

User's Guide: e² studio

RX Smart Configurator

R20AN0451ES0160 Rev.1.60 Apr 16, 2024

Introduction

This application note describes the basic usage of the RX Smart Configurator (hereafter called the Smart Configurator), which is an e² studio plug-in tool.

References to the e² studio integrated development environment in this application note apply to the following versions.

• e² studio 2024-01 and later

Target Devices and Compilers

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

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

Contents

1. C	Dverview	4
1.1	Purpose	4
1.2	Features	4
1.3	Software Components	4
2. C	Creating a Project	5
2.1	Create a project not using RTOS	5
2.2	Create a RTOS project	10
2.3	Create a Blinky project	16
3. C	Dperating the Smart Configurator	21
3.1	Displaying the Smart Configurator Perspective	
3.2	Procedure for Operations	
3.3	File to be Saved as Project Information	
3.4	Window	
3.4	4.1 Project Explorer	
3.4	4.2 Smart Configurator view	
3.4	4.3 MCU/MPU Package view	
3.4	4.4 Console view	
3.4	4.5 Configuration Problems view	27
4. S	Setting of Peripheral Modules	28
4.1	Board Settings	
4.1	1.1 Selecting the board	
4.1	1.2 Exporting board settings	
4.1	1.3 Importing board settings	29



4.2	Clo	ck Settings	. 30
4.3	Sys	stem Settings	. 31
4.4	Cor	mponent Settings	. 32
4.	4.1	Adding Code Generator components	. 32
4.	4.2	Removing a software component	. 34
4.	4.3	Switching between the component view and hardware view	. 35
4.	4.4	Setting a CG driver	. 36
4.	4.5	Changing the resource for a CG configuration	. 37
4.	4.6	Downloading a FIT module	. 40
4.	4.7	Adding FIT drivers or middleware	. 42
4.	4.8	Download and import FIT sample project	. 43
4.	4.9	Setting a FIT Software Component	. 46
4.	4.10	Version change of FIT software component	. 47
4.	4.11	Solving the grey-out component	. 49
4.	4.12	Setting the RTOS Kernel	. 50
4.	4.13	Creating the RTOS Object	. 51
4.	4.14	Setting the RTOS Library	. 52
4.	4.15	Configure Analog Front End component	. 53
4.	4.16	Configure Motor Component	. 55
4.	4.17	Configure general setting of component	. 59
4.5	Pin	Settings	. 60
4.	5.1	Changing the pin assignment of a software component	. 61
4.	5.2	Assigning pins using the MCU/MPU Package view	. 62
4.	5.3	Show pin number from pin functions	
4.	5.4	Exporting pin settings	
4.	5.5	Importing pin settings	
4.	5.6	Pin setting using board pin configuration information	
4.	5.7	Pin filter feature	
4.	5.8	Pin Errors/Warnings setting	. 67
4.	5.9	Symbolic name setting	
4.6		errupt Settings	
4.	6.1	Changing the interrupt priority level and fast interrupt setting	
4.	6.2	Changing the interrupt vector number	
	6.3	Multiple interrupts setting	
4.7		U migration feature	
5. I	Mana	iging Conflicts	.78
5.1	Res	source Conflicts	. 78
5.2	Res	solving pin conflicts	. 79
6. (Gene	rating Source Code	. 80
6.1	Out	tputting Generated Source Code	. 80
6.2		ange Generated Code Location	
6.3		nfiguration of Generated Files and File Names	
6.4		alizing Clocks	
6.5	Initi	alizing Pins	. 90
6.6	Initi	alizing Interrupts	. 93



6.7 Component Settings 6.7.1 FIT module configuration	
6.7.2 FreeRTOS Kernel configuration	. 96
7. Creating User Programs	. 97
7.1 Adding Custom Code in the Case of Firmware Integration Technology (FIT)	. 97
7.2 Adding Custom Code in the Case of Code Generator	. 98
7.3 Using Generated Code in user application	100
8. Backing up Generated Source Code1	101
9. Generating Reports1	102
9.1 Report on All Configurations	102
9.2 Configuration of Pin Function List and Pin Number List (in csv Format)	103
9.3 Image of MCU/MPU Package (in png Format)	103
10. User code protection feature for Smart Configurator Code Generation component 1	104
10.1 Specific tags for the user code protection feature	104
10.2 Examples of using user code protection feature to add new user code	
10.3 What to do when merge conflict occurs	105
10.3.1 What is Merge conflict	105
10.3.2 Steps for resolving the merge conflict	106
11. Help1	108
11.1 Help	108
11.2 Developer assistance	109
12. Documents for Reference1	111
Website and Support1	112



1. Overview

1.1 Purpose

This application note describes the basic usage of the Smart Configurator and the e² studio integrated development environment, including the procedure for creating a project.

Refer to the User's Manual of the e^2 studio for how to use the e^2 studio.

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 FIT (Firmware Integration Technology) 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:

- 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. Creating a Project

The following describes the procedure for creating a C/C++ project using the Smart Configurator. Refer to the related documents on the e^2 studio for the details of the e^2 studio project creation wizard.

2.1 Create a project not using RTOS

The following describes the procedure for creating a project not using RTOS.

(1) Start e² studio and launch a workspace. After starting, select [File] □ [New] □ [Renesas C/C++ Project] □ [Renesas RX] to activate the project creation wizard.

	² studio Edit Source Refactor Navigate	Search Project Re	enesas Views Run Window Help	
	New	Alt+Shift+N >	Renesas C/C++ Project >	Renesas Debug
۵,	Open File Open Projects from File System Recent Files	>	 Synergy C/C++ Project Makefile Project with Existing Code C/C++ Project 	Renesas RA Renesas RE Renesas RL78
	Close Editor Close All Editors Save Save As	Ctrl+W Ctrl+Shift+W Ctrl+S	 Project Convert to a C/C++ Project (Adds C/C++ Nature) Source Folder Folder Source File 	Renesas RX Renesas RZ Renesas Synergy

Figure 2-1 Creating a New Project

(2) In the project creation wizard, select [Renesas CC-RX C/C++ Executable Project] or [GCC for Renesas RX C/C++ Executable Project], and click on the [Next] button.

📴 New C/C+	+ Project – 🗆 🗙
Templates f	for Renesas RX Project
All	
All C/C++	GCC for Renesas RX C/C++ Executable Project A C/C++ Executable Project for Renesas RX using the GCC for Renesas RX Toolchain.
	GCC for Renesas RX C/C++ Library Project A C/C++ Library Project for Renesas RX using the GCC for Renesas RX Toolchain.
	Renesas CC-RX C/C++ Executable Project
	Renesas CC-RX C/C++ Library Project A C/C++ Library Project for Renesas RX using the Renesas CCRX toolchain.
?	< Back Next > Finish Cancel

Figure 2-2 Templates for New C/C++ Project Dialog Box

(3) Enter project information. Click on the [Next] button to continue.

(for e.g. CC-RX executable project, Project name: "Smart_Configurator_Example")

www.Renesas CC-RX Executable Project lew Renesas CC-RX Executable Project roject name: Smart_Configurator_Example Q Use default location ocation: C:\Users\app\e2_studio\workspace\Smart_Configurator_Example Browse C:\Users\app\e2_studio\workspace\Smart_Configurator_Example Browse C:\Users\app\e2_studio\workspace\Smart_Configurator_Example Browse C:\users\app\e2_studio\workspace\Smart_Configurator_Example	2	-	_	п	×
lew Renesas CC-RX Executable Project roject name: Smart_Configurator_Example Use default location ocation: C(\Users\app\e2_studio\workspace\Smart_Configurator_Example Browse Create Directory for Project Choose file system: default Working sets Add project to working sets New					í
roject name: Smart_Configurator_Example Use default location C(\Users\app\e2_studio\workspace\Smart_Configurator_Example Create Directory for Project Working sets Add project to working sets New		-			•
Use default location ocation: C:\Users\app\e2_studio\workspace\Smart_Configurator_Example Create Directory for Project thoose file system: default Working sets Add project to working sets New	New Renesas CC-	X Executable Project			4
C:\Users\app\e2_studio\workspace\Smart_Configurator_Example Browse C:Ceate Directory for Project choose file system: default Working sets Add project to working sets New	Project name: Sr	nart_Configurator_Example			
C Create Directory for Project Choose file system: default Working sets Add project to working sets New	Use default lo	cation			
ihoose file system: default v Working sets Add project to working sets New	ocation:	C:\Users\app\e2_studio\workspace\Smart_Configurator_Example		Browse.	
Working sets New		Create Directory for Project			
Add project to working sets New	Choose file system	n: default \vee			
	Working sets				
	Add project t	p working sets		New	
working sets: Select		-		Coloct	
	working sets:	~		Delection	
	2	< Back Next > Finish		Cance	a

Figure 2-3 Creating a New Renesas CC-RX Executable Project

(4) Select C or C++, toolchain, board, device, and debug configuration. Keep RTOS as "None". Click [Next].

(for e.g. Target Board: RSKRX64M)

Note: By selecting a board, the following settings is auto selected: Initial pin setting, Clock Frequency, Target device



0	– 🗆 X
New Renesas CC-RX Executable Project	
Select toolchain, device & debug settings	
Toolchain Settings Language: OC C++	
Toolchain: Renesas CCRX Toolchain Version: v3.02.00	
Manage Toolchains RTOS: None ~	
RTOS Version:	
Device Settings	Configurations
Target Board: RSKRX64M ~	Create Hardware Debug Configuration
Download additional boards Target Device: R5F564MLCxFC Unlock Devices	E2 Lite (RX) ~ Create Debug Configuration RX Simulator ~
Endian: Little 🗸	
Project Type: Default	Create Release Configuration
⑦ < <u>B</u> ack <u>N</u> ext >	<u>F</u> inish Cancel

Figure 2-4 Selecting the Toolchain, Device, and Debug Configuration



RX Smart Configurator

- (5) In the [Select Coding Assistant settings] dialog box, select the [Smart Configurator] checkbox and click on the [Finish] button.
 - Note: [Smart Configurator] checkbox is enabled only if device supported by Smart Configurator is selected at (4).

8	– D X
New Renesas CC-RX Executable Project Select Coding Assistant settings	Ď
Use Smart Configurator Use Peripheral Code Generator	
Smart Configurator is a single User Interface that combines the functionalities of Code of which imports, configures and generates different types of drivers and middleware mo- Smart Configurator encompasses unified clock configuration view, interrupt configurati Hardware resources conflict in peripheral modules, interrupts and pins occurred in differ middleware modules will be notified. (Smart Configurator is available only for the supported devices)	dules. ion view and pin configuration view.
Application Code Software Components Middleware & Drivers Device Drivers	P
MCU Hardware	
? < Back Next >	Finish Cancel

Figure 2-5 Selecting the Coding Assistant Tool

(6) Wait for completion of project creation.

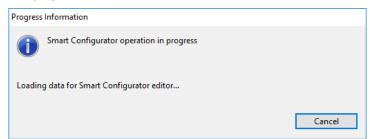


Figure 2-6 Progress of Project creation



(7) After a new C/C++ Project is successfully created, the project will be opened in the Smart Configurator perspective.

workspace - Smart_Configurator_Example/Smart_Example/Smart_Exampl	nple.scfg - e ^z studio							- 0	×
File Edit Source Refactor Source Navigate Search Project Renesa	s Views Run Window Help								
□ ● • ≪ • ○○○ ★ • ▲ • ○○ *							Q 18 BC/C++	🖉 🧟 Smart Configu	irato
🖉 陷 Project Explorer 🗡 🛛 🖹 🖻 🖯	Smart_Configurator_Example.scfg ×			- <i>B</i>	MCU/MPU Package	e ×			= E
	Overview information		Generate Code	Generate Report	G A P	<u>/ 8 8</u>	Type filter text		>
> 🥝 New_Folder	✓ General Information			(7) ^					
¹ →									
> 🗁 trash	Overview								
		Smart							
⑦ Developer Assistance	Contigurator.	0			1				
	Videos	Арр	lication Code					de la su	
	Introduction to Smart Configurator		E .						
	Browse related videos	Softwa	are Components						
	140.41.51		Middleware &						
	CON CON	RTOS		-					
	Check out what's new in the latest release.		Device Drivers			20			
	Pred: Opported Sample Arg Set: Configurator, Sample Kardesor Dolog Jawah Net: Configurator, Sample Kardesor, Jawah Net: Configurator, Jaw								
	Product Documentation	MC	U Hardware						
	User's Guide	the second se					-		
	API manual						KA04M		
	heyes Search Project Reneas Views Ran Window Heje								
	Tool news		Application Code Software Components Device Drives MCUJ Hardware 28. Pn count: 176 Configuration (,begicated) Extenses (configuration (begicated) (configuration (begicated) (configuration (begicated) (configuration (begicated) (configuration (begicated) (configuration (begicated) (configuration (begicated) (configuration (configuration (begicated) (configuration (conf						
	✓ Current Configuration					, Vianalaa aaxant	ASSA & RAARA AR ARRA		
	Selected board/device: R5F564MLCxFC (ROM size: 4Mbyte:	s, RAM size: 512KB, Pin cou	nt: 176)						
	Generated location (PROJECT_LOC\): New_Folder		Edit						
	Selected components:								
	A CONTRACTOR OF	Version Confi	outation						
				1	▶ Legend				
Console X	ii. 51 (9) 🖻	0 - P - D 1	Problems ×					78	- 6
Smart Configurator Output									
		^ D	escription		Resource	Path	Location	Туре	
c		3							

Figure 2-7 Smart Configurator Perspective



2.2 Create a RTOS project

The following describes the procedure for creating a RTOS project.

Note: For FreeRTOS project supported devices and Renesas FreeRTOS, refer to Renesas FreeRTOS related documentation ("chapter 12, Documents for Reference").

(1) Start e² studio and launch a workspace. After starting, select [File] □ [New] □ [Renesas C/C++ Project] □ [Renesas RX] to activate the project creation wizard.

	² studio Edit Source Refactor Navigate S	Search Project Re	enesas Views Run Window Help	
	New	Alt+Shift+N >	Renesas C/C++ Project >	Renesas Debug
۵,	Open File Open Projects from File System Recent Files	>	 Synergy C/C++ Project Makefile Project with Existing Code C/C++ Project 	Renesas RA Renesas RE Renesas RL78
	Close Editor Close All Editors Save	Ctrl+W Ctrl+Shift+W Ctrl+S	 Project Convert to a C/C++ Project (Adds C/C++ Nature) Source Folder Folder 	Renesas RX Renesas RZ Renesas Synergy
	Save As		Source File	

Figure 2-8 Creating a New Project

(2) In the project creation wizard, select [Renesas CC-RX C/C++ Executable Project] or [GCC for Renesas RX C/C++ Executable Project] and click on the [Next] button.

📴 New C/C+	+ Project – 🗆 🗙
Templates f	for Renesas RX Project
All C/C++	GCC for Renesas RX C/C++ Executable Project A C/C++ Executable Project for Renesas RX using the GCC for Renesas RX Toolchain.
	GCC for Renesas RX C/C++ Library Project A C/C++ Library Project for Renesas RX using the GCC for Renesas RX Toolchain.
	Renesas CC-RX C/C++ Executable Project
	Renesas CC-RX C/C++ Library Project A C/C++ Library Project for Renesas RX using the Renesas CCRX toolchain.
?	< Back Next > Finish Cancel

Figure 2-9 Templates for New C/C++ Project Dialog Box

(3) Enter project information. Click on the [Next] button to continue.

(for e.g. CC-RX executable project, Project name: "Smart_Configurator_Example")

e ²		—		×
	RX Executable Project RX Executable Project			\$
Project name:	nart_Configurator_Example			
✓ Use default lo	cation			
location:	C:\Users\app\e2_studio\workspace\Smart_Configurator_Example		Browse.	
	Create Directory for Project			
Choose file syster				
Working sets				
-				_
Add project 1	o working sets		New	
Working sets:			Select	
?	< Back Next > Finish		Cance	1

Figure 2-10 Creating a New Renesas CC-RX Executable Project

- (4) Select the toolchain and RTOS configuration. (for e.g. RTOS: FreeRTOS (with IoT libraries))
 - FreeRTOS (kernel only): create a project with FreeRTOS (kernel only).
 - FreeRTOS (with IoT libraries): create a project with FreeRTOS (with IoT libraries).
 - Azure RTOS : create a project with Azure RTOS (whole package)
 - FreeRTOS (with IoT libraries)(deprecated structure): create a project with FreeRTOS IoT libraries (old structure of AWS).
 - RI600V4: create a project with RI600V4 Real-time OS. You should install RI600V4 by yourself before this procedure: <u>https://www.renesas.com/us/en/software-tool/ri600v4-real-time-os-rx-family</u>



If FreeRTOS or Azure RTOS has not been downloaded, go to procedure (5).

If FreeRTOS or Azure RTOS has been downloaded, go to procedure (7).

Toolchain Setti	ngs		
Language:	● C ○ C++		
Toolchain:	Renesas CC-RX ~		
Toolchain Versi	on: v3.05.00 ~		
	Manage Toolchains		
RTOS:	None		
RTOS Version:	None Azure RTOS		
	FreeRTOS (kernel only)		
Device Setting	FreeRTOS (with IoT libraries) FreeRTOS (with IoT libraries)(deprecated structure)	Configurations	
Target Board:	CuRI600V4	Create Hardware Deb	ug Configurat
	Download additional boards	E1 (RX)	
Target Device:	R5F564MLCxFC	Create Debug Configu	
	Unlock Devices	RX Simulator	uration
Endian:	Little ~	KX Simulator	
Project Type:	Default ~	Create Release Config	uration

Figure 2-11 Selecting the Toolchain and RTOS Configuration

(5) In the [RTOS Module Download] dialog, select the RTOS package and click [Download].

C				\Box \times
	S Module Download ect RTOS modules for download and	specify download location		Ľ
	Title	Rev.	Issue date	Select All
	FreeRTOS Kernel	10.4.3-rx-1.0.8		Deselect All
	FreeRTOS Kernel	10.4.3-rx-1.0.7		Deselect All
	FreeRTOS Kernel	10.4.3-rx-1.0.6		
	FreeRTOS Kernel	10.4.3-rx-1.0.5		
	FreeRTOS Kernel	10.4.3-rx-1.0.4		
	FreeRTOS Kernel	10.4.3-rx-1.0.3		
	FreeRTOS Kernel	10.4.3-rx-1.0.2		
	FreeRTOS Kernel	10.4.3-rx-1.0.1		
	FreeRTOS Kernel	10.4.3-rx-1.0.0		
<			2	•
Pleas	e refer to <u>GitHub</u> for more detailed in	formation of each release		
Mo	dule Folder Path:			
	C:\Renesas\.eclipse\com.renesas.plat	tform_download\RTOS		Browse
		(Download	Cancel

Figure 2-12 [RTOS Module Download] dialog



(6) When [End User License Agreement] dialog is displayed, click [Agree].

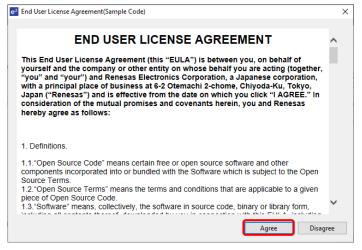


Figure 2-13 [End User License Agreement] dialog

(7) Select the board, device and debug configuration. Only device supported by the RTOS package can be selected. Click on the [Next] button

(for e.g. Target Board: RSKRX64M)

Toolchain Setting	s	
Language:	● C ○ C++	
Toolchain:	Renesas CCRX ~	
Toolchain Version	: v3.02.00 ~	
	Manage Toolchains	
RTOS:	FreeRTOS (kernel only)	
RTOS Version:	v10.0.04	
	Manage RTOS Versions	
Device Settings		Configurations
Target Board: RS	SKRX64M ~	Create Hardware Debug Configuration
	Download additional boards	E1 (RX)
Target Device: R5		Create Debug Configuration
	Unlock Devices	RX Simulator
Endian: Lit		Create Release Configuration
Project Type: De	efault 🗸 🗸	5

Figure 2-14 Selecting the Device and Debug Configuration

(8) In the [Select Coding Assistant settings] dialog box, select the [Smart Configurator] checkbox and click on the [Finish] button.

8			×
New Renesas CC-RX Executable Project Select Coding Assistant settings			3
Suse Smart Configurator Use Peripheral Code Generator The State of Code Code Generator The State of Code Code Generator The State of Code Code Code Generator The State of Code Code Generator The State of Code Code Code Code Code Code Code Code	and pin con	figuratio	
MCU Hardware			
? < Back Next > Finish		Cancel	

Figure 2-15 Select the Coding Assistant Tool

(9) Wait for completion of project creation.

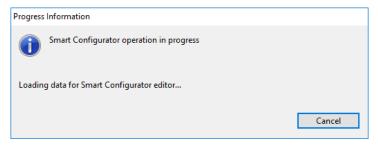


Figure 2-16 Progress of Project creation



(10) After a new C Project is successfully created, the project will be opened in the Smart Configurator perspective.

