

MCUXDQS

Quick Start Guide for MCUXpresso Config Tools

Rev. 17.0 — 17 September 2026

User guide

Document information

Information	Content
Keywords	MCUXpresso Config Tools
Abstract	MCUXpresso Configuration Tools set is a suite of evaluation and configuration tools that help you from initial evaluation to production software development.



1 Introduction

MCUXpresso Config Tools is a suite of evaluation and configuration tools that help you from initial evaluation to production software development. With **MCUXpresso Config Tools**, you can configure NXP Cortex-M processors and generate initialization of SDK drivers. The MCUXpresso Config Tools is intended for general use and aims to help hardware designers, software engineers, embedded engineers, and field application engineers (FAEs).

The tool is distributed free of charge. The installer for Windows, Linux, or Mac OS can be downloaded from <https://www.nxp.com/>.

To compile the generated code, an MCUXpresso SDK is required. You can download the SDK either from [MCUXpresso SDK Builder](#) or [MCUXpresso SDK GitHub](#). The MCUXpresso SDK contains many example projects that can help you get started.

Use the link in the installation folder to start the tool. As the first step in the tool, you must create a configuration. The following use cases are described in this document:

- **Starting with SDK example or existing project** - It is useful for new users to start with the MCUXpresso SDK example project for an NXP evaluation board or for users that already have a toolchain project with sources generated by the tool. Supported toolchains are Keil μ Vision, IAR Embedded Workbench, CodeWarrior, Arm GCC and MCUXpresso for Visual Studio Code/SDK West.
- **Starting with new configuration** - It is useful to start with a new configuration for a custom board or with an existing toolchain project that does not contain any tool configuration yet.

2 Create a new configuration

After starting the tool, you can create a new configuration or open an existing configuration from the **Start development** wizard. You can also create a new configuration by selecting **File > New** from the **Menu bar**, or open an existing configuration by selecting **File > Open**.

There are several options for creating a new configuration:

- Create a new configuration and project based on SDK example
- Create a new configuration based on existing toolchain project
- Create a new configuration for Zephyr RTOS
- Create a new standalone configuration for a processor, board or kit

If you start development for an NXP board or kit, we recommend starting by creating a configuration based on an MCUXpresso SDK example or by creating a new standalone configuration for a board or kit. Such configurations contain board-specific settings. If you select a new standalone configuration for a processor, the configuration will be empty. Use a standalone configuration to create a configuration that is not connected to any toolchain project.

After the new configuration is created, you can continue by importing an existing configuration from an MEX file. This is useful if you already have a configuration available or if you want to reuse a previous configuration. To import an existing configuration from an MEX file, select **File > Import... > Import configuration (*.mex)** from the **Menu bar**.

2.1 Cloning an SDK example

You can create a configuration by cloning an SDK example project for IAR Embedded Workbench, Keil μ Vision, CodeWarrior Development Studio, and/or GCC ARM Embedded (command line). The resulting project contains all source files and libraries to build the project and can be easily customized, shared, or put under a control version system.

Note: The creation (cloning) of the projects based on the SDK examples will no longer be supported in the future releases of the MCUXpresso Config Tools. Instead, use the MCUXpresso SDK command line tools and west build system. For detailed documentation, see the [MCUXpresso SDK Command Line Development documentation](#).

SDK example cloning is supported for MCUXpresso SDK 2.2 and higher.

Note: To be able to clone an SDK example or create a “hello_world” project, you must first download an SDK package. For more information about SDK packages offered by NXP Semiconductors, refer to the [MCUXpresso Software Development Kit](#).

Note: If the server is unavailable, and device data is not cached, creating the project fails.

