

# 1. 实验名称及目的

## 1.1 实验名称

dll模型外部输入接口inDoubCtrls调用 (python sendSILIntDouble)

## 1.2 实验目的

RflySim平台模型具备double型数据输入接口——inDoubCtrls，为28维double型输入。对应的，[平台DllSimCtrlAPI.py](#)

python接口库中有2个接口用来对该DLL模型接口输入数据，分别是sendSILIntDouble()、sendInDoubCtrls()，该例程重点对sendSILIntDouble()接口进行说明。

## 1.3 关键知识点

### 1.3.1 接口数据解析

结构体定义如下：

```
struct DllInDoubCtrls{  
  
    int checksum;//校验码1234567897  
  
    int CopterID; // 飞机的ID  
  
    double inDoubCtrls[28];//28维的double型输入  
  
};
```

### 1.3.2 输入接口的Python使用方法

本例程使用的发送接口函数为sendSILIntDouble

接口使用实例解析：

导入必需的库和仿真API

```
import numpy as np
```

```
import time
```

```
import decimal
```

```
from DllSimCtrlAPI import DllSimCtrlAPI
```

numpy: 用于创建和操作数值数组，常用于科学计算。

time: 用于处理时间相关的任务。

decimal: 提供高精度的十进制数运算。

DllSimCtrlAPI: 从DllSimCtrlAPI模块导入的DllSimCtrlAPI类，这个类包含用于传输外部数据给dll模型的多个函数。

### **初始化DLL控制接口**

```
dll1 = DllSimCtrlAPI()
```

创建DllSimCtrlAPI类的一个实例，命名为dll1。

### **初始化发送到dll的数据**

```
insilInt = np.zeros(8).astype(int).tolist()
```

```
insilInt = np.arange(0,8,1)
```

```
inSILDoubs = np.zeros(20)
```

```
inSILDoubs[0] = decimal.Decimal('3.14159265358979323846')
```

```
inSILDoubs[1] = decimal.Decimal('4.14159265358979323846')
```

```
inSILDoubs[2] = decimal.Decimal('5.14159265358979323846')
```

insilInt初始化为从0到7的整数列表，用于某种类型的标识。

inSILDoubs是一个包含20个零值的浮点数组，前三个值被设置为高精度的十进制值。

### **30Hz循环执行发送任务**

```
while True:
```

```
lastTime = lastTime + timeInterval
```

```
sleepTime = lastTime - time.time()
```

```
if sleepTime > 0:
```

```
time.sleep(sleepTime) # sleep until the desired clock
```

```
else:
```

```
lastTime = time.time()
```

```
dll.sendSILIntDouble(insilInt, inSILDoubs)
```

根据timeInterval调整lastTime。

计算sleepTime，即距离下一个周期开始的剩余时间。

如果sleepTime为正，暂停执行直到下一个周期。

如果sleepTime不为正，说明执行时间已超出预定周期，更新lastTime为当前时间。

每个周期（30Hz，即每秒30次）调用dll.sendSILIntDouble函数发送整数列表和浮点数组到仿真系统。

## 2. 实验效果

以按上述修改后的DLL进行软件在环仿真，运行UDP30101Listener.slx监听程序，再运行 [sendSILIntDouble.py](#) 程序，可在simulink中Display中观察到通过 [sendSILIntDouble.py](#) 程序输入到DLL模型inDoubCtrls接口的数据。

## 3. 文件目录

例程目录：

[[安装目录](#)]\RflySimAPIs\4.RflySimModel\0.ApiExps\11.inSILAPI\2.inDoubCtrls\1.sendSILIntDouble

文件夹/文件名称	说明
<a href="#">..\Readme.pdf</a>	inDoubCtrls接口实验原理
Exp2_MaxModelTemp.dll	修改后的动态链接库
Exp2_MaxModelTemp.slx	Simulink模型文件
<a href="#">Exp2_MaxModelTemp_init.m</a>	模型参数文件