B • • • • • • • • • • • • • • • • • • •	@ Smort_Configurator_Example.sdg ×					- 0	MCUMPU Package × Developer Assist Browser	Q B CC++ Smat Configurator
> Ed Smart_Configure_pla	Overview information					Generate Code Generate Report	RAZE BB Apetherist	Assigned Function ·
	General Information					۲		
	Coverview Get as gandings the features parent Coverview Coverview	ate.		Application Composition Compos	Smart Configurator Bives are	>	RENES	545
	Galited component:	Version 7.41	Configuration (Josphered)					
	Overview Board Clocks System Components Pins	betremants					▶ Legend	
Costole X								9100
Smart Configurator Output	r dente este sus este este entre) iterns			
Smort Configuration During M020000051 File generated sections an M040000051 File generated sections an M04000051 File generated sections and M05000052 File generated sections and M05000052 File generated sections and M05000052 File generated sections and	<pre>algenerally_set_andware_setup.c nlgenerally_set_entry.h signerally_cg_interrupt_menters.h signerally_fin.h signerally_fin.set_interrupt.c</pre>				Description	749	e	

Figure 2-17 Smart Configurator Perspective



2.3 Create a Blinky project.

The following describes the procedure for creating a project using sample project.

(1) Start e² studio and launch a workspace. After starting, select [File] □ [New] □ [Renesas C/C++ Project] □ [Renesas RX] to activate the project creation wizard.

	² studio Edit Source Refactor Navigate Sea	rch Project Re	nesas Views Run Window Help	
	New	Alt+Shift+N >	Renesas C/C++ Project >	Renesas Debug
	Open File		Synergy C/C++ Project	Renesas RA
۵,	Open Projects from File System		Makefile Project with Existing Code	Renesas RE
	Recent Files	>	C/C++ Project	Renesas RL78
	Close Editor	Ctrl+W	Project	Renesas RX
	Close All Editors C	trl+Shift+W	Convert to a C/C++ Project (Adds C/C++ Nature)	Renesas RZ
	Save	Ctrl+S	 Source Folder Folder 	Renesas Synergy
	Save As		Source File	

Figure 2-18 Creating a New Project

(2) In the project creation wizard, select [Renesas CC-RX C/C++ Executable Project] or [GCC for Renesas RX C/C++ Executable Project] and click on the [Next] button.

圆 New C/C++ Proj	ject — 🗆 🗙
Templates for Re	enesas RX Project
4	
All C/C++	GCC for Renesas RX C/C++ Executable Project A C/C++ Executable Project for Renesas RX using the GCC for Renesas RX Toolchain.
	GCC for Renesas RX C/C++ Library Project A C/C++ Library Project for Renesas RX using the GCC for Renesas RX Toolchain.
	Renesas CC-RX C/C++ Executable Project
-	Renesas CC-RX C/C++ Library Project
?	< Back Next > Finish Cancel

Figure 2-19 Templates for blinky Project Dialog Box

(3) Enter project information. Click on the [Next] button to continue.

8			×
New Renesas CC-	RX Executable Project		4
New Renesas CC-R	X Executable Project	-	4
Project name: Sar	nple_Blinky_Project		
Use default loc	ation		
Location:	C:\Renesas\Sample_project\Sample_Blinky_Project	Browse	
	Create Directory for Project		
Choose file system	default 🗠		
Working sets			
Add project to	o working sets	New	
Working sets:	~	Select	
? < B	ack Next > Finish	Cancel	

Figure 2-20 Name for blinky Project.

(4) Select C or C++, toolchain, board, device, and debug configuration. Keep RTOS as "None". Click [Next]

Note: Blinky project is only available in Renesas boards. (for e.g. Target Board: CKRX65N)

	5	
Language:	● C ○ C++	
Toolchain:	Renesas CC-RX	
Toolchain Version	v3.05.00	
	Manage	
RTOS:	None	
RTOS Version:		
Device Settings		Configurations
Target Board: C	K-RX65N ~	Create Hardware Debug Configuration
Dov	mload additional boards	E2 Lite (RX) ~
Target Device: R		Create Debug Configuration
	Unlock Devices	RX Simulator ~
Endian: Li	ttle 🗸 🗸	
	efault. 👻	Create Release Configuration

Figure 2-21 Selecting the Toolchain, Device, and Debug Configuration



(5) In the [Select Coding Assistant settings] dialog box, select the [Smart Configurator] checkbox and click on the [Next] button.

Select Coding As		able Project s			
Use Smart Co	nfigurator				
Use Periphera	Code General	tor ^a			
drivers and middl smart Configurate riew and pin conf lardware resourc ypes of drivers a	eware module: or encompasse iguration view. es conflict in p nd middleware	es unified clock conf	iguration view, i nterrupts and pi tified.	nterrupt confi	guration
		cation Code e Components Middleware & Drivers Device Drivers	Smart Configurator	Ŷ	
	MCU	l Hardware	. ⁹		

Figure 2-22 Selecting the Coding Assistant Tool

(6) In the [Project template selection] page, select the [Bare Metal - Blinky] checkbox and click on the [Finish] button.

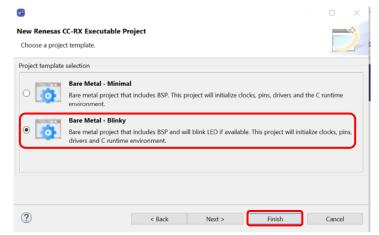


Figure 2-23 Selecting the project template

(7) After the Project is successfully created, the project will be opened in the Smart Configurator perspective.

File Edit Navigate Search Project F	•	Q. 🔡 🖽 C/C++ 🧟 Smart Configurator
Project Explorer × □	Sample Blinky Project.scfg ×	C C MCU/MPU Package X
► 😵 🏹 🖇 ❤ 😂 Sample_Blinky_Project	Overview information	Generate Code Generate Report
> 🔊 Includes		0 ^
 Sample_Blinky_Project.scfg Sample_Blinky_Project Hardw Oreveloper Assistance 	Get an accepting of the features provided by Smart Configurator. Videos Introduction to Smart Configurator Browse related videos What's New Check out what's new in the latest release. Product Documentation Inter menual and release netse	Dilication Code vare Components Middleware & Device Drivers CU Hardware
< >>	Overview Board Clocks System Components Pins Interrupts	► Legend
Console ×	🔍 🚮 🐼 🛫 🗂 🗸 🗂 🗖	1 🔛 Configuration Problems × 😗 🖇 🗖 🗖
Smart Configurator Output		0 items
		Description Type

Figure 2-24 Smart Configurator Perspective

(8) Go to the component page, existing components and setting can be found.

oftware component	configuration		😼 Generate Code Gene	ate Report
om 🚵 🛃 🖓 🖂 🕀 茸	Configure			^
🐮 ᢏ	Port selection PORT	2 (3)		
 ✓	PORTO	PORT1		
Cosp Drivers Cosp Cosp	PORT2	PORT3		- 1
Config_PORT	PORT4	PORT5		
(2)	PORT6	PORT7		
	PORT8	PORT9		
(1		PORTB		~
	k			>

Figure 2-25 Component setting(1)

oftware component config	juluton				Ge	merate Cod	de Generate Re	ро
Compone 🎦 🛃 🖓 🔁 🖽 🛛	Unused GPIO	$\bigcirc \ln$	Out	Pull-up	CMOS output	v	Output 1	
type filter text	P24							
✓ 🤭 Startup	Unused GPIO	() In	Out	Pull-up	CMOS output	~	Output 1	
 ✓ ఊ Generic er Lbsp ✓ ఊ Drivers 	P25 O Unused GPIO	() In	 Out 	Pull-up	CMOS output	Ŷ	Output 1	
Config_PORT	P26 Unused GPIO	◯In	() Out	Pull-up	CMOS output	Ŷ	Output 1	
	P27 Onused GPIO	() In	Out	Pull-up	CMOS output	~	Output 1	
					_		17	>

Figure 2-26 Component setting(2)



(9) Click [src] folder and double click [Sample_Blinky_Project.c] file in Project Explore to open the main function. Sample code will be displayed.



Figure 2-27 Main function

(10) By connecting computer with target board, user can build this project and debug it on target board to toggle LEDs.



3. Operating the Smart Configurator

3.1 Displaying the Smart Configurator Perspective

To fully utilize Smart Configurator features, ensure that the Smart Configurator perspective is opened. If it is not opened, select the perspective icon in the upper right corner of the e^2 studio window:

	(Quick A	ccess	B
e ² Open Perspective	-	_		×
Image: C/C++ (default) Image: Code Generator <				
	Open		Cancel	

Figure 3-1 Opening the Smart Configurator Perspective



3.2 Procedure for Operations

Figure 3-2 shows the procedure for using the Smart Configurator to set up peripheral modules and build the project with the e² studio. Refer to the related documents on the e² studio for the operation of the e² studio.

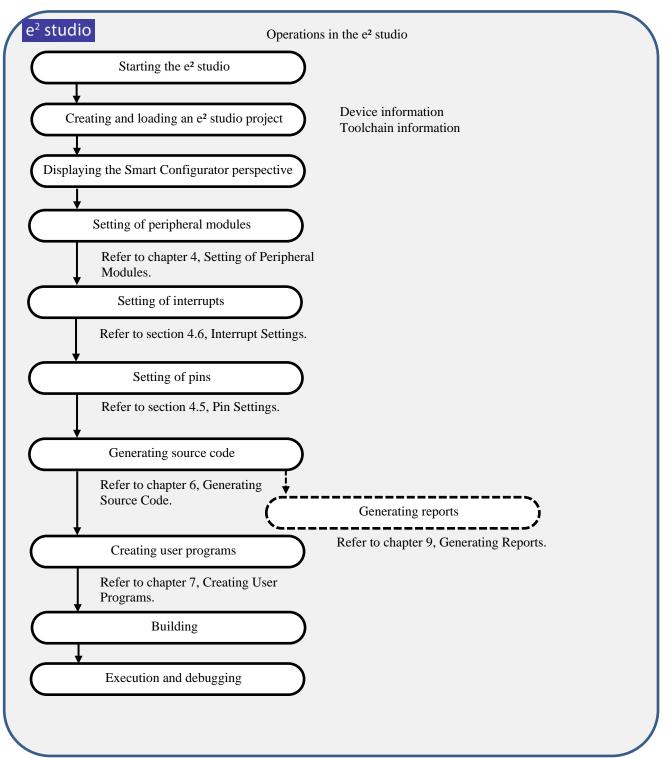


Figure 3-2 Procedure for Operations

3.3 File to be Saved as Project Information

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

The project file from the Smart Configurator is saved in "project name.scfg", which is at the same level as the project file (.project) of the e² studio.



3.4 Window

The configuration of the Smart Configurator perspective is shown in Figure 3-3, Smart Configurator Perspective.

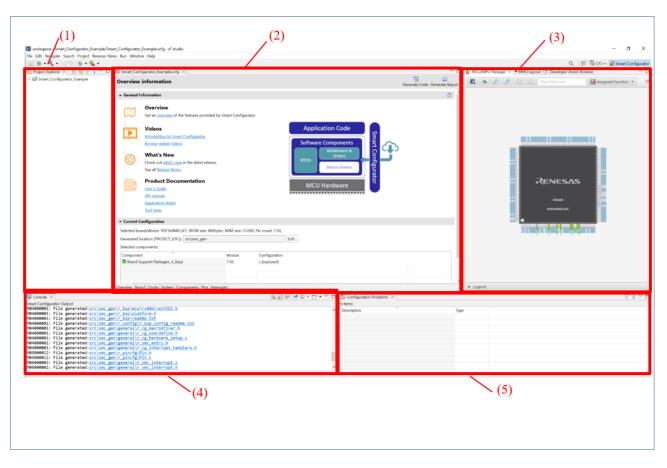


Figure 3-3 Smart Configurator Perspective

- 1) Project Explorer
- 2) Smart Configurator view
- 3) MCU/MPU Package view
- 4) Console view
- 5) Configuration Problems view



3.4.1 Project Explorer

The structure of the folders in the project is displayed in a tree form.

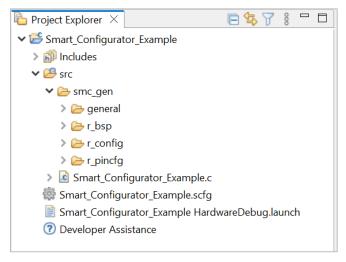


Figure 3-4 Project Explorer

When the Project Explorer is not opened, select [Window] \Box [Show View] \Box [Other] from the e² studio menu and select [General] \Box [Project Explorer] on the opened [Show View] dialog box.

3.4.2 Smart Configurator view

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

	¢\$ ‡ • % •		Q 🛛 😰 🕅 C/C++ 📓 Smart C	
Smart_Conf	figurator_Example.scfg ×			
Overview	information		Generate Code Generate	Repor
+ General I	nformation			0
m	Overview			
m	Get an overview of the features pro-	vided by Smart Configur	or.	
(T)	Videos		Application Code	
	Introduction to Smart Configurator		Ê	
	Browse related videos		Software Components	
0	What's New		Middleware & O	
(\mathbf{z})	Check out what's new in the latest re	elease.		
	See all <u>Release Notes</u> .		Software Components Middleware & Drivers Device Drivers	
r[=]e	Product Documentation			
	User's Guide		MCU Hardware	
	API manual			
	Application Notes			
	Tool news			
+ Current C	onfiguration			
	ard/device: R5F564MLCxFC (ROM size	: 4Mbytes, RAM size: 512	(B, Pin count: 176)	
Selected bo	ocation (PROJECT LOC\): src\smc gen		Edit	
	ocation (intesseet_coct). Stephne_get			
Generated lo	mponents:			
Generated lo Selected cor Componen	mponents:	Version 7.42	Configuration r.bsp(used)	

Figure 3-5 Smart Configurator View

When this view is not opened, right-click on the project file (*.scfg) in the Project Explorer and select [Open] from the context menu.



3.4.3 MCU/MPU Package view

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

Three types of package view can be switched between [Assigned Function], [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 4.1.1 Selecting the board" and "chapter 4.5.6 Pin setting using board pin configuration information").



Figure 3-6 MCU/MPU Package View

When this view is not opened, select [Renesas Views] \Box [Smart Configurator] \Box [MCU/MPU Package] from the e² studio menu.



3.4.4 Console view

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

[⊇] Console [∞]	🔒 🚮 🚱 🗹 🚍 🔫 🗖 🗖
mart Configurator Output	
NOSOCOFI: Fin 1L1 is assigned to ET1_TX_ER 105000001: Pin 149 is assigned to ET1_TX_ER 105000001: Pin 152 is assigned to ET1_ERXD3 105000001: Pin 155 is assigned to ET1_ERXD1 105000001: Pin 157 is assigned to ET1_ERXD0	
105000001: Pin 144 is assigned to ET1_RX_ER 105000001: Pin 161 is assigned to ET1_COL 105000001: Pin 32 is assigned to ET1_MDC	
105000001: Pin 33 is assigned to ET1_MDIO 105000001: Pin 159 is assigned to ET1_LINKSTA	
<	>

Figure 3-7 Console View

When this view is not opened, select [Window] \Box [Show View] \Box [Other] from the e² studio menu and select [General] \Box [Console] on the opened [Show View] dialog box.

3.4.5 Configuration Problems view

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

Configuration Problems Oitems Description		** ~ -
0 items		
Description	Туре	

Figure 3-8 Configuration Problems View

When this view is not opened, select [Renesas Views] \Box [Smart Configurator] \Box [Configuration Problems] from the e^2 studio menu.



4. Setting of Peripheral Modules

You can select peripheral modules from the Smart Configurator view.

4.1 Board Settings

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

4.1.1 Selecting the board

Click on the [] button to select a board.

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

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

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

The .bdf file of Renesas made board (for e.g. Renesas Starter Kit) can be downloaded from website and imported.

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

Depending on the board selected, the device will change, device change is reflected to the target device of e^2 studio project. It is the same with the procedure of " chapter 4.7 MCU migration feature".

Smart_C	onfigurator_Example.scfg ⊠		
Device	selection	🖫 Generate Code	Generate Report
Device se	election		è 4
Board: Device:			
	Download more boards		
Overview	Board Clocks System Components Pins Interrupts		*

Figure 4-1 Selecting the Board



4.1.2 Exporting board settings

Follow the procedure below to export the board settings.

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

♣ Smart_Configurator_Example.scfg ≅		scfg 🛛									
Device	selec	tion						🗊 te Code	Generate	Repo	ort
Device s	electio	n							<u>p</u>	4	^
Board:	RSKR	X64M (1.00)	×						(1)	
Device:	R5F56	64MLC>	κFC								
	Down	load m	ore boa	rds							
											~
Overview	Board	Clocks	System	Components	Pins	Interrupts	5				

Figure 4-2 Exporting Board Settings (bdf Format)

4.1.3 Importing board settings

Follow the procedure below to import board settings.

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

⇔ Smart_Configurator_Example.scfg ⊠				- 0
Device selection			🖫 Generate Code	Generate Report
Device selection Board: RSKRX64M (1.00) Device: RSF564MLCxFC Download more boards	(2))		
		1		~
Overview Board Clocks System Compo	onents Pins	Interrupts		

Figure 4-3 Importing Board Settings (bdf Format)

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



4.2 Clock Settings

You can set the system clock on the [Clocks] tabbed page. The settings made on the [Clocks] page are used for all drivers and middleware.

Follow the procedure below to modify the clock settings.

- (1) Specify the VCC voltage.
- (2) Select the clocks required for device operations on the board (the main clock is selected by default).
- (3) Specify the frequency of each clock in accordance with the board specifications (note that the frequency is fixed for some internal clocks).
- (4) When using the PLL circuit, select the clock source for the PLL.
- (5) For the multiplexer symbol, select the clock source for the output clocks.
- (6) To obtain a desired output clock frequency, select a frequency division ratio from the drop-down list.

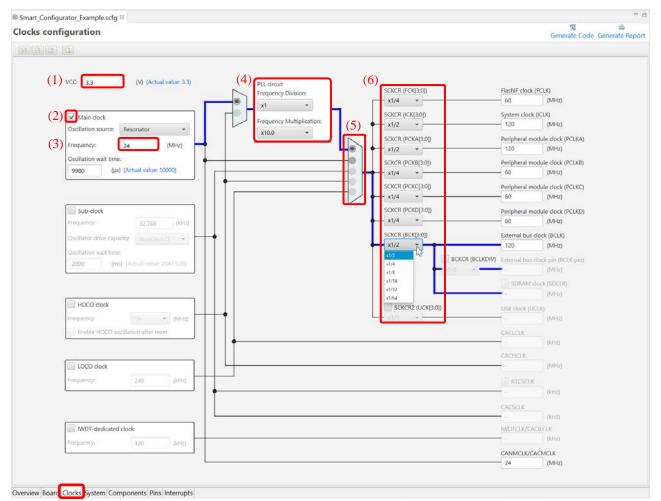


Figure 4-4 Clock Settings



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.

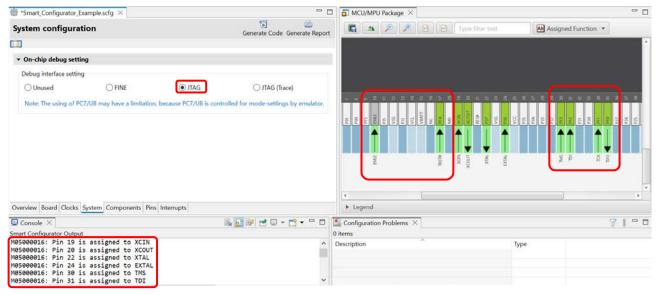


Figure 4-5 Debug Interface Setting at [System] Page



4.4 Component Settings

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

Smart_Configurator_Example.sc	Smart_Configurator_Example.scfg 🛛 🗖 🗖							
Software component co	nfiguration	🕲 Generate Code	👜 Generate Report					
Components 🛛 🖾 🖻 🖶 🕇 🔻	Configure		i					
type filter text × ✓ ≧ Startup ✓ ≧ Generic ≧ r_bsp	Components tree							
Overview Board Clocks System	omponents Pins Interrupts							

Figure 4-6 [Components] Page

The Smart Configurator supports two types of software components: Code Generator (CG) components and Firmware Integration Technology (FIT) modules.

4.4.1 Adding Code Generator components

The following describes the procedure for adding a component.

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

Software component configuration	🖫 Generate Code	👜 Generate Report
Components ^{Iª} ₂ □ ⊞ D ▼ Configure		(j)
(1) type filter text × ✓ ➢ Startup ✓ ➢ Generic ☞ r_bsp		
Overview Board Clocks System Components Pins Interrupts		

Figure 4-7 Adding a Component



- (2) Select a component from the list in the [Software Component Selection] page of the [New Component] dialog box (for e.g. Single Scan Mode S12AD).
- (3) Check that [Type] for the selected component is [Code Generator].
- (4) Click on [Next].

	nponent from those available in list					
Category	All					
Function	All					
Filter						
Compon	ents	Short Name		Туре	Version	
BD M	ode SDHI Driver	r sdhi rx		Firmware Integra	2.11	
E Serial	Sound Interface driver.	r_ssi_api_rx		Firmware Integra	2.03	
H Simpl	e IIC Driver.	r_sci_iic_rx		Firmware Integra	2.70	
🖶 Single	Scan Mode S12AD		(3)	Code Generator	2.5.0	
🖶 Smart	Card Interface Mode		X - X	Code Generator	1.12.0	
🖶 SPI CI	ock Synchronous Mode (3-wire me			Code Generator	1.12.0	
🖶 SPI O	peration Mode (4-wire method)			Code Generator	1.10.0	
	Denete and the de INADE TA TELLA DV			Pt	1 00	_
	only latest version					
✓ Hide it	ems that have duplicated functionalit	у				
Descriptio	n					
	ware component generates two units prise two 8-bit counter channels, tota	· · · · · ·		hip 8-bit timer (TMR)	module	
Download	the latest FIT drivers and middlewar	<u>e</u>				
<u>Configure</u>	general settings					

Figure 4-8 Adding a Code Generator Component

- (5) Specify an appropriate configuration name in the [Add new configuration for selected component] page of the [New Component] dialog box or use the default name (for e.g. Config_S12AD0).
- (6) Select a hardware resource or use the default resource (for e.g. S12AD0).
- (7) Click on [Finish].

e ² New Component		×
Add new configuration for selected component		
Resource: (6) S12AD0		\sim
(7).	Can	cel

Figure 4-9 Adding a Component



4.4.2 Removing a software component

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

- (1) Select a software component from the Components tree.
- (2) Click on the [(Remove component)] icon.