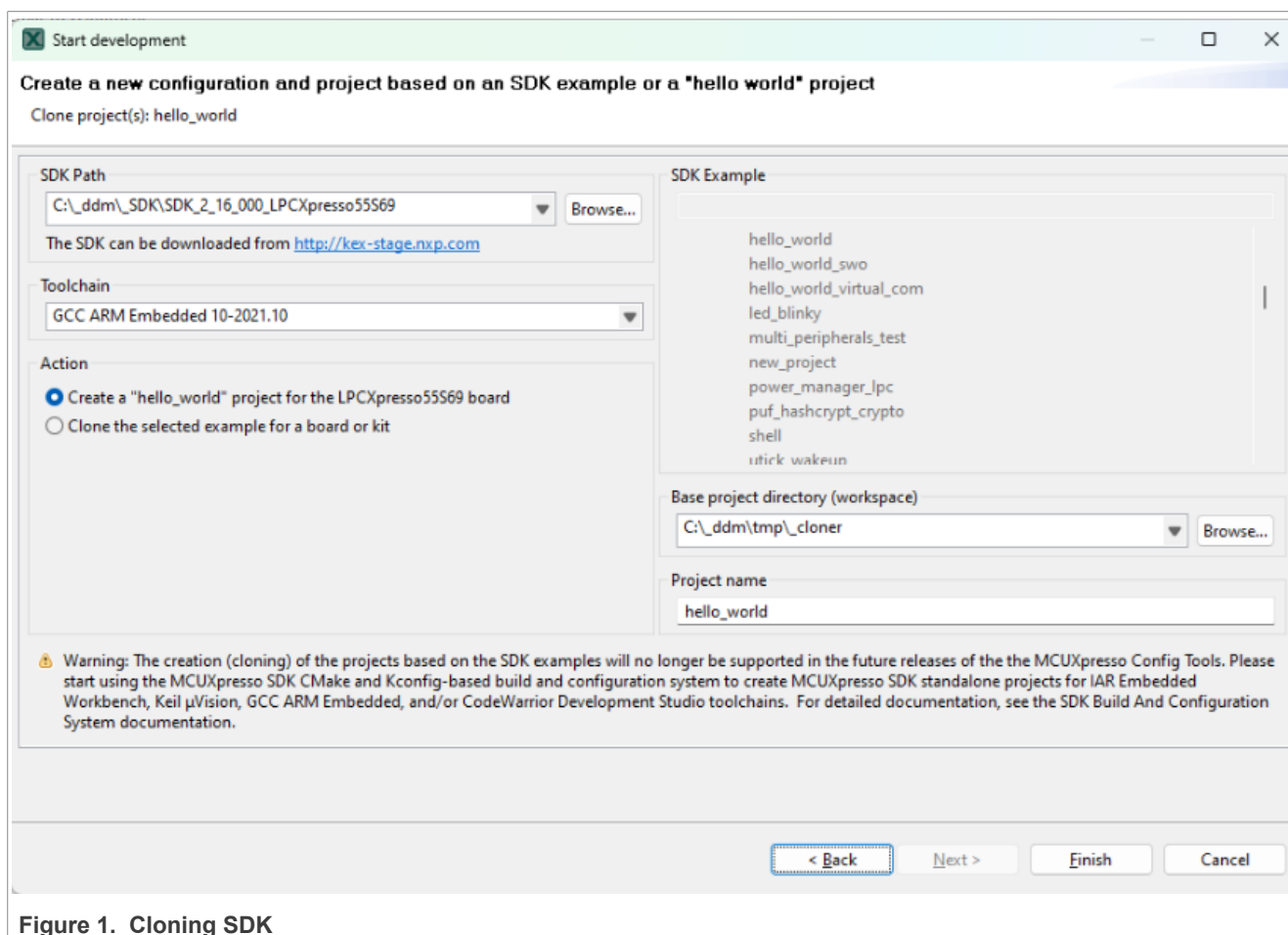


Figure 1. Cloning SDK

To clone an SDK example, do the following:

1. In the **Start development** wizard, select **Create a new configuration based on an SDK example or a “hello world” project**. Alternatively, in the **Menu bar**, select **File > New**.
2. Click **Next**.
3. Specify the path to your locally saved SDK package.
4. Choose the toolchain that you want to create the project for.
5. Choose the SDK example that you want to clone.
6. Specify a base project directory to save your project to.
7. Specify the project name.
8. Click **Finish**.

You can also create a basic, minimally customized “hello_world” project without having to select an SDK example from the package. To create a “hello_world” project, do the following:

1. In the **Start development** wizard, select **Create a new configuration based on an SDK example or a “hello world” project**. Alternatively, in the **Menu bar**, select **File > New**.
2. Click **Next**.
3. Specify the path to your locally saved SDK package.
4. Choose the toolchain that you want to create the project for.
5. Select **Create “hello_world”**.
6. Specify a base project directory to save your project to.
7. Specify the project name.
8. Click **Finish**.

The Config Tools Overview window shows the details of the configuration and supported tools. Now, select a tool by clicking its icon.

2.2 Creating a new toolchain configuration

You can create a configuration for an existing toolchain project. Once done, configuration files associated with the project are updated directly.

MCUXpresso Config Tools currently supports the following toolchains:

- MCUXpresso for VS Code/SDK West
- IAR Embedded Workbench
- Keil MDK uVision
- Arm GCC
- CodeWarrior with SDK

Note: For proper functionality of Config Tools, it is required that the toolchain project originates from the SDK package downloaded from [MCUXpresso SDK Builder](#), has been created using the cloning feature of Config Tools or has been created using [MCUXpresso for VS Code](#) and [MCUXpresso SDK GitHub](#).

To create a configuration based on an existing IDE/Toolchain project, do the following:

1. In the **Start development** wizard, select the **Create a new configuration based on an existing IDE/Toolchain project**. Alternatively, in the **Menu bar**, select **File > New**.
2. Click **Browse**.
3. Select the project file and confirm by clicking **OK**.
4. Click **Finish**.

2.2.1 Creating a new toolchain configuration in MCUXpresso for VS Code

MCUXpresso Config Tools is integrated with the [MCUXpresso for Visual Studio Code](#) extension for a seamless development experience.

MCUXpresso for VS Code supports the following:

- Installation of the MCUXpresso Config Tools using the MCUXpresso Installer.
- Open and load the configuration for the selected MCUXpresso for VS Code project in the MCUXpresso Config Tools.
- Apply changes generated in MCUXpresso Config Tools in the MCUXpresso for VS Code project.

