多旋翼飞行器设计与控制
系列实验

第四讲 实验流程介绍

戴训华 博士

dai@buaa.edu.cn

自动化科学与电气工程学院

北京航空航天大学

大纲

1. 实验流程总体介绍
2. 控制LED灯实验操作具体流程
3. 姿态控制实验操作具体流程
4. 小结

实验流程总体介绍

- 本课程包含动力系统设计、建模、估计、控制和决策相关任务。
- 每个任务分为由浅入深的三个实验，即基础实验、分析实验和设计实验。

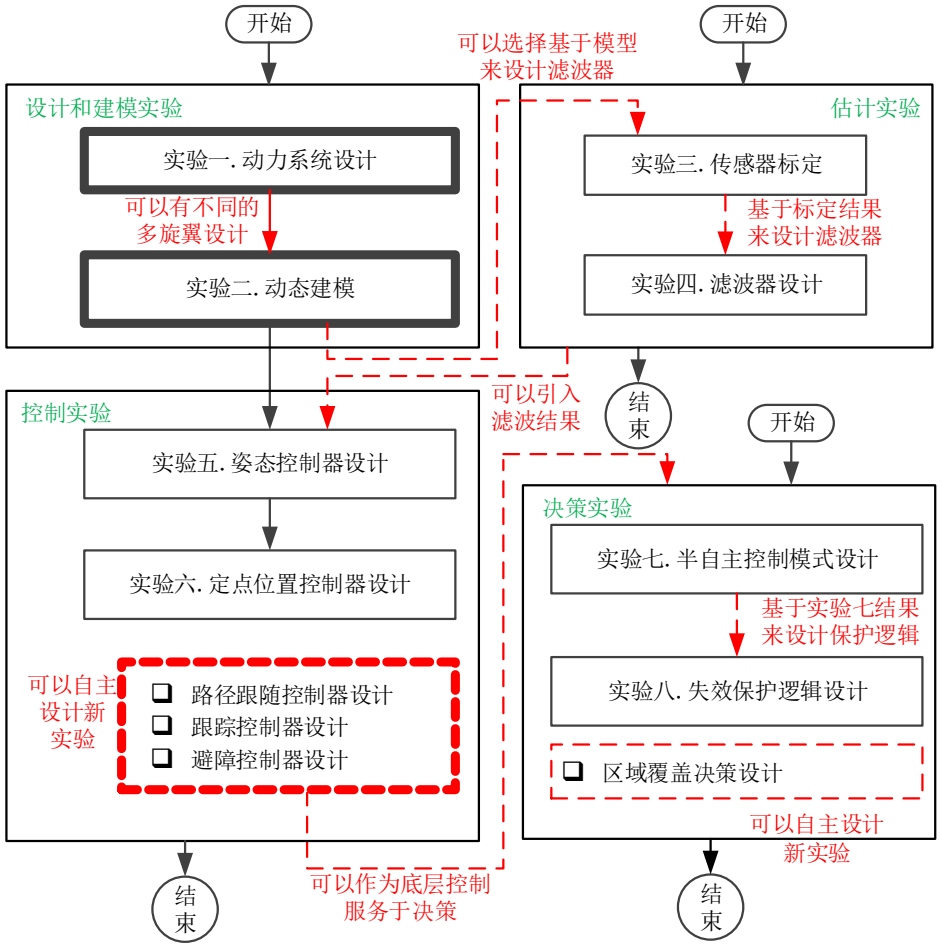


图. 实验关系图

实验流程总体介绍

（1）基础实验

打开例程，阅读并运行程序代码，然后观察、记录结果并分析数据。

（2）分析实验

指导读者修改例程，运行修改后的程序并收集和分析数据。

（3）设计实验

在上述两个实验完成的基础上，针对给定的任务，进行独立设计。模型或控制器将会被自行设计的模块替代。

对于基础实验和分析实验，本书会提供完整的例程，以此保证所有的读者都可以顺利完成实验。通过以上两个分步实验，读者能较好地了解实验课程的理论和其应用方法。在最后的设计实验中，读者只需在原有的架构上逐一替换自己的设计。整个过程由浅入深，便于一步一步达到最终的实验目标。

实验流程总体介绍

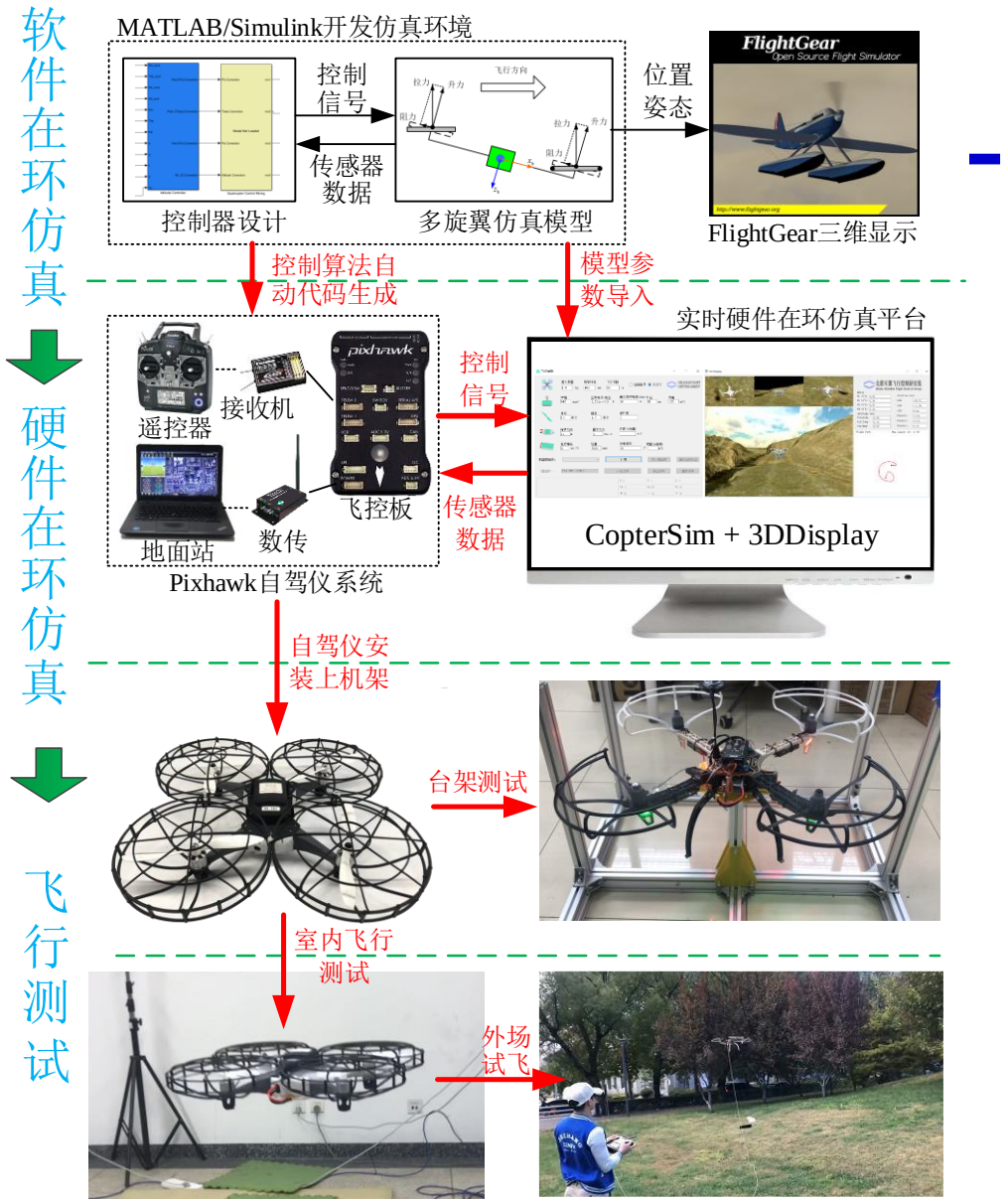
每个实验有一般包括以下三个阶段：

（1）Simulink算法设计与软件在环仿真阶段

（2）硬件在环仿真阶段

（3）飞行测试阶段

（考虑实际飞行可能带来的风险，实际飞行可以不放在本课程的实验内容中）。



实验流程总体介绍

（1）软件在环仿真阶段

整个阶段都在MATLAB环境下进行，利用给定多旋翼仿真模型和例程，在Simulink中进行控制算法设计，并正确连接模型和控制器，确保输入输出信号与实际多旋翼系统一致。类似于实际多旋翼系统，多旋翼模型将传感器数据或状态估计信息（例如，姿态角、角速率、位置和速度等）发送给控制器，控制器将每个电机PWM控制指令发回给模型，从而形成一个软件在环仿真闭环系统。在本阶段，读者可以观察控制性能，自行修改或设计控制器来达到期望的性能需求。

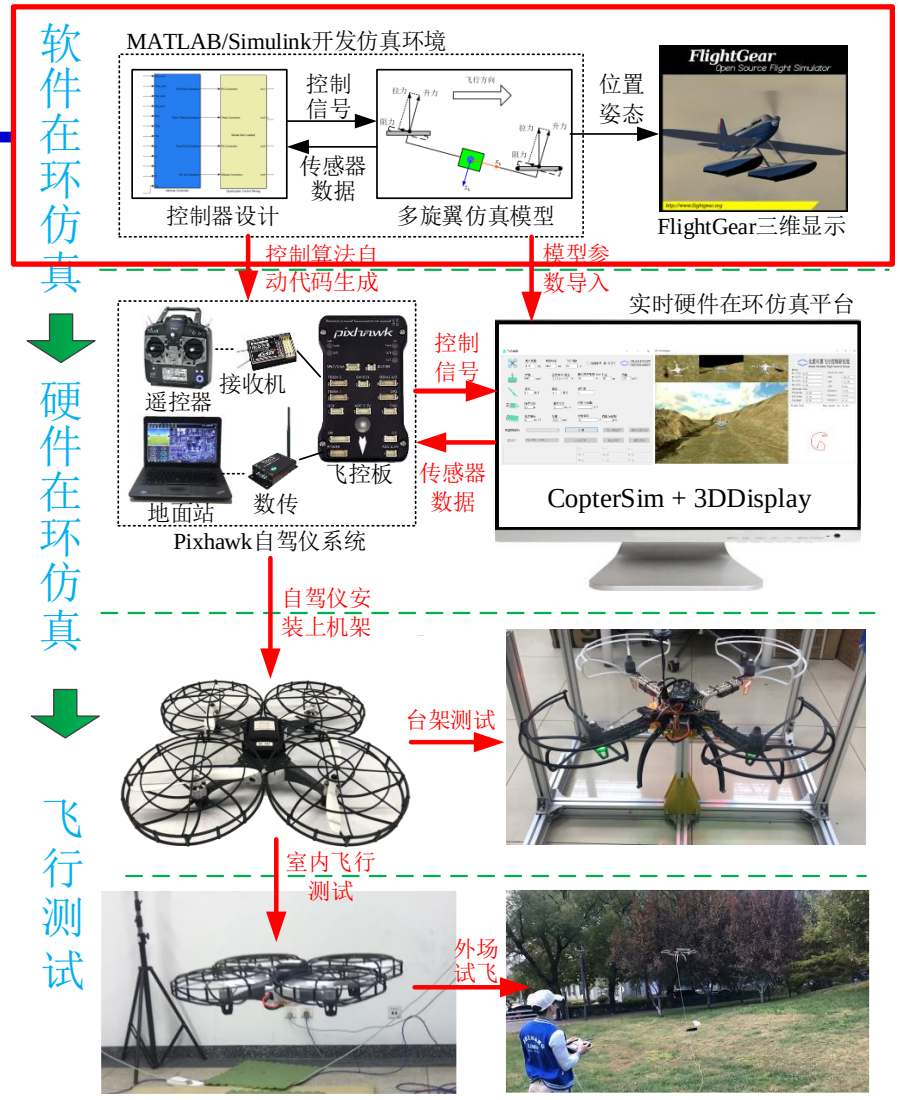


图. 实验流程图

实验流程总体介绍

（2）硬件在环仿真阶段

利用给定的模型和例程，进行实验。模型在硬件在环多旋翼飞行器仿真器里，而控制器上传到Pixhawk飞控硬件环境下，其中通讯过程是通过串口线直接连接。模型通过串口线将姿态角、姿态角速率、位置和速度发送给控制器，控制器通过串口线将每个电机PWM控制指令发回给模型，从而形成一个闭环。

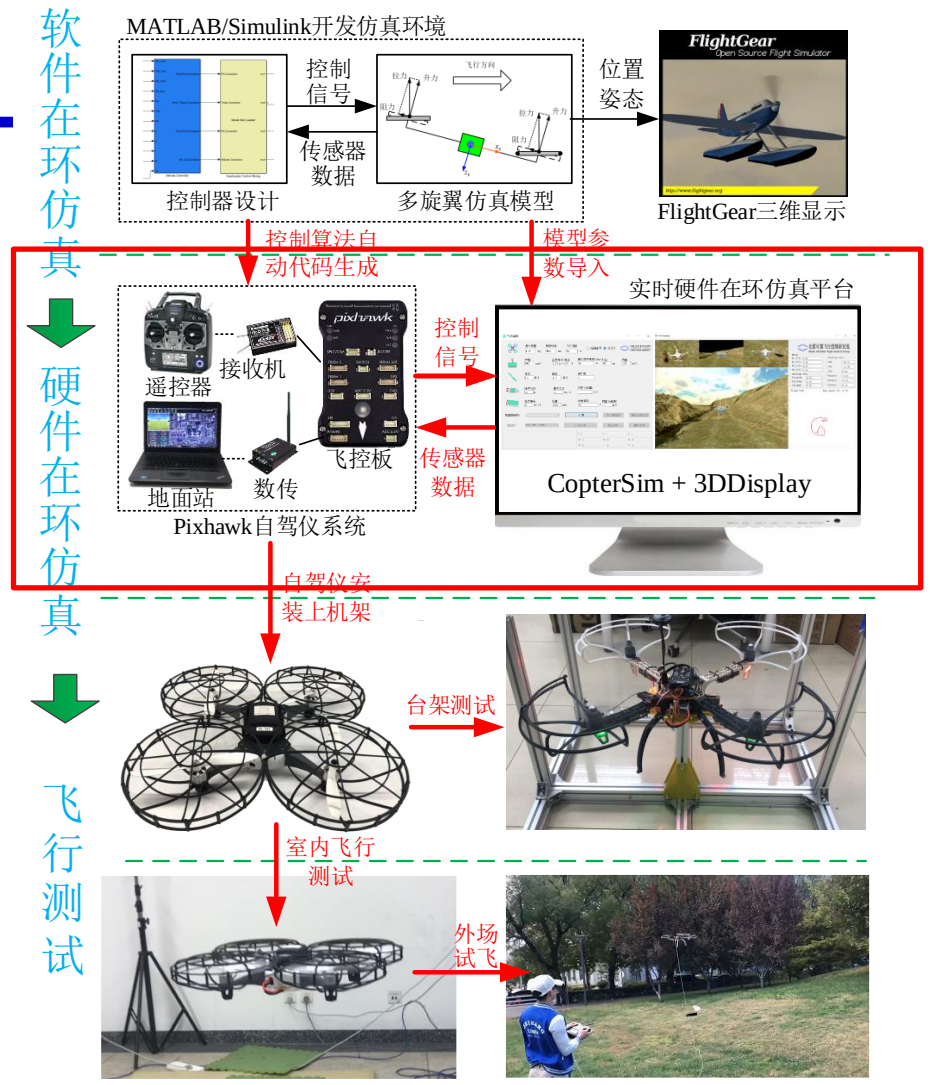


图. 实验流程图

实验流程总体介绍

（2）硬件在环仿真阶段

将Simulink多旋翼模型参数导入到CopterSim中，并将Simulink控制器算法生成代码下载到Pixhawk自驾仪，然后用USB实体信号线替代Simulink中的虚拟信号线。CopterSim将传感器数据（例如，加速度计、气压计、磁力计等）通过USB数据线发送给Pixhawk系统；Pixhawk系统中的PX4自驾仪软件将收到传感器数据进行滤波和状态估计，将估计的状态信息通过内部的uORB消息总线发送给控制器；控制器再通过USB数据线将每个电机的PWM控制指令发回给CopterSim，从而形成一个硬件在环仿真闭环。

