AN14069

How to Enable Non-XIP Boot on i.MX RT Series EVK Board

Rev. 2 — 12 October 2023 Application note

Document Information

Information	Content
Keywords	Non-XIP, i.MX RT MCUXpresso IDE
Abstract	This application note introduces how to enable non-XIP boot on the EVK board of i.MX RT 4-digit series.

1 Introduction

This application note introduces how to enable non-XIP boot on the EVK board of the i.MX RT 4-digit series.

Non-XIP boot means that the image is stored in Flash, but the code is running in RAM. During the boot, BootROM is responsible for the image copying process. Non-XIP boot is used for NAND devices (FLEXSPI NAND, SEMC NAND, SD, eMMC). FLEXSPI NOR devices can be used for both XIP and non-XIP boot.

This application note takes i.MX RT1170-EVKB as an example. It introduces in detail how to enable the non-XIP boot in MCUXpresso IDE and IAR and use the onboard debugger for attach debugging.

At the end of this application note, it also introduces how to use MCUXpresso to generate the XIP image and link some key codes to Tightly Coupled Memory (TCM).

2 Image layout introduction

The Non-XIP boot must store the image in Flash. Same as the XIP image, the non-XIP image also follows a certain layout to provide some information to the BootROM. The BootROM can copy the image to the specified location according to the information provided.

The Non-XIP image is stored in two kinds of Flash, Nor Flash and Nand Flash. This application note only discusses Nor Flash, which has relatively more application scenarios and can be XIP. The Bootable image of the RT4-digit series has several different structures according to the supported security functions. But for Nor Flash, excluding the security-related parts, the four necessary parts of a Bootable image are FDCB, IVT, BD, and Application Binary. The layout of the Bootable image of RT1170 is shown in Figure 1.

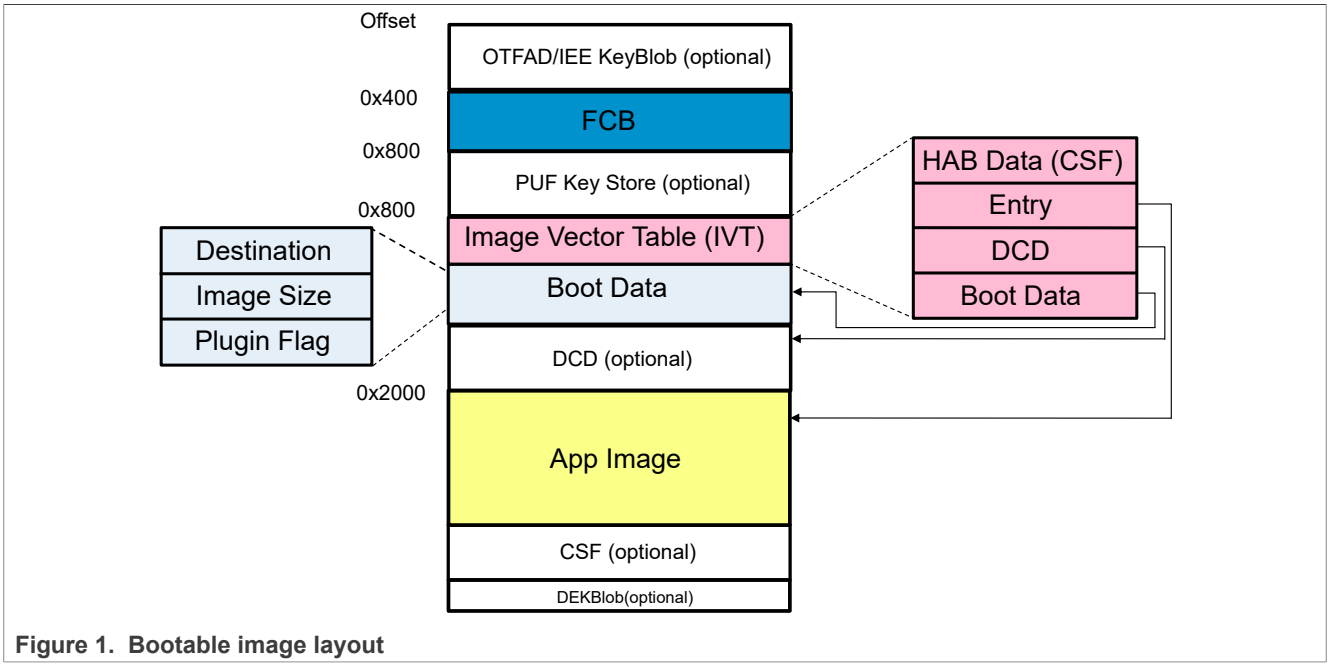

Figure 1. Bootable image layout

The FCB is used to store the specific parameters of the connected Flash. After power-on, the BootROM uses the lower-speed and general-purpose Flash interface controller configuration to access the external Flash. After obtaining the FCB, the BootROM will reconfigure the Flash controller according to the parameters and then use the new configuration to access the Flash.