See [Working with MCUXpresso Config Tools](#) in the MCUXpresso for VS Code documentation for detailed guidelines.

2.3 Creating a new standalone configuration

You can create a new configuration that is not part of any toolchain project.

You can later include this configuration in a project by saving the configuration (MEX) file in the toolchain project folder.

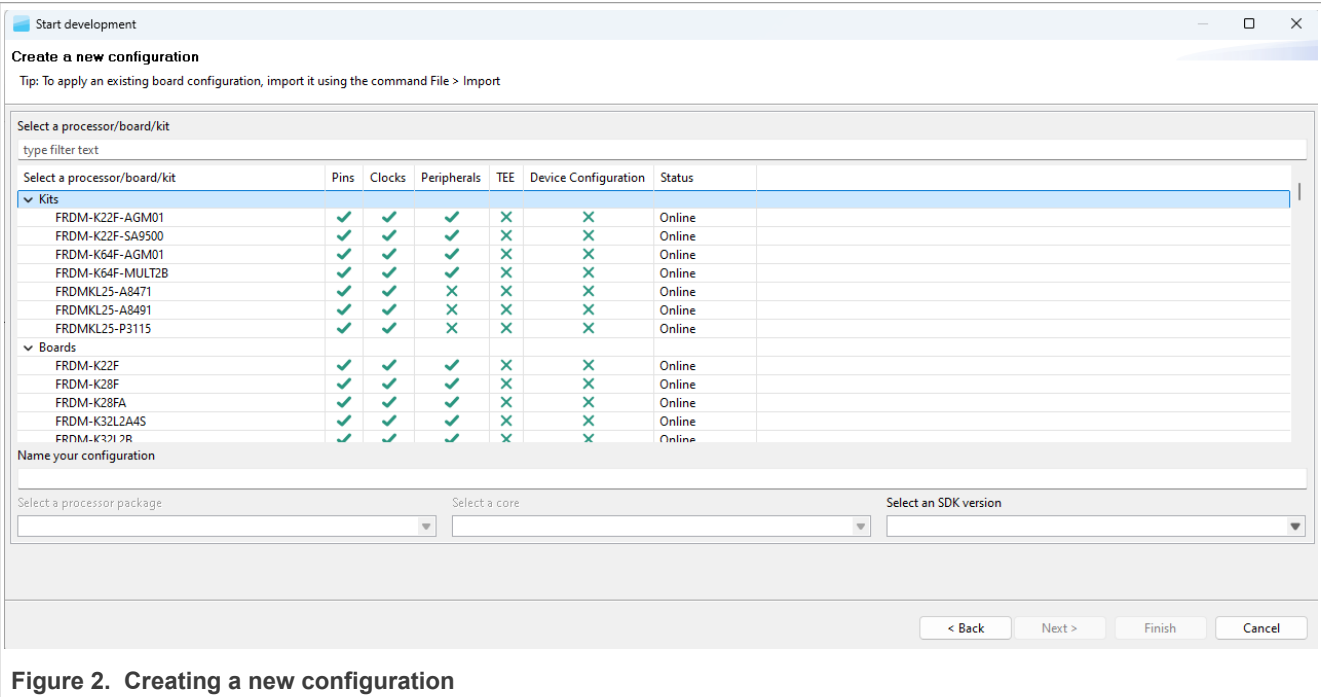


Figure 2. Creating a new configuration

- To create a standalone configuration, do the following:
1. In the **Start development** wizard select **Create a new standalone configuration for processor, board, or kit**. Alternatively, in the **Menu bar**, select **File > New**.
 2. Click **Next**.
 3. Select the processor, board, or kit from the list.
Note: If you are working offline, you will only see locally saved options. For more information, see the **Working offline** section of *User Guide for MCUXpresso Config Tools (Desktop)* (document [GSMCUXCTUG](#))
 4. Name your configuration. Optionally, you can select processor package, core, and SDK version.
 5. Click **Finish**.

3 Select tool

The right-hand side of the toolbar contains tool-selection buttons. Use these buttons to switch between the tools.

Table 1. Tools

Name	Description
Pins Tool	Enables you to configure the pins of a device. Pins tool enables you to create, inspect, change, and modify any aspect of the pin configuration and muxing of the device.
Clocks Tool	Enables you to configure initialization of the system clock (core, system, bus, and peripheral clocks) and generates the

Table 1. Tools...continued

Name	Description
	C code with clock initialization functions and configuration structures.
Peripherals Tool	Enables you to configure the initialization for the MCUXpresso SDK drivers.
Device Configuration Tool	Enables you to generate a Device Configuration Data (DCD) image using the format and constraints specified in the boot ROM reference manual.
TEE (Trusted Execution Environment) Tool	Enables you to configure the security policies of memory areas, bus masters, and peripherals, to isolate and safeguard sensitive areas of your application.

3.1 Pins Tool

In the **Pins** tool, you can display and configure the pins of the processor. Basic configuration can be done in the **Pins**, **Peripheral Signals**, or **Package** views.

