

1. 实验名称及目的

1.1 实验名称

dll模型ExtToPX4接口验证实验

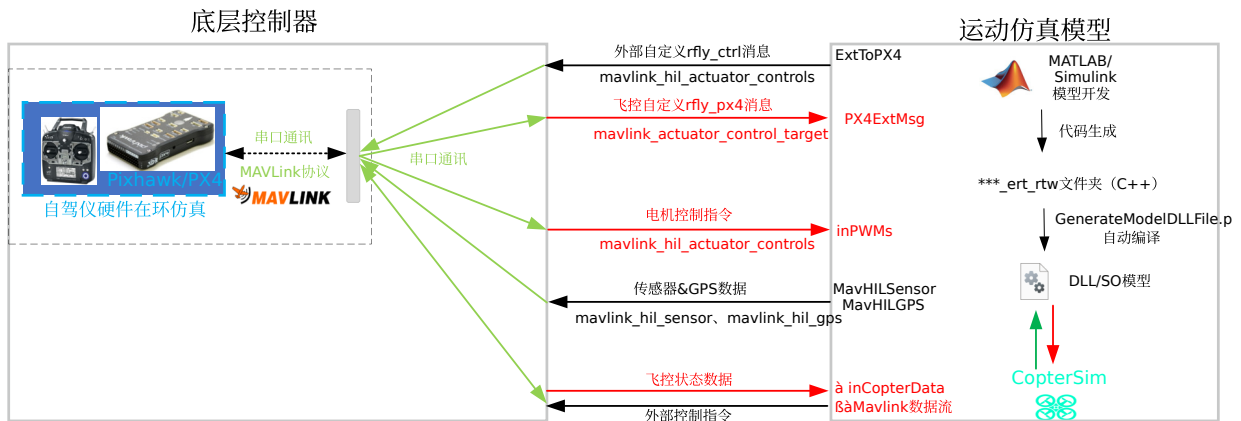
1.2 实验目的

该例程可以让用户自定义发送至最大模型ExtToPX4接口的数据，该接口为发送给PX4的uORB消息rfly_ext，用于传输其他传感器或必要数据给飞控，方便模型的开发及调试。

1.3 关键知识点

ExtToPX4接口可以将感兴趣的任意传感器数据或运动状态实时传输到飞控之中，基于此可以对模型实现更灵活的控制

1.3.1 接口数据解析



任意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

1.3.2 飞控中通过rfly_ext (uORB消息) 获取ExtToPX4接口数据

CopterSim在进行DLL模型加载时，会将ExtToPX4的16维数据转为“hil_actuator_controls”的mavlink消息，然后转发到PX4飞控中。RflySim平台修改了C:\PX4PSP\Firmware\src\modules\mavlink\mavlink_receiver.cpp飞控固件源码，增加了对hil_actuator_controls消息的支持。

PX4内部对mavlink消息的解析是通过这个mavlink_receiver.cpp文件实现的，从下图源码可见，rfly_ctrl和rfly_ext都是借用了hil_actuator_controls消息进行数据传输，当mode为123时，会通过rfly_ext发送给底层控制器，而其他情况则会通过rfly_ctrl (uORB消息) 发送给控制器。

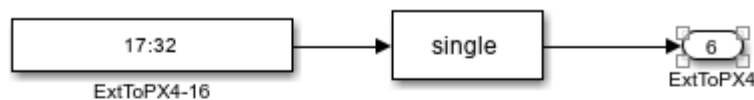
C: > PX4PSP > Firmware > src > modules > mavlink > **G+ mavlink_receiver.cpp**

```
109 MavlinkReceiver::handle_message(mavlink_message_t *msg)
110 {
111     switch (msg->msgid) {
112         case MAVLINK_MSG_ID_HIL_ACTUATOR_CONTROLS:{
113             mavlink_hil_actuator_controls_t hil_actuator_control;
114             mavlink_msg_hil_actuator_controls_decode(msg, &hil_actuator_control);
115             if(hil_actuator_control.mode==123){
116                 rfly_ext_s re{};
117                 re.timestamp = hrt_absolute_time();
118                 re.modes = 1;
119                 for(int i=0;i<16;i++){
120                     re.controls[i]=hil_actuator_control.controls[i];
121                 }
122                 _rfly_ext_pub.publish(re);
123             else{
124                 rfly_ctrl_s rc{};
125                 rc.timestamp = hrt_absolute_time();
126                 rc.modes = hil_actuator_control.mode;
127                 rc.flags = hil_actuator_control.flags;
128                 for(int i=0;i<16;i++){
129                     rc.controls[i]=hil_actuator_control.controls[i];
130                 }
131                 _rfly_ctrl_pub.publish(rc);
132             }
133             break;
134         }
```

