

APPLICATION NOTE

RH850 Smart Configurator

User's Guide: CS+

R20AN0516EJ0130 Rev.1.30 Apr 22, 2024

Introduction

This application note describes the basic usage of the RH850 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) V8.11.00 and later
- RH850 Smart Configurator V1.11.0 and later
- CS+ RH850 Smart Configurator Communication Plugins V1.11.00 and later

Target Devices and Compilers

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

https://www.renesas.com/rh850-smart-configurator



Contents

1. Overview	4
1.1 Purpose	4
1.2 Features	4
2. Before Using the Smart Configurator	5
2.1 Preparing the CS+ (CS+ for CC) Integrated Development Environment	5
2.2 Installing the Smart Configurator	5
2.3 Setting the CS+ Integrated Development Environment	5
2.3.1 Checking the plug-in settings	5
2.3.2 Checking the setting of the execution path	6
2.4 Uninstalling the Smart Configurator	6
2.5 Preparing Sample Projects	7
3. Operating the Smart Configurator	8
3.1 Procedure for Operations	8
3.2 Starting the Smart Configurator	9
3.3 File to be Saved as Project Information	9
3.4 Window	10
3.4.1 Main menu	11
3.4.2 Toolbar	11
3.4.3 Smart Configurator view	12
3.4.4 MCU/MPU Package view	13
3.4.5 Console view	14
3.4.6 Configuration Problems view	14
4. Setting of Peripheral Modules	15
4.1 Board Settings	15
4.1.1 Selecting the device	15
4.1.2 Selecting the board	15
4.1.3 Exporting board settings	17
4.1.4 Importing board settings	18
4.2 Clock Settings	19
4.3 System Settings (only for RH850/U2A)	20
4.4 Component Settings	21
4.4.1 Adding Code Generator components	21
4.4.2 Switching between the component view and hardware view	23
4.4.3 Removing software component	24
4.4.4 Setting a Code Generator Component	25
4.4.5 Changing the resource for a Code Generate Configuration	26
4.4.6 Configure general setting of the component	29
4.5 Pin Settings	
4.5.1 Changing the pin assignment of a software component	
4.5.2 Assigning pins using the MCU/MPU Package view	
4.5.3 Show pin number from pin functions	
4.5.4 Exporting pin settings	34



4.5.5 Importing pin settings	35
4.5.6 Pin setting using board pin configuration information	36
4.5.7 Pin filter feature	37
4.5.8 Pin Errors/Warnings setting	38
4.6 Interrupt Settings	39
4.6.1 Changing the interrupt priority level and OS management setting	40
4.6.2 Changing the PE <i>n</i> setting(RH850/U2A only)	41
4.6.3 Changing the interrupt handler name, Generate Entity and Generate Enable/Disable Function setting	43
5. Managing Conflicts	45
5.1 Resource Conflicts	
5.2 Resolving pin conflicts	
	40
6. Generating Source Code	47
6.1 Registering Generated Source Code with CS+	47
6.2 Loading files generated by All Toolchain (CC-RH, GHS, IAR)	48
6.3 Configuration of Generated Files and File Names	50
6.4 Initializing Clocks	53
6.5 Initializing Pins	54
6.6 Initializing Interrupts	56
7 Creating Lloor Dragrama	E7
7. Creating User Programs	
7.1 Adding Custom Code in the Case of Code Generator	57
8. Backing up Generated Source Code	59
9. Generating Reports	60
9.1 Report on All Configurations	60
9.2 Configuration of Pin Function List and Pin Number List (in csv Format)	62
10 Lloor and protection facture	60
10. User code protection feature	
10.1 Specific tags for the user code protection feature	
10.2 Image of MCU/MPU Package (in png Format)	
10.3 Examples of using user code protection feature to add new user code10.4 What to do when merge conflict occurs	
10.4.1 What is Merge conflict	
10.4.2 Steps for resolving the merge conflict	
11. Help	68
11.1 Help	68
12. Documents for Reference	69
Website and Support	70



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 two functions to support the embedding of drivers from Renesas in your systems: generating driver code and making pin settings.



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 RH850 Smart Configurator and CS+ RH850 Smart Configurator communication plug-in from the URL below. The CS+ Smart Configurator communication plug-in is required for registering source code generated by the Smart Configurator with CS+.

https://www.renesas.com/rh850-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.

Note: Source code generated with All Toolchain (CC-RH, GHS, IAR) by Smart Configurator can also be added and built in the CS+ Integrated Development Environment, but no need to install CS+ RH850 Smart Configurator communication plug-in and set the CS+ Integrated Development Environment. Please refer to 6.2 Loading files generated by All Toolchain (CC-RH, GHS, IAR) for detailed procedure.

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 RH850 Communication Plug-in". Tick it if it is not.

×
Id by the CS+. Also, on the [Basic Function] tab, it is oller of the development are not cleared.
Description
Plugin to generate the device driver automatically (for V850). Plugin to generate the device driver automatically and to vie Plugin to generate the device driver automatically and to vie DebugConsole plugin to support using standard I/O. SEditor DLL It is a console where the IronPython commands and the CS+ Plugin to define the device pin configuration. Plugin to analyze program. Plugin for application development that contains useful tools Realtime OS Analysis Control plugin to use Realtime OS Info Realtime OS built tool plugin to set building information. Realtime OS plugin to display Realtime OS resource informat Plugin to analyze the Realtime OS built in program. Plugin to communicate with Smart Configurator for RH850 th Plugin to communicate with CS+ Update Manager. 1
OK Cancel <u>H</u> elp

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 - [Prop	ierty] —		×
File Edit View Project Build Debug Tool Wind	ow Help	ø	o 👸
🚳 Start 🔒 🔚 🎒 🐰 🖻 🚳 🔊 (~)	品 🏯 🏯 🔹 🔹 100% 👻 🗄 🚮 DefaultBuild 🔹 🔹 🐁 🖣 🗣 🔤 🎼		
~ ~ ~ @ ~ @ ~ _ _ _ _ _ _ _ _ _ _	Solution List 🕴 🤃 🚑 🧐		
🐺 Project Tree 🕴 🕈 🗙	Property 💋 Solution List		- x
Smart 2 @ 2 8	Smart Configurator Property	م	- +
R7E701651 (Microcontroller)			
GHS CCRH850 (Build Tool)	 Smart configurator setting Smart Configurator for RH850 executable file path C:\Program Files\Renesas Electronics\SmartConfigurator\RH850\eclipse\SmartConfigurator 	.exe	
-			

Figure 2-2 Displaying the Property

"Smart Configurator for RH850 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 "Smart Configurator" are in the same level).

"C:\Program Files (x86)\Renesas Electronics\SmartConfigurator\RH850\eclipse\SmartConfigurator.exe"

When manually specifying the path of the executable file, "Smart Configurator for RH850 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 RH850" and "CS+ SC Communication Plugins for RH850" from [Apps and Features] in the control panel and uninstall them.



2.5 Preparing Sample Projects

The Smart Configurator outputs source files for the main function and for the initialization of peripheral modules that were set up by using Smart Configurator components. However, the Smart Configurator does not output source files for the initialization that is performed between a reset of the microcontroller and the start of the main function or for the startup routine, which initiates the main function and executes other necessary processing.

Therefore, we provide sample projects that include sample startup routines and other necessary processing so that user applications can be built immediately after peripheral modules are set up using the Smart Configurator.

Refer to either of the documents stored in the following locations and create a CS+ project from the sample project.

"C:\Program Files (x86)\Renesas Electronics\SmartConfigurator\RH850\RH850C1M_SampleProjects"

"C:\Program Files (x86)\Renesas Electronics\SmartConfigurator\RH850\RH850F1KH_SampleProjects"

"C:\Program Files (x86)\Renesas Electronics\SmartConfigurator\RH850\RH850F1KM_SampleProjects"

"C:\Program Files (x86)\Renesas Electronics\SmartConfigurator\RH850\RH850U2A_SampleProjects"

"C:\Program Files (x86)\Renesas Electronics\SmartConfigurator\RH850\RH850U2B_SampleProjects"



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

Note: Sample project is provided by Smart Configurator for RH850 for easier usage, you can refer to chapter 2.5, Preparing Sample Projects for more information.



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.

Smart Configurator (1) File Window Help (2)		(4) - • ×
© c1m.scfg ×	TMCU/MPU Packa	ige ×
	Generate Code Generate Report	
User guide API manual • Current Configuration Selected board/device: R7F701275 (ROM size: 4 MB , RAM size: 128 KB, Pin count: 252) Generated location (PROJECT_LOC): src/smc_gen Edit Selected components: Component Version Configuration		RH850C1 MA2 R77701275
Overview Board Clocks Components Pins Interrupt	► Legend	
Smart Configurator Output 0 iter	nfiguration Problems ×	
(5) Figure 3-3 Mair	n Window (6)	

- 1) Menu bar
- 2) Main toolbar
- 3) Smart Configurator view
- 4) MCU 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.

Menu		Details
File	New	The dialog box [New Smart Configuration 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 RH850 API Reference (R20UT4361) in Renesas website
	About	The version information is displayed.

Table 3-1Main Menu Items

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]
	[File] ® [Open]
	[File] ® [Save]

Table 3-2	Toolbar Buttons and Related Menu Items



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.

Note: [System] page is supported only for RH850/U2A.

👩 Smart Configurator					_		\times
File Window Help							
⇔c1m.scfg ⇔u2a.scfg ×					1	۵	
Overview information					Generate Code		Report
- General Information							0
Overview Get an overview of th Configurator. Videos Introduction to Smar Browse related video Image: See all Release Notes Product Docum User quide API manual	<u>t Configurator</u> <u>s</u> <u>v</u> in the latest rele		Application Code Software Components Middleware & Drivers Device Drivers MCU Hardware	Smart Configurator	Ŷ		
- Current Configuration							
Selected board/device: R7F702	2300A (ROM size:	16 MB, RAM size: 3	3584 KB, Pin count: 516)				
Generated location (PROJECT_	LOC\): src\smc_g	en	Edit				
- Selected components:							
Component	Version	Configuration					
Overview Board Clocks System	Components Pin	s 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 between [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.
- [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.

The initial pin setting information of the board is the pin information of the board selected by [Board:] on the [Board] page (refer to "4.1.2 Selecting the board" and "Pin setting using board pin configuration information").

Note:

Symbolic Name feature is not applied to SC for RH850/F1KM and SC for RH850/F1KH.

Symbolic Name feature is not applied to APORT, JPORT and IPORT.



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.



Figure 3-6 Console View

3.4.6 Configuration Problems view

The Configuration Problems view displays the details of conflicts between pins.

🔝 Configuration Problems 🖾		₽	∇	-	
0 items					
0 items Description	Туре				

Figure 3-7 Configuration Problems View



4. Setting of Peripheral Modules

You can select peripheral modules from the Smart Configurator view.

4.1 Board Settings

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

4.1.1 Selecting the device

Click on the [____] button to select a device. This procedure is not required if you start the Smart Configurator from CS+.

smart Configurator		- [) ×
File Window Help			
			i 🗈 🗳
Smart_Configurator_Example.scfg ×			
Device selection	Generate Code	Generate	Report
Device selection			2 4
Board: Custom User Board Device: R7F702300			
Overview Board Clocks System Components Pins Interr	upt		

Figure 4-1 Selecting the Device

Note: Device change is not reflected to the device (microcontroller) of CS+ project.

4.1.2 Selecting the board

By selecting a board, the following settings can be changed at one time.

- Pin assignment (Initial pin setting)
- Frequency of the main clock
- Frequency of the sub-clock
- Target device

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

The .bdf file of Renesas made board (for e.g., Renesas Starter Kit) 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 the device shown in [Device] is different with device in file .bdf, dialog [Confirm device change] will popup:

💰 Smart Configurator	_		\times
File Window Help			
			i 🗈 🜋
#Smart_Configurator_Example.scfg ×			- 0
Device selection	🔞 Generate Code	Generate R	eport
Device selection			2
Board: Y-ASK-RH850F1KM-S4-V 🗸			
Device: R7F701649			
Confirm device change		×	
Changing the device will refresh all configurations. Configurations that are incompatible with the new device Do you want to continue?	ce may be remove	ed.	
Overview Board Clocks Console × Save and continue Continue Smart Configurator Output	Cance	el .	
M05000001: Pin Y11 is assigned to X1			~
M05000001: Pin Y10 is assigned to X2			\sim
M0000003: Report generated: output\migration_report_2023-05-14-2	<u>1-41-25.html</u>		
			~
<			>

Figure 4-2 Selecting the Board with different device

If the device shown in [Device] is same as the device in file .bdf, dialog [Confirm board change] will popup:

Smart Configurator	_	
File Window Help		
		i 🗈 📓
#*Smart_Configurator_Example.scfg ×		
Device selection	Generate Code	Generate Report
Device selection		24
Board: Y-ASK-RH850F1KM-S4-V 🗸		
Device: R7F701649		
Confirm board change Changing the board will refresh all pin assignments. Assigned pins that are unavailable in the selected board may be re Do you want to continue?	× emoved.	
Overview It Save and continue Continue	Cancel	₫ 🗉 🔻 📬 🔽 🗖
Smart Configurator Output		
M05000001: Pin 62 is assigned to X1		^
M05000001: Pin 61 is assigned to X2 M00000003: Report generated:output\migration_report_2023-05-14-	21-50-38 h+m]	
	21 30-30.110011	
<		~ ~

Figure 4-3 Selecting the Board with same device

Note: Depending on the board selected, the device will change, Device change is not reflected to the device (microcontroller) 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 [4 (Export board setting)] button on the [Board] tabbed page.
- (2) Select the output location and specify a name (Display Name) for the file to be exported.

