
1. 实验名称及目的

1.1. 实验名称

PX4 控制器的外部通信实验

1.2. 实验目的

以外部发送不同的 rfly_ctrl 消息 ID 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。

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。

1.3. 关键知识点

打开[\[安装目录\]\5.RflySimFlyCtrl\0.ApiExps\9.PX4CtrlExternalTune](#) 中的 Init_control.m 文件，该文件为控制器初始化参数文件，定义多旋翼姿态控制和角速率控制的参数，并可能调用了用于替换姿态控制器实现的函数。

代码解析如下：

```
clear
clc
```

角度与弧度的转换

```
%Constant value
RAD2DEG = 57.2957795;
DEG2RAD = 0.0174533;
```

悬停时的油门值

```
%throttle when UAV is hovering
THR_HOVER = 0.609;
```

位置控制器的 PID 参数，包括俯仰角和横滚角的比例、积分和微分参数，以及偏航角速率的 PID 参数

```
%% control parameter
%position PID parameters
Kp_PITCH_ANGLE = 6.5;
Kp_PITCH_AngleRate = 0.1;
Ki_PITCH_AngleRate = 0.02;
Kd_PITCH_AngleRate = 0.001;
Kp_ROLL_ANGLE = 6.5;
Kp_ROLL_AngleRate = 0.1;
Ki_ROLL_AngleRate = 0.02;
Kd_ROLL_AngleRate = 0.001;
Kp_YAW_AngleRate = 0.5;
Ki_YAW_AngleRate = 0.01;
Kd_YAW_AngleRate = 0.00;
```

积分饱和限制，防止积分项过大导致系统不稳定

```
%integral saturation
Saturation_I_RP_Max = 0.3;
Saturation_I_RP_Min = -0.3;
Saturation_I_Y_Max = 0.2;
```

```
Saturation_I_Y_Min = -0.2;
```

油门幅值的范围

```
% throttle amplitude
```

```
MAX_MAN_THR = 0.9;
```

```
MIN_MAN_THR = 0.05;
```

最大控制角度的限制

```
%max control angle,default 35deg
```

```
MAX_CONTROL_ANGLE_ROLL = 35;
```

```
MAX_CONTROL_ANGLE_PITCH = 35;
```

最大控制角速率的限制

```
%max control angle rate,rad/s
```

```
MAX_CONTROL_ANGLE_RATE_PITCH = 220;
```

```
MAX_CONTROL_ANGLE_RATE_ROLL = 220;
```

```
MAX_CONTROL_ANGLE_RATE_Y = 200;
```

2. 实验效果

在 Simulink 直接控制硬件在环仿真中的飞机。

在 Python 程序直接控制硬件在环仿真中的飞机。

3. 文件目录

文件夹/文件名称	说明
Exp12_Msg2SimulinkAPI.bat	硬件在环仿真脚本
Exp12_Msg2SimulinkAPI.py	通信 API (py 版)
Exp12_Msg2SimulinkAPI.slx	通信 API (Simulink 版)
Init_control.m	初始化文件。
PX4ExtMsgReceiver.slx	PX4 外部通信接收端模型文件。
PX4ExtMsgSender.slx	PX4 外部通信发送端模型文件(Simulink 版)。

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 平台免费版及以上版本	Pixhawk 6X 或 Pixhawk 6x mini ^②	1
	MATLAB 2017B 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://doc.rflysim.com>

