MPLAB® Harmony 3 之基础篇（08）

-- 利用 Harmony 3 助力电机应用开发

（如何配置 SAME70 以用于电机控制）

Microchip Technology Inc.

MCU32 产品部

一、 概述

本文将会详细列出利用 Harmony 3 配置 SAME70 的步骤，例如：系统时钟、中断、AFEC、PWM、QEI、UART、IO 引脚。在简单的配置/自动代码生成之后，电机控制所需的 MCU 初始化程序、硬件抽象层、硬件驱动程序就已经完成。

根据本文的配置：

AFEC 采样由 PWM 模块触发；所有模拟通道都转换完成后，生成 AFEC 中断请求；并且双采保模式可以实现两路模拟信号的同时采样。

PWM 模块输出三路互补 PWM 信号。

UART 模块用于支持 X2CScope 通讯。X2CScope 是 MPLAB X 中的串口调试插件；利用它可以便捷地追踪及修改程序中的所有全局变量，实现软件示波器、数值实时现实等功能。

(一) 基础知识

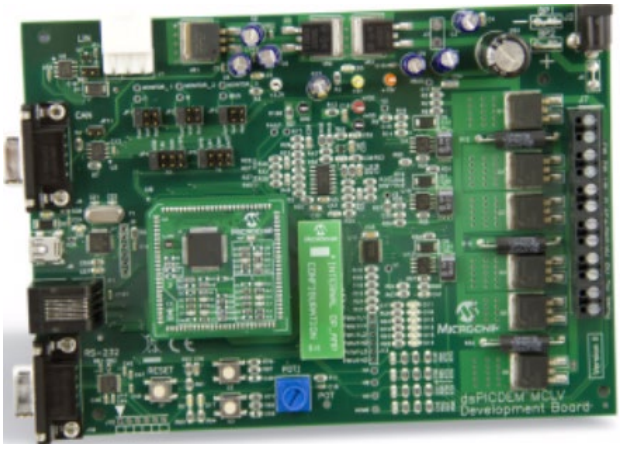
为了更好地理解本文档内容，建议具备以下知识：

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. SAME70 微控制器的基础知识

(二) 所需工具

硬件：

1. MCLV-2 低压电机控制印板

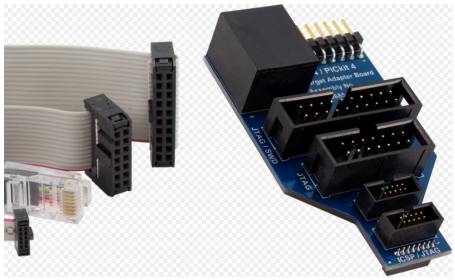


2. SAME70 PIM （plug-in-module）

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: v5.00 或更新版本
2. MPLAB XC32 编译器: v2.10 或更新版本
3. MPLAB Harmony 3 Configurator (MHC) v3.1.0.4 或更新版本
4. X2CScope 串口调试工具：v2.0.6

注意

本文内容不含有电机控制的算法实现，因此在实践本文内容时，**不应连接电机动力线**。

二、 系统时钟和 IO 引脚

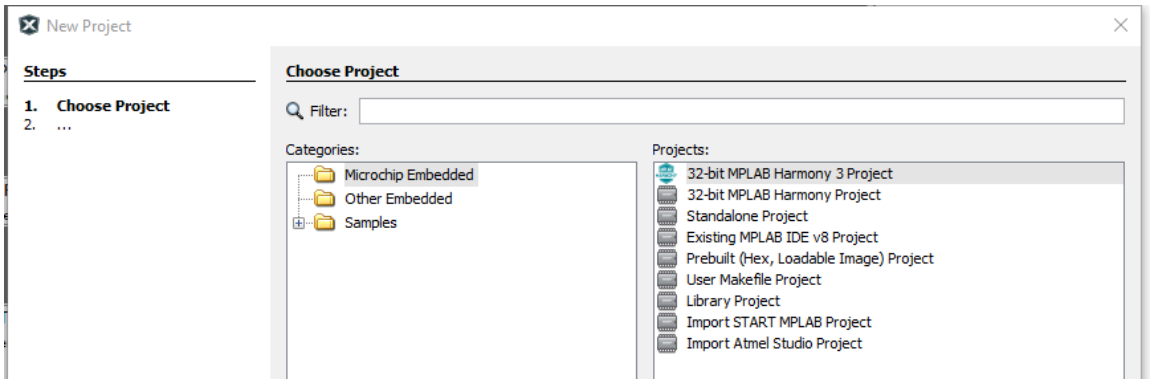
(一) 概述

本章将指导您在 MPLAB X IDE 中新建一个 Harmony 3 项目，并使用 MHC 来配置系统时钟和 IO 引脚。

(二) 步骤 1：新建 Harmony 3 项目

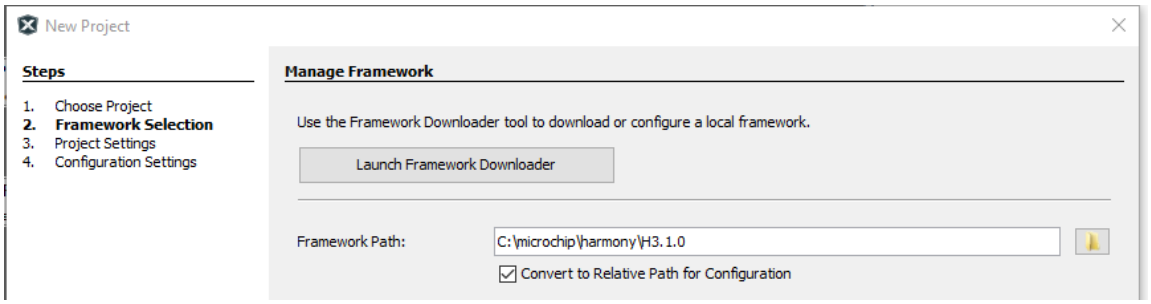
1. File → New Project。

在 New Project 对话框中，选择 Microchip Embedded 和 32-bit MPLAB Harmony 3 Project，然后点击 Next。



2. 设定 Harmony 3 路径。

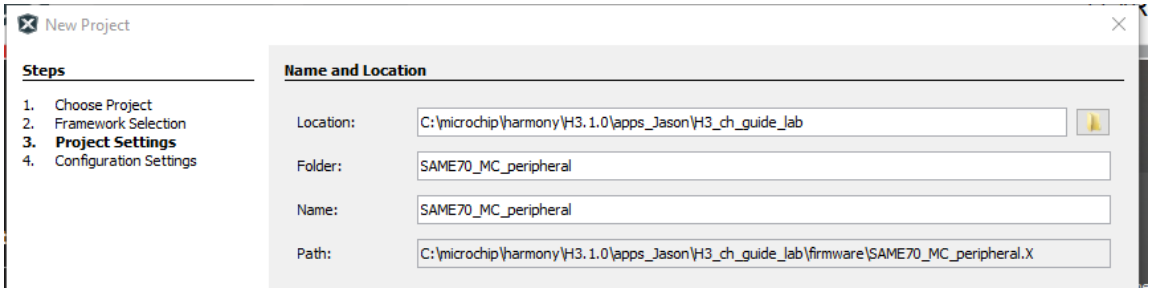
输入 Harmony 3 的本地路径，然后点击 Next。



3. 设定项目信息。

设定项目所在路径（location）、项目信息文件夹名称（folder）、项目名称（name），然后点击 Next。

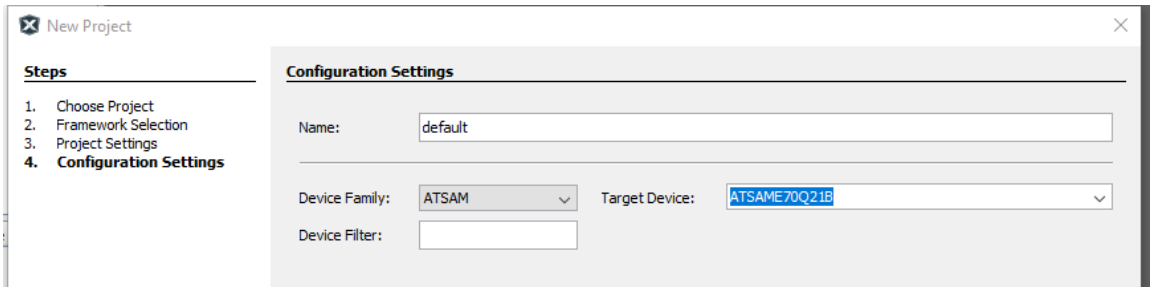
其中，项目信息文件夹是一个名称以.X 结尾的文件夹，用于存储 Harmony 3 项目的各类信息。