文件夹/文件名称	说明
<a href="#">Exp2_MaxModelTempSITL.bat</a>	软件在环仿真启动脚本
<a href="#">sendSILIntDouble.py</a>	Python接口sendSILIntDouble例程程序，向inDoubCtrls接口发送指定数据。
UDP30101Listener.slx	UDP 30101端口simulink监听程序。
<a href="#">Python38Run.bat</a>	Python程序运行脚本

## 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③。

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① 1台；\ \台；\ \台。

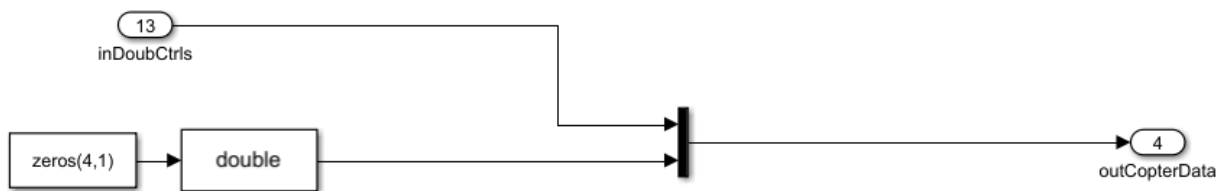
①：推荐配置请见：<https://rflysim.com/>

## 5. 实验步骤

### 5.1 必做实验：inDoubCtrls接口使用

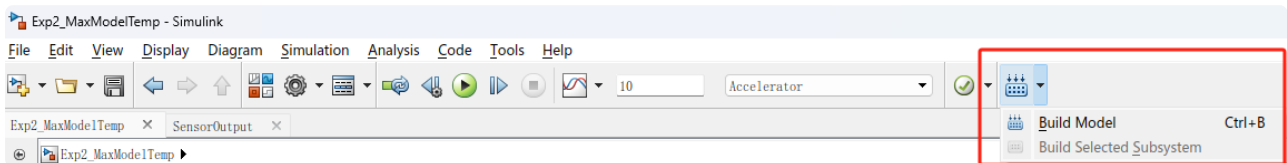
#### Step 1：增加输入输出接口

打开Exp2\_MaxModelTemp.slx文件，创建inDoubCtrls输入端口，设置为28维double型输入。将收到的数据发送到outCopterData接口[1]，为保证逻辑正确，合并上1个4维的double输入。

