AN12327

使用二级引导程序进行固件更新

版本 3 — 2020 年 10 月 应用笔记

目录

1 简介

双映像（image）可靠更新是引导程序的重要功能。它确保至少有一个 image 可启动并在任何时候都能正常工作。如果发生任何意外，boot loader 程序会检测并使用先前的 image 作为可引导 image。

但是，LPC55xx ROM 引导程序尚不支持双 image 功能。本应用笔记在 LPC55xx 上实现了一个简单的双 image 更新示例。这对于用户在 LPC55xx 系列上实现双 image 引导程序非常实用。

1.1 词汇表

表 1 列出了文档中使用的缩写和首字母缩略词。

表 1. 词汇表

内容	描述
SBL	二级引导加载程序
DSBL	双映像二级引导加载程序
DSBL_APP	用于演示双映像引导加载程序功能并与 DSBL 配合使用的示例应用程序演示
MCUBOOT	NXP 统一引导加载程序解决方案，包括协议、PC 软件、文档等。它能够在整个产品生命周期中快速轻松地进行编程。有关详细信息，请参阅 MCUBOOT。
blhost	用于实现 MCUBOOT 协议的 PC 命令行界面 (CLI) 工具。它是 MCUBOOT 软件包的一部分。

2 功能实现

本节概述了双 image 内存布局的实现、启动流程以及应用程序 image 格式。

2.1 总览

为确保可靠更新，此处采用了双 image 布局。具体做法是是将 image 下载到一个称为接收区域的临时区域。在每次上电后。boot loader 程序检查（通过完整性检查）接收区域中的 image 。如果下载的新 image 的版本号高于当前 image ，DSBL（Dual Image Boot Loader，双 image 加载程序）会将 image 从接收区域复制到主区域。位于 image 中的版本标志采用两个区域中的最新版本。

总结一下：

- 接收区域
  - boot loader 程序将新代码下载到该区域
- 主区域
  - 始终存储从接收区域复制的正确 image

— DSBL 跳转到驻留在主区域中的 image（如果存在）。为此，image 起始地址必须位于主区域中。

该应用笔记中的通信接口是通过 UART 进行演示的，用户可以轻松地将通信接口扩展到其他接口，例如 I2C, SPI 等。通信协议遵循 NXP MCUBOOT 协议，与 LPC55xx ROM 兼容。此外， MCUBOOT 协议对用户友好，因为它们可以重复用 PC blhost 软件。

