RflySim底层飞行控制算法开发

系列课程

第2讲　实验流程及相关接口介绍

飞思实验室

大纲

1. 控制LED灯实验操作具体流程
2. 姿态控制实验操作具体流程
3. 其他接口类实验
4. 小结

控制LED灯实验操作具体流程

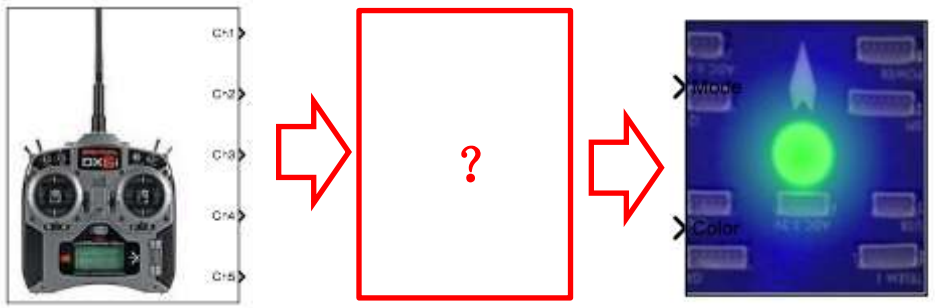
LED灯实验目标

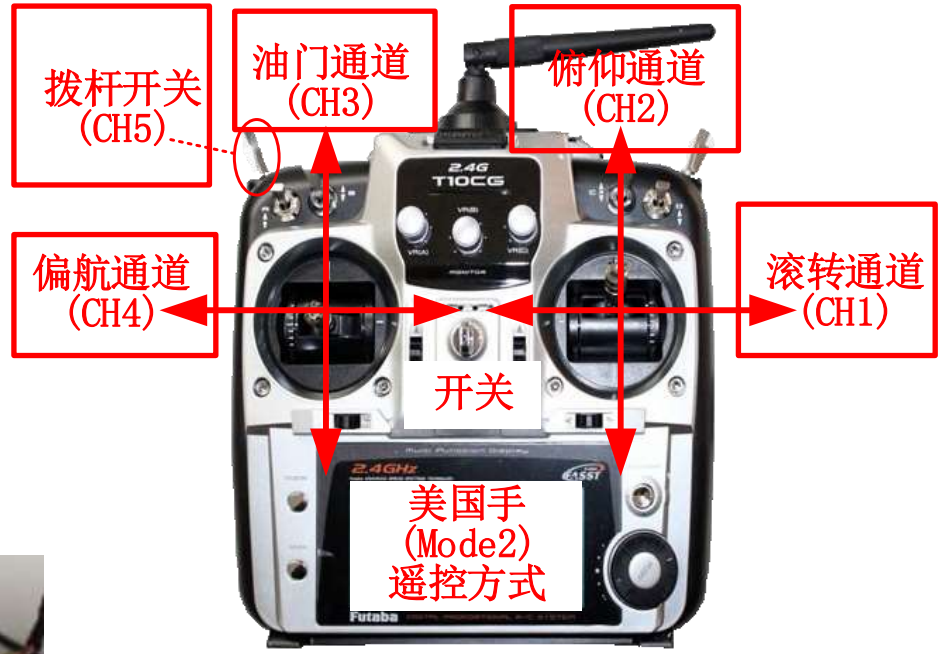
使用遥控器CH1~CH5

任意两个通道实现让

LED灯以两种颜色和两

种模式闪烁。





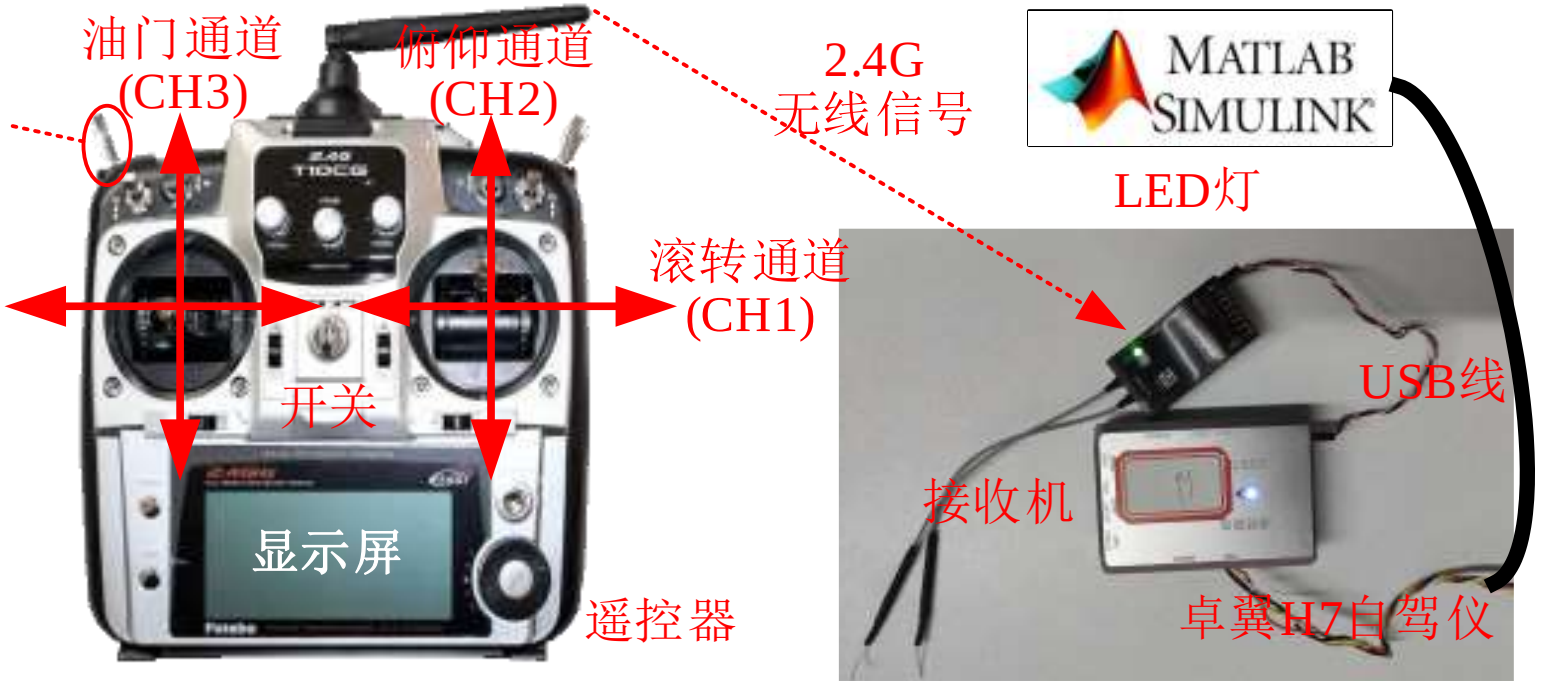
油门：控制上下运动，对应固定翼油门杆
偏航：控制机头转向，对应固定翼方向舵
俯仰：控制前后运动，对应固定翼升降舵
滚转：控制左右运动，对应固定翼副翼

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。

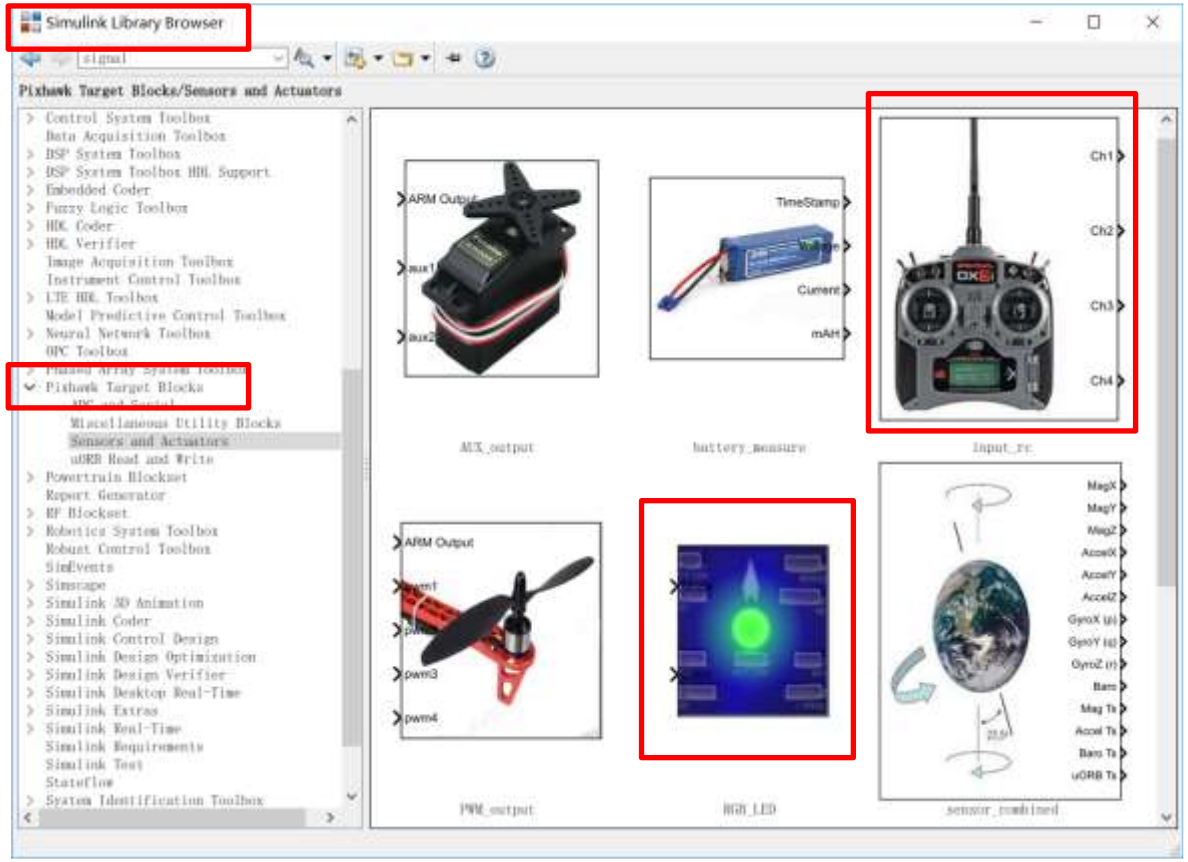
硬件连接图



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“RflySimAPI\Exp02_FlightControl\e0-PlatformStudy\2.PSPOfficialExps\RC-LED-Ctrl\px4demo_input_rc.slx”，实验具体操作见文件readme.pdf。



控制LED灯实验操作具体流程

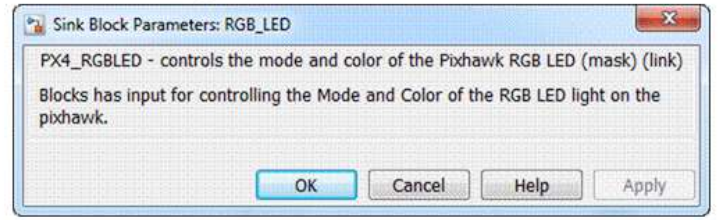
□ 设计LED灯控制模型

（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卓翼H7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.



