飞控板载应用开发实验

1. 实验目的

了解PX4源码中如何创建一个APP，并进行编译与调试，最后设置开机启动。

2. 实验要求

- 软件要求：Win10/Win11系统；RflySim工具链；MATLAB 2022B及以上[1]。
- 硬件要求：笔记本/台式电脑① 1台；Pixhawk 6x飞控 1台；数据线、杜邦线等 1台[2]。

3.实验地址

例程目录：[安装目录]\RflySimAPIs\2.RflySimUsage\1.BasicExps\e12_PX4-App

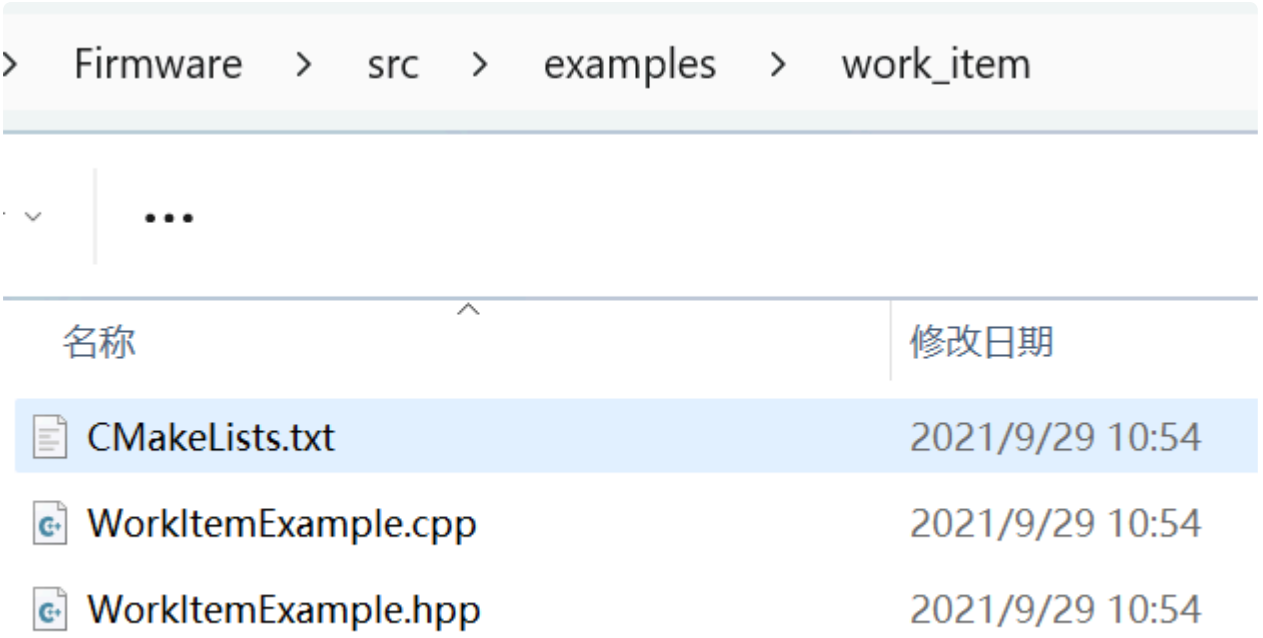
- my_app/CMakeLists.txt：应用程序的CMake文件。
- my_app/Kconfig：PX4配置文件。
- my_app/my_app.c：应用程序主文件。

4. 实验内容或步骤

4.1 步骤1：模版例程+SITL测试（必做）

进入"[安装目录]\Firmware\src\examples"目录，找到work_item文件夹，阅读源码，并参考如下链接了解实现原理。

https://docs.px4.io/main/en/modules/module_template.html#work-queue-task



注：本例程主要是基于ScheduledWorkItem创建了一个实时控制程序（固定周期调度）

在init()函数中，使用ScheduleOnInterval(5000_us)接口，就能创建一个200Hz固定频率的一个实时控制器。

```
48  bool WorkItemExample::init()
49  {
50      // execute Run() on every sensor_accel publication
51      if (!_sensor_accel_sub.registerCallback()) {
52          PX4_ERR("sensor_accel callback registration failed");
53          return false;
54      }
55
56      // alternatively, Run on fixed interval
57      // ScheduleOnInterval(5000_us); // 2000 us interval, 200 H
58
59      return true;
60  }
```