图 1 显示了 Flash 分区的概述。

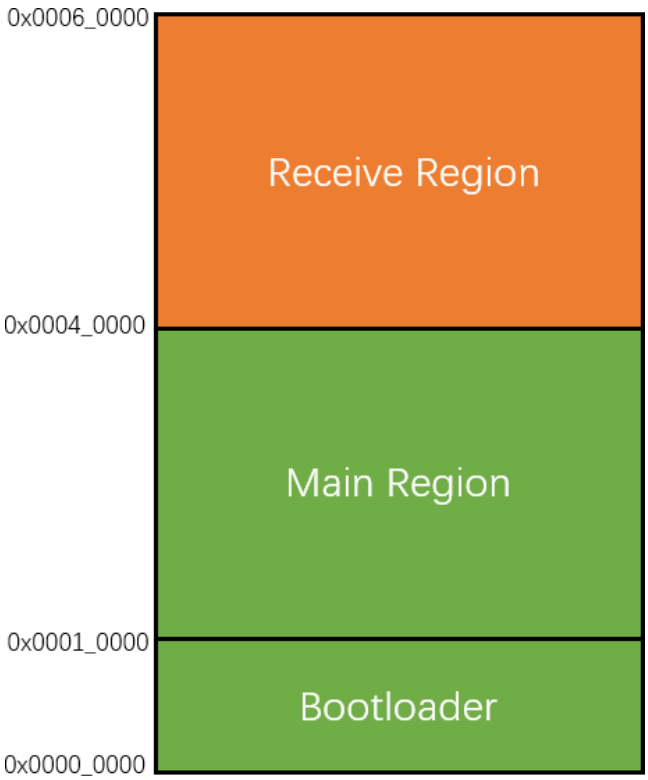

图 1. Flash 分区

2.2 启动流程

DSBL 用于管理 image 和引导应用程序。每次器件上电或发生复位时都会执行 DSBL 代码。图 2 显示了 DSBL 启动流程。

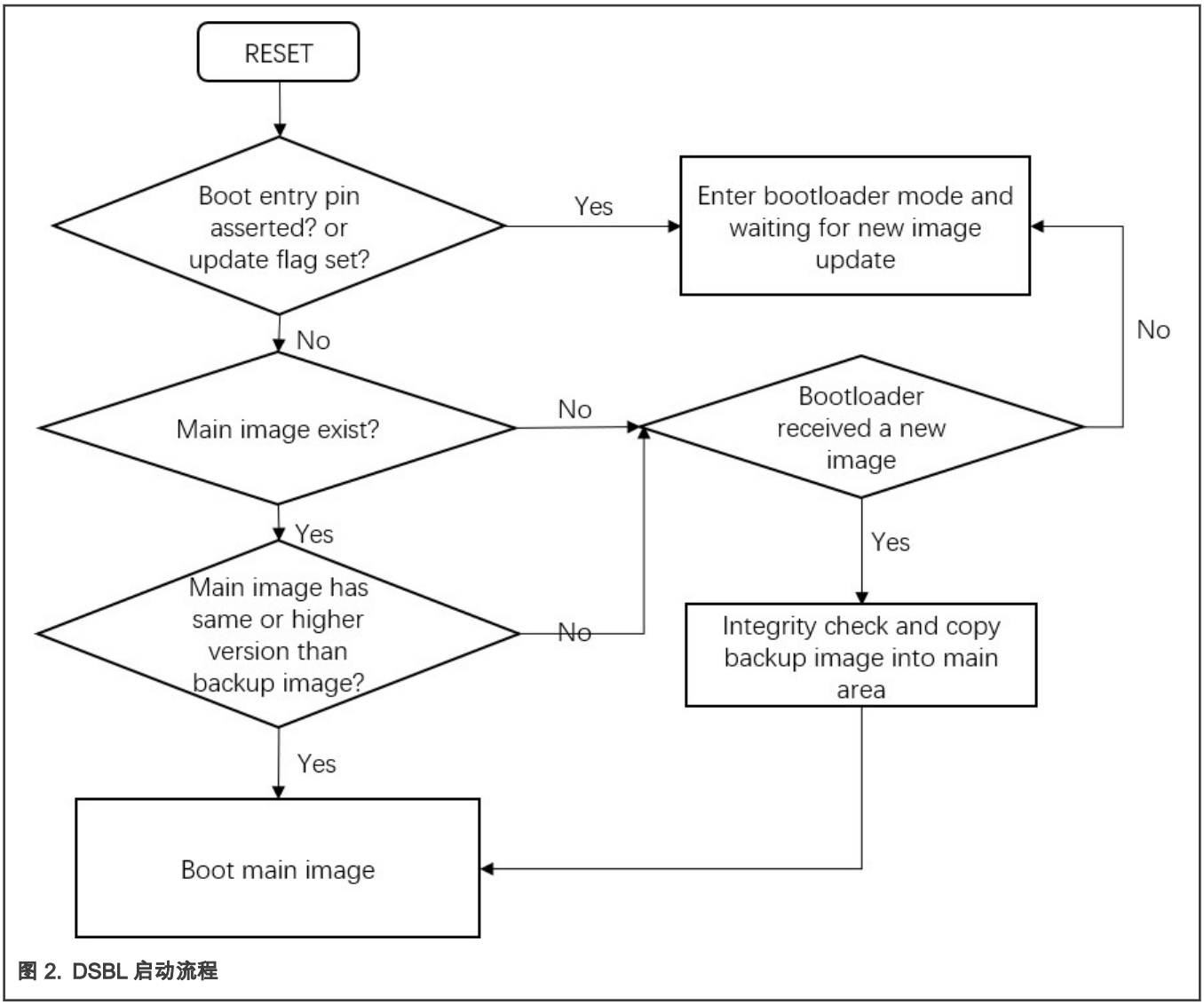

图 2. DSBL 启动流程

2.3 应用程序 image 格式

本节介绍 image 内存布局和 image 创建步骤。以 LPC55S69 为例，其他 LPC5500 系列遵循的步骤类似。

2.3.1 image 内存布局

图 3 显示了双 image 类型。它包含一个位于偏移 `0x24` 处的 image 标记。它还必须在偏移量为 `0x28` 的 image 中具有有效的 image 标头。image 的起始地址必须固定在主区域起始地址 `0x0001_0000`。image 标头本身可以位于 image 内的任何区域。在大多数情况下，image 标头放在向量表的末尾。对于 LPC55xx 系列，偏移量为 `0x140`。

