



MPLAB® Harmony 3 之基础篇（30）

-- 使用 SAME54 进行无位置传感器 PMSM 控制

Microchip Technology Inc.

MCU32 产品部

一、概述

ATSAME54 系列 MCU 是基于 Cortex-M4 架构，为电机控制应用设计的控制器。它具有：

- ❖ 带有浮点运算单元（FPU）和数字信号处理（DSP）指令集的 CPU；
- ❖ 两个脉冲宽度调制（PWM）模块，各自可输出四路互补 PWM 信号，且拥有死区、反相、模式覆盖等功能；
- ❖ 两个采样率为 1MSPS 的模数转换器（ADC），且可被 PWM 模块同时触发；
- ❖ 用于正交编码信号或霍尔信号的位置解码器（PDEC）；

本文将演示如何使用 MPLAB Harmony3 基于 SAME54 及例程进行 PMSM 无位置传感器的控制；以使用户可以快速进行电机控制应用的开发。具体来看，本文将通过对电机三种不同的控制模式来了解该控制方案：

控制模式 1-开环模式：

- 电机工作在开环模式，不需要实际的电机转子位置信息；
- 我们可以观察此模式下，开环位置与估算位置之间的差异；

控制模式 2-转矩模式：

- 电机工作在转矩模式，对电机电流进行闭环控制，不直接控制转速，转矩指令由电位器输入；

控制模式 3-速度模式：

- 电机工作在速度模式，对电机转速进行闭环控制，转速指令由电位器输入；

（一） 基础知识

为了更好地理解本文档内容，建议具备以下知识：

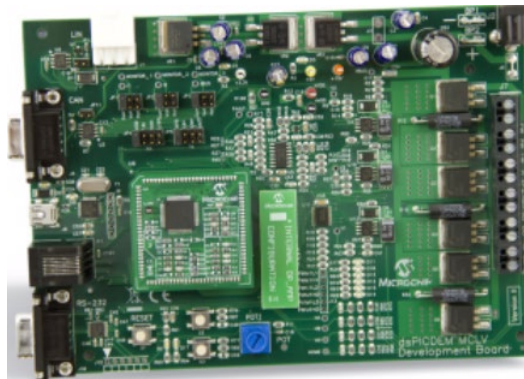
1. MPLAB X IDE 的基本操作
2. 基础的嵌入式编程/调试经验
3. C 语言编程知识

4. SAME54 微控制器的基础知识
5. 电机的 FOC 控制原理

（二） 所需工具

硬件：

1. [MCLV-2 低压电机控制印板](#)



2. [SAME54 PIM \(plug-in-module\)](#)



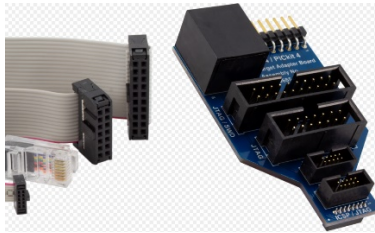
3. [PMSM 电机 \(Hurst DMA0204024B101\)](#)



4. [MPLAB ICD 4 调试器](#)



5. [MPLAB ICD 4 转接套件](#)



软件：

1. MPLAB X IDE: v5.20 或更新版本
2. MPLAB XC32 编译器: v2.15 或更新版本
3. MPLAB Harmony 3 Configurator (MHC) v3.4.0 或更新版本
4. MPLAB Harmony 3 framework
 - csp v3.5.2
 - dev_packs v3.5.0
 - mhc v3.3.4
 - motor_control v3.4.0
5. X2CScope 串口调试工具: v2.0.6 或更新版本

二、 打开 MPLAB X 工程

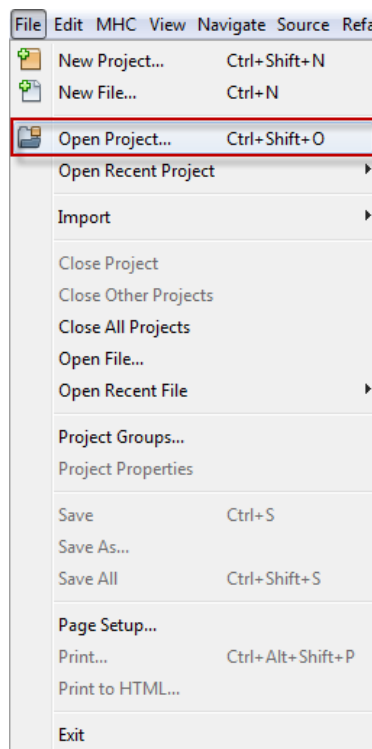
本文所需的 MPLAB X 工程文件 Lab_SAME54_MC_Sensorless.zip 是本文档的附件，可直接在本文档末尾的“附件”章节中获取。

