Smart Configurator for RL78 Plug-in in e^2 studio 2025-12
Smart Configurator for RL78 V1.16.0

Release Note

Introduction

Thank you for using the Smart Configurator for RL78.
This document describes the restrictions and points for caution. Read this document before using the product.

Contents

1. Introduction 3
1.1 System Requirements 3
1.1.1 Windows PC 3
1.1.2 Linux PC 3
1.1.3 Mac OS 3
1.1.4 Development Environments 4
1.2 Disclaimer 5

2. Support List 6
2.1 Support Devices List 6
2.2 Support Components List 8
2.3 New support 14
2.3.1 BSP (Board Support Package) revision update 14
2.3.2 SMS support UARTA 14
2.3.3 Support IIC Communication (Master mode, EEPROM communication) component 15

3. Changes 16
3.1 Correction of issues/limitations 16
3.1.1 Fixed the issue of not supporting flash serial programming security ID 16
3.1.2 Fixed the issue of not supporting "Does not use trigger input" in TMKB - Standalone Mode (Period controlled by external trigger edge) 17
3.1.3 Fixed the errors in DTC 17
3.1.4 Fixed the issue of only fIH and fIM can be selected as fCLK in low-power main mode 19
3.1.5 Fixed the issue of not supporting all address matching function in IICA 20
3.2 Specification changes 21
3.2.1 Improvement for updating the driver code in IIC Communication (Master mode) component 21
3.2.2 Improvement for updating the spec of [Clocks] page 22
3.2.3 Improvement for reducing power consumption 23
3.2.4 Improvement for updating the default status of the backup feature 23

4. List of RENESAS TOOL NEWS AND TECHNICAL UPDATE 24

5. Points for Limitation 25
5.1 List of Limitation 25
5.2 Details of Limitation 26
5.2.1 Note on extra help document issue 26
5.2.2 Note on ELCL D flip flop component GUI warning display incorrectly 27
5.2.3 Note on the unsupported setting items for some ELCL components 28
5.2.4 Note on the user code protection feature will only be supported on the files that are generated by the Code Generation component 28
5.2.5 Note on not using Flexible Application Accelerator (FAA) component in Mac OS and Linux 28
5.2.6 Note on not using SNOOZE Mode Sequencer (SMS) component in Mac OS and Linux 29
5.2.7 Note on UI display with High Contrast theme on Linux OS 29
5.2.8 Note on using a CCRL project in Visual Studio Code 29
5.2.9 Note on IIC Communication (Slave mode) can’t send or receive data correctly when using all address matching function in IICA 29
5.2.10 Note on the e2 studio property reverts to the default value after generating the code 32

6. Points for Caution 33
6.1 List of Caution 33
6.2 Details of Caution 35
6.2.1 Note on the build error message such as "section .bss virtual address range overlaps with .dtc_vectortable" 35
6.2.2 Note on the installation of the Smart Configurator 36
6.2.3 Note on using TRDIOA0 for Input capture and TRDIOB0 for Output compare at same time 36
6.2.4 Note on pulse width calculation of Timer RD input capture function 36
6.2.5 Note on the include path update issue when renaming the component’s configuration name 37
6.2.6 Note on TAU Input Signal High/Low level Measurement component 38
6.2.7 Note on CC-RL V1.12 C++ project 39
6.2.8 Note on browsing "Release Notes" and "Tool News" URL from the help menu 39
6.2.9 Note on using the user code protection feature 39
6.2.10 Note on IAR build error when using SNOOZE Mode Sequencer (SMS) component 40
6.2.11 Note on A/D conversion time setting after performing [Change device] or [Change resource] 41
6.2.12 Note on changing Hardware Debug Configuration on project generation wizard 42
6.2.13 Note on Pin Number maybe wrong in [Pins] page when loading project 42
6.2.14 Note on Pin assignment with PIOR maybe wrong in [Pins] page when changing device 42
6.2.15 Note on the case that a pin of an unsupported component is assigned when the device is changed 43
6.2.16 Note on the Pin assignement and pin setting maybe wrong when loading project 43
6.2.17 Note on the Developer Assistance can’t jump correctly to another HTML file 44
6.2.18 Note on the clock UI display issue when changing screen scale 45

Revision History 46

1. Introduction

Smart Configurator is a utility for combining software to meet your needs. It supports the following three functions related to the embedding of Renesas drivers in your systems: importing middleware, generating driver code, and setting pins.
Smart Configurator for RL78 V1.16.0 is equivalent to Smart Configurator for RL78 Plug-in in e^2 studio 2025-.

1.1 System Requirements

The operating environment is as follows.

1.1.1 Windows PC

- System: x64/x86 based processor
  Windows® 11
- Memory capacity: We recommend 4 GB or more.
- Capacity of hard disk: At least 300 MB of free space.
- Display: Graphics resolution should be at least 1024 x 768, and the mode should display at least 65,536 colors.
- Processor: 1 GHz or higher (must support hyper-threading, multi-core CPUs)

1.1.2 Linux PC

Smart Configurator for RL78 plug-in in e^2 studio 2023-01 or later is supported on Linux OS.

- System: x64 based processor, 2 GHz or faster (with multicore CPUs)
  Ubuntu 24.04 LTS Desktop (64-bit version)
  Ubuntu 22.04 LTS Desktop (64-bit version)
- Memory capacity: We recommend 2 GB or more.
- Capacity of hard disk: At least 2 GB of free space.

1.1.3 Mac OS

Smart Configurator for RL78 plug-in in e^2 studio 2024-04 or later is supported on Mac OS.

- System: Apple ARM-based systems-on-a-chip (SoCs)
  Mac OS 15 (Sequoia)
  Mac OS 14 (Sonoma)
- Memory capacity: 4 GB of RAM; 8 GB of RAM recommended.
- Capacity of hard disk: At least 2 GB of free space.
- A screen resolution of 1280 x 800 or higher.

Note: Only LLVM is available for Mac OS.

1.1.4 Development Environments

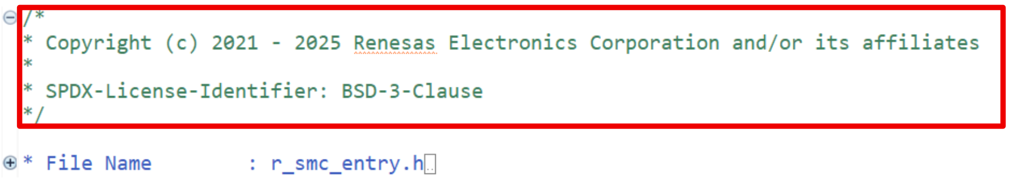
- Renesas Electronics Compiler for RL78 [CC-RL] V1.16 or later (Windows PC)
- LLVM for Renesas RL78 17.0.1.202506 or later (Windows PC, Linux PC, Mac OS)
- IAR Embedded Workbench for Renesas RL78 V5.10.3 or later (Windows PC)
- SMS Assembler V1.00.00 or later (Windows PC)
- FAA Assembler V1.06.00 or later (Windows PC)
- CS+ for CC V8.15.00 [Note1] or later (Windows PC)

Note:

1.Smart Configurator for RL78 V1.16.0 has been evaluated in the CS+ for CC V8.15.00 environment. When using Smart Configurator for RL78 V1.15.0 or lower, please refer to Release Note (R20UT5735EC0100) about the target version of CS+ for CC.

1.2 Disclaimer

From Smart Configurator for RL78 V1.16.0, the disclaimer of generated code has been updated to BSD-3-Clause.



```
/*
* Copyright (c) 2021 - 2025 Renesas Electronics Corporation and/or its affiliates
*
* SPDX-License-Identifier: BSD-3-Clause
*/

* File Name        : r_smc_entry.h
```

Figure 1-1 Disclaimer of generated code

2. Support List

2.1 Support Devices List

Below is a list of devices supported by the Smart Configurator for RL78 V1.16.0.

Table 2-1 Support Devices (1/2)

Group (HW Manual number)	PIN	Device name
RL78/G23 Group (R01UH0896EJ0120)	30pin	R7F100GAFxSP, R7F100GAGxSP, R7F100GAHxSP, R7F100GAJxSP
	32pin	R7F100GBFxNP, R7F100GBGxNP, R7F100GBHxNP, R7F100GBJxNP, R7F100GBFxFP, R7F100GBGxFP, R7F100GBHxFP, R7F100GBJxFP
	36pin	R7F100GCFxLA, R7F100GCGxLA, R7F100GCHxLA, R7F100GCJxLA
	40pin	R7F100GEFxNP, R7F100GEGxNP, R7F100GEHxNP, R7F100GEJxNP
	44pin	R7F100GFFxFP, R7F100GFGxFP, R7F100GFHxFP, R7F100GFJxFP, R7F100GFKxFP, R7F100GFLxFP, R7F100GFNxFP
	48pin	R7F100GGFxFB, R7F100GGGxFB, R7F100GGHxFB, R7F100GGJxFB, R7F100GGKxFB, R7F100GGLxFB, R7F100GGNxFB, R7F100GGFxNP, R7F100GGGxNP, R7F100GGHxNP, R7F100GGJxNP, R7F100GGKxNP, R7F100GGLxNP, R7F100GGNxNP
	52pin	R7F100GJFxFA, R7F100GJGxFA, R7F100GJHxFA, R7F100GJJxFA, R7F100GJKxFA, R7F100GJLxFA, R7F100GJNxFA
	64pin	R7F100GLFxFA, R7F100GLGxFA, R7F100GLHxFA, R7F100GLJxFA, R7F100GLKxFA, R7F100GLLxFA, R7F100GLNxFA, R7F100GLFxFB, R7F100GLGxFB, R7F100GLHxFB, R7F100GLJxFB, R7F100GLKxFB, R7F100GLLxFB, R7F100GLNxFB, R7F100GLFxLA, R7F100GLGxLA, R7F100GLHxLA, R7F100GLJxLA, R7F100GLKxLA, R7F100GLLxLA, R7F100GLNxLA
	80pin	R7F100GMGxFA, R7F100GMHxFA, R7F100GMJxFA, R7F100GMKxFA, R7F100GMLxFA, R7F100GMNxFA, R7F100GMGxFB, R7F100GMHxFB, R7F100GMJxFB, R7F100GMKxFB, R7F100GMLxFB, R7F100GMNxFB
	100pin	R7F100GPGxFB, R7F100GPHxFB, R7F100GPJxFB, R7F100GPKxFB, R7F100GPLxFB, R7F100GPNxFB, R7F100GPGxFA, R7F100GPHxFA, R7F100GPJxFA, R7F100GPKxFA, R7F100GPLxFA, R7F100GPNxFA
	128pin	R7F100GSJxFB, R7F100GSKxFB, R7F100GSLxFB, R7F100GSNxFB
RL78/F24 Group (R01UH0944EJ0100)	32pin	R7F124FBJ3xNP, R7F124FBJ4xNP, R7F124FBJ5xNP
	48pin	R7F124FGJ3xFB, R7F124FGJ4xFB, R7F124FGJ5xFB
	64pin	R7F124FLJ3xFB, R7F124FLJ4xFB, R7F124FLJ5xFB
	80pin	R7F124FMJ3xFB, R7F124FMJ4xFB, R7F124FMJ5xFB
	100pin	R7F124FPJ3xFB, R7F124FPJ4xFB, R7F124FPJ5xFB
RL78/G15 Group (R01UH0959EJ0100)	8pin	R5F12008xNS, R5F12007xNS, R5F12008xSN
	10pin	R5F12018xSP, R5F12017xSP
	16pin	R5F12048xNA, R5F12047xNA, R5F12048xSP, R5F12047xSP
	20pin	R5F12068xSP, R5F12067xSP
RL78/F23 Group (R01UH0944EJ0100)	32pin	R7F123FBG3xNP, R7F123FBG4xNP, R7F123FBG5xNP
	48pin	R7F123FGG3xFB, R7F123FGG4xFB, R7F123FGG5xFB
	64pin	R7F123FLG3xFB, R7F123FLG4xFB, R7F123FLG5xFB
	80pin	R7F123FMG3xFB, R7F123FMG4xFB, R7F123FMG5xFB
RL78/G22 Group (R01UH0978EJ0100)	16pin	R7F102G4ExNP, R7F102G4CxNP
	20pin	R7F102G6ExSP, R7F102G6CxSP
	24pin	R7F102G7ExNP, R7F102G7CxNP
	25pin	R7F102G8ExLA, R7F102G8CxLA
	30pin	R7F102GAExSP, R7F102GACxSP
	32pin	R7F102GBExNP, R7F102GBCxNP, R7F102GBExFP, R7F102GBCxFP
	36pin	R7F102GCExLA, R7F102GCCxLA
	40pin	R7F102GEExNP, R7F102GECxNP
	44pin	R7F102GFExFP, R7F102GFCxFP
	48pin	R7F102GGExFB, R7F102GGExNP, R7F102GGCxFB, R7F102GGCxNP

Table 2-2 Support Devices (2/2)

Group (HW Manual number)	PIN	Device name
RL78/G24 Group (R01UH0961EJ0100)	20pin	R7F101G6GxSP, R7F101G6ExSP
	24pin	R7F101G7GxNP, R7F101G7ExNP
	25pin	R7F101G8GxLA, R7F101G8ExLA
	30pin	R7F101GAGxSP, R7F101GAExSP
	32pin	R7F101GBGxNP, R7F101GBExNP, R7F101GBGxFP, R7F101GBExFP
	40pin	R7F101GEGxNP, R7F101GEExNP
	44pin	R7F101GFGxFP, R7F101GFExFP
	48pin	R7F101GGGxFB, R7F101GGExFB, R7F101GGGxNP, R7F101GGExNP
	52pin	R7F101GJGxFA, R7F101GJExFA
	64pin	R7F101GLGxFA, R7F101GLGxFB, R7F101GLExFA, R7F101GLExFB
RL78/G16 Group (R01UH0980EJ0100)	10pin	R5F1211AxSP, R5F1211CxSP
	16pin	R5F1214AxNA, R5F1214AxSP, R5F1214CxNA, R5F1214CxSP
	20pin	R5F1216AxSP, R5F1216CxSP
	24pin	R5F1217AxNA, R5F1217CxNA
	32pin	R5F121BAxFP, R5F121BAxNA, R5F121BCxFP, R5F121BCxNA
RL78/F25 Group (R01UH1061EJ0050)	48pin	R7F125FGL3xFB, R7F125FGL4xFB
	64pin	R7F125FLL3xFB, R7F125FLL4xFB
	80pin	R7F125FML3xFB, R7F125FML4xFB
	100pin	R7F125FPL3xFB, R7F125FPL4xFB
RL78/F22 Group (R01UH1061EJ0050)	24pin	R7F122F7G3xNP, R7F122F7G4xNP
	32pin	R7F122FBG3xNP, R7F122FBG4xNP
	48pin	R7F122FGG3xFB, R7F122FGG4xFB
RL78/L23 Group (R01UH1082EJ0100)	44pin	R7F100LFJxFP, R7F100LFLxFP, R7F100LFG, R7F100LFE
	48pin	R7F100LGJxFB, R7F100LGJxNP, R7F100LGLxFB, R7F100LGLxNP, R7F100LGG, R7F100LGE
	52pin	R7F100LJJxFA, R7F100LJLxFA, R7F100LJG, R7F100LJE
	64pin	R7F100LLJxFA, R7F100LLJxFB, R7F100LLLxFA, R7F100LLLxFB, R7F100LLG, R7F100LLE
	80pin	R7F100LMGxFA, R7F100LMGxFB, R7F100LMJxFA, R7F100LMJxFB, R7F100LMLxFA, R7F100LMLxFB
	100pin	R7F100LPGxFA, R7F100LPGxFB, R7F100LPJxFA, R7F100LPJxFB, R7F100LPLxFA, R7F100LPLxFB

2.2 Support Components List

Below is a list of Components supported by the Smart Configurator for RL78 V1.16.0.

Table 2-3 Support Components (1/3) ✓: Support, -: Non-support

No	Components	Mode	RL78/G23	RL78/F24	RL78/G15	RL78/F23	RL78/G22	RL78/G16	RL78/G24	RL78/F25	Remarks
1	12 Bit A/D Single Scan	-	-	✓	-	✓	-	-	-	✓	
2	12 Bit A/D Continuous Scan	-	-	✓	-	✓	-	-	-	✓	
3	12 Bit A/D Group Scan	-	-	✓	-	✓	-	-	-	✓	
4	A/D Converter	Normal mode	✓	-	✓	-	✓	✓	✓	-	Only RL78/G24 A/D converter has mode selection GUI. For other devices, the default mode is "Normal mode" and no GUI is provided for mode selection.
		Advanced mode	-	-	-	-	-	-	✓	-	
5	Clock Output/Buzzer Output Controller	-	✓	✓	✓	✓	✓	✓	✓	✓	
6	Comparator	-	✓	✓	✓	-	-	✓	✓	✓	
7	D/A Converter	-	✓	✓	-	-	-	-	✓	✓	
8	Data Transfer Controller	-	✓	✓	-	✓	✓	-	✓	✓	
9	Delay Counter	-	✓	✓	✓	✓	✓	✓	✓	✓	
10	Divider Function	-	✓	✓	✓	✓	✓	✓	✓	✓	
11	Event Link Controller	-	-	✓	-	-	✓	-	✓	✓	
12	External Event Counter	-	✓	✓	✓	✓	✓	✓	✓	✓	
13	IIC Communication (Master mode)	-	✓	✓	✓	✓	✓	✓	✓	✓	
14	IIC Communication (Slave mode)	-	✓	✓	✓	✓	✓	✓	✓	✓	
15	Input Capture Function	-	-	✓	-	✓	-	-	✓	✓	
16	Input Pulse Interval/Period Measurement	-	✓	✓	✓	✓	✓	✓	✓	✓	
17	Input Signal High-/Low-Level Width Measurement	-	✓	✓	✓	✓	✓	✓	✓	✓	
18	Interrupt Controller	-	✓	✓	✓	✓	✓	✓	✓	✓	
19	Interval Timer	8 bit count mode	✓	✓	✓	✓	✓	✓	✓	✓	
		12 bit count mode	-	-	✓	-	-	✓	-	-	
		16 bit count mode	✓	✓	✓	✓	✓	✓	✓	✓	
		16 bit capture mode	✓	-	-	-	✓	-	✓	-	
		32 bit count mode	✓	-	-	-	✓	-	✓	-	
20	Key Interrupt	-	✓	✓	-	✓	✓	-	✓	✓	
21	One-Shot Pulse Output	One-Shot Pulse Output	✓	✓	✓	✓	✓	✓	✓	✓	
		Two-Channel Input with One-Shot Pulse Output	-	-	✓	-	-	✓	-	-	
22	Output Compare Function	-	-	✓	-	✓	-	-	✓	✓	
23	Ports	-	✓	✓	✓	✓	✓	✓	✓	✓	
24	PWM Option Unit A	-	-	✓	-	✓	-	-	✓	✓	
25	DALI Communication (Control devices)	-	-	-	-	-	-	-	✓	-	
26	DALI Communication (Control gear)	-	-	-	-	-	-	-	✓	-	
27	Real-Time Clock	-	✓	✓	-	✓	✓	✓	✓	✓	

