
从8位AVR®到32位SAM D21 MCU的应用程序移植

本页面用作将应用程序从8位AVR单片机（MCU）移植到32位SAM MCU的参考，并举例说明了如何将应用程序从ATmega4809 MCU移植到SAM D21 MCU。

为了将应用程序移植到32位MCU，需使用SAM D21 Curiosity Nano评估工具包并在MPLAB® Harmony v3软件框架上开发应用程序。

SAM D21 Curiosity Nano评估工具包是一款用于评估SAM D21 MCU的硬件平台，由MPLAB X集成开发环境（Integrated Development Environment, IDE）和MPLAB Harmony v3软件开发框架支持。利用这款评估工具包，可轻松访问SAM D21 MCU的功能，从而将器件集成到自定义设计中。由于该评估工具包配有用于编程和调试的板上Nano调试器，因此可直接对SAM D21 MCU器件进行编程，无需使用外部工具。该评估工具包与Curiosity Nano基板（部件编号——AC164162）兼容，方便用户在使用SAM D21 MCU设计下一款创新产品时快速进行扩展与原型设计。

MPLAB Harmony v3是一款灵活的全集成嵌入式软件开发框架，适用于32位单片机（MCU）和微处理器（MPU）。MPLAB Harmony v3包括MPLAB Harmony配置器（MPLAB Harmony Configurator, MHC）工具、一组模块化外设库（PLIB）、驱动程序、系统服务、中间件和大量示例应用程序，旨在快速、轻松开发面向Microchip 32位PIC®和SAM器件的功能强大且高效的嵌入式软件。

本移植指南引用了使用MPLAB代码配置器（MPLAB Code Configurator, MCC）在ATmega4809 Curiosity Nano评估工具包上开发的现有应用程序“ATmega4809上的UART入门”，并讨论了使用MPLAB Harmony配置器（MHC）将现有应用程序移植到SAM D21 Curiosity Nano评估工具包的步骤。此外，可通过添加更多外设特定功能进一步扩展应用程序。

基于SAM D21 MCU开发的应用程序将执行以下操作：

- 利用PORT PLIB控制LED
- 利用外部中断控制器（External Interrupt Controller, EIC）PLIB和Systick PLIB在发生开关按下事件时切换LED状态
- 利用SERCOM（作为通用同步/异步收发器（Universal Synchronous Asynchronous Receiver Transmitter, USART））、STDIO库和直接存储器访问（Direct Memory Access, DMA）PLIB将信息消息打印在PC上运行的COM（串行）端口终端应用程序上并从串行终端接收命令
- 利用SERCOM（作为SPI）运行自环回测试

本指南有两种使用方法：

1. 从头开始创建项目：
 - 按照以下分步说明进行操作。
2. 以解决方案项目为例：
 - 编译解决方案项目并将其编程到SAM D21 Curiosity Nano评估工具包以观察预期行为。

以下项目符号提供了各个主题的连接：

- 使用的硬件
- 使用的软件/工具
- 硬件设置
- 说明
 - 概述
 - 安装MHC
 - 创建MPLAB Harmony v3项目
 - 配置时钟
 - 添加SERCOM（USART）外设
 - 生成代码
 - 添加代码
 - 编译并编程
 - 验证“Hello World”输出
 - 配置/使能格式化STDIO、USART接收和开关/LED功能
 - 验证格式化STDIO、USART接收和开关/LED功能
 - 扩展应用程序以使用基于DMA的传输配置SPI环回
 - 验证扩展功能——SPI环回

- 结果
- 分析
- 结论

使用的硬件:

- SAM D21 Curiosity Nano评估工具包
- 连接线

使用的软件/工具:

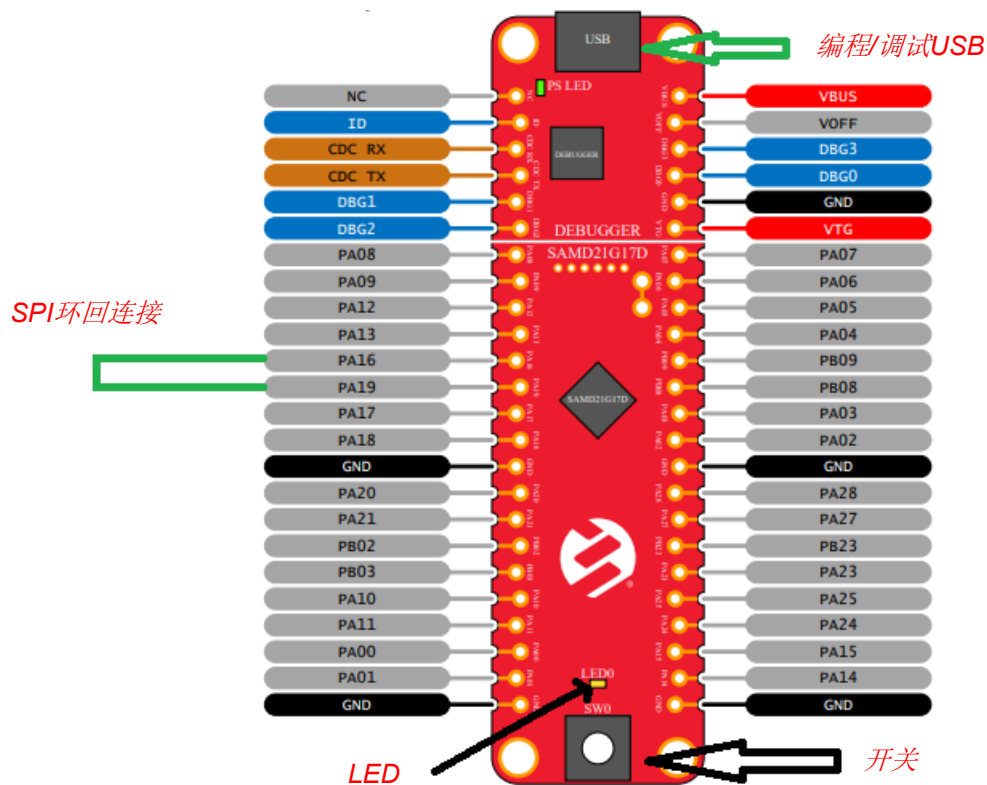
经验证, 该项目支持使用以下版本的软件工具:

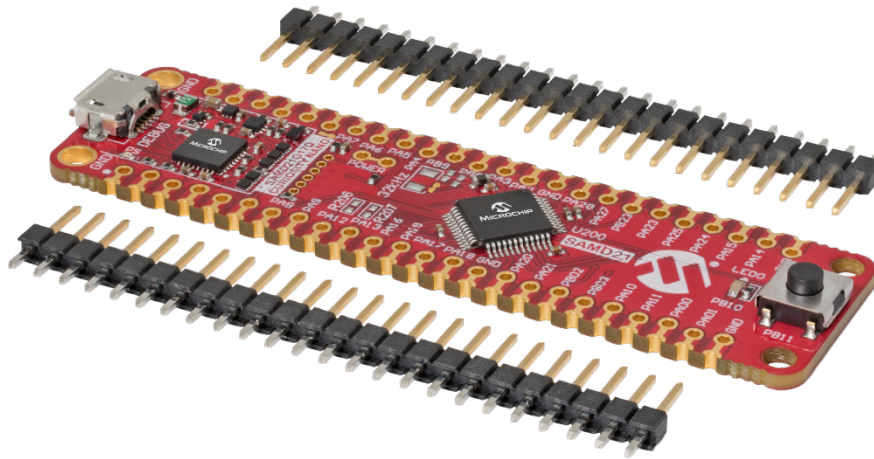
- MPLAB Harmony v3 “csp” 资源库v3.8.3
- MPLAB Harmony v3 “dev_packs” 资源库v3.8.0
- MPLAB Harmony v3 “mhc” 资源库v3.6.5
- MPLAB Harmony v3启动器插件v3.6.2
- MPLAB X IDE v5.45
- MPLAB XC32编译器v2.41
- 任何串行终端应用程序, 例如Tera Term终端应用程序

由于Microchip会定期更新工具, 因此在使用新版工具时, 可能会发现问题。如果发现项目无法正常运行, 并且怀疑是版本不兼容所致, 建议再次确认所用工具版本是否与项目测试阶段使用的版本一致。

硬件设置

- 将SAM D21 Curiosity Nano开发板作为USB设备连接到主机PC, 具体通过Type-A公头转micro-B USB线缆连接到micro-B USB (调试USB) 端口来实现
- 在SAM D21 Curiosity Nano开发工具包上的PA16与PA19之间连接一根线缆, 以进行SPI自环回测试





说明

概述

本移植指南使用的AVR ATmega4809应用程序引用用于“AVR上的UART入门”。

- [单击此处可查看/下载该应用程序。](#)

AVR ATmega4809应用程序展示了如何实现命令行接口。这样，单片机便可以通过USART接收控制命令。在该用例中，使用由MPLAB 数据可视化器发送的命令来控制LED。

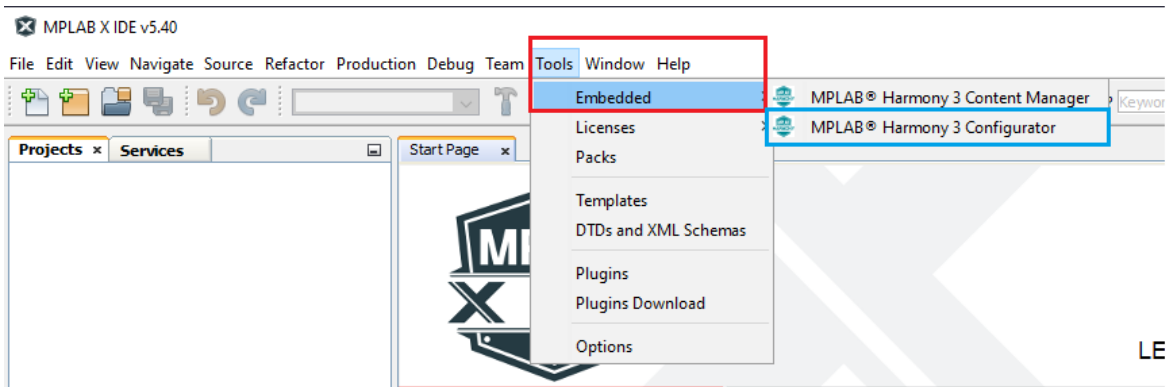
应用程序流程如下：

1. 配置系统时钟
将时钟配置为1 MHz（使用16 MHz内部振荡器）。
2. 配置USART3
将USART3的波特率配置为9600，并将帧格式配置为标准的8-N-1（8个数据位，无奇偶校验位，1个停止位）。
3. 配置引脚
配置USART TX和RX引脚以及LED引脚。
4. 实现STDIO接收和发送功能
实现STDIO库以与串行终端之间收发命令。
5. 读取并执行有效命令
从串行终端接收命令并执行。

请按照以下步骤开始移植到SAMD21，然后通过添加更多外设来扩展功能。

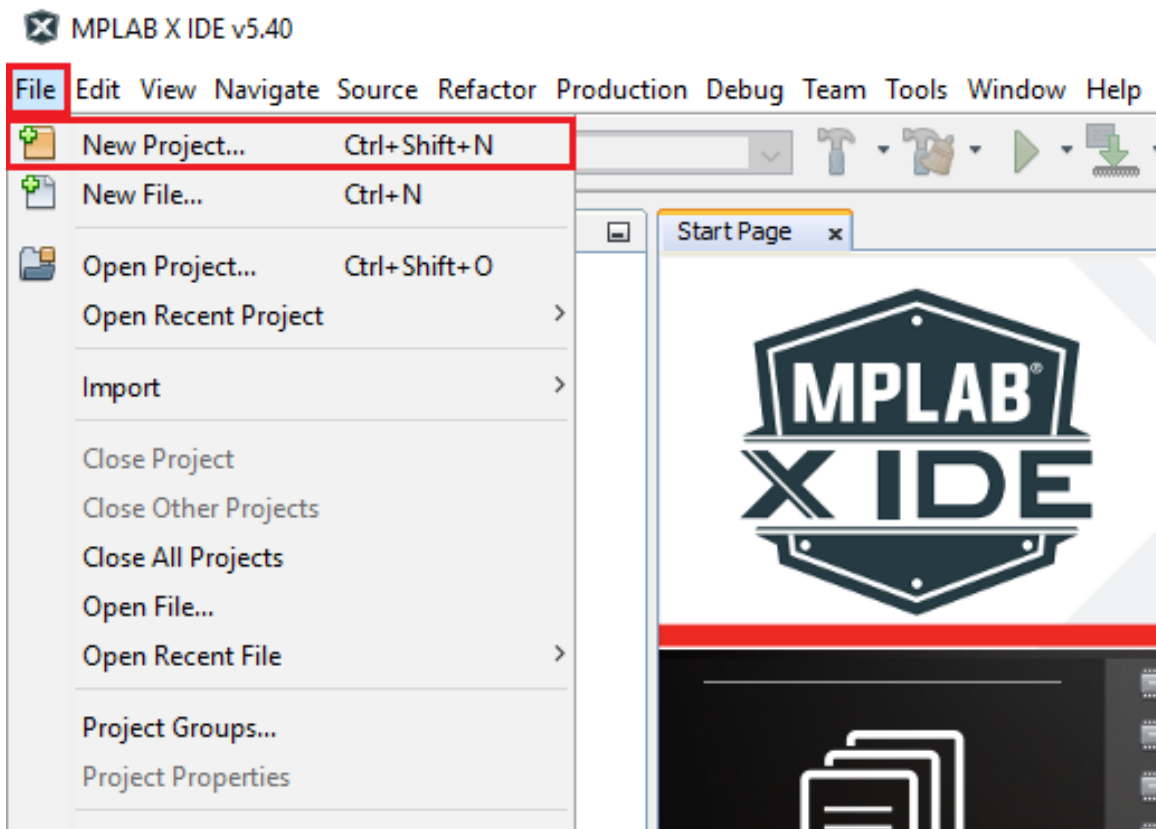
安装MHC

1. 在MPLAB X IDE中安装MPLAB Harmony配置器（MHC）插件（如果尚未安装）。
2. 验证MPLAB Harmony 配置器（MHC）插件是否安装成功。
 - 从Windows® Start（开始）菜单启动MPLAB X IDE。关闭当前打开的所有项目和文件。
 - 转到**Tools > Embedded**（工具 > 已安装工具）。
 - 菜单中将显示**MPLAB Harmony 3 Configurator**（MPLAB Harmony 3配置器）。

