

使用 MPLAB® Harmony v3 和 MPLAB 代码配置器 (MCC) 在 PIC32CM LSx 单片机上创建第一个应用程序



简介

MPLAB® Harmony v3 是一个软件开发框架，由兼容且可互操作的模块组成，其中包括外设库（PLIB）、驱动程序、系统服务、中间件和第三方库。MPLAB 代码配置器（MPLAB Harmony Configurator, MCC）是一款基于图形用户界面（Graphical User Interface, GUI）的工具，可轻松使能和配置 MPLAB Harmony 的各种模块。MCC 是 MPLAB X 集成开发环境（Integrated Development Environment, IDE）的插件。

本文档介绍如何搭配使用 MCC 与 MPLAB Harmony v3 的各种模块在基于 Arm® Cortex®-M23 内核的 PIC32CM LSx 单片机上创建简单的应用程序。该应用程序演示了 PIC32CM LSx 单片机上基于 TrustZone®的安全功能。该应用程序由两个项目组成，分别详细介绍了 PIC32CM LSx 单片机上的安全模式和非安全模式。这两个项目在器件中的可信资源与不可信资源之间提供了安全隔离。该应用程序旨在基于超时切换 LED，并在串行控制台上打印 LED 切换速率。在本演示中，使用 MPLAB Harmony v3 的以下模块并通过 MCC 进行安全和非安全配置。

安全模块包括：

- 安全端口引脚，用于切换 LED（默认情况下，所有端口引脚均配置为安全）。
- 安全实时时钟（Real-Time Clock, RTC）PLIB，用于定期采样 LED 切换速率。
- 安全外部中断控制器（External Interrupt Controller, EIC）PLIB，用于在开关被按下时更改切换速率。

非安全模块包括：

- 非安全 SERCOM（配置为 USART）和非安全直接存储器访问（Direct Memory Access, DMA）PLIB，用于将 LED 切换速率打印在 PC 上运行的 COM（串行）端口终端应用程序上。
- 非安全端口引脚（USART 引脚负责在从安全应用程序获取数据值后将其打印在终端上），用于与串行终端进行通信。

目录

简介.....	1
1. 基于 PIC32CM LS60 MCU 创建第一个可信区应用程序.....	3
1.1. 创建基于 MPLAB Harmony v3 的项目.....	3
1.2. 添加和配置 MPLAB Harmony 组件.....	5
1.3. 生成代码.....	11
1.4. 将应用程序逻辑添加到安全项目.....	11
1.5. 将应用程序代码添加到非安全项目.....	14
1.6. 编译并编程应用程序.....	15
1.7. 观察电路板和串行终端上的输出.....	16
2. 参考资料.....	17
3. 版本历史.....	18
Microchip 信息.....	19
Microchip 网站.....	19
产品变更通知服务.....	19
客户支持.....	19
Microchip 器件代码保护功能.....	19
法律声明.....	19
商标.....	20
质量管理体系.....	21
全球销售及服务网点.....	22

1. 基于 PIC32CM LS60 MCU 创建第一个可信区应用程序

本演示使用以下软件和硬件工具：

- [MPLAB X IDE v6.20](#)
- [MCC 插件 v5.5.0](#)
- [MPLAB XC32 v4.35](#)
- MPLAB Harmony v3 资源库：[csp v3.18.2](#)
- [PIC32CM LS60 Curiosity Pro 评估工具包](#)

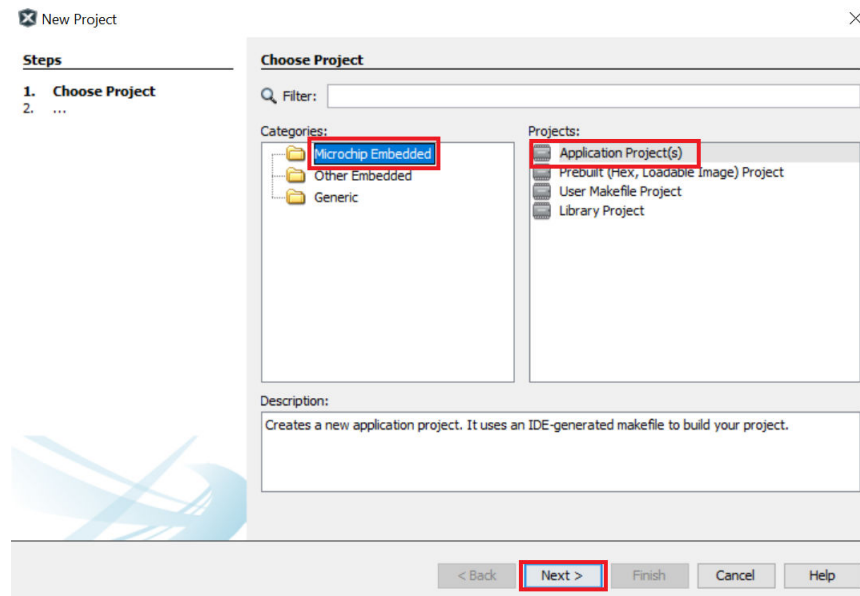
注：如果上面列出的硬件工具有更新版本，也可用于创建应用程序。

1.1 创建基于 MPLAB Harmony v3 的项目

要创建基于 MPLAB Harmony v3 的项目，请按照以下步骤操作：

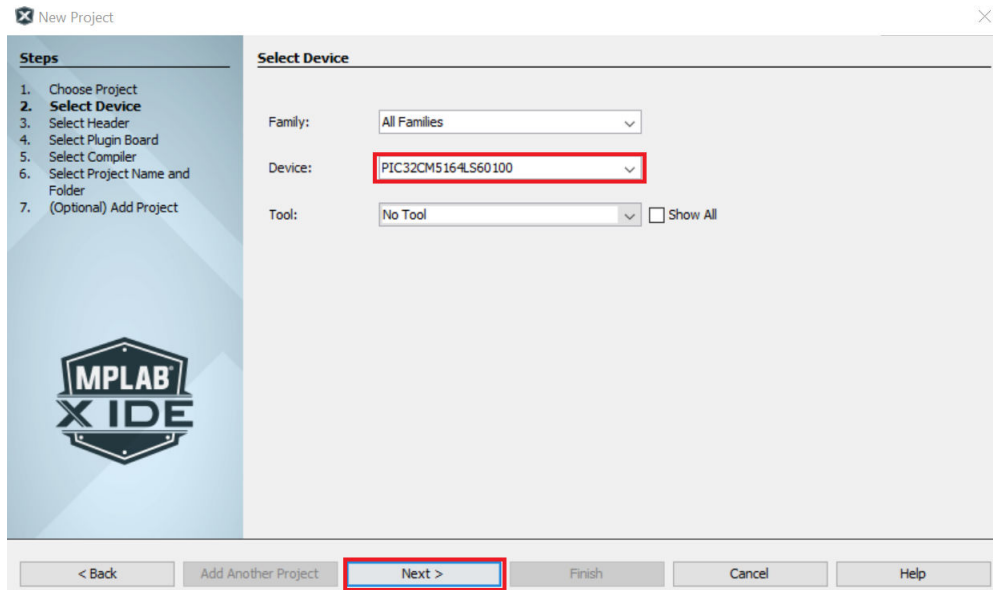
1. 从 Start（开始）菜单中启动 MPLAB X IDE。
2. 在 MPLAB X IDE 中，单击 **File**（文件）菜单中的 **New Project**（新建项目）或单击 *New Project* 图标。
3. 在 New Project 窗口左侧导航栏中的 Steps（步骤）下单击 **Choose Project**（选择项目）。
4. 在右侧 Choose Project 属性页面中：
 - a. Categories（类别）：选择 **Microchip Embedded**（Microchip 已安装工具）。
 - b. Projects（项目）：选择 **Application Project(s)**（应用程序项目）。

