AVR® DA 培训手册

采用 Curiosity Nano 的低功耗模式

准备工作

- **硬件准备工作**
  - AVR128DA48 Curiosity Nano 板（DM164151），需进行较小的硬件修改（见本文档中的描述）
  - 功率调试器（ATPOWERDEBUGGER）
  - 两根 Micro-USB 线缆：一根用于连接功率调试器，一根用于连接 Curiosity Nano 板
  - 两根短电线，两端带孔可插入引脚
- **软件准备工作**
  - MPLAB® X 5.40
  - MCC 插件（用于 MPLAB X v3.95）
  - AVR 库（用于 MCC v2.3.0）
  - 最新版 Atmel Studio

简介

本文档介绍了通过纯软件方法和使用 CIP 完成不同任务之间的功耗差异。

本动手实践培训涵盖以下主题：

- 任务 1：比较事件系统与中断
- 任务 2：比较用于模数转换器（Analog-to-Digital Converter，ADC）的软件累加与硬件累加

在每项任务开始时，都会介绍与其相关的基本理论，然后通过 Atmel Studio 的数据可视化器验证相关功能。

关键图标标识符

本文档中使用了以下图标，旨在标识不同的任务部分，并降低复杂度。

信息：提供有关特定主题的背景信息。

提示：重点介绍有用的提示和技巧。

任务：突出显示需要完成的目标。

结果：突出显示任务步骤的预期结果。

指示重要信息。

执行：重点介绍必要时要通过目标器件执行的操作。

目录

准备工作.....1

简介.....1

关键图标标识符.....2

1. 任务 1：比较事件系统与中断.....4
   - 1.1. 硬件说明.....4
   - 1.2. 修改 Curiosity 板以获得准确的功率测量结果.....4
   - 1.3. 说明所用模块.....5
   - 1.4. 创建软件实现程序.....5
   - 1.5. 创建事件系统实现项目.....13
   - 1.6. 对器件进行编程并比较效率.....24

2. 任务 2：比较用于 ADC 的软件累加与硬件累加.....28
   - 2.1. 创建软件实现项目.....28
   - 2.2. 创建硬件实现项目.....34
   - 2.3. 对器件进行编程并比较功率效率.....41

3. 版本历史.....46

Microchip 网站.....47

产品变更通知服务.....47

客户支持.....47

Microchip 器件代码保护功能.....47

法律声明.....47

商标.....48

质量管理体系.....48

全球销售及服务网点.....49

1. 任务 1：比较事件系统与中断

在此项任务中，将比较两种不同方法的功耗。第一种方法需要通过定时器溢出中断将单片机从休眠状态唤醒，从而启动 ADC 转换以读取温度传感器的数据。当转换完成后，将触发结果就绪中断，并读取转换的结果。

在第二种方法中，将使用事件系统（EVSYS）模块。该模块支持用户通过事件连接不同的外设。将对第一种方法进行修改，以使定时器溢出能够触发用于启动 ADC 转换的事件，从而读取温度传感器的数据。转换结束后，单片机将从休眠状态唤醒以读取结果。

1.1 硬件说明

Curiosity 板包含一个 AVR128DA48 单片机、一个 Nano 嵌入式调试器、一个 LED 和一个按钮。使用 Nano 调试器，用户可在不需要外部硬件的情况下对电路板进行编程并运行调试功能。该调试器还提供 UART-USB 桥接功能。

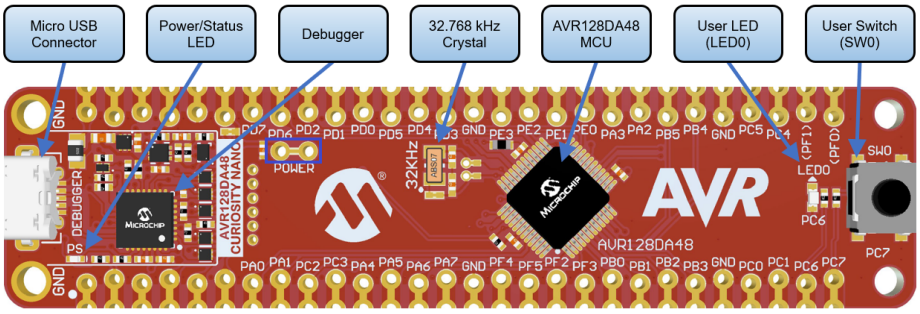
该功率调试器也是一种编程器，能够提供具有可配置电压的功率输出，并可用于测量与其连接的器件的电流消耗。

1.2 修改 Curiosity 板以获得准确的功率测量结果

任务：修改 Curiosity Nano 板，具体方法为切断一条走线并焊接两个引脚。

如果通过 Micro-USB 端口观察功率，功率调试器会产生一定的功耗，进而导致读数不准确。

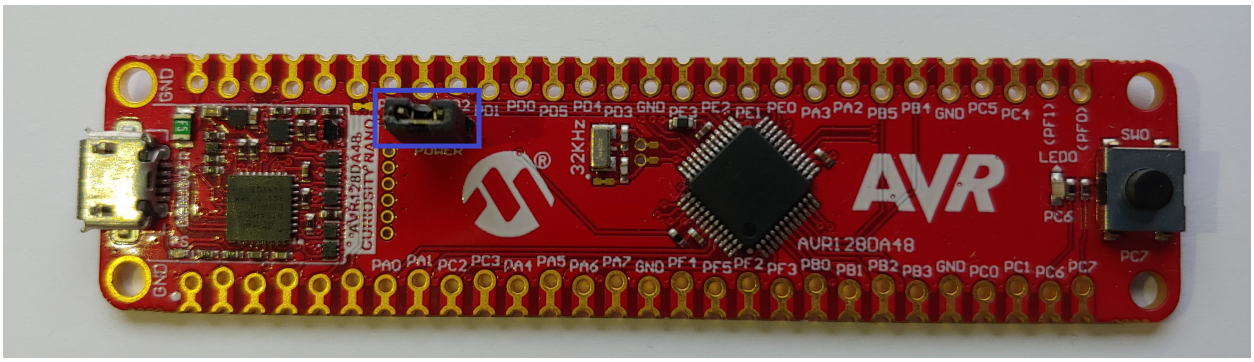
图 1-1. POWER 走线位置



Curiosity 板上有一小段标有 POWER 的走线。使用锋利的工具切断这段走线，使这两个孔不再接触。使用电压表进行检查。

切断走线后，焊接孔中的两个引脚。如要对电路板进行编程，此后必须用电线连接这两个引脚。

图 1-2. 修改后的 Curiosity 板



1.3 说明所用模块

本任务中将使用的模块包括实时时钟（Real-Time Clock，RTC）、ADC、参考电压（VREF）和事件系统（EVSYS）。

RTC 将用作定时器，可计数至一秒。

ADC 模块可将模拟电压转换为 12 位数字。可通过 EVSYS 启动 ADC 转换。完成一次转换后，ADC 还会触发一个事件和一个中断。该 ADC 将用于读取内部温度传感器的数据。

参考电压是一种外设，可为其他外设提供稳定的电压。如果将该外设设置为向 ADC 提供 2.048V 的参考电压，它还可以使能温度传感器。

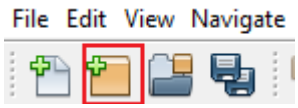
事件系统是一种路由网络，可以在没有 CPU 干预的情况下实现外设间通信。将事件从一个外设发送到另一个外设触发操作即可完成通信。事件没有延迟时间，也永远不会丢失，因此可实现更理想的实时、可靠的应用。

1.4 创建软件实现程序

任务：配置要在本项目中使用的硬件模块，并向项目中添加代码。

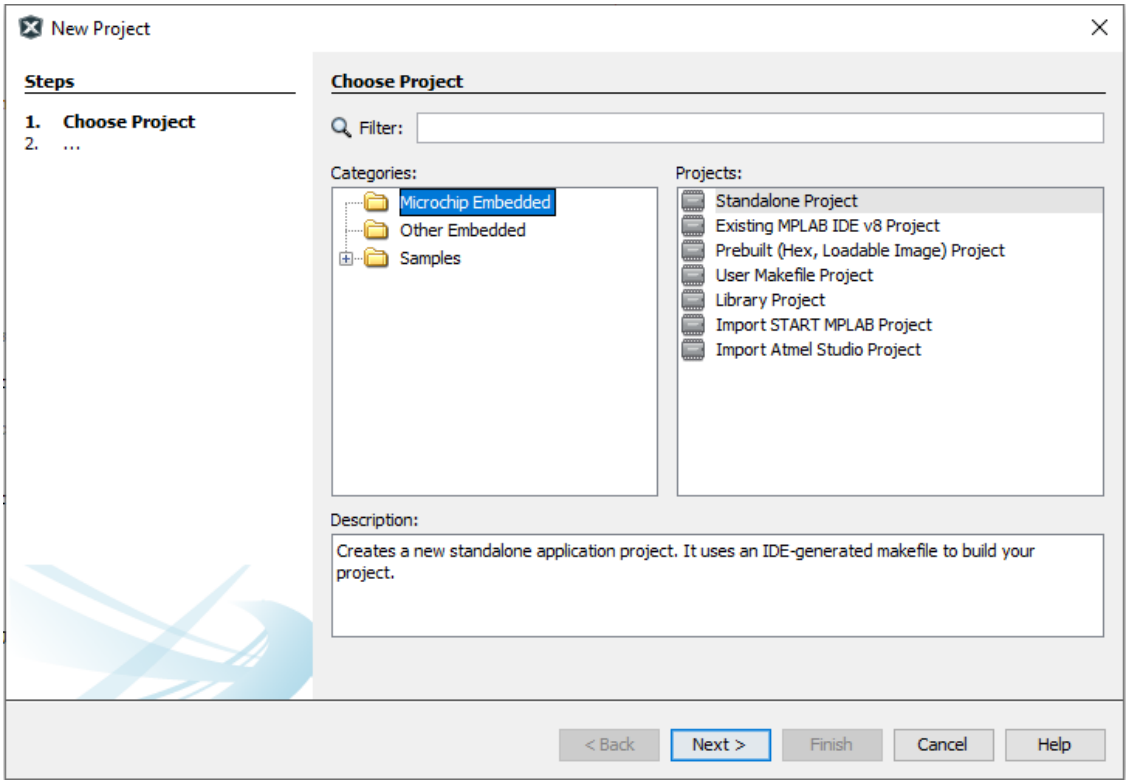
1. 为 AVR128DA48 新建一个 MPLAB X 项目。
   a. 打开 MPLAB X。
   b. 选择 **File → New Project**（文件 → 新建项目）或 **New Project**（新建项目）按钮。

      图 1-3. New Project

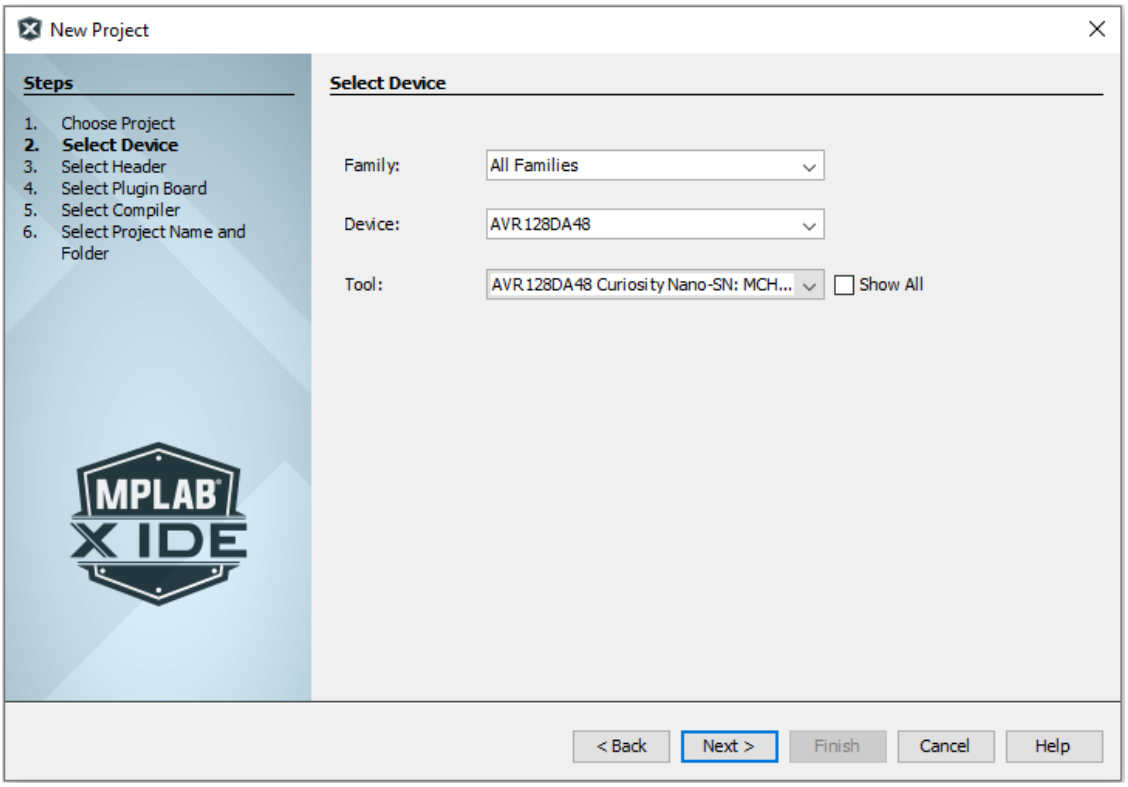
   c. 单击 **Next**（下一步）（默认选择 Microchip Embedded → Standalone Project（Microchip 已安装工具 → 独立项目））。

图 1-4. 项目类型



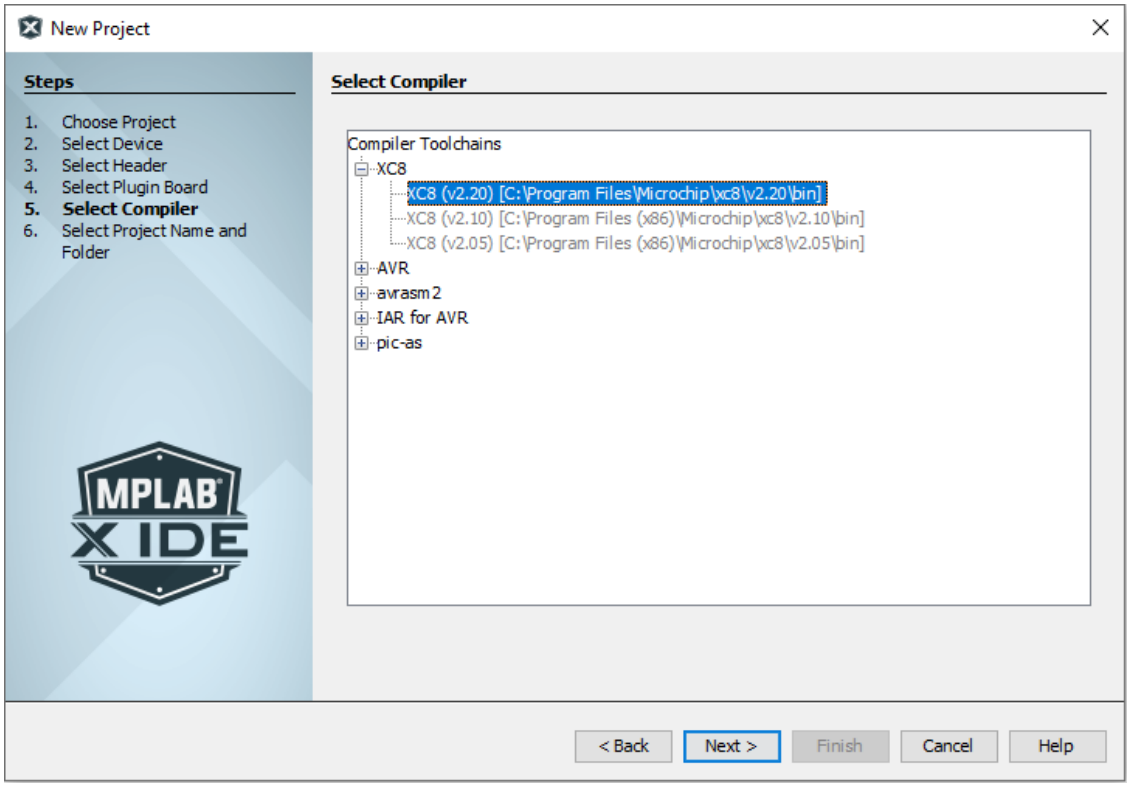
d. 在 Device（器件）字段中，搜索 AVR128DA48。在 Tool（工具）类别中，如果 Curiosity Nano 板已连接到计算机，则选择该电路板，否则选择 **None**（无）。单击 **Next**。

图 1-5. 器件选择



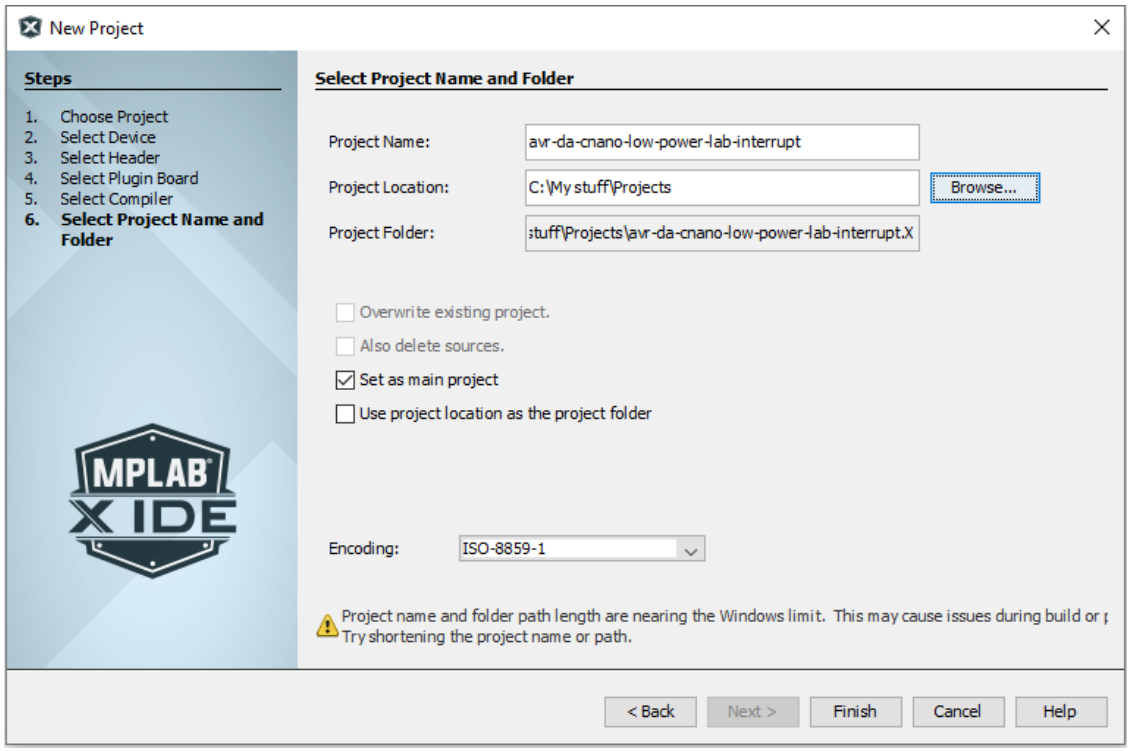
e. 选择 **XC8 2.20 编译器**，然后单击 **Next**。

图 1-6. 编译器选择



f. 为项目命名（以及确定保存位置），然后单击 **Finish**（完成）。

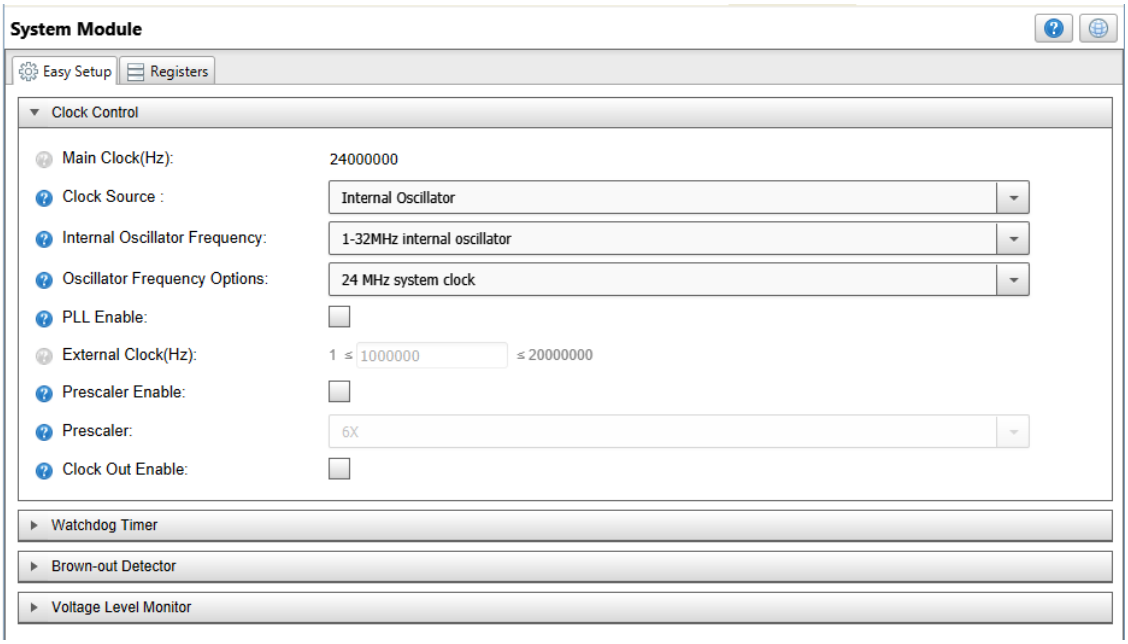
图 1-7. Project Name（项目名称）



2. 打开 MPLAB 代码配置器（MPLAB Code Configurator，MCC）并配置外设。
   a. 在 System Module（系统模块）中，选择时钟频率为 24 MHz 的内部振荡器作为时钟源。

信息：为 Clock Source（时钟源）选择 Internal Oscillator（内部振荡器），为 Oscillator Frequency Options（振荡器频率选项）选择 24 MHz system clock（24 MHz 系统时钟），并禁止预分频器。

