
1. 实验名称及目的

ExtToPX4 接口验证: 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。

2. 实验原理

2.1. 软/硬件在环仿真 (SIL/HIL) 的实现 [2][3]

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:** 在软/硬件在环仿真 (SIL/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。在软件在环仿真 (SIL) 中, 底层控制器 (通过 wsl 上的 PX4 仿真环境运行) 通过网络通信与运动仿真模型交互数据。在硬件在环仿真 (HIL) 中, 它 (将 PX4 固件在真实的飞行控制器 (即飞控) 硬件上运行) 则通过串口通信与运动仿真模型进行数据交互。飞控与 CopterSim 通过串口 (硬件在环 HITL) 或网络 TCP/UDP (软件在环 SITL) 进行连接, 使用 MAVLink 进行数据传输, 实现控制闭环。
- **三维引擎:** 这部分负责生成和处理仿真的视觉效果, 提供仿真环境和模型的三维视图, 使用户能够视觉上跟踪和分析飞行器的运动。CopterSim 发送飞机位姿、电机数据到三维引擎, 实现可视化展示。
- **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。

2.2. 飞控中通过 `rfly_ext` (uORB 消息) 获取 ExtToPX4 接口数据[1]

CopterSim 在进行 DLL 模型加载时, 会将 ExtToUE4PX4 的 32 维数据, 拆分成 16 维的 ExtToUE4 和 16 维的 ExtToPX4 结构体, 会再将其转发为 “`hil_actuator_controls`” (https://mavlink.io/en/messages/common.html#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`

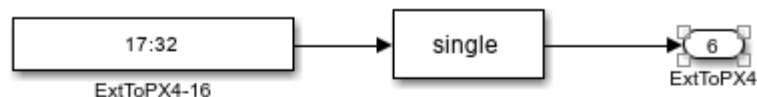
_receiver.cpp 飞控固件源码，增加了对 hil_actuator_controls 消息的支持。

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_ext 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 > 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 rlfy_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.

3. 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。

4. 文件目录

文件夹/文件名称	说明
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

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

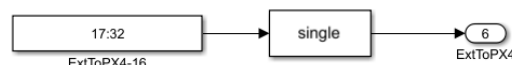
①：推荐配置请见：<https://doc.rflysim.com>

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 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。



```
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.
```

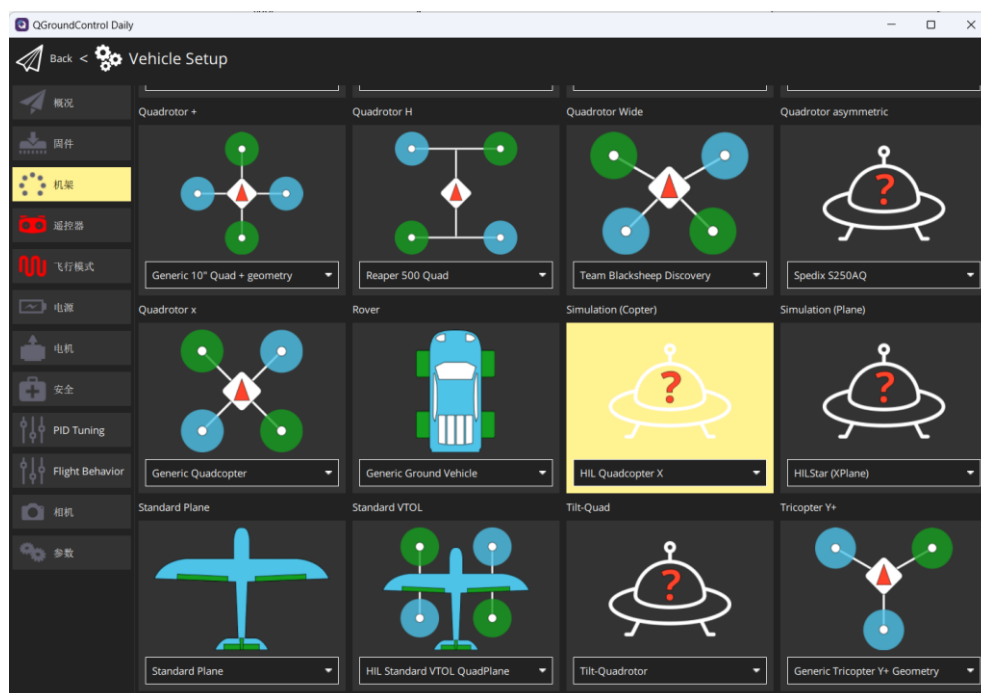
Step 2:

将飞控通过 usb-typeC 线和电脑 usb 口连接，



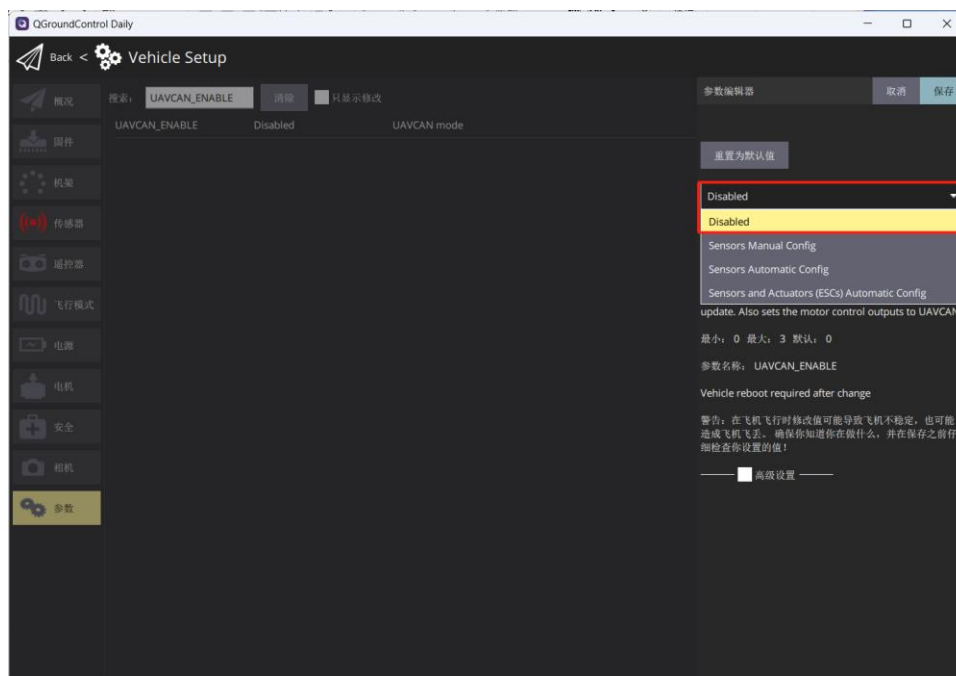
Step 3:

推荐使用 Pixhawk 6C 飞控进行硬件在环仿真，将飞控烧录至 1.13.3 固件版本，机架设置为“HIL Quadcopter X”，点击 QGC 右上角的“应用并重启”。



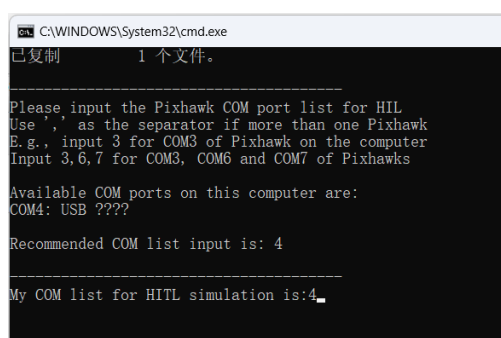
Step 4:

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。



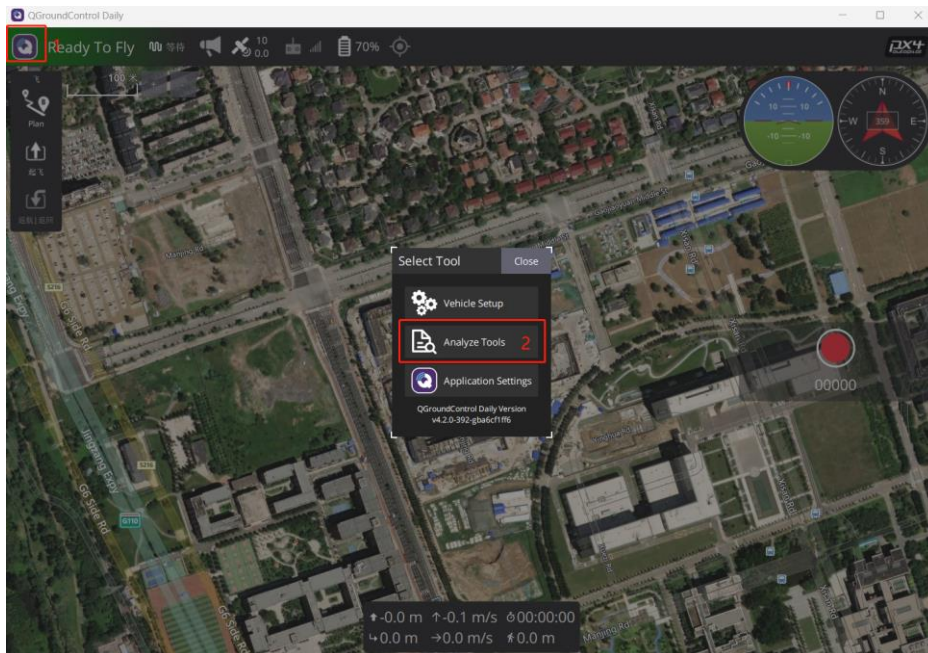
Step 5:

右键点击 Exp2_MaxModelTempHITL.bat 并以管理员身份运行，输入 4（飞控对应端口号），启动硬件在环仿真。



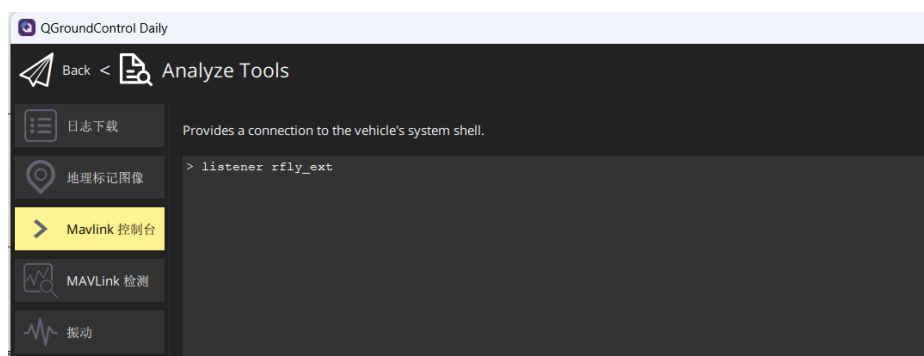
Step 6:

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



Step 7:

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:
  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> |
```

7. 参考资料

- [1]. 自动代码生成外部通信接口 <..\..\5.RflySimFlyCtrl\API.pdf>。
- [2]. DLL/SO 模型与通信接口 <..\..\API.pdf>
- [3]. 外部控制接口 <..\..\API.pdf>
- [4].

8. 常见问题

- Q1.
- A1.