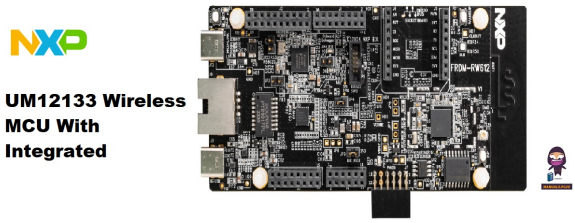
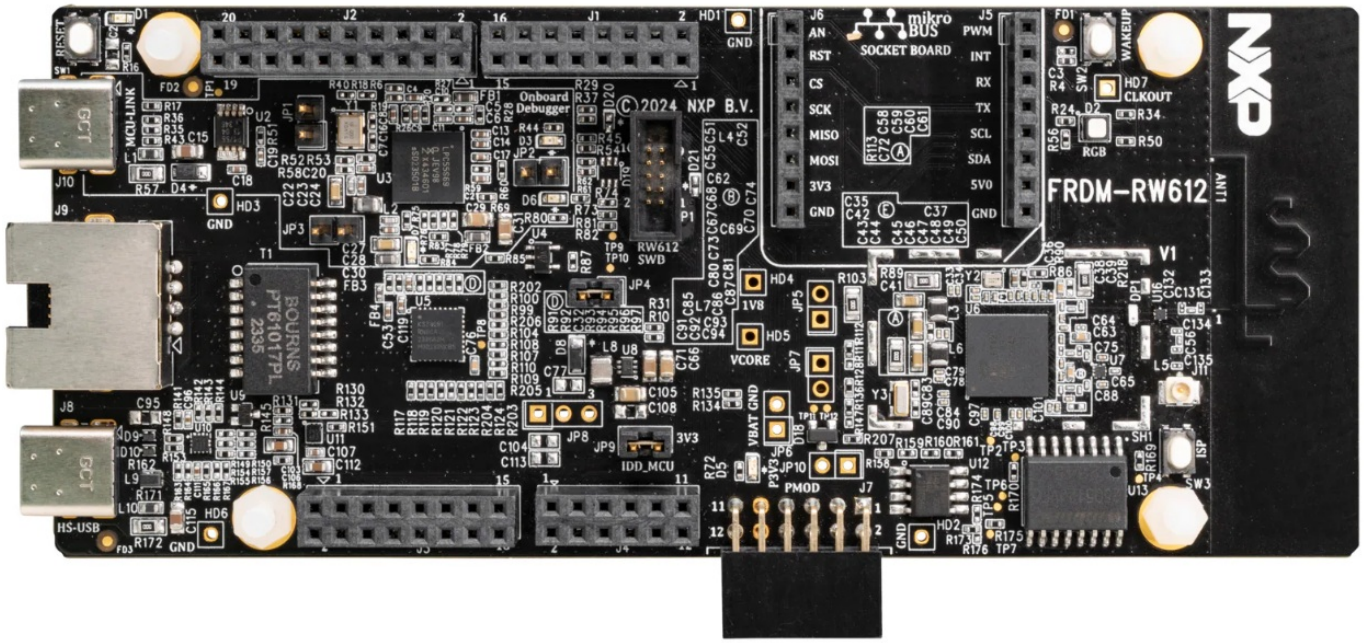


NXP UM12133 Wireless MCU With Integrated User Guide

Contents

NXP UM12133 Wireless MCU With Integrated



Specifications

- Product: NXP NCP Application Guide for RW612 with MCU Host
- Model: UM12133
- Revision: 1.0
- Date: 24 September 2024

Product Information

Abstract: Describes the hardware connections/interfaces and software modifications to enable NCP mode on the NXP MCU target host.

Description: The NXP NCP Application Guide provides instructions on setting up the RW612 Wireless MCU with the i.MX RT1060 MCU host in NCP mode to offload network connectivity tasks.

Product Usage Instructions

About this Document

Purpose and Scope: This user manual covers applications related to Wi-Fi, Bluetooth Low Energy, and OpenThread for RW612.

What is NCP Mode?

In NCP mode, the RW612 Wireless MCU handles Wi-Fi, Bluetooth LE, and IEEE 802.15.4 stacks while the I. MX RT1060 manages application code, leading to power and memory savings.

Board Setup

To enable NCP mode between RW612 and i.MX RT1060, configure the NCP application on RW612 at compile time. Supported host interfaces are UART, USB, SDIO, and SPI.

FAQ

- **Q: What are the key features of NCP mode?**
  - **A:** NCP mode offloads network connectivity tasks to RW612, allowing i.MX RT1060 to focus on application code, resulting in power and memory efficiency.
- **Q: How can I switch between different host interfaces for the NCP application?**
  - **A:** You can configure the host interface (UART, USB, SDIO, or SPI) for the NCP application during compilation on the RW612.

Document information

Information	Content
Keywords	Wireless MCU RW612, RW612 EVK board, i.MX RT1060, network co-processor (NCP), hardware connection
Abstract	Describes the hardware connections/interfaces and software modifications to enable NCP mode on the NXP MCU target host.

About this document

Purpose and scope

This user manual describes:

- The NXP NCP application for RW612 with MCU host platform i.MX RT1060 as example.
- The hardware connections for one of the four supported interfaces to enable NCP mode on the NXP RW612 BGA V4 board (UART, USB, SDIO, or SPI).
- The method to build and run the NCP applications on both the NCP host (i.MX RT1060) and the NCP device(RW612). The applications apply to Wi-Fi, Bluetooth Low Energy and OpenThread (OT).