4. 设定配置信息

设定 Harmony 3 配置文件的名称、元器件型号，然后点击 Finish。

项目会被自动建立，并且 MHC 会被开启。



(三) 步骤 2：配置系统时钟

1. 如果 MHC 没有自动开启，可以手动开启。

Tools → Embedded → MPLAB Harmony 3 Configurator

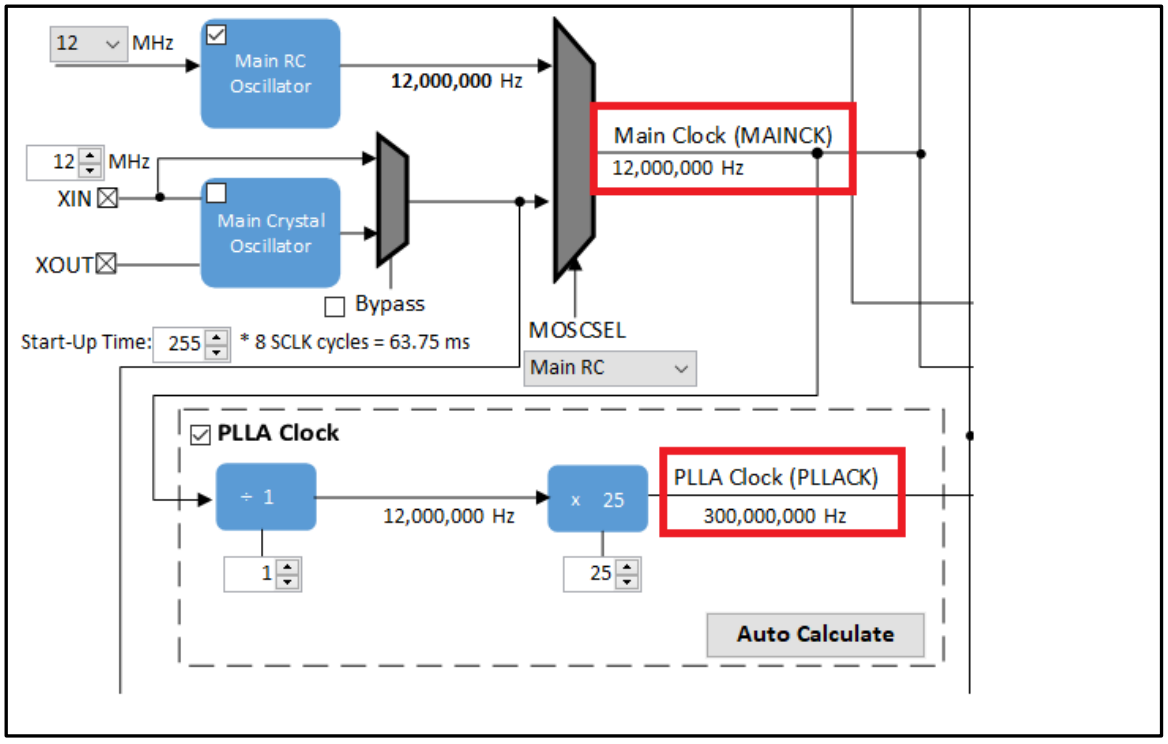
2. 打开时钟配置工具。

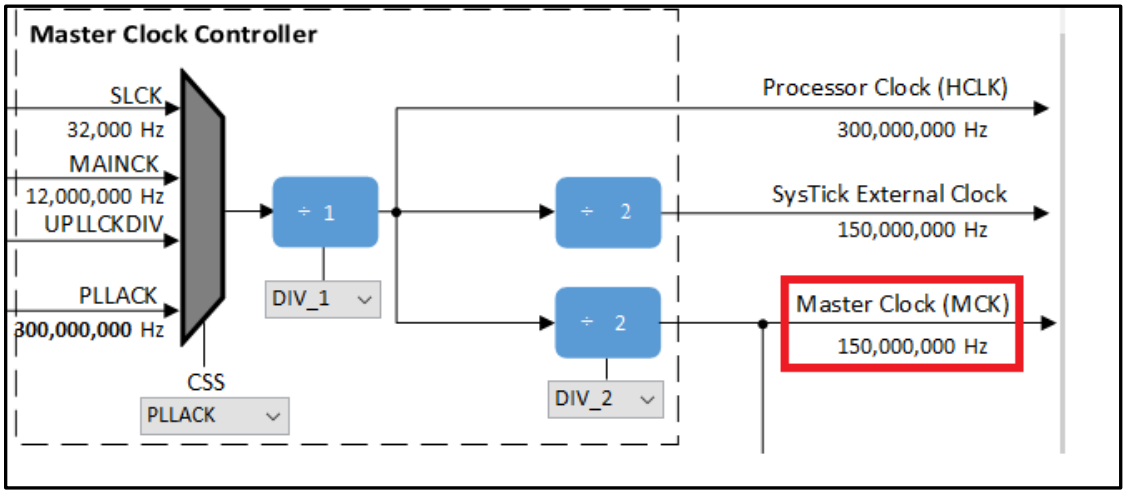
MHC → Tools → Clock Configuration

3. 配置系统时钟

CPU 时钟已经默认配置为 300MHz，无需更改；但建议确认：

1) Processor Clock = 300,000,000Hz
2) Master Clock = 150,000,000Hz





(四) 步骤 3：配置 IO 引脚

1. 打开引脚配置工具。

MHC → Tools → Pin Configuration

2. 点击 pin settings 标签。

将引脚排列顺序（Order）选择为：Ports。

3. 设定 LED 驱动引脚。

将 PC23 引脚的功能（Function）设定为：GPIO；

将其名称（Custom Name）设定为：LED；

将其方向设定为（Direction）：Out；

将其默认电平（Latch）设定为：Low。

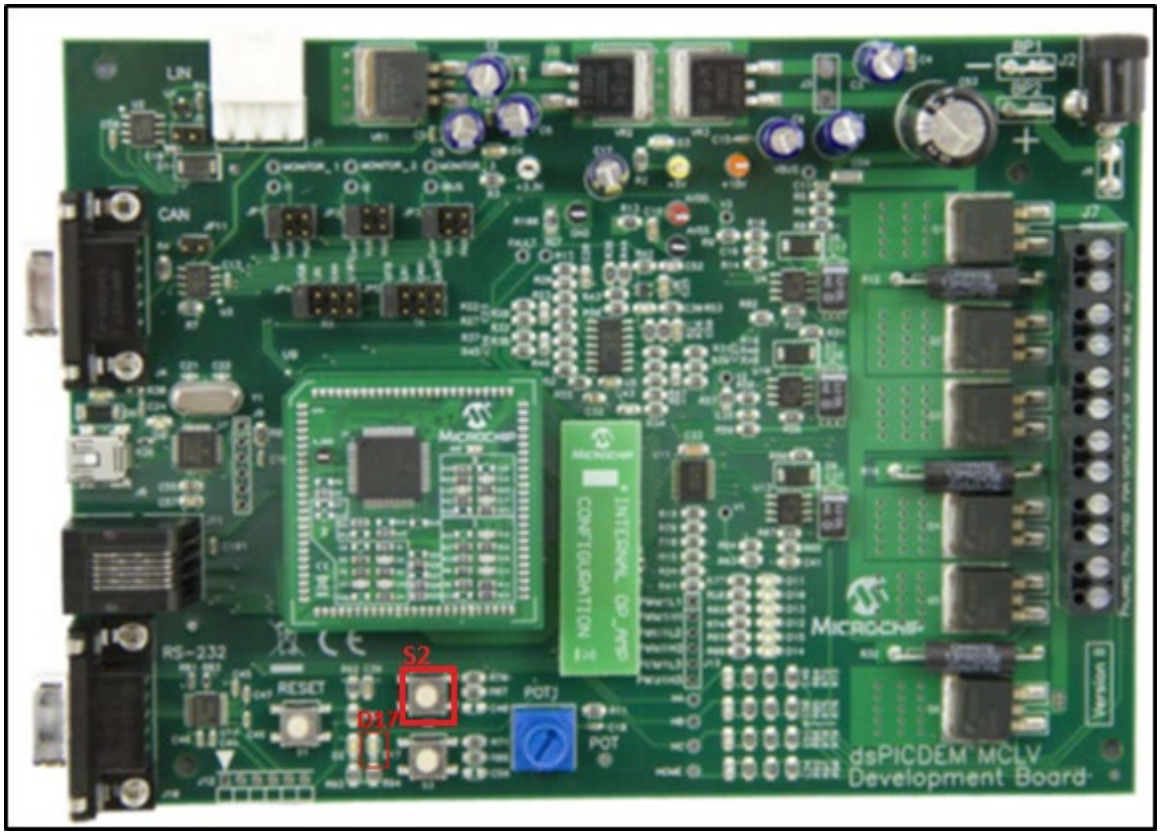
122	PC21		Available	In	n/a	☐	Dig...	Disabled	☐
124	PC22		Available	In	n/a	☐	Dig...	Disabled	☐
127	PC23	LED	GPIO	Out	L...	☐	Dig...	Disabled	☐
130	PC24		Available	In	n/a	☐	Dig...	Disabled	☐
133	PC25		Available	In	n/a	☐	Dig...	Disabled	☐

