

1. 实验名称及目的

1.1. 实验名称

底层飞控开发 PX4 模块替换实验

1.2. 实验目的

因 Simulink 控制器模块与 PX4 内部子模块是相互独立，并行运行的。因此，在本实验中只需要将 PX4 模块的输出消息屏蔽掉，用 Simulink 控制器发送该消息，就能实现模块的替换。

1.3. 关键知识点

- Simulink 控制器模块。
- PX4 版本、硬件平台和地面站。

2. 实验效果

手动成功屏蔽了 uORB 消息：actuators_0，并将 Simulink 的控制器模型烧录飞控只可实现正常硬件在环仿真。

3. 文件目录

文件夹/文件名称		说明
icon	Init.m	模型初始化参数文件。
	MavLinkStruct.mat	MAVLink 结构体数据文件。
	pixhawk.png	Pixhawk 硬件图片。
Exp6_ReplacePX4AttitudeCtrler.slx		改造后的姿态控制模型。
Init_control.m		控制器初始化参数文件。

4. 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 工具链	Pixhawk 6x 飞控 ^②	1
3	MATLAB 2022b 及以上	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_fmuv6x_default，固件版本为：1.12.3。其他配套飞控请见：<http://rflysim.com/hardware.html>

③：本实验演示所使用的遥控器为：天地飞 WFLY-ET10、配套接收器为：WFLY-RF20 9S。遥控器相关配置见：<https://rflysim.com/doc/zh/B/3.1ET10.html>

5. 实验步骤

5.1. PX4 模块替换

打开 MATLAB 软件，在 MATLAB 中打开 RflySim 安装文件夹，运行 OnekeyScript.p 一键安装脚本，在弹出的一键安装脚本对话框中，进行如下设置。设置完成后，点击确认等待安装完成。

步骤	配置项	选择
1.	工具包安装路径	C:\PX4PSP
2.	PX4固件编译命令: 见Firmware\boards目录, 模版px4_fmu-v6x_default, droneeye_racer_default等	droneeye_zyfc-h7_default
3.	PX4固件版本 (1: PX4-1.7.3, 4: PX4-1.10.2, 5: PX4-1.11.3, 6: PX4-1.12.3, 7: PX4-1.13.3)	6
4.	PX4固件编译器 (1: Win10WSL[通用], 2: Msys2[适用版本≤PX4-1.8], 3: Cygwin[适用≥PX4-1.8])	1
5.	是否重新安装PSP工具箱(是: 重装工具箱, 否: 维持现有安装)	否
6.	是否重新安装其他依赖程序包 (CopterSim、QGC地面站、硬件在环仿真软件等, 约5分钟)	否
7.	是否重新配置固件编译器编译环境 (是: 全新安装编译器, 否: 维持原样, 重装约5分钟)	是
8.	是否重新部署PX4固件代码 (是: 全新部署代码, 否: 维持现状, 大约5分钟)	是
9.	是否预先用选定命令编译固件 (是: 全新编译固件, 否: 维持现状, 大约5分钟)	是
10.	是否屏蔽PX4官方控制器输出(使用Simulink控制器选"是", 使用PX4官方控制器选"否")	否

使用 VScode 打开"*PX4PSP\Firmware\src\modules\mc_rate_control\MulticopterRateControl.cpp"文件，在文件搜索“_actuators_0_pub.publish(actuators);”，可以看出文件中有两处该程序段。

```
C:\PX4PSP > Firmware > src > modules > mc_rate_control > C++ MulticopterRateControl.cpp > Run()
251
252     if (_battery_status_scale > 0.0f) {
253         for (int i = 0; i < 4; i++) {
254             actuators.control[i] *= _battery_status_scale;
255         }
256     }
257
258
259     actuators.timestamp = hrt_absolute_time();
260     _actuators_0_pub.publish(actuators);
261
262
263 } else if (_v_control_mode.flag_control_termination_enabled) {
264     if (!_vehicle_status.is_vtol) {
265         // publish actuator controls
266         actuator_controls_s actuators();
267         actuators.timestamp = hrt_absolute_time();
268         _actuators_0_pub.publish(actuators);
269     }
270 }
271
272 }
273
274 perf_end(_loop_perf);
275
276 }
277
278 int MulticopterRateControl::task_spawn(int argc, char *argv[])
279 {
280     bool vtol = false;
281
282     if (argc > 1) {
283         if (strcmp(argv[1], "vtol") == 0) {
```

