在虚拟机上运行 OTBR 及 Bridge

实验目的

本实验将在 Ubuntu 上创建一个 Open Thread 的边界路由器。参加培训的用户也可以用我们提供的在 Raspberry PI 4B 上 提供的已经创建好的 边界路由器，对 matter 设备进行 commissioning 或者其它控制操作。完成 边界路由器的创建后，我们还实现了一个 bridge 设备，并在 bridge 的一些 endpoint 上，虚拟了一些 灯 以及 传感器设备。

主要特点

- 创建并运行 Open Thread 边界路由器
- 编译并运行 Bridge-app

硬件准备

PC/Laptop 电脑一台
MG24 Breakout开发板两块
MG21 USB Stick (用作OTBR 的RCP)
J-Link烧录器一台 (演示中将采用 BRD4001A 的 demo 板)
BLE Dongle一块 （演示中用 ORICO/奥睿科蓝牙5.0适配器）
树莓派一台（可选）

软件准备

用户也可以按照以下步奏 创建 边界路由器:
Win10 上的 VirtualBox （VirtualBox-6.1.38-153438-win.exe 可下载）
Linux环境：Ubuntu
默认用户名和密码都为：ubuntu
如果是桌面环境，也可以安装Simplicity Studio和 Simplicity Commander (用于烧入 RCP)
集成了Matter SDK和开发环境，参考之前的培训，编译得到 chip-efr32-lighting-app-v1.0.2.s37
芯科科技将提供一个 Virtual Box ubuntu 镜像，请下载 安装 （<VirtualBox-6.1.38-153438-win.exe > 以及 <ubuntu-matter.ova>）

Rufus 用于在 PC/Laptop 上，将 SilabsMatterPi.img 烧如 TF 卡，用于 RaspberryPi 4B
Putty 在 Laptop 上用于远程登陆 Raspberry PI 4B 的 UBUNTU

开始实验

在 Windows 安装 Virtual Box, 并将 ubuntu-matter.ova 导入到 Virtual Box 中
将 USB OT-RCP 插入 PC/Laptop 的一个接口
BLE Dongle 插入到 PC/Laptop 上的另外一个 USB 上
用 commander 将编译好的 灯的 固件烧入到 mg24 开发板上
启动 Ubuntu
待 Ubuntu 启动完成后，在 Virtual Box 的 Device 将 OT-RCP 以及 BLE DONGLE 连接到 Virtual Box 中的 Ubuntu 上
检查 Ubuntu 上是否 存在 /dev/ttyUSB0 或者 /dev/ttyACM0
在 Ubuntu 上，用命令 sudo hciconfig -a 检查 BLE DONGLE 是否正常

实验步骤

1. Ubuntu 环境依赖安装

```
$ sudo apt-get install git gcc g++ pkg-config libssl-dev libdbus-1-dev \
libglib2.0-dev libavahi-client-dev ninja-build python3-venv python3-dev \
python3-pip unzip libgirepository1.0-dev libcairo2-dev libreadline-dev
```

```
# 安装实用工具
$ sudo apt-get install net-tools openssh-server openssh-client

# 启用双向复制粘贴功能
$ sudo apt-get install virtualbox-guest-x11

$ sudo VBoxClient --clipboard
```

2. Matter SDK环境

从silicon labs官方github获取Matter代码：

```
$ git clone https://github.com/SiliconLabs/matter.git
$ cd matter
# 同步子模块（受网速影响，可能需要比较长的时间）
```

```
$ ./scripts/checkout_submodules.py --shallow --recursive --platform efr32
# 检测并完善编译环境（对网络有要求，需要访问国外网站。受网速影响，可能需要约1小时左右）
$ source scripts/activate.sh
# 编译 chip-tool

$ ./scripts/examples/gn_build_example.sh examples/chip-tool/ out/standalone/chip-tool/
```

