
利用 MPLAB® Harmony v3 实时更新 PIC32MZ MCU 上的应用程序

简介

PIC32MZ 单片机 (MCU) 上的双区闪存支持应用程序实现实时更新功能。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。

以下步骤给出了实时更新应用程序的流程：

1. 必须使用自举程序将实时更新应用程序编程到特定的存储单元。
2. 支持实时更新的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。
3. 实现实时更新功能的一段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。
4. PIC32MZ MCU 将存储区分为两部分：一部分用于自举程序，另一部分用于应用程序代码。
5. 自举程序负责确定用户是打算更新固件还是运行现有固件。
6. 除了执行常规的应用程序特定功能外，位于用户应用程序空间中的应用程序还能够在运行时更新固件，而无需触发自举程序 (实时更新功能)。

本文档介绍了如何使用双存储区闪存存在 PIC32MZ MCU 上实现支持实时更新的应用程序。

目录

简介.....	1
1. 硬件和软件要求.....	3
1.1. 带 FPU 的 PIC32MZ (EF) 嵌入式连接入门工具包.....	3
1.2. MPLAB X 集成开发环境 (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.....	4
2.2. 命令说明.....	5
2.3. 通信任务.....	7
2.4. 命令处理器任务.....	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. 链接描述文件.....	13
3.2. MHC 配置.....	14
3.3. 项目设置.....	16
4. 运行应用程序.....	18
4.1. 运行自举程序应用程序.....	18
4.2. 运行实时更新应用程序.....	18
5. 结论.....	20
6. 参考资料.....	21
Microchip 网站.....	22
产品变更通知服务.....	22
客户支持.....	22
Microchip 器件代码保护功能.....	22
法律声明.....	22
商标.....	23
质量管理体系.....	23
全球销售及服务网点.....	24

1. 硬件和软件要求

1.1 带 FPU 的 PIC32MZ (EF) 嵌入式连接入门工具包

带 FPU 的 PIC32MZ (EF) 嵌入式连接入门工具包用于评估 PIC32MZ 单片机，它基于运行频率为 200 MHz 的 MIPS® 内核。高性能 PIC32MZ EF 系列单片机具备行业领先的连接性和外设选项，可让嵌入式设计人员快速构建复杂的应用。

PIC32MZ EF MCU 的主要特性如下：

- 200 MHz/330 DMIPS 性能
- 最高 2 MB 的双区实时更新闪存和 512 KB RAM
- 出色的连接性选项（高速 USB、CAN 和 10/100 以太网）
- 集成的双精度 FPU 可提高流程密集型应用程序的性能
- 可选的全功能硬件加密加速器
- 丰富的外设集

可从 [Microchip 直销网站](#) 上获取带 FPU 的 PIC32MZ (EF) 嵌入式连接入门工具包。

1.2 MPLAB X 集成开发环境 (IDE) 和 XC 编译器

MPLAB® X 集成开发环境 (Integrated Development Environment, 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。本文档介绍了 MPLAB X IDE 版本 5.40。
- MPLAB XC 编译器可从 [Microchip 网站](#) 获取。本文档介绍了 MPLAB XC32 版本 2.41。

1.3 MPLAB Harmony v3

MPLAB Harmony v3 是一款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，可将资源专注于创建 32 位 PIC® 和 SAM 器件应用程序的开发，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开发工具，带有图形用户界面 (Graphical User Interface, GUI)，可简化器件设置、库选择、配置和应用程序开发。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。这些资源库可从 [GitHub](#) 下载：

- [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)
- [UART 自举程序应用程序](#)
或
- 使用 [MPLAB Harmony 3 框架下载器](#) 下载上述资源库。

1.4 Python

本文档介绍了 Python 脚本 (Python v2.7)，此类脚本用于将应用程序二进制文件从主机 PC 发送到目标 (带 FPU 的 PIC32MZ (EF) 嵌入式连接入门工具包)。

2. 设计

PIC32MZ MCU 上的实时更新应用程序的设计使用双存储区闪存功能。这两个存储区分别命名为 **BANK1** 和 **BANK2**。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。

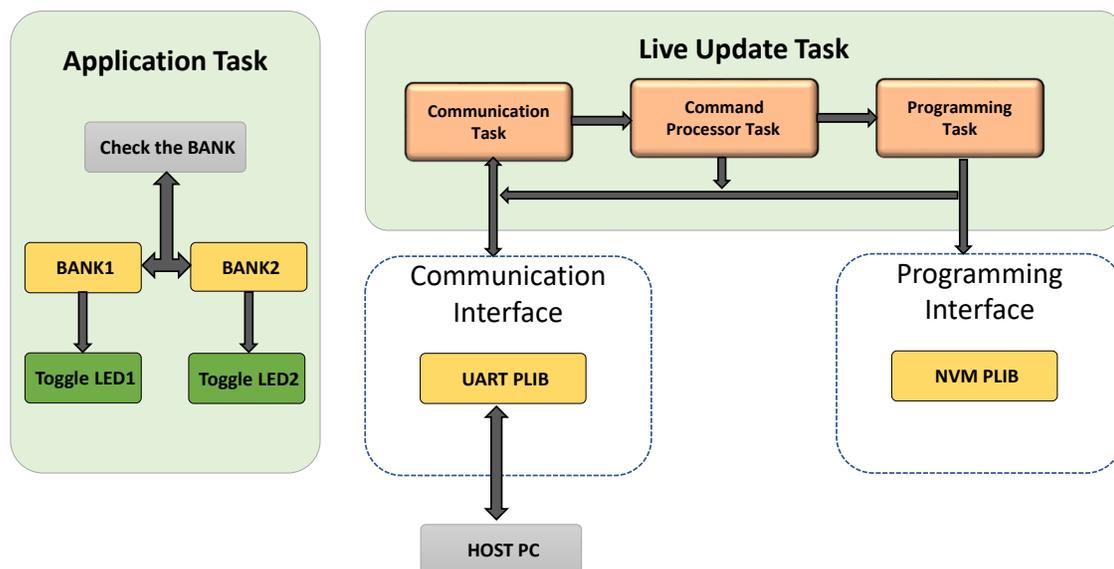
1. **应用程序任务**：代表最终用户应用程序。当前版本的最终用户应用程序将从双存储区闪存的活动存储区运行，而非活动存储区可用于使用最新版本的固件升级最终用户应用程序。本文档中讨论的应用程序任务本质上是极简的。它会标识具有最新固件版本的闪存存储区（活动存储区），并定期切换特定的 **LED**。例如，从点亮 **LED1** 开始，当有存储区交换（成功升级后活动存储区发生更改）时，**LED2** 点亮。同样，每次成功更新应用程序固件都会重复该过程。
2. **实时更新任务**：通过预定义的通信协议（在 [UART 实时更新协议](#) 中有介绍）与个人计算机主机应用程序通信并在存在来自 **HOST-PC** 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。

