Smart Configurator

Guide on Sample Projects for RH850/U2B Devices

Introduction

This document describes how to use the sample project of Smart Configurator for RH850/U2B devices in CS+ integrated development environment.

Target Device

RH850/U2B10 (BGA468, BGA373, BGA292)

RH850/U2B6 (BGA292)

Contents

1.	Overview2
1.1	Purpose2
1.2	Operating Environment2
2.	Outline of the Sample Projects
2.1	Introduction of Sample Project
2.2	Notes on the Sample Projects
3.	Basic Procedures for Operating the Smart Configurator4
4.	Description of the Sample Project5
4.1	Configuration of the Sample Project5
4.2	Basic Operating Procedure6
4.3	Procedure for Changing the Device10
4.4	Settings in the Sample Project
5.	Operations in the Smart Configurator15
5.1	Setting the Peripheral Modules (Software Components)15
5.2	Generating Drivers
5.3	Adding the Application Code to the User Code Area20
5.4	Adding the Application Code to main()21



1. Overview

1.1 Purpose

This document describes how to use the sample project of Smart Configurator for RH850/U2B 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 each integrated development environment 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

Туре	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/U2B 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/U2B sample project starts default CPU(PE0) after microcontroller has been reset.

The sample project takes "Interval Timer" as a sample and provides relative user code.

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).

- 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. 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 r-smc_intprg.c in the "smc/general" folder defines the vector table of EI maskable interrupt sources, which is the default EI vector table, is not set by the Smart Configurator.
- 5. 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/U2B 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".

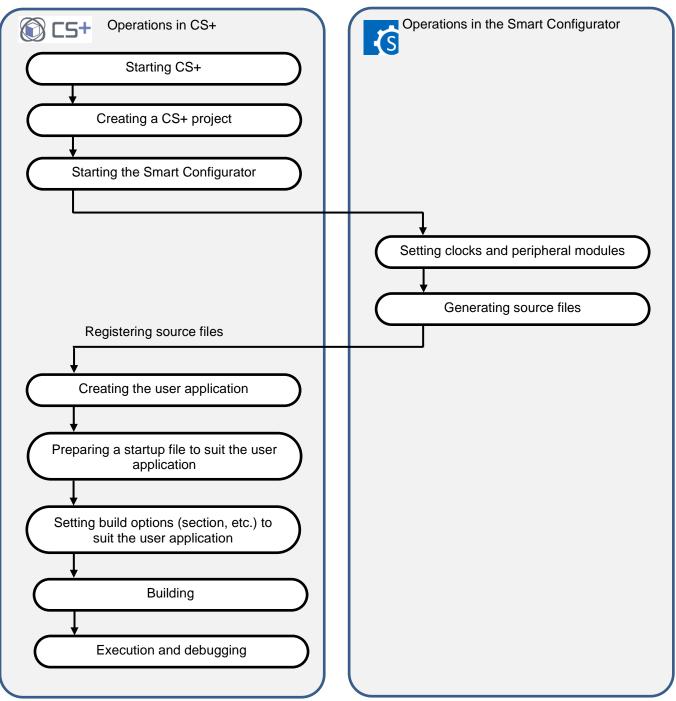


Figure 3-1. Basic Operating Procedure without using Sample Project



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:

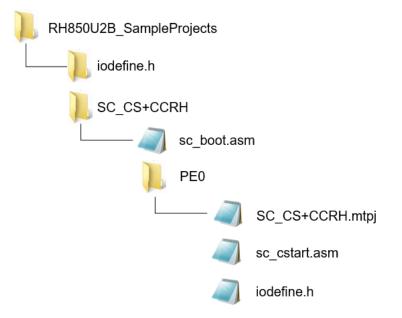


Figure 4-1. Project folder and files

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		
SC_CS+CCRH folder	Folder which includes all files belonging to CS+ project		
sc_boot.asm	Definition of the procedure from a reset to branching to the startup routine in sc_cstart.asm.		
PE0 folder	Folder which includes files belonging to default CPU(PE0)		
SC_CS+CCRH.mtpj	Project file for CS+		
sc_cstart.asm	Definition of the startup routine which is called by boot0.asm and is executed until branching to the main() function		
iodefine.h	Renesas CCRH header file that defines the registers for RH850/U2B devices		



4.2 Basic Operating Procedure

Figure 4-2 shows the operating procedure in CS+ environment when using the sample project.

