

RL78 Smart Configurator

User's Guide: CS+

R20AN0580EC0103 Rev.1.03 Apr.21.25

APPLICATION NOTE

Introduction

This application note describes the basic usage of the RL78 Smart Configurator (hereafter called the Smart Configurator), and the procedure for adding its output files to CS+ projects.

References to the Smart Configurator and CS+ integrated development environment in this application note apply to the following versions.

- CS+ (CS+ for CC) V8.13.00 and later
- RL78 Smart Configurator V1.13.0 and later
- CS+ RL78 Smart Configurator Communication Plugins V1.00.11 and later

Target Devices

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

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



Contents

1. Over	view	5
1.1 Pu	rpose	5
1.2 Fe	atures	5
1.3 So	ftware Components	5
0 D (•
	re Using the Smart Configurator	
	eparing the CS+ (CS+ for CC) Integrated Development Environment	
	talling the Smart Configurator	
	tting the CS+ Integrated Development Environment	
2.3.1	Checking the Plug-in Settings	
2.3.2	Checking the Setting of the Execution Path	
2.4 Un	installing the Smart Configurator	7
3. Oper	ating the Smart Configurator	8
•	ocedure for Operations	
	arting the Smart Configurator	
	e to be Saved as Project Information	
	ndow	
3.4.1	Main Menu	
3.4.2	Toolbar	
3.4.2	Smart Configurator View	
3.4.3 3.4.4	MCU/MPU Package View	
	Console View	
3.4.5	Configuration Problems View	
3.4.6	Conliguration Problems view	14
4. Setti	ng of Peripheral Modules	15
4.1 Bo	ard Settings	.15
4.1.1	Selecting the Device	.15
4.1.2	Selecting the Board	.16
4.1.3	Exporting Board Settings	.17
4.1.4	Importing Board Settings	.18
4.2 Clo	ock Settings	.19
4.3 Sy	stem Settings	.20
4.4 Co	mponent Settings	.22
4.4.1	Switching Between the Component View and Hardware View	.22
4.4.2	Adding a Software Component	
4.4.3	Removing Software Component	
4.4.4	Setting a Code Generator Component	
4.4.5	Changing the Resource for a Code Generator Configuration	
4.4.6	Setting SNOOZE Mode Sequencer (SMS) Component	
4.4.7	Update SMS Data Files	
4.4.8	ELCL Fixed Function Modules Download	
4.4.9	Setting a Fixed Function ELCL Component	
	Create and Edit ELCL Flexible Circuit	
	Downloading RL78 Software Integration System Modules	
	Adding a RL78 Software Integration System Module	
7.4.12	Adding a rep o convare integration cystem would and an and a second seco	71



4.4.13 Setting a RL78 Software Integration System Module	42
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 Pin Settings	47
4.5.1 Changing the Pin Assignment by PIOR Function	48
4.5.2 Changing the Pin Assignment of a Software Component	49
4.5.3 Assigning Pins Using the MCU/MPU Package View	50
4.5.4 Show Pin Number from Pin Functions	51
4.5.5 Exporting Pin Settings	52
4.5.6 Importing Pin Settings	52
4.5.7 Pin Setting Using Board Pin Configuration Information	53
4.5.8 Pin Filter Feature	54
4.5.9 Pin Errors/Warnings setting	55
4.6 Interrupt Settings	56
4.6.1 Changing Interrupt Priority Setting	56
4.6.2 Changing Interrupt Bank Setting	57
5. Managing Conflicts	
5.1 Resource Conflicts	
5.2 Resolving Pin Conflicts	59
6. Generating Source Code	60
6.1 Registering Generated Source Code with CS+	
6.2 Configuration of Generated Files and File Names	
6.3 Initializing Clocks	
6.4 Initializing Pins	
6.5 Initializing Interrupts	
7. Creating User Programs	67
7.1 Adding Custom Code	
7.2 Using Generated Code in User Application	69
8. Backing up Generated Source Code	71
8. Backing up Generated Source Code	/ 1
9. Generating Reports	72
9.1 Report on All Configurations (PDF or Text File)	
9.2 Configuration of Pin Function List and Pin Number List (in csv Format)	
9.3 Image of MCU/MPU Package (in png Format)	
10. User Code Protection Feature for Smart Configurator Code Generation Component	74
10.1 Specific Tags for the User Code Protection Feature	
10.2 Examples of Using User Code Protection Feature to Add New User Code	
10.3 What to Do When Merge Conflict Occurs	
10.3.1 What is Merge Conflict	
10.3.2 Steps for Resolving the Merge Conflict	76
11. Help	78



RL78 Smart Configurator	User's Guide: CS+
12. Documents for Reference	



1. Overview

1.1 **Purpose**

This application note describes the basic usage of the Smart Configurator and CS+ integrated development environment, including the procedure for creating a project and adding Smart Configurator output to CS+ projects.

Refer to the User's Manual of CS+ for how to use CS+.

1.2 **Features**

The Smart Configurator is a utility for combining software to meet your needs. It handles the following three functions to support the embedding of drivers from Renesas in your 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:

(1) Code Generator drivers (DTC, A/D Converter, Interrupt Controller, etc.)

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. Before Using the Smart Configurator

2.1 **Preparing the CS+ (CS+ for CC) Integrated Development Environment**

To create or build a program in the CS+ integrated development environment with the use of source code generated by the Smart Configurator, you will need to install CS+ to handle building for the target device.

2.2 Installing the Smart Configurator

Download the RL78 Smart Configurator and CS+ RL78 Smart Configurator Communication plug-in from the URL below. The CS+ RL78 Smart Configurator communication plug-in is required for registering source code generated by the Smart Configurator with CS+.

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

After activating the installer, install the Smart Configurator and the plug-in by following the procedure of the installer. You will require administrator privileges to do this.

2.3 Setting the CS+ Integrated Development Environment

Source files the Smart Configurator generates can be registered with CS+, and CS+ can be set to the configuration required to build the registered source files. This is set up automatically at the time the Smart Configurator is installed; however, you will need to check the settings against the following and modify them as required.

2.3.1 Checking the Plug-in Settings

Select [Plug-in Manager] from [Tool] of CS+ menu and confirm that there is a tick against "Smart Configurator for RL78 Communication Plug-in". Tick it if it is not.

Plug-in Manager	×
Checked plug-ins are loaded at the CS+ start-up. These settings are enabled at the next start-up. * You can never uncheck a check box of the grayout plugin that is required recommended that the checkboxes of the plug-in for the target microcontro Basic Function Additional Function	
Module Name	Description
Code Generator Plug-in Code Generator Plug-in for RH850 Code Generator/PinView Plug-in Debug Console Plug-in Code Generator/PinView Plug-in Common Code Generator/PinView Plug-in Code Generator/PinView Plug-	Plug-in to generate the device driver automatically. (for V850, Plug-in to generate the device driver automatically and to vier Plug-in to generate the device driver automatically and to vier DebugConsole plug-in to support using standard I/O. SEditor DLL It is a console where the IronPython commands and the CS+ Plug-in to define the device pin configuration. Plug-in to analyze program. Plug-in for application development that contains useful tools Realtime OS Analysis Control plug-in to use Realtime OS Infor Realtime OS plug-in to display Realtime OS resource informat Plug-in to analyze the Realtime OS built-in program. Plug-in to communicate with Smart Configurator for RL78 that Plug-in to communicate with Smart Configurator for RL78 that Plug-in to communicate with Smart Configurator for RX that g- Utility to display and adjust stack usage of each functions. Plug-in to communicate with CS+ Update Manager.
	OK Cancel Help

Figure 2-1 Plug-in Manager



2.3.2 Checking the Setting of the Execution Path

[Smart Configurator (Design Tool)] is displayed under [Project name (Project)] in the Project Tree when you open the CS+ project for the target device of the Smart Configurator.

Click on [Smart Configurator (Design Tool)], and the Smart Configurator Property panel is displayed.

🚳 Smart Configurator Example - CS+ for CC -	Property]	– 🗆 X
File Edit View Project Build Debug Tool	/indow Help	🤤 🧭 🙆
🚳 Start 🛃 🔚 🎒 🐰 🖻 🚳 🔊 (*	🔠 🏯 🚔 🔹 💌 🔽 Defa	ultBuild 🔹 🖌 🦌 🖓 🗘 🦏 🔘 🔊 🔊 🖓 🧐 🖘 🖓
i 💎 🖓 🖉 🦃 🤻 🗀 🗭 🗣 🔍 (5 i i	Solution List 🕴 🖓 🚰 🚰	
	Property	• X
Smart 2 @ 2 2	Smart Configurator Property	P - +
Smart Configurator Example (Project)	✓ Product Information	
2 R7E100GSNxEB (Microcontroller)	Version V1.00.00.	0.00 [23 Sep 2020]
Smart Configurator (Design Tool)	 Smart configurator setting 	
CC-RL (Build Tool)	Smart Configurator for RL78 executable file path D:\Applic	cation\IDE\CS+E8.05.00A\SmartConfigurator\RL78\eclipse\SmartConfigurator.exe 🛄
- Program Analyzer (Analyze Tool)		
E- File		
	10	

Figure 2-2 Displaying the Property

"Smart Configurator for RL78 executable file path" shows the executable file of the Smart Configurator. The following path is set when the Smart Configurator is installed with the default setting (where "CS+" and "SmartConfigurator" are in the same level).

➢ 64-bit environment:

"C:\Program Files (x86)\Renesas Electronics\SmartConfigurator\RL78\eclipse\SmartConfigurator.exe"

When manually specifying the path of the executable file, "Smart Configurator for RL78 executable file path" can be set as either a relative or an absolute path.

2.4 Uninstalling the Smart Configurator

If you wish to uninstall the Smart Configurator, select "Smart Configurator for RL78" and "CS+ Smart Configurator Communication Plugins for RL78" from [Apps & features] in your PC's Windows Settings Apps control panel and uninstall them.



3. Operating the Smart Configurator

3.1 **Procedure for Operations**

Figure 3-1 shows the procedure for using the Smart Configurator to generate files for setting up peripheral modules, and to use them in building after registration with CS+. Refer to the related documents on CS+ for the operation of CS+.

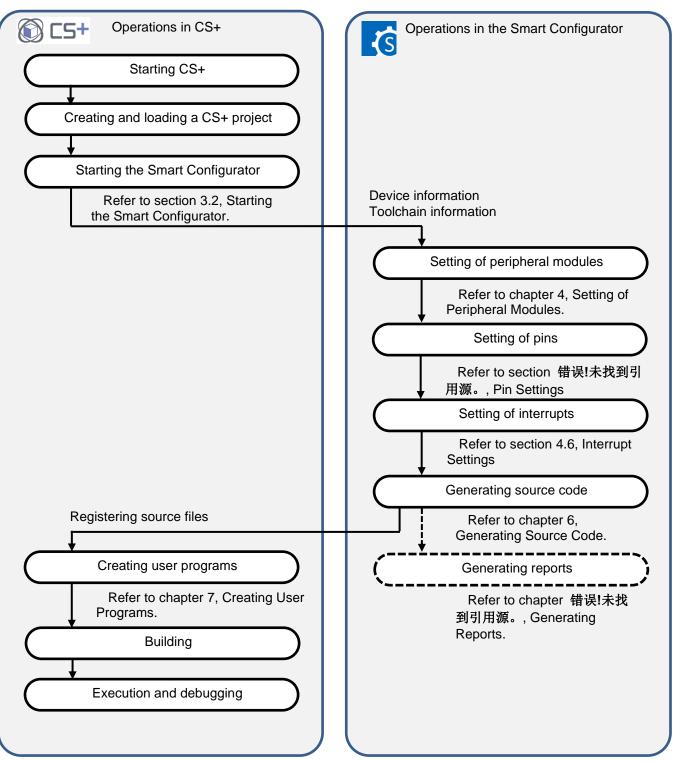


Figure 3-1 Procedure for Operations



3.2 **Starting the Smart Configurator**

Double-click on [Smart Configurator (Design Tool)] under [Project name (Project)] in the Project Tree of CS+ to start the Smart Configurator. You do not need to select a device or toolchain for the Smart Configurator, since the settings of the project for CS+ are passed over to the Smart Configurator.

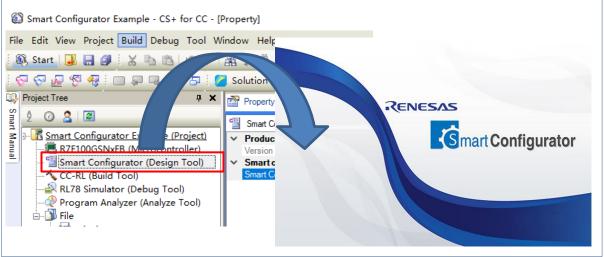


Figure 3-2 Activation of Smart Configurator

Note: The settings of CS+ are not passed over to the Smart Configurator in the following cases: when the Smart Configurator is activated from its executable file, when a new project is created from [File] menu of the Smart Configurator, or when an existing file from the Smart Configurator is opened.

3.3 File to be Saved as Project Information

The Smart Configurator saves the setting information such as the target MCU for the project, build tool, peripheral modules, and pin functions in a project file (*.scfg), and refers to this information.

When the Smart Configurator is activated from CS+, the project file from the Smart Configurator is saved in "project name.scfg", which is at the same level as the project file (*.mtpj) of CS+.



3.4 Window

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

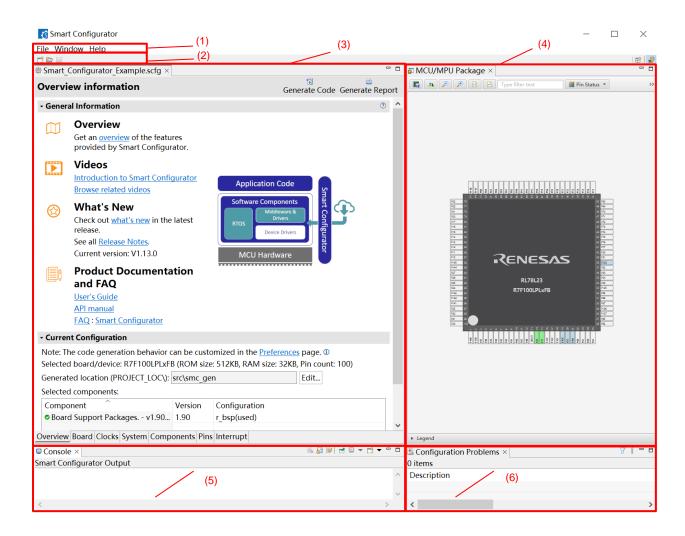


Figure 3-3 Main Window

- (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.			
		Do not use this menu item in general, as it leads to deletion of the project settings handed over from CS+.			
	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 Smart Configurator in Renesas website			
	Release Notes	Search for release notes of Smart Configurator in Renesas website			
	Tool News	Search for tool news of Smart Configurator in Renesas website			
	API Manual	Search for the RL78 API Reference in Renesas 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.