注意，这里传输时强行设置了暗号mode=123，因此mavlink_receiver.cpp会将其解析为rfly_ext再转发到飞控中。

通过上述方式，从DLL模型发送17 18 …

32这几个数到了rfly_ext，链接飞控后通过在QGC中查看数据的方式，可以查看DLL模型的消息是否正确传入。



double ExtToUE4[16]; // This signal will be sent to UE4 as the 9 to 26 of actuator's inputs. Besides, this value can be shown in UE4's D mode, so you can observe the value of the model through UE4.

float ExtToPX4[16]; // this value will be sent to PX4 with uORB msg rfly_ext. So you can transfer some sensor data to you generated PX4 controller.

2. 实验效果

将修改后的最大模型编译成动态链接库后，启动硬件在环仿真，通过QGroundControl分析工具里的Mavlink控制台监听rfly_ext消息，可以看到模型通过ExtToPX4接口发送出来的数据

3. 文件目录

例程目录：[\[安装目录\]\RflySimAPIs\4.RflySimModel\0.ApiExps\7.ExtToPX4](#)

文件夹/文件名称	说明
Readme.pdf	dll模型自定飞控数据输出接口ExtToPX4验证实验原理
Exp2_MaxModelTemp.dll	由最大模型生成的动态链接库
Exp2_MaxModelTemp.slx	最大模型源程序
Exp2_MaxModelTempHITL.bat	最大模型软件在环仿真启动脚本
Exp2_MaxModelTemp_init.m	最大模型中初始化参数

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

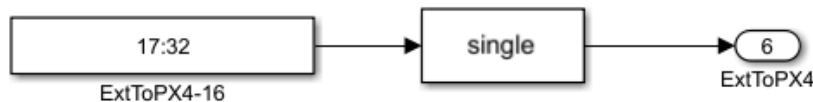
笔记本/台式电脑^① 1台；Pixhawk6X或其他飞控 1台；数据线 1台。

5. 实验步骤

Step 1: 查看Simulink模型

打开MATLAB软件，在MATLAB中打开Exp2_MaxModelTemp.slx文件，定位到模型ExtToPX4接口，最大模型模板默认输入到ExtToPX4接口的数据为[17:32]，用户可自定义该数据，本例程中我们采用默认输入，不作改动。

注：如果修改，需要重新编译模型。



```
double ExtToUE4[16]; // This signal will be sent to UE4 as  
the 9 to 26 of actuator's inputs. Besides, this value can be  
shown in UE4's D mode, so you can observe the value of  
the model through UE4.
```

```
float ExtToPX4[16]; // this value will be sent to PX4 with  
uORB msg rfly_ext. So you can transfer some sensor data  
to you generated PX4 controller.
```

Step 2: 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，例程使用Pixhawk6x飞控，其他飞控配置方法类似（推荐使用Pixhawk飞控）。