More advanced settings (pin electrical features) can be adjusted in the **Routed Pins view**.

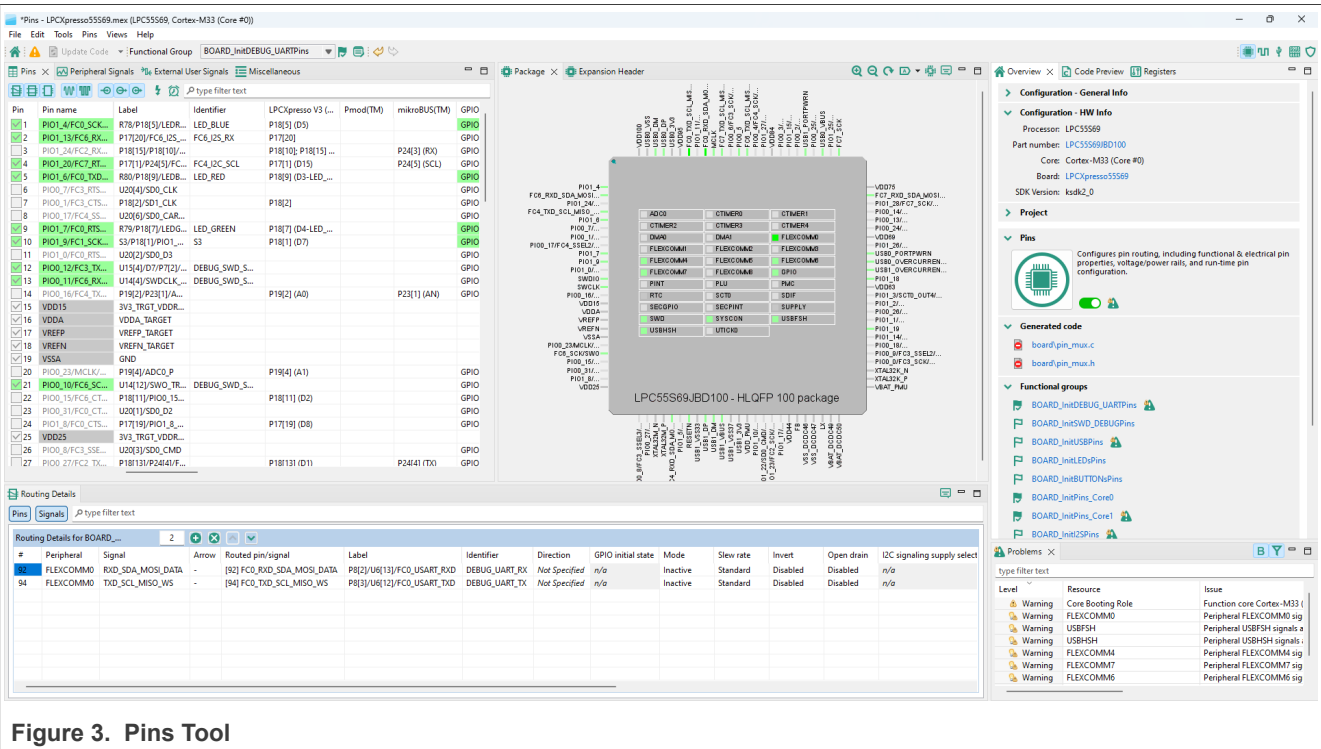


Figure 3. Pins Tool

3.2 Clocks Tool

In the **Clocks** tool, you can display and modify clock sources and outputs. Use the **Clocks Table** view to set clock sources and outputs, More advanced settings can be adjusted via the **Clocks Diagram** and **Details** views. Global settings of the clocking environment such as run modes, MCG modes, and SCG modes can be modified in the **Clocks Table**, **Clocks Diagram**, in **Functional group properties**, and in the **Details** view.

