IMXUG User Guide for Config Tools for i.MX Rev. 7 — 10 January 2024

User guide

Document information

Information	Content
Keywords	MCUXpresso Config Tools, i.MX
Abstract	The Config Tools for i.MX is part of MCUXpresso Config Tools, a suite of evaluation and configuration tools that help users from initial evaluation to production software development. Config Tools for i.MX is an easy-to-use way to configure the pins and DDR of the i.MX processor devices. The software, in general, enables you to create, inspect, change, and modify any aspect of the pin configuration and muxing of the device. It also allows you to configure and validate DDR settings. This document describes the basic components of the Config Tools for i.MX and lists the steps to configure and use them to configure both pins and DDR.



1 Introduction

The Config Tools for i.MX is part of MCUXpresso Config Tools, a suite of evaluation and configuration tools that help users from initial evaluation to production software development. Config Tools for i.MX is an easy-to-use way to configure the pins and DDR of the i.MX processor devices. The software, in general, enables you to create, inspect, change, and modify any aspect of the pin configuration and muxing of the device. It also allows you to configure and validate DDR settings. This document describes the basic components of the Config Tools for i.MX and lists the steps to configure and use them to configure both pins and DDR.

Note: Only the standalone desktop version is currently available for *i*.MX processors.

1.1 Features

The Config Tools for i.MX consists of the Pins, TEE, DDR, SERDES, PBL tools.

The Pins tool is designed for:

- Configuration of pin routing/muxing
- Managing different functions used for routing initialization
- · Configuration of pin functional/electrical properties
- · Generation of code for routing and functional/electrical properties

The DDR tool is designed for:

- · Configuration of DDR controllers
- Validation of DDR configuration

The Pins tool can be used to define routing of pins for target device/board. The tool configuration may be shared using the stored configuration in the MEX file or by using the generated C or DTSI (optional) snippet files (via Import/Export or via copy-paste of the generated source).

Note: The Pins Tool, in general, generates code for routing the pin to the peripheral, but not for the configuration of the peripheral. Some peripherals might need additional configuration of the pin to assign function or channel. For example, for some ADC the routing provide connection between pin and the ADC peripheral. You can then assign the ADC channel from within the ADC peripheral.

The DDR tool allows you to view and configure basic DDR attributes, such as memory type, frequency, number of channels and others and test the DDR configuration by a variety of tests. After you have specified the connection type, you can choose scenarios, tests to run in these scenarios, and view the test results, logs, and summary.

1.2 Versions

For i.MX, the tool is referred to as Config Tools for i.MX and is available as a desktop application only. The tool contacts the NXP server and fetches the list of the available processors. Once used, the processors data is retrieved on demand. To use the desktop tool in the offline mode, create a configuration for the given processor while online. The tool will then store the processors locally in the user folder and enable faster access and offline use.

User Guide for Config Tools for i.MX

•		▼ C Functional G			▼ ■ <									🗢 🛈 🛤 🌒
		ignals 👹 Power Gro		Jser Signals		- 8	Package [Pins Bottom] >	< 📫 Expansion Header		ତ୍ତ୍ର ୧୦	n 🔟 🗐 🍄 🖂) - 8	A Overview X Code Preview II Registers	
0		00 \$ Ø /	type filter text									î	> Configuration - General Info	
Routin	Pin name VisS0 GPI0_EM4C_15 GPI0_EM4C_15 GPI0_EM4C_18 GPI0_EM4C_28		Identifier SENC_RAG SENC_EAD SENC_VE SENC_CSD SENC_GSD SE	Arduine Header 12441 (011) 12451 (012)	GPIO4.gpic_ia, 15 GPIO4.gpic_ia, 21 GPIO4.gpic_ia, 28 GPIO4.gpic_ia, 28 GPIO4.gpic_ia, 22 GPIO3.gpic_ia, 10 GPIO3.gpic_ia, 14 GPIO3.gpic_ia, 15 GPIO4.gpic_ia, 17 GPIO4.gpic_ia, 16 GPIO4.gpic_ia, 18 GPIO4.gpic_ia, 29	LPUARTERTS_B LPUARTETTRG[_] LPUARTERTS_B[LPUARTERTS_B[LPUARTERTS_B] LPUARTERTS_B LPUARTERTS_B LPUARTETS	• _ •			 3 2 1			Configuration - HW fiel Processor MARKIT10520VL68 Core: Cortex-MF Board: MARKIT10520VL68 Core: Cortex-MF Board: MARKIT10520VL68 Soft Version: kedd2.0 Project Project Project Concentrated code Softward: S	onal & electrical pin
	Peripheral	Signal Ar	r Routed pin	Label Iden	ntifier Powerg	roup Direction GF	10 initial GPIO interr Sof	tware In Hysteresis	Pull Up/Do Pull/Keepe Pu	ull/Keepe Op	en drain Speed		A Problems ×	Y =
													type filter text	
													Level Resource Issu	e

1.3 Tools localization

Tools are available in English and Simplified Chinese only.

The locale of Tools automatically copies the global settings of your computer.

To set the locale manually, add the following parameter to the command line:

tools.exe -nl zh

You can also set the locale in the tools.ini file by adding the following line:

-Duser.language=zh

Note: Setting your system locale to Chinese automatically launches the tool with localized Chinese menu items, tool tips, and help. You may need to delete the [home_dir]/.nxp folder after switching languages because some menu items may be cached.

2 User Interface

2.1 Start Development wizard

Upon starting Config Tools, you are automatically welcomed by a startup wizard. With this wizard, you can create a configuration or open an existing one.

Note: To skip the wizard on subsequent startups, select the **Always open last configuration** checkbox below the **Open an existing configuration** option. You can also perform the same action by selecting the **Automatically open previously used configuration** checkbox in **Preferences**.

Start development	_		×
Select a way to start development			
Open an existing configuration			
Use this option to edit an existing configuration file (.mex).		_	_
Select an existing configuration (*.mex)	~	Brow	wse
Always open last configuration			
Create a new standalone configuration for a processor, board, or kit			
Use this option to create a new Pins, Clocks, and/or Peripherals configuration for a selected processor or board without association to project. Generated source code can be exported to a specified folder. It is possible to associate the configuration to any toolchain pro saving the standalone configuration file (.mex) that will be generated by Config Tools into the toolchain project directory and then op the "Open an existing configuration".	ject later by		
< Back Next >	Finish	Cano	el
Finue 2. Start development without			

Figure 2. Start development wizard

Note: The content of this wizard is similar to the wizard that you open by selecting File > New in the Menu bar.

2.2 Creating, saving, and opening a configuration

In this context, configuration stands for common tools settings stored in an MEX (Microcontrollers Export Configuration) file. This file contains settings of all available tools . The folder with the saved MEX file must contain exactly one project file to be able to parse the toolchain project.

2.2.1 Creating a new configuration

You can create a configuration from the **Start development** wizard or by selecting **File > New** from the **Menu bar**.

If you start creating your development for any NXP board or kit, we recommended you start with example to create a configuration for a board or a kit. Such configuration contains board-specific settings. If you select a processor, the configuration will be empty.

After the new configuration is created, you can continue by importing an existing configuration from an MEX file. It 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**.

2.2.1.1 Creating a new standalone configuration

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

Start development											-	_	×
Create a new configuration Tip: To apply an existing board configuration, impo	ort it using th	e com	imand f	File > Impo	ort								
Select a processor/board/kit													
type filter text													
Select a processor/board/kit	Pins	TEE	DDR	SERDES	PBL	Status							^
✓ Boards													
IMXRT1050-EVKB	~	×	X	×	×	Online							
MCIMX6D-EVB-REV-X3	~	×	X	×	×	Online							
MCIMX6D-SABRE-AI-CPU2-REV-A	1	×	X	×	×	Online							
MCIMX6D-SABRE-AI-CPU2-REV-B	~	×	X	×	×	Online							
MCIMX6D-SABRE-LITE-REV-A	~	×	X	×	×	Online							
MCIMX6D-SDB-REV-A	~	×	X	×	×	Online							
MCIMX6D-SDB-REV-B	~	×	X	×	×	Online							
MCIMX6DL-EVB-REV-X3	~	×	×	×	×	Online							
MCIMX6DL-SABRE-AI-CPU2-REV-A	~	×	X	×	×	Online							
MCIMX6DL-SABRE-AI-CPU2-REV-B	~	×	×	×	×	Online							
MCIMX6DL-SABRE-LITE-REV-A	~	×	X	×	×	Online							
MCIMX6DL-SDB-REV-A	~	×	X	×	×	Online							
MCIMX6DL-SDR-REV-R	1	×	×	×	×	Online							~
Name your configuration													
Select a processor package			(Select a co	7.0			Se	elect an SDK ver	rion			
									licer an objever	31011			
			~				~						~
									< Back	Next >	Finish	Cance	el
	-												
Figure 3. Creating a new	config	ura	tion	1									

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 <u>Working offline</u> section.
- 4. Name your configuration. Optionally, you can select processor package, core, and SDK version.
- 5. Click Finish.

2.2.2 Saving a configuration

To save your configuration for future use, select File>Save from the Menu bar.

To save a back-up of your configuration, do the following:

- 1. In the Menu bar, select File>Save Copy As.
- 2. In the dialog, specify the name and destination of the configuration.
- 3. Click Save.

The folder with the saved MEX file must contain exactly one project file to be able to parse the toolchain project.

2.2.3 Opening an existing configuration

To open an existing configuration, do the following:

- 1. In the Start development wizard, select Open an existing configuration. Alternatively, in the Menu bar, select File > Open.
- 2. Click **Browse** to navigate to your configuration file.
- 3. Select the configuration file and click **Open**.
- 4. Optionally, select **Always open last configuration** to skip the **Start development** wizard and load the lastsaved configuration by default.

2.2.4 User templates

You can export and store the current configuration as a user template for later use as a reference configuration file.

Export			×
Select Export Configuration as Template		2	5
Select an export wizard:			
type filter text			
 > Description PBL Configuration Tool > Description Pins Tool > Description Processor Data > Description Proces			
< Back Next > Fir	nish	Cancel	
Figure 4. Export template			

The exported template is available in the **New Configuration** wizard and can be used to create a configuration. You can also define custom labels for pins or identifiers prefixes for #define in generated code. You can export the configuration by selecting, in the **Menu bar**, **File > Export > Tools Configuration > Export Configuration as Template**.

Create a new configuration, import it using the command File > Import Select a processor/board/kit Type filter text Select a processor/board/kit Pins TEE DDR SERDES PBL Status MIMXRT1052xxxxB Cached Image: Ca	Create New Configuration								_		×
Select a processor/board/kit Select a processor/board/kit Pins Templates MIMXRT1052xxxxB Boards Processors Image: Select a core Select an SDK version Select a processor package Select a core Select an SDK version	Create a new configuration										
type filter text Pins TEE DDR SERDES PBL Status Select a processor/board/kit Pins TEE DDR SERDES PBL Status MIMIXRT1052xxxxB Image: Cached Image: Cached Image: Cached Image: Cached Image: Cached Boards Image: Image: Cached Image: Cached Image: Cached Image: Cached Image: Cached Processors Image: Image: Cached Image: Cached Image: Cached Image: Cached Image: Cached Processors Image: Image: Cached Image: Cached Image: Cached Image: Cached Image: Cached Image: Image: Cached Image: Image: Cached Image: Cached Image: Cached Image: Cached Image: Cached Image: Image: Cached Image: Image: Cached Image: Image: Cached Image: Image: Cached Image: Image: Cached Image: Image: Cached Image: Image: Cached Image: Image: Cached Image: Image: Cached Image: Image: Cached Image: Image: Cached Image: Image: Image: Cached Image:	Tip: To apply an existing board configura	tion, import it u	ising th	e com	mand f	File > Impo	ort				
Select a processor/board/kit Pins TEE DDR SERDES PBL Status	Select a processor/board/kit										
Templates MIMXRT1052xxxxB Boards Processors Processors Cached Cached Processors Cached Processors Processors Processors Processors Processors Processors Processors Processors Processors Processors Processors Processors <	type filter text										
MIMXRT1052xxxxB > Boards > Processors Image: Control of the second seco	Select a processor/board/kit		Pins	TEE	DDR	SERDES	PBL	Status			
> Boards > Processors </td <td></td>											
Processors Image: Control of the second seco								Cached			
Name your configuration Select a processor package Select a core Select a core Select an SDK version	·										
Select a processor package Select a core Select an SDK version	7 110(13013										
Select a processor package Select a core Select an SDK version											
Select a processor package Select a core Select an SDK version											
Select a processor package Select a core Select an SDK version											
Select a processor package Select a core Select an SDK version											
Select a processor package Select a core Select an SDK version											
Select a processor package Select a core Select an SDK version											
	Name your configuration										
	Select a processor package	Select a co	re				Se	lect an SDK versio	on		
< Back Next > Finish Cancel	×					~					~
< Back Next > Finish Cancel											
< Back Next > Finish Cancel											
< Back Next > Finish Cancel											
				< [Back	Ne	xt >	Finish		Cance	

Figure 5. Create a new configuration from the template

Note: The templates are stored in at the following location on your local hard disk: {\$user}/.nxp/{tools_folder}/{version}/templates.

2.2.5 Importing sources

You can import source code files to use as a basis for further configuration.

Note: You can import only C files containing valid YAML configuration blocks generated by the Config Tools. The configuration is reconstructed from the YAML block and the rest of the imported file is ignored.

To import source code files, do the following:

- 1. In the **Menu bar**, select **File > Import...**.
- 2. From the list, select MCUXpresso Config Tools>Import Source.

Import	_		×
Select Choose import wizard.		Ľ	5
Select an import wizard: type filter text Config Tools for i.MX Config Tools for i.MX Config Tools for i.MX Configuration (*.mex) Configuration (*.mex) C			
< Back Next > Finish		Cance	4
e 6. Import Source wizard			

- 3. Click Next.
- 4. On the next page, click **Browse** to specify the location of the source file.
- 5. Select the source file that you wish to import and click **Open**.
- 6. On the next page, select which functional groups to import (based on tools) by selecting the checkbox in the left column.
- 7. Define how to import the functional groups by selecting one of the two available options in the dropdown menu in the right column:
 - Rename All files are merged into the current configuration. It imports all the functions only. If the imported function has the same name as an existing one, it is automatically renamed to the indexed one.
 For example, if BOARD_InitPins exists in the configuration then the imported function is renamed to BOARD_InitPins1 .
 - Overwrite All files are merged into the current configuration. It imports all the functions only. If the
 imported function has the same name as an existing one, then the existing one is replaced with the
 imported one.
- 8. Click Finish.

2.2.5.1 Importing configuration

To import an existing configuration from an MEX file, do the following:

- 1. In the Menu bar, select File > Import...>.
- 2. In the Import wizard, select MCUXpresso Config Tools > Import configuration (*.mex).
- 3. Click Next.
- 4. On the next page, click **Browse** to specify the location of the registers file.
- 5. Select the MEX file that you wish to import and click Open.
- 6. On the next page, select which functional groups to import (based on tools) by selecting the checkbox in the left column.
- 7. Define how to import the functional groups by selecting one of the two available options in the dropdown menu in the right column:
 - Rename All files are merged into the current configuration. It imports all the functions only. If the imported function has the same name as an existing one, it is automatically renamed to the indexed one. For example, if BOARD_InitPins exists in the configuration then the imported function is renamed to BOARD InitPins1.
 - Overwrite All files are merged into the current configuration. It imports all the functions only. If the
 imported function has the same name as an existing one, then the existing one is replaced with the
 imported one.
- 8. Click Finish.

	Ø Import − □ ×	
	Select Import MEX Configuration	
	Select an import wizard: type filter text	
	 Config Tools for i.MX Import Board/Kit Configuration Import Configuration (*.mex) Import External User Signals Import Output path overrides Import Processor Data Import Source Pins Tool Import Legacy IOMux Tool Design Configuration (XML) Format Import Legacy i.MX Pins Configuration (PEx for i.MX) Format Run/Debug Breakpoints Launch Configurations 	
Figure 7. Im	< Back Next > Finish Cancel	

In the second			
Import the configuration			
Path to an existing configuration (*.mex):			
		Dr	owse
Functional encourse forward to be improved		ы	Jwse
Functional groups found to be imported:			
Configuration/Functional group	Import opti		

2.2.5.2 Importing Board/Kit Configuration

Use import settings from default board/kit templates provided within CFG tools data for further configuration.

To import board/kit configuration, do the following:

- 1. In the Menu bar, select File > Import...>.
- 2. In the Import wizard, select MCUXpresso Config Tools > Import Board/Kit Configuration.
- 3. Click Next.
- 4. On the next page, select the board/kit variant from the dropdown menu.
- 5. Select which functional groups to import (based on tools) by selecting the checkbox in the left column.
- 6. Define how to import the functional groups by selecting one of the two available options in the dropdown menu in the right column:
 - **Rename** All files are merged into the current configuration. It imports all the functions only. If the imported function has the same name as an existing one, it is automatically renamed to the indexed one. For example, if BOARD_InitPins exists in the configuration then the imported function is renamed to BOARD_InitPins1.
 - **Overwrite** All files are merged into the current configuration. It imports all the functions only. If the imported function has the same name as an existing one, then the existing one is replaced with the imported one.
- 7. Click Finish.

2.2.6 Exporting sources

It's possible to export the generated source using the Export wizard.

To launch the Export wizard:

1. Select File > Export from the Menu bar.

2. Select Export Source Files.

Export		_	
Select Export Source Files			Ż
Select an export wizard:			
type filter text			
 > > Processor Data > > Run/Debug > > TEE Tool > > Tools Configuration 	Jser Signals port les n CSV (Comma Separated Val n ation as Template	ues) Format	
<	Back Next >	Finish	Cancel

- 3. Click Next.
- 4. Select the target folder where you want to store the generated files.

User Guide for Config Tools for i.MX

	Export						×	
	Export Pins Source	e Files						
	(i) File(s) to export:	pin_mux.h, pin_m	nux.c					
	Cortex-M7F						-	
	C:\nxp\Project				~	Browse		
		< Back	Next >	Finish		Cancel		
Eiguro 10	Export sources – Sele	of forgot folder						
FIGULE IV. E	zaport sources - Sele	u larget ioider						

5. In case of multicore processors, select the cores you want to export.

6. Click Finish.

2.3 Menu bar

The Menu bar contains six menus: File, Edit, Tools, Views, Help, and a tool-specific menu.

The File menu contains file management items.

Description	
Description	
Create a configuration. For more information, see the <u>Creating, say</u> configuration section.	<u>ving, and opening a</u>
Open a configuration from an MEX file.	
Save the current configuration.	
Create a backup copy of the current configuration.	
·	© 2023 NXP B.V. All rights reserved
	Open a configuration from an MEX file. Save the current configuration.

All information provided in this document is subject to legal disclaimers.

User Guide for Config Tools for i.MX

Table 1. File menucontinue	20		
Menu item	Description		
Switch processor	Switch to a different processor. For more information, see the <u>Switching processor</u> section.		
Switch package	Switch to a different processor package. For more information, see the <u>Switching processor</u> section.		
Select Core	Select a processor core for further configuration.		
Data Manager	Manage local data. For more information, see the Managing data and working offline section.		
Import	Import settings from source files. For more information, see the <u>Advanced Features</u> section.		
Export	Export source files and other tool information. For more information, see the <u>Advanced</u> <u>Features</u> section.		
Exit	Exit the application. If there are any unsaved changes, you are prompted to save the changes.		

Table 1. File menu...continued

The Edit menu contains basic editing actions as well as items modifying the appearance and behavior of the whole framework.

Menu item	Description
Open Update Code Dialog	Update code after configuration change. For more information, see the <u>Section 2.4.2</u> section.
Undo ()	Cancel a previous action. The action to be undone is always appended.
Redo ()	Cancel a previous undo action. The action to be redone is always appended.
Сору	Copy the selected text to the clipboard.
Select All	Select the whole text in the current field/view.
Call from default initialization function	Set the currently selected functional group to be called from the default initialization function.
Functional Group Properties	Edit functional group properties.
Preferences	Edit preferences. For more information, see the Preferences section.
Configuration Preferences	Edit configuration preferences. For more information, see the <u>Configuration Preferences</u> section.

The Tools menu lists all the tools available in the tools framework. Use this menu to switch between the tools.

The Tool-specific menu contains items tailor-made for individual tools. Only items relevant to the currently active tool are displayed. The menu name copies the name of the currently active tool.

Item	Description	
Functional Groups	Edit functional group properties.	
Automatic Routing	Attempt to resolve routing issues. Opens the Automatic Routing dialog, which displays routing issues that have been resolved and the ones that require manuacorrection.	
Apply Expansion Board	Apply an expansion board to an already created expansion header	
Create the Default Routing	Open a dialog for the creation of a new functional group containing the after-reset state of pins and internal signals.	

© 2023 NXP B.V. All rights reserved.

Table 3. Pins menucontinued		
Item Description		
Refresh	Refresh both the generated code and the whole GUI.	
Reset to Board Defaults	Reset the configuration of the Board/Kit defaults.	
Reset to Processor Defaults	Reset the configuration of the processor's defaults.	

The **Views** menu contains a tool-specific list of available views. Select a view from the list to open it. Select an already opened view to highlight it. Choose **Reset views** to reset the current tool perspective to its default state. The **Help** menu contains assistance and general information-related items.

Table 4. Help menu

Item	Description
Contents	Display the User Guide.
Quick Start guide	Open a PDF file of the Quick Start guide.
Release Notes	Display release notes of the installed version.
Community	Display web pages of the product-related community forums.
Processor Information	Display web pages containing information about the currently used processor.
Kit/Board Information	Display web pages containing information about the currently used board or kit.
Check for updates	Check for a newer version of the product. If a new version is available, you are prompted to confirm and perform the update
Open Cheat Sheet	Display a cheat sheet to help with using the tools. You can also load a cheat sheet from a file, or from a URL.
About	Display general product information.

2.4 Toolbar

The toolbar is on the top of the window and includes buttons/menus of frequently used actions common to all tools. See the following sections for more information.

Item	Description
Config Tools Overview	Open the Overview dialog with information about currently used tools.
Show Problems View	Open the Problems view.
Update Code	Open the update dialog allowing you to update generated peripheral initialization code directly within specified toolchain project.
Generate Code	Regenerate source code when "Enable Code Preview" preference is disabled.
Functional group selection	Select functional group. Functional group in the Peripherals tool represents a group of peripherals that are initialized as a group. The tool generates a C function for each function group that contains the initialization code.
Call from default initialization	Set the current functional group to be initialized by the default initialization function.
Functional group properties	Open the Functional group properties dialog to modify name and other properties of the function group.

IMXUG

© 2023 NXP B.V. All rights reserved.

Table 5. Toolbarcontinued		
Item Description		
Tool selection	selection Display icons of individual tools. Use them to switch between tools.	
Undo/Redo	Undo/Redo last action.	

In addition, the toolbar may contain additional items depending on the selected tool. See the chapters dedicated to individual tools for more information.

2.4.1 Config tools overview

The **Config Tools Overview** provides you with general information about your currently active configuration, hardware, and project. It also provides a quick overview of the used/active and unused/inactive tools, generated code, and functional groups. By default, the **Config Tools Overview** icon is on the left side of the toolbar.

Config Tools Overview contains several items.

 Table 6. Config Tools Overview

Item	Description
Configuration – General Info	Displays the name of and the path to the MEX file of the current configuration. Click the link to open the folder containing the MEX file. To import additional settings, click the Import additional settings into current configuration button.
Configuration – HW Info	Displays the processor, part number, core, and SDK-version information of the current configuration.
Project	Displays toolchain project information.
Pins/DDR/SERDES/PBL/TEE	Displays basic information about Pins, DDR, SERDES, PBL, TEE tools.

To enable/disable the tools, click the toggle button. You can navigate to the tools by clicking their icons. Following information about the tools is also available:

Table 7. Config Tools Overview		
Item	Description	
Generated code	Contains the list of source-code files. Click the links to open the files in the Code Preview view.	
Functional groups	Contains the list of the currently active functional groups. To select the groups in the Functional groups tab in the toolbar, select the relevant links.	

User Guide for Config Tools for i.MX

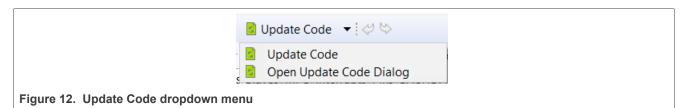
Config Tools Overview				_		×
PBL tool does not support selected processor. Select different tool.						
Configuration - General Info Configuration is not saved on the disk	✓ Configuration - HW Info Processor: MIMXRT1052xxxxB Part number: MIMXRT10520VL68 Core: Cortex-M7F Board: IMXRT1050-EVKB SDK Version: ksdk2_0		 Project No toolchain project detected 			
Pins Configures pin routing, including functional & electrical pin run-time pin configuration.	properties, voltage/power rails, and	✓ PBL The tool does not support selected p	processor			
 Generated code board\pin_mux.c board\pin_mux.h 						
 Functional groups BOARD_InitPins BOARD_InitDEBUG_UARIPins BOARD_InitSDRAMPins BOARD_InitCSIPins 						
P BOARD_InitCOPIns P BOARD_InitCANPins P BOARD_InitENETPIns P BOARD_InitUSDHCPins P BOARD_InitHyperFlashPins						
Tools not supported for the selected processor. TEE, DDR, SERDES, PBL			0 Close and Up	date Code 🛛 C	Close	

Note: Unsupported tools are not displayed in the overview.

2.4.2 Update code

To update the project without opening the **Update Files** dialog, deselect the **Always show details before Update Code** checkbox.

To access the Update Code dialog from the Update Code dropdown menu, select Open Update Code Dialog.



Note: The generated code is always overwritten.

The **Update Code** action is enabled under following conditions:

- If the MEX configuration is saved in a toolchain project, the processor selected in the tool matches with processor selected in the toolchain project
- Core is selected (for multicore processors)

2.4.3 Functional groups

Every **Pins** configuration can contain several functional groups.

These groups represent functions which will be generated into source code. Use the dropdown menu to switch between functional groups and configure them.

Functional Group	BOARD_InitPins	\mathbf{v}	7		
------------------	----------------	--------------	---	--	--

Figure 13. Functional groups

You can use two additional buttons to further configure functional groups:

Table 8. Functional Groups

Icon	Description
	Toggle "Called from default initialization function" feature (in source code)
	Opens the Functional group properties window

Note:

Red/orange background indicates errors/warnings in the configuration.

2.4.3.1 Functional group properties

In the **Functional Group Properties** window, you can configure several options for functions and code generation. Each setting is applicable for the selected function. You can specify generated function name, select core (for multicore processors only) that is affecting the generated source code, or write function description (this description is generated in the C file). You can also add, copy, and remove functional groups as needed.

Aside from name and description, you can choose to set parameters for selected functional groups.

Functional group properties are specific for individual Config Tools:

The Pins tool:

- Set custom #define prefix If this property is set, the specific custom prefix is used for macros generated into the pin mux.h. Otherwise the name of the functional group is used as the prefix.
- Prefix The custom prefix string. If it is empty, no prefix is used.
- **Clocks gate enable** If this property is enabled, the clock gate is enabled in the generated code. The clock gate is needed for access to the peripherals, so have it enabled elsewhere.
- Core (for multicore processors only) Selects the core that is used for executing this function.
- Full pins initialization If this property is set, all features of the pins are fully initialized in the generated function even if matches the after-reset state of the processor. If it is not set, the value may be "not specified" or "Reset (...)" that means no code is generated and after-reset state is expected.
- **De-initialization function** If this feature is set, an additional function that sets all pins and peripheral signals in this functional group to their after-reset state is generated. The new function has a suffix _deinit by default.
- Set custom de-initialization function name Allows specifying a user-defined name of the de-initialization function.

Clocks tool:

- Set custom #define prefix If this property is set, the custom prefix is used for macros define in clock_config.h Otherwise the name of the functional group is used as the prefix.
- Prefix The custom prefix string. If it is empty, no prefix is used.
- Other settings The processor-specific settings are specific for each processor. See the tooltips for details.

Peripherals tool:

• **Prefix** - It is used for identifiers, constants, and functions related to the functional group that is used in generated code. If it is not specified, no prefix is used.

TEE tool:

• Set custom #define prefix - If this property is checked, the custom prefix is used for macros define in generate code. Otherwise the name of the functional group is used as the prefix.

Functional group properties				×
Functional groups 🕒 🕥 🖉 💌	Name:	BOARD_InitPins		
		Called by the default initialization function		
BOARD_InitPins		Set the custom #define prefix		
BOARD_InitDEBUG_UARTPins	Prefix:	BOARD_INITPINS_		
BOARD_InitCSIPins		☐ Clock gate enable		
BOARD_InitLCDPins		Full pins initialization		
BOARD_InitCANPins		Generate de-initialization function		
BOARD_InitENETPins		Set custom de-initialization function name		
P BOARD_InitUSDHCPins	De-initialization			
P BOARD_InitHyperFlashPins	function name:	BOARD_InitPins_deinit		
	Description:	Configures pin routing and optionally pin electrical features.		<u>^</u>
				\sim
		<		>
			ОК	Cancel
Figure 14. Functional group prop	perties for the	e Pins tool		

2.4.4 Undo/Redo actions

You can reverse your actions by using Undo/Redo buttons available in the **Toolbar**. You can also perform these actions from the **Edit** menu in the **Menu bar**.

Table 9. Undo/reto actions

Icon	Description
. 4	Cancels the previous action
\$	Cancels the previous undo action

2.5 Preferences

To configure preferences in the **Preferences** dialog, select **Edit>Preferences** from the **Menu bar**.

Note: You can restore settings to default by selecting Restore Defaults in the lower right corner of the dialog.

Config Tools for i.MX	Config Tools for i.MX
> Appearance	Line ending style Windows (CR + LF) Linux/Mac (LF) Default (based on host) Generate files read-only (see supported toolchains in tooltip) Always overwrite files without asking Always show details before Update Code Undo history size: 10 Network Proxy connection Native Work offline Processor data update Auto Update Vere-Boot Loader Swap PBL image in chunks of 8 bytes Pre-Boot Loader Swap PBL image in chunks of 8 bytes Always the proventient of the first time Automatically load the last configuration on startup Enable Code Preview Enable Animations
	Restore Defaults Apply
	Apply and Close Cancel

Several settings are available.

Item	Description
Line ending style	Select between Windows (CR + LF), (LF), or Default (based on host).
Generate source folder	At build time, automatically create a folder including source files.
Create empty configuration if no yaml is available	Generates a configuration even if no yaml is present.
Always overwrite files without asking	Update existing files automatically, without prompting.
Always show details before Update Code	Review changes before the project is updated.
Undo history size	Enter the maximum number of steps that can be undone. Enter 0 to disable.
Proxy connection	 Direct – Connect directly and avoid a proxy connection. Native – Use system proxy configuration for network connection. Note: The proxy settings are copied from operating system settings. In case of error, you can specify proxy information in the tools.ini file, located in the <install_dir>, bin/ folder. Make sure that the file contains the following lines: Djava.net.useSystemProxies=true (already present by default) Dhttp.proxyHost=<somecompany.proxy.net></somecompany.proxy.net> Dhttp.proxyPort=80 Note: Authentication is not supported. </install_dir>
Work Offline	Disable both the connection to NXP cloud and the downloa of processor/board/kit data.
Processor data update	 Select from the following options: Auto Update – Update the processor data automatically. Manual – Update processor data after confirmation. Disabled – Disable processor data update.
Check for application updates	Check for application updates on a weekly basis
Show pin label & identifier table columns (Pins Tool)	Select to show the pin label and the label identifier in the relevant views.
Require an identifier for Pins (Pins Tool)	Controls generation of pins "Identifier" related warnings. With this preference enabled, warnings will be generated for bidirectional signals that have no Identifier set.
Help us improve the tool	Send device-configuration and tool-use information to NXP. Sending this information to NXP helps fix issues and improve the tools
Show Overview window on opening configuration for the first time	Open the Overview dialog on opening configuration for the first time.
Automatically load last configuration on startup	Avoid the startup window and load the last used configuration instead.
Enable Code Preview	Controls how the code is generated. When this preference is enabled, code generation is performed automatically afte every change in the configuration and the Code Preview is updated accordingly. When this preference is disabled, code generation is stopped, warning message is displayed

Table 10. Preferencescontinued	
Item	Description
	in Code Preview window, and the action can be manually triggered by using one of the available options:
	• By pressing the "generate code" link highlighted in the warning message from the Code Preview window.
	• By pressing the Update Code button in the toolbar, where code update is preceded by code generation.
Enable animations	Enables animations in the user interface, such as smoother scrolling or opening a drop-down menu.

2.6 Configuration preferences

The configuration preferences are general preferences stored within the configuration storage file (MEX).

To configure the preferences related to the configuration, select Edit > Configuration Preferences from the main menu.

Configuration Preferences	×	
Configuration-specific preferences are saved in YAML in C sources Generate YAML Warning: Cannot import a source that does no		
Custom toolchain project link	Browse	
Custom copyright Custom source file copyright header	Edit	
Output paths overrides Enabled rules: 0/0 Pins tool configuration	Edit	
Generate extended information into the heat TEE tool configuration Generate the code only for registers that are		
OK Figure 16. Configuration preferences	Cancel Restore Defaults	

The following preferences are available:

 Table 11. Configuration Preferences

Item	Description
Generate YAML	Select to generate YAML into C source files.
Custom toolchain project link	Select to set the path to the toolchain project folder, otherwise the default path in the same folder as the configuration is used. An absolute path or a relative path to a saved configuration (MEX) can be used. Only for standalone Config Tools.
Custom source file copyright header	Select to add a custom copyright header to generated source files that do not already contain copyright.
Output path overrides	Rules that are used to override the path of the output files are generated by the tools. They are applied in the Update code and Exports commands. A special dialog allows editing.
Generate extended information into header file	Select to generate extended information into the header file. For projects created in earlier MCUXpresso versions, this option is selected by default.
Generate code only for registers that are different from the after-reset state	Select to generate code for registers that are different from the after-reset state.

Warning: When the source does not contain YAML code, it cannot be imported.

2.7 Problems view

The Problems view displays issues in individual tools and in the inter-dependencies between the tools.

🖺 Problems 🗙						Y - 6
type filter text						
Level	Resource	Issue	Origin	Target	Туре	
😘 Warning	LPUART1	Peripheral LPUART1 signals are rou	Pins:BOARD_InitDEBUG_UARTPins	Peripherals: BOARD_InitPeripherals	Validation	
💁 Warning	SEMC	Peripheral SEMC signals are routed	Pins:BOARD_InitSDRAMPins	Peripherals: BOARD_InitPeripherals	Validation	
💁 Warning	CSI	Peripheral CSI signals are routed in	Pins:BOARD_InitCSIPins	Peripherals: BOARD_InitPeripherals	Validation	
💁 Warning	LPI2C1	Peripheral LPI2C1 signals are route	Pins:BOARD_InitCSIPins	Peripherals: BOARD_InitPeripherals	Validation	
💁 Warning	LCDIF	Peripheral LCDIF signals are routed	Pins:BOARD_InitLCDPins	Peripherals: BOARD_InitPeripherals	Validation	
💁 Warning	CAN2	Peripheral CAN2 signals are routed	Pins:BOARD_InitCANPins	Peripherals: BOARD_InitPeripherals	Validation	
💁 Warning	FLEXSPI	Peripheral FLEXSPI signals are rout	Pins:BOARD_InitHyperFlashPins	Peripherals: BOARD_InitPeripherals	Validation	

Figure 17. Problems view

To open the **Problems** view, click the **Show Problems view** button in the **Toolbar**, or select **Views > Problems** from the **Menu bar**.

The **Problems** table contains the following information:

	Table	12.	Problems	view
--	-------	-----	----------	------

ltem	Description
Level	Severity of the problem: Information, Warning, or Error.
Resource	Resource related to the problem, such as signal name, the clock signal.
Issue	Description of the problem.
Origin	Information on the dependency source.
Target	Tool that handles the dependency and its resolution.
Туре	Type of the problem. It is either the validation checking dependencies between tools, or a single tool issue.

© 2023 NXP B.V. All rights reserved.

Every issue comes with a context menu accessible by right-clicking the table row. Use this menu to access information about the problem or to apply a quick fix where applicable. You can also copy the rows for later use by right-clicking the row and selecting **Copy** or by using the **Ctrl+C** shortcut. You can use the **Ctrl+left-click** shortcut to add additional rows to the selection.

Note: Quick fix is only available for problems highlighted with the "light bulb" icon.

Filter buttons are available on the right side of the **Problems** view ribbon.

Table 13. Filter buttons

Button	Description
Y	Filters messages in the Problems view. If selected, only problems for the active tool are displayed. See <u>Configuration preferences</u> section for details.