4. 设定按钮信号读取引脚。

将 PC3 引脚的功能（Function）设定为：GPIO；

将其名称（Custom Name）设定为：switch2；

将其方向设定为（Direction）：In。



(五) 步骤 4（可选）：生成代码

1. 保存 Harmony 3 设置信息。

MHC → Save State

2. 生成代码。

MHC → Generate Code

在 Generate Project 对话框中，点击 Generate。

3. 编译本项目。

Production → Build Main Project

确认编译成功。

三、 PWM 模块

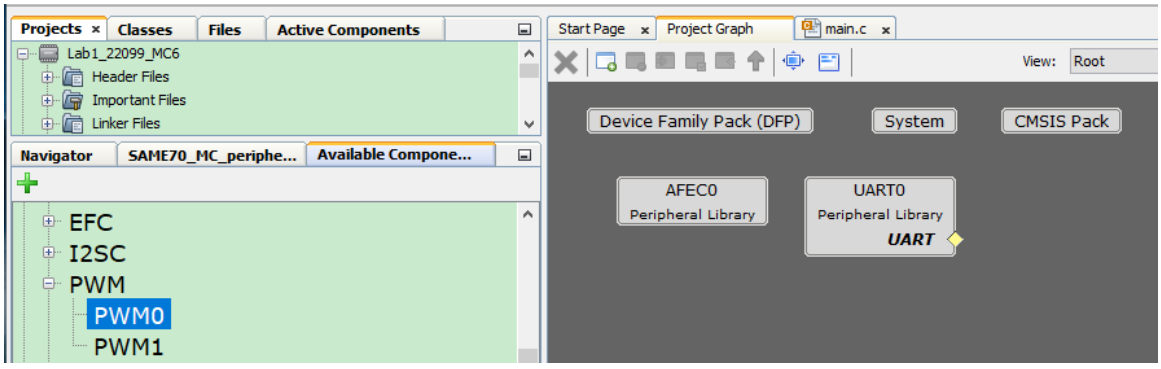
(一) 概述

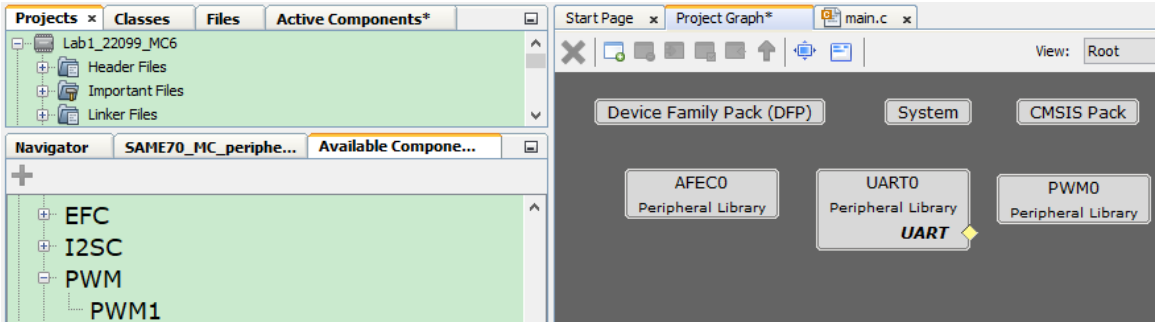
本章将指导您使用 MHC 来配置 PWM 模块，用于生成三对互补 PWM 波形输出。

此外，PWM 计数器会在特定时刻生成事件，用于触发 AFEC 采样。

(二) 步骤 1：配置 PWM 模块（PWM 波形输出）

1. 将 PWM0 拖拽到 Harmony 3 项目框图中。





2. 点击 PWM0 方框，然后在 Configuration Options 标签下进行设置。
3. 使能 Channel0、Channel1 和 Channel2。

勾选这三个通道的 Enable 选项。

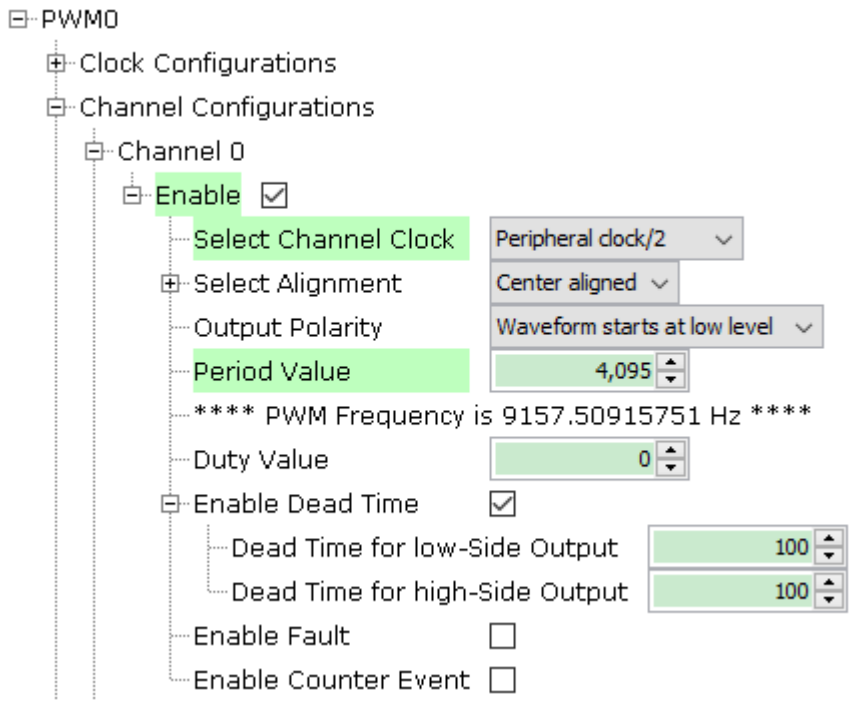
确认每个通道的 Enable Dead Time 选项已经勾选。

输入上下桥臂的死区计数值。

4. 通道 0 设置。

根据您的实际情况，对以下项目进行适当的设置：计数器时钟源（Select Channel Clock）、PWM 周期计数值（Period Value）、计数对齐方式（Alignment）、占空比更新时刻（Duty-cycle Update Trigger）、PWM 极性（Output Polarity）、初始占空比（Duty Value）……

