UM11861

NXP 802.15.4 Demo Applications for RW612

Rev. 4.0 — 17 April 2025 User manual

Document information

Information	Content
Keywords	Wireless MCU RW612, RW612 EVK board, MCUXpresso SDK, FreeRTOS image
Abstract	Provides step-by-step guidance to configure, compile, debug, flash and run the 802.15.4 sample applications available in the MCUXpresso SDK. Also covers IDE configurations and tool set-ups.

1 About this document

1.1 Purpose and scope

This document provides the steps to configure, compile, flash and run the 802.15.4 sample applications available in the MCUXpresso SDK. It also covers IDE configurations and the setup for the required tools on Windows operating system.

1.2 Considerations

RW612 is based on FreeRTOS, and integrates Wi-Fi 6, Bluetooth Low Energy, and 802.15.4 radios. This user manual does not include information about RW612 device, hardware interconnection, board settings, bring-up, IDE setup, or SDK download. For these items, refer to ref.[1].

Users must install RW612 platform related IDE and tools before using the demo applications.

2 Tool setup

2.1 Serial console tool

The serial console tool is used to input commands and read out the demo application logs on the computer connected to RW612 evaluation board.

- Download and install the terminal emulator software such as minicom (Linux / Mac OS) or Tera Term (Windows).
- Use the following settings for serial console access.

```
Serial Port settings:
- Determine COM port number
- 115200 baud rate - 8 data bits  - No parity
- One stop bit
- No flow control
```

2.2 GNU Arm embedded toolchain

- Download and run the installer from ref.[2].
  The toolchain includes the compiler, linker, and other tools.
- Check the version of GNU Arm Embedded toolchain. It should correspond to the version mentioned in MCUXpresso SDK Release Notes.

Note: *GNU Arm Embedded Toolchain version 10-2021.10 is used as an example in this document.*

- Update the Windows operating system path environment variable with the path to GNU Arm embedded toolchain
  - Go to *Control Panel->System and Security->System->Advanced System Settings*
  - Look for *Environment variables*
  - Add the path *<GNU_Arm_Embedded_Toolchain_install_dir>\bin*

Figure 1 shows an example where the default installation path is *C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2021.10*.

Note: *The toolchain will not work if the path is not set correctly.*

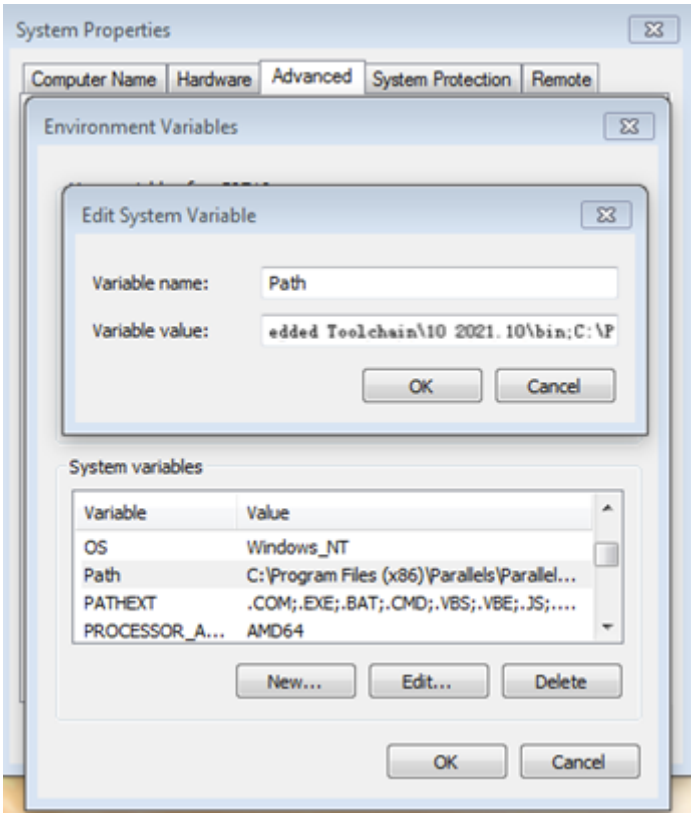

Figure 1. Adding GNU Arm embedded toolchain path to the system environment

2.3 CMake tool

- Download CMake ref.[3].
- Install CMake.
  - Select the option *Add CMake to system PATH*.
  - Choose the option to add CMake to system PATH for all users or just the current user.

