

RH850 Smart Configurator

R20AN0516EJ0100

Rev.1.00

Nov 1, 2018

User's Guide: CS+

Introduction

This application note describes the basic usage of the RH850 Smart Configurator (hereafter called the Smart Configurator), and the procedure for adding its output files to CS+ projects.

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

- CS+ (CS+ for CC) V7.00.00 and later
- RH850 Smart Configurator V1.1.0 and later
- CS+ RH850 Smart Configurator Communication Plugins V1.02.00 and later

Target Devices and Compilers

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

<https://www.renesas.com/smart-configurator>

Contents

| | |
|---|----|
| 1. Overview | 3 |
| 1.1 Purpose | 3 |
| 1.2 Features | 3 |
| 2. Before Using the Smart Configurator | 4 |
| 2.1 Preparing the CS+ (CS+ for CC) Integrated Development Environment | 4 |
| 2.2 Installing the Smart Configurator | 4 |
| 2.3 Setting the CS+ Integrated Development Environment | 4 |
| 2.3.1 Checking the plug-in settings | 4 |
| 2.3.2 Checking the setting of the execution path | 5 |
| 2.4 Uninstalling the Smart Configurator | 5 |
| 2.5 Preparing Sample Projects | 6 |
| 3. Operating the Smart Configurator | 7 |
| 3.1 Procedure for Operations | 7 |
| 3.2 Starting the Smart Configurator | 8 |
| 3.3 File to be Saved as Project Information | 8 |
| 3.4 Window | 9 |
| 3.4.1 Main menu | 10 |
| 3.4.2 Toolbar | 10 |
| 3.4.3 Smart Configurator view | 11 |
| 3.4.4 MCU Package view | 11 |
| 3.4.5 Console view | 12 |
| 3.4.6 Configuration Problems view | 12 |

| | | |
|-------|---|----|
| 4. | Setting of Peripheral Modules..... | 13 |
| 4.1 | Board Settings..... | 13 |
| 4.1.1 | Selecting the device..... | 13 |
| 4.1.2 | Selecting the board..... | 14 |
| 4.1.3 | Exporting board settings..... | 15 |
| 4.1.4 | Importing board settings..... | 15 |
| 4.2 | Clock Settings..... | 16 |
| 4.3 | Component Settings..... | 17 |
| 4.3.1 | Adding Code Generator components..... | 17 |
| 4.3.2 | Switching between the component view and hardware view..... | 19 |
| 4.3.3 | Removing a software component..... | 20 |
| 4.3.4 | Component configuration..... | 21 |
| 4.3.5 | Changing the resource for a component..... | 22 |
| 4.4 | Pin Settings..... | 25 |
| 4.4.1 | Changing the pin assignment of a software component..... | 26 |
| 4.4.2 | Resolving pin conflicts..... | 27 |
| 4.4.3 | Assigning pins using the MCU Package view..... | 29 |
| 4.4.4 | Exporting pin settings..... | 30 |
| 4.4.5 | Importing pin settings..... | 30 |
| 4.5 | Interrupt Settings..... | 31 |
| 4.5.1 | Changing the interrupt priority level and OS management setting..... | 31 |
| 5. | Generating Source Code..... | 32 |
| 5.1 | Registering Generated Source Code with CS+..... | 32 |
| 5.2 | Configuration of Generated Files and File Names..... | 33 |
| 5.3 | Initializing Clocks..... | 35 |
| 5.4 | Initializing Pins..... | 36 |
| 5.5 | Initializing Interrupts..... | 37 |
| 6. | Managing Conflicts..... | 38 |
| 6.1 | Resource Conflicts..... | 38 |
| 7. | Creating User Programs..... | 39 |
| 7.1 | Adding Custom Code in the Case of Code Generator..... | 39 |
| 8. | Backing up Generated Source Code..... | 41 |
| 9. | Generating Reports..... | 42 |
| 9.1 | Report on All Configurations (Text File)..... | 42 |
| 9.2 | Configuration of Pin Function List and Pin Number List (in csv Format)..... | 43 |
| 9.3 | Image of MCU Package (in png Format)..... | 43 |
| 10. | Help..... | 44 |
| 10.1 | Help..... | 44 |
| 11. | Documents for Reference..... | 45 |
| | Website and Support..... | 46 |

1. Overview

1.1 Purpose

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

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

1.2 Features

The Smart Configurator is a utility for combining software to meet your needs. It handles the following two functions to support the embedding of drivers from Renesas in your systems: generating driver code and making pin settings.

2. Before Using the Smart Configurator

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

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

2.2 Installing the Smart Configurator

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

<https://www.renesas.com/smart-configurator>

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

2.3 Setting the CS+ Integrated Development Environment

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

2.3.1 Checking the plug-in settings

Select [Plug-in Manager] from [Tool] of CS+ menu, and confirm that there is a tick against “Smart Configurator for RH850 Communication Plug-in”. Tick it if it is not.

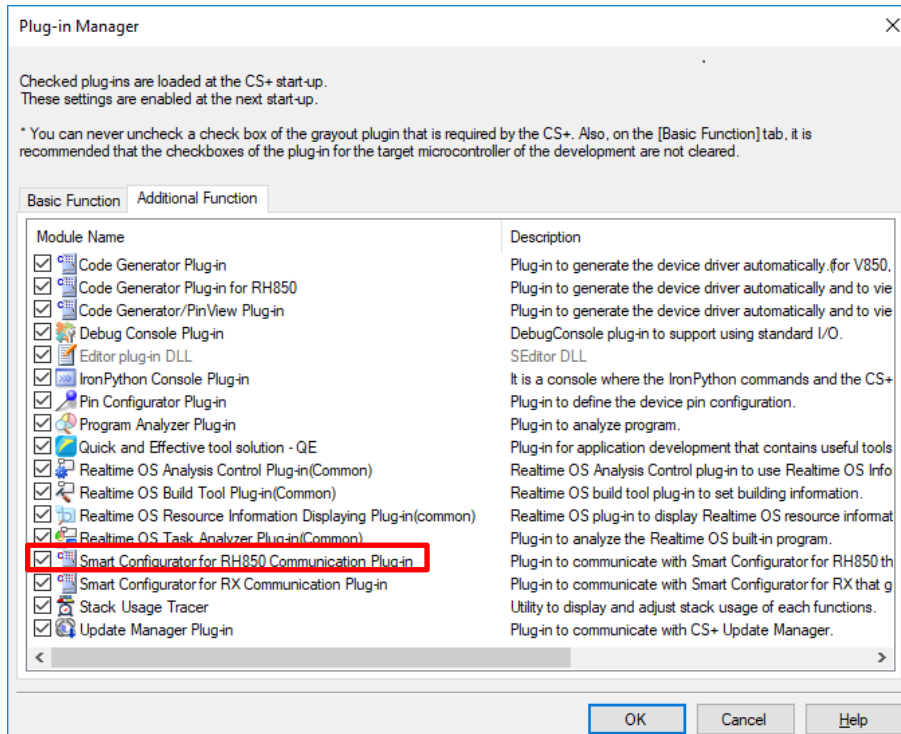


Figure 2-1 Plug-in Manager

2.3.2 Checking the setting of the execution path

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

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

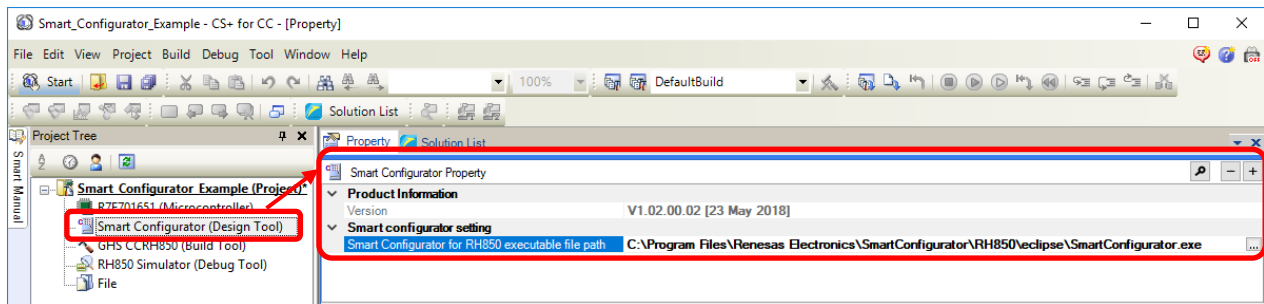


Figure 2-2 Displaying the Property

“Smart Configurator for RH850 executable file path” shows the executable file of the Smart Configurator. The following path is set when the Smart Configurator is installed with the default setting (where “CS+” and “SmartConfigurator” are in the same level).

32-bit environment:

“C:\Program Files\Renesas Electronics\SmartConfigurator\RH850\eclipse\SmartConfigurator.exe”

64-bit environment:

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

When manually specifying the path of the executable file, “Smart Configurator for RH850 executable file path” can be set as either a relative or an absolute path.

2.4 Uninstalling the Smart Configurator

If you wish to uninstall the Smart Configurator, select “Smart Configurator for RH850” and “CS+ SC Communication Plugins for RH850” from [Apps and Features] in the control panel and uninstall them.

2.5 Preparing Sample Projects

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

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

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

32-bit environment:

“C:\Program Files\Renesas Electronics\SmartConfigurator\RXRH850\RH850F1KM_SampleProjects”

64-bit environment:

“C:\Program Files (x86)\Renesas Electronics\SmartConfigurator\RXRH850\RH850F1KM_SampleProjects”

