

## 1 Overview

The MCUXpresso Software Development Kit (SDK) is a comprehensive software enablement package designed to simplify and accelerate application development with Kinetis, LPC, and i.MX MCUs based on Arm® Cortex-M cores. The MCUXpresso SDK includes production-grade software with integrated Azure RTOS, which is an advanced Industrial Grade Real-Time Operating System (RTOS) from Microsoft designed specifically for deeply embedded, real-time, and IoT applications. It provides advanced scheduling, communication, synchronization, timer, memory management, and interrupt management facilities. Azure RTOS includes some components, for example, ThreadX, FileX, NetX Duo, GUIX, USBX. For more information, please refer to the Azure RTOS Documentation (<https://docs.microsoft.com/en-us/azure/rtos/>).

### Contents

1	Overview.....	1
2	Source description.....	1
3	Azure RTOS example applications .....	2
4	Azure RTOS library projects.....	2
5	Revision history.....	6
6	Revision history.....	6

## 2 Source description

- [Azure RTOS ThreadX](#)
- [Azure RTOS FileX \(LevelX\)](#)
- [Azure RTOS NetX Duo](#)
- [Azure RTOS GUIX](#)
- [Azure RTOS USBX](#)

### 2.1 Azure RTOS ThreadX

Azure RTOS ThreadX provides advanced scheduling, communication, synchronization, timer, memory management, and interrupt management facilities. In addition, it has many advanced features, including its picokernel architecture, preemption-threshold scheduling, event-chaining, execution profiling, performance metrics, and system event tracing.

ThreadX source code is located in `<SDK_DIR>/rtos/azure-rtos/threadx`.

### 2.2 Azure RTOS FileX (LevelX)

Azure RTOS FileX embedded file system is an advanced, industrial grade solution for Microsoft FAT file formats. It supports all the Microsoft file formats, including FAT12, FAT16, FAT32, and exFAT. FileX also offers optional fault tolerance and FLASH wear leveling via an add-on product called Azure RTOS LevelX. Azure RTOS FileX is fast due to direct data write and cache optimized for speed and is easy to use due to consistent API. It includes extensive out of box examples and has been certified to ASIL D and SIL 4.

FileX source code is located in `<SDK_DIR>/rtos/azure-rtos/filex`.

LevelX source code is located in `<SDK_DIR>/rtos/azure-rtos/levelx`.

### 2.3 Azure RTOS NetX Duo

Azure RTOS NetX Duo embedded TCP/IP network stack is an advanced, industrial grade dual IPv4 and IPv6 TCP/IP network stack from Microsoft designed specifically for deeply embedded, real-time, and IoT applications. NetX Duo provides embedded



applications with core network protocols such as IPv4, IPv6, TCP, and UDP as well as a complete suite of additional, higher-level add-on protocols. Azure RTOS NetX Duo is also secure via additional add-on security products, including Azure RTOS NetX Secure IPsec and Azure RTOS NetX Secure SSL/TLS/DTLS.

Azure RTOS NetX Duo provides near wire speed and requires minimal CPU usage and it is designed for performance with zero copy and integrate with hardware features. It also provides the fastest possible UDP processing and provides a BSD-compatible socket interface which helps in migrating existing network application code to NetX Duo. NetX Duo has been certified to ASIL D and SIL 4.

NetX Duo source code is located in `<SDK_DIR>/rtos/azure-rtos/netxduo`

## 2.4 Azure RTOS GUIX

Azure GUIX embedded GUI is an advanced, industrial grade GUI solution from Microsoft designed specifically for deeply embedded, real-time, and IoT applications. Microsoft also provides a full-featured WYSIWYG desktop design tool named Azure RTOS GUIX Studio, which allows developers to design their GUI on the desktop and generate Azure RTOS GUIX embedded GUI code that can then be exported to the target. Azure RTOS GUIX is fully integrated with Azure RTOS ThreadX RTOS and is available for many of the same processors supported by Azure RTOS ThreadX.

The Azure RTOS GUIX package includes various sample user interfaces, including a medical device reference, a smart watch reference, a home automation reference, an industrial control reference, an automotive reference, and various sprite and animation examples.

