

1. 实验名称及目的

1.1 实验名称

dll模型外部输入接口inSIL28d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——inSIL28d，为28维double型输入，该接口的使用方法和inDoubCtrls类似，[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 DllInSIL28d{  
  
int checksum;//校验码1234567897  
  
int CopterID; // 飞机的ID  
  
double inSIL28d[28];//28维的double型输入  
  
};
```

DllSimCtrlAPI中的sendSILIntDouble()函数和sendInDoubCtrls()函数将输入数据按照结构体定义格式打包并以UDP的方式通过30100++2系列端口发出；CopterSim运行中会循环监听该UDP端口号，监听到后将收到的数据转发给DLL模型的inSIL28d和inDoubCtrls接口。

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()

要注意的是，sendSILIntDouble()函数分了inSILInts和inSILDoubs传入数据，inSILInts为8维数据。

```
def sendSILIntDouble(self,inSILInts=[0]*8,inSILDoubs=[0]*20,copterID=-1):
```

```
    checkSum=1234567897
```

```
    ID=copterID
```

```
    if copterID<=0:
```

```
        ID=self.CopterID
```

```
    PortNum = 30100+(ID-1)*2
```

```
    inSILInts=self.fillList(inSILInts,8)
```

```
    inSILDoubs=self.fillList(inSILDoubs,20)
```

```
    buf = struct.pack("2i28d",checkSum,ID,*inSILInts,*inSILDoubs)
```

```
    self.udp_socket.sendto(buf, (self.ip, PortNum))
```

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 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。

DllSimCtrlAPI：导入DllSimCtrlAPI模块，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

```
dll = DllSimCtrlAPI.DllSimCtrlAPI()
```

创建 DllSimCtrlAPI 类的一个对象，命名为 dll。

执行发送到dll的任务

