



实验2——Matter设备OTA

余发明

2023-02-21



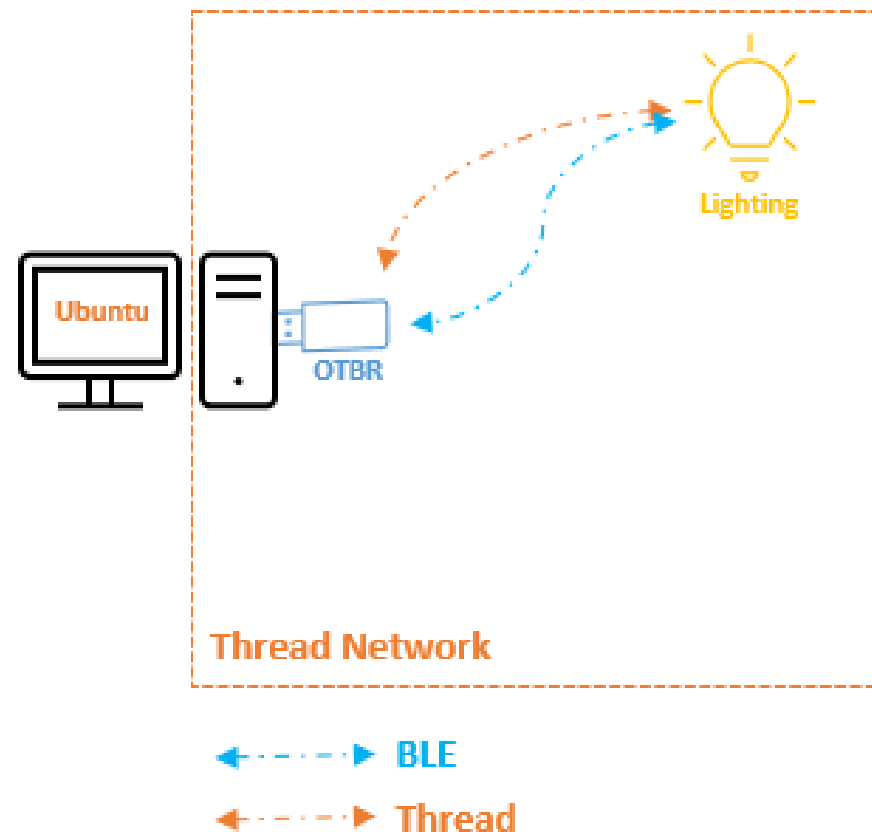
大纲

- 实验介绍
- 内部Flash OTA升级实验
 - 预留操作时间
- 外部Flash OTA升级实验
 - 预留操作时间



实验介绍——整体概述

- 本实验将以一个**Matter**灯为例，实现**OTA**升级功能。让开发者对**Matter**产品升级过程有一个初步的认识和了解。
- 整个网络中的设备说明
 - Ubuntu: Linux主机
 - OTBR: 基于Ubuntu平台的Thread边界路由器
 - Lighting: Matter灯
 - BLE: 用于Matter协议配网



实验介绍——预备知识

- 对**Ubuntu**开发环境有一定的使用基础
- 对**OpenThread**协议有一定的了解
- 对**C++**有一定的基础
- 对**Matter**协议有个初步认识
- 名词解释
 - OT-RCP: Open Thread Radio Co-Processor。Thread无线协处理器。
 - OTBR: Open Thread Board Router。Thread边界路由器。
 - chip-tool: Linux应用程序。用于Matter协议控制。
 - ot-ctl: Thread网络控制的应用程序



实验介绍——实验目的

- **Matter**设备的**OTA**功能需要用到哪些工具
- **Matter**设备的固件在**Flash**中分布
- 掌握芯科科技**Matter**设备**OTA**升级流程