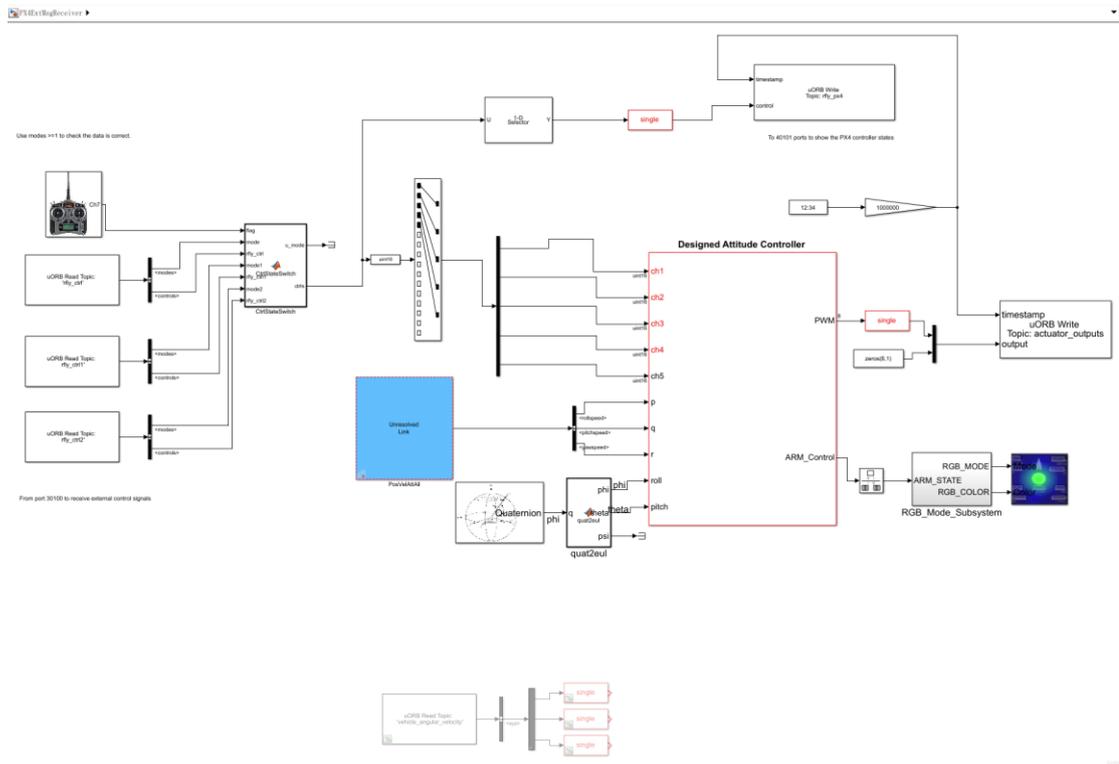
③：本实验演示所使用的遥控器为：天地飞 ET10、配套接收器为：WFLY RF209S。遥

控器相关配置见：<https://rflysim.com/doc/zh/B/3.1ET10.html>

5. 实验步骤

5.1. PX4ExtMsgReceiver.slx—PX4 外部通信接收端模型文件

如下图所示，打开[\[安装目录\]\5.RflySimFlyCtrl\0.ApiExps\9.PX4CtrlExternalTune](#)中的 PX4ExtMsgReceiver.slx 文件。该系统主要由 uORB Read and Function-Call Trigger、uORB Write Advanced、Subsystem、quat2eul、CtrlStateSwitch、input_rc 和 RGB_LED 构成。



uORB Read and Function-Call Trigger、uORB Write Advanced、input_rc 和 RGB_LED 具体用法和功能请参考：[API.pdf](#)

对于 Subsystem 模块的详细介绍请参考[\[安装目录\]\5.RflySimFlyCtrl\0.ApiExps\1.SoftwareSimExps\Intro.pdf](#)。

quat2eul 这个函数实现了从四元数到欧拉角的转换，用于将飞行器的姿态信息从四元数表示转换为更直观的欧拉角表示。

函数解析如下：

定义了一个 MATLAB 函数 quat2eul，接受一个四元数向量 q 作为输入，并返回对应的欧拉角 $[\phi, \theta, \psi]$

```
function [phi,theta,psi] = quat2eul(q)
% Conversion from Quaternion to Euler angles based on the PX4 Firmware
% v1.6.5 release.
%
% Quaternion -> DCM -> Euler Angles
% INPUT:
% q: Quaternion vector of the format: (a + bi + cj + dk)
```

```
% OUTPUT:  
% [phi,theta,psi]: Euler angles in radians.  
%
```

这里将四元数的四个分量分别赋值给变量 a, b, c, d

```
a = q(1);  
b = q(2);  
c = q(3);  
d = q(4);
```

计算了四元数的平方以及一些常量，为后续的计算做准备

```
aSq = a*a;  
bSq = b*b;  
cSq = c*c;  
dSq = d*d;  
  
d_pi2 = pi/2;
```

dcm00, dcm02, dcm10, dcm12, dcm20, dcm21, dcm22 计算了旋转矩阵（方向余弦矩阵）的各个元素

```
% quaternion_to_dcm  
dcm00 = aSq + bSq - cSq - dSq;  
% dcm01 = 2*(b*c - a*d);  
dcm02 = 2*(a*c + b*d);  
dcm10 = 2*(b*c + a*d);  
% dcm11 = aSq - bSq + cSq - dSq;  
dcm12 = 2*(c*d - a*b);  
dcm20 = 2*(b*d - a*c);  
dcm21 = 2*(a*b + c*d);  
dcm22 = aSq - bSq - cSq + dSq;
```

根据旋转矩阵计算了欧拉角 phi, theta, psi。这里使用了不同的情况判断来避免除零错误，并根据情况选择正确的欧拉角计算方式

```
% dcm_to_euler  
theta = asin(-dcm20);  
if abs(theta - d_pi2) < 1.0e-3  
    phi = single(0.0);  
    psi = atan2(dcm12, dcm02);  
elseif abs(theta + d_pi2) < 1.0e-3  
    phi = single(0.0);  
    psi = atan2(-dcm12, -dcm02);  
else  
    phi = atan2(dcm21, dcm22);  
    psi = atan2(dcm10, dcm00);  
end
```

end

CtrlStateSwitch 这个函数用于根据输入的标志 flag 来切换控制模式和相关参数。这个函数的作用是根据输入的标志值选择不同的控制模式和参数，以便于在不同的情况下实现不同的飞行控制策略。