实时更新任务分为三个子任务：

- 通信任务
- 命令处理器任务
- 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。

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)
 - Bank Swap (0xA3)
- **数据**
 - 包含根据命令要处理的实际数据
 - 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): 处理命令期间出现错误
- 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

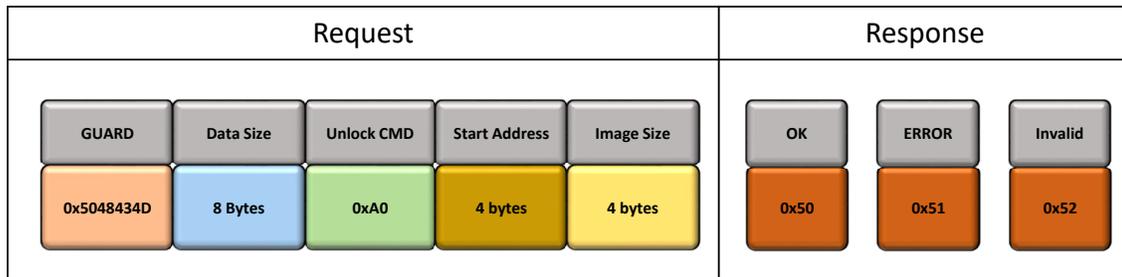
下面的命令说明部分给出了 UART 实时更新协议中使用的请求和响应代码的详细说明。

2.2 命令说明

Unlock 命令

Unlock 命令序列如下图所示，其中包含相应的响应。

图 2-3. UART 实时更新 Unlock 命令

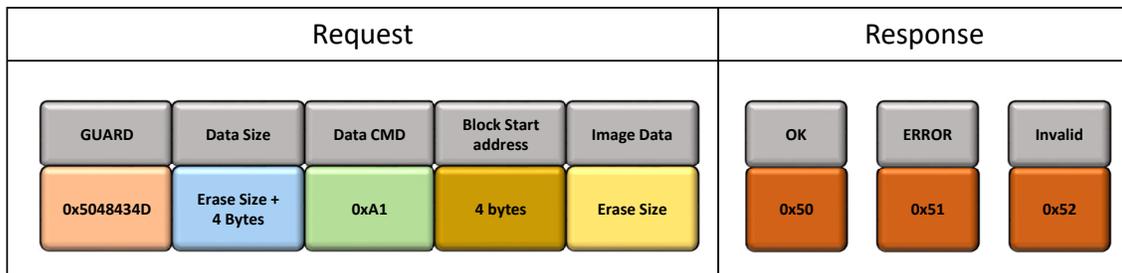


- Unlock 命令必须在第一个 Data 命令发出之前发出
 - 该命令用于计算应用程序的起始地址和结束地址
 - 该信息将用于验证发送的地址是否处于闪存范围内
 - 该命令将用于验证要编程的数据包随附的地址是否处于调用 Unlock 命令的区域内
- 要接收的数据字节数为 8 个字节（起始地址 + 映像大小）
- 起始地址：
 - 闪存的应用程序起始地址
 - 该地址取决于器件，应始终大于或等于自举程序的结束地址
 - 必须在擦除单元大小边界处对齐，该边界同样取决于器件
 - 要自行升级自举程序，该值必须设置为 0
- 映像大小必须以擦除单元字节为增量，具体也取决于器件

Data 命令

Data 命令序列如下图所示，其中包含相应的响应。

图 2-4. UART 实时更新 Data 命令

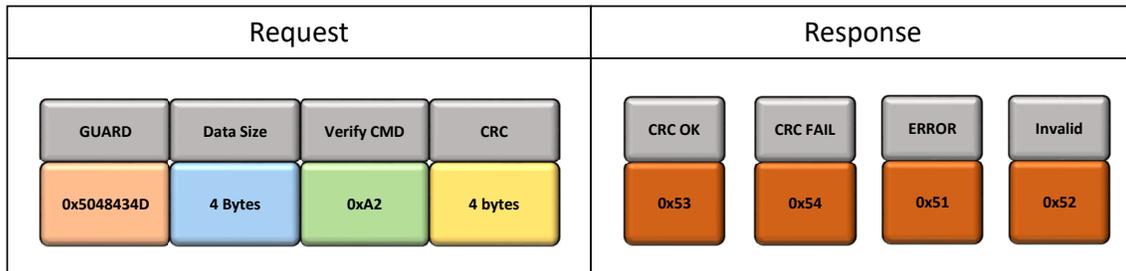


- Data 命令用于发送映像数据
- 数据大小等于块起始地址（4 字节）与擦除单元大小的总和，具体取决于器件
- 块起始地址必须位于之前通过 Unlock 命令解锁的区域内
- 如果尝试请求在解锁区域以外执行写操作，则将导致错误，提供的数据将被丢弃

Verify 命令

Verify 命令序列如下图所示，其中包含相应的响应。

图 2-5. UART 实时更新 Verify 命令

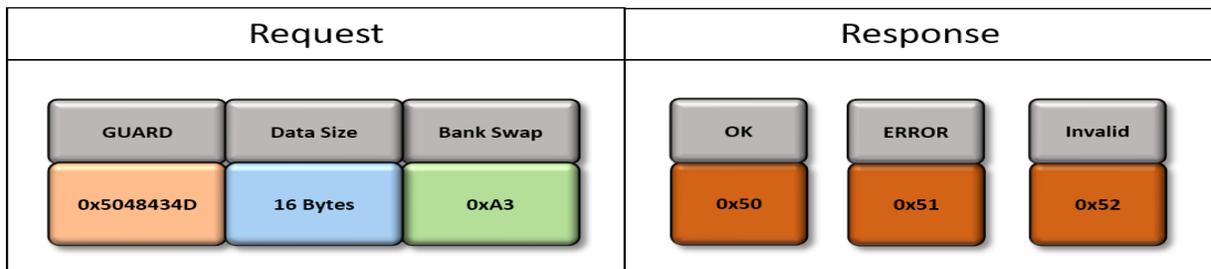


- Verify 命令用于验证发送和编程的映像数据
- 映像 CRC 是标准 IEEE CRC32，其多项式为 0xEDB88320
- 编程后，会根据从闪存中读取的值计算内部 CRC，因此该命令会进行全面验证
- 映像 CRC 计算会在之前解锁的区域上进行

