

R20AN0535EJ0140

Rev.1.40

Apr 16, 2024

RX Smart Configurator

User's Guide: IAREW

Introduction

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

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

Target device and support compiler

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

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

Contents

Contents

1. C	Over	view	4
1.1	Pu	rpose	4
1.2	Fea	atures	4
1.3	So	ftware Components	4
2. lı	nsta	Illation and uninstallation	5
2.1	Ins	stalling the Smart Configurator	5
2.2	Un	installing the Smart Configurator	5
3. C	Oper	rating the Smart Configurator	6
3.1	Pro	ocedure for Operations	6
3.2	Sta	arting the Smart Configurator	7
3.3	Cre	eate and loading a configuration file	8
3.3	3.1	Downloading FIT modules	8
3.3	3.2	Creating a New Configuration File	8
3.3	3.3	Opening an Existing Configuration File	10
3.4	Wi	ndow	11
3.4	4.1	Main menu	12
3.4	4.2	Toolbar	12
3.4	4.3	Smart Configurator view	13
3.4	4.4	MCU/MPU Package view	14
3.4	4.5	Console view	15
3.4	4.6	Configuration Problems view	15
4. S	Setti	ng of Peripheral Modules	.16



4.	1 Bo	ard setting	.16
4	4.1.1	Selecting the device	.16
	4.1.2	Selecting the board	. 17
4	4.1.3	Import of board configuration	.18
4	4.1.4	Export of board configuration	.18
4.2	2 Clo	ock settings	.19
4.:	3 Sy	stem Settings	20
4.	4 So	ftware component settings	.21
4	4.4.1	Adding component	.21
	4.4.2	Removing a component	.23
	4.4.3	Switching between the component view and hardware view	.24
	4.4.4	Component configuration settings	.25
	4.4.5	Component resource change	.26
	4.4.6	Adding FIT drivers or middleware	.29
	4.4.7	Setting of the FIT software components	. 30
	4.4.8	Changing the version of the FIT software components	.31
	4.4.9	Solving the grey-out component	.33
	4.4.10	"i" mark on FIT modules icon	.34
	4.4.11	Configure Analog Front End component	.35
4	4.4.12	Configure Motor Component	. 37
4	4.4.13	Configure general setting of component	.41
	4.4.14	Export configuration of component	.44
4	4.4.15	Import configuration of component	.44
4.	5 Pir	settings	45
4	4.5.1	Assign pins to resources	46
4	4.5.2	Pin setting using MCU/MPU package	47
	4.5.3	Show pin number from pin functions	48
	4.5.4	Export pin settings	.49
	4.5.5	Import pin settings	.49
	4.5.6	Pin setting using board pin configuration information	50
	4.5.7	Pin filter feature	.50
	4.5.8	Pin Errors/Warnings setting	51
	4.5.9	Symbolic name setting	.52
4.	6 Inte	errupt settings	54
	4.6.1	Changing the interrupt priority level and fast interrupt setting	54
4	4.6.2	Changing the interrupt priority level and fast interrupt setting	55
4	4.6.3	Multiple interrupts setting	56
5.	Mana	aging Conflicts	58
5. ⁻		source conflicts	
5.2		solving pin conflicts	
	-		-



6.	Generating Source Code60
6.1	Generating Source Code File60
6.2	Configuration of Generated Files and File Names61
6.3	Initializing Clocks
6.4	Initializing Pins
6.5	Initializing Interrupts
6.6	Backing up Generated Source Code66
7.	Loading generated files in Integrated development environment
7.1	Adding Custom Code of FIT67
7.2	Loading in IAR Embedded Workbench68
7.3	Build IAR Project File71
8.	Creating User Programs72
8.1	Adding Custom Code in the Case of Code Generator72
8.2	Using Generated Code in user application74
9.	Generating Reports75
9.1	Report on Configuration75
9.2	Configuration of Pin Function List and Pin Number List (in csv Format)76
9.3	Image of MCU/MPU Package (in png Format)76
10.	User code protection feature for Smart Configurator Code Generation component
	1 Specific tags for the user code protection feature
	2 Examples of using user code protection feature to add new user code
	3 What to do when merge conflict occurs78
	0.3.1 What is Merge conflict
1	0.3.2 Steps for resolving the merge conflict79
11.	Help80
11.	1 Help
12.	Documents for Reference82
Web	site and Support83



1. Overview

1.1 Purpose

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

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

1.2 Features

The Smart Configurator is a utility for combining software to meet your needs. It handles the following three functions to support the embedding of drivers from Renesas in your systems: importing middleware in the form of Firmware Integration Technology (FIT) modules, generating driver code and making pin settings. Graphical presentation on Smart Configurator, for instance the timing waveform, makes your configuration of middleware and drivers easy.

1.3 Software Components

The Smart Configurator supports two types of software components: Code Generator (CG) and Firmware Integration Technology (FIT). Drivers and middleware supported by each software type are as follows.

- Basic drivers:
 - CG drivers (CMT, A/D Converter, SCI, etc.)
 - FIT modules (CMT, DTC, DMAC, RSPI, SCIFA, etc.)
- Middleware:
 - FIT modules (USB, Ethernet, Flash Memory (programming the on-chip flash memory), etc.)

The basic driver is a control program for peripheral functions of microcomputer such as CMT, A/D converter, SCI, etc. It is convenient to embed a software component (CG driver or FIT module) using code generation function.

In addition, FIT modules can be embedded for using middleware such as USB, Ethernet, and Flash memory (programming the on-chip flash memory) as software components.



2. Installation and uninstallation

This section describes the installation and uninstallation.

2.1 Installing the Smart Configurator

Download the Smart Configurator from the URL below.

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

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

2.2 Uninstalling the Smart Configurator

If you wish to uninstall the Smart Configurator, select "Smart Configurator for RX" from [Programs and Features] in the control panel.



3. Operating the Smart Configurator

3.1 **Procedure for Operations**

Figure 3-1 Operating Procedure, shows the procedure for generating a source file using Smart Configurator and loading it into IAR Embedded Workbench. For the operation of IAR Embedded Workbench, refer to relevant document of IAR.







3.2 Starting the Smart Configurator

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

Note: Please replace Vx.x.x with your version.





Figure 3-2 Starting of Smart Configurator



3.3 Create and loading a configuration file

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

3.3.1 Downloading FIT modules

The FIT drivers or middlewares are available from the web page of Renesas Electronics.

Download the files from the following address and unzip them.

https://www.renesas.com/fit

3.3.2 Creating a New Configuration File

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

- (1) In [Platform:], select the device.
- (2) In [Toolchain:], select [IAR EWRX Toolchain].
- (3) In [File name:], enter the file name.
- (4) Confirm [Location:]. If you want to change it, click [Browse] and select the save destination. Note: The *.eww, *.ewp, *.ewd, main.c and buildinfo.ipcf files will be generated to this location after clicking "Generate Code" button.

Category: RX	(0)
Platform:	Toolchain: (2)
ype filter text	Renesas RXC Toolchain
> Board	A IAR EWRX Toolchain
- Device	
~ RX110	
✓ RX110 - 36pin	
R5F51101AxLM	
R5F51103AxLM	
R5F5110HAxLM	
R5F5110JAxLM	
> RX110 - 40pin	
> RX110 - 48pin	
> RX110 - 64pin	
> RX111	
> RX113	
> RX130	
> RX13T	
> RX230	
> RX231	
> RX23E-A	
> RX23T	× .

Figure 3-3 Create a Configuration File



- (5) If you want to use FIT modules or middleware, click [Next].
 - a) Configure the language setting (C or C++) through wizard page and when C++ language is selected, main.cpp file will also be generated together with the IAR project file.
 - b) Configure the bank mode through the wizard page and corresponding Linear/Dual mode device will be automatically configured when loading the IAR project generated by Smart Configurator into IAR EW for RX.
 - c) Configure the RTOS settings.
 - d) Click [Browse] and set the path of "FITModules" directory which has been unzipped in chapter '3.3.1 Downloading FIT modules'.

Kew Smart	Configuration File				\times
Smart Config	uration Settings				
	guage, bank mode, FIT i for the new configuration				
- Language set	ting O C++	(a)			
Bank mode s Cinear mo	-	(b)			
-RTOS Setting:	5				
RTOS:	None				\sim
RTOS Version:					((
			<u>Manage F</u>	RTOS Versio	<u>ons</u>
- FIT module lo	ocation				
C:\FIT				Brows	se (d
L					
	< Back	Next > Fin	ish	Cance	4

Figure 3-4 Smart Configuration Settings

- (6) Click [Finish] to create the configuration file.
- (7) Add driver component, configure the setting, generate code, and save the project. Note: The *.eww, *.ewp, *.ewd and main.c files will be generated only for the first-time code generation, but the buildinfo.ipcf file will always be generated during the code generation.



3.3.3 Opening an Existing Configuration File

On the main window, click the [Opening an Existing Configuration File] button to display the [Open] dialog box. Select the file and click [Open].

ổ Open					×
← → * ↑	$ThisPC\rightarrowLo$	cal Disk (C:) > smartconfigurator > work	space v Ö	Search workspace	م
Organize 👻 New f	older				• 🔳 🕐
Pictures	^	Name	Date modified	Туре	Size
📑 Videos		Smart_Configurator_Example.scfg	10/15/2018 1:31 PM	SCFG File	1 KB
🏪 Local Disk (C:)					
💣 Network	- 11				
	~				
Fil	le <u>n</u> ame: Smar	t_Configurator_Example.scfg	~	Smart Configurati	on files \sim
				<u>O</u> pen	Cancel

Figure 3-5 Opening an Existing Configuration File



3.4 Window

The main window is displayed when the Smart Configurator is started. The configuration of the window is shown in Figure 3-6, Main Window.



Figure 3-6 Main Window

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



3.4.1 Main menu

File Window Help

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

Table 3-1 Main Menu Items

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

3.4.2 Toolbar



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

Table 3-2 Toolbar Buttons and Related Menu Items

Toolbar button	Related menu item
1	[File] \rightarrow [New Smart Configuration File]
	$[File] \rightarrow [Open]$
	$[File] \to [Save]$



3.4.3 Smart Configurator view

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



Figure 3-7 Smart Configurator View



3.4.4 MCU/MPU Package view

Display the MCU package. You can save rotation, enlargement, reduction, and MCU/MPU package view of the display to the image file. You can also confirmation pin assignment and change it.

Three types of package view can be switched between [Assigned Function], [Symbolic Name] and [Board Function]. [Assigned Function] displays the assignment status of the pin setting, and [Board Function] displays the initial pin setting information of the board. [Symbolic Name] displays the symbolic name information of the pins. The initial pin setting information of the board is the pin information of the board selected by [Board:] on the [Board] page.

(refer to "chapter 4.1.2 Selecting the board" and "chapter 4.5.6 Pin setting using board pin configuration information").



Figure 3-8 MCU/MPU Package View



3.4.5 Console view

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



Figure 3-9 Console View

3.4.6 Configuration Problems view

The Configuration Problems view displays problems with peripheral functions, interrupts, and pin conflicts.

O items Configuration Problems Discription		⊉ ▽ □ □
0 items		
Description	Туре	

Figure 3-10 Configuration Problems View



4. Setting of Peripheral Modules

You can select peripheral modules from the Smart Configurator view.

4.1 Board setting

On the [Board] page, you can select boards and change devices.

4.1.1 Selecting the device

Click on the ____] button to select a device.

Device selection Generate Code Generate Device selection Board: Custom User Board	ف te Repor
	in 14
Board: Custom User Board ~	Read Book
Device: R5F564MLCxFC	
Download more boards	

Figure 4-1 Selecting the Device

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

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



Figure 4-2 Confirm Device Change

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

Button	Operation explanation
Yes	Change to the selected device.
No	It does not change the device.
Save and continue	After saving the current configuration contents to the configuration file, change to the selected device.
Continue	Changes to the selected device without saving the current configuration contents to the configuration file.
Cancel	It does not change the device.



4.1.2 Selecting the board

Click on the [] to select a board from the list. When peripheral functions are configured by board selection, pins are automatically set according to board connection.

Sinart_Co	onfigurator_Example.scfg ×	- 0
Device	selection	😼 🤷 Generate Code Generate Report
Device se	election	<u>ک</u> ک
Board:	Custom User Board 🛛 🗸 🗸	
Device:	R5F564MLCxFC	
	Download more boards	

Figure 4-3 Selecting the Board

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

- Pin assignment
- Frequency of the main clock
- Frequency of the sub-clock
- Target device

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

Cont	firm board change	\times				
?	Changing the board will refresh all pin assignments. Assigned pins that are unavailable in the selected board may be removed.					
Do you want to continue?						
	Save and continue Continue Cancel					

Figure 4-4 Confirm Board Change

Table 4-2 Board Change Confirmation Operation List

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

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



4.1.3 Import of board configuration

The board setting is defined in bdf (Board Description File). Follow the procedure below to import board configuration.

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

Dimanc_C.	onfigurator_Example.scfg \times					- (
Device	selection			Generate Code	Generate	
Device s	election				(1)	èn 4
Board:	Custom User Board	\sim	(2)			
Device:	R5F564MLCxFC					
	Download more boards					

Figure 4-5 Import of Board Configuration (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.

4.1.4 Export of board configuration

The current main clock frequency, sub clock frequency and pin assignment settings can be exported as board configuration. Follow the procedure below to export the board configuration.

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

@ Smart_C	onfigurator_Example.scfg $ imes$				- 0
Device	selection			∎ te Code	Generate Report
Device s	election				20
Board:	Custom User Board	~			(1)
Device:	R5F564MLCxFC				
	Download more boards				
			5		
Overview	Board Clocks System Comp	onents Pins	Interrupts		

Figure 4-6 Export of Board Configuration (bdf format)



4.2 Clock settings

On the [Clocks] page, set the clock. The [Clocks] page setting is used as the clock source for each component. Set the clock before configuring the component.

The clocks setting is performed in the following procedure.

- (1) Set the clock oscillator circuit.
- (2) Sets the clock source to be supplied to the CPU and peripheral functions.
 - (a) When you move the mouse on the screen, the clock signal is displayed in blue.
 - (b) Click on the screen to select the clock selector.



Figure 4-7 Clock Settings



4.3 System Settings

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

There are 3 types of debug interface available: FINE, JTAG, JTAG (Trace)

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

Smart Configurator Output 0 tems 0 tems 1 Vice	#Smart_Configurator_Example.scfg ×	° П б М	CU/MPU Package ×			
Debug interface setting Unused FINE ITAG ITAG (frace) Note: The using of PC7/UB may have a limitation, because PC7/UB is controlled for mode-settings by emulator. Image: Control of the control of	System configuration		A F B B B Spr Correct	Assigned Function +		
Debug interface setting Unused FINE ITAG JTAG (frace) Note: The using of PC7/UB may have a limitation, because PC7/UB is controlled for mode-settings by emulator. Image: Control of C						
Debug interface setting Unused FINE ITAG JTAG (frace) Note: The using of PC7/UB may have a limitation, because PC7/UB is controlled for mode-settings by emulator. Image: Control of C	• On-chip debug setting		R5F5	64MLCxFC		
Note: The using of PC7/UB may have a limitation, because PC7/UB is controlled for mode-settings by emulator. Overview Board Clocks System Components Pins Interrupts © Console × Smart Configurator Output Messenger to TDO	Debug interface setting					
Note: The using of PC7/UB may have a limitation, because PC7/UB is controlled for mode-settings by emulator.	O Unused O FINE ● JTAG O JT	G (Trace)				
Overview Board Clocks System Components Pris Interrupts Image: Specific Configuration Problems ×	Note: The using of PC7/UB may have a limitation, because PC7/UB is co	strolled for mode-settings by emulator.				
© Console × Image: Configuration Problems × Image: Configuration Problems × Smart Configurator Output 0 items M650000021: Pin 35 is assigned to TDO Description			1001 1000 1000 1000 1000 1000 1000 100	ANAL ANA 4 43 444 444 444 444 444 444 444 444 4		424
© Console × Image: Configuration Problems × Image: Configuration Problems × Smart Configurator Output 0 items MS5000001: Pin 35 is assigned to TDO Description	Duaniau Board Clocks Sustam Components Diss Internate					
Smart Configurator Output 0 items 0 items 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1					¥ 1 ·	
M05000001: Pin 35 is assigned to TDO	In the second		- conduction to be a set of			
MMS500001: Pin 17 is assigned to TRST# MMS500002: Pin 18 is removed from FINED MMS5000001: Pin 34 is assigned to TUS MMS5000001: Pin 34 is assigned to TCK	M65000021 Pin 35 is assigned to TDO M050000021 Pin 17 is assigned to TRST# M050000022 Pin 18 is removed from FINED M050000021 Pin 30 is assigned to TMS M05000001 Pin 31 is assigned to TDI		Description	Туг	pe	
	6					

Figure 4-8 Debug Interface Setting at [System] Page



4.4 Software component settings

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

Smart_Configurator_Example.scfg ×		5	D
Software component conf	iguration		Generate Report
Components 🚵 🖬 🖓 🖻 🗃 🛸 🔻	Configure		0
ت ت			
type filter text			
Y 🕞 Startup	, Component tree		
 Generic r_bsp 			
t Losb			
Overview Board Clocks System Com	ponents Pins Interrupts		

Figure 4-9 Component Page

4.4.1 Adding component

The following describes the procedure for adding a component.

