Matter over Thread
User Development Manual

Table of Contents

1 Overview

This document introduces how to develop Matter over Thread examples based on the HOPERF HM-MT2401 EVB Kit.

The HM-MT2401 EVB board integrates the HOPERF HM-MT2401 module and the rich of sensor devices, including temperature and humidity sensors, pressure sensors, contact sensors, PIR sensors, an RGB light and two single-color LED lights, as well as four buttons and a 1 MB external SPI Flash. Using the HM-MT2401 EVB can quickly develop Matter over Thread end devices, evaluate the functions of Matter devices, and quickly build Matter devices for functional demonstrations, etc. For hardware information about the HM-MT2401 EVB, please refer to the <HM-MT2401 EVB User Manual.pdf> document.

This document focuses on the Matter development platform of Silicon Labs, including building a Matter development environment, compiling the Bootloader and Matter examples, flashing Bootloader and Matter firmware, viewing Matter device logs, and using chip-tool to commission a Matter device to the Matter network, adding a Matter device to Apple Home ecosystem, etc.

2 Hardware Preparation

The Matter development environment need to prepare the following hardware equipment:

- One computer (Windows 10/11 system installed), the computer configuration should be as high as possible (I7 processor, 16G memory or above)
- One J-Link Programmer or one Silicon Labs WSTK(BRD4001/BRD4002)
- Two HM-MT2401 EVB development boards
- One Home Pod Mini
- One Apple mobile phone or IPAD
- One Raspberry Pi

3 Software Preparation

3.1 Install Simplicity Studio V5

HM-MT2401 EVB uses Silicon Labs' EFR32MG24 MCU and the development environment is SSV5(Simplicity Studio V5). It can be downloaded from here.

Download SSV5 from the Silicon Labs website and the installation package is an iso file. Double click it, then double-click the inside "setup.exe" file to install, as shown in figure 3.1, Click on “I accept the terms of the agreement”, then click on "Next".

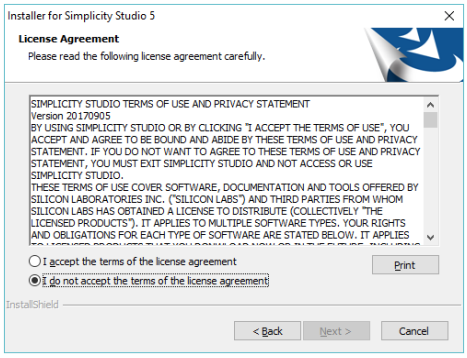

Figure 3.1

Then select the installation location, click "Next", and then click "Install". When SSV5 starts, you will see one or more additional license agreements. You can accept them individually or together. Click "DONE" when finished.

Next, log in with your Silicon Labs account. Note: You can skip login here, but some packages with restricted access must be logged in first.

For more details, please refer to the Simplicity Studio® 5 User's Guide.

3.2 Install Silicon Labs Gecko SDK

Open SSV5(Simplicity Studio V5) and click "Install" to install the SDK related packages. As shown in figure 3.2.

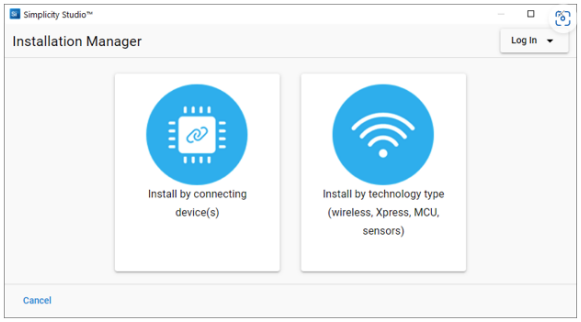

Figure 3.2

The detailed steps are as follows:

1) Click "Install by Technology Type"

2) Select "32-Bit and Wireless MCUs" and click "Next", as shown in figure 3.3

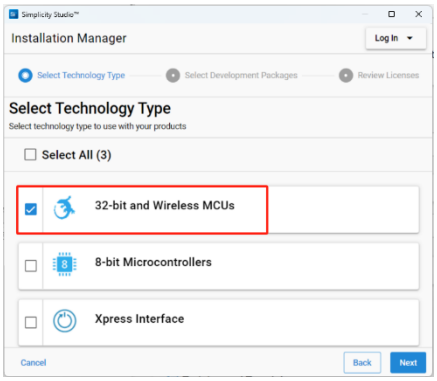

Figure 3.3

3) Select "Gecko SDK - 32-bit and Wireless MCUs-4.3.2"

4) Check "Silicon Labs Matter-2.1.1" and click "Next", start to download and install SDK.

As shown in figure 3.4 and figure 3.5:

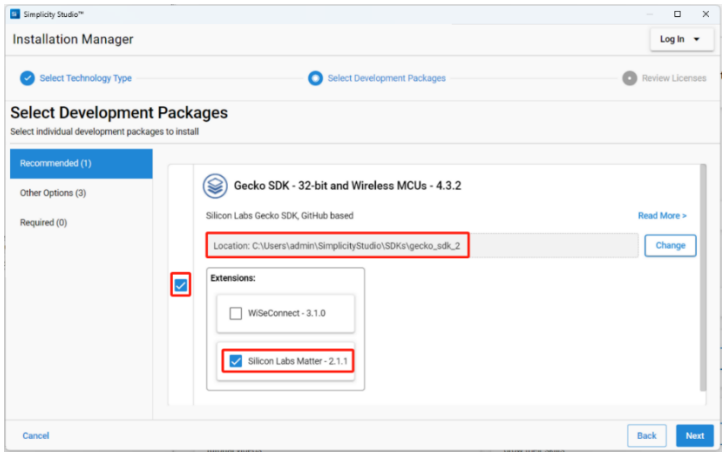

Figure 3.4

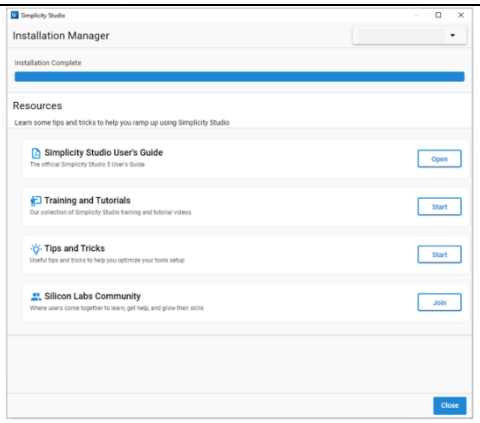

Figure 3.5

5) It may take some time to complete the download according to different network environments, as shown in figure 3.6

6) When the installation is complete and click "OK" to restart SSV5, as shown in figure 3.7

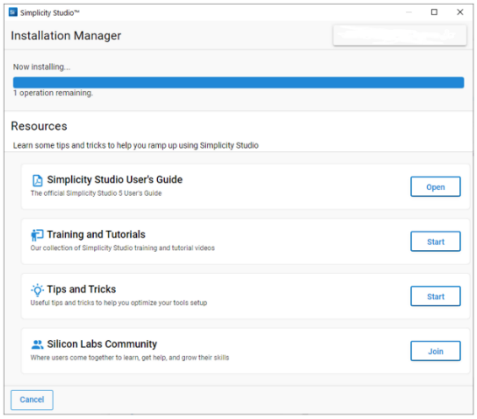

Figure 3.6

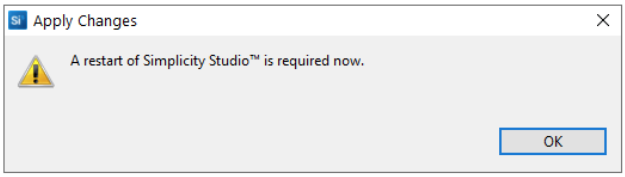

Figure 3.7

Note: Installing the SDK may fail in China, please try again at another time. The reason is that Silicon Labs' GSDK is deployed on Github, and accessing Github may be intermittently disconnected in China.

4 Compile the bootloader example

The default Matter example enable the Bootloader function, and the Matter firmware can be upgraded through OTA. The following describes how to compile the Bootloader firmware suitable for the HM-MT2401 EVB development board.

4.1 Create the Bootloader project

Click "File->New->Silicon Labs Project Wizard..." to open the project creation window, then do the following process, as shown in figure 4.1:

1) Delete the content of "Target Boards"
2) Input "EFR32MG24A420F1536IM40" for "Target Device"
3) Select "Gecko SDK Suit v4.3.2…" for "SDK"
4) Select "Simplicity IDE/GUI ARM v10.3.1" for "IDE/Toolchain".
5) Click "Next" finally.

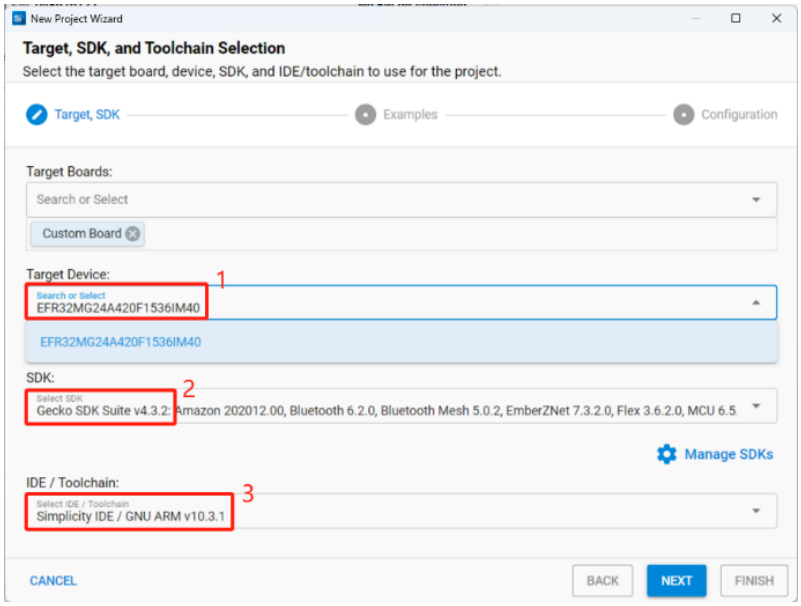

Figure 4.1

Do the following process in the pop-up "Example Project Selection" window, as shown in figure 4.2.