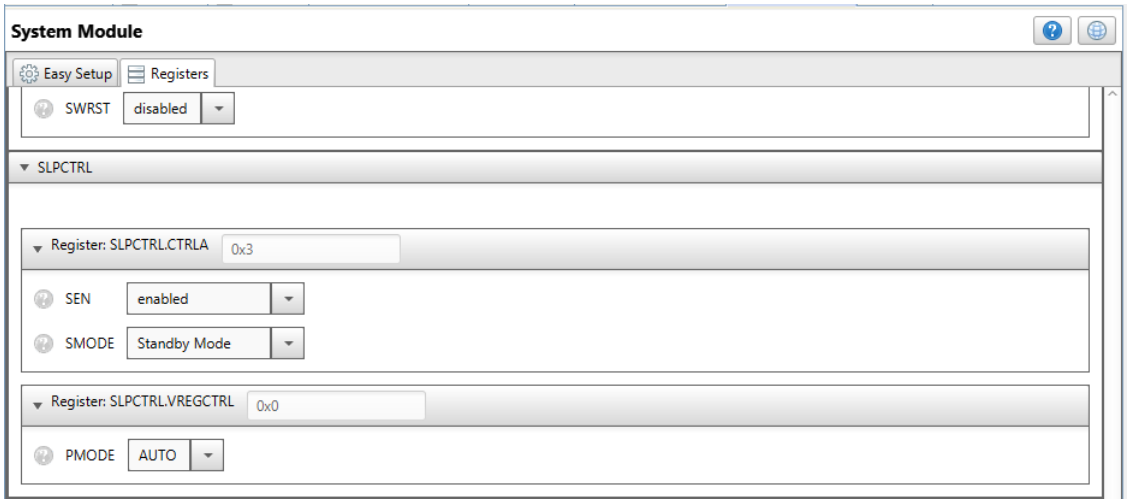
图 1-8. System Module 配置



   b. 在 System Module 的 **Registers**（寄存器）选项卡中，使能休眠并将模式设为待机。

信息：在 System Module 中，转至 **Registers** 选项卡，然后向下滚动至 SLPCTRL。修改 CTRLA 寄存器，以使能休眠并选择模式。

图 1-9. 休眠配置

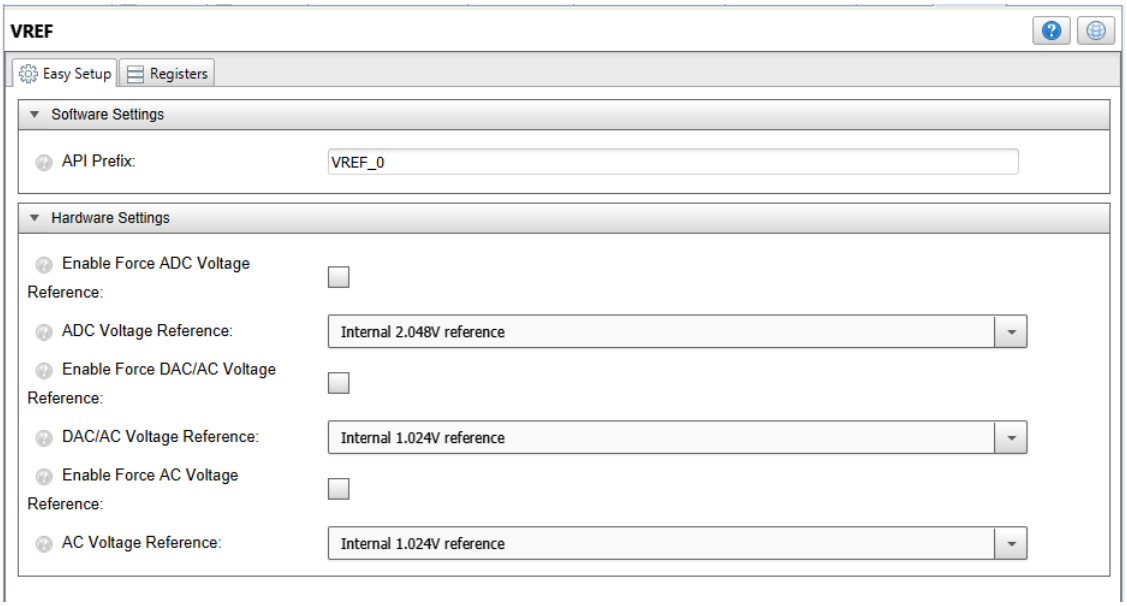


c. 从器件资源中添加 VREF 模块，并将其配置为向 ADC 提供 2.048V 的参考电压。

信息：必须将 ADC 参考电压配置为内部 2.048V 参考电压。请勿使能强制 ADC 电压选项。

信息：VREF 用于使能单片机的温度传感器。

图 1-10. VREF 配置

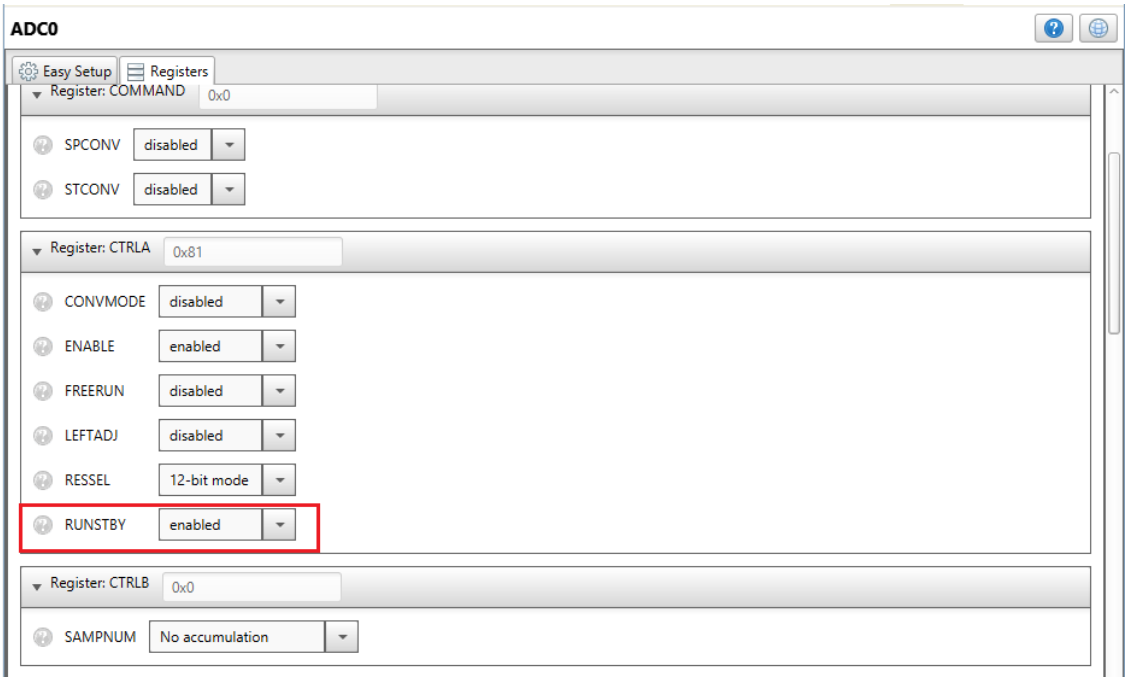


d. 从器件资源中添加 ADC 模块，并将其配置为 12 位模式、右对齐结果、无累加、采样长度为一个节拍和待机模式下运行（RUNSTBY）。

信息：必须禁止右移结果选项。采样长度和采样累加数量需要在 **Hardware Settings**（硬件设置）选项卡中选择。12 位模式为默认设置。RUNSTBY 位在 CTRLA 寄存器中，可在 **Registers** 选项卡中更改。

图 1-11. ADC 配置

图 1-12. RUNSTBY ADC 配置



e. 从器件资源中添加 RTC 模块，并将其配置为使用 32.768 kHz 的内部振荡器、周期为 1 秒且预分频系数为 1，并激活待机模式下运行（RUNSTBY）。

信息：时钟、预分频系数和周期在 **Hardware Settings** 选项卡中设置，而 RUNSTBY 位可在 CTRLA 寄存器的寄存器窗格中找到。

图 1-13. RTC 配置

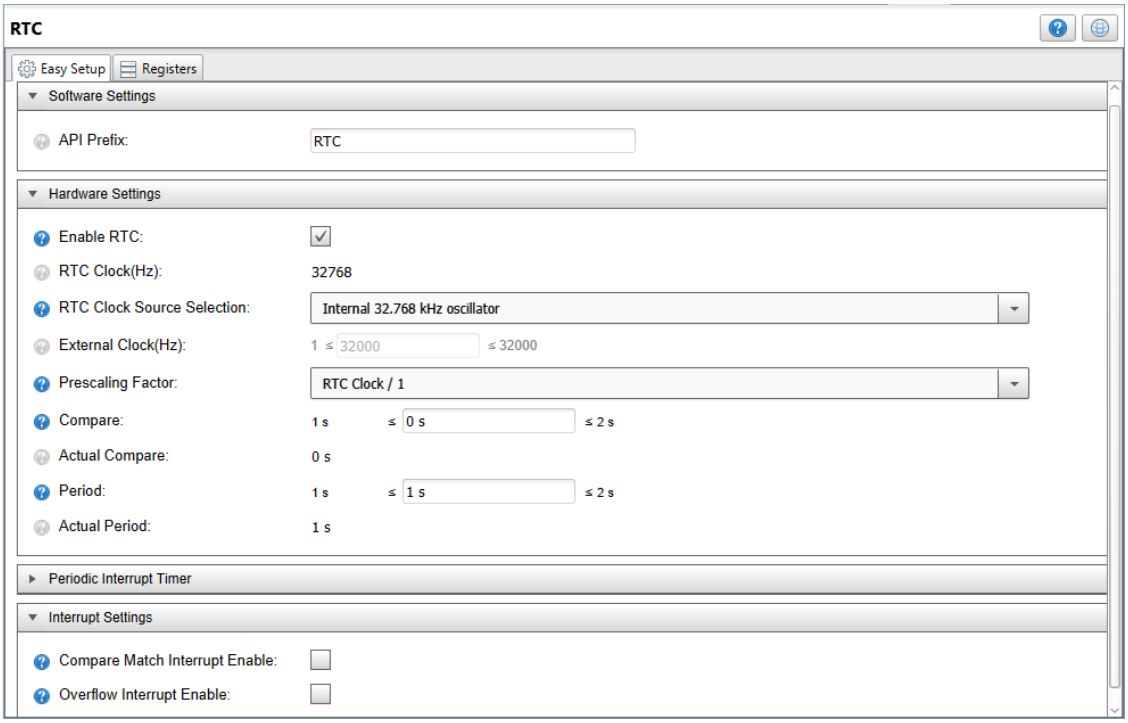
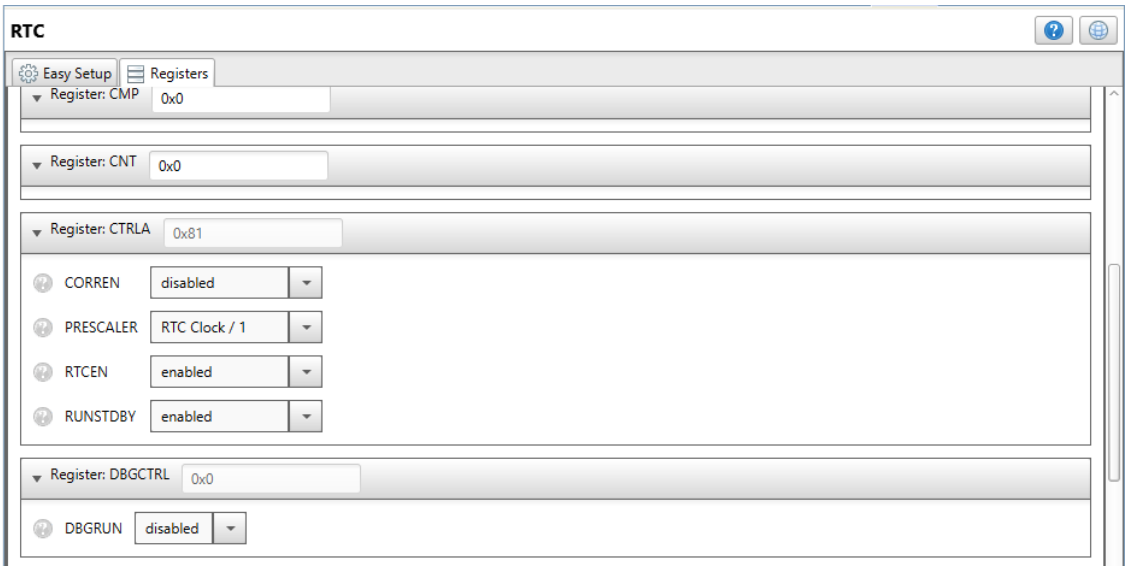


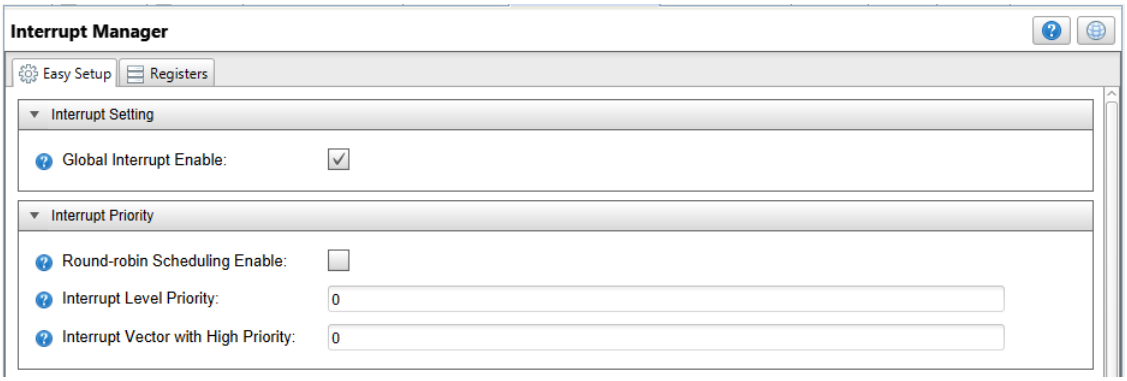
图 1-14. RUNSTBY RTC 配置



f. 允许全局中断，并允许 ADC 的结果就绪中断和 RTC 的溢出中断。

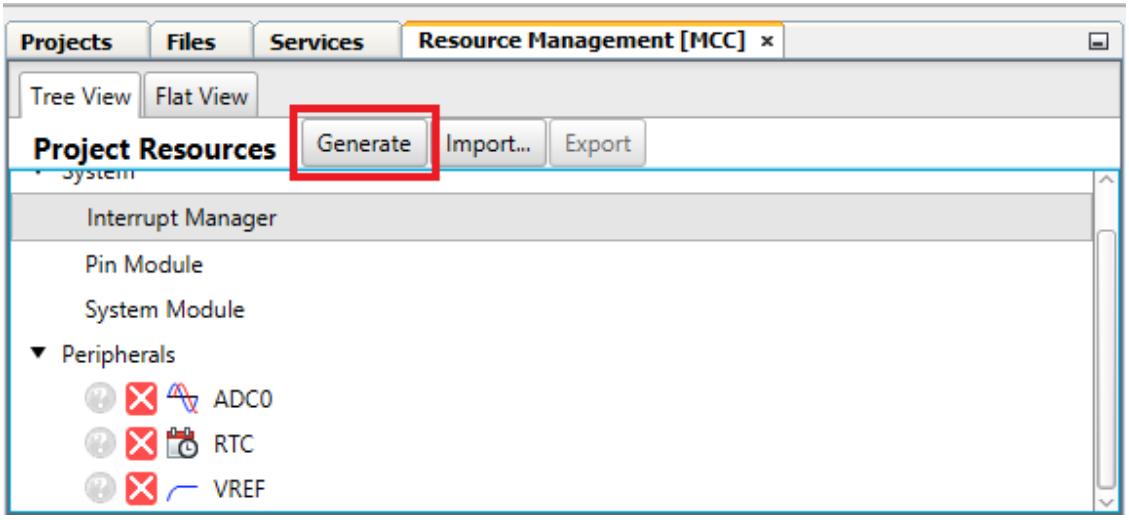
信息：在 Interrupt Manager（中断管理器）中允许全局中断，在 ADC 模块的 **Interrupt Setting**（中断设置）选项卡中允许 ADC 中断，在 RTC 模块的 **Interrupt Setting** 选项卡中允许 RTC 中断（见前面步骤的图片）。

图 1-15. Interrupt Manager 配置



g. 按下 **Generate**（生成）按钮。

图 1-16. Generate 按钮



3. 向生成的文件中添加代码。

a. 向 `main.c` 文件中添加以下代码。

```
#include <avr/io.h>
#include <avr/sleep.h>
#include "mcc_generated_files/mcc.h"

void TIMER_interrupt(void);

uint16_t result;

int main(void)
{

    SYSTEM_Initialize();
    RTC_SetOVFIsrCallback(TIMER_interrupt);

    while (1){
        sleep_cpu();
        result = ADC0_GetConversionResult();
    }
}

void TIMER_interrupt()
{
    ADC0_StartConversion(ADC_MUXPOS_TEMPSENSE_gc);
}
```

信息： `SYSTEM_Initialize()` 在 `mcc.c` 中定义，`RTC_SetOVFIsrCallback()` 在 `rtc.c` 中定义，`ADC0_StartConversion()` 和 `ADC0_GetConversionResult()` 在 `adc.c` 中定义。

- `SYSTEM_Initialize()` 用于设置 CPU 和外设的所有配置寄存器。该函数由 MCC 生成。
- `RTC_SetOVFIsrCallback()` 函数用于设置在 main 函数中定义的 `TIMER_interrupt()` 函数的回调函数（触发中断时将调用的函数）。
- `TIMER_interrupt()` 函数用于启动 ADC 转换。
- `ADC0_StartConversion()` 函数用于在 ADC 的温度通道上启动转换。
- `ADC0_GetConversionResult()` 用于返回上次转换的结果。

b. 在位于 ***Source Files → MCC Generated files → src***（源文件 → MCC 生成的文件 → src）内的 `pin_manager.c` 文件中，将引脚方向的代码替换为以下内容：

```
    PORTA.DIR = 0xFF;
    PORTB.DIR = 0xFF;
    PORTC.DIR = 0x3F;
    PORTD.DIR = 0xFF;
    PORTE.DIR = 0xFF;
    PORTF.DIR = 0xFF;
```

信息： 此操作是为了防止悬空引脚干扰功耗读数。

c. 在工具栏中，按下 **Clean and Build**（清除并编译）按钮，然后验证程序编译是否正确无误。

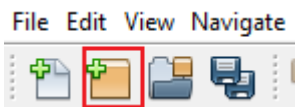
图 1-17. Clean and Build

1.5 创建事件系统实现项目

任务： 配置要在本项目中使用的硬件模块，并向项目中添加代码。

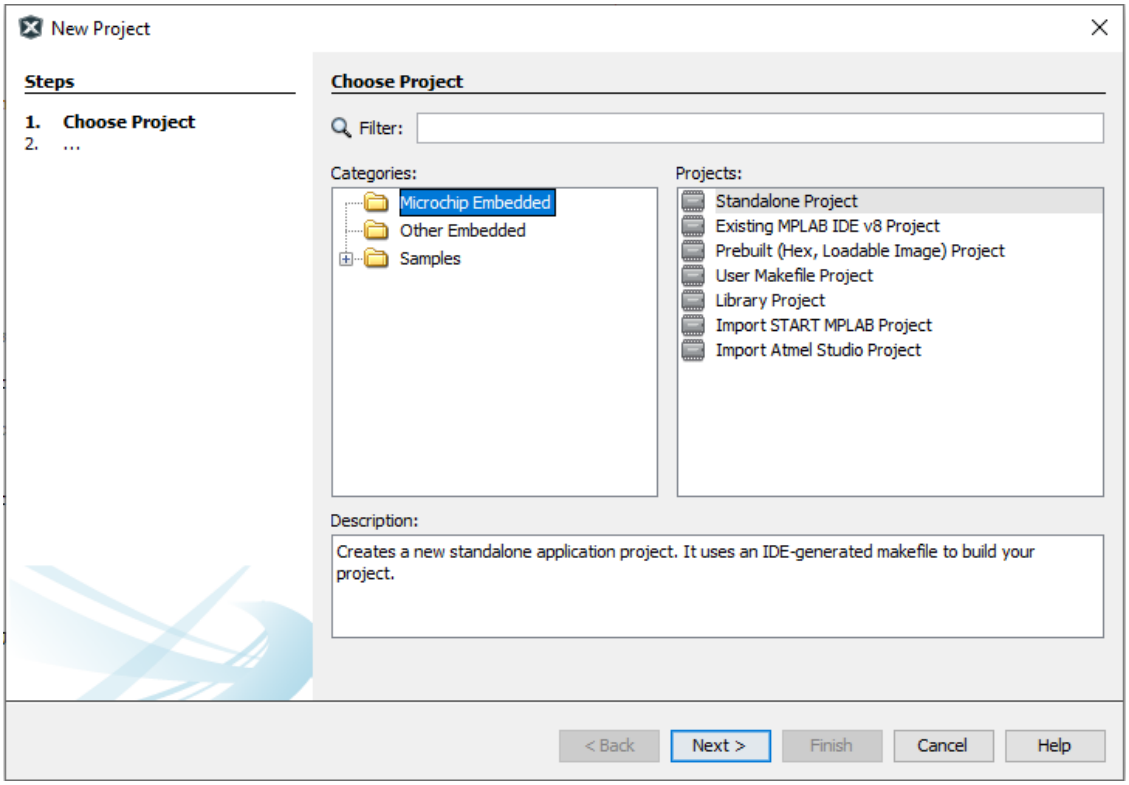
1. 为 AVR128DA48 新建一个 MPLAB X 项目。
   a. 打开 MPLAB X。
   b. 选择 *File → New Project* 或 **New Project** 按钮。