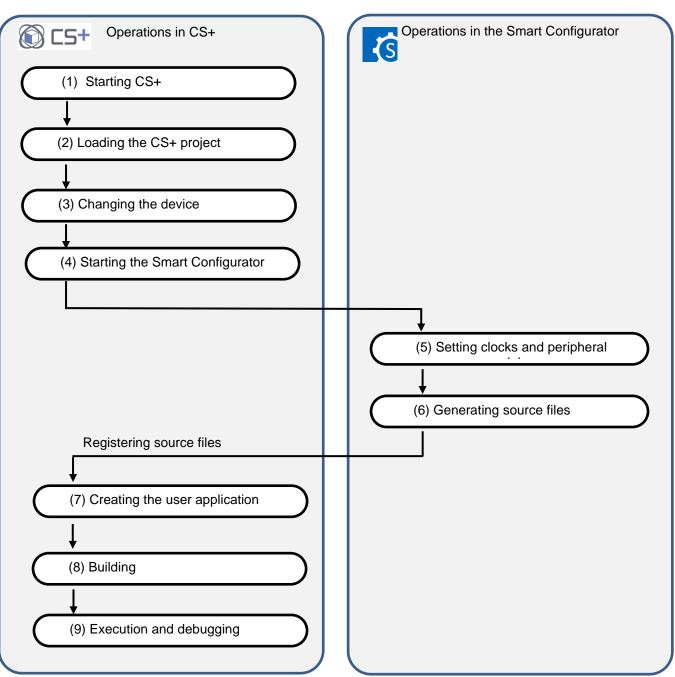


Figure 4-2. Operating Procedure for Sample Project



(1) Starting CS+

In the [Start] menu of Windows, select [Renesas Electronics CS+] \rightarrow [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 "SC_CS+CCRH.mtpj".

٢	CS+ for CC - [Start]										-	-		×
File	Edit View Project Build Debug Tool	Window	Help										3	â
	New	88 ÷			• 100%	- 6			× 50 0	h 🗈 🕞 (≥ HJ 🚳 93 C3 C3	a de		
	Open Ctrl+O	P : 6	1 44									_		
	Open with Encoding	Start												- x
	Add			The			nation on how to eff	- European CC						-
R	Close Project		GO	The u	utorial contai	ns the mon	nation on now to en	ectively use C.54	•					
J.	Close File		N											
ER.	Save Project Ctrl+Shift+S		Create N		DCT v proiect can	he created							-	
	Save Project As		GO				ated by reusing the	file configuration	n registered to an e	existing project.				
	Save Object Ctrl+S													
-	Save Object As		Create N	ew Multi	i-core Pi	roject -							+	
Ry.	Object Save Settings		Open Ex	oting Dr	ningt								-	
ø	Save All Ctrl+Shift+A		Open Ex			of CS+(form	erly CubeSuite+). C	an also be open	ed directly from th	e following link				1
	Page Setup				cent Projects				Favorite P	-				
-	Print Ctrl+P		GO		SC_CS+CCR	н			Nothing					
۵.	Print Preview		60		U2B6_Test U2B6_292									
	Recent Files				SC_CS+CCR	н								
	Recent Projects				_								_	
1	Delete the History of Files and Projects		Open Ex	sting MO	CU Simu	lator Or	nline/e² studio	/CubeSuite	e/High-perfo	ormance Emb	edded Workshop/F	PM+	-	Y
	Exit	put											ą	×
-		All Messa	aes /											•
			rowser 🔜 Or	itout										
F7	F2 Next Bookma F3 Find Previous		10	FS Stop	EG D	lisconnect fi		FR	Eg	FID	F77 Return Out	ER Ta	ig Jump	-
_	ns a file.									1.00				_

Figure 4-3. Loading the CS+ Sample Project

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

(3) Changing the device

The R7F70255x (RH850/U2B6) 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].