Figure 2 shows the example of CMake installation with the option to add CMake to the system PATH for all users.

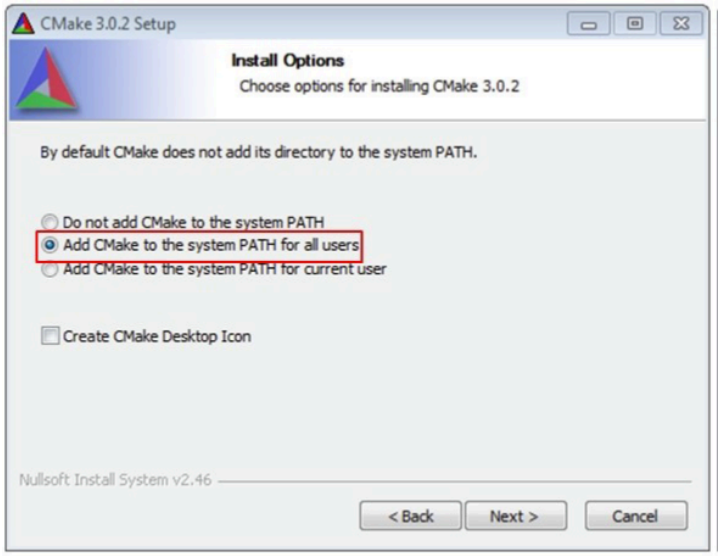

Figure 2. CMake installation options

- Follow the remaining instructions of the installer to complete the installation.
- Reboot your system for the PATH changes to take effect.

2.4 J-Link tool

- Download the latest SEGGER J-Link for Windows from ref.[5].
- Run the installer and follow the instructions to complete the installation.

2.5 Python3

- Download Python 3.x from ref.[9].
- Run the installer and follow the instructions to complete the installation.
  - Select the option *Add Python to PATH* as shown in Figure 3.

Figure 3. Python installation

- Type `python` in Windows command line to get the python version
- Make sure the version is above 3 (3.x)

```
>python
Python 3.8.9
```

2.6 Ninja build tool

- Download Ninja from ref.[6].
- Create a dedicated directory to install Ninja. For example *C:\Program Files\ninja-win*.
- Copy the *ninja.exe* executable file to the new Ninja directory.
- Add the path of the *ninja.exe* file to the Windows environment variables.
- Type `ninja --version` in Windows command line to get Ninja version.
- Check the version is above 1 (1.x).

```
>ninja --version
1.11.0
```

2.7 Git

- Download Git for Windows Standalone Installer from ref.[4].
- Run the installer and follow the instructions to complete the installation.
- Open Windows Explorer.
- Right click any directory and look for *Git Bash Here* in the menu (Figure 4).