1) Check "Bootloader" in the "MCU" options to filter irrelevant configuration items

2) Select the "Bootloader - SoC Internal Storage (single image on 1536 kB device)" project

3) Click "Next"

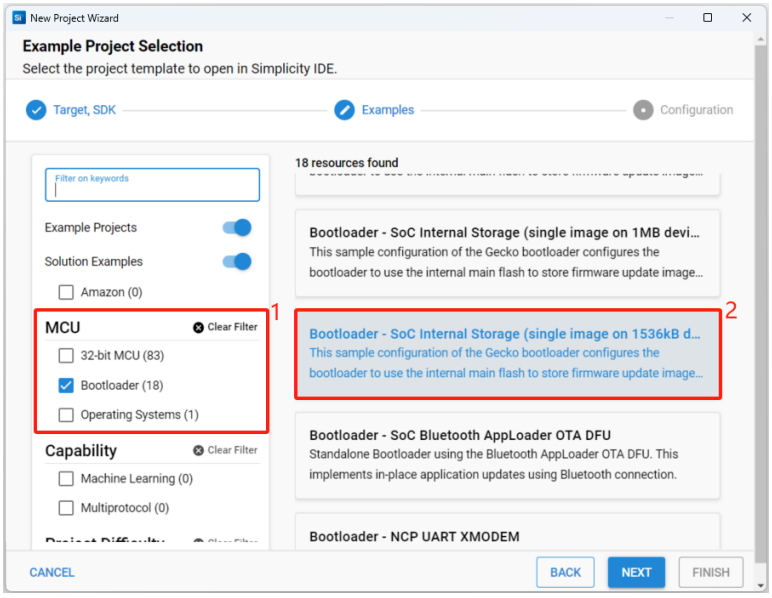

Figure 4.2

Click "FINISH" in the pop-up "Project Configuration" window to complete the creation of the Bootloader project, as shown in figure 4.3.

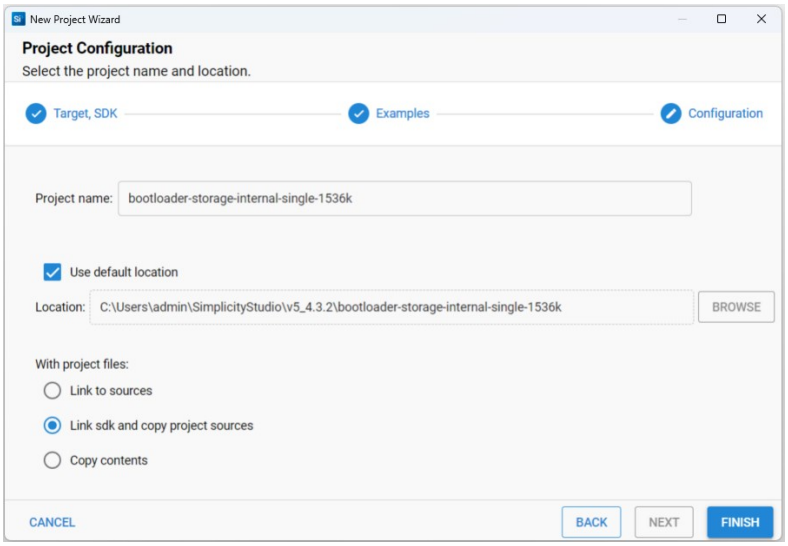

Figure 4.3

After the project is successfully created, the project will be displayed in "Project Explorer", as shown in figure 4.4.

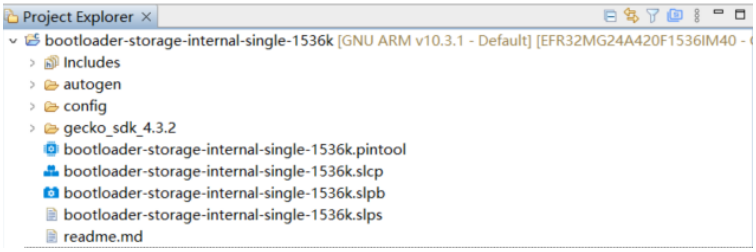

Figure 4.4

4.2 Modify the Bootloader project

The Flash size of the HM-MT2401 module is 1536KB and the distribution of each data segment is shown in Table 4.1.

Table 4.1 Flash Memory Mapping

Name	Initial address	End address	Size
Bootloader	0x08000000	0x08005FFF	24KB
Application	0x08006000	0x080EBFFF	920KB
Slot Storage (OTA)	0x080EC000	0x08171FFF	536KB
NVM3	0x08174000	0x0817DFFF	40KB
Matter Certs	0x0817E000	0x0817FFFF	8KB

Note: In order to store the Matter OTA firmware in the internal flash of MG24, the Bootloader needs to enable the "LZMA COMPRESSION" function, the OTA firmware uses "LZMA" compression, and need to change the OTA file storage address and range.

Double click the "bootloader-storage-internal-single-1536k.slcp" of the project to open to open configuration window, and modify the following 3 parameters:

1. Modify "Bootloader Storage Slot Setup"

Modify the parameters of the component "SOFTWARE COMPONENTS->Platform->Bootloader->Storage->Bootloader Storage Slot Setup", as shown in figure 4.5.

1) The "Start Address" of "Enable Slot0" is set to 0x080EC000

2) The "Slot Size" of "Enable Slot0" is set to 0x86000

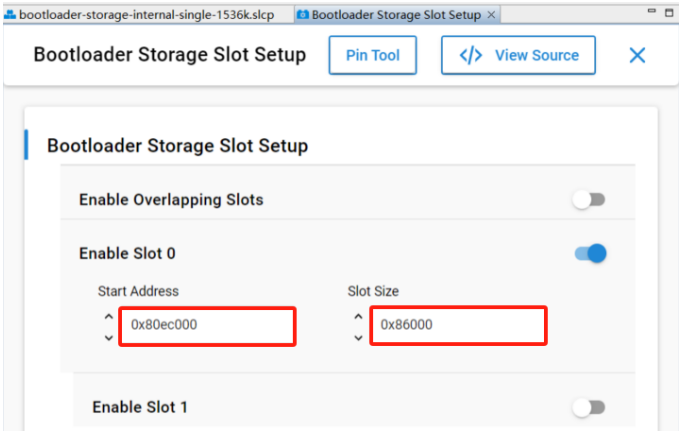

Figure 4.5

2. Modify "Common Storage"

Modify the parameters of the component "SOFTWARE COMPONENTS->Platform->Bootloader->Storage->Common Storage", as shown in figure 4.6.

1) The "Start Address of boot load info" is set to 0x080EC000, it must be the same as the "Start Address" of "Enable Slot0"

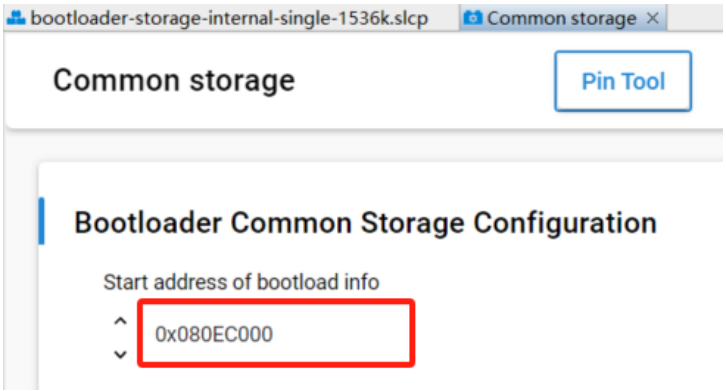

Figure 4.6

3. Add "LZMA" component

Install the GBL Compression (LZMA) component, as shown in figure 4.7.

1) Click "Install" in "SOFTWARE COMPONENTS->Platform->Bootloader->Core->GBL Compression (LZMA)" to install

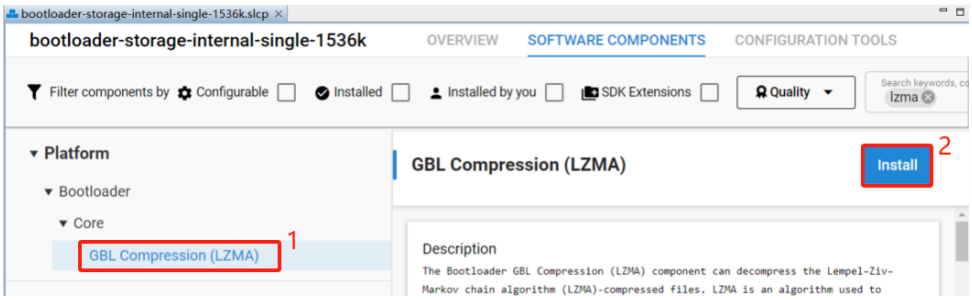

Figure 4.7

4.3 Compile the Bootloader project

In "Project Explorer", select the "bootloader-storage-internal-single-1536k" project and click "Build" to start compiling the project, as shown in figure 4.8.

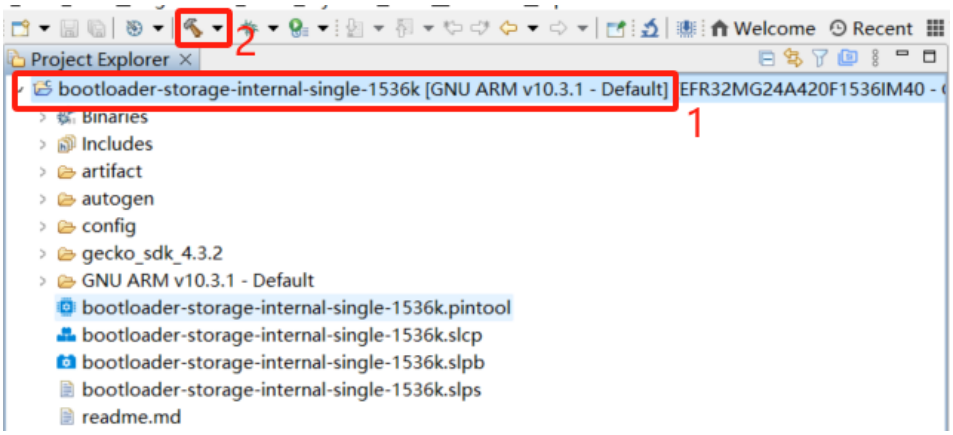

Figure 4.8

After the compilation is completed, the bootloader firmware is generated under the directory (GNU ARM v10.3.1-Default), as shown in figure 4.9.

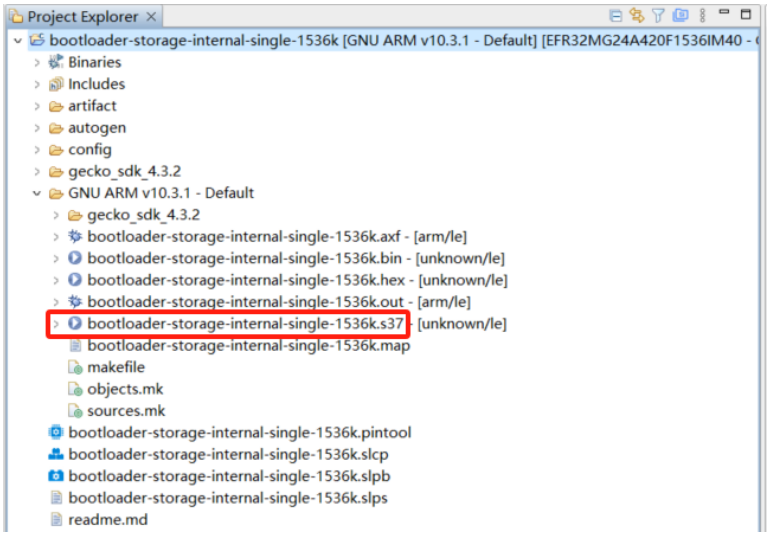

Figure 4.9

4.4 Flashing Bootloader Firmware

Please refer to Chapter 7 of "Flashing Matter Firmware" to burn the compiled Bootloader firmware "bootloader-storage-internal-single-1536k.s37" to the chip.