本文的设置如下：

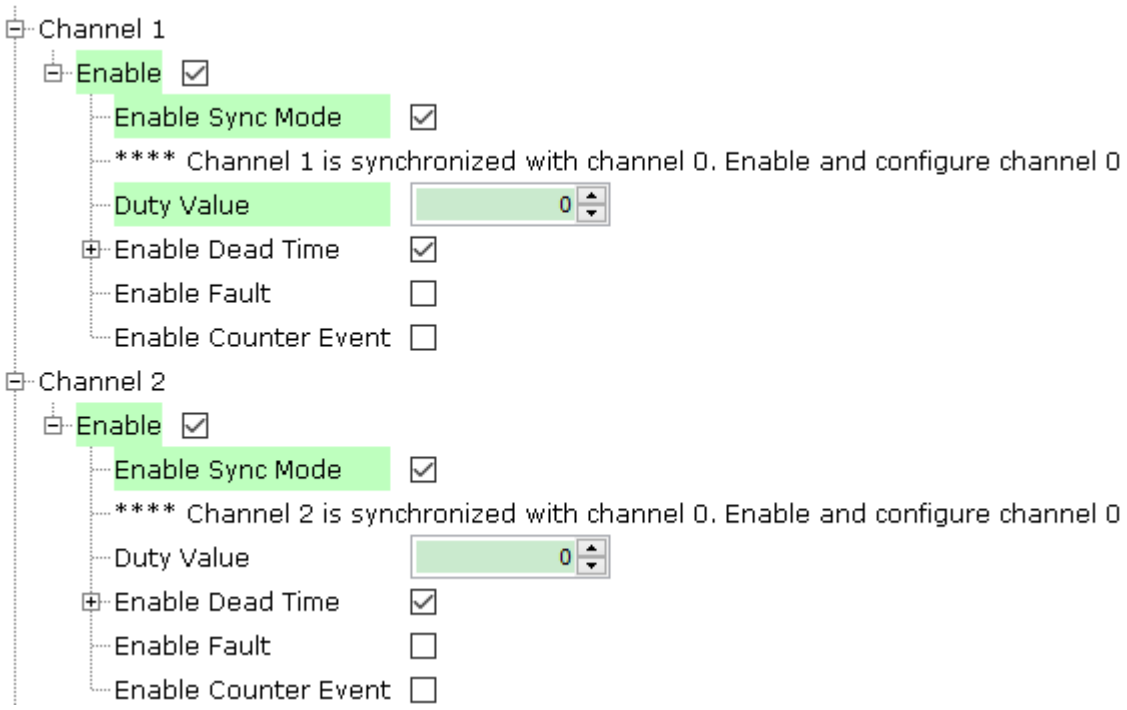


5. 同步模式设置

在 Channel1 和 Channel2 中，勾选 Enable Sync Mode；

同步模式被启用后，通道 1 和通道 2 会追随通道 0 的时钟、计数对齐方式、PWM 输出极性和 PWM 周期计数值等配置；

当然每个通道的初始占空比是需要独立设置的。



(三) 步骤 2：配置 PWM 模块（AFEC 触发事件生成）

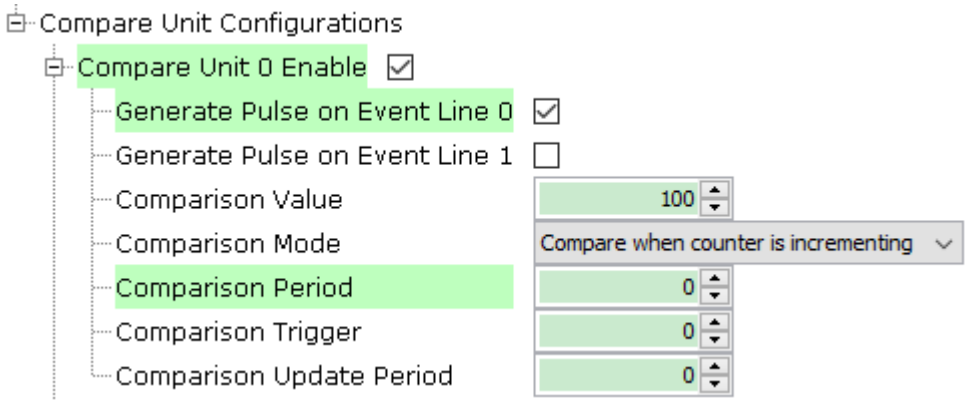
1. 使能比较单元 0。

勾选 Compare Unit 0 Enable。

2. 选择事件线（Event Line）、比较值（Comparison Value）、比较模式（Comparison mode）。

按照本文的配置，每当 PWM 计数值上升到 100 时，在事件线 0 上会生成一个比较事件；

在后面的 AFEC 设置中，我们将把这个比较事件用来触发 AFEC 采样。



(四) 步骤 3：配置 PWM 引脚

1. 打开引脚配置工具。

MHC → Tools → Pin Configuration

2. 点击 pin settings 标签。

将引脚排列顺序（Order）选择为：Ports。

3. 设定 PWM 引脚。

1) Pin ID: PA11. Function: PWM0_PWMH0
2) Pin ID: PA12. Function: PWM0_PWMH1
3) Pin ID: PA13. Function: PWM0_PWMH2
4) Pin ID: PD24. Function: PWM0_PWML0
5) Pin ID: PD25. Function: PWM0_PWML1
6) Pin ID: PD26. Function: PWM0_PWML2

(五) 步骤 4（可选）：生成代码

1. 保存 Harmony 3 配置。

MHC → Save State

2. 生成代码。

MHC → Generate Code

3. 编译本项目。

Production → Build Main Project

确认编译成功。

四、 AFEC

(一) 概述

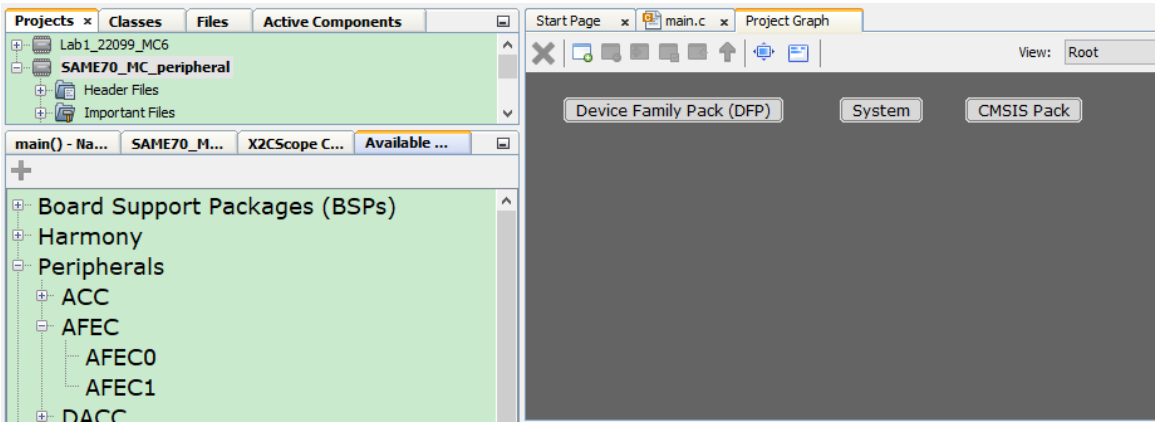
本章将指导您使用 MHC 来配置 AFEC。

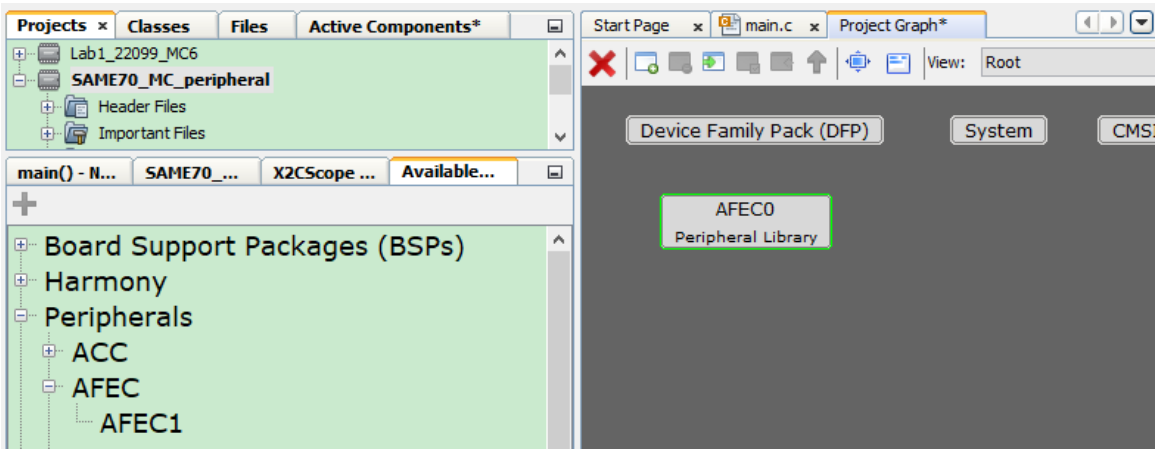
AFEC 用于转换 U 相、V 相电流和电位器所输出的模拟量；我们会利用双采保模式来实现 U 相和 V 相电流信号的同时采样。