3. Operating the Smart Configurator

3.1 Procedure for Operations

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

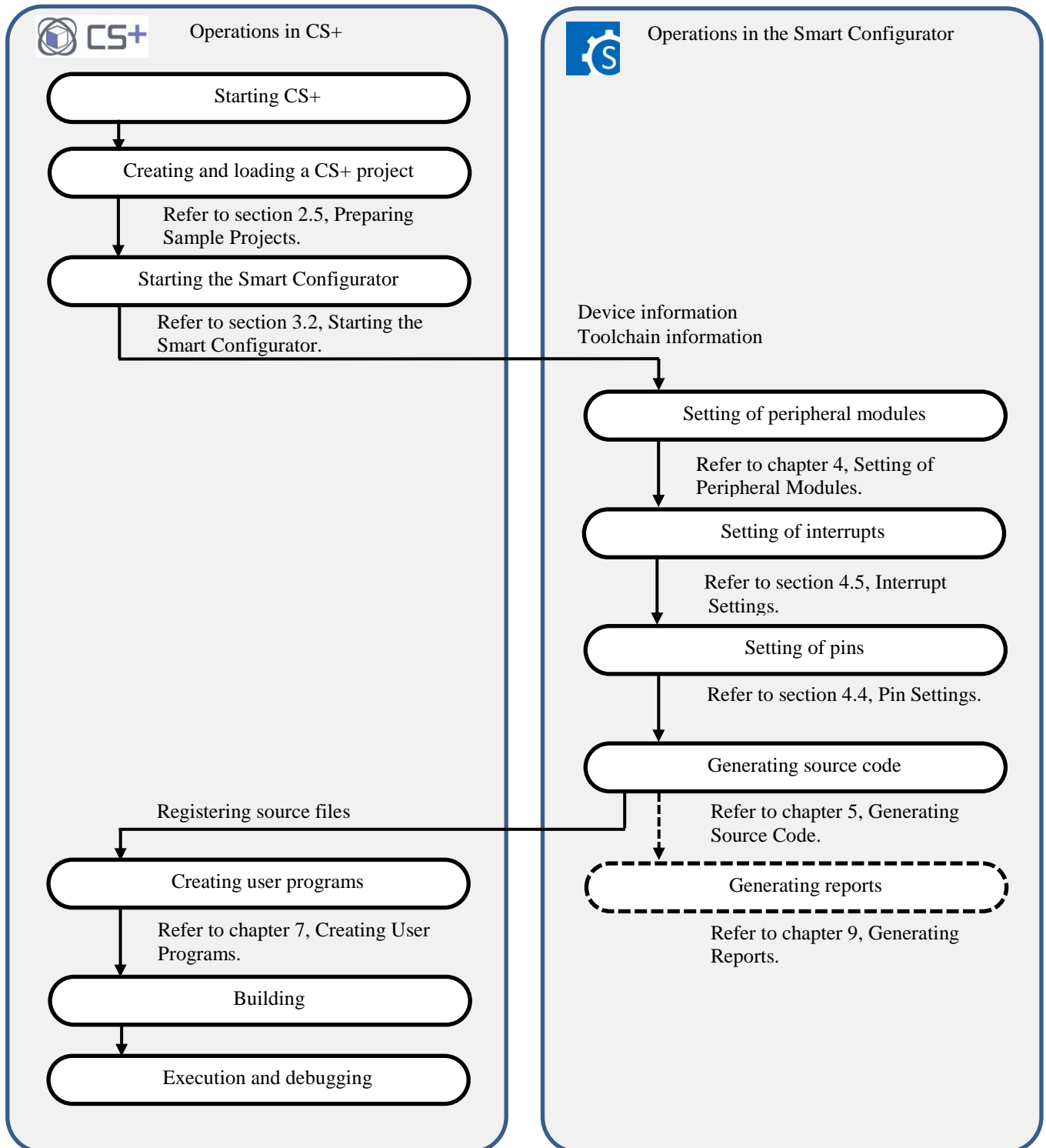


Figure 3-1 Procedure for Operations

3.2 Starting the Smart Configurator

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

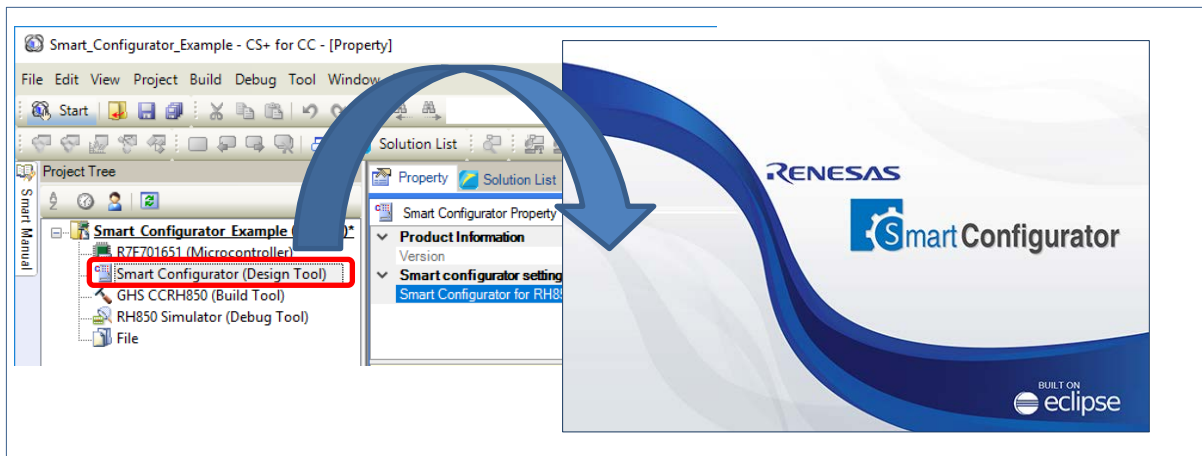


Figure 3-2 Activation of Smart Configurator

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

3.3 File to be Saved as Project Information

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

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

3.4 Window

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

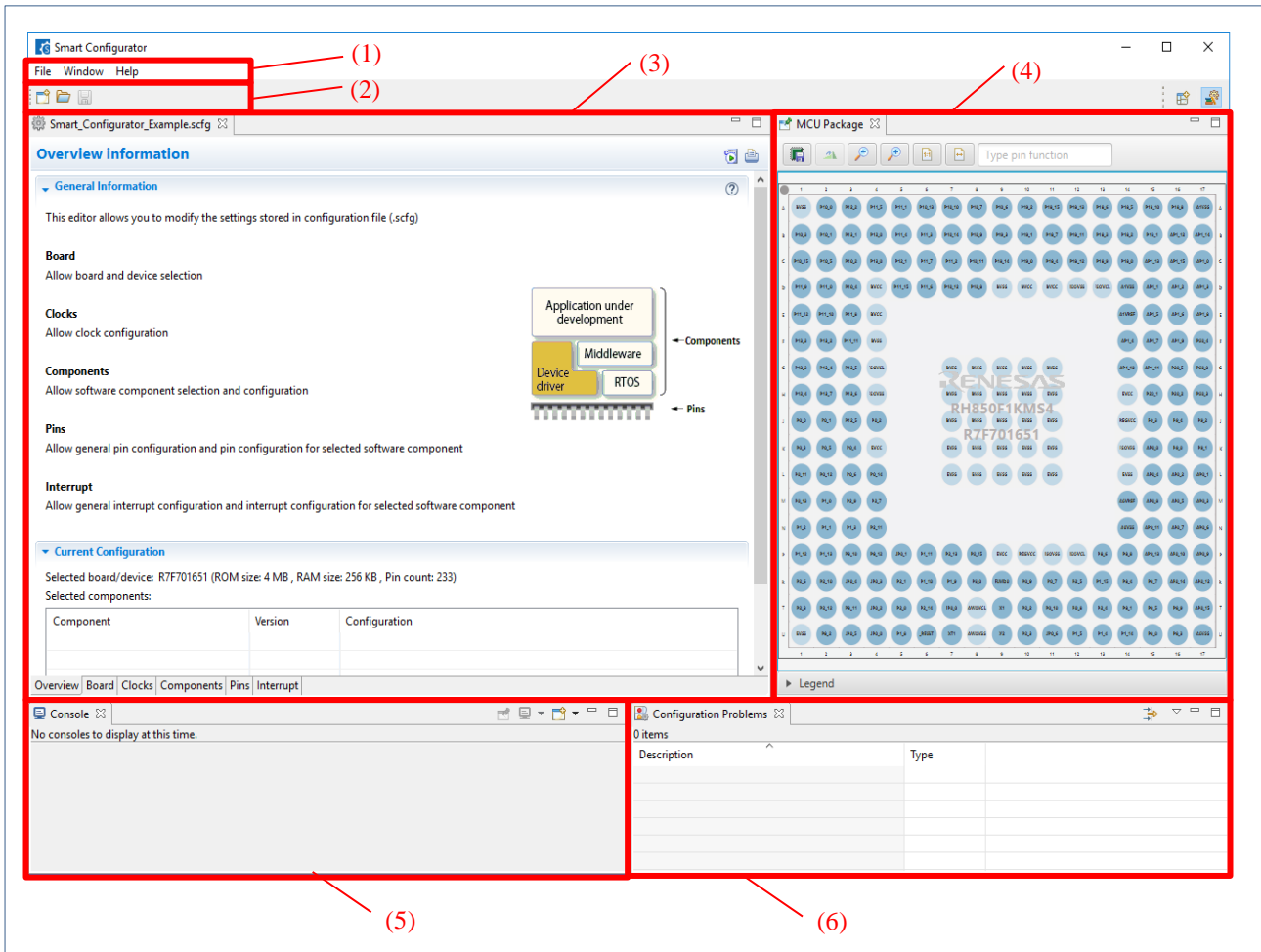


Figure 3-3 Main Window

- 1) Menu bar
- 2) Main toolbar
- 3) Smart Configurator view
- 4) MCU Package view
- 5) Console view
- 6) Configuration Problems view

3.4.1 Main menu

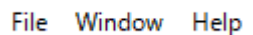


Table 3-1, Main Menu Items, lists the items of the main menu.

Table 3-1 Main Menu Items




| Menu | | Details |
|--------|---------------|---|
| File | New | The dialog box [New Smart Configuration File], which is used to create a new project, is displayed. |
| | Open | The dialog box [Open], which opens an existing project, is displayed. |
| | Save | Saves a project with the same name. |
| | Restart | Smart Configurator is restarted. Do not use this menu item in general, as it leads to deletion of the project settings handed over from CS+. |
| | Exit | Execution of the Smart Configurator is terminated. |
| Window | Preference | The dialog box [Preference], which is used to specify the properties of the project, is displayed. |
| | Show view | The dialog box [Show view], which is used to set the view of the window, is displayed. |
| Help | Help Contents | The help menu is displayed. |
| | About | The version information is displayed. |

3.4.2 Toolbar



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

Table 3-2 Toolbar Buttons and Related Menu Items

| Toolbar button | Related menu item |
|---|-------------------|
|  | [File] → [New] |
|  | [File] → [Open] |
|  | [File] → [Save] |

3.4.3 Smart Configurator view

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

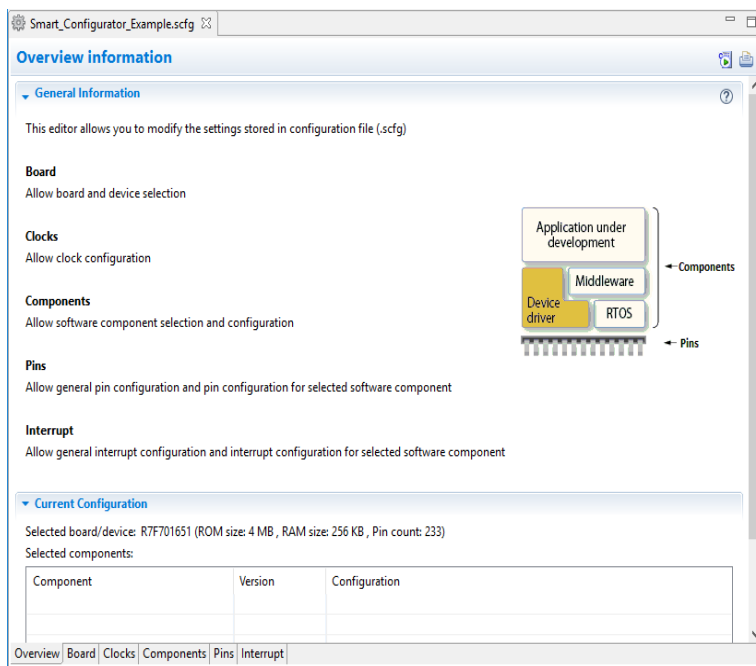


Figure 3-4 Smart Configurator View

3.4.4 MCU Package view

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

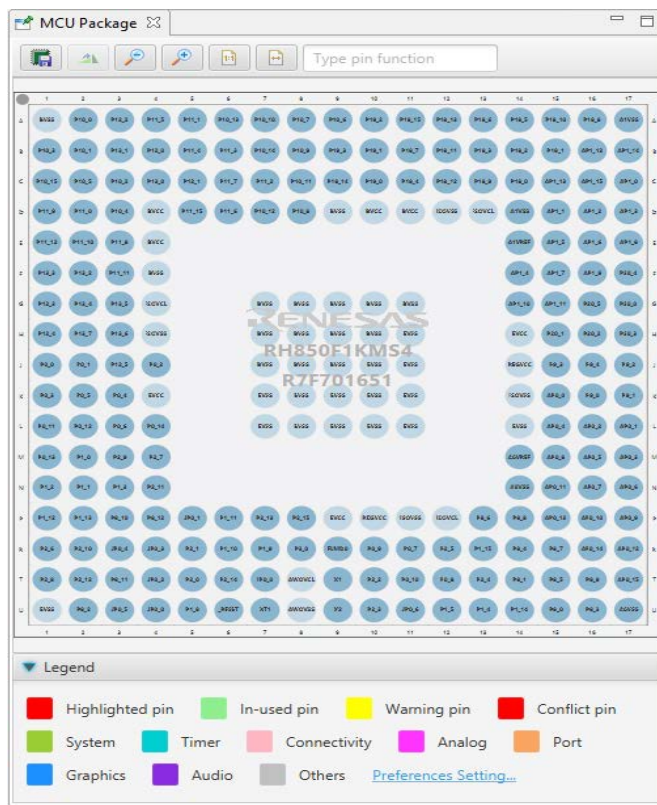


Figure 3-5 MCU Package View

3.4.5 Console view

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



Figure 3-6 Console View

3.4.6 Configuration Problems view

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

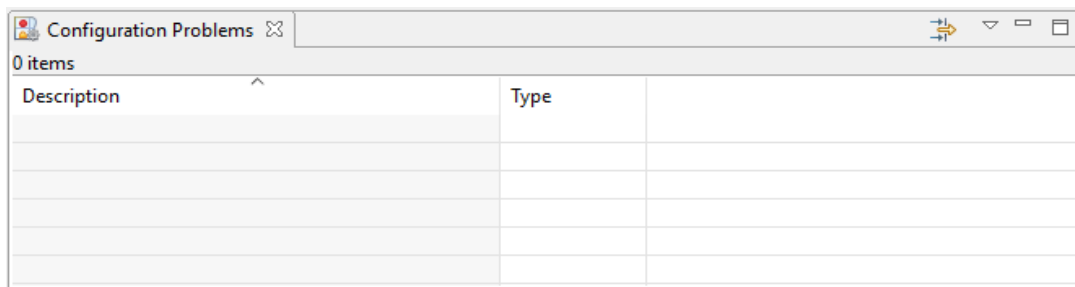


Figure 3-7 Configuration Problems View

4. Setting of Peripheral Modules

You can select peripheral modules from the Smart Configurator view.

4.1 Board Settings

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

4.1.1 Selecting the device

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

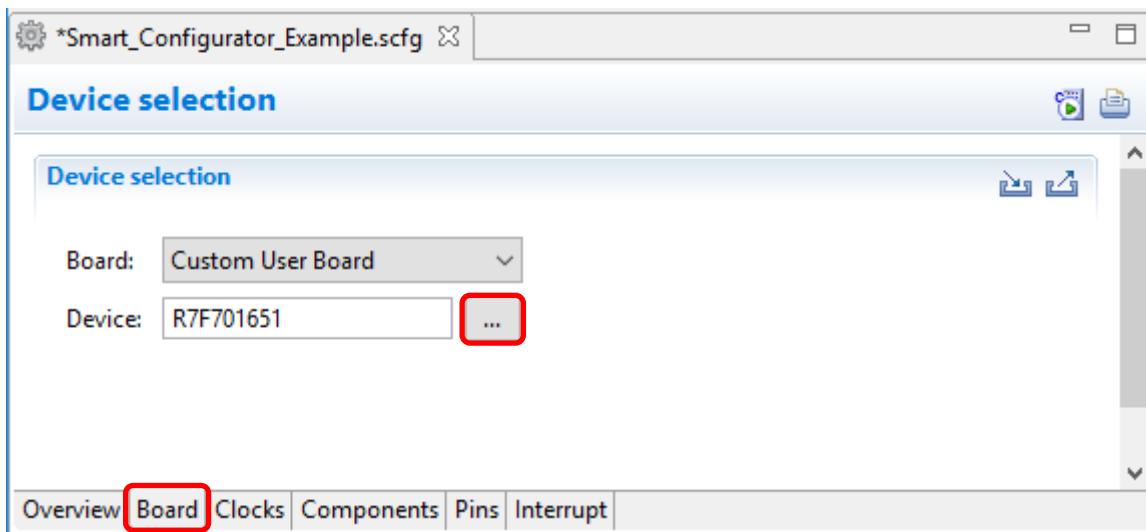


Figure 4-1 Selecting the Device

4.1.2 Selecting the board

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

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

The settings of a board are defined in a .bdf file.

A board from an alliance partner can be selected after importing the .bdf file provided by the partner.

When starting the Smart Configurator from CS+, do not select a board that causes the target device to be changed to another device.

