

## RL78 Family

## Renesas Flash Driver RL78 Type 01 SC version (Code Flash)

### Introduction

This document explains Renesas Flash Driver RL78 Type 01 for the RL78/G2x group in the case of using Smart Configurator(SC). It is a process which builds into a user program the functions for "the code flash memory driver" included in RFD RL78 Type 01, and it is a method for programming code flash memory using the attached sample program.

In this document, "Renesas Flash Driver RL78 Type 01" is described to be "RFD RL78 Type 01" or "RFD".

This document includes the contents in comparison with conventional RFD RL78 Type 01 not using SC. This document distinguishes and expresses it.

Simple version : Conventional RFD RL78 Type 01 not using SC. SC version : RFD RL78 Type 01 using "SC" currently explained by this document.

### **Target Device**

The target device group by which the operation for RFD RL78 Type 01 was confirmed.

RL78/G22 group RL78/G23 group RL78/G24 group

If this application note is applied to other microcomputers, it is necessary to modify in accordance with the specification of the microcomputer. And, be sure to evaluate enough.



## Contents

1. Specification	3
1.1 Operating Environment	3
1.2 Structure of Sample Program Folders	4
1.3 File Structure of RFD Driver	
1.3.1 File Structure of RFD Common Driver (r_rfd_rl78_common)	5
1.3.2 File Structure for RFD Code Flash Driver (r_rfd_rl78_codeflash)	6
1.4 Code Flash Programming Processing Using a Sample Program	7
2. Creating a Sample Project for Code Flash Reprogramming	8
2.1 Example of Creating a Sample Project	8
2.1.1 In Case of CS+	8
2.1.2 In Case of e <sup>2</sup> studio(CC-RL)	8
2.1.3 In Case of IAR EW for Renesas RL78	8
2.1.4 In Case of e <sup>2</sup> studio(LLVM)	9
2.2 Example of Source Code Registration	10
2.2.1 In Case of CS+	10
2.2.1.1 Notes of case used at the same time in r_rfd_rl78_codeflash and r_rfd_rl78_extraarea	13
2.2.2 In Case of e <sup>2</sup> studio(CC-RL)	14
2.2.2.1 Notes of case used at the same time in r_rfd_rl78_codeflash and r_rfd_rl78_extraarea	17
2.2.3 In Case of IAR EW for Renesas RL78	18
2.2.3.1 Notes of case used at the same time in r_rfd_rl78_codeflash and r_rfd_rl78_extraarea	23
2.2.4 In Case of e <sup>2</sup> studio(LLVM)	24
2.2.4.1 Notes of case used at the same time in r_rfd_rl78_codeflash and r_rfd_rl78_extraarea	27
2.3 Project registration of sample program	28
2.4 The Check of Operation for Sample Program	42
2.4.1 In Case of CS+	42
2.4.2 In Case of e <sup>2</sup> studio(CC-RL)	43
2.4.3 In Case of IAR EW for Renesas RL78	44
2.4.4 In Case of e <sup>2</sup> studio(LLVM)	45
3. Precautions	46
4. Reference document	47
5. Revision History	48



### 1. Specification

The sample program included in RFD RL78 Type 01 erases the block 14 (0x00007000) of a code flash area, and programs 64 bytes of data from the top of the block 14.

### 1.1 Operating Environment

- C Compiler Packages

#### Table 1-1 The target C Compiler Packages

Package	Manufacturer	Version
CC-RL(for CS+ or e <sup>2</sup> studio)	Renesas Electronics	V1.10 or later
IAR (Embedded Workbench)	IAR Systems	V4.21 or later
LLVM(e <sup>2</sup> studio)	Open source software	V10.0.0.202312 or later

## IAR Systems, IAR Embedded Workbench, IAR, and the logotype of IAR Systems are trademarks or registered trademarks owned by IAR Systems AB.

- Emulator

Table 1-2 shows the emulator on which the operation of RFD RL78 Type 01 was confirmed. **Table 1-2 Emulator on which RFD RL78 Type 01 Operation was Confirmed** 

Emulator	Manufacturer
E2 emulator Lite	Renesas Electronics

Target MCU

RL78/G22

RL78/G23

RL78/G24



### 1.2 Structure of Sample Program Folders

Figure 1.1 shows the structure of sample program folders.

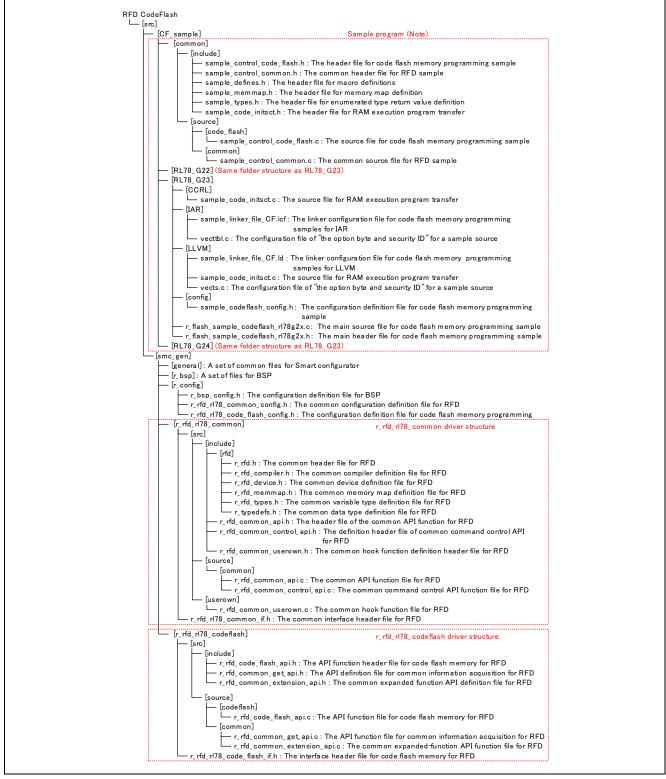


Figure 1.1 Structure of Sample Program Folders

Note: The sample program of a zip file format downloads by Smart configurator. Extract the compressed file (CF\_sample.zip) outputted to the "demo" folder, and move the [CF\_sample] folder under the [src] folder. Refer to "2.3 Project registration of sample program" for the details of project registration.



### **1.3 File Structure of RFD Driver**

#### 1.3.1 File Structure of RFD Common Driver (r\_rfd\_rl78\_common)

The difference of SC version RFD common driver and Simple version common driver is shown.Refer to the "RL78 Family Renesas Flash Driver RL78 Type 01 user's manual (R20UT4830) " for the detail specification for the RFD common driver.

## Table 1-3 Difference of the files of the SC version RFD and the simple version RFD (Common API : r\_rfd\_rl78\_common\src\source\common)

File name	Simple version	SC version
r_rfd_common_api.c	No chang	je
r_rfd_common_control_api.c	No chang	ge

# Table 1-4 Difference of the files of the SC version RFD and the simple version RFD (Common API : r\_rfd\_rl78\_common\src\userown)

File name	Simple version	SC version
r_rfd_common_userown.c	No chang	je

## Table 1-5 Difference of the files of the SC version RFD and the simple version RFD (Common header : r\_rfd\_rl78\_common\src\include)

File name	Simple version	SC version
r_rfd_common_api.h	No chang	je
r_rfd_common_control_api.h	No chang	je
r_rfd_common_userown.h	No chang	je

# Table 1-6 Difference of the files of the SC version RFD and the simple version RFD (Common header : r\_rfd\_rl78\_common\src\include\rfd)

File name	Simple version	SC version
r_rfd.h	No chang	e
r_rfd_compiler.h	No chang	e
r_rfd_device.h	No chang	e
r_rfd_memmap.h	No chang	e
r_rfd_types.h	No chang	e
r_typedefs.h	No change	

# Table 1-7 Difference of the files of the SC version RFD and the simple version RFD (Common interface header : r\_rfd\_rl78\_common)

File name	Simple version	SC version
r_rfd_rl78_common_if.h	-	Newly created.
		Include the header file for common API.



#### 1.3.2 File Structure for RFD Code Flash Driver (r\_rfd\_rl78\_codeflash)

The difference of SC version RFD code flash driver and Simple version code flash driver is shown. Refer to the "RL78 Family Renesas Flash Driver RL78 Type 01 user's manual (R20UT4830) " for the detail specification for the RFD code flash driver.

## Table 1-8 Difference of the files of the SC version RFD and the simple version RFD (Code flash API : r\_rfd\_rl78\_codeflash\src\source\codeflash)

File name	Simple version	SC version
r_rfd_code_flash_api.c	No chang	je

## Table 1-9 Difference of the files of the SC version RFD and the simple version RFD (Code flash API : r\_rfd\_rl78\_codeflash\src\source\common)

File name	Simple version	SC version
r_rfd_common_get_api.c	No chang	je
r_rfd_common_extension_api.c	No chang	je

## Table 1-10 Difference of the files of the SC version RFD and the simple version RFD (Code flash API header : r\_rfd\_rl78\_codeflash\src\include)

File name	Simple version	SC version
r_rfd_code_flash_api.h	No chang	je
r_rfd_common_get_api.h	No chang	je
r_rfd_common_extension_api.h	No chang	je

# Table 1-11 Difference of the files of the SC version RFD and the simple version RFD (Code flash interface header : r\_rfd\_rl78\_codeflash)

File name	Simple version	SC version
r_rfd_rl78_code_flash_if.h	-	Newly created.
		Include the header file for code flash API.