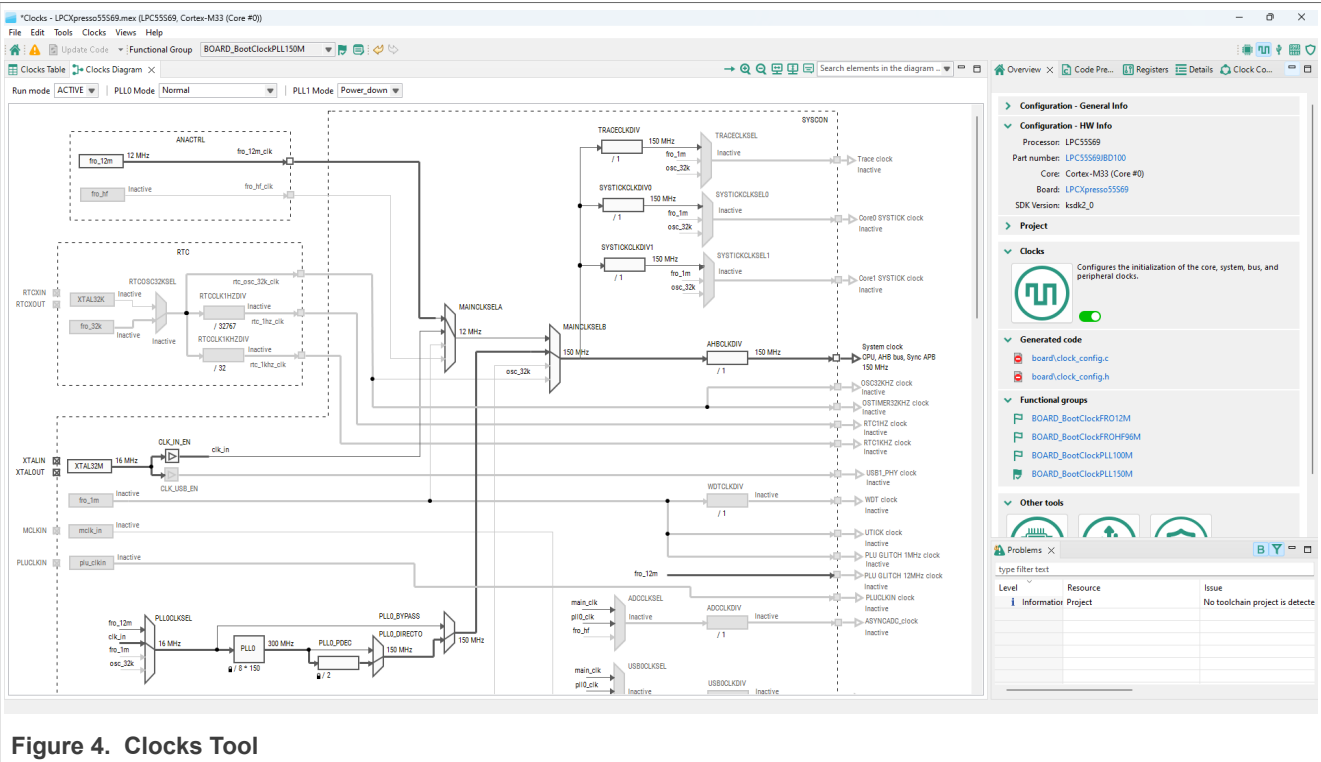


Figure 4. Clocks Tool

3.3 Peripherals Tool

In the **Peripherals** tool, you can configure the initialization of selected peripherals and generate code for them. In the **Peripherals** view, select the peripherals to configure and confirm the addition of the configuration component. Then you can select the mode of the peripheral and configure the settings within the settings editor.

