

# 1. 实验名称及目的

## 1.1 实验名称

载具系统建模模板之最大模板使用介绍

## 1.2 实验目的

该例程对如何使用平台最大模板进行软件在环和硬件在环仿真进行介绍。

## 1.3 关键知识点

Exp2\_MaxModelTemp.slx是最大系统的模版，相对于最小系统模板，包含了更丰富的附加功能。附加的输入、输出和参数不需要全部加上，只需要在使用本功能时加入模型即可，但是务必保证名称、维度和数据格式一致。

## 模型参数介绍（参考PX4PSP\RflySimAPIs\4.RflySimModel\API.pdf中DLL/SO模型与通信接口的重要参数部分）

### 重要参数

Exp2\_MaxModelTemp\_init.m 中定义了最大系统模型的各种参数，关键数据如下。

飞机的三维显示样式

```
ModelParam_3DType = int16(3);
```

%这个参数决定了飞机的三维显示样式，需要和RflySim3D的XML文件中的ClassID相匹配

飞机的类型

```
ModelParam_uavType = int16(3); %这个参数决定了飞机力和力矩的计算方式
```

飞机的初始位姿参数

```
ModelInit_PosE=[0,0,0];
```

%用于设置飞机的初始位置，对应了CopterSim上的X和Y初始值。Z值利用TerrainZ实现了从CopterSim中读取当前地形高度数据，使得飞机可以初始化在复杂地形的地表面（例如Grassland地图）。

```
ModelInit_AngEuler=[0,0,0];
```

%用于设定飞机的初始姿态。飞机姿态角的前两位（俯仰和滚转角）可以通过ModelInit\_AngEuler参数来配置，但是偏航角需要在CopterSim中配置。针对导弹等垂直起飞的飞行器，需要设定合适的俯仰和滚转值。

QGC中显示的地图坐标和高度原点（在RflySim3D的Cesium大场景中能任意指定飞机在地球三维场景中的坐标）

```
ModelParam_GPSLatLong = [40.1540302 116.2593683];%飞机初始的纬度和精度，单位度。
```

```
ModelParam_envAltitude =
```

```
-50;%原点的海拔高度，竖直向下为正，高于海平面填负值，单位米。
```

执行器的初始参数

```
ModelInit_Inputs = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];%
```

十六维输入向量，定义电机PWM初始值，默认全0，对固定翼和小车需要修改，因为它们的油门在初始状态处于最小值（-1），见“Motor

Model” 模块

故障接口参数

FaultInParams：可通过外部消息动态改变的32维参数向量，在故障注入或者可变形的异构飞行器上可用，也可动态地调整传感器模型噪声等；与inSILInts和inSILFloats 形成功能互补。

```
FaultParamAPI.FaultInParams = zeros(32,1);% 定义了一个名为 FaultInParams 的32维向量，该向量被初始化为所有元素都为零。
```

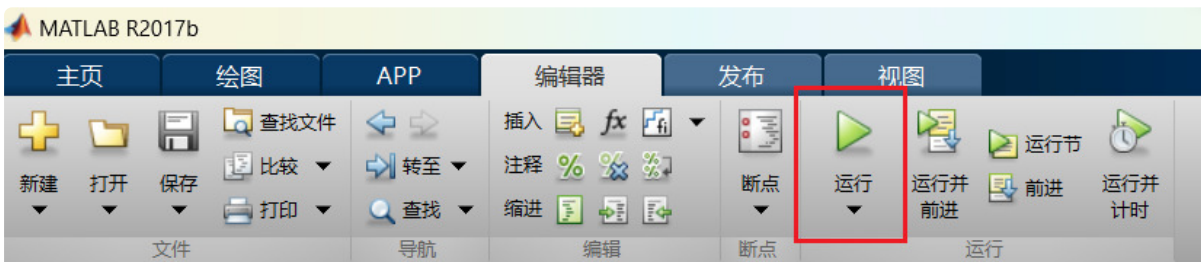
```
FaultParamAPI.FaultInParams(3)=1; %将 FaultInParams 向量的第三个元素设置为1。
```

## 参数调用过程

Exp2\_MaxModelTemp.slx是DLL模型生成的完整模板，模型启动运行（编译）时会调用[Exp2\\_MaxModelTemp\\_init.m](#)



Exp2\_MaxModelTemp\_init.m 中包含了模型的参数信息，本脚本会在 Exp2\_MaxModelTemp.slx 编译（编译所需环境配置参考 [PX4PSP\RflySimAPIs\4.RflySimModel\API.pdf](#) 中的环境配置）时被调用将参数载入 MATLAB 工作空间，也可以直接运行脚本文件将参数载入工作空间。Simulink 模型会通过参数名称读取工作空间中的参数，故需要保证 simulink 模型中设置的参数名称与 `***_init.m` 中的参数名称相同。



名称 ▲	值
ext	'slx'
FaultParamAPI	1x1 struct
filepath	'E:\d2\4.RflySimMo...
HILGPS	1x1 Bus
InitFileName	'Exp2_MaxModelTe...
MavLinkGPS	1x1 Bus
MavLinkSensor	1x1 Bus
MavLinkStateQuat	1x1 Bus
MavVehileInfo	1x1 Bus
ModelInit_AngE...	[0,0,0]
ModelInit_Inputs	1x16 double
ModelInit_PosE	[0,0,0]
ModelInit_RateB	[0,0,0]
ModelInit_RPM	0
ModelInit_VelB	[0,0,0]
ModelParam_en...	-50
ModelParam_GP...	[40.1, 1x1 double]
ModelParam_m...	842.1000
ModelParam_m...	1.2870e-04
ModelParam_m...	0.0500
ModelParam_m...	0.0214
ModelParam_m...	22.8300
ModelParam_rot...	2.7830e-07
ModelParam_rot...	1.6810e-05
ModelParam_ua...	[0.0035,0.0039,0.00...
ModelParam_ua...	0.0550
ModelParam_ua...	0.1200
ModelParam_uavJ	[0.0211,0,0;0,0.0219...
ModelParam_ua...	1.5150
ModelParam_ua...	4
ModelParam_ua...	0.2250
ModelParam ua...	3
name	'Exp2_MaxModelTe...

GenerateModelDLLFile.p是将slx模型转化为DLL模型文件的脚本，使用RflySim平台进行载具软硬件在环仿真时，需要将DLL(windows下)/SO（Linux下）模型导入到CopterSim，形成运动仿真模型，因此，在Simulink模型编译完成后，需要将模型对应的C++文件打包成DLL/SO模型。

## 输入信号（参考

## PX4PSP\RflySimAPIs\4.RflySimModel\API.pdf中DLL/SO模型与通信接口的数据协议部分）

最大模板的12个输入数据包括电机控制量、地形数据、碰撞数据等。

## 电机数据inPWMs

输入接口inPWMs，16维执行器控制量输入，已归一化到-1到1尺度(通常电机是0-1，舵机是-1~1)，它的数据来自飞控回传的电机控制MAVLink消息mavlink\_hil\_actuator\_controls\_t的controls，具体定义如下：

```
typedef struct __mavlink_hil_actuator_controls_t {  
  
    uint64_t time_usec; //时间戳，从开机后的时间，单位ms  
  
    uint64_t flags; //标志位，用于显示当前的飞行状态  
  
    float controls[16]; //控制量，16维电机的控制量，发送到模型中，驱动飞机飞行  
  
    uint8_t mode; //模型，用于显示飞机当前的飞行模式和是否上锁等信息}  
mavlink_hil_actuator_controls_t;
```

软件在环仿真时，电机控制指令从PX4 SITL控制器通过TCP 4561系列端口以MAVLink协议发送到运动仿真模型的inPWMs接口，而硬件在环仿真时，该指令是从飞控通过串口以MAVLink协议发送到运动仿真模型的inPWMs接口。

## 地形高度terrainZ

最大模板利用TerrainZ

实现了从CopterSim中读取当前地形高度数据，使得飞机可以初始化在复杂地形的地表面(例如RflySim3D中的Grassland地图)。

## inSILInts

8维Int32型输入，通过UDP协议获取，来自30100++2系列端口号，软硬件在环仿真时，可通过该端口向模型输入一些量；同时，该接口是实现综合模型的关键接口。

## inSILFloats

20维float型输入，通过UDP协议获取，来自30100++2系列端口号，软硬件在环仿真时，可通过该端口向模型输入一些量；同时，该接口是实现综合模型的关键接口。

