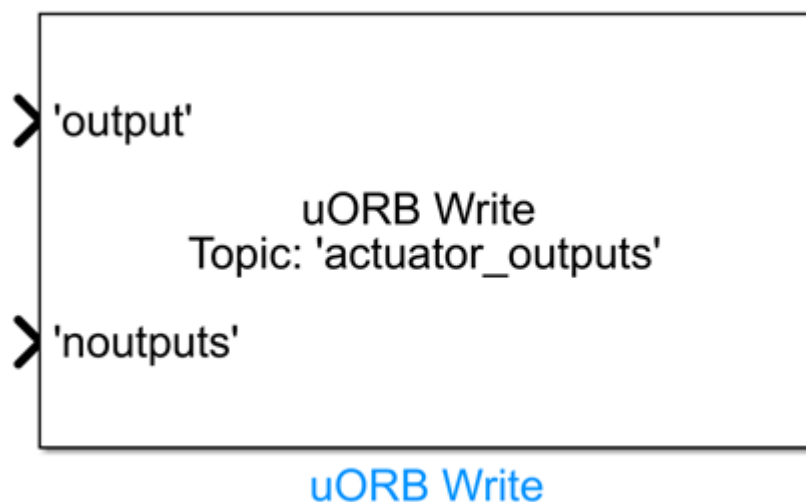
当选择了异步功能，采样时间设置框就消失了，这个时候需要设置查询超时参数和任务名。异步功能会衍生出新的线程，负责运行和函数触发信号相关的代码，它会通过查询来等待话题上新的数据的到来。此时需要另一个模块来读取话题上的数据，即“读uORB函数触发器数据模块”，如下图所示。



关键知识点3：uORB Write—uORB消息数据发布接口模块

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。

可以输入话题名，点击按钮“Open.msg file”打开对应的消息内容，点击按钮“Open.msg folder”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: uORB Write

S-Function (mask) (link)

uORB Write Output Block

Publishes to a user provided named topic structure.

uORB Topic 'actuator_outputs'

Number of Outputs 2

uORB Parameter Names and Data Type

Input Struct param :

'output' single >> Number of Elements 32

Input Struct param :

'noutputs' uint32 >> Number of Elements 1

Input Struct param :

Input Struct param :

Input Struct param :

