

MCUXSDKPACKSRN

MCUXpresso SDK CMSIS Packs Release Notes

Rev. 0 — 07 August 2020

1 Overview

The MCUXpresso Software Development Kit (SDK) is a comprehensive software enablement package designed to simplify and accelerate application development with NXP's Kinetis, LPC, and i.MX MCUs based on Arm® Cortex®-M cores. The MCUXpresso SDK includes production-grade software with integrated RTOS (optional), integrated stacks and middleware, reference software, and more.

For more information about the MCUXpresso SDK visit the [MCUXpresso SDK homepage](#).

Contents

1	Overview.....	1
2	MCUXpresso SDK CMSIS packs introduction.....	1
3	Development tools.....	2
4	Known issues.....	2

2 MCUXpresso SDK CMSIS packs introduction

It is recommended to check the latest news, descriptions of problems and their solutions, user experiences with MCUX SDK CMSIS packs on the NXP community, in the [MCUXpresso software and tools community space](#).

2.1 MCUXpresso SDK DFPs

A DFP provides minimal device support. Installing a DFP provides enough resources to create, compile, and debug MCUXpresso SDK projects.

The MCUXpresso SDK DFP contains:

- Device header files and system initialization modules
- Startup files
- Linker files
- SVD files
- Flash drivers (for some development tools)
- SDK drivers and utilities
- SDK project templates

The content of the DFP is organized into these CMSIS components:

- Startup, contains a minimal set of sources needed to create and compile MCUXpresso SDK projects
- SDK drivers
- SDK utilities
- SDK CMSIS drivers
- RTE Device SDK project template, contains the *RTE_Device.h* file required by the CMSIS drivers
- Board SDK project template, contains board, clock configuration, peripherals, and pin mux SDK configuration files with general implementation



2.2 MCUXpresso SDK BSPs

A BSP provides support and example projects for NXP evaluation boards.

The MCUXpresso SDK BSP contains:

- Example projects and demo applications
- SDK project templates containing board, clock configuration, peripherals, and pin mux SDK configuration files implemented for the specific board.

3 Development tools

The MCUXpresso SDK CMSIS packs were compiled and tested with these development tools:

- MDK-Arm Microcontroller Development Kit (Keil)® 5.30.0.2
- IAR Embedded Workbench for Arm version 8.50.1

The MCUXpresso SDK CMSIS packs were compiled and tested using the Arm::CMSIS pack version 5.4.0.

4 Known issues

4.1 Not possible to download projects to board in IAR Embedded Workbench for Arm IDE using CMSIS DAP default settings

The MCUXpresso SDK projects created in the IAR Embedded Workbench for Arm IDE cannot be downloaded to the board using the CMSIS DAP debugger default settings. Workaround: Go to **Project > Options > Debugger > Download**, enable **Override default .board file**, and select the appropriate .board file from {Embedded Workbench installation directory} \arm\config\flashloader\NXP\. In **Debugger > CMSIS DAP > Reset**, set the reset source to **Software**.