图 1-18. New Project



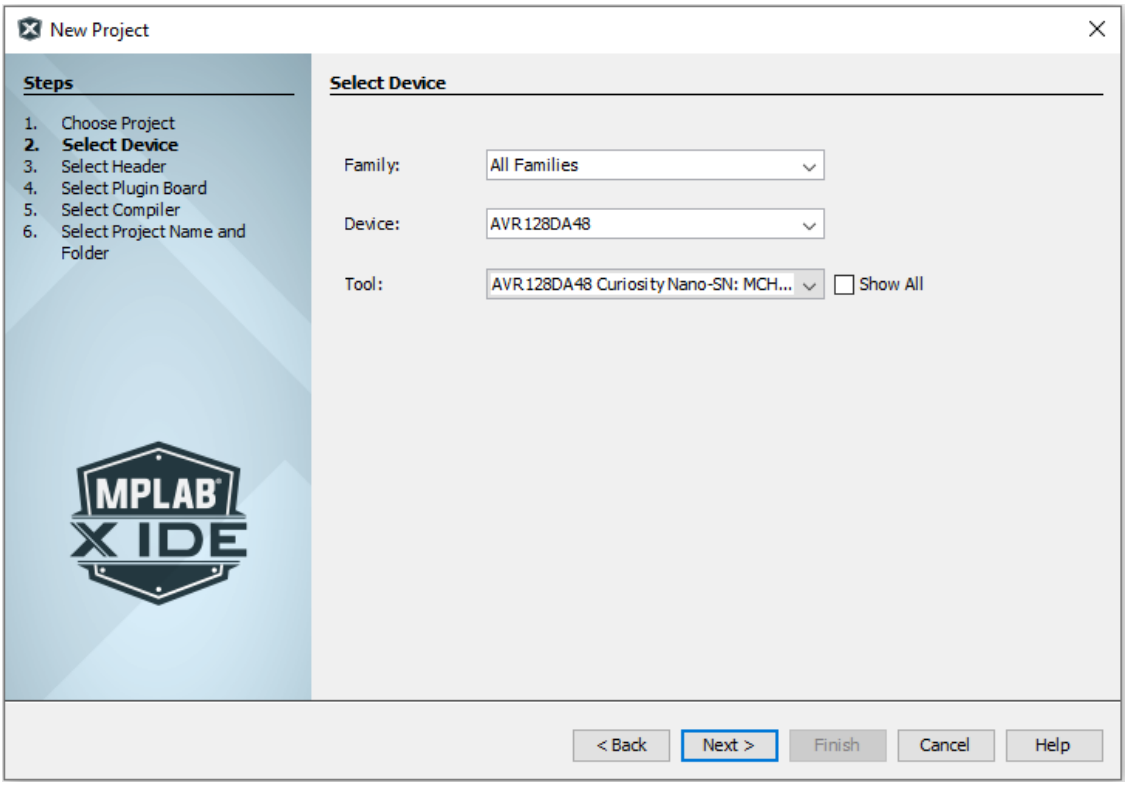
c. 单击 **Next**（默认选择 Microchip Embedded → Standalone Project）。

图 1-19. 项目类型



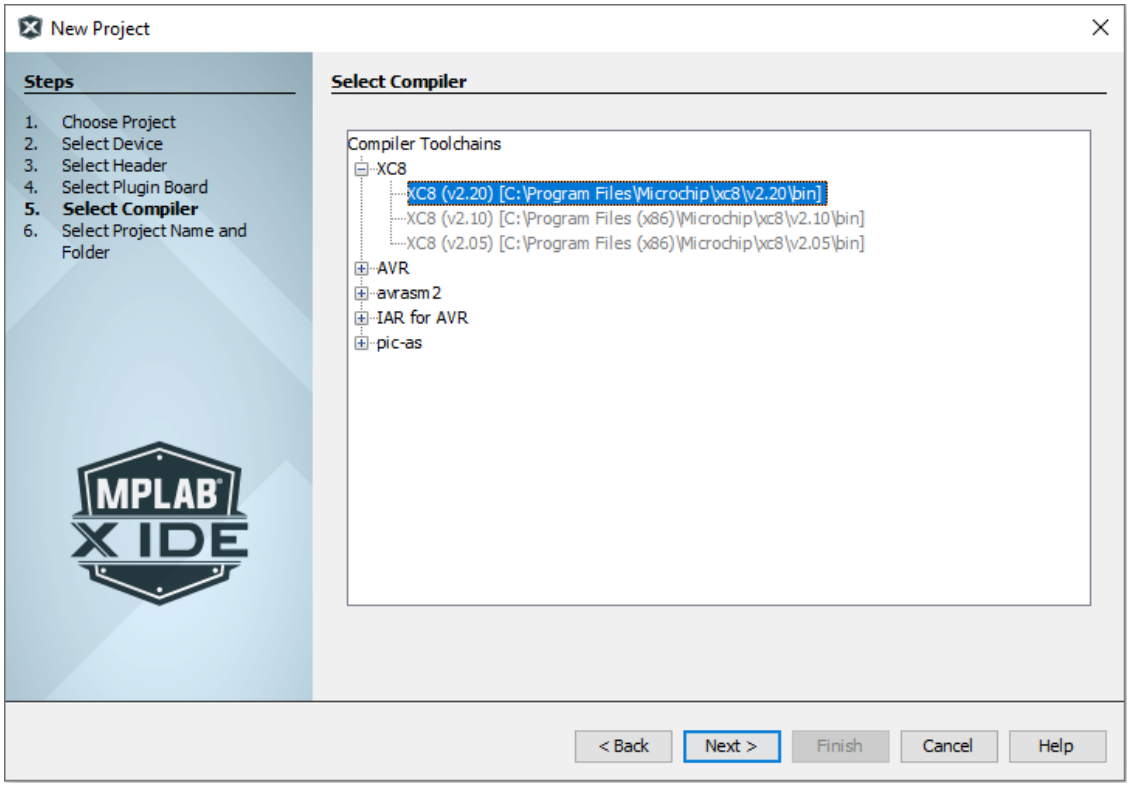
d. 在 Device 字段中，搜索 AVR128DA48。在 Tool 类别中，如果 Curiosity Nano 板已连接到计算机，则选择该电路板，否则选择 **None**。单击 **Next**。

图 1-20. 器件选择



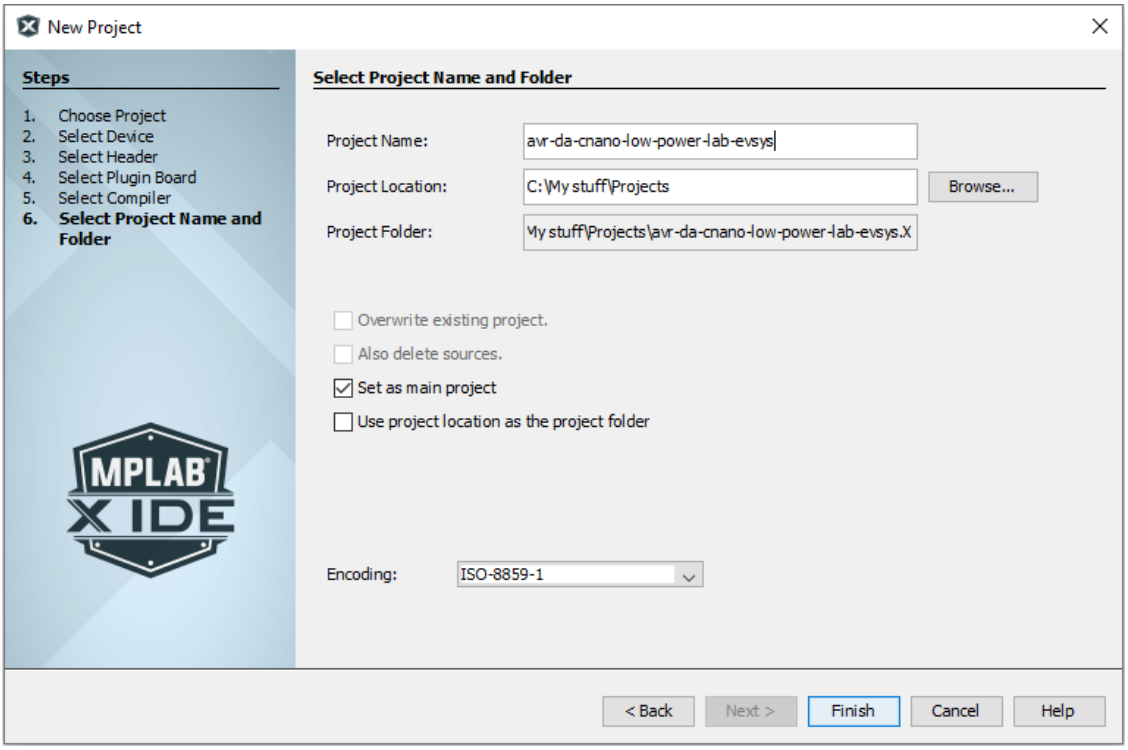
e. 选择 **XC8 2.20 编译器**，然后单击 **Next**。

图 1-21. 编译器选择



f. 为项目命名（以及确定保存位置），然后单击 **Finish**。

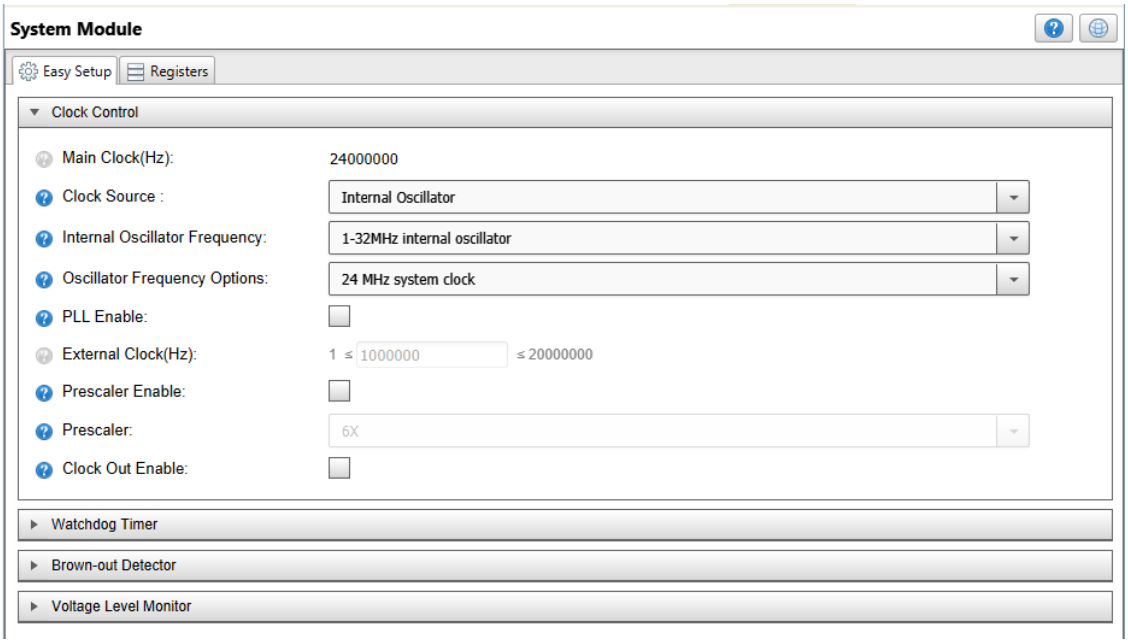
图 1-22. Project Name



2. 打开 MPLAB 代码配置器（MCC）并配置外设。
   a. 在 System Module 中，选择时钟频率为 24 MHz 的内部振荡器作为时钟源。

信息：为 Clock Source 选择 Internal Oscillator，为 Oscillator Frequency Options 选择 24 MHz System Clock，并禁止预分频器。

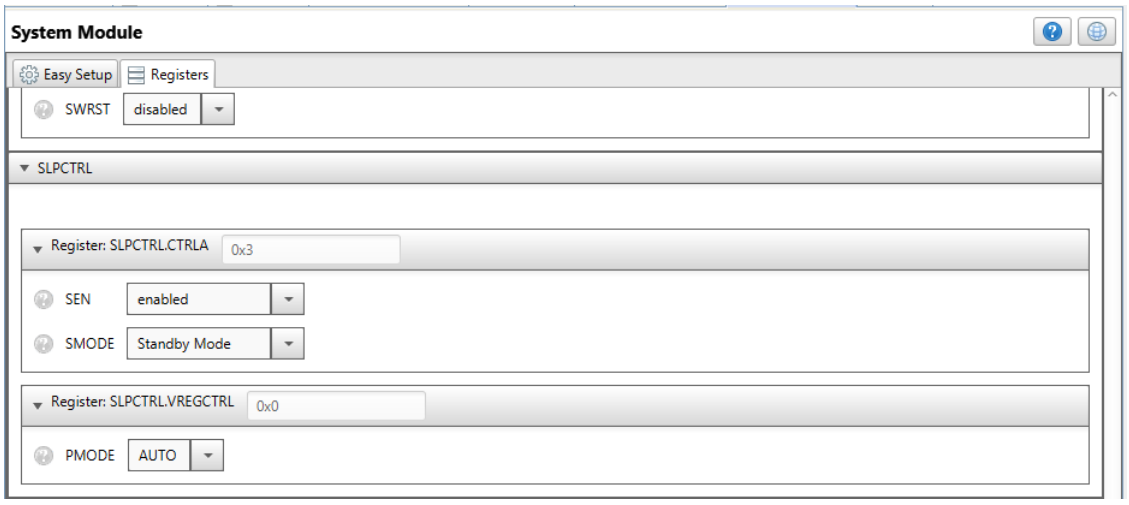
图 1-23. System Module 配置



b. 在 System Module 的 **Registers** 选项卡中，使能休眠并将模式设为待机。

信息：在 System Module 中，转至 **Registers** 选项卡，然后向下滚动至 SLPCTRL。修改 CTRLA 寄存器，以使能休眠并选择模式。

图 1-24. 休眠配置

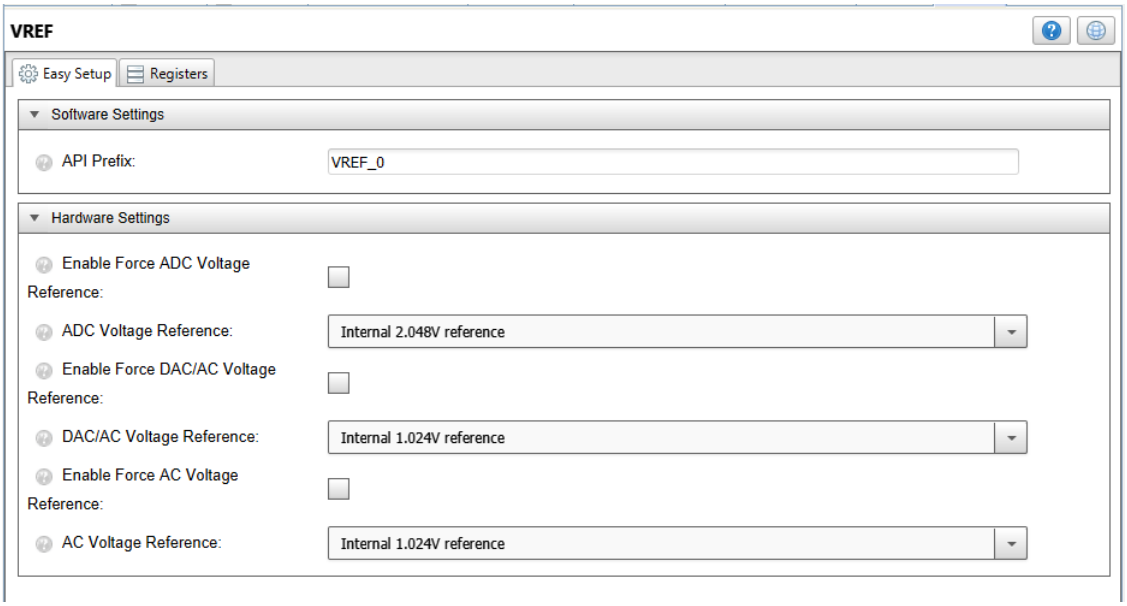


c. 从器件资源中添加 VREF 模块，并将其配置为向 ADC 提供 2.048V 的参考电压。

信息：必须将 ADC 参考电压配置为内部 2.048V 参考电压。请勿使能强制 ADC 电压选项。

信息：VREF 用于使能单片机的温度传感器。

图 1-25. VREF 配置



d. 从器件资源中添加 ADC 模块，并将其配置为 12 位模式、右对齐结果、无累加、采样长度为一个节拍和待机模式下运行（RUNSTBY）。同时，使能 STARTEI 位，以便在收到事件时启动转换。

信息：必须禁止右移结果选项。采样长度和采样累加数量需要在 **Hardware Settings** 选项卡中选择。12 位模式为默认设置。RUNSTBY 位在 CTRLA 寄存器中，可在 **Registers** 选项卡中更改。STARTEI 位在 **Registers** 选项卡的 EVCTRL 寄存器中。

图 1-26. ADC 配置

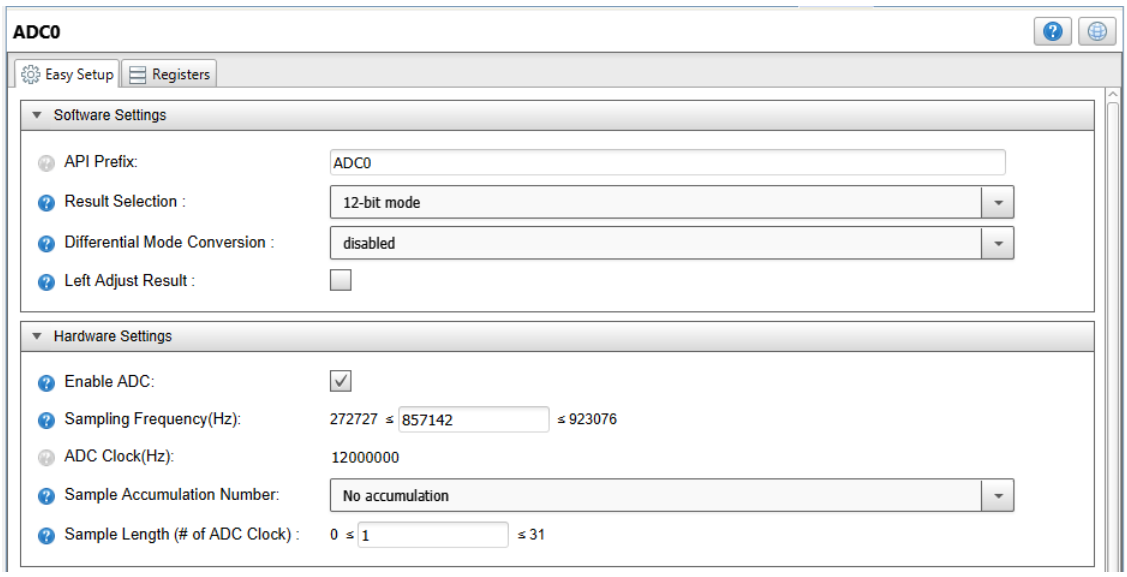


图 1-27. RUNSTBY RTC 配置

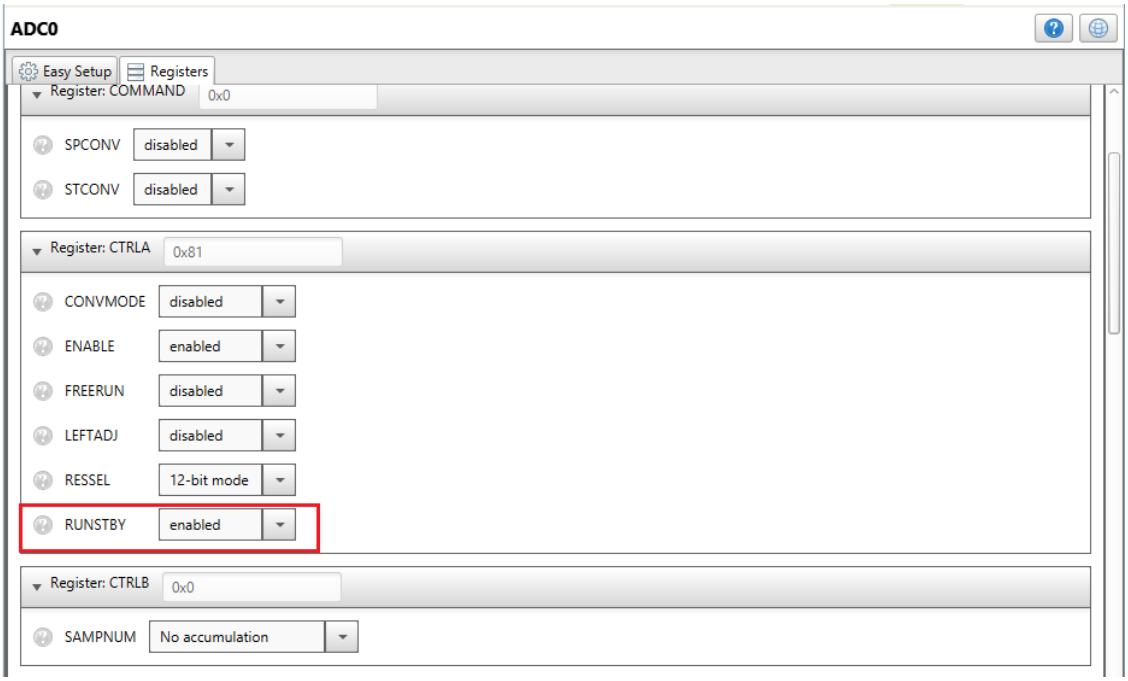
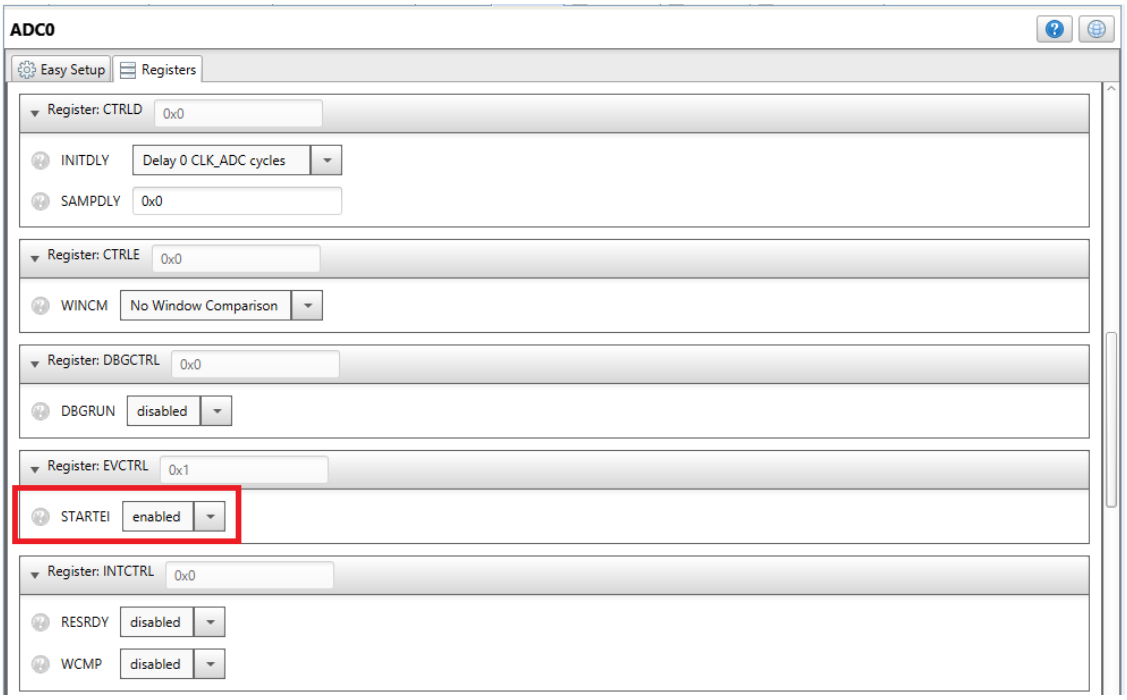