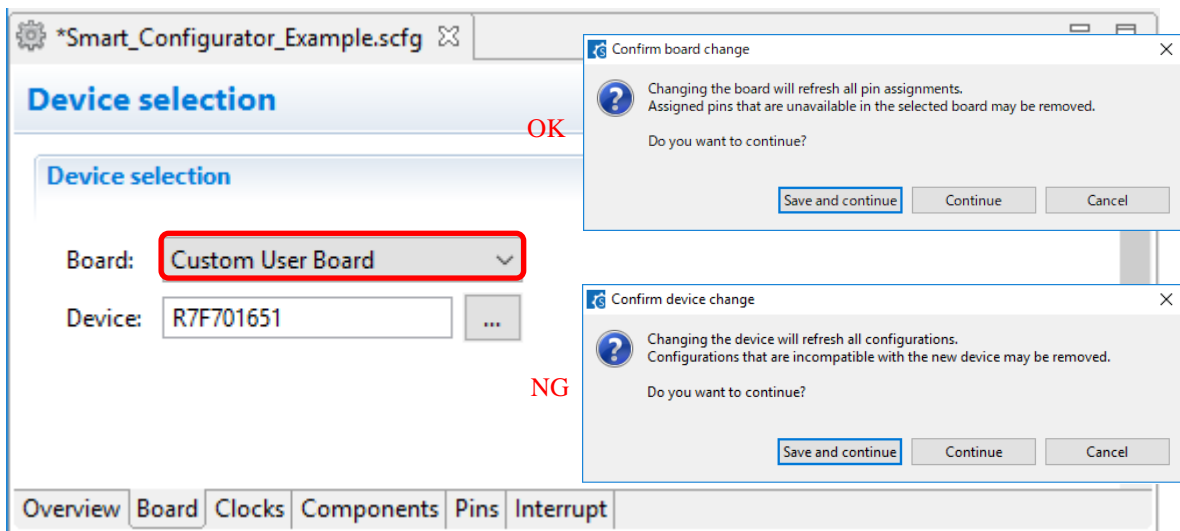



Figure 4-2 Selecting the Board

4.1.3 Exporting board settings

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

- (1) Save the {ProjName}.scfg file.
- (2) Click on the [ (Export board setting)] button on the [Board] tabbed page.
- (3) Select the output location and specify a name for the file to be exported.

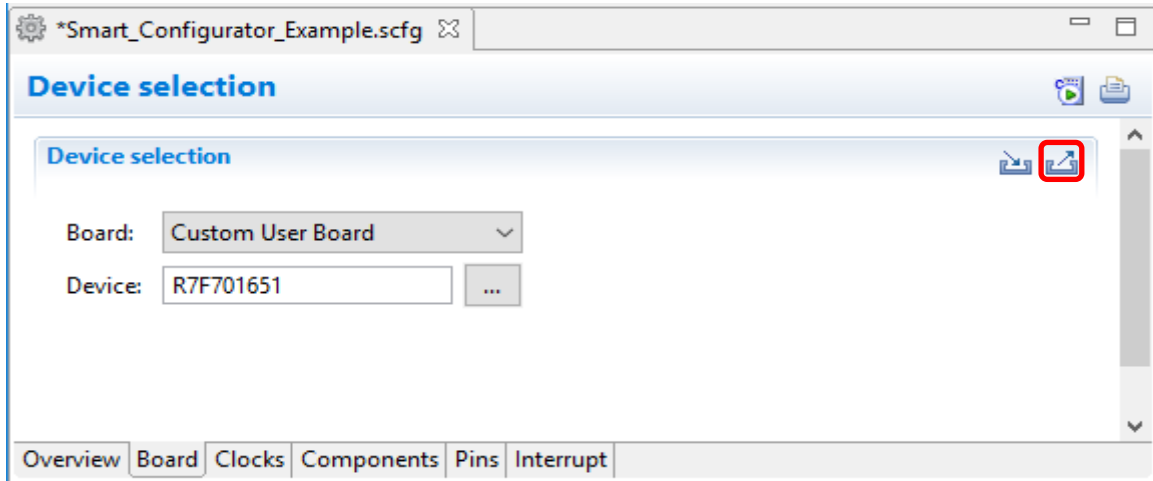



Figure 4-3 Exporting Board Settings (bdf Format)

4.1.4 Importing board settings

Follow the procedure below to import board settings.

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

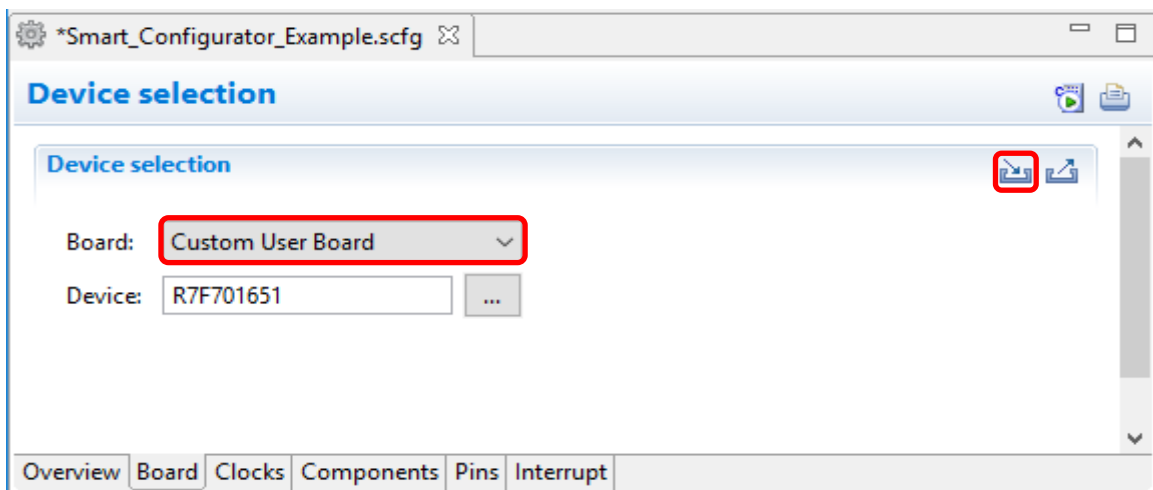


Figure 4-4 Importing Board Settings (bdf Format)

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.

Follow the procedure below to modify the clock settings.

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

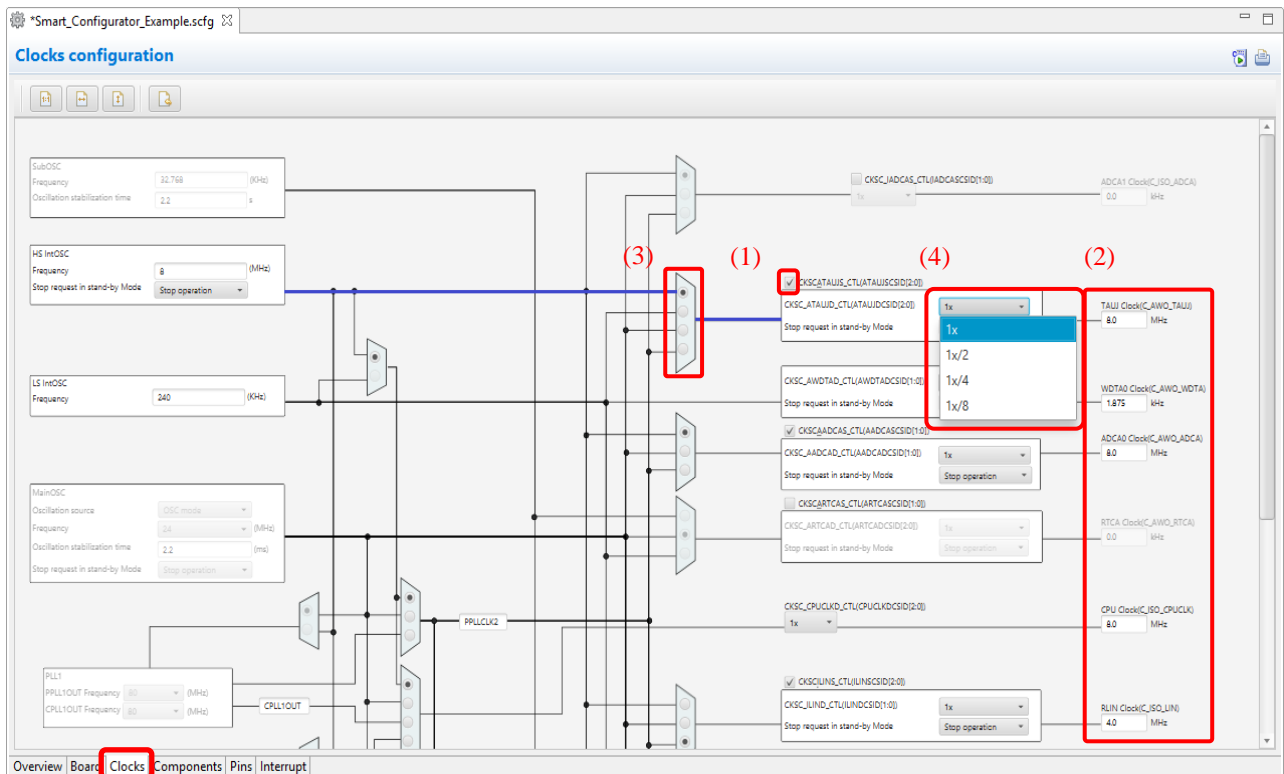


Figure 4-5 Clock Settings

4.3 Component Settings

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

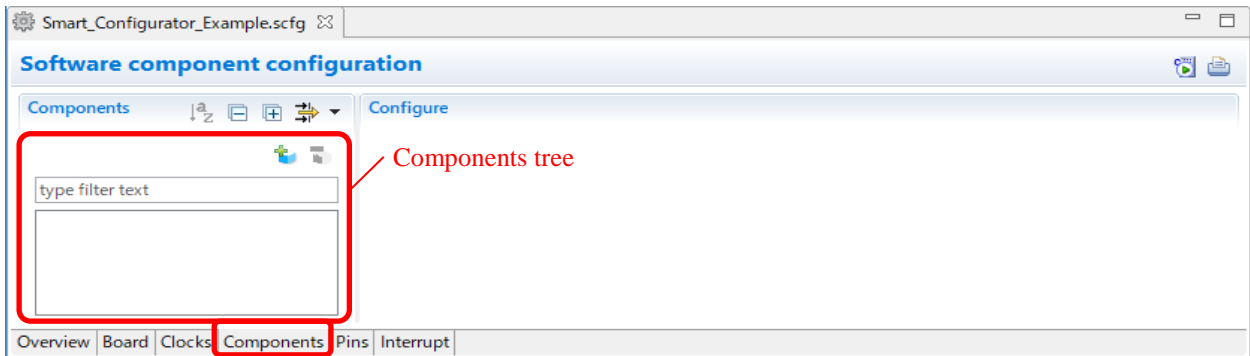


Figure 4-6 [Components] Page

4.3.1 Adding Code Generator components

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

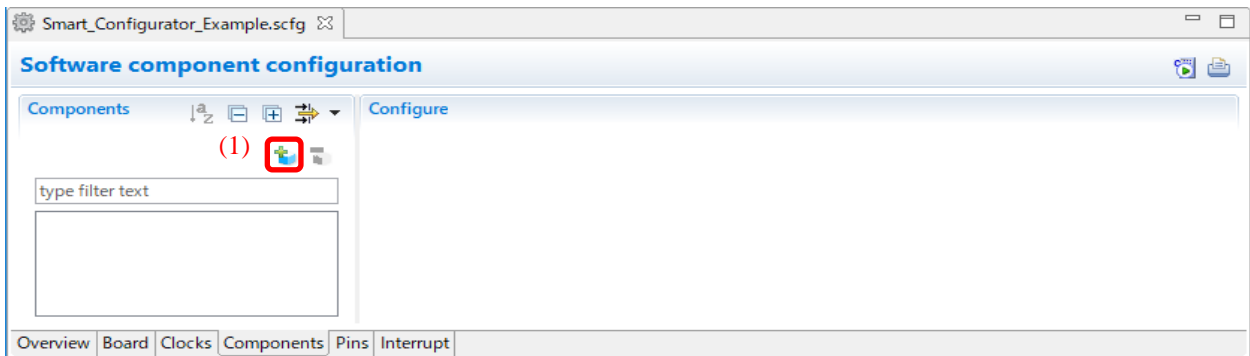


Figure 4-7 Adding a Component

- (2) Select a component from the list in the [New Component] dialog box (e.g. PWM Output Function).
- (3) Check that [Type] for the selected component is [Code Generator].
- (4) Click on [Next].

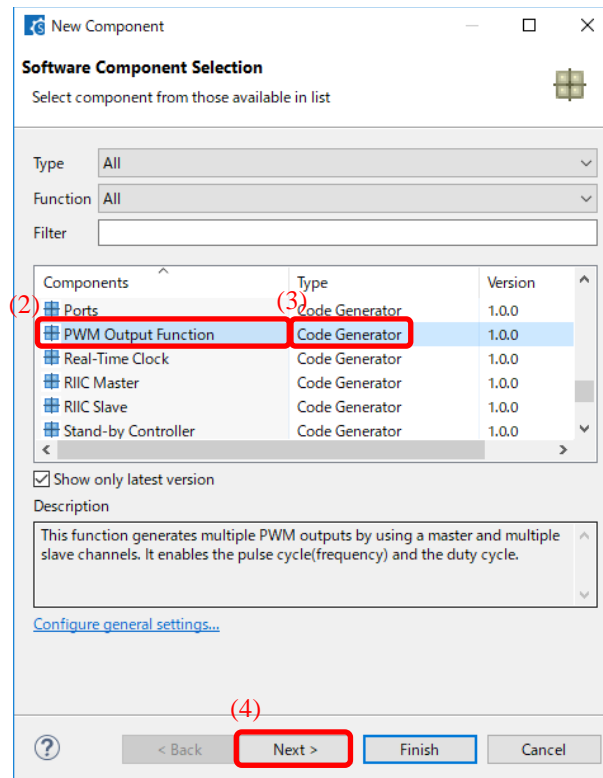


Figure 4-8 Adding a Code Generator Component

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

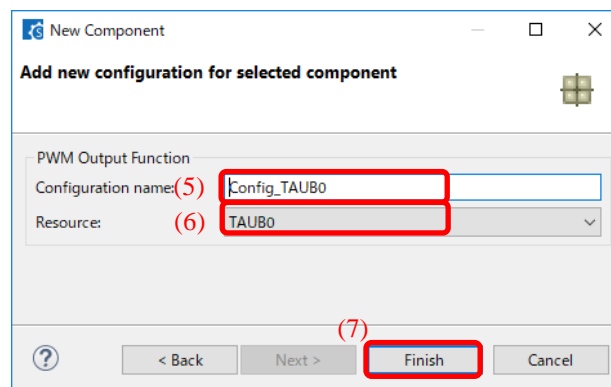


Figure 4-9 Adding a Component

4.3.2 Switching between the component view and hardware view

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

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

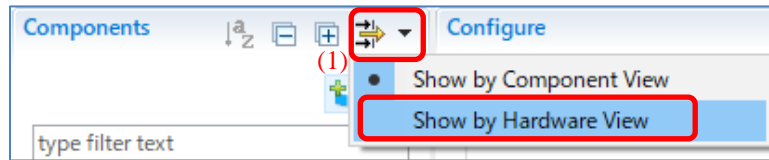


Figure 4-10 Switching to the Hardware View

- (2) Double-click on a hardware resource node (e.g. TAUB10 under Timer Array Unit B1) to open the [New Component] dialog box.
- (3) Select a component from the list (e.g. PWM Output Function) to add a new configuration as described in section 4.3.1.

