UG10178

Quick Start to the Matter Demo NXP CHIP Tool app, FRDM-RW612 and FRDM-MCU W71

Rev. 1.0 — 2 December 2024 User guide

Document information

Information	Content
Keywords	UG10178, Matter application, Matter, evaluation kit, FRDM- RW612, RW612, FRDM, W71, FRDM-MCUW71, image, pre-built package, configuration files, scripts.
Abstract	The document explains how to set NXP Matter Reference Kit components and run a Matter application example.

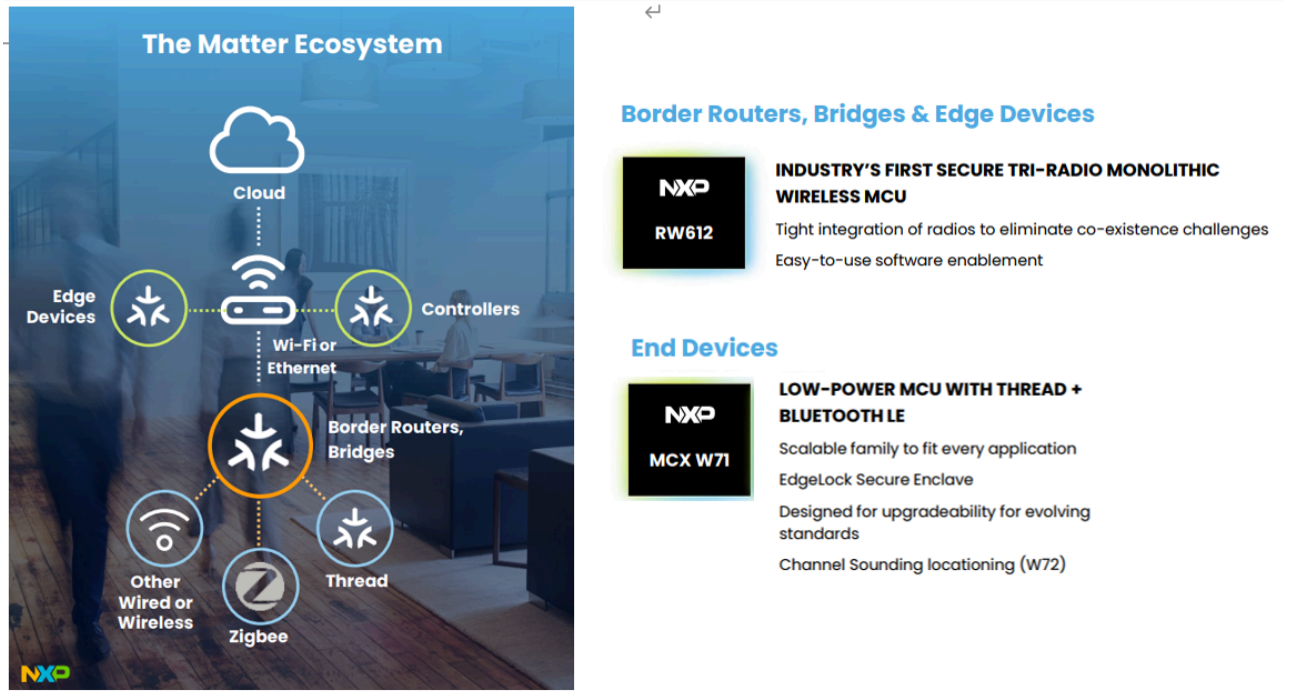
1 Introduction

Matter is a new single, unified, application-layer connectivity standard designed to enable developers to connect and build reliable, secure IoT (Internet of Things) ecosystems and increase compatibility among Smart Home and Building devices.

For enabling Matter devices, NXP offers scalable, flexible, and secure platforms to enable the variety of use cases Matter addresses – from end nodes to gateways, so device manufacturers can focus on product innovation and accelerating time to market.

This document is a step-by-step walkthrough guide to set up and run the Matter project with NXP products. The setup consists of Matter over Wi-Fi, Matter over Thread devices and an Android phone running NXP's Matter controller tool. It contains operating instructions using NXP RW612 and MCX W71 FRDM boards, also include the instructions to build the binary from source code and flash the DUT.

For more NXP Matter related product information on using the release package, see the following page on the NXP website - https://www.nxp.com/applications/enabling-technologies/connectivity/matter:MATTER.



Border Routers, Bridges & Edge Devices

INDUSTRY'S FIRST SECURE TRI-RADIO MONOLITHIC WIRELESS MCU

Tight integration of radios to eliminate co-existence challenges

Easy-to-use software enablement

End Devices

LOW-POWER MCU WITH THREAD + BLUETOOTH LE

Scalable family to fit every application

EdgeLock Secure Enclave

Designed for upgradeability for evolving standards

Channel Sounding locationing (W72)

Figure 1. NXP Matter Ecosystem and the RW612 and MCX W71 chip.

2 Matter environment

Application Notes are engineering support documents that assist the user in evaluating the operation of a device product line, package type, or general application topic. An Application note contains real-world application information about how a specific Freescale device or product family is used with other Freescale or vendor parts and or software to address a particular technical issue. Parts and or software must already exist and be available.

2.1 Hardware requirements

The following hardware components are required for this reference kit:

- Linux host computer
- Wi-Fi access point
- Android smartphone/tablet (running Android 13 or later)
- FRDM-RW612 board
- FRDM-MCXW71 board

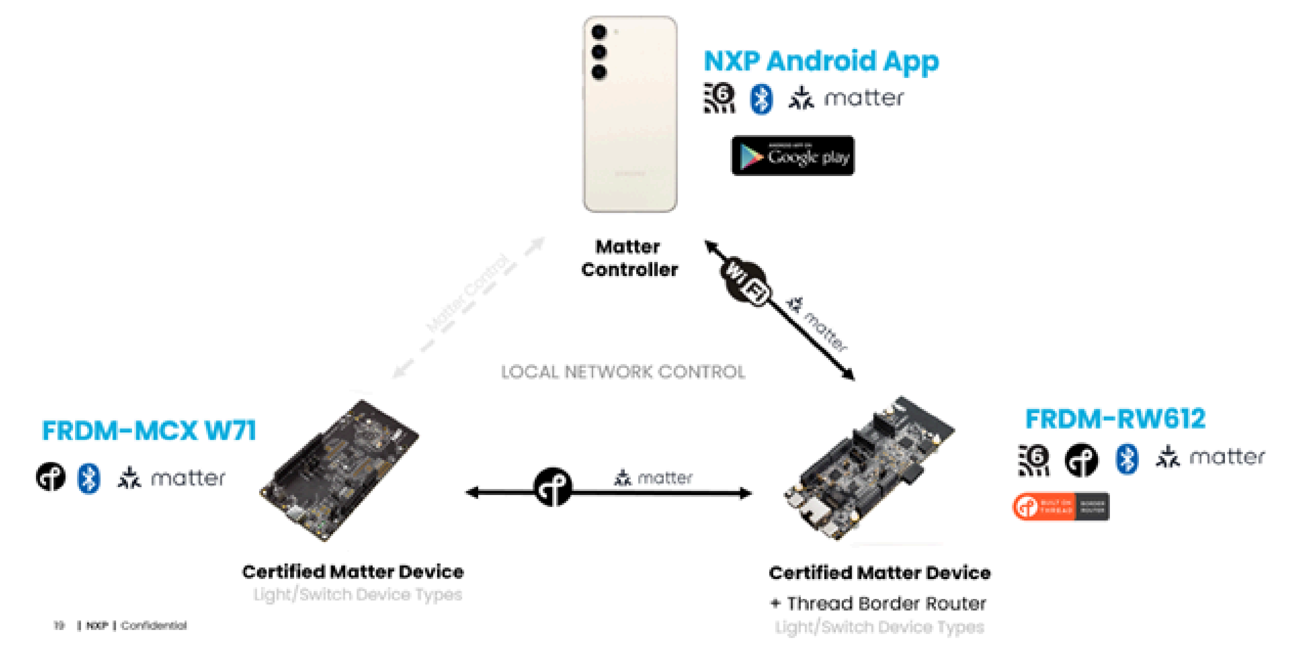


Figure 2. Matter applications based on NXP CHIP Tool App, FRDM- RW612 and FRDM-MCXW71

2.2 Software requirements

The following software is required for this reference kit:

- A serial COM terminal for the host computer

The full image contains the following SW packages that support Matter application examples:

```
── images
  ├── CHIP Tool app
  ├── RW612-FRDM
  │      └── chip-rw61x-light-switch-combo-example-frdm.srec
  └── MCU W71 FRDM
    ├── chip-mcxw71-light-switch-combo-example.srec
    └── mcxw71_nbu_ble_15_4_dyn_matter_1_0_17_1.sb3
```

3 Wi-Fi Light/Switch device and Thread Board Router – RW612

FRDM-RW612 is a compact and scalable development board for rapid prototyping of the RW61x series of Wi-Fi 6 + Bluetooth Low Energy + 802.15.4 tri-radio wireless MCUs.

It offers easy access to the MCU's I/Os and peripherals, integrated open-standard serial interfaces, external flash memory, and onboard MCU-Link debugger.

The example implements a server that can be accessed by a Matter controller and can accept basic cluster commands. The example is based on the NXP RW612 SDK and NXP Matter downstream and provides a prototype application that demonstrates device commissioning and different cluster control.

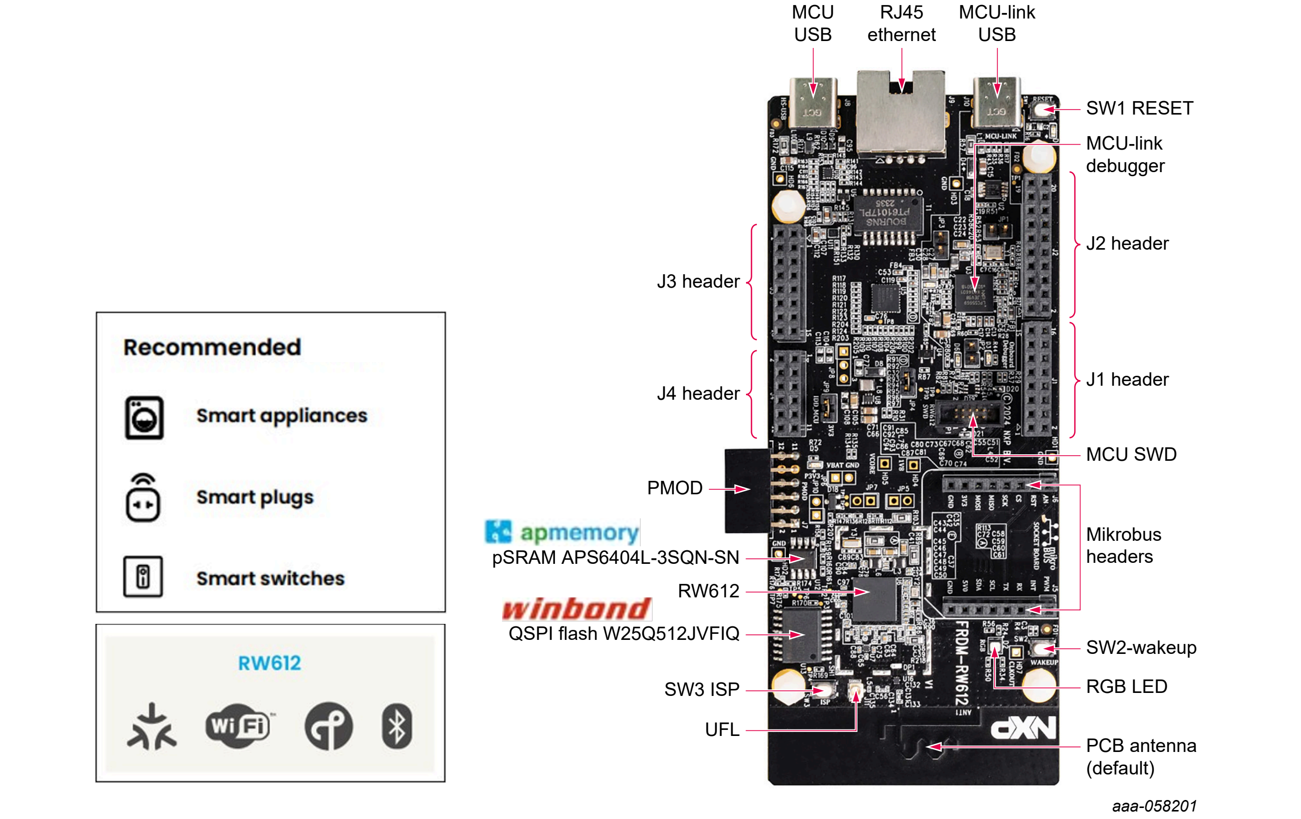


Figure 3. Recommended devices and support protocol for FRDM-RW612 with FRDM\-RW612 board top view

3.1 Build FRDM-RW612 light-switch-otbr-combo application

Matter supports configuring a build with GN, a fast and scalable meta-build system that generates inputs to ninja.