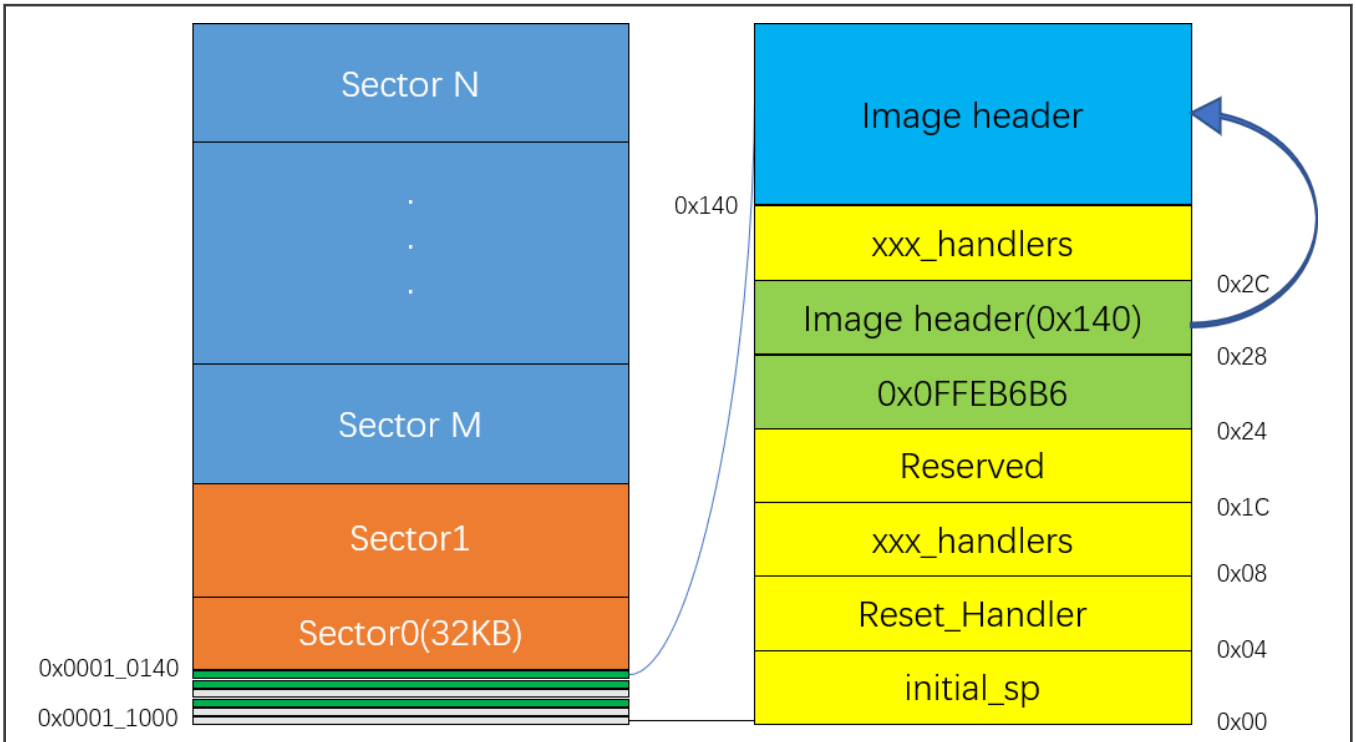

图 3. 应用程序镜像布局

image 标头本身是一个 24 字节的结构，如 表 2 中所列。

表 2. image 标头结构

偏移量	描述
0x00	标头创建者设立为 `0xFEEDA5A5`
0x04	映像类型 (NORMAL = 0 or NO_CRC = 1)
0x08	保留
0x0C	映像长度 长度应为实际长度，4，如果 CRC 值字段在长度内。
0x10	CRC 值
0x14	版本

对于 LPC55xx 部件，整个 image 二进制的 CRC32 值被添加到 image 头中。外部工具 *image_generator.exe* 文件有用于在 image 头中添加 image 二进制文件。

2.3.2 image 创建

本节介绍在 IDE 中修改启动文件和使用外部工具在 image 头中添加长度和 CRC 值的步骤。

2.3.2.1 在 IDE 中修改启动文件

添加 image 标记和 image 头是通过修改启动文件来完成的。

- IAR

注意

将 image 头放在向量表的末尾

```
                DCD     NMI_Handler
                DCD     HardFault_Handler
                DCD     MemManage_Handler
                DCD     BusFault_Handler
                DCD     UsageFault_Handler
__vector_table_0x1c
                DCD     0                       ; Checksum of the first 7 words
                DCD     0xFFFFFFFF              ; ECRP
                DCD     0x0FFEB6B6              ; Enhanced image marker, set to 0x0FFEB6B6 for deImage boot
                DCD     __deimage_header        ; Pointer to enhanced boot block, set to 0x0 for legacy boot
                DCD     SVC_Handler
                DCD     DebugMon_Handler
                DCD     0
                DCD     PendSV_Handler
                DCD     SysTick_Handler


                DCD     SMARTCARD1_IRQHandler  ; Smart card 1 interrupt
__deimage_header
                DCD     0xFEEDA5A5              ; Image marker
                DCD     0x00000000              ; Image type Normal: 0, NO CRC: 1
                DCD     0x00000000              ; Reserved
                DCD     0x00000000              ; Image length
                DCD     0x00000000              ; CRC value
                DCD     0x00000001              ; Version

                AREA    |.text|, CODE, READONLY
```

图 4. 在 IAR 中添加映像标记和映像标题

- KEIL

注意

把 image 头放在向量表末尾。

```
        DATA

__vector_table
        DCD     sfe(CSTACK)
        DCD     Reset_Handler

        DCD     NMI_Handler
        DCD     HardFault_Handler
        DCD     MemManage_Handler
        DCD     BusFault_Handler
        DCD     UsageFault_Handler
__vector_table_0x1c
        DCD     0
        DCD     0xFFFFFFFF ; ECRP
        DCD     0x0FFEB6B6 ; Single Enhanced Image Flag
        DCD     __ImageMarker
        DCD     SVC_Handler
        DCD     DebugMon_Handler
        DCD     0
        DCD     PendSV_Handler
        DCD     SysTick_Handler

        DCD     SMARTCARD1_IRQHandler  ; Smart card 1 interrupt
__ImageMarker
        DCD     0xFEEDA5A5 ; Image Marker
        DCD     0x0 ; Image Type Normal: 0, NO CRC: 1
        DCD     0x0 ; Reserved
        DCD     0x0 ; Image Length
        DCD     0x0 ; CRC Value
        DCD     0x2 ; Version
__Vectors_End

__Vectors       EQU   __vector_table
__Vectors_Size  EQU     __Vectors_End - __Vectors
```

