

## APPLICATION NOTE

# **RL78 Smart Configurator**

User's Guide: IAREW

R20AN0581EC0104 Rev.1.04 Apr.21.25

## Introduction

This application note describes the basic usage of the RL78 Smart Configurator (hereafter called the Smart Configurator), and the procedure for importing its output files to IAR Embedded Workbench.

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

- IAR Embedded Workbench for Renesas RL78 V5.10.3 and later
- RL78 Smart Configurator V1.13.0 and later

Target device and support compiler

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

https://www.renesas.com/rl78-smart-configurator



## Contents

1. Ove	rview5
1.1 Pu	rpose5
1.2 Fe	atures5
1.3 So	ftware Components5
2. Inst	alling and Uninstalling the Smart Configurator6
	stalling the Smart Configurator
	installing the Smart Configurator
2.2 01	
•	rating the Smart Configurator7
	ocedure for Operations7
	arting the Smart Configurator8
3.3 Cr	eating and Loading a Configuration File9
3.3.1	Creating a New Smart Configurator Configuration File9
3.3.2	Opening an Existing Configuration File10
3.4 W	ndow11
3.4.1	Main Menu12
3.4.2	Toolbar
3.4.3	Smart Configurator View13
3.4.4	MCU/MPU Package View14
3.4.5	Console View15
3.4.5 3.4.6	Console View
3.4.6	Configuration Problems View15
3.4.6 4. Sett	
3.4.6 4. Sett	Configuration Problems View15
3.4.6 4. Sett 4.1 Bo	Configuration Problems View
3.4.6 4. Sett 4.1 Bo 4.1.1	Configuration Problems View
3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2 4.1.3	Configuration Problems View    15      ing of Peripheral Modules    16      pard Setting    16      Selecting the Device    16      Selecting the Board    17
3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2 4.1.3 4.1.4	Configuration Problems View       15         ing of Peripheral Modules       16         pard Setting       16         Selecting the Device       16         Selecting the Board       17         Exporting Board Settings       18
3.4.6 4. Sett 4.1 Bo 4.1.1 4.1.2 4.1.3 4.1.4 4.2 CI	Configuration Problems View       15         ing of Peripheral Modules       16         bard Setting       16         Selecting the Device       16         Selecting the Board       17         Exporting Board Settings       18         Importing Board Settings       18
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	Configuration Problems View       15         ing of Peripheral Modules       16         bard Setting       16         Selecting the Device       16         Selecting the Board       17         Exporting Board Settings       18         Importing Board Settings       18         ock Settings       19
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	Configuration Problems View15ing of Peripheral Modules16bard Setting16Selecting the Device16Selecting the Board17Exporting Board Settings18Importing Board Settings18Sock Settings19stem Settings20
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	Configuration Problems View15ing of Peripheral Modules16bard Setting16Selecting the Device16Selecting the Board17Exporting Board Settings18Importing Board Settings18ock Settings19stem Settings20oftware component settings21
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	Configuration Problems View       15         ing of Peripheral Modules       16         bard Setting       16         Selecting the Device       16         Selecting the Board       17         Exporting Board Settings       18         Importing Board Settings       18         ock Settings       19         stem Settings       20         oftware component settings       21         Switching Between the Component View and Hardware View       21
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	Configuration Problems View15ing of Peripheral Modules16bard Setting16Selecting the Device16Selecting the Board17Exporting Board Settings18Importing Board Settings18ock Settings19stem Settings20oftware component settings21Switching Between the Component View and Hardware View21Adding a Software Component22
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	Configuration Problems View15ing of Peripheral Modules16bard Setting16Selecting the Device16Selecting the Board17Exporting Board Settings18Importing Board Settings18ock Settings19stem Settings20oftware component settings21Switching Between the Component View and Hardware View21Adding a Software Component22Removing Software Component24
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	Configuration Problems View15ing of Peripheral Modules16bard Setting16Selecting the Device16Selecting the Board17Exporting Board Settings18Importing Board Settings18ock Settings19stem Settings20oftware component settings21Switching Between the Component View and Hardware View21Adding a Software Component24Setting a Code Generator Component25
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	Configuration Problems View15ing of Peripheral Modules16bard Setting16Selecting the Device16Selecting the Board17Exporting Board Settings18Importing Board Settings18ock Settings19stem Settings20oftware component settings21Switching Between the Component View and Hardware View21Adding a Software Component22Removing Software Component25Changing the Resource for a Code Generator Configuration26
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	Configuration Problems View15ing of Peripheral Modules16bard Setting16Selecting the Device16Selecting the Board17Exporting Board Settings18Importing Board Settings18ock Settings19stem Settings20oftware component settings21Switching Between the Component View and Hardware View21Adding a Software Component22Removing Software Component25Changing the Resource for a Code Generator Configuration26Setting SNOOZE Mode Sequencer (SMS) Component28

4	.4.10	Create and Edit ELCL Flexible Circuit	34
4	.4.11	Downloading RL78 Software Integration System Modules	38
4	.4.12	Adding a RL78 Software Integration System Module	40
4	.4.13	Setting a RL78 Software Integration System Module	41
4	.4.14	Changing Version of BSP Configuration	42
4	.4.15	Export Component Configuration	44
4	.4.16	Import Component Configuration	44
4	.4.17	Configure General Setting of the Component	45
4.5	i Pin	Settings	47
4	.5.1	Changing the Pin Assignment by PIOR Function	49
4	.5.2	Changing the Pin Assignment of a Software Component	50
4	.5.3	Assigning Pins Using the MCU/MPU Package View	51
4	.5.4	Show Pin Number from Pin Functions	52
4	.5.5	Exporting Pin Settings	53
4	.5.6	Importing Pin Settings	53
4	.5.7	Pin Setting Using Board Pin Configuration Information	54
4	.5.8	Pin Filter Feature	55
4	.5.9	Pin Errors/Warnings setting	56
4.6	i Inte	errupt Settings	57
4	.6.1	Changing Interrupt Priority Setting	57
4	.6.2	Changing Interrupt Bank Setting	58
5.	Mana	aging Conflicts	50
5.1		source Conflicts	
5.2	_	solving Pin Conflicts	
<b>J.Z</b>			00
6.	Gene	erating Source Code	61
6.1	Ge	nerating Source Code File	61
6.2	Co	nfiguration of Generated Files and File Names	62
6.3	lnit	ializing Clocks	65
6.4	l Init	ializing Pins	66
6.5	i Init	ializing Interrupts	67
7.	heo I	ing Generated Files in Integrated Development Environment	68
		ading in IAR Embedded Workbench	
71			62
7.1		-	
7.1 7.2		ild IAR Project File	
7.2	Bui	-	70
7.2	Bui Creat	ild IAR Project File	70 71
7.2 8.	2 Bui Creat Ade	ild IAR Project File	70 71 71
7.2 8. 8.1 8.2	Creat Creat Add Usi	ild IAR Project File ting User Programs ding Custom Code	70 71 71 73

10.1 Report on All Configurations (PDF or Text File)7	5
10.2 Configuration of Pin Function List and Pin Number List (in csv Format)	6
10.3 Image of MCU/MPU Package (in png Format)7	6
11. User Code Protection Feature for Smart Configurator Code Generation Componen	
11.1 Specific Tags for the User Code Protection Feature7	
11.2 Examples of Using User Code Protection Feature to Add New User Code	7
11.3 What to Do When Merge Conflict Occurs7	8
11.3.1 What is Merge Conflict7	8
11.3.2 Steps for Resolving the Merge Conflict7	'9
12. Help8	1
13. Documents for Reference8	2



## 1. Overview

### 1.1 **Purpose**

This application note describes the basic usage of the RL78 Smart Configurator (hereafter called the Smart Configurator), and the procedure for importing its output files to IAR Embedded Workbench.

Refer to the User's Manual of IAR Embedded Workbench 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 RL78 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>Graphical Configurator (SMS, ELCL)</u> The Graphical Configurator module makes it easy to set up complex configurations by providing a graphical GUI compared to other drivers. It provides software components for SNOOZE mode sequencer (SMS) and logic and event link controller (ELCL).
- (3) <u>RL78 Software Integration System (CAPACITIVE SENSING UNIT (CTSU2L), etc.)</u> The RL78 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/rl78-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 RL78" 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 related files using Smart Configurator and loading it into IAR Embedded Workbench. For the operation of IAR Embedded Workbench, refer to relevant document of IAR.



Figure 3-1 Operating Procedure



## 3.2 Starting the Smart Configurator

Select [Smart Configurator for RL78 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.
- (2) In [Toolchain:], select [IAR RL78 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: The \*.eww, \*.ewp, \*.ewd, main.c and buildinfo.ipcf files will be generated to this location after clicking "Generate Code" button.

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

( (

	💰 New Smart Configuration File — 🗆 🗙									
	Smart Configuration Settings									
	Select pla configura	tform and toolchair tion file	n setting	gs for the ne	ew					
	Category:	RL78					~			
	Platform:			Toolchain:						
	type filter	text	(2)	<b>K</b> enesas	CCRL78	Toolchain				
(1)	> RL	78/G23 - <mark>6</mark> 4pin	^	IAR RL7						
	> RL	78/G23 - <mark>8</mark> 0pin		e <sup>2</sup> LLVM fo	r Renesas	RL78				
	> RL	78/G23 - 100pin								
	∽ RL	78/G23 - 128pin								
		R7F100GSJxFB								
		R7F100GSKxFB								
		R7F100GSLxFB								
		R7F100GSNxFB	~							
	ROM size	: 256KB, RAM size: 2	24KB, Pi	n count: 12	8					
3)	File name:	Smart_Configurato	or_Exam	ple						
•)	Location:	D:\RL78\IAR				Brow	se			
					(5)					
				<u>F</u> inish		Cancel				
			_							

Figure 3-3 Create a Configuration File

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

Note: 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.



## 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					×
← → × ↑ 📙 > This P	C >	Local Disk (C:) > smartconfigurator > workspace	~ Ō	Search workspace	م
Organize 👻 New folder					- 🔳 🕐
Pictures	^	Name	Date modified	Туре	Size
Videos		Smart_Configurator_Example.scfg	10/15/2018 1:31 PM	SCFG File	1 KB
🏪 Local Disk (C:)					
💣 Network	1				
	~				
File <u>n</u> ame	: Sm	art_Configurator_Example.scfg	~	Smart Configuration	n files 🛛 🗸
				<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.



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_0	Configurator_Example.scfg ×			- 8
Overvi	ew information		Generate Coo	🗎 de Generate Report
- Genera	al Information			?
m	Overview			
	Get an <u>overview</u> of the featu provided by Smart Configur			
	Videos			
	Introduction to Smart Config Browse related videos	<u>gurator</u>	Application Code	
	What's New Check out <u>what's new</u> in the release. See all <u>Release Notes</u> . Current version: V1.13.0	latest	Software Components Middleware & Drivers Device Drivers MCU Hardware	Ĵ
	Product Documenta and FAQ <u>User's Guide</u> API manual	ition		
	FAQ : Smart Configurator			
- Curren	nt Configuration			
Selected	d board/device: R7F100LPLxFl	3 (ROM siz	comized in the <u>Preferences</u> page. (1) e: 512KB, RAM size: 32KB, Pin cour	
	ed location (PROJECT_LOC\):	src\smc_g	en Edit	
	d components:	Version	Configuration	
Compo	Converter	1.7.0	Configuration Config ADC(ADC: used)	
	d Support Packages v1.90		r bsp(used)	
<ul> <li>Ports</li> </ul>		1.7.0	Config_PORT(PORT: used)	
Overview	Board Clocks System Comp	onents Pir	s Interrupt	· · ·

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 4.5.7 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.

📮 Console 🛛	k	B	2	₽	*	<b></b>	•		
Smart Configurator Output									
M05000001: Pin 3 is assigned to PCLBUZ0 M05000001: Pin 117 is assigned to ANI0									< >
<								>	

#### 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.

Smart_Co	onfigurator_Example.scfg $ imes$			- 8
Device se	election		🐻 Generate Code	📄 Generate Report
Device se	lection			2 2
Board:	Custom User Board	~		
Device:	R7F101GLGxFB			
	Download more boards			
Overview B	oard <mark>C</mark> locks System Componer	nts Pins Inter	rupt	

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	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_Co	onfigurator_Ex	ample.scf	g X					
Device s	election				Gene	😼 erate Code	Generate Re	eport
Device se	election						2	4
Board:	Custom Use	r Board		~				
Device:	R7F101GLG	«FB						
	Download m	ore board	<u>ls</u>					
_								
verview B	loard Clocks	System (	Components	Pins Ir	nterrupt			

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 Lis	st
------------	---	----

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_Co	onfigurator_Ex	ample.scfg	×				- 8
Device se	election				Gene	😼 erate Code	i Generate Report
Device se	election						(1) 같 <b>ট</b>
Board:	Custom Use	r Board		~			
Device:	R7F101GLG	κFB					
	<u>Download m</u>	ore boards	<u></u>				
Overview B	oard Clocks	System Co	omponents	Pins In	terrupt		

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_Co	onfigu	rator_Ex	ample.se	:fg ×					- 0
Device se	elect	ion				Gene	🔋 erate Code	Gener	ate Report
Device se	electio	n						(1)	è
Board:	Cust	om Use	r Board		~				
Device:	R7F1	101GLG	кFB	(2)		]			
	Dowr	nload m	ore boa	rds					
Overview B	oard	Clocks	System	Components	Pins In	terrupt			

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 EVDD 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.

locks configura	ation				Generate Code	Generate R
	iigh-speed main mode 4.0(V)~5.5(V) 10 V ≤ EVDD0 ≤ 5.5 V	v v				
	a oscillator 32 v (NHz) Normal arting the high-speed on-chip oscillator at from STOP mode and of transitions to		(4) (4)	(4)	fMAIN 32 fCLK	(MHz) (MHz)
Middle-speed on-c	hip oscillator				fIMP	( <b>kHz)</b> (MHz)
X1 oscillator Operation mode:	X1 oscillation	Divider x1 v			fMXP	(MHz)
Frequency: Stable time:	5 (MHz) 2^18/fx ▼ 52428.8(μs)					
Low-speed on-chip oso	illator 32.768 (kHz)	]			fiL 32.768	kHz)
The flL runs while WD on-chip oscillator	DT is operating or fSXP select Low-speed		(4)		fSXP 32.768	(kHz)
XT1 oscillator Operation mode:	XT1 oscillation	]			fSXR 32.768	(kHz)
Frequency:	32.768 (kHz)					
XT1 oscillation mode:	con porter consumption i					

Figure 4-7 [Clocks] Page



1

## 4.3 System Settings

User can set the on-chip debug setting on the [System] page. This setting is reflected in into r\_bsp file.