Table 2-4 Support Components (2/3) ✓: Support, -: Non-support

No	Components	Mode	RL78/G23	RL78/F24	RL78/G15	RL78/F23	RL78/G22	RL78/G16	RL78/G24	RL78/F25	Remarks
28	PWM Output	PWM Mode	✓	✓	✓	✓	✓	✓	✓	✓	
		PWM3 Mode	-	✓	-	✓	-	-	✓	✓	
		Extended PWM Mode	-	✓	-	✓	-	-	✓	✓	
		PWM2 Mode	-	-	-	-	-	-	✓	-	
		Timer KB3 PWM Output Gate Mode	-	-	-	-	-	-	✓	-	
		Standalone Mode (Period controlled by the TKBCRn0 register)	-	-	-	-	-	-	✓	-	
		Standalone Mode (Period controlled by external trigger input)	-	-	-	-	-	-	✓	-	
		Simultaneous Start/Stop Mode (Period controlled by the TKBCRn0 register)	-	-	-	-	-	-	✓	-	
		Simultaneous Start/Stop Mode (Period controlled by external trigger input)	-	-	-	-	-	-	✓	-	
		Simultaneous Start/Clear Mode (Period controlled by master)	-	-	-	-	-	-	✓	-	
		Interleaved PFC Output Mode	-	-	-	-	-	-	✓	-	
		Remote Control Function	-	-	-	-	-	-	-	-	
29	Remote Control Signal Receiver	-	✓	-	-	-	-	-	-	-	
30	SNOOZE Mode Sequencer	-	✓	-	-	-	✓	-	-	-	
31	SPI (CSI) Communication	Transmission	✓	✓	✓	✓	✓	✓	✓	✓	
		Reception	✓	✓	✓	✓	✓	✓	✓	✓	
		Transmission/reception	✓	✓	✓	✓	✓	✓	✓	✓	
32	Square Wave Output	-	✓	✓	✓	✓	✓	✓	✓	✓	
33	Three-phase PWM Output	Reset Synchronous PWM Mode	-	✓	-	✓	-	-	✓	✓	
		Complementary PWM Mode	-	✓	-	✓	-	-	✓	✓	
		Extended Complementary PWM Mode	-	✓	-	✓	-	-	✓	✓	
34	UART Communication	Transmission	✓	✓	✓	✓	✓	✓	✓	✓	
		Reception	✓	✓	✓	✓	✓	✓	✓	✓	
		Transmission/reception	✓	✓	✓	✓	✓	✓	✓	✓	
35	Voltage Detector	-	✓	✓	−	✓	✓	−	✓	✓	
36	Watchdog Timer	-	✓	✓	✓	✓	✓	✓	✓	✓	
37	Logic & Event Link Controller	-	✓	-	-	-	-	-	-	-	To use ELCL modules of fixed function, need download in Smart Configurator RL78.
38	Phase Counting Mode	-	-	-	-	-	-	-	✓	-	
39	Programmable Gain Amplifier	-	-	-	-	-	-	-	✓	-	
40	Flexible Application Accelerator	-	-	-	-	-	-	-	✓	-	

Table 2-5 Support Components (3/3) ✓: Support, -: Non-support

No	Components	Mode	RL78/G23	RL78/F24	RL78/G15	RL78/F23	RL78/G22	RL78/G16	RL78/G24	RL78/F25	Remarks
41	LCD Controller / Driver	-	-	-	-	-	-	-	-	-	
42	Oscillation Stop Detector	-	-	-	-	-	-	-	-	-	
43	External Signal Sampler	-	-	-	-	-	-	-	-	-	

Table 2-6 Support Components (1/3) ✓: Support, -: Non-support

No	Components	Mode	RL78/F22	RL78/L23	Remarks
1	12 Bit A/D Single Scan	-	✓	-	
2	12 Bit A/D Continuous Scan	-	✓	-	
3	12 Bit A/D Group Scan	-	✓	-	
4	A/D Converter	Normal mode	-	✓	Only RL78/G24 A/D converter has mode selection GUI. For other devices, the default mode is "Normal mode" and no GUI is provided for mode selection.
		Advanced mode	-	-	
5	Clock Output/Buzzer Output Controller	-	✓	✓	
6	Comparator	-	-	✓	
7	D/A Converter	-	-	✓	
8	Data Transfer Controller	-	✓	✓	
9	Delay Counter	-	✓	✓	
10	Divider Function	-	✓	✓	
11	Event Link Controller	-	-	-	
12	External Event Counter	-	✓	✓	
13	IIC Communication (Master mode)	-	✓	✓	
14	IIC Communication (Slave mode)	-	✓	✓	
15	Input Capture Function	-	✓	-	
16	Input Pulse Interval/Period Measurement	-	✓	✓	
17	Input Signal High-/Low-Level Width Measurement	-	✓	✓	
18	Interrupt Controller	-	✓	✓	
19	Interval Timer	8 bit count mode	✓	✓	
		12 bit count mode	-	-	
		16 bit count mode	✓	✓	
		16 bit capture mode	-	✓	
		32 bit count mode	-	✓	
20	Key Interrupt	-	✓	✓	
21	One-Shot Pulse Output	One-Shot Pulse Output	✓	✓	
		Two-Channel Input with One-Shot Pulse Output	-	-	
22	Output Compare Function	-	✓	-	
23	Ports	-	✓	✓	
24	PWM Option Unit A	-	✓	-	
25	DALI Communication (Control devices)	-	-	-	
26	DALI Communication (Control gear)	-	-	-	
27	Real-Time Clock	-	✓	✓	

Table 2-7 Support Components (2/3) ✓: Support, -: Non-support

No	Components	Mode	RL78/F22	RL78/L23	Remarks
28	PWM Output	PWM Mode	✓	✓	
		PWM3 Mode	✓	-	
		Extended PWM Mode	✓	-	
		PWM2 Mode	-	-	
		Timer KB3 PWM Output Gate Mode	-	-	
		Standalone Mode (Period controlled by the TKBCRn0 register)	-	✓	
		Standalone Mode (Period controlled by external trigger input)	-	✓	
		Simultaneous Start/Stop Mode (Period controlled by the TKBCRn0 register)	-	✓	
		Simultaneous Start/Stop Mode (Period controlled by external trigger input)	-	✓	
		Simultaneous Start/Clear Mode (Period controlled by master)	-	✓	
		Interleaved PFC Output Mode	-	✓	
		Remote Control Function	-	✓	
29	Remote Control Signal Receiver	-	-	-	
30	SNOOZE Mode Sequencer	-	-	✓	
31	SPI (CSI) Communication	Transmission	✓	✓	
		Reception	✓	✓	
		Transmission/reception	✓	✓	
32	Square Wave Output	-	✓	✓	
33	Three-phase PWM Output	Reset Synchronous PWM Mode	✓	-	
		Complementary PWM Mode	✓	-	
		Extended Complementary PWM Mode	✓	-	
34	UART Communication	Transmission	✓	✓	
		Reception	✓	✓	
		Transmission/reception	✓	✓	
35	Voltage Detector	-	✓	✓	
36	Watchdog Timer	-	✓	✓	
37	Logic & Event Link Controller	-	-	✓	To use ELCL modules of fixed function, need download in Smart Configurator RL78.
38	Phase Counting Mode	-	-	-	
39	Programmable Gain Amplifier	-	-	-	
40	Flexible Application Accelerator	-	-	-	

Table 2-8 Support Components (3/3) ✓: Support, -: Non-support

No	Components	Mode	RL78/F22	RL78/L23	Remarks
41	LCD Controller / Driver	-	-	✓	
42	Oscillation Stop Detector	-	-	✓	
43	External Signal Sampler	-	-	✓	

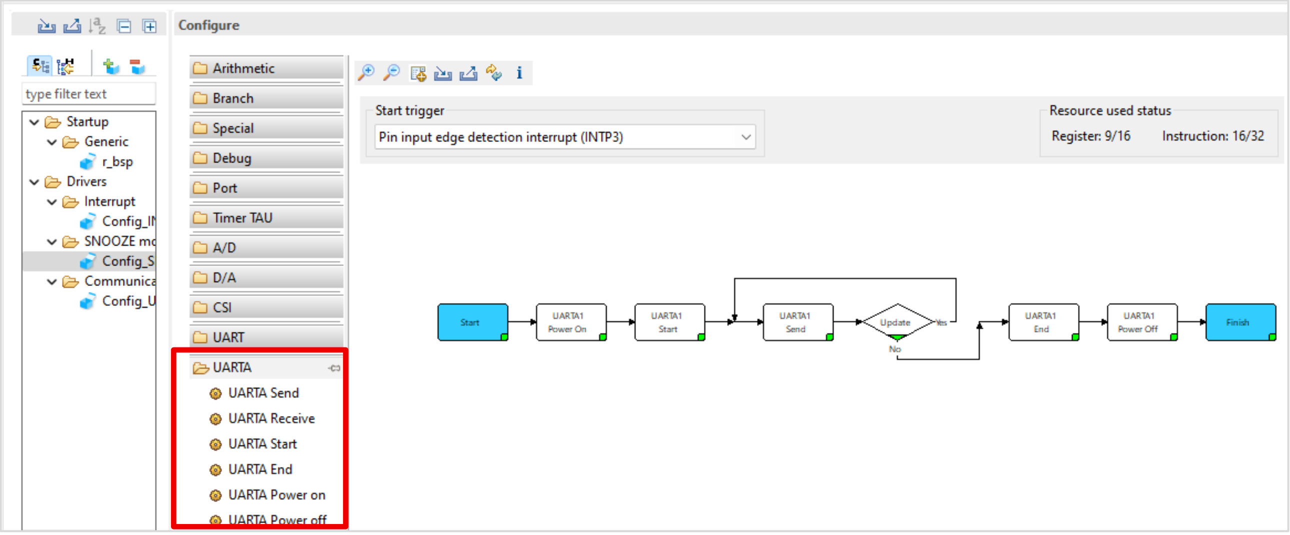
2.3 New support

2.3.1 BSP (Board Support Package) revision update

BSP rev1.92 is supported and will be added as default BSP when creating Smart Configurator project.

2.3.2 SMS support UARTA

From Smart Configurator for RL78 V1.16.0, the Graphical Configurator component: Snooze Mode Sequencer(SMS) supported UARTA blocks function, the UARTA blocks include 6: UARTA Send, UARTA Receive, UARTA Start, UARTA End, UARTA Power on and UARTA Power off. User can use these blocks by Drag&Drop to SMS canvas to implement UARTA data transfer with CPU placed in the standby state.



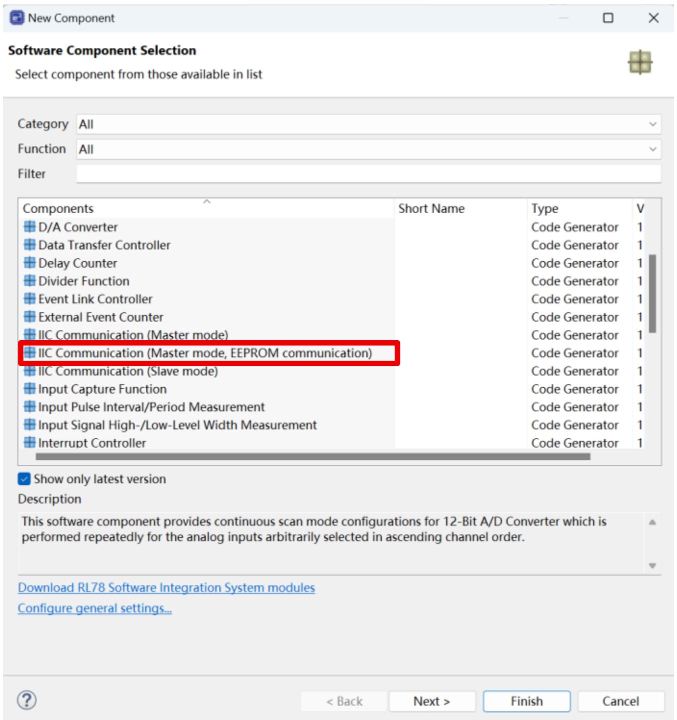
Start trigger: Pin input edge detection interrupt (INTP3)

Resource used status: Register: 9/16 Instruction: 16/32

Figure 2-1 UARTA blocks supported in SMS panel

2.3.3 Support IIC Communication (Master mode, EEPROM communication) component

From Smart Configurator for RL78 V1.16.0, the user can select IIC Communication (Master mode, EEPROM communication) component. It is a clocked communication function (Master mode) to communicate with EEPROM by using two lines: serial clock (SCL) and serial data (SDA). And it supports 1-byte memory addresses and issuing restart conditions.



Software Component Selection

Select component from those available in list

Category: All
Function: All
Filter:

Components	Short Name	Type	V
D/A Converter		Code Generator	1
Data Transfer Controller		Code Generator	1
Delay Counter		Code Generator	1
Divider Function		Code Generator	1
Event Link Controller		Code Generator	1
External Event Counter		Code Generator	1
IIC Communication (Master mode)		Code Generator	1
IIC Communication (Master mode, EEPROM communication)		Code Generator	1
IIC Communication (Slave mode)		Code Generator	1
Input Capture Function		Code Generator	1
Input Pulse Interval/Period Measurement		Code Generator	1
Input Signal High-/Low-Level Width Measurement		Code Generator	1
Interrupt Controller		Code Generator	1

☑ Show only latest version

Description

This software component provides continuous scan mode configurations for 12-Bit A/D Converter which is performed repeatedly for the analog inputs arbitrarily selected in ascending channel order.

Download RL78 Software Integration System modules

Configure general settings...

Figure 2-2 IIC Communication (Master mode, EEPROM communication) component

3. Changes

This chapter describes changes to the Smart Configurator for RL78 V1.16.0.

3.1 Correction of issues/limitations

Table 3-1 List of Correction of issues/limitations (1/2) ✓: Applicable, -: Not Applicable

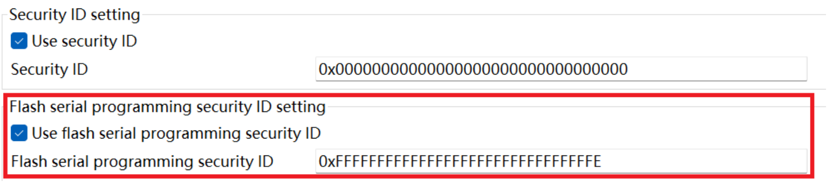
No	Description	RL78/G23	RL78/F24	RL78/G15	RL78/F23	RL78/G22	RL78/G16	RL78/G24	RL78/F25	Remarks
1	Fixed the issue of not supporting flash serial programming security ID	–	✓	–	✓	–	–	–	✓	
2	Fixed the issue of not supporting "Does not use trigger input" in TMKB - Standalone Mode (Period controlled by external trigger edge)	–	–	–	–	–	–	–	–	
3	Fixed the errors in DTC	✓	✓	–	✓	✓	–	✓	✓	
4	Fixed the issue of only fIH and fIM can be selected as fCLK in low-power main mode	–	–	–	–	–	–	–	–	
5	Fixed the issue of not supporting all address matching function in IICA	✓	–	–	–	✓	–	✓	–	

Table 3-2 List of Correction of issues/limitations (2/2) ✓: Applicable, -: Not Applicable

No	Description	RL78/F22	RL78/L23	Remarks
1	Fixed the issue of not supporting flash serial programming security ID	✓	–	
2	Fixed the issue of not supporting "Does not use trigger input" in TMKB - Standalone Mode (Period controlled by external trigger edge)	–	✓	
3	Fixed the errors in DTC	✓	✓	
4	Fixed the issue of only fIH and fIM can be selected as fCLK in low-power main mode	–	✓	
5	Fixed the issue of not supporting all address matching function in IICA	–	✓	

3.1.1 Fixed the issue of not supporting flash serial programming security ID

From Smart Configurator for RL78 V1.16.0, “flash serial programming security ID” is supported.



Security ID setting

☑ Use security ID

Security ID 0x00000000000000000000000000000000

Flash serial programming security ID setting

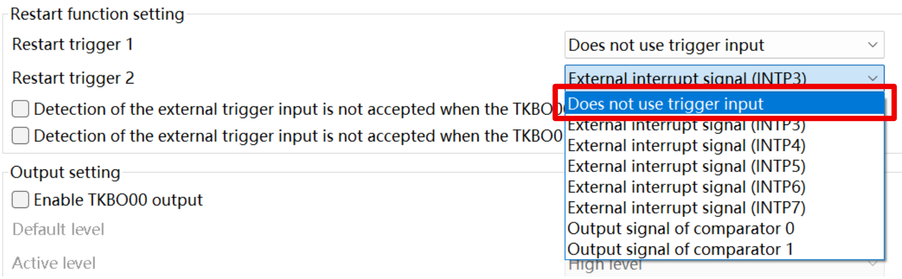
☑ Use flash serial programming security ID

Flash serial programming security ID 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE

Figure 3-1 “flash serial programming security ID” in [System] page

3.1.2 Fixed the issue of not supporting "Does not use trigger input" in TMKB - Standalone Mode (Period controlled by external trigger edge)

From Smart Configurator for RL78 V1.16.0, "Does not use trigger input" is supported for “Restart trigger 2” in TMKB - Standalone Mode (Period controlled by external trigger edge).



Restart function setting
Restart trigger 1: Does not use trigger input
Restart trigger 2: External interrupt signal (INTP3)
- Does not use trigger input
- External interrupt signal (INTP3)
- External interrupt signal (INTP4)
- External interrupt signal (INTP5)
- External interrupt signal (INTP6)
- External interrupt signal (INTP7)
- Output signal of comparator 0
- Output signal of comparator 1

□ Detection of the external trigger input is not accepted when the TKBO0
□ Detection of the external trigger input is not accepted when the TKBO0

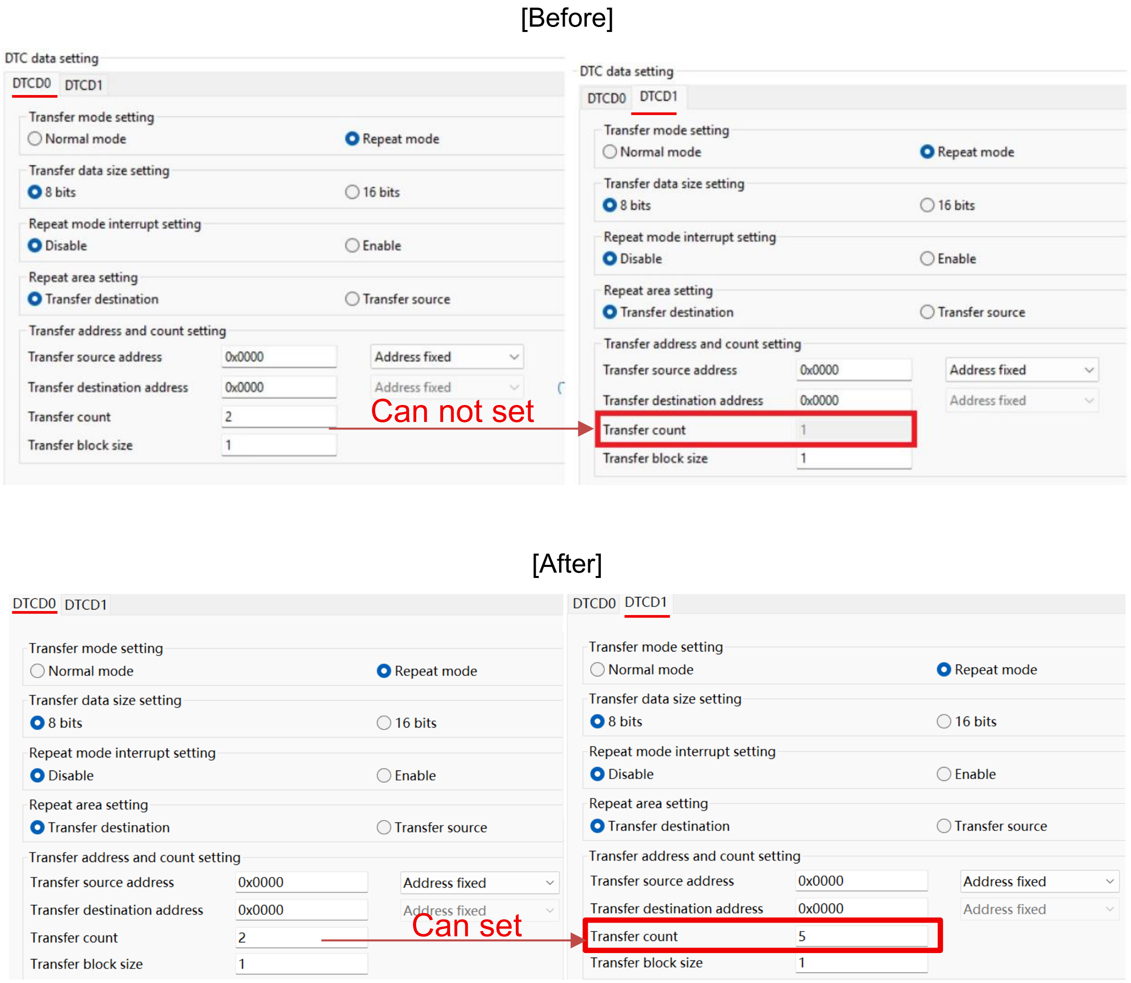
Output setting
□ Enable TKBO00 output
Default level
Active level: High level

Figure 3-2 Add "Does not use trigger input" in "Restart trigger 2"

3.1.3 Fixed the errors in DTC

From Smart Configurator for RL78 V1.16.0, the following errors in DTC are fixed.