5 Compile the Matter example

This document uses "Lighting-app Matter" as an example, it is based on the HM-MT2401 EVB development board.

5.1 Create Matter project

In "SSV5", click "File->New->Silicon Labs Project Wizard..." to open the "New Project Wizard" window, as shown in figure 5.1.

1) Delete the content of "Target Boards"

2) Input "EFR32MG24A420F1536IM40" for "Target Device"

3) Select "Gecko SDK Suit v4.3.2…" for "SDK"

4) Select "Simplicity IDE/GUI ARM v10.3.1" for "IDE/Toolchain"

5) Click "Next" finally

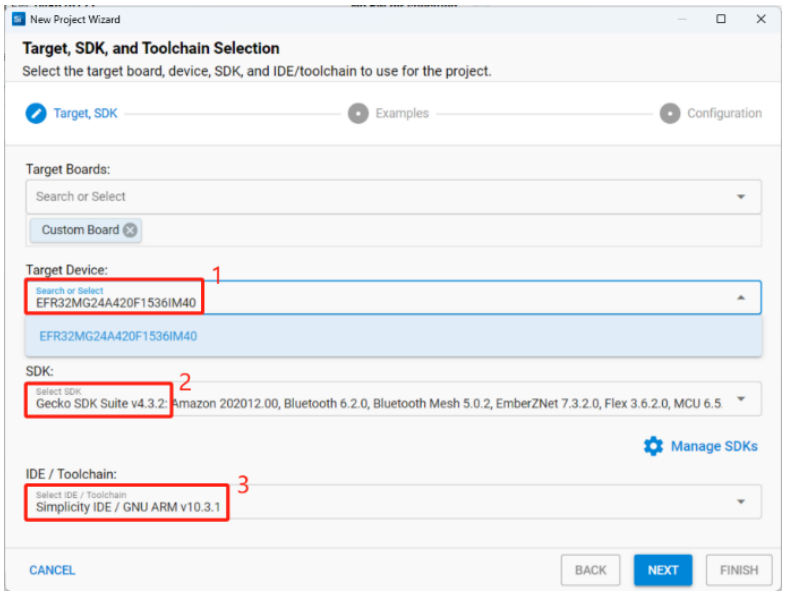

Figure 5.1

Then select the corresponding project, as shown in figure 5.2.

1) Check "Matter" in the "Wireless Technology" options to filter irrelevant configuration items

2) Select the "Wireless Technology" project

3) Click "Next"

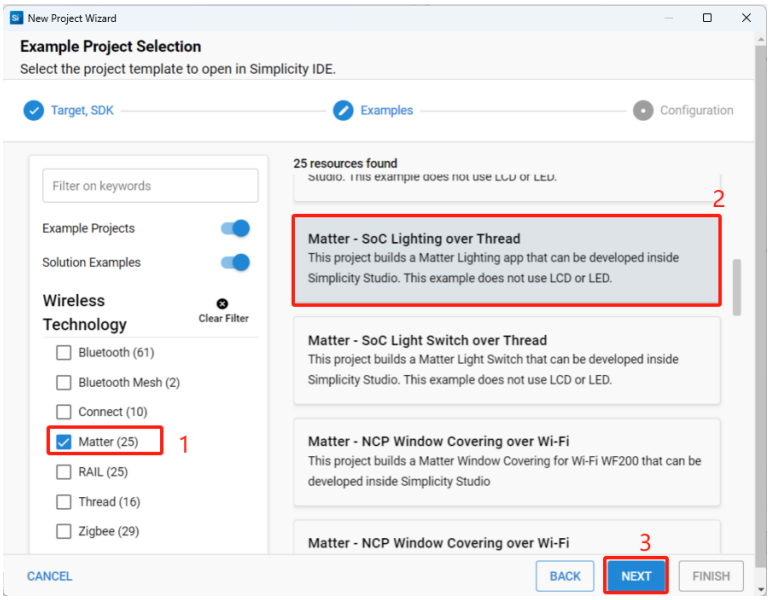

Figure 5.2

Change the project name to "Matter_lighting_mt2401_evb", use the default project directory, and click "FINISH" to complete the project creation, as shown in Figure 5.3.

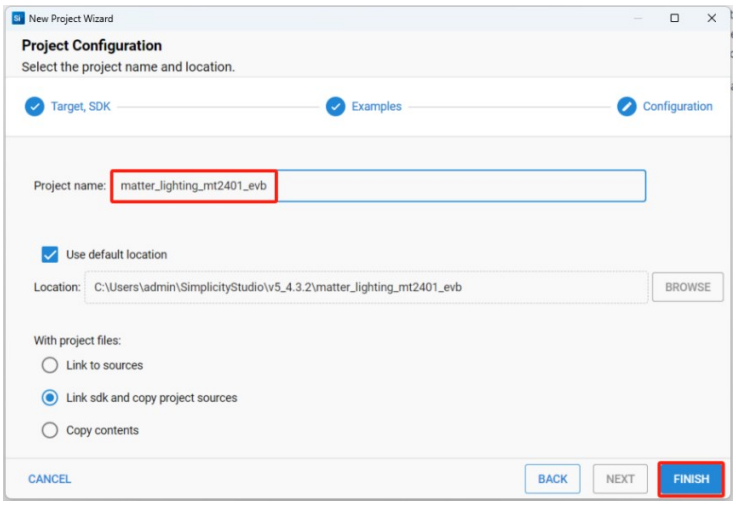

Figure 5.3

After the project is successfully created, the project will be displayed in "Project Explorer", as shown in figure 5.4.

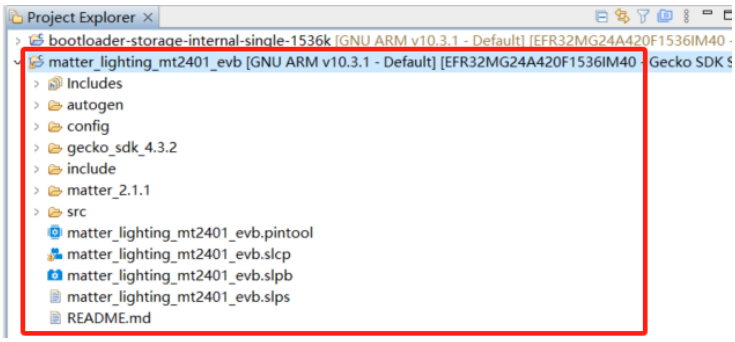

Figure 5.4

5.2 Modify the Matter project

Double click "Matter_lighting_mt2401_evb.slcp" of the project to open the configuration window, as shown in figure 5.5.

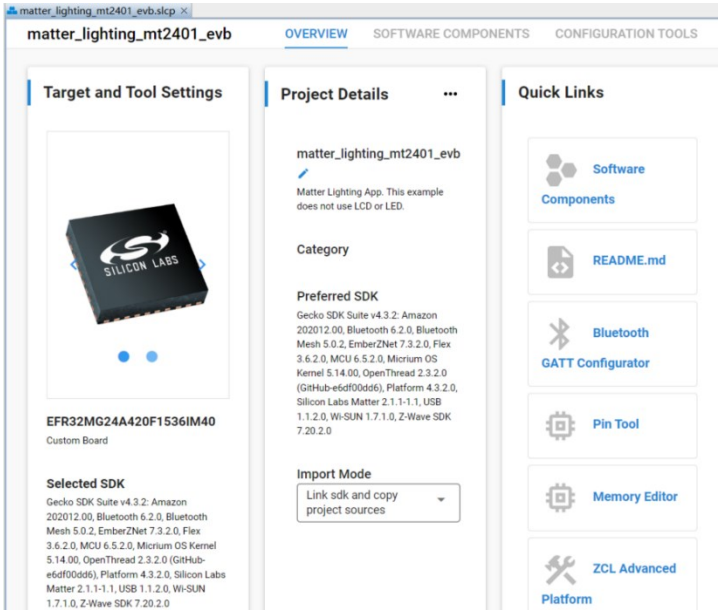

Figure 5.5

Click "SOFTWARE COMPONENTS" and configure as follows:

1. Modify LED configuration

1) Delete the "Silicon Labs Matter -> Platform -> WSTK LED Stub" component, as shown in figure 5.6

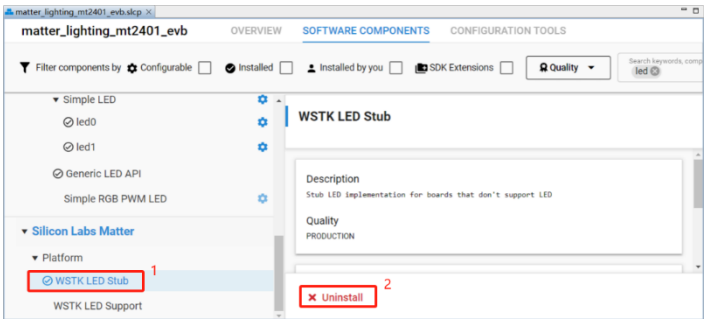

Figure 5.6

2) Install the "Silicon Labs Matter -> Platform -> WSTK LED Support" component, as shown in figure 5.7

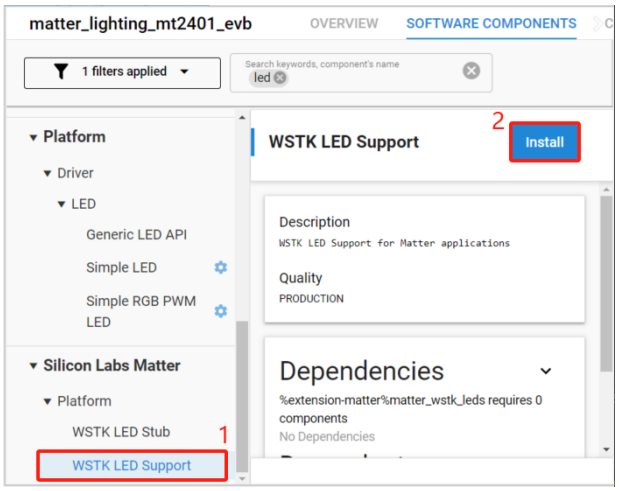

Figure 5.7

3) Install the "Platform -> Driver -> LED -> Simple LED" component, as shown in figure 5.8

