



SCM1612

Wi-Fi 6 和 BLE 5 低功耗 SoC

设备固件更新指南

文档版本 1.1

发布日期 2023-08-15

联系方式

速通半导体科技有限公司 (www.senscomm.com)

江苏省苏州市工业园区苏州大道西 2 号国际大厦 303 室

销售或技术支持，请发送电子邮件至

support@senscomm.com

免责声明和注意事项

本文档仅按"现状"提供。速通半导体有限公司保留在无需另行通知的情况下对其或本文档中包含的任何规格进行更正、改进和其他变更的权利。

与使用本文档中的信息有关的一切责任，包括侵犯任何专有权利的责任，均不予承认。此处不授予任何明示或暗示、通过禁止或以其他方式对任何知识产权的许可。

本文档中的所有第三方信息均按"现状"提供，不对其真实性和准确性提供任何保证。

本文档中提及的所有商标、商号和注册商标均为其各自所有者的财产，特此确认。

© 2024 速通半导体有限公司。保留所有权利。

版本历史

版本	日期	描述
1.1	2023-08-15	格式修改
1.0	2023-08-04	1.0 发布
0.1	2023-07-11	初稿

目录

版本历史.....	3
1 引言.....	5
2 工作流程.....	6
2.1 高级操作.....	6
2.2 闪存布局.....	8
3 构建固件.....	10
3.1 配置与构建.....	10
3.2 固件格式.....	10
4 源代码概览.....	12
4.1 主要的 MCUboot 源文件.....	12
4.2 API 函数.....	12
5 固件更新示例.....	14
5.1 设置 HTTP 服务器.....	14
5.2 文件传输.....	15
5.3 验证固件.....	16
6 安全启动（待定）.....	18
6.1 BootROM 的安全引导.....	18
6.2 引导程序的安全启动.....	18
7 闪存加密（待定）.....	19

1 引言

SCM1612 SDK 支持空中设备固件更新（DFU OTA）。本指南为您提供如何利用网络连接进行固件更新的详细说明。

在 SDK 中，我们已集成了广泛使用的开源引导加载程序 MCUBoot。此外，我们还提供了专门针对特定架构的移植层，以便于无缝集成。

需要特别注意的是，MCUBoot 专为支持完整的固件更新而设计，不支持增量更新。为了更全面地了解 MCUBoot 及其功能，建议用户查阅其官方文档，地址为：
<https://docs.mcuboot.com/>。

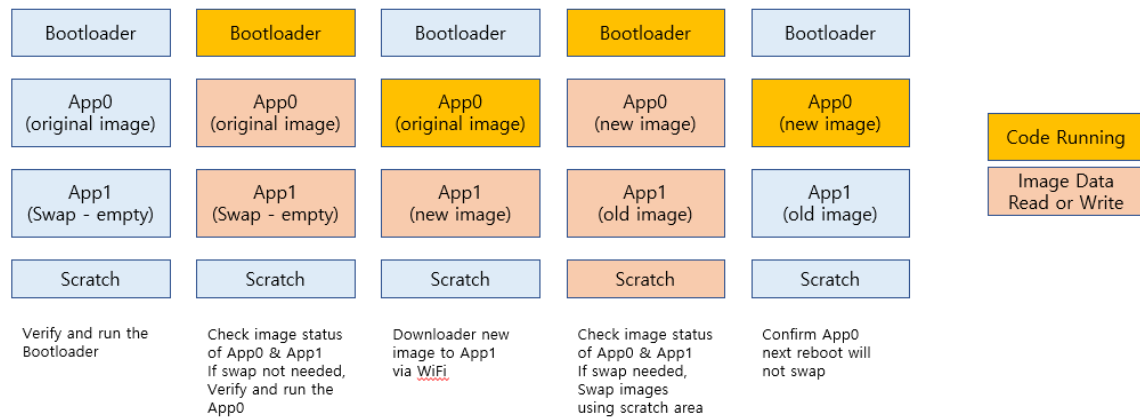
MCUBoot 提供了多种配置选项，以满足不同应用的需求。其中一些选项可以通过 menuconfig 进行访问和修改，而其他选项则需要在 mcboot_config.h 文件中手动调整。