图 1-1. Choose Project



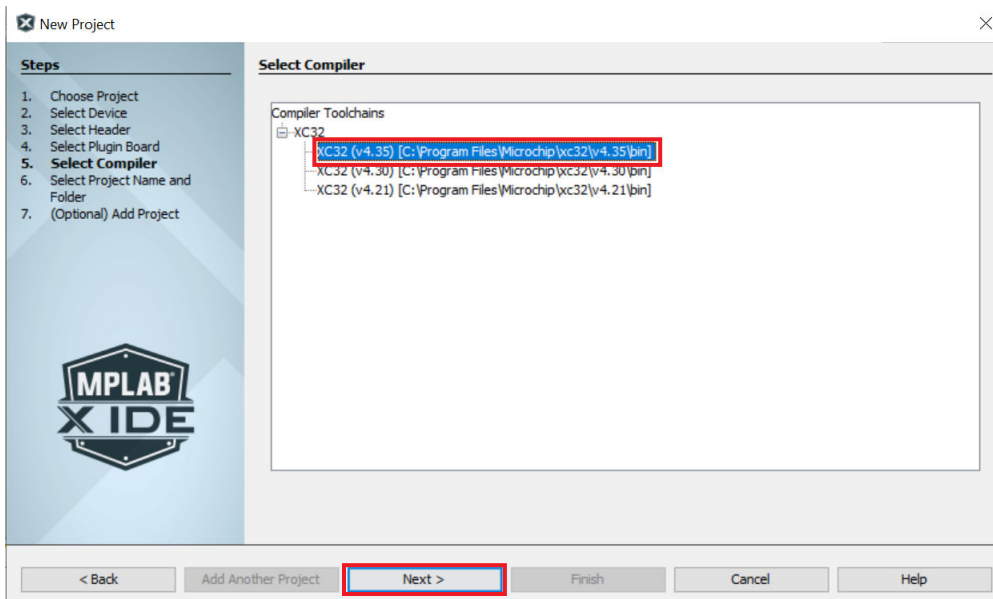
5. 单击 **Next**（下一步）。
6. 在左侧导航栏中，单击 **Select Device**（选择器件）。
7. 在 Select Device 属性页面的 Device（器件）框中，键入或选择器件 PIC32CM5164LS60100。

图 1-2. Select Device



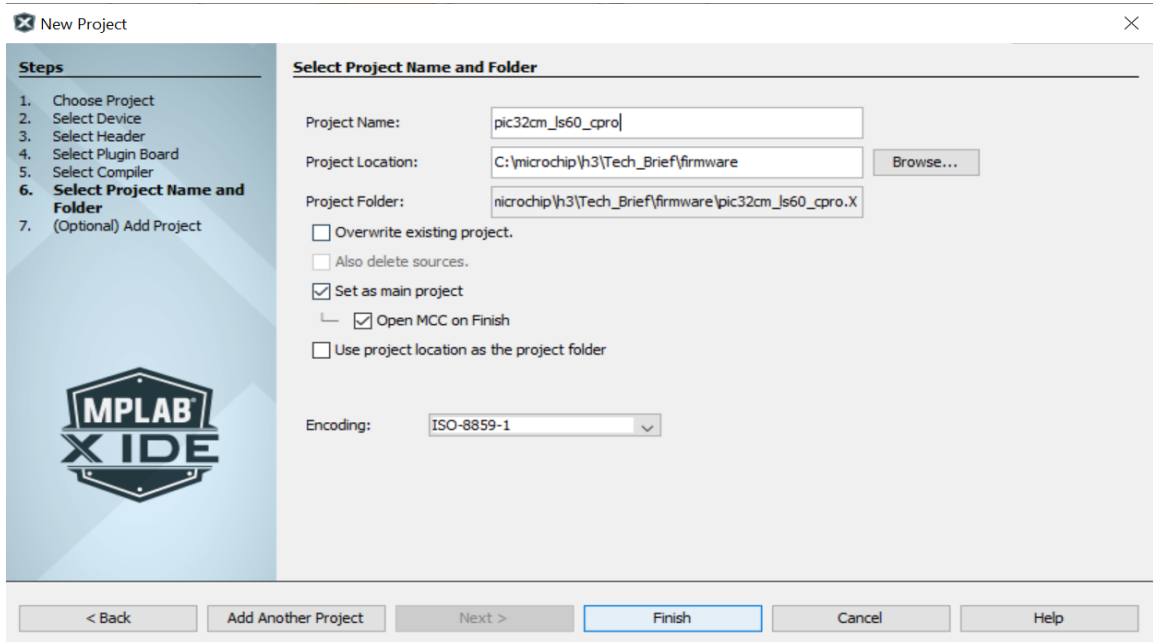
8. 单击 **Next**。
9. 在左侧导航栏中，单击 **Select Compiler**（选择编译器）。
10. 在 Select Compiler 属性页面中，单击展开 XC32 选项列表，然后选择如下所示的编译器工具链。

图 1-3. Select Compiler



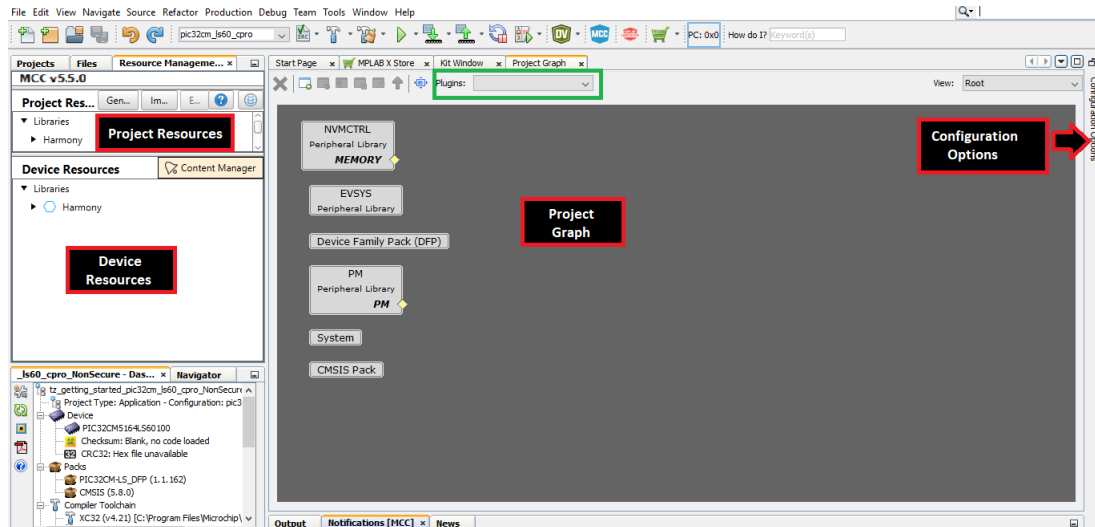
11. 单击 **Next**。
12. 在左侧导航栏中，单击 **Select Project Name and Folder**（选择项目名称和文件夹）。
13. 在右侧 Select Project Name and Folder 属性页面中：
 - Project name（项目名称）：输入 *pic32cm_ls60_cpro*。
 - Project Location（项目位置）：单击 **Browse...**（浏览...）按钮，然后选择 *C:\microchip\h31\Tech_Brief\firmware*。