Smart Configurator	_	\Box \times
File Window Help		
i 📫 🗁 🗒		: E2 🕰
Smart_Configurator_Example.scfg ×		- 8
Device selection	😼 Generate Code	🗎 Generate Report
Device selection		è C
Board: Custom User Board ~		
Device: R7F702300		
Overview Board Clocks System Components Pins	Interrupt	
🚨 Configuration Problems 💷 Console	🗟 🚮 🔂	🛃 🖃 🔻 📑 🗖
Smart Configurator Output		

Figure 4-4 Exporting Board Settings (bdf Format)



4.1.4 Importing board settings

Follow the procedure below to import board settings.

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

💰 Smart Configurator			\times
File Window Help			
			: 🖻 🖌 🐒
<pre></pre>			
Device selection	Generate Code	Generate F	Report
Device selection			
Board: Custom User Board 🗸 🗸			
Device: R7F702300			
Overview Board Clocks System Components Pins	Interrupt		
🚨 Configuration Problems 💷 Console	B. 🚮 🐶	🛃 🗉 🔻 📑	▼ □ □
Smart Configurator Output			

Figure 4-5 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

You can set the system clock on the [Clocks] tabbed 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 frequency of each clock in accordance with the board specifications (Note that the frequency is fixed for some internal clocks).
- (2) When using the PLL circuit, select the clock source for the PLL.
- (3) For the multiplexer symbol, select the clock source for the output clocks.
- (4) Enable the specific clock (only for RH850/F1KM and RH850/F1KH)
- (5) To obtain a desired output clock frequency, select a frequency division ratio from the drop-down list.



Figure 4-6 Clock Settings

4.3 System Settings (only for RH850/U2A)

You can select the CPUn (PEn) to be used at [System] tabbed page.

CPU0(PE0) is always selected to be used as the default setting.

Only RH850/U2A supports System settings.

👩 Smart Configurator	-		\times
File Window Help			
			i 🗈 💰
# Smart_Configurator_Example.scfg ×			- 8
System configuration	Generate Code	Generate R	eport
CPU core selection			^
CPU0(PE0) used (default			
r_cg_intvector_PE0.c and r_g_intc_PE0.c are generated under "\general" folder, WDT0 files are CPU1(PE1) used	generated (if W	DT0 is used).
r_cg_intvector_PE1.c and r_g_intc_PE1.c are generated under "\general" folder, WDT1 files are CPU2(PE2) used	generated (if W	DT1 is used).
r_cg_intvector_PE2.c and r_g_intc_PE2.c are generated under "\general" folder, WDT2 files are CPU3(PE3) used	generated (if W	DT2 is used).
r_cg_intvector_PE3.c and r_g_intc_PE3.c are generated under "\general" folder, WDT3 files are	generated (if W	DT3 is used).
			~
Overview Board Clocks System Components Pins Interrupt			
Figure 4-7 [System] Page			



4.4 Component Settings

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

🕼 Smart Configurator	-		\times
File Window Help			
			i 🖻 🜋
Smart_Configurator_Example.scfg ×			- 0
Software component configuration			Report
(b) [19	i		
type filter text			

Figure 4-8 [Components] Page

4.4.1 Adding Code Generator components

The following describes the procedure for adding a component.

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

👩 Smart Configurator	_		\times
File Window Help			
			i 🗈 蜜
# Smart_Configurator_Example.scfg ×			- 0
Software component configuration	Generate Code	Generate F	Report
Components 🟜 📫 🖞 🖻 🕀 🗄 Configure			(i)
(1) type filter text .			

Figure 4-9 Adding a Component

(2) Select a component from the list in the [Software Component Selection] page of the [New Component] dialog box (for e.g., PWM Output).



(3) Click on [Next].

	💽 New Co	omponent			×
	Software	Component Selection			
	Select con	nponent from those availabl	e in list		
	Category	All			~
	Function	All			~
	Filter				
	Compone		Version		^
	OS Tim		1.4.0		
		w Interrupt Output (input			
	Overflo Ports	w Interrupt Output (Widt	1.5.0		
(2)	PWM C	Dutput	1.5.0		
	#Real-Ti	me Clock	1.4.0		
	BRIIC Ma	aster	1.4.1		
	RIIC Sla	ve	1.4.1		
	SCI3 As	ynchronous Mode	1.2.0		
	SCI3 Cl	ock Synchronous Mode	1.2.0		
	Stand-l	by Controller	1.3.0		
	#TIM Bit	Compression Mode	1.1.0		
	#TIM Ga	ted Periodic Sampling Mo	1.1.0		
	#TIM Inp	out Event Mode	1.1.0		~
	Show o	nly latest version			
	Descriptio	in			
	PWM out	puts by using a master and	multiple slave cl		Ô
		general settings	al The nulse wid	th (duration) is set in the clave	•
	conligure	general settings			
			(3)		
	?	< Back	Next >	Finish Cance	:

Figure 4-10 Adding a Code Generator Component

- (4) Specify an appropriate configuration name in the [Add new configuration for selected component] dialog box or use the default name (for e.g., Config_TAUB0).
- (5) Select a hardware resource or use the default resource (for e.g., TAUB0).
- (6) Click on [Finish].

Rew Component	×
Add new configuration for selected component	
PWM Output	_
Configuration nar(14) Config_TAUB0 Resource: (5) TAUB0	~
(6)	
⑦ < Back Next > Finish Can	icel

Figure 4-11 Adding a Component



4.4.2 Switching between the component view and hardware view

The Smart Configurator also provides a function for adding a new component by directly clicking a node in the Components tree. To use this function, you need to switch the view of the Components tree from the component view to the hardware view.

(1) Click on the [] (View Menu)] icon and select [Show by Hardware View]. The Components tree will display the components in a hardware resource hierarchy.

Components	$ a_z $	E	Ð	+ → '	Configure
			(1)	•	Show by Component View
					Show by Hardware View
type filter text			5		

Figure 4-12 Switching to the Hardware View

- (2) Double-click on a hardware resource node (for e.g., TAUB10 under Timer Array Unit B1) to open the [New Component] dialog box.
- (3) Select a component from the list (for e.g., PWM Output Function) to add a new configuration as described in "chapter 4.4.1 Adding Code Generator components".



Figure 4-13 Adding a 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.

Components	è 🕹 🖓	⊑ 🛱 🜩 🕶	Compo	onents	è 4 🖞 🗉 🗗	, ∔ ⇒ ▼
					(-	1
type filter text			type f	ilter text		
🗸 🗁 Drivers			🗸 🗁 [Drivers		
Communication	ns		× 1	Communications		
Config CSIG				Config_CSIG0		
✓ Eoning_cond	0		~	⊳ DMA		
Config DMA	.C00	(1)		Config_DMAC00		
✓ Safety Function		(-)	~	Safety Function		
,				Config_DCRA0	(1)	
Config_DCR	40		~	⇒ Timers	(1)	
✓ → Timers				Config_TAUB0_0		
Config_TAUE	30_0		~	► A/D Converter		
✓ ➢ A/D Converter				Config_ADCA0		
Config_ADC/	A 0			_		

Figure 4-14 Removing a Component or Multiple Components

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

To delete the source files previously generated for the removed components from the 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 (for e.g., Config_TAUB0).
- (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 [PCLK/2] for [Clock source].
 - b. Select [Channel 1 slave], [Channel 2 slave], and [Channel 3 slave].
 - c. Specify [Pulse cycle] on the [Master0] tabbed page.
 - d. Specify [Duty] for each of the [Slave1], [Slave2], and [Slave3] tabbed pages.

🔇 Smart Configurator		_		>
ile Window Help				
* 🗁 🖫				1
Smart_Configurator_Example.sc				
Software component con	iguration 5 Generate Coc	de Gene	👜 erate Rep	ort
Components 🛛 🔤 🖆 🖳 🖶 🏶	Configure			^
type filter text Config TAUBO (1) Config TAUBO (2)	Clock setting Operation clock Clock source (2) a. PCLK/2 (Actual value: 20000kHz) PWM master select setting Master channel select 0 PWM slave setting Channel 1 slave Channel 2 slave Channel 3 slave Channel 4 slave Channel 5 slave Channel 6 slave Channel 7 slave Channel 8 slave Channel 9 slave Channel 10 slave Channel 11 slave Channel 12 slave Channel 13 slave Channel 14 slave Channel 15 slave PWM setting (2) d.			
(2)				
	<		>	

Figure 4-15 Setting a code generator configuration.

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 \checkmark 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 Generate Configuration

The Smart Configurator enables you to change the resource for a Code Generator configuration (for e.g., from TAUB0 to TAUB1). 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 configuration (for e.g., Config_TAUB0).
- (2) Select [Change resource] from the context menu.



Figure 4-16 Changing the Resource

- (3) Select a new resource (for e.g., TAUB1) in the [Resource Selection] dialog box.
- (4) The [Next] button will be active; click on it.

S Resource	Selection	1					\times
Resource Se	election						
Select resou	irce from t	those availa	ble in t	the list			
_							
	TAUB1						~
(3)	TAUB0 TAUB1						
	TAUDO		•				
	TAUJO TAUJ1						
	TAUJ2						
	TAUJ3						

Figure 4-17 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 or default settings.
- (8) Click on [Finish].

onfirm setting for resource change	● Use setting below ○ Use a	default
_{Setting} (6)	Value	Portable
Operation clock	CK0	Yes
Clock source	PCLK/2	Yes
Master channel select	0	Yes
Channel 1 slave	Enable	Yes
Channel 2 slave	Enable	Yes
Channel 3 slave	Enable	Yes
Channel 4 slave	Disable	Yes
Channel 5 slave	Disable	Yes
Channel 6 slave	Disable	Yes
Channel 7 slave	Disable	Yes
k.		-

Figure 4-18 Checking the Settings of the New Resource

The resource is automatically changed (for e.g., changed from INTTAUB0I0 to INTTAUB1I0).

👩 Smart Configurator				_		×
File Window Help						
📬 🗁 🗐						i 🗈 🌋
Smart_Configurator_Example.scfg	×					
Software component confi	guration		🕲 Generate Co	de Gene	👜 erate Rep	port
Components 🛛 🚵 🖄 🖻 🏵 🗮 🌩 🔻	Configure					^
type filter text	Clock setting Operation clock Clock source PWM master select settin Master channel select PWM slave setting Channel 1 slave Channel 4 slave Channel 7 slave Channel 10 slave Channel 13 slave PWM setting	CK0 ~ PCLK/2 ~ 9 0 ~ Channel 2 slave Channel 5 slave Channel 8 slave Channel 11 slave Channel 14 slave	(Actual value: 20000kHz) Channel 3 slave Channel 6 slave Channel 9 slave Channel 12 slave Channel 15 slave			
Overview Board Clocks Component	Priority	Slave3 4 nel 0 interrupt (INTTAUB110) Lowest) µs v (Actual value: 4)			~

Figure 4-19 Resource Changed Automatically



To change the configuration name, follow the procedure below.

- (9) Right-click on the configuration.
- (10) Select [Rename] to rename the configuration (for e.g., change Config_TAUB0 to Config_TAUB1).



Figure 4-20 Renaming the Configuration



4.4.6 Configure general setting of the component

You can change the general setting of the component such as backup settings and API function output setting. If you want to change it, go to [Window] on the menu -> [Preferences], select [Smart Configurator] -> [Component].

C Preferences	— 🗆 X
type filter text	Component $\diamond \star \star$
 Help Smart Configurat Component MCU/MPU Pac Pin Errors/War 	Backup settings
< >	Restore Defaults Apply
	Apply and Close Cancel

Figure 4-21 Configure general setting of component

Note:

 You can select [Enable the Backup settings] and limit the number of folders created in the trash folder for backup purposes by setting the [Number of trash item (1-20)] option in the figure below. Once exceeding the limit, a folder with the newer timestamp will replace the oldest folder. Setting 0 will disable this backup feature.

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

Figure 4-22 Trash number setting

2. If you want to only generate initialization API function, you 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		
API function output:	Output only initialization API function	~	
- FIT(RX) / SIS(RI 78) (Output all API functions according to the setting Output only initialization API function		
111(10() / 515(1(2/0) 0	Output only initialization API function		



This feature is supported from Smart Configurator for RH850 V1.4.0.