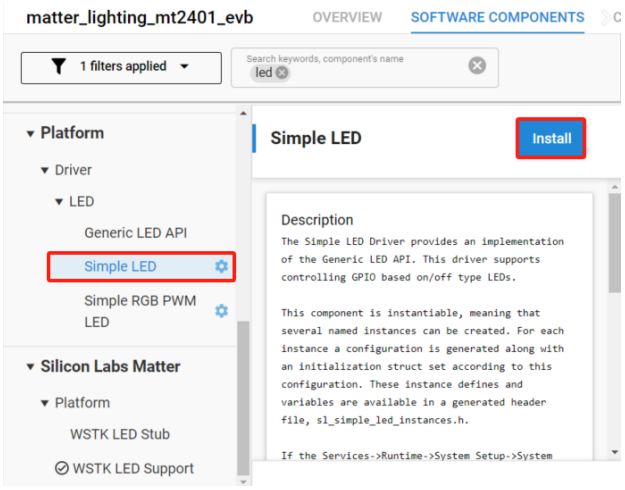

Figure 5.8

4) Change the name to led0 in the pop-up window, as shown in figure 5.9

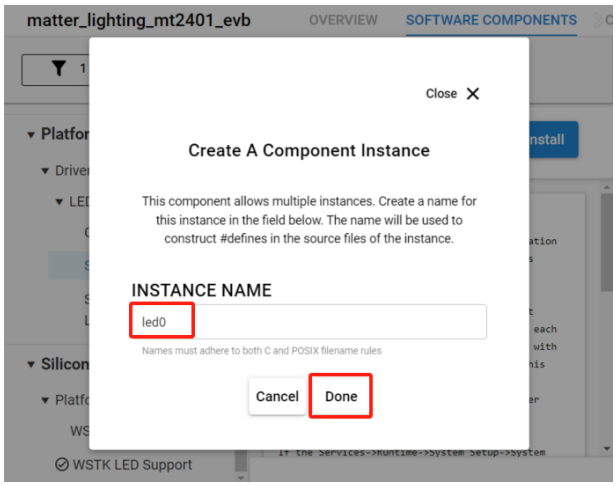

Figure 5.9

5) Repeat the above steps and add another LED (led1), as shown in figure 5.10

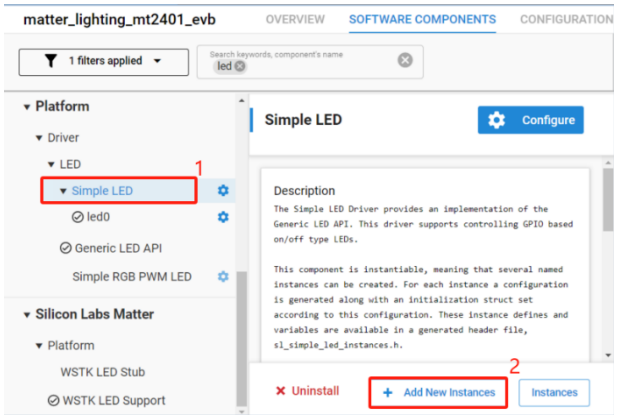

Figure 5.10

Note: Because the LED's names used in the default project are led0 and led1, the names of the newly added LEDs must be led0 and led1, too.

1) Click "Configure" to configure LED, as shown in figure 5.11 and figure 5.12.

a) Configure led0 as "Active high" and "PA08".

b) Configure led1 as "Active high" and "PD03".

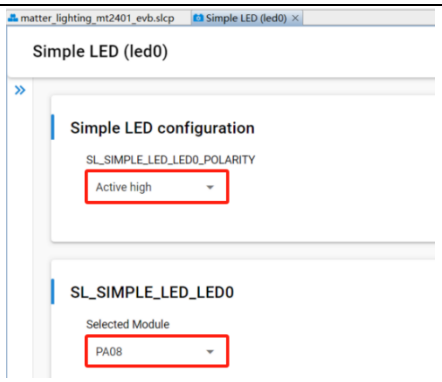

Figure 5.11

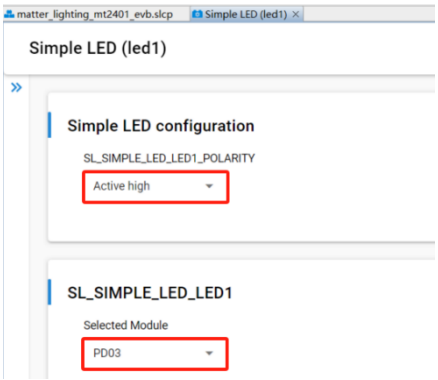

Figure 5.12

2. Modify Button configuration

1) In "Platform -> Driver -> Button -> Simple Button" component, click btn0 or btn1 to open the button configuration window, as shown in figure 5.13 and figure 5.14.

a) Configure btn0 as "Interrupt" and "PB01"

b) Configure btn1 as "Interrupt" and “PB02"

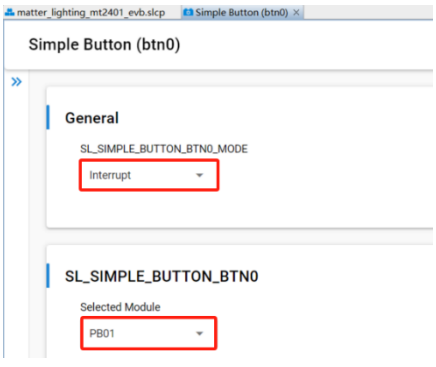

Figure 5.13

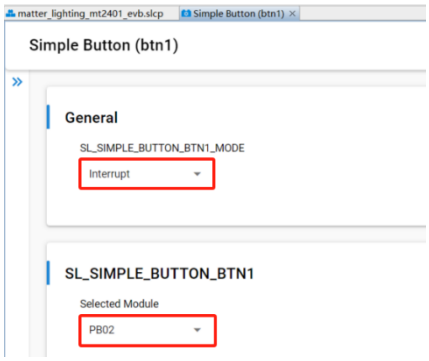

Figure 5.14

Note: The "btn0" corresponds to the "BTN1" of the EVB board, the "btn1" corresponds to the "BTN2" of the EVB board.

3. Modify serial port configuration

The Matter device can receive CLI commands through the serial port.

1) Open the "Platform -> Driver -> UART -> UARTDRV EUSART -> vcom" configuration window and configure the following parameters, as shown in figure 5.15.

a) Flow control method： None

b) CTS: None

c) RTS: None

d) RX: PA06

e) TX: PA05

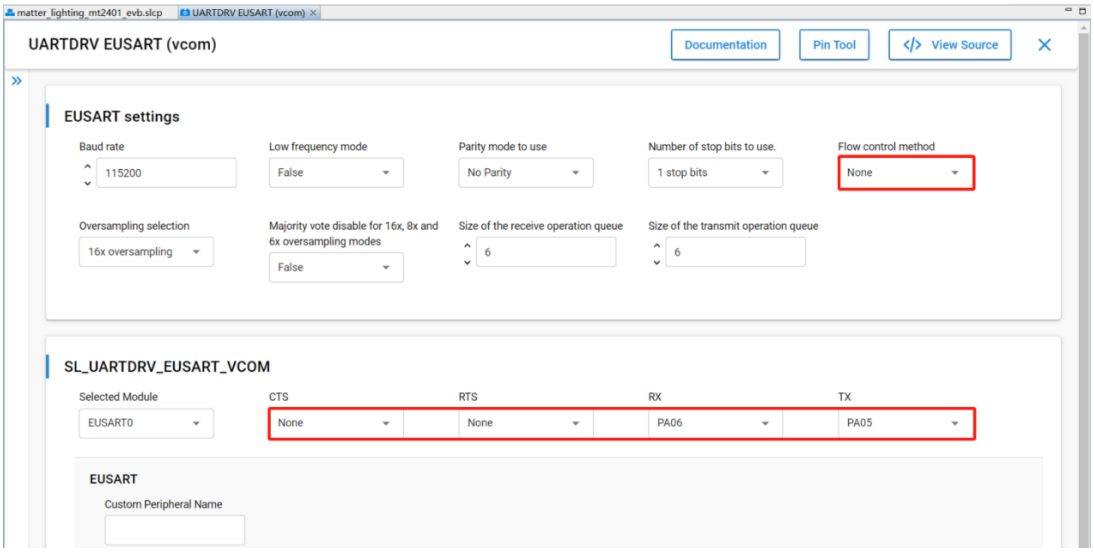

Figure 5.15

4. Enable the DAC certificate

1) Install the "Silicon Labs Matter -> Platform -> Security -> Matter Device Attestation Credentials" component, as shown in figure 5.16

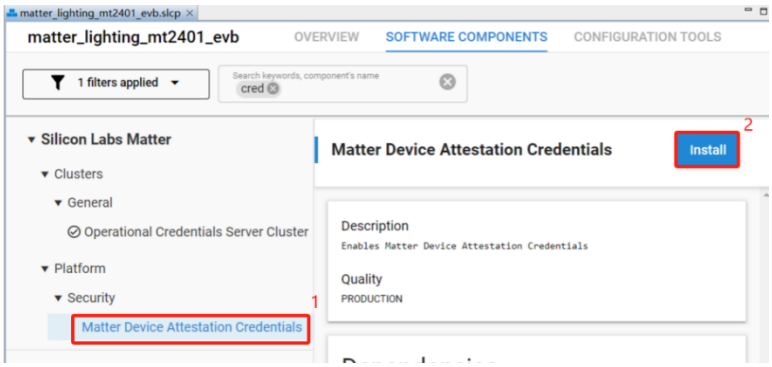

Figure 5.16

5.3 Compile the Matter project

Click "Build" to start compiling the project, after the compilation is completed the firmware is generated under the directory (GNU ARM v10.3.1 – Default), as shown in figure 5.17.

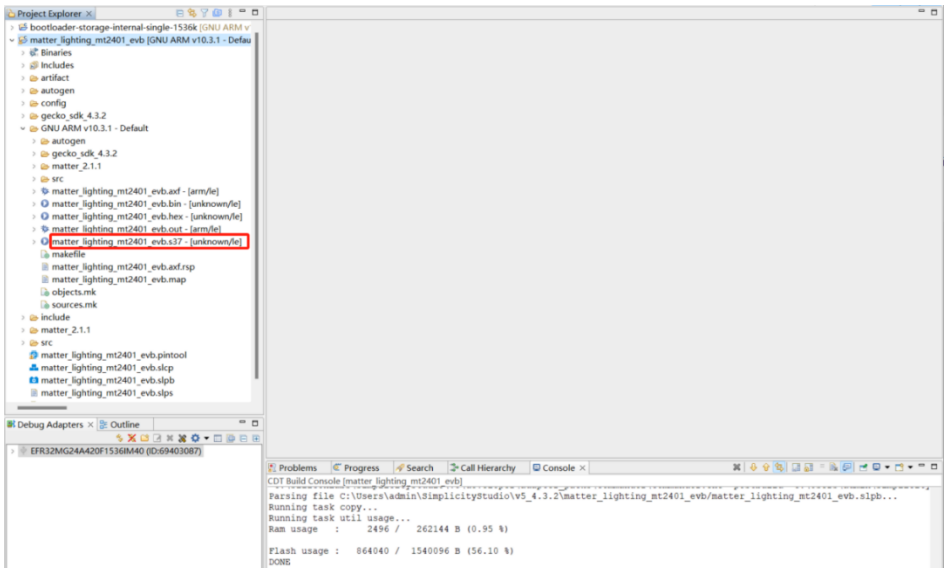

Figure 5.17

5.4 Flashing Matter firmware

Please refer to Chapter 7 of "Flashing Matter Firmware" to burn the compiled Matter firmware (Matter_lighting_mt2401_evb.s37) to the device.