如果要循环调度，就使用取消注释本函数

然后在Run()函数中编写自己的控制器即可，会被按照上述预设频率调度。

```
62  void WorkItemExample::Run()
63  {
64      if (should_exit()) {
65          ScheduleClear();
66          exit_and_cleanup();
67          return;
68      }
69
70      perf_begin(_loop_perf);
71      perf_count(_loop_interval_perf);
72
```

注：Run函数默认调度一次，如果设置了ScheduleOnInterval则会循环调度。注：Run里面的计算复杂度不能太大，不能超过调度的时间间隔，否则会破坏实时性，影响系统稳定。

注：如果设置了控制器频率，那么在Run里面，可以周期性检查uORB消息是否更新，并按时序更新自己的控制算法，这就完成了自己的算法的集成工作。

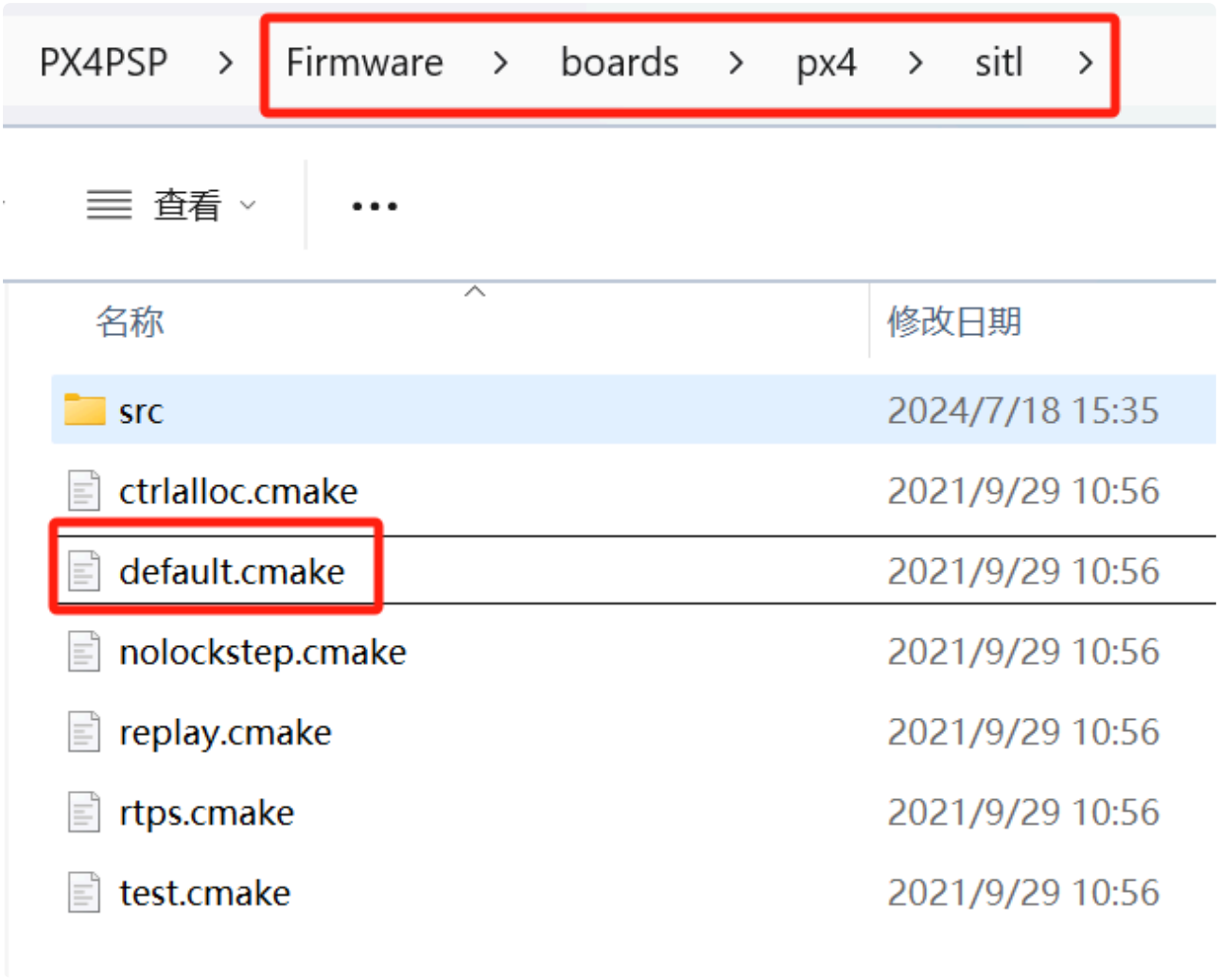
4.2 步骤2：查看CMakeLists.txt文件

查看CMakeLists.txt文件。可以了解到这个APP的模块名为：work_item（所属于examples），用于.cmake文件的编写。这个APP的名字为：work_item_example，用于后续pxh/nsh/rcS中启动程序。

```
34  px4_add_module(
35      MODULE examples__work_item
36      MAIN work_item_example
37      COMPILE_FLAGS
38          #-DDEBUG_BUILD   # uncomment for PX4_DEBUG output
39          #-O0             # uncomment when debugging
40      SRCS
41          WorkItemExample.cpp
42          WorkItemExample.hpp
43      DEPENDS
44          px4_work_queue
45      )
```

4.3 步骤3：检查应用程序是否已编译到固件中