Toolbar button	Related menu item
	[File] ® [New]
6	[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_	Smart_Configurator_Example.scfg ×					
Overvi	ew information		🕤 🤷 🗎 🔓 Generate Code 🛛 Generate Rep	ort		
- Genera	al Information		0	^		
	Overview Get an <u>overview</u> of the featu provided by Smart Configur					
	Videos Introduction to Smart Config Browse related videos What's New Check out <u>what's new</u> in the release. See all <u>Release Notes</u> . Current version: V1.13.0 Product Documenta and FAQ User's Guide API manual FAQ : Smart Configurator	latest	Application Code Software Components Middleware & Drivers Device Drivers MCU Hardware			
- Currer	nt Configuration					
Selected Generat		3 (ROM siz	tomized in the <u>Preferences</u> page. ⁽¹⁾ te: 512KB, RAM size: 32KB, Pin count: 100) en Edit			
Compo	onent	Version	Configuration			
	Converter	1.7.0	Config_ADC(ADC: used)			
	d Support Packages v1.90		r_bsp(used)			
 Ports 		1.7.0	Config_PORT(PORT: used)	~		
Overview	Board Clocks System Comp	onents Pin	ns Interrupt			

Figure 3-4 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 Settings" 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.

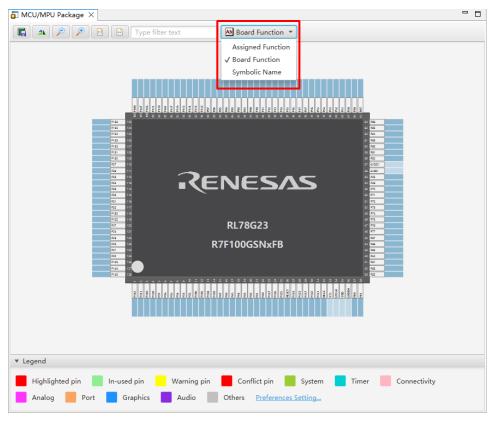


Figure 3-5 MCU/MPU Package View

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	. :	R	Ē	Ŧ	• '		
Smart Configurator Output								
M05000001: Pin 3 is assigned to PCLBUZ0 M05000001: Pin 117 is assigned to ANI0								^
								\checkmark
<							>	

Figure 3-6 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 3-7 Configuration Problems View



4. Setting of Peripheral Modules

User can select peripheral modules from the Smart Configurator view.

4.1 Board Settings

User can change the board and device on the [Board] page. For information on changing the device of the project, refer to the CS+ User's Guide.

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			Ъď
Board:	Custom User Board	~		
Device:	R7F101GLGxFB			
	Download more boards			
_				
Overview B	oard Clocks System Compone	ents Pins Inte	rrupt	

Figure 4-1 Selecting the Device

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 ^(Note*1)	After saving the current configuration contents to the configuration file, change to the selected device.
Continue ^(Note*1)	Changes to the selected device without saving the current configuration contents to the configuration file.
Cancel ^(Note*1)	It does not change the device.

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

Note *2: Device change is not reflected to the device (micro controller) of CS+ project.



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.sc	fg ×			- 8
Device se	election				🐻 Generate Cod	i Generate Report
Device se	election					2
Board:	Custom Use	r Board		~		
Device:	R7F101GLG	кFB				
	Download m	ore boar	<u>ds</u>			
Overview B	oard Clocks	System	Components	Pins Inte	errupt	

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

The board setting information is defined in the Board Description File (.bdf).

The .bdf file of Renesas made board (for e.g., Fast Prototyping Board) can be downloaded from website and imported.

In addition, by downloading the .bdf file provided by the alliance partner from website and importing it, it is possible to select alliance partner boards.

If user changes the board, the message will be displayed. For each button operation, refer to "Table 4-2, Board Change Confirmation Operation List".

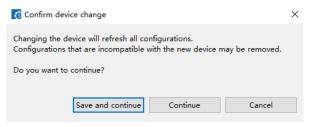


Figure 4-4 Confirm Board Change



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 target device of CS+ 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 [🖾 (Export board setting)] button on the [Board] 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	📄 Generate Report
Device se	lection						(1) ല്
Board:	Custom Use	r Board		~			
Device:	R7F101GLG	кFB					
	<u>Download</u> m	ore boards	<u>.</u>				
Overview B	oard Clocks	System Cor	mponents	Pins	nterrupt		

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 [i (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	onfigurator_Example.scfg $ imes$					- 6
Device s	election			당 Generate	Code	🕒 Generate Report
Device se	lection				(1)	24
Board:	Custom User Board		~			
Device:	R7F101GLGxFB					
	Download more boards					
Overview B	oard Clocks System Com	ponents P	ins Inte	rrupt		

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.

Clocks configura	tion				당 Generate Code	Generate Re
	igh-speed main mode 4.0(V)~5.5(V) 0 V ≤ EVDD0 ≤ 5.5 V	•				
	oscillator 32 (Millat) Normal v tring the high-speed on-chip oscillator from STOP mode and of transitions			(4)	fMAIN 32 fCLK 32000	(MHz) (MHz) (kHz)
Middle-speed on-cl	nip oscillator				fIMP -	(MHz)
Frequency:	4 • (MHz)	Divider				
X1 oscillator		- x1 ···			fMXP	
Operation mode:	X1 oscillation 👻				— ·	(MHz)
Frequency:	5 (MHz)					
Stable time:	2^18/fx ¥ 52428.8(µs)					
Low-speed on-chip osc	illator		•		fiL 32.768	(kHz)
Frequency:	32.768 (kHz)		(4)			
The flL runs while WE on-chip oscillator	T is operating or fSXP select Low-spe	d			fSXP	
on-chip oscillator				•	32.768	(kHz)
XT1 oscillator					fSXR	
Operation mode:	XT1 oscillation	-			32.768	(kHz)
	32.768 (kHz)					
Frequency:						
Frequency: XT1 oscillation mode:	Low power consumption 1	-				

Figure 4-7 [Clocks] Page

4.3 System Settings

User can set the on-chip debug setting on the [System] page. This setting is reflected in the CS + build option settings via the communication plugin.

For example, below figure shows the default CS+ link option settings:

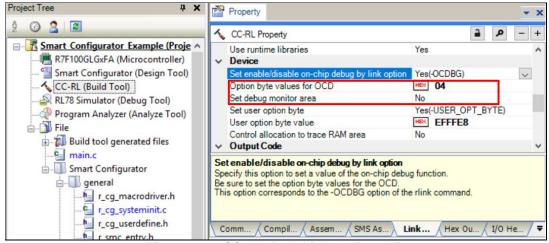


Figure 4-8 CS+ Default Link Options View

After user clicks on [System] page of Smart Configurator, make desired setting as in below figure for illustration:

System configuration		(4)	Generate Code Generate Report
			^
▼ On-chip debug setting			
On-chip debug operation setting O Unused (1)	• Use emulator		ort
Emulator setting C E2 (2)	• E2 Lite		
Pseudo-RRM/DMM function setting O Unused	Used		
Start/Stop function setting	OUsed		
Monitoring point function setting Unused	○ Used		
Trace function setting O Unused	Used		
Security ID setting	[
Security ID Security ID authentication failure setting O Do not erase flash memory data	0x000000000000000000000000000000000000		
Erase flash memory data Corriew Board Clocks System Component	ts Pins Interrupt		*

Figure 4-9 Smart Configurator [System] Page Setting

Please follow steps from (1) to (3) to make setting on [System] page, after that click on [Generate Code] button as in step (4), a dialog window will be prompted out as in below figure, to confirm with you for the linker option update in CS+ IDE:



Setting	Old value	New value	
Option byte values for OCD	04	84	
Range of debug monitor area	BFE00-BFFFF	BFD00-BFFFF	
Set debug monitor area	No	Yes(Specify address range)(-DEBUG_MONITOR=	

Figure 4-10 Confirm Linker Option Dialog

Please click [OK] button in the dialog, go back CS+ to check linker options updated as below:

Project Tree 🛛 📮 🗙	Property	• x
O 2 2 Smart Configurator Example (Project Smart Configurator Example (Project Smart Configurator (Design Tool) CC-RL (Build Tool)	CC-RL Property CC-RL Property Cutput File Library Device Set enable/disable on-chin debun. Yes(OCDBG)	- +
RL78 Simulator (Debug Tool) Program Analyzer (Analyze Tool) File	Option byte values for OCD INEX 84 Set debug monitor area Yes(Specify address range)(-DEBUG_MONITOR=) Rance of debug monitor area BFD000-BFFFFF Set user option byte Yes(-USER_OPT_BYTE) User option byte value INEX Control allocation to trace RAM ar No	
ia∎ Smart Configurator	✓ Output Code	~
	Common Options / Compile Options / Assemble Options / SMSAssemble Opti \ Link Options / Hex Output Options / 1/0 Header File Gen	/ Ŧ

Figure 4-11 CS+ Updated Link Options View

Note:

- 1. Depending on the MCU type selection or chip part numbers, these setting values varies. Please refer to the latest device User's Manual Hardware for the detail setting configuration.
- 2. The security ID setting is reflected in the security ID of Build Tool Common Options.

~	Device				
	Specify mirror area		МАА	=0(No option specified)	
	Security ID		HEX	000000000000000000000000000000000000000	
>	Build Method				1
	ild mode lects the build mode r	name to be use	əd durii	ng build.	
0	ommon Options 📈	Compile Opt	ions	AssembleOptions	F

Figure 4-12 CS+ Default Common Options View



4.4 **Component Settings**

CG drivers, Graphical Configurator and RL78 Software Integration System module can be combined as software components on the [Components] page. Added components are displayed in the tree view at the left of the page.

Image: Smart_Configurator_Example.scfg ×		
Software component configuration	Generate Code	📄 Generate Report
Components 🚵 🛃 📲 🕞 🕕 Configure		^
Tree view for components Tree view for components v > Startup v > Generic v > Tree view for components		
		~
Overview Board Clocks System Components Pins Interrupt		

Figure 4-13 [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. User can Switch two view 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.

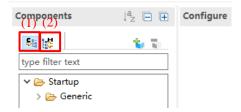


Figure 4-14 Switching to the Hardware View



4.4.2 Adding a Software Component

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

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

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

a-1. Click on the [to (Add component)] icon.

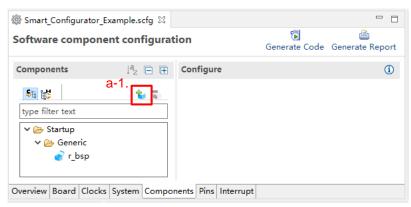


Figure 4-15 Adding a Component

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

	Kew Co	mponent					×
	Software Co	omponent	Selection				
	Select com	ponent fro	m those available in li	st			
i							
	Category	All					\sim
	Function	All					\sim
	Filter						
	Compon		^	Short Name	Туре	Versi	^
a-2.	⊞ A/D C	onverter		a-3.	Code Generator	1.4.0	
	🖶 Board	Support Pa	ackages v1.60	r_bsp	RL78 Software	1.60	
	H Clock	Output /Bu	zzer Output Contro		Code Generator	1.4.0	
	🖶 Comp	arator			Code Generator	1.3.1	
	🖶 D/A C	onverter			Code Generator	1.3.0	
	🖶 DALI 🤇	Communica	tion (Control device		Code Generator	1.0.0	
	🖶 DALI 🕻	Communica	tion (Control gear)		Code Generator	1.0.0	
	🖶 Data 1	Fransfer Co	ntroller		Code Generator	1.3.1	5
	<					>	Ť
	Show o	nly latest ve	ersion				
	Descriptio	n					
	The analo	og to digita	I (A/D) converter is fu	nction for conv	erting analog inputs	to	~
	digital sig						
							~
	Download	RL78 Softw	vare Integration Syste	m modules			
	Configure	general set	ttings				
			a-4	4			
			u				
	?		< Back	Next >	Finish	Cancel	

Figure 4-16 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 (e.g. Config_ADC).
- a-6. Select a hardware resource or use the default resource (e.g. ADC).
- a-7. Click on [Finish].

New Component	_	
Add new configuration for selected component		
A/D Converter a-5. Configuration name: Config_ADC Resource: a-6. ADC		~
a-7. ? < Back Next > Finish		Cancel

Figure 4-17 Adding a Component

To add a component on Hardware Tree directly, you can 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 (e.g. A/D Converter) to open the [New Component] dialog box.
- b-3. Select a component from the list (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.

	Software component cont	iguration		😼 Generate Code		👜 te B
	Compon 🚵 🛃 🖣 🕀 🕀	Configure	🚺 New Component)
)-1	St 🗱 🐮 🥫		Software Component Selection Select component from those available in list		1	8 8 8 8
	Timer RD Timer RG Timer RX Timer KB		Category All Function All Filter			
	 Real-Time Clock 32-bit Interval Timer Clock Output/Buzze Watchdog Timer A/D Converter 	b-3	Components Short Name	Type Code Gener	Vers 1.4.0	
	> D/A Converter > Comparator Programmable Gain > Serial Array Unit		Show only latest version Description The analog to digital (A/D) converter is func	tion for converting a	malag	
	> 🗀 Serial Interface IICA Digital Addressable Data transfer contrc		inputs to digital signals.		malog	
	 Event Link Controlle Interrupt Function Key Interrupt Voltage Detection 		Download RL78 Software Integration System Configure general settings	moaules		
	> > Others					
	< >		Sack Next >	Finish	Cano	el

Figure 4-18 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.