### 1.4 Code Flash Programming Processing Using a Sample Program

Figure 1.2 shows the flow chart of the sample program. The sample\_codeflash\_main function copies the "program executed on RAM" to RAM area from ROM area. And processing which reprograms code flash is executed on RAM.

Sample\_CodeFlashControl function processing does not have change from Simple version. Refer to the item of "Sample\_CodeFlashControl function" on Renesas Flash Driver RL78 Type 01 user's manual (R20UT4830).

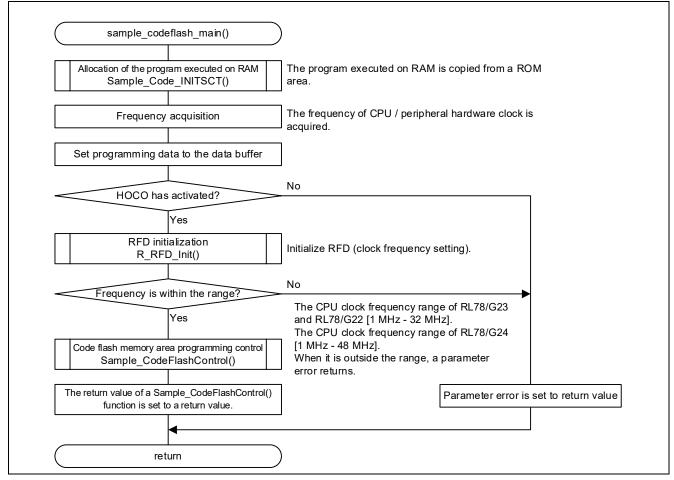


Figure 1.2 the flow chart of the sample program

- Notes: 1. In the code flash memory programming mode, the programs in the code flash memory cannot be executed. Copy the Sample\_CodeFlashControl function and the processing to be executed and data to be referenced to inside the function to RAM in advance, and execute and reference them in RAM.
  - 2."Frequency acquisition" of CPU and peripheral hardware clock settings is using the function included in the "RL78 Family Board Support Package".



### 2. Creating a Sample Project for Code Flash Reprogramming

### 2.1 Example of Creating a Sample Project

#### 2.1.1 In Case of CS+

Refer to Renesas Flash Driver RL78 Type 01 user's manual (R20UT4830) "Example of Creating a Sample Project" to create a project.

#### 2.1.2 In Case of e<sup>2</sup> studio(CC-RL)

Refer to Renesas Flash Driver RL78 Type 01 user's manual (R20UT4830) "Example of Creating a Sample Project" to create a project.

In this application note, because smart Configurator is used, press a "Next" button after selecting a target device and a debugging tool. And perform the following processes.

Select "Use Smart Configurator" and press a "Finish" button.

• - • ×
New Renesas CC-RL Executable Project Select Coding Assistant settings
Use Smart Configurator Use Peripheral Code Generator Smart Configurator is a single User Interface that combines the functionalities of Code Generator and RL78 SIS Configurator which imports, configures and generates different types of drivers and middleware modules. Smart Configurator encompasses unified clock configuration view, interrupt configuration view and pin configuration view. Hardware resources conflict in peripheral modules, interrupts and pins occurred in different types of drivers and middleware modules will be notified.
(Smart Configurator is available only for the supported devices)
American Concel

#### 2.1.3 In Case of IAR EW for Renesas RL78

Refer to Renesas Flash Driver RL78 Type 01 user's manual (R20UT4830) "Example of Creating a Sample Project" to create a project.



#### 2.1.4 In Case of e<sup>2</sup> studio(LLVM)

Refer to Renesas Flash Driver RL78 Type 01 user's manual (R20UT4830) "Example of Creating a Sample Project" to create a project.

In this application note, because smart Configurator is used, press a "Next" button after selecting a target device and a debugging tool. And perform the following processes.

Select "Use Smart Configurator" and press a "Finish" button.

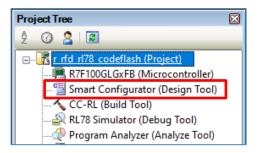
0		$\times$
LLVM for Renesas RL78 Select Smart Configurator settings		Ŷ
Use Smart Configurator The e2 studio peripheral smart configurator automatically generates programs (device drive		
peripheral functions (clocks, timers, serial interfaces, A/D converters, DMA controllers, etc.) b entered via a graphical user interface (GUI). Functions are provided as application programm (APIs) and are not limited to initialization of peripheral functions.		-
Application Code Software Components		
Software Components Middleware & Drivers Device Drivers		
MCU Hardware		
?	Can	el



### 2.2 Example of Source Code Registration

#### 2.2.1 In Case of CS+

(1) Double-click "Smart Configurator" (design Tool) of "Project Tree", and start Smart Configurator.



(2) Select a "Components" tab.

💰 Smart Configurator			_		×
File Window Help					
				: 🗈	😰   1
∰ r_rfd_rl78_codeflash.scfg ⊠		- 8	DCU/MPU Package	23	
Overview information G	enerate Code G	ienerate Report	<b>E A P</b>		•
- General Information		?			<u>^</u>
This editor allows you to modify the settings stored in configuration	on file (.scfg)				
Board					
Allow board and device selection					
Clocks					
Allow clock configuration			No i surace No i surace No i surace		
5					
Components				i i e i t	
Allow software component selection and configuration			e e		
Pins					
Allow general pin configuration and pin configuration for selected	d software comp				Ψ.
Overview Board Clocks System Components Pins Interrupt		×	▶ Legend		
📃 Console 🛛 📄 🔻 📩 🚽 🗖	🗆 🔝 Config	guration Problems	23	⇒ ⊽	
No consoles to display at this time.	0 items				
	Descriptio			Туре	
	<				>

(3) Press the "addition" button of "Components" and open the "addition" dialog of "Components".





(4) Select the following components and press a "Finish" button.

- Flash Driver[Renesas Flash Driver RL78 Type 01 Code Flash]( r\_rfd\_rl78\_t01\_codeflash)
- Flash Driver[Renesas Flash Driver RL78 Type 01 Flash Common]( r\_rfd\_rl78\_t01\_common)

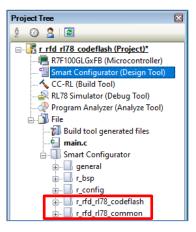
💦 New Co	omponent				$\times$
	• Component Selection mponent from those available in list				
Category	All				$\sim$
Function	All				$\sim$
Filter					
Compon	ents ^	Short Name	Туре	Version	^
🖶 Flash 🛙	Driver[Renesas Flash Driver RL78 Type01 Code Flash]	r_rfd_rl78_t01_codeflash	Generic SW	1.20	
🖶 Flash 🛙	Driver[Renesas Flash Driver RL78 Type01 Data Flash]	r_rfd_rl78_t01_dataflash	Generic SW	1.20	
🖶 Flash 🛛	Driver[Renesas Flash Driver RL78 Type01 Extra Area]	r_rfd_rl78_t01_extraarea	Generic SW	1.20	
🖶 Flash 🛛	Driver[Renesas Flash Driver RL78 Type01 Flash Common]	r_rfd_rl78_t01_common	Generic SW	1.20	~
Descriptio	only latest version on og to digital (A/D) converter is function for converting a	nalog inputs to digital sig	nals.		^
Download	RL78 Software Integration System modules				v
	d ELCL modules				
	general settings				
configure	general sectings				
?	< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish	Cancel	

(5) Press a "Generate Code" button and close "Smart Configurator" after the completion of generation for the code.

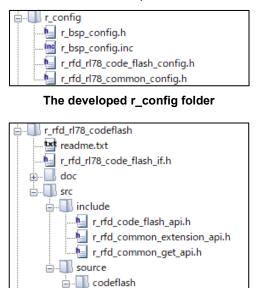
Software component configuration			🕒 Generate Report
Components 🚵 🖾 🖾 🗄 🕀	Configure		<b>(i)</b>
Image: Startup       ✓ Constraint       ✓ Constraint	Property There are no configurable items.		Value
<ul> <li>✓          ➢ Middleware     </li> <li>✓          Ø Generic     </li> <li>♂ r_rfd_rl78_t01_codeflash     </li> <li>♂ r_rfd_rl78_t01_common     </li> </ul>	< Note: r_rfd_r178_t01_codeflash_config.h file generated to r_config	folder needs to b	> oe modified

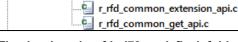


(6) The "r\_rfd\_rl78\_common" folder and the "r\_rfd\_rl78\_codeflash" folder is added to the project tree.



Each folder is developed as follows.

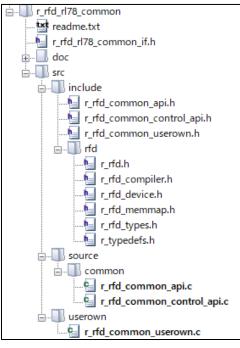




common

r\_rfd\_code\_flash\_api.c

The developed r\_rfd\_rl78\_codeflash folder



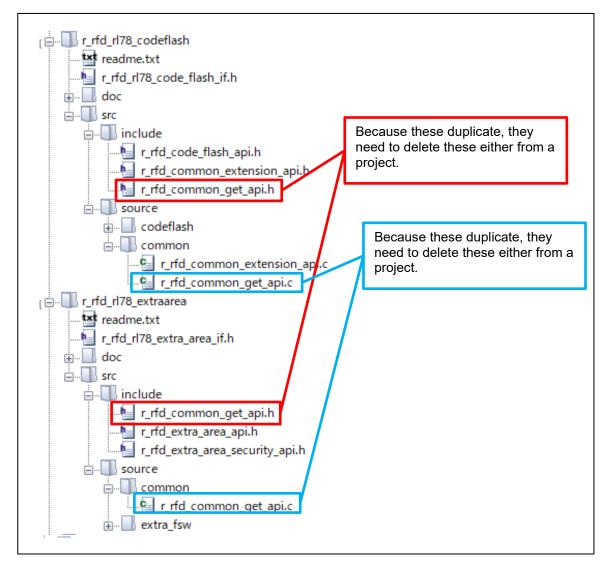
The developed r\_rfd\_rl78\_common folder



2.2.1.1 Notes of case used at the same time in r\_rfd\_rl78\_codeflash and r\_rfd\_rl78\_extraarea

In the case which uses r\_rfd\_rl78\_codeflash and r\_rfd\_rl78\_extraarea at the same time, as shown in the following figure, "r\_rfd\_common\_get\_api.c" and "r\_rfd\_common\_get\_api.h" overlap.