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

Smart_Configurator_Example.scfg ×		
oftware component configuration	Generate Code	Generate Repo
omponents 2 2 3 3 5 Configure		
type filter text		
 ✓ Startup ✓ Seneric ✓ r_bsp 		

Figure 4-10 Adding Components



- (2) Select a component from the list in the [Software Component Selection] page of the [New Component] dialog box (e.g. Single Scan Mode S12AD).
- (3) Click on [Next].

	omponent		— U)		
Software Component Selection						
Select co	mponent from those available	in list				
Category	All					
Function	All					
Filter						
Compon	ents	Short Name	Туре	١		
BSD Mo	de SDHI Driver	r_sdhi_rx	Firmware Integr	é		
	Sound Interface driver.	r_ssi_api_rx	Firmware Integr	í		
	IIC Driver.	r_sci_iic_rx	Firmware Integr	i		
	Scan Mode S12AD		Code Generator	i		
	Card Interface Mode		Code Generator	1		
	ck Synchronous Mode (3-wire	m	Code Generator	1		
SPI Op	eration Mode (4-wire method)		Code Generator	•		
<		an ata da series		>		
100000000000000000000000000000000000000	only latest version ems that have duplicated funct m	ionality				
	ware component provides Sing r where the arbitrarily selected			g		
Download	the latest FIT drivers and mide	dleware				
	general settings	(3)				

Figure 4-11 Selection of Software Components

- (4) Specify an appropriate configuration name in the [Add new configuration for selected component] page or use the default name (e.g. Config_S12AD0).
- (5) Select a hardware resource or use the default resource (e.g. S12AD0).
- (6) Click on [Finish]. The component is added to the component tree.

孩 New Compor	ent	—	
Add new config	uration for selected component		
Single Scan Mo Configuration r Resource:	de S12AD ame: (4) [config_S12AD0 (5) S12AD0		~
		(6)	
?	< Back Next >	Finish	Cancel

Figure 4-12 Add New Configuration for Selected Component (e.g. S12AD0)



4.4.2 Removing a component

Follow the procedure below to removing a software component.

- (1) Select a software component from the Components tree.
- (2) Click on the [(Remove component)] icon. The selected software is removed from the component tree. The selected software component will be removed from the Components tree.



Figure 4-13 Removing a Component

Multiple components can be selected by pressing [Ctrl] and clicking on components. Click on the [(Remove component)] icon. So multiple components can be removed at the same time.







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

Components	è 🕹 📩 🔁 🕀	₩ .	Configure
	(1)	٠	Show by Component View
type filter text			Show by Hardware View
			Note

Figure 4-15 Switch to [Show by Hardware View]

- (2) Double-click on a hardware resource node (e.g. S12AD1 under 12-bit A/D converter) to open the [New Component] dialog box.
- (3) Select a component from the list (e.g. Single Scan Mode S12AD) to add a new configuration as described in "chapter 4.4.1Adding component".



Figure 4-16 Adding CG Components form the Hardware View



4.4.4 Component configuration settings

Follow the procedure below to setting the component configuration.

- (1) Click the component in the component tree. (e.g. Config_S12AD0).
- (2) Configure the driver in the [Configure] panel to the right of the Components tree. The Figure 4-17 is an example.
 - a. Select AN000.
 - b. Select [A/D conversion start trigger pin] under [Conversion start trigger setting].
 - c. Click on [Advance setting] to expand the view.
 - d. Select [Discharge] for [Charge setting].

Smart_Configurator_Example.scfg	×		
C:\Users\a5146853\smartconfigurate	or\workspace\Smart_Confi	gurator_Example.scfg	Generate Code Generate Report
Components	2a245883≉ ▼ ***	Analog input mode setting Double trigger mode	^
type filter text Startup Startup Generic r_bsp A/D Converter A/D Converter Config_S12AD0	(2) a.] (2) b.	Analog input channel setting Analog input channel setting N000 AN001 AN002 AN005 AN006 AN007 Conversion start trigger setting Start trigger source And conversion start trigger pin Interrupt setting Enable AD conversion end interrupt (S12ADI) Priority Level 15 (highest)	
	(2) c.	Advance setting Advance setting Advance setting AN000 AN001 AN002 AN003 AN004 AN005 Self diagnosis setting Mode Unused Voltage used OV Disconnection detection assist setting Charge setting Another assist setting Dedicated cannel hold circuit channel setting	
Quantieur Board Clacks Sustem Cor			,

Figure 4-17 Component Configuration Settings

The code generation of the component is set to enabled by default.

Right click on the component and click [Generate code], it changes to [Generate code] and no code is generated.

Clicking [Generate code] will change to [Generate code] and generate code.



4.4.5 Component resource change

You can change the resource of the component (e.g. change from S12AD0 to S12AD1). Compatible configurations can be migrated from the current resource to the newly selected resource.

Follow the procedure below to change the resource.

- (1) Right-click on a component (e.g. Config_S12AD0).
- (2) Select [Change resource] from the context menu.



Figure 4-18 Resource Change

- (3) Select a new resource in the [Resource Selection] dialog box (e.g. S12AD1).
- (4) The [Next] button will be active; click on it.

Resourc	e Selection			—		×
Resource S						
Select reso	urce from the	ose available in th	e list			
Resource:	S12AD1 S12AD0					~
(3)	S12AD0 S12AD1					
		(4)				
	< Back	Next >	Finish		Cancel	

Figure 4-19 Select a New Resource

- (5) The configuration information is displayed on the [Configuration setting selection] page of the [Select Resource] dialog.
- (6) Check the portability of the settings.
- (7) Select whether to use the listed or default settings.



(8) Click on [Finish].

onfiguration setting list	(7)	
onfirm setting for resource chan	ge Use setting below Use default	t
letting (6)	Value	Portable
Double trigger mode	Unuse	Yes
AN000	Use	Yes
AN001	Unuse	Yes
AN002	Unuse	Yes
AN003	Unuse	Yes
AN004	Unuse	Yes
AN005	Unuse	Yes
AN006	Unuse	Yes
AN007	Unuse	Yes
Start trigger source	A/D conversion start trigg	er Yes
Fachie AD annualize and in		<u></u>
		,

Figure 4-20 Confirm New Resource Settings

The resource is automatically changed (e.g. changed from S12ADI0 to S12ADI1).

*Smart_Configurator_Example	early .						- 0
Software component	configuration						Generate Code Generate Report
Components	a 4 A B B 3 🔻	Configure					^
(¹	2.2	- Basic setting					
 Startup Generic 		Note When using the 12-bit	t A/D converter (unit 1), we	recommend not u	ising ports 02, 01, 00, port 9, port D, a	nd port E as output ports.	
✓ r_bsp		Analog input mode se	etting				
✓ In Drivers		Double trigger mo	de Extend analog in	iput mode			
 A/D Converter Config_\$12AD0 		Analog input channel setting AN100 AN101 AN102 AN103 AN104 AN105 AN106 AN107 AN108 AN109 AN110 AN111 AN112 AN113 AN114 AN115 AN111 AN112 AN113 AN114 AN115 AN116 AN117 AN118 AN119 AN120 Temperature sensor output Internal reference voltage Conversion start trigger setting Start trigger setting					
		A/D conversion start	10				
		Interrupt setting Enable AD convers	ion end interrup (S12ADI1)	Priority Leve	(15 (highest) ×		
		- Advance setting					
		Add/Average AD valu	e setting				
			N101 AN102 N106 AN107 N111 AN112	AN103 AN108 AN113	AN104		
Overview Board Clocks System	Comment Disclot	<					>

Figure 4-21 Resource Changed Automatically



To change the configuration name, follow the procedure below.

- (9) Right-click on the component.
- (10) Select [Rename] to rename the configuration (e.g. change Config_S12AD0 to Config_S12AD1).

Components	lª₂ 🕒 🕀 🛟 ▼				
	ت ت				
type filter text					
 ✓ ⇒ Startup ✓ ⇒ Generic > ⇒ Drivers ✓ ⇒ A/D Com 	verter				
	g_S12AD0				
V 🗁 Cor 🗸	Generate code				
S	Change resource	👩 Rename Confi	guration		Х
*	Remove				
	Duplicate	New name:	Config_S12AD1		
(10)	Rename		-		
	Reset to default				
+	Add Configuration >			OK Cano	el :

Figure 4-22 Renaming the Configuration



4.4.6 Adding FIT drivers or middleware

The following describes the procedure for adding a FIT driver or a middleware.

- (1) Click on the [to (Add component)] icon.
- (2) Select components from the list in the [Software Component Selection] page of the [New Component] dialog box (e.g. r_ether_rx and r_qspi_smstr_rx). Two or more components can be selected by clicking with the Ctrl key pressed.
- (3) Check that [Type] for the selected components is [FIT].
- (4) Click on [Finish].

💽 New	Component				×
Softwa	e Component Selection				-
Select o	omponent from those available in lis	st			-
Categor	All				,
Function	All				
Filter					
Compo	nents	Short Name	Type (3)	Version	
	nic Library with Graphical User Inter	r_emwin_rx	Firmware Integration Technology	6.32.a	
🗎 🖶 Ether	net Driver.	r_ether_rx	Firmware Integration Technology	1.23	
# Flash	API for RX100, RX200, RX600. and	. r_flash_rx	Firmware Integration Technology	5.10	
# Clock	Synchronous Control Module for	r_flash_spi	Firmware Integration Technology	3.30	
⊕FS10	15 Sensor Middleware	r_fs1015_rx	Firmware Integration Technology	1.00	
#FS20	12 Sensor Middleware	r_fs2012_rx	Firmware Integration Technology	1.03	
#FS30	00 Sensor Middleware	r_fs3000_rx	Firmware Integration Technology	1.00	
⊕HS30	0x Sensor Middleware	r_hs300x_rx	Firmware Integration Technology	1.23	
⊕ HS40	0x Sensor Middleware	r_hs400x_rx	Firmware Integration Technology	1.01	
# JPEG	Decoder for Renesas MCUs.	r_jpegd_rx	Firmware Integration Technology	2.06	
# JPEG	Encoder for Renesas MCUs.	r_jpege_rx	Firmware Integration Technology	1.01	
# Unsig	ned 32-bit circular buffer library.	r_longq	Firmware Integration Technology	2.00	
#Cont	ol Low Power States.	r_lpc_rx	Firmware Integration Technology	2.30	
# MEM	DRV Driver	r_memdrv_rx	Firmware Integration Technology	1.05	
# MMC	Mode MMCIF Driver	r_mmcif_rx	Firmware Integration Technology	1.10	
⊕OB12	03 Sensor Middleware	r_ob1203_rx	Firmware Integration Technology	1.01	
# PDC	driver	r_pdc_rx	Firmware Integration Technology	2.06	
⊕ PTP L	ight Driver	r_ptp_light_rx	Firmware Integration Technology	1.14	
⊕ PTP [)river	r_ptp_rx	Firmware Integration Technology	1.17	
⊕ QSPI	Clock Synchronous Single Master	r_qspi_smstr_rx	Firmware Integration Technology	1.21	
Shov	only latest version				
✓ Hide	items that have duplicated functiona	ality			
Descript	ion				
Depend	lencies : None				/
			on top of. It provides startup code, i		
Downlo	ad the latest FIT drivers and middlew	/are			
Configu	e general settings				
			(4)		
?	<	Back	Next > Finish	Cancel	

Figure 4-23 Adding FIT Modules



4.4.7 Setting of the FIT software components

The FIT drivers or middleware are available by setting configuration options.

The way of setting depends on each component.

- Set on the panel and settings will be generated to configuration file of FIT module automatically at each time of code generation action
- Set with modifying the configuration file of the FIT module

The configuration file is generated in the r_config folder after source code generation. See the "chapter 7.1 Adding Custom Code of FIT" to set the configuration options.

In addition, some components provide pin setting on the Configure panel. Followings are examples of pin setting on Configure panel.

oftware componer	nt configuration		Generate Code Generate Rep		
omponents	🎰 🕰 스 🖯 🕀 🕸 💌 Config	ure			
	8 T		Value		
		Y ETHERCO_MII	12		
* 😁 Startup		~ ETO_TX_CLK Pin	IV Used		
🛩 👄 Generic		~ ETO_RX_CLK Pin	IV Used		
💣 r_bsp		ETO_TX_EN Pin	IE Used		
 Drivers 		∽ ETO_ETXD3 Pin	18 Used		
Y Communications		Seto_ETXD2 Pin	IE Used		
r_ether_rx		∽ ET0_ETXD1 Pin	IR Used		
r_qspi_smstr_n		Setter Strategy Setter Strategy Setter S	IE Used		
👻 😂 Timers		ETO_TX_ER Pin	2 Used		
 r_cmt_rx 		> ETO_RX_DV Pin	IV Used		
🛩 🐸 Middleware		Seto_ERXD3 Pin	III Used		
🛩 🐸 Generic		← ETO_ERXD2 Pin	18 Used		
 r_byteq 		∽ ETO_ERXD1 Pin	III Used		
		∽ ETO_ERXD0 Pin	III Used		
		∞ ETO_RX_ER Pin	IV Used		
		≈ ETO_CRS Pin	III Used		
		> ET0_COL Pin	III Used		
		~ ETO_MDC Pin	12 Used		
		∽ ET0_MDIO Pin	20 Used		
		> ETO_LINKSTA Pin	IF Used		
		~ ETO_EXOUT Pin	ID Used		
		> ET0. WOL Pin	⊠ Used		

Figure 4-24 Pin setting of r_ether_rx

Components	àu ad 🐴 🕀 🗃 🕭 💌	Configure	
components			
type filter text		Property	Value
		 Configurations 	
✓ Startup		# Use FIT	V Used
👻 🍅 Generic		Use QSPI Channel 0	⊠ Used
dsd_1 💿		Enable debugging information	III Unused
 Drivers 		 SPTI0 interrupt priority level 	Level 10
 Communication 	6	 SPRI0 Interrupt priority level 	Level 10
a stherer		Y Resources	
💣 r_qspi_smstr_	rx	× © OSPI	
 Imers 		✓ ■ QSPI0	п
€ r_cmt_rx		► QSPCLK Pin	
 Middleware 		S QMO/QIQ0 Pin	
👻 😂 Generic		S GMI/DIOT Pro	
r_byteq		SQIO2 Pin	
		S GIO3 Pin	

Figure 4-25 Pin setting of r_qspi_smstr_rx



4.4.8 Changing the version of the FIT software components

Change the version of the FIT software component as follows.

(1) Right click the FIT software component in the component tree.

omponents	2.2	Configure			
64		Property		Value	
type filter text		Y Configura	ations		
🛩 😂 Startup		# Use FI	т	🗵 Used	
👻 😂 Generic		# Use Q	SPI Channel 0	🗷 Used	
💣 r_bsp		# Enable	debugging information	🖾 Unused	
🛩 🗁 Drivers		# SPTIO	Interrupt priority level	Level 10	
 Communications 		# SPRIO	Interrupt priority level	Level 10	
💣 r_ether_rx		✓	5		
💣 r_qspi_smstr_rx		11 B OCDI			
Y 🗁 Timers	Change version			E1	
<pre>% r_cmt_rx</pre>	Remove Remove		ELK Pin	C Used	
🛩 🐸 Middleware	Reset to default		Pin	🖾 Used	
👻 😂 Generic	Download and import sa	ample projects	Pin	/El Used	
r_byteq	Download and import a	imple projects	Pin		
		~		🖾 Used	
			on: QSPI_SMSTR_CFG_USE_FIT odules are used.		

