
利用 MPLAB® Harmony v3 实时更新 SAM E54 MCU 上的应用程序

简介

SAM E54 单片机 (MCU) 上的双存储区闪存支持应用程序实现实时更新功能。应用程序中的实时更新功能是一段代码，用于将应用程序代码 (固件) 编程到内部闪存中的非活动存储区或非易失性存储器 (Non-volatile Memory, NVM)。实时更新功能会更新到较新版本的固件，而不会影响活动存储区中正在运行的应用程序。当前应用程序的实时更新功能正在运行中时，该应用程序的现有版本将继续运行。

以下是实时更新应用程序的功能：

- SAM E54 MCU 拥有双存储区闪存，分为存储区 A 和存储区 B，每个存储区都有一个自举程序和应用程序代码段。
- 首次将 UART 故障保护自举程序编程到存储区 A 中时，器件将在复位后从存储区 A 中运行。随后，使用脚本将实时更新应用程序与 UART 故障保护自举程序合并，并使用 UART 故障保护自举程序将其编程到存储区 B 中。
- 实时更新应用程序与主机程序 (通常在 PC 上运行) 通信以通过串行接口 (例如 UART) 接收固件。
- 实现实时更新功能的代码仅在有新的固件升级请求时运行，否则将处于空闲状态，从而允许其他应用程序功能运行。

本文档介绍了如何使用双存储区闪存存在 SAM E54 MCU 上实现实时更新应用程序。

目录

简介.....	1
1. 硬件和软件要求.....	3
1.1. SAM E54 Curiosity Ultra 开发板.....	3
1.2. MPLAB® X 集成开发环境 (Integrated Development Environment, IDE) 和 XC 编译器.....	3
1.3. MPLAB Harmony v3.....	3
1.4. Python.....	3
2. 设计.....	4
2.1. UART 实时更新协议.....	5
2.2. 命令说明.....	5
2.3. 通信任务.....	7
2.4. 命令处理任务.....	7
2.5. 编程任务.....	7
2.6. 存储器布局.....	8
2.7. 执行流程.....	9
3. 配置.....	13
3.1. 自举程序链接描述文件.....	13
3.2. MHC 配置.....	14
3.3. 项目设置.....	18
4. 运行应用程序.....	20
4.1. 运行自举程序应用程序.....	20
4.2. 运行实时更新应用程序.....	23
5. 结论.....	26
6. 参考文献.....	27
Microchip 网站.....	28
产品变更通知服务.....	28
客户支持.....	28
Microchip 器件代码保护功能.....	28
法律声明.....	28
商标.....	29
质量管理体系.....	30
全球销售及服务网点.....	31

1. 硬件和软件要求

1.1 SAM E54 Curiosity Ultra 开发板

SAM E54 Curiosity Ultra 开发板是用于评估 SAM E54 单片机的开发工具包。SAM E54 基于 Arm® Cortex® -M4，能够以 120 MHz 频率运行。SAM E54 Curiosity Ultra 开发板包含集成编程器和调试器，因此无需额外硬件即可开始使用。用户可以通过 MikroElektronika mikroBUS™ Click™ 适配器板来扩展功能，也可以使用 Microchip PHY 子板添加以太网连接，使用 Microchip 扩展板添加 Wi-Fi® 连接功能以及使用 Microchip 音频子板添加音频输入和输出功能。无论是否使用扩展板，用户都可以利用 SAM E54 Curiosity Ultra 开发板自由开发各种应用，包括 Bluetooth® 音频、CAN、图形用户界面（Graphical User Interface, GUI）、物联网（Internet of Things, IoT）、机器人开发和概念验证设计。

SAM E54 Curiosity Ultra 开发板可从 [Microchip 直销网站](#) 获取。

1.2 MPLAB® X 集成开发环境（Integrated Development Environment, IDE）和 XC 编译器

MPLAB X IDE 是一款可扩展且高度可配置的软件程序，包含多种功能强大的工具，可帮助发现、配置、开发、调试和验证大多数 Microchip 单片机的嵌入式设计。

MPLAB X IDE 可从 [Microchip 网站](#) 获取。本文档中讨论的应用程序使用 MPLAB X IDE 版本 5.40。MPLAB XC 编译器可从 [Microchip 网站](#) 获取。本文档中讨论的应用程序使用 MPLAB XC32 版本 2.41。

1.3 MPLAB Harmony v3

MPLAB Harmony v3 是一个完全集成的嵌入式软件开发框架，能够提供灵活且可互操作的软件模块，允许将资源专注于 32 位 PIC® 和 SAM 单片机应用程序的开发，而无需处理器件详细信息、复杂协议和库集成等挑战。

它包括 MPLAB Harmony 配置器（MPLAB Harmony Configurator, MHC），这是一种易于使用的带有 GUI 的开发工具，可简化器件设置、库选择、配置和应用程序开发。MPLAB Harmony MHC 可作为插件与 MPLAB X IDE 集成，它具有单独的 Java 可执行文件，可独立用于其他开发环境。

本文档中讨论的应用程序使用以下 MPLAB Harmony v3 资源库。用户可以从 GitHub 下载这些资源库或使用 [MPLAB Harmony 内容管理器（MPLAB Harmony Content Manager, MHCM）](#) 工具。

- [CSP](#)（芯片支持包）
- [DEV_PACKS](#)（Harmony 3 产品数据库）
- [MHC](#)（Harmony 3 配置器）
- [自举程序](#)（自举程序）
- [bootloader_apps_uart](#)（UART 自举程序应用程序）
- [MPLAB Harmony 参考应用程序](#)（MPLAB Harmony 3 参考应用程序）

1.4 Python

应用程序使用 Python v2.7 脚本来合并二进制文件（UART 故障保护自举程序和实时更新应用程序）并将合并的二进制文件从主机 PC 发送到 SAM E54 Curiosity Ultra 开发板。

2. 设计

SAM E54 MCU 上的实时更新应用程序的设计使用双存储区闪存功能。两个存储区命名为存储区 A 和存储区 B。在任何时候，应用程序都将其当前执行所处的存储区视为活动存储区，而将另一个存储区标记为非活动存储区。

使用双存储区闪存，可在运行来自活动存储区的当前版本固件的同时，向非活动存储区中编程新版本固件。

实时更新应用程序分为两个任务：

- 应用程序任务
- 实时更新任务

应用程序任务——代表最终用户应用程序。当前版本的最终用户应用程序将从双存储区闪存的活动存储区运行，而非活动存储区可用于使用最新版本的固件升级最终用户应用程序。本文档中讨论的应用程序任务本质上是极简的。应用程序任务检查 NVM 控制状态寄存器以标识闪存存储区，该闪存存在活动存储区中具有最新版本的固件，从存储区 A 运行时，每 500 ms 点亮一次 LED1 或 LED2，从存储区 B 运行时，每 1000 ms 点亮一次 LED1 或 LED2。例如，从点亮 LED1 开始，当有实时更新请求时，在向非活动存储区编程固件后，LED2 开始点亮。同样，每次成功更新应用程序固件都会重复该过程。

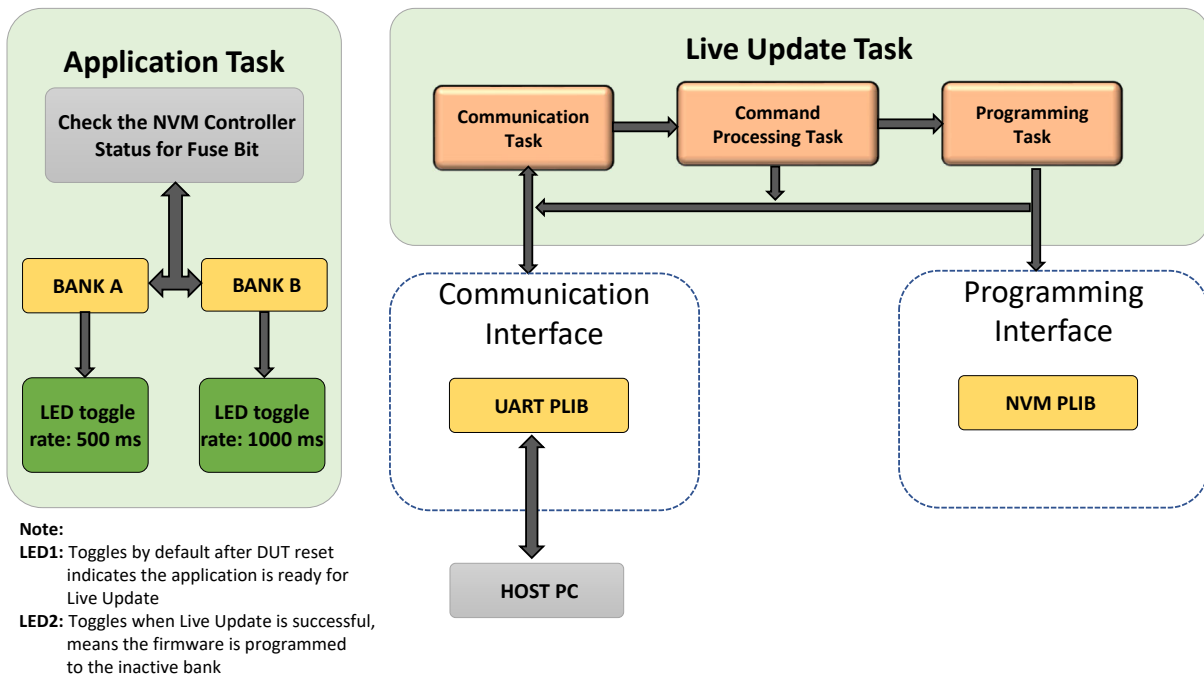
实时更新任务——通过预定义的通信协议（见 [UART 实时更新协议](#)）与个人计算机主机应用程序通信并在存在来自主机 PC 的请求时动态执行固件更新。否则，它将处于空闲状态，从而允许应用程序功能运行。

实时更新任务分为以下子任务：

- 通信任务。
- 命令处理任务。
- 编程任务。

下图说明了实时更新应用程序的设计。

图 2-1. 实时更新应用程序的设计



2.1 UART 实时更新协议

实时更新固件通过使用预定义的通信协议与个人计算机主机应用程序通信，以在目标与主机之间交换数据，协议的详细信息如下。

UART 实时更新协议包括保护、数据大小、命令和数据字节，如下图所示。

图 2-2. UART 实时更新协议



- **保护**
 - 保护字节是一个常量值 0x5048434D。
 - 该值可防止虚假命令。
 - 实时更新固件始终在数据包接收开始时检查保护值，并相应地继续后续操作。
- **数据大小**
 - 该字段指示要接收的数据字节数。
 - 该值因命令而异。
- **命令**
 - 指示要处理的命令。每条命令的宽度为 1 字节。
 - 支持的命令如下：
 - Unlock (0xA0)
 - Data (0xA1)
 - Verify (0xA2)
- **数据**
 - 包含要根据命令处理的数据。
 - 数据大小字段指示要接收的数据长度。
 - 自举程序接收的数据大小以字（4 个字节）为单位。
 - 所有数据字必须以小尾数法（LSB 在前）格式发送。