Note: *Commands are subject to change. See the "Readme" file for the branch you are trying to build (https://github.com/NXP/matter/tree/<default branch>/examples/all-clusters-app/nxp/rt/rw61x).*

1. Prerequisites
   Make sure that the below prerequisites are correctly installed with the following command:

```
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
```

   Use pip3 to install west:

```
pip3 install west
```

2. Check out the Matter code
   To check out the Matter repository with all platforms, run the following command:

```
git clone --recurse-submodules https://github.com/NXP/matter.git
```

If you already have the Matter code checked out, run the following commands to update the repository and synchronize submodules:

```
git pull
git submodule update --init
```

3. Checkout NXP-specific submodules only

```
user@ubuntu:~/Desktop/git/matter$ scripts/checkout_submodules.py --shallow --
platform nxp --recursive
```

4. Activate local environment

```
user@ubuntu:~/Desktop/git/matter $ source scripts/activate.sh
```

If the script says that the environment is out of date, update it by running the following command:

```
user@ubuntu:~/Desktop/git/matter $ source scripts/bootstrap.sh
```

5. Initialize Matter SDK(s)

```
user@ubuntu:~/Desktop/git/matter $ third_party/nxp/nxp_matter_support/
scripts/update_nxp_sdk.py --force
```

Note: *By default,* `update_nxp_sdk.py` *tries to initialize all NXP SDKs. Arg* `-- help` *can be used to view all available options.*

6. Building with Matter over Wi-Fi + OpenThread Border Router configuration on RW612
This configuration supports the Thread Border Router management cluster to provision the Thread credentials. Enabling the Matter CLI to control the Thread network on the Border Router is optional but recommended for other features like the Thread credential sharing.
Note: *The Thread Border Router management cluster is only supported by default on the thermostat application.*
Build Matter with Border Router configuration with Bluetooth LE commissioning (ble-wifi):

```
user@ubuntu:~/Desktop/git/matter/examples/light-switch-otbr-combo-app /nxp/
rt/rw610$ gn gen --args="chip_enable_wifi=true chip_enable_openthread=true
 chip_enable_matter_cli=true board_version=\"frdm\"" out/debug

user@ubuntu:~/Desktop/git/matter/examples/ light-switch-otbr-combo-app
/nxp/rt/rw610$ ninja -C out/debug
```

The image is generated in out/debug. The file name is `chip-rw61x-light-switch-combo-example`. It must be appended. elf as: `chip-rw61x-light-switch-combo-example.elf`.

3.1.1 General information

Optional GN parameters that may be added when building an application:

- To enable the matter CLI, add the argument `chip_enable_matter_cli=true` to the gn command.
- By default, the NXP RD-RW612-BGA board revision is chosen. To switch to the FRDM-RW612 board, add the argument board_version=\"frdm\" to the gn gen command.
- To build the application in debug mode, add the argument `is_debug=true optimize_debug=false` to the gn gen command.

For more options information, refer to Readme.

3.2 Flashing with JLink tool

There 2 ways to flash the application image to RW612 FRDM, Jlink, and MCUXpresso. The first way to flash the application on RW612 FRDM is using the Jlink tool.

1. Install JLink_Windows_7_89c
2. The original application file name is `chip-rw61x-light-switch-otbr-combo-example`. Add “.elf”after the file name: `chip-rw61x-light-switch-otbr-combo-example.elf`
3. Transfer the elf file to the serc file

```
arm-none-eabi-objcopy -R .NVM -O srec chip-rw61x-light-switch-combo-example chip-rw61x-light-switch-combo-example-frdm.srec
```

4. Create a file, `commands_script` with the following content (change the application name accordingly):

```
reset
halt
loadfile chip-rw61x-light-switch-combo-example-frdm.srec
reset
go
quit
```

5. Copy the application and `commands_script` in the folder where the JLink executable is placed. Execute:

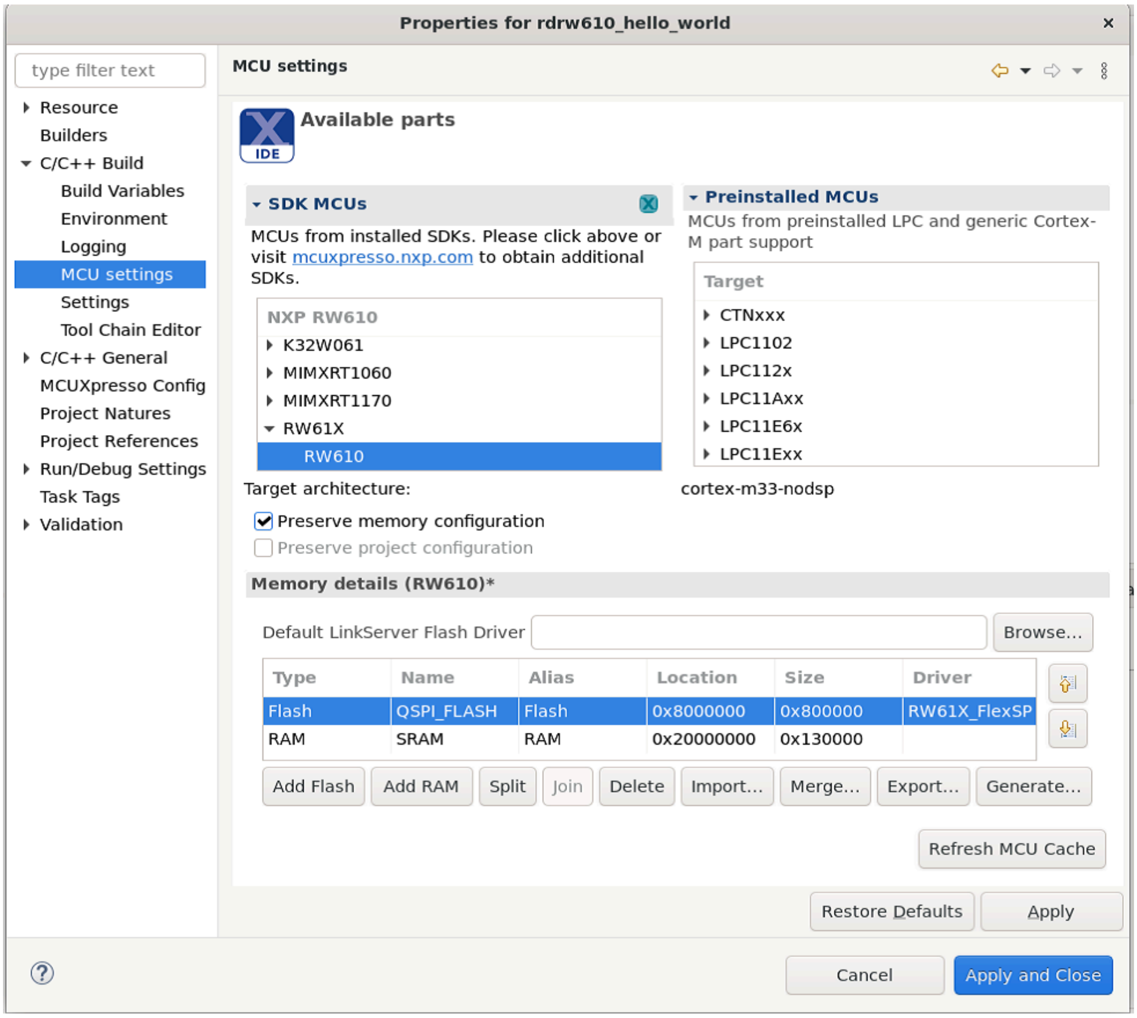
```
$  jlink -device RW612 -if SWD -speed 4000 -autoconnect 1 -CommanderScript
```

After the flash success, start the application on the RW612 FRDM.

3.3 Flashing with MCUXpresso IDE

To flash the application, we recommend using MCUXpresso IDE (version >= 11.6.0).

1. Import the previously downloaded NXP SDK into MCUXpresso IDE. To do this, right-click the empty space in the MCUXpresso IDE **Installed SDKs** tab to show the menu, select the **Import archive** (or **Import folder** if a folder is used) menu item.
2. Import the connectedhomeip repo in MCUXpresso IDE as a Makefile Project. Use none as Toolchain for Indexer Settings: `File -> Import -> C/C++ -> Existing Code as Makefile Project`
3. Configure MCU Settings. Right-click `Project -> Properties -> C/C++ Build -> MCU Settings -> Select RW612 -> Apply & Close`



Properties for rdrw610_hello_world

MCU settings

Available parts

SDK MCUs

MCUs from installed SDKs. Please click above or visit mcuxpresso.nxp.com to obtain additional SDKs.

NXP RW610
- K32W061
- MIMXRT1060
- MIMXRT1170
- RW61X
  - RW610

Preinstalled MCUs

MCUs from preinstalled LPC and generic Cortex-M part support

Target
- CTNxxx
- LPC1102
- LPC112x
- LPC11Axx
- LPC11E6x
- LPC11Exx

Target architecture: cortex-m33-nodsp

☑ Preserve memory configuration
□ Preserve project configuration

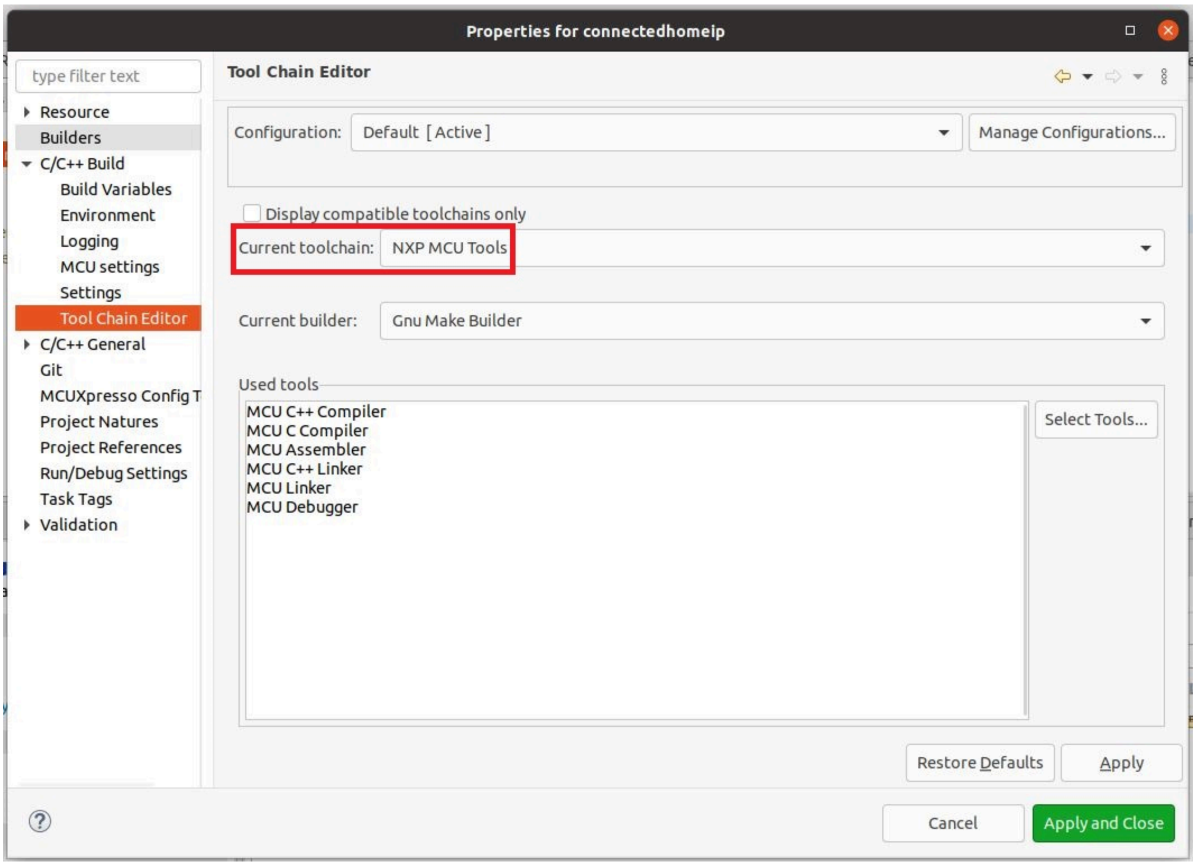
Memory details (RW610)*

Default LinkServer Flash Driver

Type	Name	Alias	Location	Size	Driver
Flash	QSPI_FLASH	Flash	0x8000000	0x800000	RW61X_FlexSP
RAM	SRAM	RAM	0x20000000	0x130000	

Figure 4. RW61X MCU setting view

4. Configure the toolchain editor. To do so, right-click `Project -> C/C++ Build-> Tool Chain Editor -> NXP MCU Tools -> Apply & Close`



Configuration: Default [Active]

□ Display compatible toolchains only

Current toolchain: NXP MCU Tools

Current builder: Gnu Make Builder

Used tools

- MCU C++ Compiler
- MCU C Compiler
- MCU Assembler
- MCU C++ Linker
- MCU Linker
- MCU Debugger

Figure 5. RW61X Tool Chain Editor view

5. Create a debug configuration. To do so, right-click `Project -> Debug -> As->SEGGER JLink probes -> OK -> Select elf file.`
6. Debug using the newly created configuration file.
   Note: *If an SDK package is used, duplicate the debug configuration from the SDK Hello World example after importing it.*
7. Debug using the newly created configuration file.

3.4 Start FRDM-RW612

1. Connect the FRDM-RW612 board to a PC using the USB Type-C cable.
2. Start a terminal emulator like PuTTY and connect to the used COM port with the following UART settings:
   - Baud rate: 115200
   - 8 data bits
   - 1 stop bit
   - No parity
   - No flow control
3. Open a terminal connection on the board and watch the printed logs.

