

1. 实验名称及目的

外部通信实验之获取平台 rfly_px4 uORB 消息：当订阅了 rfly_px4 uORB 消息，并使用平台最大模板进行硬件在环仿真时，可以通过监听 UDP40101 系列端口接收 rfly_px4 消息。

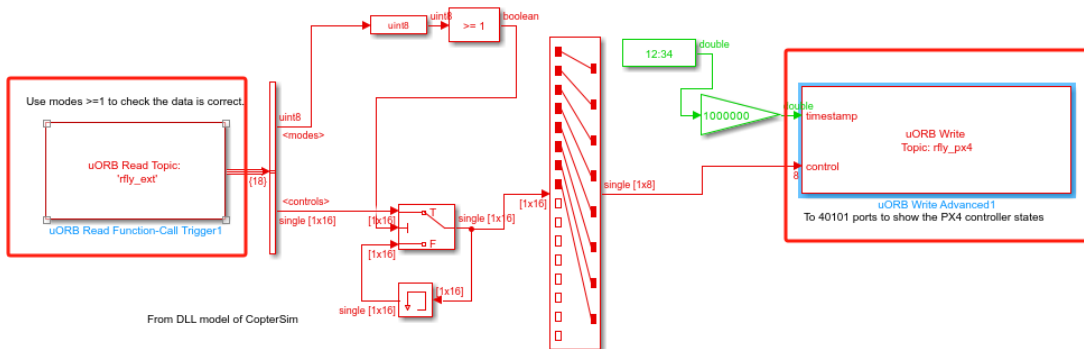
2. 实验原理

2.1. uORB 消息模块[1][2]

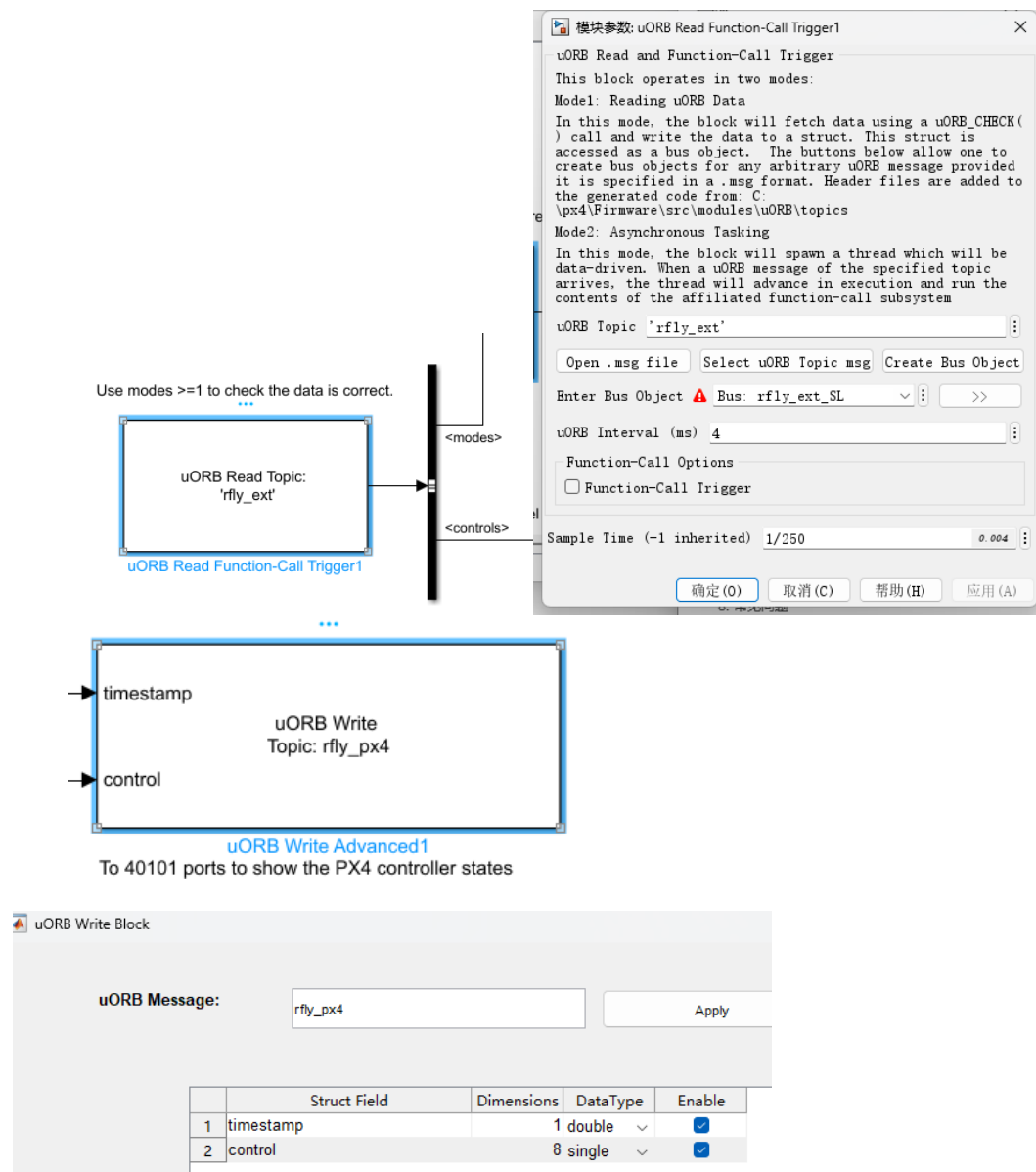
在 Simulink 中，uORB 是一个为了通信系统而设计的发布-订阅模型，提供了一种简单的方式去实现双方之间的消息传递。其中，uORB Read Function-Call Trigger 数据监听接口可以通过调用外部函数的形式响应接收到的新消息；uORB Write 数据发布接口允许用户向 uORB 主题发布指定的值或结构体；uORB Write Advanced 高级数据发布接口则可以允许用户对其发布的数据进行更灵活的控制。