For example, make desired setting as in below figure for illustration, after that click on [	Generate Code	Generate
Code] button as in step (4).		

System configuration		(4)	🐻 Generate Code	ian Generate Report
3				^
▼ On-chip debug setting				
On-chip debug operation setting O Unused (1)	) Use emulator OC	OM Po	rt	
Emulator setting O E2 (2)	) E2 Lite			
Pseudo-RRM/DMM function setting O Unused	) Used			
Start/Stop function setting Unused (3)	) Used			
Monitoring point function setting Unused	) Used			
Trace function setting O Unused	) Used			
Security ID setting Use security ID				
Security ID 0	0x000000000000000000000000000000000000			
Security ID authentication failure setting O Do not erase flash memory data				
Erase flash memory data				~
< Overview Board Clocks System Components F	Pins Interrupt			>

Figure 4-8 Smart Configurator [System] Page Setting

The on-chip debug setting is reflected into r\_bsp file in: \<ProjectDir>\src\smc\_gen\r\_config\r\_bsp\_config.h file. To change the on-chip debug setting code value, please change desired setting and generate code again.

#if def:	ined(ICCRL78)										
/* Option byte setting(When using IAR)											
#define	BSP_CFG_OPTBYTE0_VALUE	(OxEFU)	/*	Generated	value.	Do	not	edit	this	manually	*/
#define	BSP_CFG_OPTBYTE1_VALUE	(0x3AU)	/*	Generated	value.	Do	not	edit	this	manually	*/
#define	BSP_CFG_OPTBYTE2_VALUE	(0x6CU)	/*	Generated	value.	Do	not	edit	this	manually	*/
#define	BSP_CFG_OPTBYTE3_VALUE	(0x84U)	/*	Generated	value.	Do	not	edit	this	manually	*/

Figure 4-9 On-Chip Debug Setting Code Generation

Note: Depending on the MCU type selection or chip part numbers, these setting values vary. Please refer to the latest User's Manual Hardware for the detail setting configuration.



## 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.

$$ Smart_Configurator_Example.scfg $\times$			
Software component configura	tion	🖲 Generate Code	📄 Generate Report
Components       ≥       ≤       ↓         Image: startup             V <>       Startup             V <>       Generic               ✓       >       Startup                ✓       >       Startup	Configure Tree view for components		^ 
	<		>
Overview Board Clocks System Compo	onents Pins Interrupt		

Figure 4-10 [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 4-11 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 [<sup>t</sup> (Add component)] icon. a-1. Click on the [<sup>t</sup> (Add component)] icon.

Smart_Configurator_		ion	🖲 Generate Code	Generate Report
Components	↓ª E E	Configure		i
type filter text	a-1.			
✓ ➢ Startup ✓ ➢ Generic ✓ ➢ r_bsp				
Dverview Board Clock	s System Compo	nents Pins Interru	ıpt	

Figure 4-12 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-

a-4. Click on [Next].

Category	All			
Function	All			
Filter				
Compor	ents	Short Name	Туре	Vers
	onverter	a-3	Code Generator	1.4.0
H Board	Support Packages v1.60	r_bsp	RL78 Software	1.60
H Clock	Output /Buzzer Output Contro		Code Generator	1.4.
H Comp	arator		Code Generator	1.3.
H D/A C	onverter		Code Generator	1.3.
🖶 DALI (	Communication (Control device		Code Generator	1.0.
H DALI	Communication (Control gear)		Code Generator	1.0.
🖶 Data	Transfer Controller		Code Generator	1.3.
<				
Descriptic	og to digital (A/D) converter is fu	nction for conv	erting analog input	s to
Download	RL78 Software Integration Syste	<u>m modules</u>		
	general settings			

Figure 4-13 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	
A/D Converter Configuration name: Config_ADC	
Resource: a-6. ADC	~
? a-7. Finish	Cancel

Figure 4-14 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.



Figure 4-15 Adding a Code Generator Component to the Hardware View



#### 4.4.3 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.



Figure 4-16 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.



#### 4.4.4 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/fCLK] for [Conversion time].

🔅 *Smart_Configurator_Example	e.scfg	22					
Software component cor	nfigu	ration		🕲 Generate Code	Generate Re	epo	rt
Compon la 🗄	Cor	nfigure			(	i)	^
type filter text		Comparator operation setting Stop	Operation				
<ul> <li>✓ ➢ Startup</li> <li>✓ ➢ Generic</li> <li>✓ ≧ r_bsp</li> </ul>	) a.	● 10 bits VREF(+) setting	⊖8 bits	○ 12 bits			
<ul> <li>✓ ➢ Drivers</li> <li>✓ ➢ A/D converter</li> </ul>		● VDD	OAVREFP	O Internal reference volt	tage		
(1) Config_ADC		VREF(-) setting VSS	OAVREFM				
	(2) b	Trigger mode setting Software trigger no wait mode Software trigger wait mode Hardware trigger no wait mode Hardware trigger wait mode					
		INTTM01 ~	(Please set INTTM01)				
		Operation mode setting Operation mode setting	⊖ Continuous scan mode				
		One-shot select mode	One-shot scan mode				
		A/D channel selection (2) c.	ANIO	$\checkmark$			
		Conversion time setting Conversion time mode	Normal 1	~			
		Conversion time (2) d.	2112/fCLK	✓ (66 µs)			
		Conversion result upper/lower bound Generates an interrupt request (INT. Generates an interrupt request (INT.	AD) when ADLL ≤ ADCRn ≤ ADUL	:Rn			
		Upper bound (ADUL) value	255				~
Overview Board Clocks System	n Com	ponents Pins Interrupt					

Figure 4-17 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 ].



#### 4.4.5 **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 4-18 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 Selec	tion		Х
Resource Select	t <b>ion</b> om those available in the list		
Operating mode:	8 bit count mode		$\sim$
Resource:	TAU0_3 (3)		~
	TAU0_1		
	TAU0_3		
	TAU1_1		
	TAU1_3		
	ITL000		
	ITL001		
	ITL012		
	ITL013		
	(4)		
	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel	

Figure 4-19 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		—	$\times$
	Configuration setting selection			
	Configuration setting list	(7)		
	Confirm setting for resource change	● Use setting below 〇	Use default	
(6)	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	~
	<			>
		(8)		
	< Back	Next > Finish	Cance	
	< Dack	FINISH	Cance	

Figure 4-20 Checking the Settings of the New Resource

The resource is automatically changed (for e.g., changed from INTTM01 to INTTM03).

*Smart_Configurator_Example.scfg Software component configuration			Generate Code Generate Report
Components Statup	Configure Clock setting Operation clock Clock source Operation mode setting (a) Higher 8 bits Interval timer setting Interval value (higher 8 bits) Interval value (lower 8 bits)	СК02 fCLK/2 О Lower 8 bits 10 µs	Generate Code Generate Report
Overview Board Clocks System Com	Generates INTTM03 when count Interrupt setting End of timer channel 3 count, ge Priority End of timer channel 3 count, ge Priority < r>  Image: Setting the setting th	enerate an interrupt <mark>(INTTM03)</mark> Low	<ul> <li>✓</li> <li>✓</li> </ul>

Figure 4-21 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 4-22 Renaming the Configuration

#### 4.4.6 Setting SNOOZE Mode Sequencer (SMS) Component

SNOOZE Mode Sequencer (SMS) component is a new component type as "Graphical Configurator", it is list and can be selected to use directly in default component list.

💦 New Co	omponent				×		
Software	Software Component Selection						
Select cor	nponent from those availat	ole in list					
Category	All				$\sim$		
Function	All				$\sim$		
Filter							
Compon	ents	Short Name	Туре	Versi	^		
#Real-Ti	me Clock		Code Genera	1.7.0			
# Remote	e Control Signal Receiver		Code Genera	110	_		
#SNOOZE Mode Sequencer Grap				1.4.0			
# SPI (CS	I) Communication		Code Genera	1.7.0	_		
Square	Wave Output		Code Genera	1.7.0			
<b>UART</b> C	Communication		Code Genera	1.9.0			
Voltage	e Detector		Code Genera	1.6.0			
#Watcho	dog Timer		Code Genera	1.7.0			
					$\sim$		
Show o	only latest version						
Descriptio	n						
The analo	og to digital (A/D) converte	r is function for cor	werting analog in	outs to	$\sim$		
digital sig	gnals.						
Download	I RL78 Software Integration	System modules					
	ELCL modules	system modules					
configure	general settings						
?	< Back	Next >	Finish	Cancel			

Figure 4-23 Add SNOOZE Mode Sequencer



A GUI of Graphical Configurator is displayed in below SMS figure, it is more graphically compared with Code Generator. User can Drag and Drop and configure the block which user want to use.

🗁 Arithmetic	↑ P P B → ☆ i (3)	
<ul> <li>1byte transfer</li> <li>2byte transfer</li> <li>4byte transfer</li> </ul>	Start trigger Interval detection interrupt (INTITL)	8/32
➢ Branch ♥ Compare & branch ♥ Update & branch		
➡ Bit branch       ➢ Special       X Wait       ♥ DTC trigger       •(\$\$ Branch end       ✓       ☐ Debug	(5) Start Master rece	
🗀 Port	-	
🗀 Timer TAU	_	
<ul> <li>□ A/D</li> <li>□ D/A</li> </ul>	(6)	
🗁 CSI	»	
OSI Master send CSI Master receive		
CSI Master receive     CSI Master send-	_	,

Figure 4-24 SNOOZE Mode Sequencer (SMS) GUI

Table 4-3. SIMS GUI area description	Table 4-3.	SMS GUI area description
--------------------------------------	------------	--------------------------

A	rea	Description
(1) Block elements View the available blocks for SMS.		View the available blocks for SMS.
		A block is a part for forming a sequence (function), and includes A/D voltage acquisition,
	~	comparison & branching and 1-byte transfer.
(2) Toolbar	<b>*</b>	Zoom in.
	Þ	Zoom out.
	4	Display the SMS data management dialog and manage the variables to be used.
		Import the SMS sequence. User can use some sample sequences by clicking this icon.
	4	Export the SMS sequence.
	Re	Update the SMS data file.
	i	Displays the information of the SMS data file.
(3) Start trigg	er selection	Select a startup trigger.
(4) Resource	status	It shows registers and the number of instructions used.
(5) Canvas a	rea	Place the SMS block and create the sequence.
(6) SMS cons	sole	Displays message for unavailable configurations.



## **RL78 Smart Configurator**

Follow the procedure below to set up SMS block:

- (1) Select a block from Block elements list (for e.g., CSI Master receive).
- (2) Drag "CSI Master receive" block to SMS canvas between Start block and Finish block where the drop location doesn't show the indicator of  $\bigcirc$ .
- (3) Configure the block by double click to pop the "CSI Master receive setting" property setting dialog.
- (4) Specify the setting in the "CSI Master receive setting" property dialog.
- (5) Open "Data Management" setting, edit the receive data.
- (6) When configuring correctly, the color of bottom right corner will change from red to green.
- (7) Add some blocks, drag and drop to adjust the sequence.



Figure 4-25 SMS Block Configure



#### 4.4.7 Update SMS Data Files

Follow the procedure below to update SMS data file (Block, Sequence) to the latest version. Please use new blocks and sequences by updating.

- (1) Click on SMS GUI button [Update SMS data files] to check if SMS data file have the newer version and download automatically from the web.
- (2) Waiting for the operation finished.
- (3) Finished the latest version update.

Configure		í
Arithmetic		
4byte transfer	Start trigger Update SMS data files Resource used status	
2byte calculation	Pin input edge detection interrupt (INTP3) v Register: 2/16 Instruction	: 1/32
🗁 Branch 🛷		
R Compare & branch		
■ Update & branch		
🗁 Special 🔹		
TC trigger		
୍କା Branch end	Start Finish	
Ca Debug		
124 Debug		
Prograce	s Information	
Flogless	information	
(2)	Operation in progress	
	Cancel	
	nformation X	
(3)		
	No new version.	
	ОК	

Figure 4-26 SMS Data File Download



#### 4.4.8 ELCL Fixed Function Modules Download

The Software Component type for ELCL (Logic Event Link Controller) is Graphical Configurator. ELCL component have 2 types, 1 type is fixed function ELCL component such as "Slave Select Pin Function", "Chattering Prevention Function" and so on, the other one is ELCL flexible circuit, user can use it to create flexible ELCL circuit.