图 5. 在 Keil 中添加映像标记和映像标题

2.3.2.2 使用外部工具在 image 头中添加长度和 CRC 值

当 image 头中的 image 类型字为 0x00（NORMAL）时，image 需要外部工具将长度和 CRC 值添加到 image 头中。位于以下位置的工具文件夹中的 image_generator.exe 文件有助于将长度和 CRC 值添加到 image 头中。

```
\boards\<board_name>\dual_sbl\lpc55xx_dsbl_app\cm33_core0\tools
```

双击 post_build.bat，脚本调用 image_generator.exe，在该文件夹下生成名为 dsbl_app_crc.bin 的二进制文件。将 .bin 文件 image 下载到接收区域。有关如何使用这些工具的分步指南，请参阅运行 demo 的步骤。

3 演示

Demo 中有两个基于 SDK 的工程，详见 表 3。

表 3. 演示项目描述

项目名称	SDK 中位置	描述
`lpc55xx_dsbl`	*\boards\lpcxpresso55s69\dual_sbl*	双映像辅助引导加载程序项目
`lpc55xx_dsbl_app`	*\boards\lpcxpresso55s69\dual_sbl*	演示应用项目

- lpc55xx_dsbl 为 lpc55xx 双 image 辅助 boot loader 程序，它将在启动时执行。该程序将处理信任区配置、与 PC 主机的通信、image 检查和复制任务。这是您需要下载到 EVK 板的第一个项目。
- lpc55xx_dsbl_app 代表 lpc55xx 双 image 辅助 boot loader 程序应用示例，几乎与 `hello_world` 相同。 区别在于：
  1. 该 image 有一个 image 标记和一个位于矢量表之后的 image 头。 因此，它可以通过 DSBL 进行区域化。
  2. 将链接器起始地址从 `0x0000_0000` 修改为 0x0001_0000 以将加载/起始地址放入主 image 区。