GUIX source code is located in `<SDK_DIR>/rtos/azure-rtos/guix`.

## 2.5 Azure RTOS USBX

Azure RTOS USBX is a high-performance USB host, device, and on-the-go (OTG) embedded stack. Azure RTOS USBX is fully integrated with Azure RTOS ThreadX and available for all ThreadX-supported processors.

It is designed for speed and has minimal internal function call layering and support for cache and DMA utilization. It has comprehensive class support and device/host controller integration.

USBX source code is located in `<SDK_DIR>/rtos/azure-rtos/usbx`.

## 3 Azure RTOS example applications

The SDK provides a set of Azure RTOS-related applications. The examples are written to demonstrate Azure RTOS features and the interaction between peripheral drivers and the RTOS.

For more information about how to use these example projects, see the Getting Started with MCUXpresso SDK document.

The source code of examples is located in `<SDK_DIR>/boards/<board name>/azure_rtos_examples/`.

## 4 Azure RTOS library projects

To save compile time, the SDK contains pre-built libraries for these Azure RTOS components: **ThreadX**, **FileX**, **GUIX**, **NetXDuo**, **USBX**. Each library has a prefix **lib**, and a suffix **.a** or **.lib**. The libraries are available in the SDK directory, `./rtos/azure-rtos/binary/`.

The default libraries are compiled with double-precision floating-point option. For example, `-mfpu=fpv5-d16`.

The libraries with `_sp` in name are compiled with single-precision floating-point option. For example, `-mfpu=fpv5-sp-d16`.

### NOTE

- LPC5500 series MCU and i.MX RT1010 only support single-precision floating-point.
- There is no pre-built library for ArmGCC projects. The Arm GCC projects generate libraries at runtime.

If header files or source code changes, the libraries must be rebuilt. For example, `tx_user.h`, `fx_user.h`, and `ux_user.h`. These libraries have dependency relationship. For example, **FileX** library depends on **ThreadX** library, **NetX Duo** library depends on **FileX**

library, and **ThreadX** library. So, if one library is updated, other dependent libraries must be updated. For example, if **FileX** library changes **NetX Duo** and **USBX** libraries must be updated.

In MCUXpresso IDE, the library order in the **Libraries** panel is important. For example, the library order in the **azure\_iot\_mqtt** example is **netxduo**, **filex**, and **threadx**. This order is important because **NetX Duo** depends on **FileX** and **ThreadX**, and **FileX** depends on **ThreadX**.

