Application Note

MSPM0 实时固件更新 (LFU) 引导加载程序实施

Gary Gao and Luke Chen

摘要

本应用手册提供了一种无需暂停应用代码即可进行固件更新的方法。这种方法基于 MSPM0G3507 并使用 FreeRTOS 进行任务处理。还提供了一个 PC GUI 作为主机，可以帮助生成此演示的使用文件。

本文中所讨论的项目配套资料和源代码可从以下 URL 下载：https://www.ti.com/cn/lit/zip/slaaec9。

内容

插图清单

表格清单

商标

1 引言

MSPM0 器件支持基于 ROM 的 BSL (引导加载程序) 、基于闪存的引导加载程序和可用于固件升级的插件接口。然而，这些引导加载程序在固件升级期间会占用 CPU，这意味着正在执行的应用代码将被暂停，直到固件升级过程完成。

在某些应用中，不允许在固件升级过程中暂停应用代码。本应用报告提供了一种在固件升级期间不暂停应用代码的方法。固件升级过程完成后，将执行器件下电上电和较新版本的固件。

2 LFU 引导加载程序功能概述

LFU 引导加载程序的主要功能包括：

- 通过通用异步接收器/发送器 (UART) 端口与主机通信
- 基于 FreeRTOS™ 的软件示例
- 软件 BSL 调用
- 已启用密码保护
- 易于使用的 Windows 主机 GUI
- 自动生成链接器和头文件，轻松迁移到其他 MSPM0 器件
- 提供非主闪存修改解决方案
- 该引导加载程序的代码大小小于 16KB。

3 硬件和软件设置

3.1 硬件要求

- LP-MSPM0G3507 LaunchPad
- 装有 Windows 的 PC
- Miro USB 型电缆

3.2 软件设置

- Code Composer Studio™ (CCS) 12.3 或更高版本
- Uniflash 8.3 或更高版本
- MSPM0 SDK

UART 接口设置

- 波特率 9600bps
- 数据宽度 - 8 位
- 1 个停止位
- 无奇偶校验位

4 LFU 引导加载程序实现

4.1 LFU 引导加载程序和应用项目

在本应用手册中，使用了三个项目，如表 4-1 中所述。

表 4-1. 本演示所需的项目

CCS 项目	说明
LFU 引导加载程序	这是一款基于 FreeRTOS 的引导加载程序软件，大小约为 16KB，分配给主闪存的起始地址。在新器件上，需要先通过 SWD 接口对该引导加载程序进行编程，然后才能开始固件升级过程。
应用代码 1	这是固件升级演示的应用代码 1，它使用应用空间 1 (0x04000 - 0x11FFF) 来实现功能。
应用代码 2	这是固件升级演示的应用代码 2，它使用应用空间 2 (0x12000 - 0x1FFFF) 来实现功能。

