UM11443

NXP Wi-Fi and Bluetooth Debug Feature Configuration Guide for i.MX RT Platforms

Rev. 8 — 26 June 2024

User manual

Document information

Information	Content
Keywords	Debug configurations, NXP-based wireless modules, i.MX RT1060 EVK board
Abstract	Describes the debug configurations to generate various Wi-Fi driver/feature logs and Bluetooth protocol debugging methods.

1 About this document

1.1 Purpose and scope

This document describes the debug configurations to generate various Wi-Fi driver/feature logs and Bluetooth protocol debugging methods. It details Wi-Fi/Bluetooth sample application using i.MX RT1060 EVK board and NXP-based wireless module for debugging. The purpose of this document is to provide more flexibility to the user for the debug configurations and aims at providing a quick understanding of the debugging techniques.

1.2 Considerations

This document does not include wireless module information, i.MX RT product information, hardware interconnection, board settings, bring-up, IDE setup, SDK download, as these are covered in [1] and [2].

2 Wi-Fi debug features and configurations

This section shows the list of user-configurable Wi-Fi debug macros available in i.MX RT MCUXpresso SDK and how to get different Wi-Fi debug logs based on the features by enabling/defining these macros at the time of application execution. This section also explains the usage of *wifi_cli_fw_dump* application and firmware dump collection in case of Wi-Fi firmware/firmware host communication failure.

2.1 Wi-Fi debug configurations

To enable the debug logs, use the macros listed in the table below along with the source file name. Some of the debug macros are already defined and others can be defined in the header file.

For example, to define `CONFIG_ENABLE_ERROR_LOGS` macro, add the following line in *wifi_config.h* file [3].

```
#define CONFIG_ENABLE_ERROR_LOGS   1
```

Note: *The default values for all debug macros are included in wifi_config_default.h file. To change the values for your configuration, edit the file wifi_config.h.*

Table 1. Wi-Fi debug log configurations

Debug macros	Default macro value	File name	Details
CONFIG_ENABLE_ERROR_LOGS	1	*wifi_config.h*	Enable error logs for Wi-Fi (Includes DHCPD, lwIP, os [port], WLCM, Wi-Fi driver modules)
CONFIG_ENABLE_WARNING_LOGS	1	*wifi_config.h*	Enable warning logs for Wi-Fi (Includes DHCPD, WLCM, Wi-Fi driver modules)
CONFIG_WLCMGR_DEBUG	0	*wifi_config.h*	Enable wireless connection manager debug logs
CONFIG_WIFI_EXTRA_DEBUG	0	*wifi_config.h*	Additional debugging information for the Wi-Fi driver
CONFIG_WIFI_EVENTS_DEBUG	0	*wifi_config.h*	Dump event codes received from the Wi-Fi firmware
CONFIG_WIFI_CMD_RESP_DEBUG	0	*wifi_config.h*	Enable host command and response debug logs (no hex dump)
CONFIG_WIFI_SCAN_DEBUG	0	*wifi_config.h*	Enable scan debug logs
CONFIG_WIFI_IO_INFO_DUMP	0	*wifi_config.h*	Enable information dump about input/output data packets
CONFIG_WIFI_IO_DEBUG	0	*wifi_config.h*	Enable IO debug logs
CONFIG_WIFI_IO_DUMP	0	*wifi_config.h*	Enable SDIO send/receive dump
CONFIG_WIFI_MEM_DEBUG	0	*wifi_config.h*	Enable Wi-Fi module memory related debug logs like allocation and free
CONFIG_WIFI_AMPDU_DEBUG	0	*wifi_config.h*	Enable AMPDU debug level logs
CONFIG_WIFI_TIMER_DEBUG	0	*wifi_config.h*	Enable timer debug level logs
CONFIG_WIFI_SDIO_DEBUG	0	*wifi_config.h*	Enable SDIO debug level logs
CONFIG_WIFI_FW_DEBUG	0	*wifi_config.h*	Enable Wi-Fi Firmware debug logs

2.2 Collect Wi-Fi firmware dump logs using *wifi_cli_fw_dump*

This section describes the use of *wifi_cli_fw_dump* application for which support is enabled on i.MX RT1060 EVK board and explains how to get the Wi-Fi firmware dump in case of Wi-Fi firmware/host communication failure.

This application includes similar commands to *wifi_cli* application. For more details on *wifi_cli* application usage and configuration, refer to [2].

2.2.1 Software download and i.MX RT image setup

Refer to [1] for the SDK download and image setup steps.