6 Configure sensors

The HM-MT2401 EVB integrates TH09C (temperature-humidity sensor) and HP303B (air pressure sensor). This section takes these sensors as example, adds the corresponding driver code to the Matter Lighting example, uses ZAP tools to add endpoints and configures the sensor functions.

6.1 Configure I2C driver

TH09C (temperature-humidity sensor) and HP303B (pressure sensor) are both connected to the MCU by I2C, they are on the same I2C bus. Open the "SOFTWARE COMPONENTS" and install the I2CSPM component, as shown in figure 8.1.

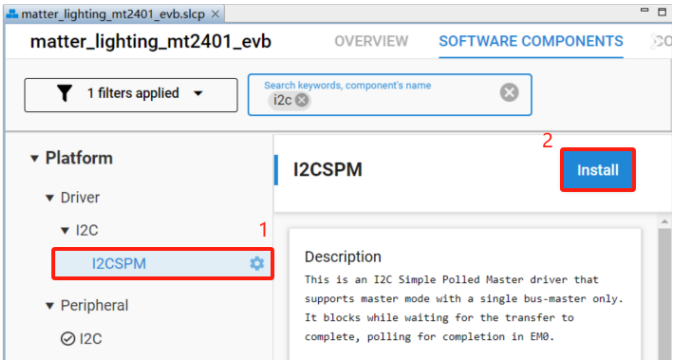

Figure 8.1

In the pop-up window, configure name as "sensor", as shown in figure 8.2.

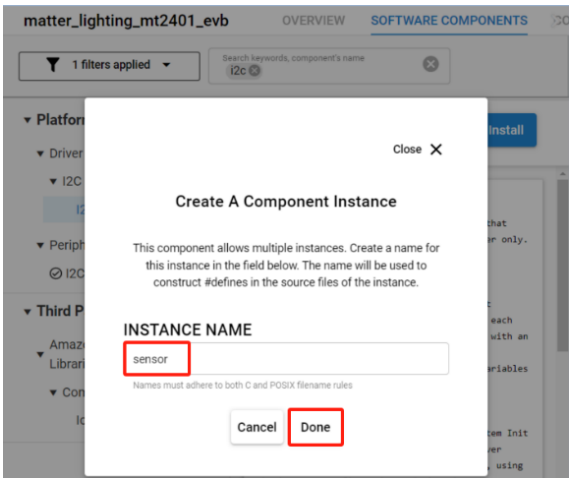

Figure 8.2

Then click "Configure" of the sensor to open the configuration window. Configure "Selected Module" as "I2C1", configure SCL's pin as "PC04", and configure SDA's pin as "PC05", as shown in figure 8.3.

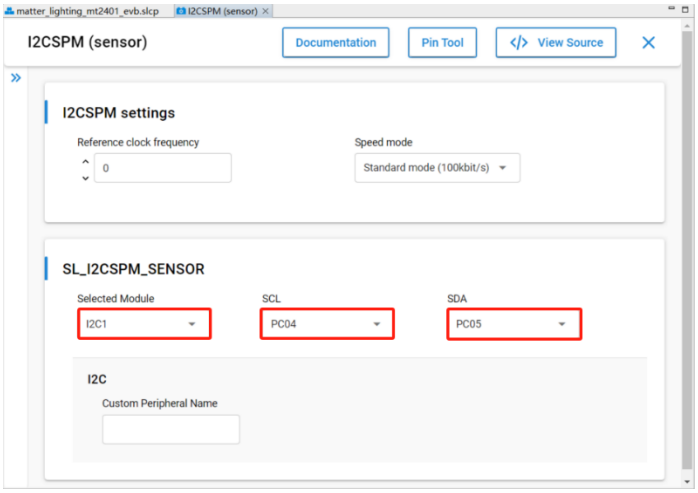

Figure 8.3

6.2 Add Endpoint

Use the ZAP tool to add and configure endpoints. Each of the temperature sensor, humidity sensor, and air pressure sensor occupy one endpoint, so that three endpoints need to be added.

1. Open ZAP tool

1) In the "CONFIGURATON TOOLS", click the "Open" of "Zigbee Cluster Configurator" to open the ZAP tool, as shown in figure 8.4

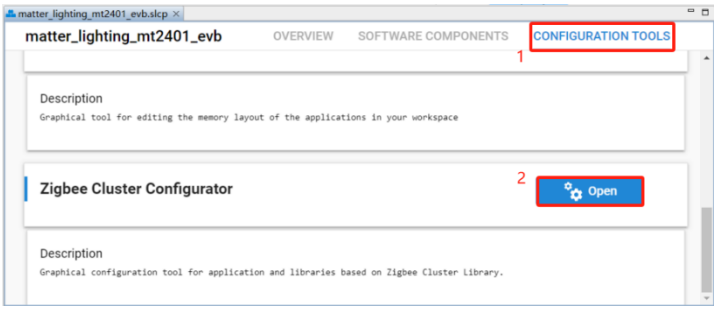

Figure 8.4

2) In the pop-up window, click "+ADD ENDPOINT" to add endpoints, as shown in figure 8.5

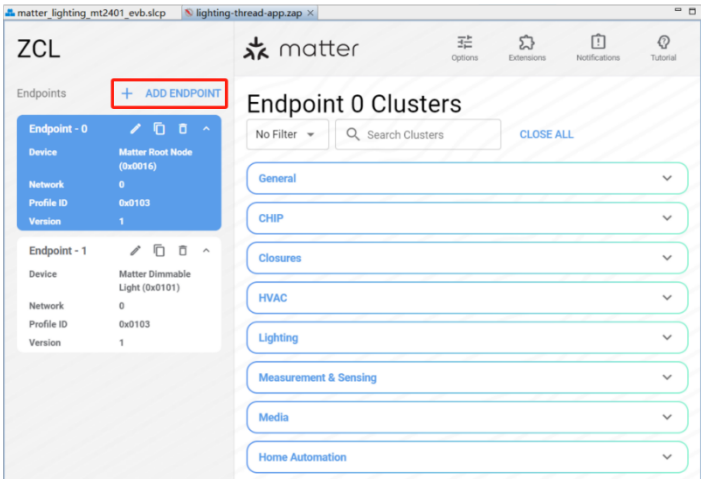

Figure 8.5

2. Add Endpoint for temperature sensor

Add and configure the following endpoint, as shown in figure 8.6.

- Configure endpoint index as 2
- Configure Profile ID as 0x0103
- Configure Device as "Matter Temperature Sensor(0x0302) "
- Configure "Primary Device" as "Matter Temperature Sensor(0x0302)"

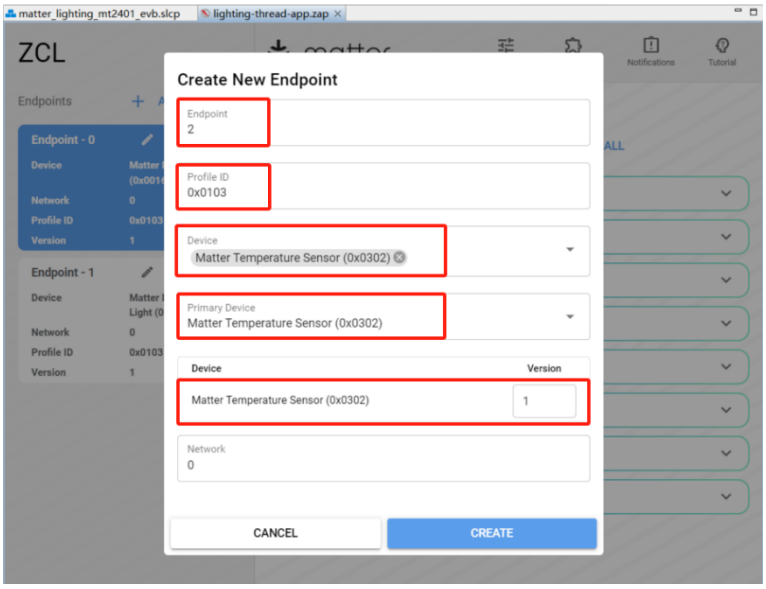

Figure 8.6

Follow the steps above, add and configure two endpoints for humidity sensor and air pressure sensor.

Add and configure endpoint for humidity sensor:

- Configure endpoint index as 3
- Configure Profile ID as 0x0103
- Configure Device as "Matter Humidity Sensor(0x0307)"
- Configure "Primary Device as "Matter Humidity Sensor(0x0307)"

Add and configure endpoint for air pressure sensor:

- Configure endpoint index as 4
- Configure Profile ID as 0x0103
- Configure Device as "Matter Pressure Sensor(0x0305)"
- Configure "Primary Device as "Matter Pressure Sensor(0x0305)"

After using the ZAP tool to complete the configuration, press "ctrl + s" to save the configuration, the SSV5 tool will automatically generate code.

6.3 Add driver and application codes for sensors

1. Add the codes to the project

Add the code files (drv_th09c.c, drv_hp303b.c, hal_sensor_rht.c, SensorManager.cpp) to the "src" directory of the project.

Add the header files (drv_th09c.h, drv_hp303b.h, hal_sensor_rht.h, SensorManager.h) to the "include" directory of the project.

To obtain the sensor driver source codes, please contact HOPERF's technical support team.

2. Modify the codes

Open the "src/AppTask.cpp" file and include "SensorManager.h" in the file, as shown in figure 8.7.

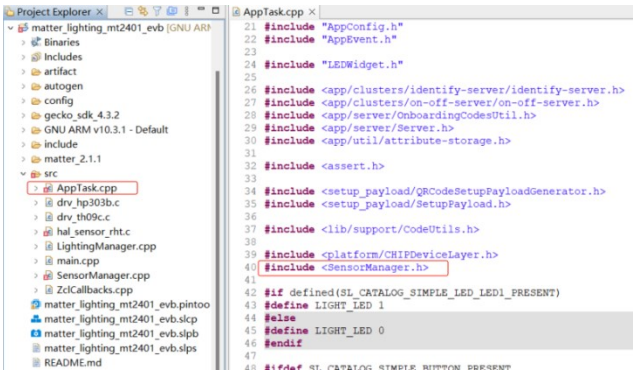

Figure 8.7

At the end of the "AppTask::Init()" function, add the sensor initialization code, as shown in figure 8.8.