$\textbf{Components} \qquad \succeq \bigsqcup^a_{\mathbb{Z}} \boxdot \boxdot$	Components 🛛 🚵 📩 📮 🕀		
(2) 1 1 1 1 1 1			
type filter text	type filter text		
🗸 🗁 Startup	🗸 🔁 Startup		
✓ 🗁 Generic	🗸 🗁 Generic		
💣 r_bsp	💣 r_bsp		
✓	✓		
✓	✓		
Config_RTC (1)	Config_RTC (1)		
Config_TAU0_0	Config TAU0 0		
Config_TKB0_TKB1	Config_TKB0_TKB1		
Config_TRD0_TRD1	Config_TRD0_TRD1		
✓ → A/D converter	✓ ➢ A/D converter (1)		
💣 Config_ADC	Config_ADC		

Figure 4-19 Removing a Software Component or Multiple Components

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

To delete the source files previously generated for the removed components from the CS+ project tree, click [¹] (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 (e.g. A/D Converter).
- (2) Configure the driver in the [Configure] panel to the right of the Components tree. The following steps and figure 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.scf	fg 🔀			
Software component config	guration		🖲 🗎 Generate Code 🛛 Generate Rep	oort
Compon l ^a _Z 🕞 🕀	Configure		i	^
type filter text	Comparator operation setting Stop	○ Operation		
✓ ➢ Startup (2) ✓ ➢ Generic ✓ ➢ r_bsp	10 bits	⊖8 bits	🔿 12 bits	
✓	VREF(+) setting VDD	OAVREFP	O Internal reference voltage	
(1) Config_ADC	VREF(-) setting VSS	OAVREFM		
(2) t	 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 © Continuous select mode O One-shot select mode A/D channel selection (2) c.	 ○ Continuous scan mode ○ One-shot scan mode ANI0 	~	
	Conversion time setting Conversion time mode Conversion time (2) d.	Normal 1 2112/fCLK	~ ~ (66 μs)	ļ
	Conversion result upper/lower bound Generates an interrupt request (INT Generates an interrupt request (INT		Rn	
Overview Board Clocks System C	Upper bound (ADUL) value	255		~
orernen board clocks bystelli e	omponente rine interrupt			

Figure 4-20 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 you to change the resource for a Code Generator configuration (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.

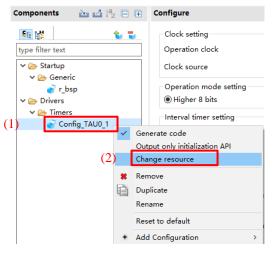


Figure 4-21 Changing the Resource

- (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					×
Resource Select	tion					
Select resource fr	rom those availa	ble in the	e list			
Operating mode:	8 bit count mc	de				~
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 4-22 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].

	C Resource Selection		— 🗆	Х
	Configuration setting selection			
	Configuration setting list	(7)		
	Confirm setting for resource change	● Use setting below ○	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	~
	<		2	>
		(8)		
	< Back	Next > Finish	Cancel	

Figure 4-23 Checking the Settings of the New Resource

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

🔅 *Smart_Configurator_Example.scfg 🛛	X		- 8
Software component configur	ration		🖲 🔒 Generate Code 🛛 Generate Report
Components 👌 🔁 🕀	Configure		i
Image: Startup ✓ Construint ✓ Construint	Clock setting Operation clock Clock source Operation mode setting Higher 8 bits 	CK02 fCLK/2	 (Clock frequency: 16000 kHz) O Higher and lower 8 bits
Config_TAU0_1	Interval timer setting Interval value (higher 8 bits) Interval value (lower 8 bits) Generates INTTM03 when counting Interrupt setting	10 μs 10 μs	۲ (Actual value: 10)
	 ☑ End of timer channel 3 count, get ☑ End of timer channel 3 count, get ☑ End of timer channel 3 count, get Priority 	Low	2
Overview Board Clocks System Com	< ponents Pins Interrupt		>

Figure 4-24 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).

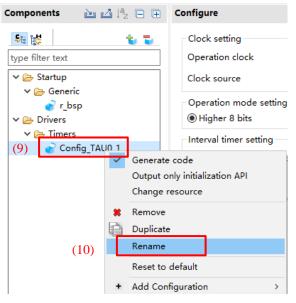


Figure 4-25 Renaming the Configuration

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	🔇 New Component – 🗆 🗙				×
	Software Component Selection				
Select cor	nponent from those availat	Die in list			
Category	All				\sim
Function	All				\sim
Filter					
Compon	ante ^	Short Name	Type	Versi	•
Beal-Ti		SHORE Name	Code Genera	1.7.0	
	Control Signal Receiver		Code Genera	1.7.0	
	ZE Mode Sequencer		Graphical Co		
	I) Communication		Code Genera		-
	Wave Output		Code Genera	1.7.0	
	Communication		Code Genera	1.9.0	
#Voltage	e Detector		Code Genera	1.6.0	
#Watch	dog Timer		Code Genera	1.7.0	
					\sim
Show o	nly latest version				
Descriptio	n				
The analo	og to digital (A/D) converte	r is function for cor	werting analog in	outs to	^
digital sig	gnals.				
Davalaad	RL78 Software Integration	Custom modules			Ť
	ELCL modules	i system modules			
	general settings				
configure	general settings				
?	< Back	Next >	Finish	Cancel	

Figure 4-26 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 wants to use.

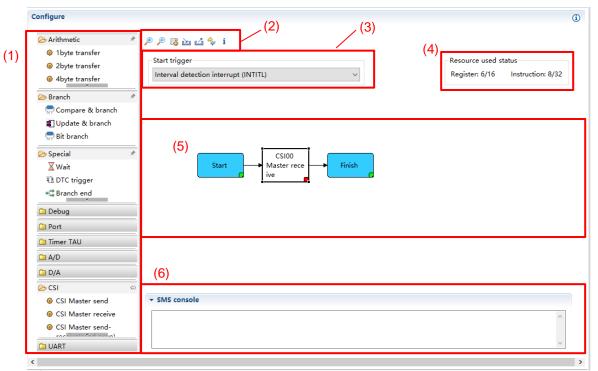


Figure 4-27 SNOOZE Mode Sequencer (SMS) GUI

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

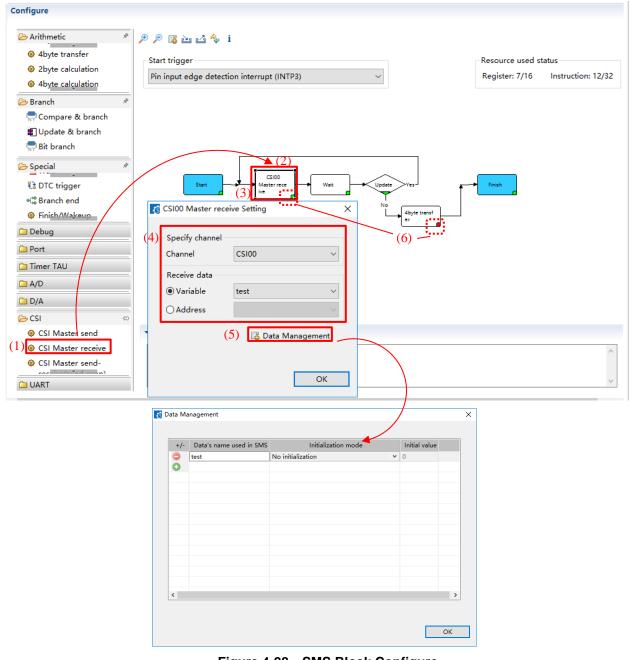
Ar	ea	Description	
(1) Block elements		View the available blocks for SMS.	
		A block is a part for forming a sequence (function), and includes A/D voltage acquisition, comparison & branching and 1-byte transfer.	
(2) Toolbar	Æ	Zoom in.	
	P	Zoom out.	
	•	Display the SMS data management dialog and manage the variables to be used.	
	2	Import the SMS sequence. You can use some sample sequences by clicking this icon.	
	4	Export the SMS sequence.	
	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	Update the SMS data file.	
	i	Displays the information of the SMS data file.	
(3) Start trigger selection		Select a startup trigger.	
(4) Resource status		It shows registers and the number of instructions used.	
(5) Canvas area		Place the SMS block and create the sequence.	
(6) SMS cons	ole	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) User can configure the block by double click to pop the "CSI Master receive setting" property setting dialog.
- (4) User can specify the setting in the "CSI Master receive setting" property dialog.
- (5) Open "Data Management" setting, use can edit the receive data.
- (6) When you correctly configure the color of bottom right corner will change from red to green.
- (7) User can add some blocks, drag and drop to adjust the sequence.





#### 4.4.7 Update SMS Data Files

Follow the procedure below to update SMS data file (Block, Sequence) to the latest version. User can 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	(1)	(i
Arithmetic		
4byte transfer	Start trigger Update SMS data files	source used status
Ø 2byte calculation		gister: 2/16 Instruction: 1/32
🗁 Branch 🤞		,,
R Compare & branch		
🗐 Update & branch		
🗁 Special 🕺		
t DTC trigger		
•C Branch end	Start Finish	
Ca Debug		
Progress In	oformation	
(2)	Operation in progress	
	1	
		Cancel
		Cancer
<i>(</i> )	Information	×
	mormation	<u>^</u>
(3)	No new version.	
	OK	

Figure 4-29 SMS Data File Download

(

#### 4.4.8 ELCL Fixed Function Modules Download

The Software Component type for ELCL (Logic and 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 C	omponent									
Software	<b>Component Selection</b>				<del></del>					
Select cor	mponent from those availal	ole in list		<b>(</b> \$						×
Category Function					78 ELCL Modules lect the RL78 ELCL		ownload	(2)	)	Ľ
# Clock ( # Compa # D/A Co # Data Ti # Delay ( # Divider	onverter Support Packages v1.90 Output /Buzzer Output C arator onverter ransfer Controller Counter r Function ND only latest version	Short Name	RL78 So Code G	KKKKKKKKKKK	Title Slave Select Pin Fi ELCL AND Selector Edge Detection Ti DFlipFlop	hinning Func on ELCL Mod Ider Function Iter Monitorin Intion Function	1.1.0 1.1.0 2.0.0 1.1.0 1.1.0 2.0.3 2.0.0			elect All
digital si			verting ar	n		(3)	Downlo	ad	Са	ncel
Download	d RL78 Software Integration <u>d ELCL modules</u> e general settings	<u>i System modules</u>								
?	< Back	Next >	Finish		Cancel					

Figure 4-30 ELCL Fixed Function Modules Download

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



Catagony	A11					
	Category All					
Function	unction All					
Filter						
Compon	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	1.7.0	_	
IN ELCL A	ND		Graphical Co	1.1.0		
🕫 ELCL ch	nattering prevention		Graphical Co	2.0.0		
IN ELCL D	flip flop		Graphical Co	1.1.0		
	lge detection thinning fu		Graphical Co			
IN ELCL EX	(OR		Graphical Co	1.1.0	~	
	nly latest version					
Descriptio	n					
	og to digital (A/D) converte	r is function for	converting analog	inputs to	· ^	
digital sig	gnals.				~	
Download	RL78 Software Integration	System module	ic .			
	ELCL modules	Systemmodule	2			
Download						

Figure 4-31 Add ELCL Fixed Function Modules

#### 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.

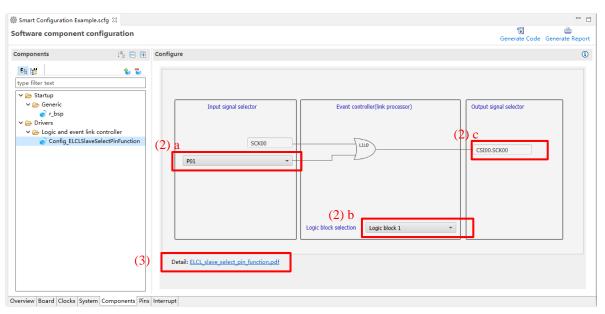


Figure 4-32 Configure a Fixed Function ELCL Component



#### 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 Component – 🗆 🗙						
Software Component Selection						
Select con	nponent from those ava	ilable in list			40	
Category All v						
Function All						
Filter						
Compone	ents	Туре	Ver	^		
🖶 Data Tr	ransfer Controller		Code Generator	1.5.0		
# Delay O	Counter		Code Generator	1.6.0		
# Divider	Function		Code Generator	1.6.0		
🕅 ELCL A	ND		Graphical Configurator	1.1.0		
🕅 ELCL cl	hattering prevention		Graphical Configurator	2.0.0		
BLCL D flip flop			Graphical Configurator	1.1.0		
🛱 ELCL edge detection thinnin			Graphical Configurator	2.0.0		
Caphical Configurator 1.1.0						
ि⇔ELCL FI	exible Circuit		Graphical Configurator	1.0.0		
Ill ELCL manchester decoder Graphical Configurator 2.0.0 V						
Descriptio The logic other spe	and event link controlle	ns through inter	als output by peripheral fu nal logic cell blocks, allowir hout CPU intervention.			
Download	RL78 Software Integrat	ion System modu	les			
Download	ELCL modules					
Configure	general settings					
?		< <u>B</u> ack	<u>N</u> ext > <u>F</u> inish	Can	cel	

Figure 4-33 Add "ELCL Flexible Circuit" Component

ELCL flexible circuit component provides 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.

onfigure				Q
Logic Contents (1)	,∌ ,⊜ (€)			
- input 🗤	Input	Logic (L)	Output	
- Input				
•© Logic Cell ⇔				
Pass-through				
D AND circuit				
② OR circuit				
D EX-OR circuit	Input pin (P10/EI10)			
D- Selector		LILO	SAU0 Ch1 fSCK/TAU0 Ch0 In	
⊐\$ Flip-Flop				
🗲 Output 🛛 🗠				
● Output	INTTMOO		P50/P01/INTELCL/P10	
Properties (2) P50/P01/INTELCL/P10				
Output signal level setting				
Negative logic output				
Interrupt setting				
Use ELCL Interrupt(INTELCL)				
Priority Level 3 (low) 🗸				
	<b>Figure 4.24</b>	ELCL Elevible Circuit Component CIII		

Figure 4-34 ELCL Flexible Circuit Component GUI



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

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

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, the user can freely do each step to create or edit ELCL circuit.