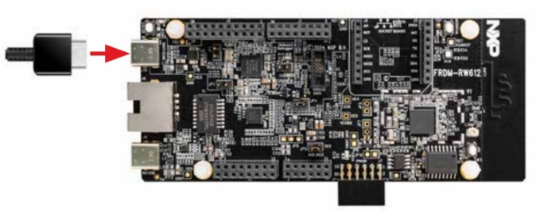


Figure 6. Connect the FRDM-RW612 board to a PC using the USB Type-C® cable

3.4.1 Testing the all-clusters application with Matter CLI enabled

The Matter CLI can be enabled with the all-clusters application.

For more information about the Matter CLI default commands, refer to the dedicated ReadMe.

The All-clusters application supports additional commands:

```
> help
[...]
mattercommissioning      Open/close the commissioning window.
                         Usage : mattercommissioning [on|off]
matterfactoryreset       Perform a factory reset on the device
matterreset              Reset the device
```

The `matterfactoryreset` command erases the file system completely (all Matter settings are erased).

The `matterreset` command enables the device to reboot without erasing the settings.

For detail OTBR setup steps, refer to Section 5.2

4 Thread Light/Switch Device – FRDM-MCXW71

The MCX W71x family features a 96 MHz Arm Cortex-M33 core coupled with a multiprotocol radio subsystem supporting Matter, Thread, Zigbee, and Bluetooth LE. The independent radio subsystem with a dedicated core and memory offloads the main CPU preserving it for the primary application and allowing firmware updates to support future wireless standards.

The MCX W71x family supports industrial and IoT devices as a single chip solution. NXP delivers a complete software solution to allow the MCX W71x to operate seamlessly as a network or radio coprocessor with NXP's broad portfolio of MCX MCUs, i.MX RT crossover MCUs, and i.MX applications processors.

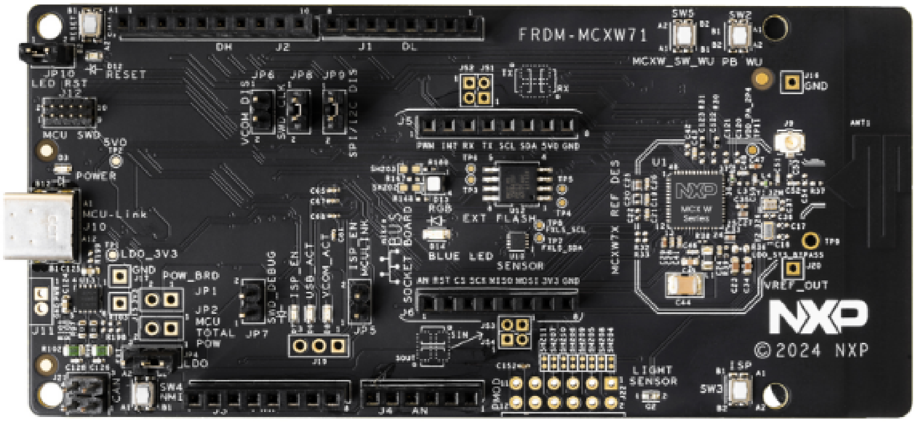


Figure 7. FRDM-MCXW71 board top view

Matter MCX W71 Light Switch Combo example provides a baseline demonstration of a dual-endpoint application. Endpoint =1 is used for the Light Device (bulb) and Endpoint = 2 is for the Light-Switch device. The light bulb is simulated using the onboard RGB LED. The bulb uses buttons to test turn on/turn off the local light bulb or the bonded lights. The example is based on Matter and the NXP MCX W71 SDK and supports remote access and control of a light bulb over a low-power, 802.15.4 Thread network. The example behaves as a Matter accessory that is a device that can be paired into an existing Matter network and can be controlled by this network.

Figure 8. FRDM-MCXW71 board top view

4.1 Build FRDM-MCXW71 Light Switch Combo Example Application

Matter supports configuring the build with GN, a fast and scalable meta-build system that generates inputs to ninja.

Note: *Commands are subject to change. See the "Readme" file for the branch you are trying to build (https://github.com/NXP/matter/tree/<default branch>/examples/light-switch-combo-app/nxp/mcxw71).*

1. Prerequisites
   Make sure that the below prerequisites are correctly installed with the following command:

```
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
```

   Use pip3 to install west:

```
pip3 install west
```

2. Checking out the Matter code

To check out the Matter repository with all platforms, run the following command:

```
git clone --recurse-submodules https://github.com/NXP/matter.git
```

If you already have the Matter code checked out, run the following commands to update the repository and synchronize submodules:

```
git pull
git submodule update --init
```

3. Check out NXP-specific submodules only

```
user@ubuntu:~/Desktop/git/matter$ scripts/checkout_submodules.py --shallow --platform nxp --
recursive
```

4. Activate the local environment

```
user@ubuntu:~/Desktop/git/matter $ source scripts/activate.sh
```

If the script says that the environment is out of date, update it by running the following command:

```
user@ubuntu:~/Desktop/git/matter $ source scripts/bootstrap.sh
```

5. Init NXP SDK(s)

```
 user@ubuntu:~/Desktop/git/matter $
third_party/nxp/nxp_matter_support/scripts/update_nxp_sdk.py --force
```

Note: *By default, `update_nxp_sdk.py` tries to initialize all NXP SDKs. Arg `-- help` can be used to view all available options.*

6. Start building the application

```
 user@ubuntu:~/Desktop/git/matter$ cd examples/light-switch-combo-app/nxp/
mcxw71
 user@ubuntu:~/Desktop/git/matter/examples/light-switch-combo-app/nxp/
mcxw71$
 gn gen out/debug --args="chip_with_ot_cli=0 is_debug=false
 chip_openthread_ftd=true chip_crypto=\"platform\""
 user@ubuntu:~/Desktop/git/matter/examples/light-switch-combo-app/nxp/mcxw71$
 ninja -C out/debug
```

In case Openthread CLI is needed, the `chip_with_ot_cli build` argument must be set to 1.
After a successful build, the elf and srec files are found in `out/debug/`. See the files prefixed with `chip-mcxw71-light-switch-combo-example.`

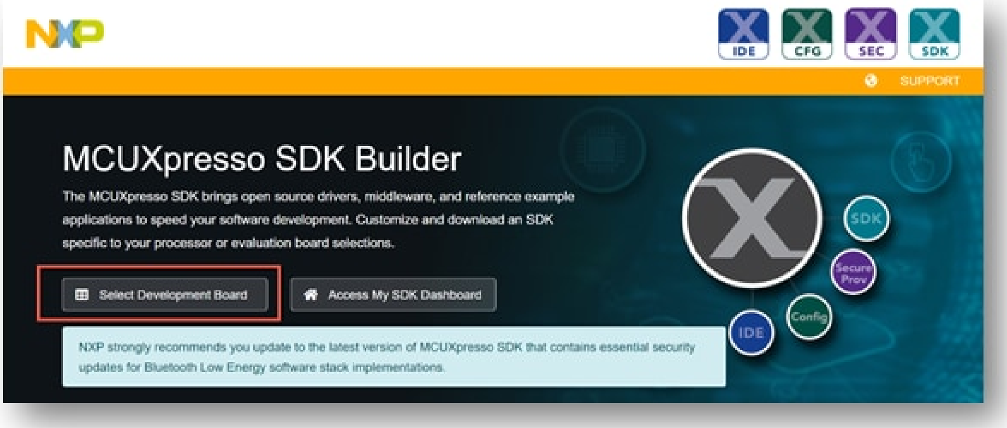
4.2 Flashing NBU image and host image

Two images must be written on the board: one for the host (CM33) and one for the NBU (CM3).

The image needed on the host side is the one generated in `out/debug/` while the one needed on the NBU side can be found in the downloaded NXP-SDK package at path - `middleware\wireless\ieee-802.15.4\bin\ncu71\ncu71_nbu_ble_15_4_dyn_matter_$version.sb3.`

4.2.1 Download the sb3 file

1. Start your design with the MCUXpresso SDK
   The MCUXpresso SDK is complimentary and includes full source code under a permissive open source license for all hardware abstraction and peripheral driver software.



MCUXpresso SDK Builder

The MCUXpresso SDK brings open source drivers, middleware, and reference example applications to speed your software development. Customize and download an SDK specific to your processor or evaluation board selections.

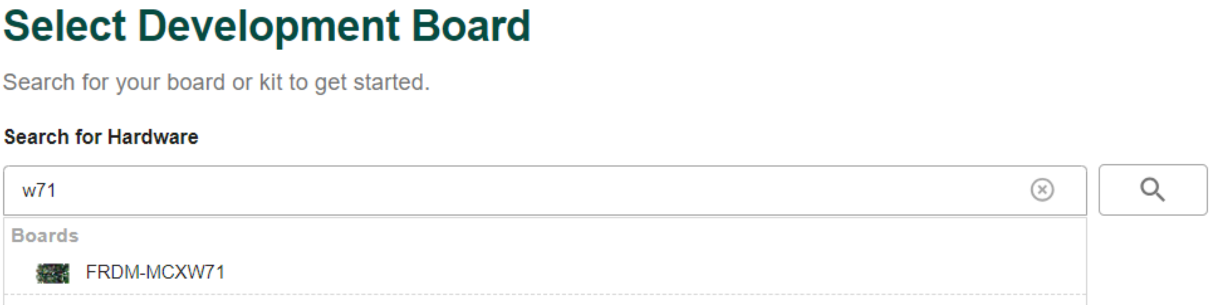
Select Development Board

Access My SDK Dashboard

NXP strongly recommends you update to the latest version of MCUXpresso SDK that contains essential security updates for Bluetooth Low Energy software stack implementations.

Figure 9. MCUXpresso SDK

2. In the **Search for Hardware** search box, type in the selected board “FRDM-MCXW71”. Click the board to select it.



Select Development Board

Search for your board or kit to get started.

Search for Hardware

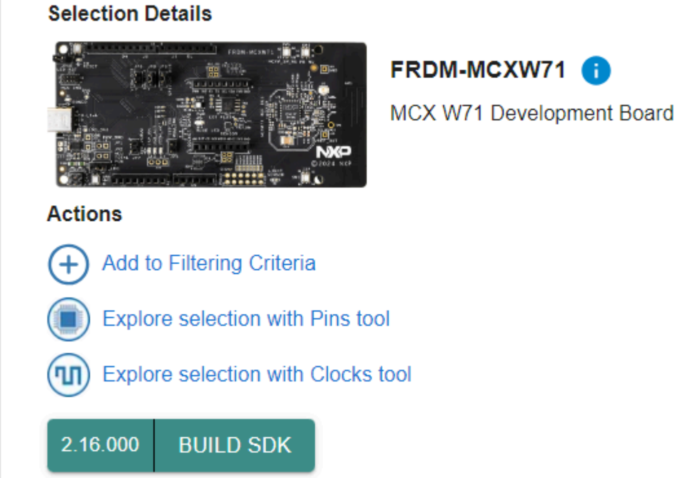
w71

Boards

FRDM-MCXW71

Figure 10. Selected board

3. On the right-hand side, click the option to build the SDK for the K32W148-EVK to add the middleware needed.



Selection Details

FRDM-MCXW71

MCX W71 Development Board

Actions

Add to Filtering Criteria

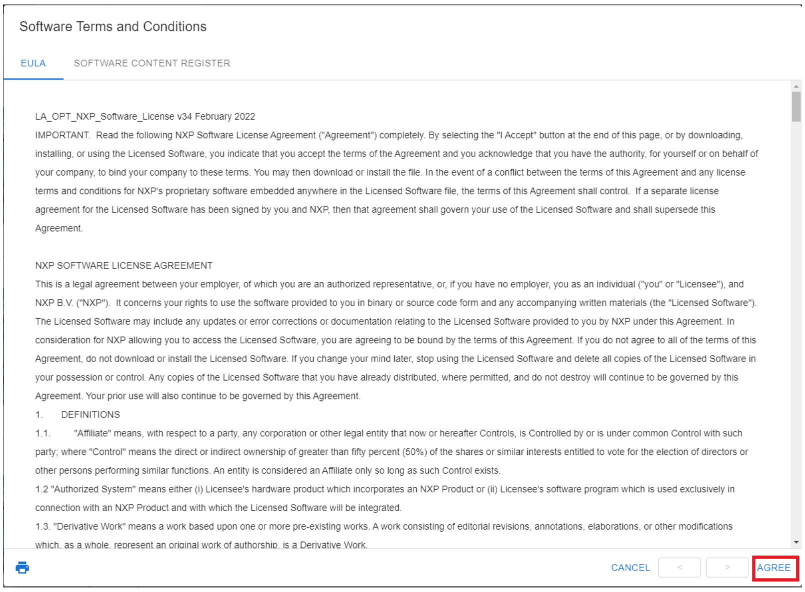
Explore selection with Pins tool

Explore selection with Clocks tool

2.16.000 BUILD SDK

Figure 11. Build W71 SDK

4. To build the SDK, Select All for the middleware available. Then scroll down and click **Download SDK**. The Software Terms and Conditions appear, select **I Agree** to begin the download.



LA_OPT_NXP_Software_License v34 February 2022

IMPORTANT. Read the following NXP Software License Agreement ("Agreement") completely. By selecting the "I Accept" button at the end of this page, or by downloading, installing, or using the Licensed Software, you indicate that you accept the terms of the Agreement and you acknowledge that you have the authority, for yourself or on behalf of your company, to bind your company to these terms. You may then download or install the file. In the event of a conflict between the terms of this Agreement and any license terms and conditions for NXP's proprietary software embedded anywhere in the Licensed Software file, the terms of this Agreement shall control. If a separate license agreement for the Licensed Software has been signed by you and NXP, then that agreement shall govern your use of the Licensed Software and shall supersede this Agreement.