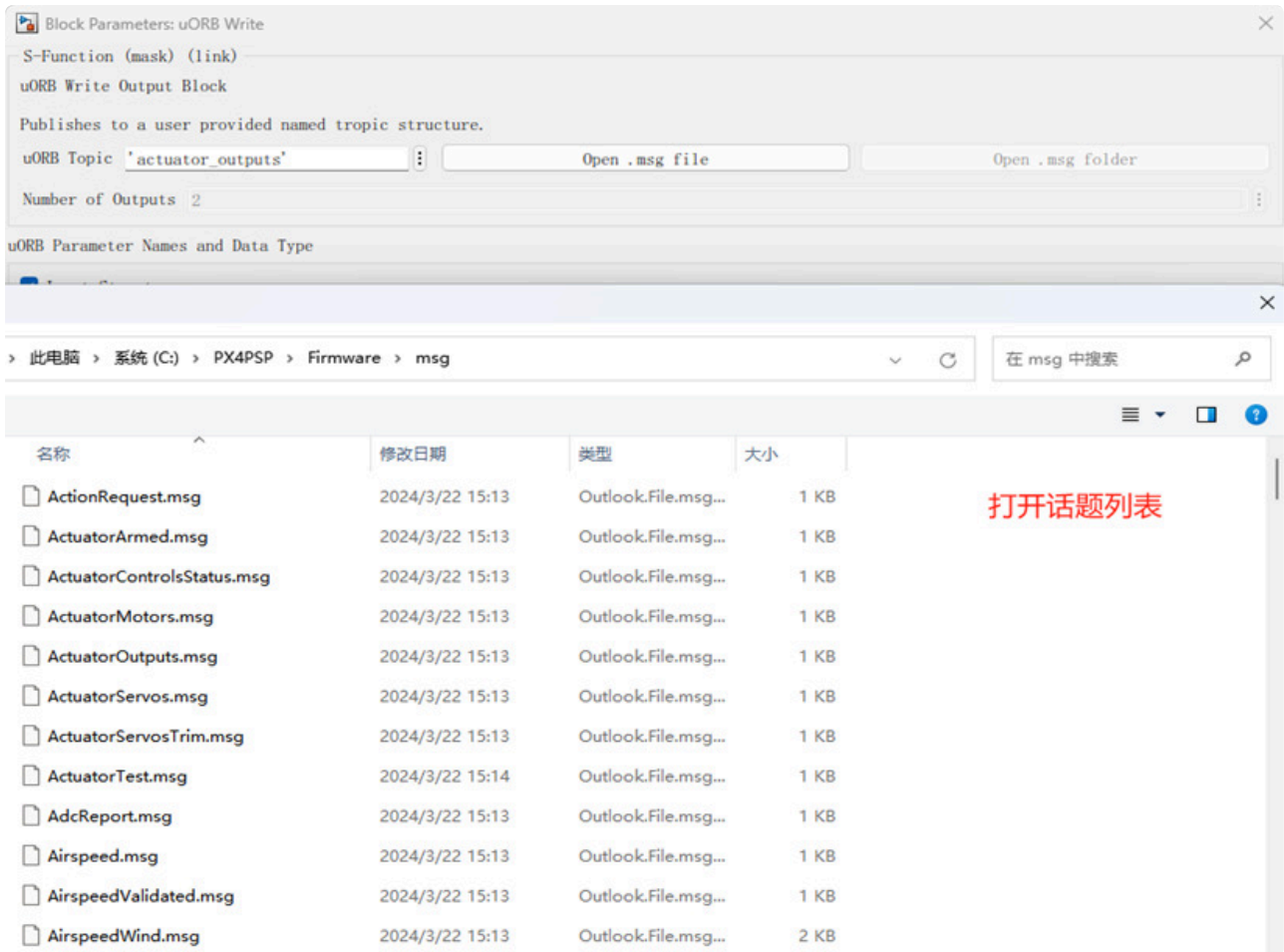
Sample Time (-1 inherited) -1

```

ActuatorOutputs.msg × +
1  uint64 timestamp           # time since system start (microseconds)
2  uint8 NUM_ACTUATOR_OUTPUTS = 16
3  uint8 NUM_ACTUATOR_OUTPUT_GROUPS = 4 # for sanity checking
4  uint32 noutputs           # valid outputs
5  float32[16] output        # output data, in natural output units
6
7  # actuator_outputs_sim is used for SITL, HITL & SIH (with an output range of [-1, 1])
8  # TOPICS actuator_outputs actuator_outputs_sim actuator_outputs_debug
9

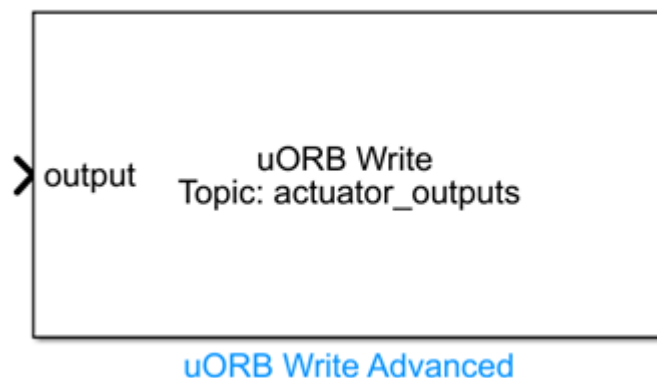
```

[查看消息内容](#)

