Smart Configurator

Guide on Sample Projects for RH850/U2A Devices

Introduction

This document describes how to use the sample project of Smart Configurator for RH850/U2A devices in CS+ integrated development environment.

Target Device

RH850/U2A16 (BGA516, BGA292)

RH850/U2A8 (BGA292)

RH850/U2A6 (BGA292, BGA156, LQFP176, LQFP144)

Contents

1. Overview

1.1 Purpose

This document describes how to use the sample project of Smart Configurator for RH850/U2A devices in CS+ integrated development environment.

When applying this application note to a microcontroller, change the contents according to the specifications of the microcontroller you are using and validate the correct operation of the sample projects.

1.2 Operating Environment

Install the Smart Configurator and tools to be used to create or build programs in CS+ based on the source files generated by the Smart Configurator with the use of the sample projects.

For details on how to use your integrated development environment, refer to the user's manual for the integrated development environment that you are using.

Table 1-1 Operating Environment

Type	Name	Abbreviation in This Manual
IDE	CS+ for CC V8.09.00 or later	CS+
Toolchain	C Compiler Package for RH850 Family	CCRH

2. Outline of the Sample Projects

The Smart Configurator for RH850/U2A devices outputs a main function and source files that initialize peripheral modules that are set by components of the Smart Configurator. But it does not output the initialization codes to be performed before the execution of the main function and the startup routine which starts the main function and handles some other processing after the microcontroller has been reset.

This sample project includes boot program and startup code as a reference so that the user application codes and the peripheral modules codes generated according to the configurations in the Smart Configurator can be built and downloaded to the debug tool immediately.

2.1 Introduction of Sample Project

RH850/U2A sample project is a multi-core project which includes a main project “u2a16_startup.mtpj” and four subprojects “PE0.mtsp”, “PE1.mtsp”, “PE2.mtsp” and “PE3.mtsp”. In this sample project, only subproject “PE0.mtsp” includes the sample code of Interval Timer generated by Smart Configurator.

Each sub project has its own folder PE*n*(*n*=0-3) and independent interrupt vector table files sc_vecttble*n*.asm(*n*=0-3) and start files sc_cstart*n*.asm(*n*=0-3).

For details on the sample projects, see the descriptions in the relevant sections.

2.2 Notes on the Sample Projects

1. When using this sample project, please copy and use it in a directory that does not restrict access.

   If you use it in a directory that restricts access, the generated codes and other files will not be saved and some errors may occur.

   In general, the following directories require administrator permission:

   - Program files folder (e.g. "C:\Program Files", "C:\Program Files (x86)")

   - System root folder (e.g. "C:\Windows")

   For your environment, please contact your system administrator(IT department).

2. The Smart Configurator outputs the register descriptors according to iodefine.h for the Renesas CCRH compiler. iodefine.h for the Renesas CCRH compiler has been included in sample project. This file is used for building files generated by the Smart Configurator.

3. The Smart Configurator uses interrupts with the table lookup method as the method for selecting the interrupt handler addresses. The address where the table starts is set as 0x00040000 in the sample projects.

4. This sample project assumes that only PE0 is used for code generation. After generating code, please be sure to rename “main()” in folder “<ProjectDir>\src\smc_gen \general\ r_cg_main.c” to main_pm0().

5. The definition of the interrupt vector table of peripheral modules that was set in the Smart Configurator is reflected in smc/general/r_cg_intvector_PE0.c, which is output by the Smart Configurator. The file sc_intprg-A16A8.c in the “intprg” folder defines the vector table of EI maskable interrupt sources, which is the default EI vector table, is not set by the Smart Configurator.

6. Settings of files and sections in the sample projects are examples. You should change or create the settings newly to match the specifications of the microcontroller used in your system.

3. Basic Procedures for Operating the Smart Configurator

This section describes the basic operating procedures when building a user application with the files output from the Smart Configurator for RH850/U2A devices.

The basic operating procedure in CS+ environment without using the sample projects is described here. For the operating procedure with using a sample project, see the relevant section 4 “Description of the Sample Project”.

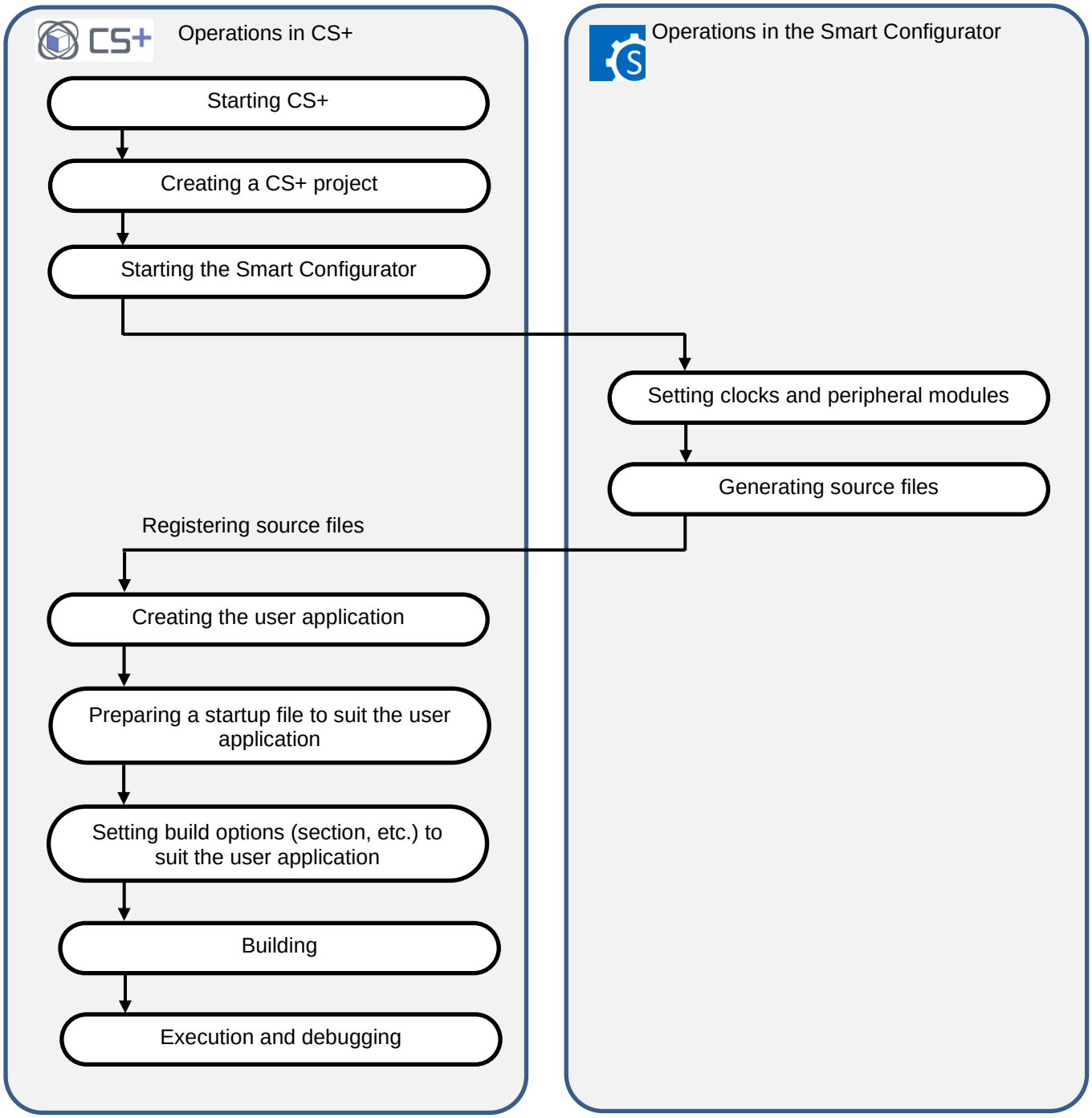

Figure 3-1. Basic Operating Procedure without using Sample Project

4. Description of the Sample Project

4.1 Configuration of the Sample Project

Figure 4-1 and Table 4-1 describes common folders and what files are included in the main project:

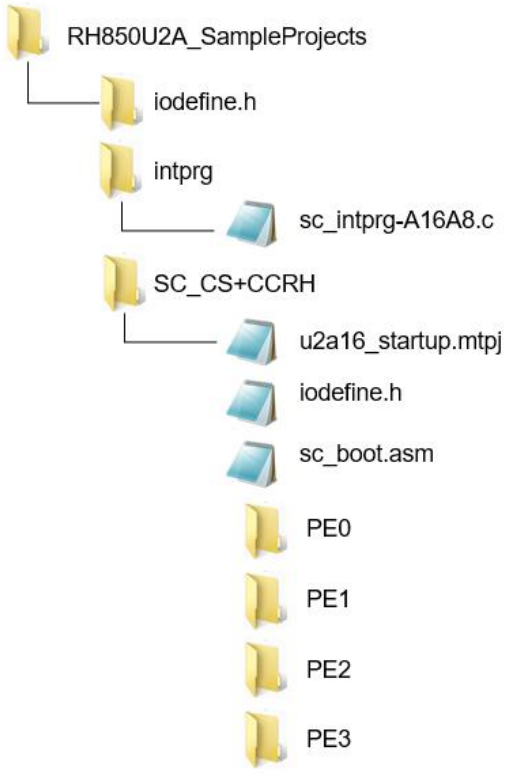

Figure 4-1. Common folder and Main project folder

Table 4-1. File Configuration of the main Project (corresponding to Figure 4-1)

File Name	Outline of File
iodefine.h folder	Folder which includes iodefine.h file
intprg folder	Folder which includes sc_intprg-A16A8.c file. This file defines default interrupt functions that are used by the EI maskable interrupt vector table for RH850/U2A devices and should be included into SC project
u2a16_startup.mtpj	Main project file for CS+
iodefine.h	Renesas CCRH header file that defines the registers for RH850/U2A devices
sc_boot.asm	Definition of root start routine and performs HW initialization for each PE, then branch to the corresponding PE*n*(*n*=0 ~3) start routine in sc_cstart*n*.asm(*n*=0~3)
PE*n* folder(*n*=0~3)	Folder of subproject which is corresponding to the specific PE*n*(*n*=0~3). The folder includes independent sc_start*n*.asm(*n*=0~3), main.c and the files output from the Smart Configurator for RH850/U2A devices.

