



实验1：基于芯科科技GSDK开发Matter灯和开关等设备

余发明

2023-02-21

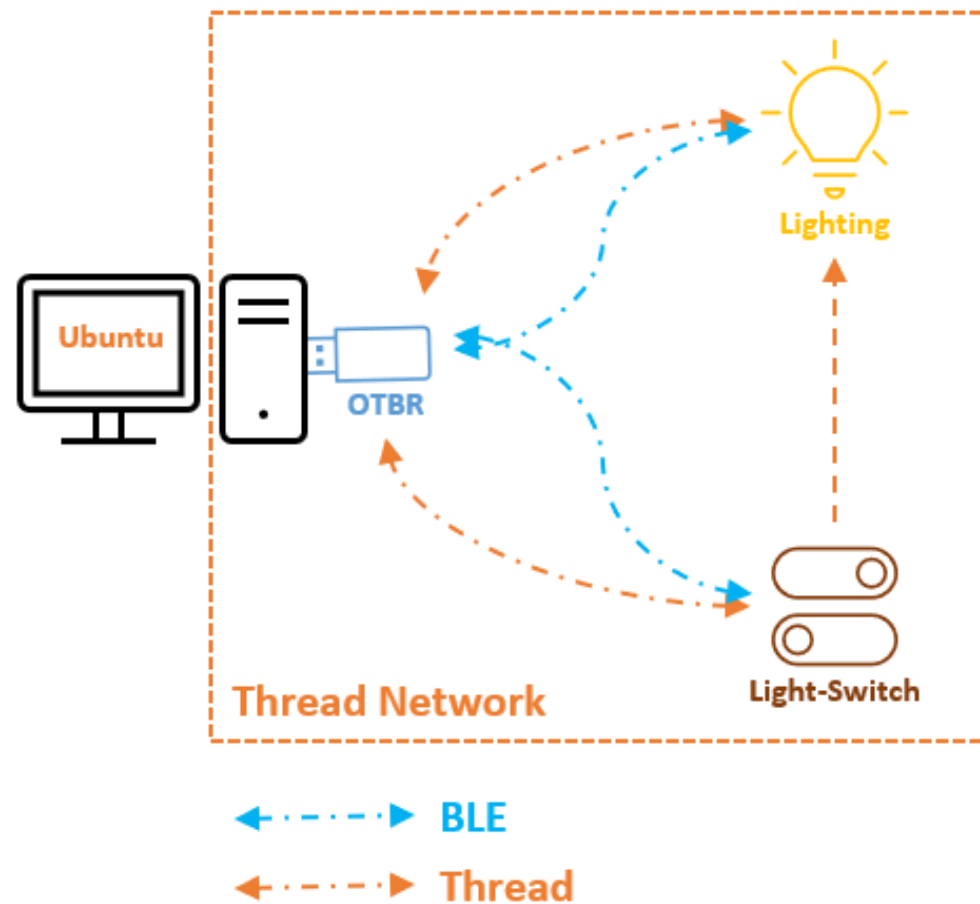


大纲

- 实验介绍
- 创建**Matter**相关工程、编译、烧录
 - 预留操作时间
- 测试**Matter**设备
 - 预留操作时间

实验介绍——整体概述

- 本实验将以**Matter over Thread**为例，搭建一个最基本的**Matter**网络。让开发者对**Matter**产品的开发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
 - Light-Switch: Matter开关
 - BLE: 用于Matter协议配网



