

Simulink中M-Function和S-Function对自驾仪系统资源占用对比实验

1. 实验目的

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 RTOS)，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 Pixhawk6C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. 实验要求

- 软件要求：Windows 10及以上版本；RflySim工具链^[1]，MATLAB2022B以上版本，平台安装时的编译命令为：px4_fmuv6x_default，推荐PX4固件版本为：1.14.3。
- 硬件要求：笔记本/台式电脑1台^[2]，遥控器和遥控器接收机；数据线和杜邦线等。

3. 实验地址

例程目录：

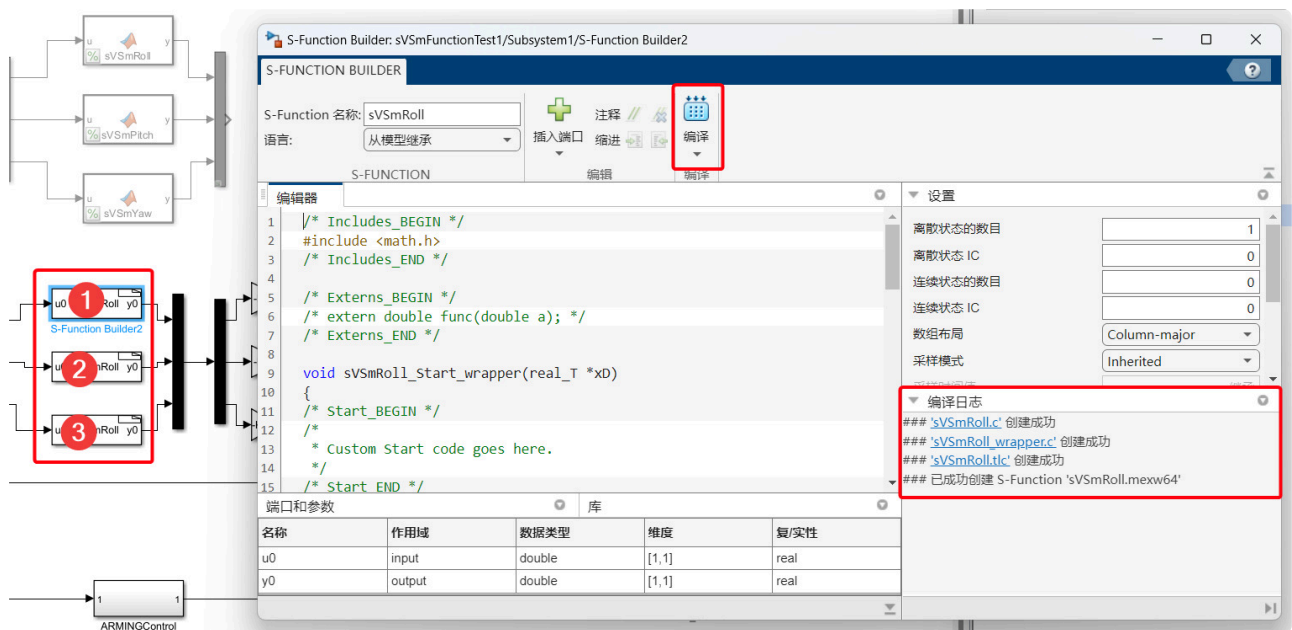
[\[安装目录\]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\13.Simulink_MS_FuncVS](#)

