

i.MX RT Flashloader 用例

URL: <https://www.nxp.com/docs/en/nxp/application-notes/AN12238.pdf>

1. 简介

The i.MX RT Flashloader 是一个独立的，完整的软件工具，用于 i.MX RT 系列 MCU 产品的开发和工厂制造。它包括运行在 MCU RAM 中的 Flashloader 二进制文件和与 Flashloader 二进制文件通信的 PC 主机工具。它支持快速和简单的烧写内部 OCOTP (eFuse) 和外部 NOR/NAND/HyperFlash 设备。主机端命令行和 GUI 工具可通过外设接口(USB-HID 或 UART)与 Flashloader 二进制文件通信。

本档中用于示例的 Flashloader 为 Flashloader_RT1050_1.1。硬件平台为 MIMXRT1050-EVKB 板。

目录

1. 简介.....	1
2. i.MX RT1050 Flashloader.....	2
2.1. 获取 i.MX RT1050 Flashloader.....	2
2.2. Flashloader 包.....	2
3. i.MX RT1050 OCOTP 和外部 flash.....	4
3.1. OCOTP (eFuse).....	4
3.2. 外部 flash.....	5
3.2.1. 可引导映像.....	6
3.2.2. 从外部 flash 启动.....	7
4. i.MX RT1050 Flashloader 用例.....	8
4.1. 目标平台环境.....	8
4.2. 串行下载模式.....	11
4.2.1. SDPHost 下载 Flashloader 映像.....	11
4.3. 烧写 OCOTP (eFuse).....	13
4.3.1. 烧写 boot mode eFuse 以使能 SD boot.....	14
4.3.2. 烧写 FlexRAM eFuse.....	14
4.4. 构建可引导映像.....	15
4.5. 烧写外接 flash 设备.....	17
4.5.1. Mfgtool.....	17
5. 结语.....	19
6. 修订历史.....	19

2. i.MX RT1050 Flashloader

2.1. 获取 i.MX RT1050 Flashloader

NXP 在官网提供 Flashloader 包。[从此](#)下载 i.MX RT1050 MCU 和 MIMXRT1050-EVK 板的 Flashloader 包。



图 1. 下载 i.MX RT1050 Flashloader

注意

不同的 MCU 平台有不同的 Flashloader 包，它们不能互换使用。确保为特定的 MCU 平台下载正确的 Flashloader 包。

2.2. Flashloader 包

Flashloader 包中的所有文件和工具一起工作，以实现这些功能：

1. 与 MCU 的 BootROM 通信并下载 Flashloader 映像。
2. 创建一个可引导映像 (SB 文件)。
3. 对 MCU 内部的 OCOTP (eFuse) 进行编程，以定义 Boot 模式、MAC 地址、安全模式等。
4. 将可引导的映像 (SB 文件) 烧写到 MCU 的外部闪存 (Nor/NAND/HyperFlash/SD)。

这是 Flashloader 包解压后的目录结构：

```

└─Flashloader_RT1050_1.1
  │ LA_OPT_Base_License.htm
  │ SW_Content_Register_Kinetis_Bootloader.txt
  └─doc
    └─example_images
      └─Flashloader
        └─Tools
  
```

```

├──bd_file
├──blhost
├──elftosb
├──Mfgtools-rel
└──sdphost

```

表 1 显示有关 Flashloader 目录和文件的详细信息。

表 1. Flashloader 目录和文件

LA_OPT_Base_License.htm	NXP 软件许可协议
SW_Content_Register_Kinetis_Bootloader.txt	Flashloader 的发布信息和软件内容
doc\	此目录包括所有文档： <ul style="list-style-type: none"> • <i>i.MX MCU Manufacturing User's Guide.pdf</i> • <i>MXRT1050 Flashloader v1.1.0 Release Notes.pdf</i> • <i>Kinetis blhost User's Guide.pdf</i> • <i>Kinetis SDPHost User's Guide.pdf</i> • <i>MCUX Flashloader Reference Manual.pdf</i>
example_images\	此目录包括可执行映像实例，Flashloader 工具可以使用它们来验证基本流程。
Flashloader\	此目录包括发布的 Flashloader 可执行映像。可以将其下载到目标设备并实现支持的特性。
Tools\bd_file\	此目录包含 i.MX RT1050 平台示例 BD 文件。BD 文件就是启动描述文件，elftosb 工具使用 BD 文件来控制可执行映像中启动命令的执行顺序。
Tools\blhost\	此目录包含可运行在 Windows/MAC/Linux 主机系统中的 blhost 工具，blhost 是一个命令行工具，用来和运行在目标板上的 Flashloader 通信，其通信接口可以是 UART（串行接口）或者 USB。并实现对 eFuse 和外部闪存的烧写功能。
Tools\elftosb\	此目录包含可运行在 Windows/MAC/Linux 主机系统中的 elftosb 工具。elftosb 用于生成带 Boot 命令的用户映像，也就是 SB（安全二进制文件）文件。
Tools\Mfgtools-rel\	此目录包含一个图形化 Mfgtool 工具和一些配置文件。Mfgtool 可以用来对外部闪存进行烧写。
Tools\sdphost\	此目录包含可运行在 Windows/MAC/Linux 主机系统中的 sdphost 工具。sdphost 采用命令行的方式支持 SDP 串口下载协议，实现 PC 主机与串行下载模式下的 NXP i.MX 器件的通信。sdphost 对于工厂生产中的程序烧写特别有用，可被其他应用程序或者自动化测试软件调用，以适应大规模生产中的开发测试环境。

3. i.MX RT1050 OCOTP 和外部 flash

Flashloader 的关键功能之一是 OCOTP(eFuse)操作和外部 flash 编程。下面的小节简单介绍 Flashloader 和 OCOTP。更多细节, 请参阅 i.MX RT1050 Processor Reference Manual (文件 [IMXRT1050RM](#))。

3.1. OCOTP (eFuse)

OCOTP (片上一次性可编程) 存储器, 又名 eFuse, 是芯片中的一种特殊的存储模块。任何 eFuse 位都可以从 0 到 1 只编程一次(熔融), 但是读操作没有限制。它包含了整个芯片配置。以下是一些关键配置:

- 引导模式。
- MAC 地址。
- FlexRAM 设置。

有关使用 Flashloader 的 eFuse 编程示例, 请参阅 [4.3 章, “烧写 OCOTP \(eFuse\)”](#)。

eFuse 内存空间不在系统的 4G 地址空间内, 所以正常的地址读/写不能被用来访问 eFuse 寄存器。需要 Flashloader 的支持来读/写一个特定的 eFuse 寄存器。