## 碰撞数据inFloatsCollision

利用inFloatsCollision实现了一个简单地物理引擎，可以根据RflySim3D回传的四周距离数据，实现碰到障碍物的回弹、碰到其他飞机便坠毁等功能

## 飞控状态量 (uORB数据) 输入inCopterData

32维，其中后8维接收PX4消息，数据来自uORB msg rfly\_px4.control[0:7]。

## inFromUE

6维double型数据，来自三维引擎（Rflysim3D/RflySimUE5），可用于实现地面交互、碰撞引擎等需要与三维引擎进行数据交互的相关功能。

## inCtrlExt1~5

和inSILInts和inSILFloats类似，用于接收局域网内的故障注入消息

## 模型模块（参考PX4PSP\RflySimAPIs\4.RflySimModel\API.pdf中的Simulink载具建模模板介绍）

推荐不改变本模型模版，而是在其中修改模型参数（质量、转动惯量等）和电机模块（转速动态响应）与力和力矩模块（驱动力、气动力、地面支撑和阻力等）两个自定义模块，来适配不同的载具的模型。

## Motor Model电机模块

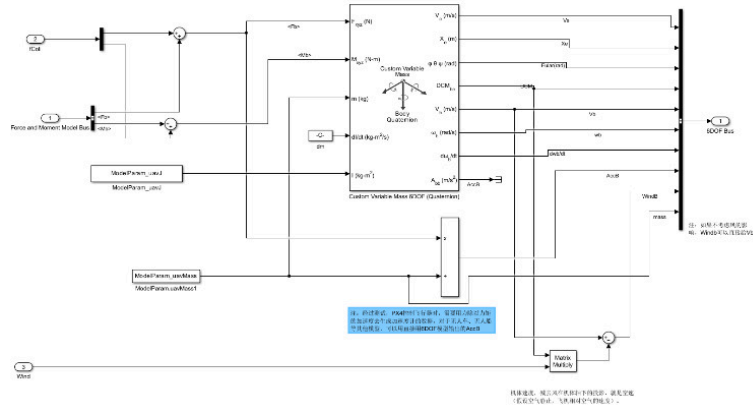
在该模块中输入为PWM值（通过inPWMs接口获取），经过各电机的非线性动力学模型后得到各电机转速，该模块的输出分别为输入给力和力矩模型的电机转速（弧度每秒）；输入给UE的电机转速（转每分）

## Force and Moment Model力和力矩模块

该模块输入为电机转速MotorRads、飞机运动学姿态6DOF和地形高度输入TerrainZ，输出为多旋翼合力、合力矩Force and Moment Model Bus。

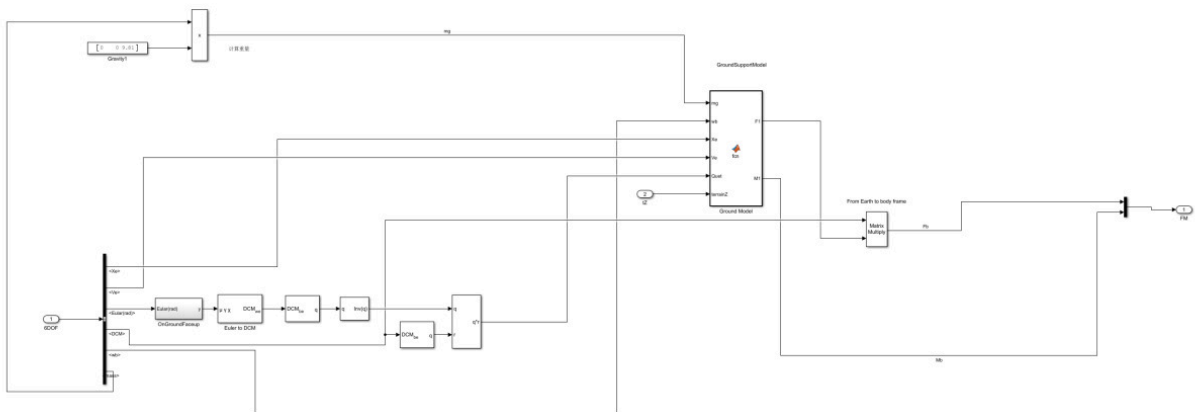
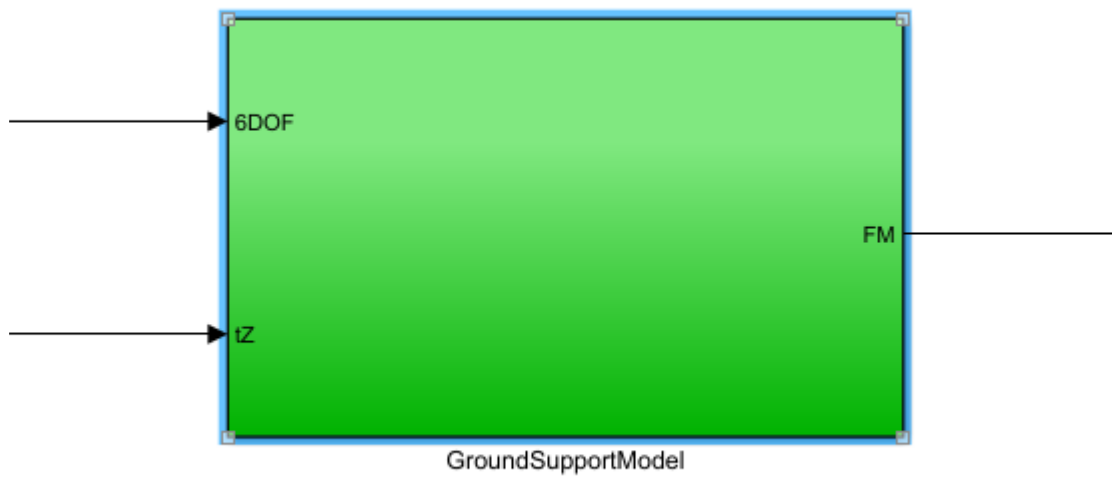
## 6DOF六自由度刚体运动学模块

用于描述无人机在空中运动时的姿态和位置变化。考虑了无人机在三个坐标轴上的旋转运动（俯仰、横滚和偏航）以及机体与地球坐标系上的平移运动（前后、左右和上下）。还可以根据实际需求对模型进行扩展，考虑更多的因素，如飞行器的非线性特性、气动力和惯性矩等。



## GroundSupportModel地面支撑模块

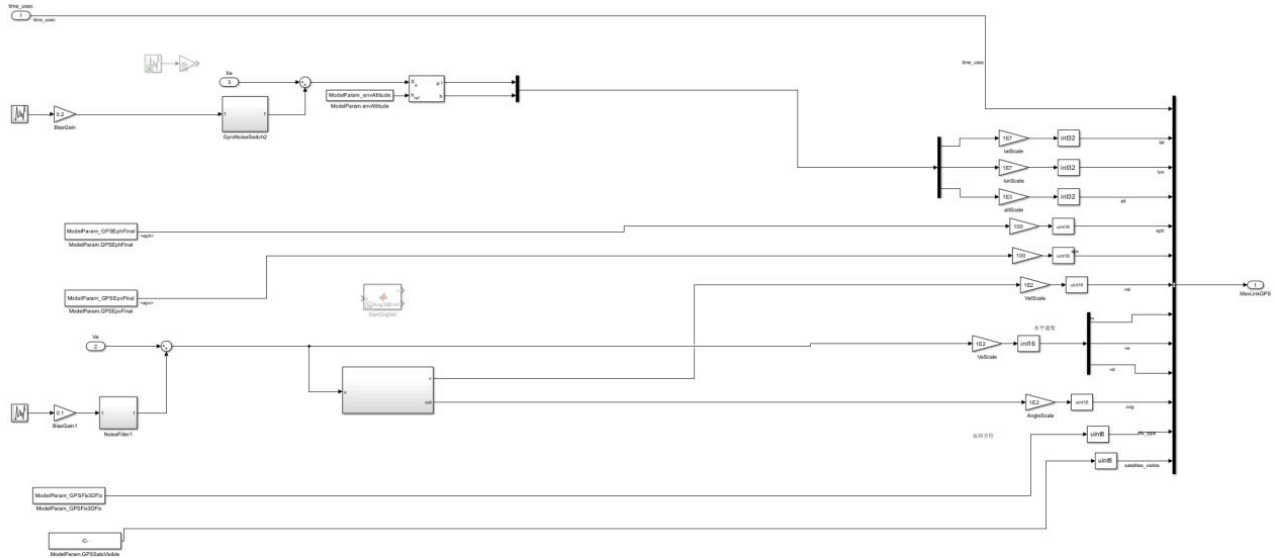
GroundSupportModel地面支撑模块实际上是 **PhysicalCollisionModel碰撞检测模块** 的一个子模块，这里将所有物体简化为较为简单的基本几何体（例如圆柱体或者长方体）来计算其与地面之间的物理接触受力。