4.2 内存分配

主闪存存储器分为三部分：

LFU 引导加载程序

主闪存存储器的前 16KB (0x0000 - 0x3FFF) 保留给 LFU 引导加载程序，其余主闪存空间分配给应用代码。

应用代码：

该内存空间分为应用空间 1 和应用空间 2，仅执行一个版本的应用代码，另一个应用代码空间用于新固件升级。

RAM 空间的前 4KB (0x2020000 - 0x20200FFF) 保留给引导加载程序，其他存储器空间可用于应用代码。

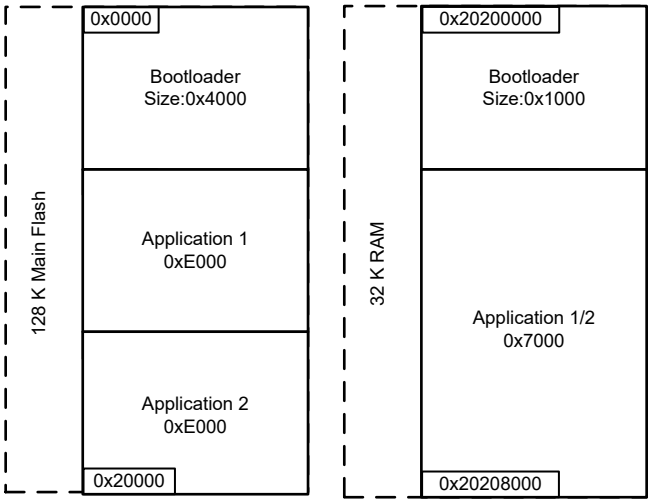

图 4-1. 存储器排列

4.3 已实现的 LFU 引导加载程序

需要一种任务切换管理工具，以便应用代码在固件升级过程中保持执行。在本应用手册中，FreeRTOS 用于处理此作业，并为此演示创建四项任务。

表 4-2. LFU 引导加载程序中的任务

任务名称	优先级	说明
引导加载程序任务	2	处理固件升级的任务。如果应用代码正在执行，此任务将挂起。
LED0 切换任务	1	每 500ms 切换一次 LED 的任务，这用于表明 FreeRTOS 正在正常工作。
应用任务	1	执行应用代码的任务。
空闲	0	当没有任何其他任务等待执行时的默认任务。

在新器件上，需要首先通过 SWD 接口对引导加载程序固件进行编程，以便引导加载程序与主机（PC GUI 或主机 MCU）通信，并更新应用代码。还可以通过 SWD 将应用代码与引导加载程序固件一起进行编程。