2 工作流程

2.1 高级操作

设备固件的更新过程可以总结为以下几个步骤：

- 初始设置：设备预先加载有引导加载程序和主应用程序。
- 启动过程：开机时，引导加载程序启动位于主应用程序分区的程序。
- 更新准备：在主应用程序中，更新代理负责下载新的应用程序固件。这个新固件存储在一个辅助应用程序分区中。
- 固件下载完成：一旦新的应用程序固件成功下载，应用程序将记录此状态并触发系统重启。
- 引导加载程序检查：在随后的启动过程中，引导加载程序检查记录的状态。如果在辅助应用程序分区中检测到更新的固件，引导加载程序将与主应用程序分区中的固件交换。
- 应用程序启动：然后，引导加载程序从主应用程序分区启动程序，该分区现在包含更新的固件。
- 状态清除：如果新的应用程序固件正常运行，应用程序可以继续清除更新状态，表示成功更新。



对于 OTA 固件更新过程，设备必须为辅助应用程序分区分配与主应用程序分区相同大小的空间。此外，还需要一个指定的临时区，作为在更新过程中交换主应用程序分区和辅助应用程序分区固件的临时空间。

2.2 闪存布局

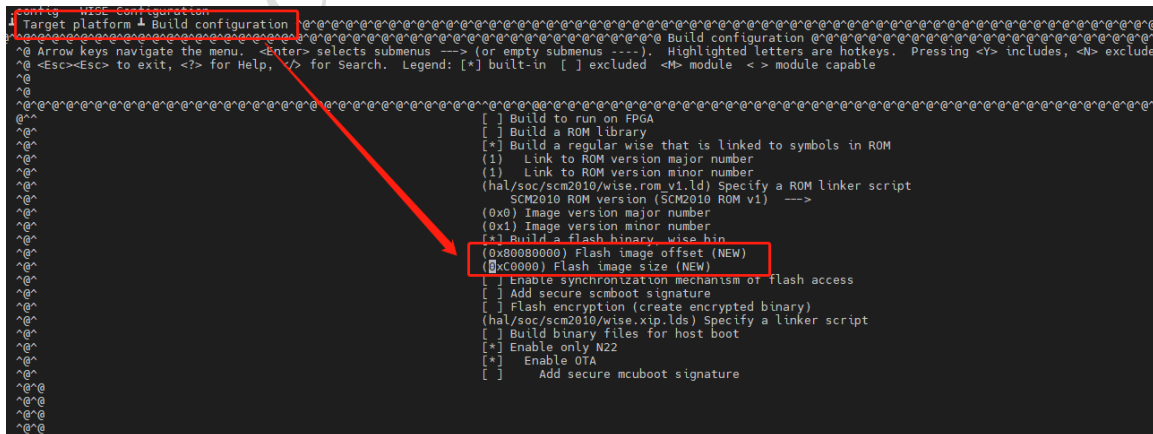
为了成功地更新固件，设备需要在闪存中指定特定的区域。这些区域包括：

- 引导加载程序
- 主应用程序分区
- 辅助应用程序分区
- 临时区

以下是 SCM1612 内部 16Mbit 闪存的推荐布局：

组件	地址	大小 (KB)
引导程序	0x80000000	64
临时区	0x80010000	16
存储文件系统	0x80014000	16
休眠备份	0x80018000	416
主应用程序分区	0x80080000	768
辅助应用程序分区	0x80140000	768