	- 🗆 ×
File Edit View Project Build Debug Tool Window Help	🧼 🥝 👘
🚳 Start 🛃 🔄 🕼 🖇 🔍 🔍 🙈 🧶 🔹 🔹 100% 🚽 🐻 DefaultBuild 💿 🔨 🖏 🛞 De 🐂 🛞 🚱 🗠	ça da Ma
💎 💎 💭 💯 🧛 📋 🗖 💭 📮 💭 🖉 Solution List 🛛 🖓 🖉	
🖓 Project Tree 🕴 🗶 🐨 Property	* 1
Smart Configurator Property	Q
R7F70255x (Microcontroller) Version V1.02.06.04 [16 Nov 2022] Smart configurator setting	
CC-RH (Build Tool) Smatt Configurator for RH850 executable file path C:\Program Files (x86)\Renesas Bectronics\SmattConfigurator	\RH850\eclipse\SmartConfigurator.exe
é-j∭ File	
Smart Configurator for RH850 executable file path	
Specify the Smart Configurator for RH850 executable file path.	
Specify the smart conligurator for Knoov executable file path.	
apecity we aman contigurator to innov executable tile part.	
Specify the singlet Computation for Prisod executable life pain.	
\ Smart Configurator Setting /	a .
	4 :
\ Smart Configurator Setting / Output	c \$
_Smart Configurator Setting / Dutput Detroiner went respectably: cer_cer_col	

Figure 4-4. 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/U2B10 Package Selection] pops up, please choose the chip user wants to use: Such as choose R7F70254*EABB -C(292pin) as following figure shows:

RH850/U2B10 Package Selection	\times
Please select RH850/U2B10 package details for Smart Configurator.	
R7F70254*EABB -C (292pin)	
O R7F70254*EABA -C (373pin)	
R7F70254*EABG -C (468pin)	
ОК	(

Figure 4-5. RH850/U2B10 Package Selection



- (5) Smart Configurator setting Clocks
- (6) Smart Configurator setting Components
- (7) Smart Configurator setting Generating code

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

💰 Smart Cor	figurator			- [1	\times
File Window	Help					
📩 🗁 🔡					8	ŝ
<pre> @ SC_CS+CCI </pre>	RH.scfg ×					
Overview	information		Generate Code	-	e Repo	ort
👻 General I	nformation				?	^
	Overview Get an <u>overview</u> of the features provided by Smart Configurator. Videos Introduction to Smart Configurator Browse related videos What's New Check out <u>what's new</u> in the latest release.	Application Code Software Components Middleware & Drivers Device Drivers MCU Hardware	Smart Configurator)		
✓ Current C	onfiguration					
	ard/device: R7F70255x (ROM size: 6 MB, RAM size: 576 KB, P ocation (PROJECT_LOC\): src\smc_gen	in count: 292) Edit				
	<u>^</u>	Configuration				U
Overview Boa	ard Clocks Components Pins Interrupt	Configuration				•
📮 Console 😕	: 📄 🚽 🔂 🖛 🖓 🖛 🗖	Configuration Problems ×		7	8	
No consoles te	o display at this time.	0 items				
		Description				
		4				>

Figure 4-6. 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.

(1) Select "R7F70255x (Microcontroller)" of main project "SC_CS+CCRH.mtpj" and then select [Change Microcontroller...] from the context menu.

Click on the [OK] button in the [Question] dialog box that appears.