- sVSmFunctionTest.slx：模型文件

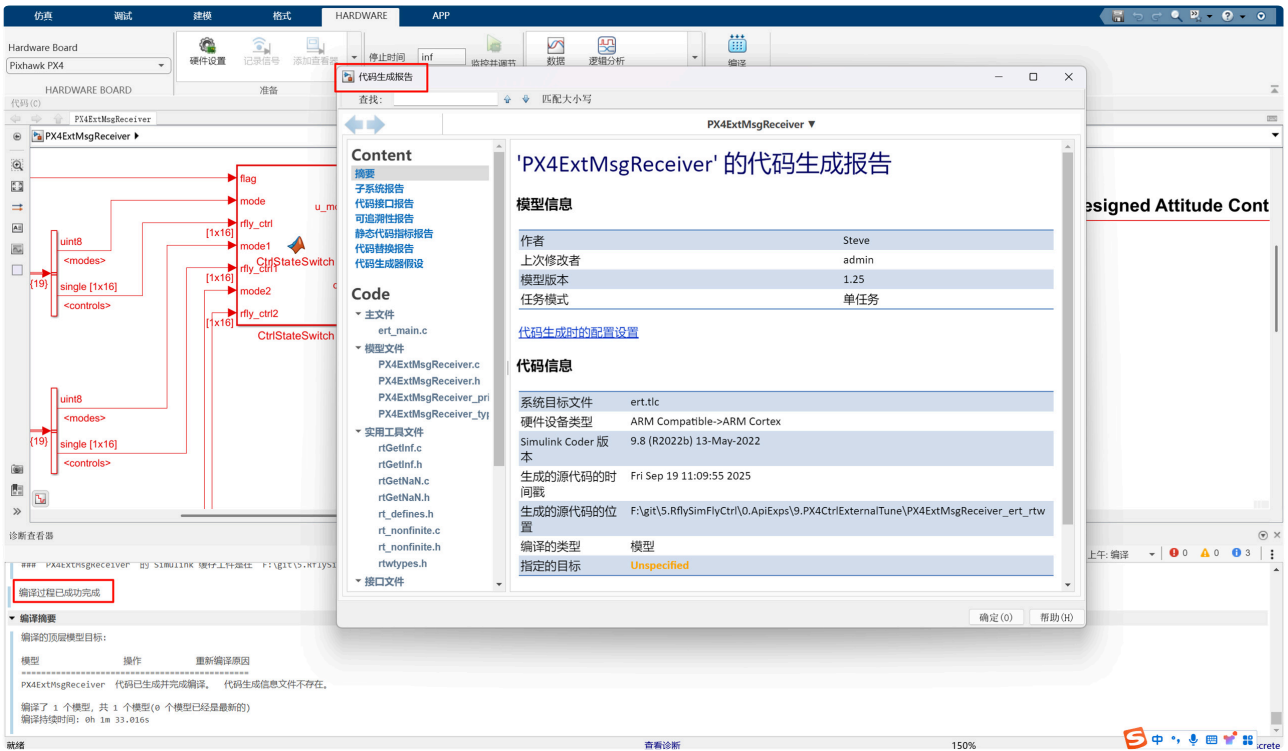
4. 实验内容或步骤

步骤1: Simulink中M-Function和S-Function对自驾仪系统资源占用对比

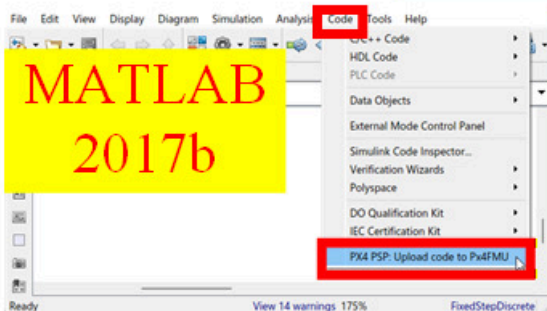
打开MATLAB软件，在MATLAB中打开[sVSmFunctionTest.slx]文件，该文件中默认下使用了三个S-Function模块分别编写了从1叠加到100的运算。点击进入每个S-Function:Builder模块中进行编译等待编译成功，如下：



在Simulink中，点击编译命令。在Simulink的下方点击查看诊断，即可弹出诊断对话框，可查看编译过程。在诊断框中弹出编译过程已成功完成，即可表示编译成功，也会弹出代码生成报告。



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PX4PSP: Upload code toPx4FMU，弹出CMD对话框，显示正在上传固件至飞控中，等待上传成功。