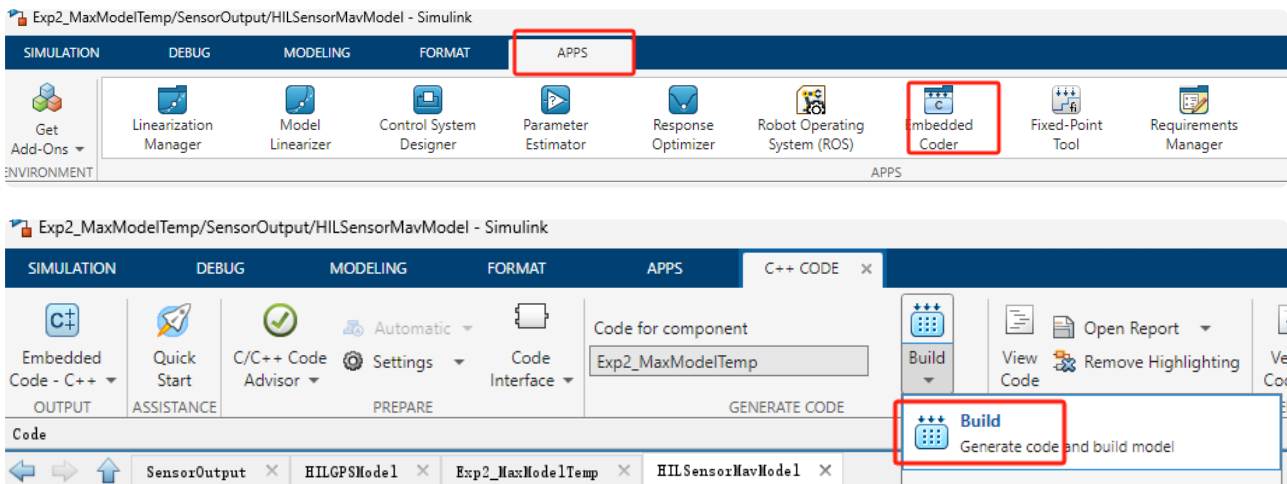


修改完成后，保存，并在Simulink中点击编译命令。

2017b



2022b以上

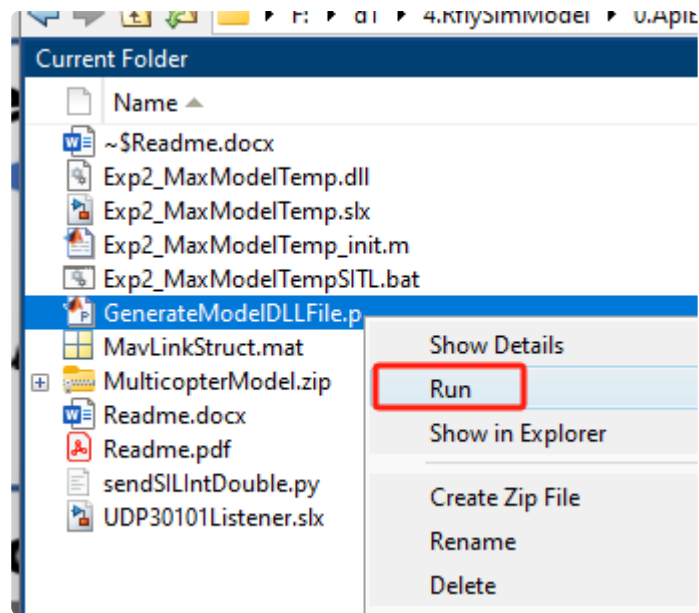


在 Simulink 的下方点击 View diagnostics

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，即表示编译成功。

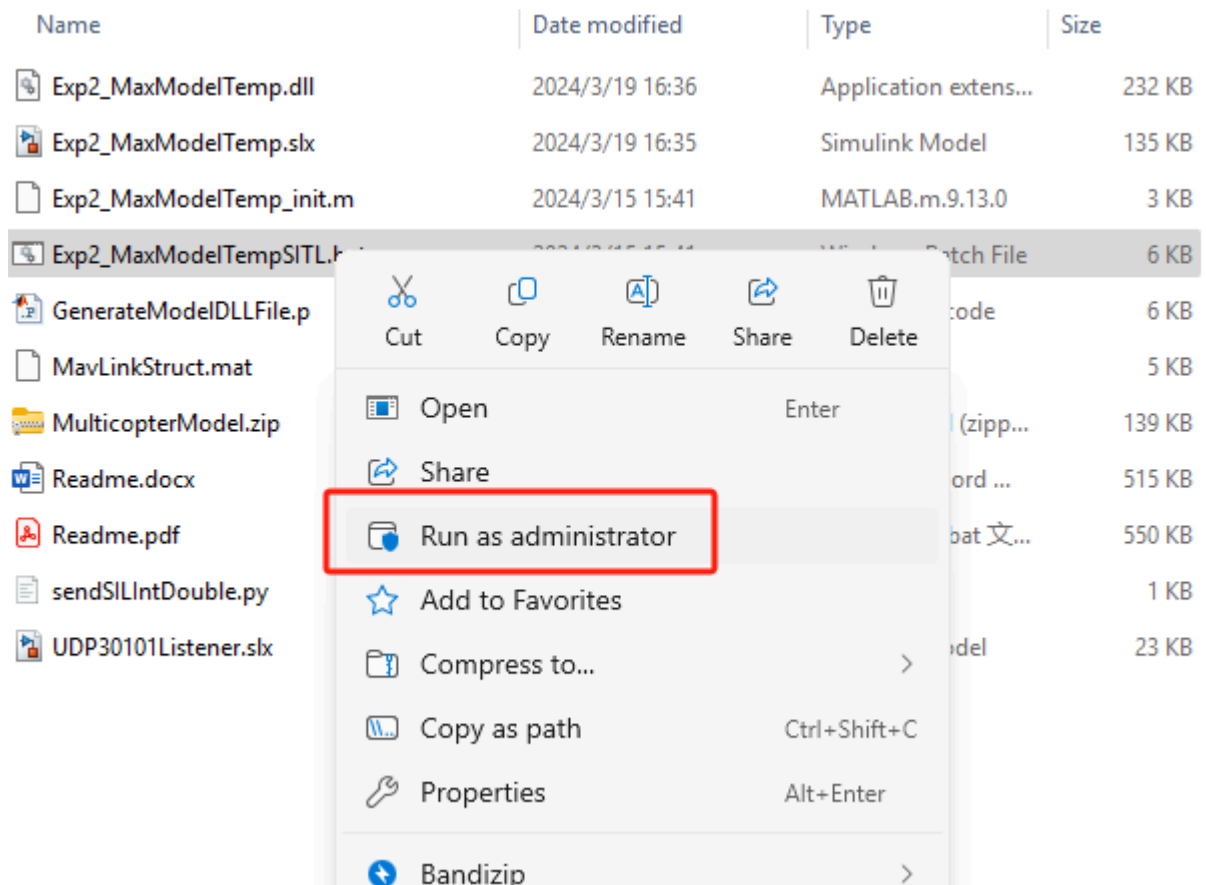
