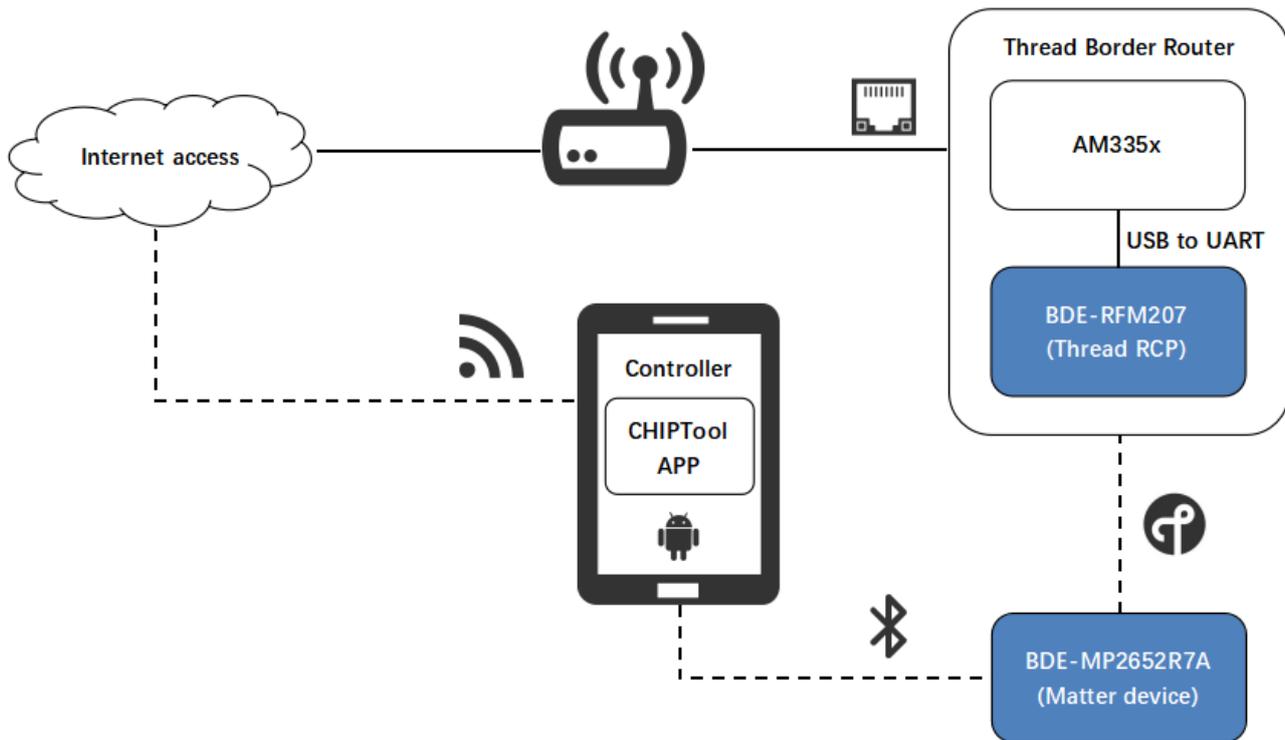


Matter Over Thread Demo Use Guide

1. Description

An application showing the use of Matter on BDE Thread modules. The result should resemble this high-level block diagram:



2. Get Ready

2.1. Software and Tools

- A Linux development environment is assumed (such as native Linux machine, or through VirtualBox/Vagrant).
- Matter*: <https://github.com/TexasInstruments/matter/tree/v1.0-ti-branch>
- Thread: RCP image for BDE-RFM207 module: rcp_CC26X2R1_LAUNCHXL_tirtos_ticlang.out (The RCP examples are built in [6.20 CC13xx/CC26xx SDK](#)).
- Thread Border Router: <https://github.com/openthread/ot-br-posix>

2.2. Hardware

- 1 x [BDE-RFM207](#) module
- 1 x [BDE-MP2652R7A](#) module
- 1 x [BeagleBone Black](#)
- 2 x Evaluation boards of [BDE-EVB07](#)

3. Setting up your environment for Matter

The following should be done in your Linux environment. You will need to set it up on a Linux PC (used to build the firmware of the Matter device).