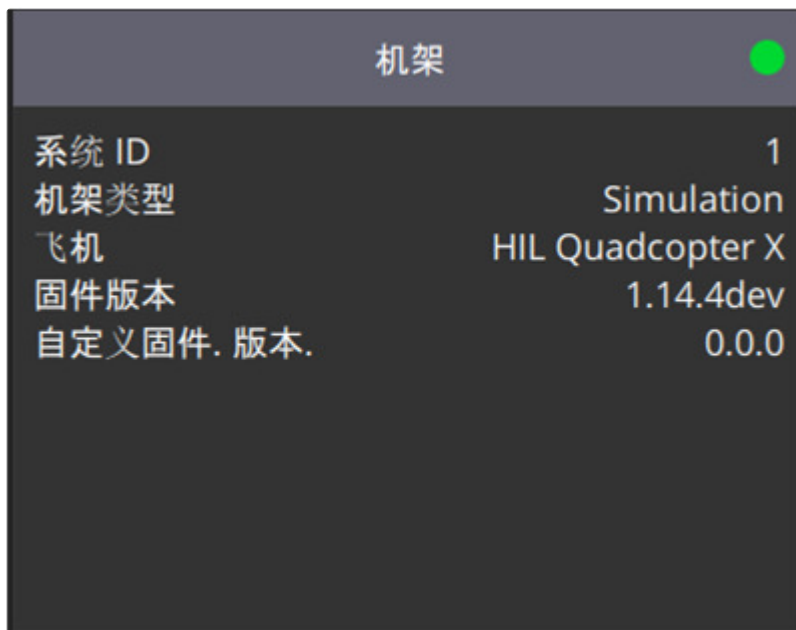


```
C:\WINDOWS\SYSTEM32\cmd  x  +  v
Loaded firmware for board id: [REDACTED] size: 1903433 bytes (92.20%), waiting for the bootloader...

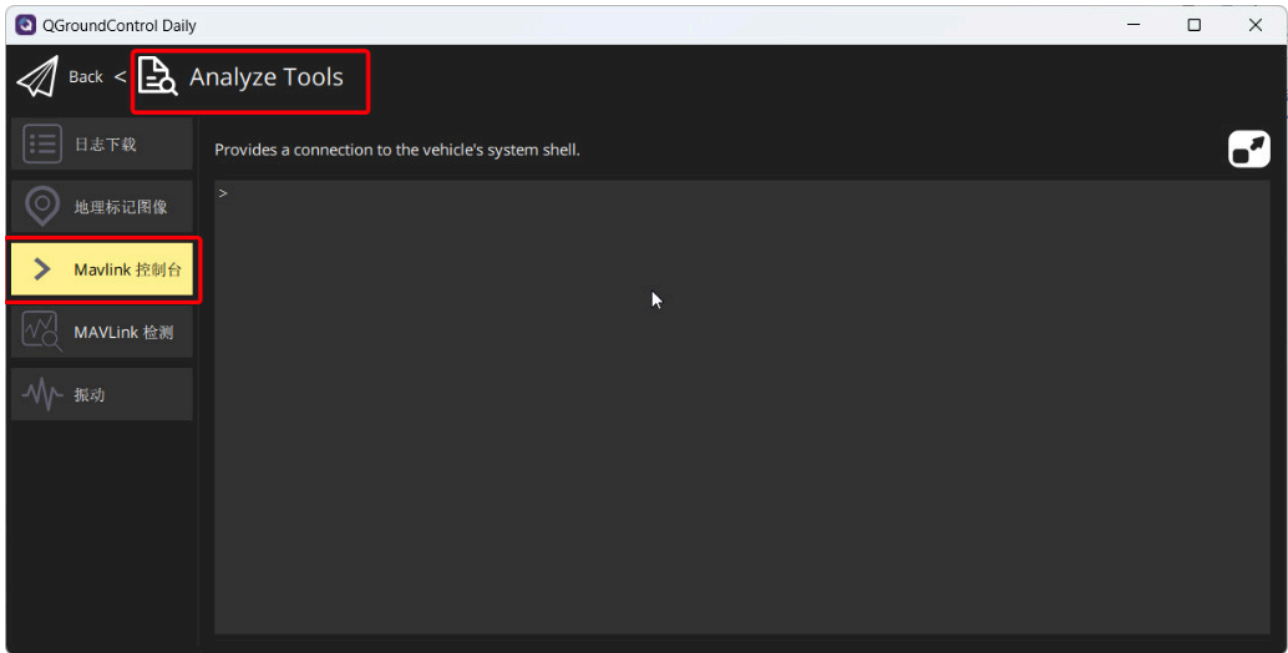
Found board id: [REDACTED] bootloader version: 5 on COM5
sn: 001e00354256500c20323441
chip: 10016451
family: b'STM32F7[6|7]x'
revision: b'Z'
flash: 2064384 bytes
Windowed mode: False

Erase : [=====] 100.0%
Program: [ ] 3.4%
```