3. 编译 OTBR

```
$ git clone https://github.com/openthread/ot-br-posix
$ cd ot-br-posix
$ ./script/bootstrap
$ INFRA_IF_NAME=enp0s8  ./script/setup (可以用 ifconfig 获取网口名称)
# 运行otbr-agent

 $ sudo vi /etc/default/otbr-agent

# 虚拟机(网络端口为enp0s8，USB Stick端口为/dev/ttyUSB0)
OTBR_AGENT_OPTS="-I wpan0 -B enp0s8 spinel+hdlc+uart:///dev/ttyUSB0?uartbaudrate=115200"
```

4. 启动otbr-agent 服务

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

#后续步骤中的指令仅作为参考，不是必须的。

5. 检查： /var/log/syslog 输出 otbr-agent 的运行日志

```
 $ tail -f /var/log/syslog
```

6. 检查主机与RCP的连接状态

```
$ sudo ot-ctl state

# 如果连接失败，会返回以下信息
connect session failed: No such file or directory
```

7. 检查ot-br-posix 的版本号

```
$ sudo ot-ctl version

OPENTHREAD/2550699; POSIX; Feb  1 2023 11:04:47
Done
```

8. 检查ot-rcp 的固件版本号

$ sudo ot-ctl rcp version

SL-OPENTHREAD/2.2.0.0_GitHub-91fa1f455; EFR32; Jan 15 2023 19:10:53
Done

9. 停止otbr-agent 服务

$ sudo systemctl stop otbr-agent.service

10. 使用工具 ot-ctl 创建 Thread 网络

$ sudo ot-ctl

> dataset init new

> prefix add fd11:22::/64 paros

> thread stop

> ifconfig down

> ifconfig up

> thread start

(Alternatively: you can start WebGUI and start the Thread network)

- Notedown the following information from CLI

> dataset active -x <operationalkey>

> extpanid

11. 查看 Thread 网络配置

$ sudo ot-ctl dataset active -x
0e080000000000010000000300000f35060004001fffe00208111111112222222220708fdb0ab
694c9b3a17051000112233445566778899aabbccddeeff030f4f70656e5468726561642d6636
6361010212340410d237761823728dd2cbfe64f477b38b4c0c0402a0f7f8