在 Simulink 中使用上述 uorb 收发接口，其参数配置包括消息类型、话题名称和消息数据。消息类型是指要写入的消息的类型，可以在 PX4 源代码中的“msg”目录下找到相应的消息类型定义文件。话题名称（消息 ID）是指要写入的消息所对应的话题名称，可以在 PX4 源代码中的“topics”目录下找到相应的话题定义文件。消息数据是指要写入的实际数据，可以在 Simulink 模型中的其他模块中生成。RflySim 在运行一键安装脚本时，平台会对 Firmware 目录下的源码进行修改，在其中增加了 4 个 uORB 消息（rfly_ctrl.msg、rfly_ext.msg、rfly_px4.msg、rfly_insils.msg），并在*PX4PSP\Firmware\msg\CMakeLists.txt 中已进行注册。

如下图，在 PX4ExtMsgReceiver.slx 中定义了对应的消息收发模块



这些模块只能在 PX4 飞控系统中使用，因此在使用之前需要确保已经正确地配置了 PX4 的开发环境，并且已经将 Simulink 模型成功地部署到 PX4 上；且在使用“uORB Read Function-Call Trigger”模块前，需要先在 PX4 中定义相应的 uORB 消息。在 PX4ExtMsgReceiver.slx 中，首先使用“uORB Read Function-Call Trigger1”模块订阅了 rfly_ext.msg 消息，最后将对应的消息数据在“uORB Write Advanced1”模块中通过 rfly_px4 消息发布。因此需要将 PX4ExtMsgReceiver.slx 编译生成的固件烧录到飞控中，且需要从 DLL 模型直接将消息数据传输进 PX4 内部的 rfly_ext.msg



2.2. 硬件在环仿真（HIL）的实现[3][4]

从实现机制的角度分析,可将 RflySim 平台分为运动仿真模型、底层控制器、三维引擎、外部控制四部分。

- **运动仿真模型**: 这是模拟飞行器运动的核心部分。在 RflySim 平台中,运动仿真模型是通过 MATLAB/Simulink 开发的,然后通过自动生成的 C++代码转化成 DLL (动态链接库) 文件。在使用 RflySim 平台进行软硬件在环仿真时,会将 DLL 模型导入到 CopterSim,形成运动仿真模型。这个模型在仿真中负责生成飞行器的运动响应,它拥有多个输入输出接口与底层控制器、三维引擎、地面控制站和外部控制进行数据交互,具体数据链路、通信协议及通信端口号见 [API.pdf 中的通信接口部分](#)。
- **底层控制器**: 在硬件在环仿真(HIL)中,真实的飞行控制硬件(如 PX4 飞行控制