Figure 4-2 and Table 4-2 describes what files are included in the subproject:

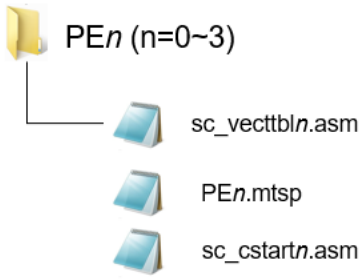

Figure 4-2. Subproject folder PE*n*(*n*=0~3)

Table 4-2. File Configuration of the subproject (corresponding to Figure 4-2)

File Name	Outline of File
sc_vecttbl*n*.asm(*n*=0~3)	Definition of PE*n*(*n*=0~3) reset entry to branch to the root start routine which is defined in sc_boot.asm file and definition of the interrupt vector table for PE*n*(*n*=0~3).
PE*n*.mtsp(*n*=0~3)	Subproject file for CS+
sc_cstart*n*.asm(*n*=0~3)	Definition of each PE*n*(*n*=0~3) startup routine which is called by sc_boot.asm and is executed until branching to the main_pm0() function.

NOTE: The Smart Configurator does not output the above files except for “PE*n*.scfg(*n*=0~3)” and folder “src”.

4.2 Basic Operating Procedure

Figure 4-3 shows the operating procedure in CS+ environment when using the sample project.

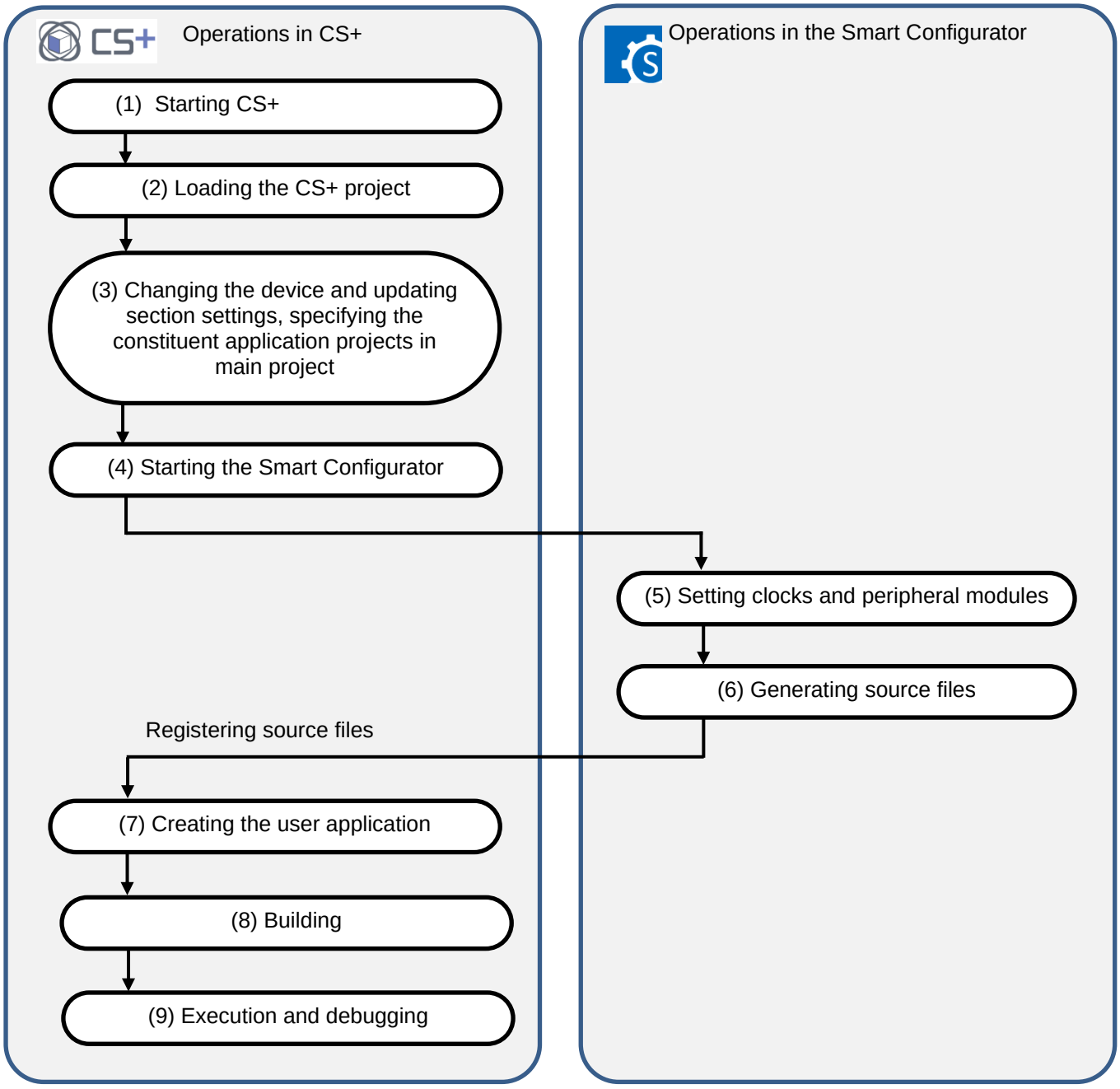

Figure 4-3. Operating Procedure for Sample Project

(1) Starting CS+

In the [Start] menu of Windows, select [Renesas Electronics CS+] → [CS+ for CC (RL78, RX, RH850)].

(2) Loading the CS+ sample project

From the [Open...] item of the [File] menu or [Open Existing Project] of CS+, select "u2a16_startup.mtpj".

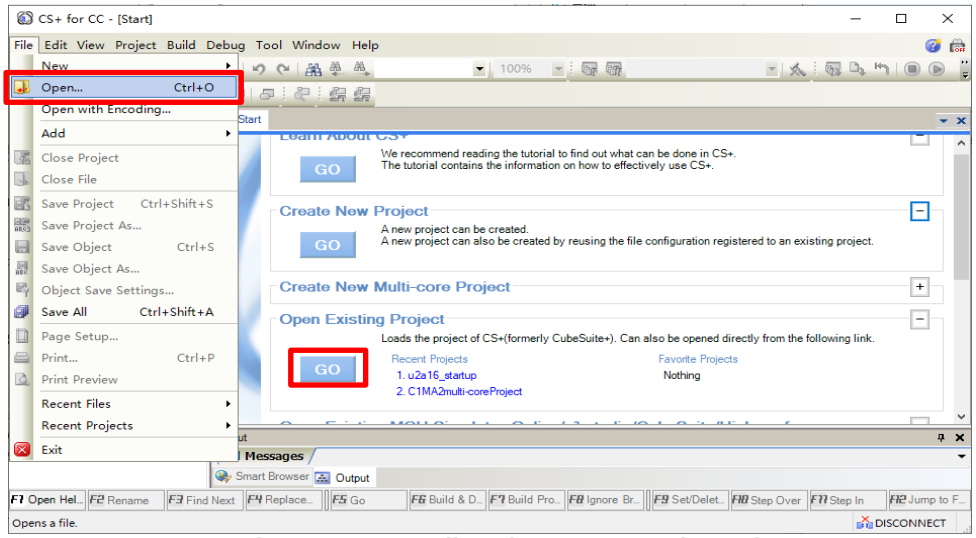

Figure 4-4. Loading the CS+ Sample Project

NOTE: Sample project has to be copied to a directory that does not restrict access before using. Please refer 2.2-1 for the detail.

(3) Changing the device

The R7F702300A (RH850/U2A) is selected as the target device in the sample project. If you are using another device, change the target device and file to be used with reference to section 4.3, "Procedure for Changing the Device". If the device does not require a change, proceed to step (4).

(4) Starting the Smart Configurator

(a) Confirm the setting of the path for the Smart Configurator for RH850. In the Project Tree panel, select [Smart Configurator (Design Tool)] and open the [Property] panel. Confirm that the path in which the Smart Configurator for RH850 was installed is set in [Smart Configurator for RH850 executable file path].

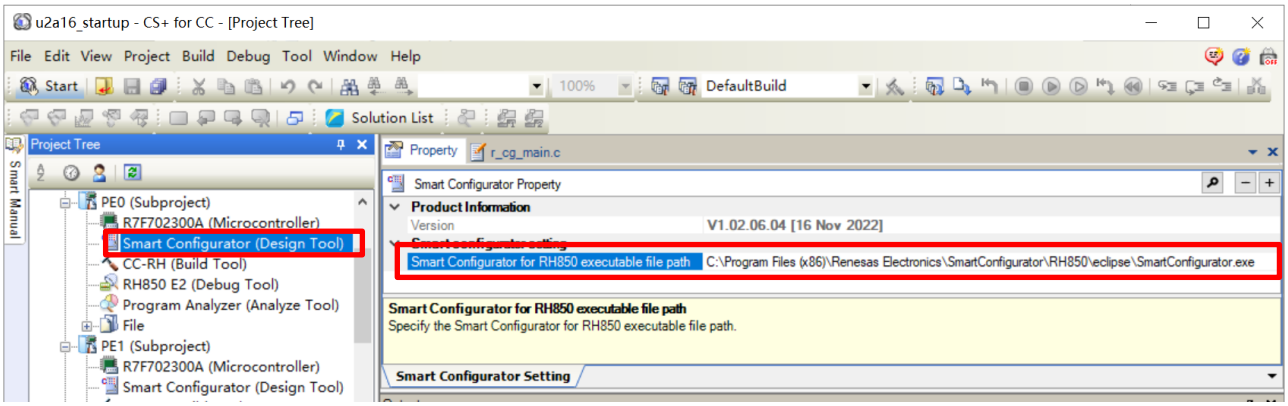

Figure 4-5. Setting the path for Smart Configurator

(b) Start the Smart Configurator for RH850 by double-clicking on [Smart Configurator (Design Tool)] in the Project Tree panel.

(c) If [RH850/U2Axx Package Selection] pops up, please choose the package user wants to use: Such as choose R7F702300(BGA516pin) as following figure shows:

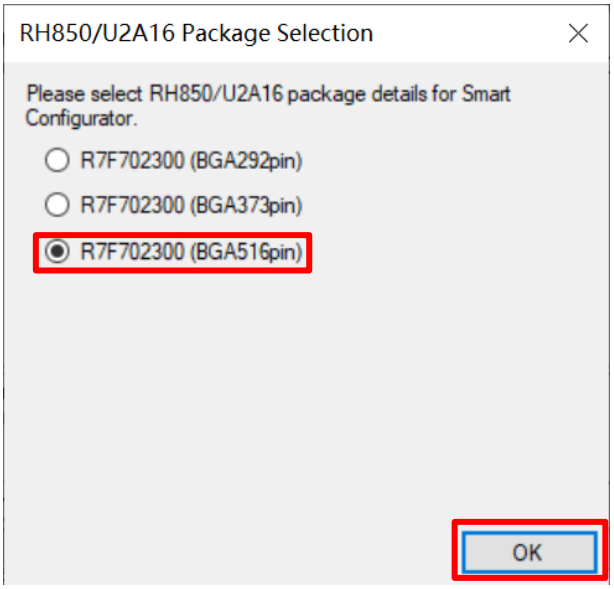

Figure 4-6. RH850/U2A16 Package Selection

(5) Smart Configurator setting - Clocks

(6) Smart Configurator setting - Components

(7) Smart Configurator setting - Generating driver

Above (5) ~ (7) are the procedures for setting clocks and components and generating driver in the Smart Configurator, please see section 5 "Operations in the Smart Configurator".