1. 打开 MPLAB X IDE

2. File → Open Project...

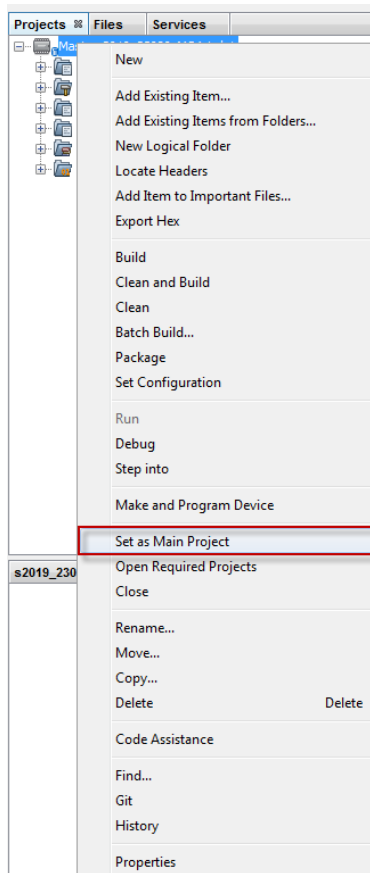
然后打开项目：

Lab_SAME54_MC_Sensorless\firmware\Masters2019_23089_MC4_Lab2.X



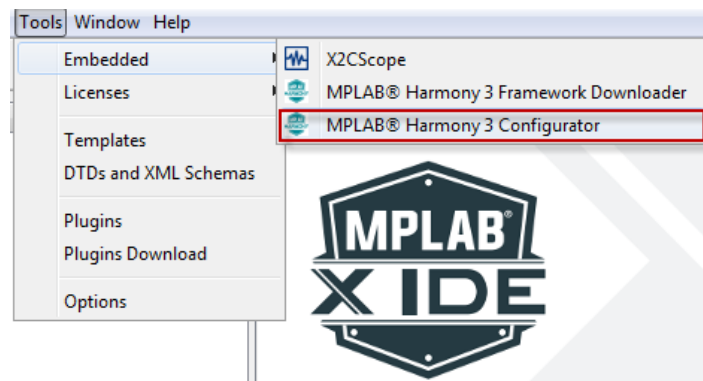
3. 将该项目设定为“main project”