器）被集成到一个虚拟的飞行环境中。在在硬件在环仿真（HIL）中，底层控制器（将 PX4 固件在真实的飞行控制器，即飞控硬件上运行）通过串口通信与运动仿真模型进行数据交互。

- 三维引擎：这部分负责生成和处理仿真的视觉效果，提供仿真环境的三维视图，使用户能够视觉上跟踪和分析飞行器的运动。
- 外部控制（offboard）：从仿真系统外部对飞行器进行的控制，包括自动飞行路径规划、远程控制指令等。在平台例程中主要通过地面控制站（QGC）、MATLAB 和 Python 调用对应接口实现

在本例程中，主要验证运动仿真模型中利用 ExtToPX4（自定义 uORB 数据输出接口）定义的 uorb 消息数据通过 UDP 40101++2 端口发送到飞控内部的 uORB 消息 rfly_extfly_ext,

2.3. UDP 通信模式

UDP（用户数据报协议）是一种无连接的网络协议，允许快速的数据传输，但不保证数据包的顺序或完整性。在仿真中，UDP 通信通常用于实时数据传输，如飞行器的状态信息（这里主要是顶层模型解算出的位姿信息）。

RflySim 平台的不同通信模式主要分为如下几种。

UDP_Full: Python 传输完整的 UDP 数据给 CopterSim，传输数据量小；CopterSim 收到数据后，再转换为 Mavlink 后传输给 PX4 飞控；适合中小规模集群（数量小于 10）仿真。

UDP_Simple: 数据包大小与发送频率比 UDP_Full 模式小；适合大规模集群仿真，无人机数量小于 100。

Mavlink_Full: Python 直接发送 MAVLink 消息给 CopterSim，再转发给 PX4，数据量较大适合单机控制；适合单机或少量飞机仿真，无人机数量小于 4；

Mavlink_Simple: 屏蔽部分 MAVLink 消息包，并降低数据频率，发送数据量比 MAVLink_Full 小很多，适合多机集群控制；适合小规模集群仿真，无人机数量小于 8。

Mavlink_NoSend: 模式下 CopterSim 不会向外发送 MAVLink 数据，此模式需要配合硬件在环仿真+数传串口通信，通过有线方式传输 MAVLink，此模式局域网内数据量最小，适合分布式视觉硬件在环仿真，无人机数量不限制

Mavlink_NoGPS: 模式下 CopterSim 不会向外发送 MAVLink 数据和 GPS 数据。

2.4. 获取平台 rfly_px4 uORB 消息

在进行硬件在环仿真时，需要向底层控制器中发送数据（传感器数据、故障触发、控制指令、参数调整等），同时接收一些感兴趣的数据。RflySim 平台提供了 rfly_ctrl 这一 uORB 消息来接收外部数据（UDP 发送指定结构体到 CopterSim 的 30100 系列端口），同时提供 rfly_px4 这一 uORB 消息来向外发送数据（向 40100 系列端口发送特定数据）。本例程以外部发送的 rfly_ext 数据来作为输入，同时会将收到的数据向 rfly_px4 发送出去，回传给外部程序。

3. 实验效果

以最大模板启动硬件在环仿真，待仿真初始化完成（GPS 3D fixed）后，监听 UDP40101 系列端口接收 rfly_px4 消息。

4. 文件目录

文件夹/文件名称		说明
DLLModelTemp	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_MaxModelTempHIT	最大模型硬件在环仿真启动脚本
	L.bat	
PX4ExtMsgReceiver.slx	\	rfly_px4 uORB 消息文件
PX4ExtMsgSender.slx	\	外部通信接口程序

5. 运行环境

序号	软件要求	硬件要求	
		名称	数量
1	Windows 10 及以上版本	笔记本/台式电脑 ^①	1
2	RflySim 平台收费版	Pixhawk6C ^②	1
3	MATLAB 2017B 及以上 ^③	数据线	1

