

1. 实验名称及目的

uORB 消息读取与写入实验：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. 实验原理

uORB（Micro Object Request Broker，微对象请求代理器）是 PX4/Pixhawk 系统中非常重要且关键的一个模块，它肩负了整个系统的数据传输任务，所有的传感器数据、GPS、P PM 信号等都要从芯片获取后通过 uORB 进行传输到各个模块进行计算处理。实际上 uORB 是一套跨「进程」的 IPC 通讯模块。在 Pixhawk 中，所有的功能被独立以进程模块为单位进行实现并工作。而进程间的数据交互就由为重要，必须要能够符合实时、有序的特点。在 PX4 中，uORB 是用于无人机模块间通信的协议机制。

飞控内部使用 NuttX 实时 ARM 系统，而 uORB 对于 NuttX 而言，它仅仅是一个普通的文件设备对象，这个设备支持 Open、Close、Read、Write、Ioctl 以及 Poll 机制。通过这些接口的实现，uORB 提供了一套“点对多”的跨进程广播通讯机制，“点”指的是通讯消息的“源”，“多”指的是一个源可以有多个用户来接收、处理。而“源”与“用户”的关系在于，源不需要去考虑用户是否可以收到某条被广播的消息或什么时候收到这条消息。它只需要单纯的把要广播的数据推送到 uORB 的消息“总线”上。对于用户而言，源推送了多少次的消息也不重要，重要的是取回最新的这条消息。也就是说在通讯过程中发送者只负责发送数据，而并不关心数据由谁接收，也不关心接收者是否能将所有的数据都接收到；而对于接收者来说并不关心数据是由谁发送的，也不关心在接收过程中是否将所有数据都接收到。

3. 实验效果

实现 uORB 消息的写入与读取。

4. 文件目录

文件夹/文件名称	说明
px4demo_uORB_read_write.slx	飞控飞行日志文件。
PX4ExtMsgSender.slx	飞控 uORB 消息监听程序

5. 运行环境

序号	软件要求	硬件要求	
		名称	数量
1	Windows 10 及以上版本	笔记本/台式电脑 ^①	1
2	RflySim 平台免费版及以上版本	Pixhawk 6C 或 Pixhawk 6C mini ^②	1

		遥控器 ^③	1
		遥控器接收器	1
		数据线、杜邦线等	若干

①：推荐配置请见：<https://doc.rflysim.com>

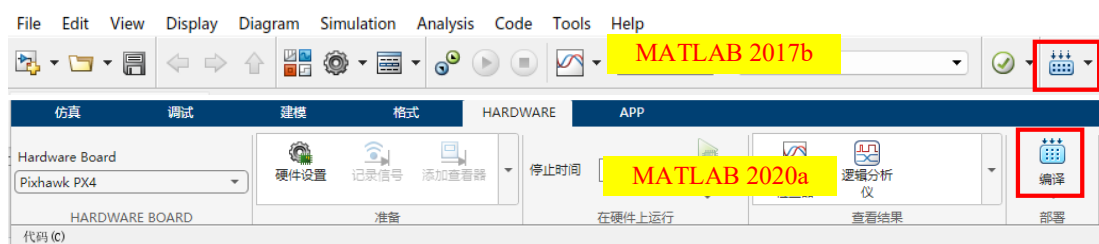
②：须保证平台安装时的编译命令为：px4_fmu-v6c_default，固件版本为：1.13.3。其他配套飞控请见：<http://doc.rflysim.com>

③：本实验演示所使用的遥控器为：天地飞 ET10、配套接收器为：WFLY RF209S。遥控器相关配置见：..\e11_RC-Config\Readme.pdf

6. 实验步骤

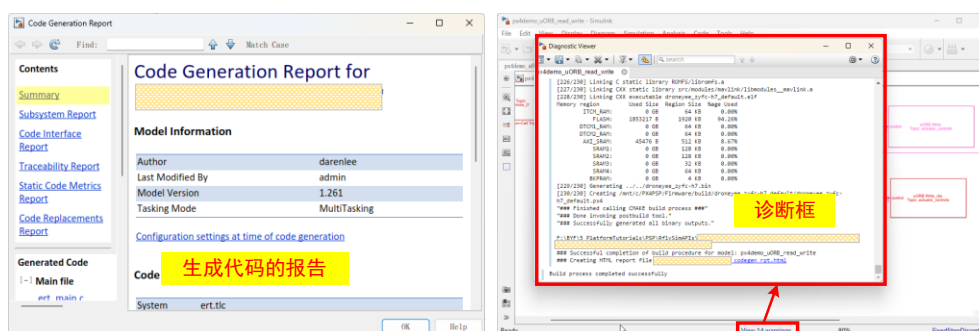
Step 1:

打开 MATLAB 软件，在 MATLAB 中打开 px4demo_uORB_read_write.slx 文件，在 Simulink 中，点击编译命令。



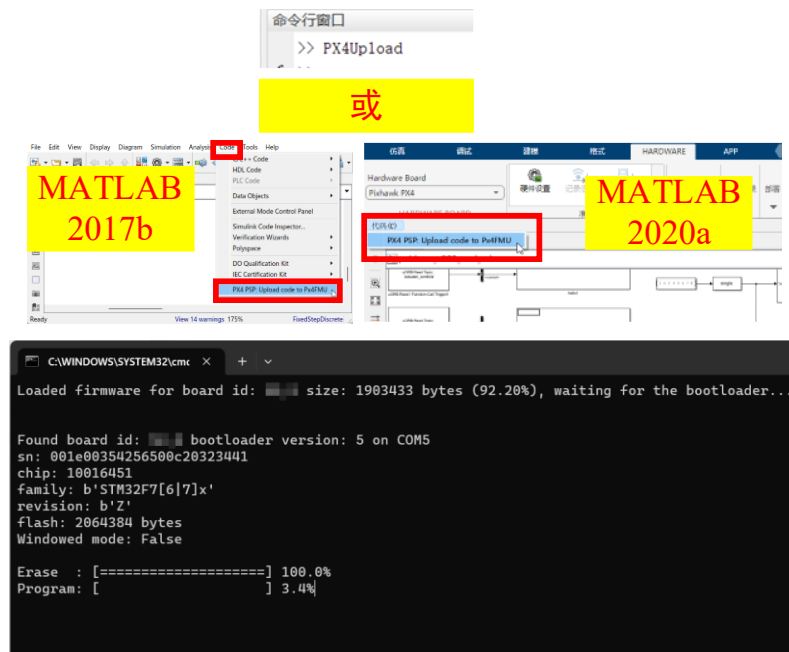
Step 2:

在 Simulink 的下方点击 View diagnostics 指令，即可弹出诊断对话框，可查看编译过程。在诊断框中弹出 Build process completed successfully，即可表示编译成功，左图为生成的编译报告。



Step 3:

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。

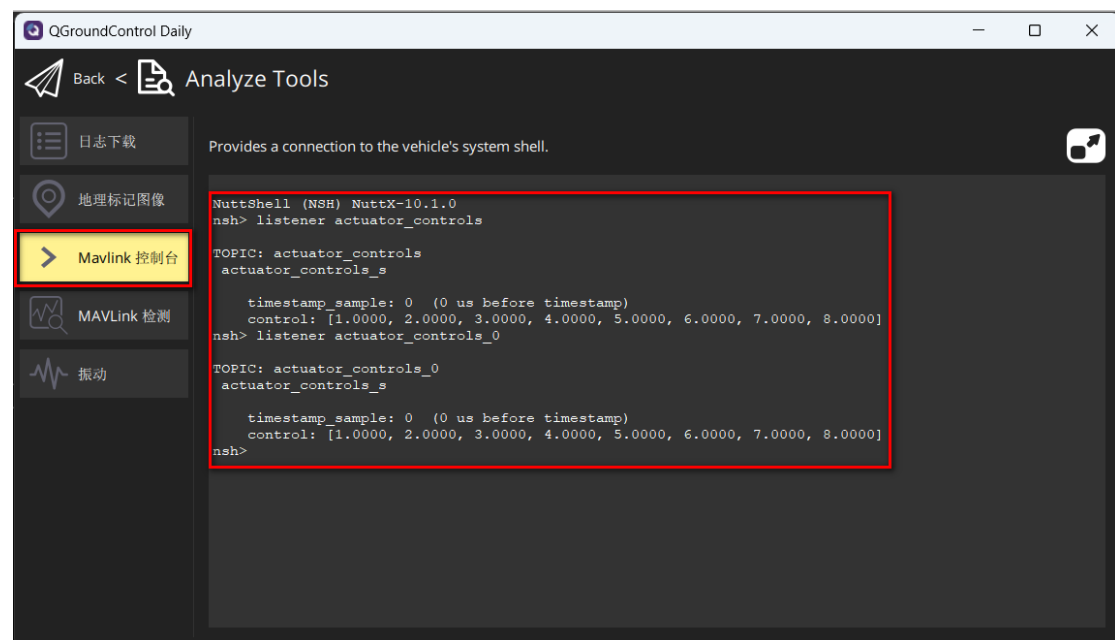


Step 4:

打开 QGroundControl 软件，点击左上角 Logo 在弹出的对话框中，选中 Analyze Tools，在 Mavlink 控制台中输入：

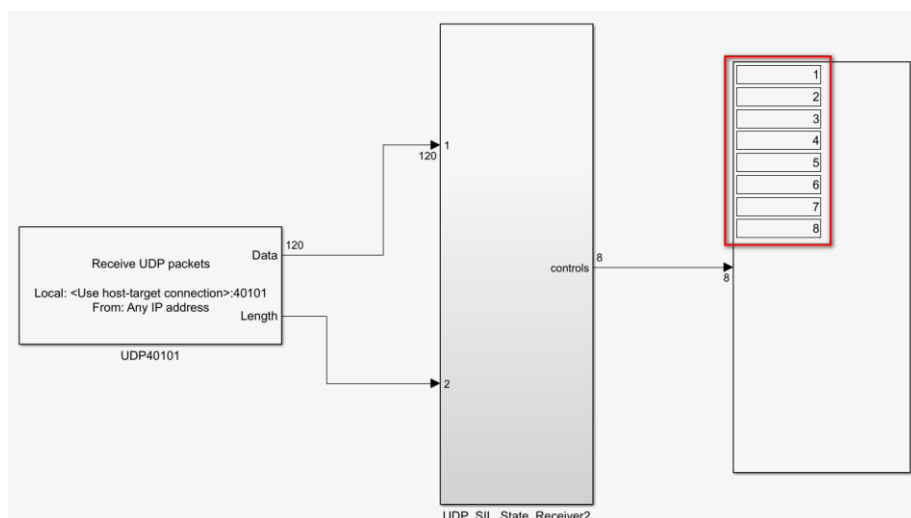
```
listener actuator_controls
```

即可得出如下图的结果。



Step 5

在进行硬件在环仿真时，在 MATLAB 中打开 PX4ExtMsgSender.slx 监听程序，点击运行即可监听到写入的 uORB 消息，如下图所示：



7. 参考资料

[1]. 暂无

8. 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/uORBTTopics.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。