

1.实验名称及目的

1.1实验名称

dll模型inCopterData输入接口（uORB消息rfly_px4）实验

1.2实验目的

在平台的模型例程中，除了必备的几个输入输出接口为平台的基本功能服务之外，还有一些输入接口能发送一些更为细致的载具仿真信息，其中inCopterData为CopterSim发往DLL模型的32维输入接口，其中25~32维可以监听rfly_px4消息；本次通过其中的inCopterData（25~32）设计实验，使用户熟悉掌握该输入接口的使用方法。

1.3关键知识点

inCopterData是32维double型数据，前8维存储PX4的状态，9-24维接收ch1-ch16 RC通道信号（遥控器输入），25-32维监听rfly_px4 uORB消息。

6

inCopterData

32-dimensional external input double signals from CopterSim, the definition is listed as follows.

1-8: PX4 state flags for simulation.
9-24: RC channel signals (ch1~ch16)
25-32: listen to rfly_px4

note:

inCopterData(1): armed flag of PX4
inCopterData(2): RCNum. Total number of RC channels being received. This value should be 0 when no RC channels are available.
inCopterData(3): simulation Mode. 0 for HITL, 1 for SITL, 2 for SimNoPX4
inCopterData(4): is 3D fixed in CopterSim.
inCopterData(5): is VTOL_STATE from PX4.
inCopterData(6): is LANDED_STATE from PX4.

Note:

The RC channel can be used to trigger animation during the simulation.

VTOL_STATE:

0: UNDEFINED
1: TRANSITION_TO_FW //正在转换到FW
2: TRANSITION_TO_MC //正在转换到MC
3: MC //多旋翼模式
4: FW //固定翼模式

7

inFromUE

来自UE的消息，32维double型数组，用于处理模型和场景的交互

自定义uORB消息rfly_px4.msg

将PX4内部数据，通过UDP或mavlink协议传输外部，:\PX4PSP\Firmware\msg文件夹下的rfly_px4.msg定义其格式：

```
uint64timestamp #timesincesystemstart(microseconds)
```

```
float32[8]control #8Dcontrolsignals
```

1) MAVLink协议

RflySim平台修改

了“Firmware\src\modules\mavlink\streams\ACTUATOR_CONTROL_TARGET.hpp”文件,会订阅rfly_px4并转发为ACTUATOR_CONTROL_TARGET的MAVLink消息,并被CopterSim接收。

```
---
109     actuator_controls_s act_ctrl;
110     if(N==0){
111         rfly_px4 s rf_px;
112         if (_rfly_px4_sub && _rfly_px4_sub->update(&rf_px)) {
113             mavlink_actuator_control_target_t msg{};
114
115             msg.time_usec = rf_px.timestamp;
116             msg.group_mlx = 123;
117
118             for (unsigned i = 0; i < sizeof(msg.controls) / sizeof(msg.controls[0]); i++) {
119                 msg.controls[i] = rf_px.control[i];
120             }
121
122             mavlink_msg_actuator_control_target_send_struct(_mavlink->get_channel(), &msg);
123
124             return true;
125     }
```

消息详细定义见:

https://mavlink.io/en/messages/common.html#ACTUATOR_CONTROL_TARGET

ACTUATOR_CONTROL_TARGET (#140)

[Message] Set the vehicle attitude and body angular rates.

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
group_mlx	uint8_t		Actuator group. The "_mlx" indicates this is a multi-instance message and a MAVLink parser should use this field to difference between instances.
controls	float[8]		Actuator controls. Normed to -1..+1 where 0 is neutral position. Throttle for single rotation direction motors is 0..1, negative range for reverse direction. Standard mapping for attitude controls (group 0): (index 0-7): roll, pitch, yaw, throttle, flaps, spoilers, airbrakes, landing gear. Load a pass-through mixer to repurpose them as generic outputs.