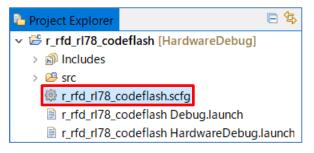
Each file needs to delete either from a project.





### 2.2.2 In Case of e<sup>2</sup> studio(CC-RL)

(1) Open the project file of "Smart Configurator" after starting e<sup>2</sup> studio.



(2) Select a "Components" tab.

Overview information
✓ General Information
This editor allows you to modify the settings stored in configuratio
Board Allow board and device selection
Clocks Allow clock configuration
Components Allow software component selection and configuration
<b>Pins</b> Allow general pin configuration and pin configuration for selected
Interrupt Allow general interrupt configuration and interrupt configuration for
Overview Board Clocks System Components Pins Interrupt

(3) Press the "addition" button of "Components" and open the "addition" dialog of "Components".

Components	
5ti 12	<b>i</b>
type filter text	
🗸 🗁 Startup	
🗸 🗁 Generic	
💣 r_bsp	



- (4) Select the following components and press a "Finish" button.
  - Flash Driver[Renesas Flash Driver RL78 Type 01 Code Flash]( r\_rfd\_rl78\_t01\_codeflash)
  - Flash Driver[Renesas Flash Driver RL78 Type 01 Flash Common]( r\_rfd\_rl78\_t01\_common)

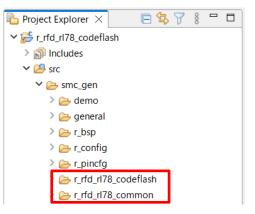
	omponent				×
Software	Component Selection				
Select cor	mponent from those available in list				
Category	All				~
Function	All				~
Filter					
	<u> </u>				
Compor		Short Name	Туре	 Version	<b>^</b>
	Driver[Renesas Flash Driver RL78 Type01 Code Flash]	r_rfd_rl78_t01_codeflash		1.20	
	Driver[Renesas Flash Driver RL78 Type01 Data Flash] Driver[Renesas Flash Driver RL78 Type01 Extra Area]	r_rfd_rl78_t01_dataflash r_rfd_rl78_t01_extraarea		1.20	
			Generic SW	1.20	
🖶 Flash	Driver[Renesas Flash Driver RL78 Type01 Flash Common]		Generic SW	1.20	~
Flash Show of Description	Driver[Renesas Flash Driver RL78 Type01 Flash Common] only latest version	r_rfd_rl78_t01_common	Generic SW	 1.20	
Hash Show of Descriptio	Driver[Renesas Flash Driver RL78 Type01 Flash Common] only latest version on	r_rfd_rl78_t01_common	Generic SW	1.20	
Flash Flash Show o Descriptio The anal Download	Driver[Renesas Flash Driver RL78 Type01 Flash Common] only latest version on og to digital (A/D) converter is function for converting an <u>d ELCL modules</u>	r_rfd_rl78_t01_common	Generic SW	 1.20	• •
Flash Flash Show o Description The anal Download Download	Driver[Renesas Flash Driver RL78 Type01 Flash Common] only latest version on og to digital (A/D) converter is function for converting an d ELCL modules d RL78 Software Integration System modules	r_rfd_rl78_t01_common	Generic SW	1.20	▼
Flash Flash Show o Description The anal Download Download	Driver[Renesas Flash Driver RL78 Type01 Flash Common] only latest version on og to digital (A/D) converter is function for converting an <u>d ELCL modules</u>	r_rfd_rl78_t01_common	Generic SW	1.20	
Flash Flash Show o Description The anal Download Download	Driver[Renesas Flash Driver RL78 Type01 Flash Common] only latest version on og to digital (A/D) converter is function for converting an d ELCL modules d RL78 Software Integration System modules	r_rfd_rl78_t01_common	Generic SW	1.20	

(5) Press a "Generate Code" button and generate the code.

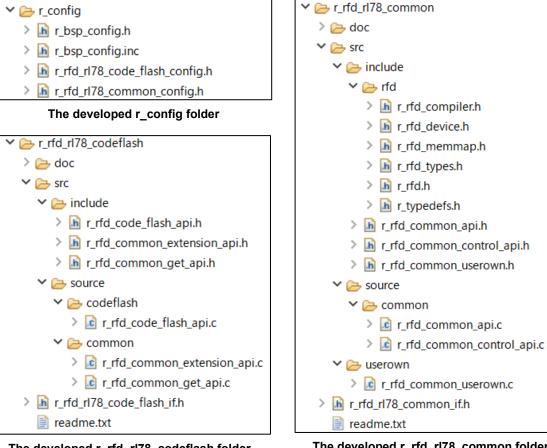
💮 r_rfd_rl78_codeflash.scfg 🗙			
Software component configur	ation	Generate Code Generate R	leport
Components 🚵 🛃 📮 🕀	Configure		í
type filter text	Property There are no config	urable items.	
✓			
✓ 🥭 Generic			
r_rfd_rl78_t01_codeflash			
♂ r_rfd_rl78_t01_common			
	<		>
	r_config folder needs to configure FIT module.	_config.h file generated to be modified manually to r to application note in "doc"	< >
Overview Board Clocks System Compon	ents Pins Interrupt		



(6) The "r\_rfd\_rl78\_common" folder and the "r\_rfd\_rl78\_codeflash" folder is added to the project tree.



Each folder is developed as follows.



The developed r\_rfd\_rl78\_codeflash folder

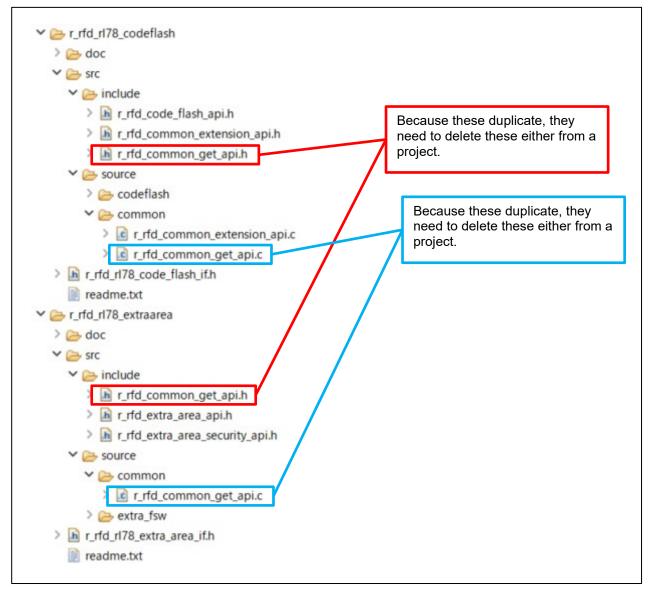
The developed r\_rfd\_rl78\_common folder



2.2.2.1 Notes of case used at the same time in r\_rfd\_rl78\_codeflash and r\_rfd\_rl78\_extraarea

In the case which uses r\_rfd\_rl78\_codeflash and r\_rfd\_rl78\_extraarea at the same time, as shown in the following figure, "r\_rfd\_common\_get\_api.c" and "r\_rfd\_common\_get\_api.h" overlap.

Each file needs to delete either from a project.



#### 2.2.3 In Case of IAR EW for Renesas RL78

(1) Select "File" [New...] after starting Smart Configurator for RL78.



(2) Select the item of "Platform" and "Toolchain".

Select the same device as the device selected in "the project of IAR EW for Renesas RL78" by "Platform".

Select "IAR RL78 Toolchain" as "Toolchain".

Input arbitrary names into "File name".

Press the "Browse..." button and set the location of the project folder for IAR EW for Renesas RL78. And press "Finish" button.

The ".setting" folder and the "<file name>.scfg" file are created to the set location.

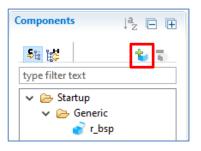
💰 New Sn	nart Configuration File	—		×
Smart Con	figuration Settings			
Select plat	form and toolchain settings for the new configuration	file		
Category:	RL78			$\sim$
Platform:	Toolchain:			
type filter	text Arenesas CCR	L78 Toolchain		
✓ Device	e 🛛 🗛 🚱 IAR RL78 Too	lchain		
√ RL	.78/G23 e <sup>2</sup> LLVM for Ren	iesas RL78		
~	RL78/G23 - 30pin			
	R7F100GAFxSP			
	R7F100GAGxSP			
	R7F100GAHxSP			
	R7F100GAJxSP			
~	RL78/G23 - 32pin			
	R7F100GBFxFP			
	R7F100GBFxNP			
ROM size:	96KB, RAM size: 12KB, Pin count: 30			
File name:	r_rfd_rl78_codeflash			
Location:	C:\WorkSpace\r_rfd_rl78_codeflash		Brow	se
	<u><u> </u></u>	nish	Cance	I



(3) Select a "Components" tab.