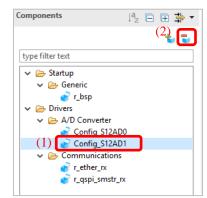


Figure 4-10 Removing a Software Component

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

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

C	omponents	è d'az	- 🕀 🖶 -			
			ء	(2		
	type filter text					
Γ	✓					
	🗸 🗁 Generic					
	💣 r_bsp					
	➤ Drivers					
	✓ ▷ A/D.C	onverter				
	Co 🐨	onfig_S12AD0				
l)	💣 Co	onfig_S12AD1				
	🗸 🕞 Comr	munications				
	💣 r_e	ether_rx				
	rq	qspi_smstr_rx				

Figure 4-11 Removing Software Components

Source files generated for this component are not removed from the e^2 studio project tree. After generating source code by clicking [\bigcirc (Generate Code)] icon, the source files generated for removed component will be removed from the e^2 studio project tree.



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

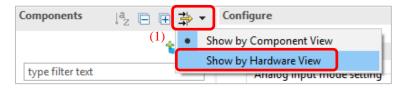


Figure 4-12 Switching to the Hardware View

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

Components	≧ 凶 № 日 田 夢・	Configure		
type filter text	wit racy measurement circuit tion nit er pulse unit 3 it generator r rW is interface face (RSPI) og timer	Show only latest version Hide items that have duplicate Description This software component provid	aliable in list Short Name () (3) (3) It functionality Ites single scan mode configuration (ted are converted for only once in	Version 1.13.0 1.12.0 2.5.0 2 3

Figure 4-13 Adding a CG Component to the Hardware View

4.4.4 Setting a CG driver

Follow the procedure below to set up a CG configuration.

- (1) Select a CG configuration from the Components tree (for e.g. Config_S12AD0).
- (2) Configure the driver in the [Configure] panel to the right of the Components tree. The following steps and figure show an example.
 - a. Select 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].

Software component co	nfiguration		Generate Code Generate Repor
Components	3683 • • •	Configure	
type filter text			
 ✓ Startup ✓ Seneric ✓ r_bsp 	^		
Drivers A/D Converter Onfig S12AD0		Analog input mode setting Double trigger mode	
	(2) a	Analog input channel setting Image: Constraint of the set of t	
	(2) t	Conversion start trigger setting Start trigger source	_
		A/D conversion start trigger pin Interrupt setting Interrupt AD conversion end interrupt (S12ADI) Priority Level 15 (highest) ~	
	(2) 0	- Advance setting	
		Add/Average AD value setting AN000 AN001 AN002 AN003 AN004 AN005 AN006 AN007	
		Self diagnosis setting	
		Mode Unused 👻	
		Voltage used	
		Disconnection detection assist setting Charge setting (2) d. Discharge	
		Period 2 ADCLK Y	
	~		>

Figure 4-14 Setting of a CG Driver

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

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

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



4.4.5 Changing the resource for a CG configuration

The Smart Configurator enables you to change the resource for a CG configuration (for e.g. from S12AD0 to S12AD1). Compatible settings can be ported from the current resource to the new resource selected.

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

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

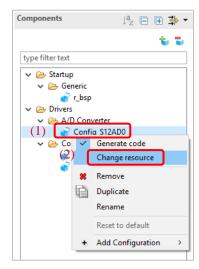


Figure 4-15 Changing the Resource

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

e ² Resour	ce Selection					×
Resource		hose availa	hla in tha	liet.		
		inose availa		list		
Resource:	S12AD1 S12AD0		_			~
(3)	S12AD1					

Figure 4-16 Components Page – Selecting a New Resource

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

(7) Confirm setting for resource change	● Use setting below ○ Use default	
Setting	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 trigger	Yes
<		2

Figure 4-17 Checking the Settings of the New Resource

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

Smart_Configurator_Example.s Software component componen	- 5					ି ଅ ସ୍ଥି 💩 Generate Code Generate Report
Components	P2 (5 (6 (5) *	Configure				^
type filter text		* Basic setting				
 Startup Generic 	^	Note When using the 12-bit A/	D converter (unit 1), we	recommend not u	ising ports 02, 01, 00, port 9, port D,	and port E as output ports.
💣 r_bsp		Analog input mode settir	ng			
👻 😂 Drivers		Double trigger mode	Extend analog in	iput mode		
 A/D Converter Config_S12AD0 		Analog input channel set	ting			
Contig_S12ADU		AN100 AN10		AN103	AN104	
		AN105 AN10	06 🗌 AN107	AN108	AN109	
		AN110 AN1	11 🗌 AN112	AN113	AN114	
		AN115 AN11	16 🗌 AN117	AN118	AN119	
		AN120 Temp	erature sensor output	🗌 Internal re	ference voltage	
		Conversion start trigger s Start trigger source	etting			
		A/D conversion start trig	ger pin			~
		Interrupt setting	end interrupt S12ADI1	Priority Leve	l 15 (highest) 😔	
		- Advance setting				
		Add/Average AD value s	etting			
		AN100 AN10	01 AN102	AN103	AN104	
		AN105 AN10	06 AN107	AN108	AN109	
		AN110 AN1	11 AN112	AN113	AN114	
		AN115 AN1	AN117	AN118	AN119	
		AN120 Temp	erature sensor output	Internal re	ference voltage	
		Self diagnosis setting				
		Mode	Unused		*	
	~					× *

Figure 4-18 Resource Changed Automatically



To change the configuration name, follow the procedure below.

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

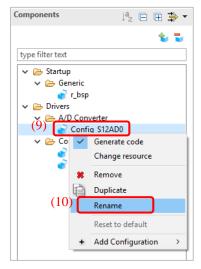


Figure 4-19 Renaming the Configuration



4.4.6 Downloading a FIT module

You need to download a desired FIT driver or middleware from the Renesas Electronics website.

- (1) Click on the 👘 (Add component)] icon.
- (2) Click the [Download the latest FIT drivers and middleware] link in the [Software Component Selection] page of the [New Component] dialog box to download a FIT module.

Category	All				
Function					
	1.70	dules are not	available for se	lection	_
Filter		duies are not	available for se		1
Compo	nents	Short Name	Type	Version	
# 8-Bit	Timer		Code Generator	1.10.0	
Buse	s		Code Generator	1.11.0	
# CRC	Calculator		Code Generator	1.11.0	
E Clock	Frequency Accuracy Measurement		Code Generator	1.11.0	
Com	pare Match Timer		Code Generator	2.3.0	
# Com	plementary PWM Mode Timer		Code Generator	1.11.0	
# Cont	inuous Scan Mode S12AD		Code Generator	1.13.0	
#D/A	Converter		Code Generator	1.11.0	
Show	only latest version				
Hide i	tems that have duplicated functionalit	y.			
Descripti	on				
	tware component generates two units nprise two 8-bit counter channels, tot		on-chip 8-bit timer (TMR) module	
Downloa	d the latest FIT drivers and middlewar	e			

Figure 4-20 Downloading More Software Components

Note: Downloading requires login to "My Renesas". If you have not logged in, the following dialog box will prompt you to log in. To register as a new user, click on the [About My Renesas] button.

e ² My Renesas		×
My Renesas		
Enter the e-mail address They allow you to down	and password that you registered for My Renesas. ad documents and software by using Smart Browser.	
Email Address:		
Password:		
	as account to use our tool download services, receive Newsletter / Update Notice, and take advantage of our other services. enesas] to register it.	
	About My Renesas	OK Cancel

Figure 4-21 Login to My Renesas



- (3) Select the checkbox of the required module in the [FIT Module Download] dialog box. If [Show RX Driver Package only] is unchecked, filtering of items is canceled.
- (4) Click on [Browse...] to select the location where the downloaded module is to be stored.
- (5) Click on [Download] to start downloading the selected FIT module.

	Select the FIT modules for download				□ ×		New Component Software Component Selection Select component from those available in list Category Al	TT modu	iles are	#	×
(3)	Title RX Family RX Driver Package Ver	Document No. R01AN5826EJ0129	Rev. Rev.1.29	Issue date 2021-04-08	Select All Deselect All	•	Components Graphic Utrary with One-Stop Develope. Band Support Rackages. Byte Isseed Guide buffer Rikary. CAN Driver Show only latest version Show only latest version	r_bsp r_byteq r_clin_rx	for sele	Version 1.00 7.41 2.10 5.40	
(3	Filtered: Show RX Driver Package only Module Folder Path: C\FITModules			(5) Download	(4) Browse Cancel		Description between despring GUI and developing MCU by implement GUI with modifying Rowerhoat & file variations and inseparate edition.] Described the bate HTI deven and middleweer Configure general antitiops.			uttiple	< >

Figure 4-22 Downloading a FIT Module



4.4.7 Adding FIT drivers or middleware

The following describes the procedure for adding FIT drivers or 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 (for 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 [Firmware Integration Technology].
- (4) Click on [Finish].

elect con	nponent from those available in list				
Category	All				~
unction	All				
ilter	(2)				
Compon	ents	Short Name (3)	Type	Version	^
	net Driver.	r ether rx	Firmware Integra	1.23	
Flash	API for RX100, RX200, RX600. and	r flash rx	Firmware Integra	5.10	1
	Synchronous Control Module for S		Firmware Integra	3.30	
	5 Sensor Middleware	r fs1015 rx	Firmware Integra	1.00	
FS201	2 Sensor Middleware	r_fs2012_rx	Firmware Integra	1.03	
FS300	00 Sensor Middleware	r fs3000 rx	Firmware Integra	1.00	1
HS30	0x Sensor Middleware	r_hs300x_rx	Firmware Integra	1.23	
HS40	0x Sensor Middleware	r hs400x rx	Firmware Integra	1.01	
Unsig	ned 32-bit circular buffer library.	r_longq	Firmware Integra	2.00	
Contr	ol Low Power States.	r lpc rx	Firmware Integra	2.30	
	DRV Driver	r memdrv rx	Firmware Integra	1.05	
MMC	Mode MMCIF Driver	r_mmcif_rx	Firmware Integra	1.10	
BOB12	03 Sensor Middleware	r ob1203 rx	Firmware Integra	1.01	
PDC o	driver	r_pdc_rx	Firmware Integra	2.06	
PTP L	ght Driver	r ptp light rx	Firmware Integra	1.14	
	river	r_ptp_rx	Firmware Integra	1.17	
OSPI	Clock Synchronous Single Master C	r_qspi_smstr_rx	Firmware Integra	1.21	~
Hide it		у			
The Ethe Ethernet	ncy : r_bsp version(s) 5.52 met fit module provides a method to controller (ETHERC), Ethernet DMA co I the latest FIT drivers and middlewar	ontroller (EDMAC).	net / IEEE802.3 frame u	sing	~
		<u>e</u>			
onfigure	general settings				
			(4)		

Figure 4-23 Adding FIT Modules

Download and import FIT sample project 4.4.8

When the FIT driver or middleware icon is [¹], you can download the sample project. (1) Select the FIT driver or middleware of the [¹] icon and select [Download and import sample projects] from the right menu. (for e.g. CMT)

Smart_Configurator_Example Software component			10 m
Software component	configuration		Generate Code Generate Report
Components	15 E E 🏕 🔻	Configure	0
	22	Property	Value
type filter text		✓ [◎] Configurations	
 Startup Generic r.bsp Drivers A/D Converter Config_S12AD1 Communications r_ether_rx r_ether_rx r_ether_rx r_ether_rx 	Î	CMT interrupts priority level	5
<pre>% r_cmt_rx </pre> ✓ Middleware	Change version		
 Seneric 	* Remove		
🐮 r_byteq	Reset to default		
	Download and	import sample projects	>
Overview Board Clocks System	~		~

Figure 4-24 Download and import sample projects

(2) The sample code is displayed on the [Application Notes] tab of the Smart Browser, so select [Sample Code (import projects)] on the right menu.

🔝 Configuration Problems 🛛 🛶 Smart Browser 🐹			🔗 💠 🏇 🗄	🍇 🙀 🖌	~ - 0					
Device: R5F564MLC(RX64M) Last updated: 2019/01/15 at 17:50:49 JST										
Context Help User's Manual Technical Update Application Notes Tool News Notifications										
1 matches (filtering)										
Title	Document	Rev.	Issue Date	Sample Co	Remarks					
UPDATED RX Family CMT Module Using Firmware Integration Technology	R01AN185	Rev.3.31	2018/11/16	available						
Open										
Sample Code (download)										
Sample Code (import project	s)									
Property										
<					>					

Figure 4-25 Sample Code (import projects)

(3) Specify the save destination of the sample code and click [Save].

e ² Save As								×
← → * ↑	« Loc	al Disk (C:)	> Renesas > e2_studio	Downloads	ٽ ~	Search Download	5	P
Organize 👻 Ne	w folder							?
> 🁌 Music	^	Name	^	Date	modified	Туре	Size	
 > Pictures > Videos 				No items matc	h your search.			
> 🏪 Local Disk (C:)							
> 💣 Network	~							
File name:	an-r01	an1856ej033	1-rx-timer.zip					~
Save as type:	*.*							~
∧ Hide Folders						Save	Cancel	

Figure 4-26 Sample code save destination



(4) When [End User License Agreement] dialog is displayed, click [Agree].

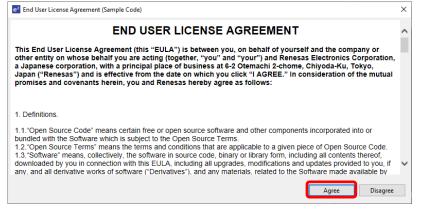


Figure 4-27 [End User License Agreement] dialog

(5) If the [Select import package] dialog is not displayed, go to procedure (6).

If the [Select import package] dialog box is displayed, select the package to import and click [OK].

e ² Select import	r package	\times
Import package		\sim
	an-r01an1856ej0331-rx-timer\FITDemos\cmt_demo_rskrx113.zip an-r01an1856ei0331-rx-timer\FITDemos\cmt_demo_rskrx231.zip	
	an-r01an1856ej0331-rx-timer\FITDemos\cmt_demo_rskrx64m.zip	
	an-r01an1856ej0331-rx-timer\FITDemos\cmt_demo_rskrx65n.zip an-r01an1856ej0331-rx-timer\FITDemos\cmt_demo_rskrx65n_2m.zip an-r01an1856ej0331-rx-timer\FITDemos\cmt_demo_rskrx71m.zip	
e ² Select import	package	×
Import package	an-r01an1856ej0331-rx-timer\FITDemos\cmt_demo_rskrx64m.zip	~
	OK Cancel	

Figure 4-28 [Select import package] dialog



(6) When the [Import] dialog of the project appears, select the project and click [Finish].

📴 Import				\times
Import Projects				
Select a directory to search for existing Ec	lipse projects.			
O Select root directory:	~		Browse	
Select archive file: Demos\cmt_der	no_rskrx64m.zip 🐱		Browse	
Projects:				
✓ cmt_demo_rskrx64m_gcc (cmt_dem			Select Al	I
✓ cmt_demo_rskrx64m (cmt_demo_rs	krx64m/)		Deselect A	All
			Refresh	
Options		4		
Search for nested projects				
Copy projects into workspace				
Close newly imported projects upon c	ompletion			
Hide projects that already exist in the	workspace			
Working sets				
Add project to working sets			New	
Working sets:	\sim		Select	
(?)	Finish		Cancel	

Figure 4-29 Import project

(7) It is added to [Project Explorer], and import of sample project is completed.

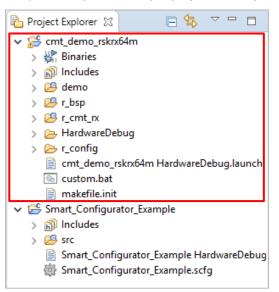


Figure 4-30 Addition to Project Explorer



4.4.9 Setting a FIT Software Component

To use FIT drivers or middleware, set configuration option. Setting methods depends on components,

- Set configuration options on Configure panel and settings will be generated to configuration file of FIT module automatically at each time of code generation action
- ✓ Set configuration options in configuration file of FIT module by manually

Configuration file of FIT module will be generated in the folder r_config. For the settings of the configuration options, refer to "chapter 7.1, Adding Custom Code in the Case of Firmware Integration Technology (FIT)".

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

figuration	Generate	Code Generate Repo		
🐴 🖯 🏵 🕈 🔻 Configure				
Property	Value	^		
	2			
	and the second			
	IVI Used			
	🕅 Used			
	2 Used			
ETO ERXD1 Pin	🗵 Used			
~ ETO ERXDO Pin	🗵 Used			
TO RX ER Pin	I Used			
~ ETO_CRS Pin	🗵 Used			
~ ETO COL Pin	2 Used			
~ ETO_MDC Pin	🕅 Used			
~ ETO_MDIO Pin	🗵 Used			
>> ET0 LINKSTA Pin	IV Used			
~ ETO_EXOUT Pin	🗉 Used			
→ ET0 WOL Pin	🖾 Used	4		
	Property ✓ ● ETHERC ✓ ● ETHERC ✓ ● ETHERC ✓ ● ETHERC ✓ ● ETO_TX_CLK Pin ~ ETO_TX_EN Pin ~ ETO_TX_EN Pin ~ ETO_TX_EN Pin ~ ETO_TX_ER Pin ~ ETO_EXDO Pin ~ ETO_ERXDV Pin ~ ETO_ERXD2 Pin ~ ETO_ERXD2 Pin ~ ETO_ERXD2 Pin ~ ETO_ERXDP Pin ~ ETO_COL Pin ~ ETO_COL Pin ~ ETO_MDIO Pin ~ ETO_MDIO Pin ~ ETO_INIC Pin ~ ETO_INIC Pin ~ ETO_INIC Pin	Property Value * ● ETHERC Property * ● ETHERC Ø * ● ETHERC Ø * ● ETHERC Ø * ● ETO_TX_CLK Pin Ø * ● ETO_TX_EN Pin Ø * ● ETO_TX_EN Pin Ø * ● ETO_ETXD2 Pin Ø * ● ETO_ETXD0 Pin Ø * ● ETO_ERXDV Pin Ø * ● ETO_ERXD2 Pin Ø * ● ETO_ERXDP Pin Ø * ● ETO_COL Pin Ø * ● ETO_COL Pin Ø * ● ETO_EXOUT Pin Ø * ● ETO_EXOUT Pin Ø * ● ETO_EXOUT Pin Ø		

Overview Board Clocks System Components Pins Interrupts

Figure 4-31 Pin Settings for r_ether_rx

Smart_Configurator_Example.scfg		5 0
Software component confi	guration	Generate Code Generate Repor
Components	් 🗄 🕀 🏚 🔻 Configure	G
- Formation	Property	Value
type filter text	 Configurations 	
Y 👄 Startup	# Use FIT	100 Used
Y 😂 Generic	# Use QSPI Channel 0	🕅 Used
💣 r_bsp	# Enable debugging information	🖾 Unused
✓ ➢ Drivers	# SPTI0 Interrupt priority level	Level 10
 Communications 	# SPRI0 Interrupt priority level	Level 10
r ether rx	✓ ^{II} Resources	
💣 r_qspi_smstr_rx	✓ [©] QSPI	
✓ Imers	QSPIO	
😵 r_cmt_rx	➤ QSPCLK Pin	□Used
Y 😂 Middleware	► QIO0 Pin	□Used
👻 😂 Generic	► QIO1 Pin	D Used
😵 r_byteq	∼ QIO2 Pin	🖾 Used
	▲ QIO3 Pin	🖾 Usød
		^
		~

Figure 4-32 Pin Settings for r_qspi_smstr_rx



4.4.10 Version change of FIT software component

The following describes the procedure for version change of FIT software component.

(1) From the component tree, right-click the FIT software component that you want to change version.

Software componer	it configuration		Generate Code Generate Repor
Components	15 E B 🕸 🔻	Configure	Q
	\$ 5	Property	Value
type filter text		✓ ● Configurations	
✓ ➡ Startup		# Use FIT	⊠ Used
👻 😂 Generic		# Use QSPI Channel 0	⊠ Used
💣 r_bsp		# Enable debugging information	🗉 Unused
Y 😂 Drivers		# SPTI0 Interrupt priority level	Level 10
 Communications 		# SPRI0 Interrupt priority level	Level 10
r_ether_rx		✓	
💰 r_qspi_smstr_n			
Y 🗁 Timers	Change version		13
😵 r_cmt_rx	Remove P3	LK Pin	E Used
Y 🗁 Middleware	Reset to default	Pin	DUsed
Y 🗁 Generic	Download and import s	ample projects	DUsed
😵 r_byteq		Pin Pin	DUsed
		∼ QIO3 Pin	DUsed
		Macro definition: QSPI_SMSTR_CFG_USE_FIT Specify if FIT modules are used.	~
		(1) 2) ▲ al 100 ▲ alconomentational de local alconomical de la constantina de la constantin Constantina de la constantina de la const	

Figure 4-33 Version change of FIT software component

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

Version Selectio		
Component name:	r_qspi_smstr_rx	
Current version:	1.10	
Available versions:		
	1.14 1.13 1.12	

Figure 4-34 Select version of FIT software component

(4) Click [Next].