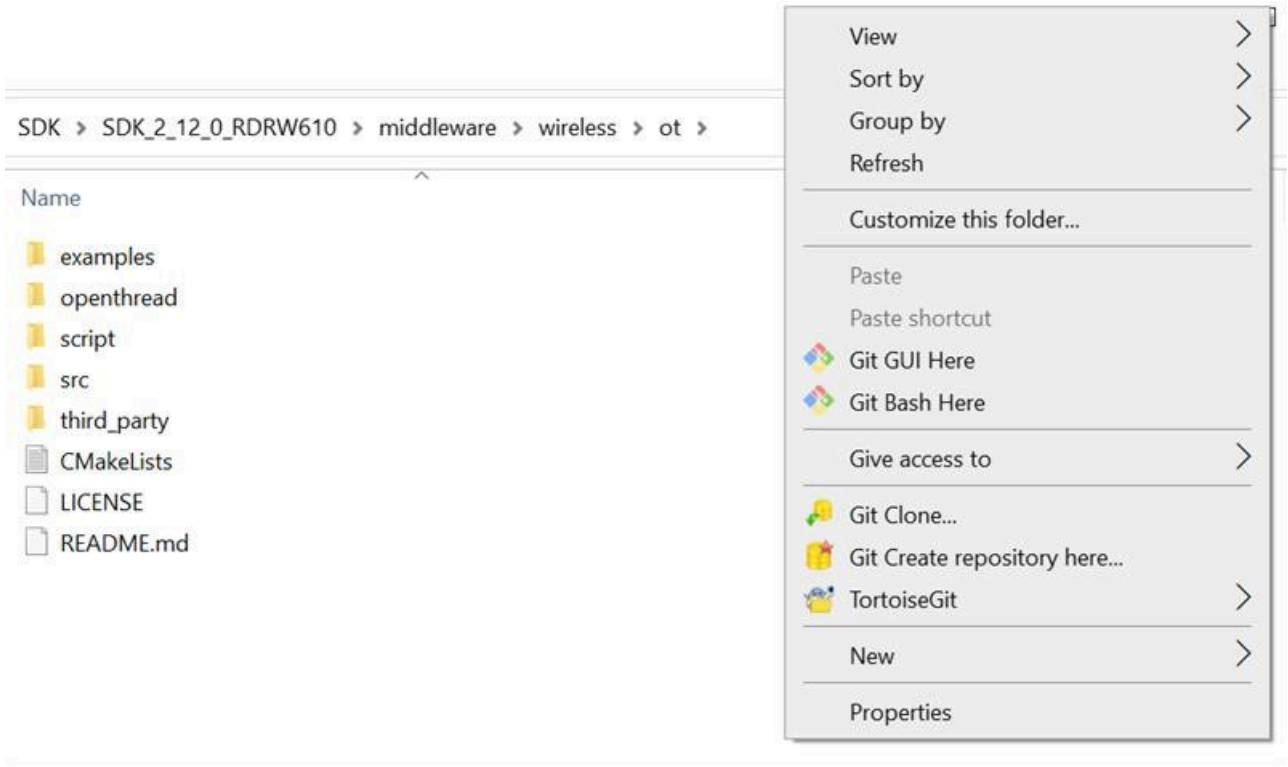

Figure 4. *Git Bash Here* in right-click menu of Windows Explorer

3 Sample applications

The sample applications for 802.15.4 are available in the SDK. This section covers the steps to configure, compile, debug, flash, and execute these examples.

3.1 Ot-cli sample application

The *ot-cli* sample application demonstrates OpenThread features such as how to create a personal area network (PAN), auto negotiation between nodes, and data transmission. The application is based on the open source OpenThread CLI application and uses additional vendor-specific commands.

The test setup uses two RW612 evaluation boards running the same *ot-cli* sample application where:

- One evaluation board acts as leader node
- The second evaluation board acts as child node
- The two devices have the same network configuration
- The role negotiation between the two nodes happens autonomously

3.1.1 Build a demo application

This section describes the steps to build the demo applications using the ARM GCC toolchain on Windows operating system. Refer to Section 2.2 for ARMGCC toolchain installation details. The *ot-cli* application is used as an example.

- Apply a right mouse click on the directory where you want to save OpenThread and select “Git Bash Here” in the menu
- Get OpenThread from NXP GitHub directory

```
$ git clone https://github.com/NXP/ot-nxp.git
$ cd ot-nxp
$ git submodule update --init
```

Note: *Use the default ot-nxp branch and the latest RW612 SDK.*

- Configure the environment variables. Change SDK and ARMGCC paths

```
$ export NXP_RW612_SDK_ROOT='/c/NXP/<SDK-top-dir>'
$ export ARMGCC_DIR='/c/Program Files (x86)/GNU Arm Embedded Toolchain/10 2021.10'
```

- Build OpenThread application image

```
$ ./script/build_rw612 ot_cli
```

- Look for the output application image binary at:

ot-nxp\build_rw612\rw612_ot_cli\bin\ot-cli-rw612.bin

3.1.2 Flash 802.15.4 firmware

RW612 OpenThread application image and 802.15.4 firmware binary are stored in different partitions of FlexSPI NOR flash. The flash layout for 802.15.4 is shown in Figure 5. RW612 OpenThread application loads 802.15.4 firmware into the CPU used for Bluetooth LE/802.15.4.

Figure 5. Partitions of FlexSPI NOR flash

This section describes the steps to flash 802.15.4 firmware with SEGGER J-Link tool.

- Open J-Link commander on Windows command line and connect RW612 device

```
J-Link>con
Device>RW612
TIF>S
Speed><Enter>
```

- Flash 802.15.4 firmware. The path to 802.15.4 firmware binary is:

${SDK-top-dir}\components\conn_fwloader\fw_bin\rw610_sb_ble_15d4_combo.bin

```
> loadbin
${SDK-top-dir}/components/conn_fwloader/fw_bin/rw610_sb_ble_15d4_combo.bin,0x085e0000
```

- Repeat the above operations for another RW612 evaluation board if using another RW612 evaluation board as end device

Note: *802.15.4 firmware is not erased when the application firmware is flashed into the device, and will only need to be reprogrammed into the device if the Flash is fully erased.*

3.1.3 Flash the application image

This section provides the steps to flash the application image on RW612 evaluation board.