- In chain transfer repeat mode, it is necessary to set the number of transfers.



[Before]

DTC data setting — DTCD0

Transfer mode setting: ○ Normal mode ● Repeat mode
Transfer data size setting: ● 8 bits ○ 16 bits
Repeat mode interrupt setting: ● Disable ○ Enable
Repeat area setting: ● Transfer destination ○ Transfer source
Transfer address and count setting
Transfer source address: 0x0000 Address fixed
Transfer destination address: 0x0000 Address fixed
Transfer count: 2
Transfer block size: 1

DTC data setting — DTCD1

Transfer mode setting: ○ Normal mode ● Repeat mode
Transfer data size setting: ● 8 bits ○ 16 bits
Repeat mode interrupt setting: ● Disable ○ Enable
Repeat area setting: ● Transfer destination ○ Transfer source
Transfer address and count setting
Transfer source address: 0x0000 Address fixed
Transfer destination address: 0x0000 Address fixed
Transfer count: 1
Transfer block size: 1

Can not set

[After]

DTCD0

Transfer mode setting: ○ Normal mode ● Repeat mode
Transfer data size setting: ● 8 bits ○ 16 bits
Repeat mode interrupt setting: ● Disable ○ Enable
Repeat area setting: ● Transfer destination ○ Transfer source
Transfer address and count setting
Transfer source address: 0x0000 Address fixed
Transfer destination address: 0x0000 Address fixed
Transfer count: 2
Transfer block size: 1

DTCD1

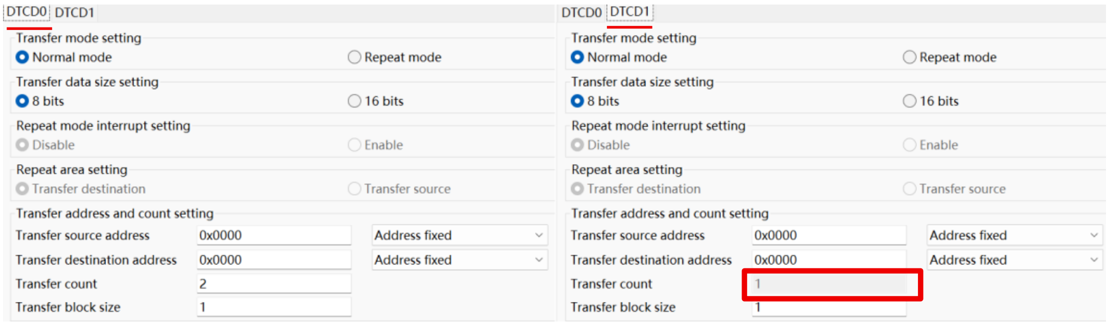
Transfer mode setting: ○ Normal mode ● Repeat mode
Transfer data size setting: ● 8 bits ○ 16 bits
Repeat mode interrupt setting: ● Disable ○ Enable
Repeat area setting: ● Transfer destination ○ Transfer source
Transfer address and count setting
Transfer source address: 0x0000 Address fixed
Transfer destination address: 0x0000 Address fixed
Transfer count: 5
Transfer block size: 1

Can set

Figure 3-3 Transfer count in chain transfer repeat mode

- In chain transfer normal mode, set the number of transfers from the transfer source to DTCCT*x* for DTC chain transfer. (*x* means the number of control data.)

[Before]



DTCD0 DTCD1

Transfer mode setting
● Normal mode ○ Repeat mode

Transfer data size setting
● 8 bits ○ 16 bits

Repeat mode interrupt setting
● Disable ○ Enable

Repeat area setting
● Transfer destination ○ Transfer source

Transfer address and count setting

Transfer source address	0x0000	Address fixed
Transfer destination address	0x0000	Address fixed
Transfer count	2	
Transfer block size	1	

DTCD0 DTCD1

Transfer mode setting
● Normal mode ○ Repeat mode

Transfer data size setting
● 8 bits ○ 16 bits

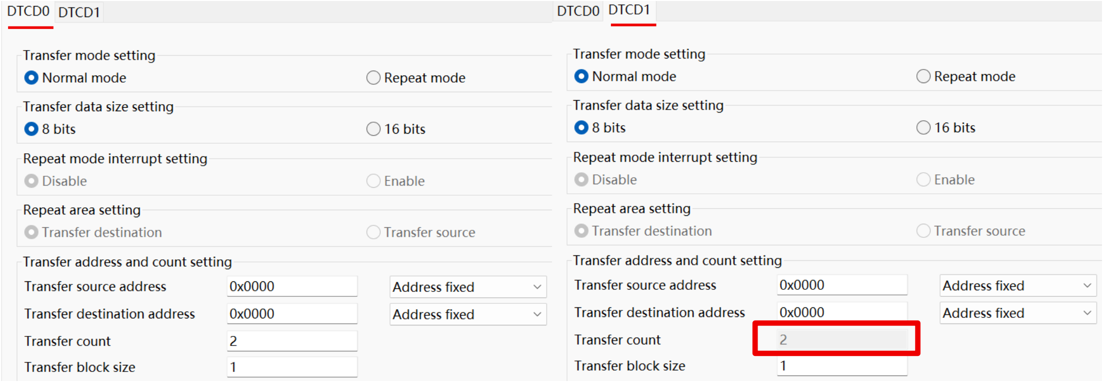
Repeat mode interrupt setting
● Disable ○ Enable

Repeat area setting
● Transfer destination ○ Transfer source

Transfer address and count setting

Transfer source address	0x0000	Address fixed
Transfer destination address	0x0000	Address fixed
Transfer count	1	
Transfer block size	1	

[After]



DTCD0 DTCD1

Transfer mode setting
● Normal mode ○ Repeat mode

Transfer data size setting
● 8 bits ○ 16 bits

Repeat mode interrupt setting
● Disable ○ Enable

Repeat area setting
● Transfer destination ○ Transfer source

Transfer address and count setting

Transfer source address	0x0000	Address fixed
Transfer destination address	0x0000	Address fixed
Transfer count	2	
Transfer block size	1	

DTCD0 DTCD1

Transfer mode setting
● Normal mode ○ Repeat mode

Transfer data size setting
● 8 bits ○ 16 bits

Repeat mode interrupt setting
● Disable ○ Enable

Repeat area setting
● Transfer destination ○ Transfer source

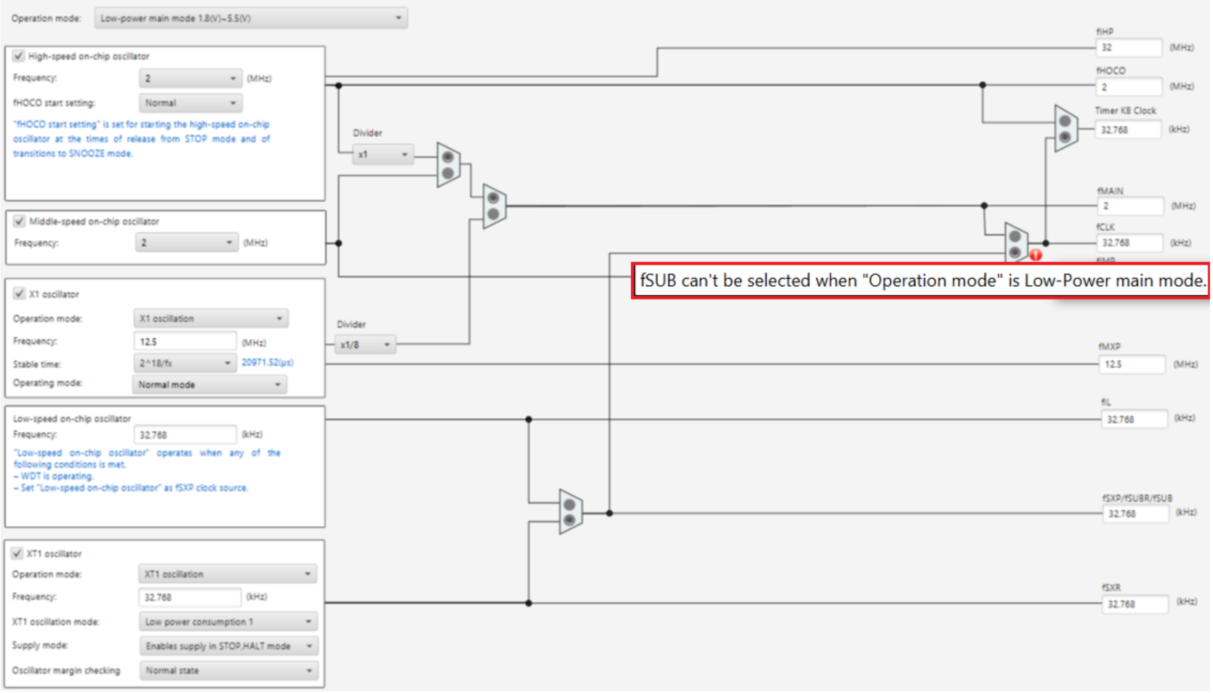
Transfer address and count setting

Transfer source address	0x0000	Address fixed
Transfer destination address	0x0000	Address fixed
Transfer count	2	
Transfer block size	1	

Figure 3-4 Transfer count in chain transfer normal mode

3.1.4 Fixed the issue of only fIH and fIM can be selected as fCLK in low-power main mode

When selecting low-power main mode, the user can only use fIH and fIM as fCLK in [Clocks] page. If the user selects other clock source, it will dispear error icon and message.



Operation mode: Low-power main mode 1.8(V)~5.5(V)

☑ High-speed on-chip oscillator
Frequency: 2 (MHz)
fHOCO start setting: Normal
"fHOCO start setting" is set for starting the high-speed on-chip oscillator at the times of release from STOP mode and of transitions to SNOOZE mode.

☑ Middle-speed on-chip oscillator
Frequency: 2 (MHz)

☑ X1 oscillator
Operation mode: X1 oscillation
Frequency: 12.5 (MHz)
Stable time: 2^18/fx 20971.52(μs)
Operating mode: Normal mode

Low-speed on-chip oscillator
Frequency: 32.768 (kHz)
"Low-speed on-chip oscillator" operates when any of the following conditions is met.
- WDT is operating.
- Set "Low-speed on-chip oscillator" as fSXP clock source.

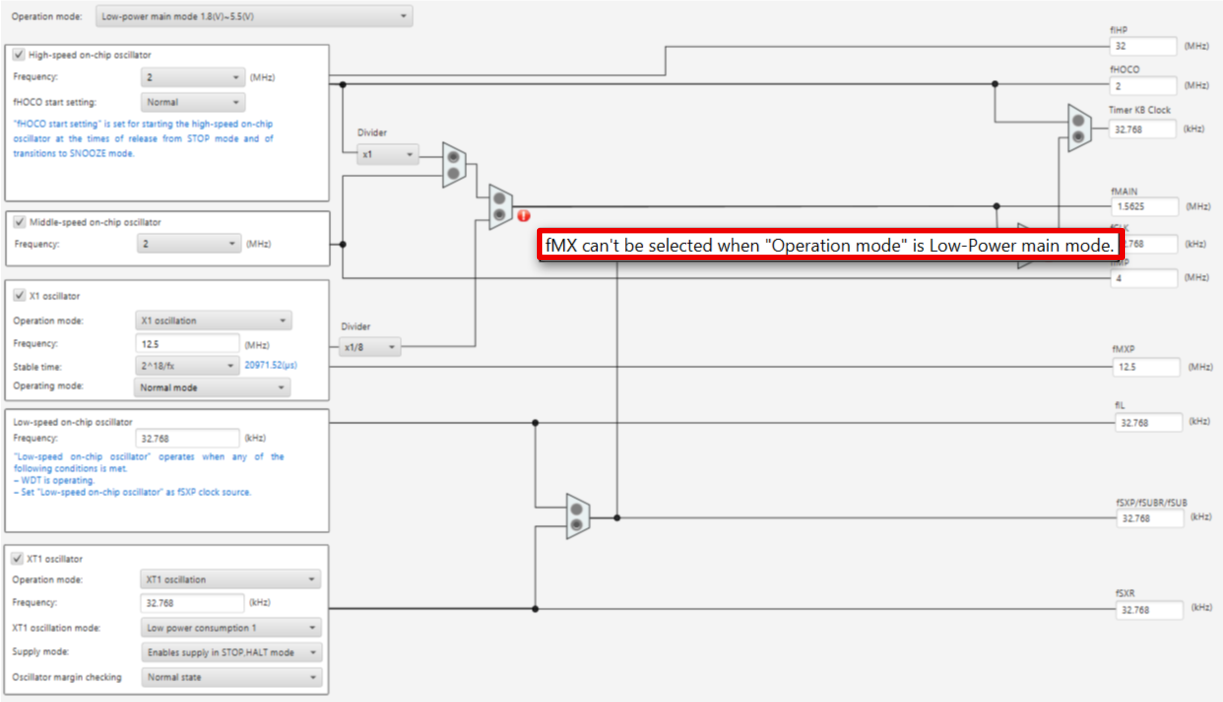
☑ XT1 oscillator
Operation mode: XT1 oscillation
Frequency: 32.768 (kHz)
XT1 oscillation mode: Low power consumption 1
Supply mode: Enables supply in STOP,HALT mode
Oscillator margin checking: Normal state

Divider: x1
Divider: x1/8

fIHP: 32 (MHz)
fHOCO: 2 (MHz)
Timer KB Clock: 32.768 (kHz)
fMAIN: 2 (MHz)
fCLK: 32.768 (kHz)
fMXP: 12.5 (MHz)
fIL: 32.768 (kHz)
fSXP/fSUBR/fSUB: 32.768 (kHz)
fSXR: 32.768 (kHz)

fSUB can't be selected when "Operation mode" is Low-Power main mode.

Figure 3-5 Select fSUB as fCLK in low-power main mode



Operation mode: Low-power main mode 1.8(V)~5.5(V)

☑ High-speed on-chip oscillator
Frequency: 2 (MHz)
fHOCO start setting: Normal
"fHOCO start setting" is set for starting the high-speed on-chip oscillator at the times of release from STOP mode and of transitions to SNOOZE mode.

☑ Middle-speed on-chip oscillator
Frequency: 2 (MHz)

☑ X1 oscillator
Operation mode: X1 oscillation
Frequency: 12.5 (MHz)
Stable time: 2^18/fx 20971.52(μs)
Operating mode: Normal mode

Low-speed on-chip oscillator
Frequency: 32.768 (kHz)
"Low-speed on-chip oscillator" operates when any of the following conditions is met.
- WDT is operating.
- Set "Low-speed on-chip oscillator" as fSXP clock source.

☑ XT1 oscillator
Operation mode: XT1 oscillation
Frequency: 32.768 (kHz)
XT1 oscillation mode: Low power consumption 1
Supply mode: Enables supply in STOP,HALT mode
Oscillator margin checking: Normal state

Divider: x1
Divider: x1/8

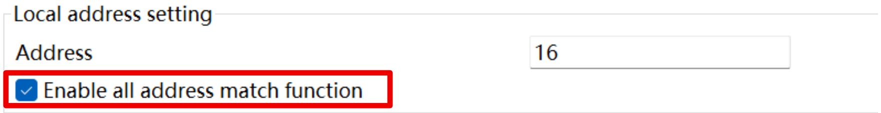
fIHP: 32 (MHz)
fHOCO: 2 (MHz)
Timer KB Clock: 32.768 (kHz)
fMAIN: 1.5625 (MHz)
768 (kHz)
4 (MHz)
fMXP: 12.5 (MHz)
fIL: 32.768 (kHz)
fSXP/fSUBR/fSUB: 32.768 (kHz)
fSXR: 32.768 (kHz)

fMX can't be selected when "Operation mode" is Low-Power main mode.

Figure 3-6 Select fMX as fCLK in low-power main mode

3.1.5 Fixed the issue of not supporting all address matching function in IICA

From Smart Configurator for RL78 V1.16.0, Smart Configurator supports all address matching function in IICA (slave mode). When the user selects “Enable all address match function”, IICA (slave mode) considers any address as address match, and performs the same operation as that when an extension code is received.



Local address setting

Address: 16

☑ Enable all address match function

Figure 3-7 Support “all address matching function”

3.2 Specification changes

Table 3-3 List of Specification changes (1/2) ✓: Applicable, -: Not Applicable

No	Description	RL78/G23	RL78/F24	RL78/G15	RL78/F23	RL78/G22	RL78/G16	RL78/G24	RL78/F25	Remarks
1	Improvement for updating the driver code in IIC Communication (Master mode) component	✓	✓	✓	✓	✓	✓	✓	✓	
2	Improvement for updating the spec of [Clocks] page	✓	✓	✓	✓	✓	✓	✓	✓	
3	Improvement for reducing power consumption	✓	✓	✓	✓	✓	✓	✓	✓	
4	Improvement for updating the default status of the backup feature	✓	✓	✓	✓	✓	✓	✓	✓	

Table 3-4 List of Specification changes (2/2) ✓: Applicable, -: Not Applicable

No	Description	RL78/F22	RL78/L23	Remarks
1	Improvement for updating the driver code in IIC Communication (Master mode) component	✓	✓	
2	Improvement for updating the spec of [Clocks] page	✓	✓	
3	Improvement for reducing power consumption	✓	✓	
4	Improvement for updating the default status of the backup feature	✓	✓	

3.2.1 Improvement for updating the driver code in IIC Communication (Master mode) component

Smart Configurator updates the specification of driver code in IIC Communication (Master mode) component. Make it easier for the beginners to use the serial interface IICA. The detailed updated points:

・Prevent data corruption during communication.
・Add a function to wait for the completion of communication.
・Add a process to confirm that the stop condition has been issued successfully.

Please refer to the User's Manual: RL78 API Reference for usage and samples.