Bank Swap 命令

Bank Swap 命令序列如下图所示，其中包含相应的响应。

图 2-6. UART 实时更新 Bank Swap 命令



- Bank Swap 命令用于将非活动存储区交换到活动存储区
- 器件复位运行在非活动存储区中编程的新应用程序

2.3 通信任务

此任务负责在中断模式下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 命令处理器任务

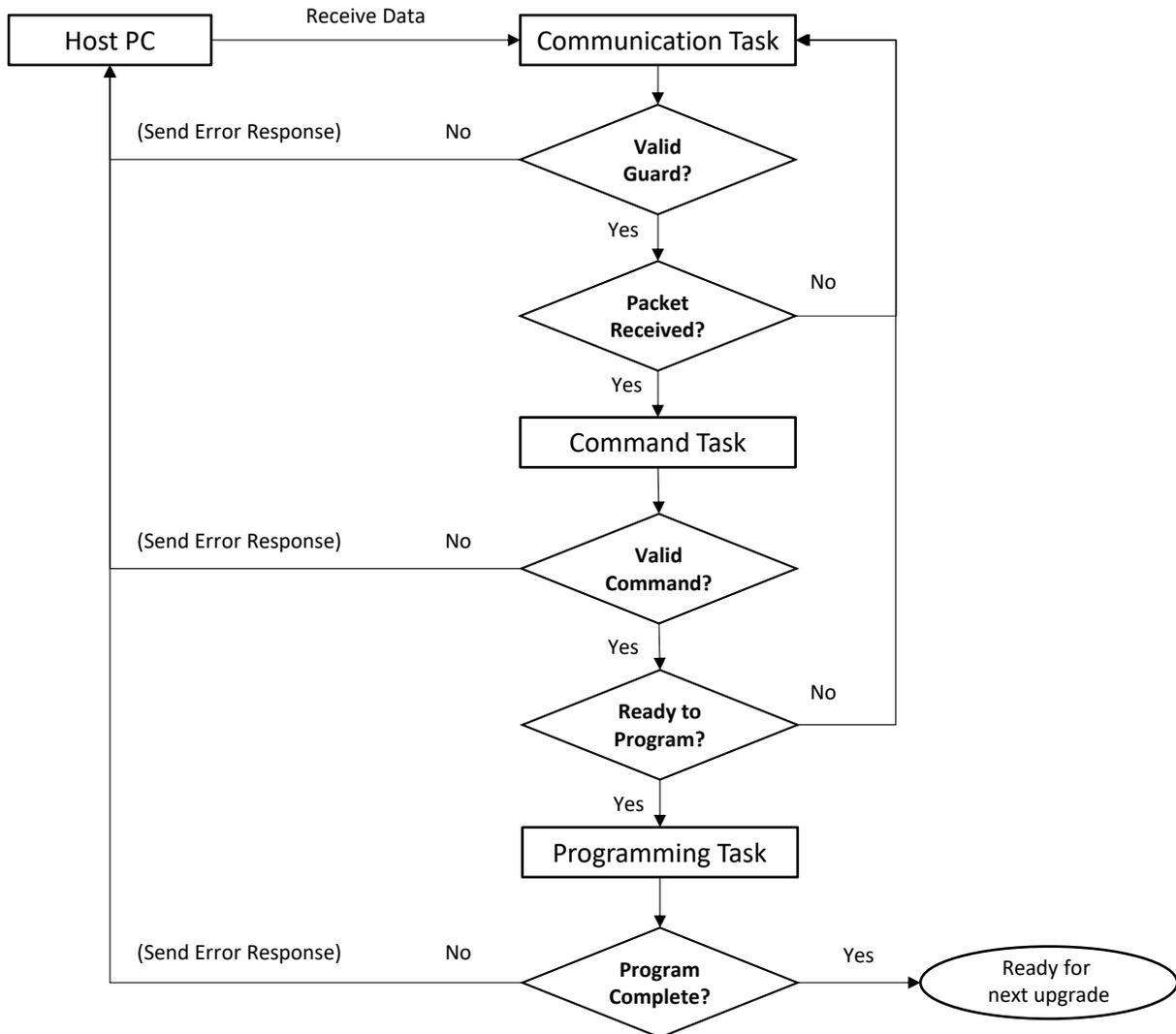
该任务处理从通信任务接收的命令并作出响应，然后相应地将响应返回给主机 PC。如果接收到的命令是编程命令，则该任务会将控制权交给编程任务。

2.5 编程任务

此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。

下图所示为固件升级执行流程图。

图 2-7. 固件升级执行流程图



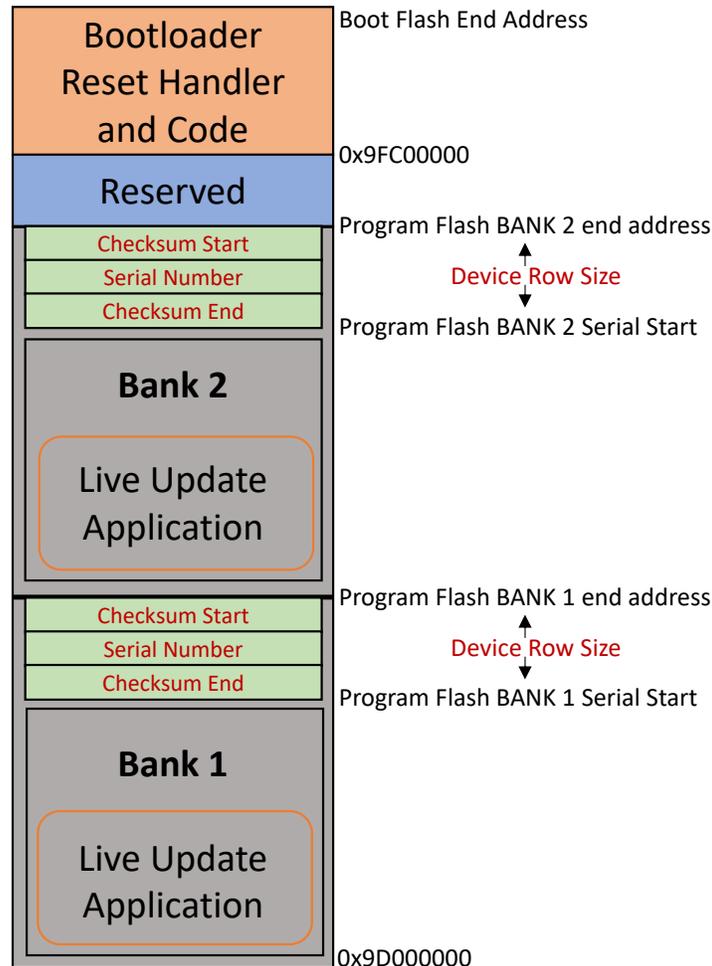
2.6 存储器布局

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。以下各部分介绍了 PIC32MZ 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。