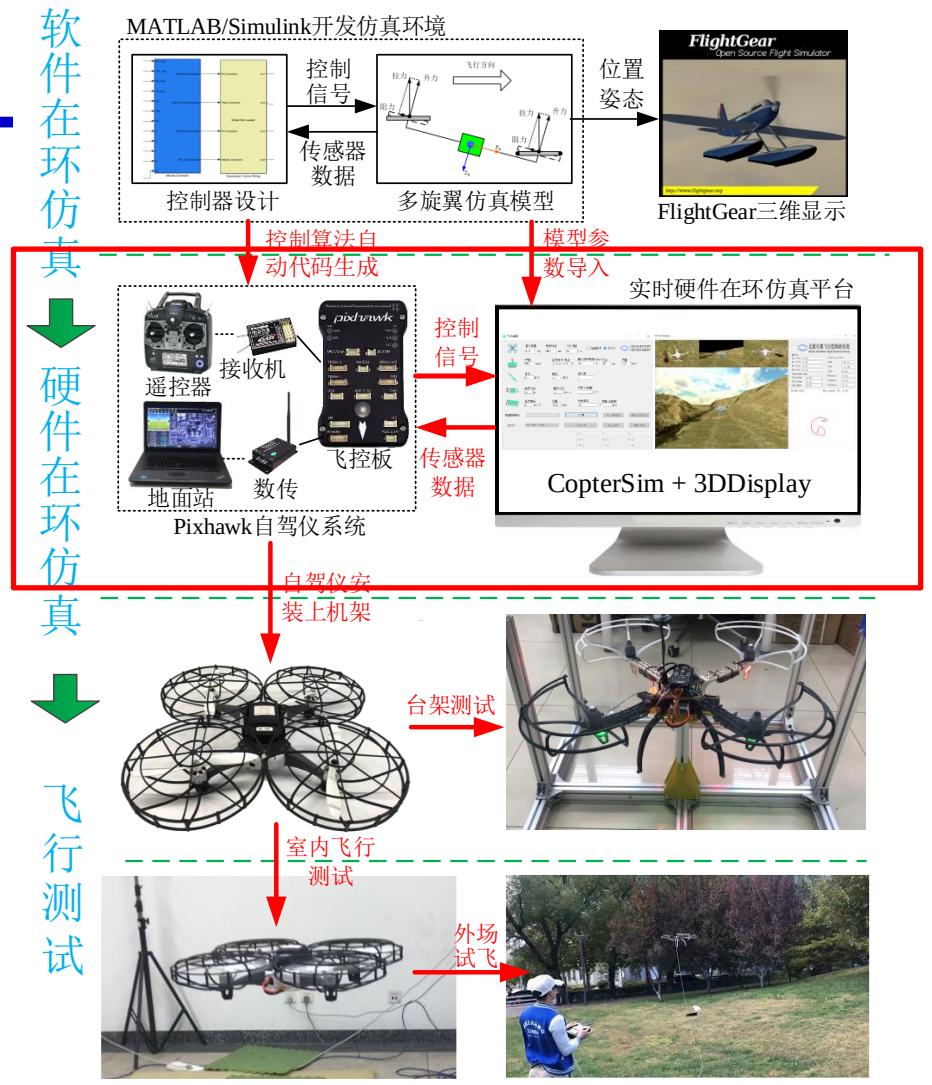


图. 实验流程图

实验流程总体介绍

（2）硬件在环仿真阶段

相对于软件在环仿真，硬件在环仿真中多旋翼模型运行速度与实际时钟是一致的，以此保证仿真的实时性，同时控制算法可以部署并运行在真实的嵌入式系统中，更加接近实际多旋翼系统。需要注意的是，实际硬件通讯中可能会存在传输延迟，同时硬件在环系统的仿真模型和控制器所运行环境也难免与软件在环系统存在一定差异，因此控制器的参数可能需要进一步调节来达到设计需求，这也恰恰反映实际中的情况。

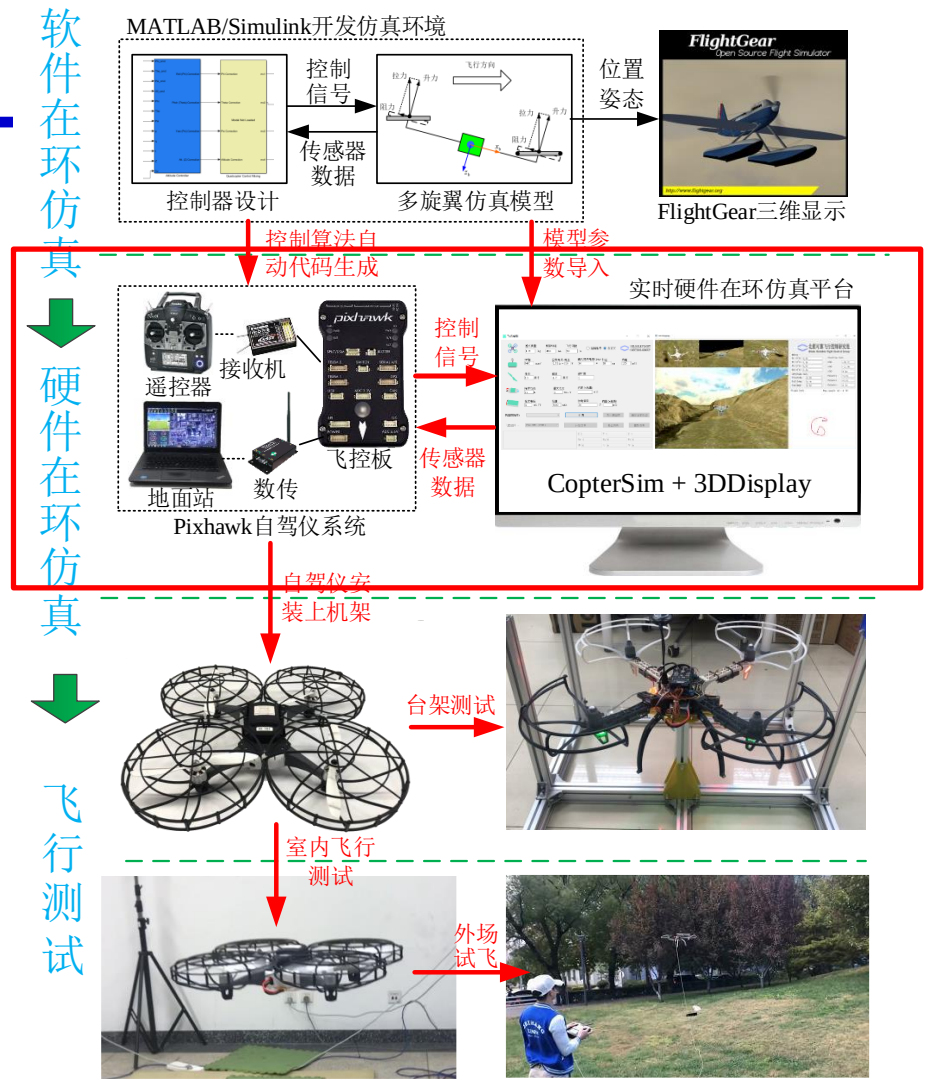
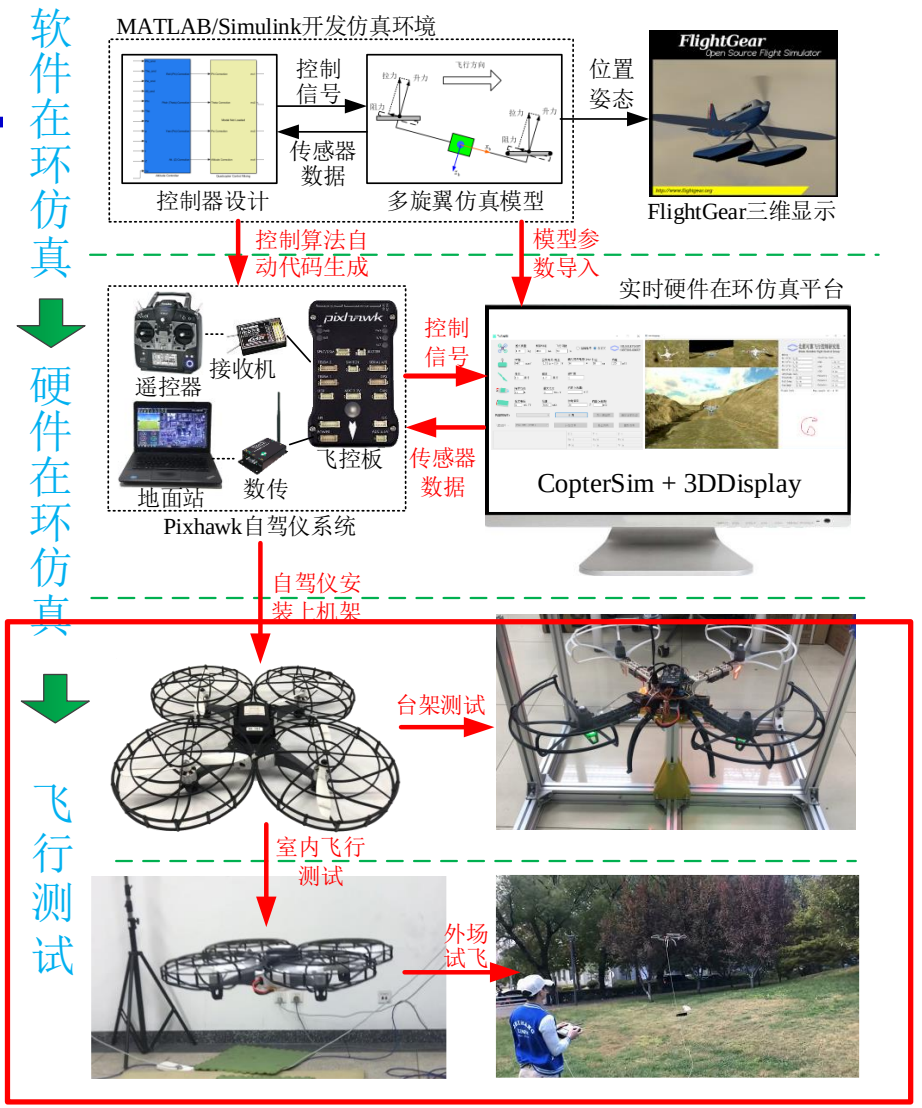


图. 实验流程图

实验流程总体介绍

（3）飞行测试阶段

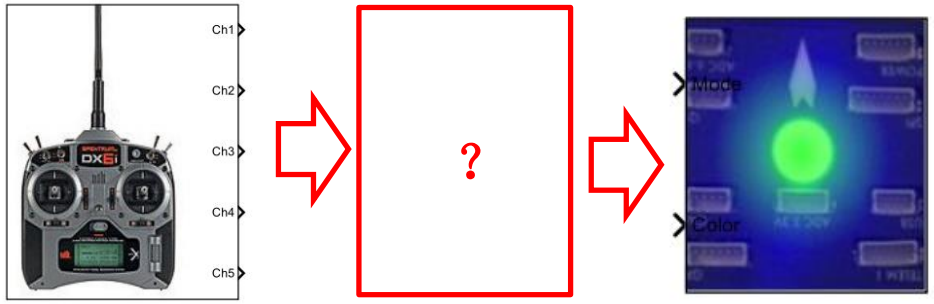
在这个阶段，CopterSim的虚拟仿真模型进一步由真实多旋翼飞行器替代，传感器数据直接由传感器芯片感知飞行运动状态得到，控制器信号直接输出给电机，从而实现真实飞机的控制。需要注意的是，无论是硬件在环仿真还是软件在环仿真，其仿真模型都难以与真实飞机保持完全一致，因此进一步的参数调整也是必要的。