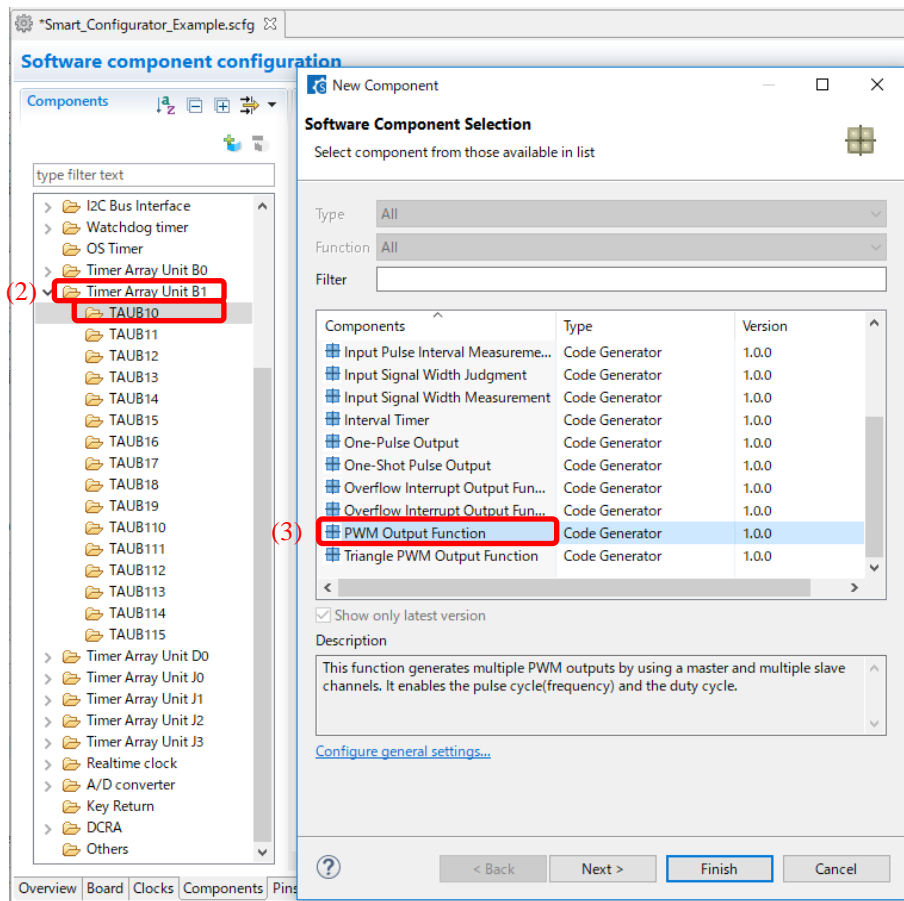



Figure 4-11 Adding a Component to the Hardware View

4.3.3 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.

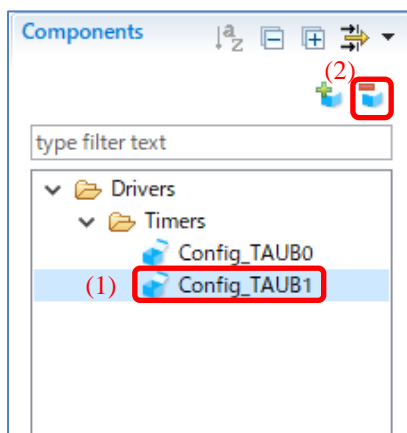


Figure 4-12 Removing a Software Component

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

This operation will also remove the source files generated for this component from the CS+ project tree.

4.3.4 Component configuration

Follow the procedure below to set up a components.

- (1) Select a configuration from the Components tree (e.g. Config_TAUB0).
- (2) Configure the driver in the [Configure] panel to the right of the Components tree. The following steps and figure show an example.
 - a. Select [PCLK/2] for [Clock source].
 - b. Select [Channel 1 slave], [Channel 2 slave], and [Channel 3 slave].
 - c. Specify [Pulse cycle] on the [Master0] tabbed page.
 - d. Specify [Duty] for each of the [Slave1], [Slave2], and [Slave3] tabbed pages.

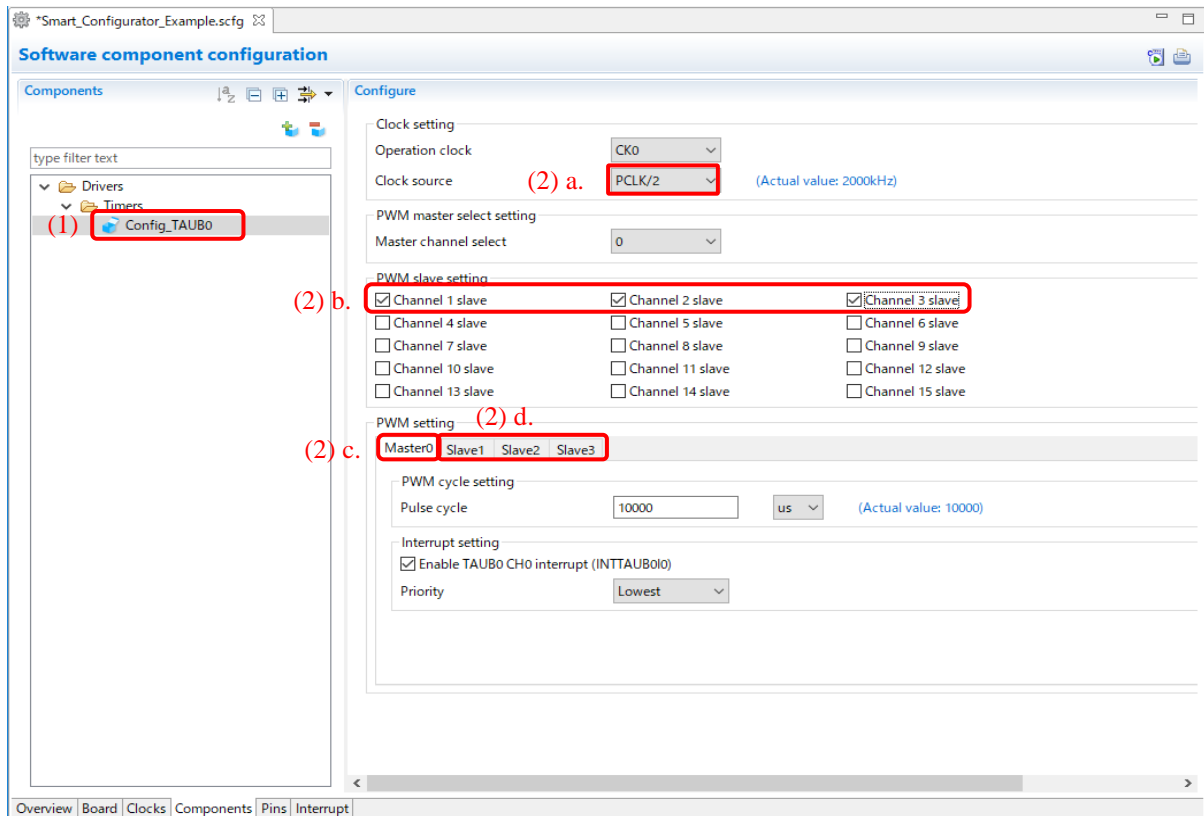






Figure 4-13 Component configuration

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

Right-clicking on a component and then selecting the  Generate code icon changes the icon to  Generate code and disables code generation for the component.

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

4.3.5 Changing the resource for a component

The Smart Configurator enables you to change the resource for a component (e.g. from TAUB0 to TAUB1). Compatible settings can be ported from the current resource to the new resource selected.

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

- (1) Right-click on a configuration (e.g. Config_TAUB0).
- (2) Select [Change resource] from the context menu.

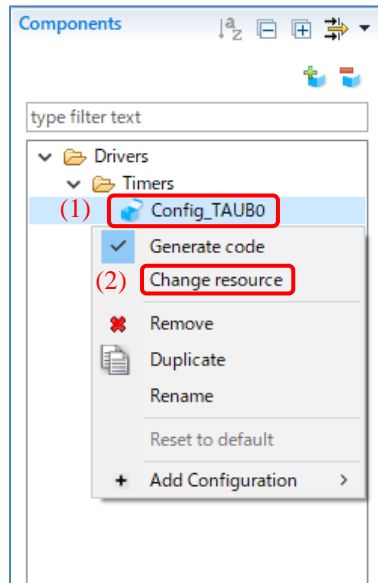


Figure 4-14 Changing the Resource

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

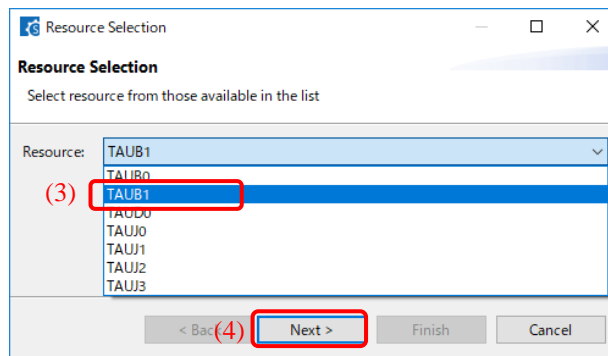


Figure 4-15 Components Page – Selecting a New Resource

- (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].

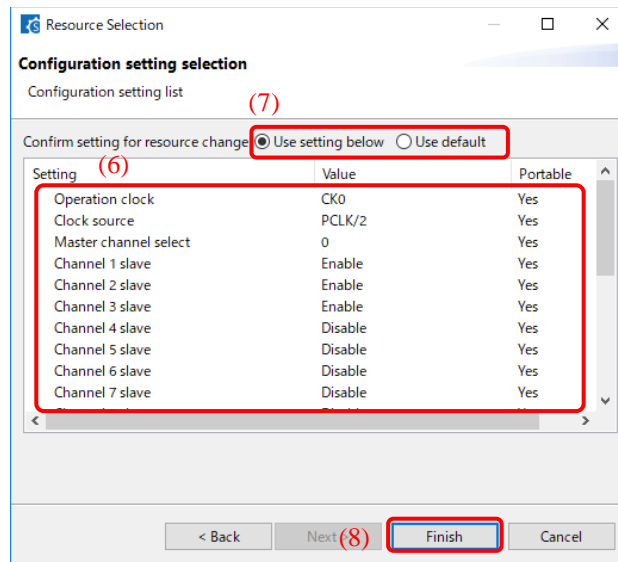


Figure 4-16 Checking the Settings of the New Resource

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

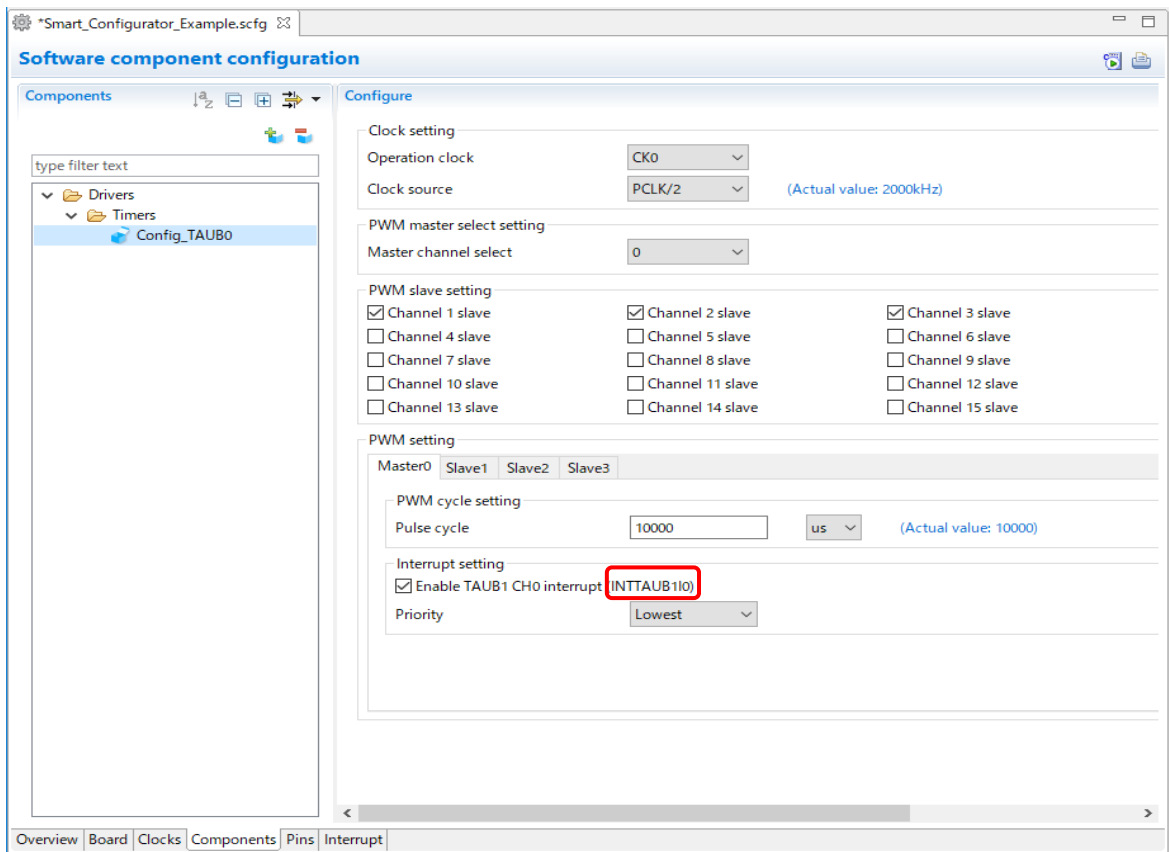


Figure 4-17 Resource Changed Automatically

(9) Right-click on the configuration.