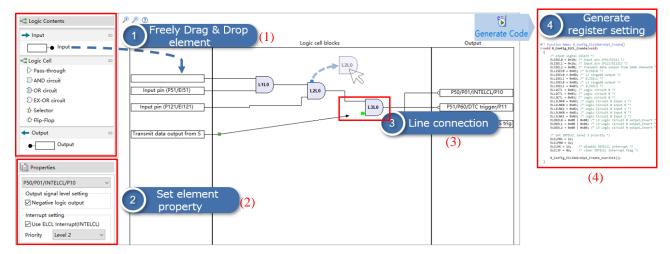
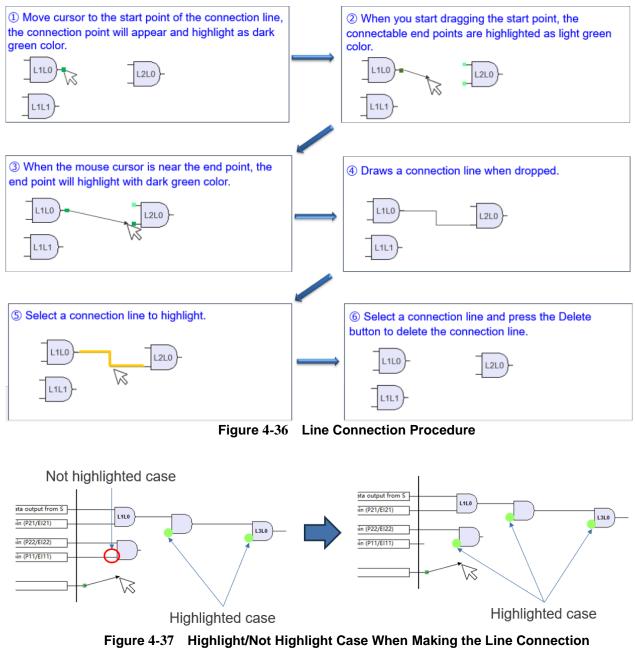


Figure 4-35 ELCL Flexible Circuit Creation Procedure



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

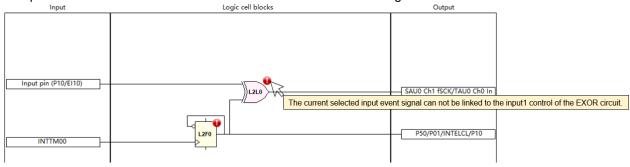
(1) Connect the line through drag & drop the start point to end point.



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



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





#### Table 4-5. 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 the user wants to configure this setting, the user can 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, the user can 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 exists unallocated resources 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, the user 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 [^t (Add component)] as in Figure 4-15 to open new component dialog.
- (2) Click the [Download RL78 Software Integration System module] link in the [New Component] dialog box to start the download.

Category Function	All				~
Filter					
Compon	ents	Short Name	Туре	Versi	^
<b>#A/D</b> Co	onverter		Code Genera	1.7.0	
Board	Support Packages v1.90	r_bsp	RL78 Softwar	1.90	
#Clock (	Output /Buzzer Output C		Code Genera	1.7.0	
#Compa	arator		Code Genera	1.5.0	
<b>₽D/A</b> Co	onverter		Code Genera	1.5.0	
#Data T	ransfer Controller		Code Genera	1.6.0	
# Delay Counter			Code Genera	1.7.0	
# Divide	r Function		Code Genera	1.7.0	~
Descriptio The analo digital si	og to digital (A/D) converte gnals.		-	inputs to	
Download	3 RL78 Software Integration 3 ELCL modules	System module	<u>25</u>		

Figure 4-39 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.

Renesas Login	MyRene
ter the details used to register your	r My Renesas account.
Email address:	
Password:	
-	Forgot password
	Proxy Setting
Privacy Policy & Information Co.	
O I do not consent to usage data <u>Privacy Policy</u>   Information Collect	
	ted

Figure 4-40 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	for download		(3	8
	Title	Document No.	Rev.	Issue date	^
	RL78 Family FS2012 Sensor Control Module So	R01AN6196EJ0112	Rev.1.12	2023-04-26	
$\checkmark$	RL78 Family HS400X Sensor Control Module S	R01AN6446EJ0102	Rev.1.02	2023-04-26	
$\checkmark$	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	~
٢.	-			)	Þ
Мо	dule Folder Path:				
[	C:\Users\ \.eclipse\com.renesas.smc.rcp.rl7	8.product download	RL78 Modul	es\GenericMod	ules

Figure 4-41 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 [1] (Add component)] icon as Figure 4-15 Adding a Component.
- (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].

Select com	omponent Selection	list	
Category	All		
Function	All		
Filter			
Compon	^	Short Name	Туре
	onverter (2)	Short Name	Code Generator (3)
	Support Packages v1.60	r bsp	RL78 Software Integration System
	titive Sensing Unit driver.	r ctsu	RL78 Software Integration System
	Output /Buzzer Output Contro.	-	Code Generator
Comp			Code Generator
+ D/A C			Code Generator
🖶 Data 1	Fransfer Controller		Code Generator
🖶 Delay	Counter		Code Generator
🖶 Divide	er Function		Code Generator
🕼 ELCL /	AND		Graphical Configurator
<			>
☑ Show o Descriptio	nly latest version n		
			Sensing Unit. The CTSU2L module layer, but can also be accessed
from the	RL78 Software Integration Syst	em modules	
from the Download	RL78 Software Integration Syst	em modules	
from the Download Download		<u>em modules</u>	
from the Download Download	ELCL modules	<u>æm modules</u>	(4)
from the Download Download	ELCL modules	<u>æm modules</u>	(4)
from the Download Download	ELCL modules general settings	ack Next	

Figure 4-42 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 🚵 🛃 🖧 🕒 🕀			
15	Property	Value	
filter text	✓ ⊕ Configurations		
Startup	# Parameter check	Use system default	
Generic	# Data transfer of INTCTSUWR and INTCTSURD	Interrupt handler	
💣 r bsp	# DTC setting	Setting in r_ctsu	
Drivers	# Auto-judgment function in Snooze mode using SMS	Disable	
Logic and event link control	# Data storage address setting for CTSURD	0xFF500	
Config AND	# Data storage address setting for CTSUWR	0xFF800	
Middleware	# Interrupt level for INTCTSUWR	Level 2	
And 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	
	✓ I Resources		

Figure 4-43 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.

ftware component c	onfiguration		Generate Code	Generate Repo
mpo J ^a _Z E E	Configure			0
ste iter text	Property v		Value	^
✓ ➢ Startup	# Start up select		Enable (use BSP startup)	
<ul> <li>Generic</li> </ul>	# Control of invalid memory ac	cess detection	Disable	
P denenc	# RAM guard space(GRAM0-1)		Disabled	
e i_osp (	Change version	rt function(GPORT)	Disabled	
	Remove	unction(GINT)	Disabled	
** *	Reset to default	ock control function, voltage detector, and RAM parity erro	Disabled	
1	Reset to default	)	Disables	
[	Download and import sample projects	ons by Code Generator/Smart Configurator	Enable	
	# API functions disable		Enable	
	# Parameter check enable		Enable	
		speed on-chip oscillator at the times of release from STOP mc	High-speed	~
	<			>
				-
				~

Figure 4-44 Version Change of BSP Configuration

(2) Select [Change Version ...] from the context menu.



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

💰 Change Versior	n				—		×
Version Selection							
Select available ve	ersion						
Component name:	k ben						
Current version:	1.00						
Available versions	1.01						~
•							
	< Back	Г	Next >	Finish		Cance	ł

Figure 4-45 Select Version of BSP Component

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

Change Version		×
Setting Overview		
The following settings will be added or removed		
Setting	Status	
There are no differences		
		_
<		>
< Back Next > Finish	Cance	ł

Figure 4-46 Confirm Setting Change Item

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





(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 [¹] (Export Configuration)] button on the [Components] tabbed page.

Components	ère] ^{ia} z ⊟ ⊞
- <b>5</b> 13	😜 🗟
type filter text	
🗸 🗁 Startup	
🗸 🗁 Generic	
💣 r_bsp	
🗸 🗁 Drivers	
🗸 🗁 Timers	
💣 Config_1	rau0_1

Figure 4-48 Export Configuration (xml format)

#### 4.4.16 Import Component Configuration

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

Components	Ľı La La E E
	😜 🗟
type filter text	
🗸 🗁 Startup	
🗸 🗁 Generic	
💣 r_bsp	
🗸 🗁 Drivers	
🗸 🗁 Timers	
💣 Config_	TAU0_1

Figure 4-49 Import Configuration (xml format)



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

The user can change the general setting of the component such as location and dependency. If user wants to change it, click the [Configure general settings...] link on the [Software Component Selection] page displayed in the [New Component] dialog (Figure 4-16 Adding a Code Generator Component), and display the [Preferences] dialog.

S Preferences		$\Box$ $\times$
type filter text	Component	← ▼ ⇒ <
Help Logging Module Downlo My Renesas	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	
MCU/MPU Pa	API code style: Value with macro description	$\sim$
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 Location settings have moved to the <u>Module Download</u> page	
< >	Restore Defaults	Apply
	Apply and Close	Cancel

Figure 4-50 Configure General Setting of Component

#### Notes:

1. The 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 it exceeds 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 4-51 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.

Figure 4-52	[Code generation behavior] Change
	Re-generate all component files
	Update configuration files
Code generation behavior:	Undate configuration files
FIT(RX) / SIS(RL78 / RISC-V	MCU) component settings



3. If user wants to only generate initialization API function, user can 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 you 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	
· · · · · · · · · · · · · · · · · · ·	Output only initialization API function	-

Figure 4-53 [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>Drivers</li> <li>Interrupt</li> </ul>			
Config_INTC	~	Generate code	
		Output only initialization API	
		Change resource	
	8	Remove Duplicate Rename	
		Reset to default	
	+	Add Configuration	>

#### Figure 4-54 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 the user changes back to default option setting: [Value with macro description], then all API with macro description will be generated again.

Code Generator cor	nponent settings	
API function output	Output all API functions according to the setting	~
API code style:	Value with macro description	1
FIT(RX) / SIS(RL78 /	Value with macro description Value without macro description (raw HEX)	

Figure 4-55 [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 checks the revision history of the module and its dependencies and you do not need to change the module you are using, you can ignore this warning. To clear this warning, select [Do not check for dependent component] in the [Checking dependency] list box in component preferences, then click [OK].

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

Figure 4-56 [Checking dependency] Change



## 4.5 **Pin Settings**

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

Smart_Configurator_Example	.scfg ×							6		-
in configuration								Generate	Code Genera	ate Rep
Hardware Resource	⊞ ⊟ ↓ <mark>2</mark> 🕹	Pin Functio	on						ર 🔳 🖬	2
Type filter text		type filter	text (* = any s	string, ? = any	character)				All	
📥 All	^	Enabled	Function	PIOR	Assignment	Pin Number	Direction	Remarks		Con
\$∰ I/O Ports			ANI0		P20/ANI0/AVREFP/EI20	/ 117	1	There is no softwar	e 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			
TAU02			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	None			
TAU14	Display swite	ching	ANI18		Not assigned	Not assigned				
TAU15     TAU16     TAU16		<	ANII10		/ Not acclanad	/ Not acclanad	Mono			>

Figure 4-57 [Pins] Page ([Pin Function])

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

Smart_Config	gurator_Example.scfg $ imes$								
n configu	iration							Generate Code Gener	iate Repor
n Number									<b>1</b>   21 2
type filter tex	kt (* = any string, ? = any cha	aracter)						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	None			J9-5[IO~4]		
13	P104/TI15/TO15		•	None			J9-6[IO~5]		
14	P103/TI14/TO14		· · · · ·	None			J9-7[IO~6]		
15	P47/INTP2	P47	3	None			Pmod2-7[IO1]		
16	P46/INTP1/TI05/TO05		•	None			J9-8[IO~7]		
17	P45/SO01		<b>.</b>	None			J4-7		
18	P44/SI01/SDA01	SDA01		None			Grove3-2[SDA]		
19	P43/SCK01/SCL01	SCL01		None			Grove3-1[SCL]		
20	P42/TI04/TO04		Not assigned	None			J4-1[IO~8]		~

Figure 4-58 [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 the user changes one of pin assignment, all pin function assignments which related to PIOR1 will be re-assigned automatically.
- (3) The pin error messages may display in [Remark] column and [Configuration Problems view].
- (4) The user need to re-configure pin assignment.

	tion						-3 E	
pior1						(3)	<i>₿_</i> All	
Enabled	Function	PIOR	Assignment	Pin Number	Direction	Remarks		Comment
	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			
$\checkmark$	🔕 SCK00	PIOR1	P55/PCLBUZ1/SCK00	/ 72	10	Multiple pin functions on the	same 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			
$\checkmark$	S100	PIOR1	P16/EO16/CCD00/TI01/TO01/INTP5/SI00/RxD0	/ 76	I.			
	SI20	PIOR1	Not assigned	Not assigned	None			
	SO00	PIOR1	Not assigned	Not assigned	None			
	🔕 SO20	PIOR1	Not assigned	Not assigned	None	Component requires a pin		
<	T 00	PIOD4	<b>* ** *</b>	- KI - 1	N1			
Config errors, 0			^				Туре	
L Config errors, 0 Descriptic ✓ Ø Pin	warnings, 0 o on (7 items)	thers						
2 Config errors, 0 Descriptic ✓ ② Pin ⊗ E	warnings, 0 o on (7 items) 604010002: SC	thers K20 used by Co	nfig_CSI20 is not allocated to any pin.				Pin	
Config errors. 0 Descriptic V O Pin O E O E	warnings, 0 o on (7 items) 604010002: SC 604010002: SO	thers K20 used by Co 20 used by Con	nfig_CSI20 is not allocated to any pin. fig_CSI20 is not allocated to any pin.	in Dia Allocator, a	in urad by S	5/00 in Confin (5100	Pin	
Config errors, 0 Descriptic V O Pin O E O E O E	warnings, 0 o on (7 items) 204010002: SC 204010002: SO 204010003: Pin	thers K20 used by Co 20 used by Con used by PCLBU	nfig_CSI20 is not allocated to any pin. fig_CSI20 is not allocated to any pin. Z1 in Config_PCLBUZ1 conflicts with pin used by SCK00				Pin Pin Pin	
Config errors. 0 Descriptic © Pin © E © E © E © E © E	warnings, 0 o on (7 items) 04010002: SC 04010002: SO 04010003: Pin 04010003: Pin	thers K20 used by Co 20 used by COn used by PCLBU used by SCK00	nfig_CSI20 is not allocated to any pin. fig_CSI20 is not allocated to any pin. Z1 in Config_PCLBUZ1 conflicts with pin used by SCK00 in Config_CSI00 conflicts with pin used by PCLBUZ1 in (	Config_PCLBUZ1, p			Pin Pin Pin Pin	
Config errors. 0 Descriptic © Pin © E © E © E © E © E © E	(7 items) 04010002: SC 04010002: SC 04010002: SO 04010003: Pin 04010003: Pin 05000010: Pin	thers K20 used by Co 20 used by Con used by PCLBU used by SCK00 72 cannot be u	nfig_CSI20 is not allocated to any pin. fig_CSI20 is not allocated to any pin. Z1 in Config_PCLBUZ1 conflicts with pin used by SCK00	Config_PCLBUZ1, p			Pin Pin Pin	

Figure 4-59 PIOR Filter Function

The PIOR setting can be reflected into r_bsp file in: \<ProjectDir>\src\smc_gen\r_bsp\r_config\r_bsp_config.h file. If user wants to change the PIOR setting code value, change the assignment of related pin and generate code again.