在项目名称上右键单击鼠标；在弹出菜单中点击“Set as Main Project”。



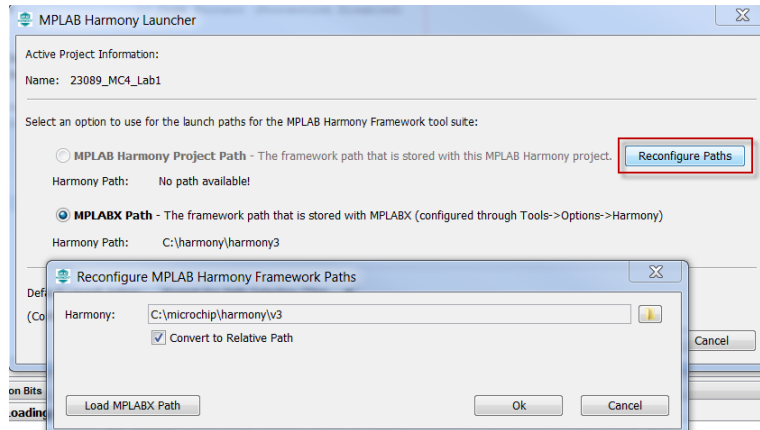
4. 打开 Harmony3 配置器

Tools → Embedded → MPLAB Harmony 3 Configurator

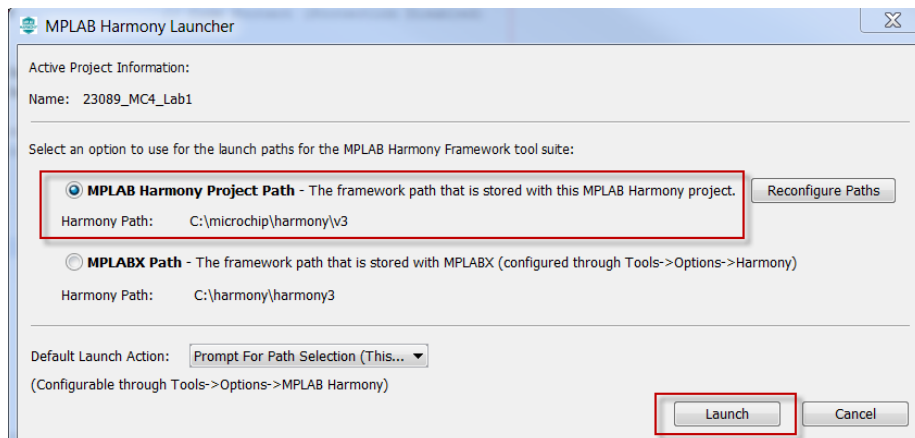


5. 选择或确认 Harmony3 框架（framework）所在路径

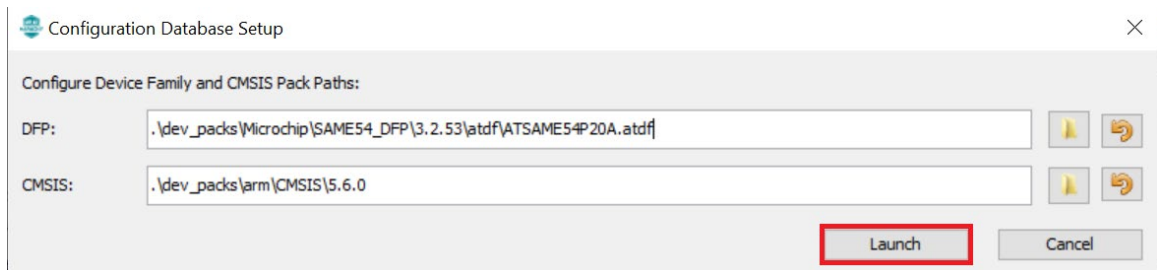
如果默认路径不正确，则点击“Reconfigure Paths”按钮；然后输入正确的路径。



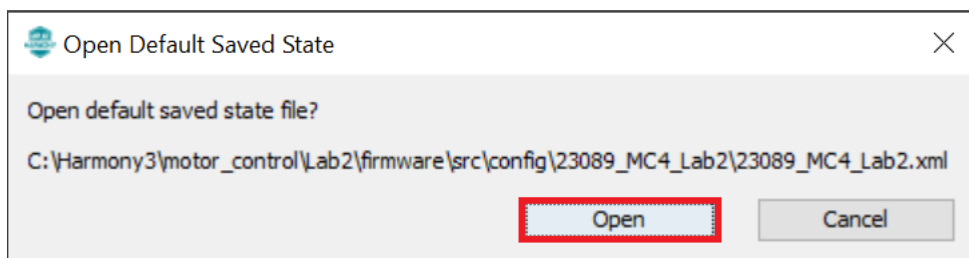
确认路径正确后，点击“Launch”



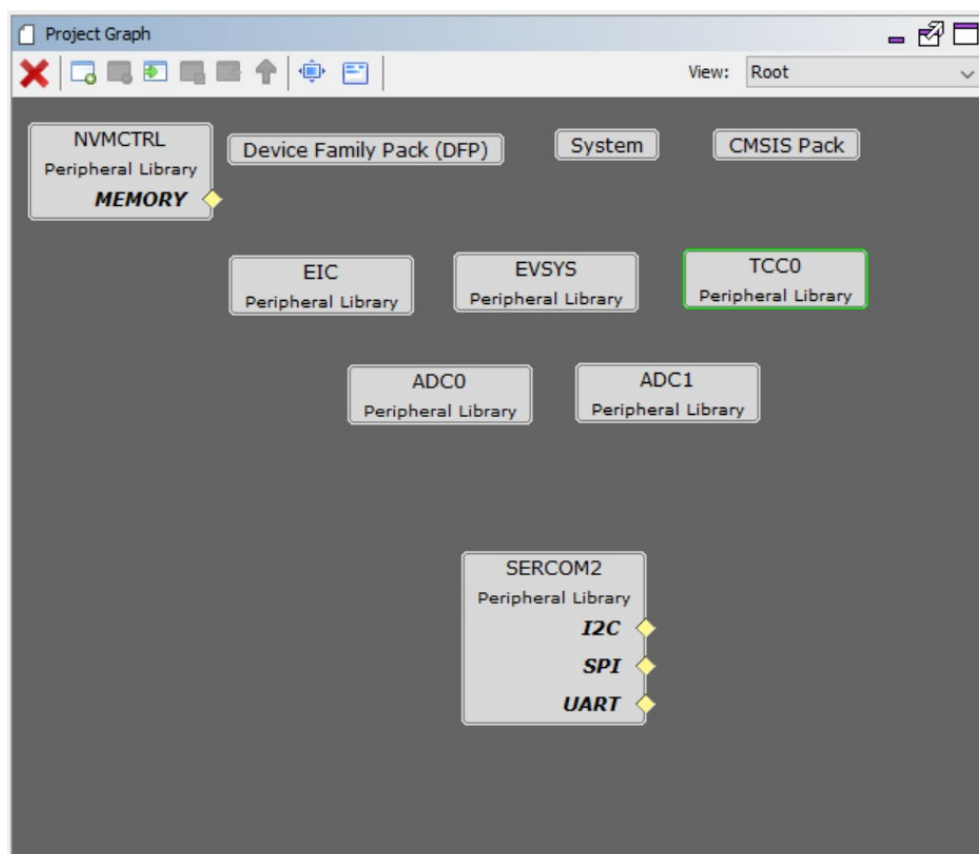
弹出以下窗口，直接使用默认路径，点击“Launch”