当器件启动并且没有应用代码时，引导加载程序等待来自主机的固件升级命令。

图 4-2 展示了引导加载程序流程图。

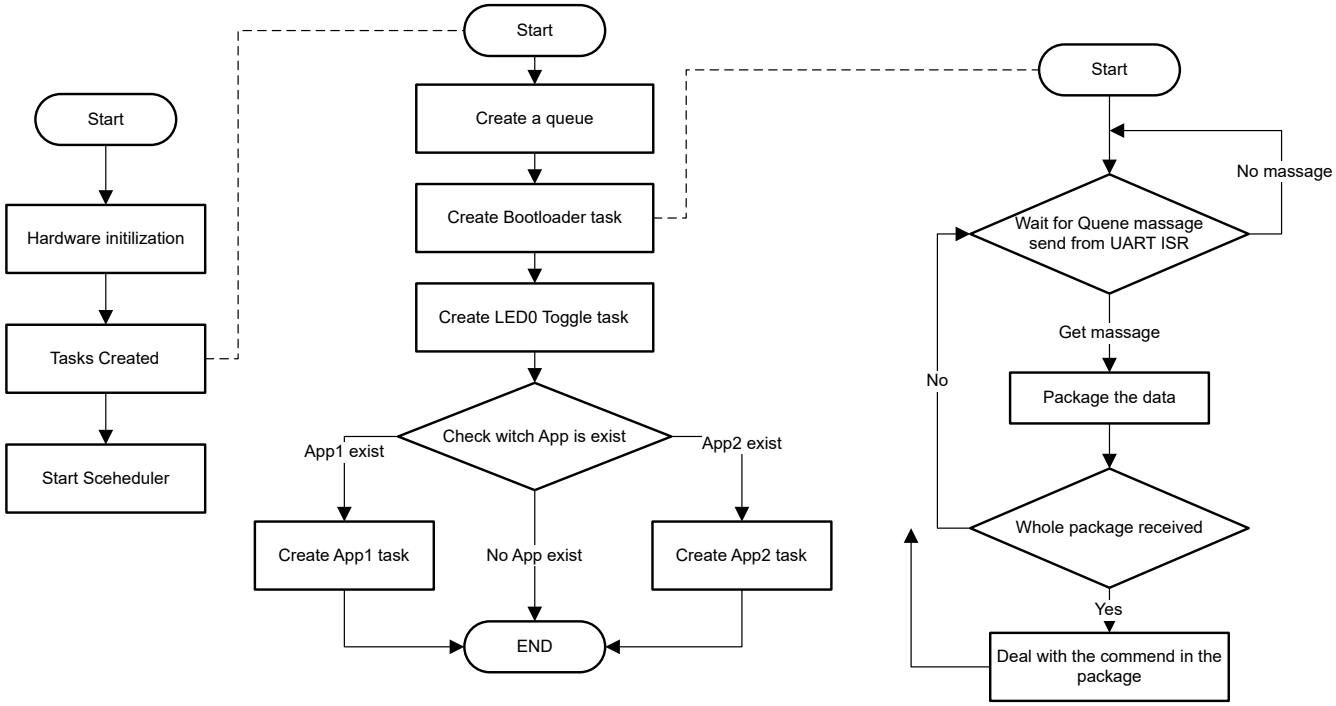

图 4-2. 引导加载程序代码进度图

引导加载程序支持以下命令。相关详细信息，请参阅节 6。

- CMD_UNLOCK_BSL
- CMD_FLASH_RANGE_ERASE
- CMD_PROGRAM_DATA
- CMD_START_APPLICATION

4.4 LFU 应用代码实现

有两个应用项目：应用代码 1 和应用代码 2，分别位于不同的闪存区域。这两个项目切换不同的 LED。应用代码可由引导加载程序中的 FreeRTOS 作为任务线程调用。您必须在应用代码中使用 FreeRTOS 中定义的延迟函数来实现延时目的。在此演示示例中，引导加载程序项目将 FreeRTOS 延迟函数的起始地址放置在固定闪存地址 0x3FF0 到大约 0x3FFF 处，此范围称为共享区域，应用项目代码只需调用延迟函数即可选取共享区域中的起始地址。

4.4.1 应用的链接器命令文件

链接器命令文件由链接器在项目构建过程中使用，PC GUI 工具可以自动生成此文件，无需为项目处理此链接器命令文件。

4.4.2 外设和中断初始化

引导加载程序会初始化一些模块，如时钟模块、UART0 和电源模块。建议不要重新初始化已经被引导加载程序初始化的模块，否则引导加载程序的设置可能会丢失。

下面的函数用于注册中断。它将中断表复制到 SRAM，并根据输入参数修改 ISR 起始地址。

```
void DL_Interrupt_registerInterrupt(uint32_t exceptionNumber, void (*intHandler)(void));
```

4.4.3 应用项目调试

由于以下原因，CCS 无法直接下载并执行应用项目：

- 生成的链接器命令 (cmd) 文件中的中断向量表未分配给器件启动时的默认地址 0x0000。
- 应用项目中没有器件模块初始化函数。
- 闪存共享区域中不能使用 FreeRTOS 延迟功能。

以下是调试应用项目的推荐权变措施。

- 使用 SDK 示例中的默认 cmd 文件将中断向量表分配到启动时的地址 0x0000。
- 使用 Sysconfig 工具为应用项目生成外设初始化代码。
- 将 FreeRTOS 延迟函数替换为在 SDK 中定义的函数 delay_cycles();。

4.5 调用固件升级过程

要启动固件升级过程时，有两种情况：

- 如果器件中仅执行了引导加载程序代码，引导加载程序任务会等待来自主机的固件升级命令，因此在这种情况下，在开始固件升级过程之前不需要执行额外的操作。
- 如果器件中同时执行引导加载程序和应用代码，在执行应用代码时，引导加载程序任务将挂起。在这种情况下，您需要在开始固件升级过程之前强制引导加载程序任务状态从挂起状态变为活动状态，您可以通过 UART 发送一字节调用命令 0xAA 来实现此目标。

5 主机 GUI 工具简介

主机可以是一个通过 UART 端口发送命令的 MCU、处理器或 PC。有关详细的主机命令信息，请参阅节 6。

在本演示中，使用板载仿真器 XDS110 来处理 USB 转 UART 功能，以便您可以使用 PC 作为主机，并通过 USB 端口发送 UART 命令。Python 创建的 PC GUI 工具用于发送 UART 命令，该 GUI 工具有三个功能。

- LFU 固件等级。
- 应用项目链接器命令文件生成。
- 非主闪存配置固件生成。

5.1 LFU 固件更新

在开始 LFU 固件升级过程之前确认以下各项：

- 确保引导加载程序固件已通过 SWD 接口编程到器件中。
- 构建应用项目，并生成固件升级所需的 .txt 文件。建议同时生成应用代码 1 和应用代码 2 以用于评估。

- 准备 .txt 格式的 BSL 密码文件。

现在，您可以通过双击“…\MSPM0 LFU Bootloader Implementation v1.1\BSL_GUI_EXE”文件夹中的 MSPM0_LFU_BSL_GUI.exe 文件来启动 PC GUI 工具。以下步骤介绍了如何执行固件升级过程：

1. 检查应用代码状态，了解哪些应用代码已执行或器件中没有哪些应用代码。
2. 选择用于固件升级评估的应用代码。
   a. 如果器件中没有应用代码，可以选择应用代码 1 或应用代码 2 进行固件升级。
   b. 如果器件中执行了应用代码 1，则选择应用代码 2 进行固件升级。
   c. 如果器件中执行了应用代码 2，则选择应用代码 1 进行固件升级。
3. 根据输入文件夹中默认密码文件的格式选择一个密码文件。
4. 点击“Download”按钮进行固件更新。