- 自举程序代码始终位于引导闪存的起始地址 **KSEG0**（可高速缓存虚拟地址 **0x9FC00000**）或 **KSEG1**（不可高速缓存虚拟地址 **0xBFC00000**）处。在本应用程序中，自举程序代码位于 **KSEG0** 虚拟地址处。复位后，器件从引导闪存起始地址运行。
- 闪存程序存储器分为两个相等的存储区（**BANK1** 和 **BANK2**）。每个存储区都将映射到应用程序必须在其中运行的较低区域（**0x1D000000**）：
 - 映射到较低区域的活动存储区的起始地址始终为 **0x9D000000**
 - 非活动存储区的起始地址来自闪存程序存储器的中间，具体因器件而异。有关闪存布局的详细信息，请参见相应的数据手册。
- 非活动存储区的起始地址来自闪存程序存储器的中间，具体因器件而异。有关闪存布局的详细信息，请参见相应的数据手册。
- 在每个存储区末尾保留行大小字节数，用于存储序列号。实时更新应用程序将使用此序列号将适当的存储区映射到较低存储区，并从中运行实时应用程序。

- 应用程序起始地址必须始终位于低映射区域（0x9D000000 至闪存中间值）。链接描述文件中应用程序的大小不得超过闪存的中间值。
- 根据执行的更新，在编程期间传递给自举程序的地址必须处于活动存储区或非活动存储区内。

Live Update memory layout PIC32M



2.7 执行流程

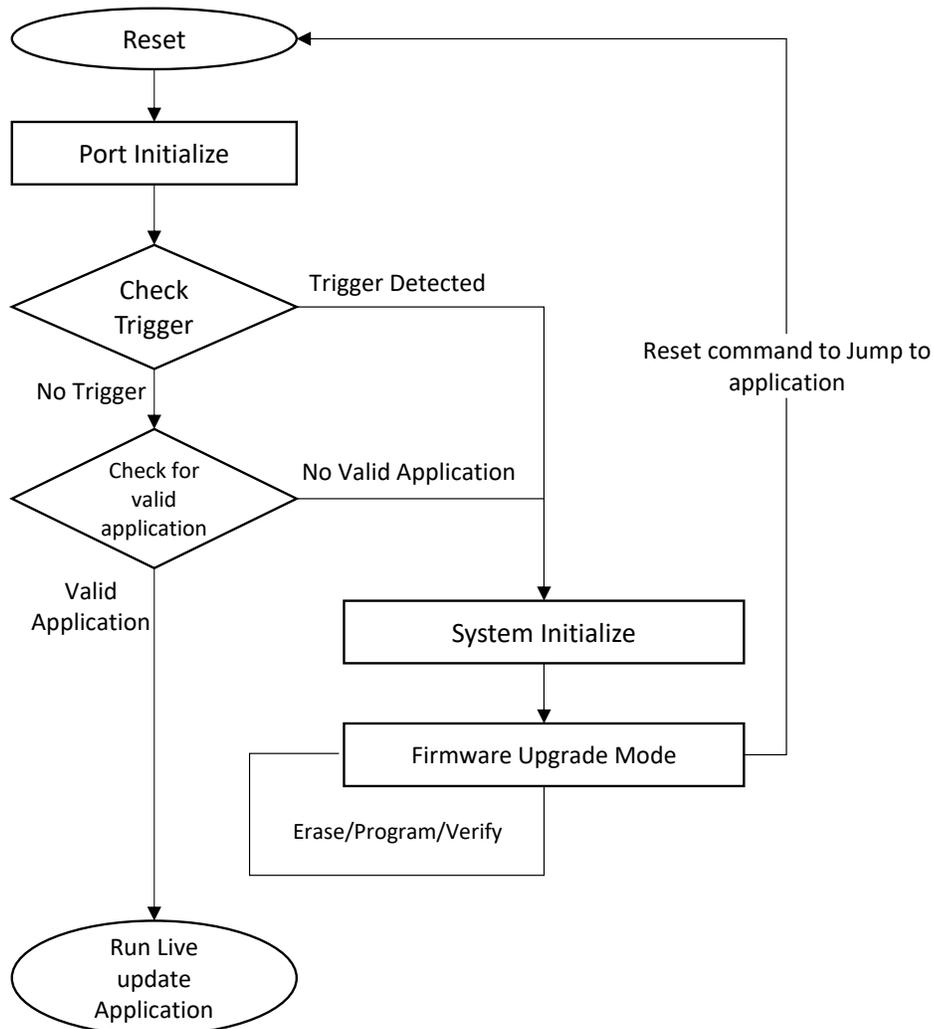
首次使用 MPLAB Harmony v3 基本自举程序或 UART 故障安全自举程序对实时更新应用程序进行编程。以下部分将讨论 MPLAB Harmony v3 基本自举程序和实时更新应用程序的系统级执行流程。

基本自举程序系统级执行流程：

基本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，则基本自举程序开始执行用户应用程序。基本自举程序在固件升级模式下可以执行闪存擦除/编程操作。