3.1. Checking out the Matter code

- Execute the following commands:

- `$ cd ~`
- `$ git clone --recurse-submodules https://github.com/TexasInstruments/matter.git`

If you already have a checkout, run the following command to sync submodules:

- `$ git submodule update --init`

3.2. Installing prerequisites on Linux

- Execute the following commands:

- `$ 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`

4. Prepare the lighting-app (on Linux PC)

4.1. Download and install Sysconfig

- This can be done simply with the following commands:

- `$ cd ~`
- `$ wget https://dr-download.ti.com/software-development /ide-configuration-compiler-or-debugger/MD-nsUM6f7Vvb/1.15.0.2826/sysconfig-1.15.0_2826-setup.run`
- `$ chmod +x sysconfig-1.15.0_2826-setup.run`
- `$./sysconfig-1.15.0_2826-setup.run`

4.2. Run the bootstrap script to setup the build environment

- Execute the following commands:

- `$ cd ~/matter`
- `$ source ./scripts/bootstrap.sh`

4.3. Activate the build environment with the repository activate script

It is necessary to activate the environment in every new shell. Then run GN and Ninja to build the executable.

- Execute the following commands:

- `$ cd ~/matter`
- `$ source ./scripts/activate.sh`

4.4. Run the build to produce a default executable

By default on Linux both the TI SimpleLink SDK and Sysconfig are located in a TI folder in the user's home directory, and you must provide the absolute path to them. For example /home/username/ti/sysconfig_1.15.0.. Take note of this installation path, as it will be used in the next step.

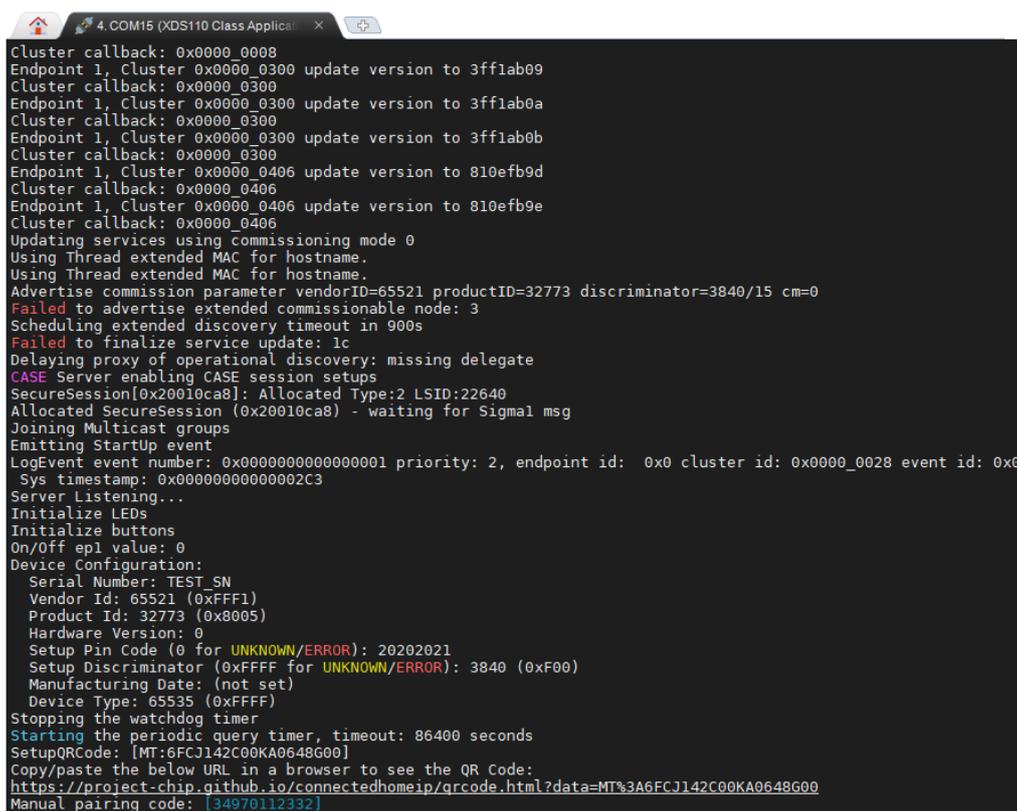
- This can be done with the following commands:
 - \$ cd ~/matter/examples/lighting-app/cc13x2x7_26x2x7
 - \$ gn gen out/debug --args="ti_sysconfig_root=\"\$HOME/ti/sysconfig_1.15.0\""
 - \$ ninja -C out/debug

4.5. Programming with UniFlash

Download and install [UniFlash](#).