NXP SOFTWARE LICENSE AGREEMENT

This is a legal agreement between your employer, of which you are an authorized representative, or, if you have no employer, you as an individual ("you" or "Licensee"), and NXP B.V. ("NXP"). It concerns your rights to use the software provided to you in binary or source code form and any accompanying written materials (the "Licensed Software"). The Licensed Software may include any updates or error corrections or documentation relating to the Licensed Software provided to you by NXP under this Agreement. In consideration for NXP allowing you to access the Licensed Software, you are agreeing to be bound by the terms of this Agreement. If you do not agree to all of the terms of this Agreement, do not download or install the Licensed Software. If you change your mind later, stop using the Licensed Software and delete all copies of the Licensed Software in your possession or control. Any copies of the Licensed Software that you have already distributed, where permitted, and do not destroy will continue to be governed by this Agreement. Your prior use will also continue to be governed by this Agreement.

1. DEFINITIONS

1.1. "Affiliate" means, with respect to a party, any corporation or other legal entity that now or hereafter Controls, is Controlled by or is under common Control with such party; where "Control" means the direct or indirect ownership of greater than fifty percent (50%) of the shares or similar interests entitled to vote for the election of directors or other persons performing similar functions. An entity is considered an Affiliate only so long as such Control exists.

1.2 "Authorized System" means either (i) Licensee's hardware product which incorporates an NXP Product or (ii) Licensee's software program which is used exclusively in connection with an NXP Product and with which the Licensed Software will be integrated.

1.3. "Derivative Work" means a work based upon one or more pre-existing works. A work consisting of editorial revisions, annotations, elaborations, or other modifications which, as a whole, represent an original work of authorship, is a Derivative Work.

Figure 12. Agree with caption

5. Once the SDK package is downloaded, the sb3 file can be found in the downloaded NXP-SDK package at path `middleware\wireless\ieee-802.15.4\bin\mcu71\mcu71_nbu_ble_15_4_dyn_matter_$version.sb3`.

4.2.2 Flashing the NBU image

Install the MCUXpresso SDK directly from the MCUXpresso SDK website at mcuxpresso.nxp.com. Click **Select Development Board** to search for the evaluation board.

The NBU image should be written only when a new NXP-SDK is released.

K32W148 board quick start guide can be used for updating the NBU/radio core:

Section 3.3 Updating NBU for Wireless examples - use the corresponding .sb3 file found in the SDK package at path `middleware\wireless\ieee-802.15.4\bin\k32w1\`.

4.2.3 Flashing the host image

The Host image is the one found under `out/debug/`. It should be written after each build process.

Flashing is needed, then JLink can be used:

- Plug MCXW71 to the USB port (no need to keep the SW4 button pressed while doing this)
- Create a file `commands_script` with the following content (change the application name accordingly):

```
reset
halt
loadfile chip-mcxw71-light-example.srec
reset
go
quit
```

- Copy the application and `commands_script` in the same folder that the JLink executable is placed. Execute:

```
$  jlink -device MCXW71 -if SWD -speed 4000 -autoconnect 1 -CommanderScript
```

4.3 Start FRDM-MCUW71

1. Connect the FRDM-MCXW71 board to a PC using the USB-Type C cable.
2. Start a terminal emulator like PuTTY and connect to the used COM port with the following UART settings:
   - Baud rate: 115200
   - 8 data bits
   - 1 stop bit
   - No parity
   - No flow control
3. Open a terminal connection on the board and watch the printed logs.

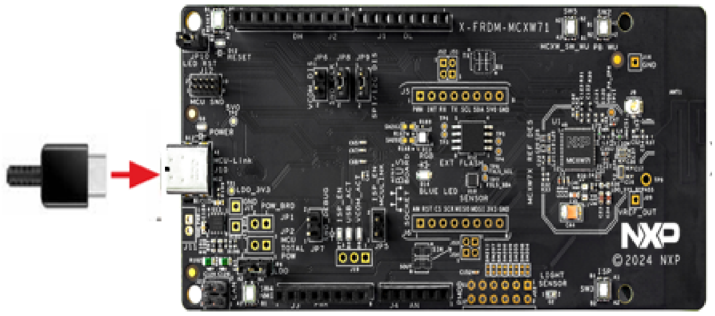


Figure 13. Connect the FRDM-MCXW71board to a PC using the USB-Type C cable

4.4 Device UI

The Thread Light/Switch device example of the FRDM-MCXW71 board provides a simple UI that depicts the state of the device and offers basic user control. This UI is implemented via the general-purpose LEDs and buttons built in the FRDM-MCXW71 board.

LED 2 shows the overall state of the device and its connectivity. Four states are depicted:

- Short Flash On (50 ms on/950ms off) — The device is in an unprovisioned (unpaired) state and is waiting for a commissioning application to connect.
- Rapid Even Flashing (100 ms on/100ms off) — The device is in an unprovisioned state and a commissioning application is connected via Bluetooth LE.
- Short Flash Off (950 ms on/50ms off) — The device is fully provisioned but does not yet have full network (Thread) or service connectivity.
- Solid On — The device is fully provisioned and has full network and service connectivity.

Note: *LED2 is disabled when CHIP_DEVICE_CONFIG_ENABLE_OTA_REQUESTOR is enabled. On the FRDM-MCXW71 board, PTB0 is wired to LED2 and is wired to CS (chip select) External flash memory. OTA image is stored in external memory because of its size. If LED2 is enabled, it affects External Memory CS and OTA will not work.*

RGB LED shows the state of the simulated light bulb. When the LED is lit, the light bulb is on; when not lit, the light bulb is off.

Button SW2 can be used to start Bluetooth LE advertising.

- A SHORT press enables Bluetooth LE advertising for a predefined time.
- A LONG press (hold for 3 seconds) initiates a factory reset.

After an initial period of 3 seconds, LED 2 and RGB LED flash in unison to signal the pending reset. After 6 seconds, the device resets its persistent configuration and initiated a reboot. The reset action can be canceled by pressing the SW2 button at any point before the 6-second window.

Button SW4 can be used to toggle a binded light device.

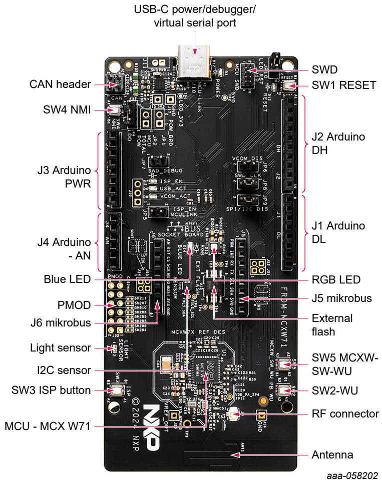


Figure 14. W71 Device UI

5 Demo setup with NXP Android Matter controller

Note: *To maximize operational success rates, all devices must be equipped with an antenna connection. This setup enhances signal stability and overall system reliability.*

In this section, a Matter demo is executed using an NXP Matter controller app. The app supports Matter 1.3 and most used device types.

Supported device types in the current release:

- Light
- Light switch
- Door/Window sensor
- Thermostat
- Motion Sensor
- Temperature Sensor
- Speaker
- Door Lock
- Cook Surface
- Cooktop
- EVSE

Below are frequently used features supported by the application.

- **Pair** and **control** end devices with the controller.
- **Binding/Unbinding** with the other device (FRDM-RW612 and FRDM-MCXW71).
- **Share** Matter-compatible smart home devices between different ecosystems. (multiadministration).
- **Unpair** end devices in the app.

Figure 15 shows the setup topology where the cell phone acts as the Matter controller. The FRDM-RW612 board acts as a light, switch, and OTBR combo device, and FRDM-MCXW71 as a light and switch combo Matter.

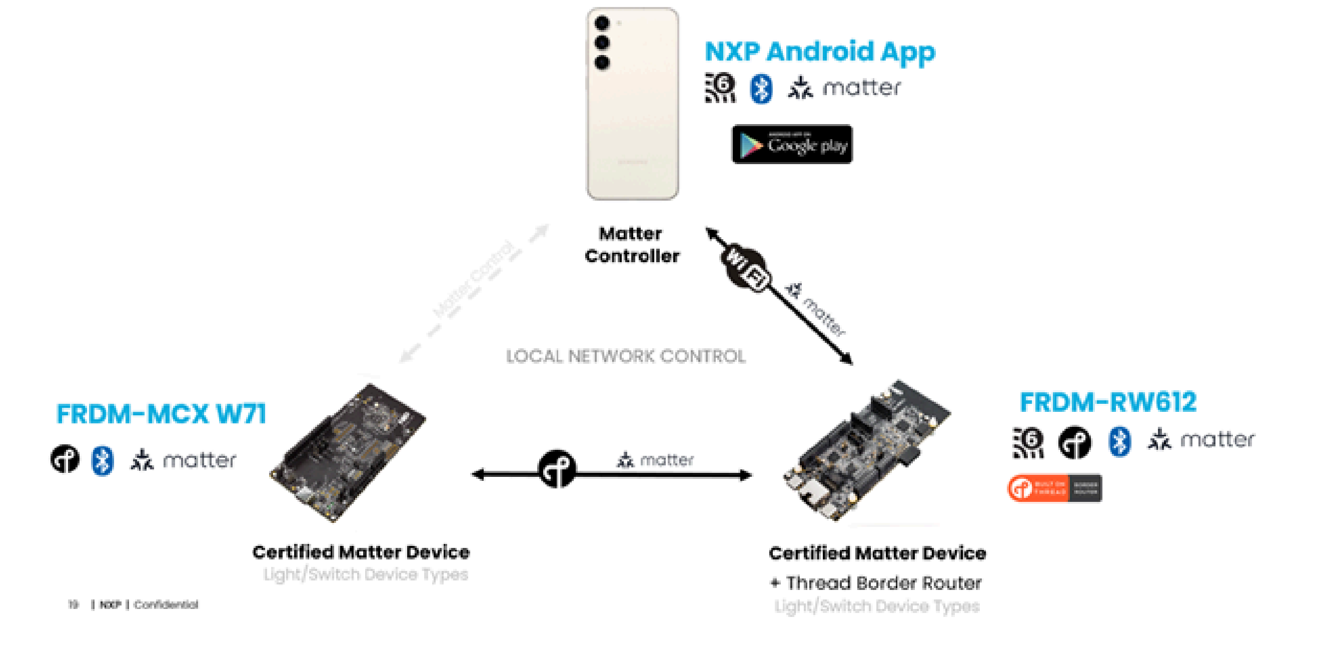


Figure 15. Matter applications setup topology

Install the NXP Matter chip-tool application.

- Using the Android phone, go to the Google Play Store
- Search "**NXP Matter Chip-tool**" and select the app file for install as shown in Figure 16
- Complete the installation as shown in Figure 16

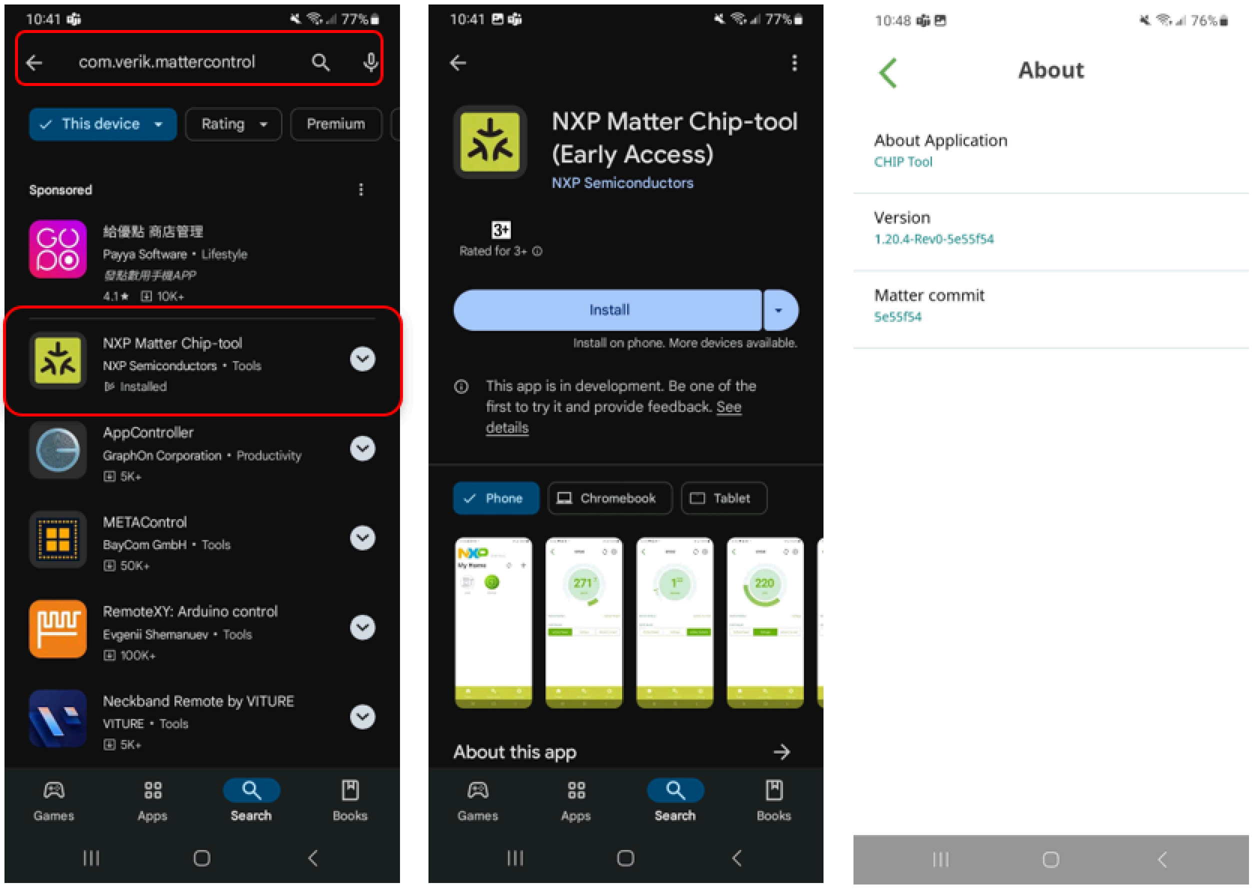
Wi-Fi Prerequisites

The Android phone and RW612-FRDM must be connected to an Access Point and able to reach each other via its IPv6 address.