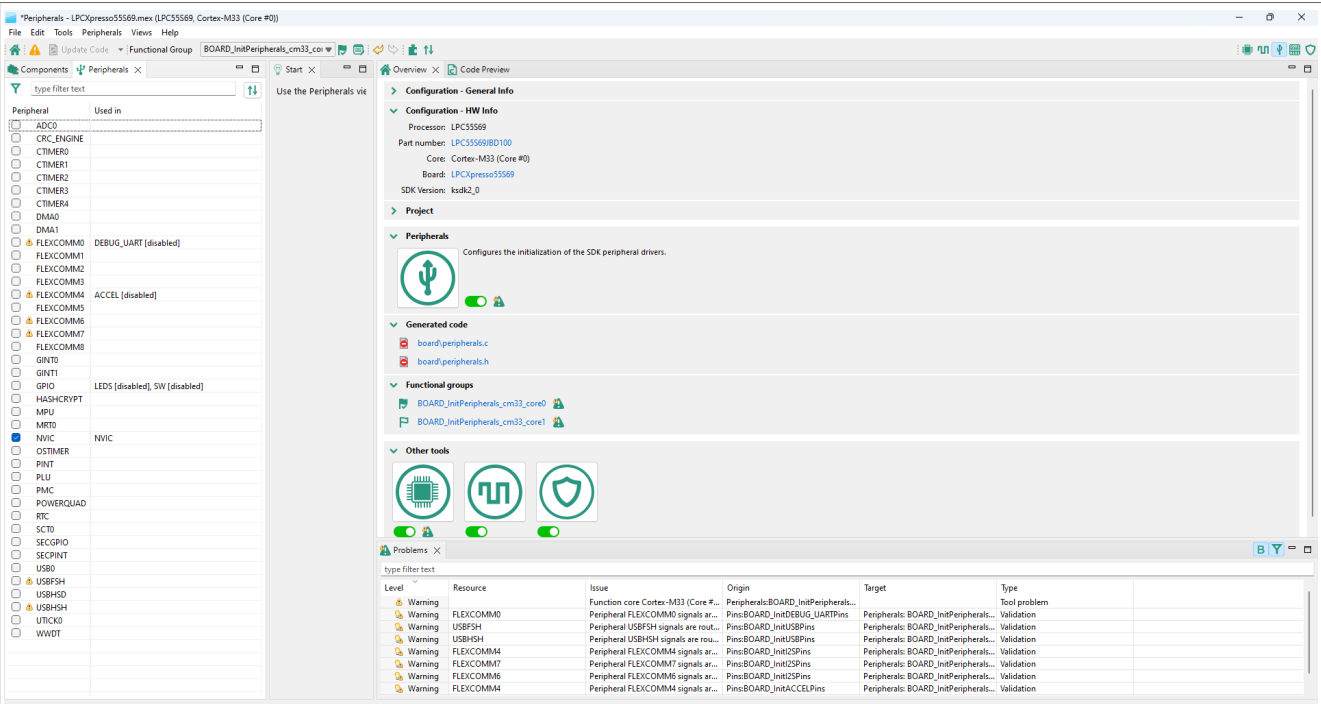


Figure 5. Peripherals Tool

3.4 Device Configuration Tool

The **Device Configuration** tool allows you to configure the initialization of memory interfaces of your hardware. Use the **Device Configuration Data (DCD)** view to create different types of commands and specify their sequence, define their address, values, sizes, and polls.

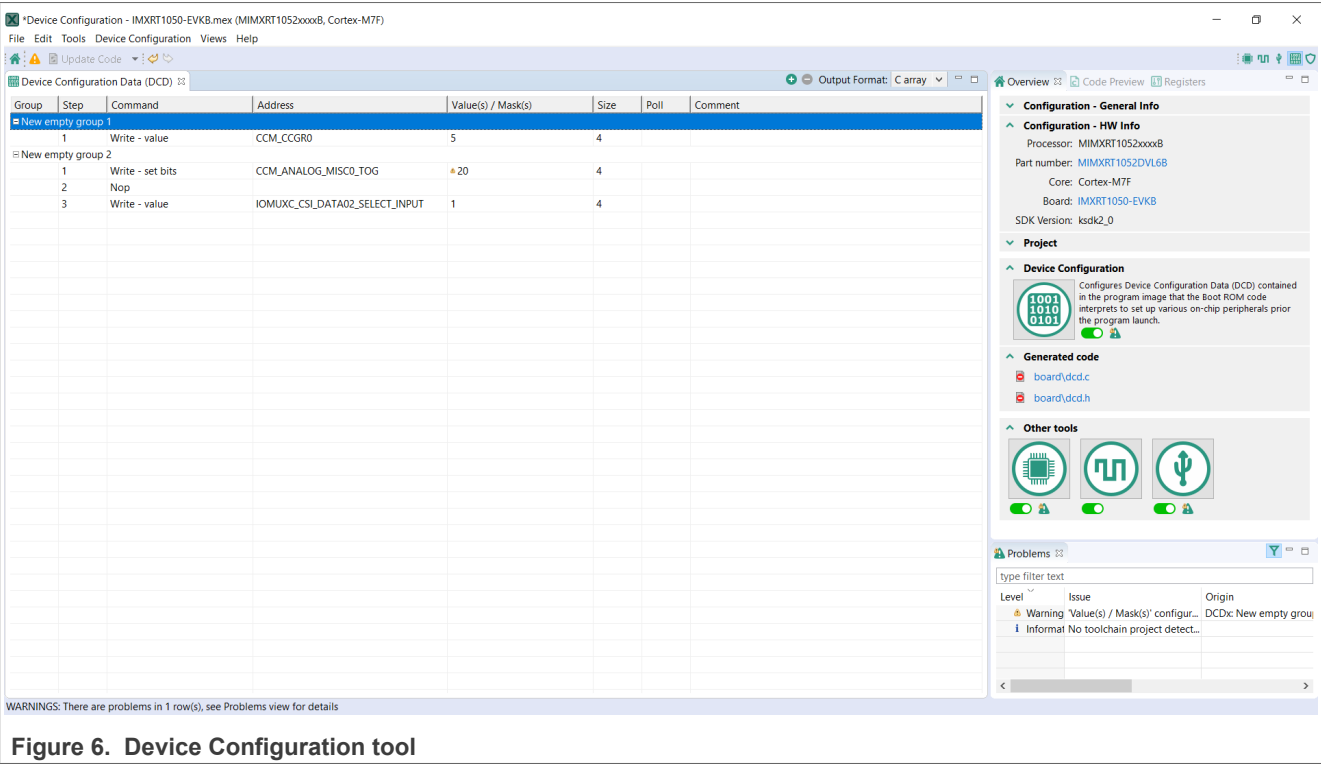


Figure 6. Device Configuration tool

3.5 Trusted Execution Environment Tool

In the **Trusted Execution Environment**, or **TEE** tool, you can configure the security policies of memory areas, bus masters, and peripherals, to isolate and safeguard sensitive areas of your application.

You can set the security policies of different parts of your application in the **Security Access Configuration** and its subviews, and review these policies in the **Memory Attribution Map**, **Access Overview** and **Domains Overview** views. Use the **User Memory Regions** view to create a convenient overview of memory regions and their security levels.

You can also view the registers handled by the **TEE** tool in the **Registers** view, and inspect the code in the **Code Preview** tool.

Note: For your configuration to come into effect, make sure you have enabled the relevant enable secure check option in the **Miscellaneous** subview of the **Security Access Configuration** view.