2.8 Registers view

The **Registers** view lists the registers handled by the tool models. You can see the state of the processor registers that correspond to the current configuration settings and also the state that is in the registers by default after the reset. The values of the registers are displayed in the hexadecimal and binary form. If the value of the register (or bit) is not defined, an interrogation mark "?" is displayed instead of the value.

User Guide for Config Tools for i.MX

• III Show the m	odified registers o	anly	
	iounicu registers e	niy.	
type filter text			
Reg. Name	Set Value	Reset Value	Value Description
> ANACTRL_FRO192M_CTRL	0x0?80d0?a	0x0?80d0?a	192MHz Free Running OScilla
> ANACTRL_X032M_CTRL	0x?021428?	0x?021428?	High speed Crystal Oscillator
> PMC_OSTIMER	0x??????8	0x?????8	OS Timer control register [Re
> PMC_PDRUNCFG0	0x??deff?4	0x??deff?4	Controls the power to variou
> PMC_RTCOSC32K	0x??ff???8	0x??ff???8	RTC 1 KHZ and 1 Hz clocks sc
> RTC_CTRL	0x??????0?	0x?????0?	RTC control register
> SYSCON_ADCCLKDIV	0x????????	0x???????	ADC clock divider
> SYSCON_ADCCLKSEL	0x????????	0x???????	ADC clock source select
> SYSCON_AHBCLKDIV	0x?????00	0x?????00	System clock divider
> SYSCON_CLKOUTDIV	0x?????00	0x?????00	CLKOUT clock divider
> SYSCON_CLKOUTSEL	0x????????	0x???????	CLKOUT clock source select
> SYSCON_CLOCK_CTRL	0x?????01	0x?????01	Various system clock control
> SYSCON_CTIMERCLKSEL0	0x????????	0x???????	CTimer 0 clock source select
> SYSCON_CTIMERCLKSEL1	0x????????	0x???????	CTimer 1 clock source select
> SYSCON_CTIMERCLKSEL2	0x????????	0x???????	CTimer 2 clock source select
> SYSCON_CTIMERCLKSEL3	0x????????	0x???????	CTimer 3 clock source select
> SYSCON_CTIMERCLKSEL4	0x????????	0x???????	CTimer 4 clock source select
> SYSCON_FCCLKSEL0	0x????????	0x???????	Flexcomm Interface 0 clocks
> SYSCON FCCLKSEL1	0x???????	0x???????	Flexcomm Interface 1 clocks
> SYSCON_FCCLKSEL2	0x???????	0x???????	Flexcomm Interface 2 clocks
> SYSCON_FCCLKSEL3	0x????????	0x???????	Flexcomm Interface 3 clocks
> SYSCON_FCCLKSEL4	0x????????	0x???????	Flexcomm Interface 4 clocks
> SYSCON_FCCLKSEL5	0x????????	0x???????	Flexcomm Interface 5 clocks
> SYSCON_FCCLKSEL6	0x????????	0x???????	Flexcomm Interface 6 clocks
> SYSCON FCCLKSEL7	0x????????	0x???????	Flexcomm Interface 7 clocks
> SYSCON_FLEXFRG0CTRL	0x????00ff	0x????00ff	Fractional rate divider for flex
> SYSCON_FLEXFRG1CTRL	0x????00ff	0x????00ff	Fractional rate divider for flex
> SYSCON_FLEXFRG2CTRL	0x????00ff	0x????00ff	Fractional rate divider for flex
> SYSCON_FLEXFRG3CTRL	0x????00ff	0x????00ff	Fractional rate divider for flex
> SYSCON_FLEXFRG4CTRL	0x????00ff	0x????00ff	Fractional rate divider for fle
> SYSCON_FLEXFRG5CTRL	0x????00ff	0x????00ff	Fractional rate divider for fle
> SYSCON FLEXFRG6CTRL	0x????00ff	0x????00ff	Fractional rate divider for flex
<			>

Figure 18. Registers view

The **Registers view** contains several items.

Table 14. Registers

Item	Description
Peripheral filter drop-down list	List the registers only for the selected peripheral. Select all to list registers for all the peripherals.
Show modified registers only checkbox	Hide the registers that are left in their after-reset state or are not configured.
Text filter	Filter content by text.
Import/Export registers	Import/Export registers from/to CSV (Import is available only for the Clocks tool)

The following table lists the color highlighting styles used in the **Registers** view.

IMXUG User guide

Table 15.	Color	codes
-----------	-------	-------

Color	Description
Yellow background	Indicates that the bitfield has been affected by the last change made in the tool.
Gray text color	Indicates that the bitfield is not edited and the value is the after-reset value.
Black text	Indicates the bit-fields that the tool modifies.

Note: When the Peripherals tool is active and register initialization components are used, the user can perform manual changes to some of the displayed values.

2.9 Log view

The **Log** view shows user-specific information about Tools operations. The **Log** view can show up to 100 records across all tools in chronological order.

Each log entry consists of a timestamp, the name of the tool responsible for the entry, severity level, and the actual message. If no tool name is specified, the entry was triggered by shared functionality.

You can filter the content of the **Log** view using the combo boxes to display only specific tool and/or severity level information. Filters in different tools can be set independently.

Buffered log records are cleared using the clear button. It affects Log views across all tools.

	🏝 Problems 💼 Global settings 📄 Log 🛛 🖓		
	Filter: All V All V	8	
	Nov 06, 2019 10:32:11 AM INFO: Working offline: false Nov 06, 2019 12:37:59 PM INFO: Configuration saved to: C:\Users \nxf43171\Documents\FRDM-K64F.mex	~	
Figure 19. Log view			

3 Pins Tool

Pins tool is an easy-to-use tool for configuration of device pins. The **Pins** tool software helps create, inspect, change, and modify any element of pin configuration and device muxing.

User Guide for Config Tools for i.MX

	0 ⇔ 🗘 🐐 .	🛕 📓 Update Code	- Function	al Group BOARD_	nitPins		■: 🖉 🐃													
Pins	× M Peripheral S	ignals 🛛 👹 Power Gro	oups 🐴 External U	Jser Signals		- 0	Package [Pins	Bottom] 🗙 🧯	Expansion Expansion	on Header			ହ ହ	৫ 🖿 📗	🔟 😳 🖃 😐	Over	view × [Code Preview 🚺 Registers		-
₽	🖸 W 🎹 🗝	oo \$ 00	P type filter text													^ > c	onfigurati	on - General Info		
A1	Pin name VSS0	Label GND	Identifier	Arduino Header		LPUA ^		14 13	12 11	ю 9 	8 7 	6 5 	к з 	2 1			-	on - HW Info MIMXRT1052xxxxB		
81	GPIO_EMC_15	SEMC_A6	SEMC_A6		GPIO4:gpio_io,15	LPUA										Par	t number:	MIMXRT1052DVL68		
C1	GPIO_EMC_21	SEMC_BA0	SEMC_BA0		GPIO4:gpio_io,21		A							•••	A		Core:	Cortex-M7F		
D1 E1	GPIO_EMC_28	SEMC_WE	SEMC_WE		GPIO4:gpio_io,28				12 ATI	A10 A9	A8 A1	A6 A5	A4 A3	A2 A1				IMXRT1050-EVKB		
1	GPIO_EMC_29 GPIO_EMC_22	SEMC_CS0 SEMC BA1	SEMC_CS0 SEMC BA1		GPIO4:gpio_io,29 GPIO4:gpio_io,22	LPUA	•	B14 B13 B	12 811	810 89	80 87	16 85	84 83	82 81	- •					
51 51	GPIO_EMC_22 GPIO_EMC_10	SEMC_BAT	SEMC_BAT		GPIO4:gpio_io,22 GPIO4:gpio_io,10		°	C14 C13 C			ca c1	C6 C5			c	SD	K Version:	ksdk2_0		
41	GPIO_EMC_10 GPIO_EMC_12	SEMIC_AT	SEMC_A1		GPIO4:gpio_io,10 GPIO4:gpio_io,12	LDUA	۰		12 011	C10 C9	C8 C/	C6 C5	C4 C3	62 61	0	> P	roject			
1	GPIO_SD_B0_02	SD1_D0/J24[4]/S	SD1_D0	J24[4] (D11)	GPIO3:gpio_io,14			014 013 0	12 011	010 09	D8 D7	D6 D5	D4 D3	02 01						
К1	GPIO SD B0 03	SD1_D1/J24[5]/S		J24[5] (D12)	GPIO3:gpio_io,15		ء	E14 E13 E	12 811	E10 E9				E2 E1	_ *	🗸 P	ins			
L1	DCDC IN0	MCU_DCDC_IN					· _								_ /		\sim	Configures pin routing, inclu	uding function	nal &
M1	DCDC_LP0	VDD_SOC_IN					•				F0 F1	F6 F5	F4 F3	F2 F1			шш	electrical pin properties, volta run-time pin configuration.	age/power rail	is, and
11	DCDC_GND0	GND					• _	G14 G13 G	12 011 1	G10 G9	08 07	G8 G5	64 63	62 61	_ °			runname pin configuration.		
1	VSS1	GND					м —	H14 H13 H		H10 H9					— н		- mm-			
2	GPIO_EMC_27	SEMC_CKE	SEMC_CKE		GPIO4:gpio_io,27	LPUA	·	H14 H13 H	112 811	H10 H9	на н/	нь нь	H4 H3	H2 H1	·			C 1		
2	GPIO_EMC_18	SEMC_A9	SEMC_A9		GPIO4:gpio_io,18	LPUA		111 112	12 J11	J10 J9	18 11	J6 J5	J4 J3	J2 J1		~ 6	ienerated	rode		
2	GPIO_EMC_09	SEMC_A0	SEMC_A0		GPIO4:gpio_io,09		к	K14 K13 H	12 KU	KID KR	K2 K2	10 15	K4 K3	F2 K1	— к					
2	GPIO_EMC_25	SEMC_RAS	SEMC_RAS		GPIO4:gpio_io,25	LPUA	۰ <u> </u>								_ ·		board\pi	n_mux.c		
2	VSS2	GND								L10 L9	0 0	16 15	4 0	12 11			board\pi	n_mux.h		
2	GPIO_EMC_04	SEMC_D4	SEMC_D4		GPIO4:gpio_io,04		• -	N14 N13 N	N2 W11	NID N9	MB N7	NG NG	M4 M3	N2 N1	- "					
2	GPIO_EMC_23	SEMC_A10	SEMC_A10		GPIO4:gpio_io,23		N	N14 N13 N			NB N7			N2 N1	N	✓ F	unctional <u>c</u>	groups		
2	GPIO SD B0 04	SD1 D2	SD1 D2		GPIO3:apio io.16	LPUA *	, _					ND ND	N4 M3	N2 N1			BOARD_I	nitPins		
							• =	812 813 8	12 811	#10 #D	28 27	16 05	07 03	12 B1		·		nitDEBUG UARTPins		
uti	ing Details														= -					
	Signals & type filt	er text														P	BOARD_I	nitSDRAMPins 🖀		
rtin	g Details for BOARD	InitPins 0	0 🛛 🗖 🗸													Prob	lems X			T -
	Peripheral	Signal Ai	rr Routed pin	Label Ider	tifier Power gi	oup Dire	ction GPIO initi	al GPIO inter	r Softv	ware In	Hysteresis	Pull U	p/Do P	ull/Keepe	Pull/Keepe Op	type fill	her text			
																	~			
																Level		Resource	Issue	
																		LPUART1		neral LPUA
																		SEMC		neral SEM
																		CSI LPI2C1		neral CSI is neral LPI20
																	Warning Warning			neral LPI20
															>			CCD1	reipn	ierai LCDI

Figure 20. Pins tool

3.1 Pins routing principle

The Pins tool is designed to configure routing peripheral signals either to pins or to internal signals.

This routing configuration can be done in the following views:

- Pins
- Peripheral Signals
- Package
- Routing Details

Following two sections describe the two methods that you can use to define the routing path.

3.1.1 Beginning with pin/internal signal selection

You can select a pin or an internal signal in the **Routing Details** view.

- 1. Select the pin/internal signal (Routed pin/signal).
- 2. Select one of the available **Peripherals**.
- 3. For the selected peripheral, select one of the available **Signals**. Items in **Peripheral** column in **Routing Details** view have the following symbols:
 - Exclamation mark and default text color indicate that such item selection can cause a register conflict or the item does not support selected signal.
 - Exclamation mark and gray text color indicate that the item cannot be routed to the selected pin/internal signal. The item is available for different pin/internal signal using the same signal. **Note:** In the **Pins** view and the **Package** view, you can configure only pins and not internal signals.

3.1.2 Routing of peripheral signals

Peripheral signals representing on-chip peripheral input or output can be connected to other on-chip peripherals or to a pin through an inter-peripheral crossbar. You can configure this connection in the **Routing Details** view.

Three types of peripheral signal routing are available:

1. Routing the signal from the output of an internal peripheral (A) into the input of another internal peripheral (B)

The signal leads from the output of one internal peripheral (A) to the input node of another internal peripheral (B). In other words, signal leads from A to B (A > B). To configure a signal in this way, perform the following steps (PWM triggering ADC (PWM > ADC) used as example):

- a. Add a row in the Routing Details view.
- b. Select peripheral B from the drop-down list in the **Peripheral** column.

Routi	ng Detail	s fo	r BOARD	1	• • •				
#	Periphe	ral	Signal	Arrow	Routed pin/signal	Label	Identifier	Direction	GPIO in
8		^	×		×	n/a	n/a	n/a	n/a
	ADC	^							
	CAN0	Cycli	ic Analog-to	-Digital	Converter (ADC)				
	СМРА								
	CMPB								

Figure 21. Selecting the peripheral (B)

c. Select the input node of peripheral B from the drop-down list in the **Signal** column.

Routi	ng Details fo		1	$\bullet \otimes \land \vee$				
#	Peripheral	Signal	Arrow	Routed pin/signal	Label	Identifier	Direction	GPIO in
8	ADC ⁶	^		×	n/a	n/a	n/a	n/a
		ANB, 4	^					
		ANB, 5						
		ANB, 6						
		ANB, 7						
		SYNC, A						
		SYNC, E <mark>Syn</mark> VDDA	<mark>nc input</mark>	Α				

d. Select the output signal of peripheral A from the drop-down list in the Routed pin/signal column.

User Guide for Config Tools for i.MX

#	Peripheral	Signal	Arrow	Routed pin/signa	al Label	Identifier	Direction	GPIO in
0	ADC	SYNC, A	20		^ n/a	n/a	n/a	n/a
				EVTG3_OUTA				^
				EVTG3_OUTB				
				PIT0_SYNC_OUT				
				PIT1_SYNC_OUT				
				PWMA0_MUX_TF	RIGO			
				PWMA0_MUX_TF	RIG1			
				PWMA0_OUT_TR	IG0	PWMA0 m	iux trigger 0	(XBARA input
				PWMA0 OUT TR	IG1			G

Figure 23. Selecting the output signal

Once the configuration is done, the row looks like this:

Rout	ng Details fo	or BOARD										
#	Peripheral	Signal	Arrow	Routed pin/signal	Label	Identifier	Direction	GPIO in				
n.	ADC	SYNC, A	<-	PWMA0_MUX_T	n/a	n/a	Input	n/a				
ure 24. I	Result											

Note: It is necessary to select the ADC peripheral where the signal leads to (input in ADC). It is a limitation of the Pins tool that the signal is not listed for the PWM peripheral (output). Notice the direction of the signal in the **Arrow** column.

2. Routing the signal from a pin on the package to internal peripheral input signal through an inter-peripheral crossbar

Note: Only if a crossbar switch is present.

The signal leads from a pin on the package (XB_IN) connected through an inter-peripheral crossbar, to an internal peripheral (B) input node. In other words, the signal leads from XB_IN to B (XB_IN > B). To configure a signal in this way, perform the following steps (routing pin 55 using XB_IN6 to EVTG0 input A (XB_IN6 > EVTG0) used as example):

a. Add a row in the Routing Details view.

b. Select peripheral B from the drop-down list in the **Peripheral** column.

User Guide for Config Tools for i.MX

Routing De	etails for B	OA	RD	1	Đ	\otimes \sim \vee
#	Peripher	ral	Signal		Ar	Routed
8	8	^	8			8
	CAN0	^				
	CMPA					
	CMPB					
	CMPC					
	CMPD					
	DACA					
	DACB					
	DMA0					
	EVTG					
	EWM Eve GPIOA	ent	Generat	or (E	VTG)	
	GPIOB	¥				
ure 25. Selecting the peripheral (E	3)>					

c. Select the input node of peripheral B from the drop-down list in the **Signal** column.

1	Routing Det	ails for BOA	RD	1	€	8 ^ ~
	#	Peripheral	Signal		Ar	Routed
	0	EVTG		^		8
			EVTG0,	A	^	
			EVTG0,		EVTG	Input A
			EVTG0,			
			EVTG0,			
			EVTG0_			
			EVTG0_		В	
			EVTG1,			
			EVTG1,	В		
			EVTG1,			
			EVTG1,	D		
			EVTG1_			
			EVTG1_	OUT,	B 🗸	
gure 26. Selecting the inp	out node (B)					

d. Select the XB_IN pin from the drop-down list in the **Routed pin/signal** column.

User Guide for Config Tools for i.MX

#	Peripheral	Signal	Ar	Routed	Label	Identifier	Direction	GPIO init	GPIO int	Slew rate	Open dr	Drive str
8	EVTG	8		^	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
				[72] GPIC	E8/PWMB_2	2B/PWMA_F	AULT0/XB_O	UT8	^			
				[73] GPIC	E9/PWMB_2	2A/PWMA_F	AULT1/XB_C	UT9				
				[55] GPIC	F0/XB_IN6/	TB2/SCLK1						
				[77] GPIC	F1 GPIOF pi	n 0:XBAR ini	out 6:Ouad t	imer B input	2 or chann	el 2 output;Q	SPI1 serial cl	ock
					F1 output							
				[45] GPIC	F11/TXD0/X	(B_IN11						
				[46] GPIC	F15/RXD0/X	KB_IN10						
				[60] GPIC	F2/SCL1/XB	OUT6/MIS	D1					
				[61] GPIC	F3/SDA1/XI	B_OUT7/MO	SI1					
				1			MA_0X/PWN	A_FAULT6				
				[63] GPIC	F5/RXD1/XI	B OUT9/PW	MA_1X/PWN	/A FAULT7	~			

Figure 27. Selecting the pin

Once the configuration is done, the row looks like this:

	Routing Details for BOARD			€	8 ^ ~	
	#	Peripheral	Signal	Ar	Routed pin/signal	
	55	EVTG	EVTG0, A	<-	[55] XB_IN6	
Figure 28. Result						

Note: In this example, GPIOF0 is multiplexed with XB_IN6, QTimerB channel 2 output/input and QSPI1 SCLK signal. In this case, the tool will automatically pick XB_IN6 for the pin as XB_IN6 is the only option to be routed to EVTG0 input A.

3. Routing the signal from internal peripheral (A) output to a pin via inter-peripheral crossbar *Note: Only if a crossbar switch is present.*

The signal leads from internal peripheral (A) output to a pin connected through an inter-peripheral crossbar on the package (XB_OUT). In other words, the signal leads from A to XB_OUT (A > XB_OUT). To configure a signal in this way, perform the following steps (routing EVTG0 output to a pin 87 using XB_OUT4 used as an example):

- a. Add a row in the Routing Details view.
- b. Select peripheral A from the drop-down list in the Peripheral column.

User Guide for Config Tools for i.MX

Routing De	tails for B	OA	RD	1	Ð	8 ^ ~
#	Peripher	al	Signal		Ar	Routed
(8)		^	8			8
	CAN0	^				
	CMPA					
	CMPB					
	CMPC					
	CMPD					
	DACA					
	DACB					
	DMA0					
	EVTG					
	EWM Eve GPIOA	ent	Generato	r (E	VTG)	
	GPIOB	\checkmark]			
jure 29. Selecting the peripheral (A)					

c. Select the input node of peripheral A from the drop-down list in the Signal column.

# Peripheral Signal Ar Routed Label Identifier Direction Image: EVTG EVTG Image: Properties of the stress of the stre	Routing De	tails for BOA	RD	1	Ð	\otimes \sim			
EVTG0, A EVTG0, B EVTG0, C EVTG0, D EVTG0_OUT, A EVTG0_C EVTG Output A (XBARA output pins routing) EVTG1, A EVTG1, B EVTG1, C EVTG1, D	#	Peripheral	Signal		Ar	Routed	Label	Identifier	Direc
EVTG0, B EVTG0, C EVTG0, D EVTG0_OUT, A EVTG0_C EVTG Output A (XBARA output pins routing) EVTG1, A EVTG1, B EVTG1, C EVTG1, D	8	EVTG ⁶		^		×	n/a	n/a	n/a
EVTG0, C EVTG0, D EVTG0_OUT, A EVTG0_C EVTG Output A (XBARA output pins routing) EVTG1, A EVTG1, B EVTG1, C EVTG1, D			EVTG0,	A	^				
EVTG0, D EVTG0_OUT, A EVTG0_C EVTG Output A (XBARA output pins routing) EVTG1, A EVTG1, B EVTG1, C EVTG1, D			EVTG0,	В					
EVTG0_OUT, A EVTG0_C EVTG Output A (XBARA output pins routing) EVTG1, A EVTG1, B EVTG1, C EVTG1, D			EVTG0,	С					
EVTG0_C EVTG Output A (XBARA output pins routing) EVTG1, A EVTG1, B EVTG1, C EVTG1, D			EVTG0,	D					
EVTG1, A EVTG1, B EVTG1, C EVTG1, D					_				
EVTG1, B EVTG1, C EVTG1, D			EVTG0_	d EV	TG O	utput A (XBA	RA output p	oins routing)	
EVTG1, C EVTG1, D									
EVTG1, D			EVTG1,	В					
EVIGI OUT A			EVTG1,	D					
			EVTG1_	OUT	, A				
EVTG1_OUT, B 🗸			EVTG1_	OUT	, B ∨				

Figure 30. Selecting the output signal (A)

d. Select the XB_OUT pin from the drop-down list in the **Route to** column.

User Guide for Config Tools for i.MX

#	Peripheral	Signal	Ar	Routed		Label	Identifier	Direction	GPIO init	GPIO int	Sle
8	EVTG	EVTG0	8		^	n/a	n/a	n/a	n/a	n/a	n/a
				[37] GPI	OA	11/CMPC_O/	XB_IN9/XB_	OUT10/USB	SOFOUT	~	
				[54] GPI	0C [.]	10/MOSI0/X	B_IN5/MISO	0/XB_OUT9			
				[87] GPI	0C'	14/SDA0/XB	OUT4/PWN	1A_FAULT4			
				[88] GP	100	pin 14;I2C0	Serial Data	Line;XBAR c	output 4;PWN	/ fault input	4
						7/SSO_B/TXE					
				[53] GPI	009	9/SCLK0/XB	IN4/TXD0/>	(B_OUT8			
				[10] GPI	OD	5/RXD2/XB_	IN5/XB OUT	F9			

Figure 31. Selecting the pin

Once the configuration is done, the row looks like this:

	Routing De	tails for BOA	RD 1	Ð	\otimes \sim
	#	Peripheral	Signal	Ar	Routed pin/signal
	87	EVTG	EVTG0	->	[87] XB_OUT4
Figure 32. Result					

Note: In this example, GPIOC14 is multiplexed with XB_OUT4, SDA of I2C0 and fault4 of eFlexPWMA. In this case, the tool will automatically configure XB_OUT4 for the pin GPIOC14 (pin 87) as XB_OUT4 is the only option for EVTG0 output A.

3.2 Example usage

This section lists the steps to create an example pin configuration, which can then be used in a user project.

In this example, three pins (**UART4_TX**, **UART4_RX** and **GPIO_2**) routed on an MCIMX6Q-SDB-REV-B board are reconfigured to match changed (for example, customer modified) board design which is using UART4 pins instead of UART3 ones and must re-route and/or adjust electrical properties for a red LED pin, so then the tool generated files with application modifying the default board configuration.

1. Create new configuration for MCIMX6Q-SDB-REV-B board.

Cre	ate a new configuration	
Tij	p: To apply existing board configuration, import it using the command File > Import	
Sel	lect Processor/Board	
Μ	1CIMX6Q	l
4	Boards MCIMX6Q-SDB-REV-B	
	Processors	
	▲ i.MX 6Quad	
	MCIMX6QxxVT	
	MCIMX6QxxZK	

down to show **Routed Pins** for it.

Functional Group	init_uart_pins	Ŧ	Р	
i ancaonar oroup	intellantellante		—	i

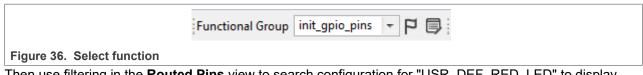
Figure 34. Select function

3. To change original RX/TX pins routing from UART3 to UART4 you must change configuration in **Peripheral** and **Route to** columns of the **Routed Pins** view for init_uart_pins selection and change pins configuration to re-route the RX/TX pins to different ones for UART4 as required in modified board design.

pe filt	ter text							
Route	d Pins for init_u	uart_pins	4 🔁 😫) 🔼 🖂				
#	Peripheral	Signal	Route to	Label	Identifier	Power group	Direction	Software Input
M3	UART1	rxd_mux	CSI0_DAT11	UART1_RX/TP27	UART1_RXD	NVCC_CSI (1.8V)	Input	0b0: Disabled
M1	UART1	txd_mux	CSI0_DAT10	UART1_TX/TP28	UART1_TXD	NVCC_CSI (1.8V)	Output	0b0: Disabled
L1	UART4	rxd_mux	CSI0_DAT13	CSI0_DAT13	n/a	NVCC_CSI (1.8V)	Input	0b0: Disabled
M2	UART4	txd_mux	CSI0_DAT12	CSI0_DAT12	n/a	NVCC_CSI (1.8V)	Output	0b0: Disabled

Figure 35. Routed Pins view

- 4. You can also adjust electrical properties configuration for these pins on the right side of the table in specific property column selection.
- 5. To change configuration of red LED pin routed originally to GPIO_2 pad, you must to select functional group 'init_gpio_pins'.



6. Then use filtering in the **Routed Pins** view to search configuration for "USR_DEF_RED_LED" to display simplified table content to easily modify current GPIO_2 pin routing selection.

ŧ	Route	d Pins						- 0		
l	JSR_DEF	F_RED_LED								
	Routed Pins for init_gpio_pins 65									
	#	Peripheral	Signal	Route to	Label	Identifier	Power group	Dire		
	T1	GPIO1	gpio, 2	GPIO_2	KEY_ROW6/USR_DEF_RED_LED/J508[35]	USR_DEF_RED_LED	NVCC_GPIO (3.3V)	Not		
F	igure	37. Route	ed Pins vi	ew						

- 7. You can then adjust either electrical properties or re-route the pin to a different one if required in your modified design.
- 8. The Pins Tool automatically generates the source code of imx6q-board.dtsi, pin_mux.c and pin_mux.h in the **Code Preview** view on the right.

User Guide for Config Tools for i.MX

```
Code Preview 🔀
                                                                                           ്പ
 imx6q-board.dtsi@Cortex-A9 (Core #0) pin_mux.c@Cortex-A9 (Core #0) pin_mux.h@Cortex-A9 (Core #0)
                                                                                             ٠
  !!GlobalInfo
 product: Pins v4.1
                                                                                             Ξ
 processor: MCIMX6QxxVT
 package_id: MCIMX6Q7CVT08AD
 mcu data: i mx 1 0
 processor version: 3.0.0
 board: MCIMX6Q-SDB-REV-B
 power_domains: {PCIE_VPH: '2.5', SATA_VPH: '2.5', NVCC_SD3: '3.3', NVCC_NANDF: '3.3'
NVCC_EIM0: '3.3', NVCC_MIPI: '2.5', NVCC_EIM1: '3.3', HDMI_VPH: '2.5', NVCC_EIM2:
NVCC_ENET: '3.3', NVCC_DRAM: '1.5'}
  * BE CAREFUL MODIFYING THIS COMMENT - IT IS YAML SETTINGS FOR TOOLS **********
  */
 /dts-v1/;
 #include "skeleton.dtsi"
 #include "imx6q-pinfunc.h"
 / {
      model = "Freescale i.MX 6Quad User Board";
     compatible = "fsl,imx6q-board", "fsl,imx6q";
     soc {
          #address-cells = <1>;
          #size-cells = <1>;
          iomuxc: iomuxc@020e0000 {
              compatible = "fsl,imx6q-iomuxc";
              reg = <0x020e0000 0x4000>;
          };
     };
 };
 &iomuxc {
     pinctrl-names = "default";
Figure 38. Generated sources
```

Figure 38. Generated sources

9. You can now copy-paste the content of the source(s) to your application or IDE. Alternatively, you can export the generated files. To export the files, click Export button on the right up corner of Code Preview view or select the menu File > Export, in the Export dialog expand the tree control for the Pins Tool and select the Export Source Files option.

Note: Tool generated board-oriented device tree (DTS) DTSI file is only a snippet and not a full device tree file content. There are just basic device tree elements, initial skeleton, and processor-specific "pinfunc.h" includes together with functional groups of fsl, pins = <...> content definitions which provide the initial IOMUXC module configuration according to the tool UI defined pin routing and functional configurations. Content itself must be manually merged together with existing Linux BSP device tree file(s) in order to apply the tool generated pins configuration. This tool also does not generate nor export processor-specific "pinfunc.h" file that is containing definition of all supported DTS pin functional configuration macros. This file

is not purposely integrated within the tool output because it is a part of separate Linux BSP support package deliverables.

3.3 User interface

The Pins tool consists of several views.

Pins	🗙 🛃 Peripheral Si	gnals 🛛 👹 Power Gro	oups 🐴 External U	Iser Signals			💿 Package [Pins Bottom] 🗙 🧔 Ex	pansion Header	ତ୍ର ର୍ 🕈 🖿 🔳	🔲 😳 🖃 🗖	A Overview 🗙 💽 Code Preview 🛐 Registe	rs 🗖
	⊡ w w -●	oo \$ 00	P type filter text							^	> Configuration - General Info	
n A1 B1 C1 D1 E1 F1 G1 H1 J1 K1 L1 M1 N1 P1 A2 B2 C2 D2 E2 F2 G2 H2	Pin name VSS0 GPI0_EMC_15 GPI0_EMC_21 GPI0_EMC_26 GPI0_EMC_28 GPI0_EMC_28 GPI0_EMC_22 GPI0_EMC_12 GPI0_EMC_12 GPI0_EMC_12 GPI0_EMC_12 GPI0_EMC_12 GPI0_EMC_12 GPI0_EMC_12 GPI0_EMC_27 GPI0_EMC_27 GPI0_EMC_23 GPI0_EMC_23 GPI0_EMC_23 GPI0_EMC_23 GPI0_EMC_23 GPI0_EMC_24 GP	Label GND SSMC_A6 SSMC_BA0 SSMC_BA0 SSMC_VE SSMC_C90 SSMC_C90 SSMC_C90 SSMC_A1 SSMC_A1 SSMC_A1 SSMC_A1 SSMC_A1 SSMC_A1 SSMC_A2 SSMC_A	Identifier SEMC_A6 SEMC_BA0 SEMC_CS0 SEMC_CS0 SEMC_BA1 SEMC_A1 SEMC_A3 SD1_D0	Arduino Header 324(4) (011) 324(5) (012)	GPIO GPIO43gpio.jo.15 GPIO43gpio.jo.21 GPIO43gpio.jo.22 GPIO43gpio.jo.22 GPIO43gpio.jo.22 GPIO43gpio.jo.22 GPIO43gpio.jo.12 GPIO43gpio.jo.12 GPIO43gpio.jo.15 GPIO43gpio.jo.16 GPIO43gpio.jo.25 G	LPUA LPUA LPUA LPUA LPUA LPUA LPUA LPUA		Image Image <th< th=""><th>1 3 2 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2</th><th> A C C<</th><th>Configuration - HW Info Processor: MIM.KRT1052xxx8 Part number: MIM.KRT1050-EVK8 SDK Version: ksdk2,0 SDK Version: ksdk2,0 Fras Fras Fras Configuration in properties, x un-time pin configuration configu</th><th>Itage/power rails, and</th></th<>	1 3 2 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	 A C C<	Configuration - HW Info Processor: MIM.KRT1052xxx8 Part number: MIM.KRT1050-EVK8 SDK Version: ksdk2,0 SDK Version: ksdk2,0 Fras Fras Fras Configuration in properties, x un-time pin configuration configu	Itage/power rails, and
utir	Signals P type filte ng Details for BOARD_ Peripheral	InitPins 0	• × • •	l shal dar	ntifier Power ar	oup Direc	ion GPIO initial GPIO interr	Soffware In Hustereric	Pull Up/Do Pull/Keepe	Bull/Keepe On	P BOARD_InitSDRAMPins A	7
σ	renpnéfál	ugnar Al	n Nouteu piñ		rowergr	up Difec	SPIO Inteller OPIO Inteller	Source II Pysteres	run op/ Uu run Neepe	Pull/Keepe Op	Iype filter text Level Resource Warning EPUART1 Warning SEMC Warning CS Warning LCD/F Warning LCD/F	lssue Peripheral LPL Peripheral SEN Peripheral CSI Peripheral LPL Peripheral LCC

3.3.1 Pins view

The **Pins** view shows all the pins in a table format.

	ins × E	\Lambda Peripheral Sig	gnals 🛛 👹 Power Gro	ups →le Exte	rnal User Sign	als		- 0
B	₿0	₩ Ш 🗕	o o 🕇 💢 🖇	type filter te	xt			
Pin	Pin	name	Label	Identifier	Arduino	GPIO	LPUART	PWM ^
\checkmark	A1 VSS0	0	GND					
		O_EMC_15	SEMC_A6	SEMC_A6		GPIO4:gpi	LPUART1	PWM1:EX
	C1 GPIC	O EMC 21	SEMC_BA0	SEMC_BA0		GPIO4:gpi		PWM3:A,
	01 GPIC	O EMC 28	SEMC_WE	SEMC_WE		GPIO4:gpi	LPUART5	PWM1:B,
	1 GPIC	O_EMC_29	SEMC_CS0	SEMC_CS0		GPIO4:gpi	LPUART6	PWM3:A,
	1 GPIC	O_EMC_22	SEMC_BA1	SEMC_BA1		GPIO4:gpi		PWM3:B,
	G1 GPIC	O_EMC_10	SEMC_A1	SEMC_A1		GPIO4:gpi		PWM2:A,
\checkmark	H1 GPIC	O_EMC_12	SEMC_A3	SEMC_A3		GPIO4:gpi	LPUART1	PWM1:EX
×	1 GPIC	O_SD_B0_02	SD1_D0/J24[4]/S	SD1_D0	J24[4] (D	GPIO3:gpi	LPUART8	PWM1:A,
	C1 GPIC	O_SD_B0_03	SD1_D1/J24[5]/S	SD1_D1	J24[5] (D	GPIO3:gpi	LPUART8	PWM1:B,
\checkmark	1 DCD	DC_IN0	MCU_DCDC_IN					
\checkmark	M1 DCD	DC_LP0	VDD_SOC_IN					
\checkmark	N1 DCD	DC_GND0	GND					
\checkmark	VSS	1	GND					
	A2 GPIC	O_EMC_27	SEMC_CKE	SEMC_CKE		GPIO4:gpi	LPUART5	PWM1:A,
	B2 GPIC	O_EMC_18	SEMC_A9	SEMC_A9		GPIO4:gpi	LPUART4	PWM4:B,
	C2 GPIC	O_EMC_09	SEMC_A0	SEMC_A0		GPIO4:gpi		PWM2:B,
	02 GPIC	O_EMC_25	SEMC_RAS	SEMC_RAS		GPIO4:gpi	LPUART6	PWM1:A,
\checkmark	2 VSS2	2	GND					
	2 GPIC	O_EMC_04	SEMC_D4	SEMC_D4		GPIO4:gpi	LPUART1	PWM4:A,
	G2 GPIC	O_EMC_23	SEMC_A10	SEMC_A10		GPIO4:gpi	LPUART5	PWM1:A,
	H2 GPIC	O SD B0 04	SD1 D2	SD1 D2		GPIO3:api	LPUART8	PWM1:A. Y
			<					>
gure 40. Pins table view								

This view shows the list of all the pins available on a given device. The **Pin name** column shows the default name of the pin, or if the pin is routed. The next columns are optional. They are **Label**, **Identifier, External**

User Signals and **Expansion header connections** (One column for each expansion header). The pin name is changed to show appropriate function for selected peripheral if routed. The next column of the table shows peripherals and signals and pin name(s) on given peripheral. Peripherals with few items are cumulated in the last column.