```
dll.sendSILIntDouble([1,2,3],np.array([4,5,6,7]))
```

- 调用 dll 对象的 sendSILIntDouble 方法，该方法接受两个参数：一个整数列表和一个浮点数组。
- 第一个参数 [1,2,3] 是一个整数列表，表示要发送给 dll 的三个整数值。
- 第二个参数 np.array([4,5,6,7]) 是一个浮点数组，表示要发送给 dll 的四个浮点值。

2. 实验效果

在CopterSim目录下创建csv日志文件，以修改后的DLL进行软件在环仿真，运行 [inSIL28dTest.py](#) 程序，结束仿真后打开csv文件，可以看到通过 [inSIL28dTest.py](#) 程序输入到DLL模型inSIL28d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\3.inSIL28d\1.sendSILIntDouble](#)

文件夹/文件名称	说明
..\Readme.pdf	inDoubCtrls接口实验原理
Exp1_MinModelTemp.dll	修改后的动态链接库
Exp1_MinModelTemp.slx	Simulink模型文件
Exp1_MinModelTemp_init.m	模型参数文件
Exp1_MinModelTemp.bat	软件在环仿真启动脚本
inSIL28dTest.py	Python接口sendSILIntDouble例程程序，向inSIL28接口发送数据
Python38Run.bat	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。

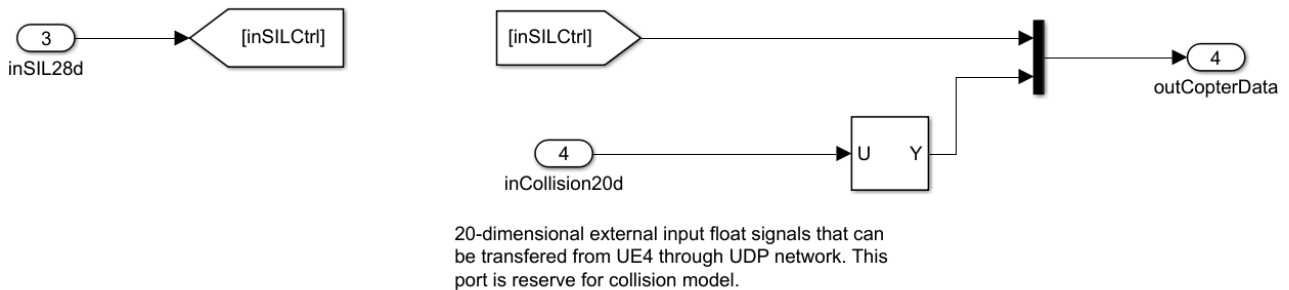
①：推荐配置请见：<https://rflysim.com/>

5. 实验步骤

5.1. 必做实验：inSIL28d接口使用

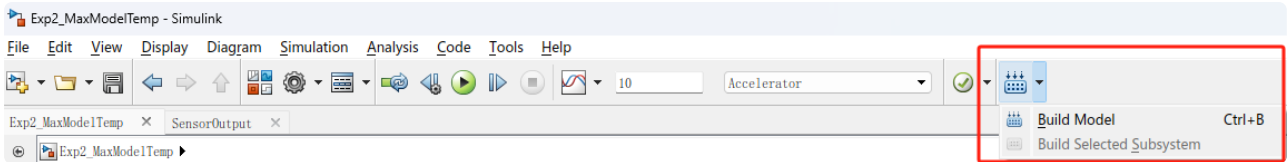
Step 1：增加输入输出接口

打开Exp1_MaxModelTemp.slx文件，创建inSIL28d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]。

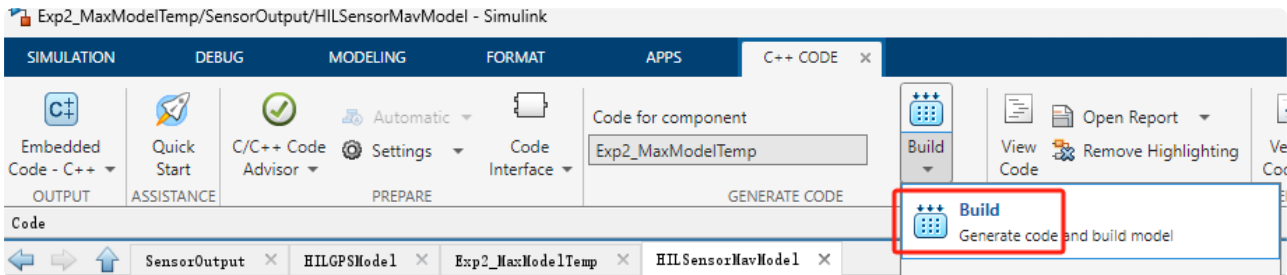
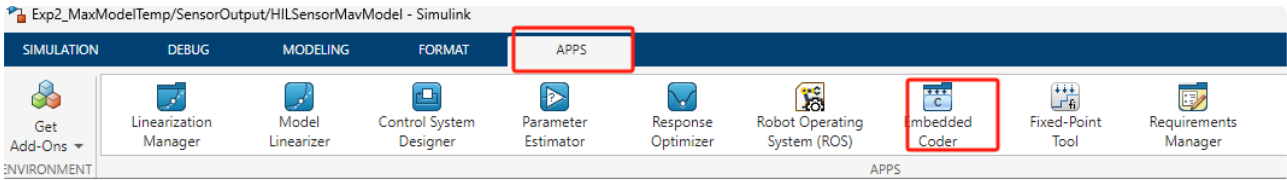


修改完成后，保存，并在Simulink中点击编译命令。

2017b



2022b以上



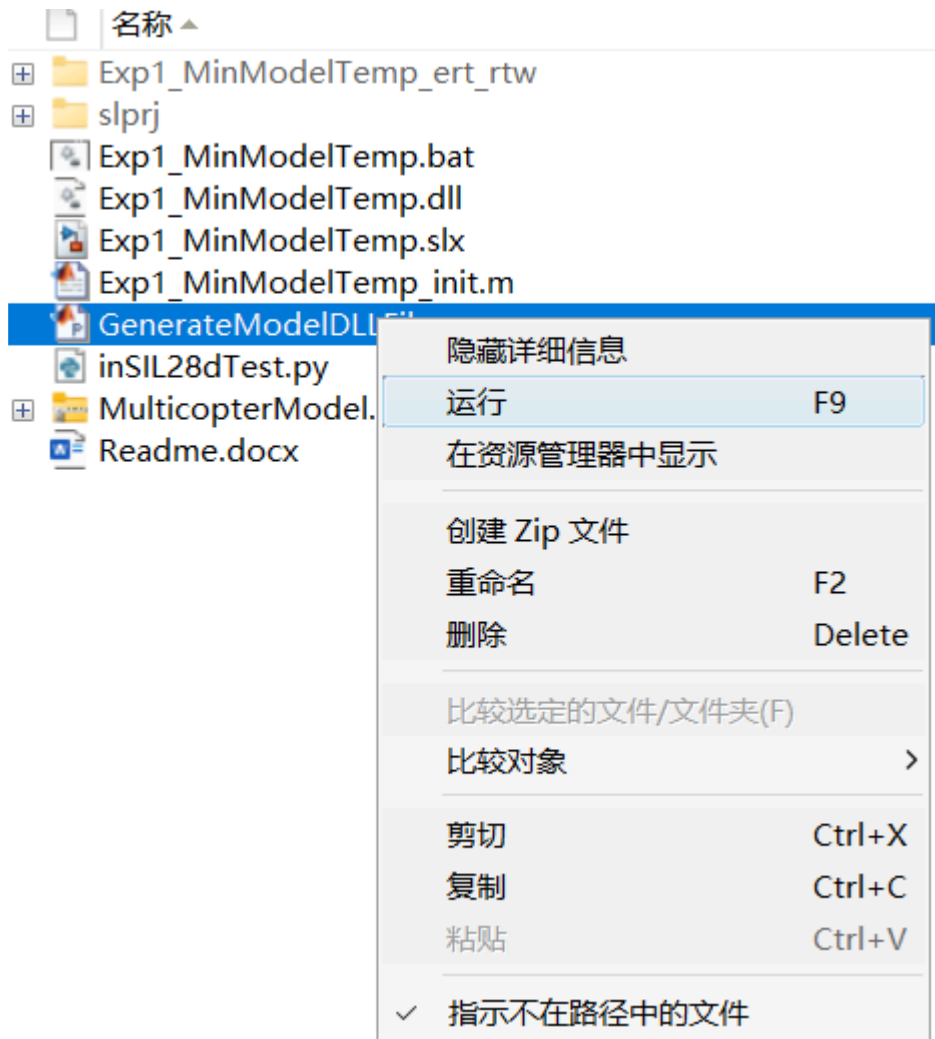
在 Simulink 的下方点击 View diagnostics

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Exp1_MaxModelTemp.dll。



Step 3: 新建日志文件

为了观察到实验数据，在平台CopterSim安装目录下新建1个csv文件，并以CopterSim1命名。

🔗 🗑️ ↑↓ 排序 ▾ ≡ 查看 ▾ ⋮