ELCL fixed function modules can be added from component list in New Component dialog. If user wants to use other ELCL fixed function modules which are not included in Component list, the user can click on [Download ELCL modules] link in New Component dialog to check and download more ELCL fixed function modules:

New Co	omponent	1			×			
Select com	ponent from those a	availat	ole in list	t	÷			
Category Function Filter	All		Modules Download				_	
<ul> <li>Clock</li> <li>Comp</li> <li>CSI Co</li> <li>D/A Co</li> <li>Data To</li> </ul>	onverter I Support Packages. Output /Buzzer Outp arator ommunication	(2) ☑	ct the RL78 modules for download. Title Slave Select Pin Function Button Debounce Function Manchester Decoder Function Edge Detection Thinning Function Multiple Parameter Monitoring RL78/G23 Common ELCL Module	Type ELCL ELCL ELCL ELCL ELCL	Rev. 1.0.0 1.0.0 1.0.0 1.0.0 2.1.0		(2)	Select All Deselect All
digital sig	og to digital (A/D) co		System modules			(3) Dow	nload	Cancel
	general settings	3ack	Next > Finish	Cance	1			

Figure 4-27 ELCL Fixed Function Modules Download

After downloading, all ELCL fixed function modules are auto added to component list:



Category	All				~
Function	nction All				~
Filter					
Compone	ents	Short Name	Туре	Versi	^
#D/A Co	nverter		Code Genera	1.5.0	
# Data Tr	ansfer Controller		Code Genera	1.6.0	
# Delay Counter Code Genera				1.7.0	
# Divider Function Code Genera					_
IN ELCL A	ND		Graphical Co	1.1.0	
BELCL ch	attering prevention		Graphical Co	2.0.0	
IN ELCL D	flip flop		Graphical Co	1.1.0	
IN ELCL ec	Ige detection thinning fu		Graphical Co	2.0.0	
IN ELCL EX	(OR		Graphical Co	1.1.0	~
Descriptio	g to digital (A/D) converte	r is function for	converting analog	inputs to	
Download	RL78 Software Integration	System module	<u>'s</u>		
Download	ELCL modules				
Configure	general settings				

Figure 4-28 Add ELCL Fixed Function Module

#### 4.4.9 Setting a Fixed function ELCL Component

Follow the procedure below to set up a fixed function ELCL component.

- (1) Select a fixed function ELCL component from Software Component Selection list (for e.g., ELCL slave select pin function).
- (2) Configure the driver in the [Configure] panel. The following steps and figure show an example.
  - a. Select the input signal under [Input signal selector] UI part.
  - b. Select the logic block under [Event controller (link processor)] UI part.
  - c. Select the output signal under [Output signal selector] UI part.
- (3) If the user wants more details about current fixed function ELCL component usage, the user can click the [ELCL\_slave\_select\_pition.pdf] link to open the application notes for check.



Figure 4-29 Configure a Fixed Function ELCL Component



#### 4.4.10 Create and Edit ELCL Flexible Circuit

ELCL (Logic and Event Link Controller) flexible circuit component is a new component type as "Graphical Configurator", it is list and can be selected to use directly in default component list.

💰 New Co	🐻 New Component — 🗆 🗙					
Software	Software Component Selection					
Select con	Select component from those available in list					
	•					
Category	All				~	
Function	Function All				$\sim$	
Filter	Liltor					
Compon	ents	Short Name	Name Type		^	
🖶 Data Ti	ransfer Controller		Code Generator	1.5.0		
🖶 Delay 🕻	Counter		Code Generator	1.6.0		
🖶 Divider	r Function		Code Generator	1.6.0		
🕼 ELCL A	ND		Graphical Configurator	1.1.0		
🕅 ELCL c	hattering prevention		Graphical Configurator	2.0.0		
🛤 ELCL D	flip flop		Graphical Configurator	1.1.0		
🛤 ELCL e	dge detection thinnin		Graphical Configurator	2.0.0		
	XOR		Graphical Configurator	1.1.0		
5⊕ ELCL F	lexible Circuit		Graphical Configurator	1.0.0		
ដោ ELCL m	nanchester decoder		Graphical Configurator	2.0.0	~	
Show o	only latest version					
Descriptio	on					
The logic	and event link controlle	r (ELCL) links sigr	als output by peripheral fu	inctions to	<b>^</b>	
other spe	cified peripheral function	ons through inter	nal logic cell blocks, allowi			
communications between peripheral functions without CPU intervention.						
<u>Download</u>	Download RL78 Software Integration System modules					
Download ELCL modules						
Configure	general settings					
(?)		< <u>B</u> ack	<u>N</u> ext > <u>F</u> inish	Can	icel	

Figure 4-30 Add "ELCL Flexible Circuit" Component

ELCL flexible circuit component provide intuitive GUI supporting drag & drop operation for ELCL circuit creation and editing, after user finished circuit design, the ELCL registers setting can be generated automatically.





#### Table 4-4 ELCL flexible circuit GUI area description

Area		Description		
(1) ELCL elements		View the available elements for ELCL.		
(2) Property		Displays the setting of selected ELCL element.		
(3) Toolbar	€	Zoom in.		
	۶	Zoom out.		
	1	Show help.		
(4) Canvas area	Input	The area is ELCL input elements are placed.		
	Logic cell blocks	The area is ELCL logic cell elements are placed.		
	Output	The area is ELCL output elements are placed.		

Follow the procedure below to create ELCL flexible circuit:

- (1) Drag and drop the ELCL elements to canvas. The elements include input, logic cell and output.
- (2) Select element and set the property for each dropped element.
- (3) Make line connection by drag & drop from start point to end point.
- (4) After ELCL circuit created, click "Generate Code" button will generate ELCL registers setting automatically.

Note: Procedure (1)(2)(3) is not fixed sequence steps, user can freely do each step to create or edit ELCL circuit.



Figure 4-32 ELCL Flexible Circuit Creation Procedure



Below list some GUI operation details which help the user create ELCL circuit with ease and guide user towards correct designs.

(1) Connect the line through drag & drop the start point to end point.



Note: When you drag the start point, the connectable end points are highlighted as light green.


(2) When user make connection or settings out of hardware limitation, errors will be displayed in each parts. User can resolve the error base on the indicated message.





## Table 4-1 ELCL flexible circuit GUI error message list

No.	Error message
1	The current selected input event signal can not be linked to the inputx control of the pass- through/AND/OR/EX-OR circuit.
2	Signal select register xxx are all occupied, no resources available for allocation.
3	Both ELL1SEL4 and ELL1SEL5 are occupied, if you want to configure this setting, please use the same signal source as another Flip-Flop in same logic cell block.
4	The selected event signal for clock is not consistent with another flip-flop, please use same event signal with another flip-flop.
5	Set and Reset of flip-flop must select different event signal to be linked.
6	The ELCL circuit exist unallocated resource or not finished line connection, so some correspondence code will not be generated or incorrect.
7	When connectING an interrupt request signal to an input signal, can connect only hardware triggers for event-receiving peripheral functions. Please connect a hardware trigger to output signal.
8	Logic cell block Lx celly (flip-flop n) is already occupied, please select other available resources.



#### 4.4.11 Downloading RL78 Software Integration System Modules

RL78 Software Integration System modules are another software component type which can provide simple view for user to make driver/middle/application SW configuration and generate the code. The available RL78 Software Integration System modules can be downloaded from Renesas web.

- (1) Click on [<sup>1</sup> (Add component)] as in Figure 4-12 to open a dialog.
- (2) Click the [Download RL78 Software Integration System module] link in the [New Component] dialog box to start the download.

	omponent					×	
	Component Se		ble in list				
Category						~	
Function All Filter							
Compon # A/D Co			Short Name	Type Code Genera	Versi 1.7.0	^	
<ul> <li>Board S</li> <li>Clock C</li> <li>Compa</li> <li>D/A Co</li> <li>Data Tr</li> <li>Delay C</li> <li>Divider</li> </ul>	Support Packag Dutput /Buzzer arator onverter ransfer Controlli	Output C er	r_bsp	Code Genera RL78 Softwar Code Genera Code Genera Code Genera Code Genera	1.90 1.7.0 1.5.0 1.5.0 1.6.0	~	
Descriptic The analo digital signature Download	on og to digital (A/ gnals.	'D) converte	er is function for a <u>System modul</u> e	converting analog	inputs to	> ^	
?	<	< Back	Next >	Finish	Cance	el	

Figure 4-36 RL78 Software Integration System Download Link

Note: Downloading requires login to "My Renesas". If the user has not logged in, the following dialog box will prompt the user to log in.

y Renesas Login	MyRene
nter the details used to register your My Rene	sas account.
Email address:	
Password:	
	Forgot password
	Proxy Setting
Maintenance. C I consent to usage data being sent to Ren Privacy Policy & Information Collected (lii	
I consent to usage data being sent to Ren Privacy Policy & Information Collected (lii I do not consent to usage data being reco Privacy Policy   Information Collected	nked below) orded & sent to Renesas
I consent to usage data being sent to Ren Privacy Policy & Information Collected (lin I do not consent to usage data being reco	nked below) orded & sent to Renesas
I consent to usage data being sent to Ren Privacy Policy & Information Collected (lii I do not consent to usage data being reco Privacy Policy   Information Collected	nked below) orded & sent to Renesas o's preferences. <u>encesas</u> is easy, and allows you to submit technical support requests and
Consent to usage data being sent to Ren Privacy Policy & Information Collected (iii O I do not consent to usage data being reco Privacy Policy   Information Collected You can change this setting later in e <sup>2</sup> studic Don't have an account? Registering at My R request free samples for selected products, s	nked below) orded & sent to Renesas o's preferences. <u>enesas</u> is easy, and allows you to submit technical support requests and

Figure 4-37 Login to My Renesas



- (3) Select the checkbox of the required module in the [RL78 Software Integration System Modules Download] dialog box.
- (4) Click on [Browse...] to select the location where the downloaded module is to be stored.
- (5) Click on [Download] to start downloading the selected RL78 Software Integration System Modules module.

Sele	ect the RL78 Software Integration System modules f	or download		(4	(4)		
	Title	Document No.	Rev.	Issue date	^	Se	
$\checkmark$	RL78 Family FS2012 Sensor Control Module So	R01AN6196EJ0112	Rev.1.12	2023-04-26		Des	
$\checkmark$	RL78 Family HS400X Sensor Control Module S	R01AN6446EJ0102	Rev.1.02	2023-04-26		Des	
$\checkmark$	RL78 Family HS300x Sensor Control Module S	R01AN6194EJ0122	Rev.1.22	2023-03-03			
$\checkmark$	RL78 Family ZMOD4410, ZMOD4450 and ZMO	R01AN6197EJ0121	Rev.1.21	2023-03-03			
$\checkmark$	RL78 Family Sensor I2C Communication Middl	R01AN6193EJ0111	Rev.1.11	2023-03-03			
$\checkmark$	RL78 Family OB1203 Sensor Control Module S	R01AN6379EJ0102	Rev.1.02	2023-03-03			
$\checkmark$	RL78 Family FS3000 Sensor Control Module So	R01AN6195EJ0101	Rev.1.01	2023-03-03			
$\checkmark$	RL78 Family FS1015 Sensor Control Module So	R01AN6198EJ0101	Rev.1.01	2023-03-03			
$\checkmark$	RL78 Family CTSU Module Software Integratio	R11AN0484EJ0130	Rev.1.30	2023-02-13	~		
•				2	•		
Mo	dule Folder Path:						
[	C:\Users\ \.eclipse\com.renesas.smc.rcp.rl7	8.product.download	RI 78 Modul	es\GenericMod	ules	Bro	

Figure 4-38 Downloading RL78 Software Integration System Modules



### 4.4.12 Adding a RL78 Software Integration System Module

The following describes the procedure for adding a RL78 Software Integration System Module.

- (1) Click on the [to (Add component)] icon.
- (2) Select components which [Type] is [RL78 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].

	omponent		- □ >
	omponent Selection	in list	
Category	All		
Function	ΔΙΙ		
Filter			,
Compon H A/D C	converter (2)	Short Name	Type Code Generator (3)
H Board	d Support Packages v1.60	r_bsp	RL78 Software Integration System
🗄 Capac	citive Sensing Unit driver.	r_ctsu	RL78 Software Integration System
Clock	Output /Buzzer Output Contr	ro	Code Generator
H Comp	parator		Code Generator
🖶 D/A C	Converter		Code Generator
🖶 Data 1	Transfer Controller		Code Generator
🖶 Delay	Counter		Code Generator
🖶 Divide	er Function		Code Generator
S ELCL	AND		Graphical Configurator
<	only latest version		>
The CTSU is configu			e Sensing Unit. The CTSU2L module e layer, but can also be accessed
Download	RL78 Software Integration S	<u>ystem modules</u>	
Download	ELCL modules		
Configure	general settings		
			(4)

Figure 4-39 Adding RL78 Software Integration System Module



### 4.4.13 Setting a RL78 Software Integration System Module

To use RL78 Software Integration System module, set configuration option. Setting methods depends on components,

✓ Set configuration options on Configure panel and settings will be generated to configuration file of RL78 Software Integration System module automatically at each time of code generation action.