*Smart_Configurator_Example.scfg	8						
n configuration						Generate (	Code Generate Re
oftware Com 🕀 🖃 🖧 👪	Pin Functio	on				,	9 🗉 🖬 🗠
Type filter text	type filter	text (* = an	y string, ?	= any character)			All
🛩 📥 r_bsp	Enabled	Function	PIOR	Assignment	Pin Number	Direction	Remarks
<ul> <li>r_bsp</li> <li>dock Output /Buzzer Ou</li> </ul>		PCLBUZO	PIOR3	/ P140/PCLBUZ0/INTP6	/ 3	0	
Config PCLBUZ0							
<u> </u>							
< >>	<						
Function Pin Number							
erview Board Clocks System Co	mponents Pi	ns Interrup	t				
r_bsp_config. h🛛							
07 128/100-	pin						
08 0 : PCLBUZ	0 - P14	0					
09 1 : PCLBUZ	0 - P31						
10 */							
511 #define BSP	_CFG_PI	OR3 (0	) /*	Generated value.	Do not ed	it this	manually
512							

Figure 4-60 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 [🚠 (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.

									_
Smart_Configurator_Example.scfg ×									-
n configuration							Gar	nerate Code Generate	-
(1)	_						Ger	ierate code Generate	кер
oftware Components 🛛 🕀 📮	Pin Functi	on					(5	) 🧐 🖪 📳	<u>ک</u> ۱
Type filter text		text (* = ar	iy string, ? =	any character)				All	
A a Losb	3) Enabled	Function	PIOR	Assignment	Pin Number	Direction	Remarks	Comments	
Linterrupt Controller		INTPO		Not assigned	Not assigned	None			
Config_INTC		INTP1	PIOR5	Not assigned	Not assigned	None			
		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 software initialising		
		INTP6	PIOR5	Not assigned	Not assigned	None			
		INTP7	PIOR5	Not assigned	Not assigned	None			
		INTP8	PIOR5	Not assigned	Not assigned	None			
		INTP9	PIOR5	Not assigned	Not assigned				
		INTP10	PIOR1	Not assigned	Not assigned				
		INTP11	PIOR1	Not assigned	Not assigned	None			

Figure 4-61 Pin Settings – Assigning Pins on the [Pin Function] List

The Smart Configurator allows the 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...].

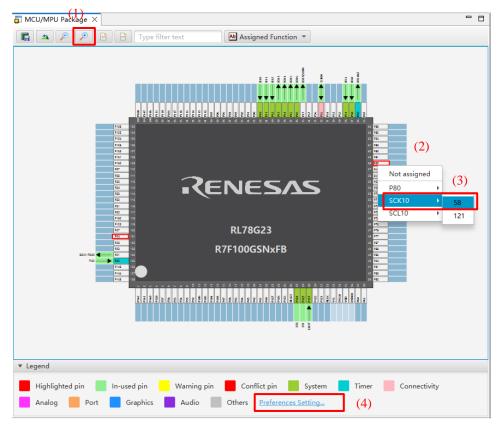


Figure 4-62 Assigning Pins Using the MCU/MPU Package View

#### 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.

-	ration								Generate	Code Ger	nerate Rep
oftware Cor	mpo 🕀 🖻 📲 Pin	Function								୫ 🖬	<b>1</b>   2
Type filter te	ext ty	pe filter text (* =	any string, ? = a	ny character)						All	
> 📥 r_bsp	E	nabled Function	PIOR	Assignment		Pin Number	Direction R	emarks	Cor	nments	
🗸 🚣 Interr	upt Controller	INTP1		Not assigne	d	Not assigned	None				
💣 Co	nfig_INTC	INTP2		Not assigne		Not assigned					
		INTP3		Not assigne		Not assigned					
				Not assigne		Not assigned					
		INTP5	PIOR4	P16/EO16/C	CD00/TI01/				<u>4</u>		
			PIOR5 /	Not assigne	d	Jump to Pin Nu	mber				
		INTP7		Not assigne		Merge comme	nt to Pin Numbe	r tab			
		INTP8	PIOR5	Not assigne	d	Clear comment	s				
		INTP9	PIOR5	Not assigne	d	Assign selected	pins				
		INTP10	PIOR1 /	Not assigne	d	Unassign select	ed pins				
		INTP11	PIOR1 /	Not assigne	d	Not assigned					
	d Clocks System Compone	nts Pins Interru	ot	4	₽						
rview Boar	d Clocks System Compone	nts Pins Interru	ot	4	┍				Generate	Code Ger	۵
rview Boar	d Clocks System Compone	nts Pins Interru	ot	4	₽						ierate Rep
nview Boar mart_Config a configu a Number	d Clocks System Compone		ot	4	ŀ						ierate Rep
nview Boar mart_Config a configu a Number	d Clocks System Compone gurator_Example.scfg ×		Function	Direction	Remarks	Symbolic Name	Comments				ierate Rep
rview Boar mart_Config a <b>configu</b> a <b>Number</b> upe filter tex	d Clocks System Compone gurator_Example.scfg × ration	racter)	Function	Direction	Remarks	Symbolic Name	Comments J1-19				ierate Rep
mart_Config a configu Number pe filter tex	d Clocks System Compone gurator_Example.scfg × ration t (* = any string, ? = any cha Pin Name	racter)			Remarks	Symbolic Name					ierate Rep
rview Boar mart_Config o configu o Number pe filter tex in Numb 74	d Clocks System Compone gurator_Example.scfg × ration t (* = any string, ? = any cha Pin Name P57/INTP3	acter) Board Functi	Function Not assigned	None	Remarks There is	Symbolic Name	J1-19				ierate Rep
rview Boar mart_Configu a configu a Number pe filter tex in Numb 74 75	d Clocks System Compone gurator_Example.scfg × ration tt (* = any string, ? = any cha Pin Name P57/INTP3 P17/EO17/CCD01/TI02/TO	acter) Board Functi	Function Not assigned Not assigned	None None		Symbolic Name	J1-19 J1-20	]			ierate Rep
rview Boar mart_Configu n configu Number pe filter tex in Numb 74 75 76	d Clocks System Compone gurator_Example.scfg × ration tt (* = any string, ? = any cha Pin Name P57/INTP3 P17/EO17/CCD01/TI02/TO. P16/EO16/CCD00/TI01/TO	racter) Board Functi INTP5	Function Not assigned Not assigned INTP5	None None I		Symbolic Name	J1-19 J1-20 Pmod2-1[INT]	]			ierate Rep
rview Boar mart_Configu number pe filter tex in Numb 74 75 76 77	d Clocks System Compone gurator_Example.scfg × ration tt (* = any string, ? = any cha Pin Name P57/INTP3 P17/EO17/CCD01/T102/TO P16/EO16/CCD00/T101/TO P15/EO15/SCK20/SCL20/T1	racter) Board Functi INTP5 RxD2	Function Not assigned Not assigned INTP5 Not assigned	None None I None		Symbolic Name	J1-19 J1-20 Pmod2-1[INT] Grove2-1[SCL	] ] ]]/Grov			ierate Rep
rview Boar mart_Configu a configu a Number pe filter tex rin Numb 74 75 76 77 78	d Clocks System Compone gurator_Example.scfg × ration tt (* = any string, ? = any cha Pin Name P57/INTP3 P17/E017/CCD01/T102/TO. P16/E016/CCD00/T101/TO P15/E015/SCK20/SCL20/TI. P14/VCOUT1/E014/RxD2/	racter) Board Functi INTP5 RxD2	Function Not assigned Not assigned INTP5 Not assigned Not assigned	None I None None		Symbolic Name	J1-19 J1-20 Pmod2-1[INT Grove2-1[SCL J8-3[IO19/RX	] ] ]]/Grov			ierate Rep
nview Boar mart_Configu n configu n Number pe filter tex rin Numb 74 75 76 77 78 79	d Clocks System Compone gurator_Example.scfg × ration t (* = any string, ? = any cha Pin Name P57/INTP3 P12/EO17/CCD01/TI02/TO. P16/EO15/SCK20/SCL20/TI. P16/EO15/SCK20/SCL20/TI. P14/VCOUT1/EO14/RxD2/ P13/IVREF1/EO13/TxD2/S	racter) Board Functi INTP5 RxD2 TxD2	Function Not assigned INTP5 Not assigned Not assigned Not assigned	None None I None None None		Symbolic Name	J1-19 I1-20 Pmod2-1[INT] Grove2-1[SCL J8-3[IO19/RX J8-4[IO18/TX1	] ] ]]/Grov			ierate Rep
nview Boar mart_Configu a configu number pe filter tex in Number 74 75 76 77 78 79 80	d Clocks System Compone gurator_Example.scfg × ration t (* = any string, ? = any cha Pin Name P57/INTP3 P12/E012/CCD01/TI02/TO. P16/E015/SCK20/SCL20/TI. P15/E015/SCK20/SCL20/TI. P13/IVREF1/E013/TxD2/s P12/E112/E012/S000/TxD	racter) Board Functi INTP5 RxD2 TxD2	Function Not assigned INTP5 Not assigned Not assigned Not assigned Not assigned	None None I None None None None		Symbolic Name	J1-19 J1-20 Pmod2-1[INT] Grove2-1[SCL J8-3[IO19/RX' J8-4[IO18/TX1 J1-21	] ] ]]/Grov			
nview Boar mart_Configu a configu number pe filter tex in Number 74 75 76 77 78 79 80 81	d Clocks System Compone gurator_Example.scfg × ration tt (* = any string, ? = any cha Pin Name P57/INTP3 P12/EO12/CCD01/TI02/TO. P16/EO16/CCD00/TI01/TO P15/EO15/SCK20/SCL20/TI P13/IVREF1/EO13/TxD2/S P12/EI12/EO12/SO00/TxD P11/EI11/EO11/SI00/RxD0	racter) Board Functi INTP5 RxD2 TxD2	Function Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned	None None I None None None None None		Symbolic Name	J1-19 J1-20 Pmod2-1[INT] Grove2-1[SCL J8-3[IO19/RX' J8-4[IO18/TX1 J1-21 J1-22	] ] ]]/Grov			ierate Rep
nview Boar mart_Configu n Number pe filter tex in Numb 74 75 76 77 78 79 80 81 82	d Clocks System Compone gurator_Example.scfg × ration tt (* = any string, ? = any cha Pin Name P57/INTP3 P17/EQ12/CCD01/TI02/TO. P16/EQ16/CCD00/TI01/TO P15/EQ15/SCK20/SCL20/TI. P15/EQ15/SCK20/SCL20/TI. P13/IVREF1/EQ13/TxD2/ P13/IVREF1/EQ13/TxD2/ P13/IVREF1/EQ13/TxD2/ P11/EI11/EQ11/SI00/RxD0 P10/EI10/EQ10/SCK00/SCL.	racter) Board Functi INTP5 RxD2 TxD2	Function Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned	None None None None None None None None		Symbolic Name	J1-19 J1-20 Pmod2-1[INT] Grove2-1[SCL J8-3[IO19/RX' J8-4[IO18/TX1 J1-21 J1-22 J1-23	] ] ]]/Grov			ierate Rep
nview Boar mart_Configu n Configu n Number pe filter tex in Numb 74 75 76 77 78 80 81 82 83	d Clocks System Compone gurator_Example.scfg × ration t (* = any string, ? = any cha Pin Name P57/INTP3 P17/EO17/CCD01/TI02/TO. P16/EO16/CCD00/TI01/TO P15/EO15/SCK20/SCL20/TI P14/VCOUT1/EO14/RxD2/ P13/IVREF1/EO13/TxD2/S P12/EI12/EO12/SO00/TxD P12/EI12/EO12/SO00/TxD P10/EI10/EO10/SCK00/SCL P90	racter) Board Functi INTP5 RxD2 TxD2	Function Not assigned INTP5 Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned	None I None None None None None None None None		Symbolic Name	J1-19 J1-20 Pmod2-1[INT] Grove2-1[SCL J8-3[IO19/RX' J8-4[IO18/TX1 J1-21 J1-22 J1-23 J1-24	] ] ]]/Grov			ierate Rep

Figure 4-63 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							Generate Code	e Generate Re	ерс
oftware Com 🕀 🖻 🛃	Pin Function	on					3		Ľ
Type filter text	type filter	text (* = ar	ny string, ?	= any character)			A	11	`
> 🚣 r_bsp	Enabled	Function	PIOR	Assignment	Pin Number	Direction	Remarks	Comments	^
🗸 🚣 Interrupt Controller		INTP1	PIOR5	Not assigned	Not assigned	None			÷
💣 Config_INTC		INTP2		Not assigned	Not assigned	None			
		INTP3	PIOR5	Not assigned	Not assigned	None			
		INTP4	PIOR5	Not assigned	Not assigned				
		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		>	1
	<b>`</b>								

Figure 4-64 Exporting Pin Settings to an XML File

The Smart Configurator can also export the pin settings to a CSV file. Click on the [I] (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 R	ep
oftware Com 🕀 🖃 🖓 🖁	Pin Functi	on					ર	II 🖬 🕹	I
Type filter text	type filter	text (* = ar	ny string, ?	= any character)			А	.11	,
> 🚣 r_bsp	Enabled	Function	PIOR	Assignment	Pin Number	Direction	Remarks	Comments	
V 🚣 Interrupt Controller		INTP1	PIOR5	Not assigned	Not assigned	None			ł
💣 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	I	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			1
	<							>	

Figure 4-65 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 according to the Renesas board that you selected to use. User can check the board that selected to use in [Board] tabbed page.

The following describes the procedure for collective setting of pins.