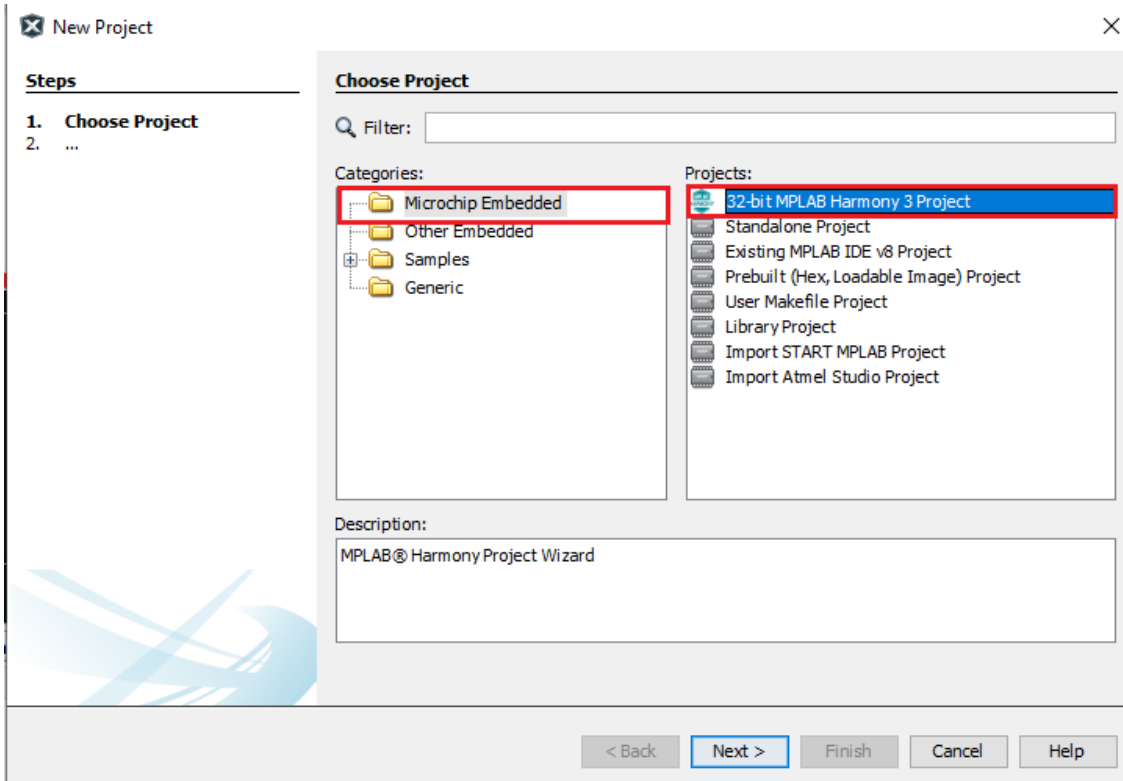


创建MPLAB Harmony v3项目

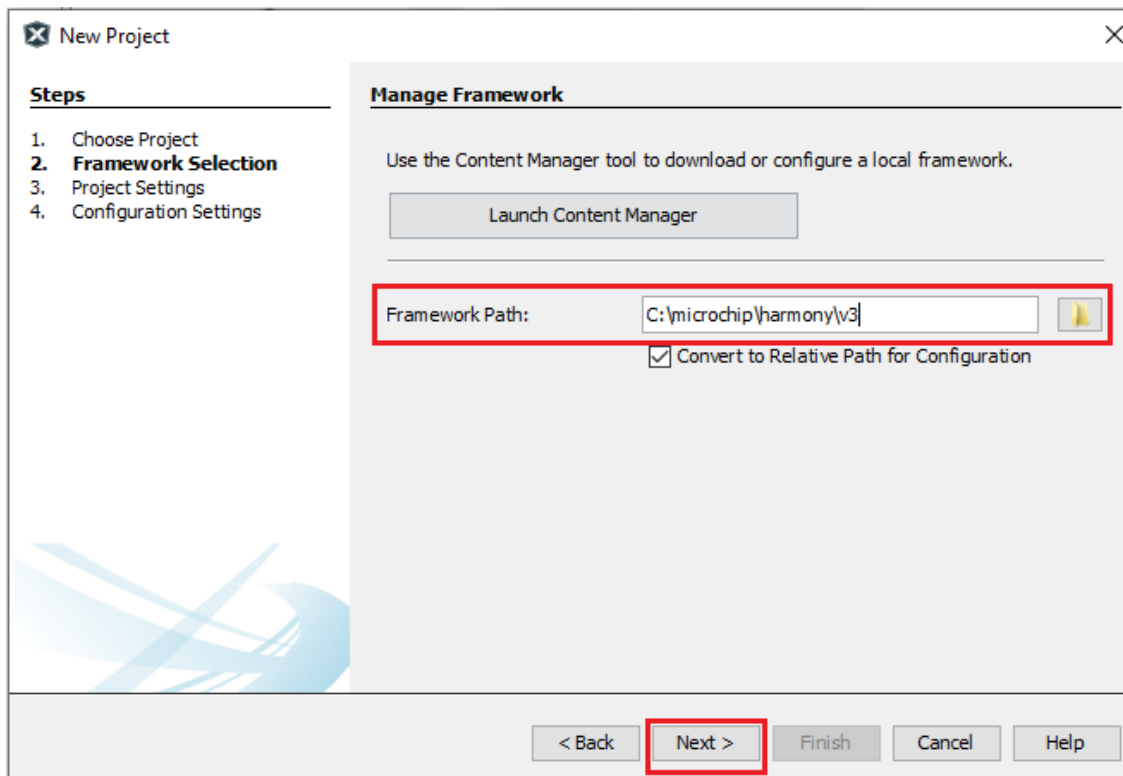
1. 使用MPLAB X IDE创建MPLAB Harmony v3项目
 - 从主IDE菜单中选择**File > New Project**（文件 > 新建项目）



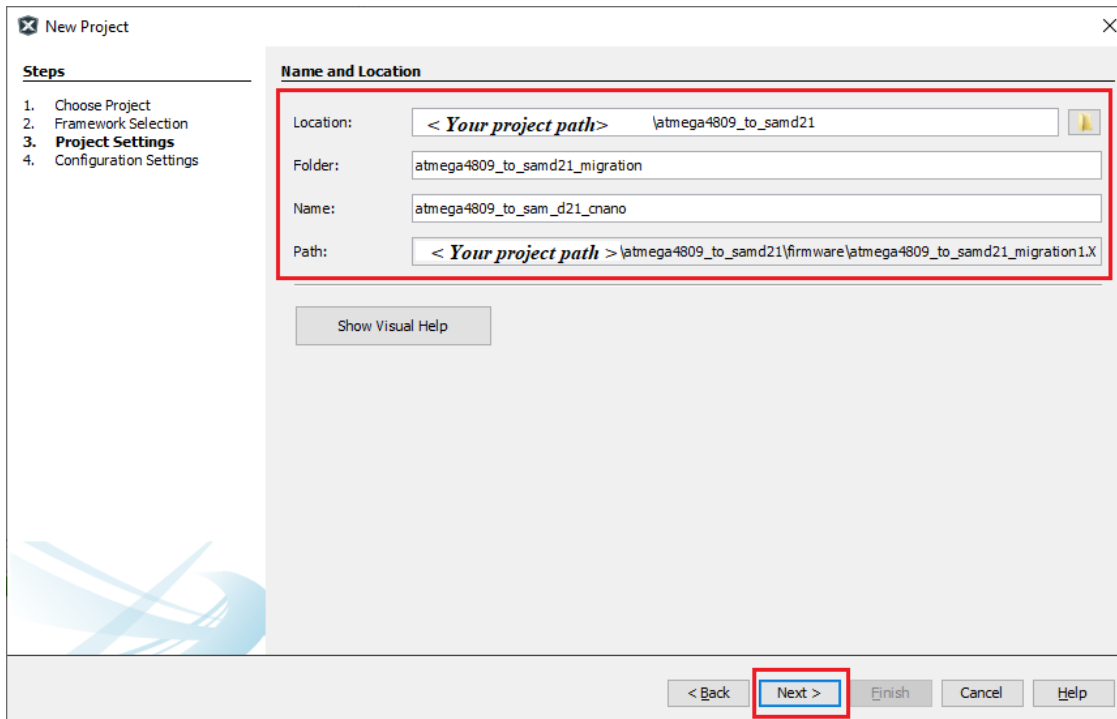
- 在**New Project**（新建项目）对话框窗口的**Categories**（类别）窗格中，选择**Microchip Embedded**（Microchip已安装工具）。在**Projects**（项目）窗格中，选择**32-bit MPLAB Harmony 3 Project**（32位MPLAB Harmony 3项目），然后单击**Next**（下一步）



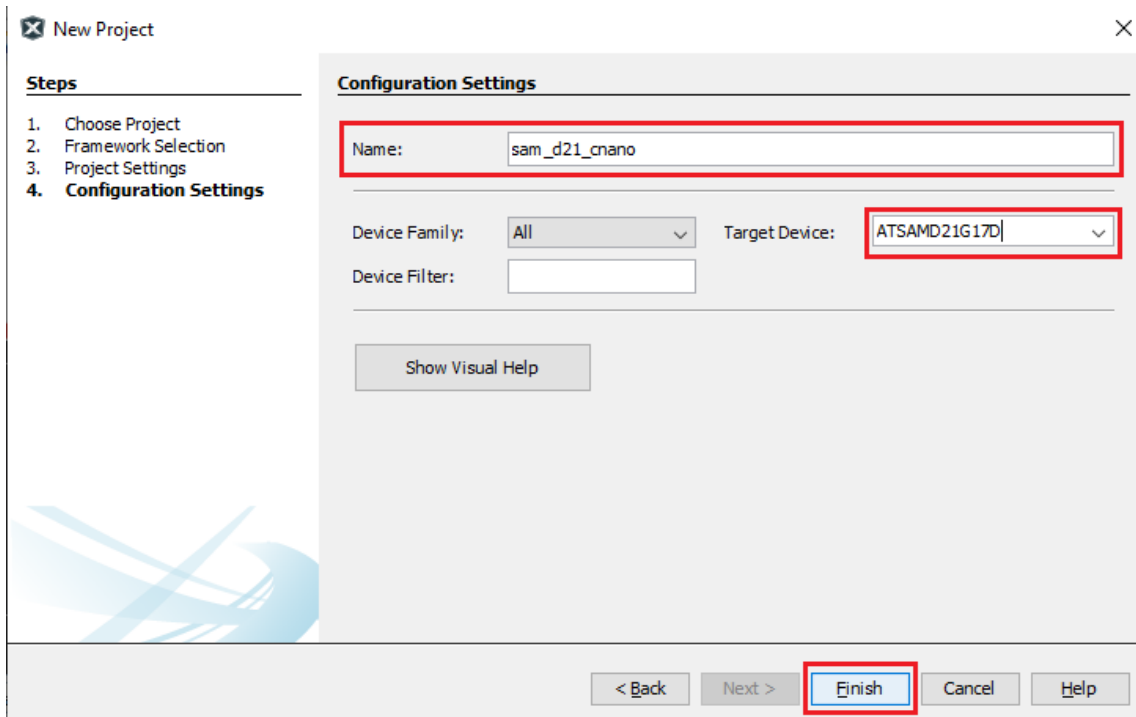
- 在**Framework Path**（框架路径）编辑框中，浏览到下载的框架所在的文件夹，然后选择**Next**



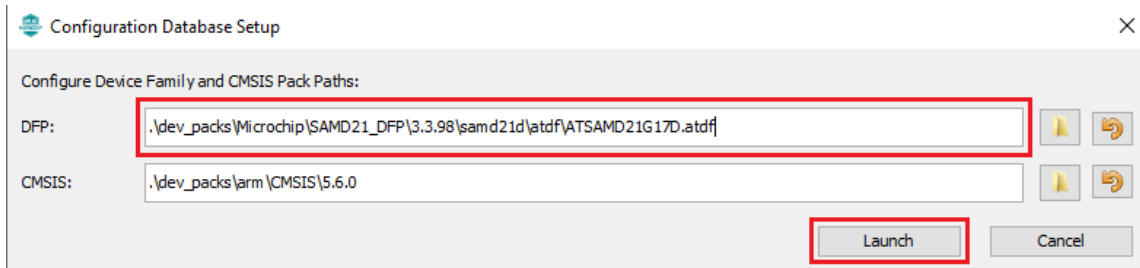
- 在**Project Settings**（项目设置）窗口中，应用以下设置：
 - Location**（位置）：指示新项目根文件夹的路径。所有项目文件均将置于该文件夹中。项目位置可以是任何有效路径，例如：`<您的项目路径>/atmega4809_to_samd21`
 - Folder**（文件夹）：指示MPLAB X .X文件夹的名称。输入“`atmega4809_to_samd21_migration`”，创建一个`atmega4809_to_samd21_migration.X`文件夹。
 - Name**（名称）：输入项目的逻辑名称“`atmega4809_to_samd21_cnano`”。这是将在MPLAB X IDE中显示的名称。
- 单击**Next**，继续进行配置设置