在"[安装目录]\Firmware\boards\px4\sitl"找到px4_sitl_default的cmake文件，并查看work_item应用程序是否已经处于编译列表中。针对不同版本的固件，该cmake文件的位置有所不同。



对于1.13及之前的版本为："[安装目录]\Firmware\boards\px4\sitl\default.cmake"。在该文件的EXAMPLES中应该有work_item项。

```
87          work_queue
88      EXAMPLES
89          dyn_hello # dynamically loading modules example
90          fake_gps
91          fake_imu
92          fake_magnetometer
93          fixedwing_control # Tutorial code from https://px4.io/dev
94          hello
95          #hwtest # Hardware test
96          #matlab_csv_serial
97          px4_mavlink_debug # Tutorial code from http://dev.px4.io/
98          px4_simple_app # Tutorial code from http://dev.px4.io/en/
99          rover_steering_control # Rover example app
100         uuv_example_app
101         work_item
102     )
103
```

对于1.14及之后的版本为："[安装目录]\Firmware\boards\px4\sitl\default.px4board"。在该文件中有CONFIG_EXAMPLES_WORK_ITEM=y项。

```
71  CONFIG_EXAMPLES_HELLO=y
72  CONFIG_EXAMPLES_PX4_MAVLINK_DEBUG=y
73  CONFIG_EXAMPLES_PX4_SIMPLE_APP=y
74  CONFIG_EXAMPLES_WORK_ITEM=y
75
```