Configure the AP as shown below.

- If the AP supports the IPv6 DHCP server, it must be disabled.
- WPA3 is not supported in the Matter demo binary for RW612. The AP must be configured in WPA2 mode.

For Wi-Fi connection details, refer to Section 6.1.



This app is in development. Be one of the first to try it and provide feedback. See details

About Application
CHIP Tool

Version
1.20.4-Rev0-5e55f54

Matter commit
5e55f54

Figure 16. Install NXP Matter chip-tool application

5.1 Commissioning FRDM-RW612

Follow the below steps to commission the FRDM-RW612 board to the AP.

1. Click Pair New Device as shown in Figure 17

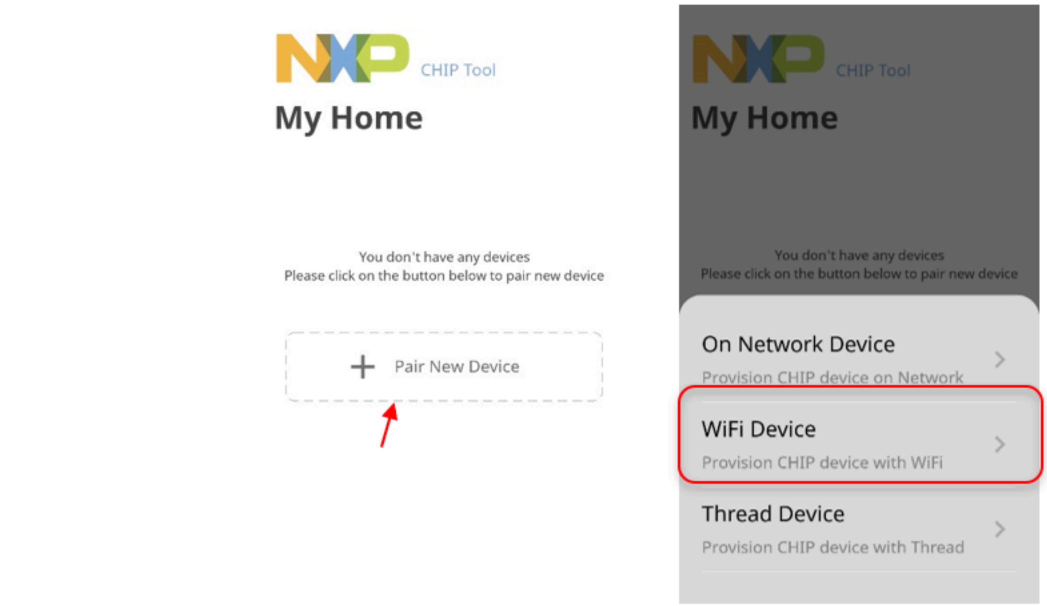
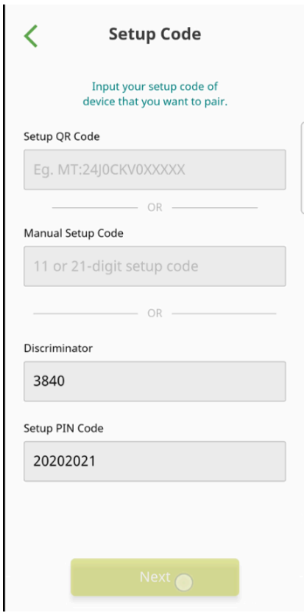

CHIP Tool
My Home
You don't have any devices
Please click on the button below to pair new device
Pair New Device
CHIP Tool
My Home
You don't have any devices
Please click on the button below to pair new device
On Network Device
Provision CHIP device on Network
WiFi Device
Provision CHIP device with WiFi
Thread Device
Provision CHIP device with Thread

Figure 17. Start Matter chip-tool app

2. Enter the discriminator and PIN (the default ones are 3840 and 20202021 respectively) on the below page as shown in Figure 18.

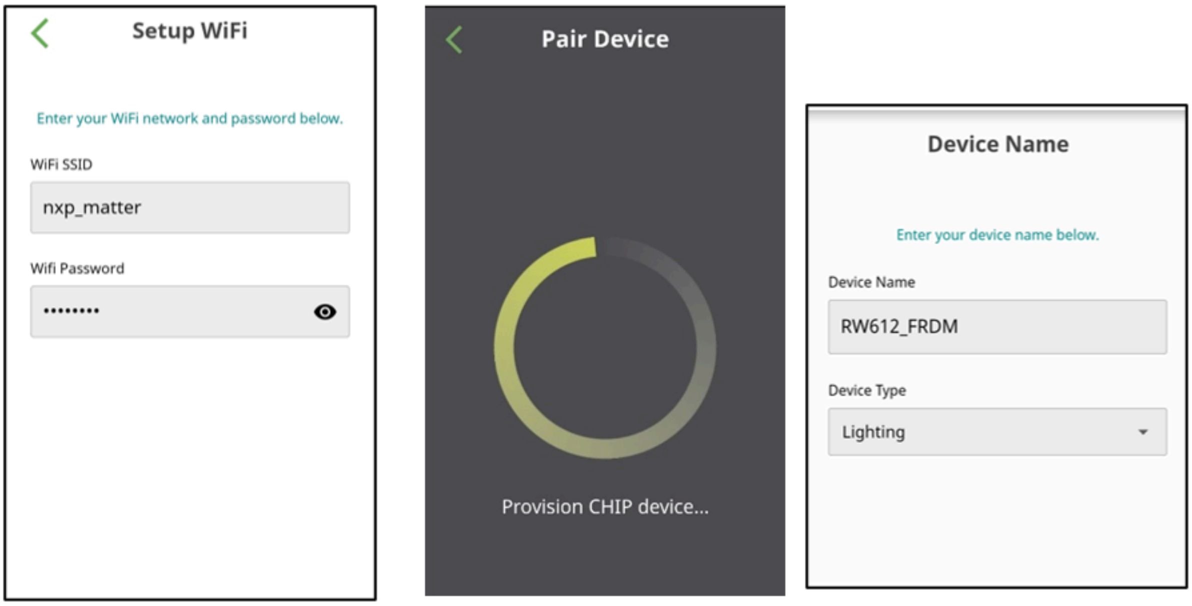

Setup Code
Input your setup code of device that you want to pair.
Setup QR Code
Eg. MT:24J0CKV0XXXXX
OR
Manual Setup Code
11 or 21-digit setup code
OR
Next

Discriminator
3840

Setup PIN Code
20202021

Figure 18. Enter the discriminator and PIN code of FRDM-RW612

3. Enter your Wi-Fi SSID and password. After pairing is successful, enter the device name as shown on Figure 19.



Setup WiFi

Enter your WiFi network and password below.

WiFi SSID: nxp_matter

Wifi Password: ••••••••

Device Name

Enter your device name below.

Device Name: RW612_FRDM

Device Type: Lighting

Figure 19. Start Matter chip-tool app

Note: *If there is still an error, wait a moment and repeat the step.*

4. The device is now successfully paired and the switch can be controlled. For details, see Figure 20

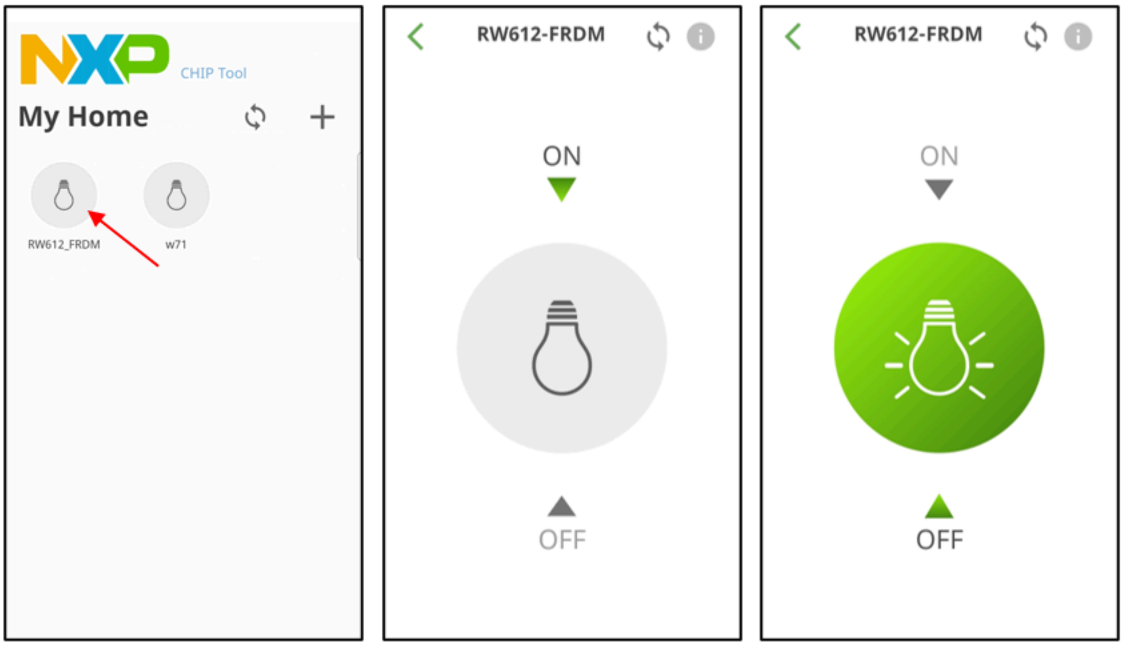
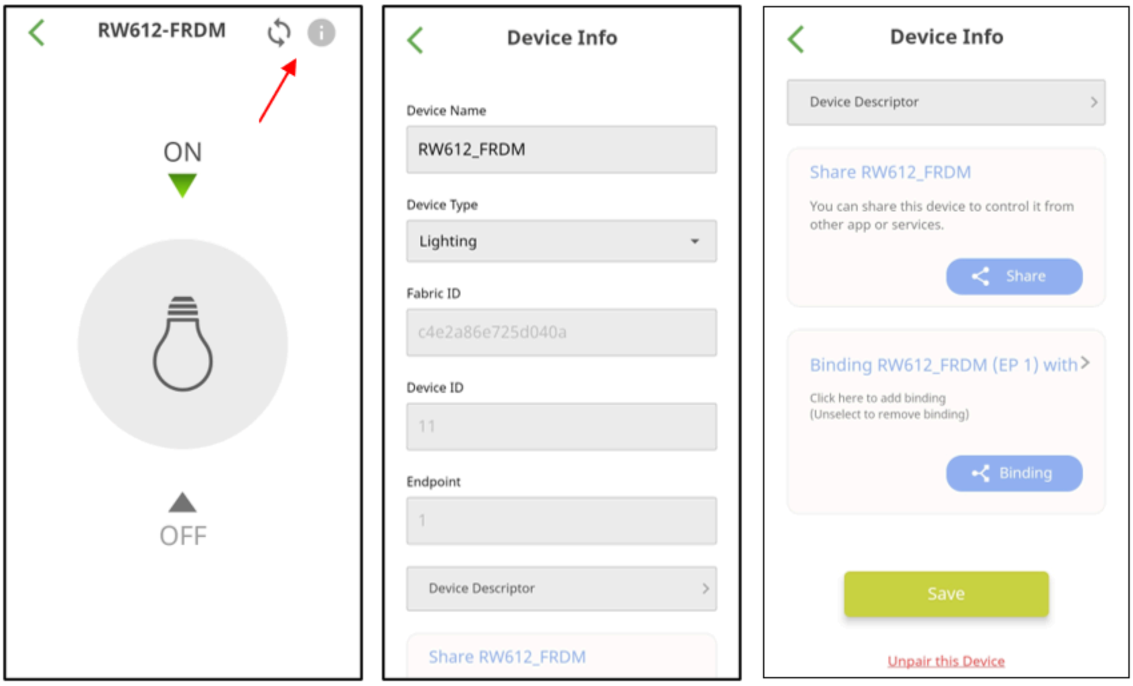


Figure 20. RW612 control screen (EP#1)

The detailed RW612 device information is shown on Figure 21



Device Name: RW612_FRDM

Device Type: Lighting

Fabric ID: c4e2a86e725d040a

Device ID: 11

Endpoint: 1

Share RW612_FRDM

You can share this device to control it from other app or services.