响应代码

实时更新将发送一个单字符响应代码来响应每条命令。只能在接收到前一条命令的响应代码后或在 100 ms 超时（没有响应）后发送后续命令。

有效的响应代码如下：

- OK (0x50)——成功接收和处理命令
- Error (0x51)——处理命令期间出错
- Invalid (0x52)——接收到无效命令
- CRC OK (0x53)——CRC 验证成功
- CRC Fail (0x54)——CRC 验证失败

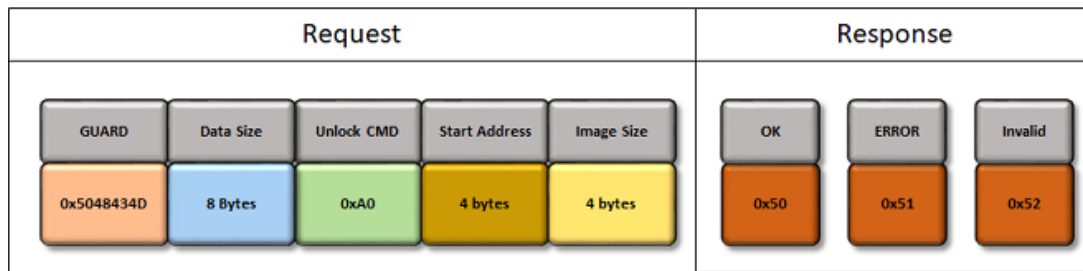
下一节[命令说明](#)给出了 UART 实时更新协议中使用的请求和响应代码的详细说明。

2.2 命令说明

Unlock 命令

带相应响应的 Unlock 命令序列如下图所示。

图 2-3. UART 实时更新 Unlock 命令

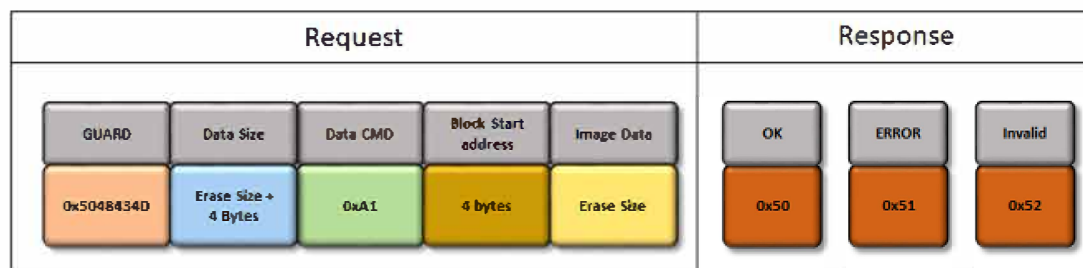


- Unlock 命令必须在第一个 Data 命令发出之前发出：
 - 该命令用于计算应用程序的起始地址和结束地址。
 - 该信息将用于验证发送的地址是否处于闪存范围内。
 - 该命令将用于验证要编程的数据包随附的地址是否处于调用 Unlock 命令的区域内。
- 要接收的数据字节数为 8 个字节（起始地址 + 映像大小）
- 起始地址：
 - 它是应用程序的起始地址。
 - 该地址取决于器件，应始终大于或等于自举程序的结束地址。
 - 必须在擦除单元大小边界处对齐。
 - 要自行升级自举程序，该值必须设置为 0。
- 映像大小必须以擦除单元字节为增量，具体取决于器件。有关擦除单元大小的更多信息，请参见 [MPLAB Harmony 自举程序帮助](#)中的“自举程序大小和注意事项”。

Data 命令

带相应响应的 Data 命令序列如下图所示。

图 2-4. UART 实时更新 Data 命令

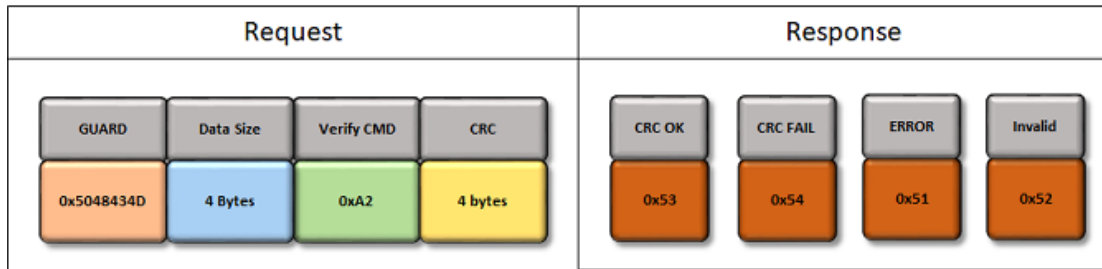


- Data 命令用于发送映像数据。
- 数据大小等于块起始地址的总和（4 个字节）。
- 擦除大小是要擦除的闪存的大小（取决于器件）。
- 块起始地址必须位于之前通过 Unlock 命令解锁的区域内。
- 如果尝试请求在解锁区域以外执行写操作，则将导致错误，提供的数据将被丢弃。

Verify 命令

带相应响应的 Verify 命令序列如下图所示。

图 2-5. UART 实时更新 Verify 命令



- Verify 命令用于验证发送和编程的映像数据。
- 映像 CRC 是标准 IEEE[®] CRC32，其多项式为 0xEDB88320。
- 编程后，会根据从闪存中读取的值计算内部 CRC，因此该命令会进行全面验证。
- 映像 CRC 计算会在之前解锁的区域上进行。

2.3 通信任务

通信任务负责以中断模式通过选定的通信接口从主机 PC 或嵌入式主机接收数据。在将主机传入的数据包传送到命令处理器任务之前，通信任务会验证该数据包是否包含预期的报头信息。

2.4 命令处理任务

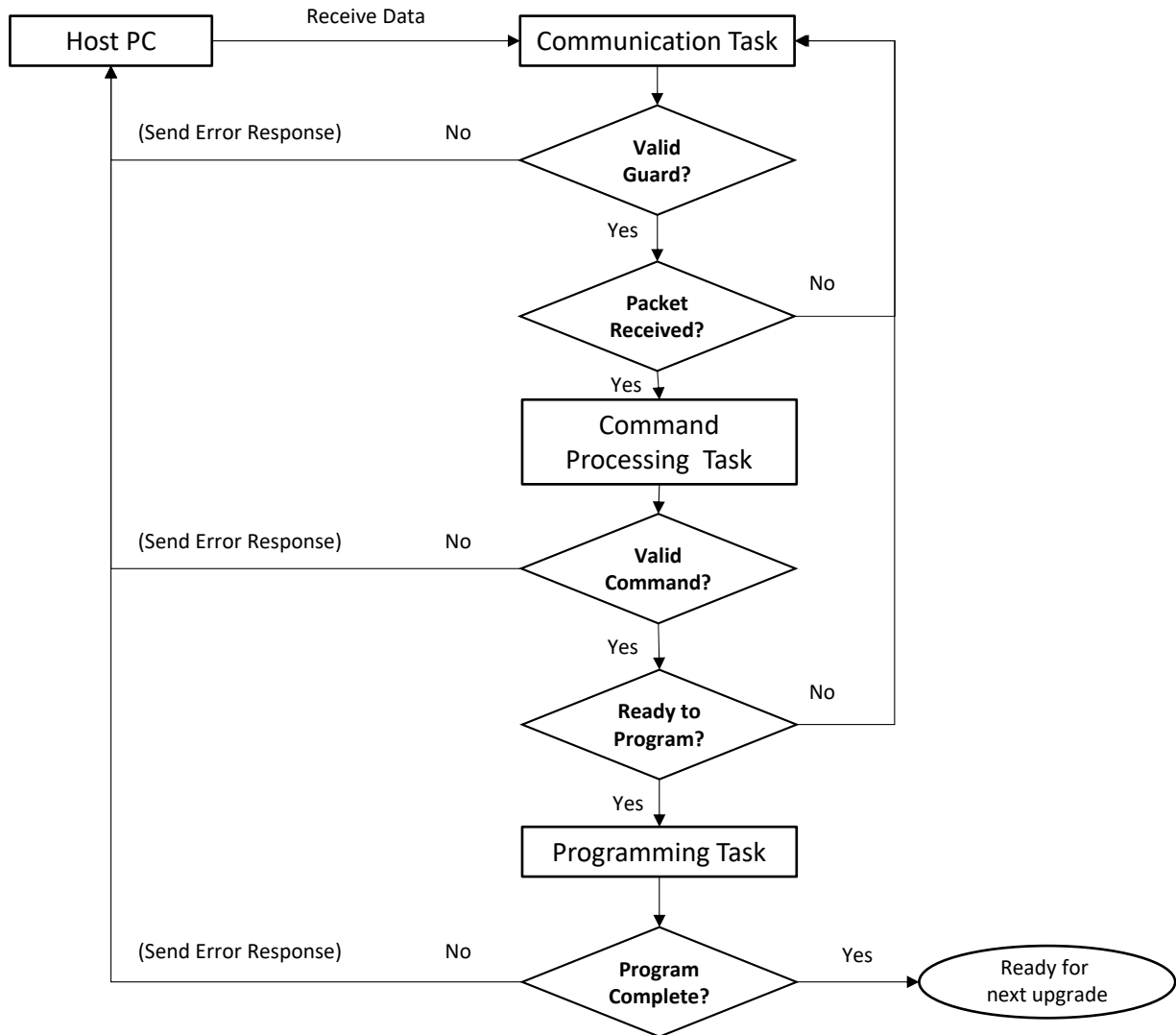
命令处理任务处理从通信任务接收的命令并作出响应。该任务将响应相应地返回给主机 PC，如果接收到的命令是编程命令，则该任务会将控制权交给编程任务。

2.5 编程任务

编程任务负责使用接收到的数据包对内部闪存进行编程。它使用 NVM 外设库执行解锁、擦除或写入操作，在等待闪存操作完成的同时并行调用通信任务以接收下一个数据包。

下图给出了固件升级执行的流程图：

图 2-6. 固件升级执行流程图



2.6 存储器布局

闪存中具有双存储区支持的 MCU 器件支持实时更新应用程序功能。以下各节介绍了 SAM E54 双存储区闪存的存储器布局以及自举程序和实时更新应用程序的位置和配置。