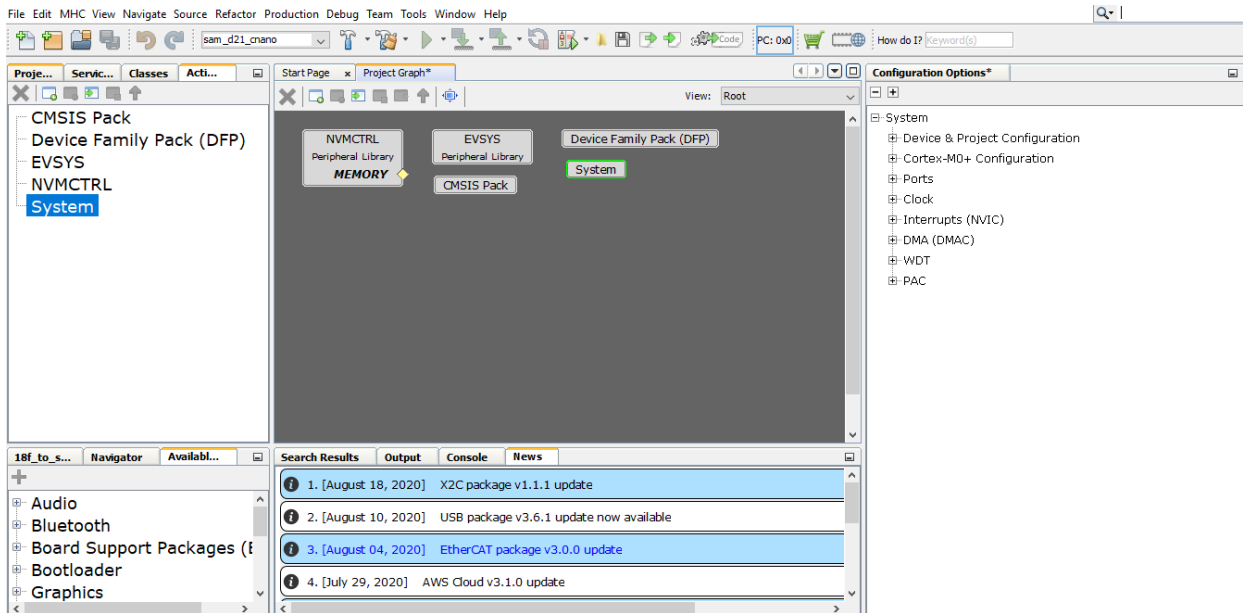
- 按照以下步骤设置项目的配置设置：
 - Name:** 输入配置名称“*sam_d21_cnano*”。
 - Target Device**（目标器件）：选择**ATSAMD21G17D**作为目标器件。
 - 选择目标器件后，单击**Finish**（完成）以启动MHC。



- 可以在以下窗口中修改器件系列包路径。当前使用默认值



- 该项目的MHC插件主窗口将如下图所示

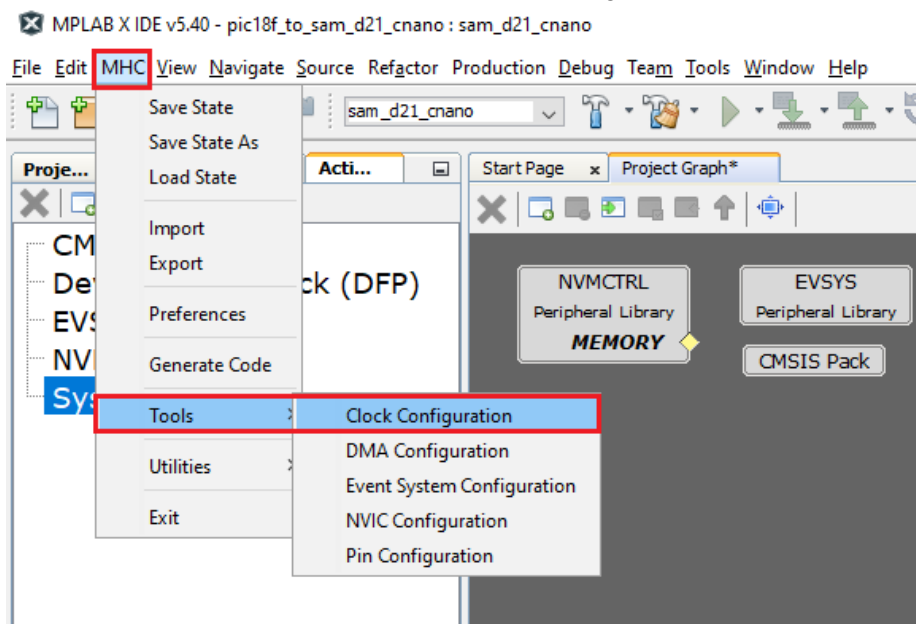


配置时钟

1. 配置时钟设置

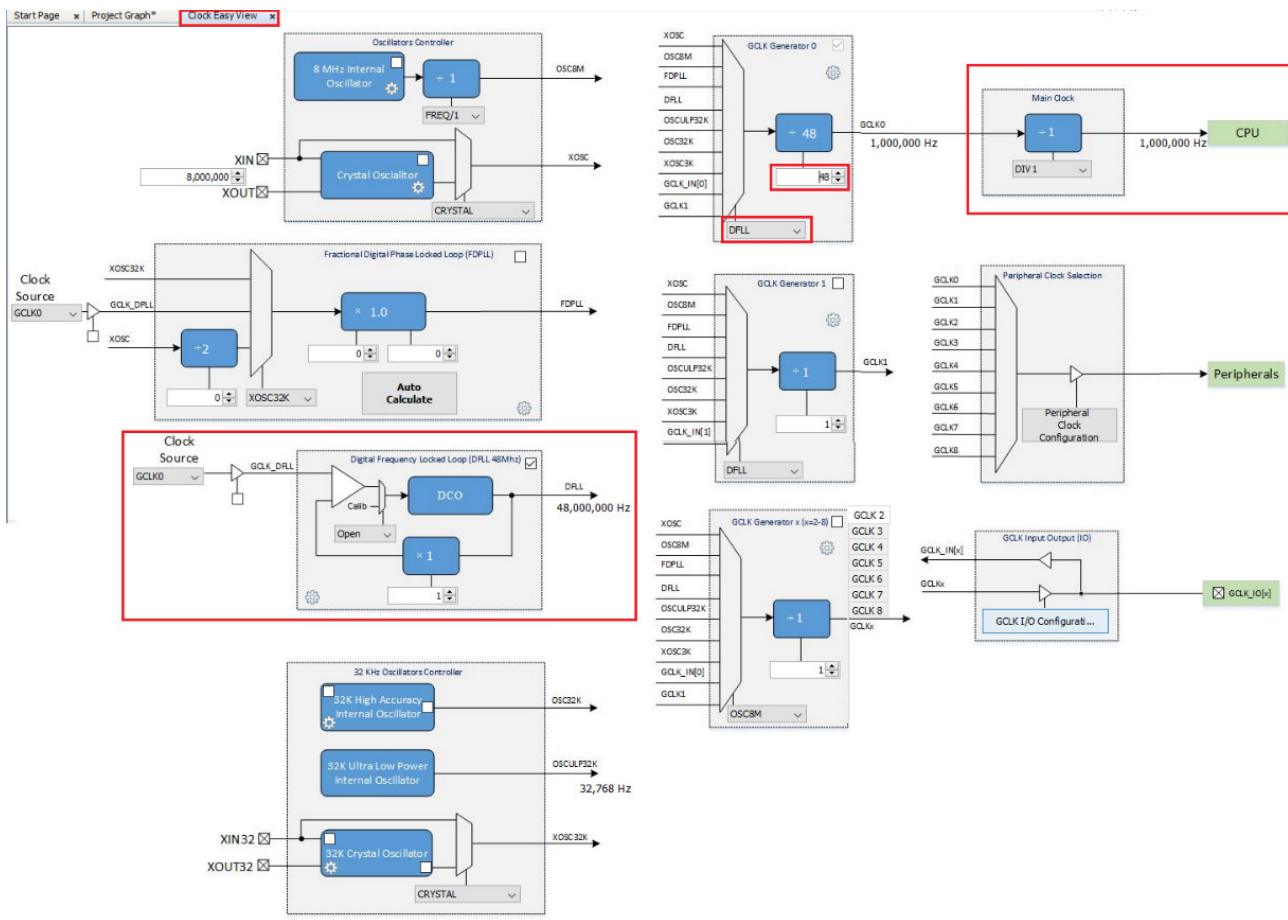
SAM D21系列单片机（MCU）包含一个复杂的时钟分配系统，旨在为用户应用程序提供最大的灵活性。时钟系统允许以动态方式调节器件的性能和功耗。MHC提供了易于使用的用户界面（User Interface, UI）窗口和Clock Easy View（时钟简化视图）窗口来配置系统和外设时钟。

- 转到MHC窗口的**Tools**（工具）菜单以启动**Clock Easy View**，然后选择**Clock Configuration**（时钟配置）



- 在**Clock Easy View**中，SAM D21最高可在48 MHz频率下工作，因此将**Main Clock**（主时钟）设置为48 MHz
- 将**Main Clock**更改为1 MHz

- 配置并使能开环数字锁频环（DFLL 48 MHz）以运行主时钟。配置的DFLL作为输入馈入GCLK发生器“0”，必须选择合适的时钟分频器（48）和屏蔽器，以实现1 MHz的最大频率。请参见下图将主时钟配置为在1 MHz频率下运行



MHC时钟简化视图以UI形式提供了SAMD21中可用的所有时钟选项，这样更加方便在需要时配置时钟。

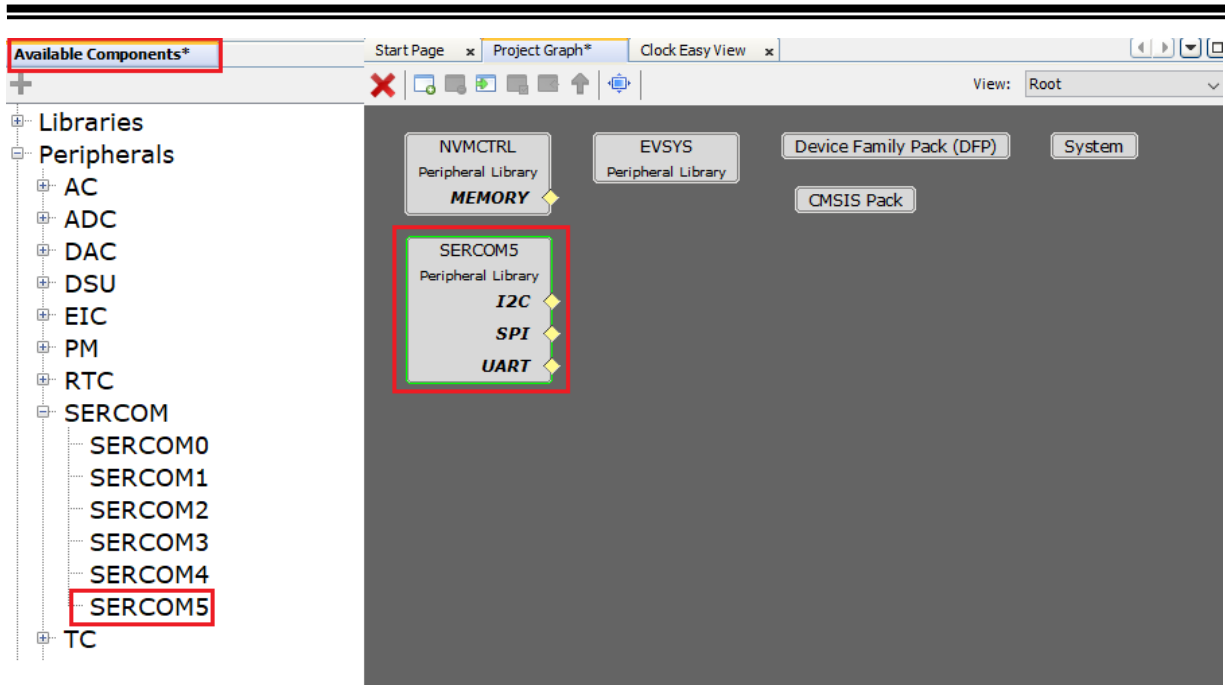
添加SERCOM（USART）外设

1. 添加SERCOM（USART）外设

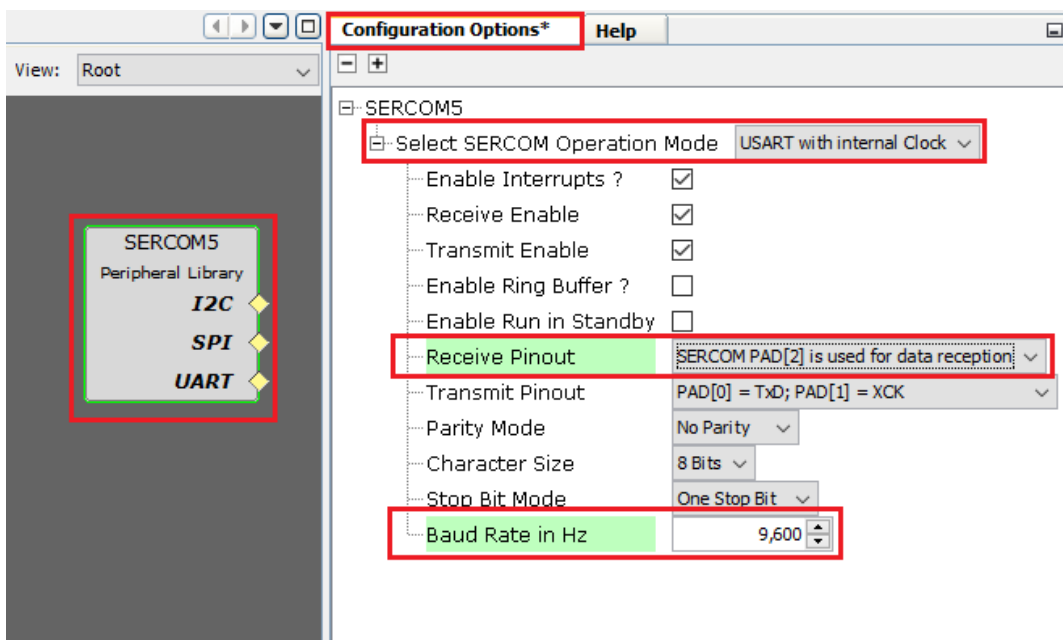
MHC图形用户界面有助于选择和配置项目所需的外设。其中的**Active Component**（活动组件）选项卡会列出SAMD21中可用的所有外设。