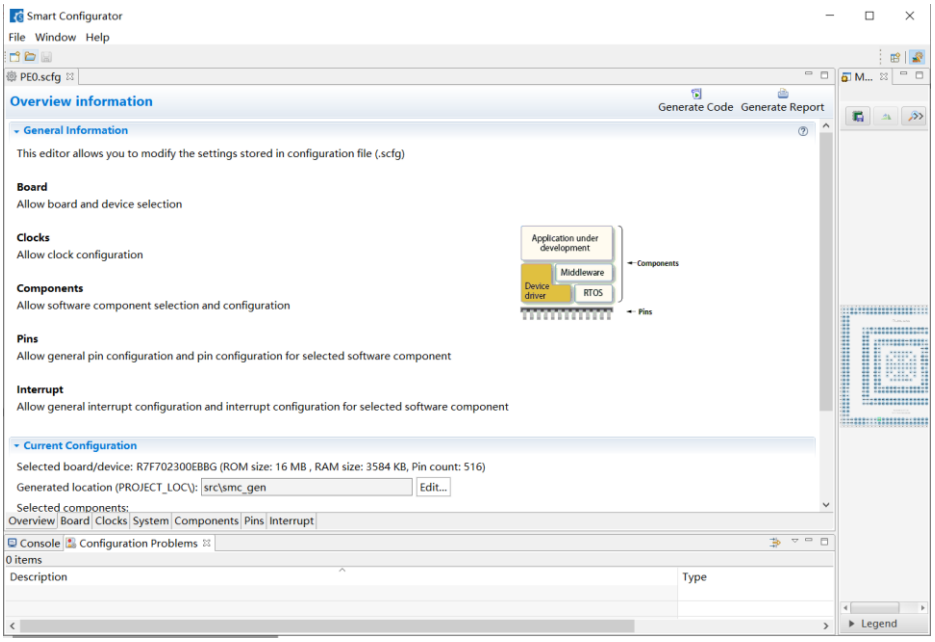

Figure 4-7. Smart Configurator setting

(8) Building

Build the driver and application code. Select [Build Project] from the [Build] menu or click on the [Builds the project. (F7)] button in the toolbar of CS+.

(9) Execution and debugging

For program execution and debugging in the emulator, refer to CS+ Integrated Development Environment User's Manual: RH850 Debug Tool (Obtained the latest information from the website of Renesas Electronics).

4.3 Procedure for Changing the Device

When the target device of the sample project differs from the device that is to be used, the target device or file to be used must be changed according to the following procedure.

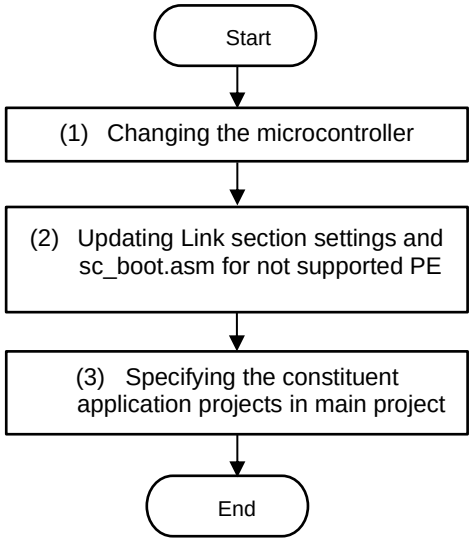

Figure 4-8. Changing the Target Device of the Sample Project

(1) Changing the microcontroller

(a) Select "R7F702300A (Microcontroller)" of main project “u2a16_startup.mtpj” and then select [Change Microcontroller...] from the context menu.
Click on the [OK] button in the [Question] dialog box that appears.

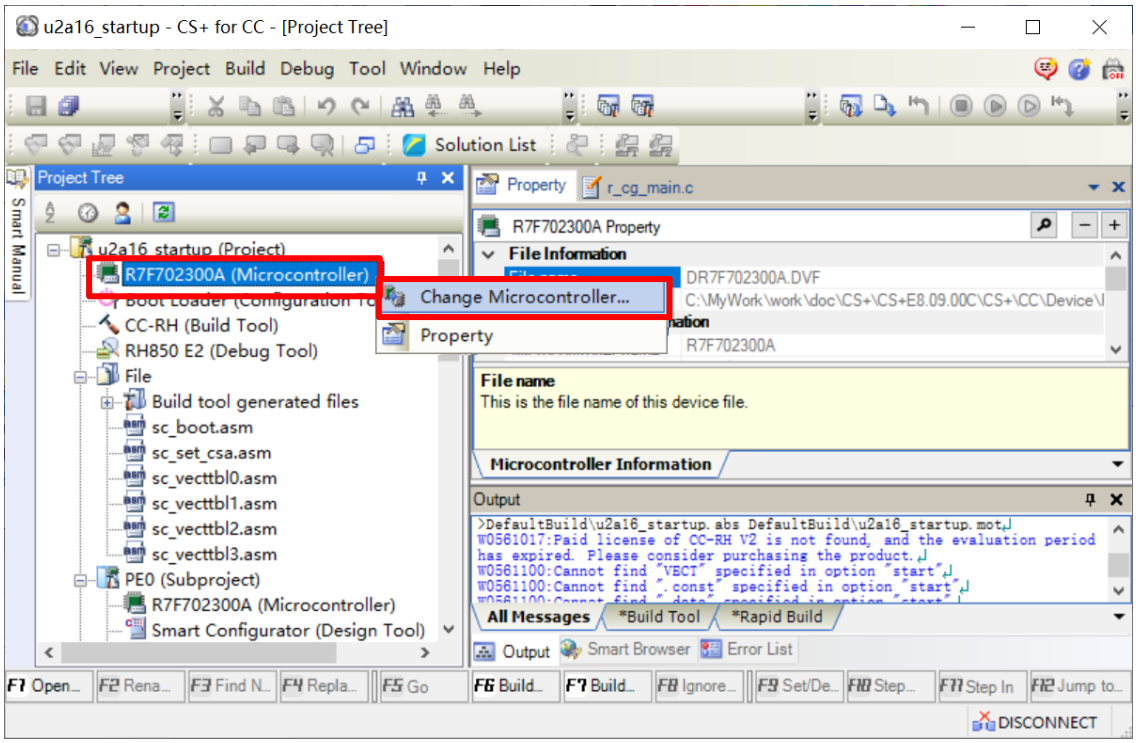

Figure 4-9. Change Microcontroller

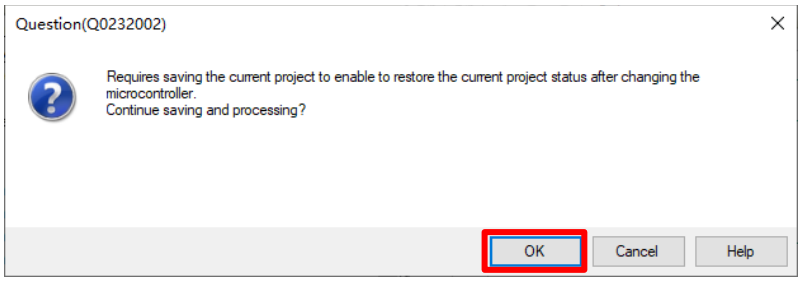

Figure 4-10. Save and continue

(b) In the [Change Microcontroller] dialog box, select the RH850/U2A16 or RH850/U2A8 device to be used.
Example: Changing from R7F702300A (RH850/U2A16 516pin) to R7F702301A (RH850/U2A8 292pin)

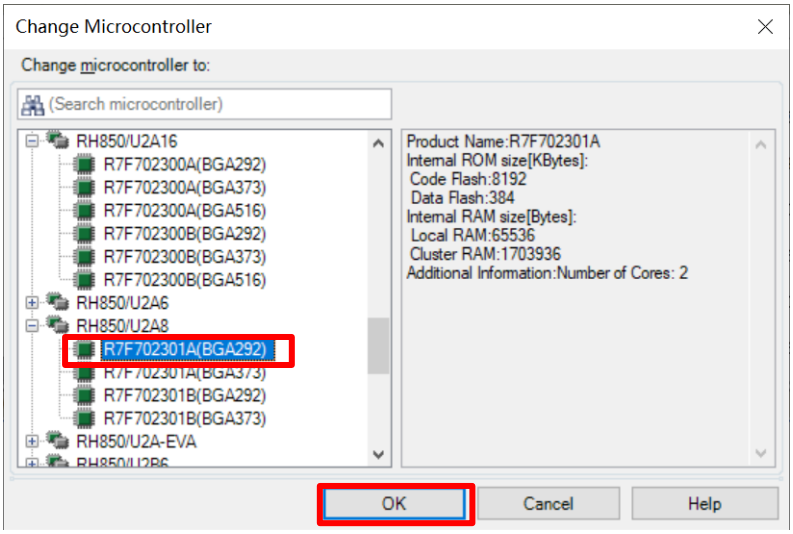

Figure 4-11. Select device to be changed

(c) Change subprojects microcontroller.

Using the same method as (a) and (b), change the microcontroller of each PE to the same device with the main project, which means to change to R7F702301A(RH850/U2A8 292 pin).