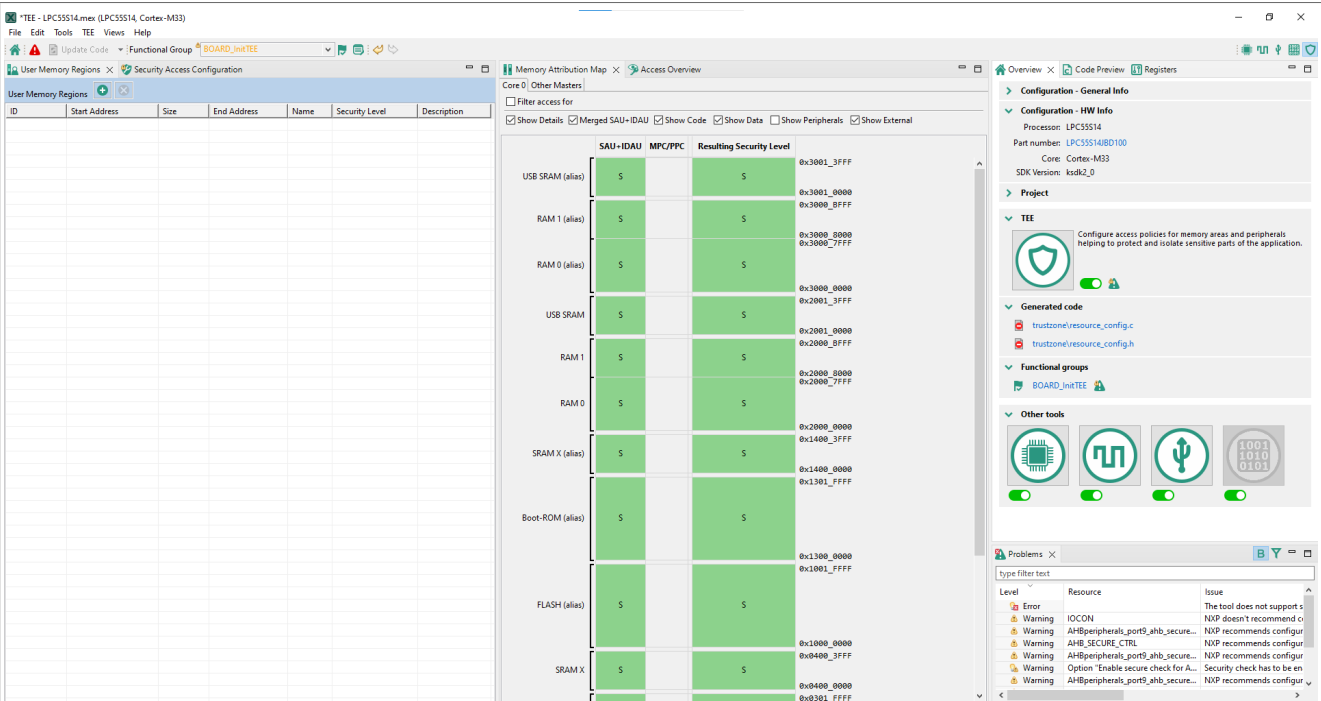


Figure 7. TEE tool user interface (SAU+IDAU)