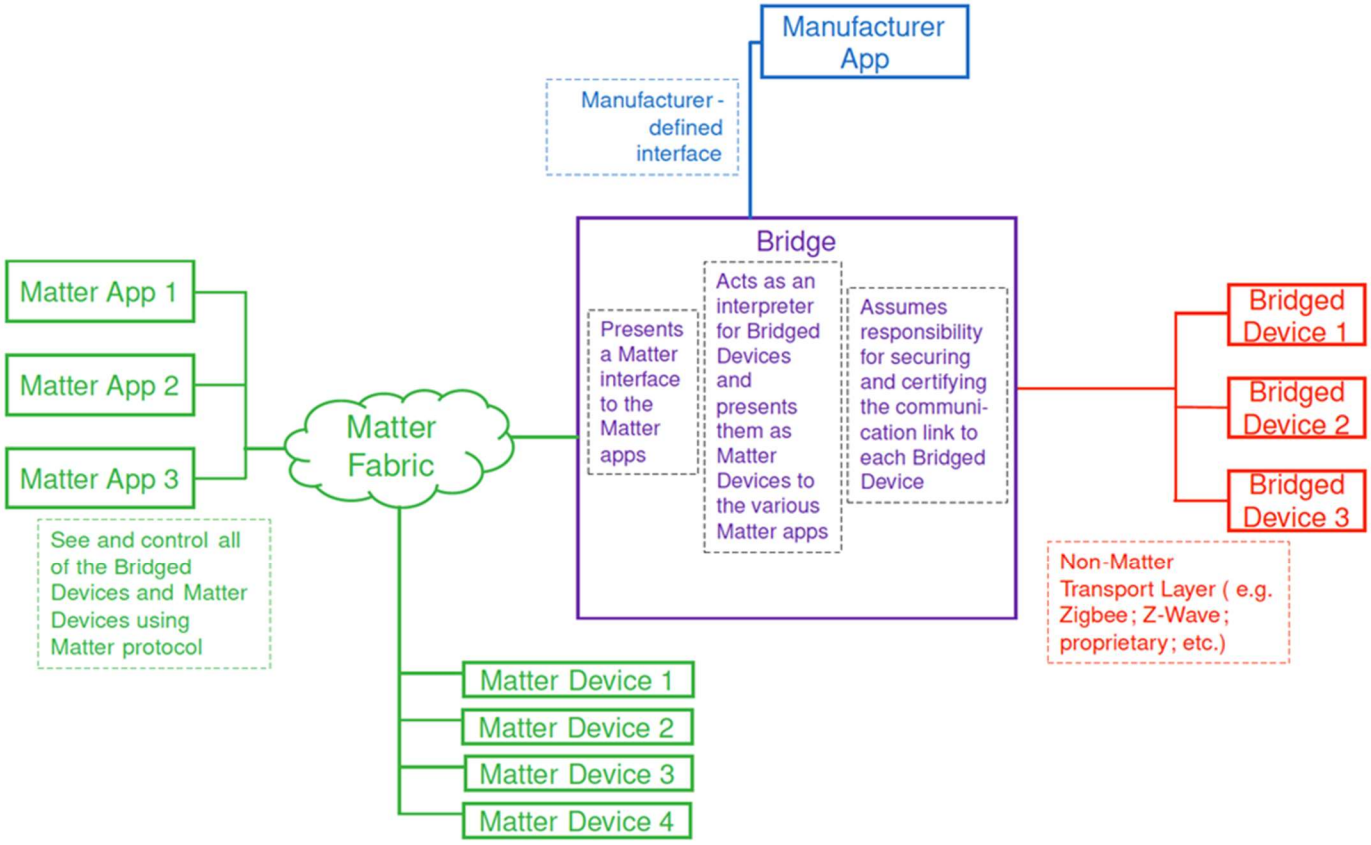
12. 使用chip-tool配置设备入网

配置Matter灯入网

```
$ cd ~/matter/out/standalone/chip-tool
# ./chip-tool pairing ble-thread ${NODE_ID} hex:${DATASET} ${PIN_CODE}  ${DISCRIMINATOR}

$ ./chip-tool pairing ble-thread 1001\
hex:0e080000000000010000000300000f35060004001fffe00208111111112222222220708fd67d3\
ca68dbeac60510001122334455667788 99aabbccddeeff030f4f70656e5468726561642d30653764\
010212340410b58c67a8a3aaa68557be489b35798ad60c0402a0f7f8 20202021 3840

# 上面指令运行正常的话，最后会看到类似这样的信息
[1673925106.632759][2792:2792] CHIP:DL: Inet Layer shutdown
[1673925106.632762][2792:2792] CHIP:DL: BLE shutdown
[1673925106.632764][2792:2792] CHIP:DL: System Layer shutdown

./chip-tool onoff toggle 1001 1
```

12. Matter 上的 bridge-app

1. Bridge把非Matter的IoT设备带到Matter Fabric里,同在一个Matter Fabric里面的Matter节点就可以跟这些非Matter的IoT设备进行交互
2. Bridge在这里扮演Matter和其他非Matter协议之间翻译的角色
3. 用户可参考Matter代码里面的 bridge-app



目前的 Matter 代码里面的 Dynamic Bridge App

Bridge App 作为一个虚拟 Matter 设备。每个 ZigBee 设备会被映射成为这个虚拟设备的一个端点 （Endpoint）。Bridge App 除了要有端点 0 也就是根节点（Root Node Endpoint），也需要为给每个非 Matter 网络建立一个桥接汇聚设备（Aggregator）端点，也就是 endpoint 1

