



多旋翼飞行器设计与控制 实践

第三讲 实验平台使用

戴训华 副教授
中南大学 计算机学院

dai.xh@csu.edu.cn

https://faculty.csu.edu.cn/daixunhua/zh_CN



大纲

1. 实验平台总体介绍
2. 控制器设计与仿真平台
3. PSP工具箱
4. Pixhawk硬件系统
5. 硬件在环多旋翼飞行器仿真器
6. 小结



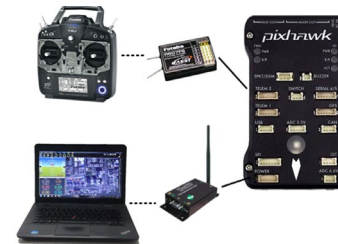
平台总体介绍

平台组成

基于Simulink的控制器设计与仿真平台



Pixhawk 自驾仪系统



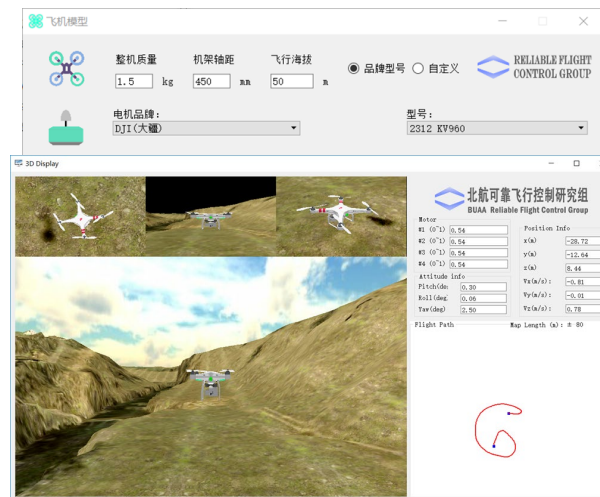
实验指导包



多旋翼硬件系统



硬件在环仿真平台





平台总体介绍

平台组成

(1) 基于Simulink的控制器设计与仿真平台

控制器设计与仿真



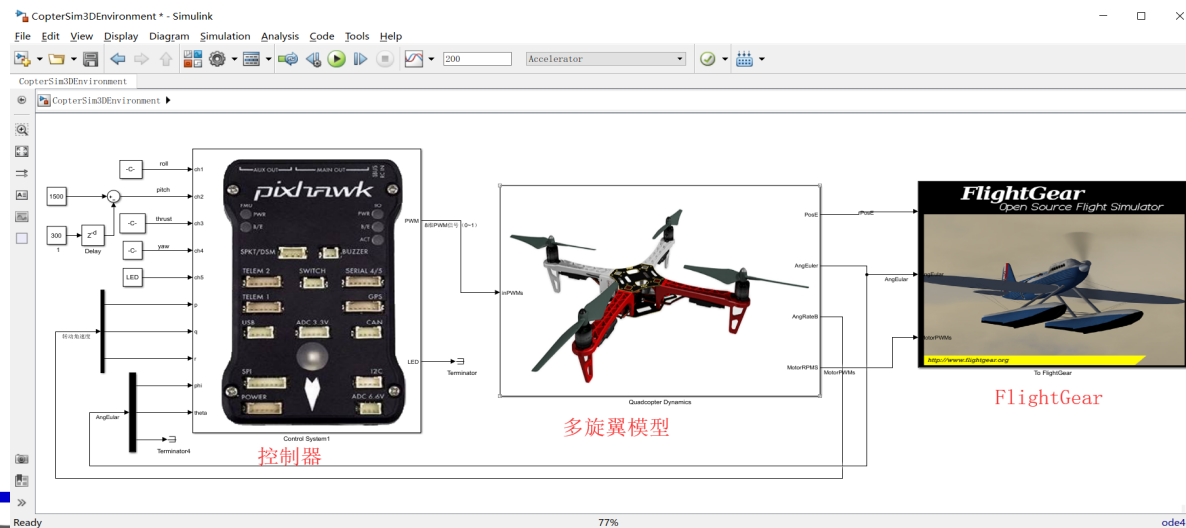
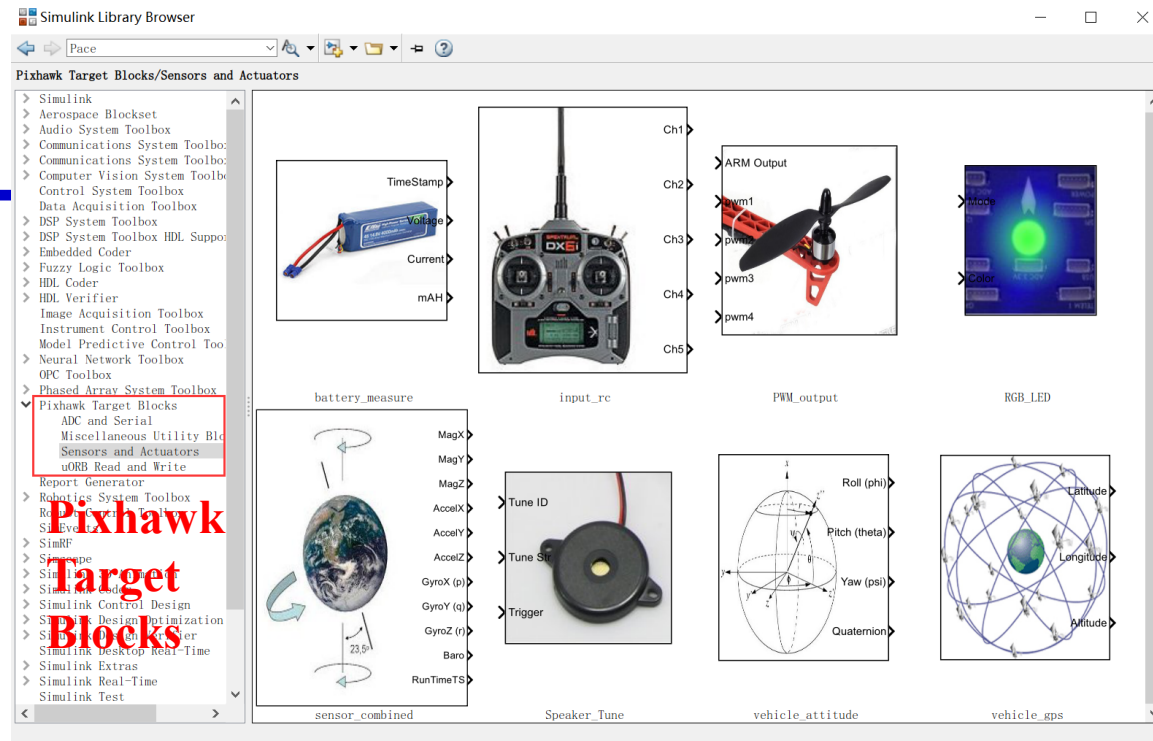
代码自动生成与固件编译



代码阅读



- 多旋翼高逼真非线性模型
- 模块化控制器设计
- FlightGear飞行显示
- 自动代码生成与上传功能





平台总体介绍

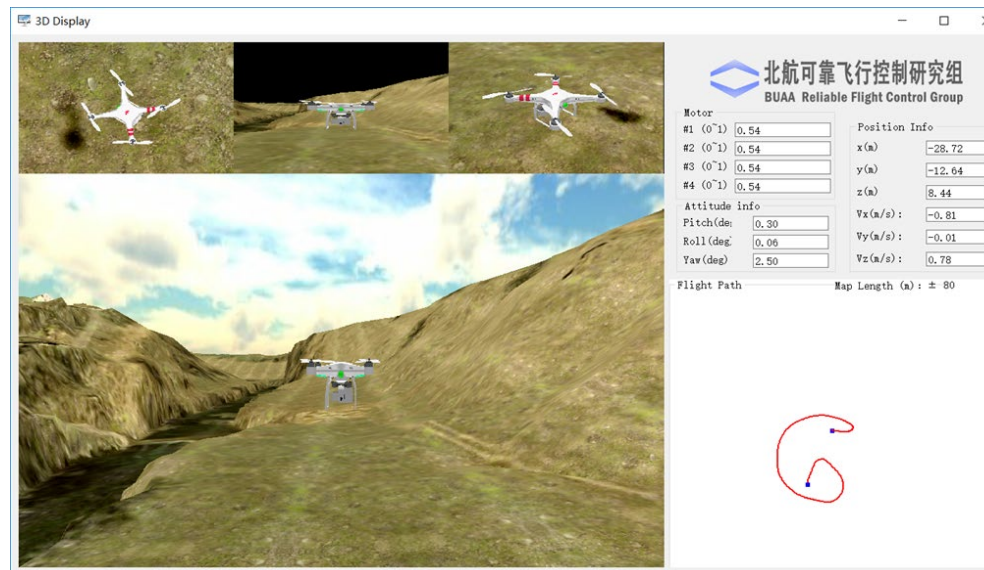
平台组成

(2) 硬件在环仿真平台

处理器在环仿真



(1) 多旋翼模型配置与运动模拟器



(2) 多旋翼三维飞行显示程序



平台总体介绍

□ 平台组成

(3) Pixhawk 自驾仪系统

- Pixhawk 自驾仪
- 遥控器与接收机
- USB线连接电脑
- QGC地面站程序



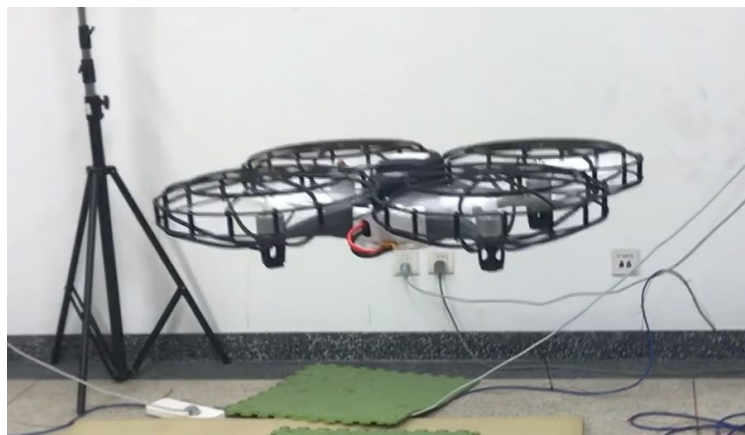


平台总体介绍

□ 平台组成

(4) 多旋翼硬件系统

- 机架系统
- 动力系统
- 外部传感器
- 测试台架





平台总体介绍

□ 平台组成

(4) 实验指导包

- 教材
- 实验指导书
- 程序包
- 详细视频教程





平台总体介绍

□ 平台优势

- 本实验平台在MATLAB/Simulink中提供了**多旋翼完整的仿真模型与控制器设计的相关接口**
- 初学者、学生或工程师可以利用所学知识**快速**进行控制器设计与验证
- 本平台提供代码自动生成与下载功能
- 本平台还提供硬件在环仿真测试功能，用户可以在计算机上对飞控硬件飞行效果进行**初步模拟测试**，排除试飞实验可能存在的各种问题
- 测试通过之后，将Pixhawk装在一个多旋翼硬件系统上，就可以进行**实际飞行测试**，调整参数，通过实验验证来评估设计控制算法的性能



控制器设计与仿真平台

本实验提供了一套基于Simulink/FlightGear的较为完整逼真的仿真环境例程，主文件见“e0\1. SoftwareSimExps\CopterSim3DEnvironment.slx”。

e0 > 1.环境展示



图. 提供的环境展示例程



图. Simulink仿真环境例程



控制器设计与仿真平台

□ 控制器



左图所示例程：简单的俯仰和滚转姿态的保持与控制，控制器响应遥控器的控制输入，将多旋翼控制到制定的俯仰滚转角度。

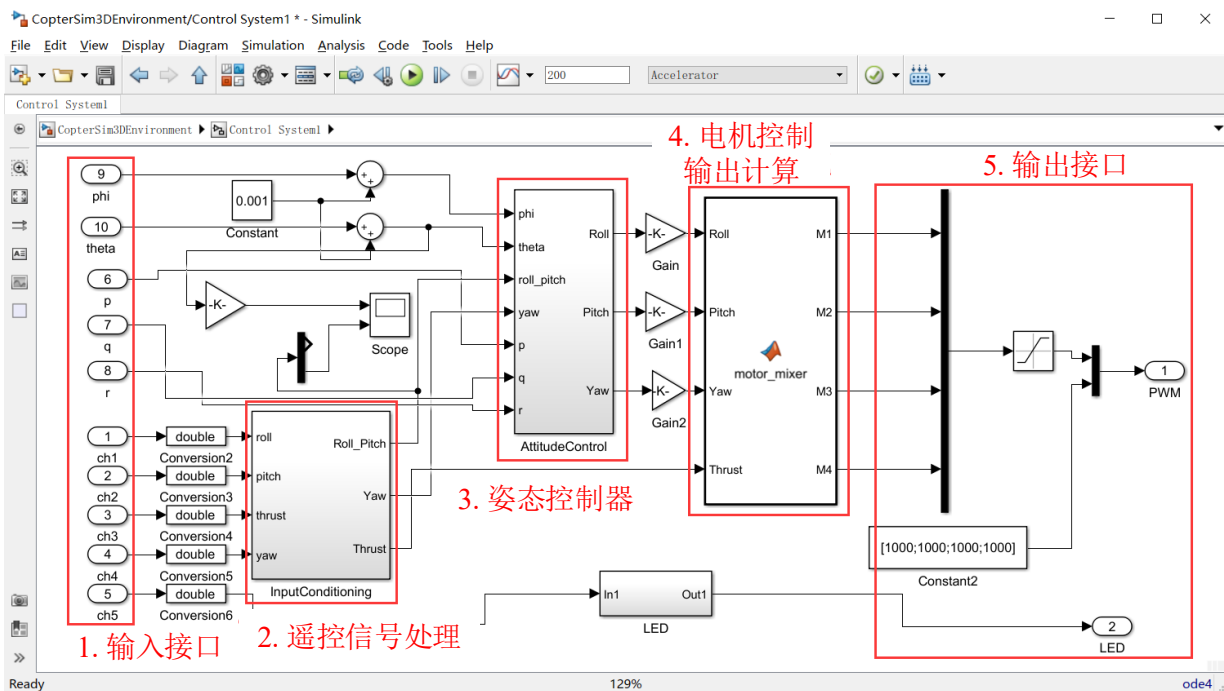


图. 控制器模块内部示意图

1. 输入接口：接收遥控器信号和飞机状态观测信号
2. 遥控信号处理：将通道信号映射为期望姿态角
3. 姿态控制器：计算期望输出力和力矩大小
4. 电机输出分配：将力和力矩的控制量映射为四个电机的油门控制量
5. 输出接口：将控制量补齐并映射为电机PWM信号



控制器设计与仿真平台

多旋翼模型

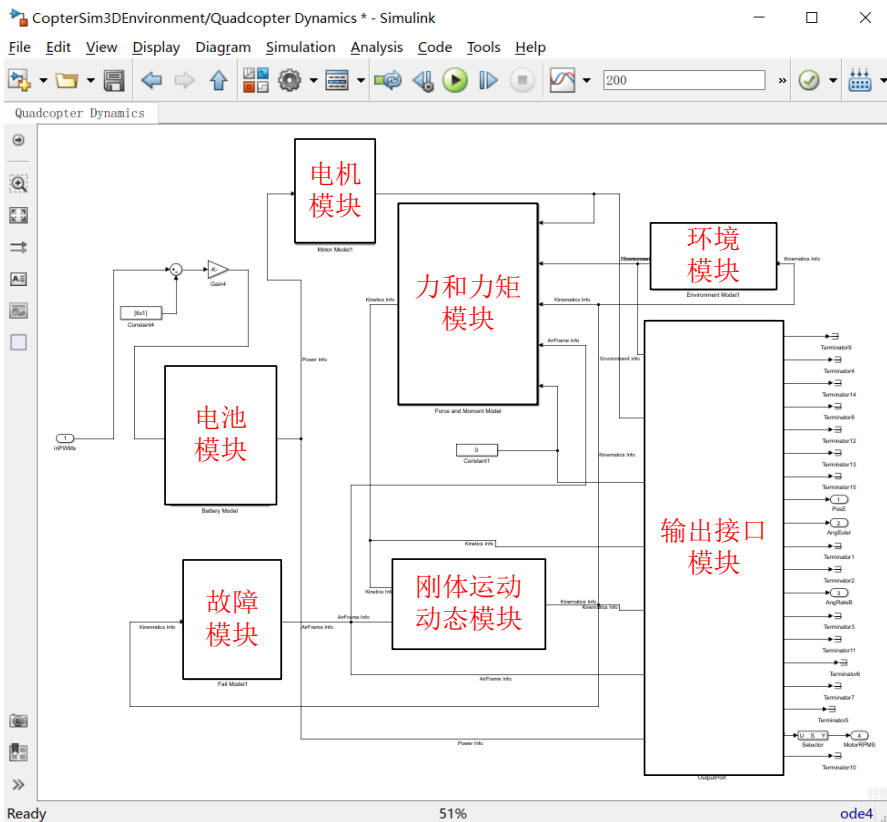


图. 多旋翼模型内部示意图

1) 模型输入: 电机的PWM控制量

2) 模型输出: 多旋翼的状态和传感器信息

- 电机模型: 模拟电机动态;
- 力和力矩模块: 模拟螺旋桨拉力、机身气动力、自身重力以及地面支撑力等所有的外部力和力矩;
- 刚体运动动态模型: 计算多旋翼的速度、位置和姿态等运动学状态;
- 环境模型: 根据高度计算加速度、空气密度等数据;
- 故障模型: 主要用于注入模型不确定(质量和转动惯量有关的)和故障数据;
- 电池模型: 模拟电池的放电过程;
- 输出接口模块: 将数据打包成需要的格式。