To route/unroute a pin to the given peripheral, select the relevant cell in the **Pin** column. Routed pins are highlighted in green. If a conflict in routing exists, the pins are highlighted in red.

Every routed pin appears in the **Routed pins** table.

When multiple functions are specified in the configuration, the **Pins** view shows pins for selected function primarily. Pins for different functions are shown with light transparency and cannot be configured until switched to this function.

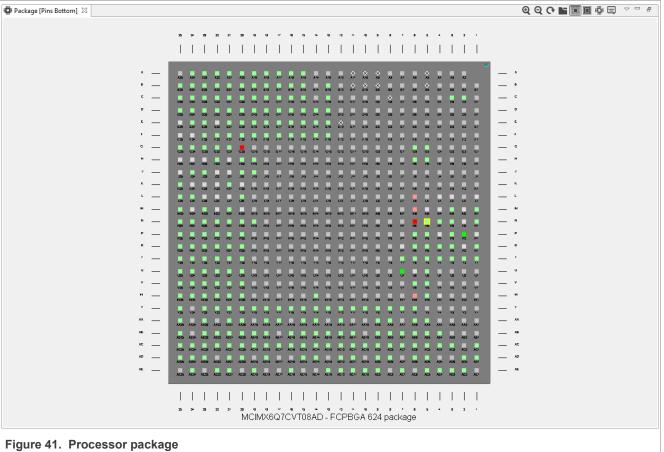
Select a row to open a drop-down list that offers the following options:

- Route/Unroute the pin.
- Highlight the pin in the **Package** view.
- Set the label and identifier for the pin.
- Add a comment to the pin. You can later inspect the comment in the Code Preview view.

Tip: The option to route more signals to a single pin is indicated by an ellipsis (...). Select the cell to open a dialog to choose from multiple available signals. The dialog also displays which signals are routed by default.

3.3.2 Package

The **Package** view displays the processor package. The processor package provides an overview of the package including resource allocation.



© 2023 NXP B.V. All rights reserved.

This view shows package overview with pins location. In the center are the peripherals.

To highlight the pin/peripheral configuration in the **Pins** and **Routing Details** views, right-click the pin or peripheral and select **Highlight**.

For BGA packages, use the **Resources** icon to see them.

- Green color indicates the routed pins/peripherals.
- Gray color indicates that the pin/peripheral is not routed.
- Dark Gray color indicates that the pin/peripheral is dedicated. It is routed by default and has no impact on generated code.

The view also shows the package variant and the description (type and number of pins).

The following icons are available in the toolbar:

Table 1	I 6 . '	Toolbar	options
---------	----------------	---------	---------

Icon	Description
ପ୍	Zoom in package image.
Q	Zoom out package image.
0	Rotate package image.
	Show pins as you can see it from the bottom. This option is available on BGA packages only.
	Show pins as you can see it from the top. This option is available on BGA packages only.
-	Show resources. This option is available on BGA packages only.
ÿ	Switch package.
	Package legend.
	Select the information displayed as pin labels. This option is not available on BGA packages.

Note: Depending on the processor package selected, not all views are available.

The Switch package icon launches Switch package for the Processor.

User Guide for Config Tools for i.MX

Switch package for the Processor 🛛 🖓 🖓	×
Available Processor Packages	
MIMXRT1052DVJ6B - MAPBGA 196 package MIMXRT1052CVJ5B - MAPBGA 196 package MIMXRT1052CVL5B - MAPBGA 196 package MIMXRT1052DVL6B - MAPBGA 196 package	
Please note that changing processor or package will result in configuration changes (settings changed or deleted) in case the new package/process doesn't provide the same resources (e.g. pins/clocks/peripherals).	
OK Cancel	
Figure 42. Switch package	

The **Switch package for the Processor** window shows list of available processor packages, showing package type and number of pins.

3.3.3 Peripheral Signals view

The **Peripheral Signals** view shows a list of peripherals and their signals. Only the **Peripheral Signals** and **Pins** view show the checkbox (allocated) with status.

Table 17. Status codes

Color code	Status
	Error
	Configured
	Not configured
	Warning
\checkmark	Dedicated: Device is routed by default and has no impact on the generated code.

IMXUG User guide

User Guide for Config Tools for i.MX

🗄 Pins 🐼 Peripheral Signals 🗙 👹 Power Groups 🍬 External User Signals 🗢 🗖	3
E I I W W → O O O I I I I I I I I I I I I I I I I	
> SAI1	
> SA12	
> SAI3	
> SAI4	
> SEMC	
SPDIF	
DCDC_GND, 0 - [K6] DCDC_GND0	
DCDC_GND, 1 - [L6] DCDC_GND1	
DCDC_IN, 0 - [M5] DCDC_IN0	
CDC_IN, 1 - [N5] DCDC_IN1	
DCDC_IN_Q - [L5] DCDC_IN_Q	
DCDC_LP, 0 - [T3] DCDC_LP0	
✓ DCDC_LP, 1 - [U3] DCDC_LP1 ✓ DCDC_PSWITCH - [P3] DCDC_PSWITCH	
✓ NVCC_SD1 - [D14] NVCC_SD1	
TEST_MODE - [T11] TEST_MODE	
VDD_SNVS_IN - [U12] VDD_SNVS_IN	
VDD_SOC_IN, 0 - [H8] VDD_SOC_IN0	
VDD_SOC_IN, 1 - [J8] VDD_SOC_IN1	
VDD_SOC_IN, 2 - [H9] VDD_SOC_IN2	
VDD_SOC_IN, 3 - [J9] VDD_SOC_IN3	
✓ VDD_SOC_IN, 4 - [H10] VDD_SOC_IN4 ✓ VDD_SOC_IN, 5 - [J10] VDD_SOC_IN5	
✓ VDD_SOC_IN, 6 - [K10] VDD_SOC_IN6	
✓ VDDA_ADC_3P3 - [J13] VDDA_ADC_3P3	
VSS, 0 - [A 1] VSS0	
VSS, 1 - [U1] VSS1	1
< >>	
Figure 43. Peripheral Signals view	

Use the checkbox to route/unroute the pins.

To highlight the pin/routing configuration about the peripheral in the **Package** and **Routing Details** views, rightclick the signal and select **Highlight**.

To route/unroute multiple pins, click the peripheral and select the options in the **Select signals** dialog.

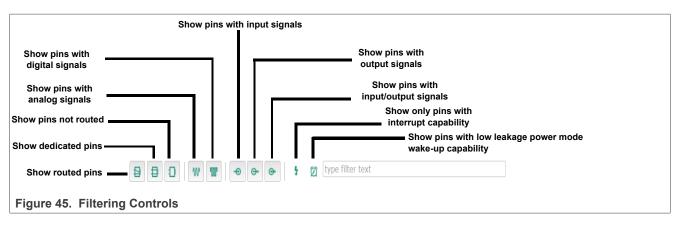
User Guide for Config Tools for i.MX

Peripheral GPIO3	×
All GPIO3 signals for routing:	
gpio_io, 00 - [N22] NAND_ALE; routed by default	^
gpio_io, 01 - [N24] NAND_CE0_B; routed by default	
gpio_io, 02 - [P27] NAND_CE1_B; routed by default	
gpio_io, 03 - [M27] NAND_CE2_B; routed by default	
gpio_io, 04 - [L27] NAND_CE3_B; routed by default	
gpio_io, 05 - [K27] NAND_CLE; routed by default	
gpio_io, 06 - [P23] NAND_DATA00; routed by default	
gpio_io, 07 - [K24] NAND_DATA01; routed by default	
gpio_io, 08 - [K23] NAND_DATA02; routed by default	
gpio_io, 09 - [N23] NAND_DATA03; routed by default	
gpio_io, 10 - [M26] NAND_DATA04; routed by default	
gpio_io, 11 - [L26] NAND_DATA05; routed by default	
gpio_io, 12 - [K26] NAND_DATA06; routed by default	
gpio_io, 13 - [N26] NAND_DATA07; routed by default	
gpio_io, 14 - [R22] NAND_DQS; routed by default	
gpio_io, 15 - [N27] NAND_RE_B; routed by default	
gpio_io, 16 - [P26] NAND_READY_B; routed by default	
gpio_io, 17 - [R26] NAND_WE_B; routed by default	
gpio_io, 18 - [R27] NAND_WP_B; routed by default	
gpio_io, 19 - [AB15] SAI5_RXFS; routed by default	
gpio_io, 20 - [AC15] SAI5_RXC; routed by default	~
Route All Unroute All	
Make sure pin/signal assignment is correct in Routing Det	ails view.
Don	e
ure 44. Select signals dialog	

3.3.3.1 Filtering in the Pins and Peripheral Signals views

The following image illustrates the filtering controls in the **Pins** and **Peripheral Signals** views.

User Guide for Config Tools for i.MX



Type any text to search across the table/tree. It will search for the pins/peripheral signals containing the specified text. You can also use wildcards "*" and "?" to help you filter results you want. Use "space" to search for multiple strings at the same time.

3.3.4 Routing Details view

In the **Routing Details** view, you can inspect and configure routed pins and internal signals. You can also configure the electrical properties of pins and view them. It displays the pad configuration available in a configuration where each pin is associated with the signal name and the function.

Note: The electrical features are configured only for pins in the table. For example, the routed pins.

The table is empty when a new configuration is created, which means no pin is configured. Each row represents configuration of a single pin and if there are no conflicts, then the code is immediately updated. For Boards/Kits, the pins are routed already.

ins Sig	gnals 🔎	type filter	text												=	C
Routing	Details fo	r M	2	• • •												
# Pe	eripheral	Signal	Arrow	Routed pin/signal	Label	Identifier	Direction	GPIO initial state	GPIO interrupt	Slew rate	Open drain	Drive strength	Pull select	Pull enable	Passive filter	T
17 A	DC0	VALTH	<-	[17] VREF_OUT	J2[1]	PWM	Input	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
63 A	DC0	SE, 7b	<-	[63] ADC0_SE7b	J2[8]	GND	Input	n/a	n/a	Fast	Disabled	Low	Pulldown	Disabled	Disabled	
<																>
				Details												

Add a row with the Add new row button in the view toolbar.

Configure the pin/signal by selecting the **Peripheral** first, then the required **Signal**, and finally, the pin to **Route to**.

Use the columns in the right side of the table to configure the electrical features.

You can also use the **Pins** and **Peripheral Signals** views to route pins and peripheral signals and view/modify the configuration in the **Routing Details** view. If the feature is not supported, *n/a* is displayed.

To highlight peripheral/pin information in the **Package** and **Pins** views, right-click the row and select **Highlight**.

To filter rows, type the text or the search phrase in the filter area in the view toolbar.

Note: When you enter the search text, it also searches the text in the full pin names displays rows that contain the search text.

User Guide for Config Tools for i.MX

To display pins or signals only, use the **Pins** and **Signals** buttons in the view toolbar.

To add a row to the end of table, click the **Add new row** button.

To remove the selected row, click the **Delete the selected row** button.

To delete a specific row or insert a new row at a given position, right-click and use the dropdown list commands.

To add a specific number of rows, enter the number in the field.

To clear the table, type 0.

To change the order of the rows, use the arrow icons to move one row up or down.

To filter table entries by text, enter the text string in the **type filter text** field.

To copy the row, right-click any cell in the row and select **Copy**. You can later paste the copied row into the **Routing Details** view of another functional group or configuration by right-clicking the table and choosing **Paste**.

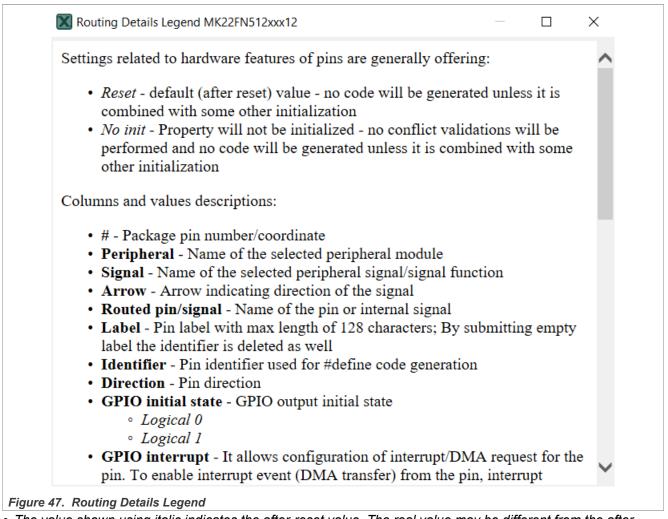
The gray background indicates read-only items.

The italic value indicates that the value is not configured and it shows the after-reset value and no code is generated, so the configuration relies on the after reset value or the values configured from the different functions.

Tip:

• Click the **Routing Details Legend** button in top right corner of the view to display a dialog explaining the fields.

User Guide for Config Tools for i.MX



[•] The value shown using italic indicates the after-reset value. The real value may be different from the after reset value, if configured in other functions.

Use the drop-down menu to select the required value.

- If you select the same value as the after-reset value, the tool will always generate code to set this feature. Use the drop-down "Reset" value to reset the value to its after-reset state.
- If an item does not support reset to after reset value, the **Reset** menu is not available.
- The first row shows pin number or coordinate on BGA package.

3.3.4.1 Labels and identifiers

You can define the label of any pin that can be displayed in user interface for ease of identification.

Boards and kits have pre-defined labels. However, it is also possible to define a pin label listed in the **Pins** and **Routing Details** views.

To set\update the Labels and Identifier columns visibility, select Edit> Preferences from the Menu bar, and select the Show pin label & identifier table columns (Pins tool) checkbox.

ins S	ignals 🔎 t	ype filter text									
Routing	g Details for	r init 14	• •	▲							
#	Peripheral	Signal	Arrow	Routed pin/signal	Label	Identifier	Power group	Direction	Pull Resistors Enable Field	Control IO ports PS	Open Drain Enable Field
AC27	ENET1	enet_mdc	->	[AC27] ENET_MDC	ENET_MDC	ENET_MDC 🔺	NVCC_ENET (0V)	Output	0b0: Disabled	0b0: Disabled	0b0: Disabled
AB27	ENET1	enet_mdio	<->	[AB27] ENET_MDIO	ENET_MDIO	ENET_MDC	VCC_ENET (0V)	Input/Output	0b0: Disabled	0b0: Disabled	ObO: Disabled
AF25	ENET1	enet_rgmii_td, 3	->	[AF25] ENET_TD3	ENET_TD3	Not Spec Pin id	dentifier used for #	define code ge	neration abled	0b0: Disabled	0b0: Disabled
AG25	ENET1	enet_rgmii_td, 2	->	[AG25] ENET_TD2	ENET_TD2	ENET_TD2	NVCC_ENET (0V)	Output	0b0: Disabled	0b0: Disabled	ObO: Disabled
AF26	ENET1	enet_rgmii_td, 1	->	[AF26] ENET_TD1	ENET_TD1	ENET_TD1	NVCC_ENET (0V)	Output	0b0: Disabled	0b0: Disabled	ObO: Disabled
AG26	ENET1	enet_rgmii_td, 0	->	[AG26] ENET_TD0	ENET_TD0	ENET_TD0	NVCC_ENET (0V)	Output	0b0: Disabled	0b0: Disabled	ObO: Disabled
<											>

Figure 48. Labels and Identifiers

The pin identifier is used to generate the #define in the pin_mux.h file. However, it is an optional parameter. If the parameter is not defined, the code for #define is not generated. Additionally, you can define multiple identifiers, using the ";" character as a separator. You can also set the identifier by typing it directly into the cell in the **Identifier** column in the **Routing Details** views.

PI	CSI0_PIACEK	CSID_PIXCEK	
🔽 R1	GPIO_17	SPDIF_OUT/PCIE_RST_B/J504[40]	PCIE_RST_B
🗹 T1	GPIO_2	KEY_ROW6/USR_DEF_RED_LED/J508[35]	USR_DEF_RED_LED;KEY_ROW
U1	LVDS0_TX0_P	LVDS0_TX0_P	
V1	LVDS0_TX2_P	LVDS0_TX2_P	
14/4	11/000 7/0 0		

In this case, it is possible to select from values if the pin is routed. See the **Identifier** column in the **Routing Details** view.

#	Peripheral	Signal	Route to	Label	Identifier		Po
T1	GPIO1	gpio, 2	GPIO_2	KEY_ROW6/USR_DEF_RED_LED/J508[35]	USR_DEF_RED_LED	*	NV
					USR_DEF_RED_LED		
					KEY_ROW		
					Not Specified		

A check is implemented to ensure whether the generated defines are duplicated in the pin_mux.h file. These duplications are indicated in the identifier column as errors. See <u>Identifier errors</u>

louted P	ins for init_gp	io_pins	72 🔁 🔕	3 🔼 🗹	* BE CAREFUL MODIFYING THIS COMMENT - IT IS YAML SETTINGS FOR TOOL */
#	Peripheral	Signal	Route to	Label	Identifier #include "pin mux.h"
Τ4	GPIO1	gpio, 1	GPIO_1	KEY_ROW5/USR_DEF_GRN_LED/J508[81]	Not Specified
🧿 T1	GPIO1	gpio, 2	GPIO_2	KEY_ROW6/USR_DEF_RED_LED/J508[35]	KEY_ROW A
R6	GPIO1	gpio, 4	GPIO_4	KEY_COL7/KEY_VOL_UP/J508[58]	KEY_ROW
R4	GPIO1	gpio, 5	GPIO_5	KEY_ROW7/KEY_VOL_DN/J508[27]	USP Pin identifier used for #define code generation. Use Pins view table to define it.
A21	GPIO1	gpio, 16	SD1_DAT0	CSI0_PWN	Not
B21	GPIO1	gpio, 18	SD1_CMD	ACCL_INT_IN	ACCL ERRORS:
C20	GPIO1	gpio, 17	SD1_DAT1	CSI0_RST_B	CSID F The identifier is duplicated in function(s) init_gpio_pins. This can lead to duplicated #defines in the generated header file(s).
E19	GPIO1	gpio, 19	SD1_DAT2	CSI_PWN	CSLPWvvv
3 T2	GPIO1	gpio, 9	GPIO_9	KEY_COL6/MICROPHONE_DET/J508[56]	* BE CAREFUL MODIFYING THIS COMMENT - IT IS YAML SETTINGS FOR TOOL
D20	GPIO1	gpio, 20	SD1_CLK	CSI_RST_B	CSL_RST_B */
W23	GPIO1	gpio, 24	ENET_RX_ER	USB OTG ID	USB_OTG_ID /*FUNCTION************************************

Figure 51. Identifier errors

IMXUG

© 2023 NXP B.V. All rights reserved.

You can also select the pin to use in a given routing from the **Routing Details** view. However, the identifier must be a valid C identifier and must be used in the source code.

Punctional group properties			×
Functional groups 💽 🛐 🗶 🔨 💌	Name:	init_gpio_pins	
P init_audmux_pins		Set custom #define prefix	
P init_can_pins	Prefix:	GPIO_PINS_	
P init_gpio_pins	Core:	Cortex-A9 (Core #0)	-
P init_ccm_pins P init_ecspi pins	Description:	Configures pin routing and optionally pin electrical features.	
Figure 52. Pins macros prefix			
Figure 52. Fins macros prenx			

If multiple functions are used, each individual function can include a special prefix. Check the **Pins** > **Functional Group Properties** > **Set custom #define prefix** checkbox to enter prefix of macros in particular function used in the generated code of the pin_mux.h file. Entered prefix text must be a C identifier. If unchecked, the **Function name** is used as a default prefix.

3.3.5 Expansion Header

In the **Expansion Header** view, you can add and modify an expansion header configuration, map the connectors, and route the pin signals. You can also import and apply an expansion board to the header.

Certain boards, such as LPCXpresso55S69, come with preconfigured expansion headers.

📋 Package 📋 Expansion Header 🗙
Labels Expansion header 👻 🧕 🧲
R NC NC NC NC NC 3.3V NC 3.3V NC 3.3V NC SV NC GND NC GND NC GND NC SV RC A3 P19[1] A3 P19[1] A5

The expansion header is not automatically preset for every supported device. If the header is not preconfigured, follow these steps to create and modify an expansion header configuration:

IMXUG

- 1. Open the view by selecting Views >Expansion Header from the Toolbar.
- 2. Add a header by selecting the Add button in the view toolbar.
- 3. In the Add New Expansion Header window, select the Header type from the drop-down list.

🙆 Add New Exp	oansion Header	×
Header name: Header type:	Breakout Board Expansion Connector	
Connectors n	Breakout Board Expansion Connector Freedom Header (Arduino compatible) LPCXpresso V3 (Arduino compatible) LPCXpresso V3 Mirrored (Arduino compatible) mikroBUS(TM) Pmod(TM)	
	Arduino Header	
	OK Cancel	

Figure 54. Adding new expansion header

4. Name the header and map the connectors.

@ A	Add New Exp	pansion Header	Х
	г		_
Hea	ader name:	Example	
Hea	ader type:	Arduino Header 🗸 🗸	•
Co	onnectors m	napping	
C	1 connector	: A	
c	2 connector	: В	
C	3 connector	: C	
0	4 connector	: D	
		OK Cancel	
Figure 55. Adding new expan	sion heade	r	
5. Select OK .			

© 2023 NXP B.V. All rights reserved.

Expansion Header view now displays the connector layout. You can point your cursor over the pins to display additional information. Right-click the pin to display a shortcut menu of additional options.

📋 Package 📋 Expans					
Labels Expansion he	ader 🗸 🧕 🤤 Header	LPCXpresso V3 (Arduino compatible) 🗸	🕂 😣 🗐 🗈	No expansion board is selected	0 😣 🗉
				۹.	
				D15 NC	
	_				
				D14 NC	
NC	NC			3.3V P 17 (6)	
NC	NC			GND P17[6]	
NC	Connector Pin: P16[4] Not connected			D 13 P 17 (10)	
NC	³² The connector pin is not	connected to any processor pin or extern	nal signal.	D 12 P 17 [12]	
NC R	ESET			D11 P17[19]	
NC	3.3V			D 10 P 17 [16]	
NC	5V				
	50			D9 P 17 [18]	
NC (IND	pansion Header: LPCXpresso V3 (Arduin		D8 P 17 [20]	
NC	SND	pansion Header: LPCApresso V3 (Arduin	o compatible)		
NC	sv			D7 P 18[2]	
				D6 P 18[4]	
1 - C				D5 P18[6]	
NC	AD				
NC	A1			D 4-LE P 18(9)	
NC	A2			D3-LE P 18(10)	
				D2 P 18[12]	
ព្រុម ។	A3				
P 19[9]	AL			D1 P18[14]	
P 19(1 ()	A5			DO P 18[16]	
				P 18[17] P 18[18]	
				P 18[19] P 18[20]	
				· · · · · · · · · · · · · · · · · · ·	

Figure 56. Expansion header

6. To map the header pin to processor pin, right-click the header pin and select Connect.

7. In the Connector Pin dialog, select the processor pin/external signal from the list and click OK.

IMXUG

User Guide for Config Tools for i.MX

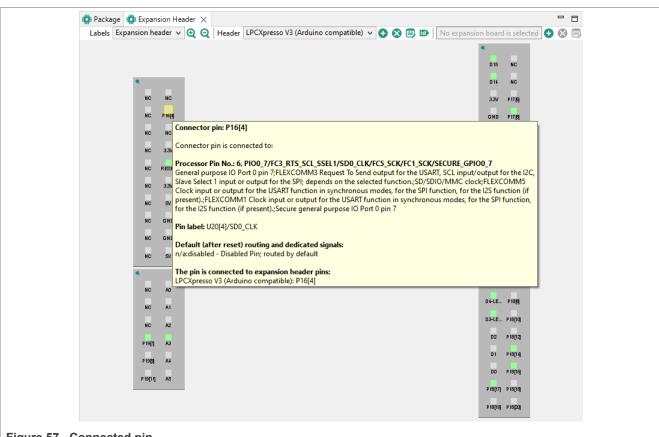


Figure 57. Connected pin

- 8. To route the pin, right-click the header pin and select **Route**.
- 9. In the Pin dialog, select the signal from the list and click OK.
- The connector pin is now routed.

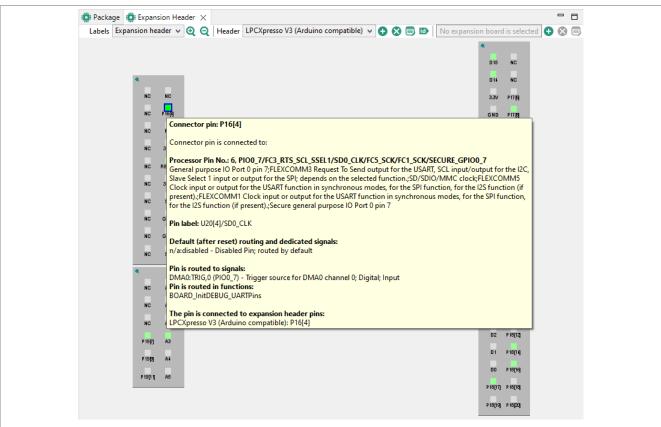


Figure 58. Routed pin

You can create more than one expansion header configuration. Switch between the configurations in the view's drop-down list.

To highlight the pin/routing configuration in the **Pins** and **Routing Details** views, right-click the connector pin and select **Highlight**.

Modify the configuration parameters at any time by selecting the **Edit** button. Information in the **Pins** view is updated automatically. Pin connections between the header and the processor and their labels can be locked to prevent any modifications.

Remove a configuration by selecting the **Remove** button.

Use the **Label** drop-down list to switch between display information for header, board, and routing.

The **ID** button allows setting expansion header pin labels as processor pin identifiers. Upon user selection, the new identifiers can be also explicitly selected.

3.3.5.1 Expansion Board

In the **Expansion Header** view, you can also apply an expansion board to an already created expansion header. The expansion board configuration can be imported into Pins tool in the form of an XML file. Based on the chosen processor, the tool will then recommend adequate routing.

Note: Only a single expansion board can be configured per expansion header.

- 1. In the Expansion Header view, click the Apply expansion board to the selected header. Alternatively, select Pins>Apply expansion board from the Menu bar.
- 2. In the **Apply expansion board** dialog, click **Browse** to locate the XML file with expansion board information and click **OK**.

User Guide for Config Tools for i.MX

X	Apply expansion b	oard X
[Board file (.xml): sktop\FRDM K64 C xpansion header w	lick shield.xml Browse here to apply the board:
	FRDM Expansion H	eader 🗸
	Board Properties	
	Board name:	FRDM K64 Click shield
	Header reference:	frdm_arduino
	Adapter headers:	Mikro BUS 1 Mikro BUS 2
	OK	Cancel
Figure 59. Apply expansion board		

- 3. Click **OK** to apply the expansion board.
- 4. On the next page, choose if you want to create a new functional group for the expansion board, or modify an existing functional group. In the latter case, use the dropdown list to select from available functional groups.
- 5. In the **Expansion Board Routing** table, inspect the suggested routing of expansion board pins. If you want to change the route of a pin, click the pin cell in the **Route** column and select the signal in the **Connector pin** dialog and click **Done**.

User Guide for Config Tools for i.MX

xpansion bo	<i>i</i> functional g	roup	MikroE Tester	MikroE_Tester_click_board_1				
·		roup.						
elect functio	nal group:		BOARD_InitD	EBUG_UARTPins	\			
Expansion bo	ard pins rout	ing						
Header Pin	Board pin	Sigr	al type	Route	Description			
NC	AN	ana	og	n/a (not connect	Analog			
NC	RST	gpio)	n/a (not connect	Reset			
NC	CS	gpio)	n/a (not connect	Chip Select			
NC	SCK	i2c_	clock	n/a (not connect	SPI Clock			
NC	SDO	digi	tal_out	n/a (not connect	SPI Data Out			
NC	SDI	digi	tal_in	n/a (not connect	SPI Data In			
3.3V	3.3V	ром	er_supply_3	n/a (external)	Power supply +3.3V			
GND	GND	grou	und	n/a (external)	Ground			
NC	PWM	gpio)	n/a (not connect	PWM			
NC	INT	inte	rrupt_out	n/a (not connect	Interrupt			
NC	TX	uart	_tx	n/a (not connect	UART Transmit			
NC	RX	uart	_rx	n/a (not connect	UART Receive			
NC	SCL	i2c_	clock	n/a (not connect	I2C Clock			
NC	SDA	digi	tal_out	n/a (not connect	I2C Data			
5V	5V	ром	er_supply_5V	n/a (external)	Power supply +5V			
GND	GND	grou	und	n/a (external)	Ground			
Populate ider	ntifiers for coo	de: U	sing expansion	header names Apply	✓ Cancel			

- 6. Choose how you want to populate identifiers for code. Following options are available:
 - Expansion header names
 - Expansion board names
 - None
- 7. Click **Apply** to apply the settings.

You can change the expansion board signal routing at any time by clicking the **Configure routing for** expansion board button in the Expansion Header view.

3.3.6 Power groups

If your processor supports power groups, an additional tab will appear next to Pins and Peripheral Signals.

Note: This feature is not supported for all devices.

User Guide for Config Tools for i.MX

📰 Pins 🛛 Peripheral	l Signals 👹 Power Groups × 斗t € External User Signals 🗖	• 🗖
₽ type filter text		
Power Group Name	Voltage Level [V]	^
ADC_VREFH	0.0	
DAC_OUT	0.0 Power group 'ADC_VREFH'.	
DCDC_ANA	0.0	
DCDC_ANA_SENSE	0.0	
DCDC_DIG	0.0	
DCDC_DIG_SENSE	0.0	
DCDC_GND	0.0	
DCDC_IN	0.0	
DCDC_IN_Q	0.0	
DCDC_LN	0.0	
DCDC_LP	0.0	
DCDC_MODE	0.0	
DCDC_PSWITCH	0.0	
NVCC_DISP1	0.0	
NVCC_DISP2	0.0	
NVCC_EMC1	0.0	
NVCC_EMC2	0.0	
NVCC_GPIO	0.0	
NVCC_LPSR	0.0	
NVCC_SD1	0.0	
NVCC_SD2	0.0	
NVCC_SNVS	0.0	
VDDA_1P0	0.0	
VDDA_1P8_IN	0.0	
VDDA_ADC_1P8	0.0	
VDDA_ADC_3P3	0.0	
VDD_LPSR_ANA	0.0	
VDD LPSR DIG	0.0	×

Figure 61. Selecting power group

3.3.7 External User Signals view

This view allows the user to define a custom description of the signals. An External User Signal has a defined unique ID within the table, pins to which it is connected, and any amount of additional text information. All of it can be customized.

E	Pins	🐼 Peripheral Signals 🛛	😼 Power Groups	The External User Signals $ imes $	- 0	
	⊕ ⊗	P type filter text				
	ID	Connected pins	Description			
	signal 1	[B1] GPIO_EMC_15	User descripti	on		
	signal 2	[C1] GPIO_EMC_21				
Figure 62. External User Signa	als vi	iew				

Connecting to a pin(s) can be done from a context menu of the selected signal. Multiple pins can be connected to the signal as well as multiple signals can be connected to the pin. When some signals are defined, the External User Signals column is added to the **Pins** view. The connection between pins and signals can be also done from there.

Additional columns can be specified using the table header context menu.

	Pins	🐼 Peripheral Signals 💧	Power Groups 🏄 E	xternal User Signals $ imes$	- 0
	⊕ ⊗	<mark>≣ ד</mark> א			
	-	Connected pins [B1] GPIO_EMC_15 [C1] GPIO_EMC_21	Description User description	Add column Remove column Rename column	
Figure 63. External User Sign	als h	eader context	menu		

The button with the **Routing Details** view icon can be used to add routed pins to the table or to display columns from **Routing Details** view in the **External User Signals** view.

Pins	🐼 Peripheral Signals 🛛 👹	Power Groups	३ी∉ External User Signals →			
€⊗	달 - ∕의					
ID	Add Routed Pins					
signal 1	Routing Details Colur	nns t	tion			
signal 2	[C1] GPIO_EMC_21					

Figure 64. External User Signals view

The **Add Routed Pins** option collects routed pins from all functional groups and adds them to the table when they are not already present. A new signal is created for each added pin.

Select Routing Details Columns to open a dialog with Routing Details columns. The selected columns will be displayed in the table. Those columns are read-only and they always reflect actual values in the **Routing Details** view for all functional groups.

User Guide for Config Tools for i.MX

🔀 Routing Details Columns	×
Select Routing Details columns that are visible in the External	User Signals table:
#	
Peripheral	
Signal	
Arrow	
Routed pin/signal	
Label	
Identifier	
Power group	
Direction	
GPIO initial state	
GPIO interrupt	
Software Input On	
Hysteresis enable	
Pull Up/Down Config	
Pull/Keeper select	
Pull/Keeper enable	
Open drain	
Speed	
Drive strength	
Slew rate	
	Done

Figure 65. Routing Details columns dialog

E Pins	🐼 Peripheral Signals 🛛 성	Power Groups 🏻 🕯 🕻 E	xternal User Signals	; X	- 8
• •	P type filter text				
ID	Connected pins	Description	Direction	Pull/Keeper select	Drive strength
signal 1	[B1] GPIO_EMC_15	User description	Output	Keeper	R0/6
signal 2	[C1] GPIO_EMC_21		Input; Output	Keeper	R0; R0/3; R0/6
ure 66. R	Routing Details table				

When needed, External User Signals can be also exported to CSV and then imported to another configuration. Merging of signals is not supported so when some signals are defined for the configuration, they are replaced by imported signals.

IMXUG

3.3.8 Functions

Functions are used to group a set of routed pins, and they create code for the configuration in a function which then can be called by the application.

The tool allows to creates multiple functions that can be used to configure pin muxing.

The usage of pins is indicated by 50% opacity in **Pins**, **Peripheral Signals**, and **Package** views. Each function can define a set of routed pins or re-configure already routed pins.

When multiple functions are specified in the configuration, the package view primarily shows the pins and the peripherals for the selected function. Pins and peripherals for different functions are shown with light transparency and cannot be configured, until switched to this function.

3.3.9 Highlighting and color coding

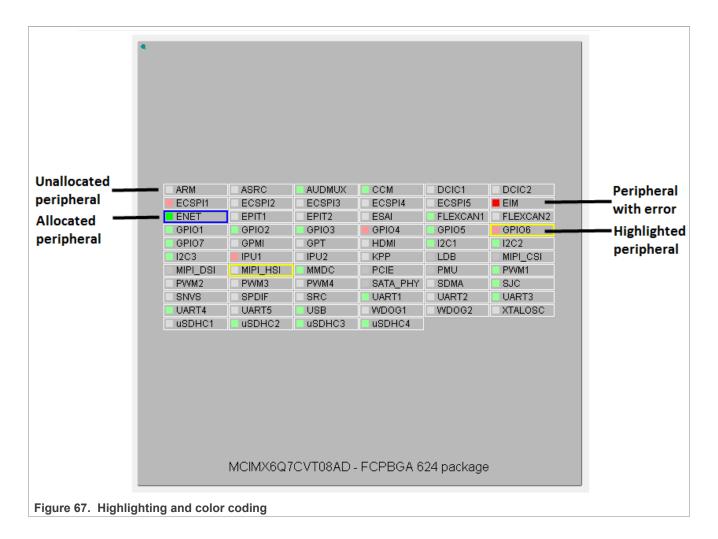
You can easily identify routed pins/peripherals in the package using highlighting. By default, the current selection (pin/peripheral) is highlighted in the **Package** view.

- The pin/peripheral is highlighted by yellow border around it in the **Package** view. If the highlighted pin/ peripheral is selected, then it has a blue border around it.
- Red indicates that the pin has an error.
- Green indicates that the pin is muxed or used.
- Light gray indicates that the pin is available for mux, but is not muxed or used.
- Dark gray indicates that the pin/peripheral is dedicated. It is routed by default and has no impact on generated code.