根节点（Root Node Endpoint）：

Descriptor Cluster

DeviceTypeList: Root Node

PartList: EP 1, 11, 12, 13, 14, 15....m

Basic Information

桥接汇聚设备（Aggregator）端点

Descriptor Cluster

DeviceTypeList: Aggregator

PartList: EP 11, 12, 13, 14, 15....m

桥接端点（Bridged Endpoints）

Descriptor Cluster

Device Type List

Bridged Device Basic Information Cluster

Node Label

编译 bridge-app

$ cd ~/matter/examples/bridge-app/linux

$ git submodule update --init

$ source third_party/connectedhomeip/scripts/activate.sh

$ gn gen out/debug

$ ninja -C out/debug

配置 bridge-app 入网

1. PC/LAPTOP 上运行 编译的 chip-bridge-app
2. Raspberry Pi 上运行 chip-tool

   ./chip-tool pairing onnetwork 1245 20202021

获取 bridge-app 设备上 根节点上的端口列表

./chip-tool descriptor read parts-list 1245 0

```
[1676278476.209844][13465:13470] CHIP:TOO: Endpoint: 0 Cluster: 0x0000_001D Attribute 0x0000_0003 DataVersion: 3030698757
[1676278476.209908][13465:13470] CHIP:TOO:   PartsList: 12 entries
[1676278476.209939][13465:13470] CHIP:TOO:     [1]: 1
[1676278476.209963][13465:13470] CHIP:TOO:     [2]: 3
[1676278476.209986][13465:13470] CHIP:TOO:     [3]: 4
[1676278476.210009][13465:13470] CHIP:TOO:     [4]: 5
[1676278476.210032][13465:13470] CHIP:TOO:     [5]: 6
[1676278476.210054][13465:13470] CHIP:TOO:     [6]: 7
[1676278476.210077][13465:13470] CHIP:TOO:     [7]: 8
[1676278476.210100][13465:13470] CHIP:TOO:     [8]: 9
[1676278476.210122][13465:13470] CHIP:TOO:     [9]: 10
[1676278476.210145][13465:13470] CHIP:TOO:     [10]: 11
[1676278476.210168][13465:13470] CHIP:TOO:     [11]: 12
[1676278476.210191][13465:13470] CHIP:TOO:     [12]: 13
```

获取 bridge-app 设备的汇聚端口上 （endpoint 1） 汇聚的端口列表

./chip-tool descriptor read parts-list 1245 1

```
[1676272702.512284][10444:10454] CHIP:TOO: Endpoint: 1 Cluster: 0x0000_001D Attribute 0x0000_0003 DataVersion: 2449835073
[1676272702.512344][10444:10454] CHIP:TOO:   PartsList: 11 entries
[1676272702.512370][10444:10454] CHIP:TOO:     [1]: 3
[1676272702.512391][10444:10454] CHIP:TOO:     [2]: 4
[1676272702.512410][10444:10454] CHIP:TOO:     [3]: 5
[1676272702.512429][10444:10454] CHIP:TOO:     [4]: 6
[1676272702.512449][10444:10454] CHIP:TOO:     [5]: 7
[1676272702.512468][10444:10454] CHIP:TOO:     [6]: 8
[1676272702.512509][10444:10454] CHIP:TOO:     [7]: 9
[1676272702.512530][10444:10454] CHIP:TOO:     [8]: 10
[1676272702.512550][10444:10454] CHIP:TOO:     [9]: 11
[1676272702.512569][10444:10454] CHIP:TOO:     [10]: 12
[1676272702.512589][10444:10454] CHIP:TOO:     [11]: 13
```

获取 Vendor Name

./chip-tool basic read vendor-name 1245 0

```
[1676274854.693956][11707:11712] CHIP:DMG:                                AttributePathIB =
[1676274854.693988][11707:11712] CHIP:DMG:                                {
[1676274854.694028][11707:11712] CHIP:DMG:                                        Endpoint = 0x0,
[1676274854.694069][11707:11712] CHIP:DMG:                                        Cluster = 0x28,
[1676274854.694103][11707:11712] CHIP:DMG:                                        Attribute = 0x0000_0001,
[1676274854.694142][11707:11712] CHIP:DMG:                                }
[1676274854.694172][11707:11712] CHIP:DMG:
[1676274854.694215][11707:11712] CHIP:DMG:                                Data = "TEST_VENDOR" (11 chars),
[1676274854.694245][11707:11712] CHIP:DMG:                            },
[1676274854.694282][11707:11712] CHIP:DMG:
[1676274854.694303][11707:11712] CHIP:DMG:                    },
[1676274854.694328][11707:11712] CHIP:DMG:
[1676274854.694355][11707:11712] CHIP:DMG:            ],
[1676274854.694381][11707:11712] CHIP:DMG:
[1676274854.694401][11707:11712] CHIP:DMG:            SuppressResponse = true,
[1676274854.694435][11707:11712] CHIP:DMG:            InteractionModelRevision = 1
[1676274854.694455][11707:11712] CHIP:DMG: }
[1676274854.694591][11707:11712] CHIP:TOO: Endpoint: 0 Cluster: 0x0000_0028 Attribute 0x0000_0001 DataVersion: 4235857191
[1676274854.694632][11707:11712] CHIP:TOO:   VendorName: TEST_VENDOR
```