## Step 2: 生成DLL文件

右键运行 GenerateModelDLLFile.p 文件或在命令行窗口中输入 GenerateModelDLLFile后回车，得到修改后的动态链接库Exp2\_MaxModelTemp.dll。



### Step 3: 启动仿真

右键管理员运行 [Exp2\\_MaxModelTempSITL.bat](#) ,



输入1，启动1架无人机的软件在环仿真。

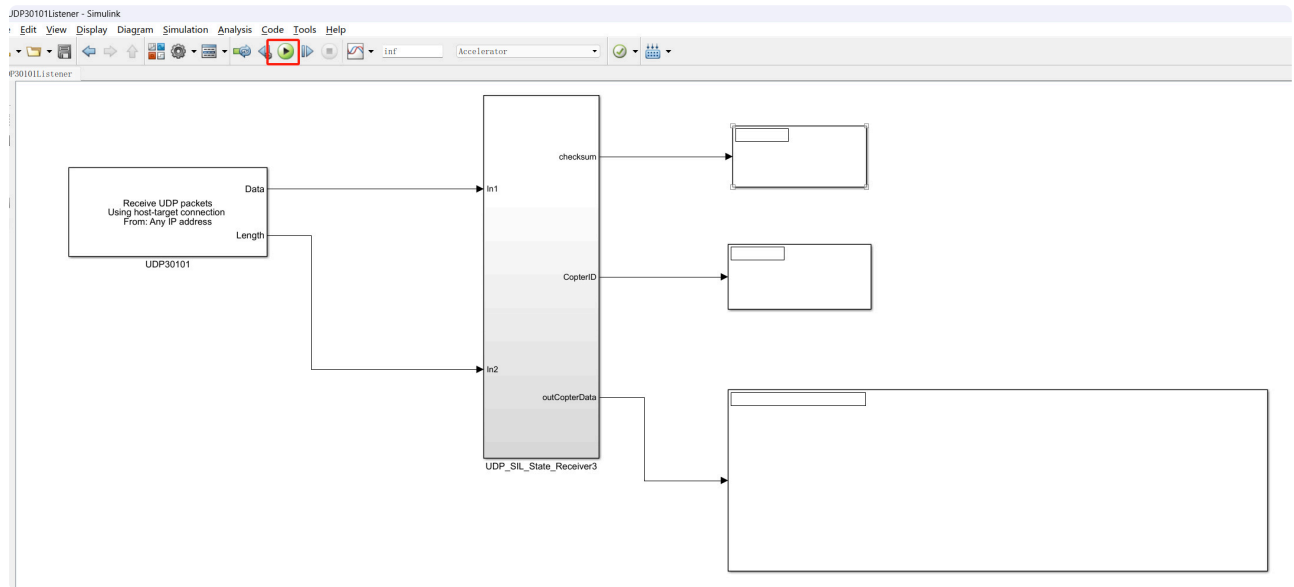
```
C:\WINDOWS\System32\cmd.exe

1 file(s) copied.

-----
Please input UAV swarm number:1
```