NXP Semiconductors

IMXUG

User Guide for Config Tools for i.MX



User Guide for Config Tools for i.MX

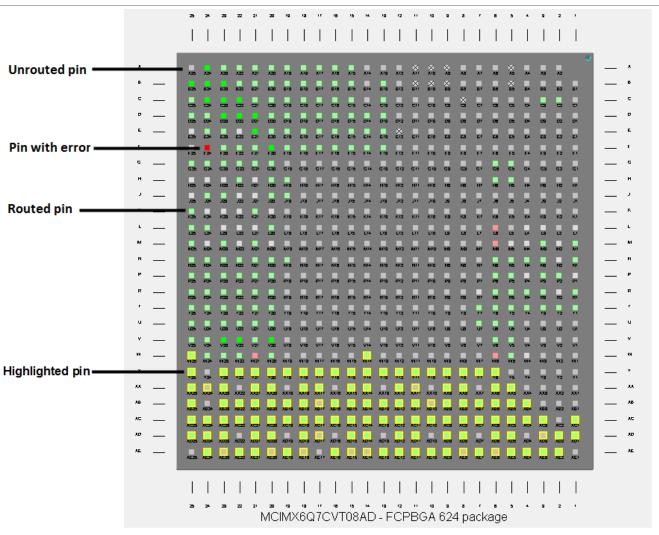


Figure 68. BGA package on pins side

pe filter	text							
Routed F	Pins for init_ecs	spi_pins	7 💽	8 🔼 🗹				
#	Peripheral	Signal	Route to	Label	Identifier	Power group	Direction	Software Input Or
U7	ECSP11	miso	KEY_COL1	CSPI1_MISO	SPI_IN	NVCC_GPIO (3.3V)	Input/Output	0b0: Disabled
V6	ECSP11	mosi	KEY_ROW0	CSPI1_MOSI	SPI_OUT	NVCC_GPIO (3.3V)	Input/Output	0b0: Disabled
W5	ECSP11	sclk	KEY_COL0	CSPI1_CLK	SPI_CLK	NVCC_GPIO (3.3V)	Input/Output	0b0: Disabled
📀 U6	ECSP11	ss0	[×] KEY_ROW1	CSPI1_CS0	[#] SPI_CS0	NVCC_GPIO (3.3V)	Output	0b0: Disabled
📀 U6	UART5	rxd_mux	[®] KEY_ROW1	CSPI1_CS0	[©] SPI_CS0	NVCC_GPIO (3.3V)	Input	0b0: Disabled
🚷 A18	×	×	NANDF_D0	SD3_CD_B/J507[14]	SD3_CD_B		n/a	0b0: Disabled
8	×	×	×		n/a		n/a	n/a
•	Pins not	configured	l Con	flicting pins				•

IMXUG

#	Peripheral	Signal	Route to	Label	Identifier	Direction	Slew rate
۵ 💧	GPIOE	GPIO, 26	PTE26	J2[1]/D12[4]/LEDRGB_GREEN	Not Specified	Input	Slow
🙆 71	FTM0	CH, 0	FTM0_CH0	J1[5]	ªn∕a	Output	Fast

Figure 70. Warnings

- Package view
 - Click the peripheral or use the pop-up menu to highlight peripherals:
 - and all allocated pins (to selected peripheral).
 - or all available pins if nothing is allocated yet.
 - Click the pin or use the pop-up menu to highlight the pin and the peripherals.
 - Click outside the package to cancel the highlight.
- Peripherals / Pins view
 - The peripheral and pin behaves as described above.

3.4 Errors and warnings

The Pins Tool checks for any conflict in the routing and also for errors in the configuration. Routing conflicts are checked across all **INIT** functions (default initialization functions). It is possible to configure different routing of one pin in different functions (not INIT functions) to allow dynamic pins routing reconfiguration.

pe fil	lter text										
Route	ed Pins for BOA	RD_Init 6 💽 🗵									
#	Peripheral	Signal	Route to	Label	Identifier	Direction	GPIO initial state	Mode	Slew rate	Invert	Open drain
3	FLEXCOMM0	RXD_SDA_MOSI_DATA	FC0_RXD_SDA_MOSI_DATA	P17[17]/P24[1]/PIO1_5_GPIO	n/a	Not Specified	n/a	Inactive	Standard	Disabled	Disabled
5	FLEXCOMM0	TXD_SCL_MISO_WS	FC0_TXD_SCL_MISO_WS	R80/P18[9]/LEDB/PWM_ARD	LED_RED	Not Specified	n/a	Inactive	Standard	Disabled	Disabled
1	PMC	VREFINPUT_1	VDDA	VDDA_TARGET	n/a	Input	n/a	n/a	n/a	n/a	n/a
9	USBFSH	USB_VDD	USB0_3V3	VDD_TARGET_3.3	n/a	Input	n/a	n/a	n/a	n/a	n/a
<mark>0</mark> 2	CTIMER3	CAPTURE, 3	CT_INP10	U14[12]/SWO_TRGT	Not Specified	Input	n/a	Inactive	Standard	Disabled	Disabled
Ø 1	CTIMER3	CAPTURE, 3	CT_INP4	P19[2]/P23[1]/ADC0_N	n/a	Input	n/a	Inactive	Standard	Disabled	Disabled

Figure 71. Error and warnings

If an error or warning is encountered, the conflict in the **Routing Details** view is represented in the first column of the row and the error/warning is indicated in the cell, where the conflict was created. The last two rows in the figure above show the peripheral/signal where the erroneous configuration occurs. The detailed error/warning message appears as a tooltip.

For more information on error and warnings color, see the Highlighting and Color Coding section.

3.4.1 Incomplete routing

A cell with incomplete routing is indicated by a red background. To generate proper pin routing, click the dropdown arrow and select the suitable value. A red decorator on a cell indicates an error condition.

IMXUG

#	Peripheral	Signal	Rou	te to Label	Identifier	Power group
8	ECSP11	8	8	Name of the pin or	internal signal.	
				ERRORS:		
				Unassigned pin		

The tooltip of the cell shows more details about the conflict or the error, typically it lists the lines where conflict occurs.

You can also select **Pins > Automatic Routing** from the Main menu to resolve any routing issues.

Note: Not all routing issues can be resolved automatically. In some cases, manual intervention is required.

3.4.2 Power groups voltage level conflicts

The Pins tool provides information about possible voltage level conflicts when the peripheral signals routed pins are configured from a different power groups and the power groups have different voltage level value set in **Power groups** view.

In case of a potential voltage level conflict, a warning is displayed - a useful feature for hardware board designers.

		1 1115 111	voltage-level o	onnet					
Routed	Pins for BOAR	D_InitPins	5 💽	8 ^ 🖌					
#	Peripheral	Signal	Route to	Label	Identifier	Power group	Direction	Software Input On Field	Hyst
🙆 M4	UART5	rxd_mux	CSI0_DAT14	UART5_RX	UART_RXD	NVCC_CSI (3.3V)	Input	0b0: Disabled	0b1:
💧 U6	UART5	txd_mux	KEY ROW1	UART5 TX	UART TXD	NVCC GPIO (1.5V)	Output	0b0: Disabled	<u>06</u> 1:
				iS:		ng power group(s) use	different volta	ge level: NVCC_CSI(item 'Rou	ute
					Warning	Voltage leve	1		

3.5 Code generation

If the settings are correct and no error is reported, the code generation engine instantly regenerates the source code. You can view the resulting code the **Code Preview** view of the **Pins** tool.

Code Preview automatically highlights differences between the current and immediately preceding iteration of the code. You can choose between two modes of highlighting by clicking the **Set viewing style for source differences**. You can also disable highlighting altogether from the same dropdown menu. Such features as Copy, Search, Zoom-in, Zoom-out, and Export source are available in the **Code Preview** view. The search can also be invoked by CTRL+F or from the context menu.

For multicores, the sources are generated for each core. Appropriate files are shown with @Core #{number} tag.

Note: The tag name may be different depending on the selected multi-core processor family/type.

You can also copy and paste the generated code into the source files. The view generates code for each function. In addition to the function comments, the tool configuration is stored in a YAML format. This comment is not intended for direct editing and can be used later to restore the pins configuration.

Note: Code Preview view contains the Export button and possibility of exporting sources - link to export wizard - Pins Tool > Export Source Files option-allowing export sources per used cores in multi-core enabled configuration.

YAML configuration contains configuration of each pin. It stores only non-default values.

Tip: For multicore processors, it will generate source files for each core. If processor is supported by SDK, it can generate BOARD_InitBootPins function call from main by default. You can specify "Call from BOARD_InitBootPins" for each function, in order to generate appropriate function call.

3.6 Using pins definitions in code

The Pins tool generates definitions of named constants that can be leveraged in the application code. Using such constants based on user-specified identifiers allows you to write code which is independent of configured routing. In the case you change the pin where the signal is routed, the application will still refer to the proper pin.

For example, when the *LED_RED* is specified an identifier of a pin routed to *PTB22*, the following defines are generated into the pin_mux.h:

#define BOARD_LED_RED_GPIO GPIOB /*!<@brief GPIO device name: GPIOB */
#define BOARD_LED_RED_PORT PORTB /*!<@brief PORT device name: PORTB *'/
#define BOARD_LED_RED_PIN 22U /*!<@brief PORTB pin index: 22 */</pre>

The name of the define is composed from function group prefix and pin identifier. For more details, see <u>Section 2.4.3</u> and <u>Section 3.3.4.1</u> sections.

To write to this GPIO pin in application using the SDK driver (fsl_gpio.h), you can, for example, use the following code referring to the generated defines for the pin with identifier *LED_RED*:

GPIO_PinWrite(BOARD_LED_RED_GPIO, BOARD_LED_RED_PIN, true);

3.7 Full initialization of pins

In some cases, the default values are not reliable, as there may be code running before the application that modifies the pin configuration (for example, a bootloader). The option **Full initialization of pins** ensures that the initialization is fully done even for items that use after-reset state. This option is specific for each **Functional group** allowing to force full initialization of routing. **Full initialization of pins** is not enabled by default. When enabled, the electrical properties of existing routing are changed. The "Reset" values are changed to explicit values corresponding with them. When the option is disabled, the pins tool changes the values that are matching after-reset state to the "Reset" values.

3.8 Create Default Routing

If necessary, it is possible to create a new functional group that will route default signals to pins and internal signals. The functionality is available in **Pins -> Create Default Routing**. There the user can select:

• Whether all pins and signals will be routed, or only the ones that are not routed in other functional groups.

- The name of the new functional group.
- Whether the routing is created for pins and/or internal signals.

In the created functional group, the Full initialization function of the pins feature will be set.	The electrical
properties of pins will be set to their after-reset state.	

🔀 C	reate the Default Rou	ting	×
	ew functional group v default routing of the	vill be created. It will conta selected items.	in
Rout	te:	All routable items	*
Fund	ctional group name:	SetDefaultRouting	
Item	is to route:	✓ Pins ✓ Signals	
		OK Cancel	
Figure 74. Create the Default Routi	ng		

4 DDR Tool

This section introduces the DDR configuration and validation tool, which is an embedded componen of Config tools for i.MX.

The DDR tool provides two main functionalities: configuration and validation.

Supported devices are indicated in the new project configuration page.

Note: DDR tool is provided "as is" to aid customer capabilities of evaluating, debugging, and optimizing their designs. The results, or any part thereof, provided by the tool cannot be under any circumstances seen as a substitute for the traditional validation and compliance methods, which still need to be performed to declare compliance of the designs with the respective JEDEC standards.

4.1 Create a new DDR tool project

To use the DDR tool, you first must create a new project.

To create a new DDR tool project, follow these steps:

- 1. Open the Config tools for i.MX.
- 2. Choose Create a new standalone configuration for processor, board, or kit and click Next.
- 3. From Processors, choose one of the devices with DDR tool support and click Finish.
- 4. To open the DDR tool view, click the **DDR** tool icon.

IMXUG

User Guide for Config Tools for i.MX

Config Tools Overview		—	×
i Project opened for the first time. Click on a tool ic	con to select the tool.		
 Configuration - General Info Configuration is not saved on the disk 	 Configuration - HW Info Processor: MIMX8MM4xooKZ Part number: MIMX8MM4CVTKZ Core: Cortex-A53(core#0) SDK Version: ksdk2_0 	 Project No toolchain project detected 	
Pins Configures pin routing, including functional voltage/power rails, and run-time pin config ©	guration. DDR c	DR Tool provides user friendly configuration of initialization for ontrollers and allows you to validate the configuration using s validation scenarios.	
 Generated code board\pin_mux.c board\pin_mux.dts board\pin_mux.h 			
		Close and Update Code <u>Close</u>	•

Figure 75. Config Tools overview

5. To use the DDR tool, accept the Disclaimer.

	Disclaimer X
	All information provided is accurate to the best of NXP's knowledge and will not operate to increase NXP's warranty obligations. All information is provided "AS IS" and NXP makes no representation or warranty, express or implied, of accuracy, completeness, that products will be suitable for any specified use. NXP will not be liable for any damages or loss arising from, in connection with or incident to the information or assistance provided by NXP. Unless otherwise provided in a signed, written agreement, all sales transactions by NXP are subject to NXP's general terms and conditions of commercial sale: http://www.nxp.com/about/about-nxp/our-terms-and-conditions-of-commercial-sale:TERMSCONDITIONSSALE
	ОК
Figure 76. Discla	imer

4.2 DDR configuration

The DDR configuration provides a user-friendly graphical interface to configure the DDR interface and other associated subsystems. You can use it to change the DDR controller and PHY configuration when a different memory module is used to the configuration and to optimize the parameters associated with signal integrity.

4.2.1 Import initialization script

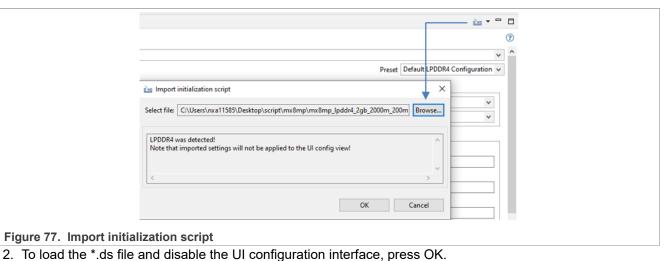
Import initialization script allows loading the initialization script provided by the **Register Programming Aid** (**RPA**) tool and bypassing the UI configuration. To obtain the latest RPAs, refer to the following link on <u>NPX</u> community.

1. To import the RPA initialization script, use the **Import initialization script** button and browse for the desired *.ds file

NXP Semiconductors

IMXUG

User Guide for Config Tools for i.MX



	5
Y DDRC	Preset Default LPDDR4 Configuration
✓ Device Information	
Memory type	LPDDR4
Density per channel per chip select (Gb)	8
Number of Channels	2
Number of Chip Selects used	1
Total DRAM density (Gb)	16
Number of ROW Addresses	16
Number of COLUMN Addresses	10

Figure 78. Disabled UI

3. The contents of the imported *.ds file is shown in **Code Preview**, *ddr_config.ds*

4.2.2 Import from target

Import from target allows loading the DDR initialization of an already configured working target and bypassing the UI configuration. Use this option only for devices with JTAG connection available.

User Guide for Config Tools for i.MX

			🔤 🔻 🗖	
				?
				¥
	Preset	Default DDR4 Co	nfiguration	~
🔤 Import from target	7	×		
Connections: Select connection type: ITAG USB O Ether USB device:	net	•	> > > >	
80. Import from target	ОК	Cancel	> > >	

4.2.3 Enable manual configuration

To switch back to UI configuration, press the button "Enable manual config".

	🔤 🥖 🖓 🗖	
	Preset Default LPDDR4 Configuration v	
Figure 81. Enable manual config		

4.2.4 UI configuration

The UI configuration allows you to change manually Device Information, PHY options, or Design-specific configuration. There are two modes available, **Basic** and **Advanced**.

Note: Advanced mode is only recommended for experienced users.

Basic mode allows you to configure the parameters that are design-dependent.

1. Device information

User Guide for Config Tools for i.MX

Device Information																
Memory type			PDDR	4												
Density per channel per chip select	(Gb)	8														
Number of Channels		2														
Number of Chip Selects used		1														
Total DRAM density (Gb)		1	6													
Number of ROW Addresses		1	6													
Number of COLUMN Addresses		1	0													
Number of BANK addresses		3														
Number of BANKS		8														
Bus Width		3	2													
Number of frequency setpoints		1														
Clock Cycle Freq (MHz)		1	500 N	ИНz												
Clock Cycle Time		6	66.66	7 ps												
igure 82. Device Information UI . Board data bus configuration fo	or LF	DDF	84													
✓ Channels																
#	<							В								>
DRAM data bus	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRAM data bus (User Input ->)	7	6	5	4	3	2	1	0	14	15	10	13	12	11	9	8
Byte lane	<			0				>	<			1				>
Data bus bits within byte lane	7	6	5	4	3	2	1	0	6	7	2	5	4	3	1	0
																>

3. UART port selection

UART Port	UART2	¥
Figure 84. UAR	T selection UI	
4. DBI selection	n (for LPDDR4)	
DBI	Disable	×

Figure 85. DBI selection

Inline ECC configuration (for devices with Inline ECC support) allows selection of the following parameters:
 a. Enable/Disable Inline ECC

User Guide for Config Tools for i.MX

ne ECC		E	Enabled			
re 86. Inlii	ne ECC					
b. ECC re	gion granularity					
region gra	nularity (divided by	v) 8				
region giu	indiancy (divided b)		·			
ro 97 EC(C region granularity					
		-		_	•	
 ECC pr 	rotection configurat	tion for e	each Main Memor	y Reg	ion	
The over	erview of the ECC	configura	ation is summariz	ved in	the ECC config bir	nary/non-binary aligne
		comgan				
section	•					
C config non-binary	v aligned					
						broken up into three equally sized non-contir
regions, with	each region containing the ECC par	ity section for the	at memory region. To avoid such a	situation, i	t is recommended to use a binary-aligr	ted density.
CC Memory Regio	n < >					
ice memory negro						
	COM DI LA FOOM	Plask 2				
Memory Block 0 E	CC Memory Block 1 ECC Memory B	DIOCK 2				
		DIOCK 2				
Memory Block 0 E		DIOCK 2				
ECC parity region		1	f each ECC Parity Region Section	Density of	each ECC Parity Region Section (MB)	ECC Parity Region Section memory attribut
ECC parity region	space n Section for Main Memory Region	1	f each ECC Parity Region Section		each ECC Parity Region Section (MB)	ECC Parity Region Section memory attribut INACCESSIBLE
ECC parity region ECC Parity Region ECC Parity Region	space n Section for Main Memory Region n 0 Selection	Start Address of	f each ECC Parity Region Section	Density of 32MB 32MB	each ECC Parity Region Section (MB)	
ECC parity region	space n Section for Main Memory Region n 0 Selection n 1 Selection	Start Address of 0x0BE000000	f each ECC Parity Region Section	321MB	each ECC Parity Region Section (MB)	INACCESSIBLE
ECC parity region ECC Parity Regio ECC Parity Regio ECC Parity Regio	space n Section for Main Memory Region n 0 Selection n 1 Selection n 2 Selection	Start Address of 0x0BE000000 0x0BC000000	f each ECC Parity Region Section	32MB 32MB	each ECC Parity Region Section (MB)	INACCESSIBLE INACCESSIBLE
ECC parity Region ECC Parity Region ECC Parity Region ECC Parity Region ECC Parity Region ECC Parity Region	space n Section for Main Memory Region n 0 Selection n 1 Selection n 3 Selection n 3 Selection	Start Address of 0x0BE000000 0x0BC000000 0x0BA000000	f each ECC Parity Region Section	32IMB 32IMB 32IMB	each ECC Parity Region Section (MB)	INACCESSIBLE INACCESSIBLE INACCESSIBLE
ECC parity region ECC Parity Region ECC Parity Regio ECC Parity Regio ECC Parity Regio ECC Parity Regio ECC Parity Regio	space n Section for Main Memory Region n 0 Selection n 1 Selection n 2 Selection n 4 Selection n 4 Selection	Start Address of 0x0BE000000 0x0BC000000 0x0BA000000 0x0BA000000 0x0B8000000	f each ECC Parity Region Section	32MB 32MB 32MB 32MB	each ECC Parity Region Section (MB)	INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE
ECC parity region ECC Parity Region ECC Parity Regio ECC Parity Regio ECC Parity Regio ECC Parity Regio ECC Parity Regio ECC Parity Regio	space n Section for Main Memory Region n 0 Selection n 1 Selection n 2 Selection n 3 Selection n 4 Selection n 5 Selection	Start Address of 0x0BE000000 0x0BC000000 0x0BA000000 0x0BA000000 0x0B8000000 0x0B8000000 0x0B6000000	f each ECC Parity Region Section	32MB 32MB 32MB 32MB 32MB	each ECC Parity Region Section (MB)	INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE
ECC parity region ECC Parity Region ECC Parity Regio ECC Parity Regio ECC Parity Regio ECC Parity Regio ECC Parity Regio	space n Section for Main Memory Region n 0 Selection n 2 Selection n 3 Selection n 4 Selection n 5 Selection n 6 Selection	Start Address of Dx0BE000000 0x0BE000000 0x0BE000000 0x0BA000000 0x0B8000000 0x0B6000000 0x0B6000000 0x0B6000000 0x0B6000000	f each ECC Parity Region Section	32MB 32MB 32MB 32MB 32MB 32MB	each ECC Parity Region Section (MB)	INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE
ECC parity region ECC Parity Regio ECC Parity Regio	space n Section for Main Memory Region n 0 Selection n 2 Selection n 3 Selection n 4 Selection n 5 Selection n 6 Selection	Start Address of 0x0BE000000 0x0BC000000 0x0BA000000 0x0BA000000 0x0B6000000 0x0B6000000 0x0B4000000 0x0B4000000 0x0B4000000	f each ECC Parity Region Section	32MB 32MB 32MB 32MB 32MB 32MB 32MB	each ECC Parity Region Section (MB)	INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE
ECC parity region ECC Parity Regio ECC Parity Regio	space n Section for Main Memory Region n 0 Selection n 2 Selection n 3 Selection n 4 Selection n 5 Selection n 6 Selection	Start Address of 0x0BE000000 0x0BC000000 0x0BA000000 0x0BA000000 0x0B6000000 0x0B6000000 0x0B4000000 0x0B4000000 0x0B4000000	f each ECC Parity Region Section	32MB 32MB 32MB 32MB 32MB 32MB 32MB	each ECC Parity Region Section (MB)	INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE
ECC parity region ECC Parity Regio ECC Parity Regio	space n Section for Main Memory Region n 0 Selection n 2 Selection n 3 Selection n 4 Selection n 5 Selection n 6 Selection ssible	Start Address of 0x0BE000000 0x0BC000000 0x0BA000000 0x0BA000000 0x0B6000000 0x0B6000000 0x0B4000000 0x0B4000000 0x0B4000000	f each ECC Parity Region Section	32MB 32MB 32MB 32MB 32MB 32MB 32MB	each ECC Parity Region Section (MB)	INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE
ECC parity region ECC Parity Regio ECC Parity Regio Always user acce	space n Section for Main Memory Region n 0 Selection n 1 Selection n 3 Selection n 4 Selection n 5 Selection n 5 Selection sible gion Space	Start Address of 0x08E00000 0x08C00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08B000000 0x08A00000 0x08B000000 0x08000000 0x08000000 0x08000000		32MB 32MB 32MB 32MB 32MB 32MB 32MB 32MB		INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE
ECC parity region ECC Parity Regio ECC Parity Regio Always user acce	space n Section for Main Memory Region n 0 Selection n 3 Selection n 4 Selection n 5 Selection n 5 Selection n 6 Selection ssible gion Space egion Start Address of each Main N	Start Address of 0x08E00000 0x08C00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08B000000 0x08A00000 0x08B000000 0x08000000 0x08000000 0x08000000	Density of each Main Memory Ro	32MB 32MB 32MB 32MB 32MB 32MB 32MB 32MB	ECC Protection Configuration for eac	INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE
ECC parity region ECC Parity Regio ECC Parity Regio Always user acce Main Memory Re Region 6	space n Section for Main Memory Region n 0 Selection n 2 Selection n 3 Selection n 5 Selection n 6 Selection ssible gion Space egion Start Address of each Main N 0x0A0000000	Start Address of 0x08E00000 0x08C00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08B000000 0x08A00000 0x08B000000 0x08000000 0x08000000 0x08000000	Density of each Main Memory Re 256MB	32MB 32MB 32MB 32MB 32MB 32MB 32MB 32MB	ECC Protection Configuration for eac PROTECTED	INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE
ECC parity region ECC Parity Regio ECC Parity Regio Always user acce Main Memory Re Region 6 Region 5	space n Section for Main Memory Region n 0 Selection n 1 Selection n 3 Selection n 3 Selection n 5 Selection gion Space gion Start Address of each Main N 0x040000000 0x090000000	Start Address of 0x08E00000 0x08C00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08B000000 0x08A00000 0x08B000000 0x08000000 0x08000000 0x08000000	Density of each Main Memory Re 256MB 256MB	32MB 32MB 32MB 32MB 32MB 32MB 32MB 32MB	ECC Protection Configuration for eac PROTECTED PROTECTED	INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE
ECC parity region ECC Parity Regio ECC Parity Regio ECC Parity Regio ECC Parity Regio ECC Parity Regio ECC Parity Regio ECC Parity Regio Always user acce Main Memory Re Region 6 Region 5 Region 4	space n Section for Main Memory Region n 0 Selection n 1 Selection n 2 Selection n 3 Selection n 4 Selection n 5 Selection n 5 Selection n 6 Selection n 5 Selection gion Space egion Start Address of each Main N 0x00000000 0x00000000	Start Address of 0x08E00000 0x08C00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08B000000 0x08A00000 0x08B000000 0x08000000 0x08000000 0x08000000	Density of each Main Memory Re 256MB 256MB 256MB	32MB 32MB 32MB 32MB 32MB 32MB 32MB 32MB	ECC Protection Configuration for eac PROTECTED PROTECTED PROTECTED	INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE ACCESSIBLE
ECC parity region ECC Parity Regio ECC Parity Regio Always user acce Main Memory Re Region 6 Region 5 Region 4 Region 3	space n Section for Main Memory Region n 0 Selection n 1 Selection n 3 Selection n 4 Selection n 5 Selection n 6 Selection n 6 Selection sible gion Space sgion Start Address of each Main N 0x0A0000000 0x030000000 0x070000000 0x070000000	Start Address of 0x08E00000 0x08C00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08B000000 0x08A00000 0x08B000000 0x08000000 0x08000000 0x08000000	Density of each Main Memory Re 256MB 256MB 256MB	32MB 32MB 32MB 32MB 32MB 32MB 32MB 32MB	ECC Protection Configuration for eac PROTECTED PROTECTED PROTECTED PROTECTED PROTECTED	INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE ACCESSIBLE
ECC parity region ECC Parity Regio ECC Parity Regio Always user acce Main Memory Re Region 6 Region 5 Region 3 Region 2	space n Section for Main Memory Region n 0 Selection n 1 Selection n 3 Selection n 5 Selection n 5 Selection ssible gion Space egion Start Address of each Main N 0x0A000000 0x08000000 0x0800000 0x08000000 0x0800000 0x0800000 0x0800000 0x0800000 0x08000000 0x08000000 0x08000000 0x08000000 0x0800000 0x08000000 0x0800000 0x0800000 0x0800000 0x0800000 0x0800000 0x0800000 0x0800000 0x0800000 0x0800000 0x080000 0x0800000 0x0800000 0x0800000 0x0800000 0x0800000 0x0800000 0x080000 0x0800000 0x0800000 0x0800000 0x0800000 0x0800000 0x08000000 0x08000000 0x08000000 0x08000000 0x080000000 0x08000000 0x080000000 0x080000000 0x080000000 0x08000000 0x080000000 0x080000000 0x080000000 0x080000000 0x0800000000	Start Address of 0x08E00000 0x08C00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08B000000 0x08A00000 0x08B000000 0x08000000 0x08000000 0x08000000	Density of each Main Memory Re 256MB 256MB 256MB 256MB 256MB	32MB 32MB 32MB 32MB 32MB 32MB 32MB 32MB	ECC Protection Configuration for eac PROTECTED PROTECTED PROTECTED PROTECTED PROTECTED PROTECTED	INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE
ECC parity region ECC Parity Regio ECC Parity Regio Always user acce Main Memory Re Region 6 Region 7 Region 2 Region 1	space n Section for Main Memory Region n 0 Selection n 1 Selection n 3 Selection n 5 Selection ssible gion Space egion Start Address of each Main N 0x0A000000 0x0000000 0x0000000 0x0000000 0x000000	Start Address of 0x08E00000 0x08C00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08B000000 0x08A00000 0x08B000000 0x08000000 0x08000000 0x08000000	Density of each Main Memory Re 256MB 256MB 256MB 256MB 256MB 256MB	32MB 32MB 32MB 32MB 32MB 32MB 32MB 32MB	ECC Protection Configuration for eac PROTECTED PROTECTED PROTECTED PROTECTED PROTECTED PROTECTED PROTECTED	INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE
ECC parity region ECC Parity Regio ECC Parity Regio Always user acce Main Memory Re Region 6 Region 5 Region 3 Region 2	space n Section for Main Memory Region n 0 Selection n 1 Selection n 3 Selection n 5 Selection n 5 Selection ssible gion Space egion Start Address of each Main N 0x0A000000 0x08000000 0x0800000 0x08000000 0x0800000 0x0800000 0x0800000 0x0800000 0x08000000 0x08000000 0x08000000 0x08000000 0x0800000 0x08000000 0x0800000 0x0800000 0x0800000 0x0800000 0x0800000 0x0800000 0x0800000 0x0800000 0x0800000 0x080000 0x0800000 0x0800000 0x0800000 0x0800000 0x0800000 0x0800000 0x080000 0x0800000 0x0800000 0x0800000 0x0800000 0x0800000 0x08000000 0x08000000 0x08000000 0x08000000 0x080000000 0x08000000 0x080000000 0x080000000 0x080000000 0x08000000 0x080000000 0x080000000 0x080000000 0x080000000 0x0800000000	Start Address of 0x08E00000 0x08C00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08B000000 0x08A00000 0x08B000000 0x08000000 0x08000000 0x08000000	Density of each Main Memory Re 256MB 256MB 256MB 256MB 256MB	32MB 32MB 32MB 32MB 32MB 32MB 32MB 32MB	ECC Protection Configuration for eac PROTECTED PROTECTED PROTECTED PROTECTED PROTECTED PROTECTED	INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE ACCESSIBLE ACCESSIBLE
ECC parity region ECC Parity Regio ECC Parity Regio Always user acce Main Memory Re Region 6 Region 7 Region 2 Region 1	space n Section for Main Memory Region n 0 Selection n 1 Selection n 3 Selection n 5 Selection ssible gion Space egion Start Address of each Main N 0x0A000000 0x0000000 0x0000000 0x0000000 0x000000	Start Address of 0x08E00000 0x08C00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08B000000 0x08A00000 0x08B000000 0x08000000 0x08000000 0x08000000	Density of each Main Memory Re 256MB 256MB 256MB 256MB 256MB 256MB	32MB 32MB 32MB 32MB 32MB 32MB 32MB 32MB	ECC Protection Configuration for eac PROTECTED PROTECTED PROTECTED PROTECTED PROTECTED PROTECTED PROTECTED	INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE ACCESSIBLE
ECC parity region ECC Parity Regio ECC Parity Regio Always user acce Main Memory Re Region 6 Region 7 Region 2 Region 1	space n Section for Main Memory Region n 0 Selection n 1 Selection n 3 Selection n 5 Selection ssible gion Space egion Start Address of each Main N 0x0A000000 0x0000000 0x0000000 0x0000000 0x000000	Start Address of 0x08E00000 0x08C00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08B000000 0x08A00000 0x08B000000 0x08000000 0x08000000 0x08000000	Density of each Main Memory Re 256MB 256MB 256MB 256MB 256MB 256MB	32MB 32MB 32MB 32MB 32MB 32MB 32MB 32MB	ECC Protection Configuration for eac PROTECTED PROTECTED PROTECTED PROTECTED PROTECTED PROTECTED PROTECTED	INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE ACCESSIBLE
ECC parity region ECC Parity Regio ECC Parity Regio Always user acce Main Memory Re Region 6 Region 7 Region 2 Region 1	space n Section for Main Memory Region n 0 Selection n 1 Selection n 3 Selection n 5 Selection ssible gion Space egion Start Address of each Main N 0x0A000000 0x0000000 0x0000000 0x0000000 0x000000	Start Address of 0x08E00000 0x08C00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08A00000 0x08B000000 0x08A00000 0x08B000000 0x08000000 0x08000000 0x08000000	Density of each Main Memory Re 256MB 256MB 256MB 256MB 256MB 256MB	32MB 32MB 32MB 32MB 32MB 32MB 32MB 32MB	ECC Protection Configuration for eac PROTECTED PROTECTED PROTECTED PROTECTED PROTECTED PROTECTED PROTECTED	INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE INACCESSIBLE ACCESSIBLE

1. The firmware version is the one officially supported by the BSP, but the **DDR tool** offers the possibility to

select between multiple versions.

Phy Firmware Version

FW2020.06

Figure 89. Firmware version selection UI

Note: Use only the Firmware version for your specific SoC and BSP GA version. Not all Firmware versions are supported for each SOC.

2. PHY log level selection

¥

NXP Semiconductors

IMXUG

User Guide for Config Tools for i.MX

Ph	iy Log L	.evel	Fi	rmware complete	2	~	
Fic	oure 90.	PHY log level se	lec	tion			
		K configuration					
~	ΙΟΜU	K config +	×				
	#	Command	A	Address	Size	Value	
	0	memory set	C	x30330214	32	0x00000010	
Fic	ure 91.	IOMUX configur	atio	on Ul			
	-	configuration sec					
		~ PN		config			
		PMI	C co	nfig options PMIC i	nitializa	initialization enabled (for NXP boards) ation disabled initialization enabled	
	-	PMIC configurat	ior	n UI			
		n configuration					_
	Lustom	config					7
	 Custo 	om config +	×	:			
	#	Command		Address	Size	Value ^	
	0	memory set	^	0x30300000	32	0x0000000	
	1	memory set		x30300000	32	0x00000000 🗸	
		memory setbit					
		memory clrbit		-			
		memory chkbit1					
		1	417	ł			
Fig	gure 93.	Custom configu	rat	ion UI			

Note: Any write to an incorrect address may cause unexpected behavior.

6. DQ ODT and DS configuration

User Guide for Config Tools for i.MX

✓ DQ ODT and DS cont	fig	
✓ Read config		
PHY ODT	60.0 ohm	~
DRAM driver strength	40 ohm	~
SOC ODT	60	
✓ Write config		
PHY driver strength	34.3 ohm	~
DRAM ODT	40 ohm	~

Figure 94. DQ ODT and DS configuration UI

CA ODT and DS config		
PHY driver strength (CA)	40 ohm	
DRAM ODT (CA)	40 ohm	

4.2.5 Code generation

You can generate the configuration as C code in the **Code Preview View**, which can be used by the U-Boot SPL driver.

You can trigger code generation by any change in the GUI, it is highlighted in the Code Preview.

ddr_timing.c ddr_config.ds	QQ 🗠 🗹	▼
1/*		^
2 * Copyright 2019 NXP		
3 *		
4 * SPDX-License-Identifier: GPL-2.0+		
5 *		
6 * Generated code from MX8M_DDR_tool		
7 * Align with uboot version:		
8 * imx_v2018.03_4.14.78_1.0.0_ga ~ imx_v2018.04_4.19.35_1.1.0_ga		
9 */		
10		
11 #include <linux kernel.h=""></linux>		
12 #include <asm arch="" imx8m_ddr.h=""></asm>		
13		
14 struct dram_cfg_param ddr_ddrc_cfg[] = {		
15 /** Initialize DDRC registers **/		
16 {0x3d400304, 0x1},		

You can save files from Code Preview on the disk by using DDR tool Export Wizard.

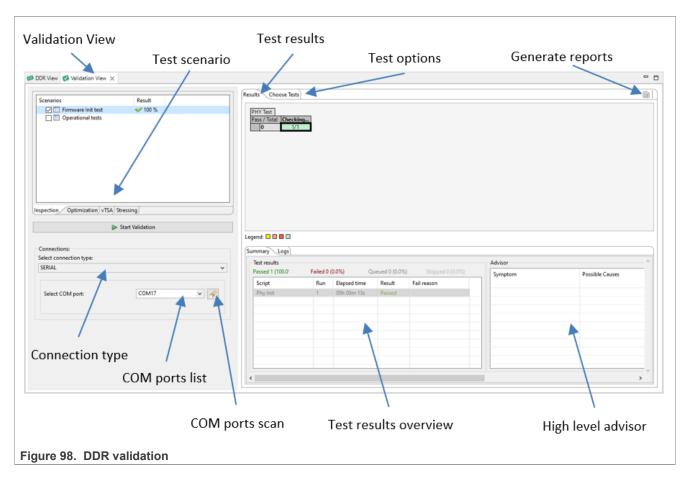
	~	Browse
	~	Browse
		browse
Finish		Cancel
	Finish	Finish

4.3 DDR validation

The DDR validation uses different scenarios to assess DDR performance, by downloading a test image to the processor's internal RAM.

DDR validation can help to assess stability of the DDR interface on the board in a non-OS environment.

User Guide for Config Tools for i.MX



4.3.1 Connection

This section describes connection to boards of various types.