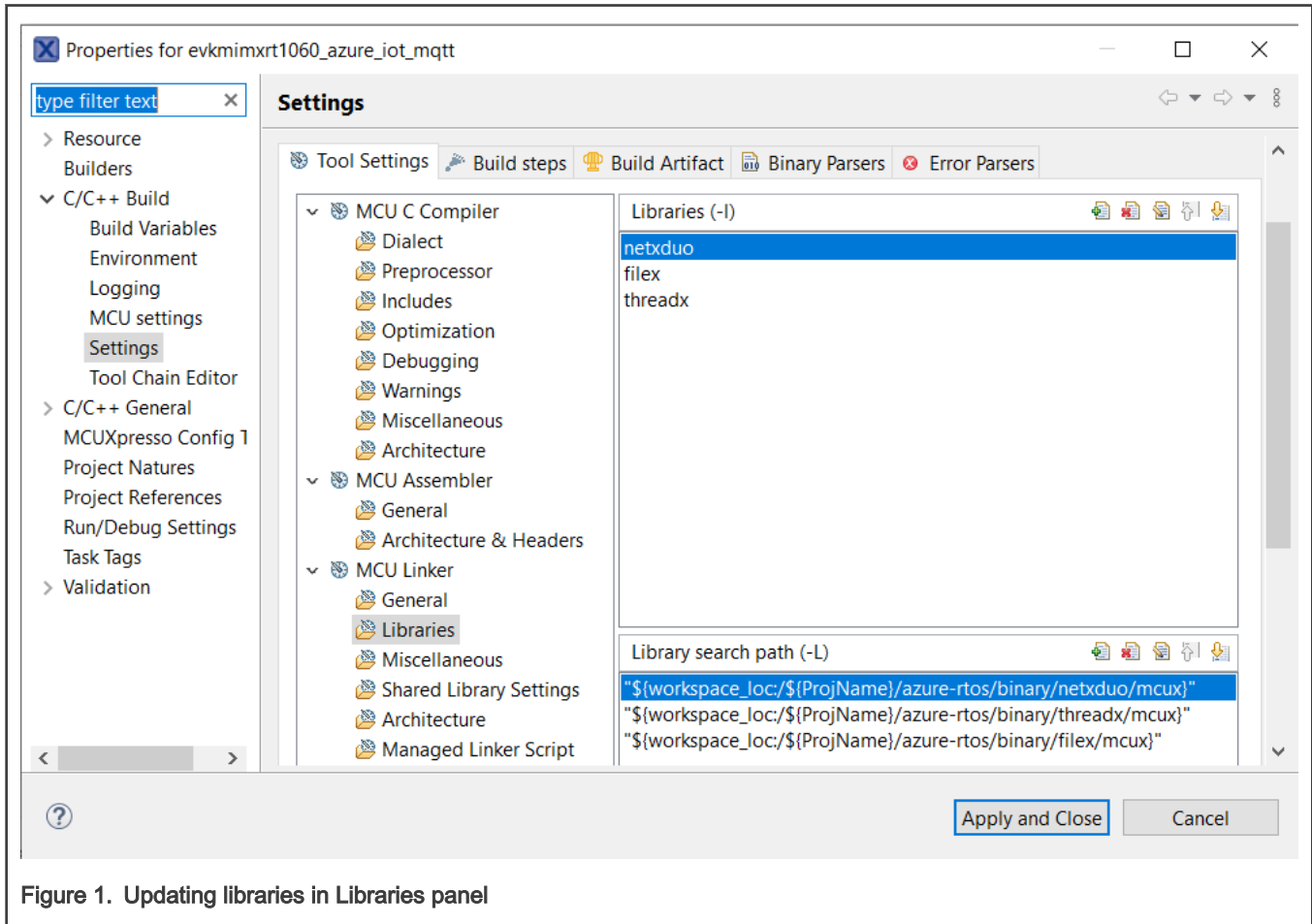


Figure 1. Updating libraries in Libraries panel

### 4.1 Build and use a new library

To build and use a new library in MCUXpresso IDE, perform these steps. Ensure that MCUXpresso SDK with Azure RTOS is pre-installed.

1. On the **Quickstart Panel**, click **Import SDK example(s)**.
2. In the **Import** dialog box, select a board, click **Next**, and then select a library you want to build. For example, **threadx\_lib**.
3. Import the library project.