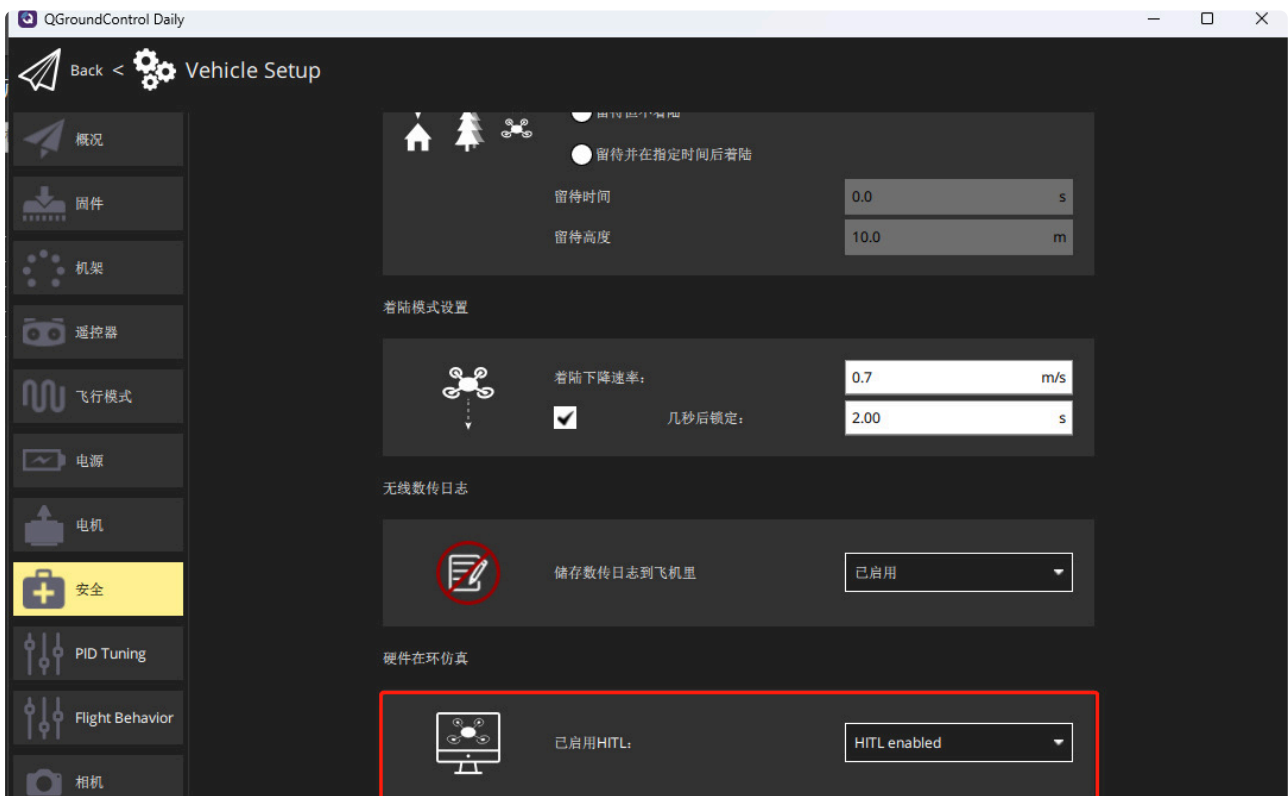
Step 3: 设置硬件在环机架

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

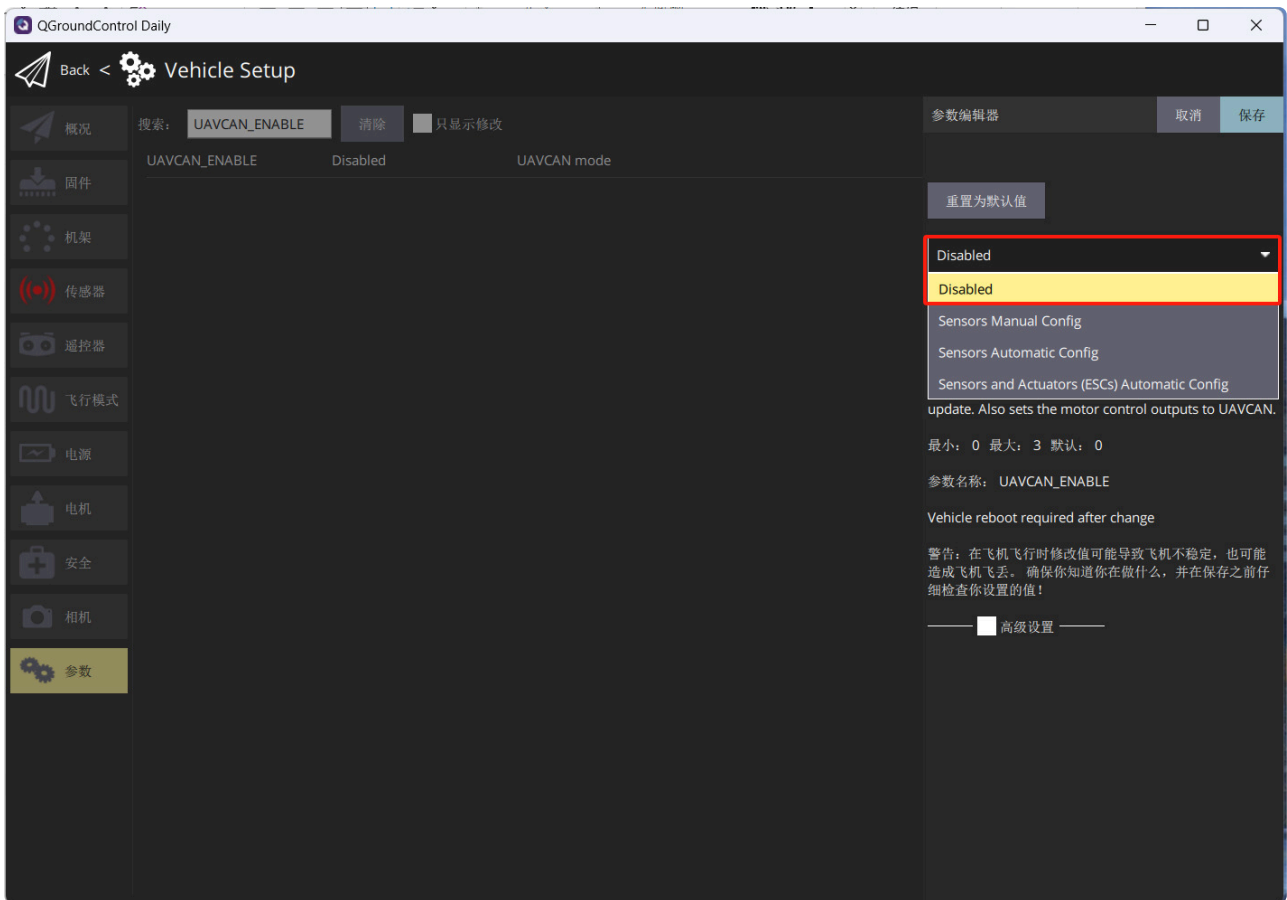


Step 4: 配置硬件在环参数

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。



如果使用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 5: 启动仿真

右键点击 [Exp2_MaxModelTempHITL.bat](#) 并以管理员身份运行，根据提示输入飞控对应端口号，启动硬件在环仿真。

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

已复制      1 个文件。

-----

Please input the Pixhawk COM port list for HIL
Use ',' as the separator if more than one Pixhawk
E.g., input 3 for COM3 of Pixhawk on the computer
Input 3,6,7 for COM3, COM6 and COM7 of Pixhawks

Available COM ports on this computer are:
COM4: USB ???