2.2.2 Pre-requisites before running the application

Below are the prerequisite to collect the firmware dump:

- Define `CONFIG_WIFI_FW_DEBUG` macro in the *wifi_config.h* header file ([3]).
- Rebuild and flash *wifi_cli_fw_dump* application.
- Plug the USB stick in the USB OTG (J9) slot which is located beside the Ethernet slot on i.MX RT1060 EVK board. Since J9 is a Micro USB slot, use a USB stick with a Micro USB to USB converter. J9 slot is shown in Figure 1.

Note: *Format a USB 2.0 stick as a FatFS disk. Other types like NTFS are not supported.*

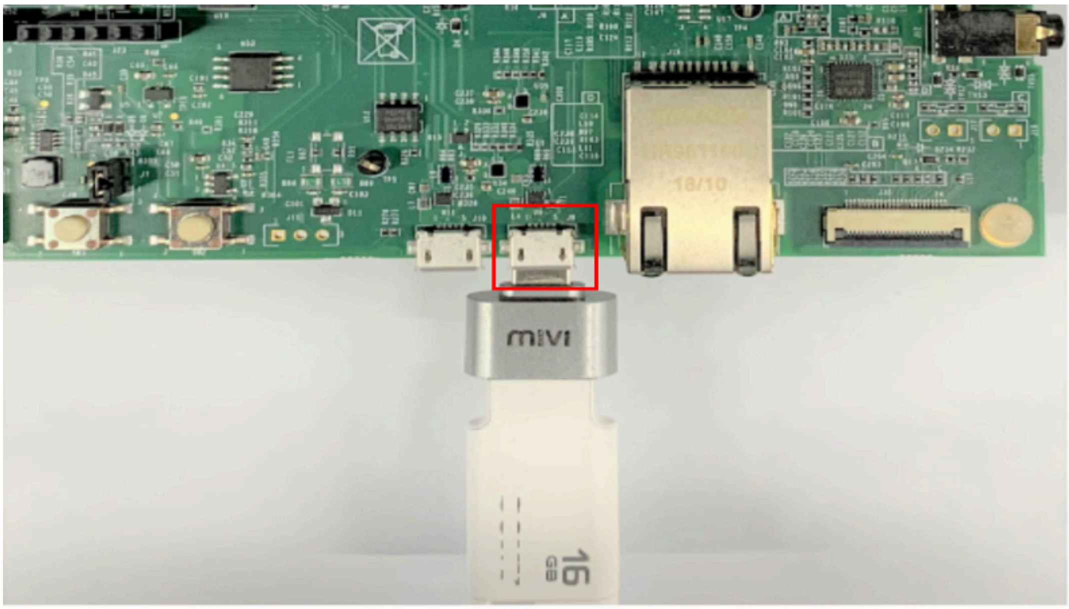


Figure 1. USB stick plugged in i.MXRT1060 EVK board

2.2.3 Run the Wi-Fi demo application

This section describes how to capture the Wi-Fi firmware dump logs and save them on the USB drive plugged into i.MX RT1060 EVK board.

Once the image is flashed on the board, power reset the i.MX RT1060 EVK board and check the console for application start up logs.

Demo start-up logs

The following logs can be observed once the devices—i.MX RT1060 EVK board and NXP-based wireless module—are up and running.

First, the Wireless module firmware is loaded through the SDIO interface. Once the firmware is successfully initialized and loaded, `Wi-Fi MAC address` is printed on the console as shown below.

```
========================================
wifi cli fw dump demo
========================================
Initialize CLI
========================================
Initialize WLAN Driver
========================================
MAC Address: 00:13:43:7F:9C:9F
host init done
[net] Initialized TCP/IP networking stack
========================================
app_cb: WLAN: received event 10
========================================
app_cb: WLAN initialized
========================================
WLAN CLIs are initialized
========================================
CLIs Available:
========================================
help
wlan-version
wlan-mac
wlan-scan
wlan-scan-opt ssid <ssid> bssid ...
wlan-add <profile_name> ssid <ssid> bssid...
wlan-remove <profile_name>
wlan-list
wlan-connect <profile_name>
wlan-start-network <profile_name>
wlan-stop-network
wlan-disconnect
wlan-stat
wlan-info
wlan-address
wlan-get-uap-channel
wlan-get-uap-sta-list
wlan-ieee-ps <0/1>
wlan-deep-sleep-ps <0/1>
ping [-s <packet_size>] [-c <packet_count>] [-W <timeout in sec>] <ip_address>
iperf [-s|-c <host>|-a|-h] [options]
dhcp-stat
========================================
```

Whenever a Wi-Fi firmware or SDIO communication failure occurs, the firmware dump is collected and stored in the USB stick plugged into iMXRT1060 EVK board. This dump is used later to analyze the crashed task call stack.