(10) Select [Rename] to rename the configuration (e.g. change Config_TAUB0 to Config_TAUB1).

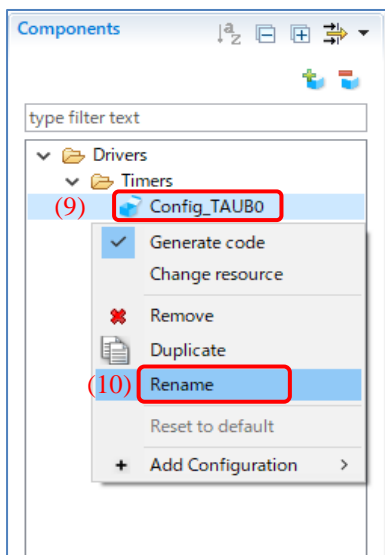


Figure 4-18 Renaming the Configuration

4.4 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.

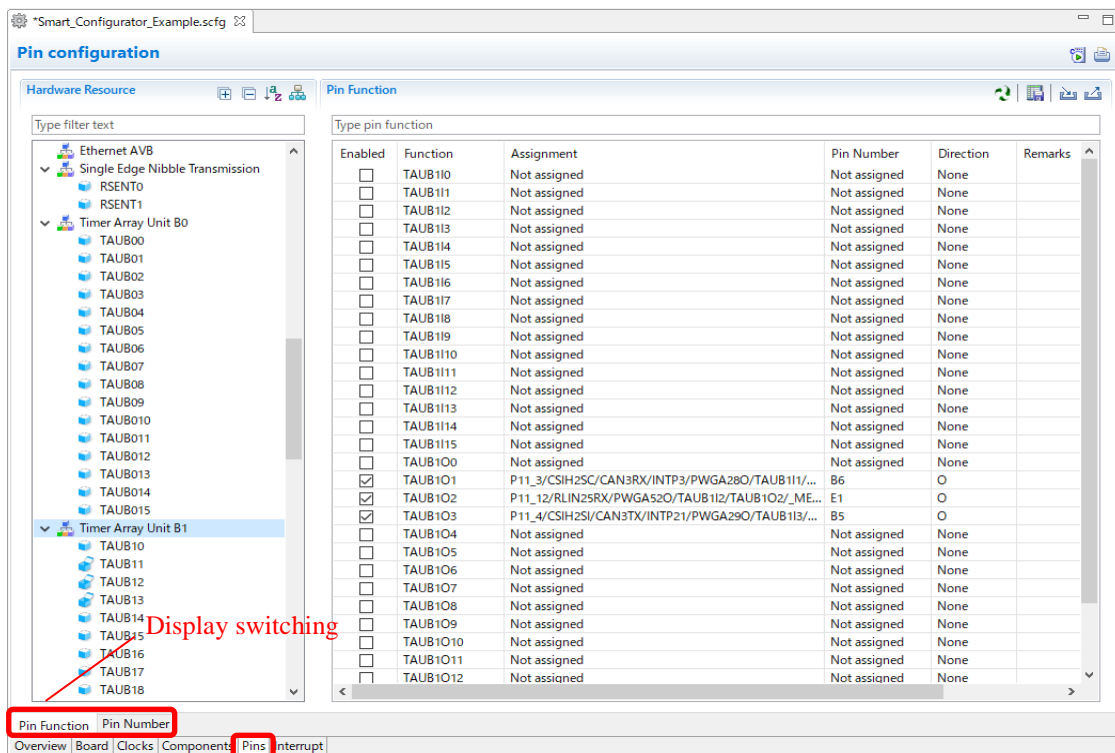


Figure 4-19 [Pins] Page ([Pin Function])

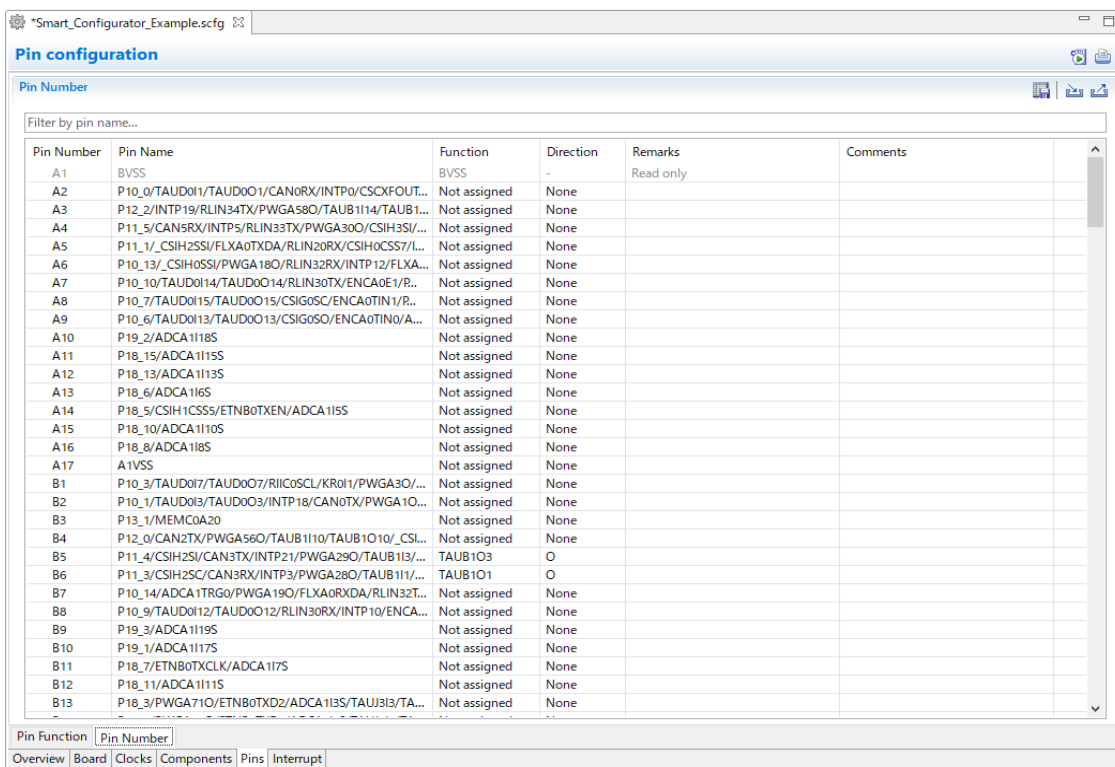




Figure 4-20 [Pins] Page ([Pin Number])

4.4.1 Changing the pin assignment of a software component

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

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

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

- (1) Click on  (Show by Hardware Resource or Software Components) to switch to the Hardware Resource view.
- (2) Select the target software component (e.g. Config_ICU).
- (3) Click the [Enabled] header to sort by pins used.
- (4) In the [Assignment] column on the [Pin Function] list, change the pin assignment (e.g. change from P10_0 to P0_1).
- (5) Assignment of a single pin or multiple pins that belong to the same peripheral channel can be changed by clicking on the  (Next group of pins for the selected resource)) button.

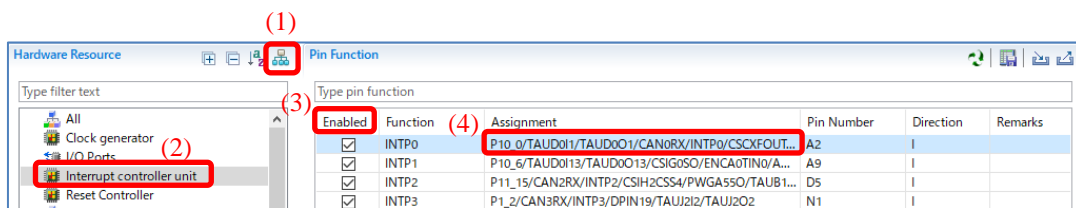



Figure 4-21 Pin Settings – Assigning Pins on the [Pin Function] List

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 current version of Smart Configurator does not support the component view for the [Pins] page. Use the Hardware Resource view to change the pin assignment.

4.4.2 Resolving pin conflicts

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

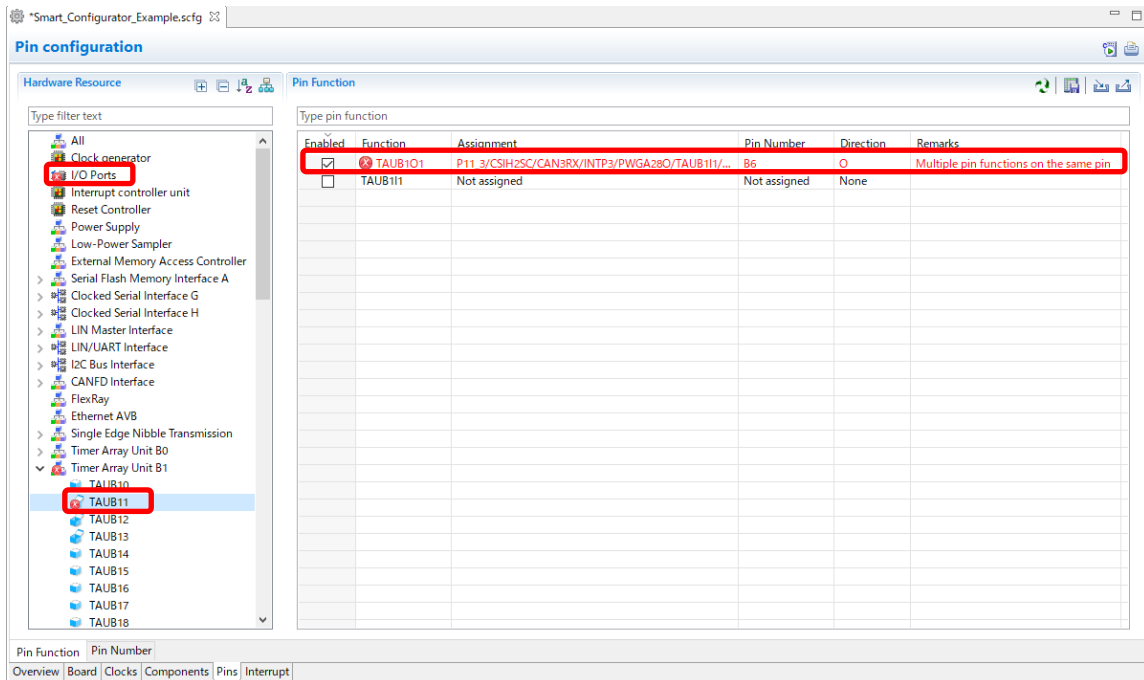


Figure 4-22 Pin Conflicts

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

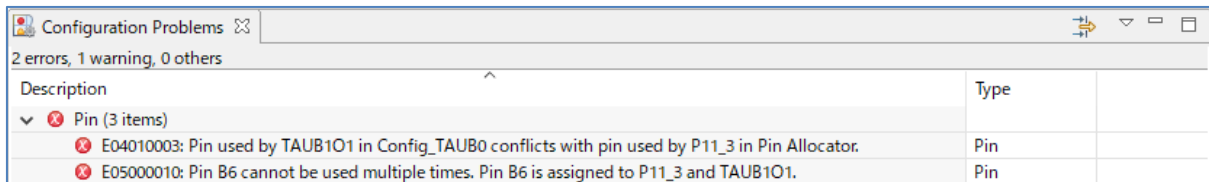


Figure 4-23 Pin Conflict Messages

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

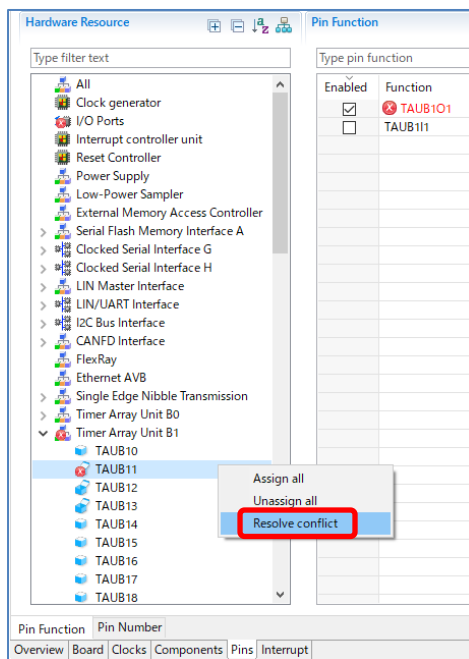



Figure 4-24 Resolving Pin Conflicts

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

4.4.3 Assigning pins using the MCU Package view

The Smart Configurator visualizes the pin assignment in the MCU Package view. You can save the MCU 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 Package view.