Considerations

This document does not include details about RW612 or i.MX RT1060. It is assumed that you are familiar with the following:

- RW612 wireless microcontroller:
- Bring-up of Wi-Fi, Bluetooth, or 802.15.4 radios
- Hardware interconnection
- IDE setup
- SDK download
- i.MX RT1060 board:
- Board settings
- Flashing of the BSP
- IDE setup

For information about the Wi-Fi, Bluetooth, or 802.15.4 radios, hardware interconnection, board settings, bringup,

IDE setup, and SDK download for RW612, refer to [2]. For information about board settings, bring-up, IDE setup, SDK download for i.MX RT1060, refer to [3]. UM12133

What is NCP mode

Network co-processor (NCP) is a module designed to offload the network connectivity tasks from the main microcontroller to a wireless MCU. The connectivity tasks relate to Wi-Fi, Bluetooth LE, and OpenThread (OT). In this document, RW612 is the Wireless Microcontroller (MCU) and i.MX RT1060 is the application processor (MCU). RW612 is a standalone device: no external processor is required to run TCP/IP, wireless stacks and application layers, lower layers (MAC/LL/PHY), drivers, security, filesystem, and application codes. Figure 1 shows the diagram of standalone RW612.

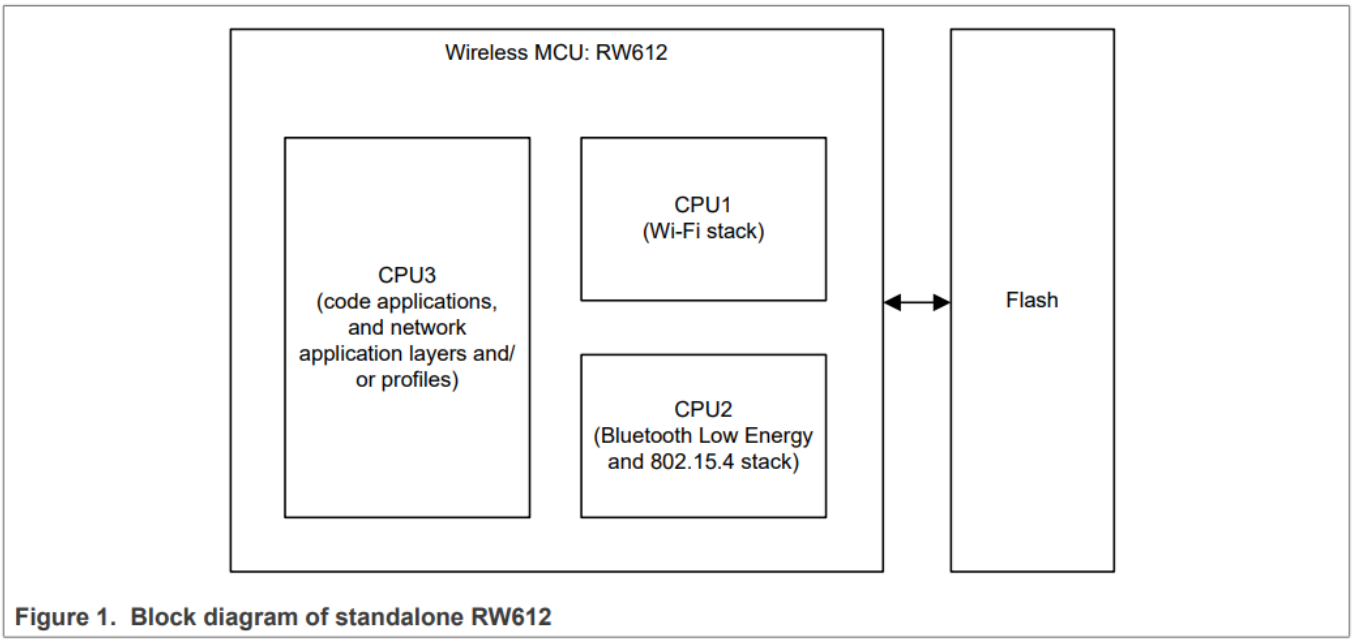


Figure 1. Block diagram of standalone RW612

With NCP mode, i.MX RT1060 manages the application code while RW612 handles Wi-Fi, Bluetooth Low Energy, and IEEE 802.15.4 stacks. The memory and processing power are split between RW612 and i.MX RT1060, which contributes to power and memory savings.

Figure 2 shows the block diagram of offloading with NCP mode.

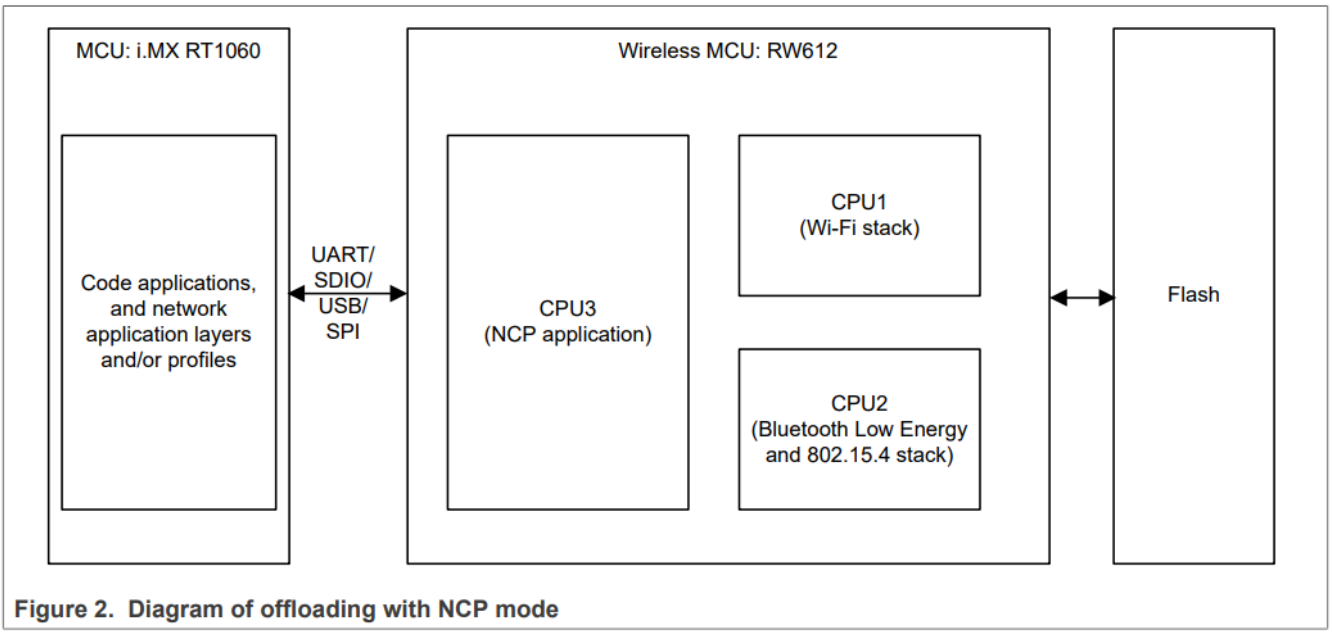


Figure 2. Diagram of offloading with NCP mode

Board setup

This section describes how to enable NCP mode between RW612 and the MCU host (i.MX RT1060). The NCP application on the RW612 is configurable at compile time, and supports one of the following host interfaces: UART, USB, SDIO, and SPI.