4.5 Pin Settings

The [Pins] page is used for assigning pin functions. You can switch the view by clicking on the [Pin Function] and [Pin Number] tabs. 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.

le Window Help							E
Smart_Configurator_Example.scfg ×							-
in configuration					Generate	Code Gen	erate Repo
Hardware Resource 🛛 🕀 🛱 🖧	Pin Fu	nction				3	🗉 🔛 🔤 i
	type f	ilter text (* = ar	iy string, ? = any character)			All	8
👗 All 🔨	Ena	Function	Assignment	Pin Num	Direct	Remarks	1
Clock generator		CAN6RX	Not assigned	Not assig	None		
🗯 I/O Ports		CAN6TX	Not assigned	Not assig	None		
Interrupt controller unit		CAN7RX	Not assigned	Not assig	None		
Reset Controller		CAN7TX	Not assigned	Not assig	None		_
Power Supply		CSCXFOUT	Not assigned	Not assig	None		
Low-Power Sampler		CSIGORYI	Not assigned	Not assig	None		
Sternal Memory Access Cor		CSIG0RYO	Not assigned	Not assig	None		
✓ ₩ Serial Flash Memory Interfac	\checkmark	CSIG0SC	/ P10_7/TAUD0I15/TAUD0O15/CS	/ C11 (2)	IO		
SFMA0		CSIG0SI	Not assigned	Not assig	None		
✓ [™] Clocked Serial Interface G	\checkmark	CSIG0SO	P10 6/TAUD0I13/TAUD0013/CS	/ A12 (2)	0		
		CSIG1RYI	Not assigned	Not assig	None		
CSIG1		CSIG1RYO	Not assigned	Not assig	None		
CSIG2		CSIG1SC	Not assigned	Not assig	None		
CSIG3		CSIG1SI	Not assigned	Not assig	None		
✓ NI Clocked Serial Interface H		CSIG1SO	Not assigned	Not assig	None		
CSIH0		CSIG2RYI	Not assigned	Not assig	None		
CSIH1		CSIG2RYO	Not assigned	Not assig	None		
CSIH2 Display swit	ching	CSIG2SC	Not assigned	Not assig	None		
CSIH3 ✓		CSIG2SI	Not assigned	Not assio	None		×
< >	<						>

Figure 4-24 [Pins] Page ([Pin Function])

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

7 🔛 Smart_Co	onfigurator_Example.scfg ×								
'in con	figuration					Generate		ie Generate f	Repor
in Numb	er								2
type filter	text (* = any string, ? = any	character)					All		~
Pin Nu	Pin Name	Board Functions	Function	Directi	Remarks	Symbolic N	lame	Comments	^
L25	X2	X2	X2	0		-			
K25	X1	X1	X1	I.		-			
AK10	P1_1/RLIN312TX/CAN8RX	TAUD001	■ JD001 N	1)					
AJ10	P1_0/RLIN312RX/INTP28/	TAUD0O0	CAN8RX	^					
C1	P11_0/MSPI6SC/TAUD0I0	TAUD0I0	INTP8						
G18	P2_10/GTM1I7/GTMAT1O	RLIN30TX	INTP14						
H6	P10_0/ICUMGPIO2/GTM1I.	RIICOSDA	DPIN14						
H7	P10_1/ICUMGPIO3/GTM0I.	RIICOSCL	MSPI9SI						
F17	P2_13/GTM1I2/GTMAT1O	MSPI0SO	TAUD00						
G17	P2_12/GTM0I4/GTMAT1O	MSPI0SC	PWGC14	D					
J21	P2_3/GTM1I4/GTMAT1O5	MSPI0CSS0	MSPIOCSSO	0					
AD9	AP0_0/ADCJ0I0	ADCJ0I0	ADCJ0I0	I.		-			
AE18	P4_5/GTM1I0/GTMAT0O4	ADCJ0CNV0	ADCJ0CNV0	0					
A1	VSS		VSS	-	Read only	-			
<	VCC		NCC		Deed end.				> ``

Figure 4-25 [Pins] Page ([Pin Number])



4.5.1 Changing the pin assignment of a software component

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

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

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

- (1) Click on [🚠 (Show by Hardware Resource or Software Components)] to switch to the component view.
- (2) Select the target software component (for e.g., Config_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 P10_13 to P11_2).
- (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.

ile Window Help						
Smart Configurator Example	e.scfa ×					
				5		a
Pin configuration (1)				Generate (Code Gener	ate Rep
Software Com 🐵 🖻 🎼 👪	Pin Function			(5) 🔃	🖬 🗠
Type filter text	type filter tex	xt (* = any string	g, ? = any character)		All	,
v 👗 Interrunt Controller	Enabled	Function	Assignment	Pin Number	Direct	Rema
) Config INTC		INTP1	P0 2/TAUD0I6/TAUD0O6/CAN1RX/INTP1/RLIN30TX/PWGA12O/		I	e
		INTP12 (4)	P11 2/CSIH2SO/RLIN32RX/INTP12/RLIN20TX/PWGA27O/TAU	/ <u>_</u> _9		
		INTPO	Not assigned			
		INTP2	P10 13/ CSIHOSSI/PWGA18O/RLIN32RX/INTP12/FLXA0TXENB/TA		IEMC0AD7	CAN71
		INTP3	P11 2/CSIH2SO/RLIN32RX/INTP12/RLIN20TX/PWGA27O/TAUB01			
		INTP4 (4				
		INTP5	P0 9/INTP12/CSIH1CSS0/DPIN7/RLIN22RX/TAUB0I4/TAUB0O4/C/			
		INTP6	 Not assigned 	Not assigned	None	
		INTP7	Not assigned	Not assigned	None	
		INTP8	Not assigned	Not assigned	None	
		INTP9	Not assigned	Not assigned	None	
		INTP10	Not assigned	Not assigned	None	
		INTP11	Not assigned	Not assigned	None	
		INTP13	/ Not assigned	Not assigned	None	
		INTP13 INTP14	Not assignedNot assigned	Not assignedNot assigned	None	
			5			
		INTP14	 Not assigned 	Not assigned	None	
		INTP14 INTP15	Not assignedNot assigned	Not assignedNot assigned	None None	
		INTP14 INTP15 INTP16	 Not assigned Not assigned Not assigned 	Not assignedNot assignedNot assigned	None None None	>

Figure 4-26 Pin Settings – Assigning Pins on the [Pin Function] List

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



4.5.2 Assigning pins using the MCU/MPU Package view

The Smart Configurator visualizes the pin assignment in the MCU/MPU Package view. You 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 [19 (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 4-27 Assigning Pins Using the MCU/MPU Package View



4.5.3 Show pin number from pin functions

You 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.

t Configurator	Example.scfg ×								
							1		
onfiguratio	n					Ge	nerate Co	ode Gener	rate R
are Componer	nts 🗉 🖻 🔩 🛃 Pir	n Functior	n						
filter text	ty	pe filter t	ext (* = any strir	ng, ? = any charac	ter)			All	
Interrupt Cont	roller E	nă Funo	ction Assignm	ient	Pin	Number	Direct	Remarks	
Config_INT			P1 / P10 6/	TAUD0I13/TAUD	0013/CSIG0 / A1	2	1		
		INTP	P12 / P10 13	3/ CSIH0SSI/PWG	A180/RLIN3 / C1	0	1		
		INTP	20 🔮 Not 📘	Jump to Pin N	lumber		None		
				Merge comme	ent to Pin Number	tab	None		
				Clear commer	nts		None		
				Assign selecte	d pins		None		
				Unassign selec	cted pins		None		
				signed	. Ni	ot assigned	None None		
						ot assigned	None		
				-		ot assigned	None		
						ot assigned	None		
						ot assigned	None		
						ot assigned	None		
				•		ot assigned	None		
				-		ot assigned	None		
		INTP				ot assigned	None		
		INTP		signed	* No	ot assigned	None		
						ot assigned	None		
				0		ot assigned	None		
							None		
nction in Nun w Board Clock	nber s Components Pins Ir		20 / Not as		* IN	ot assigned			2
w Board Clock	nber		20 Not as		2 Nr		-		×
W Board Clock	nber s Components Pins Ir		20 Not as		2 Nr		-	-	×
w Board Clock	ber s Components Pins Ir Configurator dow Help Run	nterrupt	20 V Not as		2 Nr		-		×
w Board Clock Smart File Wint C D Smart Smart Smart Smart Smart Smart	ber Components Pins Ir Configurator dow Help Run	nterrupt	20 / Not as	signed			-		×
W Board Clock	ber s Components Pins Ir configurator dow Help Run Configurator_Example.scfg	nterrupt	20 / Not as	signed	· N		-	e Generate	× Report
w Board Clock Smart File Wint C D Smart Smart Smart Smart Smart Smart	ber s Components Pins Ir configurator dow Help Run Configurator_Example.scfg	nterrupt	20 / Not as	signed			s nerate Code	e Generate ■] II	× Report
W Board Clock	ber s Components Pins Ir configurator dow Help Run Configurator_Example.scfg	nterrupt		₽	Direction		S erate Code	e Generate	× Report
w Board Clock	ber s Components Pins Ir configurator dow Help Run Configuration her Pin Name AP1_13/ADCA1113	nterrupt	Board Functions	Function Not assigned	Direction None	Gen	S erate Code	e Generate ₪ ⊑ All	× Report
W Board Clock	ber s Components Pins Ir configurator dow Help Run configurator_Example.scfc figuration ber Pin Name AP1_13/ADCA1113 P22_4	nterrupt		Function Not assigned Not assigned	Direction None None	Gen	S erate Code	e Generate ₪ ⊑ All	× Repor
W Board Clock	ber s Components Pins Ir configurator dow Help Run configuration ber Pin Name AP1_13/ADCA1113 P22_4 P22_5	nterrupt		Function Not assigned Not assigned	Direction None None None	Gen	S erate Code	e Generate ₪ ⊑ All	× Report
W Board Clock	ber s Components Pins Ir configurator dow Help Run configurator Example.scfg figuration her Pin Name AP1_13/ADCA1113 P22_4 P22_5 P10_2/TAUDOIS/TAUDO	a ×		Function Not assigned Not assigned Not assigned Not assigned	Direction None None	Gen	S erate Code	e Generate ₪ ⊑ All	× Report
W Board Clock	ber s Components Pins Ir configurator dow Help Run configurator Example.scl figuration ber Pin Name AP1_13/ADCA1113 P22_4 P22_5 P10_2/TAUD0IS/TAUDO P10_0/TAUD0IS/TAUDO P10_2/JNIP19/RLIN341	a ×		Function Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned	Direction None None None None None None None	Gen	S erate Code	e Generate ₪ ⊑ All	× Report
W Board Clock	ber s Configurator dow Help Run configurator Example.scfc figurator Example.scfc figurator District Configuration ber Pin Name Pin Name Pin Name Pin JAADCA1113 P22_4 P22_5 P10_2/TAUDOIS/TAUDO P10_2/TAUDOIS/TAUDO P10_2/TAUDOIS/TAUDO P10_2/TAUDOIS/TAUDO P10_2/TAUDOIS/TAUDOI P12_2/10/TAUDOI P10_2/TAUDOIS/TAUDOI P12_2/10/TAUDOI P12_2/1	DO5/RIIC X/PWGA		Function Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned	Direction None None None None None None None No	Gen	S erate Code	e Generate ₪ ⊑ All	× Report
W Board Clock	ber s Components Pins Ir configurator dow Help Run configurator Example.sc(s figuration ber Pin Name API 13/ADCA1113 P22_4 P22_4 P22_4 P22_4 P10_27/AUD015/TAUDO P10_27/AUD015/TAUDO P10_27/AUD017/AUDC P11_6/RUN33RX/INTP1 P11_6/RUN33RX/INTP1	DO5/RIIC X/PWGA		Function Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned	Direction None None None None None None None No	Gen	S erate Code	e Generate ₪ ⊑ All	Report
W Board Clock	ber s Configurator dow Help Run configurator Example.scfc figurator Example.scfc figurator District Configuration ber Pin Name Pin Name Pin Name Pin JAADCA1113 P22_4 P22_5 P10_2/TAUDOIS/TAUDO P10_2/TAUDOIS/TAUDO P10_2/TAUDOIS/TAUDO P10_2/TAUDOIS/TAUDO P10_2/TAUDOIS/TAUDOI P12_2/10/TAUDOI P10_2/TAUDOIS/TAUDOI P12_2/10/TAUDOI P12_2/1	DO5/RIIC DO5/RIIC DO1/CAN X/PWGA I3/CAN5		Function Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned	Direction None None None None None None None No	Gen	S erate Code	e Generate ₪ ⊑ All	× Report
W Board Clock	Pin Pins Ir configurator dow Help Run configurator Example.sclg figuration iber Pin Name AP1_13/ADCA1113 P22_5 P10_27/AUD015/TAUDO P10_27/AUD015/TAUDO P11_67/CINP32RZ/INTP19/RLIN33RZ/INTP1 P22_2 (AP116)/22/200015/TAUDO P11_67/CINP32RZ/INTP1 P22_14 P11_3/CS1H23SC/CAN38 P10_137_CS1H23SC/CAN38 P10_137/CS1H23SC/PWC P10_137/CS1H23SC/PWC	DO5/RIIC DO5/RIIC DO1/CAN X/PWGA I3/CAN5 I3/CAN5		Function Not assigned Not assigned	Direction None None None None None None None No	Gen	S erate Code	e Generate ₪ ⊑ All	× Report
W Board Clock	bber s Configurator dow Help Run Configurator Example.scfg figuration hber Pin Name AP1 13/ADCA1113 P22 4 P22 5 P10 2/TAUD0I5/TAUDC P10 2/TAUD0I5/TAUDC P12 2/NTP19/RUN34T P22 10 P12 2/IAUT93RX/INTP1 P12 3/ACSHI2SZ/CANB P10 13/ CSHI02SZ/PMC	005/RIIC 0075/RIIC 001/CAN X/PWGA 13/CAN5 X/INTP3. A180/R 00015/C		Function Not assigned Not assigned	Direction None None None None None None None No	Gen	S erate Code	e Generate ₪ ⊑ All	× Report
W Board Clock	Pin Name Pin Name AP1 13/ADCA1113 PPIN Name AP1 13/ADCA1113 P22 4 P22 5 P10 07ALD001/TAUDO P12 2/INTP19/RLIN34T P22 10 P12 2/INTP19/RLIN34T P22 10 P13 2/SIL2S/CAN38 P10 13/ CSIL2S/CAN38 P10 13/CSIL2S/CAN38 P10 13/CSIL2S/CAN38 P10 13/CSIL2S/CAN38 P10 13/CSIL2S/CAN38 P10 13/CSIL2S/CAN38 P10 13/CSIL2S/CAN38 P10 13/CSIL2S/CAN38 P10 7/TAUD015/TAUDO P10 2/TAUD015/TAUDO P10 7/TAUD0112/TAUD P10 2/TAUD0112/TAUD	005/RIIC 0075/RIIC 001/CAN X/PWGA 13/CAN5 X/INTP3. A180/R 00015/C		Function Not assigned Not assigned	Direction None None None None None None None No	Gen	S erate Code	e Generate ₪ ⊑ All	× Report
W Board Clock	bber s Configurator dow Help Run Configurator Example.scfg figuration hber Pin Name AP1 13/ADCA1113 P22 4 P22 5 P10 2/TAUD0I5/TAUDC P10 2/TAUD0I5/TAUDC P12 2/NTP19/RUN34T P22 10 P12 2/IAUT93RX/INTP1 P12 3/ACSHI2SZ/CANB P10 13/ CSHI02SZ/PMC	005/RIIC 0075/RIIC 001/CAN X/PWGA 13/CAN5 X/INTP3. A180/R 00015/C		Function Not assigned Not assigned	Direction None None None None None None None No	Gen	S erate Code	e Generate ₪ ⊑ All	× Report
W Board Clock	Pin Pins Ir configurator configurator configurator dow Help Run configurator configurator Example.scl configurator figuration biber configurator Pin Name API 13/ADCA1113 P22_4 P22_4 P22_4 P10_07/AUD017/AUDC P10_07/AUD017/AUDC P11_s/RUIN33RX/INTP1 P21_14 P10_s/CAUDO12/CAUE P10_s/CAUDO12/CAUE P10_s/CAUDO12/CAUE P10_s/CAUDO12/CAUE P10_s/LAUDCA1112S P10_s/LAUDCA1112S P10_s/LAUDCA1112S P18_11/ADCA1112S	005/RIIC 0075/RIIC 001/CAN X/PWGA 13/CAN5 X/INTP3. A180/R 00015/C		Function Not assigned Not assigned	Direction None None None None None None None No	Gen	S erate Code	e Generate ₪ ⊑ All	× Report
W Board Clock	Pin Pins Ir configurator configurator configurator dow Help Run configurator Example.scfg figuration tiguration ber Pin Pin Name AP1_13/ADCA1113 P22_4 P22_5 P10_2/TAUD0I5/TAUD P11_6/RLIN33RX/INTPI P21_16/RLIN33RX/INTPI P10_3/CSH25C/CANB P10_3/CSH25C/CANB P10_13/CSH25C/CANB P10_13/CSH25C/CANB P10_13/CAND P10_13/CAND P10_13/CAND P10_13/CAND P10_13/CADC P10_13/CAND	005/RIIC 0075/RIIC 001/CAN X/PWGA 13/CAN5 X/INTP3. A180/R 00015/C		Function Not assigned Not assigned	Direction None None None None None None None No	Gen	S erate Code	e Generate ₪ ⊑ All	× Report
W Board Clock	bbr s Components Pins Ir s Configurator Ir Ir dow Help Run Ir Ir configurator Example.scfg Ir Ir Ir dow Help Run Ir I	005/RIIC 0075/RIIC 001/CAN X/PWGA 13/CAN5 X/INTP3. A180/R 00015/C		Function Not assigned Not assigned	Direction None None None None None None None No	Gen	S erate Code	e Generate ₪ ⊑ All	× Report
W Board Clock	Pin Configurator configurator configurator dow Help Run configurator Example.sclg figuration bit Pin Name AP1_13/ADCA1113 P22_4 P22_7/AUD015/TAUD0 P10_2/7/AUD015/TAUD0 P10_2/TAUD015/TAUD0 P11_6/RUIN33RX/INTP1 P22_10 P11_6/RUIN33RX/INTP1 P10_2/TAUD015/TAUD0 P10_2/TAUD015/TAUD0 P10_2/TAUD015/TAUD015/TAUD0 P10_2/TAUD015/TAUD0 P11_6/RUIN33RX/INTP1 P22_14 P11_5/RUIN33RX/INTP1 P11_2/ROC1115 P10_9/TAUD015/TAUD0 P10_9/TAUD015/TAUD0 P11_9/RUIN3 P11_9/RUIN	005/RIIC 0075/RIIC 001/CAN X/PWGA 13/CAN5 X/INTP3. A180/R 00015/C		Function Not assigned Not assigned	Direction None None None None None None None No	Gen	S erate Code	e Generate ₪ ⊑ All	× Report
W Board Clock	bbr s Components Pins Ir s Configurator Ir Ir dow Help Run Ir Ir configurator Example.scfg Ir Ir Ir dow Help Run Ir I	005/RIIC 0075/RIIC 001/CAN X/PWGA 13/CAN5 X/INTP3. A180/R 00015/C		Function Not assigned Not assigned	Direction None None None None None None None No	Gen	S erate Code	e Generate ₪ ⊑ All	× Report