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

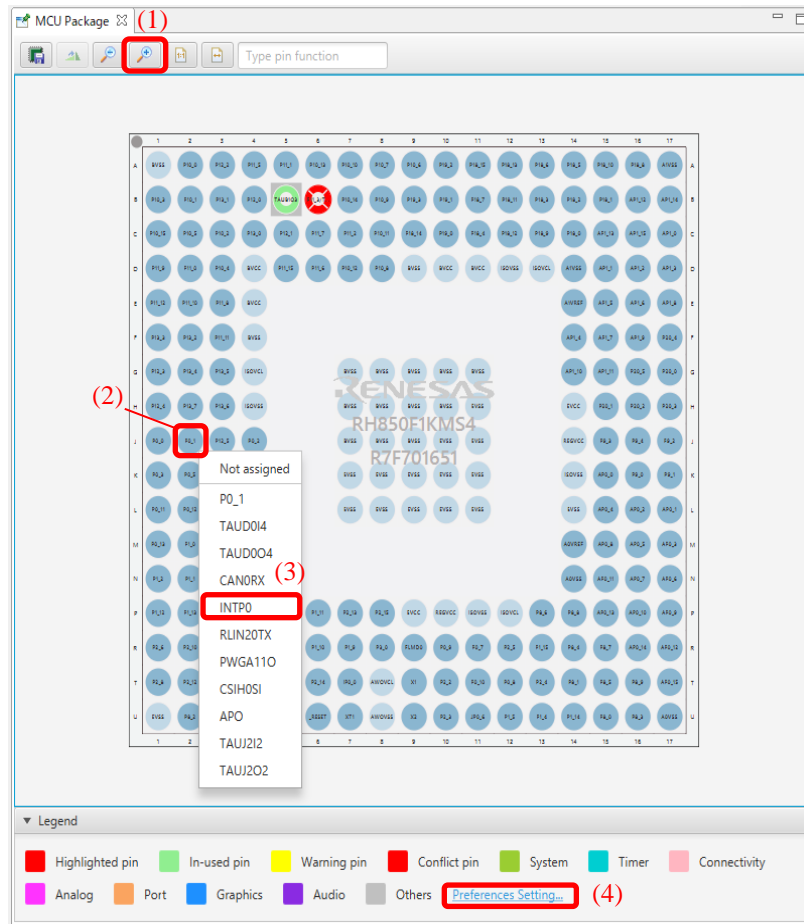



Figure 4-25 Assigning Pins Using the MCU Package View

4.4.4 Exporting pin settings

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

- (1) Save the {ProjName}.scfg file.
- (2) Click on the  (Export board setting) button on the [Pins] page.
- (3) 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.

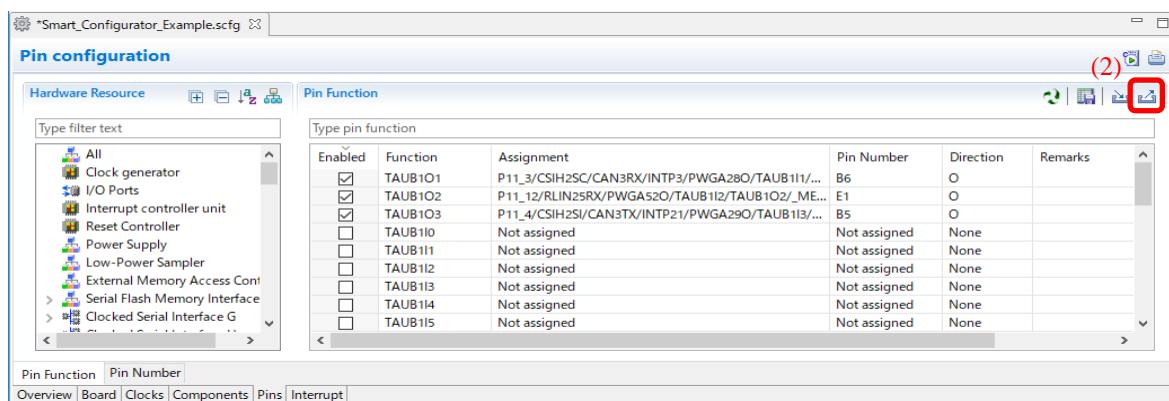




Figure 4-26 Exporting Pin Settings to an XML File

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

4.4.5 Importing pin settings

To import pin settings into the current project, click on the  (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.

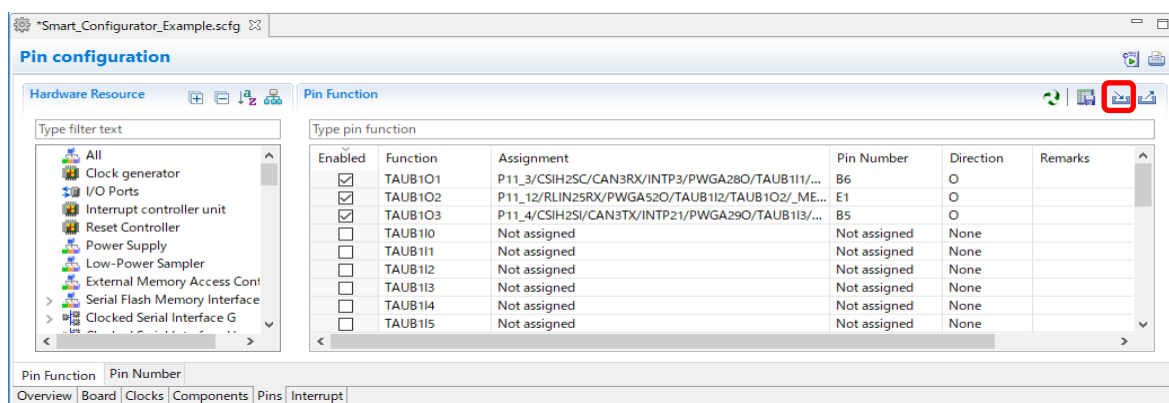


Figure 4-27 Importing Pin Settings from an XML File

4.5 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.

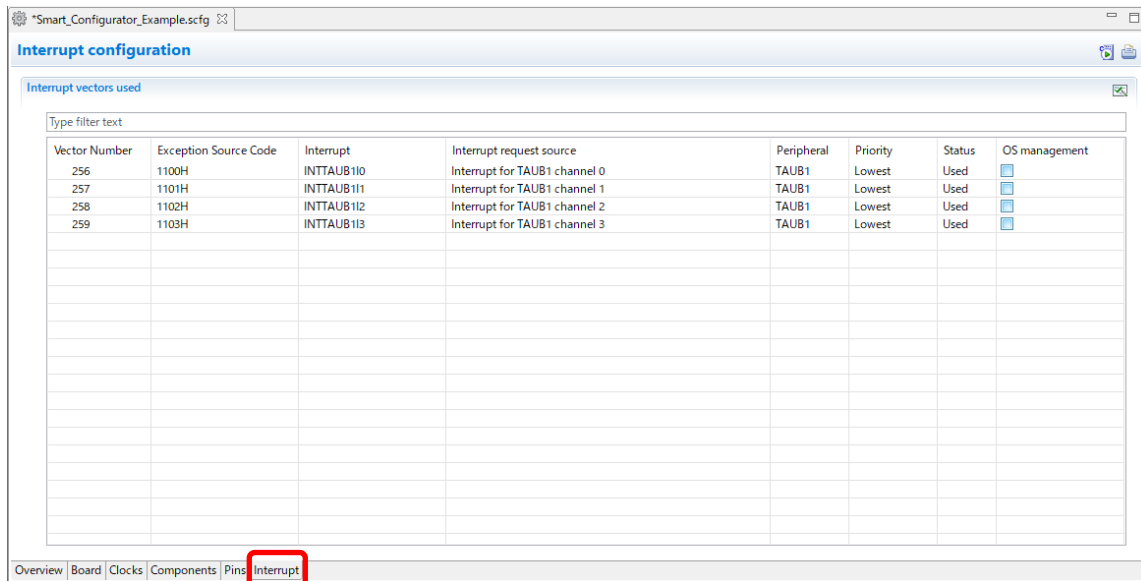


Figure 4-28 [Interrupt] Page

4.5.1 Changing the interrupt priority level and OS management setting

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

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

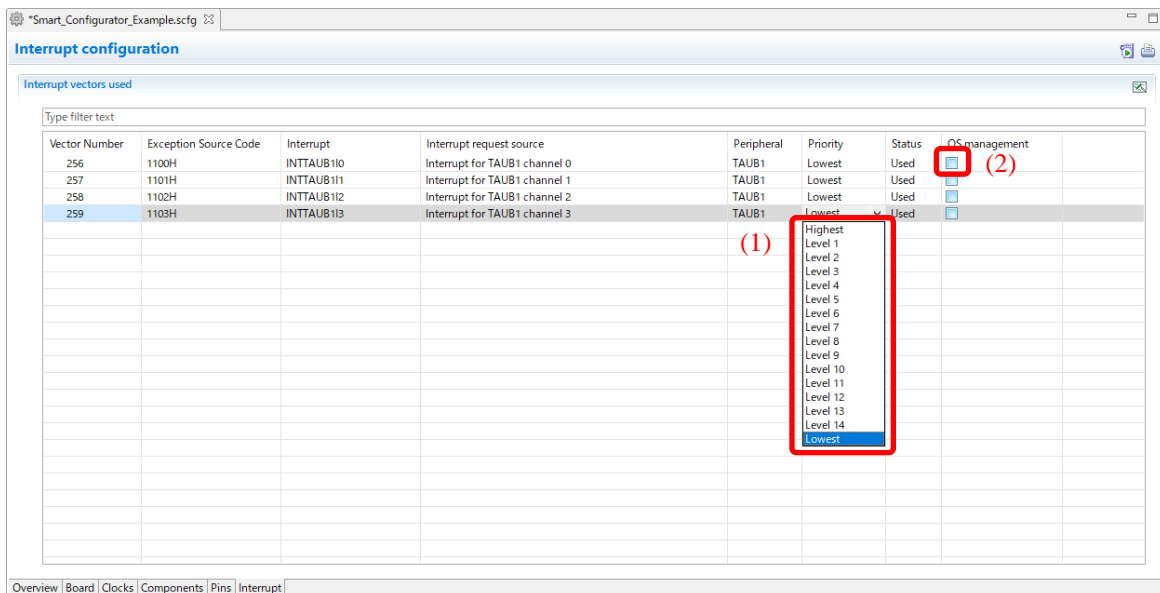



Figure 4-29 Interrupt Settings

5. Generating Source Code

5.1 Registering Generated Source Code with CS+

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

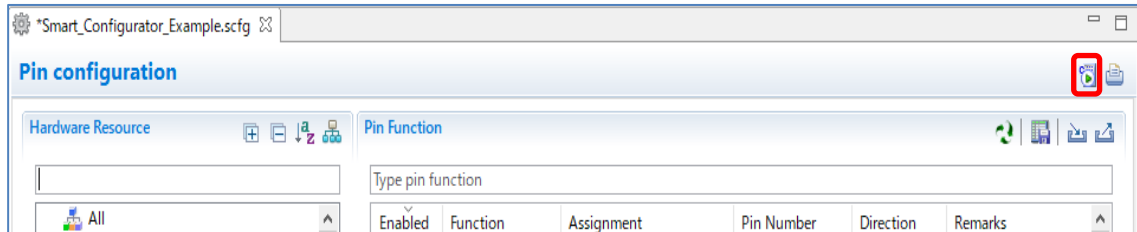


Figure 5-1 Generating a Source File

The Smart Configurator generates a source file in $\langle\text{ProjectDir}\rangle\backslash\text{src}\backslash\text{smc_gen}$, and the file is registered with the given project of CS+. If your Smart Configurator has already generated a file, a backup copy of that file is also generated (refer to chapter 8, Backing up Generated Source Code).

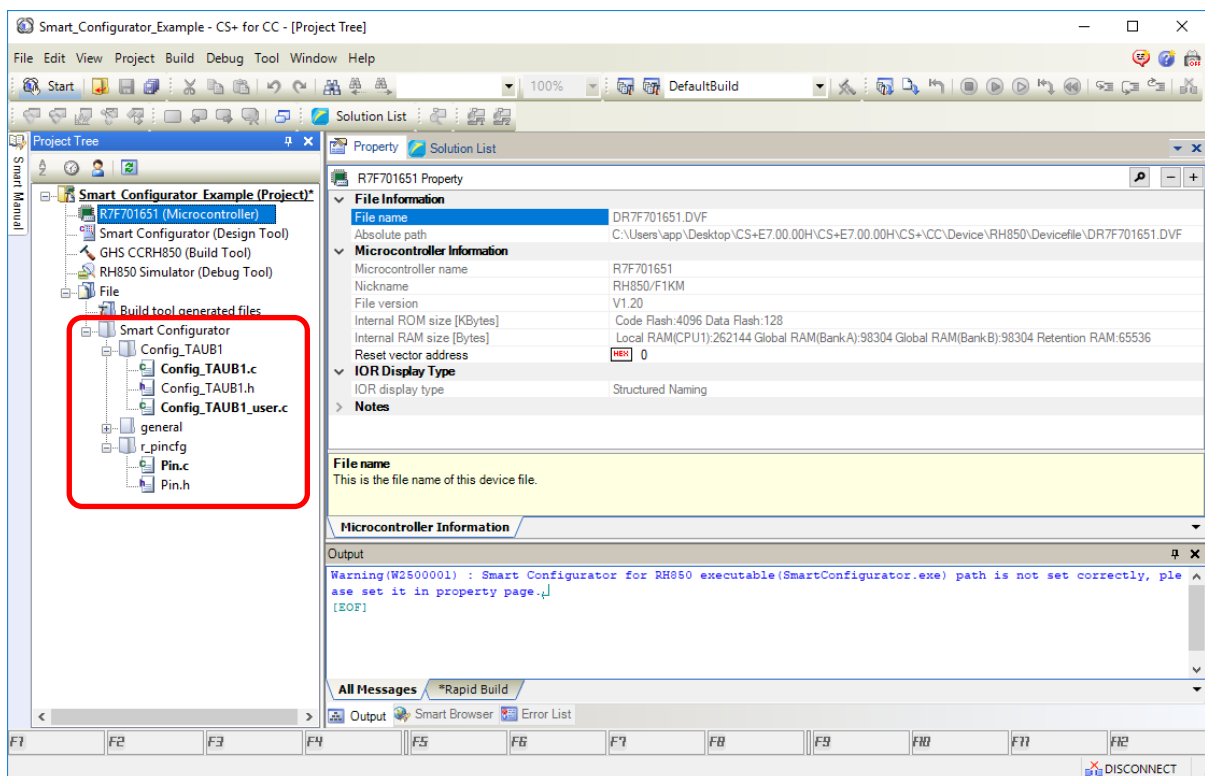


Figure 5-2 Registering a Source File with the CS+ Project

5.2 Configuration of Generated Files and File Names

Figure 5-3, Configuration of Generated Files and File Names, shows the folders and files output by the Smart Configurator. Function *main()* is included in *main.c*, which is generated when the project is created by CS+.