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)

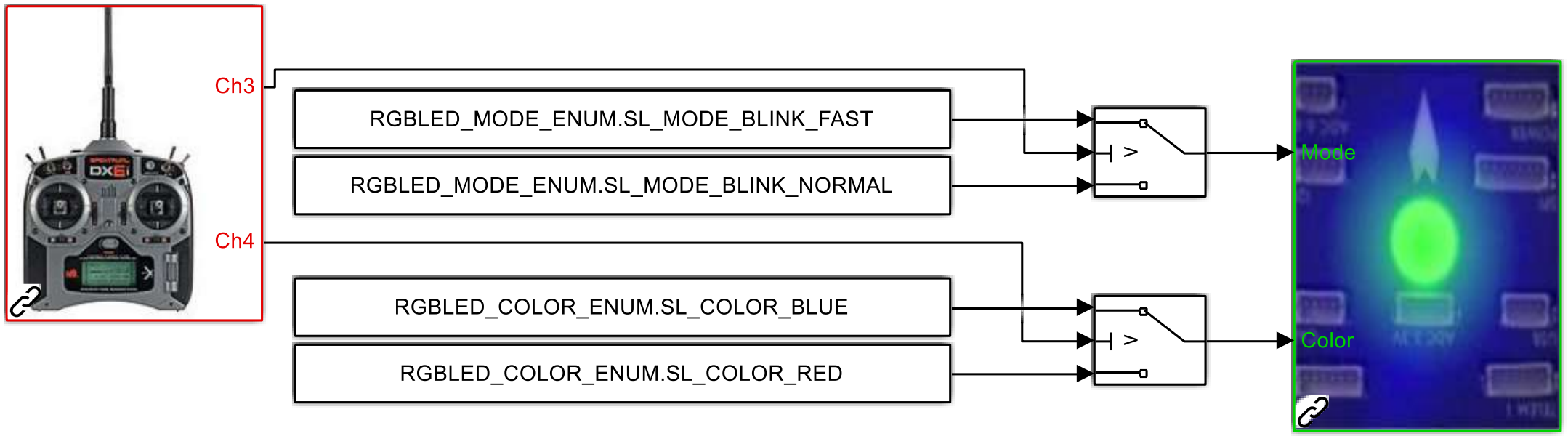
RGBLED_COLOR_ENUM

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

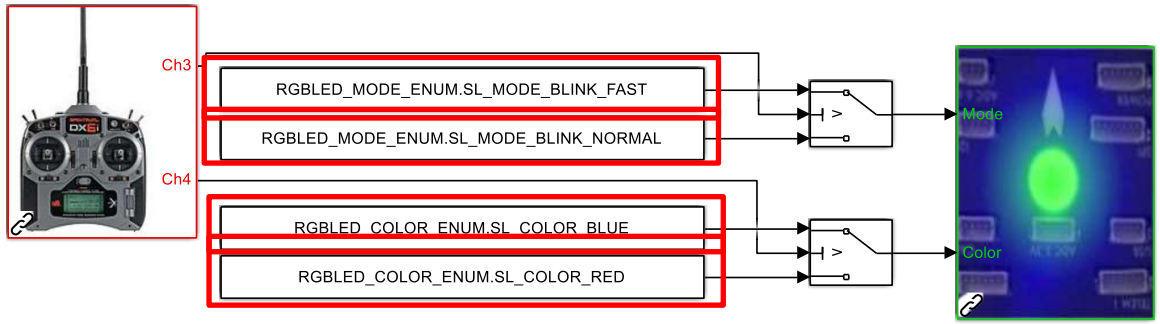
□ 设计LED灯控制模型

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。

大纲

1. 控制LED灯实验操作具体流程
2. 姿态控制实验操作具体流程
3. 其他接口类实验
4. 小结

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-PlatformStudy\3.DesignExps\Exp1_AttitudeController.slx”,具体实验操作见文件readme.pdf

□ 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：

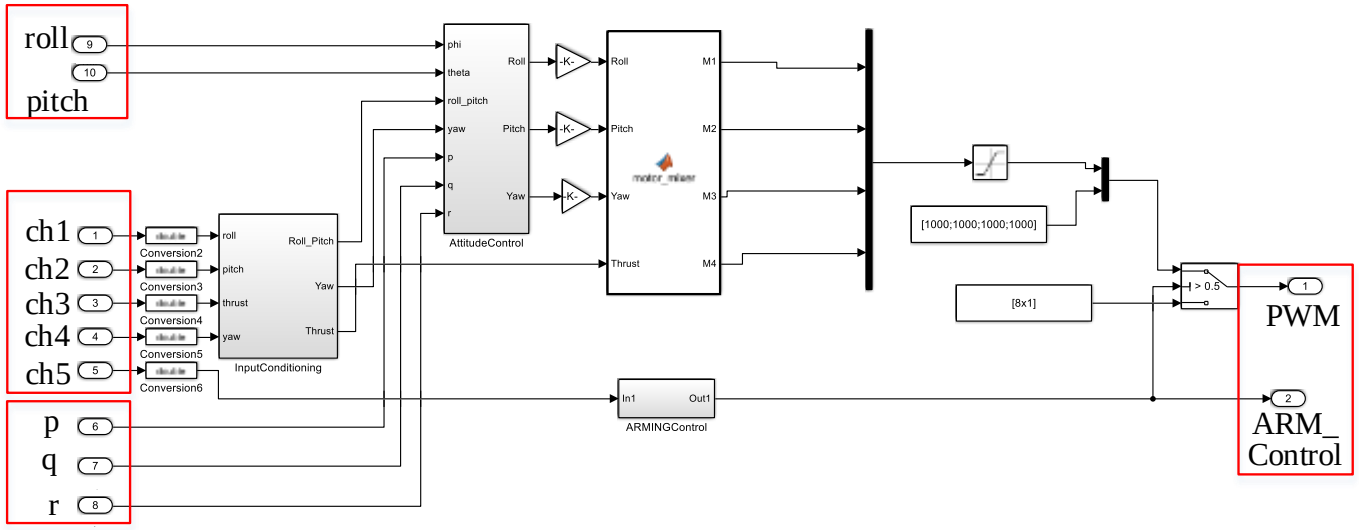
- 输入数据：1）遥控器Ch1~Ch5通道信号，数据范围大约为1100-1900，在处理遥控器数据时需要校准或考虑死区；2）角速度反馈量AngRateB，三个分量用p,q,r表示（单位：rad/s），分别代表滚转角速度（沿机体x轴转动）、俯仰角速度（沿机体y轴转动）和偏航角速度（沿机体z轴转动）；3）多旋翼欧拉角（单位为rad）。这里主要考虑滚转角和俯仰角，暂不考虑偏航控制。
- 输出数据：1）四个电机的PWM控制信号，数据范围1000~2000；2）是否解锁标识符，数据类型bool型
- 实现效果：CH3油门通道控制飞机升降；向前推俯仰摇杆（即CH2<1500）控制多旋翼向前飞；向左推滚转摇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

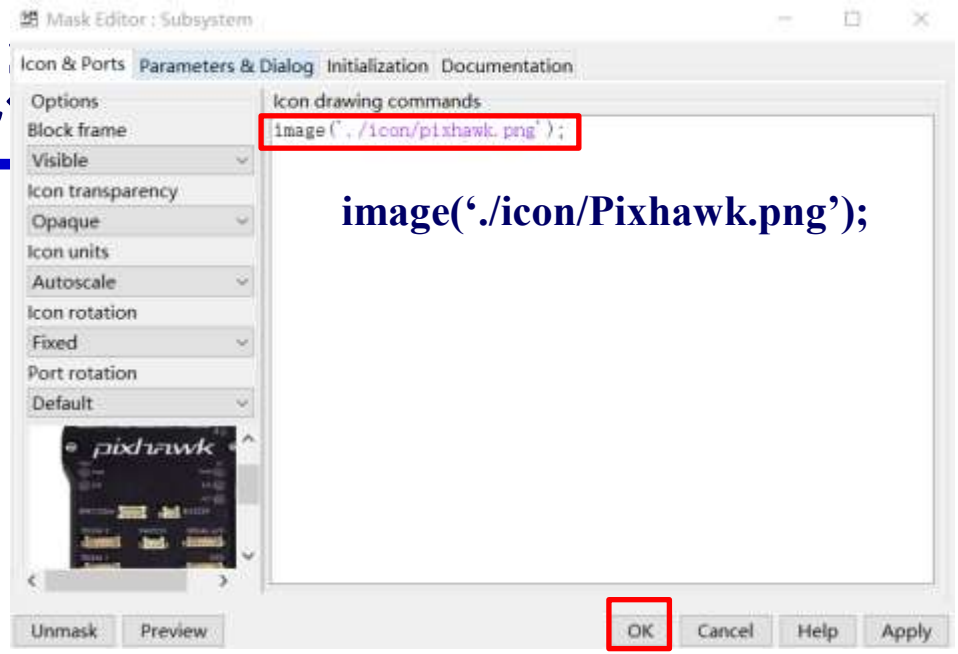


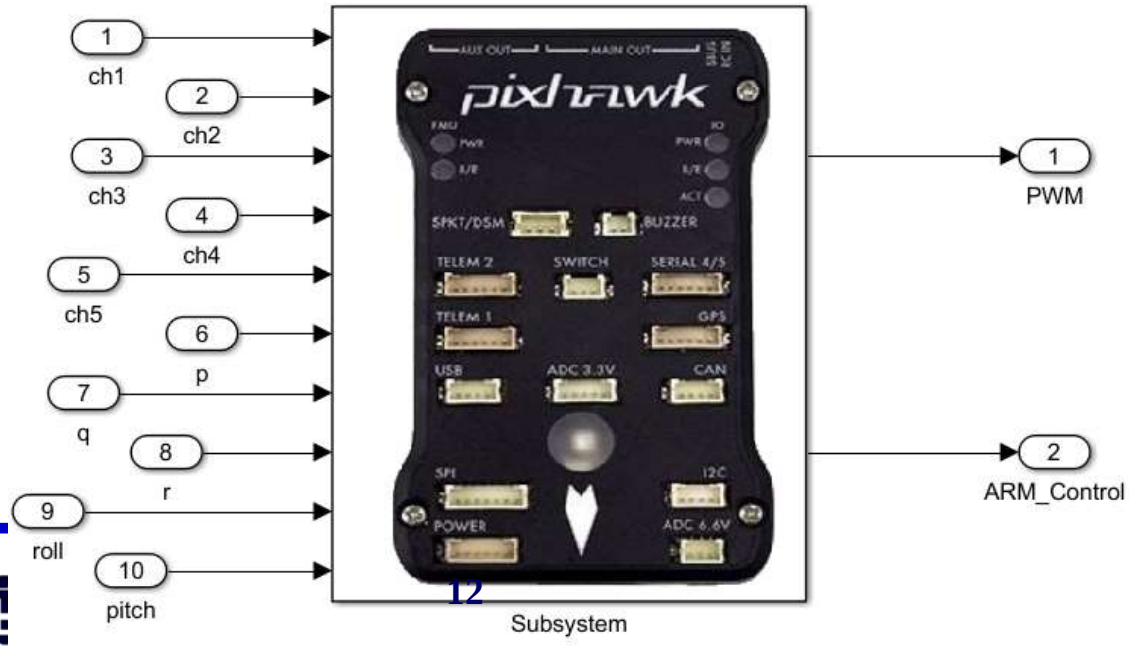
姿态控制实验操作具体流

□ 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、模型及RflySim显示模块整合

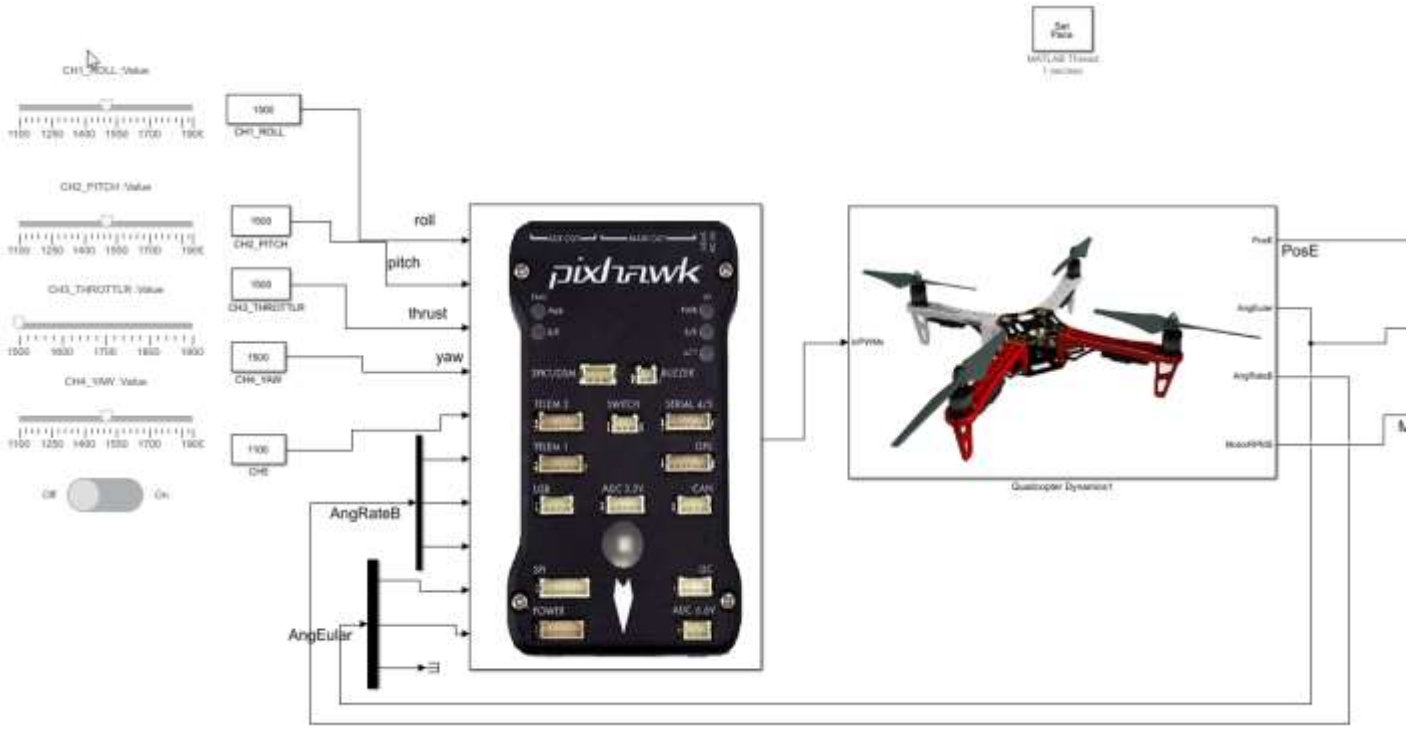
通过前文对控制器的设计，我们需要设置控制器的输入和输出，输入为链接模拟遥控器的CH1~CH5和反馈量，输出为PWM，PWM作为模型的输入，模型输出飞机的状态量分别作为控制器的反馈量和RflySim3D显示模块的输入量。