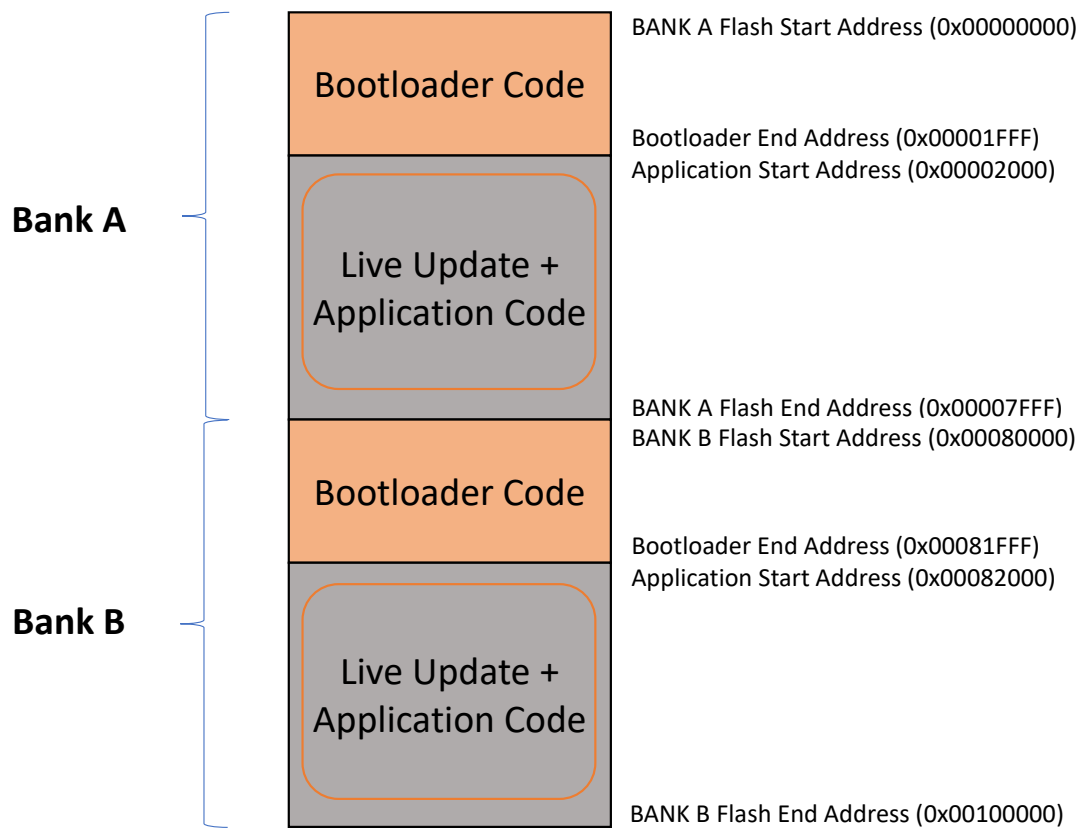
SAM E54 闪存配置为两个存储区：存储区 A 和存储区 B。两个存储区的开头均为自举程序，随后是应用程序映像。

默认情况下，存储区 A 映射到地址 0x00000000，存储区 B 映射到地址 0x00080000。映射到地址 0x00000000 的存储区称为活动存储区（默认为存储区 A），而映射到地址 0x00080000 的另一个存储区称为非活动存储区。

注：映射到地址 0x00000000 的存储区称为活动存储区，因为 Cortex-M CPU 架构设计为从地址 0x00000000 运行起始指令。因此，复位时需要运行的代码需要映射到 0x00000000。

下图给出了实时更新应用程序的 SAM E54 双存储区闪存布局：

图 2-7. 双存储区存储器中的实时更新应用程序



2.7 执行流程

使用 MPLAB Harmony v3 UART 故障保护自举程序首次对实时更新应用程序进行编程。以下主题将讨论 MPLAB Harmony v3 UART 故障保护自举程序和实时更新应用程序的系统级执行流程。

UART 故障保护自举程序系统级执行流程

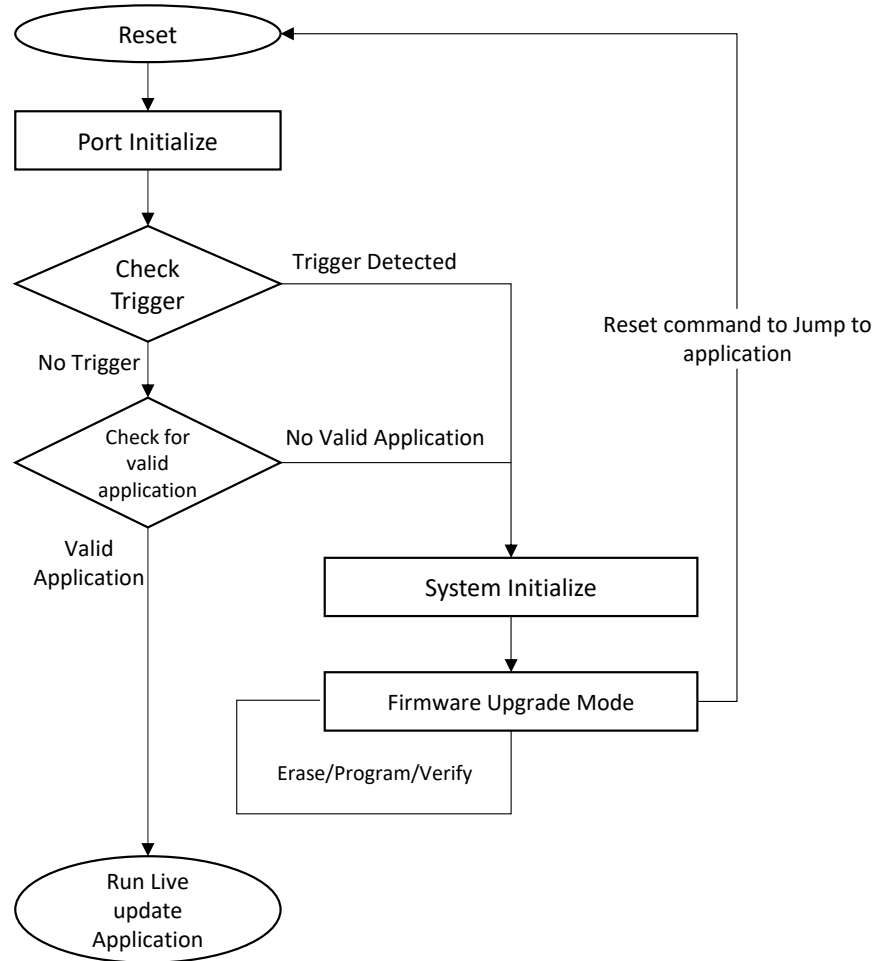
UART 故障保护自举程序代码在器件复位时开始执行。如果没有进入固件升级模式的条件，则 UART 故障保护自举程序开始执行用户应用程序。UART 故障保护自举程序在固件升级模式下执行闪存擦除和编程操作。

触发方法

- 触发方法将板上开关用作自举程序触发引脚，以便在器件复位时强制进入自举程序。
- 触发方法通过 RAM 的起始 16 个字节检查自举程序请求模式 (0x5048434D)，以便在器件复位时强制进入自举程序。

下图给出了首次编程实时更新应用程序的 MPLAB Harmony v3 UART 故障保护自举程序系统级执行流程。

图 2-8. 编程实时更新应用程序的 MPLAB Harmony v3 自举程序流程



从活动存储区运行的自举程序接收合并的映像（自举程序 + 实时更新应用程序）来升级地址 $0x00080000$ 处的非活动存储区。在自举程序成功执行升级后，会通知主机应用程序，实时更新应用程序等待用户按下开关 2 来执行存储区交换和系统复位（BKSWRST）。按下开关 2 会执行以下操作：

- 交换存储区以使非活动存储区处于活动状态，活动存储区处于非活动状态。存储区 A 处于非活动状态而存储区 B 处于活动状态。
- 发出复位命令以运行升级后的应用程序。

闪存的特殊熔丝位中自带有关映射到闪存地址 $0x00000000$ 的存储区的信息。这些熔丝位可以单独擦除或编程。当自举程序从主机接收到 BKSWRST 命令时，它会将闪存（NVM）控制寄存器中的 BKSWRST 位置 1。BKSWRST 置 1 时，闪存（NVM）控制器将交换存储区并根据熔丝位（STATUS.AFIRST）的最新状态设置熔丝位（STATUS.AFIRST），如下所示：

- STATUS.AFIRST = 0；存储区 B 的起始地址映射到 $0x00000000$
- STATUS.AFIRST = 1；存储区 A 的起始地址映射到 $0x00000000$

复位时，闪存（NVM）控制器将检查熔丝位（STATUS.AFIRST）的状态，并跳转到活动存储区以运行代码。

有关 MPLAB Harmony v3 UART 故障保护自举程序的更多信息，请参见 `<Harmony folder>/bootloader_apps_uart/apps_uart_fail_safe_bootloader/bootloader`。