Figure 4-26 Changing the version of FIT software component

- (2) Select the [Change version...] in the context menu.
- (3) If supported version is selected, the [Next] button will be clickable. Otherwise, the message "Selected version doesn't support current device or toolchain" is displayed.

Change Version						×
Version Selection Select available ver						
Component name:	r_qspi_smstr_rx					
Current version:	1.10					
Available versions:	1.14					~
	1.14 1.13					
	1.15					
	n 1		in a f		c 1	
	< Back	Next >	Finish		Cancel	

Figure 4-27 Select version of FIT software component

(4) Click the [Next] button.



(5) The setting change items are displayed.

Click the [Finish] button if there are no problems with changing the settings.

Change Version			×
Setting Overview			
The following settings will be added or removed			
Setting	Status	;	
✓ Configurations			
Use FIT	Adde	d	
Use QSPI Channel 0	Adde	d	
Enable debugging information	Adde	d	
SPTI0 Interrupt priority level	Adde	d	
SPRI0 Interrupt priority level	Adde	d	
< Back Next > Finish		Cance	l

Figure 4-28 Confirm setting change item

(6) The message "Confirm to change version and proceed to generate code" is displayed. Click [Yes] button if there are no problems.



Figure 4-29 Confirm setting change item

(7) The version of software component has been changed then the source codes are generated automatically.



4.4.9 Solving the grey-out component

When a component version is not available, it will be greyed out. Follow the procedure below to fix a greyedout component.

(1) From the component tree, right-click the greyed out component and select [Change version...]. Refer to chapter "chapter 4.4.8 Changing the version of the FIT software components" to change to an available version.

Components	15 日 田 🌢 🔻	Configure			
	1 U	Property		Value	
type filter text		✓	rations		
👻 😂 Startup		# Use F	IT	🖾 Used	
Y 😂 Generic		# Use C	QSPI Channel 0	🗷 Used	
💣 r_bsp		# Enabl	e debugging information	🖾 Unused	
Y 🗁 Drivers			Interrupt priority level	Level 10	
 Communications) Interrupt priority level	Level 10	
💣 r_ether_rx		 Resource 	25		
r_qspi_smstr_rx		11 B OCDI			
Y 🗁 Timers	Change version				
📽 r_cmt_rx	Remove Reset to default		CLK Pin	D Used	
🛩 🗁 Middleware) Pin	🖾 Used	
Y 😕 Generic	Download and import se	ample projects	Pin	D Used	
🐮 r_byteq			Pin	🖾 Used	
		~		🗆 Used	
			tion: QSPI_SMSTR_CFG_USE_FIT nodules are used.		^

Figure 4-30 Change version of a greyed out component

(2) If there is no available version for this component, refer to chapter "chapter 3.3.1 Downloading FIT modules" to download this component from Renesas website.



4.4.10 "i" mark on FIT modules icon

If the icon of FIT driver or middleware has an "i" mark [*], there are two meanings.

(1) Sample project of this FIT module is available

When there is sample project available for the FIT module, an "i" mark will appear to inform you.

Download the sample program files from Renesas Electronics home page.

https://www.renesas.com/fit

mponents	😀 🖬 👌 🖻 🗃 🔻	Configure	
pe filter text	1 U U	Property	Yalue
		Configurations # Parameter check	
Startup		# DICER control	System Default
			Clear all DTCER registers in R_DTC Full address mode
r_bsp		Address mode	
Drivers		# Transfer Data Read Skip	Enable transfer data read skip
Y 📴 DMA		DMAC FIT check	DMAC FIT module is used with E
e r_dtc_rx		# Sequence transfer	Sequence transfer not used
 Communications r_ether_rx 			
		Macro definition: DIC_CFG_PARAM_CHECKING_ENABLE Selects whether to include parameter checking in the code. BSP_CFG_PARAM_CHECKING_ENABLE = System default. 0 = compile out parameter checking. 1 = indudes parameter checking.	

Figure 4-31 FIT driver with sample project available for download in Renesas homepage

(2) Higher version of this FIT module is available in the computer

When the project is using an older version of FIT module, an "i" mark will appear if a higher version is available in the computer. A message will also appear when you mouse-over the FIT module.

Image: Smart_Configurator_Example.scfg ≥			- 0
Software component confi	guration	😼 👜 Generate Code 🛛 Generate Re	port
Components	Configure		í
🐮 🗟	Property	Value	^
type filter text	✓ [⊕] Configurations		
🕆 🗁 Startup 🔨	# Ethernet interface	MII (Media Independent Interface)	
🕆 🗁 Generic	# PHY-LSI address setting for ETHER0	0	
💣 r_bsp	# PHY-LSI address setting for ETHER1	1	
Y 🗁 Drivers	# The number of Rx descriptors	1	
Y 🗁 DMA	# The number of Tx descriptors	1	
💱 r_dtc_rx	# Transmit and Receive buffer size	1536	
Communications	# EINT interrupt priority level	Level 2	
💱 r_ether_rx	# Group AL1 interrupt priority level	Level 2	
12 A A A A A A A A A A A A A A A A A A A		Use ETHER1	~
	on of r_ether_rx is available: r_ether_rx[1.21]. sion, right click on the component, select "Change version"		^
~			~
Overview Board Clocks System Comp	oonents Pins Interrupts		

Figure 4-32 A newer version of FIT module is available in the computer

You can right-click the FIT module to change from current version to a higher version. Please refer to chapter "4.4.8 Changing the version of the FIT software components".



4.4.11 Configure Analog Front End component

The RX23E-A group microcontrollers are equipped with an analog front end (AFE) that can measure temperature, pressure, flow, and weight with less than 0.1% precision without calibration, making it ideal for high-precision sensing, test and measurement equipment.

When creating project for RX23E-A, you can use the AFE configuration tool for:

- Easy setting AFE on GUI
- Easy checking pins confliction
- Easy checking analog multiplexer connection

This chapter will describe how to use analog multiplexer connection:

- (1) In RX23E-A project, open smart configurator, select [Components] tab and add new component "Analog Front End" and "Continuous Scan Mode DSAD"
- (2) Select [Config_DSAD0] from the Components Tree. Perform setting as following:
 - Analog input channel setting: enable channel 0
 - [Channel setting] > [Channel 0] > Positive input signal: AIN1
 - [Channel setting] > [Channel 0] > Negative input signal: AIN3

Components		Software component configuration			Generate Code Gene
	2015 E E 2 *	Configure			
		- Basic setting			
type filter text		Analog input channel setting			
🛩 🇁 Startup		Channel D			
Y 😂 Genetic		Channel 1			
✓ r_bsp		Channel 2			
 Drivers A/D Converter 		Channel 3			
Config AFE		Channel 4			
Config_DSAD0		Channel 5			
		ΔΣ A/D Converter operation voltage setting			
		3.6V to 5.5V (High precision)	0 2.7V to 5.5V		
		ΔΣ A/D Converter operation mode setting			
		Normal mode	C Low-power mode		
		Operation clock setting	C ton ponte mode		
			PCLKB / 8	6.000 (MHz)	
		Operation clock			
		The operation clock can be set from 3.44 MHz to	4.56 MHz. Please set the operation clock within	n the range	
		Conversion start trigger setting			
		Start trigger source	Software trigger 🔗		
		Interrupt setting			
		Enable ΔΣ A/D conversion completion interrup	t (ADI0) Priority Level 15 (high	hest) ~	
		Senable ΔΣ A/D conversion scan completion int			
		Enable bz A/O conversion scan completion int	errup: (SCHAENDID) Priority Level 15 (nigi	nest) *	
		- Advance setting	Scroll down		
verview Board Clocks System Co	omponents Pins Interru	pts			
					51
oftware component cor	figuration				
oftware component cor	nfiguration	Channel 0 Grannel 1 Grannel 2 Grannel 3 Gran	inel 4 Channel 5		
oftware component cor	figuration	Channel 0 Dannel 1 Channel 2 Channel 3 Chan Anaiog input setting	inel 4. Channel 5		
offware component cor omponents type filter text	nfiguration	Analog input setting			
oftware component cor omponents type filter text. * Startup	nfiguration	Analog input setting Positive input signal	AIN1 ~	Rig	
offware component cor omponents type filter text	nfiguration	Analog input setting			
bottware component cor omponents type filter ted * Startup * Startup * Genetic * rbsp * Drivers	nfiguration	Analog input setting Positive input signal	AIN1 ~		
oftware component cor omponents type filter text	nfiguration	Analog input setting Positive input signal Negative input signal	AIN1 ~ AIN3 ~		Generate Code Gener
oftware component cor omponents type filter text. * § Startup * § Genetic. @ r_bsp * © Drivers * © ADC converter @ Config AFE	nfiguration	Analog input setting Positive input signal Negative input signal Reference input	AIN1 ~ AIN3 ~		
oftware component cor omponents type filter ted.	nfiguration	Analog input setting Positive input signal Negative input signal Reference input	AIN1 ~ AIN3 ~		
oftware component cor omponents type filter text.	nfiguration	Analog input setting Positive input signal Negative input signal Reference input Positive reference voltage buffer Negative reference voltage buffer	AIN1 ~ AIN3 ~		
oftware component cor omponents type filter text > Startup > Gerweix * type > Drivers > Drivers > Confg.AFE	nfiguration	Analog input setting Positive input signal Negative input signal Reference input Positive reference voltage buffer Negative reference voltage buffer Amplifier setting	AIN1 ~ AN3 ~ AVCCU/AVSS0 ~		
oftware component cor omponents type filter text.	nfiguration	Analog input setting Positive input signal Negative input signal Reference input Positive reference voltage buffer Ampölier setting Ampölier setting Buffer ampiliar:	AIN1		
 r_bsp Drivers A/D Converter Config_AFE 	nfiguration	Analog input setting Positive input signal Negative input signal Reference input Positive reference voltage buffer Ampölier setting Ampölier setting Buffer ampiliar: PGA gain setting	AIN1 ~ AN3 ~ AVCCD/AVSSO ~		
oftware component cor omponents type filter text > Startup > Gerweix * type > Drivers > Drivers > Confg.AFE	nfiguration	Analog input setting Positive input signal Reference input Positive reference voltage buffer Negative inference voltage buffer Amplifier setting Amplifier setted Buffer amplifier PICA glin setting ΔI A/D conversion setting	AIN1 AIN3 AIN3 AINCOLARYSSO Unused Positive buffer amplifier in		
oftware component cor omponents type filter text.	nfiguration	Analog input setting Positive input signal Negative input signal Reference input Positive reference voltage buffer Positive reference voltage buffer Amplifier setting Amplifier setting Amplifier setection 0.ffro amplifier PICA gain setting AJD AD conversion mode	AIN1 AIN3 AIN3 AVCCU/AVSS0 Unused Positive buffer amplifier X 1 Normal operation		
oftware component cor omponents type filter text.	nfiguration	Analog input setting Positive input signal Negative input signal Reference input Positive reference voltage buffer Anagative inference voltage buffer Ampilie setting Ampilie setting Anglifier selection Buffer ampifier PGA yain setting A/D conversion setting A/D conversion mode Data format	AIN1 AIN3 AIN3 AINCOVAVSSO Unused Positive buffer amplifier is 1 Normal operation Two's complement		Generate Code Gener
oftware component cor omponents type filter text.	nfiguration	Analog input setting Positive input signal Negative input signal Reference input Positive reference voltage buffer Positive reference voltage buffer Amplifier setting Amplifier setting Amplifier setection 0.ffro amplifier PICA gain setting AJD AD conversion mode	AIN1 AIN3 AIN3 AVCCU/AVSS0 Unused Positive buffer amplifier X 1 Normal operation		
oftware component cor omponents type filter text.	nfiguration	Analog input setting Positive input signal Negative input signal Reference input Positive reference voltage buffer Anagative inference voltage buffer Ampilie setting Ampilie setting Anglifier selection Buffer ampifier PGA yain setting A/D conversion setting A/D conversion mode Data format	AIN1 AIN3 AIN3 AINCOJ/RVSSO Unused Positive buffer amplifier is 1 Normal operation Two's complement Exponential operation mode (the number		Generate Code Gener
oftware component cor omponents type filter text.	nfiguration	Analog input setting Positive input signal Negative input signal Reference input Positive reference voltage buffer Anagative inference voltage buffer Ampilie setting Ampilie setting Anglifier selection Buffer ampifier PGA yain setting A/D conversion setting A/D conversion mode Data format	AIN1 AIN3 AIN3 AINCOJ/RVSSO Unused Positive buffer amplifier is 1 Normal operation Two's complement Exponential operation mode (the number	\sim af A/D conversions is from 1 to 8032) $\ \simeq$	Generate Code Gener
oftware component cor omponents type filter text.	nfiguration	Analog input setting Positive input signal Negative input signal Reference input Positive reference voltage buffer Negative inference voltage buffer Amplifier setting Amplifier setting POS dyn setting A/D conversion mode Data format A/D conversion number Oversampling ratio	AIN1 ~ AIN3 ~ AVCCD/AVSSD ~ Unused ~ Positive buffer amplifier ~ x 1 ~ Normal operation ~ Two's complement Exponential operation mode (the number 0 ~	of A/D conversions is from 1 to 8032) ↓ 1 ↓	Generate Code Gener
oftware component cor omponents type filter text.	nfiguration	Analog input setting Positive input signal Reference input Positive reference voltage buffer Negative raference voltage buffer Amplifier setting Amplifier setting Amplifier setting Applifier setting Applifier setting Applifier setting Applifier setting Applifier setting Applier setting A	AIN1 ~ AIN3 ~ AVCCD/AVSSD ~ Unused ~ Positive buffer amplifier ~ x 1 ~ Normal operation ~ Two's complement Exponential operation mode (the number 0 ~	of A/D conversions is from 1 to 8032) ↓ 1 ↓	Generate Code Gener
oftware component cor omponents type filter text.	nfiguration	Analog input setting Positive input signal Negative input signal Peference input Positive reference voltage buffer Positive reference voltage buffer Amptifier setting Amptifier settering Amptifier settering AMP conversion mode Date format A/D conversion number Oversampting ratio OSERT register value	AIN1 ~ AIN3 ~ AVCCD/AVSSD ~ Unused ~ Positive buffer amplifier ~ x 1 ~ Normal operation ~ Two's complement Exponential operation mode (the number 0 ~	of A/D conversions is from 1 to 8032) ↓ 1 ↓	Generate Code Gener

Figure 4-33 Config_DSAD0 Setting



- (3) Select [Config_AFE] from the Components Tree. In the [AFE setting] tab, change the [Bias output setting] as follows:
 - Enable bias voltage output: checked.
 - AIN1 pin output: checked
 - AIN3 pin output: checked

Software component configuration			Si Generate Co	Generate Code Generate Report	
Components	西山ら日常事・	Configure		0 ^	
type filter text		Setting Analog pins' connection			
ype tiret text → tire Sartup → tire Sartup → tire Cennic → tip Dolves → tip ADJ Converter → Config AFE → Config DSAD0	Note AB64 - AN11 pers conversion accuracy Bits cutoput cettino Or Inside bits voltage output ANA2 pin output ANA2 pin output ANA pin output ANA pin output ANA pin output ANA pin output ANA pin output	Vill be degraded if set the same pin in \$12AD			
		Excitation current output setting	International Character Reserves		
		Enable excitation current output			
		Operation mode	Z-channel output mode 🥣		
		Exotation current	50 µA		
		IEXC0 output pin	Output disabled		
		IEXC0 disconnect detection assist			
		IEXC1 output pin	Output deabled		
		EXC1 disconnect detection assist			
		1EXC2 output pin	Output disabled		
		IEXC2 disconnect detection assist			
		IEXC3 output pin	Output disabled	~	

Figure 4-34 Config_AFE setting

(4) Select [Analog pin's connection] tab, you can see the block diagram of the AFE multiplexed pin connection. The active connection of analog multiplexer is highlighted. So, you can check the analog multiplexer connection easily and confirm the configuration.