- 单击MPLAB Harmony配置器（MHC）窗口左侧窗格中的**Available Components**（可用组件）选项卡。展开**Peripherals > SERCOM**（外设 > SERCOM）
- 双击或拖放**SERCOM5**，将USART外设库（PLIB）添加到项目图中

注：SAM D21 Curiosity Nano板上的Nano调试器还包括一个虚拟COM端口，该端口连接到SERCOM5，因此在本应用程序中选择了SERCOM5。



- 在图中选择**SERCOM5 Peripheral Library**（SERCOM5外设库）。验证默认的“SERCOM工作模式”配置是否设置为USART，然后将波特率配置为9600并将帧格式配置为标准的8-N-1（8个数据位，无奇偶校验位，1个停止位）。



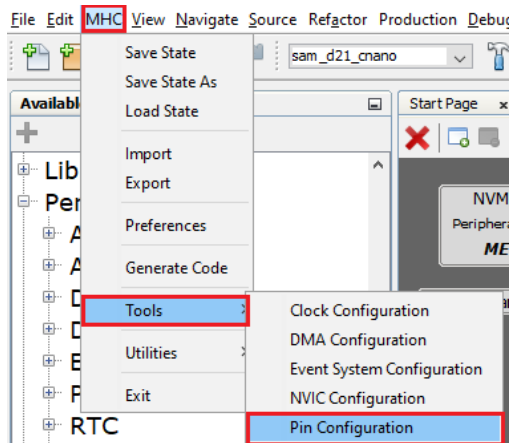
2. 使用MHC引脚配置器配置USART引脚

引脚管理器包含“Pin Settings”（引脚设置）、“Pin Diagram”（引脚图）和“Pin Table”（引脚表）选项卡，用户可以在其中进行配置（分配外设功能、设置引脚方向和配置上拉/下拉等）以及映射I/O引脚。

图形视图或表视图中引脚的相关配色如下：

- 灰色：** 该引脚不能用于选定配置，并且没有任何已使能模块在该引脚上具有功能。白色背景上的灰色锁表示引脚已被选定的系统功能锁定。
- 蓝色：** 该引脚可用，可以分配给模块。
- 绿色（带锁）：** 该引脚已被选定并分配给模块。引脚旁边显示的名称是模块上下文中的引脚名称或输入的自定义名称。

- 单击**MHC > Tools > Pin Configuration**（MHC > 工具 > 引脚配置），打开**Pin Configuration**（引脚配置）选项卡。



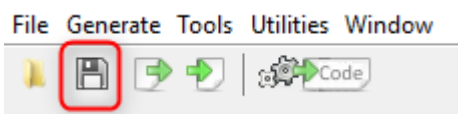
- 启动Pin Configuration时将打开Pin Settings、Pin Diagram和Pin Table选项卡。
- 选择MHC **Pin Table**选项卡，然后向下滚动至**SERCOM5**模块，如下所示。
 - 在PA22（引脚#31）上使能USART-TX
 - 在PB22（引脚#37）上使能USART-TX

| Package: TQFP48 | | PA12 | PA13 | PA14 | PA15 | PA16 | PA17 | PA18 | PA19 | PA20 | PA21 | SERCOM.. | PA23 | PA24 | PA25 | GNDIO | DDIO | SERCOM.. | PB23 | PA27 | RESET_... | PA28 | GNDIO |
|-----------------|--------------|------|------|------|------|------|------|------|------|------|------|----------|------|------|------|-------|------|----------|------|------|-----------|------|-------|
| Module | Function | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 |
| | SERCOM4_PAD3 | | | | | | | | | | | | | | | | | | | | | | |
| SERCOM5 | SERCOM5_PAD0 | | | | | | | | | | | | | | | | | | | | | | |
| | SERCOM5_PAD1 | | | | | | | | | | | | | | | | | | | | | | |
| | SERCOM5_PAD2 | | | | | | | | | | | | | | | | | | | | | | |
| | SERCOM5_PAD3 | | | | | | | | | | | | | | | | | | | | | | |

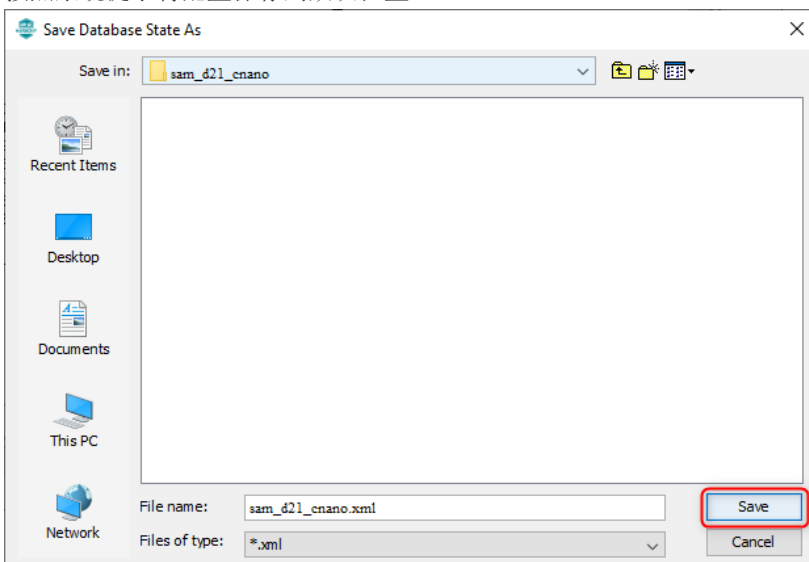
生成代码

1. 生成代码

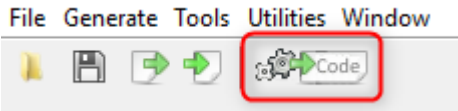
- 完成后，在生成代码之前，单击MPLAB Harmony配置器（MHC）中的**Save**（保存）图标，如下所示。



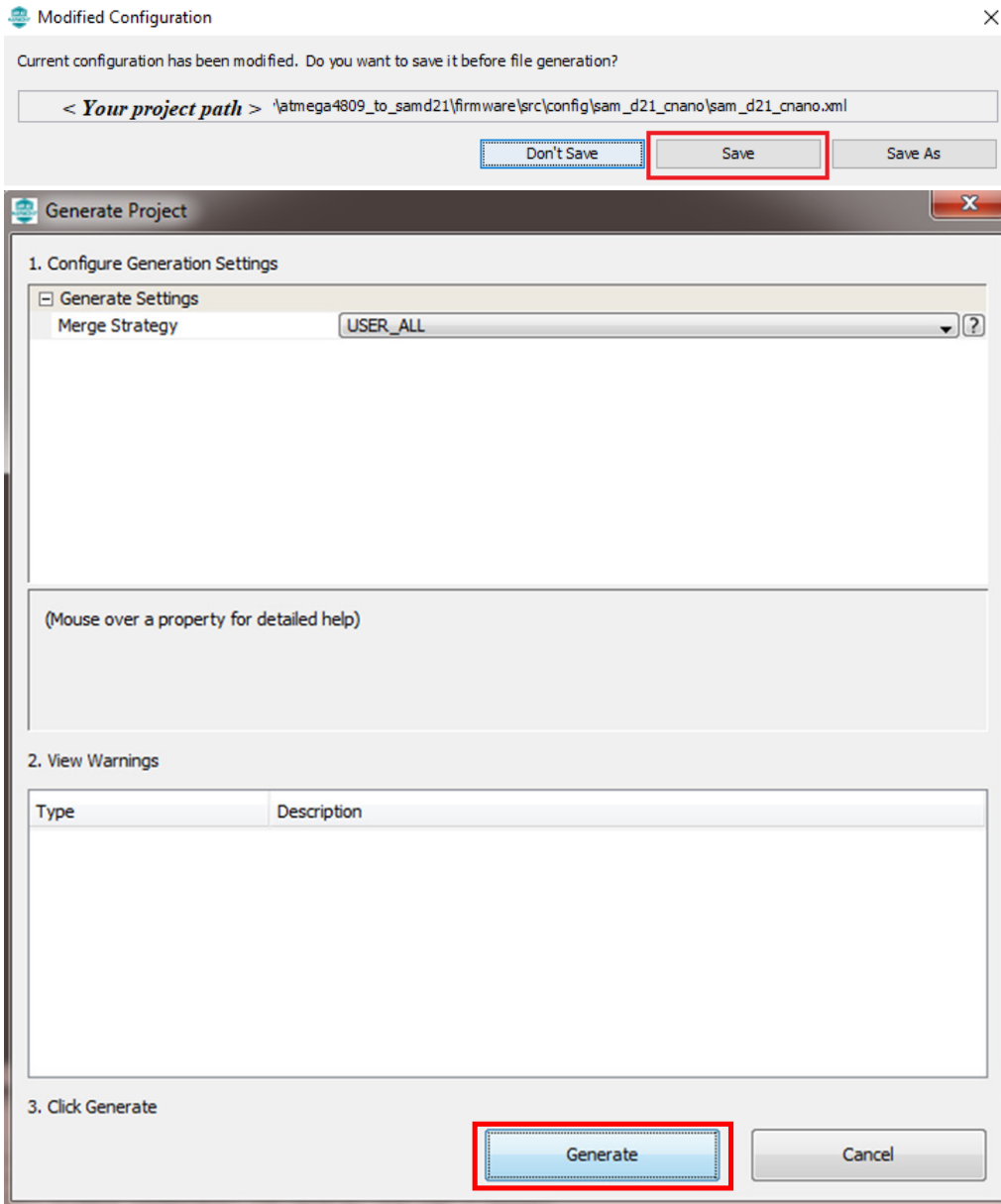
- 按照系统提示将配置保存到默认位置。



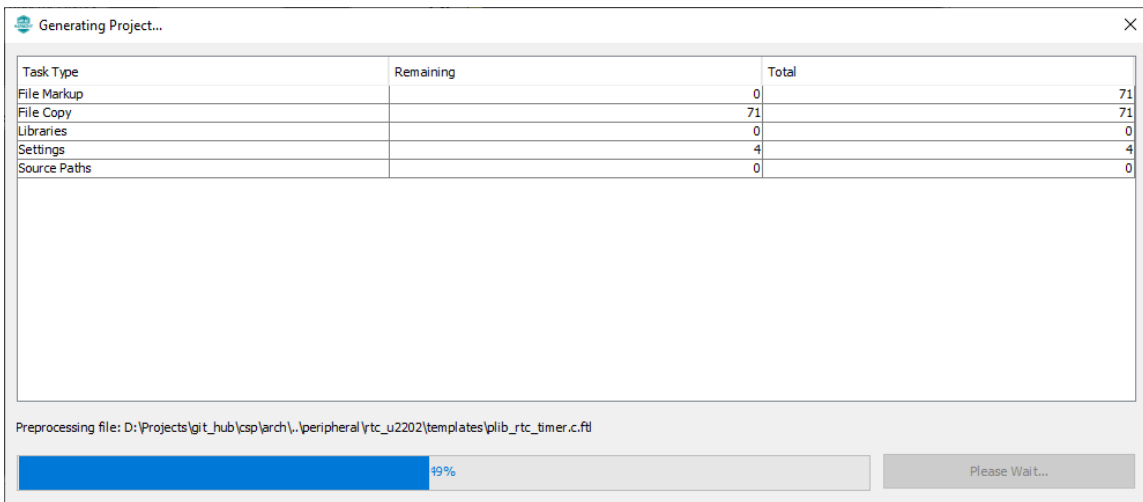
- 单击如下所示的**Code Generate**（代码生成）按钮，开始生成代码。



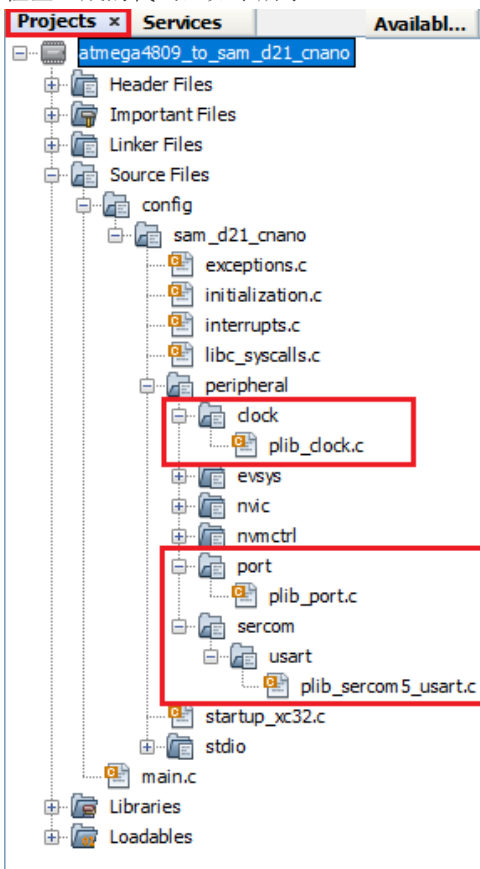
- 单击**Generate Project**（生成项目）窗口中的**Generate**（生成）按钮，保持如下所示的默认设置。如果系统提示保存配置，单击**Save**。