UART interface

The power supply of i.MX RT1060 EVK board

Two power supply methods are available for I. MX RT1060 EVK board:

- Using an external 5 V power supply.
- Enabling a 5 V power supply from the USB connector.

To enable the 5 V power supply from the USB connector, implement the following changes:

- Install R31 with 0 Ω.
- Connect J40 Pin5–6. Keep the other pins of J40 not connected.

Note: In this document, the USB connector is used to power i.MX RT1060.

Board configuration

To enable NCP mode over UART interface, configure the RW612 board.

- Disconnect JP19.
- Connect JP9 and JP23.
- Connect JP47 to GND.

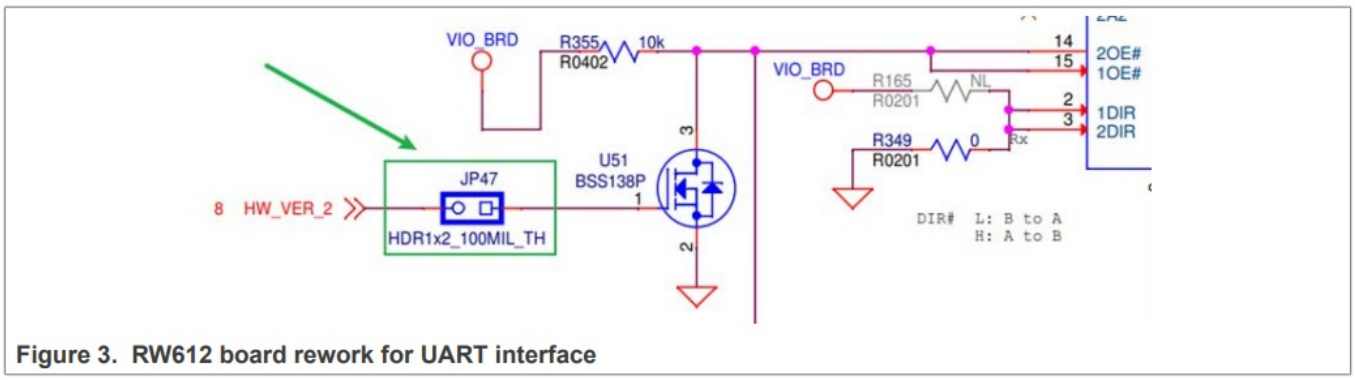


Figure 3. RW612 board rework for UART interface

Note:

- The rework prevents the signal interference from SPI to UART RX.

Pin connections between RW612 and i.MX RT1060

Table 1 lists the pin connections between the Flexcomm0 UART interface on RW612 BGA V4 board and

LPUART3 interface on i.MX RT1060 EVKB board.

Table 1. UART pin connections for RW612 and i.MX RT1060

RW612 BGA V4 board		i.MX RT1060 EVKB board	
Signal	Board location	SIgnal	Board location
UART_RXD	HD2 Pin3	UART_TXD	J16 Pin3
UART_TXD	HD2 Pin4	UART_RXD	J16 Pin2
UART_CTS	HD11 Pin14	UART_RTS	J33 Pin4
UART_RTS	HD11 Pin8	UART_CTS	J33 Pin1
GND	HD2 Pin15	GND	J2 Pin20

Hardware connection

Figure 4 shows the hardware connection between RW612 BGA V4 board and i.MX RT1060 EVKB board with UART interface.

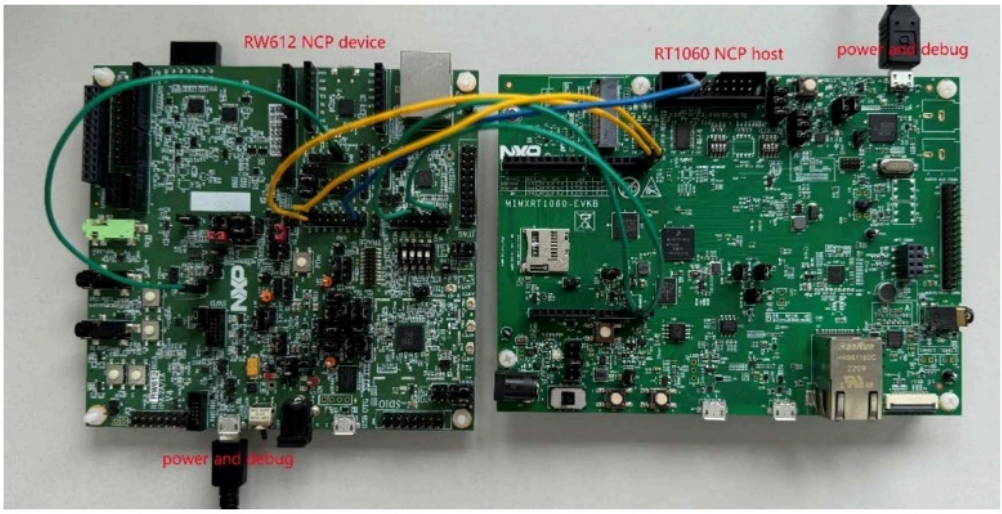


Figure 4. Connection of RW612 EVK to i.MX RT1060 EVKB over UART interface

USB interface

Figure 5 shows the connection between RW612 BGA V4 board and i.MX RT1060 EVKB board with USB interface.