控制器设计与仿真平台

□ FlightGear接口

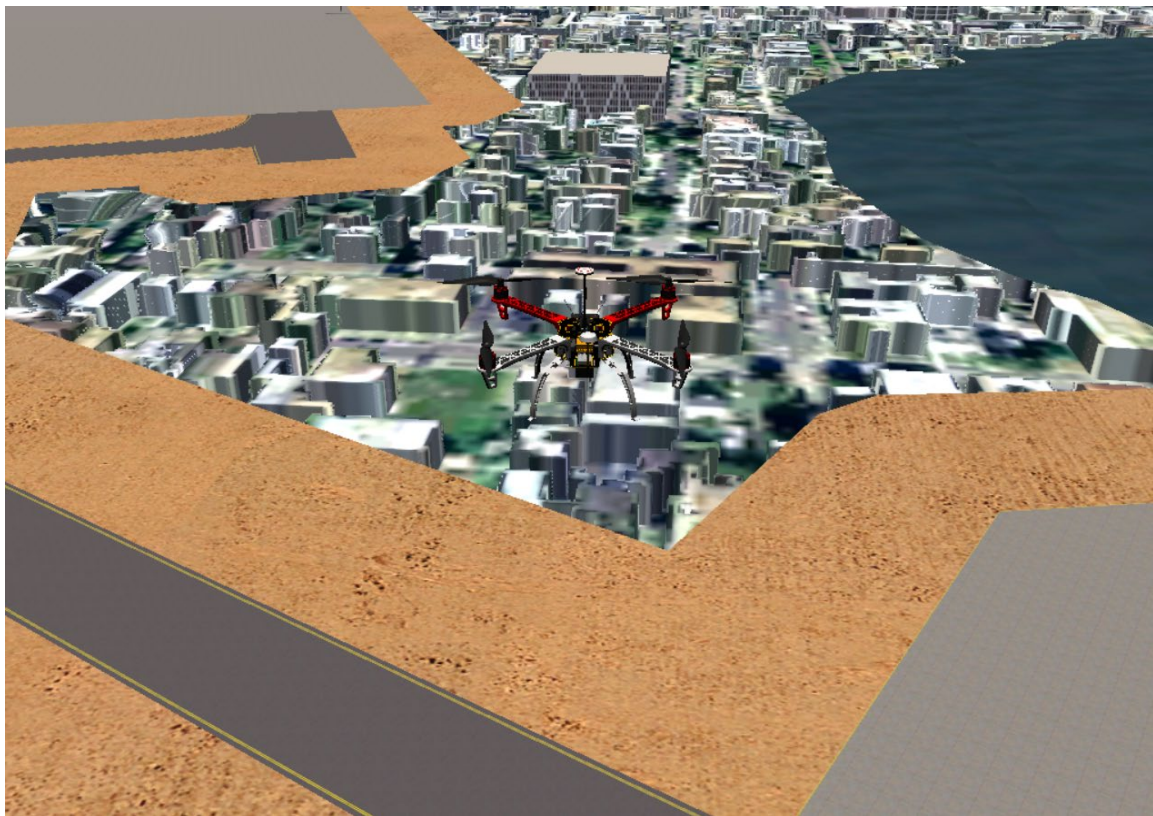
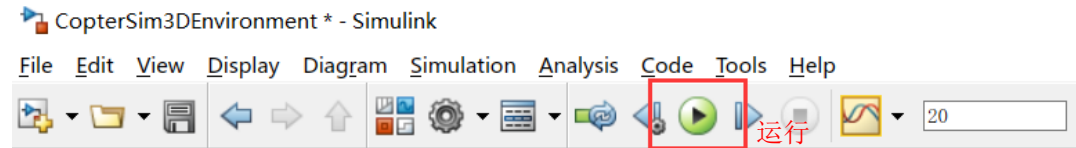
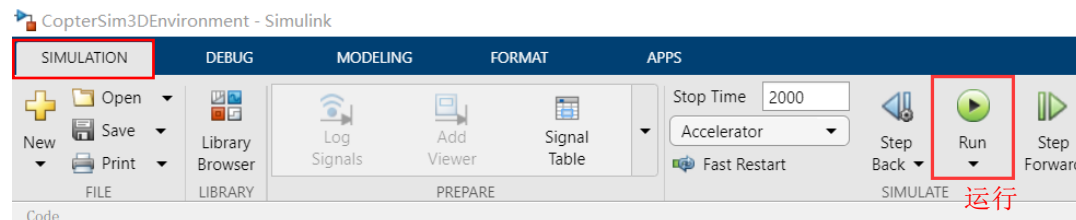


图. FlightGear界面



(a) Simulink “运行”按钮（MATLAB 2017b-2019a）



(b) Simulink “运行”按钮（MATLAB 2019b及更高版本）


■ 功能

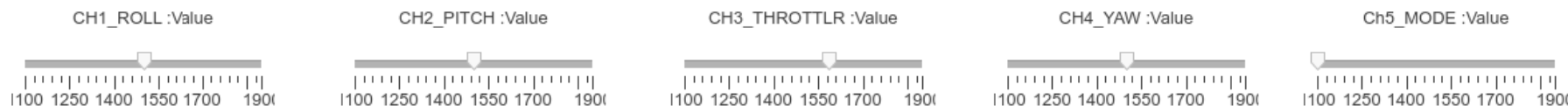
有三个输入量：多旋翼位置，多旋翼姿态欧拉角和电机的PWM信号。该模块会自动向本地的FlightGear相关接口发送多旋翼的飞行数据，从而在三维场景中看到多旋翼的飞行状态。

注：如果FlightGear无法正常显示，也可以打开3DDisplay来观察飞行效果

■ 效果

1、双击桌面“FlightGear-F450”快捷方式，打开FlightGear三维视景软件

2、当点击Simulink的  按钮，仿真开始。可以在FlightGear界面中看到多旋翼从地面缓慢起飞。你可以拖动下图Simulink界面的滑块来仿真用遥控器控制飞机的俯仰、滚转、偏航和油门等通道的运动。

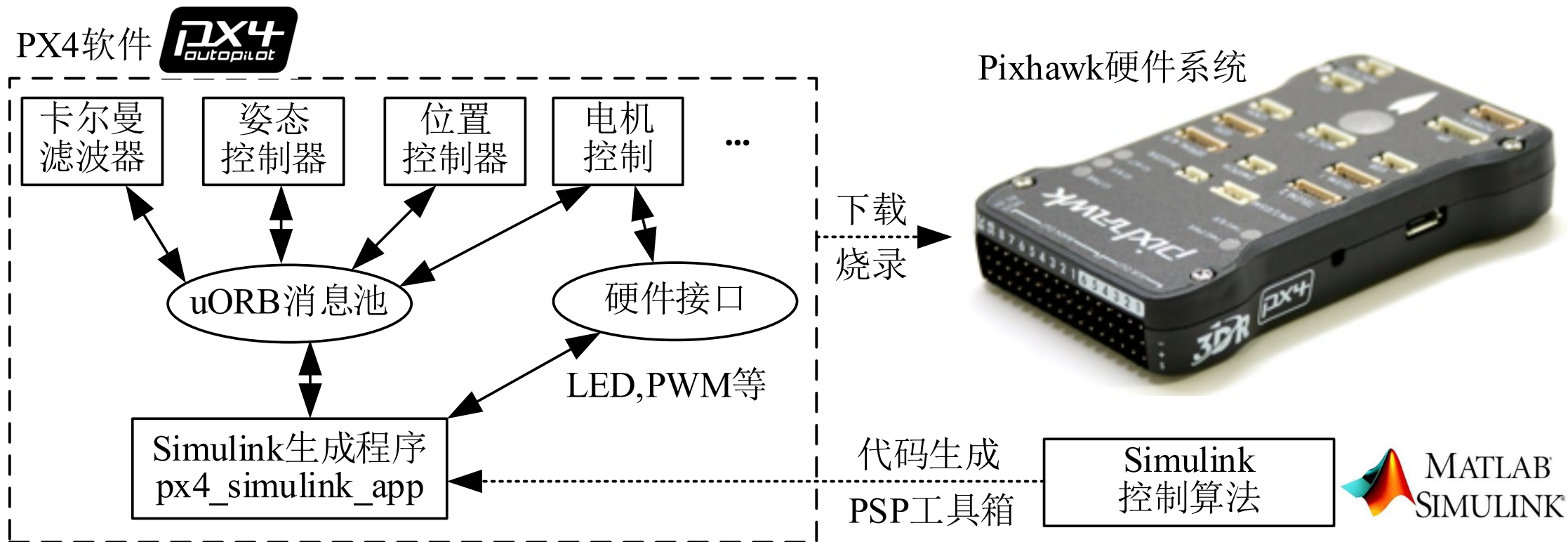




PSP工具箱

下图展示了PSP工具箱、PX4软件系统和Pixhawk硬件系统之间的关系。

Pixhawk Pilot Support Package (PSP, 自动驾驶仪支持包)
工具箱是Mathworks公司官方为Pixhawk推出的一个工具箱。该工具箱能在Simulink中利用嵌入式代码产生器 (Embedded Coder) 将Simulink模型自动驾驶仪算法自动编译和部署到Pixhawk硬件系统中。





PSP工具箱

□ 主要功能

1. 能在Simulink中对不同的飞机模型和自驾仪算法进行仿真和测试，并能自动将算法部署到Pixhawk自驾仪中；
2. 工具箱提供了一些实用实例，包括灯光控制、遥控器数据处理和姿态控制器等；
3. 工具箱中提供了很多接口模块，用于访问Pixhawk的硬件组件；
4. 能自动记录传感器、执行机构以及自己部署进去的控制器的飞行数据；
5. 能订阅和发布uORB话题消息。PX4自驾仪软件的所有数据都暂存在一个uORB消息池中，通过uORB订阅功能可以从消息池中读取感兴趣的话题，通过uORB发布功能可以特定的话题发布到消息池中供其他模块使用。



PSP工具箱

□ 与Pixhawk自驾仪系统关系

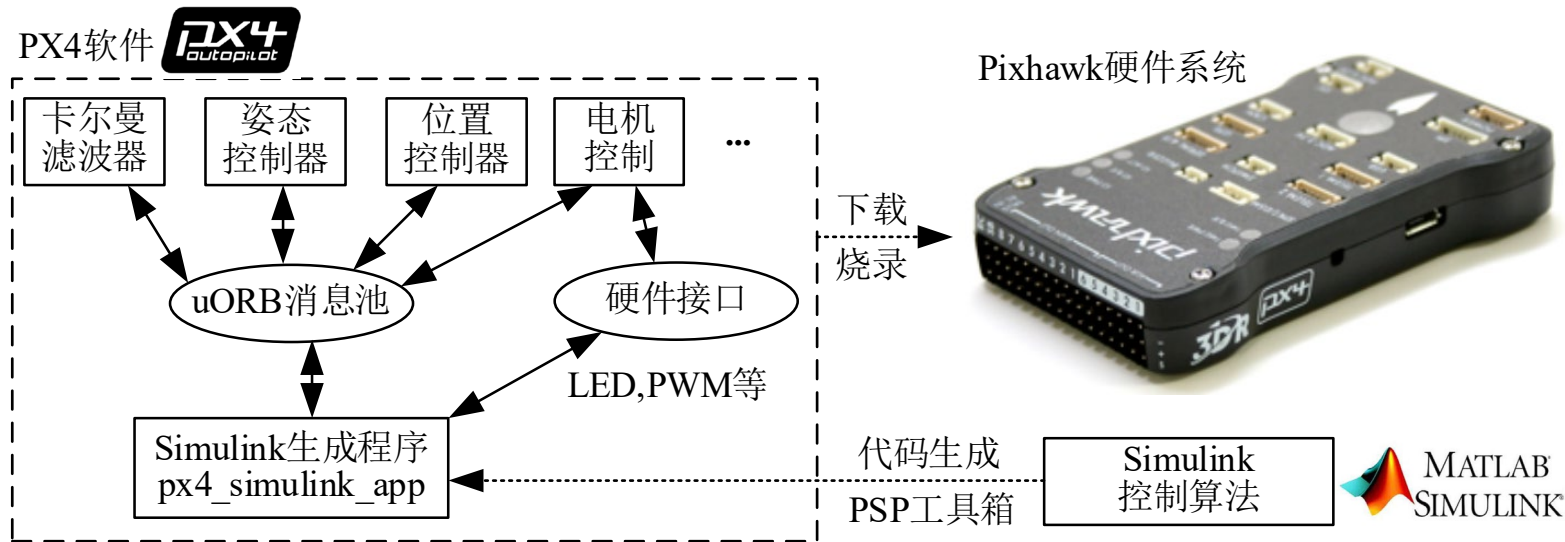
1. Pixhawk自驾仪系统的软硬件结构包括：硬件系统（类似于电脑主机）+PX4自驾仪软件系统（类似于主机上运行的操作系统和应用程序）的构架方式。
2. 整个PX4自驾仪软件系统可以分为若干个小模块，每个模块独立运行（多线程并行），各个模块通过uORB消息模块的订阅与发布功能实现数据的传输与交互。
3. Simulink生成的代码部署到PX4自驾仪软件之后，不会影响原生PX4自驾仪软件的运行，而是新增一个名为“px4_simulink_app”的独立模块（独立线程）并行于其他模块运行。
4. 由于原生PX4控制算法可能需要访问和“px4_simulink_app”同样的硬件输出资源，这会产生读写冲突。因此，平台一键部署脚本提供了自动屏蔽PX4原生固件对执行器的选项，以确保只有“px4_simulink_app”模块能够输出电机控制量。



PSP工具箱

□ 代码生成与部署过程

1. PSP工具箱将在Simulink中设计的控制算法生成C代码；
2. 将该代码导入到PX4自动驾驶仪的源代码中，生成一个“px4_simulink_app”独立运行的程序；
3. PSP工具箱调用编译工具将所有代码编译为“.px4”的PX4自动驾驶仪软件固件文件；
4. 将得到的固件下载到Pixhawk自动驾驶仪中并烧录，由Pixhawk自动驾驶仪执行带有生成的算法代码的PX4软件。



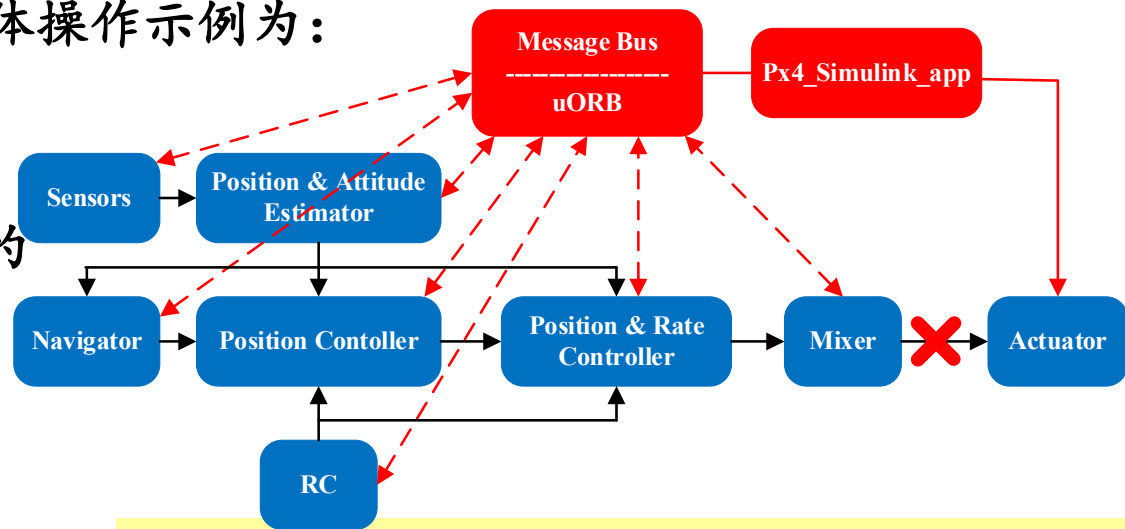


PSP工具箱

高级扩展功能

生成的Simulink代码也可以用于替换如右图所示PX4控制软件的某些原生模块（传感器、滤波器、姿态控制器等），但是需要手动修改PX4固件代码来屏蔽原始模块的输出接口。例如，想用Simulink实现一个滤波器模块（输入传感器数据，输出状态滤波数据）来替换PX4原始的滤波器，需要手动屏蔽右图中的“Position & Attitude Estimator”滤波器模块向uORB消息池发布滤波后姿态数据（对应名为vehicle_attitude的uORB消息）。具体操作示例为：

1. 打开“Firmware\src\modules\ekf2\ekf2_main.cpp”文件（扩展卡尔曼滤波器模块对应代码）中；
2. 在其中屏蔽掉“ORB_ID(vehicle_attitude)”消息相关的发送代码。例如，搜索关键字“_att_pub”的代码行，并找到其中含有“publish”和“att”的发送代码行，将其替换为“UNUSED(att);”。这里UNUSED用于防止编译器出现变量未使用的警告。
3. 在Simulink中编写姿态滤波器，并用uORB Write模块发送vehicle_attitude消息即可实现姿态滤波器功能的替换。