- 在生成代码的过程中，MHC将显示进度，如下所示。



- 导航到MPLAB X IDE中的**Projects**选项卡，查看项目树结构。
- 检查生成的代码，如下所示。



MHC将包含所有MPLAB Harmony库文件，并根据MHC选择生成代码。生成的代码会将文件和文件夹添加到Harmony项目中，如图中所示。在生成的代码中，请注意为SERCOM 5（作为通用同步/异步收发器（USART））、时钟和PORT外设生成的外设库（PLIB）文件。

添加代码

1. 导航到MPLAB X IDE中的**Projects**选项卡并选择main.c，然后在main函数中添加以下代码片段，以在串行终端上打印“Hello World”字符串。

```
uint8_t welcomeData[] = "Hello World\r\n";
```

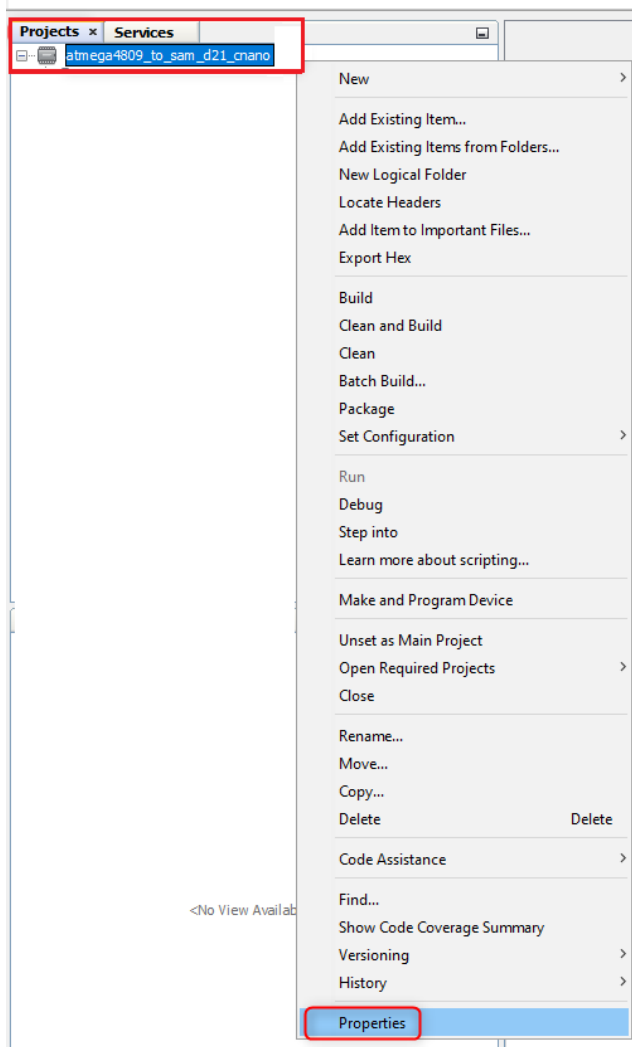
```
while ( true )
{
```

```
SERCOM5_USART_Write(&welcomeData[0],sizeof(welcomeData));  
}
```

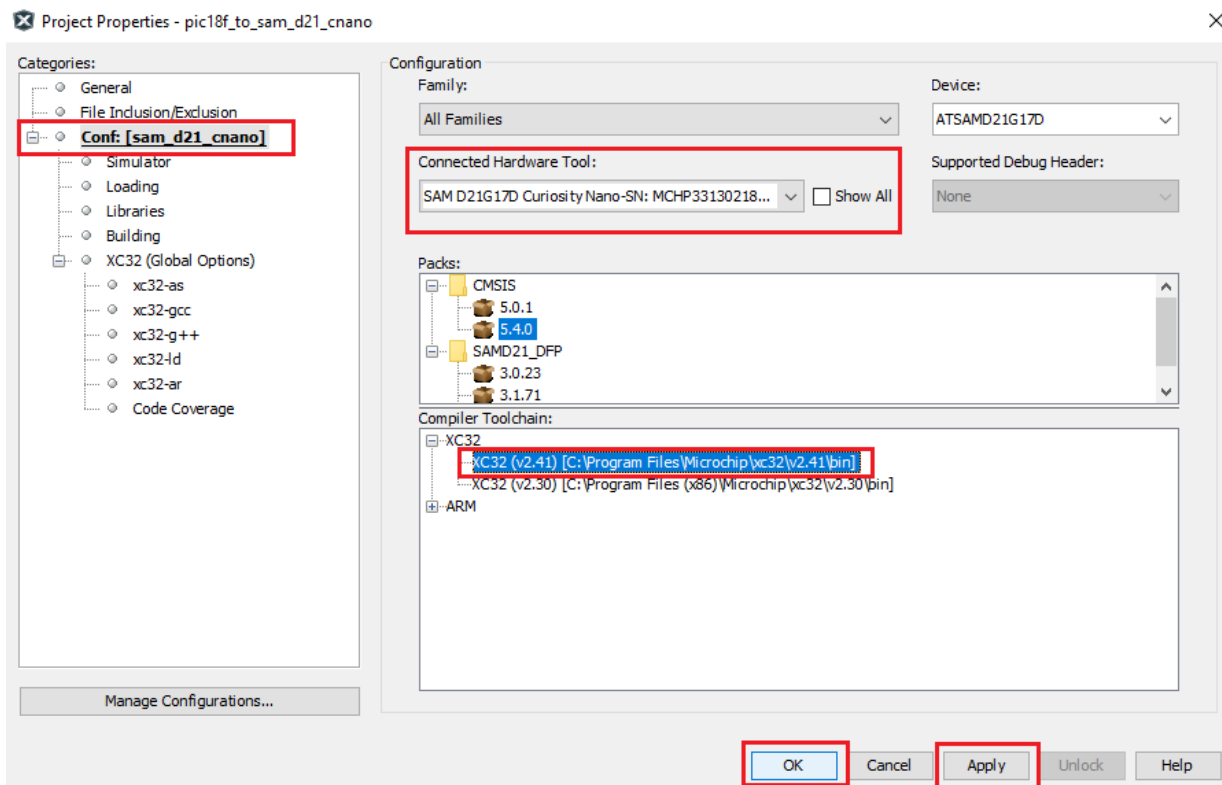
```
int main ( void )  
{  
    /* Initialize all modules */  
    SYS_Initialize ( NULL );  
  
    uint8_t welcomeData[] = "Hello World\r\n";  
  
    while ( true )  
    {  
        SERCOM5_USART_Write(&welcomeData[0],sizeof(welcomeData));  
    }  
  
    /* Execution should not come here during normal operation */  
  
    return ( EXIT_FAILURE );  
}
```

编译和编程

1. 在继续进行编译和编程之前，请设置编译器工具链。单击MPLAB X IDE左上方窗格中的**Projects**选项卡。右键单击项目名称**pic18f_to_sam_d21_cnano**，然后转到**Properties**（属性）。



2. 确保已选择XC32 (v2.41) 作为XC32的编译器工具链。单击**Apply** (应用), 然后单击**OK** (确定)。



3. 单击**Clean and Build** (清除并编译) 按钮, 清除并编译您的应用程序, 如下所示。



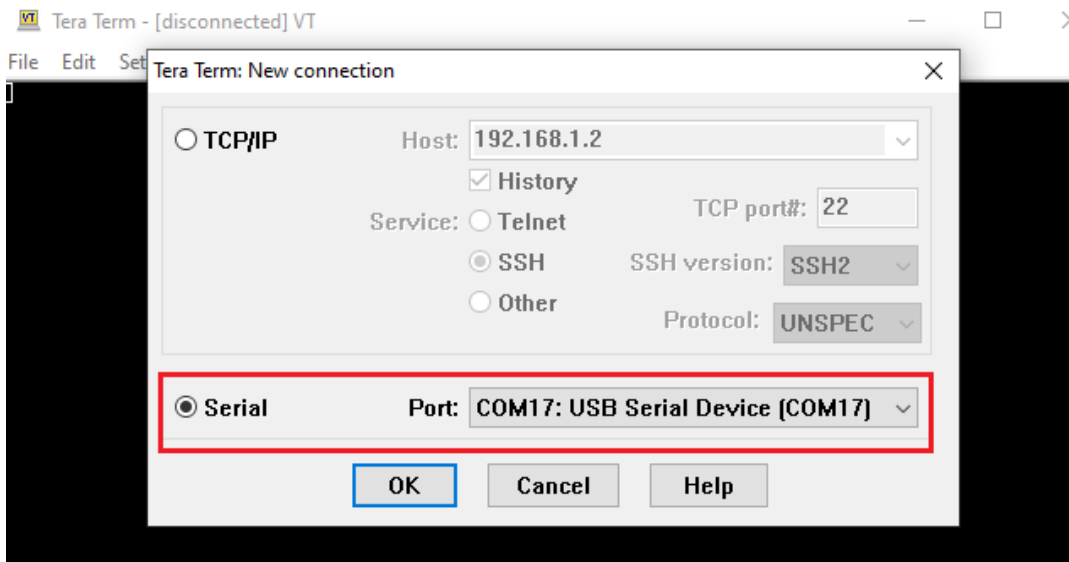
4. 单击**Make and Program** (编译并编程) 按钮, 将您的应用程序编程到器件, 如下所示。



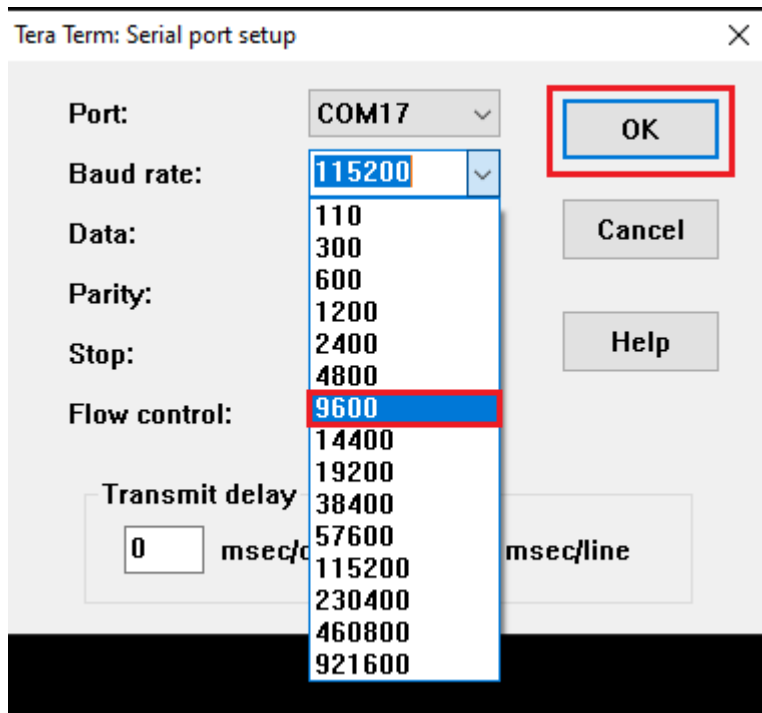
实验应成功编译并编程。

验证“Hello World”输出

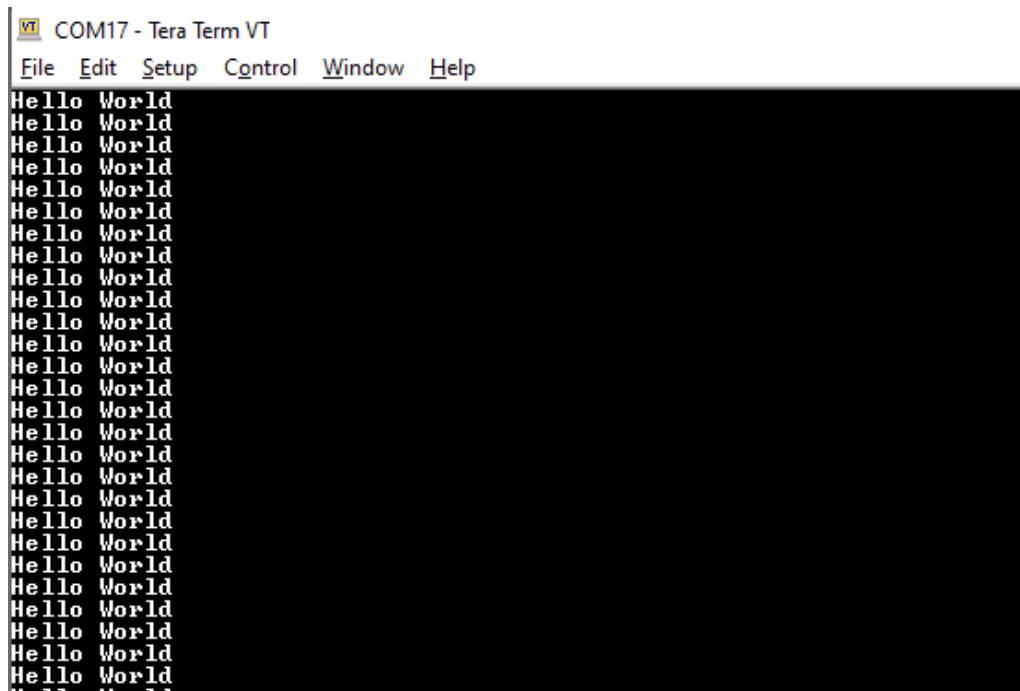
1. 现在, 打开PC上的**Tera Term**终端应用程序 (从Windows开始菜单中单击**Start**按钮)。选择**Serial Port** (串口), 如下所示。