2) 40100系列端口

CopterSim接收到ACTUATOR_CONTROL_TARGET消息后,会判断group_mlx是否等于暗号123,如果是则转发如下结构体到40100系列端口。

```
structPX4ExtMsg{
```

```
intchecksum;//1234567898

intCopterID;

doublerunnedTime;//Currentstamp(s)

floatcontrols[8];

}
```

注意：CopterSim向外发的端口是奇数号，收的端口是偶数号，因此1号飞机，向外发布PX4ExtMsg消息对应的40100系列端口实际上是40101。

根据上述协议，inCopterData（25~32）发送的是监听到的PX4消息。通过ExtToPX4向PX4发送特定数组后，再写入rfly_px4 uORB消息中，那么inCopterData（25~32）发送的也是相同的特定数组。

2. 实验效果

以最大模型启动硬件在环仿真，待仿真初始化完成后，inCopterData（25~32）可以监听rfly_px4。

3. 文件目录

例程目录：

[\[安装目录\]\RflySimAPIs\4.RflySimModel\0.ApiExps\9.inCopterData\3.rfly_px4](#)

文件夹/文件名称	说明
Exp2_MaxModelTemp.dll	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	最大模型硬件在环启动脚本
PX4ExtMsgReceiver.slx	rfly_px4 uORB消息文件
PX4ExtMsgSender.slx	外部通信接口程序