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

图 2-9. 编程实时更新应用程序的 MPLAB Harmony v3 自举程序流程



有关 MPLAB Harmony v3 基本或故障安全自举程序的更多信息，请参见以下链接。

`<Harmony framework download folder>\bootloader_apps_uart\apps\`

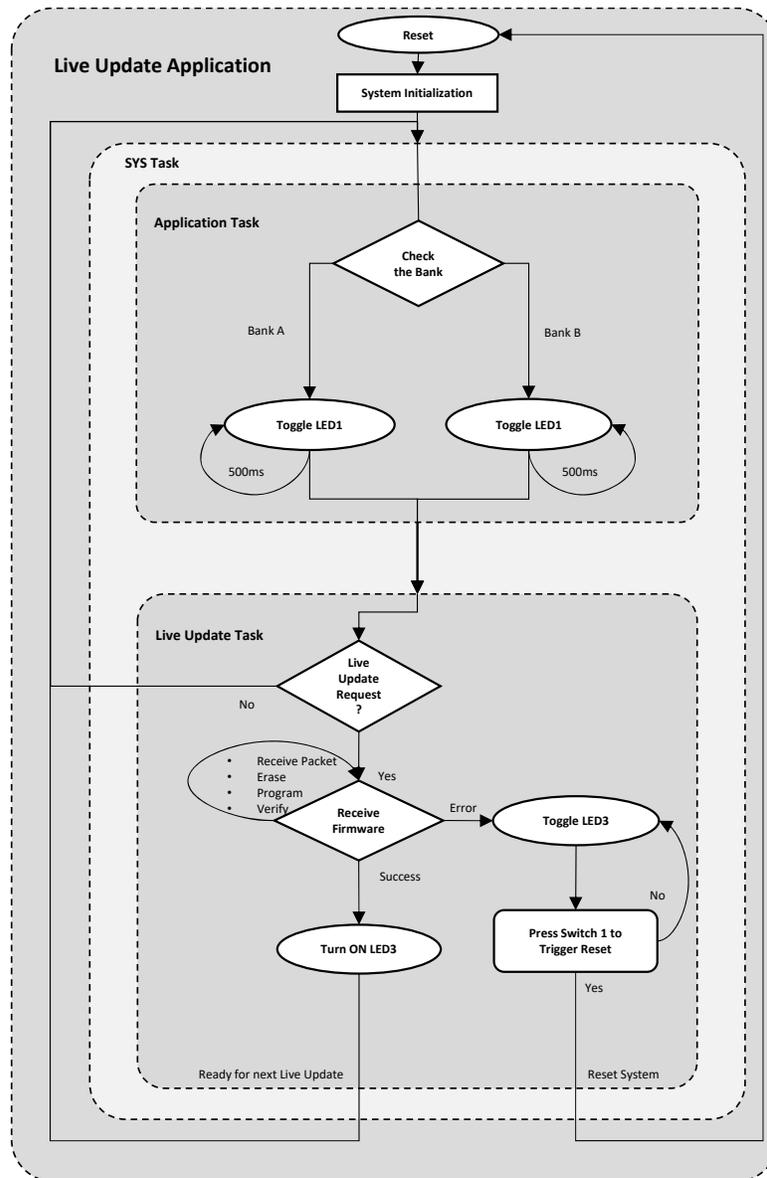
实时更新系统级执行流程：

通过 MPLAB Harmony v3 自举程序成功编程实时更新应用程序后，自举程序将开始执行该实时更新应用程序。

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。新更新的固件将在下一个上电周期或系统复位时加载。

下图说明了实时更新应用程序的执行流程。

图 2-10. 实时更新应用程序执行流程



流程图序列:

1. 复位后，器件自举程序会将实时更新应用程序加载到双存储区的活动存储区。
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. 应用程序任务会检查正在执行实时更新应用程序的活动存储区，并相应地翻转 LED 1 或 LED 2。
4. 实时更新任务仅在有固件升级请求时运行，否则它将处于空闲状态并且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：
 - 仅在从主机 PC 接收到 UNLOCK 命令后，才开始接收数据包。
 - 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 命令时，会验证接收到的固件映像。
 - 接收到 SWAP Bank 命令时，会从活动存储区读取序列号，以 1 为增量，然后写入存储区 2 序列号。
6. 如果主机应用程序请求更新活动存储区并且地址处于活动存储区串行扇区，则实时更新任务会发送错误响应，中止编程操作，并翻转到 LED3。

7. 如果在步骤 5 中发生任何错误，则会从实时更新模式返回，并且每 1000 ms 翻转一次 LED3，直到按下 Switch1 以触发系统复位来再次重新启动实时更新。
8. 如果实时更新成功，则会点亮 LED3 并跳转到步骤 5 以进行下一次实时更新。

3. 配置

实时更新应用程序包含以下应用程序：

- **自举程序** (uart_fail_safe_bootloader_pic32mz_ef_sk)：用于升级实时更新应用程序。
- **实时更新应用程序** (pic32mz_uart_live_update)：本应用程序包含实时更新功能的实现。

注：可以使用以下路径在 MPLAB Harmony v3 自举程序资源库中找到 MPLAB Harmony v3 自举程序项目：

<Harmony framework download folder>\bootloader_apps_uart\apps\uart_fail_safe_bootloader\

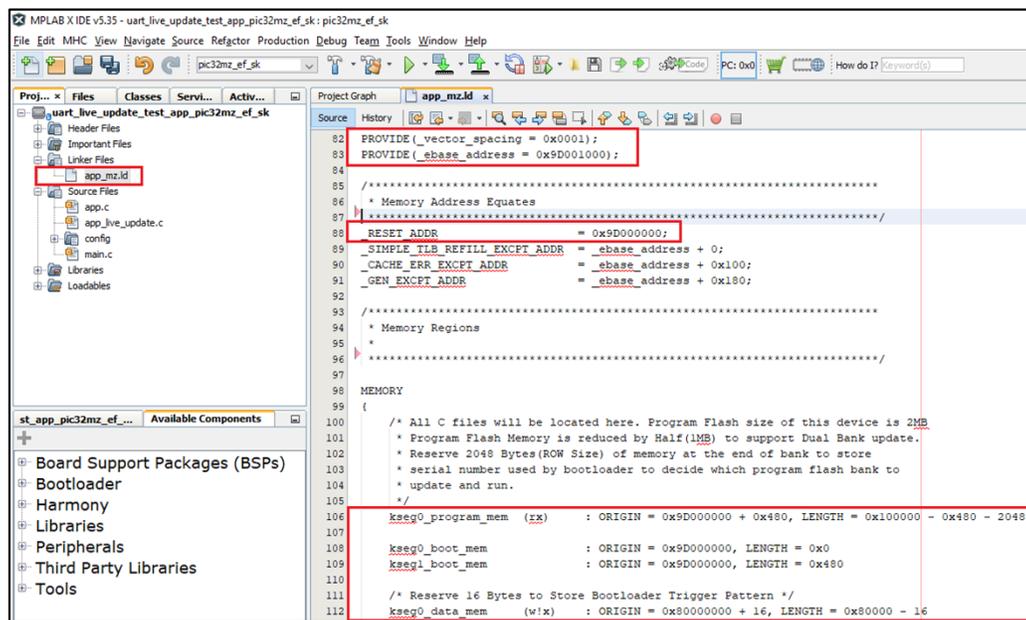
检查以下路径以获取有关如何配置 UART 故障安全自举程序的更多信息：

<Harmony framework download folder>\bootloader\doc\help_bootloader.chm

3.1 链接描述文件

PIC32MZ 器件的 XC32 链接描述文件带有默认复位地址和存储区地址。由于自举程序处于引导闪存中，因此包括复位处理程序和中断向量表 (Interrupt Vector Table, IVT) 在内的实时应用程序代码必须置于闪存程序存储器中。更多信息，请参见存储器布局部分。因此，必须创建自定义链接描述文件 (app_mz.ld)，并将其添加到实时更新应用程序项目的应用程序项目中。下图显示了 PIC32MZ 器件的应用程序自定义链接描述文件和突出显示的更改。

图 3-1. 实时更新应用程序自定义链接描述文件



在 PIC32MZ 自定义链接描述文件中进行了以下更改：

- 使用异常基址 (EBASE) CPU 寄存器计算给定中断的向量地址，而该异常基址寄存器在内核段 (kseg) 地址空间提供 4 KB 页对齐的基址值。
- MIPS32 内核要求将 IVT 置于 4 KB 边界上。
- 该地址使用 EBASE 和 VS 值进行计算。VS 位提供两个相邻向量地址之间的向量间距。
- 表中的每个向量都根据 _ebase_address 和 _vector_spacing 符号的值创建为绝对地址处的输出段。
- 通过自举程序加载的应用程序复位地址应与自举程序项目中提到的实时应用程序起始地址相匹配。
- 对于具有较大引导闪存的器件，自举程序完全置于引导闪存中，应用程序起始地址默认为闪存程序存储器的起始地址 0x9D000000。

```
_RESET_ADDR = 0x9D000000;
kseg0_program_mem (rx) : ORIGIN = 0x9D000000 + 0x480, LENGTH = 0x200000 - 0x480
```

```

kseg0_data_mem      (w!x) : ORIGIN = 0x80000000, LENGTH = 0x80000
kseg1_boot_mem     : ORIGIN = 0x9D000000, LENGTH = 0x480
/* 引导部分 */
.reset _RESET_ADDR :
{
    KEEP(*(.reset))
    KEEP(*(.reset.startup))
} > kseg1_boot_mem