Note: The configuration file of RL78 Software Integration System module will be generated in the r\_config folder.

oonents 🖻 🖾 🖧 🖻 🕀	Configure		
18 🕹 😈	Property	Value	
e filter text	✓		
→ Startup	# Parameter check	Use system default	
✓ 🗁 Generic	# Data transfer of INTCTSUWR and INTCTSURD	Interrupt handler	
💣 r_bsp	# DTC setting	Setting in r_ctsu	
<ul> <li></li></ul>	# Auto-judgment function in Snooze mode using SMS	Disable	
	# Data storage address setting for CTSURD	0xFF500	
	# Data storage address setting for CTSUWR	0xFF800	
⇒ Middleware	# Interrupt level for INTCTSUWR	Level 2	
🖌 🗁 Generic	# Interrupt level for INTCTSURD	Level 2	
💣 r ctsu	# Interrupt level for INTCTSUFN	Level 2	
• · _ · · _ · · ·	# Output port number for external trigger	PORT14	
	# Bit number for exrernal trigger output	BITO	
	# Interrupt port number for external trigger	INTP1	
	V 🔲 Resources		
	< CONTRACTOR		>
			-

Overview Board Clocks System Components Pins Interrupt

Figure 4-40 Setting RL78 Software Integration System Module



#### 4.4.14 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.



Figure 4-41 Version Change of BSP Configuration

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

ổ Change Version						—		×
Version Selection								
Select available ver	rsion							
Component name:	k hen							
component name.	1_psb							
Current version:	1.00							
Available versions:	1.01							$\sim$
		_		_				
	< Back		Next >		Finish		Cancel	

Figure 4-42 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 There are no differences		Status	
			>
< Back Next > Finis	h	Cance	el

Figure 4-43 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 4-44 Confirm Version Change

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



#### 4.4.15 **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.



Figure 4-45 Export Configuration (xml format)

## 4.4.16 Import Component Configuration

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



Figure 4-46 Import Configuration (xml format)



#### 4.4.17 **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 4-13) and display the [Preferences] dialog. Or click "Preferences" of "Window" in Main Menu.

S Preferences			$\times$			
type filter text	Component	¢ •	- 🗢 🔹 🖇			
<ul> <li>Help</li> <li>Logging</li> <li>Module Downlo</li> <li>My Renesas</li> </ul>	Backup settings ✓ Enable Backup settings Number of trash item (1-20): 5					
<ul> <li>Scripting</li> <li>Smart Configura</li> </ul>	Code Generator component settings					
Component	API function output: Output all API functions according to the setting	g	$\sim$			
MCU/MPU Pa			$\sim$			
Pin Errors/Wa	FIT(RX) / SIS(RL78 / RISC-V MCU) component settings					
	Code generation behavior: Update configuration files ~					
	Dependency settings Change these options to control how a component is added Adding dependency: Add dependent component		~			
	Checking dependency: Ignore if dependent component is newer		~			
	Location settings have moved to the <u>Module Download</u> page					
< >	Restore Defaults	A	pply			
	Apply and Close	Cano	cel			

Figure 4-47 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 4-48 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.

	Code generation behavior] Change
	Update configuration files Re-generate all component files
Code generation behavior:	
FIT(RX) / SIS(RL78 / RISC-V	MCU) component settings



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 com	ponent settings	
	Output all API functions according to the setting	~
API code style:	Output all API functions according to the setting	
/ a reode style.	Output only initialization API function	

Figure 4-50 [RL78 API function output] Change

From Smart Configurator for RL78 V1.4.0, output only initialization API feature can be applied for individual configuration (Code Generator component). Please right-click the selected component and select the "Output only initialization API" from the context menu.

<ul> <li>Brivers</li> <li>Interrupt</li> </ul>			
Config_INTC	~	Generate code	
		Output only initialization API	
		Change resource	
		Remove Duplicate Rename	
		Reset to default	
	+	Add Configuration	>

Figure 4-51 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 con	mponent settings	
API function output	t: Output all API functions according to the setting	$\sim$
API code style:	Value with macro description	~
FIT(RX) / SIS(RL78 /	Value with macro description Value without macro description (raw HEX)	

Figure 4-52 [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	1
Specify location of com	Strict check for dependent component	J

Figure 4-53 [Checking dependency] Change



## 4.5 **Pin Settings**

The [Pins] page is used for assigning pin functions. Users 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.

mart_Configurator_Example.s	cfg ×									-
n configuration								Generate Co		👜 ate Rep
ardware Resource	E E 🖧 歳	Pin Functio	on					4	) 🖬 🖬	🗠 🛛
Type filter text		type filter	text (* = any s	tring, ? = any	character)				All	
📥 All	^	Enabled	Function	PIOR	Assignment	Pin Number	Direction	Remarks		Con '
\$∰ I/O Ports			ANIO		P20/ANI0/AVREFP/EI20	/ 117	1	There is no software	initialising	
🏢 Clock Generator			ANI1		Not assigned	Not assigned	None			
🗸 🦓 Timer Array Unit			ANI2		Not assigned	Not assigned	None			
✓ TAU0			ANI3		Not assigned	Not assigned	None			
📦 TAU00			ANI4		Not assigned	Not assigned	None			
📦 TAU01			ANI5		Not assigned	Not assigned	None			
STAU02			ANI6		Not assigned	Not assigned	None			
TAU03			ANI7		Not assigned	Not assigned	None			
TAU04			ANI8		Not assigned	Not assigned	None			
TAU05			ANI9		Not assigned	Not assigned	None			
TAU06			ANI10		Not assigned	Not assigned	None			
TAU07			ANI11		Not assigned	Not assigned	None			
✓ TAU1			ANI12		Not assigned	Not assigned	None			
TAU10			ANI13		Not assigned	Not assigned	None			
TAU11			ANI14		Not assigned	Not assigned	None			
TAU12			ANI16		Not assigned	Not assigned	None			
TAU13			ANI17		Not assigned	Not assigned				
TAU14 TAU15	Displa	ay swite	ahing		Not assigned	Not assigned				
TAU15			ANIIIO	1	/ Not accigned	/ Not accigned	None			>

Figure 4-54 [Pins] Page ([Pin Function])



## RL78 Smart Configurator

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 [**I**] icon displayed in the [Function] selection list indicates the initial pin function of the board.

	Help Run							
	gurator_Example.scfg $\times$							
in configu	ration							😼 📄 Generate Code 🛛 Generate Rep
in Number								🔣   📑   🚵
type filter tex	t (* = any string, ? = any char	acter)						All
Pin Numb	Pin Name	Board Functi	Function	Direction	Remarks	Symbolic Name	Comments	
1	P142/SCK30/SCL30	P142	Not assigned	None			Pmod1-4[RTS]	
2	P141/PCLBUZ1/INTP7	P141	Not assigned	None			Pmod1-1[CTS]	
3	P140/PCLBUZ0/INTP6	INTP6	Not assigned	None			Pmod1-7[INT]	
4	P120/ANI19/IVCMP1/EI120		Not assigned	None			J4-12	
5	P37/ANI21		Not assigned	None			J4-11	
6	P36/ANI22		Not assigned	None			J4-10	
7	P35/ANI23		Not assigned	None			J4-9	
8	P34/TxDA1	TxDA1	Not assigned	None			J9-2[IO1/TX]	
9	P33/RxDA1	RxDA1	Not assigned 👻	None			J9-1[IO0/RX]	
10	P32/CLKA1		Not assigned	None			J4-8	
11	P106/TI17/TO17		P33	None			J9-4[IO~3]	
12	P105/TI16/TO16		RxDA1	one			J9-5[IO~4]	
13	P104/TI15/TO15		Not assigned	None			J9-6[IO~5]	
14	P103/TI14/TO14		Not assigned	None			J9-7[IO~6]	
15	P47/INTP2	P47	Not assigned	None			Pmod2-7[IO1]	
16	P46/INTP1/TI05/TO05		Not assigned	None			J9-8[IO~7]	
17	P45/SO01		Not assigned	None			J4-7	
18	P44/SI01/SDA01	SDA01	Not assigned	None			Grove3-2[SDA]	
19	P43/SCK01/SCL01	SCL01	Not assigned	None			Grove3-1[SCL]	
20	P42/TI04/TO04		Not assigned	None			J4-1[IO~8]	

Figure 4-55 [Pins] Page ([Pin Number])



### 4.5.1 Changing the Pin Assignment by PIOR Function

PIOR "Filter Function" is a powerful feature to help user manage pin function settings, re-configure pin function settings or check pin function conflicts.

Follow the procedure below to change the assignment by PIOR function.

- (1) Type "pior1" in the tool text input box, all pin functions which related to PIOR1 will be listed out.
- (2) If user change one of pin assignment, all pin function assignments which related to PIOR1 will be reassigned automatically.
- (3) The pin error messages may display in [Remark] column and [Configuration Problems view].
- (4) Please re-configure pin assignment.

pior1					(3	5)	all 🖉	
Enabled	Function	PIOR	Assignment	Pin Number	Direction	Remarks	Comr	ner
	INTP10	PIOR1	/ P110/INTP10	/ 98	1			
	INTP11	PIOR1	/ P111/INTP11	/ 99	1			
	RxD0	PIOR1	Not assigned (2)	Not assigned	None			
	RxD2	PIOR1	Not assigned	Not assigned	None			
	🙆 SCK00	PIOR1	P55/PCLBUZ1/SCK00	/ 72	IO	Multiple pin functions on the sa	me pin	
	🙆 SCK20	PIOR1	Not assigned	Not assigned	None	Component requires a pin		
	SCL00	PIOR1	Not assigned	Not assigned	None			
	SCL20	PIOR1	Not assigned	Not assigned	None			
	SDA00	PIOR1	Not assigned	Not assigned	None			
	SDA20	PIOR1	Not assigned	Not assigned	None			
	SI00	PIOR1	P16/EO16/CCD00/TI01/TO01/INTP5/SI00/RxD0	/ 76	1			
	SI20	PIOR1	Not assigned	Not assigned	None			
	SO00	PIOR1	Not assigned	Not assigned	None			
	🙆 SO20	PIOR1	Not assigned	Not assigned		Component requires a pin		
<	T 00	PIOD4	with the second se	La Kristin (* 1946)				
	guration Proble 0 warnings, 0 o							
Descripti	ion		^				Туре	Т
🗸 🙆 Pin	ı (7 items)							
-			nfig_CSI20 is not allocated to any pin.				Pin	
-			fig_CSI20 is not allocated to any pin.				Pin	
۵		used by PCLBL	IZ1 in Config PCLBUZ1 conflicts with pin used by SCK00	in Pin Allocator, p			Pin	
0 0								
0 0 0	E04010003: Pin	used by SCK00	in Config_CSI00 conflicts with pin used by PCLBUZ1 in ( sed multiple times. Pin 72 is assigned to PCLBUZ1 and 9		oin used by F	CLBUZ1 in Pin Allocator.	Pin Pin	

Figure 4-56 PIOR Filter Function

The PIOR setting can be reflected into r\_bsp file in: \<ProjectDir>\src\smc\_gen\r\_config\r\_bsp\_config.h file. To change the PIOR setting code value, please change the assignment of related pin and generate code again.

*Smart_Configurator_Example.scfg	23							
n configuration						Generate C	ode Generate I	
oftware Com 🕀 🖃 🖓 😹	Pin Functi	on					2   💷   🖬   2	×9 g
Type filter text	type filter	text (* = an	y string, ?	= any character)			All	`
🛩 💑 r_bsp	Enabled	Function	PIOR	Assianment	Pin Number	Direction	Remarks	
🔍 r_bsp		PCLBUZ0	PIOR3		/ 3	0		
<ul> <li>Clock Output /Buzzer Ou</li> <li>Config PCLBUZ0</li> </ul>								
Config_PCLB020								
< >	۲							3
Function Pin Number								
erview Board Clocks System Co	mpopents P	ins Internus	+					
	inponents P	ins interrup	·					
r_bsp_config.h🔛								
1_bsp_conrig. nea								_
	pin							
07 128/100-		0						
07 128/100- 08 0 : PCLBUZ	0 - P14							
07 128/100- 08 0 : PCLBUZ 09 1 : PCLBUZ	0 - P14							
07 128/100- 08 0 : PCLBUZ 09 1 : PCLBUZ	0 - P14 0 - P31		) <b>(</b> *	Generated value.	Do not ed	it thig	manually	*

Figure 4-57 PIOR Code Generation



### 4.5.2 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 [4] (Show by Hardware Resource or Software Components)] to switch to the component view.
- (2) Select the target software component (for e.g., Config\_INTC).
- (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 P12 to P16).
- (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.