But RH850/U2A8 292 pin does not support PE2 and PE3, so remove subproject PE2 and PE3 and interrupt vector table sc_vecttbl2.asm and sc_vecttbl3.asm directly by following way:

Selecting [PE2(Subproject)] and [PE3(Subproject)] and then right-click it and then select [Remove from project]:

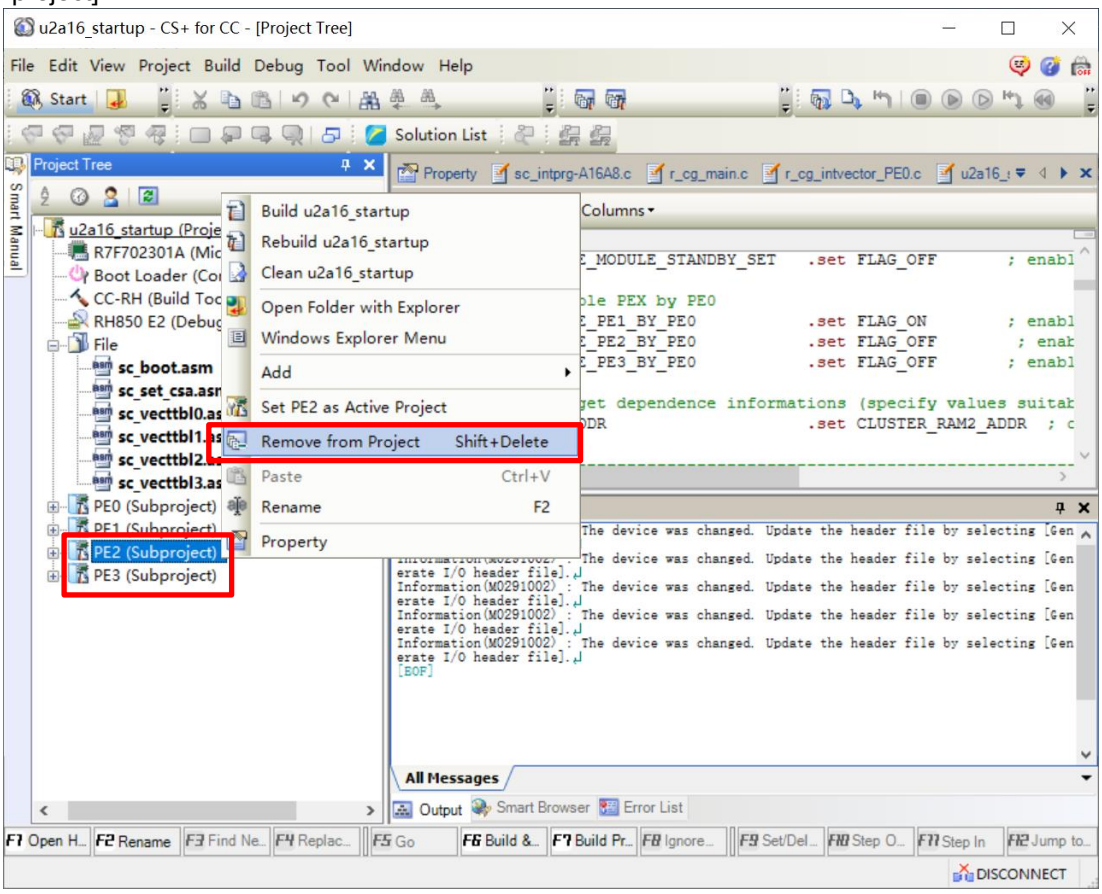

Figure 4-12. Remove unsupported subprojects

Selecting [sc_vecttbl2.asm] and [sc_vecttbl3.asm] and then right-click it and then select [Remove from project]:

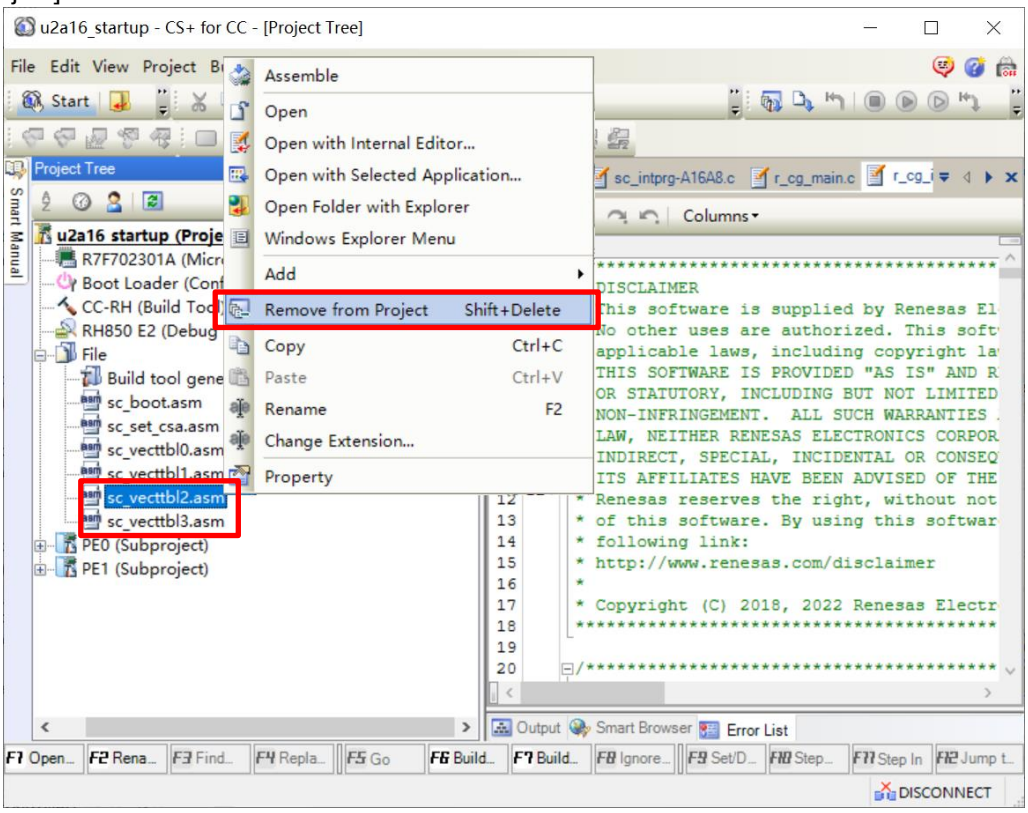

Figure 4-13. Remove unsupported interrupt vector table

(d) Confirm that the microcontroller displayed in the Project Tree panel has become the device to be used after the change.

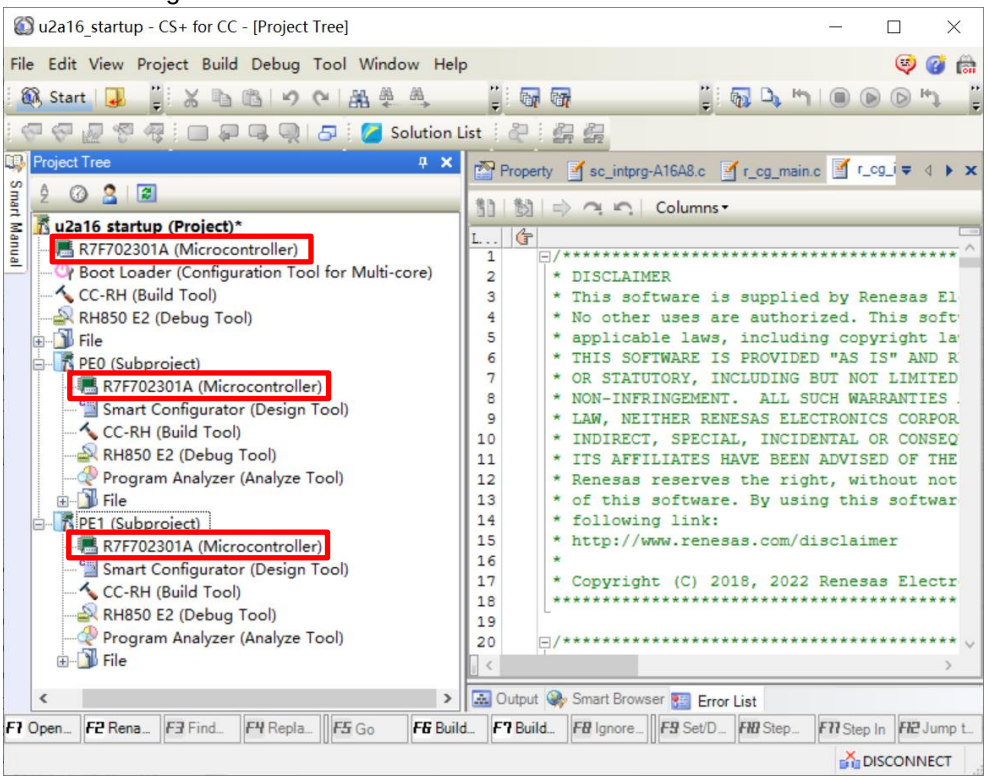

Figure 4-14. Confirm microcontrollers are changed

(e) Save the project by selecting [Save Project] from the [File] menu.

(2) Updating Link section settings, sc_boot.asm for not supported PE
Because PE2 and PE3 are not supported by RH850/U2A8 292pin, the following update is needed:

(a) Section settings for PE2/PE3 that need to be deleted from main project Section Settings.
[Property] panel from [CC-RH (Build Tool)] of main project “u2a16_startup.mtpj” → [Link Options] tab → [Section] → [Section start address]

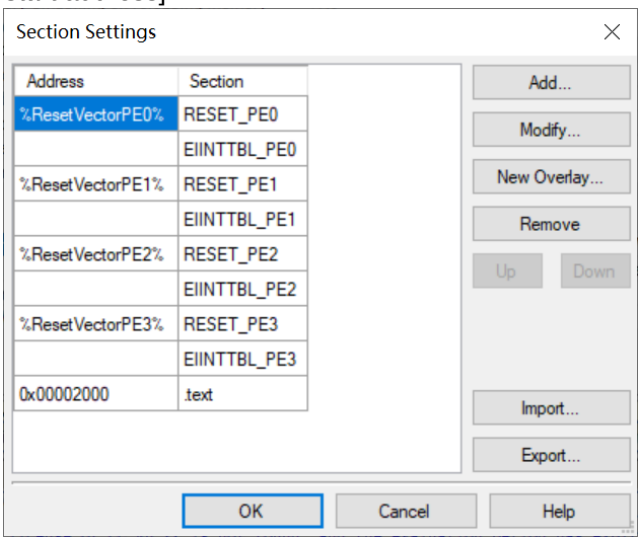

Figure 4-15. Link Section Settings before changing microcontroller

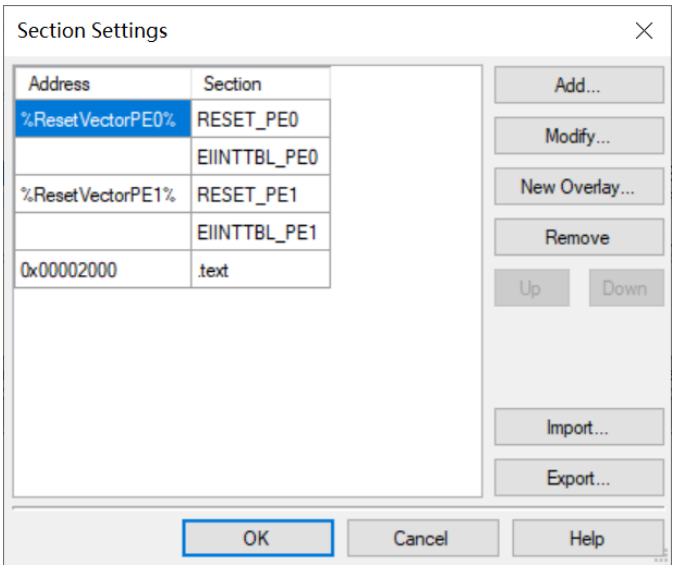

Figure 4-16. Link Section Settings after changing microcontroller

(b) PE2 and PE3 should be disabled in sc_boot.asm.
Line 101~Line102 in sc_boot.asm, FLAG_ON is changed into FLAG_OFF:

```
ENABLE_PE2_BY_PE0           .set FLAG_OFF       ; disable PE2 by PE0
ENABLE_PE3_BY_PE0           .set FLAG_OFF       ; disable PE3 by PE0
```