3.1 硬件设置

硬件使用 LPC55S69 EVK 板，如 图 6 所示。请确保您已阅读板用户指南并熟悉板的基本功能，例如 Reset 按钮和调试连接器的位置等。

本演示使用 Debug 和 UART USB 连接器 (1) 作为调试接口和 UART。

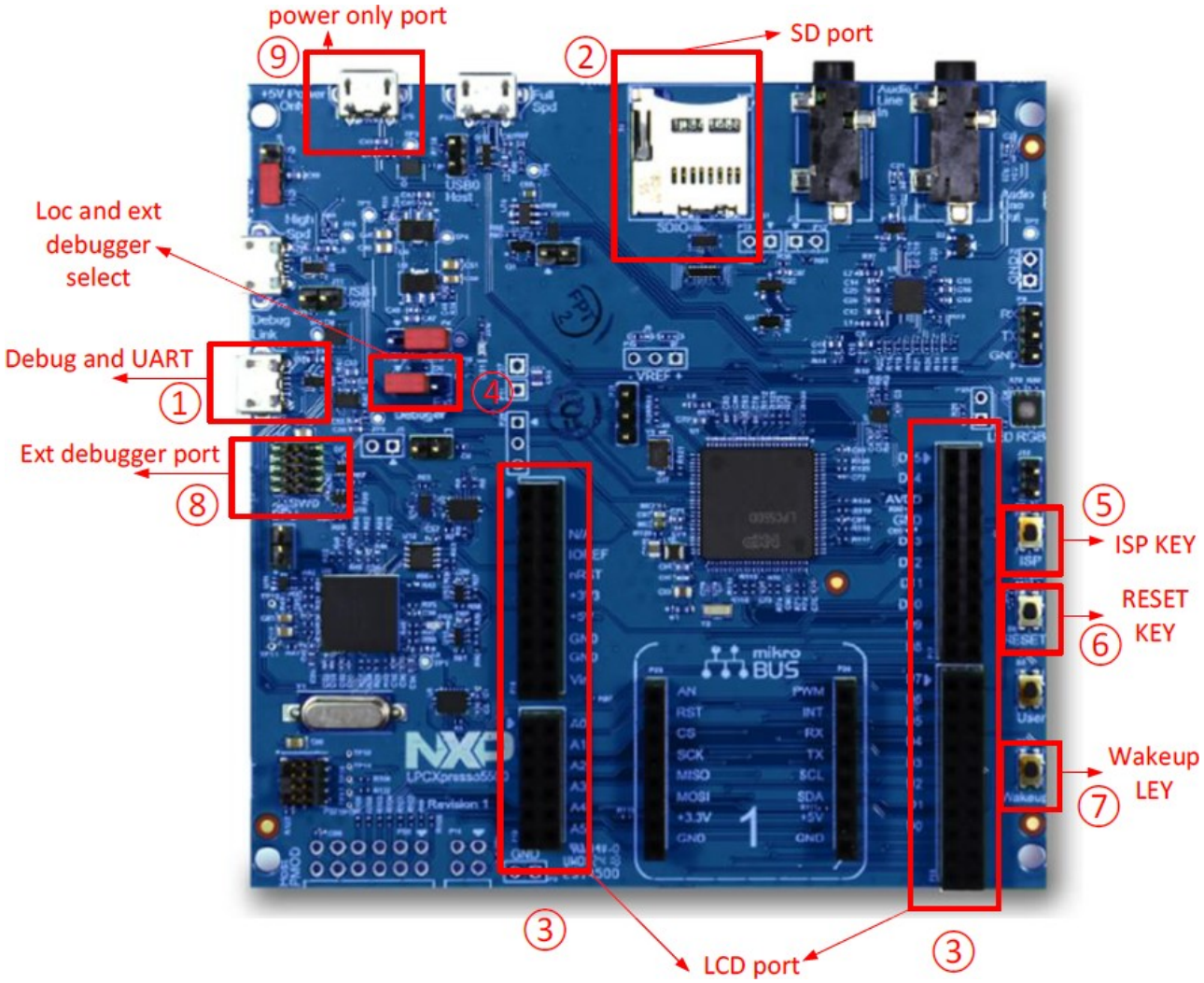

图 6. LPC55S69 EVK 板

USB 桥接器和 WAKEUP 按钮 (7) 用作 boot loader 程序的选择引脚。

硬件使用 LPCXpresso55S16 板卡，如 图 7 所示。确保您已阅读板卡用户指南并熟悉板卡的基本功能，例如 RESET 按钮和调试连接器的位置等。本演示使用 Debug 和 UART USB 连接器 (J1) 作为调试接口和 UART-USB 桥接器。此外，WAKEUP 按钮 (SW1) 用作 boot loader 程序的入口引脚。

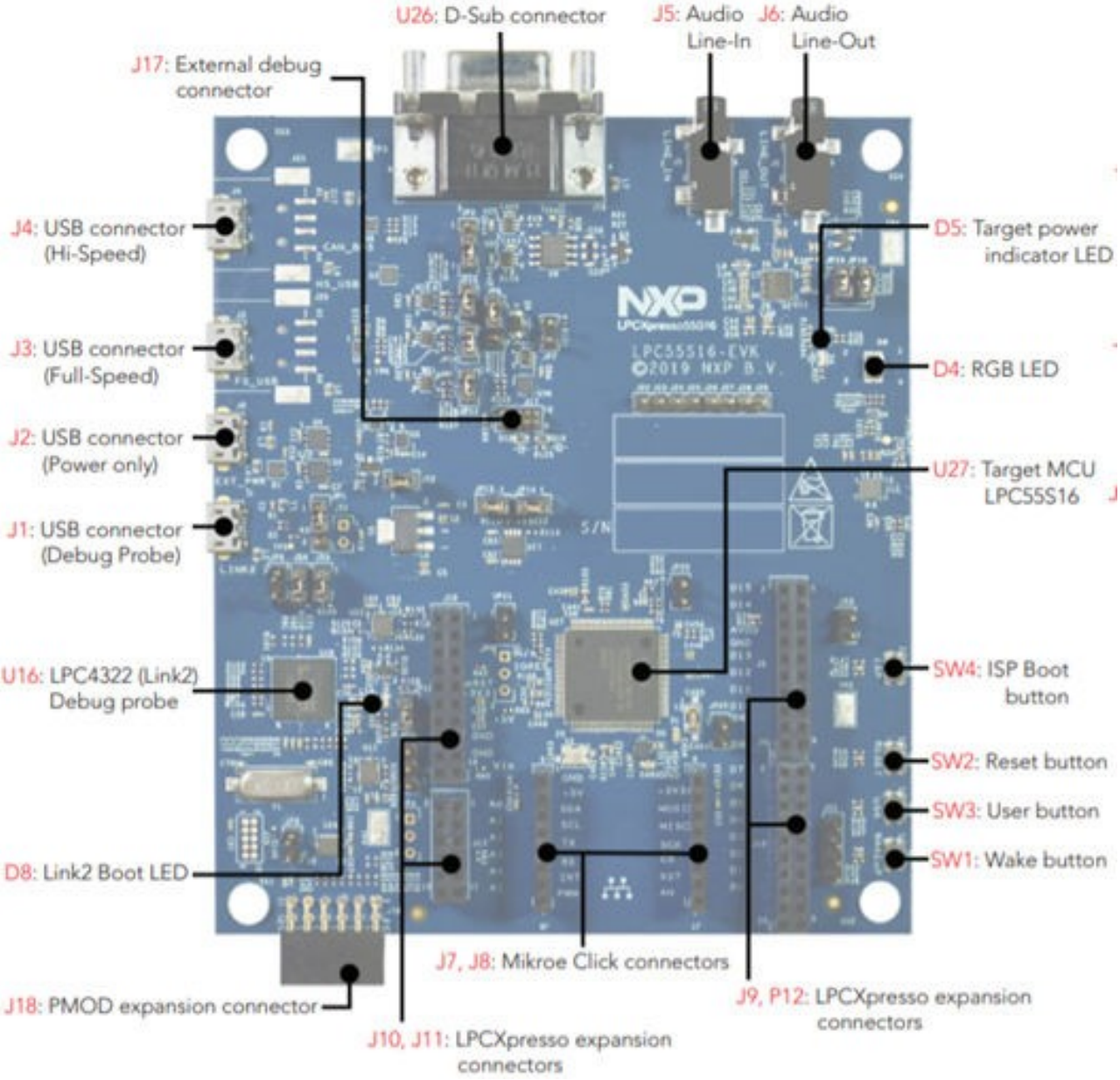

图 7. LPCXpresso55S16 板

其他 LPC55xx 板具有类似的设置步骤。有关详细信息，请参阅 EVK 板用户指南。

3.2 运行 demo 的步骤

注意

确保您对 LPC5500 系列 EVK 板有基本的了解，已安装相关的 LPC-Link II 调试器驱动程序，已成功运行 SDK 文件夹中的 hello_world 示例，并已验证与 PC 的 UART 通信。