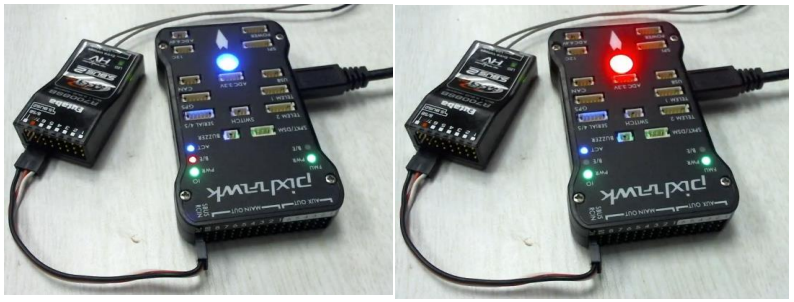


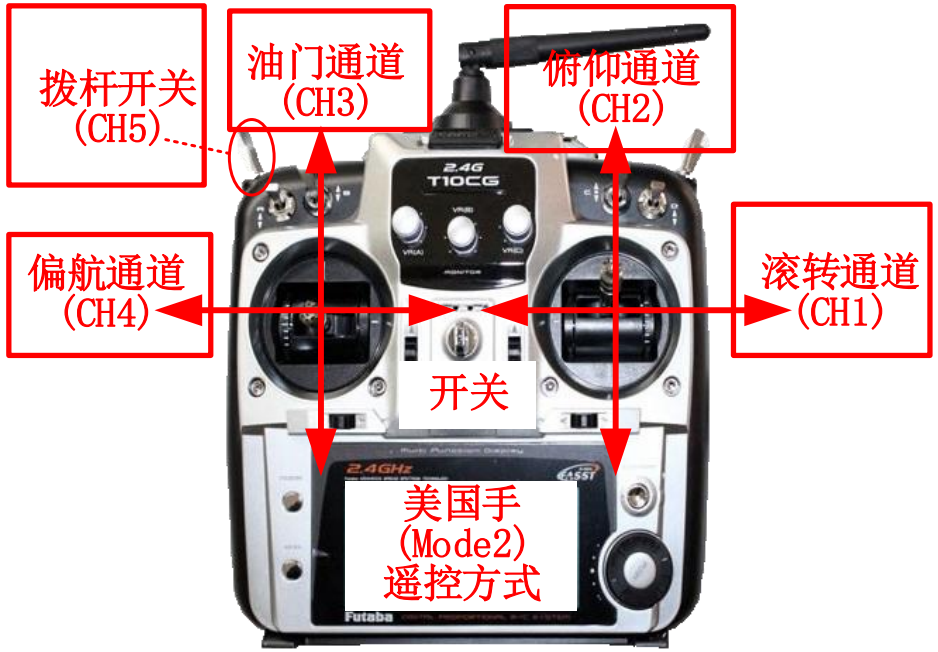
控制LED灯实验操作具体流程

□ LED灯实验目标

使用遥控器CH1~CH5任意两个通道实现让LED灯以两种颜色和两种模式闪烁。







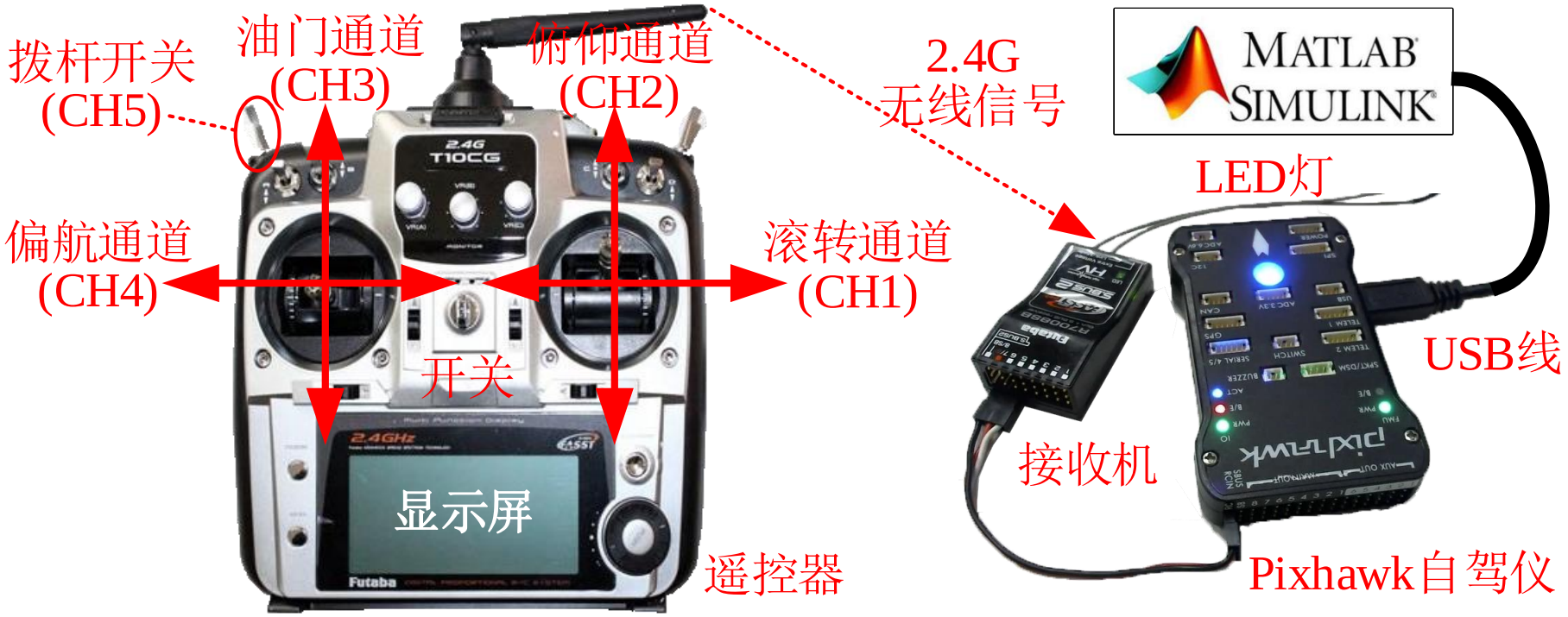
油门：控制上下运动，对应固定翼油门杆
偏航：控制机头转向，对应固定翼方向舵
俯仰：控制前后运动，对应固定翼升降舵
滚转：控制左右运动，对应固定翼副翼

控制LED灯实验操作具体流程

□ LED灯实验目标

使用遥控器CH1~CH5任意两个通道实现让LED灯以两种颜色和两种模式闪烁。

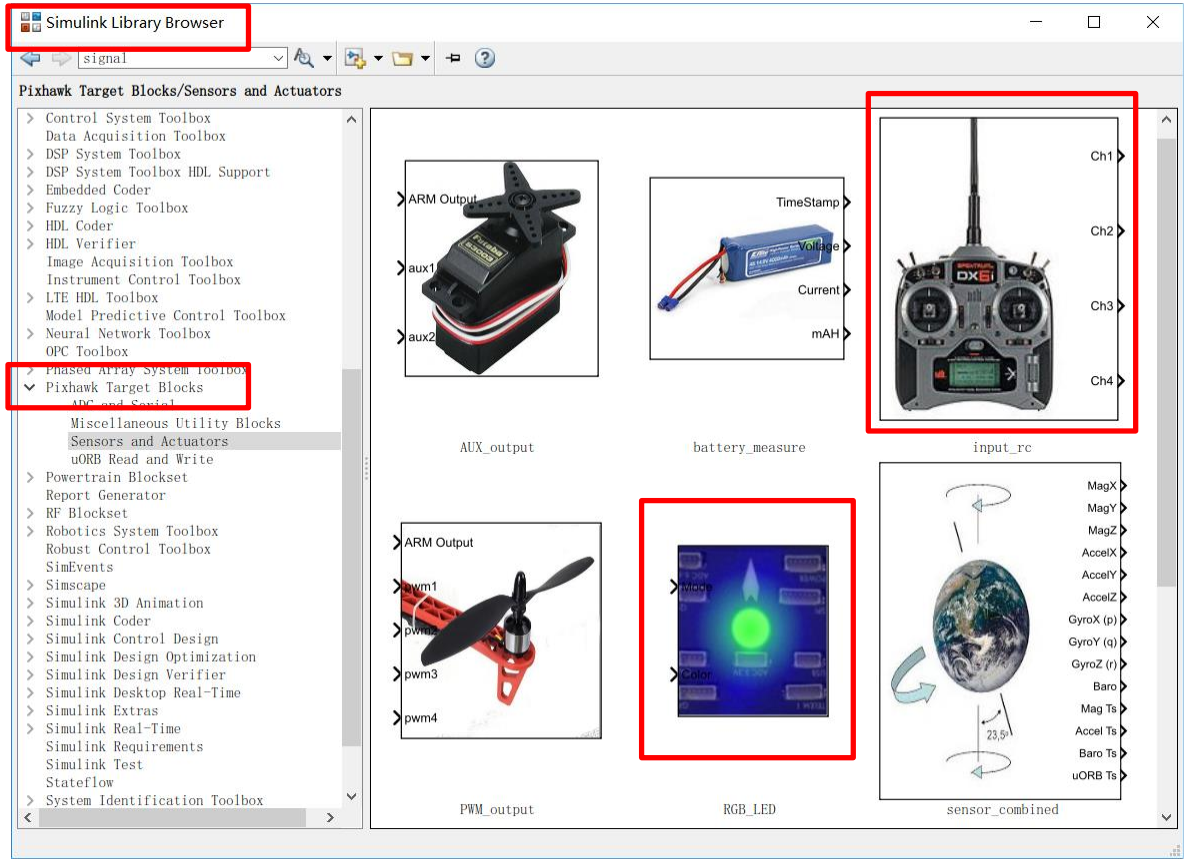
硬件连接图



控制LED灯实验操作具体流程

□ 设计LED灯控制模型

（1）新建一个Simulink模型文件，在Simulink“Library Browser”库的PSP工具箱中找到“RGB_LED”模块并添加到新建的模型文件中。因为还需用到遥控器来控制LED灯，所以把遥控器模块“input_rc”也添加到模型中。注：这里也提供一个可供参考的的Simulink例程，详见“e0\2.PSPOfficialExps\px4demo_input_rc.slx”。



控制LED灯实验操作具体流程

□ 设计LED灯控制模型

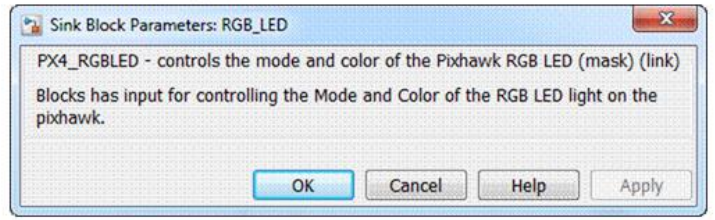
（2）查看“RGB_LED”模块说明。双击“RGB_LED”模块，选择help查看预定义的控制枚举变量。这个模块可以用来控制Pixhawk上LED灯的模式和颜色。

注：右侧Help文档中的枚举变量在安装PSP工具箱的时候已经在MATLAB全局参数中注册了，因此可以直接调用。例如在模块的“Mode”口用“Constant”模块输入“RGBLED_MODE_ENUM.MODE_BLINK_FAST”。

Pixhawk Target Block: RGB_LED

This block gives the user control over various lighting modes of the RGB LED available on the PX4 hardware.

RGB_LED



This block accepts 2 inputs: Mode and Color. These are enumeration data types. You can find out what values are valid in the MATLAB command window by typing:

RGBLED_MODE_ENUM

Value
SL_MODE_OFF (0)
SL_MODE_ON (1)
SL_MODE_DISABLED (2)
SL_MODE_BLINK_SLOW (3)
SL_MODE_BLINK_NORMAL (4)
SL_MODE_BLINK_FAST (5)
SL_MODE_BREATHE (6)

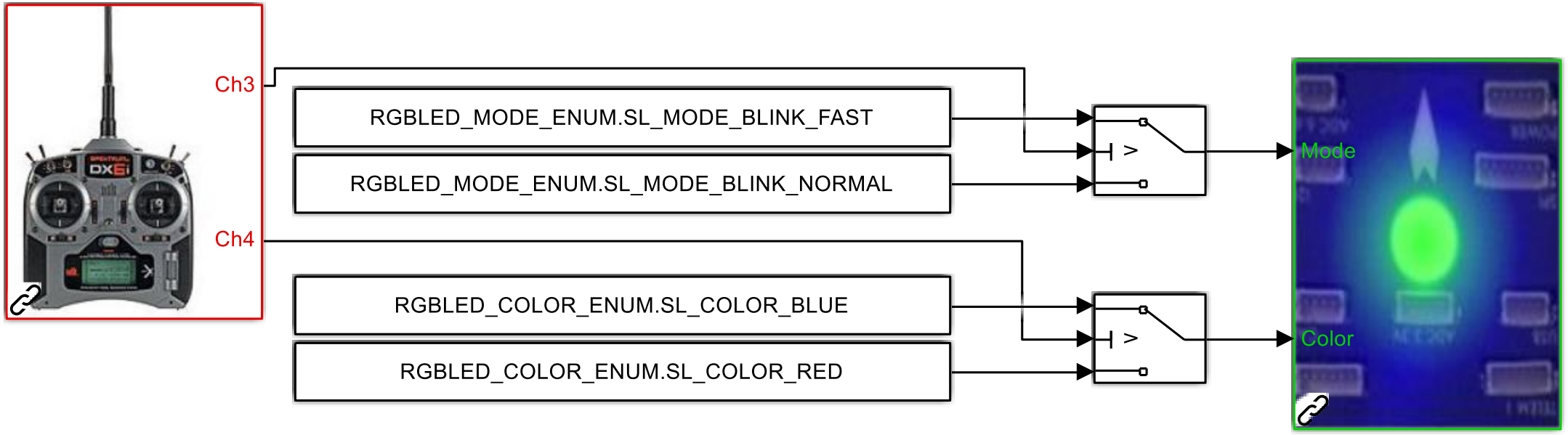
RGBLED_COLOR_ENUM

Value
SL_COLOR_OFF (0)
SL_COLOR_RED (1)
SL_COLOR_GREEN (2)
SL_COLOR_BLUE (3)
SL_COLOR_YELLOW (4)
SL_COLOR_PURPLE (5)
SL_COLOR_AMBER (6)
SL_COLOR_CYAN (7)
SL_COLOR_WHITE (8)

控制LED灯实验操作具体流程

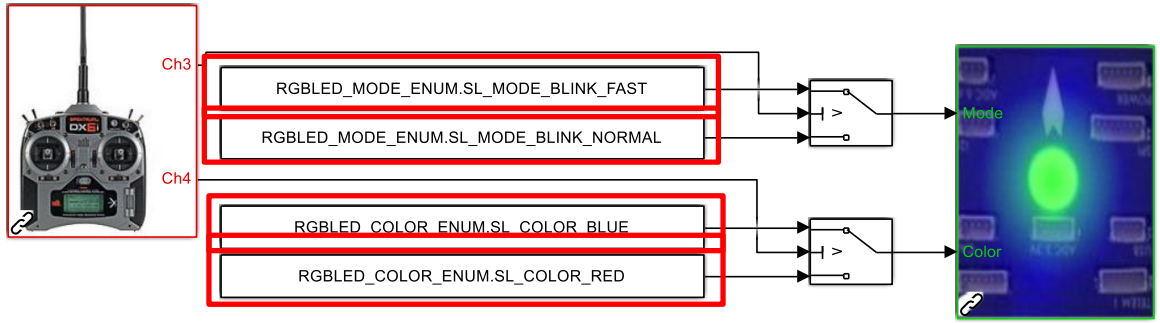
□ 设计LED灯控制模型

3）实现模型。由于遥控器的PWM输出范围为1100~1900，这里选择1500为Switch模块切换量，使用遥控器的两个通道分别控制LED灯的模式和颜色，如下图所示



控制LED灯实验操作具体流程

□ 设计LED灯控制模型

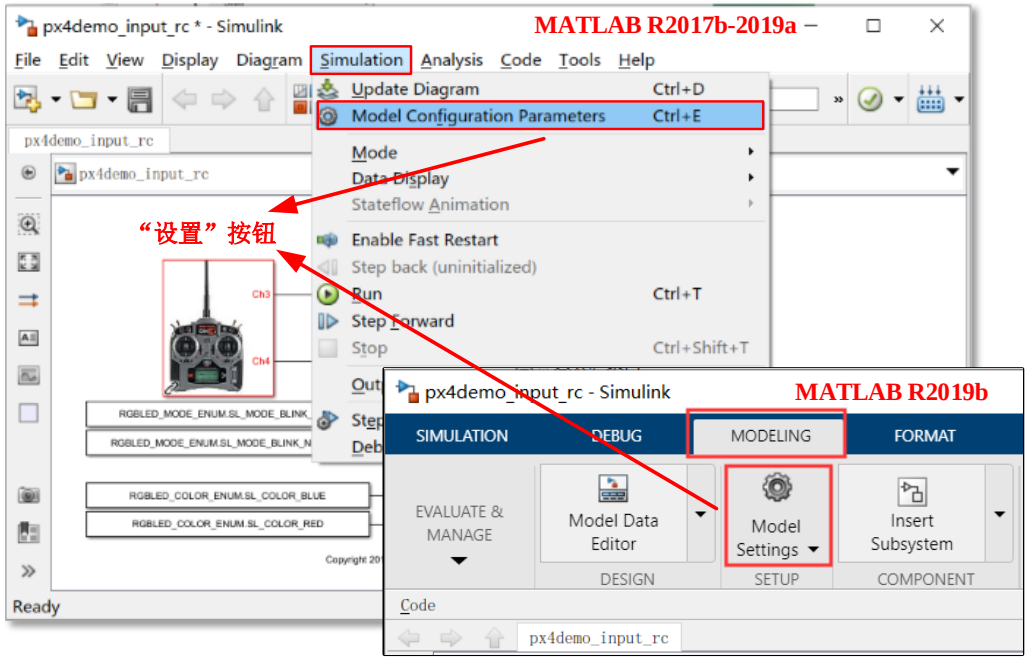


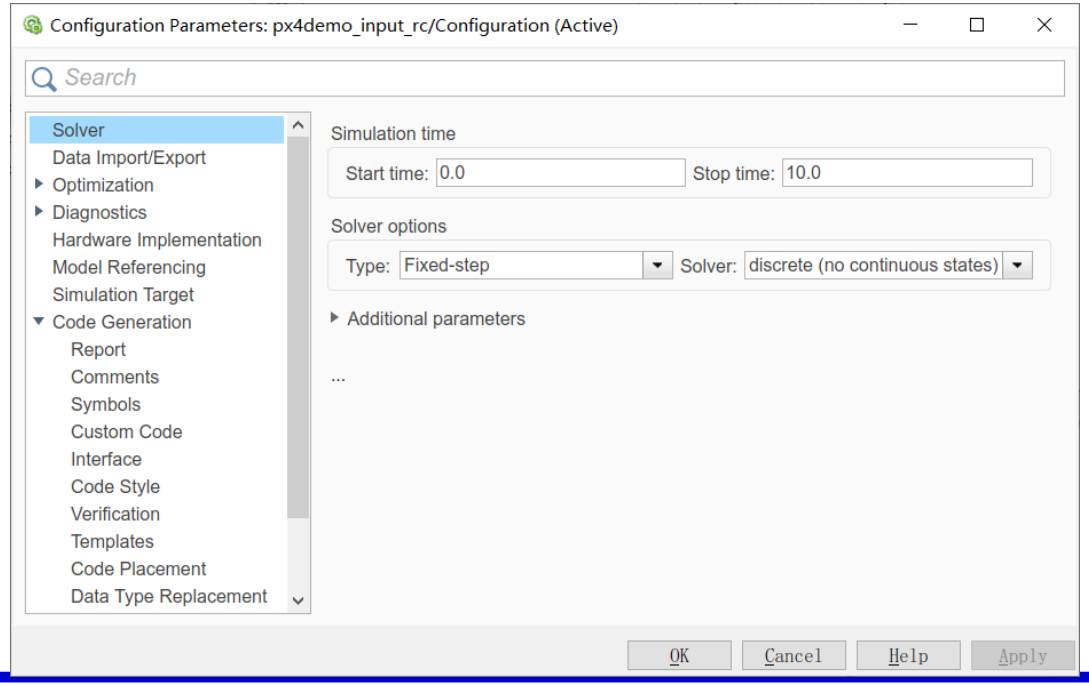
- 1) CH3通道改变LED灯的模式。当CH3>1500时，“Mode”接收“RGBLED_MODE_ENUM.SL_MODE_BLINK_FAST”参数；当CH3 ≤ 1500时，“Mode”接收“RGBLED_MODE_ENUM.SL_MODE_BLINK_NORMAL”参数。
- 2) CH4通道改变LED灯的颜色。当CH4 ≤ 1500时，“Color”接收“RGBLED_COLOR_ENUM.COLOR_BLUE”；当CH4>1500时，“Color”接收“RGBLED_COLOR_ENUM.COLOR_RED”参数。