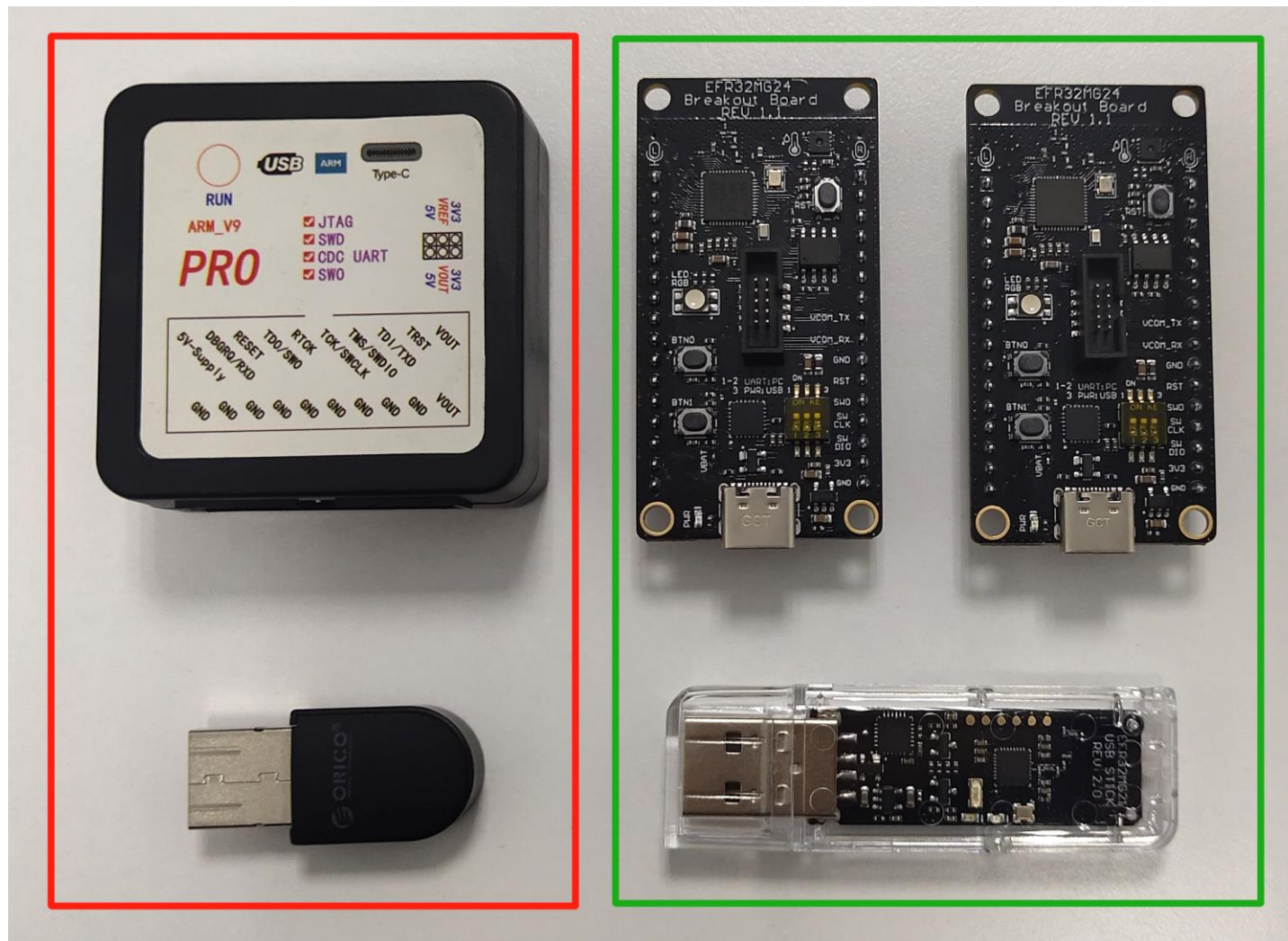
实验介绍——实验内容

- 使用芯片内部Flash实现OTA
- 使用外部Flash实现OTA



实验介绍——实验前准备

- 安装Ubuntu虚拟机的电脑一台
- EFR32MG24 Breakout Board 1块
- USB Stick
- J-Link烧录器
- BLE Dongle
- 树莓派一台（可选）
- 开发板、烧录器需要用到的连接线



内部Flash OTA——准备流程介绍

- 修改**Matter**灯的固件（**V2**）
- 烧录**Matter**灯固件到设备中并查看**Flash Map**
- 创建一个支持内部**Flash OTA**功能的**Bootloader**
- 烧录**Bootloader**到**Matter**灯设备中
- 配置**Matter**灯入网
- 修改并编译一个新版本的**Matter**灯固件（**V3**）
- 将**v3**版固件**s37**文件转换成**ota**文件

内部Flash OTA——修改Matter灯固件V2 (1/7)