Table 3-5 List of API

API Name	Status
R_{Config_IICA*n*}_Create	Changed
R_{Config_IICA*n*}_StopCondition	Changed
R_{Config_IICA*n*}_Stop	No change
R_{Config_IICA*n*}_Master_Send	Changed
R_{Config_IICA*n*}_Master_Receive	Changed
R_{Config_IICA*n*}_Create_UserInit	No change
r_{Config_IICA*n*}_interrupt	No change
r_{Config_IICA*n*}_master_handler	Changed
r_{Config_IICA*n*}_callback_master_sendend	No chang
r_{Config_IICA*n*}_callback_master_receiveend	No chang
r_{Config_IICA*n*}_callback_master_error	No chang
R_{Config_IICA*n*}_Check_Comstate	New addition
R_{Config_IICA*n*}_Poll	New addition
R_{Config_IICA*n*}_Wait_Comend	New addition
R_{Config_IICA*n*}_Bus_Check	New addition
R_{Config_IICA*n*}_StartCondition	New addition
R_{Config_IICA*n*}_Wait_Time	New addition

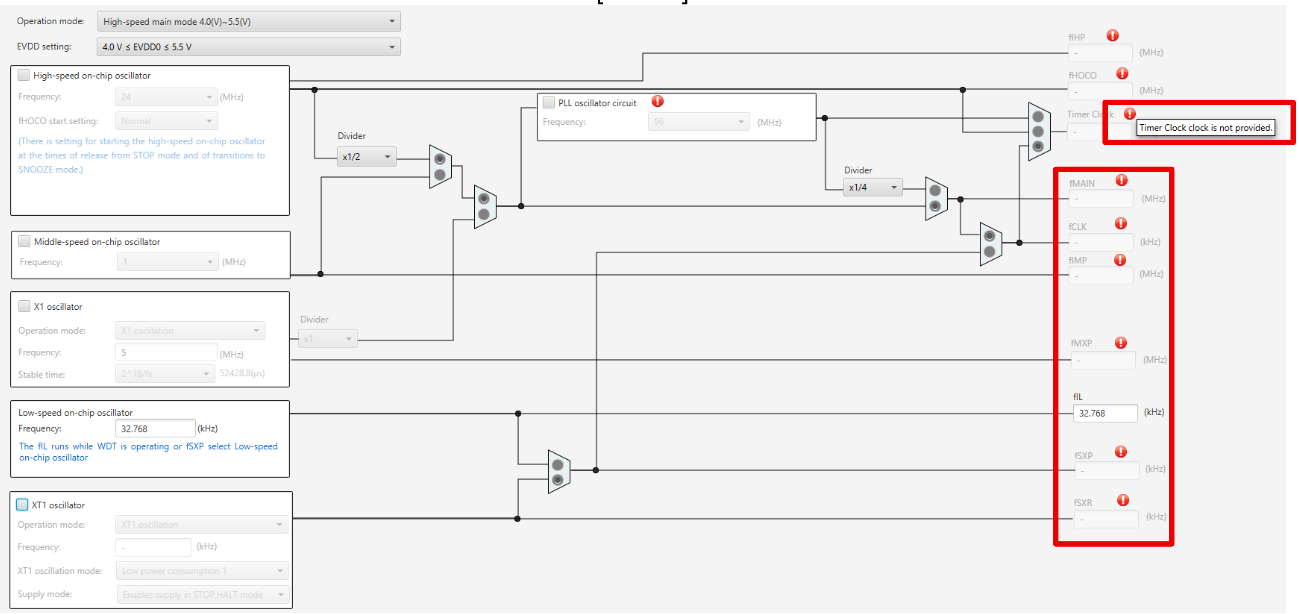
3.2.2 Improvement for updating the spec of [Clocks] page

Smart Configurator updates the spec of Clock page. The updated points:

- Using red error icon (❗) when there is an error in Clock page which must solve.
- Remove red error icon if the clock which mustn't be used is unused.
- Clock name should be same as User's Manual. Easy to find out in Clock page.
- Update the message of error icon. Make it easy to understand how to solve the error.
- The red error icon location should be easy to understand how to solve the error.

For example:
1. When clocks are unused, remove the red error icon behind the clock name.
2. Update the message of the error icon. Make it easy to understand how to solve the error.

[Before]



Operation mode: High-speed main mode 4.0(V)~5.5(V)
EVDD setting: 4.0 V ≤ EVDD0 ≤ 5.5 V

□ High-speed on-chip oscillator
Frequency: 24 (MHz)
fHOCO start setting: Normal
(There is setting for starting the high-speed on-chip oscillator at the times of release from STOP mode and of transitions to SNOOZE mode.)

□ Middle-speed on-chip oscillator
Frequency: 1 (MHz)

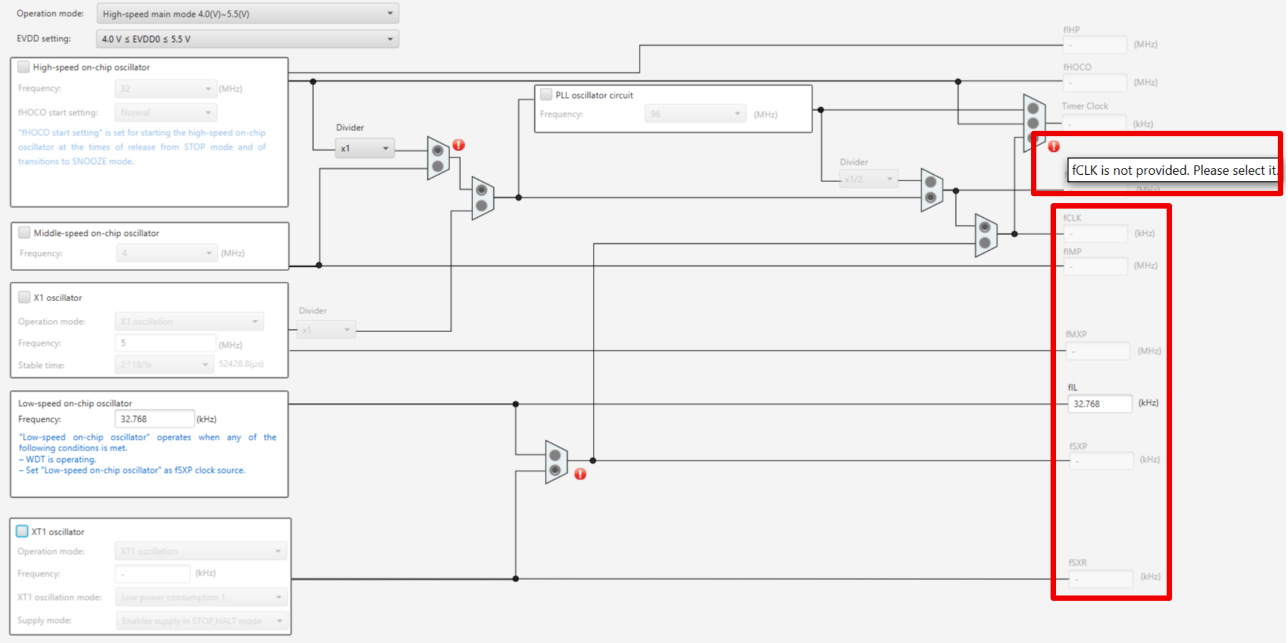
□ X1 oscillator
Operation mode: X1 oscillation
Frequency: 5 (MHz)
Stable time: 2^18/fx 52428.8(μs)

Low-speed on-chip oscillator
Frequency: 32.768 (kHz)
The fIL runs while WDT is operating or fSXP select Low-speed on-chip oscillator

□ XT1 oscillator
Operation mode: XT1 oscillation
Frequency: - (kHz)
XT1 oscillation mode: Low power consumption 1
Supply mode: Enables supply in STOP,HALT mode

Timer Clock clock is not provided.

[After]



Operation mode: High-speed main mode 4.0(V)~5.5(V)
EVDD setting: 4.0 V ≤ EVDD0 ≤ 5.5 V

□ High-speed on-chip oscillator
Frequency: 32 (MHz)
fHOCO start setting: Normal
"fHOCO start setting" is set for starting the high-speed on-chip oscillator at the times of release from STOP mode and of transitions to SNOOZE mode.

□ Middle-speed on-chip oscillator
Frequency: 4 (MHz)

□ X1 oscillator
Operation mode: X1 oscillation
Frequency: 5 (MHz)
Stable time: 2^18/fx 52428.8(μs)

Low-speed on-chip oscillator
Frequency: 32.768 (kHz)
"Low-speed on-chip oscillator" operates when any of the following conditions is met.
– WDT is operating.
– Set "Low-speed on-chip oscillator" as fSXP clock source.

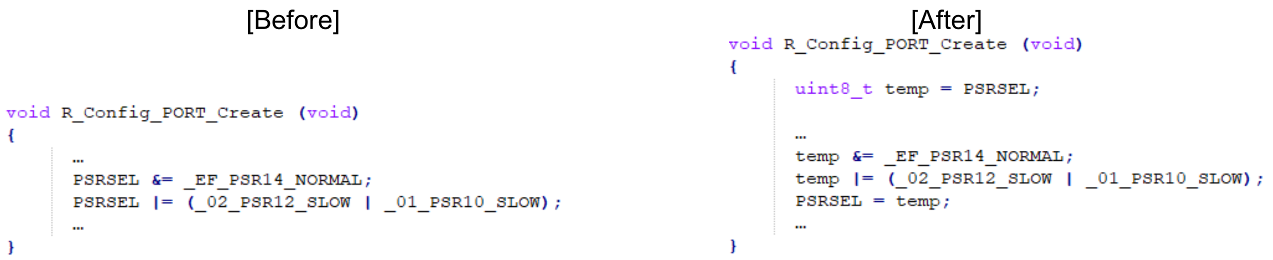
□ XT1 oscillator
Operation mode: XT1 oscillation
Frequency: - (kHz)
XT1 oscillation mode: Low power consumption 1
Supply mode: Enables supply in STOP,HALT mode

fCLK is not provided. Please select it.

Figure 3-8 Update the spec of [Clocks] page

3.2.3 Improvement for reducing power consumption

Smart Configurator updates the generated driver code to reduce power consumption. The following are some examples.



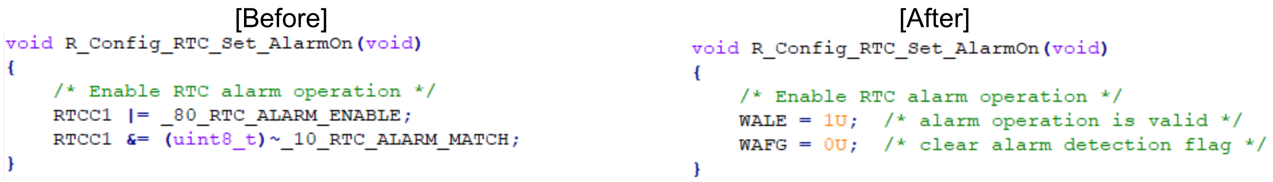
[Before]

```
void R_Config_PORT_Create (void)
{
        …
        PSRSEL &= _EF_PSR14_NORMAL;
        PSRSEL |= (_02_PSR12_SLOW | _01_PSR10_SLOW);
        …
}
```

[After]

```
void R_Config_PORT_Create (void)
{
        uint8_t temp = PSRSEL;

        …
        temp &= _EF_PSR14_NORMAL;
        temp |= (_02_PSR12_SLOW | _01_PSR10_SLOW);
        PSRSEL = temp;
        …
}
```

Figure 3-9 Add temp



[Before]

```
void R_Config_RTC_Set_AlarmOn(void)
{
    /* Enable RTC alarm operation */
    RTCC1 |= _80_RTC_ALARM_ENABLE;
    RTCC1 &= (uint8_t)~_10_RTC_ALARM_MATCH;
}
```

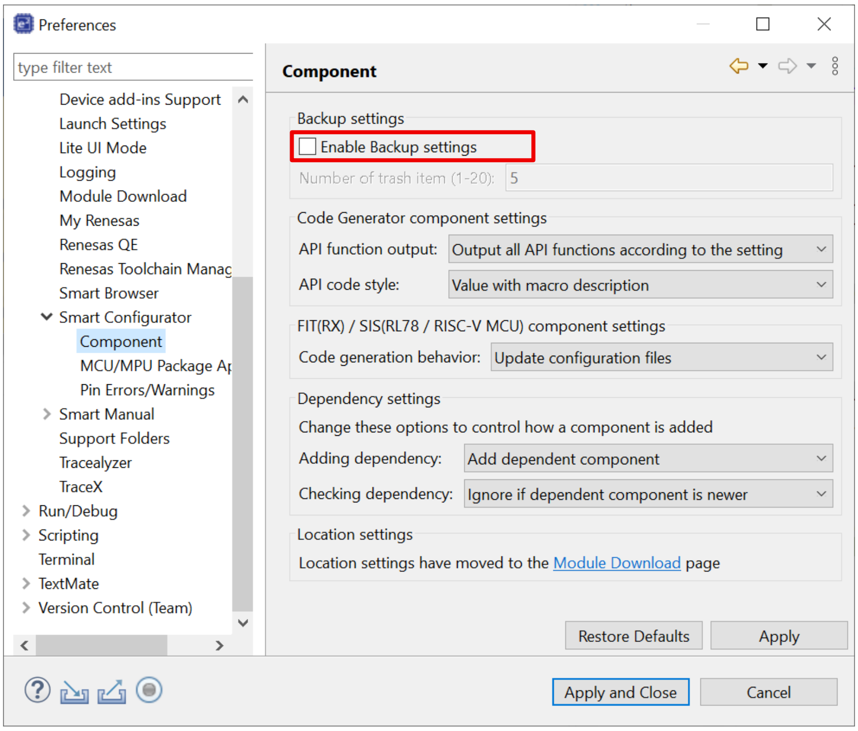
[After]

```
void R_Config_RTC_Set_AlarmOn(void)
{
    /* Enable RTC alarm operation */
    WALE = 1U;  /* alarm operation is valid */
    WAFG = 0U;  /* clear alarm detection flag */
}
```

Figure 3-10 Use 1-bit memory manipulation instruction

3.2.4 Improvement for updating the default status of the backup feature

From Smart Configurator for RL78 V1.16.0, the backup feature is disabled by default to improve build time in e^2 studio environment. If the user wants to use the backup feature, please enable it by setting “Enable Backup settings” in Preferences view:



Preferences

type filter text

Device add-ins Support
Launch Settings
Lite UI Mode
Logging
Module Download
My Renesas
Renesas QE
Renesas Toolchain Manag
Smart Browser
Smart Configurator
Component
MCU/MPU Package Ap
Pin Errors/Warnings
Smart Manual
Support Folders
Tracealyzer
TraceX
Run/Debug
Scripting
Terminal
TextMate
Version Control (Team)

Component

Backup settings
□ Enable Backup settings
Number of trash item (1-20): 5

Code Generator component settings
API function output: Output all API functions according to the setting
API code style: Value with macro description

FIT(RX) / SIS(RL78 / RISC-V MCU) component settings
Code generation behavior: Update configuration files

Dependency settings
Change these options to control how a component is added
Adding dependency: Add dependent component
Checking dependency: Ignore if dependent component is newer

Location settings
Location settings have moved to the Module Download page

Figure 3-11 Enable Backup settings

4. List of RENESAS TOOL NEWS AND TECHNICAL UPDATE

Below is a list of notifications delivered by RENESAS TOOL NEWS and TECHNICAL UPDATE.

Issue date	Document No.	Description	Applicable MCUs	Fixed version
Oct. 01, 2021	R20TS0757	1. Notes on creating LLVM for Renesas RL78 C/C++ Executable Project 2. Notes on using Port Input buffer function https://www.renesas.com/document/tnn/notes-e-studio-smart-configurator-plug-smart-configurator-rl78	RL78/G23	V1.2.0
Mar. 16, 2022	R20TS0822	1. Notes when build or clean e^2 studio Smart Configurator project https://www.renesas.com/document/tnn/notes-e-studio-smart-configurator-plug-smart-configurator-rl78-0	RL78/G23	V1.3.0
Dec. 01, 2022	R20TS0895	1. Notes when changing version of Board Support Program (BSP) or RL78 Software Integration System (SIS) modules https://www.renesas.com/us/en/document/tnn/notes-e-studio-smart-configurator-rl78-plug-smart-configurator-rl78	RL78/G23 RL78/F24 RL78/G15	V1.5.0

5. Points for Limitation

This section describes points for limitation regarding the Smart Configurator for RL78 V1.16.0.

5.1 List of Limitation

Table 5-1 List of Limitation (1/2) ✓: Applicable, -: Not Applicable

No	Description	RL78/G23	RL78/F24	RL78/G15	RL78/F23	RL78/G22	RL78/G16	RL78/G24	RL78/F25	Remarks
1	Note on extra help document issue	✓	✓	✓	✓	✓	✓	✓	✓	
2	Note on ELCL D flip flop component GUI warning display incorrectly	✓	–	–	–	–	–	–	–	
3	Note on the unsupported setting items for some ELCL components	✓	–	–	–	–	–	–	–	
4	Note on the user code protection feature will only be supported on the files that are generated by the Code Generation component	✓	✓	✓	✓	✓	✓	✓	✓	
5	Note on not using Flexible Application Accelerator (FAA) component in Mac OS and Linux	–	–	–	–	–	–	✓	–	
6	Note on not using SNOOZE Mode Sequencer (SMS) component in Mac OS and Linux	✓	–	–	–	✓	–	–	–	
7	Note on UI display with High Contrast theme on Linux OS	✓	✓	✓	✓	✓	✓	✓	✓	
8	Note on using a CCRL project in Visual Studio Code	–	–	–	–	–	–	–	–	
9	Note on IIC Communication (Slave mode) can't send or receive data correctly when using all address matching function in IICA	✓	–	–	–	✓	–	✓	–	
10	Note on the e^2 studio property reverts to the default value after generating the code	✓	✓	✓	✓	✓	✓	✓	✓	

Table 5-2 List of Limitation (2/2) ✓: Applicable, -: Not Applicable

No	Description	RL78/F22	RL78/L23	Remarks
1	Note on extra help document issue	✓	✓	
2	Note on ELCL D flip flop component GUI warning display incorrectly	–	–	
3	Note on the unsupported setting items for some ELCL components	–	–	
4	Note on the user code protection feature will only be supported on the files that are generated by the Code Generation component	✓	✓	
5	Note on not using Flexible Application Accelerator (FAA) component in Mac OS and Linux	–	–	
6	Note on not using SNOOZE Mode Sequencer (SMS) component in Mac OS and Linux	–	✓	
7	Note on UI display with High Contrast theme on Linux OS	✓	✓	
8	Note on using a CCRL project in Visual Studio Code	–	✓	
9	Note on IIC Communication (Slave mode) can't send or receive data correctly when using all address matching function in IICA	–	✓	
10	Note on the e^2 studio property reverts to the default value after generating the code	✓	✓	

5.2 Details of Limitation

5.2.1 Note on extra help document issue

For Smart Configurator, there is an extra help “Smart Browser” under “[Help] > [Help Contents]”.
Please ignore it.

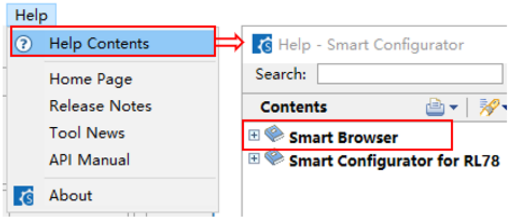


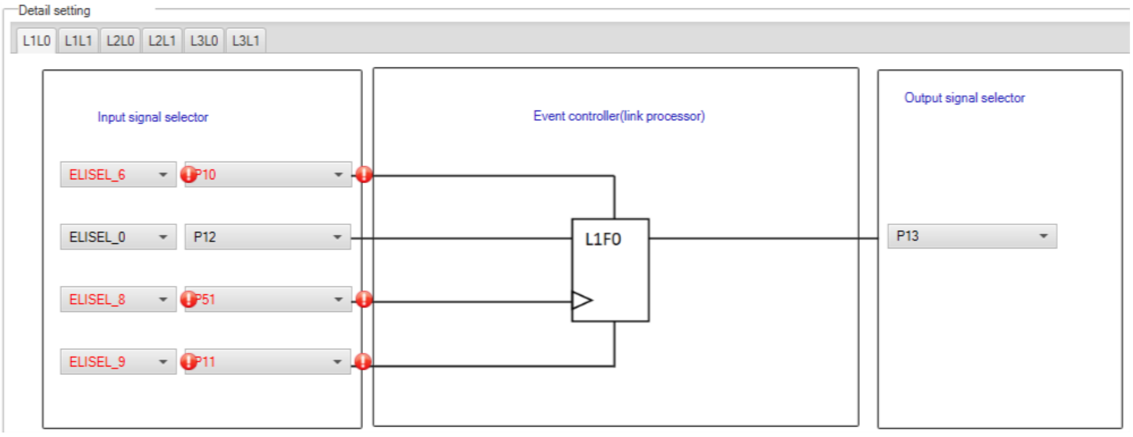
Figure 5-1 Extra help issue

5.2.2 Note on ELCL D flip flop component GUI warning display incorrectly

When selecting the event signal in ELCL D flip flop component, even if the selected signal consists with the hardware specification, there still displays the warning on the GUI.

[Workaround]
Make reference to the hardware manual and set the selectable event signal though warning appeared in GUI, the waring is no impact for the code generation.

The following is example of using flip-flop 0 and flip-flop 1 in ELCL logic cell block L1.