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

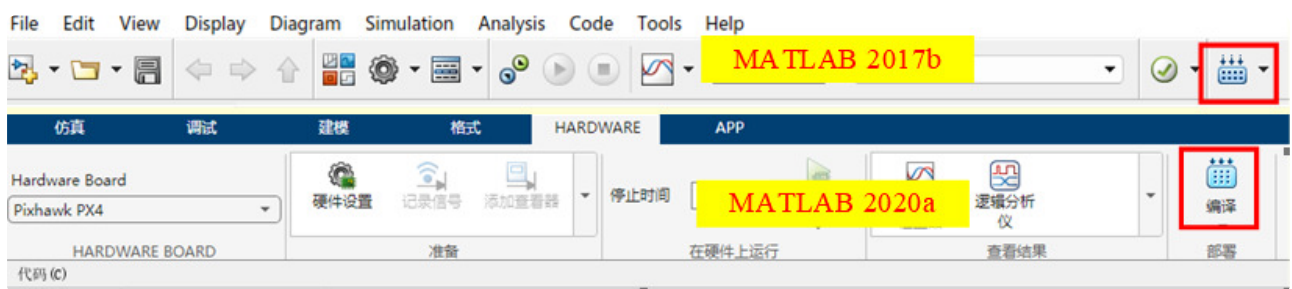
笔记本/台式电脑① 1台；Pixhawk 6X或其他飞控 1台；数据线 1台。

①：推荐配置请见：<https://rflysim.com/>

5. 实验步骤

Step 1: 编译模型

打开MATLAB软件，在MATLAB中打开PX4ExtMsgReceive.slx文件，在Simulink中，点击编译命令。



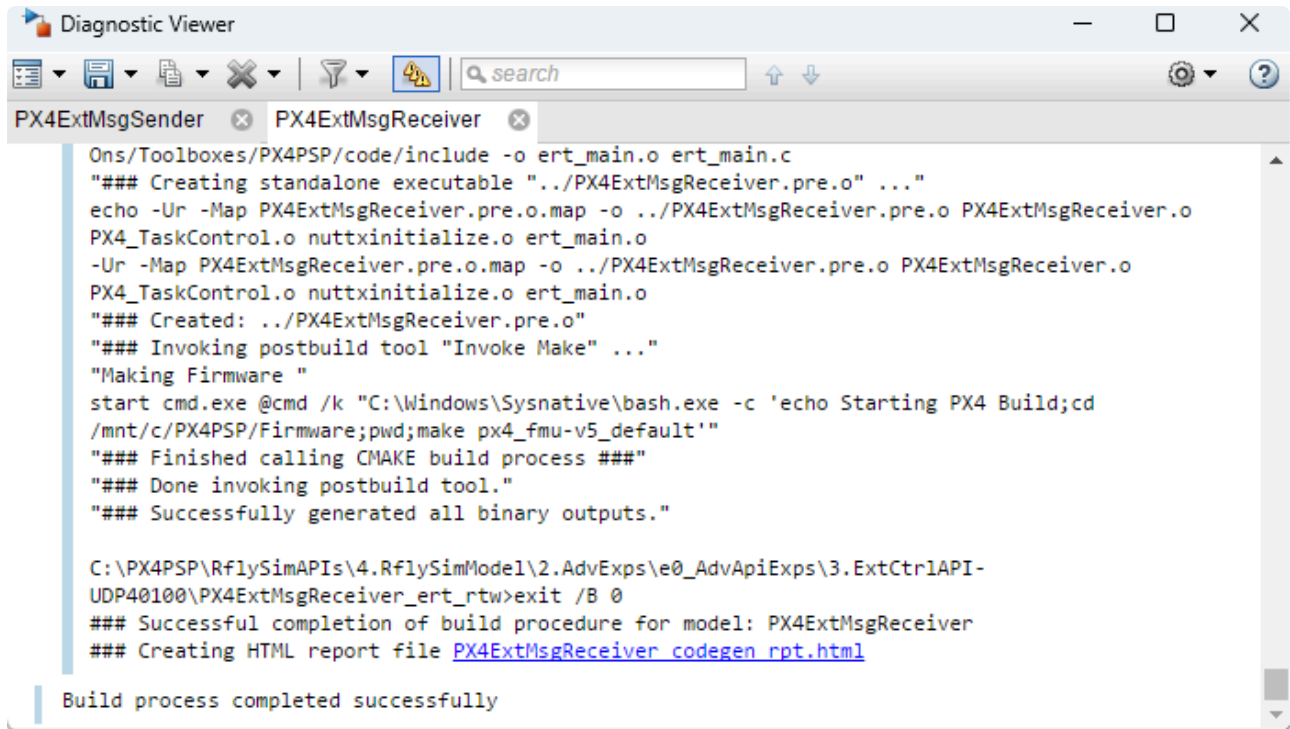
注意：编译PX4ExtMsgReceive.slx文件与下面的编译Exp2_MaxModelTemp.slx文件不同，编译PX4ExtMsgReceive.slx文件需要设置Pixhawk硬件，本文件中已设置。