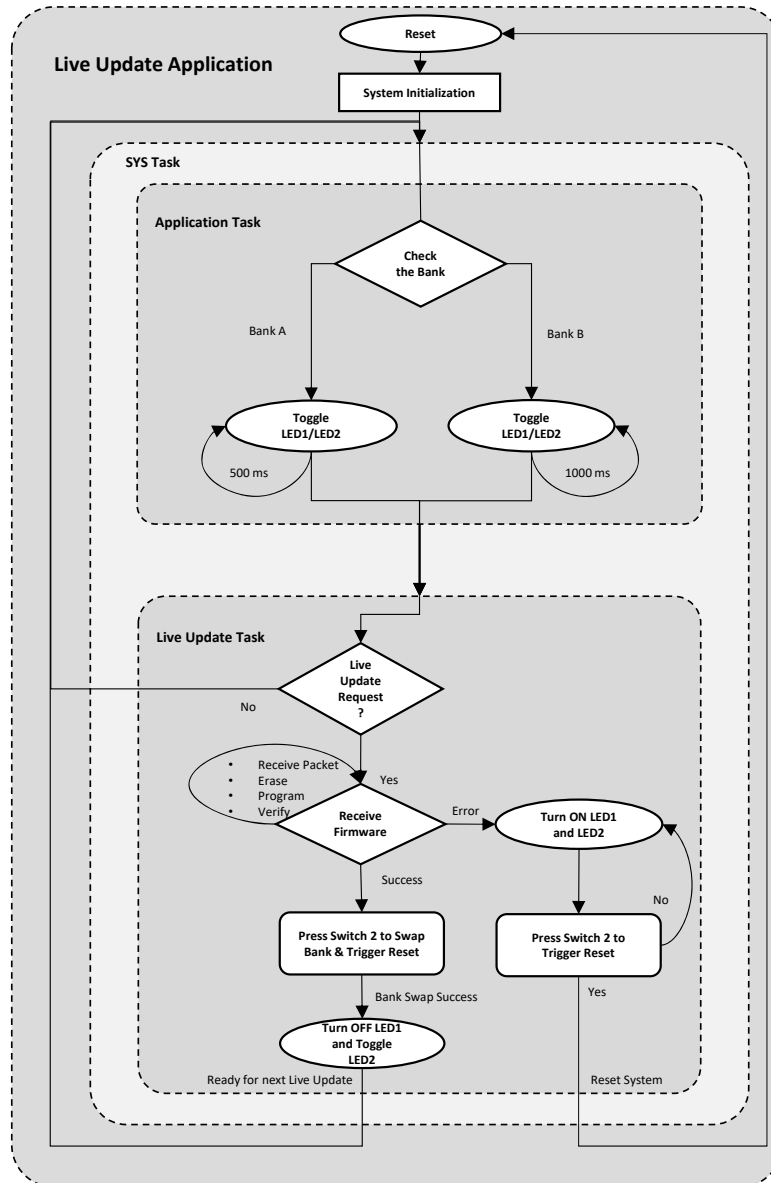
实时更新系统级执行流程

通过 MPLAB Harmony v3 UART 故障保护自举程序成功编程实时更新应用程序后，自举程序将加载该实时更新应用程序，然后 CPU 将开始执行该程序。

实时更新应用程序从主机接收新的应用程序映像。在实时更新应用程序成功执行升级后，将切换到空闲状态并等待下一个实时更新请求。按下开关 SW2 后，将加载新更新的固件。按下开关会导致实时更新应用程序交换存储区，从而使非活动存储区处于活动状态，使活动存储区处于非活动状态。存储区 A 处于非活动状态而存储区 B 处于活动状态。另外，将发出复位命令以运行升级后的应用程序。

下面的流程图说明了实时更新应用程序的执行流程。

图 2-9. 实时更新应用程序执行流程



实时更新应用程序的执行顺序如下：

1. 器件复位后，自举程序将从双存储区闪存的活动存储区中加载实时更新应用程序。
2. 实时更新应用程序执行系统初始化并调用应用程序任务，然后调用实时更新任务。

-
3. 应用程序任务检查正在执行实时更新应用程序的活动存储区，并每 500 ms 点亮一次 LED1（从存储区 A 运行时）或每 1000 ms 点亮一次 LED2（从存储区 B 运行时）。
 4. 实时更新任务仅在固件升级请求时才会运行。否则，该任务将处于空闲状态，应用程序任务将继续运行。
 5. 如果应用程序收到固件升级请求，则实时更新任务会执行以下操作：
 - 从主机 PC 接收到 **Unlock** 命令后，开始接收数据包。
 - 从主机 PC 接收到 **Data** 命令后，接收固件映像并在非活动存储区中进行编程。
 - 从主机 PC 接收到 **Verify** 命令后，验证接收到的固件映像，并等待用户按下开关 2 来交换存储区并执行器件复位以运行升级后的应用程序。
 6. 如果在步骤 5 中发生任何错误，或者主机应用程序请求更新活动存储区而地址位于活动存储区串行扇区，则实时更新任务将发送错误响应，中止编程操作，并点亮 LED1 和 LED2，直到用户按下开关 2 来触发系统复位以再次重新启动实时更新。
 7. 如果实时更新成功，则将熄灭 LED1 并点亮 LED2，直到用户按下开关 2 来交换存储区并触发系统复位以运行升级后的应用程序。

3. 配置

实时更新应用程序包含以下应用程序：

- UART 故障保护自举程序 (uart_fail_safe_bootloader_sam_e54_xpro)。其中包含用于首次编程实时更新应用程序的自举程序。
- 实时更新应用程序 (uart_same54_uart_live_update)。其中包含用户应用程序任务和实时更新功能的实现。

注： 可以使用以下路径在 MPLAB Harmony v3 自举程序 UART 应用程序[资源库](#)中找到 MPLAB Harmony v3 自举程序项目：<Harmony folder>/bootloader_apps_uart/apps/uart_fail_safe_bootloader/bootloader。

以下路径提供了有关自举程序和 UART 故障保护自举程序配置的更多信息：<Harmony folder>/bootloader/doc/help_bootloader.chm 和 <Harmony folder>/bootloader_apps_uart/docs/index.html。

3.1 自举程序链接描述文件

自举程序库使用通过 MHC 生成的自定义链接描述文件 (bt1.ld)。MHC 生成以下内容：

- 指定的自举程序大小。
- 代码和只读数据的起始地址，ROM（只读存储器）段地址。
- 读写数据的起始地址，RAM（随机访问存储器）段地址，如下图所示。

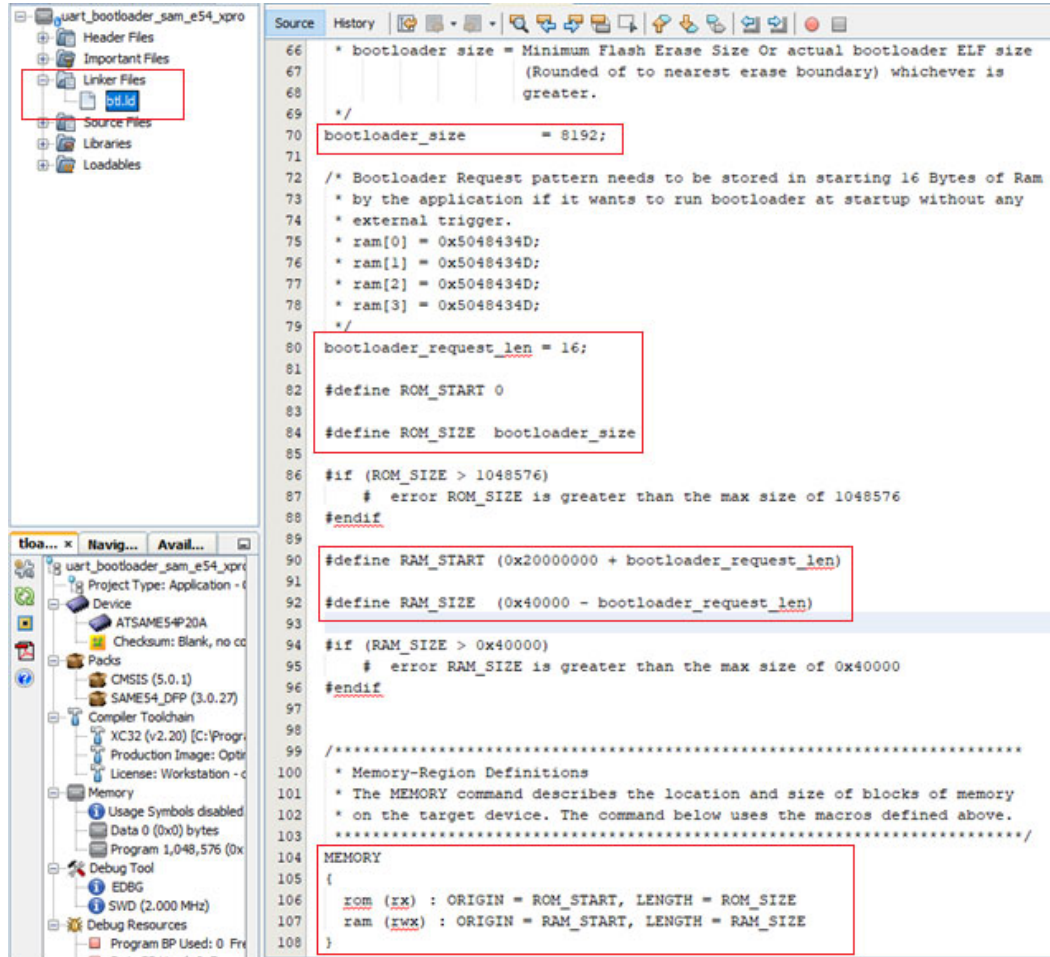
链接描述文件中填充的值基于 MHC 配置（自举程序配置）的自举程序组件。

配置链接描述文件，使自举程序从 RAM 运行，以便闪存写入操作和下一个数据块的接收操作同时进行。

如果应用程序想要在启动时不借助任何外部触发来运行自举程序，必须将自举程序请求模式存储在 RAM 开头的 16 个字节中，如下图所示。

即使在 -O1 优化中自举程序的大小为 1672 字节，SAM E54 的自举程序大小也会四舍五入为最接近的擦除单元大小（8192 字节）。这样便可在自举程序上进一步添加其他功能并避免应用程序与自举程序重叠。