## SensorOutput传感器输出模块

该模块中包括了传感器模型和GPS模型。





### 3DOutput三维显示模块

该模块会将\*\*\*\_init.m中的ModelParam\_uavType（三维显示ID）、来自电机模型的ActuatorToUE以及来自6DOF模型的6DOF Bus的位置、速度、姿态和加速度等输出为MavVehile3DInfo，并按协议对输入信息进行数据打包后通过该接口将数据发送至三维引擎

### 输出信号（参考

[PX4PSP\RflySimAPIs\4.RflySimModel\API.pdf](#)中DLL/SO模型与通信接口的数据协议部分)

最大模型模版包含了6个输出信号，分别是HILSensor30d、HILGPS30d、VehileInfo60d、outCopterData、ExtToUE4、ExtToPX4。

### HILSensor30d（传感器接口集合）

模型发送给飞控的各种传感器数据的集合，对应了MAVLink的mavlink\_hil\_sensor\_t消息，本结构体包含了，加速度传感器的加速度值、陀螺仪传感器的角速度值、磁罗盘传感器的磁场值，气压和空速传感器的气压值等。这些传感器的值在仿真时由我们的模型提供，在真机飞行时由真实传感器芯片提供。

```

typedef struct __mavlink_hil_sensor_t {
    uint64_t time_usec; /*时间戳，单位毫秒ms*/
    float xacc; /*机体坐标系x方向加速度，单位m/s^2 */
    float yacc; /*机体坐标系y方向加速度，单位m/s^2 */
    float zacc; /*机体坐标系z方向加速度，单位m/s^2 */
    float xgyro; /*机体坐标系x方向角速度，单位rad/s */
    float ygyro; /*机体坐标系y方向角速度，单位rad/s */
    float zgyro; /*机体坐标系z方向角速度，单位rad/s */
    float xmag; /*机体坐标系x方向磁通量，单位Gauss =T/10000*/
    float ymag; /*机体坐标系y方向磁通量，单位Gauss =T/10000*/
    float zmag; /*机体坐标系z方向磁通量，单位Gauss =T/10000*/
    float abs_pressure; /*绝对气压值，单位 millibar=100Pa*/
    float diff_pressure; /*相气压值，单位 millibar=100Pa*/
    float pressure_alt; /*气压解算高度值，单位m*/
    float temperature; /*温度，单位摄氏度*/
    uint32_t fields_updated; /*传感器参数初始化标志位， bit 0 = xacc, bit 12:
    temperature, bit 31:全部重新初始化 */
} mavlink_hil_sensor_t;

```

## ■ HILGPS30d (GPS接口)

模型发送给飞控的GPS数据值，它对应了MAVLink消息的mavlink\_hil\_gps\_t结构体。输出信号中包含了经纬高、水平竖直精度、地速、北东地的速度、偏航角、定位状态和卫星数量等数据。

这些传感器的值在仿真时由我们的模型提供，在真机飞行时由真实GPS模块提供。

```

typedef struct __mavlink_hil_gps_t {
    uint64_t time_usec; /*时间戳，单位毫秒ms*/
    int32_t lat; /*纬度(WGS84地球模型), 单位度，再乘以 1E7*/

```

```

int32_t lon; /*经度(WGS84地球模型), 单位度, 再乘以 1E7*/

int32_t alt; /*高度 (AMSL地球模型, 而不是 WGS84), 单位m, 再乘以1000
(向上为正)*/

uint16_t eph; /*GPS水平方向定位精度, 单位cm, 如果不知道设为 65535*/

uint16_t epv; /*GPS竖直方向定位精度, 单位cm, 如果不知道设为 65535*/

uint16_t vel; /*GPS地速, 单位cm/s, 如果不知道设为 65535*/

int16_t vn; /*GPS地速朝北方向分量, 单位cm/s */

int16_t ve; /*GPS地速朝东方向分量, 单位cm/s */

int16_t vd; /*GPS地速朝下方向分量, 单位cm/s */

uint16_t cog; /*运动方向, 单位和范围0~359.99度, 再乘以100 degrees *
100, 如果不知道设为 65535*/

uint8_t fix_type; /*定位类型 0-1: no fix, 2: 2D fix, 3: 3D fix. */

uint8_t satellites_visible; /*可见卫星数, 如果不知道设为255*/

}) mavlink_hil_gps_t;

```

注：GPS数据的发送频率与真实传感器硬件基本相同为10Hz，因此飞控的实时位置并不能靠GPS直接提供，需要与IMU等传感器进行融合滤波估计得到。

## ■ VehileInfo60d（真实仿真数据输出）

模型发送给RflySim3D的真实仿真数据，是平滑的理想值，这些数据可用于Simulink下的飞控与模型进行软件仿真测试。

```

struct SOut2Simulator {

int copterID; //飞机ID, 用于区分局域网内不同飞机

int vehicleType;
//飞机样式, 区分同种飞机（如四旋翼）下的不同样式（例如, 大疆、AR.Drone）

double runnedTime; //时间戳, 当前时刻的时间, 单位毫秒

float VelE[3]; //速度向量, 地球坐标系的xyz速度（z向下为正）, 单位m/s

```

```
float PosE[3];  
//位置向量，地球坐标系下的xyz方向（z向下为正，单位m，以起飞点为坐标原点  
  
float AngEuler[3]; //姿态角，飞机的欧拉角，定义于机体坐标系，单位弧度  
  
float AngQuatern[4]; //四元数，飞机姿态的四元数，定义于机体坐标系  
  
float MotorRPMS[8]; //电机转速，飞机的各个旋翼转速，单位转每分  
  
float AccB[3]; //加速度，飞机的运动加速度，单位m/s^2  
  
float RateB[3]; //角速度，飞机的转动角速度，单位rad/s  
  
double PosGPS[3]; //GPS坐标，飞机的经纬高坐标，单位度、度、米  
  
};
```

## ■ OutCopterData（自定义日志输出）

32维double型，里面的内容可自定义发送数据。发往本接口的数据，一方面会写入到本地的log日志中（在C:\PX4PSP\CopterSim下新建CopterSim\*.csv，才会开始记录\*号飞机的数据，注意这里\*要换成飞机的ID）。另一方面，本数据会通过UDP传输到30101系列端口

## ■ ExtToUE4（自定义显示数据输出）

16维double型数据，通过20100系列端口发送给RflySim3D作为第9-24维执行器控制消息显示

## ■ ExtToPX4（自定义uORB数据输出）

16维float型数据，以串口的方式发送给PX4的uORB消息rfly\_ext，用于传输其他传感器或必要数据给飞控

# 2. 实验效果

将平台最大模板生成dll文件后，可进行软硬件在换仿真。

# 3. 文件目录

例程目录：[\[安装目录\]](#)\RflySimAPIs\4.RflySimModel\2.AdvExps\e1\_MaxModelTemp

文件夹/文件名称	说明
Exp2_MaxModelTemp.slx	最大模板源程序文件。
MavLinkStruct.mat	包含数据结构体。
<a href="#">Exp2_MaxModelTempSITL.bat</a>	软件在环仿真批处理文件。
<a href="#">Exp2_MaxModelTempHITL.bat</a>	硬件在环仿真批处理文件。
GenerateModelDLLFile.p	DLL格式转化文件。
<a href="#">Exp2_MaxModelTemp_init.m</a>	最大模型参数初始化文件。

## 4. 运行环境

### 4.1 软件要求

Windows 10及以上版本；RflySim工具链；MATLAB 2017B及以上<sup>③</sup>。

①：若使用Pixhawk 6X飞控，平台安装时的编译命令为：px4\_fmu-v6x\_default，推荐PX4固件版本为：1.12.3。其他配套飞控及编译命令请见：