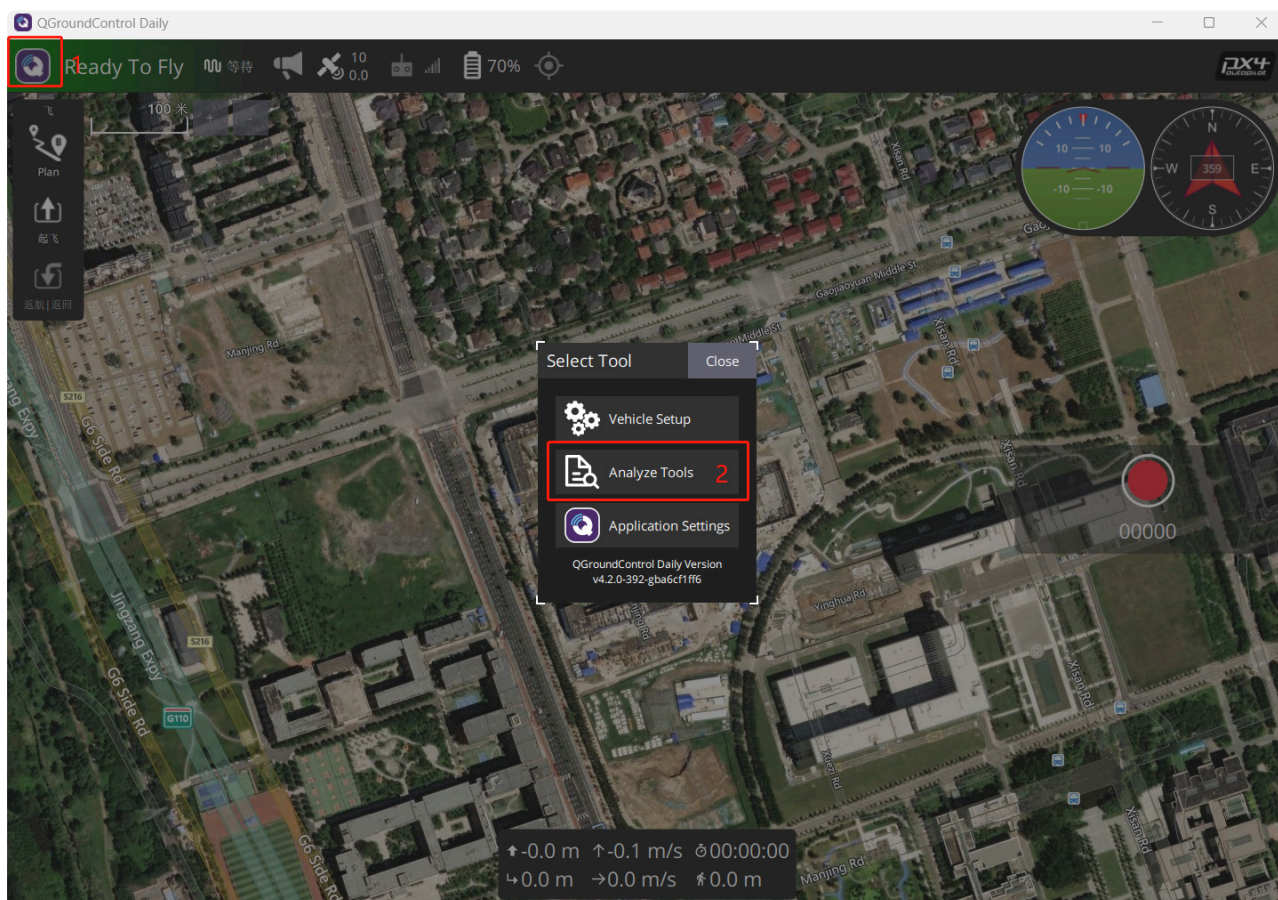
Recommended COM list input is: 4

-----

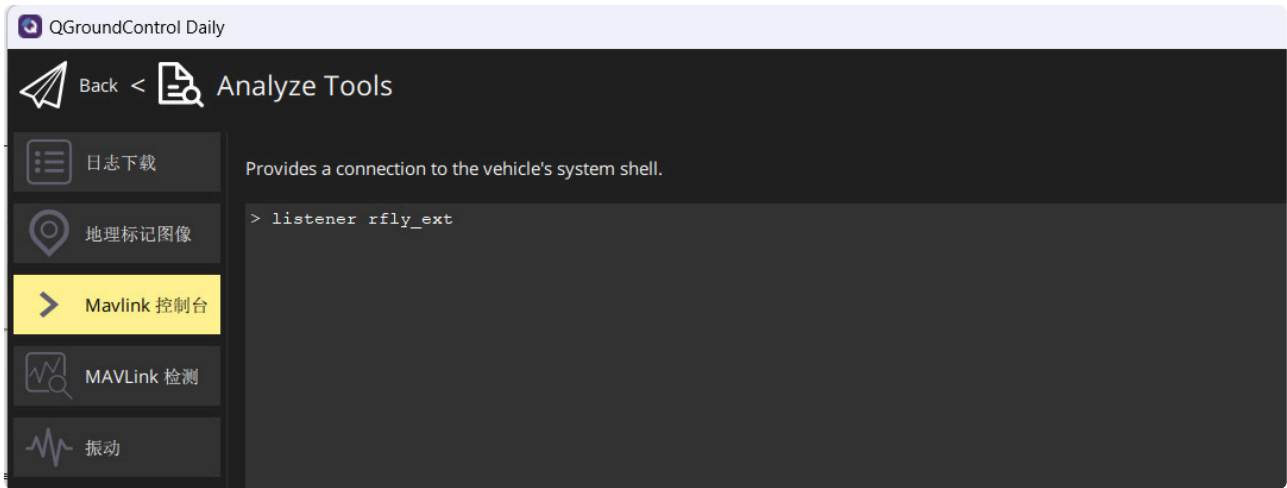
My COM list for HITL simulation is:4_
```

Step 6: 观察结果

待GPS 3D Fixed完成后，点击QGroundControl软件左上角图标，进入“Analyze Tools”。



选择“Mavlink控制台”，输入listener rfly_ext，回车。



如下图所示，可看到收到的controls为17-32，与输入一致。

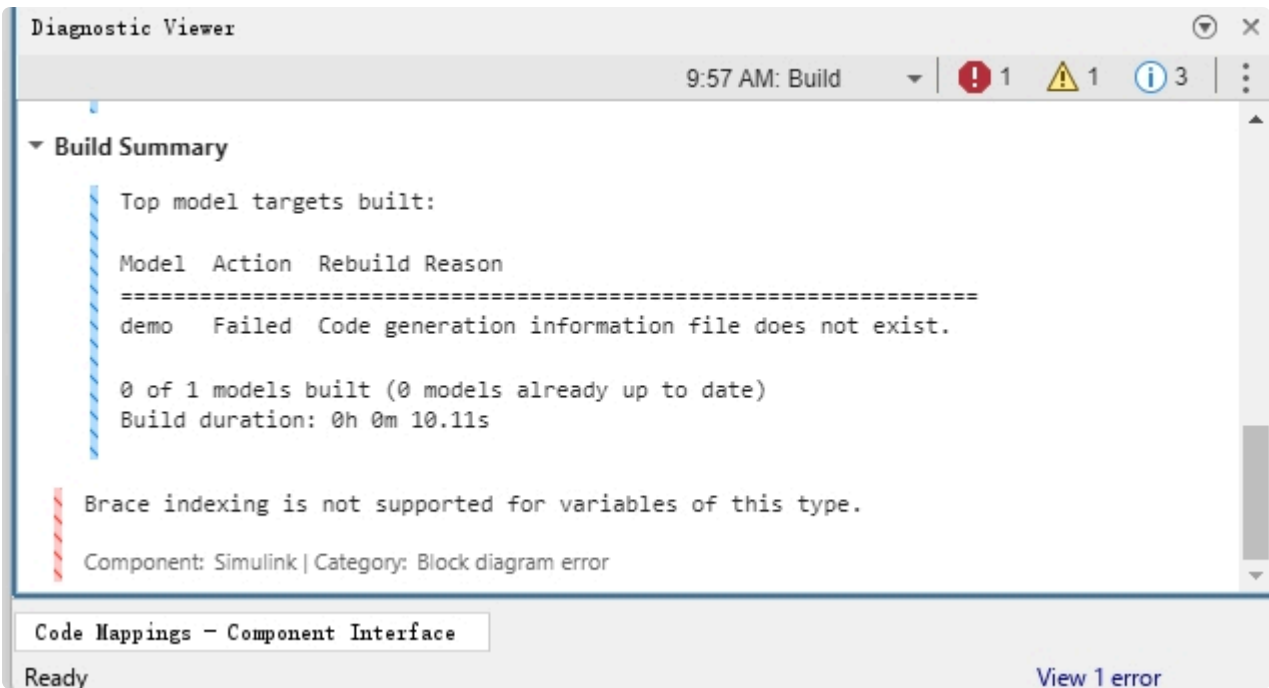
```
listener rfly_ext
TOPIC: rfly_ext
rfly_ext_s
timestamp: 767168634 (0.004366 seconds ago)
controls: [17.0000, 18.0000, 19.0000, 20.0000, 21.0000, 22.0000, 23.0000, 24.0000, 25.0000, 26.0000, 27.0000, 28.0000, 29.0000, 30.0000, 31.0000, 32.0000]
modes: 1
nsh> |
```

6.参考资料

1. 自动代码生成外部通信接口 [..\..\PX4PSP\RflySimAPIs\5.RflySimFlyCtrl\API.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](#)
- 4.

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



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