函数解析如下：

函数接受七个输入参数：

flag：表示一个标志，根据其值来选择不同的控制模式。

mode、**rfly_ctrl**：表示默认的控制模式和相关参数。

mode1、**rfly_ctrl1**：表示第一种备选的控制模式和相关参数。

mode2、**rfly_ctrl2**：表示第二种备选的控制模式和相关参数。

函数返回两个输出参数：

u_mode：表示根据 **flag** 选择的控制模式。

ctrls：表示根据 **flag** 选择的相关参数。

```
function [u_mode,ctrls]= CtrlStateSwitch(flag,mode,rfly_ctrl,mode1,rfly_ctrl1,mode2,rfly_
ctrl2)
```

如果 **flag** 小于等于 1200，则选择默认的控制模式和参数。

```
if flag<=1200
    u_mode=mode;
    ctrls=rfly_ctrl;
```

如果 **flag** 大于 1200 且小于 1600，则选择第一种备选的控制模式和参数。

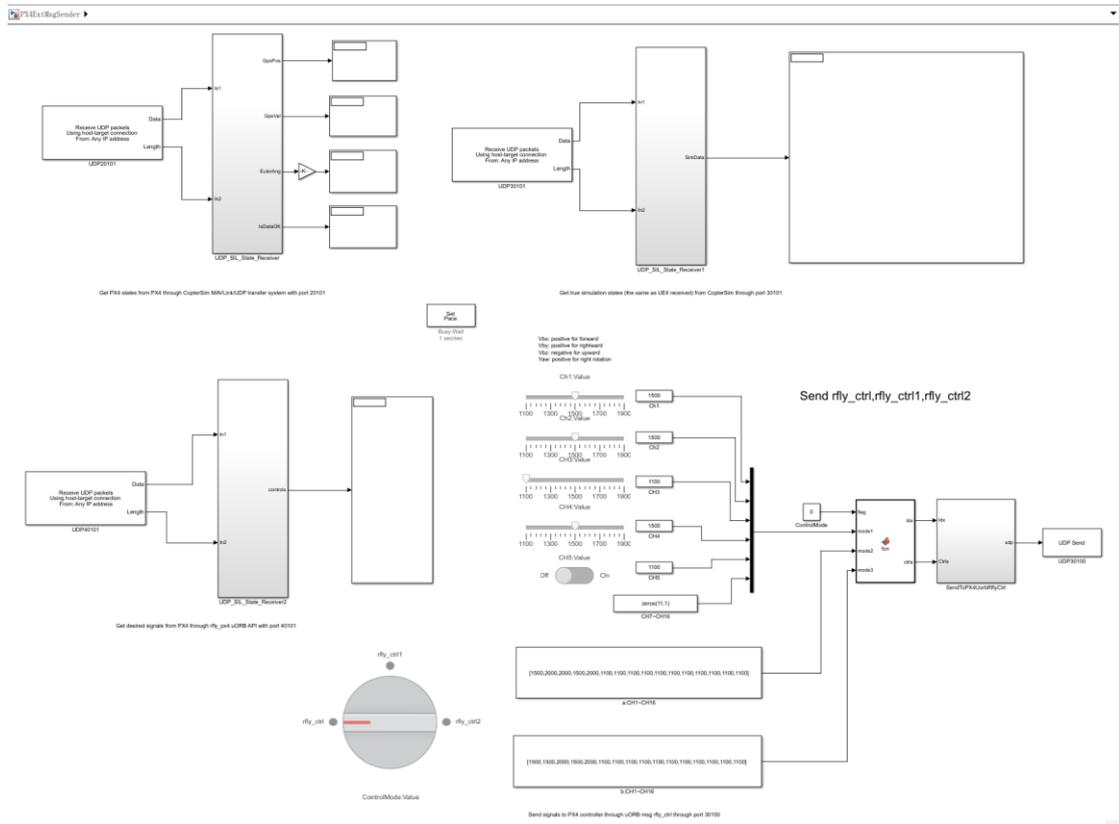
```
elseif flag>1200 && flag<1600
    u_mode=mode1;
    ctrls=rfly_ctrl1;
```

如果 **flag** 大于等于 1600，则选择第二种备选的控制模式和参数

```
else
    u_mode=mode2;
    ctrls=rfly_ctrl2;
end
end
```