🐼 Smart Configurator			-		×
File Window Help					
				8	<u></u>
In r_rfd_rl78_codeflash.scfg ⋈				- 0	8
Overview information		Generate Code	Generate	Report	<b>D</b>
← General Information				?	•
This editor allows you to modify the settings stored in configura	tion file (.scfg)				
Board					
Allow board and device selection				_	
Clocks			cation und elopment		
Allow clock configuration				_	
Components		Device	Middlev	vare	
Allow software component selection and configuration		driver	R	TOS	
		1110		77	
Pins					
Allow general pin configuration and pin configuration for select	ed software component				,
Overview Board Clocks System Components Pins Interrupt					
🖳 Console 🛛 🚽 🔄 🕆 🗂 🗖	Configuration Problems	22	*		
No consoles to display at this time.	0 items				_
	Description			Туре	
	<			>	•

(4) Press the "addition" button of "Components" and open the "addition" dialog of "Components".





(5) Select the following components and press a "Finish" button.

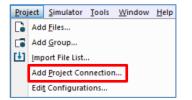
- Flash Driver[Renesas Flash Driver RL78 Type 01 Code Flash]( r\_rfd\_rl78\_t01\_codeflash)
- Flash Driver[Renesas Flash Driver RL78 Type 01 Flash Common]( r\_rfd\_rl78\_t01\_common)

💦 New Co	omponent					×
Software	Component Selection					
Select cor	mponent from those available in list					-
Catalana	All					
	All					~
Function	All					$\sim$
Filter						
Compon	ents ^	Short	Name	Type	Version	^
🖶 Flash [	Driver[Renesas Flash Driver RL78 Type01 Code Flas	sh] r_rfd_	rl78_t01_codef	lash Generic SW	1.20	
🖶 Flash [	Driver[Renesas Flash Driver RL78 Type01 Data Flas	h] r_rfd_	rl78_t01_datafl	ash Generic SW	1.20	•
🕀 Flash (	Driver/Renesas Flash Driver RL78 Type01 Extra Are	al rrfd	rl78 t01 extraa	rea Generic SW	1.20	
🖶 Flash [	Driver[Renesas Flash Driver RL78 Type01 Flash Cor	nmon] r_rfd_	rl78_t01_comm	on Generic SW	1.20	~
Show o	only latest version					
Descriptio	n					
The anal	og to digital (A/D) converter is function for conve	rting analog i	nputs to digita	l signals.		~
						$\sim$
Download	RL78 Software Integration System modules					
Download	ELCL modules					
Configure	general settings					
					•	
?	< <u>E</u>	<u>B</u> ack	<u>N</u> ext >	<u>F</u> inish	Cance	el

(6) Press a "Generate Code" button and close "Smart Configurator" after the completion of generation for the code.

Software component configura	tion	🖫 Generate Code	Generate Report
Components 🚵 🖾 🖾 🗈 🕀	Configure		i
Image: Startup       ✓ Ima	Property There are no configurable items.	Value	
✓ Generic	< Note: r_rfd_rl78_t01_codeflash_config.h file generated t	to r_config folder n	eeds to be
Overview Board Clocks System Component	s Pins Interrupt		

(7) Start IAR EW for Renesas RL78. And select "Project" menu [Add Project Connection], and open the additional dialog of Project Connection.





(8)Select "IAR Project Connection", and press an "OK" button.

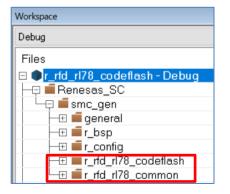
Add Project Con	×	
Connect using:	IAR Project Connection	~
	OK	Cancel

(9) Select the icpf file created by Smart Configurator and press an "Open" button.

Select IAR Project C	Connection File			×
$\leftrightarrow \rightarrow \cdot \uparrow$	« RL78G2x > WorkSpace > r_rfd_rl78	8_codeflash → v ੋ	,  ○ Search r	rfd_rl78_codeflash
Organize 🔻 New	v folder			··· · ·· ·· ·· ·· ·· ··· ·· ··· ·· ··· ·· ··· ·· ··· ·· ··· ·· ····
💻 This PC	^ Name	Date modified	Туре	Size
🧊 3D Objects	.settings	3/24/2023 11:38 AM	File folder	
	Debug	3/24/2023 11:38 AM	File folder	
Documents	settings	3/24/2023 11:38 AM	File folder	
Downloads	src	3/24/2023 11:38 AM	File folder	
Music	trash	3/24/2023 11:38 AM	File folder	
-	buildinfo.ipcf	12/20/2022 11:50 AM	IPCF File	18 KB
Pictures Videos				
🚔 Windows (C:)	× <			3
	File <u>n</u> ame:	~	IAR Project Co	nnection File (*.ip
			<u>O</u> pen	Cancel

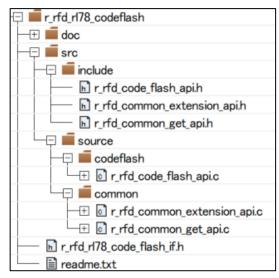


(10) "r\_rfd\_rl78\_common" and "r\_rfd\_rl78\_codeflash" are added to Workspace.

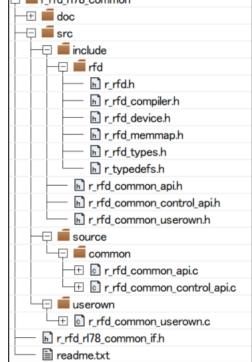


Each folder is developed as follows.

🗇 📕 r_config	🗐 r_rfd_rl78_common
h r_bsp_config.h	
r_bsp_config.inc	
├── h r_rfd_rl78_code_flash_config.h	— 🖓 🛑 include
▶ r_rfd_rl78_common_config.h	- 🖓 💼 rfd
The developed r_config folder	□ r_rfd.h
The developed I_coming tolder	r_rfd_o



The developed r\_rfd\_rl78\_codeflash folder



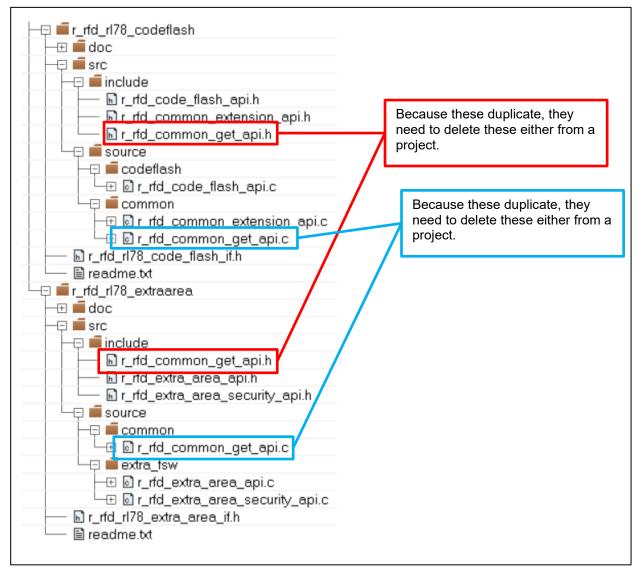
The developed r\_rfd\_rl78\_common folder



2.2.3.1 Notes of case used at the same time in r\_rfd\_rl78\_codeflash and r\_rfd\_rl78\_extraarea

In the case which uses r\_rfd\_rl78\_codeflash and r\_rfd\_rl78\_extraarea at the same time, as shown in the following figure, "r\_rfd\_common\_get\_api.c" and "r\_rfd\_common\_get\_api.h" overlap.

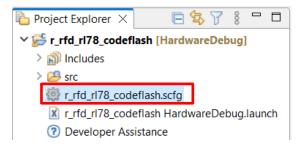
Each file needs to delete either from a project.



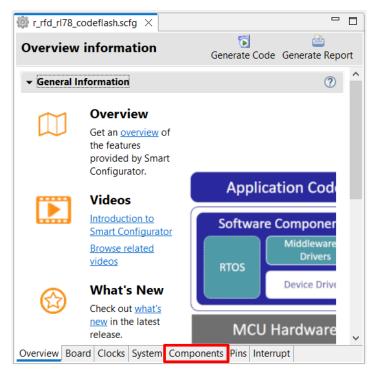


### 2.2.4 In Case of e<sup>2</sup> studio(LLVM)

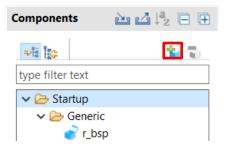
(1) Open the project file of "Smart Configurator" after starting e<sup>2</sup> studio.



(2) Select a "Components" tab.



(3) Press the "addition" button of "Components" and open the "addition" dialog of "Components".