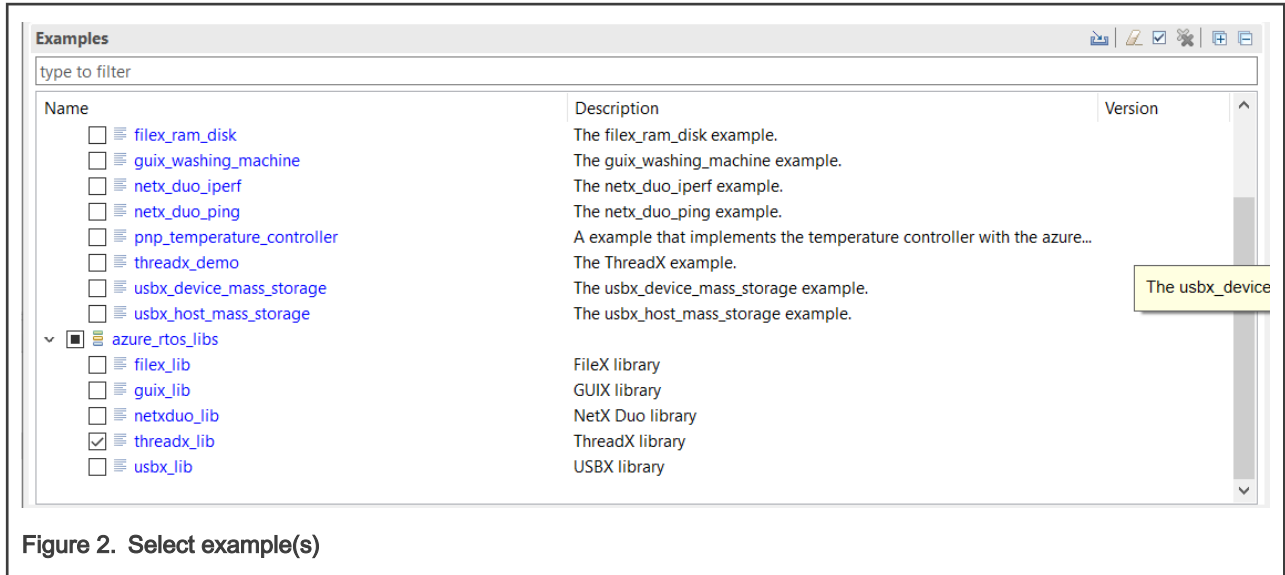


Figure 2. Select example(s)

4. Right-click the threadx\_lib project in the workspace and select **Build Project** to start building.
5. After successful build and compilation, a new library appears in the Debug/Release directory. It has a name like **libevkmimxrt1060\_threadx\_lib.a**.

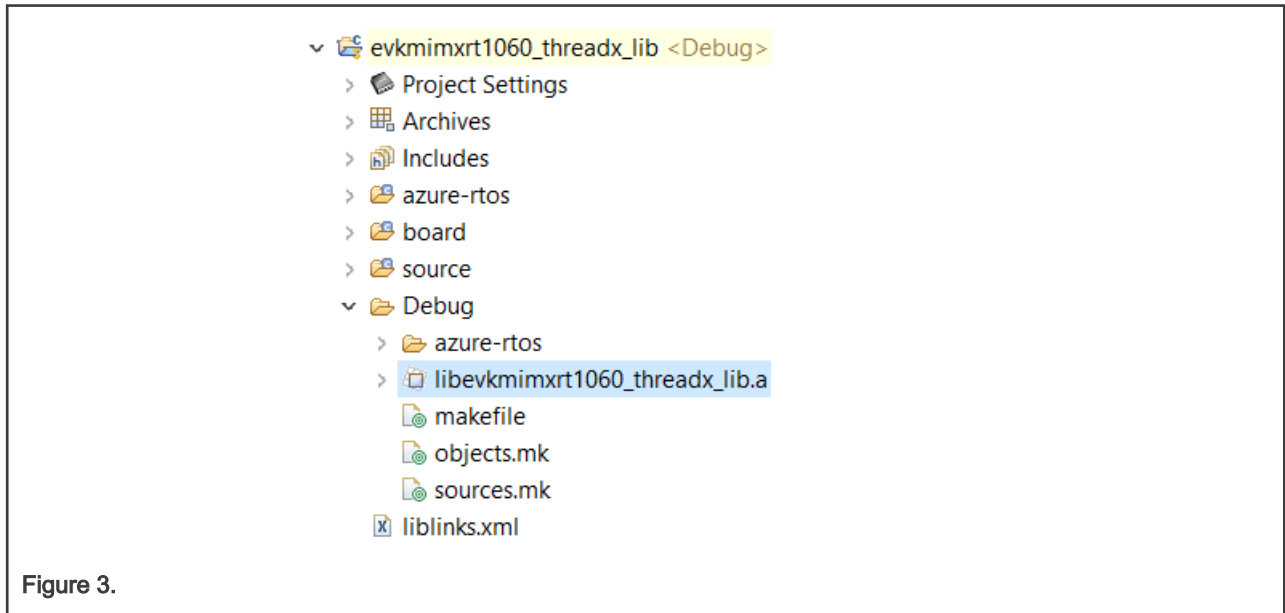


Figure 3.