5.2. PX4ExtMsgSender.slx—PX4 外部通信发送端模型文件

如下图所示，打开[\[安装目录\]\5.RflySimFlyCtrl\0.ApiExps\9.PX4CtrlExternalTune](#) 中的 P
X4ExtMsgSender.slx 文件。



关于该 Simulink 模型的详细描述请参考:[Readme.pdf](#)

5.3. Exp12_Msg2SimulinkAPI.slx—通信 API（Simulink 模型）

如下图所示，打开[\[安装目录\]\5.RflySimFlyCtrl\0.ApiExps\9.PX4CtrlExternalTune](#)中的 Exp12_Msg2SimulinkAPI.slx 文件。

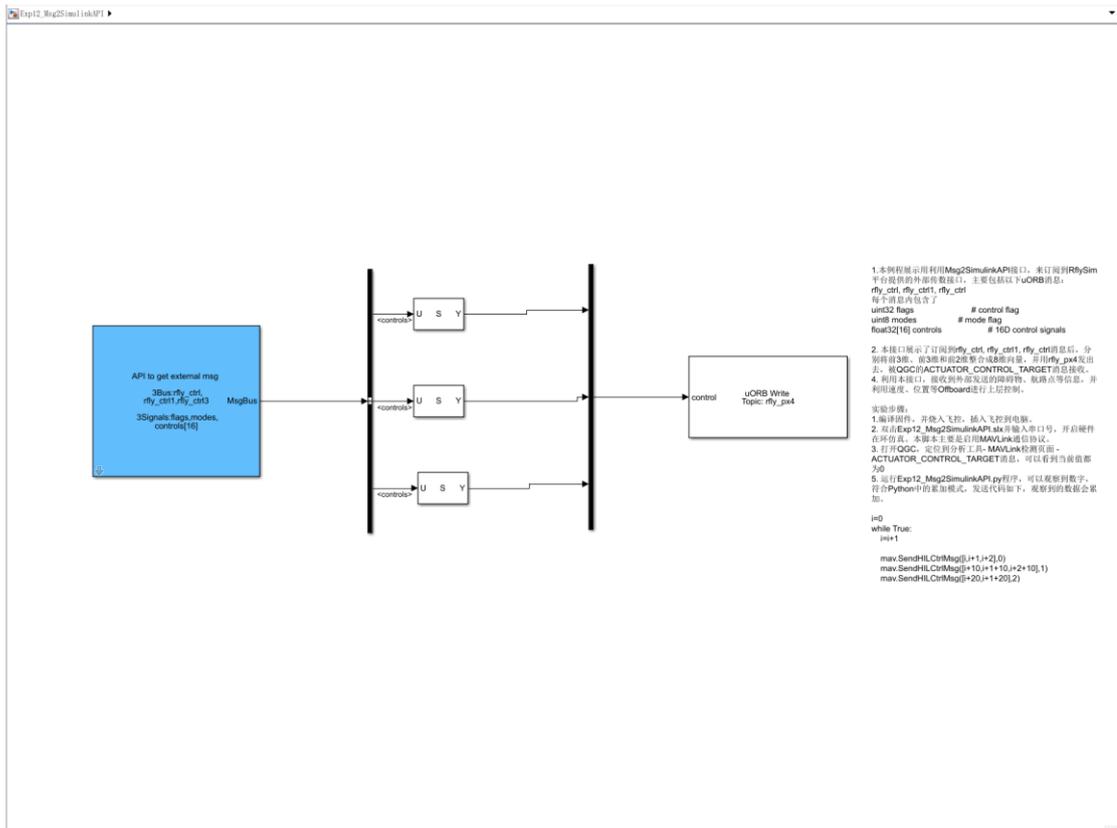
本例程展示利用 Msg2SimulinkAPI 接口，来订阅到 RflySim 平台提供的外部传数接口，主要包括以下 uORB 消息：

rfly_ctrl, rfly_ctrl1, rfly_ctrl

每个消息内包含了：

```
uint32 flags # control flag
uint8 modes # mode flag
float32[16] controls # 16D control signals
```

本接口展示了订阅到 rfly_ctrl, rfly_ctrl1, rfly_ctrl 消息后，分别将前 3 维、前 3 维和前 2 维整合成 8 维向量，并用 rfly_px4 发出去，被 QGC 的 ACTUATOR_CONTROL_TARGET 消息接收。利用本接口，接收到外部发送的障碍物、航路点等信息，并利用速度、位置等 Offboard 进行上层控制。