Pin configuration								1	<b>b</b>
(1)								jenerate Code Genera	ate Repo
Software Components 🕞 📄 🛃 👪	Pin Fu	nction					(	 2 🗉 🖬	è e
Type filter text	type	ilter text (* =	any string	, ? = any character)				All	~
v≛r_bsp (3	) Enaŭ	led Function	PIOR	Assignment	Pin Number	Direction	Remarks	Comments	^
2) _ • r_bsp		INTP2	(4)	/ P51/EI51/EO51/CCD02/INTP2/SO11	/ 34	1	There is no software initialising this pin		
- Interrupt Controller		INTP5	PIOR4	P16/EO16/CCD00/TI01/TO01/INTP5/SI00/RxD0	/ 40	L			
Config_INTC		INTP0		Not assigned	Not assigned	None			
		INTP1		Not assigned	Not assigned	None			
		INTP3		Not assigned	Not assigned	None			
		INTP4		Not assigned	Not assigned	None			
		INTP6		Not assigned	Not assigned	None			
		INTP7		Not assigned	Not assigned	None			

Figure 4-58 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. In this case, no initialization code will be generated, so add the component.



### 4.5.3 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* (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 4-59 Assigning Pins Using the MCU/MPU Package View



#### 4.5.4 **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.

configurat	lion								Ge	nerate Co	de Gener	ate Repor
ftware Compo	o 🕂 🖻 ↓² 🐰 Pi	Funct	tion							ર	🖪   🖫	2 4
Гуре filter text	t	pe filte	er text (* = any	/ string, ? = any	character)						All	~
> 🚣 r_bsp		nabled	f Function	PIOR As	signment	1	Pin Number	Direction	Remarks	Comm	ients	^
🗸 💑 Interrupt	Controller		INTP1	PIOR5 /	Not assigned		Not assigned	None				
💣 Config	LINTC		INTP2		Not assigned		Not assigned					
			INTP3		Not assigned		Not assigned					
			INTP4		Not assigned		Not assigned					
			INTP5		P16/EO16/CCD				Thank 1	÷		
			INTP6		Not assigned		ump to Pin Nu	mber		Ê.		
			INTP7		Not assigned		Merge commer	nt to Pin Nur	nber tab			
			INTP8		Not assigned		Clear comments	5		L		
			INTP9		Not assigned		Assign selected			-		
			INTP10		Not assigned		Unassign select	1		L		
			INTP10		Not assigned		Not assigned			J		
			INTPTT	PIORI	Not assigned		Not assigned	INONE				~
	Clocks System Compon	nts Pir	ns Interrupt									
rview Board (		nts Pir	ns Interrupt			ŀ						
view Board (	Clocks System Compon gurator_Example.scfg ×	nts Pir	ns Interrupt			ŀ				Generate	Code Ge	<b>a</b>
view Board ( Smart_Config Pin configu	Clocks System Compon gurator_Example.scfg ×	nts Pir	ns Interrupt			ŀ					Code Ge	enerate Re
view Board ( Smart_Config Pin configu Pin Number	Clocks System Compon gurator_Example.scfg ×										Code Ge	enerate Re
rview Board ( Smart_Config Pin configu Pin Number	Clocks System Compon gurator_Example.scfg × ration	charac		Function	Direction	Remarks	Symbolic Nam	e Comm	ients		Code Ge	enerate Re
rview Board ( Smart_Config Pin configu Pin Number	Clocks System Component gurator_Example.scfg × ration	charac	ter)	Function		Remarks	Symbolic Nam	e Comm J1-19	ients		Code Ge	enerate Re
rview Board ( Smart_Config Pin configu Pin Number type filter tex Pin Numb	Clocks System Component gurator_Example.scfg × ration t (* = any string, ? = any Pin Name	charac E	ter)		d None	Remarks	Symbolic Nam		ients		Code Ge	enerate Re
view Board ( Smart_Config Pin configu Pin Number type filter tex Pin Numb 74	Clocks System Component gurator_Example.scfg × ration t (* = any string, ? = any Pin Name P57/INTP3	charac E	ter) Board Functi	Not assigne	d None	Remarks There is		J1-19 J1-20	ients 2-1[INT]		Code Ge	enerate Re
view Board ( Smart_Config Pin configu Pin Number type filter tex Pin Numb 74 75	Clocks System Component gurator_Example.scfg × ration tt (* = any string, ? = any Pin Name P57/INTP3 P17/EO17/CCD01/TI02/	charac E TO II	ter) Board Functi	Not assigne	d None d None I			J1-19 J1-20 Pmod2			Code Ge	enerate Re
view Board ( Smart_Config Pin configu Pin Number type filter tex Pin Numb 74 75 76	Ilocks System Component gurator_Example.scfg × ration tt (* = any string, ? = any Pin Name P57/INTP3 P17/EO17/CCD01/TI02/ P16/EO16/CCD00/TI01/	charac [ [ [ [ [ [ [ [ [ [ [ [ [ [ [ [ [ [ [	ter) Board Functi	Not assigne Not assigne INTP5	None None I None			J1-19 J1-20 Pmod2 Grove	2-1[INT]		Code Ge	enerate Re
view Board ( Smart_Configu Pin configu Pin Number type filter tex Pin Numb 74 75 76 77	clocks System Component gurator_Example.scfg × ration ti (* = any string, ? = any Pin Name P57/INTP3 P17/EO17/CCD01/TI02/ P16/EO16/CCD00/TI01/ P15/EO15/SCK20/SCL20	charac E TO II (TI 2/ R	ter) Board Functi NTP5 txD2	Not assigne Not assigne INTP5 Not assigne	d None I None I None d None			J1-19 J1-20 Pmod2 Grove J8-3[10	2-1[INT] 2-1[SCL]		Code Ge	enerate Re
view Board C Smart_Configu Pin configu Pin Number type filter tex 74 75 76 77 78	Clocks System Component gurator_Example.scfg × ration ti (* = any string, ? = any Pin Name P57/INTP3 P12/E015/CCD01/TI02/ P16/E015/SCC20/SCL20 P15/E015/SCK20/SCL20 P14/VCOUT1/E014/RxE	charac [0 [0] [70]	ter) Board Functi NTP5 txD2	Not assigne Not assigne INTP5 Not assigne Not assigne	d None I None I None d None d None			J1-19 J1-20 Pmod2 Grove J8-3[10	2-1[INT] 2-1[SCL] D19/RX1]/Grov		Code Ge	enerate Re
view Board C Smart_Config Pin configu Pin Number type filter tex 74 75 76 77 78 79	Clocks     System     Component       gurator_Example.scfg     ×       ration       tt (* = any string, ? = any       Pin Name       P57/INTP3       P16/E015/CCD01/TI02/       P16/E015/SCK20/SCL2       P14/VCOUT1/E014/RxE       P13/IVREF1/E013/TxD2	characc FO II 771 22/ R S T D	ter) Board Functi NTP5 txD2	Not assigne Not assigne INTP5 Not assigne Not assigne Not assigne	d None I None d None d None d None d None			J1-19 J1-20 Pmod2 Grove J8-3[I0 J8-4[I0	2-1[INT] 2-1[SCL] D19/RX1]/Grov		Code Ge	enerate Re
rview Board ( Smart_Config Pin configu Pin Number type filter tex Pin Numb 74 75 76 77 78 79 80	Ilocks System Component gurator_Example.scfg × ration tt (* = any string, ? = any Pin Name P57/INTP3 P15/EO15/CCD01/TI02/ P16/EO16/CCD00/TI01/ P15/EO15/SCK20/SCL20 P14/VCOUT1/EO14/RxE P13/IVREF1/EO13/Tx02 P12/EI12/EO12/SO00/T P11/EI11/EO11/SI00/Rx	characc TO II 771 7 2/ R S T D 00	ter) Board Functi NTP5 txD2	Not assigne Not assigne INTP5 Not assigne Not assigne Not assigne Not assigne	d None I None d None d None d None d None d None			J1-19 J1-20 Pmod2 Grove J8-3[10 J8-4[10 J1-21	2-1[INT] 2-1[SCL] D19/RX1]/Grov		Code Ge	enerate Re
rview Board ( Smart_Config Pin configu Pin Number type filter tex Pin Numb 74 75 76 77 78 79 80 81	Idocks System Component gurator_Example.scfg × ration ti (* = any string, ? = any Pin Name P57/INTP3 P12/E015/SCK20/SCL20 P16/E016/CCD00/T101/ P15/E015/SCK20/SCL20 P13/IVREF1/E013/TxD2 P12/E112/E012/SO00/T	characc TO II 771 7 2/ R S T D 00	ter) Board Functi NTP5 txD2	Not assigne Not assigne INTP5 Not assigne Not assigne Not assigne Not assigne Not assigne	d None I None None None None None None None None			J1-19 I1-20 Pmod2 Grove J8-3[I0 J8-4[I0 J1-21 J1-22	2-1[INT] 2-1[SCL] D19/RX1]/Grov		Code Ge	enerate Re
Smart_Config Pin configu Pin Number type filter tex Pin Numb 74 75 76 77 78 79 80 81 82 83	clocks System Component gurator_Example.scfg × ration tt (* = any string, ? = any Pin Name P57/INTP3 P12/EO12/CCD01/TI02/ P16/EO16/CCD00/TI01/ P15/CO15/SCK20/SCL20 P14/VCOUT1/EO14/Rxt P13/IVREF1/EO13/TxD2 P12/E112/EO12/SO00/T P12/E112/EO12/SO00/T P12/E112/EO10/SCK00/ P90	characc TO II 771 7 2/ R S T D 00	ter) Board Functi NTP5 txD2	Not assigne Not assigne INTP5 Not assigne Not assigne Not assigne Not assigne Not assigne Not assigne	d None I None I None d None d None d None d None d None d None			J1-19 I1-20 Pmod2 Grove J8-3[10 J8-4[10 J1-21 J1-22 J1-23 J1-23 J1-24	2-1[INT] 2-1[SCL] D19/RX1]/Grov		Code Ge	_
view Board ( Smart_Config Pin configu Pin Number type filter tex Pin Numb 74 75 76 77 78 79 80 81 82	Clocks System Component gurator_Example.scfg × ration t (* = any string, ? = any Pin Name P57/INTP3 P17/EO17/CCD01/TI02/ P16/EO16/CCD00/TI01/ P15/EO15/SCK20/SCL20 P14/VCOUT1/EO14/RxE P13/IVREF1/EO13/TxD2 P13/EF1/EO12/SC000/T P11/EI11/EO11/SI00/Rx P10/EI10/EO10/SCK00/	characc TO II 771 7 2/ R S T D 00	ter) Board Functi NTP5 txD2	Not assigne Not assigne INTP5 Not assigne Not assigne Not assigne Not assigne Not assigne	d None I None d None d None d None d None d None d None d None d None			J1-19 I1-20 Pmod2 Grove J8-3[10 J8-4[10 J1-21 J1-22 J1-23	2-1[INT] 2-1[SCL] D19/RX1]/Grov		Code Ge	enerate Re

Figure 4-60 Jump to Pin Number



### 4.5.5 **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 [14] (Export board setting)] 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							词 Generate Code	Generate Re	epc
oftware Com 🕀 📄 🖧 👪	Pin Function	on					3	∎   <mark>(1</mark> )₂	Ľ
Type filter text	type filter	text (* = ar	ny string, ?	= any character)			Al	l	~
> 💑 r_bsp	Enabled	Function	PIOR	Assignment	Pin Number	Direction	Remarks	Comments	^
V 🚣 Interrupt Controller		INTP1	PIOR5	Not assigned	Not assigned	None			i.
Config_INTC		INTP2		Not assigned	Not assigned	None			
		INTP3	PIOR5	Not assigned	Not assigned	None			
		INTP4	PIOR5	Not assigned	Not assigned	None			
		INTP5	PIOR4	/ P16/EO16/CCD00/TI01/T	/ 76	1	There is no soft		
		INTP6	PIOR5	Not assigned	Not assigned	None			
		INTP7	PIOR5	Not assigned	Not assigned	None			
		INTP8	PIOR5	Not assigned	Not assigned	None			~
	<							>	

Figure 4-61 Exporting Pin Settings to an XML File

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

#### 4.5.6 Importing Pin Settings

To import pin settings into the current project, click on the [22] (Import board setting)] 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.

n configuration							🐻 Generate Code	e Generate Re	ерс
oftware Com 🕀 🖻 🖧 😹	Pin Functi	on					ર	🖬   🖪 🔁	
Type filter text	type filter	text (* = ar	ny string, ?	= any character)			Al	11	~
> 💑 r_bsp 🛩 💑 Interrupt Controller	Enabled	Function INTP1	PIOR PIOR5	Assignment / Not assigned	Pin Number / Not assigned	Direction None	Remarks	Comments	^
Config_INTC		INTP2 INTP3	PIOR5	<ul> <li>Not assigned</li> <li>Not assigned</li> </ul>	<ul><li>Not assigned</li><li>Not assigned</li></ul>				
		INTP4	PIOR5	Not assigned	Not assigned	None	There is no soft		
		INTP5 INTP6	PIOR4 PIOR5	<ul> <li>P16/EO16/CCD00/TI01/T</li> <li>Not assigned</li> </ul>	<ul> <li>Not assigned</li> </ul>	None	There is no soπ		
		INTP7 INTP8	PIOR5 PIOR5	<ul><li>Not assigned</li><li>Not assigned</li></ul>	<ul> <li>Not assigned</li> <li>Not assigned</li> </ul>				
	<							>	-

Figure 4-62 Importing Pin Settings from an XML File

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



#### 4.5.7 Pin Setting Using Board Pin Configuration Information

User can set the initial pin configuration of the board at once. The following describes the procedure for collective setting of pins.

- (1) Select a board setting information except [Custom User Board] in [Board] page. Please refer to 4.1.2 Selecting the Board
- (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].