图 1-28. 通过事件 ADC 配置启动转换



e. 从器件资源中添加 RTC 模块，并将其配置为使用 32.768 kHz 的内部振荡器、周期为 1 秒且预分频系数为 1，并激活待机模式下运行（RUNSTBY）。

信息：时钟、预分频系数和周期在 **Hardware Settings** 选项卡中设置，而 RUNSTBY 位可在 CTRLA 寄存器的寄存器窗格中找到。如果已允许溢出中断，请将其禁止。

图 1-29. RTC 配置

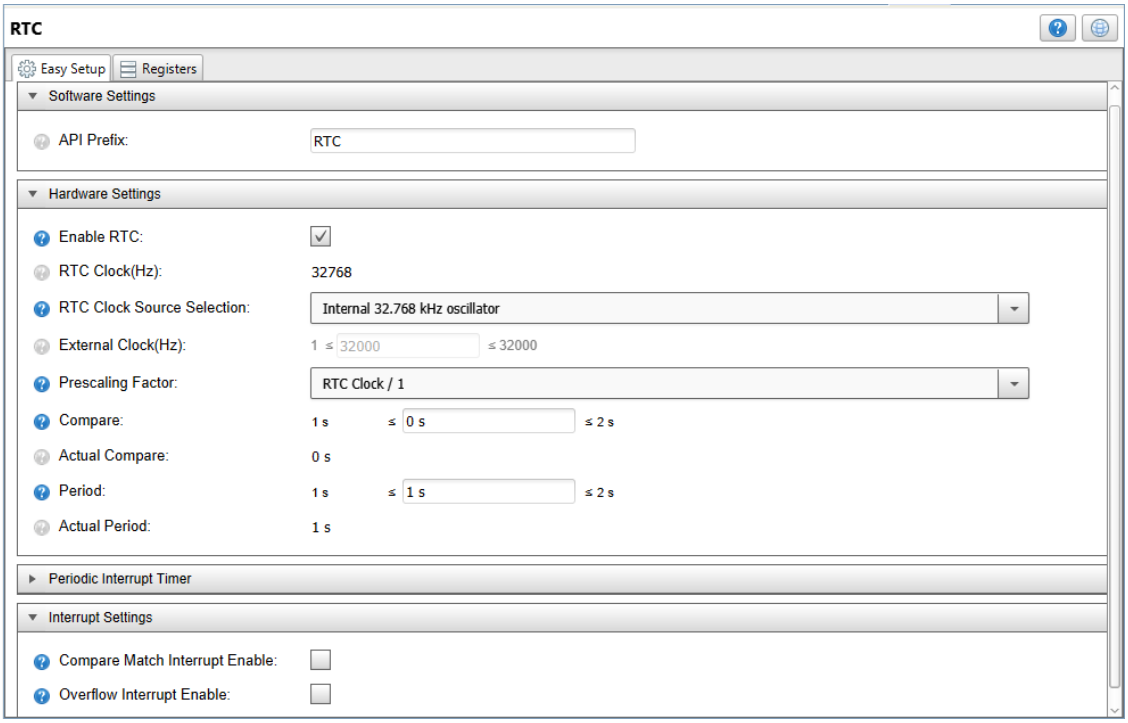
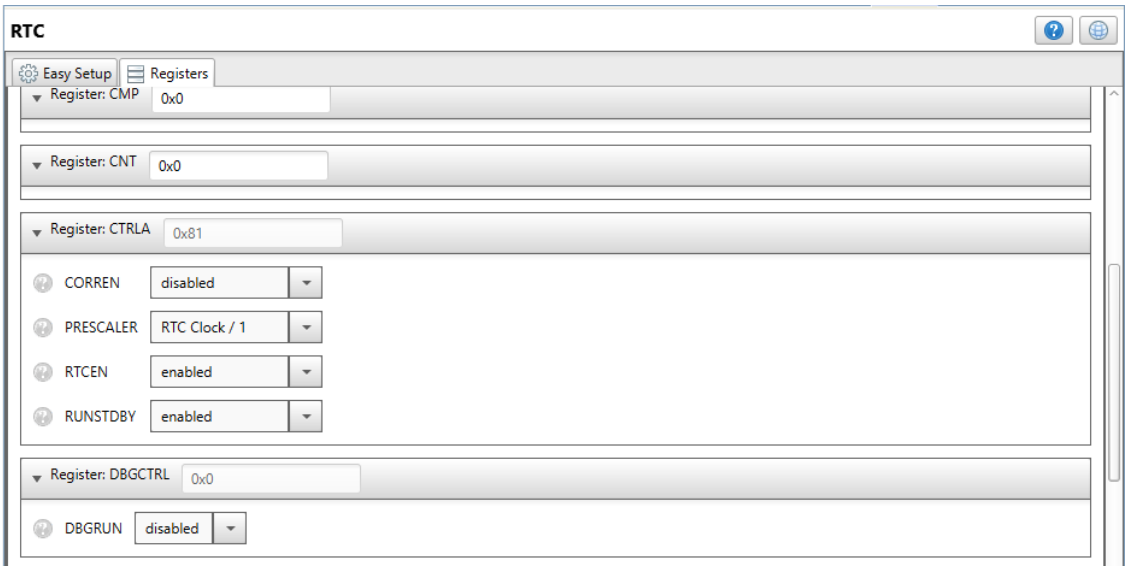


图 1-30. RUNSTBY RTC 配置



f. 添加 EVSYS 模块。将通道 0 的事件生成器配置为 RTC_OVF，将要触发的事件配置为 ADC0START。将通道 1 的事件生成器配置为 ADC0_RESRDY，将要触发的事件配置为 EVSYSEVOUTC。

图 1-31. EVSYS 配置

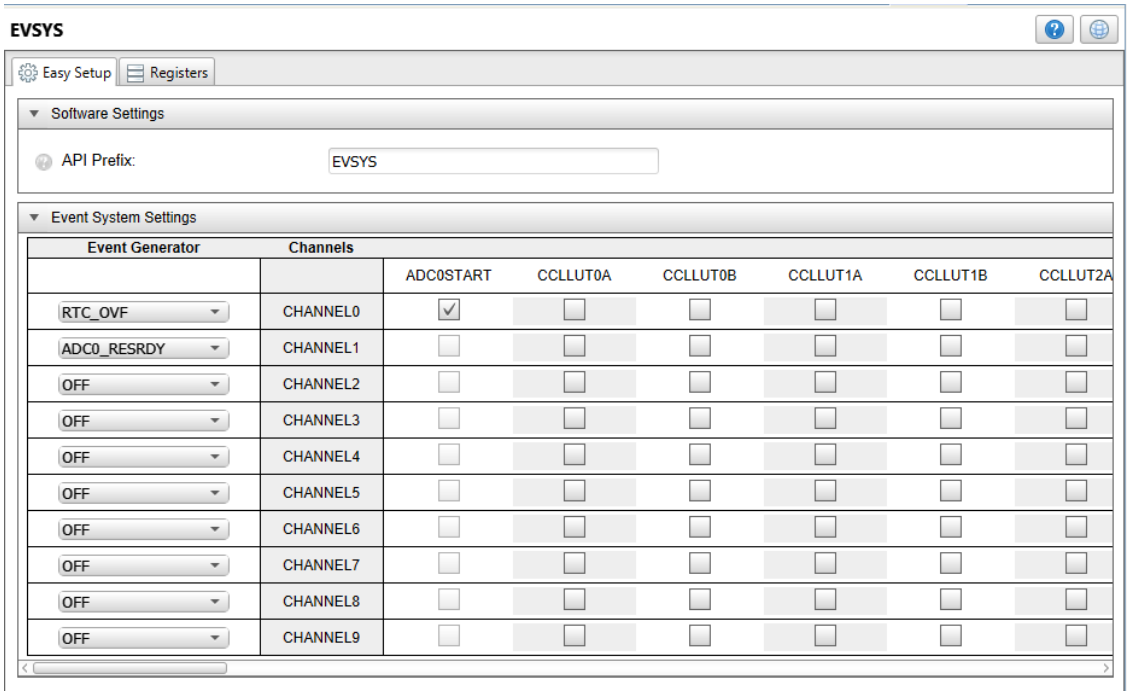
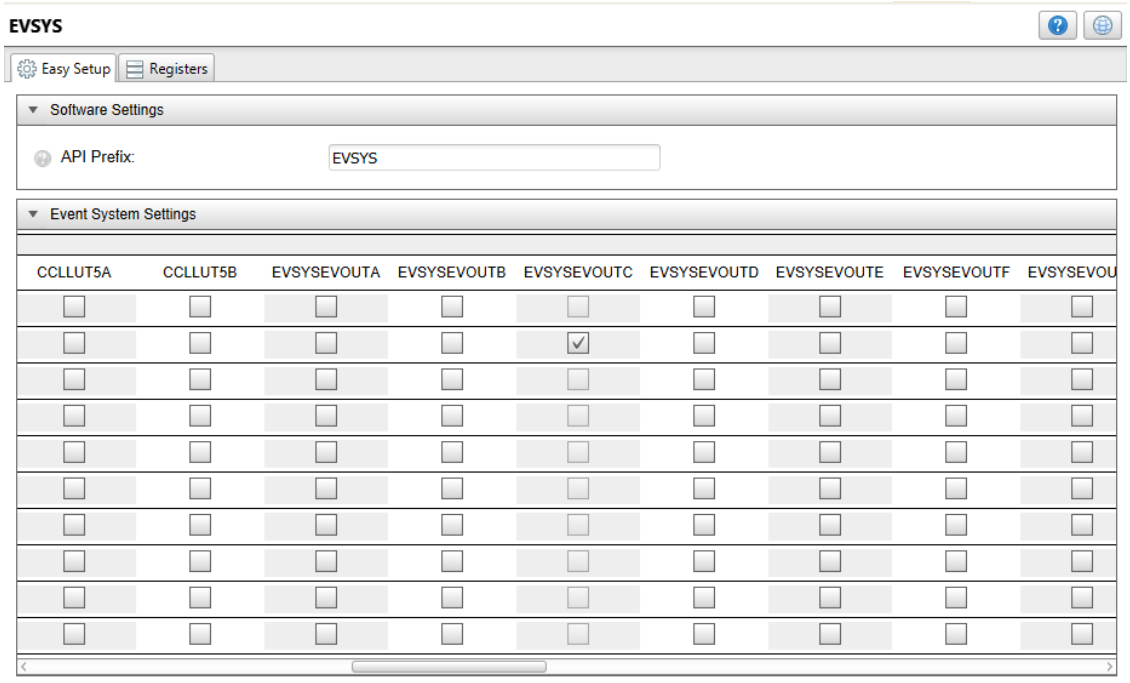
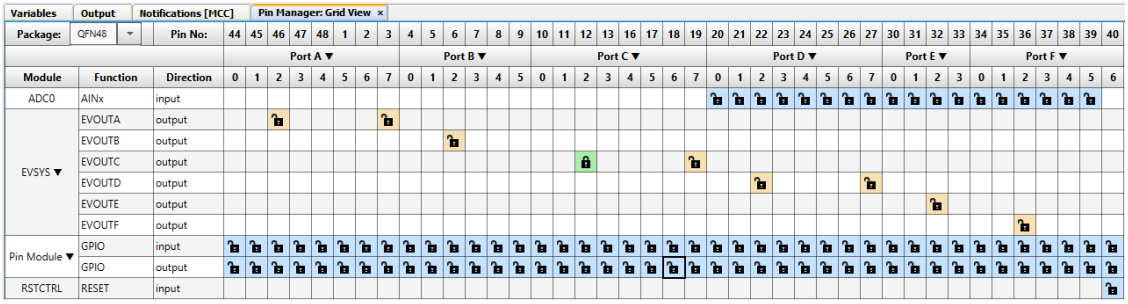


图 1-32. EVSYS 配置



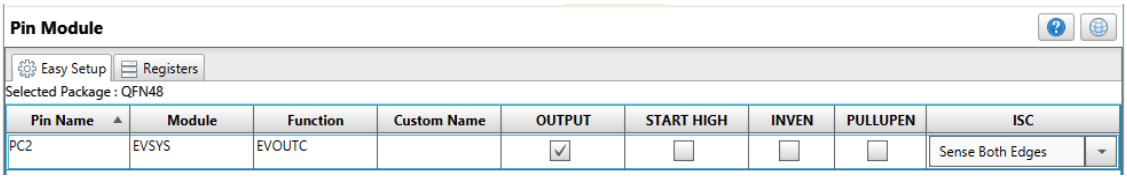
g. 在 Pin Manager: Grid View（引脚管理器：网格视图）中，选择 **PC2** 作为 EVSYS 的 EVOUTC。

图 1-33. Pin Manager: Grid View



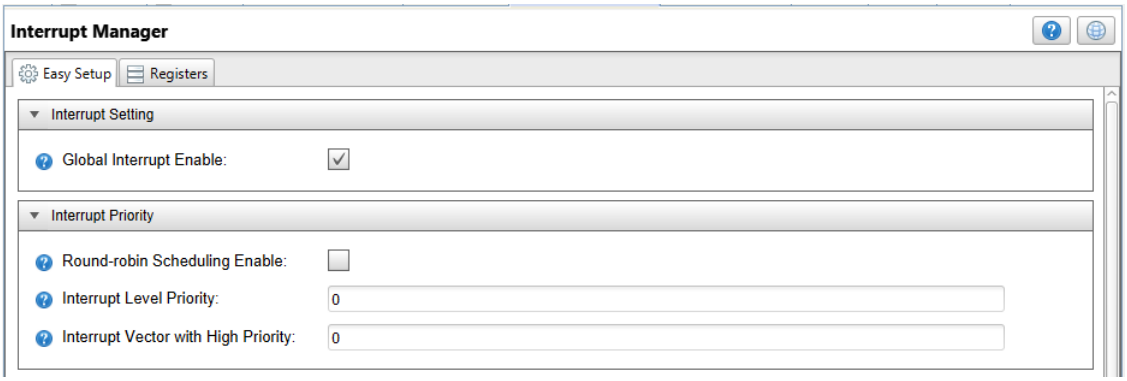
h. 在 Pin Module（引脚模块）中，允许 PC2 引脚的电平变化中断，以检测两个边沿。

图 1-34. Pin Module 配置



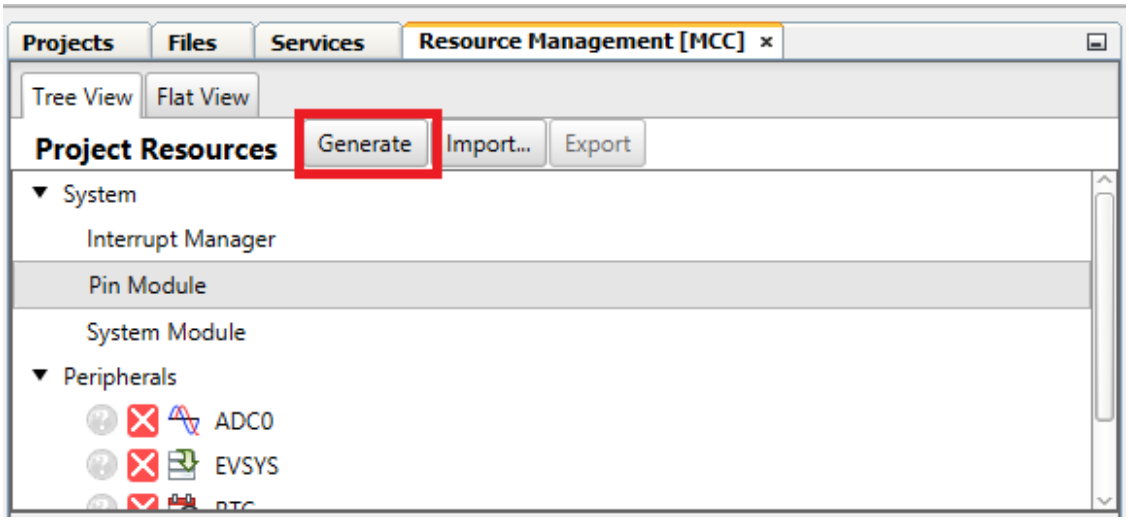
i. 在 Interrupt Manager 中允许全局中断。

图 1-35. Interrupt Manager 配置



j. 按下 **Generate** 按钮。

图 1-36. Generate 按钮



3. 向项目中添加代码。

a. 向 main.c 文件中添加以下代码：

```
#include "mcc_generated_files/mcc.h"
#include <avr/sleep.h>


void GPIO_Interrupt(void);

uint16_t volatile result;

int main(void)
{

    SYSTEM_Initialize();
    ADC0.MUXPOS = ADC_MUXPOS_TEMPSENSE_gc;
    PORTC_PC2_SetInterruptHandler(GPIO_Interrupt);

    while (1){

        sleep_cpu();

    }
}

void GPIO_Interrupt(void)
{
    result = ADC0_GetConversionResult();
}
```

信息： SYSTEM_Initialize()在 mcc.c 中定义，ADC0_GetConverisonResult()在 adc.c 中定义，PORTC_PC2_SetInterruptHandler()在 pin_manager.c 中定义。

- SYSTEM_Initialize()用于设置 CPU 和外设的所有配置寄存器。该函数由 MCC 生成。
- 在 main 函数中，将寄存器的 ADC0.MUXPOS 置 1，因为事件系统会立即启动转换，此时无法使能通道。
- PORTC_PC2_SetInterruptHandler(GPIO_Interrupt)用于设置当触发引脚 PC2 上的电平变化中断时将要调用的函数。
- ADC0_GetConversionResult()用于返回上次转换的结果。

b. 在位于 ***Source Files → MCC Generated files → src*** 内的 pin_manager.c 文件中，将引脚方向的代码替换为以下内容：

```
    PORTA.DIR = 0xFF;
    PORTB.DIR = 0xFF;
    PORTC.DIR = 0x3F;
    PORTD.DIR = 0xFF;
    PORTE.DIR = 0xFF;
    PORTF.DIR = 0xFF;
```

信息： 此操作是为了防止悬空引脚干扰功耗读数。

c. 在工具栏中，按下 **Clean and Build** 按钮，然后验证程序编译是否正确无误。

图 1-37. Clean and Build

View Code Examples on GitHub
Click to browse repositories

1.6 对器件进行编程并比较效率

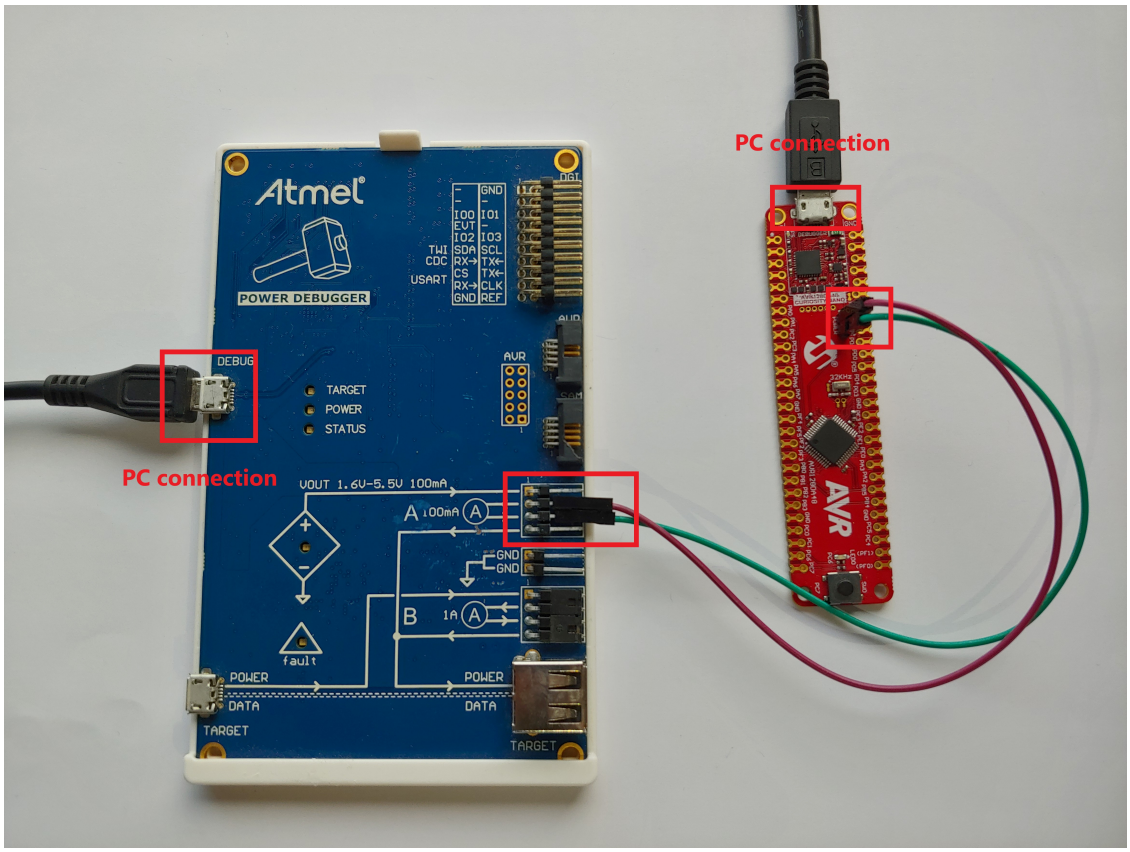
任务：将项目装载到单片机上，并获取功耗读数。

1. 从软件实现程序中获取功耗读数。
   a. 将 Curiosity Nano 板连接到计算机，并将两个焊接引脚连接在一起。
   b. 将软件实现程序设置为主程序后，在工具栏中按下 **Make and Program Device**（编译并编程器件）按钮。此时可能会弹出一个窗口，要求用户选择将用作编程器的工具。选择 AVR128DA48 Curiosity Nano 板。