6. There are two methods to use the new library in an application project. Take the **evkmimxrt1170\_threadx\_demo** project as an example.
  - a. If the new library will not change often, rename it to **libthreadx.a** and copy it to the directory, **azure-rtos/binary/threadx/mcux/**, in the **evkmimxrt1160\_threadx\_demo** project to replace the original one.

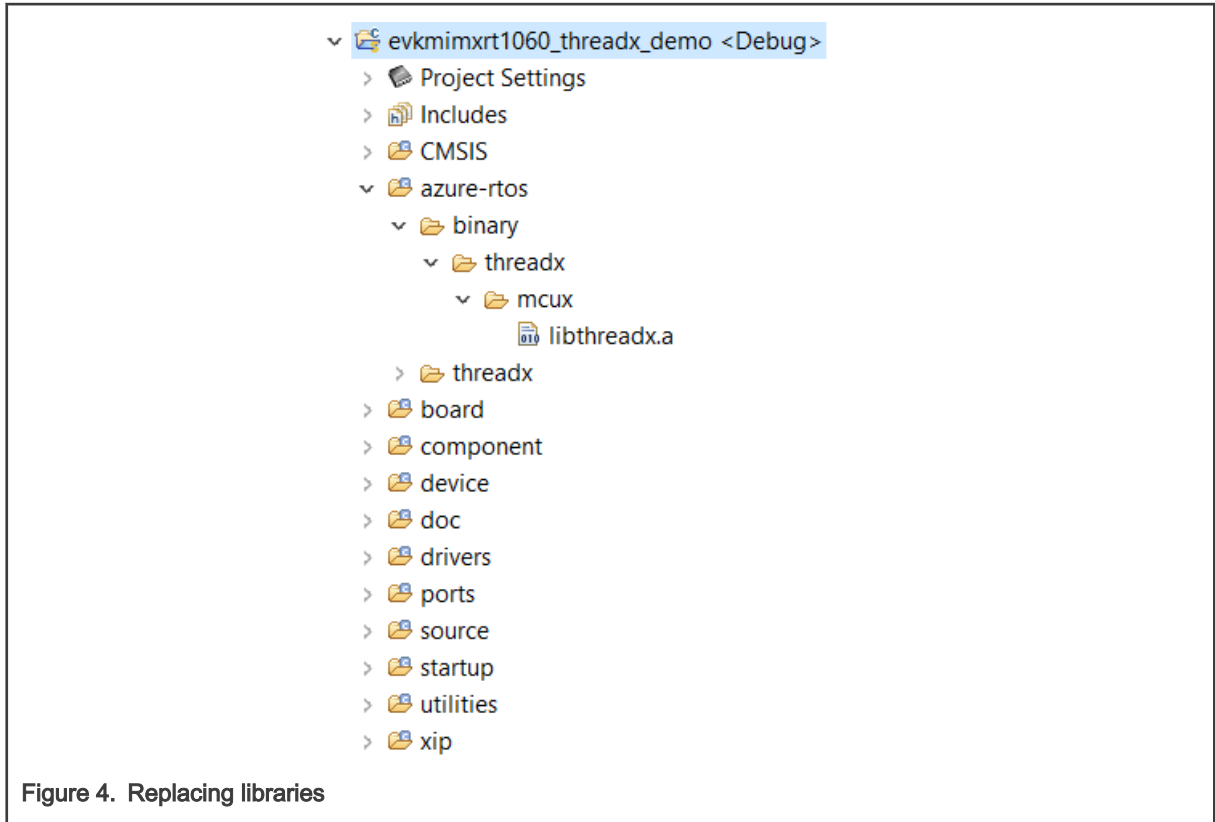


Figure 4. Replacing libraries

- b. If the new library will change frequently, change library settings in application project properties to directly use the new library. In the **Properties** dialog box, change the library name from **threadx** to **evkmimxrt1060\_threadx\_lib** and change **Library search path** to "**\$(workspace\_loc:/evkmimxrt1060\_threadx\_lib/Debug)**".

**NOTE**

Do not change the order of the libraries.