实验介绍——预备知识

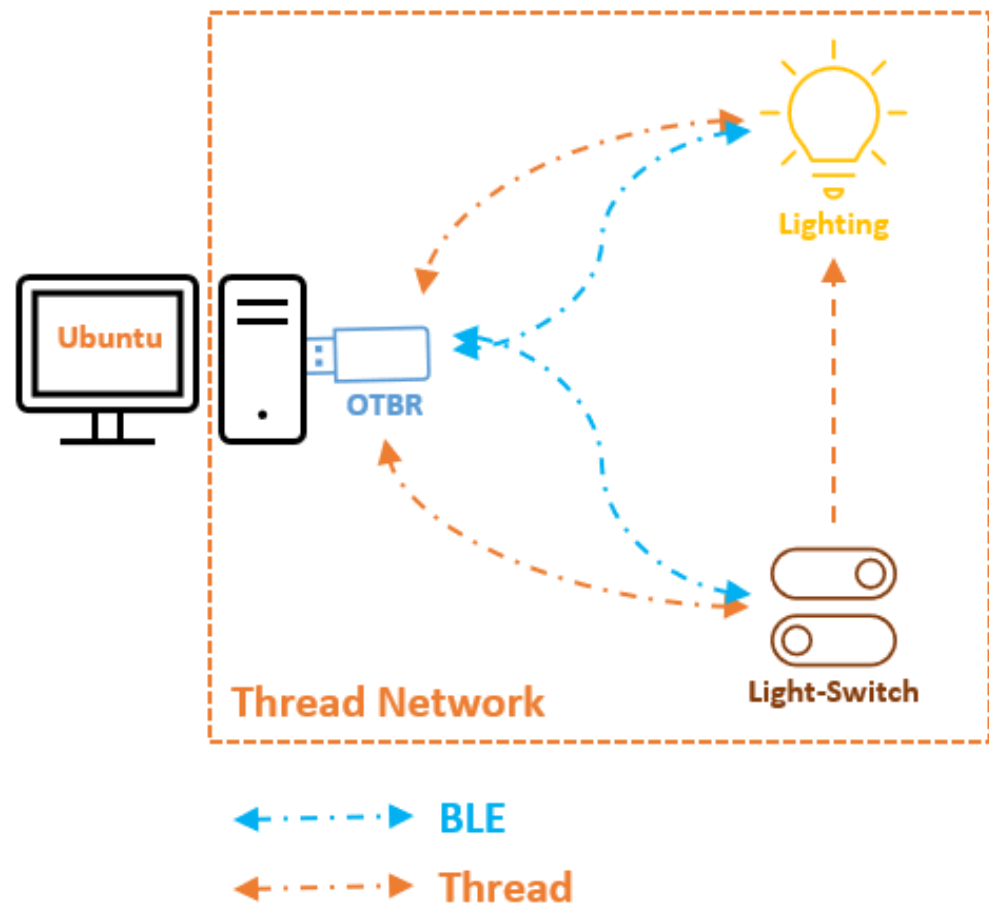
- 对**Ubuntu**开发环境有一定的使用基础
- 对**OpenThread**协议有一定的了解
- 对**C++**有一定的基础
- 对**Matter**协议有个初步认识
- 名词解释
 - OT-RCP: Open Thread Radio Co-Processor。Thread无线协处理器。
 - OTBR: Open Thread Board Router。Thread边界路由器。
 - chip-tool: Linux应用程序。用于Matter协议控制。
 - ot-ctl: Thread网络控制的应用程序

实验介绍——实验目的

- 熟悉芯科科技全线产品的开发环境**Simplicity Studio v5**
- 了解**Matter**，对**Matter**协议不再陌生
- 让开发者认为开发一全**Matter**设备的门槛也不是那么高
- 掌握如何基于芯科科技**IoT**平台开发**Matter**设备