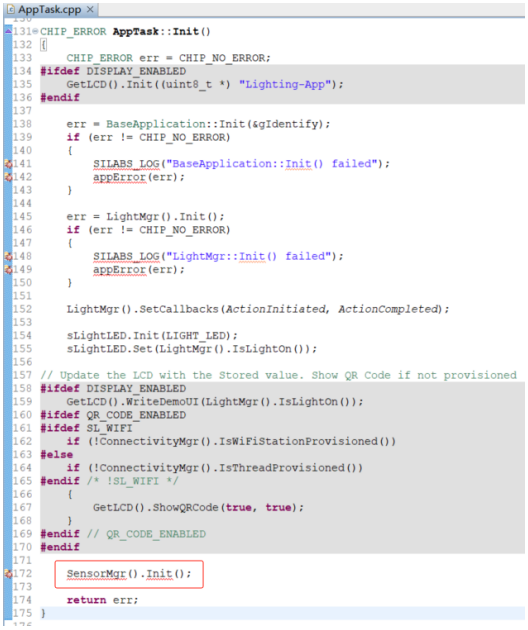

Figure 8.8

6.4 Recompile the Matter firmware

Click "Build" of the project to recompile the Matter example, after compilation is completed the firmware is in the directory, as shown in figure 8.9.

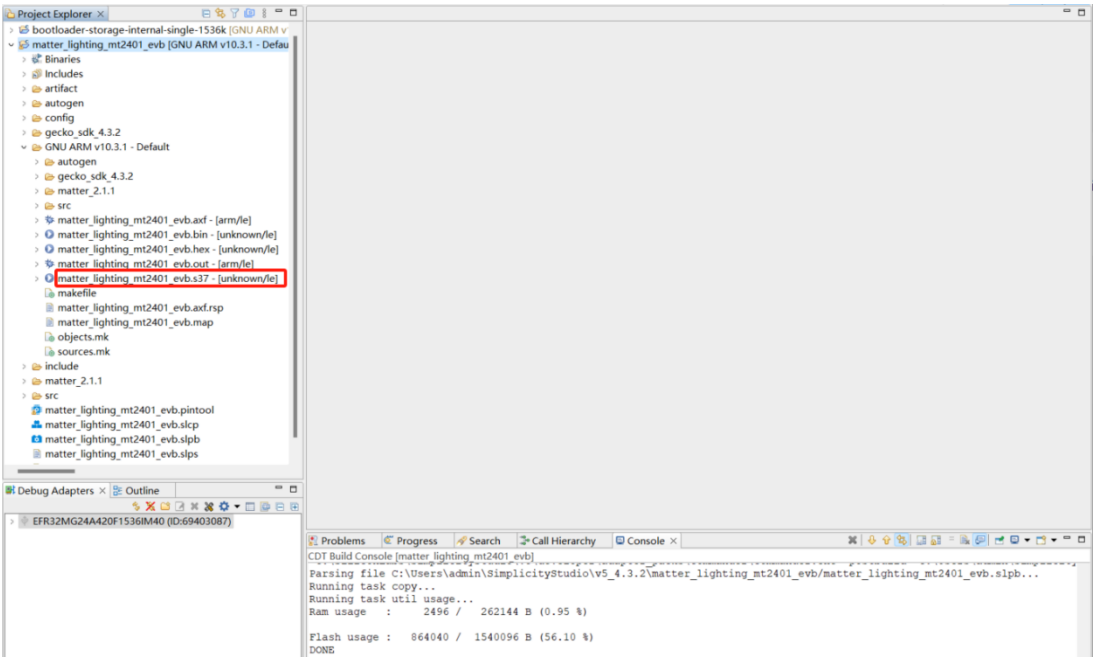

Figure 8.9

Note: HM2401 EVB also integrates occupancy sensor (device id is 0x0107) and contact sensor (device id is 0x0015). If need these sensors, need to add endpoints and import the corresponding driver codes to project.

7 Flashing Matter firmware

Use the JLink debugger to burn in firmware or debug online for HM-MT2401 EVB.

As shown in figure 7.1, use JLink to connect the SWD interface of HM-MT2401 EVB. There are two methods to burn Matter firmware, they are the SSV5 Integrated tool (Flash Programmer) or the Simplicity Commander.

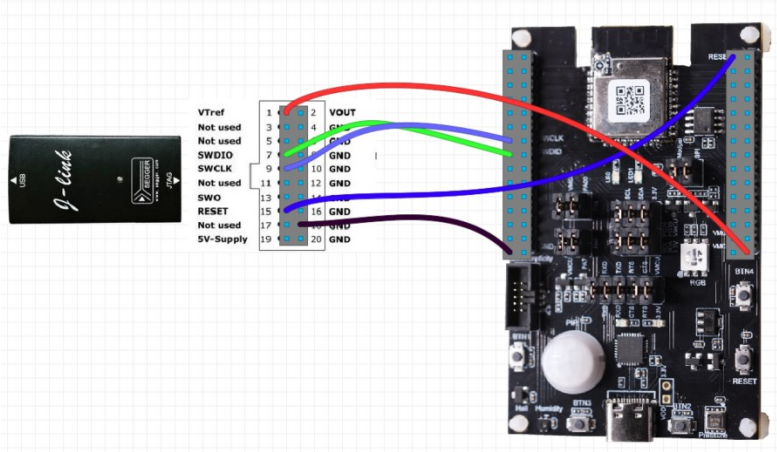

Figure 7.1

7.1 Use SSV5 Flash programmer to burn firmware

Open SSV5, displays the connected JLink device in "Debug Adapters", as shown in Figure 7.2.

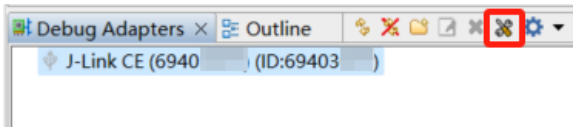

Figure 7.2

Select the corresponding JLink device, right click, select "Device Configuration", then select "Device hardware" in the pop-up window.

Input the chip model (EFR32MG24A420F1536IM40) of HM-MT2401 EVB, as shown in Figure 7.3.

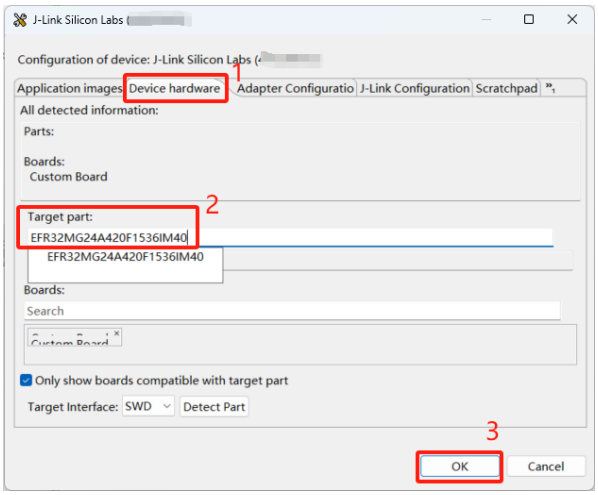

Figure 7.3

Then go back to the directory (GNU ARM v10.3.1 – Default) of the project. Select the s37 or hex format firmware, right-click and select "Flash to Device" to open the burning window, click "Program" to burn the firmware to EVB, as shown in figure 7.4.

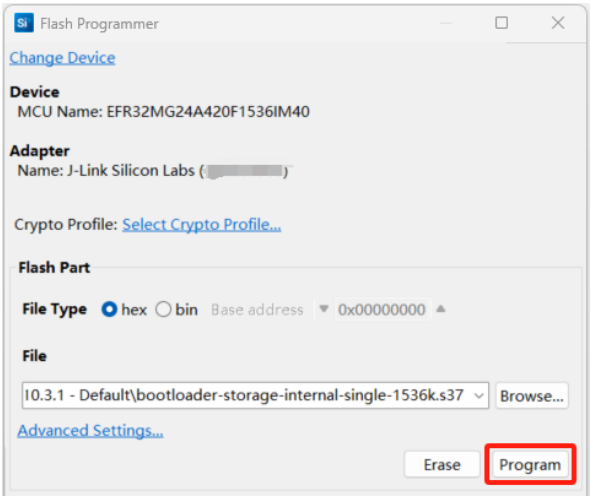

Figure 7.4

Note: When manufacturing in the factory, the module has already been burned with the Matter DAC certificate and the QR code information. **DO NOT ERASE** the chip when reburning the firmware. Otherwise, the DAC certificate will also be erased, causing subsequent Matter commissoining to fail.

7.2 Use Simplicity Commander to burn firmware

The Simplicity Commander tool can be found in the SSV5 installation directory, such as: C:\SiliconLabs\SimplicityStudio\v5\developer\adapter_packs\commander\commander.exe. Please refer to this document for how to use Commander: UG162: Simplicity Commander Reference Guide.

The method of using "Simplicity Commander" tool is as follows, as shown in figure 7.5.

1) Select the connected JLink device

2) Confirm that Debug Interface is SWD mode

3) Confirm the JLink connection rate (default is 1200kHz, if the connected wire is too long, the rate can be reduced appropriately)

4) Click on the "Flash" option

5) Click "Browse..." to import the burned firmware (s37 or hex format)

6) Click the "Flash" to start burning

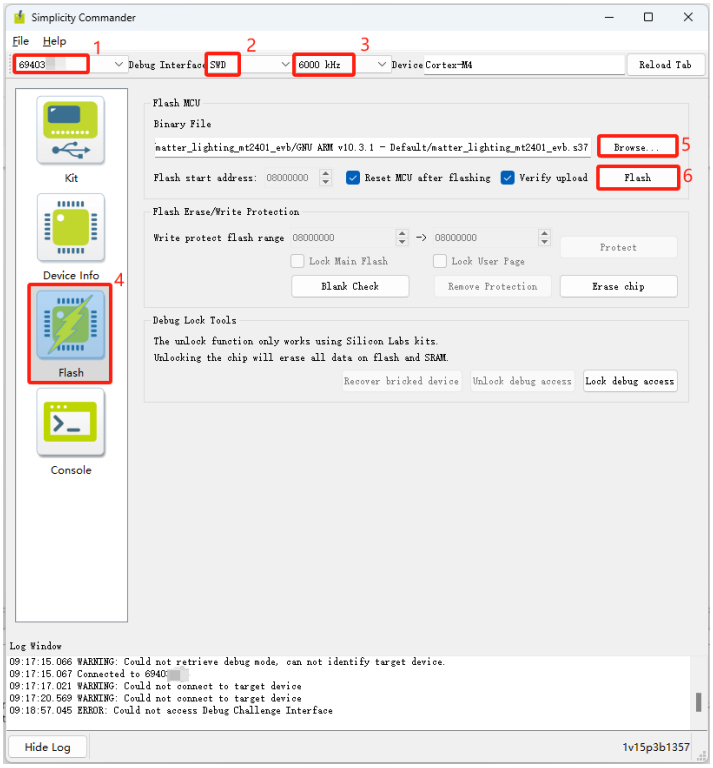

Figure 7.5

8 View Matter device log

The logs on the Matter device are output through the RTT interface by default. real time logs can be read using the JLink debugger and displayed via Simplicity Commander's RTT terminal. Open the Simplicity Commander tool, as shown in figure 8.1.