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

Change Version					\times
Setting Overview	ı				
The following sett	ngs will be added or remov	ved			
Setting				Status	;
✓ Configurations					
Use FIT				Adde	d
Use QSPI Ch	innel 0			Adde	b
Enable debu	ging information			Adde	b
SPTI0 Interru	ot priority level			Adde	b
SPRI0 Interru	pt priority level			Adde	d
<					>
?	< <u>B</u> ack	<u>N</u> ext >	<u> </u>	Cancel	

Figure 4-35 Confirm setting change item

(6) As [Confirm to change version and proceed to generate code] is displayed, if you are fine to proceed, click [Yes].



Figure 4-36 Confirm version change

(7) The FIT software component version is changed. Code generation is executed automatically.



4.4.11 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 4.4.10 Version change of FIT software component" to change to an available version.

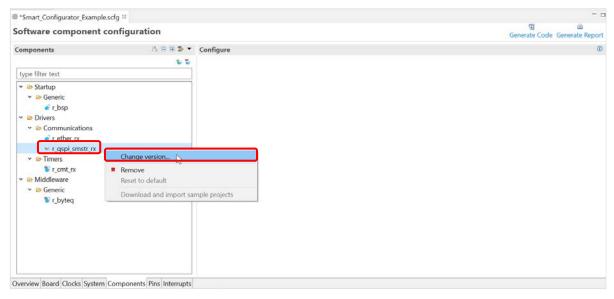


Figure 4-37 Change version of a greyed-out component

(2) If there is no available version for this component, refer to "chapter 4.4.6 Downloading a FIT module" to download this component from Renesas website.



4.4.12 Setting the RTOS Kernel

The following describes the procedure for setting the FreeRTOS Kernel.

For Renesas FreeRTOS, refer to Renesas FreeRTOS related documentation.

- (1) Select [FreeRTOS_Kernel] in the component tree.
- (2) Parameters corresponding to the RTOS kernel are displayed in the Configure panel, and configuration settings can be changed.
- (3) Description of the parameter selected in the Configure panel and the corresponding macro definition are displayed in this area.

		Generate Code Generate Rep
Components 🚵 🖄 🖓 🖯 🖽 🛱		
(4	Property	Value
type filter text	# Tick hook	🗹 Enable
V 🧽 Startup	# The frequency of the CPU dock	
V 😂 Generic	# The frequency of the PERPHERAL stock	
r_bsp	# The frequency of the RTOS tick interrupt	(TickType_t) 1000
V Con RTOS	# The size of the stack used by the idle task	(unsigned short) 140
V 🕞 RTOS Kernel	# The configTOTAL_HEAP_SIZE_N	8
FreeRTOS Kernel	# The total amount of RAM available in the FreeBTOS heap	(size_1) (cooligitOTAL_HEAP_SIZE_N * 1024)
✓ → RIOS Object	# The maximum permissible length of name	12
FreeRTOS_Object	# Use trace facility	🗹 Enable
- receives_object	# Use 16bit ticks	Disable
	# Idle should yield	2 Enable
	# Co-routine	Disable
	# Mutex functionality	2 Enable
	# Run time statistics	Disable
	# Check for stack overflow	Check by tick value and stack pointer.
	# Recursive mutex functionality	V Enable
	# Queue registry size	0
	# The malloc() failed function	Z Enable
	# Use application task tag	Disable
	# Ourwa cot functionality Macro definition:configTICK RATE HZ	III Enables

Figure 4-38 Settings for FreeRTOS_Kernel



4.4.13 Creating the RTOS Object

The following describes the procedure for setting the FreeRTOS Object.

- (1) Select [FreeRTOS_Object] in the component tree.
- (2) Configuration Tabs corresponding to the RTOS objects are displayed in the Configure panel. Select a tab corresponding to the object you want to create.
- (3) Click on the [+] button to create a new object. Click on the [-] button to remove an object.
- (4) Use text boxes and combo boxes to change the object setting.

	on						Generate Code Generat
nponents 🛛 🚵 🛃 📮 🕀 🐳 🕆	▼ Configure						
😜 ह	(2) Heap Estimation Tasks	Semaphores	Queues Softwa	re Timers Even	t Groups Stream Buffers Messag	ge Buffers	
pe filter text	(3) +/- Initialize kernel start ×	Task Code task_1	Task Name task_1	Stack Size 512	Task Handler (4) Parameter NULL NULL	Priority 1	Heap Usage 2160
FreeRTOS_Kernel							

Figure 4-39 Create FreeRTOS_Object



4.4.14 Setting the RTOS Library

The following describes the procedure for setting the FreeRTOS Library.

(1) Select a library under [RTOS_Library] folder in the component tree.

Note: The RTOS Library is only available for "FreeRTOS (with IoT libraries)" project.

- (2) Parameters corresponding to the RTOS Library are displayed in the Configure panel, and configuration settings can be changed.
- (3) Description of the parameter selected in the Configure panel and the corresponding macro definition are displayed in this area.

Software component config	uration		🕤 🔐
Components in 🖄 🗄	🗄 🕏 🔻 Configure		
type filter text	(2) Property	Value	
Sr_flash_rx	 # Number of jsmn token 	s 64	
 Communications 	# The JSON key	"clientToken"	
 r_ether_rx r_sci_rx Middleware Generic r_byteq RTOS RTOS Kernel FreeRTOS_Kernel RTOS Object 	# ShadowConfigUNIQUE	_CLIENT Disable	
 AWS_device_shadow AWS_device_shadow AWS_ggd AWS_mqtt AWS_secure_socket AWS_tcp_ip 	(3) Macro definition:shadowConfi Number of jsmn tokens to use i		

Figure 4-40 Settings for RTOS Library



4.4.15 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

oftware component							
omponents	P2 🖻 🛱 🕏 🔻	Configure					
Call .	÷ 5	* Basic setting					
type filter text		Analog input channel setting					
Startup	^	Channel 0					
 Generic r_bsp 		Channel 1					
Drivers		Channel 2					
✓		Channel 3					
Config_AFE		Channel 4					
Config_DSAD0		Channel 5					
		ΔΣ A/D Converter operation voltage settin					
		 3.6V to 5.5V (High precision) 	2.7V to 5.5V				
		ΔΣ A/D Converter operation mode setting	12000				
		Normal mode	O Low-power mode				
		Operation clock setting				10 NO 10 A 10 A	
		Operation clock	PCLKB / 8	~	4.000	(MHz)	
		The operation clock can be set from 3.44 M	MHz to 4.56 MHz. Please set the operation	a clock within	the range		
		Conversion start trigger setting					
mart_Configurator_Example	.scfg ≅		Cotherso trianor	n		13	2
imart_Configurator_Example	.scfg ≌ configuration	< Interrupts			Gen	চ্চ erate Code Genera	
mart_Configurator_Example	.scfg ≅	< Interrupts			Gen		2
imart_Configurator_Example oftware component omponents	.scfg ≌ configuration	< Interrupts	Scroll down		Gen		-
Smart_Configurator_Example Structure component Supponents Support filter text.	scfg ∺ configuration A: □ ⊕ ⊉ ▼ & ⊽	< i Interrupts Channel setting	Scroll down		Gen		-
Smart_Configurator_Example offware component omponents type filter text >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	scfg ≌ configuration P₂ ⊜ ⊞ ⊉ ▼	Channel setting Channel 0 Channel 1 Channel 2 Channe Analog input setting	Scroll down		Gen		-
Smart_Configurator_Example offware component omponents ype filter text Saturp Saturp Saturp Saturp Saturp Saturp Saturp	scfg ∺ configuration A: □ ⊕ ⊉ ▼ & ⊽	Channel setting Channel 0 Channel 1 Channel 2 Channel 3 Channel 3	Scroll down		Gen		-
Smart_Configurator_Example offware components ype filter text Startup Startup Startup Startup Startup Startup Startup Startup Startup	scfg ∺ configuration A: □ ⊕ ⊉ ▼ & ⊽	Channel setting Channel 0 Channel 1 Channel 2 Channe Analog input setting	Scroll down		Gen		-
Smart_Configurator_Example offware component omponents ype filter text Startup Generic f, bsp Drivers A/D Converter 	scfg ∺ configuration A: □ ⊕ ⊉ ▼ & ⊽	Channel setting Channel O Channel 1 Channel 2 Channe Analog input setting Positive input signal	I3 Channel 4 Channel 5		Gen		-
Smart_Configurator_Example Strart_Configurator_Example Strartup Startup	scfg ∺ configuration A: □ ⊕ ⊉ ▼ & ⊽	Channel setting Channel of Channel 1 Channel 2 Channel Analog input setting Positive input signal Negative input signal Reference input	I3 Channel 4 Channel 5		Gen		-
Smart_Configurator_Example offware component omponents ype filter text Startup Generic f, bsp Drivers A/D Converter 	scfg ∺ configuration A: □ ⊕ ⊉ ▼ & ⊽	Channel setting Channel O Channel 1 Channel 2 Channe Analog input setting Positive input signal Negative input signal Reference input	I3 Channel 4 Channel 5		Gen		-
Smart_Configurator_Example Strart_Configurator_Example Strartup Startup	scfg ∺ configuration A: □ ⊕ ⊉ ▼ & ⊽	Channel setting Channel of Channel 1 Channel 2 Channel Analog input setting Positive input signal Negative input signal Reference input Dositive reference voltage buffer Negative reference voltage buffer Amplifier setting	Scroll down		Gen		-
Smart_Configurator_Example offware component omponents type filter text	scfg ∺ configuration A: □ ⊕ ⊉ ▼ & ⊽	Channel setting Channel setting Channel 0 Channel 1 Channel 2 Channe Analog input setting Positive input signal Negative input signal Reference input Positive reference voltage buffer Negative reference voltage buffer	I3 Channel 4 Channel 5		Gen		-
Smart_Configurator_Example offware component omponents type filter text * © Startup * © Startup * © Generic @ r_bsp * © Drivers * © A/D Converter @ Config_AFE	scfg ∺ configuration A: □ ⊕ ⊉ ▼ & ⊽	Channel setting Channel of Channel 1 Channel 2 Channel Analog input setting Positive input signal Negative input signal Reference input Dositive reference voltage buffer Negative reference voltage buffer Amplifier setting	Scroll down		Gen		-
imart_Configurator_Example offware component omponents ype filter text	scfg ∺ configuration A: □ ⊕ ⊉ ▼ & ⊽	Channel setting Channel setting Channel 0 Channel 1 Channel 2 Channe Analog input setting Positive input signal Negative input signal Reference input Dositive reference voltage buffer Negative reference voltage buffer Amplifier setting Amplifier setting	Scroll down		Gen		-
Smart_Configurator_Example offware component omponents type filter text * © Startup * © Startup * © Generic @ r_bsp * © Drivers * © A/D Converter @ Config_AFE	scfg ∺ configuration A: □ ⊕ ⊉ ▼ & ⊽	Channel setting Channel setting Channel 0 Channel 1 Channel 2 Channe Analog input setting Positive input signal Negative input signal Reference input Positive reference voltage buffer Negative reference voltage buffer Amplifier setting Amplifier setting Buffer amplifier	Scroll down Scroll down I3 Channel 4 Channel 5 I3 AIN1 AIN3 AVCC0/AVSS0 Unused Positive buffer amplifier		Gen		-
Smart_Configurator_Example offware component omponents type filter text * © Startup * © Startup * © Generic @ r_bsp * © Drivers * © A/D Converter @ Config_AFE	scfg ∺ configuration A: □ ⊕ ⊉ ▼ & ⊽	Channel setting Channel 0 Channel 1 Channel 2 Channe Analog input setting Positive input signal Reference input Positive reference voltage buffer Regative reference voltage buffer Amplifier selection Buffer amplifier PGA gain setting	Scroll down Scroll down I3 Channel 4 Channel 5 I3 AIN1 AIN3 AVCC0/AVSS0 Unused Positive buffer amplifier		Gen		-
 e² r_bsp ✓ ⁽²⁾ Orivers ✓ ⁽²⁾ A/D Converter e² Config_AFE 	scfg ∺ configuration A: □ ⊕ ⊉ ▼ & ⊽	<	Scroll down		Gen		-

Figure 4-41 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

Smart_Configurator_Exan	all and the second second			
Software compone	ent configuration	1		Generate Code Generate Repor
Components	12 E E 🕸 🔻	Configure		0 -
	÷ 5	Setting Analog pins' connection		
type filter text		Note		
 ✓ Startup ✓ Startup ✓ Generic ✓ r_bsp 	^	AIN6 ~ AIN11 pins conversion accuracy will Bias output setting	be degraded if set the same pin in \$12AD.	
 Drivers A/D Converter 		Enable bias voltage output	AIN1 pin output	
Config_AFE		AIN2 pin output	AIN3 pin output	
Config_DSAD	00	AIN4 pin output	AIN5 pin output	
		AIN6 pin output	AIN7 pin output	
		AIN10 pin output	AIN11 pin output	
		Excitation current output setting		
		Operation mode	2-channel output mode 🗁	
		Excitation current	50 µA ~	
		IEXC0 output pin	Output disabled 🚽	
		IEXCO disconnect detection assist		
	*	IEXC1 output pin	Output disabled	

Figure 4-42 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.

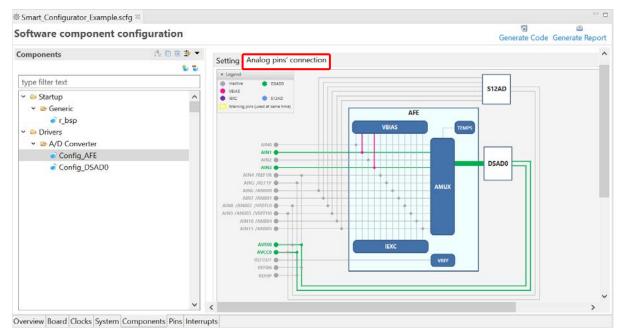


Figure 4-43 Block diagram of the AFE multiplexed pin connection



4.4.16 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.

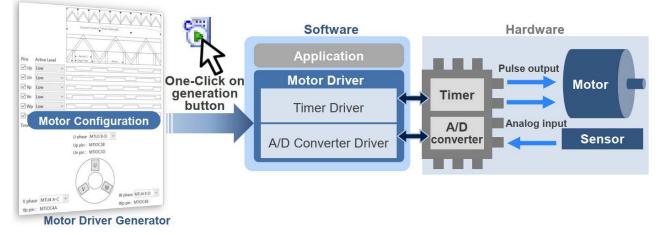


Figure 4-44 Motor Driver Generator

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)
oftware Component Selection Select component from those available in list			8	-	Add new configura	tion for selected component		8
Category All Function All Filter Components	Short Name	Туре	Version	~	Motor Configuration name: Motor type: Resource:	Config_MTU3_MTU4 3-Phase Brushless DC Motor 3-Phase Brushless DC Motor 2-Phase Stepping Motor (Fast Decay) 2-Phase Stepping Motor (Slow Decay)		
Conv Power Consumption MEMDRV Driver Memory Driver Interface for Open Sourc. Motor Motor Open Source FAI File System. Open Source FAI File System. Show only latest version	r_memdrv_rx r_tfat_driver_rx r_ob1203_rx r_tfat_rx	Code Generator Firmware Integra_ Firmware Integra_ Code Generator Code Generator Firmware Integra_ Firmware Integra_	2.20 1.1.0 1.12.0 1.01	-		2-mase stepping motor (slow Decay)		
Hide items that have duplicated functional	ity							
This software component provides driver core (MIU) or general PWM timer (GPT)) and 12-b The nenerated code is specific for the motor. Download the latest FIT drivers and middlewar Configure general settings	oit A/D converter (S12/	AD) for basic motor contro	ol.	•				
Rack	Next >	Finish	Cancel		0	< Back Next > Finish	Can	icel

Figure 4-45 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.

ortifure component ce	onfiguration				Generate Code Generate Repor
Components	P2 🖯 🕀 🎝 💌	Configure			0
	* *				
type filter text		Timer Setting A/D Converter	r Setting		
✓ Startup	^	Period setting			
✓ Generic		Timer Operation Period	100 µs	 (Actual frequency: 10.000 kHz) 	
 ✓ ➢ Drivers 		Counter clock division rate	1		Period
👻 🔒 Motor		TGRA register value	2400		I CHOU
Config_MTU3_MTU4	3	Dead time	10 µs	× (Actual value: 10)	
		Pins Active Level Up Low Un Low Vp Low Vp Low	• Periodi ? • Data Time		Output Level

Figure 4-46 Timer Driver Setting (1)

Smart_Configurator_Example.scfg	22				
Software component con	figuration				Generate Code Generate Report
Components	13, 10, 10, 10 € • € • €	Timer Pulse Output Pin Selection	hase MTU3 B-D		Ŷ
 ✓ Startup ✓ Startup ✓ Generic ✓ r_bsp ✓ Drivers ✓ Motor ✓ Config_MTU3_MTU4 		Up	pin : Not Used pin : Not Used	phase MTU4 8-D p in : Not Used	Output Pin Select
	~	Crest Interrupt Interval 10.0	able skipping function	ot (TGIA3))	Timer Interrupt

Figure 4-47 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.

oftware component c	onfiguration			🕲 👜 Generate Code - Generate Repor
Components	P2 🖻 🗃 🎐 🔻	Configure		0
type filter text	65	Timer Setting A/D Converter Setting		
 ✓ Startup ✓ Seneric ✓ r_bsp 	^	A/D Conversion Setting Detected Input Pin	Analog Input Channel	
 > Invers > Motor Inversion Config_MTU3_MTU4 	4		AN000(sample-and-hold used)	
	2	□ Iw □ Vdc	AN000(sample-and-hold used) ··· AN000(sample-and-hold used) ···	Analog Pin Setting
		□ Vu □ Vv	AN000(sample-and-hold used) ···	
		□Vw	AN000(sample-and-hold used)	
		A/D Conversion End Interrupt Priority Level Call user function from interrupt handler	Level 15 (highest)	A/D Interrupt
		Function Name	AdFunction	

Figure 4-48 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	🚵 🖆 🖳 🕀 🏶 🍷	Configure					
	🐮 😨	💿 New Component					×
type filter text							
 ✓ ⇐ Startup ✓ ⇐ Generic r_bsp 		Add new configuration				}	
Coop.	بوم ا	Motor					
		Configuration name:	Config_MTU3_MTU4				
		Motor type:	3-Phase Brushless DC Motor			ş	~
		Resource:	MTU3_MTU4				4
			MTU3_MTU4 MTU6_MTU7 GPT0_GPT1_GPT2 GPT4_GPT5_GPT6 Triangle_GPT			_	
		0	< Back Next >	Finish	Cance	2	Ĩ

Figure 4-49 Select MOTOR resource



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

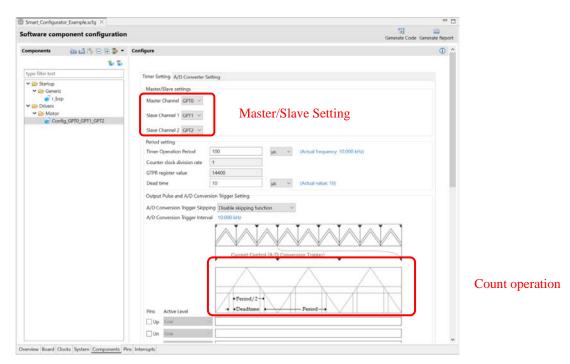


Figure 4-50 Timer Driver Setting(1)

Smart_Configurator_Example.scfg ×	° C	3
Software component configuration	Generate Code Generate Report	
Components 🚵 🖂 🖄 🖶 🗃 🌩 🔹	Win Tow	
2.2	Note: Dead time setting is not reflected to the pictures.	
type filter text	Timer Pulse Output Pin Selection	
 ✓ Startup ✓ Config.GP10_GP11_GP12 	U phase GPTO A-B Up pin: Not Used Un pin: Not Used Un pin: Not Used Un pin: Works W phase GPTI A-B	Output Pin Select
	Vp pin : Not Used Vn pin : Not Used Vn pin : Not Used	
	Wn pin : Not Used Timer Interrupt setting Use Creat Interrupt (GPT Master Overflow Interrupt (GTCM)) Interrupt Skipping Count Imable Skipping function Creat Interrupt Interrupt Anoles Call user function from Interrupt Anoles Creat Interrupt Priority Level Interlupt Skipping Tuckion Creat Interrupt Priority Level Interrupt Skipping Tuckion	Interrupt setting

Figure 4-51 Timer Driver Setting (2)



4.4.17 Configure general setting of component

You can change the general setting of the component such as location and dependency. If you want to change it, click the [Configure general settings...] link on the [Software Component Selection] page displayed in the [New Component] dialog (Figure 4-8), and display the [Preferences] dialog.

pe filter text	Component				
Breakpoints ^ Device add-ins Suppc	Backup settings				
Launch Settings	Enable Backup settin	gs			
Logging Module Download	Number of trash item (1-20): 5			_
My Renesas	Code Generator compo	nent settings			
> Renesas QE	API function output: (Dutput all API functions according to the setting			- 4
Renesas Toolchain Ma Smart Browser	API code style:	alue with macro description			2
 Smart Configurator 	FIT(RX) / SIS(RL78) com	ponent settings			
Component MCU/MPU Packag	Code generation behav	ior: Update configuration files			
Pin Errors/Warning	Dependency settings				
> Smart Manual	Change these options t	o control how a component is added			
Support Folders Tracealyzer	Adding dependency:	Add dependent component			
TraceX	Checking dependency	lanore if dependent component is newer			
Run/Debug	encening dependency:	ignore il dependent component is newer			
Scripting	Location settings				
Terminal	Location settings have	moved to the Module Download page			
TextMate					
Version Control (Team) 🗸		Restore Defa	nulte	Apple	h.e
)		nestore Dera	Juits	Appi	у

Figure 4-52 Configure general setting of component

Notes: 1. If the version of the module and its dependency do not match, a warning message W04020011 is displayed. If you check the revision history of the module and its dependencies and you do not need to change the module you are using, you can ignore this warning. To clear this warning, select "Do not check for dependent component" in the [Checking dependency] list box in component preferences, then click [Apply].