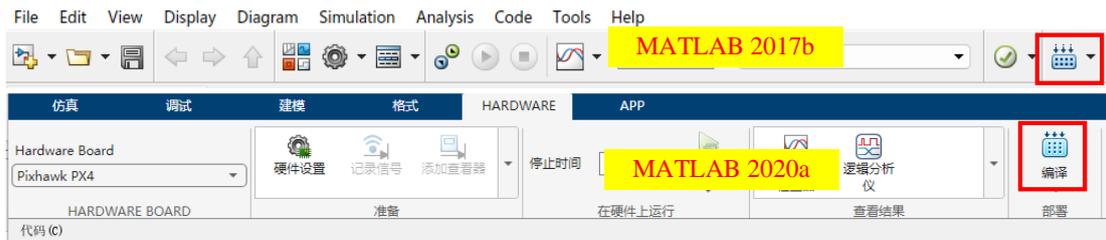


uORB Read and Function-Call Trigger 和 uORB Write Advanced 具体用法和功能请参考：

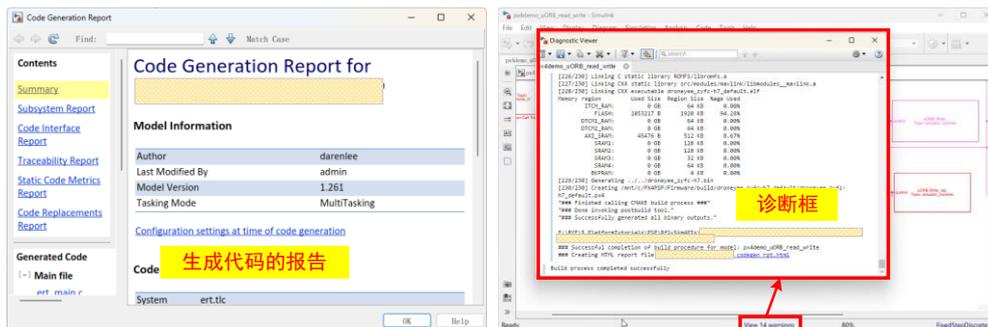
[API.pdf](#)

5.4. Simulink 控制硬件在环仿真

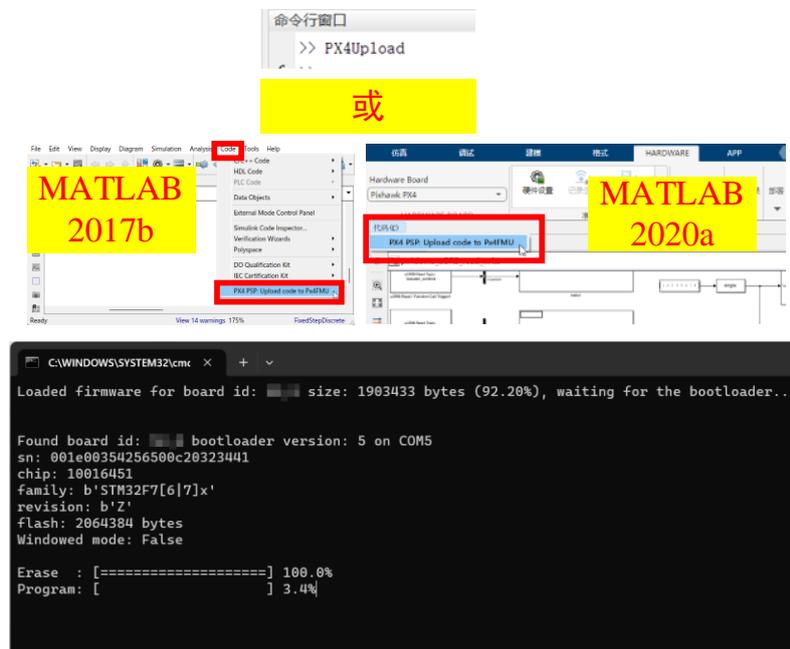
打开 MATLAB 软件，运行 Init_control.m 文件，同时将打开 [PX4ExtMsgReceiver.slx](#) 文件，在 Simulink 中，点击编译命令。



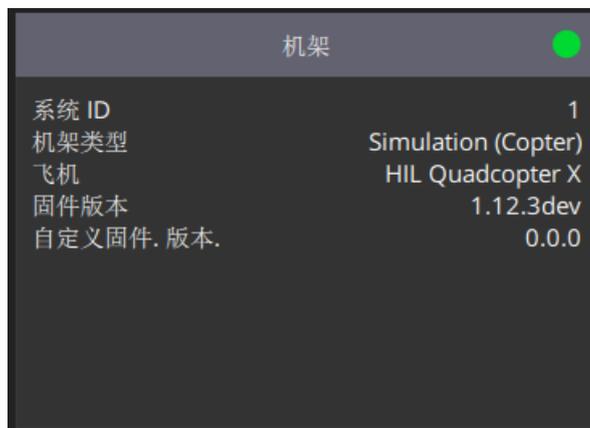
在 Simulink 的下方点击 View diagnostics 指令，即可弹出诊断对话框，可查看编译过程。在诊断框中弹出 Build process completed successfully，即可表示编译成功，左图为生成的编译报告。