- Connect the board to the Windows host system. Open J-Link commander and connect to RW612

```
J-Link>con
Device>RW612
TIF>S
Speed><Enter>
```

- Flash the application image (*ot-cli-rw612.bin*) to RW612 EVK FlexSPI NOR flash.

```
J-Link>loadbin ${sdk}/middleware/wireless/ot/build_rw612/bin/ot-cli-rw612.bin,0x08000400
```

- Reset RW612 evaluation board power.
- Refer to Section 2.1 to access the serial console log.
- Type `help` to get the list of available commands. For more information about ot-cli application commands, see ref.[8].

```
> help
bbr
br
bufferinfo
ccathreshold
ccm
channel
child
childip
childmax
…
```

- Repeat the above operations for another RW612 evaluation board if using another RW612 evaluation board as end device

3.1.4 Ot-cli application execution

3.1.4.1 Create a Thread network

- Connect RW612 evaluation board to the host computer using the USB Micro connector of the evaluation kit.
- Open a terminal emulator program, refer to section 3.1 for serial console access settings.
- On one RW612 evaluation board, stop any existing Thread network and do a factory reset for 802.15.4.

```
> thread stop
Done
> ifconfig down
Done
> factoryreset
```

- Generate a new network configuration.

```
> dataset init new
Done
```

- View the network configuration.

```
> dataset
Active Timestamp: 1
Channel: 13
Channel Mask: 0x07fff800
Ext PAN ID: c153be17b92af7f7
Mesh Local Prefix: fddf:6356:52c3:ec41::/64
Network Key: 6067eade92ca712d1b6d9070cfa4ae43
Network Name: OpenThread-6548
PAN ID: 0x6548
PSKc: b959fc9eb5d78692b7138ae2cbe33277
Security Policy: 672 onrc 0
Done
```

- To change the network parameters, use:

```
> dataset panid 0xabcd
Done
> dataset channel 20
Done
```

- Commit the new dataset and save the dataset in the non-volatile memory.

```
> dataset commit active
Done
```

- Start the Thread network and ensure the state of the device is leader (it may take a few seconds).

```
> ifconfig up
Done
> thread start
Done
> state
Leader
Done
> dataset active -x
0e080000000000010000035060004001fffe0020842af793f623aab540708fd6ec35870785a8d0510f824658f79
d8ca033fbb85ecc3ca91cc030f4f70656e5468726561642d623837300102b8700410f438a194a5e968cc43cc4b
3a6f560ca40c0402a0f7f8000300000b
Done
```

3.1.4.2 Join a Thread network from the end device

Follow the steps below for the end device — the second RW612 evaluation board or a Thread end device. The end device will join the Thread network created in Section 3.1.4.1.

- From the other RW612 evaluation board or from a Thread End device, stop any existing Thread network and do a factory reset for 802.15.4.

```
> thread stop
Done
> ifconfig down
Done
> factoryreset
```

- Issue the command to attach the device to a Thread network:

```
> dataset set active 0e080000000000010000350600040001fffe0020842af793f623aab540708fd6ec
35870785a8d0510f824658f79d8ca033fbb85ecc3ca91cc030f4f70656e5468726561642d623837300102b
8700410f438a194a5e968cc43cc4b3a6f560ca40c0402a0f7f8000300000b
Done
```

- Commit the new data set and save to the non-volatile memory.

```
> dataset commit active
Done
```

- Start the Thread network.

```
> ifconfig up
Done
> thread start
Done
```

- Wait for the state to change to child when the leader has accepted the child.

```
> state
Child
Done
> ipaddr
fddf:6356:52c3:ec41:0:ff:fe00:8002         # Routing Locator (RLOC)
fddf:6356:52c3:ec41:455:3512:7a39:9dda     # Mesh-Local EID (ML-EID)
fe80:0:0:0:a9:c4b:dffa:b068                # Link-Local Address (LLA)
Done
```

Note: *For more information about ot-cli application commands, refer to ref.[7].*

3.1.4.3 Connectivity test

- Ping the child node using the mesh-local address from the leader node

```
> ping fddf:6356:52c3:ec41:455:3512:7a39:9dda
```

The example below indicates a successful connection:

```
> ping fddf:6356:52c3:ec41:455:3512:7a39:9dda
16 bytes from fddf:6356:52c3:ec41:455:3512:7a39:9dda: icmp_seq=1 hlim=64 time=42ms
1 packets transmitted, 1 packets received. Packet loss = 0.0%. Round-trip min/avg/max =
 42/42.0/42 ms.
Done
```

3.1.5 NXP vendor commands

3.1.5.1 Get/set TX power limit

This section shows how to set/get TX power limit for 802.15.4 radio using NXP vendor specific command.