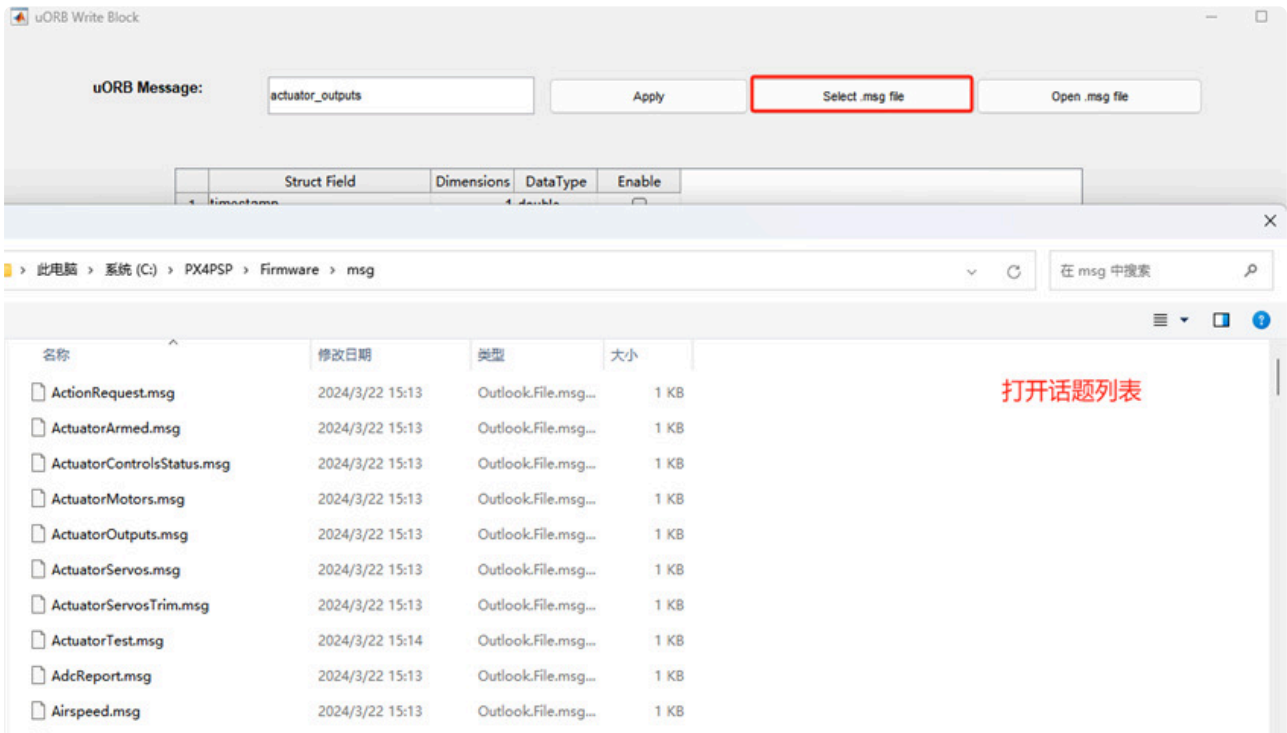
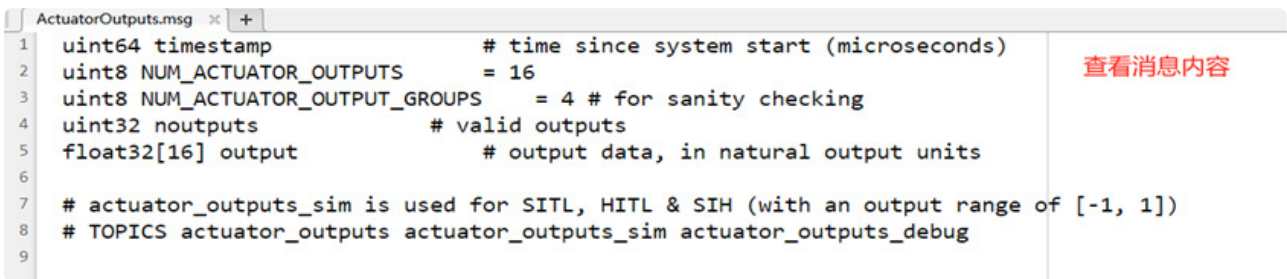
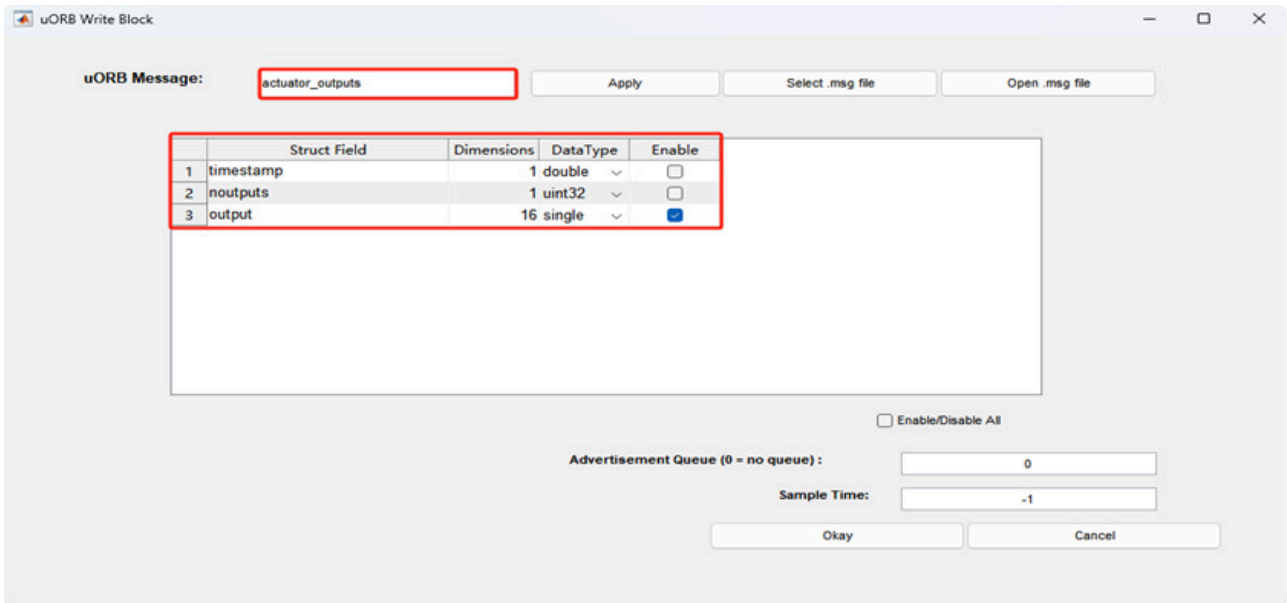