## Step 4: 运行Simulink模型

在MATLAB中打开UDP30101Listener.slx，运行，注意观察outCopterData端口的display。



## Step 5: 运行Python程序

在文件夹下，双击 [Python38Run.bat](#)，打开集成好的python环境，在该环境下运行 [sendSILIntDouble.py](#) 文件，输入 `python sendSILIntDouble.py`

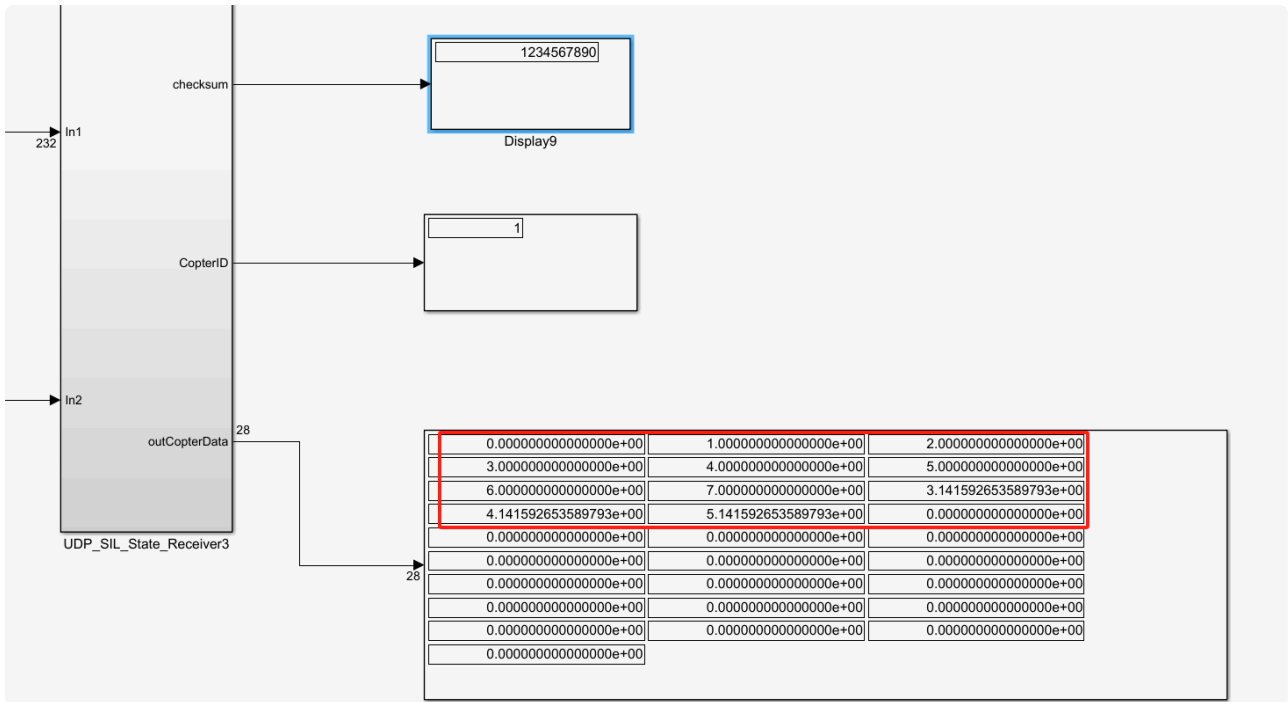
```
C:\Windows\system32\cmd.e...

Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.
You can use pip or pip3 command to install other libraries
Put Python38Run.bat into your code folder
Use the command: 'python XXX.py' to run the script with Python

D:\RfLySim\RfLySimLearn\4.RfLySimModel\0.ApiExps\11.inSILAPI\2.inDoubCtrls\1.sendSILIntDouble> python sendSILIntDouble.py
```