<https://rflysim.com/doc/zh/1/Hardware.html>

### 4.2 硬件要求

笔记本/台式电脑<sup>①</sup> 1台；Pixhawk 6X或其它飞控<sup>②</sup> 1台；数据线 1台。

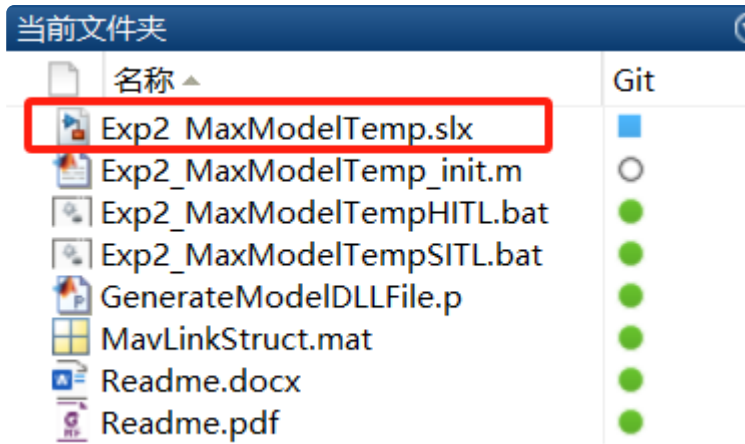
①：推荐配置请见：<https://rflysim.com/>

## 5. 实验步骤

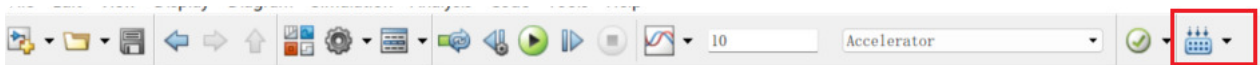
### 5.1. 必做实验：DLL模型生成

#### Step 1: 编译模型

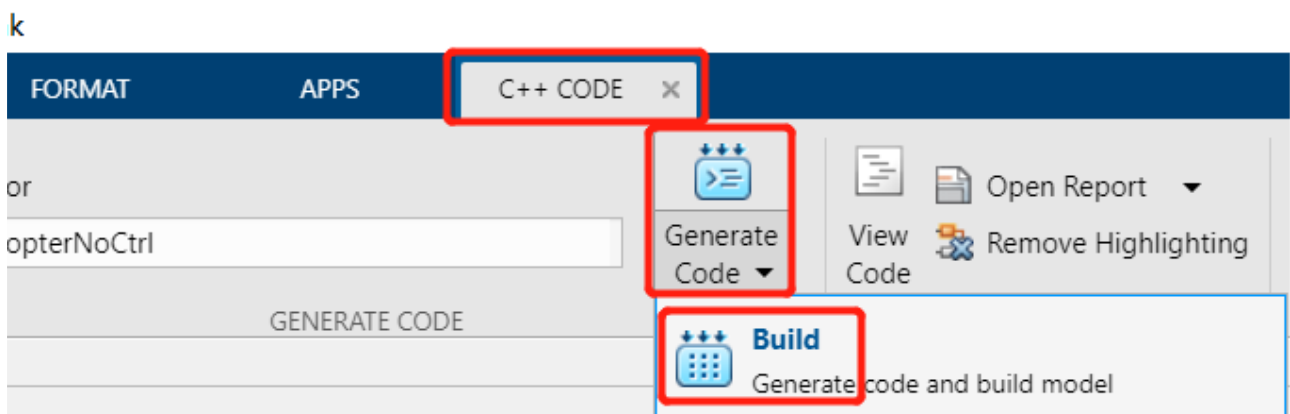
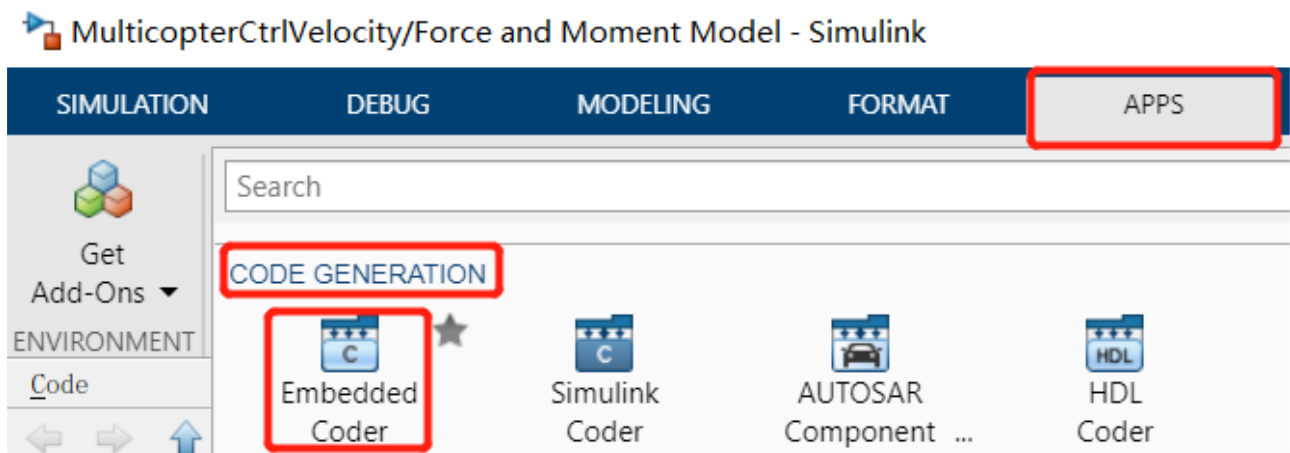
打开“Exp2\_MaxModelTemp.slx”文件，点击编译（“Build Model”）按钮。



对于MATLAB 2019a及之前版本，工具栏样式见下图，直接点击它的编译按钮“Build”即可。



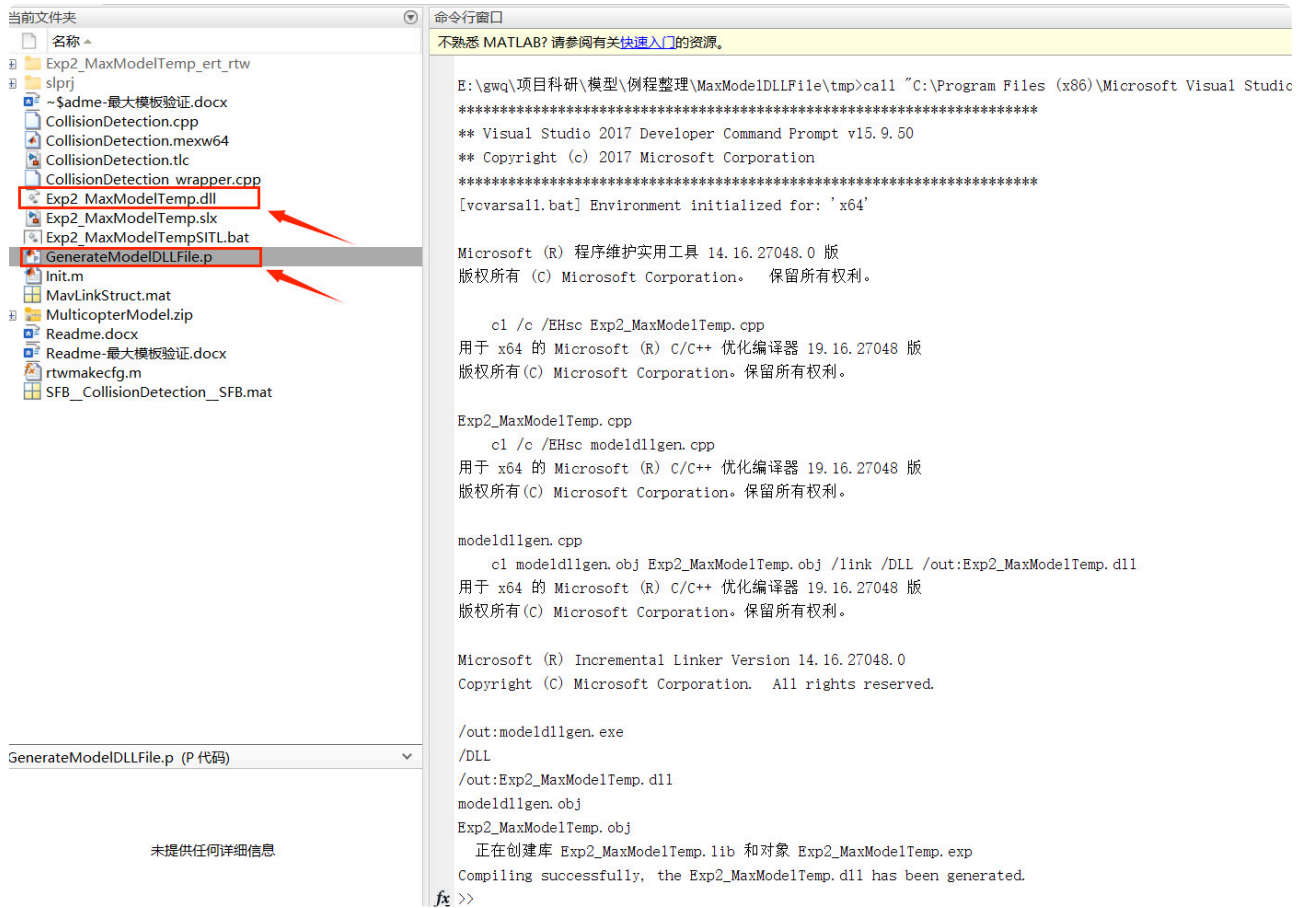
对于2019b及之后版本，点击APPS - CODE GENERATION - Embedded Coder才能弹出代码生成工具栏，在其中如下图所示点击“C++CODE” - “Generate Code” - “Build”按钮就能编译生成代码。



