
1. 实验名称及目的

Simulink 中 M-Function 和 S-Function 对自驾仪系统资源占用对比实验：PX4 固件的飞控系统都是基于 Nuttx 这个操作系统的。Nuttx 是一个实时嵌入式操作系统（Embedded R TOS），它很小巧，在微控制器环境中使用。Nuttx 完全可扩展，可从从小型（8 位）至中型嵌入式（32 位）系统。它的设计目的还在于完全符合 POSIX 标准，完全实时，并完全开放。以 Holybro Pixhawk 6C 为例，其所使用主处理器为 STM32H743，频率为 480MHz，内存为 2MB，协处理器 STM32F103，频率为 72MHz，静态随机存取存储器(Static Random-Access Memory, SRAM)为 64KB。因此在运行较大的算法程序时，可能会造成内存使用爆满，CPU 超载等情况。本实验将分别烧录由 M-Function 和 S-Function 搭建的 Simulink 模型，通过分析自驾仪系统的资源占用情况，可得 S-Function 搭建的 Simulink 模型占用自驾仪资源更少。

2. 实验原理

Fcn：适用于简单的表达式。所谓简单，不一定是表达式很短小，而是指输入输出之间的关系可以直接用一个表达式描述，即不使用状态，也不使用流程控制，而且需要注意，表达式中可用的函数比较有限（基本上就是一些简单的数学函数）。可以生成 C 代码。

MATLAB Fcn：很多人可能更容易掌握这个模块，这个模块对应一个 M 文件，对输入信号可以进行任何处理后得到输出。可以使用 MATLAB 的任何函数。限制主要有两个方面：（1）不能进行代码生成，所以准备做控制原形或半实物仿真的要尽量避免使用；（2）原则上不能使用状态量，也就是说，只使用当前时刻的输入经计算处理得到输出，而不能使用前一时刻的信息——如果需要使用前一时刻的信息，一般需要配合使用外部其他模块（如积分器或单位延时）。

Embedded MATLAB Fcn：和 MATLAB Fcn 类似，区别主要有以下几个方面：（1）可以生成 C 代码；（2）要求比 MATLAB Fcn 严格很多，而且有很多函数或工具箱在 EMF 中不允许使用（因为代码生成的限制）；（3）模块自身支持多输入多输出。本实验中所示的 m-Function 即为本模块。

S-Function：这是 Simulink 自定义模块最难掌握也是最强大的一种，原则上可以用它实现 Simulink 任何模块的功能。S 函数与前几种模块的最大区别是可以使用状态，其根据仿真过程把代码分成几个部分：初始化、计算导数（积分由 Simulink 系统完成）、更新状态、计算输出、终止处理（其中中间几个是在仿真过程中周期性调用的），这给了用户以非常大的自由来进行代码的组织。S 函数可以用 C 语言或 M 语言实现，而 M 语言又分不同的 Level。写 S 函数有相应的模板和示例，想学的话需要花上一些时间才能搞明白。

3. 实验效果

M-Function 对飞控硬件系统的占有率是 S-Function 占用的 **1.026 倍**。

4. 文件目录

文件夹/文件名称	说明
无	无

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	Pixhawk 6C ^②	1
		数据线、杜邦线等	若干