图 3-1. 自举程序自定义链接描述文件



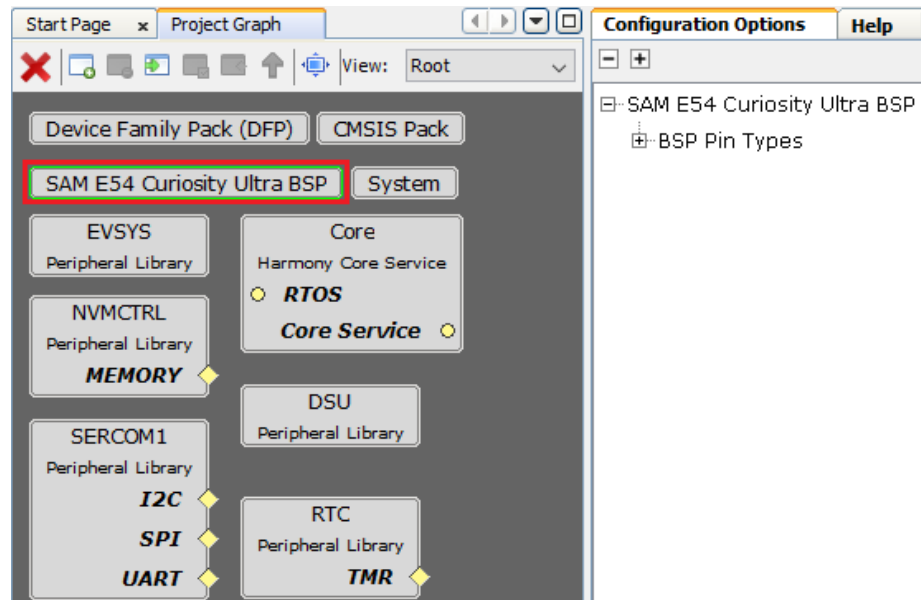
注： 确保用户应用程序的存储区不得与预留给自举程序的存储区重叠。

3.2 MHC 配置

按照以下步骤配置 MHC，这是实现实时更新应用程序的必要操作。

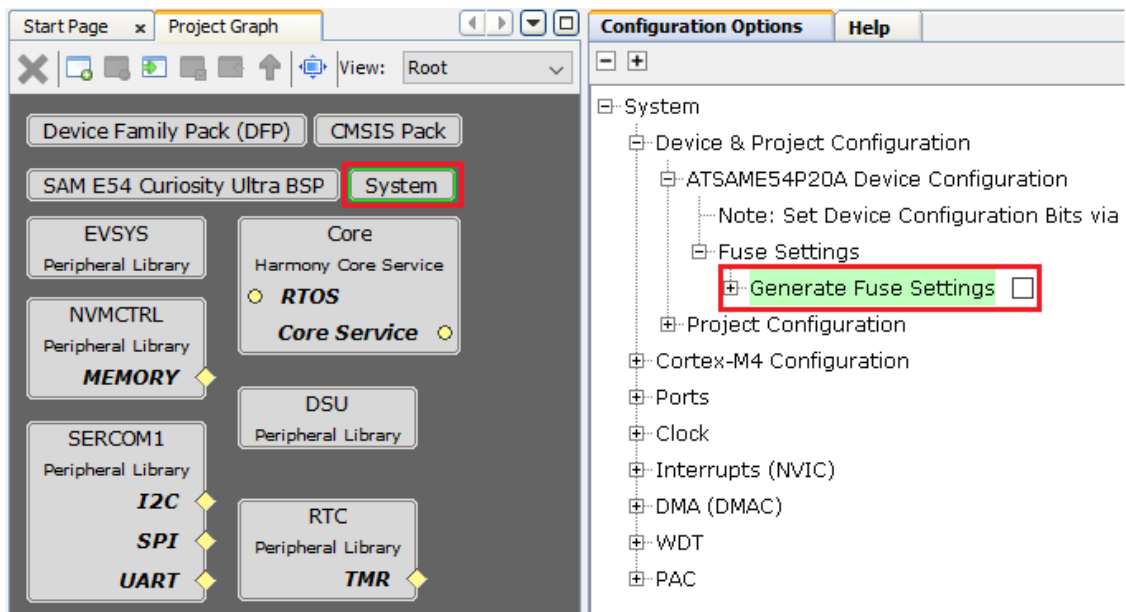
1. 将 SAM E54 Curiosity Ultra BSP 添加到开关和 LED 配置中。

图 3-2. 实时更新应用程序 BSP 配置



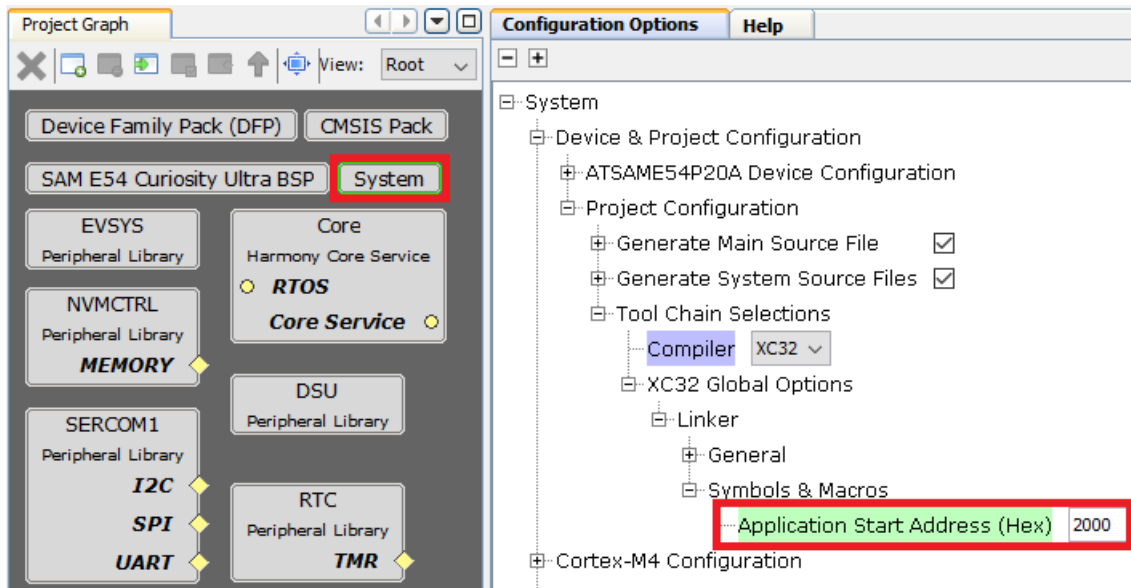
2. 从自定义链接描述文件中移除器件熔丝配置，因为该配置将由自举程序项目更新。

图 3-3. 实时更新应用程序禁止熔丝设置



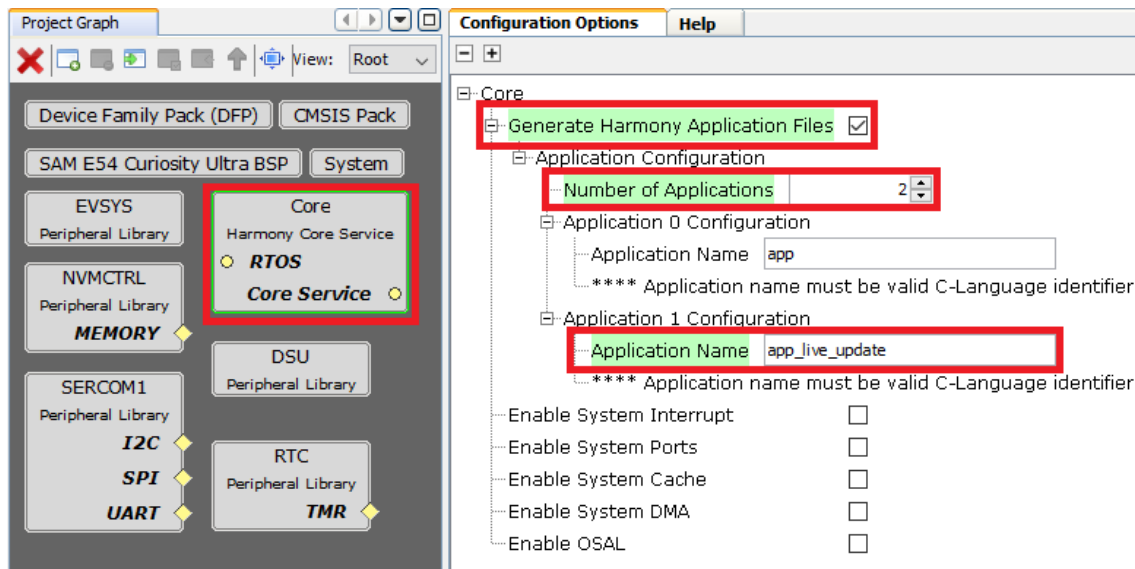
3. 配置与自举程序项目中的应用程序起始地址相匹配的实时更新应用程序起始地址。

图 3-4. 实时更新应用程序起始地址配置



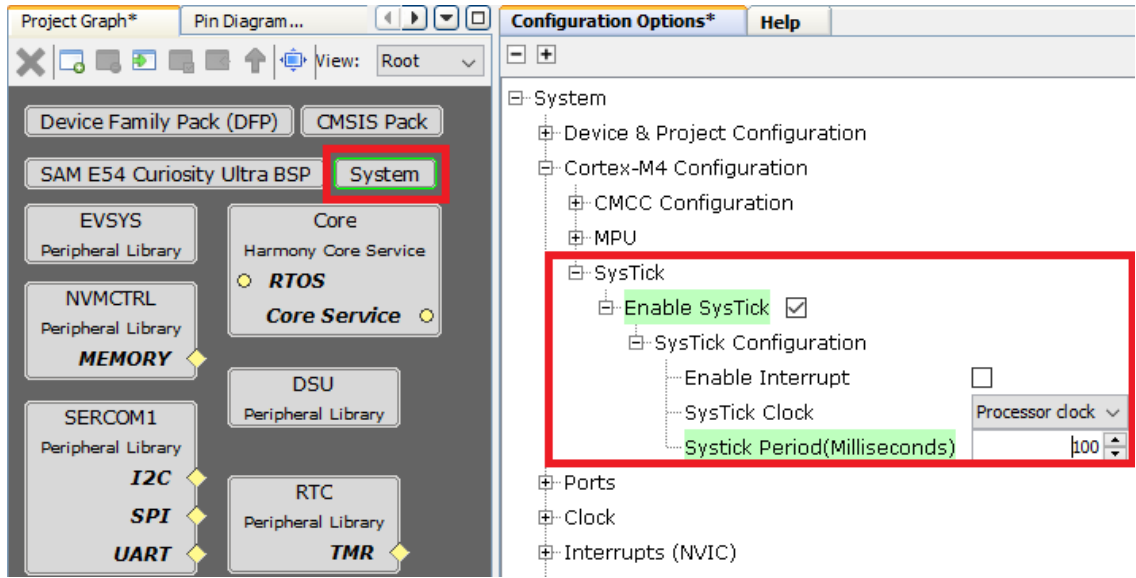
4. 添加 MPLAB Harmony Core 组件并配置应用程序和实时更新任务。这将生成源文件，例如可用于添加任何子任务的 `tasks.c`，以及用于添加用户配置的 `user.h`。

