

Matter 开发：基于 Linux 的应用程序

本文档简单介绍了 Matter 产品开发需要用到的 Linux 环境，以及如何在 Linux 环境中编译出开发 Matter 设备过程中需要用到的一些应用程序。若使用芯科科技提供的 Ubuntu 镜像，已集成了开发环境，并且编译好本文中所提及的应用程序，开发者直接使用即可。

主要内容

- Linux 环境
- 应用程序: chip-tool
- 应用程序: OTBR(Open Thread Border Router)
- 应用程序: fw_upgrade_utility
- 应用程序: OTA Provider

Table of Contents

- 1 **Linux 环境**..... 2
- 2 **应用程序: chip-tool**..... 2
 - 2.1 编译应用程序2
- 3 **应用程序: OTBR(OpenThread Border Router)** 2
 - 3.1 编译应用程序2
- 4 **应用程序: fw_upgrade_utility**..... 3
 - 4.1 编译应用程序3
 - 4.2 升级 USB Stick 的固件 (OT-RCP)3
- 5 **应用程序: OTA Provider** 3
 - 5.1 编译 OTA Provider 应用程序3
- 6 **参考资料**..... 4
- 7 **文档修订历史**..... 4
 - Revision 1.0.04

1 Linux 环境

- Linux 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
```

- 从 silicon labs 官方 github 获取 Matter SDK

```
$ 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，可能达数小时之久）
$ ./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 小时左右或更长时间）
$ source scripts/activate.sh
```

2 应用程序：chip-tool

2.1 编译应用程序

前提条件：需要事先完成 Matter SDK 环境配置。

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

编译完成后，可在`out/debug/standalone/`目录下看到编译好的`chip-tool`工具。

3 应用程序：OTBR(OpenThread Border Router)

3.1 编译应用程序

1. 下载`GSDK`

```
$ git clone https://github.com/SiliconLabs/gecko_sdk.git
```

2. 创建一个符号链接指向`Openthread`协议栈

```
$ cd gecko_sdk
$ cd util/third_party/ot-br-posix
$ cd third_party/openthread
# 创建一个软链接 repo
$ ln -s ../../../../openthread repo
```

3. 运行引导脚本`bootstrap`安装依赖

```
$ cd gecko_sdk
$ cd util/third_party/ot-br-posix
# 添加可执行权限
$ chmod +x script/*
$ sudo ./script/bootstrap
```

4. 编译`otbr-agent`应用

```
$ cd gecko_sdk
$ cd util/third_party/ot-br-posix
# en0p3 为网络接口名
$ sudo INFRA_NAME=en0p3 ./script/setup
```

注：这里的`en0p3`需要修改成实际使用的网络端口。

4 应用程序：fw_upgrade_utility

4.1 编译应用程序

从芯科科技获取源码包：`usbstick_fw_upgrade_util_v0.2.tar.bz2`

1. 解压应用源码到文件夹`fw_upgrade_util`

```
$ tar -jxvf usbstick_fw_upgrade_util_v0.2.tar.bz2
```

2. 编译应用`fw_upgrade_utility`

```
$ cd fw_upgrade_util
$ rm -rf build
$ mkdir build
$ cd build/
# 使用 CMake 进行编译
$ cmake ..
$ make
```

编译完成后，`build/`目录下将生成可独立运行的程序`fw_upgrade_utility`。

4.2 升级 USB Stick 的固件（OT-RCP）

1. 查看 USB Stick 的端口

```
$ ls /dev/ttyUSB*
ttyUSB0
```

2. 停止`otbr-agent`以释放`tty`端口

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

3. 升级固件到 USB Stick 中

```
$ sudo ./fw_upgrade_utility -f ~/ot-rcp-mg21-usbstick.gbl -p /dev/ttyUSB0
```

4. 重新启动 otbr-agent

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

5 应用程序：OTA Provider

5.1 编译 OTA Provider 应用程序

```
$ cd matter
$ ./scripts/examples/gn_build_example.sh ./examples/ota-provider-app/linux ./out/debug chip_config_net-
work_layer_ble=false
```

编译完成后，生成路径为：`./out/debug/chip-ota-provider-app`本次培训硬件清单：

6 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/>

7 文档修订历史

Revision 1.0.0

Jan 31, 2023

- 初始版本

- 文档结束 -