Figure 4-35 Block diagram of the AFE multiplexed pin connection


4.4.12 Configure Motor Component

Motor Driver Generator is a utility tool to generate drivers for all peripheral functions used for motor control from one GUI setting.

Note: The supported devices are RX13T, RX23T, RX24T, RX24U, RX26T, RX66T, RX72T, and RX72M.



Figure 4-36 Motor Driver Generator

This chapter will describe how to use Motor Driver Generator:

(1) In the project of supported device (for e.g. RX24T), open Smart Configurator, select [Components] tab and add new component "Motor". In the [New Component] dialog, select the Motor type as you wish and click [Finish] button.

🚺 New Component				×	🐼 New Component 📃	×
Software Component Selection				#	Add new configuration for selected component	-
Select component from those available in I	ist					
Category All				~	Motor	
Function All				~	Configuration name: Config_MTU3_MTU4	_
Filter					Motor type: 3-Phase Brushless DC Motor	1
Components	Short Name	Туре	Version	^	Resource: 3-Phase Brushless DC Motor 2-Phase Stepping Motor (Fast Decay)	
Low Power Consumption		Code Generator	2.3.0		2-Phase Stepping Motor (Slow Decay)	
# MEMDRV Driver	r_memdrv_rx	Firmware Integr	1.05			_
B Memory Driver Interface for Open Source	r_tfat_driver_rx	Firmware Integr	2.20	. 1		
# Motor		Code Generator	1.1.0			
Normal Mode Timer		Code Generator	1.12.0			
BOB1203 Sensor Middleware	r_ob1203_rx	Firmware Integr				
Open Source FAT File System.	r_tfat_rx	Firmware Integr	4.02			
Phase Counting Mode Timer		Code Generator	2.4.0	~		
Show only latest version						
Hide items that have duplicated function	ality					
Description						
This software component provides configu	uration for 8-Bit Timer	(TMR).		^		
TRAP or a duty new loc could be an orbital stress		and the second set on the second set		~		
Download the latest FIT drivers and middle						
Configure general settings						
(2) < Back	Next >	Finish	Cancel			
A Date	meat 2	ennall	cancer		C < Back Next > Finish Cancel	el

Figure 4-37 Add Motor component



(2) Select "Config_MTU3_MTU4" in the component tree, on the Configure panel, select the [Timer Setting] tab. In this tab, you will be able to use the GUI for Timer driver setting, including: Period Setting, Output Level Setting, Output Pin Select and Timer Interrupt Setting.

	6a 43 15 15 18 19 •	Configure			
	82				
type filter text		Timer Setting A/D Convertee	Setting		
 Santup Seneric e r_bsp b Drivers w Motor e Config_MTU3_MTU4 		Period setting Timer Operation Period Counter clock division rate TGRA register value Dead time	100 1 17280 10	us v (Actual frequency: 10.000 kHz)	Period
		A/D Conversion Trigger Inte	Current Control (A/Q) Co	mentin Triggel	

Figure 4-38 Timer Driver Setting (1)

Software component configuration			
Components	a d 3, 17 17 数 🔹	Timer Pulse Output Pin Selection	
	65	Timer Pulse Output Pin Selection	
type filter text		U phase MTU3 B-D	
 ✓ Startup ✓ Seneric 		Up pin : Not Used Un pin : Not Used	
🔹 r_bsp		on pin : Not osed	
- Drivers			
Y & Motor			
Config_MTU3_MTU4			
		W W	
		V phase MTU4 A-C W phase MTU4 B-D	
		Vp pin : Not Used Wp pin : Not Used	Output Pin Select
		Vn pin : Not Used Wn pin : Not Used	1
		Timer Interrupt setting	
		Use Crest Interrupt (MTU3.TGRA Compare Match Interrupt (TGIA3))	
		Interrupt Skipping Count Disable skipping function	
		Crest Interrupt Interval 10.000 kHz	
		Call user function from interrupt handler	Timer Interrupt
		Function Name CrestFunction	r·
		Crest Interrupt Priority Level	
		Level 15 (highest)	
		AND ALL OF ALL AND ALL ALL ALL ALL ALL ALL ALL ALL ALL AL	

Figure 4-39 Timer Driver Setting (2)



(3) Select the [A/D Converter Setting] tab. In this tab, you will be able to use the GUI for A/D Converter driver setting, including: Analog Pin Select, A/D Interrupt Setting.

	色 凸 凸 田 🏚 🔻	Configure		
	10 D			
type filter text		Timer Setting A/D Converter Setting		
👻 🗁 Startup		A/D Conversion Setting	1	
Y 🙆 Generic		Detected Input Pin	Analog Input Channel	
 e r_bsp. ∽ to brivers 			AN000(sample-and-hold used) ~	
 Motor 			AN000(sample-and-hold used)	
Config_MTU3_MTU4		D Iw	AN000(sample-and-hold used)	Angles Die Cattin
		Vdc	AN000(sample-and-hold used)	Analog Pin Settin
		Vu	AN000(sample-and-hold used)	
			AN000(sample-and-hold used)	
		□Vw	AN000(sample-and-hold used)	
		A/D Conversion End Interrupt Priority Level	Level 15 (highest) =	
		Call user function from interrupt handler		A/D Interrupt
		Function Name	AdFunction	and a survey of a
		the state intervention in the state of the s		A/D Interru

Figure 4-40 A/D Converter Driver Setting

(4) GPT peripheral is supported in some devices (for e.g., RX26T). Add new component "Motor". In the [New

Component] dialog, select the [Triangle_GPT] or [GPT0_GPT1_GPT2], [GPT4_GPT5_GPT6]resource and click

[Finish] button.

Note: The [GPT0_GPT1_GPT2] and [GPT4_GPT5_GPT6] resources are exclusively designed for GPT Complementary Mode, which is accessible only in RX26T

The [Triangle_GPT] resource is intended for GPT Triangle PWM Mode. With this resource, users can modify GPT channel configurations for both Master and Slave channels, utilizing the currently available GPT channels. The [Triangle_GPT] resource is available on RX24T, RX24U, RX26T, RX66T, RX72M, RX72T

Components	ù 🖆 🖧 🖂 🕀 🌞 🍷	Configure					
	🐮 😨	New Component				×	
type filter text							
 ✓ Startup ✓ Seneric ✓ r_bsp 		Add new configuration	on for selected component				
		Motor					
		Configuration name:	Config_MTU3_MTU4				
		Motor type:	3-Phase Brushless DC Motor			~	
		Resource:	MTU3_MTU4			÷	
			MTU3_MTU4 MTU6_MTU7 GPT0_GPT1_GPT2 GPT4_GPT5_GPT6 Triangle_GPT				
		0	< Back Next >	Finish	Cance	2	1

Figure 4-41 Select MOTOR resource



(5) GPT resource device has some differences between MTU resources

Smart_Configura	tor_Example.scfg ×				- 0	
Software com	ponent configuration				te Code Generate Report	
Components	▲山鳥日田夢・	Configure			•	
	23					
type filter text		Timer Setting A/D Converter S	Setting			
✓ Startup✓ Startup		Master/Slave settings				
e r_bs		Master Channel GPT0 ~				
V Drivers			Ma	stor/Slava Sotting		
	fig_GPT0_GPT1_GPT2	Slave Channel 1 GPT1 V	Ivia	ster/Slave Setting		
		Slave Channel 2 GPT2 🗸				
		Period setting				
		Timer Operation Period	100	ps × (Actual frequency: 10.000 kHz)		
		Counter clock division rate	1			
		GTPR register value	14400			
		Dead time	10	us \vee (Actual value: 10)		
		Output Pulse and A/D Conve	rsion Trigger Setting			
		A/D Conversion Trigger Skip	And a second sec	nction 😪		
		A/D Conversion Trigger Inter	val 10,000 kHz			
				ALALALALA	1	
					77	
			Current Con	trol (A/D Conversion Trieger)		
				• •		
						O
			\rightarrow			Count operation
			+Period/2-			
		Pins Active Level	- • Deadtime	Period +		
		Up tow	8 []	
		Un Low			1	

Figure 4-42 Timer Driver Setting(1)

Software component configuration	Gener	rate Code Generate Report
Components 🚵 🖂 🗄 🖶 🕸 🔹	Wn Low	^
2.2	Note: Dead time setting is not reflected to the pictures.	-
type filter text.	Timer Pulse Output Pin Selection	
 ✓ Startup ✓ Startup ✓ Startup ✓ Chep ✓ Startup ✓ Config. GP10. GP11. GP12 	U phase COPID A-B Up pin : Not Used Un pin : Not Used	Output Pin Selec
	V phase GPT1 A-8 W phase GPT2 A-8	10 I I I I I I I I I I I I I I I I I I I
	Vp pin : Not Used Wp pin : Not Used	
	Vn pin : Not Used Wn pin : Not Used	
	mer Interrupt setting Use Crest Interrupt (GPT Master Overflow Interrupt (GTCM)) Interrupt Stopping Count Crest Interrupt Notion Crest Interrupt Notion Function Func	Interrupt setting

Figure 4-43 Timer Driver Setting (2)



4.4.13 Configure general setting of component

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

If you want to change the settings, click the [Configure general settings...] link on the [Software Component Selection] page displayed in the [New Component] dialog (Figure 4-11), and the [Preferences] dialog will be displayed, as shown in Figure 4-44.

C Preferences		— 🗆 X
type filter text	Component	
 > Help Module Download > Smart Configurator Component MCU/MPU Package App Pin Errors/Warnings 	Backup settings Number of trash item Note: Set to 0 to disabl Code Generator comp	ie the feature.
	(a) API function output:	Output all API functions according to the setting \sim
	(b)API code style:	Value with macro description \sim
	Dependency settings Change these options	to control how a component is added
	Adding dependency:	Add dependent component V
	(C) Checking dependency	: Ignore if dependent component is newer \sim
	(e) Location settings Location settings have	moved to the <u>Module Download</u> page
< >>		Restore Defaults Apply
		Apply and Close Cancel

Figure 4-44 Configure General Setting of Component

(a) The API function output has two options: "Output all API functions according to the setting" and "Output only initialization API function". "Output all API functions according to the setting" is the default selection.

If "Output all API functions according to the setting" is selected, all API functions will be generated.

If "Output only initialization API function" is selected, only initialization API function will be generated. (Only void R_{ConfigurationName}_Create (void), void R_{ConfigurationName}_Create_UserInit (void) in *.h *, *c * are generated out.)

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

Figure 4-45 Updates of "API function output"



Output only initialization API feature is supported for individual configuration (Code Generator component). For using this feature, please right-click the selected component and select the "Output only initialization API" from the context menu.

✓		Register value (CMCOR)
🗸 🗁 Timers		Compare match interrupt (CN
Config_CMT0	~	Generate code
		Output only initialization API
		Change resource
	*	Remove
		Duplicate
		Rename
		Reset to default
	+	Add Configuration >

Figure 4-46 Output only initialization API

(b) The API code style has two options: "Value with macro description" and "Value without macro description (raw HEX)". "Value with macro description" is the default selection.

If "Value with macro description" is selected, all API with macro description will be generated.

If "Value without macro description (raw HEX)" is selected, code with HEX value will be generated.

API code style:	Value with macro description	~
	Value with macro description Value without macro description (raw HEX)	

Figure 4-47 Updates of "API code style"

(c) The code generation behavior has two options: "Update configuration files" and "Re-generate all component files". "Update configuration files" is the default selection.

If "Update configuration files" is selected and generate code, Smart Configurator will check whether the files are existing inside the user project. If the file exists, the file will not be overwritten. However, configuration files (e.g., xxx_config.h) will still be refreshed when code is generated.

If "Re-generate all component files" is selected and generate code, Smart Configurator does not check the existence of the file and the file will always be overwritten.

FIT(RX) / SIS(RL78) component settings					
Code generation behavior:	Update configuration files	~			
Dependency settings Re-generate all component files					
Change these options to con	ntrol how a component is added				

Figure 4-48 Updates of "Code generation behavior"

(d) Checking dependency has three options: "Do not check for dependent component", "Ignore if dependent component is newer" and "Strict check for dependent component". "Ignore if dependent component is newer" is the default selection.

If the version of the module and its dependency do not match, a warning message with code W04020011 will be displayed. If the user checks the revision history of the module and its dependencies and determines that there is no need to change the module being used, they can ignore this warning.



To clear this warning, select "Do not check for dependent components" in the "Checking dependency" list box in component preferences and then click "Apply".

Checking dependency:	Ignore if dependent component is newer	~
	Do not check for dependent component	
Location settings	Ignore if dependent component is newer	
Location settings have n	Strict check for dependent component	

Figure 4-49 Updates of "Checking dependency"

(e) If you want to change the location of components, click [Module Download] link, then find [Location (RX)], click [Browse...] and select another folder.

Location settings	
Location settings have moved to the Module Download	page

Figure 4-50 Updates of "Location settings"



4.4.14 Export configuration of component

To export the current configuration of a component, click the ²⁴ [Export Configuration] button on the [Components] tab page. The configuration will be saved as an *.xml file.



Figure 4-51 Export Configuration (xml format)

4.4.15 Import configuration of component

To import the configuration of a component, click the 🔤 [Import Configuration] button and select the exported XML file.



Figure 4-52 Import Configuration (xml format)



4.5 Pin settings

The [Pins] page is used for allocating pin functions. You can switch the display 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.

Pin configuration			You can sort	1	-	Ge	nerate Code Gen	erate Report
Software Components	E Pin Function		by clicking	, the title.			2) II	8 6 6 6 6
Type filter text	type filter text (* =	any string, ? = any character)					All	×
✓ ▲ r_bsp ♥ r_bsp	Ena Function	Assignment P17/MTIOC3A/MTIOC3B/MT	Pin Number IOC48 / 46	Direction	Remarks	Comments		^
Y 🚣 r_ether_rx	AN100	· PE2/D10/MTIOC4A/GTIOC0	3-A/PC / 133	4				
r_ether_rx	AN101	 Not assigned 	/ Not assigned	None				
r_qspi_smstr_rx	AN102	Not assigned	Not assigned	None				
r_qspi_smstr_rx	AN103	Not assigned	Not assigned	None				
 Single Scan Mode S12AD 	AN104	Not assigned	# Not assigned	None				
Config_S12AD1	AN105	Not assigned	Not assigned	None				
	AN106	Not assigned	/ Not assigned	None				
	AN107	Not assigned	Not assigned	None				
	AN108	Not assigned	Not assigned	None				
	AN109	Not assigned	Not assigned	None				Y



in Number	You can so	-	Лау				an enter		R 222
type filter tex	t (* = any string, ? = any charactery	the title.				comm	ents.	All	v
Pin Number	Pin Name	Board Functions	Function	Direction	Remarks	Symbolic Name	Comments		^
1	AVSS0	AVSS0	AVSS0		Read only		AVSS0		
2	P05/IRQ13/DA1	P05	Not assigned	None		LED1	LED1		
3	AVCC1	AVCC1	AVCC1		Read only		AVCC1		
4	P03/IRQ11/DA0	P03	Not assigned	None		LEDO	LEDO		
5	AVSS1	AVSS1	AVSS1		Read only		AVSS1		
6	P02/TMCI1/SCK6/IRQ10/AN120	SCK6	Not assigned	None			SCK6		
7	P01/TMCI0/RXD6/SMISO6/SSCL6/IRQ9/AN119	SMISO6	Not assigned 👻	None			SMISO6		
8	P00/TMRI0/TXD6/SMOSI6/SSDA6/IRQ8/AN118	SMOSI6	Not assigned	∧ one			SMOS16		
9	PF5/IRQ4	PF5	P01	one			SDPWREN		
10	EMLE	EMLE	TMCI0	1.000			EMLE		
11	PJ5/POE8#/CTS2#/RTS2#/SS2#	PJS	RXD6	one			XDRIVE		
12	VSS		SMISO6	100	Read only				
13	PJ3/EDACK1/MTIOC3C/ET0_EXOUT/CTS6#/RTS6#/CTS.	MTIOC3C	SSCL6	one			MTIOC3C		
14	VCL		IRQ9	Y	Read only				
15	VBATT	1000.00	VBATT	1	Read only				
16	NC	NC	Not assigned	None		·**	NC		,

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

When you select a board on the [Board] page, the initial pin setting information of the board is displayed in [Default Function]. In addition, the [1] icon displayed in the [Function] selection list indicates the initial pin function of the board.