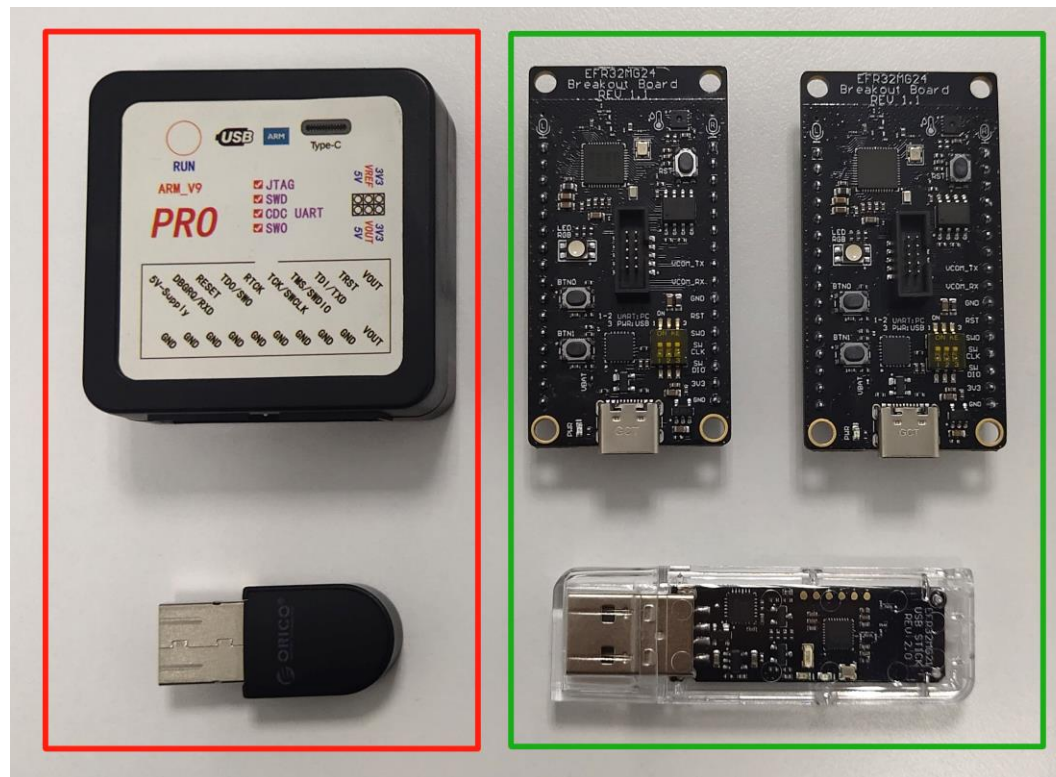
实验介绍——实验内容

- 基于SS5 GSDK开发一个Matter灯
- 基于SS5 GSDK开发一个Matter开关
- 基于SS5 GSDK开发一个bootloader
 - Matter设备的启动引导代码
- 基于SS5 GSDK开发一个OT-RCP
 - Thread边界路由器的无线协处理器固件
- 通过终端命令行控制Matter灯
- 使用Matter开关控制Matter灯



实验介绍——实验前准备

- 安装Ubuntu虚拟机的电脑一台
- EFR32MG24 Breakout Board两块
- 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



创建工程——Matter - Light over Thread

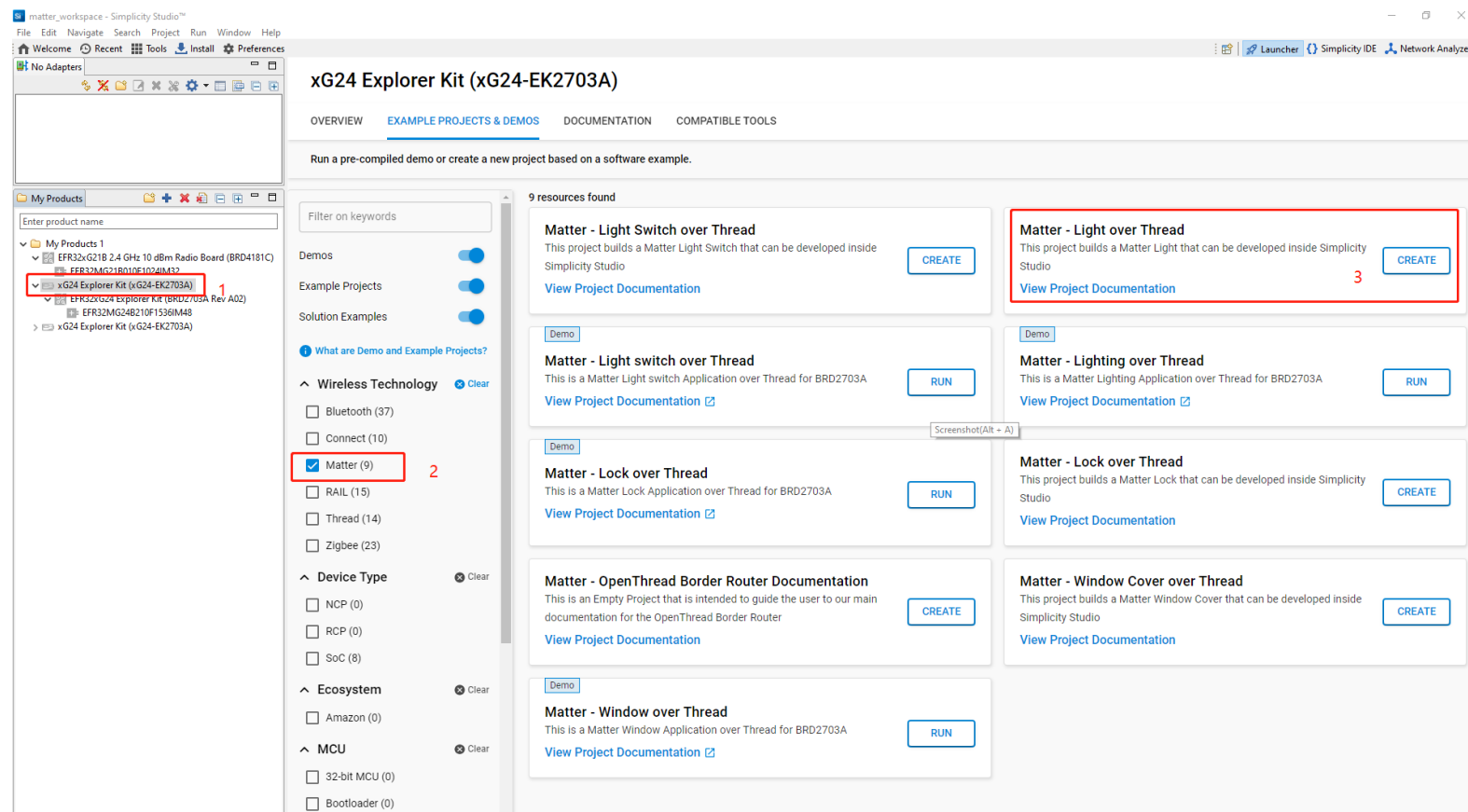
- 参考下图基于“BRD2601”开发板选择“Matter - Light over Thread”创建工程

