

## APPLICATION NOTE

# **RISC-V MCU Smart Configurator**

User's Guide: IAREW, SEGGER Embedded Studio

## Introduction

This application note describes the basic usage of the RISC-V MCU Smart Configurator (hereafter called the Smart Configurator), and the procedure for importing its output files to IAR Embedded Workbench and SEGGER Embedded Studio.

References to the Smart Configurator and Integrated Development Environment (IDE) in this application note apply to the following version.

- IAR Embedded Workbench for RISC-V V3.30.1 and later
- SEGGER Embedded Studio 8.10 and later
- RISC-V Smart Configurator V1.1.0 and later

Target device and support compiler

Refer to the following URL for the range of supported devices:

https://www.renesas.com/software-tool/risc-v-smart-configurator



R20AN0731EC0101 Rev.1.0.1 Apr.22.24

## Contents

1. Ove	rview	5
1.1 Pu	urpose	5
1.2 Fe	eatures	5
1.3 So	oftware Components	5
2 Inct	alling and Uninstalling the Smart Configurator	6
2. 11150 2.1 Jn	aning and oninstaining the Smart Configurator	00 6
2.1 10	stalling the Smart Configurator	0 C
2.2 01	ninstailing the Smart Configurator	0
3. Ope	erating the Smart Configurator	7
3.1 Pr	ocedure for Operations	7
3.2 St	arting the Smart Configurator	8
3.3 Cı	reating and Loading a Configuration File	9
3.3.1	Creating a New Smart Configurator Configuration File	9
3.3.2	Opening an Existing Configuration File	10
3.4 W	indow	11
3.4.1	Main Menu	12
3.4.2	Toolbar	12
3.4.3	Smart Configurator View	13
3.4.4	MCU/MPU Package View	14
	• · · · ·	45
3.4.5	Console View	15
3.4.5 3.4.6	Console View Configuration Problems View	15 15
3.4.5 3.4.6 4. Sett	Console View Configuration Problems View	15 15 16
3.4.5 3.4.6 4. Sett 4.1 B	Console View Configuration Problems View ing of Peripheral Modules	15 15 16
3.4.5 3.4.6 4. Sett 4.1 Bo 4.1.1	Console View Configuration Problems View ing of Peripheral Modules oard Setting Selecting the Device	15 15 16 16 16
3.4.5 3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2	Console View Configuration Problems View ing of Peripheral Modules oard Setting Selecting the Device Selecting the Board	15 15 16 16 16 17
3.4.5 3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2 4.1.3	Console View Configuration Problems View ing of Peripheral Modules oard Setting Selecting the Device Selecting the Board Exporting Board Settings.	15 15 16 16 16 17 18
3.4.5 3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2 4.1.3 4.1.3	Console View Configuration Problems View ing of Peripheral Modules oard Setting Selecting the Device Selecting the Board Exporting Board Settings	15 15 16 16 16 17 18
3.4.5 3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2 4.1.3 4.1.4 4.2 Cl	Console View Configuration Problems View ing of Peripheral Modules oard Setting Selecting the Device Selecting the Board Exporting Board Settings Importing Board Settings	15 15 16 16 16 17 18 18
3.4.5 3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2 4.1.3 4.1.4 4.2 Cl 4.3 Sv	Console View Configuration Problems View ing of Peripheral Modules oard Setting Selecting the Device Selecting the Board Exporting Board Settings Importing Board Settings lock Settings	15 15 16 16 16 17 18 18 19 20
3.4.5 3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2 4.1.3 4.1.4 4.2 Cl 4.3 Sy 4.4 So	Console View Configuration Problems View ing of Peripheral Modules oard Setting Selecting the Device Selecting the Board Exporting Board Settings Importing Board Settings lock Settings	15 15 16 16 16 18 18 18 19 20 21
3.4.5 3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2 4.1.3 4.1.4 4.2 Cl 4.3 Sy 4.4 So 4.4.1	Console View Configuration Problems View ing of Peripheral Modules oard Setting Selecting the Device Selecting the Board Exporting Board Settings Importing Board Settings lock Settings ystem Settings oftware component settings Switching Between the Component View and Hardware View	15 15 16 16 16 16 18 18 18 19 20 21
3.4.5 3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2 4.1.3 4.1.4 4.2 Cl 4.3 Sy 4.4 So 4.4 So 4.4.1 4.2.2	Console View Configuration Problems View ing of Peripheral Modules oard Setting Selecting the Device Selecting the Board Exporting Board Settings Importing Board Settings lock Settings ystem Settings oftware component settings Switching Between the Component View and Hardware View Adding a Software Component	15 15 16 16 16 17 18 18 19 20 21 21 21
3.4.5 3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2 4.1.3 4.1.4 4.2 Cl 4.3 Sy 4.4 So 4.4 So 4.4.1 4.4.2 4.4.3	Console View Configuration Problems View ing of Peripheral Modules oard Setting Selecting the Device Selecting the Board Exporting Board Settings Importing Board Settings lock Settings ystem Settings oftware component settings Switching Between the Component View and Hardware View Adding a Software Component Setting a Code Generator Component	15 15 16 16 16 17 18 18 19 20 21 21 21 22
3.4.5 3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2 4.1.3 4.1.4 4.2 Cl 4.3 Sy 4.4 So 4.4.1 4.4.2 4.4.3 4.4.3 4.4.4	Console View Configuration Problems View ing of Peripheral Modules oard Setting Selecting the Device Selecting the Board Exporting Board Settings Importing Board Settings lock Settings ystem Settings oftware component settings Switching Between the Component View and Hardware View Adding a Software Component Setting a Code Generator Component Changing the Resource for a Code Generator Configuration	15 15 16 16 16 17 18 18 19 21 21 21 21 22 25 26
3.4.5 3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2 4.1.3 4.1.4 4.2 Cl 4.3 Sy 4.4 So 4.4.1 4.4.2 4.4.3 4.4.3 4.4.4 4.4.5	Console View Configuration Problems View ing of Peripheral Modules oard Setting Selecting the Device Selecting the Board Exporting Board Settings Importing Board Settings lock Settings ystem Settings oftware component settings Switching Between the Component View and Hardware View Adding a Software Component Setting a Code Generator Component Changing the Resource for a Code Generator Configuration Adding a RISC-V MCU Software Integration System Module.	15 15 16 16 16 17 18 18 19 20 21 21 21 21 22 25 26 28
3.4.5 3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2 4.1.3 4.1.4 4.2 Cl 4.3 Sy 4.4 So 4.4.1 4.4.2 4.4.3 4.4.3 4.4.4 4.4.5 4.4.6	Console View Configuration Problems View ing of Peripheral Modules oard Setting Selecting the Device Selecting the Board Exporting Board Settings Importing Board Settings lock Settings ystem Settings oftware component settings Switching Between the Component View and Hardware View Adding a Software Component Setting a Code Generator Component Changing the Resource for a Code Generator Configuration Adding a RISC-V MCU Software Integration System Module Changing Version of BSP Configuration	15 15 16 16 16 17 18 18 19 20 21 21 21 21 22 25 26 28 29
3.4.5 3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2 4.1.3 4.1.4 4.2 Cl 4.3 Sy 4.4 So 4.4.1 4.4.2 4.4.3 4.4.4 4.4.5 4.4.6 4.4.7	Console View Configuration Problems View ing of Peripheral Modules	15 15 16 16 16 17 18 19 21 21 21 21 21 21 25 26 28 29 31
3.4.5 3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2 4.1.3 4.1.4 4.2 Cl 4.3 Sy 4.4 So 4.4.1 4.4.2 4.4.3 4.4.4 4.4.5 4.4.5 4.4.6 4.4.7 4.4.8	Console View Configuration Problems View ing of Peripheral Modules board Setting. Selecting the Device Selecting the Board Exporting Board Settings Importing Board Settings Importing Board Settings lock Settings Switching Between the Component View and Hardware View Switching Between the Component View and Hardware View Adding a Software Component Setting a Code Generator Component Changing the Resource for a Code Generator Configuration Adding a RISC-V MCU Software Integration System Module Changing Version of BSP Configuration Export Component Configuration	15 15 16 16 16 17 18 18 19 20 21 21 21 21 22 26 28 29 31
3.4.5 3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2 4.1.3 4.1.4 4.2 Cl 4.3 Sy 4.4 So 4.4.1 4.4.2 4.4.3 4.4.3 4.4.4 4.4.5 4.4.5 4.4.6 4.4.7 4.4.8 4.4.9	Console View Configuration Problems View ing of Peripheral Modules	15 15 16 16 16 17 18 18 19 20 21 21 21 21 22 25 26 28 29 31 31 31

4.5 Pir	) Settings	. 34
4.5.1	Changing the Pin Assignment of a Software Component	. 35
4.5.2	Assigning Pins Using the MCU/MPU Package View	. 36
4.5.3	Show Pin Number from Pin Functions	. 37
4.5.4	Exporting pin settings	. 38
4.5.5	Importing pin settings	. 38
4.5.6	Pin setting using board pin configuration information	. 39
4.5.7	Pin Filter Feature	. 40
4.5.8	Pin Errors/Warnings setting	. 41
4.6 Int	errupt Settings	. 41
4.6.1	Changing Interrupt Priority Setting	. 42
4.6.2	Interrupt Filter Feature	. 43
4.6.3	Changing Interrupt Vector or Group Setting	. 43
4.6.4	Resolving Interrupt error	. 44
5 Man	aging Conflicts	46
51 Ro	source Conflicts	46
52 Ro	solving Pin Conflicts	47
0.2 100		
6. Gene	erating Source Code	.48
6.1 Ge	nerating Source Code File	. 48
6.2 Co	nfiguration of Generated Files and File Names	. 49
6.3 Ini	tializing Clocks	. 54
6.4 Ini <sup>.</sup>	tializing Pins	. 55
6.5 Ini	tializing Interrupts	. 55
7. Load	ling Generated Files in Integrated Development Environment	.56
7.1 Lo	ading in IAR Embedded Workbench	.56
7.2 Lo	ading in SEGGER Embedded Studio	. 58
7.3 Bu	ild IAR Project File	. 59
7.4 Bu	ild SEGGER Embedded Studio Project File	. 59
	······································	
8. Crea	ting User Programs	.60
8.1 Ad	ding Custom Code	. 60
8.2 Us	ing Generated Code in User Application	. 62
9. Back	ing up Generated Source Code	.62
10. How	to make debug connections via J-Link OB on the FPB Board	.63
11. Gene	erating Reports	.66
11.1 Re	port on All Configurations (PDF or Text File)	. 66
11.2 Co	nfiguration of Pin Function List and Pin Number List (in csv Format)	. 67
11.3 Im	age of MCU/MPU Package (in png Format)	. 67



12. User Code Protection Feature for Smart Configurator Code Generation Comp	oonent
12.1 Specific Tags for the User Code Protection Feature	68
12.2 Examples of Using User Code Protection Feature to Add New User Code	68
12.3 What to Do When Merge Conflict Occurs	69
12.3.1 What is Merge Conflict	69
12.3.2 Steps for Resolving the Merge Conflict	70
13. Help	72
14. Documents for Reference	73



## 1. Overview

### 1.1 **Purpose**

This application note describes the basic usage of the RISC-V MCU Smart Configurator (hereafter called the Smart Configurator), and the procedure for importing its output files to IAR Embedded Workbench / SEGGER Embedded Studio.

Refer to the User's Manual of IAR Embedded Workbench / SEGGER Embedded Studio for how to use them.

## 1.2 **Features**

The Smart Configurator is a utility for combining software to meet user's needs. It handles the following three functions to support the embedding of drivers from Renesas in user's systems: importing middleware in the form of SW integration feature, generating driver code, and making pin settings.

## 1.3 **Software Components**

The Smart Configurator supports three types of software components: Code Generator, Graphical Configurator, and RISC-V MCU Software Integration System:

- <u>Code Generator drivers (DTC, A/D Converter, Interrupt Controller, etc.)</u> The Code Generator drivers is a control program for peripheral functions of microcomputer such as DTC, A/D converter, Interrupt Controller, etc. It is convenient to embed a software component using code generation function.
- (2) <u>RISC-V MCU Software Integration System (BSP, etc.)</u> The RISC-V MCU Software Integration System module is a software component of drivers, middleware SW that provides a simple GUI for generating code.



## 2. Installing and Uninstalling the Smart Configurator

## 2.1 Installing the Smart Configurator

Download the Smart Configurator from the URL below.

https://www.renesas.com/software-tool/risc-v-smart-configurator

After activating the installer, install the Smart Configurator by following the procedure of the installer. User will require administrator privileges to do this.

## 2.2 Uninstalling the Smart Configurator

To uninstall the Smart Configurator, please select "Smart Configurator for RISC-V MCU" from [Apps & features] in the control panel.



## 3. Operating the Smart Configurator

### 3.1 **Procedure for Operations**

Figure 3-1 Operating Procedure, shows the procedure for generating IAR / SEGGER Embedded Studio related files using Smart Configurator and loading it into IAR Embedded Workbench / SEGGER Embedded Studio. For the operation of IAR Embedded Workbench / SEGGER Embedded Studio, refer to relevant document of IAR / SEGGER Embedded Studio.



Figure 3-1 Operating Procedure



## 3.2 **Starting the Smart Configurator**

Select [Smart Configurator for RISC-V MCU Vx.x.x] of [Renesas Electronics Smart Configurator] from the Windows start menu. The main window of the Smart Configurator will be starting.

Note: Please replace Vx.x.x with user's version.



Figure 3-2 Starting of Smart Configurator



## 3.3 Creating and Loading a Configuration File

Smart Configurator saves and refers to the configuration file (\*. scfg) the configuration information of the microcontroller, build tool, peripheral function, pin function etc. used in the project.

#### 3.3.1 Creating a New Smart Configurator Configuration File

On the main window, click the 🗋 [New Configuration File] button to display the [New Smart Configuration File] dialog box.

- (1) In [Platform:] panel, select the Device or Board.
- (2) In [Toolchain:] panel, select the toolchain.

To use the IAR compiler, select [IAR RISC-V Toolchain].

To use the SEGGER compiler, select [SEGGER RISC-V Toolchain].

To use the LLVM compiler, select [LLVM for RISC-V Toolchain].

- (3) In [File name:], enter the file name.
- (4) Confirm [Location:], To change the location, please click [Browse] and select the save destination.

Note:

For IAR RISC-V Toolchain, the \*.eww, \*.ewp, \*.ewd, main.c and buildinfo.ipcf files will be generated to this location after clicking "Generate Code" button. The \*.eww, \*.ewp, \*.ewd and main.c files will be generated only for the first-time code generation, while the buildinfo.ipcf file will be generated always for each time code generation.

For SEGGER RISC-V Toolchain, the \*\_MemoryMap.xml, SEGGER\_Flash\_RV32.icf, \*. emProject, main.c files will be generated to this location after clicking "Generate Code" button. The \*\_MemoryMap.xml, SEGGER\_Flash\_RV32.icf and main.c files will be generated only for the first-time code generation, while the \*. emProject file will be generated always for each time code generation.

(5) Click [Next] to select the project template.

"Bare Metal – Blinky" is available for selected Board in step (1).

(6) Click [Finish] to create the configuration file.

Kow Smart Configuration File	- 🗆 X	
Smart Configuration Settings Select platform and toolchain sett	ings for the new configuration file	Smart Configuration File  Smart Configuration Settings  Choose a project template for the new configuration file
Category: RISC-V MCU Platform: (1) type filter text > Board > Device > G021 > G021 - 16pin > G021 - 24pin > G021 - 32pin > G021 - 48pin R9A02G0214CNE	Toolchain: (2) > IAR RISC-V Toolchain SEGGER RISC-V Toolchain CLLVM for RISC-V Toolchain	Project template selection 5 Bare Metal - Minimal Bare metal project that includes BSP. This project will initialize clocks, pins, drivers and the C runtime environment. Bare Metal - Blinky Bare metal project that includes BSP and will blink LED if available. This project will initialize clocks, pins, drivers and C runtime environment.
Download more boards ROM size: 128 KB, RAM size: 16 KB File name: Smart_Configuration_Ex Location: C:/Users/yang-xiaolin/st < Back(5	, Pin count: 48 ample (3) martconfigurator/workspace(4) Browse Next > Finish Cancel	(6) < Back Next > Finish Cancel

Figure 3-3 Create a Configuration File

(7) Add driver component, configure the setting, generate code, and save the project.



## 3.3.2 **Opening an Existing Configuration File**

On the main window, click the 🗁 [Opening an Existing Configuration File] button to display the [Open] dialog box. Select the (\*. scfg) file and click [Open].

