

1、实验名称及目的

平台固件生成介绍：通过该例程，使用户理解如何配置平台一键安装脚本、如何使用平台完成底层自主开发控制器固件生成和原生固件生成。

2、实验原理

该实验通过使用 RflySim 平台的一键安装脚本，可实现将无人系统开发过程中所运用到的各种软件及驱动等程序一键部署到本地电脑上；另外，MATLAB/Simulink 的 Embedded Coder 模块可快速将 Simulink 中所搭建的模型生成对应的 C/C++代码；同时，基于 PX4 软件的源码开放环境，可将固件上传到对应的硬件中。

3、实验效果

通过 MATLAB/Simulink 编译自主开发的底层控制器后，可以得到对应的固件文件；以特定配置重新安装平台，可以得到原生固件。

4、文件目录

文件夹/文件名称		说明
icon	Init.m	模型初始化参数文件。
	MavLinkStruct.mat	MAVLink 结构体数据文件。
	pixhawk.png	Pixhawk 硬件图片。
	readme.pdf	机架类型修改说明文件。
	UE_Logo.jpg	RflySim3D 软件图片。
	F450.png	F450 飞机模型图片。
Init_control.m		控制器初始化参数文件。
AttitudeSystemCodeGen.slx		底层自主开发控制器源程序。

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 平台免费版	\	\
3	MATLAB 2017B 及以上	\	\

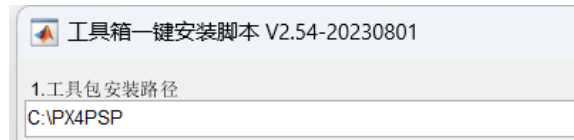
①：推荐配置请见：<https://doc.rflysim.com/1.1InstallMethod.html>

6、实验步骤

6.1、平台一键安装脚本配置介绍

1) 工具包安装路径:

本平台的所有依赖文件都会安装在本路径下，大约需要 20G 的空间。默认安装路径是“C:\PX4PSP”，如果 C 盘空间不够可以选择其他盘符下的路径。注意：路径名称必须正确，且只能用纯英文的路径，否则会导致编译失败。



2) PX4 固件编译命令:

主要对应底层控制器开发需求，并使用代码生成功能，需要根据飞控硬件来选择编译命令（注：顶层视觉和集群算法开发用户不需要配置，保持默认即可）。默认为“px4_fmu-v6c_default”对应 Pixhawk 6C。更多飞控编译指令请见：<https://doc.rflysim.com/hardware.html>。

2. PX4 固件编译命令：见 Firmware\boards 目录，模板 px4_fmu-v6x_default、droneyee_racer_default 等
px4_fmu-v6c_default

注：第一次安装完成后，除了重新运行本安装脚本，另一种针对不同的 Pixhawk 硬件板子想更换不同的编译命令（例如换成 px4_fmu-v3_default）的方法，只需要在 MATLAB 中输入命令：PX4CMD('px4_fmu-v3_default') 或者使用命令：PX4CMD px4_fmu-v3_default

```
fx >> PX4CMD px4_fmu-v3_default
```

3) PX4 固件版本:

PX4 源代码每年都会进行更新，目前最新的固件版本为 1.13.3。随着固件版本的升级，功能会逐渐增加，支持的新产品也越多，但是对旧的一些自驾仪硬件的兼容就会变差。本例题推荐使用 Pixhawk V6C，对应的编译指令为“px4_fmu-v6c_default”，选用的固件版本 PX4-1.13.3。

3. PX4 固件版本 (1: PX4-1.7.3, 4: PX4-1.10.2, 5: PX4-1.11.3, 6: PX4-1.12.3, 7: PX4-1.13.3)
7

4) PX4 固件编译器:

由于 PX4 源代码的编译依赖于 Linux 编译环境和相关组件，本平台提供了三套编译环境来实现 Windows 平台下对 Linux 编译环境的模拟，它们分别是：基于 Windows Subsystem for Linux (WSL) 的编译环境 Win10WSL 编译器、基于 Msys2 的 Msys2Toolchain 编译环境和基于 Cygwin 的 CygwinToolchain 编译器。注意，如果需要编译 \geq PX4-1.8 版本以上固件，请选择 CygwinToolchain 编译器；编译 \leq PX4-1.8 版本的固件，可选择 Msys2Toolchain 编译器。基于 Msys2 或 Cygwin 的本地编译器，支持 Windows 7~11 平台，而且部署方便，