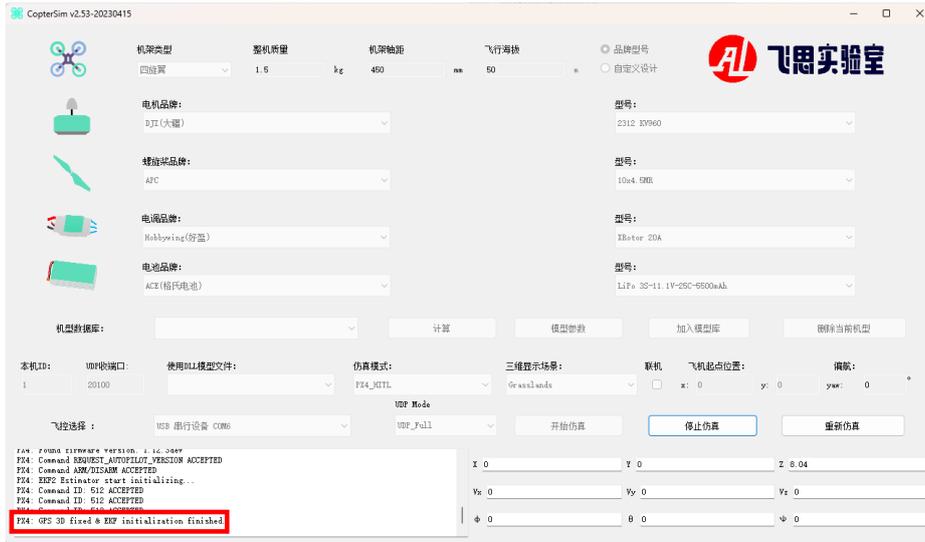
用 USB 数据线链接飞控与电脑。在 MATLAB 命令行窗口输入：PX4Upload 并运行或点击 PX4 PSP: Upload code to Px4FMU，弹出 CMD 对话框，显示正在上传固件至飞控中，等待上传成功。



打开 QGroundControl 软件。确认无人机机架设置如下：

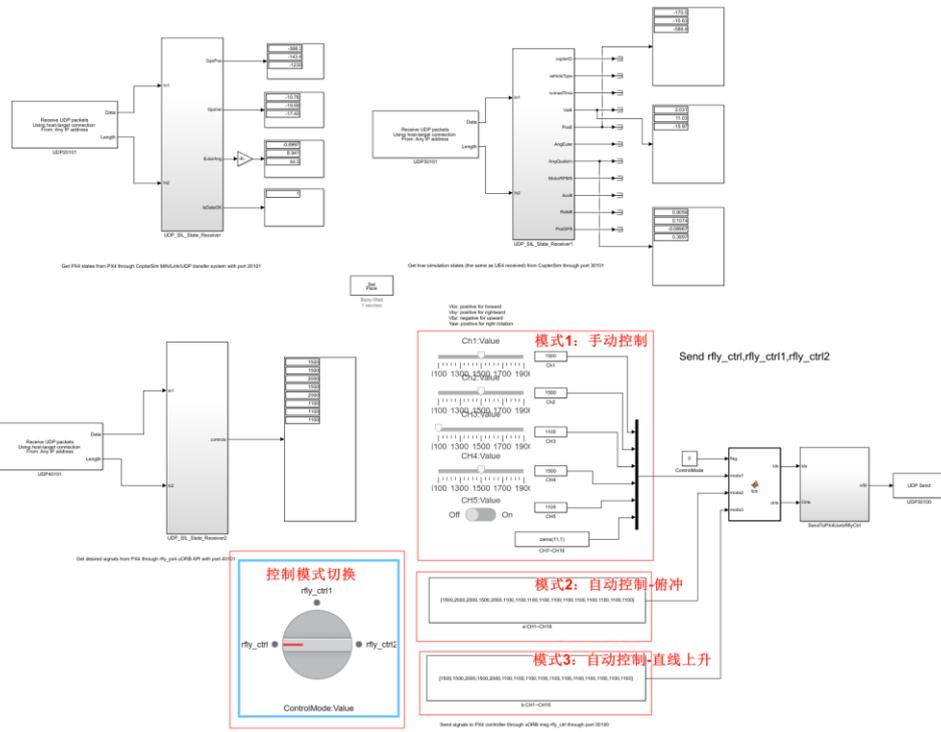


上传成功后，双击打开"*\桌面\RflyTools\HITLRun.lnk"或"*\PX4PSP\RflySimAPIs\HITLRun.bat"文件，在弹出的 CMD 对话框中输入插入的飞控 Com 端口号，即可自动启动 RflySim3D、CopterSim、QGroundControl 软件，等待 CopterSim 的状态框中显示：PX4: GPS 3D fixed & EKF initialization finished。