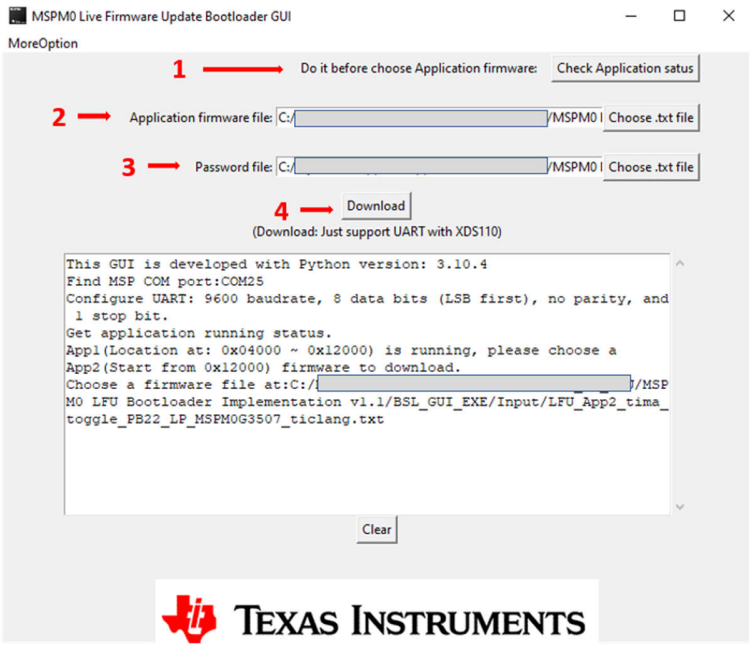

图 5-1. 使用 GUI 更新固件的步骤

5.2 应用项目链接文件生成

如果您想将此示例项目迁移到任何其他 MSPM0 器件，需要针对特定器件的应用代码的不同链接器命令文件。GUI 工具可以做到这一点，根据您输入的 MSPM0 器件型号自动生成 cmd 文件。以 MSPM0G3507 为例，按照以下步骤生成所需文件：

1. 在 GUI 工具中，点击“More Option”菜单，并选择选项“Create Linker Files”。
2. 输入 MSPM0 器件型号，例如 MSPM0G3507。
3. 选择一个文件夹来保存生成的文件。
4. 点击“Generate”按钮生成文件。

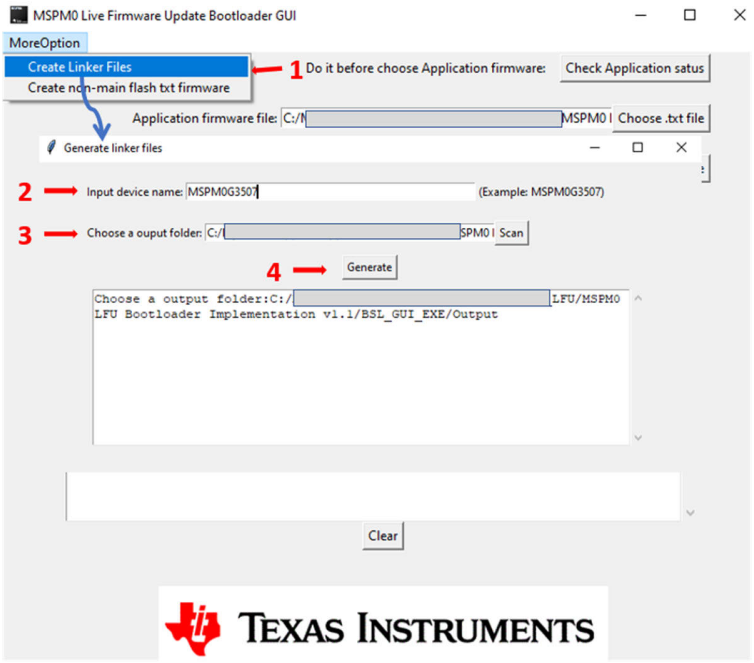

图 5-2. 使用 GUI 生成链接文件的步骤

5. 生成以下三个文件：
   a. mspm0g3507_App1.cmd，这是应用代码 1 的 cmd 文件。
   b. mspm0g3507_App2.cmd，这是应用代码 2 的 cmd 文件。
   c. device.h，这是引导加载程序和两个应用代码所需的文件。