Syntax: `radio_nxp txpwrlimit <parameter>`

Table 1. Command parameters

Command parameter	Description
`parameter`	0 = no TX power limit 1 to 30 = 0.5 dBm to 15 dBm. TX power limit with 0.5 dBm step No parameter: get TX power limit setting

Examples

Vendor command to set TX power limit to 10 dBm:

```
> radio_nxp txpwrlimit 20
Done
```

Vendor command to get TX power limit:

```
> radio_nxp txpwrlimit
20
Done
```

Note:

1. *If no TX power limit is set in the OTP, the default value is 0 which means no power limit is applied.*
2. *The TX power limit command is specified in units of 0.5 dBm. For example, to set the limit to 10 dBm, use a value of 20.*

3.1.5.2 RF test mode

This section introduces the usage of RF test mode with the production firmware and ot-cli sample application.

Syntax: `radio_nxp mfgcmd <Command ID> <Parameters…>`

Table 2 describes RF test mode commands.

Table 2. RF test mode commands

Feature	Command ID	Parameters	Description
RF test mode	`1` (enable) / `0` (disable)	–	Enable/disable 802.15.4 RF test mode. By default RF test mode is disabled.
Channel	`11` (get) / `12` (set)	Channel 11 to 26	RF channel for RF tests
TX power	`15` (get) / `16` (set)	-20 to 15 in the unit of 1 dBm	TX power used for RF tests
Continuous TX	`17`	`1` (enable) / `0`(disable)	Start or stop continuous transmission using the channel and TX power defined above
Payload size	`20` (get) / `21` (set)	17 to 116 bytes	Size of the payload used for transmission
Start RX test	`32`	–	Start receiving frames. Count the frames and calculate the average RSSI
Get RX test result	`31`	–	Get the result from RX test (command `32`)
Burst TX	`33`	`mode`(0-7) + `packet gap`	Start to transmit a specific number of packets with a specific packet gap `mode`: mode options (unit: packet): 0=1, 1=25, 2=100, 3=500, 4=1000, 5=2000, 6=5000, 7=10000 `Packet gap`: must be > 5ms
Duty cycle TX	`35`	`1` (enable) / `0` (disable)	Start or stop to transmit in duty cycle TX mode
CCA threshold	`47` (get) / `48` (set)	0 dBm to -110 dBm	Configure CCA threshold for following test
Continuous CCA test	`49`	`1` (enable) / `0` (disable) + `mode`	Start CCA tests `mode`= mode options (1/2/3)
Get CCA status	`50`	–	Get the result from CCA tests
Continuous ED Test	`55`	`1` (enable) / `0` (disable)	Start/stop energy detection (ED) test
Get ED value	`56`	–	Get ED test results

RF test mode commands can return the following error codes:

- `INVALID PARAMETER`: out-of-range parameter or wrong number of parameters
- `NOT IMPLEMENTED`: unknown command ID
- `FAILED`: no response or response with an error code

Examples

Command to enable the RF test mode functionality:

```
> radio_nxp mfgcmd 1
Done
```

Command to disable the RF test mode functionality:

```
> radio_nxp mfgcmd 0
Done
```

Command to set RF channel to channel 11:

```
> radio_nxp mfgcmd 12 11
Done
```

Command to get RF channel setting:

```
> radio_nxp mfgcmd 11
11
Done
```

Command to set TX power to 10 dBm:

```
> radio_nxp mfgcmd 16 10
Done
```

Command to get TX power setting:

```
> radio_nxp mfgcmd 15
10
Done
```

Command to set TX payload size to 30 bytes:

```
> radio_nxp mfgcmd 21 30
Done
```