图 1-38. Make and Program Device

   c. 将功率调试器连接到计算机。将第一个引脚（距离 Nano 嵌入式调试器最近的引脚）连接到 A 通道电流表的内向箭头，将第二个引脚（距离单片机更近的引脚）连接到 A 通道电流表的外向箭头。

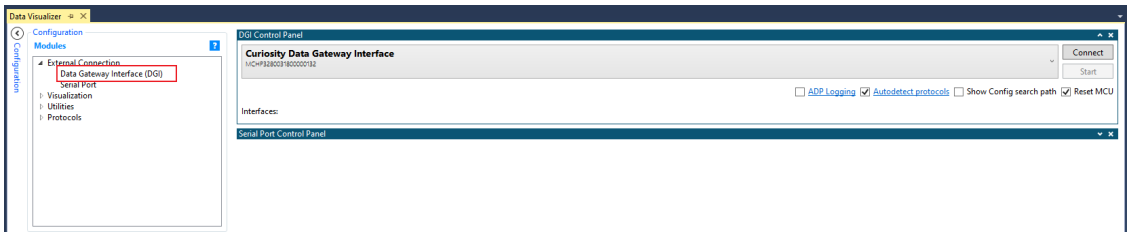
图 1-39. 硬件连接



   d. 在 Atmel Studio 中，转至 *Tools → Data Visualizer*（工具 → 数据可视化器）。

e. 随后将出现两个框。选中名称为 **DGI (Data Gateway Interface) Control Panel**（DGI（数据网关接口）控制面板）的框。如果该框没有出现，可以访问左侧的 Configuration（配置）菜单，在 *Modules → External Connection*（模块 → 外部连接）部分，可以找到 Data Gateway Interface（数据网关接口）。

图 1-40. Data Gateway Interface

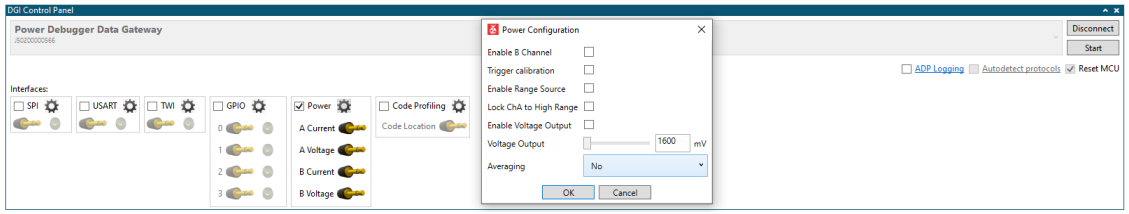


f. 按下 **Connect**（连接），随即出现一个标有“Power”（功率）的框。选中该框，然后按下滚轮以打开菜单。禁止 B 通道。

⚠WARNING 如果 Power 字段未显示，请断开电路板的连接，然后重新连接。

图 1-41. 连接 DGI

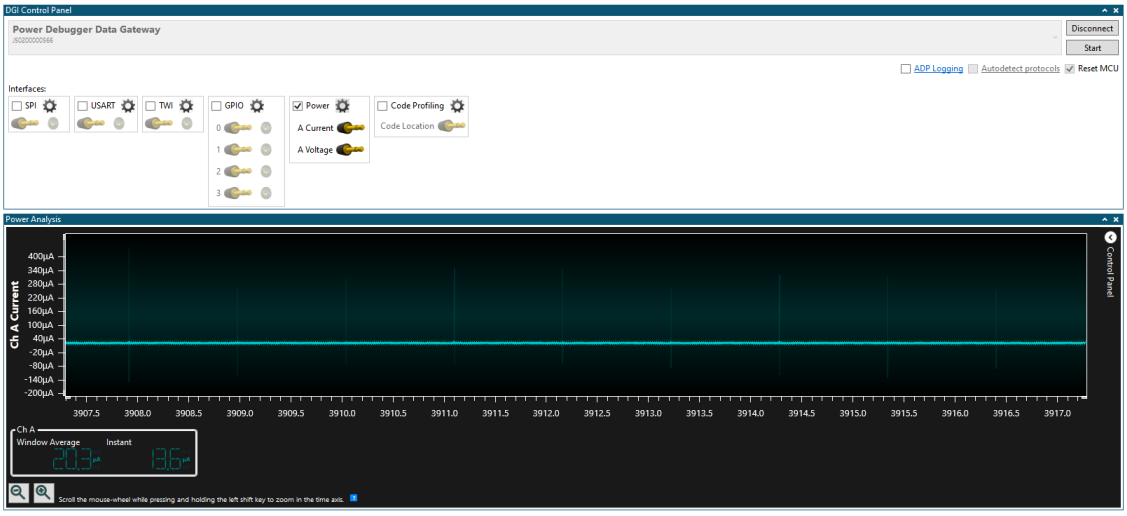
图 1-42. 功率设置



g. 当正确完成所有配置后，按下 **Start**（启动），随即出现一个新框，其中实时显示功耗。

提示：打开框左侧的控制面板，调整显示内容。取消选中 Show Zero（显示零点）将会使画面更清晰。

图 1-43. 开始监视



h. 在读数稳定后，按下 **Stop**（停止），然后断开连接。含功率测量结果的框仍然保持打开状态。

2. 从事件系统实现程序中获取功耗读数。

a. 将 Curiosity 板上的两个焊接引脚连接在一起。

b. 将事件系统实现项目设置为主程序后，在 MPLAB X 的工具栏中按下 **Make and Program Device** 按钮。

图 1-44. Make and Program Device

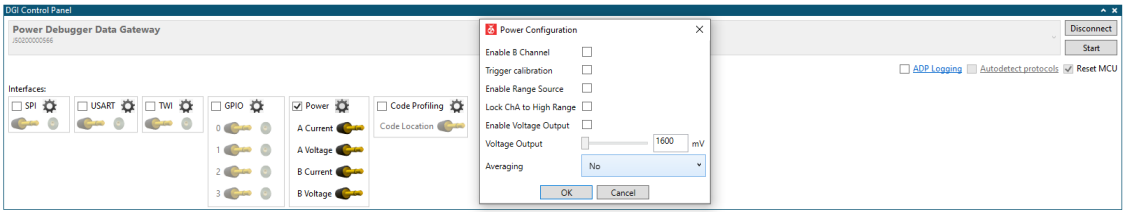
c. 将 Curiosity Nano 板连接到功率调试器。有关详细信息和图片，请参见本章的步骤 1.3。

d. 返回 Atmel Studio 中的 Data Visualizer，并在 DGI 面板上按下 **Connect**。

图 1-45. Data Gateway Interface

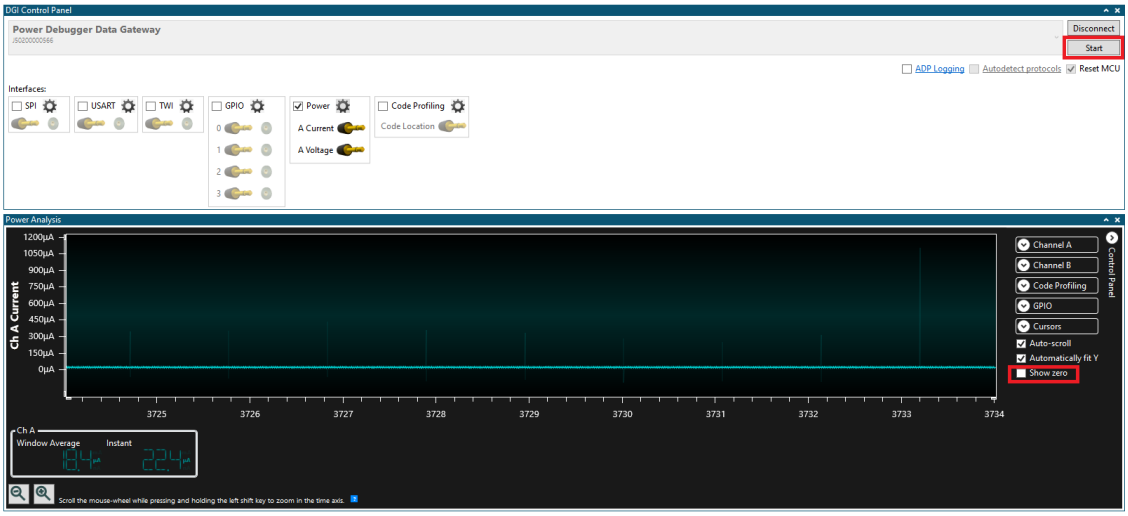
e. 选中 Power 框，然后编辑设置，以禁止 B 通道。

图 1-46. 功率设置



f. 按下 **Start**，随即出现一个显示实时功耗的新框。禁止 Show Zero 将会使画面更清晰。

图 1-47. 开始监视

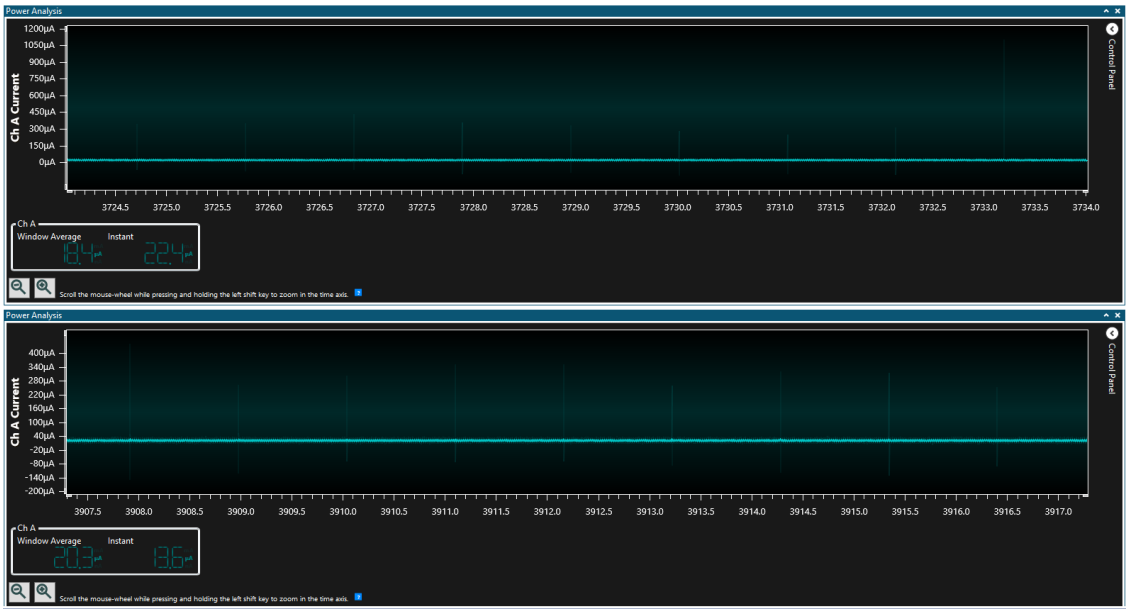


g. 在读数稳定后，按下 **Stop** 即可产生一个固定结果。

3. 比较这两个测量结果。

a. 两幅功耗图都在同一个屏幕中打开。现在可以对两幅图进行分析，以观察程序的行为方式。

图 1-48. 比较软件实现与 EVSYS



信息：上面的部分显示的是事件系统实现的功耗，下面的部分显示的是中断实现的功耗。从图中可以看出，差异相当小，约为 1.5 µA。这是由于单片机有很长一段时间处于休眠状态。

2. 任务 2：比较用于 ADC 的软件累加与硬件累加

在此项任务中，将比较两种不同的 ADC 累加技术的功耗。在 ADC 中累加结果可以起到提高结果准确性的作用，同时能够最大限度地降低噪声的影响。

第一种方法将反复触发转换，直到 128 次转换全部完成。这些结果将添加到一个用作累加器的变量中。

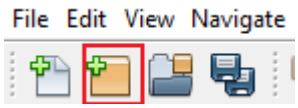
第二种方法利用了 ADC 的内置累加器的优势。它将使用 ADC 的累加转换模式来运行 128 次转换，并将结果添加到硬件累加器中。

2.1 创建软件实现项目

任务：配置要在本项目中使用的硬件模块，并向项目中添加代码。

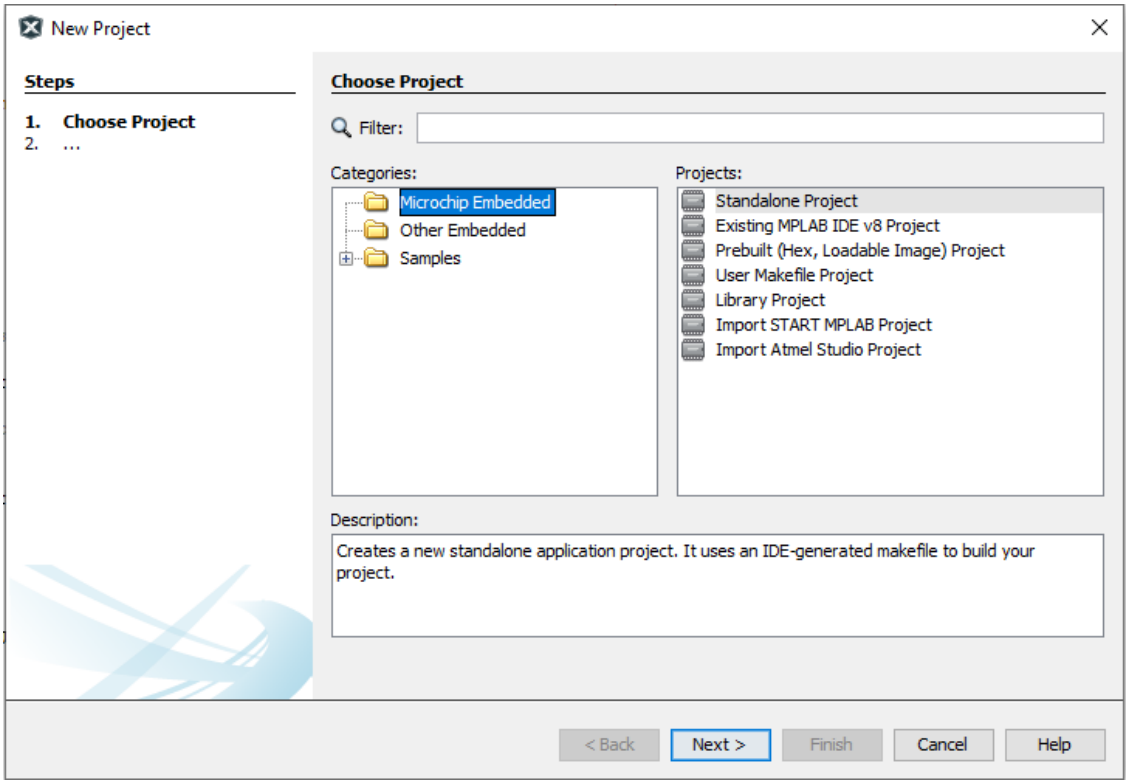
1. 为 AVR128DA48 新建一个 MPLAB X 项目。
   a. 打开 MPLAB X。
   b. 选择 *File → New Project* 或 **New Project** 按钮。

   图 2-1. New Project

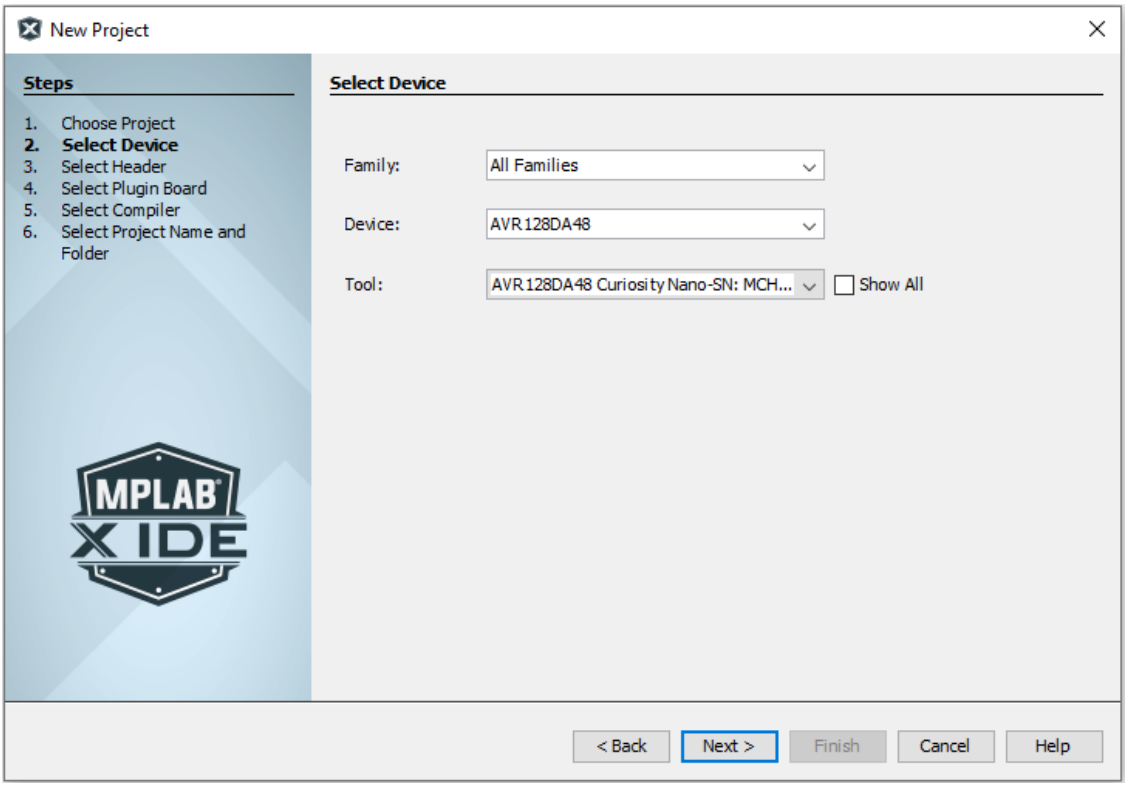
   c. 单击 **Next**（默认选择 Microchip Embedded → Standalone Project）。

   图 2-2. 项目类型

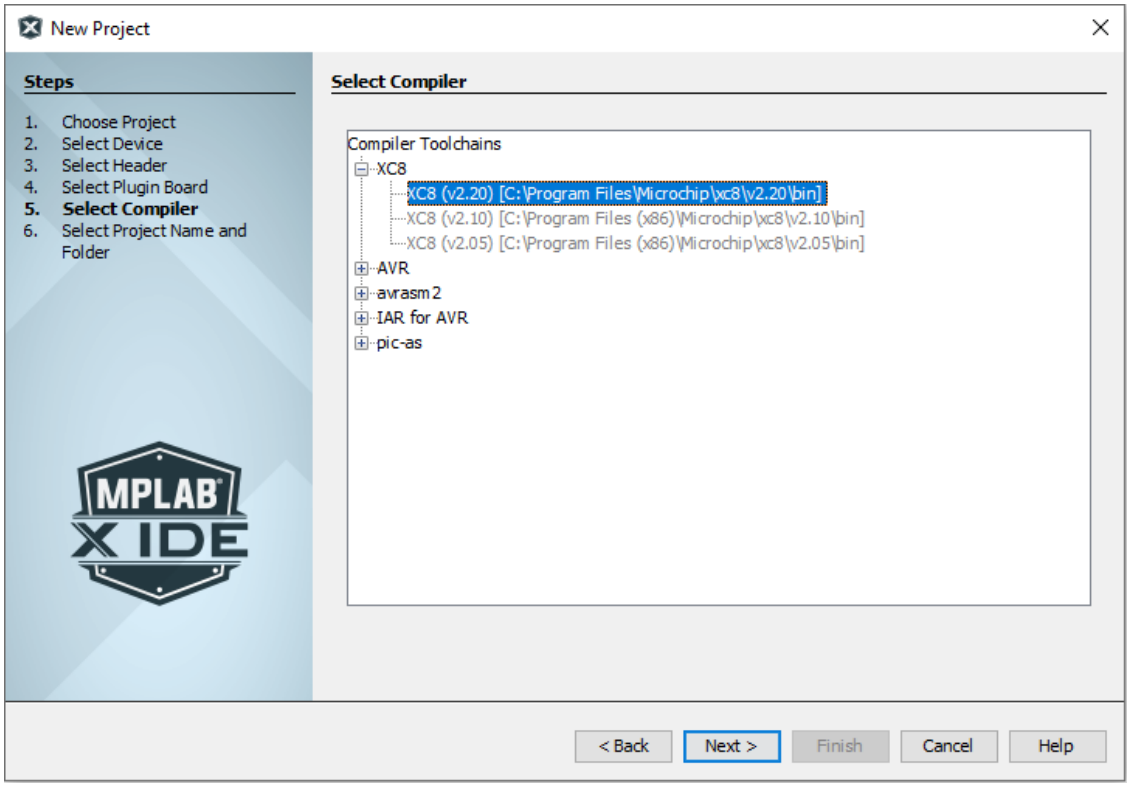
d. 在 Device 字段中搜索：AVR128DA48。在 Tool 类别中，如果 **Curiosity Nano 板**已连接到计算机，则选择该电路板，否则选择 **None**。单击 **Next**。