- 在`config/sl_matter_config.h`中修改日志配置
 - 由于内部Flash空间有限，需要关闭日志来减小固件大小

```
#define SL_MATTER_STACK_LOCK_TRACKING_MODE SL_MATTER_STACK_LOCK_TRACKING_NONE
#define SL_MATTER_LOG_LEVEL SL_MATTER_LOG_NONE
#define EFR32_LOG_ENABLED 0
```

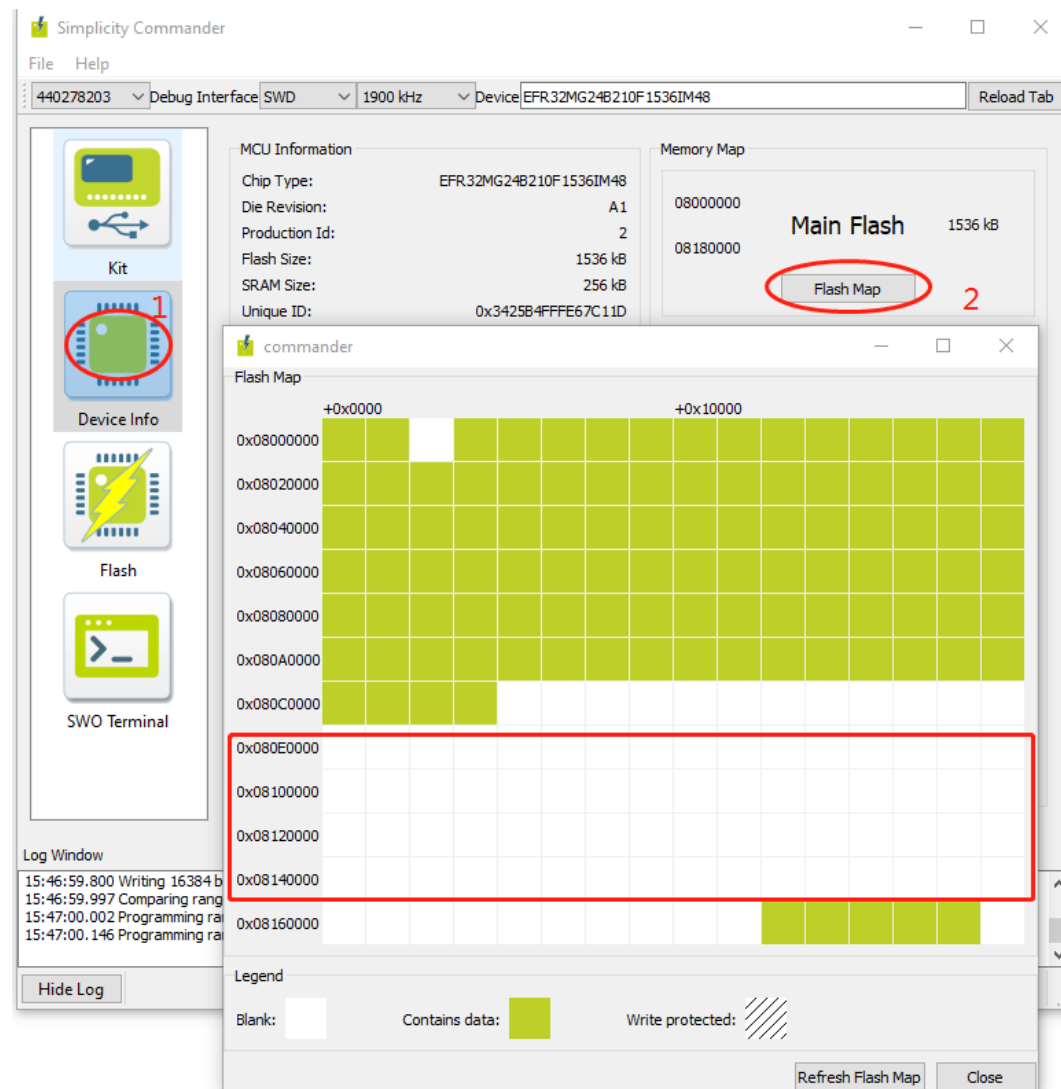
- 在`include/CHIPProjectConfig.h`中修改软件版本号为2

```
#ifndef CHIP_DEVICE_CONFIG_DEVICE_SOFTWARE_VERSION_STRING
#define CHIP_DEVICE_CONFIG_DEVICE_SOFTWARE_VERSION_STRING "0.2" //modify "0.1ALPHA" to "0.2"
#endif
#ifndef CHIP_DEVICE_CONFIG_DEVICE_SOFTWARE_VERSION
#define CHIP_DEVICE_CONFIG_DEVICE_SOFTWARE_VERSION 0x0002 //modify 0x0001 to 0x0002
#endif
```