注：另一种可行方法是修改PX4模块启动脚本文件“Firmware\ROMFS\px4fmu_common\init.d\rcS”，并注释掉想要屏蔽的模块



PSP工具箱

□ Pixhawk Target Blocks模块库介绍

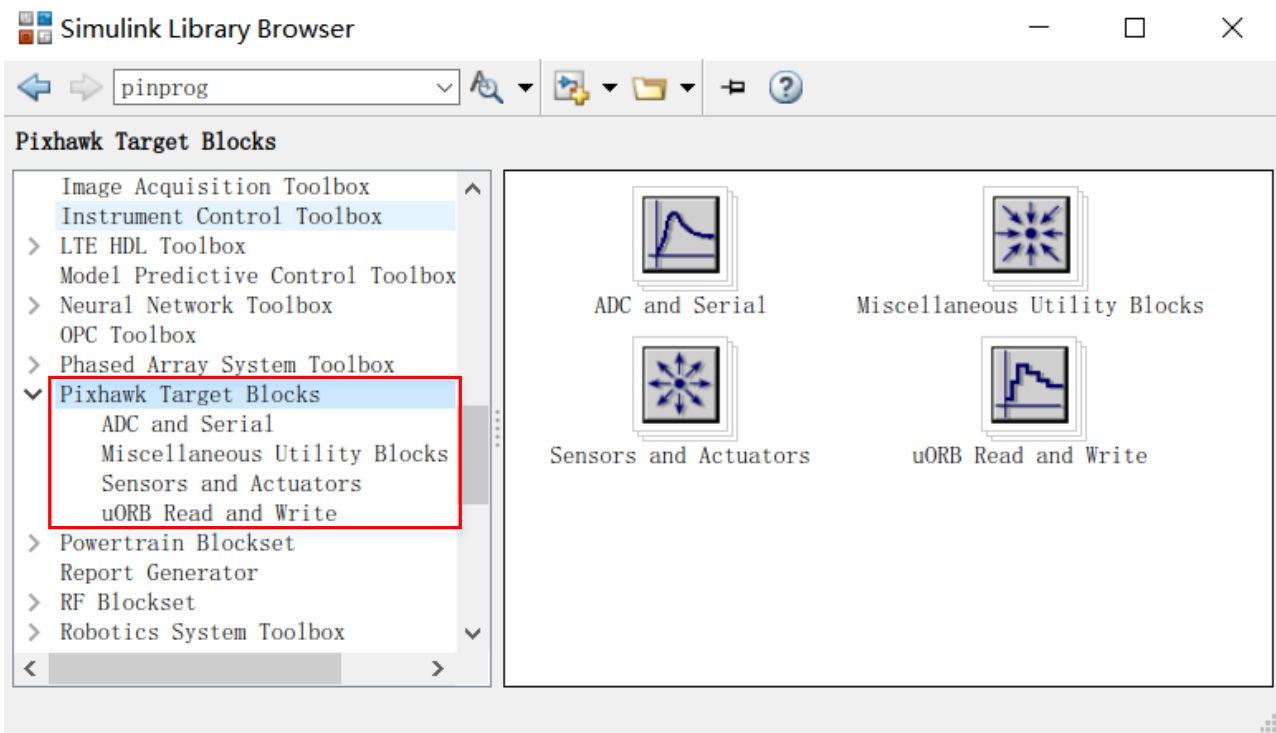


图. Simulink PSP工具箱展示

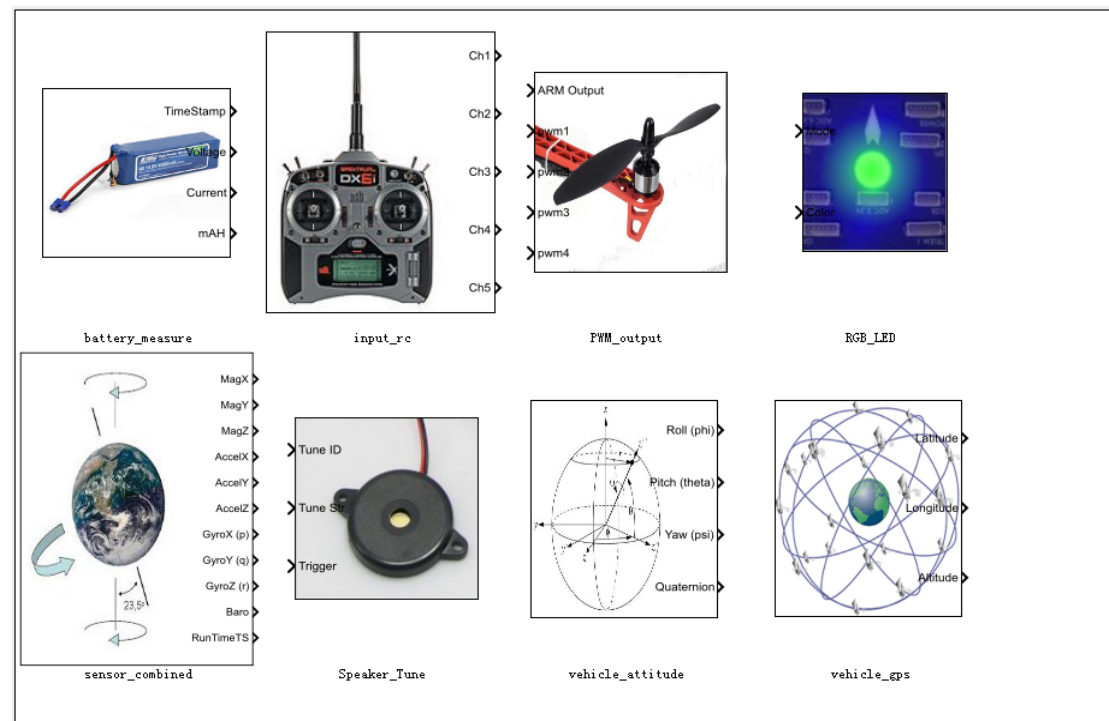
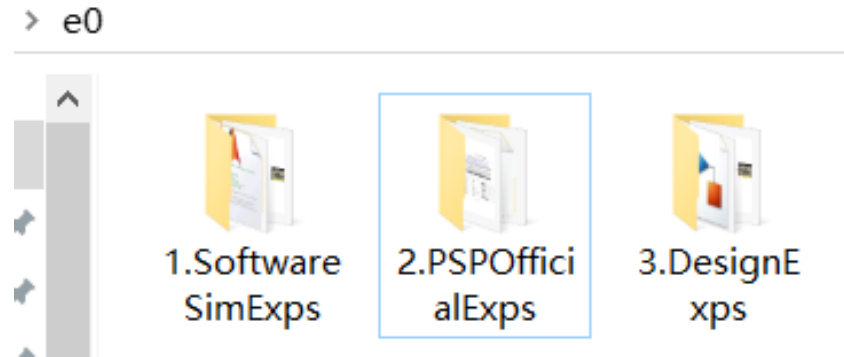
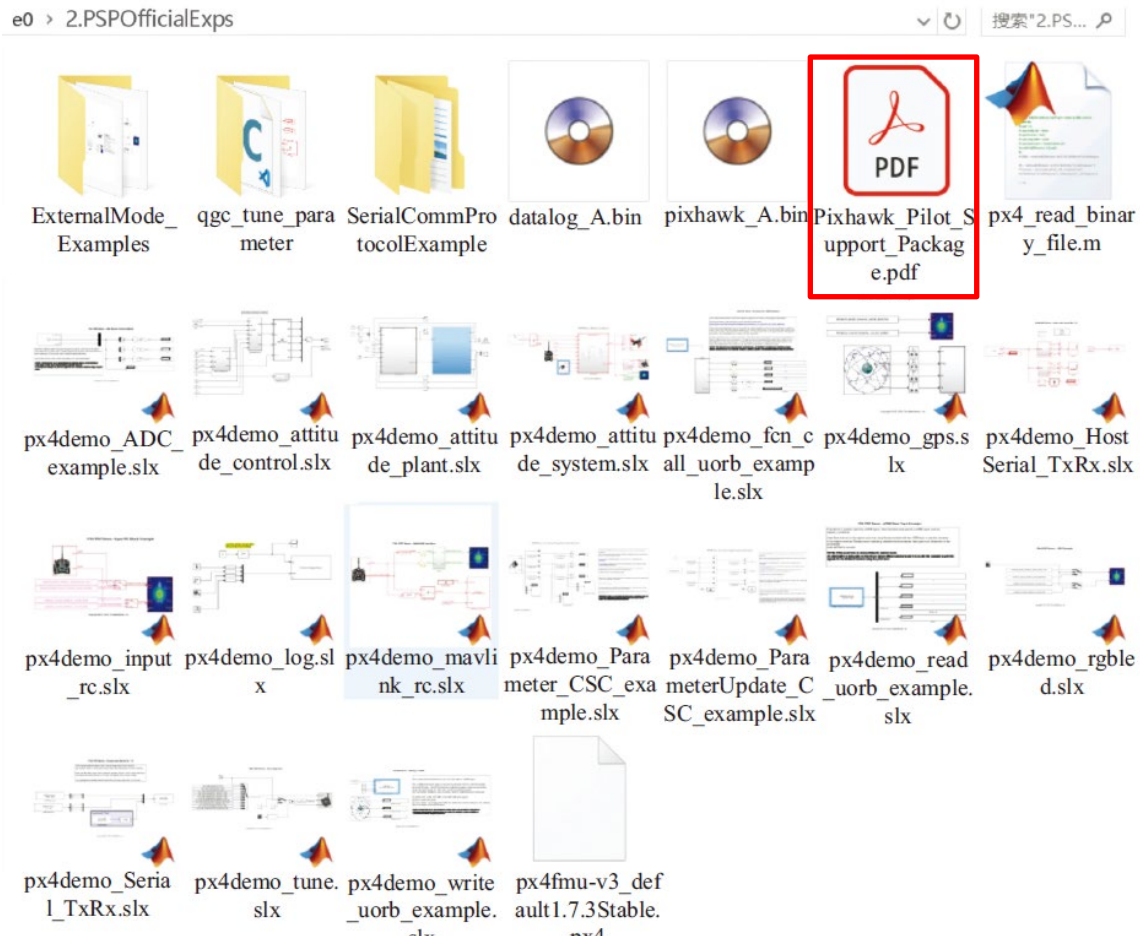


图. PSP工具箱传感器和执行器接口库示意图



PSP工具箱

□ Pixhawk Target Blocks模块库介绍



PSP工具箱官方还提供了大量的例程（详见文件夹“e0\2.PSPOfficialExps”）以及官方使用手册（详见文档“e0\2.PSPOfficialExp\Pixhawk_Pilot_Support_Package.pdf”）。通过上述例程和文档，用户可以快速掌握各个关键模块的使用方式，以及代码生成与下载流程。

图. PSP工具箱官方例程及使用手册



PSP工具箱

工具箱内各个模块的使用说明

电机PWM模块：通过这个模块可以发送PWM信号到PX4IO的输出端口以控制电机转动，可以选择PWM的更新率及输入通道。



PWM_output

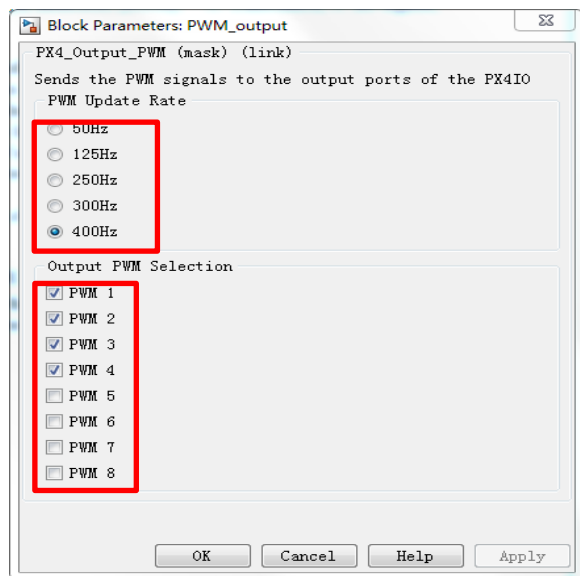


图. 电机PWM输出模块及其参数设置框

遥控器输入模块：通过这个模块可以选择接收遥控器的通道数量，以及其他的一些信息。每个选项的含义可点击对话框的“help”按钮查看或者查阅官方PDF文档。（PSP工具箱提供的该模块学习例程见“e0\2.PSPOfficialExps\px4demo_input_rc.slx”）



input_rc

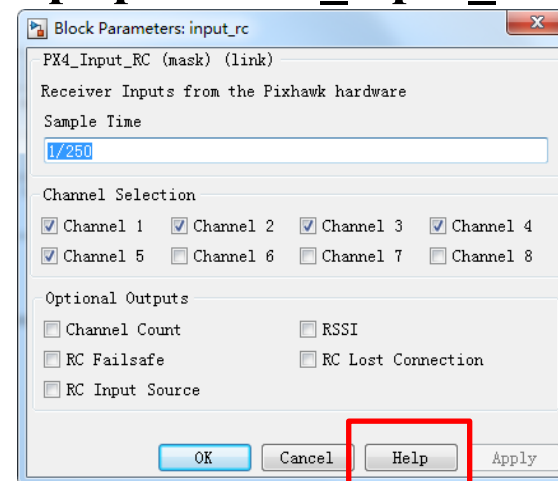


图. 遥控器输入模块及其参数设置框



PSP工具箱

□ 工具箱内各个模块的使用说明

蜂鸣器模块：通过这个模块可以控制蜂鸣器在特定的事件发出特定的音调。（PSP工具箱提供的该模块学习例程见“PX4PSP\examples\px4demo_tune.slx”）

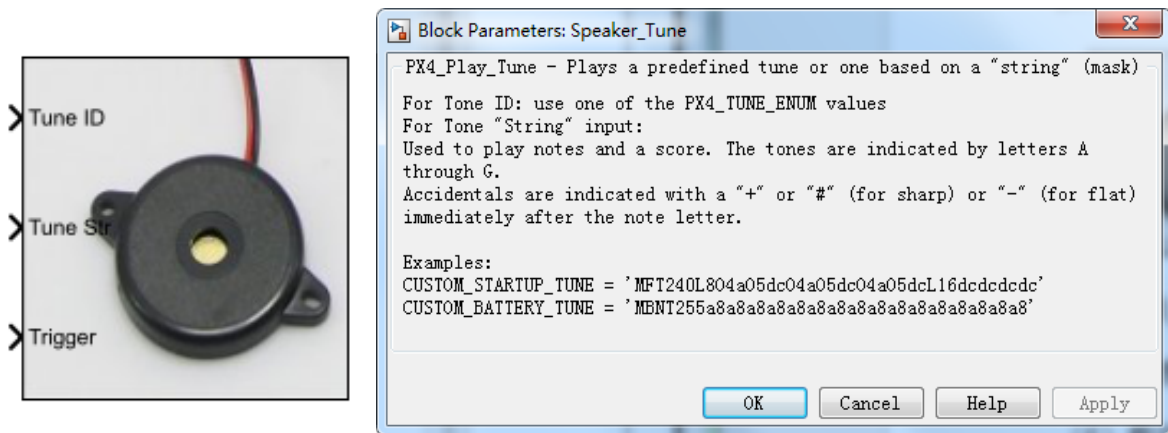


图. 蜂鸣器模块及其参数设置框

RGB_LED模块：通过这个模块可以控制LED灯闪烁的模式和颜色。如下图所示，模块接收两个输入，一个是模式（Mode），另一个是颜色（Color），可用模式和可用颜色可以点击对话框的“help”按钮查看。（PSP工具箱提供的该模块学习例程见“e0\2.PSPOfficialExps\px4demo_rgbled.slx”）

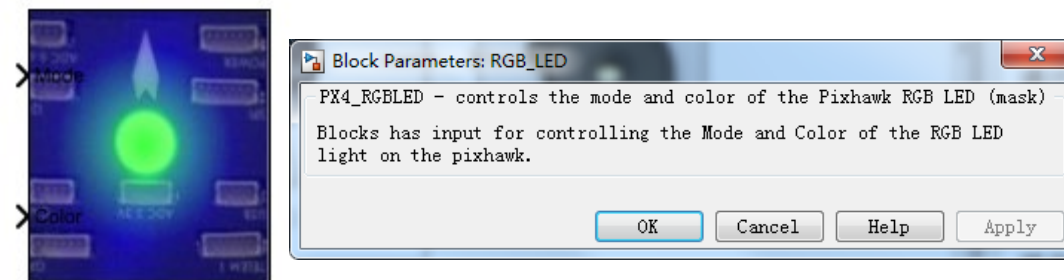


图. 多彩LED灯模块及其参数设置框



PSP工具箱

工具箱内各个模块的使用说明

传感器组合模块：通过这个模块可以获取Pixhawk中可用的传感器数据，然后这些数据可以用于控制模型的设计。可获取的数据包括磁力计、加速度计、陀螺仪、气压计和时间戳。(PSP工具箱提供学习例程见“e0\2.PSPOfficialExps\px4demo_attitude_system.slx”)

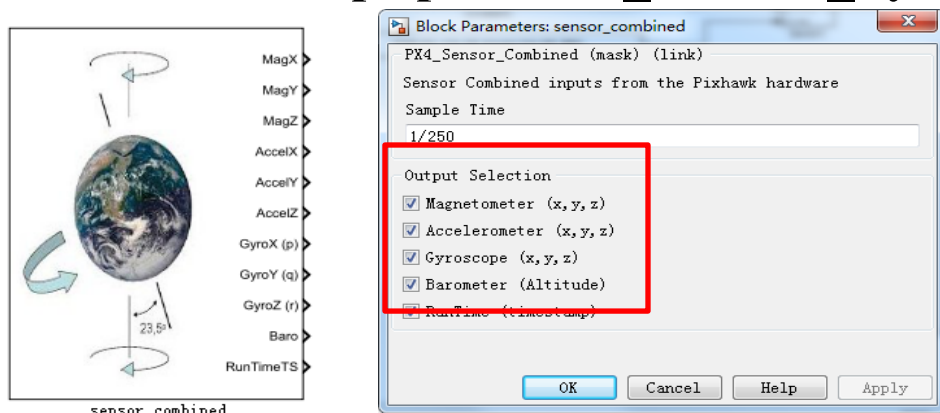


图. 传感器数据组合模块及其参数设置框

姿态数据模块：该模块提供了经过滤波的姿态数据（欧拉角和四元数），通过复选框可以勾选想要输出的姿态数据。(PSP工具箱提供的该模块学习例程见“e0\2.PSPOfficialExps\px4demo_attitude_system.slx”)

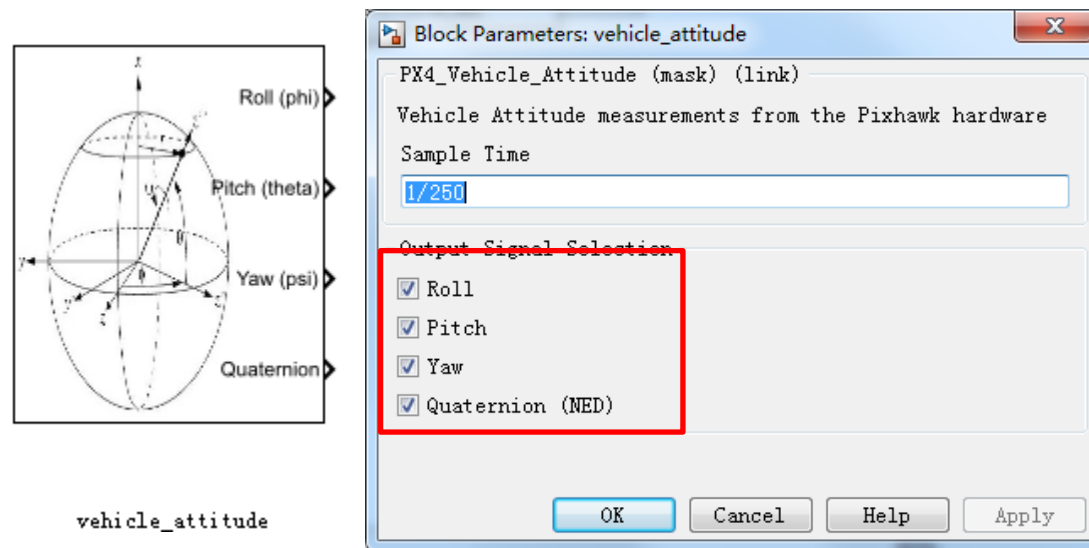


图. 姿态模块及其参数设置框



PSP工具箱

□ 工具箱内各个模块的使用说明

GPS数据模块：通过这个模块可以获取Pixhawk的GPS数据，它是通过订阅uORB话题

“vehicle_gps”实现的，所以在实际运行时需要保证Pixhawk上插入GPS模块才能获取正确数据。

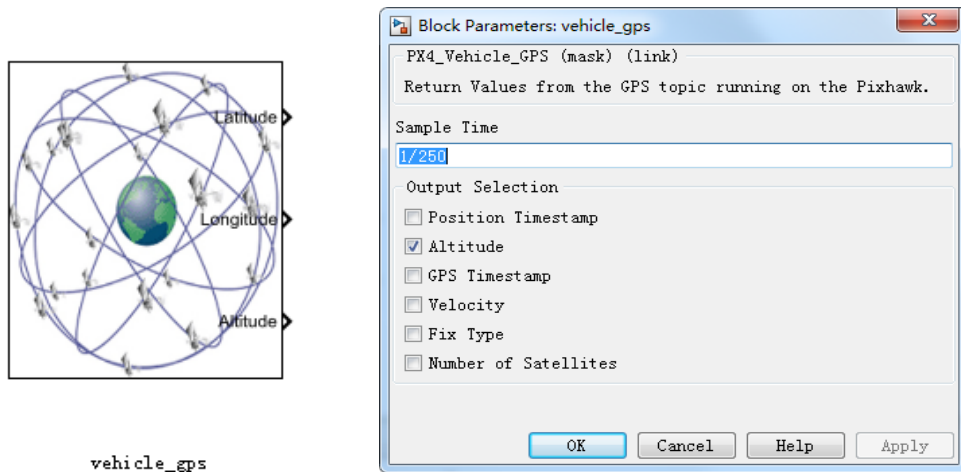


图. GPS数据模块及其参数设置框

电池数据模块：通过这个模块可以获取电池的实时数据，它也是通过订阅uORB话题“battery_status”实现的，所以在实际运行时需要保证Pixhawk上插入电源模块才能获取正确数据。