将这两处进行修改，修改之后的程序如下图，修改完成之后点击保存。

```
251
252     if (_battery_status_scale > 0.0f) {
253         for (int i = 0; i < 4; i++) {
254             actuators.control[i] *= _battery_status_scale;
255         }
256     }
257
258
259     actuators.timestamp = hrt_absolute_time();
260     // _actuators_0_pub.publish(actuators);
261     UNUSED(actuators);
262
263
264 } else if (_v_control_mode.flag_control_termination_enabled) {
265     if (!_vehicle_status.is_vtol) {
266         // publish actuator controls
267         actuator_controls_s actuators();
268         actuators.timestamp = hrt_absolute_time();
269         // _actuators_0_pub.publish(actuators);
270         UNUSED(actuators);
271     }
272 }
273
274 }
275
276 }
277
278 perf_end(_loop_perf);
279
280 }
281
282 int MulticopterRateControl::task_spawn(int argc, char *argv[])
283 {
284     bool vtol = false;
285
286     if (argc > 1) {
287         if (strcmp(argv[1], "vtol") == 0) {
```

此操作是为了能够替换掉飞控的姿态控制器（手动模式），屏蔽掉原来PX4姿态控制器mc_rate_control的最关键输出uORB消息，也就是“actuator_controls_0”。但是，由于上述代码中涉及到了actuators变量，直接注释可能导致actuators变量定义了但是未被使用，在PX4严格的代码检查模式，这种不规范的编程行为（定义变量但是未使用，这段代码就没有意义）将会视为错误，导致编译不通过。因此，我们添加UNUSED宏来标注未使用的变量，避免编译器报错。

打开桌面"%桌面\RflyTools\Win10WSL.lnk"的WSL子系统快捷方式，输入：

```
make droneyee_zyfc-h7_default
```

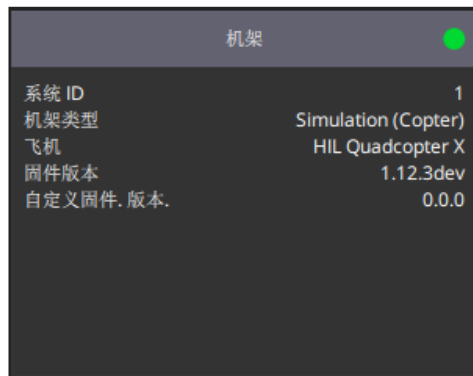
等待编译成功，如下图：

```
root@RFLYSIM: /mnt/c/PX4PSP
root@RFLYSIM:/mnt/c/PX4PSP/Firmware# make droneeye_zyfc-h7_default
[2/8] Performing build step for 'bootloader_firmware'
ninja: no work to do.
[6/8] Linking CXX executable droneeye_zyfc-h7_default.elf
Memory region      Used Size  Region Size  %age Used
ITCM_RAM:          0 GB      64 KB        0.00%
FLASH:            1789792 B    1792 KB      97.54%
DTCM1_RAM:        0 GB      64 KB        0.00%
DTCM2_RAM:        0 GB      64 KB        0.00%
AXI_SRAM:         42080 B     512 KB       8.03%
SRAM1:            0 GB      128 KB       0.00%
SRAM2:            0 GB      128 KB       0.00%
SRAM3:            0 GB      32 KB        0.00%
SRAM4:            0 GB      64 KB        0.00%
BKPRAM:           0 GB      4 KB         0.00%
[8/8] Creating /mnt/c/PX4PSP/Firmware/build/droneeye_zyfc-h7_default/droneeye_zyfc-h7_default.px4
root@RFLYSIM:/mnt/c/PX4PSP/Firmware#
```

打开 QGC 烧录上一步编译的固件，此固件位置在：“*\PX4PSP\Firmware\build\droneeye_zyfc-h7_default\droneeye_zyfc-h7_default.px4”，具体烧录步骤请查看视频：https://www.bilibili.com/video/BV1sa4y1V7hv/?spm_id_from=333.999.0.0&vd_source=1654a620e9867b8f22757a07c243c61d，或扫码观看。