当跳出窗口询问，是否打开默认状态？则点击“Open”

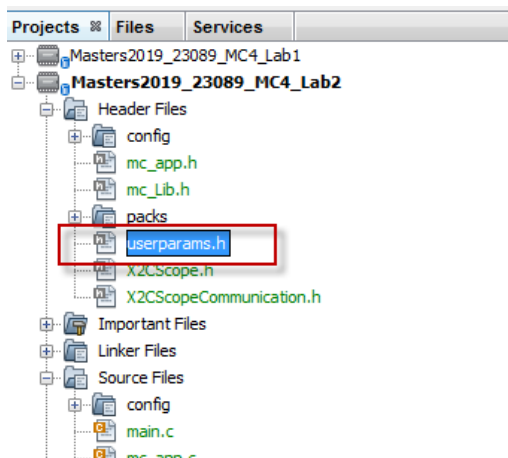


本例程涉及的模块如下所示，用户可查看每个模块的具体配置，具体关于电机应用相关模块的 MHC 配置操作可参考“[MPLAB® Harmony 3 之基础篇](#)
[\(25\) -- 使用 Harmony 3 配置 SAME54 的电机控制周边模块](#)”



三、控制模式 1-开环模式


1. 打开头文件 “userparams.h”




2. 在 “userparams.h” 文件中，定义宏: “OPEN_LOOP_FUNCTIONING”

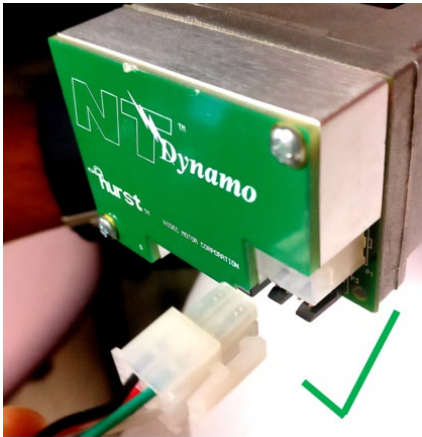
```
/*TODO1:Define the macro OPEN_LOOP_FUNCTIONING - Runs the motor  
in  
Open Loop Mode*/  
  
/* In this mode, the rotor angle is assumed and is not in sync  
with the actual rotor angle. This mode can be used to debug  
current loops and rotor position estimator*/  
  
#define OPEN_LOOP_FUNCTIONING
```

3. 编译与烧写

点击  按钮，编译 MPLAB X 工程。

点击  按钮，烧写程序。

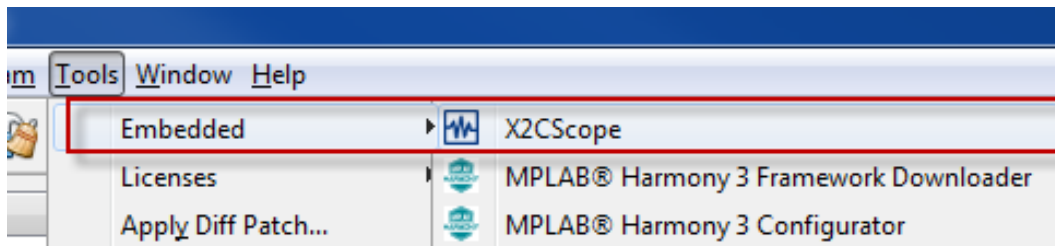
4. 连接配套电机端子



5. 按下开关 S2 启动电机运行

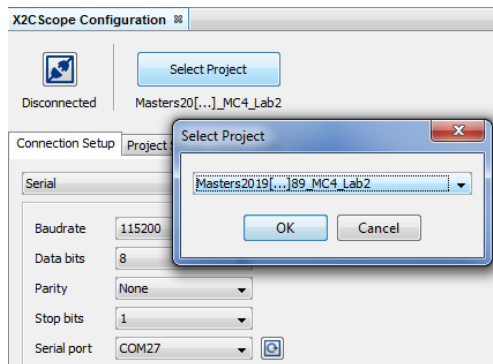
6. 观察运行结果


- 1) 由于基于反电动势的转子位置估算器无法在低速下正常运行，所以在位置估算器正常运行之前，需要先让电机在开环模式运行到一定速度，以保证位置估算器能可靠跟踪电机转子位置。
- 2) 在开环模式下，因没有准确的转子位置，电流无法全部施加在 Q 轴上，从而导致转矩输出下降。当你尝试抓住电机轴时，可能会导致电机堵转。
- 3) 可通过串口工具实时对比开环的转子位置和估算器输出的转子位置：
 - a. 打开插件 X2CScope: Tools → Embedded → X2Cscope