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

Figure 4-53 [Checking dependency] change

2. If you downloaded the FIT module directly from the website, unzip the downloaded zip file and copy the xml file and zip file in the FIT Modules folder to the [Module Download] - [Location (RX)] folder. To change the location, click on the [Module Download] link, then find [Location (RX)], click [Browse...] and select another folder.

🕑 Preferences		- D X	Preferences		0	
type filter text	Component	0.0.1	type filter text	Module Download		• • • •
Breakpoints A Device add-ins Suppo Launch Settings Logging Module Download	Backup settings Enable Backup settings Number of trash item (1-20) 5		Breakpoint: ^ Device add	Location settings		D
My Renesas > Renetas QE	Code Generator component settings		Emulator	Location (RL78): C:\Users\F	Renesas\.eclipse\com.renesas.platfor	Browse
Renesas Toolchain Ma Smart Browser	API function output: Output all API functions according to the setting API code style: Value with macro description		> FSP Config Launch Set	Location (RX); C:\Renesa	s\FITModules	Browse
 Smart Configurator Component 	FIT(R0) / SIS(RL78) component settings		Logging	Location (RZ): C:\Users\F	Renesas\.eclipse\com.renesas.platfor	Browse
MCU/MPU Packag Pin Erron/Warning > Smart Manual Support Folders Tacealyzer	Code generation behavior: Update configuration files Dependency settings Ohange these options to control how a component is added Adding dependency: Add dependent component	v	Module Dc My Renesa Renesas QE	Location (RTOS): C:\Users\F	Renesas\.eclipse\com.renesas.platfor	<u>B</u> rowse
FaceX FaceX > Rur/Debug > Scripting Terminal > TextMate	Checking dependency: Ignore if dependent component is never Location settings Location settings have moved to the <u>Module Download</u> hage	G (Renesas To Smart Brov Smart Coni V		Restore Defaults	Apply
Version Control (Inam) +	Restore De Apply and G		⑦ № ௴ Θ		Apply and Close	Cancel

Figure 4-54 [Location (RX)] change



4.5 Pin Settings

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

oftware Components	🖻 🔩 🚨 Pi	n Function					èu
Type filter text	t	ype filter text (* =	any string, ? = any character)		Al	1	
 ✓ ▲ r_bsp ✓ ▲ r_ether_rx 	E	Function ADTRG1#	Assignment P17/MTIOC3A/MTIOC38/MTIOC48/GTIOC08-B/TIOC80/TCLK PE2/D10/MTIOC4A/GTIOC08-A/PO23/TIC3/RXD12/SMISO12		Direction I	Remarks	
 <i>e</i> r_ether_rx ✓ ▲ r_qspi_smstr_rx 		AN101	 Not assigned 	Not assignedNot assigned	None None		
 r_qspi_smstr_rx Single Scan Mode S12AD Config_S12AD1 		AN103 AN104 AN105	Not assigned Not assigned Not assigned	 Not assigned Not assigned Not assigned 	None None None		
		AN106	Not assignedNot assigned	 Not assigned Not assigned 	None None		
Display switching		AN108 AN109 AN110	Not assigned Not assigned Not assigned	 Not assigned Not assigned Not assigned 	None None None		
		AN111	 Not assigned 	Not assignedNot assigned	None None		
		AN113	 Not assigned 	Not assigned	None		>



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

in con	figuration						Generate Code Generate Repor
in Numb	er						11 III - 프 프
type filte	r text (* = any string, ? = any character)					All ~
Pin Nu	Pin Name	Default Function	Function	Direction	Remarks	Comments	^
1	AVSS0	AVSS0	AVSS0	÷	Read only	AVSS0	
2	P05/IRQ13/DA1	P05	Not assigned	None		LED1	
3	AVCC1	AVCC1	AVCC1	-	Read only	AVCC1	
4	P03/IRQ11/DA0	P03	Not assigned	None		LED0	
5	AVSS1	AVSS1	AVSS1	-	Read only	VREFL	
6	P02/TMCI1/SCK6/IRQ10/AN120	SCK6	Not assigned	None		SCK6	
7	P01/TMCI0/RXD6/SMISO6/SSCL6/I	RXD6	Not assigr 🐱	None		RXD6	
8	P00/TMRI0/TXD6/SMOSI6/SSDA6/	TXD6	Not assigne	d ^ e		TXD6	
9	PF5/IRQ4	PF5	P01	5		SDPWREN	
10	EMLE	EMLE	TMCIO	5		EMLE	
11	PJ5/POE8#/CTS2#/RTS2#/SS2#	PJ5	RXD6	8		XDRIVE	
12	VSS		SMISO6 4		Read only		
13	PJ3/EDACK1/MTIOC3C/ET0_EXOUT	MTIOC3C	SSCL6	2		MTIOC3C	
14	VCL		IRO9	~	Read only		
15	VBATT		VBATT	5	Read only		
16	NC	NC	Not assigned	None		NC	2

Figure 4-56 [Pins] Page ([Pin Number])



4.5.1 Changing the pin assignment of a software component

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

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

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

- (1) Click on [🚠 (Show by Hardware Resource or Software Components)] to switch to the component view.
- (2) Select the target software component (for e.g. Config_S12AD1).
- (3) Click the [Enabled] header to sort by pins used.
- (4) In the [Assignment] column or [Pin Number] column on the [Pin Function] list, change the pin assignment (for e.g. change from P17 to P13).
- (5) In addition, assignment of a pin can be changed by clicking on the [(Next group of pins for the selected resource)] button. Pin that has peripheral function is displayed each time the button is clicked.

-			ode Generate Rep
oftware Components	🖲 🖻 🖪 🤷 Pin Function	(5) 🛃 🖩 🖬 🔤
Type filter text	type filter text (* = any string, ? = any character)	All	
∽ 👗 r_bsp	(3) Enabled Function Assignment Y	Number Direction	Remarks
🔍 r_bsp	AN100 / PE2/D10/MTIOC4A/GTIOC0B-A/PO23/TIC3/RXD12/SMISO12 / 133	3 I	
Y 👗 r_ether_rx	ADTRG1#4 P17/MTIOC3A/MTIOC3B/MTIOC4B/GTIOC0B-B/TIOCB0/I	1	
矿 r_ether_rx	AN101 Not assigned		
Y 👗 r_qspi_smstr_rx	AN102 / P17/MTIOC3A/MTIOC3B/MTIOC4B/GTIOC0B-B/TIOCB0/TCLKD/	O1/PO15/POE8#/SCK1	/TXD3/SMOSI3/
r_qspi_smstr_rx	AN103 (4) P13/WR2#/BC2#/MTIOC0B/TIOCA5/TMO3/P013/TXD2/SM 312/S		
Single Scan Mode S12AD		t assigned None	
2) Config_S12AD1	AN105 / Not assigned / No	t assigned None	
	AN106 / Not assigned / No	t assigned None	
	AN107 / Not assigned / No	t assigned None	
	AN108 / Not assigned / No	t assigned None	
	AN109 / Not assigned / No	t assigned None	
	AN110 / Not assigned / No	t assigned None	
	AN111 / Not assigned / No	t assigned None	
	AN112 / Not assigned / No	t assigned None	
	AN113 / Not assigned / No	t assigned None	
	<		>

Figure 4-57 Pin Settings – Assigning Pins on the [Pin Function] List

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

Note:

The function for assigning pins is not available for some FIT modules.

For the method of assigning pins to such a FIT module, refer to the application note in the <ProjectDir>¥src¥smc_gen¥r_xxx¥doc folder for the FIT module.



4.5.2 Assigning pins using the MCU/MPU Package view

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

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

- (1) Zoom in to the view by clicking the [19] (Zoom in)] button or scrolling the view with the mouse wheel.
- (2) Right-click on the target pin.
- (3) Select the signal to be assigned to the pin.
- (4) The color of the pins can be customized through [Preference Setting...].

MCU/MPU Package × (1)		-	° 🗆
🚦 🔺 🔎 💌 🖻 📄 Type filter text	Assigned Function	n 🔹	
	74 Pos 73 VSS		
	72 P50		
	71 P51		
	70 P52		
JESAS	69 P53	(2)	
	65 P10 67 P11	(-)	-
	se vcc.u	Not assigned	
	65 VSS1_ 64 USBA	P11	•
	63 USBA	MTIC5V	
	62 VSS2_ 61 PVS5_	TMCI3	
	60 AVSS_	SCK2	
RX64M	59 USBA, 55 AVCC	USBA_VBUS	•
	in the second	USBA_VBUSEN	•
Legend		IRQ1	·
Highlighted pin 📃 In-used pin 📃 Warning pin	Conflict pin	System	
Timer Connectivity Analog Port	Graphics	Audio	
Others Preferences Setting (4)			

Figure 4-58 Assigning Pins Using the MCU/MPU Package View



RX Smart Configurator

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 Functi	on				2 0 0	21
Type filter text	type filte	r text (* = ar	iy string, ? = any character)		All		~
Y ≛ r_bsp	Enabled	Function	Assignment	Pin Number	Direction	Remarks	~
👻 r_bsp		AN100	PE2/D10/MTIOC4A/GTIOC0B-A/PO23/TIC3/RXD12/SMIS	<u></u>			-
Y 🚣 r_ether_rx	\checkmark	ADTRG1#	P17/MTIOC3A/MTIOC3B/MTIOC4B/GTIOC0B-B/TIOCB0/T	Jump to Pin Nu			
💞 r_ether_rx		AN101	Not assigned	Merge commer	nt to Pin Num	ber tab	
* 4 r_qspi_smstr_rx		AN102	* Not assigned	Clear comment	s		
r_qspi_smstr_rx		AN103	Not assigned	Assign selected	pins		
 Šingle Scan Mode S12AD 		AN104	/ Not assigned	Unassign select			
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		
		AN110	Not assigned	Not assigned	None		
		AN111	Not assigned	Not assigned	None		
		AN112	/ Not assigned	Not assigned	None		
		AN113	Not assigned	Not assigned	None		
	<					3	>
n Function Pin Number							
verview Board Clocks System Components Pins	Interrunte						
erview bound cocca system components [1113]	interrupts						
			~				

in Number	Pin Name	Board Functions	Function	Direction	Remarks	Symbolic Name	
127	VCC		VCC	-	Read only	-	
128	P70/SDCLK		Not assigned	None			
129	VSS		VSS	-	Read only	-	
130	PE5/D13/MTIOC4C/MTIOC2B/GTIOC0A-A/ET0_RX_CL		Not assigned	None			
131	PE4/D12/MTIOC4D/MTIOC1A/GTIOC1A-A/PO28/ET0		Not assigned	None			
132	PE3/D11/MTIOC4B/GTIOC2A-A/PO26/POE8#/TOC3/C		Not assigned	None			
133	PE2/D10/MTIOC4A/GTIOC0B-A/PO23/TIC3/RXD12/S		AN100	1			
134	PE1/D9/MTIOC4C/MTIOC3B/GTIOC1B-A/PO18/TXD1		Not assigned	None			
135	PE0/D8/MTIOC3D/GTIOC2B-A/SCK12/MMC_D4-B/AN		Not assigned	None			
136	P64/CS4#/WE#		Not assigned	None			
137	P63/CS3#/CAS#		Not assigned	None			
138	P62/CS2#/RAS#		Not assigned	None			
139	P61/CS1#/SDCS#		Not assigned	None			
140	VICC		VICC		Pood only		

Pin Function Pin Number

Overview Board Clocks System Components Pins Interrupts

Figure 4-59 Jump to pin number



4.5.4 Exporting pin settings

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

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

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

ardware Resource 🗉 🗉 🗄	<mark>ه، يا</mark>	Pin Functi	ion				ି ଏ 💷 🖪	1)
Type filter text		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			AN100	PE2/D10/MTIOC4A/GTIOC0B-A/PO23/TIC3/RXD12/SMISO12	/ 133	1		
₽¦8 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_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	~	<		* DCD/A4C/A4TIOCOC/TCLVC/DO47/CTCL#/DTCL#/CCLA4	• 01			

Figure 4-60 Exporting Pin Settings to an XML File

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

4.5.5 Importing pin settings

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

in configuration						Generate Co	👜 de Generate	Repo
Hardware Resource 🗉 🗄	*	Pin Functi	on				3 🗉 🖬	
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			AN100	PE2/D10/MTIOC4A/GTIOC0B-A/PO23/TIC3/RXD12/SMISO12	/ 133	1		
₽ <mark>8</mark> 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			ETO_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	~		ETO EDVDO	* DCD/A4C/A4TIOCOC/TCLVC/DO47/CTCL#/DTCL#/CCLA4	* 01			, ×
< >		< .						,

Figure 4-61 Importing Pin Settings from an XML File

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



4.5.6 Pin setting using board pin configuration information

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

The following describes the procedure for collective setting of pins.

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

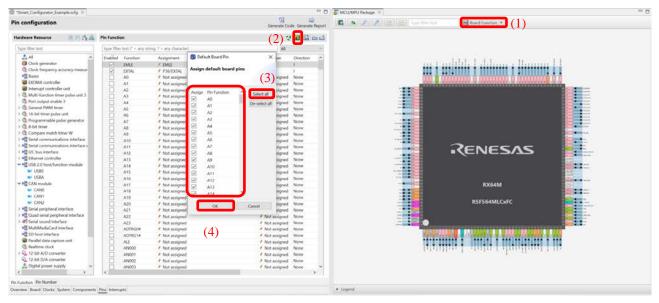


Figure 4-62 Setting for initial pin configuration

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 Function	ı					<u>्</u>) 🔳 🔛 🔤 e
type filter t	text						All 🔊
Enabled	Function	Assignment	Pin Number	Direction	Remarks	Comn	All Function
	A0	Not assigned	Not assigned	None			Assignment
	A1	Not assigned	Not assigned	None			Pin Number
	A2	Not assigned	Not assigned	None			Direction Remarks
	A3	Not assigned	Not assigned	None			Comments

Figure 4-63 Filter for [Pin Function] tab

type filter tex	tt (* = any string, ? =	any character)					All
Pin Number	Pin Name	Board Functions	Function	Direction	Remarks	Symbolic Name	C All Pin Number
1	AVSS0	AVSS0	AVSS0	-	Read only	-	A Pin Name
2	P05/IRQ13/DA1	P05	Not assigned	None			LE Board Functions
3	AVCC1	AVCC1	AVCC1	-	Read only	-	AFunction
4	P03/IRQ11/DA0	P03	Not assigned	None			LE Direction
	AVSS1	AVSS1	AVSS1		Read only	-	A Remarks

Figure 4-64 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 [Renesas] > [Smart Configurator] > [Pin Errors/Warnings] and use the combo boxes to change the errors/warning setting.

Preferences				\times
type filter text		Pin Errors/Warnings	⇔ ▼ ⇔	▼ 8 8
Renesas Toolchain Management Smart Browser	^	 Pin Conflict Multiple functions are assigned in one pin number 	Frror	~
 Smart Configurator Component MCU Package Appearance 		 No Pin Allocation Function used by software but not allocated to any pin 	Error	~
MMU Pin Errors/Warnings		- Mutually Exclusive Pins		
Smart Demo > Smart Manual		Mutually exclusive pins cannot be allocated to the same pin at the same time • No Software	Error	~
Support Folders Synergy Configuration Editor		Assigned pins but there's no software using them • Different Group	Info	~
Synergy License Tracealyzer		Functions in same channel but different group Board Mismatch	Warning) ~
TraceX > Run/Debug		Pin assignment does not match the board suggested pin assignment	Warning	y ~
 Scripting Team 				
> Terminal	~	Restore Defaults	<u>A</u> pply	
? ù 🖆 🖲		Apply and Close	Cancel	

Figure 4-65 Pin Errors/Warnings settings at Preferences

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

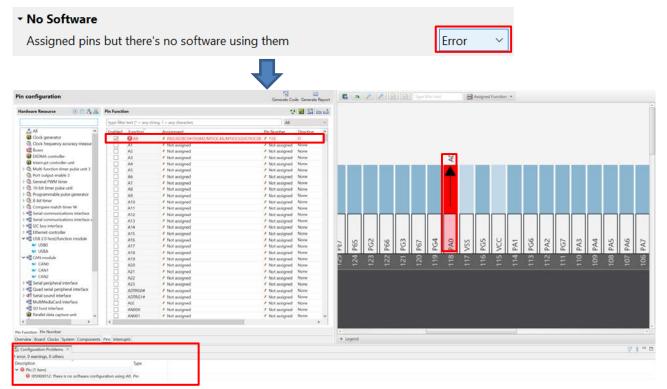


Figure 4-66 Change "No Software" setting from "Info" to "Error"



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	ration						(2)	😼 enerate Code	iii) Generate Report
in Number							(_)		
type filter text	(* = any string, ? = any character)							All	~
Pin Number	Pin Name AVSS0	Board Functions AVSS0	Function AVSS0	Direction	Remarks Read only	Symbolic Name	Comments AVSS0		^
2	P05/IRQ13/DA1 AVCC1	P05 AVCC1	Not assigned AVCC1	None	Read only	LED1	LED1 AVCC1		_
4	P03/IRQ11/DA0 AVSS1	P03 AVSS1	Not assigned AVSS1	None	Read only	LED0	LED0 AVSS1		
6	P02/TMCI1/SCK6/IRQ10/AN120 P01/TMCI0/RXD6/SMISO6/SSCL6/IRQ9/AN119	SCK6 SMISO6	Not assigned	None None	and the second		SCK6 SMISO6		
8	P00/TMRI0/TXD6/SMOSI6/SSDA6/IRQ8/AN118 PF5/IRQ4	SMOSI6 PF5	Not assigned			(1)	SMOSI6 SDPWREN		
10	EMLE PJ5/POE8#/CTS2#/RTS2#/SS2#	EMLE PJ5	TMCI0 RXD6	one		-	EMLE		
12	V55 PJ3/EDACK1/MTIOC3C/ET0 EXOUT/CTS6#/RTS6#/CTS		SMISO6	one	Read only		MTIOC3C		
14	VCL VBATT	milococ	SSCL6 IRQ9 VBATT	v	Read only Read only	-	mitococ		
16	NC	NC	Not assigned	None			NC		

Figure 4-67 pin setting of symbolic name



After generating code check at Pin.h file.

🖕 Project Explorer 🛛 🔚 😫 🍟 🗖] 🗈 Pin.h 🛛			- 0
 Smart Configurator Example [HardwareDebug]) Indudes Sinc Sinc Sinc Config S12AD0 Sinc general Sinc r_Dsp Sinc r_config Sin r_cher_nx Config Sinc Sinc r_qspi smstr_nx Sinc sinstr_configurator_Example.c Sinc Sinc 	39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 55 56 57 58	Example: Toggle LED1 at Pin P54. There are 2 ways to toggle LED1 1) Using symbolic name macro Assuming the symbolic name for P #define LED1 5,4	ectly	^
Smart_Configurator_Example.scfg Smart_Configurator_Example HardwareDebug.launch Ø Developer Assistance	59 60 61 62 63 64	<pre>/* Pin write helper */ #define PIN_WRITE_HELPER(x,y) /* Pin read helper */ #define PIN_READ_HELPER(x,y) ##if !(defined(CCRX) && defined(cCRX)</pre>	<pre>((PORT##x.PODR.BIT.B##y)) ((PORT##x.PIDR.BIT.B##y)) pluspluspluspl</pre>	

Figure 4-68 Symbolic Name in generated code

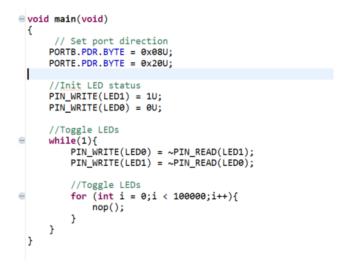


Figure 4-69 Using Symbolic Name in main function



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.

rrupt co	nfiguration						Generate Code Ge	nerate Repo
errupt 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:	rity settings made here n						

Figure 4-70 [Interrupts] Page



4.6.1 Changing the interrupt priority level and fast interrupt setting

When an interrupt is used in a CG configuration on the [Components] page, the status of the interrupt will be changed to "Used". To display the used interrupts only, click on the [$\[Mathbb{Mathbbb{Mathbbb{Mathbbb{Mathbb{Mathbb{Mathbb}Mathbb{Math$

- (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.
- (3) 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.

errupt co	nfiguration							Generate Code Gene	iate Rep
nterrupt vecto	ors used								E
Up	Type filter text								
Down	Vector Number 42	Interrupt SPRI	Peripheral QSPI	Priority Level 10		Status Used	Multiple Interrupts	Fast Interrupt	
(2)	43	SPTI	QSPI	Level 10		Used	E		
(3)	> 11	GROUPBL1		Level 15	~	Used			
	> 113	GROUPAL1	(1)	Level 3	^	Used			
	190	INTB190 (S12ADI)	S12AD (1)	Level 4 Level 5 Level 6 Level 7 Level 8 Level 9 Level 10 Level 11 Level 12 Level 13 Level 14 Level 15		Used		(2)	
		rity settings made here m configuration files of each			IS.	0			

Figure 4-71 Interrupt Settings

Note:

The function for setting up interrupts is not available for the FIT modules.