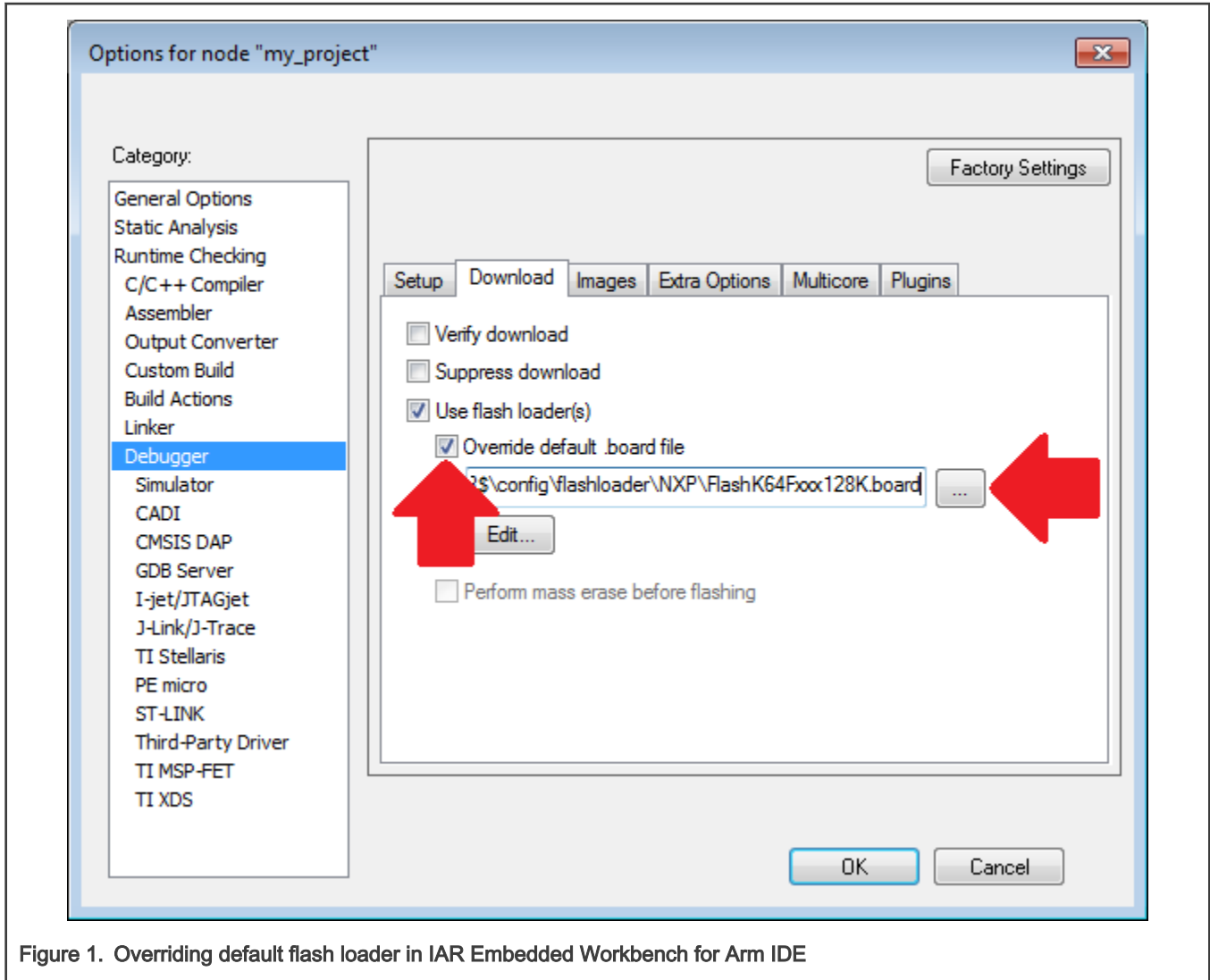


Figure 1. Overriding default flash loader in IAR Embedded Workbench for Arm IDE

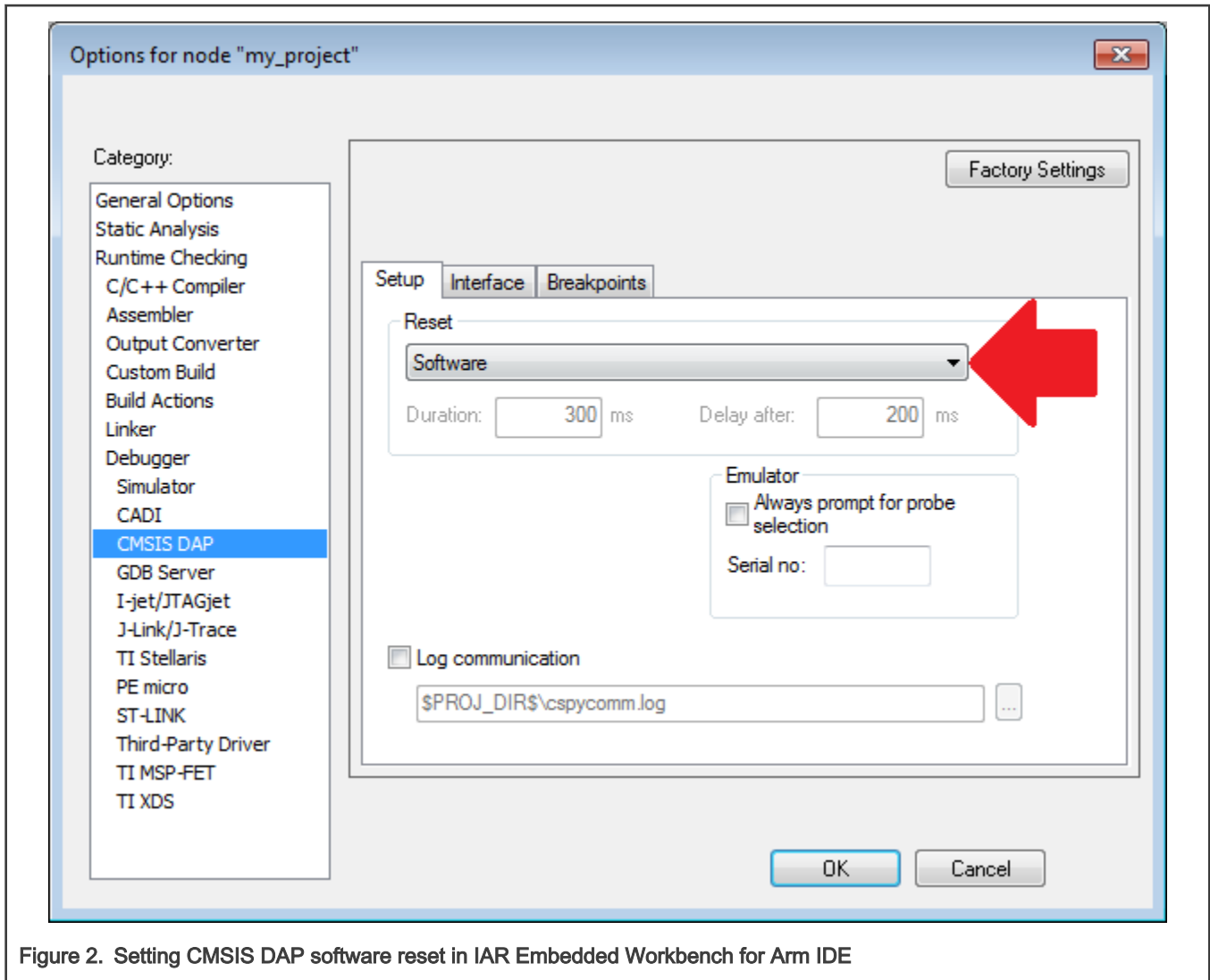


Figure 2. Setting CMSIS DAP software reset in IAR Embedded Workbench for Arm IDE

Alternatively, switch the processor from the **CMSIS-Pack** variant to the **Device** variant in the **Project > Options > General Options > Processor** variant.

4.2 Not possible to debug K64 and K65 projects in IAR Embedded Workbench for Arm using P&E micro

In case of the FRDM-K64F, TWR-K64F120M, and TWR-K65F180M boards, the MCUXpresso SDK projects created in the IAR Embedded Workbench for Arm IDE cannot be debugged using the P&E micro debugger. Workaround: Switch the processor from the **CMSIS-Pack** variant to the **Device** variant in **Project > Options > General Options > Processor** variant.

4.3 Multicore example project build failure in IAR Workbench (SDKGEN-679)

When building the `low_power_cm4` example project on a multicore device, for example, LPCXpresso54114, there is a failure despite having built the `low_power_cm0` project first.

Workaround to this problem

1. Build the project for `cm0plus` core.
2. Build the project for `cm4` core.
3. As there is a wrong path reported for `cm0plus` binary file.

- a. Open project settings.
 - b. Go to **Linker > Raw binary image**.
 - c. Browse binary file (*.bin) from the `cm0plus` project located in `iar\debug` or `iar\release` subfolder.
4. Then build the project for cm4 core again.

4.4 Build failure with default project settings in MDK

Creating a new project using the New project wizard in MDK causes build issues. Newly created projects based on NXP CMSIS packs cannot be built using the default project settings. For example, RT1050. The default settings must be changed to select the scatter file from the NXP CMSIS pack.