4.3.1.1 Boards with Serial Download mode/Manufacture mode

To connect to a board, do the following:

- 1. Configure the board to boot in the Serial Download mode/Manufacture mode and power up the board.
- 2. Connect a UART cable from the host computer to the UART of the A-core on the board.
- 3. Connect a USB cable from the host computer to the USB port on the board that is used by the Serial Download mode. An "HID-compliant device" or a "USB Input Device" is shown in the Windows Device Manager.

With the board connected to the host computer, search the UART ports by using **COM port scan**. COM port drop list is populated with all the available UART ports.

	Connections: Select connection type:		
	SERIAL		*
	Select COM port:	COM15 🗸	9
Figure 99. COM port sel	lection		

Choose the correct UART that is used as the A-core debug UART port.

4.3.1.2 Boards with JTAG connection available

- 1. Connect the JTAG probe (only the CWTAP probe is supported) to the board.
- 2. Select between USB or Ethernet connection.

	Connections: Select connection type:			
	JTAG O Ethernet		*	
	USB device:	00:00:00:00:00:00		
Figure 100 Conn	ection type selection			

4.3.2 Test scenarios

Once the DDR configuration and board connection are set up, you can execute different **Test scenarios**. You can customize each test by setting the parameters from **Test options**.

Depending on the test and options selected, the execution time may differ. By default a 90 seconds timeout is set, to assure that in case of an issue the test finishes. To change the default value, edit the **Timeout (seconds)** option:

	Results Choose Tests
	Select which tests the scenario will run, a
	Sync selection across all scenarios
	Timeout (seconds): 90
Figure 101. Timeout selection	

When initial timeout expires, provide the input to continue or not the test execution.

🚳 Tes	t execution timeout (s)	×	
0	Test execution timeout expired. Press Yes if you want to continue the test, or No if you want to stop execution.		
	Yes	No	
igure 102. Test execution ti	meout(s)		

To start test execution, press the "**Start Validation**" button. You can check the status of the running test from the **Logs** console. By default, the log level is set to **ERROR**. Additional log-level options are available, with different output in the console:

Summary Logs	
View log for: Phy Init - Run 1 (Passed)	ERROR 🖂
######################################	DEBUG
Microsoft Windows [Version 10.0.18363.1139]	INFO
(c) 2019 Microsoft Corporation. All rights reserved.	WARN
	ERROR
Figure 103. Log level selection	

At the end of the test, the PASS/FAIL status is displayed in **"Results**". The test summary is displayed in **"Summary**".

	ass / Total	DRAM driver strength					
	ass / lotal	240 ohm	120 ohm	80 ohm	60 ohm	48 ohm	40 ohm
	240 ohm	0/1	0/1	0/1	0/1	0/1	0/1
TUC	120 ohm	0/1	1/1	1/1	1/1	1/1	1/1
	80 ohm	0/1	1/1	1/1	1/1	1/1	1/1
1	60 ohm	1/1	1/1	1/1	1/1	1/1	1/1
	40 ohm	1/1	1/1	1/1	1/1	0/1	0/1
Ea	ch cell c	olor car	be dec	oded us	ing the	legend	
04. Test status							

- Yellow is for Test failed
- Orange is for *Configuration error*
- Red is for Target connection error or exception in the script
- Green is for Test passed

If a test fails, the **Summary** view in the **Advisor** section displays **Symptom** and **Possible Causes** to provide high-level guidance on the debug process when looking for possible DDR subsystem issues.

Symptom	Possible Causes
Generic failure	1. Use the latest version of the DDR tool to generate the proper setting for the customer's configuration
	2. Ensure the bit swapping or byte swapping properly follow the limitation that may apply and the DDR tool has been provide the correct
	3. Review and verify schematics is correct
	4. Verify if the used memory device is compatible with the SoC
	5. Verify the proper input frequency and correct PLL configuration are applied
	6. Verify the DDR layout guideline document from NXP has been followed
	7. Verify the voltages on the board
	8. Verify the board has been properly reset and the power up sequence
	9. Check HW for manufacturing issues

The DDR tool offers several test scenarios that can be split into Inspection, Optimization, vTSA, and Stressing.

4.3.2.1 Inspection

Inspection shows the status of the DDR Controller and DDR PHY configuration, by executing following tests:

- 1. *Firmware Init* executes the DDR PHY training to check the DDR PHY configuration.
- Operational performs basic memory access test by running Write-Read-Compare/ Walking Ones/ Walking Zeros tests. Such options as Start Address, Size, Enable DDR Memory cache, Access mode/ Pattern option are available for each test.

Start addres	s 0x4000000	hex
Size	32MB	~
✓ Enable D	DR Memory cache	
Use rand	om pattern	
Pattern		
0xAABBCCI 0xAABBCCI	DD,0x01234567,0xFFFFFFF,0xAAAAAAAA, DD,0x01234567,0xFFFFFFF,0xAAAAAAAA, DD,0x01234567,0xFFFFFFF,0xAAAAAAAA, DD,0x01234567,0xFFFFFFF,0xAAAAAAAAA	~
		~

3. *ECC Regions test* – tests each ECC region with 1-bit error injection to verify the region's ECC capability (protected or unprotected).

Note: The text is possible only for devices with inline ECC support.

4.3.2.2 Optimization

DQ ODT and driver strength tests sweep the DQ IO configurations to create board-specific Driver Strength vs. ODT PASS/FAIL map for the Reads and the Writes.

Note: Optimization is not available when UI configuration is bypassed by RPA initialization script import.

Read ODT and driver									
Pass / Total			DRAM driver strength						
Fa	ss/ iotai	240 ohm	120 ohm	80 ohm	60 ohm	48 ohm	40 ohm		
	240 ohm	0/1	0/1	0/1	0/1	0/1	0/1		
늄	120 ohm	0/1	1/1	1/1	1/1	1/1	1/1		
2	80 ohm	0/1	1/1	1/1	1/1	1/1	1/1		
Η	60 ohm	1/1	1/1	1/1	1/1	1/1	1/1		
	40 ohm	1/1	1/1	1/1	1/1	1/1	1/1		

Figure 107. DQ ODT and DS map

For passing cells (Green cells), the option *Apply current selection in DDR configuration* (right-click on the cell) is enabled. It sets the respective Driver Strength and ODT value into the configuration for use in other scenarios.

Rea	d ODT and	driver						
Da	cc / Total		D					
Pass / Total		240 ohm	120 ohm	80 ohm	60 ohm	48 ohm	40 ohm	
	240 ohm	0/1	0/1	0/1	0/1	0/1	0/1	
늄	120 ohm	0/1	1/1	1/1	1/1	1/1	1/1	
9	80 ohm	0/1	1/1	Apply	current sel	ection in D	DR configu	uration
Ĩ	60 ohm	1/1	1/1	1/1	1/1	1/1	1/1	
	40 ohm	1/1	1/1	1/1	1/1	1/1	1/1	

Figure 108. Apply DQ ODT and DS configuration

Note: You can use the Driver Strength vs. ODT map as one of the criteria when deriving optimal ODT/Driver Strength values. This map cannot serve as all-comprising output to make this determination.

Note: NXP strongly recommends using the default ODT and Drive strength values that are tested and validated as part of our GA BSP. To ensure that your design adheres to the board layout requirements, refer to the device *i*.MX 8M Hardware Developer's Guide.

Vref for 1D optimization test sweeps the PHY Vref and/or DRAM Vref to determine the values for the PHY training to pass in case PHY training fails and to determine the trained values after the 2D training.

When the test passes, *Apply current selection in DDR configuration* option is enabled. It sets the respective PHY Vref and DRAM Vref values into the configuration.

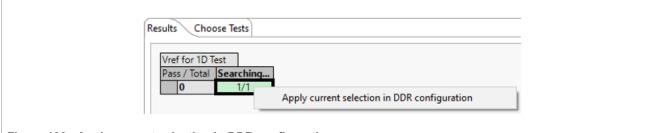


Figure 109. Apply current selection in DDR configuration

Vref for the CA optimization test executes only 1D training and reads the VrefCA after the training is complete.

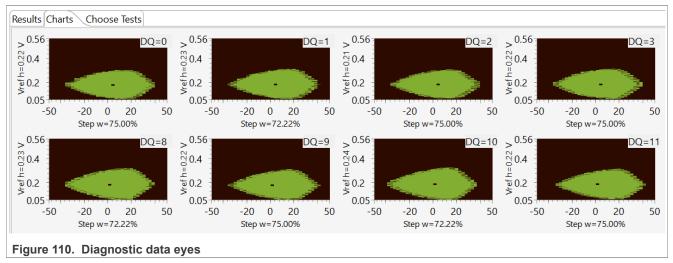
When the test passes, the **Apply the current selection in the DDR configuration** option, which sets the VrefCA value into the configuration, is enabled.

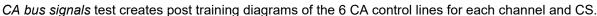
IMXUG

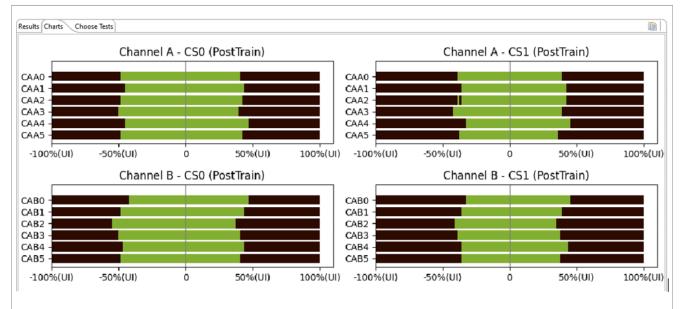
© 2023 NXP B.V. All rights reserved.

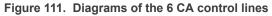
4.3.2.3 vTSA

vTSA performs Virtual Timing Signal Analysis by running tests to determine margins of DDR subsystem. Diag Write Margin/ Diag Read Margin tests creates virtual data eye diagram for each DQ lanes.

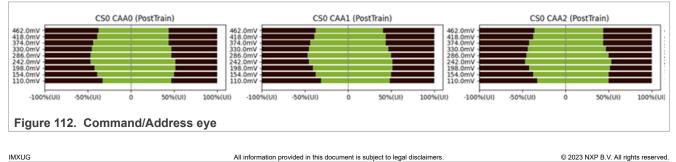








CA eye test creates post training eyes of the 6 CA control lines for each channel and CS.



Note: Details about vTSA are provided in the FAQ section.

4.3.2.4 Stressing

To test the stability of the DDR configuration more extensively, you can use the *Stressing* scenario, with its suite of tests that covers different situations.

Two ways of running *Stress* tests are available:

 Single run runs the test suite one time with different options selected (Size, Enable DDR Memory cache, Stop on fail). In case of failure, you can check the status of each test in the suite in the Logs console, with Log level set to INFO

lesults	Choose Tests	D
Stress Pass / 0	Total Checking	
View lo	og for: Stress tests - Run 1 (Failed) VINFO	~
	test_app num stress test for inequency pointow roominiz	~
INFO	test app [INFO]: Start Stress tests	
INFO	test_app [INFO]: Start Stress tests test_app [INFO]: Data is address test fail with 2097152 fails test_app [INFO]: Row hop read test fail with 4194287 fails	
INFO INFO	test_app [INFO]: Data is address test fail with 2097152 fails	
INFO INFO INFO	test_app [INFO]: Data is address test fail with 2097152 fails test_app [INFO]: Row hop read test fail with 4194287 fails	
INFO INFO INFO INFO	test_app [INFO]: Data is address test fail with 2097152 fails test_app [INFO]: Row hop read test fail with 4194287 fails test_app [INFO]: SSN memcpy x32 test1 fail with 1629438 fails test_app [INFO]: SSN memcpy x32 test2 fail with 1686524 fails test_app [INFO]: SSN memcpy x32 test3 fail with 480248 fails	
INFO INFO INFO INFO INFO	test_app [INFO]: Data is address test fail with 2097152 fails test_app [INFO]: Row hop read test fail with 4194287 fails test_app [INFO]: SSN memcpy x32 test1 fail with 1629438 fails test_app [INFO]: SSN memcpy x32 test2 fail with 1686524 fails	

Figure 113. Stress tests

2. Test duration runs the test suite for a selected time. This is suitable for overnight tests.

Stress tests para	s tests parameters									
Source address	0x4000000 hex									
Size	32MB ~									
✓ Enable DDR✓ Stop on fail	Memory cache									
Test duration	✓ 1 → Hours 0 → Minutes									
Figure 114. Stress	s tests options									

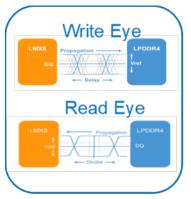
In the Logs console, you can monitor test execution and see the number of iterations and the duration.

Results Choose Tests Stress Tests Pass / Total O Legend:	(
Summary Logs		
View log for: Stress tests - Run 1 (Passed)	✓ ERROR ✓	
Os: PASSED (1/1)	- · · ·	
1s: PASSED (2/2)		
2s: PASSED (3/3)		
3s: PASSED (4/4)		
4s: PASSED (5/5)		
5s: PASSED (6/6)		
Figure 115. Stress tests results		

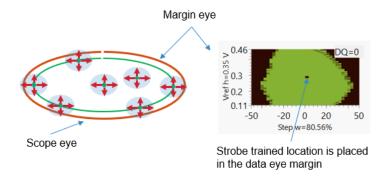
Note: Make sure the **Timeout (seconds)** setting is higher than the **Test duration** setting, otherwise the test ends with timeout.

4.4 FAQ

- 1. What does vTSA mean?
 - a. vTSA is an abbreviation for Virtual Timing Signal Analysis.
 - b. A "virtual" TSA uses the memory controller itself to test margins without test equipment. "Virtual" does not mean simulation!
 - c. Memory controllers have the ability the alter timings, voltage references, termination settings, and so on, for both incoming and outgoing signals.
 - d. "Training" is a process when the memory controller sweeps these parameters and finds the configuration with the most margin for operation.
 - e. A vTSA simply logs this information for output, which provides insight into the signaling margin of the system without the need for test equipment.
 - f. Initialization and calibration settings can be dumped to a file for analysis as well.
- 2. What is the vTSA output?
 - a. Virtual Timing Signal Analysis(vTSA) provides write and read data eye diagrams virtually by running a series of write/read transactions as opposed to the hardware method of using a high-speed oscilloscope to perform manual physical TSA (pTSA) measurements.
 - b. This being the case, vTSA output itself approximates the actual write/read eyes.
 - c. You should expect some variation between trained values of delay lines and VREF in comparison to the vTSA report of these values.
 - d. vTSA only reports the values it detects in the "widest" part of the reported eye, which may itself vary from run-to-run.

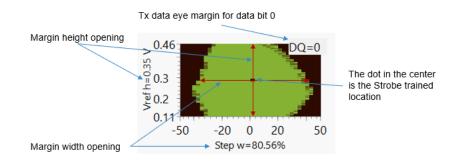


- e.
- 3. Need more information about vTSA?
 - a. The vTSA tool is an approximation of an actual pTSA. Thus, you may note some variation between the trained delay line value and the vTSA "mid" value. It is observed and expected that this variation may be up to ~20 ps. The key takeaway from the generated eye diagrams should be focused on verifying the ample margin of the trained delay line value within the data eye.
 - b. For the trained VREF value, the LPDDR4 device Mode Register 14 (MR14 which holds the trained VREF value) applies to ALL byte lanes. In other words per JEDEC, there is not an MR14 per byte lane and instead, MR14 applies to all byte lanes.
 - c. For a board that follows the NXP DDR layout guidelines, there should be plenty of margin around the VREF trained value. You can find the guidelines in the respective NXP Hardware Developer Guide.
- 4. How a data eye margin is generated?
 - a. There are several delay steps available to shift each DQ and DQS.
 - b. As DQ crosses the unit interval, from zero-step delay to 1 unit internal step delay, each step is tested with a write-read-compare test to determine pass or fail.
 - c. The DQ traverse of the unit interval is repeated for all available VREF steps.
 - d. Delay steps generate a line, and repeating the lines at each VREF step generates data eye margin.
 - e. The crossing of DQS signal with trained VREF and delay step is placed in the generated data eye margin.
 - f. Each passing dot in the margin eye already meets the setup, hold, and voltage requirement.



- 5. What represents the information next to the data eye?
 - a. Unit interval = 1/data rate; for example, at 3200MT/s data rate the unit interval = 312.5 ps
 - b. The x-axis displays the time. It is one unit interval in percentage. -50 % to +50 %
 - c. The x-axis data eye margin width opening is displayed as the percentage of one unit interval. For example: Step w=80.56 % of UI
 - d. The y-axis displays the voltage.

User Guide for Config Tools for i.MX



- e.
- 6. How much margin is considered as good?
 - a. To determine the required margin mask, you must do the following:
 - Optimize The DDR interface
 - Settings have been optimized, generate the worst-case data eye margin using the worst-case conditions (temperature, voltage, frequency, pattern) for a customer board DDR interface.
 - Use the DDR training optimizing/centering the strobe to the eye margin
 - b. A green pixel in the data eye margin indicates a passing cell. It means for that green pixel the setup and hold time as well as the VIHLac/dc are satisfactory.
 - c. Any additional green pixels around the strobe location in the data eye margin are additional margin available to DDR for that DQ.
- 7. Why is the trained Vref sometimes not in the exact center of the eye?
 - a. The VREF training must select a value that corresponds to all of the byte lanes' passing VREF window and then program this value into the LPDDR4 MR14 register. It means that for a one-byte lane, though the trained VREF value may not seem to be in the exact center of the data eye, it is selected to provide the best possible margin for this byte lane along with satisfying the other byte lanes.
- 8. How to check that wrong UART is selected?
- Set the Log Level to DEBUG and check the messages from console

File "C:\nxp\i.MX_CFG_v9\bin\python38\serial\serialwin32.py", line 62, in open raise SerialException("could not open port {!r}: {!r}".format(self.portstr, ctypes.WinError())) serial.serialutil.SerialException: could not open port 'COM4': PermissionError(13, 'Access is denied.', None, 5)



9. How to proceed in case of test timeout?

	🙆 Validati	ion Error		\times
		Connection timed-out!		
а.			ОК	
	Tracaback (may	st recent call last):		
b.	File "C:\Progra	amData\NXP\mcu_data_v9\processors\MIMX8MM4xxxKZ\ksdk2_0\mem_validation\ddrc\scripts\cor vaiting_for_input()	nmon\base_test.py",	line 224,
-				

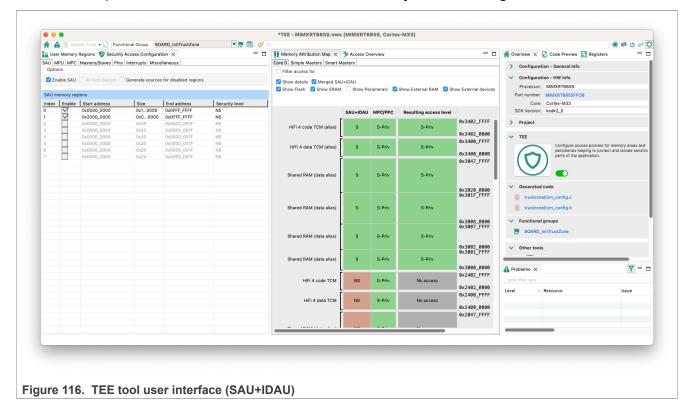
5 Trusted Execution Environment Tool

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

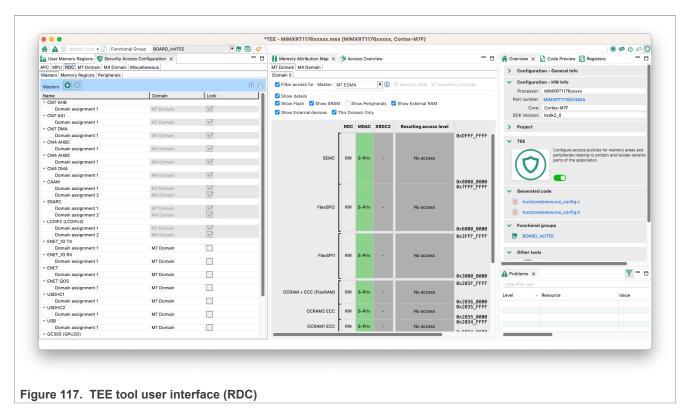
You can set security policies of different parts of your application in the **Security Access Configuration** and its subviews, and review these policies in the **Memory Attribution Map**, **Access Overview** and **Domains Overview** views. Use the **User Memory Regions** view to create a convenient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

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

Note: In order 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.



User Guide for Config Tools for i.MX



5.1 AHBSC with security extension-enabled devices

The features and appearance of the TEE tool are based on the security model of the loaded device.

This section describes the features and appearance of the tool for devices with security extensions TrustZone-M with AHBSC.

Currently, following devices of this type are supported:

- LPC55Sxx
 - LPC55S69, LPC55S66
 - LPC55S16, LPC55S14
- LPC55S06, LPC55S04
- RT6xx, RT5xx
 - MIMXRT685S, MIMXRT633S
 - MIMXRT595S, MIMXRT555S, MIMXRT533S1q
 Note: Pre-production only.

5.1.1 User Memory Regions view

In the **User Memory Regions** view, you can create and maintain a high-level configuration of memory regions and their security levels. You can create the regions, name them, specify their address, size, security level, and provide them with a description. You can then fix any errors in the settings with the help of the **Problems** view.

Create a new memory region by clicking the Add new memory region button in the view's header.

Enter/change the memory region's parameters by clicking the row's cells. In the **Security Level** column, you have these options to choose from:

• NS-User - Non-secure user

- NS-Priv Non-secure privileged
- S-User Secure user
- S-Priv Secure privileged
- NSC-User Non-secure callable user
- NSC-Priv Non-secure callable privileged
- Any

Errors in configuration are highlighted by a red icon in the relevant cell. In the case the issue is easily fixed, you can right-click the cell to display a dropdown list of offered solutions.

Remove the memory region by selecting the table row and clicking the **Remove selected memory region(s)** button in the view's header.

User memor	ry regions 🕒 😒					
ID	Start address	Size	End address	Name	Security level	Description
Region_1	0x0000_0000	0x1	0x0000_0000	Region_1	NS-Priv	A
					Any NS-User NS-Priv S-User S-Priv NSC-User NSC-Priv	

5.1.2 Security Access Configuration view

In the **Security Access Configuration** view, you can configure your application's security policies in a number of ways. See the following sections for more details.

5.1.2.1 SAU

In the **SAU** subview, you can enable and configure SAU (Security attribution unit).

When enabled, you can set up SAU memory regions, specify their start and size or end address, and specify their access level. SAU automatically sets the entire memory space to a Secure access level when disabled. When enabled, SAU deems every uncovered (that is, unconfigured) memory region as Secure, so only NS or NSC can be selected for a covered (configured) memory region.

You can choose between two access levels:

- NS Non-secure
- NSC Non-secure callable

Alternatively, you can set all the SAU memory regions to non-secure access level by selecting the **All Non-Secure**.

Note: This option is only available when SAU is disabled.

You can also decide to generate code even for disabled memory regions by selecting the option **Generate sources for disabled regions**.

SAU Se	cure MPU	Regions 🦃 Security Act		ins Interrupts ^{&} Miscellan		
Option	s ole SAU	All Non-Secure Ger	nerate sources for disabl	ed regions		
				5		
SAU Me	mory Reg	jions				
Index	Enable	Start Address	Size	End Address	Security Level	
0		0x0000_0000	0x20	0x0000_001F	NS	
1	\checkmark	0x0000_0000	0x20	0x0000_001F	NS	
2		0x0000_0000	0x20	0x0000_001F	NS	
3		0x0000_0000	0x20	0x0000_001F	NS	
4		0x0000_0000	0x20	0x0000_001F	NS	
5		0x0000_0000	0x20	0x0000_001F	NS	
6		0x0000_0000	0x20	0x0000_001F	NS	
7		0x0000_0000	0x20	0x0000_001F	NS	
	_					
Figure	119. SA					

5.1.2.2 Interrupts

In the **Interrupts** subview, you can set security designation for device's peripheral interrupts. In case if the processor contains more than a single core or processing unit, additional **Handling by Core** tables might appear. In these tables, you can specify if the interrupts coming from the peripheral can be handled by the core or processing unit.

All interrupts are set to **Secure** by default. If you want to change the interrupt source's security designation, left-click the **Secure** cell of the interrupt and choose from the dropdown menu. Alternatively, right-click the interrupt's **Name** cell and choose the security designation from the context menu. To select multiple entries, use the **Ctrl+Left-click** shortcut, then right-click the selected area for the context menu. Alternatively, you can use **Shift+Up/Down** after selecting the row to expand the selection.

NXP Semiconductors

IMXUG

User Guide for Config Tools for i.MX

U Secure MPU Non-secure MPU ^(a) MPC ^(a) Masters/Slaves	Pins Interrupts ^a Miscellaneous
terrupt security configuration	
lame	Interrupt security configuration
ADC	Secure
Analog comparator Sub-system	Secure
CAN0 interrupt 0	Secure
CAN0 interrupt 1	Secure
CASPER	Secure
Code watchdog timer interrupt	Secure
Flexcomm interface 0	Secure
Flexcomm interface 1	Secure
Flexcomm interface 2	Secure
Flexcomm interface 3	Secure
Flexcomm interface 4	Secure
Flexcomm interface 5	Secure
Flexcomm interface 6	Secure
Flexcomm interface 7	Secure
Flexcomm interface 8	Secure
GPIO group 0	Secure
GPIO group 1	Secure
HASH-AES	Secure
Micro Tick timer	Secure
Multi-Rate timer	Secure
OS event timer and OS event wake up	Secure
Pin interrupt 0 or pattern match engine slice 0	Secure
Pin interrupt 1 or pattern match engine slice 1	Secure
Pin interrupt 2 or pattern match engine slice 2	Secure
Pin interrupt 3 or pattern match engine slice 3	Secure

5.1.2.3 Secure/Non-secure MPU

In the **Secure MPU** and **Non-secure MPU** sub-views, you can enable and configure MPU (Memory Protection Unit). You can create regions, specify their address, size, and other parameters. Use the **Secure MPU** sub-view for the configuration of the secure, and **Non-secure MPU** for the configuration of the non-secure security level.

User Guide for Config Tools for i.MX

Secure N	APU Non-se	ecure MPU							
Optio	ns								
		Enable MPU dur	ing HardF	ault and NM	11 ha	ndlin	1		
		d software access	-						
		s for disabled reg			Ĩ.,				
		-							
Secure	MPU memo	ry attributes							
		1	1		Inn	er attr	butes		0
Index	ID	Memory type	Device	- F	С		R W	Т	с
0	Code	Normal	n/a		$\overline{\Box}$		1ĕ n/	_	
1	RAM	Normal	n/a		H		1a n/		
2	Peripheral		nGnRE		n a		1a n/		_
3	3	Device	nGnRE				1a n/		
4	4	Device	nGnRE				1a n/		
5	5	Device	nGnRE				1a n/		
6	6	Device	nGnRE				1a n/		
7	7	Device	nGnRE				1a n/		
· ·		Device	HOHINE						
<									>
									>
	MPU memo	ry regions							>
		ny regions Start address	Size	End addre	ss	Exec	Per	nissic	
Secure	Enable		Size 0x20	End addres	_	Exec	Perr		
Secure Index	Enable	Start address			IF	Exec	_	priv	
Secure Index 0	Enable	Start address	0x20	0x0000_001	IF IF	Exec	RW	priv priv	
Secure Index 0 1	Enable	Start address 0x0000_0000 0x0000_0000	0x20 0x20	0x0000_001 0x0000_001	IF IF	Exec	RW RW	priv priv priv	
Secure Index 0 1 2	Enable (Start address 0x 0000_0000 0x 0000_0000 0x 0000_0000 0x 0000_0000	0x20 0x20 0x20	0x0000_001 0x0000_001 0x0000_001	IF IF IF	Exec	RW RW RW	priv priv priv priv	
Secure Index 0 1 2 3	Enable (Start address 0x0000_0000 0x0000_0000 0x0000_0000 0x0000_0000 0x0000_0000	0x20 0x20 0x20 0x20 0x20	0x0000_001 0x0000_001 0x0000_001 0x0000_001	IF IF IF IF	Exec	RW RW RW RW	priv priv priv priv priv	
Secure Index 0 1 2 3 4	Enable (Start address 0x0000_0000 0x0000_0000 0x0000_0000 0x0000_0000 0x0000_0000 0x0000_0000 0x0000_0000	0x20 0x20 0x20 0x20 0x20 0x20	0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001	IF IF IF IF	Exec	RW RW RW RW RW	priv priv priv priv priv priv	
Secure Index 0 1 2 3 4 5	Enable	Start address 0x0000_0000 0x0000_0000 0x0000_0000 0x0000_0000 0x0000_0000 0x0000_0000 0x0000_0000 0x0000_0000 0x0000_0000	0x20 0x20 0x20 0x20 0x20 0x20 0x20 0x20	0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001	IF IF IF IF IF		RW RW RW RW RW RW	priv priv priv priv priv priv priv	
Secure Index 0 1 2 3 4 5 6	Enable	Start address 0x0000_0000 0x0000_0000	0x20 0x20 0x20 0x20 0x20 0x20 0x20 0x20	0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001	IF IF IF IF IF		RW RW RW RW RW RW RW	priv priv priv priv priv priv priv	
Secure Index 0 1 2 3 4 5 6	Enable	Start address 0x0000_0000 0x0000_0000	0x20 0x20 0x20 0x20 0x20 0x20 0x20 0x20	0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001	IF IF IF IF IF		RW RW RW RW RW RW RW	priv priv priv priv priv priv priv	
Secure Index 0 1 2 3 4 5 6	Enable	Start address 0x0000_0000 0x0000_0000	0x20 0x20 0x20 0x20 0x20 0x20 0x20 0x20	0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001	IF IF IF IF IF		RW RW RW RW RW RW RW	priv priv priv priv priv priv priv	
Secure Index 0 1 2 3 4 5 6	Enable	Start address 0x0000_0000 0x0000_0000	0x20 0x20 0x20 0x20 0x20 0x20 0x20 0x20	0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001	IF IF IF IF IF		RW RW RW RW RW RW RW	priv priv priv priv priv priv priv	
Secure Index 0 1 2 3 4 5 6	Enable	Start address 0x0000_0000 0x0000_0000	0x20 0x20 0x20 0x20 0x20 0x20 0x20 0x20	0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001	IF IF IF IF IF		RW RW RW RW RW RW RW	priv priv priv priv priv priv priv	
Secure Index 0 1 2 3 4 5 6	Enable	Start address 0x0000_0000 0x0000_0000	0x20 0x20 0x20 0x20 0x20 0x20 0x20 0x20	0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001	IF IF IF IF IF		RW RW RW RW RW RW RW	priv priv priv priv priv priv priv	
Secure Index 0 1 2 3 4 5 6	Enable	Start address 0x0000_0000 0x0000_0000	0x20 0x20 0x20 0x20 0x20 0x20 0x20 0x20	0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001	IF IF IF IF IF		RW RW RW RW RW RW RW	priv priv priv priv priv priv priv	
Secure Index 0 1 2 3 4 5 6	Enable	Start address 0x0000_0000 0x0000_0000	0x20 0x20 0x20 0x20 0x20 0x20 0x20 0x20	0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001 0x0000_001	IF IF IF IF IF		RW RW RW RW RW RW RW	priv priv priv priv priv priv priv	

Figure 121. MPU

MPU is disabled by default and must be enabled by selecting the Enable MPU option.

Note: Not every device supports MPU.

Use the **MPU Memory Attributes** table to name and configure MPU memory attribute sets. Click the cells of the **Memory Type** and **Device Attributes** columns to display the available choices.

Use the **MPU Memory Regions** table to enable and configure MPU memory regions.

- 1. Enable the region.
- 2. Specify the Address.
- 3. Specify either the Size or the End Address.

- 4. Set the **Exec** option if you want the region to be able to run code.
- 5. Set the **Permissions** (Read Only or Read/Write).
- 6. Set the **Privileges**. **Note:** Privileged access can be set by default for all memory regions not handled by MPU by selecting the **Enable privileged software access to the default memory map** option.
- 7. Set the **Shareability**, or the caching options.
- 8. Allocate one of the sets from the **MPU Memory Attributes** table in **Mem.Attr.**. Sets can be allocated to more than one region.

5.1.2.4 MPC

In the **MPC** (Memory Protection Checker) subview, you can set security policies on entire memory sectors as defined by physical addresses.

Set the memory sector security level by left-clicking the relevant cell in the **Security level** column and choosing from the dropdown list. Alternatively, you can right-click the relevant cell in the **Sector** column and choose the security level from the context menu. To select multiple entries, use the **Ctrl+Left-click** shortcut, then right-click the selected area for the context menu.

You have four security levels to choose from, in ascending order of security:

- NS-User Non-secure user
- NS-Priv Non-secure privileged
- S-User Secure user
- S-Priv Secure privileged

User Memory Regions 🦻 Security Access Configuration $ imes$		
AU Secure MPU Non-secure MPU 🌢 MPC 🌢 Masters/Slaves Pins Interrupts 🜢 Misce	llaneous	
IPC Sectors		
ector	Security Level	
FLASH: 0x0000_0000 - 0x0001_FFFF (0x1000_0000 - 0x1001_FFFF)		
0x0000_0000 - 0x0000_7FFF (0x1000_0000 - 0x1000_7FFF)	NS-User	
0x0000_8000 - 0x0000_FFFF (0x1000_8000 - 0x1000_FFFF)	NS-User	
0x0001_0000 - 0x0001_7FFF (0x1001_0000 - 0x1001_7FFF)	NS-User	
0x0001_8000 - 0x0001_FFFF (0x1001_8000 - 0x1001_FFFF)	NS-User	
Boot ROM: 0x0300_0000 - 0x0301_FFFF (0x1300_0000 - 0x1301_FFFF)		
0x0300_0000 - 0x0300_0FFF (0x1300_0000 - 0x1300_0FFF)	NS-User	
0x0300_1000 - 0x0300_1FFF (0x1300_1000 - 0x1300_1FFF)	NS-User	
0x0300_2000 - 0x0300_2FFF (0x1300_2000 - 0x1300_2FFF)	NS-User	
0x0300_3000 - 0x0300_3FFF (0x1300_3000 - 0x1300_3FFF)	NS-User	
0x0300_4000 - 0x0300_4FFF (0x1300_4000 - 0x1300_4FFF)	NS-User	
0x0300_5000 - 0x0300_5FFF (0x1300_5000 - 0x1300_5FFF)	NS-User	
0x0300_6000 - 0x0300_6FFF (0x1300_6000 - 0x1300_6FFF)	NS-User	
0x0300_7000 - 0x0300_7FFF (0x1300_7000 - 0x1300_7FFF)	NS-User	
0x0300_8000 - 0x0300_8FFF (0x1300_8000 - 0x1300_8FFF)	NS-User	
0x0300_9000 - 0x0300_9FFF (0x1300_9000 - 0x1300_9FFF)	NS-User	
0x0300_A000 - 0x0300_AFFF (0x1300_A000 - 0x1300_AFFF)	NS-User	
0x0300_B000 - 0x0300_BFFF (0x1300_B000 - 0x1300_BFFF)	NS-User	
0x0300_C000 - 0x0300_CFFF (0x1300_C000 - 0x1300_CFFF)	NS-User	
0x0300_D000 - 0x0300_DFFF (0x1300_D000 - 0x1300_DFFF)	NS-User	
0x0300_E000 - 0x0300_EFFF (0x1300_E000 - 0x1300_EFFF)	NS-User	
0x0300_F000 - 0x0300_FFFF (0x1300_F000 - 0x1300_FFFF)	NS-User	
0x0301_0000 - 0x0301_0FFF (0x1301_0000 - 0x1301_0FFF)	NS-User	
0x0301_1000 - 0x0301_1FFF (0x1301_1000 - 0x1301_1FFF)	NS-User	
0x0301_2000 - 0x0301_2FFF (0x1301_2000 - 0x1301_2FFF)	NS-User	

5.1.2.5 Masters/Slaves

In the Masters/Slaves subview, you can configure security levels for bus masters and slaves.

Set the bus master/slave security level by left-clicking the relevant cell in the **Security level** column and choosing from the dropdown list. Alternatively, you can right-click the relevant cell in the **Master** and **Slave** column and choose from the security level from the context menu. To select multiple entries, use the **Ctrl+Left-click** shortcut, then right-click the selected area for the context menu.

You have four security levels to choose from, in ascending order of security:

- NS-User Non-secure user
- NS-Priv Non-secure privileged
- S-User Secure user
- S-Priv Secure privileged

You can further specify the interrelation between master and slave security levels by selecting the following options:

- Simple Master in Strict Mode Select to allow simple bus master to read and write on same level only. Deselect to allow to read and write on same and lower level.
- Smart Master in Strict Mode Select to allow smart bus master to execute, read, and write to memory at same level only. De-select to allow to execute on same level only, read and write on same and lower level.