图 3-5. 实时更新应用程序 Harmony Core 配置



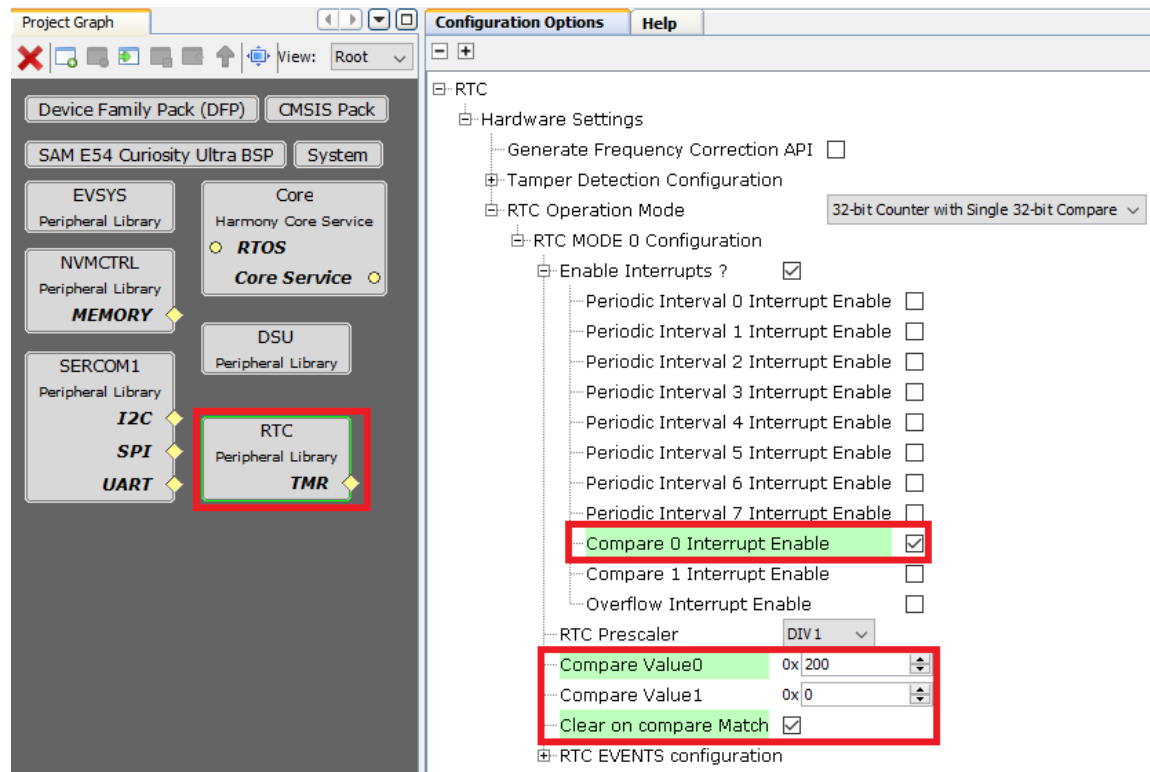
5. 添加用于 CRC 计算的 DSU 模块。
6. 为 SysTick 配置 100 ms 的周期（毫秒级），以便在 100 ms 周转时间内从主机接收数据包。

图 3-6. 实时更新应用程序 SysTick 配置



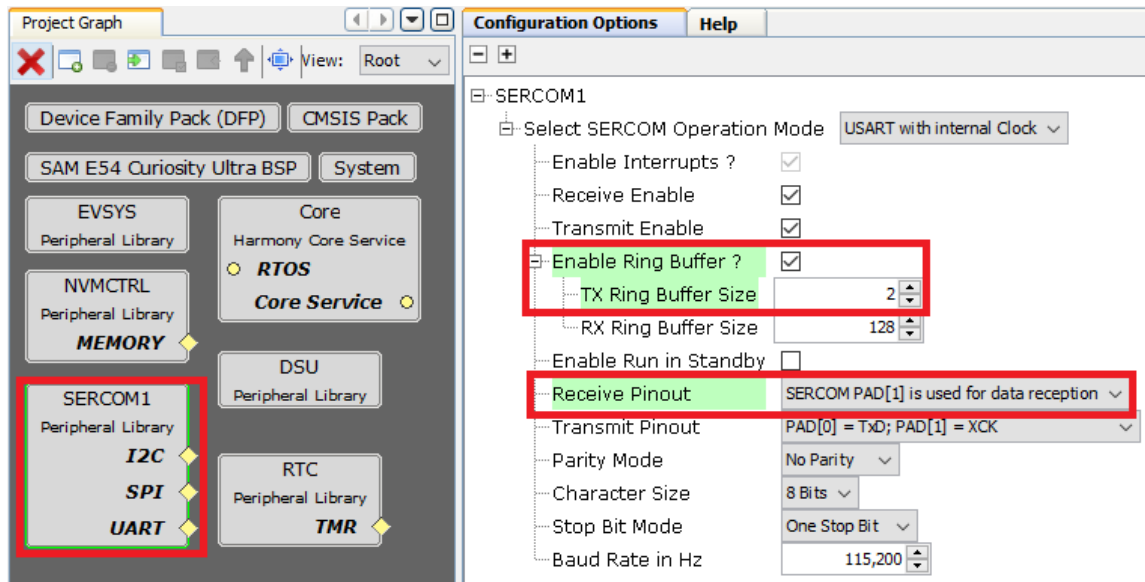
7. 添加 RTC 模块以按照不同的速率翻转 LED。

图 3-7. 实时更新应用程序 RTC 配置



8. 添加 UART 模块并使能环形缓冲区，以便在运行时从主机 PC 接收实时更新应用程序。

图 3-8. 实时更新应用程序 UART 配置



9. 通过 *MHC > Tools > Pin Configurations* (MHC > 工具 > 引脚配置) 启动引脚配置插件。
10. 配置 UART 引脚以从主机 PC 接收数据。

图 3-9. 实时更新应用程序 UART 引脚配置

Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull Down	Drive Strength
111	PC27	SERCOM1_PAD0	SERCOM1_P...	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
112	PC28	SERCOM1_PAD1	SERCOM1_P...	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL

注：保留 SERCOM1 用于实时更新任务。不应将其用于其他用途。

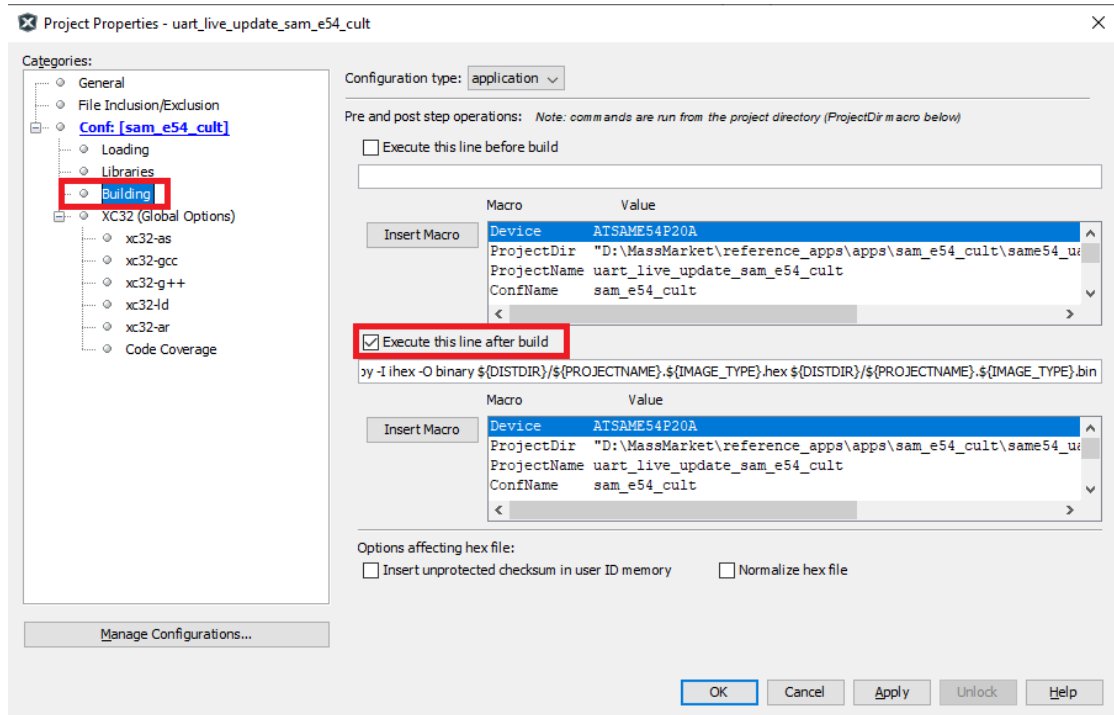
3.3 项目设置

- **Preprocessor macro definitions (预处理器宏定义) :**
 - ROM_ORIGIN 和 ROM_LENGTH 为 XC32 链接器变量，将由 MHC 中提供的值改写。请参见[实时更新应用程序起始地址配置](#)。
 - 将使用 MHC 中提供的应用程序起始地址的值自动填充链接描述文件中的应用程序起始地址值。请参见[自举程序自定义链接描述文件](#)。（重新生成后）
- **Execute command line after Build (在编译后执行命令行) :**
 - 以下编译选项可用于在编译完成后自动从 .hex 文件生成二进制文件。

```

${MP_CC_DIR}/xc32-objcopy -I ihex -O binary ${DISTDIR}/${PROJECTNAME}.${
IMAGE_TYPE}.hex
${DISTDIR}/${PROJECTNAME}.${IMAGE_TYPE}.bin
```