👩 Open						×
$\leftarrow \rightarrow \cdot \cdot \uparrow$	> This PC	→ Local Disk (C:) → sr	martconfigurator > workspace	~ Ō	Search workspace	م
Organize 🔻	New folder					- 🔳 🕐
Pictures		^ Name	^	Date modified	Туре	Size
📑 Videos		Smart_Confi	gurator_Example.scfg	10/15/2018 1:31 PM	SCFG File	1 KB
🏪 Local Disk	(C:)					
i Network						
		<b>~</b>				
	File <u>n</u> ame:	Smart_Configurator_Exa	ample.scfg	~	Smart Configuratio	n files $\vee$
					<u>O</u> pen	Cancel:

Figure 3-4 Opening an Existing Configuration File



## 3.4 Window

The main window is displayed when the Smart Configurator is started. The configuration of the window is shown in Figure 3-5, Main Window.

Smart Configurator     (1)     (2)	- <sup>(3)</sup>
Smart Configuration Example.scg ×	
	Generate Code Generate Repor
Overview     Get an overview of the features provided by Smart Configurator.     Videos     Introduction Smart Configurator     Introduction     Intro	
Carrent Configuration Selected board/device fR3A0200214CNE (RCM size: 128 KR, RAM size: 16 KR, Pin count: 48) Generated location (RR0IOET, LOCU) (scismic, gen Selected components Configuration  Board Support Packages v1.00 1.00 r_bip(used) Conview (Board Clocks System Components. Pins Interrupt) Conview (Board Clocks System Components. Pins Interrupt)	
© Console 🐘 🗟 🔍 🖻 ♥ 🗂 ♥ 🗂 ♥	🗟 Configuration Problems 🔗 👔 = 🗖
Smart Configurator Output (5)	Otens Description Type (6)

Figure 3-5 Main Window

- (1) Menu bar
- (2) Main toolbar
- (3) Smart Configurator view
- (4) MCU/MPU Package view
- (5) Console view
- (6) Configuration Problems view



#### 3.4.1 Main Menu

File Window Help

Table 3-1, Main Menu Items, lists the items of the main menu.

#### Table 3-1. Main Menu Items

	Menu	Details
File	New	The dialog box [New Smart Configurator File], which is used to create a new project, is displayed.
	Open	The dialog box [Open], which opens an existing project, is displayed.
	Save	Saves a project with the same name.
	Restart	Smart Configurator is restarted.
	Exit	Execution of the Smart Configurator is terminated.
Window	Preference	The dialog box [Preference], which is used to specify the properties of the project, is displayed.
	Show View	The dialog box [Show view], which is used to set the view of the window, is displayed.
Help	Help Contents	The help menu is displayed.
	Home Page	Open the home page of the Smart Configurator on the Renesas Electronics website.
	Release Notes	Open the release note of the Smart Configurator on the Renesas Electronics website.
	Tool News	Open the tool news of the Smart Configurator on the Renesas Electronics website.
	API Manual	Open the API manual of the Smart Configurator on the Renesas Electronics website.
	About	The version information is displayed.

#### 3.4.2 **Toolbar**

# 📩 🗁 🔡

Some functions of the main menu are allocated to the buttons on the toolbar. Table 3-2 Toolbar Buttons and Related Menu Items, shows the description of those tool buttons.

Table 3-2. Toolbar Buttons and Related Menu Items

Toolbar button	Related menu item
	[File]→[New]
	[File]→[Open]
	[File]→[Save]



## 3.4.3 Smart Configurator View

The Smart Configurator view consists of seven pages: [Overview], [Board], [Clocks], [System], [Components], [Pins], and [Interrupts]. Select a page by clicking on a tab; the displayed page will be changed.

Smart_Configuration_Example.scfg	×			0	٥
Overview information			😼 Generate Code	Generate Repor	t
- General Information				0	^
Overview         Get an overview of the feat         Smart Configurator.         Videos         Introduction to Smart Confi         Browse related videos         What's New         Check out what's new in th         • Current Configuration         Selected board/device: R9A02G0214         Generated location (PROJECT LOCK)	igurator e latest rele	d by Application Code Software Components Middleware & Drivers Device Drivers MCU Hardware sse.	Smart Configurator ount: 48)		
Selected components:	src\smc_g	Eur			
Component © Board Support Packages v1.00.	Version 1.00	Configuration r_bsp(used)			
Overview Board Clocks System Com	oonents Pir	s Interrupt			Ý

Figure 3-6 Smart Configurator View



#### 3.4.4 MCU/MPU Package View

The states of pins are displayed on the figure of the MCU/MPU package. The settings of pins can be modified from here.

Three types of package view can be switched among [Assigned Function], [Board Function] and [Symbolic Name].

- [Assigned Function] displays the assignment status of the pin setting.
- [Board Function] displays the initial pin setting information of the board. The initial pin setting
  information of the board is the pin information of the board selected by [Board:] on the [Board] page
  (refer to "chapter 4.1 Board Setting" and "chapter 4.5.6 Pin setting using board pin configuration
  information").
- [Symbolic Name] displays the symbolic name defined by user for the pin. Macro definition for the symbolic name will be generated together with port read or write functions in Pin.h file.



Figure 3-7 MCU/MPU Package View



#### 3.4.5 Console View

The Console view displays details of changes to the configuration made in the Smart Configurator or MCU/MPU Package view.



Figure 3-8 Console View

#### 3.4.6 Configuration Problems View

The Configuration Problems view displays the details of conflicts between driver used interrupts, configured peripherals, used pins, used settings.

🔝 Configuration Problems 🔀		
40 errors, 0 warnings, 2 others		
Description	Туре	
> 😣 Interrupt (5 items)		
> 🔕 Peripheral (7 items)		
> 🔇 Pin (26 items)		
> 🔕 Setting (4 items)		

Figure 3-9 Configuration Problems View



## 4. Setting of Peripheral Modules

User can select peripheral modules from the Smart Configurator view.

## 4.1 **Board Setting**

User can change the board and device on the [Board] page.

#### 4.1.1 Selecting the Device

Click on the [ \_\_\_\_ ] button to select a device.

戀 Smart_Co	onfigurator_Ex	ample.so	:fg ×					- 0
Device s	election				Gen	😼 erate Code	Generat	) e Report
Device se	election							2 4
Board:	Custom Use	r Board			$\sim$			
Device:	R7F101GLG	кFB						
	Download m	ore boa	rds	-				
		-	_	- 1	-			
Overview E	Board Clocks	System	Components	Pins	Interrupt	1		

Figure 4-1 Selecting the Device

Note: Device change is not reflected to the device (micro controller) of IAR project.

The following message is displayed when changing the device. For each button operation, refer to "Table 4-1 Device Change Confirmation Operation List".



Figure 4-2 Confirm Device Change

Table 4-1. Device	e Change	Confirmation	Operation	List
-------------------	----------	--------------	-----------	------

Button	Operation explanation
Yes	Change to the selected device.
No	It does not change the device.
Save and continue <sup>(Note*1)</sup>	After saving the current configuration contents to the configuration file, change to the selected device.
Continue <sup>(Note*1)</sup>	Changes to the selected device without saving the current configuration contents to the configuration file.
Cancel <sup>(Note*1)</sup>	It does not change the device.

Note \*1: Smart Configurator view is marked with dirty \*.



#### 4.1.2 Selecting the Board

Click on the [ ] to select a board from the list. After board selection, the pins, clock and system setting will be automatically configured according to board connection.

Smart_Configuration_Example.scfg ×	- 0
Device selection	Generate Code Generate Report
Device selection	ම ක්
Board: Custom User Board	
Device: R9A02G0214CNE	
Download more boards	
Overview Board Clocks System Components Pins Interrupt	

Figure 4-3 Selecting the Board

The following items are changed according to the configuration of the selected board.

- Pin assignment (Initial pin setting)
- Frequency of the main clock
- Frequency of the subsystem clock
- Target device
- On-chip debug operation setting and emulator setting

If user change the board, the message shown in "Figure 4-2" or the following message will be displayed. For each button operation, refer to "Table 4-2 Board Change Confirmation Operation List".

🔞 Confirm device change	×				
Changing the device will refresh all configurations. Configurations that are incompatible with the new device may be removed.					
Do you want to continue?					
Save and continue Continue Cancel					

Figure 4-4 Confirm Board Change

#### Table 4-2. Board Change Confirmation Operation List

Button	Operation explanation
Save and continue	After saving the current configuration contents to the configuration file, change to the selected device.
Continue	Changes to the selected device without saving the current configuration contents to the configuration file.
Cancel	It does not change the device.

Note: Depending on the board selected, the device will change, Device change is not reflected to the device (microcontroller) of IAR project.



#### 4.1.3 Exporting Board Settings

The board settings can be exported for later reference. Follow the procedure below to export the board settings.

- (1) Click on the [ 44 (Export board setting)] button on the [Board] tabbed page.
- (2) Select the output location and specify a name (Display Name) for the file to be exported.

⊜ Smart_C	onfiguration_Example.scfg ×	• 8
Device s	selection	Generate Code Generate Report
Device se	election	(1) 社区
Board:	Custom User Board V	
Device:	R9A02G0214CNE	
	Download more boards	
Overview	Board Clocks System Compo	nents Pins Interrupt

Figure 4-5 Exporting Board Settings (bdf Format)

#### 4.1.4 Importing Board Settings

Follow the procedure below to import board settings.

- (1) Click on the [ 🚵 (Import board setting)] button and select a desired bdf file.
- (2) The board of the imported settings is added to the board selection menu.

Smart_Configuration_Example.scfg ×	° 0
Device selection	Generate Code Generate Report
Device selection	(1) 🖬 🖾
Board: Custom User Board ~	
Device: R9A02G0214CNE (2)	
Download more boards	
Overview Board Clocks System Components Pins Interrupt	

Figure 4-6 Importing Board Settings (bdf Format)

Once a board setting file is imported, the added board is also displayed in the board selection menu of other projects for the same device group.



## 4.2 Clock Settings

User can set the system clock on the [Clocks] page. The settings made on the [Clocks] page is used for all drivers.

Follow the procedure below to modify the clock settings.

- (1) Specify the operation mode and VCC setting.
- (2) Select the clocks required for device operations on the board (the high-speed on-chip oscillator is selected by default).
- (3) Specify the frequency of each clock in accordance with the board specifications (note that the frequency is fixed for some internal clocks).
- (4) For the multiplexer symbol, select the clock source for the output clocks.



Figure 4-7 [Clocks] Page



## 4.3 System Settings

You can set the debug interface pins at [System] tabbed page.

There are 1 type of debug interface available: cJTAG

You can check the pins configured from Console message or MCU/MPU Package view.

*Smart_Configuration_Example.scfg ×	MCU/MPU Package ×	
System  Configuration Generate Code Generate Report	Image: Specific text     Image: Specific text     Image: Specific text       Image: Specific text     Image: Specific text     Image: Specific text       Image: Specific text     Image: Specific text     Image: Specific text       Image: Specific text     Image: Specific text     Image: Specific text       Image: Specific text     Image: Specific text     Image: Specific text       Image: Specific text     Image: Specific text     Image: Specific text       Image: Specific text     Image: Specific text     Image: Specific text       Image: Specific text     Image: Specific text     Image: Specific text       Image: Specific text     Image: Specific text     Image: Specific text       Image: Specific text     Image: Specific text     Image: Specific text       Image: Specific text     Image: Specific text     Image: Specific text       Image: Specific text     Image: Specific text     Image: Specific text       Image: Specific text     Image: Specific text     Image: Specific text       Image: Specific text     Image: Specific text     Image: Specific text       Image: Specific text     Image: Specific text     Image: Specific text       Image: Specific text     Image: Specific text     Image: Specific text       Image: Specific text     Image: Specific text     Image: Specific text       Image: Specific text     Image	*
• On-chip debug setting     Debug interface setting     Ounused     Ounused	24     P109       23     P108       22     P306       21     P207	
	RENESAS         20         P305           19         P304	
	R9A02G021       18       P303         R9A02G0214CNE       17       P302         16       P301       ▲       TMSC         15       P300       ▲       TCKC         14       RES       13       P203	
Coverview Board Clocks System Components Pins     >>	Ν Μ 4 10 0 00 00 00 00 00 00 00 00 00 00 00 0	•
Console ×	🖗 🖻 🔻 🖻 🗖 🔝 Configuration Problems × 🛛 🖓 🕴 🗖	
Smart Configurator Output	0 items	
M05000001: Pin 15 is assigned to TCKC M05000001: Pin 16 is assigned to TMSC	Description	Т

Figure 4-8 Smart Configurator [System] Page Setting



## 4.4 **Software component settings**

Drivers and middleware can be combined as software components on the [Components] page. Added components are displayed in the tree view at the left of the page.

Image: Smart_Configurator_Example.scfg ×			- 8
Software component configura	tion G	enerate Code	🕒 Generate Report
Components       ≥       ≥       ≥       E         Startup              V <>       Startup              V <>       Startup                V <>       Startup	Configure Tree view for components		^
	<		>
Overview Board Clocks System Compo	onents Pins Interrupt		

#### Figure 4-9 [Components] Page

#### 4.4.1 Switching Between the Component View and Hardware View

The Smart Configurator provides two tree view: Component View and Hardware View. Please switch two views by clicking the following icons:

- (1) Click on the [ 5 (Component View)] icon. The tree view will display the components by component category.
- (2) Click on the [ [] (Hardware View)] icon. The tree view will display the components in a hardware resource hierarchy.



Figure 4-10 Switching to the Hardware View



### 4.4.2 Adding a Software Component

The Smart Configurator provides two methods for adding a new component:

- a. Click on the [ (Add component)] icon.
- b. On Hardware Tree, double-click on a hardware resource node.

The following describes the procedure for adding a component by clicking on the [\* (Add component)] icon. a-1. Click on the [\* (Add component)] icon.

∰ Smart_Configurator_Example.scfg ⊠			
Software component configurat	🛐 Generate Code	📄 Generate Report	
Components	Configure		<b>i</b>
type filter text  ✓ ➢ Startup  ✓ ➢ Generic  ✓ ➢ r_bsp			
Overview Board Clocks System Compo	nents Pins Interrupt		

Figure 4-11 Adding a Component

- a-2. Select a component from the list in the [Software Component Selection] page of the [New Component] dialog box (e.g. A/D Converter).
- a-3. Check that [Type] for the selected component is [Code Generator].
- a-4. Click on [Next].

	New Co	mponent						×
	Software Co							
	Select com		τ					
	Category	All						~
	Function	۵						~
								_
	Filter							_
	Compone	ents	Short Name	Туре	Versi			^
2-2	⊞ A/D C	onverter	a-3.	Code Generator	1.0.0			
a-2.	Board	Support Packages v1.00	r_bsp	RISC-V MCU S	1.00			
	- Comp	arator		Code Generator	1.0.0			
	B D/A C	onverter		Code Generator	1.0.0			
	# Delay	Counter		Code Generator	1.0.0			
	🖶 Divide	r Function		Code Generator	1.0.0			
	# Extern	al Event Counter		Code Generator	1.0.0			
	HIC Co	mmunication (Master mode)		Code Generator	1.0.0			
	HIC Co	mmunication (Slave mode)		Code Generator	1.0.0			
	🖶 Input I	Pulse Interval/Period Measure		Code Generator	1.0.0			
	# Input	Signal High-/Low-Level Width		Code Generator	1.0.0			
	🖶 Interru	upt Controller		Code Generator	1.0.0			
	# Interva	al Timer		Code Generator	1.0.0			
	🖶 Key In	terrupt		Code Generator	1.0.0			
	H One-S	hot Pulse Output		Code Generator	1.0.0			
	# Ports			Code Generator	1.0.0			~
	Show o	nly latest version						
	Descriptio	n						
	The analo	og to digital (A/D) converter is fu	nction for converting ar	alog inputs to digi	tal signals.			^
								$\sim$
	Download	RISC-V MCU Software Integratio	on System modules					
	Configure	general settings						
				0.4				
				d-4.				
	Ø			< Back	Next >	Finish	Cance	el 🛛

Figure 4-12 Adding a Code Generator Component



- a-5. Specify an appropriate configuration name in the [Add new configuration for selected component] page of the [New Component] dialog box or use the default name (for e.g., Config\_ADC).
- a-6. Select a hardware resource or use the default resource (for e.g., ADC).
- a-7. Click on [Finish].

New Component -	
Add new configuration for selected component	8 81 8 8
A/D Converter <b>a-5.</b> Config_ADC Resource: <b>a-6.</b> ADC	~
? < Back Next > Finish	Cancel

Figure 4-13 Adding a Component

To add a component on Hardware Tree directly, please use the following procedure:

- b-1. Click on the [ 🛃 (Hardware View Menu)] icon. The tree will display in a hardware resource hierarchy.
- b-2. Double-click on a hardware resource node (for e.g., A/D Converter) to open the [New Component] dialog box.
- b-3. Select a component from the list (for e.g., A/D Converter) to add a new configuration.
- b-4. Follow the same procedure as above "adding a component by clicking adding icon" step a-3 to a-7.