- Connect the USB-OTG (J12) port on RW612 EVK board to J48 port on i.MX RT1060 EVK board via an USBto-Micro-USB converter

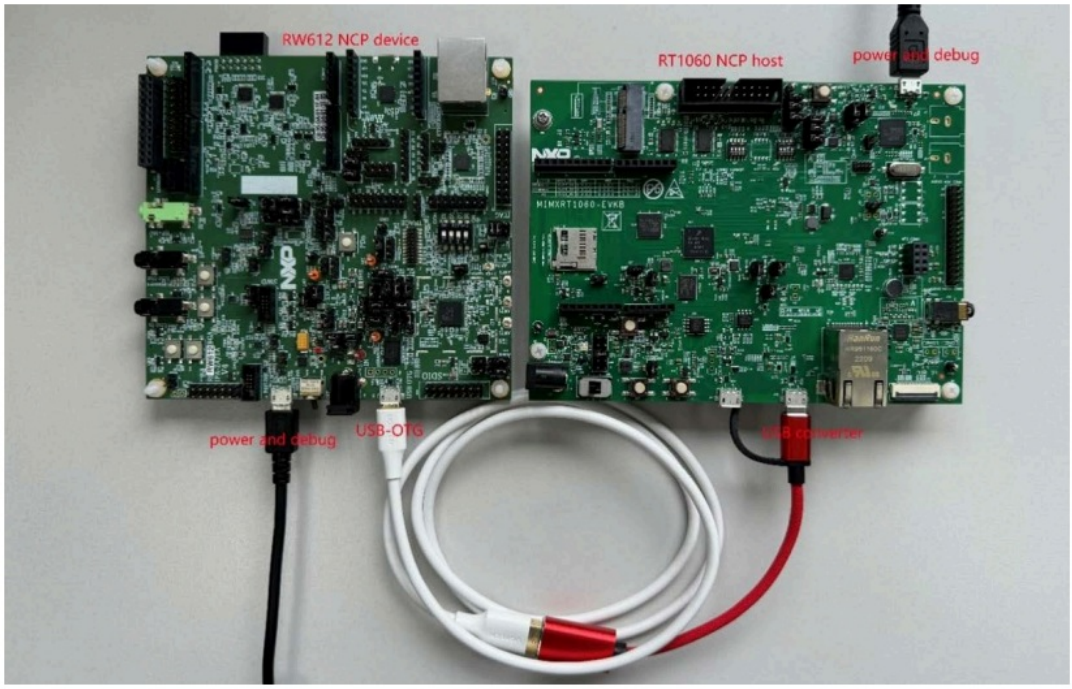


Figure 5. Connection between RW612 and i.MX RT1060 with USB interface

SPI interface

Board configuration

To enable NCP mode over SPI:

- Configure the RW612 EVK board (SPI target):
- Connect JP30 1-2, JP9, JP19, JP23, and JP51.
- Disconnect JP47 and connect JP47-1 to GND (for example HD3 pin 5)
- Remove R97, R415, R594, and R656.
- Install R409, R49, R13, R43, and R520.
- Configure i.MX RT1060 EVKB board (SPI controller):
  - Add R356, R350, R346, and R362.

    Note: If the connection of JP19 causes UART3 to stop working, remove R101.

Pin connections

Table 2 lists the pin connections required to route RW612 SPI signals (target) to i.MX RT1060 SPI signals(controller). For instance, connect RW612 J5 Pin5 to i.MX RT1060 J17 Pin5. See Section 3.3.3.

- GPIO1_17: The SPI target notifies the SPI controller to transmit.
- GPIO1_16: The SPI target notifies the SPI controller that Direct Memory Access (DMA) is ready and transmit can start.

Table 2. Pin connections between RW612 board and i.MX RT1060 EVK

SPI controller (i.MX RT1060 EVKB)		SPI target (RW612 EVK)	
Pin name	Board location	Pin name	Board location
CITO	J17 Pin5	CITO	J5 Pin5
COTI	J17 Pin4	COTI	J5 Pin4
SCK	J17 Pin6	SCK	J5 Pin6
CS	J17 Pin3	CS	J5 Pin3
GND	J17 Pin7	GND	J5 Pin7
GPIO1_17	J17 Pin10	GPIO27	HD2 Pin11
GPIO1_16	J17 Pin9	GPIO11	HD11 Pin13

Hardware connection

Figure 6 shows the connection of RW612 board with i.MX RT1060 EVK board with SPI interface.

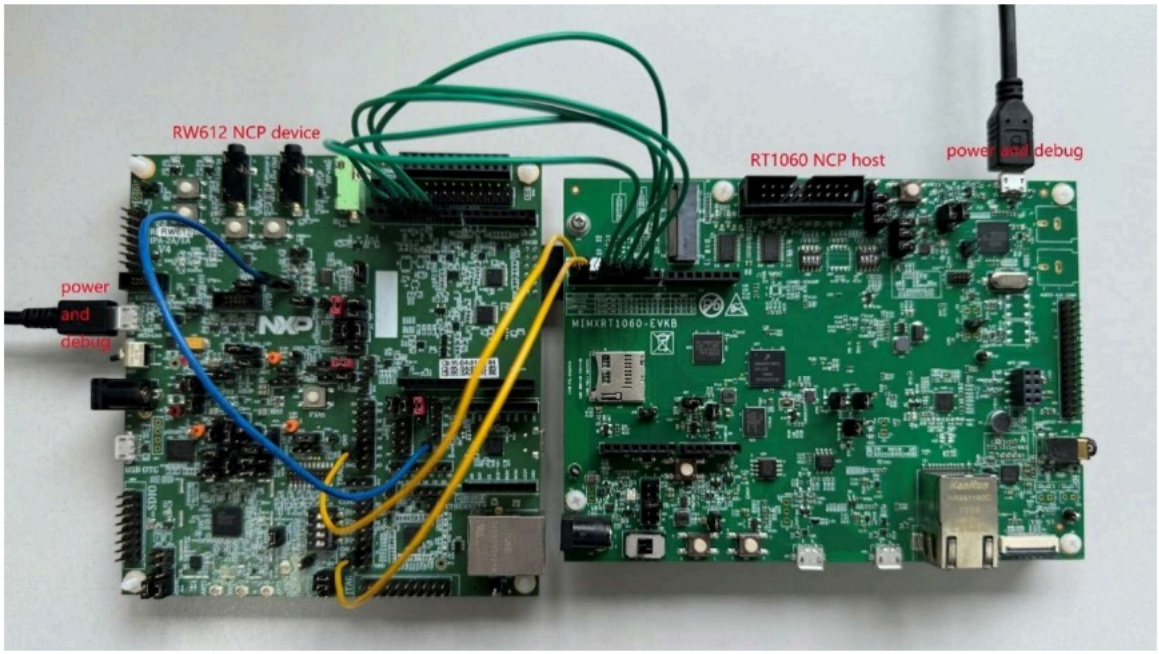


Figure 6. Connection between RW612 board and i.MX RT1060 EVK board over SPI

SDIO interface

Board configuration

To enable NCP mode over SDIO, configure the RW612 BGA board:

- Connect JP16 1-2

Note: By default, RW612 BGA board works with 3.3 V IO voltage. To change IO voltage to 1.8 V, connect JP16 1-2.

Hardware connection

Connect i.MX RT1060 board to RW612 board with the mini-SDIO cable (Figure 7).

Figure 6 shows the connection of RW612 board with i.MX RT1060 EVK board with SPI interface.

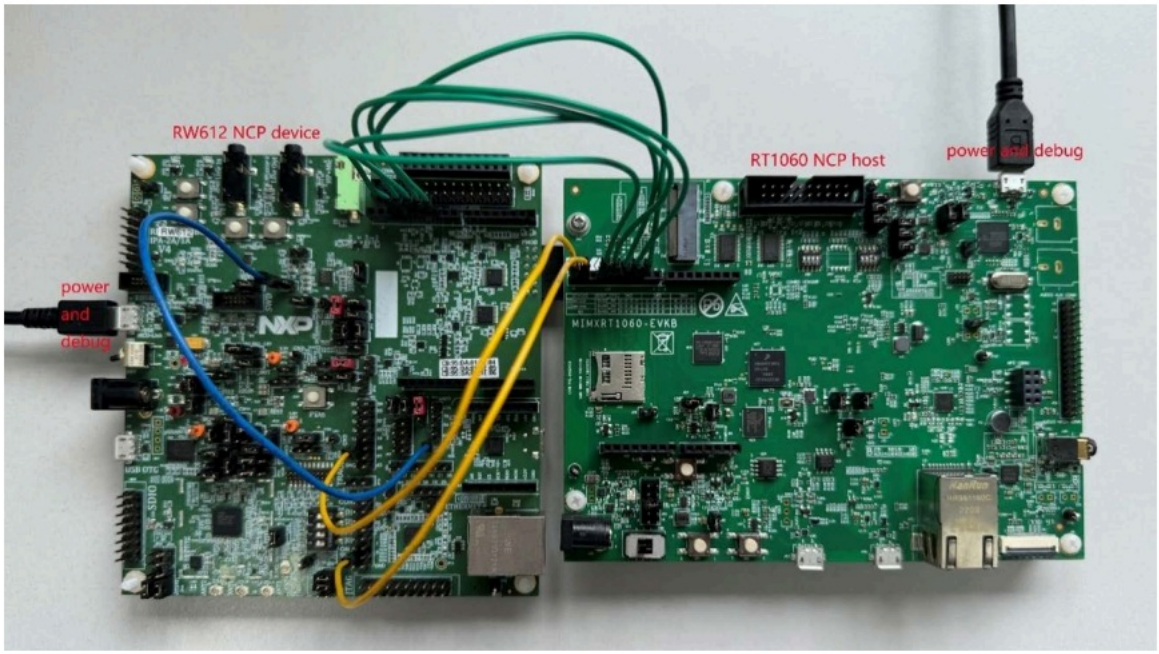


Figure 6. Connection between RW612 board and i.MX RT1060 EVK board over SPI

Compile NCP host application

The NCP Application on i.MX RT1060 is called ncp_host. Download the i.MX RT1060 SDK (version 2.16.000 and above) from [4].

Path to the NCP host application example: <i.MX RT1060-SDK-top-dir>\boards\evkbmimxrt1060\ncp_examples\ncp_host.

The suggested toolchains to compile NCP applications are:

- IAR: 9.50.1
- ARMGCC: 12.3.1

Wi-Fi

Steps to build the NCP host application for Wi-Fi:

- Step 1 – Open the example in IAR, or use ARMGCC as a compilation tool.
- Step 2 – Check that CONFIG_NCP_WIFI is defined in ncp_host_config.h.

Path to ncp_host_config.h:

<i.MX RT1060-SDK-top-dir>\boards\evkbmimxrt1060\ncp_examples\ncp_host\ncp_host_config.h

```
#define CONFIG_NCP_WIFI 1
```

Step 3 – Set the hardware interface in ncp_host_config.h based on the hardware connection.

```
/* Interface options */
#define CONFIG_NCP_UART 1
#define CONFIG_NCP_SPI 0
#define CONFIG_NCP_USB 0
#define CONFIG_NCP_SDIO 0
```

Note: Multiple interfaces are not supported in parallel. Enable only one NCP interface at a time and disable the other interfaces.

Step 4 – Set CONFIG_NCP_SUPP to 1 or 0 according to the setting of ncp_device. The value must be the same for the host and for the device.

```
#define CONFIG_NCP_SUPP 1
```

- `CONFIG_NCP_SUPP` is used to indicate whether wpa_supplicant is supported or not.
- If `CONFIG_NCP_SUPP` is defined for the NCP device, define the macro `CONFIG_NCP_SUPP` for NCP host.
- If `CONFIG_NCP_SUPP` is not defined for the NCP device, undefine the macro `CONFIG_NCP_SUPP` for the NCP host.

Step 5 – Compile the example and program the application image to i.MX RT1060. Find more details in [1].

Bluetooth Low Energy

The example for Bluetooth LE NCP host application is located in the directory: <i.MXRT1060-SDK-top-dir>\boards\evkbmimxrt1060\ncp_examples\ncp_host.