控制LED灯实验操作具体流程

□ 控制器代码生成与固件下载

（1）对于MATLAB 2017b~2019a，点击Simulink界面的“Simulation”菜单，在下拉框中选择“Model Configuration Parameters”选项；对于MATLAB 2019b及更高版本，可点击工具标签栏上的“设置”按钮，进入右下图所示的Simulink设置页面。

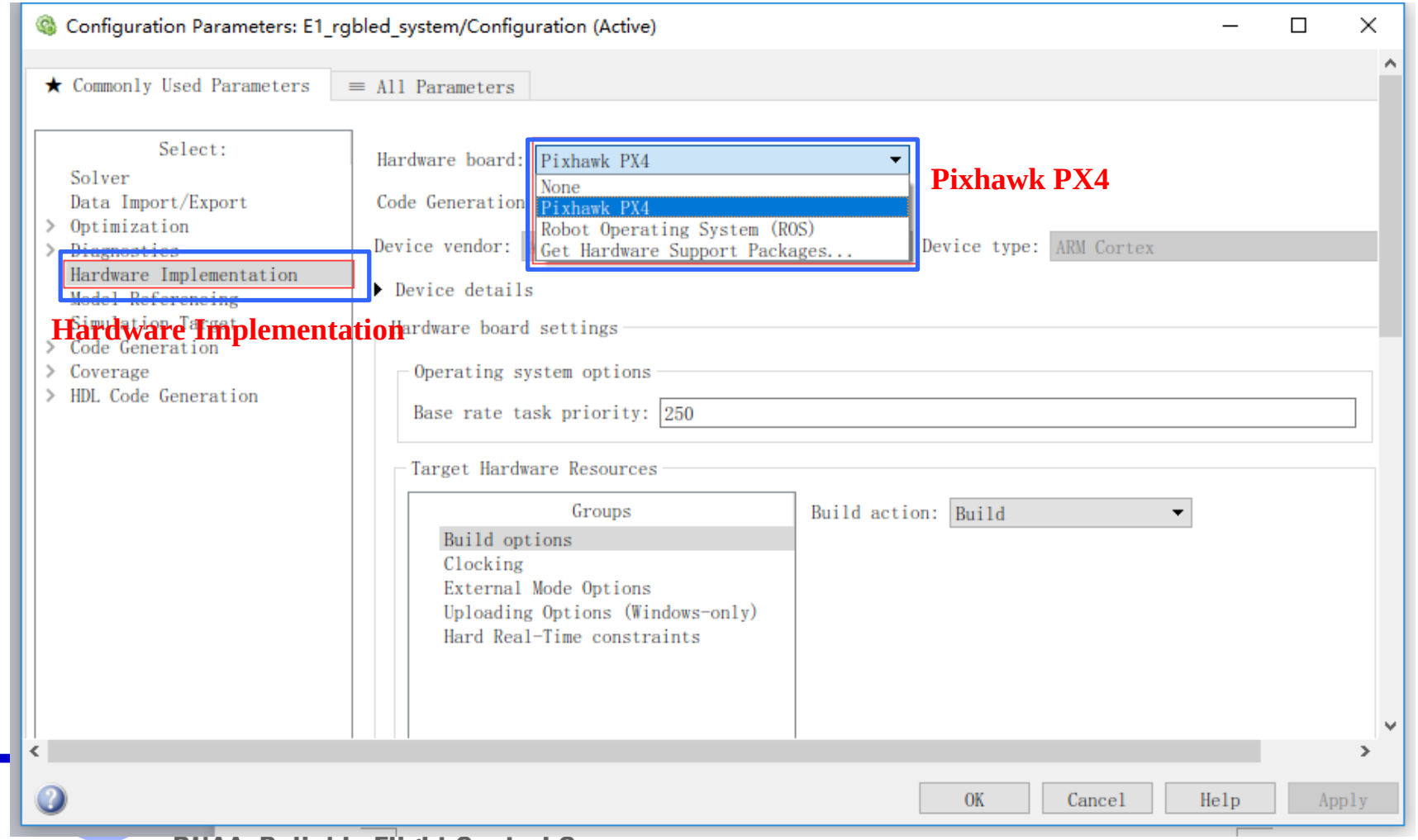




控制LED灯实验操作具体流程

□ 控制器代码生成与固件下载

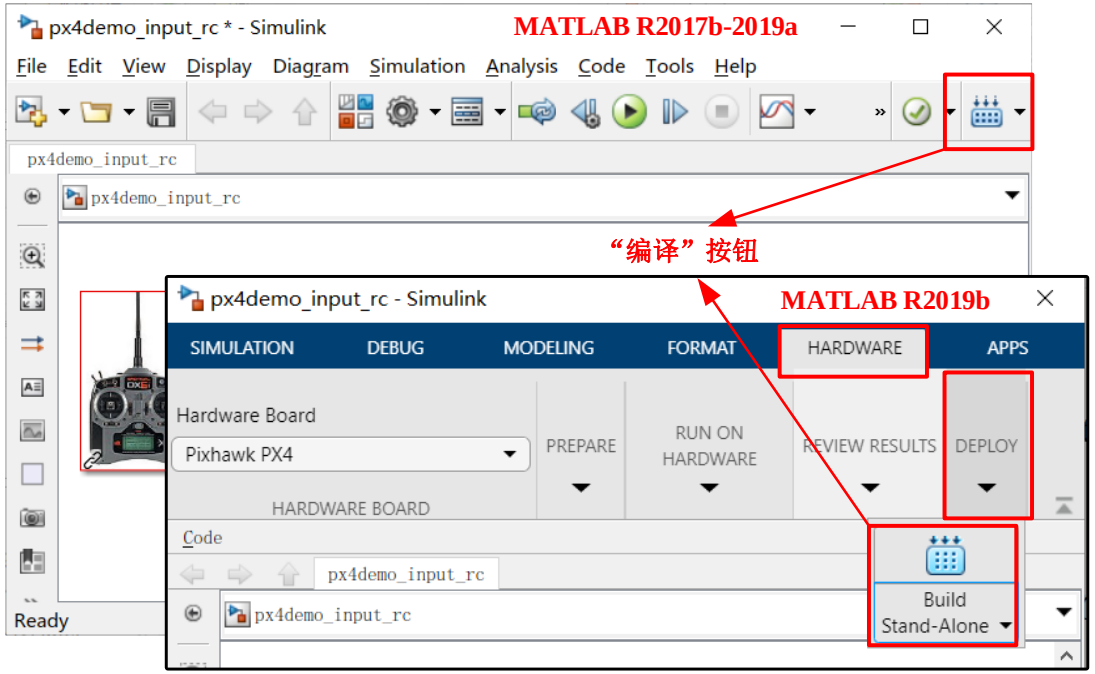
（2）选择目标硬件：在Simulink模型配置窗口中，将“Hardware Implementation” - “Hadware Board”设置为“Pixhawk PX4”，使Simulink安装PX4的规则生成控制器代码。