The reference console log which represents the failure is shown below:

```
# [wifi_io] Error: sdio_drv_write failed (0)
[wifi] SDIO Func0 (0-0x9): 43 03 02 02 03 02 00 02 03 00
[wifi] SDIO Func1 (0x10-0x17): 00 00 00 80 f8 ff ff ff
[wifi] SDIO Func1: (0x8) c3 (0x58) 00 (0x5c) 40 (0x5d) 00 (0x60) 07 (0x61) 0c (0x62) 00
 (0x64) 00 (0x65) 00 (0x66) 00 (0x68) 00 (0x69) 00 (0x6a) 00
[wifi] SDIO Func1 (0xe8-0xf2): dc fe f2 00 a9 00 00 00 00 01 70
[wifi] SDIO Func1 (0xe8-0xf2): dc fe f3 00 aa 00 00 00 00 01 70
............................fatfs test.....................
fatfs mount as logical drive 1......success
[wifi] ==== DEBUG MODE OUTPUT START: 187392904.187392904 ====
[wifi] Start DUMP output 188965782, please wait...
[wifi] ==== DEBUG MODE OUTPUT END: 417214601 ====
```

Note: *Wait until the firmware dump write with the file name FW_DUMP.BIN is fully transferred to the attached USB stick. Upon write completion,* `DEBUG MODE OUTPUT END` *print shows on the console.*

3 Bluetooth debug features and configurations

This section shows the steps to capture HCI logs for Bluetooth using *a2dp_sink* application. The HCI logs are used to analyze the Bluetooth Host and Controller communication. It also provides the steps to extract the link key for the Bluetooth Classic used to decrypt the Bluetooth sniffer logs.

3.1 Bluetooth Classic/Bluetooth LE debug configurations

To enable the debug logs, use the macros listed in the table below along with the source file name.

For example, to define `CONFIG_BT_SNOOP` macro, add the following line in *app_bluetooth_config.h* file [4].

```
#define CONFIG_BT_SNOOP 1
```

Table 2. Bluetooth debug log configurations

Debug macros	Default macro value	File name	Details
CONFIG_BT_SNOOP	Undefined	app_bluetooth_config.h	Enable the HCI logs capturing and store data in USB driver.
CONFIG_BT_DEBUG	Undefined	app_bluetooth_config.h	Enable the debug print feature.
CONFIG_BT_DEBUG_HCI_CORE	Undefined	app_bluetooth_config.h	Enable the debug prints for HCI interface.
CONFIG_BT_DEBUG_CONN	Undefined	app_bluetooth_config.h	Enable the debug prints for connection.
CONFIG_BT_DEBUG_GATT	Undefined	app_bluetooth_config.h	Enable the debug prints for GATT module.
CONFIG_BT_DEBUG_ATT	Undefined	app_bluetooth_config.h	Enable the debug prints for ATT module.
CONFIG_BT_DEBUG_L2CAP	Undefined	app_bluetooth_config.h	Enable the debug prints for L2CAP module.
CONFIG_BT_DEBUG_A2DP	Undefined	app_bluetooth_config.h	Enable the debug prints for A2DP module.
CONFIG_BT_DEBUG_HFP_AG	Undefined	app_bluetooth_config.h	Enable the debug prints for HFP Audio gateway.
CONFIG_BT_DEBUG_HFP_HF	Undefined	app_bluetooth_config.h	Enable the debug prints for HFP device.
CONFIG_BT_DEBUG_SPP	Undefined	app_bluetooth_config.h	Enable the debug prints for SPP
CONFIG_BT_DEBUG_RFCOMM	Undefined	app_bluetooth_config.h	Enable the debug prints for RFCOMM

3.2 Capture and analyze HCI logs using *a2dp_sink*

This section describes the use of *a2dp_sink* application for which support is enabled on i.MX RT1060 EVK board with an NXP-based wireless module and it helps to capture Bluetooth HCI logs. For more details on *a2dp_sink* application usage and configuration, refer to [2].

3.2.1 Software download and i.MX RT image setup

Refer to *UM11441* for the SDK download and image setup steps.