1. 将 USB 与 Debug 和 UART USB 连接器 (1) 连接到板上并建立调试和 UART 连接。
2. 打开、编译和下载 lpc55xx_dsbl 项目。使用 115200-N-8-N-1 打开您的串行终端。
3. 按住唤醒按钮 (7)，然后按下 RESET 按钮。这会强制 DSBL 进入 boot loader 程序模式。在此模式下，DSBL 不会启动任何应用程序，而是等待 UART 连接。

4. 默认情况下，lpc55xx_dsbl 启用调试日志。终端提供 图 2 信息，说明 DSBL 运行成功，进入 boot loader 模式。
5. 打开并编译项目：lpc55xx_dsbl_app。不要使用 IDE 下载 lpc55xx_dsbl_app 项目。否则，演示 boot loader 程序功能是没有意义的。

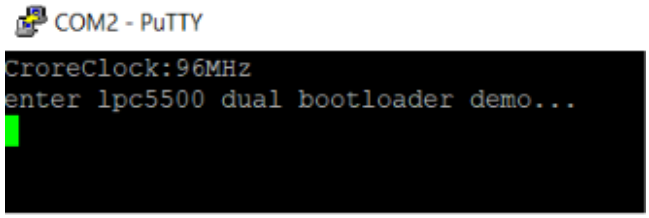

图 8. DSBL 进入引导加载程序模式

6. 打开\boards\<board_name>\dual_sbl\lpc55xx_dsbl_app\cm33_core0\tools 文件夹并双击 post_build.bat。这将生成 dsbl_app_crc.bin，它将 CRC 和 image 长度信息添加到 image_generator.exe。

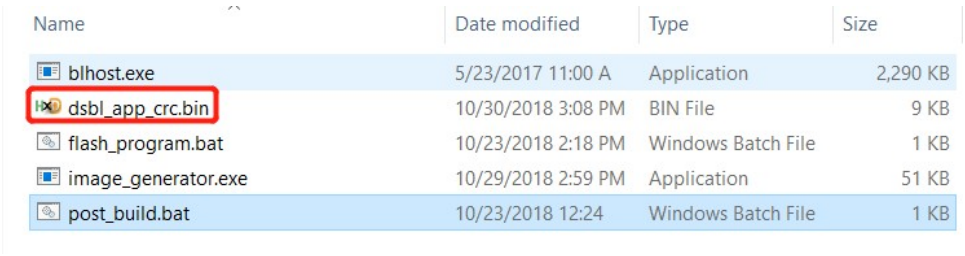

图 9. 使用 post_build.bat 生成 dsbl_app_crc.bin

dsbl_app_crc.bin 是要在接收区域下载的二进制 image。

7. 关闭串口终端。打开 bash 窗口或命令窗口并执行 flash_program.bat。此脚本调用 blhost.exe 并下载 Receive 区域中的 dsbl_app_crc.bin。运行 flash_program.bat 需要两个参数：UART COM 索引和应用程序 image 的全名。
8. 随着脚本的执行，新的 image 被下载到接收区域。