Detail setting

L1L0 | L1L1 | L2L0 | L2L1 | L3L0 | L3L1

Input signal selector

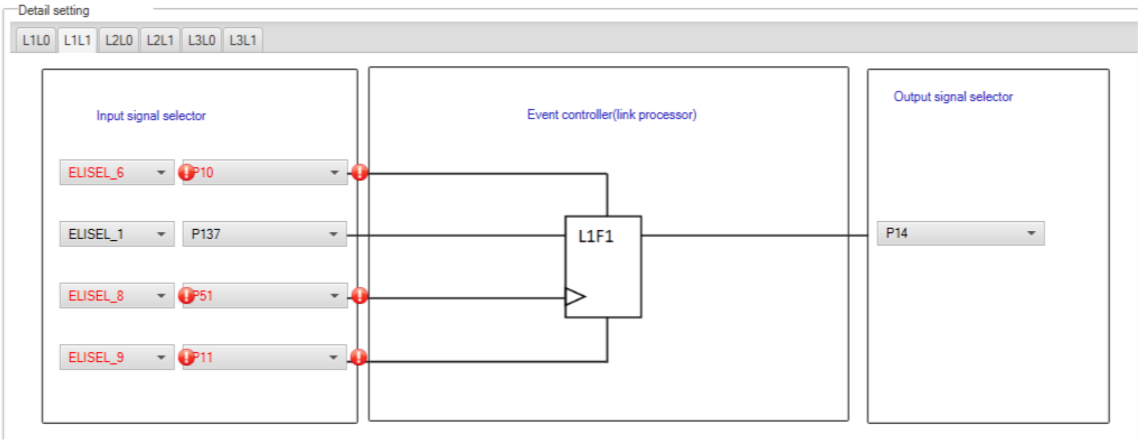
ELISEL_6 P10
ELISEL_0 P12
ELISEL_8 P51
ELISEL_9 P11

Event controller(link processor)

Output signal selector

P13

Figure 5-2 The flip-flop 0 in ELCL logic cell block L1 usage example



Detail setting

L1L0 | L1L1 | L2L0 | L2L1 | L3L0 | L3L1

Input signal selector

ELISEL_6 P10
ELISEL_1 P137
ELISEL_8 P51
ELISEL_9 P11

Event controller(link processor)

Output signal selector

P14

Figure 5-3 The flip-flop 1 in ELCL logic cell block L1 usage example

5.2.3 Note on the unsupported setting items for some ELCL components

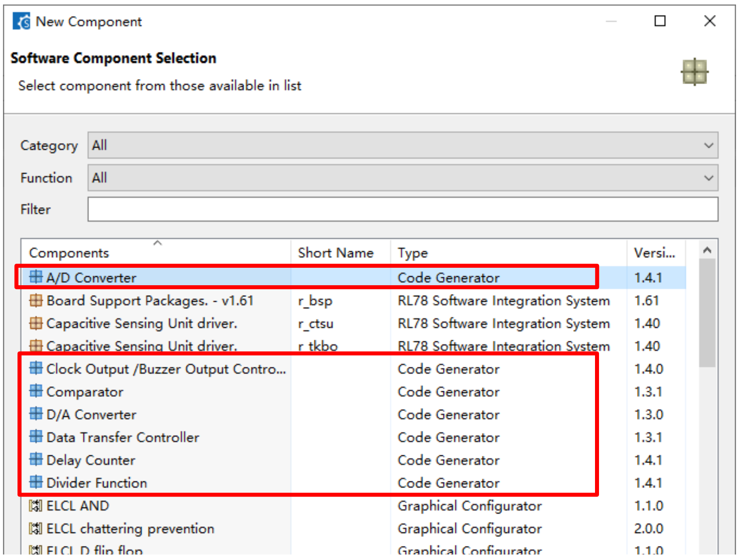
In the following ELCL modules, it is not possible to set “no selection (fixed to 0)” as the input signal of the logic cell block and “negative logic output (inverted)” as the output level of the event signal.

- ELCL AND
- ELCL D flip flop
- ELCL EXOR
- ELCL selector
- ELCL Through

[Workaround] None

5.2.4 Note on the user code protection feature will only be supported on the files that are generated by the Code Generation component

The user code protection feature will only be supported on the files that are generated by the Code Generation component. Hence, the user code protection feature is not available for non-Code Generation components.



New Component

Software Component Selection

Select component from those available in list

Category: All

Function: All

Filter:

Components	Short Name	Type	Versi...
A/D Converter		Code Generator	1.4.1
Board Support Packages. - v1.61	r_bsp	RL78 Software Integration System	1.61
Capacitive Sensing Unit driver.	r_ctsu	RL78 Software Integration System	1.40
Capacitive Sensing Unit driver.	r_tkbo	RL78 Software Integration System	1.40
Clock Output /Buzzer Output Contro...		Code Generator	1.4.0
Comparator		Code Generator	1.3.1
D/A Converter		Code Generator	1.3.0
Data Transfer Controller		Code Generator	1.3.1
Delay Counter		Code Generator	1.4.1
Divider Function		Code Generator	1.4.1
ELCL AND		Graphical Configurator	1.1.0
ELCL chattering prevention		Graphical Configurator	2.0.0
ELCL D flip flop		Graphical Configurator	1.1.0

Figure 5-4 Code Generation component in red frame

5.2.5 Note on not using Flexible Application Accelerator (FAA) component in Mac OS and Linux

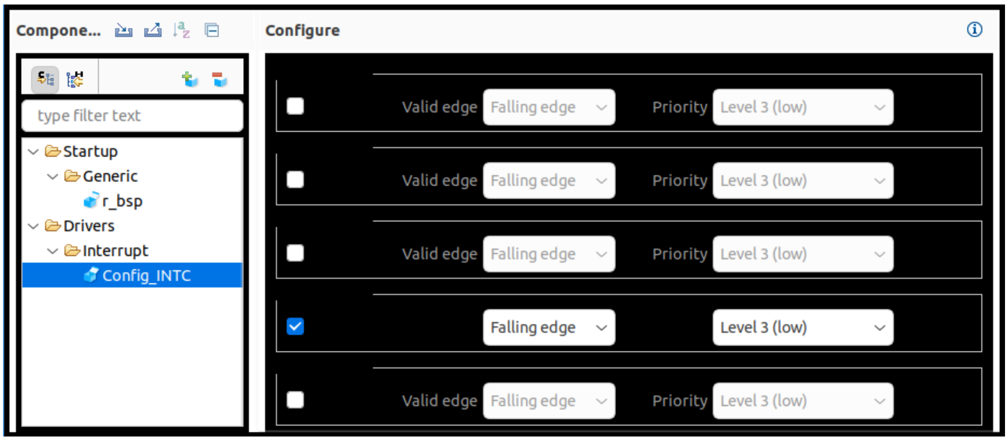
In Smart Configurator for RL78 V1.10.0 or later, Flexible Application Accelerator component was not supported in Mac OS and Linux. Though the user can add Flexible Application Accelerator component in Mac OS and Linux, but the generated Flexible Application Accelerator source code can’t be built successfully and works for running and debugging.

5.2.6 Note on not using SNOOZE Mode Sequencer (SMS) component in Mac OS and Linux

In Smart Configurator for RL78 V1.10.0 or later, SNOOZE Mode Sequencer component was not supported in Mac OS and Linux. Though the user can add SNOOZE Mode Sequencer component in Mac OS and Linux, but the generated SNOOZE Mode Sequencer source code can't be built successfully and works for running and debugging.

5.2.7 Note on UI display with High Contrast theme on Linux OS

When using e^2 studio with High Contrast theme on Linux OS, some display texts of Smart Configurator can't be seen. To avoid this issue, please use other themes.



☐ Valid edge Falling edge Priority Level 3 (low)
☐ Valid edge Falling edge Priority Level 3 (low)
☐ Valid edge Falling edge Priority Level 3 (low)
☑ Falling edge Level 3 (low)
☐ Valid edge Falling edge Priority Level 3 (low)

Figure 5-5 UI display with High Contrast theme

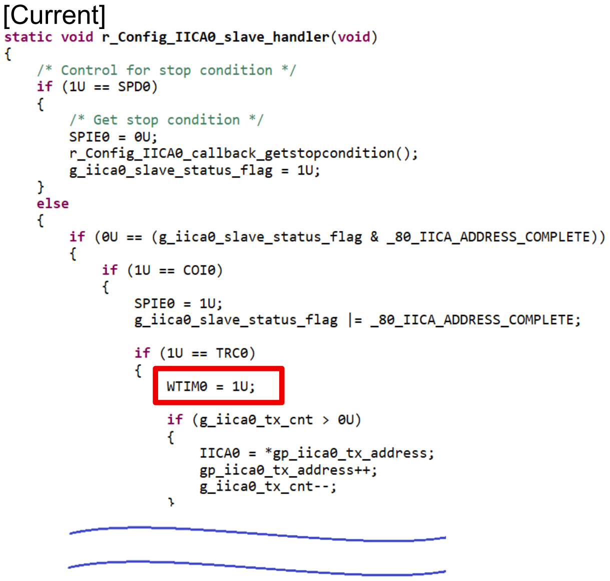
5.2.8 Note on using a CCRL project in Visual Studio Code

When creating a CCRL project in Visual Studio Code, it can't be built successfully and works for running and debugging.

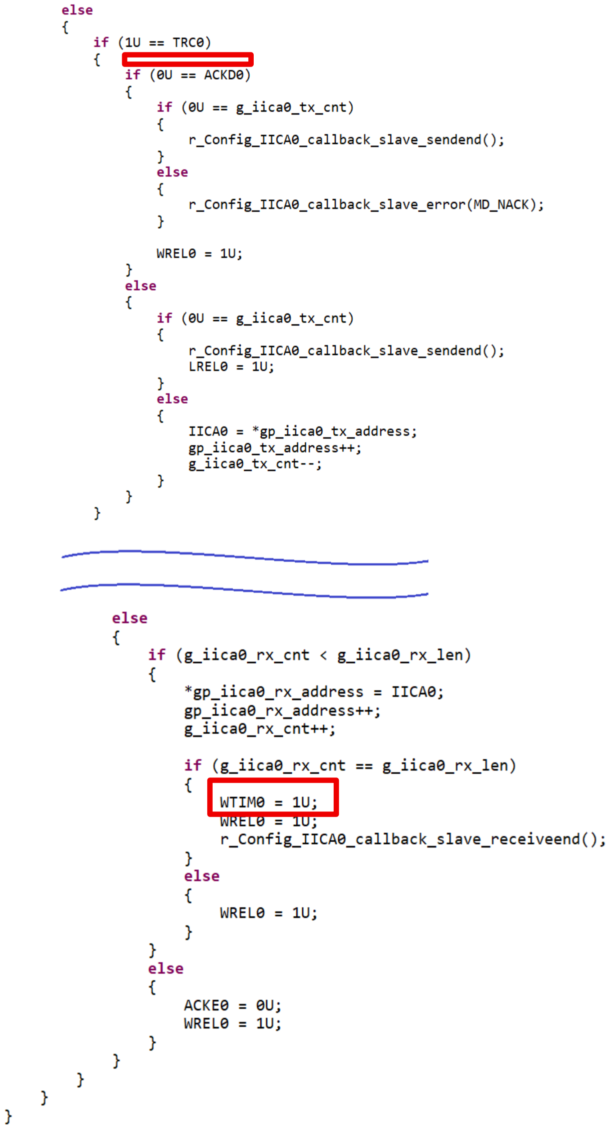
5.2.9 Note on IIC Communication (Slave mode) can't send or receive data correctly when using all address matching function in IICA

When the user selects IIC Communication (Slave mode) – IICA*n* and uses all address matching function, the driver code has some error in r_{Config_IICA*n*}_slave_handler(). It leads to send or receive data incorrectly. Smart Configurator will correct these errors in next release. (*n* means the channel number.)

[Workaround] Modify the driver code as following manually and modify it again after re-generating code.
[Current]



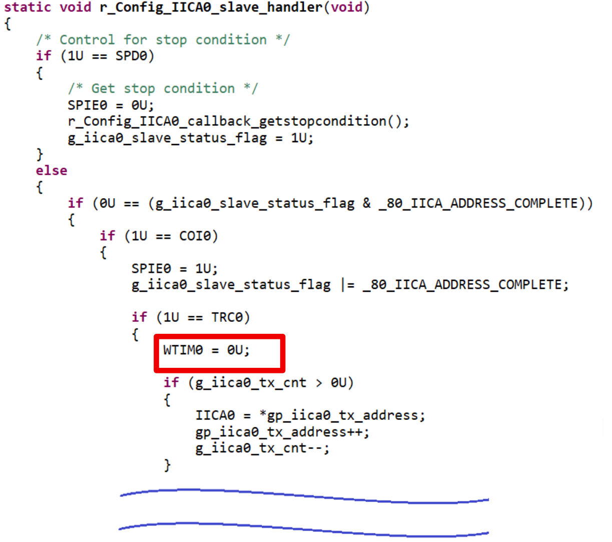
```
static void r_Config_IICA0_slave_handler(void)
{
    /* Control for stop condition */
    if (1U == SPD0)
    {
        /* Get stop condition */
        SPIE0 = 0U;
        r_Config_IICA0_callback_getstopcondition();
        g_iica0_slave_status_flag = 1U;
    }
    else
    {
        if (0U == (g_iica0_slave_status_flag & _80_IICA_ADDRESS_COMPLETE))
        {
            if (1U == COI0)
            {
                SPIE0 = 1U;
                g_iica0_slave_status_flag |= _80_IICA_ADDRESS_COMPLETE;

                if (1U == TRC0)
                {
                    WTIM0 = 1U;

                    if (g_iica0_tx_cnt > 0U)
                    {
                        IICA0 = *gp_iica0_tx_address;
                        gp_iica0_tx_address++;
                        g_iica0_tx_cnt--;
                    }
```



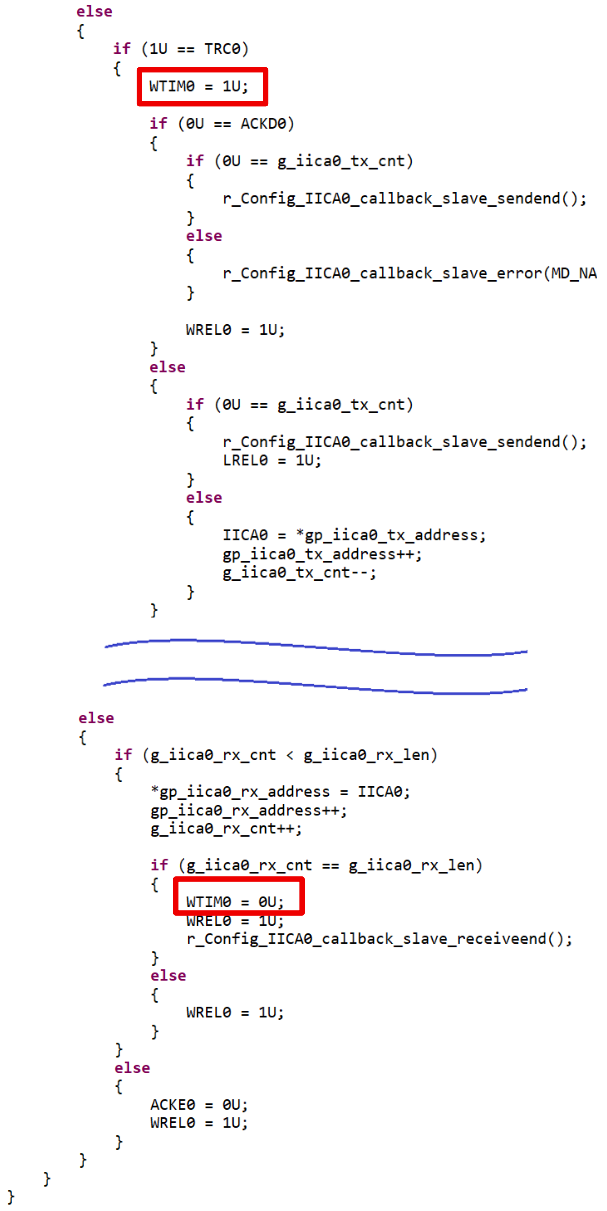
```
        else
        {
            if (1U == TRC0)
            {
                if (0U == ACKD0)
                {
                    if (0U == g_iica0_tx_cnt)
                    {
                        r_Config_IICA0_callback_slave_sendend();
                    }
                    else
                    {
                        r_Config_IICA0_callback_slave_error(MD_NACK);
                    }

                    WREL0 = 1U;
                }
                else
                {
                    if (0U == g_iica0_tx_cnt)
                    {
                        r_Config_IICA0_callback_slave_sendend();
                        LREL0 = 1U;
                    }
                    else
                    {
                        IICA0 = *gp_iica0_tx_address;
                        gp_iica0_tx_address++;
                        g_iica0_tx_cnt--;
                    }
                }
            }
```

```
            else
            {
                if (g_iica0_rx_cnt < g_iica0_rx_len)
                {
                    *gp_iica0_rx_address = IICA0;
                    gp_iica0_rx_address++;
                    g_iica0_rx_cnt++;

                    if (g_iica0_rx_cnt == g_iica0_rx_len)
                    {
                        WTIM0 = 1U;
                        WREL0 = 1U;
                        r_Config_IICA0_callback_slave_receiveend();
                    }
                    else
                    {
                        WREL0 = 1U;
                    }
                }
                else
                {
                    ACKE0 = 0U;
                    WREL0 = 1U;
                }
            }
        }
    }
}
```

[Modified]



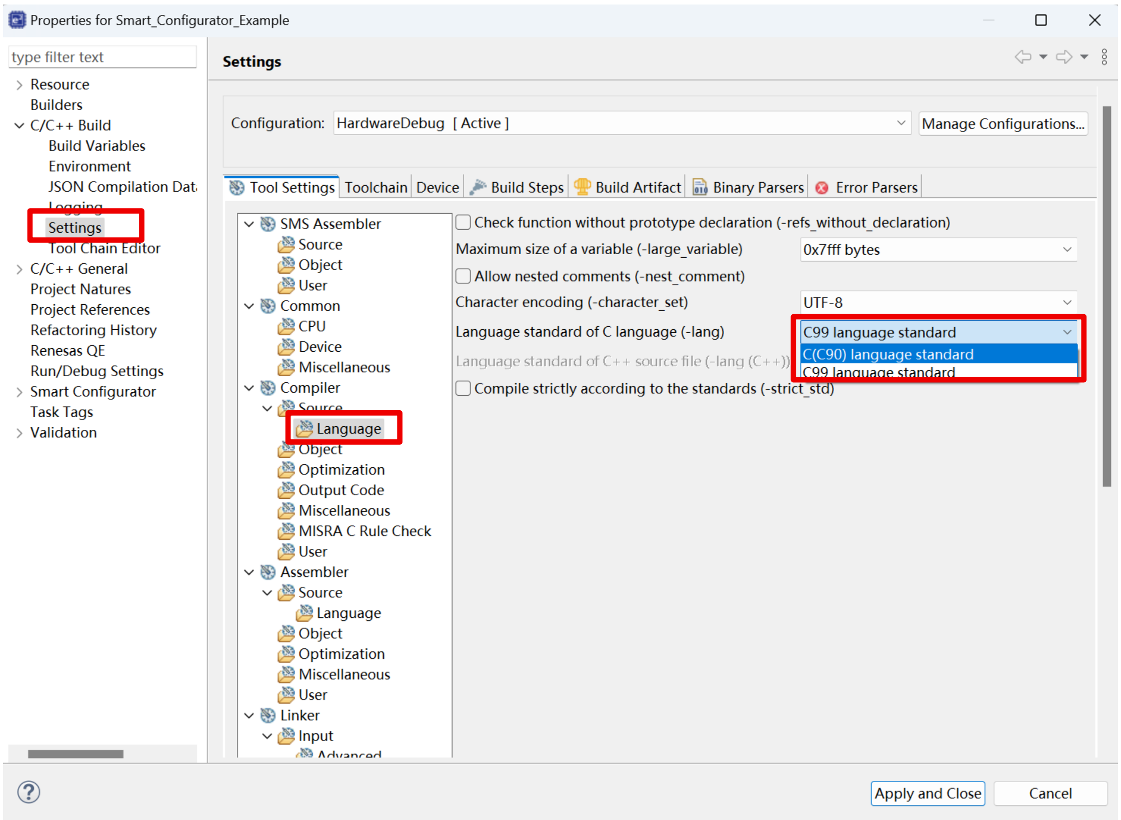
```
static void r_Config_IICA0_slave_handler(void)
{
    /* Control for stop condition */
    if (1U == SPD0)
    {
        /* Get stop condition */
        SPIE0 = 0U;
        r_Config_IICA0_callback_getstopcondition();
        g_iica0_slave_status_flag = 1U;
    }
    else
    {
        if (0U == (g_iica0_slave_status_flag & _80_IICA_ADDRESS_COMPLETE))
        {
            if (1U == COI0)
            {
                SPIE0 = 1U;
                g_iica0_slave_status_flag |= _80_IICA_ADDRESS_COMPLETE;

                if (1U == TRC0)
                {
                    WTIM0 = 0U;

                    if (g_iica0_tx_cnt > 0U)
                    {
                        IICA0 = *gp_iica0_tx_address;
                        gp_iica0_tx_address++;
                        g_iica0_tx_cnt--;
                    }
```



```
        else
        {
            if (1U == TRC0)
            {
                WTIM0 = 1U;
                if (0U == ACKD0)
                {
                    if (0U == g_iica0_tx_cnt)
                    {
                        r_Config_IICA0_callback_slave_sendend();
                    }
                    else
                    {
                        r_Config_IICA0_callback_slave_error(MD_NA
                    }

                    WREL0 = 1U;
                }
                else
                {
                    if (0U == g_iica0_tx_cnt)
                    {
                        r_Config_IICA0_callback_slave_sendend();
                        LREL0 = 1U;
                    }
                    else
                    {
                        IICA0 = *gp_iica0_tx_address;
                        gp_iica0_tx_address++;
                        g_iica0_tx_cnt--;
                    }
                }

        else
        {
            if (g_iica0_rx_cnt < g_iica0_rx_len)
            {
                *gp_iica0_rx_address = IICA0;
                gp_iica0_rx_address++;
                g_iica0_rx_cnt++;

                if (g_iica0_rx_cnt == g_iica0_rx_len)
                {
                    WTIM0 = 0U;
                    WREL0 = 1U;
                    r_Config_IICA0_callback_slave_receiveend();
                }
                else
                {
                    WREL0 = 1U;
                }
            }
            else
            {
                ACKE0 = 0U;
                WREL0 = 1U;
            }
        }
    }
}
```

5.2.10 Note on the e^2 studio property reverts to the default value after generating the code

When code generation is performed, the e^2 studio property about "Language standard of C language (-lang)" reverts to the default values (C99 language standard). This issue will fix in next release.