Select the ELF image to load on the device with the Browse button. This file is placed in the out/debug folder by this guide. First flash the *-example-bim.hex, followed by the *-example.bin.

Finally click the Load Image button to load the executable image onto the device. You should be able to see the log output over the XDS110 User UART.



```
Cluster callback: 0x0000_0008
Endpoint 1, Cluster 0x0000_0300 update version to 3ff1ab09
Cluster callback: 0x0000_0300
Endpoint 1, Cluster 0x0000_0300 update version to 3ff1ab0a
Cluster callback: 0x0000_0300
Endpoint 1, Cluster 0x0000_0300 update version to 3ff1ab0b
Cluster callback: 0x0000_0300
Endpoint 1, Cluster 0x0000_0406 update version to 810efb9d
Cluster callback: 0x0000_0406
Endpoint 1, Cluster 0x0000_0406 update version to 810efb9e
Cluster callback: 0x0000_0406
Updating services using Commissioning mode 0
Using Thread extended MAC for hostname.
Using Thread extended MAC for hostname.
Advertise commission parameter vendorID=65521 productID=32773 discriminator=3840/15 cm=0
Failed to advertise extended commissionable node: 3
Scheduling extended discovery timeout in 900s
Failed to finalize service update: 1c
Delaying proxy of operational discovery: missing delegate
CASE Server enabling CASE session setups
SecureSession[0x20010ca8]: Allocated Type:2 LSID:22640
Allocated SecureSession (0x20010ca8) - waiting for Signal msg
Joining Multicast groups
Emitting StartUp event
LogEvent event number: 0x0000000000000001 priority: 2, endpoint id: 0x0 cluster id: 0x0000_0028 event id: 0x0
Sys timestamp: 0x000000000000002C3
Server Listening...
Initialize LEDs
Initialize buttons
On/Off epl value: 0
Device Configuration:
  Serial Number: TEST_SN
  Vendor Id: 65521 (0xFFFF1)
  Product Id: 32773 (0x8005)
  Hardware Version: 0
  Setup Pin Code (0 for UNKNOWN/ERROR): 20202021
  Setup Discriminator (0xFFFF for UNKNOWN/ERROR): 3840 (0xF00)
  Manufacturing Date: (not set)
  Device Type: 65535 (0xFFFF)
Stopping the watchdog timer
Starting the periodic query timer, timeout: 86400 seconds
SetupQRCode: [MT:6FCJ142C00KA0648G00]
Copy/paste the below URL in a browser to see the QR Code:
https://project-chip.github.io/connectedhomeip/qrcode.html?data=MT%3A6FCJ142C00KA0648G00
Manual pairing code: [34970112332]
```

Copy/paste the below URL in a browser to see the QR Code:

<https://project-chip.github.io/connectedhomeip/qrcode.html?data=MT%3A6FCJ142C00KA0648G00>

5. Prepare the Matter controller

5.1. Source files

You can find source files of the Android applications in the `~/matter/examples/android` directory.

5.2. Requirements for building

You need Android SDK 21 & NDK 21.4.7075529 downloaded to your machine. Set the `$ANDROID_HOME` environment variable to where the SDK is downloaded and the `$ANDROID_NDK_HOME` environment variable to point to where the NDK package is downloaded.

- Install [Android Studio](#)
- Install NDK:
 - Tools -> SDK Manager -> SDK Tools Tab
 - Click [x] Show Package Details
 - Select NDK (Side by Side) -> 21.4.7075529
 - Apply
- Install Command Line Tools:
 - Tools -> SDK Manager -> SDK Tools Tab -> Android SDK Command Line Tools (latest)
 - Apply
- Install SDK 21:
 - Tools -> SDK Manager -> SDK Platforms Tab -> Android 5.0 (Lollipop) SDK Level 21
 - Apply
- Install Emulator:
 - Tools -> Device Manager -> Create device -> Pixel 5 -> Android S API 31 -> Download
- Execute the following commands in the shell (Linux):
 - `$ export ANDROID_HOME=~/.Android/Sdk`
 - `$ export ANDROID_NDK_HOME=~/.Android/Sdk/ndk/21.4.7075529`

5.3. ABIs and TARGET_CPU

`TARGET_CPU` can have the following values, depending on your smartphone CPU architecture:

ABI	TARGET_CPU
armeabi-v7a	arm
arm64-v8a	arm64
x86	x86
x86_64	x86_64

5.4. Preparing for build

- Run bootstrap (only required for the first time):
 - `$ cd ~/matter`
 - `$ source scripts/bootstrap.sh`

5.5. Building Android CHIPTool from scripts

- In the command line, run the following command from the top CHIP directory:
 - `$./scripts/build/build_examples.py --target android-arm64-chip-tool build`
- See the table above for other values of `TARGET_CPU`.
- The debug Android package `app-debug.apk` will be generated at `out/android-$TARGET_CPU-chip-tool/outputs/apk/debug/`, and can be installed with
 - `$ adb install out/android-$TARGET_CPU-chip-tool/outputs/apk/debug/app-debug.apk`

6. Prepare the Thread Border Router

- Build the border router - instructions here: <https://openthread.io/guides/border-router/build>
- For the `ot-br-posix` repository, it is recommended to use the commit specified above, in the "[Get Ready](#)" section.
- Use the RCP image specified above, in the "[Get Ready](#)" section.

7. Running Demo

It is assumed that these devices are all on the same WLAN network (e.g. this can be accomplished by connecting all to a single Wi-Fi Router): Border Router and Matter controller

7.1. Start the Thread network on Border Route

- Start the `otbr-agent` service:
 - `$ sudo systemctl start otbr-agent.service`
- Create the network with the following commands:
 - `$ 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`
- Submit network configuration:
 - `$ sudo ot-ctl dataset commit active`
- Enable the IPV6 Interface:
 - `$ sudo ot-ctl ifconfig up`
- Start the Thread network:
 - `$ sudo ot-ctl thread start`

7.2. Device pairing

- Open the CHIPTool APP, click "PROVISION CHIP DEVICE WITH THREAD" button and scan the [QR Code](#) generated by the above link. After the scan is successful, the following will be displayed:

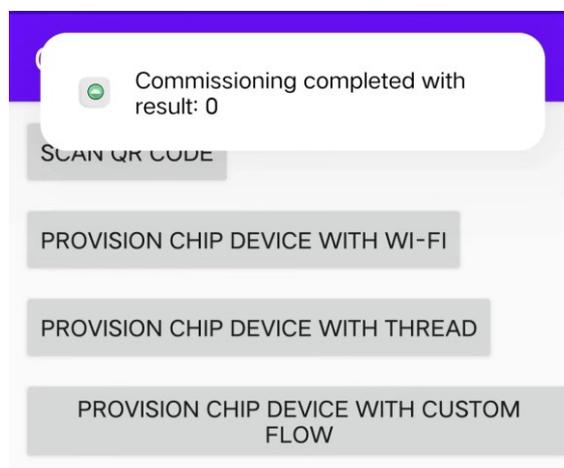


The screenshot shows the CHIPTool app interface with the following fields and values:

- Channel:** 15
- PAN ID:** 1234
- Extended PAN ID:** 11:11:11:11:22:22:22:22
- Master Key:** 00:11:22:33:44:55:66:77:88:99:AA:BB:CC:DD:EE:FF

A "SAVE NETWORK" button is located at the bottom right of the form.

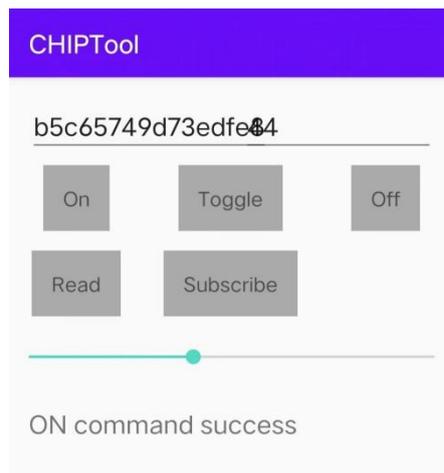
- On the premise that Matter device (BDE-MP2652R7A) is running properly, press and hold the right button labeled BTN-2 on BDE-EVB07 for more than 1 second. After the launch, Bluetooth LE advertising will begin. Once the device is fully configured, BLE advertising will stop.
- Click the "SAVE NETWORK" button in the above interface of the APP to enter the pairing mode. The following message will pop up when the device is successfully paired:



You can also see the log output through the XDS110 User UART:

```
t)
Handling via exchange: 56577r, Delegate: 0x200096dc
Received command for Endpoint=0 Cluster=0x0000_0030 Command=0x0000_0004
GeneralCommissioning: Received CommissioningComplete
Metadata for Fabric 0x1 persisted to storage.
Committing Last Known Good Time to storage: 2023-08-01T17:12:38
OpCreds: Fabric index 0x1 was committed to storage. Compressed Fabric Id 0xB5C65749D73EDFE8, FabricId 00000000
000000001, NodeId 0000000000000002C, VendorId 0xFFFF4
GeneralCommissioning: Successfully committed pending fabric data
Fail-safe cleanly disarmed
Endpoint 0, Cluster 0x0000_0030 update version to 54d75e02
Cluster callback: 0x0000_0030
Command handler moving to [Preparing]
Command handler moving to [AddingComm]
Command handler moving to [AddedComm]
Decreasing reference count for CommandHandler, remaining 0
<<< [E:56577r M:166591292 (Ack:172538075)] (S) Msg TX to 1:000000000001B669 [DFE8] --- Type 0001:09 (IM:InvokeCommandResponse)
(S) Sending msg 166591292 on secure session with LSID: 22640
Thread packet SENT: UDP, len 118
  src FD43:CE22:86D4:1:29AD:3BEA:CB3:C07A, port 5540
  dest FD11:1111:1122:2222:89BB:EDA9:707A:EF9C, port 5541
Command handler moving to [CommandSen]
Command handler moving to [AwaitingDe]
Long dispatch time: 209 ms, for event type 3
>>> [E:56576r M:109123238 (Ack:258897440)] (U) Msg RX from 0:29038265B2958C3C [0000] --- Type 0000:10 (SecureChannel:StandaloneAck)
Found matching exchange: 56576r, Delegate: 0
Rxd Ack; Removing MessageCounter:258897440 from Retrans Table on exchange 56576r
Commissioning completed successfully
Updating services using commissioning mode 0
Using Thread extended MAC for hostname.
Advertise operational node B5C65749D73EDFE8-000000000000002C
Using Thread extended MAC for hostname.
Advertise commission parameter vendorID=65521 productID=32773 discriminator=3840/15 cm=0
Expiring all PASE sessions
SecureSession[0x20010d60]: MarkForEviction Type:1 LSID:22641
SecureSession[0x20010d60]: Moving from state 'kActive' --> 'kPendingEviction'
SecureSession[0x20010d60]: Released - Type:1 LSID:22641
Clearing BLE pending packets.
Releasing end point's BLE connection back to application.
Device commissioned, schedule a default provider query
Commissioning complete, notify platform driver to persist network credentials.
Thread packet RCV: UDP, len 107
  src FD11:1111:1122:2222:89BB:EDA9:707A:EF9C, port 5541
  dest FD43:CE22:86D4:1:29AD:3BEA:CB3:C07A, port 5540
```

- You can click "LIGHT ON/OFF & LEVEL CLUSTER" on the main interface of the APP to enter the operation interface and control the Matter device.



- On: Switch on the LED
- Off: Switch off the LED
- Toggle: Toggle the status of LED
- Read: Read the current status of the LED

By now, you should run the lighting-app demo successfully.

8. Revision History

Revision	Date	Description
V0.1	3-August-2023	Preliminary version, draft

You can find the latest documentations with this [Link](#).

Important Notice and Disclaimer

The information contained herein is believed to be reliable. BDE makes no warranties regarding the information contained herein. BDE assumes no responsibility or liability whatsoever for any of the information contained herein. BDE assumes no responsibility or liability whatsoever for the use of the information contained herein. The information contained herein is provided "AS IS, WHERE IS" and with all faults, and the entire risk associated with such information is entirely with the user. All information contained herein is subject to change without notice. Customers should obtain and verify the latest relevant information before placing orders for BDE products. The information contained herein or any use of such information does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other intellectual property rights, whether with regard to such information itself or anything described by such information.

Contact

BDE Technology Inc.

USA: 67 E Madison St, # 1603A, Chicago, IL 60603, US

Tel: +1-312-379-9589

Website: <http://www.bdecomm.com> Email: info@bdecomm.com

China: B2-403, 162 Science Avenue, Huangpu District, Guangzhou 510663, China

Tel: +86-20-28065335

Website: <http://www.bdecomm.com> Email: shu@bdecomm.com