(3) Specifying the constituent application projects in main project

[Property] panel from [Boot Loader (Configuration Tool for Multi-core)] → [Boot Loader] tab → [Constituent Projects] → [Constituent application projects]

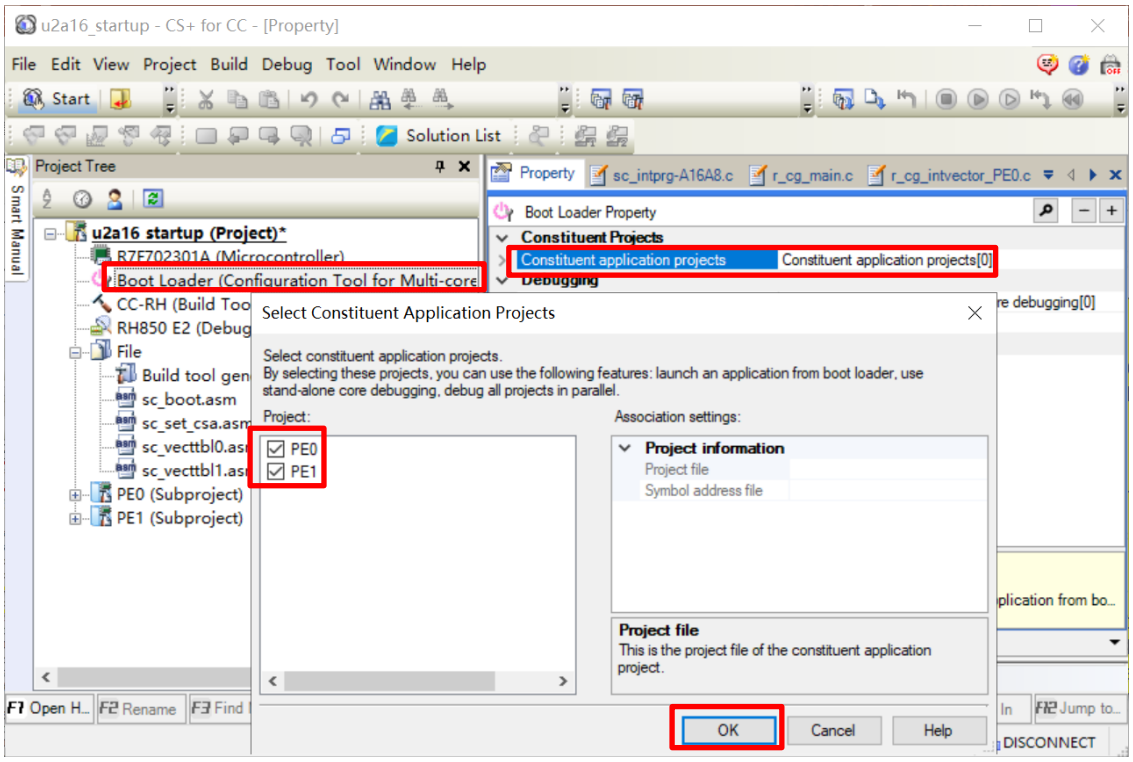

Figure 4-17. Specify the constituent of main project

4.4 Settings in the Sample Project

The main project in sample project is created in CS+ as a [Boot Loader for Multi-core (CC-RH)] project. The subproject is created as an [Application for Multi-core (CC-RH)] project. The include path is added and settings of the following options are changed.

(1) [Property] panel from [CC-RH (Build Tool)] → [Link Options] tab → [Section] → [Section start address]

Settings in the main project u2a16_startup:

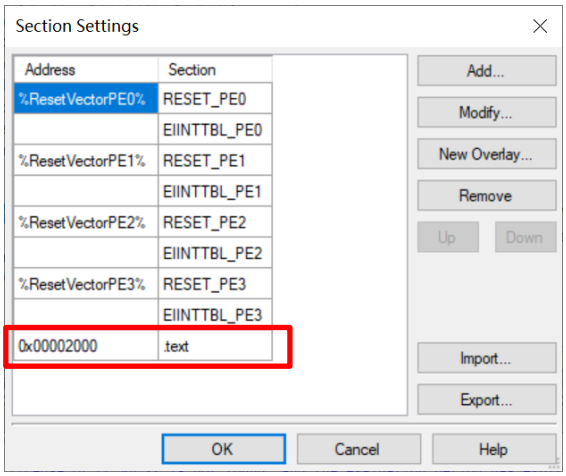

Figure 4-18. Setting in the main project u2a16_startup

Settings in the subproject PE0:

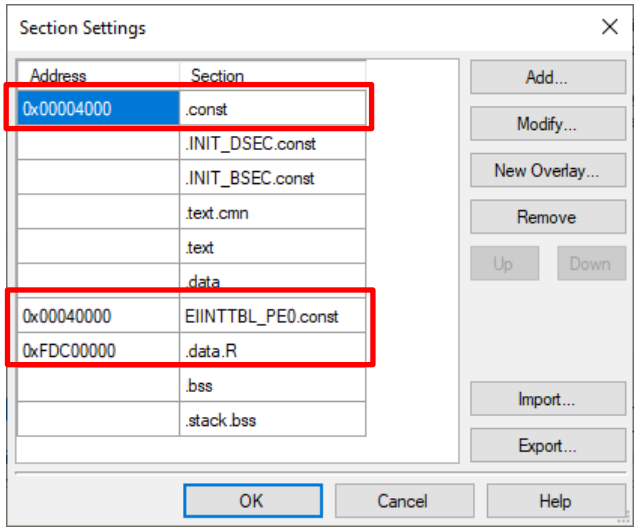

Figure 4-19. Setting in the subproject PE0

Settings in the subproject PE1:

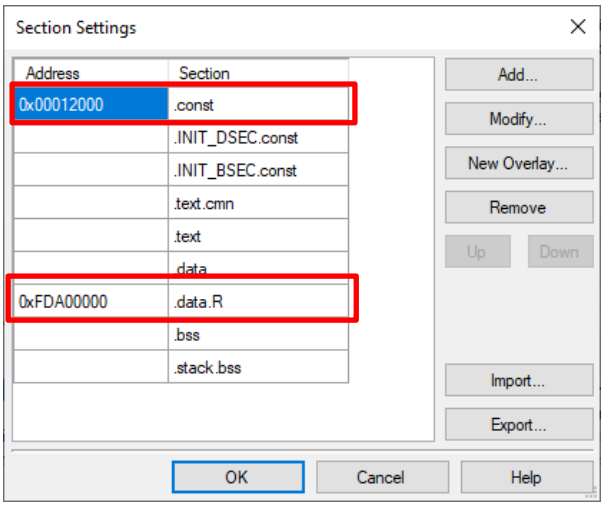

Figure 4-20. Setting in the subproject PE1

Settings in the subproject PE2:

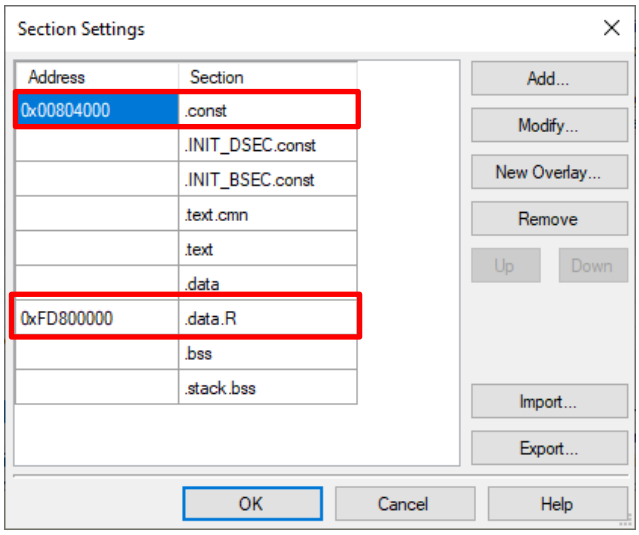

Figure 4-21. Setting in the subproject PE2

Settings in the subproject PE3:

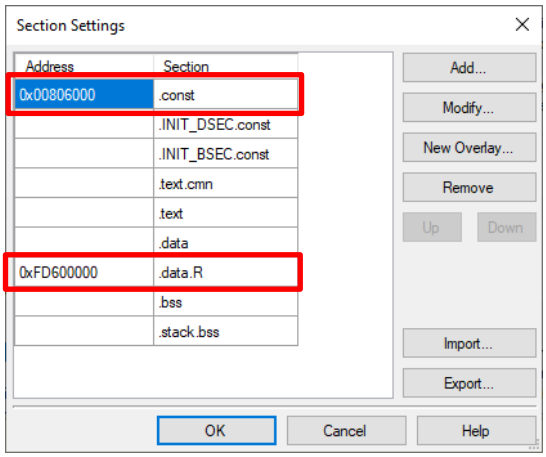

Figure 4-22. Setting in the subproject PE3

(2) [Property] panel from [CC-RH (Build Tool)] → [Link Options] tab → [Section] → [(For multi-core) Section that outputs external defined symbols to the file]

Settings in the main project u2a16_startup:

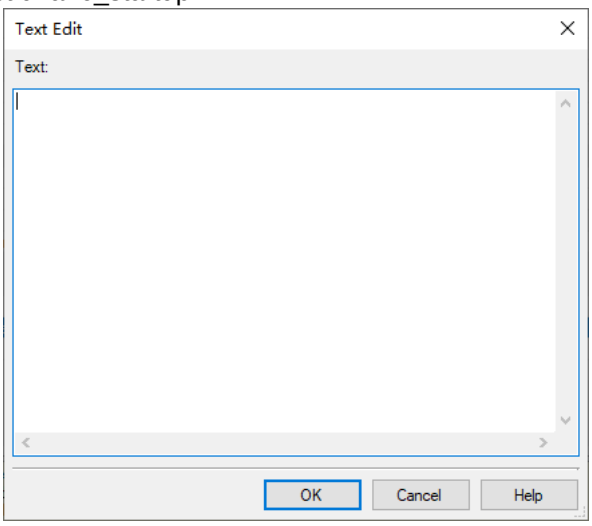

Figure 4-23. Setting in the main project u2a16_startup

Settings in the subproject PE0:

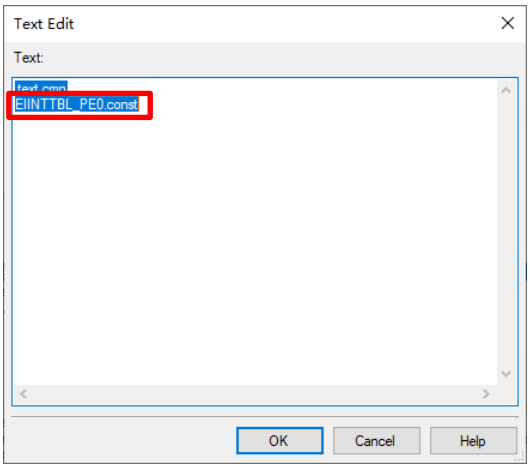

Figure 4-24. Setting in the subproject PE0

Settings in the subproject PE1:
Settings in the subproject PE2:
Settings in the subproject PE3:

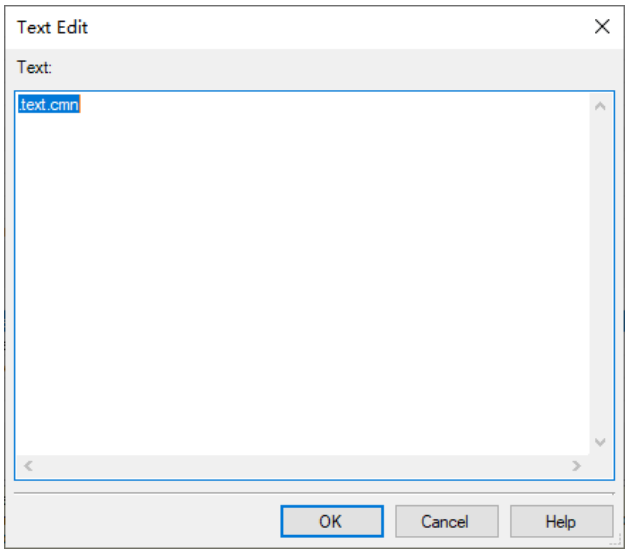

Figure 4-25. Setting in the subproject PE1/PE2/PE3

(3) [Property] panel from [CC-RH (Build Tool)] → [I/O Header File Generation Options] tab → [I/O Header File] → [Update I/O header file on build]

Settings in all projects including main project and all subprojects:

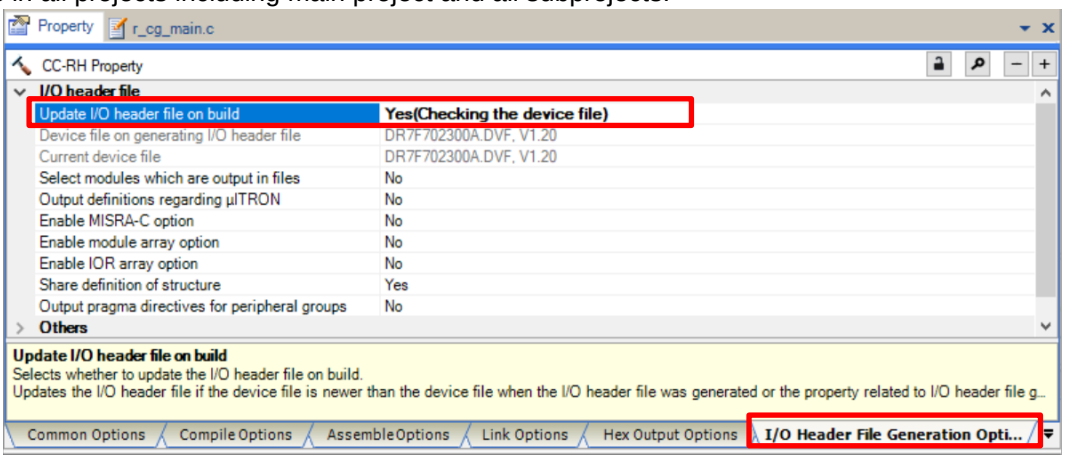

Figure 4-26. Setting in the main project and subproject PE0/PE1/PE2/PE3

5. Operations in the Smart Configurator

This section gives an overview of setting the drivers of peripheral modules of the device and handling of the Smart Configurator for the generation of code.

For details, refer to Smart Configurator User's Guide: CS+ (R20AN0516).

5.1 Setting the Peripheral Modules (Software Components)

(1) Configure the clocks of the device on the [Clocks] tab page.

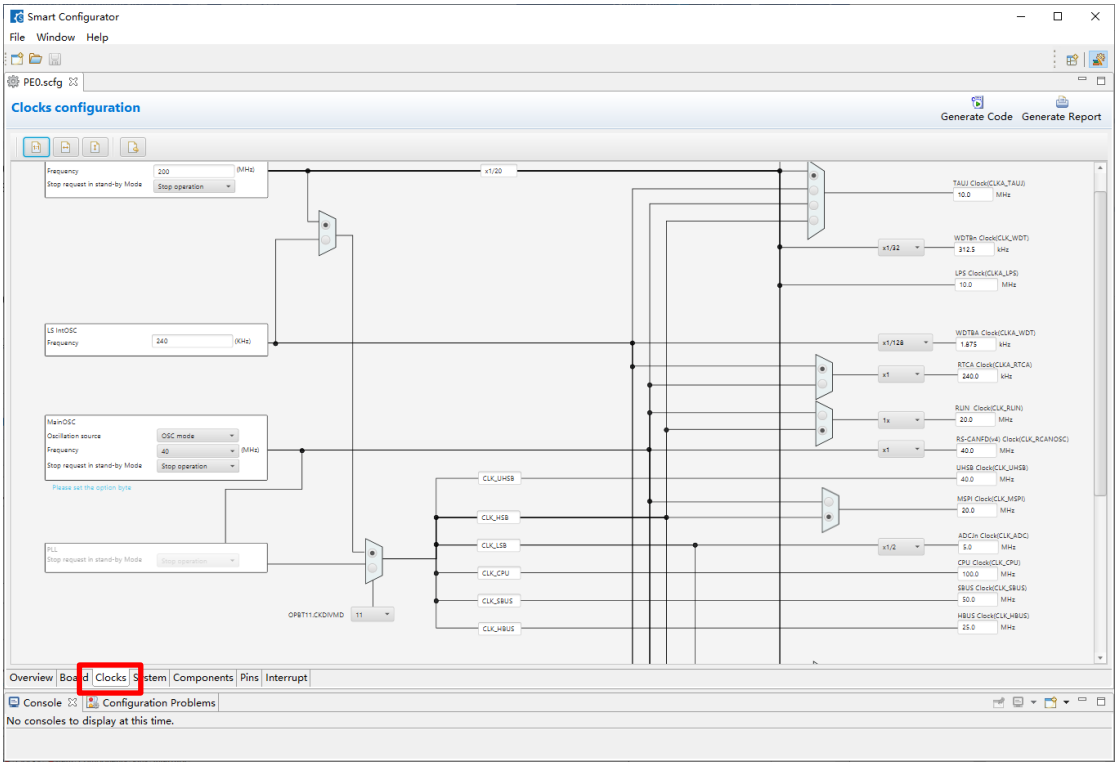

Figure 5-1. Configure clock

(2) Add or set the peripheral modules of the device on the [Components] tab page. The peripheral modules are set as software components. Click on the [Add component] icon.

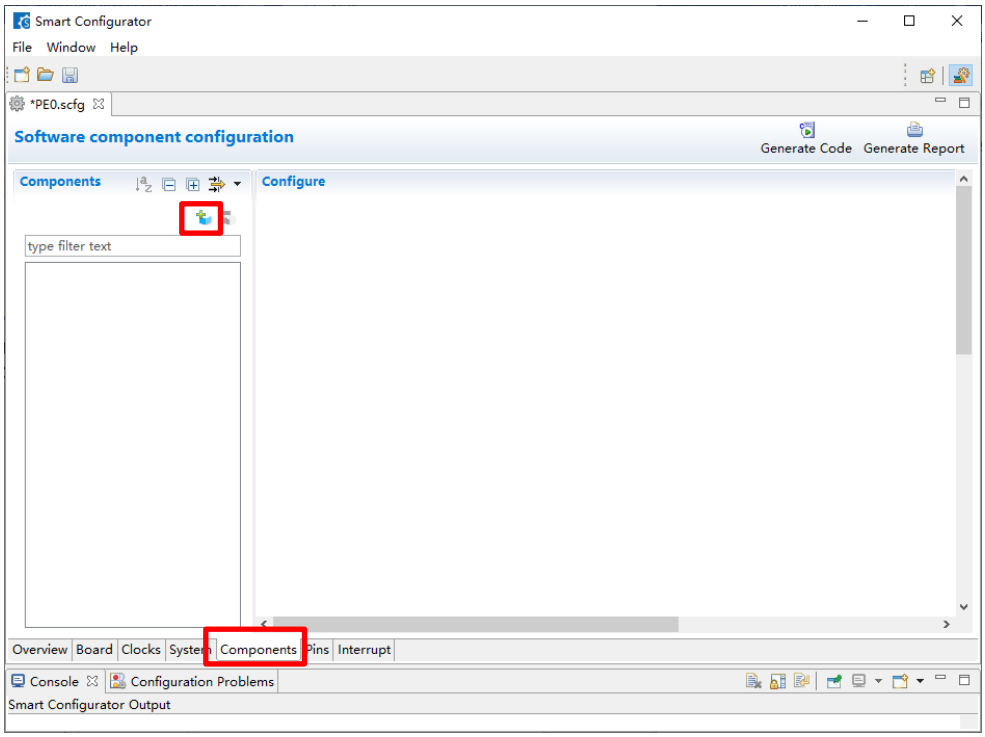

Figure 5-2. Add modules

(3) Select components on the [Software Component Selection] page of the [New Component] dialog box.
Select each component to be used from the list and click on the [Next] button.
Example [Interval Timer] is selected in this guide.