Figure 4-63 Setting for Initial Pin Configuration

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



#### 4.5.8 **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.

oftware Co 🔃 🖃 🔩 品	્ર	2   🔳   🖬   🔤					
Type filter text	type filter	text (* = ar	y string, ?	= any character)	4	All	~
> 🚣 r_bsp 🗸 🚣 Interrupt Controller	Enabled	Function INTP0	PIOR	Assignment / Not assigned		All Function	
Config_INTC		INTP1 INTP2	PIOR5	<ul> <li>Not assigned</li> <li>Not assigned</li> </ul>	<ul> <li>Not assigned</li> <li>Not assigned</li> </ul>	Assignment	
		INTP3	PIOR5	<ul> <li>Not assigned</li> <li>Not assigned</li> </ul>	✓ Not assigned	Direction Remarks	
		INTP5	PIOR4	/ P16/EO16/CCD00/TI	/ 76	Comments	
		INTP6 INTP7	PIOR5 PIOR5	<ul><li>Not assigned</li><li>Not assigned</li></ul>	<ul><li>Not assigned</li><li>Not assigned</li></ul>		
		INTP8 INTP9	PIOR5 PIOR5	<ul> <li>Not assigned</li> <li>Not assigned</li> </ul>	<ul><li>Not assigned</li><li>Not assigned</li></ul>		
		INTP10	PIOR1	Not assigned	Not assigned		
	<	INTP11	PIOR1	Not assigned	Not assigned	None	. 1

Figure 4-64 Filter for [Pin Function] Page

n Number							🛛 🔣   🔛 1
ype filter tex	tt (* = any string, ? = any chara	acter)					All
Pin Numb	Pin Name	Board Functi	Function	Direction	Remarks	Symbo	All Pin Number
74	P57/INTP3		Not assigned	None			Pin Number Pin Name
75	P17/EO17/CCD01/TI02/TO		Not assigned	None			Board Functions
76	P16/EO16/CCD00/TI01/TO	INTP5	INTP5	1	There is		Function
77	P15/EO15/SCK20/SCL20/TI		Not assigned	None			Direction
78	P14/VCOUT1/EO14/RxD2/	RxD2	Not assigned	None			Remarks
79	P13/IVREF1/EO13/TxD2/S	TxD2	Not assigned	None			Symbolic Name Comments
80	P12/EI12/EO12/SO00/TxD		Not assigned	None			11 21
81	P11/EI11/EO11/SI00/RxD0		Not assigned	None			J1-22
82	P10/EI10/EO10/SCK00/SCL		Not assigned	None			J1-23
83	P90		Not assigned	None			J1-24
84	P91		Not assigned	None			J1-25
85	P92		Not assigned	None			J1-26
							>

Figure 4-65 Filter for [Pin Number] Page



#### 4.5.9 **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.

🔇 Preferences		$\Box$ $\times$
type filter text	Pin Errors/Warnings	⇔ ◄ ⇔ ▼ 8
> Help	- Pin Conflict	
<ul> <li>Smart Configurator Component</li> </ul>	Multiple functions are assigned in one pin number	Frror ~
MCU/MPU Package	✓ No Pin Allocation	
Pin Errors/Warnings	Function used by software but not allocated to any pin	Error ~
	Mutually Exclusive Pins	
	Mutually exclusive pins cannot be allocated to the same pin at the same time.	Error 🗸
	- No Software	
	Assigned pins but there's no software using them	nfo 🗸
	- Different Group	
	Functions in same channel but different group	Warning \vee
	- Board Mismatch	
	Pin assignment does not match the board suggested pin assignment	Warning 🖂
< >>	Restore Defaults	Apply
	Apply and Close	Cancel

Figure 4-66 Pin Errors/Warnings Settings at Preferences

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





## 4.6 Interrupt Settings

The [Interrupt] page displays all interrupt 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".

- (1) To display the used interrupts only, click on the [K] (Show used interrupts)] button.
- (2) Group interrupts are collapsed in the interrupt table. Click on the [> (Open)] button to expand the view and see the interrupts in the group interrupt list.

nte	errupt configu	ration				Ge	enerate Code	Generate Rep
nt	terrupt vectors us	ed						(1)
	Type filter text				Vector Numb	er		~
	Vector Number	Vector Table	Interrupt	Interrupt request source	Periphe	Priority	Status	Bank specify
	3	0000AH	INTP1	Pin input edge detection	INTC	Low	Used	None
	> 18	00028H	INTSRE1/INTTM03H			Low	Used	None
	24	00034H	INTAD	End of A/D conversion	ADC	Low	Used	None
	<							>

Figure 4-68 [Interrupts] Page

#### 4.6.1 Changing Interrupt Priority Setting

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

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

rrupt configu	ration				Gei	nerate Code	denerate Re
errupt vectors us	ed						
Type filter text				Vector Numb	ber		
Vector Number	Vector Table	Interrupt	Interrupt request source	Periphe	Priority	Status	Bank specif
3	0000AH	INTP1	Pin input edge detection	INTC	Low	Used	None
> 18	00028H	INTSRE1/INTTM03H			Low	Used	None
24	00034H	INTAD	End of A/D conversion	ADC	Low	u Used	None
					High		
<				(2)	Level 1 Level 2		

Figure 4-69 Interrupt Settings



#### 4.6.2 Changing Interrupt Bank Setting

User can change the interrupt bank level on the [Interrupts] page using the following procedure:

- (1) Find the interrupt which user want to change bank setting on this page.
- (2) Click the [Bank specify] cell and select a bank setting from the drop-down list (There are four levels [None / 1 / 2 / 3])
- (3) If the same bank levels are selected for different interrupt priorities, a warning mark will be displayed, and warning message is displayed in [Remarks]. User should check and re-set the bank setting.

errup	t vectors									×.
Type f	ilter text					(1)			Vector Number	~
	Vector Tab 00004H	Interrupt INTWDTI	Interrupt request source Watchdog timer interval	Periph WDT	Priority Level 0 (high)		Bank specify None	Remarks		^
1	00006H	INTLVI	Voltage detection	LVD	Level 1	Used	@1	One register bank cannot be specified for multiple interrupt fun	ctions with different priorities.	
2	00008H	INTPO	Pin input edge detection	INTC	Level 2	Used	2	(3)		
3	0000AH	INTP1	Pin input edge detection	INTC	Level 3 (low)	Used	3	(3)		
4	0000CH	INTP2	Pin input edge detection	INTC	Level 3 (low)	Used	æ 1	One register bank cannot be specified for multiple interrupt fun	ctions with different priorities.	
5	0000EH	INTP3	Pin input edge detection	INTC	Level 3 (low)	Used	3			
6	00010H	INTP4	Pin input edge detection	INTC	Level 3 (low)	Used	None v			
7	00012H	INTP5	Pin input edge detection	INTC	Level 3 (low)	(2)	None			
> 8	00014H	INTST2/INT			Level 3 (low)		1			
> 9	00016H	INTSR2/INT			Level 3 (low)		2			
> 10	00018H	INTSRE2/IN			Level 3 (low)		3			
11	0001AH	INTELCL	Event link interrupt	ELCL	Level 3 (low)		None			~
view	Board Clocks	System Comp	onents Pins Interrupt							
		ns 🖾 Console								81.
-	warning, 0 oth									
riptic					^			Type		

Figure 4-70 Change Interrupt Bank Setting Example

The interrupt bank setting can be reflected into generated code in component's {ConfigurationName}\_user.c file.



Figure 4-71 Interrupt Bank Setting Example (IAR Project)

The concrete generated code specification is different for different compilers. User can get more information in corresponding IDE user guide.



## 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 (<sup>13</sup>) 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.

🌼 *Smart_Configurator_Example.scfg 🗙			- 8
Software component configurat	tion	Gene	rate Code Generate Report
Components 🚵 🛃 🖧 🕀 🕀	Configure		^
₩ 🗱 👘 😜	Comparator operation setting Stop	Operation	
✓	Resolution setting ① 10 bits	08 bits	() 12 bits
<ul> <li>✓</li></ul>	VREF(+) setting VDD	○ AVREFP	) Interna
Config_ADC	VREF(-) setting VREF() setting	○ AVREFM	
	Trigger mode setting Software trigger no wait mode Software trigger wait mode <	de	· ·
Overview Board Clocks System Compo	nents Pins Interrupt		
Console Sconfiguration Problems			
6 errors, 0 warnings, 0 others Description		<u></u>	
<ul> <li>Interrupt (2 items)</li> <li>E04010005: Interrupt vector used</li> <li>E04010005: Interrupt vector used</li> <li>Peripheral (2 items)</li> <li>E04010001: Peripheral A/D Convolution</li> <li>E04010001: Peripheral A/D Convolution</li> <li>E04010001: Peripheral A/D Convolution</li> <li>E04010003: Pin used by ANI0 in (2</li> </ul>	by INTAD0 in Config_ADC conflicts enter used by Config_ADC_Duplicate enter used by Config_ADC is already	with vector used by INTAD0 ir is already used by Config_AD0 used by Config_ADC_Duplicat	n Config_ADC_Duplicate. C. e.
8 E04010003: Pin used by ANI0 in 0	Config_ADC conflicts with pin used I	y ANI0 in Config_ADC_Duplica	te.

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.

in configuration							🕞 Generate Code 🛛 Gener	ate Repo
Software Compon 🕀 🖻 🖓	🔒 Pin Functi	on					ર 🔳	24 2
Type filter text	type filter	text (* = any	/ string,	? = any character)			All	~
🕆 📥 r_bsp	Enabled	Function	PIOR	Assignment	Pin Number	Direction	Remarks	Com ^
🔋 r_bsp		🛛 ANIO		P20/ANI0/AVREFP/EI20	/ 117	1	Multiple pin functions on th	1 I
V 📩 A/D Converter		ANI1		Not assigned	Not assigned	None		
Config_ADC		ANI2		Not assigned	Not assigned	None		
V To Ports		ANI3		Not assigned	Not assigned	None		
Config_PORT		ANI4		Not assigned	Not assigned	None		
		ANI5		Not assigned	Not assigned	None		
		ANI6		Not assigned	Not assigned	None		
	<					••		>

Figure 5-2 Pin Conflicts

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



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].

S	oftwar	e Co	omp	)	Ŧ	$\Box\downarrow^a_{\mathbf{Z}}$	品	Pin Fun	ctic	on	
	Type f	ilter	text	t				type fi	lter	text (* = any	
	∨ 🚣 r_bsp							Enable		Function	
	<ul> <li>■ r_bsp</li> <li>✓ ▲ A/D Converter</li> <li></li></ul>									🔇 P20	
										CCD00	
										CCD01	
	× 📥	-								CCD02	
		🖌 C	onfi	iç	A	ssign al	I			CCD03	
					ι	- Inassign	all			CCD04	
	Resolve con					conflict	flict				
Pir	n Functi	on	Pin	Nu	mbe	r					
Ov	erview	Boa	ard	Clo	cks	System	Comp	onents	Pin	s Interrupt	

Figure 5-4 Resolving Pin Conflicts

The pins of the selected node 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	- 8
Software component configu	Generate Code Generate Report	
Components 斗 🛓 🕀	Configure	^
हा हिंदू के प्रति क	Port selection PORT2	_
<ul> <li>✓ ➢ Startup</li> <li>▲ ➢ Generic</li> <li>▲ ➢ r_bsp</li> <li>▲ ➢ Drivers</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 a source file in <ProjectDir>\src\smc\_gen and IAR 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



Folder	File	Description
{ProjectName}	{ <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.
	{ <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.
	{ <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 IAR
		Embedded Workbench.
	main.c	This file generates once only in the first code generation.
		It contains function main ().
	buildinfo.ipcf	This file is always generated.
		It contains source file registration information. From Smart
		Configurator for RL78 V1.4.0, the name of .ipcf file is
		"buildinfo.ipcf". If user loads a project which created before
		Smart Configurator for RL78 V1.4.0, the .ipcf file will be re-
		generated as "buildinfo.ipcf" and the original .ipcf file
		({ <i>ProjectName</i> }.ipcf) still exist under the folder, won't be removed.
gonoral		
general		This folder is always generated. It contains header files and source files commonly used by
		Code Generator drivers of the same peripheral function.
	r_cg_xxx.h <sup>(Note*1)</sup>	The files contain macro definitions for setting SFR registers.
	r_smc_entry.h	This file is always generated.
	1_sinc_entry.in	This file includes the header files of Code Generator drivers
		that are added to the project.
		When using functions of Code Generator drivers in source
		files added by user, including this file is necessary.
	r_cg_macrodriver.h	This file is always generated.
	_ 0_	This header file contains common macro definitions used in
		drivers.
	r_cg_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_bsp		This folder is always generated.
		It consists of multiple subfolders (board, doc, mcu) with:
		- Initialization codes to start up the MCU before entering
		main () (e.g. setup stack, initialize memory)
		- Definitions of all SFR registers in iodefine.h (mcu folder)
		- Application note of r_bsp (doc folder)
		It also contains platform.h that will include r_bsp.h of the
		device used in the project.