Figure 4-28 Jump to Pin Number



4.5.4 Exporting pin settings

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

- (1) Click on the [1] (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.

							B
Smart_Configurator_Example.scfg \times							- 0
in configuration				Ge	nerate Co	ode Generate Re	port
Hardware Resource 🛛 🕀 🛱 😹	Pin Fu	nction				-2 ⊞ ⊑ ≧	· ⊿
Type filter text	type f	filter text (*	= any string, ? = any character)			All	\sim
🛓 All 📃 🔨	Enă	Function	Assignment	Pin Number	Direct	Remarks	^
Clock generator	\checkmark	CSIGOSC	P0_14/INTP17/RLIN32TX/PWGA470,	/ P3 (1)	Ю		
\$⊯ I/O Ports	\checkmark	CSIG0SO	/ P0_13/RLIN32RX/INTP12/PWGA46O	/ R1 (1)	0		
Interrupt controller unit	\checkmark	INTP1	P10_6/TAUD0I13/TAUD0O13/CSIG05	/ A12	1		
Reset Controller	\checkmark	INTP7	P8_3/TAUJ0I1/TAUJ001/DPIN3/CSIH	/ Y15	1		
Power Supply	\checkmark	INTP8	P8_4/TAUJ0I2/TAUJ0O2/DPIN4/CSIH	# W15	1		
Low-Power Sampler		INTP12	/ P11_2/CSIH2SO/RLIN32RX/INTP12/F	/ D9	1		
External Memory Access Cor		P0 1	P0 1/TAUD0I4/TAUD0O4/CAN0RX/I	/ M2	Ю	There is no soft	N I
 Serial Flash Memory Interfac 	\checkmark	TAUB015	/ P10_13/_CSIH0SSI/PWGA18O/RLIN3.	/ C10	1		
SFMA0	\checkmark	TAUD0I9	/ P10_4/TAUD0I9/TAUD0O9/RLIN21R	/ E3	1	There is no soft	N I
 Markov Markov M Markov Markov Ma Narkov Markov Marko		AOVREF	Not assigned	Not assigned	None		~
		A 01/CC	• Not assigned	• Not accloned	Mana	>	*

Figure 4-29 Exporting Pin Settings to an XML File

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



4.5.5 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.

						i e	Ľ
Smart_Configurator_Example.scfg ×							-
Pin configuration				Ge	enerate Co	ode Generate Rep	рс
Hardware Resource 🛛 🕀 🖻 🖧 🍰	Pin Fur	nction				२। 🗉 🖪 🖻	- 1
Type filter text	type f	ilter text (*	= any string, ? = any character)			All	~
All Clock generator I/O Ports Interrupt controller unit Reset Controller Power Supply Low-Power Sampler External Memory Access Cor Serial Flash Memory Interfac SFMA0 SEMA0		Function CSIG0SC CSIG0SO INTP1 INTP7 INTP8 INTP12 P0_1 TAUB0I5 TAUD019 A0VREF	Assignment P0_14/INTP17/RLIN32TX/PWGA47O, P0_13/RLIN32RX/INTP12/PWGA46O P10_6/TAUD0I13/TAUD0013/CSIG0S P8_3/TAUJ0I1/TAUJ001/DPIN3/CSIH P8_4/TAUJ0I2/TAUJ002/DPIN4/CSIH P11_2/CSIH2SO/RLIN32RX/INTP12/F P0_1/TAUD0I4/TAUD004/CAN0RX/I P10_13/_CSIH0SSI/PWGA18O/RLIN3. P10_4/TAUD0I9/TAUD009/RLIN21RC Not assigned	 R1 (1) A12 Y15 W15 D9 M2 C10 	Direct IO O I I I IO I I I None	Remarks There is no softw There is no softw	
< CSIG0 ×	<		Not assigned	* Not assigned		>	

Figure 4-30 Importing Pin Settings from an XML File

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



4.5.6 Pin setting using board pin configuration information

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

The following describes the procedure for collective setting of pins.

- (1) Select [Board Function] in the MCU/MPU Package. (The initial pin configuration of the board can be referred.)
- (2) Open the [Pin Configuration] page and click the [Assign default board pins]
- (3) When [Assign default board pins] dialog opens, click [Select all].
- (4) Click [OK].

mart Configurator Example.scfg	×	• •	MCU/MPU Package ×	
n configuration		😼 👜 Generate Code 🛛 Generate Report	Type filter text	d Function * (1)
nt : This button 🖬 will assign pin	function to the default board pins if a board is selecte	ed. Don't Show Again 🕷		
ardware Resource 🛛 🕀 🖶 🦓	Pin Function	≈ 🔳 ∓ (2*) ∞		
Type filter text	type filter text (* = any string, ? = any character)	C Default Board Pin		
All Clock generator I/O Ports Interrupt controller unit	Ena Function Assignment A0VREF / Not assigned A0VSS / Not assigned A1VREF / Not assigned	Assign default board pins (3)		
Reset Controller Power Supply Low-Power Sampler External Memory Acces Serial Flash Memory Int Clocked Serial Interface Clocked Serial Interface Clocked Serial Interface ELIN Acter Interface ELIN Acter Interface ELIN ZE Bus Interface ELIN CLORE CONTROL CONTROL Second Seco	A1VSS > Not assigned ADCA0I0 > Not assigned ADCA0I1 > Not assigned ADCA012 > Not assigned ADCA013 > Not assigned ADCA013 > Not assigned ADCA015 > Not assigned ADCA015 > Not assigned ADCA016 > Not assigned ADCA017 > Not assigned ADCA018 > Not assigned ADCA019 > Not assigned ADCA019 > Not assigned ADCA019 > Not assigned ADCA019 > Not assigned	Assign Pin Function ADCA010 ADCA011 ADCA012 ADCA012 ADCA013 ADCA013 ADCA03 ADCA03 ADCA03 CAN3RX CAN3RX CAN4RX		
ಿ FlexRay ಟಿ Ethernet AVB > ಟಿ Single Edge Nibble Trar	ADCA0110 / Not assigned ADCA0111 / Not assigned ADCA0111 / Not assigned ADCA0112 / Not assigned	(4) OK Cancel	R7F701649	
Source Service Se	ADCA0I13 / Not assigned ADCA0I14 / Not assigned ADCA0I15 / Not assigned ADCA0I5 / Not assigned	Not assig None Not assig None Not assig None Not assig None		
 & Reartime clock & Encoder Timer 	ADCA0I0S / Not assigned ADCA0I10! / Not assigned	Not assig None Not assig None		

Figure 4-31 Setting for Initial Pin Configuration

If you do not set pin settings all at once, specify them individually in procedure (3).