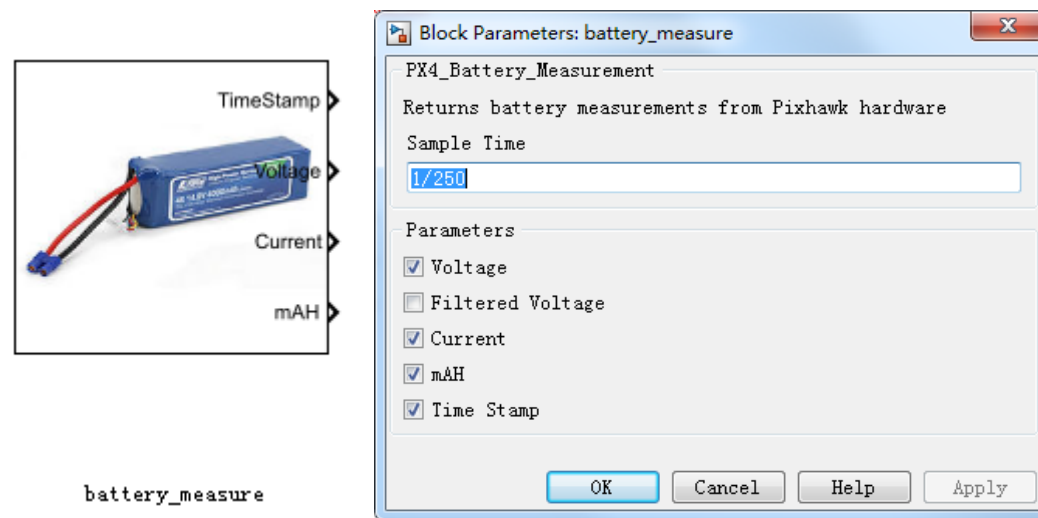


图. 电池数据模块及其参数设置框



PSP工具箱

实际上，本小节前面提到的所有模块在底层均是通过uORB消息的读写来实现的。理论上通过uORB消息的读写，我们可以获取并改变飞控中的所有数据，实现更大的自由度。更多使用教程可以参考<https://dev.px4.io/master/en/middleware/uorb.html>

□ 工具箱内各个模块的使用说明

uORB模块。通过左下图所示的uORB模块可以从PX4自驾仪软件中读取或发送uORB消息，其中PX4自驾仪支持的所有uORB消息类型可以在软件包安装目录（默认路径是“C:\PX4PSP”）下的“Firmware\msg”目录中查看。双击左下图中的“uORB write”模块可以打开右下图所示的uORB消息配置界面，在其中可以选择发送的uORB消息名称和数据。

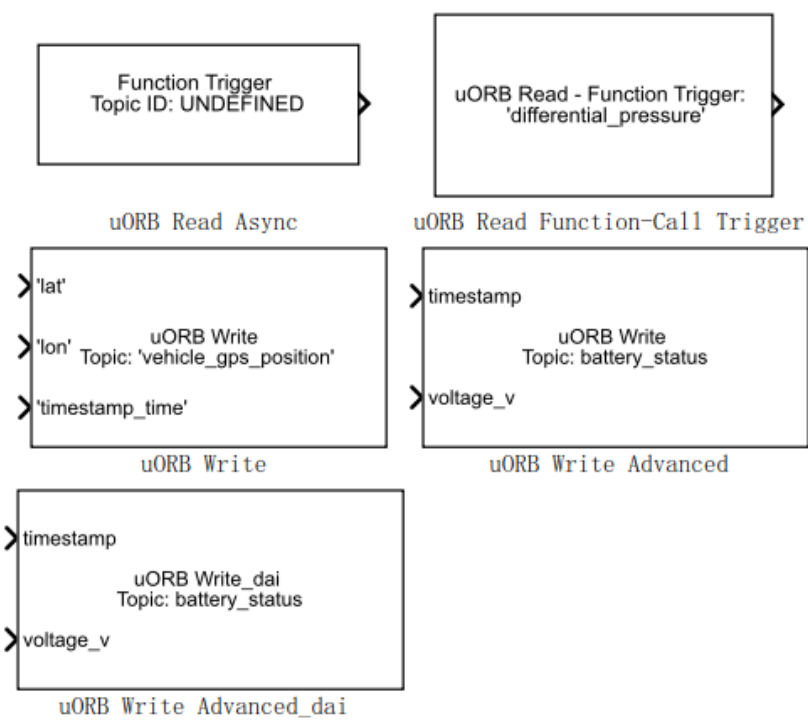


图. 用于消息读写的uORB模块

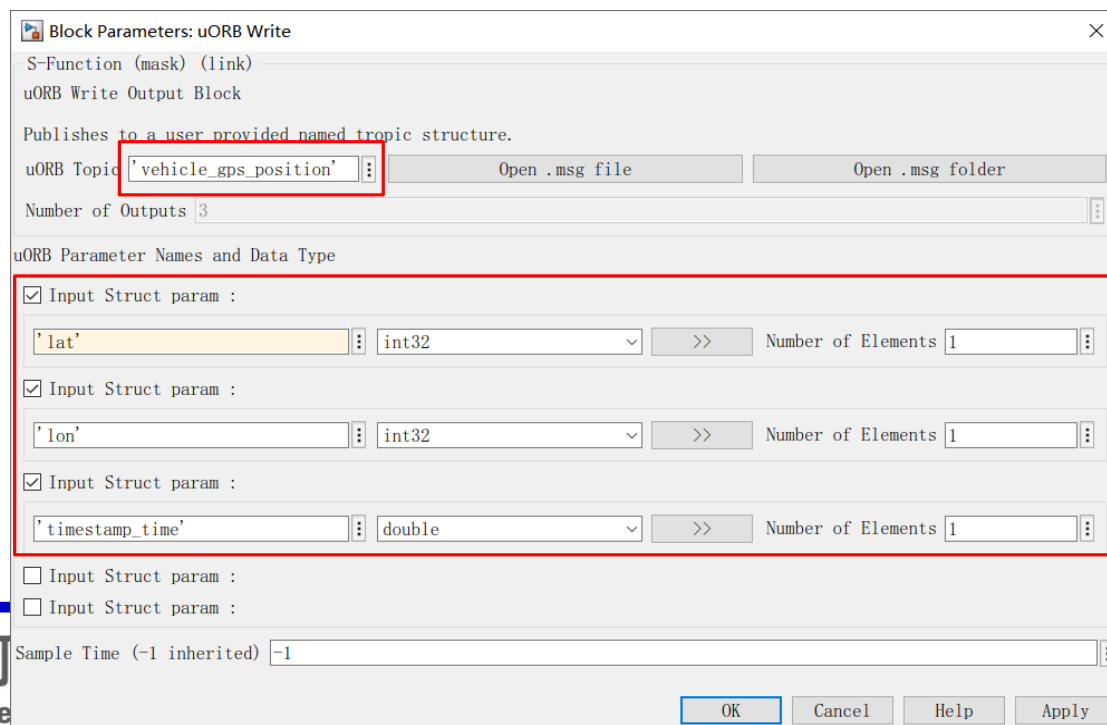


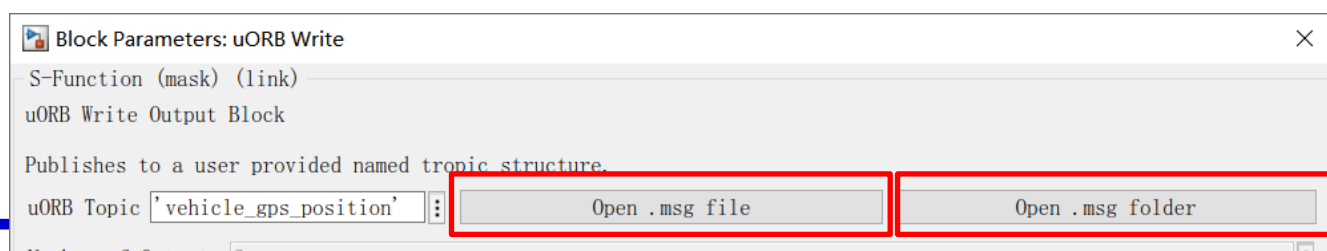
图. uORB写接口模块参数设置框



PSP工具箱

工具箱内各个模块的使用说明

点击右上图“uORB write”模块配置界面中的“Open .msg file”按钮可以打开选定的uORB消息文件（见图左下图）；点击按钮“Open .msg folder”打开消息目录（见右下图）。



```
D:\PX4PSP\Firmware\msg\vehicle_gps_position.msg
编辑器 视图
1 # GPS position in WGS84 coordinates.
2 # the auto-generated field 'timestamp' is for the position & velocity (microseconds)
3 int32 lat # Latitude in 1E-7 degrees
4 int32 lon # Longitude in 1E-7 degrees
5 int32 alt # Altitude in 1E-3 meters above MSL, (millimetres)
6 int32 alt_ellipsoid # Altitude in 1E-3 meters above Ellipsoid, (millimetres)
7
8 float32 s_variance_m_s # GPS speed accuracy estimate, (metres/sec)
9 float32 c_variance_rad # GPS course accuracy estimate, (radians)
10 uint8 fix_type # 0-1: no fix, 2: 2D fix, 3: 3D fix, 4: RTCM code differential, 5: Real-time kinematic
11
12 float32 eph # GPS horizontal position accuracy (metres)
13 float32 epv # GPS vertical position accuracy (metres)
```

图. uORB 消息文件

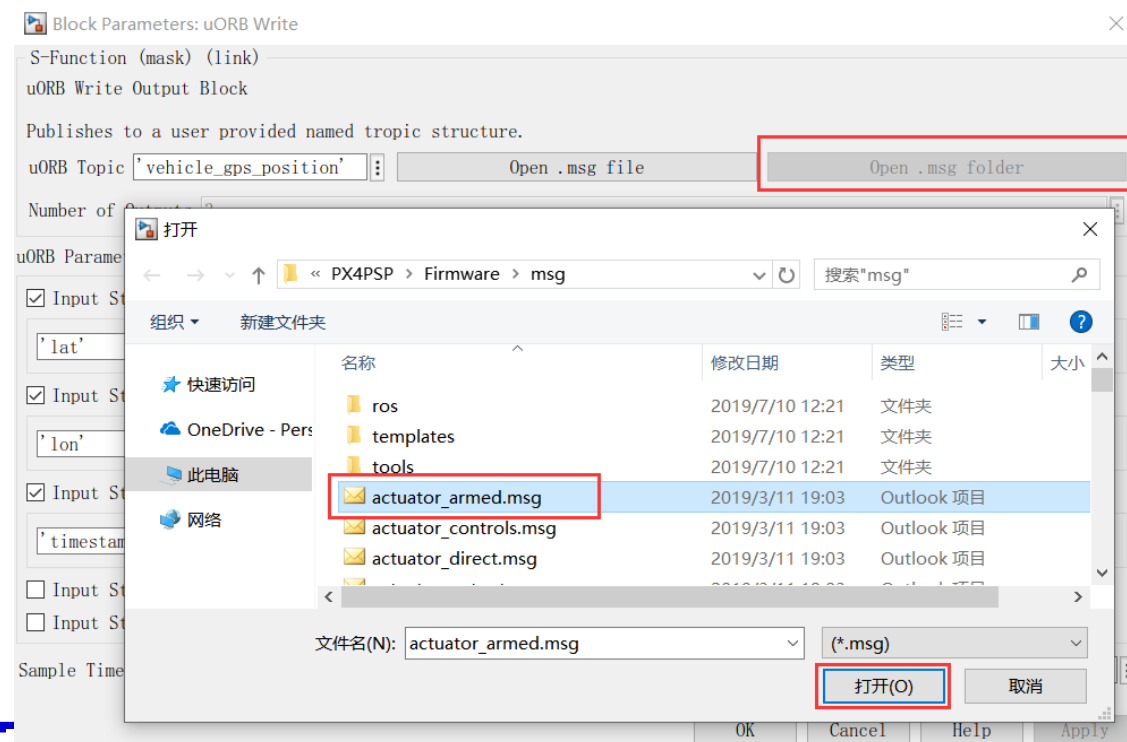


图. uORB 模块参数设置框“Open .msg folder”按钮弹出框

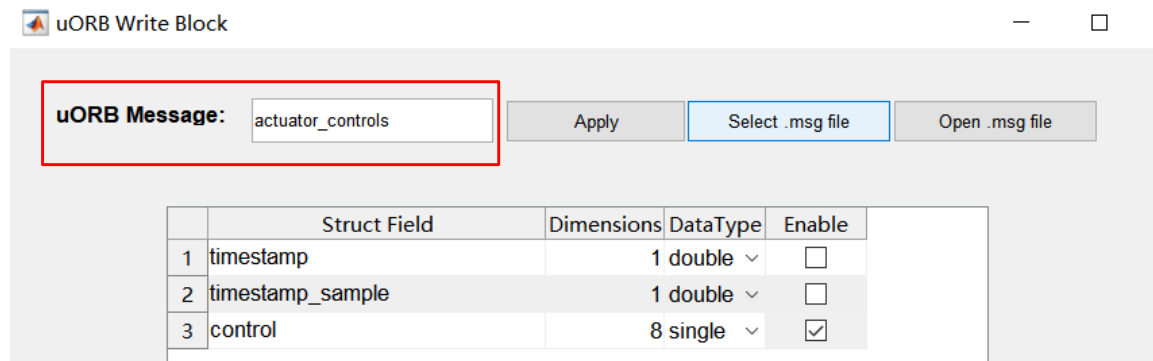


PSP工具箱

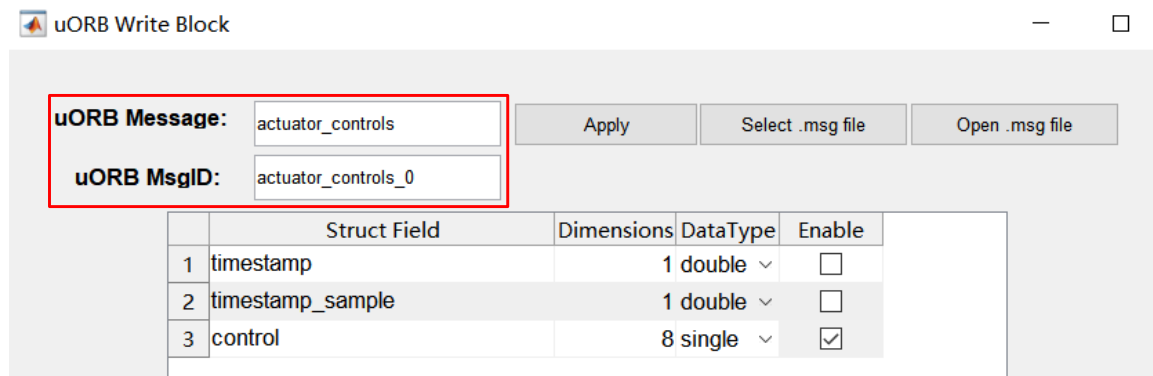
□ 工具箱内各个模块的使用说明

实际上，本小节前面提到的所有模块在底层均是通过uORB消息的读写来实现的。理论上通过uORB消息的读写，**我们可以获取并改变自动驾驶仪内的所有消息和中间变量，实现更高级的控制器功能。因此掌握该模块的使用，我们拥有更大的权限与自由度，使得实现更为复杂的功能称为可能。**（PSP工具箱提供的该模块学习例程见“e0\2.PSPOfficialExps\px4demo_fcn_call_uorb_example.slx”和“e0\2.PSPOfficialExps\px4demo_write_uorb_example.slx”）

uORB写操作有两个模块可以选择，右图带后缀“_dai”的模块允许发送基于某uORB消息基类（例如actuator_controls）的扩展消息（例如，actuator_controls_0）。



(a) uORB Write Advanced模块，消息ID默认与.msg文件名相同

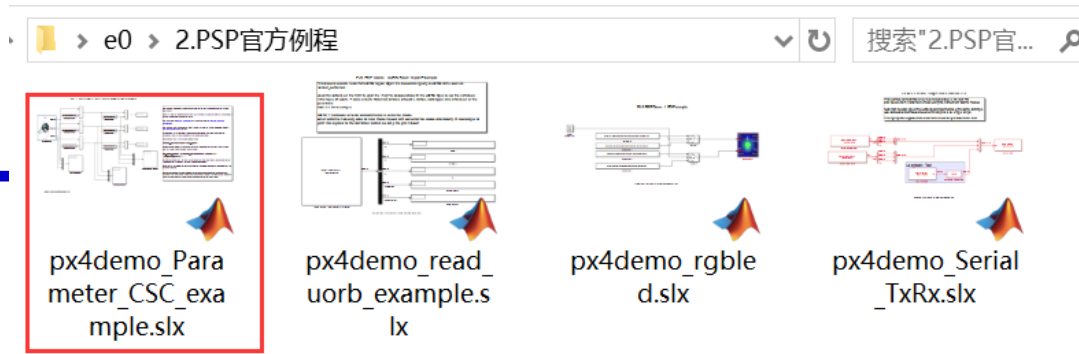


(b) uORB Write Advanced_dai模块，消息ID可以设置与.msg文件名不同



PSP工具箱

工具箱内各个模块的使用说明



为了方便调参与测试，PSP工具箱还提供了访问Pixhawk内部参数的方法，这样可以在飞行测试实验中，通过地面站软件来修改Simulink生成控制器参数，而不需要在Simulink中修改后再生成代码。官方例子见“e0\2.PSPOfficialExps\px4demo_Parameter_CSC_example.slx”文件。使用时需要将参数声明函数放入Simulink的菜单栏的“File” – “Model Properties” - “Callbacks” - “InitFcn”中。