Binding RW612_FRDM (EP 1) with

Click here to add binding
(Unselect to remove binding)

Figure 21. RW612 device information screen

5.2 Setup OTBR on FRDM-RW612

After successfully commissioning FRDM-RW612, use the following otcli command to start the OTBR operation. For details, see Section 3.4

Start RW612-FRDM opens the com port terminal

```
>ble adv stop
Done
> otcli dataset init new
Done
>otcli dataset
Done
>otcli dataset commit active
Done
>otcli ifconfig up
Done
>otcli thread start
Done
>otcli state
Done
[It should be leader after a few seconds…,
need to wait a moment and double check with the same command]
 >otcli dataset active
[use for check the OTBR state]
```

Get the OTBR table list as hown in the following example:

```
>otcli dataset active
Active Timestamp: 1
Channel: 17
```

```
Channel Mask: 0x07fff800
Ext PAN ID: 1111111122222222
Mesh Local Prefix: fdca:a943:242e:b986::/64
Network Key: 00112233445566778899aabbccddeeff
Network Name: OpenThread-b04c
PAN ID: 0x1234
PSKc: 8d4874b9ac7e0ad605179fc71af9a1c8
Security Policy: 672 onrc 0
Done
```

5.3 Commissioning FRDM-MCXW71

1. Add a device, Pair New Device, select Thread Device as shown in Figure 22.

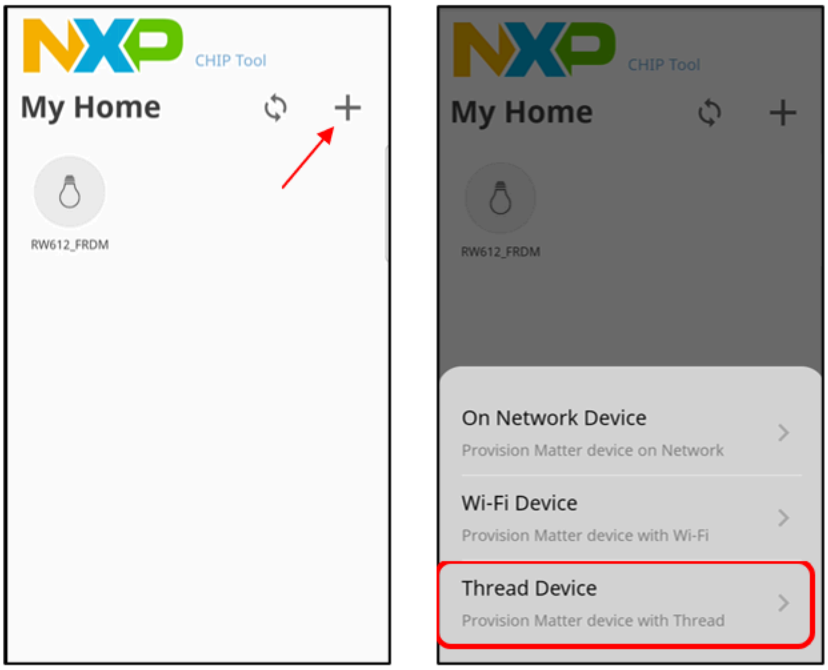
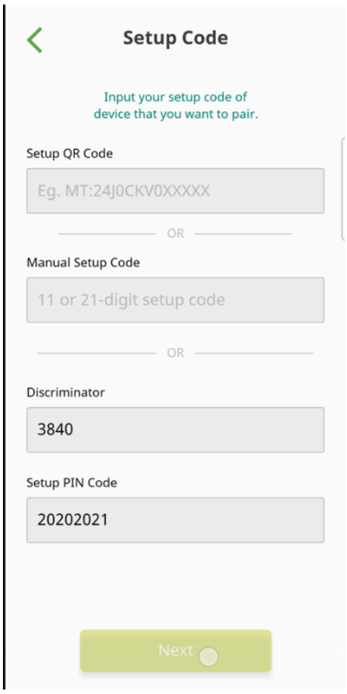

CHIP Tool
My Home
RW612_FRDM
CHIP Tool
My Home
RW612_FRDM
On Network Device
Provision Matter device on Network
Wi-Fi Device
Provision Matter device with Wi-Fi
Thread Device
Provision Matter device with Thread

Figure 22. Add thread device screen

2. Enter the discriminator and PIN code (the default ones are 3840 and 20202021 respectively) to commission FRDM-MCXW71 as shown in Figure 23



Setup Code

Input your setup code of device that you want to pair.

Setup QR Code

Eg. MT:24J0CKV0XXXXX

OR

Manual Setup Code

11 or 21-digit setup code

OR

Discriminator

3840

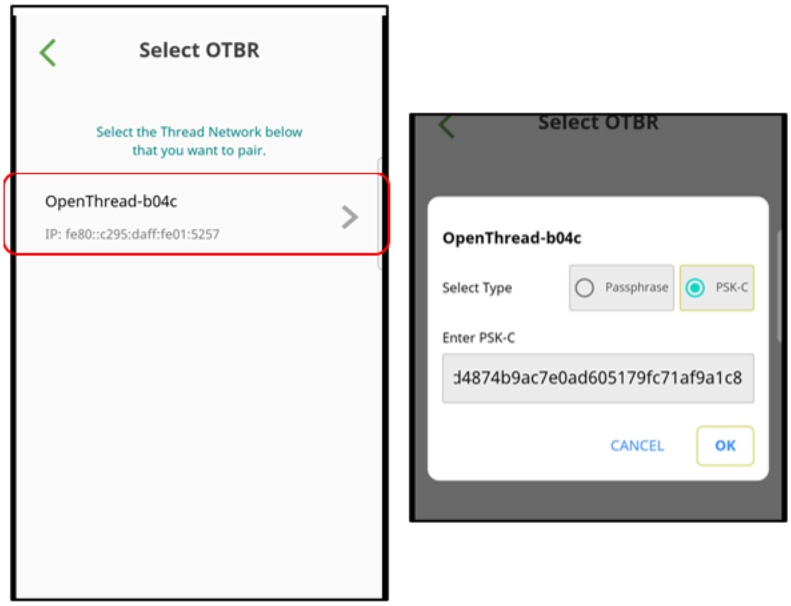
Setup PIN Code

20202021

Next

Figure 23. Enter the discriminator and PIN code of W71

3. **Select OTBR (Open Thread Border Router) network information on the Android tool**: As shown in Figure 24, select your network name and the correct key type (PSK-C), enter the PSK-C:<psck key>.
   Note: *That OTBR discovery can take a few minutes. One to two minutes is normal.*



Select OTBR

Select the Thread Network below that you want to pair.

OpenThread-b04c

IP: fe80::c295:daff:fe01:5257

Select OTBR

OpenThread-b04c

Select Type ○ Passphrase ● PSK-C

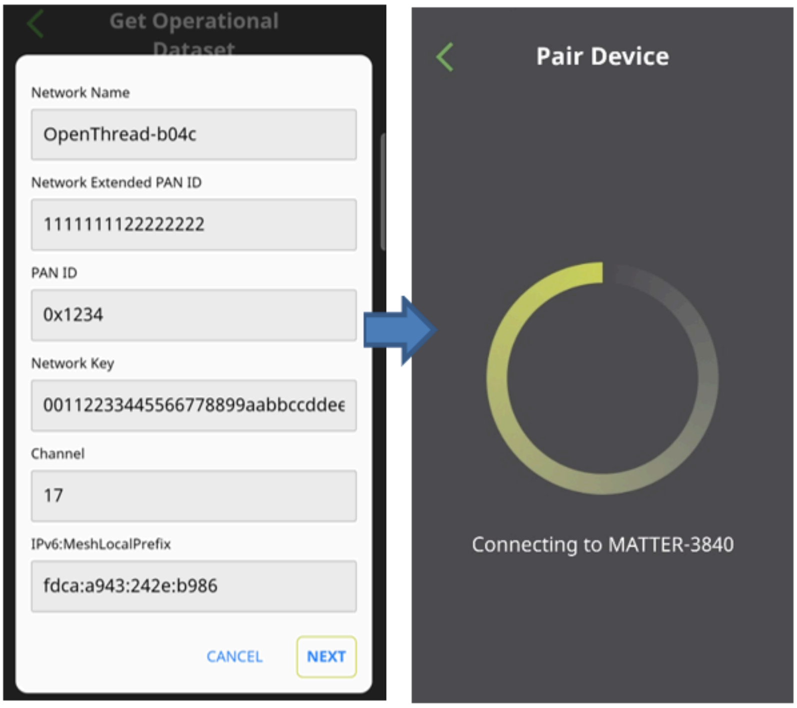
Enter PSK-C

d4874b9ac7e0ad605179fc71af9a1c8

CANCEL OK

Figure 24. Select OTBR and enter the key value screen

4. Wait for the new prompt screen and then press the SW2 button on the W71 board to start the Bluetooth LE advertising. Press the NEXT button, as shown in Figure 25.



Get Operational Dataset

Network Name: OpenThread-b04c

Network Extended PAN ID: 1111111122222222

PAN ID: 0x1234

Network Key: 00112233445566778899aabbccddee

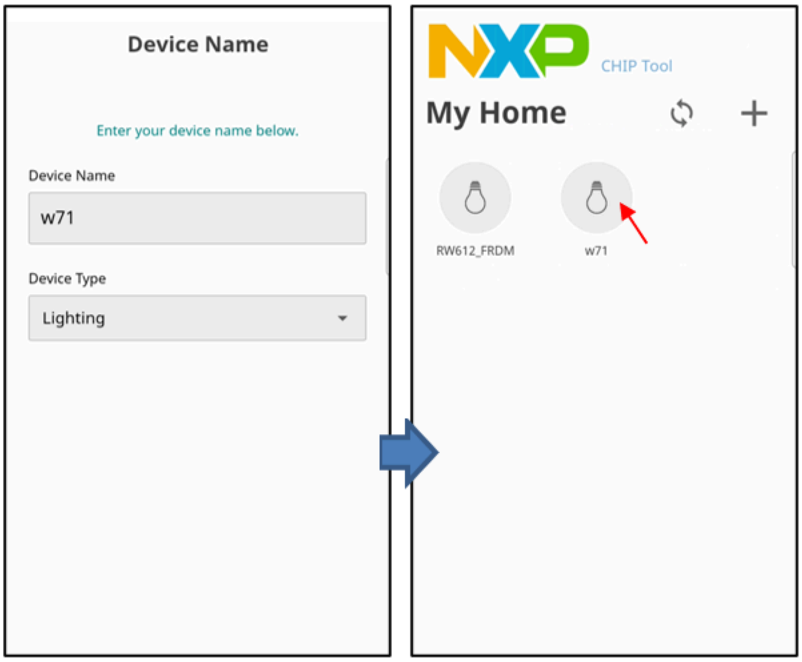
Channel: 17

IPv6:MeshLocalPrefix: fdca:a943:242e:b986

Connecting to MATTER-3840

Figure 25. OTBR network information screen and pairing screen

5. After entering the device name, you can add the devices as shown in Figure 26.



Device Name

Enter your device name below.

Device Name: w71

Device Type: Lighting

Figure 26. Add K32W148(EP#1) device success screen

6. With the use FRDM-MCXW71 switch, click the information button in the upper right corner as shown in Figure 27 to see the FRDM-MCXW71 device information (Figure 28).

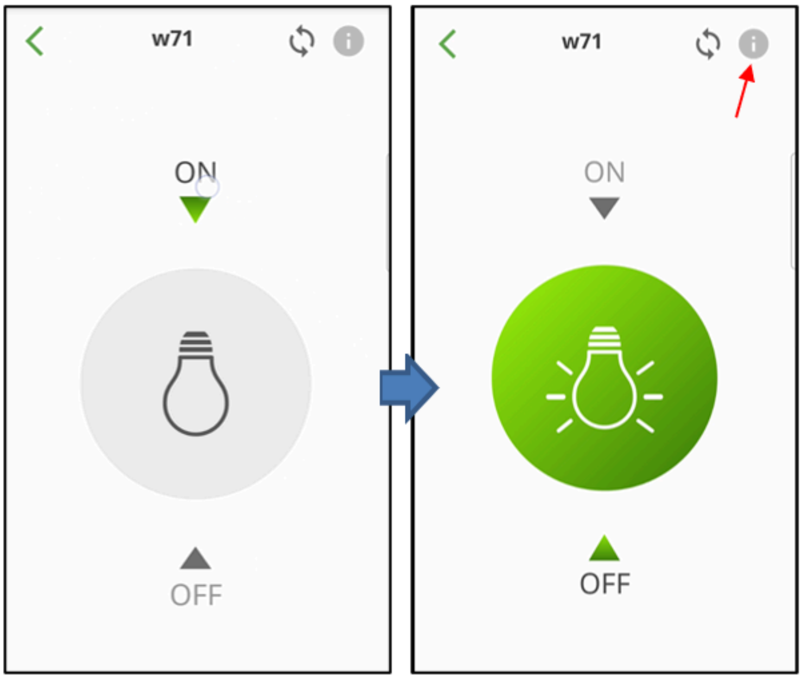
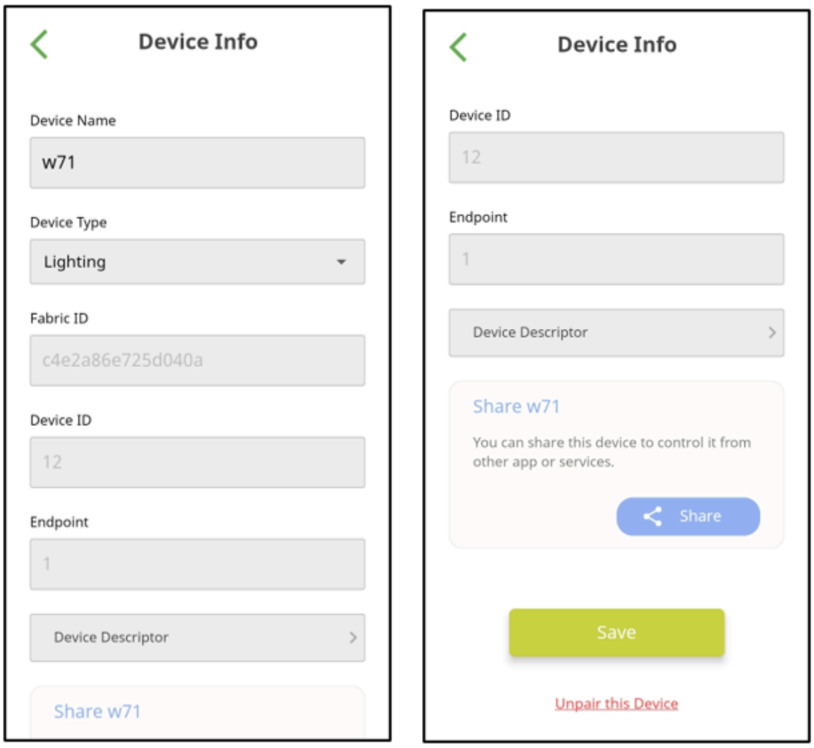


Figure 27. FRDM-MCXW71 control light screen


Device Info

Device Name

w71

Device Type

Lighting

Fabric ID

c4e2a86e725d040a

Device ID

12

Endpoint

1

Device Descriptor

Share w71

Device Info

Device ID

12

Endpoint

1

Device Descriptor

Share w71

You can share this device to control it from other app or services.