但是编译效率较低。对于 Windows10 1809 及以上的系统版本，推荐安装 Win10WSL 编译器，这种方式可以大大加快编译速度，而且兼容所有版本的 PX4 飞控固件。

4. PX4 固件编译器 (1: Win10WSL[通用], 2: Msys2[适用版本≤PX4-1.8], 3: Cygwin[适用≥PX4-1.8])

1

5) 是否全新安装 PSP 工具箱:

如果该选项设置为“是”，会将 PSP 工具箱安装在本地 MATLAB 软件中。如果 PSP 工具箱已经安装过，则会对 PSP 工具箱进行全新安装。如果选择“否”，脚本对 PSP 工具箱不做任何更改（不会卸载掉安装的 PSP 工具箱或其他动作）。

5. 是否重新安装 PSP 工具箱 (是: 重装工具箱, 否: 维持现有安装)

否

6) 是否全新安装其他依赖程序包:

如果该选项设置为“是”，会将 QGC 地面站、CopterSim、3DDisplay 等软件部署在设定的安装路径上，并安装 Pixhawk 硬件的相关驱动程序，以及在桌面生成这些软件的快捷方式。如果安装路径上已经部署过相关依赖软件，选择“是”则会删除旧的安装包并进行全新重新安装。如果该选项设置为“否”则不做任何修改。

6. 是否重新安装其他依赖程序包 (CopterSim、QGC地面站、硬件在环仿真软件等, 约5分钟)

否

7) 是否全新配置固件编译器编译环境:

如果该选项设置为“是”，会将选定的编译器（Win10WSL、CygwinToolchain 或 Msys2Toolchain）部署在设定的安装路径上，如果环境已经存在，则会清空旧的编译环境，进行还原与全新部署。反之，如果该选项设置为“否”则不会进行任何更改。

7. 是否重新配置固件编译器编译环境 (是: 全新安装编译器, 否: 维持原样, 重装约5分钟)

否

8) 是否全新部署 PX4 固件代码:

如果该选项设置为“是”，会将选定的 PX4 Firmware 源代码部署在设定的安装路径上，如果固件存在，会删除旧的固件文件夹，并进行全新部署。如果该选项设置为“否”则不会进行任何更改。

8. 是否重新部署 PX4 固件代码 (是: 全新部署代码, 否: 维持现状, 大约5分钟)

否

9) 是否全新编译固件:

如果该选项设置为“是”，会对部署固件进行预编译，这样可以大大节省后续代码生成与编译的时间，同时可以检测环境安装是否正常。如果该选项设置为“否”则不会进行任何更改。

9.是否预先用选定命令编译固件（是：全新编译固件，否：维持现状，大约5分钟）

否

10) 是否屏蔽 PX4 自身控制器输出：

如果该选项设置为“是”，会对 Firmware 中对电机的控制信号进行屏蔽，防止与生成代码发生冲突（注：本选项不会屏蔽 PX4_SITL 控制器的输出，因此可以正常进行软件在环仿真）。如果选择“否”，则不会对固件输出进行屏蔽，可以用于测试 PX4 自带的控制算法，因此如果要生成官方固件，本选项请选择“否”。

10.是否屏蔽PX4官方控制器输出(使用Simulink控制器选"是", 使用PX4官方控制器选"否")

是

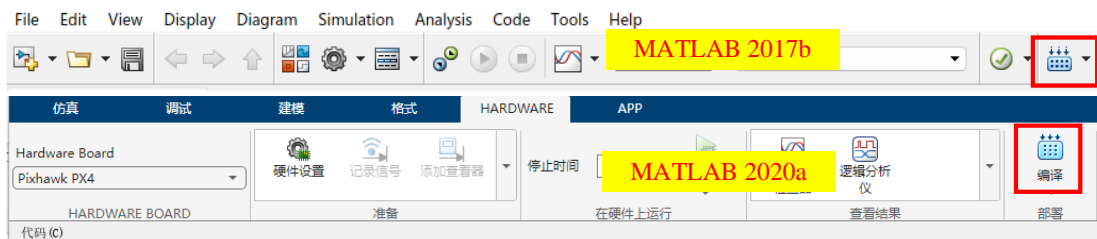
注：在以 MATLAB/Simulink 完成底层控制算法开发、需要使用平台来验证算法时，此处选择“是”；若想使用官方原生固件来进行软硬件在环仿真/验证运动模型，则此处选择“否”。