图 1-4. Select Project Name and Folder



14. 单击 **Finish**（完成）以启动 MCC。
注：默认情况下，启动 MCC 时会将非安全项目设置为主项目。
15. 在启动 MCC 之前，将显示 Configuration Database Setup（配置数据库设置）窗口，此时可以根据需要在其中更改器件系列包（Device Family Pack, DFP）和 Cortex 单片机软件接口标准（Cortex Microcontroller Software Interface Standard, CMSIS）路径。在本演示中，使用默认设置。
16. MCC 插件将在新窗口中打开，如下图所示。

图 1-5. MPLAB®代码配置器窗口



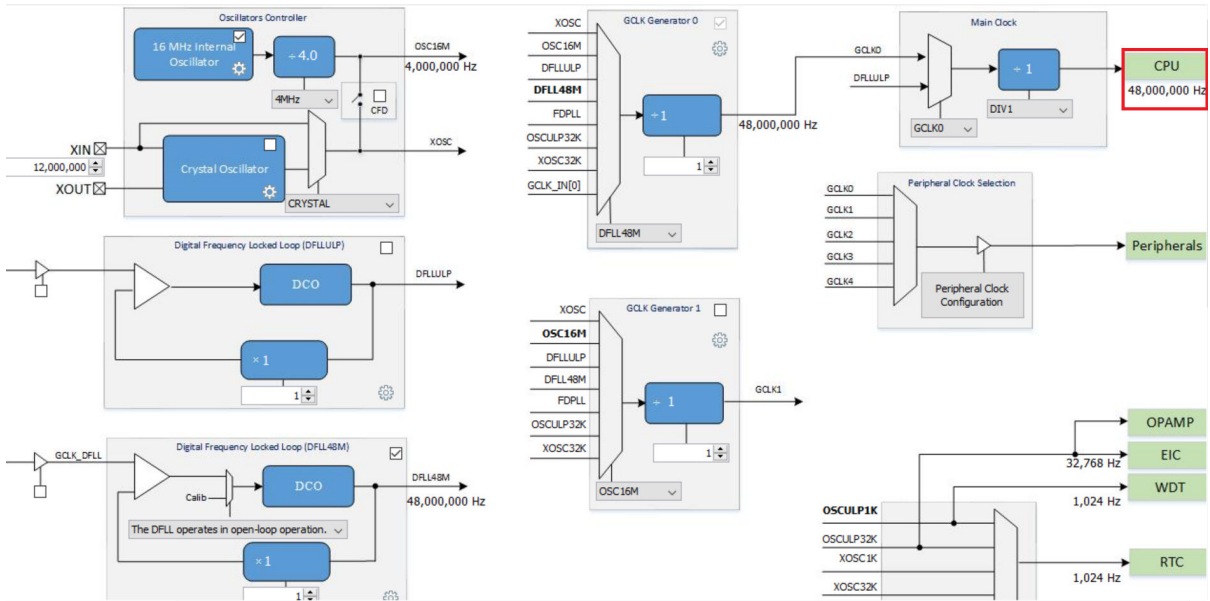
1.2 添加和配置 MPLAB Harmony 组件

要使用 MCC 添加和配置 MPLAB Harmony 组件，请按照以下步骤操作：

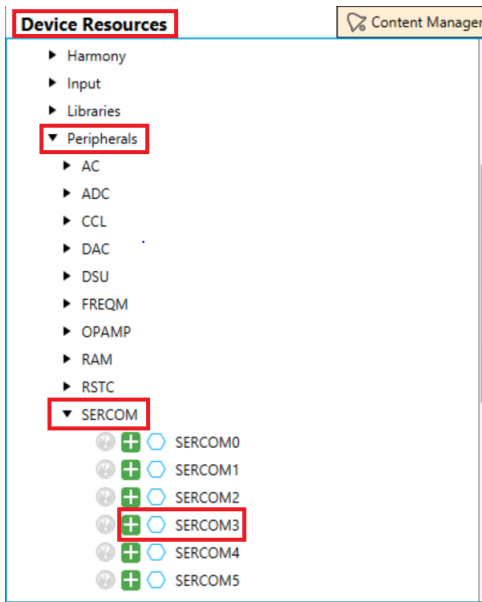
1. 在 MCC 窗口中，单击 **Project Graph**（项目图）。
2. 在 **Plugins**（插件）下拉列表（图 1.5 中的绿框部分）中，选择 **Clock Configuration**（时钟配置）。

- 随后将打开 Clock Easy View (时钟简化视图) 窗口, 该窗口内置于 MCC 中。确认主时钟是否设置为 48 MHz。

图 1-6. Clock Easy View 窗口



- 在 MCC 中的 Device Resources (器件资源) 部分下, 依次单击 *Harmony > Peripherals > SERCOM* (Harmony > 外设 > SERCOM) 以展开选项列表。
- 通过单击添加 **SERCOM3**。随后可观察到 *SERCOM3* 外设库模块已添加到 Project Graph 窗口中。

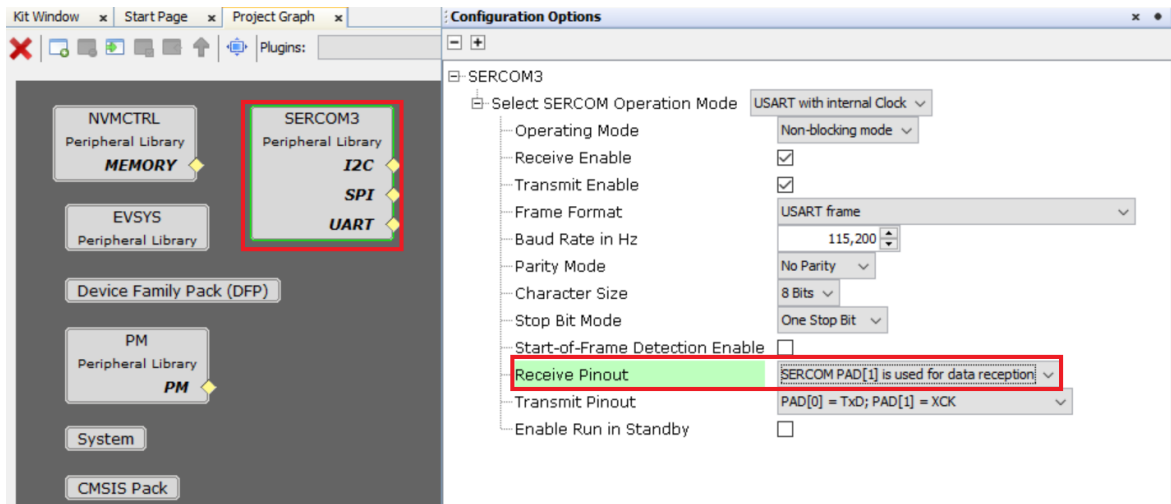


注: 用户可以按照同样的方式选择添加 *Device Resources > Harmony > Peripherals* (器件资源 > Harmony > 外设) 下的所有外设。