姿态控制实验操作具体流程

□ 算法设计与仿真阶段

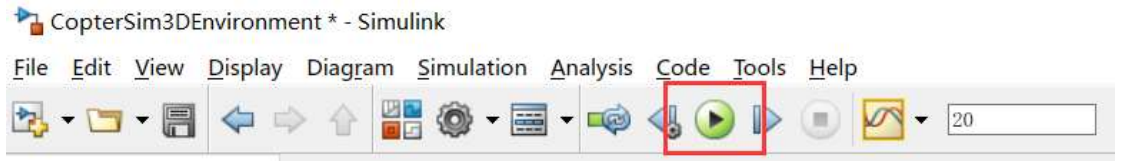
（4）步骤四：连线与输入输出配置

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-PlatformStudy\3.DesignExps\Exp2_ControlSystemDemo.slx”，文件内部见右图。



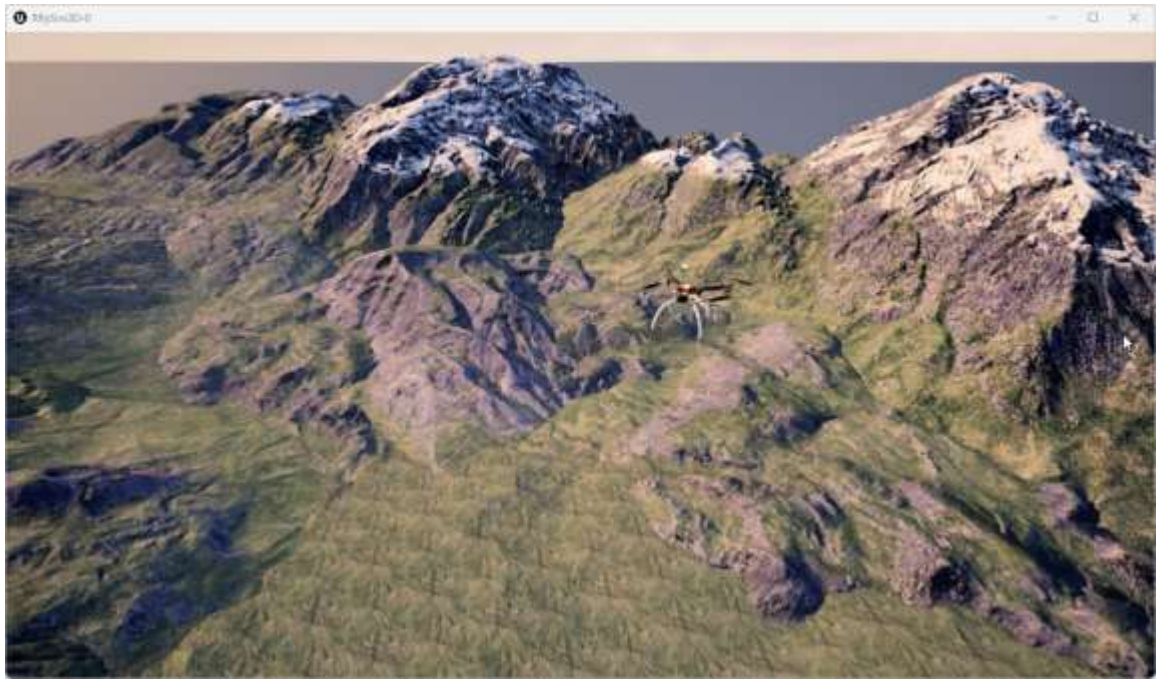
姿态控制实验操作具体流程

□ 算法设计与仿真阶段



（5）步骤五：开始联合仿真

双击文件“Rflysim3D”打开,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Rflysim3D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。

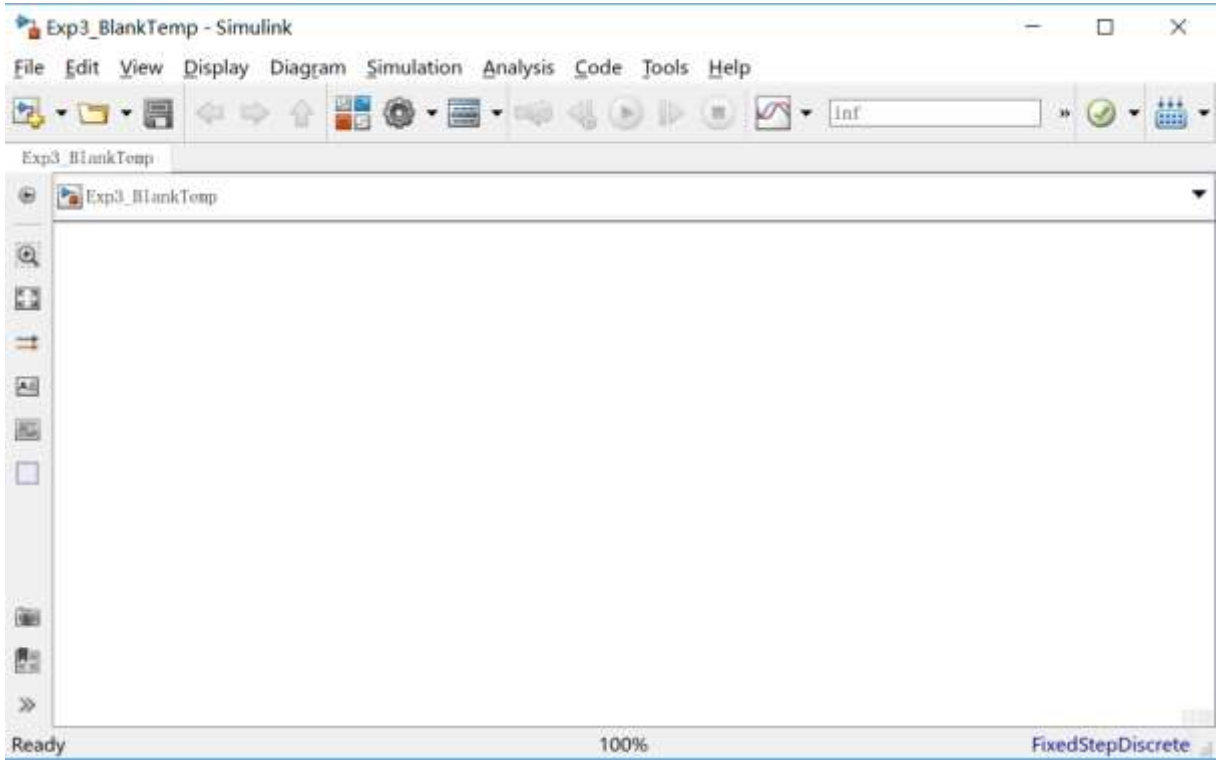


姿态控制实验操作具体流程

□ 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-PlatformStudy\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）。



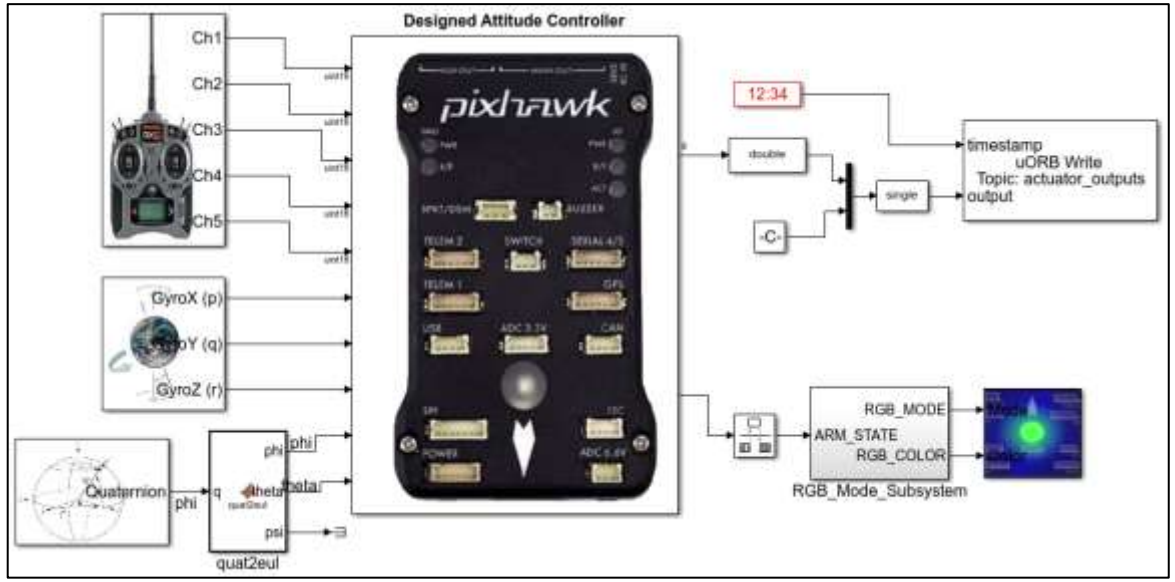
姿态控制实验操作具体流程

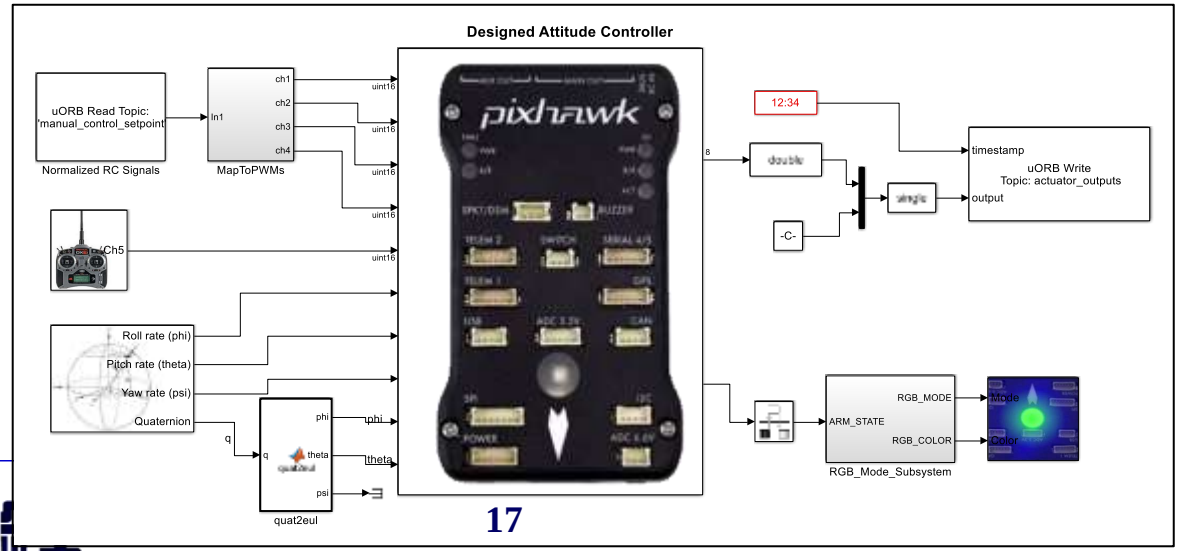
□ 代码生成与配置阶段

（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-PlatformStudy\3.DesignExps\Exp4_AttitudeSystemCodeGen_old.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。

上面的例子直接读取了遥控器原始数据，而右下图的“Exp4_AttitudeSystemCodeGen.slx”例子采用了读取“mannual_control_setpoint”的uORB消息来获取校准并归一化后的遥控器数据（俯仰滚转偏航：-1~1，油门：0~1）而不是直接读取遥控器PWM信号，这种方式更方便可靠且可以兼容任意遥控器而不需要担心校准问题