- 按键说明

- BTN0：功能按键
 - ▶ 长按6秒进入配网状态
- BTN1：灯控按键

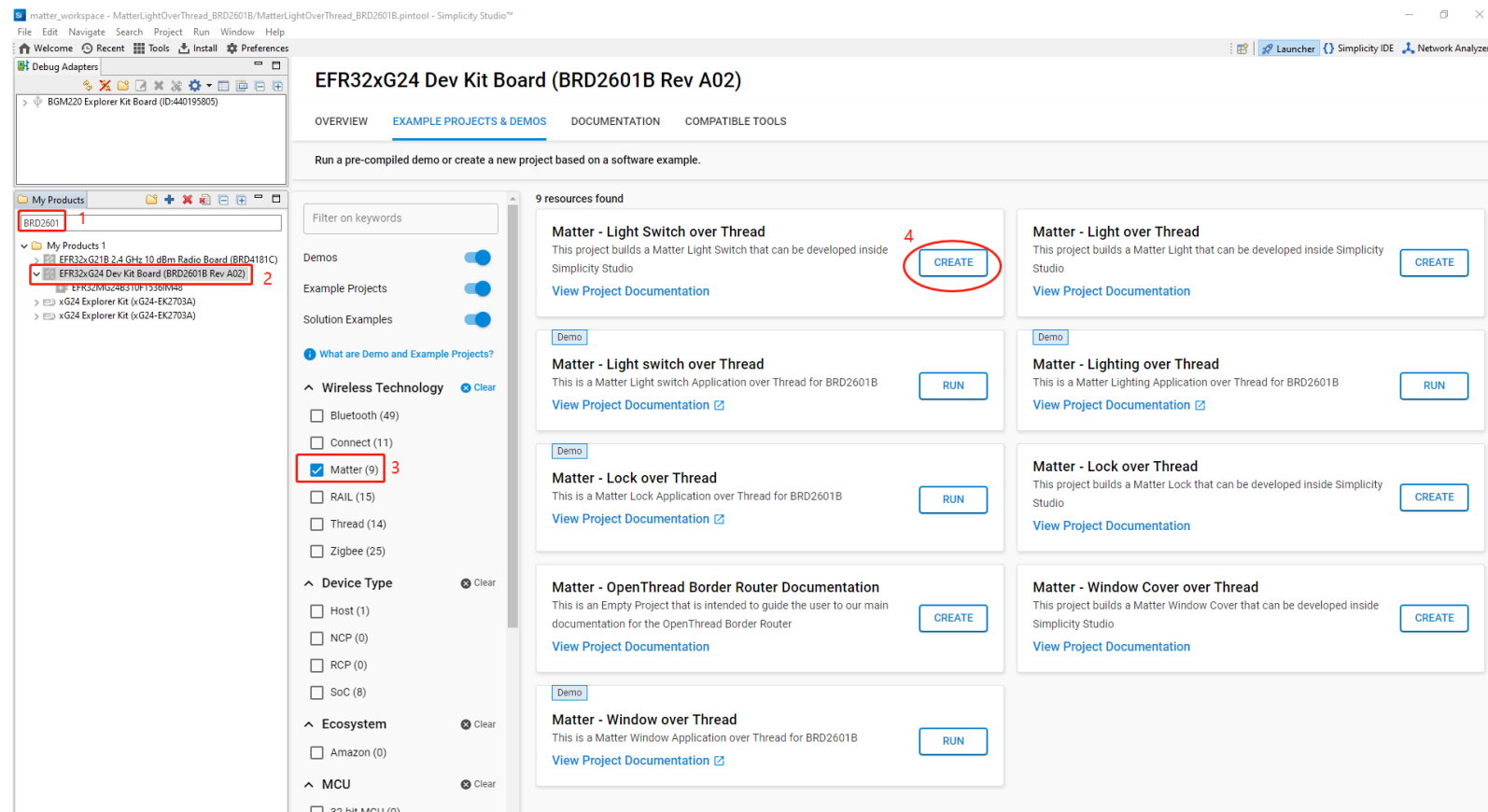
- LED说明

- 红色：功能指示灯
 - ▶ 入网后常亮
- 绿色：LED受控灯



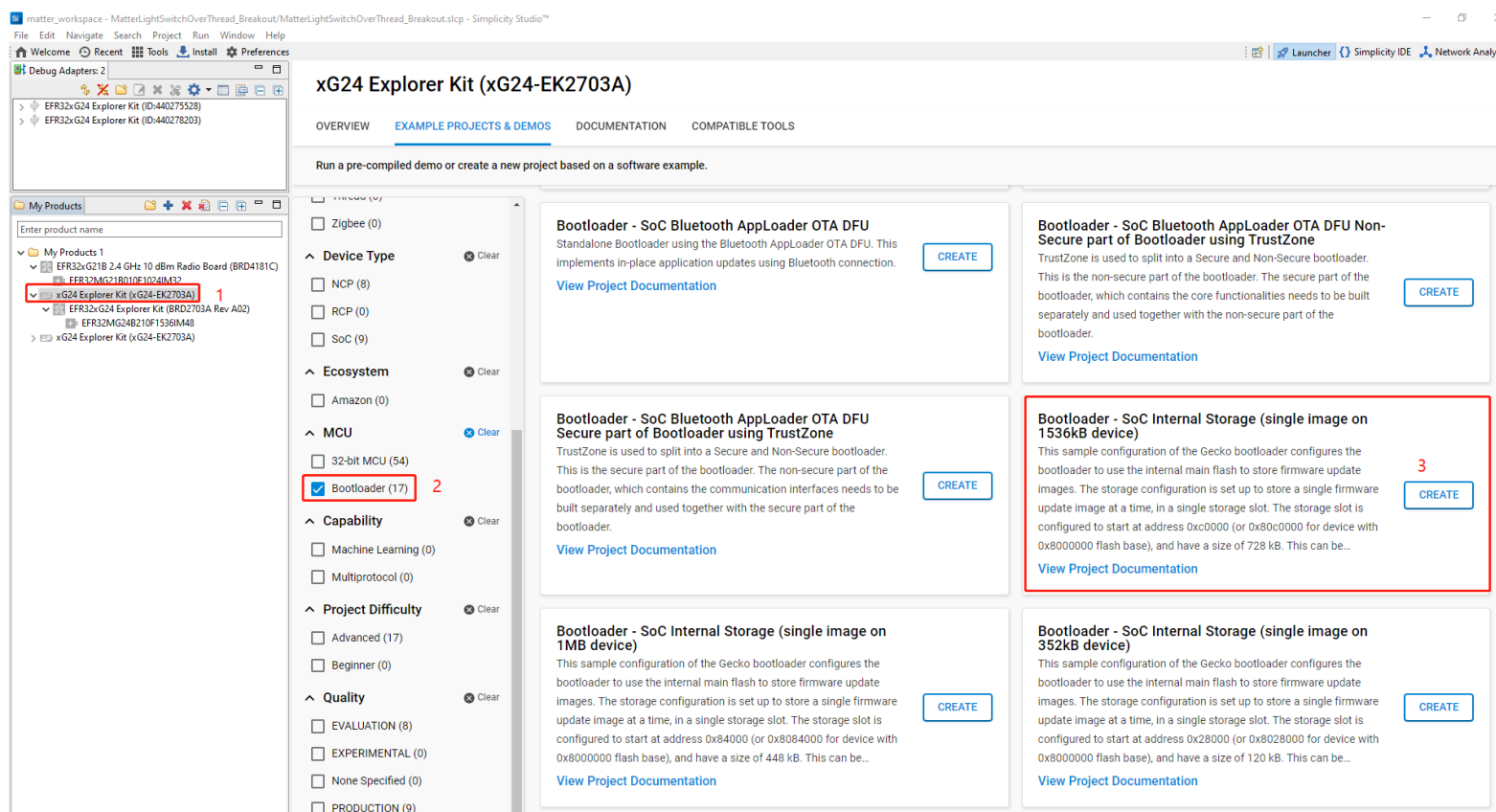
创建工程——Matter - Light Switch over Thread

- 参考下图基于“BRD2601”开发板选择“Matter – Light Switch over Thread”创建工程
- 按键说明
 - BTN0: 功能按键
 - BTN1: 灯控按键
- LED说明
 - 红色: 功能指示灯



创建工程——Bootloader - SoC Internal Storage(single image on 1536kB device)