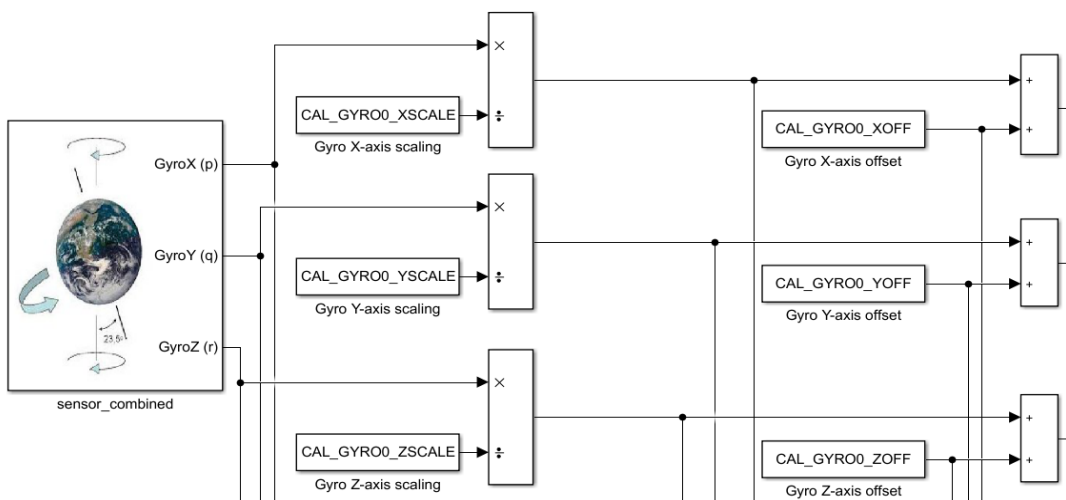


图. Pixhawk参数读取例程

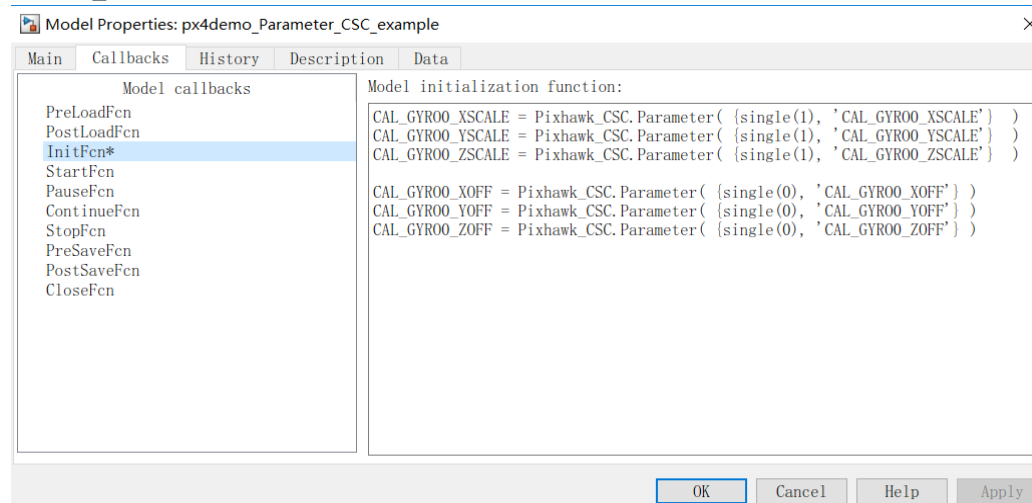


图. Pixhawk参数读取的Simulink初始脚本设置



PSP工具箱

□ Simulink配置实现PSP工具箱自动代码生成

1) 控制器代码生成前的准备工作

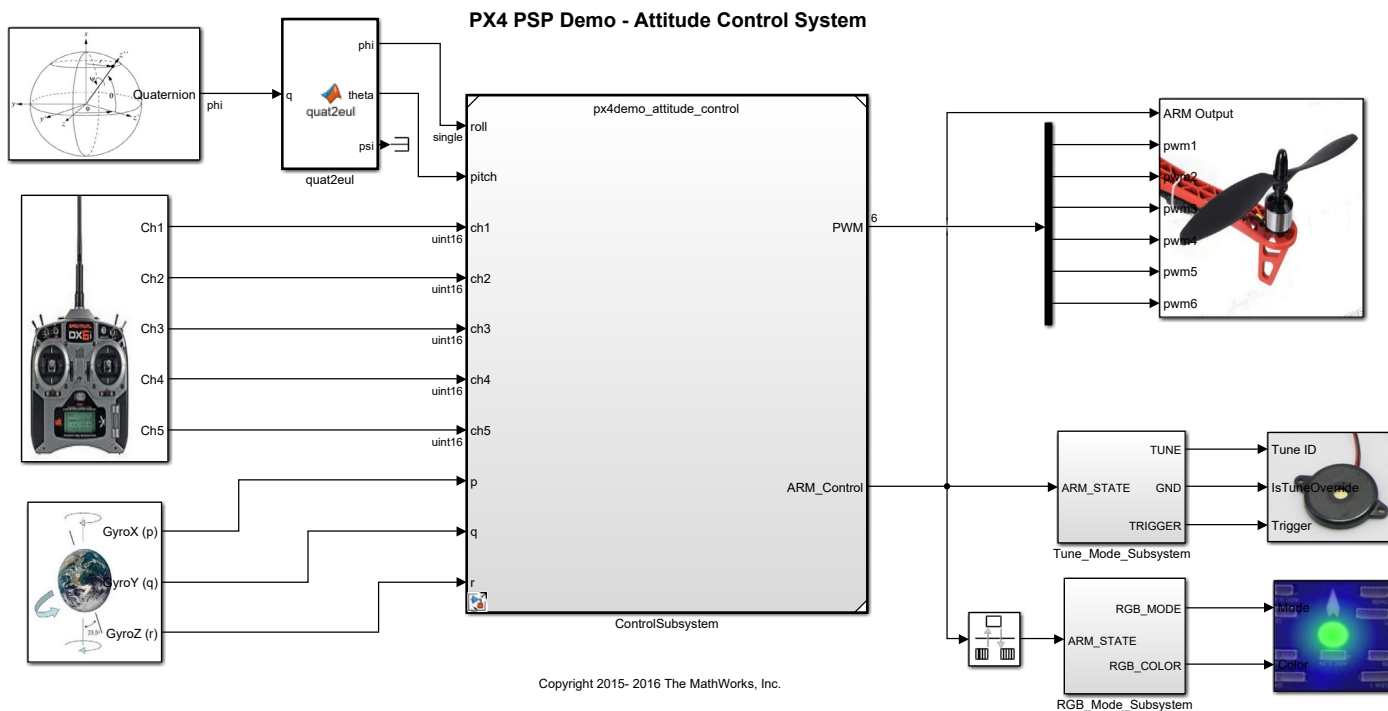


图. 控制器连接PSP接口模块

- 在Simulink中设计一个控制器，并进行软件在环仿真验证；
- 将通过验证的控制器复制到一个新的Simulink文件中；
- 将控制器的输入输出口与PSP工具箱模块库中的输入（传感器、遥控器）和输出（电机PWM、LED灯和蜂鸣器）接口进行连接；
- 完整的例程见“e0\2.PSPOfficialExps\px4demo_attitude_system.slx”文件。

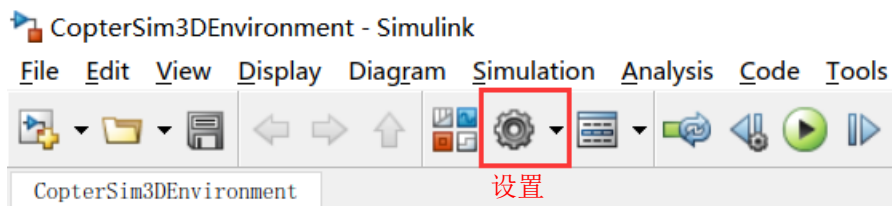


PSP工具箱

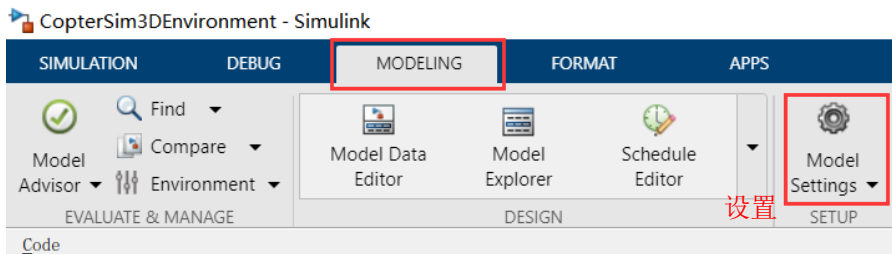
□ Simulink配置实现PSP工具箱自动代码生成

2) 选择目标硬件

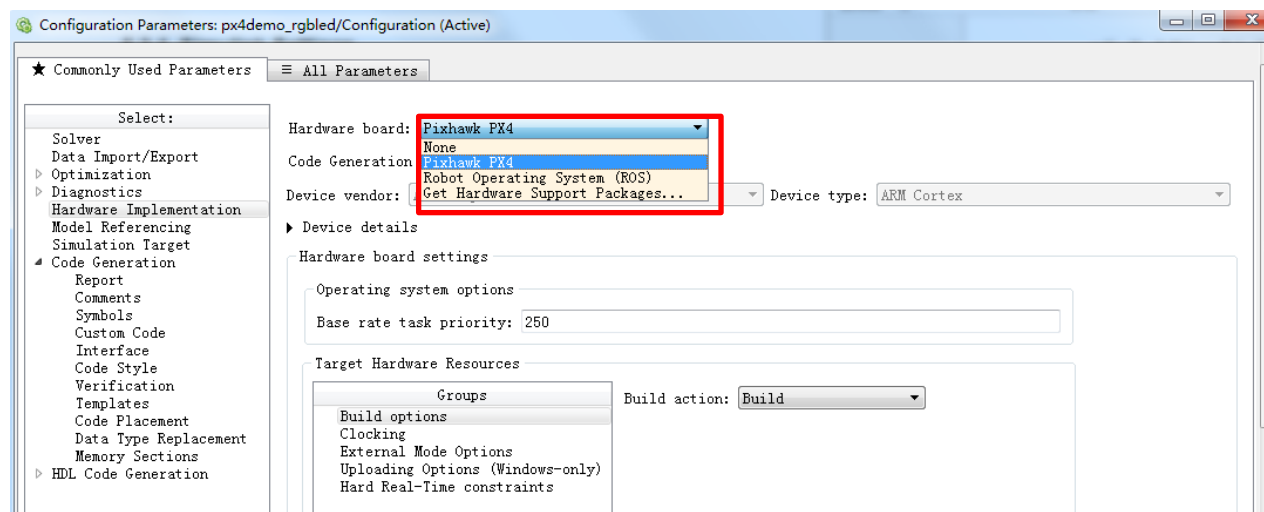
为了让Simulink根据目标硬件编译和组建模型，需要点击Simulink的“设置”按钮进行相应的设置。在弹出的模型配置页面，在“Hardware Implementation”标签页的“Hardware Board”下拉框中设置硬件板卡为“Pixhawk PX4”。



(a) Simulink “设置”按钮（MATLAB 2017b~2019a）



(b) Simulink “设置”按钮（MATLAB 2019b及更高版本）



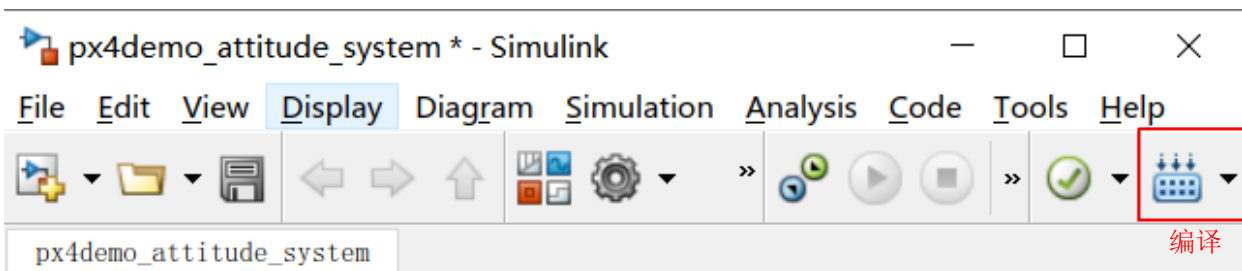


PSP工具箱

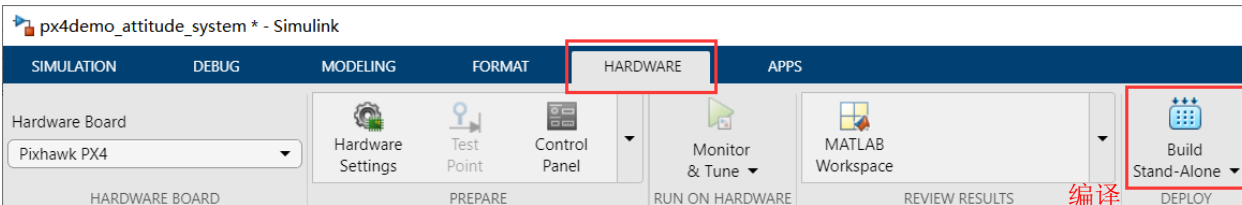
□ Simulink配置实现PSP工具箱自动代码生成

3) 编译与固件生成

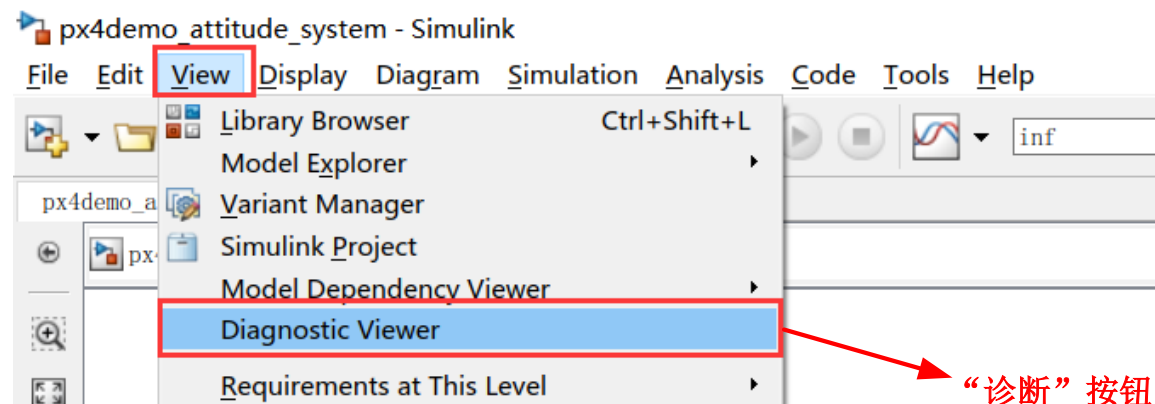
如下图点击Simulink的“编译”按钮，再如右图点击Simulink的“诊断”按钮



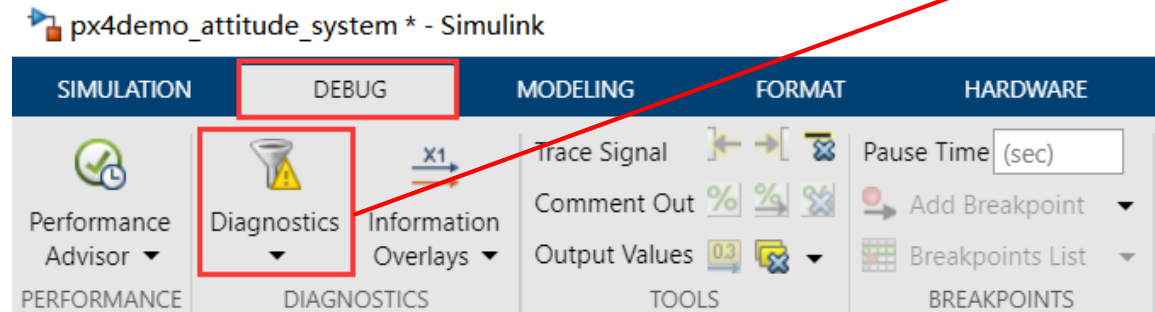
(a) Simulink “编译”按钮 (MATLAB 2017b~2019a)



(b) Simulink “编译”按钮 (MATLAB 2019b及更高版本)



(a) Simulink “诊断”按钮 (MATLAB 2017b~2019a)



(b) Simulink “诊断”按钮 (MATLAB 2019b及更高版本)



PSP工具箱

□ Simulink配置实现PSP工具箱自动代码生成

4) 编译过程与结果

一次成功的编译过程在

“Diagnostic Viewer”窗口中的状态如图(a)所示，可以看到编译进度条完成，同时显示“Successfully generated all binary outputs”消息。此外，在编译完成之后，Simulink会自动弹出如图(b)所示的“Code Generation Report”代码生成文档。

```
px4demo_input_rc
[5/8] Linking C static library
src/modules/px4_simulink_app/libmodules_
px4_simulink_app.a
[6/8] Linking CXX executable
nuttx_px4fmu-v3_default.elf
[7/8] Generating px4fmu-v2.bin
[8/8] Creating
/mnt/c/PX4PSP/Firmware/build/px4fmu-
v3_default/px4fmu-v3_default.px4
"### Finished calling CMAKE build process
###"
"### Done invoking postbuild tool."
"### Successfully generated all binary
outputs."

C:\PX4PSP\examples\px4demo_input_rc_ert_r
tw>exit /B 0
### Successful completion of build
procedure for model: px4demo_input_rc

Build process completed successfully
```

(a) 编译过程

Code Generation Report for 'px4demo_input_rc'

Model Information

Author	skuznick
Last Modified By	skuznick
Model Version	1.48
Tasking Mode	MultiTasking

[Configuration settings at time of code generation](#)

Code Information

System	ert.tlc
Target File	
Hardware	ARM Compatible->ARM Cortex
Device	
Type	

Generated Code

- [-] Main file
 - [ert_main.c](#)
- [-] Model files
 - [px4demo_in...](#)

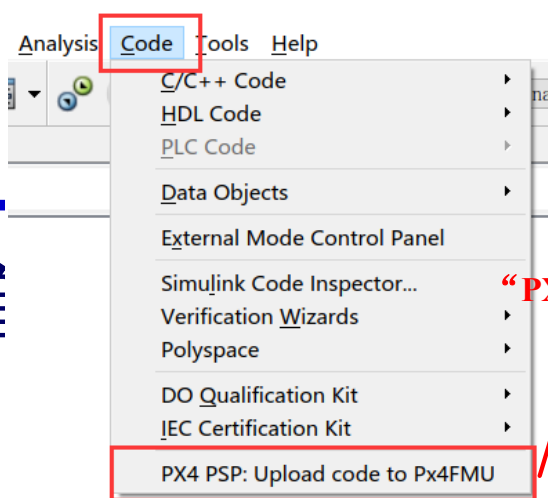
(b) 代码生成报告



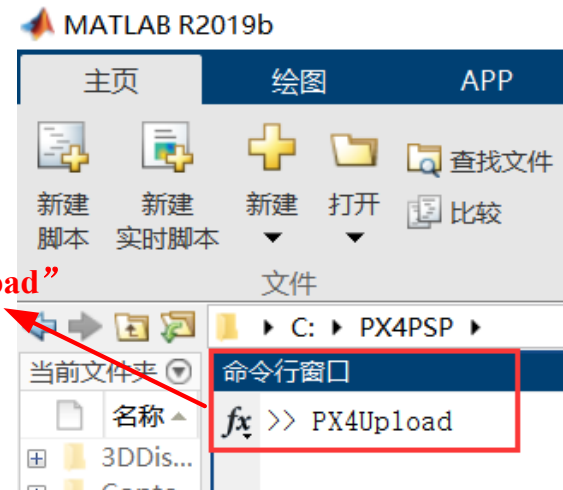
PSP工具箱

□ Simulink配置实现PSP工具箱

5) 固件的上传



(a) Simulink的“PX4Upload”按钮
(MATLAB 2017b~2019a)