Step 2: 等待编译成功

在Simulink的下方点击View

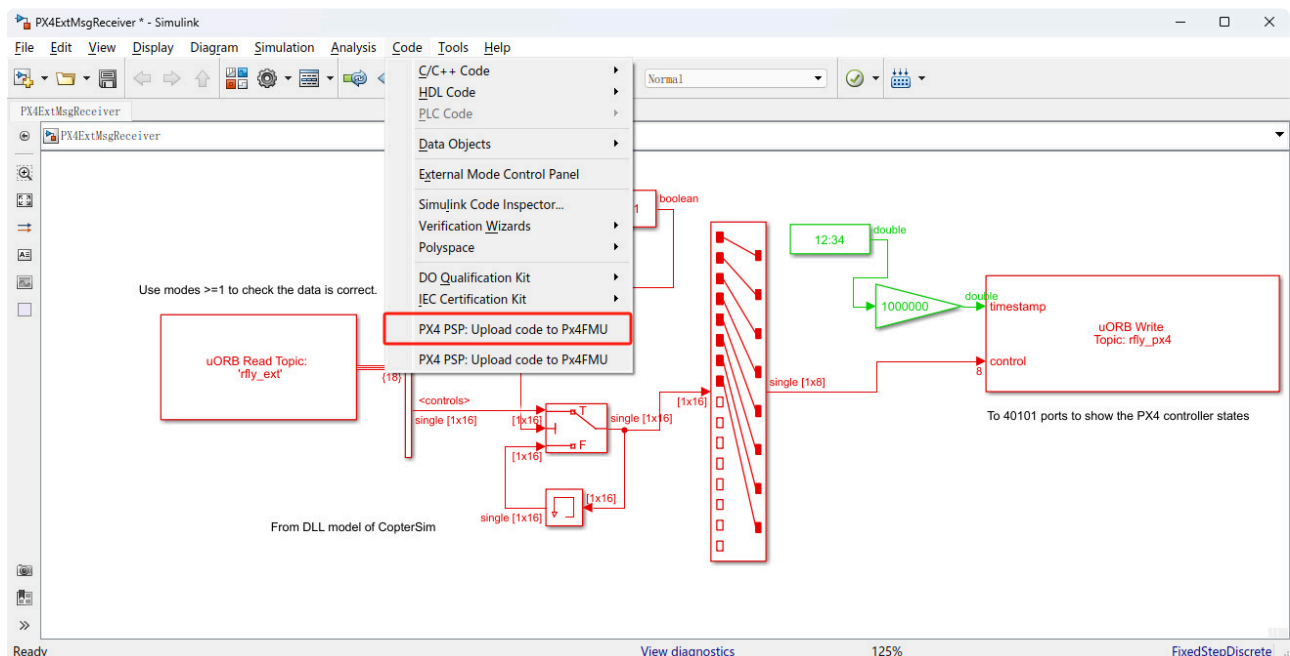
diagnostics指令，即可弹出诊断对话框，可查看编译过程。在诊断框中弹出Build process

completed successfully, 即表示编译成功。



Step 3: 烧录固件

点击Simulink的Code选项，在下拉条目中选择PX4 PSP: Upload code to Px4FMU，弹出命令框后将飞控通过USB线连接至电脑，开始固件烧录。



当进度条到达100%时，表明固件烧录完毕。

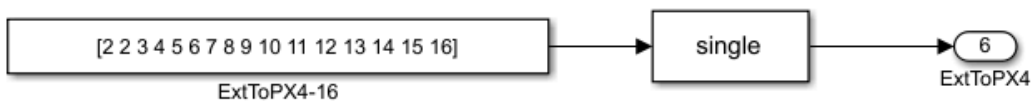
```
C:\WINDOWS\SYSTEM32\cmd.exe
Loaded firmware for board id: 1010,0 size: 1793056 bytes (97.71%), waiting for the bootloader...
Attempting reboot on COM4 with baudrate=57600...
If the board does not respond, unplug and re-plug the USB connector.

Found board id: 1010,0 bootloader version: 5 on COM4
sn: 003200223231510f38383932
chip: 20036450
family: b'STM32H7[4]5x'
revision: b'V'
flash: 1966080 bytes
Windowed mode: False

Erase : [=====] 100.0%
Program: [=====] 100.0%
Verify : [=====] 100.0%
Rebooting. Elapsed Time 75.586
```