图 3-10. 实时更新应用程序二进制文件生成设置



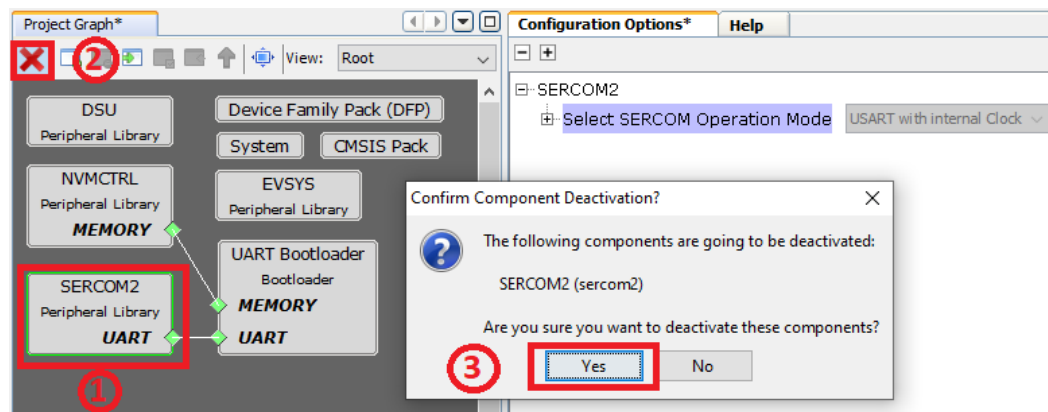
4. 运行应用程序

4.1 运行自举程序应用程序

按照以下步骤使用 MPLAB Harmony v3 UART 故障保护自举程序对实时更新应用程序进行编程：

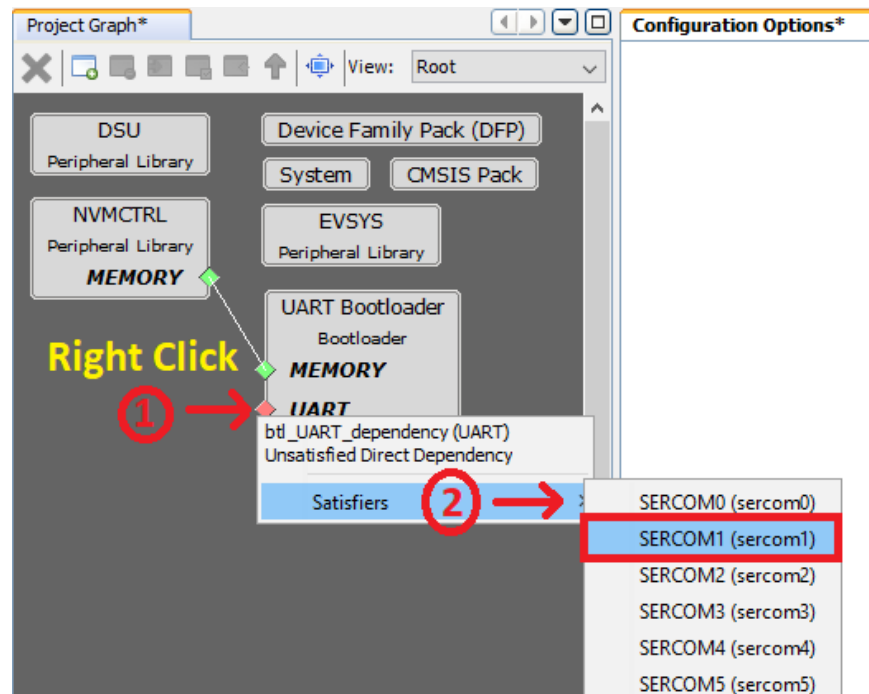
1. 下载 MPLAB Harmony v3 [自举程序包](#)。
2. 下载 MPLAB Harmony v3 [UART 自举程序应用程序包](#)。
3. 将 micro-USB 线缆连接到 SAM E54 Curiosity Ultra 开发板的调试端口。
4. 使用 MPLAB X IDE 打开 UART 故障保护自举程序（路径：<Harmony folder>/bootloader_apps_uart/apps/uart_fail_safe_bootloader/bootloader/firmware/sam_e54_xpro.X）。
注： 必须将项目重新配置为可以在 SAM E54 Curiosity Ultra 开发板上工作。
5. 启动 MHC 并按照以下步骤为 SAM E54 Curiosity Ultra 开发板重新配置项目。
 - 5.1. 删除 SERCOM2。

图 4-1. 在 MHC 中重新配置自举程序——删除 SERCOM2 外设



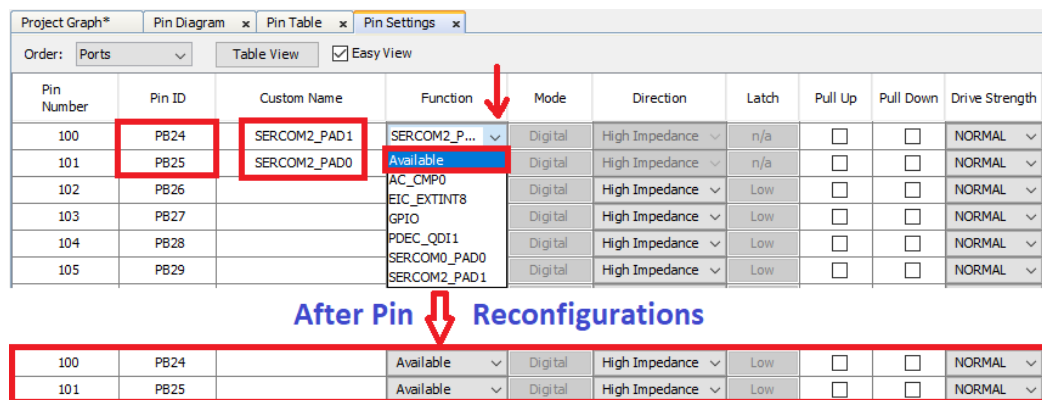
- 5.2. 将 SERCOM1 添加到 Project Graph（项目图）。

图 4-2. 在 MHC 中重新配置自举程序——添加 SERCOM1 外设



- 5.3. 通过 *MHC > Tools > Pin Configurations* (MHC > 工具 > 引脚配置) 启动引脚配置插件。
- 5.4. 删除 SERCOM2 引脚配置。

图 4-3. 在 MHC 中重新配置自举程序——删除 SERCOM2 引脚配置



- 5.5. 配置 SERCOM1 引脚。

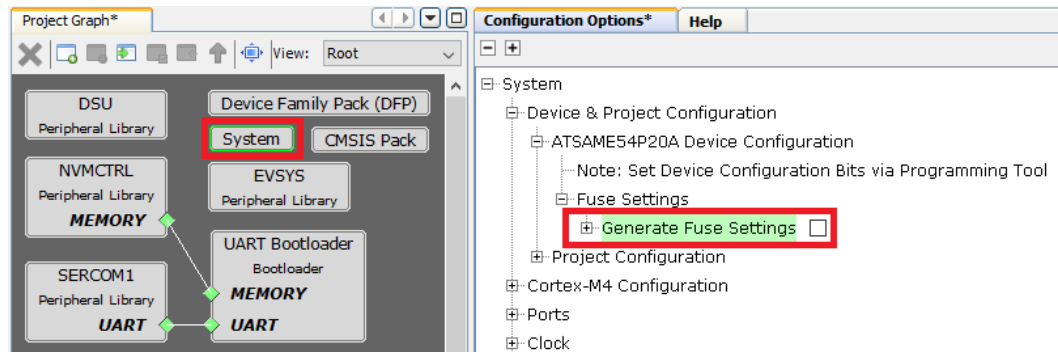
图 4-4. 在 MHC 中重新配置自举程序——配置 SERCOM1 引脚

Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull Down	Drive Strength
111	PC27	SERCOM1_PAD0	SERCOM1_P...	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
112	PC28	SERCOM1_PAD1	SERCOM1_P...	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL

注:

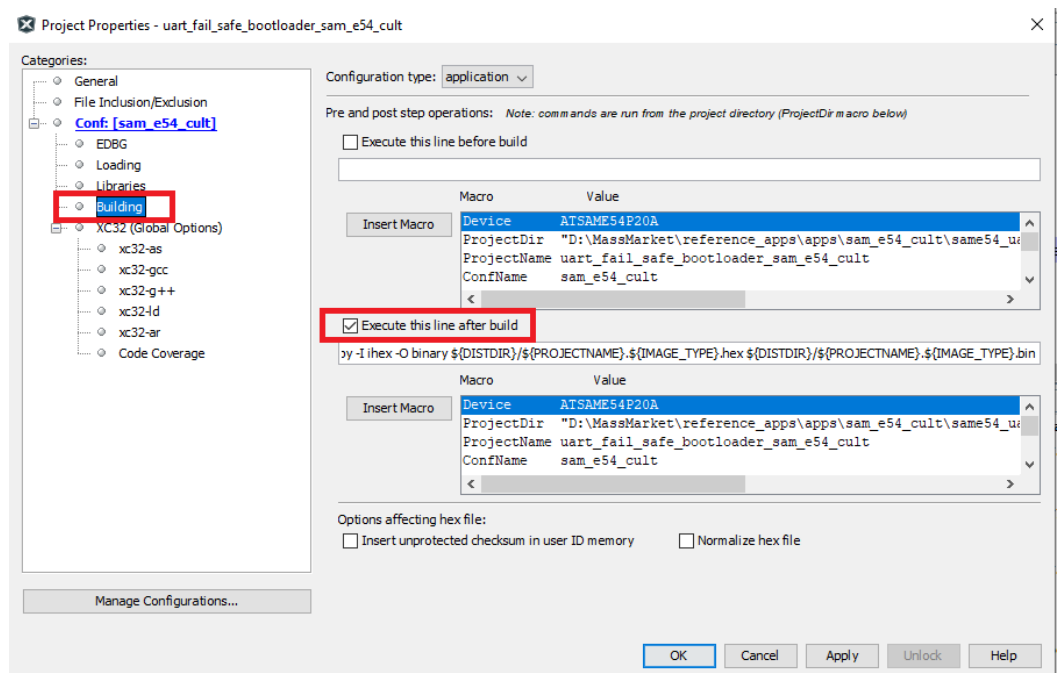
- SAM E54 Curiosity Ultra 开发板将 SERCOM1 外设用于 EDBG。因此，必须对 SERCOM 外设及其引脚进行重新配置。
 - 项目将被重新编译，因此请勿关闭 MHC。
6. 重新生成项目。使用 MPLAB X IDE 编译并编程 UART 故障保护自举程序。
 7. 使用 MPLAB X IDE 按照以下步骤重新编译 UART 故障保护自举程序。此步骤会创建自举程序二进制映像，该映像将与实时更新应用程序合并以创建单个二进制映像，如步骤 9 所示。
 - 7.1. 从自定义链接描述文件中删除器件熔丝配置，因为该配置将由 UART 故障保护自举程序项目更新。

图 4-5. 在 MHC 中重新配置自举程序——删除熔丝配置



- 7.2. 在 Project Properties (项目属性) 窗口中，选择 Building (编译)，然后在 MPLAB X IDE 中勾选 Execute this line after build (在编译后执行该代码行)，如下图所示。

图 4-6. 在 MHC 中重新配置自举程序——使能编译选项



8. 使用 MPLAB X IDE 再次编译 UART 故障保护自举程序应用程序，但不进行编程。
9. 使用 MPLAB X IDE 下载并编译实时更新应用程序 (路径: `<Live Update application folder>/firmware/sam_e54_cult.X`)，但不进行编程。

- 从命令提示符处运行 `btl_app_merge_bin.py` 描述文件以合并生成的自举程序二进制文件和实时更新应用程序二进制文件。

```
python <Harmony folder>/bootloader/tools_archive/btl_app_merge_bin.py -o 0x2000 -b
<Harmony
  folder>/bootloader_apps_uart/apps/uart_fail_safe_bootloader/bootloader/firmware/
sam_e54_xpro.X/dist/<Config_Name>/production/<Project_Name>.production.bin
-a <Live Update application
  folder>/firmware/sam_e54_cult.X/dist/<Config_Name>/production/<Project
  Name>.production.bin
```

- 命令提示符处必定会显示以下输出：

图 4-7. 自举程序二进制文件合并结果

```
('\r\n#### Merged Bootloader and Application binaries to ', 'btl_app_merged.bin', '####')
```