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

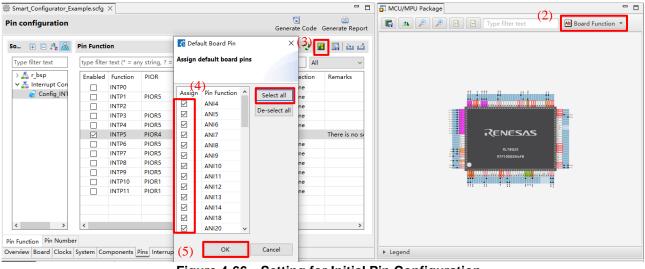


Figure 4-66 Setting for Initial Pin Configuration

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.

ftware Co 🕀 🖃 📲 👪	Pin Function	on			્ર		è
ype filter text	type filter	text (* = ar	ny string, ?	= any character)	1	All	
> 🚣 r_bsp	Enabled	Function	PIOR	Assignment	Din Number	All Function	
🗸 💑 Interrupt Controller		INTP0		Not assigned	Not assigned		
Config_INTC		INTP1	PIOR5	Not assigned	Not assigned		
		INTP2		Not assigned	Not assigned	Pin Number	
		INTP3	PIOR5	Not assigned	Not assigned	Direction	
		INTP4	PIOR5	Not assigned	/ Not assigned	Remarks Comments	
		INTP5	PIOR4	/ P16/EO16/CCD00/TI(	/ 76	'	
		INTP6	PIOR5	Not assigned	Not assigned	None	
		INTP7	PIOR5	Not assigned	Not assigned	None	
		INTP8	PIOR5	Not assigned	Not assigned	None	
		INTP9	PIOR5	Not assigned	Not assigned	None	
		INTP10	PIOR1	Not assigned	Not assigned	None	
	<	INTP11	PIOR1	/ Not assigned	Not assigned	None	>

Figure 4-67 Filter for [Pin Function] Page

n Number							📕 🔛 🔛
ype filter tex	tt (* = any string, ? = any chara	acter)					All
Pin Numb	Pin Name	Board Functi	Function	Direction	Remarks	Symboli	All Pin Number
74	P57/INTP3		Not assigned	None			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 Symbolic Name
79	P13/IVREF1/EO13/TxD2/S	TxD2	Not assigned	None			Comments
80	P12/EI12/EO12/SO00/TxD		Not assigned	None			J1-2
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			J1-20
2							>

Figure 4-68 Filter for [Pin Number] Page

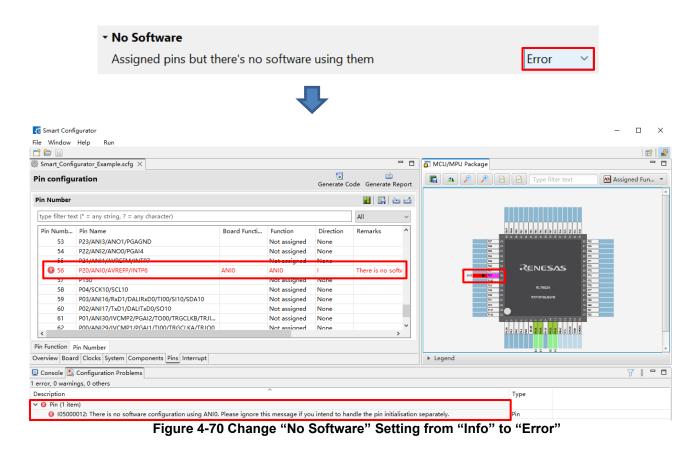
#### 4.5.9 **Pin Errors/Warnings setting**

User can control how pin problem is displayed on Configuration Problems view by using the Pin Errors/Warnings setting. If user wants 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.

C Preferences			$\times$
type filter text	Pin Errors/Warnings	⇔ ◄ ⇔	<b>▼</b> 8
> Help	- Pin Conflict		
<ul> <li>Smart Configurator Component</li> </ul>	Multiple functions are assigned in one pin number	Error	~
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	$\sim$
	- No Software		
	Assigned pins but there's no software using them	Info	$\sim$
	- Different Group		
	Functions in same channel but different group	Warnin	g ~
	- Board Mismatch		
	Pin assignment does not match the board suggested pin assignment	Warnin	g ~
< >	Restore Defaults	Appl	у
	Apply and Close	Cancel	

Figure 4-69 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.

ite	errupt configu	ration				Ge	nerate Code	denerate Rep
Int	errupt vectors us	ed						(1)
	Type filter text			Vector Numb	ber		`	
	Vector Number	Vector Table	Interrupt	Interrupt request source	Periphe	Priority	Status	Bank specify
	3	0000AH	INTP1	Pin input edge detection	INTC	Low	Used	None
	> 8	00028H	INTSRE1/INTTM03H			Low	Used	None
	24	00034H	INTAD	End of A/D conversion	ADC	Low	Used	None
	<							2

Figure 4-71 [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 you want to change priority setting on this page.
- (2) Click the priority cell and select an interrupt priority level from the drop-down list.

Smart_Configurator						1	
errupt configu	ration				Ge		e Generate Rep
terrupt vectors use	ed						E
Type filter text				Vector Numb	ber		,
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
					High		
<				(2)	Level 1		
-					Level 2 Low		

Figure 4-72 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 you 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.

errupt	vectors									8
Type fi	lter text					(1)			Vector Number	~
	Vector Tab		Interrupt request source		. Priority	Status	Bank specify	Remarks		^
-	00004H		Watchdog timer interval	WDT	Level 0 (high)	Used	None			
	00006H		Voltage detection	LVD	Level 1	0.000	æ 1	One register bank cannot be specified for multiple interrupt func-	tions with different priorities.	
	00008H		Pin input edge detection	INTC	Level 2	Used	2	(3)		
	0000AH		Pin input edge detection	INTC	Level 3 (low)	Used	3			
	0000CH		Pin input edge detection	INTC	Level 3 (low)		<b>&amp;</b> 1	One register bank cannot be specified for multiple interrupt func-	tions with different priorities.	
-	0000EH		Pin input edge detection	INTC	Level 3 (low)	Used	3			
6	00010H		Pin input edge detection	INTC	Level 3 (low)	Used	None 🗸			
	00012H		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)		5			
11	0001AH	INTELCL	Event link interrupt	ELCL	Level 3 (low)		None			~
view 8	Board Clocks	System Comp	onents Pins Interrupt							
nfigur	ation Problem	ns 🖾 Console								71-
ors, 1 v	varning, 0 oth	ners								
riptio					^			Type		

Figure 4-73 Change Interrupt Bank Setting Example

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

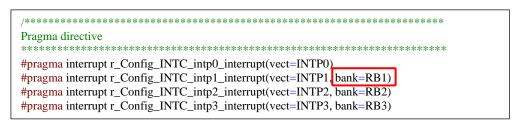


Figure 4-74 Interrupt Bank Setting Example (CS+ Project)

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 adds 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. User can 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 (¹³) 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 🗙			- 0
Software component configurat	ion	Generate Code	ian Generate Report
Components 🚵 🛃 ෫ 🕀 🕀	Configure		^
Ne liter text	Comparator operation setting	○ Operation	
<ul> <li>✓</li></ul>	Resolution setting	⊖ 8 bits	) 12 bits
<ul> <li>✓</li></ul>	VREF(+) setting VDD	OAVREFP	) Interna
Config_ADC	VREF(-) setting VSS	OAVREFM	
Overview Board Clocks System Compo	Trigger mode setting Software trigger no wait mode Software trigger wait mode c nents Pins Interrupt		>
Console 🔝 Configuration Problems			
6 errors, 0 warnings, 0 others			
Description   Solution   Output  Outpu	by INTAD0 in Config ADC Duplicate confl	ite with vector used by INTADO in	Config ADC
<ul> <li>E04010005: Interrupt vector used</li> <li>Peripheral (2 items)</li> </ul>	by INTAD0 in Config_ADC conflicts with ve	ector used by INTAD0 in Config_AD	
	erter used by Config_ADC_Duplicate is alrea erter used by Config_ADC is already used b		
8 E04010003: Pin used by ANIO in C	Config_ADC_Duplicate conflicts with pin use Config_ADC conflicts with pin used by ANIO		

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.

n configuration					G	enerate C	ode Genera	ate Repo
oftware Co 🕀 🖃 🕌	Pin Functi	on					9   🔳   🖫	🖻 🗹
Type filter text	type filter	text (* = ar	y string, ?	= any character)			All	~
> 🚣 r_bsp	Enabled	Function	PIOR	Assignment	Pin Nu	mber	Direction	Ren ^
✓ <u>♣</u> A/D Converter		🔇 ANIO		P20/ANI0/AVREFP/IN	/ 56		I.	Mul
or Config_ADC		ANI1		Not assigned	/ Not	assigned	None	
🗸 🚣 Ports		ANI2		Not assigned	/ Not	assigned	None	
or Config_PORT		ANI3		Not assigned	/ Not	assigned	None	
		ANI4		Not assigned	/ Not	assigned	None	
		ANI5		Not assigned	/ Not	assigned	None	
		ANI6		Not assigned	/ Not	assigned	None	
		ANI7		Not assigned	/ Not	assigned	None	
		ANI16		Not assigned	/ Not	assigned	None	~
	<							>
Function Pin Number								

Figure 5-2 Pin Conflicts

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

Sonfiguration Problems 🛛	<b>→</b>	$\nabla$	
3 errors, 0 warnings, 0 others			
Description		Туре	
🗸 😢 Pin (3 items)			
😣 E04010003: Pin used by ANI0 in Config_ADC conflicts with pin used by P20 in Config_PORT, pin used by P20 in Pin Allocator.	ļ	Pin	
8 E04010003: Pin used by P20 in Config_PORT conflicts with pin used by ANI0 in Pin Allocator, pin used by ANI0 in Config_ADC.	ļ	Pin	
📀 E05000010: Pin 117 cannot be used multiple times. Pin 117 is assigned to P20 and ANI0.	ļ	Pin	

Figure 5-3 Pin Conflict Messages

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

Software Comp	∃ ⊑ ↓ª₂ 🚜	Pin Func	tion
Type filter text		type filt	er text (* = any
∨ 🚣 r_bsp		Enable	d Function
🧉 r_bsp			🔇 P20
✓ 🚣 A/D Conver			CCD00
👩 Config_A		CCD01	
V 💑 Ports			CCD02
👩 Config	Assign all		CCD03
	Unassign all		CCD04
	Resolve conf	lict	
Pin Function Pin Nu	mber		
Overview Board Clo	cks System Co	mponents [	ins 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 Registering Generated Source Code with CS+

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

🔅 *Smart_Configurator_Example.scfg 💈	3	
Software component configur	ation	Generate Code Generate Report
Components	Configure	^
हीत 🔯 🔪 😜	Port selection PORT2	
✓ 🦢 Startup ∧ ✓ 🍃 Generic	□ Apply to all	
💣 r_bsp 🗸 🗸	<	>
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 the file is registered with the given project of CS+. If the Smart Configurator has already generated a file, a backup copy of that file is also generated (refer to chapter 8 Backing up Generated Source Code).

Smart Configurator Example - CS+ for CC - [Output]	– 🗆 X
	a a A
File Edit View Project Build Debug Tool Window Help	🤤 🕑 😭
🚳 Start 🚽 🗒 🖄 🖻 🖻 り 🤍 晶 単 斗 👘 🐻	📜 🔂 🗣 🐂 🔘 🕞 🗠 🚳 🏺
💎 💎 🔬 🥙 🤫 🗆 💷 📮 🥋 🖉 🖉 Solution List 🕴 🖉 🦉 🥁	
Project Tree 🕈 🗙 🏧 Property	- x
Son 2 Configurator Example (Project) R7F100GSNxFB (Microcontroller) R7F100GSNxFB (Microcontroller)	<b>→ → →</b>
Smart Configurator Example (Project)	
R7F100GSNxFB (Microcontroller)	Yes
Smart Configurator (Design Tool)	C source file
	No
ia j File	
main.c	
⊟ Smart Configurator	
E Config_ADC Build	
Config_ADC.c	
Config_ADC_user.c Build Settings / Build Settings /	Category Information /
general Output	4 ×
u r_bsp	+ ^
E-Config	
r_bsp_config.h	
E- Config_TAU0_1	
Config_TAU0_1.c	
Config_TAU0_1.h	
Config TAU0 1 user.c	-
Smart Browser Sector State	or List
F7 Open H F2 Rename F3 Find Ne F4 Replac F5 Go F6 Build & F7 Build Fr	
	1 Line 1 Column Read only

Figure 6-2 Registering a Source File with the CS+ Project



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

Figure 6-3 Configuration of Generated Files and File Names, shows the folders and files output by the Smart Configurator. Function *main ()* is included in {*Project name*}.c, which is generated when the project is created by CS+.

*r_xxx* indicates the names of Software Integration System Modules, "*ConfigName*" indicates the name of the configuration formed by the component settings, and "Project name" indicates a project name set in CS+.

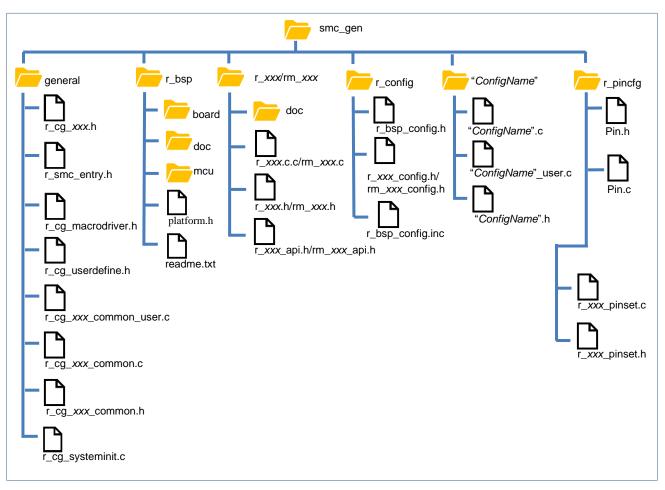


Figure 6-3 Configuration of Generated Files and File Names



Folder	File	Description
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 ^(Note*1)	The files contain macro definitions for setting SFR registers.
	r_smc_entry.h	This file is always generated.
		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.
		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 ^{(N} ote*1)	The files contain common interrupt API of used peripherals.
	r_cg_xxx_common.c ^(Note*1)	This file is generated when related peripherals are used.
	r_cg_xxx_common.h ^(Note*1)	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.
r_ <i>xxx</i> /		This folder is generated for the RL78 Software Integration System
rm_xxx ^(Note*1)		module that is added to the project.
		It consists of:
		<ul> <li>doc folder: Application note of this RL78 Software Integration</li> <li>System module</li> </ul>
		<ul> <li>r_xxx.c/rm_xxx.c^(Note*1): RL78 Software Integration System module source file</li> </ul>
		<ul> <li>r_xxx.c/rm_xxx.h^(Note*1): RL78 Software Integration System header file</li> </ul>
		- r_xxx_api.h/rm_xxx_api.h ^(Note*1) : List of all API calls and interface
		definitions of this RL78 Software Integration System module
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 manually.
	r_bsp_config.inc	This file is always generated.
		It generates configuration header file.
	r_xxx_config.h/rm_xxx_conf ig.h ^(Note*1)	These are configuration header files for all RL78 Software Integration drivers/middleware that are added to the project.