5.3 非主闪存配置固件生成

NONMAIN 是闪存的专用区域，用于存储 BCR（引导配置例程）和 BSL 用于引导器件的配置数据。本示例项目中使用的密码也可用于 BSL，如果要更改密码，需要修改 NONMAIN 闪存配置。

GUI 工具不仅支持 BSL 密码修改，还支持所有 NONMAIN 闪存配置选项。

5.3.1 生成非主闪存配置固件的步骤

以下是使用 GUI 修改密码的步骤：

1. 点击“More Option”菜单，并选择选项“Create non-main flash txt firmware”。
2. 如果需要更改为其他器件系列，请点击“Change”按钮。
3. 点击 BSLPW 按钮。
4. 输入您的新 BSL 密码。
5. 点击“OK”按钮保存新密码，然后关闭此对话窗口。
6. 输入此修改的新版本号。
7. 点击“Generate”按钮，生成 NONMAIN 闪存配置数据和密码文件。

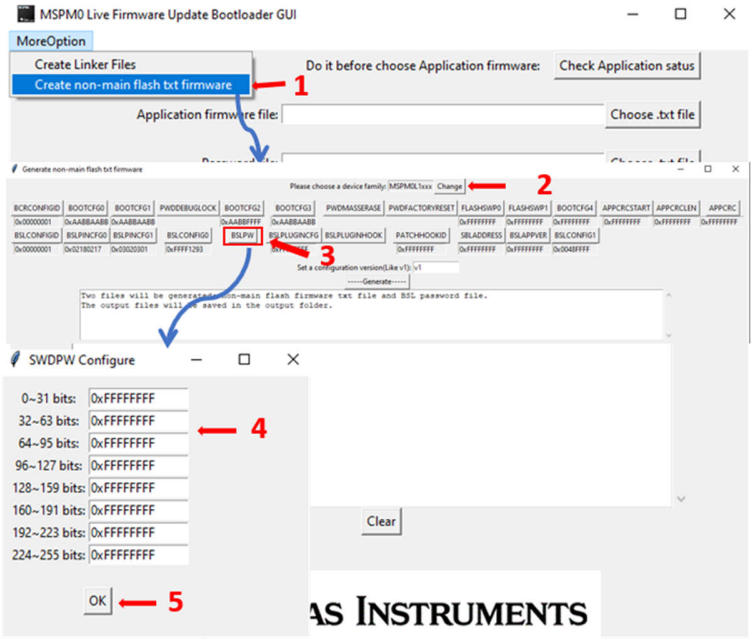

图 5-3. 使用 GUI 修改密码的步骤

默认输出文件夹中保存了两个生成的文件，如下所示：

- Non_main_flash_firmware_v1.txt
- BSL_Password_v1.txt

您可以使用 UNIFLASH 工具通过 SWD 接口将 Non_main_flash_firmware_v1.txt 编程到器件中，配置 NONMAIN 闪存区域。BSL_Passward_v1.txt 文件用于 GUI 工具固件升级过程。