AFEC 的采样将配置成由 PWM 模块触发。

(二) 步骤 1：配置 AFEC

1. 将 AFEC0 拖拽到 Harmony 3 项目框图中。





2. MHC → Tools → AFEC 0 Configuration

AFEC0 Easy View 标签会自动打开。

3. 设定转换模式（Conversion Mode）。

为了实现由 PWM 模块触发 AFEC 采样，应将转换模式设定为：Hardware Trigger。

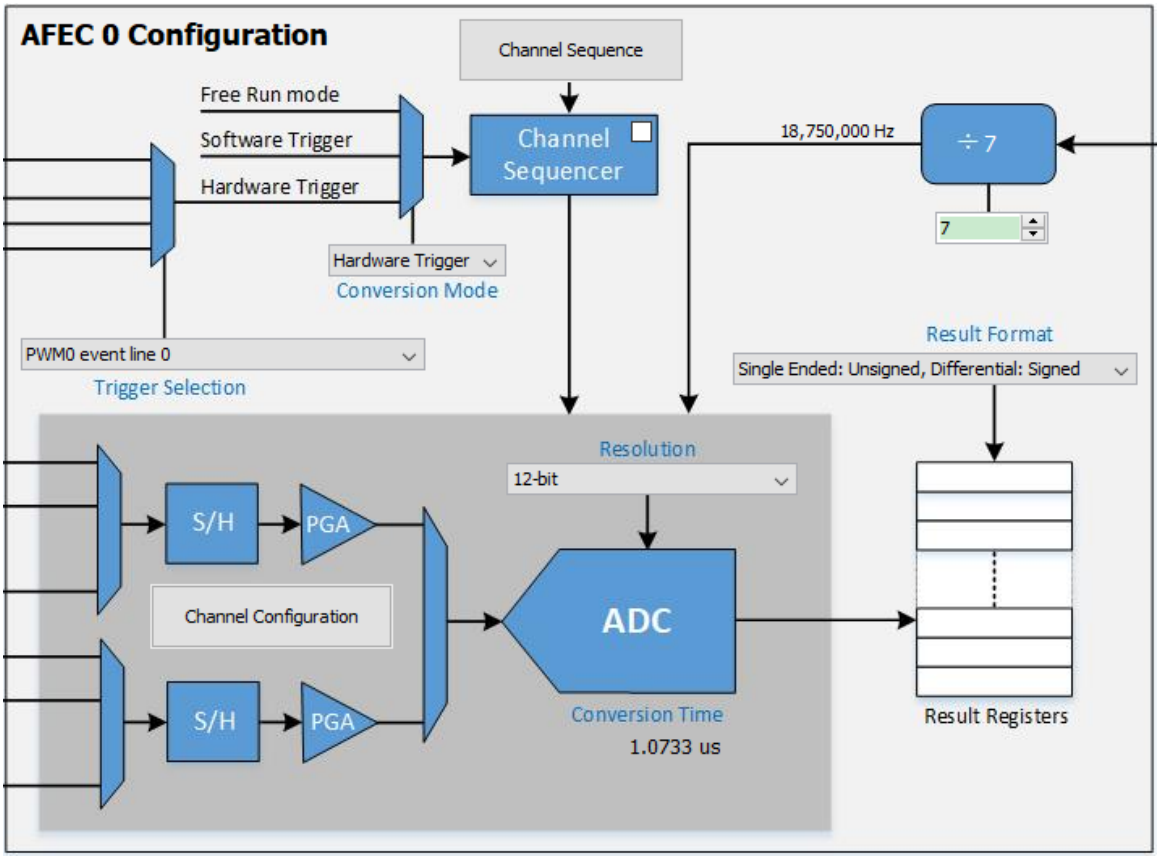
4. 设定触发源（Trigger Selection）。

将触发源设定为：PWM0 event line 0。

5. 设定 AFEC 时钟频率。

根据您的需要选择预分频倍数，并确认所显示的时钟频率如您所愿；

本文将预分频倍数设定为：7。



6. 设定模拟通道。

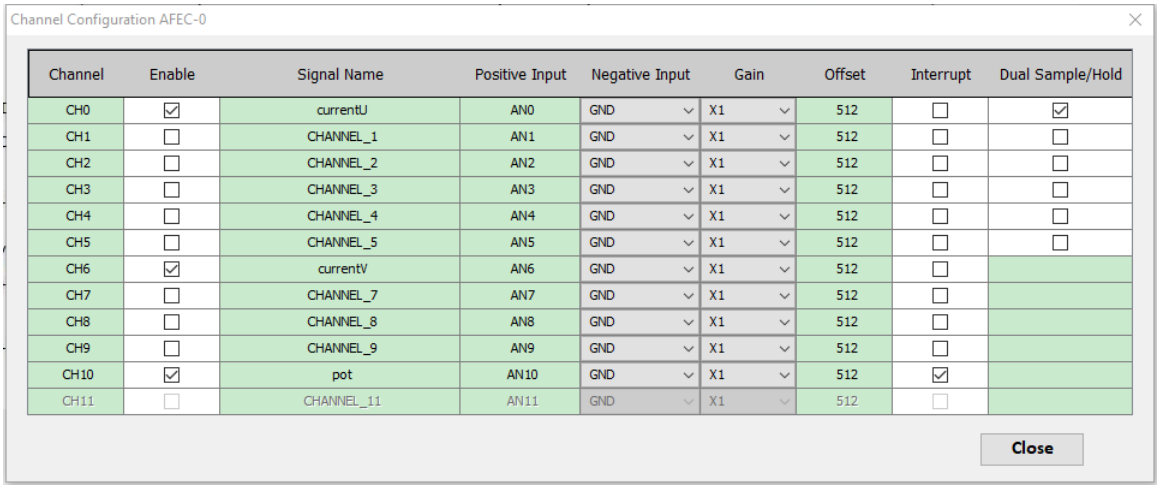
点击 Channel Configuration；通道配置窗口会自动弹出；

使能（Enable）通道 0、通道 6、通道 10；

为以上通道指定信号名称（Signal Name）；

勾选通道 0 的双采保模式（Dual Sample/Hold）；通道 0 和通道 6 会被同时采样；

勾选通道 10 的中断（Interrupt）；在通道 10 信号转换完成后，会立即生成中断请求。



(三) 步骤 2：配置 AFEC 引脚

1. 打开引脚配置工具。

MHC → Tools → Pin Configuration

2. 点击 pin settings 标签。

将引脚排列顺序（Order）选择为：Ports。

3. 设定 AFEC 引脚。

1) Pin ID: **PD30**. Function: AFEC0_AD0
2) Pin ID: **PA17**. Function: AFEC0_AD6
3) Pin ID: **PB0**. Function: AFEC0_AD10

(四) 步骤 3（可选）：生成代码

1. 保存 Harmony 3 配置。

MHC → Save State

2. 生成代码。

MHC → Generate Code

3. 编译本项目。

Production → Build Main Project

确认编译成功。

五、 正交编码接口

(一) 概述

本章将指导您使用 MHC 来配置定时器（TC）模块，用于解码正交信号。

(二) 步骤 1：配置 TC

1. 将 TC0 拖拽到 Harmony 3 项目框图中。
2. 点击 TC0，然后在 Configuration Options 标签下进行设置。
3. 将 TC0 配置成为正交编码接口模式。