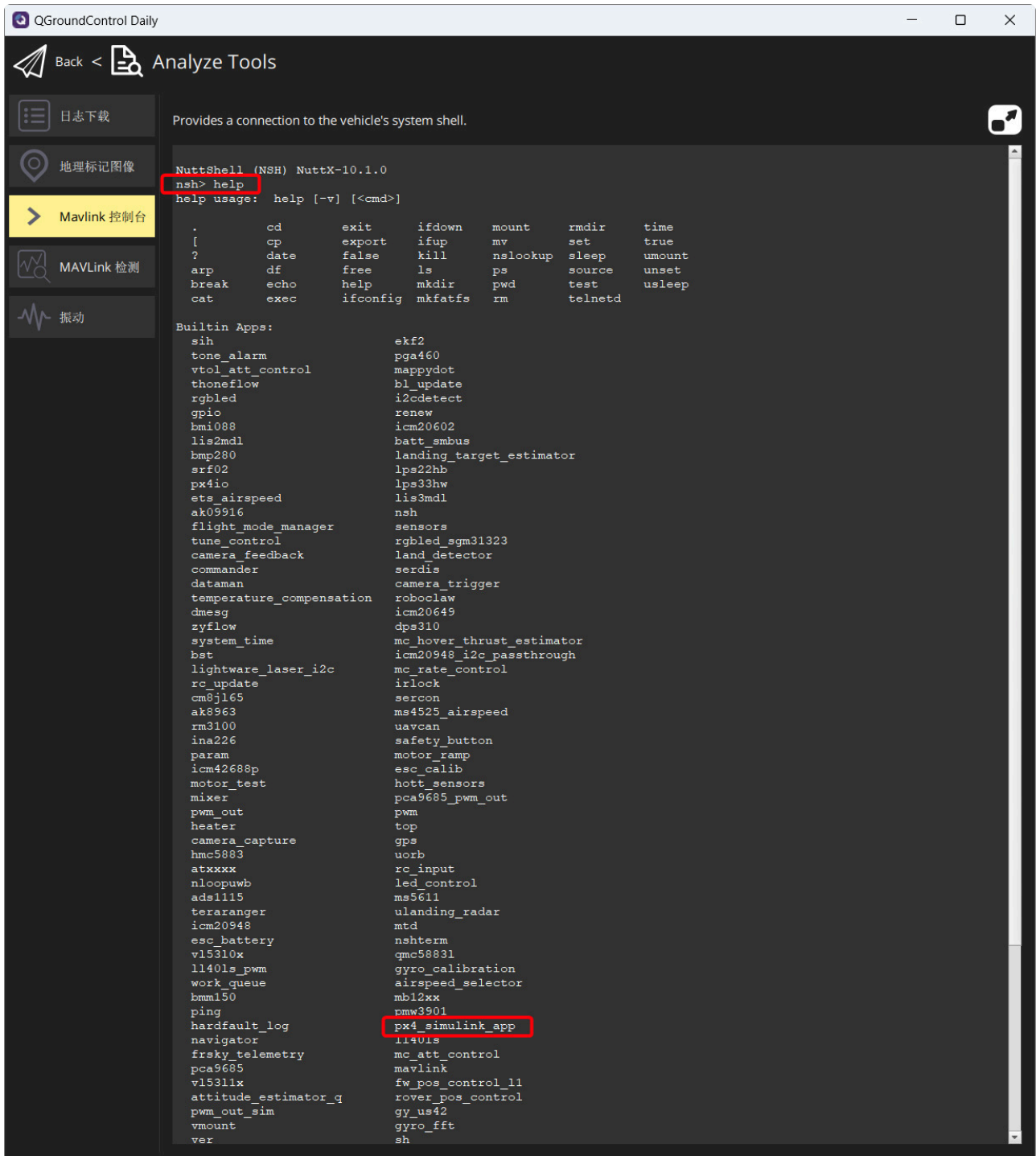
打开QGrounControl软件，设置为硬件在环的机架，



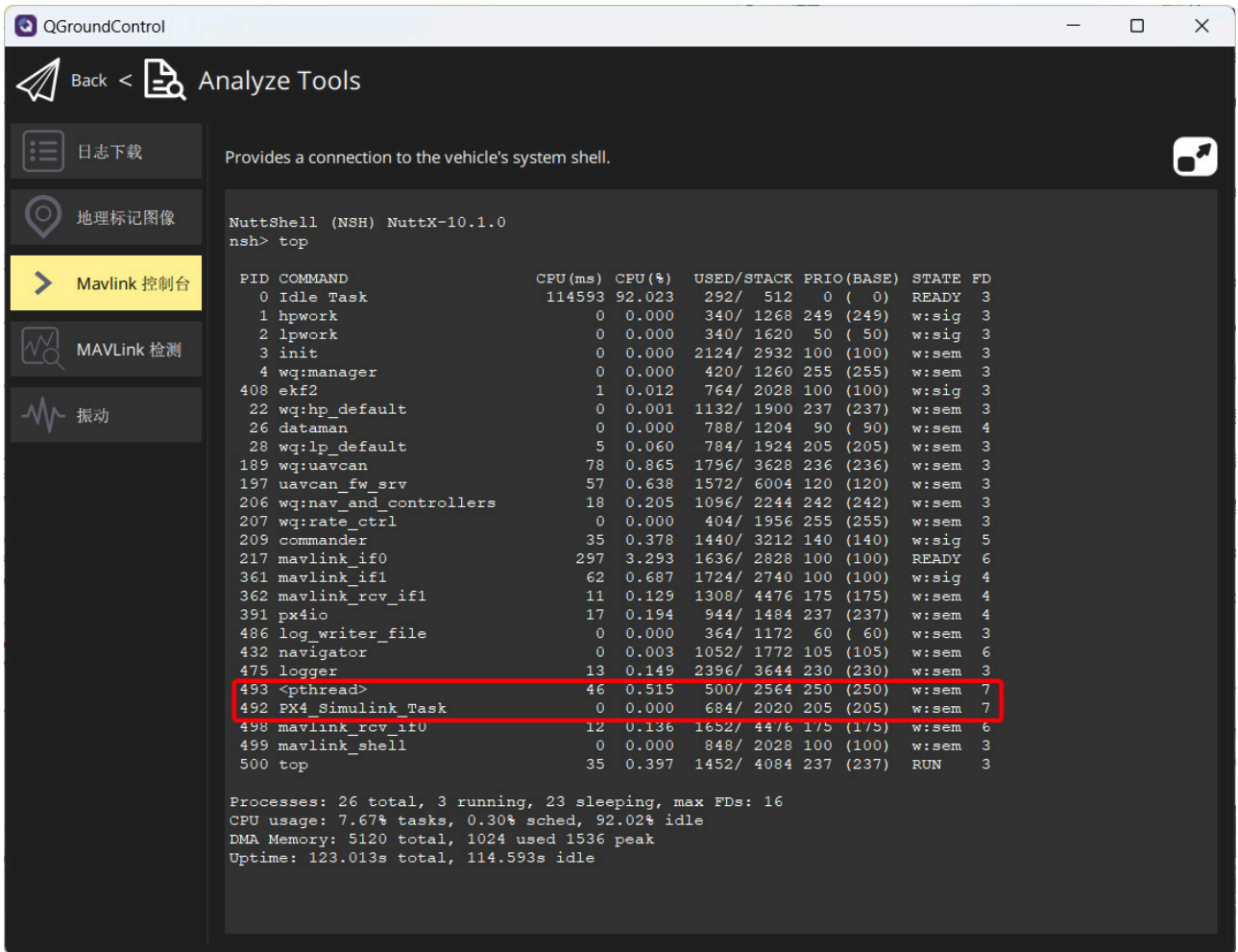
在QGrounControl软件，点击“Q” Logo->Analyze Tools->Mavlink控制台，该控制台即为在自驾仪中运行的Nuttx系统的shell界面。