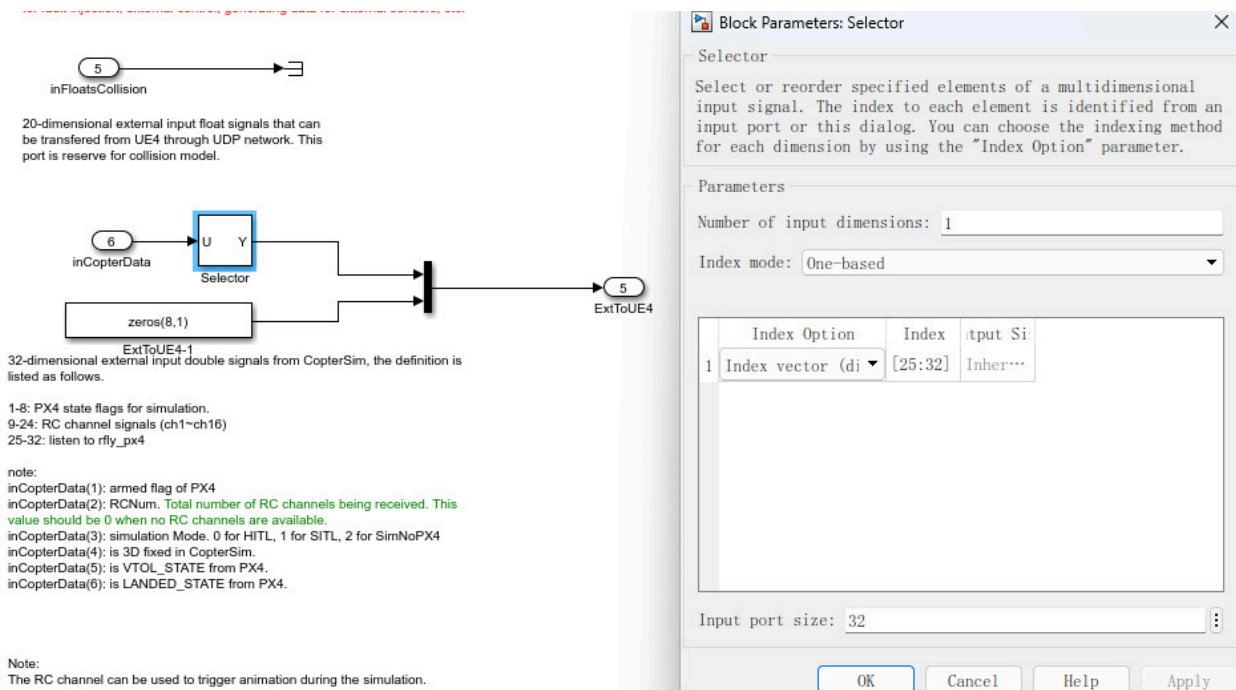
Step 4: 修改模型并编译

打开 DLLModelTemp 文件夹，在 MATLAB 中打开 Exp2_MaxModelTemp.slx 文件，将 simulink 文件中的 ExtToPx4 接口数据输入修改为自定义 16 维数据。



```
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.
```