Properties for Smart_Configurator_Example

Settings

Configuration: HardwareDebug [Active]

□ Check function without prototype declaration (-refs_without_declaration)

Maximum size of a variable (-large_variable): 0x7fff bytes

□ Allow nested comments (-nest_comment)

Character encoding (-character_set): UTF-8

Language standard of C language (-lang): C99 language standard

C(C90) language standard
C99 language standard

Language standard of C++ source file (-lang (C++))

□ Compile strictly according to the standards (-strict_std)

Figure 5-6 The property of "Language standard of C language (-lang)"

[Workaround] Set "Language standard of C language (-lang)" manually and modify it again after re-generating code.

6. Points for Caution

This section describes points for caution regarding the Smart Configurator for RL78 V1.16.0.

6.1 List of Caution

Table 6-1 List of Caution (1/2) ✓: Applicable, -: Not Applicable

No	Description	RL78/G23	RL78/F24	RL78/G15	RL78/F23	RL78/G22	RL78/G16	RL78/G24	RL78/F25	Remarks
1	Note on the build error message such as "section .bss virtual address range overlaps with .dtc_vectortable"	✓	✓	–	✓	✓	–	✓	✓	
2	Note on the installation of the Smart Configurator	✓	✓	✓	✓	✓	✓	✓	✓	
3	Note on using TRDIOA0 for Input capture and TRDIOB0 for Output compare at same time	–	✓	–	✓	–	–	✓	✓	
4	Note on pulse width calculation of Timer RD input capture function	–	✓	–	✓	–	–	✓	✓	
5	Note on the include path update issue when renaming the component's configuration name	✓	✓	✓	✓	✓	✓	✓	✓	
6	Note on TAU Input Signal High/Low level Measurement components.	✓	✓	✓	✓	✓	✓	✓	✓	
7	Note on CC-RL V1.12 C++ project	✓	✓	✓	✓	✓	✓	✓	✓	
8	Note on browsing "Release Notes" and "Tool News" URL from the help menu	✓	✓	✓	–	–	–	–	–	
9	Note on using the user code protection feature	✓	✓	✓	✓	✓	✓	✓	✓	
10	Note on IAR build error when using SNOOZE Mode Sequencer (SMS) function	✓	–	–	–	–	–	–	–	
11	Note on A/D conversion time setting after performing [Change device] or [Change resource]	✓	✓	✓	✓	✓	✓	✓	✓	
12	Note on changing Hardware Debug Configuration on project generation wizard	✓	✓	✓	✓	✓	✓	✓	✓	
13	Note on Pin Number maybe wrong in [Pins] page when loading project	–	–	–	–	–	–	✓	–	
14	Note on Pin assignment with PIOR maybe wrong in [Pins] page when changing device	✓	✓	✓	✓	✓	✓	✓	✓	
15	Note on the case that a pin of an unsupported component is assigned when the device is changed	✓	✓	✓	✓	✓	✓	✓	✓	
16	Note on the Pin assignement and pin setting maybe wrong when loading project	–	–	–	–	–	–	–	–	
17	Note on the Developer Assistance can't jump correctly to another HTML file	✓	✓	✓	✓	✓	✓	✓	✓	
18	Note on the clock UI display issue when changing screen scale	✓	✓	✓	✓	✓	✓	✓	✓	

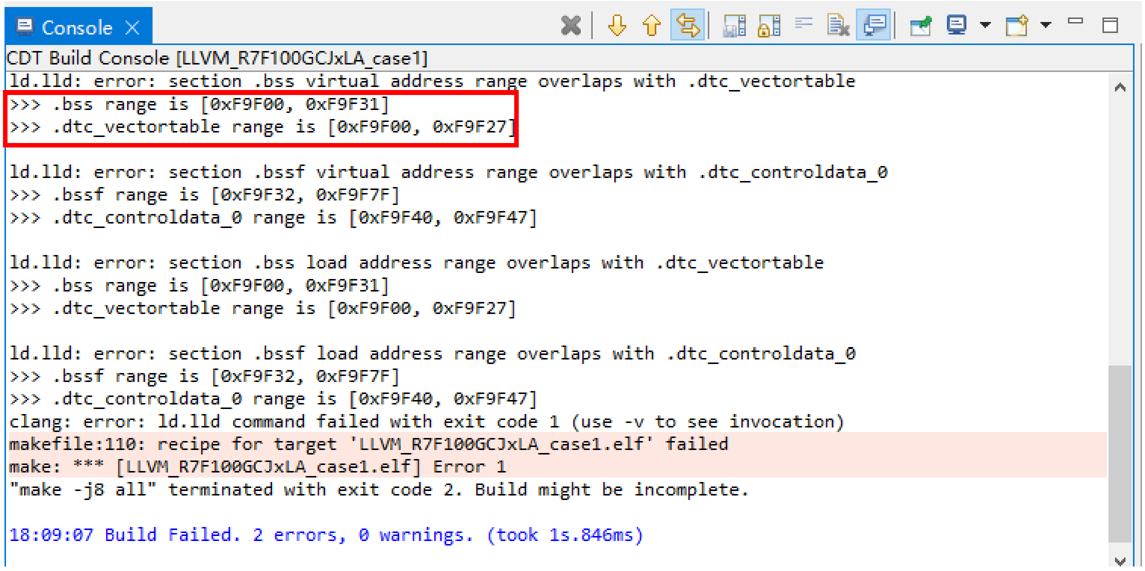
Table 6-2 List of Caution (2/2) ✓: Applicable, -: Not Applicable

No	Description	RL78/F22	RL78/L23	Remarks
1	Note on the build error message such as "section .bss virtual address range overlaps with .dtc_vectortable"	✓	✓	
2	Note on the installation of the Smart Configurator	✓	✓	
3	Note on using TRDIOA0 for Input capture and TRDIOB0 for Output compare at same time	✓	–	
4	Note on pulse width calculation of Timer RD input capture function	✓	–	
5	Note on the include path update issue when renaming the component’s configuration name	✓	✓	
6	Note on TAU Input Signal High/Low level Measurement components.	✓	✓	
7	Note on CC-RL V1.12 C++ project	✓	✓	
8	Note on browsing "Release Notes" and "Tool News" URL from the help menu	–	–	
9	Note on using the user code protection feature	✓	✓	
10	Note on IAR build error when using SNOOZE Mode Sequencer (SMS) function	–	–	
11	Note on A/D conversion time setting after performing [Change device] or [Change resource]	✓	✓	
12	Note on changing Hardware Debug Configuration on project generation wizard	✓	✓	
13	Note on Pin Number maybe wrong in [Pins] page when loading project	–	–	
14	Note on Pin assignment with PIOR maybe wrong in [Pins] page when changing device	✓	✓	
15	Note on the case that a pin of an unsupported component is assigned when the device is changed	✓	✓	
16	Note on the Pin assignement and pin setting maybe wrong when loading project	–	✓	
17	Note on the Developer Assistance can’t jump correctly to another HTML file	✓	✓	
18	Note on the clock UI display issue when changing screen scale	✓	✓	

6.2 Details of Caution

6.2.1 Note on the build error message such as "section .bss virtual address range overlaps with .dtc_vectortable"

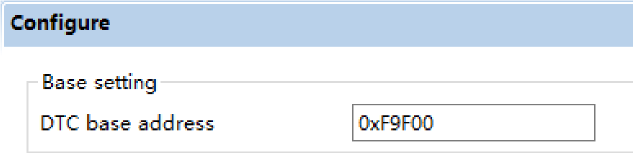
When the user uses many components and DTC component together, the generated code build might fail due to some section address overlaps.



```
CDT Build Console [LLVM_R7F100GCJxLA_case1]
ld.lld: error: section .bss virtual address range overlaps with .dtc_vectortable
>>> .bss range is [0xF9F00, 0xF9F31]
>>> .dtc_vectortable range is [0xF9F00, 0xF9F27]

ld.lld: error: section .bssf virtual address range overlaps with .dtc_controldata_0
>>> .bssf range is [0xF9F32, 0xF9F7F]
>>> .dtc_controldata_0 range is [0xF9F40, 0xF9F47]

ld.lld: error: section .bss load address range overlaps with .dtc_vectortable
>>> .bss range is [0xF9F00, 0xF9F31]
>>> .dtc_vectortable range is [0xF9F00, 0xF9F27]

ld.lld: error: section .bssf load address range overlaps with .dtc_controldata_0
>>> .bssf range is [0xF9F32, 0xF9F7F]
>>> .dtc_controldata_0 range is [0xF9F40, 0xF9F47]
clang: error: ld.lld command failed with exit code 1 (use -v to see invocation)
makefile:110: recipe for target 'LLVM_R7F100GCJxLA_case1.elf' failed
make: *** [LLVM_R7F100GCJxLA_case1.elf] Error 1
"make -j8 all" terminated with exit code 2. Build might be incomplete.

18:09:07 Build Failed. 2 errors, 0 warnings. (took 1s.846ms)
```

Figure 6-1 Build error message

[Workaround]
The Smart Configurator cannot set ".bss" and ".bssf" section address. So user should consider to modify ".bss" and ".bssf" section address manually in "linker_script.ld" file or change the DTC base address to avoid such section overlap error.



Configure

Base setting

DTC base address 0xF9F00

Figure 6-2 DTC base address setting

6.2.2 Note on the installation of the Smart Configurator

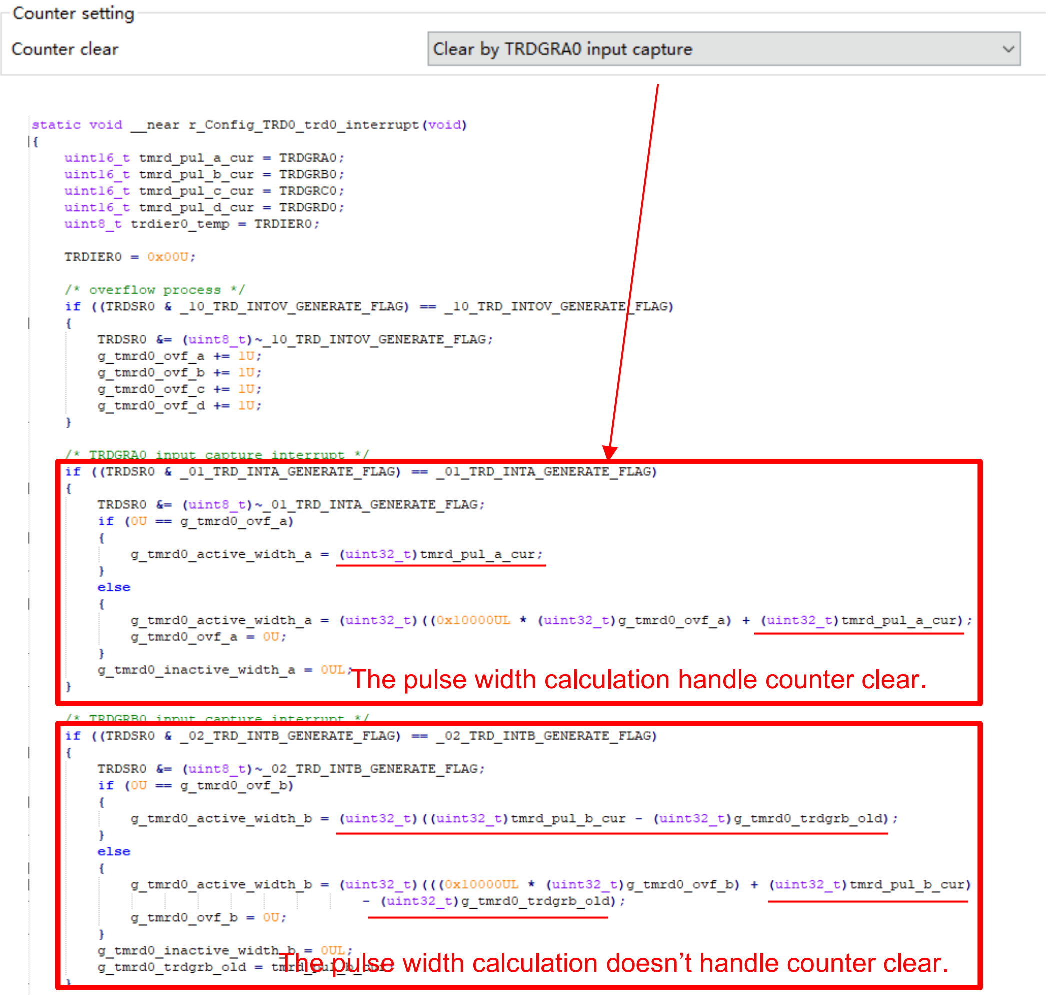
Do not set more than 64 characters for the installation directory.
The user might see an error message "The specified path is too long" and will not be able to install Smart Configurator.

6.2.3 Note on using TRDIOA0 for Input capture and TRDIOB0 for Output compare at same time

If the user sets up TRDIOA0 for Input capture and TRDIOB0 for Output compare at the same time. Smart Configurator will output a Peripheral conflict error.
The user can ignore this Smart Configurator error message and use these two functions at the same time.

6.2.4 Note on pulse width calculation of Timer RD input capture function

The pulse width calculation code is with the assumption that the counter is not cleared between two interrupts occurrence, except the input pulse width which is selected as counter clear trigger on GUI.
For example, when "Clear by TRDGRA*n* input capture" is selected, only TRDIOA*n* pulse width calculation handle counter clear, other input pulse width calculation doesn’t handle counter clear.



Counter setting
Counter clear: Clear by TRDGRA0 input capture

```
static void __near r_Config_TRD0_trd0_interrupt(void)
{
    uint16_t tmrd_pul_a_cur = TRDGRA0;
    uint16_t tmrd_pul_b_cur = TRDGRB0;
    uint16_t tmrd_pul_c_cur = TRDGRC0;
    uint16_t tmrd_pul_d_cur = TRDGRD0;
    uint8_t trdier0_temp = TRDIER0;

    TRDIER0 = 0x00U;

    /* overflow process */
    if ((TRDSR0 & _10_TRD_INTOV_GENERATE_FLAG) == _10_TRD_INTOV_GENERATE_FLAG)
    {
        TRDSR0 &= (uint8_t)~_10_TRD_INTOV_GENERATE_FLAG;
        g_tmrd0_ovf_a += 1U;
        g_tmrd0_ovf_b += 1U;
        g_tmrd0_ovf_c += 1U;
        g_tmrd0_ovf_d += 1U;
    }

    /* TRDGRA0 input capture interrupt */
    if ((TRDSR0 & _01_TRD_INTA_GENERATE_FLAG) == _01_TRD_INTA_GENERATE_FLAG)
    {
        TRDSR0 &= (uint8_t)~_01_TRD_INTA_GENERATE_FLAG;
        if (0U == g_tmrd0_ovf_a)
        {
            g_tmrd0_active_width_a = (uint32_t)tmrd_pul_a_cur;
        }
        else
        {
            g_tmrd0_active_width_a = (uint32_t)((0x10000UL * (uint32_t)g_tmrd0_ovf_a) + (uint32_t)tmrd_pul_a_cur);
            g_tmrd0_ovf_a = 0U;
        }
        g_tmrd0_inactive_width_a = 0UL;
    }

    /* TRDGRB0 input capture interrupt */
    if ((TRDSR0 & _02_TRD_INTB_GENERATE_FLAG) == _02_TRD_INTB_GENERATE_FLAG)
    {
        TRDSR0 &= (uint8_t)~_02_TRD_INTB_GENERATE_FLAG;
        if (0U == g_tmrd0_ovf_b)
        {
            g_tmrd0_active_width_b = (uint32_t)((uint32_t)tmrd_pul_b_cur - (uint32_t)g_tmrd0_trdgrb_old);
        }
        else
        {
            g_tmrd0_active_width_b = (uint32_t)(((0x10000UL * (uint32_t)g_tmrd0_ovf_b) + (uint32_t)tmrd_pul_b_cur)
                                    - (uint32_t)g_tmrd0_trdgrb_old);
            g_tmrd0_ovf_b = 0U;
        }
        g_tmrd0_inactive_width_b = 0UL;
        g_tmrd0_trdgrb_old = tmrd_pul_b_cur
    }
```

The pulse width calculation handle counter clear.

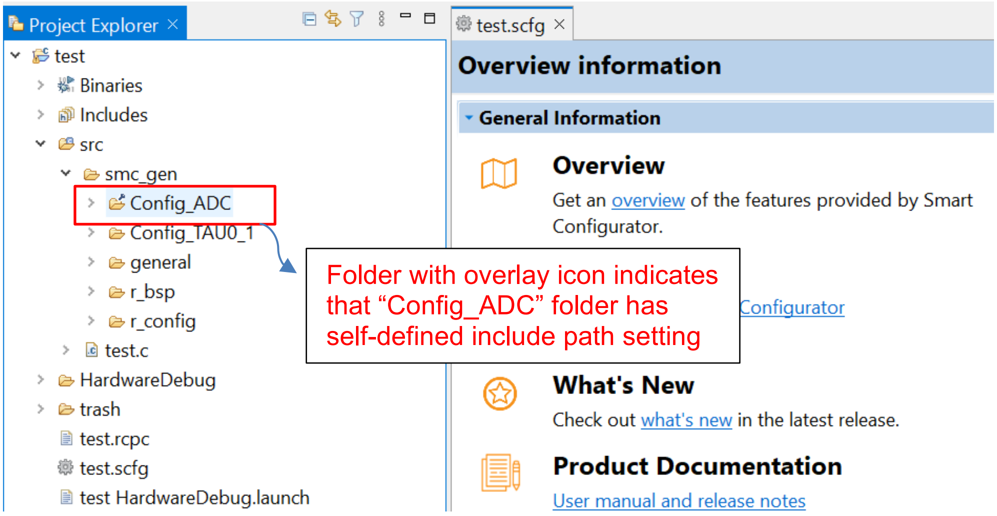
The pulse width calculation doesn’t handle counter clear.