2. 将波特率更改为9600。



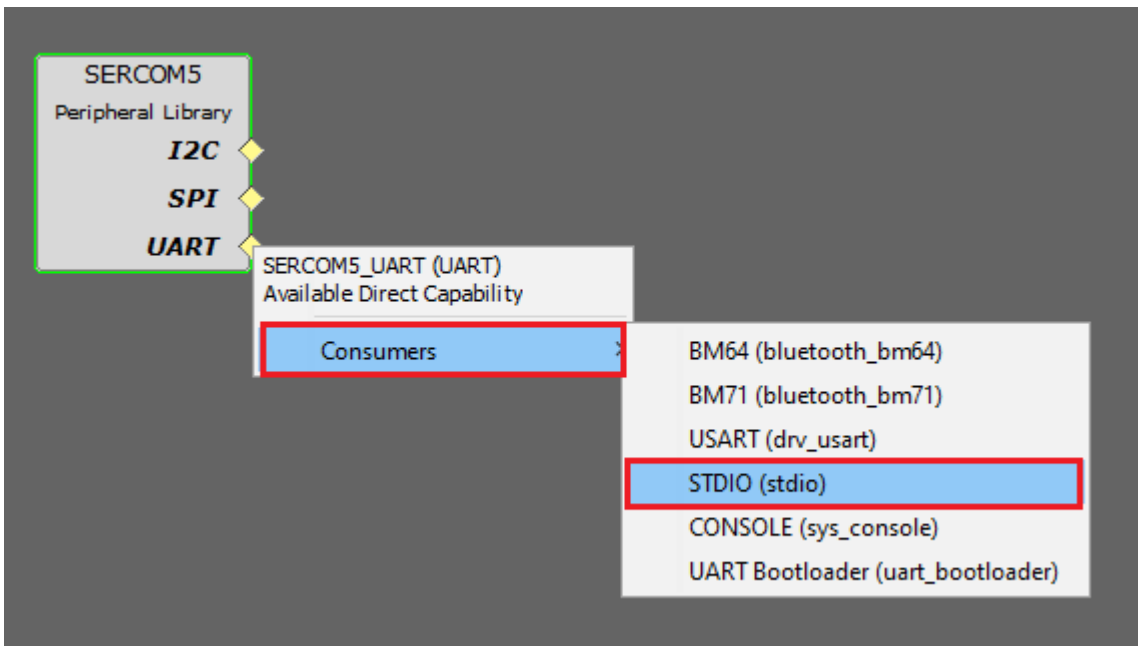
3. 终端上应显示“Hello World”消息，如下所示。



在串行终端上打印“Hello World”消息已成功完成。接下来继续扩展应用程序的功能。

配置/使能格式化STDIO、USART接收和开关/LED功能

- 配置STDIO库以使用**printf**函数向串行终端写入数据，使用**scanf**函数从串行终端读取数据
- 从串行终端接收命令并控制SAM D21 Curiosity评估工具包上的LED0
- 使用SAM D21 Curiosity评估工具包上的开关SW0，按下时可将LED状态切换两次。外部中断控制器（EIC）和Systick外设用于实现此目的
 - 1 如果不小心关闭了MHC，想要再次打开，则可以转到MPLAB X IDE中的**Tools > Embedded > MPLAB Harmony 3 Configurator**（工具 > 已安装工具 > MPLAB Harmony 3配置器）。
 - 2 在**Project Graph**（项目图）中，选择**SERCOM5**。右键单击黄色菱形，然后选择**STDIO**作为使用对象，如下所示：



- 3 选择MHC **Pin Settings**选项卡，然后向下滚动至**Pin ID**（引脚ID）列中的PORT引脚**PB10**（引脚编号19），并将**PB10**配置为LED功能的输出引脚，如下所示。

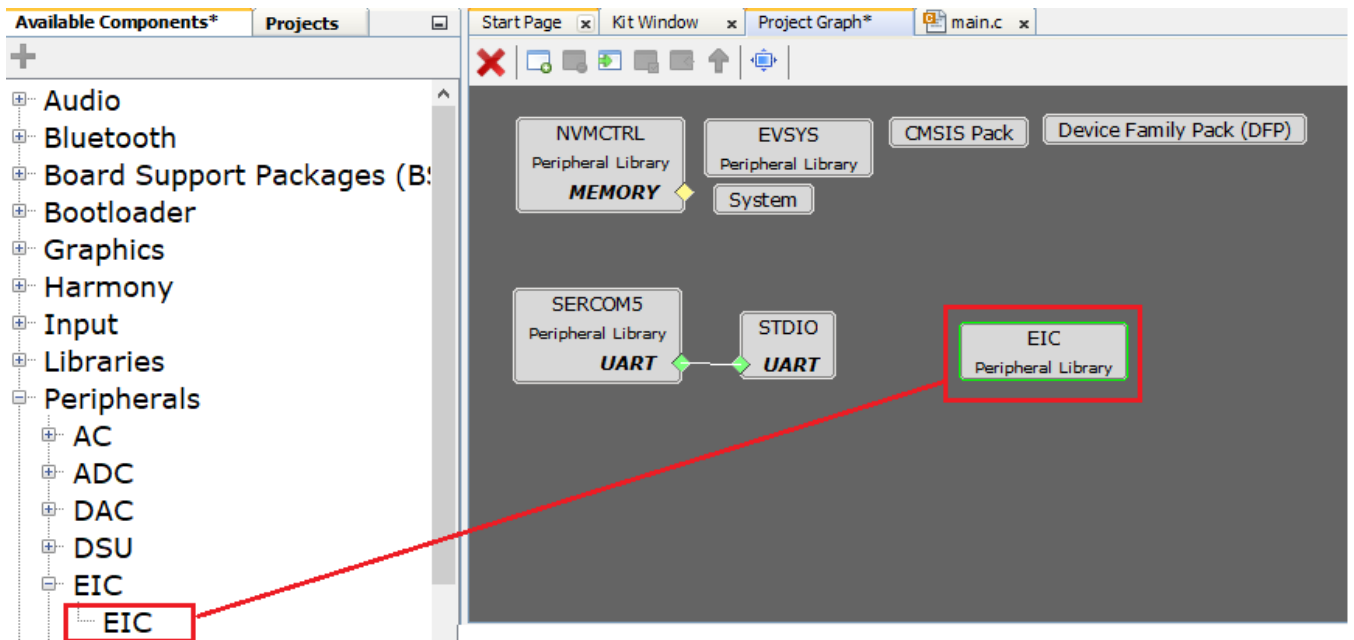
| Pin Number | Pin ID | Custom Name | Function | Mode | Direction | Latch | Pull Up | Pull Down | Drive Strength |
|------------|--------|--------------|--------------|---------|----------------|-------|--------------------------|--------------------------|----------------|
| 33 | PA24 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 34 | PA25 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 39 | PA27 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 41 | PA28 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 45 | PA30 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 46 | PA31 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 47 | PB02 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 48 | PB03 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 7 | PB08 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 8 | PB09 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 19 | PB10 | LED | GPIO | Digital | Out | High | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 20 | PB11 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 37 | PB22 | SERCOM5_PAD2 | SERCOM5_P... | Digital | High Impedance | n/a | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 38 | PB23 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 5 | GNDANA | | | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 6 | VDDANA | | | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 17 | VDDIO | | | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 18 | GNDIO | | | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |

- 4 选择**Pin Settings**选项卡，然后向下滚动至**Pin ID**（引脚ID）列中的PORT引脚**PB11**（引脚编号20），并将**PB11**配置为开关功能的外部中断引脚，如下所示。使能内部上拉以避免发生边沿误检，因为SAM D21 Curiosity Nano评估工具包上没有外部上拉。

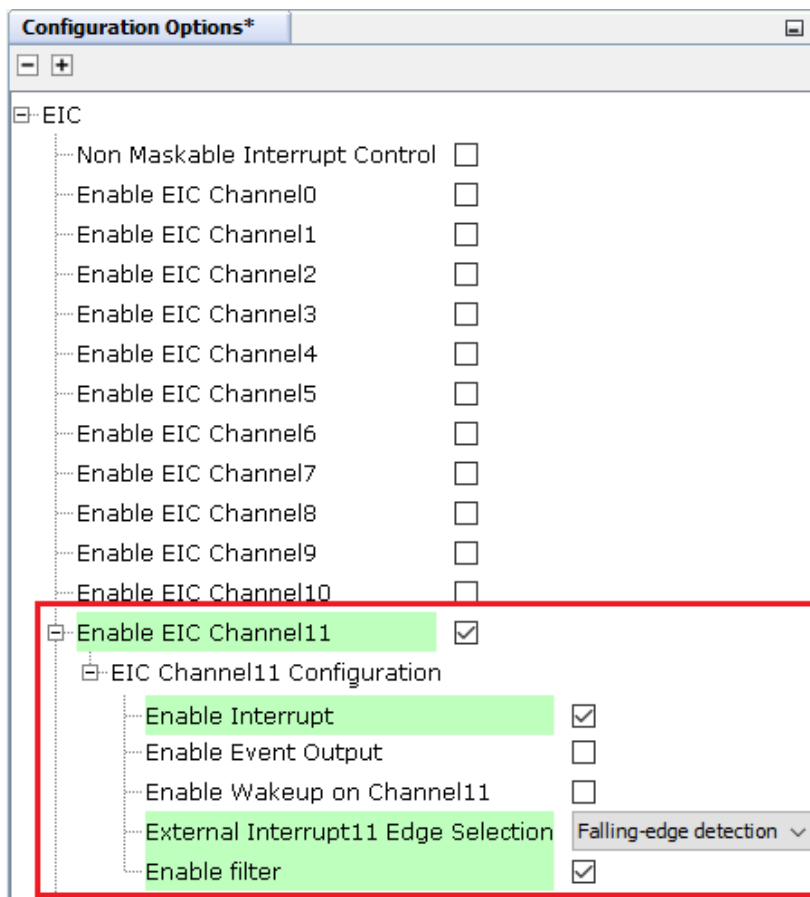
| Pin Number | Pin ID | Custom Name | Function | Mode | Direction | Latch | Pull Up | Pull Down | Drive Strength |
|------------|--------|--------------|--------------|---------|----------------|-------|-------------------------------------|--------------------------|----------------|
| 32 | PA23 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 33 | PA24 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 34 | PA25 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 39 | PA27 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 41 | PA28 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 45 | PA30 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 46 | PA31 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 47 | PB02 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 48 | PB03 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 7 | PB08 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 8 | PB09 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 19 | PB10 | LED | GPIO | Digital | Out | High | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 20 | PB11 | EIC_EXTINT11 | EIC_EXTINT11 | Digital | High Impedance | n/a | <input checked="" type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 37 | PB22 | SERCOM5_PAD2 | SERCOM5_P... | Digital | High Impedance | n/a | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 38 | PB23 | | Available | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |
| 5 | GNDANA | | | Digital | High Impedance | Low | <input type="checkbox"/> | <input type="checkbox"/> | NORMAL |

5 选择Project Graph选项卡。

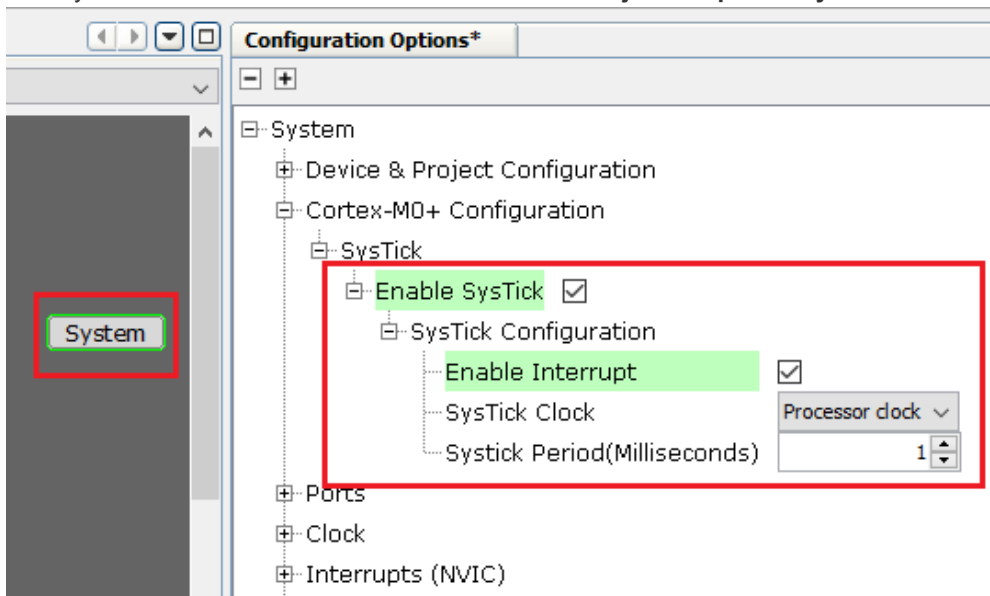
- 在Available Components选项卡下，展开Peripheral > EIC（外设 > EIC）。选择并双击EIC，将EIC模块添加到项目中。



- 配置EIC模块以在每次用户按下开关SW0时产生中断（如下所示），并使能滤波器功能以避免开关引脚上出现电噪声。



- 1 使能Systick来创建用于切换LED0状态的延时，单击**Project Graph**中的**System**（系统）并按如下所示进行配置。



- 2 完成后，在生成代码之前，单击MPLAB Harmony配置器（MHC）中的**Save**图标。按照系统提示将配置保存到默认位置。
 - 3 单击**Code Generate**按钮，开始生成代码。
 - 4 单击**Generate Project**窗口中的**Generate**按钮，保持默认设置。如果系统提示保存配置，请单击**Save**。
 - 5 导航到MPLAB X IDE中的**Projects**选项卡并选择main.c，然后添加以下代码片段以接收来自串行终端的命令以及用于控制SAM D21 Curiosity Nano评估工具包上的LED0的函数。
- 在main.c中“main”函数上方的#include中添加以下函数

```
void executeCommand(char *command)
{
    if(strcmp(command, "ON") == 0)
```

```

        {
            LED_Clear();
            printf("OK, LED ON.\r\n");
        }
        else if (strcmp(command, "OFF") == 0)
        {
            LED_Set();
            printf("OK, LED OFF.\r\n");
        }
        else
        {
            printf("Incorrect command.\r\n");
        }
    }
}