## Step 2: 生成DLL文件

编译完成后，会自动生成多个相关文件，右键GenerateModelDLLFile.p并点击运行，就可以生成

“Exp2\_MaxModelTemp.dll”的DLL模型文件。



```
命令窗口
不熟悉 MATLAB? 请参阅有关快速入门的资源。

E:\gwq\项目科研\模型\例程整理\MaxModelDLLFile\tmp>call "C:\Program Files (x86)\Microsoft Visual Studio
*****
** Visual Studio 2017 Developer Command Prompt v15.9.50
** Copyright (c) 2017 Microsoft Corporation
*****
[vcvarsall.bat] Environment initialized for: 'x64'

Microsoft (R) 程序维护实用工具 14.16.27048.0 版
版权所有 (C) Microsoft Corporation。保留所有权利。

    cl /c /EHsc Exp2_MaxModelTemp.cpp
用于 x64 的 Microsoft (R) C/C++ 优化编译器 19.16.27048 版
版权所有 (C) Microsoft Corporation。保留所有权利。

Exp2_MaxModelTemp.cpp
    cl /c /EHsc modeldllgen.cpp
用于 x64 的 Microsoft (R) C/C++ 优化编译器 19.16.27048 版
版权所有 (C) Microsoft Corporation。保留所有权利。

modeldllgen.cpp
    cl modeldllgen.obj Exp2_MaxModelTemp.obj /link /DLL /out:Exp2_MaxModelTemp.d11
用于 x64 的 Microsoft (R) C/C++ 优化编译器 19.16.27048 版
版权所有 (C) Microsoft Corporation。保留所有权利。

Microsoft (R) Incremental Linker Version 14.16.27048.0
Copyright (C) Microsoft Corporation. All rights reserved.

/out:modeldllgen.exe
/DLL
/out:Exp2_MaxModelTemp.d11
modeldllgen.obj
Exp2_MaxModelTemp.obj
正在创建库 Exp2_MaxModelTemp.lib 和对象 Exp2_MaxModelTemp.exp
Compiling successfully, the Exp2_MaxModelTemp.d11 has been generated.

fx >>
```

注意事项：

与最小模型Exp1\_MinModelTemp相比,最大模型新增了如下输入、输出接口，可实现更多功能。









1. inSILInts:一个8维的自定义输入端口。
2. inSILFloats:一个20维的自定义输入端口。
3. inFloatsCollision: 一个为碰撞模型预留的20维端输入口，信号从UE4传输。
4. inCopterData:一个信号由CopterSim传输的32维输入端口。
5. inFromUE:UE发送的32维输入端口，用于处理模型和场景的交互。
6. inCtrlExt1~5:  
和inSILInts和inSILFloats类似，用于接收局域网内的故障注入消息。
7. outCopterData: 向CopterSim输出外部信号的32维输出端口，定义如下。数据将存储在日志文件CopterSim\*.csv中(应该创建一个文件来存储数据)，其中\*是CopterSim的CopterID。
8. ExtToUE4:一个发送给UE4的16维输出端口。

9. ExtToPX4:一个发送给PX4的16维输出端口。

## 5.2. 必做实验：软件在环仿真

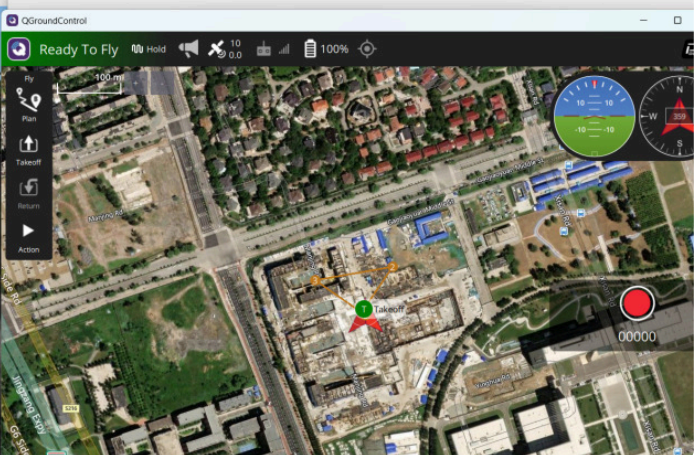
### Step 1: 启动仿真

运行“Exp2\_MaxModelTempSITL.bat”脚本，可以自动打开软件在环仿真，并完成所有需要配置。

 Exp2_MaxModelTemp.slx	2023/10/23 16:30	Simulink Model	72 KB
 Exp2_MaxModelTempHITL.bat	2023/10/17 15:11	Windows 批处理...	6 KB
 Exp2_MaxModelTempSITL.bat	2023/10/17 15:11	Windows 批处理...	6 KB
 GenerateModelDLLFile.p	2023/10/17 15:11	MATLAB.p.9.14.0	6 KB
 MavLinkStruct.mat	2023/10/17 15:11	MATLAB Data	5 KB
 Readme.docx	2023/10/17 15:11	Microsoft Word ...	8,760 KB
 Readme.pdf	2023/10/17 15:11	Foxit PhantomP...	1,109 KB
 Exp2_MaxModelTemp_init.m	2023/10/23 14:15	Objective C 源文件	3 KB