Folder	File	Description
r_pincfg	Pin.h	This file is always generated.
		It is generated for supporting pin symbol and included in smc_entry.h.
	Pin.c	This file is always generated.
		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_ <i>xxx</i> _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 <i>ConfigName</i> (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).
	{ConfigName}_user.c	This file contains interrupt service routines and functions for user to add code after the driver initialization (R_ <i>ConfigName</i> _Create). User can add codes and functions in the dedicated user code areas.
	{ConfigName}.h	This is header file for {ConfigName}.c and {ConfigName}_user.c.

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



## 6.3 Initializing Clocks

Configurations of the clock source selected in the [Clocks] page are generated to the macros in the r_bsp_config.h file located in \src\smc_gen\r_config folder. Clock initialization codes will be handled by r_bsp before entering main ().

The r_bsp_config.h file also contains other MCU related settings (for e.g., package, stack size).

Operation mode:       high-speed main mode 40(v)-55(v)         EVD0 setting:       40 V sEVD00 s 55 V         If high-speed on-chip oscillator       number of the scillator         Frequency:       32         V Middle-speed on-chip oscillator       number of the scillator         Frequency:       4         V Middle-speed on-chip oscillator       number of the scillator         Frequency:       4         V Middle-speed on-chip oscillator       number of the scillator         Frequency:       4         V for scillator       number of the scillator         Frequency:       5         V for scillator       number of the scillator         Frequency:       5         V for scillator       10         Frequency:       5         V for scillator       10         Frequency:       10         Size       0         V for scillator       10         Frequency:       10         Size							Generate	Code Generate Re
EXDD setting:       40 v ≤ EVDD s 5.5 v         I high-speed on-chip oscillator       100 million         Requency:       2         V dodle-speed on-chip oscillator       100 million         Requency:       0.00 million         V dodle-speed on-chip oscillator       0.00 million         Requency:       0.00 million         V dodle-speed on-chip oscillator       0.00 million         Requency:       5       0.04 million         Stable time:       0.01 million       100 million         Vider       100 million       100 million         Ti oscillator       100 million       100 million         Vider       0.01 million       100 million         Ti oscillator       100 million	Overation moder	nh-maad main movie 4.01/0 - E.E.0.0		-				
Image: section mode:       Image: section mode: <td< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></td<>								
Ingle resconding scaling       32       (MHz)         Prequency:       32       (MHz)         Imple resconding scaling       (MHz)       (MHz)         Imple rescondin				· ·				
Hequency:       2       0 (MHz)         Widdle-speed on-chip oscillator       132         Frequency:       4       0 (MHz)         Vider       1000       0 (MHz)         Stable time:       22 (MHz)       1000         Low-speed on-chip oscillator       1000       0 (MHz)         Frequency:       32 768       0 (Hz)         X11 oscillaton       5       0 (Hz)         VX11 oscillaton       5 (KHz)       5 (KHz)         X11 oscillaton       6 (Hz)       6 (Hz)         X11 oscillaton       6 (Hz)       7			_					(MHz)
Image: speed on-chip oscillator       32       (MHz)         Frequency:       4       (MHz)         Image: speed on-chip oscillator       8000       (MHz)         Image: speed on-chip oscillator       8000       (MHz)         Image: speed on-chip oscillator       9000       (MHz)         Image: speed on-chip oscillator       12.768       (Hz)	Frequency:	32 * (MHz)	T					
Image: Stable time:       21/10 collision         Vide=       0000         Bit Operation mode:       0000         Vide=       0000         Stable time:       2018/n/n         Vide=       0000         Vide=       0000 <td></td> <td></td> <td></td> <td></td> <td>_<u>)</u></td> <td>•</td> <td></td> <td>(MHz)</td>					_ <u>)</u>	•		(MHz)
Image: Speed on-chip oscillator     Image: Speed on-chip oscillator       Frequency:     4       Operation mode:     Stable time:       Stable time:     22144(us)								
Frequency: 4   Wider   Operation mode:   X11 oscillator   Frequency:   X2768 (Hz)   (Hz)   K1 oscillator   Frequency:   32.768 (Hz) K1 oscillator   Frequency:   X11 oscillator Frequency: S2.768 (Hz) K11 oscillator Frequency: S2.768 (Hz) Frequency: S2.768 (Hz)<	Middle-speed on-ch	ip oscillator						(kHz)
Operation mode:     X1 oscillation       Frequency:     32.768       (MHz)			<b>.</b>					(MHz)
Operation mode:     X1 oscillation       K1 oscillation     6kH2       X11 oscillation     52768       K11 oscillation     6kH2       K11 oscillation     5278       K11 oscillation     6kH2								
Frequency:     5     (MHz)       Stable time:     2*18/1x     262144(ux)       Low-speed on-chip oscillator     fl.       Frequency:     32.768     (MHz)       X11 oscillator     52768     (Hz)       Operation mode:     K11 oscillation     55/8       Frequency:     32.768     (Hz)       X11 oscillation     +     55/8       Y11 oscillation mode:     Low power consumption 1     +	_		Divider					
Stable time:         242144(µ)           Low-speed on-chip oscillator         fil.           Requency:         32.768           VI1 oscillator         52.768           Operation mode:         KT1 oscillaton           Frequency:         32.768           VI1 oscillator         52.768           VI1 oscillator         52.768           VI1 oscillator         52.768           VI1 oscillator         6.412	Operation mode:	X1 oscillation -	- x1 -				fMXP	
Low-speed on-chip oscillator         fil.         12.768         (kHz)           Frequency:         32.768         (kHz)         52.768         (kHz)           VI1 oscillation         *         52.768         (kHz)           Frequency:         32.768         (kHz)         52.768         (kHz)           X11 oscillation         *         52.768         (kHz)           X11 oscillation mode:         Low power consumption 1         *         52.768         (kHz)	Frequency:	5 (MHz)						(MHz)
X11 oscillator     32.768     (kHz)       X11 oscillator     22.768     (kHz)       Frequency:     32.768     (kHz)       Tri oscillation mode:     X11 oscillation     *       K11 oscillation     *     *       1000000000000000000000000000000000000	Stable time:	2^18/fx = 262144(µs)						
XT1 oscillator     32.768     (kHz)       YT1 oscillator     22.768     (kHz)       YT1 oscillaton     22.768     (kHz)       Frequency:     32.768     (kHz)       YT1 oscillation mode:     VT1 oscillation     15.00       YT1 oscillation mode:     Low power consumption 1     32.768			_					
X11 oscillator Operation mode: X11 oscillation Frequency: 32.768 (kHz) X11 oscillation mode: Low power consumption 1 *					•			(kHz)
XT1 oscillation         32.768         (kHz)           Operation mode:         XT1 oscillation         +           Frequency:         32.768         (kHz)           XT1 oscillation mode:         Low power consumption 1         +	Frequency:	32.768 (kHz)			I I			
Operation mode:     XT1 oscillation       Frequency:     32.768       XT1 oscillation mode:     Low power consumption 1							fSXP	
Frequency: 32.768 (kHz) 15XR 22.768 (kHz) 22.768 (kHz) 22.768 (kHz)	XT1 oscillator						32.768	(kHz)
Frequency:         32.768         (kHz)         32.768         (kHz)           X11 oscillation mode:         Low power consumption 1         *         *	Operation mode:	XT1 oscillation	•				FSYR	
	Frequency:	32.768 (kHz)			<b>.</b>			(kHz)
Supply mode: Enables supply in STOP.HALT mode +	XT1 oscillation mode:	Low power consumption 1	•					
	Supply mode:	Enables supply in STOP, HALT mode						

Middle-speed on-chip oscillator clock frequency Middle-speed on-chip oscillator frequency select register(MOCODIV) <u>MOCODIV1, MOCODIV0</u> 0 : 4MHz 1 : 2MHz 2 : 1MHz Other than above : Setting prohibited */ #define BSP CFG MOCO DIVIDE (0) /* Generated value. Do not edit this manually */



Folder	File	Macros/Functions	Description
r_config	r_bsp_config.h	Macros related to clocks	These settings are generated by Smart Configurator based on user's selection in the
			[Clocks] page for the clock source. <i>r_bsp</i> will handle the clock initialization before entering <i>main ()</i> .
		Macros related to MCU settings	Some MCU related settings are generated by Smart Configurator (e.g. package type) macros. For the detail macro information, user can refer to the application note in <i>r_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 8 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 ().

🐵 *Smart_Configurator_Example.scfg 🛛								- 0
Pin configuration							Generate Code	Generate Report
Software Compon $\textcircled{H} \boxdot \downarrow^{a}_{Z}$	Pin Functi	on					3	i 📓 🔤 🗹
	type filter	r text (* = any st	ring, ? = any ch	aracter)			All	~
∽ 💑 r_bsp	Enabled	Function	PIOR	Assignment	Pin Number	Direction	Remarks	
Image: state of the state o		TO01		P16/EO16/CCD00/TI01/TO01/INTP5/SI00/RxD0	/ E4	0		
Config_TAU0_1								
	<							>
Pin Function Pin Number								
Overview Board Clocks System Compo	onents Pins	Interrunt						

Figure 6-5 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 ().

n configuration						Gen	erate Code	Generate R	lepo
oftware 🕀 🖻 🖓 😹	Pin Functi	on					2		9 C.
Type filter text	type filter	text (* = a	ny string	g, ? = any character)		All			~
¥ 🚣 r_bsp € r_bsp	Enabled	Function TS00	PIOR	Assignment P50/TS00/EI50/EO50/CCD03/INTP1/SI11/SDA11	Pin Number / 33	Direction O	Remarks	Comments	^
✓ [™] / _™ r ctsu		TS01		/ P31/TS01/EI31/TI03/TO03/INTP4/PCLBUZ0	/ 21	0			
1_000		TSCAP		P30/VCOUT0/TSCAP/EI30/INTP3/RTC1HZ/SCK11/SCL11	/ 32	0			1

Overview Board Clocks System Components Pins Interrupt

Figure 6-6 *r_ctsu* in Software Components View

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

mart_Configuration rrupt configu	on_Example.scfg 🛛						
nupt coningu	lation						
errupt vectors							
Type filter text				Vector N	lumber		
Vector Number	Vector Table Address	Interrupt	Interrupt request source	Peripheral	Priority	Status	Bank spe
Vector Number 0	Vector Table Address 00004H	Interrupt INTWDTI	Interrupt request source Watchdog timer interval	Peripheral WDT	Priority Low	Status	Bank spe None
						Status	
	00004H	INTWDTI	Watchdog timer interval	WDT	Low	Status	None

Overview Board Clocks System Components Pins Interrupt

#### Figure 6-7 Interrupts Configuration in Interrupts View

Item	Folder	File	Component type	Description
Priority	{ConfigName}	{ConfigName}.c	Code Generator	It is initialized in <i>R_ConfigName_Create</i> of this file. r_cg_systeminit will call this function before entering main () function.
Bank	{ConfigName}	{ConfigName}_user.c	Code Generator	Declaration of interrupt as: #pragma interrupt "Interrupt API Name"(vect="Interrupt Name", bank=RBbankNumber), please see example in Figure 4-74



## 7. 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 files generated by the Smart Configurator.

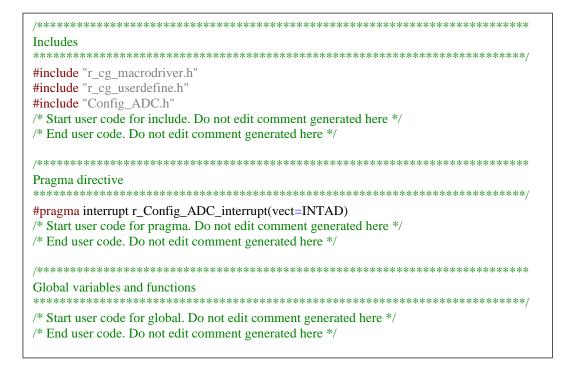
## 7.1 Adding Custom Code

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).





* 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 static void __near 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 */ } /* Start user code for adding. Do not edit comment generated here */ /* End user code. Do not edit comment generated here */



## 7.2 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 r_smc_entry.h".

Project Tree 4 X	Property 📓 main.c 🗸 🗸 🗸
2 🕜 🙎 🔳	11   10   ⇒ ⊂ ≤ S   Columns -
Smart Configurator Example (Project)	
R7F100GSNxFB (Microcontroller)	L C 23 /* DESCRIPTION :Main Program
Smart Configurator (Design Tool)	24 /* CPU TYPE :
CC-RL (Build Tool)	25 /*
RL78 Simulator (Debug Tool)	26 /* NOTE:THIS IS A TYPICAL EXAMPLE.
Program Analyzer (Analyze Tool)	27 /*
- File	28 /************************************
Build tool generated files	<pre>29 #include "r_smc_entry.h"</pre>
- I main.c	<pre>30 #include "r_ctsu.h"</pre>
Smart Configurator	31
B Config ADC	<pre>32 void main(void);</pre>
	33
i	34 void main(void) 35 -{
	36
□ r_config	37 }
□ □ r_ctsu	38
r ctsu.c	
r_ctsu_api.h	Output 🕂 🗸
r_ctsu_sms_asm.smsasm	[EOF]
r_ctsu_sms_asm_mutual.sms	
i≟⊷ 🔜 r_pincfg	
	All Messages *Rapid Build *Build Tool -
< >	🔜 Output 🍓 Smart Browser 🛅 Error List

Figure 7-1 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).

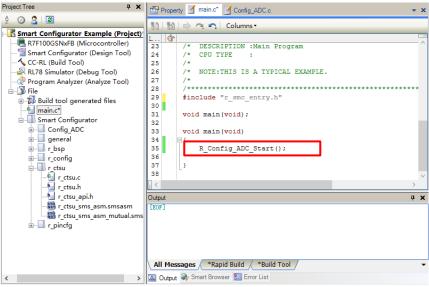


Figure 7-2 Call Code Generator Functions

**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 12 Documents for Reference".



## 8. 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.



## 9. Generating Reports

The Smart Configurator generates a report on the configurations that the user works on. Follow the procedure below to generate a report.