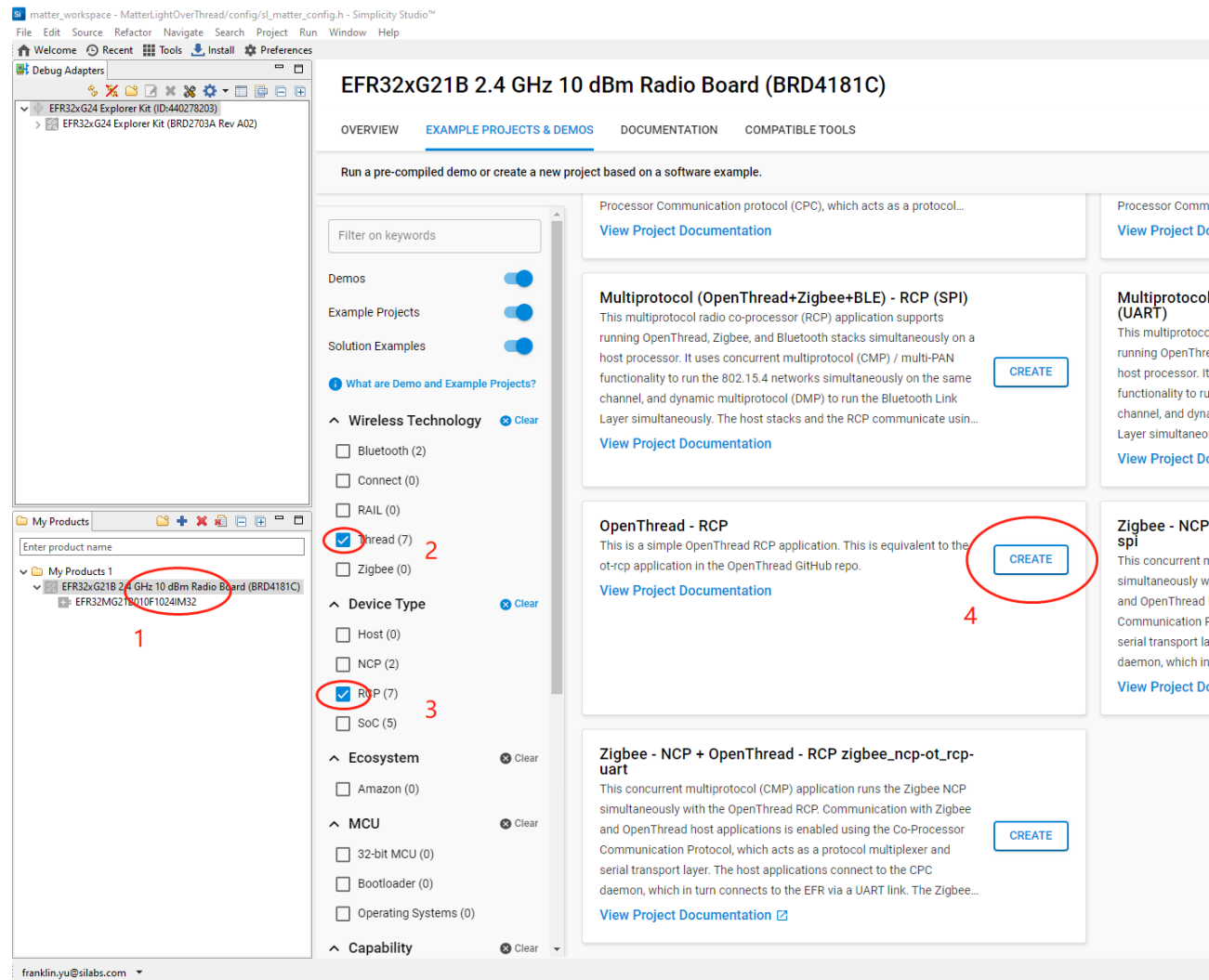
- 参考下图基于“BRD2601”开发板选择“Bootloader - SoC Internal Storage(single image on 1536kB device)”创建工程
- 按照默认配置编译即可



创建工程——OpenThread - RCP

- 参考下图选择“OpenThread - RCP”创建工程
 - 注意这里需要选择MG21的BRD4181C开发板进行选择。
- 添加LED控件
 - LED配置PD03
- 添加x-modem功能
 - 通过x-modem协议进行升级
- 为bootloader功能修改链接脚本
 - 修改flash配置

EFR32xG21B 2.4 GHz 10 dBm Radio Board (BRD4181C)



烧录固件

- 通过前面步骤，我们可以得到下面这些固件
- 将其他分别烧录到对应的开发板中，就可以开始**Matter**功能的验证了
- 具体请大家参考实验手册“实验1：基于芯科科技**GSDK**开发**Matter**灯和开关等设备”完成

固件	目标板	说明
MatterLightOverThread_Breakout.s37	EFR32MG24 Breakout Board REV 1.1	Matter灯
MatterLightSwitchOverThread_Breakout.s37	EFR32MG24 Breakout Board REV 1.1	Matter开关
ot-rpc-mg21-usbstick.gbl	EFR32MG21 USB STICK REV 2.0	OTBR固件（RCP模式）
bootloader-storage-internal-single-1536k_mg24.s37	EFR32MG24 Breakout Board REV 1.1	Bootloader

现在是实验时间.....