同时将inCopterData (25~32) 结合8维0数组发送给ExtToUE4输出接口在UE界面进行实时显示。

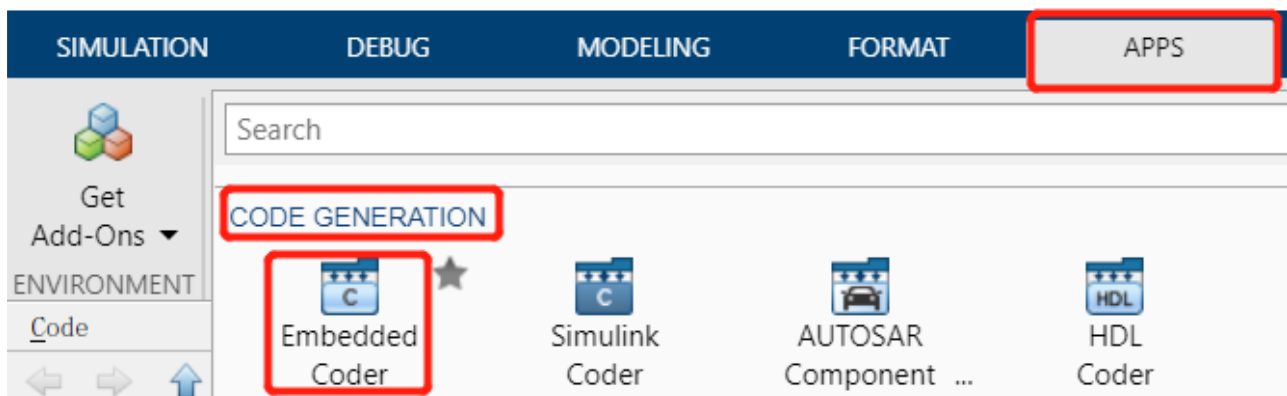


将模型保存后点击编译命令。对于MATLAB 2019a及之前版本，工具栏样式见下图，直接点击它的编译按钮“Build”即可。

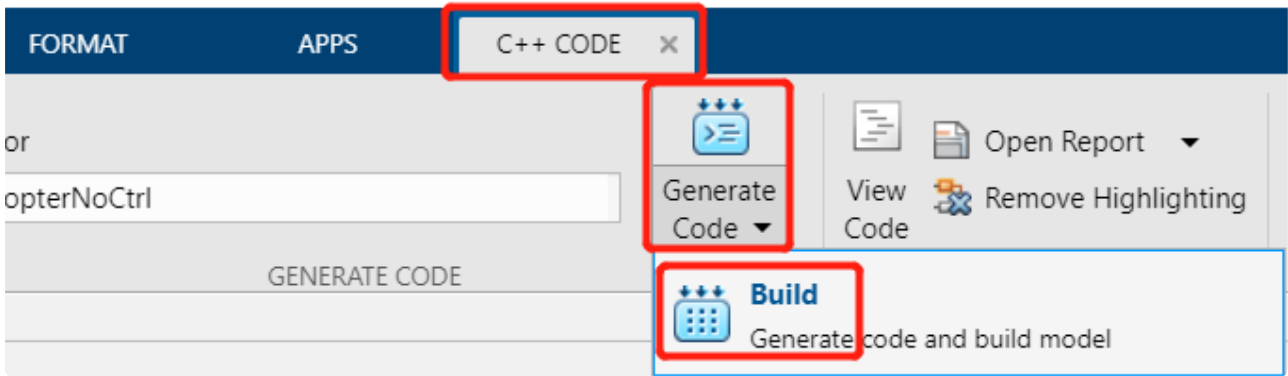


对于2019b及之后版本，点击APPS - CODE GENERATION - Embedded Coder才能弹出代码生成工具栏，在其中如下图所示点击“C++CODE” - “Generate Code” - “Build”按钮就能编译生成代码。

MulticopterCtrlVelocity/Force and Moment Model - Simulink



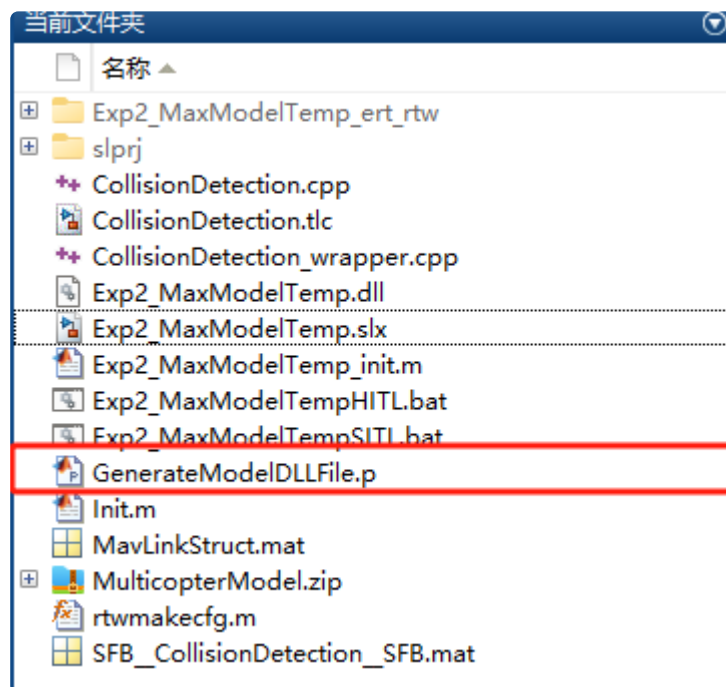
k



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 5: 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，得到最大模型动态链接库 Exp2_MaxModelTemp.dll。