获取 bridge-app 的产品名称

./chip-tool basic read product-name 1245 0

```
[1676275052.503001][11810:11815] CHIP:DMG:                      AttributeDataIB =
[1676275052.503032][11810:11815] CHIP:DMG:                      {
[1676275052.503064][11810:11815] CHIP:DMG:                              DataVersion = 0xfc7a0d27,
[1676275052.503094][11810:11815] CHIP:DMG:                              AttributePathIB =
[1676275052.503126][11810:11815] CHIP:DMG:                              {
[1676275052.503158][11810:11815] CHIP:DMG:                                      Endpoint = 0x0,
[1676275052.503191][11810:11815] CHIP:DMG:                                      Cluster = 0x28,
[1676275052.503225][11810:11815] CHIP:DMG:                                      Attribute = 0x0000_0003,
[1676275052.503256][11810:11815] CHIP:DMG:                              }
[1676275052.503288][11810:11815] CHIP:DMG:
[1676275052.503321][11810:11815] CHIP:DMG:                              Data = "TEST_PRODUCT" (12 chars),
[1676275052.503351][11810:11815] CHIP:DMG:                      },
[1676275052.503382][11810:11815] CHIP:DMG:
[1676275052.503406][11810:11815] CHIP:DMG:              },
[1676275052.503435][11810:11815] CHIP:DMG:
[1676275052.503454][11810:11815] CHIP:DMG:      ],
[1676275052.503479][11810:11815] CHIP:DMG:
[1676275052.503499][11810:11815] CHIP:DMG:      SuppressResponse = true,
[1676275052.503519][11810:11815] CHIP:DMG:      InteractionModelRevision = 1
[1676275052.503538][11810:11815] CHIP:DMG: }
[1676275052.503657][11810:11815] CHIP:TOO: Endpoint: 0 Cluster: 0x0000_0028 Attribute 0x0000_0003 DataVersion: 4235857191
[1676275052.503689][11810:11815] CHIP:TOO:   ProductName: TEST_PRODUCT
```

获取 endpoint 3 上 产品的标签，可以看到 是 lighting 设备

./chip-tool bridgeddevicebasic read node-label 1245 3

而用 chip-tool 查询 endpoint 4 上的设备, 结果为 TempSensor

./chip-tool bridgeddevicebasic read node-label 1245 4

```
[1676279453.981116][14028:14039] CHIP:DMG: ReportDataMessage =
[1676279453.981138][14028:14039] CHIP:DMG: {
[1676279453.981156][14028:14039] CHIP:DMG:      AttributeReportIBs =
[1676279453.981181][14028:14039] CHIP:DMG:      [
[1676279453.981201][14028:14039] CHIP:DMG:              AttributeReportIB =
[1676279453.981228][14028:14039] CHIP:DMG:              {
[1676279453.981249][14028:14039] CHIP:DMG:                      AttributeDataIB =
[1676279453.981283][14028:14039] CHIP:DMG:                      {
[1676279453.981309][14028:14039] CHIP:DMG:                              DataVersion = 0x5e552bb3,
[1676279453.981339][14028:14039] CHIP:DMG:                              AttributePathIB =
[1676279453.981371][14028:14039] CHIP:DMG:                              {
[1676279453.981403][14028:14039] CHIP:DMG:                                      Endpoint = 0x4,
[1676279453.981436][14028:14039] CHIP:DMG:                                      Cluster = 0x39,
[1676279453.981469][14028:14039] CHIP:DMG:                                      Attribute = 0x0000_0005,
[1676279453.981499][14028:14039] CHIP:DMG:                              }
[1676279453.981532][14028:14039] CHIP:DMG:
[1676279453.981567][14028:14039] CHIP:DMG:                              Data = "TempSensor 1" (12 chars),
[1676279453.981634][14028:14039] CHIP:DMG:                      },
[1676279453.981667][14028:14039] CHIP:DMG:
[1676279453.981689][14028:14039] CHIP:DMG:              },
[1676279453.981714][14028:14039] CHIP:DMG:
[1676279453.981733][14028:14039] CHIP:DMG:      ],
[1676279453.981758][14028:14039] CHIP:DMG:
[1676279453.981779][14028:14039] CHIP:DMG:      SuppressResponse = true,
[1676279453.981800][14028:14039] CHIP:DMG:      InteractionModelRevision = 1
[1676279453.981837][14028:14039] CHIP:DMG: }
[1676279453.981962][14028:14039] CHIP:TOO: Endpoint: 4 Cluster: 0x0000_0039 Attribute 0x0000_0005 DataVersion: 1582640051
[1676279453.982005][14028:14039] CHIP:TOO:   NodeLabel: TempSensor 1
```