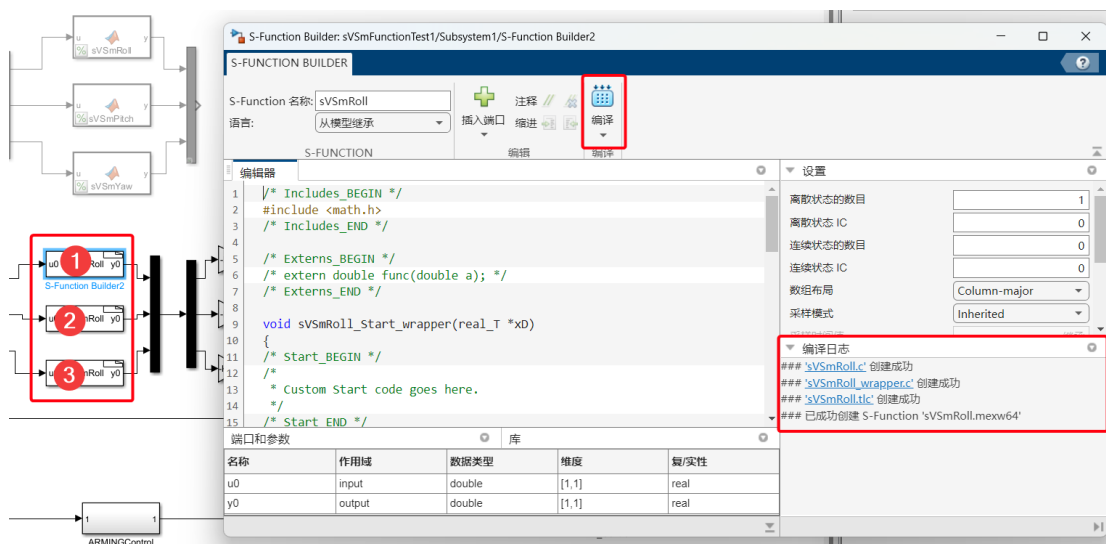
①：推荐配置请见：<https://doc.rflysim.com>

②：须保证平台安装时的编译命令为：px4_fmu-v6c_default，固件版本为：1.13.3。其他配套飞控请见：<http://doc.rflysim.com>

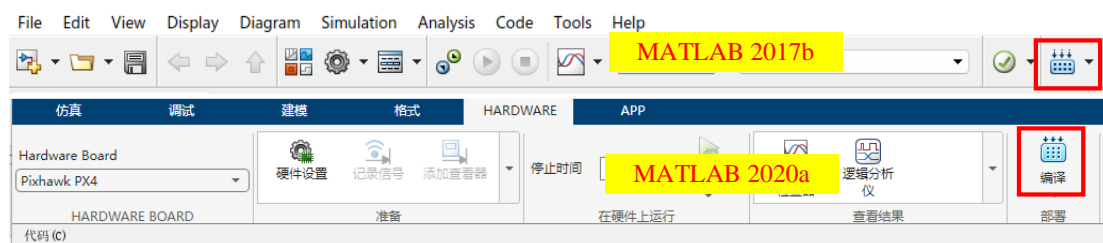
6. 实验步骤

Step 1:

打开 MATLAB 软件，在 MATLAB 中打开 [sVSFunctionTest.slx](#) 文件，该文件中默认下使用了三个 S-Function 模块分别编写了从 1 叠加到 100 的运算。点击进入每个 S-Function Builder 模块中进行编译等待编译成功，如下：

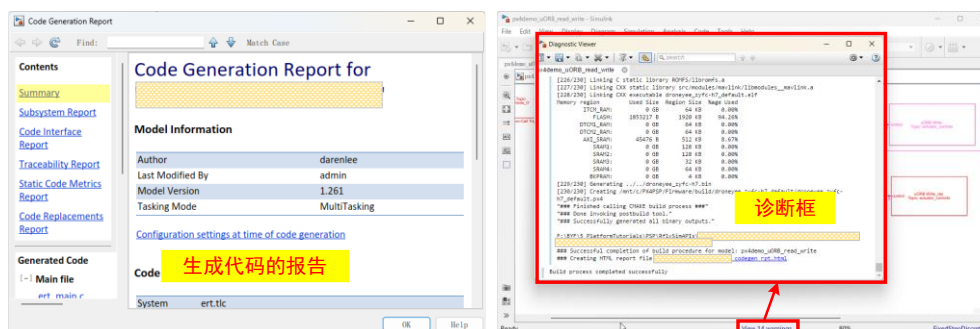


在 Simulink 中，点击编译命令。



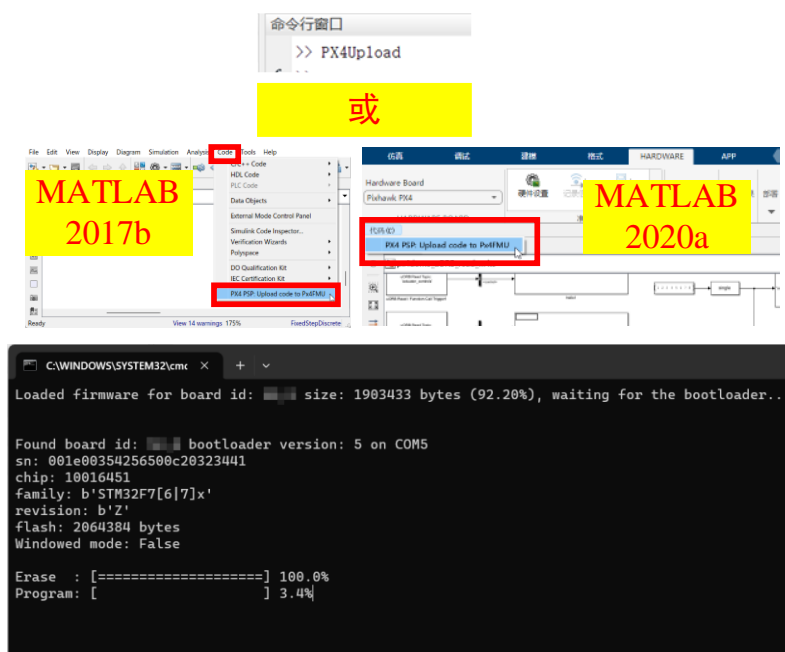
Step 2:

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，左图为生成的编译报告。



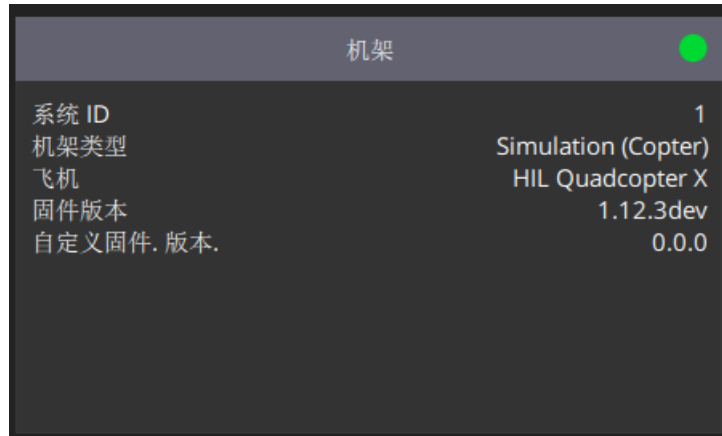
Step 3:

用 USB 数据线链接飞控与电脑。在 MATLAB 命令行窗口输入：PX4Upload 并运行或点击 PX4 PSP: Upload code to Px4FMU，弹出 CMD 对话框，显示正在上传固件至飞控中，等待上传成功。

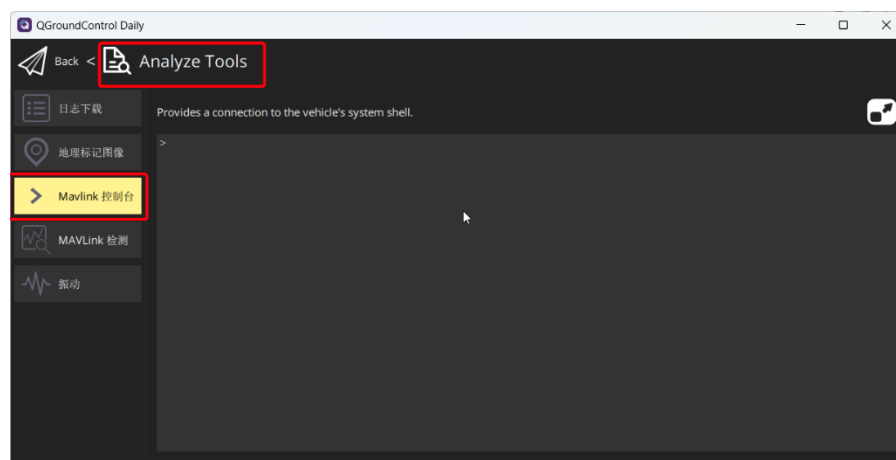


Step 4:

打开 QGroundControl 软件，设置为硬件在环的机架，

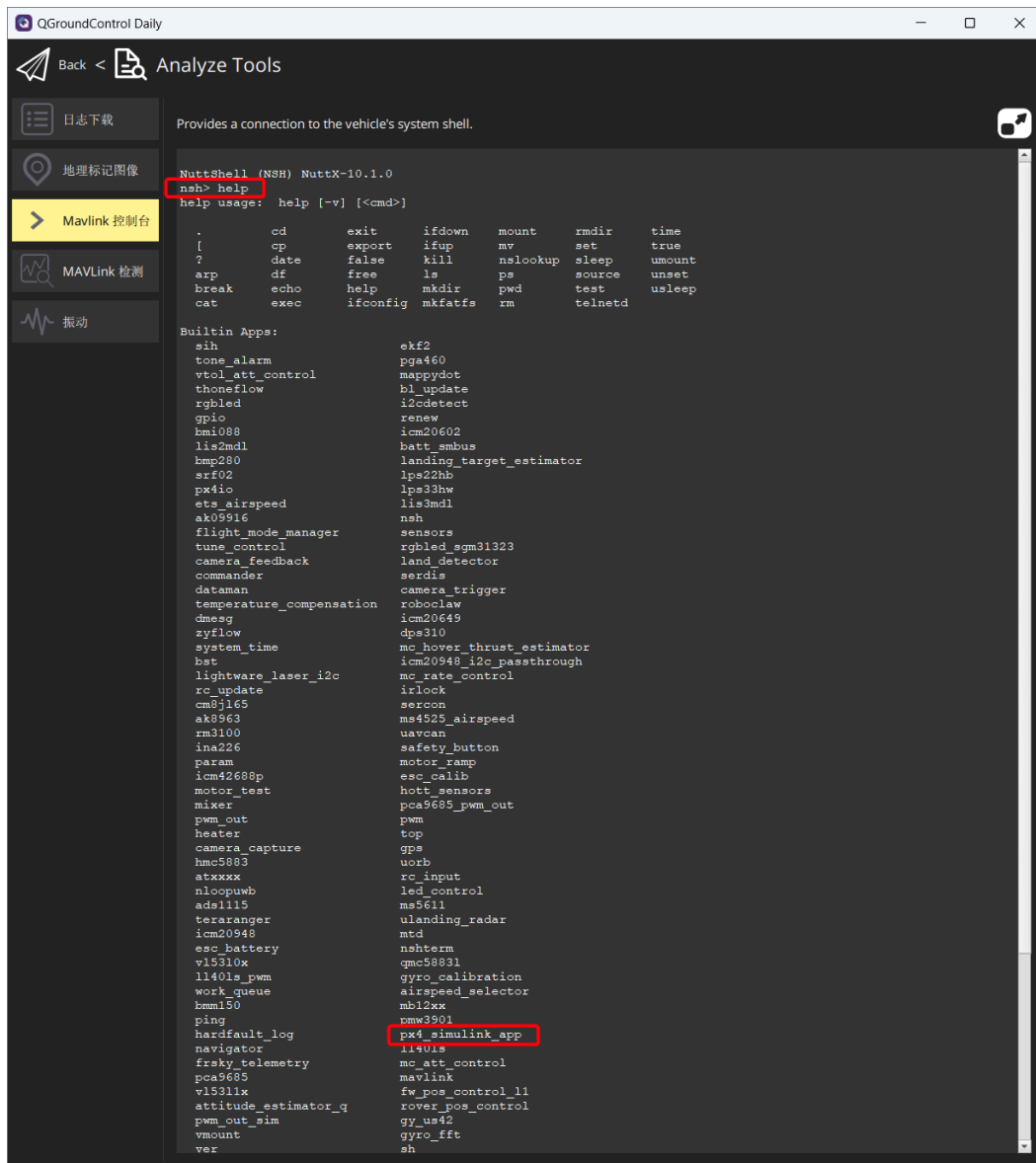


在 QGroundControl 软件，点击 “Q” Logo->Analyze Tools->Mavlink 控制台，该控制台即为在自驾仪中运行的 Nuttx 系统的 shell 界面。