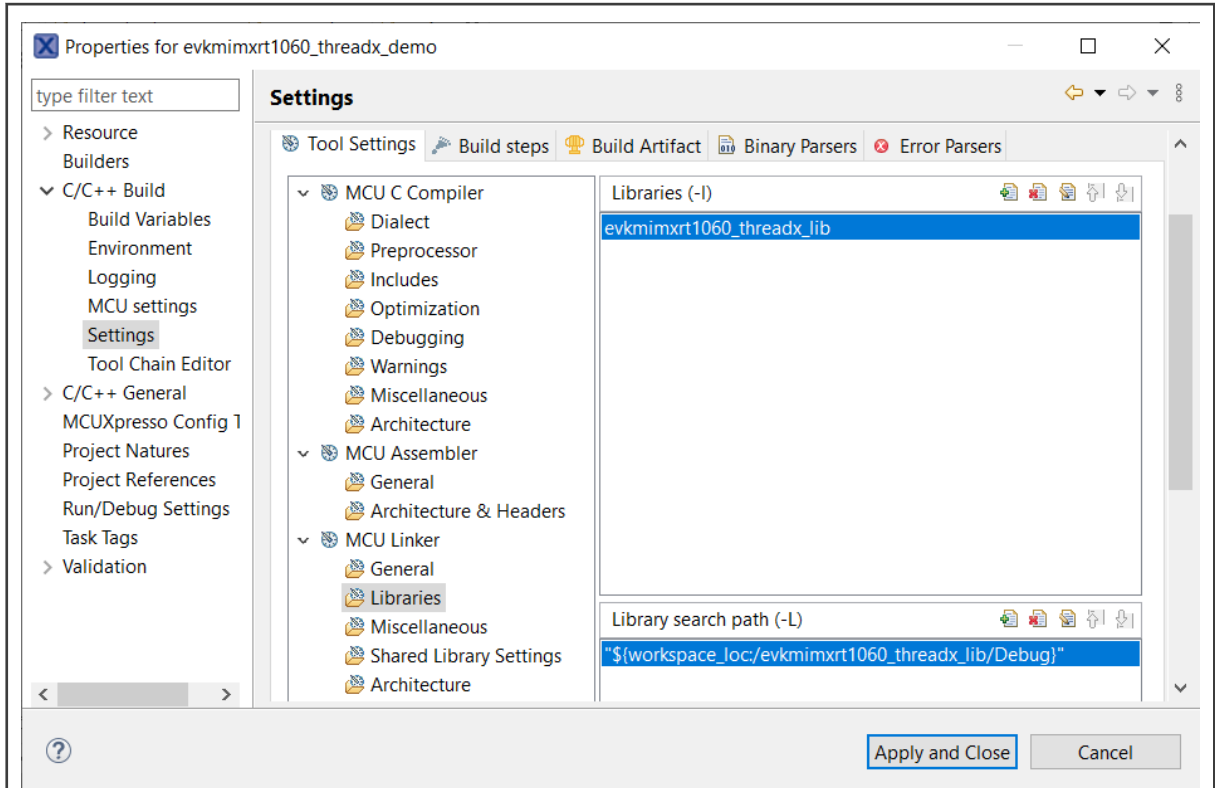


Figure 5. Changing library name

## 5 Revision history

This table summarizes revisions to this document.

Table 1. Revision history

Revision number	Date	Substantive changes
0	20 December 2020	Initial release
2.10.0	10 July 2021	Updated for 2.10.0

**How To Reach Us**

**Home Page:**

[nxp.com](http://nxp.com)

**Web Support:**

[nxp.com/support](http://nxp.com/support)

**Limited warranty and liability** — 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](http://nxp.com/SalesTermsandConditions).

**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.

**Security** — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. 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](mailto:PSIRT@nxp.com)) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP 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,  $\mu$ 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. M, M Mobileye and other Mobileye trademarks or logos appearing herein are trademarks of Mobileye Vision Technologies Ltd. in the United States, the EU and/or other jurisdictions.

© NXP B.V. 2021.

All rights reserved.

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

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

Date of release: 10 July 2021

Document identifier: MCUSDKAZRTOSGSUG