4.5.1 Assign pins to resources

In the Pins page, assign pin to the resource used by the component. Pin assignment can be done in either [Pin Function] list or [Pin Number] list.

The procedure for pin assignment in the [Pin Function] list is described below.

- (1) Click on 🔚 (Show by Hardware Resource or Software Components)] to switch to the software component view.
- (2) Select the target software component (e.g. Config_S12AD1).
- (3) Click the [Enabled] header to sort by pins used.
- (4) Pin assignment is performed with the [Assignment], [Pin Number] column or [(Next group of pins for the selected resource)] button.
 - (a) Click [Assignment] or [Pin Number] and assign a terminal from the list (e.g. change from P17 to P13).
 - (b) Click the [(Next group of pins for the selected resource)] button and change the pin assignment. Each time you click, the pin with the function switches.

(1)												_	
Software Compo 🕀 🕞 🕴 😹	P	in Function								(4))-(b)	्र 🖬 🖫	è d
Type filter text		type filter te	xt	(A) (a)								All	\sim
V 着 Gingle Core Marks \$124.0		Enabled	unction	(4)-(a)		Die Number	Direction	Personales	Commente				
(2) Config_S12AD0	3		ADTRG0#	P07/IRQ15/ADTRG0#	~	/ 176	1						
	(3		AN000	Not assigned									
			AVCC0	P25/CS5#/EDACK1/MTIOC4C/MTCLKB/T	IOCA4/	PO5/RXD3/SMISC	3/SSCL3/SSIDA	TA1/HSYNC/ADTRG	0#				
		\checkmark	AVSS0	P16/MTIOC3C/MTIOC3D/TIOCB1/TCLKC	/TM02/	/PO14/RTCOUT/T	XD1/RXD3/SM0	DSI1/SMISO3/SSDA1	/SSCL3/SCL2/USB0_VI	BUS/USB	VBUSEN/USB	0_OVRCURB/IRQ	6/ADTRG0#
			AN001	P07/IRQ15/ADTRG0#									
			AN002	X	_			-					

Figure 4-55 Pin Assignments in the [Pin function] List

When the component is set, the check box in the [Enabled] column is checked. Pin assignment is possible even when the component is not set. If pin assignment is done without component being set, we will display "No component is using this pin" in the [Remarks] column.



4.5.2 Pin setting using MCU/MPU package

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

- (1) Zoom in to the view by clicking the (2 or 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.



Figure 4-56 Assigning Pins Using the MCU/MPU Package View

(a) The color of the pins can be customized through [Preference Setting...].



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

oftware Components	🗈 🎭 🚵 Pin Function						Q 🗉	🖾 èn e
Type filter text	type filter text (* =	any string, ? = any character)					All	~
× ắ r_bsp ■ r_bsp	Ena Function	Assignment / P17/MTIOC3A/MTIOC3B/MT	Pin Number	Direction	Remarks	Comments		^
✓ ▲ r_ether_rx	AN100	/ PE2/D10/MTIOC4A/GTIOC0	3-A/PC / 133	o Pin Number				
💣 r_ether_rx	AN101	Not assigned	Not a second					
4 4 r_qspi_smstr_rx	AN102	Not assigned	r Not a 🚽	comment to Pin Nu	mber tab			
💣 r_qspi_smstr_rx	AN103	Not assigned	Not a Clear of All Clear of	omments				
🖌 🚣 Single Scan Mode S12AD	AN104	Not assigned	Not a Assign	selected pins				
Config_S12AD1	AN105	Not assigned	Not a Unassi	gn selected pins				
	AN106	Not assigned	Not assigned	None				
	AN107	Not assigned	Not assigned	None				
	AN108	Not assigned	Not assigned	None				
	AN109	Not assigned	Not assigned	None				
	AN110	Not assigned	Not assigned	None				
	AN111	Not assigned	Not assigned	None				
	AN112	 Not assigned 	Not assigned	None				
	AN113	Not assigned	Not assigned	None				
Function Pin Number								
erview Board Clocks System Components Pins Inte	rrupts							

Pin Number							8 B a d
							All 🗸
Pin Nu Pin Name	Board Functions	Function	Directi	Remarks	Symbolic Name	Comments	^
128 P70/SDCLK		Not assigned	None				
129. VSS		VSS		Read only			
130 PE5/D13/MTIOC4C/MTIOC28/GTL.		Not assigned	None				
131 PE4/D12/MTIOC4D/MTIOC1A/GTL.		Not assigned	None				
132 PE3/D11/MTIOCAR/GTIOC2A-A/P		Not assigned	None				
133 PE2/D10/MTIOC4A/GTIOC0B-A/P		AN100	1				
134. PE1/D9/MTIOC4C/MTIOC3B/GTIO	N 1	Not assigned	None				
135 PE0/D8/MTIOC3D/GTIOC2B-A/SCK.	2	Not assigned	None				
136 P64/CS4#/WE#		Not assigned	None				
137 P63/CS3#/CAS#		Not assigned	None				×
in Function Pin Number							
Werview Board Clocks System Components P	ins Interrupts						

Figure 4-57 Jump to pin number



4.5.4 Export pin settings

You can export pin assignment settings in XML format. Exported files can be imported into projects of the same device family. Follow the procedure below to export the pin settings.

- (1) Click on the [12] (Export board setting)] button on the [Pins] page.
- (2) In the [Export] dialog, enter the file name to export.

Hardware Resource 🗈 🖻	l¶2 &∰	Pin Functi	on				(1)	
Type filter text							ି ଏ 🗐 🖬	è
		type filte	r text (* = an	y string, ? = any character)		All		
🚣 All	^	Enabled	Function	Assignment	Pin Number	Direction	Remarks	
Clock generator			ADTRG1#	P17/MTIOC3A/MTIOC3B/MTIOC4B/GTIOC0B-B/TIOCB0/TCLK	/ 46	1		
Clock frequency accuracy measurement	nt		AN100	PE2/D10/MTIOC4A/GTIOC0B-A/PO23/TIC3/RXD12/SMISO12		1		
₽¦ä Buses			ET0_COL	PC7/UB/A23/CS0#/MTIOC3A/MTCLKB/GTIOC3A-D/TMO2/TC	/ 76	1		
EXDMA controller			ETO_CRS	P83/EDACK1/MTIOC4C/GTIOC0A-D/CTS10#/ET0_CRS/RMII0	/ 74	1		
Interrupt controller unit			ETO_ERXDO	P75/CS5#/PO20/SCK11/RTS11#/ET0_ERXD0/RMII0_RXD0/MI	/ 87	1		
 Multi-function timer pulse unit 3 			ET0_ERXD1	P74/A20/CS4#/PO19/CTS11#/ET0_ERXD1/RMII0_RXD1	/ 88	1		
MTU0			ETO_ERXD2	PC1/A17/MTIOC3A/TCLKD/PO18/SCK5/SSLA2-A/ET0_ERXD2,	/ 89	1		
MTII1	× 1		ETO EDVDO	* DC0/A4C/MITIOCOC/TCLVC/DO47/CTCF#/DTCF#/CCF#/CCLA4	• 01			
	·					_		<i>_</i>
in Function Pin Number								

Figure 4-58 Export Pin Settings (XML format)

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

4.5.5 Import pin settings

You can import XML format files including pin assignment settings. When you import a file, the terminal assignment is reflected. Follow the procedure below to import the pin settings.

- (1) Click on the i (Import board setting)] button on the [Pins] page.
- (2) In the [Import] dialog, enter the file name to import.

Smart_Configurator_Example.scfg 🛛						5	<u>a</u>	-
in configuration						Generate Cod		Rep
Hardware Resource 🗉 🖻	Jª₂ 💑	Pin Funct	ion				3	è
Type filter text		type filte	r text (* = an	y string, ? = any character)		All		
🚣 All	^	Enabled	Function	Assignment	Pin Number	Direction	Remarks	
Clock generator		\checkmark	ADTRG1#	P17/MTIOC3A/MTIOC3B/MTIOC4B/GTIOC0B-B/TIOCB0/TCLK	/ 46	1		
Clock frequency accuracy measurement	nt		AN100	PE2/D10/MTIOC4A/GTIOC0B-A/PO23/TIC3/RXD12/SMISO12	/ 133	1		
ම් Buses			ET0_COL	PC7/UB/A23/CS0#/MTIOC3A/MTCLKB/GTIOC3A-D/TMO2/TC	/ 76	1		
EXDMA controller		~	ET0_CRS	P83/EDACK1/MTIOC4C/GTIOC0A-D/CTS10#/ET0_CRS/RMII0	/ 74	1		
Interrupt controller unit			ETO_ERXD0	P75/CS5#/PO20/SCK11/RTS11#/ET0_ERXD0/RMII0_RXD0/MI	/ 87	1		
 Multi-function timer pulse unit 3 		~	ET0_ERXD1	P74/A20/CS4#/PO19/CTS11#/ET0_ERXD1/RMII0_RXD1	/ 88	1		
MTU0		\checkmark	ET0_ERXD2	PC1/A17/MTIOC3A/TCLKD/PO18/SCK5/SSLA2-A/ET0_ERXD2	/ 89	1		
• MTU1	>	<	ETO EDVDO	* DC0/A4C/MITIOC2C/TCLVC/DO47/CTCL#/DTCL#/CCLA4	1.01			>

Overview Board Clocks System Components Pins Interrupts

Figure 4-59 Import Pin Settings (XML format)



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] 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)] button.
- (3) When [Assign default board pins] dialog opens, click [Select all].
- (4) Click [OK].

Smart_Configurator_Example.scfg ×				C.	nerate Code Generate Report	MCU/MPU Package ×
in Number				Ge	(2)	
type filter text (* = any string, ? = any characte)		C Default Board Pin	×	All ~	
Pin Nu Pin Name 1 AVSS0 2 P05/IRQ13/DA1 3 AVCC1 4 P03/IRQ11/DA0	Board Fu AVSS0 P05 AVCC1 P03	AVSS0 Not assig AVCC1		(3)	Comments ^ AVSS0 LED1 AVCC1 LED0	
S AVSS1 P02/TMC11/SCK6/IRQ10/AN120 P01/TMCI0/RXD6/SMISO6/SSCL6/I. P00/TMRI0/TXD6/SMOSI6/SSDA6/- PF5/IRQ4	AVSS1 P02 IRQ9 IRQ8 IRQ4	AVSS1 Not assig Not assig Not assig Not assig	P46	Select all	AVSS1 PMOD1-IO1 SW2 SW1 RL78G1CCTS	Renesas
10 EMLE 11 PIS/PDE8#/CTS2#/RTS2#/SS2# 12 VSS 13 PI3/EDACK1/MTIOC3C/ET0_EXOUT_ 14 VCL	EMLE PJ5 PJ3	EMLE Not assig VSS Not assig VCL	A6 A7 A8 A9 A10		EMLE LED3 RL78G1CRTS	RX65N
15 VBATT 16 MD/FINED 17 XCIN 18 XCOUT 19 RES# 20 P37/XTAL	MD XCIN XCOUT RES# XTAL	VBATT Not assig XCIN XCOUT Not assig XTAL			MD/FINED XCIN XCOUT RESn XTAL	
20 P3//XAL 21 VSS 22 P36/EXTAL	EXTAL	VSS (4) ок	Cancel	EXTAL	

Figure 4-60 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.

'in Fun	n Function											
type fi	type filter text (* = any string, ? = any character)											
Ena	Function	Assignment	Pin Number	Direction	Remarks	Comments	All					
	AO	Not assigned	Not assigned	None			Function					
	A1	Not assigned	Not assigned	None			Assignment Pin Number					
	A2	Not assigned	Not assigned	None			Direction					
	A3	Not assigned	Not assigned	None			Remarks					
	A 4	f Mat assigned	f National	Mana			Comments					

Figure 4-61 Filter for [Pin Function] Tab

ype filte	r text (* = any string, ? = any	y character)						All
Pin Nu	Pin Name	Board Fu	Function	Directi	Remarks	Symbolic Name	Comments	All
1	AVSS0	AVSS0	AVSS0		Read only		AVSS0	Pin Number
2	P05/IRQ13/DA1	P05	Not assigne	None			LED1	Pin Name Board Functions
3	AVCC1	AVCC1	AVCC1		Seed only		AWEC1	Function
-4	P03/IRQ11/DA0	P03	Not assigne	None			LEDO	Direction
5	AVSS1	AVSS1	AVSS1		Read only		AVSS1	Remarks
-	and the second se							Symbolic Name
n Functio	n Pin Number	onents Pins Interrupts						Comments

Figure 4-62 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, on the [New Component] dialog, click the [Configure general settings...] link to display the [Preferences] dialog. Then select [Smart Configurator] > [Pin Errors/Warnings] and use the combo boxes to change the errors/warning setting.

C Preferences			\times
type filter text	Pin Errors/Warnings	⇔ ▼ ⇔	• •
> Help Module Download	- Pin Conflict		
✓ Smart Configurator	Multiple functions are assigned in one pin number	Error	\sim
Component	 No Pin Allocation 		
MCU Package Appearance	Function used by software but not allocated to any pin	Error	\sim
Pin Errors/Warnings	 Mutually Exclusive Pins 		
	Mutually exclusive pins cannot be allocated to the same pin at the same time.	Error	\sim
	 No Software 		
	Assigned pins but there's no software using them	Info	~
	- Different Group		
	Functions in same channel but different group	Warning	g ~
	▼ Board Mismatch		
	Pin assignment does not match the board suggested pin assignment	Warning	g ~
	Restore <u>D</u> efaults	<u>A</u> pply	
	Apply and Close	Cancel	

Figure 4-63 Pin Errors/Warnings settings at Preferences

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



Figure 4-64 Change "No Software" setting from "Info" to "Error"



4.5.9 Symbolic name setting

[Symbolic Name] is an attribute of pins and can be found in [Pin Number] page and [MCU/MPU Package] page. It allows users to utilize their own symbols. The use of symbolic names in the user's application allows the source code to remain unchanged even when the MCU is changed, and pin assignments remapped. When a symbolic name is entered into the Pin page or the MCU/MPU Package view for any port pin, a macro definition will be generated in the Pin.h file

in configu	auton -						(2) Ge	enerate Code Ger	nerate Report
in Number									6 202
type filter text	t (* = any string, ? = any character)							All	Ý
Pin Number	Pin Name	Board Functions	Function	Direction	Remarks	Symbolic Name	Comments		^
1	AVSSO	AVSS0	AVSS0		Read only		AVSS0		
2	P05/IRQ13/DA1	P05	Not assigned	None		LED1	LED1		
3	AVCC1	AVCC1	AVCC1		Read only	-	AVCC1		
4	P03/IRQ11/DA0	P03	Not assigned	None		LED0	LED0		
5	AVSS1	AVSS1	AV5S1		Read only	÷	AVSS1		
6	P02/TMCI1/SCK6/IRQ10/AN120	SCK6	Not assigned	None			SCK6		
7	P01/TMCI0/RXD6/SMISO6/SSCL6/IRQ9/AN119	SMISO6	Not assigned 😽	None			SMISO6		
8	P00/TMRI0/TXD6/SMOSI6/SSDA6/IRQ8/AN118	SMOSI6	Not assigned	∧ one		(1)	SMOS16		
9	PF5/IRQ4	PF5	P01	one		(1)	SDPWREN		
10	EMLE	EMLE	TMCI0			-	EMLE		
11	PJ5/POE8#/CTS2#/RTS2#/SS2#	PJ5	RXD6	one			XDRIVE		
12	VSS		SMISO6		Read only				
13	PJ3/EDACK1/MTIOC3C/ET0_EXOUT/CTS6#/RTS6#/CTS.	MTIOC3C	SSCL6	one			MTIOC3C		
14	VCL		IRQ9	~	Read only				
15	VBATT		VBATT	1	Read only	*			
16	NC	NC	Not assigned	None			NC		v

Figure 4-65 pin setting of symbolic name

After generating code check at pin.h file.