姿态控制实验操作具体流程

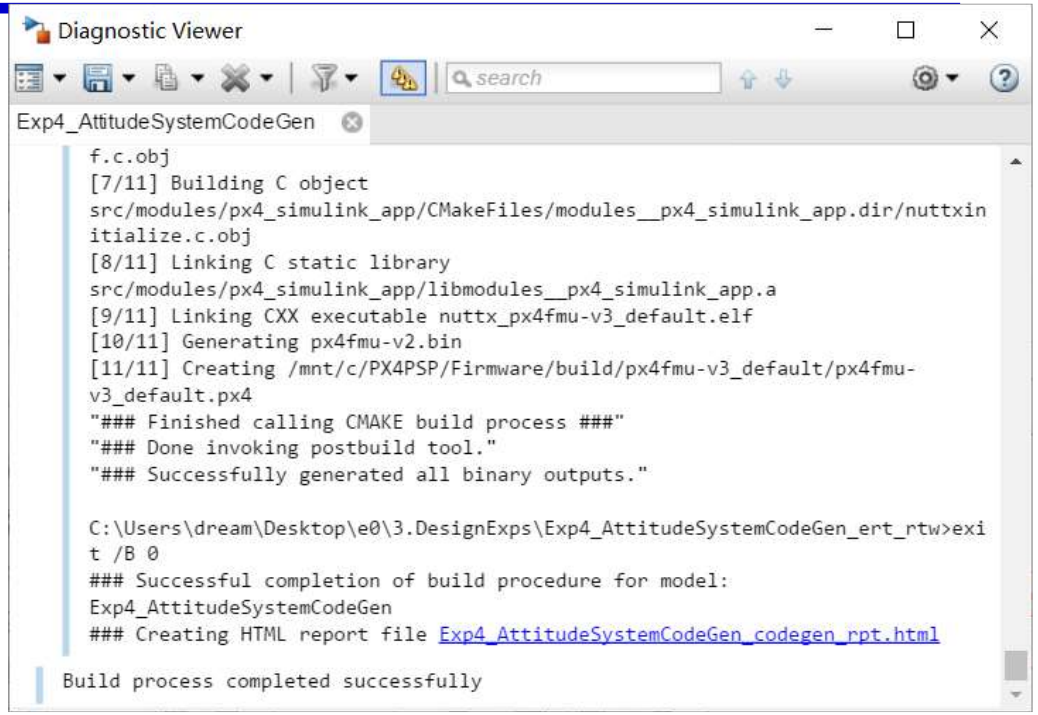
□ 代码生成与配置阶段

（8）步骤八：编译并产生固件

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。

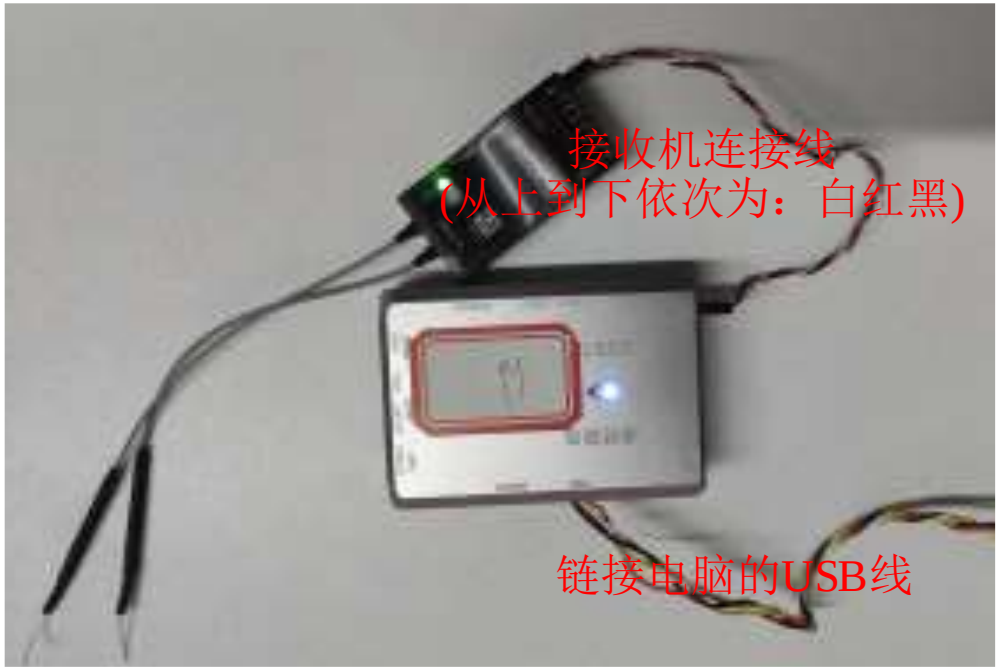


姿态控制实验操作具体流程

□ 硬件在环仿真阶段

（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卓翼H7，然后卓翼H7与电脑通过USB数据线连接，此时可以看到卓翼H7上的蓝灯亮起，接收机上的灯光为红色常亮。此时打开遥控器开关（油门杆拉到最低位置），可以观察接收机上的灯光变为蓝色常亮。



姿态控制实验操作具体流程

□ 硬件在环仿真阶段

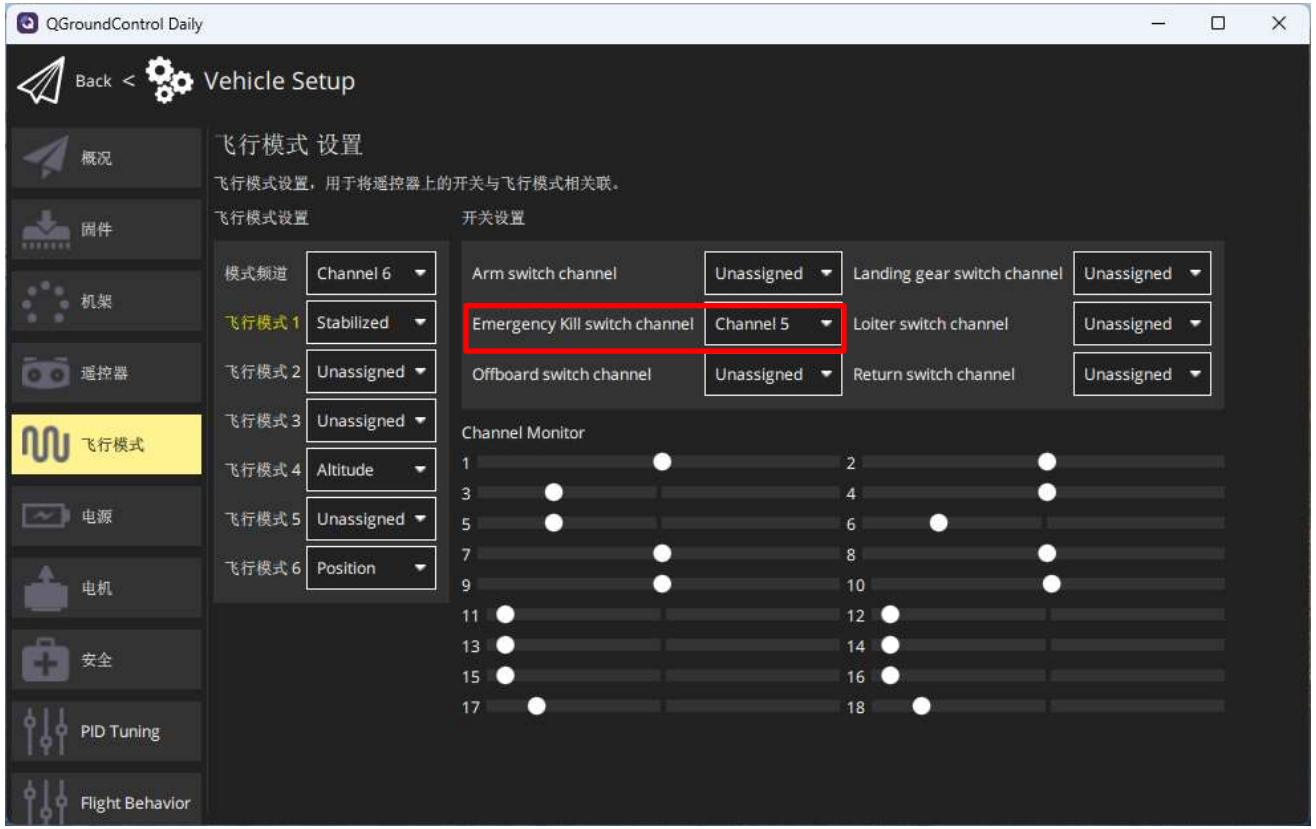
（11）步骤十一：模型仿真器软件配置

打开QGC地面站，连接上卓翼H7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“Emergency Kill switch channel”切换开关为CH5。

3）关闭QGC地面站

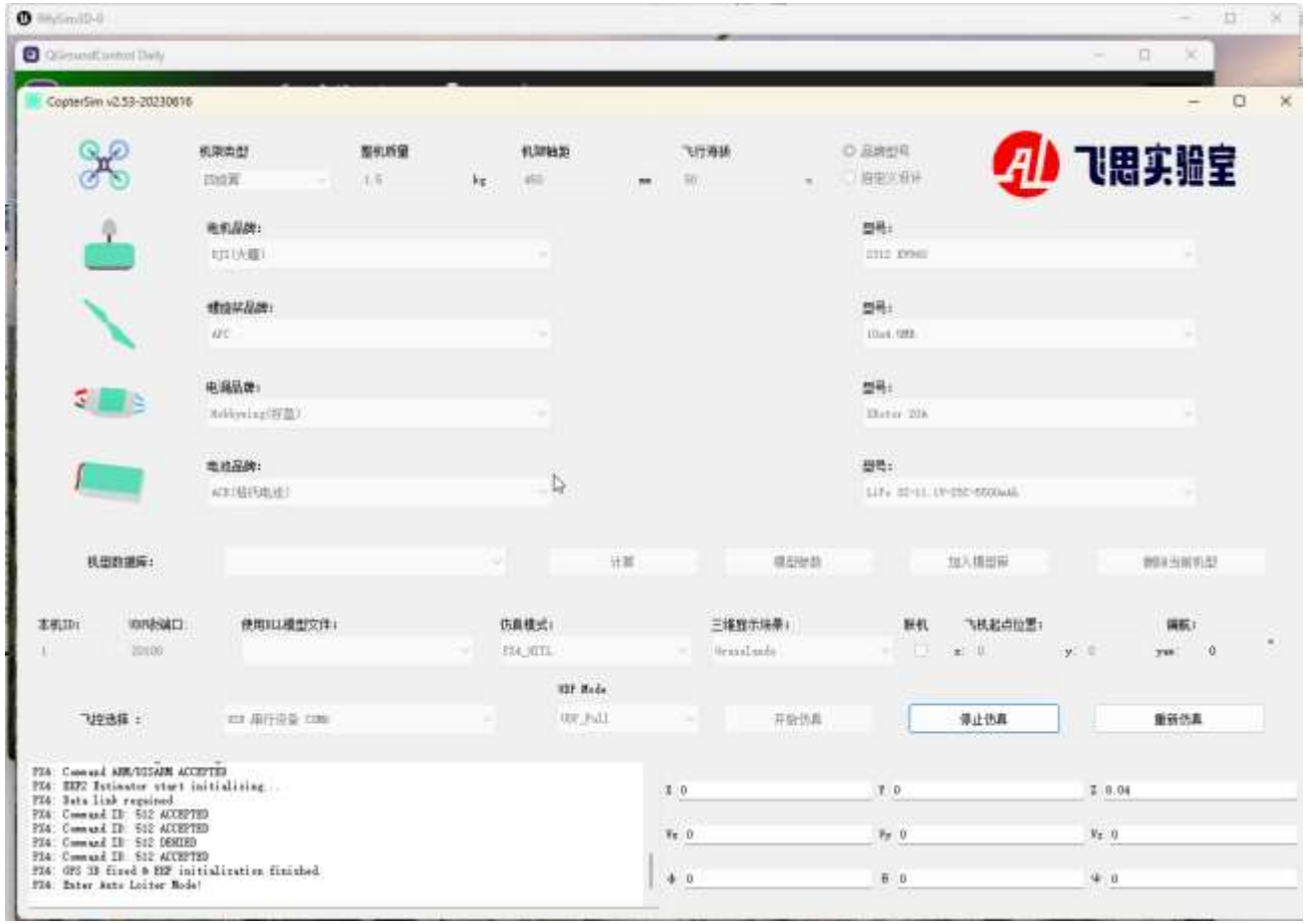


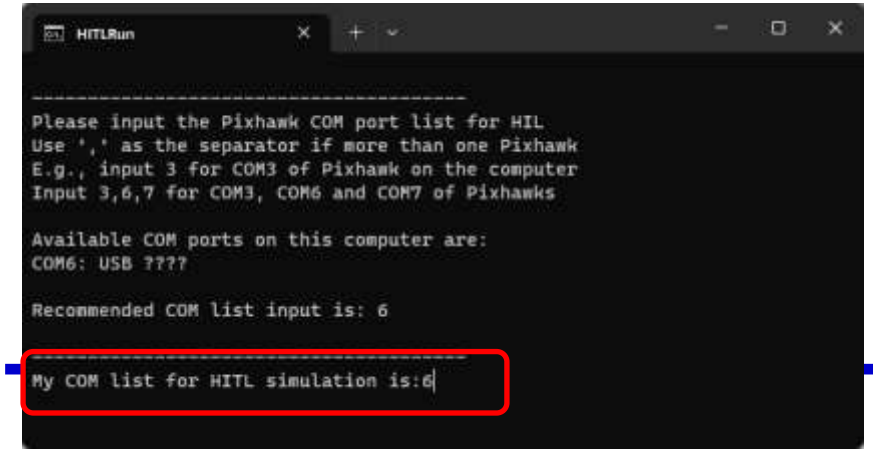
姿态控制实验操作具体流程

□ 硬件在环仿真阶段

（11）步骤十二：一键启动硬件在环仿真

飞控通过USB线链接电脑，双击桌面的HITLRun快捷方式，在弹出对话框中输入飞控的COM端口，电脑将自动打开RlySim3D、QGroundControl以及CopterSim软件，实现一键启动硬件在环仿真。