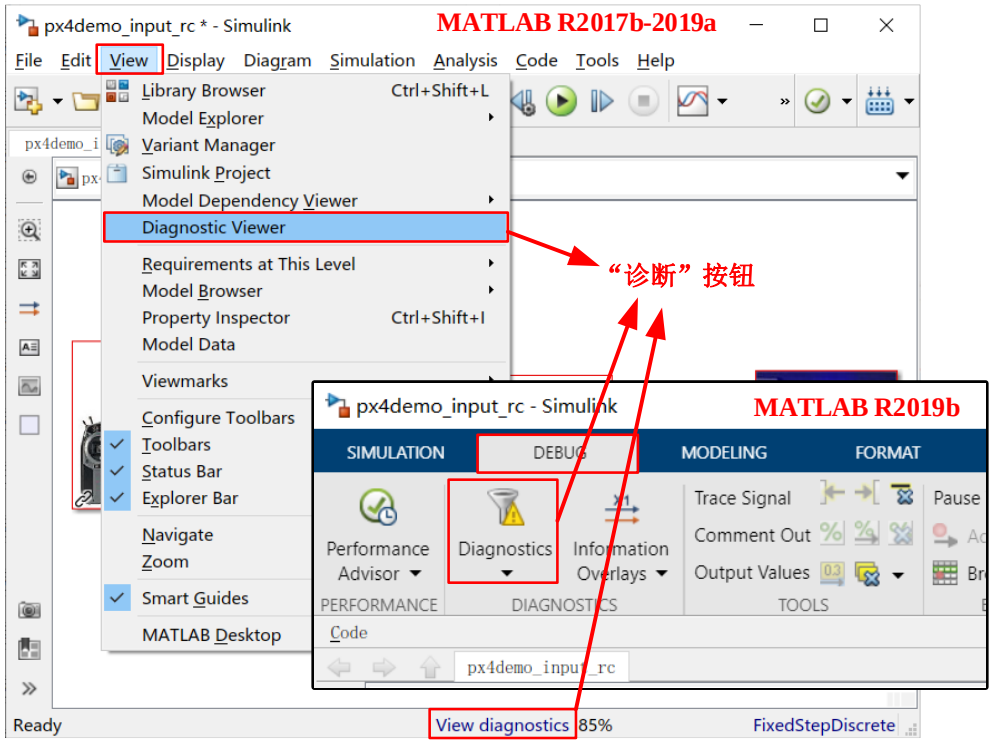


控制LED灯实验操作具体流程

□ 控制器代码生成与固件下载

（3）编译模型。点击图中操作，点击“编译” 按钮即可开始Simulink中设计模块与PX4固件的编译；再点击“诊断”按钮，可以观察到编译的详细过程。

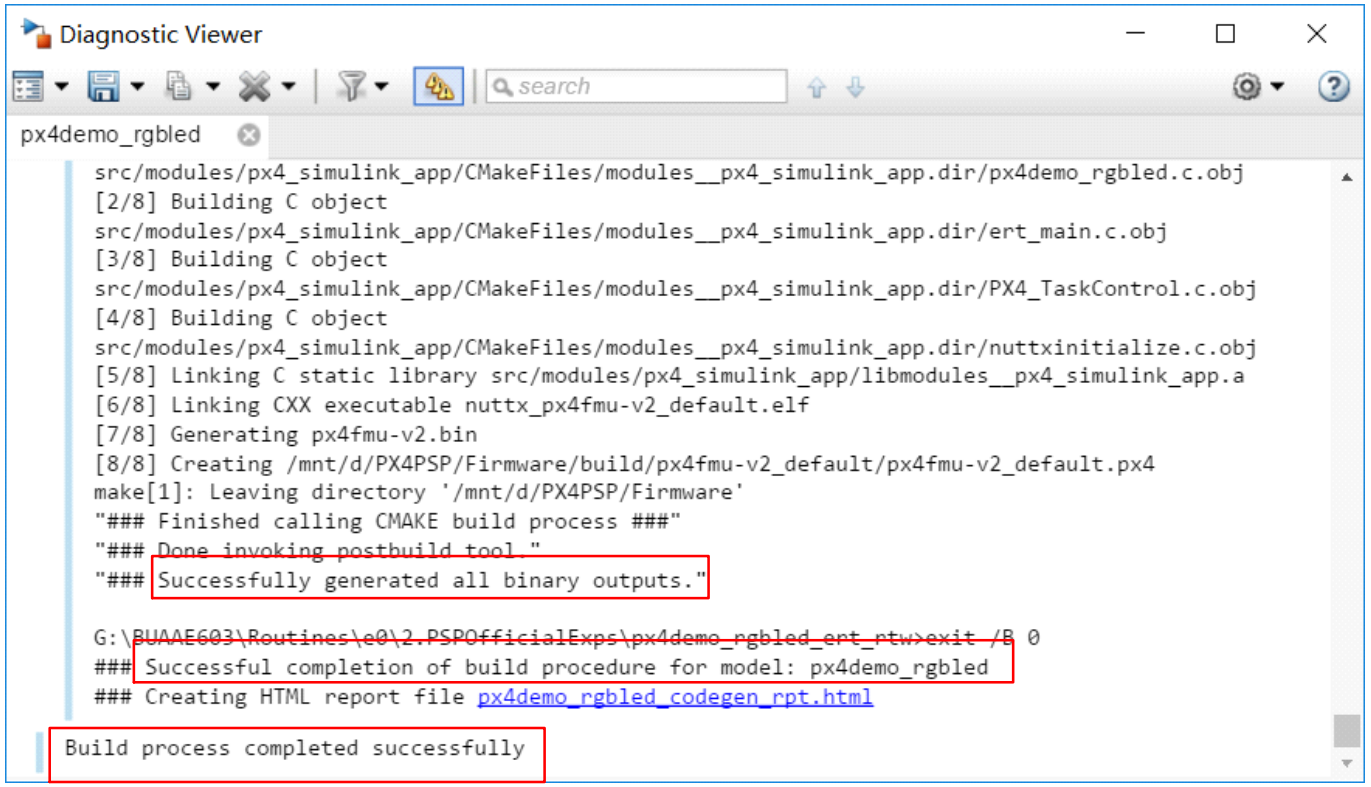


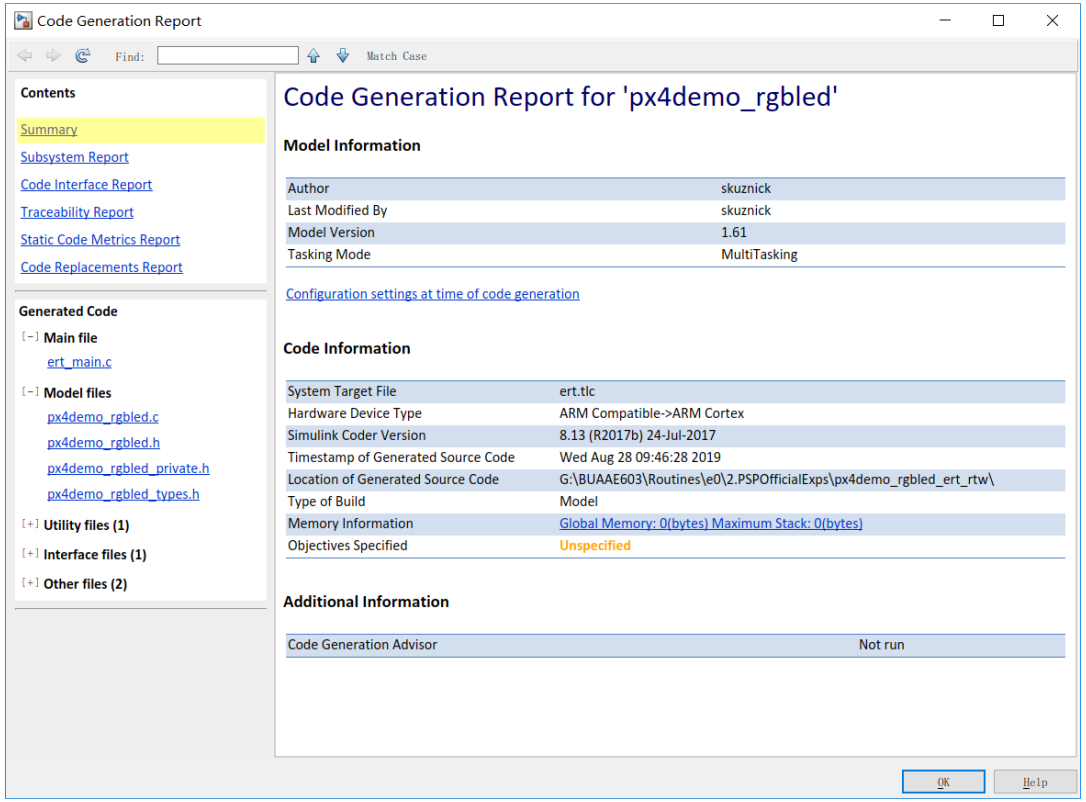


控制LED灯实验操作具体流程

□ 控制器代码生成与固件下载

（3）编译过程如下图，当代码生成完毕时，会弹出右下侧窗口，可以查看报告。



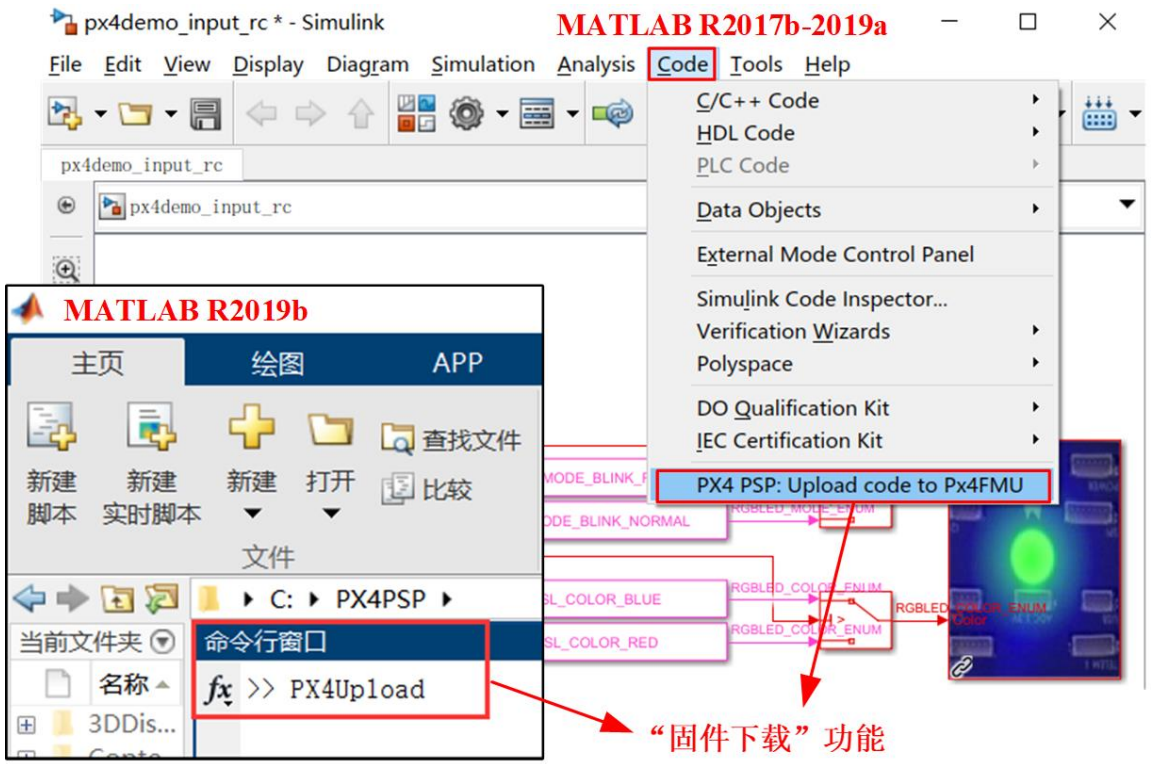


控制LED灯实验操作具体流程

□ 控制器代码生成与固件下载

（4）下载固件。利用PSP工具箱提供的固件一键下载功能，具体步骤如下：

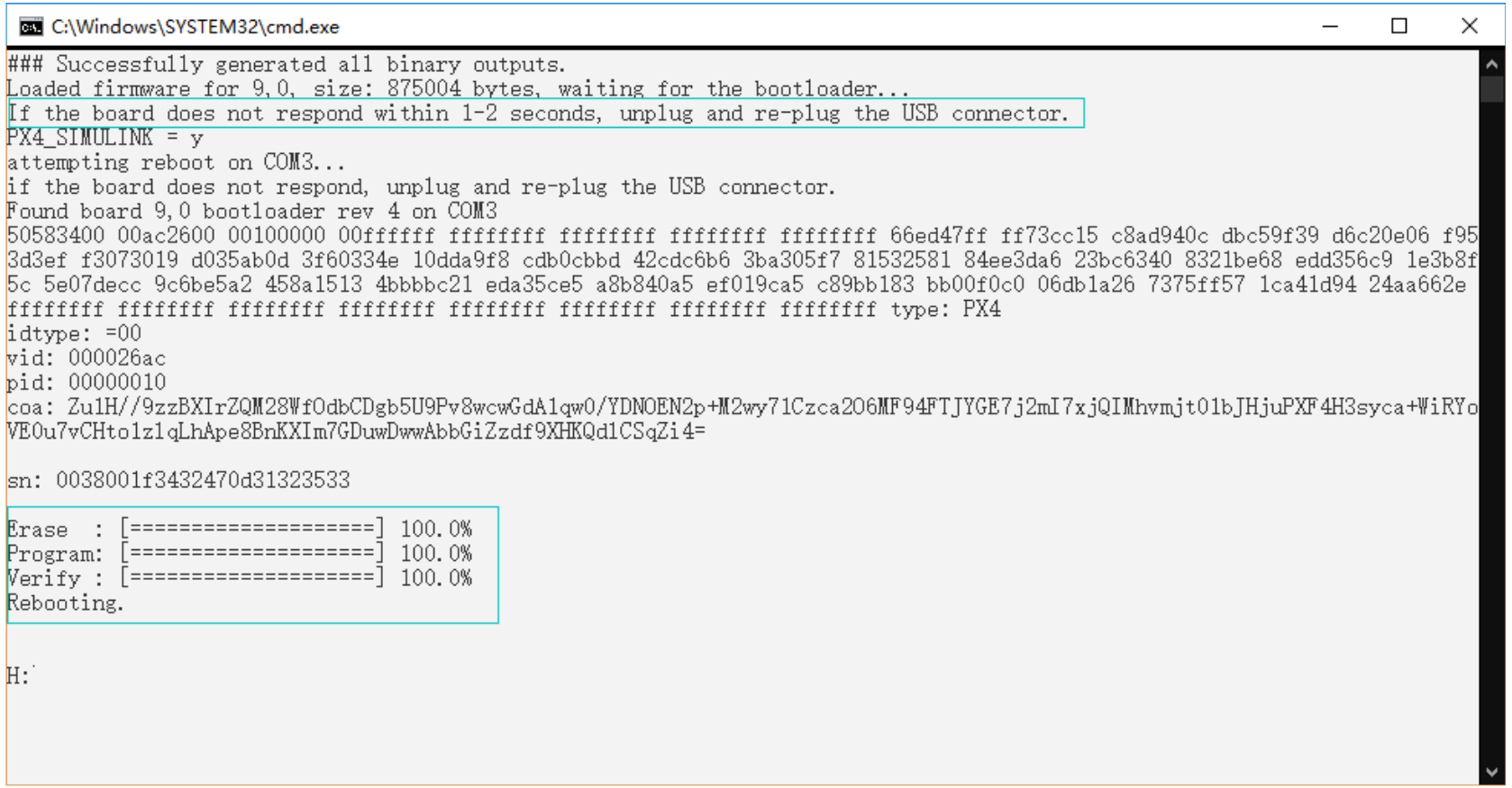
- 用USB线连接计算机和Pixhawk的MicroUSB口
- 对于MATLAB 2017b~2019a，在Simulink的Code下拉菜单中点击“PX4 PSP: Upload code to Px4FMU”选项；对于MATLAB 2019b及更高版本，在MATLAB主界面的“命令行窗口”输入“PX4Upload”命令来开始固件下载。



控制LED灯实验操作具体流程

□ 控制器代码生成与固件下载

- Simulink会自动识别Pixhawk自驾仪，并将编译得到的PX4固件下载与部署。当进度条达到100%说明部署成功。注意：有时需要根据提示重新插拔Pixhawk才能开始下载与部署流程。



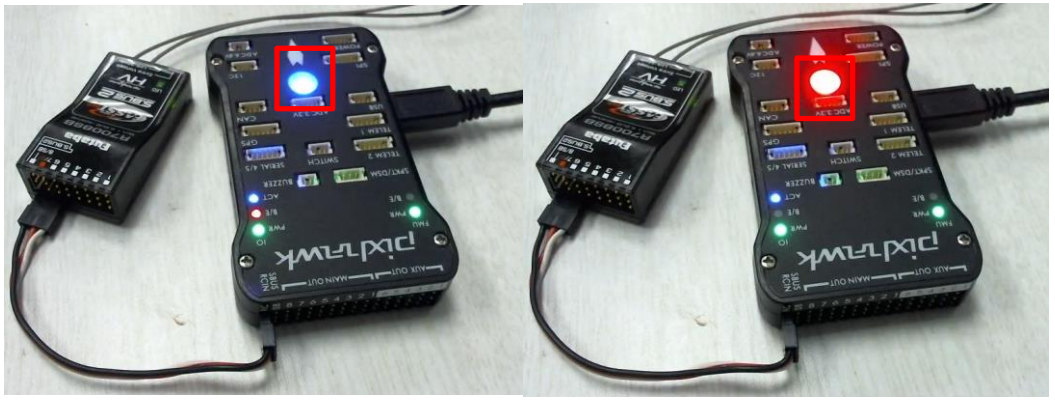
控制LED灯实验操作具体流程

□ 实验效果

默认状态下，即不操作遥控器时，LED灯是蓝色慢闪状态。

- 当遥控器的左手油门摇杆置于右上方位置（CH3>1500且CH4>1500）时，Pixhawk自驾仪的LED灯光为蓝色快闪；
- 当遥控器的油门摇杆置于左上方位置（CH3>1500且CH4<1500）时，Pixhawk自驾仪的LED灯光为红色快闪；
- 当遥控器的油门摇杆置于左下方位置（CH3<1500且CH4<1500）时，Pixhawk自驾仪的LED灯光为红色慢闪；
- 当遥控器的油门摇杆置于右下方位置（CH3<1500且CH4>1500）时，Pixhawk自驾仪的LED灯光为蓝色慢闪。

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



姿态控制实验操作具体流程

本章节以一个设计好的姿态控制系统为例，介绍整个实验的基本操作流程。例程见“e0\3.DesignExps\Exp1_AttitudeController.slx”

□ 算法设计与仿真阶段

（1）步骤一：控制器设计

新建一个Simulink文件，在其中设计多旋翼的姿态控制器。设计要求：

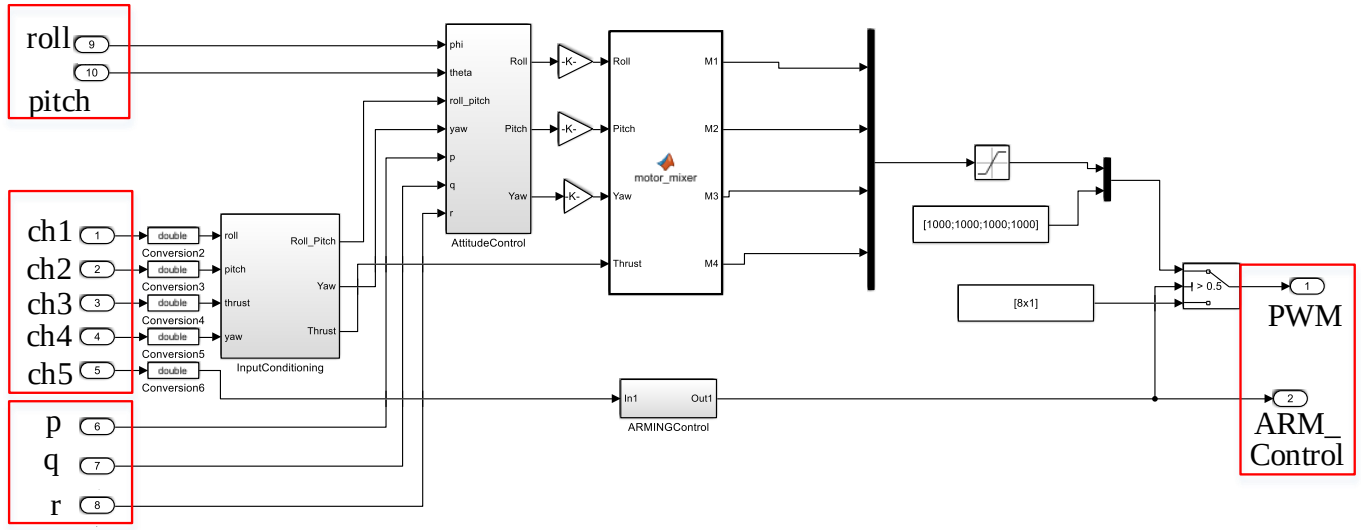
- 输入数据：1）遥控器Ch1~Ch5通道信号，数据范围大约为1100-1900，在处理遥控器数据时需要校准或考虑死区；2）角速度反馈量AngRateB，三个分量用p,q,r表示（单位：rad/s），分别代表滚转角速度（沿机体x轴转动）、俯仰角速度（沿机体y轴转动）和偏航角速度（沿机体z轴转动）；3）多旋翼欧拉角（单位为rad）。这里主要考虑滚转角和俯仰角，暂不考虑偏航控制。
- 输出数据：1）四个电机的PWM控制信号，数据范围1000~2000；2）是否解锁标识符，数据类型bool型
- 实现效果：CH3油门通道控制飞机升降；向前推俯仰摇杆（即CH2<1500）控制多旋翼向前飞；向左推滚转摇杆（即CH1<1500）控制多旋翼向左飞；向下拉拨杆开关（即CH5>1500）解锁控制器。

姿态控制实验操作具体流程

□ 算法设计与仿真阶段

（1）步骤一：控制器设计

这里我们给出一个设计好的例子，见文件“e0\3.DesignExps\Exp1_AttitudeController.slx”，打开该文件后的Simulink框图见右图。请仔细阅读其中的子模块的实现方法，并进行功能的完善，可以将偏航通道的控制加入。