3.2.2 Pre-requisites before running the application

- **Define**`CONFIG_BT_SNOOP` macro in *app_bluetooth_config.h* file [4]. See Section 3.1.
- **Rebuild** and **flash***a2dp_sink* application
- **Connect** the USB Drive
  Plug the USB Drive into the i.MX RT1060 EVK board. See Section 2.2.2.
- **Setup *Wireshark* tool**
  The Wireshark tool is required to open and analyze the HCI logs. Download and install *Wireshark* tool for Windows and Mac OS from here.
  Steps to install *Wireshark* tool on a computer running Linux Ubuntu:

```
sudo add-apt-repository ppa:wireshark-dev/stable
sudo apt update
sudo apt install wireshark
```

3.2.3 Run the Bluetooth demo application

This section describes how to capture the Bluetooth HCI logs saved in the USB drive plugged into i.MX RT1060 EVK board.

Once the image is flashed on the board, power reset the i.MX RT1060 EVK board.

The demo application first loads the Wi-Fi and Bluetooth module firmware through the SDIO interface.

Next, the application automatically turns on the discoverable and connectable mode for Bluetooth Classic.

The following logs can be observed once the i.MX RT EVK board and NXP-based wireless module are up and running.

```
Bluetooth initialized
BR/EDR set connectable and discoverable done
```

Pair a phone with *a2dp_sink*

At this point, the stack is ready to accept incoming connections from any peer device.

Take the mobile phone and use the **Pair new device** option in Bluetooth settings to scan, connect and pair with the i.MX RT1060 EVK and NXP-based wireless module named as *a2dp_sink*.

The following log shows on the console upon the successful Bluetooth connection.

```
Connected
Security changed: 7A:5A:2B:2E:9E:C3 (0xad) level 2
a2dp connected success
```

Disconnect *a2dp_sink* from the phone

The following log shows on the console.

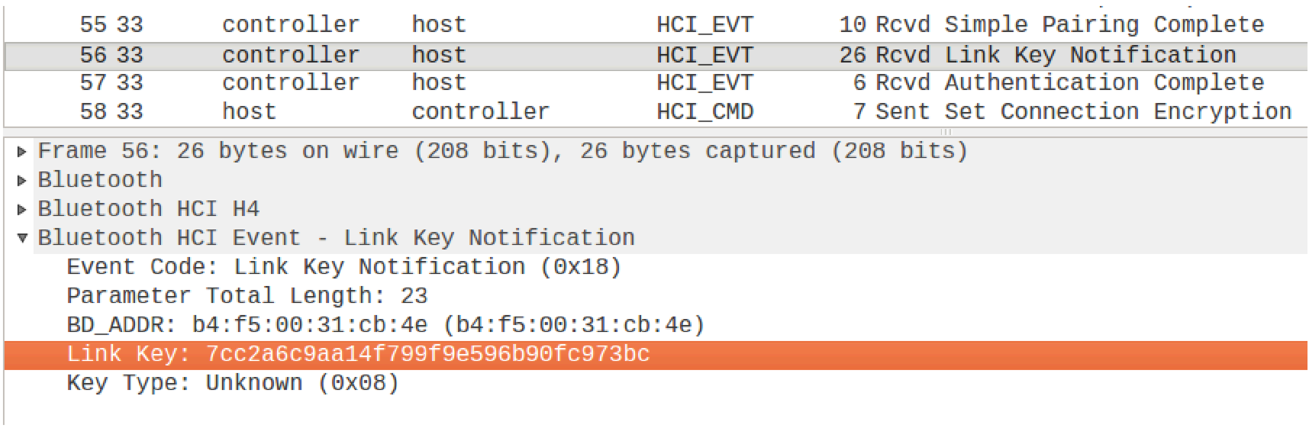
```
Disconnected (reason 0x13)
```

Unplug the USB drive and connect it to the laptop

The file named “*btsnoop*” is available in the USB drive. The *Wireshark* tool can be used to open the file and analyze the logs.