- 修改完成后编译出V2版本的Matter灯固件

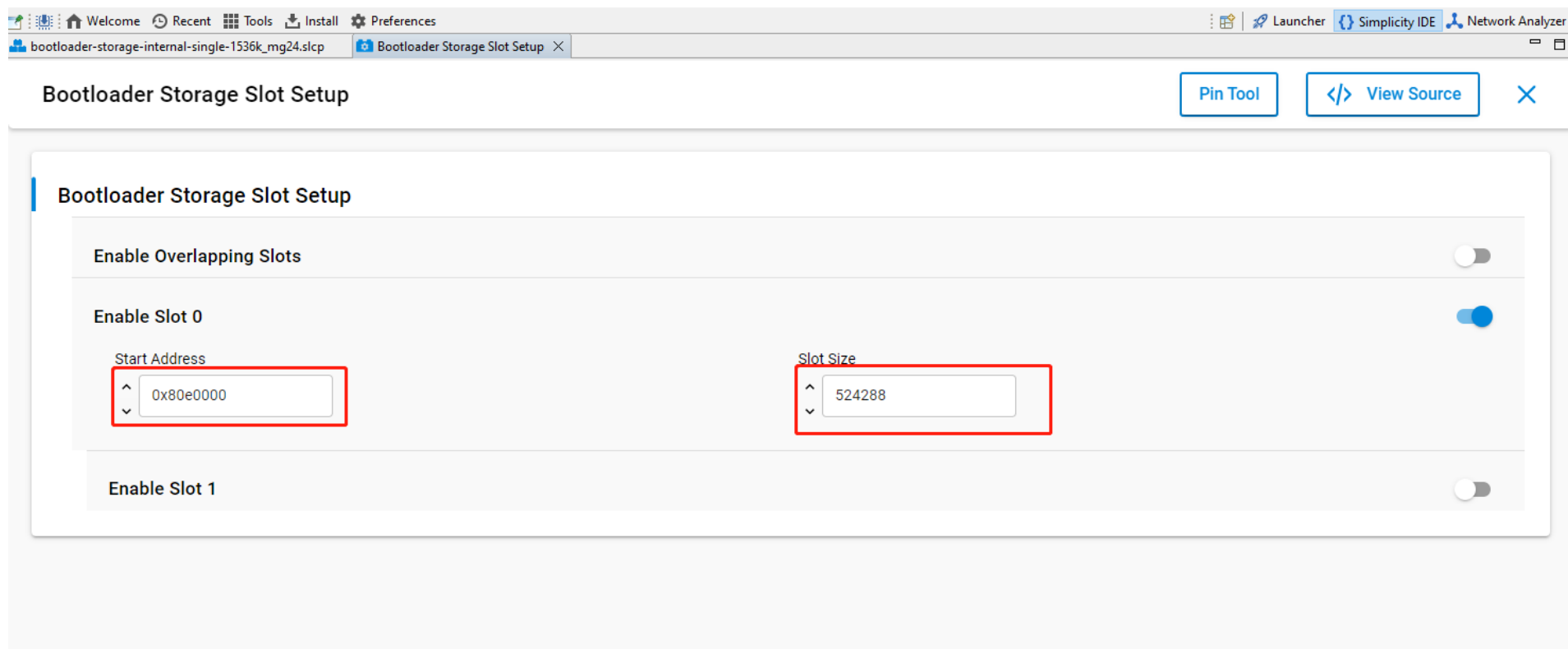
内部Flash OTA——烧录固件并查看Flash Map (2/7)

- 将V2版本烧录后可通过Simplicity Commander中Device Info页面的Flash Map查看片上Flash的使用情况
 - 这一步建议先擦除芯片再烧录，这样才会比较清晰的查看固件在Flash中的分部情况
- 如右图所示，我们可以利用0x080E0000 ~ 0x0815FFFF这段空间来存储OTA升级文件