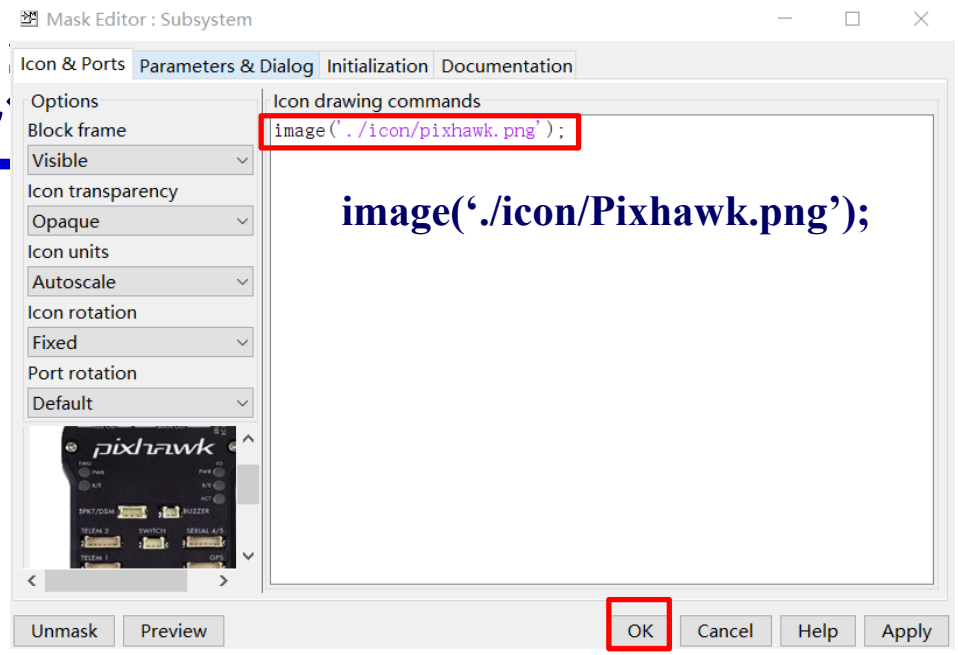


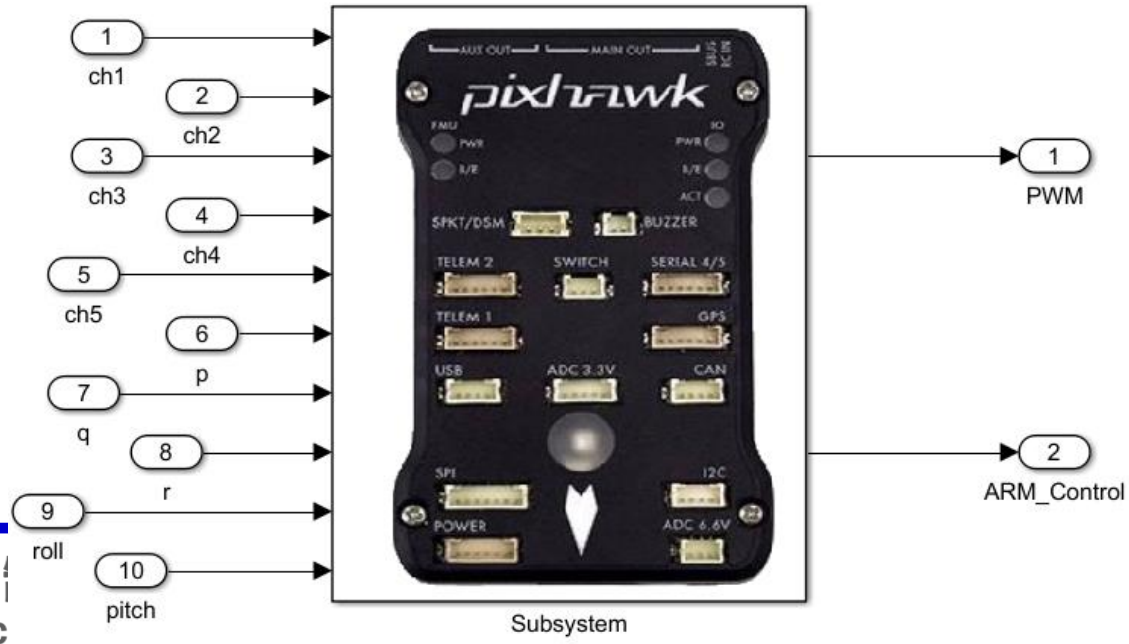
姿态控制实验操作具体流

□ 算法设计与仿真阶段

（2）步骤二：生成控制器子模块

将上文的控制器用鼠标全部选中（或者按下键盘CTRL+A），右键鼠标，点击“Create Subsystem For Selection”即可将控制器封装为一个子模块。右键该子模块，点击“Mask”-“Create Mask”在“Icon drawing commands”输入框（见右上图）中输入“image(‘./icon/Pixhawk.png’);”，再点击“OK”，并调整接口位置就可以得到如右下所示的子模块。



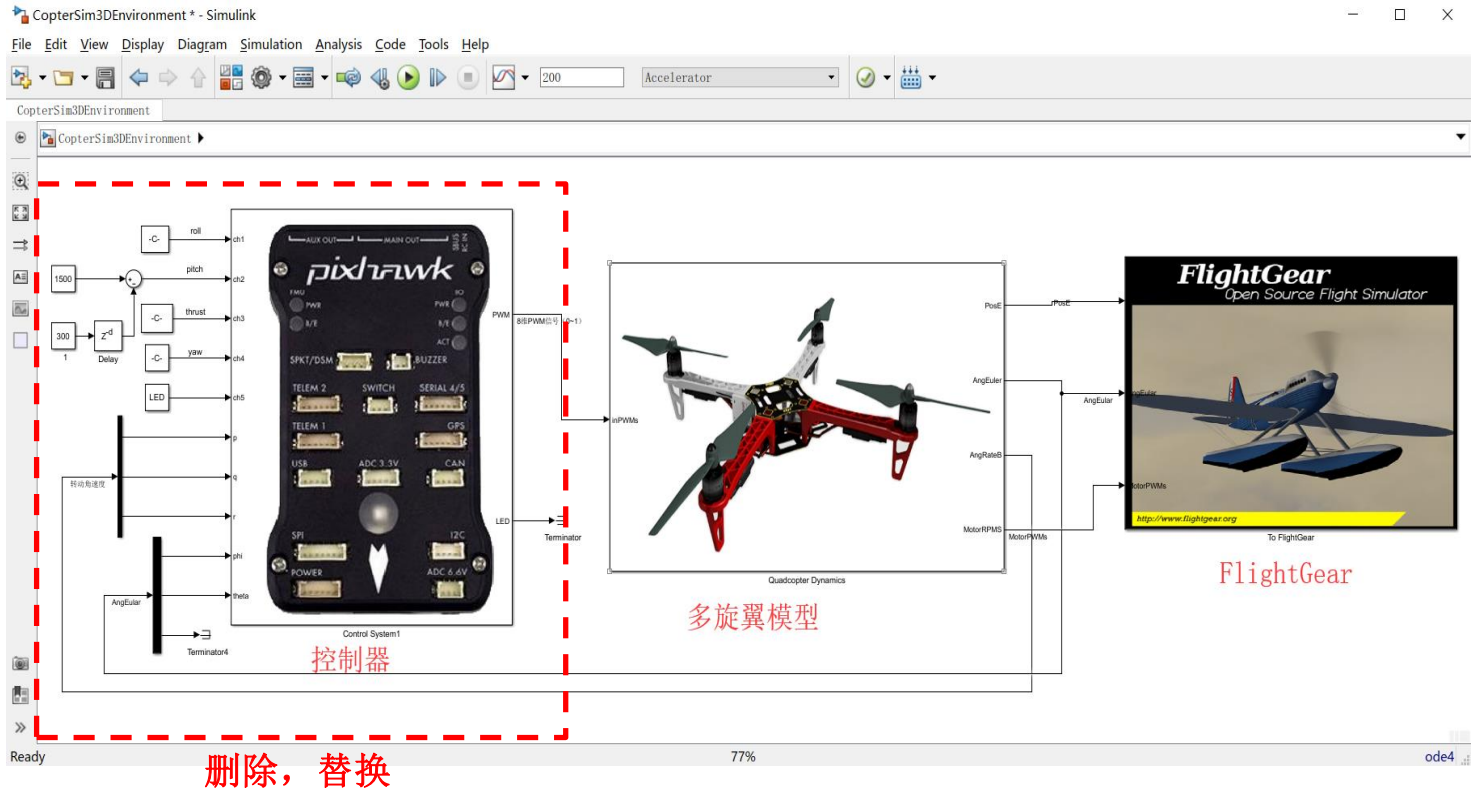


姿态控制实验操作具体流程

□ 算法设计与仿真阶段

（3）步骤三：控制器与模型整合

打开前文给的Simulink多旋翼仿真程序“e0\1.SoftwareSimExps\CopterSim3DEnvironment.slx”（如右图所示），删掉其中的原有的控制器子模块（注意备份），然后将步骤二中得到的新控制器子模块复制进来进行替换。

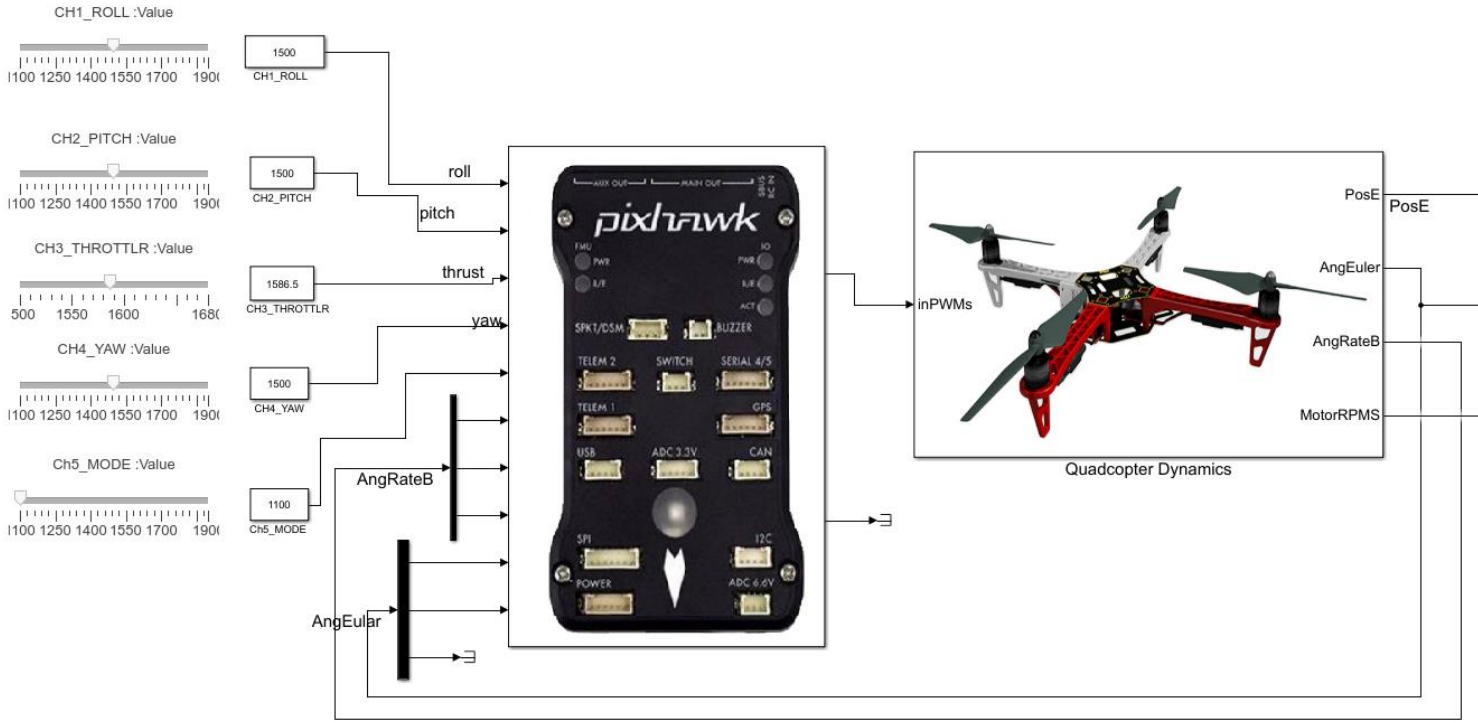


姿态控制实验操作具体流程

□ 算法设计与仿真阶段

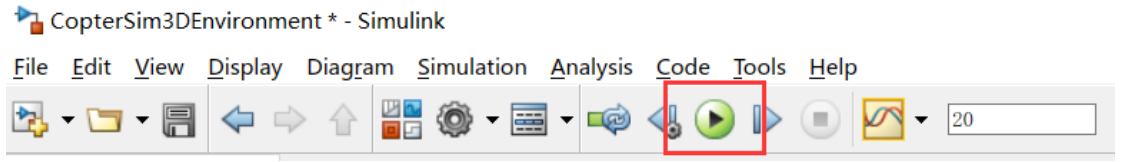
（4）步骤四：连线与输入输出配置

将控制器与多旋翼模型进行重新连线。由于此时遥控器信号无法获取，可以用常值来代替，或者用函数模拟相应的遥控器动作。这里我们也给出一个例子，已经连接好了控制器和虚拟遥控器信号，见文件“e0\3.DesignExps\Exp2_ControlSystemDemo.slx”，文件内部见右图。



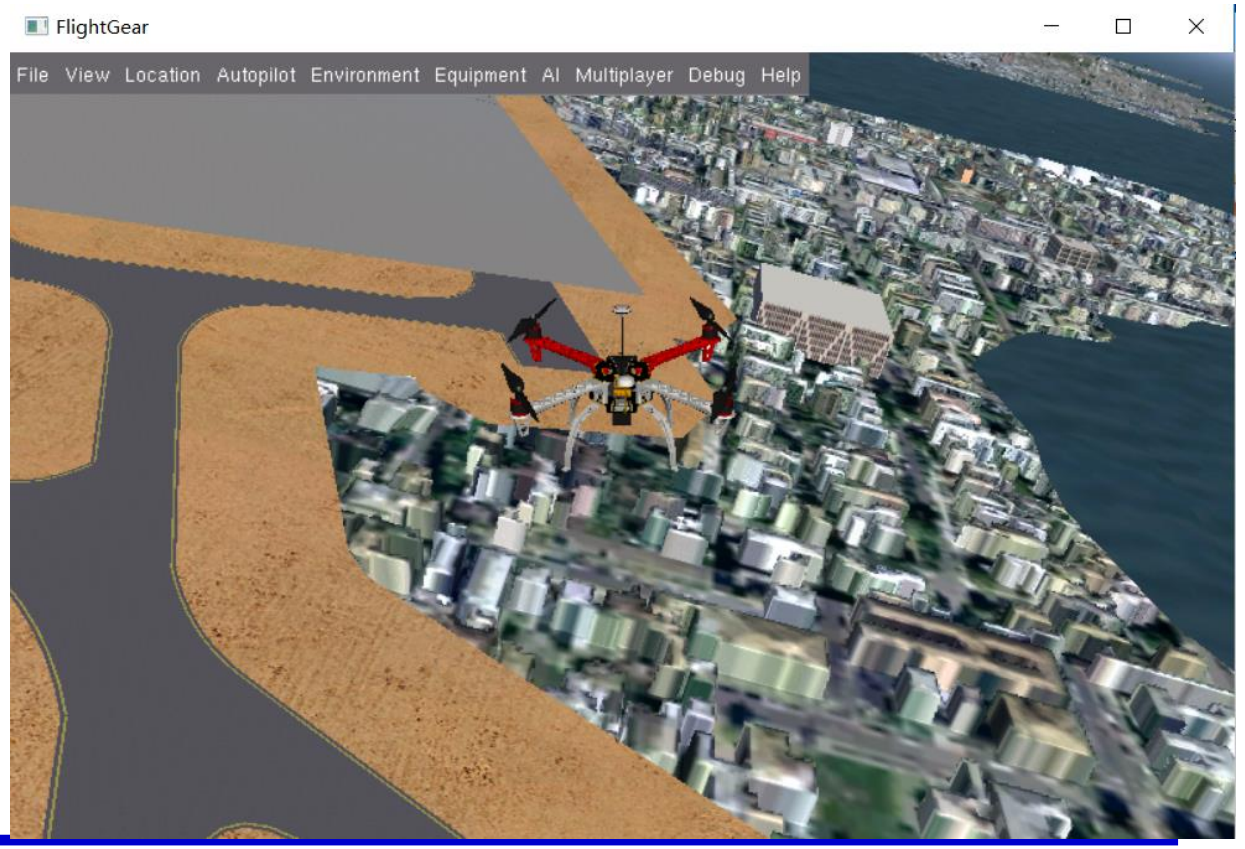
姿态控制实验操作具体流程

□ 算法设计与仿真阶段



（5）步骤五：开始联合仿真

如果FlightGear没有处于打开状态，双击文件“FlightGear-Start”打开FlightGear，然后点击Simulink工具栏“开始仿真”按钮（见右上图）开始仿真。此时可以在FlightGear中（见右图）观察到，多旋翼爬升一段时间后，可以滑动Simulink中的滑块来模拟用遥控器控制四旋翼的基本操作。