00 : 30 : 00

Change Clock Type

Digital

Duration: 00 30 00

TimeUp Reminder (Optional): -- -- --

Choose Sound Effect None

Choose TimeUp Sound None

☐ Enable Count Up ☐ Combine With Bar Clock

Start

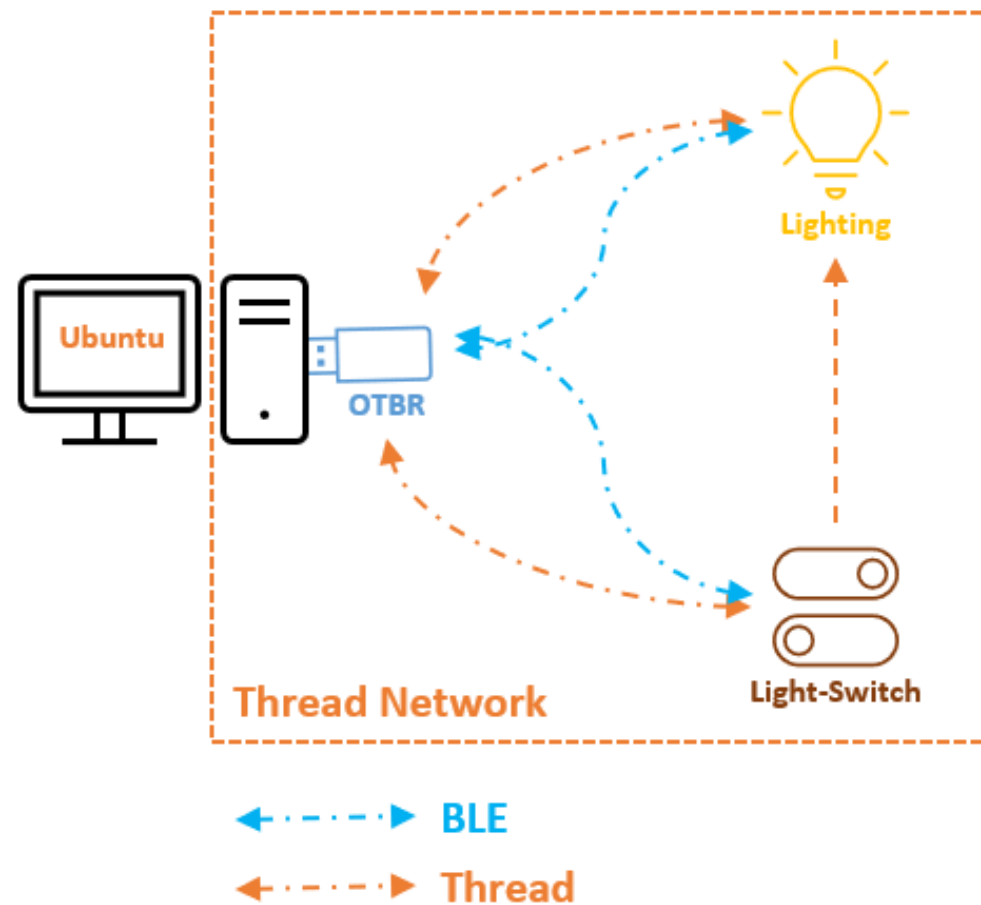
Pause

Stop

Reset

测试Matter设备

- 运行otbr-agent
- 创建Threa网络
- 将Matter灯添加到网络中
- 通过命令行指令控制Matter灯
- 将Matter开关添加到网络中
- 在Matter灯和Matter开关之间建立绑定关系
- 使用Matter开关控制Matter灯



测试Matter设备——运行otbr-agent (1/7)

- 配置otbr-agent参数

```
$ sudo vi /etc/default/otbr-agent
```

```
# 虚拟机(网络端口为enp0s3, USB Stick端口为/dev/ttyUSB0)
```

```
OTBR_AGENT_OPTS="-I wpan0 -B enp0s3 spinel+hdlc+uart:///dev/ttyUSB0?uart-baudrate=115200"
```

```
# 树莓派
```

```
OTBR_AGENT_OPTS="-I wpan0 -B eth0 spinel+hdlc+uart:///dev/ttyUSB0?uart-baudrate=115200"
```

- otbr-agent服务启动与停止

```
# 启动`otbr-agent`服务
```

```
$ sudo systemctl start otbr-agent.service
```

```
# 停止`otbr-agent`服务
```

```
$ sudo systemctl stop otbr-agent.service
```


测试Matter设备——创建Tthread网络 (2/7)

- 不理解的可以参考**OpenThread**相关知识

创建Thread网络

```
$ sudo ot-ctl dataset init new
```

```
$ sudo ot-ctl dataset networkkey 00112233445566778899aabbccddeeff
```

```
$ sudo ot-ctl dataset extpanid 1111111122222222
```

```
$ sudo ot-ctl dataset panid 0x1234
```

```
$ sudo ot-ctl dataset channel 15
```

将以上配置提交为活动配置

```
$ sudo ot-ctl dataset commit active
```

打开ipv6接口

```
$ sudo ot-ctl ifconfig up
```

启动Thread协议

```
$ sudo ot-ctl thread start
```

查看Thread网络配置

```
$ sudo ot-ctl dataset active -x
```

```
0e0800000000000010000000300000f35060004001fffe0020811111111222222220708fdb0ab694c9b3a1  
7051000112233445566778899aabbccddeeff030f4f70656e5468726561642d66366361010212340410d  
237761823728dd2cbfe64f477b38b4c0c0402a0f7f8
```

```
Done
```