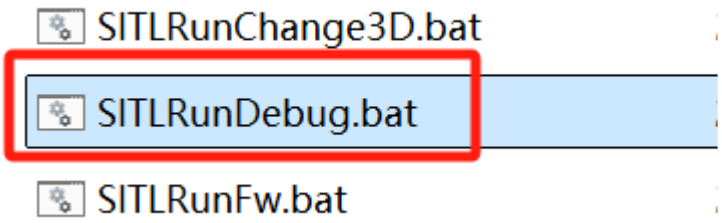
如果存在上述项，说明应用程序已经在编译列表中。

注：如果不在列表中，要在合适位置模仿上述语法，增加一个条目。

4.4 步骤4：开始软件在环仿真

方案一：自动方式

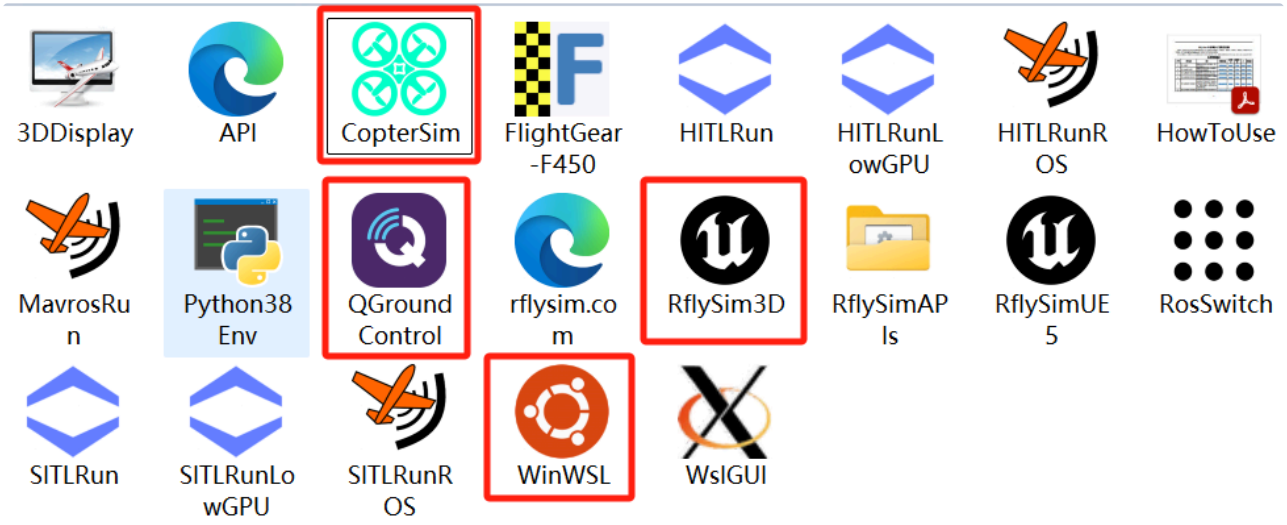
双击*[安装目录]\RflySimAPIs\SITLRunDebug.bat"，会自动进入PX4_SITL模式，并开启pxh终端功能。



方案二：手动方式

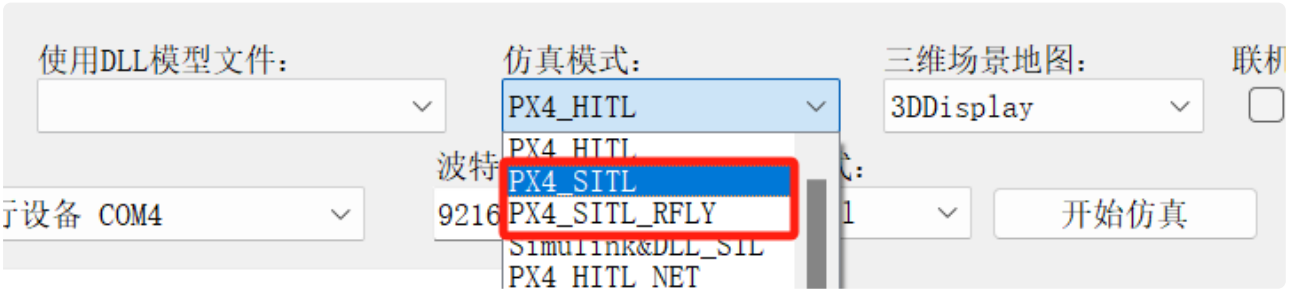
1）打开"桌面\RflyTools\WinWSL"并输入make px4_sitl none，开启软件在环仿真。

2）手动打开RflySim3D（低配电脑请打开3DDisplay），CopterSim和QGroundControl，并在CopterSim仿真模式中选择"PX4_SITL"，会开始SITL仿真并开启pxh终端功能。