5.3.2 用于对 NONMAIN 闪存配置数据进行编程的 UNIFLASH 工具

1. 将 LP-MSPM0G3507 连接到 PC 并启动 UNIFLASH 工具。
2. 手动选择 EVM 或让 UNIFLASH 自动检测器件型号。

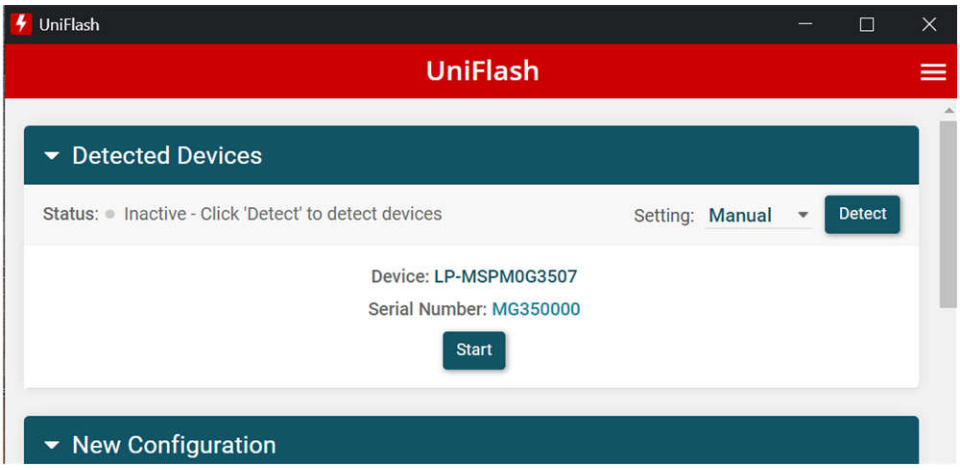

图 5-4. Uniflash 检测到主板

3. 选择“Settings and Utilities”选项，然后选中“Erase main and non-main memory”。

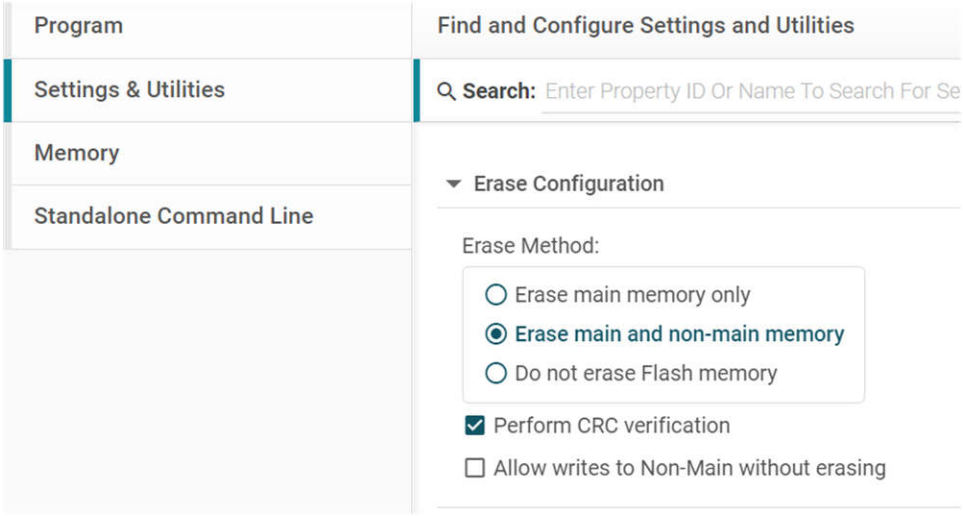

图 5-5. 更改擦除方法

4. 选择“Program”选项，浏览生成的文件 Non_main_flash_firmware_v1.txt，然后点击“Load Image”按钮。（如果启用了静态闪存保护，建议添加需要下载静态保护区域的图像）。