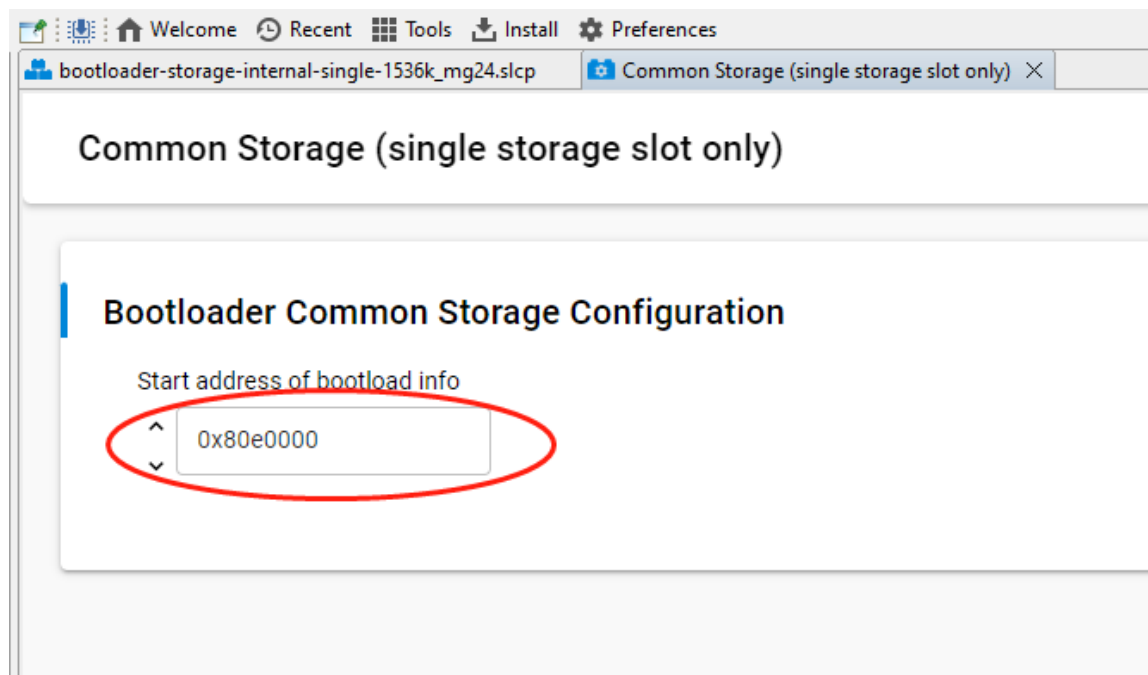
内部Flash OTA——创建Bootloader并添加OTA功能 (3/7)

- 参考实验手册创建“Bootloader - SoC Internal Storage(single image on 1536kB device)”工程
- 这里我们需要修改“Bootloader Storage Slot Setup”的参数
 - 具体参考上一步中查看Flash Map分配的空闲空间：这里我们选择0x080E0000开始的512KB大小的空间