- 在 Project Graph 窗口的左侧导航栏中, 选择 **SERCOM3 Peripheral Library** (SERCOM3 外设库)。
- 在右侧 Configuration Option 属性页面中, 进行如下配置以将 LED 切换速率打印在串行控制台上。

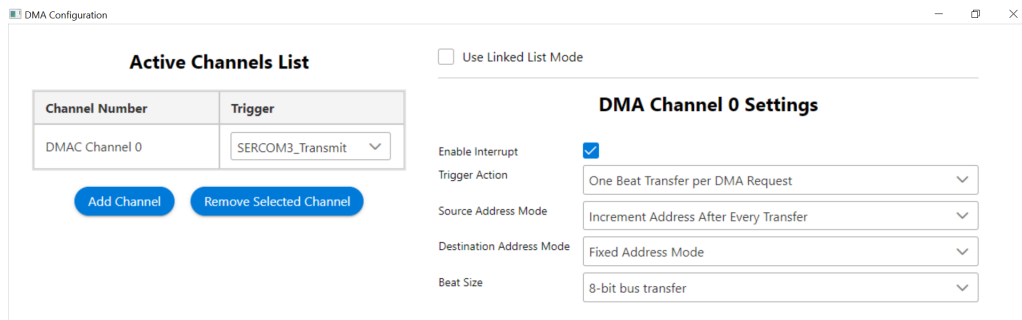
- a. 对于 Receive Pinout（接收引脚分配），从下拉列表中选择 SERCOM PAD[1]，其余参数保留默认设置，如下图所示。

图 1-7. MPLAB® 代码配置器——SERCOM3 配置窗口



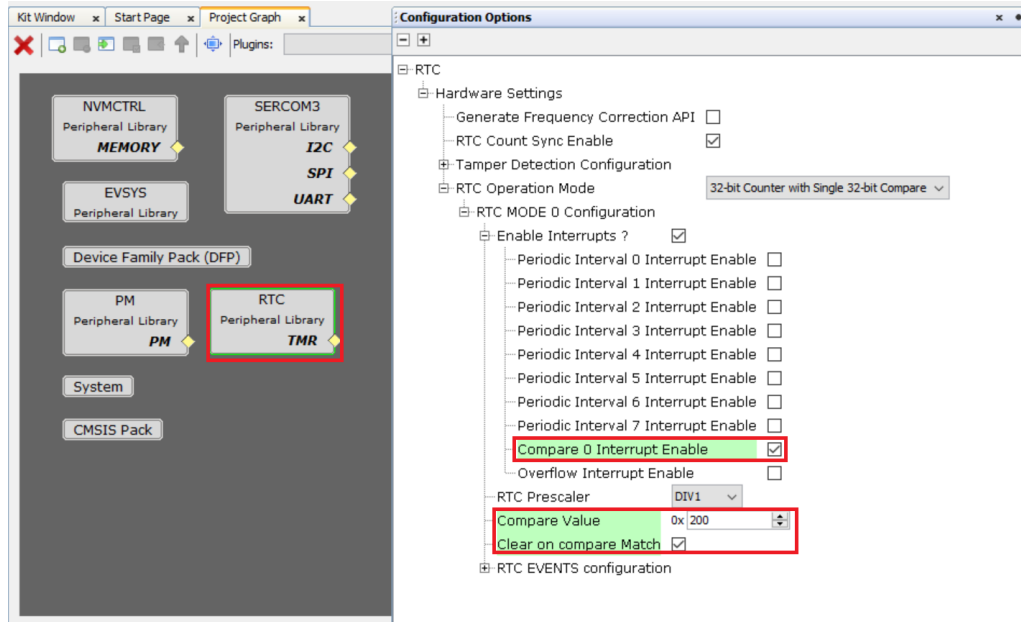
8. 从 **Plugins** 项目列表中选择 **DMA Configuration**（DMA 配置）。
9. 在 DMA Configuration 对话框中，单击 **Add Channel**（添加通道）并配置 DMA 通道 0 将应用程序缓冲区的内容传输到 USART TX 寄存器。每次触发时，DMA 都会将 1 个字节的用户缓冲区内容传输到 USART 发送缓冲区。

图 1-8. DMA 配置



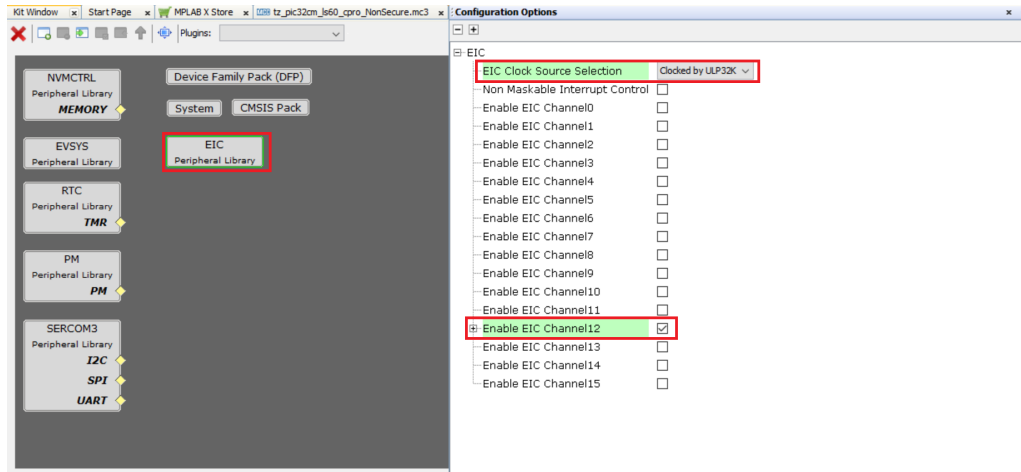
10. 在 Device Resources 下，单击展开 *Harmony > Peripherals > RTC*（Harmony > 外设 > RTC）。通过单击添加 **RTC**，随后可观察到 *RTC* 外设库已添加到 Project Graph 窗口中，每 500 毫秒产生一次比较中断。