图 2 中的 OTP 内存定义情况是按可锁定区域来分组的。

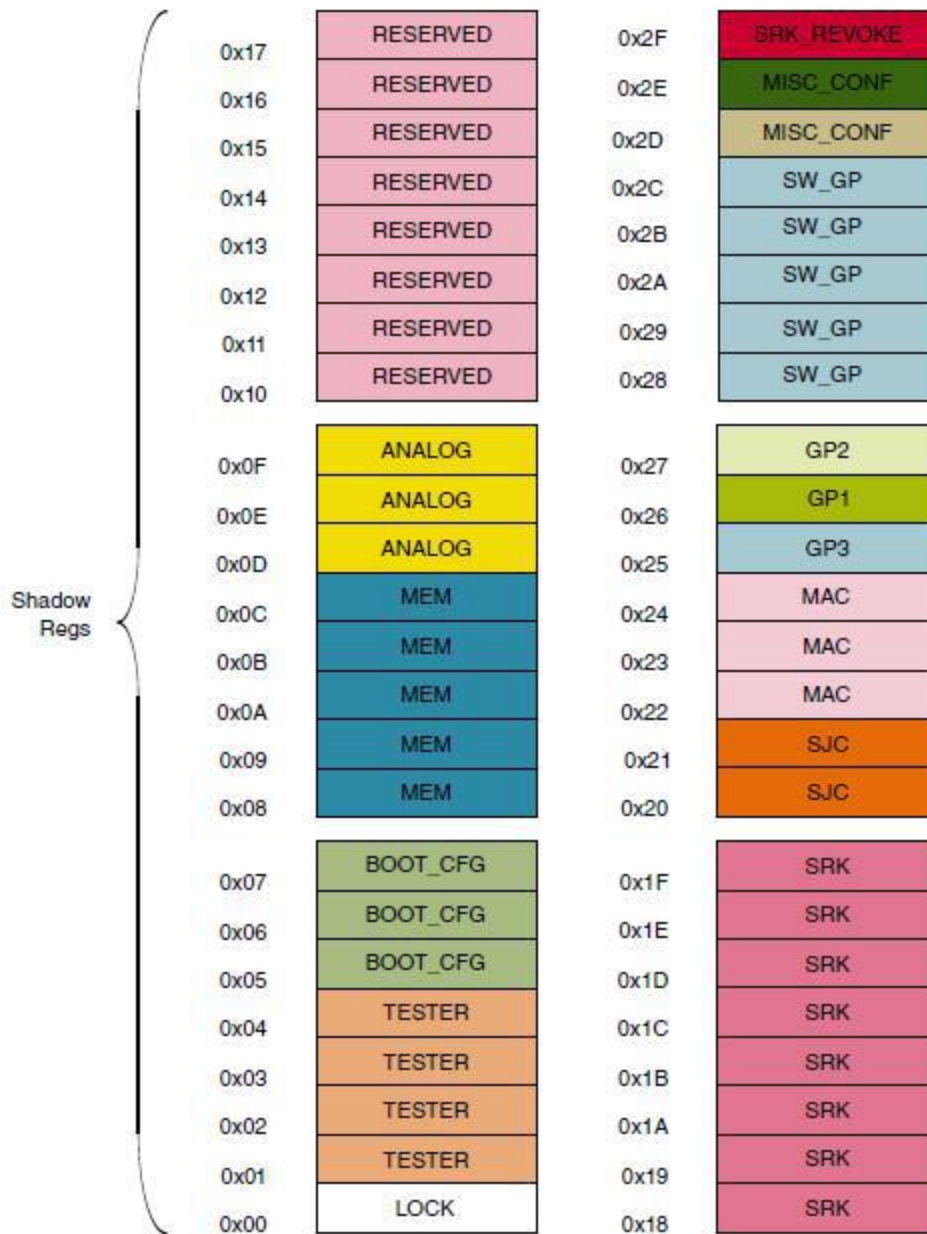


图 2. OTP 内存定义

3.2. 外部 flash

i.MX RT1050 设备提供多种外部闪存接口:

- 8/16 位 SLC NAND FLASH, 带软件可处理的 ECC.
- SD/eMMC.
- HyperFlash.
- 支持 XIP 的并行 NOR FLASH .
- 支持 XIP 的单/双通道四线 SPI FLASH

外部 flash 可用于存储应用程序映像并使 i.MX RT1050 从 flash 映像启动。Flashloader 包括各种 flash 编程算法，以支持在开发和制造阶段的 flash 映像烧写。

3.2.1. 可引导映像

对于 i.MX RT1050 平台, 应用程序映像必须存储在外部 flash 中。这和内部有并行 NOR flash 的 MCU 不同。内部并行 NOR flash 空间会被分配在系统 4G 地址空间内，因此可以通过地址直接访问。core 可以直接获取引导映像并就地运行(XIP)。

芯片电源复位后，i.MX RT1050 中的 BootROM 总是先运行。它检查引导模式并帮助 core 从特定的外部闪存启动。

对于不同的 flash 接口和引导模式，BootROM 必须从外部 flash 设备中的应用程序映像获得一些额外信息。通过将附加的必要信息与应用程序映像相结合，会得到最终可烧写的引导映像。

其它必要信息包括：

- FCB (Flash Configuration Block):
 - 可选项 (用于串行/并行 NOR FLASH).
 - 偏移: 0x0000.
 - 说明: 定义外部 flash 接口的结构。
- IVT (Image Vector Table):
 - 必选项.
 - 偏移: 0x0400 (non-XIP flash)/0x1000 (XIP flash).
 - 说明: 该结构包括应用程序、DCD、BD 和 CSF 的地址信息。
- BD (Boot Data):
 - 必选项.
 - 偏移: 0x0420 (non-XIP)/0x1020 (XIP).
 - 说明: 该结构包括 SB 映像的起始地址和大小.
- DCD (Device Configuration Data):
 - 可选项.
 - 偏移: 在 IVT 中定义
 - 说明: 当前用于配置 SDRAM (SEMC 接口)。
- 应用程序二进制文件:
 - 必选项.
 - 偏移: 0x2000 (典型值).
 - 说明: 纯应用程序二进制文件.
- CSF (Command Sequence File):
 - 可选项.