- 从命令提示符处运行 `btl_host.py`，以便将合并的二进制文件编程到双存储区闪存的非活动存储区中。合并的二进制文件 `btl_app_merged.bin` 将在调用 `btl_app_merge_bin.py` 的路径中生成。

```
python <Harmony folder>/bootloader/tools_archive/btl_host.py -v -s -i <COM PORT> -d
same5x -a <Address> -f btl_app_merged.bin
```

注：

- 有关自举程序主机描述文件的更多信息，请参见 `<Harmony folder>/bootloader/doc/help_bootloader.chm` 和 `<Harmony folder>/bootloader_apps_uart/docs/index.html` 中提供的主机描述文件设置帮助。

- 下图显示了成功编程应用程序二进制文件的结果。

图 4-8. 应用程序二进制文件编程的自举程序主机描述文件日志

```
Unlocking
Uploading 2 blocks at address 524288 (0x80000)
... block 1 of 2
... block 2 of 2
Verification
... success
Swapping Bank And Rebooting
Reboot Done
```

4.2 运行实时更新应用程序

- 按照 [运行自举程序应用程序](#) 的步骤执行并下载 [实时更新应用程序](#)（如果尚未完成）。
- 如果上一步骤成功，则 SAM E54 Curiosity Ultra 开发板上的 LED1 必定会开始闪烁。
- 从位于 `<Live Update application folder>/scripts` 的命令提示符处运行主机描述文件 `live_update.py`，以便在从活动分区运行当前版本的应用程序的同时将新版本的固件编程到非活动分区。

```
python <Live Update application folder>/scripts/live_update.py -v -i <COM PORT> -d
same5x -a <Address> -f btl_app_merged.bin.
```

- 合并的二进制文件 `btl_app_merged.bin` 将在调用 `btl_app_merge_bin.py` 的路径中生成。
- 下图给出了实时更新描述文件帮助。

图 4-9. 实时更新应用程序主机描述文件帮助窗口

```
Usage: live_update.py [options]

Options:
  -h, --help            show this help message and exit
  -v, --verbose         enable verbose output
  -r BAUD, --baud=BAUD  UART baudrate
  -t, --tune            auto-tune UART baudrate
  -i PATH, --interface=PATH
                        communication interface
  -f FILE, --file=FILE  binary file to program
  -a ADDR, --address=ADDR
                        destination address
  -p SectSize, --sectorSize=SectSize
                        Device Sector Size in Bytes
  -b, --boot            enable write to the bootloader area
  -d DEV, --device=DEV  target device (samc2x/samd1x/samd2x/samd5x/samda1/same
                        7x/same5x/samg5x/saml2x/samha1/pic32mk/pic32mx/pic32mz
                        )
```

```
Command: python <python script> -v -s -i <COM PORT> -d <Device Name> -a
<Address> -f <live_update_application_image>
<python script>: live_update.py
<COM PORT>: Serial Communication Port
<Device Name>: same5x
<Address>: Application start address 0x9D100000
Example: python <Live update application folder \scripts\ live_update.py -v -s -
i COM6 -d same5x -a 0x80000 -f sam_e54_cult.X.production.bin
Note: Running the help command provides a brief overview of options available as
shown below.
Command: python <python script> --help
<python script>: live_update.py
```

注：有关自举程序主机描述文件的更多信息，请参见<Harmony folder>/bootloader/doc/help_bootloader.chm 中提供的主机描述文件设置帮助。

- 下图给出了实时更新固件升级输出。

图 4-10. 实时更新应用程序固件升级输出

```
Live Update request started.

Unlocking
Uploading 2 blocks at address 524288 (0x80000)

... block 1 of 2
... block 2 of 2
Verification
... success

Live Update is success.

Press Switch (SW2) to reset the device for the programmed application firmware to run.
```

- LED1 停止闪烁，LED2 开始闪烁，指示应用程序编程成功。
 - LED1 停止闪烁，LED2 每 500 ms 点亮一次，指示应用程序从存储区 A 运行。
 - LED1 停止闪烁，LED2 每 1000 ms 点亮一次，指示应用程序从存储区 B 运行。
- 按下开关 SW2 交换存储区并复位器件，以运行经过编程的应用程序固件。
 - LED2 停止闪烁，LED1 每 500 ms 点亮一次，指示应用程序从存储区 A 运行。
 - LED2 停止闪烁，LED1 每 1000 ms 点亮一次，指示应用程序从存储区 B 运行。
- 如果在实时更新期间发生任何错误，则 LED1 和 LED2 都将点亮。使用以下步骤从错误中恢复。
 - 长按开关 SW2 以复位系统，然后重复步骤 4 对实时更新应用程序进行编程。

- 或
- 7.2. 对器件进行复位或掉电再上电，然后从步骤 4 开始重复，对实时更新应用程序进行编程。

5. 结论

本文档介绍了实时更新应用程序的实现。实时更新应用程序可以增强应用程序功能，能够在器件继续运行现有版本应用程序软件的同时更加轻松地引入新功能。它允许器件在现场工作更长的时间，并有助于降低产品的维护成本。它还可以安排定期更新与按需求更新，从而帮助简化产品的部署。

本文档可用作参考资料，帮助用户在应用程序中实现实时更新功能。

实时更新任务在中断模式下运行，以便在运行时更新固件；因此，用户必须确保只要有固件更新请求，应用程序任务就不会进行持续轮询。否则，在从主机 PC 接收固件时将导致数据丢失或超时。

6. 参考文献

- MPLAB Harmony v3:
www.microchip.com/mplab/mplab-harmony
- 在 SAM E54 Curiosity Ultra 开发板上实现实时更新应用程序:
microchiptech.github.io/MPLAB-Harmony-Reference-Apps/apps/sam_e54_cult/same54_uart_live_update/readme.html
- 基于 SAM E54 Curiosity Ultra 开发板 +使用 Legato Graphics 的 maXTouch® Curiosity Pro 板的 SD 卡 USB 音频播放器:
microchiptech.github.io/MPLAB-Harmony-Reference-Apps/apps/sam_e54_cult/same54_sdcard_usb_audio_player/readme.html
- MPLAB Harmony v3 自举程序帮助:
<Harmony 框架下载文件夹>\bootloader\doc\help_bootloader.chm
- MPLAB Harmony v3 Core 帮助:
microchip-mplab-harmony.github.io/core/frames.html?frmname=topic&frmfile=index.html
- MPLAB Harmony v3 自举程序应用程序:
<Harmony 框架下载文件夹>\bootloader\apps
- SAM D5x/E5x MCU 上的 MPLAB Harmony v3 外设库入门:
microchipdeveloper.com/harmony3:same54-getting-started-training-module
- 使用 MPLAB Harmony v3 实现 SAM E54 单片机 (MCU) 上的双存储区自举程序:
ww1.microchip.com/downloads/en/Appnotes/Dual-Bank-Bootloader-on-SAME54-Microcontroller-Using-MPLAB-Harmonyv3-DS00003508A.pdf
- MPLAB Harmony v3 参考应用程序:
github.com/MicrochipTech/MPLAB-Harmony-Reference-Apps
- MPLAB Harmony v3 入门文章和其他文档:
www.microchip.com/mplab/mplab-harmony/mplab-harmony-articles-and-documentation
- MPLAB Harmony v3 配置器概述:
microchipdeveloper.com/harmony3:mhc-overview
- 使用 MPLAB Harmony v3 创建第一个电机控制应用程序:
microchipdeveloper.com/harmony3:motor-control-getting-started-training-module
- TB3253——如何通过向现有的 MPLAB® Harmony v3 项目添加新的 PLIB、驱动程序或中间件来构建应用程序:
<http://www.microchip.com.cn/newcommunity/Uploads/202103/6064070d67eaa.pdf>
- MPLAB Harmony v3 SD 卡音频播放器/读卡器教程:
microchipdeveloper.com/harmony3:audio-player
- PIC32MZ 和 SAM MCU 图形快速入门应用程序:
github.com/Microchip-MPLAB-Harmony/gfx/wiki/Application-QuickStart
- MPLAB Harmony v3 USB 协议栈:
microchip-mplab-harmony.github.io/usb/frames.html?frmname=topic&frmfile=index.html
- 创建第一个 USB 设备 CDC 单个应用程序:
github.com/Microchip-MPLAB-Harmony/usb/wiki/Create-your-first-usb-device-cdc-single-application
- 创建第一个 USB 主机 MSD 应用程序:
github.com/Microchip-MPLAB-Harmony/usb/wiki/Create-your-first-usb-host-msd-application
- MPLAB Harmony v3 TCP/IP 帮助:
microchip-mplab-harmony.github.io/net/frames.html?frmname=topic&frmfile=index.html
- 创建第一个 TCP/IP 应用程序:
github.com/Microchip-MPLAB-Harmony/net/wiki/Create-your-first-tcpip-application
- 面向 MPLAB Harmony v2 用户的 MPLAB Harmony v3 应用程序开发指南
ww1.microchip.com/downloads/en/Appnotes/MPLAB_Harmonyv3_Application_Development_%20Guide_for_%20MPLAB_Harmonyv2_Users_DS00003388A.pdf

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 器件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：在正常使用的情况下，Microchip 系列产品非常安全。
- 目前，仍存在着用恶意、甚至是非法的方法来试图破坏代码保护功能的行为。我们确信，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这种试图破坏代码保护功能的行为极可能侵犯 Microchip 的知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- 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。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案 (Digital Millennium Copyright Act)》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

法律声明

提供本文档的中文版本仅为为了便于理解。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。

本出版物中提供的信息仅仅是为方便您使用 Microchip 产品或使用这些产品来进行设计。本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。

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、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、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。

AgileSwitch、APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、IntelliMOS、Liberio、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus 徽标、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、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、IdealBridge、In-Circuit Serial Programming、ICSP、INICnet、Intelligent Paralleling、Inter-Chip Connectivity、JitterBlocker、maxCrypto、maxView、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、RTAX、RTG4、SAM-ICE、Serial Quad I/O、simpleMAP、SimpliPHY、SmartBuffer、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 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 在其他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, 版权所有。

ISBN: 978-1-5224-8353-3

AMBA、Arm、Arm7、Arm7TDMI、Arm9、Arm11、Artisan、big.LITTLE、Cordio、CoreLink、CoreSight、Cortex、DesignStart、DynamIQ、Jazelle、Keil、Mali、Mbed、Mbed Enabled、NEON、POP、RealView、SecurCore、Socrates、Thumb、TrustZone、ULINK、ULINK2、ULINK-ME、ULINK-PLUS、ULINKpro、µVision 和 Versatile 是 Arm Limited（或其子公司）在美国和/或其他国家/地区的商标或注册商标。

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。

全球销售及服务中心

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 - 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
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 诺布尔斯特维尔, 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			