姿态控制实验操作具体流程

□ 硬件在环仿真阶段

（12）步骤十三：解锁飞行

等待CopterSim显示“PX4: GPS 3D fixed & EKF initialization finished”，即可控制遥控器CH5拨杆开关解锁多旋翼，便可操纵遥控器使多旋翼完成相应动作。如右图所示， RflySim3D即可显示多旋翼飞行状态。

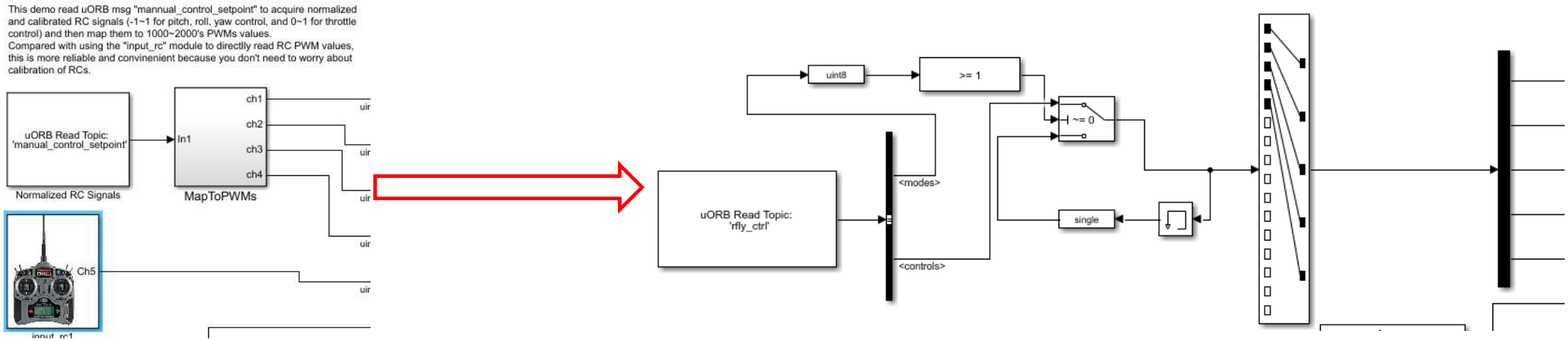
```
PX4: Command REQUEST_AUTOPILOT_VERSION ACCEPTED
PX4: Command ARM/DISARM ACCEPTED
PX4: EKF2 Estimator start initializing...
PX4: GPS 3D fixed & EKF initialization finished.
PX4: Command ID: 512 ACCEPTED
PX4: Data link regained
PX4: Command ID: 512 ACCEPTED
PX4: Command ID: 512 DENIED
PX4: Command ID: 512 ACCEPTED
```

姿态控制实验操作具体流程

□ 转台实验阶段

（12）步骤十四：模型修改1

基于对所设计的模型进行硬件在环仿真之后，在进行实飞之前需在台架上进行测试，将硬件在环仿真的模型的输入替换为RflySim平台自定义的uORB名称为“rfly_ctrl”的消息，烧录之后，将通过MAVLINK协议用于飞控与上位机之间进行通信。

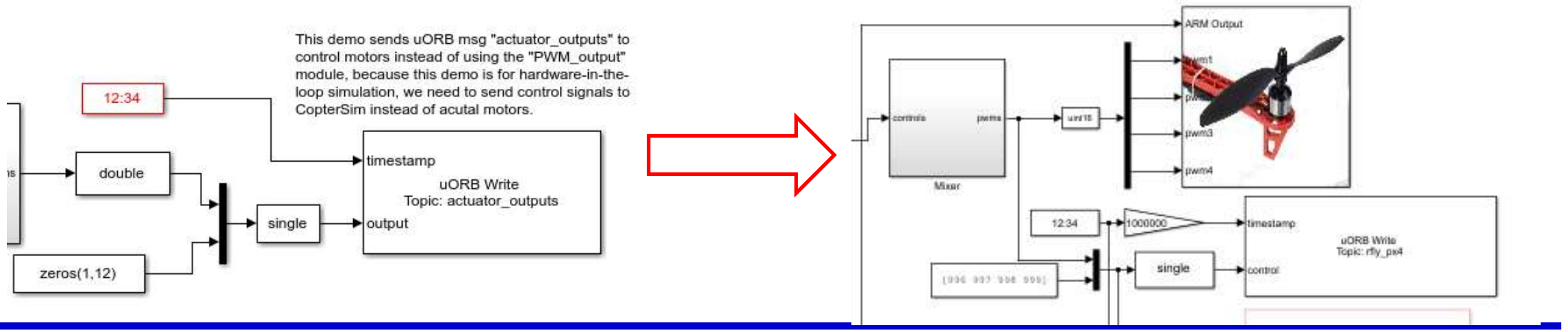


姿态控制实验操作具体流程

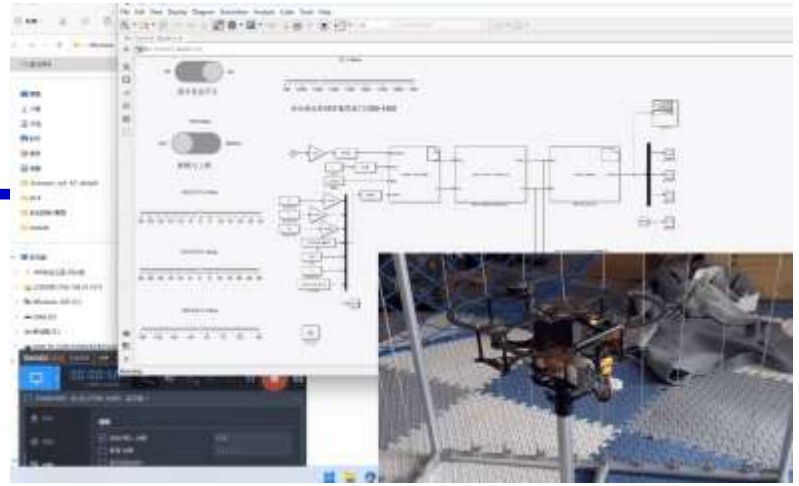
□ 转台实验阶段

（12）步骤十五：模型修改2

基于对所设计的模型进行硬件在环仿真之后，在进行实飞之前需在台架上进行测试，将硬件在环仿真的模型的输出替换为**RflySim**平台**PSP**工具箱中的电机模块，并添加**uORB**名称为“**rfly_px4**”的消息，烧录之后，将通过**MAVLINK**协议可在上位机程序上显示飞机的飞行状态。