勾选 Enable Quadrature Encoder Mode 选项；

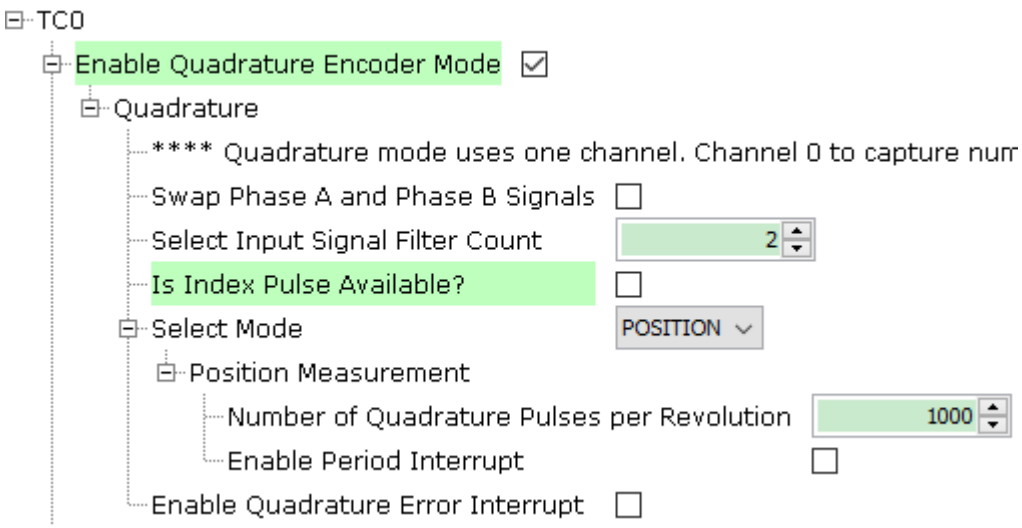
取消勾选 Is Index Pulse Available；

4. 设定计数周期值

将 Number of Quadrature Pulses Per Revolution 设定为 1000；

需要说明的是，该值应设定为多少，是由编码器特性和您的应用情况决定的。

例如本文使用电机（DMA0204024B101）所配套的编码器；其每个机械圈输出 250 个 A 相和 B 相脉冲。于是，一个机械圈的正交信号计数值为 1000。并且，我们希望以机械圈为计数周期。所以本文将该数值设定为 1000。



(三) 步骤 2：配置 TC 引脚

1. 打开引脚配置工具。

MHC → Tools → Pin Configuration

2. 点击 pin settings 标签。

将引脚排列顺序（Order）选择为：Ports。

3. 设定 TC 引脚。

1) Pin ID: PA0. Function: TC0_TIOA0
2) Pin ID: PA1. Function: TC0_TIOB0

(四) 步骤 3（可选）：生成代码

1. 保存 Harmony 3 配置。

MHC → Save State

2. 生成代码。

MHC → Generate Code

3. 编译本项目。

Production → Build Main Project

确认编译成功。

六、 UART 和 X2CScope

(一) 概述

本章将指导您使用 MHC 来配置 UART；添加 X2CScope 文件和代码。

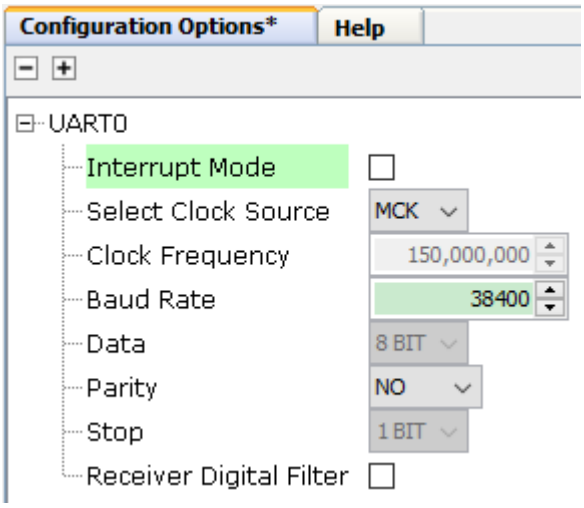
UART 模块用于支持 X2CScope 通讯。

(二) 步骤 1：配置 UART

1. 将 UART0 拖拽到 Harmony 3 项目框图中。
2. 点击 UART0 方框。
3. 在 Configuration Options 标签下进行配置。

取消勾选 Interrupt Mode；

设定波特率为 38400。



(三) 步骤 2：配置 UART 引脚

1. 打开引脚配置工具。

MHC → Tools → Pin Configuration

2. 点击 pin settings 标签。

将引脚排列顺序（Order）选择为：Ports。

3. 设定 AFEC 引脚。

1) Pin ID: **PA9**. Function: UART0_URXD0
2) Pin ID: **PA10**. Function: UART0_UTXD0

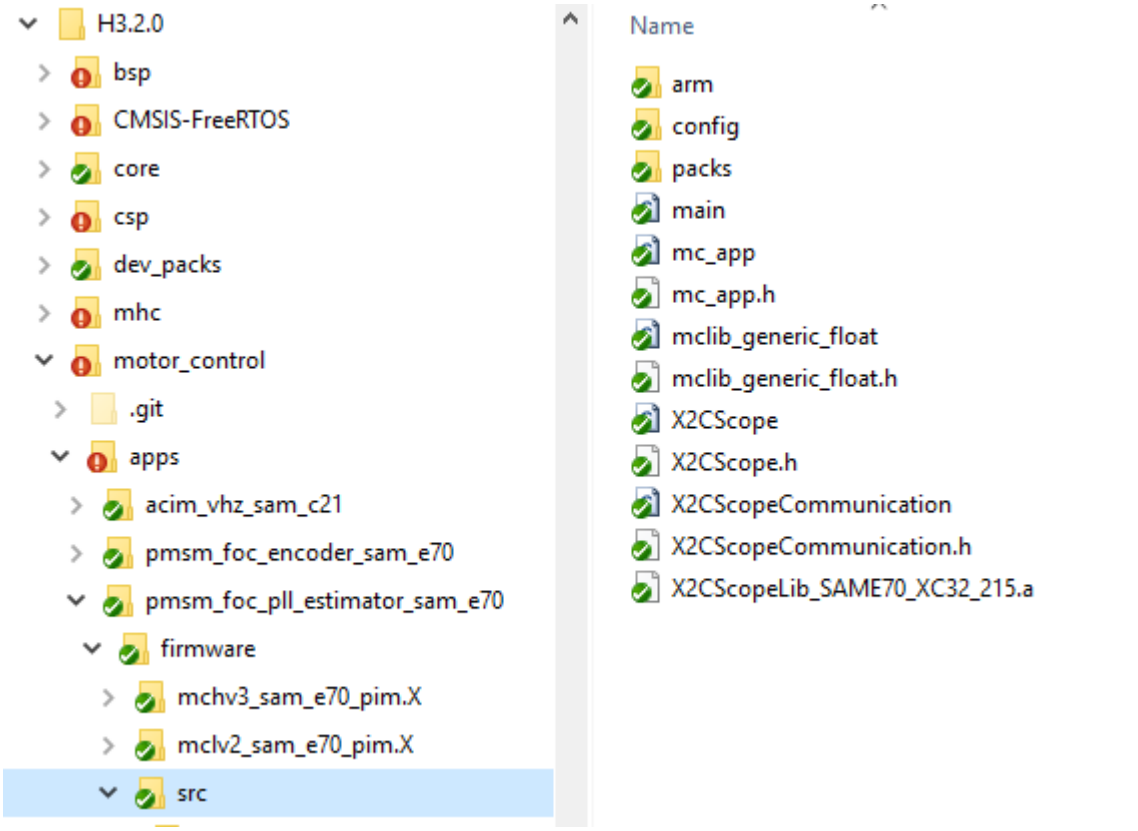
(四) 步骤 3：添加 X2CScope 文件

1. 获取 X2CScope 所需文件。

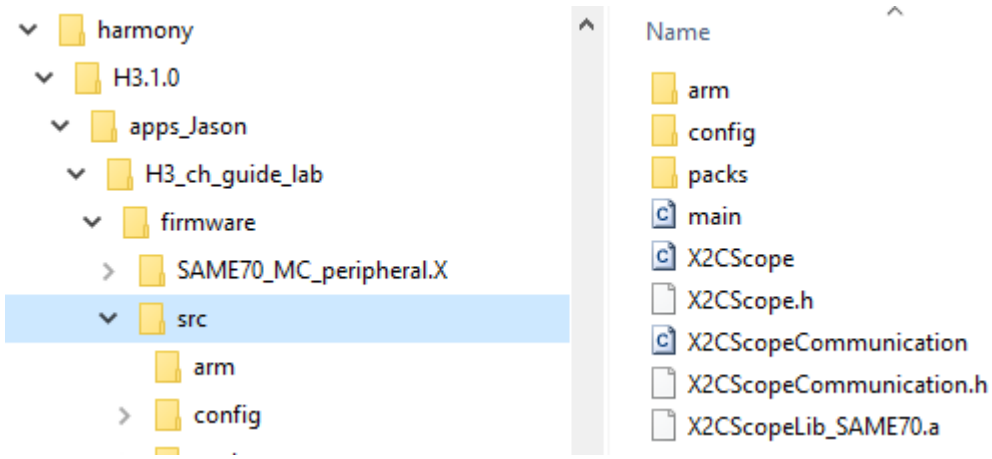
所需文件为：

1) X2CScope 库文件：X2CScopeLib_SAME70.a
2) X2CScope 源文件和头文件：X2CScopeCommunication.c、X2CScopeCommunication .h、X2CScope.c、X2CScope.h