“ConfigName” indicates the name of the configuration formed by the component settings.

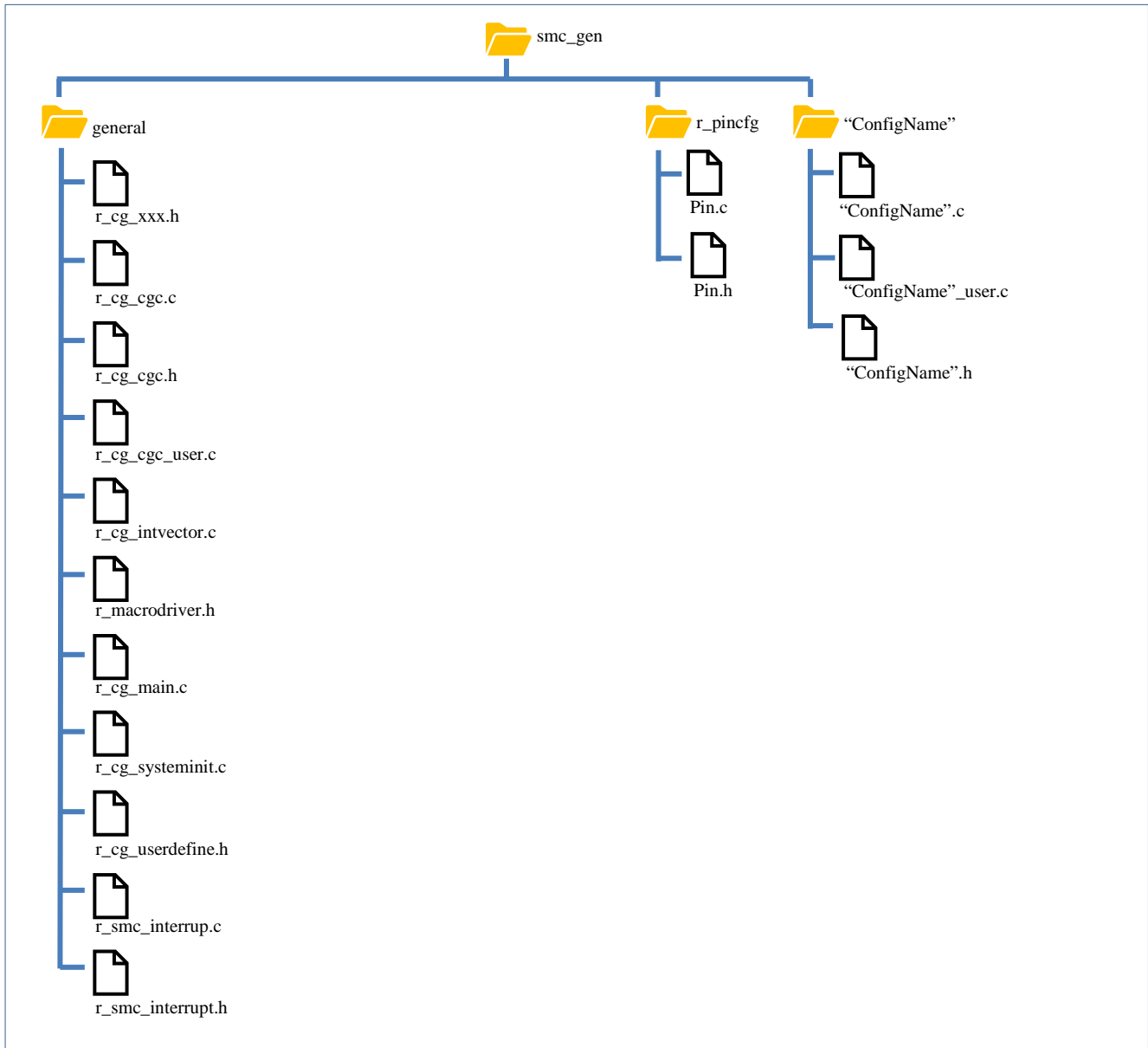


Figure 5-3 Configuration of Generated Files and File Names

| Folder | File | Description |
|---------------------|---------------------------------------|--|
| general | | This folder is always generated. It contains header files and source files commonly used by drivers of the same peripheral function. |
| | <i>r_cg_xxx.h</i> ^(Note*1) | These files are always generated. The files contain macro definitions for setting SFR registers. |
| | <i>r_cg_cgc.c</i> | This file is always generated. It contains the initialization of clock sources in accordance with the settings in the [Clocks] page. |
| | <i>r_cs_cgc.h</i> | This file is always generated. This header file contains macro definitions to initialize clocks. |
| | <i>r_cs_cgc_user.c</i> | This file contains functions to be added to <i>R_CGC_Create</i> . User can add codes and functions in the dedicated user code areas. |
| | <i>r_cg_intvector.c</i> | This file is always generated. It contains interrupt vector table definitions. |
| | <i>r_cg_macrodriver.h</i> | This file is always generated. This header file contains common macro definitions used in drivers. |
| | <i>r_cg_main.c</i> | This file is always generated. It defines the <i>main()</i> function. |
| | <i>r_cg_systeminit.c</i> | 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. |
| | <i>r_cg_userdefine.h</i> | This file is always generated. User can add macro definitions in the dedicated user code areas. |
| | <i>r_smc_interrupt.c</i> | This file is always generated. |
| | <i>r_smc_interrupt.h</i> | This file is always generated. |
| r_pincfg | <i>Pin.c</i> | This file is always generated. It is a reference of pin function initialization for all peripherals configured in the [Pins] tabbed page (except I/O Ports). |
| | <i>Pin.h</i> | This file is always generated. It contains the function prototypes of pin settings in <i>Pin.c</i> |
| {ConfigName} | | This folder is generated for the components that are added to the project. API functions in this folder are named after the <i>ConfigName</i> (configuration name). |
| | <i>{ConfigName}.c</i> | 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 (<i>R_ConfigName_Stop</i>). |
| | <i>{ConfigName}_user.c</i> | This file contains interrupt service routines and functions for user to add code after the driver initialization (<i>R_ConfigName_Create</i>). User can add codes and functions in the dedicated user code areas. |
| | <i>{ConfigName}.h</i> | This is header file for <i>{ConfigName}.c</i> and <i>{ConfigName}_user.c</i> |

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

5.3 Initializing Clocks

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

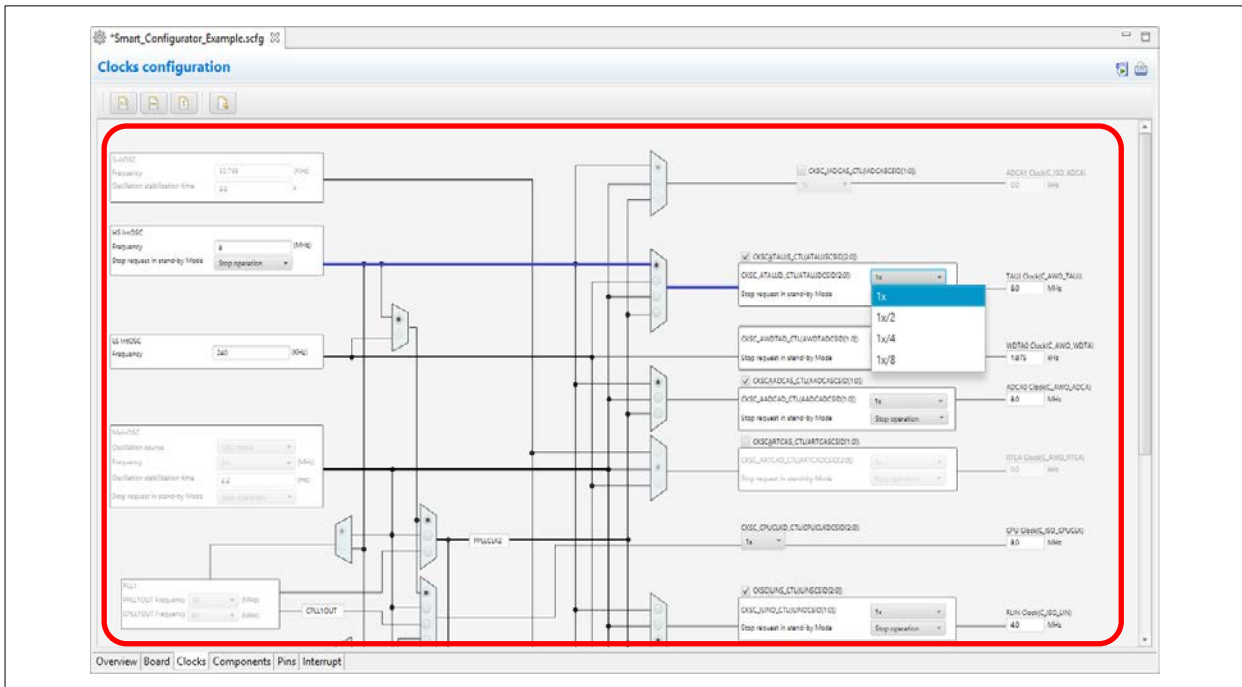


Figure 5-4 Clocks Configuration with Main Clock Selected as Clock Source

| No | Folder | File | Macros/Functions | Description |
|----------------|--------|------------------------------|------------------------------------|---|
| general | | <code>r_cg_cgc.c</code> | <code>R_CGC_Create</code> | This API function initializes clocks. <code>R_Systeminit</code> in <code>r_cg_systeminit.c</code> will call this function during execution of the <code>main()</code> function. |
| | | <code>r_cg_cgc.h</code> | Macros related to clocks | These macros are for clock initialization in <code>R_CGC_Create</code> . |
| | | <code>r_cg_cgc_user.c</code> | <code>R_CGC_Create_UserInit</code> | This API function is used to add code to <code>R_CGC_Create</code> after the CGC initialization. |

5.4 Initializing Pins

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

(1) Pin initialization for drivers with {ConfigName}

Pin functions are initialized in *R_ConfigName_Create* of this file $\%src\%smc_gen\%{ConfigName}\%{ConfigName}.c$.

Pin initialization codes will be handled in *main()*.

Note: The current version of Smart Configurator does not support the component view for the [Pins] page.

Figure 5-5 Pins Configuration for Config_TAUB1

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

(2) Reference to pin initialization codes

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

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

5.5 Initializing Interrupts

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

| Vector Number | Exception Source Code | Interrupt | Interrupt request source | Peripheral | Priority | Status | OS management |
|---------------|-----------------------|-----------|-------------------------------|------------|----------|--------|-------------------------------------|
| 142 | 108EH | INTTAUB0 | Interrupt for TAUB0 channel 0 | TAUB0 | Level 5 | Used | <input checked="" type="checkbox"/> |

(1) (2)

Figure 5-6 Interrupts Configuration in Interrupts View

| No | Item | Folder | File | Description |
|-----|---------------|--------------|---------------------|---|
| (1) | Priority | {ConfigName} | {ConfigName}.c | Interrupt priority level settings are initialized in <i>R_ConfigName_Create</i> in this file. <i>R_Systeminit</i> in <i>r_cg_systeminit.c</i> will call this function during execution of the <i>main()</i> function. |
| (2) | OS management | {ConfigName} | {ConfigName}_user.c | The interrupt functions defined in this file are output in the interrupt format that can be managed by the OS. |

6. Managing Conflicts

A user adding a component or configuring a pin or interrupt might cause problems in terms of resource conflicts. This information will be displayed in the Configuration Problems view. User can refer to the displayed information to fix the conflict issues.

6.1 Resource Conflicts

When two software components are configured to use the same resource (e.g. TAUB1), an error mark (❌) will be displayed in the Components tree.