```

```

#include <string.h>

void executeCommand(char *command)
{
    if(strcmp(command, "ON") == 0)
    {
        LED_Clear();
        printf("OK, LED ON.\r\n");
    }
    else if (strcmp(command, "OFF") == 0)
    {
        LED_Set();
        printf("OK, LED OFF.\r\n");
    }
    else
    {
        printf("Incorrect command.\r\n");
    }
}
}

```

- 将“SERCOM5_USART_Write”函数移至while循环上方，并将scanf添加到while循环内部，如下所示。

```

int main ( void )
{
    /* Initialize all modules */
    SYS_Initialize ( NULL );
    uint8_t command[8] = {0};
    uint8_t welcomeData[] = "Hello World\r\n";

    SERCOM5_USART_Write(&welcomeData[0], sizeof(welcomeData));

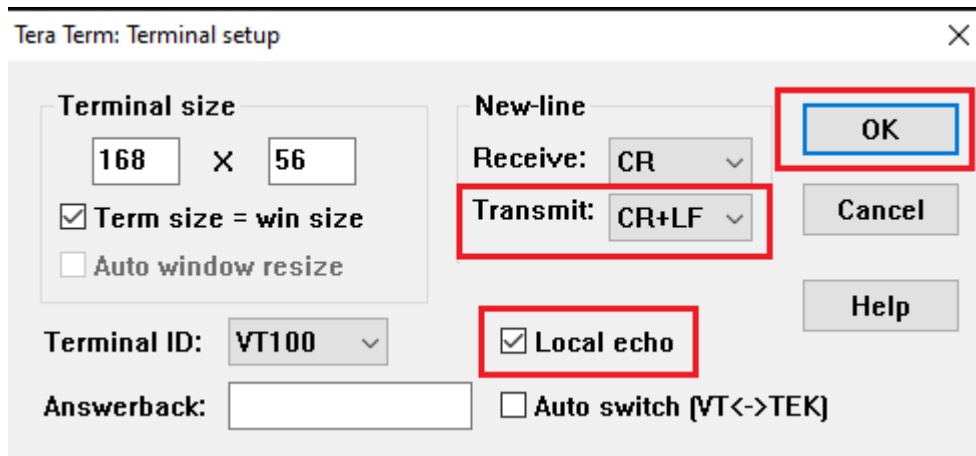
    while ( true )
    {
        scanf("\r\n%s", command);
        executeCommand((char*) command);
    }

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

    return ( EXIT_FAILURE );
}

```

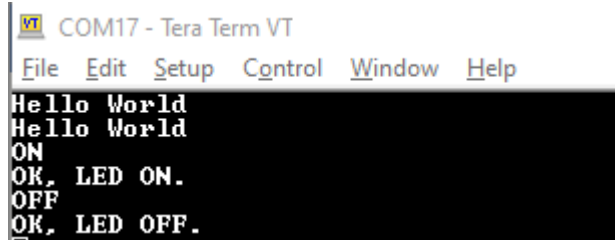
- 1 现在，打开PC上的Tera Term终端应用程序。使能Setup > Terminal（设置 > 终端）中的Local echo（本地回显），如下所示。



2. 单击**Clean and Build**按钮，清除并编译您的应用程序。

验证格式化STDIO、USART接收和开关/LED功能

1. 单击**Make and Program**按钮，将您的应用程序编程到器件。实验应成功编译并编程。
2. Tera Term中应显示“**Hello World**”消息。
3. 在串行终端中输入**ON**并按下**Enter**以点亮SAM D21 Curiosity Nano工具包上的LED0，然后在串行终端上输入**OFF**并按下**Enter**以熄灭SAM D21 Curiosity Nano评估工具包上的LED0。



4. 使用EIC外设将开关SW0配置为中断，因此在main函数中注册回调。每当用户按下开关时，均会通知回调并在回调内部将LED0状态切换两次。

- 在main.c文件的main函数内部添加以下代码：
EIC_CallbackRegister(EIC_PIN_11,EIC_User_Handler, 0); SYSTICK_TimerStart();
- 在main.c中main函数上方的static void EIC_User_Handler(uintptr_t context) { uint8_t i = 0;中添加以下代码

```
while(i<4){
    LED_Toggle();
    SYSTICK_DelayMs(500);
    i++;
}
}
```

```

static void EIC_User_Handler(uintptr_t context)
{
    uint8_t i = 0;

    while(i<4){
        LED_Toggle();
        SYSTICK_DelayMs(500);
        i++;
    }
}

int main ( void )
{
    /* Initialize all modules */
    SYS_Initialize ( NULL );
    uint8_t command[8] = {0};
    uint8_t welcomeData[] = "Hello World\r\n";

    EIC_CallbackRegister(EIC_PIN_11,EIC_User_Handler, 0);
    SYSTICK_TimerStart();

    SERCOM5_USART_Write(&welcomeData[0],sizeof(welcomeData));
}

```

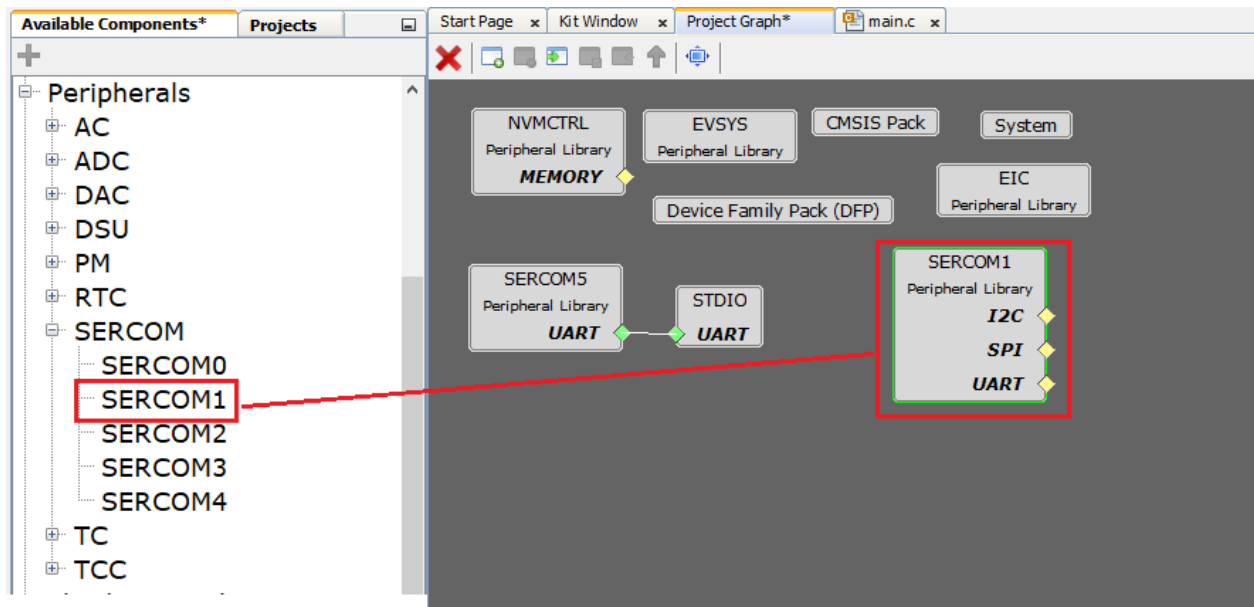
5. 单击**Clean and Build**按钮，清除并编译您的应用程序。
6. 单击**Make and Program**按钮，将您的应用程序编程到器件。
7. 按下SAM D21 Curiosity Nano评估工具包上的开关SW0，观察到LED0状态以500 ms的速率切换两次。

通过USART从串行终端接收命令以控制SAM D21 Curiosity Nano评估工具包上的LED0以及使用EIC和Systick外设。在按下开关SW0时切换LED0状态均已成功完成。

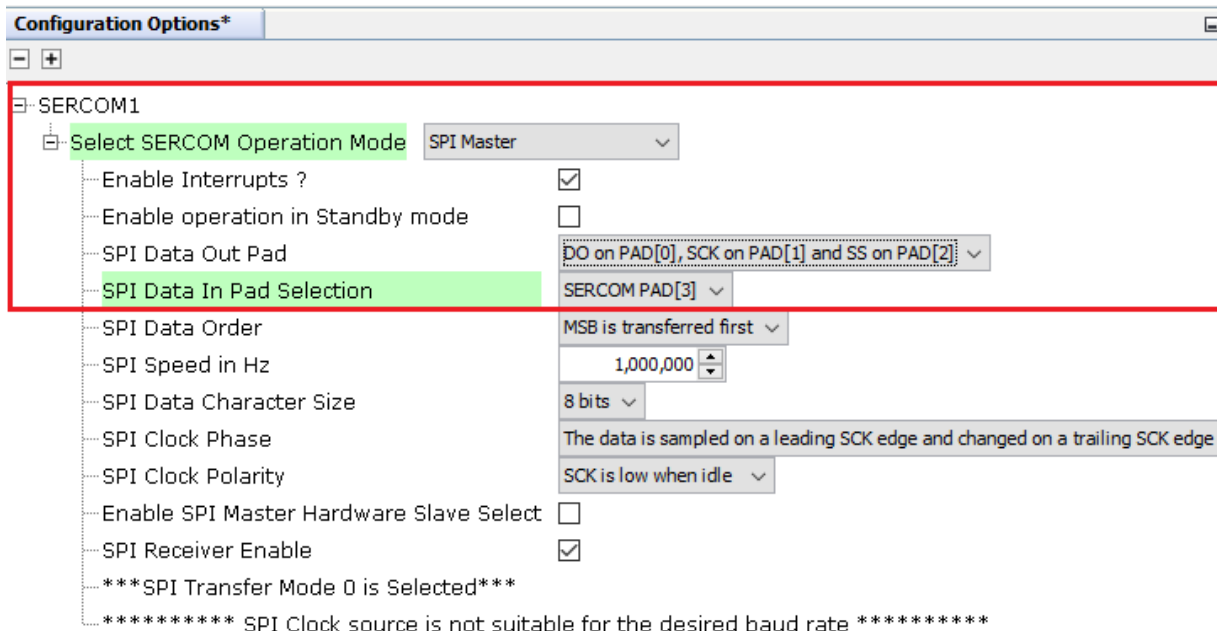
扩展应用程序以使用基于DMA的传输配置SPI环回

让我们扩展功能，添加串行外设接口（Serial Peripheral Interface, SPI）以执行自环回测试并添加直接存储器访问（DMA）。

1. 选择**Project Graph**选项卡。在**Available Components**选项卡下，展开**Peripheral > SERCOM**（外设 > SERCOM）。选择并双击**SERCOM1**，将SPI模块添加到项目中。



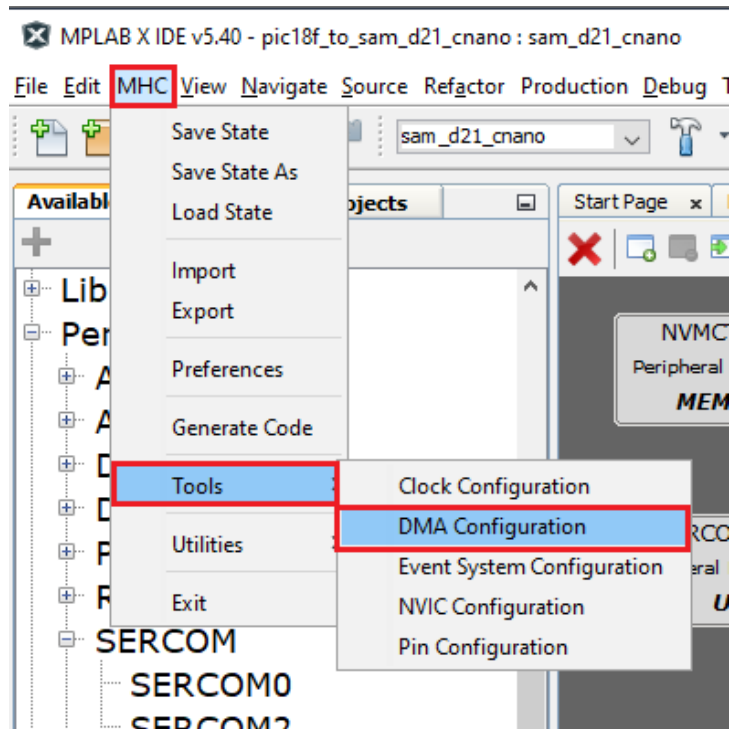
2. 在**Project Graph**中，选择**SERCOM1**并按如下所示进行配置