Note: Instruction-type bus master security level must be equal to bus slave security level. Data and others security level must be equal or higher than bus slave security level.

	ions 🧐 Security Access Config	ers/Slaves Pins Interrupts ^{&} Miscellaneous	
Options	Strict Mode Smart Master in S		
Master	Security Level	Slave	Security Lev
Simple master		ADC0	NS-User
CANFD	NS-User	AHB_SECURE_CTRL	NS-User
DMA0	NS-User	ANACTRL	NS-User
DMA1	NS-User	CANØ	NS-User
HASHCRYPT	NS-User	CASPER	NS-User
USBFSD	NS-User	CRC_ENGINE	NS-User
USBFSH	NS-User	CTIMERØ	NS-User
		CTIMER1	NS-User
		CTIMER2	NS-User
		CTIMER3	NS-User
		CTIMER4	NS-User
		DBGMAILBOX	NS-User
		DMAØ	NS-User
		DMA1	NS-User
		FLASH	NS-User
		FLEXCOMMØ	NS-User
		FLEXCOMM1	NS-User
		FLEXCOMM2	NS-User
		FLEXCOMM3	NS-User
		FLEXCOMM4	NS-User
		FLEXCOMM5	NS-User
		FLEXCOMM6	NS-User
		FLEXCOMM7	NS-User
		GINTØ	NS-User

5.1.2.6 Pins

In the **Pins** subview, you can specify if the reading GPIO state is allowed or denied.

All pins' reading GPIO state is set to **Allow** by default. If you want to change the pins reading GPIO state, leftclick the **Reading GPIO state** cell of the pin and choose from the dropdown menu. Alternatively, right-click the pin's **Name** cell and choose the reading GPIO state from the context menu. To select multiple entries, use the **Ctrl+Left-click** shortcut, then right-click the selected area for the context menu. Alternatively, you can use **Shift** +**Up/Down** after selecting the row to expand the selection.

NXP Semiconductors

IMXUG

User Guide for Config Tools for i.MX

User Memory Regions 🧐 Security Access (AU Secure MPU Non-secure MPU 🌢 MPC 🌢	Masters/Slaves Pins Interrupts Miscellaneous	
eading GPIO state		Ē.
Name	Reading GPIO state	
General purpose input/output port 0		
Pin 0 – [54] PIO0_0/FC3_SCGPI0/SECUR	E_GPIO0_0/ACMP0 Allow	
Pin 1 – [7] PIO0_1/FC3_CTSGPI1/CMP0	_OUT/SECURE_GPI Allow	
Pin 2 - [81] PIO0_2/FC3_TXDUT0/SCT_G	PI2/SECURE_GPIOC Allow	
Pin 3 - [83] PIO0_3/FC3_RXDUT1/SCT_G	SPI3/SECURE_GPIO(Allow	
Pin 4 [86] PIO0_4/CAN0_RTS_SDA_SSE	EL0/SECURE_GPIO0. Allow	
Pin 5 - [88] PIO0_5/CAN0_TDL_SSEL1/N	ACLK/SECURE_GPIC Allow	
Pin 6 - [89] PIO0_6/FC3_SCAT0/SCT_GP	PI6/SECURE_GPIO0_ Allow	
Pin 7 - [6] PIO0_7/FC3_RTSSCK/FC1_S	SCK/SECURE_GPIOC Allow	
Pin 8 - [26] PIO0_8/FC3_SSSI_DATA/SW	O/SECURE_GPIO0_ Allow	
Pin 9 - [55] PIO0_9/FC3_SSWS/SECURE	E_GPIO0_9/ACMP0_ Allow	
Pin 10 - [21] PIO0_10/FC6_S/SWO/SECU	JRE_GPIO0_10/ADC Allow	
Pin 11 - [13] PIO0_11/FC6_RXSWCLK/SE	ECURE_GPIO0_11/A Allow	
Pin 12 - [12] PIO0_12/FC3_TWS/SECUR	RE_GPIO0_12/ADC0_Allow	
Pin 13 - [71] PIO0_13/FC1_CTDATA/PLU	J_IN0/SECURE_GPICAllow	
Pin 14 - [72] PIO0_14/FC1_RTO_WS/PLU	J_IN1/SECURE_GPIC Allow	
Pin 15 - [22] PIO0_15/FC6_COUT2/SECU	JRE_GPIO0_15/ADC Allow	
Pin 16 - [14] PIO0_16/FC4_TXINP4/SEC	CURE_GPIO0_16/AD Allow	
Pin 17 - [8] PIO0_17/FC4_SSEOUT0/PLU	I_IN2/SECURE_GPIC Allow	
Pin 18 - [56] PIO0_18/FC4_CIN3/SECUR	E_GPIO0_18/ACMP Allow	
Pin 19 - [90] PIO0_19/FC4_RO_WS/PLU	IN4/SECURE_GPIO Allow	
Pin 20 - [74] PIO0_20/FC3PIO0_20/FC4	TXD_SCL_MISO_W Allow	
Pin 21 - [76] PIO0_21/FC3_RTL3/PLU_C		
Pin 22 - [78] PIO0_22/FC6US/PLU_OUT	-	
Pin 23 - [20] PIO0_23/MCLKSEL0/SECU		
Pin 24 - [70] PIO0_24/FC0_RP8/SCT_GP	I0/SECURE_GPIO0_1 Allow	
Pin 25 - [79] PIO0_25/FC0_TNP9/SCT_G		
Pin 26 - [60] PIO0_26/FC2_RHS_SPI_MC		
Figure 124. Pins tab on LPC55S69		

User Guide for Config Tools for i.MX

🞴 User Memory Regions 🎈	Security Access Configuration 3	×												
SAU MPU Masters MRC0	MBC0 MBC1 MBC2 Pins Intern	upts Mis	cella	neou	IS AC	cess	tem	plate	s					
Access template 🕒 😒														
	1	S-F	Priv		S-U	Jser		NS-	Priv		NS	-Use	r	
ID	Name	R	w	х	R	w	х	R	W	x	R	w	х	Lock
NO_ACCESS	No access													\checkmark
R_s	R for S	\checkmark			\checkmark									\checkmark
RW_s_priv	RW for S-Priv	\checkmark												\checkmark
RW_s	RW for S	\sim			\checkmark									\checkmark
RW_sR_ns_priv	RW for S, R for NS-Priv	\checkmark			~			~						\checkmark
RW_sR_ns	RW for S, R for NS	\checkmark			~			~						\checkmark
RW_sRW_ns_priv	RW for S and NS-Priv				~			~	V					\checkmark
ALL	ALL													

Figure 125. Global Access Templates

			curity Access Config					_												
			MBC1 MBC2 Pi	ns Interrupts	Miso	cellar	eous	Ac	cess	temp	blate	s								
Use g	global acce	ess templates																		
Access	template																			
					S-P	riv		s-u	ser		NS-	Priv		NS-	Use	r				
ID			Name		R	wl		R	w	х	R	w	х	R		x	Lock			
NO_AC	CESS		Local_name		Ť	Ť			Ť			<u> </u>			<u> </u>		LUCK			
R_s			R for S																	
	oriv		RW for S-Priv		V												\checkmark			
RW_s			RW for S		~	~		~									\checkmark			
RW_s_	R_ns_priv	/	RW for S, R for NS-	Priv	\checkmark	\checkmark		~			\checkmark	\square					\checkmark			
RW_s_	R_ns		RW for S, R for NS		\checkmark	\checkmark		\checkmark	\checkmark		\checkmark			\checkmark			\checkmark			
RW_s_	RW_ns_p	riv	RW for S and NS-P	riv	\checkmark	\checkmark		\checkmark	\checkmark		\checkmark	\checkmark					\checkmark			
ALL			ALL		\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\mathbf{V}				
MRC m	emory rea	ions for domain (0																	
Index	Enable	Start address	Size	End address		- 1	C				- 1						1			
index 0	Enable	0x4880_0000	0x2000	0x489F_FFF		Security level							Access template							
1		0x4880_0000	0x2000	0x489F_FFF			S					Local_name								
2		0x4880_0000	0x2000	0x489F_FFF		S						Local_name								
3		0x4880_0000	0x2000	0x489F_FFF			S					Local_name								
4		0x4880_0000	0x2000	0x489F_FFF		S						Local_name								
5		0x4880_0000	0x2000	0x489F_FFF			S				Local_name									
6		0x4880_0000	0x2000	0x489F_FFF			S				Local_name									
7		0x4880_0000	0x2000	0x489F_FFF	F		S					Local_name								
MRC m	emory reg	ions for domain 1	1																	
Index	Enable	Start address	Size	End address			Secu	rity	level			Acce	ess t	empl	ate					
0		0x4880_0000	0x2000	0x489F_FFF	F		S					Loca	il_na	me						
1		0x4880_0000	0x2000	0x489F_FFF			S					Loca								
2		0x4880_0000	0x2000	0x489F_FFF			S					Loca								
3		0x4880_0000	0x2000	0x489F_FFF			S					Loca	_							
4		0x4880_0000	0x2000	0x489F_FFF			S					Loca								
5		0x4880_0000	0x2000	0x489F_FFF			S					Loca								
		0x4880_0000	0x2000	0x489F_FFF	F	S Local_name							_							
6 7		0x4880 0000	0x2000	0x489F_FFF			S			Local_name										

Figure 126. Local access templates

© 2023 NXP B.V. All rights reserved.

5.1.2.7 Miscellaneous

In the **Miscellaneous** subview, you can set various configuration options. The list of these options depends on processor data, and varies greatly. All the options influence your register settings, and can be inspected in the **Register** view. Only some of the options directly influence configuration that you have made in the **Security Access Configuration** view. Point your cursor over individual options to display a tooltip explaining the function of each option.

🔽 User Memory Regions 🦻 Security Access Configuration 🕱	- 8
RDC M7 Domain M4 Domain Miscellaneous	
Use legacy source names	
A M7 Domain	
Global Configuration Lock disabled, registers can be written by any domain	V
✓ Enable Global Valid Access Control	
Enable Global Valid MDAC	
A M4 Domain	
Global Configuration Lock disabled, registers can be written by any domain	V
✓ Enable Global Valid Access Control	
Enable Global Valid MDAC	
Figure 127. Miscellaneous (RDC+XRDC2)	

5.1.3 Memory attribution map

In the Memory attribution map, you can view security levels set for memory regions. This view is read-only.

5.1.3.1 Core 0

In the **Core 0** subview, you can review security levels set for Core 0 to the code, data, and peripherals memory regions. The table is read-only.

The Access by Master table displays MSW or SAU+IDAU, MPC (Memory Protection Checker) security level, and **Resulting access level** status of listed code, data, and peripherals memory regions, alongside their physical addresses.

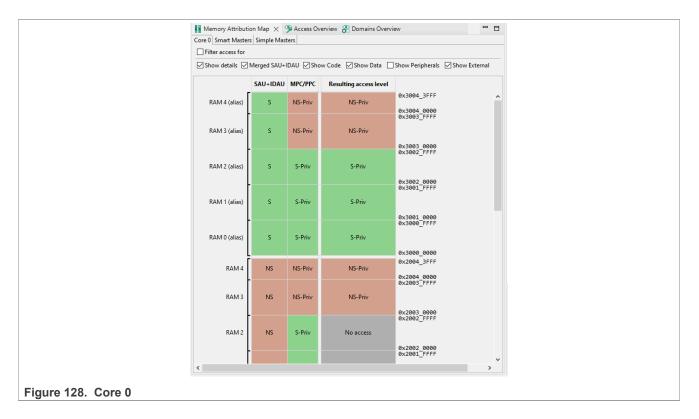
To set the display options, do the following:

- 1. Click the **Filter access for** checkbox to enable filtering options.
- 2. Select the master security access that you want to review by choosing from the Master dropdown menu.
- 3. Optionally, set the security state and execution privilege check-boxes when master allows more security levels. This setting has no effect on the configuration.
- 4. Optionally, customize the output by de-selecting the Show details and Merged SAU+IDAU options.
- 5. Optionally, filter displayed memory regions in the Filter area.

Point your cursor over the color-coded cells to display a tooltip with information about the security level combination.

Double-click the cell to open the pertinent settings in Security Access Configuration.

User Guide for Config Tools for i.MX



5.1.3.2 Simple and Smart masters

In the **Simple Masters** and **Smart Masters subviews**, you can review security attributes of memory in relation to access rights by simple/smart masters. The table is read-only.

To set the display options, do the following:

- 1. Click the **Filter access for** checkbox to enable filtering options.
- 2. Select the master type security access that you want to review by choosing from the **Master** dropdown menu.
- 3. Optionally, customize the output by de-selecting the Show Details, Show Code, Show Data, Show Peripherals, and This Domain Only options.
- 4. Optionally, filter displayed memory regions in the Filter area.

Point your cursor over the color-coded fields to display a tooltip with information about the security level combination.

Double-click the cell to open the pertinent settings in Security Access Configuration.

User Guide for Config Tools for i.MX

Memory Attribution Map × Core 0 Simple Masters Smart		光 Doma	ains Overview		
✓ Filter access for Master:]		D 🗆 se	curity state	execution privilege	
 ✓ Show details ✓ Show Flash ✓ Show SRAI 	M 🗹 Show Peripherals	🗹 Sho	w External RA	M 🗹 Show External device	15
		мsw	MPC/PPC	Resulting access level	
	USB SRAM (alias)	s	S-Priv	No access	0x5010_3FFF 0x5010_0000
	AHB_SECURE_CTRL	s	S-Priv	No access	0x500A_FFFF 0x500A_D000
	AHB_SECURE_CTRL	s	S-Priv	No access	0x500A_CFFF 0x500A_C000
	SECGPIO	s	S-Priv	No access	0x500A_A483 0x500A_8000
	DMA1	S	S-Priv	No access	0x500A_749B 0x500A_7000
	POWERQUAD	S	S-Priv	No access	0x500A_625F 0x500A_6000
AHB peripherals 3 (alias)	CASPER	S	S-Priv	No access	0x500A_5083 0x500A_5000
	HASHCRYPT	S	S-Priv	No access	0x500A_409F 0x500A_4000
	USBHSH	S	S-Priv	No access	0x500A_3053 0x500A_3000
	USBFSH	S	S-Priv	No access	0x500A_205F 0x500A_2000
	ADC0	S	S-Priv	No access	0x500A_0FFF 0x500A_0000

Figure 129. Simple/Smart masters

5.1.4 Access Overview

In Access Overview, you can review security policies you have set in Security Access Configuration view.

The vertical axis displays all masters, divided into color-coded groups by their security settings.

The horizontal axis displays memory ranges and slave buses/peripherals.

Point your cursor at an entry to display a tooltip with information about the entry.

You can group the displayed information by security or by masters by using the button on the right-hand side of the toolbar.

User Guide for Config Tools for i.MX

Nemory Attribution Map 🦻 Access Overview 🛛	L NIC								NIC	Duite	с. 11				
	INS-	User							IN2-	Priv	S-U		S-P	_	
ve	CANFD	Core 0 Data	Core 0 Iructions	DMA0	DMA1	HASHCRYPT	USBFSD	USBFSH	Core 0 Data	Core 0 Iructions	Core 0 Data	Core 0 Iructions	Core 0 Data	Core 0 Iructions	
		0	0			<u> </u>		2	0		0	0	0		
lemory PROGRAM FLASH															
0x0000 0000 - 0x0001 FFFF (0x1000 0000 - 0x1001 FFFF)	n/2			1		1	n/a	n/2	1	1					
Boot-ROM	n/a	•	•	•	•	•	n/a	n/a	•	•	•		•		
	n/2			1			n/a	n/2	1	1					
0x0300_0000 - 0x0301_FFFF (0x1300_0000 - 0x1301_FFFF)	n/a	•	•	•	•	•	n/a	n/a	•	✓ .	•				
0x0400_0000 - 0x0400_3FFF (0x1400_0000 - 0x1400_3FFF)	1	1	1	1	1	1	1	1	1	1	1		1		
BRAM 0	•	•	•	•	•	•	•	•	•	•	•		•		
0x2000_0000 - 0x2000_7FFF (0x3000_0000 - 0x3000_7FFF)	1	1	n/a	1	1	1	1	1	1	n/a	1	n/a	1	n/a	
BRAM 1	•	•	пуа	•	•	•	•	•	•	Π/a	•	пуа	•	Π/a	
0x2000_8000 - 0x2000_BFFF (0x3000_8000 - 0x3000_BFFF)	1	1	n/a	1	1	1	1	1	1	n/a	1	n/a	1	n/a	
USB SRAM	•	•	ny u	•	•		•	•		ηγα	•	ηγα	•	nya	
0x2001_0000 - 0x2001_3FFF (0x3001_0000 - 0x3001_3FFF)	1	1	n/a	1	1	1	1	1	1	n/a	1	n/a	1	n/a	
eripherals			11/ 0							11/ 0		11/ 0			
ADC0	n/a	1	n/a	1	1	n/a	n/a	n/a	1	n/a	1	n/a	1	n/a	
AHB_SECURE_CTRL	n/a		n/a		~		n/a			n/a		n/a	~	n/a	
ANACTRL	n/a	~	n/a	~	~	n/a	n/a	n/a	~	n/a	~	n/a	~	n/a	
CANO	n/a		n/a	~	~		n/a			n/a		n/a	~	n/a	
CASPER	n/a	~	n/a	~	~	n/a	n/a	n/a	~	n/a	~	n/a	~	n/a	
CRC_ENGINE	n/a	~	n/a	~	~	n/a	n/a	n/a	~	n/a	~	n/a	~	n/a	
CTIMER0	n/a	~	n/a	~	~	n/a	n/a	n/a	✓	n/a	~	n/a	~	n/a	
CTIMER1	n/a	~	n/a	~	~	n/a	n/a	n/a	1	n/a	~	n/a	~	n/a	
CTIMER2	n/a		n/a	~	~	n/a	n/a	n/a	1	n/a	~	n/a	~	n/a	
CTIMER3	n/a	✓	n/a	~	~	n/a	n/a	n/a	✓	n/a	✓	n/a	 Image: A second s	n/a	
CTIMER4	n/a	-	n/a	~	~	n/a	n/a	n/a	✓	n/a	✓	n/a	 Image: A second s	n/a	
DBGMAILBOX	n/a	✓	n/a	✓	✓	n/a	n/a	n/a	✓	n/a	~	n/a	✓	n/a	
DMA0	n/a	✓	n/a	~	~	n/a	n/a	n/a	✓	n/a	✓	n/a	✓	n/a	
DMA1	n/a	✓	n/a	✓	✓	n/a	n/a	n/a	✓	n/a	✓	n/a	✓	n/a	
FLASH	n/a	✓	n/a	~	~	n/a	n/a	n/a	✓	n/a	~	n/a	✓	n/a	
FLEXCOMM0	n/a	✓.	n/a	✓	✓	n/a	n/a	n/a	✓	n/a	✓	n/a	✓	n/a	
	nla	.1	2/2			nla	2/2	n/2		2/2		2/2	1	nla	_

5.1.5 Code generation

If the settings are correct and no error is reported, the code generation engine regenerates the source code. You can view the resulting code the **Code Preview** view of the **Trusted Execution Environment** tool.

Code Preview automatically highlights differences between the current and immediately preceding iteration of the code. You can choose between two modes of highlighting by clicking the **Set viewing style for source differences**. You can also disable highlighting altogether from the same dropdown menu. Such features as Copy, Search, Zoom-in, Zoom-out, and Export source are available in the **Code Preview** view. The search can also be invoked by CTRL+F or from the context menu.

Some AHBSC or TRDC with security extension-enabled devices support ROM preset as well as C code. You can choose to have the code generated in the ROM preset by selecting the option in the **Miscellaneous** subview.

5.2 RDC-enabled devices

The features and appearance of the TEE tool are based on the security model of the loaded device.

This section describes the features and appearance of the tool devices enabled with RDC (Resource Domain Controller) and XRDC2 (eXtended Resource Controller 2).

Currently, following devices of this type are supported:

- RT1170
 - Dual core (Cortex-M7 + Cortex-M4): MIMXRT1176, MIMXRT1175, MIMXRT1173
 - Single core only (Cortex-M7): MIMXRT1172, MIMXRT1171
- KW45
- RT118x
- i.MX93

5.2.1 User Memory Regions view

In the **User Memory Regions** view, you can create and maintain a high-level configuration of memory regions and their access templates. You can create the regions, name them, specify their address, size, security level, and provide them with a description. You can then fix any errors in the settings with the help of the **Problems** view.

ID	Start addre	ess Size	End a	address	Nan	ne	Access	Description	n
Region_1	0x0000_00	000 0x1	0x00	00_0000	Regi	ion_1	-		
	• •	Acce	ss configur	ation of the	memo	ry reg	ion		
	7000								
	TRDC								
	Domain 0		Domain 1			Domai			
	Security level	Access template		Access te			ty level	Access template	
	Any		Any			Any			
		No access R for S							
		RW for S-Priv							
		RW for S							
		RW for S, R for NS							100
		RW for S, R for NS						ОК	
		RW for S and NS-	Priv						
		ALL							

Create a new memory region by clicking the Add new memory region button in the view's header.

Enter/change the memory region's parameters by clicking the row's cells.

Modify the access policy of memory regions by clicking the cell in the **Access** column. This action opens the <u>Access templates</u> dialog.

Errors in configuration are highlighted by a red icon in the relevant cell. In the case the issue is easily fixed, you can right-click the cell to display a dropdown list of offered solutions.

Remove the memory region by selecting the table row and clicking the **Remove selected memory region(s)** button in the view's header.

5.2.1.1 Access templates

In the **Access templates** dialog, you can modify access templates for device domains. The dialog displays the device RDC domains, as well as all user-created XRDC2 domains.

Note: Make sure to first specify the number of domains in the M4 Domain/M7 Domain > Domains.

M7 Domain		
Domain 0		
Access templat	e	
No access		_
-		
M4 Domain		
Domain 0		
Access templat	e	
No access		
-		
-	ОК	_

Figure 132. Access template

Select access template by clicking the topmost cell of domain column to open a dropdown list containing all options.

Once you have selected access templates for all domains, click **OK** to return to the **User Memory Regions** view.

5.2.2 Security Access Configuration view

In the **Security Access Configuration** view, you can configure your application's security policies in a number of ways. See the following sections for more details.

5.2.2.1 RDC

In the RDC subview, you can assign masters to domains and specify access rules for slaves for each domain.

5.2.2.1.1 RDC Masters

In the **RDC Masters** subview, you can view available bus masters, allocate them to available domains (cores), and lock/unlock the allocation.

🔽 User Memory Regions 🛿 🌮 Security Access Config	uration 🔀		- 8
RDC M7 Domain M4 Domain Miscellaneous			
Masters Memory Regions Peripherals			
Masters			E
Name	Domain	Lock	
▼ CM7 AHB			
Domain assignment 1	M7 Domain		
▼ CM7 AXI			
Domain assignment 1	M7 Domain		
▶ CM7 DMA			
Domain assignment 1	M4 Domain	\checkmark	
Domain assignment 1	M4 Domain		
CM4 DMA			
▶ CAAM			
LCDIF2 (LCDIFv2)			
▼ ENET_1G TX			
Domain assignment 1	M4 Domain		
▼ ENET_1G RX			
Domain assignment 1	M4 Domain		
▼ ENET			
Domain assignment 1	M7 Domain		
▶ ENET QOS			
▼ USDHC1			
Domain assignment 1	M7 Domain	\checkmark	
▼ USDHC2			
Domain assignment 1	M4 Domain	\checkmark	
▼ USB			
Domain assignment 1	M7 Domain		
▶ GC355 (GPU2D)			
▶ PXP			
LCDIF1 (eLCDIF)			
▶ CSI			
Figure 133. RDC Masters			

Allocate a master to a domain by clicking the cell in the **Domain** column in the **Masters** table and selecting the domain from the dropdown list.

Select the **Lock** checkbox to prevent further register modifications.

Alternatively, you can select the options by right-clicking the master and using the dropdown list.

Note: Some masters are allocated to specific domains by default and cannot be reallocated.

5.2.2.1.2 Memory Regions

In the **Memory Regions** subview, you can view, enable/disable, and configure the MRC (Memory Region Controller) bus slaves and their domain access.

Memory Region Controller implements the access controls for slave memories based on the pre-programmed Memory Region Descriptor registers.

_		4 Domain Misce gions Periphera					
		nfiguration					(+) (=
Netholy R				1	1	M7 Domain	M4 Domain
ndex	Enable	Address	Size	End Address	Lock	Access Template	Access Template
CAAM S	Secure R	AM: 0x0028_0	000 - 0x002				
0		0x0000_0000	0x2000	0x0000_1FFF		RW	RW
OCRAM	M4: 0x2	020_0000 - 0	x2023_FFFF				
0	\checkmark	0x0002_0000	0x2_0000	0x0003_FFFF		R	RW
1		0x0000_0000	0x2_0000	0x0001_FFFF		RW	RW
OCRAM	L: 0x202	4_0000 - 0x2	02B_FFFF				
0	\checkmark	0x0000_0000	0x8_0000	0x0007_FFFF	\checkmark	RW	R
OCRAM2	2: 0x202	C_0000 - 0x2	033_FFFF				
0	\checkmark	0x0000_0000	0x8_0000	0x0007_FFFF		RW	R
OCRAM	M7: 0x2	036_0000 - 0	x203F_FFFF		_		
0		0x0000_0000		0x0000_007F		RW	RW
FlexS	PI1: 0x3	000_0000 - 0	x3FFF_FFFF				
0	\checkmark	0x0000_0000		0x00FF_FFFF	\checkmark	RW	R
SIM_D	[SP: 0x4	100_0000 - 0	_				
0		0x0000_0000		0x000F_FFFF		RW	RW
	4: 0x411	0_0000 - 0x4	_				
0		0x0000_0000	_	0x000F_FFFF		RW	RW
_	7: 0x414	0_0000 - 0x4	_				
0		0x0000_0000		0x000F_FFFF		RW	RW
	-12: 0x6	000_0000 - 0	_	0.0000.05	-		514
0		0x0000_0000		0x0000_0FFF	V	No Access	RW
	000820	0000 - 0xDFF	_	0.0055 5555		DW	DW
0		0x0000_0000		0x02FF_FFFF		RW	RW
1		0x0300_0000		0x03FF_FFFF		RW	RW
2		0x0400_0000	0x200_0000	0x05FF_FFFF	\mathbf{V}	No Access	RW

Use the Memory Regions Configuration table to enable and configure MRC slaves:

- 1. Enable the region.
- 2. Specify the **Address**.
- 3. Specify either the Size or the End Address.
- 4. Optional: Lock the settings to prevent further register modifications.

5. Set the Access Template for available domains.

Alternatively, you can select the options by right-clicking the master and using the dropdown list.

5.2.2.1.3 Peripherals

In the **Peripherals** subview, you can view and configure the PDAP (Peripheral Domain Access Permissions) for peripherals.

RDC M7 Domain M4 Domain Miscellaneous					
Masters Memory Regions Peripherals					
Peripherals Configuration				(+	Ξ
			M7 Domain	M4 Domain	^
Sector	Lock	Usehore	Accessmplate	Accessmplate	
ACMP1			RW	RW	
ACMP2			RW	RW	
ACMP3			RW	RW	
ACMP4			RW	RW	
ADC1			RW	RW	
ADC2			RW	RW	
ADC_ETC			RW	RW	
ANALOG / ANADIG			RW	RW	
AOI2 / AOI1 / XBAR3 / XBAR2 / XE			RW	RW	
ASRC			RW	RW	
CAAM (0x4044_0000 0x4053_3FFF			RW	RW	
CAN1			RW	RW	
CAN2			RW	RW	
CAN3			RW	RW	
CCM			RW	RW	
CSI			RW	RW	
DAC			RW	RW	
DCDC			RW	RW	
DEC1			RW	RW	
DEC2			RW	RW	
DEC3			RW	RW	
DEC4			RW	RW	

Use the **Peripherals Configuration** table to enable and configure PDAP:

- 1. Optional: **Lock** the settings to prevent further register entries.
- 2. Select **Use semaphore** to enable the semaphore function for the peripheral. *Note:* When enabled, the master cannot access this peripheral until obtaining a semaphore. During the time that the domain has the semaphore in possession, its bus masters have exclusive access to the peripheral.

3. Set the Access Template for available domains.

5.2.2.2 XRDC2 Domains view

In the **M7/M4 Domain** subviews, you can view and configure security policies of the XRDC2(eXtended Resource Domain Controller 2) domains. Each CPU can contain up to 16 domains.

5.2.2.2.1 MPU

In the **MPU** subview, you can enable and configure MPU (Memory Protection Unit). You can create regions, specify their address, size, and other parameters.

The MPU enforces privilege rules, separates processes, and enforces access rules to memory, and supports the standard ARMv7 Protected Memory System Architecture model.

MPU is disabled by default and must be enabled by selecting the **Enable MPU** option.

Note: Not every device supports MPU.

User Guide for Config Tools for i.MX

	mains N	lasters Periphe	rals M	emory Re	gions	Mem	ory Slot	s				
Option	s											
Enab	le MPU	Enable MPU	during	g HardFau	It and	NMI h	andlers					
Enab	le privile	eged software a	ccess	to the def	fault m	emory	/ map					
Gene	erate sou	rces for disable	d regi	ons								
MPU Me	emory At	tributes										
			1	Inner Attr	ributes			Oute	r Attril	outes		
Index	Î ID	Memory T	ype	С	В	1	w	С	1	3	w	
0	0	Device		n/a	n/a	r	n/a	n/a	r	n/a	n/a	
1	1	Device		n/a	n/a	r	n/a	n/a	r	n/a	n/a	
10	10	Device		n/a	n/a	r	n/a	n/a	r	n/a	n/a	
11	11	Device		n/a	n/a	r	n/a	n/a	r	n/a	n/a	
12	12	Device		n/a	n/a	r	n/a	n/a	r	n/a	n/a	
13	13	Device		n/a	n/a	r	n/a	n/a	r	n/a	n/a	
14	14	Device		n/a	n/a	r	n/a	n/a	r	n/a	n/a	
15	15	Device		n/a	n/a	r	ı/a	n/a	r	n/a	n/a	
2	2	Device		n/a	n/a	r	n/a	n/a	r	n/a	n/a	
3	3	Device		n/a	n/a		n/a	n/a		n/a	n/a	
4	4	Device		n/a	n/a		ı/a	n/a		ı/a	n/a	
5	5	Device		n/a	n/a	r	ı/a	n/a	r	1/a	n/a	
MPU Me	emory Re	egions	1	1		1	1		1	1		1
Index	Enable	Address	Size	End Ad		Exec		issions	SRD	Share	eability	N
0		0x0000_0000	0x20				RW p		0b0	No		0
1		0x0000_0000	0x20	-			RW p		0b0	No		1
10		0x0000_0000	0x20	0x0000_			RW p		0b0	No		10
11		0x0000_0000	0x20	-			RW p		0b0	No		11
12		0x0000_0000	0x20	-			RW p		0b0	No		12
13		0x0000_0000	0x20				RW p		0b0	No		13
14		0x0000_0000	0x20				RW p		0b0	No		14
15 2		0x0000_0000 0x0000_0000	0x20				RW p		0b0	No		15
2 3		0x0000_0000	0x20 0x20				RW p		0b0 0b0	No No		2
3		00000_00000	0X20	0x0000	0016		RW p	IIV	000	NO		3

Figure 136. MPU

Use the **MPU Memory Attributes** table to name and configure MPU memory attribute sets. Click the cells of the **Memory Type** and **Inner/Outer Attributes** columns to display the available options.

Use the **MPU Memory Regions** table to enable and configure MPU memory regions.

1. Enable the region.

User Guide for Config Tools for i.MX

- 2. Specify the Address.
- 3. Specify either the Size or the End Address.
- 4. Set the **Exec** option if you want the region to be able to run code.
- 5. Set the Permissions.
- 6. Set the SRD (Sub Region Disable) bits.
- 7. Set the **Shareability**, or the caching options.

5.2.2.2.2 Domains

In the **Domains** subview, you can view, add/remove, and rename XRDC2 domains. Each CPU supports up to 16 XRDC2 domains.

RDC M7 Domain M4 Domain Miscellane		
MPU Domains Masters Peripherals Men	hory Regions Memory Slots	
Domains 💽 😣		
Domain	Name	
Domain 0	Domain 0	
Domain 1	Domain 1	
Domain 2	Domain 2	

Add a new domain by clicking the Add new domain button.

Rename the domain by entering a new name in the Name column.

Remove a domain by clicking the **Remove last domain** button.

5.2.2.2.3 Masters

In the **Masters** subview, you can add/remove, view, configure XRDC2 domain assignments to available RDC masters.

Master Domain Assignment Controller (MDAC) is responsible for the generation of the DID, nonsecure and privileged attributes for every system bus transaction in the device based on pre-programmed Master Domain Assignment (MDA) registers.

User Guide for Config Tools for i.MX

💫 User Memory Regions 🤣 Security Access Configura	ition 🔀		- 8
RDC M7 Domain M4 Domain Miscellaneous			
Masters Memory Regions Peripherals			
Masters			Ē
Name	Domain	Lock	
✓ CM7 AHB			
Domain assignment 1	M7 Domain		
✓ CM7 AXI			
Domain assignment 1	M7 Domain		
▶ CM7 DMA			
▼ CM4 AHBC			
Domain assignment 1	M4 Domain		
Domain assignment 1	M4 Domain		
CM4 DMA			
▶ CAAM			
LCDIF2 (LCDIFv2)			
✓ ENET_1G TX			
Domain assignment 1	M4 Domain		
▼ ENET_1G RX			
Domain assignment 1	M4 Domain		
▼ ENET			
Domain assignment 1	M7 Domain		
ENET QOS			
▼ USDHC1			
Domain assignment 1	M7 Domain	\checkmark	
▼ USDHC2			
Domain assignment 1	M4 Domain	\checkmark	
▼ USB			
Domain assignment 1	M7 Domain		
▶ GC355 (GPU2D)			
▶ PXP			
LCDIF1 (eLCDIF)			
▶ CSI			

To add a new domain assignment:

- 1. Click the Add new domain assignment for the selected master button.
- 2. Select the Enable checkbox.
- 3. Enter the **Match Input** value. **Note:** The match field specifies the reference value for the comparison with the MDAC match input. The match field width varies by MDAC instance from 0 to 16 bits. Unimplemented bits are read as 0. A size of 0 bits generates a hit on all comparisons.
- 4. Enter the Mask Input value.

User Guide for Config Tools for i.MX

Note: The mask field specifies which bits are valid for the match comparison. Only bit positions in which the mask value is zero are compared. The mask field width is the same as the mask field which varies by MDAC instance from 0 to 16 bits. A mask value of all ones generates a hit on all comparisons.

- 5. Select the XRDC2 domain assignment from the dropdown list in the **Domain** column.
- 6. Select the security access type from the dropdown list in the Secure column.
- 7. Select the privileged access type from the dropdown list in the **Privileged** column.
- 8. Optional: select the Lock checkbox to prevent further register modifications.

5.2.2.2.4 Peripherals

In the **Peripherals** subview, you can view the access templates for PAC (Peripheral Access Controller) and configure access for all peripherals managed by PAC on the selected RDC domain.

The Peripheral Access Controller submodule performs access control for a set of peripherals connected to a peripheral bus bridge or integrated into a peripheral subsystem.

The **Access Template** table displays the ID and name of all access templates available for the PAC on the selected device. The information is data driven and display-only.