Step 6: 启动仿真

右键以管理员身份运行 Exp2_MaxModelTempHITL.bat，输入飞控和电脑连接的端号，回车启动硬件在环仿真。

```
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_
```

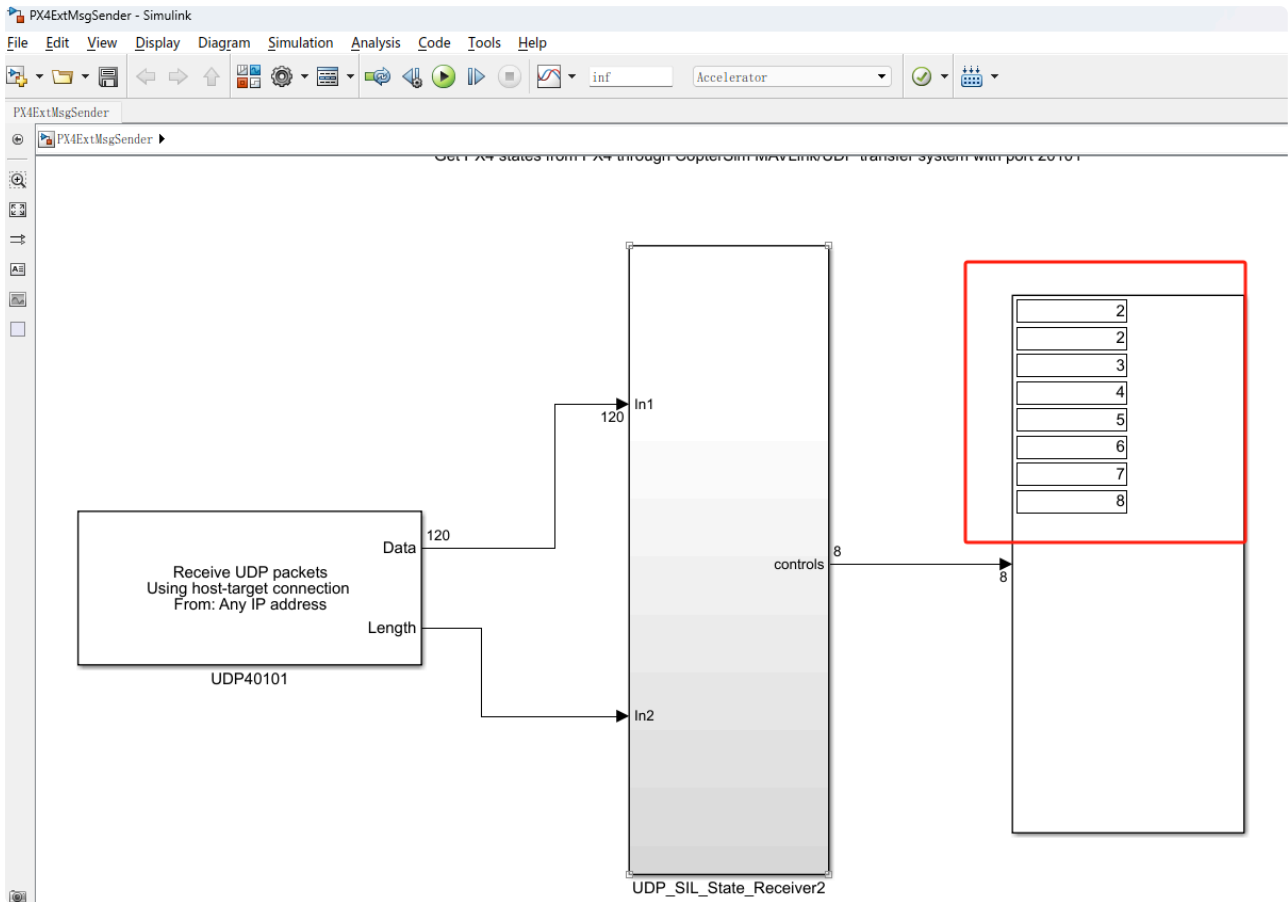
Step 7: 在UE界面查看结果

在UE界面按D可以显示包括ExtToUE4的16维输出在内的载具状态信息，可以在前8维看到inCopterData（25~32）监听的rfly_px4消息。



Step 8: 在Simulink查看结果