上传成功后，在 QGC 中校准遥控器并设置飞行模式，打开 QGroundControl 软件。确认无人机机架及遥控器通设置如下：



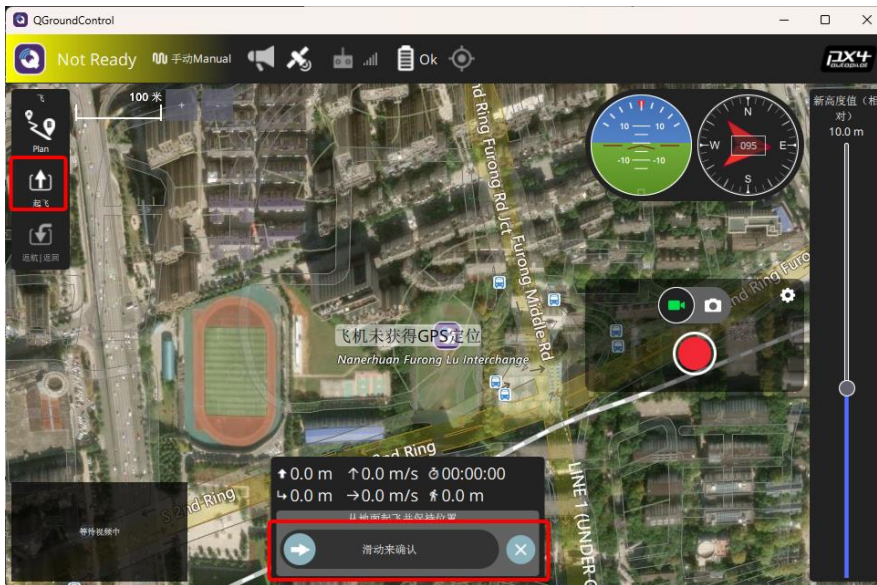
完成后双击打开“*\桌面\RflyTools\HITLRun.lnk”或“*\PX4PSP\RflySimAPIs\HITLRun.bat”文件，在弹出的 CMD 对话框中输入插入的飞控 Com 端口号，即可自动启动 RflySim3D、CopterSim、QGroundControl 软件，等待 CopterSim 的状态框中显示：PX4: GPS 3D fixed & EKF initialization finished。

```

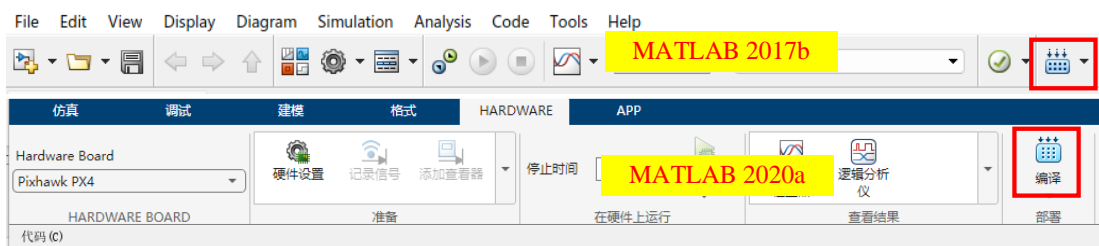
PX4: Init MAVLink
CopterSim: CopterID is 1, PX4 SysID is 1
PX4: Awaiting GPS/EKF fixed for Position control...
PX4: Enter Manual Mode!
PX4: Found firmware version: 1.12.3dev
PX4: Command ARM/DISARM ACCEPTED
PX4: Command REQUEST_AUTOPILOT_VERSION ACCEPTED
PX4: EKF2 Estimator start initializing...
PX4: GPS 3D fixed & EKF initialization finished.

```

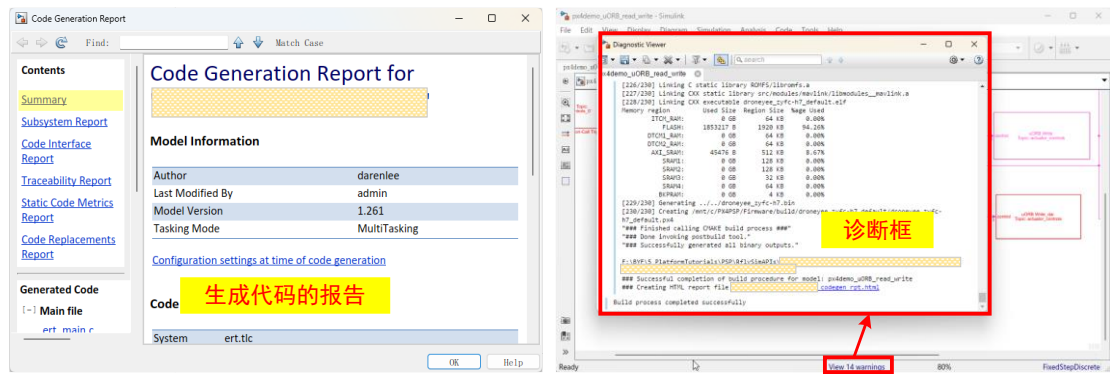
通过 QGC 软件进行起飞，如果不能起飞，说明输出接口屏蔽成功。或者通过 QGC 的 MAVLink Inspector 中观察返回的数据，来看是否有姿态控制数据发出（这种方法适用于其他模块的替换）。