(b) MATLAB的“PX4Upload”命令
(MATLAB 2019b及更高版本)

```
C:\WINDOWS\SYSTEM32\cmd.exe
Loaded firmware for 9,0, size: 879196 bytes, waiting for the bootloader...
If the board does not respond within 1-2 seconds, unplug and re-plug the USB connector.
PX4_SIMULINK = None
attempting reboot on COM3...
if the board does not respond, unplug and re-plug the USB connector.
attempting reboot on COM3...
if the board does not respond, unplug and re-plug the USB connector.
attempting reboot on COM3...
if the board does not respond, unplug and re-plug the USB connector.
Found board 9,0 bootloader rev 4 on COM3
50583400 00ac2600 00100000 00ffffff ffffffff ffffffff ffffffff 66ed47ff ff73cc15 c8ad940c dbc59f39 d6c20e06 f95
3d3ef f3073019 d035ab0d 3f60334e 10dda9f8 cdb0cbbd 42cdc6b6 3ba305f7 81532581 84e3da6 23bc6340 8321be68 edd356c9 1e3b8f
5c 5e07decc 9c6be5a2 458a1513 4b8bbbc21 eda35ce5 a8b840a5 ef019ca5 c89bb183 bb00f0c0 0c4b1a26 7375ff57 1ca41d94 24aa662e
ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff type: PX4
idtype: =00
vid: 000026ac
pid: 00000010
coa: Zu1H//9zzBXIrZQM28Wf0dbCDgb5U9Pv8wGdA1qw0/YDNOEN2p+M2wy71Czca206MF94FTJYGE7j2mI7xjQIMhvmjt01bJHjuPXr4H3syca+WiRYo
VE0u7vCHto1z1qLhApe8BnKXIm7GDuWdwwAbbG1Zzdf9XHKQd1CSqZi4=
sn: 0038001f3432470d31323533

Erase : [=====] 100.0%
Program: [=====] 100.0%
Verify : [=====] 100.0%
Rebooting.
```

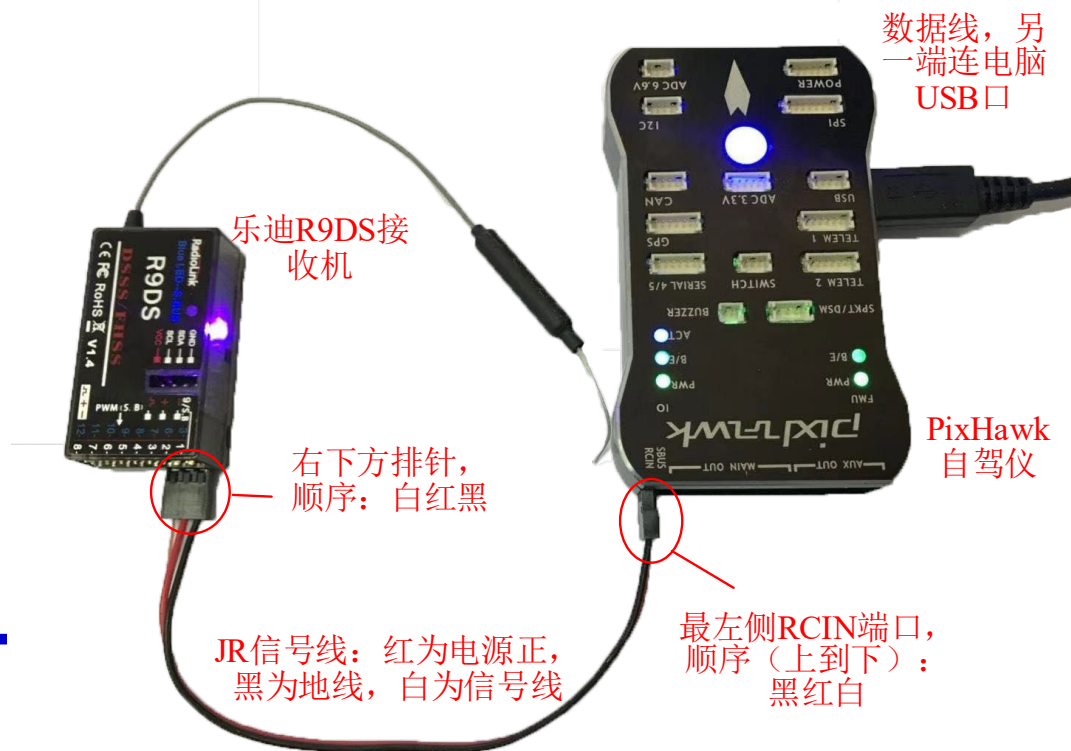
- **MicroUSB线将Pixhawk与电脑连接;**
- **对于MATLAB 2017b~2019a, 点击Simulink菜单栏【Code】-【PX4 PSP:Upload code to Px4FMU】将代码上传进入飞控; 对于MATLAB 2019b及更高版本, 在MATLAB的“命令行窗口”输入“PX4Upload”命令来下载固件。**
- **注意看窗口提示, 有时需要重新插拔Pixhawk才能上传并烧录。**



Pixhawk硬件系统

硬件系统组成与连线

如右图所示，自驾仪系统包括遥控器、接收机、JR信号线（连接Pixhawk与接收机）、Pixhawk自驾仪以及MicroUSB数据线（连接电脑与Pixhawk，为自驾仪供电并进行数据传输）。

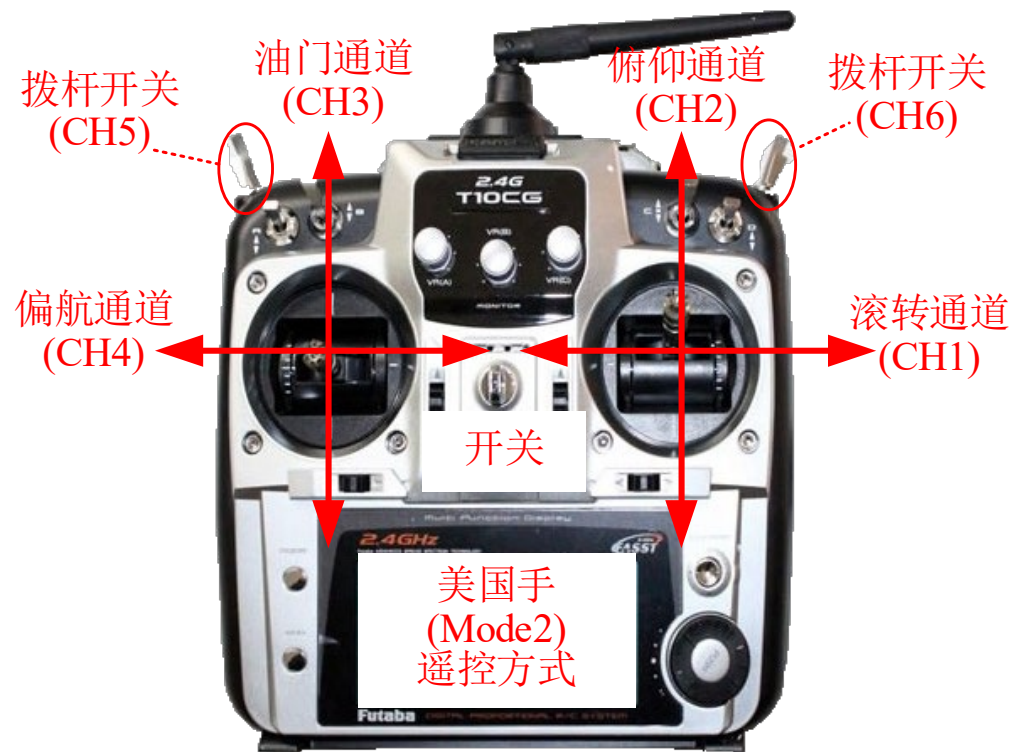




Pixhawk硬件系统

□ 遥控器基本操作方法

- 本课程使用的遥控器被设置成了“美国手”的操纵方式，即左侧摇杆对应的油门与偏航控制量，而右侧摇杆对应滚转与俯仰。
- 遥控器中滚转、俯仰、油门和偏航分别对应了接收机的CH1~CH4通道，左右上侧拨杆对应了CH5/CH6号通道，用于触发飞行模式切换。
- 油门杆（CH3通道）从最下端和最上端分别对应了PWM信号从1100到1900；滚转（CH1通道）和偏航（CH4通道）摇杆从最左端到最右端对应PWM信号从1100到1900；俯仰（CH2通道）摇杆从最下端到最上端对应PWM信号从1900到1100；CH5/6为三段开关，从顶部（最远离使用者的档位）到底部（最靠近使用者的档位）档位对应PWM信号为1100、1500和1900。



油门：控制上下运动，对应固定翼油门杆
偏航：控制机头转向，对应固定翼方向舵
俯仰：控制前后运动，对应固定翼升降舵
滚转：控制左右运动，对应固定翼副翼

图. 遥控器通道示意图



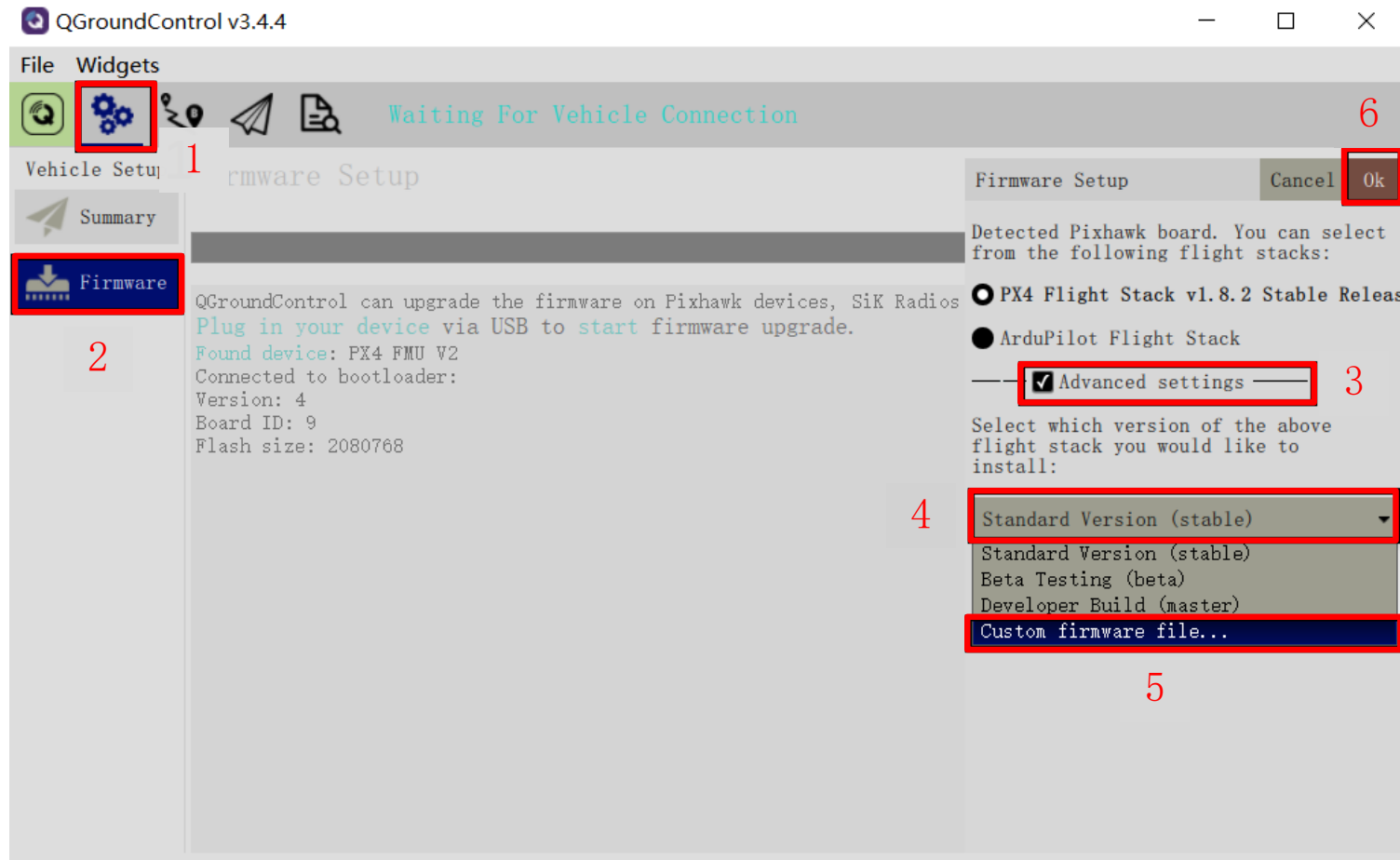
Pixhawk硬件系统



□ 地面站下载自驾仪固件方法

打开“QGroundControl”软件并进行下列操作：

- 1) 点击设置按钮（右图齿轮图标）；
- 2) 点击“Firmware”标签，此时用USB数据线连接Pixhawk自驾仪，地面站会自动检测自驾仪（注意：后续操作仅针对Pixhawk 1硬件，如果你使用的不是Pixhawk 1，这里直接点击“OK”使用官方最新固件即可，不需要进行后续步骤）
- 3) 勾选“advanced settings”选择框；
- 4) 点击“Standard Version (stable)”标签；





Pixhawk硬件系统

□ 地面站下载自驾仪固件方法

5) 在弹出标签页中选择

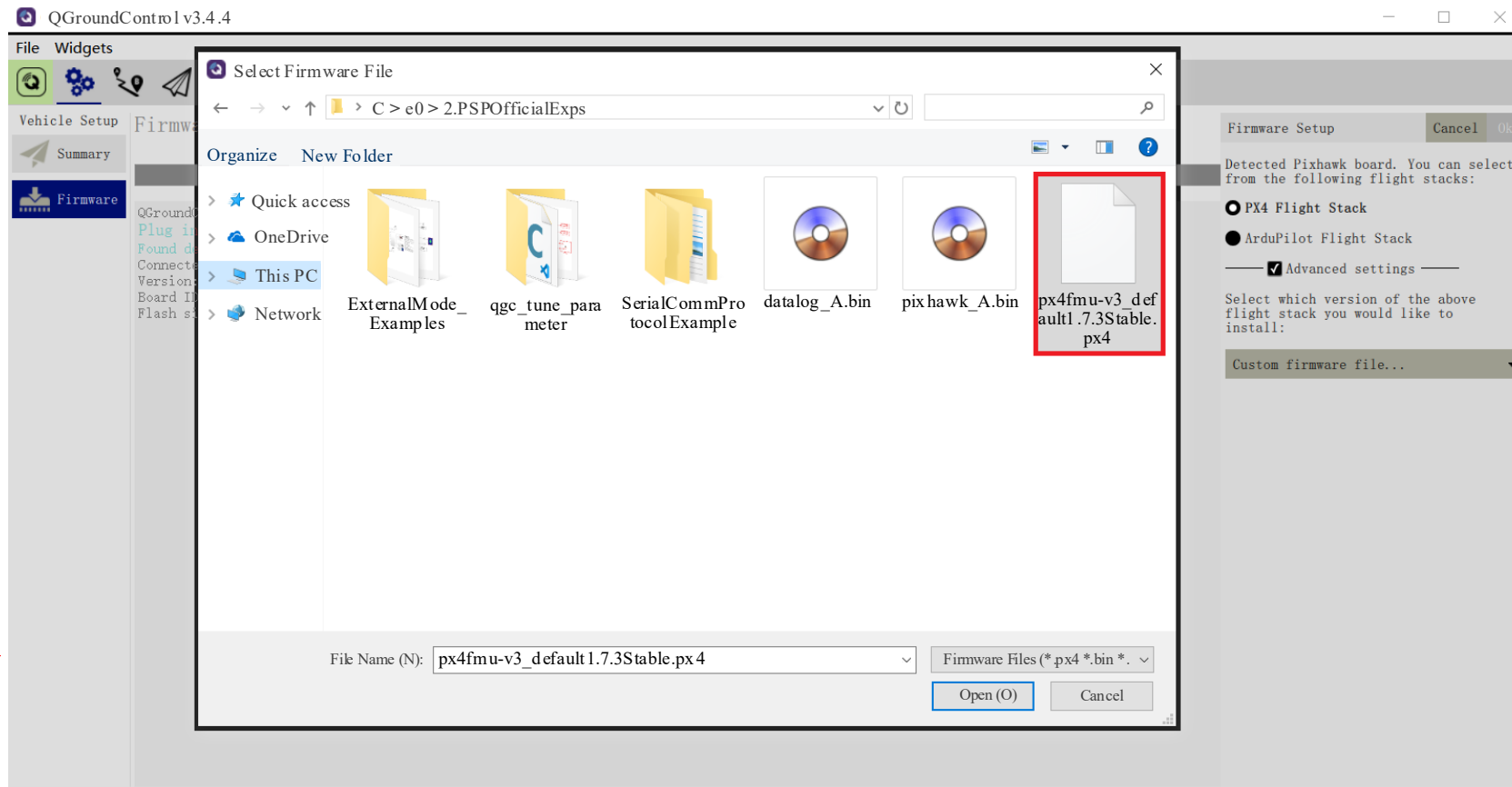
“Custom firmware file ..”选项;

6) 点击“OK”按钮;

7) 此时会弹出Windows的文件选择界面，选择

“\e0\2.PSPOfficial\ px4fm-u-v3_default1.7.3Stable.px4”文件，并点击“打开”按钮，此时地面站会将固件上传并烧录到Pixhawk自驾仪中。（注意：这里的px4固件问题只针对Pixhawk 1或Cube，对于其他Pixhawk硬件产品请从如下网址下载合适固件

<https://github.com/PX4/Firmware/releases/tag/v1.7.0>)