PC MPU RDC M7 Domain M4 Domain	Sess Configuration X											' 🗆
Domains Masters Peripherals Memory Re	egions Memory Slots											
Access template												
				S-P	riv	S-U	lser	NS-	Priv	NS-	User	
ID	Name			R	W	R	W	R	W	R	W	
NO_ACCESS	No access			' —'		' — '				' — '		
R_s	R for S			\checkmark	\square	\checkmark						
RW_s_priv	RW for S-Priv			V	\checkmark			i i i	i i i	\square	i i i	
RW_s	RW for S			V	\checkmark		\checkmark					
RW_sR_ns_priv	RW for S, R for NS-	-Priv			\checkmark	\checkmark						
RW_sR_ns	RW for S, R for NS			\checkmark	\checkmark		\checkmark			\checkmark		
RW_sRW_ns_priv	RW for S and NS-P	riv		\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark			
ALL	All			\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark		
Sector		Enable	EAL	Lo	ck			Aco	cess te	empla	ite	
								Dor	main (0		-
Sector		Enable	LEAL	LO	СК			ACC	cess to	empia	ite	
ACMP1			Disabled	Die	ablad			No	20000	0		
ACMP1 ACMP2			Disabled Disabled		abled				acces acces			-1
ACMP2			Disabled	Dis	abled			No	acces	s		-
ACMP2 ACMP3			Disabled Disabled	Dis Dis				No No	acces acces	is is		
ACMP2 ACMP3 ACMP4			Disabled	Dis Dis Dis	abled abled			No No No	acces	is is		
ACMP2 ACMP3			Disabled Disabled Disabled	Dis Dis Dis Dis	abled abled abled			No No No	acces acces acces	is is is		
ACMP2 ACMP3 ACMP4 ADC_ETC			Disabled Disabled Disabled Disabled	Dis Dis Dis Dis Dis	abled abled abled abled	 		No No No No	acces acces acces acces acces	is is is is		
ACMP2 ACMP3 ACMP4 ADC_ETC ANALOG/ANADIG			Disabled Disabled Disabled Disabled Disabled	Dis Dis Dis Dis Dis Dis	abled abled abled abled abled			No No No No	acces acces acces acces acces acces	is is is is is		
ACMP2 ACMP3 ACMP4 ADC_ETC ANALOG/ANADIG A0I1			Disabled Disabled Disabled Disabled Disabled Disabled	Dis Dis Dis Dis Dis Dis Dis Dis	abled abled abled abled abled abled			No No No No No	acces acces acces acces acces acces acces	is is is is is is is		
ACMP2 ACMP3 ACMP4 ADC_ETC ANALOG/ANADIG A011 A012			Disabled Disabled Disabled Disabled Disabled Disabled	Dis Dis Dis Dis Dis Dis Dis Dis Dis	abled abled abled abled abled abled abled			No No No No No No	acces acces acces acces acces acces acces acces	is is is is is is is is		
ACMP2 ACMP3 ACMP4 ADC_ETC ANALOG/ANADIG A011 A012 APC_IEE			Disabled Disabled Disabled Disabled Disabled Disabled Disabled	Dis Dis Dis Dis Dis Dis Dis Dis Dis Dis	abled abled abled abled abled abled abled abled			No No No No No No	acces acces acces acces acces acces acces acces acces	:S :S :S :S :S :S :S :S :S		
ACMP2 ACMP3 ACMP4 ADC_ETC ANALOG/ANADIG A011 A012 APC_IEE ASRC			Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled	Dis Dis Dis Dis Dis Dis Dis Dis Dis Dis	abled abled abled abled abled abled abled abled abled			No No No No No No No No	acces acces acces acces acces acces acces acces acces	55 55 55 55 55 55 55 55 55 55 55		
ACMP2 ACMP3 ACMP4 ADC_ETC ANALOG/ANADIG A011 A012 APC_IEE ASRC CAN1			Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled	Dis Dis Dis Dis Dis Dis Dis Dis Dis Dis	abled abled abled abled abled abled abled abled abled			No No No No No No No No	acces acces acces acces acces acces acces acces acces acces	55 55 55 55 55 55 55 55 55 55 55 55		
ACMP2 ACMP3 ACMP4 ADC_ETC ANALOG/ANADIG A011 A012 APC_IEE ASRC CAN1 CAN2			Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled	Dis Dis Dis Dis Dis Dis Dis Dis Dis Dis	abled abled abled abled abled abled abled abled abled abled			No No No No No No No No No	acces acces acces acces acces acces acces acces acces acces acces	s s s s s s s s s s s s s s s s s s s		
ACMP2 ACMP3 ACMP4 ADC_ETC ANALOG/ANADIG A011 A012 APC_IEE ASRC CAN1 CAN2 CAN3			Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled	Dis Dis Dis Dis Dis Dis Dis Dis Dis Dis	abled abled abled abled abled abled abled abled abled abled abled			No No No No No No No No No	acces acces acces acces acces acces acces acces acces acces acces acces	s s s s s s s s s s s s s s s s s s s		
ACMP2 ACMP3 ACMP4 ADC_ETC ANALOG/ANADIG A011 A012 APC_IEE ASRC CAN1 CAN2 CAN3 CCM			Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled	Dis Dis Dis Dis Dis Dis Dis Dis Dis Dis	abled abled abled abled abled abled abled abled abled abled abled abled	 		No No No No No No No No No No No	acces acces acces acces acces acces acces acces acces acces acces acces			
ACMP2 ACMP3 ACMP4 ADC_ETC ANALOG/ANADIG AOI1 AOI2 APC_IEE ASRC CAN1 CAN2 CAN3 CCM CCM (OBS)			Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled	Diss Diss Diss Diss Diss Diss Diss Diss	abled abled abled abled abled abled abled abled abled abled abled abled			No No No No No No No No No No No	acces acces acces acces acces acces acces acces acces acces acces acces acces			
ACMP2 ACMP3 ACMP4 ADC_ETC ANALOG/ANADIG A011 A012 APC_IEE ASRC CAN1 CAN2 CAN3 CCM CCM (OBS) CSI			Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled	Diss Diss Diss Diss Diss Diss Diss Diss	abled abled abled abled abled abled abled abled abled abled abled abled abled			No No No No No No No No No No No No	acces acces acces acces acces acces acces acces acces acces acces acces acces acces	S S S S S S S S S S S S S S S S S S S		

Figure 139. Peripherals

IMXUG

106 / 141

Use the Peripherals Configuration table to configure access for a peripheral:

- 1. Select the **Enable** checkbox.
- 2. Set the Lock to the desired state.
- 3. Set the Access Template for all listed domains.

Alternatively, you can select the options by right-clicking the master and using the dropdown list.

5.2.2.2.5 Memory Regions

In the **Memory Regions** subview, you can view the access templates for MRC (Memory Region Controller) and configure access for all non-peripheral memory spaces managed by MRC on the selected RDC domain.

The Memory Region Controller (MRC) provides domain-based, hardware access control for all system bus references targeted at non-peripheral memory spaces.

The **Access Template** table displays the ID and name of all access templates available for the MRC on the selected device. The information is data driven and display-only.

🔽 User Memory Regions 💖 Security Access Configuration 🗙							- 8
APC MPU RDC M7 Domain M4 Domain B Miscellaneous							
Domains Masters Peripherals Memory Regions Memory Slots							
Access template							
					S-Priv	S-User N.	iv Nr
ID	Name				RW	/ R W R	WRW
NO_ACCESS	No access	5					
R_s	R for S						
RW_s_priv	RW for S-	Priv			\checkmark		
RW_s	RW for S						
RW_sR_ns_priv	RW for S,	R for NS-Priv			2222 文文文文		
RW_sR_ns	RW for S,	R for NS			\checkmark		
RW_sRW_ns_priv	RW for S a	and NS-Priv			\checkmark		
ALL	All						
Memory Regions Configuration							
		1					Domain 0
Index	Enable	Start address	Size	End address	EAL	Lock	Access template
CAAM Secure RAM: 0x0028_0000 - 0x0028_FFFF							
0		0x0028_0000	0x1000	0x0028_0FFF	Disabled	Lock enabled	R for S
- OCRAM M4 (LMEM backdoor): 0x2020_0000 - 0x2023_FF	FF						
0	\checkmark	0x2020_0000	0x4000	0x2020_3FFF	Disabled	Disabled	No access
- OCRAM1: 0x2024_0000 - 0x202B_FFFF							
0		0x2024_0000	0x1000	0x2024_0FFF	Disabled	Disabled	No access
- OCRAM2: 0x202C_0000 - 0x2033_FFFF							
0		0x202C_0000	0x1000	0x202C_0FFF	Disabled	Disabled	No access
- OCRAM1 ECC: 0x2034_0000 - 0x2034_FFFF							
0		0x2034_0000	0x1000	0x2034_0FFF	Disabled until reset	Disabled	RW for S
- OCRAM2 ECC: 0x2035_0000 - 0x2035_FFFF							
0		0x2035_0000	0x1000	0x2035_0FFF	Enabled	Enabtself	No access
<pre>- OCRAM + ECC (FlexRAM): 0x2036_0000 - 0x203F_FFFF</pre>							
0		0x2036_0000	0x2000	0x2036_1FFF	Disabled	Disabled	No access
<pre>- SEMC: 0x8000_0000 - 0xDFFF_FFFF</pre>							
0	\checkmark	0x8000_0000	0x600_0000	0x85FF_FFFF	Disabled	Lock enabled	All
1	\checkmark	0x8000_0000	0x600_0000	0x85FF_FFFF	Disabled	Disabled	No access

Figure 140. Memory Regions

Use the Memory Regions Configuration table to configure access for a non-peripheral memory space:

- 1. Select the Enable checkbox.
- 2. Specify the Start Address.
- 3. Specify either Size or End Address.
- 4. Set the Lock to the desired state.
- 5. Set the Access Template for all listed domains.

Alternatively, you can select the options by right-clicking the master and using the dropdown list.

5.2.2.2.6 Memory Slots

In the **Memory Slots** subview, you can view the access templates for MSC (Memory Slot Controller) and configure access for all memory spaces managed by MSC on the selected RDC domain.

The Memory Slot Controller (MSC) performs access control for a peripheral or memory space with a fixed address range.

The **Access Template** table displays the ID and name of all access templates available for the MSC on the selected device. The information is data driven and display-only.

PC MPU RDC M7 Domain M4 Domain	Miscellaneous				
Domains Masters Peripherals Memory Reg	ions Memory Slots	1			
Access template					
				S-Priv S-U	Jser Niv Nr
ID	Name	Name			WRWRW
NO_ACCESS	No access				
R_s	R for S				
RW_s_priv	RW for S-Priv				
RW_s	RW for S	RW for S			
RW_sR_ns_priv	RW for S, R fo	RW for S, R for NS-Priv			
RW_sR_ns	RW for S, R fo	RW for S, R for NS			
RW_sRW_ns_priv	RW for S and	RW for S and NS-Priv			
ALL	All				
Mamory Clate Configuration					
Memory Slots Configuration					œ ⊡
Memory Slots Configuration		1	1		Domain 0
Sector		Enable	EAL	Lock	1
Sector - ROM MSC: 0x0020_0000 - 0x0023_1	FFFF	Enable	1		Domain 0 Access template
Sector - ROM MSC: 0x0020_0000 - 0x0023_1 0x0020_0000 - 0x0023_FFFF		Enable	EAL	Lock	Domain 0
Sector - ROM MSC: 0x0020_0000 - 0x0023_ 0x0020_0000 - 0x0023_FFFF - GPV0 MSC: 0x4100_0000 - 0x410F		Enable	Disabled	Disabled	Domain 0 Access template No access
Sector - ROM MSC: 0x0020_0000 - 0x0023_ 0x0020_0000 - 0x0023_FFFF - GPV0 MSC: 0x4100_0000 - 0x410F_ 0x4100_0000 - 0x410F_FFFF	_FFFF	Enable	1		Domain 0 Access template
Sector - ROM MSC: 0x0020_0000 - 0x0023_0 0x0020_0000 - 0x0023_FFFF - GPV0 MSC: 0x4100_0000 - 0x410F_ 0x4100_0000 - 0x410F_FFFF - GPV1 MSC: 0x4110_0000 - 0x411F_	_FFFF	Enable	Disabled Disabled	Disabled Disabled	Domain 0 Access template No access No access
Sector - ROM MSC: 0x0020_0000 - 0x0023_0 0x0020_0000 - 0x0023_FFFF - GPV0 MSC: 0x4100_0000 - 0x410F_ 0x4100_0000 - 0x410F_FFFF - GPV1 MSC: 0x4110_0000 - 0x411F_ 0x4110_0000 - 0x411F_FFFF	_FFFF	Enable	Disabled	Disabled	Domain 0 Access template No access
Sector - ROM MSC: 0x0020_0000 - 0x0023_0 0x0020_0000 - 0x0023_FFFF - GPV0 MSC: 0x4100_0000 - 0x410F_ 0x4100_0000 - 0x410F_FFFF - GPV1 MSC: 0x4110_0000 - 0x411F_	_FFFF	Enable	Disabled Disabled	Disabled Disabled	Domain 0 Access template No access No access

Figure 141. Memory Slots

Use the **Memory Slots Configuration** table to configure access for a memory space:

- 1. Select the **Enable** checkbox.
- 2. Set the Lock to the desired state.
- 3. Set the Access Template for all listed domains.

Alternatively, you can select the options by right-clicking the master and using the dropdown list.

5.2.2.3 XRDC (eXtended Trusted Resource Domain Controller) on Cortex-A35 in i.MX8 ULP

5.2.2.3.1 Masters

XRDC masters are similar to TRDC masters. In addition, the following features are supported:

- PID (Process Identifier) is combined with the PIDM field to determine the domain hit.
- **PIDM (PID Mask)** provides a masking capability so that multiple process identifiers can be included as part of the domain hit determination. If a bit in the PIDM is set, the corresponding bit of the PID is ignored in the comparison.
- **PID enable** provides the ability to include inclusive or exclusive sets of masked PID values. Allowed values are 00b, 01b, 10b, and 11b. For more info, see the Reference Manual (link to be provided).

🔺 🛕 📄 Update Co	de - Function	al Group	BOARD_InitTE	E		· • • E)] 🞺 🗠			
🔒 User Memory Reg	jions 🦻 Sec	urity Access	Configurati	on ×						
A35 Domain ¹ M33	Domain 🕫 Mi	scellaneous								
Access Templates	Aasters PAC	PACO PAG	C1 PAC2 M	RC0 to ROM	1 MRC1 to	FlexSPI2	MRC2 to s	SRAM2	MRC3 to SRA	M0 "12
Options										
🗹 Global Valid for	MDACs(XRD	C global ena	ble/disable)							
Masters										
Name	Enable	Dout	Domain	IDass	Secure	Prged	Lock	PID	PID mask	PID enable
- CA35										
Domaient 1	\checkmark	n/a	Domain 0		n/a	n/a		0b0	0b0	0b10
T DMA1										
Domaient 1		n/a	Domain 1		NS	User		n/a	n/a	n/a
- USB0/USB1										
Domaient 1		n/a	Domain 1		NS	User		n/a	n/a	n/a
- PXP[M10]										
Domaient 1		n/a	Domain 1		NS	User		n/a	n/a	n/a
Figure 142. XtRD)C master t	ab								

5.2.2.3.2 MRC

MRC on XRDC is similar to MRC on TRDC. There are several minor differences:

- 1. There is only one instance of the memory regions table because address ranges are shared across all domains. For each memory region, the user can specify an access template for each domain.
- 2. The code region specifies which templates would be used (0= data, 1 = code). The templates are now hybrid. It means that there are two templates for the same ID and name the first row is for the data region and the second row is for the code region. These templates, which have the lock field, can be edited by clicking the desired access box.

User Guide for Config Tools for i.MX

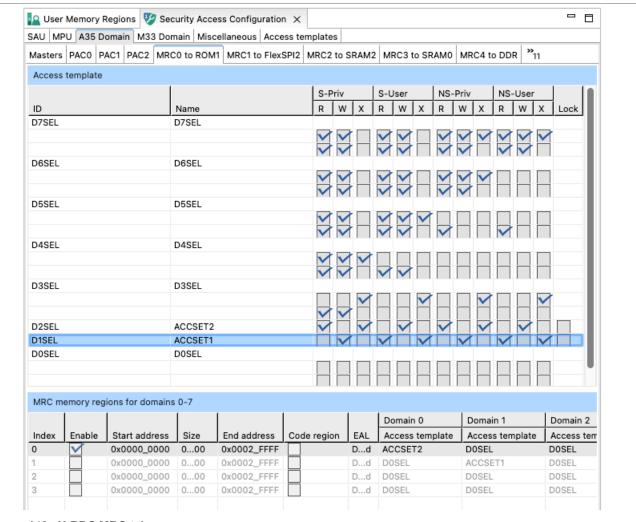


Figure 143. XtRDC MRC tab

5.2.2.3.3 Access control modes

There are two modes that can be enabled for PID.

For processors only supporting TSM, the Three-State Model (SecurePriv, SecureUser, NonsecureUser), the nonsecure[n] output signal from the MDAC submodule is forced to zero while in privileged mode to enable precise state transitions between the user and privileged modes. When SP4SM, the Special 4-State Model, is enabled, the MDAC does not use the MDA[DIDS,DID] fields. The MDAC tracks the current access level and generates specific domainIDs for specific access levels.

User Guide for Config Tools for i.MX

BOARD_InitTEE				
User Memory Regions 🦻 Security Access Configuration ×	Im Memory Attribution Map Access Overview × A Domains Overview	iow		
35 Domain [®] M33 Domain [®] Miscellaneous	A35 Domain M33 Domain			
	Domain 0 Domain 1 Domain 2 Domain 3 Domain 4 Domain 5 Domair	6 Domain 7		
Use legacy source names	Domain 0 Domain 1 Domain 2 Domain 3 Domain 4 Domain 5 Domain		1	lan: l
✓ A35 Domain		NS-User		S-Priv
Global Valid for MRCs		CA35	CA35	CA35
Global Valid for PACs/MSCs	Slave	Q	Q	J
	▼ Memory			
Global Valid for MDACs(XRDC global enable/disable)	▼ A35 ROM-BOOT			
✓ A35 PID	0x0000_0000 - 0x0001_7FFF	~	~	~
	0x0001_8000 - 0x0002_FFFF	\checkmark	~	\checkmark
Lock Register can be written by any secure privileged write	SRAW 1			
Three-state model	0x2100_0000 - 0x2100_FFFF (0x3100_0000 - 0x3100_FFFF)	~	\checkmark	\checkmark
Special 4-state model				4
	0x2117_0000 - 0x2117_FFFF (0x3117_0000 - 0x3117_FFFF)	~	\checkmark	\checkmark
 M33 Domain 	▼ HiFi4 DTCM			
Global Valid for Domain Assignment Controllers	0x2118_0000 - 0x2118_FFFF (0x3118_0000 - 0x3118_FFFF)	~	v	×
Global Valid for Memory Block Checkers		~	~	1
	0x2201_0000 - 0x2201_FFFF (0x3201_0000 - 0x3201_FFFF) • SRAM 2	~	~	~
Global Valid for Memory Region Checkers	0x2202_0000 - 0x2205_FFFF (0x3202_0000 - 0x3205_FFFF)	~	~	
	<pre>visit = visit = v</pre>	•	¥	~
	0x2600_0000 - 0x2600_7FFF (0x3600_0000 - 0x3600_7FFF)	~	1	1
	TMA1-CHx	•	•	-
	0x2902_0000 - 0x2921_FFFF (0x3902_0000 - 0x3921_FFFF)	1	1	1
	~ GPIOE_REGS	•	Ť	

5.2.2.4 Trusted Resource Domain Controller on Cortex-M33 in i.MX8 ULP and KW45 (TRDC)

5.2.2.4.1 MPU

This MPU is identical to other MPUs with Cortex-M33 (for example, LPC55S) or other cores based on the Armv8-M architecture or above with Secure/Non-Secure register banks.

5.2.2.4.2 Domains

The domains are similar to RDC/XRDC2/XRDC: assignment of chip resources to processing "domains", where a unique domain identifier (domainID, DID) is assigned to each processing domain. The number of supported DIDs is typically the number of CPUs plus one.

5.2.2.4.3 Masters

Masters are similar to Masters in <u>XRDC2</u> on MIMXRT117x. The user can also choose the domain ID input or ID bypass depending on the master type.

5.2.2.4.4 Access templates

Access templates are similar to patterns in XRDC2 on MIXRT117x. The main difference is as follows: you can switch between "global" (for the entire RDC, used by all checkers, and editable) and "local" (specific to the checker and immutable) templates; meanwhile access templates in XRDC2 are always validator-dependent and editable.

User Guide for Config Tools for i.MX

SAU MPU A35 Domain N	133 Domain Miscellaneous Access	template	s											
Access template														
		S-P	S-Priv		S-L	S-User			NS-Priv			-Use		
ID	Name	R	w	х	R	w	х	R	W	х	R	w	х	Lock
NO_ACCESS	No access													
R_s	R for S	\checkmark			\checkmark									\checkmark
RW_s_priv	RW for S-Priv	\checkmark												\checkmark
RW_s	RW for S	\checkmark												\checkmark
RW_sR_ns_priv	RW for S, R for NS-Priv	\checkmark						\checkmark						\checkmark
RW_sR_ns	RW for S, R for NS	\checkmark						\checkmark						\checkmark
RW_sRW_ns_priv	RW for S and NS-Priv	\checkmark							V					\checkmark
ALL	ALL													

Figure 145. Access templates

5.2.2.4.5 MRC

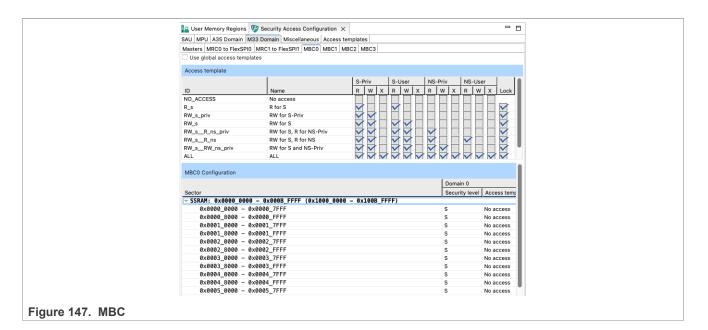
MRC on TRDC is similar to to MRC (Memory Regions) in XRDC2.

				us Access template		
			C1 to FlexSPI1 MBC	CO MBC1 MBC2 N	/BC3	
Use	e global ac	cess templates				
Acces	ss templat	9				
ID			Name	S-F	Priv S-User	NS-Priv NS-User R W X R W X Lock
	CCESS		No access			ا استقار النظر النظر النظر النظر
R_s			R for S	\checkmark		
RW_s	_priv		RW for S-Priv	\checkmark		
RW_s			RW for S	\checkmark		
	R_ns_p	iv	RW for S, R for NS	S-Priv		
	R_ns		RW for S, R for NS RW for S and NS-			
ALL	RW_ns_	priv	ALL	Priv		YYLLLY'
		gions for domai		V		
Index		Start address		End address	Security level	Access template
0		0x0400_0000		0x0BFF_FFFF	S	No access
1		0x0400_0000			S	No access
2		0x0400_0000	0 0x000	0x0BFF_FFFF	S	No access
3		0x0400_0000	0 0x000		S	No access
4		0x0400_0000			S	No access
5		0x0400_0000			S	No access
6		0x0400_0000			S	No access
7		0x0400_0000	0 0x000	0x0BFF_FFFF	S	No access
MRC	memory re	gions for domain	n 1			
Index	Enable	Start address		End address	Security level	Access template
0		0x0400_0000			S	No access
1		0x0400_0000			S	No access
2		0x0400_0000			S	No access
3		0x0400_0000			S	No access
4		0x0400_0000			S	No access No access
6		0x0400_0000			S	No access
7	H	0x0400_0000			S	No access
		-10-100_0000				

5.2.2.4.6 MBC

MBC in TRDC is similar to MSC (Memory Slots) in XRDC2 and MSC in XRDC.

User Guide for Config Tools for i.MX



5.2.2.5 Miscellaneous

In the **Miscellaneous** subview, you can set various configuration options. The list of these options depends on processor data, and varies greatly. All the options influence your register settings, and can be inspected in the **Register** view. Only some of the options directly influence configuration that you have made in the **Security Access Configuration** view. Point your cursor over individual options to display a tooltip explaining the function of each option.

🔽 User Memory Regions 🦻 Security Access Configuration 🕱									
RDC M7 Domain M4 Domain Miscellaneous									
Use legacy source names									
M7 Domain									
Global Configuration Lock disabled, registers can be written by any domain	▼								
✓ Enable Global Valid Access Control									
Enable Global Valid MDAC									
A M4 Domain									
Global Configuration Lock Lock disabled, registers can be written by any domain	▼								
✓ Enable Global Valid Access Control									
Enable Global Valid MDAC									
Figure 148. Miscellaneous (RDC+XRDC2)									

5.2.3 Memory Attribution Map

In the **Memory Attribution Map** view, you can review access levels set for all masters to the code, data, and peripherals memory regions on a domain level. The table is read-only.

NXP Semiconductors

IMXUG

User Guide for Config Tools for i.MX

M7 Domain M4 Domain					
Domain 0					
✓ Filter access for Master: C	AAM		• (ì	🗹 security state 🗹 executio	on privilege
 ✓ Show details ✓ Show Flash ✓ Show SRAM 	She	ow Periph	ierals 🔽	Show External RAM 🗹 Sho	ow External devices 🗹 This Domain Only
	RDC	MDAC	XRDC2	Resulting access level	
SEMC	RW	S-Priv		No access	0xDFFF_FFFF
FlexSP12	RW	S-Priv		No access	0×7FFF_FFFF
FlexSPI1	RW	S-Priv		No access	0x3FFF_FFFF 0x3000_0000
OCRAM + ECC (FlexRAM)	RW	S-Priv		No access	0x203F_FFFF
OCRAM2 ECC	RW	S-Priv	-	No access	0x2035_0000 0x2035_FFFF 0x2035_0000
OCRAM1 ECC	RW	S-Priv		No access	0x2034_FFFF

Figure 149. Memory Attribution Map

To set the display options, do the following:

- 1. Click the Filter access for checkbox to enable filtering options.
- 2. Select the master that you want to review by choosing from the Master dropdown menu.
- 3. Optionally, set the security state and execution privilege check-boxes when master allows more security levels. This setting has no effect on the configuration.
- 4. Optionally, customize the output by de-selecting the Show Details, Show Flash, Show SRAM, Show Peripherals, Show External RAM, Show External Devices and This Domain Only options.
- 5. Optionally, filter displayed memory regions in the Filter area.

Point your cursor over the cells to display a tooltip with information about the security level combination.

Double-click the cell to open the pertinent settings in **Security Access Configuration**.

5.2.4 Access Overview

In **Access Overview**, you can review security policies you have set in **Security Access Configuration** view. The view is divided into subviews displaying access overview for specific XRDC2 domains.

The vertical axis displays all masters, divided into color-coded groups by their security settings.

The horizontal axis displays memory ranges and slave buses/peripherals.

Memory Attribution Map 🦻 Access (Overv	iew		5 D	omai	ns Ov	/ervie	w												• -	' [
7 Domain M4 Domain																					
omain 0 Domain 1 Domain 2																					
	NS	-Usei	•				NS-	Priv				S-L	lser						S-P	riv	
	~	~	~				~	~				~	~	~				-	~	~	
	CDIFv2	AHB	AHB	AXI			AHB	AHB	AXI			CDIFv2	AHB	AHB	AXI			USDHC1	AHB	AHB	in v
Slave	5	M7	M7	M7	рхр	USB	M7	M7	M7	РХР	USB	5	M7	M7	M7	рХР	USB	ISL	M7	M7	1
Memory	-	~	~	~		-	-	~	-	-	2	-	-	-	~	-	-	2	~	~	
▼ ROMCP																					-
0x0020_0000 - 0x0023_FFFF	n/a	RW	RW	n/a	n/a	RW	RW	RW	n/a	n/a	RW	n/a	RW	RW	n/a	n/a	RW	n/a	RW	RW	n
• OCRAM M4	, u			.,, 🛥	, a				, a	.,, 4		, .			, a	, a		, a			
0x2020_0000 - 0x2023_FFFF	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n
▼ FlexSPI1											,			,							
0x3000_0000 - 0x30FF_FFFF	n/a	R-	R-	n/a	n/a	n/a	R-	R-	n/a	n/a	n/a	n/a	RW	RW	n/a	n/a	n/a	n/a	RW	RW	n
SIM_DISP	_									_					_						
▶ SIM_M7																					
▼ FlexSPI2																					
0x6000_0000 - 0x61FF_FFF	n/a			n/a	n/a	n/a			n/a	n/a	n/a	n/a			n/a	n/a	n/a	n/a			n
- SEMC																					
0x8000_0000 - 0x83FF_FFF	n/a	RW	RW	n/a	n/a	RW	RW	RW	n/a	n/a	RW	n/a	RW	RW	n/a	n/a	RW	n/a	RW	RW	n
0x8400_0000 - 0x85FF_FFFF	n/a			n/a	n/a				n/a	n/a		n/a			n/a	n/a		n/a			n
Peripherals																					
ACMP1	n/a	R-	R-	n/a	n/a	R-	R-	R-	n/a	n/a	R-	n/a	RW	RW	n/a	n/a	RW	n/a	RW	RW	n
ACMP2	n/a	R-	R-	n/a	n/a	R-	R-	R-	n/a	n/a	R-	n/a	RW	RW	n/a	n/a	RW	n/a	RW	RW	n
EMVSIM1	n/a			n/a	n/a				n/a	n/a		n/a			n/a	n/a		n/a			n
EMVSIM2	n/a			n/a	n/a				n/a	n/a		n/a			n/a	n/a		n/a			n
ENET	n/a			n/a	n/a				n/a	n/a		n/a			n/a	n/a		n/a			n
ENET_1G	n/a			n/a	n/a				n/a	n/a		n/a			n/a	n/a		n/a			n
FLEXSPI1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n
FLEXSPI2	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n
GC335 (GPU2D)	n/a			n/a	n/a				n/a	n/a		n/a			n/a	n/a		<u> </u>	RW	RW	n
GPIO1	n/a		R-	n/a	n/a		R-	R-	n/a	n/a		n/a	RW	RW	n/a	n/a		n/a		RW	
IOMUXC	n/a	RW	RW	n/a	n/a	RW	RW	RW	n/a	n/a	RW	n/a	RW	RW	n/a	n/a	RW	n/a	RW	RW	n
LCDIF	n/a	RW	RW	n/a	n/a	RW	RW	RW	n/a	n/a	RW	n/a	RW	RW	n/a	n/a	RW	n/a		RW	n
LCDIFV2	n/a	RW	RW	n/a	n/a	RW	RW	RW	n/a	n/a	RW	n/a	RW	RW	n/a	n/a	RW		RW	RW	n
LPUART1	n/a			n/a	n/a				n/a	n/a		n/a			n/a	n/a		n/a			n
MIPI_CSI	n/a			n/a	n/a				n/a	n/a		n/a			n/a	n/a		n/a			n
MU(A)	n/a			n/a		n/a		n/a			n/a				n/a	n/a			n/a		n
USDHC1	n/a	RW	RW	n/a	n/a	RW	RW	RW	n/a	n/a	RW	n/a	RW	RW	n/a	n/a	RW	n/a	RW	RW	n,
USDHC2	n/a			n/a	n/a				n/a	n/a		n/a	R-	R-	n/a	n/a	R-	n/a	R-	R-	n,

Point your cursor at an entry to display a tooltip with information about the entry.

You can group the displayed information by security or by masters by using the button on the right-hand side of the toolbar.

5.2.5 Domains Overview

In **Domains Overview**, you can review access policies of XRDC2 domains you have configured in the subviews of the **Domain** view.

Point your cursor over the cells to display a tooltip with information about the security level combination.

	M7 Domair	า		M4 Domair	ı
urce	Domain 0	Domain	1 Domain 2	Domain 0	Domain 1
sters		- <u> </u>			
CAAM		 Image: A second s			
CM4 AHBC				 ✓ 	
CM4 AHBS				 Image: A set of the set of the	
CM4 DMA				~	
CM7 AHB	~				
CM7 AXI	~				
CM7 DMA	~				
CSI					
ENET					
ENET QOS					
ENET_1G RX					\checkmark
ENET_1G TX					~
GC355 (GPU2D)			✓		
LCDIF1 (eLCDIF)			✓		
LCDIF2 (LCDIFv2)	 Image: A set of the set of the				
PXP	~				
USB	~				
USDHC1	✓				
USDHC2				✓	
mory					
ITCM (FlexRAM)					
ROMCP					
0x0020_0000 - 0x0023_FFFF	RW	RW			
CAAM Secure RAM					
0x0028_0000 - 0x0028_FFFF					
ІТСМ М4					
DTCM M4					
DTCM (FlexRAM)					
OCRAM M4					
0x2020_0000 - 0x2021_FFFF	R-			RW	RW
0x2022_0000 - 0x2023_FFFF	R-	R-		R-	
OCRAM1					
0x2024_0000 - 0x202B_FFFF	RW	RW	RW	R-	R-
OCRAM2					
0x202C_0000 - 0x2033_FFFF	RW	RW	RW	R-	R-
OCRAM1 ECC					
OCRAM2 ECC					
OCRAM M7					
FlexSPI1					
0x3000_0000 - 0x30FF_FFFF	RW	RW		R-	R-
0x3100_0000 - 0x3FFF_FFF					

IMXUG

User guide

© 2023 NXP B.V. All rights reserved.

5.2.6 Code generation

If the settings are correct and no error is reported, the code generation engine regenerates the source code. You can view the resulting code the **Code Preview** view of the **Trusted Execution Environment** tool.

Code Preview automatically highlights differences between the current and immediately preceding iteration of the code. You can choose between two modes of highlighting by clicking the **Set viewing style for source differences**. You can also disable highlighting altogether from the same dropdown menu. Such features as Copy, Search, Zoom-in, Zoom-out, and Export source are available in the **Code Preview** view. The search can also be invoked by CTRL+F or from the context menu.

Some AHBSC or TRDC with security extension-enabled devices support ROM preset as well as C code. You can choose to have the code generated in the ROM preset by selecting the option in the **Miscellaneous** subview.

6 Advanced Features

6.1 Switching the processor

You can switch the processor or the package of the current configuration to a different one. However, switching to a completely different processor may lead to various issues, such as inaccessible pin routing or unsatisfiable clock-output frequency. In that case, it's necessary to fix the problem manually. For example, go to the **Routing Details** view and reconfigure all pins which report an error or conflict. Alternatively, you may need to change the required frequencies on clock output.

To change the processor in the selected configuration, select **File > Switch processor** from the **Menu bar**.

User Guide for Config Tools for i.MX

Switch processor					-	_		\times
Switch processor								
Select Processor								
type filter text								
Select Processor/Board/Kit	Pins	DDR	Status					
✓ Processors								
✓ i.MX 6Quad								
MCIMX6QxxVT	 Image: A second s	×	Cached					
✓ i.MX 8M Mini Dual								
MIMX8MM4xxxKZ	×	 Image: A second s	Cached					
✓ i.MX RT								
MIMXRT1052xxxxB	 Image: A second s	×	Cached					
Name your configuration								
Select processor package	Select	core			Select SDK version			
×	Corte	x-A53((core#0)	\sim				\sim
					Finish		Cancel	
Figure 152. Switch processor								

To change the package of the currently selected processor, select File > Switch processor from the Menu bar.