	💰 Smart Configurator								_		
	File Window Help	👩 New C	omponent					×			
	📑 🖆 🔛							i 🖻   🔒	R		
		Software	Component	Selection							2
	*Smart_Configuration_Exa	Select cor	nponent from	n those available	e in list					5 MCU/MPU	
	Software component								eport		6
	Components 🖮 🖬 🖧 🖯	Category	All					$\sim$	^		-
h-	1	Function	All								
2	type filter text	Filter									
	type inter text	6	. ^		<b>CI (1)</b>	-					
	Positime Clock	Compon	ents		Short Name	Type		Versi			
	Watchdog Timer	#A/D Co	nverter			Code Gen	erat	1.0.0	12		
	Independent Watch										
	> Serial Array Unit								Int		
	> 🗀 Serial Interface IICA										
	> 😑 Serial Interface UAF										
h-1	Remote Control Sic	Show o	nly latest ver	sion							
<b>N</b> 1	😕 A/D Converter	Descriptio	n								
	> 🗆 8-Bit D/A Converte	The analo	og to digital (	A/D) converter i	s function for	converting analog	input	s to 🗠			
	> 🗀 Comparator	digital sig	gnals.	, . ,		y			~		
	< >							~	>		
	Overview Board Clocks Svs	Configure	general setti	ngs						+ b legend	
											_
	Smart Configurator Output									D 8	-
	Madaaaaa										
	M04000001: File genera	?		< Back	Next >	Finish	Ca	ncel			
	M04000001: File genera										
	M0400001: File genera	ted: <u>src\s</u>	mc_gen\gen	eral\r_cg_ir							
	M04000001: File genera	ted: <u>src\s</u>	mc_gen\gen	eral\r_cg_ve							
	M04000001: File genera	ted:src\s	mc_gen\gen	eral\r_cg_s							
	M04000001: File genera	ted:src\s	mc_gen\gen	eral\r_cg_ac	×						
	<			>	<					3	>

Figure 4-14 Adding a Code Generator Component to the Hardware View



#### **Removing Software Component**

Follow the procedure below to remove a software component or multiple components from a project.

- (1) Select a software component or multiple components (press and hold CTRL key while selecting the next component) on the Components tree.
- (2) Click on the [ (Remove component)] icon.

Components 🚵 🛃 🎝 🕀 🕀	Components 🖻 🛃 🎘 🕀 🕀				
(2) <b>1</b>					
type filter text	type filter text				
🗸 🗁 Startup					
🗸 🗁 Generic	∽ 🗁 Generic				
💣 r_bsp	💣 r_bsp				
✓	🗸 🗁 Drivers				
✓	✓				
(1) Config_RTC	(1) Config_RTC				
Config_TAU0_0	Config TAU0 0				
Config_TKB0_TKB1	Config_TKB0_TKB1				
Config_TRD0_TRD1	Config_TRD0_TRD1				
✓  → A/D converter	Y 🔁 A/D converter				
Config_ADC	(1) Config_ADC				

Figure 4-15 Removing a Software Component or Multiple Components

The selected software components will be removed from the Components tree.

To delete the source files previously generated for the removed components from the IAR Embedded Workbench project tree, click [ [ Generate Code)] icon.



#### 4.4.3 Setting a Code Generator Component

Follow the procedure below to set up a Code Generator configuration.

- (1) Select a Code Generator configuration from the Components tree (for e.g., A/D Converter).
- (2) Configure the driver in the [Configure] panel to the right of the Components tree. The following steps and figures show an example.
  - a. Select [10 bits] under [Resolution setting].
  - b. Select [Software trigger no wait mode] under [Trigger mode setting].
  - c. Select [ANI0] for [A/D channel selection].
  - d. Select [2112/PCLKB] for [Conversion time].
  - e. Ensure ADC\_ENDI has been configured a vector in [Interrupt] page, detailed usage please refer to 4.6 Interrupt Settings.

Smart_Configurator_Example.scfg ×		
oftware component configura	ion	🔁 👜 Generate Code 🛛 Generate Repo
Components 🖻 🖆 🛓 🛓	Configure	0
View and the second sec	Comparator operation setting	
<ul> <li>✓ E Startup</li> <li>✓ Generic</li> <li>✓ a base</li> </ul>	Resolution setting           0 10 bits         0 8 bits	◯ 12 bits
<ul> <li>✓ isp</li> <li< td=""><td>VREF(+) setting VCC  AVREFP</td><td>O Internal reference voltage</td></li<></ul>	VREF(+) setting VCC  AVREFP	O Internal reference voltage
Config_PORT	VREF(-) setting © VSS O AVREFM	
(2) b.	Software trigger no wait mode     Software trigger no wait mode     Hardware trigger no wait mode     Hardware trigger wait mode     TAU0_ENDI1	
	Operation mode setting Operation mode setting One-shot select mode A/D channel selection One-shot scan mode ANIO	~
	Conversion time setting Conversion time mode Normal 1 Conversion time (2) d. 2112/PCLKB	~ , (44 μs)
	Conversion result upper/lower bound value setting © Generates an interrupt request (ADC_ENDI) when ADLL < ADCRn < A Conversion of interrupt request (ADC_ENDI) when ADLL < ADCRn or a	

Figure 4-16 Setting of a Code Generator Driver

Generation of a code in accordance with each Code Generator configuration is enabled by default.

Right-clicking on a Code Generator configuration and then selecting the [ < Generate code ] icon changes the icon to [ Generate code ] and disables code generation for the Code Generator configuration.

To enable code generation again, click on the [ Generate code ] icon and change it to [ Generate code ].



#### 4.4.4 Changing the Resource for a Code Generator Configuration

The Smart Configurator enables user to change the resource for a Code Generator configuration (for e.g., from TAU0\_1 to TAU0\_3). Compatible settings can be ported from the current resource to the new resource selected.

Follow the procedure below to change the resource for an existing software component.

- (1) Right-click on a Code Generator configuration (for e.g., Config\_TAU0\_1).
- (2) Select [Change resource] from the context menu.



Figure 4-17 Changing the Resource

- (3) Select a new resource (for e.g., TAU0\_3) in the [Resource Selection] dialog box.
- (4) The [Next] button will be active, click on it.

Resource	Selection	—		×
Resource Sel	ection			
Select resou	rce from those available in the list			
Operation:	8 bit count mode			~
Resource:	TAU0_3			~
(3)	TAU0 1 TAU0_3 ITL000 ITL001 ITL012 ITL013			
	(4)			
?	< Back Next > Finish		Cance	el

Figure 4-18 Components Page – Selecting a New Resource



- (5) Configuration settings will be listed in the [Configuration setting selection] dialog box.
- (6) Check the portability of the settings.
- (7) Select whether to use the listed below or default settings.
- (8) Click on [Finish].

Resource Selection		_		×
Configuration setting selection				
Configuration setting list	(7)			
Confirm setting for resource change	● Use setting below 〇	Use de	efault	
Setting	Value		Portable	^
Operation clock	СК02		Yes	
Clock source	fCLK/2		Yes	
Operation mode setting	Higher 8 bits		Yes	
Interval value (higher 8 bits)	10		Yes	
Interval unit	μs		Yes	
Interval value (lower 8 bits)	10		Yes	
Interval unit	μs		Yes	$\sim$
<				>
	(8)			
. Bask	Neutra		C	1
< Back	INEXT > Finish		Cance	1

Figure 4-19 Checking the Settings of the New Resource

The resource is automatically changed (for e.g., changed from TAU0\_END1 to TAU0\_END3).

🕸 *Smart_Configurator_Example.scfg 🗙				- 8
Software component configuration	on			🖲 📄 Generate Code 🛛 Generate Report
Components       Image: Components         type filter text         type filter text         ✓ Startup         ✓ Config_TAU0_1         ✓ Config_PORT         ✓ Config_ADC	Clock setting Operation clock Clock source Operation mode setting	CK02 PCLKB/2 O Lower 8 bits 10 10 counting is started generate an interrup (TAU0_ENDI Level 15 (low) generate an interrupt (TAU0_MOD	μs ~ μs ~ 3) νE8_END13)	Generate Code Generate Report          (Clock frequency: 24000 kHz)         Higher and lower 8 bits         (Actual value: 10)
	Priority	Level 15 (low)	~	
	۲			>
Overview Board Clocks System Compone	ents Pins Interrupt			

Figure 4-20 Resource Changed Automatically



To change the configuration name, follow the procedure below.

- (9) Right-click on the Code Generator configuration.
- (10) Select [Rename] to rename the configuration (for e.g., change Config\_TAU0\_1 to Config\_TAU0\_3).



Figure 4-21 Renaming the Configuration

#### 4.4.5 Adding a RISC-V MCU Software Integration System Module

The following describes the procedure for adding a RISC-V MCU Software Integration System Module.

- (1) Click on the [ (Add component)] icon.
- (2) Select components which [Type] is [RISC-V MCU Software Integration System] from the list in the [Software Component Selection] page of the [New Component] dialog box. Two or more components can be selected by clicking with the Ctrl key pressed.
- (3) Click on [Finish].

	诸 New Co	omponent		-		$\times$
	Software	Component Selection				
	Select con	nponent from those availa	able in li	st		
	Category	All				~
	Function	All				$\sim$
	Filter					
	Compone	ents ^	Shor.	Туре	Versi	^
2	#A/D Co	nverter	(2)	Code Generator	1.0.0	
,2 <b>1</b>	Board S	Support Packages v1.10	r_bsp	RISC-V MCU Software Integration Syst	1.10	
	#Compa	rator		Code Generator	1.0.0	
	BD/A Co	nverter		Code Generator	1.0.0	
	#Data Tr	ansfer Controller		Code Generator	1.0.1	
	#Delay C	Counter		Code Generator	1.0.0	
	# Divider	Function		Code Generator	1.0.0	
	Event Li	ink Controller		Code Generator	1.0.0	
	#Externa	l Event Counter		Code Generator	1.0.0	
	#IIC Com	nmunication (Master m		Code Generator	1.0.1	$\sim$
	Show o	nly latest version				
	Descriptio	n				
	Depende	ncies : None				^
	The r_bsp	package provides a foun	dation f	or code to be built on top of. It provides s	tartup	
	Code ind	ofinac and MCII informat	ion for c	lifferent heards. There are 2 folders that m	naka un	+
	Configure	general settings	legratio	n system modules		
	comigure	general settings				
				(3)		
1	2		< Rac	k Next > Finich	Cance	J

Figure 4-22 Adding RISC-V MCU Software Integration System Module



## 4.4.6 Changing Version of BSP Configuration

The following describes the procedure for version change of BSP configuration.

(1) From the component tree, right-click the r\_bsp component whose version user wants to change.

🔅 *Smart_Configurator_Example	e.scfg 🛛			
Software component cor	nfiguration		Generate Code	🗎 Generate Report
Compo J <sup>a</sup> <sub>Z</sub> 🗭 🕀	Configure			١
Sta top type filter text	Property v 🏶 Configurations		Value	^
<ul> <li>✓</li></ul>	# Start up select # Control of invalid memory acce # RAM guard space(GRAM0-1)	ess detection	Enable (use BSP startup) Disable Disabled	
💱 r_bsp 🛛 Chi	ange version	rt function(GPORT)	Disabled	
Rer Re:	move set to default	unction(GINT) xck control function, voltage detector, and RAM parity erro )	Disabled Disabled Disables	
Do	wnload and import sample projects	ons by Code Generator/Smart Configurator	Enable	
	# API functions disable		Enable	
	# Parameter check enable		Enable	
	<pre># Setting for starting the high-sp &lt;</pre>	eed on-chip oscillator at the times of release from STOP mc	High-speed	× ×
Overview Board Clocks System	n Components Pins Interrupt			

Figure 4-23 Version Change of BSP Configuration

- (2) Select [Change Version ...] from the context menu.
- (3) In the [Change Version] dialog box, select the version you want to change. If user selects a version that the device does not support, [Selected version doesn't support current device or toolchain] will be displayed.
- (4) Click [Next].

Change Version				_		×
Version Selection						
Select available ver	rsion					
Component name:	r_bsp					
Current version:	1.00					
Available versions:	1.01					$\sim$
l	< Back	Next >	Finis	h	Cance	I

Figure 4-24 Select Version of BSP Component

(5) By version change, a list of setting items to be changed is displayed. Confirm that there is no problem and click the [Finish].



👩 Change Version	_		×
Setting Overview			
The following settings will be added or removed			
Setting		Status	
<			>
< Back Next > Finish		Cance	I

Figure 4-25 Confirm Setting Change Item

(6) As [Confirm to change version and proceed to generate code] is displayed, if user do not have any problem, click [Yes].



Figure 4-26 Confirm Version Change

(7) The BSP component version is changed, and code generation is executed automatically.



## 4.4.7 Export Component Configuration

The current configuration can be exported as \*.xml file by clicking on the [La (Export Configuration)] button on the [Components] tabbed page.

Components 🚵 🛃 🛱 🕀	
🤋 🖏	
type filter text	
∽ 🗁 Startup	
🗸 🗁 Generic	
💣 r_bsp	
✓	
✓	
Config_TAU0_1	

Figure 4-27 Export Configuration (xml format)

## 4.4.8 Import Component Configuration

Click on the [12] (Import Configuration)] button and select an exported xml file will import component configuration.

Components	è la la E
56 ig	😜 🗟
type filter text	
🗸 🗁 Startup	
🗸 🗁 Generic	
💣 r_bsp	
🗸 🗁 Drivers	
🗸 🗁 Timers	
💣 Config_	TAU0_1

Figure 4-28 Import Configuration (xml format)



#### 4.4.9 **Configure General Setting of the Component**

The general setting of the component, such as code generation component settings, dependency settings and location settings, can be configured inside the [Preferences] dialog.

If you want to change the settings, please click the [Configure general settings...] link on the [Software Component Selection] page displayed in the [New Component] dialog (Figure 4-12) and display the [Preferences] dialog. Or click "Preferences" of "Window" in Main Menu.

C Preferences	—		×
type filter text	Component	⇔ ◄ <	× 8
<ul> <li>Help Module Download</li> <li>Smart Configurator Component MCU/MPU Package Appearance Pin Errors/Warnings</li> </ul>	Backup settings ☐ Enable Backup settings Number of trash item (1-20): 5 Code Generator component settings API function output: Output all API functions according to the API code style: Value with macro description FIT(RX) / SIS(RL78 / RISC-V MCU) component settings Code generation behavior: Re-generate all component files	setting	~
	Dependency settings Change these options to control how a component is added Adding dependency: Add dependent component Checking dependency: Ignore if dependent component is new Location settings Location settings have moved to the <u>Module Download</u> page	/er	~
	Apply and Close	Cancel	y

Figure 4-29 Configure General Setting of Component

#### Notes:

1. User can limit the number of folders created in the trash folder for backup purposes by setting the [Number of trash item (1-20)] option in the figure below. Once exceeding the limit, a folder with the newer timestamp will replace the oldest folder.

Backup settings ☑ Enable Backup settings		
Number of trash item (1-20):	5	

Figure 4-30 Trash number setting

2. The code generation behavior has two options: [Update configuration files] and [Re-generate all component files]. [Update configuration files] is the default selection. If "Update configuration files" is being selected and generate code, Smart Configurator will check whether the files are existing inside the user project. If the file exists, the file will not be overwritten. However, configuration files (e.g., xxx\_config.h) will still be refreshed when code is generated. If "Re-generate all component files" being selected and generate code, Smart Configurator does not check the existence of the file and the file will always be overwritten.

FIT(RX) / SIS(RL78 / RISC-V MCU) component settings							
Code generation behavior: Re-generate all component files							
Dependency settings	Update configuration files						
Change these entires to control being and component files							

Figure 4-31 [Code generation behavior] Change



To only generate initialization API function, please change to [Output only initialization API function] option in below figure. So that only void R\_{ConfigurationName}\_Create (void), void R\_{ConfigurationName}\_Create\_UserInit (void) in \*.h \*, \*c \* are generated. If user change back to default option setting: [Output all API functions according to the setting], then all API functions will be generated again.

Code Generator component settings					
API function output	Output all API functions according to the setting	~			
API code style	Output all API functions according to the setting				
Output only initialization API function					

Figure 4-32 [RISC-V MCU API function output] Change

Please right-click the selected component and select the "Output only initialization API" from the context menu.



#### Figure 4-33 Context Menu "Output only initialization API" for Each Configuration

4. To generate code with HEX value, please change to [Value without macro description (raw HEX)] option in below figure. If user change back to default option setting: [Value with macro description], then all API with macro description will be generated again.