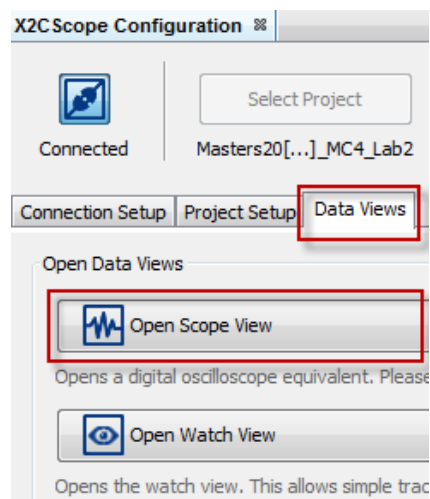


- b. 在跳出的 X2CScope Configuration 标签中，点击 Select Project，然后选择当前编译的工程。

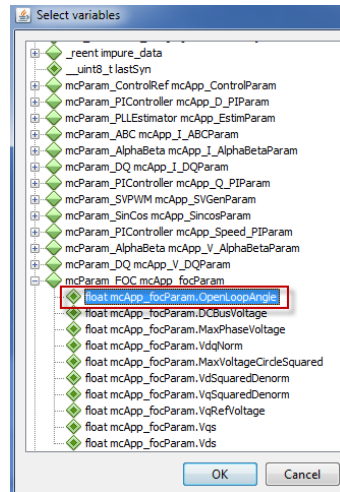
- c. 设置串口通讯波特率：115200bps，数据长度：8 位，停止位：1 位，选择正确的串口编号。



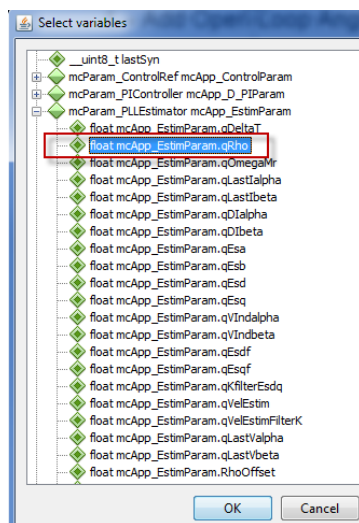
- d. 点击  按钮，建立串口连接
- e. 在“Data Views”标签下，点击“Open Watch View”按钮打开数据观察窗口。



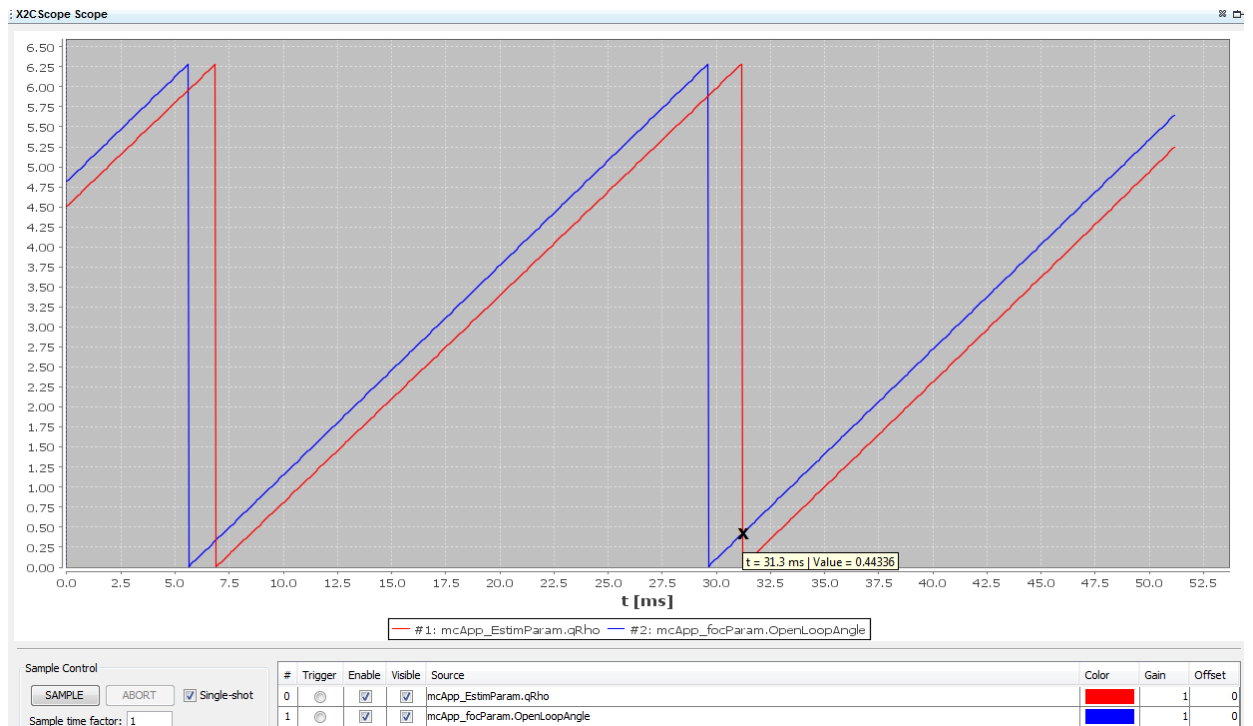
- f. 添加开环位置变量：mcApp_focParam.OpenLoopAngle