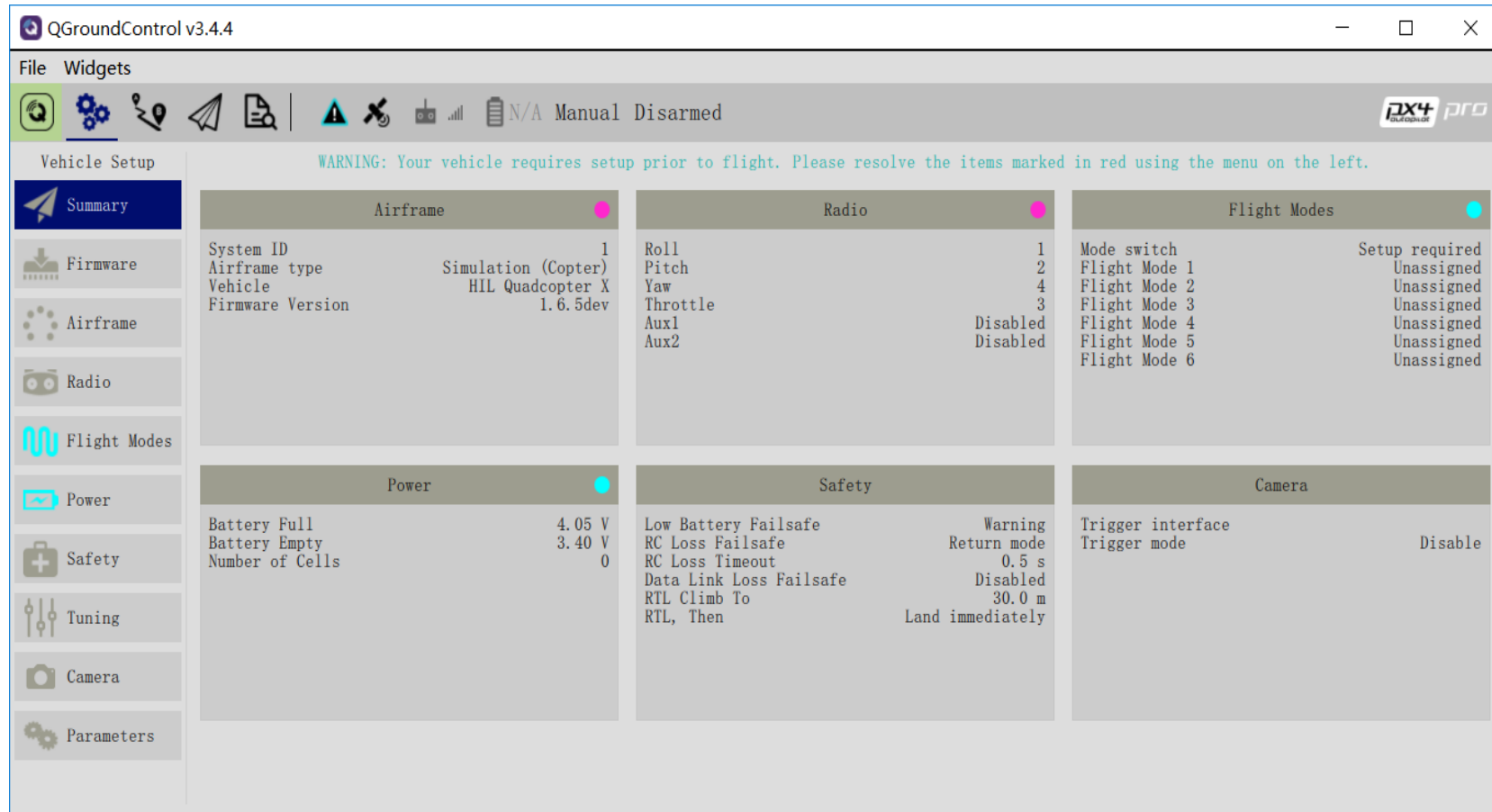




Pixhawk硬件系统

□ Pixhawk自驾仪半物理仿真模式设置

1) 打开QGroundControl地面站软件，然后用USB数据线连接到Pixhawk自驾仪，当地面站自动连接自驾仪时，可见到右上图所示界面，展示自驾仪基本信息。





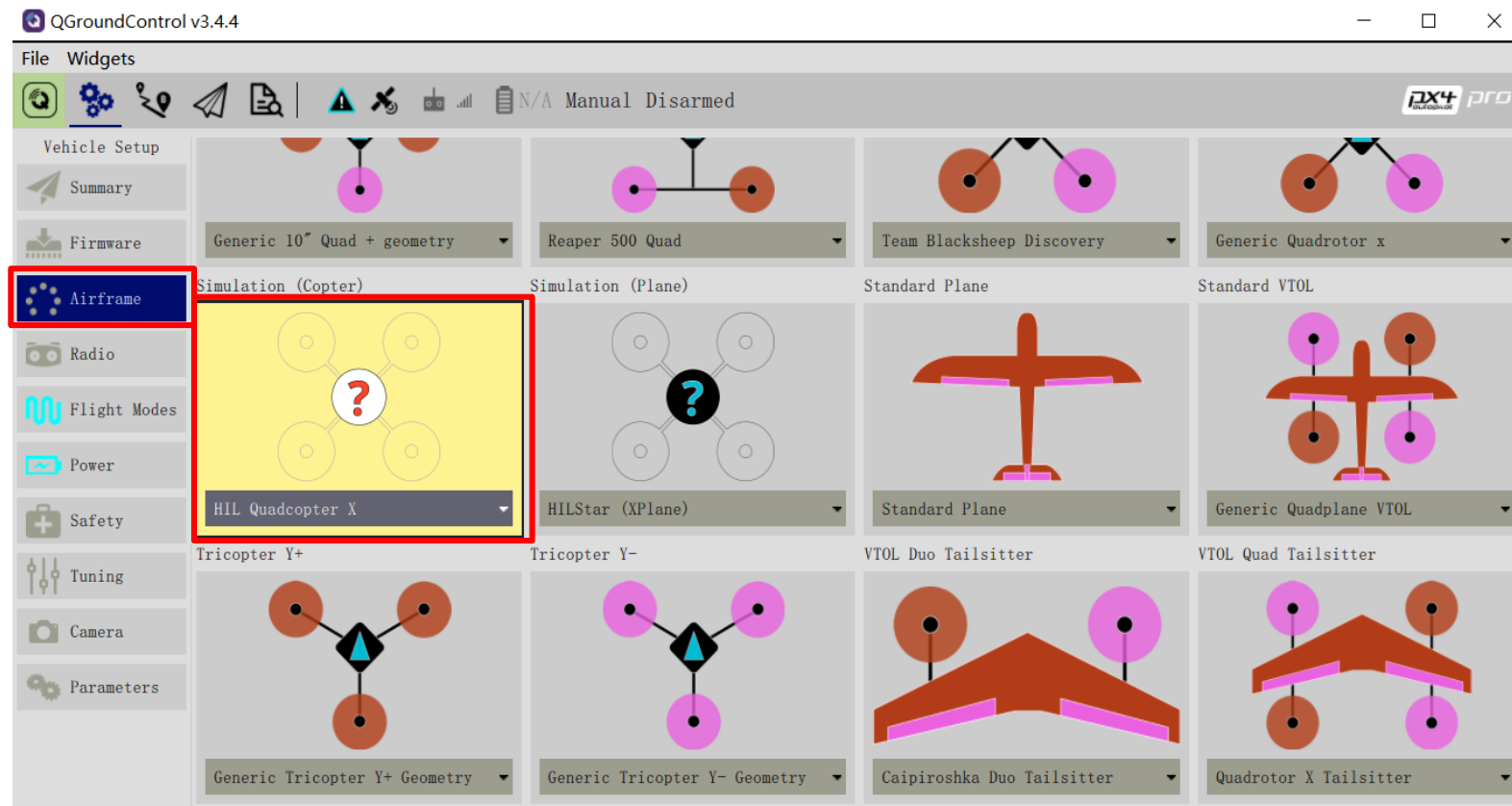
Pixhawk硬件系统

□ Pixhawk自驾仪半物理仿真模式设置

2) 如右下图所示，设置页面的

“Airframe”标签（左侧第3项），确认目前处于“HIL Quadcopter X”机架模式。这一项设置对后续的硬件在环仿真至关重要，如果不处于该模式需要手动设置。

设置方法：选中右下图中的“HIL Quadcopter X”机架图标，点击界面右上角的“Apply and Restart”按钮，自驾仪会进行设置并重新启动，重启完毕后QGroundControl会再次连接到自驾仪，此时查看是是否正确设置。





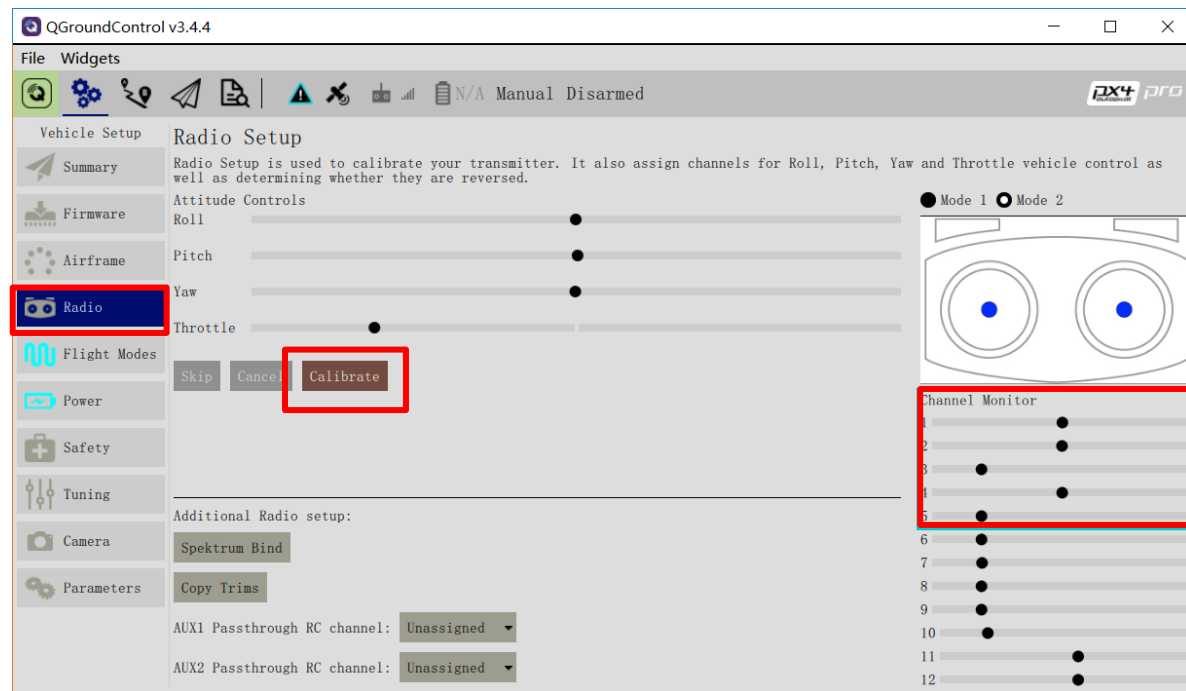
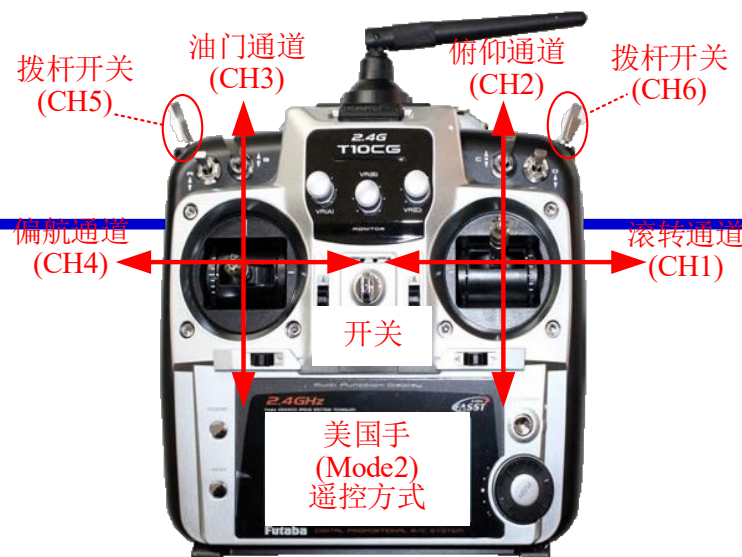
Pixhawk硬件系统

□ 遥控器配置与校准

1、正确连接自驾仪与接收机，用USB数据线连接自驾仪与电脑，打开遥控器，打开QGroundControl地面站软件，点击右下图所示的“Radio”标签页。

2、依次从左到右（或从上到下）拨动遥控器的CH1到CH5通道（见右上图），观测右下图地面站右侧红框区域中各个通道的白点。如果观测到：1、2、4、5、6号白点从左到右移动（PWM从1100到1900）；3号白点从右向左移动，说明遥控器设置正确。否则需要重新配置遥控器。

3、点击右下图的“Calibrate”按钮，按提示可以校准遥控器。





Pixhawk硬件系统

□ 遥控器配置与校准

4、点击QGC地面站上的“Calibrate” – “Next” 按钮，然后依次将摇杆置于右图所示位置（根据QGC页面的实时提示）即可完成遥控器校准。



油门：控制上下运动，对应固定翼油门杆
偏航：控制机头转向，对应固定翼方向舵
俯仰：控制前后运动，对应固定翼升降舵
滚转：控制左右运动，对应固定翼副翼

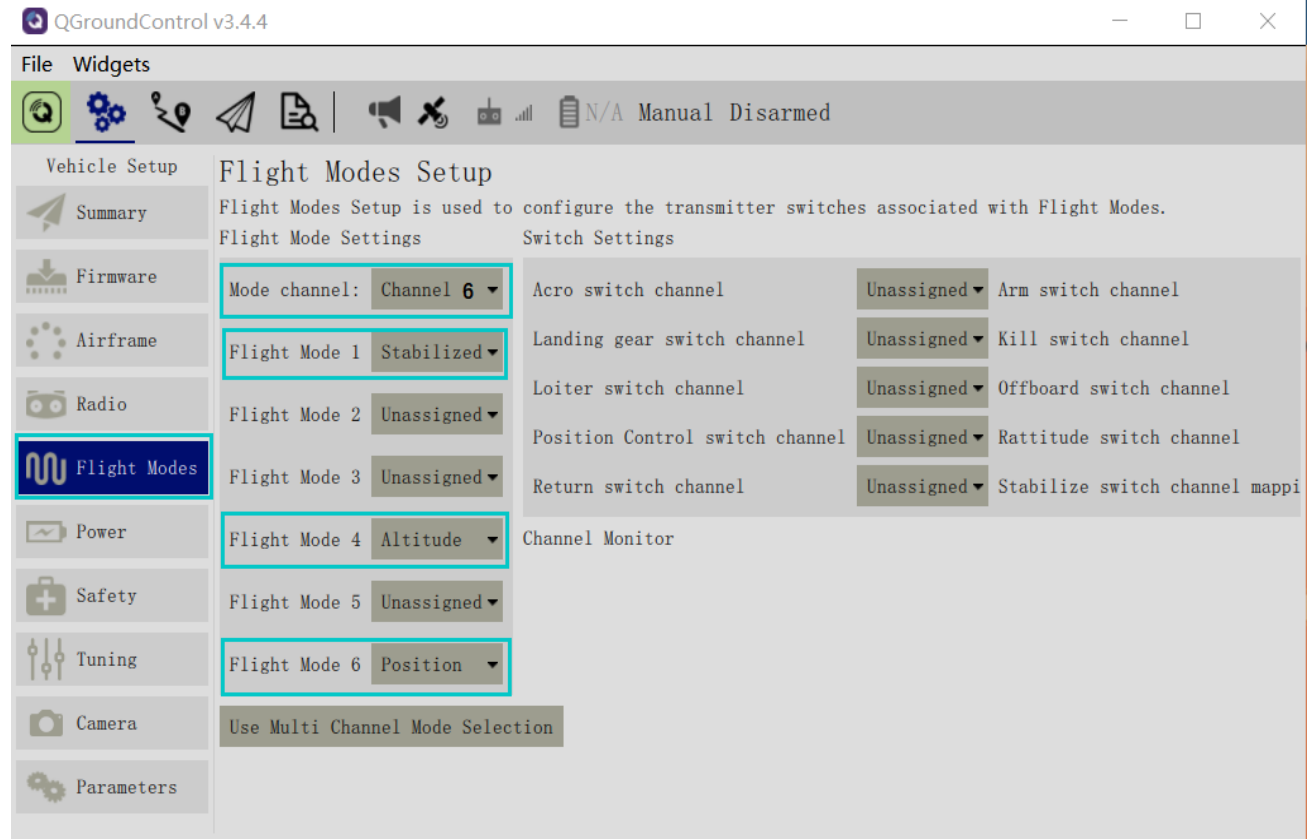


Pixhawk硬件系统

□ 自驾仪飞行模式设置

1、经过上面的遥控器校准步骤后，点击地面站进入“Flight Modes”（飞行模式）设置页面，选择“Mode Channel”（模式通道）为前面测试过的CH6通道。由于CH6通道是一个三段开关，开关的顶部、中部、下部档位分别对应了“Flight Mode 1、4、6”三个标签。

2、如右图所示，将这三个标签分别设置为“Stabilized”（自稳模式，只有姿态控制）、“Altitude”（定高模式，姿态和高度控制）和“Position”（定点模式，有姿态、定高和水平位置控制）。在后续的硬件在环仿真中，可以通过切换不同的模式来体验不同的控制效果。





硬件在环仿真器

□ 整体流程与功能

硬件在环多旋翼飞行器仿真器包括一个实时运动仿真软件——CopterSim和一个三维可视化视景软件——3DDisplay。下面，我们分别对这两个软件进行介绍。





硬件在环仿真器

□ CopterSim



1) 双击桌面的CopterSim快捷方式打开CopterSim软件，其界面如左图所示。CopterSim的仿真模型和参数与前文提到的Simulink多旋翼软件在环模型相同，因为CopterSim是通过Simulink多旋翼模型代码生成整合而成的。模拟器软件运行在Win64位电脑平台下，通过MicroUSB线与Pixhawk自驾仪进行串口通信。

2) 该界面分为两大部分，上半部分是读者自定义选择多旋翼模型的输入界面，下半部分是连接自驾仪进行仿真的界面。注意：本软件仅开放了本书所有实验需要的功能，还有许多实用的功能（多机仿真、UE4高逼真场景、固定翼等其他机型仿真等）需要注册之后才能使用。