图 2-3. 器件选择



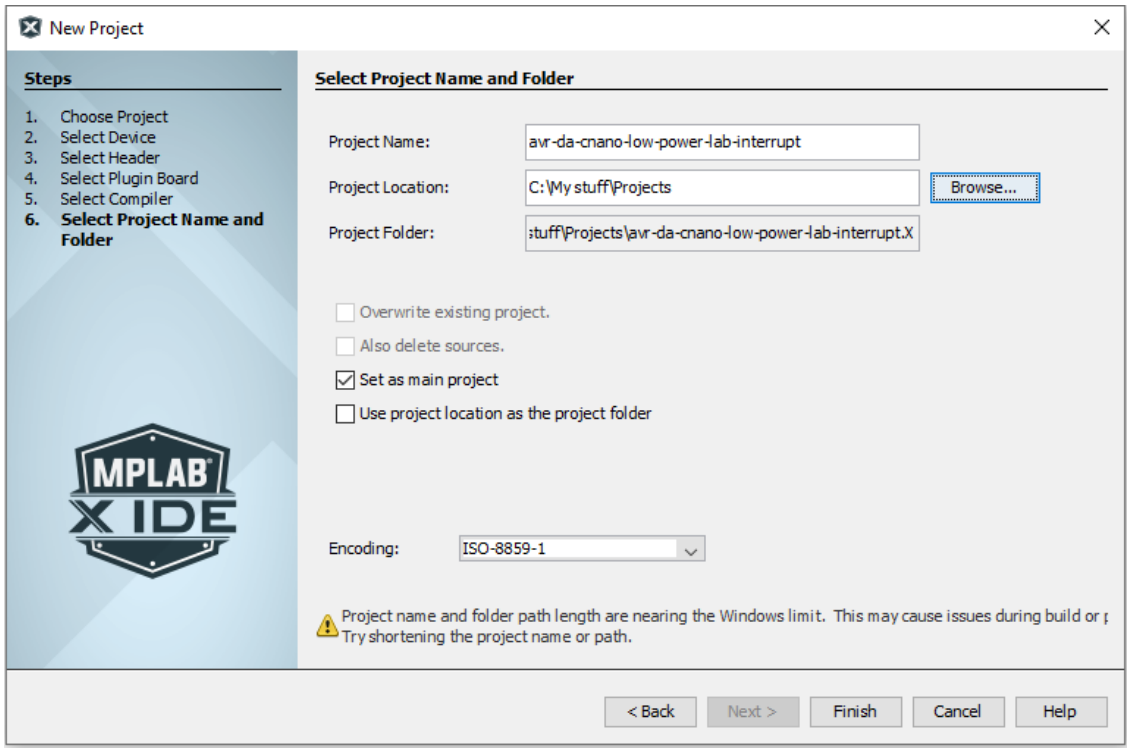
e. 选择 **XC8 2.20 编译器**，然后单击 **Next**。

图 2-4. 编译器选择



f. 为项目命名（以及确定保存位置），然后单击 **Finish**。

图 2-5. Project Name

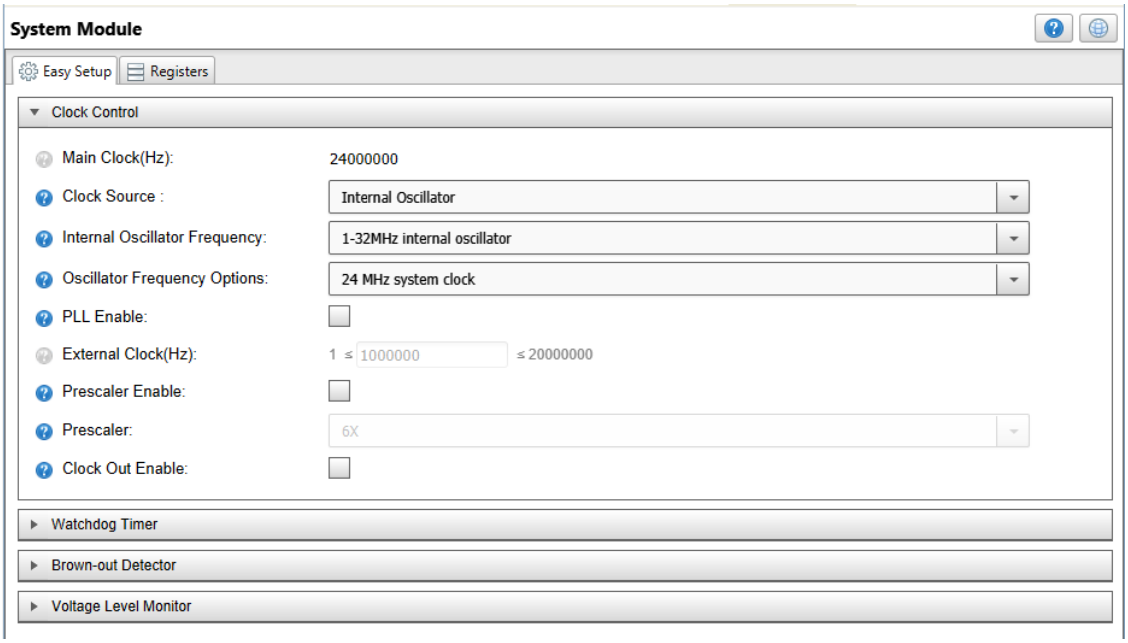


2. 打开 MPLAB 代码配置器（MCC）并配置外设。

a. 配置系统模块，使其基于 24 MHz 的内部振荡器运行。

信息：时钟源将使用内部振荡器，频率为 24 MHz。必须禁止预分频器选项。

图 2-6. System Module 配置

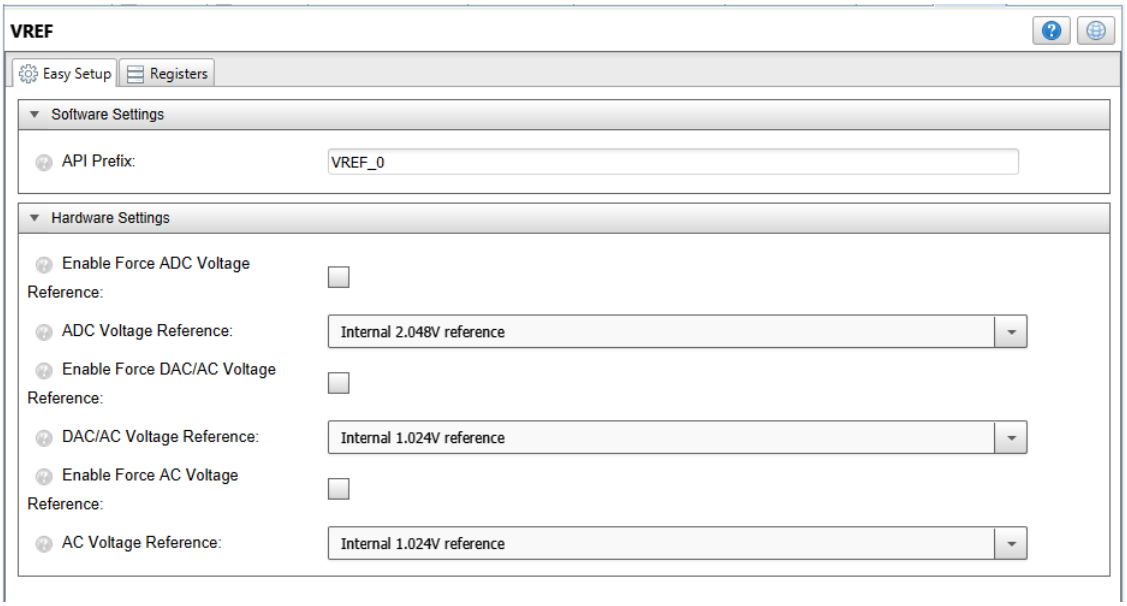


b. 从器件资源中添加 VREF 模块，并将其配置为向 ADC 提供 2.048V 的参考电压。

信息：必须将 ADC 参考电压配置为内部 2.048V 参考电压。请勿使能强制 ADC 电压选项。

信息：VREF 用于使能单片机的温度传感器。

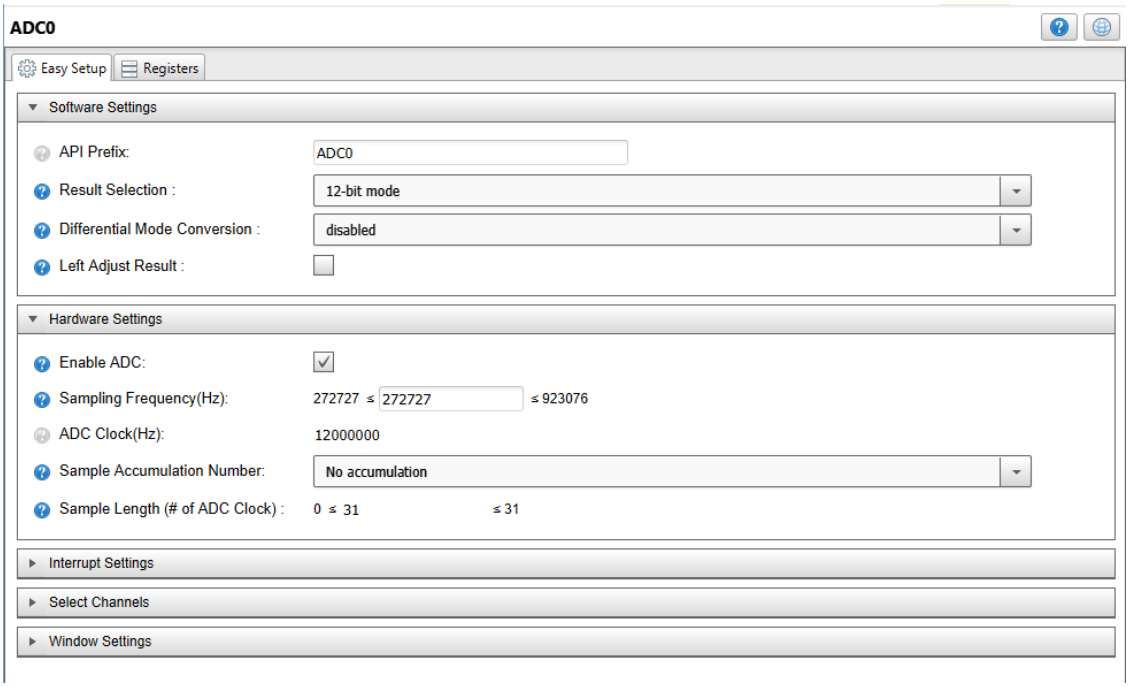
图 2-7. VREF 配置



c. 从器件资源中添加 ADC 模块，并将其配置为 12 位模式、右对齐结果、无累加和采样长度为 31 个节拍。

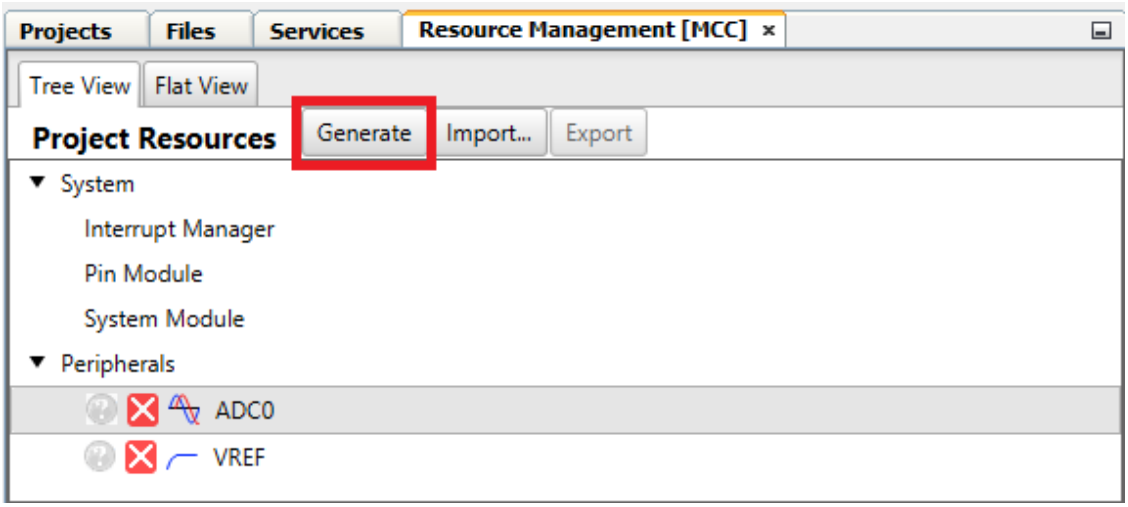
信息：必须禁止右移结果选项。采样长度在 **Hardware Settings** 选项卡中选择。12 位模式为默认设置。无累加选择也是默认设置。

图 2-8. ADC 配置



d. 按下 **Generate** 按钮。

图 2-9. Generate 按钮



3. 向项目中添加代码。
   a. 向 main.c 文件中添加以下代码。

```
#include "mcc_generated_files/mcc.h"

uint32_t accumulator = 0;

int main(void)
{

    SYSTEM_Initialize();

    while (1){
        accumulator = 0;
        for(uint8_t i = 0; i<128; i++)
        {
            ADC0_StartConversion(ADC_MUXPOS_TEMPSENSE_gc);
            while(!ADC0_IsConversionDone())
            {
                ;
            }
            accumulator += ADC0_GetConversionResult();
        }
    }
}
```

信息： SYSTEM_Initialize()在 mcc.c 中定义，ADC0_StartConversion(channel)、ADC0_IsConversionDone()和 ADC0_GetConversionResult()在 adc0.c 中定义。

- SYSTEM_Initialize()函数用于根据 MCC 中的设置来配置单片机和外设。该函数是由 MCC 生成的，并且需要在程序开始时调用。
- ADC0_StartConversion(channel)用于在指定的通道上启动 ADC 转换
- ADC0_IsConversionDone()用于返回转换的状态：0 表示正在进行，1 表示已完成
- ADC0_GetConversionResult()用于返回上次转换的结果
- for 循环将启动转换、等待转换结束和将结果添加到累加器中这三个步骤重复 128 次

   b. 在位于 *Source Files → MCC Generated files → src* 内的 pin_manager.c 文件中，将引脚方向的代码替换为以下内容：

```
    PORTA.DIR = 0xFF;
    PORTB.DIR = 0xFF;
    PORTC.DIR = 0x3F;
```

```
PORTD.DIR = 0xFF;
PORTE.DIR = 0xFF;
PORTF.DIR = 0xFF;
```

信息：此操作是为了防止悬空引脚干扰功耗读数。

c. 在工具栏中，按下 **Clean and Build** 按钮，然后验证程序编译是否正确无误。

图 2-10. Clean and Build

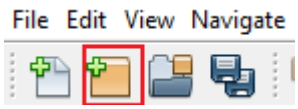
View Code Example on GitHub
Click to browse repositories

2.2 创建硬件实现项目

任务：配置要在本项目中使用的硬件模块，并向项目中添加代码。

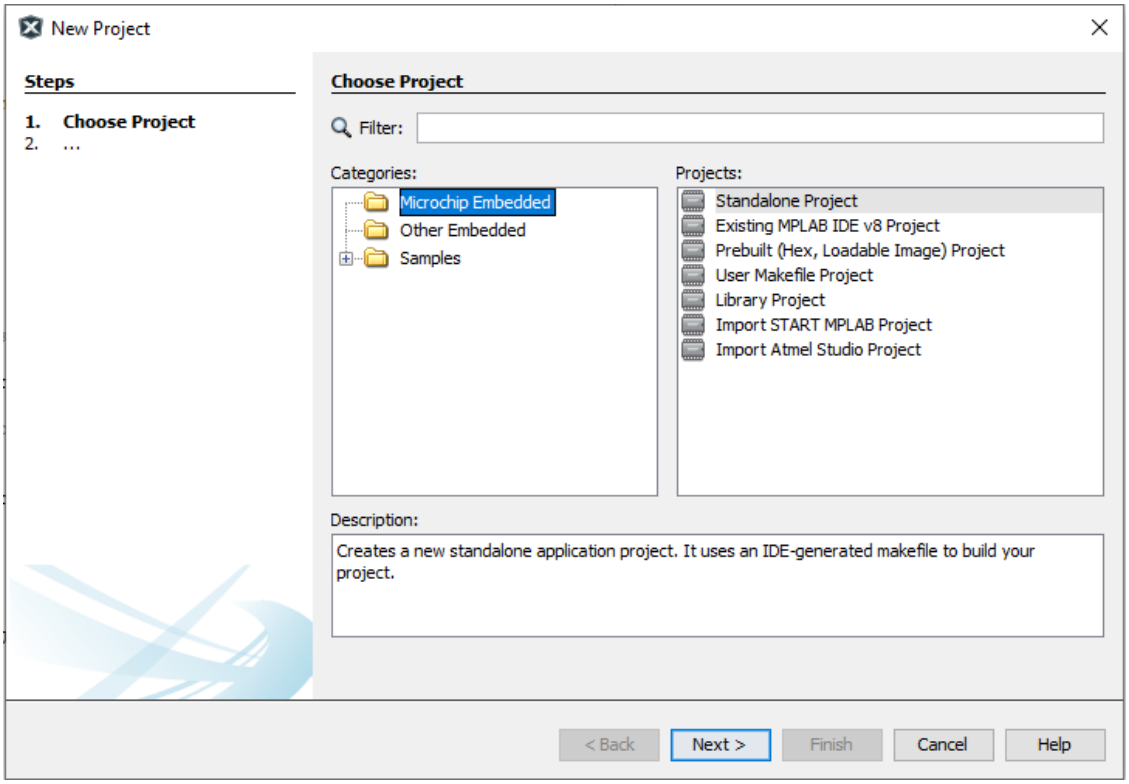
1. 为 AVR128DA48 新建一个 MPLAB X 项目。
   a. 打开 MPLAB X。
   b. 选择 *File → New Project* 或 **New Project** 按钮。

   图 2-11. New Project

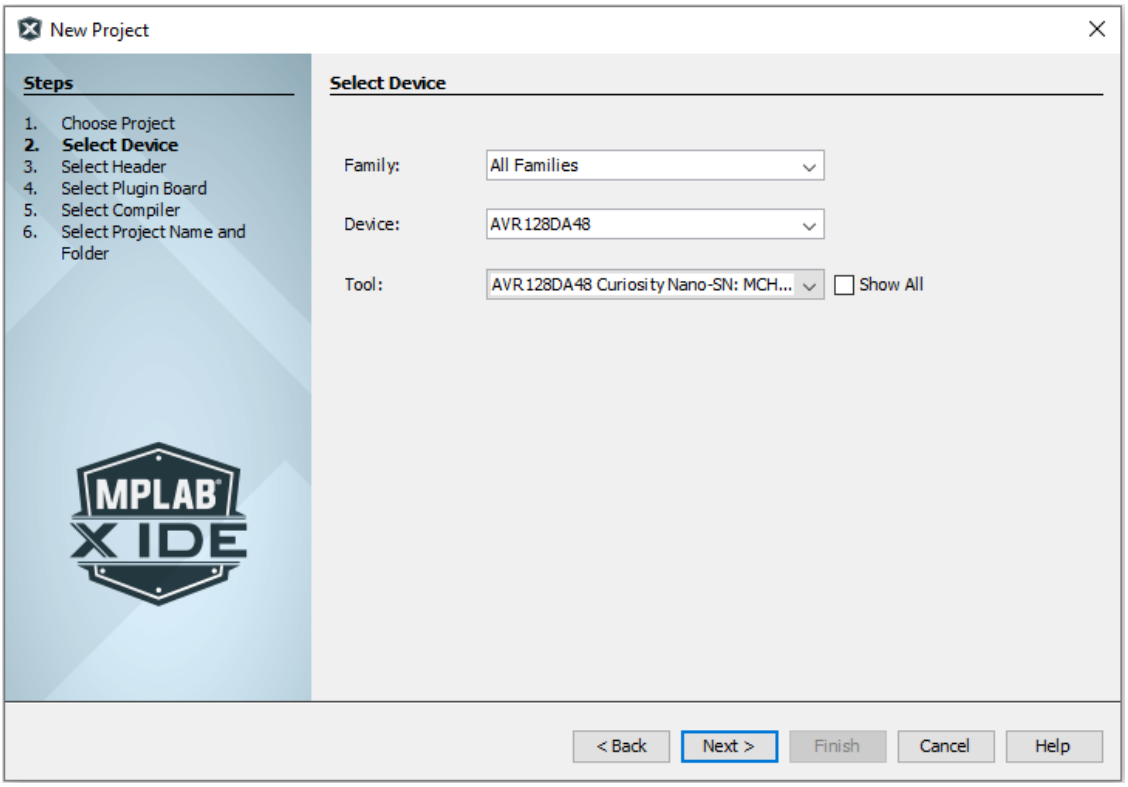
   c. 单击 **Next**（默认选择 Microchip Embedded → Standalone Project）。

图 2-12. 项目类型



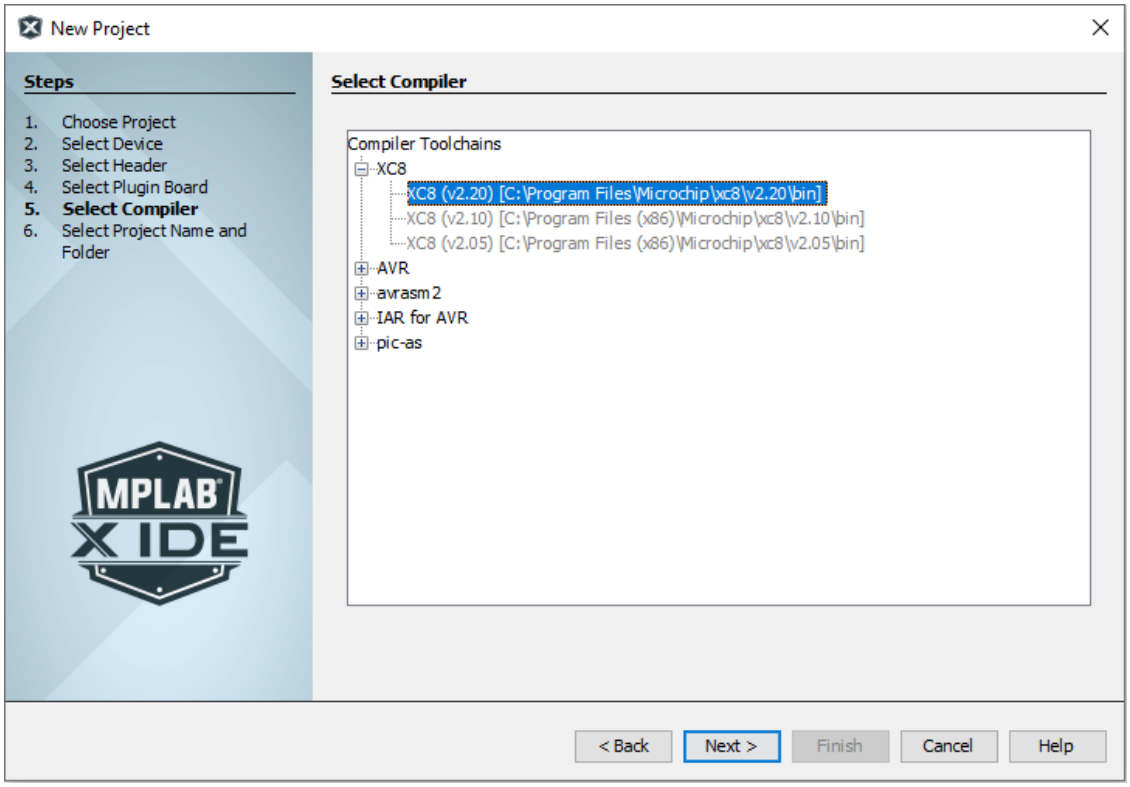
d. 在 Device 字段中搜索：AVR128DA48。在 Tool 类别中，如果 **Curiosity Nano 板**已连接到计算机，则选择该电路板，否则选择 **None**。单击 **Next**。