- (4) Select the following components and press a "Finish" button.
  - Flash Driver[Renesas Flash Driver RL78 Type 01 Code Flash]( r\_rfd\_rl78\_t01\_codeflash)
  - Flash Driver[Renesas Flash Driver RL78 Type 01 Flash Common]( r\_rfd\_rl78\_t01\_common)

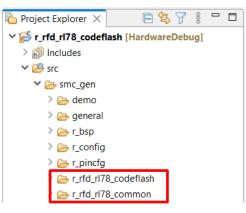
	omponent				×
Software	Component Selection				
Select cor	mponent from those available in list				
Category	All				~
Function	All				~
Filter					
	<u> </u>				
Compor		Short Name	Туре	 Version	<b>^</b>
	Driver[Renesas Flash Driver RL78 Type01 Code Flash]	r_rfd_rl78_t01_codeflash		1.20	
	Driver[Renesas Flash Driver RL78 Type01 Data Flash] Driver[Renesas Flash Driver RL78 Type01 Extra Area]	r_rfd_rl78_t01_dataflash r_rfd_rl78_t01_extraarea		1.20	
			Generic SW	1.20	
🖶 Flash	Driver[Renesas Flash Driver RL78 Type01 Flash Common]		Generic SW	1.20	~
Flash Show of Description	Driver[Renesas Flash Driver RL78 Type01 Flash Common] only latest version	r_rfd_rl78_t01_common	Generic SW	 1.20	
Hash Show of Descriptio	Driver[Renesas Flash Driver RL78 Type01 Flash Common] only latest version on	r_rfd_rl78_t01_common	Generic SW	1.20	
Flash Flash Show o Descriptio The anal Download	Driver[Renesas Flash Driver RL78 Type01 Flash Common] only latest version on og to digital (A/D) converter is function for converting an <u>d ELCL modules</u>	r_rfd_rl78_t01_common	Generic SW	 1.20	• •
Flash	Driver[Renesas Flash Driver RL78 Type01 Flash Common] only latest version on og to digital (A/D) converter is function for converting an d ELCL modules d RL78 Software Integration System modules	r_rfd_rl78_t01_common	Generic SW	1.20	▼
Flash	Driver[Renesas Flash Driver RL78 Type01 Flash Common] only latest version on og to digital (A/D) converter is function for converting an <u>d ELCL modules</u>	r_rfd_rl78_t01_common	Generic SW	1.20	
Flash	Driver[Renesas Flash Driver RL78 Type01 Flash Common] only latest version on og to digital (A/D) converter is function for converting an d ELCL modules d RL78 Software Integration System modules	r_rfd_rl78_t01_common	Generic SW	1.20	

(5) Press a "Generate Code" button and generate the code.

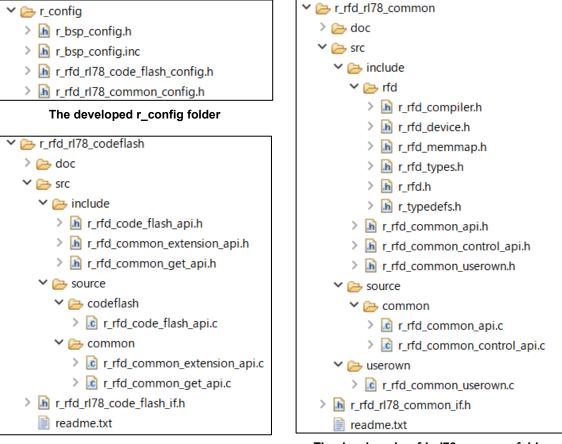
🔅 r_rfd_rl78_codeflash.scfg 🛛	
Software component configuration	Generate Code Generate Report
Components 🛛 🚵 🗳 📮 🕀	Configure (i)
type filter text	Property There are no configurable
<ul> <li>✓ ➢ Startup</li> <li>✓ ➢ Generic</li> <li>✓ r_bsp</li> <li>✓ ➢ Middleware</li> <li>✓ ➢ Generic</li> </ul>	
r_rfd_rl78_t01_codeflash	C >>
Overview Board Clocks System Compor	configurable items.



(6) The "r\_rfd\_rl78\_common" folder and the "r\_rfd\_rl78\_codeflash" folder is added to the project tree.



Each folder is developed as follows.



The developed r\_rfd\_rl78\_codeflash folder

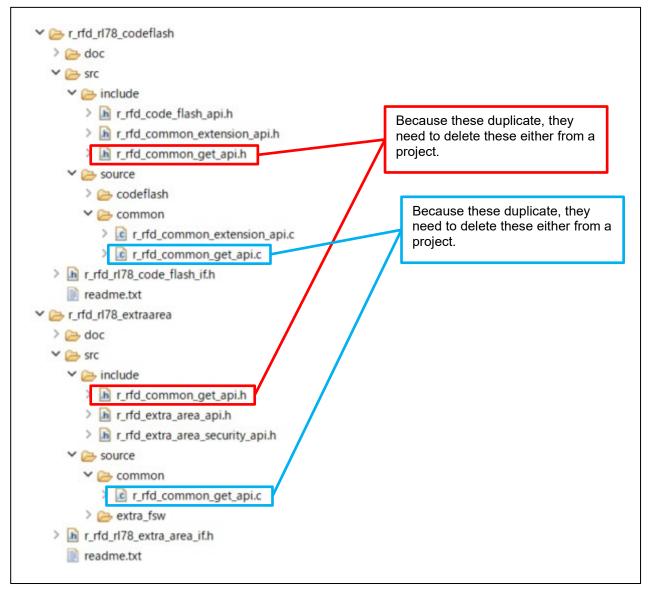
The developed r\_rfd\_rl78\_common folder



2.2.4.1 Notes of case used at the same time in r\_rfd\_rl78\_codeflash and r\_rfd\_rl78\_extraarea

In the case which uses r\_rfd\_rl78\_codeflash and r\_rfd\_rl78\_extraarea at the same time, as shown in the following figure, "r\_rfd\_common\_get\_api.c" and "r\_rfd\_common\_get\_api.h" overlap.

Each file needs to delete either from a project.



#### 2.3 Project registration of sample program

(1) Extract "CF\_sample.zip."

The folder of the device name generated after extracting leaves only the folder of a target device. Delete the folders of other device names. For example, when using the RL78/G23, leave only the "RL78\_G23" folder, and delete the untargeted "RL78\_G24" and other folders together.

The common file duplicates in the case which uses it at the same time with the sample program of a data flash or an extra area. Extract to overwrite both as the same folder name.

(2) Register the folder of the sample program into the project of CS+, e<sup>2</sup> studio, or IAR.

\* Files included in the folder other than the compiler package used, do not need to be registered.

- (3) Setting the section items
- (3-1) Setting of the section items on CS+

Setting of the section Items on CS+ inputs in the "Link Options" tab. (Common in each area)

- Setting the [Section] items

Set "No" to [Layout sections automatically]. And sections come to be displayed on [Section start address]. Press the "...." " button of the right-hand side which sections are displaying, and a "Section Settings" screen is displayed.

$\geq$	Device	
>	Output Code	
>	List	
>	Variables/functions information	
~	Section	
	Layout sections automatically	No
	Section start address	.const,.text,.RLIB,.SLIB,.textf,.constf,.data
>	Section that outputs external defined symbols to the file	Section that outputs external defined symbols to the file
>	ROM to RAM mapped section	ROM to RAM mapped section[2]
>	Verify	
>	Message	

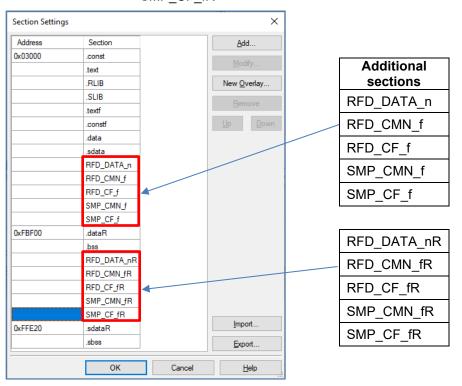
Note: Read the folder name ("RL78\_G23") of the sample for RL78/G23 as the folder name of a target device.

The folder name in the case of using RL78/G24: "RL78\_G24"



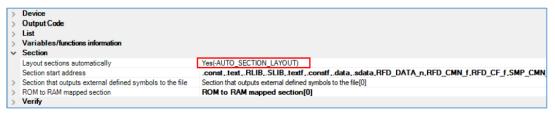
Add sections necessary for code flash memory reprogramming on a "Section Settings" screen.

Add to the program area : RFD\_DATA\_n, RFD\_CMN\_f, RFD\_CF\_f, SMP\_CMN\_f, SMP\_CF\_f Add to the RAM area : RFD\_DATA\_nR, RFD\_CMN\_fR, RFD\_CF\_fR, SMP\_CMN\_fR, SMP\_CF\_fR



Note: About setting of the sample project in the case of using each device, refer to the "Setting related to changing devices" section of the "Creating a Sample Project for RFD RL78 Type 01" chapter after "RL78 Family Renesas Flash Driver RL78 Type 01 User's Manual(R20UT4830)" Rev.1.20 or later.

Be sure to return [Layout sections automatically] to "Yes", after pressing the "OK" button.



Press the right-hand side " .... " button by [ROM to RAM mapped section], display the "Text Edit" screen, and add the section for copying to RAM from ROM.