控制 endpoint 3 上的 灯

./chip-tool onoff toggle 1245 3

```
[1676279634.123699][15311:15311] CHIP:DMG: InvokeRequestMessage =
[1676279634.123701][15311:15311] CHIP:DMG: {
[1676279634.123703][15311:15311] CHIP:DMG:      suppressResponse = false,
[1676279634.123705][15311:15311] CHIP:DMG:      timedRequest = false,
[1676279634.123707][15311:15311] CHIP:DMG:      InvokeRequests =
[1676279634.123710][15311:15311] CHIP:DMG:      [
[1676279634.123712][15311:15311] CHIP:DMG:              CommandDataIB =
[1676279634.123714][15311:15311] CHIP:DMG:              {
[1676279634.123716][15311:15311] CHIP:DMG:                      CommandPathIB =
[1676279634.123718][15311:15311] CHIP:DMG:                      {
[1676279634.123721][15311:15311] CHIP:DMG:                              EndpointId = 0x3,
[1676279634.123727][15311:15311] CHIP:DMG:                              ClusterId = 0x6,
[1676279634.123730][15311:15311] CHIP:DMG:                              CommandId = 0x2,
[1676279634.123732][15311:15311] CHIP:DMG:                      },
[1676279634.123734][15311:15311] CHIP:DMG:
[1676279634.123736][15311:15311] CHIP:DMG:                      CommandFields =
[1676279634.123739][15311:15311] CHIP:DMG:                      {
[1676279634.123743][15311:15311] CHIP:DMG:                      },
[1676279634.123745][15311:15311] CHIP:DMG:              },
[1676279634.123748][15311:15311] CHIP:DMG:
[1676279634.123750][15311:15311] CHIP:DMG:      ],
[1676279634.123753][15311:15311] CHIP:DMG:
[1676279634.123754][15311:15311] CHIP:DMG:      InteractionModelRevision = 1
[1676279634.123756][15311:15311] CHIP:DMG: },
[1676279634.123767][15311:15311] CHIP:DMG: AccessControl: checking f=1 a=c s=0x000000000001B669 t= c=0x0000_0006 e=3 p=o
[1676279634.123775][15311:15311] CHIP:DMG: AccessControl: allowed
[1676279634.123778][15311:15311] CHIP:DMG: Received command for Endpoint=3 Cluster=0x0000_0006 Command=0x0000_0002
[1676279634.123782][15311:15311] CHIP:ZCL: On/Off set value: 3 2
[1676279634.123784][15311:15311] CHIP:DL: HandleReadOnOffAttribute: attrId=0, maxReadLength=1
[1676279634.123786][15311:15311] CHIP:ZCL: Toggle on/off from 0 to 1
[1676279634.123789][15311:15311] CHIP:DL: HandleWriteOnOffAttribute: attrId=0
[1676279634.123793][15311:15311] CHIP:DL: Device[Light 1]: ON
[1676279634.123797][15311:15311] CHIP:DMG: Endpoint 3, Cluster 0x0000_0006 update version to 637006a3
```