4.5.7 Pin filter feature

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

Pin Fur	nction	🤣 🔛 🖬 🔒	<u>b</u> 2							
type fi	type filter text (* = any string, ? = any character)									
Ena	Function A0VREF A0VSS A1VREF A1VSS ADCA0I0 ADCA0I1	Assignment Vot assigned Not assigned Not assigned Not assigned Not assigned Not assigned Not assigned	Pin Num Vot assig Not assig Not assig Not assig Not assig Not assig Not assig	None None None None	Remarks	Comm	All Function Assignment Pin Number Direction Remarks Comments			

Figure 4-32 Filter for [Pin Function] tab

Pin Num	ber					💷 🖪 🗠 e
type filte	er text (* = any string, ? = any chara	cter)				Symbolic Name 🗸
Pin N	Pin Name	Board Functions		Direct	Remarks	All Pin Number
A1 A2	BVSS P22 7		BVSS Not assign	- None	Read only	Pin Name Board Functions
A3	P13_1/MEMC0A20		Not assign			Function
A4 A5	P22_9 P11 15/CAN2RX/INTP2/CSIH2C		Not assign Not assign			Direction Remarks
A 6	P22_12		Not assign			Symbolic Name
A7	P22_13		Not assign	None		Comments

Figure 4-33 Filter for [Pin Number] tab



4.5.8 Pin Errors/Warnings setting

You can control how pin problem is displayed on Configuration Problems view by using the Pin Errors/Warnings setting. If you want to control it, click Menu [Window]->[Preferences] 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	⇔ ◄ ⇔	▼ 8
 Help Smart Configurator Component MCU/MPU Package Pin Errors/Warnings 	 Pin Conflict Multiple functions are assigned in one pin number No Pin Allocation 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. No Software Assigned pins but there's no software using them 	Error Info	~
	 Different Group Functions in same channel but different group Board Mismatch Pin assignment does not match the board suggested pin assignment 	Warnin Warnin	
< >>	Restore Defaults Apply and Close	Apply Cancel	/

Figure 4-34 Pin Errors/Warnings settings at Preferences

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

ASS	igned pins bu	it there's no	o sonwa	are usir	ig them								Erro	ſ		
诸 Smart Cor	6															X
	-													_		~
ile Window	Help															
<u>) 🖻 🛛</u>					- 8			D I								
Smart_Confi	gurator_Example.scfg	×	6	el	~			Packag	L							
Pin config	uration				nerate Report		n 🔎	۶		Type filter	text	Ab	Assigned Fu	unction *		
Pin Number					8 6 è 4		A 945 92									
							· 20.6 PH		PIL 3 PIL 11				PIL PIL (» ·	
type filter te	ext (* = any string, ? = a	ny character)		Symb	olic Name $^{\vee}$		c 900,4 900.	5 MJ MJ	FIL.3 FIL.18	PILS (0.14)	1.3 (11.3) (11.3)	PR. 9 PR. 8			91.35 (FL) (
Pin Numb	Pin Name	Board Functions	Function	Direction	Remarks ^		· • •	a na a acc	BACE FELS	100 ND (na) ees (na	avcc avs	NH & ROVER RO	MG BACC AN J	RC RC +	
P4	EVCC		EVCC	-	Read only											
P17	AP0 6/ADCA0I6		Not assigne	None						R	ENES	SΔS				
P18	AP0 3/ADCA0I3		Not assigne	None										NC (10.1		
P19	AP0 1/ADCA0I1		Not assigne	None						(NTS 875 875	875		M .		
P20	AP0 0/ADCA0I0		ADCA0I0	1	There is r		× 10,4 10,	· ··· ·		(PTS (PTS (PTS	845		EVE (72).3	10.30 F20.3 4	
R1	P0_13/RLIN32RX/IN		Not assigne	None			L P2.3 P2.	9 PL 913		(nis (255 (275	845		100VCC 918,33	10.10 10.10 1	
R2	P1_1/INTP18/RLIN3		Not assigne	None			H (0.8 (0.	1 HA HAZ		(nas (nas (nas	evss		8066 (91.3	PEA PEA P	
R3	P2_6/ADCA0SEL2		Not assigne				N (0.4 (0.)	1 PEA (PEA						PM5 PR.1		
R4	P1_3/INTP19/CAN3		Not assigne								R7F7016				Ľ	
R17	A0VREF		Not assigne				1 8.3 8.3							ANS HE IS		
R18	AP0_7/ADCA0I7		Not assigne						PL 1	EICC ##245 8	10 10 10 10 10 10 10 10 10 10 10 10 10 1		RAS RAC (
R19	AP0 4/ADCA0I4		Not assigne	None	>		·		PR.0 (1.11)			· · · · · · · · · · · · · · · · · · ·	nu no (1) ns ns	NO. 10 NO. 0	
<					7		* 12.0 10		n			n. n.	R. R. (10.8 MIL.8	
Pin Function							Y						R.H. (1)		P3 898	
Overview Boa	rd Clocks Components	Pins Interrupt				► Legen	d , ,	, ,	5 6	7 8	9 10 11	и в	ы 15	x 0 u		
Configuratio	on Problems ×														7	8 🚥
	ings, 0 others															
, Description				^											Туре	
🛛 🛛 Pin (1 ite	>														21.5	

Figure 4-35 Change "No Software" setting from "Info" to "Error"



4.6 Interrupt Settings

Check and set the interrupts of the peripheral modules that have been selected on the [Components] page. The interrupts are displayed for each of the vector numbers. Generally, you can set the common settings such as interrupt priority levels, OS management, Interrupt Handler, Generate Entity and Generate Enable/Disable Function. For RH850/U2A, you can set PE*n* to decide if the interrupt is applied to the PE*n*.

Smart Conf	iguiator						- 🗆
e Window	Help						
Smart_Config	jurator_Example.so	cfg ×					
nterrupt c	onfiguration				Ge	🗓 enerate Co	de Generate Re
nterrupt vec	tors used						
Type filter	text		Vector	Number			
Vector N	Exception Sou	Interrupt	Interrupt request source	Periph	Priority	Status	OS manage
18	1012H	INTADCA0I0	ADCA0 scan group 1 (SG1) end inte	ADCA0	Lowest	Used	
> 27	101BH	INTCSIG0IC/IN			Lowest	Used	
56	1038H	INTADCA0ERR	ADCA0 error interrupt	ADCA0	Lowest	Used	
60	103CH	INTDMA0	DMA00 transfer completion	DMAC0	Lowest	Used	
142	108EH	INTTAUB010	Interrupt for TAUB0 channel 0	TAUB0	Lowest	Used	

Figure 4-36 [Interrupt] Page



4.6.1 Changing the interrupt priority level and OS management setting

When an interrupt is used in a configuration on the [Components] page, the status of the interrupt will be changed to "Used". To display the used interrupts only, click on the [M] (Show used interrupts)] button.

- (1) You can change the interrupt priority level on the [Interrupt] page.
- (2) The [OS management] column becomes active for a project that uses RTOS (RI850V4). Selecting a checkbox in the column outputs the corresponding interrupt function in the interrupt format that can be managed by the OS.

Window I	lelp								
		C							
man_conig	gurator_Example.							1	e
errupt co	onfiguration							• ate Code Gener	-
terrupt vect	ors used								
Type filter t	ext			Vector	Number				
Vector N	Exception Sou	Interrupt	Interrupt request source	Periph	Priority		Status	OS managen	nent (
18	1012H	INTADCA0I0	ADCA0 scan group 1 (SG1) end inte	ADCA0	Lowest	~	Used		
> 27	101BH	INTCSIG0IC/IN		(1)	Level 1	^	Used		
56	1038H	INTADCA0ERR	ADCA0 error interrupt	ADCA0	Level 2		Used		
60	103CH	INTDMA0	DMA00 transfer completion	DMACC	Level 3		Used		
142	108EH	INTTAUB010	Interrupt for TAUB0 channel 0	TAUB0	Level 4 Level 5		Used		
					Level 6				
					Level 7				
					Level 8				
					Level 9				
					Level 10				
					Level 11				
					Level 12 Level 13				
					Level 13				

Figure 4-37 Interrupt Settings



4.6.2 Changing the PE*n* setting(RH850/U2A only)

In Smart Configurator for RH850, you can select which PEn to respond to the interrupt in use. PEn can be set on the Interrupt page by below steps:

PEn can be set on the [Interrupt] page by below steps: PEn is chosen to be used in [System] page (please refer to chapter 0,



- (1) System Settings (only for RH850/U2A))
- (2) Check or uncheck the checkbox in column PE*n* in [Interrupt] page to select which PE to respond to the interrupt. There are two types of interrupts:
 - a Connected to INTC1 of each PE, each PE selected in the PE*n* column can respond.
 - b Connected to INTC2 shared by multiple PEs, only one PE selected in PEn column can respond.

		urator_Example_U	JEM.SCIG X									5		۵
											Gener	rate Co	de Ger	nerate R
	errupt vec													
1	Type filter t							Vector Nu						
		Exception Sou		Interrupt request source		. Priority	Status		nage I		PE1	PE2	PE3	
	26 27	101AH 101BH	INTTAUD0114	TAUD0 Channel 14 interrupt TAPA0 peak interrupt 0	TAUD	Lowest Lowest					12	(2)	121	
	27	101CH	INITTAPAOIVI VO		TAPA	Lowest				2	120	(3)	a	
, I	29	101DH	INTSDMACERR	sDMAC0 address error or sDMAC1	s DMAC	Lowest	Used	0			8	12	1	
	30	101EH	INTDTSERR	DTS transfer error	DTS	Lowest	Used		19		V	E	10	
	31	101FH	INTTPTM0	TPTM0 Timing Protection Violation	TPTM	Lowest			1	2				
	31	101FH	INTTPTM1	TPTM1 Timing Protection Violation	TPTM	Lowest					2			
	31	101FH	INTTPTM2	TPTM2 Timing Protection Violation	TPTM	Lowest						121		
	31	101FH	INTTPTM3	TPTM3 Timing Protection Violation	TPTM	Lowest							<u>(7</u>)	
	34	1022H		FPSYS0 Flash sequencer processing		Lowest			6	9		(3)	b	
	36	1024H	INTEL2ENDNM	FPSYS1 Flash sequencer processing	EACI2	Lowest			9	8	0	101	Ø	
	39	1027H		DTS ch31-0 transfer end	DTS	Lowest	Used			-	0	6	8	
	40	1027H		DTS ch63-32 transfer end	DTS	Lowest	UNCU	0	-	2 8]	0	0	10	
	40	1029H		DTS ch95-64 transfer end	DTS	Lowest			0	9				
	42	102AH		DTS ch127-96 transfer end	DTS	Lowest			18	8	0	0	0	
	43	102BH		DTS ch31-0 transfer count match	DTS	Lowest	Used				0		0	
	44	102CH		DTS ch63-32 transfer count match	DTS	Lowest				8				
	7 Smar	t Configurato	r											X
F	File Win	t Configurato dow Help		sefa X							_	-		×
100	File Win C C C C C C C C C C C C C C C C C C C	dow Help	Example_U2A	.scfg ×					Genera	🖬	Code	Gen		
1	File Win	dow Help Configurator_ I configura	Example_U2A	.scfg ×							Code	Gen		
100	File Win Smart_t System CPU co CPU co CPU co CPU r_cg_in CPU r_cg_in	dow Help Configurator_ a configurator cre selection 0(PE0) used (a tvector_PE0.c 1(PE1) used	Example_U2A ation default) and r_cg_intc_	.scfg × _PE0.c are generated under "\ _PE1.c are generated under "\	-				Genera genera	ate ((if W	'DT0 i	erate s used	Repor
100	File Win File Win Smart_t System CPU co CPU co C	dow Help Configurator_ a configurator_ a confi	Example_U2A ation default) and r_cg_intc_ (2) and r_cg_intc_	PE0.c are generated under "\	general general	" folder, " folder,	WDT1	files are files are	Genera genera genera genera	ted ted	(if W (if W (if W	'DT0 i 'DT1 i 'DT2 i	erate s used s used	Repor

Note:

Only RH850/U2A supports PEn setting.

For other microcontrollers such as RH850/F1KH-D8, only PE0 is supported, so PEn cannot be selected by peripheral functions such as DMA or interrupts.



Changing the interrupt handler name, Generate Entity and Generate Enable/Disable 4.6.3 Function setting

From Smart Configurator for RH850 V1.10.0, User-defined interrupt handler is supported for all RH850 devices supported by Smart Configurator. User can define the interrupt handler for each interrupt in [Interrupt] page and decide if to generate the interrupt handler entity and interrupt enable/disable function.

- 1) User can input his own interrupt handler name by editing column [interrupt Handler] manually. "eiintn" is displayed on column [Interrupt Handler] as default interrupt handler. Users can edit this column and enter a user-defined name (except the default name "eiintn") here according to below basic rule:

 - Only characters 'a'~'z', 'A'~'Z', '0'~'9' or '_' can be inputted. The interrupt handler name starting with a number can't be inputted.
 - The interrupt handler can't be empty
 - The reserved interrupt handler name "eiintn" except for eiintn(n=current interrupt number) can't be inputted.
 - The any two same interrupt handler names can't be inputted.