Code Generator com	ponent settings				
API function output: Output all API functions according to the setting					
API code style:	Value with macro description	~			
FIT(RX) / SIS(RL78) co	Value with macro description Value without macro description (raw HEX)				

Figure 4-34 [API code style] Change

5. If the version of the module and its dependency do not match, a warning message W04020011 is displayed. If user check the revision history of the module and its dependencies and do not need to change the module used, please ignore this warning. To clear this warning, please select [Do not check for dependent component] in the [Checking dependency] list box in component preferences, then click [OK].

Checking dependency:	Ignore if dependent component is newer 🛛 🗸
	Do not check for dependent component
Location settings	Ignore if dependent component is newer
Specify location of comp	Strict check for dependent component

Figure 4-35 [Checking dependency] Change



## 4.5 **Pin Settings**

The [Pins] page is used for assigning pin functions. User can switch the view by clicking on the [Pin Function] and [Pin Number] pages. The [Pin Function] list shows the pin functions for each of the peripheral functions, and the [Pin Number] list shows all pins in order of pin number.

n configuration						🐻 Generate Code Gen	ierate Repo
ardware Resource 🛛 🕀 🛱 🔒	Pin Functio	on				<del>ହ</del> ା 🖬	<b>R</b>   24 2
Type filter text	type filter	text (* = any	string, ? = any character)			All	~
👗 All 🔨	Enabled	Function	Assignment	Pin Number	Direction	Remarks	^
🗱 Resets		P011	Not assigned	Not assigned	None		
🗱 Clock Generation Circuit		P100	/ P100/TI05/TO05/SO00/T	/ 25	0		
🥨 Clock Frequency Accuracy Measur		P101	Not assigned	Not assigned	None		
👪 Interrupt Controller Unit		P102	Not assigned	Not assigned	None		
\$∰ I/O Ports		P103	Not assigned	Not assigned	None		
🚣 Key Interrupt Function		P104	Not assigned	Not assigned	None		
🗸 🥨 Timer Array Unit		P105	Not assigned	Not assigned	None		
> TAU0		P106	Not assigned	Not assigned	None		
🥨 Realtime Clock		P107	Not assigned	Not assigned	None		
> ≇¦∰ Serial Array Unit		P108	Not assigned	Not assigned	None		
✓ ≇¦∰ Serial Interface IICA		P109	Not assigned	Not assigned	None		
IICA0		P110	Not assigned	Not assigned	None		
📦 IICA1		P111	Not assigned	Not assigned	None		
✓ ೫∰ Serial Interface UARTA		P200	Not assigned	Not assigned	None		
UARTA0		P201	Not assigned	Not assigned	None		
UARTA1		P202	Not assigned	Not assigned	None		
📲 Remote Control Signal Receiver		P203	Not assigned	Not assigned	None		
🗟 A/D Converter		P204	Not assigned	Not assigned	None		
γ Ω. 8-Rit D/Δ Converter					••		
		a la lua cu					
Function Pin Number DIS	play Switt	uning					

Figure 4-36 [Pins] Page ([Pin Function])

When you select a board on the [Board] page, the initial pin setting information of the board is displayed in [Board Function]. In addition, the [**II**] icon displayed in the [Function] selection list indicates the initial pin function of the board.

*Smart_Cont Pin configu	igurator_Example.scfg ×					Generate Code G	enerate Report
Pin Number							
type filter te	kt (* = any string, ? = any character)					All	~
Pin Numb	Pin Name	Board Functi	Function	Direction	Remarks	Symbolic Name	Comme ^
1	VCL		VCL	-	Read only	-	
2	XT2	XT2	🖬 XT2 🗸 🗸	0		-	XT2
3	XT1	XT1	Not assigned	ł		-	XT1
4	VSS/AVSS	VSS	XT2	lone		-	
5	VCC/AVCC	VCC	Not assigned	None		-	
6	P200/NMI		Not assigned	None			
7	P201/IRQ3	P201	Not assigned	None			PMOD1
8	P202/CLKOUT/RIN0/IRQ2	P202	Not assigned	None			PMOD2
9	P204/SCK21/SCL21	P204	Not assigned	None			PMOD1
10	P205/SI21/SDA21	P205	Not assigned	None			PMOD1
11	P206/SO21	P206	Not assigned	None			PMOD1
12	P307	P307	Not assigned	None			PMOD2
13	P203/MD	MD	Not assigned	None			MD
14	RES	RES	Not assigned	None		-	ResetSV
15	P300/EXTAL/TCKC/TI07/TO07/SCK00/SCL00/IRQ0	TCKC	тскс	1			JlinkOB
16	P301/TMSC/TI06/TO06/SI00/SDA00/RxD0/IRQ1	TMSC	TMSC	10			JlinkOB
17	P302/VCOUT1/TI03/TO03/SCLA0/TxD0/IRQ3	TxD0	Not assigned	None			JlinkOB
18	P303/CLKOUT/TI04/TO04/SDAA0/RxD0/IRQ2	RxD0	Not assigned	None			JlinkOB
19	P304/SO01/KR00	P304	Not assigned	None			ARDUIN Y
<							>
Pin Function	Pin Number						
Overview Boa	d Clocks System Components Pins Interrupt						

Figure 4-37 [Pins] Page ([Pin Number])



#### 4.5.1 Changing the Pin Assignment of a Software Component

The Smart Configurator assigns pins to the software components added to the project. Assignment of the pins can be changed on the [Pins] page.

This page provides two lists: Pin Function and Pin Number.

Follow the procedure below to change the assignment of pins to a software component in the Pin Function list.

- (1) Click on [🚠 (Show by Hardware Resource or Software Components)] to switch to the component view.
- (2) Select the target software component (for e.g., Config\_ICU).
- (3) Click the [Enabled] header to sort by pins used.
- (4) In the [Assignment] column or [Pin Number] column on the [Pin Function] list, change the pin assignment (for e.g., change from P201 to P302).
- (5) In addition, assignment of a pin can be changed by clicking on the [ (Next group of pins for the selected resource)] button. Pin that has peripheral function is displayed each time the button is clicked.

*Smart_Configurator_Example.scfg	×							
Pin configuration								Generate Code 🛛 Generate Repo
(1) Software Compone (1)	æ P	in Functio	on					(5) 💽 🖬 🖬 🗠 🖉
Type filter text		type filter	text (* = any stri	ng, ? = any character)				All ~
≺ 🚣 r_bsp	(3)	Enabled	Function	Assignment	Pin Number	Direction	Remarks	
📦 r_bsp	1		IRQ0	Not assigned	Not assigned	None		
🗸 💑 Ports			IRQ1	Not assigned	Not assigned	None		
Config_PORT			IRQ2 (4)	Not assigned	Not assigned	None		
∽ 💑 A/D Converter			IRQ3	/ P201/IRQ3	/ 7	1		
Config_ADC			IRQ4	Not assigned	Not assigned	None		
🗸 💑 Interval Timer			IRQ5	Not assigned	Not assigned	None		
Config_TAU0_1			IRQ6	Not assigned	Not assigned	None		
Controller			IRQ7	Not assigned	Not assigned	None		
Config_ICU			NMI	Not assigned	Not assigned	None		
		<			-			>
Pin Function Pin Number								
Verview Board Clocks System Com	nponent	ts Pins In	terrupt					

Figure 4-38 Pin Settings – Assigning Pins on the [Pin Function] List

The Smart Configurator allows user to enable pin functions on the [Pins] page without linking the current software component to another. To distinguish these pins from other pins that are used by another software component, there will be a remark "There is no software initializing this pin" on the list.



#### 4.5.2 Assigning Pins Using the MCU/MPU Package View

The Smart Configurator visualizes the pin assignment in the MCU/MPU Package view. User can save the MCU/MPU Package view as an image file, rotate it, and zoom in to and out from it.

Follow the procedure below to assign pins in the MCU/MPU Package view.

- (1) Zoom in to the view by clicking the [*p*<sup>(2)</sup> (Zoom in)] button or scrolling the view with the mouse wheel.
- (2) Right-click on the target pin.
- (3) Select the signal to be assigned to the pin.
- (4) The color of the pins can be customized through [Preference Setting...].



Figure 4-39 Assigning Pins Using the MCU/MPU Package View


#### 4.5.3 **Show Pin Number from Pin Functions**

User can go to the pin number associated with a pin function.

Follow the procedure below to jump to pin number from a pin function.

- (1) In the [Pin Function] tab, right click on a Pin Function to open the pop-up menu.
- (2) Select "Jump to Pin Number".
- (3) The [Pin Number] tab is opened with a Pin Number being selected. This is the pin number of the pin function.

n configu	iration							Generate	Code Generat	) Renv
oftware Cor	mpon ∓ 🖻 🔒 🔒	Pin Functi	ion							èn e
Type filter to	ext	type filte	r text (* = anv	string, ? = any c	haracter)				All	
v 🕹 e hen							<b>e</b> 1 - 1			
• m r_osp		Enabled	Function	Assignment	t	Pin Number	Direction	Remarks		í
v 🕹 Dorte	,sh		IRQ0	Not assigned	gned	Not assigned	None			
2 Co	ofic PORT		IRQ1	Not assigned	gned	Not assigned	None			
× ≛ ∆/D (	Converter		IRO2	Not assig	aned	Not assigned	None			
2 Co	onfig ADC		IRQ3	/ P201/IRC	23	17	Ju	ump to Pin Number		
v 🍈 Interv	val Timer		IRQ4	Not assigned	gned	Not assigned	NO	Aerge comment to Pin Number t	ab	
Co	onfig TAU0 1		IRQS	<ul> <li>Not assig</li> </ul>	gned	<ul> <li>Not assigned</li> </ul>	NO C	lear comments		
🗸 🔬 Interr	rupt Controller		IRQ0	<ul> <li>Not assig</li> </ul>	gned	<ul> <li>Not assigned</li> </ul>	NO O	ssion selected pips		
Co	onfig ICU		IKQ/	<ul> <li>Not assig</li> </ul>	gnea	<ul> <li>Not assigned</li> </ul>		lassing extend and		
	<b>U</b>		INIVII	<ul> <li>Not assig</li> </ul>	gnea	<ul> <li>Not assigned</li> </ul>		inassign selected pins		`
rview Boar Smart_Confi	igurator_Example.scfg ×	nents Pins	Interrupt		-					6
rview  Boar Smart_Confi configu	igurator_Example.scfg × [	nents Pins	Interrupt		-			Generate	Code Generat	e Re
Smart_Confi configu	igurator_Example.scfg ×	nents Pins	Interrupt		-			Generate	Code Generat	e Re
Gmart_Confi configu Number	igurator_Example.scfg ×	nents Pins	Interrupt		-	<b></b>		Generate	Code Generat	e Re
mart_Confi configu Number	d Clocks System Compo igurator_Example.scfg × ration	nents Pins	Interrupt	Board Functi	Function	Direction	Remarks	Generate Symbolic Name	Code Generat	e Re
mart_Confi configu Number n Numb 1	igurator_Example.scfg × ration Pin Name VCL	nents Pins	Interrupt	Board Functi	Function VCL	Direction -	Remarks Read only	Generate Symbolic Name -	Code Generat	e Re
imart_Confi configu Number in Numb 1 2	igurator_Example.scfg × ration Pin Name VCL XT2	nents Pins	Interrupt	Board Functi	Function VCL XT2	Direction - O	Remarks Read only	Generate Symbolic Name - -	Code Generat	e Re
imart_Confi configu Number in Numb 1 2 3	d     Clocks     System     Compo       igurator_Example.scfg     X       ration         Pin Name       VCL       XT2       XT1	nents Pins	Interrupt	Board Functi	Function VCL XT2 XT1	Direction  O I	Remarks Read only	Generate Symbolic Name	Code Generat	ء e Re
imart_Confi configu Number	d     Clocks     System     Compo       igurator_Example.scfg     X       ration         Pin Name       VCL       XT2       XT1       VSS/AVSS	nents Pins	Interrupt	Board Functi	Function VCL XT2 XT1 Not assigned	Direction - O I None	Remarks Read only	Generate Symbolic Name - - - - - - - - -	Code Generat	e Re
imart_Configu configu Number 1 2 3 4 5	d     Clocks     System     Compo       igurator_Example.scfg     X       ration         Pin Name       VCL       XT2       XT1       VSS/AVSS       VCC/AVCC	nents Pins	Interrupt	Board Functi	Function VCL XT2 XT1 Not assigned Not assigned	Direction - O I None None	Remarks Read only	Generate Symbolic Name - - - - - - - -	Code Generat	≥ e Re   ≥
imart_Configu configu Number 1 2 3 4 5 6	d     Clocks     System     Compo       igurator_Example.scfg     X       ration         Pin Name       VCL       XT2       XT1       VSS/AVSS       VCC/AVCC       P200/NMI	nents Pins	Interrupt	Board Functi	Function VCL XT2 XT1 Not assigned Not assigned Not assigned	Direction - O I None None None	Remarks Read only	Generate	Code Generat	= e Re   ≧
Smart_Configu configu Number 1 2 3 4 5 6 7	d Clocks System Compo igurator_Example.scfg × ration Pin Name VCL XT2 XT1 VSS/AVSS VCC/AVCC P200/NMI P201/IRQ3	nents Pins	Interrupt	Board Functi	Function VCL XT2 XT1 Not assigned Not assigned Not assigned IRQ3	Direction - O I None None None I	Remarks Read only	Symbolic Name	Code Generat	■ e Re   ≧
in Number	d     Clocks     System     Compo       igurator_Example.scfg     X       ration         Pin Name       VCL       XT2       XT1       VSAVSS       VCC/AVCC       P200/NMI       P201/IRQ3       P202/CLKOUT/RIN0/IRQ2	ents Pins	Interrupt	Board Functi	Function VCL XT2 XT1 Not assigned Not assigned IRQ3 Not assigned	Direction - O I None None None I None	Remarks Read only	Symbolic Name	Code Generat	e Re
in Number	d     Clocks     System     Compo       igurator_Example.scfg     X       ration         Pin Name       VCL       XT2       XT1       VSS/AVSS       VCC       P200/NMI       P201/IRQ3       P202/CLKOUT/RIN0/IRQ2       P204/SCK21/SCL21	ents Pins	Interrupt	Board Functi	Function VCL XT2 XT1 Not assigned Not assigned IRQ3 Not assigned Not assigned	Direction - O I None None None I None None	Remarks Read only	Symbolic Name	Code Generat	e Re
Smart_Configu Smart_Configu I Number in Number 1 2 3 4 5 6 7 8 9 9	d     Clocks     System     Compo       igurator_Example.scfg     X       ration         Pin Name       VCL       XT2       XT1       VSS/AVSS       VCC/AVCC       P200/CLKOUT/RIN0/IRQ2       P204/SCK21/SCL21       P205/CLAUT/RIN0/IRQ2	ents Pins	Interrupt	Board Functi	Function VCL XT2 XT1 Not assigned Not assigned IRQ3 Not assigned Not assigned Not assigned	Direction - O I None None I None None None None None None	Remarks Read only	Generate Generate	Code Generat	e Rej

Figure 4-40 Jump to Pin Number



#### 4.5.4 Exporting pin settings

The pin settings can be exported for later reference. Follow the procedure below to export the pin settings.

- (1) Click on the [12] (Export pin assignments)] button on the [Pins] page.
- (2) Select the output location and specify a name for the file to be exported.

The exported XML file can be imported to another project having the same device part number.

n configuration							Generate Code Generate Rep
oftware Compon 🕀 🖻 💐 Z	Bin Function	on					२ 🗉 🖬 🔤
	type filter	text (* = any	string, ? = any character)				All
∨ 💑 r_bsp	Enabled	Function	Assignment	Pin Number	Direction	Remarks	
📦 r_bsp		IRQ0	Not assigned	Not assigned	None		
🗸 🚣 Ports		IRQ1	Not assigned	Not assigned	None		
💣 Config_PORT		IRQ2	Not assigned	Not assigned	None		
∨ 🚣 A/D Converter		IRQ3	/ P201/IRQ3	/ 7	1		
💣 Config_ADC		IRQ4	Not assigned	Not assigned	None		
🗸 🚣 Interval Timer		IRQ5	Not assigned	Not assigned	None		
Config_TAU0_1		IRQ6	Not assigned	Not assigned	None		
🗸 🚣 Interrupt Controller		IRQ7	Not assigned	Not assigned	None		
😴 Config_ICU		NMI	Not assigned	Not assigned	None		
	<			-			>
E di Din Number	<						>

Figure 4-41 Exporting Pin Settings to an XML File

The Smart Configurator can also export the pin settings to a CSV file. Click on the [L] (Save the list to .csv file)] button on the [Pins] page.

#### 4.5.5 Importing pin settings

To import pin settings into the current project, click on the [22] (Import pin assignments)] button and select the XML file that contains the desired pin settings. After the settings specified in this file are imported to the project, the settings will be reflected in the [Pin configuration] page.