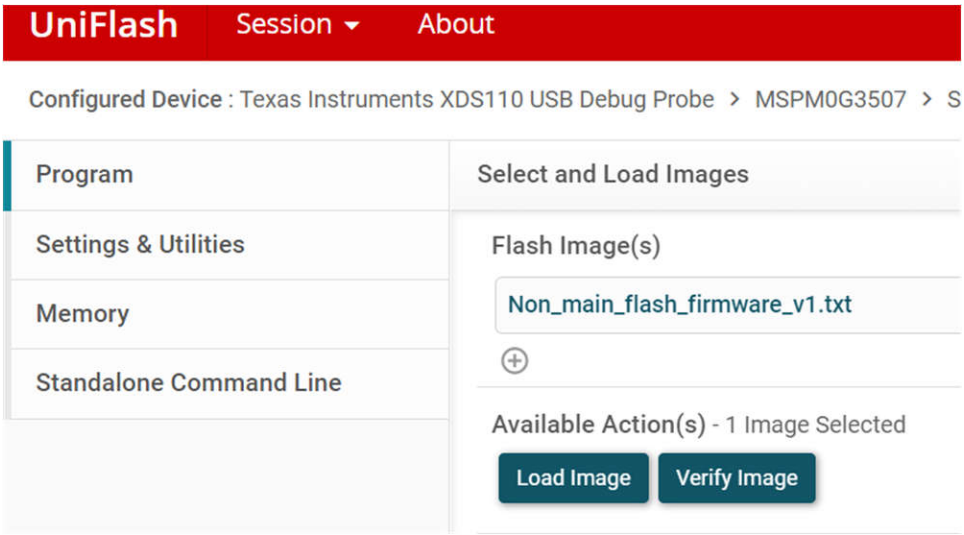

图 5-6. 下载非主闪存固件

5. 选择“Memory”选项，在“Address”字段中输入 0x41C00000，并从器件中读取数据，检查是否成功编程了新的配置数据。

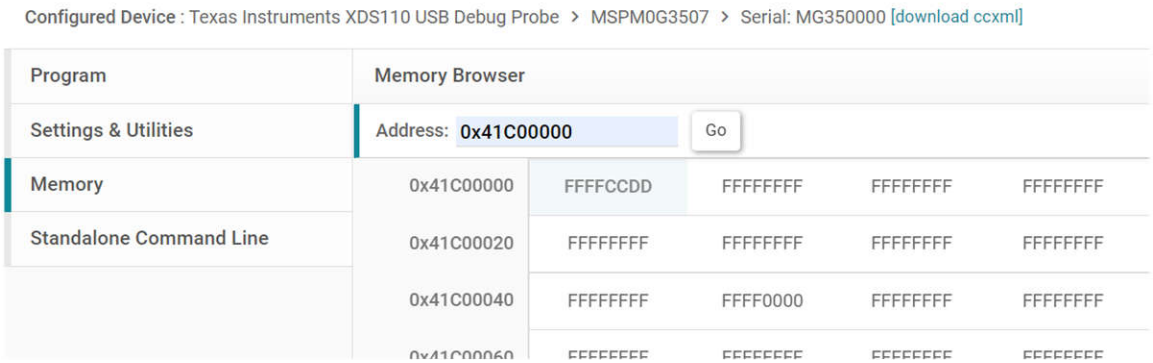

图 5-7. 验证存储器回读中的固件

6 LFU 引导加载程序协议

本节讨论引导加载程序协议，以及如何使用 UART 命令并完成固件升级过程。

6.1 数据包格式和内核命令

LFU 引导加载程序的数据包格式与基于 ROM 的 BSL 相同。有关更多详细信息，请参阅 *MSPM0 引导加载程序用户指南*。LFU 引导加载程序支持表 6-1 中列出的内核命令。

表 6-1. LFU BSL 内核命令

LFU BSL 命令	是否受保护	CMD	起始地址	数据（字节）	内核响应
CMD Unlock Bootloader	否	0x21	-	D1⋯D32（密码）	是
CMD Flash Range Erase	是	0x23	A1...A4	A1...A4（结束地址）	是
CMD Program Data	是	0x20	A1...A4	D1⋯Dn	是
CMD Start application	否	0x40	-	-	否

6.2 LFU 引导加载程序中的特殊命令

有一些特殊命令不同于基于 ROM 的 BSL 命令，它们是单字节命令，表 6-2 中列出了这些命令。

表 6-2. LFU BSL 特殊命令

LFU BSL 特殊命令	数据包	响应（封装格式）
获取应用状态	0x55	是（8 字节：0x51 + 应用状态标志（1 字节）+ 应用 1 区域起始地址（3 字节）+ 应用 2 区域起始地址（3 字节））
恢复引导加载程序任务	0xAA	是（1 字节：0xBB）