User Guide for Config Tools for i.MX

(Switch package for the Processor
	Available Processor Packages
	MCIMX6Q6AVT08AD - FCPBGA 624 package MCIMX6Q4AVT08AD - FCPBGA 624 package MCIMX6Q6AVT10AD - FCPBGA 624 package MCIMX6Q4AVT08AC - FCPBGA 624 package MCIMX6Q4AVT10AD - FCPBGA 624 package MCIMX6Q7CVT08AC - FCPBGA 624 package MCIMX6Q6AVT08AC - FCPBGA 624 package MCIMX6Q6AVT08AC - FCPBGA 624 package
	MCIMX6Q7CVT08AD - FCPBGA 624 package OK Cancel
Figure 153. Switch pac	kage

6.2 Exporting the Pins table

To export the Pins table, do the following:

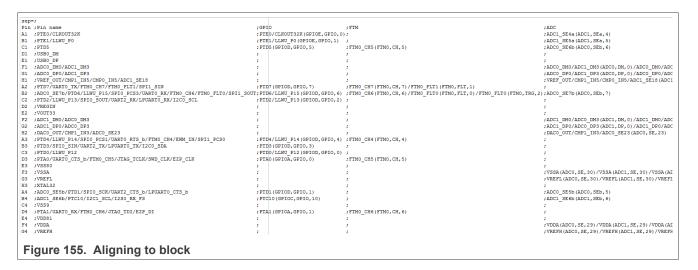
- 1. In the **Menu bar**, select **File > Export**.
- 2. In the Export wizard, select Export the Pins in CSV (Comma Separated Values) Format.
- 3. Click Next.
- 4. Select the folder and specify the filename to which you want to export.
- 5. The exported file contains content of the Pins view table, and lists the functions and the selected routed pins.

sep=;
Pin; Pin name; GPIO; FTM; ADC; UART; SPI; I2S; LLWU; I2C; CMP; SUPPLY; LPUART; USB; SIM; JTAG; RIC; EWM; Other; Routing for BOARD_InitPins
A1; PTE0/CLKOUT32K; PTE0/CLKOUT32K (GPIOE, GPIO, 0); ; ADC1_SE4a (ADC1, SEa, 4); UART1_TX (UART1_TX); SPI1_PCS1 (SPI1, PCS1);; ; I2C1_SDA (I2C1, SDA);;;; PTE0
B1; PTE1/LLWU_P0; PTE1/LLWU_P0 (GPIOE, GPIO, 1); ; ADC1_SE5a (ADC1, SEa, 5); UART1_RX (UART1, RX); SPI1_SOUT (SPI1, SOUT) / SPI1_SIN (SPI1, SIN); ; PTE1/LLWU_P0 (
C1: PTD5: PTD5 (GPIOD, GPIO, 5); FTM0_CH5 (FTM0, CH, 5); ADC0_SE6b (ADC0, SEb, 6); UART0_CT5_b (UART0, CT5); SPI0_PCS2 (SPI0, PCS2) / SPI1_SCK (SPI1, SCK); ;;;;;;;
D1;USB0_DM;;;;;;;;;USB0_DM(USB0,DM);;;;;;
E1;USB0_DP;;;;;;;;USB0_DP(USB0,DP);;;;;
F1;ADC0_DM0/ADC1_DM3;;;ADC0_DM0/ADC1_DM3 (ADC0, DM, 0) /ADC0_DM0/ADC1_DM3 (ADC0, SE, 19) /ADC0_DM0/ADC1_DM3 (ADC1, DM, 3) ;;;;;;;;;ADC0_DM0/ADC1_
G1; ADC0_DF0/ADC1_DF3; ; ; ADC0_DF0/ADC1_DF3 (ADC0, DF, 0) / ADC0_DF0/ADC1_DF3 (ADC0, SE, 0) / ADC0_DF0/ADC1_DF3 (ADC1, DF, 3) / ADC0_DF0/ADC1_DF3 (ADC1, SE, 3) ;
H1; VREF_OUT/CMP1_IN5/CMP0_IN5/ADC1_SE18;;; VREF_OUT/CMP1_IN5/CMP0_IN5/ADC1_SE18 (ADC1, SE, 18);;;;; VREF_OUT/CMP1_IN5/CMP0_IN5/ADC1_SE18 (CMP1, I
A2; PTD7/UART0_TX/FTM0_CH7/FTM0_FLT1/SPI1_SIN; PTD7 (GPIOD, GPIO, 7); FTM0_CH7 (FTM0, CH, 7) / FTM0_FLT1 (FTM0, FLT, 1); ; UART0_TX (UART0, TX); SPI1_SIN (SPI1
B2;ADC0_SE7b/PTD6/LLWU_P15/SP10_PCS3/UART0_RX/FTM0_CH6/FTM0_FLT0/SP11_SOUT;PTD6/LLWU_P15(GP100,GP10,6);FTM0_CH6(FTM0,CH,6)/FTM0_FLT0(FTM0,F
C2; PTD2/LLWU_P13/SPI0_SOUT/UART2_RX/LPUART0_RX/I2C0_SCL; PTD2/LLWU_P13 (GPI0D, GPI0, 2); ;; UART2_RX (UART2, RX); SPI0_SOUT (SPI0, SOUT); ; PTD2/LLWU_P1
D2;VREGIN;;;;;;;VREGIN(USB0,VREGIN);;;;;;
E2;VOUT33;;;;;;;;VOUT33(USB0,VOUT33);;;;;;
F2;ADC1_DM0/ADC0_DM3;;;ADC1_DM0/ADC0_DM3(ADC1,DM,0)/ADC1_DM0/ADC0_DM3(ADC1,SE,19)/ADC1_DM0/ADC0_DM3(ADC0,DM,3);;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
G2;ADC1_DP0/ADC0_DP3;;;ADC1_DP0/ADC0_DP3(ADC1,DP,0)/ADC1_DP0/ADC0_DP3(ADC1,SE,0)/ADC1_DP0/ADC0_DP3(ADC0,DP,3)/ADC1_DP0/ADC0_DP3(ADC0,SE,3);
H2;DAC0_OUT/CMP1_IN3/ADC0_SE23;;;DAC0_OUT/CMP1_IN3/ADC0_SE23(ADC0,SE,23);;;;;DAC0_OUT/CMP1_IN3/ADC0_SE23(CMP1,IN,3);;;;;;;DAC0_OUT/CMP1_I
A3; PTD4/LLWU_P14/SPI0_PCS1/UART0_RTS_b/FTM0_CH4/EWM_IN/SPI1_PCS0; PTD4/LLWU_P14 (GPI0D, GPI0, 4); FTM0_CH4 (FTM0, CH, 4); ; UART0_RTS_b (UART0, RTS); SP
B3; PTD3/SPI0_SIN/UART2_TX/LPUART0_TX/I2C0_SDA; PTD3 (GPI0D, GPI0, 3); ;; UART2_TX (UART2, TX); SPI0_SIN (SPI0, SIN); ;; I2C0_SDA (I2C0, SDA); ;; LPUART0_TX (
C3;PTD0/LLWU_P12;PTD0/LLWU_P12(GPIOD,GPIO,0);;;UART2_RTS_b(UART2,RTS);SPI0_PCS0(SPI0,PCS0/SS);;PTD0/LLWU_P12(LLWU,WAKEUP,P12);;;;LPUART0_RT
D3; FTAO/UARTO_CTS_b/FTMO_CH5/JTAG_TCLK/SWD_CLK/EZP_CLK; FTAO(GPIOA, GPIO, 0); FTMO_CH5(FTMO, CH, 5); ; UARTO_CTS_b(UARTO, CTS); ; ; ; ; ; ; ; ; JTAG_TCLK(JT
Figure 154. Exported file content
rigure 194. Exported me content

IMXUG

© 2023 NXP B.V. All rights reserved.

The exported content can be used in other tools for further processing. For example, see it after aligning to blocks in the image below.



6.3 Tools advanced configuration

Use the tools.ini file to configure the processor data directory location. You can define the "com.nxp.mcudata.dir" property to set the data directory location.

For example: -Dcom.nxp.mcudata.dir=C:/my/data/directory.

6.4 Generating HTML reports

You can generate an HTML report file displaying your configuration of Pins, Clocks, and Peripheral tool for future reference.

To generate the HTML report, select **Export > Pins > Export HTML Report**.

6.5 Exporting sources

It's possible to export the generated source using the Export wizard.

To launch the Export wizard:

- 1. Select **File > Export** from the **Menu bar**.
- 2. Select Export Source Files.

User Guide for Config Tools for i.MX

☑ Export	×
Select Export Source Files	2
Select an export wizard: type filter text PBL Configuration Tool Pins Tool Export External User Signals Export HTML Report Export Registers Export Source Files Export the Pins in CSV (Comma Separated Values) Format Processor Data Processor Data Processor Data Processor Data Processor Data FIE Tool Fools Configuration Export Configuration as Template Export Output path overrides Cack Next > Finish Ca	ncel
Figure 156. Export wizard	

3. Click Next.

4. Select the target folder where you want to store the generated files.

User Guide for Config Tools for i.MX

	🖉 Export			-			×		
	Export Pins Source								
	 File(s) to export: pin_mux.h, pin_mux.c 								
	Cortex-M7F								
	C:\nxp\Project				~	Browse.			
		< Back	Next >	Finish		Cancel	l -		
Figure 157.	Export sources – Se	lect target folder							

5. In case of multicore processors, select the cores you want to export.

6. Click Finish.

6.6 Exporting registers

You can export the content of tool-modified registers data using the Export wizard.

To export registers, follow these steps:

- 1. Select **File > Export** from the main menu.
- 2. Select the **Pins Tool > Export Registers** option.

Export	
Select Export Registers	
Select an export wizard:	
type filter text	
 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	
< Back Next > Finish	Cancel
Figure 158. Export registers	

3. Click Next.

4. Select the target file path where you want to export modified registers content.

User Guide for Config Tools for i.MX

Ø	Export			_		×
Ex	xport Registers					
()) File(s) to export: e					
То	o directory: C:\nx	p			Bro	wse
		< Back	Next >	Finish	Cance	el
Figure 159. Exp	port registers dire	ctory				

5. Click Finish.

Note: Export wizard can also be opened by clicking the Export registers to CSV button in the Registers view.

	Overview 🔓 Code Preview 👪 Registers 🗙	🏠 Overview 🚺 Code Preview 🚺 Registers 🗙			
	all Show the modified reg	jisters only.		4	
	type filter text				
	Reg. Name	Set Value	Reset Value	٧	
	> IOMUXC_LPSR_GPR_GPR2	0x00000000	0x0000000	G	
	> IOMUXC_LPSR_GPR_GPR3	0x0000000	0x0000000	G	
Figure 160.	Export registers to CSV				

124 / 141

6.7 Managing data and working offline

With the **Data Manager**, you can download, import, and export processor data. This feature is especially useful if you want to make the best out of the tools while staying offline.

🝘 Data Manager										×
Select Processor/Board/Kit	Pins	DDR	Status	Local Version	Remote Version					
✓ □ Boards				-						
IMXRT1050-EVKB	 Image: A set of the set of the	×	Cached	0.0.1	N/A					
MCIMX6Q-SDB-REV-B	~	×	Cached	0.0.1	N/A					_
✓ □ Processors				-						
✓ ☐ i.MX 6Quad				-						_
MCIMX6QxxVT	~	×	Cached	0.0.1	N/A					
✓ ☐ i.MX 8M Mini Dual				-						_
MIMX8MM4xxxKZ	×	~	Cached	0.0.1	N/A					_
V 🗌 i.MX RT				-						_
MIMXRT1052xxxxB	~	×	Cached	0.0.1	N/A					_
										_
										_
										_
										_
										_
Import Export	(Cached	only	Select outdated	Unselect a	Update / Download	Clean cached	(Close	
Figure 161. Data Manager										

6.7.1 Working offline

You can create a new configuration even without access to the Internet by working with cached processor data. To do so you must download processor-specific data before going offline, or import data downloaded and exported from an online computer.

To work offline, select **Edit > Preferences > Work offline** from the **Menu bar**.

6.7.2 Downloading data

You can download required processor data with Data Manager.

Note: By default, the data is downloaded and cached automatically during the **Creating a new standalone** configuration for processor, board, or kit process.

To download processor data, do the following:

Note: Internet connection is required for data download.

- 1. In Menu bar, select File>Data Manager.
- 2. In Data Manager, select the processor/board/kit you want to work with from the list.
- Click Update / Download and confirm.
 The data is now downloaded on your local computer, as shown by the Cached status in Data Manager.

You can now close your Internet connection and work with the data by selecting **File > New...>Create new** standalone configuration for processor, board, or kit in the Start development wizard.

6.7.3 Exporting data

With Data Manager, you can export downloaded processor data in a ZIP format.

User Guide for Config Tools for i.MX

To export data, do the following:

- 1. In Menu bar, select File>Data Manager.
- 2. In Data Manager, click Export.
- 3. In Export Processor Data window, select the processor data you want to export.
- 4. Click Browse to specify the location and name of the resulting ZIP file.
- 5. Click Finish,

Data is now saved on your local computer in a ZIP format. You can physically (for example, with a USB stick) move it to an offline computer.

Note: You can also export downloaded data by selecting **File** > **Export** > **Processor Data** > **Export Processor Data** from the **Menu bar**.

6.7.4 Importing data

You can import processor data from another computer with **Data Manager**, provided this data is available locally.

To import data, do the following:

- 1. In Menu bar, select File>Data Manager.
- 2. In Data Manager, select Import.
- 3. In Import Processor Data dialog, click Browse.
- 4. Specify the location of the ZIP file that you want to import and click **OK**.
- 5. Choose the data to import by selecting the checkbox in the table.
- 6. Click Finish.

The data is now imported to your offline computer, as shown by the **Cached** status in **Data Manager**. You can now work with the data by selecting **New...>Create new standalone configuration for processor**, **board**, **or kit** in the **Start development** wizard.

Note: You can also import data by selecting File>Import>Import Processor Data from the Menu bar.

6.7.5 Updating data

You can keep cached data up to date with the Data Manager.

Note: If you select the relevant option in **Edit>Preferences** in the **Menu bar**, data will be updated automatically or after a prompt.

Note: Internet connection is required for data update.

To update cached data, do the following:

- 1. In Menu bar, select File > Data Manager.
- 2. In Data Manager, filter outdated data by clicking Select outdated.
- Click Update / Download and confirm. You can always check versions of your data by clicking Cached only and comparing version information in the Local Version and Remote Version columns. You can clean all cached data by selecting Clean cached. It removes all processor, board, kit, and component data, as well as SDK info files from your computer. Note: This action does not affect user templates.

6.8 Output path overrides

This section contains rules that override the path, including the name, of the output files generated by the tools. The rules are applied in the Update Code, Export Wizard, and Command-Line Export commands. The rules are stored in the MEX configuration.

User Guide for Config Tools for i.MX

Note: An invalid path is logged as a warning and the original non-overridden path is used.

Rules can be edited in the Output Path Override dialog box in the configuration settings. The new rule is added to the end of the list, the removal is performed for the selected element. The rules are applied to the path in a defined order, which can be changed. The rule contains:

- Enabled defines whether the rule will be used by the applied path or skipped.
- · Description used as a user-friendly description of the rule
- Regular expression matches the overriding parts in the whole output path. The format is taken from the Java regular expression.
- Replacement expression used as a replacement of all matches in the path. Substring groups can be referenced by using placeholder \$1, \$2 and so on.

The output path override rules can be exported using the wizard to a yaml file. The structure of the yaml file is similar to that of the dialog box.

Example content of the output path override yaml file:

```
outputPathOverrides:
-description: Rule group.h
enabled:true
regex:(bo)ar(d)(/.*\.h)
replacement: $2ar$1$3
-description:Rule2
...
```

The second way to set the rules is to replace them by overriding the output path from the yaml file using wizards or the command line. Rules are used only if all rules are valid. An empty list deletes the current rules. An empty list in the output path overrides the yaml file.

```
outputPathOverrides: [
]
```

6.9 Import pins configuration from legacy tools project

The Pins tool from **Config Tools for i.MX** helps in importing pins configuration from the existing legacy tools projects with the following prerequisites.

- Before importing any pins configuration from the legacy tools project, download the required i.MX processor pins tool data to a local directory.
- Before importing, create a new configuration for the respective i.MX processor of the given package variant, to minimize the need of manual correction of the imported pins configuration.

6.9.1 Importing from an IO Mux Tool design configuration file

You can import an existing pin configuration from an IO Mux Tool Design Configuration project file (XML) that was used to keep pin routing configuration within a legacy IO Mux Tool. The import wizard is used to import pin routing configuration from existing project XML file compatible with IO Mux Tool version v3.4.0.3.

To import from an IO Mux Tool design configuration file, do the following:

- 1. Select File > Import from the Main Menu.
- 2. Select the Import Legacy IOMux Tool Design Configuration (XML) Format option.
- 3. Click Next.
- 4. Select the source file (XML) to import by using the **Browse** button in the **Import** dialog.

😰 Import		- • •			
Import from ION	Iux Tool Design Configuration XML File				
Source file (.xml):	C:\Freescale\iomux_tool\projects\i.MX6DQ_EVB_X3.IoMuxDesign.xml	Browse			
Figure 162. Import from IOMux Tool Design Configuration XML File					

5. Click Finish.

The selected source file is processed and the existing pin configuration for peripheral routing is imported for each peripheral to the Pins tool. A new function is created for each peripheral instance with all configured pins using the function name "configure_<peripheral name>_pins" and added into the **Routed Pins** view.

6.9.2 Importing from a Processor Expert project

You can import the existing pin configuration from a Processor Expert (PEx) project file (PE). The PE file is the main project file for legacy i.MX pin mux component and is available within the Processor Expert for i.MX tool.

The **Import** wizard is used to import legacy i.MX pin configuration from the existing PEx project file.

To import from a Processor Expert project, do the following:

- 1. Select **File > Import** from the main menu.
- 2. Select the Import Legacy i.MX Pins Configuration (PEx for i.MX) Format option.
- 3. Click Next.
- 4. Select the source file (.pe) to import using the **Browse** button in the **Import Pins Configuration from Processor Expert Project** dialog.

😰 Import		
Import Pins Co	nfiguration from Processor Expert Project	
Source file (.pe):	C:\nxp\Configurations\ProcessorExpert.pe	Browse
Figure 163. Impor	t Pins Configuration from Processor Expert Project	

5. Click **Finish**.

The selected source file is processed and the existing pin configuration for peripheral routing is imported for each peripheral to the Pins Tool. A new function is created for each peripheral instance with all configured pins using the function name "configure_<peripheral name>_pins" and is added into the routed pins.

6.10 Command-line execution

This section describes the Command Line Interface (CLI) commands supported by the desktop application.

On error application exits:

- Tools v4.1 and older:
- With '123321' error code. The reason should be logged.
- Tools v5.0 and newer:
 - when a parameter is missing
 - when a tool error occurs

You can chain commands in CLI.

Notes regarding command-line execution:

- Command -HeadlessTool is used as a separator of each command chain.
- · Each command chain works independently.
- Every chain starts with **-HeadlessTool** command and continues to the next **-HeadlessTool** command, or end. (only exception are commands from framework which does not need the **-HeadlessTool** command).
- Commands which don't need the **-HeadlessTool** command, can be placed before the first **-HeadlessTool** if chained, or without **-HeadlessTool** when not chained.
- · Commands from each tool are executed in given order.
- Commands from framework are not executed in given order.
- The following commands are not executed in given order:
 - ImportProject
 - Export MEX
 - ExportAll
- The application can exit with following codes when unexpected behavior occurs:
 - When a parameter is missing: 1
 - When a tool error occurs: 2

Command example:

```
-HeadlessTool Clocks -MCU MK64FX512xxx12 -SDKVersion ksdk2_0 -ExportSrc C:/
exports/src -HeadlessTool Pins -MCU MK64FX512xxx12 -SDKVersion ksdk2_0 -ExportSrc
C:/exports/src -HeadlessTool Peripherals -MCU MK64FX512xxx12 -SDKVersion ksdk2_0 -
ExportSrc C:/exports/src
```

For performance reasons, when CLI is expected to be used multiple times with the same processor, the data is only loaded **if it is not already on disk**. If there is newer data on the server, it is **not updated**.

Long-running jobs share data, so they do not get updated in the middle of execution. To update local data that may have a newer version on the server, use the -updateData parameter.

Recommended usage:

- For manual one-time usage, include the -updateData parameter on the CLI.
- For multiple executions, for example, continuous integration set-up you job:
- Use the first simple command with -updateData, which updates possibly outdated data.
- Use all other commands in a batch run without this parameter:

```
copy /Y eclipsec.exe toolsc.exe
@rem updates all local data if newer exists
tools.exe -updateData -consoleLog -HeadlessTool Pins
@rem now runs tools many times
tools.exe -consoleLog -HeadlessTool Pins -Load some.mex -ExportAll c:/directory
tools.exe -consoleLog -HeadlessTool Pins -Load other.mex -ExportAll c:/
other_directory
@rem and so on.
```

The following commands are supported in the **framework**:

Command name	Definition and parameters	Description	Restriction	Example
Version of the product	-version	Shows the build version of the product into the stdout and continues with parsing other parameters. (since 6.0)		-version
Force language	-nl {lang}	Forces set language {lang} is in <u>ISO-639-1</u> standard	Removal of the '.nxp' folder from home directory is recommended, as some text might be cached Only 'zh' and 'en' are	-nl zh
			supported	
Show console	-consoleLog	Logs output is also sent to Java's System.out (typically back to the command shell if any)	None	
Empty configuration	-EmptyConfig	Ensures creating a new empty configuration without any content	Requires the -Headless Tool command	-HeadlessTool Pins -EmptyConfig
Select MCU	-MCU	MCU to be selected by framework Changes the processor in the result configuration of the previous chain	Requires the -Headless Tool and -SDKversion commands.	-MCU MK64 FX512xxx12 -SDKversion ksdk2_0 -HeadlessTool Pins -Load C:/conf/ conf.mex -SDKVersion ksdk2_0 -MCU MK64FN1 M0xxx12 -ExportMEX C:/ conf/MK64.mex
Select core	-Core	Select core for the configuration		-Core core0
Select Board	-Board	Board to be selected by framework (MCU is automatically selected too)(since 6.0)	Requires the – SDKversion command	-Board FRDM-K22F -SDKversion ksdk2_0
Select Kit	-Kit	Kit to be selected by framework (MCU and board is automatically	Requires the – SDKversion command	-Kit FRDM-K22F- AGM01
		selected too)(since 6.0)		-SDKversion ksdk2_0
Select SDK version	-SDKversion	Version of the MCU to be selected by framework	Requires the -MCU command	-MCU MK64 FX512xxx12
				-SDKversion ksdk2_0
Select part number	-PartNum	Selects specific package of the MCU	Requires the -MCU and - SDKversion commands	-MCU MK64 FX512xxx12
		Changes package in result configuration of		-SDKversion ksdk2_0 -PartNum MK64 FX512VLL12

Table 18. Commands supported in the framework

IMXUG

© 2023 NXP B.V. All rights reserved.

Command name	Definition and parameters	Description	Restriction	Example
		previous chain, see the command about "-MCU"		
Configuration name	-ConfigName	Name of newly created configuration - used in export Rename the configuration	Requires the -Headless Tool command	-MCU MK64 FX512xxx12 - SDKversion ksdk2_0 -ConfigName "My Config" -ConfigName "My RenameConfig"
Select tool	-HeadlessTool	Selects a tool that must be run in headless mode	If the tool is disabled in the configuration, it will not be enabled by this option. Use -Enable if the tool must be enabled via cmd line.	-HeadlessTool
Load configuration	-Load	Loads the existing configuration from a (*.mex) file	None	-Load C:/conf/conf. mex
Export Mex	-ExportMEX	Exports the.mex configuration file after tools run The argument is expected to be a folder or path to specify the .mex file	Requires the -Headless Tool command	-MCU xxx - SDKversion xxx -ExportMEX C:/ exports/my_ config_folder -ExportMEX C:/ exports/my_ config_folder/ my.mex
Export all generated files	-ExportAll	Exports all generated files (with source code, and so on). Code is regenerated before export Includes -ExportSrc and in framework - ExportMEX The Argument is expected to be a folder name.	Requires the -Headless Tool command	-HeadlessTool Pins -ExportAll C:/ exports/ generated
Create a new configuration by importing a toolchain project	- ImportProject {path}	Creates a new configuration by importing toolchain project The parameter is a path to the root of the toolchain project.	Requires the -Headless Tool command	-HeadlessTool Pins -ImportProject c: \test\myproject
Create a new configuration from a toolchain project	-CreateFrom Project {path}	Creates a new configuration (memory only) by importing a toolchain project based on the found .mex or YAML info. It does not do any changes on the disk	Requires the -Headless Tool command see - ImportProject	-HeadlessTool Prj Cloner -CreateFrom Project c:\test \myproject -HeadlessTool Peripherals -Create

Table 18. Commands supported in the framework...continued

Command name	Definition and parameters	Description	Restriction	Example
		(mex and update code), validates configuration. The parameter is a path to the root of the toolchain project.		FromProject c:\test \myprojectsee -ImportProject
Generate source files with custom copyright	-Custom Copyright	File content is inserted as a copyright file header comment into generated source files (.c, .h, .dts, .dtsi), that does not contain copyright	Requires the -Headless Tool command	-CustomCopyright c: \test\copyright. txt
Override the output path of the generated files	-OutputPath Overrides	Path to the file with rules that will be used to override output paths of the generated file. Empty list of rules removes the set rules.	Requires the -Headless Tool command	-OutputPathOverrides c:\test\output PathOverride Rules.yaml
Batch processing	-BatchFile	Path to the file with commands, that will be run on defined paths. For details, see <u>Section 6.11</u>	Requires the -Headless Tool command; intent without other commands	-HeadlessTool Pins -BatchFile C:/conf/ batchCommands. yaml
Project link	-ProjectLink	A custom path to the toolchain project folder. It can be used as the absolute path or relative against the saved configuration. Empty sting for default that is not saved in the configuration (since v14).	When the command is not used along with the -HeadlessTool command, the -Load command is required.	-HeadlessTool Pins -ProjectLink C: / project -HeadlessTool Pins -ProjectLink armgcc -HeadlessTool Pins -ProjectLink ""
Update locally downloaded data	-updateData	Downloads data for already locally downloaded data if they have an update.		-updateData

Table 18. Commands supported in the framework...continued

6.10.1 Command-Line execution - Pins Tool

This section describes the Command Line Interface (CLI) commands supported in the Pins Tool.

Command name	Definition and parameters	Description	Restriction	Example
Enable tool	-Enable	Enables the tool if it is disabled in the current configuration	Requires -HeadlessTool Pins	-HeadlessTool Pins - Enable
Import C files	-ImportC	Imports .c files into configuration	Requires -HeadlessTool Pins	-HeadlessTool Pins

Table 19. Commands supported in Pins

Command name	Definition and parameters	Description	Restriction	Example
		Importing is done after loading mex and before generating outputs		-ImportC C:/ imports/file1. c C:/imports/ file2.c
Import DTSI files	-ImportDTSI	Imports .dtsi files into configuration	Requires -HeadlessTool Pins Only for processors with support of the feature	-HeadlessTool Pins -ImportDTSI C:/ imports/file1. dtsi C:/imports/ file2.dtsi
Export all generated files (to simplify all exports commands to one command)	-ExportAll	Exports generated files (with the source code, and so on) The code is regenerated before the export. Includes -ExportSrc,- ExportCSV, -ExportHTML and in framework -Export MEX The argument is expected to be a folder	Requires -HeadlessTool Pins	-HeadlessTool Pins -ExportAll C:/ exports/ generated
Export Source files	-ExportSrc	Exports generated source files. The code is regenerated before the export. The argument is expected to be a folder.	Requires -HeadlessTool Pins	-HeadlessTool Pins -ExportSrc C:/ exports/src
Export CSV file	-ExportCSV	Exports the generated .csv file. The code is regenerated before export. The argument is expected to be a folder.	Requires -HeadlessTool Pins	-HeadlessTool Pins -ExportSrc C:/ exports/src
Export HTML report file	-ExportHTML	Exports the generated HTML report file. The code is regenerated before the export. The argument is expected to be a folder.	Requires -HeadlessTool Pins	-HeadlessTool Pins -ExportHTML C:/ exports/html
Export registers	-Export Registers	Exports the registers tab into a folder. The code is regenerated before the export. The argument is expected to be a folder.	Requires -HeadlessTool Pins	-HeadlessTool Pins -ExportRegisters C:/ exports/regs

Table 19. Commands supported in Pins...continued

6.10.2 Command-Line execution - TEE Tool

This section describes the Command Line Interface (CLI) commands supported in the TEE Tool.

Command name	Definition and parameters	Description	Restriction	Example
Enable tool	-Enable	Enables the tool if it is disabled in the current configuration	Requires -Headless Tool TEE	-HeadlessTool TEE - Enable
Export all generated files (to simplify all exports commands to one command)	-ExportAll	Exports generated files (with source code and so on) The code is regenerated before the export. Includes -ExportSrc, -ExportHTML, and in framework -ExportMEX The argument is expected to be a folder.	Requires - HeadlessTool TEE	-HeadlessTool TEE -ExportAll C:/ exports/generated
Export Source files	-ExportSrc	Exports generated source files The code is regenerated before the export. The argument is expected to be a folder.	Requires - HeadlessTool TEE	-HeadlessTool TEE -ExportSrc C:/ exports/src
Export registers	-ExportRegisters	Exports the registers tab into a folder. The code is regenerated before the export. The argument is expected to be a folder.	Requires -Headless Tool TEE -Enable	-HeadlessTool TEE - Enable -Export Registers C: / exports/regs

Table 20. Commands supported in TEE Tool

6.11 Batch processing

Batch processing can be used to simplify and speed up processing of multiple configurations in an automated way using the command-line interface.

Batch processing is initiated by the headless command "-BatchFile". When the command is used, the tool operation is controlled by the specified batch file passed as an argument.

Example:

-HeadlessTool Pins -BatchFile C:/conf/batchCommands.yaml

Note: It is intended to be used without specifying other tools commands except "-HeadlessTool".

6.11.1 Content of the batch file

The batch file content is in YAML format and consists of the folders or files list and the list of command steps. All the steps are executed for each individual path in the given order. If the 'folders' entity is used, the 'files' cannot be used and vice versa.

Note: Paths can be defined as absolute or relative to the batch file.

6.11.1.1 Folders list

The entity "folders" contains a list of the folders to be processed.

Example of the folders list:

```
!!batch_process
folders:
- c:/ test/frdm64
- c:/_ddm/tmp/test/lpc69
steps:
- action: ImportC
tool: Pins
args: ${folder}/src/board/pin_mux.c
- action: ImportC
tool: Clocks
args: ${folder}/src/board/clock_config.c
- action: ExportAll
tool: Clocks
args: ${folder}/export/clocks
```

Note: The steps element description is given below.

6.11.1.2 Files list

The entity "files" specifies the list of the files to be processed.

Example of the list of files:

```
!!batch_process
files:
- c:/_ddm/tmp/test/frdm64/src/board/pin_mux.c
- c:/_ddm/tmp/test/lpc69/src/board/pin_mux.c
steps:
- action: ImportC
args: ${file}
tool: Pins
- action: ImportC
tool: Clocks
args: ${folder}/clock_config.c
```

6.11.1.3 Steps list

The steps entity contains a list of steps to be processed for every listed path in a similar way as standard headless command-line tool commands.

Each step is defined as a structure:

- Action the name of a command-line tool command without "-" at the beginning.
- Tool the name of the tool in the product that runs the action.
- Args arguments of the actions, a space between the parameters is expected.
 - The following variables can be used and will be substituted with the appropriate value. In case the "files" list
 is processed:
 - \${folder} replaced by folder (without separator at the end) where the file is located
 - \${file} the full file path

© 2023 NXP B.V. All rights reserved.

User Guide for Config Tools for i.MX

- \${fileName} for filename without path
- In case the "folders" list is processed:
- \${folder} the folder path without separator at the end

See the section <u>Section 6.10</u> for the list of commands and their description.

The configuration is always automatically cleaned after processing each path (as if the -EmptyConfig command was used). The batch file without any paths runs once and cannot use any variables.

Note: All paths on loading are converted to the path format with "/" as a separator.

7 Support

If you have any questions or need additional help, perform a search on the forum or post a new question. Visit <u>community.nxp.com/community/imx</u>.

8 Note about the source code in the document

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

Copyright 2024 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.

9 Revision history

Table 21. Revision history		
Document ID	Release date	Description
IMXUG v.7	10 January 2024	Updated for v.15
IMXUG v.6	31 July 2023	Section 6.11 is added.
IMXUG v.5	2 January 2023	Section 4 is updated.
IMXUG v.4	20 September 2022	Updated for v.12.1
IMXUG v.3	30 June 2022	Updated for v.12

Table 21. Revision mistoryconunded		
Document ID	Release date	Description
IMXUG v.2	22 December 2021	New features are added, screenshots are updated.
IMXUG v.1	01 July 2021	Minor changes
IMXUG v.0	27 April 2020	Initial version

Table 21. Revision history...continued

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 <u>PSIRT@nxp.com</u>) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

 $\ensuremath{\mathsf{NXP}}\xspace \mathsf{B.V.}$ — 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.

User Guide for Config Tools for i.MX

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, µVision, 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. IAR — is a trademark of IAR Systems AB.

i.MX — is a trademark of NXP B.V.

Kinetis — is a trademark of NXP B.V.

Oracle and Java — are registered trademarks of Oracle and/or its affiliates.

 $\label{eq:processor} \textbf{Processor} ~ \textbf{Expert} - \text{is a trademark of NXP B.V.}$

User Guide for Config Tools for i.MX

Contents

1	Introduction	2
1.1	Features	2
1.2	Versions	2
1.3	Tools localization	3
2	User Interface	
2.1	Start Development wizard	3
2.2	Creating, saving, and opening a	
	configuration	4
2.2.1	Creating a new configuration	4
2.2.1.1	Creating a new standalone configuration	
2.2.2	Saving a configuration	
2.2.3	Opening an existing configuration	
2.2.4	User templates	
2.2.5	Importing sources	
2.2.5.1	Importing configuration	
2.2.5.2	Importing Board/Kit Configuration	
2.2.6	Exporting sources	.12
2.3	Menu bar	
2.4	Toolbar	
2.4.1	Config tools overview	. 16
2.4.2	Update code	
2.4.3	Functional groups	
2.4.3.1	Functional group properties	
2.4.4	Undo/Redo actions	
2.5	Preferences	
2.6	Configuration preferences	
2.7	Problems view	
2.8	Registers view	
2.9	Log view	
3	Pins Tool	
3.1	Pins routing principle	
3.1.1	Beginning with pin/internal signal selection	
3.1.2	Routing of peripheral signals	
3.2	Example usage	
3.3	User interface	
3.3.1	Pins view	-
3.3.2	Package	-
3.3.3	Peripheral Signals view	
3.3.3.1	Filtering in the Pins and Peripheral Signals	
	views	42
3.3.4	Routing Details view	
3.3.4.1	Labels and identifiers	
3.3.5	Expansion Header	
3.3.5.1	Expansion Board	
3.3.6	Power groups	
3.3.7	External User Signals view	
3.3.8	Functions	
3.3.9	Highlighting and color coding	
3.4	Errors and warnings	
3.4.1	Incomplete routing	
3.4.2	Power groups voltage level conflicts	
3.5	Code generation	
3.6	Using pins definitions in code	
3.7	Full initialization of pins	
3.8	Create Default Routing	
4	DDR Tool	
10/10		
IMXUG	All information provided in	n this docume

4.1	Create a new DDR tool project	63
4.2	DDR configuration	
4.2.1	Import initialization script	
4.2.1		
	Import from target	
4.2.3	Enable manual configuration	
4.2.4	UI configuration	
4.2.5	Code generation	
4.3	DDR validation	
4.3.1	Connection	72
4.3.1.1	Boards with Serial Download mode/	
	Manufacture mode	72
4.3.1.2	Boards with JTAG connection available	
4.3.2	Test scenarios	73
4.3.2.1	Inspection	75
4.3.2.2	Optimization	76
4.3.2.3	vTSA	77
4.3.2.4	Stressing	
4.4	FAQ	
5	Trusted Execution Environment Tool	
5.1	AHBSC with security extension-enabled	
	devices	83
5.1.1	User Memory Regions view	83
5.1.2	Security Access Configuration view	84
5.1.2.1	SAU	
5.1.2.2	Interrupts	
5.1.2.3	Secure/Non-secure MPU	
5.1.2.4	MPC	
5.1.2.5	Masters/Slaves	
5.1.2.6	Pins	
5.1.2.7	Miscellaneous	
5.1.3	Memory attribution map	93
5.1.3.1	Core 0	
5.1.3.2	Simple and Smart masters	
5.1.4	Access Overview	
5.1.5	Code generation	
5.2	RDC-enabled devices	
5.2.1	User Memory Regions view	97
5.2.1.1	Access templates	98
5.2.2	Security Access Configuration view	98
5.2.2.1	RDC	
5.2.2.2	XRDC2 Domains view	
5.2.2.3	XRDC (eXtended Trusted Resource	
0.2.2.0	Domain Controller) on Cortex-A35 in i.MX8	
	ULP	108
5.2.2.4	Trusted Resource Domain Controller on	100
5.2.2.4	Cortex-M33 in i.MX8 ULP and KW45	
		444
	(TRDC)	
5.2.2.5	Miscellaneous	
5.2.3	Memory Attribution Map	
5.2.4	Access Overview	
5.2.5	Domains Overview	
5.2.6	Code generation	117
6	Advanced Features	117
6.1	Switching the processor	117
6.2	Exporting the Pins table	
6.3	Tools advanced configuration	
ent is subject to lega	al disclaimers. © 2023 NXP B.V. All rights	reserved.

User Guide for Config Tools for i.MX

6.4	Generating HTML reports120
6.5	Exporting sources
6.6	Exporting registers
6.7	Managing data and working offline
6.7.1	
6.7.2	Working offline
	Downloading data
6.7.3	Exporting data
6.7.4	Importing data
6.7.5	Updating data
6.8	Output path overrides126
6.9	Import pins configuration from legacy tools
	project 127
6.9.1	Importing from an IO Mux Tool design
	configuration file127
6.9.2	Importing from a Processor Expert project128
6.10	Command-line execution
6.10.1	Command-Line execution - Pins Tool
6.10.2	Command-Line execution - TEE Tool
6.11	Batch processing134
6.11.1	Content of the batch file
6.11.1.1	Folders list
6.11.1.2	Files list135
6.11.1.3	Steps list
7	Support
8	Note about the source code in the
U	document
9	Revision history136
J	Legal information
	Legal information130

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

© 2023 NXP B.V.

All rights reserved.

For more information, please visit: https://www.nxp.com