Command to get TX payload size setting:

```
> radio_nxp mfgcmd 20
30
Done
```

Command to start continuous TX mode:

```
> radio_nxp mfgcmd 17 1
Done
```

Command to start burst TX mode for 10000 packets with the packet gap of 10 ms:

```
> radio_nxp mfgcmd 33 7 10
Done
```

Command to start duty cycle TX mode:

```
> radio_nxp mfgcmd 35 1
Done
```

Command to start RX test mode:

```
> radio_nxp mfgcmd 32
Done
```

Command to get the result from RX test:

```
> radio_nxp mfgcmd 31
Status : 0
rx_packet_count : 500
total_packet_count : 500
rssi : -32
Done
```

Command sequence example for 802.15.4 transmitter/receiver test:

802.15.4 transmitter test:

```
> radio_nxp mfgcmd 1          # Enable 15.4 RF test mode
Done
> radio_nxp mfgcmd 12 11      # Set channel to 11
Done
> radio_nxp mfgcmd 11         # Get the channel setting
11
Done
> radio_nxp mfgcmd 21 30      # Set TX payload size to 30 bytes
Done
> radio_nxp mfgcmd 20         # Get TX payload size setting
30
Done
> radio_nxp mfgcmd 16 10      # Set TX power to 10 dBm
Done
> radio_nxp mfgcmd 15         # Get TX power setting
10
Done
> radio_nxp mfgcmd 33 7 10    # start Burst TX mode for 10000 Packets with the
                              # Packet gap of 10ms
Done
```

802.15.4 receiver test:

```
> radio_nxp mfgcmd 1          # Enable 15.4 RF test mode
Done
> radio_nxp mfgcmd 12 11      # Set channel to 11
Done
> radio_nxp mfgcmd 11         # Get the channel setting
11
Done
> radio_nxp mfgcmd 59 1 0xface 0xabcd 0xef12 # Set panid source and destination id
> radio_nxp mfgcmd 32         # Start 15.4 RX test
Done
# After 15.4 RX test started, start TX with corresponding settings from another device
> radio_nxp mfgcmd 31         # End RX test and get the result
Status : 0
rx_packet_count : 500
total_packet_count : 500
rssi : -32
Done
```

4 Note about the source code in the document

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

Copyright 2022-2025 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

5 Abbreviations

Table 3. Abbreviations

Abbreviation	Definition
CLI	Command line interface
ED	Energy detection
EVK	Evaluation kit
FW	Firmware
IDE	Integrated development environment
IP	Internet protocol
OT	OpenThread
OTP	One time programmable
PAN	Personal area network
SDK	Software development kit
SW	Software

6 References

[1] User manual – UM11798: Getting Started with Wireless on RW61x Evaluation board Running FreeRTOS
[2] Webpage – Arm GNU Toolchain (link)
[3] Webpage – cMake (link)
[4] Webpage – Git Downloads (link)
[5] Webpage – J-Link (link)
[6] Webpage – ninja-build project on GitHub (link)
[7] Webpage – OpenThread on NXP RW612 Example (link)
[8] Webpage – OpenThread project on GitHub (link)
[9] Webpage – Python (link)

7 Contact Us

For more product details, queries and support, use:

- Web support: link
- NXP community: link

8 Revision history

Table 4. Revision history

Rev	Date	Description
UM11861 v.4.0	17 April 2025	• Changed the document access to public. • Section 6 "References": updated.
UM11861 v.3.0	22 April 2024	• Section 2.4 "J-Link tool": updated. • Section 3.1.1 "Build a demo application": updated. • Section 3.1.4.1 "Create a Thread network": updated. • Section 3.1.4.2 "Join a Thread network from the end device": updated. • Section 3.1.4.3 "Connectivity test": updated. • Section 3.1.5.2 "RF test mode": updated the steps for 802.15.4 receiver test. • Section 6 "References": updated.
UM11861 v.2.0	17 July 2023	• Section 2.4 "J-Link tool": updated the link to J-Link patch for RW61x • Section 3.1.1 "Build a demo application": updated • Section 3.1.2 "Flash 802.15.4 firmware": updated • Section 3.1.3 "Flash the application image": updated the command to flash the application image • Section 3 "Sample applications": updated the introduction • Section 3.1 "Ot-cli sample application": updated • Section 3.1.5 "NXP vendor commands": added the section • Section 5 "Abbreviations": added ED and OTP • Section 4 "Note about the source code in the document": added
UM11861 v.1.0	7 December 2022	• 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.

Tables

Figures

Contents