图 2-13. 器件选择



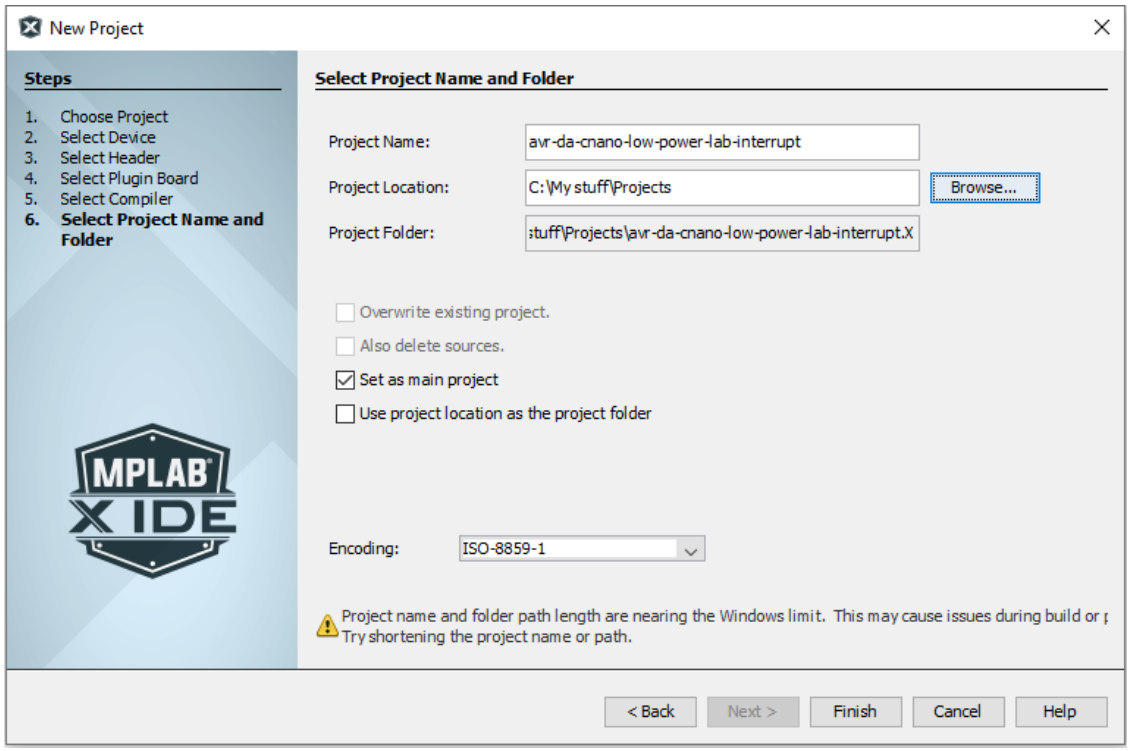
e. 选择 **XC8 2.20 编译器**，然后单击 **Next**。

图 2-14. 编译器选择



f. 为项目命名（以及确定保存位置），然后单击 **Finish**。

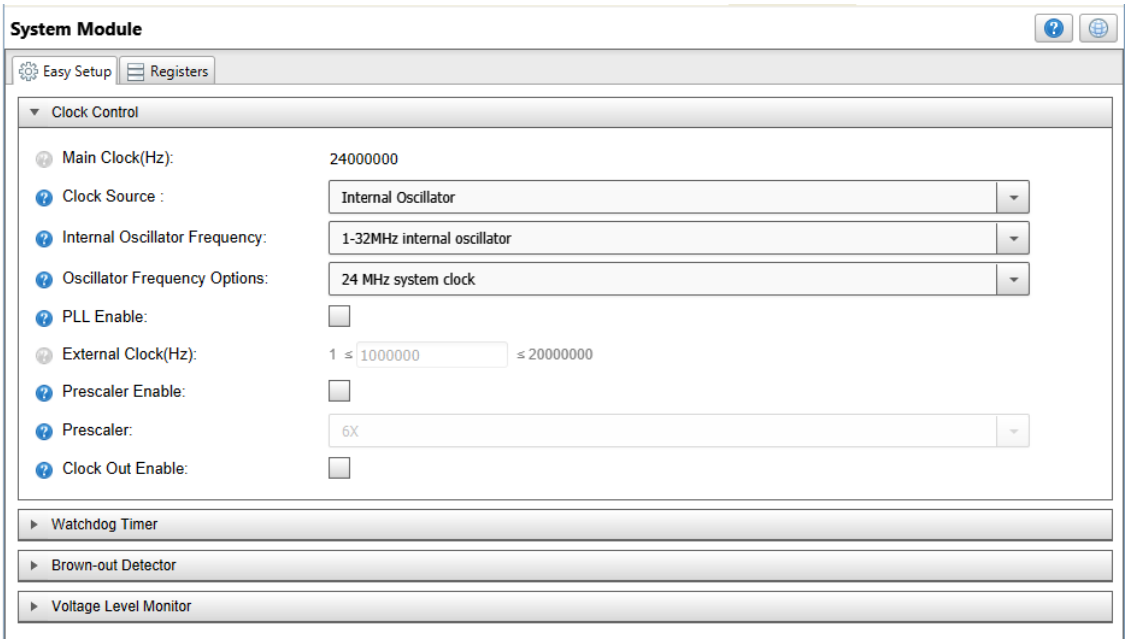
图 2-15. Project Name



2. 打开 MPLAB 代码配置器（MCC）并配置外设。
   a. 配置系统模块，使其基于 24 MHz 的内部振荡器运行。

信息：时钟源将使用内部振荡器，频率为 24 MHz。必须禁止预分频器选项。

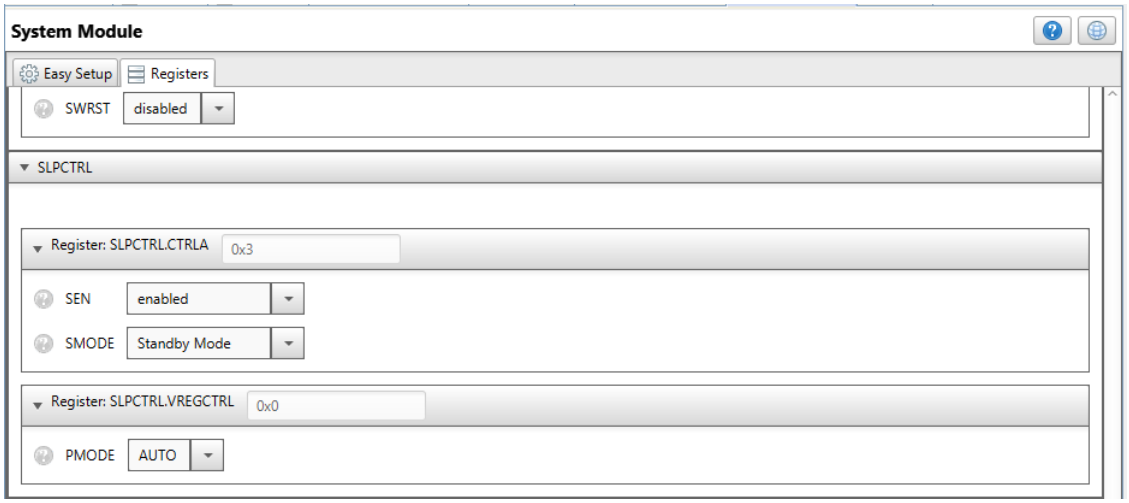
图 2-16. System Module 配置



   b. 配置休眠选项，以使能休眠，并将模式设为待机。

信息：转至 System Module 的 **Registers** 页面，向下滚动至 SLPCTRL。在 SLPCTRL.CTRLA 寄存器中，修改要使能的选项和待机模式。

图 2-17. 休眠配置

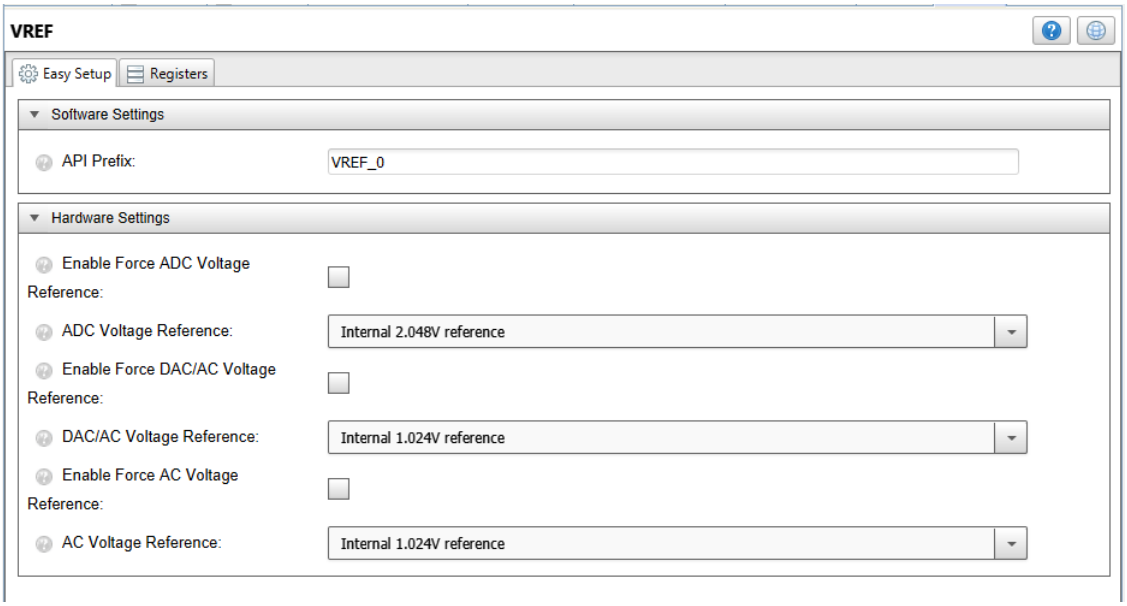


c. 从器件资源中添加 VREF 模块，并将其配置为向 ADC 提供 2.048V 的参考电压。

信息：必须将 ADC 参考电压配置为内部 2.048V 参考电压。请勿使能强制 ADC 电压选项。

信息：VREF 用于使能单片机的温度传感器。

图 2-18. VREF 配置



d. 从器件资源中添加 ADC 模块，并将其配置为 12 位模式、右对齐结果、128 个结果累加、采样长度为 31 个节拍和待机模式下运行（RUNSTBY）。

信息：必须禁止右移结果选项。采样长度和采样累加数量需要在 **Hardware Settings** 选项卡中选择。12 位模式为默认设置。RUNSTBY 位在 CTRLA 寄存器中，可在寄存器视图中更改。

图 2-19. ADC 配置

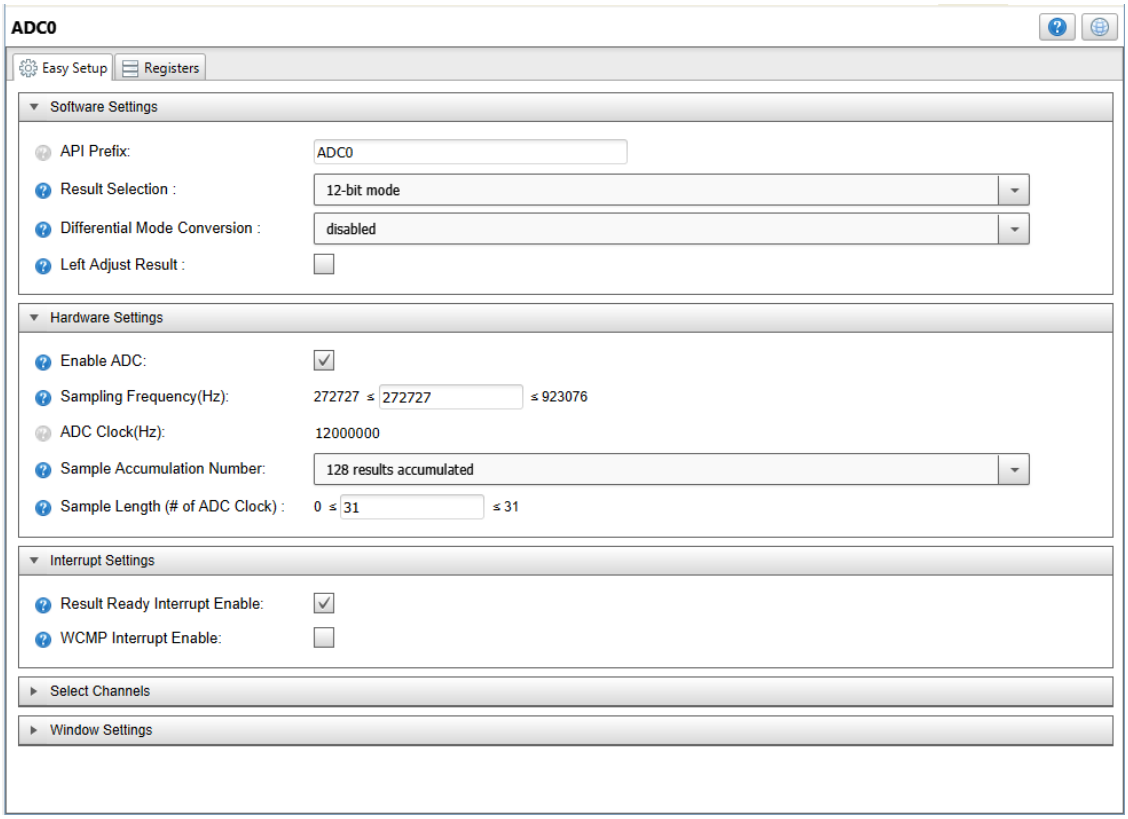
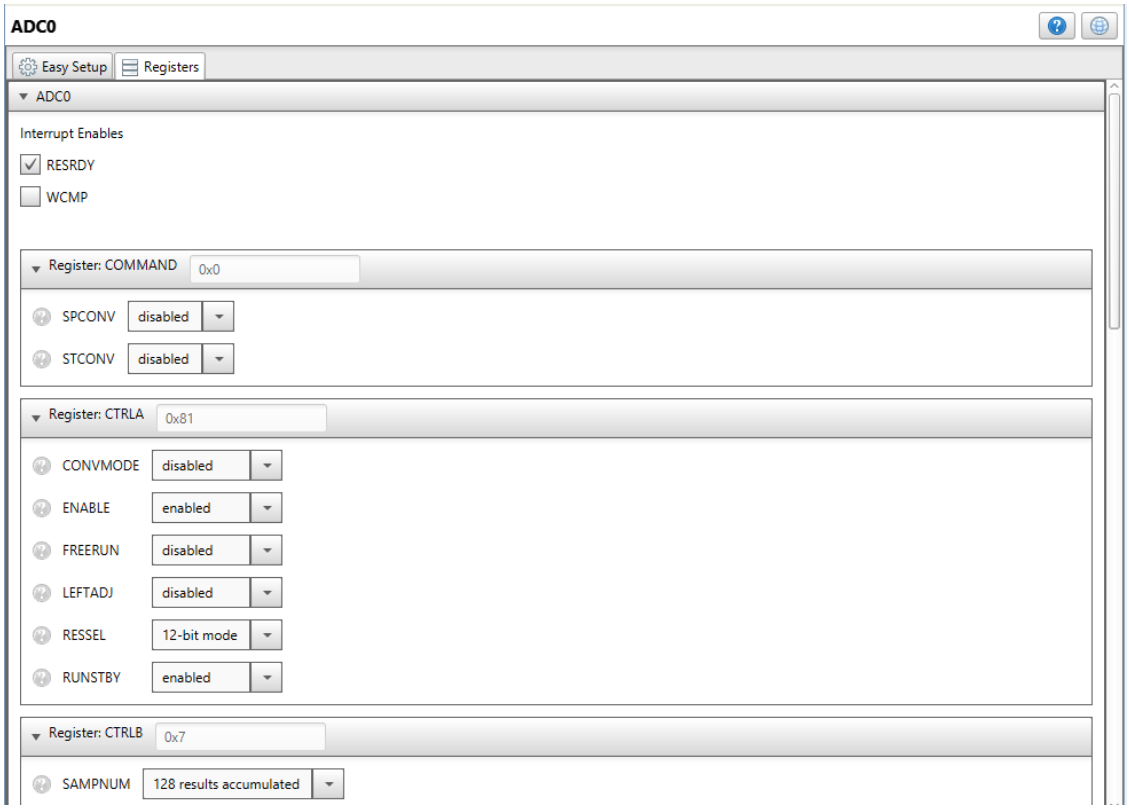


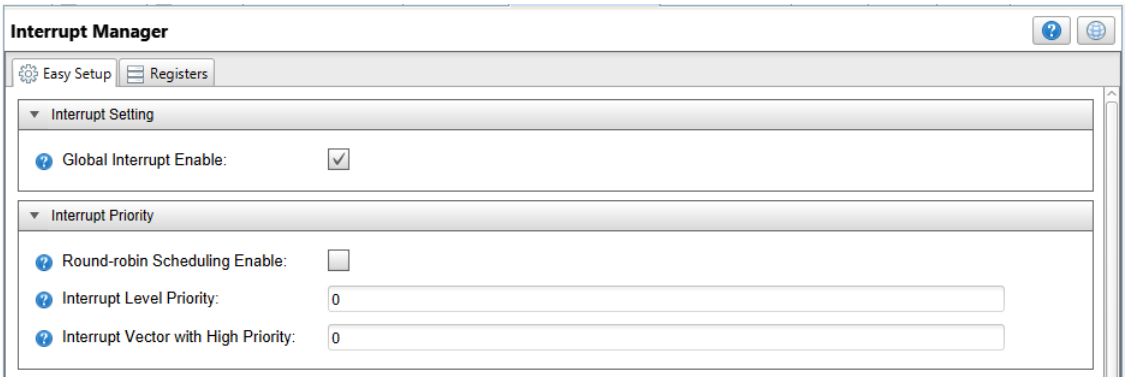
图 2-20. 待机模式下运行 ADC 配置



e. 允许 ADC 的结果就绪中断和全局中断。

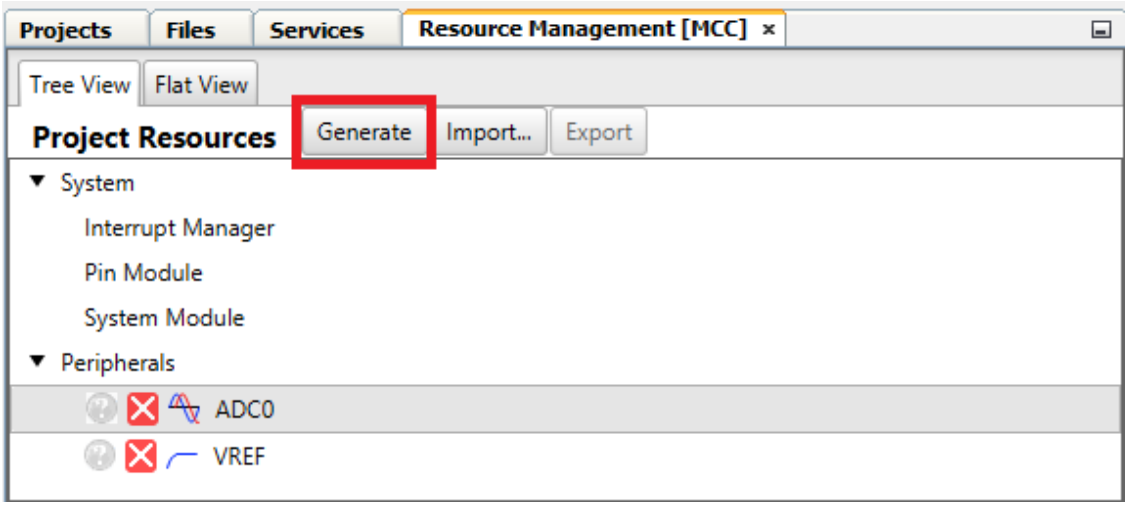
信息：在 ADC0 模块中，在 Interrupt Setting 中激活 Result Ready Interrupt Enable（结果就绪中断允许）。此设置见上一步的图片。在 Interrupt Manager 中，允许全局中断。

图 2-21. Interrupt Manager 配置



f. 按下 **Generate** 按钮。

图 2-22. Generate 按钮



3. 向项目中添加代码。

a. 向 `main.c` 文件中添加以下代码：

```
#include "mcc_generated_files/mcc.h"
#include <avr/sleep.h>

uint16_t result;

int main(void)
{

    SYSTEM_Initialize();
    while (1){
        ADC0_StartConversion(ADC_MUXPOS_TEMPSENSE_gc);
        sleep_cpu();
        result = ADC0_GetConversionResult();
    }
}
```

信息： SYSTEM_Initialize()在 mcc.c 中定义，ADC0_StartConversion(channel)和 ADC0_GetConversionResult()在 adc0.c 中定义。

- SYSTEM_Initialize()函数用于根据 MCC 中的设置来配置单片机和外设。该函数是由 MCC 生成的，并且需要在程序开始时调用。
- ADC0_StartConversion(channel)用于在指定的通道上启动 ADC 转换。
- ADC0_GetConversionResult()用于返回上次转换的结果。
- 程序启动一次转换，然后进入休眠状态。结果就绪中断将唤醒程序，读取结果，然后启动新的转换。

b. 在位于 *Source Files → MCC Generated files → src* 内的 pin_manager.c 文件中，将引脚方向的代码替换为以下内容：

```
PORTA.DIR = 0xFF;
PORTB.DIR = 0xFF;
PORTC.DIR = 0x3F;
PORTD.DIR = 0xFF;
PORTE.DIR = 0xFF;
PORTF.DIR = 0xFF;
```