The IVT records the location information of Boot Data, DCD, Application, and CSF. BootROM obtains the location of each subsection by reading the IVT.

The location of Boot Data follows the IVT and is used to record the start address of the application image and the total length of the image. After the BootROM obtains the information, it can correctly move the image to the specified position of RAM.

3 Use MCUXpresso IDE to enable non-XIP boot and debug

The default configuration of the project in MCUXpresso IDE is to generate the XIP image. So, when debugging the project, it always downloads the image to the Flash. By modifying the link options, it is easy to switch to RAM debug. But to generate a non-XIP image, some additional tools are needed. This chapter introduces how to use the MCUXpresso IDE and MCUBootUtility tool to generate, download, and debug the non-XIP image. The steps are explained below:

1. Import a `hello_world` project in MCUXpresso IDE. As shown in Figure 2, open the **Settings** tab of the project and check the **Link application to RAM** option.

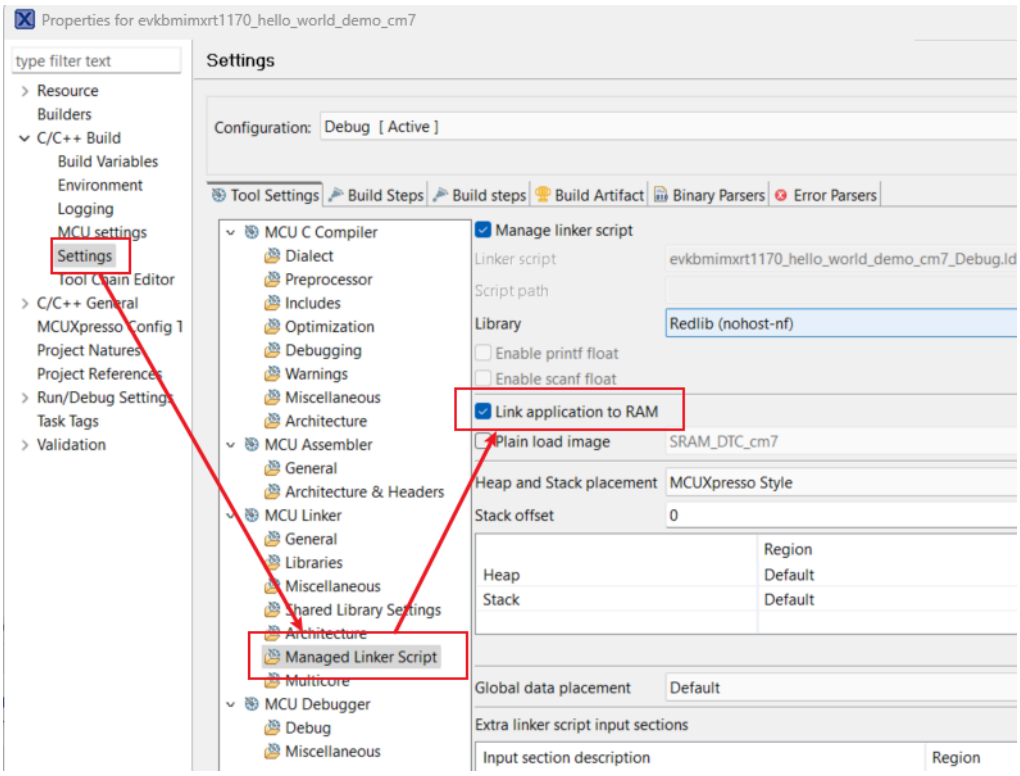

Figure 2. Linker Script configuration

2. Modify the RAM information in the **MCU settings** tab, as shown in Figure 3.

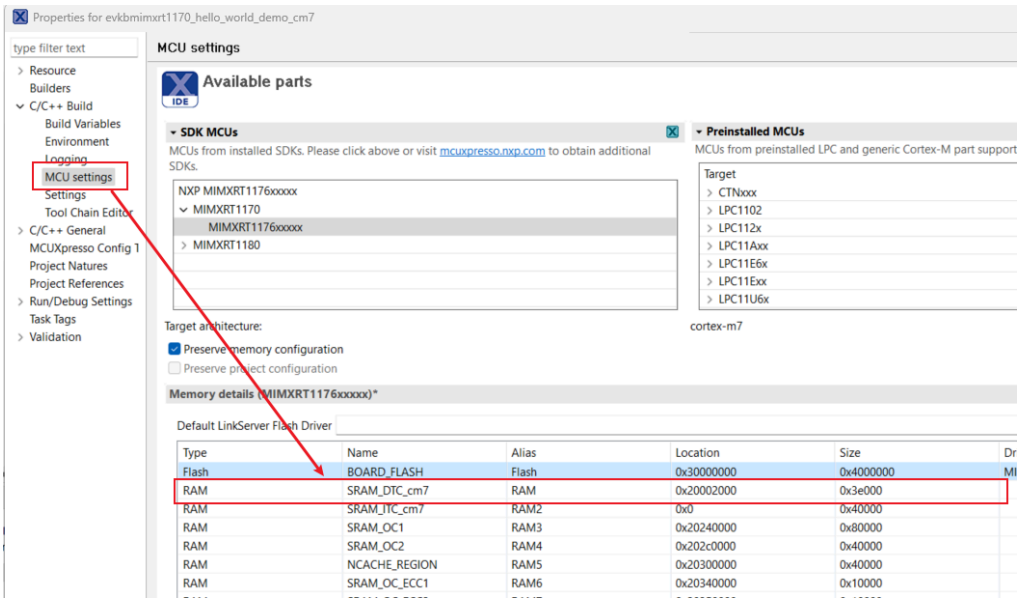