图 1-9. RTC PLIB 配置



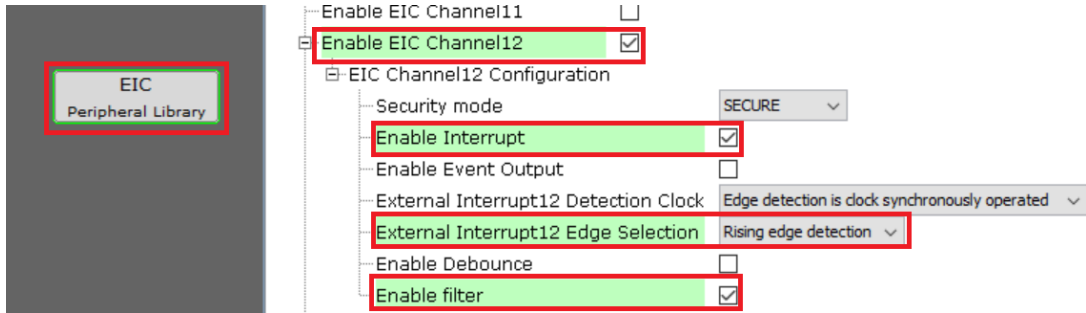
11. 在器件资源部分下，单击展开 *Harmony > Peripherals > EIC*（*Harmony > 外设 > EIC*）。
12. 通过单击添加 **EIC**，随后可观察到 *EIC* 外设库已添加到 Project Graph 窗口中。

图 1-10. 使能 EIC 通道 12



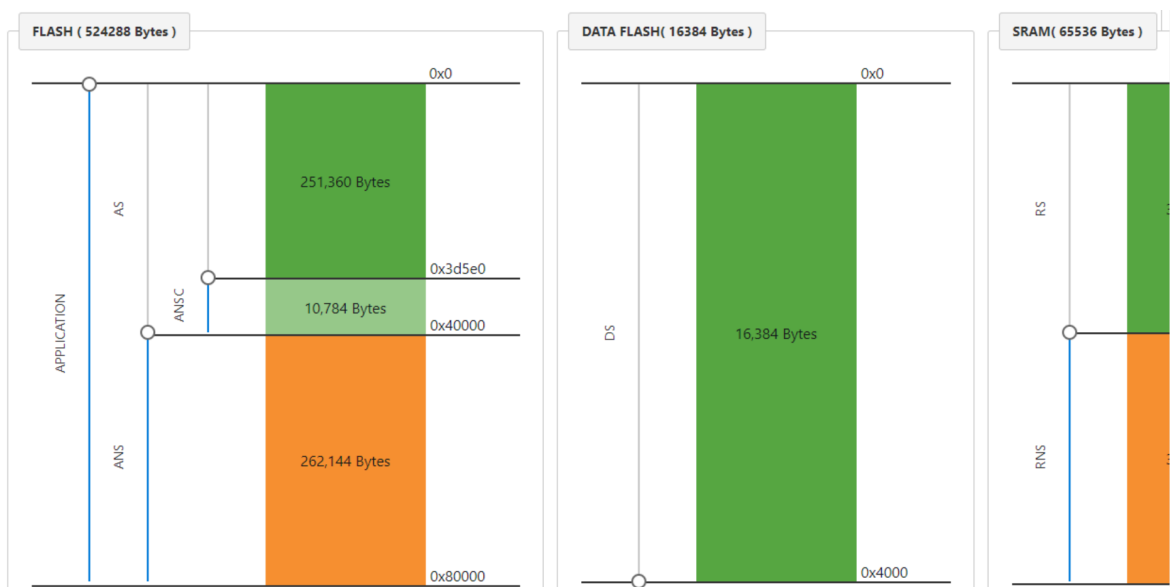
13. 单击 **EIC Peripheral Library**（*EIC* 外设库），然后展开 *Enable EIC Channel12*（使能 *EIC* 通道 12）并按照下图所示进行配置。

图 1-11. MPLAB® 代码配置器——EIC PLIB 配置



- 从 **Plugins** 列表中选择 **Arm TrustZone for Armv8-M**（Armv8-M 的 Arm TrustZone），然后检查应用程序的安全区域和非安全区域的存储器配置。

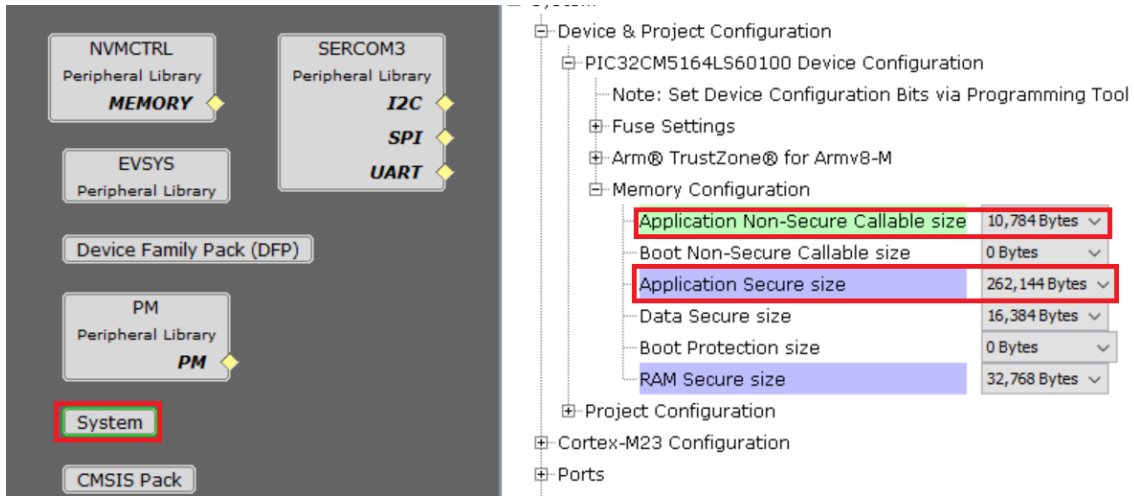
图 1-12. Arm® TrustZone® 存储器配置窗口



注：如果存储器配置需要进行任何更改，可以使用标记来配置存储器。建议不要更改此应用程序的默认配置。

可以使用树形视图界面来配置存储器，方法是选择 **System**（系统），然后按照下面屏幕截图中的红框部分进行选择。

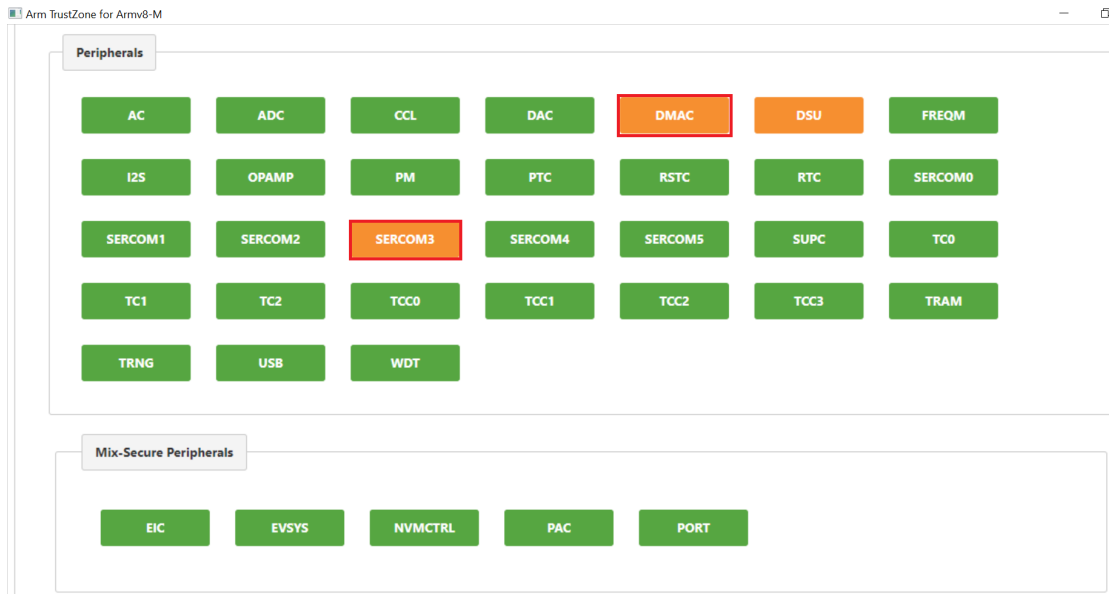
图 1-13. 使用 System 进行存储器配置



应用程序非安全可调用（Application Non-Secure Callable, ANSC）存储器大小已根据存储空间需求适当增大，以使用户将来根据需要添加代码。

