# PX4源码修改记录

Firmware/msg

1. 添加uorb消息 fault\_injection.msg
2. Cmakelists.txt 添加编译规则

Firmware/src/modelus

1. 添加了三个文件 .cpp、.h、Cmakelists.txt

Firmware/cmake/configs

1. nuttx\_px4fmu-v3\_default.cmake添加编译规则modules/fault\_injection

电机故障注入（1,2,3,4）

Firmware/src/drivers/px4io/px4io.cpp

1. 87行头文件

#include <uORB/topics/fault\_injection.h>

1. 264行变量定义

int \_t\_fault\_injection; ///< fault\_injection topic

1. 894行 orb\_subscribe

\_t\_fault\_injection = orb\_subscribe(ORB\_ID(fault\_injection));

1. 1278行 orb\_copy和数据处理

struct fault\_injection\_s fault\_injection\_topic;

orb\_copy(ORB\_ID(fault\_injection), \_t\_fault\_injection, &fault\_injection\_topic);

**......**

1. 堆栈大小增加了几百

accelerometer（5）、gyroscope（6）、magnetometer（7）、baro（8）故障注入

Firmware/src/modelus/sensors/ sensors.cpp

1. 头文件

#include <uORB/topics/fault\_injection.h>

1. 变量定义

int \_t\_fault\_injection; ///< fault\_injection topic

1. orb\_subscribe

\_t\_fault\_injection = orb\_subscribe(ORB\_ID(fault\_injection));

1. orb\_copy和数据处理

struct fault\_injection\_s fault\_injection\_topic;

orb\_copy(ORB\_ID(fault\_injection), \_t\_fault\_injection, &fault\_injection\_topic);

**......**

gps（9）故障注入

Firmware/src/drivers/gps/gps.cpp

同上基本的四步

Firmware/platforms/nuttx/nuttx-configs/px4fmu-v2/nsh/defconfig

修改 CONFIG\_USERMAIN\_STACKSIZE = 2600 （原为2500，需要全部重新编译）

修改了dji\_4011\_f450里面的参数

# 坑

1. Px4io的程序和px4fum的程序共享了lib库，但两者间无法直接跳转，要分别建立工程。
2. Px4iofirmware/mixer.cpp中的r\_page\_actuators[]变量直接链接到了pwm输出，直接修改其值即(1000-2000 or any)可改变输出pwm。
3. Mixer由于加入了想象中的非线性电机模型以及很多饱和操作，使得理论上输入actuator\_controls\_0和输出actuator\_outputs之间变得并不线性。但实际上\_thrust\_factor=0.0f，所以只要输出不饱和，还是线性的，线性变换关系为：

*max = 1950;*

*min = 900;*

*G = [-1,1,1,-1;1,-1,1,-1;1,1,-1,-1;1,1,1,1]';*

*input = [0;0;0;油门];*

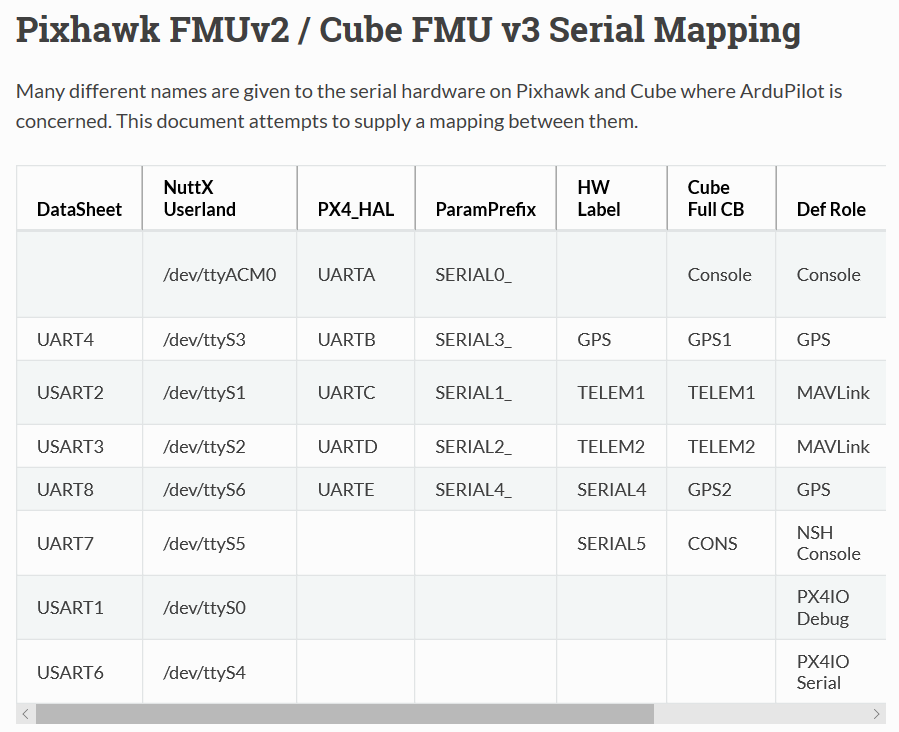
*input = (input - [0;0;0;0.5]) .\*[1.414;1.414;2;2];*

*output = G\*i;*

*pwm\_o = output \*(max-min)/2 + (max+min)/2;*

*disp(pwm\_o')*

加红部分定义于/home/myros/PX4learn/Firmware/build/nuttx\_px4io-v2\_default/src/lib/mixer的175行\_config\_quad\_x \* 2

1. /home/myros/PX4learn/Firmware/ROMFS/px4fmu\_common/init.d/4011\_dji\_f450 文件夹下全部是各个机型的参数配置
2. /home/myros/PX4learn/Firmware/build/nuttx\_px4io-v2\_default/src/lib/mixer 这个文件夹下保存了各个机型的mixer
3. 手写MAVLINK协议时顺序一定要看源码定义的载荷顺序顺序来写，源码顺序和官网顺序不对应！
4. 手写MAVLINK协议时注意在crc\_check中加入每个消息对应的不同特征值
5. 

<http://ardupilot.org/copter/docs/common-pixhawk-serial-names.html> 硬件导览

1. PX4IO的debug方法：使用debug(args\*)打印debug信息，打印的信息通过cat /dev/ttyS0显示
2. 用数传远程连接QGC打不开端口时，把Application Setting -> General 中的AutoConnect全部关掉即可。
3. Matlab的mavlink的每个消息，赋值给Payload结构体的每个参数全部要加一个强制类型转换，如*msg.Payload.ind = uint8(2); msg.Payload.time\_boot\_ms = uint32(1000); msg.Payload.value = single(123)* 等等
4. 要用CP2102的USB转串口模块，便宜的CH340在115200下连10次才能连上一次，921600下往往是乱码，很不稳定。
5. Mavlink消息长度是可变的，对同一个消息，根据内部零值的数量会自动改变长度，从而减少发送的数据量。因此不要尝试自己写解析函数！

# PIXHAWK参数修改

CBRK\_IO\_SAFETY = 22027 //失能安全开关

CBRK\_USB\_CHK = 197848 //连接USB的情况下也可解锁

SYS\_COMPANION = Companion Link(921600 baud, 8N1) //波特率改为921600，注意：虽然参数介绍和源码下都写了TELEM2，但具体飞控板上不一定是标了TEMEL2的口生效，需要测试。

# 南航项目的程序文档内容

1. 整个系统框图，包括软件和硬件