Figure 3. Memory configuration

This example chooses Data Tightly Coupled Memory (DTCM) as the destination that the code section link. Customers can also use Instruction Tightly Coupled Memory (ITCM) or On-Chip RAM (OCRAM) as the link memory of the code. In this configuration, move the corresponding RAM to the lower line of the Flash item. Since the BootROM uses the first 8K of the linked RAM during image copying, in this step, the start address of DTCM is offset by `0x2000`.

3. Compile the project after doing all the steps above, an executable file with the suffix *.axf* is obtained. Currently, open the MCUBootutility tool, dial the RT1170-EVKB to the SDP mode, and connect to the MCUBootUtility. As shown in Figure 4, in the **Application Image File** option, select the *.axf* file generated in MCUXpresso ID, click **All-In-One Action** to generate a Bootable image, and download it to FlexSPI NOR Flash of EVKB.

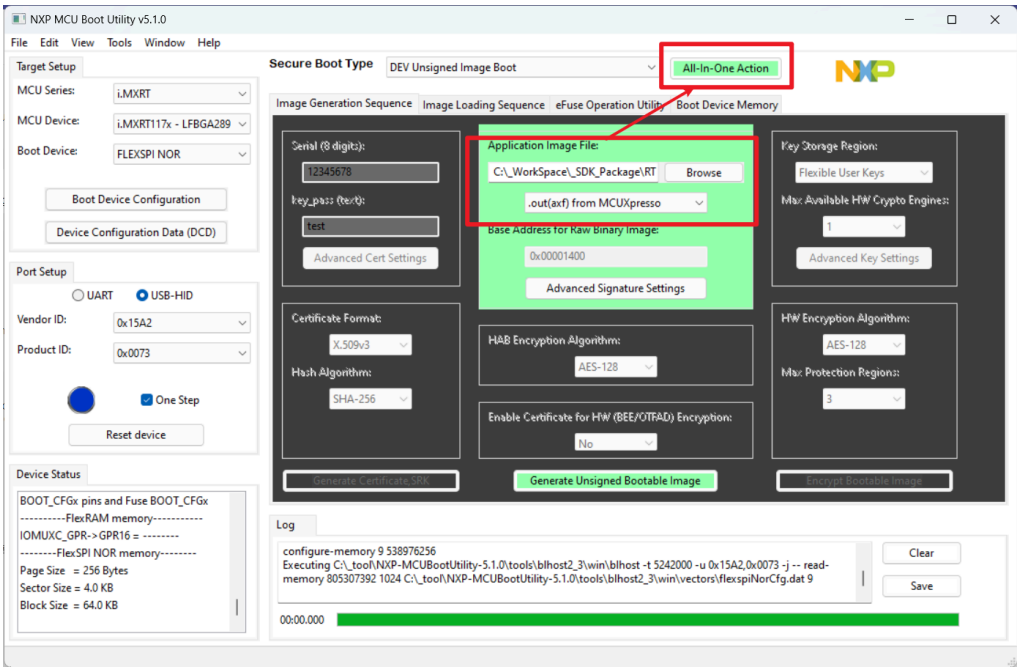

Figure 4. MCUBootUtility configuration

4. Dial the DIP switch of EVKB to FlexSPI NOR Boot mode, and the non-XIP image can boot normally.
5. Go back to the MCUXpresso IDE project, debug once to generate a debug configuration file, and then check the **Attach only** option in the debug configuration, as shown in Figure 5.