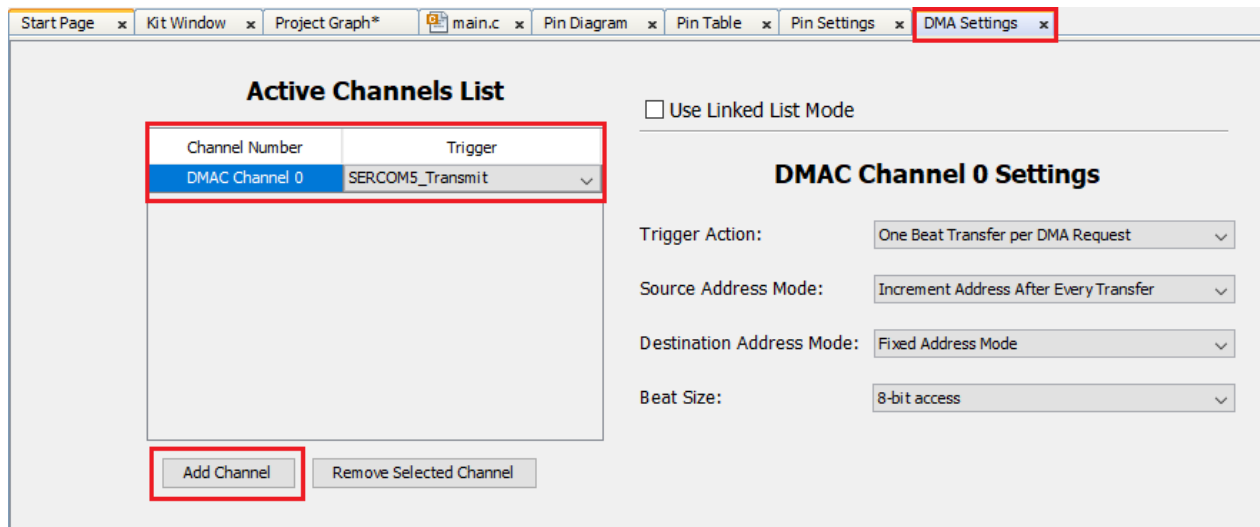
3. 打开Pin Table，选择SERCOM1并按如下所示配置引脚

| Module | Function | PA07 | PA08 | PA09 | PA10 | PA11 | VDDIO | GNDDIO | LED | EIC_EX... | PA12 | PA13 | PA14 | PA15 | SERCOM1 | SERCOM1 | SERCOM1 | SERCOM1 | PA20 | PA21 | SERCOM1 | PA23 | PA24 | PA25 | GNDDIO | VDDIO | SERCOM1 | PE23 | |
|---------|--------------|------|------|------|------|------|-------|--------|-----|-----------|------|------|------|------|---------|---------|---------|---------|------|------|---------|------|------|------|--------|-------|---------|------|--|
| | SERCOM0_PAD3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SERCOM1 | SERCOM1_PAD0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | SERCOM1_PAD1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | SERCOM1_PAD2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | SERCOM1_PAD3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

4. 从MHC > Tools > DMA Configuration (MHC > 工具 > DMA配置) 启动DMA Configurator (DMA配置器)



5. 单击ADD Channel (添加通道)，然后选择SERCOM5_Transmit来配置DMA



6. 完成后，在生成代码之前，单击MPLAB Harmony配置器（MHC）中的*Save图标。按照系统提示将配置保存到默认位置。
7. 单击**Code Generate**按钮，开始生成代码。
8. 单击**Generate Project**窗口中的**Generate**按钮，保持默认设置。
9. 如果系统提示保存配置，单击**Save**。
10. 在main.c的SERCOM1_SPI_WriteRead(&txData[0], sizeof(txData), &rxData[0], sizeof(rxData));中添加以下代码片段

```

/* Compare received data with the transmitted data */
if ((memcmp(txData, rxData, sizeof(txData)) == 0))
{
    /* Pass: Received data is same as transmitted data */
    DMAC_ChannelTransfer(DMAC_CHANNEL_0, "Received SPI Valid Data\r\n", \
        (const void *)&(SERCOM5_REGS->USART_INT.SERCOM_DATA), \
        25);
}
else
{
    /* Fail: Received data is not same as transmitted data */
    DMAC_ChannelTransfer(DMAC_CHANNEL_0, "Received Invalid SPI Data\r\n", \
        (const void *)&(SERCOM5_REGS->USART_INT.SERCOM_DATA), \
        27);
}

```

- o 在main函数上方添加以下代码片段：
`uint8_t txData[] = "SELF LOOPBACK DEMO FOR SPI!";`
`uint8_t rxData[sizeof(txData)];`

```

uint8_t txData[] = "SELF LOOPBACK DEMO FOR SPI!";
uint8_t rxData[sizeof(txData)];
int main ( void )
{
    /* Initialize all modules */
    SYS_Initialize ( NULL );
    uint8_t command[8] = {0};
    uint8_t welcomeData[] = "Hello World\r\n";

    EIC_CallbackRegister(EIC_PIN_11,EIC_User_Handler, 0);
    SYSTICK_TimerStart();

    SERCOM5_USART_Write(&welcomeData[0],sizeof(welcomeData));

    SERCOM1_SPI_WriteRead(&txData[0], sizeof(txData), &rxData[0], sizeof(rxData));

    /* Compare received data with the transmitted data */
    if ((memcmp(txData, rxData, sizeof(txData)) == 0))
    {
        /* Pass: Received data is same as transmitted data */
        DMAC_ChannelTransfer(DMAC_CHANNEL_0, "Received SPI Valid Data\r\n", \
        (const void *)&(SERCOM5_REGS->USART_INT.SERCOM_DATA), \
        25);
    }
    else
    {
        /* Fail: Received data is not same as transmitted data */
        DMAC_ChannelTransfer(DMAC_CHANNEL_0, "Received Invalid SPI Data\r\n", \
        (const void *)&(SERCOM5_REGS->USART_INT.SERCOM_DATA), \
        27);
    }
}

```

11.单击Clean and Build按钮，清除并编译您的应用程序。

12.单击Make and Program按钮，将您的应用程序编程到器件。

验证扩展功能——SPI环回

1. 使用一根连接线将SAM D21 Curiosity Nano评估工具包上的PA16与PA19相连。
2. 打开Tera Term，然后观察数据，如下所示。

```

COM17 - Tera Term VT
File Edit Setup Control Window Help
Hello World
Received SPI Valid Data
Hello World
Received SPI Valid Data

```

仍可以通过串行终端发送命令来控制LED0。

结果

观察到应用程序从ATmega4809移植到SAMD21。成功使用USART和STDIO打印消息与接收命令。成功使用端口和EIC控制LED，还配置了SPI和DMA以进行自环回测试。

分析

您已经使用MPLAB Harmony v3基于SAM D21单片机成功创建了第一个应用程序。您的应用程序使用了编译实时应用程序所需的全部基本元素。您已经成功将应用程序从ATmega4809移植到SAMD21。在本应用程序中，您使用了MPLAB Harmony配置器（MHC）来配置SAM D21，并且还使用了MPLAB Harmony v3框架。您使用了时钟配置器来设置CPU时钟，并且还配置了Systick。您配置了SERCOM 5（作为通用同步/异步收发器（USART））、SERCOM 1（SPI）和外部中断控制器（EIC）外设库（PLIB）。您还使用DMA配置器配置了直接存储器访问（DMA）。您使用了引脚配置器来设置用于LED和开关功能的引脚。

结论

本指南概述了将应用程序从ATmega4809移植到SAM D21的过程，还提供了使用MPLAB Harmony v3框架基于SAM D21单片机编译应用程序时所需的全部基本组件的配置和使用步骤。

参考链接



32-bit MCUs



32-bit MPUs



MPLAB X IDE



MPLAB Harmony



MPLAB XC32 Compiler

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之中。Microchip承诺将不断改进产品的代码保护功能。

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，不得暗或以其他方式转让任何许可证。

有关Microchip质量管理体系的更多信息，请访问 www.microchip.com/quality。

商标

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

AgileSwitch、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、TrueTime、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、Augmented Switching、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、Espresso T1S、EtherGREEN、GridTime、IdealBridge、In-Circuit Serial Programming、ICSP、INICnet、Intelligent Paralleling、Inter-Chip Connectivity、JitterBlocker、Knob-on-Display、maxCrypto、maxView、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、NVM Express、NVMe、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、RTAX、RTG4、SAM-ICE、Serial Quad I/O、simpleMAP、SimpliPHY、SmartBuffer、SmartHLS、SMART-I.S.、storClad、SQI、SuperSwitcher、SuperSwitcher II、Switchtec、SynchroPHY、Total Endurance、TSHARC、USBCheck、VariSense、VectorBlox、VeriPHY、ViewSpan、WiperLock、XpressConnect和ZENA均为Microchip Technology Incorporated在美国和其他国家或地区的商标。

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

Adaptec徽标、Frequency on Demand、Silicon Storage Technology、Symmcom和Trusted Time均为Microchip Technology Inc.在除美国外的国家或地区的注册商标。

GestIC为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。

© 2021, Microchip Technology Incorporated及其子公司版权所有。



全球销售及服务中心

美洲

公司总部Corporate Office

2355 West Chandler Blvd.

Chandler, AZ 85224-6199

Tel: 1-480-792-7200

Fax: 1-480-792-7277

技术支持:

<http://www.microchip.com/support>

网址: www.microchip.com

亚特兰大Atlanta

Duluth, GA

Tel: 1-678-957-9614

Fax: 1-678-957-1455

奥斯汀Austin, TX

Tel: 1-512-257-3370

波士顿Boston

Westborough, MA

Tel: 1-774-760-0087

Fax: 1-774-760-0088

芝加哥Chicago

Itasca, IL

Tel: 1-630-285-0071

Fax: 1-630-285-0075

达拉斯Dallas

Addison, TX

Tel: 1-972-818-7423

Fax: 1-972-818-2924

底特律Detroit

Novi, MI

Tel: 1-248-848-4000

休斯敦Houston, TX

Tel: 1-281-894-5983

印第安纳波利斯

Indianapolis

Noblesville, IN

Tel: 1-317-773-8323

Fax: 1-317-773-5453

Tel: 1-317-536-2380

洛杉矶Los Angeles

Mission Viejo, CA

Tel: 1-949-462-9523

Fax: 1-949-462-9608

Tel: 1-951-273-7800

罗利Raleigh, NC

Tel: 1-919-844-7510

纽约New York, NY

Tel: 1-631-435-6000

圣何塞San Jose, CA

Tel: 1-408-735-9110

Tel: 1-408-436-4270

加拿大多伦多Toronto

Tel: 1-905-695-1980

Fax: 1-905-695-2078

亚太地区

中国 - 北京

Tel: 86-10-8569-7000

中国 - 成都

Tel: 86-28-8665-5511

中国 - 重庆

Tel: 86-23-8980-9588

中国 - 东莞

Tel: 86-769-8702-9880

中国 - 广州

Tel: 86-20-8755-8029

中国 - 杭州

Tel: 86-571-8792-8115

中国 - 南京

Tel: 86-25-8473-2460

中国 - 青岛

Tel: 86-532-8502-7355

中国 - 上海

Tel: 86-21-3326-8000

中国 - 沈阳

Tel: 86-24-2334-2829

中国 - 深圳

Tel: 86-755-8864-2200

中国 - 苏州

Tel: 86-186-6233-1526

中国 - 武汉

Tel: 86-27-5980-5300

中国 - 西安

Tel: 86-29-8833-7252

中国 - 厦门

Tel: 86-592-238-8138

中国 - 香港特别行政区

Tel: 852-2943-5100

中国 - 珠海

Tel: 86-756-3210040

台湾地区 - 高雄

Tel: 886-7-213-7830

台湾地区 - 台北

Tel: 886-2-2508-8600

台湾地区 - 新竹

Tel: 886-3-577-8366

亚太地区

澳大利亚Australia - Sydney

Tel: 61-2-9868-6733

印度India - Bangalore

Tel: 91-80-3090-4444

印度India - New Delhi

Tel: 91-11-4160-8631

印度India - Pune

Tel: 91-20-4121-0141

日本Japan - Osaka

Tel: 81-6-6152-7160

日本Japan - Tokyo

Tel: 81-3-6880-3770

韩国Korea - Daegu

Tel: 82-53-744-4301

韩国Korea - Seoul

Tel: 82-2-554-7200

马来西亚

Malaysia - Kuala Lumpur

Tel: 60-3-7651-7906

马来西亚Malaysia - Penang

Tel: 60-4-227-8870

菲律宾Philippines - Manila

Tel: 63-2-634-9065

新加坡Singapore

Tel: 65-6334-8870

泰国Thailand - Bangkok

Tel: 66-2-694-1351

越南Vietnam - Ho Chi Minh

Tel: 84-28-5448-2100

欧洲

奥地利Austria - Wels

Tel: 43-7242-2244-39

Fax: 43-7242-2244-393

丹麦

Denmark - Copenhagen

Tel: 45-4485-5910

Fax: 45-4485-2829

芬兰Finland - Espoo

Tel: 358-9-4520-820

法国France - Paris

Tel: 33-1-69-53-63-20

Fax: 33-1-69-30-90-79

德国Germany - Garching

Tel: 49-8931-9700

德国Germany - Haan

Tel: 49-2129-3766400

德国Germany - Heilbronn

Tel: 49-7131-72400

德国Germany - Karlsruhe

Tel: 49-721-625370

德国Germany - Munich

Tel: 49-89-627-144-0

Fax: 49-89-627-144-44

德国Germany - Rosenheim

Tel: 49-8031-354-560

以色列Israel - Ra'anana

Tel: 972-9-744-7705

意大利Italy - Milan

Tel: 39-0331-742611

Fax: 39-0331-466781

意大利Italy - Padova

Tel: 39-049-7625286

荷兰Netherlands - Drunen

Tel: 31-416-690399

Fax: 31-416-690340

挪威Norway - Trondheim

Tel: 47-7288-4388

波兰Poland - Warsaw

Tel: 48-22-3325737

罗马尼亚

Romania - Bucharest

Tel: 40-21-407-87-50

西班牙Spain - Madrid

Tel: 34-91-708-08-90

Fax: 34-91-708-08-91

瑞典Sweden - Gothenberg

Tel: 46-31-704-60-40

瑞典Sweden - Stockholm

Tel: 46-8-5090-4654

英国UK - Wokingham

Tel: 44-118-921-5800

Fax: 44-118-921-5820