- Select the connected JLink device
- Confirm that the debug Interface is SWD mode
- Click on the "Console" option
- Check the RTT
- Click Connect, and the log information of the Matter device will be printed in the window

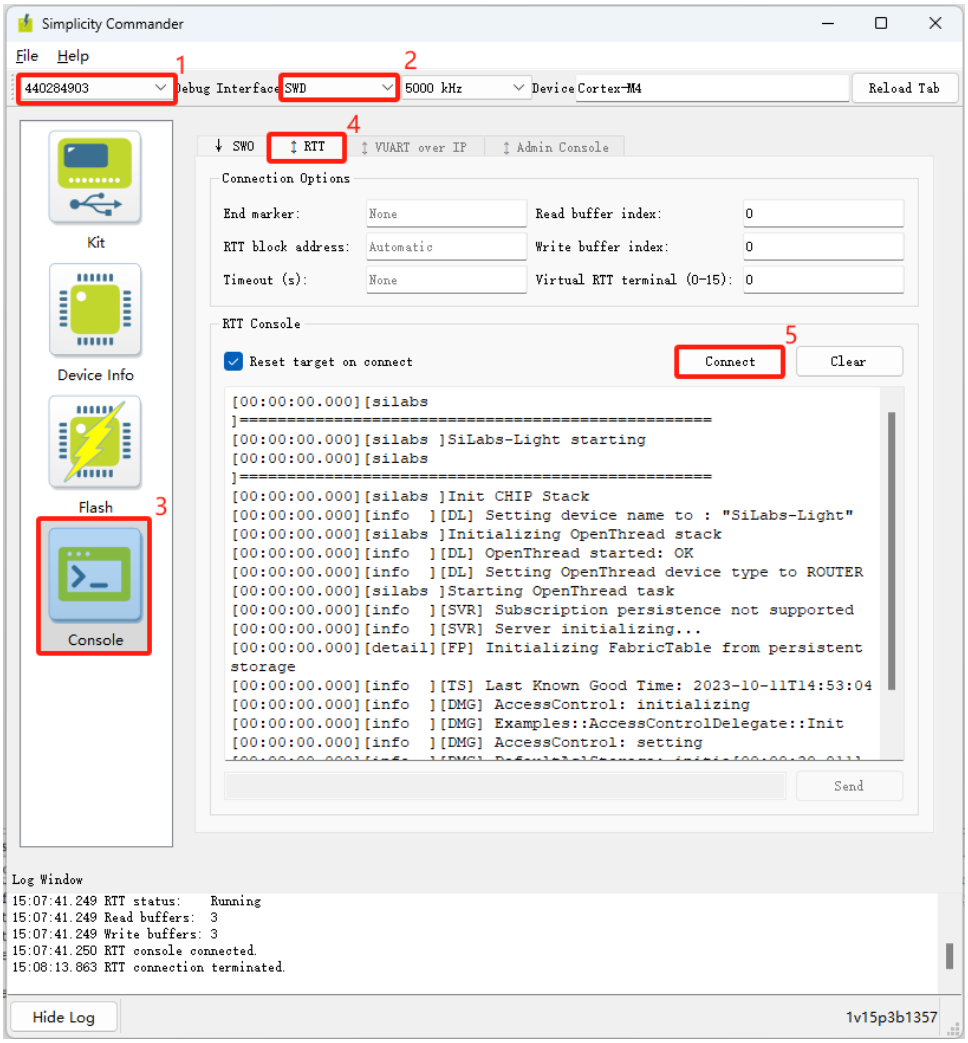

Figure 8.1

9 Matter CLI

Matter firmware provides CLI commands, CLI commands are input and output through the Serial terminal. The HM-MT2401 EVB provides a USB-to-serial port, PC can connect the EVB by Type-C port. The default baud rate of the serial port is 115200.

Run Matter CLI as follows:

```
MatterCli> help
  base64          Base64 encode / decode utilities
  exit            Exit the shell application
  help            List out all top level commands
  version         Output the software version
  ble             BLE transport commands
  config          Manage device configuration. Usage to dump value: config [param_name] and to set some values (discriminator): config [param_name] [param_value].
  device          Device management commands
  onboardingcodes Dump device onboarding codes. Usage: onboardingcodes none|softap|ble|onnetwork [qrcode|qrcodeurl|manualpairingcode]
  dns             Dns client commands
  ota             OTA commands
  echo            Echo back provided inputs
  log             Logging utilities
  rand            Random number utilities
  otcli           Dispatch OpenThread CLI command
```

For example, check the VID/PID of Matter device:

```
>  config
VendorId:       5232 (0x1470)
ProductId:      65281 (0xFF01)
HardwareVersion: 1 (0x1)
Done
```

Check the current thread network status of Matter device:

```
> otcli state

child

Done
```

Scan the current thread network:

```
> otcli scan

| PAN  | MAC Address      | Ch | dBm | LQI |

+------+------------------+----+-----+-----+

Done

MatterCli> | 060f | b23050db03f00000 | 11 | -82 | 255 |

| eb92 | b60002f0b08c0000 | 11 | -54 | 255 |

| fa8e | ae01096036640001 | 24 | -53 | 255 |

| 5ef9 | 2a0c60a1030f0002 | 25 | -38 | 255 |

| 5ef9 | 46cf20306fc00100 | 25 | -27 | 255 |

Done
```

10 Use chip-tool to commission the network

Before using chip-tool to configure the network, need to setup Open Thread Border Router (OTBR) first. If use Silicon Labs' Matter Hub as an OTBR, refer to Setting up the Matter Hub (Raspberry Pi) to setup OTBR. The OTBR consists of a Raspberry Pi and a HM-MT2401 EVB (using as RCP), as shown in figure 10.1.

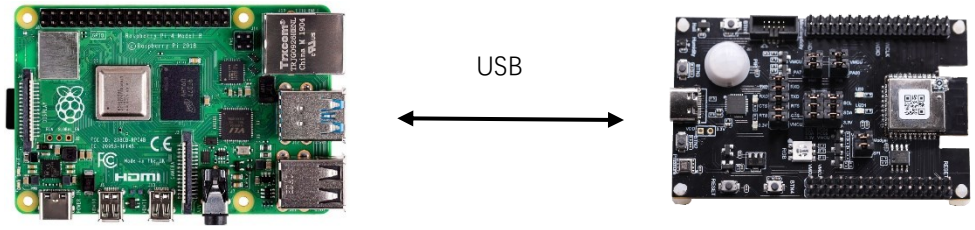

Figure 10.1

If need to obtain the RCP firmware of HM-MT2401 EVB, please contact the HOPERF technical support team.

10.1 Read the network parameters of OTBR

Execute the following command in OTBR:

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

```
ubuntu@MatterTool:~/connectedhomeip/out/standalone$ sudo ot-ctl dataset active -x
sudo: unable to resolve host MatterTool: Name or service not known
0e080000000000010000000300001835060004001fffe0020815c1d5d2462b71b70708fdb91f1d184caefb0510bfefc3f76f0d9c39da4c02a
3fae70c0402a0f7f8
Done
```

10.2 Obtain Setup QR Code

"Setup QR Code" of the HM-MT2401 EVB has been burned into device's flash. It can be obtained by scanning the QR code on the EVB or by viewing device's log (using RTT tool). After pressing EVB's BTN1 more than 6 seconds, the Setup QR Code will be printed out, as follows:

```
[00:04:10.568][info ][SVR] SetupQRCode: [MT:K2CA0IR.03T3C.0C320]
[00:04:10.568][info ][SVR] Copy/paste the below URL in a browser to see the QR Code:
```

```
[00:04:10.568][info ][SVR] https://project-chip.github.io/connectedhomeip/qrcode.html?data=MT%3AK2CA0IR.03T3C.0C320
```

10.3 Obtain discriminator/passcode from "Setup QR Code"

Run the following command:

```
./chip-tool payload parse-setup-payload MT:K2CA0IR.03T3C.0C320
```

The result is as follows:

```
/out/standalone$ ./chip-tool payload parse-setup-payload MT:K2CA0IR.03T3C.0C320
CHIP:SPL: Parsing base38Representation: MT:K2CA0IR.03T3C.0C320
CHIP:SPL: Version:              0
CHIP:SPL: VendorID:             5232
CHIP:SPL: ProductID:            65281
CHIP:SPL: Custom flow:          0    (STANDARD)
CHIP:SPL: Discovery Bitmask:    0x02 (BLE)
CHIP:SPL: Long discriminator:   1979   (0x7bb)
CHIP:SPL: Passcode:             98766751
```

10.4 Import PAA certificate to OTBR

The "DAC certificate" of the HM-MT2401 EVB has been burned into MG24. when use chip-tool to add the EVB to the Matter network, the DAC certificate chain must be verified. So, need to put the HOPERF PAA certificate into the environment where chip-tool is running, and configure the path of the PAA certificate through "--paa-trust-store-path" parameter, as shown below.

```
ubuntu@MatterTool:~/connectedhomeip/out/standalone$ ./chip-tool pairing ble-thread 45 hex:0e
080000000000010000000300001835060004001fffe0020815c1d5d2462b71b70708fdb91f1d184caefb0510bfef
c3f76f0d9c39da4c02a9dab3b380030f4f70656e5468726561642d666138650102fa8e0410718297753da1eabef2
0182135ad3fae70c0402a0f7f8 98766751 1979 --paa-trust-store-path ~/hoperf-paa/
```

For example:

The "HOPERFPAACertificateChain.der" file can be placed in the "~/hoperf-paa/". If need to obtain HOPERF's PAA certificate, please contact HOPERFP's technical support team.