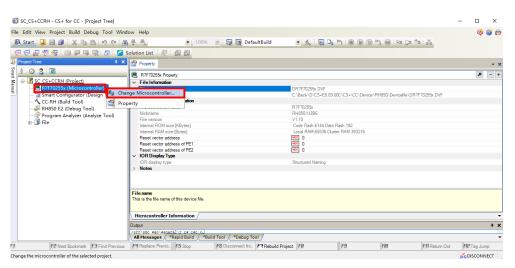


Figure 4-7. Change Microcontroller

Question((Q0232002) ×	
?	Requires saving the current project to enable to restore the current project status after changing the microcontroller. Continue saving and processing?	
		1

Figure 4-8. Save and continue

OK

Cancel

Help



(2) In the [Change Microcontroller] dialog box, select the RH850/U2B device to be used. Example: Changing from R7F70255x to R7F70254x

Change Microcontroller					
Change microcontroller to:					
🏭 (Search microcontroller)					
 RH850/U2B6 R7F70255x(XXXX) R7F702222(XXXX) RH850/U2B10 R7F702Z21(Performance) R7F702254x(Performance) R7F70254x(Performance) R7F70254x(FuSa) RH850/U2B24 	~	Internal F Code Fla Data Fla Internal F Local R/ Cluster F	Vame:R7F70254 IOM size[KBytes] sh:10240 IAM size[Bytes]: AM:65536 IAM:1048576 I Information:Nun	:-	s: 4
[0	К	Cancel		Help

Figure 4-9. Select device to be changed

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

SC CS+CCRH - CS+ for CC - [Project Tree]							_	- 🗆 X	
File Edit View Project Build Debug Tool	Monday, Hala							o 👔 😸	
🚳 Start 🔒 🖩 🕼 🛪 🐚 🖄 🕫	× 🔒 🎄 🐴	▼ 100%	🔐 😽 Default	Build	- 🔨 🐻	א ויי 🕫	D 14) 🚳 1	e Caça 👷	
	💋 Solution List 🕴 🖉 🕴 🚝	<i>G</i>							
Project Tree 🗛 🕽	× Property 💋 Solution List							→ ×	
Smart 2 @ 2 2	R7F70254x_P Property							+ – ۹	
	 File Information 							^	
R7F70254x_P (Microcontroller)	File name		DR7	F70254x_P.DVF					
Smart Configurator (Design Tool)	Absolute path				00C\CS+\CC\Devic	e\RH850\Devicefi	le\DR7F70254x F	P.DVF	
CC-RH (Build Tool)	 Microcontroller Information 	n							
RH850 E2 (Debug Tool)	Microcontroller name		R7F	70254x_P					
Program Analyzer (Analyze Tool)	Nickname		RH8	50/U2B10					
File	File version		V1.0						
E Plie	Internal ROM size [KBytes]	s] Local RAM:65536 Cluster RAM:1048576							
	Internal RAM size [Bytes]								
	Reset vector address		HEX						
	Reset vector address of PE		HEX						
	Reset vector address of PE		HEX	0					
	Reset vector address of PE	3	HEX	0				×	
	File name This is the file name of this devi							-	
	Output 👎 🗙								
	Information(M0291002) : The [EOF]	e device was change	d. Update the 1	meader file by	selecting [Genes	ate I/O header	لم.[file	^	
	All Messages							•	
<	> 🔝 Output 🌸 Smart Browser	🐮 Error List							
F7 Open Help f. F2 Rename F3 Find Next	F4 Replace Next F5 Go	FE Build & Dow F	7 Build Project	FB Ignore Brea	F9 Set/Delete	FHI Step Over	F77 Step In	FN2 Jump to Fun	
								× DISCOMMINGT	

Figure 4-10. Confirm microcontrollers are changed

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



4.4 Settings in the Sample Project

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

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

Settings in the sample project:

SC_CS+CCRH - CS+ for CC - [Property]			- 🗆 X
File Edit View Project Build Debug Tool Wind	ow Help		🧔 🔞
🚳 Start 🚚 🔚 🗿 🐰 🖻 🖄 🔊 연 🖓	🐴 🐴 🔹 100% 💌 📴 📴 DefaultBuild	- 🕺 🚳 🗗 🗝 🖲 🕞 🖬 🖷	ea ĉa ça 👷
💎 🖓 🖉 🤻 i 🗆 🗭 👒 🔍 🗗 i 🌌 s	olution List 🕴 🤄 🚑 🧱		
😳 Project Tree 4 🗙	Property		- x
S 2 3 2 3	CC-RH Property		+ – م <u>د</u>
E B C CS+CCRH (Project)	> System libraries	System libraries[0]	^
R7F70255x (Microcontroller)	Use standard libraries	Yes	
	Use "Standard/Mathematical Library" function	Yes	
CC-RH (Build Tool)	Check memory smashing on releasing memory	No	
RH850 E2 (Debug Tool)	Use "Non-local jump Library" function Output Code	No	
- Program Analyzer (Analyze Tool)	Specify execution start address	No	
🗈 - 🗿 File	Fill with padding data at the end of a section	No	
	Work around overrun fetch	No	
	Reserve prefetch area	No(No option specified)	
	Generate function list used for detecting illegal indirect function call	No	
	> List		
	 Section 		
	Section start address	RESET PEO, EIINTTBL PEO/%Reset VectorPE0%,	
	Section that outputs external defined symbols to the file	Section that outputs external defined symbols to the file[0]	
	> Section alignment	Section alignment[0]	
	> ROM to RAM mapped section	ROM to RAM mapped section[1]	~
	Section start address Specify the section start address. The sample value is set in this property by default. You need to set the appr		
	Common Options Compile Options Assemble Options Link	Options (Hex Output Options / I/O Header File Genera	ition Options 🗸 👻
	Output		4 X
	All Messages		-
F7 F2 Next Bookma_ F3 Find Previous F4	Replace Prev. FS Stop FS Disconnect fr. F7 Rebuild Proje.	F8 F9 F10 F	Return Out ReTag Jump

Figure 4-11. Link options setting

Section Settings			×
Address	Section	^	Add
%ResetVectorPE0%	RESET_PE0		Modify
	EIINTTBL_PE0		-
0x00010000	.const		New Overlay
	.INIT_DSEC.c		Remove
	.INIT_BSEC.c		Up Down
	.text.cmn		op Domi
	.text		
	.data		
0x00040000	EIINTTBL_PE		Import
0xFDE00000	.data.R		•
	.bss	~	Export
	ОК	Cancel	Help

Figure 4-12. Setting in the sample project



(5) [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 the sample project:

SC_CS+CCRH - CS+ for CC - [Property]		- 🗆 ×
File Edit View Project Build Debug Tool Wi	ndow Help	🤤 😨 🛱
🙉 Start 退 🔒 🗿 🐰 🖻 🖄 😕 🖉	😤 🚇 🛝 💽 100% 💌 💀 🐼 DefaultBuild	■ K 0, m 0,
중 중 🖉 😤 🤻 : 🗆 두 두 ରା 🍙 : 🖉	Solution List 🦷 🖉 🚰	
Project Tree 🕂 🗘	Property	• 1
2 @ 🙎 🔳	CC-RH Property	- a E
SC CS+CCRH (Project)* R7F70255x (Microcontroller)	V I/O header file	
R7F70255x (Microcontroller)	Update I/O header file on build	Yes(Checking the device file)
Smart Configurator (Design Tool)	Device file on generating I/O header file	DR7F70255x.DVF, V1.10
CC-RH (Build Tool)	Current device file	DR7F70255x.DVF, V1.10
RH850 E2 (Debug Tool)	Select modules which are output in files	No
Program Analyzer (Analyze Tool)	Output definitions regarding µITRON	No
File	Enable MISRA-C option	No
E- prie	Enable module array option	No
	Enable IOR array option	No
	Share definition of structure	Yes
	Output pragma directives for peripheral groups	No
	Update I/O header file on build Selects whether to update the I/O header file on build. Updates the I/O header file if the device file is mever than the device file w	hen the I/O header file was generated or the property related to I/O header file generation was cha.
	Common Options / Compile Options / Assemble Options / Link	Options / Hex Output Options / I/O Header File Generation Options
	Output TBOPT All Messages	4 :
Open Help fo F2 Rename F3 Find Next	FY Replace Next F5 Go F6 Build & Down F7 Build Project	FB Ignore Break FB Set/Delete Br FB Step Over FT Step In FB Jump to Fund
		A DISCONNECT