- Steps to build the NCP host application for Bluetooth LE:
- Step 1 – Import the NCP example to IAR, or use ARMGCC as compilation tool.
- Step 2 – Make sure CONFIG_NCP_BLE is defined as 1 in ncp_host_config.h.

```
#define CONFIG_NCP_BLE 1
```

Step 3 – Set the hardware interface in ncp_host_config.h based on the hardware connection.

Note: Multiple interfaces are not supported in parallel. Enable only one NCP interface at a time and disable the other interfaces

```
/* Interface options */
#define CONFIG_NCP_UART 1
#define CONFIG_NCP_SPI 0
#define CONFIG_NCP_USB 0
#define CONFIG_NCP_SDIO 0
```

Step 4 – Compile the example and program the application image to i.MX RT1060. Find more details in [1].

Thread

The NCP host application for Thread located is available at: **https://github.com/NXP/ot-nxp/**.

Note: For Thread, only ARMGCC can be used to compile NCP host (i.MX RT1060) and device (RW612) application.

Step 1 – Clone the repository, update the SDK and compile the OT NCP host example:

```
$ git clone -b v1.4.0-pvw1 https://github.com/NXP/ot-nxp.git # Clone from GitHub
$ cd ot-nxp                                  # Switch to the ot-nxp folder
$ git submodule update --init                # Update submodules
$ cd third_party/github_sdk/                 # Switch to the sdk folder for RW612
$ west init -l manifest --mf west.yml        # prepare for update RW612 SDK online
$ west update                                # update RW612 SDK online
$ cd <path-to-ot-nxp>l                       # Switch back to the ot-nxp folder
```

Step 2 – Move to the ot-nxp directory and build the NCP host application for different interfaces. Command for the UART interface:

Command for the UART interface:

```
$ ./script/build_rt1060 rw612_ncp_host -DOT_NCP_RW612_INTERFACE=UART
```

Command for the USB interface:

```
$ ./script/build_rt1060 rw612_ncp_host -DOT_NCP_RW612_INTERFACE=USB
```

Command for the SDIO interface:

```
$ ./script/build_rt1060 rw612_ncp_host -DOT_NCP_RW612_INTERFACE=SDIO
```

Command for the SPI interface:

```
$ ./script/build_rt1060 rw612_ncp_host -DOT_NCP_RW612_INTERFACE=SPI
```

The OT NCP host application binary ot-cli-rt1060.bin is located in the build_rt1060/rw612_ncp_host/bin directory.

Compile NCP device application

The NCP Application on RW612 is called ncp_device. The path to the source code of the application in RW612 SDK is: <RW612-SDK-top-dir>\boards\rdRW612bga\ncp_examples\ncp_device.

The suggested toolchains to compile NCP device applications are:

- IAR: 9.50.1
- ARMGCC: 12.3.1

Wi-Fi

Steps to compile the ncp_device application for Wi-Fi:

Step 1 – Import the example to IAR, or use ARMGCC as a compilation tool.
Step 2 – Make sure CONFIG_NCP_WIFI is defined as 1 in app_config.h.

The path to app_config.h file is: <RW612-SDK-top-dir>\boards\rdRW612bga\ncp_examples\ncp_device\app_config.h

Step 3 – Choose the hardware interface in app_config.h file based on the hardware connection

```
#define CONFIG_NCP_UART 1
#define CONFIG_NCP_SPI 0
#define CONFIG_NCP_USB 0
#define CONFIG_NCP_SDIO 0
```

Note: Multiple interfaces are not supported in parallel. Enable only one NCP interface at a time and disable the other interfaces.

Step 4 – Define CONFIG_NCP_SUPP to 1 or 0 according to the setting of NCP host side. The value should be the same on the host and on the device sides.

```
#define CONFIG_NCP_SUPP  1
```

CONFIG_NCP_SUPP is defined in wifi_config.h file.

The path to wifi_config.h file is: <RW612-SDK-top-dir>\boards\rdRW612bga\ncp_examples\ncp_device\wifi\wifi_config.h

Step 5 – Compile the example and program the application image to RW612. Find more details in [2]. UM12133

Bluetooth Low Energy

The path to the source code of the Bluetooth LE application in RW612 SDK is: <RW612-SDK-top-dir>\boards\rdRW612bga\ncp_examples\ncp_device

- Steps to compile the ncp_device application for Bluetooth LE:
- Step 1 – Import the example to IAR, or use ARMGCC as a compilation tool.
- Step 2 – Make sure CONFIG_NCP_BLE is defined in app_config.h.

```
#define CONFIG_NCP_BLE 1
```

Step 3 – Choose the hardware interface in app_config.h file based on the hardware connection.

```
/* Interface options */
#define CONFIG_NCP_UART 1
#define CONFIG_NCP_SPI 0
#define CONFIG_NCP_USB 0
#define CONFIG_NCP_SDIO 0
```

Note: Multiple interfaces are not supported in parallel. Enable only one NCP interface at a time and disable the other interfaces.
Step 4 – Compile the example and program the application image to RW612. Find more details in [2].

Thread

The NCP device application for Thread is located at: **https://github.com/NXP/ot-nxp/**

Note: For Thread, only ARMGCC can be used to compile NCP host (i.MX RT1060) and device (RW612) application.

Step 1 – Clone the repository, update the SDK and compile the OT NCP device example:

```
$ git clone -b v1.4.0-pvw1 https://github.com/NXP/ot-nxp.git # Clone the Git repo
$ cd ot-nxp                                                  # Switch to the ot-nxp directory
$ git submodule update --init                                # Update submodules
$ cd third_party/github_sdk/                                 # Switch to the sdk directory for RW612
$ west init -l manifest --mf west.yml                        # prepare for update RW612 SDK online
$ west update                                                # update RW612 SDK online
$ cd <path-to-ot-nxp>                                        # Switch back to the ot-nxp directory
```

Step 2 – Move to the ot-nxp directory and build the NCP device application for the different interfaces.Command for the UART interface:

```
$ ./script/build_rw612 ot_cli -DOT_NCP_RADIO=ON -DOT_NXP_NCP_UART_INTERFACE=ON
```

Command for USB interface:

```
$ ./script/build_rw612 ot_cli -DOT_NCP_RADIO=ON -DOT_NXP_NCP_USB_INTERFACE=ON
```

Command for SDIO interface:

```
$ ./script/build_rw612 ot_cli -DOT_NCP_RADIO=ON -DOT_NXP_NCP_SDIO_INTERFACE=ON
```

Command for the SPI interface:

```
$ ./script/build_rw612 ot_cli -DOT_NCP_RADIO=ON -DOT_NXP_NCP_SPI_INTERFACE=ON
```

The NCP device application binary located in *build_rw612/rw612_ot_cli/bin* directory.