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。

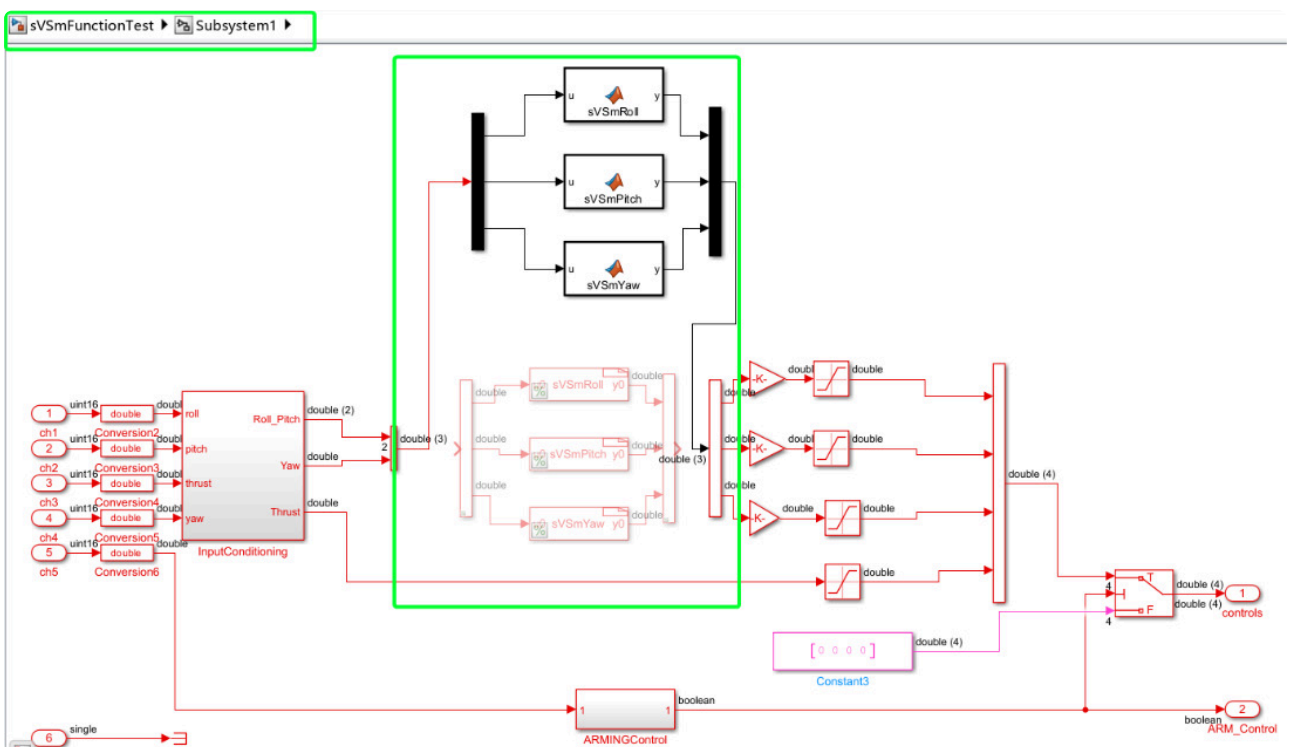


输入top即可查看当前系统的应用运行情况，可看到占用的CPU的总占用为**92.023%**

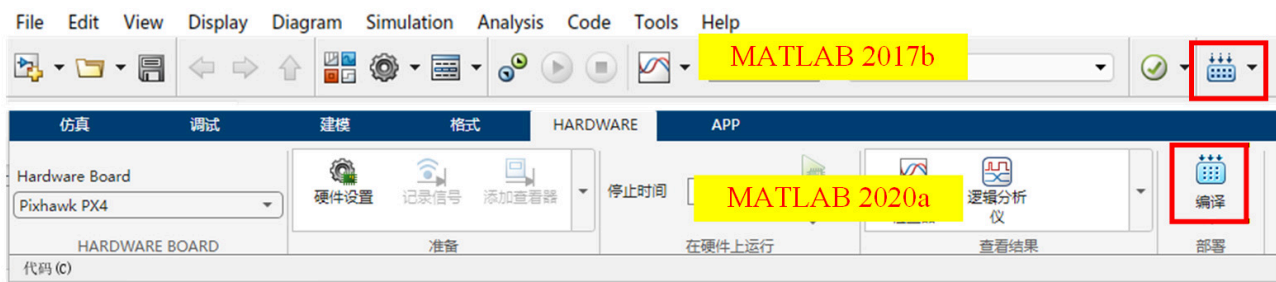


返回MATLAB软件的[sVSmFunctionTest.slx]