```

注:

- 实时更新应用程序自定义链接描述文件中删除了器件配置位，因为此位必须由自举程序更新
- 必须从应用程序项目的最终十六进制文件中丢弃器件配置。

```
/DISCARD/ : { *(.config_*) }
```

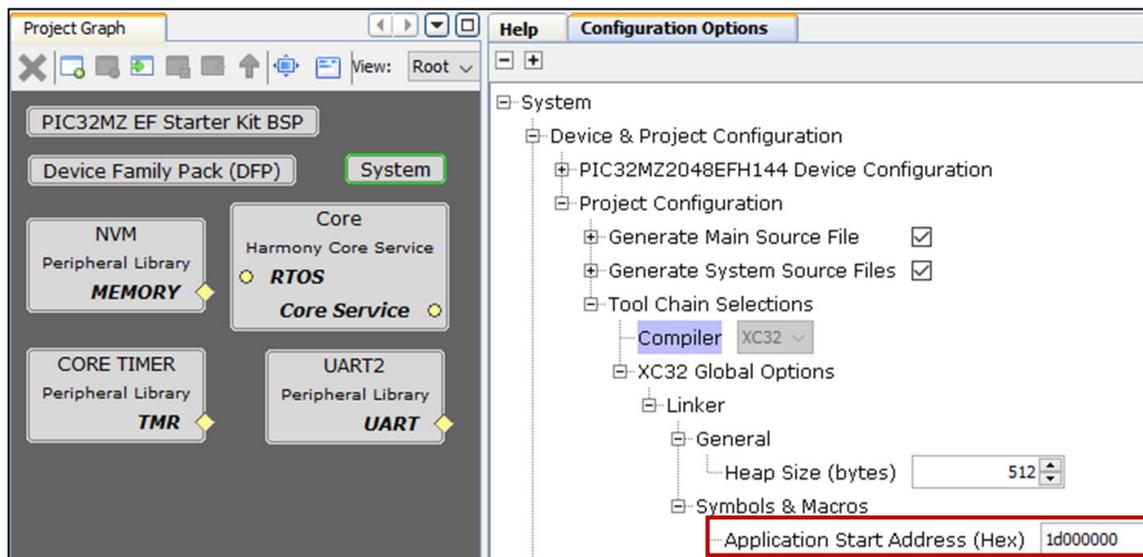
- 请参见位于 <Live update application folder>/firmware/src/config/pic32mz_ef_sk/app_mz.ld 中预先开发的实时更新应用程序自定义链接描述文件。

3.2 MHC 配置

以下步骤介绍了实现实时更新应用程序所需的 MHC 配置：

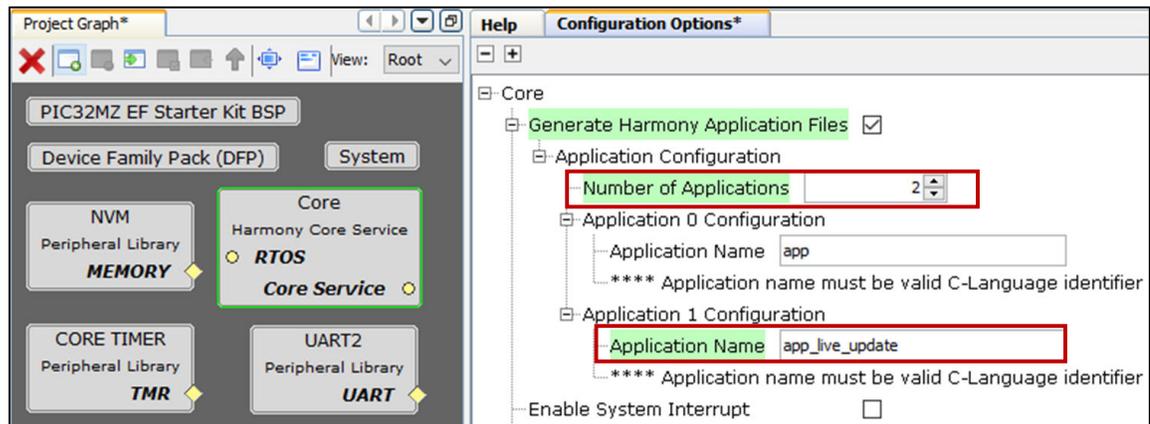
1. 从自定义链接描述文件中移除器件熔丝配置，因为该配置将由自举程序项目更新。
2. 为与自举程序项目中的应用程序起始地址相匹配的应用程序起始地址，提供自定义实时应用程序链接描述文件。

图 3-2. 实时更新应用程序起始地址配置



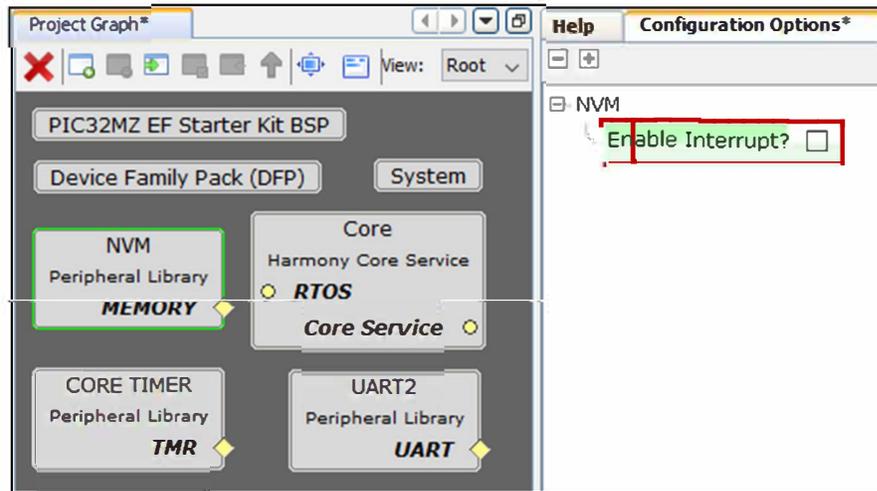
3. 添加 MPLAB Harmony v3 核心组件并配置应用程序和实时更新任务。将生成源文件，例如可用于添加任何子任务的 tasks.c，以及用于添加任何用户配置的用户.h。