- 偏移: 在 IVT 中定义
- 说明: 由 High-Assurance Boot (HAB)使用
- KeyBlob:
 - 可选项.
 - 偏移: 在 IVT 中定义.
 - 说明: 安全启动密钥信息

Flashloader 中的 elftosb 工具可以用来创建可引导的映像。另外还提供了一些 BD 示例文件。图 3 显示了可引导的映像布局和每个块的功能。

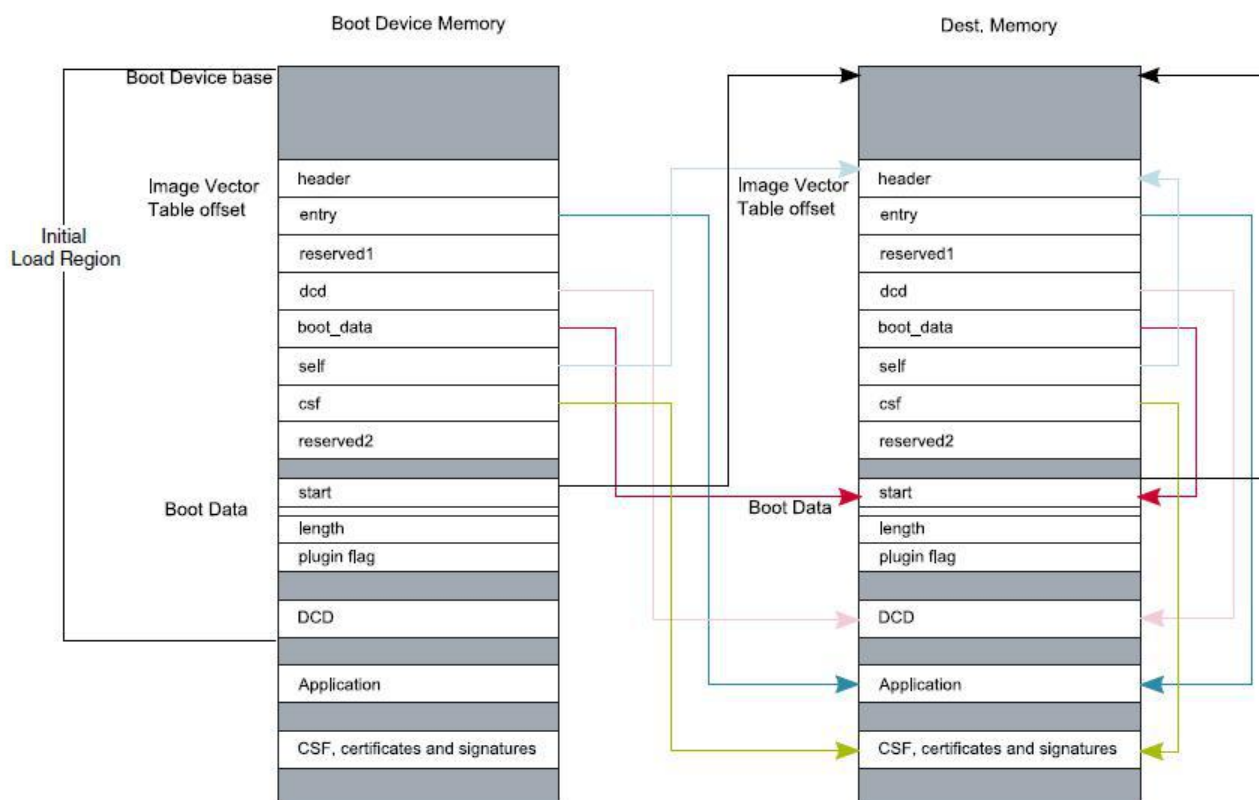


图 3. 可引导映像布局

3.2.2. 从外部 flash 启动

有了 BootROM, i.MX RT1050 可以在 XIP(NOR flash)或非 XIP 模式下从各种外部闪存启动。根据可引导映像中的 IVT 和 BD 信息, BootROM 直接启动应用程序二进制文件(XIP), 或者将可引导映像复制到 RAM 并启动应用程序二进制文件(NON-XIP)。

图 4 显示了非 XIP 引导的过程。

- 阶段 1: 可引导的映像烧写在外部 flash 中.
- 阶段 2: BootROM 将开始的 4 KB 数据从可引导映像加载到内部 SRAM (OCRAM)。它包括 IVT 和 BD 信息, 将用于应用程序映像加载。

i.MX RT1050 Flashloader use cases

- 阶段 3: BootROM 将从内部 SRAM (OCRAM)开始的 4 KB 数据搬移到可引导映像的目标地址空间。
- 阶段 4: BootROM 继续将可引导映像的其余部分从外部闪存加载到目标地址空间，并启动应用程序二进制文件。

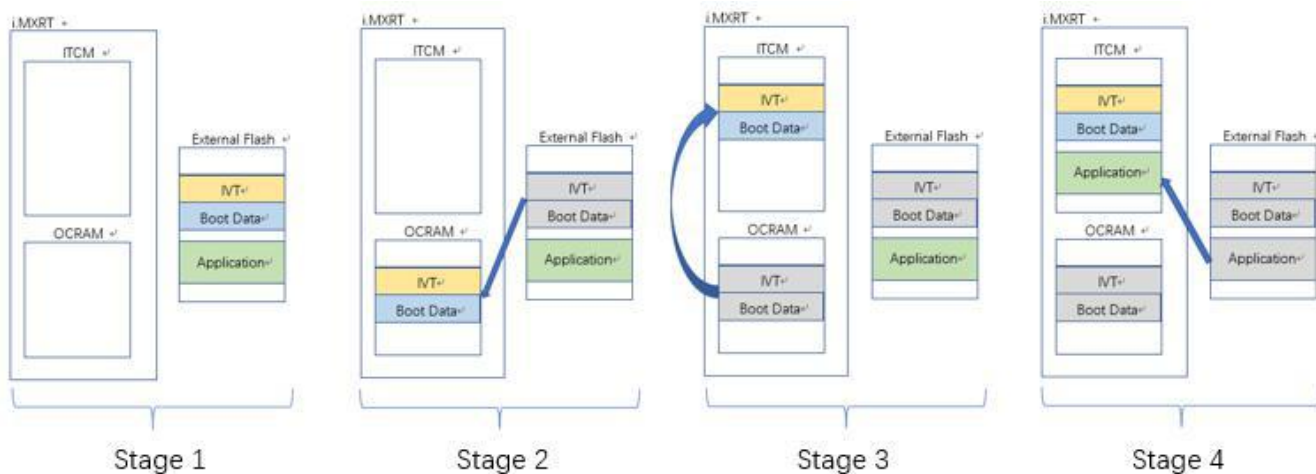


图 4. 非 XIP 启动

在阶段 2 中, 如果 BootROM 发现目标地址等于外部闪存地址, 它将跳过其余阶段, 直接在闪存地址空间中启动应用程序二进制文件, 这就是 XIP 引导。

4. i.MX RT1050 Flashloader 用例

本章节详细描述 Flashloader 的使用用例, 并提供命令行的简单描述。

4.1. 目标平台环境

所有 Flashloader 用例都使用 MIMXRT1050 EVK 目标平台进行演示, 如图 5.所示。

对于 Flashloader 的用例, 设置如下配置:

- 设置串行下载模式, 将“Boot Mode Switch”(SW7) 设置为“0001b”
- BootROM/Flashloader 支持“OpenSDA/UART”和“USB-HID”端口作为与 PC 主机的通信接口。
- 根据所使用的通信接口, 设置正确的“Power Supply Switch”(J1):
 - OpenSDA/UART - J1-5&J1-6
 - USB-HID - J1-3&J1-4

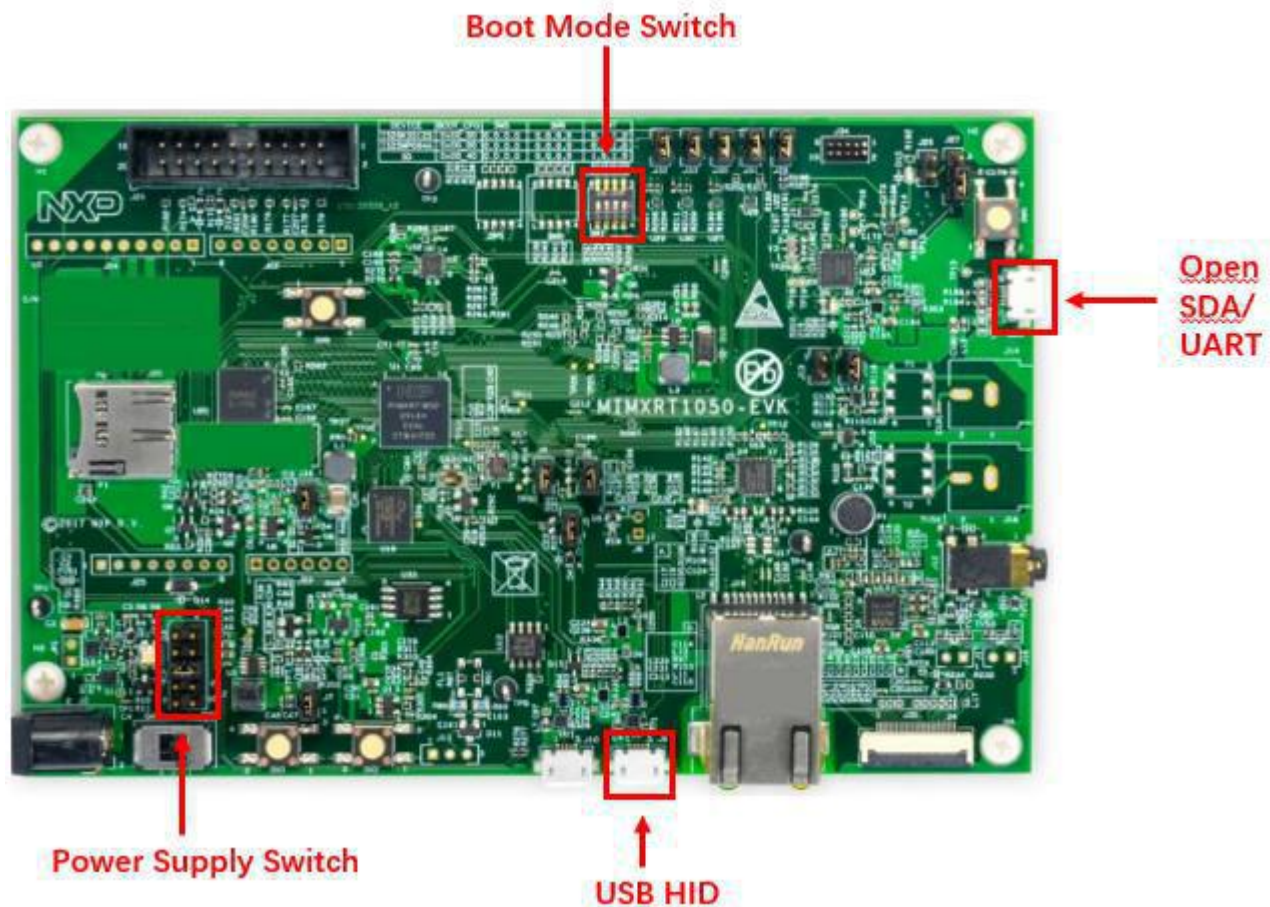


图 5. MIMXRT1050 EVK

当将 USB-HID 设置为与主机 PC (Windows OS) 的通信接口时，USB-HID 设备 (如 图 6 所示) 将出现在 Windows OS 设备管理器中。