返回上一级文件夹，以 MATLAB 打开 PX4ExtMsgSender.slx，并运行。



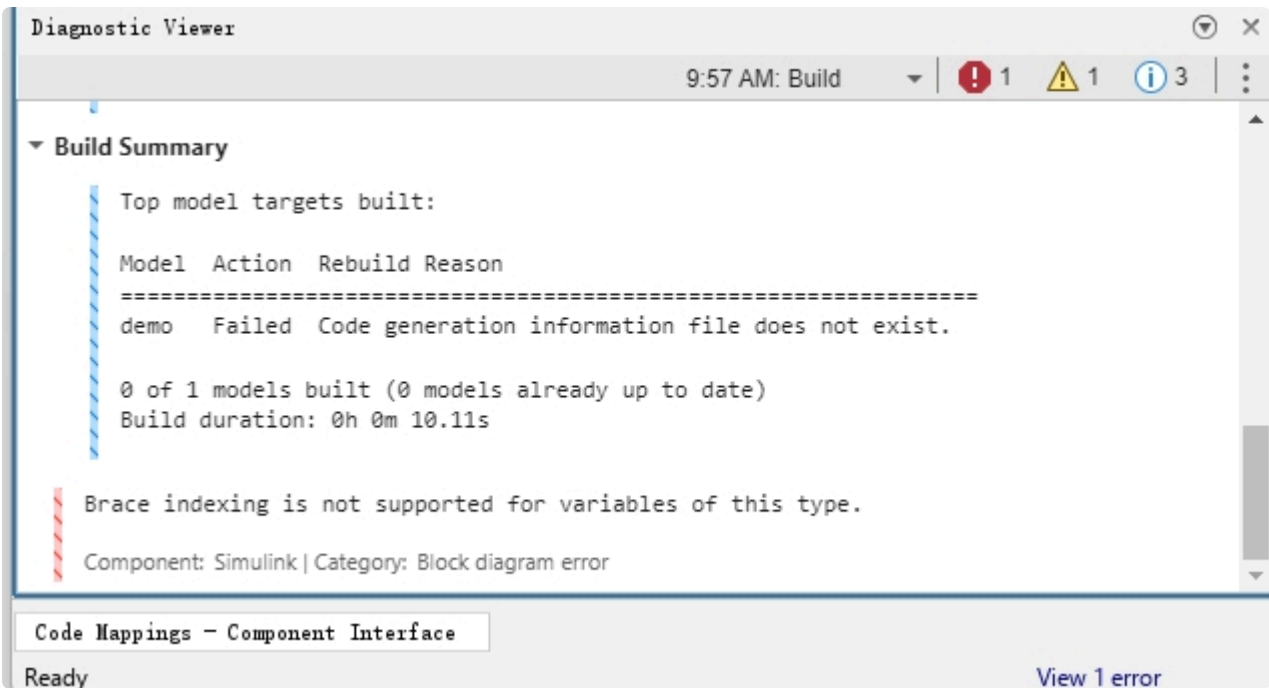
运行后，可以在 UDP_SIL_State_Receiver2 模块的 Display 实时观察到 rfly_px4 消息，此消息与 inCopterData (25~32) 监听的完全一致。

6. 参考资料

1. DLL/SO模型与通信接口 [..\..\PX4PSP\RflySimAPIs\4.RflySimModel\API.pdf](#)
- 2.

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