上述库文件、源文件和头文件可以在 Harmony 3 SAME70 电机例程中找到：



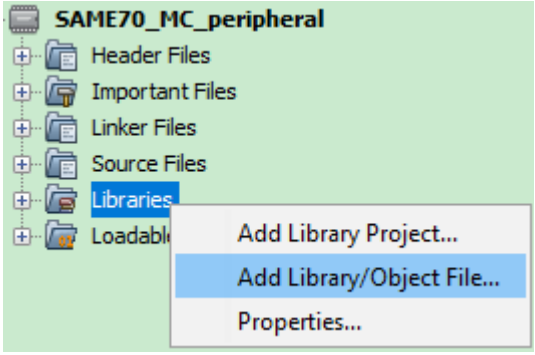
将上述库文件、源文件和头文件复制到本项目源文件所处的文件夹中：



2. 添加 X2CScope 库文件。

在 Projects 标签中，右键点击 Libraries → Add Library/Object File…

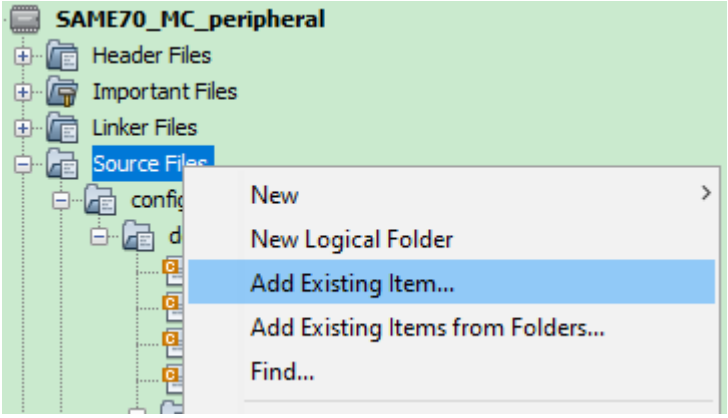
浏览并选择相应的 X2CScope 库文件：X2CScopeLib_SAME70.a。



3. 添加 X2CScope 源文件。

右键点击 Source Files → Add Existing Item…

浏览并选择刚才复制的两个.c 文件，以将其添加到本工程源文件列表中。



4. 添加 X2CScope 头文件。

右键点击 Header Files → Add Existing Item…

浏览并选择刚才复制的两个.h 文件，以将其添加到本工程头文件列表中。

(五) 步骤 4：添加 X2CScope 代码

1. 说明

该步骤中的内容可在最后统一实施；为方便查询，此处单独列出；仅做说明用。

2. 包含头文件

在 main.c 文件中，包含 X2CScope 头文件。

```
#include "X2CScope.h"

#include
"X2CScopeCommunication.h"
```

3. 初始化 X2CScope

在 main()函数中，在 SYS_Initialize()函数语句之后，调用 X2CScope_Init()函数。

```
X2CScope_Init();
```

4. 在 while(1)循环中调用 X2CScope_Communicate()函数。

```
X2CScope_Communicate();
```

```
X2CScope_Update();
```

5. 在 AFEC0 中断回调函数中调用 X2CScope_Update()函数。

(六) 步骤 5（可选）：生成代码

1. 保存 Harmony 3 配置。

MHC → Save State

2. 生成代码。

MHC → Generate Code

3. 编译本项目。

Production → Build Main Project

确认编译成功。

七、 添加代码

(一) 概述

本章将指导您在自动生成代码的基础上，添加必要的代码，用于调用或测试 PWM、AFEC、TC、UART 的相关功能。

(二) 步骤 1：生成代码

1. 保存 Harmony 3 配置。

MHC → Save State

2. 生成代码。

MHC → Generate Code

3. 编译本项目。

Production → Build Main Project

确认编译成功。

(三) 步骤 2：添加全局变量

在 main.c 文件中声明以下全局变量：

```
uint16_t adc_result;

uint32_t rotor_position;
```

(四) 步骤 3：添加 AFEC 相关代码

1. 注册 AFEC 中断回调函数

在 main()函数中，SYS_Initialize()函数之后，添加以下代码，以注册 AFEC 中断回调函数：

AFEC0_CallbackRegister(AFEC0_Callback, (uintptr_t)NULL);

其中，AFEC0_Callback 是我们要注册的函数名。

2. 添加 AFEC 中断回调函数

在 main()函数之前，添加以下函数定义：

```
void AFEC0_Callback (uintptr_t context)
{
        adc_result = AFEC0_ChannelResultGet(pot);
        rotor_position = TC0_QuadraturePositionGet();
        PWM0_ChannelDutySet(PWM_CHANNEL_0, adc_result);
        PWM0_ChannelDutySet(PWM_CHANNEL_1, adc_result);
        PWM0_ChannelDutySet(PWM_CHANNEL_2, adc_result);
        X2CScope_Update();
}
```

(五) 步骤 4：添加 PWM 相关代码

在 main()函数中，在进入 while(1)循环之前，用以下代码启动 PWM 计数器和三对波形输出：

PWM0_ChannelsStart(PWM_CHANNEL_0_MASK);

由于使能了同步模式，所以我们只需要启动通道 0 即可；通道 1 和通道 2 会同步启动。

(六) 步骤 5：添加 TC 相关代码

在 main()函数中，在进入 while(1)循环之前，用以下代码启动对正交编码信号的计数：

TC0_QuadratureStart();

调用该函数后，正交编码计数值会被清零，并开始计数。

(七) 步骤 6：添加 X2CScope 相关代码

按照第五章、步骤 4 所述内容执行。

```
int main ( void )
{
    /* Initialize all modules */
    SYS_Initialize ( NULL );
    X2CScope_Init();
    TC0_QuadratureStart();
    AFEC0_CallbackRegister(AFEC0_Callback, (uintptr_t)NULL);
    PWM0_ChannelsStart(PWM_CHANNEL_0_MASK);

    while ( true )
    {
        /* Maintain state machines of all polled MPLAB Harmony modules. */
        SYS_Tasks ( );
        X2CScope_Communicate();
    }

    /* Execution should not come here during normal operation */

    return ( EXIT_FAILURE );
}
```

八、 功能测试

(一) 概述

本章将指导您利用以上创建的项目，测试 AFEC、PWM、正交编码接口等功能。

注意

本文内容不含有电机控制的算法实现，因此在实践本文内容时，**不应连接电机动力线**。

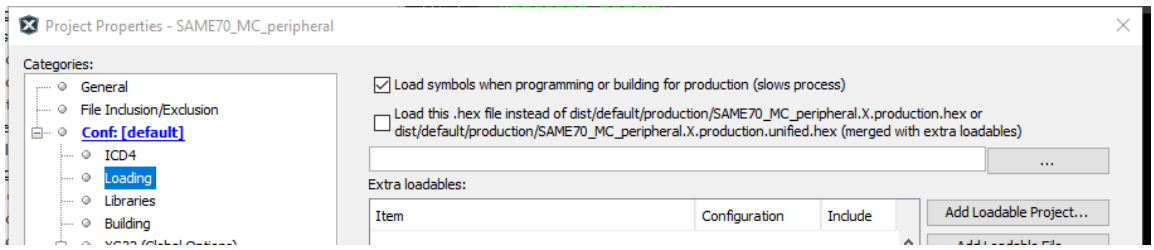
(二) 步骤 1：编译项目

1. 打开项目属性设置界面

在 Projects 标签下，右键单击项目名称 → Properties；

Project Properties 窗口会弹出。

2. 在 loading 目录下，勾选 Load symbols when programming or building for production。然后点击 OK 按钮。