名称	修改日期	类型	大小
Qt5Core.dll	2018/12/3 19:34	应用程序扩展	5,304 KB
Qt5Gui.dll	2018/12/3 19:34	应用程序扩展	6,302 KB
Qt5Network.dll	2018/12/3 19:34	应用程序扩展	1,286 KB
Qt5SerialPort.dll	2018/12/3 22:19	应用程序扩展	79 KB
Qt5Sql.dll	2018/12/3 19:31	应用程序扩展	201 KB
Qt5Svg.dll	2018/12/3 22:19	应用程序扩展	323 KB
Qt5Widgets.dll	2018/12/3 19:38	应用程序扩展	5,442 KB
Readme.pdf	2024/5/6 11:44	快捷方式	2 KB
redis.conf	2023/5/19 18:15	CONF 文件	1 KB
redis++.dll	2023/5/18 13:30	应用程序扩展	538 KB
RFLY.ico	2020/3/29 17:09	ICO 文件	78 KB
RflyAPIs.ico	2021/11/17 20:57	ICO 文件	27 KB
ubuntu.ico	2019/4/24 16:57	ICO 文件	67 KB
Wgs84.dll	2024/4/22 9:42	应用程序扩展	139 KB
CopterSim1.csv	2024/5/7 15:03	Microsoft Excel ...	211 KB

Step 4: 启动仿真

运行Exp1_MaxModelTempSITL.bat,

Exp1_MinModelTemp_ert_rtw	2024/5/7 14:50	文件夹	
slprj	2024/5/7 14:49	文件夹	
Exp1_MinModelTemp.bat	2024/4/18 16:30	Windows 批处理...	6 KB
Exp1_MinModelTemp.dll	2024/5/7 14:50	应用程序扩展	220 KB
Exp1_MinModelTemp.slx	2024/5/7 11:26	Simulink Model	63 KB
Exp1_MinModelTemp_init.m	2024/5/7 14:49	Objective C 源文件	3 KB
GenerateModelDLLFile.p	2024/4/30 16:04	MATLAB.p.23.2.0	7 KB
MulticopterModel.zip	2024/5/7 14:50	压缩(zipped)文件...	94 KB
inSIL28dTest.py	2024/5/7 15:27	Python 源文件	1 KB
Readme.docx	2024/5/7 15:19	Microsoft Word ...	405 KB

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 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 [inSIL28dTest.py](#) 文件，输入 `python inSIL28dTest.py`