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

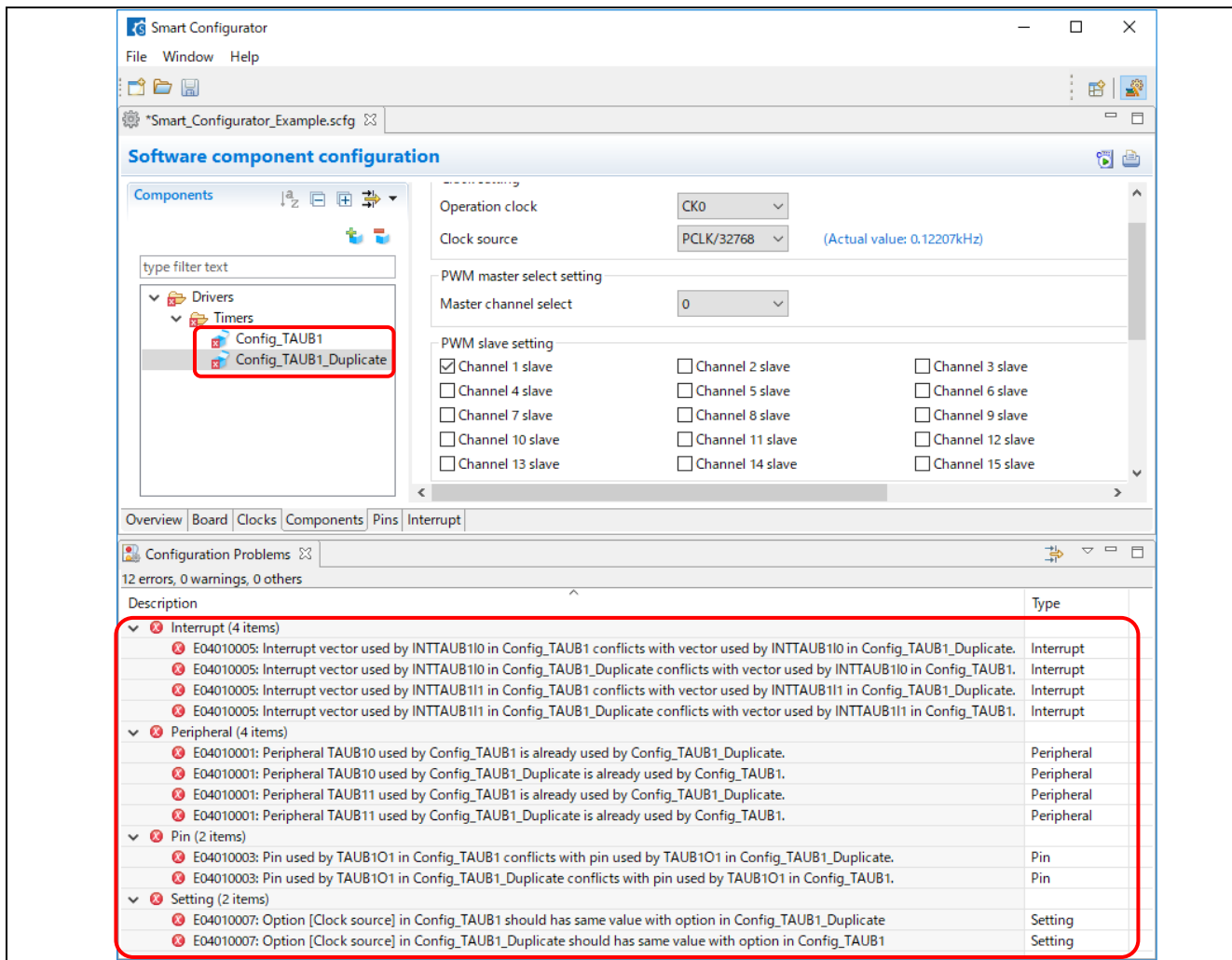


Figure 6-1 Resource Conflicts

7. Creating User Programs

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

7.1 Adding Custom Code in the Case of Code Generator

When [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 “TAUB1” for the PWM output function (resource TAUB1). The comments to indicate where to add custom code are at the start and end of each of the three files. Comments to indicate where to add user code are also added to the interrupt function for the peripheral module corresponding to Config.***_user.c. The following examples are for TAUB1 (Config_TAUB1_user.c).

```

/*****
Pragma directive
*****/
/* Start user code for pragma. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */

/*****
Includes
*****/
#include "r_cg_macrodriver.h"
#include "r_cg_userdefine.h"
#include "Config_TAUB1.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_TAUB1_Create_UserInit
* Description   : This function adds user code after initializing the TAUB1 channel
* Arguments     : None
* Return Value  : None
*****/


void R_Config_TAUB1_Create_UserInit(void)
{
    /* Start user code for user init. Do not edit comment generated here */
    /* End user code. Do not edit comment generated here */
}

```

```
/******  
* Function Name: r_Config_TAUB1_channel0_interrupt  
* Description   : This function is TAUB10 interrupt service routine  
* Arguments     : None  
* Return Value  : None  
*****/  
#pragma interrupt r_Config_TAUB1_channel0_interrupt(enable=false, channel=256, fpu=true,  
callt=false)  
void r_Config_TAUB1_channel0_interrupt(void)  
{  
    /* Start user code for r_Config_TAUB1_channel0_interrupt. Do not edit comment generated  
here */  
    /* End user code. Do not edit comment generated here */  
}  
  
/* Start user code for adding. Do not edit comment generated here */  
/* End user code. Do not edit comment generated here */
```

8. Backing up Generated Source Code

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


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

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

9. Generating Reports

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

9.1 Report on All Configurations (Text File)

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

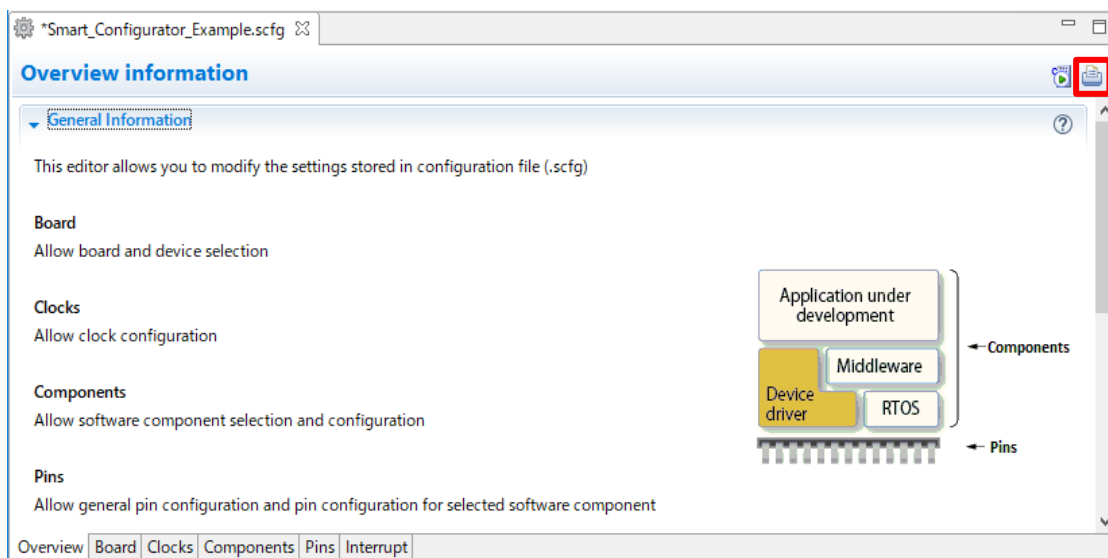


Figure 9-1 Output of a Report on the Configuration (as a Text File)

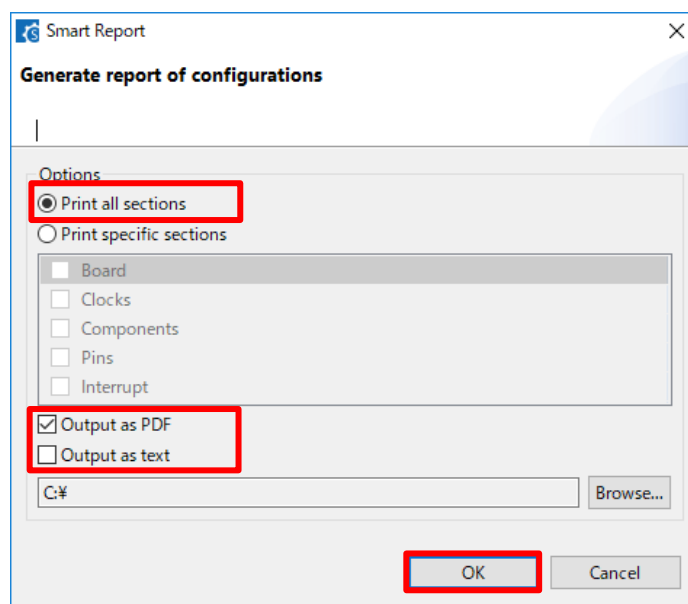



Figure 9-2 Dialog Box for Output of a Report

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.

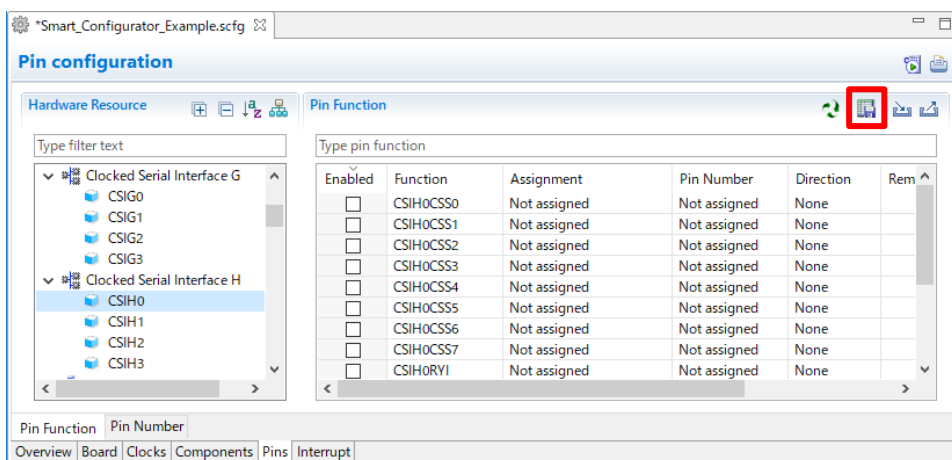



Figure 9-3 Output of a List of Pin Functions or Numbers (in csv Format)

9.3 Image of MCU Package (in png Format)

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

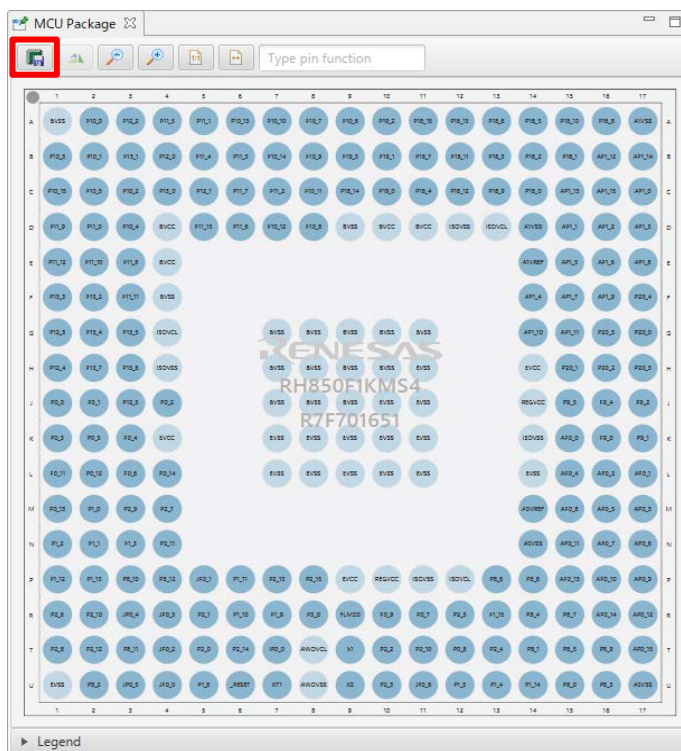


Figure 9-4 Outputting a Figure of MCU Package (in png Format)

10. Help

10.1 Help

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

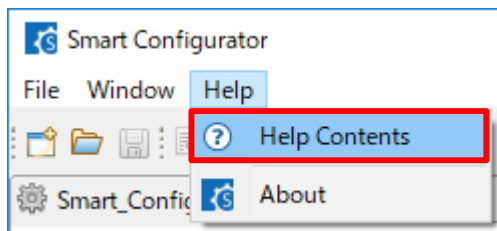


Figure 10-1 Help Menu

The help system can also be activated from the [Overview information] page.

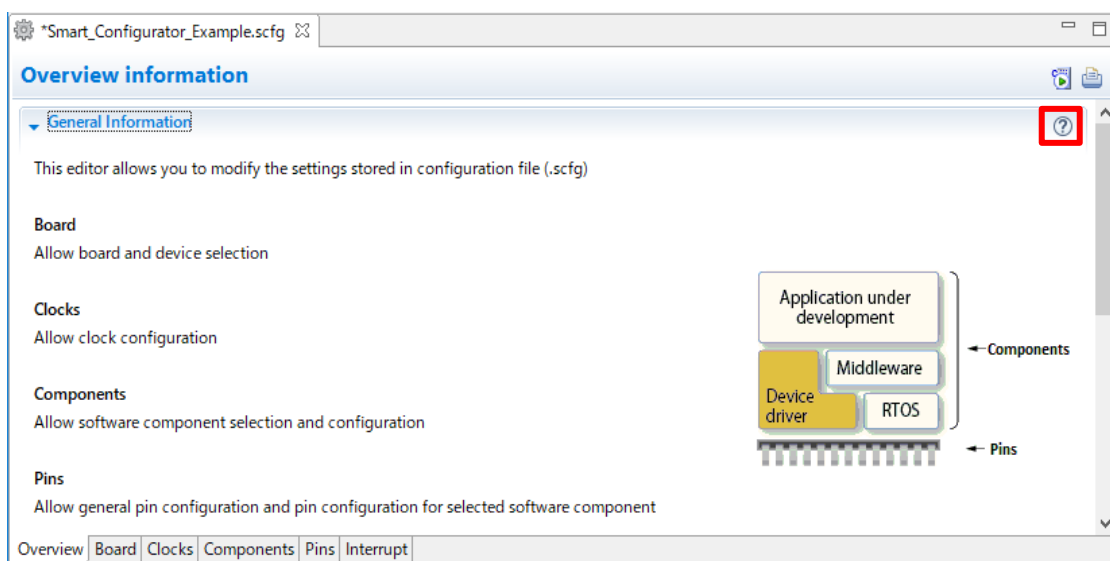


Figure 10-2 Quick Start

11. Documents for Reference

User's Manual: Hardware

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

Technical Update/Technical News

Obtain the latest information from the Renesas Electronics website.

User's Manual: Development Environment

CS+ V7.00.00 Integrated Development Environment User's Manual: Project Operation (R20UT4296)

CS+ V7.00.00 Integrated Development Environment User's Manual: RH850 Debug Tool (R20UT4299)

CS+ V7.00.00 Integrated Development Environment User's Manual: Message (R20UT4309)

CC-RH Compiler User's Manual (R20UT3516)

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

Website and Support

Renesas Electronics Website

<http://www.renesas.com/>

Inquiries

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

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

Revision History

| Rev. | Date | Description | |
|------|-------------|-------------|----------------------|
| | | Page | Summary |
| 1.00 | Nov 1, 2018 | - | First edition issued |

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

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

1. Handling of Unused Pins

Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

¾ The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

The state of the product is undefined at the moment when power is supplied.

¾ The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

¾ The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

¾ When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

¾ The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

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 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.
(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)



SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

Refer to "<http://www.renesas.com/>" for the latest and detailed information.

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

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338