- **获取应用状态**命令：

  此命令用于获取应用代码执行状态，查看应用代码 1 或应用代码 2 是否已执行，或者器件中是否没有应用代码。表 6-3 描述了应用状态标志。

表 6-3. 应用状态标志

变量	值	说明
应用状态标志	0	没有正在运行的应用
	1	应用 1（应用 1 区域中的应用）正在运行
	2	应用 2（应用 2 区域中的应用）正在运行

  此命令还会返回应用空间 1 和空间 2 的起始地址。此信息可用于在开始固件升级过程之前进行系统完整性验证

- **恢复引导加载程序任务**命令：

  当器件正在执行应用代码时，引导加载程序任务被挂起，此命令用于在开始固件升级过程之前强制引导加载程序任务恢复到活动状态。

6.3 主机器件固件升级流程

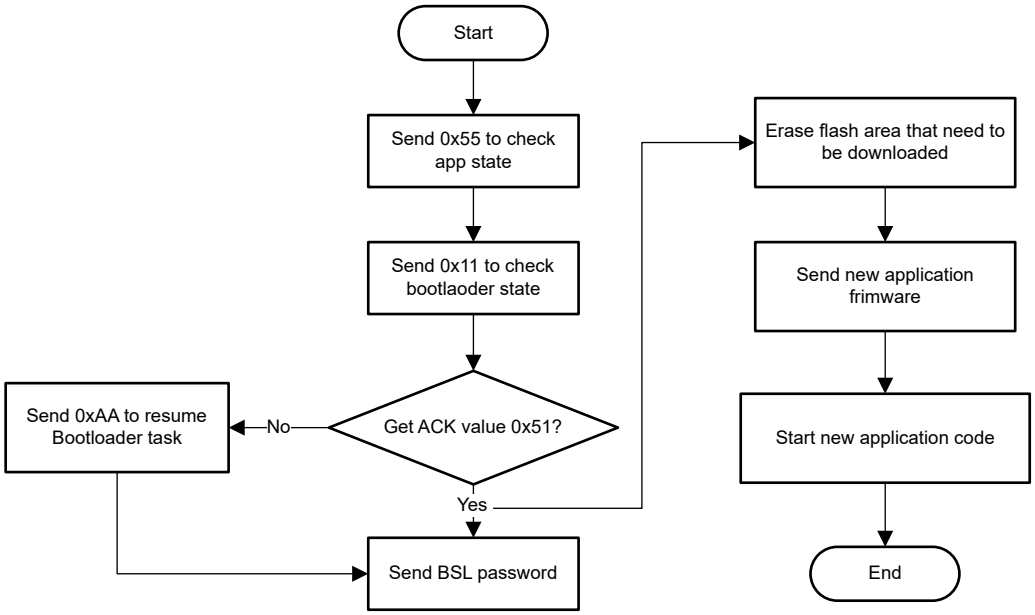

图 6-1. LFU 主机操作流程

在开始固件升级过程之前需要执行以下两个步骤：

1. 发送一字节命令 0x55 来检查应用代码执行状态，以便您知道哪些应用代码需要升级。
2. 发送一字节命令 0x11 来检查引导加载程序状态，如果器件响应 0x51，则表示引导加载程序处于活动状态，并准备好进行固件升级。否则，您需要先发送一字节命令 0xAA，以强制引导加载程序从挂起状态进入活动状态。

当您确定引导加载程序处于活动状态时，请先发送密码以解锁器件，然后擦除所需的闪存空间，并编写新的应用代码。

以下是发送应用代码的提示。

- 确保应用代码 .txt 文件是 16 字节对齐。主机软件负责用虚拟数据字节填充未使用的空间，确保 .txt 文件为 16 字节对齐，请参阅图 6-2 和图 6-3。

```
@4130
C6 E0 70 47 C4 E0 00 BF 00 BF 00 BF 00 BF 00 BF
00 BF 00 BF
```

图 6-2. 未对齐的固件

```
@4130
C6 E0 70 47 C4 E0 00 BF 00 BF 00 BF 00 BF 00 BF
00 BF 00 BF FF FF FF FF FF FF FF FF FF FF FF FF
```

图 6-3. 对齐的固件

- 最后发送新应用固件的第一行（固件中的前 16 个字节）。这是为了避免意外执行未完成的固件的情况。目前的引导加载程序只是通过检查前 8 个字节是否全为 0xFF 来检查应用固件是否存在，不会检查固件是否要完成。如果最后发送了固件的第一行，可确保整个固件已成功发送。

编写新的应用代码后，发送命令 CMD 启动应用以执行新的应用代码。

7 迁移到其他 MSPM0 器件

要将演示迁移到其他器件，需要对引导加载程序项目和应用项目进行修改。此演示中的 GUI 可以帮助生成应用项目和引导加载程序项目的链接器文件和 device.h 头文件。除此之外，还需要根据特定器件修改低级别外设配置（ti_msp_dl_config.c 和 .h 文件）和通信接口（uart.c 和 .h 文件）。

8 参考文献

- 德州仪器 (TI)：*MSPM0 Bootloader 用户指南*
- *MSPM0 G 系列 80MHz 微控制器技术参考手册*
- 德州仪器 (TI)：*MSPM0 引导加载程序 (BSL) 主机实现*

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做出任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 TI 的销售条款或 ti.com 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265