Figure 6-3 Counter clear setting in Input capture function

6.2.5 Note on the include path update issue when renaming the component's configuration name

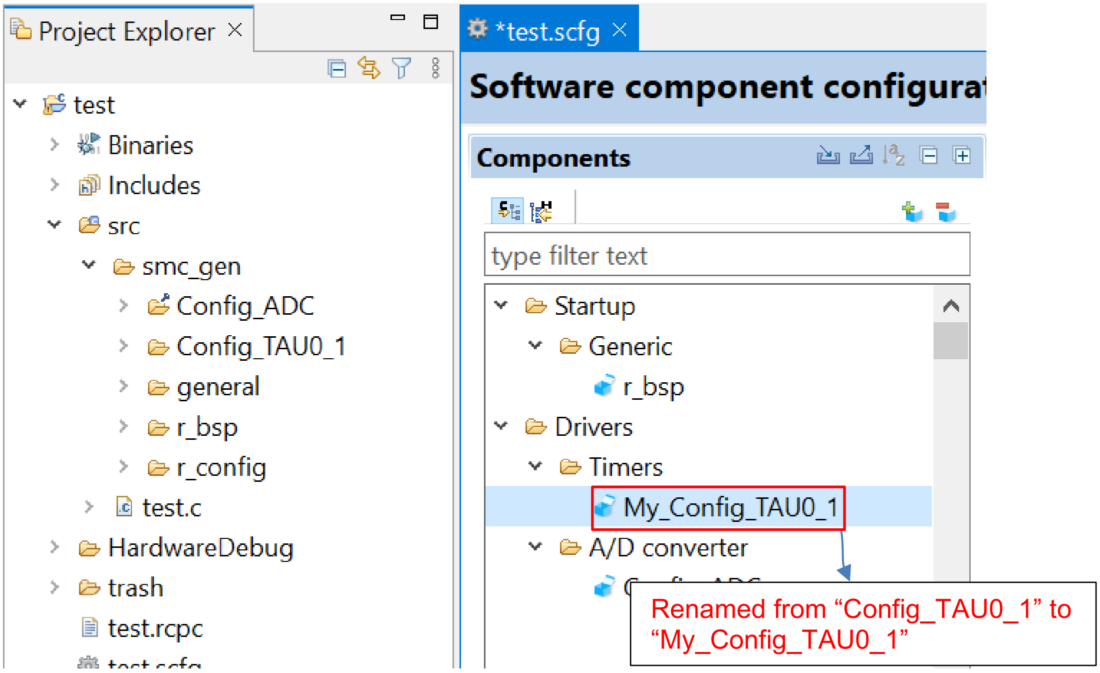
When renaming the added component's configuration in e^2 studio Smart Configurator project that has self-defined include path setting for any folder or file, include path setting for that folder or file will keep the old name setting after code generation. This will cause build error when compiling the newly generated codes so please manually update the include path.

The folder or file which has self-defined include path setting can be recognized by checking the overlay icon ( ) on that folder or file. Below is an example on how to handle the include path update after renaming Compare Match Timer component configuration.



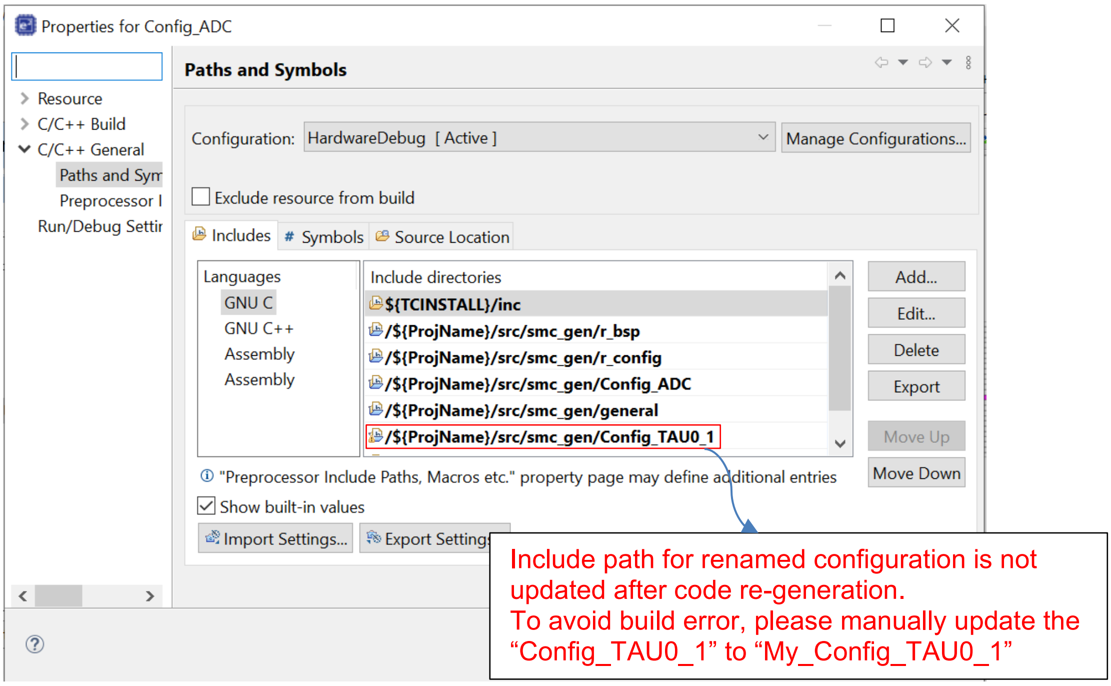
Folder with overlay icon indicates that "Config_ADC" folder has self-defined include path setting

Figure 6-4 Interval Timer component configuration before renaming



Renamed from "Config_TAU0_1" to "My_Config_TAU0_1"

Figure 6-5 The Interval Timer component configuration after renaming



Configuration: HardwareDebug [Active] Manage Configurations...

□ Exclude resource from build

Include directories
- ${TCINSTALL}/inc
- /${ProjName}/src/smc_gen/r_bsp
- /${ProjName}/src/smc_gen/r_config
- /${ProjName}/src/smc_gen/Config_ADC
- /${ProjName}/src/smc_gen/general
- /${ProjName}/src/smc_gen/Config_TAU0_1

"Preprocessor Include Paths, Macros etc." property page may define additional entries

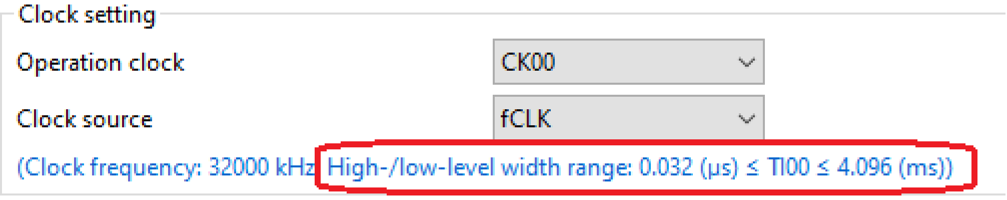
☑ Show built-in values

Include path for renamed configuration is not updated after code re-generation.
To avoid build error, please manually update the “Config_TAU0_1” to “My_Config_TAU0_1”

Figure 6-6 Include path setting for the “Config_ADC” configuration

6.2.6 Note on TAU Input Signal High/Low level Measurement component

When using TAU Input Signal High/Low level Measurement component, after used noise filter function for TImn input pulse, please make sure the High/Low level width min value needs to be greater than two times the minimum value prompted on the UI.
For example, the High/Low level width min value is 0.032us (min value), when use noise filter function, the width min value should be 0.064us.



Clock setting

Operation clock: CK00

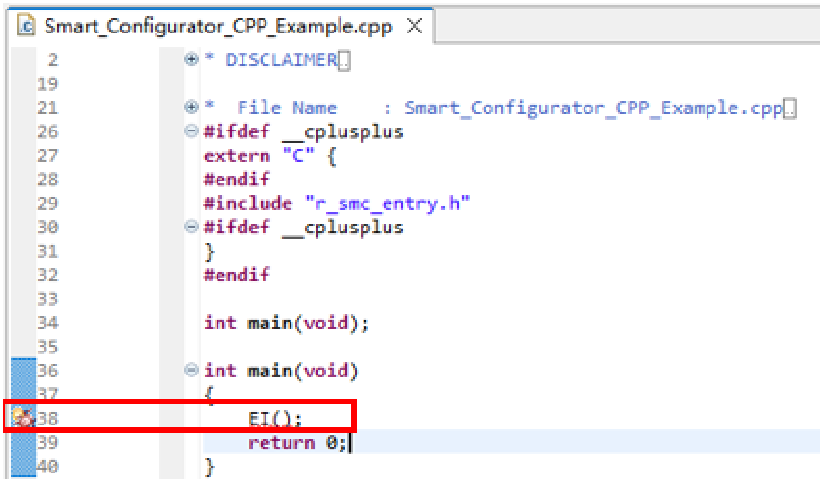
Clock source: fCLK

(Clock frequency: 32000 kHz High-/low-level width range: 0.032 (μs) ≤ TI00 ≤ 4.096 (ms))

Figure 6-7 High/Low level width min value

6.2.7 Note on CC-RL V1.12 C++ project

In CC-RL V1.12 or later C++ project, there are some dummy issues such as “EI()” in editor. However this is editor specification and does not affect the program operation. Please ignore it.



```
* DISCLAIMER
*  File Name    : Smart_Configurator_CPP_Example.cpp
#ifdef __cplusplus
extern "C" {
#endif
#include "r_smc_entry.h"
#ifdef __cplusplus
}
#endif

int main(void);

int main(void)
{
    EI();
    return 0;
}
```

Figure 6-8 CODAN issue in CC-RL V1.12 C++ project

6.2.8 Note on browsing "Release Notes" and "Tool News" URL from the help menu

For Smart Configurator for RL78 V1.15.0 or before version, "Release Notes" and "Tools News" in the help menu cannot access the correct URL. Please access the URL below directly for Smart Configurator for RL78 V1.15.0 or before version.
Release Notes:
https://www.renesas.com/en/software-tool/rl78-smart-configurator?documents-type-filter=release-note#documents
Tool News:
https://www.renesas.com/en/software-tool/rl78-smart-configurator?documents-type-filter=tool-news-note#documents

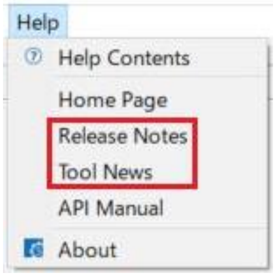


Figure 6-9 Release Notes and Tool News in Smart Configurators

6.2.9 Note on using the user code protection feature

From Smart Configurator for RL78 V1.5.0 onwards, the user code protection feature will be supported for all Code Generation components. Please use the following specific tags to add user code when using the user code protection feature. If the specific tags do not match exactly, inserted user code will not be protected after the code generation.

/* Start user code */

User code can be added between the specific tags

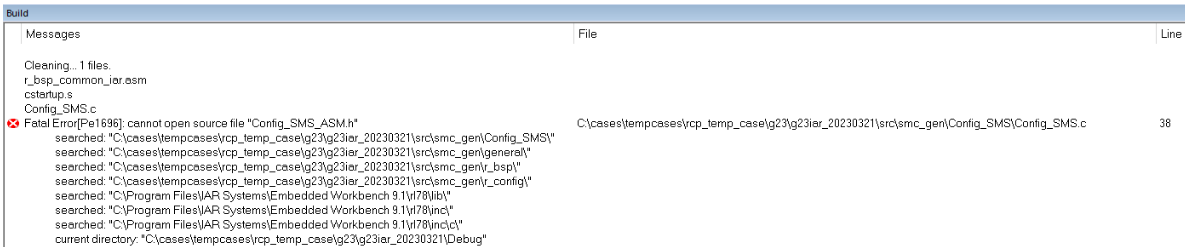
/* End user code */

6.2.10 Note on IAR build error when using SNOOZE Mode Sequencer (SMS) component

When using SNOOZE Mode Sequencer (SMS) component, if the following build error is met in IAR Embedded workbench, please check the build order setting in project [Options…] -> [Custom Build] page.

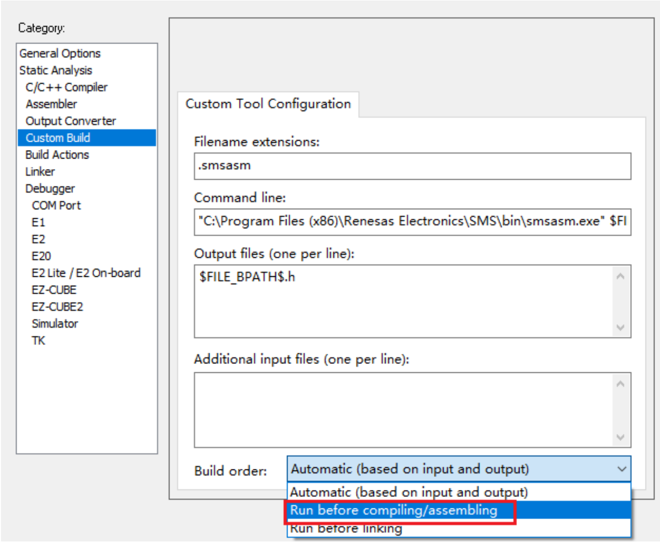
1) When using IAR Embedded workbench V5.10, select "Run before compiling/assembling" (refer to Figure 6-11)
2) When using IAR Embedded workbench V4.21, make “Run this tool before all other tools” checked (refer to Figure 6-12)

The above setting can eliminate this build error.



```
Build
Messages                                                                 File                                                                              Line

Cleaning... 1 files.
r_bsp_common_iar.asm
cstartup.s
Config_SMS.c
Fatal Error[Pe1696]: cannot open source file "Config_SMS_ASM.h"          C:\cases\tempcases\rcp_temp_case\g23\g23iar_20230321\src\smc_gen\Config_SMS\Config_SMS.c   38
        searched: "C:\cases\tempcases\rcp_temp_case\g23\g23iar_20230321\src\smc_gen\Config_SMS\"
        searched: "C:\cases\tempcases\rcp_temp_case\g23\g23iar_20230321\src\smc_gen\general\"
        searched: "C:\cases\tempcases\rcp_temp_case\g23\g23iar_20230321\src\smc_gen\r_bsp\"
        searched: "C:\cases\tempcases\rcp_temp_case\g23\g23iar_20230321\src\smc_gen\r_config\"
        searched: "C:\Program Files\IAR Systems\Embedded Workbench 9.1\rl78\lib\"
        searched: "C:\Program Files\IAR Systems\Embedded Workbench 9.1\rl78\inc\"
        searched: "C:\Program Files\IAR Systems\Embedded Workbench 9.1\rl78\inc\c\"
        current directory: "C:\cases\tempcases\rcp_temp_case\g23\g23iar_20230321\Debug"
```

Figure 6-10 IAR build error



Filename extensions: .smsasm

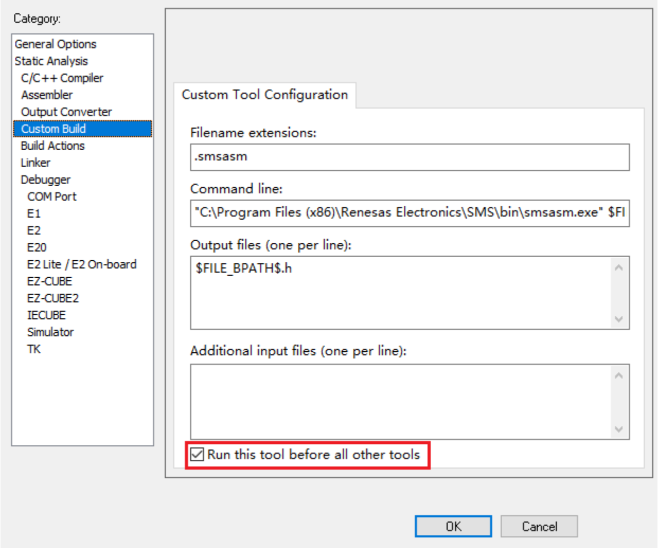
Command line: "C:\Program Files (x86)\Renesas Electronics\SMS\bin\smsasm.exe" $FI

Output files (one per line): $FILE_BPATH$.h

Additional input files (one per line):

Build order: Automatic (based on input and output)

Figure 6-11 "Build order" setting of IAR Embedded workbench V5.10



Filename extensions: .smsasm

Command line: "C:\Program Files (x86)\Renesas Electronics\SMS\bin\smsasm.exe" $FI

Output files (one per line): $FILE_BPATH$.h

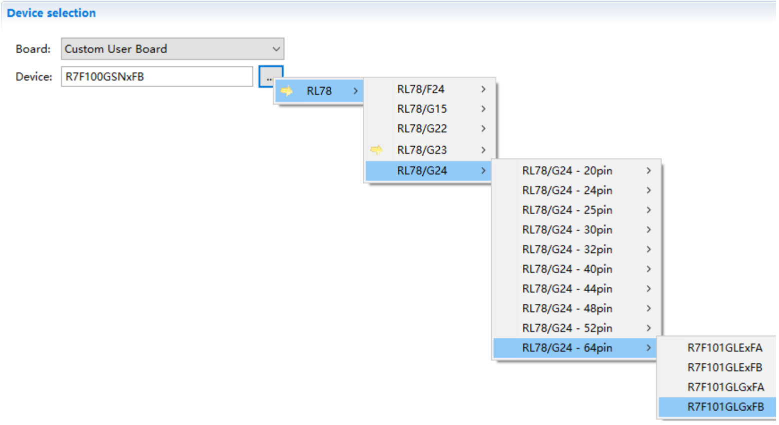
Additional input files (one per line):

☑ Run this tool before all other tools

Figure 6-12 Custom build setting of IAR Embedded workbench V4.21

6.2.11 Note on A/D conversion time setting after performing [Change device] or [Change resource]

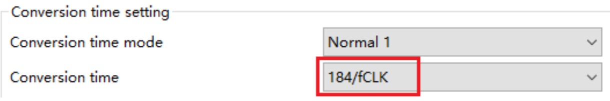
After performing [Change device] (for example, change from RL78/G23 to RL78/G24), the A/D conversion time setting can’t be kept. The user should take note to reconfirm the conversion time setting as he wants.



Board: Custom User Board

Device: R7F100GSNxFB

Figure 6-13 [Change device] operation



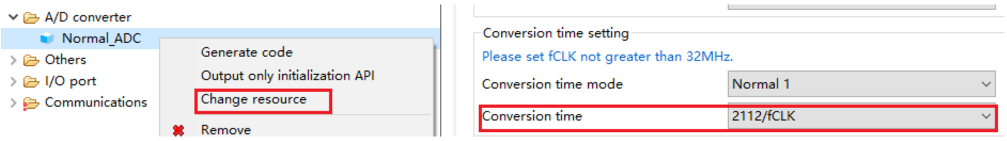
Conversion time setting

Conversion time mode: Normal 1

Conversion time: 184/fCLK

Figure 6-14 A/D conversion time setting

When changing resource, for example from RL78/G24 normal A/D and RL78/G24 advanced A/D, the A/D conversion time can’t be kept.



Conversion time setting

Please set fCLK not greater than 32MHz.

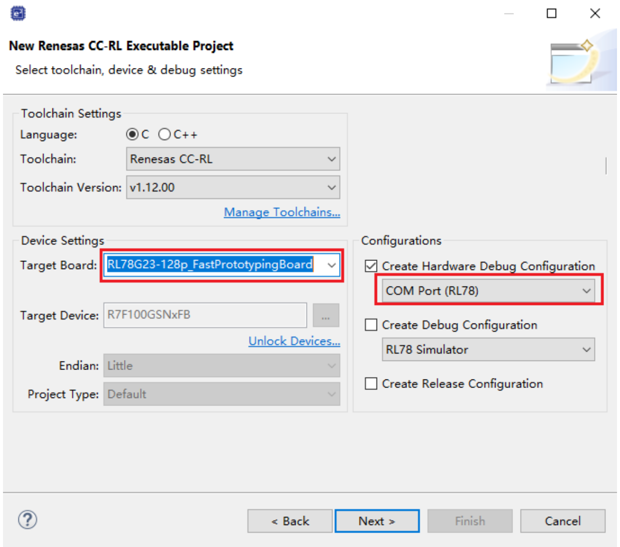
Conversion time mode: Normal 1

Conversion time: 2112/fCLK

Figure 6-15 [Change resource] operation

6.2.12 Note on changing Hardware Debug Configuration on project generation wizard

When a target board (except custom) is selected during creating a new project, please don't change the Hardware Debug Configuration manually. The reason is that the Hardware Debug Configuration has be decided by target board automatically. The user setting can't be reflected into Smart Configurator.