g. 添加估算位置变量: mcApp_EstimParam.qRho

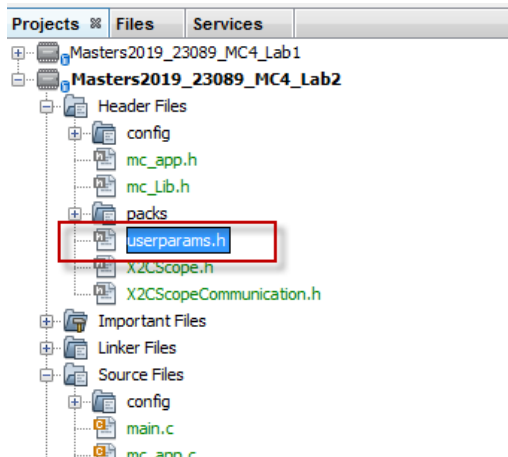


- h. 从下图可以看出，估算的转子位置滞后开环的转子位置大约 0.44rad 即 25° 。



四、 控制模式 2-转矩模式

1. 打开头文件 “userparams.h”



2. 在 “userparams.h” 文件中，取消“OPEN_LOOP_FUNCTIONING”的定义，同时定义“TORQUE_MODE”

```
/*TODO1:Define the macro OPEN_LOOP_FUNCTIONING - Runs the motor
in Open Loop Mode*/


/* In this mode, the rotor angle is assumed and is not in sync
with the actual rotor angle. This mode can be used to debug
current loops and rotor position estimator*/
#undef OPEN_LOOP_FUNCTIONING


/*TODO2: Define the macro TORQUE_MODE - Runs the motor in
Torque Mode*/

/*Define macro TORQUE_MODE to run the motor in TORQUE Control
Mode Undefine macro TORQUE_MODE to run the motor in Speed
Control Mode*/

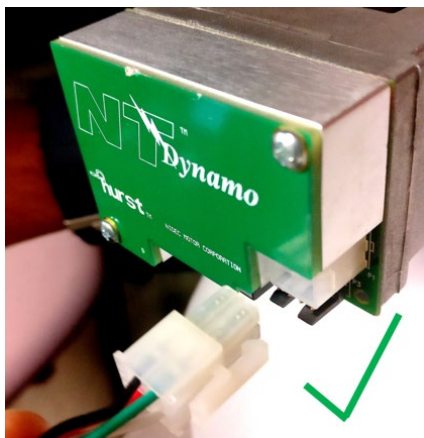
#define TORQUE_MODE
```

3. 编译与烧写

点击  按钮，编译 MPLAB X 工程。

点击  按钮，烧写程序。

4. 连接配套电机端子



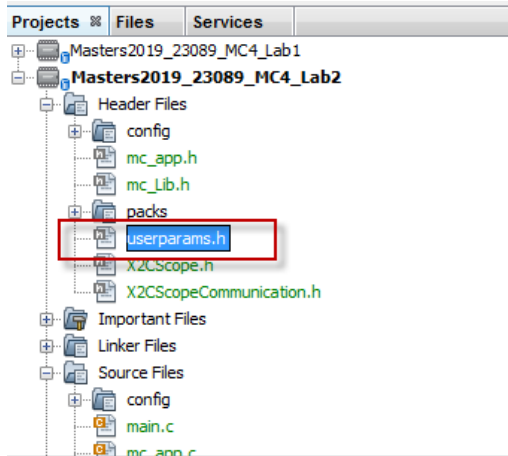
5. 按下开关 S2 启动电机运行

6. 观察运行结果

- 1) 在转矩模式下，转子位置从估算器获得，转矩指令由电位器设定；
- 2) 电机转速不被直接控制，而是随着电机负载的变化而变化。当负载增大时，转速随之下降，当负载减小时，转速随之增大；