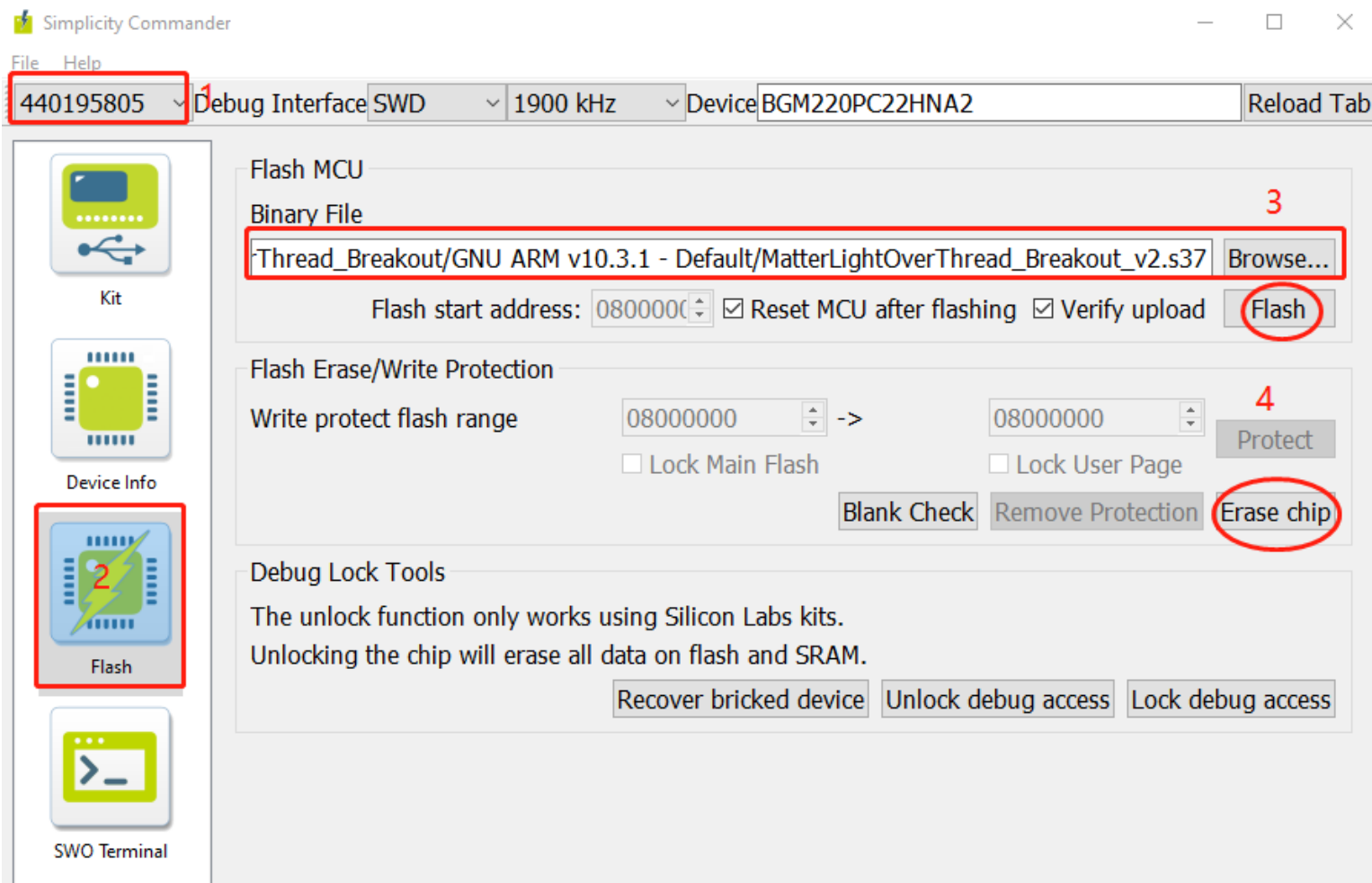
内部Flash OTA——创建Bootloader并添加OTA功能 (4/7)

- 这里我们需要修改“**Common Storage(Single storage slot only)**”的参数
 - 如果没安装此组件，我们需要先点击Install进行安装后再配置参数
 - 这里的Start address参数需要与上一步中的Start address保持一致
- 修改完成后保存并编译固件



内部Flash OTA——烧录v2版固件 (5/7)

- 重新烧录bootloader和v2版本固件到Matter灯设备中去
- 并确认v2版Matter灯是否在网
 - 比如确认一下使用chip-tool控制是否有效
 - 如果设备不在网，也可以参考实验手册删除缓存重新配网



内部Flash OTA——修改Matter灯固件V3 (6/7)

- 在`config/sl_matter_config.h`中修改日志配置

```
#define SL_MATTER_STACK_LOCK_TRACKING_MODE SL_MATTER_STACK_LOCK_TRACKING_NONE  
#define SL_MATTER_LOG_LEVEL SL_MATTER_LOG_NONE  
#define EFR32_LOG_ENABLED 0
```

- 在`include/CHIPProjectConfig.h`中修改软件版本号为3

```
#ifndef CHIP_DEVICE_CONFIG_DEVICE_SOFTWARE_VERSION_STRING  
#define CHIP_DEVICE_CONFIG_DEVICE_SOFTWARE_VERSION_STRING "0.3" //modify "0.2" to "0.3"  
#endif  
#ifndef CHIP_DEVICE_CONFIG_DEVICE_SOFTWARE_VERSION  
#define CHIP_DEVICE_CONFIG_DEVICE_SOFTWARE_VERSION 0x0003 //modify 0x0002 to 0x0003  
#endif
```