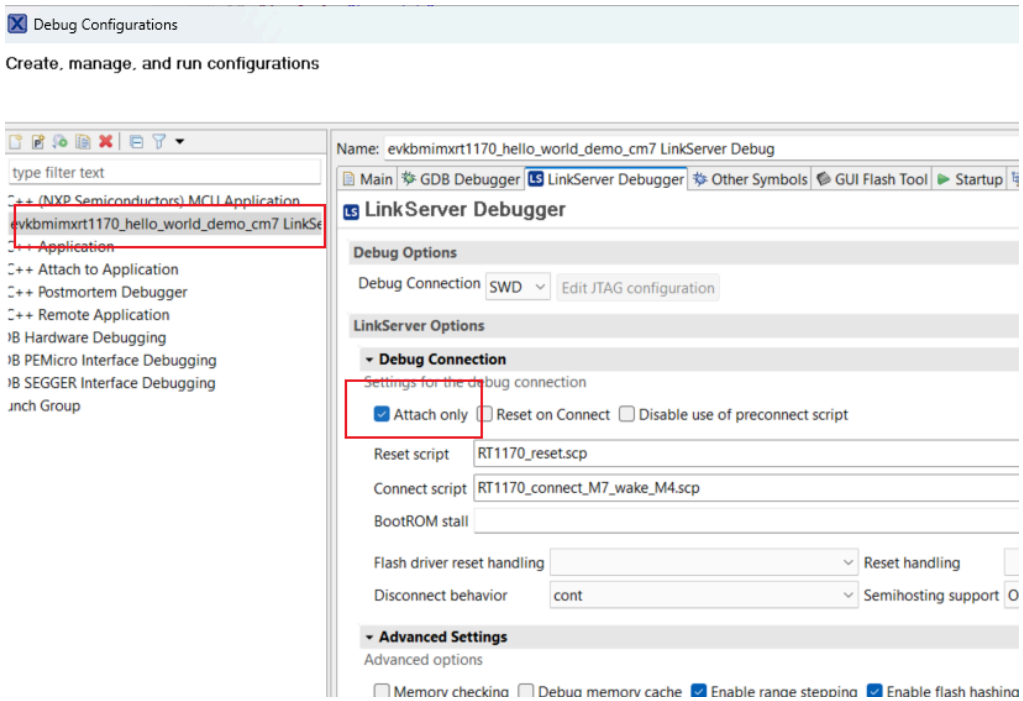

Figure 5. Debug configuration

6. After the Debug configuration is modified, click the **Debug** button to attach to the EVKB that has already run the non-XIP image for debugging.

4 Use IAR to enable non-XIP boot and debug

Compared with MCUXpresso IDE, the steps for IAR to generate the application image are simpler. Taking the debug target as an example, to generate an application image, modify the relevant information in the *.icf* file. The modified *.icf* file is as shown in Figure 6.

```
** ###################################################################
*/

define symbol m_interrupts_start      = 0x00002000;
define symbol m_interrupts_end        = 0x000023FF;

define symbol m_text_start            = 0x00002400;
define symbol m_text_end              = 0x0003FFFF;

define symbol m_data_start            = 0x20000000;
define symbol m_data_end              = 0x2003FFFF;

define symbol m_data2_start           = 0x202C0000;
define symbol m_data2_end             = 0x2033FFFF;

define exported symbol __NCACHE_REGION_START   = m_data2_start;
define exported symbol __NCACHE_REGION_SIZE    = 0x0;
```

Figure 6. IAR linker file modification

After the modification is completed, compile the project to get the executable file. As shown in Figure 4, select the *.out* file generated by IAR in the MCUBootUtility tool, click the **All-In-One Action** button to generate a Bootable image, and download it. Finally, use the **Debug without Downloading** button of IAR to debug.

5 Link part of code to TCM by using MCUXpresso IDE

Except non-XIP boot, this application also introduces another interesting topic. This chapter introduces how to link part of the code to TCM by using the MCUXpresso IDE.

Also import a `hello_world` example project, then add a test file test_ram.c to the project, and create several print functions in `test_ram.c`.

1. Compile the project once to generate related linker files. Then copy the *evkbmimxrt1170_hello_world_demo_cm7_Debug.ld* file in the **Debug** folder and rename it to *evkbmimxrt1170_hello_world_demo_cm7_Debug_test.ld*.
2. Open the configuration of the project, and choose to use the *evkbmimxrt1170_hello_world_demo_cm7_Debug_test.ld* file as the link script according to the steps shown in Figure 7.