Run NCP host application

Wi-Fi

Step 1 – Load the NCP device image on RW612 board with the specified interface (Section 3). Table 3 shows the image load addresses.

Table 3. Image load addresses for Wi-Fi NCP device application

Image Load address

- Bluetooth LE/802.15.4 combo firmware 0x085e0000
- NCP device application binary 0x08000000

Step 2 – Program the NCP host image on i.MX RT1060 board with the specified interface.
Step 3 – Connect RW612 to i.MX RT1060 with the specified interface. Refer to section 3.
Step 4 – Power on i.MX RT1060 board and RW612 board.

Note:

- If RW612 board and i.MX RT1060 EVKB are connected over UART, i.MX RT1060 EVKB must be powered on first. The requirement for i.MX RT1060 EVKB power-on timing is that the core must be powered on before the

I/O.

- If RW612 is powered on first, RW612 UART TX drives the voltage to the IO of the i.MX RT1060 EVKBUART RX.
- If I/O is powered on before the core on the i.MX RT1060 SOC, the startup of i.MX RT1060 is affected.
- If RW612 and i.MX RT1060 EVKB are connected over SDIO, RW612 must be powered on first.

Step 5 – Run the NCP host application on i.MX RT1060 board and get the list of supported Wi-Fi commands (Figure 8).

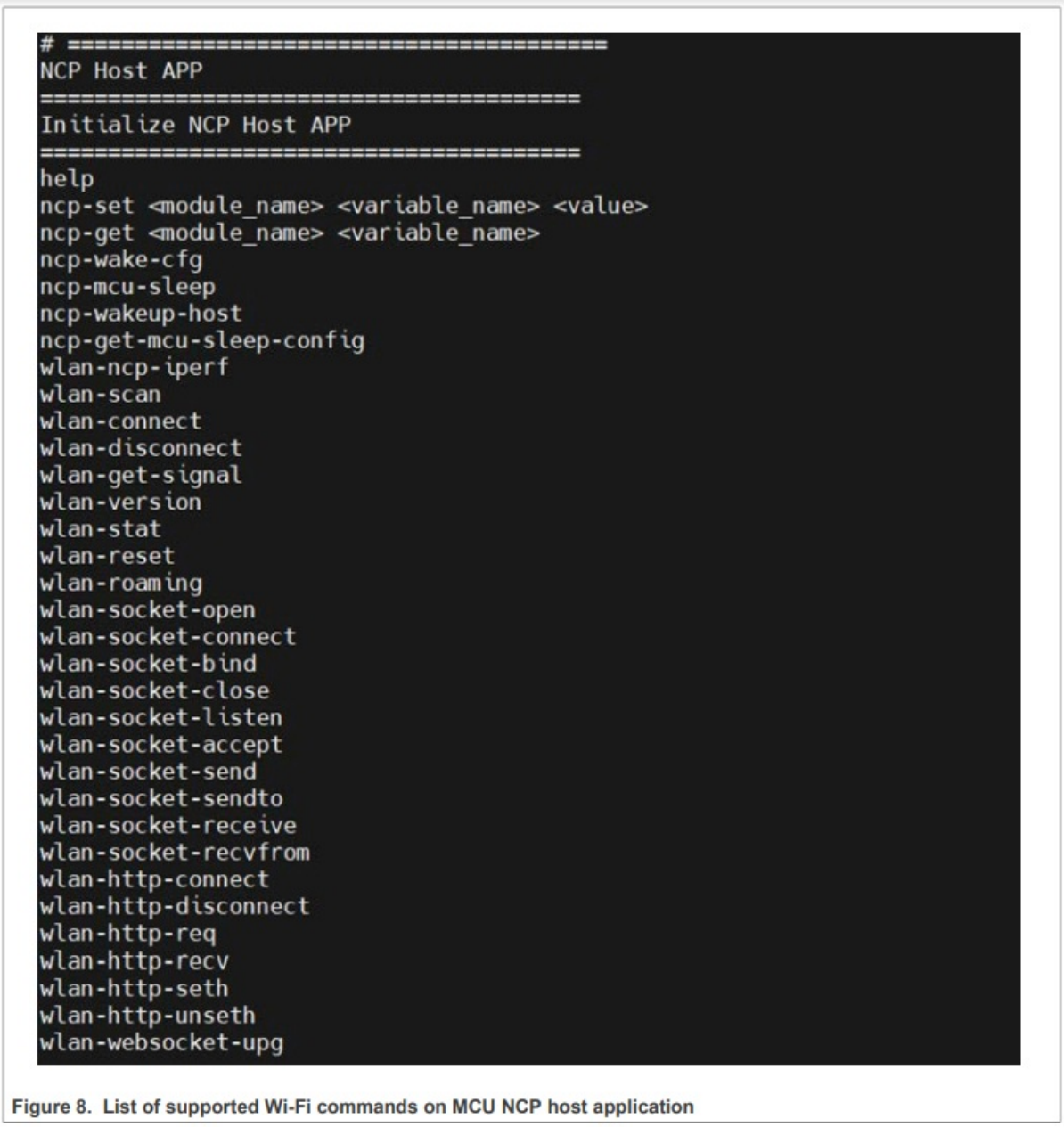


Figure 8. List of supported Wi-Fi commands on MCU NCP host application

Step 6 – Issue the Wi-Fi commands on the NCP host side.