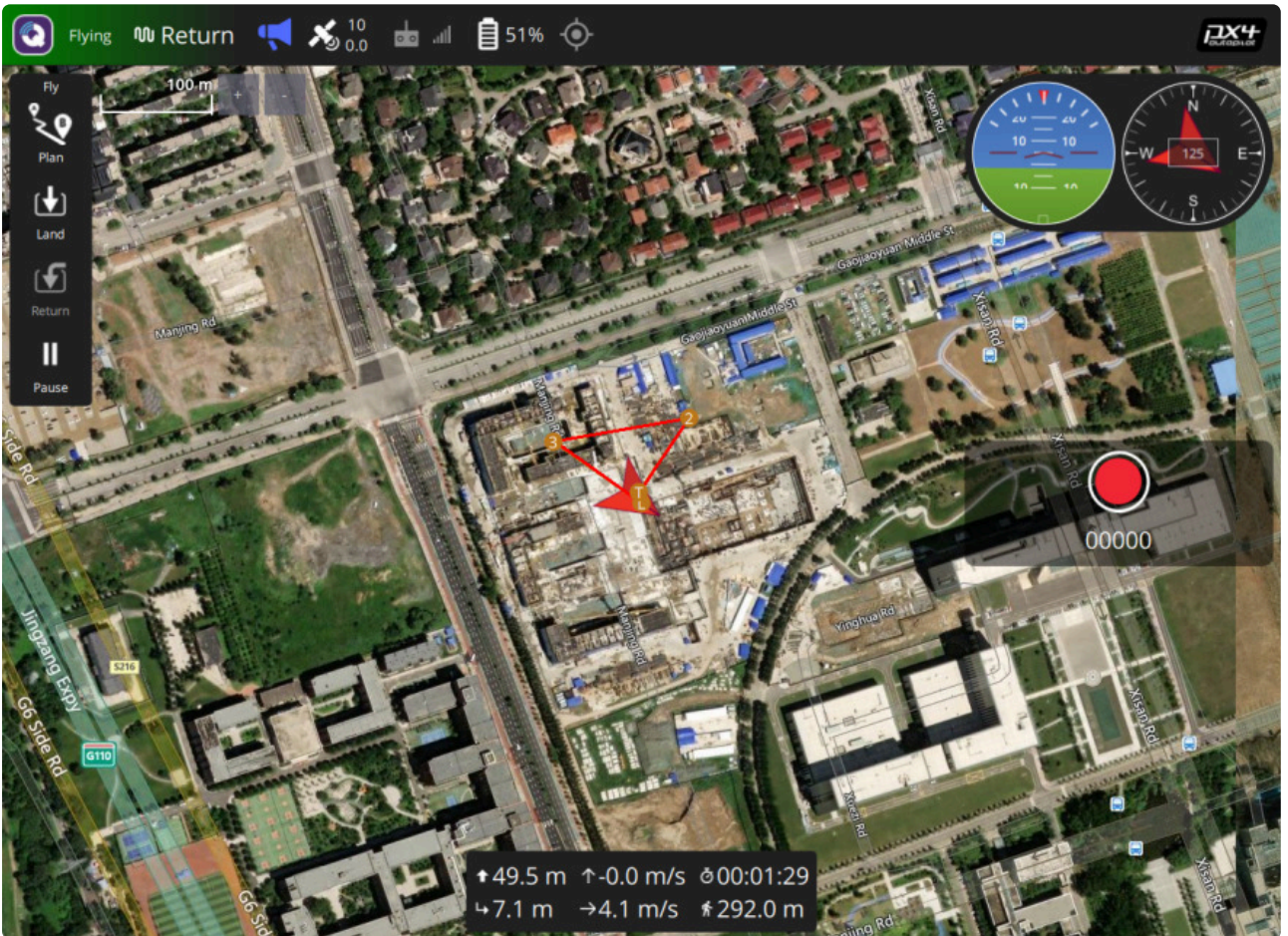
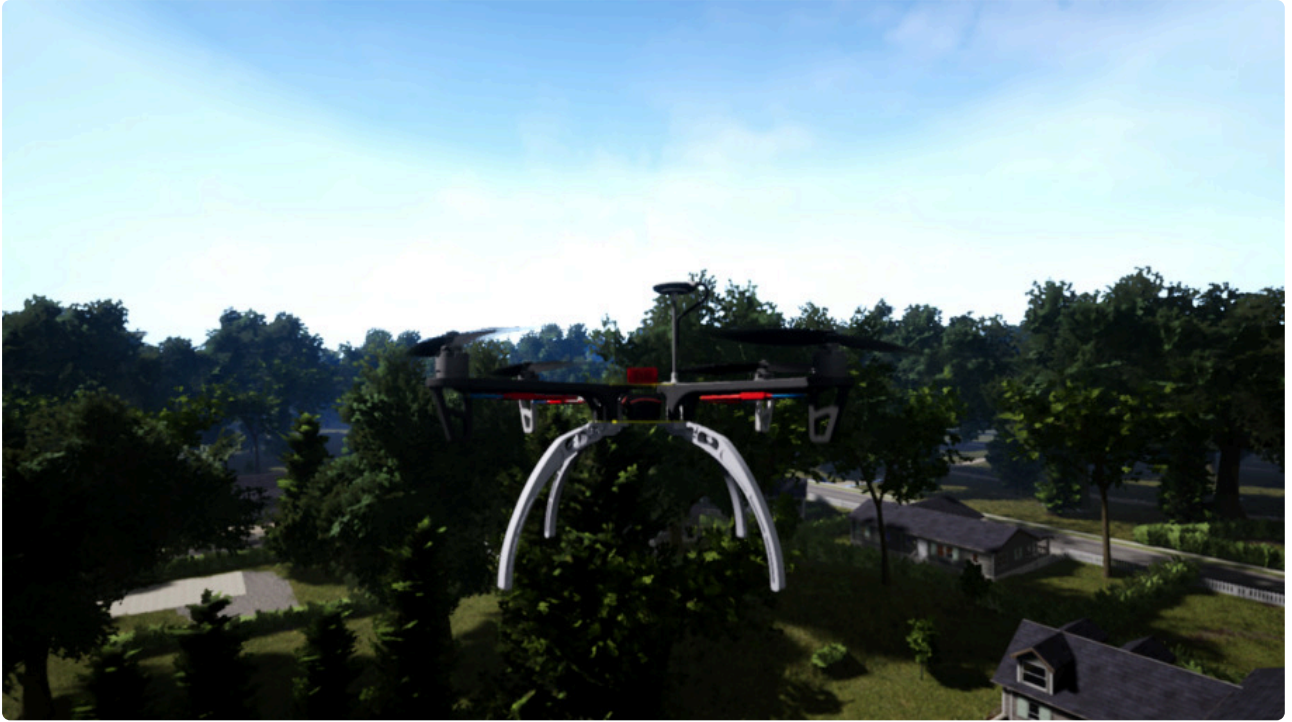
```
C:\WINDOWS\system32\cmd. x +
已复制 1 个文件。

-----
Please input UAV swarm number:1
Start QGroundControl
Kill all CopterSims
Starting PX4 Build
[1/1] Generating ../Logs
Using Airframe File: 10016_iris
starting instance 1 in /mnt/c/PX4PSP/Firmware/build/px4_sitl_default/instance_1
Copying rcS files
PX4 instances start finished
Press any key to exit
```