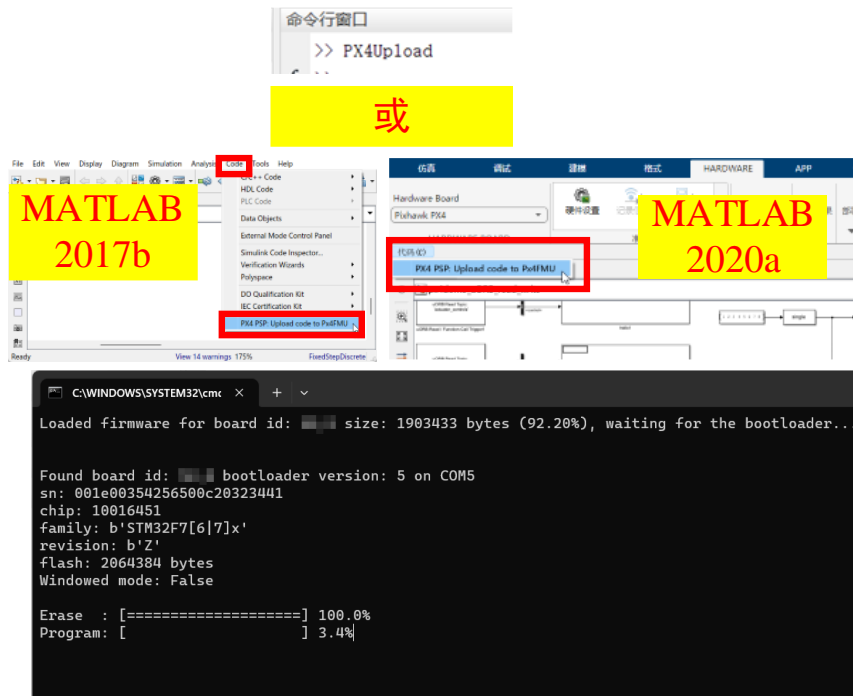
在 MATLAB 中运行 Init_control.m 文件，将自动打开 Exp6_ReplacePX4AttitudeCtrler.slx 文件，在 Simulink 中，点击编译命令。



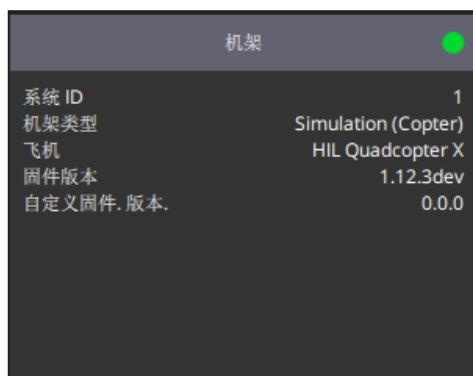
在 Simulink 的下方点击 View diagnostics 指令，即可弹出诊断对话框，可查看编译过程。在诊断框中弹出 Build process completed successfully，即可表示编译成功，左图为生成的编译报告。



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。或查看视频教程：https://www.bilibili.com/video/BV1sa4y1V7hv/?spm_id_from=333.999.0.0&vd_source=1654a620e9867b8f22757a07c243c61d。



上传成功后，打开 QGroundControl 软件。确认无人机机架及遥控器通设置如下：



重复步骤 Step5、Step6，使用 QGC 解锁，通过遥控器控制飞机。飞机能够起飞，并且能够控制姿态（应该不如原生的姿态控制效果好，特别是偏航通道）。这里我们用 Simuli

nk 编写控制器，订阅角速度 `vehicle_angular_velocity`、角度 `vehicle_attitude` 数据、和遥控输入数据 `manual_control_setpoint`，实现姿态控制器，再发送“`actuator_controls_0`”消息，从而实现姿态控制器的替换。

注：由于我们这里只修改了遥控器输入下的姿态控制，因此需要用遥控器，将飞控模式切换到自稳模式（CH5 或 CH6 通道），然后用遥控器解锁起飞来测试。

开发完成后，请务必将修改的代码回归原位，以免影响其他功能的开发。可以重新运行安装脚本，使用如下配置进行固件还原。



6. 参考资料

- [1]. 全权,杜光勋,赵峙尧,戴训华,任锦瑞,邓恒译.多旋翼飞行器设计与控制[M],电子工业出版社,2018.
- [2]. 全权,戴训华,王帅.多旋翼飞行器设计与控制实践[M],电子工业出版社,2020.

7. 常见问题

Q1: ***

A1: ***