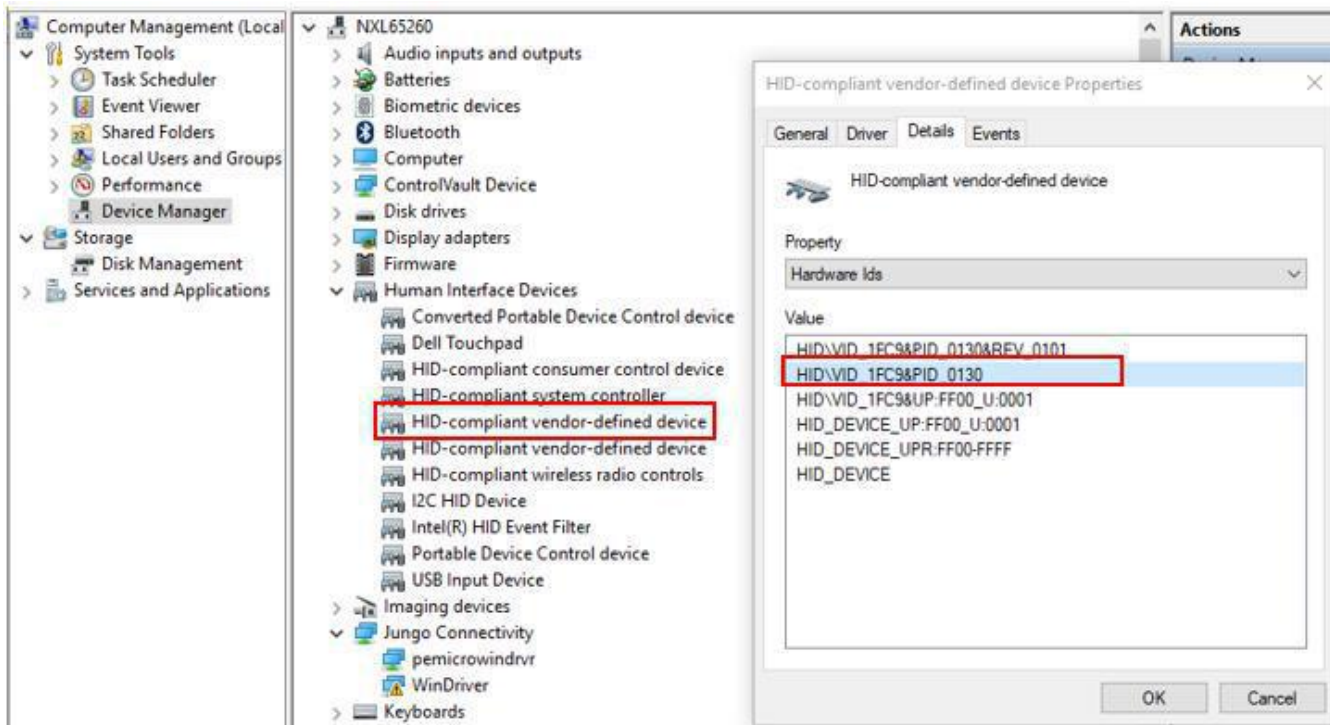


图 6. MIMXRT1050 EVK 板 USB-HID 设备

当将 UART 设置为与主机 PC (Windows OS)的通信接口时，COM 设备(如图 7 所示)将出现在 Windows OS 设备管理器中。

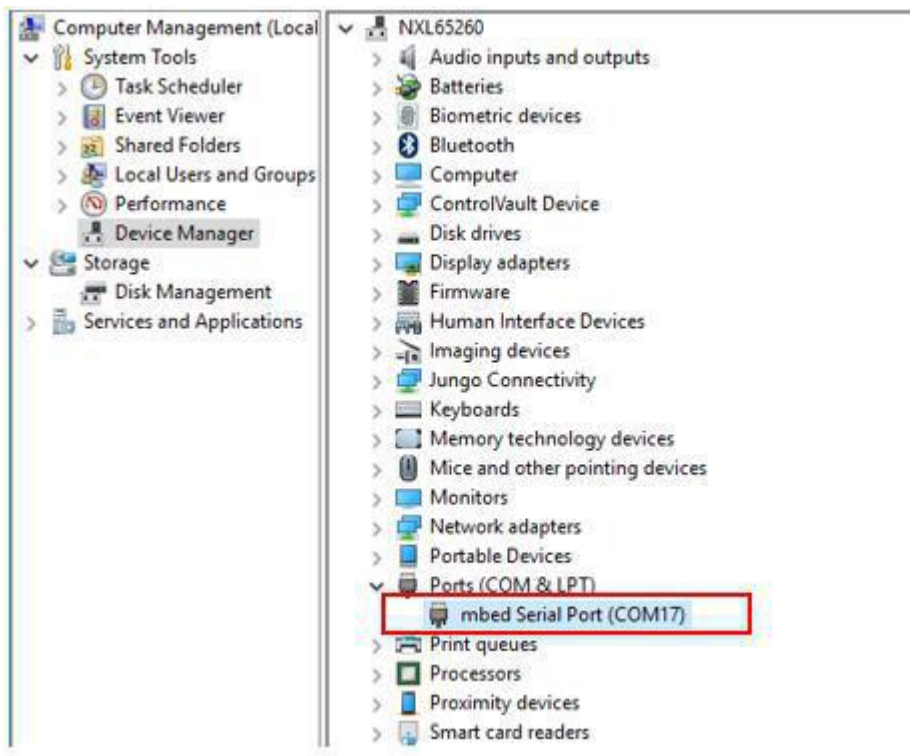


图 7. MIMXRT1050 EVK 板 UART 设备

注意

ROM 检测 USB 或 UART 端口的通信，未使用的端口将被禁用。板卡必须被复位来改变与主机 PC 的通讯端口。

4.2. 串行下载模式

BootROM 基于串行下载协议，通过 UART 或 USB-HID 接口提供串行下载功能。串行下载协议的主要目的是从 PC (SDPHost 工具) 中下载可引导映像 (Flashloader) 到设备的内部 RAM 中，并在 RAM 空间中执行可引导映像。有一组命令可以读取和写入内存/寄存器单元，获取命令的状态，从提供的地址跳转并执行映像。

4.2.1. SDPHost 下载 Flashloader 映像

固化到 i.MX RT 芯片中的 BootROM 不支持对 flash 设备和 eFuse 寄存器的烧写。对于这两点，使用 SDPHost 将 Flashloader 下载到 i.MX RT 内部 RAM (与正在运行的 BootROM 通信)，并从 BootROM 接管该设备 (通过 SDPHost 的 jump-address 命令)。然后实施烧写流程 (与 blhost 工具通信)。

另外，SDPHost jump-address 命令通过 IVT header 启动映像。因此，这里应该使用 *ivt_flashloader.bin* 映像。

1. 将 MIMXRT1050 EVK 板设置为 Serial Downloader 模式并将 UART/USB-HID 接口连接到主机 PC 上。
2. 打开 Windows OS Command Prompt 并切换到路径：
Flashloader_i.MXRT1050_GA\Flashloader_RT1050_1.1\Tools\sdphost\win.
3. 验证 SDPHost 工具是否与 MIMXRT1050-EVK 的 BootROM 通信正常。
 - 使用 UART 接口:

```
>sdphost.exe -p COM17 -- error-status
Status (HAB mode) = 1450735702 (0x56787856) HAB disabled.
Reponse Status = 4042322160 (0xf0f0f0f0) HAB Success.
```

- 使用 USB-HID 接口:

```
>sdphost.exe -u 0x1fc9,0x0130 -- error-status
Status (HAB mode) = 1450735702 (0x56787856) HAB disabled.
Reponse Status = 4042322160 (0xf0f0f0f0) HAB Success.
```