<pre># *Smart_Configurator_Example.scfg ×</pre>								- 8
Pin configuration							関 Generate Code G	ienerate Report
Software Compon $\textcircled{1} \boxdot \downarrow_z^a \textcircled{B}$	Pin Function	on					<del>ହ</del> ା 🗉	<b>I</b>   <mark>21</mark> 4
	type filter	text (* = any	string, ? = any character)				All	~
∽ 💑 r_bsp	Enabled	Function	Assignment	Pin Number	Direction	Remarks		^
🔍 r_bsp		IRQ0	Not assigned	Not assigned	None			
V 💑 Ports		IRQ1	Not assigned	Not assigned	None			
Config_PORT		IRQ2	Not assigned	Not assigned	None			
V 🚣 A/D Converter		IRQ3	/ P201/IRQ3	/ 7	1			
Config_ADC		IRQ4	Not assigned	Not assigned	None			
🗸 🚣 Interval Timer		IRQ5	Not assigned	Not assigned	None			
Config_TAU0_1		IRQ6	Not assigned	Not assigned	None			
V 🚣 Interrupt Controller		IRQ7	Not assigned	Not assigned	None			
Config_ICU		NMI	Not assigned	Not assigned	None			~
	<							>
Plant Die Number								
Pin Function Pin Number								
Overview Board Clocks System Compo	onents Pins	Interrupt						

Figure 4-42 Importing Pin Settings from an XML File

Note: The pin setting is reflected, but it is not reflected in the component setting.



#### 4.5.6 **Pin setting using board pin configuration information**

User can set the initial pin configuration according to the Renesas board that you selected to use. User can check the board that selected to use in [Board] tabbed page.

The following describes the procedure for collective setting of pins.

- (1) Select a board setting information except [Custom User Board] in [Board] page. User can refer to 4.1.3 Exporting Board Settings
- (2) Select [Board Function] in the MCU/MPU Package. (The initial pin configuration of the board can be referred.)
- (3) Open the [Pin Configuration] page and click the [Assign default board pins]
- (4) When [Assign default board pins] dialog opens, click [Select all].
- (5) Click [OK].

💮 *Smart_Configurator_Exam	ple.scfg $ imes$		📴 Defa	ult Board Pin	×			- 0	MCU/MPU Package ×     (2)
Pin configuration			Assign (	default board pins		rate C	Code Genera	ite Report	Type filter text
Softw 🕀 🖃 ↓ª₂ 🚵	Pin Functi	on		(4	)	(3	) 	22	Udüddanüdddd
Type filter text	type filter	Function	Assign	ANI2 ANI3	Select all De-select all	er	All	Ker ^	
> 🚣 A/D Converter > 🚠 Interval Timer > 💑 Interrupt Contro		ANI1 ANI2 ANI2	$\square$	ANI4 ANI5 ANI18		igned igned	None None	The	
		ANIA ANIA ANI5		ANI19 AVREFM		igned igned	None None		
		ANI17 ANI18		IRQ4 MD		igned igned	None		
Pin Function Pin Number Overview Board Clocks Sys	tem Comp	onents Pins	N N N	P009 P100 P104 v					± ε ε ε ε ε ε ε ε ε ε ε ε ε ε ε ε ε ε ε
			(5)	ОК	Cancel				

Figure 4-43 Setting for Initial Pin Configuration

If user does not set pin settings all at once, specify them individually in procedure (4).



### 4.5.7 **Pin Filter Feature**

By specifying the filter range on the [Pin Function] page and [Pin Number] page on the [Pins] page, user can refer to it more easily.

in configuration				Generate Co	ode Generate	a Rep
Softw 🕀 🖃 📮 😹	Pin Functi	on			2 🖬 🖬	èn 1
Type filter text	type filter	text (* = any	string, ? = any character)		All	
	Enabled	Function ANI3 ANI4 ANI5 ANI16 ANI17	Assignment / Not assigned / Not assigned / Not assigned / Not assigned / Not assigned	Pin Number Not assigned Not assigned Not assigned Not assigned Not assigned	All Function Assignment Pin Number Direction Remarks Comments	
		ANI18 ANI19 AVCC AVREFM	<ul> <li>Not assigned</li> <li>Not assigned</li> <li>Not assigned</li> <li>Not assigned</li> </ul>	<ul> <li>Not assigned</li> <li>Not assigned</li> <li>Not assigned</li> <li>Not assigned</li> <li>Not assigned</li> </ul>	None None None None	_
< >	<					>

Figure 4-44 Filter for [Pin Function] Page

n configu	ration		Generat	e Code Generate Repo
n Number				📕 📑 🖬 🖬
ype filter tex	tt (* = any string, ? = any character)			All 🕓
Pin Numb	Pin Name	Board Functi	Function	Di Din Number
10	P205/SI21/SDA21	P205	Not assigned	Nc Pin Name
11	P206/SO21	P206	Not assigned	Nc Board Functions
12	P307	P307	Not assigned	Nc Function
13	P203/MD	MD	Not assigned	Nc Direction
14	RES	RES	Not assigned	Nc Symbolic Name
15	P300/EXTAL/TCKC/TI07/TO07/SCK00/SCL00/IRQ0	TCKC	TCKC	Comments
16	P301/TMSC/TI06/TO06/SI00/SDA00/RxD0/IRQ1	TMSC	TMSC	10
17	P302/VCOUT1/TI03/TO03/SCLA0/TxD0/IRQ3	TxD0	Not assigned	None
18	P303/CLKOUT/TI04/TO04/SDAA0/RxD0/IRQ2	RxD0	Not assigned	None
10	0304/CO01/K000	0204	Nine and and	×>

Figure 4-45 Filter for [Pin Number] Page



#### 4.5.8 Pin Errors/Warnings setting

User can control how pin problem is displayed on Configuration Problems view by using the Pin Errors/Warnings setting. If you want to control it, on the [New Component] dialog, click the [Configure general settings...] link to display the [Preferences] dialog. Then select [Smart Configurator] > [Pin Errors/Warnings] and use the combo boxes to change the errors/warning setting.

<b>Preferences</b>			
type filter text	Pin Errors/Warnings	⇔ ▼ ⇒ ▼ §	8
<ul> <li>&gt; Help</li> <li>&gt; Smart Configurator Component</li> <li>MCU/MPU Package</li> <li>Pin Errors/Warnings</li> </ul>	<ul> <li>Pin Conflict Multiple functions are assigned in one pin number</li> <li>No Pin Allocation Function used by software but not allocated to any pin</li> <li>Mutually Exclusive Pins Mutually exclusive pins cannot be allocated to the same pin at the same time.</li> <li>No Software Assigned pins but there's no software using them</li> <li>Different Group Functions in same channel but different group</li> <li>Board Mismatch Pin assignment does not match the board suggested pin assignment</li> </ul>	Error × Error × Error × Info × Warning ×	
< >>	Restore Defaults	Apply	
	Apply and Close	Cancel	

Figure 4-46 Pin Errors/Warnings Settings at Preferences

Example: Change "No Software" setting from "Info" to "Error"



Figure 4-47 Change "No Software" Setting from "Info" to "Error"

### 4.6 Interrupt Settings

The [Interrupt] page displays all interrupts by each of the vector numbers. User can check and set the interrupts of the peripheral modules that have been selected on the [Components] page. When an interrupt is used in a Code Generator configuration on the [Components] page, the status of the interrupt will be changed to "Used".

To display the used interrupts only, click on the [K] (Show used interrupts)] button.



rrupt configu	ration					Generat	e Code G	Generate R
errupt vectors us	ed							
Type filter text				Ve	ector Numbe	er		
Vector Number	Vector Table	Interrupt	Interrupt request source	Peripheral	Group	Priority	Status	Remarks
0	0x04C	IRQ0(PORT_IRQ0)	External pin interrupt 0	PORT	0	Level 15 (low)	Used	
1	0x050	IRQ1(PORT_IRQ1)	External pin interrupt 1	PORT	1	Level 15 (low)	Used	
7	0x068	IRQ7(PORT_IRQ3)	External pin interrupt 3	PORT	7	Level 15 (low)	Used	
14	0x084	IRQ14(PORT_IRQ2)	External pin interrupt 2	PORT	6	Level 15 (low)	Used	
<								

Figure 4-48 Show used interrupts

## 4.6.1 Changing Interrupt Priority Setting

User can change the interrupt priority level on the [Interrupt] page using the following procedure:

- (1) Find the interrupt which user wants to change priority setting on this page.
- (2) Click the priority cell and select an interrupt priority level from the drop-down list.

mart_Configurator	_Example.scfg >	<							-
errupt configu	ration					Generate	Code (	🕒 Generate Re	вро
errupt vectors									2
Type filter text				Ve	ctor Numb	er			~
Vector Number	Vector Table	Interrupt	Interrupt request source	Peripheral	Group	Priority	Status	Remarks	^
0	0x04C	IRQ0(PORT_IRQ0)	External pin interrupt 0	PORT	82)	Level 15 (low) 🗸	Used		
1	0x050	IRQ1(PORT_IRQ1)	External pin interrupt 1	PORT	(2)	Level 0 (high)	Used		
2	0x054	IRQ2(TAU0_ENDI2)	Interrupt when counting/ca	TAU0	2	Level 1			
3	0x058	IRQ3(TAU0_ENDI3)	Interrupt when counting/ca	TAU0	3	Level 2			
4	0x05C	IRQ4(TAU0_ENDI4)	Interrupt when counting/ca	TAU0	4	Level 4			
5	0x060	IRQ5(TAU0_ENDI5)	Interrupt when counting/ca	TAU0	5	Level 5			
6	0x064	IRQ6(TAU0_ENDI6)	Interrupt when counting/ca	TAU0	6	Level 6			
7	0x068	IRQ7(PORT_IRQ3)	External pin interrupt 3	PORT	7	Level 7	Used		
8	0x06C	IRQ8(SAU0_ENDI0)	UART0 transmission transfe	SAU0	0	Level 8			
9	0x070	IRQ9(SAU0_ENDI1)	UART0 reception transfer e	SAU0	1	Level 9 Level 10			
10	0x074	IRQ10(TAU0_MOD	Interrupt when counting/ca	TAU0	2	Level 11			
11	0x078	IRO11(TAU0 MOD	Interrupt when counting/ca	TAU0	3	Level 12			`
<						Level 13		>	
in Read Clear	Contract Contract	Disc.				Level 14			_
lew board Clock	cs   System   Comp	ponents Pins Interrup	t			Level 15 (low)			

Figure 4-49 Change the interrupt priority level



### 4.6.2 Interrupt Filter Feature

By specifying the filter range on the [Interrupts] page, user can find interrupt more easily.

rupt configu	ration					Generat	e Code	Generate R
rrupt vectors								
Type filter text					Vector Nur	nber		
Vector Number	Vector Table	Interrupt	Interrupt request source	Periphera	Vector Nun	nber		
0	0x04C	IRQ0(PORT_IRQ0)	External pin interrupt 0	PORT	Interrupt	le Address		
1	0x050	IRQ1(PORT_IRQ1)	External pin interrupt 1	PORT	Interrupt re	quest source		
2	0x054	IRQ2(TAU0_ENDI2)	Interrupt when counting/ca	TAU0	Peripheral			
3	0x058	IRQ3(TAU0_ENDI3)	Interrupt when counting/ca	TAUO	Group			
4	0x05C	IRQ4(TAU0_ENDI4)	Interrupt when counting/ca	TAU0	4	Level 15 (low)		
5	0x060	IRQ5(TAU0_ENDI5)	Interrupt when counting/ca	TAU0	5	Level 15 (low)		
6	0x064	IRQ6(TAU0_ENDI6)	Interrupt when counting/ca	TAU0	6	Level 15 (low)		
7	0x068	IRQ7(PORT_IRQ3)	External pin interrupt 3	PORT	7	Level 15 (low)	Used	
8	0x06C	IRQ8(SAU0_ENDI0)	UART0 transmission transfe	SAU0	0	Level 15 (low)		
9	0x070	IRQ9(SAU0_ENDI1)	UART0 reception transfer e	SAU0	1	Level 15 (low)		
10	0x074	IRQ10(TAU0_MOD	Interrupt when counting/ca	TAUO	2	Level 15 (low)		
< 11	0x078	IRO11/TAU0 MOD	Interrupt when counting/ca	TAU0	3	Level 15 (low)		>

Figure 4-50 Filter for [Interrupts] Page

#### 4.6.3 Changing Interrupt Vector or Group Setting

User can change the interrupt Vector or Group setting on the [Interrupts] page using the following procedure:

(1) Find the interrupt which you want to change Vector Number or Group setting on this page.

If the interrupt has been used, please start from step a.

If the interrupt hasn't been used, please start from step b.

a. Please release interrupt by right-clicking on a Code Generator configuration and then selecting the
 [ Generate code ] icon changes the icon to [ Generate code ] and disables code generation for the Code Generator configuration.

🔅 *Smart_Configurator_Example.scfg >	<		- 0
Software component configur	ration		🔂 🚵 Generate Code 🛛 Generate Report
Components 🕴 👌 🕀	Configure		^
type filter text	Comparator operation setting Stop	Operation	
> 🥪 Generic	<ul> <li>10 bits</li> </ul>	0 8 bits	🔿 12 bits
Contract Contrac	VREF(+) setting VCC	OAVREFP	O Internal reference voltage
> 🗁 I/O port > 🍃 A/D converter	VREF(-) setting VSS	OAVREFM	
(1) a.	✓ Generate code		
	Output only initialization API Change resource		
	<ul> <li>Remove</li> <li>Duplicate</li> <li>Rename</li> </ul>	v	
	Reset to default		~
Overview Board Clocks System Co	+ Add Configuration >		

Figure 4-51 Release used interrupt



- b. Change the filter to [interrupt].
- c. Input filter text (for e.g., ADC\_ENDI).
- (2) Find the Vector Number or Group you want to change.

Note: You cannot select the item which status is Used.

- (3) Click the [Interrupt] cell and select an Interrupt setting from the drop-down list.
- (4) When you use related components or enable code generation by clicking on the [ Generate code ] icon and change it to [ Generate code ], the interrupts with specified vector or group will be used.

戀 *Si	mart_Configurator	_Example.scfg >							- 8
Inte	rrupt configu	ration					Generate	Code G	ienerate Report
Int	errupt vectors								×.
(1) c.	ADC_ENDI				(1) <b>b</b> . Inte	rrupt			$\sim$
	Vector Number	Vector Table	Interrupt	Interrupt request source	Peripheral	Group	Priority	Status	Remarks
(2)	3	0x058	TAU0_ENDI3 v	Interrupt when counting/ca	TAU0	3	Level 15 (low)		
	7	0x068	IRQ3	ternal pin interrupt 3	PORT	7	Level 15 (low)	Used	
	11	0x078	IWDT_NMIUNDF	errupt when counting/ca	TAU0	3	Level 15 (low)		
	15	0x088	KEY_INTKR	D conversion end	ADC12	7	Level 15 (low)		
	19	0x098	SALIO ENDIS	RT1 reception transfer e	SAU0	3	Level 15 (low)		
	23	0x0A8	SAU0 INTSRE1		KINT	7	Level 15 (low)		
	27	0x0B8	TAU0_ENDI3	RT1 reception communic	SAU0	3	Level 15 (low)		
	31	0x0C8 (2)	TAU0_MODE8_ENDI	DT underflow	IWDT	7	Level 15 (low)		
		(3)	ADC_ENDI						
			DOC_DOPCI						
	<								>
Oven	view Board Clock	s System Comp	oonents Pins Interrup	ot					

Figure 4-52 Change Interrupt Vector Setting Example

#### 4.6.4 **Resolving Interrupt error**

Use IRQ5 as example, when IRQ5 is used in interrupt component, but an interrupt error is shown in "Configuration Problems" view. It means all interrupt vectors which can been used as IRQ5 have been occupied.

Software component configur	ation					Gene	rate Code	Generate Report	t
Components	IKQ4 setting							,	~
	IRQ4	Detection type	Falling edge	$\sim$	Digital filter	No filter	$\sim$		
🥫 🐩 🦆		Priority	Level 15 (low)	$\sim$					
type filter text	IRQ5 setting								
✓ ➢ Startup	IRQ5	Detection type	Falling edge	~	Digital filter	No filter	~		
r_bsp		Priority	Level 15 (low)	~					
✓ → Drivers Intervent	IRQ6 setting								
Config_ICU	IRQ6	Detection type	Falling edge	~	Digital filter	No filter	$\sim$		
< >	<							>	1
Overview Board Clocks System Comp	onents Pins Interrupt								
Configuration Problems $\times$									
6 errors, 0 warnings, 0 others									
Description	^					Туре			
🗸 😣 Interrupt (1 item)									
Section 2010/2011 Environment of the section of the	Config_ICU is not allocate	ed to any interrupt v	ector.			Interr	upt		

Figure 4-53 An interrupt is not allocated to any interrupt vector number



To solve this error, please release an occupied vector by other components and select a vector number for IRQ5 in [Interrupt] page using the following procedure:

- (1) Change the filter to [interrupt].
- (2) Input filter text (for e.g., IRQ5).
- (3) Find the Peripheral you want to release.
- (4) Right-clicking on ELC and then selecting the [ Generate code ] icon changes the icon to [ Generate code ] and disables code generation for the Code Generator configuration.
- (5) The interrupt of ELC has been released.
- (6) Click the [Interrupt] cell and select PORT\_IRQ5 from the drop-down list.
- (7) Generate Code again.