### Step 2: 观察结果

在QGC中可以如之前常规步骤一样，控制飞机的起飞、降落、航路飞行等。



注：由于本接口使用地形模块，利用TerrainZ实现了从CopterSim中读取当前地形高度数据，因此能适用于Grassland等带地形的地图。

## 5.3. 选做实验：硬件在环仿真












### Step 1: 连接飞控

如下图所示，将飞控通过USB线连接电脑，并确保完成硬件在环仿真配置。注意，本图使用Pixhawk6x飞控，其他飞控配置方法类似（推荐使用Pixhawk飞控）。

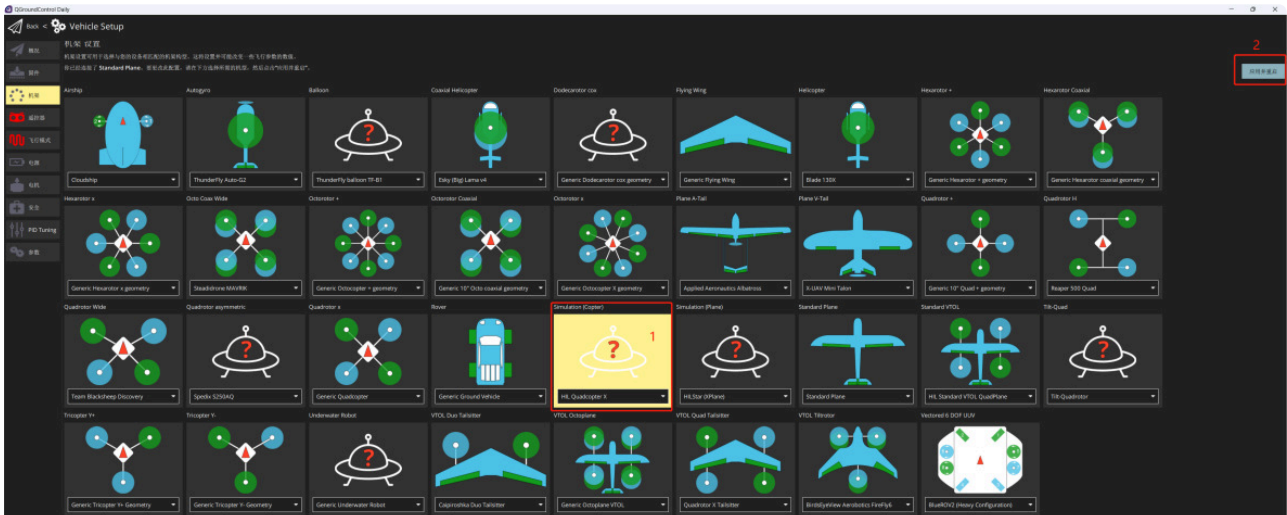


### Step 2: 设置硬件在环机架

在 Rflytools 文件夹中打开 QGC 地面站。

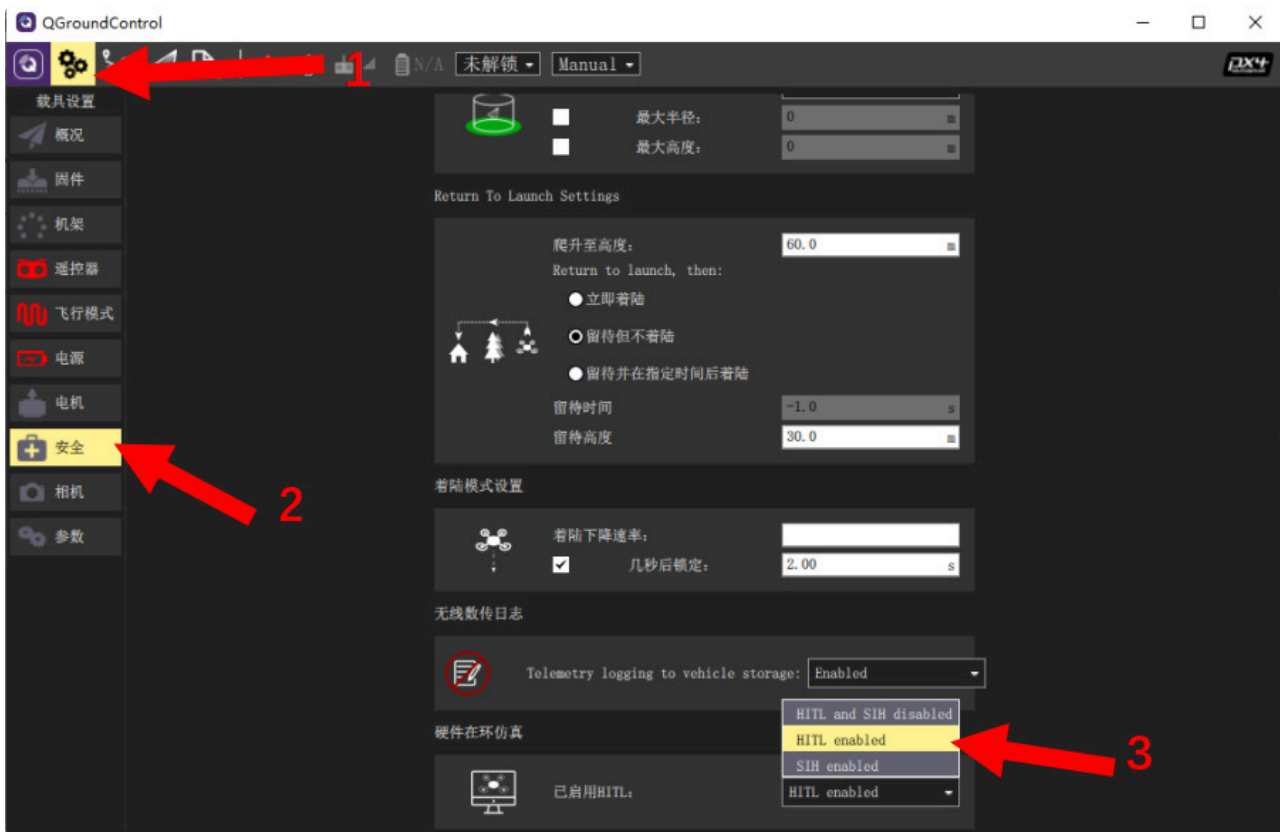
 3DDisplay	2023/7/27 15:02	快捷方式	1 KB
 CopterSim	2023/7/27 15:02	快捷方式	1 KB
 FlightGear-F450	2023/7/27 15:02	快捷方式	2 KB
 HITLRun	2023/7/27 15:02	快捷方式	2 KB
 Python38Env	2023/7/27 15:02	快捷方式	2 KB
 QGrounControl	2023/7/27 15:02	快捷方式	1 KB
 RflySim3D	2023/7/27 15:02	快捷方式	1 KB
 RflySimAPIs	2023/7/27 15:02	快捷方式	1 KB
 RflySimUE5	2023/7/27 15:02	快捷方式	1 KB
 SITLRun	2023/7/27 15:02	快捷方式	2 KB
 Win10WSL	2023/7/27 15:02	快捷方式	2 KB