注：PX4的SITL仿真通信规则，可以参考 https://docs.px4.io/main/en/simulation。

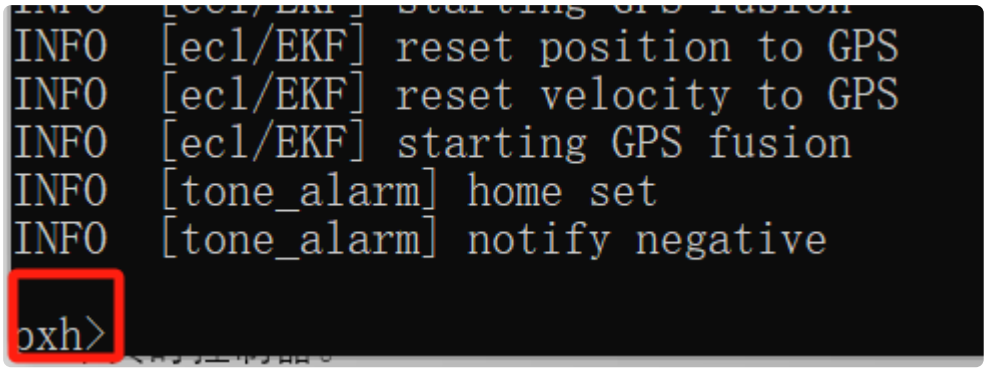
注：从网页可以看出PX4原生的SITL端口规则（对应CopterSim的PX4_SITL模式），最多支持10个飞机，就会发生端口冲突，因此CopterSim还有一种PX4_SITL_RFLY模式，用于兼容大规模集群仿真。



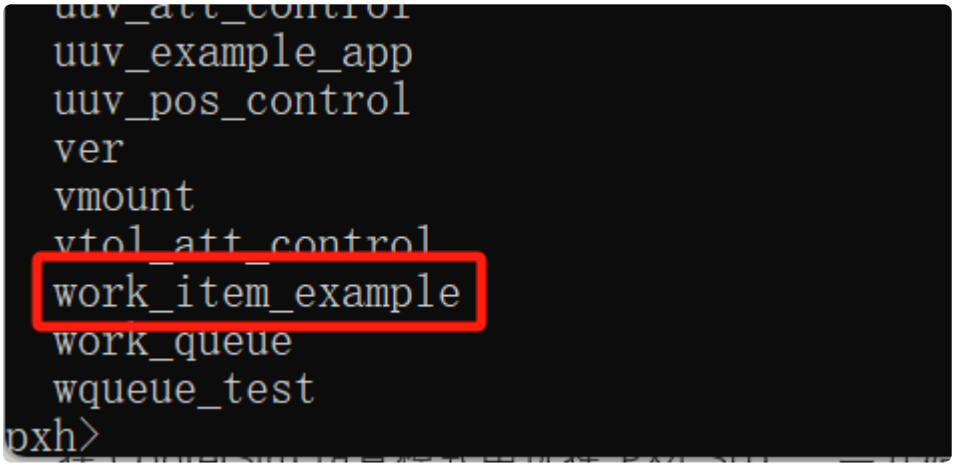
注：SITLRunDebug.bat只适合单机仿真。SITLRun.bat因为要适用于多机，默认不会开启pxh输入功能；此外SITLRun.bat使用的是PX4_SITL_RFLY模式，它的端口和上述PX4原生端口定义有所区别。

4.5 步骤5：自编程序的调试

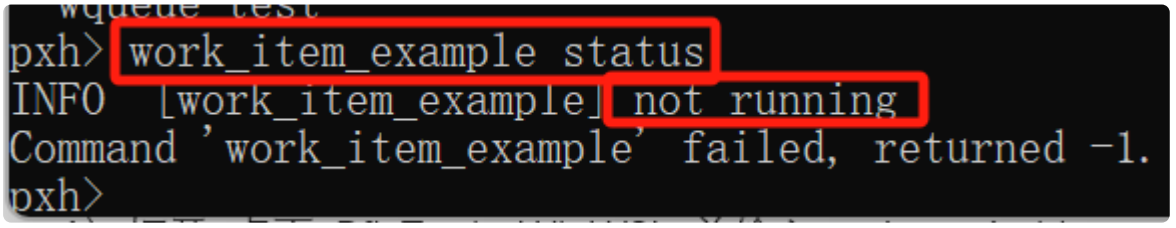
在弹出WinWSL窗口中，按一下回车，就能看到"pxh>"前缀，可以在其中输入指令。