Workspace 👻 🖡 🗙	Pin.h x	•
Debug ~		fo
Files Files Smart_Configurator_Example ✓ Renesss_AP Groups and a second Files	<pre>* 2) Not using symbolic name macro * Call the symbolic name APIs directly * PIN_WRITE(5,4) = ~PIN_READ(5,4) /* Symbolic name */ #define LED0 0,5 #define LED0 0,5 #define PIN_WRITE_HELPER(x,y) ((PORT##x.PODR.BIT.B##y)) /* Pin read helper */ #define PIN_READ_HELPER(x,y) ((PORT##x.PIDR.BIT.B##y)) ##if !(defined_CCRX_) && defined(_cplusplus)) /* Pin write API */ #define PIN_WRITE() (PIN_WRITE_HELPER(_VA_ARGS_)) /* Pin read API */ #define PIN_READ() (PIN_WRITE_HELPER(_VA_ARGS_)) ##else /* CC-RX' (-++ mode does not support variadic macros */ /* Pin write API */</pre>	
Smart_Configurator_Example		Image: A start and a start

Figure 4-66 Symbolic Name in generated code



```
void main(void)
{
    // Set port direction
    PORTB.PDR.BYTE = 0x08U;
    PORTE.PDR.BYTE = 0x20U;

    //Init LED status
    PIN_WRITE(LED1) = 1U;
    PIN_WRITE(LED0) = 0U;

    //Toggle LEDs
    while(1){
        PIN_WRITE(LED0) = ~PIN_READ(LED1);
        PIN_WRITE(LED1) = ~PIN_READ(LED0);

        //Toggle LEDs
        for (int i = 0;i < 100000;i++){
            nop();
        }
    }
}
</pre>
```

Figure 4-67 Using Symbolic Name in main function



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. Set the interrupt priority levels, the source of the fast interrupt, or a dynamic interrupt vector number.

nterrupt co	nfiguration						Generate Code Ge	ierate Repor
Interrupt vecto	ors used							
Up	Type filter text							
Down	Vector Number	Interrupt SPRI	Peripheral QSPI	Priority Level 10	Status Used	Multiple Interrupts	Fast Interrupt	
	43	SPTI	QSPI	Level 10	Used			
	✓ 111	GROUPBL1		Level 15	Used			
	20	S12CMPI	S12AD		Used			
	> 113	GROUPAL1		Level 2	Used			
	190	INTB190 (S12ADI)	S12AD	Level 15	Used			
	Note:							
	The interrupt prio	rity settings made here m	av not be utilized in	some FIT components.				

Figure 4-68 [Interrupts] Page

4.6.1 Changing the interrupt priority level and fast interrupt 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 [Interrupts] page.
- (2) To use an interrupt as a fast interrupt, tick the checkbox in the [Fast Interrupt] column. Only one interrupt can be specified as a fast interrupt among all interrupts and components used.

terrupt conf	iguration						Generate Code Ger	ierate Repo
Interrupt vectors	used							
Up	Type filter text							
Down	Vector Number 42 43	Interrupt SPRI SPTI	Peripheral QSPI QSPI (1)	Priority Level 10 Level 10	Status Used Used	Multiple Interrupts	Fast Interrupt	
(a)	> 111	GROUPBL1 GROUPAL1		Level 15 Level 3 Level 4	Y Used ∧ Used	- (2		
	190	INTB190 (S12ADI)	S12AD	Level 4 Level 5 Level 6 Level 7 Level 8 Level 9 Level 10 Level 10 Level 11 Level 13 Level 14 Level 14 Level 15	Used	. (2		
	Store and the second second	rity settings made here m configuration files of each		some FIT components. the correct priority settings	el .			

Figure 4-69 Interrupt Settings

(a) Group interrupts are collapsed in the interrupt table. Click on the [> (Open)] button to expand the view and see the interrupts in the group interrupt list.



4.6.2 Changing the interrupt priority level and fast interrupt setting

The [Interrupt configuration] page enables you to change the vector numbers of software configurable interrupts A and B.

- (1) Select a desired software configurable interrupt.
- (2) The [Up] and [Down] buttons will be enabled. Click on a button to change the vector number.

terrupt con	figuration						Generate Code Gene	iate Rep
nterrupt vector	s used							B
Up	Type filter text							
Down	Vector Number	Interrupt	Peripheral	Priority	Status	Multiple Interrupts	Fast Interrupt	
Down	42	SPRI	QSPI	Level 10	Used			
	43	SPTI	QSPI	Level 10	Used			
	✓ 111	GROUPBL1		Level 15	Used			
	20	S12CMPI	S12AD		Used			
	> 113	GROUPAL1		Level 2	Used			
(1)	190	INTB190 (S12ADI)	S12AD	Level 15	Used			
	-							
	Note:							
	The interrupt prio	rity settings made here n	nav not be utilized in	some FIT components.				
				the correct priority settings.				

Figure 4-70 Changing the Vector Number of Software Configurable Interrupt A or B



4.6.3 Multiple interrupts setting

The multiple interrupt feature on the RX MCU allows the processing of another interrupt while the current interrupt is running. The setting of multiple interrupts can be configurated from both the Interrupt page and the Component configuration.

- (1) Select a component(supported multiple interrupt) and enable its multiple interrupts settings.
- (2) Multiple interrupts setting is bidirectional synchronization in [Interrupts] page.
- (3) Open generated file in project explorer, generated code can be found.

	è d 🔓 🖻 🖽	🔆 🔻 Configure						
onents								
		Count clock set						
e filter text		PCLK/8	○ PCLK/32	O PCLK/128		/512		
Startup		Compare mate	h setting					
🛛 🗁 Generic	c	Interval value		100		μs	 (Actual value) 	e: 100)
Drivers						μο	(Actual Val	. 100)
A/D Co		Register value	(CMCOR)	749				
Comm Comm Comm Comm		Compare r	natch interrupt (CMI0)					
	nfig_CMT0	Fnable mu	ltiple interrupts (CMI0) (2)				
			tupic interrupts (cimb) (
(1)	Priority		Level 15 (highest)	×			
w Board C	Clocks System Com	Pins Interrupts Figure 4-7	1 Multiple inte	errupts in com	ponent	page		
	Clocks System Com		1 Multiple inte	errupts in com	ponent	page	(3) Generate Cod	e Generate
	nfiguration		1 Multiple inte	errupts in com	ponent	page	(3) Generate Cod	
rupt con	nfiguration		1 Multiple inte	errupts in com	ponent	page	(3) Generate Cod	e Generate
rrupt con rrupt vector Up	nfiguration		1 Multiple inter	Priority	ponent	page	(3) Generate Cod	e Generate Fast I 1
rrupt con	Type filter text Vector Number 16	Figure 4-7	Peripheral BSC	Priority Level 15	ponent		(3) Generate Cod Multiple Interrupts	e Generate
rrupt con rrupt vector Up	Type filter text Vector Number 16 18	Figure 4-7	Peripheral BSC RAM	Priority Level 15 Level 15	ponent		(3) Generate Cod	Fast I
rrupt con rrupt vector Up	Type filter text Vector Number 16 18 21	Figure 4-7	Peripheral BSC RAM FCU	Priority Level 15 Level 15 Level 15	ponent		(3) Generate Cod	Fast I
rrupt con rrupt vector Up	Type filter text Vector Number 16 18 21 23	Figure 4-7	Peripheral BSC RAM FCU FCU	Priority Level 15 Level 15 Level 15 Level 15 Level 15			(3) Generate Cod	Fast I /
rrupt con rrupt vector Up	Type filter text Vector Number 16 18 21 23 26	Figure 4-7	Peripheral BSC RAM FCU FCU ICU	Priority Level 15 Level 15 Level 15 Level 15 Level 15 Level 15	(2) .		(3) Generate Cod	Fast I /
rrupt con rrupt vector Up	Type filter text Vector Number 16 18 21 23 26 27	Figure 4-7	Peripheral BSC RAM FCU FCU ICU ICU	Priority Level 15 Level 15 Level 15 Level 15 Level 15 Level 15 Level 15		Status	(3) Generate Cod	Fast I 4
rrupt con rrupt vector Up	Type filter text Vector Number 16 18 21 23 26 27 28	Figure 4-7	Peripheral BSC RAM FCU FCU ICU ICU ICU ICU CMT0	Priority Level 15 Level 15 Level 15 Level 15 Level 15 Level 15 Level 15 Level 15			(3) Generate Cod	Fast I
rrupt con rrupt vector Up	Type filter text Vector Number 16 18 21 23 26 27 28 29	Figure 4-7	Peripheral BSC RAM FCU FCU ICU ICU ICU ICU ICU ICU ICU ICU ICU I	Priority Level 15 Level 15		Status	(3) Generate Cod	Fast I
rrupt con rrupt vector Up	Type filter text Vector Number 16 18 21 23 26 27 28 29 30	Figure 4-7	Peripheral BSC RAM FCU FCU ICU ICU ICU ICU CMT0 CMT1 CMTW0	Priority Level 15 Level 15		Status	(3) Generate Cod	Fast I
rrupt con rrupt vector Up	Type filter text Vector Number 16 18 21 23 26 27 28 29 30 31	Figure 4-7	Peripheral BSC RAM FCU FCU ICU ICU ICU ICU ICU CMT0 CMT1 CMTW0 CMTW1	Priority Level 15 Level 15		Status	(3) Generate Cod	Fast I 4
rrupt con rrupt vector Up	Type filter text Vector Number 16 18 21 23 26 27 28 29 30 31 32	Figure 4-7	Peripheral BSC RAM FCU FCU ICU ICU ICU ICU CMT0 CMT1 CMTW0 CMTV1 USBA	Priority Level 15 Level 15		Status	(3) Generate Cod	Fast
rrupt con rrupt vector Up	Type filter text Vector Number 16 18 21 23 26 27 28 29 30 31 32 33	Figure 4-7	Peripheral BSC RAM FCU FCU ICU ICU ICU CMT0 CMT1 CMTW0 CMTW1 USBA USBA	Priority Level 15 Level 15 Lev		Status	(3) Generate Cod	Fast
rrupt con rrupt vector Up	Type filter text Vector Number 16 18 21 23 26 27 28 29 30 31	Figure 4-7	Peripheral BSC RAM FCU FCU ICU ICU ICU ICU ICU CMT0 CMT1 CMTW0 CMTW1	Priority Level 15 Level 15		Status	(3) Generate Cod	Fast I
rrupt con rrupt vector Up	Type filter text Vector Number 16 18 21 23 26 27 28 29 30 31 32	Figure 4-7	Peripheral BSC RAM FCU FCU ICU ICU ICU ICU CMT0 CMT1 CMTW0 CMTV1 USBA	Priority Level 15 Level 15		Status	(3) Generate Cod	Fast

Figure 4-72 Multiple interrupts in component in Interrupts page





Figure 4-73 Multiple interrupts in generated code



5. Managing Conflicts

Adding components, setting pins and interrupts may cause problems related to resource mismatch. This information will be displayed in the **Configuration Problems** view. You can refer to the information displayed to fix the conflict issues.

5.1 Resource conflicts

When two software components are configured to use the same resource (e.g. S12AD1), an error mark (S) will be displayed in the [Components tree].

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

	Smart_Configurator_Example.scfg ⊠		
S	oftware component configu	ration Generate C	and the content of th
C	omponents 🕴 🗄 🖶 🛱 🔻	Configure	^
	type filter text	Basic setting Note When using the 12-bit A/D converter unit 0, do not use the P40 to P47, P03, P05, and P07 pins as o We also recommend not using the P00 to P02, P90 to P93, PD0 to PD7, and PE0 to PE7 pins as outp Analog input mode setting Double trigger mode Analog input channel setting Anuo0 AN001 AN002 AN003 AN004 AN005 AN006 AN007 Conversion start trigger setting Start trigger source A/D conversion start trigger pin Interrupt setting Enable AD conversion end interrupt (S12ADI) Priority Level 15 (highest) ▼	
0	verview Board Clocks System Compone	ents Pins Interrupts	
	Subscript Configuration Problems ≈		
ſ	4 errors, 0 warnings, 0 others		
^	Description		Туре
	 Interrupt (2 items) 		
		ed by S12ADI in Config_S12AD01 conflicts with vector used by S12ADI in Config_S12AD0.	Interrupt
	I	sed by S12ADI in Config_S12AD0 conflicts with vector used by S12ADI in Config_S12AD01.	Interrupt
	 Ø Peripheral (2 items) 		
	· ·) used by Config_S12AD01 is already used by Config_S12AD0.	Peripheral
	E04010001: Peripheral S12AD0 E04010001: Peripheral S12AD0) used by Config_S12AD0 is already used by Config_S12AD01.	Peripheral

Figure 5-1 Resource Conflicts



5.2 Resolving pin conflicts

When multiple pin functions are assigned to the same pin, an error mark 🐼 is displayed in the tree and [Pin Function] list on the [Pins] page.

lardware Resource	🕀 🖻 📲 💑	Pin Functi	on				2 🖬 👪 🔤
Type filter text		type filte	r text (* = an	y string, ? = any character)		All	,
羞 SD slave interface	^	Enabled	Function	Assignment	Pin Num	ber	Direction
🕮 Parallel data capture unit			ADTRG0#	P07/IRQ15/ADTRG0#	/ 176		1
👗 Graphic-LCD controller			AN000	P40/IRQ8/AN000	/ 173		I
🚳 Realtime clock			AN005	P45/IRQ13/AN005	/ 167		1
🗸 🎕 12-bit A/D converter			AN001	Not assigned	Not as	signed	None
of \$12AD0			AN002	Not assigned	Not as	0	None
🔍 S12AD1			AN003	Not assigned	Not as	signed	None
强 12-bit D/A converter			AN004	Not assigned	Not as	signed	None
👗 Digital power supply			AN006	Not assigned	Not ass	signed	None
🗰 Operating mode control	~		AN007	Not assigned	Not ass	signed	None
<	>	<					

Figure 5-2 Pin Conflicts

The detailed information regarding conflicts is displayed in the [Configuration Problems view].

😫 Configuration Problems 🛱	
3 errors, 0 warnings, 0 others	
Description	Туре
✓ ❷ Pin (3 items)	
• E04010003: Pin used by ADTRG0# in Config_S12AD0 conflicts with pin used by IRQ15 in Pin Allocator, pin used by IRQ15 in Config_ICU.	Pin
• E04010003: Pin used by IRQ15 in Config_ICU conflicts with pin used by ADTRG0# in Config_S12AD0, pin used by ADTRG0# in Pin Allocator.	Pin
E05000010: Pin 176 cannot be used multiple times. Pin 176 is assigned to IRQ15 and ADTRG0#.	Pin

Figure 5-3 Pin Conflict Message

To resolve a conflict, right-click on the node with an error mark on the tree and select [Resolve conflict]. The pins of the selected node will be re-assigned to other pins.

n configuration					Generate Code	Generate Re
ardware Resource 🕀 🗎 ↓ªz	品	Pin Functi	ion		·	3 🖪 🔡 🖻
Type filter text		type filte	r text (* = any	y string, ? = any character)	All	
📥 SD slave interface	^	Enabled	Function	Assignment	Pin Number	Direction
🗰 Parallel data capture unit		\checkmark	ADTRG0#	P07/IRQ15/ADTRG0#	/ 176	1
🚣 Graphic-LCD controller		\checkmark	AN000	P40/IRQ8/AN000	/ 173	1
🚳 Realtime clock			AN005	P45/IRQ13/AN005	/ 167	1
🕯 🗟 12-bit A/D converter			AN001	Not assigned	Not assigned	None
© \$12AD0			AN002	Not assigned	Not assigned	None
S12AD1 Assign all			AN003	Not assigned	Not assigned	None
强 12-bit D/A o Unassign all			AN004	Not assigned	Not assigned	None
🚣 Digital powe Resolve conflict	Ν		AN006	Not assigned	Not assigned	None
Operating mode control	~		AN007	Not assigned	Not assigned	None
		<				

Figure 5-4 Resolving Pin Conflicts



6. Generating Source Code

6.1 Generating Source Code File

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

Smart_Configurator_Ex		n	Generate Code Senerate Repor
Components	↓ª_ □ ■ 🚔 ▼	Configure	ũ
type filter text	10 T	Property	Value ^
👻 🗁 Startup	^	# Startup select	Enable (use BSP startup)
👻 🗁 Generic		# User stack setting	2 stacks
💣 r_bsp		# User stack size	0x400

Figure 6-1 Generating a Source File

The Smart Configurator generates a source file in <ConfigurationFileDir>\src\smc_gen. If your Smart Configurator has already generated a file, a backup copy of that file is also generated (refer to the section 6.6, Backing up Generated Source Code).