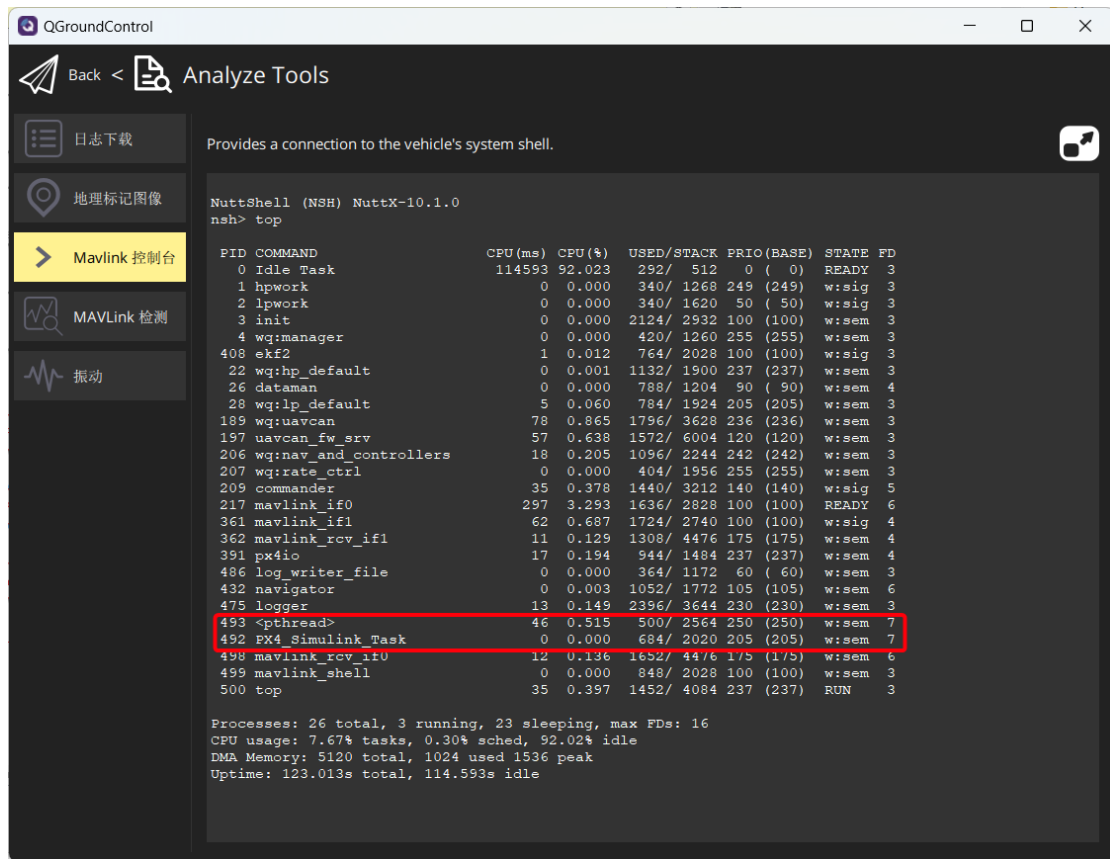


Step 5:

在该 shell 窗口中运行 `help` 将显示当前系统中所有的进程，可看到生成的 `px4_simulink_app` 即为 MATLAB 生成的应用模块。

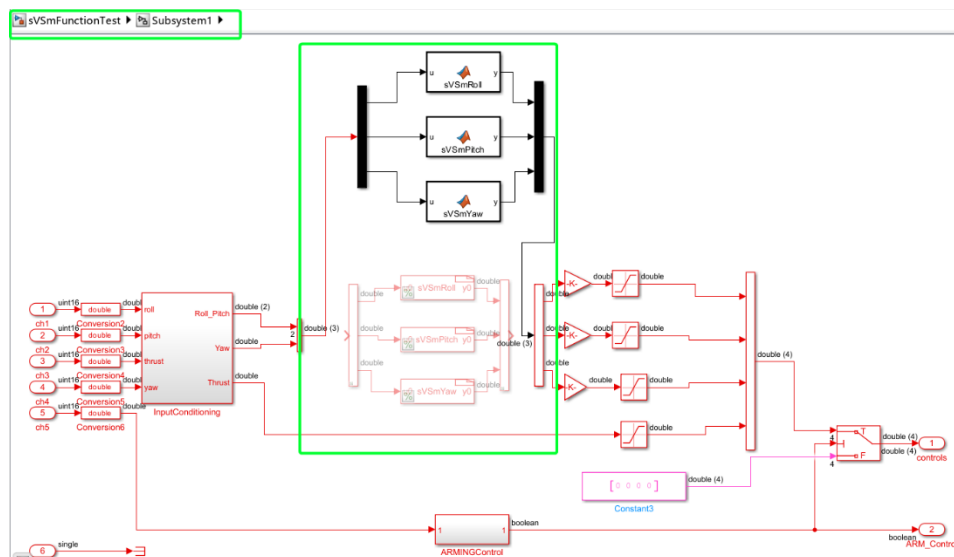


输入 **top** 即可查看当前系统的应用运行情况，可看到占用的 CPU 的总占用为 **92.023%**

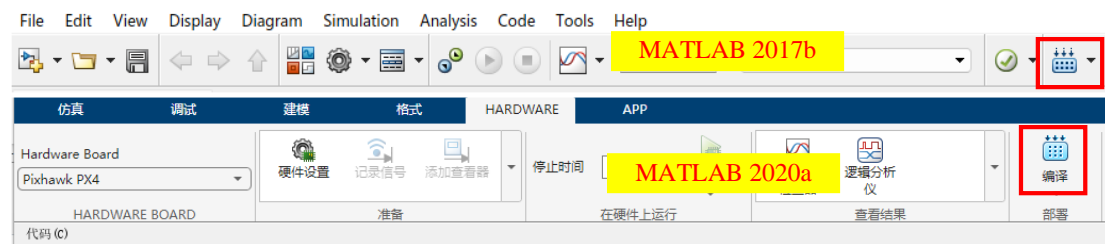


Step 6:

返回 MATLAB 软件的 [sVSmFunctionTest.slx](#) 文件，该文件中在三个 S-Function 模块旁边还有三个 m-Function 模块也是分别编写了从 1 叠加到 100 的运算。将三个 S-Function 模块注释，取消注释三个 m-Function 模块，如下图所示。

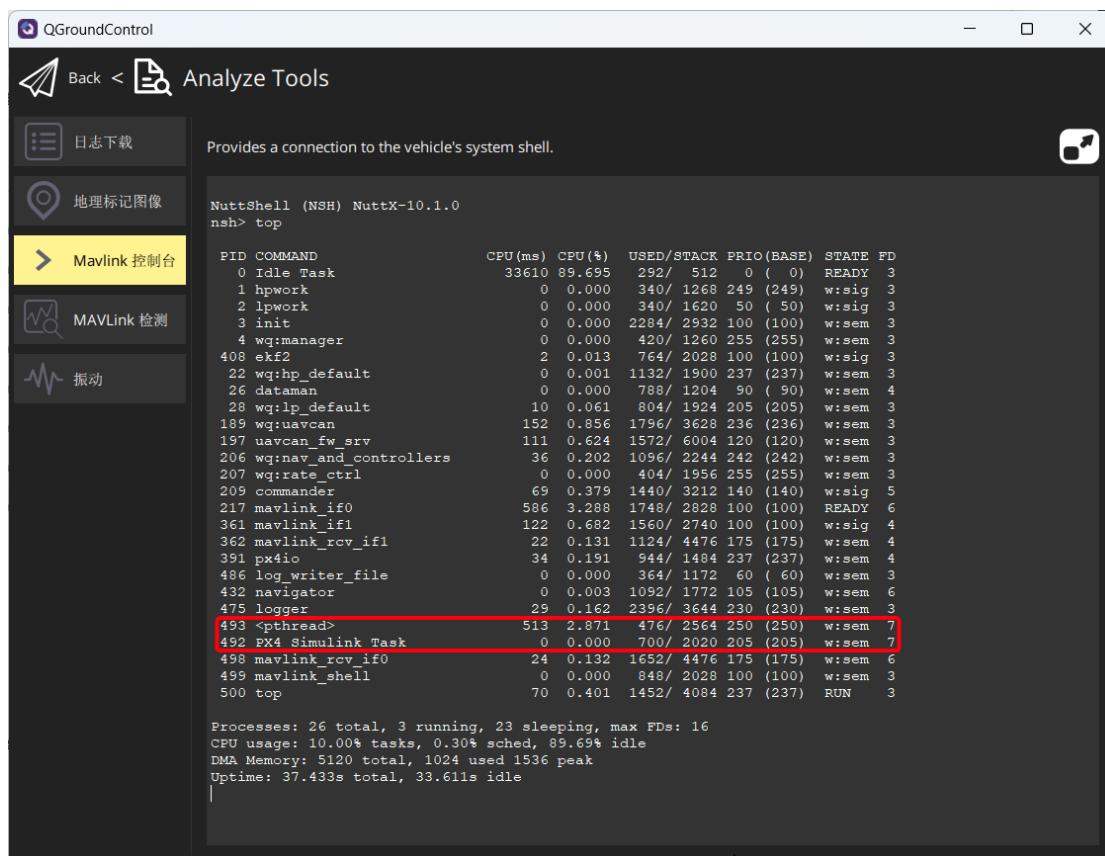


在 Simulink 中，点击编译命令。



Step 7:

重复上述步骤 [Step 2:](#) ~[Step 5:](#)。在 Nuttx 系统的 shell 界面输入 `top` 即可查看当前系统的应用运行情况，可看到占用的 CPU 总占用内存为 **89.695%**。



对比可得出，同样的功能通过 S-Function 和 M-Function 来实现，前者对于飞控的资源占用率更少，后者占用的飞控资源更多，具体来说 **M-Function 对飞控硬件系统的占有率是 S-Function 占用的 1.026 倍。**

7. 参考资料

[1]. 无

8. 常见问题

Q1: 暂无

A1: 暂无