Inte	errupt configu	ration					당 Generate Code	칠 Generate Report
Int	errupt vectors				(1)			×
(2)	IRQ5				× Interru	pt		~
	Vector Number 5	Vector Table Address 0x060	Interrupt IRQ5(ELC_SWEVT	Interrupt request source Software event 1 (3)	Peripheral ELC	Group 5	Priority Level 15 (low)	Status Used
	<							>
Over	view Board Clock	s System Components	Pins Interrupt					



errupt vectors							
IRQ5				× Inter	rupt		
Vector Number	Vector Table Address	Interrupt	Interrupt request source	Peripheral	Group	Priority (5)	Status
5	0x060	ELC_SWEVT1	<ul> <li>Software event 1</li> </ul>	ELC	5	Level 15 (low)	
		IRQ5					
	(0)	PORT_IRQ5					
		RTCALM					
		IIC1 ENDI/IIC1 W	/UI				
		ELC SWEVT1					
		SAU0_ENDI1					
		SAU1_ENDI1					
		TALIO ENDIS					
		11100_211015					
<							

#### Figure 4-54 Release an occupied Interrupt

Note: If want to use ELC interrupt again, please start from step (1).



### 5. Managing Conflicts

When user add a component or configuring a pin or interrupt may cause problems in terms of resource conflict and missing dependency modules. This information will be displayed in the Configuration Problems view. User can refer to the displayed information to fix the conflict issues and generate code even if there are conflicts.

### 5.1 **Resource Conflicts**

When two software components are configured to use the same resource (for e.g., ADC), an error mark (19) will be displayed in the Components tree.

The Configuration Problems view will display messages on peripheral conflicts to inform user in which software configurations peripheral conflicts have been detected.

			- 0
Software component configuration		🗊 Generate Code	Generate Report
Components 🚵 🛃 📳 🕢 Com	figure		١
Image: Startup         V @ Startup         Config Port         Config ADC         Config ADC duplicate	s Interrupt		
🔝 Configuration Problems 🗙 📮 Console 🔗 Se	arch		7 8 🗖 🗖
6 errors, 0 warnings, 0 others			
Description	Туре		^
E04010005: Interrupt vector used by ADC	Interrupt		
8 E04010005: Interrupt vector used by ADC	Interrupt		
✓			
8 E04010001: Peripheral A/D Converter used	Peripheral		
6 E04010001: Peripheral A/D Converter used	Peripheral		
✓ Ø Pin (2 items)			
6 E04010003: Pin used by ANIO in Config_AL	Pin		
E04010003: Pin used by ANIO in Config_AL	Pin		v

Figure 5-1 Resource Conflicts



# 5.2 **Resolving Pin Conflicts**

If there is a pin conflict, an error mark 🔕 will appear on the tree and [Pin Function] list.

configuration						🐻 Generate Co	de Generate Rep
ftware Components	E ⊑ ↓ª₂ 品	Pin Functio	on			2	🛛   🚮   🗠 1
ype filter text		type filter	text (* = any s	tring, ? = any character)			All
🛩 💑 r_bsp		Enabled	Function	Assignment	Pin Number	Direction	Remarks
📦 r_bsp			IRQ0	Not assigned	Not assigned	None	
🗸 🚣 Ports			IRQ1	Not assigned	Not assigned	None	
© Config_PORT			🛞 IRQ2	P303/CLKOUT/TI04/TO04/SDAA0/RxD0,	/ 18	1	Multiple pin fur
A/D Converter			IRQ3	/ P201/IRQ3	17	1	
🛓 Interval Timer			IRQ4	Not assigned	Not assigned	None	
/ 🚠 Interrupt Controller			IRQ5	Not assigned	Not assigned	None	
Onfig_ICU			IRQ6	Not assigned	Not assigned	None	
			IRQ7	Not assigned	Not assigned	None	
			NMI	Not assigned	Not assigned	None	
		<					

Figure 5-2 Pin Conflicts

Detailed information regarding conflicts is displayed in the Configuration Problems view.

Sconfiguration Problems X 🔗 Search	
3 errors, 0 warnings, 0 others	
Description	Туре
V 🔞 Pin (3 items)	
8 E04010003: Pin used by IRQ2 in Config_ICU conflicts with pin used by P303 in Config_PORT, pin used by P303 in Pin Allocator.	Pin
8 E04010003: Pin used by P303 in Config_PORT conflicts with pin used by IRQ2 in Config_ICU, pin used by IRQ2 in Pin Allocator.	Pin
8 E05000010: Pin 18 cannot be used multiple times. Pin 18 is assigned to P303 and IRQ2.	Pin

Figure 5-3 Pin Conflict Messages

To resolve a conflict, right-click on the node with an error mark on the tree and select [Resolve conflict].

Pin configuration	
Software Components	🕀 🖃 🖓 😹 Pi
Type filter text	t
<ul> <li>✓ ▲ r_bsp</li> <li>✓ r_bsp</li> <li>✓ ▲ Ports</li> <li>✓ Config_PORT</li> <li>&gt; ▲ A/D Converter</li> <li>&gt; ▲ Interval Timer</li> <li>✓ ▲ Interval Controller</li> </ul>	
Config_ICU	Assign all Unassign all Resolve conflict
Pin Function Pin Number	
Overview Board Clocks System	Components Pins Interrupt

Figure 5-4 Resolving Pin Conflicts

The pin function of the selected nodes will be re-assigned to other pins.



### 6. Generating Source Code

Source generation can be generated even if there is a conflict in the Configuration Problems view.

### 6.1 Generating Source Code File

Output a source file for the configured details by clicking on the [ Generate Code ] button in the Smart Configurator view.

🔅 *Smart_Configurator_Example.scfg 2	3	
Software component configu	ation	Generate Code Generate Report
Components 👌 🛓	Configure	^
हा स्ट्र	Port selection PORT2	_
<ul> <li>✓ ➢ Startup</li> <li>▲ ➢ Generic</li> <li>▲ p r_bsp</li> <li>▲ ➢ Driver</li> </ul>	Apply to all Unused In Out	□ ° <b>~</b> >
Overview Board Clocks System Com	ponents Pins Interrupt	

Figure 6-1 Generating a Source File

The Smart Configurator generates the source files in <ProjectDir>\src\smc\_gen and IAR / SEGGER Embedded Studio related files in save location (refer to 3.3.1 Creating a New Smart Configurator Configuration File). If user's Smart Configurator has already generated a file, a backup copy of that file is also generated (refer to the chapter 9 Backing up Generated Source Code).



## 6.2 **Configuration of Generated Files and File Names**

Figure 6-2 Configuration of Generated Files and File Names, shows the folders and files output by the Smart Configurator. Function main () is included in main.c which is generated when clicking "Generated Code" in the Smart Configurator.

*r\_xxx* indicates the names of Software Integration System Modules, "*ConfigName*" indicates the name of the configuration formed by the component settings and "*ProjectName*" indicates a project name set in the Smart Configurator.



Figure 6-2 Configuration of Generated Files and File Names (IAREW)





Figure 6-3 Configuration of Generated Files and File Names (SEGGER Embedded Studio)

Note: The generated files and file name under smc\_gen folder is same for IAREW and SEGGER Embedded Studio.



Folder	File	Description
Workspace	-	This folder is not generated by Smart Configurator, but is
		the location that user selects to create Smart Configurator
		project.
		All folders and files generated by Smart Configurator will be
		generated into this folder.
	{ <i>ProjectName</i> }.eww	This file generates once only in the first code generation.
		{ <i>ProjectName</i> }.ewp file path is specified in this file. It is the workspace file for IAREW.
	{ <i>ProjectName</i> }.ewp	This file generates once only in the first code generation.
		It appends the "buildinfo.ipcf" and "main.c" files at the end of
		this file. It is the project file for IAREW.
	{ <i>ProjectName</i> }.ewd	This file generates once only in the first code generation.
		It is totally same as the default *.ewd file generated by
		IAREW. It is the debug file for IAREW.
	main.c	I his file generates once only in the first code generation.
		It contains function main ().
	buildinfo.ipcf	This file is generated when clicking button "Generate Code" in Smart Configurator each time. It is the connection file for IAREW.
		It includes the path for all files generated for each
		component by Smart Configurator
	{ProjectName}.emProject	This file is generated when clicking button "Generate Code"
		in Smart Configurator each time. It is the project file for
		SEGGER Embedded Studio.
	SEGGER_Flash_RV32.icf	This file generates once only in the first code generation.
		It is the linker script file for SEGGER Embedded Studio
		section placement file.
	"devicename"_MemoryMap.xml	This file generates once only in the first code generation.
		It is the memory map file for SEGGER Embedded Studio.
general	-	This folder is always generated.
		It contains header files and source files commonly used by
	L (Noto*1)	Code Generator drivers of the same peripheral function.
		The files contain macro definitions for setting SFR registers.
	r_smc_entry.h	This file is always generated.
		I his file includes the header files of Code Generator drivers
		that are added to the project.
		files added by user including this file is necessary
	r. cg. macrodriver h	This file is always generated
		This header file contains common macro definitions used in
		drivers.
	r ca userdefine.h	This file is always generated.
		User can add macro definitions in the dedicated user code
		areas.
	r_cg_systeminit.c	This file is always generated.
		This file contains all component's Create () function. it is
		used for peripheral modules initialization.
	r_cg_xxx_common_user.c <sup>(Note*1)</sup>	The files contain common interrupt API of used peripherals.
	r_cg_xxx_common.c <sup>(Note*1)</sup>	This file is generated when related peripherals are used.
	r_cg_xxx_common.h <sup>(Note*1)</sup>	This file is generated when related peripherals are used.



r_cg_interrupt_handlers.h	This file includes all interrupt routine declaration. If no configuration created, all interrupt routine is default routine; if specific configuration is created, corresponding interrupt routine declaration of this configuration will replace the default routine declaration.
r_cg_inthandler.c	This file includes all default interrupt routine definition.
r_cg_vect_table.c	This file includes an interrupt vector table which includes all interrupt routine entry address. if no configuration created, all interrupt routine entry address is default; if specific configuration is created, corresponding interrupt routine entry address of this configuration will replace the default routine entry address.
r_smc_cgc.c	This file is always generated. It contains the initialization of clock sources other than the clock source selected in the [Clocks] page.
r_smc_cgc.h	This file is always generated. This header file contains macro definitions to initialize clocks other than the selected clock source.



Folder	File	Description
general	r_smc_cgc_user.c	This file contains functions to be added to R_CGC_Create.
		User can add codes and functions in the dedicated user code
		areas.
	r_smc_interrupt.c	This file is always generated.
		It contains interrupt initialization (depending on hardware
		specification).
	r_smc_interrupt.h	This file is always generated.
		It contains the priority level of all interrupts that are
		configured in the [Interrupts] tabbed page. User can use
		these macro definitions in application codes.
r_bsp		I his folder is always generated.
		It consists of multiple subfolders (board, doc, mcu) with:
		- Initialization codes to start up the MCU before entering
		Definitions of all SEP registers in ideofine h (mou folder)
		- Application note of r hsp (doc folder)
		It also contains platform h that will include r hsp h of the
		device used in the project.
r config		This folder is always generated.
		It contains configuration header files for the MCU package,
		clocks, interrupts, and RISC-V MCU Software Integration
		System drivers/middleware.
	r_bsp_config.h	This file is always generated.
		It contains configurations of r_bsp for clock initialization and
		other MCU related settings. Some MCU related settings are
		generated by Smart Configurator (e.g. package type) and
		other settings (e.g. stack size) are configured by user
	r han interrunt configh	This file is always generated
	L_psp_interrupt_coning.in	It contains mapping of the software configurable interrupts
		(depending on hardware specification)
r pincfa	Pin h	This file is always generated
1_pinoig		It is generated for supporting pin symbol and included in
		smc entry.h.
	Pin.c	This file is always generated.
		It is a reference of pin function initialization for all peripherals
		configured in the [Pins] tabbed page (except I/O Ports).
{ConfigName}		This folder is generated for the Code Generator drivers that
		are added to the project.
		API functions in this folder are named after the ConfigName
		(configuration name).
	{ConfigName}.c	This file contains functions to initialize driver
		(R_ConfigName_Create) and perform operations that are
		(R_ConfigName_Stop)
	ConfigName user c	This file contains interrupt service routines and functions for
		user to add code after the driver initialization
		(R ConfigName Create).
		User can add codes and functions in the dedicated user code
		areas.
	{ConfigName}.h	This is header file for {ConfigName}.c and
		{ConfigName}_user.c.

Note \*1: *xxx* is the name of a peripheral function.



# 6.3 Initializing Clocks

Configurations of the clock source selected in the [Clocks] page are generated to the macros in the r\_bsp\_config.h file located in \src\smc\_gen\r\_config folder. Clock initialization codes will be handled by r\_bsp before entering main ().

The r\_bsp\_config.h file also contains other MCU related settings (for e.g., package, stack size).

Clocks configuration						×	mple.scfg $ imes$	onfigurator_Exam
	de Generate Repo	🕤 Generate Code						onfiguration
Operation mode:       High-speed mode         Type anding:       Type and tools Type!         Vick words:       AV x wOC x 55 V								
						i mode v	High-speed mod	Operation mode:
						Ŧ	Type A	Type setting:
Image:						C≰5.5V +	4.0 V ≤ VCC ≤ 5.	VCC setting:
Sed-tod sciller   Sed-tod sciller <td></td> <td>System clock (/CLK)</td> <td>SCKDIVCR(ICK(2:0])</td> <td></td> <td></td> <td></td> <td>input</td> <td>External clock in</td>		System clock (/CLK)	SCKDIVCR(ICK(2:0])				input	External clock in
Webckst caller       Pelphrakt model       P		- 32 (MHz)	x1 *		•	(MHz)	20	Frequency:
Operation mode:		Peripheral module clock(PCLKB)	SCKDIVCR(PCKB(2-0))	•			illator	Sub-clock oscil
Megin Clest:       Image: Cleater         Image: Cleater       Image: Cleat		34 ((0H2)				Mode *		Operation mode: Frequency:
Image: scalabor       Image: scalabor				•	•	Current 👻	Normal Curre	Margin Check:
united #ACCC decisition after reset:         Frequency:       2         @ Middle-speed on-chip socilator         Frequency:       8         @ Middle-speed on-chip socilator         Prequency:       8         @ Middle-speed on-chip socilator         Prequency:       82.768         @ Middle-speed on-chip socilator         Prequency:       82.768         @ Middle-speed on-chip socilator         Prequency:       82.768         @ Middle-speed on-chip socilator         Prequency:       15		Clock/Buzzer output(CLKOUT) (MHz)	CKOCR(CKODIV[2:0])	•			n-chip oscillator	High-speed on
Image: Control on-chip collistor       Control on-chip collistor         Prequency:       8       0.4H2         Image: Control on-chip collistor       Control on-chip collistor         Image: Control on-chip collistor       Contron-chip collistor					•	* (MHz)	32	Frequency:
An index     CalcoccurulationCCURULATION		CACMCLK/TML32MCLK/UARTAMCLK					on-chin oscillator	V Miridie-sneed c
CACHCK/TM23HLK/LARTAHLK     32     0HB       Image: Cachck/TM23HLK/LARTAHLK/LART		ACMOCLK/TML32MOCLK/UARTAMOCLK     B (MHz)				(MHz)	8	Frequency:
Mon-speed on-chip solilator         CACALKMTOX           Prequeng:         12.768         0440           CACSOLK         CACSOLK           MDT-dedicated on-chip solilator         0440           MDT-dedicated on-chip solilator         0440           Frequeng:         15         0440		CACHCLK/TML32HCLK/UARTAHCLK 32 (MHz)						
Frequency:         32.764         (AHd)         22.764         (AHd)           CACSCIX:         CACSCIX:         06H0         7ML33CLX/PRMICELX/REMICELX		CACLCLK/MTCLK					-chip oscillator	Low-speed on-
MDT-dedicated on-chip socillator         NMDT-dedicated on-chip socillator           Frequency:         15           (642)         15		- 32.768 (KHz) CACSOLK				(kHz)	32.768	Frequency:
MDT-dedicated on-chip cecilitor  Frequency:  15  (kHe)  KDT-LK/CACL/CLK  15  KHe)  KDT-LK/CACL/CLK  KDT-LKK  KDT-LK/CACL/CLK  KDT-LK/CACL/CLK KDT-LK/CACL/CLK  KDT-LK/CACL/CLK  KDT-LK/CACL/CLK  KDT-LK/CACL/CLK  KDT-LK/CACL/CLK  KDT-LK/CACL/CLK	IK .	TML32LCIK/TML32SCLK/UARTALCIK/UARTASCLK/REMCLCIK/REMCSCLK/RTCCLK						
Frequency: 15 (kHz) 15 (kHz)		IWDTCLK/CACILCLK			L		n-chip oscillator	IWDT-dedicated on
		- 15 (kHz)				(kHz)	15	Frequency:
Enternal clock input for cTAG xTBG down TOYO		e TAG elsek (TCVC)					ut for cJTAG	External clock input
Frequency: 625 (MHz) 625 (MHz)		6.25 (MHz)				(MHz)	6.25	Frequency:

	-																
-	∋/*	Syst	em Cl	ock D	ivisi	on C	ontr	ol	Register	<u>د (۶</u>	SCKDIV	CR)					
		Peri	phera	1 Mod	lule C	lock	B (	PCL	.KB) Sele	ct	(PCKB	[2:0	9])				
	0	: x	1/1														
	1	: x	1/2	_													
	2	: x	1/4														
	3	: x	1/8														
	4	: x	1/16														
	5	: x	1/32														
	6	: x	1/64														
	*1						_										
	#d	efine	BSP	CFG P	CLKB	DIV	(0)	/*	Generate	d v	/alue.	Do	not	edit	this	manuall	v */

#### Figure 6-3 Clocks Configuration and Generated Code in r\_bsp\_config.h

Folder	File	Macros/Functions	Description
r_config	r_bsp_config.h	Macros related to clocks	These settings are generated by Smart Configurator based on user's selection in the [Clocks] page for the clock source. <i>r_bsp</i> will handle the clock initialization before entering <i>main</i> ().
		Macros related to MCU settings	Some MCU related settings are generated by Smart Configurator (e.g. package type) macros. For the detail macro information, user can refer to the application note in <i>r</i> bsp folder: \src\smc_gen\r_bsp\doc

Note: r\_bsp\_config.h will be backed up to trash folder before each code generation (refer to chapter 9 Backing up Generated Source Code).