Figure 4-13. Setting in the sample project

(6) [Property] panel from [RH850 E2 (Debug Tool)] \rightarrow [Connection settings] tab.

Settings in the sample project: SC CS+CCRH - CS+ for CC - [Project Ti × File Edit View Project Build Debug Tool Window Help 🥥 🕜 🚔 🔹 100% 👻 🐻 🐻 DefaultBuild - | 🔬 : 🔂 🗅, H) | 🕘 🔊 🔿 H), 🛞 | 92 ÇI ČI | 🔏 🚳 Start 🛃 🖩 🗿 🗄 🏷 🖻 🛍 🤌 여 여 😤 🌲 🐴 💎 🖓 🜆 🐬 🍕 : 💼 💭 🗣 🔍 | 🎜 : 💋 Solution List 🕴 🦉 : 🚝 🏭 🕈 🗙 🚰 Property Project Tree 2 🕜 🙎 🗃 Smar RH850 E2 Property م SC CS+CCRH (Project)* Clock Manua R7F70255x (Microcontroller) Mount main clock on target board Yes 20.00 CPU0 - 10.00, CPU1 - 10.00, CPU2 - 10.00 Main clock frequency [MHz] CPU clock frequency [MHz] CC-RH (Build Tool) Connection with Emul AH850 E2 (Debug Tool) Emulator serial No Program Analyzer (Analyze Tool) 🗄 🗍 File Connection with Target Board Communications method LPD LPD clock frequency [kHz] Defau Power target from the emulator.(MAX 200mA) Interface for supplying the power Yes 5.0 Supply voltage [V] Initialize RAM when connecting Release the RESET before disconnecting from the target syste Debug the initial stop state and the standby mode Debug the GTM function No No Debug target Select the debug target Connect Settings Debug Tool Settings / Download File Settings / Hook Transaction Settings × Outpu All Messages / FB Build & Down... F? Build Project FB Ignore Break... F9 Set/Delete Br... FN Step Over F7 Open Help fo... F2 Rename F4 Replace Next F5 Go F3 Find Next F77 Step In FH2 Jump to Func. **DISCONNECT**

Figure 4-14. Connect Settings in the sample project (1)