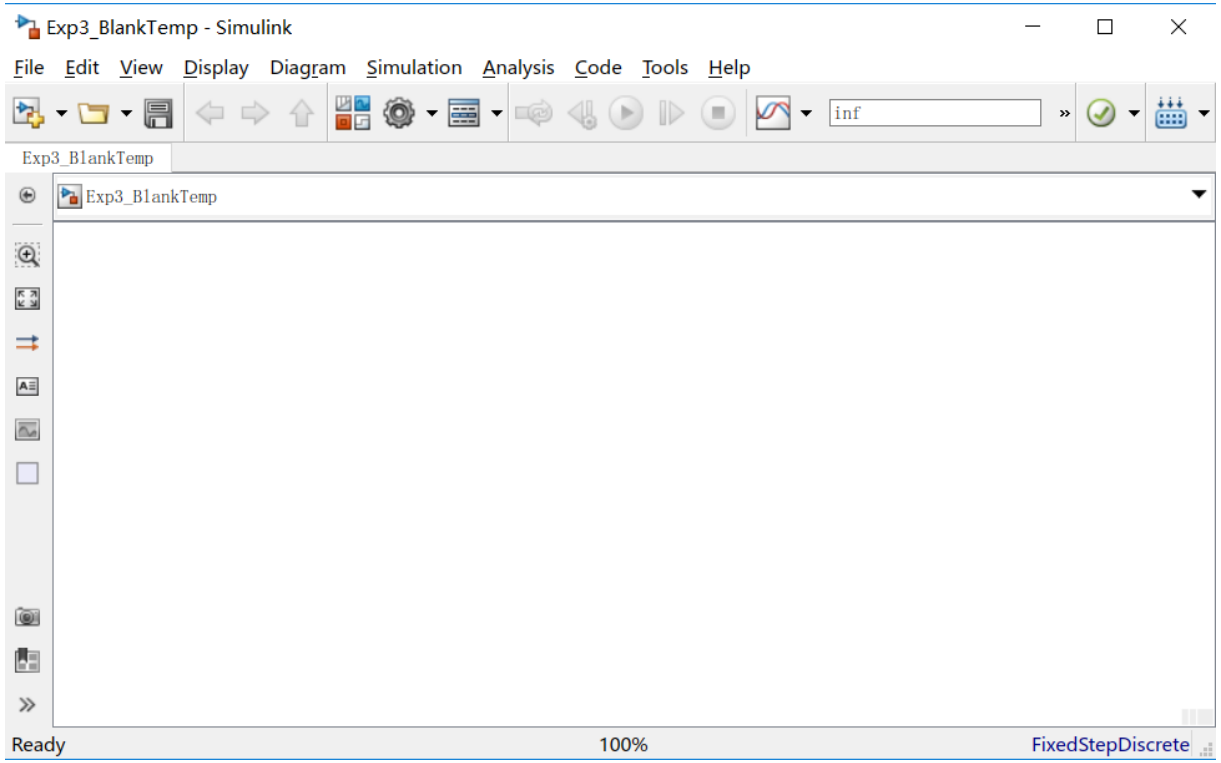


姿态控制实验操作具体流程

□ 代码生成与配置阶段

（6）步骤六：代码生成环境配置

上述Simulink中的软件在环仿真完成后，将其中的控制器模块单独复制出来，粘贴到文件“e0\3.DesignExps\Exp3_BlankTemp.slx”中（这个文件已经配置好了代码生成所需的所有设置，也可以新建一个空白Simulink文件并按前文流程对PSP工具箱进行配置）。

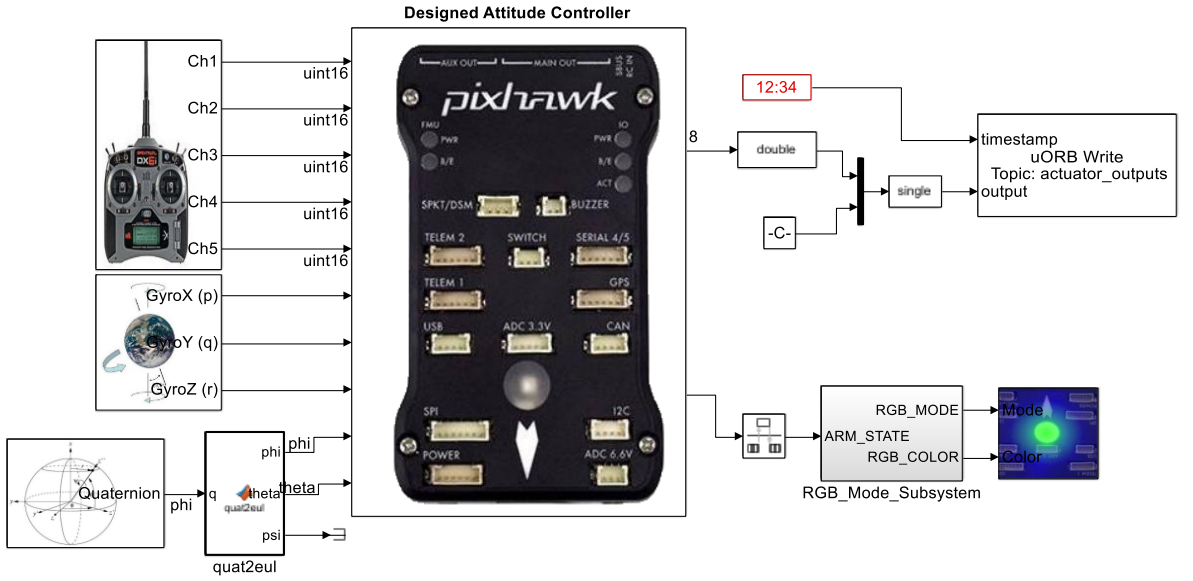


姿态控制实验操作具体流程

□ 代码生成与配置阶段

（7）步骤七：控制器与PSP模块连线

从Simulink PSP工具箱中提取相应的输入输出接口，与步骤六中的控制器进行连线，连线后结果可以参考文件“e0\3.DesignExps\Exp4_AttitudeSystemCodeGen.slx”，内部细节见右图。这里需要注意，由于后面要进行硬件在环仿真而不是实际飞行，PWM的输出接口需要通过uORB消息给Pixhawk发送actuator_outputs消息来实现，而不是直接用PSP工具箱的PWM输出模块。



姿态控制实验操作具体流程

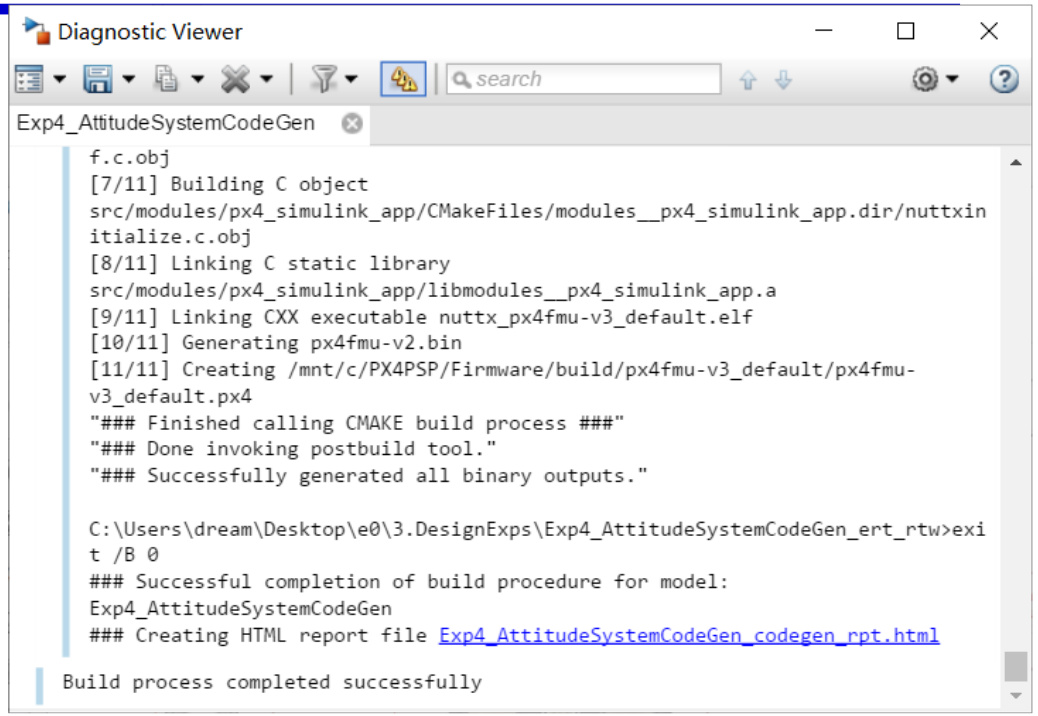
□ 代码生成与配置阶段

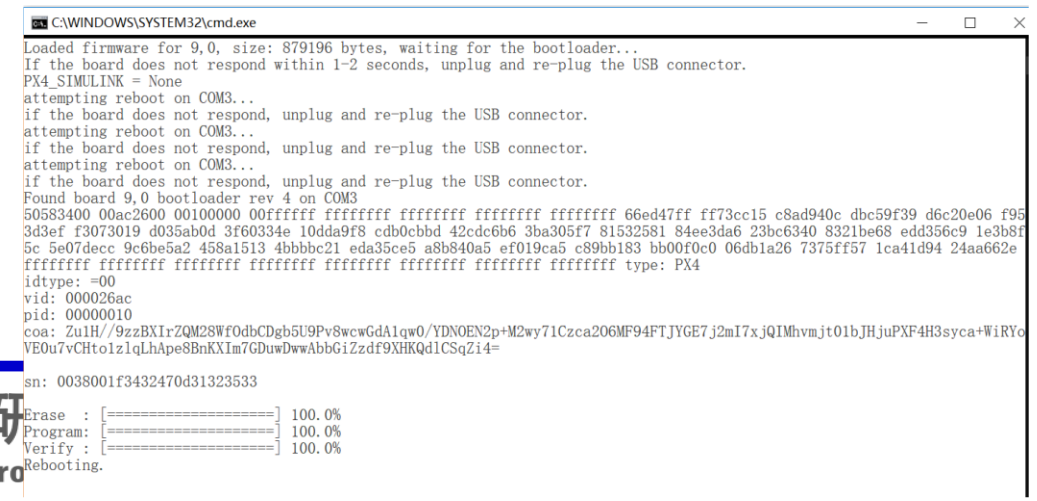
（8）步骤八：编译并生产固件

点击Simulink工具栏“编译”按钮，就可以自动编译生成代码，并生成自驾仪固件。得到右上所示结果说明编译成功。

（9）步骤九：代码下载Pixhawk自驾仪

用USB线连接计算机和Pixhawk自驾仪，然后使用“Upload Code”功能将固件下载到Pixhawk中。得到右下图说明下载成功。





姿态控制实验操作具体流程

□ 处理器在环仿真阶段

（10）步骤十：硬件系统连接

按照右图所示用三色杜邦线连接接收机与Pixhawk，然后Pixhawk与电脑通过USB数据线连接，此时可以看到Pixhawk上的蓝灯亮起并呼吸闪烁，接收机上的灯光为蓝白色常亮。此时打开遥控器开关（油门杆拉到最低位置），可以观察到Pixhawk上的LED灯快速闪烁一下，说明接收到遥控器数据。如果Pixhawk的LED灯没有任何改变，说明遥控器与接收机的连接存在问题，需要检查确认。