The commands are sent to the NCP device. The command response shows on the NCP host side. Figure 9 shows an example of wlan-version command issued on the NCP host side.

```
# wlan-version

# WLAN Driver Version   :v1.3.r48.p12
WLAN Firmware Version :rw610w-V2, IMU, FP99, 18.99.6.p8, PVE_FIX 1
```

Figure 9. Example of `wlan-version` command output

Bluetooth Low Energy

Step 1 – Load the NCP device image on RW612 board with the specified interface (section 3).

Table 4 shows the load address for Bluetooth LE NCP device application

Table 4. Load address of the images for Bluetooth LE NCP device application

Images	Load address
Bluetooth LE/15.4 combo firmware	0x085e0000
NCP device application binary	0x08000400

Step 2 – Program the NCP host image on i.MX RT1060 board with the specified interface.
Step 3 – Connect RW612 to i.MX RT1060 with the specified interface (section 3).
Step 4 – Power on i.MX RT1060 board and RW612 board.
Step 5 – Run the NCP host application on i.MX RT1060 board and get the list of supported Bluetooth LE commands (Figure 10).

```
=======================================
NCP Host APP
=======================================
Initialize NCP Host APP
=======================================
help
ncp-set <module_name> <variable_name> <value>
ncp-get <module_name> <variable_name>
ncp-wake-cfg
ncp-mcu-sleep
ncp-wakeup-host
ncp-get-mcu-sleep-config
ncp-usb-pm-cfg <1/2>
ble-set-adv-data <adv_data>
ble-start-adv
ble-stop-adv
ble-set-scan-param <filter_option> <interval> <window>
ble-start-scan <scan_type>
ble-stop-scan
ble-connect <addr_type> <addr>
ble-disconnect <addr_type> <addr>
ble-set-data-len <addr_type> <addr> <tx_max_len> [optional<tx_max_time>]
ble-set-phy <addr_type> <addr> <tx_phy> <rx_phy>
ble-conn-param-update <addr_type> <addr> <max_interval> <min_interval> <latency> <timeout>
ble-set-filter-list <filter_addr_num> <addr_type> <addr> ... <addr_type> <addr>
ble-start-encryption <addr_type> <addr>
ble-set-value <uuid_len> <uuid> <value_len> <value ...>
ble-read-characteristic <addr_type> <addr> <handle>
ble-set-power-mode <0/1>
ble-host-svc-add
ble-register-service <num_of_service> <service id_1> <service id_2> ...
ble-start-service <profile_name[hts/htc/hrs/hrc/bas]>
ble-cfg-subscribe <indicate/notify> <addr_type> <addr> <enable[0/1]> <ccc_handle>
ble-l2cap-connect <addr_type> <addr> <psm>
ble-l2cap-disconnect <addr_type> <addr>
ble-l2cap-send <addr_type> <addr> <times>
ble-l2cap-register <psm>
host init finished
=======================================
```

Figure 10. List of supported Bluetooth LE commands on the MCU NCP host application

Thread

Step 1 – Load the NCP device image to RW612 on RW612 board with the specified interface (section 3). Table 5 shows the image load addresses

Table 5. Image load addresss for OT NCP device application

Image	Load address
Bluetooth LE/802.15.4 combo firmware	0x085e0000
NCP device application binary	0x08000400

Step 2 – Program the NCP host image on i.MX RT1060 board with the specified interface.
Step 3 – Connect RW612 to i.MX RT1060 with the specified interface (Section 3.
Step 4 – Power on i.MX RT1060 board and RW612 board.
Step 5 – Run the NCP host application on i.MX RT1060.

- Input help to get the list of supported commands.
- Input version to get the current OpenThread version and check that NCP feature is working (Figure 11).

Figure 11. List of supported Thread commands on MCU NCP host application

Acronyms and abbreviations

Table 6. Acronyms and abbreviations

Acronym Definition

- CITO Controller input target output[1]
- CLI Command line interface
- COTI Controller output target input[1]
- EVK Evaluation kit
- FW Firmware
- HW Hardware
- IDE Integrated development environment
- NCP Network co-processor
- SDK Software development kit
- SW Software

[1] The master/slave replacement in this document follows the recommendation of NXP.

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 2024 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 ANYEXPRESS 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 SUCHDAMAGE.

References

[1] User guide – Getting Started with MCUXpresso SDK for MIMXRT1060-EVKB (available in the docs directory in i.MX RT1060 SDK)
[2] User manual – UM11798 – Getting Started with Wireless on RW61x Evaluation Board Running RTOS (link)
[3] Web page – i.MX-RT1060: Crossover MCU with Arm® Cortex®-M7
[4] Web page – NXP MCUXpresso SDK Builder (link)

Revision history

Table 7. Revision history

Document ID	Release date	Description
UM12133 v.1.0	24 September 2024	• Initial version

Contact us

Please refer to following links for more product details, queries and support.

- Home Page: **nxp.com**
- Web Support: **nxp.com/support**
- NXP Community: **https://community.nxp.com/**

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 the customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards the customer for the products described herein shall be limited in accordance with the Terms and conditions of the 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 the 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 the 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 that is based on any weakness or default in the customer's applications or products, or the application or use by the customer's third-party customer(s). The 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 the 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 the customer.
- Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require 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 the 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. The 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. The customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in the customer's applications. NXP accepts no liability for any vulnerability. Customers 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.

Amazon Web Services, AWS, the Powered by AWS logo, and FreeRTOS — are trademarks of **Amazon.com**, Inc. or its affiliates.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, ULINKPLUS, 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.

Documents / Resources

	**NXP UM12133 Wireless MCU With Integrated** [pdf] User Guide UM12133 Wireless MCU With Integrated, UM12133, Wireless MCU With Integrated, MCU With Integrated, With Integrated, Integrated

References

- Amazon.com
- Support | NXP Semiconductors
- Home - NXP Community
- GitHub - NXP/ot-nxp: OpenThread on NXP examples.
- GitHub - NXP/ot-nxp: OpenThread on NXP examples.
- User Manual

Manuals+, Privacy Policy

This website is an independent publication and is neither affiliated with nor endorsed by any of the trademark owners. The "Bluetooth®" word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. The "Wi-Fi®" word mark and logos are registered trademarks owned by the Wi-Fi Alliance. Any use of these marks on this website does not imply any affiliation with or endorsement.