For the method of setting up interrupts for each FIT module, refer to the application note in the <ProjectDir>¥src¥smc_gen¥r_xxx¥doc folder for the FIT module.



4.6.2 Changing the interrupt vector number

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 cor	nfiguration						Generate Code Gene	erate
Interrupt vecto	rs used							
Up	Type filter text							
Down	Vector Number 42	Interrupt SPRI	Peripheral QSPI	Priority Level 10	Status Used	Multiple Interrupts	Fast Interrupt	
	43	SPTI	QSPI	Level 10	Used			
	✓ 111	GROUPBL1		Level 15	Used			
(1)	20	S12CMPI	S12AD		Used			
(1)	> 113	GROUPAL1		Level 2	Used			
	190	INTB190 (S12ADI)	S12AD	Level 15	Used			
	Note:							
		rity settings made here n		n some FIT components. r the correct priority settings.				

Figure 4-72 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.

	u 🛃 🖁 🗉 🕀	-					
e filter text		🔆 🔻 Configure					
e filter text	1	Count clock se	etting				
		PCLK/8	O PCLK/32	O PCLK/128	O PCLK/512		
➢ Startup							
> 🧁 Generic		Compare mat					
Drivers		Interval value	2	100	μs	 (Actual value) 	ue: 100)
> 🗁 A/D Conv	verter	Register valu	e (CMCOR)	749			
> 🗁 Commun	nications	Compare	match interrupt (CMI0)				
✓ → Timers				(2)			
💣 Config	g_CMT0	✓ Enable m	ultiple interrupts (CMI0)	(2)			
	(1)	Priority		Level 15 (highest)	~		
		Pins Interrupts Figure 4-7	73 Multiple int	errupts in comp	oonent pag	e	
rrupt confi	iguration		73 Multiple int	errupts in comp	oonent pag	e (3) Generate Cod	J
rrupt confi	iguration		73 Multiple int	errupts in comp	oonent pag		
rrupt confi	iguration		73 Multiple int	errupts in comp	oonent pag		le Generate l
rrupt confi	iguration Type filter text Vector Number	Figure 4-	Peripheral	Priority	ponent pag	(3) Generate Cod	le Generate I Fast I A
rrupt confi errupt vectors	iguration Type filter text Vector Number 16	Figure 4-	Peripheral BSC	Priority Level 15		(3) Generate Cod Multiple Interrupts	Fast I
rrupt confi errupt vectors Up	Type filter text Vector Number 16 18	Figure 4-	Peripheral BSC RAM	Priority Level 15 Level 15		(3) Generate Cod	Fast I
errupt confi errupt vectors Up	Type filter text Vector Number 16 18 21	Figure 4-	Peripheral BSC RAM FCU	Priority Level 15 Level 15 Level 15		(3) Generate Cod Multiple Interrupts	Fast I A
errupt confi errupt vectors	iguration Type filter text Vector Number 16 18 21 23	Figure 4-	Peripheral BSC RAM FCU FCU	Priority Level 15 Level 15 Level 15 Level 15 Level 15	Status	(3) Generate Cod	Fast I A
errupt confi errupt vectors Up	Type filter text Vector Number 16 18 21 23 26	Figure 4-	Peripheral BSC RAM FCU FCU ICU	Priority Level 15 Level 15 Level 15 Level 15 Level 15 Level 15		(3) Generate Cod	Fast I A
errupt confi errupt vectors Up	Type filter text Vector Number 16 18 21 23 26 27	Figure 4-	Peripheral BSC RAM FCU FCU ICU ICU	Priority Level 15 Level 15 Level 15 Level 15 Level 15 Level 15 Level 15	Status (2)	(3) Generate Cod	Fast I 4
rrupt confi errupt vectors Up	Type filter text Vector Number 16 18 21 23 26 27 28	Figure 4-	Peripheral BSC RAM FCU FCU ICU ICU ICU ICU ICU	Priority Level 15 Level 15 Level 15 Level 15 Level 15 Level 15 Level 15 Level 15	Status	(3) Generate Cod	Fast I
errupt confi errupt vectors Up	Type filter text Vector Number 16 18 21 23 26 27 28 29	Figure 4-	Peripheral BSC RAM FCU FCU ICU ICU ICU ICU ICU ICU ICU ICU ICU I	Priority Level 15 Level 15	Status (2)	(3) Generate Cod	Fast I
errupt confi errupt vectors Up	Type filter text Vector Number 16 18 21 23 26 27 28 29 30	Figure 4-	Peripheral BSC RAM FCU FCU ICU ICU ICU ICU ICU CMT0 CMT1 CMTW0	Priority Level 15 Level 15	Status (2)	(3) Generate Cod	Fast I
rrupt confi errupt vectors Up	Type filter text Vector Number 16 18 21 23 26 27 28 29 30 31	Figure 4-	Peripheral BSC RAM FCU FCU ICU ICU ICU ICU ICU CMT0 CMT1 CMTW0 CMTW1	Priority Level 15 Level 15	Status (2)	(3) Generate Cod	Fast I ^
rrupt confi errupt vectors Up	Type filter text Vector Number 16 18 21 23 26 27 28 29 30 31 32	Figure 4-	Peripheral BSC RAM FCU FCU ICU ICU ICU ICU ICU CMT0 CMT1 CMTW0 CMTW1 USBA	Priority Level 15 Level 15	Status (2)	(3) Generate Cod	Fast
rrupt confi errupt vectors Up	Type filter text Vector Number 16 18 21 23 26 27 28 29 30 31	Figure 4-	Peripheral BSC RAM FCU FCU ICU ICU ICU ICU ICU CMT0 CMT1 CMTW0 CMTW1	Priority Level 15 Level 15	Status (2)	(3) Generate Cod	Fast I ^

Figure 4-74 Multiple interrupts in component in Interrupts page

RX Smart Configurator

Project Explorer 🛛 🗖 🗖	Config_CMT0_user.c ×	- [
Smart_Configurator_Example [Har Smart_Configurator_Example [Har Smart_Config_CMT0 Config_CMT0 Config_CMT0 Config_CMT0 Config_CMT0 Config_CMT0 Config_CMT0 Config_S12AD0 Config_	<pre>43 # Global variables and functions[43 # Global variables and functions] 44 * Start user code for global. Do not edit comment generated here */ 45 * function Name: R_Config_CMT0_Create_UserInit[47 * function Name: R_Config_CMT0_Create_UserInit] 48 * void R_Config_CMT0_create_UserInit(void) 49 * void R_Config_CMT0_create_UserInit. Do not edit comment generated here */ 49 * function Name: r_config_CMT0_crei0_interrupt] 49 * Function Name: r_config_CMT0_crei0_interrupt(vect=VECT(CMT0,CMI0),fint) 49 * # Function Name: r_Config_CMT0_cmi0_interrupt(vect=VECT(CMT0,CMI0),fint)</pre>	
 > Commy > Commy	<pre>70 ##else 71 #pragma interrupt r_Config_CMT0_cmi0_interrupt(vect=VECT(CMT0,CMI0)) 72 #static void r_Config_CMT0_cmi0_interrupt(void) 73 static void r_Config_CMT0_cmi0_interrupts */ 74 { 75</pre>	

Figure 4-75 Multiple interrupts in generated code



4.7 MCU migration feature

The MCU migration feature helps to convert your project settings from device A to device B. Conversion of project settings can be done within the same family and can be done from e² studio project menu as follows.

- Note: Project settings may change due to device change. Back up the project before executing the device change.
 - (1) Select the project and choose [Change Device] from the [Project] menu.

	vorkspace - Smart_Config	_				- e² studio		- 0	\times
File	Edit Navigate Search	Pro		Run Wine	dow Help			~	
	🛞 - % - 🖉 🐤		Open Project			Q	🗄 😰 🗟 C/C++ 🧧	-	
8	눱 Project Explorer 🛛 🛛		Close Project			ble.scfg ×		- 1	
۲		010	Build All		Ctrl+Alt+B	n	1	a	5
 Ø	> 📂 Smart_Configurate		Build Configuration	าร	>	<i>a</i> r	Generate Code	Generate Repor	rt
B			Build Project		Ctrl+B			(?)	^ (?
_			Build Working Set		>			$\mathbf{\Psi}$	
			Clean						
			Build Automatically	/		ew of the features provided by Smart Configurator.			
			C/C++ Index		>	of the reactines provided by private earling and the			
		e ²	Update All Depend	lencies	Alt+D	Application Code			
			Change Device			to Smart Configurator			
			Change Toolchain V	Version		d videos Software Components	\sim		
		3	C/C++ Project Setti	ings	Ctrl+Alt+P	Middleware & 5			
			Properties			ew Drivers 9			
		_		S	Check out w	nat's new in the latest release.			
					See all <u>Relea</u>	to Smart Configurator divideos ew hat's new in the latest release. is Notes. Documentation			
				ſ <u></u> ∎a	Product	Documentation MCU Hardware			
					User's Guide				
					API manual				
					Application	<u>lotes</u>			
					Tool news				
			-	Current Co	onfiguration				
			Se	elected boa	ard/device: R5	564MLCxFC (ROM size: 4Mbytes, RAM size: 512KB, Pin count: 176)			
			G	ienerated lo	cation (PROJE	T_LOC\): src\smc_gen Edit			~
			Ove	erview Boar	rd Clocks Sv	tem Components Pins Interrupts			
L									

Figure 4-76 Select [Change Device]

(2) Select the target board from the board list, the device will be auto selected.

Refactorin	9					×
Change Dev	vice					Real
Select the ne	w device for Smart_Cor	figurator_Exa	mple			
Current Devic	e: R5F565NEDxFC					
Current Board	: RSKRX65N-2MB					
Target Board:	RSKRX24T					~
Target Device	Custom RSKRX111 RSKRX113					^
	RSKRX130 RSKRX130-512KB RSKRX231 RSKRX23T RSKRX24T RSKRX24U					
	RSKRX64M RSKRX65N RSKRX65N-2MB RSKRX66T					
	ale					
(?)		< Back	Next >	Finish	Cano	

Figure 4-77 Select target board

If you want to remain target board as "Custom", select the target device manually from the device selection list and click "OK". (Wild card search is supported)

Refactoring	- 🗆 🗙	8				o x
Change Device	Bash	Device Selection				
Select the new device for Smart_Configurator_Example		You can filter devices by regular expr	ression			
Current Device: R5F565NEDxFC		65*				
Current Board: RSKRX65N-2MB		Device	RAM	ROM	Pin	
Target Board: Custom		~ RX600				
larget Board, Custom		✓ RX651				
	Download additional boards	> RX651 - 64pin				
Target Device: R5F565NEDxFC		> RX651 - 100pin				
	Unlock Devices	> RX651 - 144pin				
	STRUCK DEVICES.	> RX651 - 145pin				
		> RX651 - 176pin				
	,	> RX651 - 177pin				
		✓ RX65N				
		✓ RX65N - 100pin				
		R5F565N4AxFP	264 KB	512 KB	100	
		R5F565N4AxLJ	264 KB	512 KB	100	
		R5F565N4BxFP	264 KB	512 KB	100	
		R5F565N4BxLJ	264 KB	512 KB	100	
		R5F565N4ExFP	264 KB	512 KB	100	

Figure 4-78 Select target device

(3) Confirm the message displayed in [Found problems] and click [Next].

😨 Refacto	oring					×
Change [Device					
Review th or 'Finish'		rided in the list below.	Click 'Next >' to vi	iew the n	ext item	E
Found pro	blems					0 f
This cha	nge cannot be und	one. Please make sure	you backup this p	project be	fore con	tinuing
		No context informat				
0	< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish		Cance	I

Figure 4-79 Found problems

Message	Explanation
Target device is not supported by Smart Configurator.	Displayed before changing to a device not supported by the Smart Configurator. You can't convert Smart Configurator, but you can convert Project, Builder, Linker, Debugger.
This change cannot be undone. Please make sure you backup this project before continuing.	If you change the device, it can't be restored before change, so please execute it after backing up the project.



(4) Confirm items to be changed and click "Finish".

e ² Refactoring	_			×
Change Device The following changes are necessary to perform the refactoring.				
Changes to be performed		Ŷ	Ŷ	◄
 ✓ E Change Device for Smart_Configurator_Example ✓ Launch Configurations ✓ Smart_Configurator_Example HardwareDebug ✓ Build Settings ✓ Project Files ✓ Smart Configurator 				
No preview available				
			Cance	el

Figure 4-80 Changes to be performed

(5) The device name on the [Overview] page is updated.

Smart_Conf	figurator_Example.scfg $ imes$	
Overview	information	Generate Code Generate Report
	nformation	0
	Overview Get an <u>overview</u> of the features provided by Smart Configurator.	
►	Videos Introduction to Smart Configurator Browse related videos What's New Check out <u>what's new</u> in the latest release. See all <u>Release Notes</u> . Product Documentation User's Guide API manual Application Notes	Application Code Software Components Middleware & Drivers Device Drivers MCU Hardware
• Current C	Tool news	
	ard/devic : R5F564MLCxFC (ROM size: 4Mbytes, RAM size: 512KB, Pin count: 176) ocation (PROJECT LOCy): src\smc gen Ed	
	ard Clocks System Components Pins Interrupts	

Figure 4-81 Device Update Confirmation

(6) A report of the configurations' conversion status is generated out in the console.

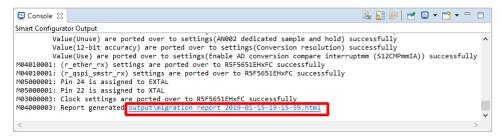


Figure 4-82 Configuration conversion status report



5. Managing Conflicts

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

5.1 Resource Conflicts

When two software components are configured to use the same resource (for e.g. S12AD0), an error mark (B) will be displayed in the Components tree.

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

Image: Smart_Configurator_Example.scfg Image: Image: Smart_Configurator_Example.scfg		- 0
Software component configu	Iration State	ade Generate Report
	Generate CC	de Generate Report
Components $I_{\mathbb{Z}}^{a} \boxdot \textcircled{B} \textcircled{P} \blacksquare$	Configure	^
<pre>type filter text type filter text</pre>	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 out We also recommend not using the P00 to P02, P90 to P93, PD0 to PD7, and PE0 to PE7 pins as output Analog input mode setting Double trigger mode Analog input channel setting ✓ AN000	
Overview Board Clocks System Compon	ant Pins Interrunts	· · · · ·
Configuration Problems A errors, 0 warnings, 0 others		
Description		Туре
 Ø Interrupt (2 items) 		type
	sed by S12ADI in Config_S12AD01 conflicts with vector used by S12ADI in Config_S12AD0.	Interrupt
	sed by S12ADI in Config_S12AD0 conflicts with vector used by S12ADI in Config_S12AD01.	Interrupt
✓ ❷ Peripheral (2 items)		
E04010001: Peripheral S12AD	0 used by Config_S12AD01 is already used by Config_S12AD0.	Peripheral
E04010001: Peripheral S12AD	0 used by Config S12AD0 is already used by Config S12AD01.	Peripheral

Figure 5-1 Resource Conflicts



5.2 Resolving pin conflicts

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

in configuration					😼 Generate Code	Generate Rep
lardware Resource		Pin Functi	on		a	2 🖬 🖬 🔤
Type filter text		type filte	r text (* = ar	y string, ? = any character)	All	
🛓 SD slave interface	^	Enabled	Function	Assignment	Pin Number	Direction
🛢 Parallel data capture unit			Ø ADTRG0 [#]	# / P07/IRQ15/ADTRG0#	<i>∎</i> 176	
🛓 Graphic-LCD controller			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
@ S12AD0			AN002	Not assigned	Not assigned	None
S12AD1			AN003	Not assigned	Not assigned	None
强 12-bit D/A converter			AN004	Not assigned	Not assigned	None
🛓 Digital power supply			AN006	Not assigned	Not assigned	None
Operating mode control	~		AN007	Not assigned	Not assigned	None
<	>	<				

Overview Board Clocks System Components Pins Interrupts

Figure 5-2 Pin Conflicts

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

² Configuration Problems [∞]	
3 errors, 0 warnings, 0 others	
Description	Туре
♥ 9 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 Pin Allocator, pin used by ADTRG0# in Config_S12ADC). Pin
E05000010: Pin 176 cannot be used multiple times. Pin 176 is assigned to IRQ15 and ADTRG0#.	Pin

Figure 5-3 Pin Conflict Messages

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

rdware Resource 🕀 🖻	↓ªz 🏯	Pin Funct	on			🥹 🔳 🔜 🔤
ype filter text		type filte	r text (* = any	v string, ? = any character)	All	
 SD slave interface Parallel data capture unit Graphic-LCD controller Realtime clock 12-bit A/D converter S12AD0 S12AD1 Assign all 	^	Enabled	Function ADTRGO# AN000 AN005 AN001 AN002	Assignment P07/IRQ15/ADTRG0# P40/IRQ8/AN000 P45/IRQ13/AN005 Not assigned Not assigned	Pin Number 176 173 167 Not assigned Not assigned	Direction I I I None None
 S12AD1 Assign all 12-bit D/A c Digital powe Resolve confli Operating mode control 	ct		AN003 AN004 AN006 AN007	 Not assigned Not assigned Not assigned Not assigned 	 Not assigned Not assigned Not assigned Not assigned 	None None None None

Figure 5-4 Resolving Pin Conflicts

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



6. Generating Source Code

6.1 Outputting Generated Source Code

Output a source file for the configured details by clicking on the [¹ (Generate Code)] button in the Smart Configurator view.

Smart_Configurator_Exa		n	Generate Code Generate Repor
Components	.ª₂ 🗐 🕀 幹 ▼	Configure	C
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 <ProjectDir>¥src¥smc_gen and updates the source file list in the Project Explorer. If the Smart Configurator has already generated a file, a backup copy of that file is also generated (refer to chapter 8, Backing up Generated Source Code).

Note: If you put a self-created source file in sms_gen folder, it will be erased at time of generating source code.

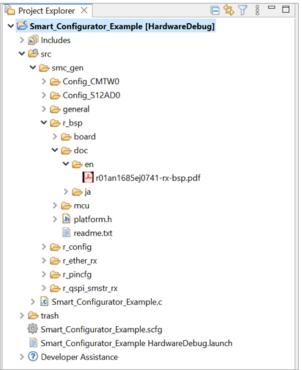


Figure 6-2 Source Files in the Project Explorer



6.2 Change Generated Code Location

(1) To change the generated code location, click on the [Edit] button under Current Configurations at [Overview] page.

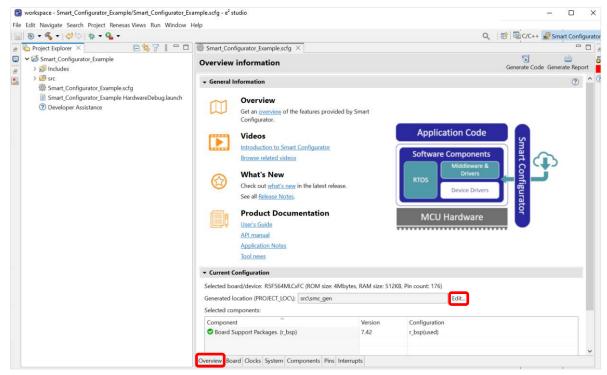


Figure 6-3 Edit the Generated Code Location

(2) In the Folder Selection dialog, select an empty folder for code generation or create a new folder.

New Folder Folder name: New_Folder OK Cancel	art_Configurator_Example		
New_Folded	🔁 New Folder		
	Folder name:		
OK Cancel	New_Folder		
		OK	Cancel

Figure 6-4 Folder Selection



(3) Click on [Definition Generate Code] button. The source code will be generated in the new location. You can also check for the current generate code location in [Overview] page.

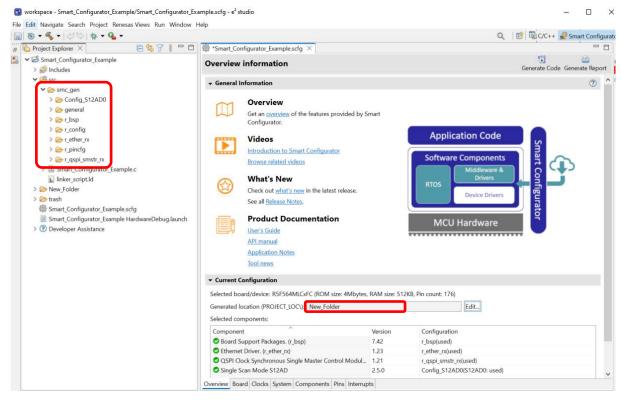


Figure 6-5 New generate code location



6.3 Configuration of Generated Files and File Names

The below figure "Configuration of Generated Files and File Names", shows the folders and files output by the Smart Configurator. Function *main()* is included in *{Project name}.c,* which is generated when the project is created by the

e² studio.

r_xxx indicates the names of FIT modules, "ConfigName" indicates the name of the configuration formed by the component settings, and "Project name" indicates a project name set in the e² studio.