6.2、底层自主开发控制器固件生成

“底层自主开发控制器”指的是用户基于 MATLAB/Simulink 自主搭建的、用于控制特定无人载具（比如多旋翼、固定翼等）运动的算法，这里以 AttitudeSystemCodeGen.slx 为例，对如何生成底层自主开发控制器固件过程进行说明。

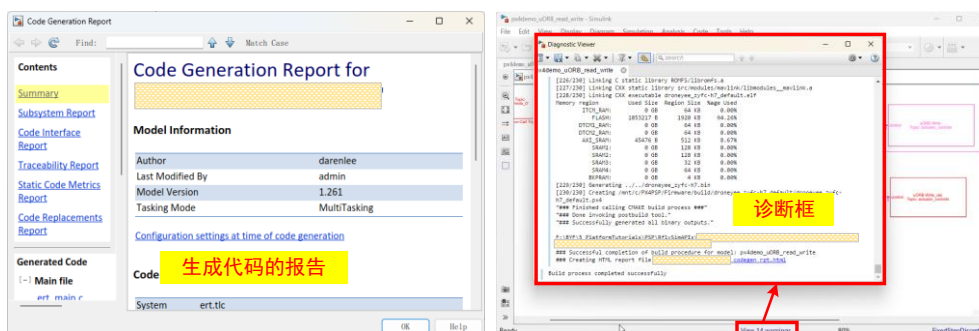
Step 1:

打开 AttitudeSystemCodeGen.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，即可表示编译成功，左图侧为生成的编译报告。

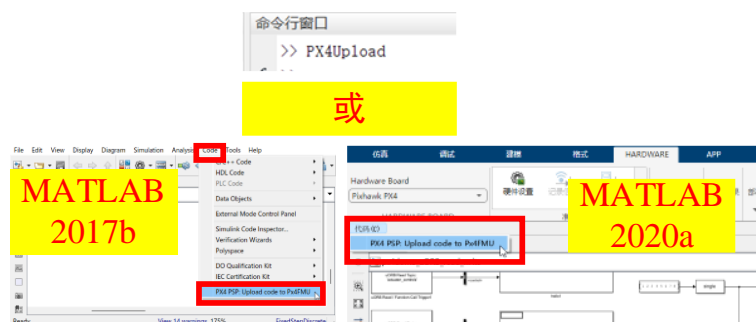


编译成功后，即可以在平台安装路径下找到 .px4 源文件，路径为

*\PX4PSP\Firmware\build\px4_fmu-v6x_default\px4_fmu-v6c_default.px4。

Step 3:

用 USB 数据线链接飞控与电脑。在 MATLAB 命令行窗口输入：PX4Upload 并运行，弹出 CMD 对话框，显示正在上传固件至飞控中，等待上传成功。

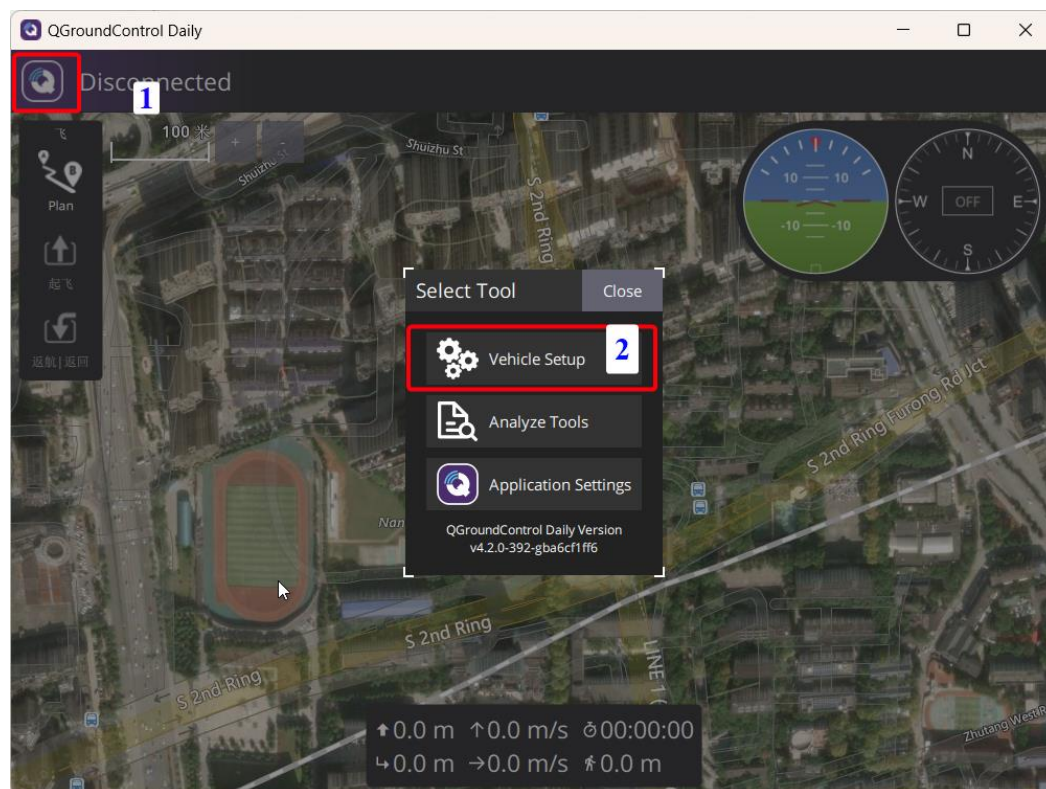


6.3、原生固件生成

原生固件指的是官方发布的、未经修改的飞控固件，这个固件包含了用于控制无人机(多旋翼、固定翼、无人车等)飞行的关键算法和程序，例如姿态控制、位置控制、导航、传感器融合等。下面以 Pixhawk 6C 飞控、需求原生固件版本为 1.13.3 为例，对通过平台生成原生固件的过程进行说明。本小节提供两种烧录飞控原生固件的方式。