您可以使用`menuconfig`过程修改闪存布局的地址和大小。如果对闪存布局进行了任何更改，都必须重新构建引导加载程序和应用程序，以识别这些修改。



```

config - wise configuration
Target platform Build configuration
@ Arrow keys navigate the menu. <Enter> selects submenus --> (or empty submenus ---). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes
@ <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module <> module capable
@
@
[*] Build to run on FPGA
[ ] Build a ROM library
[*] Build a regular wise that is linked to symbols in ROM
(1) Link to ROM version major number
(1) Link to ROM version minor number
(hal/soc/scm2010/wise.rom.v1.ld) Specify a ROM linker script
SCM2010 ROM version (SCM2010 ROM v1) -->
(0x0) Image version major number
(0x1) Image version minor number
[*] Build a flash binary wise bin
(0x80000000) Flash image offset (NEW)
(0x00000000) Flash image size (NEW)
[ ] Enable synchronization mechanism of flash access
[ ] Add secure scmboot signature
[ ] Flash encryption (create encrypted binary)
(hal/soc/scm2010/wise.xip.ld) Specify a linker script
[ ] Build binary files for host boot
[*] Enable only N22
[*] Enable OTA
[ ] Add secure mcuboot signature
  
```



```
config - WISE Configuration
MCUBoot - OTA partition on the flash
MCUboot application image primary slot offset
MCUboot application image secondary slot offset
MCUboot application image slot size (in bytes)
MCUboot scratch partition offset
MCUboot scratch partition size (in bytes)
```

此外，由于闪存擦除块的大小，地址必须与 4KB 边界对齐。

3 构建固件

3.1 配置与构建

SDK 中的默认配置已经启动了固件更新功能。因此，生成的应用固件将包括 MCUBoot 头部，使其适用于更新。

以下是使能固件更新功能的 menuconfig 选项的示例：

```
[ ] Build to run on FPGA
[ ] Build a ROM library
[*] Build a regular wise that is linked to symbols in ROM
    (1) Link to ROM version major number
    (1) Link to ROM version minor number
    (hal/soc/scm2010/wise.rom_v1.ld) Specify a ROM linker script
        SCM2010 ROM version (SCM2010 ROM v1) --->
    (0x0) Image version major number
    (0x1) Image version minor number
[*] Build a flash binary, wise.bin
[ ] Enable synchronization mechanism of flash access
[ ] Add secure scmboot signature
[ ] Flash encryption (create encrypted binary)
    (hal/soc/scm2010/wise.xip.lds) Specify a linker script
[ ] Build binary files for host boot
[*] Enable only N22
[*] Enable OTA
[ ] Add secure mcuboot signature
```

当设备固件更新被启用时，典型的输出名为 wise.mcuboot.bin。

3.2 固件格式

MCUboot 要求一个提供关于固件详细信息的镜像头部。一个固定大小的头部总是附加在可执行固件的前面，如果需要更多信息，则会附加可变大小的 TLV 格式数据。在构建过程中，imgtool 工具会附加这些额外的数据。

头部包括以下细节：

- 魔法数字
- 加载内存地址
- 头部大小
- TLV 大小
- 固件大小
- 标志位
- 版本

尾部可能包含以下信息：

- 公钥或密钥哈希
- 固件哈希
- 签名

4 源代码概览

4.1 主要的 MCUBoot 源文件

以下是一些重要的 MCUBoot 源代码。

源代码	描述
lib/scm_mcuboot/mcuboot	原始 MCUBoot 库代码
lib/scm_mcuboot/mcuboot/wise	用于 SDK 的 Senscomm 移植层
lib/scm_mcuboot/loader	MCUBoot 引导程序主函数
lib/scm_mcuboot/update_agent	MCUBoot 更新代理示例