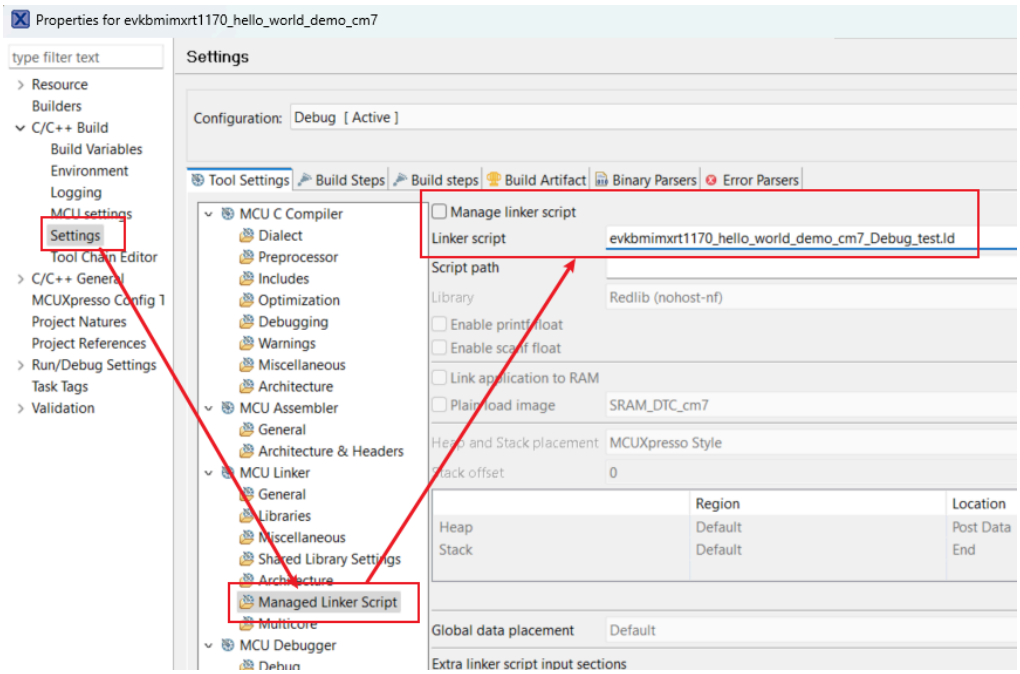

Figure 7. Choose a linker file

3. Open *evkbmimxrt1170_hello_world_demo_cm7_Debug_test.ld* and modify the values, as shown in Figure 8. The purpose of this modification is to link all the functions in *test_ram.c* to ITCM.

```
92          __section_table_end = . ;
93          /* End of Global Section Table */
94
95          *(.after_vectors*)
96
97          *(EXCLUDE_FILE(
98              *test_ram.o
99          ) .text*)                       /* *(.text*)*/
100         *(.rodata .rodata.* .constdata .constdata.*)
101         . = ALIGN(4);
102     } > BOARD_FLASH
103     /*

123     .data_RAM2 : ALIGN(4)
124     {
125         FILL(0xff)
126         PROVIDE(__start_data_RAM2 = .) ;
127         PROVIDE(__start_data_SRAM_ITC_cm7 = .) ;
128         *(.ramfunc.$RAM2)
129         *(.ramfunc.$SRAM_ITC_cm7)
130         *(.data.$RAM2)
131         *(.data.$SRAM_ITC_cm7)
132         *(.data.$RAM2.*)
133         *(.data.$SRAM_ITC_cm7.*)
134         *test_ram.o (.text.*)
135         . = ALIGN(4) ;
136         PROVIDE(__end_data_RAM2 = .) ;
137         PROVIDE(__end_data_SRAM_ITC_cm7 = .) ;
138     } > SRAM_ITC_cm7 AT>BOARD_FLASH
```

Figure 8. Linker file modification

4. After completing the above steps, compile, debug, and then find that all the functions in *test_ram.c* run in ITCM.

6 Note about the source code in the document

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

Copyright 2023 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 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.

7 Revision history

Table 1 summarizes the revisions to this document.

Table 1. Revision history

Revision number	Release date	Description
2	12 October 2023	• Updated Figure 4 • Added Section 6
1	26 September 2023	Initial public release

8 Legal information

8.1 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.

8.2 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 http://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 data sheet 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.

8.3 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.

i.MX — is a trademark of NXP B.V.

Contents

Date of release: 12 October 2023
Document identifier: AN14069