Text Edit		ROM to RAM mapped section		
<u>T</u> ext:			.data=.dataR	
.data=.dataR .sdata=.sdataR			.sdata=.sdataR	
RFD_DATA_n=RFD_DATA_nR RFD_CMN_f=RFD_CMN_fR			RFD_DATA_n=RFD_DATA_nR	
RFD_CF_f=RFD_CF_fR SMP_CMN_f=SMP_CMN_fR			RFD_CMN_f=RFD_CMN_fR	
SMP_CF_f=SMP_CF_fR	J		RFD_CF_f=RFD_CF_fR	
			SMP_CMN_f=SMP_CMN_fR	
5		•	SMP_CF_f=SMP_CF_fR	



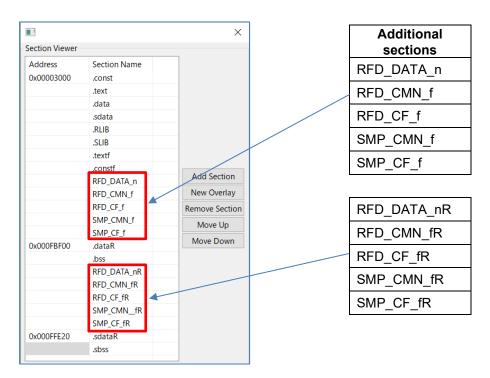
(3-2) Setting of the section items on e<sup>2</sup> studio(CC-RL)

On e<sup>2</sup> studio(CC-RL), sections are set on a "Section Viewer" screen.

The process which displays a "Section Viewer" screen is shown. Select the [Properties] from the [Project] menu and open the window of the properties for a target project. And select "Linker" [Section] from [Settings] of "C/C++ Build". Remove a check mark from [Layout sections automatically] (-auto\_section\_layout), and press the "

type filter text Settings	↓ ↓ ↓
<ul> <li>&gt; Resource Builders</li> <li>&gt; SMS Assembler</li> <li>&gt; SMS Assembler</li> <li>&gt; C(C++ Build Build Variables Environment</li> <li>&gt; Common</li> <li>&gt; Section</li> <li>&gt; Sec</li></ul>	Specify execution start address (-entry)     Execution start address (-entry= <symbol>) _start     .ayout sections automatically (-auto_section_layout) Sections (-start)     .const.text,data,sdata,RLIB,SLIB,textf,constf,</symbol>

#### Add to the program area : RFD\_DATA\_n, RFD\_CMN\_f, RFD\_CF\_f, SMP\_CMN\_f, SMP\_CF\_f Add to the RAM area : RFD\_DATA\_nR, RFD\_CMN\_fR, RFD\_CF\_fR, SMP\_CMN\_fR, SMP\_CF\_fR



Note: About setting of the sample project in the case of using each device, refer to the "Setting related to changing devices" section of the "Creating a Sample Project for RFD RL78 Type 01" chapter after "RL78 Family Renesas Flash Driver RL78 Type 01 User's Manual(R20UT4830)" Rev.1.20 or later.

Be sure to put a check mark to [Layout sections automatically (-auto\_section\_layout)], then press the "OK" button.

Specify execution start address (-entry)	
Execution start address (-entry= <symbol>)</symbol>	_start
Layout sections automatically (-auto_sect	tion_layout)
Sections (-start)	.const,.text,.data,.sdata,.RLIB,.SLIB,.textf,

Select "C/C++ Build" [Setting] - "Linker" [Section], display the "ROM to RAM mapped section (-rom)" screen, and add the section for copying to RAM from ROM.

ROM to RAM mapped section (-	rom)	ROM to RAM mapped section
.data=.dataR .sdata=.sdataR		.data=.dataR
RFD_DATA_n=RFD_DATA_nR		.sdata=.sdataR
RFD_CMN_f=RFD_CMN_fR RFD_CF_f=RFD_CF_fR		RFD_DATA_n=RFD_DATA_nR
SMP_CMN_f=SMP_CMN_fR SMP_CF_f=SMP_CF_fR		RFD_CMN_f=RFD_CMN_fR
		RFD_CF_f=RFD_CF_fR
		SMP_CMN_f=SMP_CMN_fR
		SMP_CF_f=SMP_CF_fR



(3-3) Setting of the section items on IAR EW for Renesas RL78

On IAR Embedded Workbench, Linker configuration file (\*. icf) describes link setting executed by building. Select "Options" by the click right mouse button of project with tree. Select [Linker] by "Category" in the displayed window, And put a check mark to "Override default" of the [Config] tab. Select Linker configuration file (\*. icf) in the "Open" window of " ... " button. Select the "sample\_linker\_file\_(area name).icf" file prepared for RFD RL78 Type 01. Linker configuration file (\*. icf) for every reprogramming area is as follows.

- For code flash memory reprogramming : sample\_linker\_file\_CF.icf

RL78/G23:(\CF\_sample\RL78\_G23\IAR\)

Category: General Options Static Analysis							Factor	y Settings
C/C++ Compiler	#define	Diag	nastics	Check		Encodings	Extra	Ontions
Assembler		Diag Library	nostics Input		zations	Encodings Advanced	Output	Options List
•	Linker co ☑ Overn C: 8_G23 → IAR	onfigurati ride defau	ion file Ilt		flash¥src¥	CF_sample¥RI	L78_G23¥IA	×
Organize  New folde	r	^				:== ▼		2
Downloads     Music     Pictures     Videos	Name	r_file_CF.icf			Date mod 3/15/2021		Type ICF File	
🔛 Windows (C:) 🗸 <								>
File <u>n</u> a	me: inker_file_CF	.icf		~	Icf File	es (*.icf)		,

- \* On the case used at the same time with data flash area or extra area, it is necessary to modify the icf file suitable for the sample program for the area used.
- Note: About setting of the sample project in the case of using each device, refer to the "Setting related to changing devices" section of the "Creating a Sample Project for RFD RL78 Type 01" chapter after "RL78 Family Renesas Flash Driver RL78 Type 01 User's Manual(R20UT4830)" Rev.1.20 or later.



Set the items of [General Options] - [Target] tab in the "Options" screen. Select the target device for [Device] and "Far" for [Code model].

Options for node "r_rfd_rl78_code	flash"				×
Category: General Options Static Analysis C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger COM Port E1 E2 E20 E2 Lite / E2 On-board EZ-CUBE EZ-CUBE EZ-CUBE Simulator TK	Library Options 2 Target Output Device RL78 - R7F100GLG Code model Far Use far runtime lib Data model Near Near constant location Qverride default addr Mirror ROM 0	rary calls		ouble':	
		[	OK	Cancel	



(3-4) Setting of the section items on e<sup>2</sup> studio(LLVM)

On e<sup>2</sup> studio (LLVM), Linker script (\*. ld) describes link setting executed by building. Select the [Property] in a project, and open a Properties window. Input the linker script directory path in the window displayed by selection of "C/C++" build [Setting] - "Linker" [Source].

That linker script path to add is shown below.

\${workspace\_loc:/\${ProjName}/src/CF\_sample/RL78\_G23/LLVM/sample\_linker\_file\_CF.ld}

Properties for r_rfd_rl78_code	iflash — 🗆 X
type filter text	Settings $\Leftrightarrow \bullet \bullet \bullet \$$
<ul> <li>&gt; Resource Builders</li> <li>&gt; C/C++ Build Build Variables Environment Logging Settings Tool Chain Editor</li> <li>&gt; C/C++ General Project Natures Project References Refactoring History Renesas QE Run/Debug Settings Task Tags</li> <li>&gt; Validation</li> </ul>	Configuration:       HardwareDebug [Active]         Image: Configuration:       Image: Configuration:         Image: CPU       Device         Image: CPU       Entry point:         Image: CPU       Image: CPU         Image: CPU       Entry point:         Image: CPU       Image: CPU         Image: CPU       Entry point:         Image: CPU       Image: CPU         Image: CPU       Image: CPU         Image: CPU       Image: CPU         Image: CPU       Entry point:         Image: CPU       Image: CPU         Image: CPU       Image: CPU         Image: CPU       Image: CPU         Image: CPU       Entry point:         Image: CPU       Image: CPU         Image: CPU       Image: CPU
?	Apply and Close Cancel

- \* It is an example of the include path in the case which extracted CF\_sample.zip directly under the src folder.
- Note: About setting of the sample project in the case of using each device, refer to the "Setting related to changing devices" section of the "Creating a Sample Project for RFD RL78 Type 01" chapter after "RL78 Family Renesas Flash Driver RL78 Type 01 User's Manual(R20UT4830)" Rev.1.20 or later.



(4) Include Path Settings

The figure shows the case of RL78/G23. This case also read the folder name ("RL78\_G23") of the sample for RL78/G23 as the folder name of a target device.

The folder name in the case of using RL78/G24: "RL78\_G24"

(4-1) Setting of the include path on CS+

Setting of the include path on CS+ inputs path in "Common Options" tab.

- Add the Include directory path in the " Path Edit" window displayed by selection of [Frequently Used Options(for Compile)] - [Additional include paths].

In the phase where code generation was performed by (5) of "2.2.1 In Case of CS+", the include path of files other than the sample program is registered. For a reason, it needs to register the include path of the sample program.

Those include path to add is shown below.

src\CF_sample\RL78_G23\config		
src\CF_sample\RL78_G23		
src\CF_sample\common\include		
Path Edit	×	
Path(One path per one line):		
src¥CF_sample¥RL78_G23¥config src¥CF_sample¥RL78_G23 src¥CF_sample¥common¥include src¥smc_gen¥r_rfd_r178_extraarea src¥smc_gen¥r_rfd_r178_extraarea		
src¥smc_gen¥r_ifd_i178_common¥src¥include¥fd src¥smc_gen¥r_ifd_i178_common¥src¥include src¥smc_gen¥r_ifd_i178_common	Ų	
<	>	
Browse		
Permit non-existent path		
Include subfolders automatically		

\* It is an example of the include path in the case which extracted CF\_sample.zip directly under the src folder.