图 3-3. 实时更新应用程序 MPLAB Harmony v3 核心配置



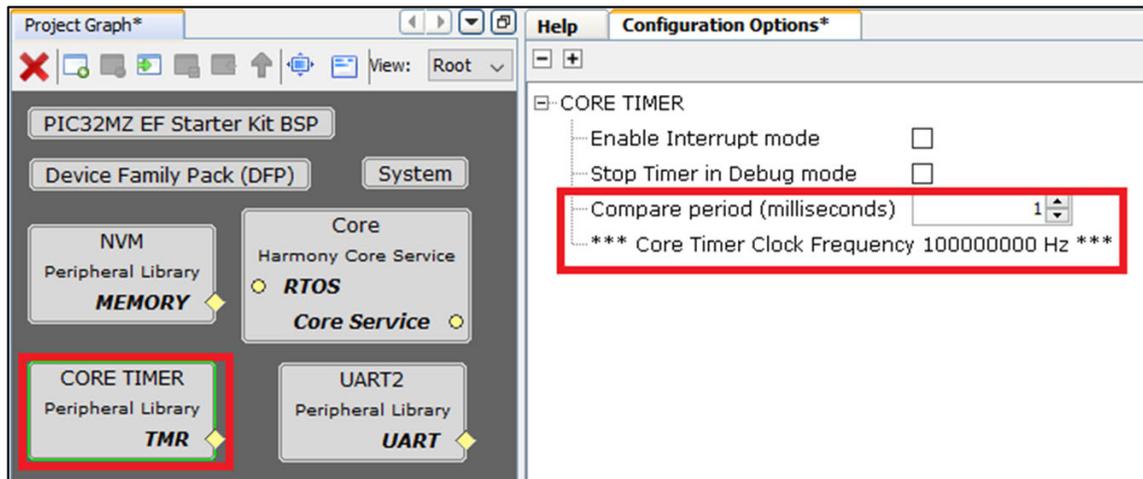
4. 添加 NVM 模块并禁止中断。实时更新应用程序在轮询模式下使用 NVM PLIB，因此必须禁止中断，如下图所示。

图 3-4. 实时更新应用程序 NVM 配置



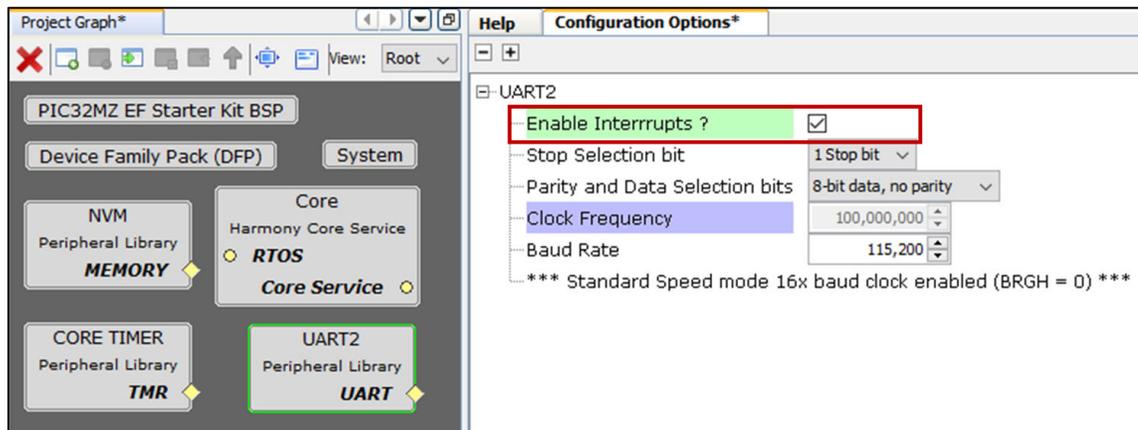
5. 用 1 ms 的比较周期配置 CORE TIMER（内核定时器），使 LED 以不同的速率闪烁。

图 3-5. 实时更新应用程序内核定时器配置



6. 如下图所示，添加 UART 模块以接收来自主机的数据并允许中断。

图 3-6. 实时更新应用程序 UART 配置



3.3 项目设置

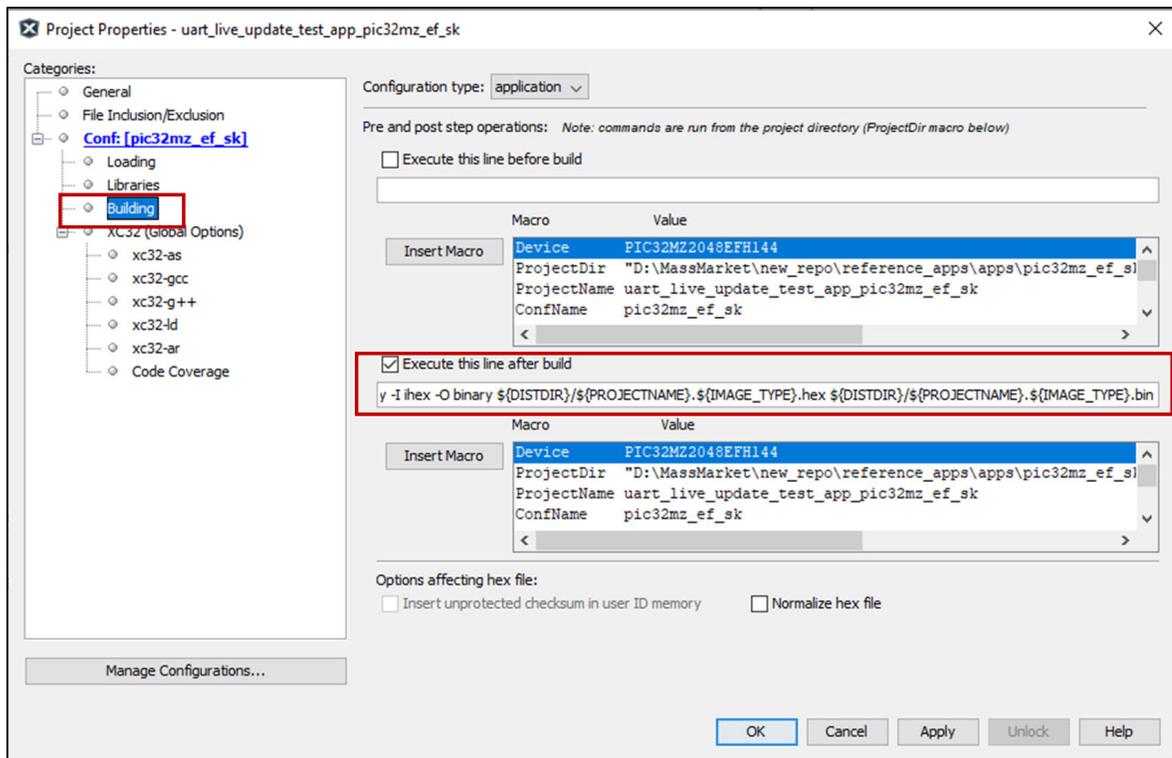
- **Preprocessor macro definitions (预处理器宏定义) :**
 - ROM_ORIGIN 和 ROM_LENGTH 为 XC32 链接器变量，将由提供的值改写。
 - 重新生成后，将使用 MHC 中提供的应用程序起始地址的值自动填充链接描述文件中的应用程序起始地址值（请参见实时更新应用程序自定义链接描述文件）。
- **Execute this line after build (在编译后执行该行) :**
 - 以下编译选项可用于在编译完成后自动从十六进制文件生成二进制文件。

```

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