Note: interrupt handler which is used by components is non-editable.

2) User can specify whether user-defined interrupt handler entity is generated by checking/unchecking [Generated Entity].

The default setting is always checked. When you change the setting to unchecked, the interrupt handler code won't be generated by Smart Configurator, then user can use his own handler code.

3) Users can specify whether the interrupt enable/disable function is generated by checking/unchecking [Generate Enable/Disable Function].

The default setting is unchecked. When user changes the setting to checked, a pair of interrupt enable/disable functions will be provided in "r_smc_interrupt.c" file. User can easily use interrupt by calling these APIs directly.

For code samples of the interrupt enable/disable functions, see

Figure 4-41 Interrupt Enable/Disable Functions Code Example.

Window Help	P										
<u> </u>											10
iic.scfg ×											
errupt config	guration									Generate	Code Generate F
terrupt vectors											
										Vector Number	
Vector Number		Interrupt	Interrupt request source	Peripheral	Priority	Status	OS management	Interrupt Handler	Generate Entity	Generate Enable/Disable Function	
550	1226H	INTDFE1CND10	CH0 condition match interrupt 1	DFE1	Lowest			eiint550	V		
551	1227H	INTDFE1CND11	CH1 condition match interrupt 1	DFE1	Lowest			eiint551	V	21 C	
552	1228H	INTDFE1CND12	CH2 condition match interrupt 1	DFE1	Lowest			eiint552	V		
553	1229H	INTDFE1CND13	CH3 condition match interrupt 1	DFE1	Lowest			eiint553	V		
554	122AH	INTDFE1SUBOUT0	Subtraction data interrupt 0	DFE1	Lowest			eiint554	V		
555	122BH	INTDFE1SUBOUT1	Subtraction data interrupt 1	DFE1	Lowest			eiint555	V		
556	122CH	INTDFE1SUBOUT2	Subtraction data interrupt 2	DFE1	Lowest			eiint556	V		
557	122DH	INTDFE1FEND0	CH0 Filter Processing End Interrupt Request	DFE1	Lowest			eiint557	V		
558	122EH	INTDFE1FEND1	CH1 Filter Processing End Interrupt Request	DFE1	Lowest			eiint558	V		
559	122FH	INTDFE1FEND2	CH2 Filter Processing End Interrupt Request	DFE1	Lowest			eiint559	V		
560	1230H	INTDFE1FEND3	CH3 Filter Processing End Interrupt Request	DFE1	Lowest			eiint560	V		
561	1231H	INTDFEFIFOOUTA	Buffer-A capture finished interrupt	DFE	Lowest			eiint561	V	8	
562	1232H	INTDFEFIFOOUTB	Buffer-B capture finished interrupt	DFE	Lowest			eiint562	V		
563	1233H	INTDFEFIFOERR	Error interrupt	DFE	Lowest			eiint563	V		
564	1234H	INTMSPI0TX0	MSPI0 Transmit status interrupt for channel 0		Lowest	Used		r_Config_MSPI00_channel01_interrupt_send	(2)	2 (2)	
565	1235H	INTMSPI0TX1	MSPI0 Transmit status interrupt for channel 1	MSPI0	Lowest			eiint565		(3)	
566	1236H	INTMSPI0TX2	MSPI0 Transmit status interrupt for channel 2	MSPI0	Lowest			eiint566	1		
567	1237H	INTMSPIORX0	MSPI0 Receive status interrupt for channel 0	MSP10	Lowest	Used		r_Config_MSPI00_channel00_interrupt_receive	1	2	
568	1238H	INTMSPIORX1	MSPI0 Receive status interrupt for channel 1	MSPI0	Lowest			eiint568	V	E.	
569	1239H	INTMSPIORX2	MSPI0 Receive status interrupt for channel 2	MSPI0	Lowest			eiint569	V		
570	123AH	INTMSPI1TX0	MSPI1 Transmit status interrupt for channel 0	MSPI1	Lowest			eiint570	V	111 I	
571	123BH	INTMSPI1TX1	MSPI1 Transmit status interrupt for channel 1	MSPI1	Lowest			eiint571	V		
572	123CH	INTMSPI1TX2	MSPI1 Transmit status interrupt for channel 2	MSPI1	Lowest			eiint572	V		
573	123DH	INTMSPI1RX0	MSPI1 Receive status interrupt for channel 0	MSPI1	Lowest			eiint573	V		
574	123EH	INTMSPI1RX1	MSPI1 Receive status interrupt for channel 1	MSPI1	Lowest			eiint574	V		
575	123FH	INTMSPI1RX2	MSPI1 Receive status interrupt for channel 2	MSPI1	Lowest			eiint575	V		
> 576	1240H	INTMSPIOTX/INTRCAN			Lowest	Used		r_mspi0_interrupt_send	V	E	
577	1241H	INTMSPIORX/INTRCA			Lowest	Used		r_mspi0_interrupt_receive	1		
> 578	1242H	INTMSPIOFE/INTRCAN			Lowest	Used		r_mspi0_interrupt_frameend	V	E	
579	1243H	INTMSPI0ERR/INTRCA			Lowest	Used		r_mspi0_interrupt_error	1		
> 580	1244H	INTMSPI1TX/INTRCAN			Lowest			eiint580	V	E	
581	1245H	INTMSPI1RX/INTRCA			Lowest			eiint581	V		
582	1246H	INTMSPI1FE/INTRCAN			Lowest			eiint582	V	100 m	
583	1247H	INTMSPI1ERR/INTRCA			Lowest			eiint583	V		
584	1248H	INTMSPI2TX/INTRCAN			Lowest			eiint584	V		
585	1249H	INTMSPI2RX/INTRCA			Lowest			eiint585	V	E	
586	124AH	INTMSPI2FE/INTRCAN			Lowest			eiint586	V		
> 587	124BH	INTMSPI2ERR/INTRCA			Lowest		E1	eiint587	1	19	

Figure 4-40 Interrupt Handler and Generate Entity settings



🔚 r_smc_interrupt.c 🛛



Figure 4-41 Interrupt Enable/Disable Functions Code Example



5. Managing Conflicts

When adding a component or configuring a pin or interrupt, problems in terms of resource conflict and missing dependency modules might occur. This information will be displayed in the Configuration Problems view. You can refer to the displayed information to fix the conflict issues.

5.1 Resource Conflicts

When two software components are configured to use the same resource (for e.g., DMAC00), an error mark (B) will be displayed in the Components tree.

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

ile Window Hel	р			
s 🖻 🗐				: 🗈
Smart_Configura	tor_Example.scfg ×			
Software com	ponent configura	tion	Ger	nerate Code Generate Rep
Components	èn ⊿ lª₂ 🕞 🕀 🚔 ▼	Configure		
	10 T	DMAC00		
type filter text				
 ✓ ₱ Drivers ✓ ● Commur ✓ Confid 		Chain setting Chain transfer Channel to chain	Disabled ~	
✓	g_DMAC00	Trigger source setting Trigger source	Software trigger	Valid trigger 1
 ✓ Safety Fu ✓ Config ✓ ⇒ Timers 	g_DMAC01 unction g_DCRA0 g_TAUB0_0 ~	Transfer mode setting Transfer mode	Single transfer	Valid trigger
verview Board C	locks Components Pin	Interrupt		
Console 🚨 Confi errors, 0 warning	guration Problems ×			78
Description	ems)	^		Туре
		by INTDMA0 in Config_DMAC00 conflicts with v	ector used by INTDMA0 in Config_DMAC01.	Interrupt
		by INTDMA0 in Config_DMAC01 conflicts with v	ector used by INTDMA0 in Config_DMAC00.	Interrupt
Peripheral (2)	•			
	•	used by Config_DMAC00 is already used by Confi		Peripheral
© E0401000	1: Peripheral DMAC00	used by Config_DMAC01 is already used by Confi	Ig_DMACUU.	Peripheral

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.

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

🚯 Smart Configurator							- 🗆	\times
File Window Help								
								8 📓
Smart_Configurator_Example.scfg ×								- 0
Pin configuration						Generate G	Code Generat	🗅 te Report
Hardware Resource	E 🗆 🛃 🦓	Pin Function					- 2 I ■ I	14 in 14
Type filter text		type filter text (* :	= any string, ? = any character)				All	~
Interrupt controller unit Reset Controller Power Supply Low-Power Sampler External Memory Access Controller	^	CSIGORYO	Assignment Not assigned Not assigned P10_7/TAUD015/TAUD0015/CSI Not assigned	Pin Number Not assigned Not assigned C11 (2)	Direction None None IO	Remarks	Comments	*
✓ I Serial Flash Memory Interface A			P10_6/TAUD0I13/TAUD0O13/CSI		0	Multiple pin functions on the same pin	٦	
	- 1	∠_CSIG0SSI	P10_4/TAUD0I9/TAUD009/RLIN2	7 E3	I			
 CSIG1 CSIG2 CSIG3 								
 ✓ N Clocked Serial Interface H ● CSIH0 								
● CSIH1	~							
Pin Function Pin Number								
Overview Board Clocks Components Pins In	terrupt							
Console Configuration Problems ×								78 - 0
4 errors, 0 warnings, 0 others								
Description			^				Type	
 Pin (4 items) 								
						FAUD0_13, pin used by TAUD0O13 in Conf		
						13 in Pin Allocator, pin used by CSIG0SO in		
					AUD0I13 in C	Config_TAUD0_13, pin used by CSIG0SO in		
E05000010: Pin A12 cannot be used	multiple time	es. Pin A12 is assigne	ed to TAUDUI13, TAUD0O13 and CSIC	050.			Pin	
4								>

Figure 5-2 Pin Conflicts

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

	🔇 Smart Configurator				-		\times
Smart_Configurator_Example.scfg × Pin configuration Hardware Resource I generate Code Generate Repor Hardware Resource I generate Code Generate Repor I type filter text I type filter text (* = any string, ? = any character) I generate Controller I type filter text I type filter text (* = any string, ? = any character) I generate Controller I controller I controller I controller I controller I controller I controller I controller I controller I con	File Window Help						
Pin configuration Generate Code Generate Report Hardware Resource Pin Function Type filter text Itype filter text (* = any string, ? = any character) All Image: Second Secon							
Pin configuration Generate Code Generate Report Hardware Resource Image: Second S	Smart_Configurator_Example.sc	.fg ×					° E
Type filter text type filter text (* = any string, ? = any character) All Interrupt controller un Reset Controller Power Supply Low-Power Sampler External Memory Acce SFMA0 CSIGOSC / P10_7/TAUDOI15/TAUDO015/CSIC / C11 (2) IO CSIGOSC / P10_6/TAUDO13/TAUD0013/CSIC / A12 (2) CSIGOSSI / P10_4/TAUD019/TAUD009/RLIN2 / E3 I CSIG2 Resolve conflict CSIG2 Resolve conflict Sessing all CSIG2 Sessing all CSIG2 Sessing all 	Pin configuration					Generate Rep	ort
Interrupt controller un Image: Function Assignment Pin Number Direction Reset Controller CSIGORYI Not assigned Not assigned None Power Supply CSIGORYO Not assigned Not assigned None External Memory Acce CSIGOSC P10_7/TAUD0115/TAUD0015/CSIC C11 (2) IO Serial Flash Memory In CSIGOSO P10_6/TAUD013/TAUD0013/CSIC A12 (2) O SFMA0 CSIGOSSI P10_4/TAUD019/TAUD009/RLIN2 E3 I CSIG0 Assign all Unassign all I I CSIG2 Resolve conflict S S S	Hardware Resource 🗉 🖻 💐 🎎	Pin Fu	nction			3 🗉 🖬 🔤	4
Image: Controller Image: Controller Image: Controller Image: Controller Image: Controller Image: Controller Image: Controller Image: Controller Image: Controller Image: Controller Image: Controller Image: Controller Image: Controller Image: Controller Image: Controller Image: Controller Image: Controler Image: Contr	Type filter text	type f	filter text (* =	any string, ? = any character)	All	,	~
CSIG1 Unassign all CSIG2 Resolve conflict CSIG2 Solve conflict	 Reset Controller Power Supply Low-Power Sampler External Memory Acce Serial Flash Memory In SFMA0 		CSIGORYI CSIGORYO CSIGOSC CSIGOSI & CSIGOSO	 Not assigned Not assigned P10_7/TAUD0I15/TAUD0015/CSIC Not assigned P10_6/TAUD0I13/TAUD0013/CSIC 	 Not assigned Not assigned C11 (2) Not assigned A12 (2) 	None None IO None	
	CSIG1 CSIG2 CSIG2 CSIG2 CSIG2 CSIG2				>		
		ents Pins	Interrupt				

Figure 5-3 Resolving Pin Conflicts

The pins of the selected node will be re-assigned to other pins.