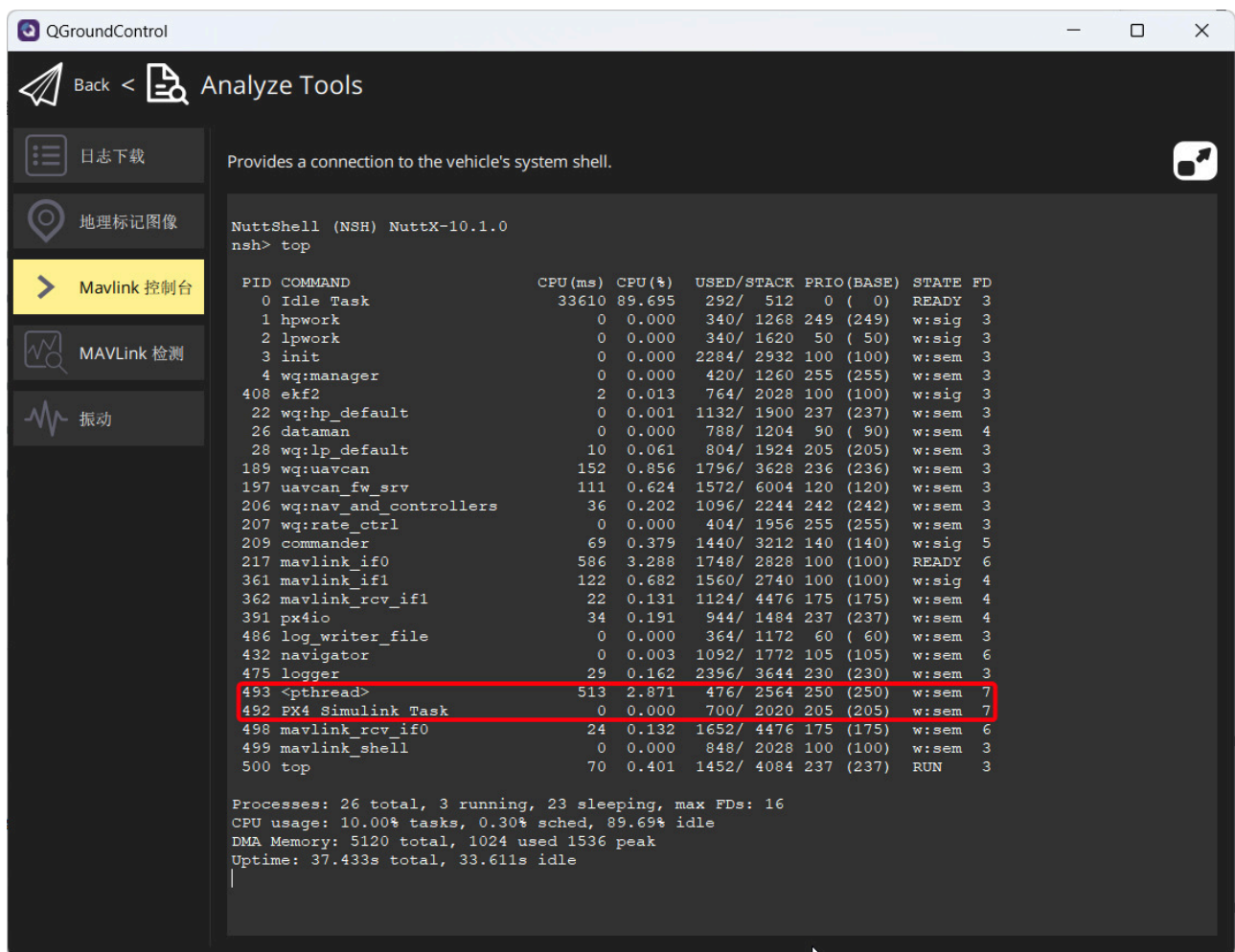
(file:///C:/Users/86181/Desktop/sVSmFunctionTest.slx)文件，该文件中在三个S-Function模块旁边还有三个m-Function模块也是分别编写了从1叠加到100的运算。将三个S-Function模块注释，取消注释三个m-Function模块，如下图所示。