10.5 Run chip-tool to add the EVB to the Matter network

Use the "Open thread network parameters" (refer to chapter 10.1) as "operationalDataset", and get "Passcode/discriminator" from "Setup QR Code" (refer to chapter 10.3), fill these into the following command.

```
./chip-tool pairing ble-thread node-id operationalDataset passcode discriminator [--paa-trust-store-path]
```

Note: Before configuring the network, must press and hold the "BTN1" button of the HM-MT2401 EVB for more than 6 seconds to do factory reset. The network commissioning command is as follows:

```
ubuntu@MatterTool:~/connectedhomeip/out/standalone$ ./chip-tool pairing ble-thread 45 hex:0e
080000000000010000000300001835060004001fffe0020815c1d5d2462b71b70708fdb91f1d184caefb0510bfef
c3f76f0d9c39da4c02a9dab3b380030f4f70656e5468726561642d666138650102fa8e0410718297753da1eabef2
0182135ad3fae70c0402a0f7f8 98766751 1979 --paa-trust-store-path ~/hoperf-paa/
```

If the following information is printed, it indicates that the network commissioning is successful.

```
[1697012018.052826][103482:103484] CHIP:SC: SecureSession[0xffffac024800]: Moving from state 'kActive' --> 'kPend
ingEviction'
[1697012018.053006][103482:103484] CHIP:IN: SecureSession[0xffffac024800]: Released - Type:1 LSID:40994
[1697012018.053111][103482:103484] CHIP:CTL: Successfully finished commissioning step 'Cleanup'
[1697012018.053200][103482:103484] CHIP:TOO: Device commissioning completed with success
[1697012018.053323][103482:103483] CHIP:DL: BluezDisconnect peer=B0:C7:DE:61:CC:B3
[1697012018.053373][103482:103484] CHIP:DMG: ICR moving to [AwaitingDe]
[1697012018.054403][103482:103484] CHIP:EM: <<< [E:6462i M:250994469 (Ack:238000288)] (S) Msg TX to 1:00000000000
000EA [2BE9] --- Type 0000:10 (SecureChannel:StandaloneAck)
[1697012018.054515][103482:103484] CHIP:IN: (S) Sending msg 250994469 on secure session with LSID: 40995
[1697012018.056077][103482:103484] CHIP:EM: Flushed pending ack for MessageCounter:238000288 on exchange 6462i
[1697012018.056851][103482:103482] CHIP:CTL: Shutting down the commissioner
[1697012018.056957][103482:103482] CHIP:CTL: Stopping commissioning discovery over DNS-SD
[1697012018.057247][103482:103482] CHIP:CTL: Shutting down the controller
[1697012018.057309][103482:103482] CHIP:IN: Expiring all sessions for fabric 0x1!!
[1697012018.057359][103482:103482] CHIP:IN: SecureSession[0xffffac028880]: MarkForEviction Type:2 LSID:40995
[1697012018.057407][103482:103482] CHIP:SC: SecureSession[0xffffac028880]: Moving from state 'kActive' --> 'kPend
ingEviction'
[1697012018.057455][103482:103482] CHIP:IN: SecureSession[0xffffac028880]: Released - Type:2 LSID:40995
[1697012018.057517][103482:103482] CHIP:FP: Forgetting fabric 0x1
[1697012018.057583][103482:103482] CHIP:TS: Pending Last Known Good Time: 2023-02-21T10:33:46
[1697012018.057996][103482:103482] CHIP:TS: Previous Last Known Good Time: 2023-02-21T10:33:46
[1697012018.058063][103482:103482] CHIP:TS: Reverted Last Known Good Time to previous value
[1697012018.058162][103482:103482] CHIP:CTL: Shutting down the commissioner
[1697012018.058212][103482:103482] CHIP:CTL: Stopping commissioning discovery over DNS-SD
[1697012018.058512][103482:103482] CHIP:CTL: Shutting down the controller
[1697012018.058569][103482:103482] CHIP:CTL: Shutting down the System State, this will teardown the CHIP Stack
[1697012018.059252][103482:103482] CHIP:DMG: IM WH moving to [Uninitialized]
[1697012018.059323][103482:103482] CHIP:DMG: IM WH moving to [Uninitialized]
[1697012018.059369][103482:103482] CHIP:DMG: IM WH moving to [Uninitialized]
[1697012018.059414][103482:103482] CHIP:DMG: IM WH moving to [Uninitialized]
[1697012018.059464][103482:103482] CHIP:DMG: All ReadHandler-s are clean, clear GlobalDirtySet
[1697012018.059681][103482:103482] CHIP:BLE: BleConnectionDelegate::CancelConnection is not implemented.
[1697012018.059778][103482:103482] CHIP:FP: Shutting down FabricTable
[1697012018.059966][103482:103482] CHIP:TS: Pending Last Known Good Time: 2023-02-21T10:33:46
[1697012018.060234][103482:103482] CHIP:TS: Previous Last Known Good Time: 2023-02-21T10:33:46
[1697012018.060405][103482:103482] CHIP:TS: Reverted Last Known Good Time to previous value
[1697012018.060900][103482:103482] CHIP:DL: writing settings to file (/tmp/chip_counters.ini-V8hwT1)
[1697012018.062531][103482:103482] CHIP:DL: renamed tmp file to file (/tmp/chip_counters.ini)
[1697012018.062694][103482:103482] CHIP:DL: NVS set: chip-counters/total-operational-hours = 0 (0x0)
[1697012018.062747][103482:103482] CHIP:DL: Inet Layer shutdown
[1697012018.062794][103482:103482] CHIP:DL: BLE shutdown
[1697012018.065789][103482:103482] CHIP:DL: System Layer shutdown
ubuntu@MatterTool:~/connectedhomeip/out/standalone$
```

10.6 Use chip-tool to control LED

The chip-tool command format is as follows:

```
chip-tool onoff toggle node-id endpoint-id
```

The "node-id" is specified during network configuration, "endpoint-id" refers to the endpoint number, the default value is 1, for example:

```
ubuntu@MatterTool:~/connectedhomeip/out/standalone$ ./chip-tool onoff toggle 45 1
```

11 Join Apple Home Ecosystem

The "Home Pod" or "Home Pod Mini" devices have integrated OTBR functions. Use them to commission Matter devices into the Apple Home Ecosystem.

First, make sure "Home Pod Mini" and the mobile phone are added to the same Wi-Fi router; then open the "Home" APP on the "IPhone/IPad"; then confirm that "Home Pod Mini" has been added to the home, as shown in figure 11.1.

Figure 11.1

Before configuring network, must press and hold the "BTN1" button of the HM-MT2401 EVB for more than 6 seconds to do factory reset; open the "Home" APP in "Iphone/Ipad", then click "Add Accessories" in the upper right corner, scan the QR code of the Matter device, start configuring network process. as shown in figure 11.2.

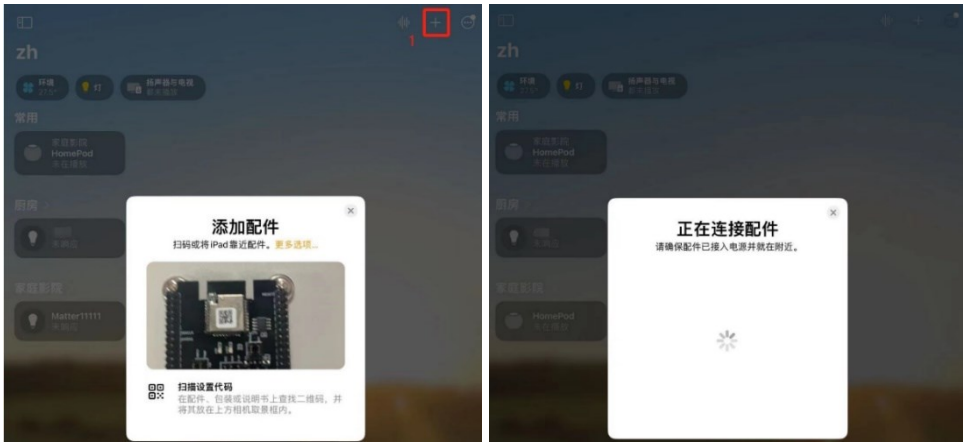

Figure 11.2

After the device is configured successfully, set the device name, as shown in figure 11.3.

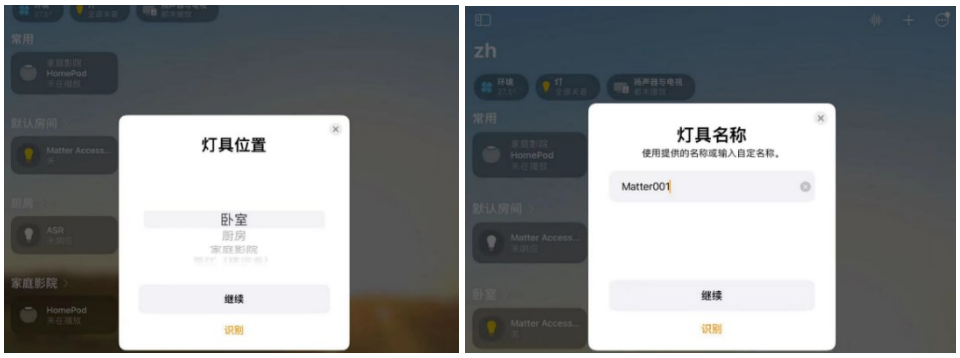

Figure 11.3

After successfully adding the Matter Lighting device, it will be displayed in the "Home" APP, as shown in figure 11.4.

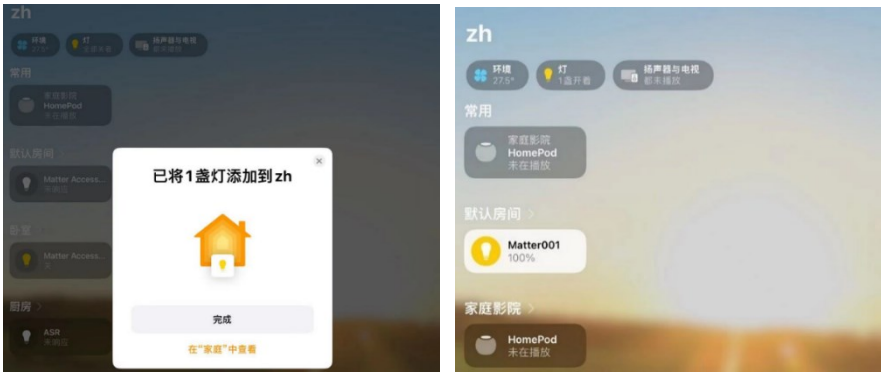

Figure 11.4

In the "Home" APP, click the icon of Matter Lighting Device to control open or close, etc.

12 Revision History

Table 12.1

Versions	Description	Updated date
V1.0	first release	2023.8.30

13 Contact Information

Shenzhen Hope Microelectronics Co., Ltd.

Address: 30th floor of 8th Building, C Zone, Vanke Cloud City, Xili Sub-district, Nanshan, Shenzhen, GD, P.R. China

Tel: + 86-0755-82973805

Email: sales@hoperf.com

Website: https://www.hoperf.com/