```

C:\Windows\system32\cmd.exe
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\3.inSIL28d\1.sendSILIntDouble>python inSIL28dTest.py

```

Step 6: 观察结果

仿真结束，关闭CopterSim等软件，打开CopterSim1.csv。

	A	B	C	D	E	F	G	H	I	J	K	L	M
2915	0	0	0	0	0	0	0	0	0	0	0	0	0
2916	0	0	0	0	0	0	0	0	0	0	0	0	0
2917	0	0	0	0	0	0	0	0	0	0	0	0	0
2918	0	0	0	0	0	0	0	0	0	0	0	0	0
2919	0	0	0	0	0	0	0	0	0	0	0	0	0
2920	0	0	0	0	0	0	0	0	0	0	0	0	0
2921	0	0	0	0	0	0	0	0	0	0	0	0	0
2922	0	0	0	0	0	0	0	0	0	0	0	0	0
2923	0	0	0	0	0	0	0	0	0	0	0	0	0
2924	0	0	0	0	0	0	0	0	0	0	0	0	0
2925	0	0	0	0	0	0	0	0	0	0	0	0	0
2926	0	0	0	0	0	0	0	0	0	0	0	0	0
2927	0	0	0	0	0	0	0	0	0	0	0	0	0
2928	0	0	0	0	0	0	0	0	0	0	0	0	0
2929	1	2	3	0	0	0	0	0	4	5	6	7	
2930	1	2	3	0	0	0	0	0	4	5	6	7	
2931	1	2	3	0	0	0	0	0	4	5	6	7	
2932	1	2	3	0	0	0	0	0	4	5	6	7	
2933	1	2	3	0	0	0	0	0	4	5	6	7	
2934	1	2	3	0	0	0	0	0	4	5	6	7	
2935	1	2	3	0	0	0	0	0	4	5	6	7	
2936	1	2	3	0	0	0	0	0	4	5	6	7	
2937	1	2	3	0	0	0	0	0	4	5	6	7	
2938	1	2	3	0	0	0	0	0	4	5	6	7	
2939	1	2	3	0	0	0	0	0	4	5	6	7	
2940	1	2	3	0	0	0	0	0	4	5	6	7	
2941	1	2	3	0	0	0	0	0	4	5	6	7	
2942	1	2	3	0	0	0	0	0	4	5	6	7	
2943	1	2	3	0	0	0	0	0	4	5	6	7	
2944	1	2	3	0	0	0	0	0	4	5	6	7	