在Simulink中，点击编译命令，



重复上述步骤。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。

5. 关键知识点

关键知识点1: Simulink函数模块类型对比

Fcn模块:

- 适用于简单的数学表达式
- 输入输出关系可直接用表达式描述
- 不使用状态和流程控制
- 可用函数有限, 但可生成C代码

MATLAB Fcn模块:

- 对应M文件, 可进行任何处理得到输出
- 可使用MATLAB任何函数
- 限制: 不能进行代码生成, 不能使用状态量
- 如需前一时间信息需配合外部模块

Embedded MATLAB Fcn模块:

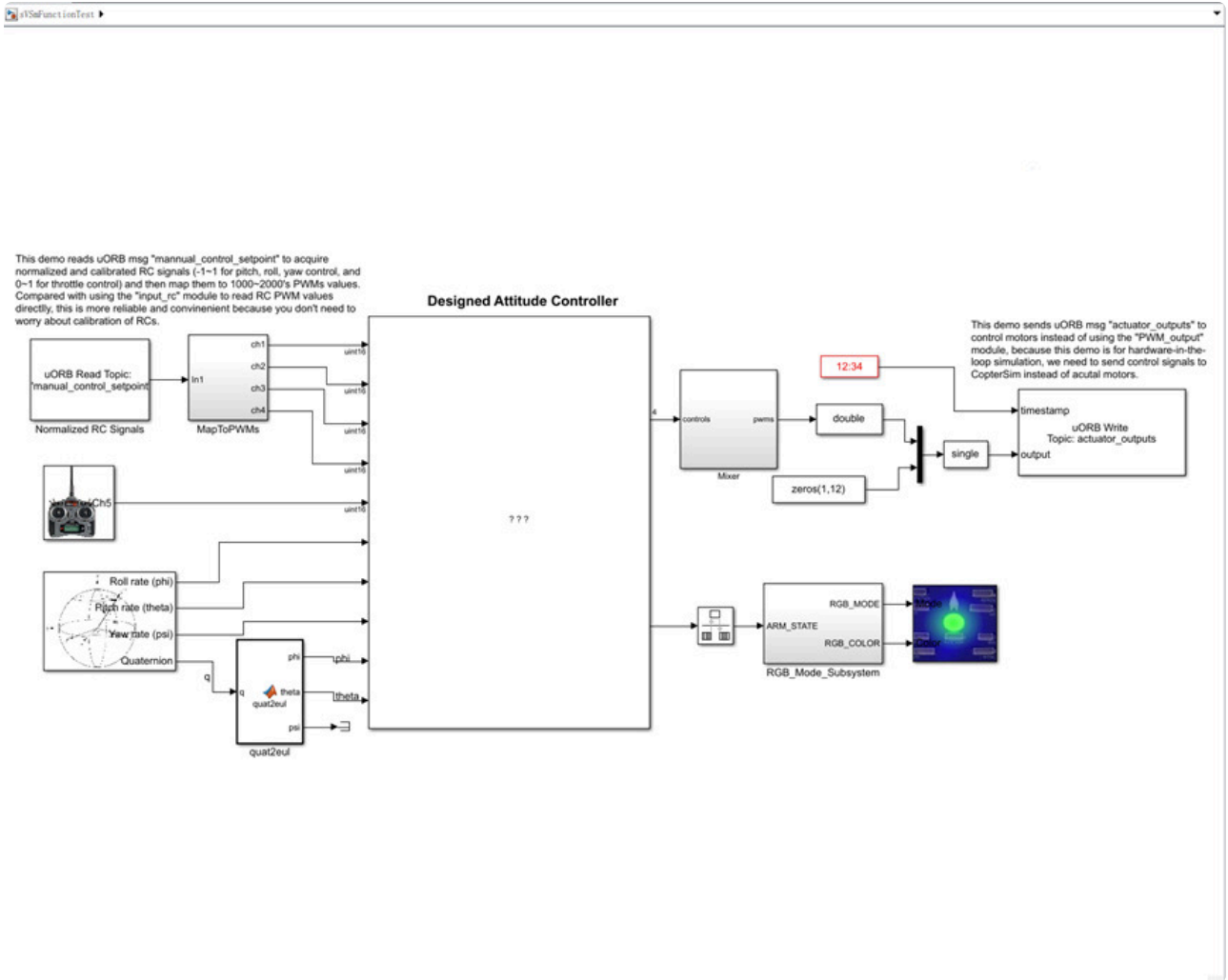
- 可生成C代码
- 要求比MATLAB Fcn严格
- 有很多函数或工具箱在EMF中不允许使用
- 模块自身支持多输入多输出