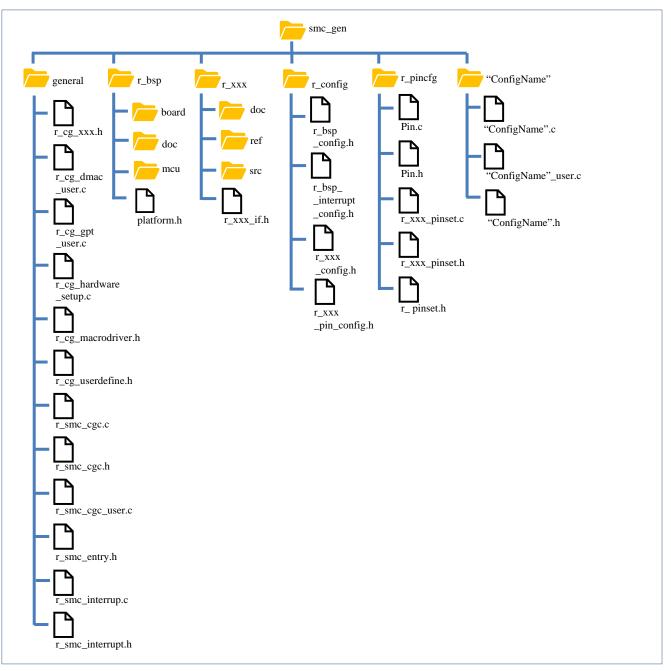


Figure 6-6 Configuration of Generated Files and File Names



Folder	File	Description
general		This folder is always generated.
		It contains header files and source files commonly used by
		CG drivers of the same peripheral function.
	r_cg_xxx.h ^(Note*1)	These files are always generated.
		The files contain macro definitions for setting SFR registers.
	r_cg_dmac_user.c	This file is always generated for a device with a DMAC
		function.
		It contains interrupt service routines and callback functions shared among some DMAC channels (depending on the hardware specifications).
	r_cg_gpt_user.c	This file is always generated for a device with a GPT function.
		It contains interrupt service routines and callback functions shared among some GPT channels (depending on the hardware specifications).
	r_cg_hardware_setup.c	This file is always generated. It contains <i>R_Systeminit</i> that calls all driver initialization functions with the name <i>R_ConfigName_Create</i> .
		<i>R_Systeminit</i> 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_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 . 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_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.

r_bsp	 This folder is always generated. It consists of multiple subfolders (<i>board, doc, mcu</i>) with: Initialization codes to start up the MCU before entering <i>main()</i> (e.g. setup stack, initialize memory) Definitions of all SFR registers in <i>iodefine.h</i> (<i>mcu</i> folder)
	 Application note of r_bsp It also contains <i>platform.h</i> that will include r_bsp.h of the device used in the project.



Folder	File	Description
r_xxx ^(Note*1)		This folder is generated for the FIT module that is added to
		the project.
		It consists of:
		- doc folder: Application note of this FIT module
		- ref folder: Reference of FIT module configuration file
		and pin configuration file
		- src folder: FIT module source files and header files
		 r_xxx_if.h (Note*1): List of all API calls and interface
		definitions of this FIT module
		Note: Folders in <i>r_xxx</i> depends on the requirements of
		each FIT module.
r_config		This folder is always generated.
		It contains configuration header files for the MCU package,
		clocks, interrupts, and driver initialization functions with the
		name R_xxx_Open (Note*1).
	r_bsp_config.h	This file is always generated.
		It contains configurations of <i>r_bsp</i> for clock initialization
		and other MCU related settings. Some MCU related
		settings are generated by Smart Configurator (e.g.
		package type) and other settings (e.g. stack size) are
		configured by user manually.
	r_bsp_interrupt_config.h	This file is always generated.
		It contains mapping of the software configurable interrupts
	(1)	A and B (depending on hardware specification).
	r_xxx_config.h ^(Note*1)	These are configuration header files for all FIT drivers that
		are added to the project. This file is configured by user
	r_xxx_pin_config.h ^(Note*1)	manually. These pin configuration header files are dedicated for FIT
		drivers with specific requirements in pin setting sequence.
r_pincfg	Pin.c	This file is always generated.
	1 11.0	It is a reference of pin function initialization for all
		peripherals configured in the [Pins] tabbed page (except
		I/O Ports).
	Pin.h	This file is always generated.
		It contains the function prototypes of pin settings in <i>Pin.c</i>
	r_xxx_pinset.c (Note*1)	This file contains pin function initialization for the FIT
		drivers that are added to the project. API function in this
		file is for user to call in the application codes.
	r_xxx_pinset.h (Note*1)	This file contains pin setting function prototypes in
		r_xxx_pinset.c
	r_pinset.h	This file includes all pin setting header files named with
		<i>r_xxx_pinset.h</i> ^(Note*1) in <i>r_pincfg</i> folder.
{ConfigName}		This folder is generated for the CG drivers that are added
		to the project.
		API functions in this folder are named after the
		ConfigName (configuration name).
	{ConfigName}.c	This file contains functions to initialize driver
		(<i>R_ConfigName_Create</i>) and perform operations that are
		driver-specific, e.g. start (<i>R_ConfigName_Start</i>) 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
Ι		code areas.



{ConfigName}.h	This is header file for {ConfigName}.c and
	{ConfigName}_user.c.

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



6.4 Initializing Clocks

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

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



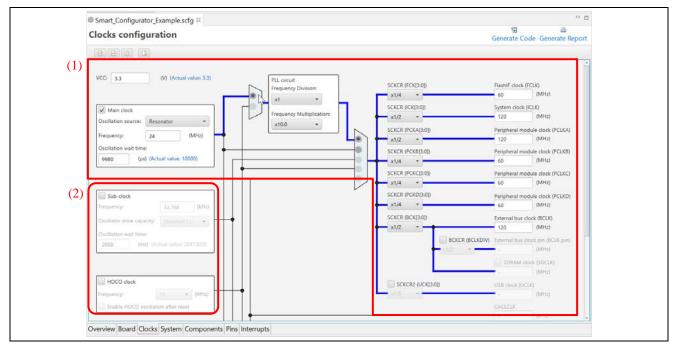


Figure 6-7 Clocks Configuration with Main Clock Selected as Clock Source



No	Folder	File	Macros/Functions	Description
(1)	r_confi g	r_bsp_config.h	Macros related to clocks	These settings are generated by Smart Configurator based on user's selection in the [Clocks] page for the clock source. Only one clock can be selected as the clock source at a time. <i>r_bsp</i> will handle the clock initialization before entering <i>main()</i> .
			Macros related to MCU settings	Some MCU related settings are generated by Smart Configurator (e.g. package type) and other settings (e.g. stack size) are configured by user manually. Refer to the application note in <i>r_bsp</i> folder before configuring these macros: <i>¥src¥smc_gen¥r_bsp¥doc</i>
(2)	general	r_smc_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_smc_cgc.h	Macros related to clocks	These macros are for clock initialization in <i>R_CGC_Create</i> .
		r_smc_cgc_user .c	R_CGC_Create_UserI nit	This API function is used to add code in R_CGC_Create after the CGC initialization.

r_bsp_config.h will be backed up to trash folder before each code generation (refer to chapter 8, Backing up Generated Source Code).



6.5 Initializing Pins

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

(1) Pin initialization for drivers with {ConfigName}

Pin functions are initialized in *R_ConfigName_Create* of the file *¥src¥smc_gen¥{ConfigName}*ConfigName*C.*

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

in configuration				Generate C	Code Generate Repo
oftware Components 🛛 🕀 🛱 😹	Pin Functi	on			२। 🖬 🔛 🗠 ।
Type filter text	type filte	r text (* = ar	iy string, ? = any character)	A	II ~
∽ ≛ r_bsp	Enabled	Function	Assignment	Pin Number	r Direction
♥ r_bsp		ADTRG0#	/ P07/IRQ15/ADTRG0#	/ 176	1
🕆 📥 Single Scan Mode S12AD	\checkmark	AN000	/ P40/IRQ8/AN000	/ 173	I. I.
Config_S12AD0	\checkmark	AN005	P45/IRQ13/AN005	/ 167	I.
∽ ≛ r_ether_rx		AN001	Not assigned	Not assigned	ned None
💣 r_ether_rx		AN002	Not assigned	Not assigned	ned None
🕆 📥 r_qspi_smstr_rx		AN003	Not assigned	Not assigned	ned None
💣 r_qspi_smstr_rx		AN004	Not assigned	Not assigned	ned None
		AN006	Not assigned	Not assign	ned None
		AN007	Not assigned	Not assign	ned None
	<				2

Figure 6-8 Config_S12AD0 in Software Components View

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



(2) Pin initialization for drivers with r_xxx (Note2)

The pin setting source file will be generated in $\pm src \pm smc_gen \pm r_pincfg$ folder with the name $r_xxx_pinset.c.$

The API functions in this file are called by the user from application codes.

oftware Components 🛛 🗉 🗄	🕹 🚨 Pin Funct	ion			9 🔳 🖬 i	è
Type filter text	type filte	er text (* = any	y string, ? = any character)	All		
∽ 🏯 r_bsp	Enabled	Function	Assignment	Pin Number	Direction	
🔍 r_bsp	\checkmark	ET0_COL	PC7/UB/A23/CS0#/MTIOC3A/MTCLKB/TMO2/PO31/TOC0/CF	/ 76	1	
🖞 🚣 Single Scan Mode S12AD	\checkmark	ETO CRS	P83/EDACK1/MTIOC4C/ET0 CRS/RMII0 CRS DV/SCK10/SS10	/ 74	1	
💣 Config_S12AD0		ET0_ERXD0	P75/CS5#/PO20/ET0_ERXD0/RMII0_RXD0/SCK11/RTS11#/SC	/ 87	1	
🖞 🚣 r_ether_rx		ET0_ERXD1	P74/A20/CS4#/PO19/ET0_ERXD1/RMII0_RXD1/SS11#/CTS11	/ 88	1	
矿 r_ether_rx		ET0_ERXD2	PC1/A17/MTIOC3A/TCLKD/PO18/ET0_ERXD2/SCK5/SSLA2-A	/ 89	1	
🖞 🚣 r_qspi_smstr_rx		ETO_ERXD3	PC0/A16/MTIOC3C/TCLKC/PO17/ET0_ERXD3/CTS5#/RTS5#/S	/ 91	1	
💣 r_qspi_smstr_rx		ET0_ETXD0	P81/EDACK0/MTIOC3D/PO27/ET0_ETXD0/RMII0_TXD0/SMIS	/ 80	0	
	\checkmark	ET0_ETXD1	P82/EDREQ1/MTIOC4A/PO28/ET0_ETXD1/RMII0_TXD1/SMO	/ 79	0	
		ET0_ETXD2	PC5/D3/A21/CS2#/WAIT#/MTIOC3B/MTCLKD/TMRI2/PO29/E	/ 78	0	
		ET0_ETXD3	PC6/D2/A22/CS1#/MTIOC3C/MTCLKA/TMCI2/PO30/TIC0/ETC	/ 77	0	
	\checkmark	ETO_LINKST/	P34/MTIOC0A/TMCI3/PO12/POE10#/ET0_LINKSTA/SCK6/SCk	/ 27	1	
		ET0_MDC	P72/A19/CS2#/ET0_MDC/LCD_DATA23-A	/ 101	0	
		ET0 MDIO	P71/A18/CS1#/ET0 MDIO	/ 102	Ю	
		ETO_RX_CLK	P76/CS6#/PO22/ET0_RX_CLK/REF50CK0/SMISO11/SSCL11/R	/ 85	1	
	\checkmark	ETO_RX_DV	PC2/A18/MTIOC4B/TCLKA/PO21/ET0_RX_DV/RXD5/SMISO5/	/ 86	1	
		ETO_RX_ER	P77/CS7#/PO23/ET0_RX_ER/RMII0_RX_ER/SMOSI11/SSDA11	/ 84	1	
	\checkmark	ETO_TX_CLK	PC4/A20/CS3#/MTIOC3D/MTCLKC/TMCI1/PO25/POE0#/ET0_	/ 82	1	
		ETO_TX_EN	P80/EDREQ0/MTIOC3B/PO26/ET0_TX_EN/RMII0_TXD_EN/SCk	/ 81	0	
		ET0_EXOUT	Not assigned	Not assigned	None	
		ET0_TX_ER	Not assigned	Not assigned	None	
		ET0_WOL	/ Not assigned	Not assigned	None	
		REF50CK0	Not assigned	Not assigned	None	
	<				>	

Figure 6-9 *r_ether_rx* in Software Components View

Folder	File	Function	Driver	Description
r_pincfg	r_xxx_pinset.c (Note*2)	R_xxx_PinSet_xxxn (Note*2,3)	FIT	This API function initializes the pins used by this driver. Refer to the application note in the corresponding <i>r_xxx</i> folder before calling this API function: <i>¥src¥smc_gen¥r_xxx¥doc</i> (Note*2)

Note *2: xxx is the name of a peripheral function.

*3: n is a peripheral channel number.



(3) Pin initialization for drivers with r_xxx_smstr (Note4)

The pin setting header file will be generated in ¥src¥smc_gen¥r_config folder with the name *r_xxx_smstr_rx_pin_config.h.*

The macro definitions in this file will be handled in the *r_xxx_smstr* source files.

Type filter text type filter text (* = any string, ? = any character) All 	Pin configuration				🕲 Generate	Code	👜 Generate Repo
✓ ▲ r_bsp Function Assignment Pin Number Direction ✓ ▲ Single Scan Mode S12AD ✓ QlO0 / PD6/D6/MTICSV/MTIOC8A/POE4#/SSLC2-A/QMO-B/QIO0-B / 145 (B) IO ✓ ▲ Single Scan Mode S12AD ✓ QlO1 / PD7/D7/MTICSU/POE0#/SSLC3-A/QMI-B/QIO1-B/SDHI_D1-E / 143 (B) IO ✓ △ QlO2 / PD2/D2/MTIOC4D/TIC2/MISOC-A/CRX0/QIO2-B/SDHI_D2-B / 154 (B) IO ✓ △ QlO3 / PD3/D3/MTIOC8D/TOC2/POE8#/RSPCKC-A/QIO3-B/SDHI_D. / 150 (B) IO ✓ △ QSPCLK / PD5/D5/MTIC5W/MTIOC8C/POE10#/SSLC1-A/QSPCLK-B/SD / 147 (B) O	Software Components 🛛 🕀 🕀 🖧	Pin Functi	on			3	2 🖬 🖬 🔤 e
• r_bsp QIOO • PD6/D6/MTIC5V/MTIOC8A/POE4#/SSLC2-A/QMO-B/QIOO-B • 145 (B) IO • Single Scan Mode S12AD QIO1 • PD7/D7/MTIC5U/POE0#/SSLC3-A/QMI-B/QIO1-B/SDHI_D1-E • 143 (B) IO • Config_S12AD0 QIO2 • PD2/D2/MTIOC4D/TIC2/MISOC-A/CRX0/QIO2-B/SDHI_D2-B • 154 (B) IO • T_ether_rx QIO3 • PD3/D3/MTIOC8D/TOC2/POE8#/RSPCKC-A/QIO3-B/SDHI_D • 150 (B) IO • QSPCLK • PD5/D5/MTIC5W/MTIOC8C/POE10#/SSLC1-A/QSPCLK-B/SD • 147 (B) O	Type filter text	type filte	r text (* = a	ny string, ? = any character)	,	All	~
<pre> r_qspi_smstr_rx </pre>	 r_bsp [*] Single Scan Mode S12AD [*] Config_S12AD0 [*] r_ether_rx [*] r_ether_rx [*] r_qspi_smstr_rx [*] 		QIO0 QIO1 QIO2 QIO3	 PD6/D6/MTIC5V/MTIOC8A/POE4#/SSLC2-A/QMO-B/QIO0-B PD7/D7/MTIC5U/POE0#/SSLC3-A/QMI-B/QIO1-B/SDHI_D1-E PD2/D2/MTIOC4D/TIC2/MISOC-A/CRX0/QIO2-B/SDHI_D2-B PD3/D3/MTIOC8D/TOC2/POE8#/RSPCKC-A/QIO3-B/SDHI_D. 	 145 (B) 143 (B) 154 (B) 150 (B) 	per	IO IO IO

Figure 6-10 r_qspi_smstr_rx in Software Components View

Folder	File	Function	Driver	Description
r_config	r_xxx_smstr_rx_pin_config.h (Note*4)	-	FIT	Macro definitions in this header file initialize the pins used by this driver. These macros will be called in <i>r_xxx_smstr</i> source files.

Note *4: xxx is the name of a peripheral function.

(4) Reference to pin initialization codes

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

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



6.6 Initializing Interrupts

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

Refer to the application note in the corresponding ¥src¥smc_gen¥r_xxx¥doc folder to initialize interrupts used in *r_xxx* modules (xxx is the name of peripheral function).

	Vector Number V 111	Interrupt GROUPBL1	Peripheral (1)	Priority Level 15	Status Used	Multiple Interrupts	Fast Interrupt
	20	S12CMPI	S12AD	_	Used		
(2)	> 113	GROUPAL1		Level 2	Used		-
(3)	190	INTB190 (S12ADI)	S12AD (2)	Level 15	Used		(4)

Figure 6-11 Interrupts Configuration in Interrupts View

No	Item	Folder	File	Driver	Description
(1)	Priority	general	r_smc_interrupt.c	CG	This interrupt priority level setting is for group interrupts (Note5) It is initialized in <i>R_Interrupt_Create</i> of this file. <i>R_Systeminit</i> in
					<pre>r_cg_hardware_setup.c will call this function before entering main() function.</pre>
(2)	Priority	{ConfigName}	{ConfigName}.c	CG	This interrupt priority level setting is for normal interrupts and software configurable interrupts A and B ^(Note5) . 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.
(3)	Vector Number	r_config	r_bsp_interrupt_config.h	CG FIT	Vector number of software configurable interrupts A and B ($Note5$) in the [Interrupts] tabbed page will be mapped in this file and handled by r_bsp .
(4)	Fast Interrupt	general	r_smc_interrupt.c	CG	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	CG	Vector number of fast interrupt will be defined in this file. { <i>ConfigName}_user.c</i> will use this macro definition to prepare a fast interrupt service routine.
(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.

Note *5: The type of interrupt depends on hardware specifications.



6.7 Component Settings

6.7.1 FIT module configuration

1) Configuration for *r_bsp*

Configuration file of r_bsp is generated as $r_bsp_config.h$ under the $\pm src\pm smc_gen\pm r_config$ folder. It contains clock-initialization and other MCU-related settings (for e.g. the package).

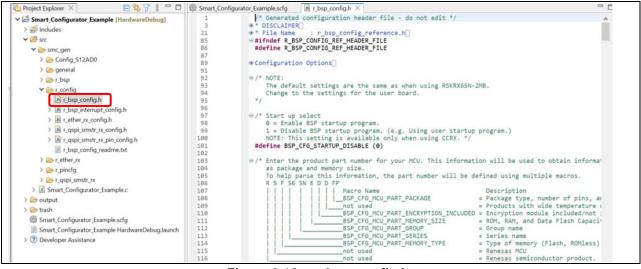


Figure 6-12 r_bsp_config.h

2) Configuration of FIT modules

Configuration files of FIT modules that are added to the project are generated as *r_xxx_config.h* under ¥src¥smc_gen¥r_config folder. (*r_xxx* is the name of FIT module)

For FIT modules that have configuration GUI at [Component] page, the configuration will be generated by Smart Configurator. Therefore, you do not need to change the *.h file manually.

For FIT modules that do not have configuration GUI at [Component] page, you need to modify these configurations at *.h file manually. As shown in the figure below, read (1) *Explanation information* before setting the macro definition value in (2) *Configuration*.

Refer to the application note in ¥src¥smc_gen¥r_xxx¥doc folder on how to modify r_xxx_config.h.



r_ptp_rx_cor	nfig.h ⊠	- 8
215		~
216	%/* grandmasterPriority2: Equal to parentDS.grandmasterPriority2 field */	
217	/* From 0 to 255 can be set and lower value has higher priority */	
218	<pre>#define PTP_CFG_GM_PRIORITY20 (0x00) /* (Port0) */</pre>	
219	#define PTP_CFG_GM_PRIORITY21 (0x00) /* (Port1) */ (1) Explanation information	
220		
221	<pre>/* grandmasterClockQuality: Equal to parentDS.grandmasterClockQuality field */</pre>	
222	/* b31 to b24: clockClass, default value(=248, = 0xF8), 255 is slave only clock */	
223	/* b23 to b16: clockAccuracy, default value(=0x21)is within 100 nsec, 0x20 to 0x31 */	
224	/* b15 to b0: offsetScaledLogVariance, default value(=0xFFFF) is not calculated yet */	
225	<pre>#define PTP_CFG_GM_CLK_QUALITY0 (0xF821FFFF) /* (Port0) */</pre>	
226	<pre>#define PTP_CFG_GM_CLK_QUALITY1 (0xF821FFFF) /* (Port1) */</pre>	
227	(2) Configuration	
228	<pre>/* grandmasterIdentity: Equal to parentDS.grandmasterIdentity field */</pre>	
229	<pre>#define PTP_CFG_GM_CLK_ID0_U (0x000000000) /* Clock-ID hi (Port0) */</pre>	
230	<pre>#define PTP_CFG_GM_CLK_ID0_L (0x000000000) /* Clock-ID lo (Port1) */</pre>	
231	<pre>#define PTP_CFG_GM_CLK_ID1_U (0x00000000) /* Clock-ID hi (Port0) */</pre>	
232	<pre>#define PTP_CFG_GM_CLK_ID1_L (0x00000000) /* Clock-ID lo (Port1) */</pre>	
233		
234	<pre>/* currentUtcOffset: Equal to timePropertiesDS.currentUtcOffset field */</pre>	
235	<pre>#define PTP_CFG_CUR_UTC_OFFSET0 CURRENT_UTC_OFFSET /* (Port0) */</pre>	
236	<pre>#define PTP_CFG_CUR_UTC_OFFSET1 CURRENT_UTC_OFFSET /* (Port1) */</pre>	
237		
238	<pre>/* timeSource: Equal to timePropertiesDS.timeSource field */</pre>	
239	<pre>#define PTP_CFG_TIME_SOURCE0 (0xA0) /* Timesource is internal oscillator (Port0) */</pre>	
240	<pre>#define PTP_CFG_TIME_SOURCE1 (0xA0) /* Timesource is internal oscillator (Port1) */</pre>	
241		~
	<	>

Figure 6-13 Example of *r_xxx_config.h* (*r_ptp_rx_config.h*)



6.7.2 FreeRTOS Kernel configuration

Configuration file of Renesas *FreeRTOS_Kernel* is generated as *FreeRTOS_Kernel.h* under the ¥src¥frtos_config.