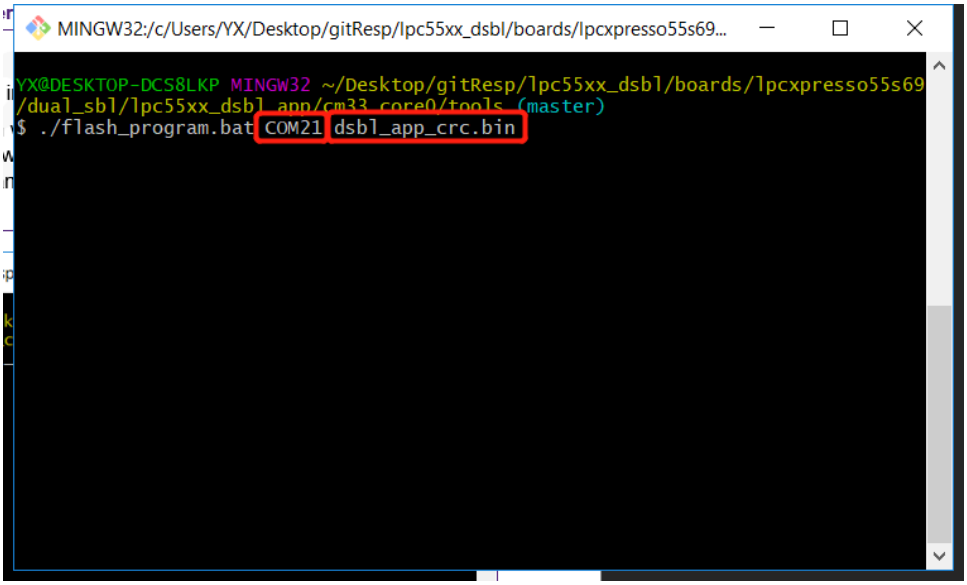

图 10. 下载 image

9. 重新打开 UART 终端并按下 RESET 按钮。

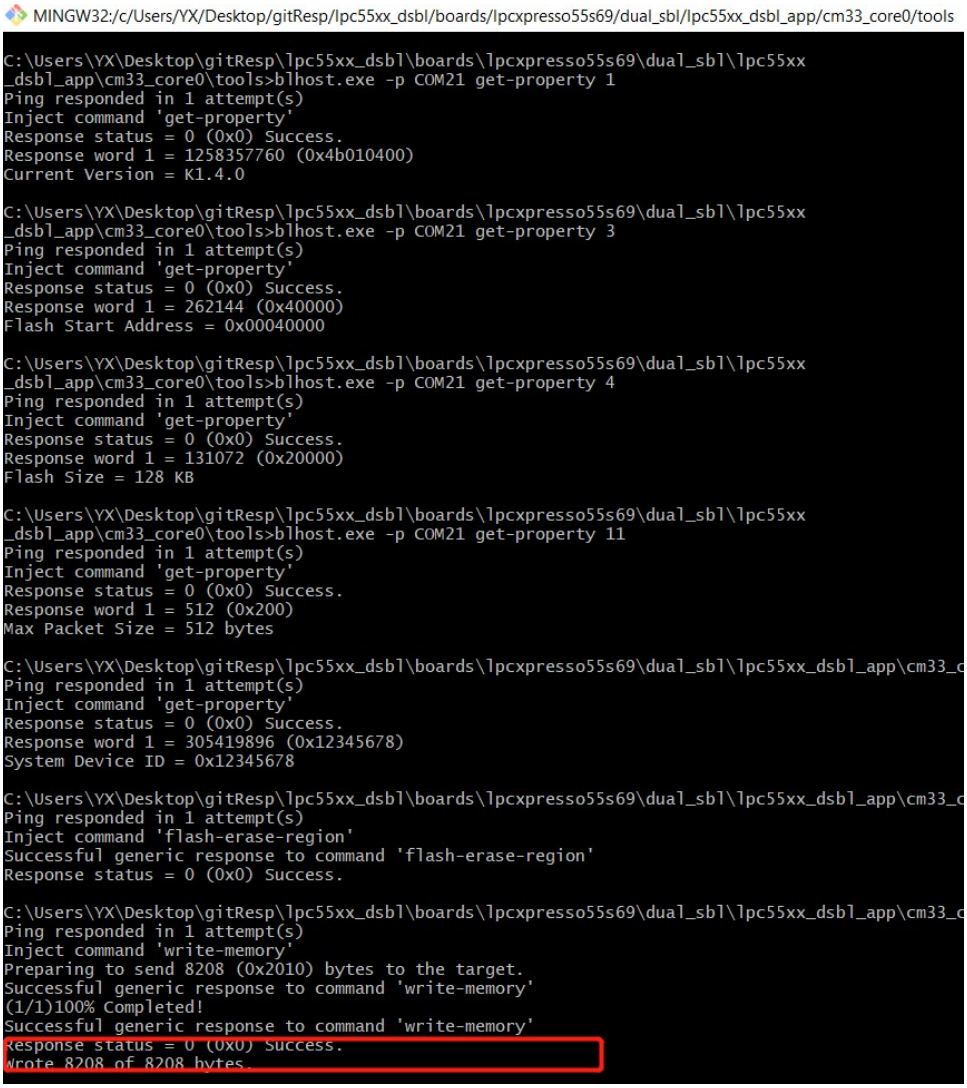

图 11. flash_program.bat 的下载日志

找到的日志“image found：0x00040000”表示 DSBL 已检测到接收区域中存在 image。由于主区域没有任何有效的 image，DSBL 处理接收主区域中的 image 并引导它。