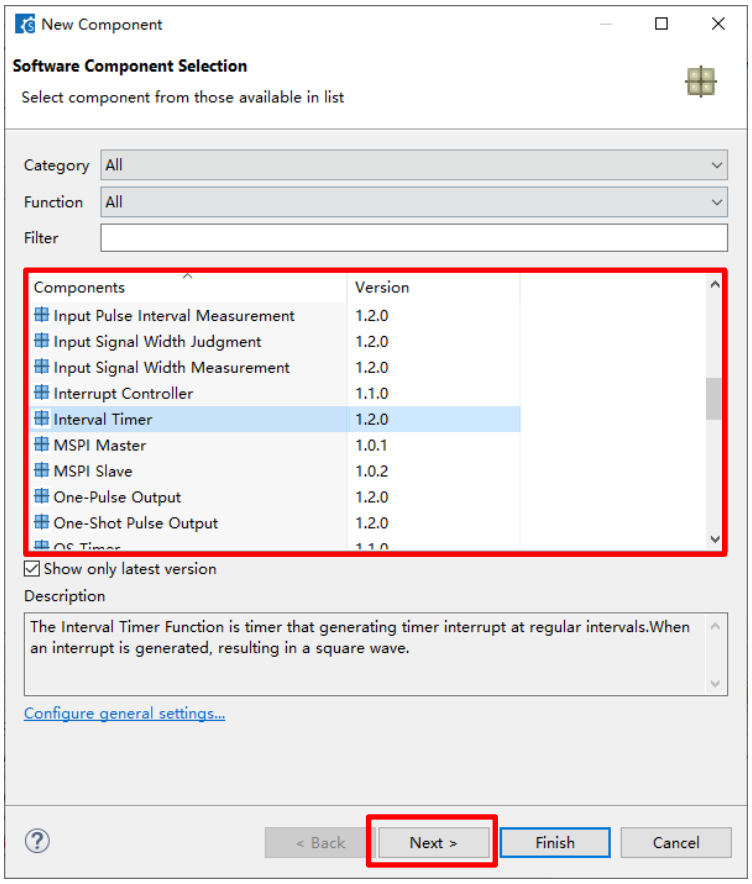

Figure 5-3. Select components

(4) Select the configuration name and resource of the selected component. On the [Add new configuration for selected component] page of the [New Component] dialog box, enter an appropriate configuration name or use the default name. Select the resource or use the default resource. After you have made the selections, click on the [Finish] button.

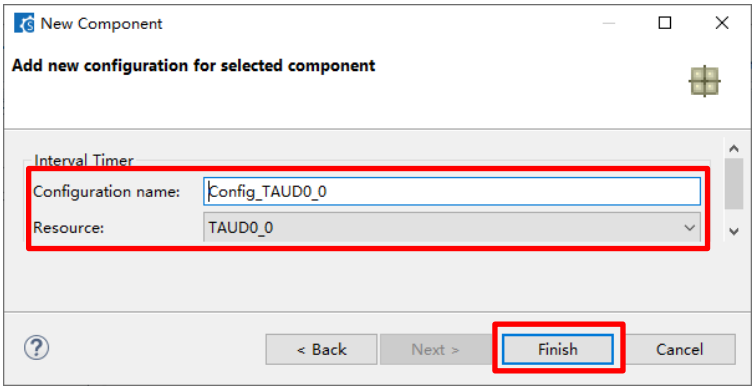

Figure 5-4. Input Configuration name and select Resource

(5) Set the configuration of the component. Click on the configuration icon in the Component Tree panel and make detailed settings in the right-hand panel.

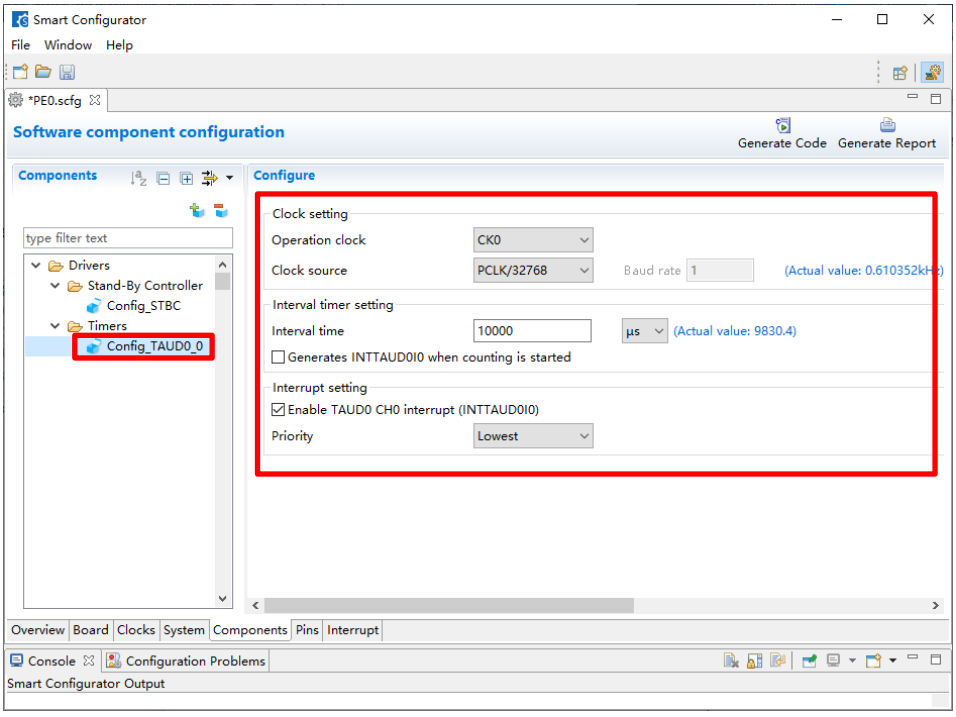

Figure 5-5. Set component configuration

(6) Repeat steps (2) to (5) for each component that you intend to use.
In this sample project, Stand-by Controller component must be added for providing the module standby mode cancelling function.
RH850/U2A supports Module Standby Mode. After the reset is released, all peripherals enter module standby modes. Register access to the module in module standby mode is prohibited. So before using any modules, module standby mode should be cancelled in advance.
For RH850/U2A Stand-by Controller component, all module standby mode set and cancel functions are always generated and there is no need to configure UI.

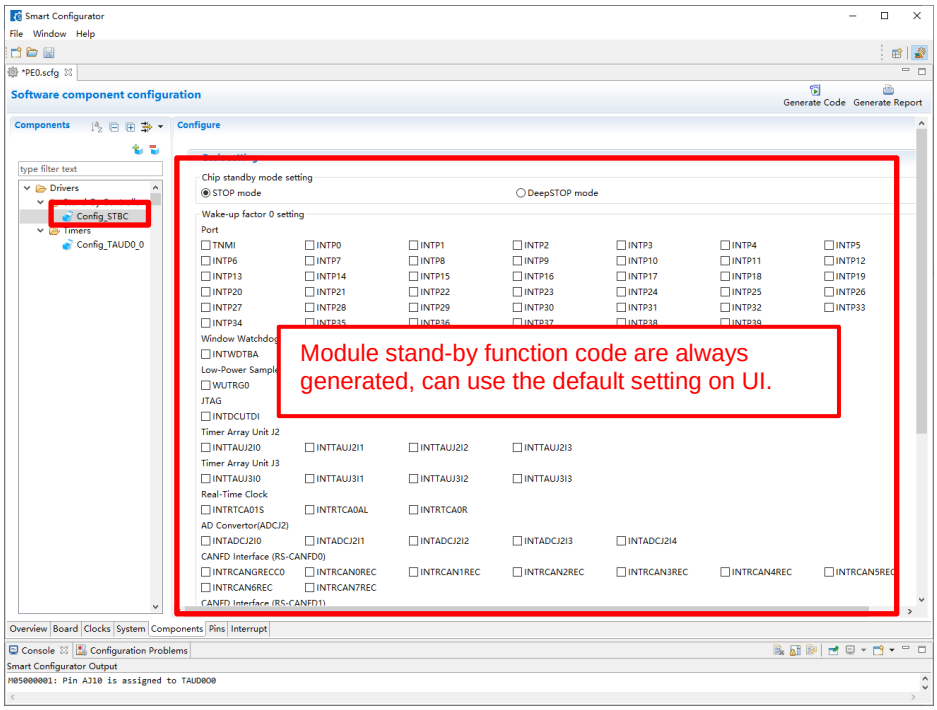

Figure 5-6. Add Stand-by Controller component

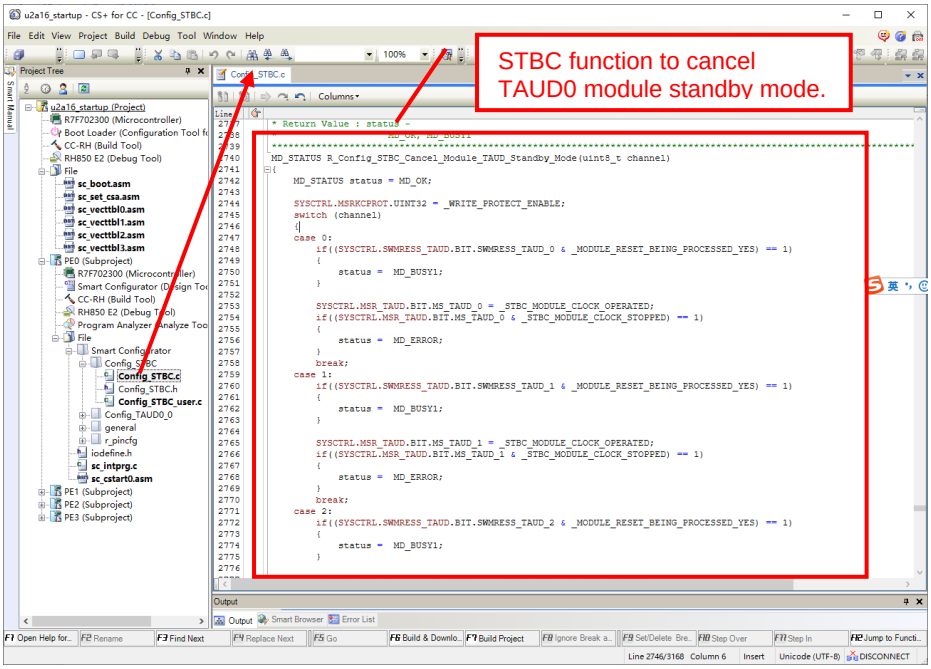

Figure 5-7. Cancel stand-by mode function for TAUD

5.2 Generating Drivers

Click on the [Code Generator] button to generate code. The source files generated by the Smart Configurator are stored in the <ProjectDir>\src\smc_gen folder.

<ProjectDir> is the folder containing the project files (.scfg) for the Smart Configurator.