硬件在环仿真器

□ CopterSim

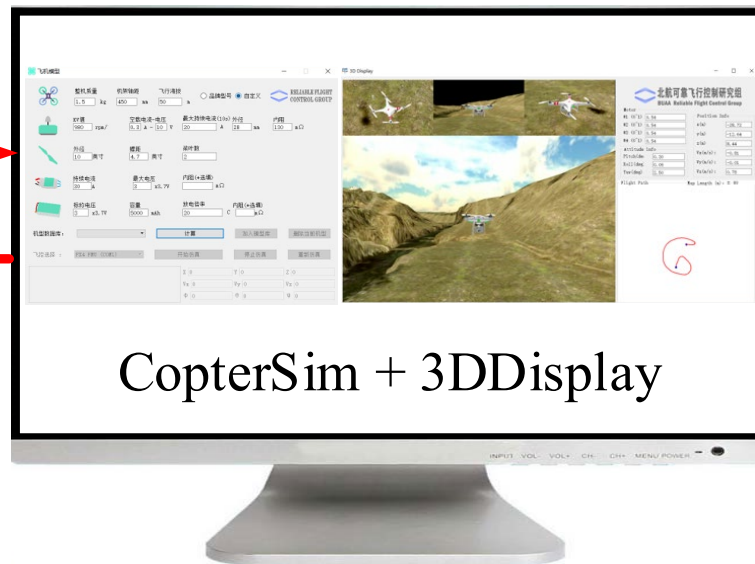


Pixhawk 自动驾驶系统

实时硬件在环仿真平台

控制
信号

传感器
数据



CopterSim + 3DDisplay

硬件在环仿真原理如下：

- CopterSim软件会发送传感器数据给Pixhawk自动驾驶仪，然后自动驾驶仪会解算得到的电机PWM控制信号并返回给CopterSim。
- 因此Pixhawk自动驾驶仪可以实时控制CopterSim中的模拟多旋翼，实现和真实飞行同样的效果。
- 与此同时，飞机的姿态与位置信息会以UDP协议向本地网络中发送，而3DDisplay软件会接收这些信息来完成实时三维视景显示。



硬件在环仿真器

机型数据库:

模型参与数管理页面

计算

模型参数

加入模型库

删除当前机型

□ CopterSim

1) 直接点击“模型参数”按钮，就会弹出左图所示模型参数页面，默认显示的是上一次仿真存储的模型参数。

2) 该界面主要包含了悬停信息（悬停时间、油门、输出功率、电机转速等）和飞机模型的基本参数（总质量、转动惯量、半径、拉力系数和阻力系数等）。

3) 点击该页面的“还原默认值”可以将模型参数还原到初始值；点击“存储并使用参数”按钮可以将当前参数存储到数据库中，作为本次和后续仿真的默认参数（重启软件参数也会保留，下次直接点击“开始仿真”按钮即可）。

4) “噪声水平（0~1）”按钮是控制模型噪声的系数，“0”是启用最低噪声，“1”是噪声水平与实际Pixhawk自驾仪传感器噪声一致。

多旋翼总质量: m	1.51	kg
重力加速度: g	9.8	m/s ²
转动惯量矩阵: Jxx	0.01492	kg.m ²
J=diag(Jxx, Jyy, Jzz):Jyy	0.01492	kg.m ²
:Jzz	0.02712	kg.m ²
多旋翼机身半径(1/2轴距): d	0.225	m
螺旋桨推力系数(Tp/ω ²):CT	1.276e-05	N/(rad/s) ⁴
螺旋桨力矩系数(Mp/ω ²):CM	2.25e-07	N.m/(rad/s) ⁴
油门(σ)到稳态转速(ωss):CR	716.91	rad/s
(ωss=CR*σ+ωb): ωb	158.17	rad/s
电机螺旋桨转动惯量: Jm	0.0001287	kg.m ²
电机响应时间常数: Tm	0.0571	s
空气阻力系数(D/v ²): Cd	0.075	N/(m/s) ²
旋转阻尼系数(M/ω ²): Cm	0.0103	N.m/(rad/s) ²

机架构型
旋翼数 4 机臂数 4 构型(x, +) x

悬停性能
悬停时间 17.9 min 电机输出功率 34.8 W
油门百分比 52.8 % 电池输出电压 11.8 V
电机电流 3.64 A 电池输出电流 15.1 A
电机转速 5124.7 rpm 能量效率 77 %

噪声水平(0-1): 0

还原默认参数 存储并使用参数 取消

图. 模型参数页面

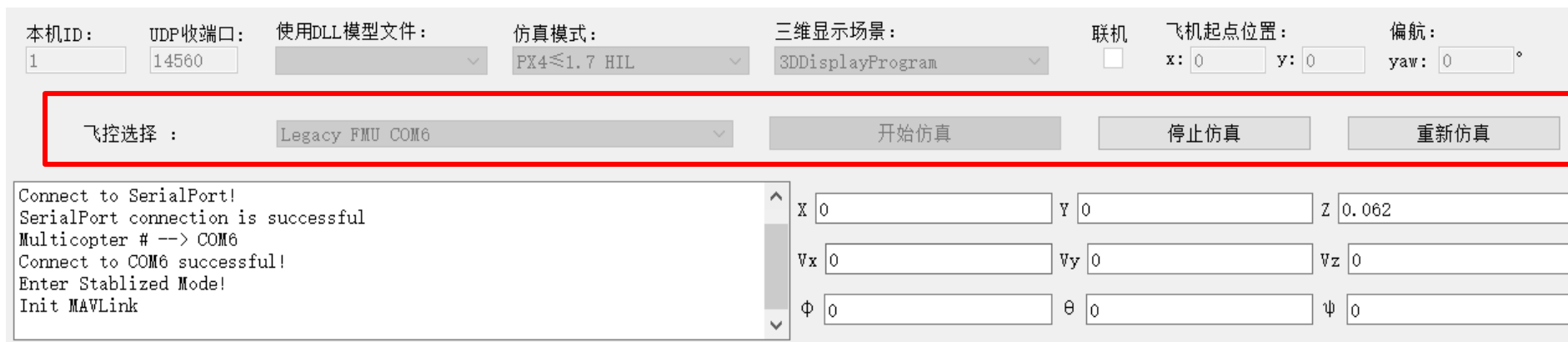


硬件在环仿真器

特别注意1: 如果使用硬件在环仿真，飞控连上电脑后需要等待10秒钟左右，等Pixhawk启动完毕后才能在CopterSim点击“开始仿真”按钮

特别注意2: CopterSim开始仿真后，需要等待消息栏显示“** EKF initialization finished”后，才能使用使用遥控器解锁并控制飞机

□ CopterSim



当多旋翼配置并且计算完成，将Pixhawk插入到电脑的USB口上，在主界面的“飞控选择”下拉菜单中就会出现可连接的Pixhawk自驾仪串口。选中需要的Pixhawk自驾仪串口（通常名字中包含字符串“FMU”），点击“开始仿真”按钮，就可以开始硬件在环仿真。如上图所示，CopterSim可以接收到Pixhawk回传的消息，说明硬件仿真正常运行。在仿真过程中也可以点击“停止仿真”来停止硬件在环仿真，或者点击“重新仿真”来初始化多旋翼到原始位置。



硬件在环仿真器



3DDisplay.exe

□ 3DDisplay



图.三维显示器程序主界面

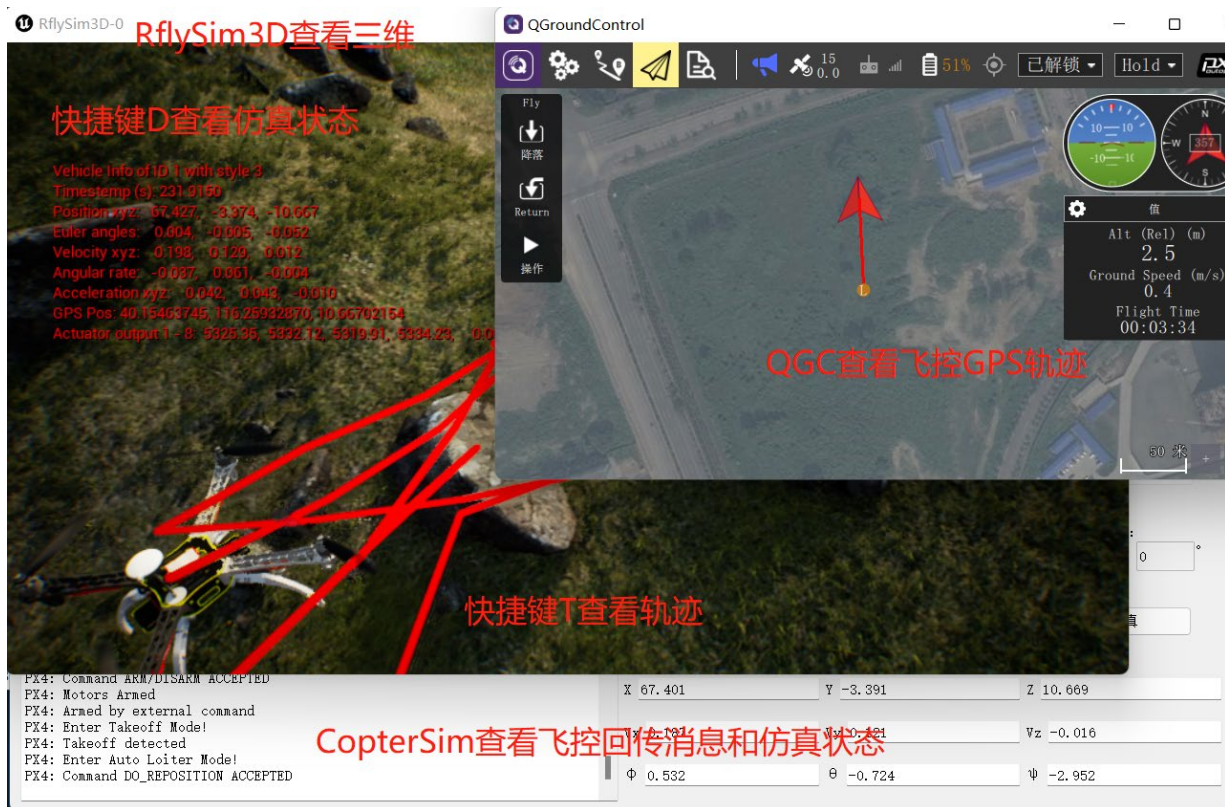
双击桌面的3DDisplay快捷方式即可打开3DDisplay软件。如左图所示，3DDisplay主界面窗口左侧以3D图形的方式展示多旋翼当前的飞行状态；3DDisplay主界面右上角窗口展示了基本的飞行数据，包括电机转速，位置信息，姿态信息；3DDisplay主界面右下角窗口显示多旋翼的飞行轨迹。



硬件在环仿真器

注意1：由于性能和兼容性问题，RflySim高级版中已经取消3DDisplay软件，全面使用RflySim3D来进行三维场景的观察

□ RflySim3D（针对高级版）



双击桌面的RflySim3D快捷方式即可打开基于UE4开发的三维场景软件。如左图所示，在主界面中可以观察到非常逼真的飞机三维模型，同时按下快捷键“D”可以预览当前飞机的状态数据，按下快捷键“T”可以记录三维轨迹，同时在QGroundControl地面站中可以实时观察飞控的数据与轨迹等。

注意：更多更强大的功能请参考高级版的使用文档
<https://doc.rflysim.com>

图.三维显示器程序主界面

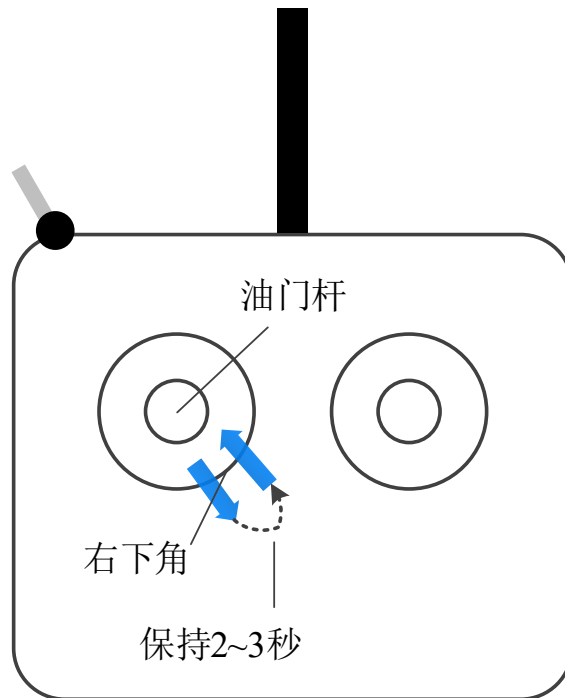


硬件在环仿真器

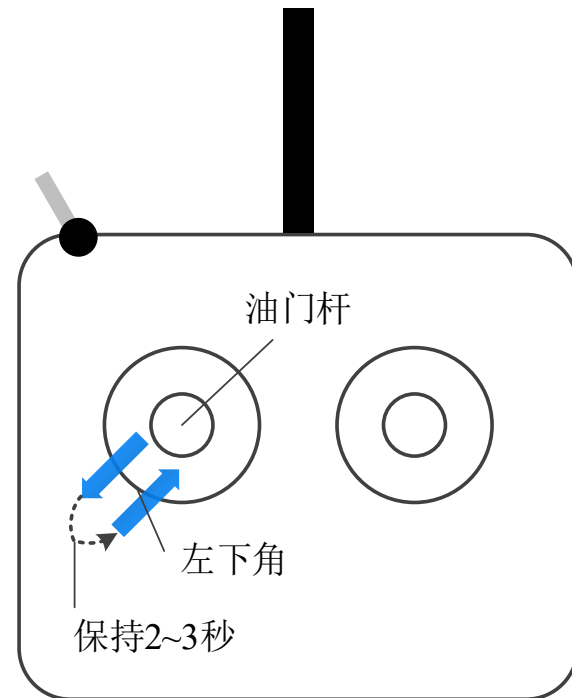
□ 硬件在环飞行测试

在硬件在环仿真中，可以和控制真实飞机一样，通过遥控器仿真的多旋翼完成解锁、起飞、手动飞行、降落等动作。具体步骤如下：

1. 首先将遥控器电源开关推上去，打开遥控器；
2. 按上文步骤正确连接Pixhawk软硬件，并通过CopterSim开始硬件在环仿真；
3. 如图 (a)所示，将遥控器左侧油门摇杆置于右下角，保持2~3秒来解锁Pixhawk自驾仪；



(a) 解锁动作



(b) 上锁动作

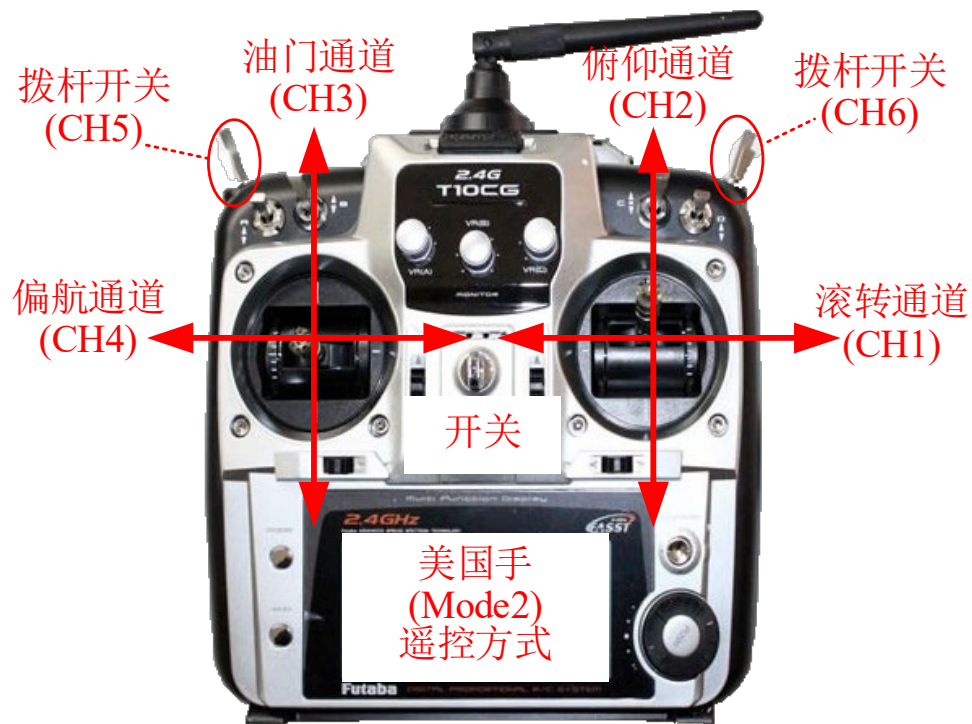
4. 此时可以看到Pixhawk上的LED灯变为常亮高版本Pixhawk硬件（例如FMUV3）开始不自带LED灯模块，需要使用外置I2C接口LED灯观察效果，且CopterSim左下角的消息框收到“Detect Px4 Armed”消息，说明解锁成功。如果解锁不成功，则断开所有软硬件连接，重复上述步骤；



硬件在环仿真器

□ 硬件在环飞行测试

5. 向上拨遥控器左侧的操纵杆，使多旋翼起飞到一定高度，然后上下拨动油门杆，确认多旋翼的上下运动控制功能；
6. 左右拨动遥控器左侧的操纵杆，确认多旋翼的偏航方向转动控制功能；
7. 上下拨动遥控器右侧操纵杆，控制多旋翼俯仰角的大小，确认多旋翼的前后运动控制功能；
8. 左右拨动遥控器右侧操纵杆，控制多旋翼的滚转角的大小，确认实现多旋翼的左右运动功能；
9. 拨动遥控器右上角的三段开关，确认多旋翼的控制模式切换功能；



10. 向下拨遥控器左侧的油门摇杆，使多旋翼降落在地面；
11. 将遥控器左侧油门摇杆置于左下角，保持2~3秒来锁定Pixhawk自驾仪；
12. 在CopterSim中点击“结束仿真”按钮，退出HIL仿真，然后断开Pixhawk与电脑硬件连接。



小结

- (1) 本讲主要介绍了RflySim基础版平台的基本结构和使用方法，包括：实验平台总体介绍、控制器设计与仿真平台、PSP自动代码生成工具箱、Pixhawk硬件系统、硬件在环多旋翼飞行器仿真器等
- (2) 本讲的几个实际操作的小例子有助于掌握平台的基本操作和多旋翼飞行器的基本操作方法，为后续课程的学习打下坚实基础。
- (3) 如有疑问，请到 <https://rflysim.com> 查询更多信息。



资源

- 本课程的所有课件、视频和源码将会发布在官方网站：<https://rflysim.com/zh>
- 更详细的内容可以参考我们的教材：全权, 戴训华, 王帅著. 《多旋翼飞行器设计与控制实践》. 北京: 电子工业出版社. 2020
- 扫二维码关注可靠飞行控制研究组公众号 **buarfly** (文章、资讯等)
- 多旋翼控制实践课程交流QQ群: **951534390** (答疑、资料分享等)
- 如果遇到任何问题, 也可以在我们的Github页面查找答案或提问
<https://github.com/RflySim/RflyExpCode/issues>
- 针对无人机/无人车的视觉/集群/AI等顶层控制算法的快速开发与验证, RflySim同时推出了高级版平台和课程, 请访问我们的网站了解更多
https://rflysim.com/zh/4_Pro/Advanced.html
- 我们的官方联系邮箱是: rflysim@163.com





谢谢!