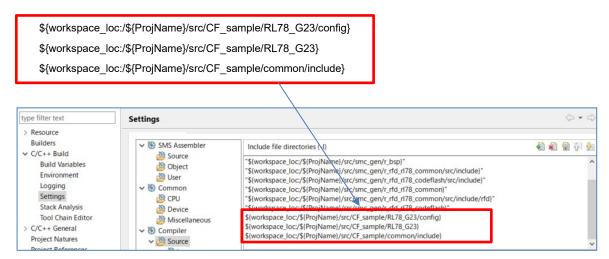


(4-2) Setting of the include path on e<sup>2</sup> studio(CC-RL)

Setting of the include path on e<sup>2</sup> studio(CC-RL) inputs path in "Properties" window.

- Input the Include directory path in the window displayed by selection of "C/C++" build [Setting] - "Compiler" [Source].

Those include path to add is shown below.



\* It is an example of the include path in the case which extracted CF\_sample.zip directly under the src folder.



(4-3) Setting of the include path on IAR EW for Renesas RL78

Setting of the include path on IAR Embedded Workbench selects "C/C++ Compiler" of "Category", and inputs path in "Preprocessor" tab.

- Input the Include directory path in the "Edit include Directories" window displayed by selection of [Additional include directories: (one per line)].

Those include path to add is shown below.

\$PROJ_DIR\$\src\CF_san \$PROJ_DIR\$\src\CF_san \$PROJ_DIR\$\src\CF_san							
Category:				Facto	ory Settings		
General Options	Multi-file Compilat	ion					
Static Analysis	Discard Unus	sed Publics					
C/C++ Compiler	Language 1 L	anguage 2	Optimizations	Output	List		
Assembler	Preprocessor	Disgnostics	Encodings	Extr	a Option <i>s</i>		
Output Converter	į́gnore standard	include divector	rie si				
Custom Build Build Actions	Additional include d						
Linker			/ 178_common¥src¥ 179_common¥src¥	include	A		
Debugger	\$PROJ_DIR\$¥src			in ely de ¥rfd			
COM Port	\$PROJ_DIR\$¥src \$PROJ_DIR\$¥src				~		

\* It is an example of the include path in the case which extracted CF\_sample.zip directly under the src folder.



(4-4) Setting of the include path on e<sup>2</sup> studio(LLVM)

Setting of the include path on e<sup>2</sup> studio(LLVM) inputs path in "Properties" window.

- Input the Include directory path in the window displayed by selection of "C/C++" build [Setting] -

"Compiler" [Includes].

Those include path to add is shown below.

\${workspace_loc:/\${ProjName}/src/CF_sample/RL78_G23/config} \${workspace_loc:/\${ProjName}/src/CF_sample/RL78_G23} \${workspace_loc:/\${ProjName}/src/CF_sample/common/include}						
😵 Tool Settings 🗞 Toolchain 🛞 Device 🎤 Build Steps 🙅 Build Artifact 🗟 Binary Parsers 😣 Error Parsers						
<ul> <li>CPU</li> <li>Optimization</li> <li>Debug</li> <li>Warnings</li> <li>SMS Assembler</li> <li>Source</li> <li>Object</li> <li>Solitorary Generator</li> <li>Settings</li> <li>Source</li> <li>Compiler</li> <li>Source</li> <li>Includes</li> <li>Sesembler</li> </ul>	Include file directories (-1) "\${workspace_loc:/\${ProjName}/src/smc_gen/r_pincfg}" "\${workspace_loc:/\${ProjName}/src/smc_gen]" "\${workspace_loc:/\${ProjName}/src/smc_gen]" "\${workspace_loc:/\${ProjName}/src/smc_gen/r_bap}" "\${workspace_loc:/\${ProjName}/src/smc_gen/r_fd_r178_con "\${workspace_loc:/\${ProjName}/src/smc_gen/r_fd_r178_con "\${workspace_loc:/\${ProjName}/src/smc_gen/r_fd_r178_con "\${workspace_loc:/\${ProjName}/src/smc_gen/r_fd_r178_con "\${workspace_loc:/\${ProjName}/src/smc_gen/r_fd_r178_con "\${workspace_loc:/\${ProjName}/src/smc_gen/r_fd_r178_con \${workspace_loc:/\${ProjName}/src/smc_gen/r_fd_r178_con \${workspace_loc:/\${ProjName}/src/smc_gen/r_fd_r178_con \${workspace_loc:/\${ProjName}/src/Src_sample/RL78_623/co \${workspace_loc:/\${ProjName}/src/CF_sample/RL78_623} \${workspace_loc:/\${ProjName}/src/CF_sample/common/inc	nmon/src/include)" leflash)" leflash/src/include)" nmon)" nnfig)				

\* It is an example of the include path in the case which extracted CF\_sample.zip directly under the src folder.



- (5) The setting of user definition macro
- (5-1) Setting of the user definition macro on CS+

Refer to the chapter of "The setting of user definition macro" of Renesas Flash Driver RL78 Type 01 user's manual (R20UT4830).

(5-2) Setting of the user definition macro on e<sup>2</sup> studio(CC-RL)

Refer to the chapter of "The setting of user definition macro" of Renesas Flash Driver RL78 Type 01 user's manual (R20UT4830).

(5-3) Setting of the user definition macro on IAR EW for Renesas RL78

Refer to the chapter of "The setting of user definition macro" of Renesas Flash Driver RL78 Type 01 user's manual (R20UT4830).

(5-4) Setting of the user definition macro on e<sup>2</sup> studio(LLVM)

Refer to the chapter of "The setting of user definition macro" of Renesas Flash Driver RL78 Type 01 user's manual (R20UT4830).



- (6) Device Item Settings
- (6-1) Setting of the device Items on CS+

Refer to the chapter of "Device Item Settings" of Renesas Flash Driver RL78 Type 01 user's manual (R20UT4830).

(6-2) Setting of the device Items on e<sup>2</sup> studio(CC-RL)

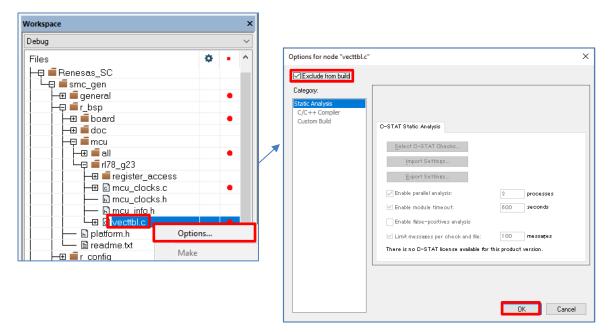
Refer to the chapter of "Device Item Settings" of Renesas Flash Driver RL78 Type 01 user's manual (R20UT4830).

(6-3) Setting of the device Items on IAR EW for Renesas RL78

Build including vecttbl.c prepared as a sample program. A user option byte's value is set to 0x6EFFE8, and an on-chip debugging option byte's value is set to 0x85.

When the code is generated by Smart Configurator, vecttbl.c is generated to a "smc\_gen\r\_bsp\mcu\rl78\_g23\" folder. And because vecttbl.c duplicates, it is necessary to repeal this file.

Right-click a mouse by "Renesas\_SC\smc\_gen\r\_bsp\mcu\rl78\_g23\vecttbl.c" in the [project] on a tree. And select an "option" and set a "check" to [Exclude from build] in the displayed screen. And exclude vecttbl.c from build target.



(6-4) Setting of the device Items on e<sup>2</sup> studio(LLVM)

Build including vects.c prepared as a sample program. A user option byte's value is set to 0x6EFFE8, and an on-chip debugging option byte's value is set to 0x85.

When the code is generated by Smart Configurator, r\_cg\_vect\_table.c is generated to a "src\general\" folder. And because r\_cg\_vect\_table.c and vects.c duplicate the settings, it is necessary to repeal this file.

Right-click a mouse by "src\general\r\_cg\_vect\_table.c" in the [project] on a Project Explorer. And select [Resource Configuration] - [Exclude from build...] and use it to exclude r\_cg\_vect\_table.c from build target.



(7) Execute the sample program from a main function.

Describe the sample\_codeflash\_main function included in r\_flash\_sample\_codeflash\_rl78g2x.c like the "main function" for the project. And build, download and execute it.

\* The header file which described the prototype declaration for sample\_codeflash\_main function. Include prepared "r\_flash\_sample\_codeflash\_rl78g2x.h."



## 2.4 The Check of Operation for Sample Program

The block 14 (0x00007000) of a code flash area is erased. And 16 Words(64 bytes) of data is programmed from the top of the block 14. Confirm this operation by the following methods.

## 2.4.1 In Case of CS+

- (1) Select [Debug]menu [Download] and start debugging.
- (2) After selecting [View]menu [Memory], select "Memory1", "Memory2", "Memory3", or "Memory4", and display a memory window.
- (3) Set "0x00007000" to the address of a memory window, and display the block 14 of a code flash area.

Memory1	X
Notation • Size Notation • Encoding • View •	
Move when Stop 0x00007000	Move

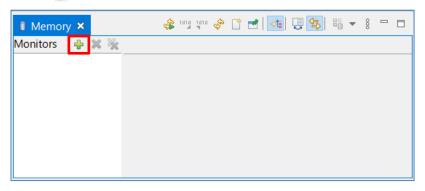
(4) Select [Debug]menu - [Go] and the program is executed. Check that the value of "0x00007000-0x0000703F" displayed on the memory window has changed after stopping the program.