New Renesas CC-RL Executable Project
Select toolchain, device & debug settings

Toolchain Settings
Language: ● C ○ C++
Toolchain: Renesas CC-RL
Toolchain Version: v1.12.00
Manage Toolchains...

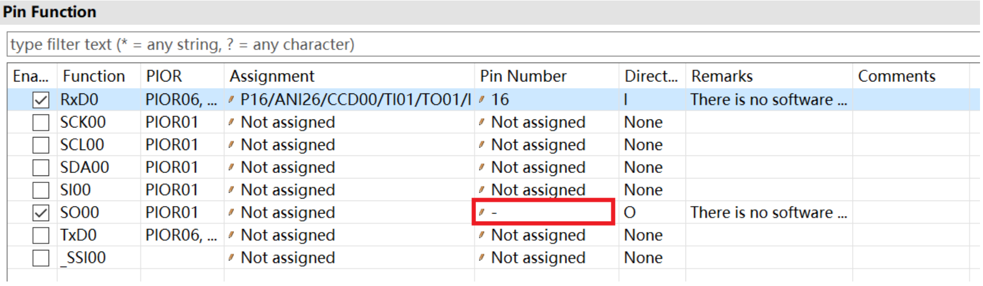
Device Settings
Target Board: RL78G23-128p_FastPrototypingBoard
Target Device: R7F100GSNxFB
Unlock Devices...
Endian: Little
Project Type: Default

Configurations
☑ Create Hardware Debug Configuration
COM Port (RL78)
☐ Create Debug Configuration
RL78 Simulator
☐ Create Release Configuration

Figure 6-16 Select a target board when creating a project in e^2 studio

6.2.13 Note on Pin Number maybe wrong in [Pins] page when loading project

The Pin Number maybe wrong for SCL00, SDA00, SI00, SO00, SCK00 when the user loads a 48/52/64pin project. The user needs to re-assign these pins manually.



Pin Function

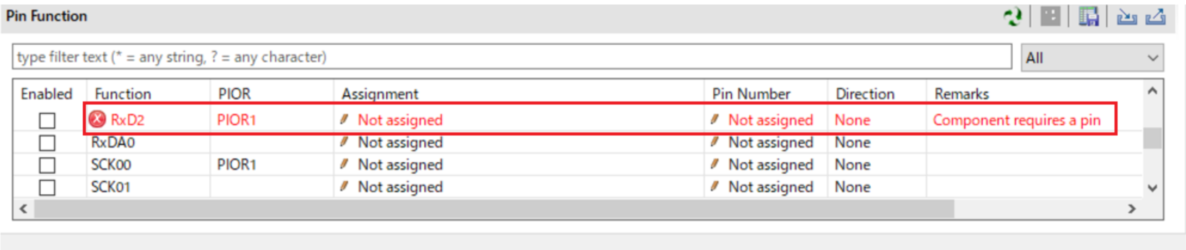
type filter text (* = any string, ? = any character)

Ena...	Function	PIOR	Assignment	Pin Number	Direct...	Remarks	Comments
☑	RxD0	PIOR06, ...	P16/ANI26/CCD00/TI01/TO01/I	16	I	There is no software ...	
☐	SCK00	PIOR01	Not assigned	Not assigned	None		
☐	SCL00	PIOR01	Not assigned	Not assigned	None		
☐	SDA00	PIOR01	Not assigned	Not assigned	None		
☐	SI00	PIOR01	Not assigned	Not assigned	None		
☑	SO00	PIOR01	Not assigned	-	O	There is no software ...	
☐	TxD0	PIOR06, ...	Not assigned	Not assigned	None		
☐	_SSI00		Not assigned	Not assigned	None		

Figure 6-17 Pin Number maybe wrong in [Pins] page

6.2.14 Note on Pin assignment with PIOR maybe wrong in [Pins] page when changing device

After performing [Change device] (for example, change from RL78/G23 to RL78/F24), the Pin assignment with PIOR may change according to PIOR setting automatically. When it has pin conflict, Smart Configurator will output pin conflict message and doesn't change pin assignment automatically. The user needs to re-assign these pins manually.



Pin Function

type filter text (* = any string, ? = any character)　All

Enabled	Function	PIOR	Assignment	Pin Number	Direction	Remarks
☐	RxD2	PIOR1	Not assigned	Not assigned	None	Component requires a pin
☐	RxDA0		Not assigned	Not assigned	None	
☐	SCK00	PIOR1	Not assigned	Not assigned	None	
☐	SCK01		Not assigned	Not assigned	None	

Figure 6-18 Pin assignment with PIOR maybe wrong in [Pins] page

6.2.15 Note on the case that a pin of an unsupported component is assigned when the device is changed

After performing [Change device] (for example, change from RL78/L23 to RL78/G15), the Pin assignment may keep “Enabled” though the component doesn’t exist. When the pin is enabled and exists in changed device, Smart Configurator doesn’t change pin assignment automatically. The user needs re-assign these pins manually.

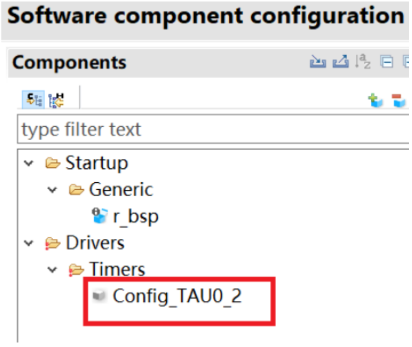
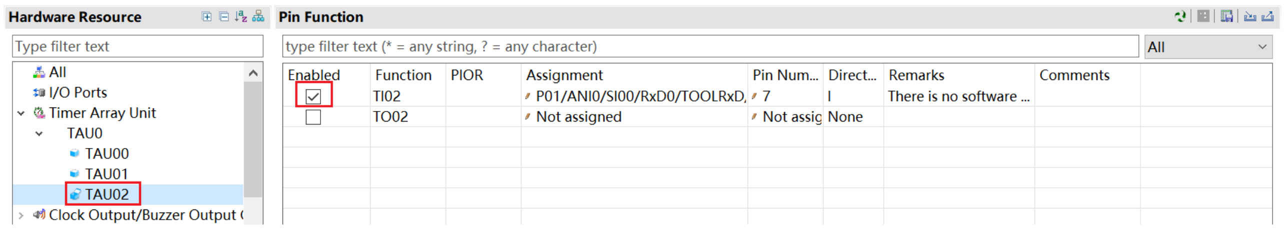


Figure 6-19 The component doesn’t exist after changing device

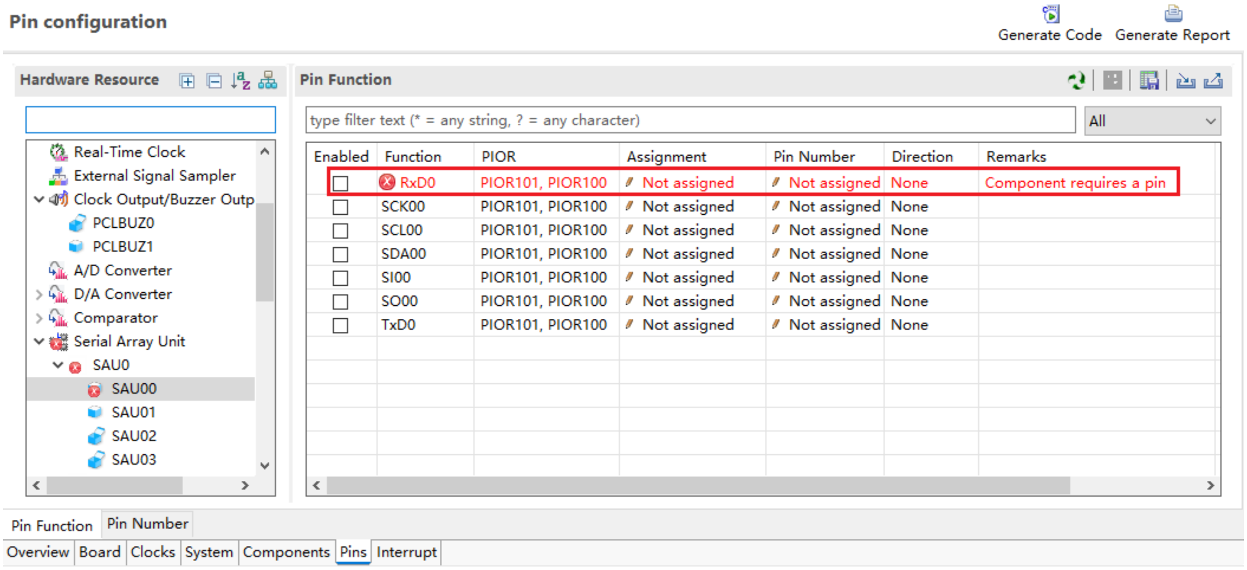


Enabled	Function	PIOR	Assignment	Pin Num...	Direct...	Remarks	Comments
☑	TI02		P01/ANI0/SI00/RxD0/TOOLRxD,	7	I	There is no software ...	
☐	TO02		Not assigned	Not assig	None		

Figure 6-20 Pin assignment kees “Enabled” in [Pins] page after changing device

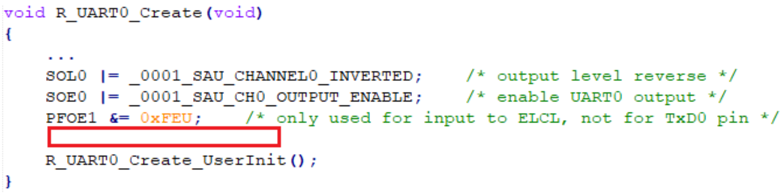
6.2.16 Note on the Pin assignement and pin setting maybe wrong when loading project

From Smart Configurator for RL78 V1.14.0, Smart Configurator supports RL78/L23 100 pin LQFP package for R7F100LPGxFA, R7F100LPJxFA and R7F100LPLxFA. If the user loads a project which is created by R7F100LPGxFA, R7F100LPJxFA and R7F100LPLxFA in Smart Configurator for RL78 V1.13.0, the Pin assignement and pin setting maybe wrong. The following is an example.



Enabled	Function	PIOR	Assignment	Pin Number	Direction	Remarks
☐	RxD0	PIOR101, PIOR100	Not assigned	Not assigned	None	Component requires a pin
☐	SCK00	PIOR101, PIOR100	Not assigned	Not assigned	None	
☐	SCL00	PIOR101, PIOR100	Not assigned	Not assigned	None	
☐	SDA00	PIOR101, PIOR100	Not assigned	Not assigned	None	
☐	SI00	PIOR101, PIOR100	Not assigned	Not assigned	None	
☐	SO00	PIOR101, PIOR100	Not assigned	Not assigned	None	
☐	TxD0	PIOR101, PIOR100	Not assigned	Not assigned	None	

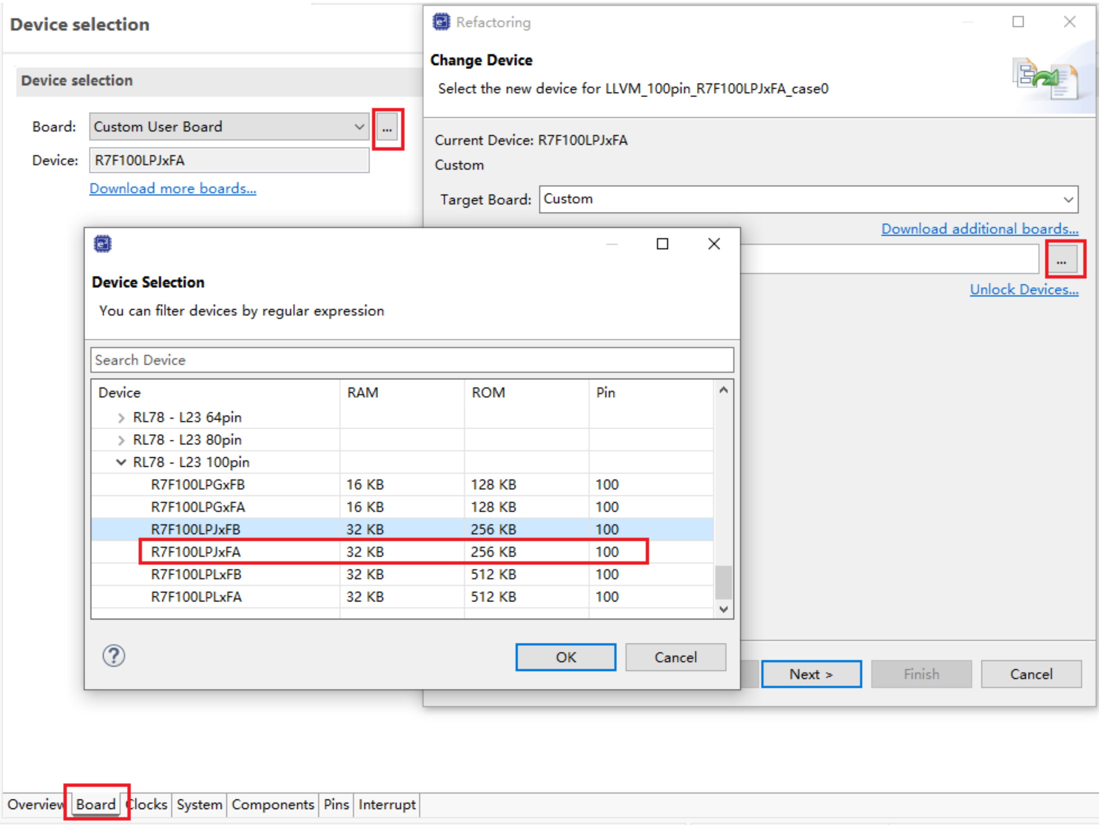
Figure 6-21 RxD0 isn’t assigned in [Pins] page



```
void R_UART0_Create(void)
{
    ...
    SOL0 |= _0001_SAU_CHANNEL0_INVERTED;    /* output level reverse */
    SOE0 |= _0001_SAU_CH0_OUTPUT_ENABLE;    /* enable UART0 output */
    PFOE1 &= 0xFEU;    /* only used for input to ELCL, not for TxD0 pin */

    R_UART0_Create_UserInit();
}
```

Figure 6-22 RxD0 pin setting isn't generated in Create()

[Workaround]
The user performs [Change device] (select same chip). And then the user should check and re-assign all the pins in [Pins] page manually. When the user generates code again, the pin setting will generate correctly.



Device selection

Device selection

Board: Custom User Board

Device: R7F100LPJxFA

Refactoring

Change Device

Select the new device for LLVM_100pin_R7F100LPJxFA_case0

Current Device: R7F100LPJxFA
Custom

Target Board: Custom

Device Selection

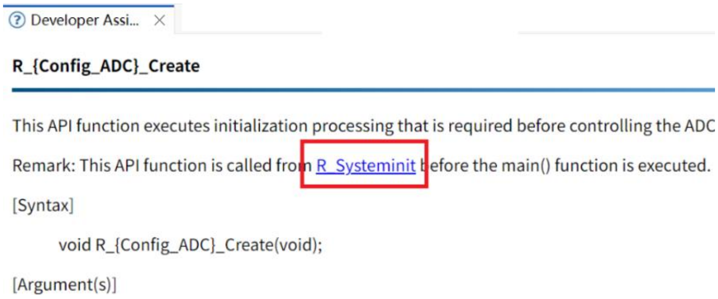
You can filter devices by regular expression

Device	RAM	ROM	Pin
RL78 - L23 64pin			
RL78 - L23 80pin			
RL78 - L23 100pin			
R7F100LPGxFB	16 KB	128 KB	100
R7F100LPGxFA	16 KB	128 KB	100
R7F100LPJxFB	32 KB	256 KB	100
R7F100LPJxFA	32 KB	256 KB	100
R7F100LPLxFB	32 KB	512 KB	100
R7F100LPLxFA	32 KB	512 KB	100

Figure 6-23 Change device in [Board] page

6.2.17 Note on the Developer Assistance can't jump correctly to another HTML file

When the user clicks the link in Developer Assistance, it can't jump correctly to another HTML file and will show "File not found".



R_{Config_ADC}_Create

This API function executes initialization processing that is required before controlling the ADC

Remark: This API function is called from R_Systeminit before the main() function is executed.

[Syntax]

void R_{Config_ADC}_Create(void);

[Argument(s)]

Figure 6-24 Can't jump correctly when clicking the link to another HTML file

[Workaround] Please click “Developer assistance” node in project tree and select the API function in Developer Assist Browser to check API guide information.

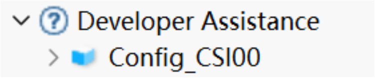


Figure 6-25 “Developer assistance” node

6.2.18 Note on the clock UI display issue when changing screen scale

The clock UI display incorrectly after changing the screen scale.

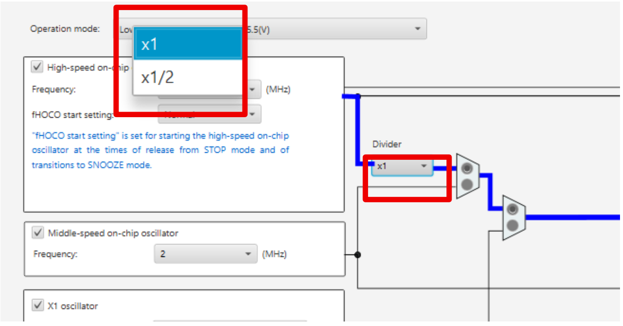
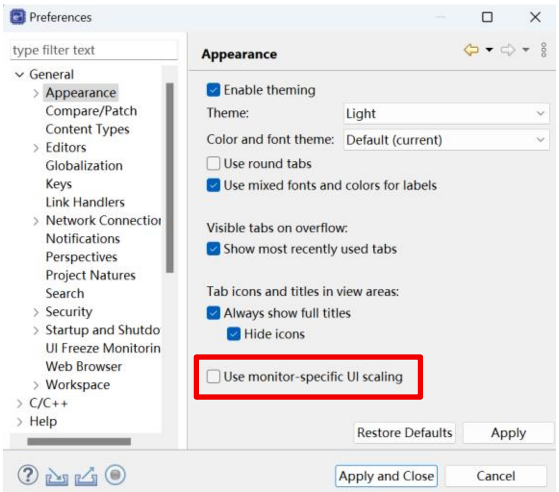

Operation mode:
x1
x1/2
High-speed on-chip
Frequency:
(MHz)
fHOCO start setting:
"fHOCO start setting" is set for starting the high-speed on-chip oscillator at the times of release from STOP mode and of transitions to SNOOZE mode.
Divider
x1
Middle-speed on-chip oscillator
Frequency:
2
(MHz)
X1 oscillator

Figure 6-26 UI display incorrectly when clicking

[Workaround]
Uncheck “Use monitor-specific UI scaling” in Preferences -> General -> Appearance in e^2 studio.



Appearance

☑ Enable theming
Theme: Light
Color and font theme: Default (current)
□ Use round tabs
☑ Use mixed fonts and colors for labels

Visible tabs on overflow:
☑ Show most recently used tabs

Tab icons and titles in view areas:
☑ Always show full titles
☑ Hide icons

□ Use monitor-specific UI scaling

Figure 6-27 Uncheck “Use monitor-specific UI scaling”

Revision History

Rev.	Section	Description
1.00	-	First edition issued

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

2. Processing at power-on

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

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

6. Voltage application waveform at input pin

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

www.renesas.com

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/.

Trademarks

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