4.2 API 函数

以下函数对于引导程序和应用程序都是不可或缺的：

名称	描述
boot_go	引导程序调用它来启动应用程序的引导。 过程将根据状态数据包括验证、交换和引导。
boot_set_pending_multi	应用程序调用它来标记镜像处于挂起状态，以便引导程序在下次重启时交换并引导镜像。
boot_set_confirmed_multi	应用程序调用它来标记当前镜像有效并永久设置。
flash_area_open	根据给定的标识从闪存映射中检索闪存区域。
flash_area_read	将闪存内容读取到缓冲区
flash_area_write	将缓冲区写入闪存

flash_area_close	完成闪存操作
------------------	--------

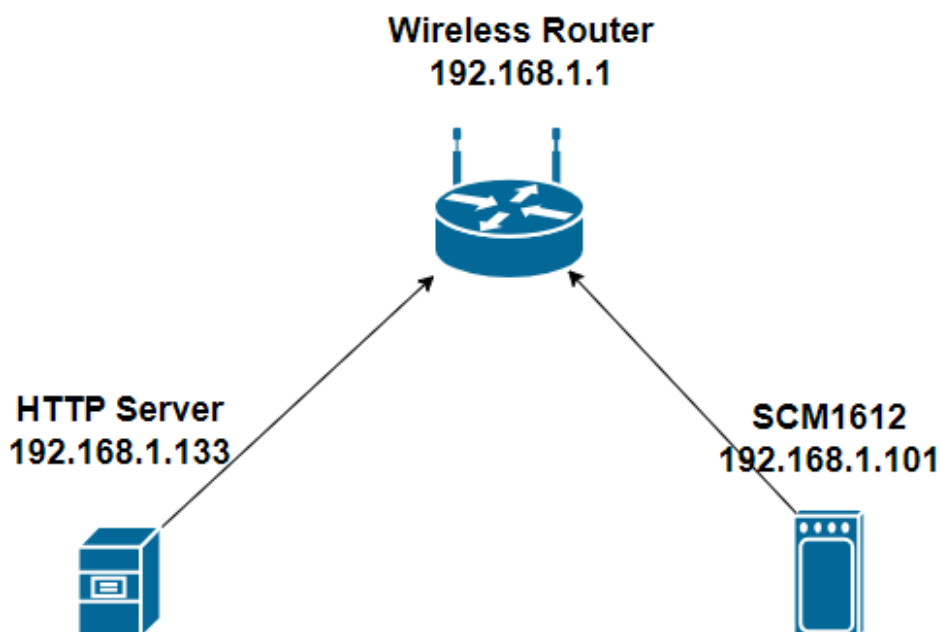
Senscomm Confidential

5 固件更新示例

设备固件可以使用特定的测试命令进行更新。例如，我们的示例更新代理使用 HTTP 协议进行固件下载。但值得注意的是，其他协议也可以用于此目的。在更新时，应用程序必须调用适当的闪存 API 函数来擦除并将更新的固件写入次要分区。