注意

“-p COM17”和“-u 0x1fc9,0x0130”用来表示 COM 和 USB-HID 端口。
 “COM17”和“0x1fc9,0x0130”的值可以在 4.1 章,“目标平台环境”中得到。对于 USB-HID 接口, PID 和 VID 的值可以在命令中被省略。
 下面的示例仅显示使用 USB-HID 接口的命令。

4. 下载 IVT Flashloader 映像到 MIMXRT1050-EVK 板

```
>sdphost.exe -u 0x1fc9,0x0130 -- write-file 0x20000000 "..\..\Mfgtools-
rel\Profiles\MXRT105X\OS Firmware\ivt_flashloader.bin" Preparing to
send 90039 (0x15fb7) bytes to the target.
(1/1)1%Status (HAB mode) = 1450735702 (0x56787856) HAB disabled.
Reponse Status = 2290649224 (0x88888888) Write File complete.
```

5. 启动 Flashloader 映像

```
>sdphost.exe -u 0x1fc9,0x0130 -- jump-address 0x20000400
Status (HAB mode) = 1450735702 (0x56787856) HAB disabled.
```

USB-HID 由运行的 Flashloader 映像重新枚举。通过 USB-HID 的通信从 BootROM 变成运行在内部 RAM 中的 Flashloader。

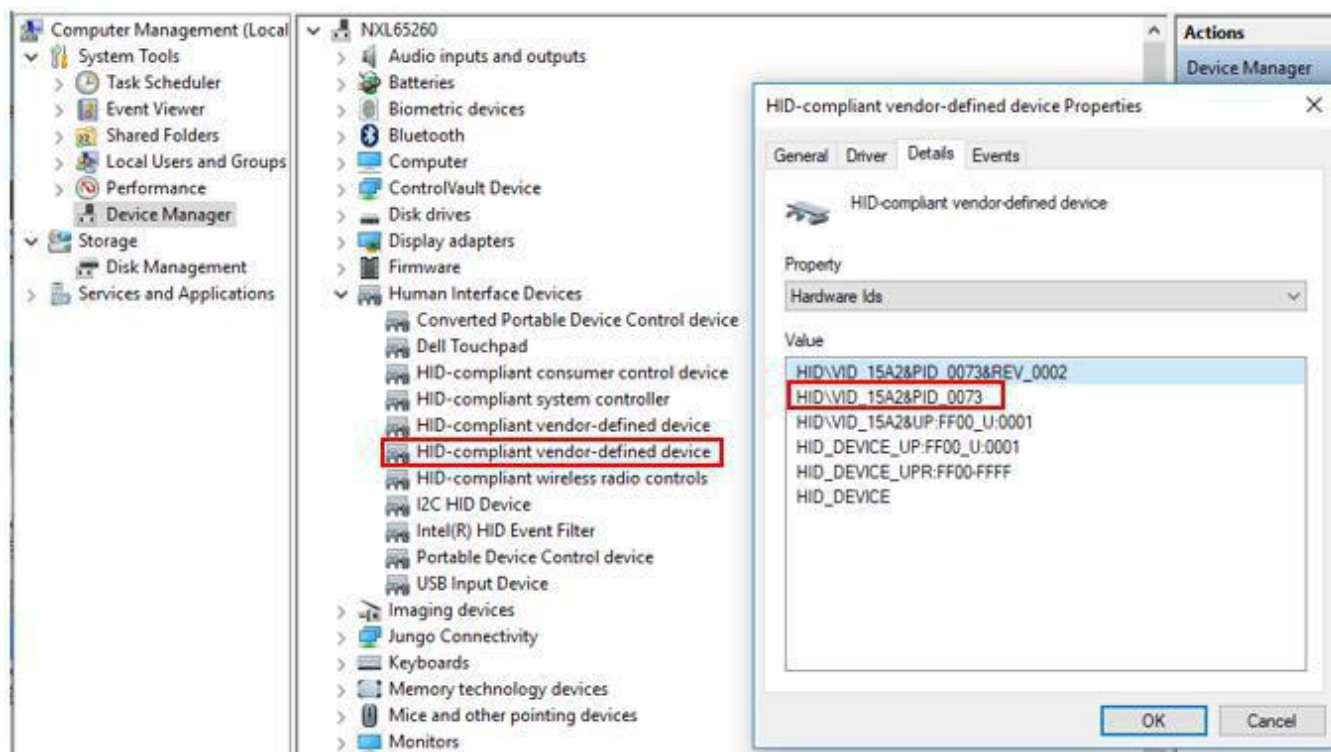


图 8. 重新枚举的 USB-HID 设备

6. 使用 blhost 工具验证与正在运行的 Flashloader 的通信。

```
# change the directory to
"Flashloader_i.MXRT1050_GA\Flashloader_RT1050_1.1\Tools\blhost\win"
>blhost.exe -u -- get-property 1
Inject command 'get-property'
Response status = 0 (0x0) Success.
Response word 1 = 1258422528 (0x4b020100)
Current Version = K2.1.0
```

4.3. 烧写 OCOTP (eFuse)

1. 下载和启动 Flashloader 映像, 如 4.2.1 章, “SDPHost 下载 Flashloader 映像”所示。
2. 验证 blhost 工具是否与运行在 MIMXRT1050-EVK 板上的 Flashloader 通信正常。

```
# change the directory to
"Flashloader_i.MXRT1050_GA\Flashloader_RT1050_1.1\Tools\blhost\win"
>blhost.exe -u 0x15a2,0x0073 -- get-property 1 Inject command
'get-property'
Response status = 0 (0x0) Success.
Response word 1 = 1258422528 (0x4b020100)
Current Version = K2.1.0
```

3. 显示有关 eFuse 操作命令的 blhost 帮助信息。 .

```
>blhost.exe -?
.....
Command:
.....
    efuse-program-once <addr> <data>
                                Program one word of OCOTP Field
                                <addr> is ADDR of OTP word, not the shadowed memory
address.
                                <data> is hex digits without prefix '0x'
    efuse-read-once <addr>
                                Read one word of OCOTP Field
                                <addr> is ADDR of OTP word, not the shadowed memory
address.
```