Folder	File	Description
r_ <i>xxx</i> /		This folder is generated for the RL78 Software Integration
rm_xxx <sup>(Note*1)</sup>		System module that is added to the project.
		It consists of:
		- doc folder: Application note of this RL78 Software
		Integration System module
		<ul> <li>r_xxx.c/rm_xxx.c<sup>(Note*1)</sup>: RL78 Software Integration System module source file</li> </ul>
		<ul> <li>r_xxx.c/rm_xxx.h<sup>(Note*1)</sup>: RL78 Software Integration System header file</li> </ul>
		<ul> <li>r_xxx_api.h/rm_xxx_api.h<sup>(Note*1)</sup>: List of all API calls and interface definitions of this RL78 Software Integration System module</li> </ul>
r_config		This folder is always generated.
		It contains configuration header files for the MCU package,
		clocks, interrupts, and RL78 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 config inc	manually.
	r_bsp_config.inc	This file is always generated.
	r + / (Noto*1)	It generates configuration header file.
	r_xxx_config.h/rm_xxx_config.h <sup>(Note*1)</sup>	These are configuration header files for all RL78 Software Integration drivers/middleware that are added to the project.
r_pincfg	Pin.h	This file is always generated.
I_pincig	1 01.11	It is generated for supporting pin symbol and included in
	Pin.c	smc_entry.h. This file is always generated.
	FILLC	It is generated pin setting enabled in [Pins] page. It only
		generate pin setting which no need to set PIOR.
	r_ <i>xxx</i> _pinset.c	This file is RL78 Software Integration System module pin
		setting source file.
	r_xxx_pinset.h	This file is RL78 Software Integration System module pin setting header file.
{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
		driver-specific, e.g. start (R_ConfigName_Start) and stop
		(R_ConfigName_Stop).
	{ <i>ConfigName</i> }_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 { <i>ConfigName</i> }.c and { <i>ConfigName</i> }_user.c.
	1	

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).

*Smart_Configurator_Ex Clocks configuration			1	
clocks configuration	•		Generate Code	Generate Report
Operation mode:	High-speed main mode 4.0(V)~5.5(V)	•		
	4.0 V ≤ EVDD0 ≤ 5.5 V	•		
High-speed on-ch	ip oscillator	7	fiHP	
Frequency:	32 ~ (MHz)		- 32 (MHz	
			fMAIN 32 (MHz)	
			fCLK 32000 (kHz)	
Middle-speed on-			fIMP	
Frequency:	4 ~ (MHz)		4 (MHz)	
X1 oscillator		Divider		
Operation mode:	X1 oscillation v	x1 •	fMXP - (MHz	
Frequency: Stable time:	5 (MHz) 2^18/fx = 262144(µs)		- (MHz	
	a tay m			
Low-speed on-chip or	scillator		fiL	
Frequency:	32.768 (kHz)		32.768 (kHz)	
			fSXP	
XT1 oscillator Operation mode:	XT1 oscillation		— 32.768 (kHz)	
Frequency:	32.768 (kHz)		fSXR 	
XT1 oscillation mode:	Low power consumption 1 *		52.700 (kH2)	
Supply mode:	Enables supply in STOP,HALT mode +			
Overview Board Clocks	ystem Components Pins Inf	terrupt		
E/*	Selection of middl	e-speed on-chip oscillator clock frequen	cv	
	Middle-speed on-ch	ip oscillator frequency select register (		
0	MOCODIVI, MOCODIVO	MHz		
1		2MHz		
2		MHz		
Ot	ther than above : S	ecting prohibited		
#de	fine BSP CFG MOCO	DIVIDE (0) / Generated value. Do not ed	it this man	ually */

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. r_bsp will handle the clock initialization before entering main ().
		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_bsp</i> 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) Pin initialization for drivers with {ConfigName}

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 ().

Image: Smart_Configurator_Example.scfg ≥								- 0
Pin configuration							Generate Code Gene	rate Report
Software Compon 🕒 📄 🗸 🔜	Pin Functi	on					<b>3</b>	1 2 2
	type filter	text (* = any	string, ? = any cl	haracter)			All	$\sim$
∽ 🚣 r_bsp	Enabled	Function	PIOR	Assignment	Pin Number	Direction	Remarks	
i r_bsp x ♣ Square Wave Output		TO01		P16/E016/CCD00/TI01/T001/INTP5/SI00/RxD0	/ E4	0		
Config_TAU0_1								
	<							>
Pin Function Pin Number								
Overview Board Clocks System Compo	nents Pins	Interrupt						

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) Pin initialization for RL78 Software Integration System component

Pin functions are initialized in R\_{*PeripheralName*}\_PinSetInit of the file \src\smc\_gen\r\_pincfg\{*ConfigName*}\_pinset.c.

User will call the pin initialization codes in main ().

in configuration						Ger	nerate Code	Generate R	lepor
Software 🗊 🖻 📲	Pin Functi	on					3	11   <b>11</b>   2	. 4
Type filter text	type filter text (* = any string, ? = any character)						Al	1	~
✓ 💑 r_bsp ♥ r_bsp	Enabled	Function TS00	PIOR	Assignment P P50/TS00/EI50/EO50/CCD03/INTP1/SI11/SDA11	Pin Number Ø 33	Direction O	Remarks	Comments	^
✓ 📥 r ctsu		TS01		/ P31/TS01/EI31/TI03/TO03/INTP4/PCLBUZ0	/ 21	0			
💕 r_ctsu		TSCAP		P30/VCOUT0/TSCAP/EI30/INTP3/RTC1HZ/SCK11/SCL11	/ 32	0			¥

rview Board Clocks System Components Pins Interrupt

Figure 6-5 *r\_ctsu* in Software Components View

Folder	File	Function	Component type	Description
r_pincfg	{ <i>ConfigName</i> } _pinset.c	R_{ <i>PeripheralName</i> }_ PinSetInit	RL78 Software Integration System	This API function initializes the pins used by this driver. User need call this function in main () function.



## 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}\{ConfigName}.c.

rrupt configu	ration						
errupt vectors							
Type filter text				Vector N	lumber		
Vector Number	Vector Table Address	Interrupt	Interrupt request source	Peripheral	Priority	Status	Bank spe
0	00004H	INTWDTI	Watchdog timer interval	WDT	Low		None
1	00006H	INTLVI	Voltage detection	LVD	Low		None
2	00008H	INTP0	Pin input edge detection	INTC	Low		None
3	0000AH	INTP1	Pin input edge detection	INTC	Low		None

Overview Board Clocks System Components Pins Interrupt

### Figure 6-6 Interrupts Configuration in Interrupts View

ltem	Folder	File	Component type	Description
Priority	{ConfigName}	{ConfigName}.c	Code Generator	It is initialized in R_ <i>ConfigName</i> _Create of this file. r_cg_systeminit will call this function before entering <i>main ()</i> function.
Bank	{ConfigName}	{ <i>ConfigName</i> }_user. c	Code Generator	Declaration of interrupt as: #pragma vector = "InterruptVectorName"_vect #pragma bank = "bankNumber" please see example in Figure 4-71



## 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				×
$\leftarrow$ $\rightarrow$ $\checkmark$ $\uparrow$ $\square$ $\ll$ IAR $\Rightarrow$	Smart_Configurator_Example	ب< 5	Search Smart_Config	urator
Organize 🔻 New folder				. ?
Desktop ^	Name	Date modified	Туре	Size
Documents	settings	10/10/2022 14:58	File folder	
🖊 Downloads	Debug	10/10/2022 17:03	File folder	
b Music	settings	11/10/2022 09:11	File folder	
Pictures	src	23/09/2022 13:49	File folder	
Videos	Smart_Configurator_Example.eww	23/09/2022 13:49	IAR IDE Workspace	1
: (C:) Windows				
×	<			>
File name	Smart_Configurator_Example.eww	~ Wo	rkspace Files (*.eww)	$\sim$
			Open Ca	ncel .:

Figure 7-1 Load a \*.eww File

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



Figure 7-2 New Files Added to IAR Workspace



- (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.

Category: General Options Static Analysis					
C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger COM Port E1 E2 COM Port E20 E2 Litt / E2 On-board E2-CUBE2 E2-CUBE2 IECUBE Simulator	Target Device RL78 con Code mo Near Use	Device RL78 core S3 - Unspecified the		MISRA-C:2004 MISRA-C:199 nfiguration Library Options 1 Floating-point Size of type 'double':	
			addresses St	art address: xF3000	End address: 0xF9EFF

Figure 7-3 Change Target Device



# 7.2 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	🌣 • ^ int main( void )
🗉 🌒 Smart_Configurator_Example	
Renesas_SC Renesas_SC Config_ADC General Config_ADC General Config_ADC Config Config_ADC Config_ADC Config Conf	Qptions     rn     0;       Make
⊢⊞ <b>≡</b> mcu ├── <b>ⓑ</b> platform.h	<u>R</u> ename
Smart_Configurator_Example	Version Control System
Build	Open Containing Folder
	File Properties
Messages Linking	S <u>e</u> t as Active
Total number of errors: 0 Total number of warnings: 0 <	
Build Debug Log	

Figure 7-4 Build C Project File in IAR



## 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).



/**************************************
Includes ************************************
<pre>#include "r_cg_macrodriver.h"</pre>
#include "r_cg_userdefine.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 */
/**************************************
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 INTAD interrupt service routine.
* Arguments: None * Return Value: None
***************************************
#pragma vector = INTAD_vect
interrupt static 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 */ }
J
/* 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 RL78 Software Integration System Modules and Code Generator, follow the below steps:

1) Open the {*Project name*}.c file, add code to include the header files of the modules user wants to use.

In case of RL78 Software Integration System Modules, it is r\_xxx.h.

In case of Code Generator, it is added for you in "r\_smc\_entry.h" by automatically.

Workspace 🔻 🗭 🦕	main.c X	•
Debug 🗸		fo
Files       Image: Signature State Sta	<pre>#include "r_smc_entry.h" #include "r_ctsu.h" int main(void) {     return 0; }</pre>	
Smart_Configarator_Sample	4	Þ

Figure 8-1 Add Header Files

2) In the main function, call the functions generated and add application codes.

**In case of Code Generator**, 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*).

Workspace 👻 🖣 🗙	main.c* x Config_ADC.c	,
Debug 🗸 🗸	f	0
Second       Image: Second secon	<pre>#include "r_smc_entry.h" int main(void) {     R_Config_ADC_Start();     return 0; }</pre>	
Smart_Configarator_Sample	4	

Figure 8-2 Call Code Generator Functions

**In case of Software Integration System Modules,** refer to the examples provided in the "API Functions" chapter of corresponding Application Note.

For more reference, refer to "Smart Configurator Application Examples" in "chapter 13 Documents for Reference".



## 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 (Generate Code)] button. <Date-and-Time> indicates the date and time when the backup folder is created after code generation.



## 10. Generating Reports

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

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

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

Smart	Configurator_Example.scfg ×	
Overvi	ew information	ت نے Generate Code Generate Report
- Gener	al Information	
	<b>Overview</b> Get an <u>overview</u> of the features provided by Smart Configurator.	
	Videos Introduction to Smart Configurator Browse related videos What's New Check out what's new in the latest release. See all <u>Release Notes</u> . Current version: V1.13.0 Product Documentation and FAO	Application Code Software Components Middleware & Drivers Device Drivers Device Drivers MCU Hardware
- Currer	User's Guide API manual FAQ : Smart Configurator ht Configuration	
	he code generation behavior can be customized	
	d board/device: R7F100LPLxFB (ROM size: 512)	
	ed location (PROJECT_LOC\): src\smc_gen	Edit 🗸
Overview	Board Clocks System Components Pins Inter	rupt

Figure 10-1 Output of a Report on the Configuration (as a PDF/Text File)

Smart Report       -       ×         Generate report of configurations       Options            • Print all sections        •         Print specific sections        •         Board       •         Clocks       •         System       •         > ✓ Components       •         •       ● Pins         □ Internet       ✓         ✓ Output as PDF       Select Font         ○ Output as text       ●         C:\       ●					
Options <ul> <li>Print all sections</li> <li>Print specific sections</li> <li>Board</li> <li>Clocks</li> <li>System</li> <li>Components</li> <li>Prins</li> <li>Voltput as PDF</li> <li>Select Font</li> <li>Output as text</li> </ul>	💦 Smart Report				$\times$
● Print all sections     Print specific sections     Board     Clocks     System     ✓ Components     ● Pins     Unterrunt     Output as PDF     Select Font     Output as text	Generate report of configurations				
● Print all sections     Print specific sections     Board     Clocks     System     ✓ Components     ● Pins     Unterrunt     Output as PDF     Select Font     Output as text					
● Print all sections     Print specific sections     Board     Clocks     System     ✓ Components     ● Pins     Unterrunt     Output as PDF     Select Font     Output as text					
<ul> <li>○ Print specific sections</li> <li>○ Board</li> <li>○ Clocks</li> <li>○ System</li> <li>&gt; ○ Components</li> <li>&gt; ○ Pins</li> <li>✓ Output as PDF</li> <li>○ Output as text</li> </ul>	Options				
Board         Clocks         System         ✓ Components         ✓ Components         ✓ Internation         ✓ Output as PDF         Output as text	Print all sections				
Clocks System Components Components  Pins Coutput as PDF Coutput as text	O Print specific sections				
System     ✓ Components     Pins     Interrupt     Output as PDF     Select Font     Output as text	Board				^
	Clocks				
Pins      Interrupt      Output as PDF      Output as text					
□ Interrupt     ✓       ✓ Output as PDF     Select Font       □ Output as text     ✓	Components				
Output as PDF     Select Font       Output as text     Output as text					
Output as text					
	Output as PDF			Select	Font
C:\ Browse	Output as text				
	C:\			Brow	wse
			_		
OK Cancel		OK		Cancel	

Figure 10-2 Dialog Box for Output of a Report (Example is selecting "Output as PDF")



## 10.2 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 [ III (Save the list to .csv file)] button on the [Pins] page of the Smart Configurator view.

n configuration						Generate Co	de Generate	Repo
lardware Resource 🛭 🕀 🕌	Pin Functi	on				2		2
	type filter	text (* = ;	any stri	ng, ? = any character)			All	~
🚣 All 🔷	Enabled	Functi	PIOR	Assignment	Pin Number	Direction	Remarks	^
🤯 Clock Generator		ANI0		P20/ANI0/AVREFP/EI2	/ 117	1		
✓ G Comparator		ANI1		Not assigned	Not assigned	None		
COMP0		ANI2		Not assigned	Not assigned	None		
COMP1		ANI3		Not assigned	Not assigned	None		
<ul> <li>Capacitive Touch Sensing</li> <li>Voltage Regulator</li> <li>Remote Control Signal Re</li> </ul>		ANI4		Not assigned	Not assigned	None		
		ANI5		Not assigned	Not assigned	None		
		ANI6		Not assigned	Not assigned	None		
🚺 On-Chip Debug		ANI7		Not assigned	Not assigned	None		
Power Supply		ANI8		Not assigned	Not assigned	None		
Interrupt Function V	<	ΔΝΙ9		Not assigned	Not assigned	None		>
								-

Figure 10-3 Output of a List of Pin Functions or Numbers (in csv Format)

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

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



Figure 10-4 Outputting a Figure of MCU/MPU Package (in png Format)



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

The Smart Configurator for RL78 V1.6.0 and the later version now incorporates an enhanced user code protection feature. This feature empowers users to insert codes to any location in the generated codes by utilizing the specific tags, as shown in Figure 11-1. 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 only be supported on the files that are generated by the "Code Generation component".

