

PX4 学习笔记之 uorb

——添加 **topic** 以及消息的发布、订阅

说明

添加自定义 **topic**，简单例子实现 new **topic** 的 **publish** 和 **subscribe**

平台

Pixhawk、PX4 原生固件 **Firmware**

要求

学会 **px4_simple_app** 例程，了解 **uorb** 通讯机制

本文档讲详细介绍如何在 **Firmware** 原生代码添加我们自己的 **topic**，然后如何进行消息的发布和订阅。

添加自定义 **topic**

首先在 **msg/** 文件夹下添加后缀名 **.msg** 的文件，文件名要和你将要添加的 **topic** 名称一样。可以参考其他现有的文件。例如，要声明的 **topic** 为 **myuorb_test**，那文件名就应该为 **myuorb_test.msg**，在文件里声明你的存放数据的结构体成员变量（注意：仅仅是成员变量。如结构体为 **struct myuorb_test_s{int r};**，那么 **.msg** 文件中就写 **int r** 就可以了）。

然后在该文件夹下的 **CMakeLists.txt** 里注册新添加的 **.msg** 文件。注册后，在编译固件时会自动生成 **myuorb_test.h** 文件，该文件里定义了数据结构体 **myuorb_test_s** 及 **ORB_DECLARE**。

最后在需要用到 **topic** 的地方包含相应头文件就行了。**#include <uORB/topics/myuorb_test.h>**。

这样新的 **topic** 就添加成功了。

订阅、发布消息

如果了解了 **uorb** 的话，这部分应该比较简单。在此之前必须学会写 **PX4** 应用程序，如果不会可以先学习 **px4_simple_app** 例程。下面直接给出我的测试程序。

myuorb_test.c

```
//include head files
#include <px4_config.h>
#include <px4_tasks.h>
#include <px4_posix.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <poll.h>
#include <string.h>
```

```

#include <uORB/uORB.h>
#include <uORB/topics/myuorb_test.h>

//ORB_DEFINE(myuorb_test,struct myuorb_test_s);

__EXPORT int myuorb_test_publisher_main(int argc, char *argv[]);

static orb_advert_t topic_handle;
int topics_handle;

int myuorb_test_publisher_main(int argc, char *argv[])
{
    PX4_INFO("uorb publishint test");
    topics_handle = orb_subscribe(ORB_ID(myuorb_test));
    //publish data
    //update data
    struct myuorb_test_s rd = {.r=rand()%10000};
    //advertise the topic
    topic_handle = orb_advertise(ORB_ID(myuorb_test),&rd);
    //publish data,update the topic
    orb_publish(ORB_ID(myuorb_test),topic_handle,&rd);
    PX4_WARN("the new data is: %t%d",rd.r);

    bool updated;
    struct myuorb_test_s rds;
    //subscribe a topic
    //check to see whether the topic has updated since the last
//time we read it
    orb_check(topics_handle,&updated);
    if(updated)
    {
        //make a local copy of the updated data structure
        orb_copy(ORB_ID(myuorb_test),topics_handle,&rds);
        PX4_WARN("the updated data is: %t%d",rds.r);
    }
    else
    {
        PX4_WARN("data is not updated");
    }
    return 0;
}

```

其中需要注意的是topics_handle = orb_subscribe(ORB_ID(myuorb_test)) 这个订阅消息的函数一定要写在前面，不然订阅不成功。其他部分注释也比较详细，就不再详细说了。

测试

使用 make px4fmu-v2_default 命令生成固件，使用 make px4fmu-v2_default uoload 命令上传固件到飞控。

启动系统终端（快捷键 Ctrl+Alt+T），输入 screen /dev/ttyACM0 57600 8N1

连接飞控。也可以使用 Qgroundcontrol 地面站的 terminal 终端，不过此功能新版本没添加，需要使用旧版本（如 v2.2）。注意：连接终端必须取下 SD 卡，否则电脑会接收到许多乱码的数据导致终端崩溃。

在 NSH 终端输入 myuorb_test，如果程序没有问题，则会出现以下信息：

```
zhuwei@zhuwei-Inspiron-5547: ~/PX4/Firmware
nsh> myuorb_test_publisher
INFO uorb publishint test
WARN the new data is: 0
WARN the updated data is: 0
nsh> myuorb_test_publisher
INFO uorb publishint test
WARN the new data is: 5407
WARN the updated data is: 5407
nsh> 
```

这样就成功地发布以及订阅了消息。

官方链接: <http://dev.px4.io/advanced-uorb.html>
<http://dev.px4.io/tutorial-hello-sky.html>
<http://dev.px4.io/advanced-system-console.html>

由于本人也是新手，如果有什么写的不正确的地方，欢迎大家提出一起交流心得。

QQ: 1322901615

Email: 1322901615@qq.com