4. 烧写 eFuse 寄存器 SRK_REVOKE 作为示例。
 - SRK_REVOKE eFuse OCOTP 索引: 0x2F.
 - SRK_REVOKE eFuse 影子寄存器地址: 0x401F46F0.
 - 将 SRK_REVOKE eFuse LSB 烧写为: 0x5A.
 - 将 SRK_REVOKE eFuse MSB 烧写为: 0xFE.
 - 通过影子寄存器验证 SRK_REVOKE eFuse

```
>blhost.exe -u 0x15a2,0x0073 -- efuse-program-once 0x2F 0000005A
Inject command 'efuse-program-once'
Successful generic response to command 'efuse-program-once'
Response status = 0 (0x0) Success.

>blhost.exe -u 0x15a2,0x0073 -- efuse-program-once 0x2F FE000000
Inject command 'efuse-program-once'
```

```
Successful generic response to command 'efuse-program-once'
Response status = 0 (0x0) Success.
```

```
>blhost.exe -u 0x15a2,0x0073 -- efuse-read-once 0x2F
Inject command 'efuse-read-once' Response status = 0
(0x0) Success.
Response word 1 = 4 (0x4)
Response word 2 = -33554342 (0xfe00005a)
```

5. 验证 SRK_REVOKE eFuse 的影子寄存器

```
>blhost.exe -u 0x15a2,0x0073 -- read-memory 0x401F46F0
4 Inject command 'read-memory'
Successful response to command 'read-memory'
5a 00 00 fe (1/1)100% Completed!
Successful generic response to command 'read-memory'
Response status = 0 (0x0) Success.
Response word 1 = 4 (0x4)
Read 4 of 4 bytes.
```

6. 一些关键点.

- eFuse 位只能从 0 到 1 被烧写。OCOTP 忽略从 1 变化到 0 的写操作。对于一个 eFuse 寄存器，efuse-program-once 命令可以在多个步骤中针对特定的位字段操作。
- efuse-program-once 命令默认包括 eFuse 寄存器重载命令。最新的 eFuse 寄存器值可以从“efuse-program-once”命令后的影子寄存器中获得。

4.3.1. 烧写 boot mode eFuse 以使能 SD boot

- BOOT_CFG eFuse OCOTP 索引: 0x05.
- BOOT_CFG eFuse OCOTP 索引: 0x06.
- BOOT_CFG (0x05) eFuse 影子寄存器地址: 0x401F4450.
- BOOT_CFG (0x06) eFuse 影子寄存器地址: 0x401F4460.
- 将 BOOT_CFG (0x06) eFuse 设置为: 0x00000010.
- 将 BOOT_CFG (0x05) eFuse 设置为: 0x00000040.
- 通过影子寄存器验证 eFuse 寄存器。

首先执行 4.3 章,“烧写 OCOTP (eFuse)”. 中的第 1 步到第 3 步, 然后执行:

```
>blhost.exe -u -- efuse-program-once 0x06
00000010 >blhost.exe -u -- efuse-program-once
0x05 00000040 >blhost.exe -u -- efuse-read-once
0x06 >blhost.exe -u -- efuse-read-once
0x05 >blhost.exe -u -- read-memory 0x401F4460
4 >blhost.exe -u -- read-memory 0x401F4450 4
```

4.3.2. 烧写 FlexRAM eFuse

- MISC_CFG eFuse OCOTP 索引: 0x2D.
- MISC_CFG (0x2D) eFuse 影子寄存器地址: 0x401F46D0.
- 选择 0011 组: DTCM 128 KB, ITCM 32 KB, ORAM 352 KB.

- 编写 MISC_CFG (0x2D) eFuse 到: 0x00030000.
- 通过影子寄存器验证 eFuse 寄存器。

表 2 显示了 i.MX RT1050 FlexRAM RAM 的分区信息.

Table 2. i.MX RT1050 FlexRAM 分区

Parameter	DTCM	ITCM	ORAM
0000	128 KB	128 KB	256 KB
0001	128 KB	64 KB	320 KB
0010	128 KB	256 KB	128 KB
0011	128 KB	32 KB	352 KB
0100	64 KB	128 KB	320 KB
0101	64 KB	64 KB	384 KB
0110	64 KB	256 KB	192 KB
0111	0 KB	448 KB	64 KB
1000	256 KB	128 KB	128 KB
1001	256 KB	64 KB	192 KB
1010	192 KB	256 KB	64 KB
1011	448 KB	0 KB	64 KB
1100	0 KB	128 KB	384 KB
1101	32 KB	32 KB	448 KB
1110	0 KB	256 KB	256 KB
1111	0 KB	0 KB	512 KB

首先, 执行 4.3 章, “烧写 OCOTP (eFuse)” 中的第 1 步到第 3 步, 然后执行:

```
>blhost.exe -u -- efuse-program-once 0x2D
00030000 >blhost.exe -u -- efuse-read-once
0x2D >blhost.exe -u -- read-memory 0x401F46D0 4
```

4.4. 构建可引导映像

elftosb 工具可用来创建一个二进制文件, 其中包含应用程序映像以及一系列 Flashloader 命令。输出文件被称为 SB(Secure Binary)文件, 这些文件的扩展名是 .sb。elftosb 使用一个命令文件来控制输出文件中的 Flashloader 命令序列。这个命令文件称为 BD(Boot Descriptor)。

QSPI NOR flash 的 XIP “hello_world”项目用于演示创建可引导映像的过程。

1. 编译生成 XIP *hello_world.out*, 并且设置宏
“XIP_BOOT_HEADER_ENABLE=0” 和
“XIP_BOOT_HEADER_DCD_ENABLE=0”。
2. 将 *hello_world.out* 复制到 *elftosb/win* 目录。



图 9. 将 *hello_world.out* 复制到 *elftosb*

3. 打开 Windows 操作系统命令提示符并将目录更改为

Flashloader_i.MXRT1050_GA\Flashloader_RT1050_1.1\Tools\elftosb\win.

```
>elftosb.exe -f imx -V -c ..\..\bd_file\imx10xx\imx-flexspinor-normal-unsigned.bd -o
ivt_flexspi_nor_hello_world.bin hello_world.out
```

执行上面的命令后，有两个带着 IVT 信息的可引导映像：

- *ivt_flexspi_nor_hello_world.bin*
从 0 偏移地址到 *ivt_offset* 的区域由填充字节填充(所有都是 0x00)。
- *ivt_flexspi_nor_hello_world_nopadding.bin*
在 *ivt_offset* 之前没有填充字节。
用于为 QSPI NOR flash 生成 SB 文件。