关键知识点4: uORB Write Advanced—uORB消息数据发布接口高级模块

该接口可允许用户对其发布的数据进行更灵活的控制。在Simulink中使用uORB WriteAdvanced接口，可以实现更加复杂和精确的消息发布方式。可选择要写入的消息文件和一个消息ID。此外，还可以设置消息的优先级、队列大小等高级选项。

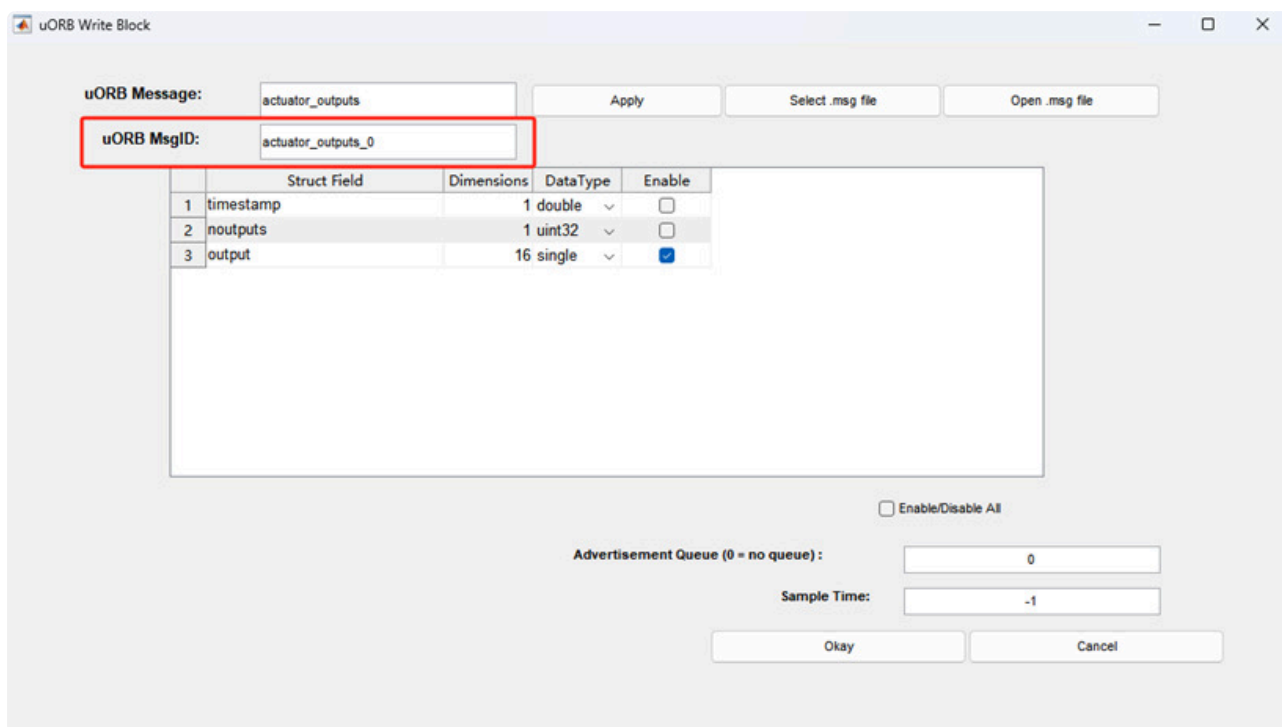
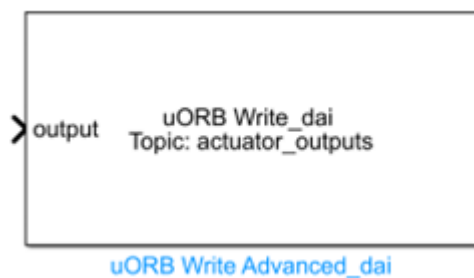