6.2 Configuration of Generated Files and File Names

The Figure 6-2 below, shows the folders and files output by the Smart Configurator. "ConfigName" indicates the configuration name set in the component.







Folder	File	Description
{ProjectName}	{ProjectName}.eww	This file is generated only once during the first code
		generation.
		{ProjectName}.ewp file path is specified in this file.
	{ProjectName}.ewp	This file is generated only once during the first code
		generation.
		It appends the "buildinfo.ipcf" and "main.c" files at the end
		of this file.
	{ProjectName}.ewd	This file is generated only once during the first code
		generation.
		It is the same as the default *.ewd file that is generated by
	main a	IAR Embedded Workbench.
	main.c	This file is generated only once in the first code
		generation.
	buildinfo.ipcf	It contains main () function.
	bullulillo.ipci	This file is always generated.
		It contains source file registration information. From Smart Configurator for RX V2.15.0, the name of .ipcf file is
		updated into "buildinfo.ipcf".
		If user loads a project that was created before
		Smart Configurator for RX V2.15.0, the .ipcf file will be
		regenerated as "buildinfo.ipcf" but the original .ipcf file
		({ProjectName}.ipcf) will still exist in the folder and will not
		be removed.
general	-	This folder is always generated. It contains header files
		and source files commonly used by drivers of the same
		peripheral function.
	r_cg_hardware_setup.c	This file is always generated. It contains R_Systeminit that
		calls all driver initialization functions with the name
		R_ConfigName_Create. R_Systeminit also calls the
		functions for initializing clocks other than the clock source, fast interrupt, and group interrupts.
	r_cg_macrodriver.h	This file is always generated. This header file contains
		common macro definitions used in drivers.
	r_cg_xxx.h ^(Note*1)	These files are always generated. The files contain macro
		definitions for setting SFR registers.
	r_cg_userdefine.h	This file is always generated. User can add macro
		definitions in the dedicated user code areas.
	r_smc_cgc.c	This file is always generated. It contains the initialization of
		clock sources other than the clock source selected in the
		[Clocks] page.
	r_smc_cgc.h	This file is always generated. This header file contains macro definitions to initialize clocks other than the
		selected clock source.
	r_smc_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_smc_entry.h	This file is always generated. This file includes the header
		files of CG drivers that are added to the project. When
		using functions of CG drivers in source files added by
		user, including this file is necessary.
	r_smc_interrupt.c	This file is always generated. It contains fast interrupt and
		group interrupt initialization (depending on hardware
		specification).



r_bsp	r_smc_interrupt.h	This file is always generated. It contains macro definitions for fast interrupt and group interrupt initialization. It also contains the priority level of all interrupts that are configured in the [Interrupts] tabbed page. User can use these macro definitions in application codes. This folder is always generated. It consists of multiple subfolders (board, doc, mcu) with: - Initialization codes to start up the MCU before entering main () - Definitions of all SFR registers in iodefine.h (mcu folder) - Application note of r_bsp (doc folder) It also contains platform.h that will include r_bsp.h of the dovice used in the project
		device used in the project.
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 Pin.c
r_config	r_bsp_config.h	The file is always generated. It contains the configuration of BSP.
	r_bsp_interrupt_config.h	This file is always generated. It contains mapping of the software configurable interrupts A and B (depending on hardware specification).
{ConfigName}	-	This folder is generated for the added component. API functions in this folder are named after the ConfigName (configuration name).
	{ConfigName}.c	This file contains functions to initialize driver (R_ConfigName_Create) and perform operations that are driver-specific, e.g. start (R_ConfigName_Start) and stop (R_ConfigName_Stop).
	{ConfigName}_user.c	This file contains interrupt service routines and functions for user to add code after the driver initialization (R_ConfigName_Create). User can add codes and functions in the dedicated user code areas.
	{ConfigName}.h	This is header file for {ConfigName}.c and {ConfigName}_user.c.

Note *1: xxx is the name of a component.



6.3 Initializing Clocks

Configurations of clock source in [Clocks] page are generated in \src\smc_gen\r_config folder.



Figure 6-3 Clocks Source Configuration

Table 6-1	Clock Source File Description	
-----------	--------------------------------------	--

Folder	File	Macros/Functions	Description
general	r_cg_cgc.c	R_CGC_Create	This API function initializes clocks other than the selected clock source. <i>R_Systeminit</i> in <i>r_cg_hardware_setup.c</i> will call this function before entering <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.



6.4 Initializing Pins

Pin configuration settings are generated by the component into source files as shown in (1) and (2) below.

(1) Pins initialization for drivers with {*ConfigName*}

The pin function is initialized with *R_ConfigName_Create* of *\src\smc_gen\{ConfigName}\\ConfigName}.c.*

Table 6-2 File to Initialize Pins

Folder	File	Function	Description
{ConfigName}	{ConfigName}.c	R_ConfigName_Create	This API function initializes pins used by this component. <i>R_Systeminit</i> in <i>r_cg_hardware_setup.c</i> will call this function before entering <i>main()</i> function.

(2) Reference pins initialization codes

Refer to *Pin.c* in the *\src\smc_gen\r_pincfg* folder for the initialization code of all pin functions set on the [Pins] page (except I/O ports).

Table 6-3 Reference File for Initialization of All Pins

Folder	File	Function	Description
r_pincfg	Pin.c	R_Pins_Create	This function contains the initialization codes of all pins function configured at [Pins] page except I/O ports.



6.5 Initializing Interrupts

Configurations in [Interrupt] page are generated in few source files.

	Vector Number	Interrupt	Peripheral	Priority	Status	Fast Interrupt
	✓ 111	GROUPBL1	(1)	Level 15	Used	
	21	S12CMPI1	S12AD1		Used	(A)
	> 113	GROUPAL1		Level 2	Used	(4)
(3)	192	INTB192 (S12ADI1)	s12ad1 (2)	Level 15	Used	

Figure 6-4 Interrupt Configuration

Table 6-4	Interrupt	Generation	File	Description
-----------	-----------	------------	------	-------------

No	Item	Folder	File	Description
(1)	Priority	general	r_smc_interrupt.c	This interrupt priority level setting is for group interrupts ^(Note2) . It is initialized in <i>R_Interrupt_Create</i> of this file. <i>R_Systeminit</i> in <i>r_cg_hardware_setup.c</i> will call this function before entering <i>main()</i> function.
(2)	Priority	{ConfigName}	{ConfigName}.c	This interrupt priority level setting is for normal interrupts and software configurable interrupts A and B ^(Note2) . It is initialized in <i>R_ConfigName_Create</i> of this file. <i>R_Systeminit</i> in <i>r_cg_hardware_setup.c</i> will call this function before entering <i>main()</i> function.
(1) (2)	Priority	general	r_smc_interrupt.h	Priority level of all interrupts configured in the [Interrupts] tabbed page is defined in this file. User can use these macro definitions in the application codes.
(3)	Vector Number	r_config	r_bsp_interrupt_config.h	Vector number of software configurable interrupts A and B $^{(Note2)}$ in the [Interrupts] tabbed page will be mapped in this file and handled by r_bsp .
(4)	Fast Interrupt	general	r_smc_interrupt.c	Fast interrupt setting will be initialized in <i>R_Interrupt_Create</i> of this file. <i>R_Systeminit</i> in <i>r_cg_hardware_setup.c</i> will call this function before entering <i>main()</i> function.
			r_smc_interrupt.h	Priority level of all interrupts configured in the [Interrupts] tabbed page is defined in this file. User can use these macro definitions in the application codes.

Note *2: The type of interrupt depends on hardware specifications.

6.6 Backing up Generated Source Code

The smart configurator has a source code backup function.

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

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



7. Loading generated files in Integrated development environment

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

7.1 Adding Custom Code of FIT

When [FIT] is selected as the component type, the configuration options are set in r_xxx_config.h in the folder r_config. For the settings of the configuration options, refer to the application note (in the doc folder) on the FIT module (r_xxx) which you have added to the project tree.

If the target file already exists, the existing contents of the file are protected when source code is output.



Figure 7-1 Tree Structure of Directories and Files for a FIT Module



7.2 Loading in IAR Embedded Workbench

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

IAR Embedded Workbench.

The usage procedure is as follows.

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

Open Workspace						×
\leftarrow \rightarrow \checkmark \uparrow \square \rightarrow This PC	\rightarrow (C:) Windows \rightarrow IAR	~	Ō	,○ Search IAR		
Organize 👻 New folder					•	?
▲ Name	^	Date modified		Туре	Size	
🗯 📙 .settings		3/27/2023 5:21 P	M	File folder		
Src Src		3/27/2023 5:21 P	M	File folder		
trash		3/27/2023 5:21 P	M	File folder		
Stest.eww		3/27/2023 5:21 P	M	IAR IDE Workspace		1 KB
✓ <						>
File name:	test.eww		~	Workspace Files (*.	eww)	\sim
				Open	Cancel	

Figure 7-2 Load *.eww file



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

Files	\$	•
🗉 🌒 test - Debug *	4	
- 🛱 🛋 Renesas_AP		
🖵 📮 🛋 smc_gen		
– – – – – – – – – – – – – – – – – – –		
G Config_TMR0.c		•
Config_TMR0.h		
		•
– 🖓 🛋 general		
🗕 🕂 🖸 r_cg_hardware_setup.c		•
🛛 🛏 🗟 r_cg_macrodriver.h		
⊨ 🗕 r_cg_tmr.h		
🛛 🛏 🗟 r_cg_userdefine.h		
		•
🛛 📙 🖬 r_smc_cgc.h		
I III IIII IIII IIIIIIIIIIIIIIIIIIIII		•
📃 🚽 🖬 r_smc_entry.h		
H → E I r_smc_interrupt.c		•
└──_ k r_smc_interrupt.h		
⊢Ę ∎r_bsp		
🛛 🗕 🖬 board		•
doc		
l - ⊡ i mcu		•
📃 🚽 🖬 platform.h		
🖳 🖳 🗎 readme.txt		
│		
🛛 🚽 🖬 r_bsp_config.h		
🛛 🚽 🖹 r_bsp_config_readme.txt		
🛛 🖳 🖾 r_bsp_interrupt_config.h		
└─── 🖬 r_pincfg		
Here 🖸 Pin.c		•
📙 🔄 🖵 🖬 Pin.h		
D buildinfo.ipcf		
Here in the main.c		•
└─⊞ 🛑 Output		

Figure 7-3 New files added to IAR workspace

(4) Select "Options..." from the "Project" menu of IAR Embedded Workbench.



(5) Change the target device to match with the target device selected when creating Smart Configurator's configuration file inside the "Options for node" dialog box.

Figure 7-4 Options for node



7.3 Build IAR Project File

After successfully loading the Smart Configurator project file to IAR Embedded Workbench, user can rightclick on the project name and select [Rebuild All] from the context menu. This will execute the build operation successfully.

Debug	~	
Files	¢ • ^	
🗆 🌒 test - Debug 📃 📃	<u>ب</u>	
├-=	Options	
└──; 🛋 smc_gen	Make	
-🖓 🛋 Config_TMR(Compile	
□	Rebuild All	
│	Clean	
	C-STAT Static Analysis	>
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	Stop Build	
test	Add	>
Debug Log	Remove	
Log	Rename	
Mon Mar 27, 2023 17:24:45	Version Control System	>
Build	Onen Containing Folder	
Messages	Open Containing Folder File Properties	
r_cg_hardware_setup.c r_rx_intrinsic_functions.c	Set as Active	
r_smc_cgc.c		
r_smc_cgc_user.c		_
r_smc_interrupt.c vecttbl.c		
resetprq.c		
Linking		
Total number of errors: 0 Total number of warnings: 0		

Figure 7-5 Build C Project File in IAR



8. Creating User Programs

Create a user program in the IDE. This chapter describes how to add custom code to the source file generated by the SC.

8.1 Adding Custom Code in the Case of Code Generator

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

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

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

In the case of [Code Generator], three files are generated for each of the specified peripheral functions. The file names are "Config_xxx.h", "Config_xxx.c", and "Config_xxx_user.c" as the default, with "xxx" representing the name of the peripheral module. For example, "xxx" will be "CMT3" for the compare-match timer (resource CMT3). 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 CMT3 (Config_CMT3_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" #include "r_cg_userdefine.h" #include "Config_CMT3.h"</pre>
/* Start user code for include. Do not edit comment generated here */ /* End user code. Do not edit comment generated here */
/*************************************
/* Start user code for global. Do not edit comment generated here */ /* End user code. Do not edit comment generated here */
/**************************************
 * Function Name: R_Config_CMT3_Create_UserInit * Description : This function adds user code after initializing the CMT3 channel * Arguments : None * Return Value : None ************************************
<pre>void R_Config_CMT3_Create_UserInit(void) { /* Start user code for user init. Do not edit comment generated here */ /* End user code. Do not edit comment generated here */ }</pre>


* Function Name: r_Config_CMT3_cmi3_interrupt * Description : This function is CMI3 interrupt service routine * Arguments : None * Return Value : None **#if** FAST_INTERRUPT_VECTOR == VECT_PERIB_INTB129 #pragma interrupt r_Config_CMT3_cmi3_interrupt(vect=VECT(PERIB,INTB129),fint) #else #pragma interrupt r_Config_CMT3_cmi3_interrupt(vect=VECT(PERIB,INTB129)) #endif static void r_Config_CMT3_cmi3_interrupt(void) { /* Start user code for r_Config_CMT3_cmi3_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 */



8.2 Using Generated Code in user application

To use the generated code of FIT and Code Generator, follow the below steps:

1) Open the *main.c* file, add code to include the header files of the modules you want to use.

In case of FIT, it is <r_xxx_if.h>.

In case of Code Generator, it is <r_smc_entry.h>.

Workspace 💌 🕫 🗴	x main.c x	•
Debug		fo
Files Image: Configurator_Example - Debug Image: Config: CMT0 Image: Config: CMT0 Image: CMT0 Image: Config: CMT0 Image: CMT0 Image: Configurator_Example.ipcf Image: Configurator_Example.ipcf Image: Configurator_Example.ipcd Image: Configurator_Example.ipcf Image: Configurator_Example.ipcd Image: Configurator_Example.ipcf Image: Configurator_Example.ipcd Image: Configurator_Example.ipcf Image: Configurator_Exampl	<pre>include <r_smc_entry.h> int main(void) (/* Start CMT0 counter operation */ R_Config_CMT0_Start(); while(U) { nop(); } }</r_smc_entry.h></pre>	
Smart_Configurator_Example	<	>

Figure 8-1 Add header files

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

In case of Code Generator, driver initialization functions ($R_ConfigName_Create$) including initialization of pins have been called in $R_Systeminit$ function of $r_cg_hardware_setup.c$ by default. You just need to add application codes to perform operations that are driver-specific, for e.g. start ($R_ConfigName_Start$) and stop ($R_ConfigName_Stop$).

Workspace	* # X	main.c x	*
Debug	14		fo
Files • Smart_Configurator_Example - Debug • Renesss_AP • Smart_Config_CMT0 • Config_CMT0 • Config_PORT • Config_PORT • General • T_rbsp • Filer_rx • B rencipe • Filer_rx • B redeme.tx • B redeme.tx • B redeme.tx • B redeme.tx • B red_Configurator_Example ipcf • Output B Smart_Configurator_Example.map • B Smart_Configurator_Example.out Smart_Configurator_Example Smart_Configurator_Example	100	<pre>finclude <r_ether_rx_if.h> finclude <r_ether_rx_if.h> finclude <r_smc_entry.h3 (="" *="" cmt0="" counter="" int="" main(void)="" nop();="" operation="" pre="" r_contig_cmt0_start();="" start="" while(10)="" {="" }="" }<=""></r_smc_entry.h3></r_ether_rx_if.h></r_ether_rx_if.h></pre>	
augiterungeigeigeigei			

Figure 8-2 Call Code Generator functions

In case of FIT module, refer to the examples provided in the "API Functions" chapter of corresponding Application Note. You can find the Application Note in [doc] folder under each FIT module.