To see the default linker settings, see [Figure 3](#)

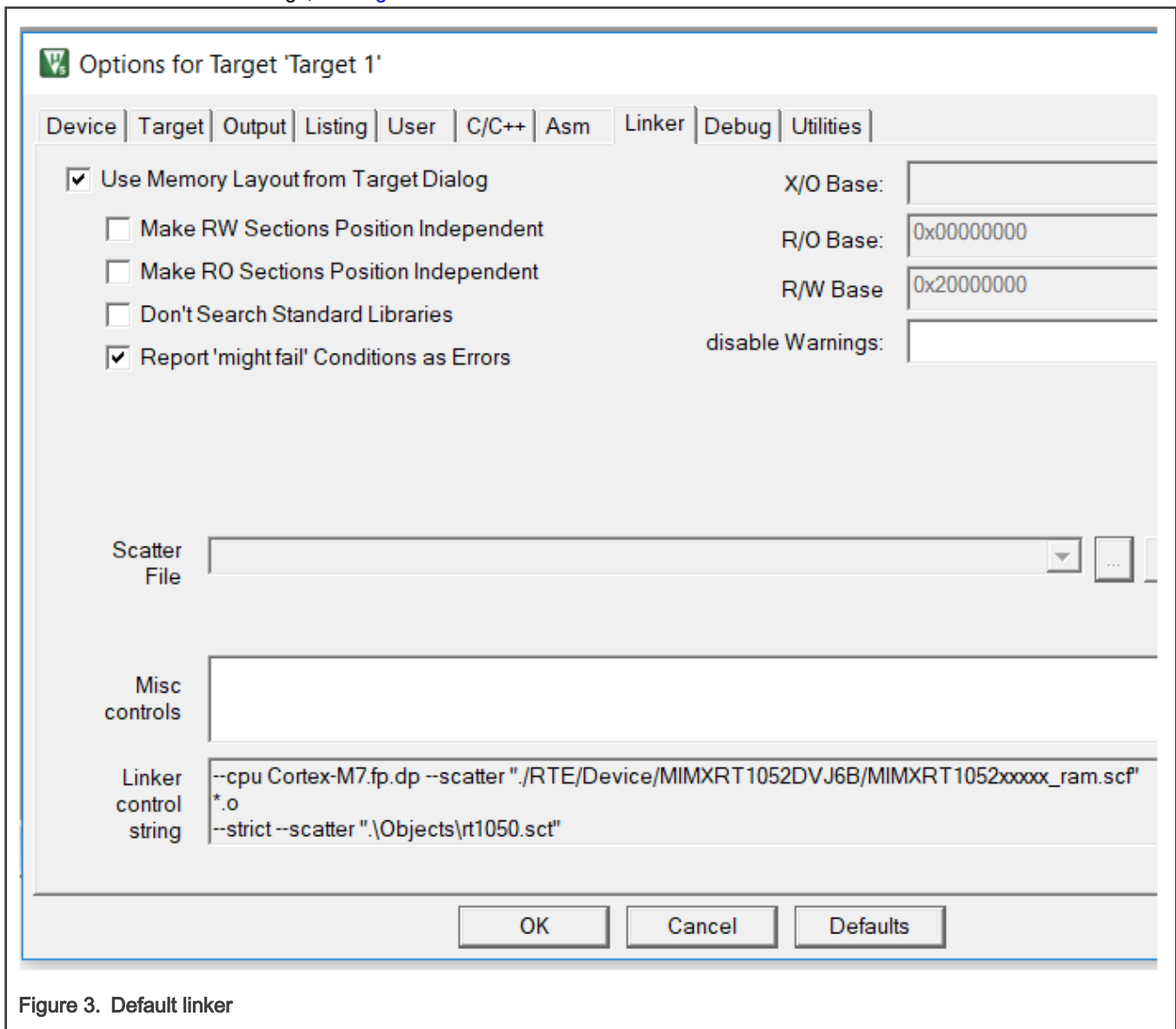


Figure 3. Default linker

To ensure a smooth build process:

1. Uncheck the Use Memory Layout from Target Dialog option in top left corner of the window.
2. Select an appropriate scatter file in the Scatter file selector. For details, see [Figure 4](#).