## Step 6: 观察结果

如下图所示，运行UDP30101Listener.slx监听程序，可以在display中观察到 [sendSILIntDouble.py](#) 输入到Dll模型inDoubCtrls接口的数据。



## 5.2 选做实验（VS Code调试运行）

### 准备工作

- 先确保已经按 [RflySimAPIs\1.RflySimIntro\2.AdvExps\e3.PythonConfig\Readme.pdf](#) 步骤，正确配置VS Code环境。或者配置了自己的Pycharm等自定义Python环境。
- 其他步骤与上文相同，运行 [sendSILIntDouble.py](#) 时，可使用VS Code（或Pycharm等工具）来打开 [sendSILIntDouble.py](#) 文件，并阅读代码，修改代码，调试执行等。

### 扩展实验

- 请自行使用VS Code阅读 [sendSILIntDouble.py](#) 源码，通过程序跳转，了解每条代码的执行原理；再通过调试工具，验证每条指令的执行效果。

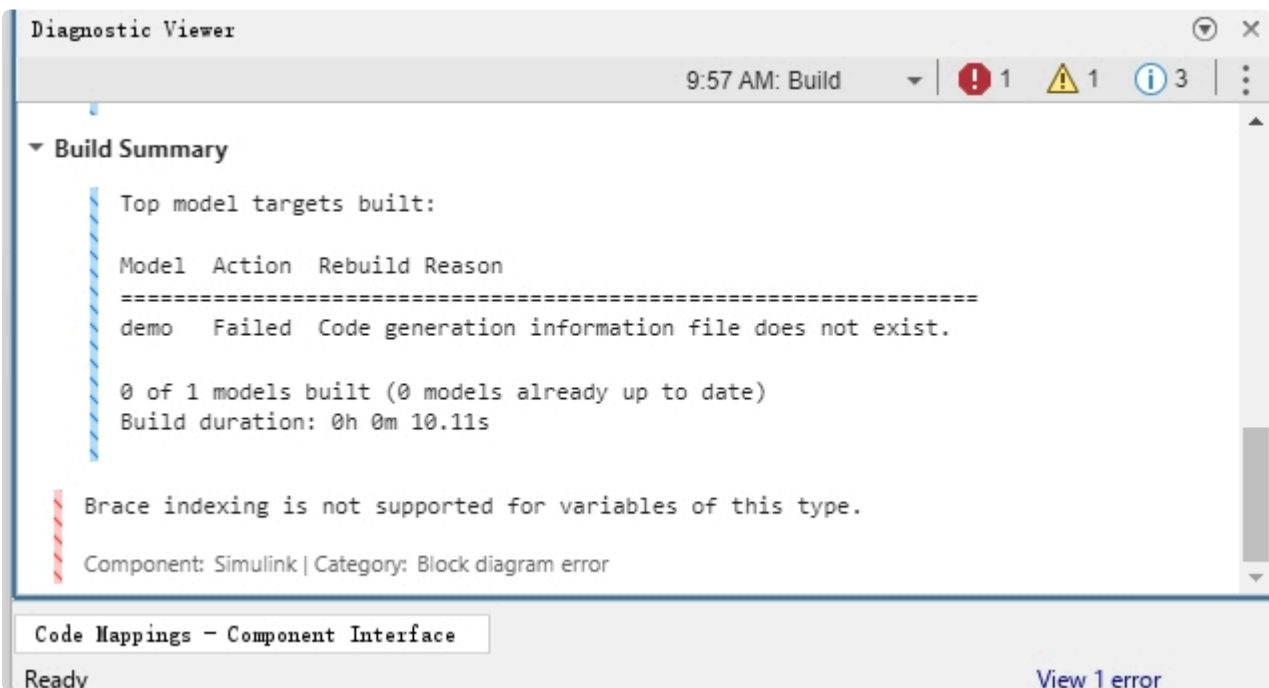
```
sendSILIntDouble.py X
E: > Test > cloud > inDoubCtrl > sendSILIntDouble > sendSILIntDouble.py > ...
17
18 insilInt=np.zeros(8).astype(int).tolist()
19 insilInt=np.arange(0,8,1)
20
21 inSILDoubs=np.zeros(20)
22 inSILDoubs[0]=decimal.Decimal('3.14159265358979323846')
23 inSILDoubs[1]=decimal.Decimal('4.14159265358979323846')
24 inSILDoubs[2]=decimal.Decimal('5.14159265358979323846')
25
26 while True:
27     lastTime = lastTime + timeInterval
28     sleepTime = lastTime - time.time()
29     if sleepTime > 0:
30         time.sleep(sleepTime) # sleep until the desired clock
31     else:
32         lastTime = time.time()
33     ## The following code will be executed 30Hz (0.0333s)
34     dll.sendSILIntDouble(insilInt,inSILDoubs)
35
```