## 6.4 Initializing Pins

Configurations in the [Pins] page are generated in some source files depending on driver's requirements and hardware specifications.

(1) <u>Pin initialization for drivers with {ConfigName}</u>

Pin functions are initialized in R\_ConfigName\_Create of the file \src\smc\_gen\{ConfigName}\(ConfigName).c.

Pin initialization codes will be handled before entering main ().

in configuration				Gen	erate Code G	enerate Repo
Software Compon 🕀 🖻 🛱 🔜	Pin Functi	on				<b>[]</b>   24 23
Type filter text	type filter	text (* = any s	tring, ? = any character)		All	~
<ul> <li>✓ ▲ r_bsp</li> <li>● r_bsp</li> <li>✓ ♣ Square Wave Output</li> <li>✔ Config TAU0 1</li> </ul>	Enabled	Function TO01	Assignment P102/TI01/TO01/SCLA0/SI20/SDA20/Rx	Pin Number / 29	Direction O	Remarks
n Function Pin Number	<					>

Figure 6-4 *Config\_TAU0\_1* in Software Components View

Folder	File	Function	Component type	Description
{ConfigName}	{ConfigName}.c	R_ConfigName_ Create	Code Generator	This API function initializes the pins used by this driver. r_cg_systeminit will call this function before entering main () function.

#### (2) Reference to pin initialization codes

Refer to *Pin.c* in \src\smc\_gen\r\_pincfg folder for all peripheral pin functions used in the project (except I/O ports).

Folder	File	Function	Driver	Description
r_pincfg	Pin.c	R_Pins_Create	-	This file contains the initialization codes of all pin functions configured in the [Pins] page except I/O ports.

## 6.5 Initializing Interrupts

Configurations in the [Interrupts] page are generated in some source files. Interrupt functions are initialized in R\_*ConfigName*\_Create of the file \src\smc\_gen\{*ConfigName*}\c.



terrupt co	nfigurat	tion				Generate Code (	ienerate R	lep
terrupt vec	tors							E
Type filter te	ext		X	/ector Number	ř			~
Vector N	Vecto	Interrupt	Interrupt request source	Peripheral	Group	Priority	Status	^
0	0x04C	IRQ0(PORT_IRQ0)	External pin interrupt 0	PORT	0	Level 15 (low)		-
1	0x050	IRQ1(TAU0_ENDI1)	Interrupt when counting/capturing is completed	TAUO	1	Level 15 (low)	Used	
2	0x054	IRQ2(TAU0_ENDI2)	Interrupt when counting/capturing is completed	TAUO	2	Level 15 (low)		
3	0x058	IRQ3(TAU0_ENDI3)	Interrupt when counting/capturing is completed	TAUO	3	Level 15 (low)		
4	0x05C	IRQ4(TAU0_ENDI4)	Interrupt when counting/capturing is completed	TAUO	4	Level 15 (low)		
< 5	0x060	IRO5(TAU0 ENDI5)	Interrupt when countina/capturina is completed	TAU0	5	Level 15 (low)	>	~

Figure 6-5 Interrupts Configuration in Interrupts View

ltem	Folder	File	Component type	Description
Priority	{ConfigName}	{ConfigNa me}.c	Code Generator	It is initialized in R_ <i>ConfigName</i> _Create of this file. <i>R_systeminit</i> in <i>r_cg_systeminit.c</i> will call this function before entering <i>main ()</i> function.

# 7. Loading Generated Files in Integrated Development Environment

Load source code outputted by Smart Configurator on Integrated Development Environment Platform.

## 7.1 Loading in IAR Embedded Workbench

When IAR environment is selected for the compiler to be used, Smart Configurator outputs the related files (.eww/.ewp/.ewd/main.c) together with the source file. It is not necessary for the user to create project files in IAR Embedded Workbench.

The usage procedure is as follows.

- (1) Select [Open Workspace...] from the [File] menu of IAR Embedded Workbench.
- (2) In the [Open Workspace] dialog box, browse to the folder where the project file is saved, select the project file (.eww), and click the [Open] button.

Open Workspace							×
← → • ↑ <mark> </mark>	≪ IAR ≯ S	Smart_Configurator_Example	~	ō	<u>,</u>	Search Smart_Conf	ïgurator
Organize 🔻 New	v folder						
📃 Desktop	^	Name	Date	modifie	d	Туре	Size
Documents		.settings	10/10	/2022 14	4:58	File folder	
🕂 Downloads	- 11	Debug	10/10	/2022 17	7:03	File folder	
👌 Music		settings	11/10	/2022 09	9:11	File folder	
Pictures		src	23/09	/2022 13	3:49	File folder	
📑 Videos		Smart_Configurator_Example.eww	23/09	/2022 13	3:49	IAR IDE Workspace	: 1
C:) Windows	<b>v</b> -	<					>
	File name:	Smart_Configurator_Example.eww		~	Wo	rkspace Files (*.eww)	~
						Open (	Cancel

Figure 7-1 Load a \*.eww File



(3) The source file output by the Smart Configurator is added to the IAR C project workspace.

Workspace	<b>▼</b> ‡ ×
Debug	~
Files	¢
🖻 🌒 SC_Example - Debug *	~
🛛 🛏 🛱 🖬 Renesas_SC	
🔄 🦵 🛋 smc_gen	
🛛 🛛 🛏 🖬 Config_ADC	
🛛 🕂 🛋 general	
-⊞ 🛋 r_bsp	
-⊞ 🖬 r_config	
🗆 🖃 🛋 r_pincfg	
🛛 🛏 🖹 buildinfo.ipcf	
- 🛨 🗟 main.c	
🖵 🖬 Output	



- (4) Select [Options...] from the [Project] menu of IAR Embedded Workbench.
- (5) In the [Options for node "ProjectName"] dialog box, change the target device to match with the target device selected when creating Smart Configurator's configuration file.

ategory: eneral Options tatic Analysis			
C/C++ Compiler Assembler	Library Configuration	Library Ontions 1	Library Ontions 2
Output Converter	Target ISA Extension	s Code Generatio	on Output
Linker Build Actions Debugger GDB Server	Device R9A02G021	<b>D</b> +	Base ISA
I-jet Simulator Third-Party Driver	Standard extensions	Float	RV64I ing-point None ~
	Bit manipulation ☑ Zba  ☑ Zbb    Zbc	Zbs Xbcountzero	es
	Code size reduction	Scalar cryptogra	iphy
	Zcb	Zkn	
	Zcmp	Zks	

Figure 7-3 Change Target Device



## 7.2 Loading in SEGGER Embedded Studio

When SEGGER environment is selected for the compiler to be used, Smart Configurator outputs the related files (.emProject/Linker script file/Memory map file/main.c) together with the source file. It is not necessary for the user to create project files in SEGGER Embedded Studio.

The usage procedure is as follows.

- (1) Select [Open Solution...] from the [File] menu of SEGGER Embedded Studio.
- (2) In the [Open Solution] dialog box, browse to the folder where the project file is saved, select the project file (.emProject), and click the [Open] button.

Open Solution						;
← → • ↑ 🔤 « Do	cuments > SEGGER Embedded Stu	dio Projects > SEGGER_Blinky	~ 1	ර් Search	SEGGER_Blinky	م
Organize 🔻 New folde	r					
OneDrive - Renesa ^	Name	Date modified	Туре	Size		
My PC: VNM-5CG	.settings	4/1/2024 9:23 AM	File folder			
	- Output	4/1/2024 9:24 AM	File folder			
J 3D Objects	src	4/1/2024 9:23 AM	File folder			
Desktop	SEGGER_Blinky.emProject	4/1/2024 9:23 AM	SEGGER Embedde	7 KB		
😫 Documents						
👆 Downloads						
👌 Music						
E Pictures						
Videos						
🏪 (C:) OS						
🗙 (N:) MobAP2 (\\						
🛫 (Y:) MobAP2 (\\ı						
🗙 (Z:) DTV (\\rvc-v						
🔿 Network 🗸 🗸						
File na	ame: SEGGER_Blinky.emProject			~ Soluti	on Files (*.emProje	ct*.em ∨
					lpen C	ancel

Figure 7-1 Load a \*.emProject File

(3) The source file output by the Smart Configurator is added to the SEGGER Embedded Studio project workspace.

Project Explorer
🗘 Debug 🔹 🔹 🔛
Project Items
Solution 'Smart_Configuration_Example'
Project 'Smart_Configuration_Example'
▲ Source Files 45 files, modified options
Renesas_SC 44 files
Config_PORT 3 files
general 13 files
▷ 🗎 r_bsp 24 files
Image: Price of the second
i r_pincfg 2 files
🖹 main.c

Figure 7-2 New Files Added to SEGGER Embedded Studio Workspace



## 7.3 Build IAR Project File

After loading Smart Configurator project file to IAR Embedded Workbench successfully, user can right-click on project name, select [Rebuild All] from context manual, then build operation will be executed successfully.

Workspace	▼ ‡ × main.c ×
Debug	~
Files	🌣 • ^
Smart_Configurator_Example	
- 🖓 🛋 Renesas_SC	Options rn 0;
smc_gen smc_gen smc_gen scheduler smc_gen scheduler sche	Make       Compile       Rebuild All       Clean       C-STAT Static Analysis       Stop Build       Add
- 🕀 🛑 mcu	Remove
platform.h	<u>R</u> ename
Smart_Configurator_Example	Version Control System
Build Messages Linking	Open Containing Folder File Properties Set as Active
Total number of errors: 0 Total number of warnings: 0	
Build Debug Log	

Figure 7-4 Build C Project File in IAR

## 7.4 Build SEGGER Embedded Studio Project File

After loading Smart Configurator project file to SEGGER Embedded Studio successfully, user can right-click on project name, select [Build] from context manual, then build operation will be executed successfully.

-							
Project Explorer	🔄 🗙 mair	n.c					×
🗘 Debug 🔹 🔁 🔛	€ 🗘 🕆 👚	$\rightarrow$					) 👔
Project Items	Code Data+RO	29 30 int ma Alt+Return	<pre>ain(void) ile(1) /* Toggle LED status PIN_WRITE(LED2) = ~PI /* Delay 500 millisec R_BSP_SoftwareDelay(5 turn 0;</pre>	<pre>*/ N_READ(LED2); onds before returning */ e0, BSP_DELAY_MILLISECS);</pre>	:		^
	<ul> <li>Add New File</li> <li>Add Existing File</li> <li>Add Existing File</li> <li>New Folder</li> <li>Open Solution in Editor Import Package Files Debug</li> <li>Dependencies</li> </ul>	Ctrl+P, O					
	X Cut	Ctrl+X					~
	Paste	Ctrl+V					> ÷
	Remove	Curre	- 🌂 🀐 Tasks -				©AA× ¢
	Edit Linker Script Edit Memory Map	Ctrl+P, L	onfiguration_Example' from sole	ution 'Smart_Configuration_Examp	le' in configuration 'Debug'	19 targets in 3.0s 6 targets/s	ок
	Import Memory Map						ок
	Open Linker Map File	Ctrl+P, M	FLASH2	RAM1	RAM2		

Figure 7-4 Build Project File in SEGGER Embedded Studio



### 8. Creating User Programs

The Smart Configurator can add custom code to the output source files. This chapter describes how to add custom code to the source file generated by the Smart Configurator. Please follow the procedure in , to create an IAR project file first and make configuration setup, build and run the project.

## 8.1 Adding Custom Code

When [Code Generator] or [Graphical Configurator] is selected as the component type, if files which have the same name already exist, new code will be merged only with the existing code that is between the comments below.

/\* Start user code for xxxx. Do not edit comment generated here \*/

/\* End user code. Do not edit comment generated here \*/

In the case of [Code Generator], three files are generated for each of the specified peripheral functions. The file names are "Config\_xxx.h", "Config\_xxx.c", and "Config\_xxx\_user.c" as the default, with "xxx" representing the name of the peripheral module. For example, "xxx" will be "ADC" for the A/D Converter (resource ADC). The comments to indicate where to add custom code are at the start and end of \*.c files, and at the end of \*.h file. Comments to indicate where to add user code are also added to the interrupt function for the peripheral module corresponding to Config\_xxx\_user.c. The following example is for ADC (Config\_ADC\_user.c).



```
Pragma directive
  /* Start user code for pragma. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */
.
****************
Includes
    #include "r_cg_macrodriver.h"
#include "Config_ADC.h"
/* Start user code for include. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */
#include "r_cg_userdefine.h"
*****
Global variables and functions
/* Start user code for global. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */
*****
* Function Name: R_Config_ADC_Create_UserInit
* Description : This function adds user code after initializing the AD converter.
* Arguments : None
* Return Value : None
void R_Config_ADC_Create_UserInit(void)
{
  /* Start user code for user init. Do not edit comment generated here */
  /* End user code. Do not edit comment generated here */
}
*************
* Function Name: r_Config_ADC_interrupt
* Description : This function is ADC_ENDI interrupt service routine.
* Arguments : None
* Return Value : None
void r_Config_ADC_interrupt(void)
{
  /* Start user code for r_Config_ADC_interrupt. Do not edit comment generated here
*/
  /* End user code. Do not edit comment generated here */
  R_ICU->IELSR15 &= 0xFFFEFFFU; /* clear ADC_ENDI interrupt flag */
}
/* Start user code for adding. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */
```



## 8.2 Using Generated Code in User Application

To use the generated code of RISC-V MCU Software Integration System Modules and Code Generator, follow the below steps:

Driver initialization functions (R\_ConfigName\_Create) including initialization of pins have been called in  $R_Systeminit$  function of  $r_cg_systeminit.c$  by default. User just need to add application codes to perform operations that are driver-specific, for e.g., start ( $R_ConfigName_Start$ ) and stop ( $R_ConfigName_Stop$ ).



Figure 8-1 Call Code Generator Functions

# 9. Backing up Generated Source Code

The Smart Configurator has a function for backing up the source code at:

<ProjectDir>\trash\<Date-and-Time>

The Smart Configurator generates a backup folder for the previously generated source code when new code

is generated by clicking on the [Generate Code] button. <Date-and-Time> indicates the date and time when the backup folder is created after code generation.



# 10. How to make debug connections via J-Link OB on the FPB Board

Connecting the PC and the target Board by using a J-Link OB.