姿态控制实验操作具体流程

□ 处理器在环仿真阶段

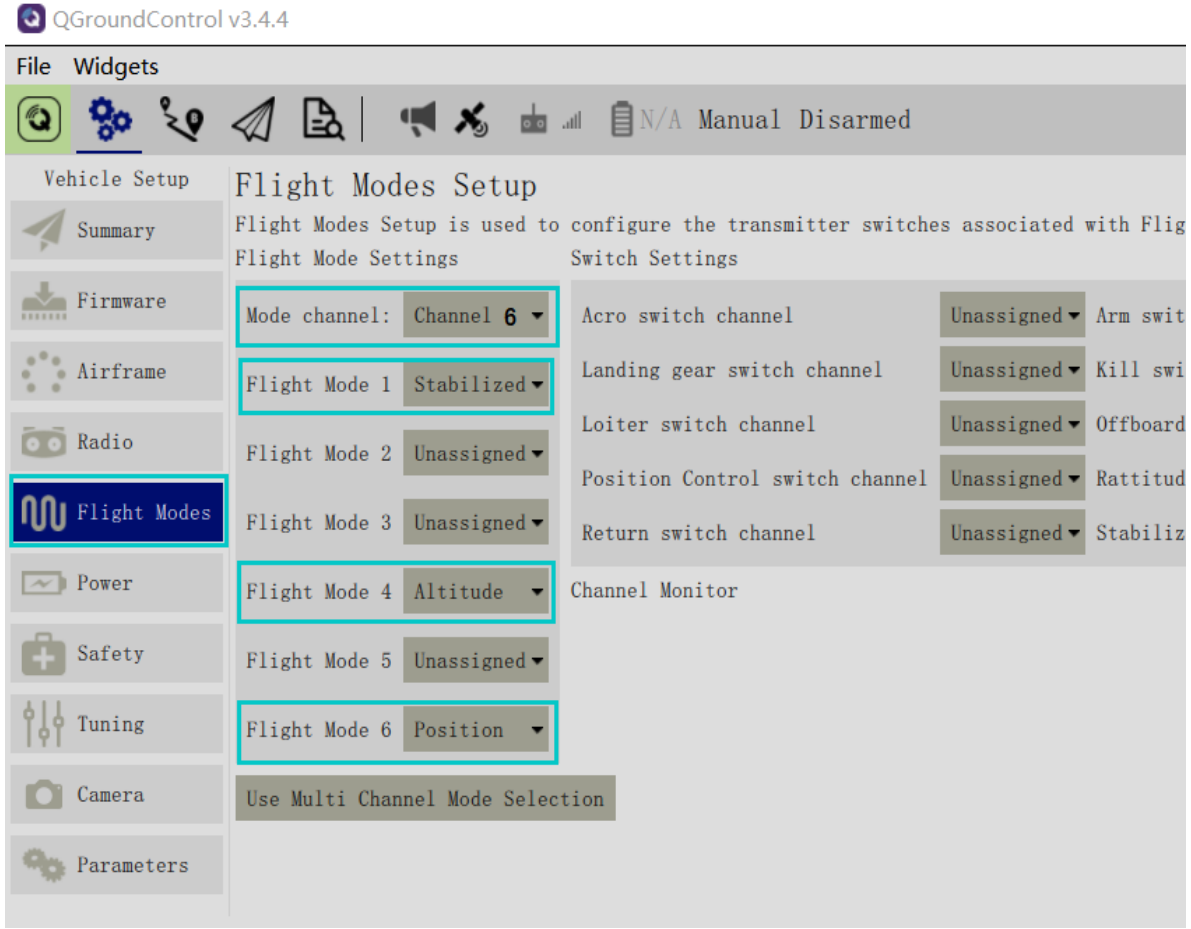
（11）步骤十一：模型仿真器软件配置

打开QGC地面站，连接上Pixhawk自驾仪，

1）进入“Airframe” 标签，确保模型处于“HIL Quadcopter X”机架模式

2）进入“Flight Modes”标签页，确认模式切换开关不是CH5，避免PX4模式切换开关对CH5通道占用，影响灯光等效果

3）关闭QGC地面站

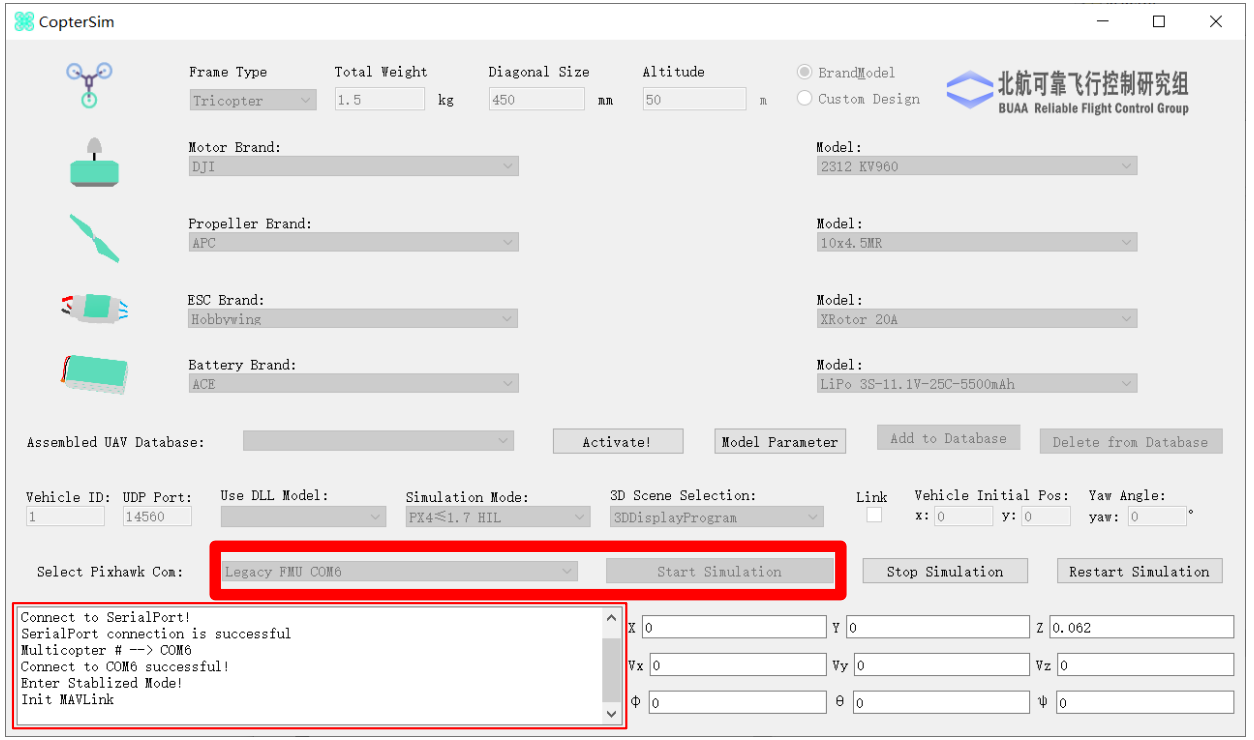


姿态控制实验操作具体流程

□ 处理器在环仿真阶段

（11）步骤十一：模型仿真器软件配置

双击桌面的CopterSim快捷方式即可以打开多旋翼模拟器软件。不用配置任何参数，直接在“飞控选择”下拉框中选择上一步查到的串口号（例如“**FMU COM3”），再点击“开始仿真”按钮（保证QGC地面站关闭）就可以进入硬件在环仿真模式。此时可以看到如右图所示的界面左下角收到自驾仪返回的相关消息，以及Pixhawk飞控上的灯光从蓝色变为绿色闪烁（如果自己编写的程序有控制灯光的作用，那么灯光会按程序设置的来闪动）。



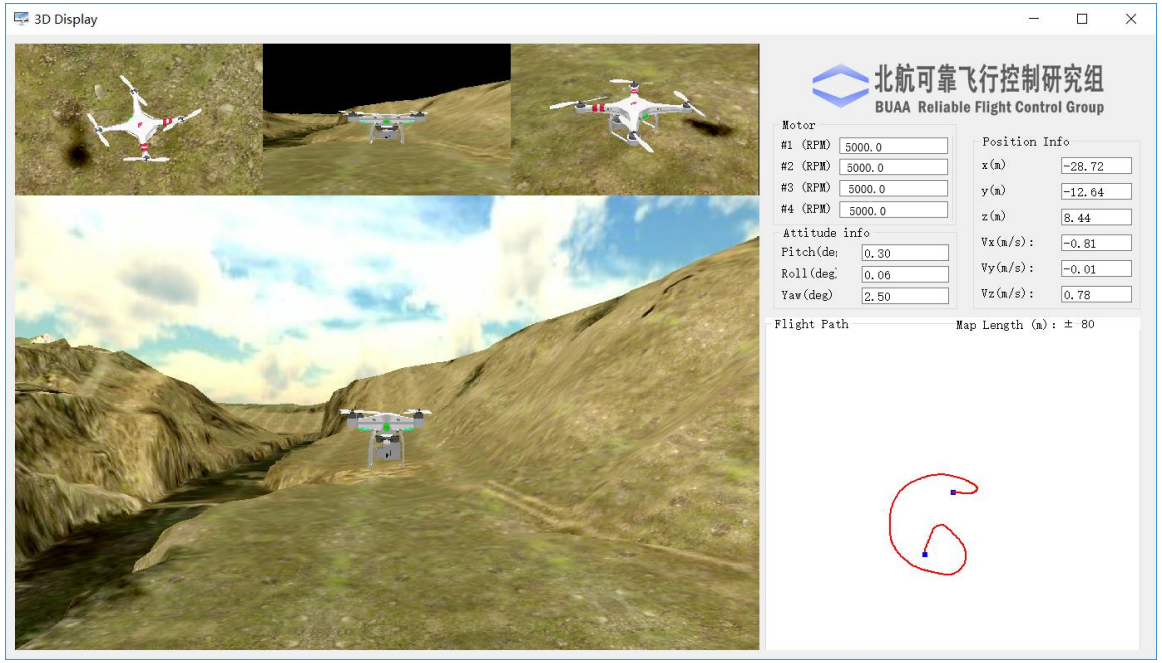
姿态控制实验操作具体流程

□ 处理器在环仿真阶段

（12）步骤十二：多旋翼三维飞行显示程序配置

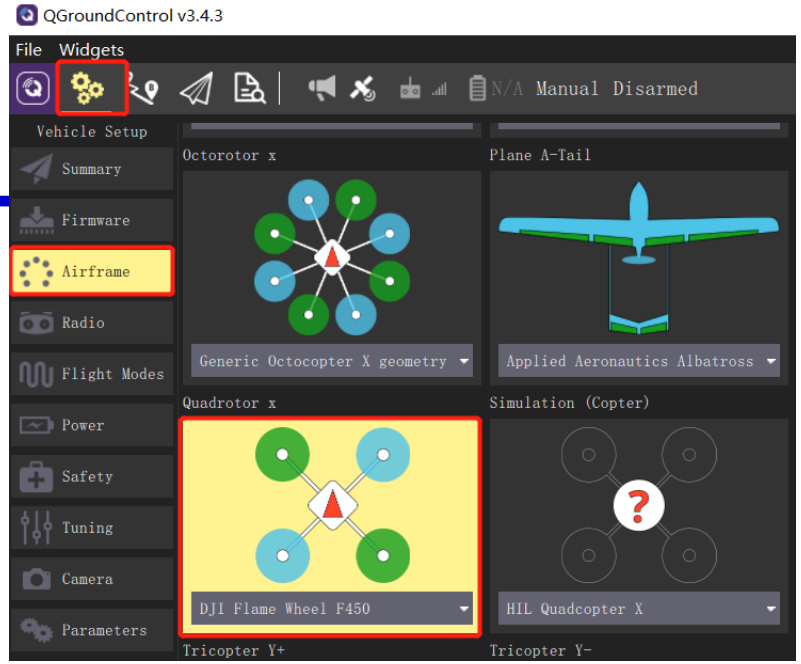
双击桌面的3DDisplay快捷方式即可打开3DDisplay三维显示软件。这个软件不需要任何配置，它会被动地接收模型仿真软件发送的飞机的飞行姿态与轨迹信息并实时显示。

控制遥控器解锁多旋翼（油门杆右下三秒解锁Pixhawk自驾仪，将CH5拨杆开关拨到最下解锁控制器），便可操纵遥控器使多旋翼完成相应动作。如右图在3多旋翼三维飞行显示程序界面左侧观察多旋翼位置和姿态变化，界面右上角观察实时飞行数据，界面右下角观察多旋翼运动轨迹。



姿态控制实验操作具体流程

□ 实际飞行实验阶段与结果对比



（13）步骤十三：安装Pixhawk到多旋翼机架上

实际飞行试验所采用的多旋翼为F450四旋翼，见下图；多旋翼的参数经过精确测量与系统辨识，保证实际模型与Simulink仿真模型是一致的。在实际飞行时需要在QGC中将Pixhawk的机架类型从“HIL Quadcopter X”修改为“DJI Flame Wheel F450”，并完成传感器校准。

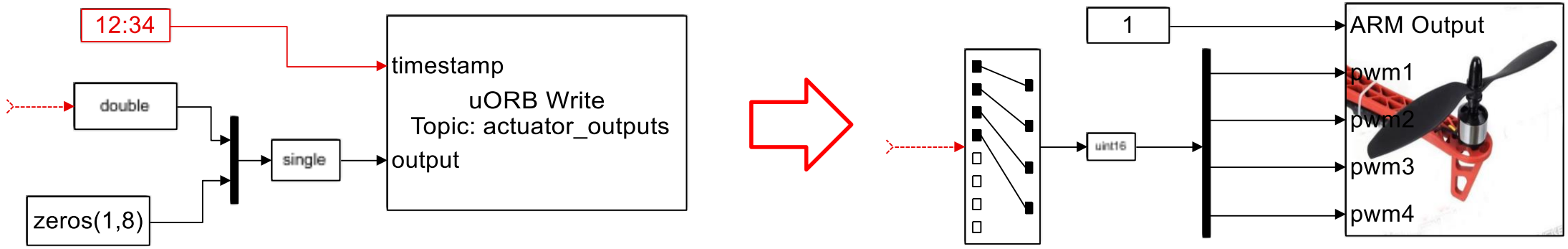
F450机架

姿态控制实验操作具体流程

□ 实际飞行实验阶段与结果对比

（14）步骤十四：调整Simulink控制器

打开Simulink文件，按下图所示将uORB的输出改成PSP工具箱提供的PWM_out输出模块，重新生成代码并下载到Pixhawk中。



姿态控制实验操作具体流程

□ 实际飞行实验阶段与结果对比

（15）步骤十五：参数设置和测试

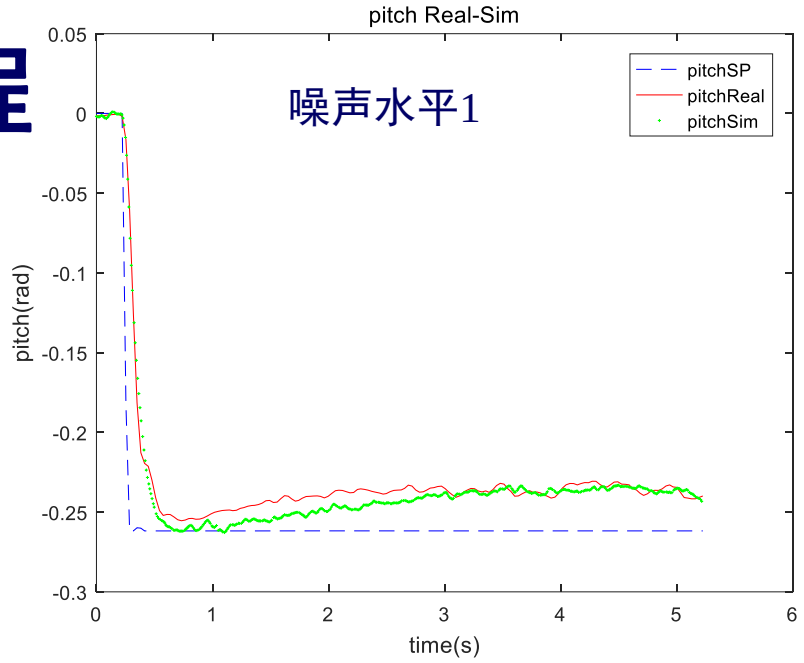
考虑到实际飞行的不确定性，以及自身生成的控制算法缺乏完整的失效保护逻辑，在实际飞行时应该充分考虑安全性问题。实际飞行实验应该选在相对空旷的区域（例如草地），同时保证天气良好，风速较低。在上述条件满足情况下，将电池连接到Pixhawk自驾仪上，并按下Pixhawk上安全开关超过三秒钟，然后用遥控器控制多旋翼来验证控制器的实际效果。

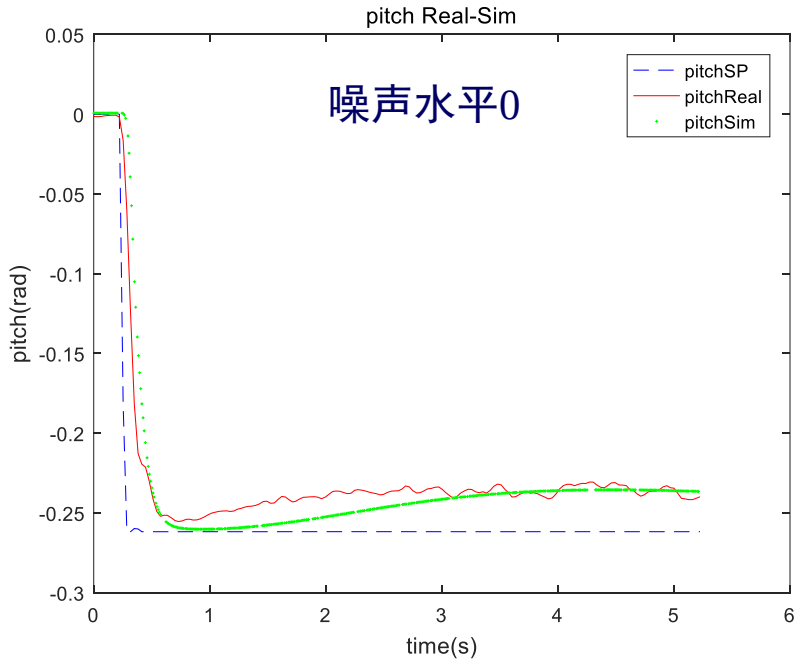
姿态控制实验操作具体流程

□ 实际飞行实验阶段与结果对比

（16）步骤十六：测试结果及分析

读取实际飞行和半物理仿真的log数据。右上为设置噪声水平为1，可以看到半物理仿真阶跃响应与实飞阶跃响应无论是动态过程还是噪声水平都比较接近。图右下设置噪声水平为0，可以看到半物理仿真下解算的角度没有噪声，动态过程与实飞接近。注意，由于仿真模式的机架类型“HIL Quadcopter X”与实际飞行所用的“DJI Flame Wheel F450”并不完全一致，其控制器参数也存在区别，因此响应曲线存在误差是正常的。同时，实际飞行时多旋翼的气动非常复杂，而模型中用了简化的气动模型，因此在最终的角度稳态响应曲线上存在一定的误差是可以接受的。





谢谢！