信息： 此操作是为了防止悬空引脚干扰功耗读数。

c. 在工具栏中，按下 **Clean and Build** 按钮，然后验证程序编译是否正确无误。

图 2-23. Clean and Build

2.3 对器件进行编程并比较功率效率

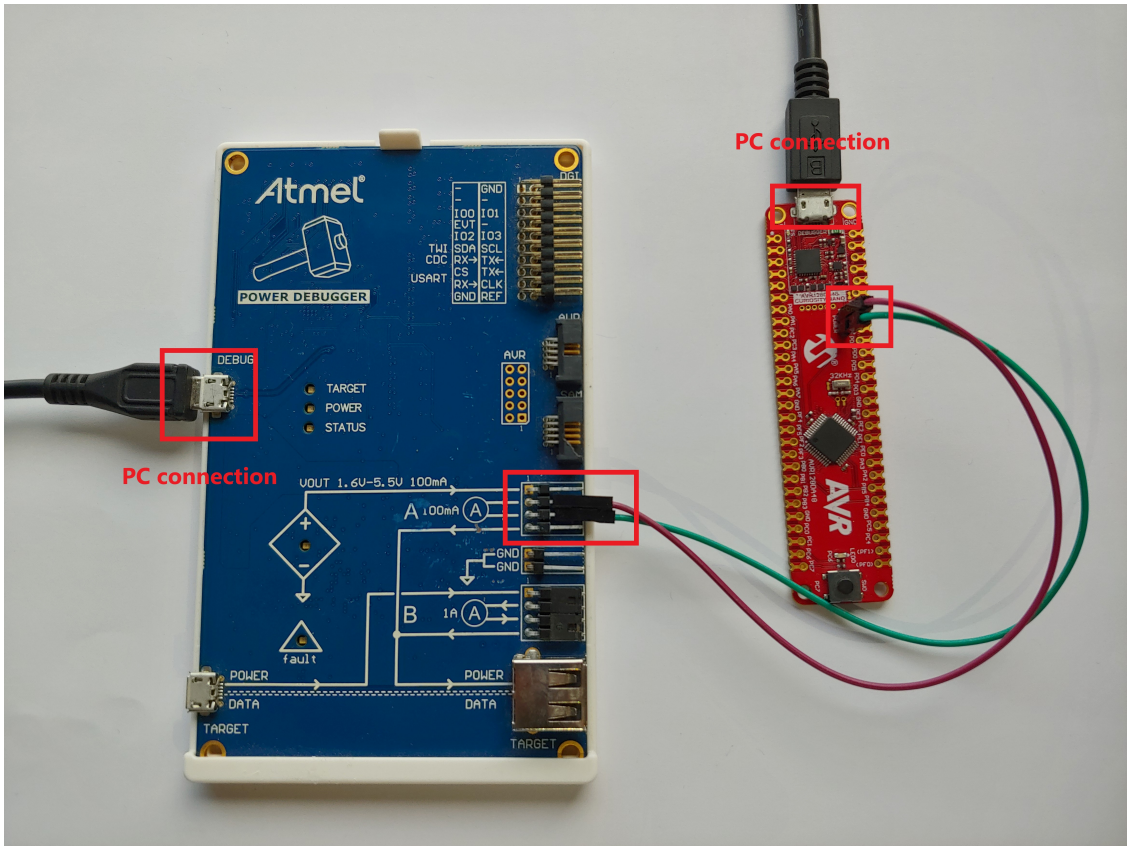
任务： 将项目装载到单片机上，并获取功耗读数。

1. 对器件进行编程，并从软件实现程序中获取功耗读数。
   a. 将 Curiosity Nano 板连接到计算机。将两个焊接引脚连接在一起。将软件累加程序设置为主程序后，在工具栏中按下 **Make and Program Device** 按钮。此时可能会弹出一个窗口，要求用户选择要进行编程的工具。选择 AVR128DA48 Curiosity Nano 板。

   图 2-24. Make and Program Device

   b. 打开 Atmel Studio，然后将功率调试器连接到计算机。将 Curiosity Nano 板连接到功率调试器的 A 通道电流表。将距离 Nano 嵌入式调试器最近的引脚连接到 A 电流表的内向箭头，将距离单片机最近的引脚连接到电流表的外向箭头。

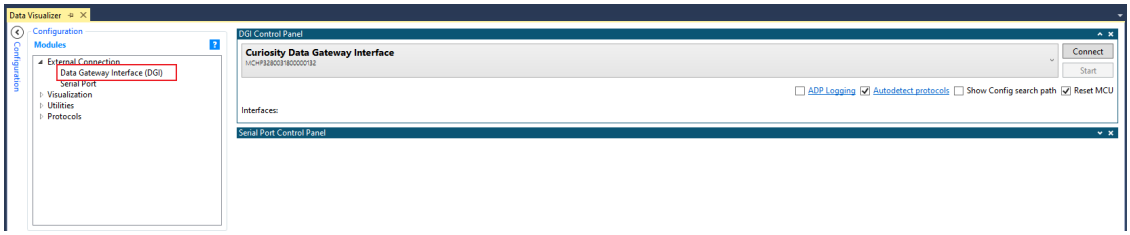
图 2-25. 硬件设置



c. 在 Atmel Studio 中，转至 *Tools → Data Visualizer*。

d. 随后将出现两个框。选中名称为 DGI (Data Gateway Interface) Control Panel 的框。如果该框没有出现，可以访问左侧的 Configuration 菜单，在 *Modules → External Connection* 部分，可以找到 Data Gateway Interface。

图 2-26. Data Gateway Interface

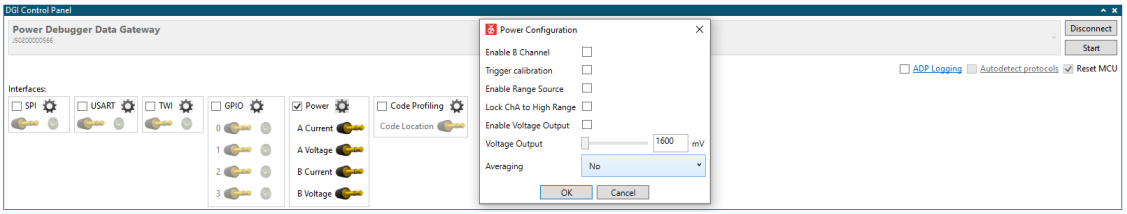


e. 按下 **Connect**，随即出现一个标有“Power”的框。选中该框，然后按下滚轮以打开菜单。禁止 B 通道。

WARNING 如果 Power 字段未显示，请断开电路板的连接，然后重新连接。

图 2-27. DGI 连接

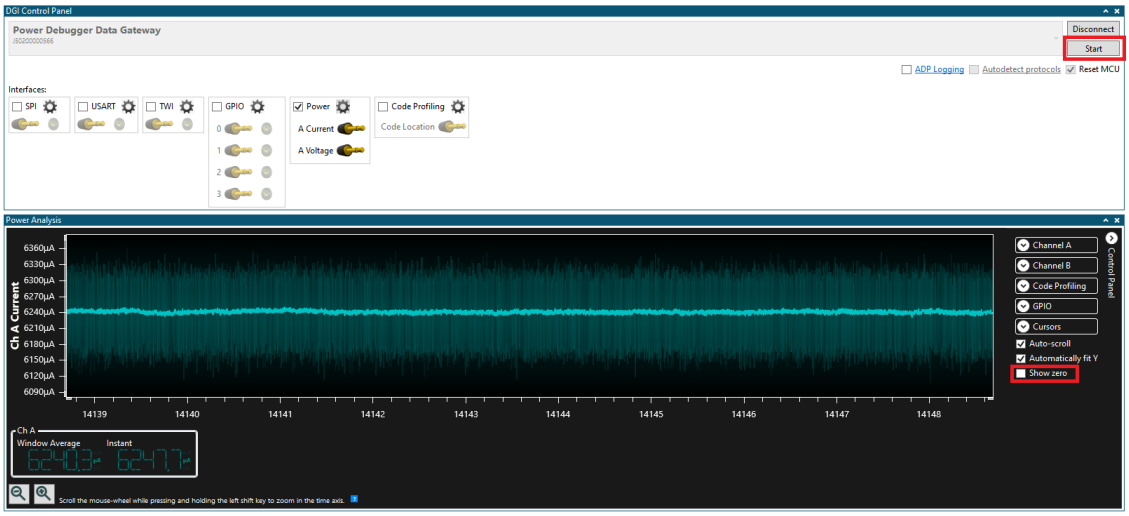
图 2-28. 功率设置



f. 当正确完成所有配置后，按下 **Start**，随即出现一个新框，其中实时显示功耗。

提示：要调整显示内容，请打开框左侧的控制面板。取消选中 Show Zero 将会使画面更清晰。

图 2-29. 开始监视



g. 在读数稳定后，按下 **Stop**，然后断开连接。含功率测量结果的框仍然保持打开状态。

2. 从硬件累加实现程序中获取功耗读数。
   a. 将 Curiosity 板上的两个焊接引脚连接在一起。
   b. 将硬件累加项目设置为主程序后，在 MPLAB X 的工具栏中按下 **Make and Program Device** 按钮。

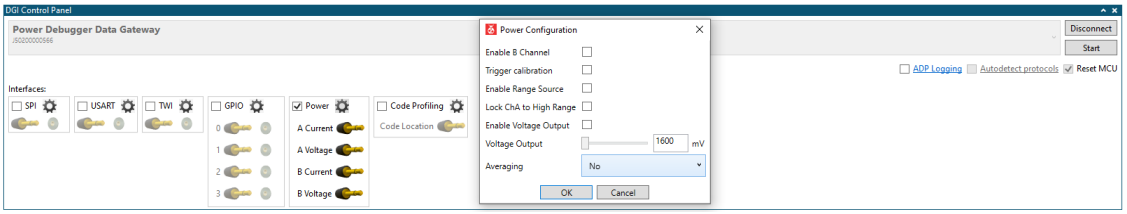
图 2-30. Make and Program Device

   c. 将 Curiosity Nano 板的两个焊接引脚连接到功率调试器。请参见步骤 1.2。
   d. 返回 Atmel Studio 中的 Data Visualizer，并在 DGI 面板上按下 **Connect**。

图 2-31. DGI 连接

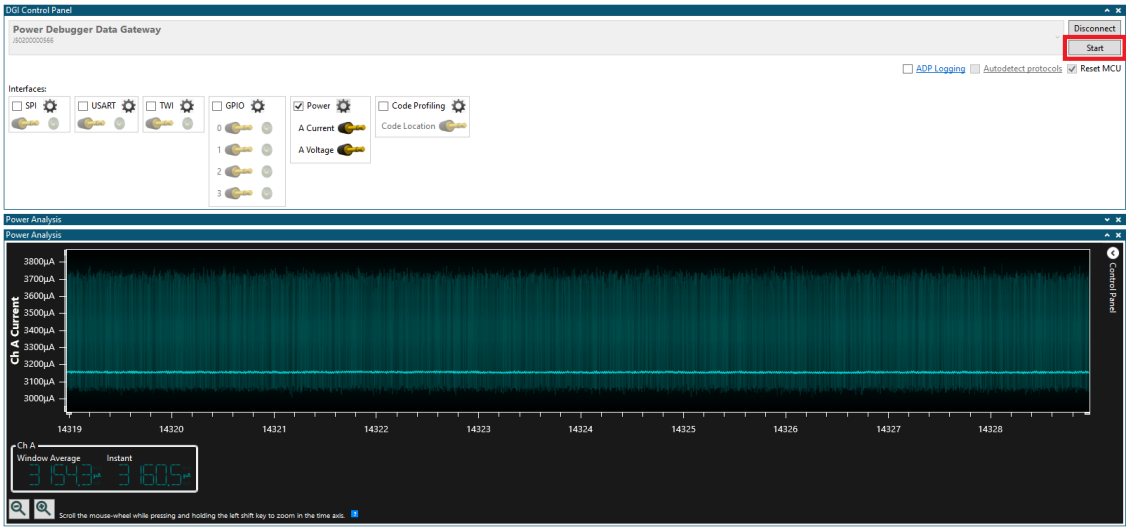
   e. 选中 Power 框，然后编辑设置，以禁止 B 通道。

图 2-32. 功率设置



f. 按下 **Start**，随即出现一个显示实时功耗的新框。禁止 Show Zero 将会使画面更清晰。

图 2-33. 开始监视

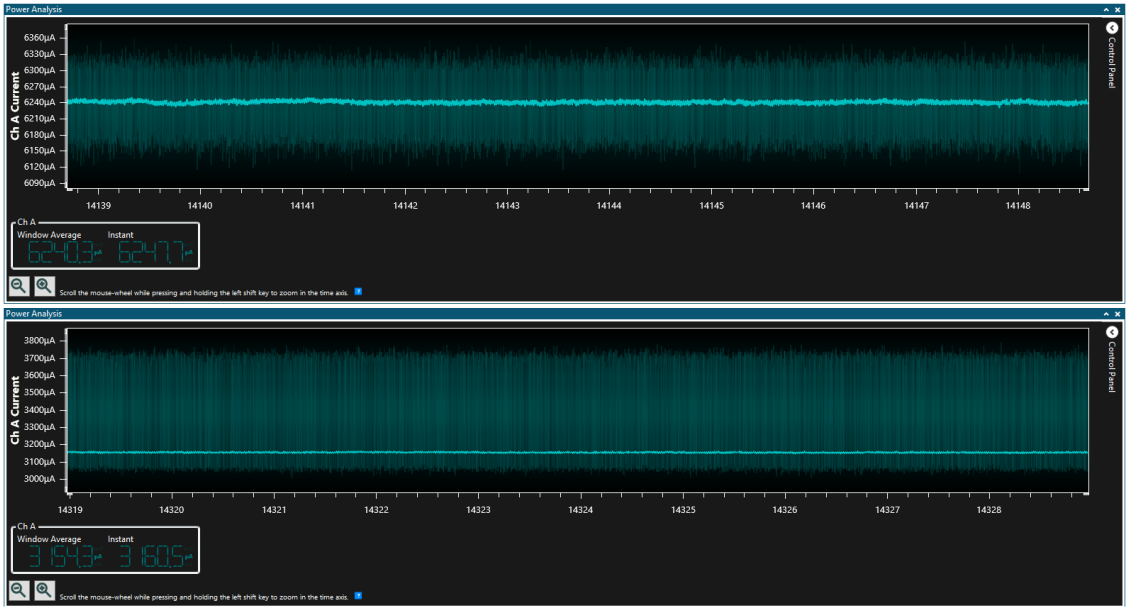


g. 在读数稳定后，按下 **Stop** 即可产生一个固定结果。

3. 比较这两个测量结果。

a. 两幅功耗图都在同一个屏幕中打开。现在可以对两幅图进行分析，以观察程序的行为方式。

图 2-34. 比较软件累加与硬件累加



信息：上面的部分显示的是软件累加，下面的部分是硬件累加。从图中可以看出，两者之间的差异为 3.1 mA，这使得硬件实现的效率几乎是软件实现的二倍。

3. 版本历史

版本	日期	说明
A	2020 年 8 月	文档初始版本

Microchip 网站

Microchip 网站（www.microchip.com）为客户提供在线支持。客户可通过该网站方便地获取文件和信息。我们的网站提供以下内容：

- **产品支持**——数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持**——常见问题解答（FAQ）、技术支持请求、在线讨论组以及 Microchip 设计伙伴计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

产品变更通知服务

Microchip 的产品变更通知服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请访问 www.microchip.com/pcn，然后按照注册说明进行操作。

客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师（ESE）
- 技术支持

客户应联系其代理商、代表或 ESE 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 www.microchip.com/support 获得网上技术支持。

Microchip 器件代码保护功能

请注意以下有关 Microchip 产品代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术规范。
- Microchip 确信：在正常使用且符合工作规范的情况下，Microchip 系列产品非常安全。
- Microchip 注重并积极保护其知识产权。严禁任何试图破坏 Microchip 产品代码保护功能的行为，这种行为可能会违反《数字千年版权法案》（Digital Millennium Copyright Act）。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。

法律声明

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc.及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc.的英文原版文档。

本出版物及其提供的信息仅适用于 Microchip 产品，包括设计、测试以及将 Microchip 产品集成到您的应用中。以其他任何方式使用这些信息都将被视为违反条款。本出版物中的器件应用信息仅为您提供便利，将来可能会发生更新。如需额外的支持，请联系当地的 Microchip 销售办事处，或访问 https://www.microchip.com/en-us/support/design-help/client-supportservices。

Microchip“按原样”提供这些信息。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对非侵权性、适销性和特定用途的适用性的暗示担保，或针对其使用情况、质量或性能的担保。

在任何情况下，对于因这些信息或使用这些信息而产生的任何间接的、特殊的、惩罚性的、偶然的或间接的损失、损害或任何类型的开销，Microchip 概不承担任何责任，即使 Microchip 已被告知可能发生损害或损害可以预见。在法律允许的最大范围内，对于因这些信息或使用这些信息而产生的所有索赔，Microchip 在任何情况下所承担的全部责任均不超出您为获得这些信息向 Microchip 直接支付的金额（如有）。如果将 Microchip 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切损害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任。除非另外声明，在 Microchip 知识产权保护下，不得暗中或以其他方式转让任何许可证。

商标

Microchip 的名称和徽标组合、Microchip 徽标、Adaptec、AnyRate、AVR、AVR 徽标、AVR Freaks、BesTime、BitCloud、chipKIT、chipKIT 徽标、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi 徽标、MOST、MOST 徽标、MPLAB、OptoLyzer、PackeTime、PIC、picoPower、PICSTART、PIC32 徽标、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST 徽标、SuperFlash、Symmetricom、SyncServer、Tachyon、TempTrackr、TimeSource、tinyAVR、UNI/O、Vectron 和 XMEGA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的注册商标。

APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus 徽标、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、Vite、WinPath 和 ZL 均为 Microchip Technology Incorporated 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet 徽标、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的商标。

SQTP 是 Microchip Technology Incorporated 在美国的服务标记。

Adaptec 徽标、Frequency on Demand、Silicon Storage Technology 和 Symmcom 为 Microchip Technology Inc.在其他国家或地区的注册商标。

GestIC 是 Microchip Technology Inc.的子公司 Microchip Technology Germany II GmbH & Co. KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2022，Microchip Technology Incorporated，美国印刷，版权所有。

ISBN：978-1-6683-2484-4

质量管理体系

有关 Microchip 的质量管理体系的信息，请访问 www.microchip.com/quality。

全球销售及服务网点

美洲	亚太地区	亚太地区	欧洲
公司总部 2355 West Chandler Blvd. Chandler, AZ 85224-6199 电话：480-792-7200 传真：480-792-7277 技术支持： www.microchip.com/support 网址： www.microchip.com **亚特兰大** 德卢斯，佐治亚州 电话：678-957-9614 传真：678-957-1455 **奥斯汀，德克萨斯州** 电话：512-257-3370 **波士顿** 韦斯特伯鲁，马萨诸塞州 电话：774-760-0087 传真：774-760-0088 **芝加哥** 艾塔斯卡，伊利诺伊州 电话：630-285-0071 传真：630-285-0075 **达拉斯** 阿迪森，德克萨斯州 电话：972-818-7423 传真：972-818-2924 **底特律** 诺维，密歇根州 电话：248-848-4000 **休斯顿，德克萨斯州** 电话：281-894-5983 **印第安纳波利斯** 诺布尔斯维尔，印第安纳州 电话：317-773-8323 传真：317-773-5453 电话：317-536-2380 **洛杉矶** 米慎维荷，加利福尼亚州 电话：949-462-9523 传真：949-462-9608 电话：951-273-7800 **罗利，北卡罗来纳州** 电话：919-844-7510 **纽约，纽约州** 电话：631-435-6000 **圣何塞，加利福尼亚州** 电话：408-735-9110 电话：408-436-4270 **加拿大 - 多伦多** 电话：905-695-1980 传真：905-695-2078	**澳大利亚 - 悉尼** 电话：61-2-9868-6733 **中国 - 北京** 电话：86-10-8569-7000 **中国 - 成都** 电话：86-28-8665-5511 **中国 - 重庆** 电话：86-23-8980-9588 **中国 - 东莞** 电话：86-769-8702-9880 **中国 - 广州** 电话：86-20-8755-8029 **中国 - 杭州** 电话：86-571-8792-8115 **中国 - 香港特别行政区** 电话：852-2943-5100 **中国 - 南京** 电话：86-25-8473-2460 **中国 - 青岛** 电话：86-532-8502-7355 **中国 - 上海** 电话：86-21-3326-8000 **中国 - 沈阳** 电话：86-24-2334-2829 **中国 - 深圳** 电话：86-755-8864-2200 **中国 - 苏州** 电话：86-186-6233-1526 **中国 - 武汉** 电话：86-27-5980-5300 **中国 - 西安** 电话：86-29-8833-7252 **中国 - 厦门** 电话：86-592-2388138 **中国 - 珠海** 电话：86-756-3210040	**印度 - 班加罗尔** 电话：91-80-3090-4444 **印度 - 新德里** 电话：91-11-4160-8631 **印度 - 浦那** 电话：91-20-4121-0141 **日本 - 大阪** 电话：81-6-6152-7160 **日本 - 东京** 电话：81-3-6880-3770 **韩国 - 大邱** 电话：82-53-744-4301 **韩国 - 首尔** 电话：82-2-554-7200 **马来西亚 - 吉隆坡** 电话：60-3-7651-7906 **马来西亚 - 槟榔屿** 电话：60-4-227-8870 **菲律宾 - 马尼拉** 电话：63-2-634-9065 **新加坡** 电话：65-6334-8870 **台湾地区 - 新竹** 电话：886-3-577-8366 **台湾地区 - 高雄** 电话：886-7-213-7830 **台湾地区 - 台北** 电话：886-2-2508-8600 **泰国 - 曼谷** 电话：66-2-694-1351 **越南 - 胡志明市** 电话：84-28-5448-2100	**奥地利 - 韦尔斯** 电话：43-7242-2244-39 传真：43-7242-2244-393 **丹麦 - 哥本哈根** 电话：45-4485-5910 传真：45-4485-2829 **芬兰 - 埃斯波** 电话：358-9-4520-820 **法国 - 巴黎** 电话：33-1-69-53-63-20 传真：33-1-69-30-90-79 **德国 - 加兴** 电话：49-8931-9700 **德国 - 哈恩** 电话：49-2129-3766400 **德国 - 海尔布隆** 电话：49-7131-72400 **德国 - 卡尔斯鲁厄** 电话：49-721-625370 **德国 - 慕尼黑** 电话：49-89-627-144-0 传真：49-89-627-144-44 **德国 - 罗森海姆** 电话：49-8031-354-560 **以色列 - 若那那市** 电话：972-9-744-7705 **意大利 - 米兰** 电话：39-0331-742611 传真：39-0331-466781 **意大利 - 帕多瓦** 电话：39-049-7625286 **荷兰 - 德卢内市** 电话：31-416-690399 传真：31-416-690340 **挪威 - 特隆赫姆** 电话：47-72884388 **波兰 - 华沙** 电话：48-22-3325737 **罗马尼亚 - 布加勒斯特** 电话：40-21-407-87-50 **西班牙 - 马德里** 电话：34-91-708-08-90 传真：34-91-708-08-91 **瑞典 - 哥德堡** 电话：46-31-704-60-40 **瑞典 - 斯德哥尔摩** 电话：46-8-5090-4654 **英国 - 沃金厄姆** 电话：44-118-921-5800 传真：44-118-921-5820