在 MATLAB 中运行 PX4ExtMsgSender.slx 文件，在运行过程中，可通过 ControlMode 名称的旋转开关来切换，通过 30100 端口发送的数据类型，其中，rfly_ctrl 代表发送 rfly_ctrl 消息 ID 为：rfly_ctrl 的数据，rfly_ctrl 代表发送 rfly_ctrl1 消息 ID 为：rfly_ctrl1 的数据；rfly_ctrl 代表发送 rfly_ctrl2 消息 ID 为：rfly_ctrl2 的数据。本实验三种类型的数据切换步骤如下：

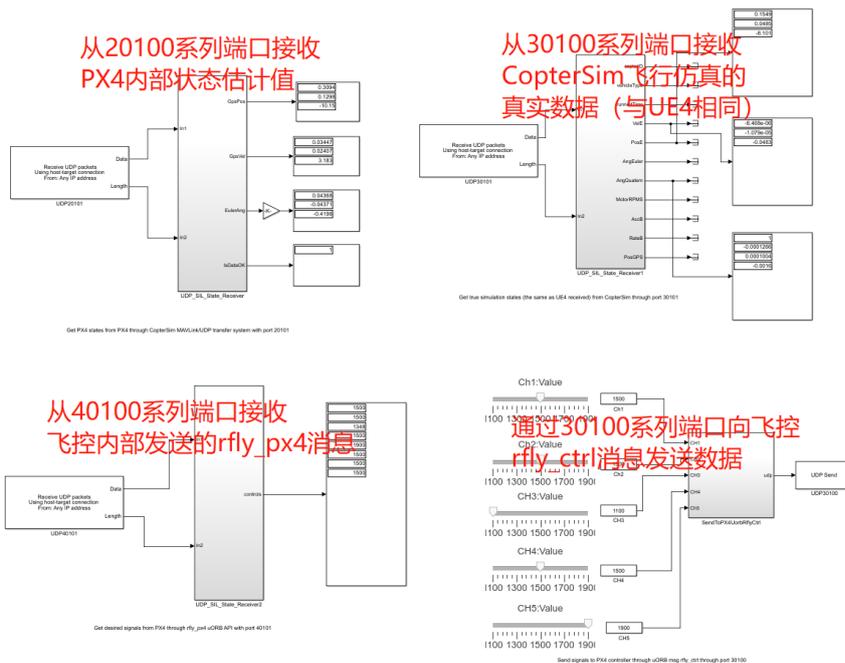
1. 先将 ControlMode 旋转开关切换到：rfly_ctrl，再将遥控器 CH7 通道切换到最低位，进入模式 1：手动控制，可通过滑块和开关控制解锁等操作。
2. 先将 ControlMode 旋转开关切换到：rfly_ctrl1，再将遥控器 CH7 通道切换到中间位，进入模式 2：自动控制俯冲。
3. 先将 ControlMode 旋转开关切换到：rfly_ctrl2，再将遥控器 CH7 通道切换到最高位，进入模式 1：自动控制直线上升。



同时，在上图的左小脚的 UDP40101 中可以看到对应的控制量变化。

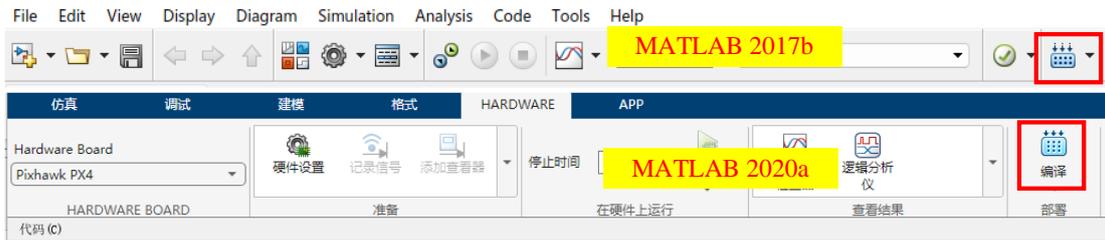


也可看到飞机的一些状态量，具体定义如下：

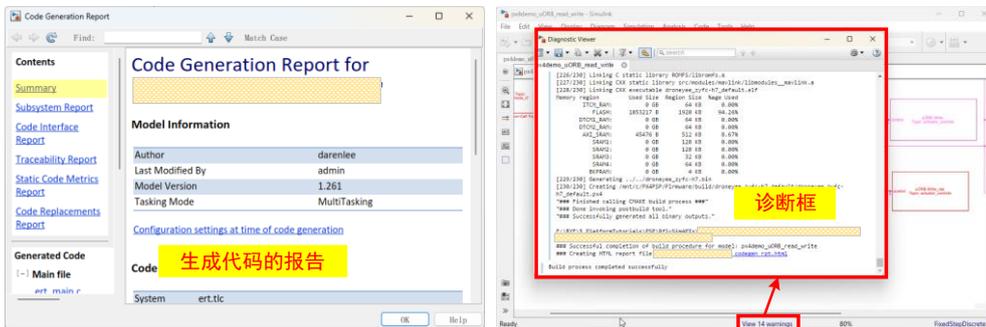


5.5. Python 控制硬件在环仿真

打开 MATLAB 软件，运行 9.PX4CtrlExternalTune 文件夹下的 Init_control.m 文件，同时将打开 PX4ExtMsgReceiver.slx 文件，在 Simulink 中，点击编译命令。