Start Image: Solution List Image: Solutio	SC_CS+CCRH - CS+ for CC - [Project Tree] File Edit View Project Build Debug Tool Wir	tow Help	× □ -
Image: Source of the second secon	🚳 Stort 🔒 🗟 🗿 🐰 🖻 🖄 🔊 연구	4 중 초, · · · · · · · · · · · · · · · · · ·	
Field SC 53 - CCFH (Rerject)	Project Tree 4 >	Property 1	- × ×
Image: Sec: Sec: Sec: Sec: Sec: Sec: Sec: Se			+ - ۹
Use authentication di Ivo	R7F70255x (Microcontroller) Smart Configurator (Design Tool) CC-RH (Build Tool) R1R850 E2 (Debug Tool) Program Analyzer (Analyze Tool)	Use the E2 expansion interface No use • Flash • Interfield (Constraint) • Costomer ID Customer ID • Customer ID Customer ID • Desta Flash ID Data Flash ID • Destore specific ID SetS Programmer ID, C-TEST ID, ID0, ID1, ID2 SetSVR parameter No Using the code flash elf programming No Change the colx to flash writing Yes Mag mode Single mag mode Vok RAM start address © FCOGF000 A.06	
		Select the debug target. Connect Settings Debug Tool Settings / Download File Settings / Hook Transaction Settings /	
		Output	ų x
Connect Settings Debug Tool Settings / Download File Settings / Hook Transaction Settings / Tool Settings / To		TEOP	
Connect Settings Debug Tool Settings / Download File Settings / Hook Transaction Settings / Output 9 X Tappi	F7 Open Help fo F2 Rename F3 Find Next	The Project FB Set/Delete Br., F	FI2 Jump to Func.
Connect Settings Debug Tool Settings Download File Settings Hook Transaction Settings Image: Connect Settings<			

Figure 4-15. Connect Settings in the sample project (2)



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.

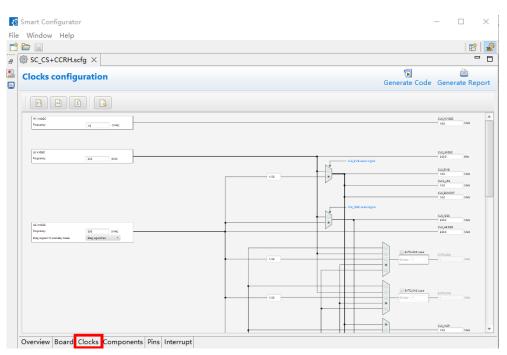


Figure 5-1. Configure clock



(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.

Smart Configurator					_		\times
File Window Help							
📬 🗁 🔚						: 🗈	<u> </u>
₽ 🕸 SC_CS+CCRH.scfg ×						-	- 0
Software component configu	iration			Generate Code	Gener	ate Rej	port
Components 🚵 🛃 🎝 🕞 🕀 I	Configure						
S							
type filter text							
	<						>
Overview Board Clocks Component	s Pins Interrup	t					

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.

🔥 New Co	mponent			\Box \times		
oftware Component Selection						
Select com	ponent from those available ir	n list				
Category	All			~		
Function	All			~		
Filter						
Compone	ents	Version		•		
H Input I	Pulse Interval Judgment	1.5.0				
🗄 Input I	Pulse Interval Measurement	1.5.0				
🗄 Interru	upt Controller	1.4.0				
🗄 Interva	al Timer	1.5.0				
🖶 MSPI I	Master	1.3.0				
🖶 MSPI 🤋	Slave	1.3.0				
🖶 One-P	ulse Output	1.5.0				
🖶 One-S	hot Pulse Output	1.5.0				
🖶 Overfl	ow Interrupt Output (Width	1.4.0				
🖶 Ports		1.5.0		v		
Show o	nly latest version					
Descriptio	-					
	val Timer Function is timer that pt is generated, resulting in a		ot at regular interva	ls.When \land		
	1			V		
<u>Configure</u>	<u>ceneral settings</u>					
?	< [Back Next >	Finish	Cancel		

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.

Rew Component	or selected component	
		-
Interval Timer		
Configuration name:	Config_TAUD0_0	
Resource:	TAUD0_0	~
?	< Back Next > Finish	Cancel

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.