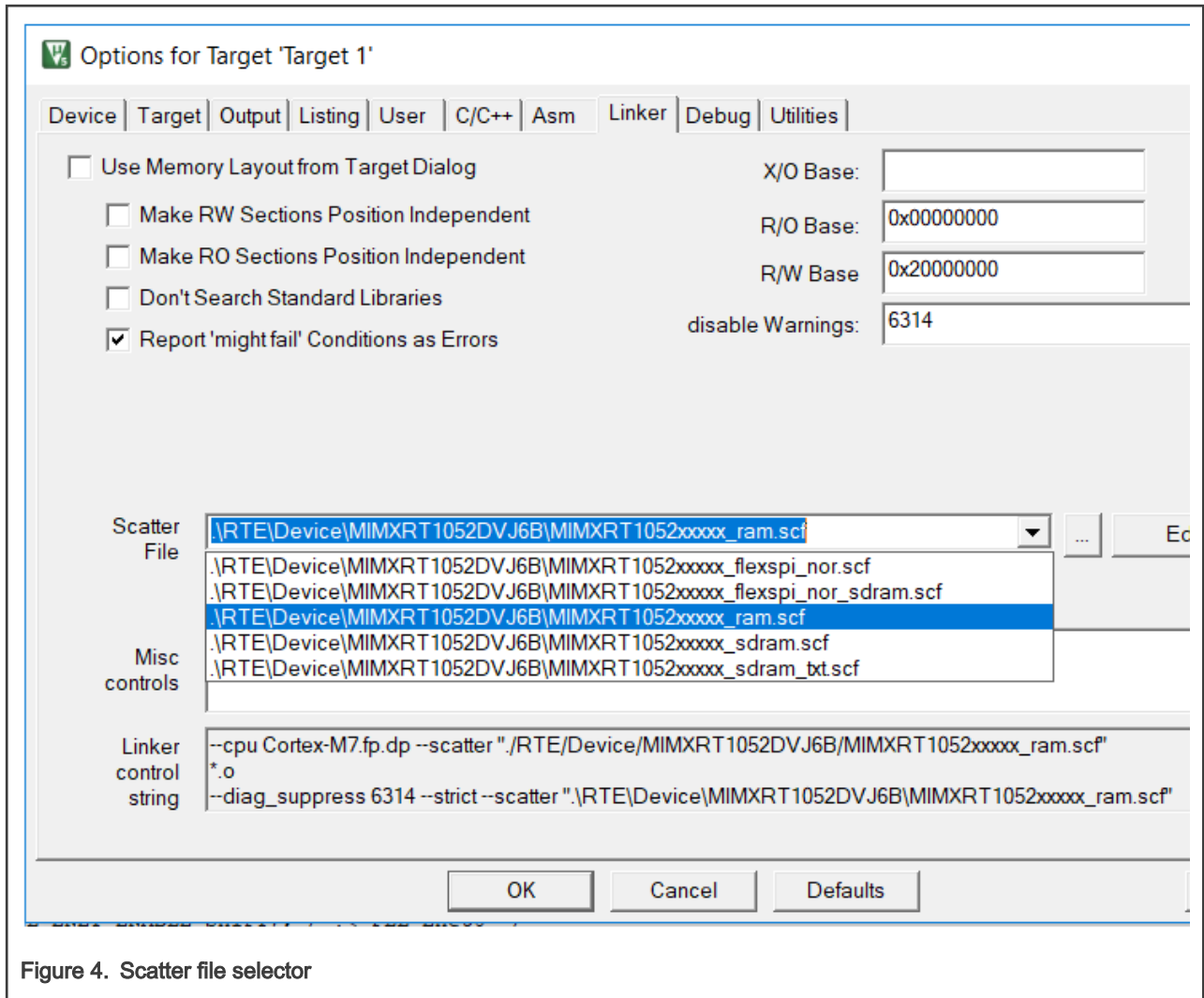


Figure 4. Scatter file selector

4.5 Undefined symbol Image\$\$RW_m_config_text\$\$Base error when building RAM targets of example applications in ARM MDK

You may face this issue while building RAM targets of example applications in ARM MDK.

Undefined symbol Image\$\$RW_m_config_text\$\$Base

If you see this linker error, then manually modify the RTE_Components.h file:

Change values of the these two symbols from to 0:

```
XIP_EXTERNAL_FLASH
XIP_BOOT_HEADER_ENABLE
```

NOTE

Definitions of those two symbols are duplicated in the file.

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μ Vision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) 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. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© NXP B.V. 2020.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 07 August 2020

Document Identifier: MCUXSDKPACKSRN