Share

Save

Unpair this Device

Figure 28. FRDM-MCXW71 information screen

5.4 Device binding

- Bind MCU W71 from RW612-FRDM
  1. Enter the information page of RW612-FRDM (**EP2**), select the binding device MCU W71 FRDM (**EP1**) as shown in Figure 29

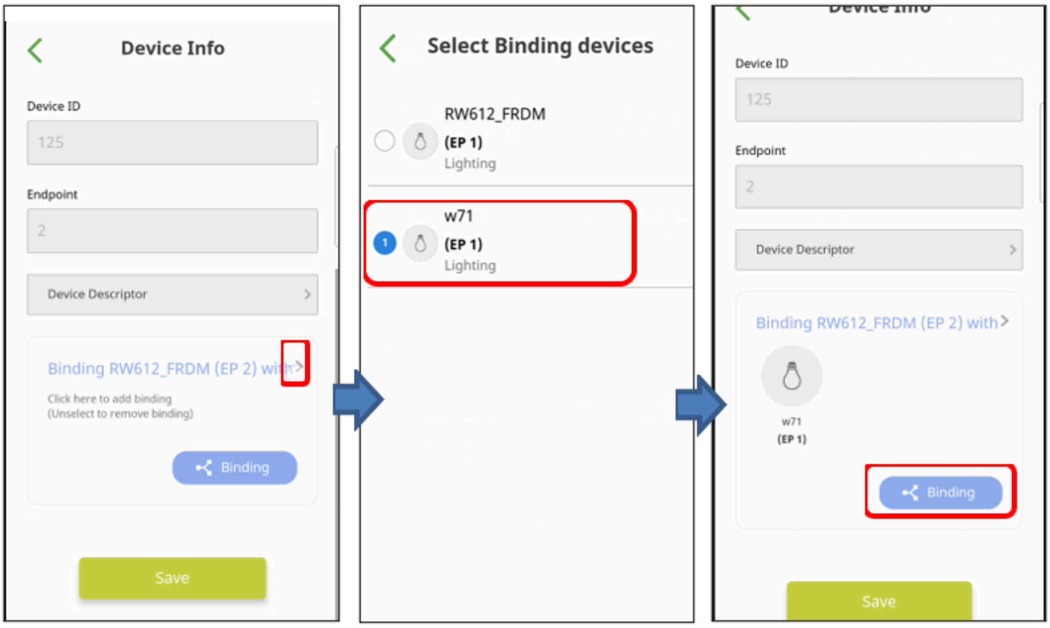
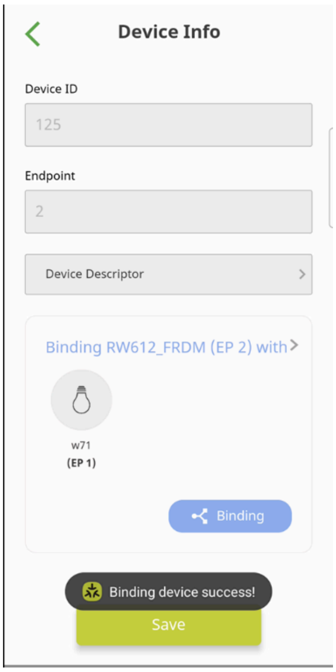


Figure 29. Binding RW612-FRDM with MCU W71 FRDM screen

  2. A prompt message is displayed after the pairing is successful. When the binding is successful, click **SW2** of the RW612-FRDM board and check that it controls the **RGB LED** of MCU W71 FRDM.



Device Info

Device ID: 125

Endpoint: 2

Binding RW612_FRDM (EP 2) with

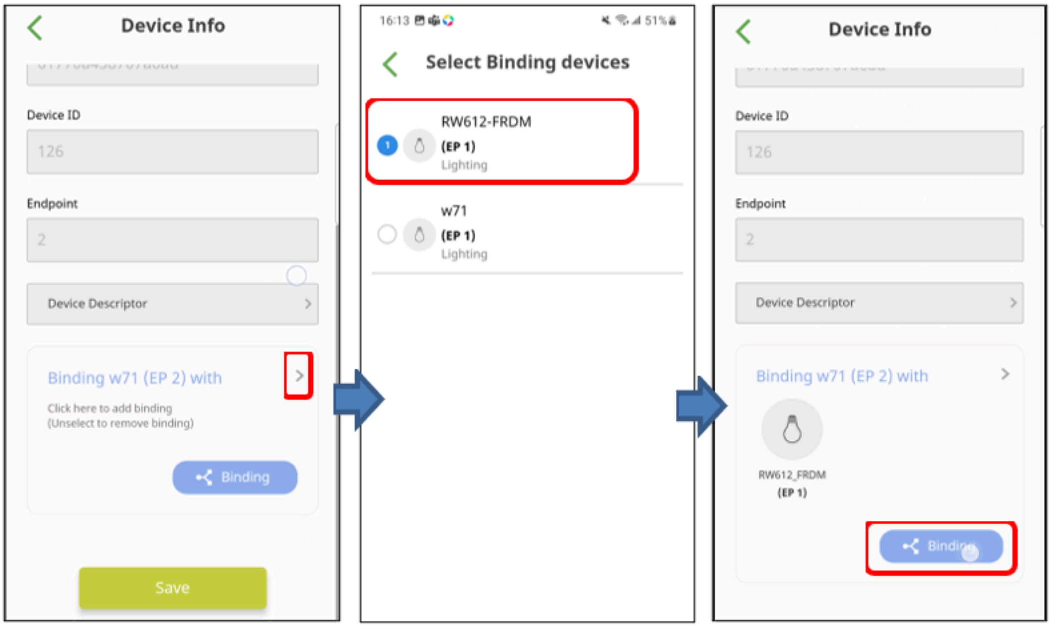
w71 (EP 1)

Binding device success!

Figure 30. Binding MCU W71 FRDM success screen

- Binding RW612-FRDM from MCU W71FRDM

1. Enter the information page of MCU W71 FRDM (**EP2**) and select binding device RW612- FRDM (**EP1**) as shown in Figure 31



Device Info

Device ID: 126

Endpoint: 2

Binding w71 (EP 2) with

Click here to add binding
(Unselect to remove binding)

16:13 51%

Select Binding devices

● RW612-FRDM (EP 1) Lighting

○ w71 (EP 1) Lighting

Device Info

Device ID: 126

Endpoint: 2

Binding w71 (EP 2) with

RW612_FRDM (EP 1)

Figure 31. Binding MCU W71 FRDM with RW612-FRDM screen

2. A prompt message is displayed after the pairing is successful. When the binding is successful, click **SW4** of the RW612-MCU W71 FRDM board and check that it controls **RGB LED** of RW612-FRDM.

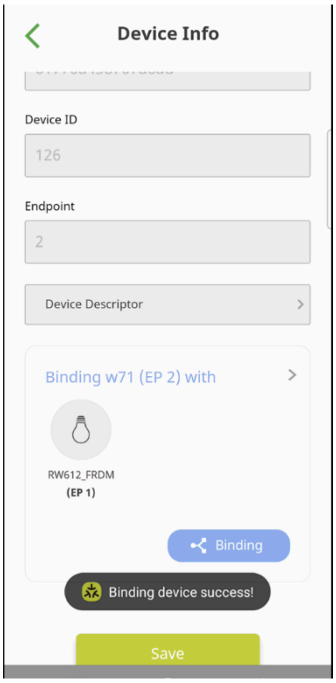


Figure 32. Binding RW612-FRDM success screen

5.5 Device unbinding

To unbind a device, follow the steps below.

1. Enter the information page of RW612-FRDM (EP2), select the unbinding device MCU W71 FRDM (**EP1**) and click the **Binding** button again.

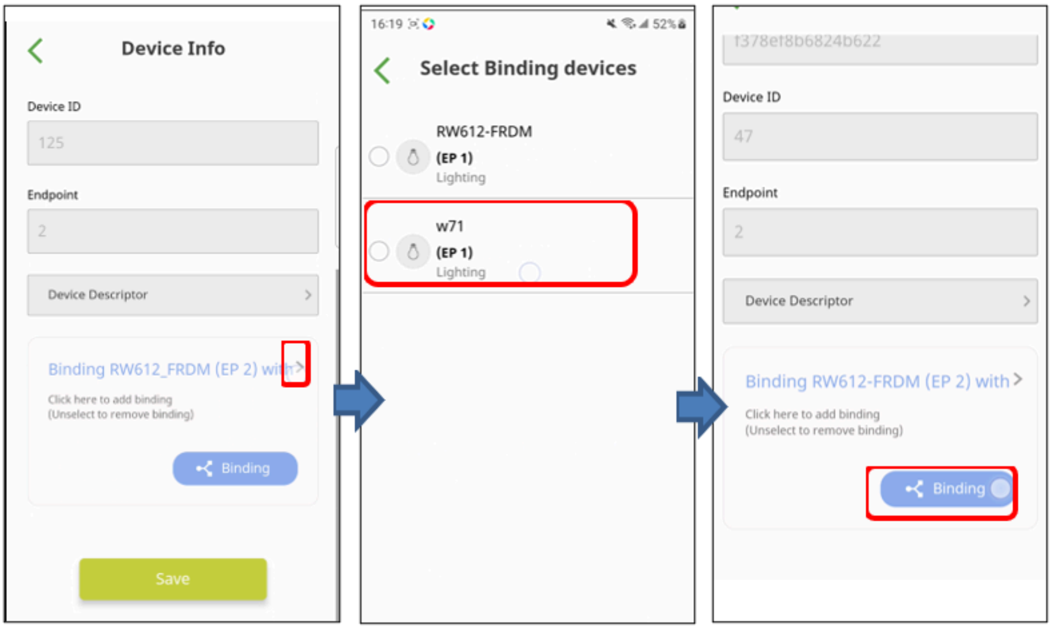
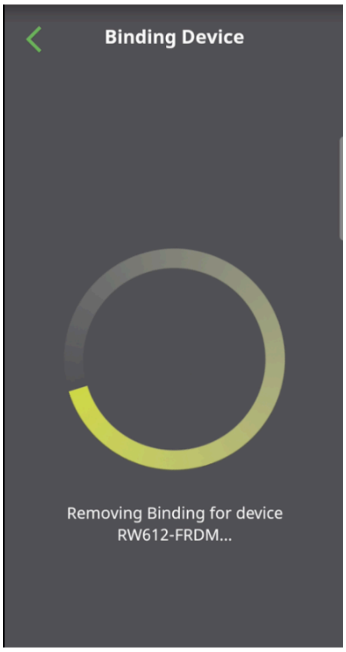


Figure 33. Unbinding RW612-FRDM with MCU W71 FRDM screen

2. A prompt message "removing binding for device" is displayed, then the device is successfully unbinded.



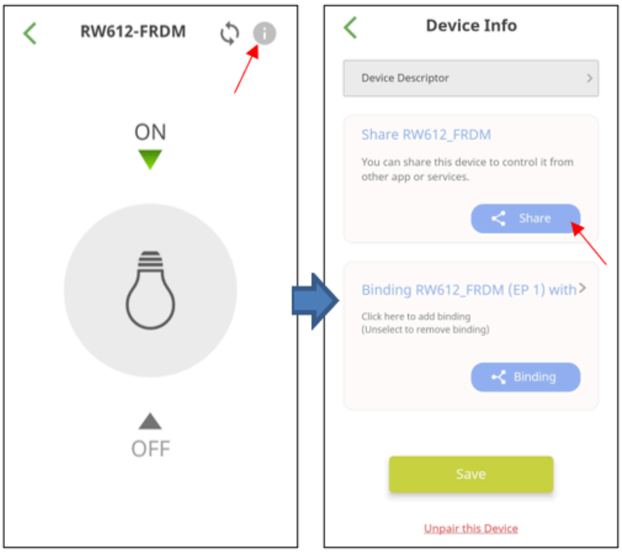
Removing Binding for device RW612-FRDM...

Figure 34. QR code for remove binding device screen

5.6 Multi-admin

Multi-admin is one of the core features of Matter as it provides simultaneous operation in multiple smart home systems. So, family or household members can operate the device with the digital assistant or smartphone application of their choice.