Memory	1																	X
2 🛞	N	otati	ion	-	Size	Not	tatio	n -	E	nco	dind	-	Vie	w <del>+</del>				
<u> </u>	1			_							-							
Move	wher	n Sto	p	0x0	0007	7000											Move	
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+a	+b	+c	+d	+e	+f	ASCII	~
07000		नन	FF	नन	मम	TT	मम	नन	FF	नन	मम	FF	मम	मम	TT	नम	222222222222222222222222	- "
07010	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	22222222222222222222	
07020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	222222222222222222222222222222222222222	
07030	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	2222222222222222222	
07040	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	2222222222222222222	
07050	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	2222222222222222222	
07060	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	2222222222222222222	
07070	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???????????????????????????????????????	
07080	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???????????????????????????????????????	
07090	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???????????????????????????????????????	
070a0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???????????????????????????????????????	
070Ъ0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	2222222222222222222	
070c0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???????????????????????????????????????	
070d0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	2222222222222222222	
070e0		FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???????????????????????????????????????	
070£0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???????????????????????????????????????	~
07100 L	नन	नन	नन	नन	नन	नन	नन	नन	नन	नन	नन	नन	न न ।	नन	नन	नन	, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
<																	>	·



# 2.4.2 In Case of e<sup>2</sup> studio(CC-RL)

- (1) Select [Run]menu [Debug] and start debugging.
- (2) Select [Window]menu [Show View] [Memory] and display Memory view.
- (3) Press " 🐺 " button and display the Monitor Memory window.



(4) Input "0x00007000" into the address to monitor, and press the OK button. The block 14 of a code flash area is displayed on the memory view.

Monitor	Memory	×
Enter addre	ess or expressi	on to monitor:
		~
?	OK	Cancel

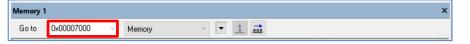
(5) After the program is executed, check that the value of "0x00007000-0x0000703F" displayed on the memory window is changing.

🖡 Memory 🗙		4	1019 1010 🚸 📑	🛃 🔤 😳	💫 📲 🗕 🖇 🗖
Monitors 🛛 🕂 🕷 💥	0x00007000 : 0x7000 <	Hex Integer> 🗙	💠 New Reno	lerings	
Ox00007000	Address	0 - 3	4 - 7	8 - B	C - F
	000000000007000	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007010	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007020	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007030	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007040	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007050	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007060	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007070	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007080	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007090	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	0000000000070A0	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	00000000000070B0	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	0000000000070C0	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	00000000000070D0	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	00000000000070E0	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	00000000000070F0	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF



### 2.4.3 In Case of IAR EW for Renesas RL78

- (1) Select [Project] menu [Download and Debug] and start debugging.
- (2) After select [View]menu [Memory], select either from "Memory 1" to "Memory 4", and the memory window is displayed.
- (3) Set "0x00007000" to the address of the memory window. And the block 14 of code flash area is displayed.



(4) After the program is executed, check that the value of "0x00007000-0x0000703F" displayed on the memory window is changing.

Memory 1					×
Go to 0x0000	7000 ~	Memory			
0x00007000 0x00007010	ff ff f ff ff f	f ff ff ff f 6 66 66 66 6	f ff ff ff ff ff ff	ff ff ff ff ff	^
0x00007020	ff ff f	f ff ff ff f	f ff ff ff ff ff	ff ff ff ff	
0x00007030 0x00007040	ff ff f ff ff f	f ff ff ff ff f f ff ff ff f	f ff ff ff ff ff ff	ff ff ff ff ff ff ff ff	
0x00007050 0x00007060	ff ff f ff ff f	f ff ff ff f f ff ff ff f	f ff ff ff ff ff ff f ff ff ff ff ff	ff ff ff ff ff	
0x00007070 0x00007080	ff ff f ff ff f	f ff ff ff f f ff ff ff f	f ff ff ff ff ff ff f ff ff ff ff ff	ff ff ff ff	
0x00007090 0x000070a0	ff ff f ff ff f	f ff ff ff f f ff ff ff f	f ff ff ff ff ff ff f ff ff ff ff ff	ff ff ff ff	
0x000070b0 0x000070c0	ff ff f			ff ff ff ff	
0x000070d0	ff ff f		f ff ff ff ff ff	ff ff ff ff	
0x000070e0 0x000070f0	ff ff f ff ff f		f ff ff ff ff ff ff f ff ff ff ff ff	LL LL LL LL	~



# 2.4.4 In Case of e<sup>2</sup> studio(LLVM)

- (1) Select [Run]menu [Debug] and start debugging.
- (2) Select [Window]menu [Show View] [Memory] and display Memory view.
- (3) Press " 🐺 " button and display the Monitor Memory window.

Memory	/ <b>X</b>		<b>\$</b>	1010	1010 N	Ŷ	ľ	-	⇒tġ	IJ	<b>₽</b> ₽	80	Ŧ	000	
Monitors	4	K %													

(4) Input "0x00007000" into the address to monitor, and press the OK button. The block 14 of a code flash area is displayed on the memory view.

Monitor	Memory	×
Enter addre	ess or expressi	on to monitor:
		~
?	OK	Cancel

(5) After the program is executed, check that the value of "0x00007000-0x0000703F" displayed on the memory window is changing.

🖡 Memory 🗙		4	1012 1010 🚸 📑	🛃 🔤 😳	💫 📲 🗕 🖇 🗖
Monitors 🛛 🕂 🕷 💥	0x00007000 : 0x7000 <	Hex Integer> 🗙	💠 New Rend	lerings	
Ox00007000	Address	0 - 3	4 - 7	8 - B	C - F
	000000000007000	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007010	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007020	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007030	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007040	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007050	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007060	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007070	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007080	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	000000000007090	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	0000000000070A0	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	00000000000070B0	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	0000000000070C0	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	00000000000070D0	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	00000000000070E0	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF
	0000000000070F0	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF



# 3. Precautions

- (1) Reprogramming of the code flash memory or extra area
- Place the reprogramming code in RAM when reprogramming the code flash memory or extra area.
- (2) Precondition for control of the data flash area Be sure to set the DFLEN bit (bit 0) of the data flash control register (DFLCTL) to 1 (enable access to the data flash area) before controlling the data flash area.
- (3) Program execution during reprogramming of the flash memory Self-programming in the RL78/G23 uses the flash memory sequencer to control the reprogramming of the flash memory. In the following flash memory control modes in which the flash memory can be reprogrammed, the CPU cannot read data from the target flash memory.
  - In the code flash memory programming mode, the CPU cannot read data from the code flash memory. The API functions of RFD RL78 Type 01 and the user program to be executed in the code flash memory programming mode should be copied from ROM to RAM in advance and executed and referenced in RAM.
  - In the data flash memory programming mode, the CPU cannot read data from the data flash memory. The data to be read in the data flash memory programming mode should be copied from the data flash memory to RAM in advance and referenced in RAM.
- (4) Clock setting under self-programming execution

The high-speed on-chip oscillator should be kept operating during self-programming. When selfprogramming is executed, it is necessary to use the main system clock and to select "X1 oscillator or the high-speed on-chip oscillator." Cannot execute self-programming, if the middle-speed on-chip oscillator or the subsystem clock is selected. Stop the middle-speed on-chip oscillator (MIOEN = 0) and select the high-speed on-chip oscillator (MCM1 = 0) as the main on-chip oscillator clock (fOCO).

- \* For the clock setting under self-programming execution, refer to the user's manual of the target RL78 microcontroller.
- (5) The precautions in the case of debugging self-programming with an on-chip debugger

In the case which debugs self-programming with an on-chip debugger, because 128 bytes of area is used from the top address of RAM when a debugger is executed, it is necessary to vacate this area. Additionally, in case CS+ or e<sup>2</sup> studio is used as the development environment, the debugger settings need to be configured to use flash self-programming

• Example settings for CS+:

On the project, select "Connect Settings" tab from "RL78 E2 [Lite] (Debug Tool)", and set "Yes" to "Flash" - "Using the flash self programming".

• Example settings for e<sup>2</sup> studio:

On the project, select "Property" - "Run/Debug Settings", and edit the target "HardwareDebug" setting. On the displayed screen, select "Debugger" tab - "Connection Settings" tab, and set "Yes" to "Flash" -"Program uses flash self programming".



# 4. Reference document

Please get the latest version of each document from the Renesas Electronics Corp. website (https://www.renesas.com).

No	Document Title Document Num					
1	RL78/G22 User's Manual Hardware	R01UH0978				
2	RL78/G23 User's Manual Hardware R01UH0896					
3	RL78/G24 User's Manual Hardware R01UH0961					
4	RL78 Family Board Support Package Module R01AN5522					
5	RL78 Family Renesas Flash Driver RL78 Type 01 User's Manual	R20UT4830				
6	E1/E20/E2 Emulator, E2 Emulator Lite Additional Document for User's Manual (Notes on Connection of RL78)	R20UT1994				



# 5. Revision History

Davi	Det.		Description
Rev.	Rev. Date -	Page	Summary
1.00	Dec.20.21	—	Newly created.
1.10	May.31.23		RL78/G22 and RL78/G24 were added.
1.20	Mar.22.24	—	The LLVM compiler was supported.



# 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  $V_{L}$  (Max.) and  $V_{IH}$  (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.).

#### 7. Prohibition of access to reserved addresses

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

8. Differences between products

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

## Notice

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

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

- 7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
- 8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
- 9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for 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 guestions regarding the information contained in this document or Renesas
- Electronics products.
  (Note1) "Renesse Electronics" as used in this document means Renesse Electronics Corporation and also includes its directly or indir
- (Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
- (Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

## **Corporate Headquarters**

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

## **Contact information**

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

# Trademarks

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