```

图 3-7. 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

首次使用 MPLAB Harmony v3 基本自举程序或 UART 故障安全自举程序对实时更新应用程序进行编程时，请执行以下步骤：

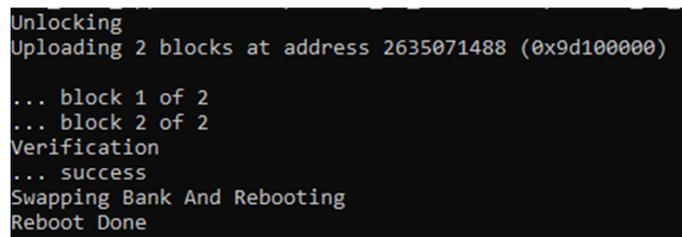
1. 下载 MPLAB Harmony v3 [自举程序包](#)。
2. 下载 MPLAB Harmony v3 [UART 自举程序应用程序包](#)。
3. 将 mini USB 电缆连接到带 FPU 的 PIC32MZ (EF) 嵌入式连接入门工具包的 DEBUG 端口。
4. 使用 MPLAB X IDE 编译和编程 (<Harmony Framework download folder>/bootloader_apps_uart/apps/uart_fail_safe_bootloader/bootloader/firmware/pic32mz_ef_sk.X) 中提供的 UART 故障安全自举程序 (双存储区自举程序)。
5. 使用 MPLAB X IDE 编译实时更新应用程序 (<Live Update application folder>/firmware/pic32mz_ef_sk.X)，但不进行编程。
6. 根据命令提示符，运行<Harmony Framework download folder>/bootloader/tools_archive 中提供的自举程序主机描述文件 btl_host.py，以编程应用程序二进制文件：

```
- python <Your Harmony Framework download folder>/bootloader/tools_archive/  
btl_host.py -v -s -i <COM PORT> -d pic32mz -a 0x9D100000 -f <Live Update  
application folder>/firmware/pic32mz_ef_sk.X/dist/pic32mz_ef_sk/production/  
pic32mz_ef_sk.X.production.bin
```

注： 有关自举程序主机描述文件的更多信息，请参见设置主机描述符的帮助文件，这些文件位于 <Harmony Framework download folder>/bootloader/doc/help_bootloader.chm。

7. 下图显示了成功编程实时更新应用程序二进制文件的结果。消息 “Swapping Bank And Rebooting” (正在交换存储区和正在重启) 以及 “Reboot Done” (重启完成) 表示重启成功。

图 4-1. 首次使用自举程序进行实时更新应用程序编程



```
Unlocking  
Uploading 2 blocks at address 2635071488 (0x9d100000)  
  
... 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

1. 如果尚未完成，请执行[运行自举程序应用程序](#)步骤。
2. 如果上述步骤成功，带 FPU 的 PIC32MZ (EF) 嵌入式连接入门工具包上的 LED1 或 LED2 将开始闪烁，这表明应用程序正在从 BANK 1 或 BANK 2 运行，具体取决于运行应用程序的位置。
3. 对器件进行复位或掉电再上电。
4. 从位于<Live Update application folder>/scripts 的命令提示符处针对实时更新应用程序运行主机描述文件 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最新版本的固件编程到非活动分区。

```
python <Live Update application folder>/scripts/live_update.py  
-v -s -i <COM PORT> -d pic32mz -a 0x9D100000 -f <Live Update application folder>/  
firmware/pic32mz_ef_sk.X  
/dist/pic32mz_ef_sk/production/pic32mz_ef_sk.X.production.bin
```

- 下图给出了实时更新描述文件帮助。

图 4-2. 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
  -s, --swap          swap banks after programming
  -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>: PIC32MZ
<Address>: Application start address 0x9D100000
Example: python <Live update application folder >\scripts\live_update.py -v -s -i COM6 -
d pic32mz -a 0x9D100000 -f pic32mz_ef_sk.X.production.bin
Note: 运行帮助命令时会提供关于可用选项的简要概述，如下所示。
Command: python <python script> --help
<python script>: live_update.py
```

5. 下图给出了实时更新固件升级输出。

图 4-3. 实时更新应用程序固件升级输出

```
Live Update request started.

Unlocking
Uploading 2 blocks at address 2635071488 (0x9d100000)
... block 1 of 2
... block 2 of 2
Verification
.. success
Swapping Bank
... success

Live Update is success.

Reset the device for the programmed application firmware to run.
```

6. 如果实时更新成功，则 LED3 将点亮。
7. 如果在步骤 4 的实时更新期间出错，则 LED3 会熄灭。使用以下步骤从错误中恢复：
 - 7.1. 长按开关 SW1 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。
8. 如果实时更新成功，LED1 或 LED2 将开始闪烁，这表明实时更新应用程序正在从 BANK 1 或 BANK 2 运行，具体取决于运行程序的位置。
 - LED1 对应于 BANK 1
 - LED2 对应于 BANK 2
9. 将器件复位或掉电再上电，以运行新编程的实时更新应用程序。
 - 观察 LED 翻转与上一步中翻转的 LED 相比发生的变化。这表明新更新的映像正常运行。

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，从而帮助简化产品的部署。

本文档可用作参考资料，帮助用户在最终用户应用程序中实现实时更新功能。

6. 参考资料

- MPLAB Harmony v3 自举程序:
<Harmony 框架下载文件夹>\bootloader\doc\help_bootloader.chm
- MPLAB Harmony v3 自举程序应用程序:
<Harmony 框架下载文件夹>\bootloader_apps_uart\apps
- PIC32MZ EF MCU 上 Harmony v3 外设库入门:
microchipdeveloper.com/harmony3:pic32mzef-getting-started-training-module
- 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 FreeRTOS 的 PIC32MZ EF MCU Harmony v3 驱动程序和中间件入门:
microchipdeveloper.com/harmony3:pic32mz-get-start-tm-drvr-middleware-freertos
- PIC32MZ EF MCU 上 Harmony v3 外设库的入门:
microchipdeveloper.com/harmony3:pic32mzef-getting-started-training-module
- 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 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 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

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 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、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 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。

© 2021, Microchip Technology Incorporated 版权所有。

ISBN: 978-1-5224-8355-7

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			