Extract the Link Key for Bluetooth Classic

Open the captured HCI Logs in *Wireshark* tool and search for `Link Key Notification` event. Copy the `Link Key` to use for the sniffer logs decryption.



```
   55 33     controller    host             HCI_EVT      10 Rcvd Simple Pairing Complete
   56 33     controller    host             HCI_EVT      26 Rcvd Link Key Notification
   57 33     controller    host             HCI_EVT       6 Rcvd Authentication Complete
   58 33     host          controller       HCI_CMD       7 Sent Set Connection Encryption
▸ Frame 56: 26 bytes on wire (208 bits), 26 bytes captured (208 bits)
▸ Bluetooth
▸ Bluetooth HCI H4
▾ Bluetooth HCI Event - Link Key Notification
    Event Code: Link Key Notification (0x18)
    Parameter Total Length: 23
    BD_ADDR: b4:f5:00:31:cb:4e (b4:f5:00:31:cb:4e)
    Link Key: 7cc2a6c9aa14f799f9e596b90fc973bc
    Key Type: Unknown (0x08)
```

Figure 2. Copying the `Link Key` for Bluetooth Classic using *Wireshark* tool

4 Acronyms and abbreviations

Table 3. Acronyms and abbreviations

Acronym	Definition
A2DP	Advanced audio distribution profile
AMPDU	Aggregate – MAC protocol data unit
AMSDU	Aggregate – MAC service data unit
AP	Access point
APPL	Application
DHCP	Dynamic host configuration protocol
EVK	Evaluation kit
FW	Firmware
HCI	Host controller interface
IDE	Integrated development environment
IE	Information element
IP	Internet protocol
lwIP	Lightweight IP
OTG	On the go
SD	Secure digital
SDK	Software development kit
STA	Station/client
SW	Software
USB	Universal serial bus
WLAN	Wireless local area network
WLCM	Wireless connection manager
WMM	Wi-Fi multimedia

5 References

[1] User manual - NXP - UM11441 - Getting Started with NXP-based Wireless Modules and i.MX RT Platform Running RTOS (link)

[2] User manual - NXP - UM11442 - Wi-Fi and Bluetooth Demo Applications for i.MX RT Platforms User Guide SDK Documents bundle: *SDK_<version>_EVK-<RT-Platform>\docs\wireless*

[3] Configuration file - NXP – *wifi_config.h*
evk<RT-Platform>wifi_<example>\source\wifi_config.h

[4] Configuration file - NXP – *app_bluetooth_config.h*
evk<RT-Platform>\edgefast_bluetooth<example>\source\app_bluetooth_config.h NXP – wifi_config_default.h

6 Note about the source code in the document

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

7 Revision history

Revision history

Document ID	Date	Description
UM11443 v.8	26 June 2024	• Debug macro configurations restructured • Section 2.1 "Wi-Fi debug configurations": – Added a note about the default values for the debug macros. – Updated the entries for the default macro value in the table. • Section 3.1 "Bluetooth Classic/Bluetooth LE debug configurations": updated the file name of the configuration file. • Section 2.2.2 "Pre-requisites before running the application": updated the file name of the configuration file. • Section 5 "References": updated.
UM11443 v.7	09 January 2024	• Section 6 "Note about the source code in the document": added the section
UM11443 v.6	14 March 2022	• Section 5 "References": updated. • Table 2 "Bluetooth debug log configurations": – Added *CONFIG_BT_DEBUG_SPP* macro – Added *CONFIG_BT_DEBUG_RFCOMM* macro – Removed *CONFIG_WMM* • Section 3.2.3 "Run the Bluetooth demo application": removed the content on demo start-up logs
UM11443 v.5	07 September 2021	• Table 1 "Wi-Fi debug log configurations": added *CONFIG_WMM* debug log • Section 4 "Acronyms and abbreviations": added *WMM* acronym
UM11443 v.4	08 June 2021	• Section 5 "References": updated. • Section 2.2.3 "Run the Wi-Fi demo application": updated the command output example • Section 3.1 "Bluetooth Classic/Bluetooth LE debug configurations": added • Linktext-Section_'number-title: updated
UM11443 v.3	24 March 2021	• Section 2.1 "Wi-Fi debug configurations": updated • Section 2.2.2 "Pre-requisites before running the application": added details about the USB stick format • Section 3 "Bluetooth debug features and configurations": updated
UM11443 v.2	13 January 2021	• Section 1 "About this document": updated • Section 2.1 "Wi-Fi debug configurations": updated • Section 2.2 "Collect Wi-Fi firmware dump logs using wifi_cli_fw_dump": added • Section 3 "Bluetooth debug features and configurations": added • Section 4 "Acronyms and abbreviations": added
UM11443 v.1	17 July 2020	Initial version

8 Contact us

Refer to the 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 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.

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.

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

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.

i.MX — is a trademark of NXP B.V.

Tables

Figures

Contents

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