- 从 **Plugins** 列表中选择 **Arm TrustZone for Armv8-M**，然后打开 **外设配置** 并选中 **SERCOM3** 和 **DMAC** 框作为非安全外设。选中后，框的颜色将从绿色变为橙色。

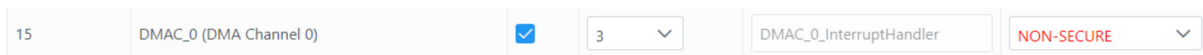
图 1-14. SERCOM3 和 DMAC 外设配置



注：SERCOM3 和 DMA 外设库配置为非安全外设。这两个库通过非安全可调用（Non-Secure Callable, NSC）API 从安全应用程序获取 LED 切换速率，以便在 PC 上运行的串行控制台上打印 LED 切换速率。

- 从 **Plugins** 列表中打开 **NVIC Configuration**（NVIC 配置），然后将 **DMAC_0** 通道设为非安全。

图 1-15. NVIC 配置（DMAC_0 设为非安全）



17. 从 **Plugins** 列表中打开 Pin Configuration（引脚配置）窗口并配置所需的引脚，具体如下所示：

图 1-16. SERCOM3 引脚配置

Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull Down	Drive Strength	Security Mode
68	PB20		SERCOM3_PAD0	Digital	High Impedance	n/a	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NORMAL	NON-SECURE
69	PB21		SERCOM3_PAD1	Digital	High Impedance	n/a	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NORMAL	NON-SECURE

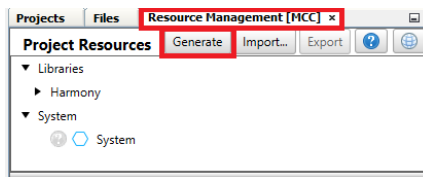
图 1-17. 开关和 LED 引脚配置

Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull Down	Drive Strength	Security Mode
58	PC18		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL	SECURE
59	PC19	LED0	GPIO	Digital	Out	High	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL	SECURE
60	PC20		EIC_EXTINT12	Digital	High Impedance	n/a	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NORMAL	SECURE
61	PC21		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL	SECURE

1.3 生成代码

配置外设后，单击 **Resource Management [MCC]**（资源管理[MCC]），然后单击 **Generate**（生成）选项卡。

图 1-18. 生成代码



注：生成的代码会将文件和文件夹添加到 32 位 MCC Harmony 项目中。请注意，在生成的代码中，针对实时时钟（RTC）、外部中断控制器（EIC）和 PORT 外设生成的外设库文件在安全项目中，而针对 SERCOM3（作为 USART）和 DMA 外设生成的外设库文件在非安全项目中。此外，MCC 还会在安全项目和非安全项目中生成 *main.c* 文件。

注：MCC 提供了一个选项来更改生成的文件名，如果不使用此选项，则生成的文件名默认为 *main.c*。

1.4 将应用程序逻辑添加到安全项目

要开发和运行该应用程序，请按照以下步骤操作：

1. 打开安全项目的 *main.c* 文件并添加以下应用程序逻辑。在 `main()` 函数和 `SYS_Initialize (NULL)` 函数中添加以下代码来注册针对 500 ms 比较事件的 RTC 事件处理程序

`RTC_Timer32CallbackRegister (rtcEventHandler, 0);`和针对开关按下事件的 EIC 回调事件处理程序 `EIC_CallbackRegister (EIC_PIN_12, sw0_eventHandler, 0);`，具体如下所示：

```
RTC_Timer32CallbackRegister(rtcEventHandler, 0);
EIC_CallbackRegister(EIC_PIN_12, sw0_eventHandler, 0);
sprintf((char*)uartTxTempBuffer,
"***** Printing Toggling LED rate
*****\r\n");
readUartTxStatus = true;
```

2. 添加如下代码行，以在注册回调事件处理程序之后调用 `RTC_Timer32Start ();` 函数。

图 1-19. 添加应用程序逻辑以注册回调事件处理程序

```

int main ( void )
{
    uint32_t msp_ns = *((uint32_t *) (TZ_START_NS));
    volatile funcptr_void NonSecure_ResetHandler;

    /* Initialize all modules */
    SYS_Initialize ( NULL );

    RTC_Timer32CallbackRegister(rtcEventHandler, 0);
    EIC_CallbackRegister(EIC_PIN_12, sw0_eventHandler, 0);

    sprintf((char*)uartTxTempBuffer, "***** Printing Toggling LED rate *****\r\n");
    readUartTxStatus = true;

    RTC_Timer32Start();
}

```

3. 通过在 main() 函数之外添加以下代码来实现针对安全外设注册的回调事件处理程序。

```

static void sw0_eventHandler(uintptr_t context)
{
    changeSamplingRate = true;
}

static void rtcEventHandler (RTC_TIMER32_INT_MASK intCause, uintptr_t context)
{
    if (intCause & RTC_TIMER32_INT_MASK_CMP0)
    {
        isRTCTimerExpired = true;
    }
}

```

4. 在 secureApp() 函数中，在安全项目的 main() 函数之前添加应用程序逻辑，以在用户按下电路板上的开关时按照 500 ms、1s、2s 和 4s 的不同速率切换 LED。

注：在 main() 函数之外添加该函数。

```

void secureApp(void)
{
    /* Basic Functionality: Demonstrates an LED toggle, i.e. LED0 toggles when
    * the switch SW0 is pressed on a timeout basis and prints the LED toggling
    * rate on the serial terminal.*/
    if (printLedToggleRate == true)
    {
        memset((char*)uartTxTempBuffer, 0x00, 100);
        sprintf((char*)uartTxTempBuffer, "***** Printing Toggling LED rate
        *****\r\n");
        printLedToggleRate = false;
        readUartTxStatus = true;
    }
    if (isRTCTimerExpired == true)
    {
        isRTCTimerExpired = false;
        memset((char*)uartTxTempBuffer, 0x00, 100);
        sprintf((char*)uartTxTempBuffer, "Toggling LED at %s rate \r\n",
        &timeouts[(uint8_t)tempSampleRate][0]);
        LED0_Toggle();
        readUartTxStatus = true;
    }
    if(changeSamplingRate == true)
    {
        changeSamplingRate = false;
        if(tempSampleRate == SAMPLING_RATE_500MS)
        {
            tempSampleRate = SAMPLING_RATE_1S;
            RTC_Timer32CompareSet(PERIOD_1S);
        }
        else if(tempSampleRate == SAMPLING_RATE_1S)
        {
            tempSampleRate = SAMPLING_RATE_2S;
            RTC_Timer32CompareSet(PERIOD_2S);
        }
        else if(tempSampleRate == SAMPLING_RATE_2S)
        {
            tempSampleRate = SAMPLING_RATE_4S;
        }
    }
}