五、 控制模式 3-速度模式

1. 打开头文件 “userparams.h”



2. 在 “userparams.h” 文件中，取消“OPEN_LOOP_FUNCTIONING”和“TORQUE_MODE”的定义

```
/*TODO1:Define the macro OPEN_LOOP_FUNCTIONING - Runs the motor
in Open Loop Mode*/


/* In this mode, the rotor angle is assumed and is not in sync
with the actual rotor angle. This mode can be used to debug
current loops and rotor position estimator*/
#undef OPEN_LOOP_FUNCTIONING


/*TODO2: Define the macro TORQUE_MODE - Runs the motor in
Torque Mode*/

/*Define macro TORQUE_MODE to run the motor in TORQUE Control
Mode Undefine macro TORQUE_MODE to run the motor in Speed
Control Mode*/

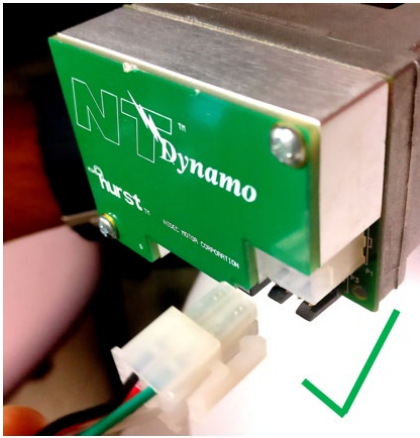
#undef TORQUE_MODE
```


3. 编译与烧写

点击  按钮，编译 MPLAB X 工程。

点击  按钮，烧写程序。

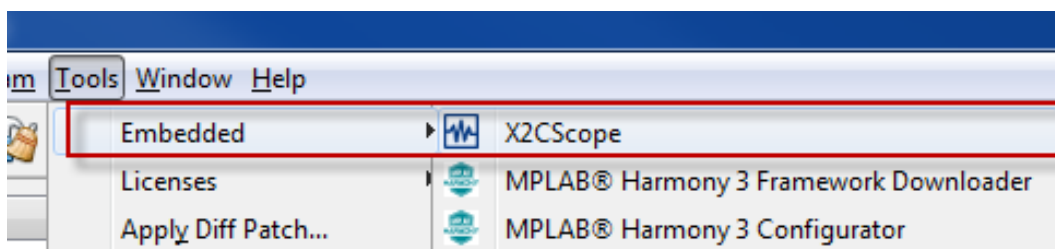
4. 连接配套电机端子



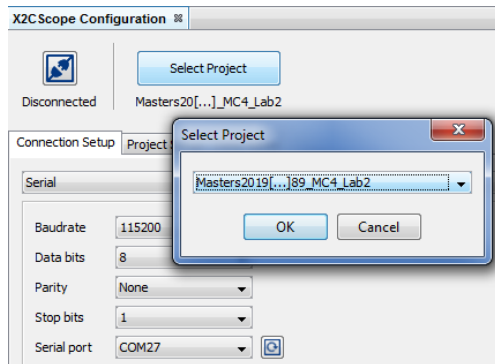
5. 按下开关 S2 启动电机运行


6. 观察运行结果

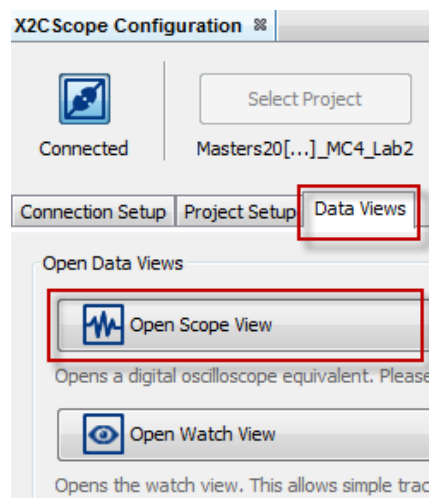
- 1) 在速度模式下，转子位置从估算器获得，转矩指令由速度控制器给定，速度指令由电位器设定。
 - 2) 当电机负载变化时，速度控制器会尝试维持电机速度在设定值。
 - 3) 我们可以通过观察 D 轴的反电动势来判断估算的转子位置是否准确，当估算的转子位置接近实际的转子位置时，D 轴的反电动势值也接近 0:
- a. 打开插件 X2CScope: Tools → Embedded → X2CScope