可以看到当前话题名，点击按钮“Open.msg file”打开对应的消息内容，点击按钮“Select.msg file”打开话题列表，并且可以设置输入端口名及数据类型以对应话题消息。



关键知识点5: uORB Write Advanced_dai—uORB消息数据发布接口进阶模块

相较于uORB Write Advanced, uORB Write Advanced_dai新增自定义uORBMsgID功能。



关键知识点6：其他模块

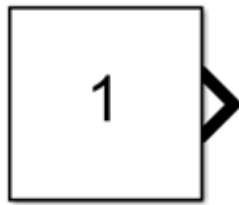
以下模块均为Simulink官方模块。

1. Data Type Conversion模块



该模块将输入转换为输出的数据类型和定标。转换有两个可能的目标。一个目标是使输入和输出的真实值相等。另一个目标是使输入和输出的存储的整数值相等。溢出和量化误差会妨碍目标的完全实现。

2. Constant模块



Constant

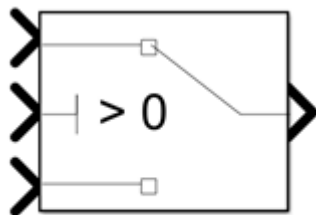
Constant模块生成实数或复数常量值信号。使用此模块提供常量信号输入。此模块是生成标量、向量还是混合输出要取决于：

l 常量值参数的维度

l 将向量参数解释为一维向量参数的设置

此模块的输出与常量值参数具有相同的维度和元素。如果为此参数指定向量，即希望模块将其解析为向量，请选中将向量参数解释为一维向量复选框。否则，如果为常量值参数指定了向量，模块会将该向量视为一个矩阵。

3. Switch模块



Switch

Switch模块根据第二个输入的值，传递第一个输入或第三个输入。第一个和第三个输入称为数据输入。第二个输入称为控制输入。可以使用首个输入传递条件和阈值参数指定模块传递第一个输入的条件。

要立即将已知的输出数据类型反向传播到第一个和第三个输入端口，将输出数据类型参数设置为继承:通过内部规则继承并选中要求所有数据端口输入具有相同的数据类型复选框。

4. Bus Selector模块



Bus
Selector

Bus Selector模块从输入总线层次结构中按名称提取您选择的元素。该模块可以分别输出选定的各个元素或在一个新的虚拟总线中输出所有选定元素。当模块分别输出选定元素时，每个选定元素对应于一个输出端口。当模块输出新的虚拟总线时，对于包含所有选定元素的虚拟总线，该模块具有一个与之对应的输出端口。

5. Clock模块



Clock模块在每个仿真时间步输出当前仿真时间。

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. [QGroundControl MAVLink控制台文档](#)
3. [PX4 控制台文档](#)
4. [RflySim 安装指南](#)
5. [RflySim 遥控器配置指南](#)

7. 常见问题

Q1: 自定义的uORB消息在导出的.ulg日志文件中无法生成日志数据

A1: 打开

“*:\PX4PSP\Firmware\src\modules\logger\logged_topics.c” 文件。将自定义的uORB消息添加到该文件中，即在第一个函数中添加代码 “add_topic("rfly_test");” 如下：

C++ logged_topics.cpp 9+, M ●

src > modules > logger > C++ logged_topics.cpp > add_default_topics()

```
37 #include <parameters/parameters.h>
38 #include <px4_platform_common/log.h>
39 #include <px4_platform_common/px4_config.h>
40 #include <uORB/topics/uORBTopics.hpp>
41
42 #include <string.h>
43
44 using namespace px4::logger;
45
46 void LoggedTopics::add_default_topics()
47 {
48     add_topic("action_request");
49     add_topic("rfly_test");
50     add_topic("actuator_armed");
51     add_topic("actuator_controls_0", 50);
52     add_topic("actuator_controls_1", 100);
53     add_topic("actuator_controls_2", 100);
54     add_topic("actuator_controls_3", 100);
```

将该文件保存后，双击打开"*\桌面\RflyTools\Win10WSL.lnk"的WSL子系统，进行编译固件编译完成后，重复Step3烧录飞控当中，即可在QGroundControl导出.ulg文件，处理后可得到自定义消息发布的数据。

1. <https://rflysim.com/> ↵
2. 推荐配置请见：<https://rflysim.com/doc/zh/HowToInstall.pdf> ↵
3. 遥控器相关配置见：<https://rflysim.com/doc/zh/B/3.1ET10.html> ↵