```

```

        RTC_Timer32CompareSet(PERIOD_4S);
    }
    else if(tempSampleRate == SAMPLING_RATE_4S)
    {
        tempSampleRate = SAMPLING_RATE_500MS;
        RTC_Timer32CompareSet(PERIOD_500MS);
    }
    else
    {
        ;
    }
    RTC_Timer32CounterSet(0);
    sprintf((char*)uartTxTempBuffer, "LED Toggling rate is changed to %s\r\n",
&timeouts[(uint8_t)tempSampleRate][0]);
    readUartTxStatus = true;
}
}

```

添加以下代码片段以包含必要的头文件，并定义不同 RTC 比较值的宏。

注：在文件开头添加该逻辑，以包含文件中使用的函数定义所在的必要文件。

```

#include <stdio.h>
#include <string.h>

#define PERIOD_500MS          512
#define PERIOD_1S            1024
#define PERIOD_2S            2048
#define PERIOD_4S            4096

```

在包含头文件之后添加以下代码片段。

以下代码声明各种标志，这些标志的状态由应用程序中的事件处理程序进行监视和更改。

```

static volatile bool isRTCTimerExpired = false;
static volatile bool changeSamplingRate = false;
static volatile bool printLedToggleRate = false;
static const char timeouts[4][20] = {"500 milliSeconds", "1 Second", "2 Seconds", "4
Seconds"};

volatile bool readUartTxStatus = false;

uint8_t uartTxTempBuffer[100] = {0};

typedef enum
{
    SAMPLING_RATE_500MS = 0,
    SAMPLING_RATE_1S    = 1,
    SAMPLING_RATE_2S    = 2,
    SAMPLING_RATE_4S    = 3,
} SAMPLING_RATE;

static SAMPLING_RATE tempSampleRate = SAMPLING_RATE_500MS;

```

5. 在 `nonsecure_entry.c` 文件（位于 *Source Files* > *trustZone*（源文件 > *trustZone*）下）中实现以下非安全可调用函数，以便从非安全应用程序访问和请求安全应用程序。

注：删除生成的模板代码并添加以下代码。

```

bool __attribute__((cmse_nonsecure_entry)) readUartTxData(uint8_t *lcluartTxBuffer)
{
    bool localSecureUartStatus = readUartTxStatus;
    if(localSecureUartStatus == true)
    {
        memset((char*)lcluartTxBuffer, 0x00, 100);
        memcpy(lcluartTxBuffer, uartTxTempBuffer, strlen((const char
*)&uartTxTempBuffer[0]));
        readUartTxStatus = false;
    }
    return (localSecureUartStatus);
}

void __attribute__((cmse_nonsecure_entry)) secureAppEntry(void)
{

```

```
secureApp();
}
```

添加以下代码片段以包含必要的头文件，将安全 main.c 文件中实现的 *SecureApp()* 函数的变量和原型声明为外部。

```
#include <stdint.h>
#include <stdio.h>
#include <stddef.h>           // Defines NULL
#include <stdbool.h>         // Defines true
#include <stdlib.h>          // Defines EXIT_FAILURE
#include <string.h>
extern uint8_t uartTxTempBuffer[];
extern volatile bool readUartTxStatus;
extern void secureApp(void);
```

1.5 将应用程序代码添加到非安全项目

1. 在 main() 函数中的 SYS_Initialize() 下，添加以下代码来注册回调事件处理程序。

```
DMAC_ChannelCallbackRegister(DMAC_CHANNEL_0, uartTxDmaChannelHandler, 0);
```

2. 在 main() 函数之前实现已注册的回调事件处理程序。

```
static void uartTxDmaChannelHandler(DMAC_TRANSFER_EVENT event, uintptr_t contextHandle)
{
    if (event == DMAC_TRANSFER_EVENT_COMPLETE)
    {
        isUSARTTxComplete = true;
    }
}
```

3. 移除 SYS_Tasks() 函数调用并替换为以下代码，以在 while 循环中从非安全应用程序进入安全功能。

```
while ( true )
{
    if (readUartTxData(nonSecureUartTxBuffer) == true)
    {
        /* Maintain state machines of all polled MPLAB Harmony modules. */
        DMAC_ChannelTransfer(DMAC_CHANNEL_0, nonSecureUartTxBuffer, \
            (const void *)&(SERCOM3_REGS->USART_INT.SERCOM_DATA), \
            strlen((const char*)nonSecureUartTxBuffer));
    }
    secureAppEntry();
}
```

图 1-20. 添加应用程序逻辑以从非安全应用程序进入安全功能

```

int main ( void )
{
    /* Initialize all modules */
    SYS_Initialize ( NULL );

    DMAC_ChannelCallbackRegister(DMAC_CHANNEL_0, usartTxDmaChannelHandler, 0);
    while ( true )
    {
        if (readUartTxData(nonSecureUartTxBuffer) == true)
        {
            /* Maintain state machines of all polled MPLAB Harmony modules. */
            DMAC_ChannelTransfer(DMAC_CHANNEL_0, nonSecureUartTxBuffer, \
                (const void *)&(SERCOM3_REGS->USART_INT.SERCOM_DATA), \
                strlen((const char*)nonSecureUartTxBuffer));
        }
        secureAppEntry();
    }
}

```

添加以下代码片段以包含非安全 main.c 文件中使用的头文件和变量声明。

```

#include <stdio.h>
#include <string.h>
#include "trustZone/nonsecure_entry.h"

static volatile bool isUSARTTxComplete = false;
static volatile bool isUSARTRxComplete = false;
static uint8_t nonSecureUartTxBuffer[100] = {0};

```

- 在 nonsecure_entry.h 文件（位于 *Header Files* > *trustZone* 下）中添加以下代码，以通过使用 extern 关键字声明 NSC 来从非安全应用程序访问和请求安全应用程序。

注：删除生成的模板代码并添加以下代码。

```

extern bool readUartTxData(uint8_t *lcluartTxBuffer);
extern void secureAppEntry(void);