测试Matter设备——将Matter灯添加到网络中 (3/7)

- 为Matter灯分配NODE_ID为1001
- 通过chip-tool将Matter灯添加到前面创建的Thread网络中

```
# 为Matter灯分配NODE_ID为1001
$ sudo ./chip-tool pairing ble-thread 1001
hex:0e0800000000000010000000300000f35060004001fffe002081111111222222220708fd67d3ca68d
beac6051000112233445566778899aabbccddeeff030f4f70656e5468726561642d30653764010212340
410b58c67a8a3aaa68557be489b35798ad60c0402a0f7f8 20202021 3840
```

测试Matter设备——通过编程命令行控制Matter灯（4/7）

- 待Matter灯入网成功
- 使用chip-tool的onoff cluster对Matter灯进行控制

```
# 翻转Matter灯
$ sudo ./chip-tool onoff toggle 1001 1

# 打开Matter灯
$ sudo ./chip-tool onoff on 1001 1

# 关闭Matter灯
$ sudo ./chip-tool onoff off 1001 1

# 读取Matter灯的当前状态
$ sudo ./chip-tool onoff read on-off 1001 1
```

测试Matter设备——将Matter开关添加到网络中 (5/7)

- 为Matter开关分配NODE_ID为1002
- 通过chip-tool将Matter开关添加到前面创建的Thread网络中

```
# 为Matter开关分配NODE_ID为1002
$ sudo ./chip-tool pairing ble-thread 1002
hex:0e0800000000000010000000300000f35060004001fffe00208111111122222220708fd67d3ca68d
beac6051000112233445566778899aabbccddeeff030f4f70656e5468726561642d30653764010212340
410b58c67a8a3aaa68557be489b35798ad60c0402a0f7f8 20202021 3840
```

测试Matter设备——在Matter灯和Matter开关之间建立绑定关系 (6/7)

- 待**Matter**开关入网成功
- 使用**chip-tool**设置**Matter**的**ACL**权限，让**Matter**开关有控制权

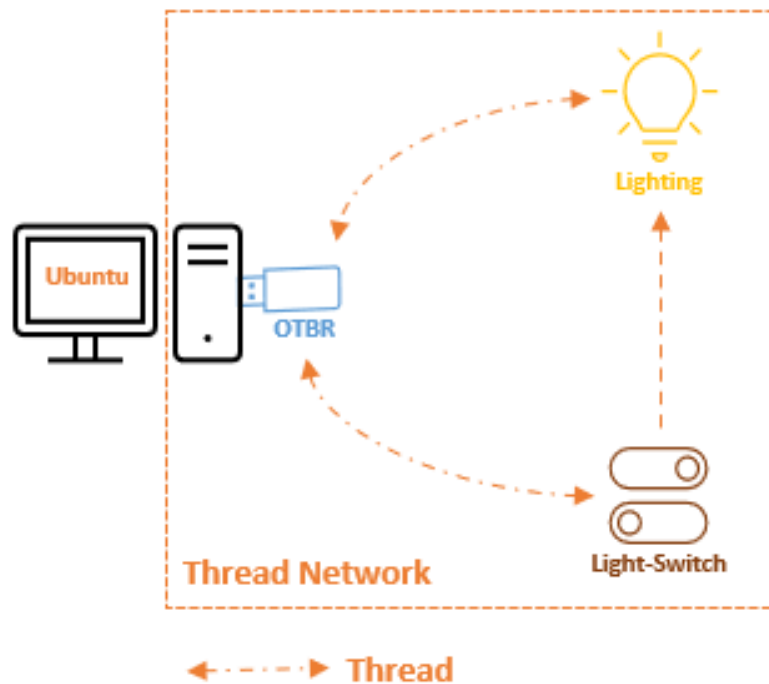
```
# 设置灯的ACL，让开关可以控制它
$ sudo ./chip-tool accesscontrol write acl '[{"fabricIndex":1, "privilege":5, "authMode":2,
"subjects":[112233, 1002], "targets":null}]' 1001 0
```

- 在**Matter**灯和**Matter**开关之间建议绑定关系

```
# 让开关绑定灯
$ sudo ./chip-tool binding write binding '[{"fabricIndex":1, "node":1001, "endpoint":1, "cluster":6}]' 1002
1
```

测试Matter设备——使用Matter开关控制Matter灯（7/7）

- 完成以前面6个步骤后，就可以使用**Matter**开关来控制**Matter**灯的亮灭了。



现在是实验时间.....

00 : 10 : 00

^

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