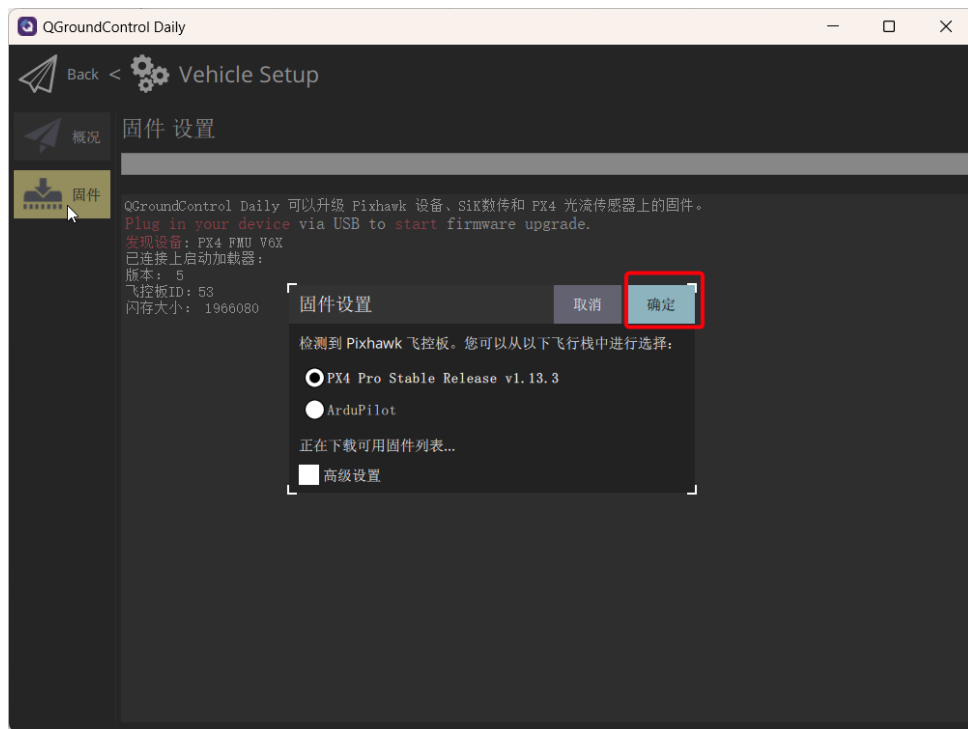
方式一：

打开 QGC 地面站后，按照下图进入 Vehicle Setup 中。



点击到“固件”一栏，将 Pixhawk 6C 飞控通过 USB 链接电脑，在弹出的对话框中，点击

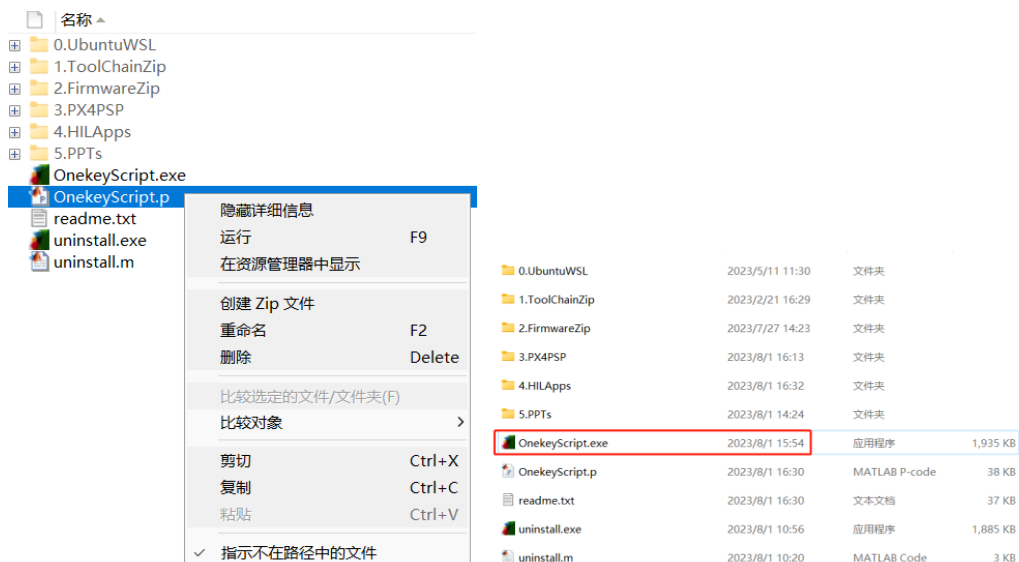
“确定”即可烧录固件。



方式二：

Step 1:

打开 MATLAB 软件，定位到平台安装包路径，右键运行 OnekeyScript.p，或者直接双击运行 OnekeyScript.exe。



Step 2:

脚本运行完毕后，弹出平台一键安装脚本配置栏，假设需要生成固件的飞控类型为 Pixhawk 6C、固件版本为 1.13.3，那此时需要修改一键安装脚本配置为“C:\PX4PSP”、

“px4_fmu-v6c_default”、“7”、“1”、“否”、“否”、“否”、“是”、“是”、“否”，如下图所示，修改完毕后，点击确定。（如果是其他类型 飞控，则“PX4 固件编译命令”修改为对应编译命令，更多飞控编译指令请见：<https://doc.rflysim.com/hardware.html>。）

工具箱一键安装脚本 V2.54-20230801

1.工具包 安装路径
C:\PX4PSP

2.PX4固件编译命令：见Firmware\boards目录，模版px4_fmu-v6x_default、droneyee_racer_default等
px4_fmu-v6c_default

3.PX4固件版本（1: PX4-1.7.3, 4: PX4-1.10.2, 5: PX4-1.11.3, 6: PX4-1.12.3, 7: PX4-1.13.3）
7

4.PX4固件编译器（1: Win10WSL[通用], 2: Msys2[适用版本≤PX4-1.8], 3: Cygwin[适用≥PX4-1.8]
1

5.是否重新安装PSP工具箱(是：重装工具箱，否：维持现有安装)
否

6.是否重新安装其他依赖程序包（CopterSim、QGC地面站、硬件在环仿真软件等，约5分钟）
否

7.是否重新配置固件编译器编译环境（是：全新安装编译器，否：维持原样，重装约5分钟）
否

8.是否重新部署PX4固件代码（是：全新部署代码，否：维持现状，大约5分钟）
是

9.是否预先用选定命令编译固件（是：全新编译固件，否：维持现状，大约5分钟）
是

10.是否屏蔽PX4官方控制器输出(使用Simulink控制器选"是", 使用PX4官方控制器选"否")
否

确定 取消

Step 3:

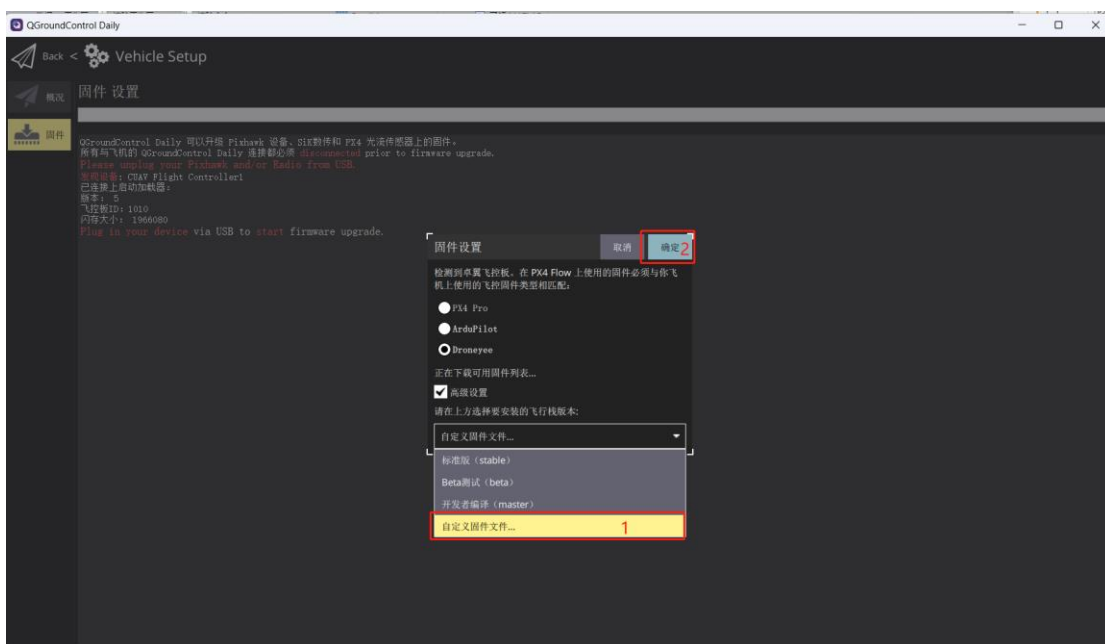
当看到如下图所示情形，表明固件编译结束。

```
C:\Windows\system32\cmd.exe
-- The CXX compiler identification is GNU 7.5.0
-- The C compiler identification is GNU 7.5.0
-- The ASM compiler identification is GNU
-- Found assembler: /usr/bin/cc
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Building for code coverage
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.6.9", minimum required is "3")
-- build type is RelWithDebInfo
-- PX4 ECL: Very Lightweight Estimation & Control Library v1.9.0-rc1-591-gb3fed06
-- ROMFS: ROMFS/px4fmu_common
-- Configuring done
-- Generating done
-- Build files have been written to: /mnt/c/PX4PSP/Firmware/build/px4_sitl_default
[6/812] git submodule src/drivers/gps/devices
[15/812] git submodule src/lib/ecl
[16/812] git submodule mavlink/include/mavlink/v2.0
[812/812] Linking CXX shared library src/examples/dyn_hello/examples_dyn_hello.px4mod
Press Enter to exit
```

此时可在*PX4PSP\Firmware\build\px4_fmu-v6x_default\px4_fmu-v6c_default.px4路径下找到生成的官方固件 px4_fmu-v6c_default.px4。

Step 4:

该固件可通过 QGroundControl 的“固件设置-高级设置-自定义固件文件-选择.px4 文件所在位置”的方式烧录到 Pixhawk 6C 飞控种中。



7、参考资料

[1]. 无。

8、常见问题

Q1: 暂无

A1: 暂无