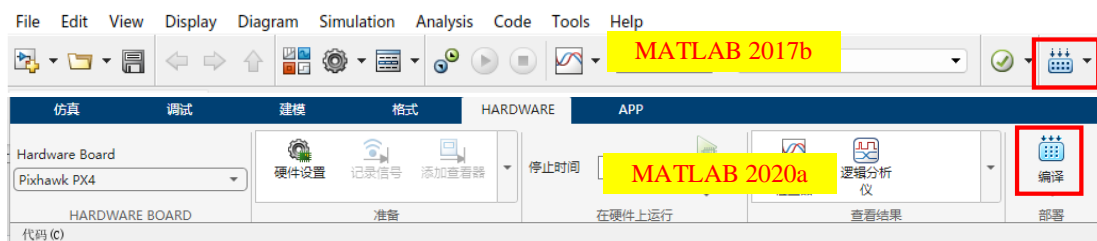
①：推荐配置请见：<https://doc.rflysim.com>

②：平台安装时的推荐编译命令为：px4_fmu-v6c_default，PX4 固件版本为：1.13.3。其他配套飞控请见：<http://doc.rflysim.com/hardware.html>。

6. 实验步骤

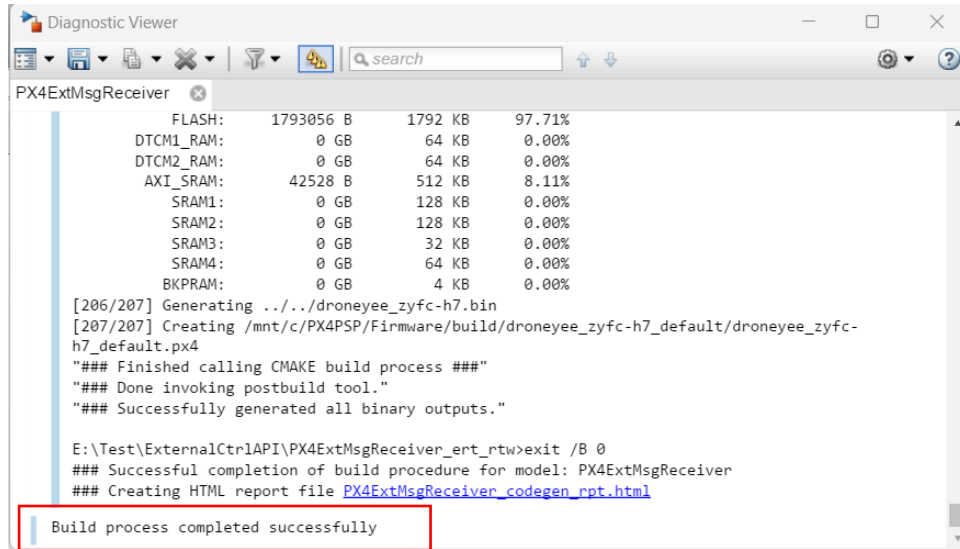
Step 1:

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



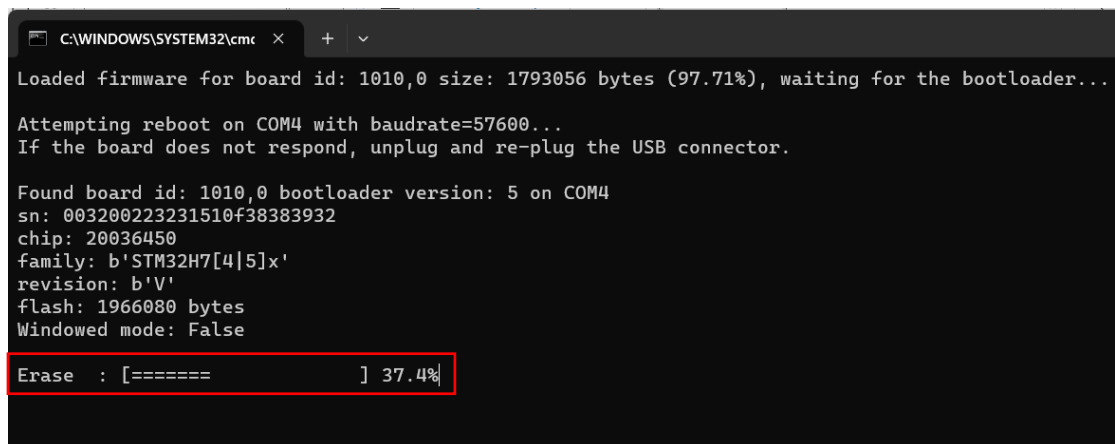
Step 2:

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。

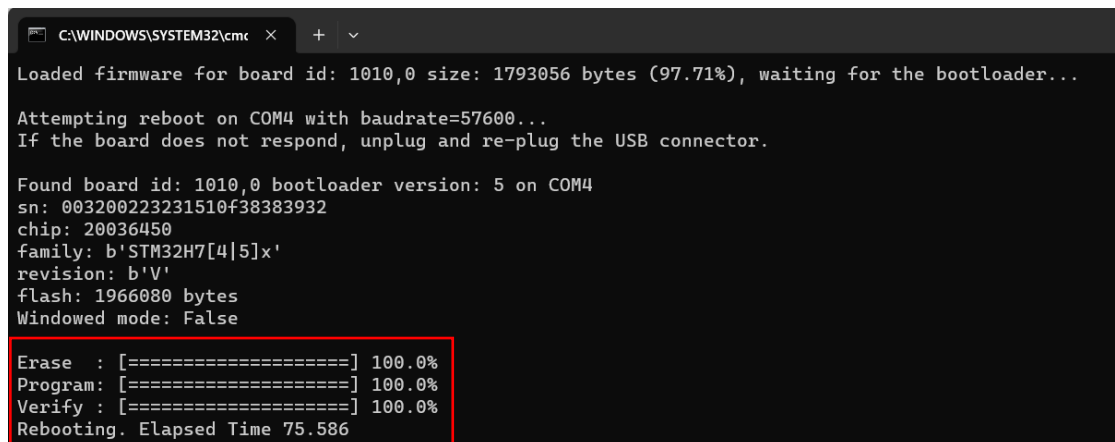


Step 3:

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。



Step 4:

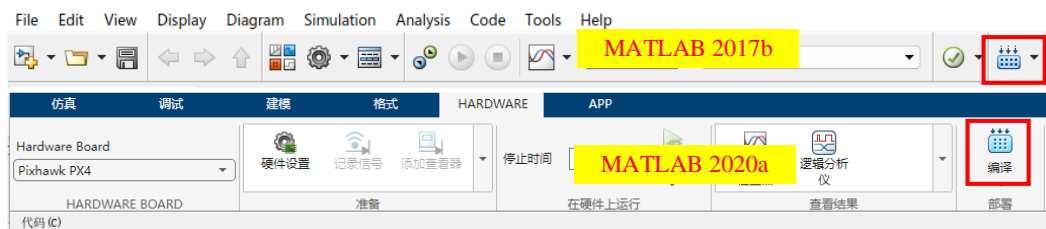
打开 DLLModelTemp 文件夹，在 MATLAB 中打开 Exp2_MaxModelTemp.slx 文件，将 simulink 文件中的 ExtToPx4 接口数据输入修改为自定义 16 维数据。



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.

修改完成后，在 Simulink 中，点击编译命令。

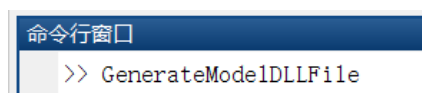


在 Simulink 的下方点击 View diagnostics 指令，即可弹出诊断对话框，可查看编译过程。在诊断框中弹出 Build process completed successfully，即表示编译成功。

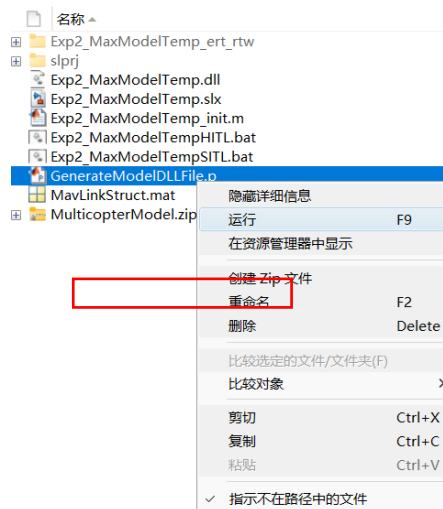


Step 5:

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。



或



Step 6:

右键以管理员身份运行 Exp2_MaxModelTempHITL.bat, 输入飞控和电脑连接的端口号, 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

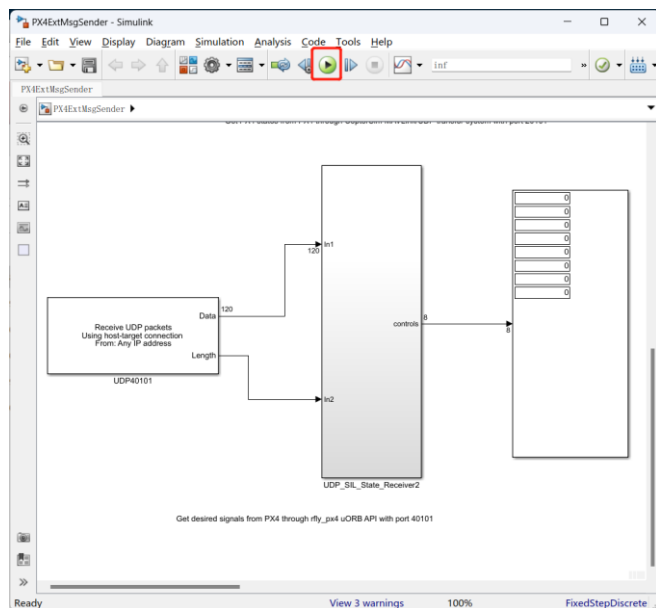
Available COM ports on this computer are:
COM4: USB ???

Recommended COM list input is: 4

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

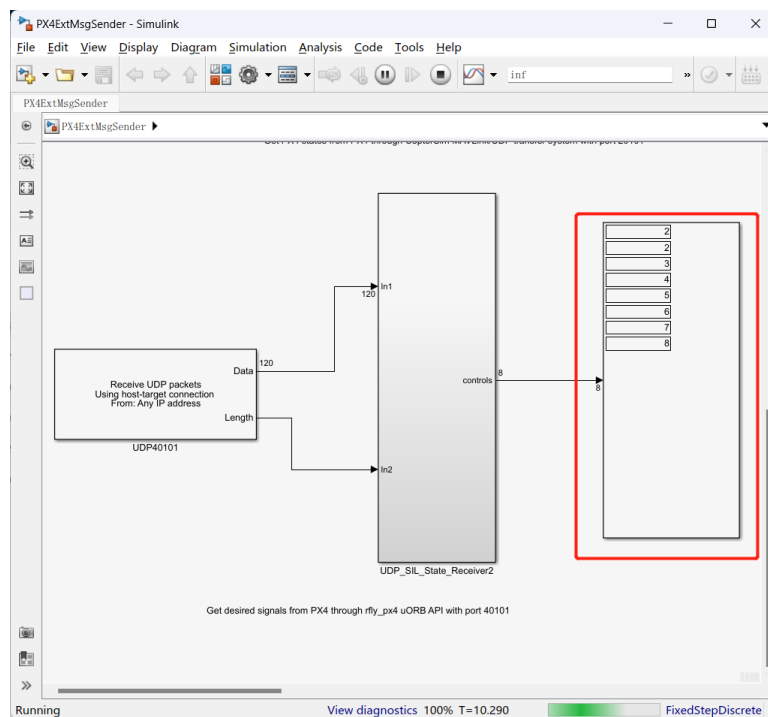
Step 7

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



Step 8

正常现象为，PX4ExtMsgSender.slx 运行后，可以在 UDP_SIL_State_Receiver2 模块的 Display 实时观察到 rfly_px4 消息，该消息为 8 维。



7. 参考资料

- [1]. 自动代码生成外部通信接口 [..\..\..\5.RflySimFlyCtrl\API.pdf](#)
- [2]. uORB Read and Write—uORB 消息读取和写入库 [..\..\..\5.RflySimFlyCtrl\API.pdf](#)
- [3]. DLL/SO 模型与通信接口 [..\..\..\API.pdf](#)
- [4]. 外部控制接口 [..\..\..\API.pdf](#)

[5].

8. 常见问题

Q1.

A1.