3. 编译本项目。

Production → Build Main Project

确认编译成功。

(三) 步骤 2：烧写项目

1. 硬件准备。

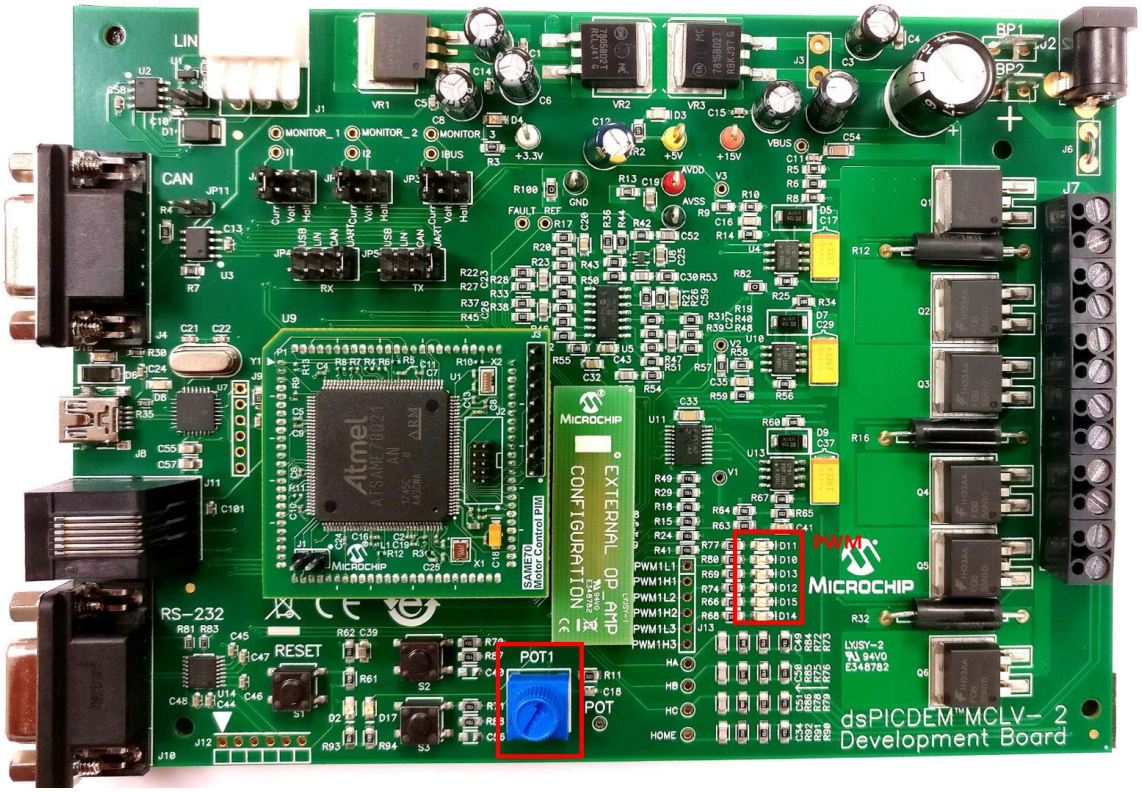
连接 ICD4、转接头、SAME70 PIM、PC 主机；

为 MCLV-2 印板供电（直流 24V）；

2. 点击  以烧写项目。

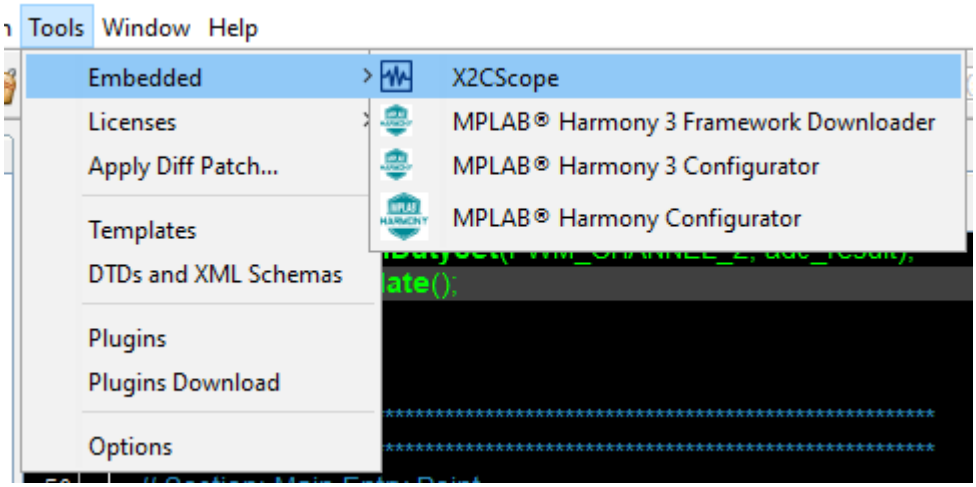
(四) 步骤 3：测试 PWM 波形输出

1. 观察 MCLV-2 印板上的 6 个 PWM 波形指示灯；
2. 旋转电位器，观察指示灯明暗的同步变化；
3. 也可以用示波器观察 PWM 波形，以进行验证。



(五) 步骤 4：观察 AD 转换结果和正交计数值

1. 打开 X2CScope 工具



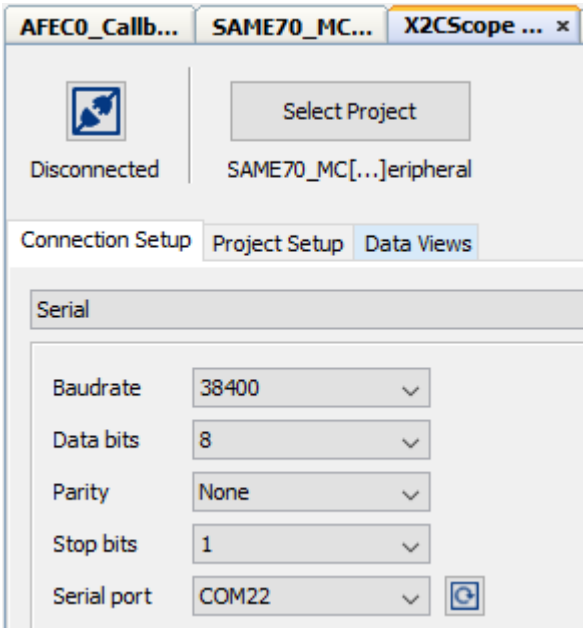
2. 连接 X2CScope

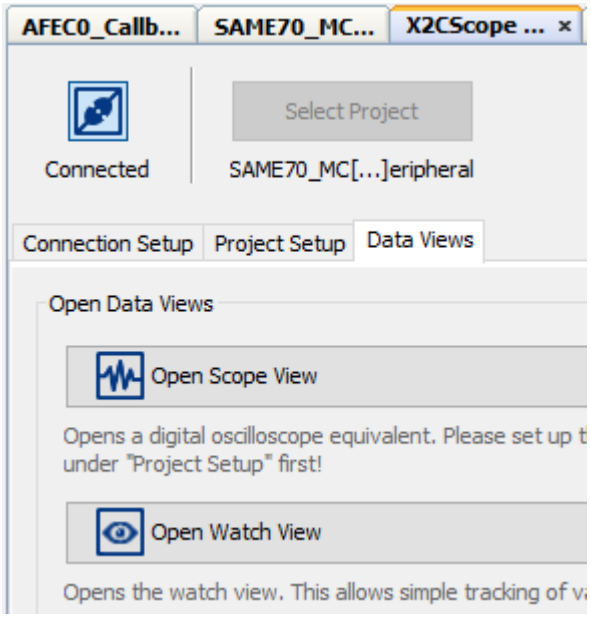
在 X2CScope Configuration 标签下，点击 Select Project 按钮；在下拉列表中，选择本项目；

设定 UART 参数，例如波特率、数据位数、极性、停止位、PC 串口。

点击 Disconnected 按钮，以进行连接；

连接成功后，按钮显示变为 Connected。





3. 观察全局变量

点击 Data Views 标签；

点击 Open Watch View；

点击  ，添加希望观察的全局变量；

旋转电位器，观察 AD 转换结果值的同步变化；

旋转电机转子，观察正交编码计数值的同步变化。