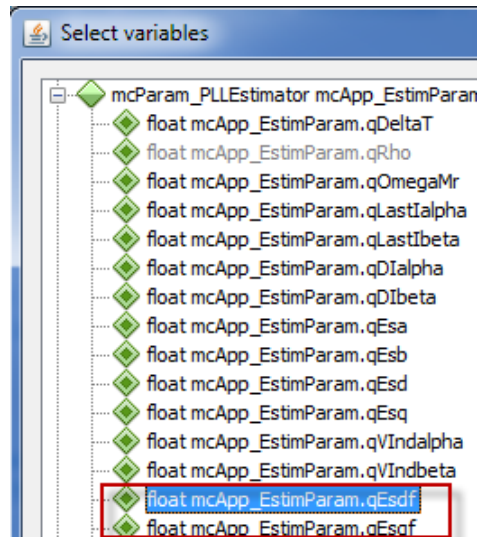
- b. 在跳出的 X2CScope Configuration 标签中，点击 Select Project，然后选择当前编译的工程。
- c. 设置串口通讯波特率：115200bps，数据长度：8 位，停止位：1 位，选择正确的串口编号。



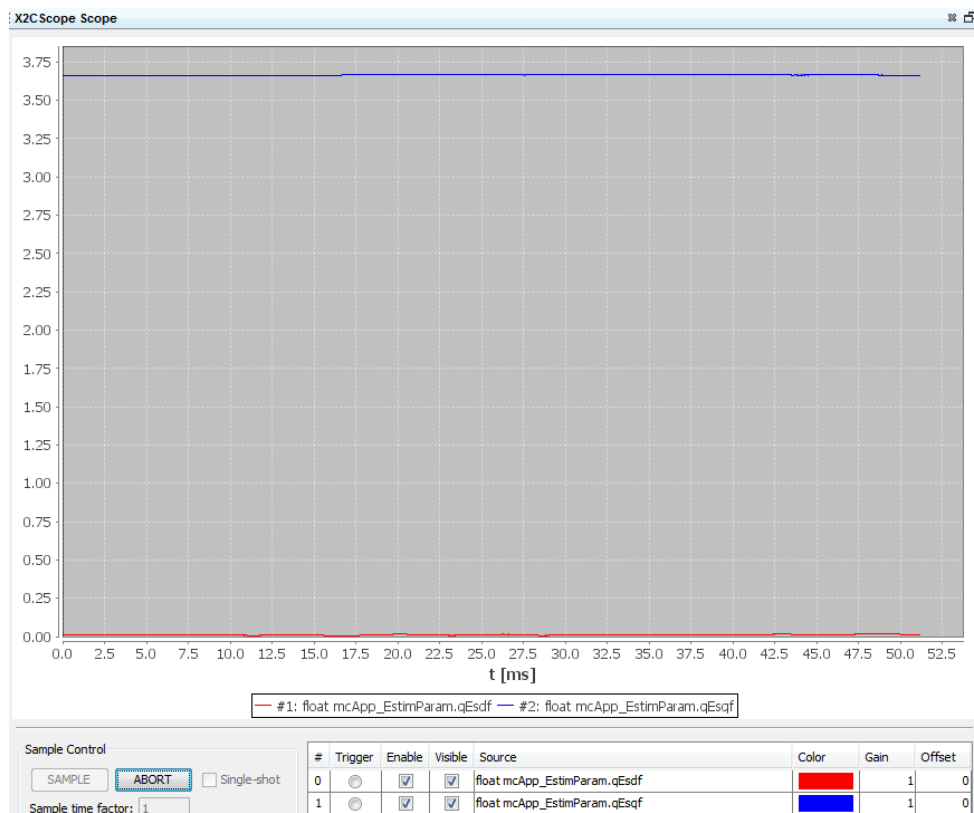
- d. 点击  按钮，建立串口连接
- e. 在“Data Views”标签下，点击“Open Watch View”按钮打开数据观察窗口。



- f. 添加 DQ 轴的反电动势变量：mcApp_EstimParam.qEsdf, mcApp_EstimParam.qEsqf



- g. 如下图，D 轴的反电动势接近 0，而 Q 轴反电动势则随着电机转速的变化在 0~13.85V 之间变化；



六、 总结

本文演示了如何使用 MPLAB Harmony 3 基于 SAME54 及例程进行无位置传感器 PMSM 控制；并且在 MLCV2 电机演示平台上验证了控制效果。

本文所使用的例程是针对实验而制作：如第三章 ~ 第五章所述，需要读者自行修改或增加部分代码，以实现电机在各种模式下的运行。在 MPLAB Harmony 3 中，有相应的非实验版本例程：

...\motor_control\apps\pmsm_foc_pll_estimator_sam_e54

读者参考了本文和上述两个例程后，可以利用 MPLAB Harmony 3 快速进行电机控制应用的开发。

七、 附件

本文档配套 MPLAB X 工程 Masters2019_23089_MC4_Lab2.X

如果 Adobe Reader 禁止用户下载附件，可以使用火狐浏览器或者福昕阅读器来下载附件。