打开QGC地面站，机架设置为“HIL Quadcopter X”，点击QGC右上角的“应用并重启”。



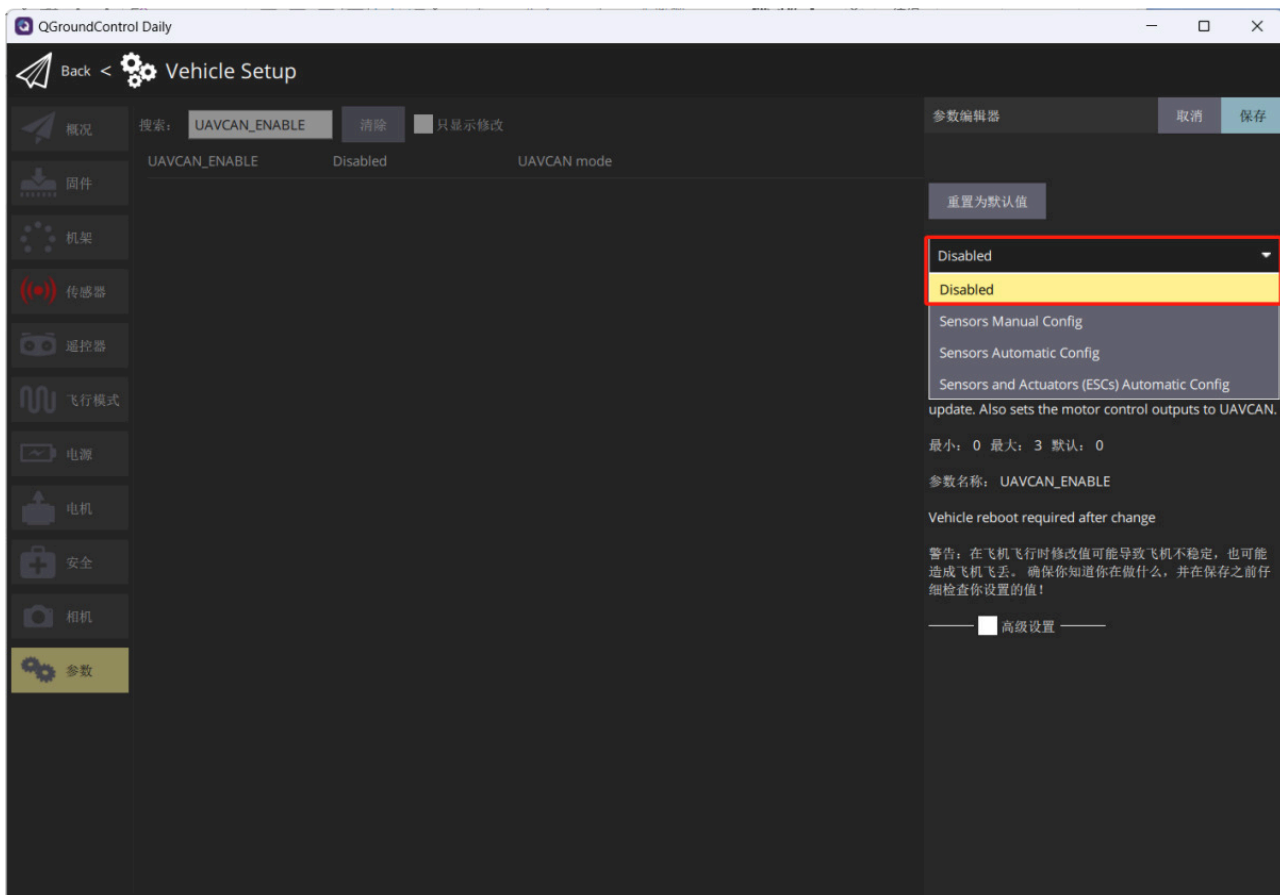
### Step 3: 配置硬件在环参数

在“安全”界面，选择“HITL enabled”启动硬件在环仿真，之后在概况界面中确认配置完成后，重新插拔飞控完成设置。



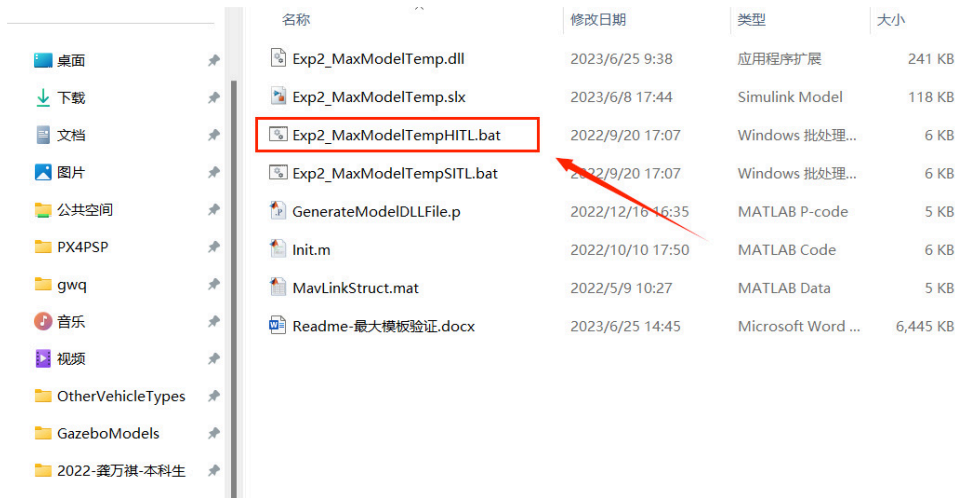
若使用PX4

1.13版本固件，还需要点击“参数”，在搜索栏中输入“UAVCAN\_ENABLE”，在弹出框中设置为“Disabled”，保存后重新插拔飞控即可。



## Step 4: 启动仿真

运行“Exp2\_MaxModelTempHITL.bat”脚本，可以自动打开硬件在环仿真，并完成所有需要配置，启动一架飞机的硬件在环仿真。



名称	修改日期	类型	大小
Exp2_MaxModelTemp.dll	2023/6/25 9:38	应用程序扩展	241 KB
Exp2_MaxModelTemp.slx	2023/6/8 17:44	Simulink Model	118 KB
Exp2_MaxModelTempHITL.bat	2022/9/20 17:07	Windows 批处理...	6 KB
Exp2_MaxModelTempSITL.bat	2022/9/20 17:07	Windows 批处理...	6 KB
GenerateModelDLLFile.p	2022/12/16 16:35	MATLAB P-code	5 KB
Init.m	2022/10/10 17:50	MATLAB Code	6 KB
MavLinkStruct.mat	2022/5/9 10:27	MATLAB Data	5 KB
Readme-最大模板验证.docx	2023/6/25 14:45	Microsoft Word ...	6,445 KB

## Step 5: 仿真过程

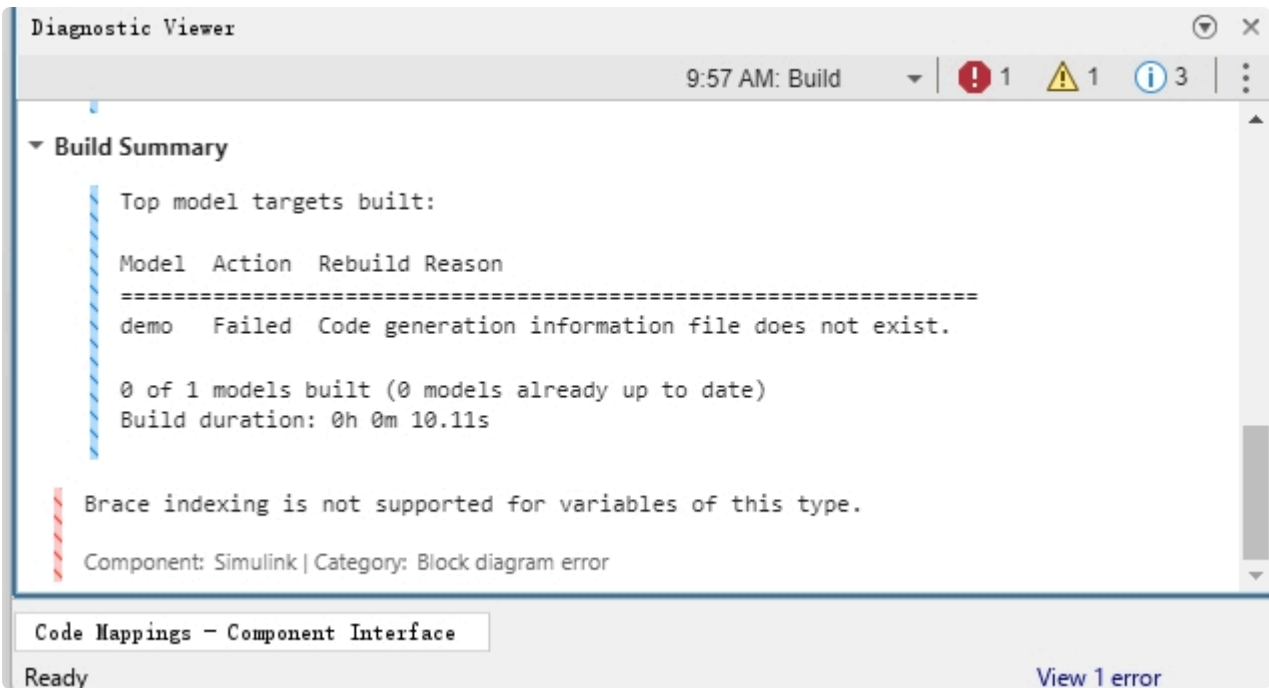
之后测试步骤参照5.2软件在环中的步骤进行。

## 6.参考资料

- 1.

## 7.常见问题

Q1: 未正确安装visual studio c++编译环境并配置mex，导致Simulink文件编译失败



A1: 首先将低于当前MATLAB版本的Visual Studio C++编译环境安装到VS默认安装目录，然后在MATLAB的命令行窗口中输入指令“mex -setup”，一般来说会自动识别并安装上支持的编译器，命令行显示“MEX 配置使用 ‘Microsoft Visual C++ 2017’ 以进行编译”的字样说明安装正确。详细环境配置参考” [RflySim平台安装目录]\RflySimAPIs\4.RflySimModel\API.pdf “中的环境配置



Q2: 编译报错，无法加载库文件



A2: 这可能是由于安装平台时PX4PSP工具箱未更新到最新版，更新RflySim安装包后按照如下配置重新安装平台即可

Toolbox one-key installation script: RflySimA...

(1) Software package installation directory  
C:\PX4PSP

(2) PX4 firmware compiling command: firmware versions <= PX4-1.8 use format px4fmu-v3\_default; >= PX4-1.9 use format px4\_fmu-v3\_default  
px4\_fmu-v6c\_default

(3) PX4 firmware version (1: PX4-1.7.3, ... , 6: PX4-1.12.3, 7: PX4-1.13.2, 8: PX4-1.14.4, 9: PX4-1.15.0)  
9

(4) PX4 firmware compiling toolchain (1: WinWSL[suitable for all versions], 2: Msys2[suitable for <= PX4-1.8], 3: Cygwin[for >=PX4-1.8])  
1

(5) Whether to reinstall PSP toolbox (yes to reinstall and no to remain current installation)  
yes

(6) Whether to reinstall the dependent software packages (CopterSim, QGroundControl, CopterSim, etc. About 5 minites)  
no

(7) Whether to reinstall the selected compiling toolchain (yes to reinstall and no to remain unchanged, about 5 minites)  
no

(8) Whether to reinstall the selected PX4 firmware source code (yes to reinstall and no to remain unchanged, about 5 minites)  
no

(9) Whether to pre-compile the selected firmware with the selected command (yes to compile and no to remain unchanged, about 5 minites)  
no

(10) Whether to block the actuator outputs in the PX4 firmware code ("yes" to use Simulink controller, "no" to use PX4 official controller)  
no

OK Cancel