```

图 1-21. 用于从非安全应用程序访问和请求安全应用程序的全局 NSC

```

#ifndef NONSECURE_ENTRY_H_
#define NONSECURE_ENTRY_H_

/* Non-secure callable functions */
extern bool readUartTxData(uint8_t *lcluartTxBuffer);
extern void secureAppEntry(void);

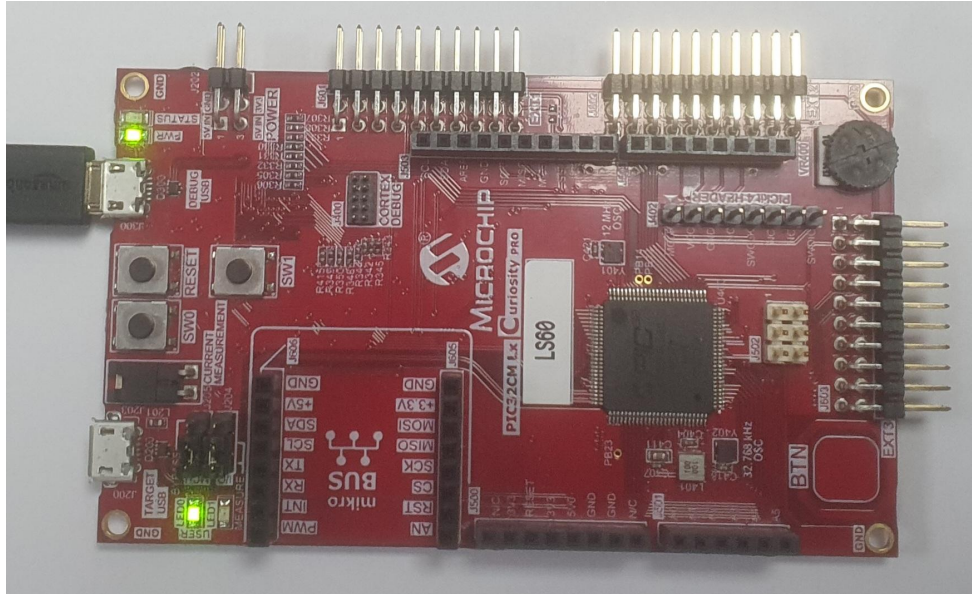
#endif /* NONSECURE_ENTRY_H_ */

```

1.6 编译并编程应用程序

- PIC32CM LS60 Curiosity Pro 评估工具包支持使用嵌入式调试器（EDBG）进行调试。将“Type-A 公头转 micro-B” USB 线缆连接到 PIC32CM LS60 Curiosity Pro 评估工具包上的 micro-B 调试 USB 端口进行供电和调试。

图 1-22. 硬件设置




2. 将非安全项目设置为主项目，然后从 Project Properties（项目属性）中选择最新编译器版本（v4.35）。



3. 单击 （Clean and Build（清除并编译）图标），清除并编译项目。

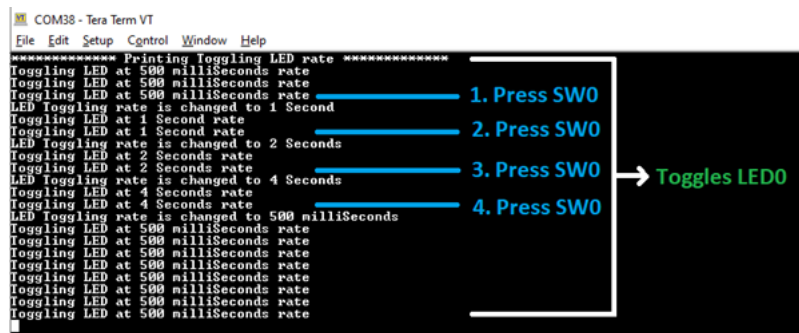


4. 单击 （Program the Device（编程器件）图标）对应用程序进行编程。

1.7 观察电路板和串行终端上的输出

1. 当应用程序编译并完成编程时，打开 PC 上的 *Tera Term* 工具。选择 **Serial Port**（串行端口）并将 *baud rate*（波特率）设置为 115200。
2. 按下 PIC32CM LS60 Curiosity Pro 评估工具包上的 *RESET* 按钮，LED 默认将以 500 ms 的速率切换，后续每次按下 *SWO* 开关时，LED 切换速率将依次变为 1s、2s 和 4s。

图 1-23. 串行终端上的 LED 切换速率



- a. 当串行终端上的 LED 切换速率随着开关每次被按下而发生变化时，可观察到评估工具包上 LED0 的切换速率也随之变化。

2. 参考资料

- 有关 MPLAB Harmony v3 的更多信息，请访问 Microchip 网站：<https://www.microchip.com/mplab/mplab-harmony> 和 microchipdeveloper.com/harmony3:start
- 有关该项目的更多详细信息，请观看 YouTube 频道上的 Microchip 开发人员帮助：www.youtube.com/watch?v=5w0JYHnSzPM
- 有关各种应用程序的更多信息，请参见 https://github.com/Microchip-MPLAB-Harmony/reference_apps。
- 有关 PIC32CM LS60 Curiosity Pro 评估工具包上的安全引导，请参见 <https://microchipdeveloper.com/harmony3:secure-boot-application-on-pic32cm-ls60>
- PIC32CM LS60 Curiosity Pro 评估工具包
- 有关 32 位单片机资料和解决方案的更多信息，请参见：<https://www.microchip.com.cn/newcommunity/Uploads/202404/6620805f4a70b.pdf>

3. 版本历史

版本 B——2024 年 4 月

本版本包含以下更新：

文档	少量格式更改。
简介	拆分了安全模块特性与非安全模块特性。
1. 基于 PIC32CM LS60 MCU 创建第一个可信区应用程序	<p>更新了软件工具的版本。</p> <p>更新了章节 1.1. 创建基于 MPLAB Harmony v3 的项目中的步骤。</p> <p>更新了图 1-1、图 1-2、图 1-3、1.4. 将应用程序逻辑添加到安全项目和图 1-5。</p> <p>更新了章节 1.2. 添加和配置 MPLAB Harmony 组件中的步骤。</p> <p>增加了图 1-6、图 1-13、图 1-14 和图 1-15。</p> <p>更新了图 1-7、图 1-9 和图 1-12。</p> <p>更新了章节 1.4. 将应用程序逻辑添加到安全项目中的步骤。</p> <p>更新了章节 1.5. 将应用程序代码添加到非安全项目中的步骤。</p> <p>更新了图 1-20。</p>
2. 参考资料	增加了新的参考资料。

版本 A——2023 年 9 月

本文档的初始版本。

Microchip 信息

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 www.microchip.com/en-us/support/design-help/client-support-services。

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。

商标

Microchip 的名称和徽标组合、Microchip 徽标、Adaptec、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、ClockWorks、The Embedded Control Solutions Company、EtherSynch、Flashtec、Hyper Speed Control、HyperLight Load、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus 徽标、Quiet-Wire、SmartFusion、SyncWorld、TimeCesium、TimeHub、TimePictra、TimeProvider 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、Clockstudio、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、Espresso T1S、EtherGREEN、EyeOpen、GridTime、IdealBridge、IGaT、In-Circuit Serial Programming、ICSP、INICnet、Intelligent Paralleling、IntelliMOS、Inter-Chip Connectivity、JitterBlocker、Knob-on-Display、MarginLink、maxCrypto、maxView、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、mSiC、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICKit、PICtail、Power MOS IV、Power MOS 7、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、Trusted Time、TSHARC、Turing、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 和 Symmcom 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。

© 2024, Microchip Technology Incorporated 及其子公司版权所有。

ISBN: 978-1-6683-4674-7

质量管理体系

有关 Microchip 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	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 - 霍德夏沙隆 电话: 972-9-775-5100 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
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			