从日志中观察到的数据与发送的一致。

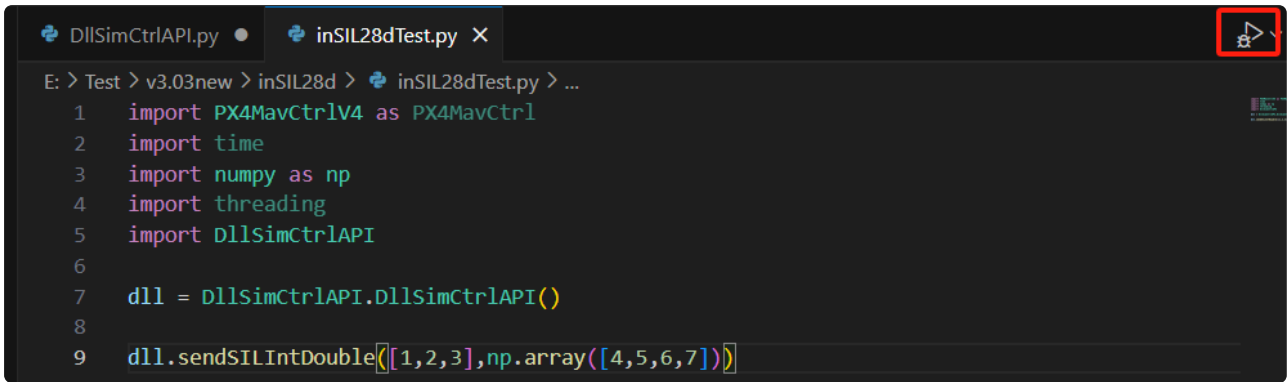
5.2. 选做实验（VS Code调试运行）

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 [inSIL28dTest.py](#) 时，可使用VS Code（或Pycharm等工具）来打开 [inSIL28dTest.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 [inSIL28dTest.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。



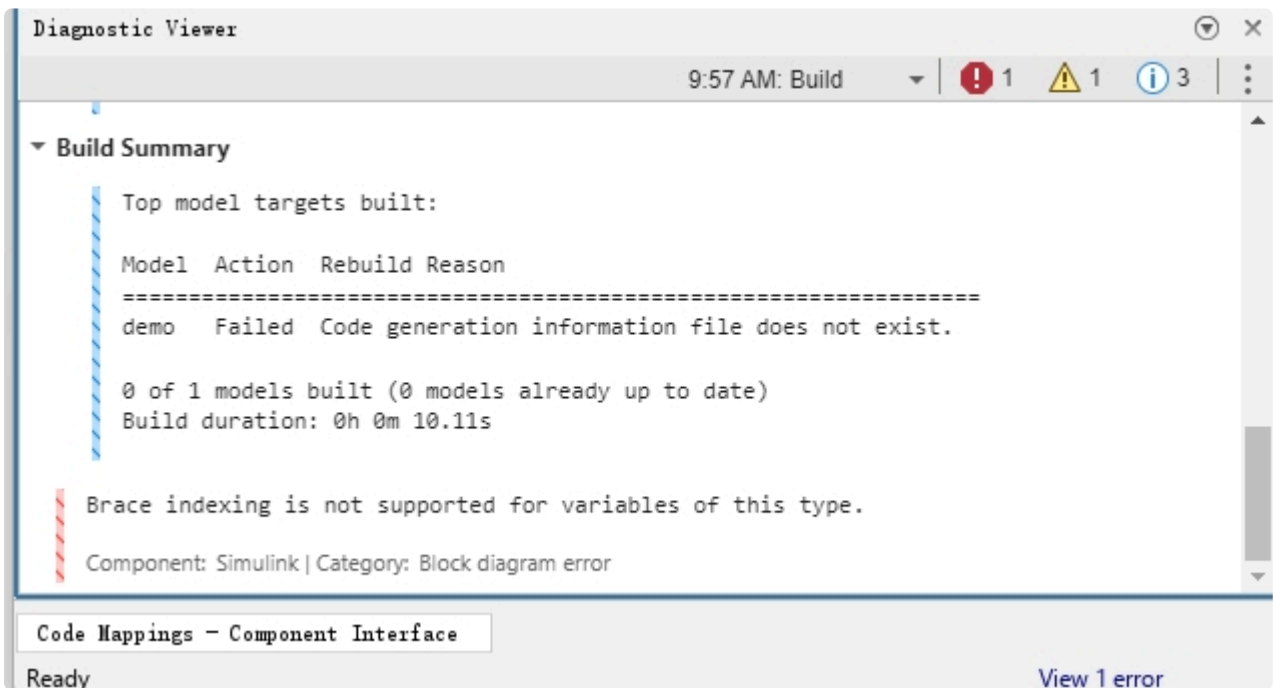
```
E: > Test > v3.03new > inSIL28d > inSIL28dTest.py > ...
1 import PX4MavCtrlV4 as PX4MavCtrl
2 import time
3 import numpy as np
4 import threading
5 import DllSimCtrlAPI
6
7 dll = DllSimCtrlAPI.DllSimCtrlAPI()
8
9 dll.sendSILIntDouble([1,2,3],np.array([4,5,6,7]))
```

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’ 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



```
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



```
诊断查看器
下午4:48: 编译
-----
Exp1_MintModelTemp 信息保存文件完成工作状况。 无法编译。有关详细信息, 请参阅编译日志。 ad
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 'pishawk_slib_adv/CosDerforcermodel' 引用的库 'pishawk_slib_adv'。
附件: Simulink | 美国 | Block diagram 错误
代码映射 - 应用程序
```

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