S-Function模块:

- 最强大但最难掌握的自定义模块
- 可使用状态, 功能最全面
- 代码分为初始化、计算导数、更新状态、计算输出、终止处理等部分
- 可用C语言或M语言实现

关键知识点2: 子模块介绍

打开[\[安装目录\]\5.RflySimFlyCtrl\1.BasicExps\e9-ReplacePX4Module](#)中的sVSmFunctionTest.slx文件。该系统主要由uORB Read and Function-Call Trigger、uORB Write Advanced、input_rc、vehicle_attitude、Subsystem和RGB_LED构成。



uORB Read and Function-Call Trigger、uORB Write Advanced、input_rc、vehicle_attitude和RGB_LED具体用法和功能请参考：
[API.pdf](#)

对于Subsystem模块的详细介绍请参考 [readme.pdf](#)。

关键知识点3:

关键知识点4:

6.参考资料

1. RflySim官方文档：<https://rflysim.com/doc/zh/>
2. PX4固件相关文档：项目基于PX4固件的NuttX实时嵌入式操作系统开发
3. Simulink建模与仿真：使用MATLAB/Simulink进行飞控系统建模，特别是S-Function和M-Function模块的使用

7. 常见问题

Q1: 为什么S-Function相比M-Function占用更少的飞控资源?

A1: S-Function是Simulink中最强大的自定义模块，它将代码分为初始化、计算导数、更新状态、计算输出、终止处理等部分，这种结构化的组织方式使得生成的代码更加高效。相比之下，M-Function虽然易于使用且功能全面，但在代码生成效率上不如S-Function优化得好，导致在资源受限的嵌入式系统（如飞控）上运行时占用更多的CPU和内存资源。

Q2: 如何在Simulink中切换使用S-Function和M-Function进行对比测试?

A2: 在提供的sVSmFunctionTest.slx模型文件中，默认使用三个S-Function模块执行累加运算。要测试M-Function，需要注释掉三个S-Function模块，并取消注释旁边的三个m-Function模块，然后重新编译并上传到飞控硬件进行资源占用测试。

Q3: 飞控系统资源受限会对算法实现产生什么影响? 如何优化?

A3: 以Holybro Pixhawk6C为例，其主处理器STM32H743仅有2MB内存，协处理器STM32F103的SRAM仅64KB，在运行复杂算法时容易出现内存耗尽、CPU过载等问题。为了优化资源使用，建议：

- 使用S-Function替代M-Function实现关键算法
- 尽可能简化算法复杂度
- 避免使用占用资源较多的函数和数据结构
- 在建模阶段就考虑目标硬件的资源限制

1. <https://rflysim.com/> ↩

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