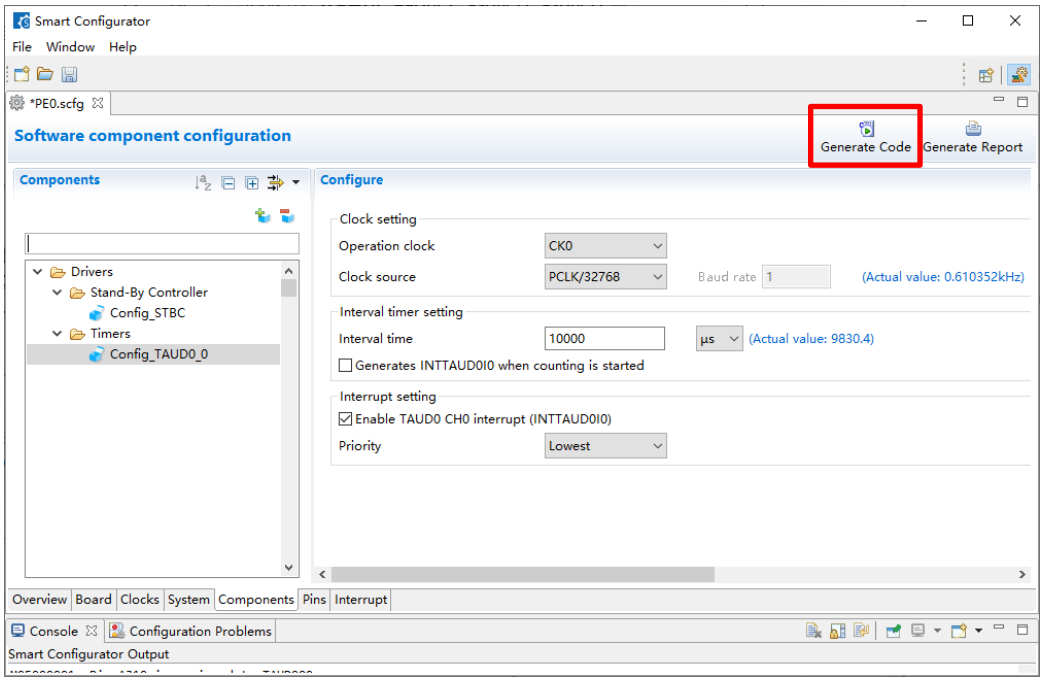

Figure 5-8. Generating code

5.3 Adding the Code to the User Code Area

Some generated source files have a user code area for the writing of user code. Open such files in an editor from CS+ environment that you are using and add the code (e.g. code for interrupt processing) to the user code areas as necessary.

Example: File generated for the interval timer component

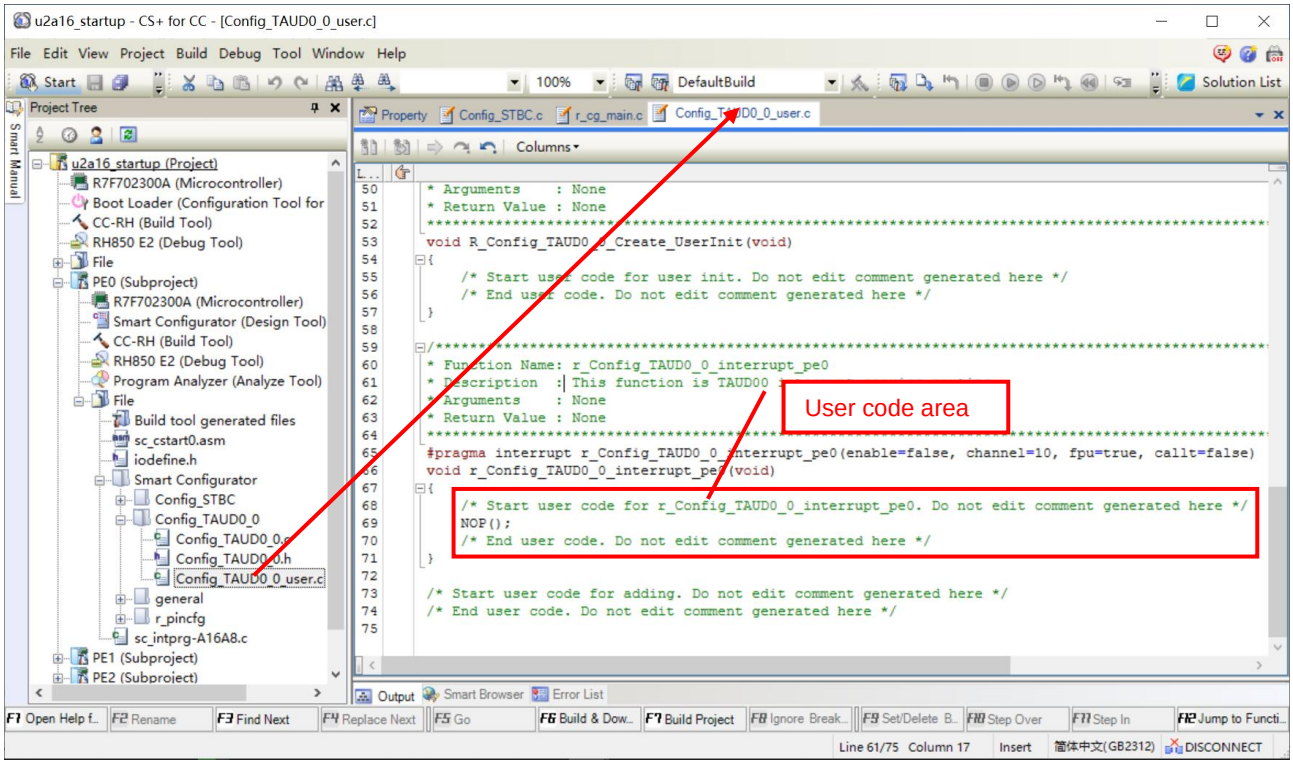

Figure 5-9. Add user code to the user code area of interrupt routine

5.4 Adding the Code to main_pm0()

The main function is in "<ProjectDir>\src\smc_gen\general\r_cg_main.c". Open the file in an editor from CS+ environment that you are using and add the code to the user code area.

NOTE: After re-generating code, please rename function main() to main_pm0(). Please refer 2.2-4 for the detail.

Example: Add code to the main_pm0() function for PE0 subproject.

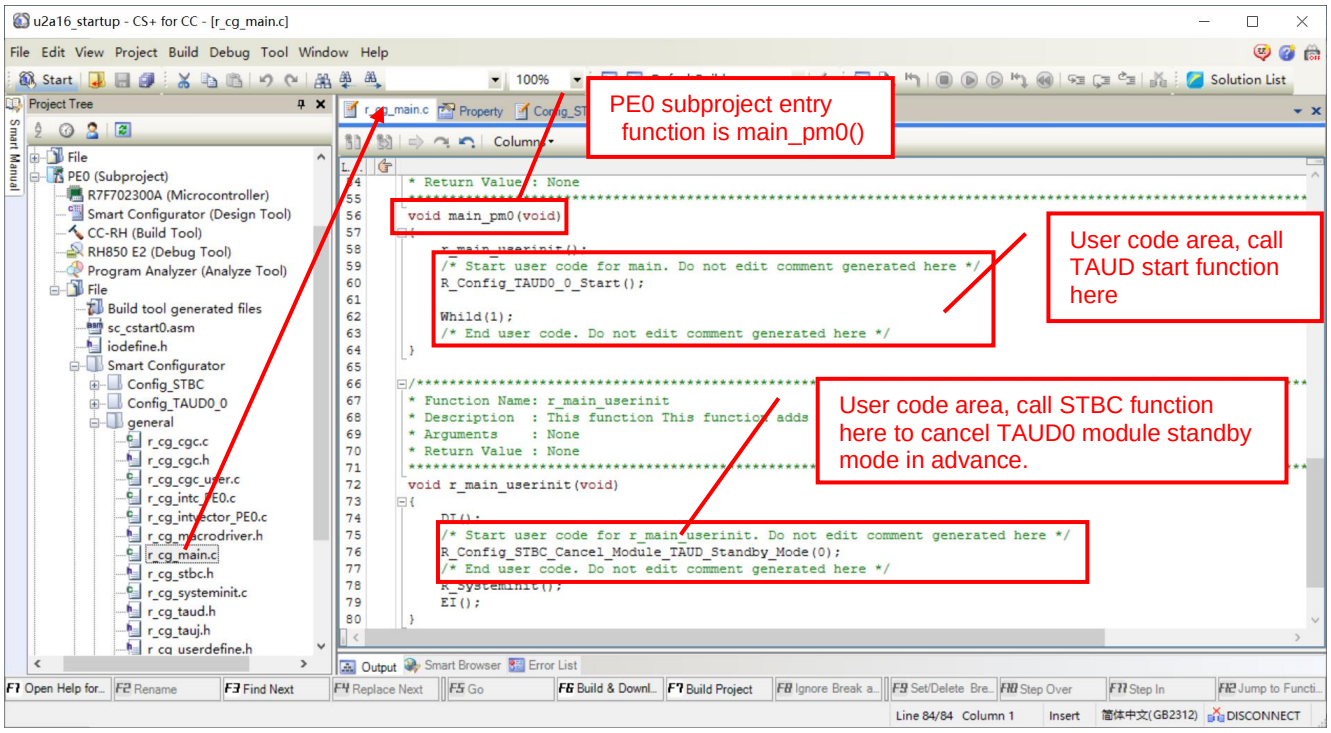

Figure 5-10. Add user code to main_pm0() function

Revision History

Rev.	Section	Description
1.00	All	New version
1.01	2. Outline of the Sample Projects	2.3 Notes on the Sample Projects: added two new notes: No.1 and No.4
1.02	Smart Configurator page	Target Device: add U2A6
	1. Overview	Table 1-1 Operating Environment: CS+V7.00.00 is updated to CS+ V8.08.00
	4. Description of the Sample Project	4.1 Configuration of the Sample Project: Update figure: Figure 4-1. Common folder and Main project folder Figure 4-2. Subproject folder PEn(n=0~3) Update table: Table 4-2. File Configuration of the subproject (corresponding to Figure 4-2)
		4.2 Basic Operation Procedure: Update figure Figure 4-5. Setting the path for Smart Configurator Update CS+ Integrated Development Environment User's Manual version and document number in (9) Execution and debugging Update (4) Starting the Smart Configurator by adding c) content and relative figure Figure 4-6. RH850/U2A16 Package Selection
		4.3 Procedure for Changing the Device Update figure: Figure 4-9. Change Microcontroller Figure 4-11. Select device to be changed Figure 4-12. Remove unsupported subprojects Figure 4-14. Confirm microcontrollers are changed Figure 4-15. Link Section Settings before changing microcontroller Figure 4-16. Link Section Settings after changing microcontroller Figure 4-17. Specify the constituent of main project Add content about removing not supported interrupt vector table and relative figure Figure 4-13. Remove unsupported interrupt vector table in (1) Changing the microcontroller, c) Change subprojects microcontroller

Rev.	Section	Description
1.02	4. Description of the Sample Project	4.4 Settings in the Sample Project Update figure Figure 4-26. Setting in the main project and subproject PE0/PE1/PE2/PE3
	5. Operations in the Smart Configurator	5.3 Adding the Code to the User Code Area Update figure: Figure 5-9. Add user code to the user code area of interrupt routine Figure 5-10. Add user code to main_pm0() function

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 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

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

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

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

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

Contact information

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