## 9.1 **Report on All Configurations (PDF or Text File)**

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

<pre> @ Smart_C </pre>	onfigurator_Example.scfg ×	
Overvie	w information	تا 🛁 👜 Generate Code Generate Report
- Genera	I Information	0
	<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 FAQ User's Guide API manual	Application Code Software Components Middleware & Drivers Device Drivers MCU Hardware
	FAQ : Smart Configurator t Configuration e code generation behavior can be customize	d in the Preferences page. ①
	board/device: R7F100LPLxFB (ROM size: 512	
Generate	ed location (PROJECT_LOC\): src\smc_gen	Edit 🗸
Overview	Board Clocks System Components Pins Inte	rrupt

Figure 9-1 Output of a Report on the Configuration (as a PDF/Text File)

👩 Smart Report				×
Generate report of configurations				
Options				
Print all sections				
O Print specific sections				
Board				^
Clocks				
System				
> 🗸 Components				
> Pins				
				~
✓ Output as PDF			<u>Select</u>	Font
Output as text				
C:\			Brov	/se
		_		
	ОК		Cancel	

Figure 9-2 Dialog Box for Output of a Report (Example is selecting "Output as PDF")



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

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

in configuration							Generate Co	de Generate	Repo
Hardware Resource 🕀 🕒 🖓	*	Pin Functio	on				3		<u>è</u> 2
		type filter	text (* =	any stri	ng, ? = any character)			All	~
🚣 All	^	Enabled	Functi	PIOR	Assignment	Pin Number	Direction	Remarks	^
👹 Clock Generator			ANIO		P20/ANI0/AVREFP/EI2	/ 117	1		
🗸 🎧 Comparator			ANI1		Not assigned	Not assigned	None		
COMP0			ANI2		Not assigned	Not assigned			
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	<b>`</b>	≪	ΔΝΙΟ		Not assigned	Not assigned	None		>

Figure 9-3 Output of a List of Pin Functions or Numbers (in csv Format)

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

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

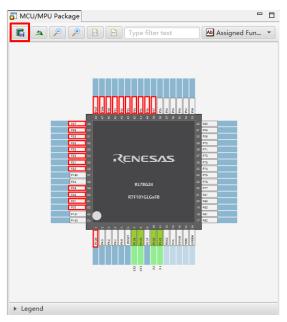


Figure 9-4 Outputting a Figure of MCU/MPU Package (in png Format)



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

The Smart Configurator for RL78 V1.5.0 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 10-1. 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".

## 10.1 **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 10-1 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.

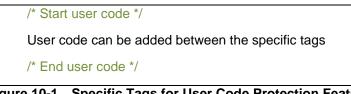


Figure 10-1 Specific Tags for User Code Protection Feature

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

Figure 10-2 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 10-1. 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.

	void R_Cor	nfig_ADC_Create(void)	
	{		
	ADCEN	= 1U; /* supply AD clock	*/
<pre>void R_Config_ADC_Create(void)</pre>	ADMK0	= 1U; /* disable INTAD0 i	nte:
{	ADIF0	= OU; /* clear INTAD0 int	err
ADCEN = 1U; /* supply AD clock */	/* Set	t INTAD0 priority */	
ADMK0 = 1U; /* disable INTAD0 int		0 = 10;	
ADIFO = OU;	ADPR0(		
/* set INTA Inserted the user code	/* Set	Licor codes will outon	natically be
ADPR10 = 10 with the specific tags		<pre>I= @ merged into the new</pre>	generated file
ADPR00 = 10	PM2  =		generated me
/* Set ANIO pin */		t AVREFP pin */	
PMCA2 = 0x01U;		$ = 0 \times 0 1 U;$	
PM2 = 0x01U;	PM2 [=	= 0x01U;	
ADM0 = 00 AD OPERMODE SELECT   00	ADM0 =	= 00 AD OPERMODE SELECT   0	0 A
ADM1 = C0 AD TRIGGER HARDWARE WAIT		= CO AD TRIGGER HARDWARE WAI	
ADM2 = 00 AD NEGATIVE VSS   00 AD		= 00 AD NEGATIVE VSS   00 A	
ADUL = _FF_AD_ADUL_VALUE;		= _FF_AD_ADUL_VALUE;	
ADLL = 00 AD ADLL VALUE;		= 00 AD ADLL VALUE;	
/* Start user code */		art user code */	
AWC = 0U;	AWC =		
/* End user code */		d user code */	
ADS = 00 AD INPUT CHANNEL 0;		00 AD INPUT CHANNEL 0;	
ADM2 &= 3F AD POSITIVE CLEAR; /*			/*
ADM2  = 00 AD POSITIVE VDD; /* s		= 40 AD POSITIVE AVREFP;	/*
	110110		·
R Config ADC Create UserInit();	R Cont	fig ADC Create UserInit();	
N_config_ncc_create_oserinit(),	1 1 10011	<u></u>	
3	1		

Figure 10-2 User Code Protection with Auto Merge



## 10.3 What to Do When Merge Conflict Occurs

#### 10.3.1 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 10-3 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: <u>src\smc gen\Config RTC\Config RTC.h</u>	
M0400001: File generated:src\smc gen\Config RTC\Config RTC.c	
M0000005: The above files highlighted in red color have user code merge conflicts, please ope	n the file and resolve the conflict manually
M00000002: Code generation is successful: <u>C:\cases\tempcases\rcp temp case\g24\src\smc gen</u>	v

Figure 10-3 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 10.3.2 Steps for Resolving the Merge Conflict described.



#### 10.3.2 **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 10-4 Code before Resolving Conflict).
- (2) Click on "Copy Current Change from Left to Right" (Figure 10-4 Code before Resolving Conflict).

g24_example.scfg			- 0
🕹 Text Compare		M 📰 😫 🔛 🚱 🕹	
Existing code	٠	New code	
/* Disable RTC clock operation */		/* Disable RTC clock operation */	A =
RTCE = 0U;		RTCE = 0U;	
/* Disable INTRTC interrupt */		/* Disable INTRTC interrupt */	
RTCMK = 1U;		RTCMK = 1U;	1.00
/* Clear INTRTC interrupt flag */		/* Clear INTRTC interrupt flag */	0
RTCIF = 0U;		RTCIF = 0U;	
/* Set fRTCCK */		/* Set fRTCCK */	
RTCC0 = 00 RTC CLK 32KHZ;		RTCC0 = 00 RTC CLK 32KHZ;	
/* Set 12-/24-hour system and period of Constant-period interrupt (INTRTC) */		/* Set 12-/24-hour system and period of Constant-period interrupt (INTRTC) */	
RTCC0 = ( 00 RTC RTC1HZ DISABLE   00 RTC 12HOUR MODE   00 RTC INTRTC NOT GENERATE);		RTCC0  = ( 20 RTC RTC1HZ ENABLE   00 RTC 12HOUR MODE   00 RTC INTRTC NOT GENERAT	E):
/* Start user code */	10	/* Set alarm detect function */	
POM3 &= ØxFEU:	L ( 4	WALE = 0U;	
/* End user code */	$\Lambda^{\infty}$	WALIE = 0U:	
, end det code ,	Γ.	/* Alarm function setting */	
R Config RTC Create UserInit();	, cý	ALARMWM = 00 BTC ALARM MTN:	
3	0	Copy Current Change from Left to Right	
	- 5	ALARMWW = 01 RTC ALARM WEEK;	
/**************************************		/ Set RTC1HZ pin /	
* Function Name: R Config RTC Start		POM3 &= 0xFEU:	
* Description : This function enables the real-time clock.		P3 &= 0xFEU;	
* Arguments : None		PM3 &= 0xFEU:	
* Return Value : None			
		R Config RTC Create UserInit();	
unid D Config DTC Start(unid)			×
< >>		٢	>
🖳 Console 🔝 Configuration Problems			B. 🚮 😥
mart Configurator Output			
M04050001: CWDW 0 is used for One shunt feature			
Me4050001: CWDW 0 is used for One shuft feature			
M04050001: CWDW 2 is used for One shunt feature			
100000001: Color generation is started (1)			
M04000001: File generated:src\smc gen\Config RTC\Config RTC.c			
TOTODOOT, ITTE Selectored, Sic (Sine Self Colling Kic (Colling Kic.)			

Figure 10-4 Code before Resolving Conflict

(3) Delete the codes that user does not want to use (Figure 10-5 Code after Applying "Copy Current Change from Left to Right").

i g24_example.scfg	
👃 Text Compare	M 🚍 😤 🔂 🚱 🕰 42 C
Existing code	New code
<pre>/* Clear INTRC interrupt flag */ RTCTF = 00; /* Set fRTCCK */ RTCC6 = 00 RTC_CLK_32HZ; /* Set 12-/24-hour system and period of Constant-period interrupt (INTRTC) */ /* Set 12-/24-hour system and period of Constant-period interrupt (INTRTC) */ /* Set 12-/24-hour system and period of Constant-period interrupt (INTRTC) */ /* Set 12-/24-hour system and period of Constant-period interrupt (INTRTC) */ RTCC0 = (.00_RTC_SIDEXBLE = 00_RTC_12HOUR_MODE = 00_RTC_2HHTC_NOT_GENERATE); /* Set user code */ PONS &amp;= 0xFEU; /* End user code */ R_COnfig_RTC_Create_UserInit(); } /* Function Name: R_Config_RTC_Start * Description :: This function enables the real-time clock. * Arguments : None *Return Value : None void R_Config_RTC_Start(void) { /* Enable RTC clock operation */ RTCE = 10; } </pre>	<pre>/* Set FRITCK */ RTCCG = 00 RTC_CLL_32KHZ; /* Set 12/24-hour_system and period of Constant-period interrupt (INTRTC) */ RTCCG = / 20 RTC RTCHZ_ENABLE   00 RTC_12HOUR_MODE   00 RTC_INTRTC_MOT_GENERATE); /* Start user code */ MALE = 00; /* Set alarm detect function */ WALE = 00; /* Set alarm detect function */ WALE = 00; /* Set RTCLALAR_MERCH; /* Set RTCLALAR_MERCH; /* Set RTCLALAR_MERCH; /* Set RTCLALAR_MERCH; P3 &amp;= 0xFEU; P3</pre>
Console 🔝 Configuration Problems	R. 🔐
Smart Configurator Output	
N94959801: CUDN 1 is used for One shunt feature N94959801: CUDN 1 is used for One shunt feature N94959801: CUDN 1 is used for One shunt feature N94989801: CuDM 2 is used for One shunt feature N949898001: File generated: <u>src\smc_gen\Config_RTC\Config_RTC.c</u>	

Figure 10-5 Code after Applying "Copy Current Change from Left to Right"



(4) Save the modified code (Figure 10-6 Code after Deleting and Saving).

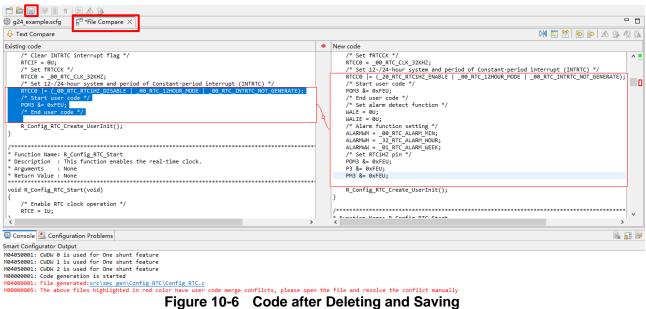


Figure 10-6 Code after Deleting and Saving

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 "File Compare" view.



## 11. Help

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

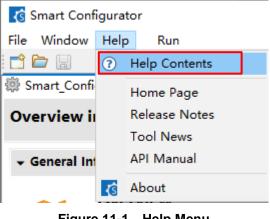


Figure 11-1 Help Menu

The help system can also be activated from the [Overview information] page by clicking ²⁰ button.

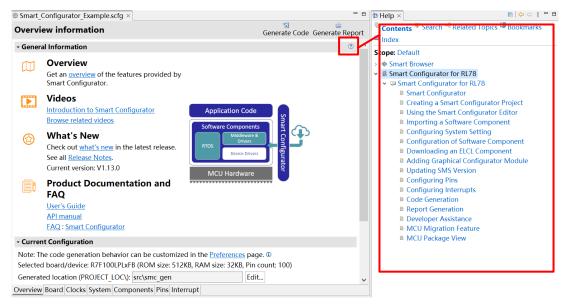


Figure 11-2 Quick Start

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



## 12. Documents for Reference

User's Manual: Hardware

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

Technical Update/Technical News

Obtain the latest information from the Renesas Electronics website.

User's Manual: Development Environment CS+ Integrated Development Environment User's Manual: Project Operation (R20UT4691) CS+ Integrated Development Environment User's Manual: RL78 Debug Tool (R20UT4692) CS+ Integrated Development Environment User's Manual: Message (R20UT4690) CC-RL Compiler User's Manual (R20UT3284) Smart Configurator User's Manual: RL78 API Reference (R20UT4852) (Obtain the latest version from the Renesas Electronics website.)

SMS & ELCL Application Notes:

Obtain the latest information from the website of Renesas Electronics.



## **Revision History**

		Description
Rev.	Section	Description
1.00	-	First edition issued
1.01	Section Introduction	URL was updated.
	Section 2 Before Using the Smart Configurator	2.2 Installing the Smart Configurator: URL was updated.
	Section 4 Setting of	4.4.12 Changing Version of BSP Configuration: Note was deleted.
	Peripheral Modules	4.4.13 Configure General Setting of Component: Figure 4-38 Configure General Setting of Component was updated.
		4.4.13 Configure General Setting of Component: Note 1 was updated.
		4.4.13 Configure General Setting of Component: Note 2 was updated.
		4.4.13 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.
	Section 11 Documents for Reference	SMS & ELCL Application Notes: SMS and ELCL reference was deleted.
1.02	Section 3 Operating the Smart Configurator	3.4.4 MCU/MPU Package View: Update description and Figure 3-5. MCU/MPU Package View.
	Section 4 Setting of	4.1.2 Selecting the Board: modify description
	Peripheral Modules	4.3 System Settings: Update description
		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 4-54.
	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 7 Creating User Programs	Add 7.2 Using Generated Code in User Application
	Section 10 User code protection feature for Smart Configurator Code Generation component	Add Section 10 User code protection feature for Smart Configurator Code Generation component.
1.03	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 reseting 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

6.

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. 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.
- 2. 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 electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
  - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

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

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

(Rev.5.0-1 October 2020)

## **Corporate Headquarters**

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

### www.renesas.com

#### Trademarks

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

## **Contact information**

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