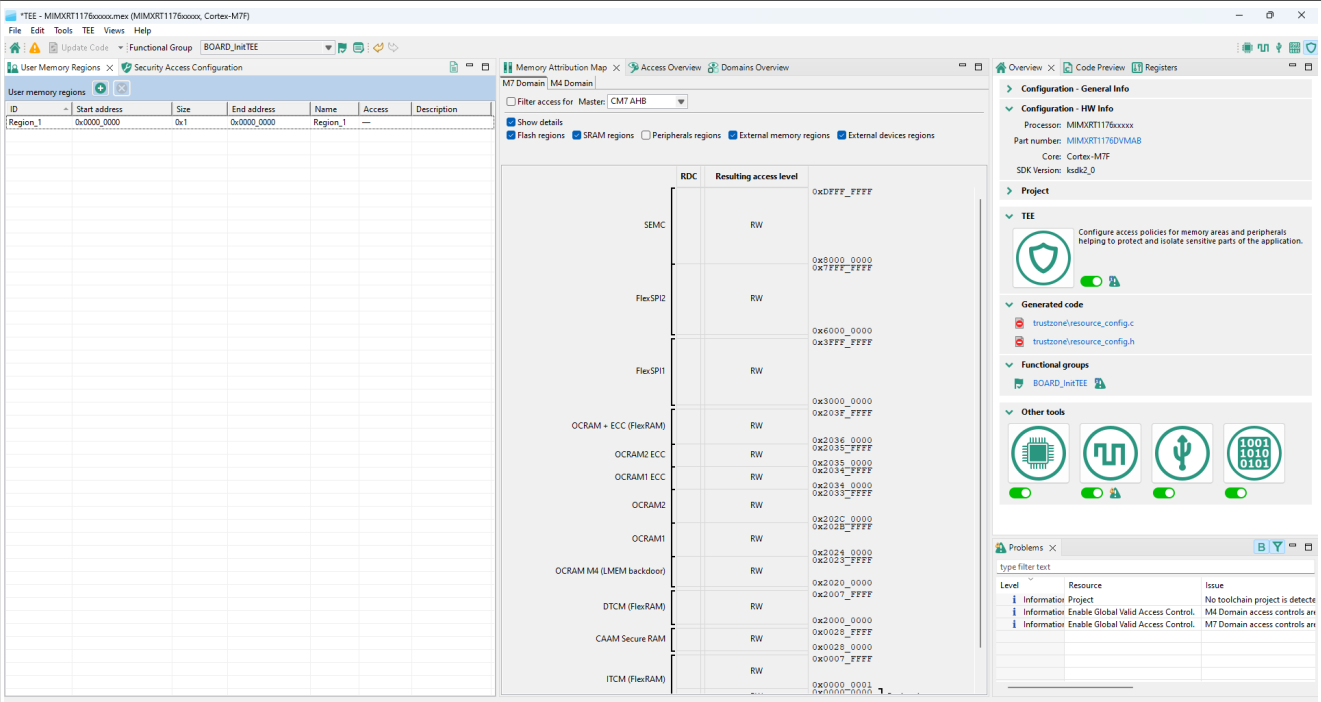


Figure 8. TEE tool user interface (RDC)

4 Generate code

You can update the source files generated by the **MCUXpresso Config Tools** by clicking the **Update Code** button. The command opens a dialog with a list of files that will be regenerated and allows you to select which tools will generate the code.

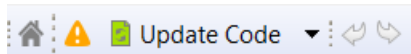


Figure 9. Update Code button

Note: In order to generate the code, the configuration has to be saved. This is described later in this document. Alternatively, you can also export a selected source file by clicking the export button in the **Code Preview** view.

```

1 /*
2  * Copyright 2025 NXP
3  * All rights reserved.
4  *
5  * SPDX-License-Identifier: BSD-3-Clause
6  */
7
8 /*****
9  * This file was generated by the MCUXpresso Config To
10 * will be overwritten if the respective MCUXpresso Co
11 *****/
12
13 /*
14 * TEXT BELOW IS USED AS SETTING FOR TOOLS *****
15 !!GlobalInfo
16 product: Pins v17.0
17 processor: MIMXRT1176xxxxx
18 package_id: MIMXRT1176DVMAB
19 mcu_data: kSDK2_0
20 processor_version: 0.2509.20
21 * BE CAREFUL MODIFYING THIS COMMENT - IT IS YAML SETT
22 */
23
24 #include "fsl_common.h"
25 #include "pin_mux.h"
26
27 /* FUNCTION *****/
28 *
29 * Function Name : BOARD_InitBootPins
30 * Description   : Calls initialization functions.
31 *
32 * END *****/
33 void BOARD_InitBootPins(void) {
34     BOARD_InitPins();
35 }
36

```

Figure 10. Code Preview

5 Note about the source code in the document

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

Copyright 2026 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 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.

6 Revision history

Table 2. Revision history

Document ID	Release date	Description
MCUXDQS v.17.0	17 September 2026	Updated for v.26.09
MCUXDQS v.16.0	24 June 2026	Updated for v.26.06
MCUXDQS v.15.0	25 March 2026	Updated for v.26.03
MCUXDQS v.14.0	16 December 2025	Updated for v.25.12
MCUXDQS v.13.0	18 September 2025	Updated for v.25.09
MCUXDQS v.12.0	20 June 2025	Updated for v.25.06
MCUXDQS v.11.0	17 March 2025	Updated for v.25.03
MCUXDQS v.10.0	15 January 2025	Updated for v.24.12
MCUXDQS v.9.0	24 September 2024	Updated for v.16.1
MCUXDQS v.8.0	1 July 2024	Updated for v.16
MCUXDQS v.7.0	19 April 2024	Updated for v.15.1
MCUXDQS v.6.0	10 January 2024	Updated for v.15
MCUXDQS v.5.0	31 July 2023	Updated for v.14
MCUXDQS v.4.0	2 January 2023	Updated for v. 13
MCUXDQS v.3.0	20 September 2022	Updated for v.12.1
MCUXDQS v.2.0	30 June 2022	Updated for v.12
MCUXDQS v.1.0	22 December 2021	Screenshots are updated.

Table 2. Revision history...continued

Document ID	Release date	Description
MCUXDQS v.0	23 June 2021	Initial version

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.

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. 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.

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 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.

CodeWarrior — is a trademark of NXP B.V.

IAR — is a trademark of IAR Systems AB.

Contents

1 Introduction2

2 Create a new configuration2

2.1 Cloning an SDK example2

2.2 Creating a new toolchain configuration4

2.2.1 Creating a new toolchain configuration in
MCUXpresso for VS Code4

2.3 Creating a new standalone configuration5

3 Select tool5

3.1 Pins Tool6

3.2 Clocks Tool6

3.3 Peripherals Tool7

3.4 Device Configuration Tool8

3.5 Trusted Execution Environment Tool9

4 Generate code10

5 Note about the source code in the
document12

6 Revision history12

Legal information14

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