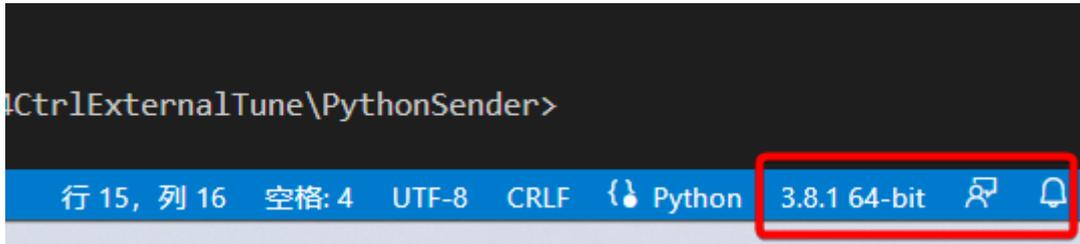


在 Simulink 的下方点击 View diagnostics 指令，即可弹出诊断对话框，可查看编译过程。在诊断框中弹出 Build process completed successfully，即可表示编译成功，左图为生成的编译报告。

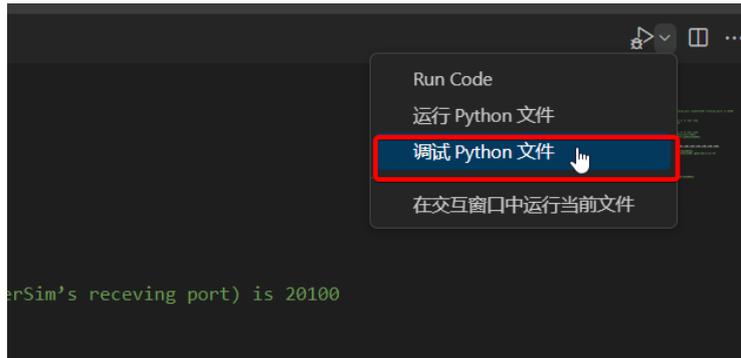


用 USB 数据线链接飞控与电脑。在 MATLAB 命令行窗口输入：PX4Upload 并运行或点击 PX4 PSP: Upload code to Px4FMU，弹出 CMD 对话框，显示正在上传固件至飞控中，等待上传成功。

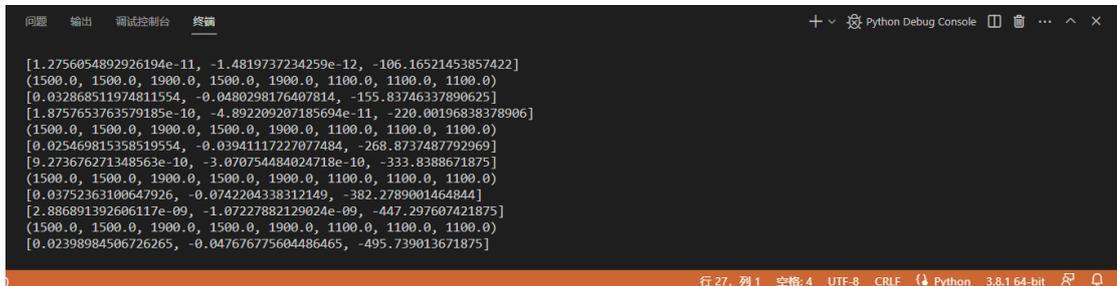
打开 VScode，在 VScode 中文件->打开文件夹，打开问实验文件夹目录，请确认选择的编译器为：`*\PX4PSP\Python38\python.exe`，即 RflySim 平台的 Python38Env 环境。



在 VScode 中打开 PythonSender.py 文件，点击右上角的“调试 Python 文件”按钮。



在 RflySim3D 中可看到飞机起飞，同时在 VScode 的终端框中分别循环实时打印出：分别来自 20100、30100、40100 端口的 PX4 内部状态估计值、CopterSim 飞行仿真真实数据以及飞控内部发送的 `rfly_px4` 消息。



6. 参考资料

[1] 无。

7. 常见问题

Q1: ***

A1: ***