🔇 Sr	nart Configurator				- 🗆 ×
File	Window Help				
1					E 🛛 🖉
	*SC_CS+CCRH.scfg ×				- D
S	oftware component configur	ation		G	enerate Code Generate Report
(Components 🚵 🛃 🎘 🕞 🕀 🚔	Configure			
	te 😨	Clock setting			
	type filter text	Operation clock	СК0 ~		
	 ✓ ⇒ Drivers ∧ ★ Timers ♦ 	Clock source	PCLK/32768 ~	Baud rate 1	(Actual value: 0.610352kHz)
	Config_TAUD0_0	Interval timer setting			
		Interval time	10000	μs	330.4)
		Generates INTTAUD010	when counting is started		
		Interrupt setting			
		Enable TAUD0 channel 0	interrupt (INTTAUD010)		
		Priority	Lowest \checkmark		
	×	<			>
0	veniew Board Clocks Components	Pine Interrupt			

Figure 5-5. Set component configuration

(6) Repeat steps (2) to (5) for each component that you intend to use.



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.

₹ Ŝ	Smart Configurator			- 0	\times
File	Window Help				
					} 🔏
8	#SC_CS+CCRH.scfg ×				
3	Software component configura	tion		Generate Code Generate Re	port
	Components 🚵 🛃 🎘 🕀 🕀 🛱	Configure			
	te T	Clock setting			
	type filter text	Operation clock CK0) ~		
	✓ ➢ Drivers ^ ✓ ➢ Timers	Clock source PCI	.K/32768 V Baud rate	1 (Actual value: 0.610352k	Hz)
	Config_TAUD0_0	Interval timer setting			
		Interval time 100	μs ~ (Ac	tual value: 9830.4)	
		Generates INTTAUD010 when count	ng is started		
		Interrupt setting			
		Enable TAUD0 channel 0 interrupt (I	NTTAUD010)		
		Priority Lov	vest \checkmark		
	~	<			>
	Overview Board Clocks Components	ins Interrupt			

Figure 5-6. Generating code



5.3 Adding the Application 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 application code (e.g. code for interrupt processing) to the user code areas as necessary.

Example: File generated for the interval timer component

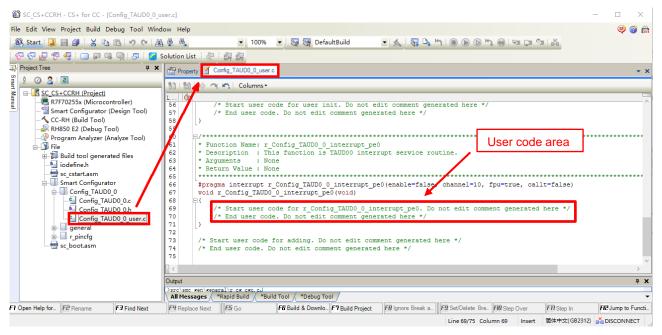


Figure 5-7. Add user code to the user code area of interrupt routine



5.4 Adding the Application Code to main()

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 application code to the user code area.

Example: Add application code to the main() function.

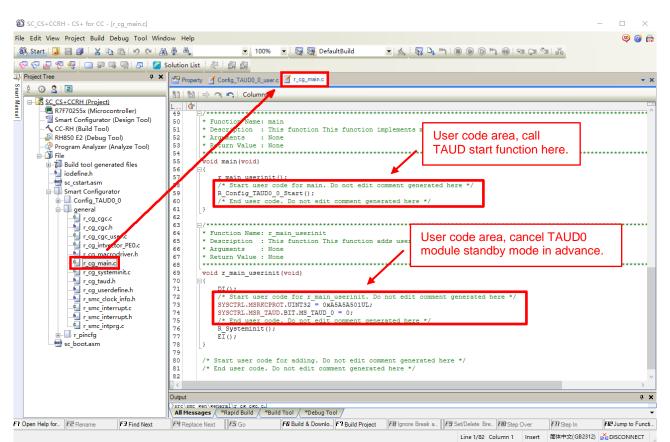


Figure 5-8. Add user code to main function



Revision History

Rev.	Section	Description
1.00	All	New version



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

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

1. Precaution against Electrostatic Discharge (ESD)

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

2. Processing at power-on

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

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

6. Voltage application waveform at input pin

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

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

Notice

- Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
- 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

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

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

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

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