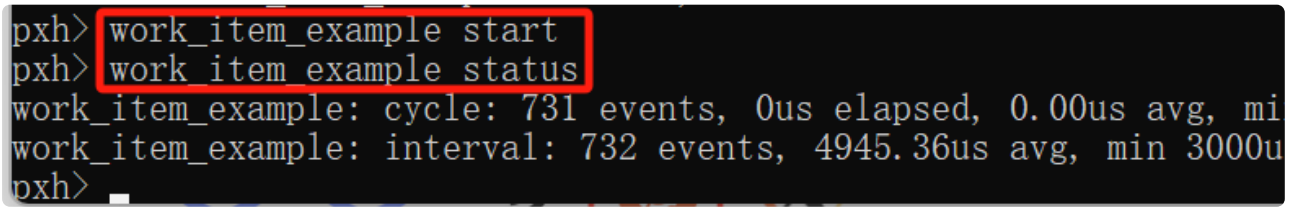
在其中输入"help"，可以查看到我们的work_item_example应用程序。



再输入"work_item_example status"，可以发现我们的程序没有自动启动。



输入"work_item_example start"，可以将程序手动启动，然后再运行命令"work_item_example status"，再次查看程序运行状态。



可以发现能收到正确的打印信息。

4.6 步骤6：自编程序的开机启动

关闭前面打开的所有仿真程序和WinWSL的命令窗口。

用文本编辑器打开SITL的启动脚本文件"*:\PX4PSP\Firmware\ROMFS\px4fmu_common\init.d-posix\rcS"，并在其最后，加入启动命令"work_item_example start"。

```
277    mavlink boot_complete
278    replay trystart
279
280    work_item_example start
```

再次双击运行[安装目录]\RflySimAPIs\SITLRunDebug.bat，并输入work_item_example status，可以发现，我们编写的程序已经默认开机启动状态。

```
INFO  [tone_alarm] notify negative

pxh>
pxh>
pxh> work_item_example status
work_item_example: cycle: 6103 events, 0us elapsed, 0.00us avg, min 0us max 0us 0.000us rms
work_item_example: interval: 6103 events, 5178.93us avg, min 1000us max 33000us 1866.214us rms
pxh>
```

每次实验结束后，如果不是需要开机自启动，将在启动脚本文件中新添加的启动命令删除。

5. 关键知识点

关键知识点1：

PX4构架以及如何编写应用程序。访问如下网址，自学了解PX4自定义代码开发的一些基本原理。

https://docs.px4.io/main/en/modules/module_template.html

关键知识点2：

软件在环SITL仿真的pxh和硬件在环HITL仿真的nsh，是PX4实时和底层操作系统交互的接口，可以参考如下链接。

https://docs.px4.io/main/en/debug/consoles.html#using_the_console

关键知识点3：

编写的应用程序，要去"[安装目录]\Firmware\boards"目录下找到对应的cmake文件（1.14开始是.px4board后缀），并将其名字写入，才能正常编译到固件中。

关键知识点4：

编写的应用程序，烧录到固件后，默认不会随飞控开机启动，需要去pxh或nsh中手动开启，或者写入到如下启动脚本中。

SITL仿真：[安装目录]\Firmware\ROMFS\px4fmu_common\init.d-posix\rcS

HITL仿真和真机：[安装目录]\Firmware\ROMFS\px4fmu_common\init.d\rcS

6.参考资料

1. https://docs.px4.io/main/en/modules/module_template.html
2. https://docs.px4.io/main/en/debug/consoles.html#using_the_console
3. https://docs.px4.io/main/en/modules/module_template.html#work-queue-task

7.常见问题

Q1：如何使应用程序随飞控开机自启动?

A1：在SITL仿真中，需要修改启动脚本文件"[:PX4PSP\Firmware\ROMFS\px4fmu_common\init.d-posix\rcS](file:///f%3A/PX4PSP/Firmware/ROMFS/px4fmu_common/init.d-posix/rcS)"，在最后添加启动命令；在HITL仿真和真机中，需要修改"[安装目录]\Firmware\ROMFS\px4fmu_common\init.d\rcS"文件。

Q2：应用程序无法正常编译到固件中是什么原因?

A2：需要检查应用程序是否添加到了对应的cmake文件中（1.13及之前版本在default.cmake中，1.14及之后版本在default.px4board中）。

Q3：SITL仿真和HITL仿真的主要区别是什么?

A3：SITL（Software In The Loop）是纯软件仿真，运行在主机上；HITL（Hardware In The Loop）是硬件在环仿真，使用真实的飞控硬件参与仿真过程。

---

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 ↩
2. 推荐配置请见：https://rflysim.com/doc/zh/HowToInstall.pdf ↩