6. Generating Source Code

6.1 Registering Generated Source Code with CS+

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

Smart Configurator				_		\times
File Window Help						
						i 🗈 🜋
Smart_Configurator_Example	.scfg ×					- 0
Software component co	onfiguration		Generate	Code Gen	ierate F	leport
Components 🔌 🗳 🔩 🕀 🖶	Configure					^
÷ •	Clock setting					
	Operation clock	СКО	\sim			
	Clock source	PCLK/32768	\sim	(Actual v	value: 0).122
Config_CSIG0	Input setting	as channel () innut				> ~
Overview Board Clocks Compo	onents 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 your 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 - [Property]	oject Tree]						-		\times
File Edit View Project Build Debug Tool Win	dow Help							🧔 (7 🗟
- 🎊 Start 🛄 🔛 🍠 🐰 🐚 🖄 🕫 🔍		▼ 100%	🚽 🚮 🚮 Default	Build	- 🔨 🖓	L, h 🗐 🖗	D 1 🖌 🕢	a ça ès	i A
······································	🔽 Solution List 🤅 🖓 🗄 🚝	1 82							
Project Tree T ×									
	Solution L	ist							▼ ×
	R7F701651 Property							م	- +
Smart Configurator Example (Project)*	 File Information File name 		DR7F701651.DVF						
Smart Configurator (Design Tool)	Absolute path		C:\Users\app\Desk	top\CS+E7.00.00	H\CS+E7.00.00H\0	CS+\CC\Device\R	H850\Devicefile\DF	R7F701651	DVF
GHS CCRH850 (Build Tool)	 Microcontroller Inform 	ation							
	Microcontroller name		R7F701651						
🖕 🎒 File	Nickname File version		RH850/F1KM V1.20						
Build tool generated files	Internal ROM size [KByt	es]	Code Flash:4096 D	ata Flash:128					
Smart Configurator	Internal RAM size [Bytes		Local RAM(CPU1):	262144 Global R	AM(BankA):98304 G	lobal RAM(BankB)	98304 Retention R	AM:65536	
Config_TAUB1	Reset vector address		HEX 0						_
Config_TAUB1.c	 IOR Display Type 		a						
Config_TAUB1.n	IOR display type Notes		Structured Naming						
	/ Notes								
general									
Pin.c	File name								
Pin.h	This is the file name of this of	device file.							
	Microcontroller Informa	ation							-
	Output								џх
	Warning(W2500001) :		urator for RH850 ex	ecutable(Sma	artConfigurato	r.exe) path i	s not set cor	rectly, p	ple 🔺
	ase set it in proper [EOF]	ty page.							
									~
	All Messages *Rapid	Build /							-
< >	Output 🌸 Smart Brows	er 🔠 Error List							
	-4 F5	FG		8	F9	FID	FTI	FR2	
								DISCONNE	ст

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



6.2 Loading files generated by All Toolchain (CC-RH, GHS, IAR)

When start the Smart Configurator standalone and select All Toolchain (CC-RH, GHS, IAR) as toolchain, Smart Configurator outputs the source files that can adapt to all three toolchains: CC-RH, IAR and GHS. When loading these generated files in CS+ project, please follow below procedures:

- (1) Create a new project according to Project Wizard in CS+ project.
- (2) Remove default file "main0.c" or "main.c" created by CS+ from CS+ project.



Figure 6-3 Remove main0.c

Note: "r_cg_main.c" generated by Smart Configurator generate is used as the entry function.

(3) Add include path of "iodefine.h" by [Compile Options].

🕼 u2b - CS+ for CC - [Property]		- @ ×
File Edit View Project Build Debug Tool Window Help	 ★ ★	÷ 😵 🦁
🐺 Project Tree 🕴 🗶 🚰 Property 📝 r_cg_main.c 🗹 r_smc_entry.h 🧭 Solution List 🗹 iodefine	s.h	+ x
Image: Construction of the second	Path Edit × Path(One path per one line):	folder as the reference point.
Common Options Compile Options Assemble Options Link Optio Output Information (M0291001) : iodefine. h was generated. J BOT All Messages Rapid Build	Placeholder Value ActiveProjectDir C:\MyWork\test\u2b ActiveProjectNicomName R77/0254x_P ActiveProjectName u2b BuildModeName DefaultBuild MainProjectDir C:\MyWork\test\u2b OK Cancel	
FI Open Help for. F2 Rename F3 Find Next F4 Replace Next F5 Go F5 Build & DownL F7 Build Proje	ect F8 Ignore Break a. F9 Set/Delete Bre. F8 Step Over F71 Step	p In FI2 Jump to Functio

Figure 6-4 Add include path of "iodefine.h"



(4) Manually add the source files output by the Smart Configurator to the project, such as below figure shows:



Figure 6-5. Add new files to CS+ project

Note: Because CS+ Smart Configurator communication plug-in does not support All Toolchain (CC-RH, GHS, IAR), it is necessary for the user to add or delete source files after configuration change in the Smart Configurator.



6.3 Configuration of Generated Files and File Names

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

"ConfigName" indicates the name of the configuration formed by the component settings.







Folder	File	Description
general	-	This folder is always generated.
3		It contains header files and source files commonly
		used by Code Generator drivers of the same
		peripheral function.
	r_cg_xxx.h ^(Note*1)	These files are only generated for the used
		components.
		The files contain macro definitions for setting SFR
		registers.
	r_cg_cgc.c	This file is always generated.
		It contains the initialization of clock sources in
		accordance with the settings in the [Clocks] page.
	r_cg_cgc.h	This file is always generated.
		This header file contains macro definitions to
		initialize clocks.
	r_cg_cgc_user.c	This file contains functions to be added to
		R_CGC_Create after the CGC initialization.
		User can add codes and functions in the dedicated
		user code areas.
	r_cg_intvector.c	r_cg_intvector.c is generated only for:
	r_cg_intvector_PEn.c	Smart Configurator for RH850/F1KM
		Smart Configurator for RH850/F1KH
		r_cg_intvector_PEn.c is generated only for:
		Smart Configurator for RH850/U2A(only
		PEn(n=0~3) which are chosen to be used, the
		r_cg_intvector_PE <i>n</i> .c(n=0~3) are generated.)
		Smart Configurator for
		RH850/C1M(r_cg_intvector_PE1.c is generated.)
		Smart Configurator for
		RH850/U2B(r_cg_intvector_PE0.c is generated.
		It contains interrupt vector table definitions.)
	r_cg_macrodriver.h	This file is always generated.
		This header file contains common macro
		definitions used in drivers.
	r_cg_main.c	This file is always generated. It defines the main()
		function.
	<i>r_cg_</i> systeminit. <i>c</i>	This file is always generated.
		It contains <i>R_Systeminit</i> that calls all driver
		initialization functions with the name
		R_ConfigName_Create.
		<i>R_Systeminit</i> also calls the functions for initializing
	r og uppredefine h	clocks.
	r_cg_userdefine.h	This file is always generated.
		User can add macro definitions in the dedicated
	r omo interret e	user code areas.
	r_smc_interrupt.c	This file is always generated.
	r_smc_interrupt.h	This file is always generated. It contains the priority
		level definition of all interrupts that are configured
		in the [Interrupts] tabbed page. User can use these
		macro definitions in application codes.



	r_smc_entry.h	This file is always generated.
		It contains the "include" clause which include:
		"r_cg_xxx_common.h"
		"r_cg_macrodriver.h"
		"r_cg_userdefine.h"
		•
		"r_cg_cgc.h"
		{ConfigName}.h
		This file is included by file "r_cg_main.c".
	r_cg_intc_PEn.c	This file is generated only for:
		RH850/U2A:
		only when PE PE <i>n</i> (n=0~3) is chosen to be used,
		the r_cg_intc_PE <i>n</i> .c(n=0~3) is generated.
		RH850/C1M: r_cg_intc_PE1.c is generated.
		RH850/U2B: r_cg_intc_PE0.c is generated.
		This file contains interrupt initialization API definitions.
	r_cg_xxx_common.c ^(Note*1)	This file is generated only for components which
		have some common settings shared by all
		resources of the component. Normally, it contains
		the shared API for multiple configurations and will
		be called by users.
	r_cg_xxx_common.h ^(Note*1)	This is header file for r_cg_xxx_common.c and
		r_cg_xxx_common_user.c.
		It is generated only for components which have
		some common settings shared by all resources of
		the component. Normally, it contains the shared
		API declaration for multiple configurations.
	r_cg_xxx_common_user.c ^(Note*1)	This file is generated only for components which
		have some common settings shared by all
		resources of the component. Normally, it contains
		the interrupt service routines for interrupts which
		are shared by multiple configurations.
		User can add codes and functions in the dedicated
		user code areas.
	r_smc_clock_info.h	This file is generated only for Smart Configurator
		for RH850/U2B.
		It contains macro definition for the clock source
		and module clock setting from [Clock] page.
		User can use the clock setting macro by including
		this file.
r_pincfg	Pin.c	This file is always generated.
		It is a reference of pin function initialization for all
		peripherals configured in the [Pins] tabbed page
		(except I/O Ports).
	Pin.h	This file is always generated.
		It contains the function prototypes of pin settings in
		<i>Pin.c,</i> Symbolic name definition, Symbolic name
		user guide and Symbolic name API.
{ConfigName}	-	This folder is generated for the Code Generator
		drivers that are added to the project.
		API functions in this folder are named after the
		ConfigName (configuration name).
L		



{ConfigName}.c	This file contains functions to initialize driver (<i>R_ConfigName_Create</i>) and perform operations that are driver-specific, for e.g. start (<i>R_ConfigName_Start</i>) and stop (<i>R_ConfigName_Stop</i>).
{ConfigName}_user.c	This file contains interrupt service routines and functions for user to add code after the driver initialization (<i>R_ConfigName_Create</i>). 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.

6.4 Initializing Clocks

Configurations of the clock sources in the [Clocks] page are generated to the macros in the r_cg_cgc.h file located in \src\smc_gen\general folder.



Figure 6-7 Clocks Configuration with Main Clock Selected as Clock Source

Folder	File	Macros/Functions	Description
general	r_cg_cgc.c	R_CGC_Create	This API function initializes clocks. <i>R_Systeminit</i> in <i>r_cg_systeminit.c</i> will call this function during execution of the <i>main()</i> function.
	r_cg_cgc.h	Macros related to clocks	These macros are for clock initialization in <i>R_CGC_Create</i> .
	r_cg_cgc_user.c	R_CGC_Create_UserInit	This API function is used to add code to <i>R_CGC_Create</i> after the CGC initialization.



6.5 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 this file *src**smc_gen*\{*ConfigName*}.*c*.

Pin initialization codes will be handled in main().

े 🗁 🖫 Smart_Configurator_Example.scfg	g ×			
in configuration			Generate Code	ے Generate Repo
Software Compon 🗉 🖻 🔩 😹	Pin Function			2 🗉 🖬 🔤 Z
	type filter text (* =	any string, ? = any character)	All	~
 A/D Converter Config_ADCA0 CSI Slave Config_CSIG0 Data CRC Config_DCRA0 Input Period Count Detec Config_TAUB0_0 PWM Output Config_TAUB1 	✓ TAUB101 ✓ TAUB102 ✓ TAUB103 □ TAUB103 □ TAUB104 □ TAUB105 □ TAUB106 □ TAUB106 □ TAUB107 □ TAUB107 □ TAUB108 □ TAUB109 □ TAUB1010 □ TAUB1011 □ TAUB1012 □ TAUB1013 □ TAUB1014	Assignment P11_3/CSIH2SC/CAN3RX/INTP3/P P11_12/RLIN25RX/PWGA52O/TAU P11_4/CSIH2SI/CAN3TX/INTP21/F Not assigned Not assigned	∥ H2	Direction O O O None None None None None None None None
< >	<			>

Figure 6-8 Pins Configuration for Config_TAUB1

Folder	File	Function	Description
{ConfigName}	{ConfigName}.c	R_ConfigName_Create	This API function initializes the pins used by this driver. <i>R_Systeminit</i> in <i>r_cg_systeminit.c</i> will call this function during execution of the <i>main()</i> function.



(2) Reference to pin initialization codes

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

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



6.6 Initializing Interrupts

Configurations in the [Interrupts] page are generated in some source files.



Figure 6-9 Interrupts Configuration in Interrupts View

RH850/C1M, F1KM, F1KH and U2B:

No	Item	Folder	File	Description
(1)	Priority	{ConfigNa me}	{ConfigName}.c	Interrupt priority level settings are initialized in <i>R_ConfigName_Create</i> in this file. <i>R_Systeminit</i> in <i>r_cg_systeminit.c</i> will call this function during execution of the <i>main()</i> function.
(2)	OS management	{ConfigNa me} or general	{ConfigName}_u ser.c Or r_cg_xxx_comm on_user.c	The interrupt functions defined in this file are output in the interrupt format that can be managed by the OS.
(3)	Interrupt Handler/Generat e Entity	general	r_smc_intprg.c r_cg_intvector_ PE0.c	The interrupt handler displayed on [Interrupt Handler] will be generated in file " <i>r_smc_intprg.c</i> " if [Generate Entity] is checked.
(4)	Generate Enable/Disable Function	general	r_smc_interrupt. c r_smc_interrupt. h	Interrupt enable/disable functions will be generated in <i>r_smc_interrupt.c</i> if [Generate Enable/Disable Function] is checked.