- 参考实验手册将**LED**组件中将其颜色改为蓝色
- 修改完成后编译出**V3**版本的**Matter**灯固件

内部Flash OTA——将s37文件转换成ota文件 (7/7)

- 通过**commander**工具将**s37**转换成**lzma**压缩的**gbl**文件

```
# 创建经lzma压缩压缩的gbl文件
$ commander gbl create ./MatterLightOverThread_Breakout_v3.gbl --app
./MatterLightOverThread_Breakout_v3.s37 --compress lzma
```

- 通过`ota_image_tool.py`脚本将**gbl**转换成**ota**文件（这一步需要在**ubuntu**环境中完成）

```
$ cd matter
# 通过gbl文件生成ota文件
$ ./src/app/ota_image_tool.py create -v 0xFFF1 -p 0x8005 -vn 3 -vs "0.3" -da sha256
../firmware/MatterLightOverThread_Breakout_v3_lzma.gbl
../firmware/MatterLightOverThread_Breakout_v3_lzma.ota
```

- 到这里设备端的准备工作已准备完毕

OTA升级操作——内容流程介绍

- 启动OTA-Provider服务
- 配置OTA-Provider
- 通知设备OTA升级



OTA升级操作——启动OTA - Provider服务 (1/3)

- 启动**OTA**服务，并指定**ota**升级文件

```
$ cd matter
```

```
# 在一个新的终端窗口启动OTA服务。指向ota文件
```

```
$ sudo ./out/debug/ota-provider/chip-ota-provider-app -f ../firmware/MatterLightOverThread_Breakout_v3_Izma.ota
```

OTA升级操作——配置OTA - Providereb服务 (2/3)

- 为OTA-Provider分配一个NODE_ID，并加入Matter网络

```
$ cd matter/out/debug/standalone/  
# 为OTA-Provider分配一个节点ID: 5678  
$ sudo ./chip-tool pairing onnetwork 5678 20202021
```

- 授予网络中所有设备操作OTA Provider cluster (0x0029)的权限

```
# 授予网络中所有设备操作OTA Provider cluster (0x0029)的权限。5678为OTA Provider的node id, 后面的0为  
endpoint。  
$ sudo ./chip-tool accesscontrol write acl '[{"fabricIndex": 1, "privilege": 5, "authMode": 2, "subjects": [112233],  
"targets": null}, {"fabricIndex": 1, "privilege": 3, "authMode": 2, "subjects": null, "targets": null}]' 5678 0
```

- 到这里OTA功能相关配置工作已经完成

OTA升级操作——通知Matter设备升级 (3/3)

- 通知**Matter**设备升级

```
# 通知Matter设备升级。5678为OTA Provider的node id, 1001为Matter Lighting的node id。  
$ sudo ./chip-tool otasoftwareupdaterequestor announce-ota-provider 5678 0 0 0 1001 0
```

- **Matter**设备开始升级时，可在**OTA-Provider**服务窗口看到相关日志信息

现在是实验时间.....