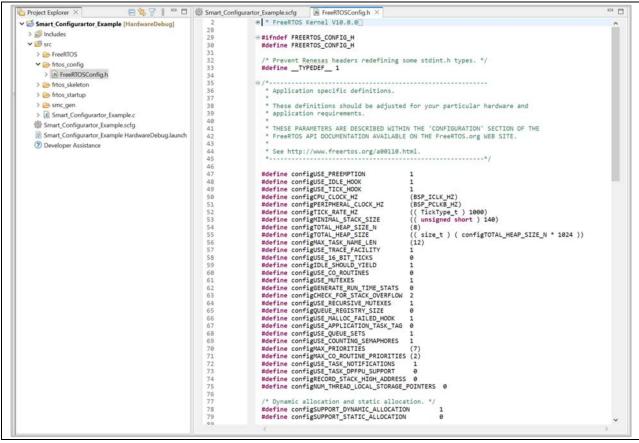


Figure 6-14 FreeRTOSConfig.h



7. Creating User Programs

The Smart Configurator handles two component types, [Firmware Integration Technology] and [Code Generator], with each requiring different methods to add custom code to the output source files. This section describes the methods to add custom code for both components.

7.1 Adding Custom Code in the Case of Firmware Integration Technology (FIT)

When [Firmware Integration Technology] 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.

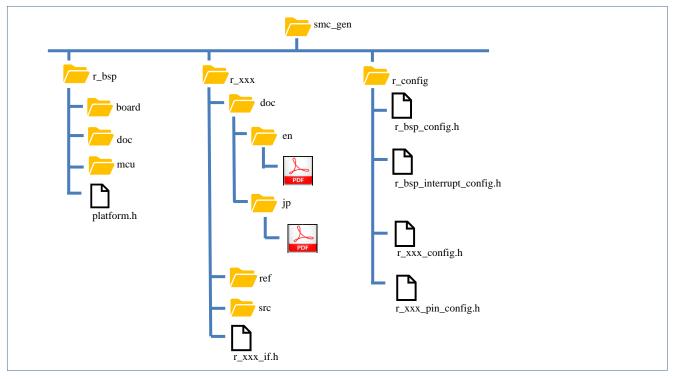


Figure 7-1 Tree Structure of Directories and Files for a FIT Module



7.2 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
 #include "r_cg_macrodriver.h"
#include "r_cg_userdefine.h"
#include "Config_CMT3.h"
/* Start user code for include. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */
Global variables and functions
  /* Start user code for global. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */
* Function Name: R_Config_CMT3_Create_UserInit
* Description : This function adds user code after initializing the CMT3 channel
* Arguments
        : None
* Return Value : None
            *****
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 */
}
```



RX Smart Configurator

```
******
* 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 */
```



7.3 Using Generated Code in user application

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

1) Open the {*Project name*}.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 added for you in "r_smc_entry.h" by automatically.

Project Explorer X 🗧 🗞 🍟 📱 🗆 📧 Smart_Configurator_Example.c X	
<pre>v Us Shart Configurator_Example [HardwareDebug] > @ Indudes v @ src v @ snc.gen > @ config.S12AD0 > @ general > @ rether rx > @ config.</pre>	

Figure 7-2 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 ($R_ConfigName_Stop$).

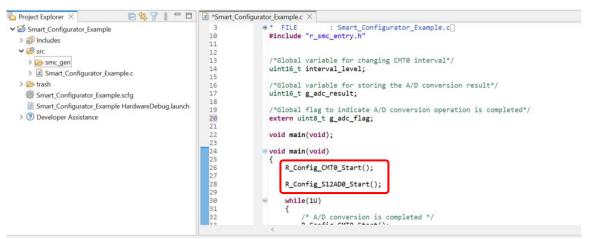


Figure 7-3 Call Code Generator functions

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.

For more reference, refer to "Smart Configurator Application Examples" in "chapter 12 Documents for Reference".



8. Backing up Generated Source Code

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

The Smart Configurator generates a backup folder for the previously generated source code when new code is generated by clicking on the [¹] (Generate Code)] button. <Date-and-Time> indicates the date and time when the backup folder is created after code generation.

<ProjectDir>¥trash¥<Date-and-Time>



9. Generating Reports

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

9.1 Report on All Configurations

A report is output in response to clicking on the [🚔 (Generate Report)] button in the Smart Configurator view. Two selections of output files are available (PDF, Text).

Smart_Configurator_Example.scfg ×			⊂ a
Overview information			Generate Code Generate Report
			0 ^
Overview Get an Exercise of the features provided by S Videos Introduction to Smart Configurator Browne related videos What's New Check out sphart's nore in the latest release. See all Release Notes. Product Documentation User's Guide All manual Application Notes Product Documentation	mart Configur	Application Code Software Components Middleware & Drivers Device Drivers MCU Hardware	
Tool news			
✓ Current Configuration			
Selected board/device: R5F564MLCxFC (ROM size: 4Mbytes,	RAM size: 512		
Generated location (PROJECT_LOC\): New_Folder		Edit	
Selected components:			
Component	Version	Configuration	
Board Support Packages. (r_bsp)	7.42	r_bsp(used)	
Ethernet Driver. (r_ether_rx)	1.23	r_ether_rx(used)	
QSPI Clock Synchronous Single Master Control Modul		r_qspi_smstr_rx(used)	
Single Scan Mode S12AD	2.5.0	Config_S12AD0(S12AD0: used)	
Overview Board Clocks System Components Pins Interruc	de .		•



e ² Smart Report	×
Generate report of configurations	
Options Print all sections Print specific sections	
Board Clocks	^
Components	
Pins	> V
Output as PDF	Select Font
Output as text	Preview
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 the [🔜 (Save the list to .csv file)] button on the [Pins] page of the Smart Configurator view.

ardware Resource 🛛 🕀 🖻 🗎	a <mark>z 🟯</mark>	Pin Functi	on			ଏ 🖬 🖪 ଧ	<u>×9</u> g
Type filter text		type filter	r text (* = any	r string, ? = any character)	All		~
📥 All	^	Enabled	Function	Assignment	Pin Number	Direction	1
😻 Clock generator		\checkmark	ADTRG0#	P07/IRQ15/ADTRG0#	/ 176	1	
Clock frequency accuracy measur	e	\checkmark	AN000	/ P40/IRQ8/AN000	/ 173	1	
📲 Buses		\checkmark	AN005	/ P45/IRQ13/AN005	/ 167	1	
EXDMA controller		\checkmark	ET0_COL	PC7/UB/A23/CS0#/MTIOC3A/MTCLKB/TMO2/PO31/TOC0/CF	/ 76	L	
Interrupt controller unit		\checkmark	ET0_CRS	P83/EDACK1/MTIOC4C/ET0_CRS/RMII0_CRS_DV/SCK10/SS10	/ 74	1	
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	I.	
MTU1		\checkmark	ET0_ERXD2	PC1/A17/MTIOC3A/TCLKD/PO18/ET0_ERXD2/SCK5/SSLA2-A	ø 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	I	
MTU8	\sim	\checkmark	ET0_MDC	P72/A19/CS2#/ET0_MDC/LCD_DATA23-A	/ 101	0	
< >		<				>	



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 [IIII] (Save Package View to external image file)] button of the [MCU/MPU Package] view.

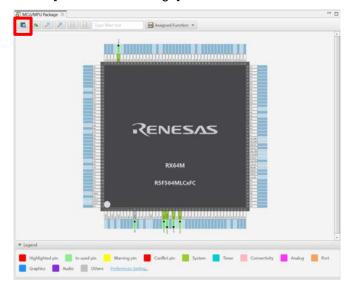


Figure 9-4 Outputting a Figure of MCU/MPU Package (in png Format)

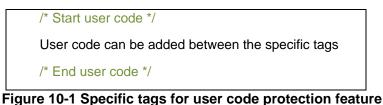
10. User code protection feature for Smart Configurator Code Generation component

The Smart Configurator for RX Plug-in in e2 studio 2023-01 and 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.

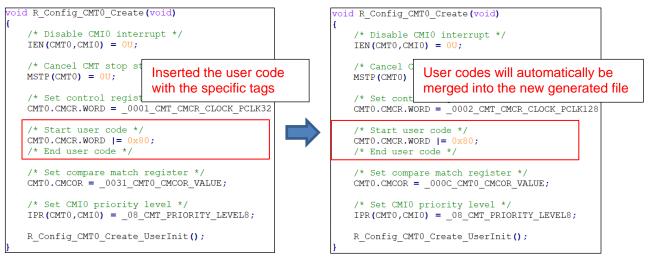


Figure 10-2 User code protection with auto merge



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.

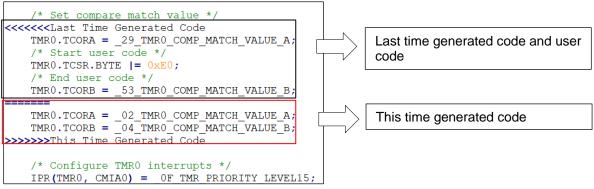


Figure 10-3 User code protection with merge conflict

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

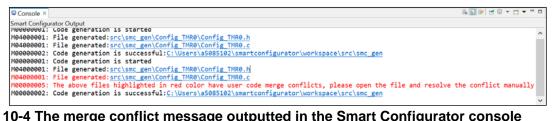


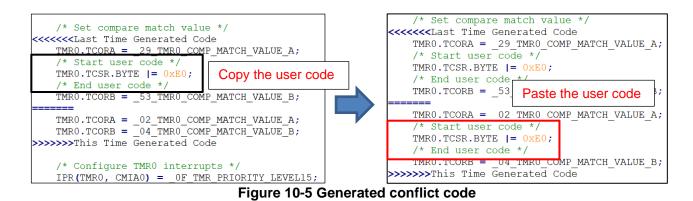
Figure 10-4 The merge conflict message outputted in the Smart Configurator console



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.

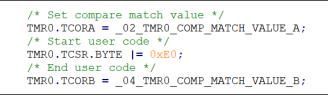


Figure 10-6 The codes after resolving the merge conflict

Another way to solve merge conflict:

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

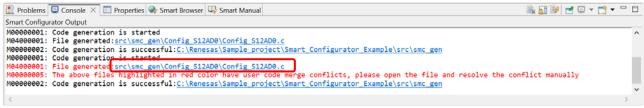


Figure 10-7 Error message in console



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

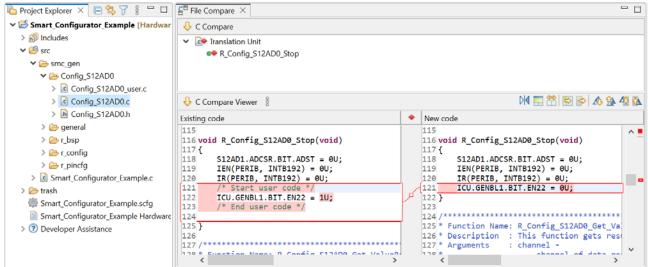


Figure 10-8 Compare Viewer for conflict code



11. Help

11.1 Help

Refer to the help system from the e² studio menu for detailed information on the Smart Configurator.



Figure 11-1 Help Menu

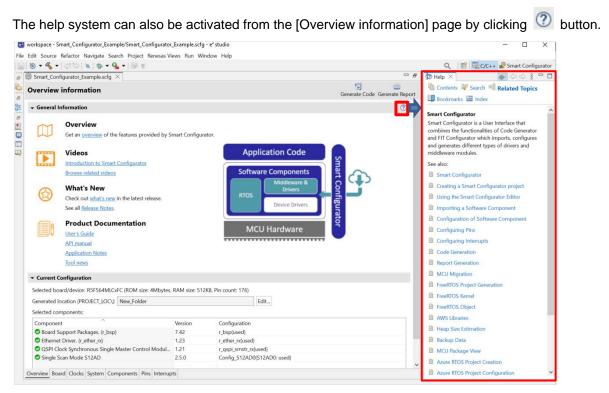


Figure 11-2 Smart Configurator Quick Start information

11.2 Developer assistance

Developer assistance provides user a tool to display API information and calling examples of components they added so user can easily develop without having to know the auto-generated code.

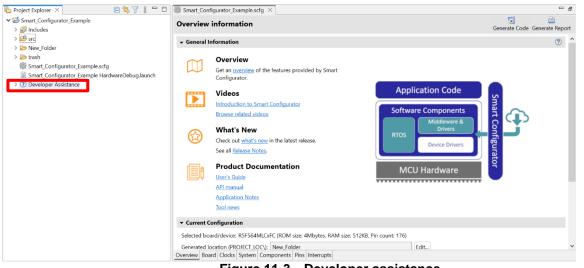


Figure 11-3 Developer assistance



- (1) Add component and generate code.
- (2) Check generated file exist.
- (3) Expand [Developer assistance] in project explore and double click the component.
- (4) Read API information and examples.

Project Explorer X	🔯 Smart_Configurator_Example.scfg >	<	
 ✓ Smart_Configurator_Example >	Software component config	guration	(1).b Generate Code Generate Report
> 😂 src > 🗁 trash	Components	èn 🖆 🗦 🗉 🏵 🆆 🕶	Configure
 Smart_Configurator_Example.scfg Smart_Configurator_Example HardwareDebug.launch ⑦ Developer Assistance Config_S12AD0 	type filter text	(1).a	Basic setting Note When using the 12-bit A/D converter (unit 1), we recommend not usin Analog input mode setting Double trigger mode Extend analog input mode Analog input channel setting AN100 AN101 AN102 AN103 AN105 AN106 AN107 AN103 AN105 AN106 AN107 AN103 AN110 AN111 AN112 AN113 AN115 AN116 AN117 AN118 AN120 Temperature sensor output Interne Conversion start trigger setting Start trigger source Software trigger Interrupt setting *
``````````````````````````````````````	contentient board clocks bystern com	ponento interropto	



✓ Smart_Configurator_Example > Sin Includes ✓ Sinc gen (2)	Single Scan Mode S12AD The Code Generator outputs the following API function	ns for the single scan mode S12AD. (4)
> 😓 Config_S12AD0 > 🗁 general	API Function Name	Function
> 😂 r_bsp > 😂 r_config > 😂 r_pincfg	R <config s12ad0=""> Create</config>	Executes initialization processing that is required before controlling the single scan mode S12AD.
> C Smart_Configurator_Example.c	R <config_s12ad0>_Start</config_s12ad0>	Starts A/D conversion.
> 🗁 trash	R_ <config_s12ad0>_Stop</config_s12ad0>	Stops A/D conversion.
Smart Configurator Example HardwareDebug.launch	R <config s12ad0=""> Get ValueResult</config>	Gets the result of conversion.
> Config_S12AD0	R <config s12ad0=""> Set CompareValue</config>	Sets the compare level.
(2)	R_ <config_s12ad0>_Set_CompareAValue</config_s12ad0>	Sets the compare level for window A.
(3)	R <config s12ad0=""> Set CompareBValue</config>	Sets the compare level for window B.
	R <config s12ad0=""> Create UserInit</config>	Executes user-specific initialization processing for the single scan mode S12AD.
	r_ <config_s12ad0>_interrupt</config_s12ad0>	Executes processing in response to scan end interrupts.
	r <config s12ad0=""> compare interrupt</config>	Executes processing in response to compare interrupts.

Figure 11-5 Developer assistance in Project Explorer



#### 12. Documents for Reference

#### User's Manual: Hardware

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

#### Technical Update/Technical News

Obtain the latest information from the Renesas Electronics website.

#### User's Manual: Development Environment

e2 studio Integrated Development Environment User's Manual: Getting Started Guide (R20UT4374) CC-RX Compiler User's Manual (R20UT3248) RX family Renesas FreeRTOS Application Note (R01AN4307) e2 studio Partner RTOS Aware Debugging for RX (R20AN0586) (Obtain the latest version from the Renesas Electronics website.)

#### Smart Configurator Application Examples

Smart Configurator Application Examples: Ethernet (R20AN0495) Smart Configurator Application Examples: CMT, A/D, SCI, DMA, USB (R20AN0469)



# 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**

Revision History			Description
Rev.	Date	Page	Summary
1.00	Jul 31, 2017	-	First edition issued
1.10	Mar 20, 2018	-	Update to e ² studio v6.2
		4	1.2 Operating Environment updated
		8	Figure 2-4 Select toolchain, device and debug setting updated
		15	Figure 3-6 Add component updated
		16	Figure 3-7 Add Code Generator component updated
		19	Figure 3-12 Add FIT modules updated
		26	Figure 3-22 Configure pins for r_ether_rx updated
		30	Note added
		36	4.1 Generated files structure
			Description of the r_bsp folder in the table corrected
		50	8.1 Troubleshooting added
		53	9 Application Example updated
1.20	Nov 1, 2018	-	Update to e ² studio v7.0
	,	3	1. Overview updated
		9	Figure 3-1 Procedure for Operations added
		10	3.3 File to be Saved as Project Information added
		11	3.4 Window added
		19	4.1.3 Exporting board setting, 4.1.4 Importing board setting
			added
		21	Figure 4-7 [Components] Page added
		33	Figure 4-26 [Pins] Page ([Pin Function]), Figure4-26 [Pins]
			Page ([Pin Number]) added
		39	Figure4-35 [Interrupts] Page
		42	Figure 5-1 Generating a Source File, Figure5-2 Source Files i
			the Project Explorer added
		55	7. Creating User Programs added
		60	9.2 Configuration of Pin Function List and Pin Number List (in csv Format), 9.3 Image of MCU Package (in png Format)
			added
		61	10.1 Help added
1.30	Jan 1, 2019	-	Update to e ² studio v7.3
		9	2.2 Create a FreeRTOS project added
		21	3.4.3 MCU Package view updated
		23	4.1.1 Selecting the device updated
		31	4.3.4 Download and import FIT sample project added
		40	4.3.9 Setting a FIT Software Component update
		41	4.3.12 Version change of FIT software component added
		43	4.3.11 Setting the FreeRTOS Kernel added
		44	4.3.12 Configure general setting of component added
		45	Figure 4-41 [Pins] Page ([Pin Number]) updated
		49	4.4.5 Pin setting using board pin configuration information
			added
			4.4.6 Pin filter feature added
		53	4.6 MCU migration feature added
		70	6.6.2 FreeRTOS Kernel configuration added
1.40	Jun 21, 2021	-	Update to e ² studio v2021-04
		5	2. Creating a project updated
		19	3.4.2 Smart Configurator view updated
		22	4.1 Board Settings updated

		25	4.3 System Settings added
		34	4.4.4 Downloading a FIT module updated
		35	4.4.6 Adding FIT drivers or middleware updated
		39	4.4.9 Setting a FIT Software Component updated
		40	4.4.10 Version change of FIT software component updated
		42	4.4.11 Solving the grey-out component added
		43	4.4.13 Setting the RTOS Kernel added
		44	4.4.14 Creating the RTOS Object added
		45	4.4.15 Setting the RTOS Library added
		46	4.4.16 Configure Analog Front End component added
		48	4.4.17 Configure Motor Component added
		57	4.4.18 Configure general setting of component
		58	4.5.8 Pin Errors/Warnings setting added
		65	5 Managing Conflicts updated
		68	6.2 Change Generated Code Location added
		84	7.3 Using Generated Code in user application added
		89	11 Documents for Reference updated
1.50	Apr 16, 2023	1	Updated the URL for RX Smart Configurator
		95 - 97	Added new chapter 10 User code protection feature for Smart
			Configurator Code Generation Component
1.60	Apr 16, 2024	-	Figures are updated to e ² studio 2024-01
		16	2.3 Create a Blinky project added
		26	3.4.3 MCU Package view updated
		34	4.4.2 Removing a software component updated
		55	4.4.16 Configure Motor Component updated
		59	4.4.17 Configure general setting of component updated
		68	4.5.9 Symbolic name setting added
		73	4.6.3 Multiple interrupts setting added
		106	10.3.2 Steps for resolving the merge conflict updated
		109	11.2 Developer assistance added

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

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

1. 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 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.
- 2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
- 3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
- 4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
- 5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
- 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, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
- 10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
- 12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
- This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
   Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas
- Electronics products. (Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled
- subsidiaries.
- (Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

# **Corporate Headquarters**

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan www.renesas.com

# **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: <u>www.renesas.com/contact/</u>.

### Trademarks

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