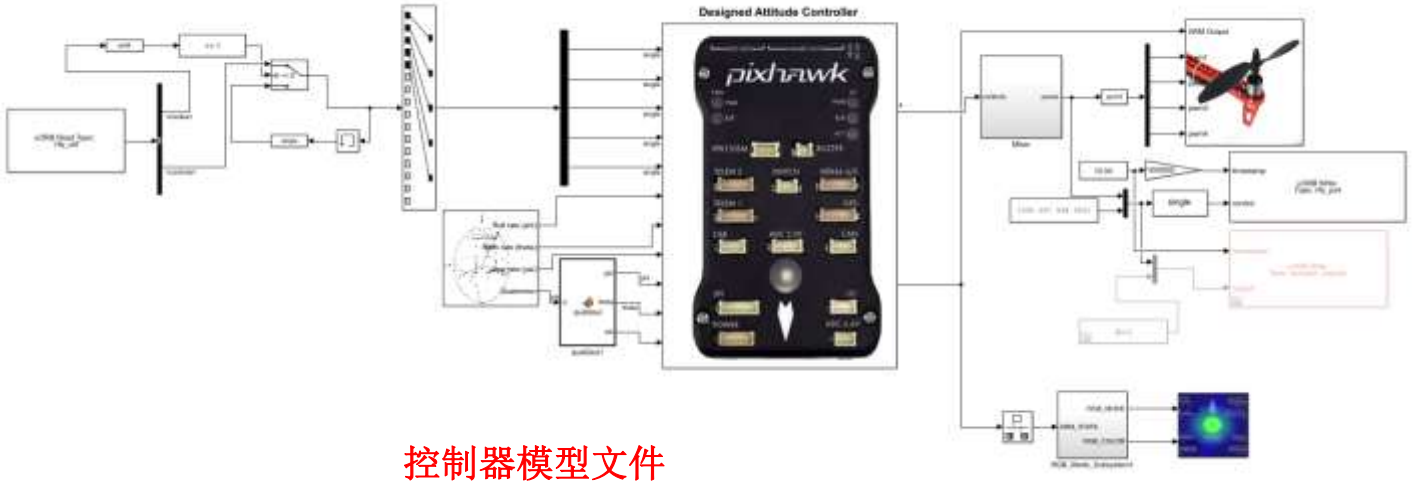
姿态控制实验操作具体流程



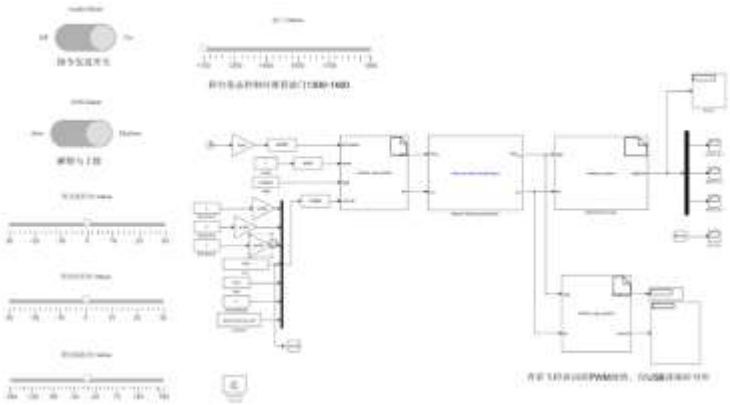
□ 转台实验阶段

（12）步骤十六：转台实飞步骤

将前文修改的通过自动代码生成的方式烧录到飞机当中，给飞机上电，本地端电脑链接数传后，在Simulink中打开上位机控制文件，设置COM端口后，即可解锁进行实验。



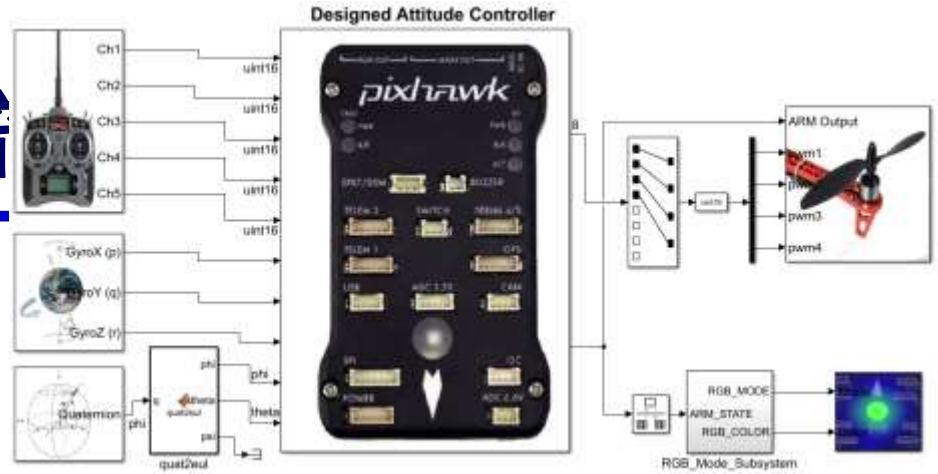
控制器模型文件



Simulink上位机控制文件

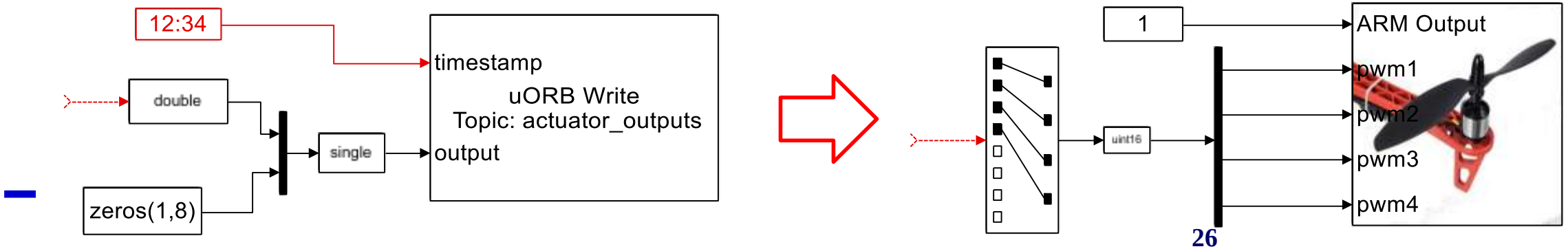
姿态控制实验操作具体流

□ 实际飞行实验阶段与结果对比



（14）步骤十七：调整Simulink控制器

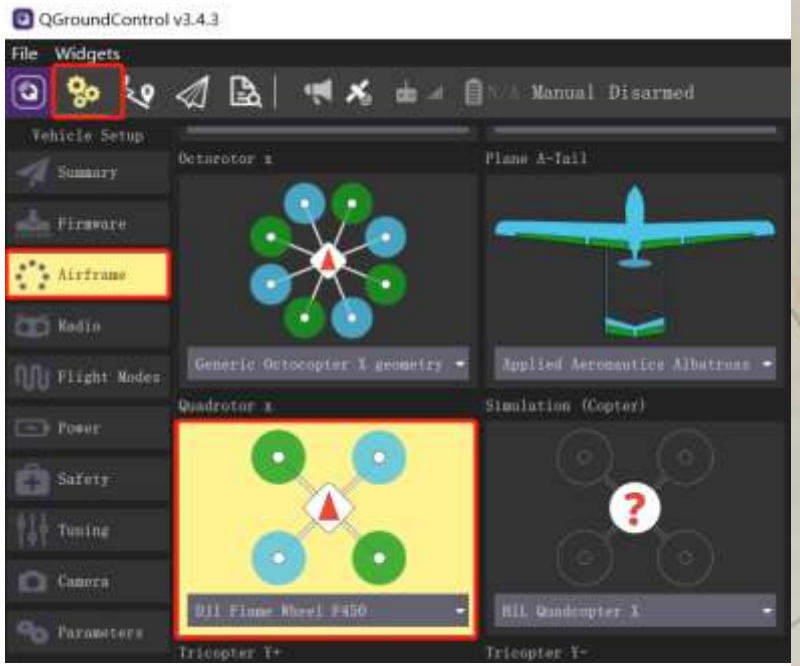
打开Simulink文件，按下图所示将uORB的输出改成PSP工具箱提供的PWM_out输出模块，重新生成代码并下载到卓翼H7中。这里给出了一个例子，见文件“e0-PlatformStudy\3.DesignExps\Exp5_AttitudeSystemCodeGenRealFlight_old.slx”（或使用不带old后缀的slx例程）。注意，该例子在真机飞行时，不需要通过遥控器油门杆解锁，但是需要拨动CH5开关来解锁（有时需要拨动两次）。注：一些特殊自驾仪不支持PWM_out模块，可参照“_all.slx”后缀例程使用actuator_controls_0的uORB消息来输出控制信号（归一化的力和力矩）到PX4混控器mixer，从而间接实现电机的控制。



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）步骤十八：飞机组装。本次课程线下实飞飞机为飞思X450飞机，见右图；飞机出厂前已对进行各类校准和参数调试。在实际飞行时需要在QGC中将卓翼H7的机架类型从“HIL Quadcopter X”修改为“DJI Flame Wheel F450”，导入官方提供的飞机参数文件，检查飞机各类传感器情况，确认无误后即可进行试飞。



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）步骤十九：参数设置和测试

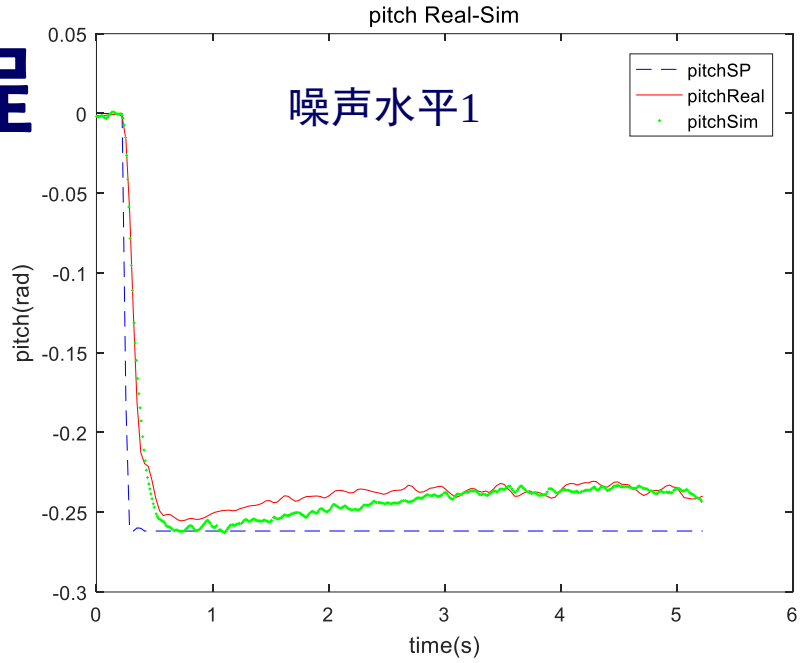
考虑到实际飞行的不确定性，以及自身生成的控制算法缺乏完整的失效保护逻辑，在实际飞行时应该充分考虑安全性问题。实际飞行实验应该选在相对空旷的区域（例如草地），同时保证天气良好，风速较低。在上述条件满足情况下，请先烧录官方提供的飞思X450飞机版本所对应的固件和参数，进行试飞，确认飞机可正常起飞后，再烧录本次实验所设计的控制器模型生成的固件，用遥控器控制多旋翼来验证控制器的实际效果。

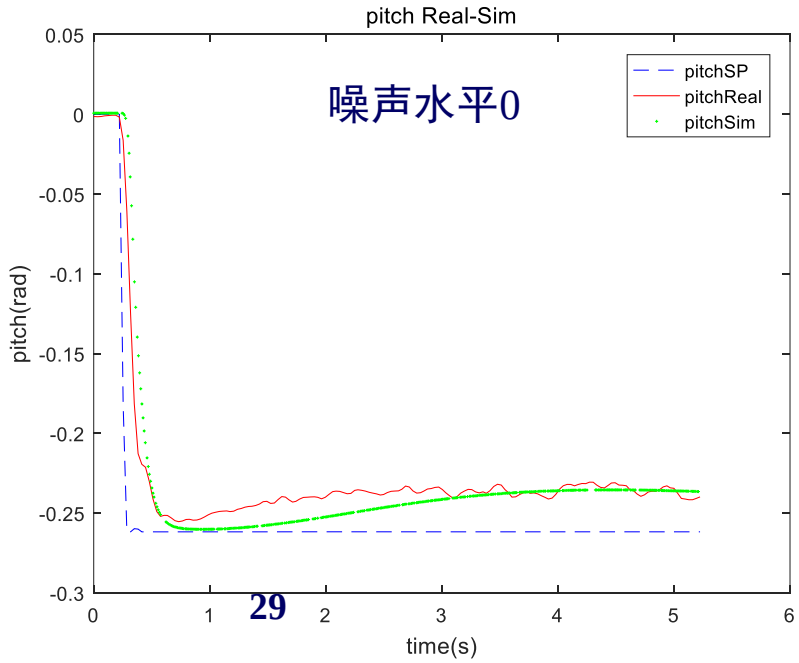
姿态控制实验操作具体流程

□ 实际飞行实验阶段与结果对比

（16）步骤二十：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。





飞思实验室

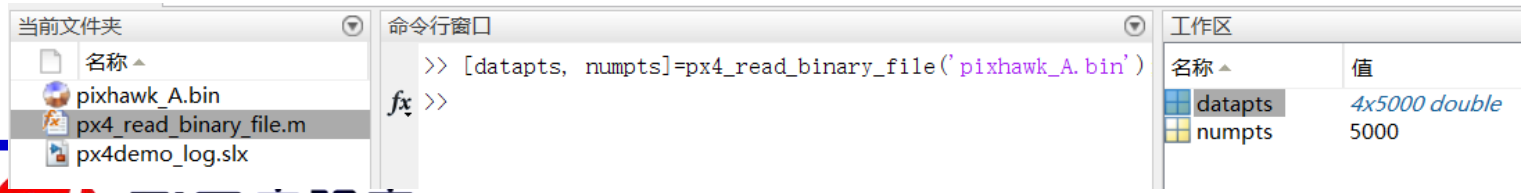
大纲

1. 控制LED灯实验操作具体流程
2. 姿态控制实验操作具体流程
3. 其他接口类实验
4. 小结

其他接口类实验

Log数据记录与读取

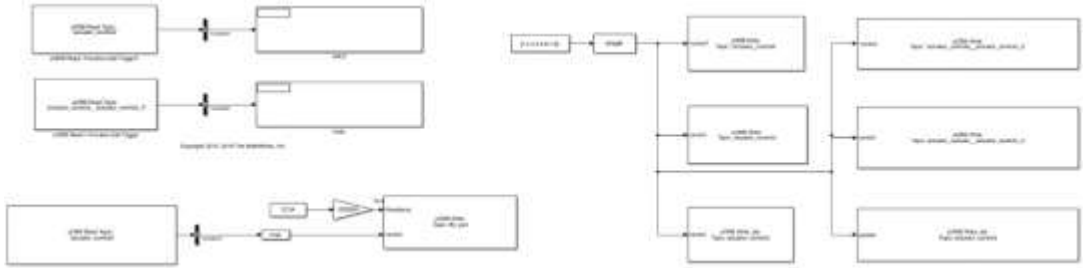
1. 用MATLAB打开例程文件夹为“RflySimAPIs\Exp02_FlightControl\e0-PlatformStudy\5.Log-Write-Read”，具体实验操作见文件readme.pdf。
2. 本例程使用了“binary_logger”模块，设定了记录20s（程序步长250Hz，因此有5000个数据点）的四维随机数据，数据存储位置“/fs/microsd/log/pixhawk”。
3. 按前面实验相同步骤，将slx编译并上传Pixhawk
4. 飞控重启后等待20s以上，断电，拔出内存卡，将其插入电脑，可以在log文件夹看到“Pixhawk_A.bin”的日志文件（_A.bin后缀是系统自动加的）
5. 将Pixhawk_A.bin拷贝到slx的目录下来，运行如下指令： [datapts, numpts]=px4_read_binary_file(‘pixhawk_A.bin’);即可得到四维5000个点的数据。
6. 再用MATLAB进行数据处理即可。



其他接口类实验

uORB通信的读写

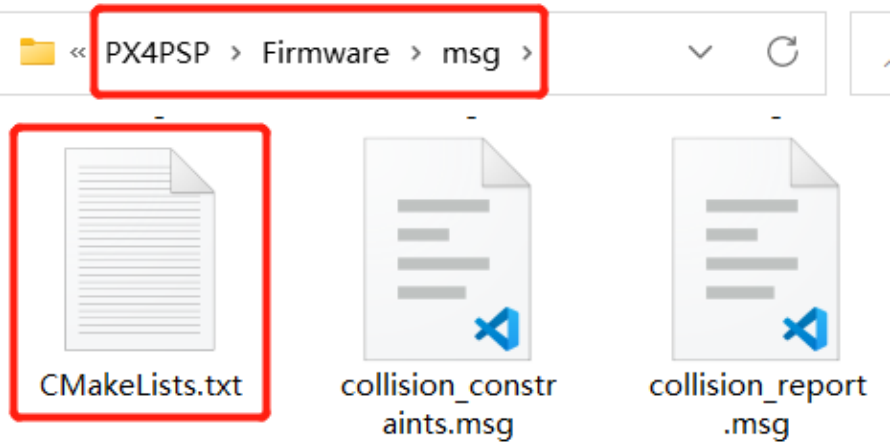
1. 打开例程文件夹“RflySimAPIs\Exp02_FlightControl\e0-PlatformStudy\6.uORB-Read-Write”，具体实验操作见文件readme.pdf。
2. PX4的uORB消息系统是提供了非常强大且方便的内部模块间数据交互能力，所有模块都可以将数据放在消息池中，其他模块可以从消息池订阅到所需数据。
3. 打开px4demo_uORB_read_write.slx例程文件，可以看到用uORB订阅和发布了一些消息。
4. uORB模块的使用方法，请参考基础版教程；本例程主要展示了，可以发布和订阅一些消息ID与消息文件名不符的情况，例如actuator_controls.msg的消息文件，可以读取或发送消息ID为actuator_controls_***的消息。