00 : 30 : 00

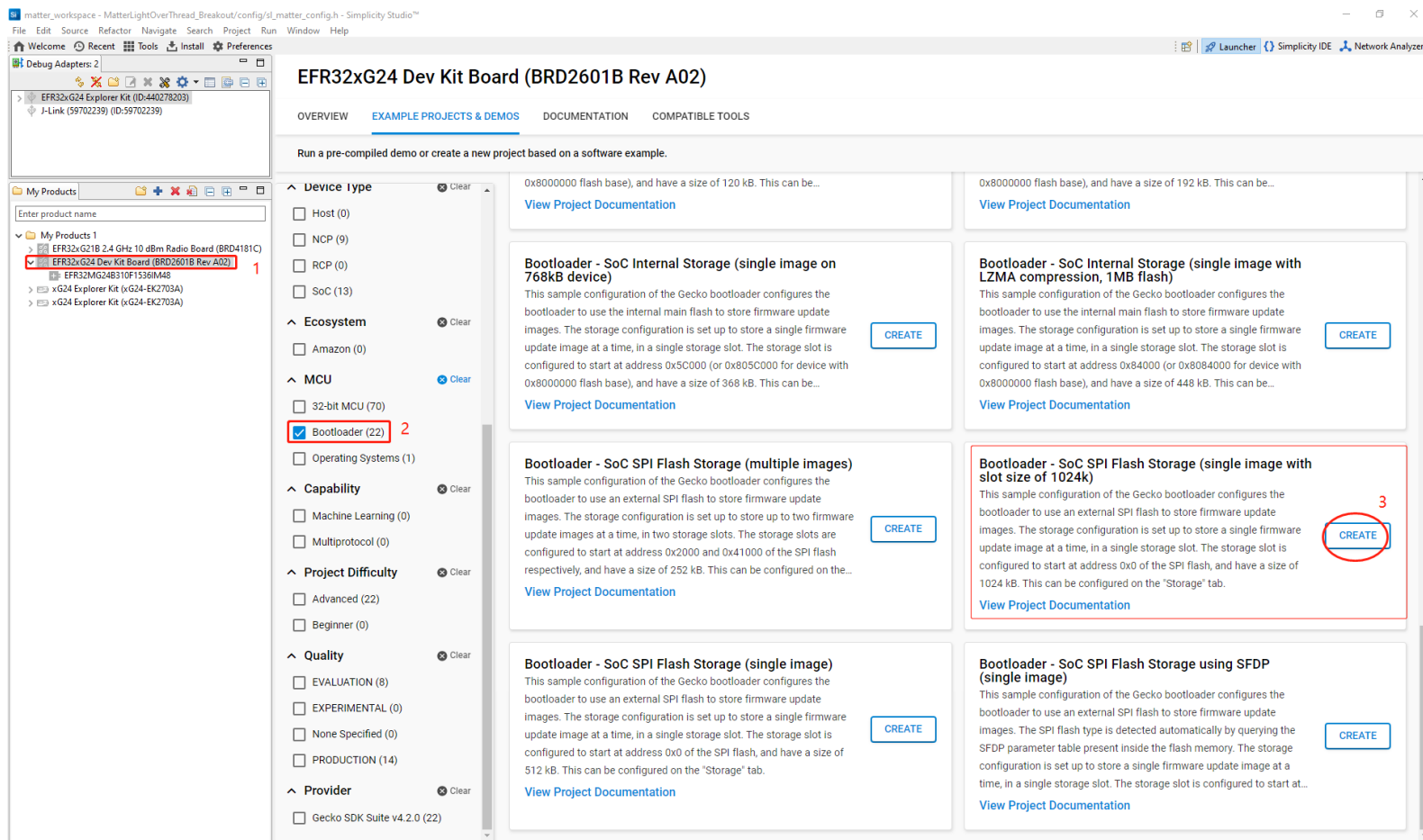


外部Flash OTA——内容流程介绍

- 修改Matter灯的固件（V2）
- 烧录Matter灯固件到设备中并查看Flash Map
- 创建一个支持外部Flash OTA功能的Bootloader
- 烧录Bootloader到Matter灯设备中
- 配置Matter灯入网
- 修改并编译一个新版本的Matter灯固件（V3）
- 将新版本的Matter灯固件（V3）以OTA方式升级到设备中

外部Flash OTA——创建Bootloader

- 参考下图选择“Bootloader-SoC SPI Flash Storage(single image with slot size of 1024k)”
- 注意事项：
 - 确保Flash型号已经支持
 - 可在“SPI Flash Storage”组件中查看
 - 是否需要支持LZMA压缩技术



现在是实验时间.....

00 : 10 : 00

Change Clock Type

Digital

Duration:

00

10

00

TimeUp Reminder (Optional):

--

--

--

Choose Sound Effect

None

Choose TimeUp Sound

None

☐ Enable Count Up ☐ Combine With Bar Clock

Start

Pause

Stop

Reset



参考资料

- 芯科科技**Simplicity-studio**集成开发环境: <https://www.silabs.com/developers/simplicity-studio>。
- 芯科科技开发者文档: <https://docs.silabs.com/>
- 芯科科技**Matter**方案介绍: <https://www.silabs.com/wireless/matter>
- 芯科科技**Matter**开发文档: <https://docs.silabs.com/matter/1.0.1/matter-start/>
- **Matter**协议规格书: <https://csa-iot.org/developer-resource/specifications-download-request/>
- **OpenThread**参考资料: <https://openthread.google.cn/>





—
谢谢!

Silicon Labs
官方网站



Silicon Labs
微信公众号



Silicon Labs
在线社区