## 6.参考资料

1. outCopterData接口 ..\..\8.OutCopterData\Readme.pdf
2. DLL/SO模型与通信接口 ..\..\PX4PSP\RflySimAPIs\4.RflySimModel\API.pdf
3. 外部控制接口 ..\..\PX4PSP\RflySimAPIs\4.RflySimModel\API.pdf

## 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（例如Visual C++ 2017），命令行显示“MEX 配置使用 ‘Microsoft Visual C++ 2017 (C)’ 以进行 C 语言编译。” [RflySim平台安装目录]\RflySimAPIs\4.RflySimModel\API.pdf “中的环境配置



```
命令窗口
>> mex -setup
MEX 配置为使用 'Microsoft Visual C++ 2017 (C)' 以进行 C 语言编译。
警告: MATLAB C 和 Fortran API 已更改, 现可支持
包含 2^32-1 个以上元素的 MATLAB 变量。您需要
更新代码以利用新的 API。
您可以在以下网址找到更多的相关信息:
http://www.mathworks.com/help/matlab/matlab\_external/upgrading-mex-files-to-use-64-bit

要选择不同的 C 编译器, 请从以下选项中选择一种命令:
Microsoft Visual C++ 2013 (C) mex -setup:D:\MATLAB\R2017b\bin\win64\mexopts\msvc2013.xml C
Microsoft Visual C++ 2015 (C) mex -setup:D:\MATLAB\R2017b\bin\win64\mexopts\msvc2015.xml C
Microsoft Visual C++ 2017 (C) mex -setup:C:\Users\dream\AppData\Roaming\MathWorks\MATLAB\R2

要选择不同的语言, 请从以下选项中选择一种命令:
mex -setup C++
mex -setup FORTRAN
fx >>
```

Q2: 编译报错，无法加载库文件



```
诊断台日志
下午4:48: 编译
-----
Exp1_modelTemp 信息: 保存文件完成工作结束。 无法编译。 有关详细信息, 请参阅编译日志。  ed
编译了 0 个模型, 共 1 个模型(0 个模型已经是最新的)
编译持续时间: 0h 0m 3.7699s

无法加载 "pixhawk_slib_adv\ControlForceModel" 引用的库 "pixhawk_slib_adv1"。
附件: Simulink | 类别: Block diagram 错误
代码映射 - 组件接口
```

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