1. Enter the device information page (**EP 1**), and go to the screen bottom, click **Share**.

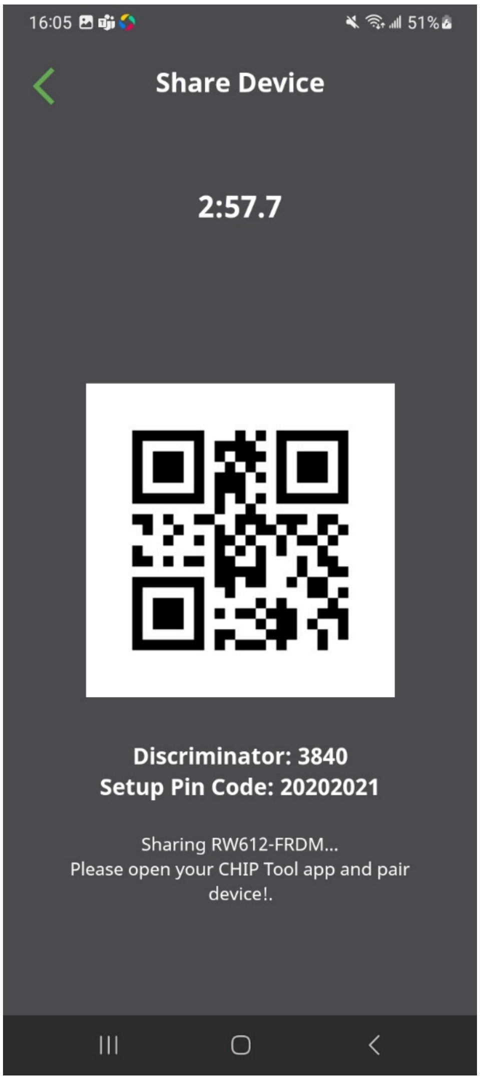


You can share this device to control it from other app or services.

Click here to add binding
(Unselect to remove binding)

Figure 35. Unpair device flow screen

2. If the QR code is displayed, the device sharing was successful. Users can use other digital assistants or a smartphone application to commission it.



Discriminator: 3840
Setup Pin Code: 20202021

Sharing RW612-FRDM...
Please open your CHIP Tool app and pair device!.

Figure 36. QR code for sharing device screen

5.7 Unpairing

To unpair devices, factory-reset your devices first. For details on the factory-reset devices skill and for a complete reset process, see Section 6.2 and Section 6.3 . To unpair a device, enter the device information page (**EP 1**) and click **Unpair this Device** as shown in Figure 37.

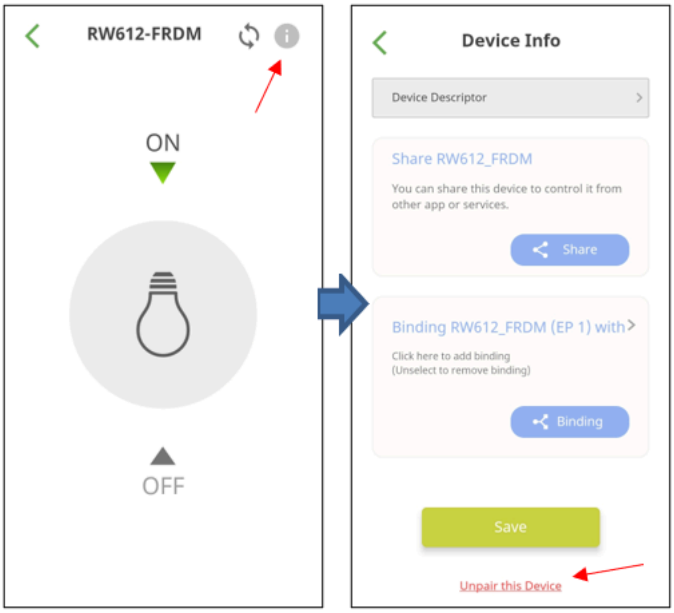

RW612-FRDM
ON
OFF
Device Info
Device Descriptor
Share RW612_FRDM
You can share this device to control it from other app or services.
Share
Binding RW612_FRDM (EP 1) with
Click here to add binding
(Unselect to remove binding)
Binding
Save
Unpair this Device

Figure 37. Unpair device flow screen

6 Troubleshooting

This section provides information about potential issues during the Matter demo setup process and workaround.

6.1 SSID and passphrase update on chip-tool app (Android phone)

To confirm the AP the Android phone is connected to, refer to Figure 38.

- Go to the settings of the Android phone
- Confirm the Wi-Fi connection to the same SSID

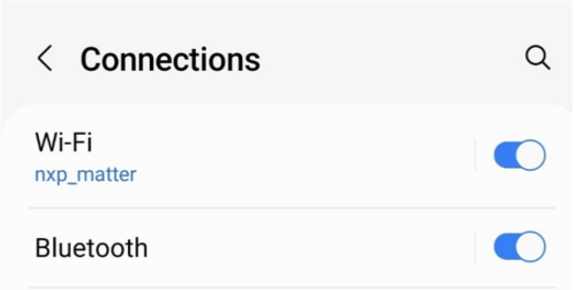


Figure 38. Wi-Fi setup on Android phone

6.2 FRDM-RW612 saved parameters reset

- Connect the RW612 FRDM evaluation board console and enter the command to reset

```
> matterfactoryreset
```

- For the factory reset: on RW612, hold SW2 for 3 seconds.

6.3 FRDM-MCUW71 saved parameters reset

FRDM-MCU W71 saves the pairing information after a successful pairing.

In case you need to reset the saved parameters:

- Hold the SW2 button for more than 3 seconds.
- Look for the following message on the MCU W71-FRDM console:

```
Factory Reset Triggered. Push the RESET button within 6000 ms to cancel!
```

The saved parameters are cleared after a power cycle.

6.4 Commissioning failure rate is high

Note: *To maximize operational success rates, all devices must be equipped with an antenna connection. This setup enhances signal stability and overall system reliability.*

This can occur in a heavy noisy environment. Try to configure a different thread channel with less noise. For details on the OTBR setup on the RW612 FRDM, see Section 5.2

```
>ble adv stop
Done
> otcli dataset init new
Done
>otcli dataset
Done
>otcli dataset channel 17
Done
>otcli dataset commit active
Done
>otcli ifconfig up
Done
>otcli thread start
Done
>otcli state
Done
[It should be leader after a few seconds…,
need to wait a moment and double check with the same command]
 >otcli dataset active
[use for check the OTBR state]
```

6.5 Stuck state after successful pairing

In case of a stuck state after successful pairing, power cycle or click the **Reset** button on both devices. After the two devices reset successfully and get connection, they are in the original success state, and the APP can get the control back.

Note: *After the device reboots, rebind.*

6.6 MCUW71 Fails to be Flashed: Flashing Fuse Keys

If flash w71 meets an issue, try the following check commands:

- Check the flash command work success:

```
.\blhost.exe -p COM9 -- receive-sb-file .
\mcxw71_nbu_ble_15_4_dyn_matter_1_0_17_1.sb3
```

```
Ping responded in 1 attempt(s)
Inject command 'receive-sb-file'
Preparing to send 132948 (0x20754) bytes to the target.
Successful generic response to command 'receive-sb-file'
Data phase write aborted by status 0x2712 kStatus_AbortDataPhase
Possible JUMP or RESET command received.
Response status = 1 (0x1) Failure.
Wrote 0 of 132948 bytes.
```

- Check the Device Life Cycle:

```
.\blhost.exe -p COM9 -- fuse-read 0xa 1
Ping responded in 1 attempt(s)
Inject command 'fuse-read'
Successful response to command 'fuse-read'
07
(1/1)100% Completed!
Successful generic response to command 'fuse-read'
Response status = 0 (0x0) Success.
Response word 1 = 1 (0x1)
Read 1 of 1 bytes.
```

- Check the Fuse. If it has 0x00, apply the following command:

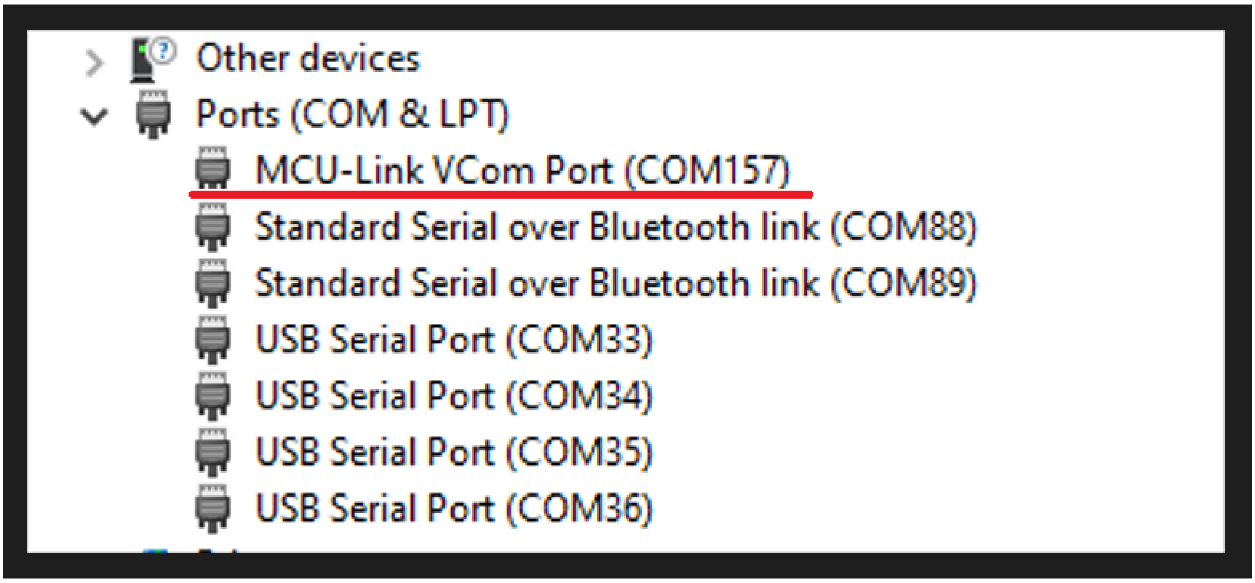
```
.\blhost.exe -p COM9 -- fuse-read 0x20 48
Ping responded in 1 attempt(s)
Inject command 'fuse-read'
Successful response to command 'fuse-read'
Data phase aborted by sender
Response status = 1 (0x1) Failure.
Response word 1 = 0 (0x0)
Read 0 of 48 bytes. Response word 1 = 1 (0x1)
Read 1 of 1 bytes.
```

- The keys are SB3KDK and RoTKTH. Program to FRDM-MCU W71.

```
./blhost.exe -p COMX9 set-property 0x16 1
./blhost.exe -p COMX9 -- fuse-program 0x20
"{{7aa7ef9813b3561257b8837dab26225301df3511217f2733c71dadcd447722d1}}"
./blhost.exe -p COMX9 -- fuse-program 0x1F
"{{650d8097079ff27a3e8a2da14781b922fd8295b6c00bfa067f00e87f1a16b8b304
bf710d45cbd591e2e24be83183922c}}"
./blhost.exe -p COMX9 -- fuse-program 0xD "{{1}}"
./blhost.exe -p COMX9 set-property 0x16 0
```

6.7 FRDM-MCUW71 flash issue – MCU-Link issue

If you cannot flash correctly, the MCXW71 board was detected as MCU-Link. If you encunter the error shown in Figure 39, see the following link to convert it from MCU-Link to J-Link: https://docs.nxp.com/bundle/UM12063/page/topics/Updating_MCU_Link_firmware_using_firmware_update_utility.html.



Other devices
Ports (COM & LPT)
MCU-Link VCom Port (COM157)
Standard Serial over Bluetooth link (COM88)
Standard Serial over Bluetooth link (COM89)
USB Serial Port (COM33)
USB Serial Port (COM34)
USB Serial Port (COM35)
USB Serial Port (COM36)

Figure 39. MCU-Link issue

7 References

- NXP Matter
  1. GitHub repository
  2. Web page
- MCU W71 FRDM
  1. Matter MCXW71 Light switch combo Example Application
  2. MCX W71x Secure and Ultra-Low-Power MCUs for Matter, Thread, Zigbee and Bluetooth LE
- RW612 FRDM
  1. Matter RW61x All-clusters Application
  2. FRDM Development Board for RW612 Wi-Fi 6 + Bluetooth Low Energy + 802.15.4 Tri-Radio Wireless MCU
  3. *FRDM-RW612 Quick Start Guide* (document CS_FRDM-RW612-QSG-LR)
- MCUXpresso
  1. MCUXpresso SDK
  2. MCUXpresso Integrated Development Environment (IDE)
- Matter reference kit
  1. *Getting Started with Matter Reference Kit* (document UM11857)
  2. *NXP Matter Zigbee Bridge User Guide* (document UG10148)
  3. *Getting Started with i.MX 8M Mini and Matter* (document GSG-IMX8MMINI-MATTER)
  4. *Connecting a Matter Device to a Matter Border Router* (document AN14144)

8 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2024 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

9 Revision history

Table 1. Revision history

Document ID	Release date	Description
UG10178 v.1.0	02 December 2024	Initial version

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Bluetooth — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

J-Link — is a trademark of SEGGER Microcontroller GmbH.

Matter, Zigbee — are developed by the Connectivity Standards Alliance. The Alliance's Brands and all goodwill associated therewith, are the exclusive property of the Alliance.

Microsoft, Azure, and ThreadX — are trademarks of the Microsoft group of companies.

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.