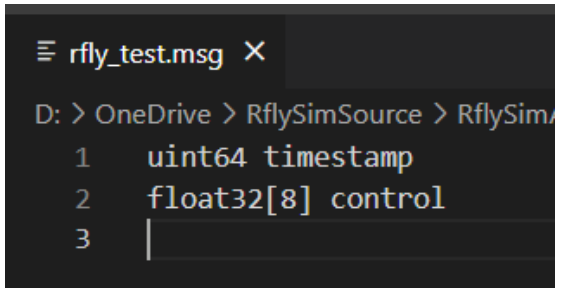


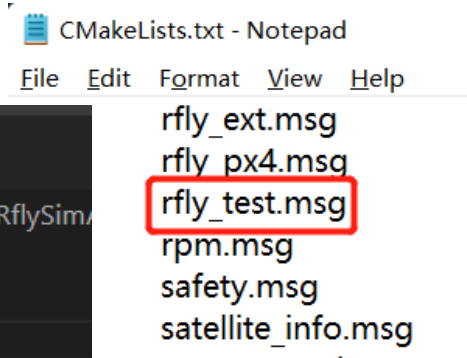
其他接口类实验

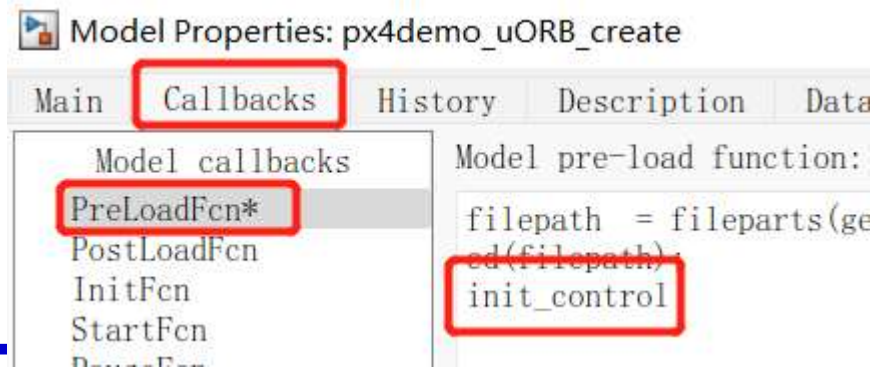
自定义uORB消息

1. 本例程文件夹“RflySimAPI\Exp02_FlightControl\e0-PlatformStudy\7.uORB-Create”，具体实验操作见文件readme.pdf。
2. PX4的若要新增uORB消息，需要在PX4PSP\Firmware\msg文件夹中新建.msg后缀的消息文件，并需要在CMakeLists.txt中添加消息名字。RflySim平台提供了一个自动化的脚本PX4uORBMsgGen.m来实现上述注册过程。
3. 在slx控制器文件同目录下新建一个文件夹（例如，msg），在其中新建自定义的.msg格式消息文件，再将本例程提供的PX4uORBMsgGen.m置于此文件夹即可。
4. 新建一个init_control，并在slx的File-Model Properties-Callbacks中添加打开slx文件时自动调用init_control脚本。
5. 在init_control.m加入如下代码，来调用PX4uORBMsgGen
6. addpath(‘.\msg’) % 将子目录添加到路径方便脚本调用
7. PX4uORBMsgGen %执行脚本PX4uORBMsgGen.m
8. 经过上述步骤，可以实现打开slx底层控制器文件时自动运行PX4uORBMsgGen脚本，它会自动搜索本目录下的所有.msg格式的uORB消息，并自动注册到Firmware/msg路径中。







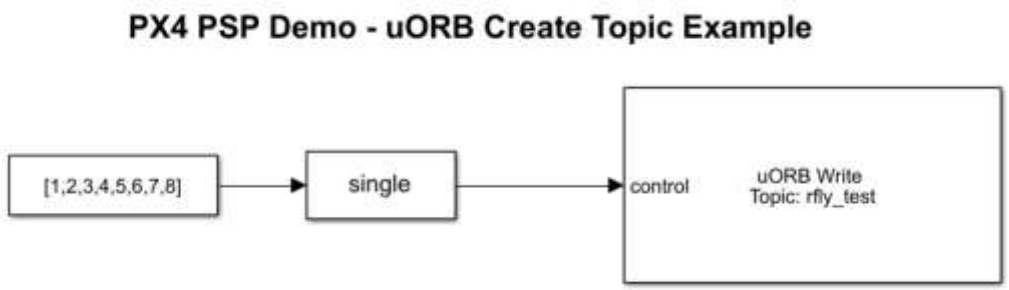


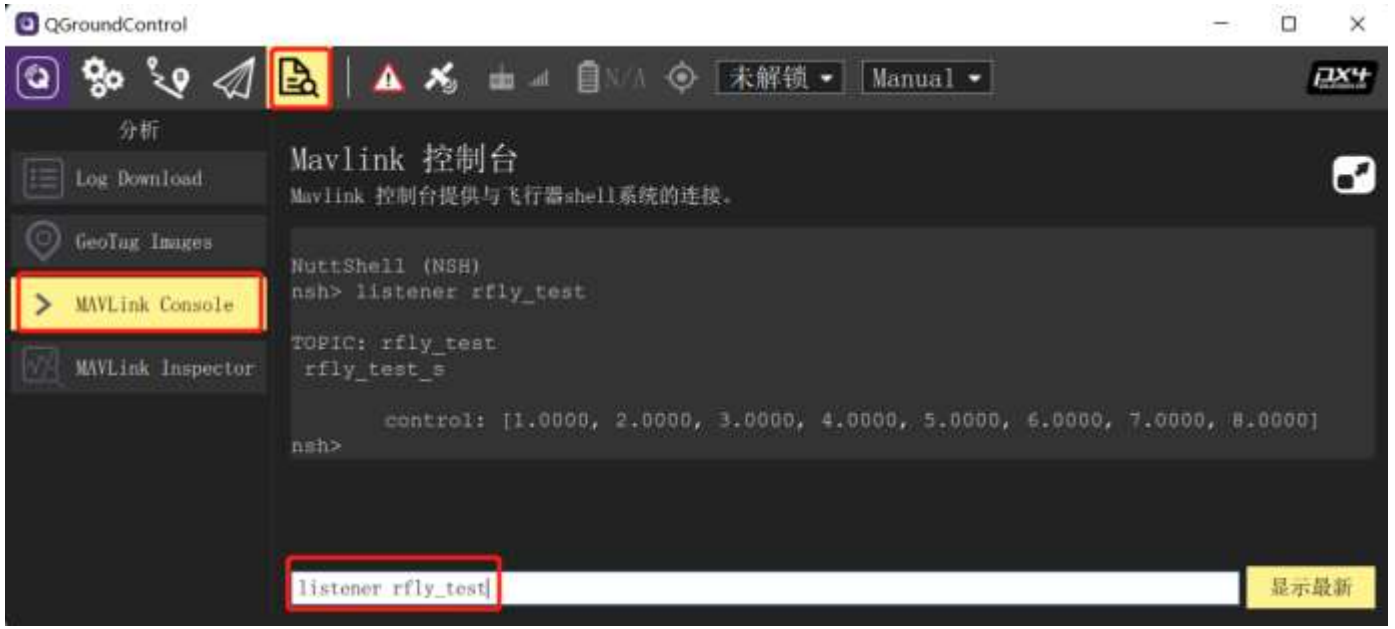
注：在MATLAB 2019b之后，挪到了MODELING-Model Settings-Model Properties-Callbacks中。

其他接口类实验

自定义uORB消息

具体实现效果如下：

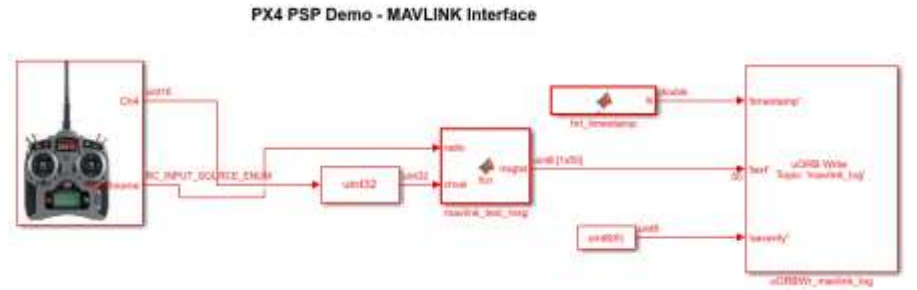


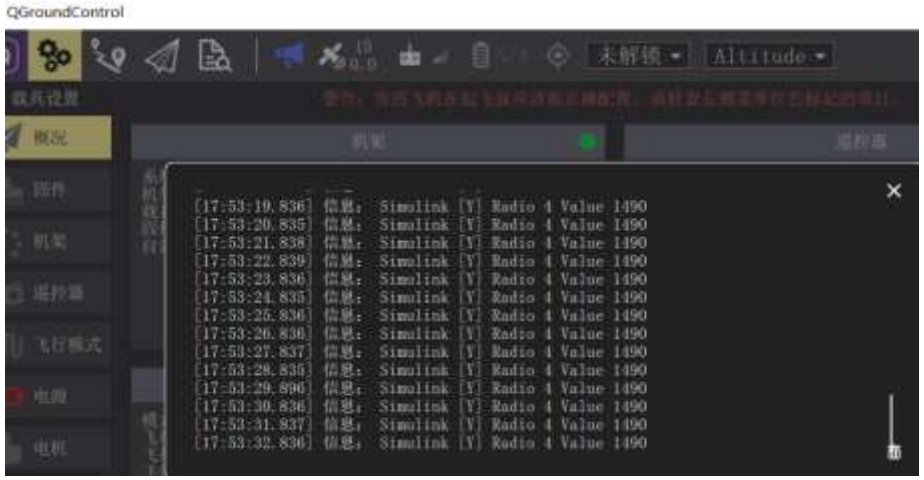


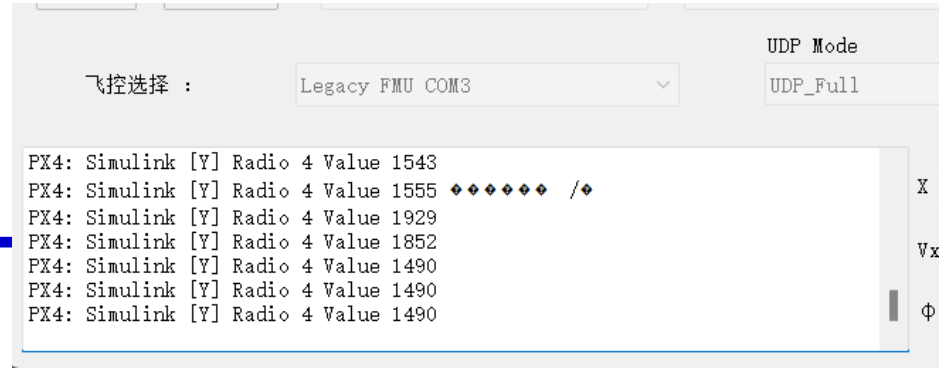
其他接口类实验

回传提示消息

1. 本例程文件夹“RflySimAPI\Exp02_FlightControl\e0-PlatformStudy\8.Mavlink-Msg-Echo”，具体实验操作见文件readme.pdf。
2. 在飞控中，我们时常需要向外发布一些文字消息，来反映系统当前的运行状态，这个功能可以通过发送“mavlink_log”的uORB消息来实现。
3. 打开px4demo_mavlink_rc.slx例程文件，编译并上传到Pixhawk中。
4. 运行桌面的HITLRun快捷方式，输入飞控串口号，开始硬件在环仿真。
5. 此时，可以在CopterSim的左下消息栏和QGC的消息栏接收到发送的文本信息，横向拨动遥控器油门杆，可以返回的信息变化。