注意

如果输入文件(.out)包含引导头部分，则命令可能会崩溃。确保在构建.out 文件时设置了宏“XIP_BOOT_HEADER_ENABLE=0”和“XIP_BOOT_HEADER_DCD_ENABLE=0”。

4. 创建最终的 SB 映像。

```
>elftosb.exe -f kinetis -V -
c ..\..\bd_file\imx10xx\program_flexspinor_image_qspinor.bd -o boot_image.sb
ivt_flexspi_nor_hello_world_nopadding.bin
```

boot_image.sb 文件生成在 *elftosb\win* 目录中

图 10. 创建 *boot_image.sb*

4.5. 烧写外接 flash 设备

对于 flash 烧写，Flashloader 提供了一个易于使用的 GUI 编程工具(Mfgtool)。

4.5.1. Mfgtool

Mfgtool 是一个帮助烧写外部 flash 的 GUI 工具。它集成了 SDPHost 和 blhost 工具的功能，可以检测连接到 PC 主机的 i.MX MCU BootROM。

以下步骤展示了如何使用 Mfgtool 烧写上一章中生成的 SB 映像。

1. 复制 *boot_image.sb* 文件到 `<Mfgtool_root_dir>\Profiles\MXRT105X\OS Firmware` 文件夹。
2. 将 `<Mfgtool_root_dir>` 目录下 *cfg.ini* 文件中 “[List]” 下的 “name” 更改为 “MXRT105x-DevBoot”。



图 11. 设置 LIST 项的名称

3. 将 MIMXRT1050-EVK 板设置为 Serial Downloader 模式，将 USB-HID 接口连接到主机 PC 上。
4. 打开 Mfgtool 并连接到 MIMXRT1050-EVK 板。

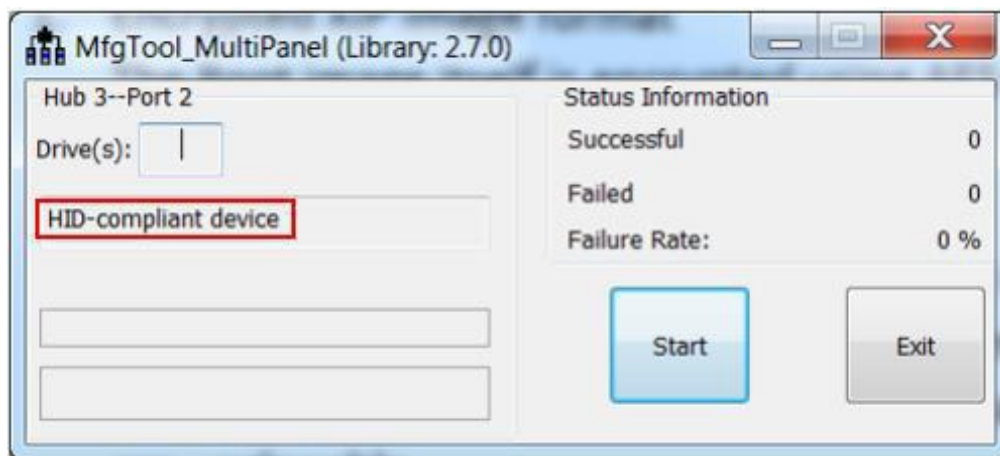


Figure 12. 连接 MIMXRT1050 EVK

5. 对可引导的映像进行烧写。单击“Start”按钮，并等待它完成，如图 13. 所示。要退出 Mfgtool，点击“Stop”和“Exit”按钮。

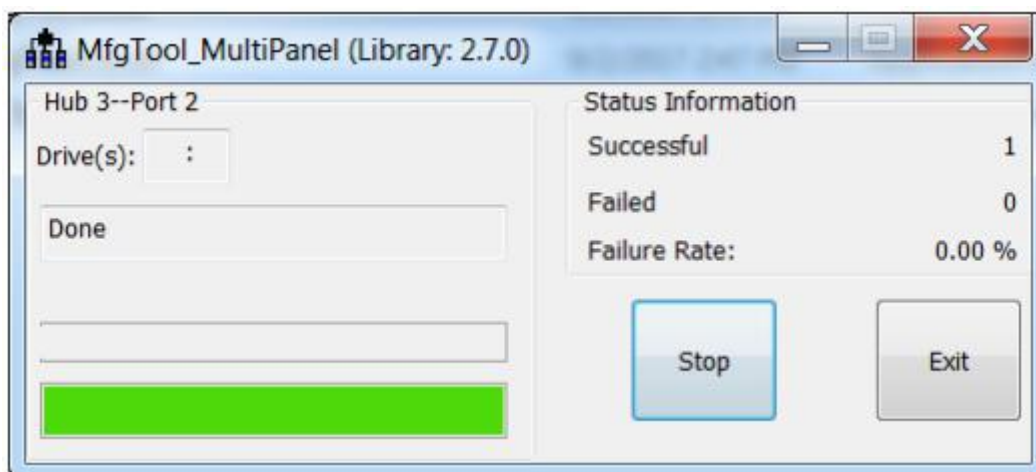


图 13. flash 烧写成功

6. 将 MIMXRT1050-EVK 板切换到正确引导模式，并验证应用。

有关构建可引导映像和烧写外部闪存的更多信息，请参见 *How to Enable Boot from Octal SPI Flash and SD Card* (文档 [AN12107](#))，以及 *How to Enable Boot from QSPI Flash* (文档 [AN12108](#))。

5. 结语

本应用笔记描述了 flash 加载程序的背景知识和 flash 加载程序的用例。欲了解更多信息，请参阅以下文件：

- *i.MX MCU Manufacturing User's Guide.pdf*
- *Kinetis blhost User's Guide.pdf*
- *Kinetis SDPHost User's Guide.pdf*
- *MCUX Flashloader Reference Manual.pdf*

6. 修订历史

表 3 总结了自最初版本以来对该文档所做的更改。

表 3. 修订历史

修订编号	日期	实质变化
0	08/2018	最初的版本
1	09/2018	修订 4.3.2 章, “烧写 FlexRAM eFuse”中的错误

How to Reach Us:

Home Page:

www.nxp.com

Web Support:

www.nxp.com/support

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 sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "ty 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:

www.nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these v and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C- 5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C- Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QoriQ, QoriQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. Arm, AMBA, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and μ Vision are registered trademarks (series) in the EU and/or elsewhere. Arm7, Arm9, Arm11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, Mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2018 NXP B.V.

Document Number: AN12238

Rev. 1

09/2018



arm