5.1 设置 HTTP 服务器

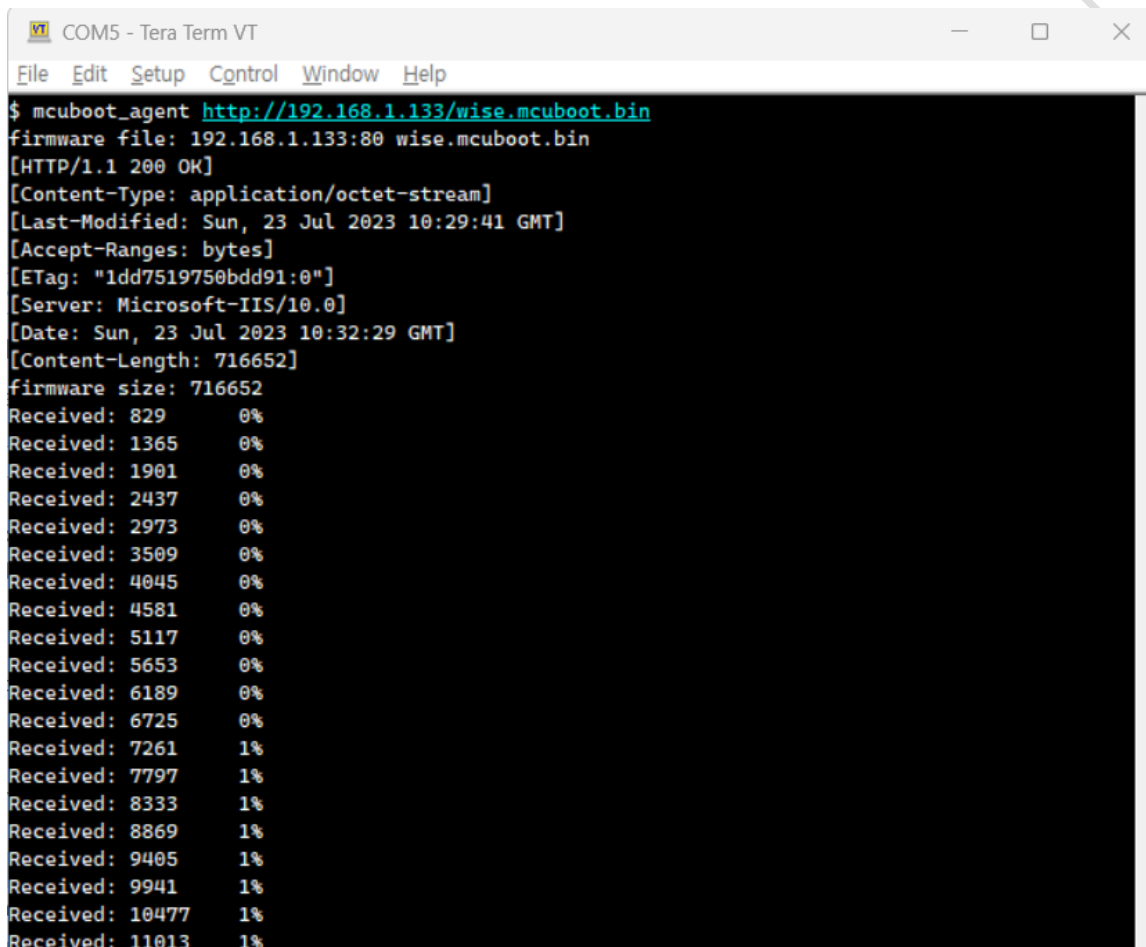
本节概述了在局域网(LAN)环境中建立过程。在这里，无线路由器充当网关。此外，我们将深入探讨部署 HTTP 服务器的步骤，该服务器将存储设备固件。如何通过 Wi-Fi 将 SCM1612 连接到这个无线路由器的过程详细描述在"SCM1612 Wi-Fi 软件开发指南"中。



5.2 文件传输

下图显示了镜像更新过程的开始。新镜像放置在 IP 地址为 192.168.1.133 的 HTTP 服务器上。

```
# mcuboot_agent http://192.168.1.133/wise.mcuboot.bin
```



```
COM5 - Tera Term VT
File Edit Setup Control Window Help
$ mcuboot_agent http://192.168.1.133/wise.mcuboot.bin
firmware file: 192.168.1.133:80 wise.mcuboot.bin
[HTTP/1.1 200 OK]
[Content-Type: application/octet-stream]
[Last-Modified: Sun, 23 Jul 2023 10:29:41 GMT]
[Accept-Ranges: bytes]
[ETag: "1dd7519750bdd91:0"]
[Server: Microsoft-IIS/10.0]
[Date: Sun, 23 Jul 2023 10:32:29 GMT]
[Content-Length: 716652]
firmware size: 716652
Received: 829      0%
Received: 1365    0%
Received: 1901    0%
Received: 2437    0%
Received: 2973    0%
Received: 3509    0%
Received: 4045    0%
Received: 4581    0%
Received: 5117    0%
Received: 5653    0%
Received: 6189    0%
Received: 6725    0%
Received: 7261    1%
Received: 7797    1%
Received: 8333    1%
Received: 8869    1%
Received: 9405    1%
Received: 9941    1%
Received: 10477   1%
Received: 11013   1%
```