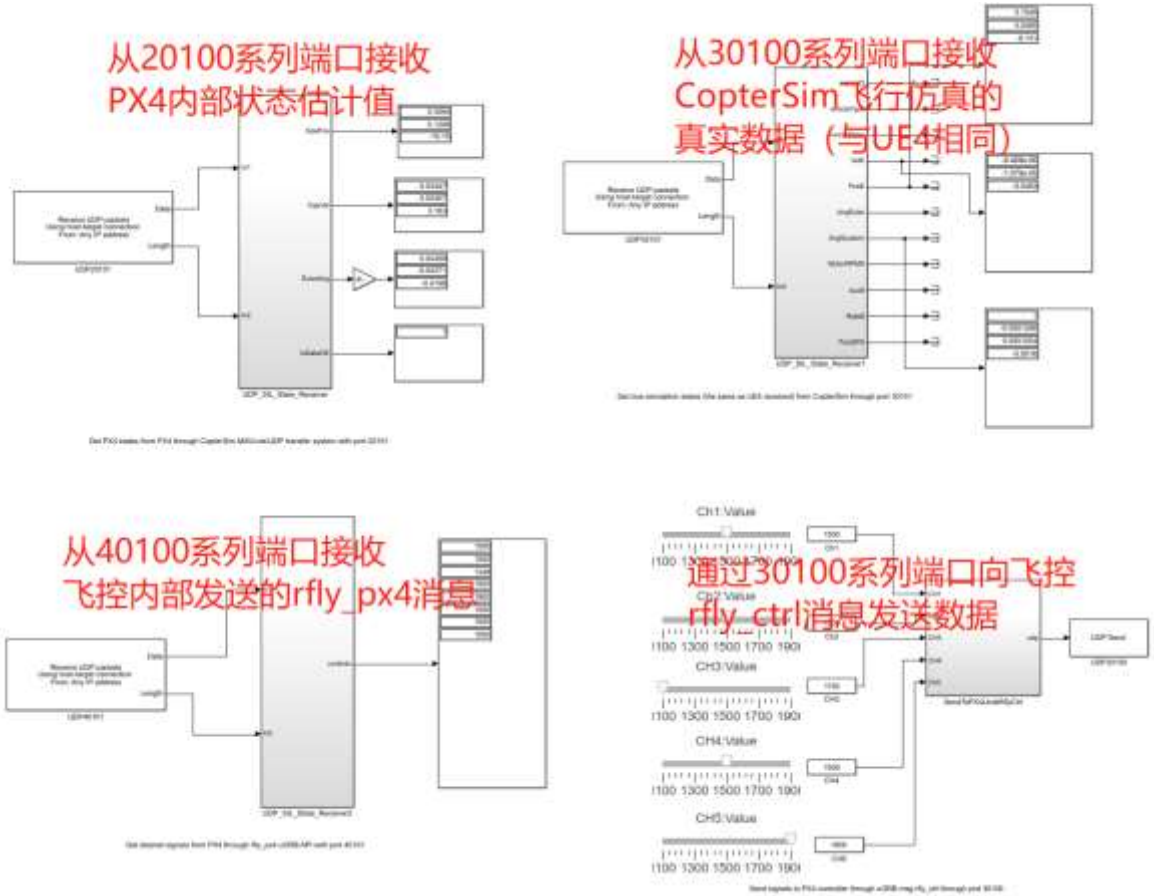




其他接口类实验

PX4控制器的外部通信

1. 本例程文件夹“RflySimAPI\Exp02_FlightControl\e0-PlatformStudy\9.PX4CtrlExternalTune”，具体实验操作见文件readme.pdf。
2. 在进行硬件在环仿真时，我们常常需要向设计的Simulink控制器中发送数据（传感器数据、故障触发、控制指令、参数调整等），同时接收一些感兴趣的数据。
3. RflySim平台的Simulink控制器设计功能，提供了rfly_ctrl这一uORB消息来接收外部数据（UDP发送指定结构体到CopterSim的30100系列端口），同时提供rfly_px4这一uORB消息来向外发送数据（向40100系列端口发送特定数据）。



注：本例子作为外界与Simulink控制值器的桥梁，能够大大提高控制器开发和参数调整的效率。

注：也可以手动修改Firmware的源码，将感兴趣的数据发送到rfly_px4，同时在需要的地方接收rlfy_ctrl的uORB消息，实现PX4原生控制器的通信。

其他接口类实验

PX4控制器的外部通信

4. 打开文件PX4ExtMsgReceiver.slx，编译并烧录到Pixhawk中。本例程以外部发送的rfly_ctrl数据来作为遥控器输入，同时会将收到的数据向rfly_px4发送出去，回传给外部程序。
5. 运行HITLRun，输入飞控串口号，开启硬件在环仿真。
6. 用Simulink运行PX4ExtMsgSender.slx程序，它的接收和发送的数据如上一页图。在右下角拨动CH3（模拟油门杆）可以实现飞机起飞。
7. 在PX4ExtMsgReceiver.slx中定义了左侧rfly_px4的数据与rfly_ctrl数据相同，实验结果与此一致。
8. 在QGC的MAVLink Inspector也可以通过ACTUATOR_CONTROL_TARGET观察到rfly_px4的值。

其他接口类实验

QGC实时调整控制器参数

1. 例程文件夹“RflySimAPI\Exp02_FlightControl\e0-PlatformStudy\10.QGC-Param-Tune”文件夹，具体实验操作见文件readme.pdf。
2. 在进行硬件在环仿真和真机实验时，常常需要在QGC地面站中观察飞行状态，并对控制器参数进行实时调整，以使得飞机达到最佳的控制效果。
3. 新建一条自定义的参数总共需要四步。
4. 在slx文件所在目录，新建一个px4_simulink_app_params.c的空白文件，并填写参数信息（包括注释），例如右图所示例程，使用PARAM_DEFINE_FLOAT定义浮点数参数名和默认值，用PARAM_DEFINE_INT32定义整型数据，并在注释中定义参数最小、最大、精度和增长量。在本例中，我们定义了SL_RFLY_FLT和SL_RFLY_INT两个参数。
5. 在MATLAB工作空间中，调用Pixhawk_CSC.Parameter函数来注册参数。这里我们通过init_control.m来在slx打开时自动注册两参数。

```c
/**
 * RflySim QGC Tune Param RFLY_FLT
 *
 * <longer description, can be multi-line>
 *
 * @min 1
 * @max 1000
 * @decimal 2
 * @increment 1
 * @group simulink
 */
PARAM_DEFINE_FLOAT(SL_RFLY_FLT, 100.0);
/**
 * RflySim QGC Tune Param RFLY_INT
 *
 * <longer description, can be multi-line>
 *
 * @min 1
 * @max 1000
 * @decimal 2
 * @increment 1
 * @group simulink
 */
PARAM_DEFINE_INT32(SL_RFLY_INT, 50);
```

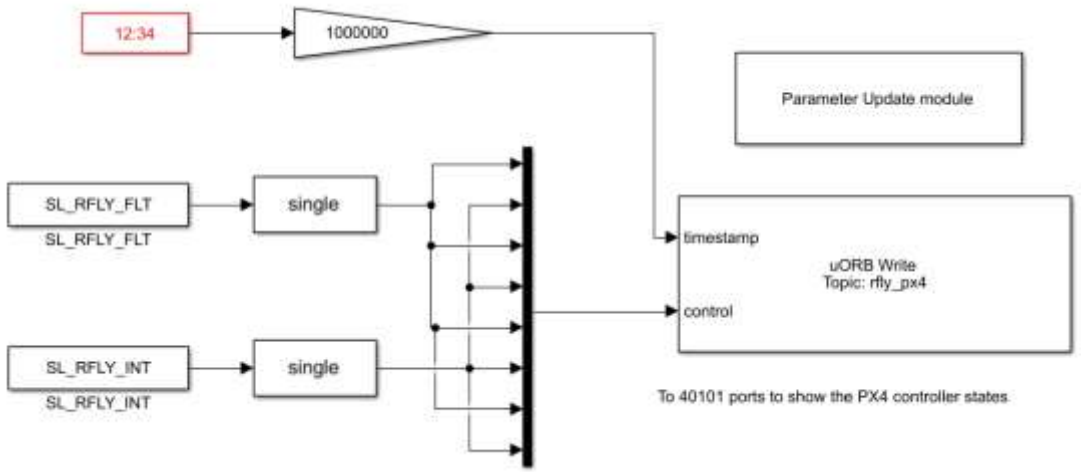
```
SL_RFLY_FLT = Pixhawk_CSC.Parameter({single(100.0), 'SL_RFLY_FLT'});
SL_RFLY_INT = Pixhawk_CSC.Parameter({int32(50), 'SL_RFLY_INT'});
```

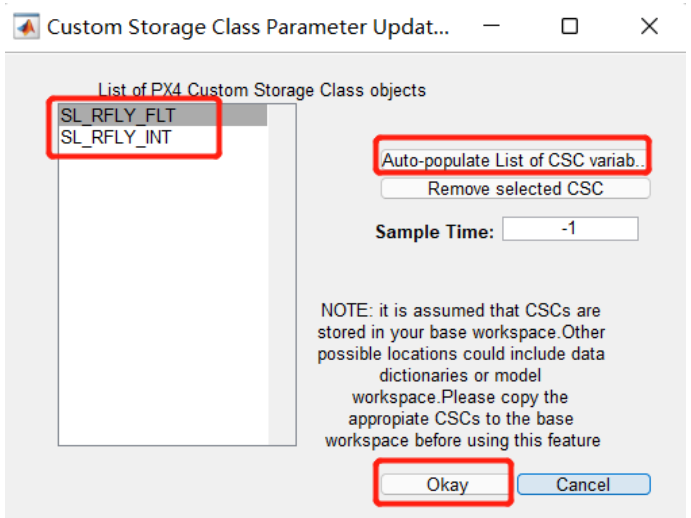
注：参数名只支持全大写+数字形式，不支持小写或大小写混合。

其他接口类实验

QGC实时调整控制器参数

6. 在Simulink中增加了一个“Parameter Update Module”实现定义参数的自动更新，双击本模块，点“Auto-populate List ***”，来确认参数正确导入。注：这里可以打开“PX4QGCTune.slx”来进行测试。
7. 然后就可以在Simulink中直接调用新建的参数，进行控制器的设计，参数可以用于常数模块、积分模块、比例模块等需要配置参数的地方，在使用时需要注意参数类型的转换（浮点数还是整型）。注：在本例中，我们使用SL_RFLY_FLT和SL_RFLY_INT来发送rfly_px4的uORB消息。

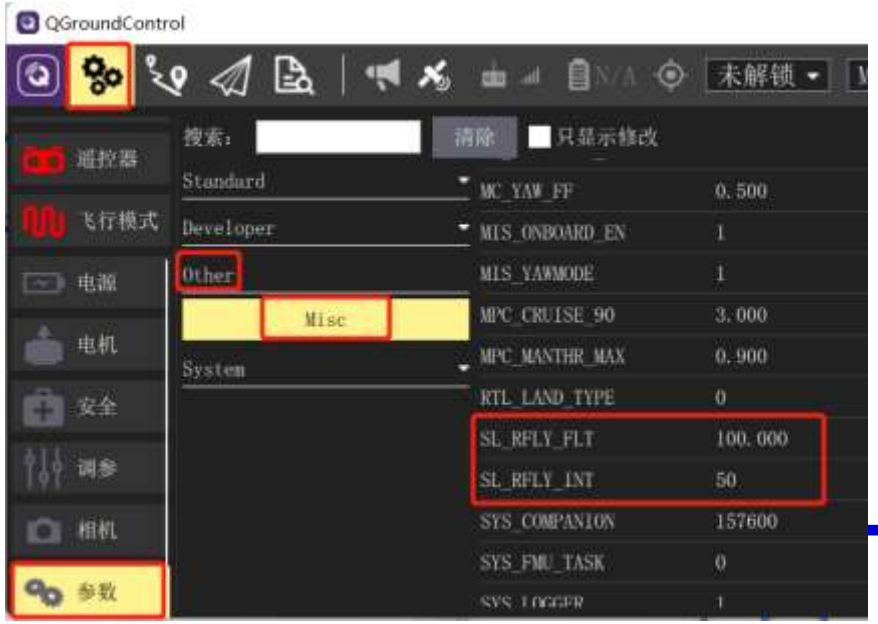




其他接口类实验

QGC实时调整控制器参数

8. 打开PX4QGCTune.slx文件，编译并上传到Pixhawk中。
9. 连接Pixhawk到电脑，并打开QGC地面站，等待与Pixhawk建立连接。
10. 在QGC的设置 - 参数 – Other – Misc 中可以看到刚才新建的两条消息（如果没找到自己的消息，可以用搜索框搜索），然后手动修改两个参数的值。
11. 去QGC的MAVLink Inspector中查看ACTUATOR_CONTROL_TARGET消息（接收来自rfly_px4的数据），查看两个参数的值，是否为刚才修改的值。



小结

(1) 本讲首先总结介绍了课程实验的基本流程，然后以LED灯控制器设计和姿态控制器设计两个实验为例子，详细介绍了平台的使用方法。

(2) LED灯控制实验能让读者熟悉Simulink自动代码生成的基本流程与具体操作方法，同时可以确认整个平台是否运转正常。

(3) 姿态控制器实验展示了后续实验课程的基本流程，包括Simulink控制器设计与仿真、自动代码生成导入Pixhawk自驾仪、硬件在环仿真、实际飞行实验。

(4) 其他小节例程主要展示了飞控底层数据读取、通信等方法，通过学习可了解飞控底层数据的获取途径和通信方式等，为后续实验例程建立基础。

如有疑问，请到 https://doc.rflysim.com 查询更多信息。

谢谢！