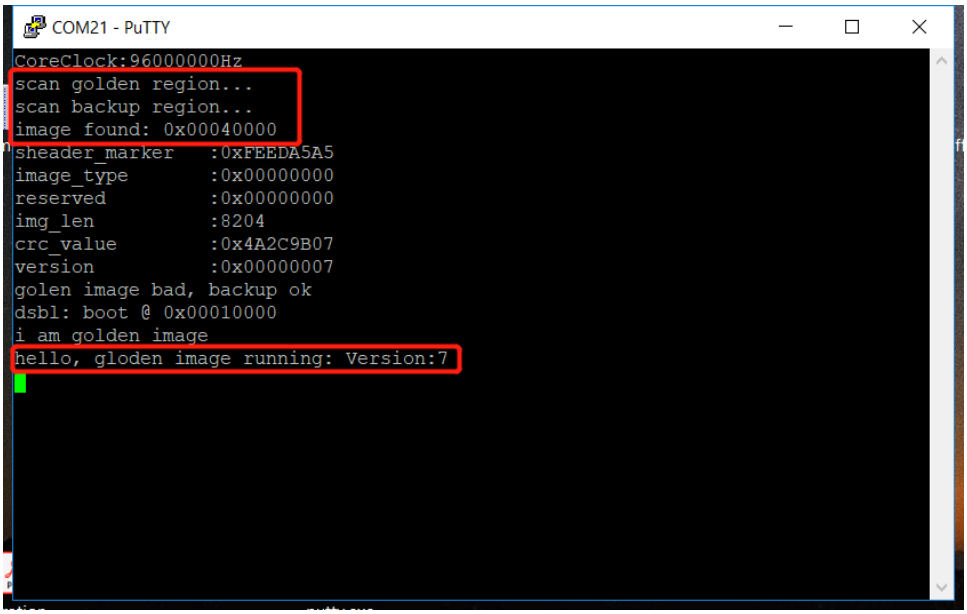

图 12. 将镜像复制到主区域并启动

日志 “hello, main image running, Version: 7” 表示主 image 已经运行。

3.3 重新进入 DSBL 的方法

除了使用唤醒按钮进入 DSBL 之外，还有两种方法可以进入 DSBL 以进行应用程序更新。

3.3.1 重新调用

在您的应用程序中定义 sbl_api 结构，如 图 13 所示。然后，调用 re_invoke。

调用 re-invoke 强制 CPU 立即跳转到 DSBL，就像在传统 LPC 部分中重新调用 ROM_API 一样。

```
typedef struct
{
    void (*reinvoke)(void);
    void (*set_update_flag)(void);
    void (*test)(void);
}sbl_api_t ;

static  sbl_api_t  *sbl_api  = (sbl_api_t *)(0x400);
```

图 13. DSBL API 结构

3.3.2 Set_update_flag

与 reinvoke 不同，该 API 不会立即进入 DSBL。它让 DSBL 在下下次上电时进入更新模式。 由于设置了非易失性更新标志。DSBL 计算更新失败次数。如果接收区域中的 image 更新失败超过三次，则 DSBL 清除 update_flag 并启动主 image。否则，在每次重新启动时，DSBL 不会启动主 image 并等待成功下载。

```
sbl_api->set_update_flag();
```

3.4 修改应用 image

更新应用程序 image 版本信息很简单。您只需要修改 image 标题中的版本字即可。

注意

仅当接收到的 image 的版本号高于主 image 时，DSBL 才会将接收到的 image 复制到主区域。

```
        DCD     PVTVF1_RED_IRQHandler  ; PVT interrupts
__deimage_header
                DCD     0xFEEDA5A5              ; Image marker
                DCD     0x00000000              ; Image type Normal: 0, NO CRC:
                DCD     0x00000000              ; Reserved
                DCD     0x00000000              ; Image length
                DCD     0x00000000              ; CRC value
                DCD     0x00000007              ; Version
__Vectors_End
```

图 14. 更新 DSBL_APP 镜像版本信息

4 总结

本节提供有关闪存读取操作、UART 多路复用以及启用或禁用调试日志的步骤信息。

4.1 Flash 读操作

大多数情况下，AHB 总线直接读取 Flash。但是，在 LPC55xx 中，由于 Flash ECC 机制，任何直接读取已擦除闪存（已擦除但未写入）的尝试都会导致 Hard Fault。该限制不方便 boot loader 程序开发。为了解决这个问题，代码实现了一个 Non-AHB 方法来读取 flash 数据的 API 来代替 AHB 总线的直接读取。Non-AHB 方法读取 Flash 数据 API 的代码在 memory.c 中。检查代码以获取详细信息。

4.2 UART 复用

在此演示中，三个函数使用相同的 UART：

1. DSBL 调试日志输出
2. 应用演示日志输出
3. DSBL 下载图片的通讯接口

因此，存在 UART 多路复用冲突问题。每当使用 blhost 下载图像时，必须关闭 UART 终端以释放 PC COM 端口资源供 blhost 使用。

4.3 启用/禁用调试日志

DSBL 调试日志由宏启用和禁用。在 dimage.h 中注释宏 DIMAGE_DEBUG 会禁用所有调试输出。参见 图 15。

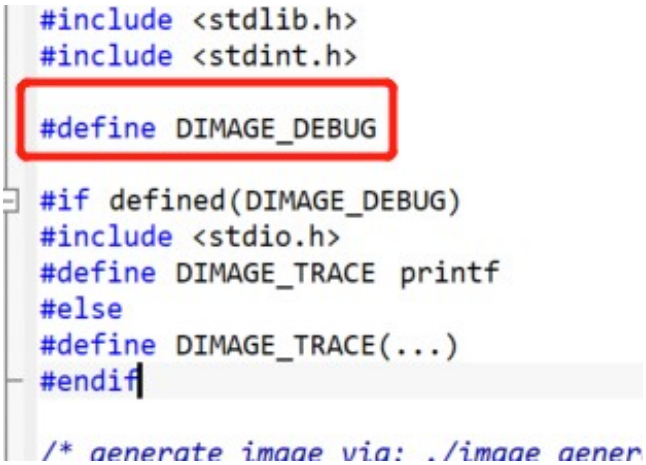

图 15. 更新 DSBL_APP 镜像版本信息

5 修订记录

表 4 总结了自初始版本以来的修订记录。

表 4. 修订记录

版本号	日期	说明
0	2019 年 1 月 23 日	初次版本
1	2020 年 2 月 26 日	更新了运行 demo 的步骤和其他一般更改
2	2020 年 5 月 21 日	更新 图 8
3	2020 年 10 月 30 日	用 LPC5500 系列替换 LPC55S16

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Limited warranty and liability — Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Right to make changes - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 2020 年 10 月

Document identifier: AN12327

arm