## 11.1 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 11-1 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 11-1 Specific Tags for User Code Protection Feature

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

Figure 11-2 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 11-1. 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 11-2 User Code Protection with Auto Merge



## 11.3 What to Do When Merge Conflict Occurs

#### 11.3.1 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 in red will be displayed in the Smart Configurator console, as shown in Figure 11-3 The Merge Conflict Message Outputted in the Smart Configurator Console.

📮 Console 🔝 Configuration Problems	🖹 🔐 🛃 🖃 🛨 📑 🛨 🗖 🗖
Smart Configurator Output	
M00000002: Code generation is successful:C:\cases\tempcases\rcp temp case\g24\src\smc gen	^
M04050003: ****FAA memory area Usage****	
Code size: 168	
Data size: 106	
M05000001: Pin 32 is assigned to RTC1HZ	
M0000001: Code generation is started	
M04000001: File generated:src\smc gen\Config RTC\Config RTC.h	
M04000001: File generated: <u>src\smc gen\Config RTC\Config RTC.c</u>	
M00000005: The above files highlighted in red color have user code merge conflicts, please ope	en the file and resolve the conflict manually
M00000002: Code generation is successful: <u>C:\cases\tempcases\rcp temp case\g24\src\smc gen</u>	×

Figure 11-3 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 11.3.2 Steps for Resolving the Merge Conflict.



#### 11.3.2 **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 11-4 Code before Resolving Conflict).
- (2) Click on "Copy Current Change from Left to Right" (Figure 11-4 Code before Resolving Conflict).



Figure 11-4 Code before Resolving Conflict

(3) Delete the codes that user does not want to use (Figure 11-5 Code after Applying "Copy Current Change from Left to Right").

i g24_example.scfg f <sup>□</sup> *File Compare ×	- 0
🕂 Text Compare	M 📰 😤 🔛 🗞 🎪 🕸
Existing code	New code
<pre>/* Clear INTRIC interrupt flag */ RTCIF = 01; /* Clear INTRIC interrupt flag */ RTCIF = 0; /* Set fATCCK */ RTCC0 = 09 RTC_CLK_32KHZ; /* Set 12-/24-hour system and period of Constant-period interrupt (INTRIC) */ /* Set 12-/24-hour system and period of Constant-period interrupt (INTRIC) */ /* Set 12-/24-hour system and period of Constant-period interrupt (INTRIC) */ /* Set 12-/24-hour system and period of Constant-period interrupt (INTRIC) */ /* Set 12-/24-hour system and period of Constant-period interrupt (INTRIC) */ /* Set 12-/24-hour system and period of Constant-period interrupt (INTRIC) */ /* Set 12-/24-hour system and period of Constant-period interrupt (INTRIC) */ /* Set 12-/24-hour system and period of Constant-period interrupt (INTRIC) */ /* Set 12-/24-hour system and period of Constant-period interrupt (INTRIC) */ /* Set 12-/24-hour system and period of Constant-period interrupt (INTRIC) */ /* Set 12-/24-hour system and period of Constant-period interrupt (INTRIC) */ /* Set 12-/24-hour system and period of Constant-period interrupt (INTRIC) */ /* Function Name: R_Config_RTC_Start * Function Name: R_Config_RTC_Start * Function Name: R_Config_RTC_Start * Return Value : None ************************************</pre>	<pre>/* Set farCCK */ RTCCB = 00 pTC_CLL_32kH2; /* Set 12-724-hour system and period of Constant-period interrupt (INTRTC) */ RTCCB = 00 pTC_LL_32kH2; /* Set 12-724-hour system and period of Constant-period interrupt (INTRTC) */ RTCCB = 00; (3). Add used code /* Set alarm detect function */ WALE = 00; WALTE = 00; /* Alarm function setting */ ALARMMM = _00_RTC_ALARM_NUN; ALARMMM = _00_RTC_NTARC_ALARM_NUN; ALARMMANANN = _00_RTC_ALARM_NUN; ALARMMANANN = _00_RTC_ALARM_NUN; ALARMMANN = _00_RTC_NTARC_ALARM_NUN; ALARMMANN = _00_RTC_ALARM_NUN; ALARMANN = _00_RTC_NTARC_ALARM_NUN; ALARMANN = _00_RTC_NTARC_ALARM_NUN; ALARMANN = _00_RTC_NTARC_ALARM_NUN; ALARMANN = _00_RTC_NTARC_ALARM_NUN; ALARM_NUN; ALARM_NUN; ALARMANNN; ALARMANNN; ALARMANNN;</pre>
<pre>Vola K_contg_m(c_start(vola) {     /* Enable RTC clock operation */     RTCE = 10;     c</pre>	KTCC0 = [00_KTC_MTURDLE ]_00_KTC_12HOUR_MODE ]_00_KTC_MTURDLE]_00_KTC_MTURDLE           /* Statuser code */           /* End-eser code */
E Console 🔝 Configuration Problems	
Smart Configurator Output M04050001: CNDW 0 is used for One shunt feature M04050001: CNDW 1 is used for One shunt feature M04050001: CNDW 2 is used for One shunt feature M04000001: Code generation is started M04000001: File generatedisrc\smc gen\Config RTC.c	

M0000005: The above files highlighted in red color have user code merge conflicts, please open the file and resolve the conflict manually

Figure 11-5 Code after Applying "Copy Current Change from Left to Right"



(4) Save the modified code (Figure 11-6 Code after Deleting and Saving).

₿ g24_example.scfg  🚰 *File Compare ×		
🖓 Text Compare		M 🚍 😤 🔛 🐼 🏖 4
xisting code	٠	New code
<pre>/* Clar INTRC interrupt flag */ RTCF = 0; /* Set fRTCCK */ RTCC = _00 RTC_L(X_32KH2; /* Set 12-724-hour system and period of Constant-period interrupt (INTRTC) */ r* Set 12-724-hour system and period of Constant-period interrupt (INTRTC) */ r* Set 12-724-hour system and period of Constant-period interrupt (INTRTC) */ POR3 &amp; 0xFEU; /* Set 12-724-hour system and period of Constant-period interrupt (INTRTC) */ RTCC = _00 RTC_RTC_RTC_12+RULE _00 RTC_12+RULE PODE _ 00-RTC_INTRTC_HOT GENERATE); /* Set 12-824-RULE _00 RTC_12+RULE PODE _ 00-RTC_INTRTC_HOT GENERATE); /* Constant user code */ R_Config_RTC_Create_UserInit(); /* Set 12-824-RULE _00 RTC_Start Description : This function enables the real-time clock. * Arguments : None * Return Value : None</pre>	] \	<pre>/* Set fRTCCK */ RTCCB = 000 RTC CLK_32KHZ; /* Set 12-/24-hour system and period of Constant-period interrupt (INTRTC) */ RTCCB  = (20 RTC RTCHM2EHABLE   _00_RTC_12HOUR_MODE   _00_RTC_INTRTC_NOT_GENERATE); /* Statuser code */ PON3 &amp;= 0xFEU; /* Set alarm detect function */ WALE = 0U; WALTE = 0U; WALTE = 0U; WALTE = 0U; ALARMMM = _00_RTC_ALARM_MIN; ALARMMM = _02_RTC_ALARM_MIN; ALARMMM = _02_RTC_ALARM_MEEK; /* Set RCHATE in */ PON3 &amp;= 0xFEU; P3 &amp;= 0xFEU; P3 &amp;= 0xFEU;</pre>
void R_Config_RTC_Start(void) {     /* Enable RTC clock operation */     RTCE = 1U;     <		R_Config_RTC_Create_UserInit(); }
Console 🔝 Configuration Problems		🗎 🔒
nart Configurator Output		
040550001: (NDW 0 is used for One shunt feature 04050001: (NDW 1 is used for One shunt feature 04050001: (ADW 2 is used for One shunt feature 00000001: (Ode generation is started 04000001: He generated:src\swc gen\Config RTC\Config RTC.c		

Figure 11-6 Code after Deleting and Saving

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

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



## 12. Help

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



Figure 12-1 Help Menu

The help system can also be activated from the [Overview information] page by clicking without button.



Figure 12-2 Quick Start

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



## 13. 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: RL78 API Reference (R20UT4852)

Obtain the latest version of IAR Embedded Workbench for Renesas RL78 manual from the website of IAR.

SMS & ELCL Application Notes:

Obtain the latest information from the website of Renesas Electronics.



# **Revision History**

Rev.	Section	Description
1.00	-	First edition issued
1.01	Section 2 Installing and Uninstalling the Smart Configurator	Removed 2.1 Creating a New IAR C Project File
	Section 3 Operating the Smart	3.1 Procedure for Operations: Updated Figure 3-1. Operating Procedure
	Configurator	3.3 Creating and Loading a Configuration File: description was updated
	Section 4 Setting of	4.4.13 Export Component Configuration: description was updated
	Peripheral Modules	4.4.14 Import Component Configuration: description was updated
		4.4.15 Configure General Setting of Component: description was updated
	Section 6 Generating Source Code	6.2 Configuration of Generated Files and File Names: description was updated
	Section 7 Loading Generated Files in Integrated Development Environment	7. Loading Generated Files in Integrated Development Environment: description was updated
1.02	Section 4 Setting of	4.4.12 Changing Version of BSP Configuration: Note was deleted.
	Peripheral Modules	4.4.15 Configure General Setting of Component: Figure 4-38 Configure General Setting of Component was updated.
		4.4.15 Configure General Setting of Component: Note 1 was updated.
		4.4.15 Configure General Setting of Component: Note 2 was updated.
		4.4.15 Configure General Setting of Component: Note 3 was added.
		4.6.2 Changing Interrupt Bank Setting:
		The description of step (3) was updated.
		4.6.2 Changing Interrupt Bank Setting:
		Figure 4-55. Change Interrupt Bank Setting Example was modified.
1.03 Section 3 Operating the Smart Configurator	3.4.4 MCU/MPU Package View: Update description and Figure 3-7. MCU/MPU Package View	
	Section 4 Setting of	4.1.2 Selecting the Board: modify description
	Peripheral Modules	4.4.3 Removing Software Component: Add description about removing multiple components from a project
		4.4.10 Downloading RL78 Software Integration System Modules: Update description
		Add 4.4.11 Adding a RL78 Software Integration System Module
		4.4.12 Setting a RL78 Software Integration System Module: Update description
		4.5 Pin Settings: Update description and Figure 4-50 and 4-51.
		4.5.3 Assigning Pins Using the MCU/MPU Package View: Update description and Figure 4-55.
		Add 4.5.4 Show pin number from pin functions.
		Add 4.5.9 Pin Errors/Warnings setting.
	Section 6 Generating Source Code	6.2 Configuration of Generated Files and File Names: Update the description and Figure 6-3 Configuration of Generated Files and File Names for supporting pin symbol.
	Section 8 Creating User Programs	Add 8.2 Using Generated Code in User Application
	Section 11 User code protection feature for Smart Configurator Code Generation component	Add Section 11 User code protection feature for Smart Configurator Code Generation component.

Rev.	Section	Description
1.04	Section 4 Setting of Peripheral Modules	Update 4.4.8 ELCL Fixed Function Modules Download and 4.4.9 Setting a Fixed Function ELCL Component
		Add 4.4.10 Create and Edit ELCL Flexible Circuit
	Section 6 Generating Source Code	Update 6.2 Configuration of Generated Files and File Names

### 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 system-evaluation 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.
- 6. 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

"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, 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>.