Figure 10-1 Connecting the PC and the target Board by using a J-Link OB

Use IAREW GDB server debugger as an example:

1. Create a Blinky project of IAR toolchain. Select the GDB server debugger in project's Options:

Category:	Factory Settings
General Options Static Analysis C/C++ Compiler Assembler	Setup Download Images Multicore Extra Options Plugins
Output Converter Custom Build	Driver: I Run to:
Linker Build Actions	GDB Server ~ main
I-jet Simulator Third-Party Driver	Use macro file:
	\$TOOLKIT_DIR\$\config\debugger\Renesas\R9A02G021.dc

Figure 10-2 Using GDB server debugger

2. Ensure that project is configured to use the IAREW flash loader.

Category: General Options Static Analysis		Factory Settings
C/C++ Compiler Assembler Output Converter Custom Build	Setup Download Images Multicore Extra Options Plug	ins
Linker Build Actions Debugger GDB Server I-jet	Curry Gardinad     Suppress download     ✓ Use flash loader(s)     Override default.board file	
Third-Party Driver	\$TOOLKIT_DIR\$\config\flashloader\Renesas\FlashR9A0.	

Figure 10-3 Using IAR flash loader



Using "localhost,2331" for TCP/IP address or hostname of GDB server. 3.

Options for node "blinky"		$\times$
Category:	Factory Settin	IQS
General Options Static Analysis C/C++ Compiler Assembler	Setup Breakpoints	
Output Converter Custom Build Linker Build Actions Debugger <b>CDB Server</b> I-jet Simulator Third-Party Driver	TCP/IP address or hostname [,port] localhost,2331	
	Log communication     \$PROJ_DIR\$\cspycomm.log	
	OK Cancel	

Figure 10-4 Set TCP/IP address or hostname

4. Open Command Prompt, input JLinkGDBServerCL.exe file location, device, and port information: "C:\Program Files\SEGGER\JLink\_V794\JLinkGDBServerCL.exe" -device risc-v -if cjtag -port 2331.

C:\Users\yang-xiaolin>"C:\Prog SEGGER J-Link GDB Server V7.94	ram Files\SEGGER\JLink_V794\JLinkGDBServerCL.exe" Command Line Version	-device risc-v -if	cjtag -port 2331
JLinkARM.dll V7.94 (DLL compile	ed Nov 29 2023 13:41:24)		
Command line: -device risc-v - GDB Server start settings-	if cjtag -port 2331 		
GDBInit file: GDB Server Listening port:	none 2331		
SWO raw output listening port: Terminal I/O port:	2332 2333		
Accept remote connection:	localhost only		
Generate logfile: Verify download:	off		
Init regs on start:	off		
Silent mode:	off		
Target connection timeout:	0 ms		
J-Link related settings			
J-Link Host interface: I-Link script:	USB		
J-Link settings file:	none		
Target related settings			
Target device: Target device parameters:	r1sc-v none		
Target interface:	cJTAG		
Target interface speed:	4000kHz		
larget endran.	111111		
Connecting to J-Link			
J-LINK is connected. Firmware: I-Link OB-RA4M2 compi	iled Oct 30 2023 12:13:20		
Hardware: V1.00			
S/N: 1086999010 Chacking target voltage			
Target voltage: 3.30 V			
Listening on TCP/IP port 2331			
Connecting to target			
J-Link found 1 JTAG device, Tot	tal IRLen = 5		
Halting core			
RISC-V RV32 detected. Using RV3	32 register set for communication with GDB		
Core implements no FPU Connected to target			
Waiting for GDB connection			

Figure 10-5 Set .exe file location/device/port information in Command Prompt

Note: J-Link version V7.94 or later.



### 5. Click on [Download and Debug] in IAREW:

File Edit View Project GDB Server	Tools Window Help	
: 🗅 🗅 🕒 🕋 🔚 🕹 🖒 🗋	5 C	
Workspace 🔻 🕂 🗙	main.c x	Download and Debug (Ctrl+D)
Debug ~		Download the application and start
Files 🗢	/************************************	*

#### Figure 10-6 Click on [Download and Debug] in IAREW

6. The code can be downloaded successfully and start from main() function :

File Edit View Project Debug	GDB Serve	r Tools	Window Help		
i 🗅 🗅 🕒 🕋 🖴 🕹 🖄 🖆 🗇	50	4	- < Q > ⇆ HE < 📮 > 🕢 🖻 🔳 🌒 🖷	G	े 🛛 📑 🕞 न न 🖻 में में 🕨 🔲
Workspace	<b>▼</b> ₽ ×	main.c	×	Di	sassembly 🗸 🕈 🗸
Debug	~	main()	fo		Go to:
Files □ ● blinky - Debug □ ● blinky - Debug □ ■ Buildinto.ipcf □ ■ main.c □ ■ Output ■ Output blinky Debug Log Log ▲ Mon Apr 08, 2024 16:57:30: Usi Mon Apr 08, 2024 16:57:30: Usi	ng non-st ded deb	andard-d	<pre>of this software. By using this software, you agree to the ac folLowing link: http://www.renesas.com/disclaimer Copyright (C) 2024 Renesas Electronics Corporation. All right File Name : moin.c Description : Main Program Creation Date: 2024-04-08 This file was generated by Smart Configurator. include "r_smc_entry.h" nt main(void) while(1) {     /* Toggle LED status */     PIN_WRITE(LED2) = ~PIN_READ(LED2);     /* Delay 500 milliseconds before returning */     R_BSP_SoftwareDelay(500, BSP_DELAY_MILLISECS); } compliant entry point 0, from the ELF file. Use the '-suppress_entrypoint_wam Userstyang-xiaolint_smartconfigurator/workspace/20240408(Debug(Exekblink) </pre>	ing'	Disassembly uint32_t vector = 0x20 0xd7c: 0x2000'4537 asm( "add t0, zerc, X0" 0xd80: 0x00a0'02b3 asm( "csrv 0x307, t0" ) 0xd84: 0x3072'9073 ): 0xd88: 0x8082 0xd8a: 0x0000 int main(void) { main: 0xd8e: 0xc606 PIN_WRITE(LED2) = ~ ?main_0: 0xd90: 0x4004'0537 0xd94: 0x0245'0513 0xd98: 0x4108 0xd92: 0x0015'7613 0xd92: 0x0015'7613 0xd92: 0x0015'7613 0xd92: 0x0015'7613 0xd92: 0x0015'7613 0xd92: 0x0025'0513 0xda6: 0x0205'0513 0xda6: 0x205'0513 0xda8: 0x410c
Mon Apr 08, 2024 16:57:30: Loc Mon Apr 08, 2024 16:57:30: Do Mon Apr 08, 2024 16:57:30: Th Mon Apr 08, 2024 16:57:30: Th Build Debug Log	aded deb wnload co get reset ere was 1	ugee: C.ª ompleted warning	Users\vang-xiaolin\smartconfigurator\workspace\20240408\Debug\Exe\blink I. during the initialization of the debugging session.	ky.o	ut v

Figure 10-7 The code can be downloaded successfully



### 11. Generating Reports

The Smart Configurator can output the configuration information of the project to the report. Follow the procedure below to generate a report.

# 11.1 Report on All Configurations (PDF or Text File)

A report is output in response to clicking on the [Generate Report] (Generate Report)] button in the Smart Configurator view.



Figure 11-1 Output of a Report on the Configuration (as a PDF/Text File)

Smart Report			×
Generate report of configurations			
Options			
Print all sections			
O Print specific sections			
Board			^
System			
> 🗸 Components			
> Pins			
Output as PDF		 Selec	t Font
Output as text			
C:\		Bro	wse
[	ОК	Cancel	

Figure 11-2 Dialog Box for Output of a Report (Example is selecting "Output as PDF")



## 11.2 **Configuration of Pin Function List and Pin Number List (in csv Format)**

A list of the configuration of pin functions and pin numbers (whichever is selected at the time) is output in response to clicking on the [ 🔜 (Save the list to .csv file)] button on the [Pins] page of the Smart Configurator view.

Smart_Configurator_Example.scfg ×						-
in configuration				Gener	ate Code G	ienerate Rep
Hardware Resource 🛛 🕀 📮 🚜	Pin Function	on				R ès
Type filter text	type filter	text (* = any s	string, ? = any character)		All	
💑 All 🔨	Enabled	Function	Assignment	Pin Number	Direction	Remarks /
🗾 Resets		ANIO	P002/AVREFP/ANI0	/ 48	1	
Clock Generation Circuit		ANI1	Not assigned	Not assigned	None	
🙆 Clock Frequency Accuracy		ANI2	Not assigned	Not assigned	None	
Interrupt Controller Unit		ANI3	Not assigned	Not assigned	None	
\$∰ I/O Ports		ANI4	Not assigned	Not assigned	None	
💑 Key Interrupt Function		ANI5	Not assigned	Not assigned	None	
> 🖏 Timer Array Unit		ANI16	Not assigned	Not assigned	None	
🖏 Realtime Clock		ANI17	Not assigned	Not assigned	None	
> 📲 Serial Array Unit		ANI18	Not assigned	Not assigned	None	
✓ ﷺ Serial Interface IICA		ANI19	Not assigned	Not assigned	None	
📦 IICA0		AVCC	Not assigned	Not assigned	None	
📦 IICA1		AVREFM	Not assigned	Not assigned	None	
✓ ≇∰ Serial Interface UARTA		AVREFP	Not assigned	Not assigned	None	
UARTA0		AVSS	Not assigned	Not assigned	None	
UARTA1		CACREF	Not assigned	Not assigned	None	
📲 Remote Control Signal Rec		CLKA0	Not assigned	Not assigned	None	
< A/D Converter >	<	~~~~		- · · · · ·	••	>
Function Pin Number						
erview Board Clocks System Comp	oonents Pins	Interrupt				

Figure 11-3 Output of a List of Pin Functions or Numbers (in csv Format)

## 11.3 Image of MCU/MPU Package (in png Format)

An image of the MCU/MPU package is output in response to clicking on the [ I (Save Package View to external image file)] button of the [MCU/MPU Package] view.



Figure 11-4 Outputting a Figure of MCU/MPU Package (in png Format)



### 12. User Code Protection Feature for Smart Configurator Code Generation Component

This feature empowers users to insert codes to any location in the generated codes by utilizing the specific tags, as shown in Figure 12-1. After the next code generation, the inserted user codes will be protected and automatically merged into the generated files.

The user code protection feature will be supported on the files that are generated by the "Code Generation" component and "Clock" tool.

## 12.1 Specific Tags for the User Code Protection Feature

When using the user code protection feature, please insert /\* Start user code \*/ and /\* End user code \*/ as shown in Figure 12-1 and add the user codes between these tags. If the specific tags do not match exactly, the inserted user code will not be protected after the code generation.



Figure 12-1 Specific Tags for User Code Protection Feature

# 12.2 Examples of Using User Code Protection Feature to Add New User Code

Figure 12-2 shows an example of adding new user code into the Create API of A/D Converter module by using the specific tags shown in Figure 12-1. After updating the configuration in the A/D Converter GUI and re-generating the codes, the inserted user codes will be automatically merged into the newly generated file.



Figure 12-2 User Code Protection with Auto Merge



### 12.3 What to Do When Merge Conflict Occurs

#### 12.3.1 What is Merge Conflict

When the lines of generated codes before and after the inserted user codes are updated due to changes in GUI configuration or the version update of Smart Configurator, merge conflict codes will be generated out.

If the merge conflict occurs, conflict message will be displayed in the Smart Configurator console, as shown in Figure 12-3 The Merge Conflict Message Outputted in the Smart Configurator Console.

Console X
Smart Configurator Output
TOSOCOULT. THE BENEFACEA. SHE BENEFICIAL (L'SHE EBENE
M03000002: File generated: <u>src\smc gen\general\r smc cgc.h</u>
M03000002: File generated: <u>src\smc gen\general\r smc cgc user.c</u>
M04000001: File generated: <u>src\smc gen\Config RTC\Config RTC.h</u>
M04000001: File generated: <u>src\smc gen\Config RTC\Config RTC.c</u>
M04000001: File generated: <u>src\smc gen\Config RTC\Config RTC user.c</u>
M04000001: File generated: <u>src\smc gen\general\r cg interrupt handlers.h</u>
M04000001: File generated: <u>src\smc gen\general\r cg vect table.c</u>
M05000012: File generated: <u>src\smc gen\r pincfg\Pin.c</u>
M00000004: File removed:src\smc_gen\r_config\r_bsp_interrupt_config.h
M03000004: File modified:src\smc gen\r config\r bsp interrupt config.h
M00000005: The above files highlighted in red color have user code merge conflicts, please open the file and resolve the conflict manually
M00000002: Code generation is successful:C:\Users\yang-xiaolin\e2 studio\1102\Smart Configurator Example\src\smc gen

Figure 12-3 The Merge Conflict Message Outputted in the Smart Configurator Console

User can click the conflicted file in the console message to open the File Compare view and then can resolve the conflict as next chapter 12.3.2 Steps for Resolving the Merge Conflict described.



#### 12.3.2 Steps for Resolving the Merge Conflict

User can follow the steps below to solve the merge conflicts.

- (1) Click on the conflicting file in the console to open the "File Compare" view (Figure 12-4 Code before Resolving Conflict).
- (2) Click on "Copy Current Change from Left to Right" (Figure 12-4 Code before Resolving Conflict).



Figure 12-4 Code before Resolving Conflict

(3) Move user codes to the place you want to use (Figure 12-5 Code after Applying "Copy Current Change from Left to Right").





Figure 12-5 Code after Applying "Copy Current Change from Left to Right"

(4) Save the modified code (Figure 12-6 Code after Moving and Saving).



#### Figure 12-6 Code after Moving and Saving

User can also resolve the conflict by editing the code in the right panel directly.

Note: After conflict resolved, if click the confliction message, it still can open [File Compare] view.



### 13. Help

Refer to the help system for detailed information on the Smart Configurator by clicking the [Help Contents] menu..



Figure 13-1 Help Menu

The help system can also be activated from the [Overview information] page by clicking 2 button.

Overview information Generate Ref.	Contents # Search ** Related Topics
- General Information 📀	Bookmarks - Index
Overview     Get an overview of the features provided by Smart Configurator.     Videos     Interduction to Smart Configurator.	Smart Configurator Smart Configurator is a User Interface that combines the functionalities of Code Generator and FIT Configurator which imports, configures and generates different types of drivers and middleware modules.
Browse related videos	See also:
What's New	■ 1. Overview
Check out <u>what's new</u> in the latest release.	2. Creating a Project
See all <u>Release Notes</u> . MCU Hardware	3. Operating the Smart Configurator
Product Documentation	4. Setting of Peripheral Modules
User's Guide	5. Managing Conflicts
<u>API manual</u>	6. Generating Source Code
- Current Configuration	7. Creating User Programs
Selected board/device: R9A02G0214CNE (ROM size: 128 KB, RAM size: 16 KB, Pin count: 48)	8. Backing up Generated Source Code
Generated location (PROJECT_LOC): src\smc_gen Edit	9. Generating Reports
Selected components:	10. Developer Assistance
Component Version Configuration	11. User Code Protection Feature for Smart Configurator Code Generation Component

Figure 13-2 Quick Start

In both ways to check Help information, the whole Help contents is the same.


### 14. Documents for Reference

User's Manual: Hardware

Obtain the latest version of the manual from the website of Renesas Electronics.

Technical Update/Technical News

Obtain the latest information from the website of Renesas Electronics.

User's Manual: Development Environment

Smart Configurator User's Manual: RISC-V MCU API Reference (R20UT5385)

Obtain the latest version of IAR Embedded Workbench for Renesas RISC-V manual from the website of IAR.



# **Revision History**

Rev.	Section	Description
1.00	-	First edition issued
1.01	Chapter Introduction	Add the SEGGER information and URL.
	Chapter 2.1	Add URL.
	Chapter 3.1	Add SEGGER information for operating procedure.
	Chapter 3.3.1	Add SEGGER information for creating a new Smart Configurator
		Configuration File
	Chapter 4.3	Update Figure 4-8.
	Chapter 6.2	Add Figure 6-3 for SEGGER.
		Update table for SEGGER.
	Chapter 7.2	Add SEGGER information for loading in SEGGER Embedded Studio
	Chapter 7.4	Add SEGGER information for build SEGGER Embedded Studio Project
		File
	Chapter 10	Add SEGGER information for how to make debug connections via J-
		Link OB on the FPB Board

### General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

#### 1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power is supplied until the power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins

Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

6. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.).

7. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a systemevaluation test for the given product.

### Notice

- Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
- Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
- 3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
- 4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
- 5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
- Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

- 7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
- 8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
- 9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
- 10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
- 12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
- This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
  Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.
- (Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries
- (Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

# **Corporate Headquarters**

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

## Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

# **Contact information**

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: <a href="http://www.renesas.com/contact/">www.renesas.com/contact/</a>.