RH850/U2A:

No	Item	Folder	File	Description
(1)	Priority	general	r_cg_intc_PEn.c	Interrupt priority level settings are initialized in <i>R_Interrupt_Initialize_ForPE</i> in this file. <i>R_Systeminit</i> in <i>r_cg_systeminit.c</i> will call this function during execution of the <i>main()</i> function.
(2)	OS management	{ConfigNa me} or general	{ConfigName}_u ser.c or r_cg_xxx_comm on_user.c	The interrupt functions defined in this file are output in the interrupt format that can be managed by the OS.
(3)	Interrupt Handler/Generat e Entity	general	r_smc_intprg.c r_cg_intvector_ PE0.c	The interrupt handler displayed on [Interrupt Handler] will be generated in file " <i>r_smc_intprg.c</i> " if [Generate Entity] is checked.
(4)	Generate Enable/Disable Function	general	r_smc_interrupt. c r_smc_interrupt. h	Interrupt enable/disable functions will be generated in <i>r_smc_interrupt.c</i> if [Generate Enable/Disable] Function is checked.
(5)	PE <i>n</i> (the UI setting is only for RH850/U2A)	general	r_cg_intc_PEn.c	Interrupt binding is initialized in <i>R_Interrupt_Initialize_ForPE</i> in this file. <i>R_Systeminit</i> in <i>r_cg_systeminit.c</i> will call this function during execution of the <i>main()</i> function.



7. Creating User Programs

The Smart Configurator for RH850 only handles one component type: [Code Generator]. This chapter describes the method to add custom code for the Code Generator components.

7.1 Adding Custom Code in the Case of Code Generator

When creating configuration for Code Generator component, 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 "TAUB1" for the PWM output function (resource TAUB1). The comments to indicate where to add custom code are at the start and end of each of the three files. 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 examples are for TAUB1 (Config_TAUB1_user.c).

/**************************************
Pragma directive ************************************
/* Start user code for pragma. Do not edit comment generated here */ /* End user code. Do not edit comment generated here */
/**************************************
Includes ************************************
<pre>#include "r_cg_macrodriver.h"</pre>
#include "r_cg_userdefine.h"
<pre>#include "Config_TAUB1.h" /* Start user code for include. Do not edit comment generated here */</pre>
/* End user code. Do not edit comment generated here */
/**************************************
Global variables and functions
/* Start user code for global. Do not edit comment generated here */ /* End user code. Do not edit comment generated here */
/**************************************
* Function Name: R_Config_TAUB1_Create_UserInit
* Description : This function adds user code after initializing the TAUB1 channel * Arguments : None
* Return Value : None

<pre>void R_Config_TAUB1_Create_UserInit(void)</pre>
{ /* 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_TAUB1_channel0_interrupt
* Description : This function is TAUB10 interrupt service routine
* Arguments : None
* Return Value : None

#pragma interrupt r_Config_TAUB1_channel0_interrupt(enable=false, channel=256, fpu=true,
callt=false)
void r_Config_TAUB1_channel0_interrupt(void)
/* Start user code for r_Config_TAUB1_channel0_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 */



8. Backing up Generated Source Code

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

The Smart Configurator generates a backup folder for the previously generated source code when new code is generated by clicking on the [Generate Code)] button. <Date-and-Time> indicates the date and time when the backup folder is created after code generation.

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



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

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

Two selections of output files are available (PDF, Text).

👩 Smart Configurator	– 🗆 X
File Window Help	
Smart_Configurator_Example.scfg ×	
Overview information	Generate Code Generate Report
Get an <u>overview</u> of the features provided by Smart Configurator.	
Videos Introduction to Smart Configurator Browse related videos	Application Code Software Components Middleware & Drivers Device Drivers
What's New Check out <u>what's new</u> in the latest release. See all <u>Release Notes</u> .	MCU Hardware
Product Documentation User's Guide API manual	
Current Configuration	
Selected board/device: R7F701653 (ROM size: 4 MB, RAM size	: 256 KB, Pin count: 272)
Generated location (PROJECT_LOC\): src\smc_gen	Edit
Selected components:	
Component ^ Version Configuration	n v
Overview Board Clocks Components Pins Interrupt	
	Total Sconfiguration Problems × Image: Sconfiguration Problems Image: Sconfiguration Problems
Smart Configurator Output	0 items
M00000004: File removed:src\smc_gen\general\r_smc_ M00000002: Code generation is successful: <u>C:\Users\</u>	
<	> <>

Figure 9-1 Output of a Report on the Configuration



💰 Smart Report			\times	
Generate report of configurations				
Options				
Print all sections				
O Print specific sections				
Board			~	
Clocks				
➤ Components				
> Pins				
☐ Interrupt			~	
✓ Output as PDF		Select	Font	
Output as text				
C:\MyWork\work\runtime-smc.rh850.product\output		Brow	se	
ОК		Cance	1	

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

e Window Help			
🔁 🗒			
Smart_Configurator_Example.scfg	×		
n configuration		😼 Generate Code	e Generate Rep
oftware Compon 🗉 🖻 峰 👪	Pin Function		2 E E 24
	type filter text (* = any string, ? = any character)	A	I
 A/D Converter Config ADCA0 		Pin Number	Direction O
🗸 🛓 CSI Slave	TAUB1O2 / P11_12/RLIN25RX/PWGA52O/TAU	/ H2	0
	✓ TAUB103 / P11_4/CSIH2SI/CAN3TX/INTP21/F ✓ TAUB104 / P12 5/PWGA700/ETNB0MDC/CSI		0
Config_DCRA0	TAUB105 / P11_5/CAN5RX/INTP5/RLIN33TX/		0
 Input Period Count Detec Config TAUB0 0 	 TAUB106 / P12_3/RLIN27RX/PWGA680/CSIG TAUB107 / P11_6/RLIN33RX/INTP13/CAN5T> 		0
🗸 🛓 PWM Output	TAUB108 / P11_15/CAN2RX/INTP2/CSIH2CSS	• A5	0
Config_TAUB1	 TAUB109 / P11_7/INTP5/PWGA320/CSIH3SC TAUB1010 / P12_0/CAN2TX/PWGA560/TAUB1 		0
	TAUB1011 / P11_8/_CSIG1SSI/RLIN35TX/PWG/	/ E2	0
	TAUB1012 / P12_1/RLIN34RX/INTP14/CSIH2CS	 B5 Not assigned 	O None
	TAUB1014 / Not assigned	Not assigned	None
	TAUB1015 / Not assigned	 Not assigned 	None
<	\$		
n Function Pin Number			

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

10. User code protection feature

The Smart Configurator for RH850V1.9.0 and the later version incorporates the enhanced user code protection feature for Smart Configurator Code Generation component; from the Smart Configurator for RH850V1.10.0, the user code protection feature is extended to Clock generated file and Interrupt generated files. 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.

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 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 10-2 Outputting a Figure of MCU Package (in png Format)



10.3 Examples of using user code protection feature to add new user code

Figure 10-3 shows an example of adding new user code into the Create API of PWM Output module by using the specific tags shown in Figure 10-1. After updating the configuration in the PWM output GUI and regenerating the codes, the inserted user codes will be automatically merged into the newly generated file.



Figure 10-3 User code protection with auto merge



10.4 What to do when merge conflict occurs

10.4.1 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 10-4 The merge conflict message outputted in the Smart Configurator console.

🖻 Console ×	•	8
Smart Configurator Output		
M00000002: Code generation is successful: <u>C:\MyWork\work\runtime-smc.rh850.product\src\smc_gen</u>		
M00000001: Code generation is started		
M04000001: File generated:src\smc_gen\Config_TAUB1\Config_TAUB1.c		
M05000012: File generated:src\smc_gen\r_pincfg\Pin.h		
M05000012: File generated:src\smc_gen\r_pincfg\Pin.c		
M00000005: The above files highlighted in red color have user code merge conflicts, please open the file and resolve the conflict manually		
M00000002: Code generation is successful:C:\MyWork\vork\runtime-smc.rh850.product\src\smc gen		
	~	
	>	

Figure 10-4 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 10.4.2, Steps for resolving the merge conflict described.



10.4.2 Steps for resolving the merge conflict

- To resolve this 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 10-5 Code before resolving conflict).
- 2) Click on "Copy Current Change from Left to Right" (Figure 10-5 Code before resolving conflict).
- Delete the codes that you do not want to use (Figure 10-6 Code after applying "Copy Current Change from Left to Right").
- 4) Save the modified code (Figure 10-7 Code after deleting and saving).



Figure 10-5 Code before resolving conflict



Figure 10-6 Code after applying "Copy Current Change from Left to Right"



RH850 Smart Configurator

User's Guide: CS+



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



11. Help

11.1 Help

Refer to the help system for detailed information on the Smart Configurator.



Figure 11-1 Help Menu



Figure 11-2 Smart Configurator Quick Start information



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+ V8.11.00 Integrated Development Environment User's Manual: Project Operation (R20UT5199)

CS+ V8.11.00 Integrated Development Environment User's Manual: RH850 Debug Tool (R20UT5202)

CS+ V8.11.00 Integrated Development Environment User's Manual: Message (R20UT5200)

CC-RH Compiler User's Manual (R20UT3516)

(Obtain the latest version from the Renesas Electronics website.)



Website and Support

Renesas Electronics Website <u>http://www.renesas.com/</u>

Inquiries

http://www.renesas.com/contact/

All trademarks and registered trademarks are the property of their respective owners.



Revision History

Rev.	Section	Description
1.00	All	First edition issued
1.10	Introduction	Update to CS+ (CS+ for CC) V8.10.00, RH850 Smart Configurator V1.9.0 and CS+ RH850 Smart Configurator Communication Plugins V1.10.00.
	All	All figures updated.
	2.5 Preparing Sample Projects	32 bit Environment, sample project path deleted.
		64 bit environment, 4 new sample project path added.
	3.4.4 MCU/MPU Package view updated	Description and Figure updated.
	4.1.2 Selecting the board updated	Description and Figure updated.
	4.2 Clock Settings	Description and Figure updated.
	4.3 System Settings (only for RH850/U2A)	New added.
	4.4.3 Removing software component	Add description and figure about removing multiple components.
	4.4.6 Configure general setting of component	New added.
	4.5.3 Show pin number from pin functions	New added.
	4.5.6 Pin setting using board pin configuration information	New added.
	4.5.7 Pin filter feature	New added.
	4.5.8 Pin Errors/Warnings setting	New added.
	4.6.2 Changing the PEn setting	New added.
	4.6.3 Changing the interrupt handler name and Generate Entity setting new added	New added.
	5.2 Resolving pin conflicts	Moved to chapter 5. Managing Conflicts from chapter 4.5.2 Resolving pin conflicts.
	6.2 Configuration of Generated	Figure 6-3 Configuration of Generated Files and File
	Files and File Names	Names and relative table content updated.
	6.5 Initializing Interrupts	RH850/U2A, RH850/U2B, RH850/C1M, RH850/F1KH interrupt description added.
	9.1 Report on All Configurations	PDF format file report added.
	10. User code protection feature for Smart Configurator Code Generation component	New added
	12. Documents for Reference updated	User manual is updated to the latest.
1.20	Introduction	Update to CS+ (CS+ for CC) V8.11.00, RH850 Smart Configurator V1.10.0 and CS+ RH850 Smart Configurator Communication Plugins V1.11.00.
	3.4 Window	Update:
	9.1 Report on Configuration	Figure 3-3 Main Window
	11. Help	Figure 3-4 Smart Configurator View
		Figure 9-1 Output of a Report on the Configuration Figure 11-2 Smart Configurator Quick Start information
	4.6.3 Changing the interrupt handler	Interrupt handler edit function and Generate Entity
	name, Generate Entity and	function are extended to all RH850 devices.
	Generate Enable/Disable Function setting	New function Generate Enable/Disable Function is added.
	6.5 Initializing Interrupts	Add interrupt handler and interrupt enable/disable function initialization.

	10. User code protection feature	User code protection feature is extended to Clock generated files and Interrupt generated files.
1.30	2.2 Installing the Smart Configurator	Add note for All Toolchain (CC-RH, GHS, IAR)
	4.6.3 Changing the Interrupt Handler Name, Generate Entity and Generate Enable/Disable Function Setting	Add detailed interrupt handler name rule
	6.2 Loading files generated by All Toolchain (CC-RH, GHS, IAR)	New added

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. 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.
- 2. Processing at Power-on
- The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.

- 3. Prohibition of Access to Reserved Addresses
- Access to reserved addresses is prohibited.
- The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
- 4. Clock Signals
- After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during
 program execution, wait until the target clock signal has 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. Moreover, 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.
- 5. Differences between Products
- Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.
- The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

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