成功传输完成后，设备将自动重启。随后，引导程序将交换主要和次要的固件，如下图所示。

```
COM5 - Tera Term VT
File Edit Setup Control Window Help
*** Booting MCUboot ***
Swap Progress: 4096 of 716800 bytes [0 %]
Swap Progress: 8192 of 716800 bytes [1 %]
Swap Progress: 12288 of 716800 bytes [1 %]
Swap Progress: 16384 of 716800 bytes [2 %]
Swap Progress: 20480 of 716800 bytes [2 %]
Swap Progress: 24576 of 716800 bytes [3 %]
Swap Progress: 28672 of 716800 bytes [4 %]
Swap Progress: 32768 of 716800 bytes [4 %]
Swap Progress: 36864 of 716800 bytes [5 %]
Swap Progress: 40960 of 716800 bytes [5 %]
Swap Progress: 45056 of 716800 bytes [6 %]
Swap Progress: 49152 of 716800 bytes [6 %]
Swap Progress: 53248 of 716800 bytes [7 %]
Swap Progress: 57344 of 716800 bytes [8 %]
Swap Progress: 61440 of 716800 bytes [8 %]
Swap Progress: 65536 of 716800 bytes [9 %]
Swap Progress: 69632 of 716800 bytes [9 %]
Swap Progress: 73728 of 716800 bytes [10 %]
Swap Progress: 77824 of 716800 bytes [10 %]
Swap Progress: 81920 of 716800 bytes [11 %]
Swap Progress: 86016 of 716800 bytes [12 %]
Swap Progress: 90112 of 716800 bytes [12 %]
Swap Progress: 94208 of 716800 bytes [13 %]
Swap Progress: 98304 of 716800 bytes [13 %]
Swap Progress: 102400 of 716800 bytes [14 %]
Swap Progress: 106496 of 716800 bytes [14 %]
```

完成交换后，引导程序将启动新的应用程序。

5.3 验证固件

要将新的固件指定为永久的，请使用以下命令：

```
# mcuboot_confirm
```

```
COM5 - Tera Term VT
File Edit Setup Control Window Help
WISE 2018.02+ (Jul 23 2023 - 18:45:19 +0900)
$ mcuboot_confirm

$ mcuboot_version
Image version 1.0.0.0
$
```


重要提示：如果跳过这个确认步骤，引导程序在下次设备重启时，将在主要和次要分区中还原固件，有效地恢复之前的固件。这是一个安全措施，假设更新的固件可能遇到执行问题。

Senscomm Confidential

6 安全启动（待定）

当启用安全启动时，设备只会启动使用用户的安全密钥签名的固件。安全启动过程分为两步：

1. BootROM 验证引导程序
2. 引导程序反过来验证应用程序

6.1 BootROM 的安全引导

如果设置了相应的 eFuse 位，BootROM 将使能安全启动。它使用 eFuse 安全密钥验证引导程序的签名，或位于闪存开头的固件。此固件应具有 Senscomm 特定的头部和必要的详细信息。

6.2 引导程序的安全启动

如果引导程序是使用相关的 menu 选项启用构建的，则会使能安全启动。在启动过程中，使用安全密钥构建的引导程序将验证应用程序固件的签名。此应用程序应具有 MCUBoot 特定的头部和带有必要详细信息的 TLVs。

7 闪存加密（待定）

当启用闪存加密时，更新过程会进行轻微的修改。考虑到原始固件数据（无加密）不应该被暴露，因此用于更新的固件在下载阶段之前也应该被加密。

构建系统支持闪存加密。启用后，它会生成加密的固件和原始固件。

Senscomm Confidential