9. Generating Reports

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

9.1 Report on Configuration

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).

File Wind	ionfigurator ow Help onfigurator_Example.scfg × _ ew information						ি Generate Code	
	Information						Generate Code	oenerate Report
	Overview Get an <u>overview</u> of the feature	res provided	by Smart Configurator.					÷
	Videos Introduction to Smart Config Browse related videos	<u>gurator</u>			Application Code Software Components Middleware & Drivers	Smart Con		
	What's New Check out <u>what's new</u> in the See all <u>Release Notes</u> .				MCU Hardware	figurator		
	Product Documenta User's Guide API manual	ation						
	Configuration							
Generate	board/device: R5F564MLCxFC d location (PROJECT_LOC\): s components:		Mbytes, RAM size: 512KE	, Pin count: 176) Edit				
Compo		Version 7.42	Configuration r_bsp(used)					
Overview	Board Clocks System Compo	onents Pins Ir	iterrupts					

Figure 9-1 Output of a Report on the Configuration

💰 Smart Report	×
Generate report of configurations	
Options Print all sections Print specific sections 	
Board Clocks	
Components Pins	
Output as PDF	
C:\	Browse
	OK Cancel

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



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

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

rdware Resource 🗉 🖻 🛱	🖧 P	in Functi	on			२। 🖬 🔣 🔤
vpe filter text		type filte	r text (* = any	v string, ? = any character)	All	
🚣 All	^	Enabled	Function	Assignment	Pin Number	Direction
Clock generator		\checkmark	ADTRG0#	P07/IRQ15/ADTRG0#	/ 176	
Clock frequency accuracy measure		\checkmark	AN000	P40/IRQ8/AN000	/ 173	1
alia Buses		\checkmark	AN005	P45/IRQ13/AN005	/ 167	1
EXDMA controller		\checkmark	ET0_COL	PC7/UB/A23/CS0#/MTIOC3A/MTCLKB/TMO2/PO31/TOC0/CF	/ 76	1
Interrupt controller unit		\checkmark	ET0_CRS	P83/EDACK1/MTIOC4C/ET0_CRS/RMII0_CRS_DV/SCK10/SS10	/ 74	1
⁶ Multi-function timer pulse unit 3		\checkmark	ET0_ERXD0	P75/CS5#/PO20/ET0_ERXD0/RMII0_RXD0/SCK11/RTS11#/SC	/ 87	1
MTU0		\checkmark	ET0_ERXD1	P74/A20/CS4#/PO19/ET0_ERXD1/RMII0_RXD1/SS11#/CTS11	/ 88	1
MTU1		\checkmark	ET0_ERXD2	PC1/A17/MTIOC3A/TCLKD/PO18/ET0_ERXD2/SCK5/SSLA2-A	/ 89	1
MTU2		\checkmark	ET0_ERXD3	PC0/A16/MTIOC3C/TCLKC/PO17/ET0_ERXD3/CTS5#/RTS5#/S	/ 91	1
MTU3		\checkmark	ET0_ETXD0	P81/EDACK0/MTIOC3D/PO27/ET0_ETXD0/RMII0_TXD0/SMIS	/ 80	0
MTU4		\checkmark	ET0_ETXD1	P82/EDREQ1/MTIOC4A/PO28/ET0_ETXD1/RMII0_TXD1/SMO	/ 79	0
MTU5		\checkmark	ET0_ETXD2	PC5/D3/A21/CS2#/WAIT#/MTIOC3B/MTCLKD/TMRI2/PO29/{	/ 78	0
MTU6		\checkmark	ET0_ETXD3	PC6/D2/A22/CS1#/MTIOC3C/MTCLKA/TMCI2/PO30/TIC0/ET(/ 77	0
MTU7		\checkmark	ETO_LINKST/	P34/MTIOC0A/TMCI3/PO12/POE10#/ET0_LINKSTA/SCK6/SCk	/ 27	1
MTU8	~	\checkmark	FT0 MDC	P72/A19/CS2#/ET0 MDC/LCD DATA23-A	/ 101	0

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

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

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



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



10. User code protection feature for Smart Configurator Code Generation component

The Smart Configurator for RX V2.16.0 and the later version now incorporates an enhanced user code protection feature. This feature empowers users to insert codes to any location in the generated codes by utilizing the specific tags, as shown in Figure 10-1. After the next code generation, the inserted user codes will be protected and automatically merged into the generated files.

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

10.1 Specific tags for the user code protection feature

When using the user code protection feature, please insert /* Start user code */ and /* End user code */ as shown in Figure 10-1 and add the user codes between these tags. If the specific tags do not match exactly, the inserted user code will not be protected after the code generation.



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

Figure 10-2 shows an example of adding new user code into the Create API of CMT module by using the specific tags shown in Figure 10-1. After updating the configuration in the CMT GUI and re-generating the codes, the inserted user codes will be automatically merged into the new generated file.

<pre>void R_Config_CMT0_Create(void) {</pre>	<pre>void R_Config_CMT0_Create(void) {</pre>	
<pre>/* Disable CMI0 interrupt */ IEN(CMT0,CMI0) = 00;</pre>	<pre>/* Disable CMI0 interrupt */ IEN(CMT0,CMI0) = 00;</pre>	
/* Cancel CMT stop st MSTP(CMT0) = 0U; /* Set control regist	MSIP(CMI0) = 00	
CMT0.CMCR.WORD = _0001_CMT_CMCR_CLOCK_PCLK32	2 CMT0.CMCR.WORD = _0002_CMT_CMCR_CLOCK_PCLK128	
<pre>/* Start user code */ CMT0.CMCR.WORD = 0x80; /* End user code */</pre>	<pre>/* Start user code */ CMT0.CMCR.WORD = 0x80; /* End user code */</pre>	
<pre>/* Set compare match register */ CMT0.CMCOR = _0031_CMT0_CMCOR_VALUE;</pre>	<pre>/* Set compare match register */ CMT0.CMCOR = _000C_CMT0_CMCOR_VALUE;</pre>	
<pre>/* Set CMI0 priority level */ IPR(CMT0,CMI0) = _08_CMT_PRIORITY_LEVEL8;</pre>	<pre>/* Set CMI0 priority level */ IPR(CMT0,CMI0) = _08_CMT_PRIORITY_LEVEL8;</pre>	
<pre>R_Config_CMT0_Create_UserInit(); }</pre>	<pre>R_Config_CMT0_Create_UserInit(); }</pre>	

Figure 10-2 User code protection with auto merge



10.3 What to do when merge conflict occurs

10.3.1 What is Merge conflict

When the lines of generated codes before and after the inserted user codes are updated due to changes in GUI configuration or the version update of Smart Configurator, merge conflict codes will be generated out, as shown in Figure 10-3.



Figure 10-3 User code protection with merge conflict

If the merge conflict occurs, conflict message will be displayed in the Smart Configurator console, as shown in Figure 10-4.

	- 0
mart Configurator Output	
WUUUUUUI: Lode generation is started	~
104000001: File generated:src\smc gen\Config TMR0\Config TMR0.h	
104000001: File generated:src\smc_gen\Config_TMR0\Config_TMR0.c	
100000002: Code generation is successful: <u>C:\Users\a5085102\smartconfigurator\workspace\src\smc_gen</u>	
10000001: Code generation is started	
104000001: File generated: <u>src\smc_gen\Config_TMR0\Config_TMR0.h</u>	
104000001: File generated:src\smc_gen\Config_TMR0\Config_TMR0.c	
100000005: The above files highlighted in red color have user code merge conflicts, please open the file and resolve the conflict manuall	ly 👘
100000002: Code generation is successful:C:\Users\a5085102\smartconfigurator\workspace\src\smc_gen	
	\sim

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



10.3.2 Steps for resolving the merge conflict

To resolve this merge conflict, open the highlighted conflict files and follow the steps below to solve the merge conflicts manually.

1) Copy the user code from "Last Time Generated Code" and paste it into the new position in "This Time Generated Code" as shown in Figure 10-5.



2) Remove last time generated code and the conflicts commend (<<<<<Last Time Generated Code, ====== and >>>>>This Time Generated Code) as shown in Figure 10-6.



Figure 10-6 The codes after resolving the merge conflict

Another way to solve merge conflict:

1) Click this console message to open the compare view.

Software componen	t configuration					🕤 😅 Generate Code Generate Report
Components	▲山小日日⇒ ▼	Configure				٥
type filter text	\$ T	Count clock setting PCLK/8 PCLK/32	O PCLK/128	O PCLK/512		
 Startup Generic r_bsp Drivers Timers 		Compare match setting Interval value Register value (CMCOR) Compare match interrupt (CMI0) Enable multiple interrupts (CMI0)	100 399	μs	✓ (Actual value: 100)	
Config_CMT0		Priority	Level 15 (highest)	~		
	em Components Pins Interrup	pts				
Console ×						
104000001: File gener	ated:src\smc_gen\Confi ated:src\smc_gen\Confi	ig_CMT0\Config_CMT0.c				^
		red color have user code merge cor \Users\a5146853\OneDrive - Renesa				

Figure 10-7 Error message in console



RX Smart Configurator

2) After compare view is opened, user can apply left change to the right. Or user can edit right side code manually.

5° File Compare ×			- 0
V Text Compare		M 📰 🛣 🗑 🖗 🗄 🕯	9.45 5
Existing code	٠	New code	
<pre>void R_Config_CMT0_Start(void) { /* Enable CMI0 interrupt in ICU */ IEN(CMT0,CMI0) = 1U;</pre>		<pre>void R_Config_CMT0_Start(void) { /* Enable CMI0 interrupt in ICU */ IEN(CMT0,CMI0) = 10;</pre>	** ^
/* Start user code */ CMT.CMSTR0.BII.STR0 = 0U; /* End user code */	¢	/* Start CMT0 count */ CMT.CMSTR0.BIT.STR0 = 1U; }]
<pre>} //* Function Name: R_Config_CMT0_Stop * Description : This function stop the CMT0 channel counter * Arguments : None * Return Value : None void R_Config_CMT0_Stop(void)</pre>		<pre>* Function Name: R_Config_CMT0_Stop * Description : This function stop the CMT0 channel counter * Arguments : None * Return Value : None void R_Config_CMT0_Stop(void) { /* Stop CMT0 count */ CMT.CMSTR0.BIT.STR0 = 0U;</pre>	•
{ /* Stop CNT0 count */		<pre>/* Disable CMIB interrupt in ICU */ IEN(CMT0,CMI0) = 00; /* CMID: type = 00;</pre>	.
© Console ×			- -
Meddedeligurator Output Meddeded1: Code generation is started Meddeded1: File generated: <u>src\smc_gen\Config_CMT0\Config_CMT0.h</u> Meddeded1: File generated: <u>src\smc_gen\Config_CMT0\Config_CMT0.c</u> Meddeded01: File generated: <u>src\smc_gen\Config_CMT0.c</u> Meddeded02: The above files highlighted in red color have user code merge conflicts, pl Meddeded02: Code generation is successful:C:\Users\a5146853\OneDrive - Renesas Electron			
<			>

Figure 10-8 Compare Viewer for conflict code

11. Help

11.1 Help

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

💰 Smart Configurator				
File	Window	Help		
i 🖆 I		?	Help Contents	
			Home Page	
			Release Notes	
			Tool News	
			API Manual	
		ß	About	

Figure 11-1 Help Menu



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

Smart Configurator File Window Help				- 0	×
🖆 🗁 🔤			- 0	Di Help × 수수 응	
Overview information			Generate Code Generate Report	🖻 Contents 🏁 Search 📽 Related Topics 🎟 Bookmarks 📟 Index	
General Information Overview				mart Configurator Smart Configurator is a User Interface that combines the functionalities of Code Generator and FIT Configurator which imports, configures and generates different types of drivers and	a î
Get an <u>overview</u> of the feat Configurator.	ures provided	by Smart		middleware modules.	
Videos		Application Code		Smart Configurator	
Introduction to Smart Conf	igurator	Software Components		Creating a Smart Configurator project	
Browse related videos		Middleware & Drivers	G Smart Configurator	Using the Smart Configurator Editor	
😭 What's New		RTOS Device Drivers	figu	Importing a Software Component	
Check out what's new in the	e latest release		rator	Configuration of Software Component	
See all <u>Release Notes</u> .		MCU Hardware		Configuring Pins	
Product Document	ation			Configuring Interrupts	
User's Guide				Code Generation	
API manual				Report Generation	
 Current Configuration 				B MCU Migration	
		Mbytes, RAM size: 512KB, Pin count: 176)		FreeRTOS Project Generation	
Generated location (PROJECT_LOC\):	src\smc_gen	Edit		FreeRTOS Kernel	
Selected components:				FreeRTOS Object	
Component Board Support Packages. (r bsp)	Version 7.42	Configuration r bsp(used)		AWS Libraries	
Board Support Packages. (r_bsp)	1.42	r_bsp(used)		B Heap Size Estimation	
				Backup Data	
				MCU Package View	
				Azure RTOS Project Creation	
Overview Board Clocks System Comp	onents Pins I	nterrupts		Azure RTOS Project Configuration	~

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 web site of Renesas Electronics.

Technical Update/Technical News

Obtain the latest information from the web site of Renesas Electronics.

User's Manual: Development Environment

Obtain the latest version of the manual from each company web site.



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

		Description	on
Rev.	Date	Page	Summary
1.00	Jan 25, 2019	-	First edition issued
1.10	Sep 09, 2019	4	1.2 Features updated, 1.3 Software Components added
		8	3.3.1 Downloading FIT modules added
			3.3.2 Creating a New Configuration File updated
		12	3.4.1 Main menu updated
		29	
		- 00	Adding FIT drivers or middleware added
		30	4.4.7 Setting of the FIT software components added
		31	4.4.8 Changing the version of the FIT software components added
		31	4.4.9 Solving the grey-out component added
			4.4.10 "i" mark on FIT modules icon added
		32	
		33 - 34	4.4.11 Configure Analog Front End component added
		35 – 37	4.4.12 Configure Motor Component added
		38	4.4.13 Configure general setting of component added
		45	7.1 Adding Custom Code of FIT added
		50	11.1 Help updated
1.20	Jun 20, 2021	All	Smart Configurator images updated
		13	3.4.4 MCU/MPU Package view updated
		18	4.3 System Settings added
		30	4.4.9 Solving the grey-out component added
		31	4.4.10 "i" mark on FIT modules icon added
		32	4.4.11 Configure Analog Front End component added
	34	4.4.12 Configure Motor Component added	
	37	4.4.13 Configure general setting of component updated	
		40	4.5.2 Pin setting using MCU/MPU package updated
		41	4.5.3 Show pin number from pin functions added
		44	4.5.8 Pin Errors/Warnings setting added
		57	8.2 Using Generated Code in user application added
		59	9.3 Image of MCU/MPU Package (in png Format) updated
4.00	10,0000	60	10 Help updated
1.30	Apr 16,2023	<u>1</u>	Updated the URL for RX Smart Configurator
		5	2.1 Updated the URL for RX Smart Configurator
		8 - 9 39 - 41	3.3.2 Updated the steps to create a New Configuration File
		42	4.4.13 Updated the general setting of component 4.4.14 Added the export configuration of component
		42	4.4.15 Added the Import configuration of component
		55 - 57	6.2 Update the file structure and added explaination for the
			table
		62 - 64	7.2 Update the flow of loading in IAR Embedded Workbench
		65	7.3 Added the build of IAR project file
		71 - 73	10 Added new chapter 10 User code protection feature for
			Smart Configurator Code Generation Component
1.40	Apr 16, 2024	-	Figures are updated to RX Smart Configurator V2.20
		14	3.4.4 MCU/MPU Package View updated
		23	4.4.2 Removing a component updated
		37 - 40	4.4.12 Configure Motor Component updated
		41	4.4.13 Configure general setting of component updated
		52	4.5.9 Symbolic name setting added
		56	4.6.3 Multiple interrupts setting added
		79	10.3.2 Steps for resolving the merge conflict updated.

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

2. Processing at power-on

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

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

6. Voltage application waveform at input pin

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

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

Notice

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

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

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

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

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

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/.

(Rev.5.0-1 October 2020)