Smart Configurator for RL78 Plug-in in e^2 studio 2024-07
Smart Configurator for RL78 V1.11.0
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

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 V1.11.0 is equivalent to Smart Configurator for RL78 Plug-in in e^2 studio 2024-07.

1.1 System Requirements

The operating environment is as follows.

1.1.1 Windows PC

- System: x64/x86 based processor
  Windows® 11
  Windows® 10 (64-bit version)
  Windows® 8.1 (64-bit version)
- 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 22.04 LTS Desktop (64-bit version)
  Ubuntu 20.04 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: 1.8 GHz or faster 64-bit processor. Dual-core or better recommended. Apple Silicon (arm64) processors are only supported.
  MacOS 13 (Ventura)
- 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] V1.14 or later (Windows PC)
- LLVM for Renesas RL78 17.0.1.202403 or later (Windows PC, Linux PC, Mac OS)
- IAR Embedded Workbench for Renesas RL78 V5.10.3 or later (Windows PC)
- SMS Assembler Note1 V1.00.00 or later (Windows PC)
- FAA Assembler Note1 V1.04.02 or later (Windows PC)
- CS+ for CC V8.12.00 Note2 or later (Windows PC)

Note:

1.If you want to add SMS Assembler or FAA Assembler to e^2 studio, install it from the integrated installer of e^2 studio 21-04 or later. (e^2 studio)

As with other compilers, select and install from the [Additional Software] - [Renesas Toolchains & Utilities] tab of the e^2 studio setup wizard.

2.Smart Configurator for RL78 V1.11.0 has been evaluated in the CS+ for CC V8.12.00 environment. When using Smart Configurator for RL78 V1.10.0 or lower, please refer to Release Note (R20UT5473EC0100) about the target version of CS+ for CC.

2. Support List

2.1 Support Devices List

Below is a list of devices supported by the Smart Configurator for RL78 V1.11.0.

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

2.2 Support Components List

Below is a list of Components supported by the Smart Configurator for RL78 V1.11.0.

Table 2-3 Support Components (1/2) ✓: 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 (2/2) ✓: 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	−	−	−	−	−	−	✓	-	
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	-	✓	−	−	−	−	−	−	-	Need download in Smart Configurator RL78
38	Phase Counting Mode	-	−	−	−	−	−	−	✓	-	
39	Programmable Gain Amplifier	-	−	−	−	−	−	−	✓	-	
40	Flexible Application Accelerator	-	−	−	−	−	−	−	✓	-	

2.3 New support

2.3.1 BSP (Board Support Package) revision update

BSP rev1.70 is supported and will be added as default BSP when creating Smart Configurator project.

2.3.2 Support chip R5F12008xSN for RL78/G15

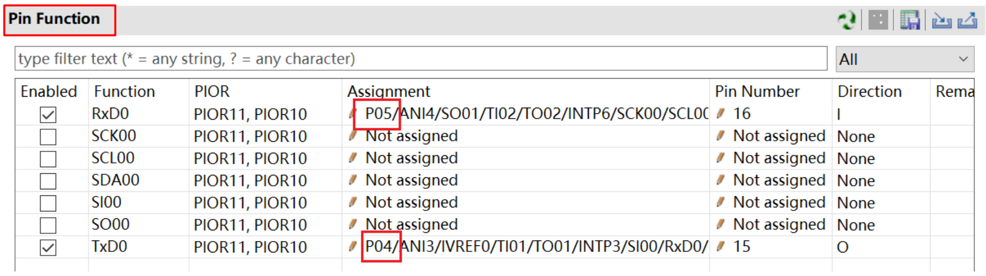
See 2.1 Support Devices List for details on supported packages

2.3.3 Support RL78/F25 devices

See 2.1 Support Devices List for details on supported packages.

2.3.4 Support macro definitions for the pin with PIOR setting into Pin.h

From Smart Configurator for RL78 V1.11.0, new feature to define macro definitions is supported in Pin.h when generating code. Only pin assignments that have PIOR registers will have macro definitions generated.

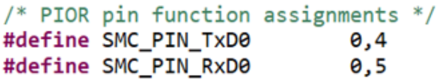


Pin Function

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

Enabled	Function	PIOR	Assignment	Pin Number	Direction	Rema
☑	RxD0	PIOR11, PIOR10	P05/ANI4/SO01/TI02/TO02/INTP6/SCK00/SCL0C	16	I	
☐	SCK00	PIOR11, PIOR10	Not assigned	Not assigned	None	
☐	SCL00	PIOR11, PIOR10	Not assigned	Not assigned	None	
☐	SDA00	PIOR11, PIOR10	Not assigned	Not assigned	None	
☐	SI00	PIOR11, PIOR10	Not assigned	Not assigned	None	
☐	SO00	PIOR11, PIOR10	Not assigned	Not assigned	None	
☑	TxD0	PIOR11, PIOR10	P04/ANI3/IVREF0/TI01/TO01/INTP3/SI00/RxD0/	15	O	

Figure 2-1 Assign pin functions with PIOR settings in [Pins] page

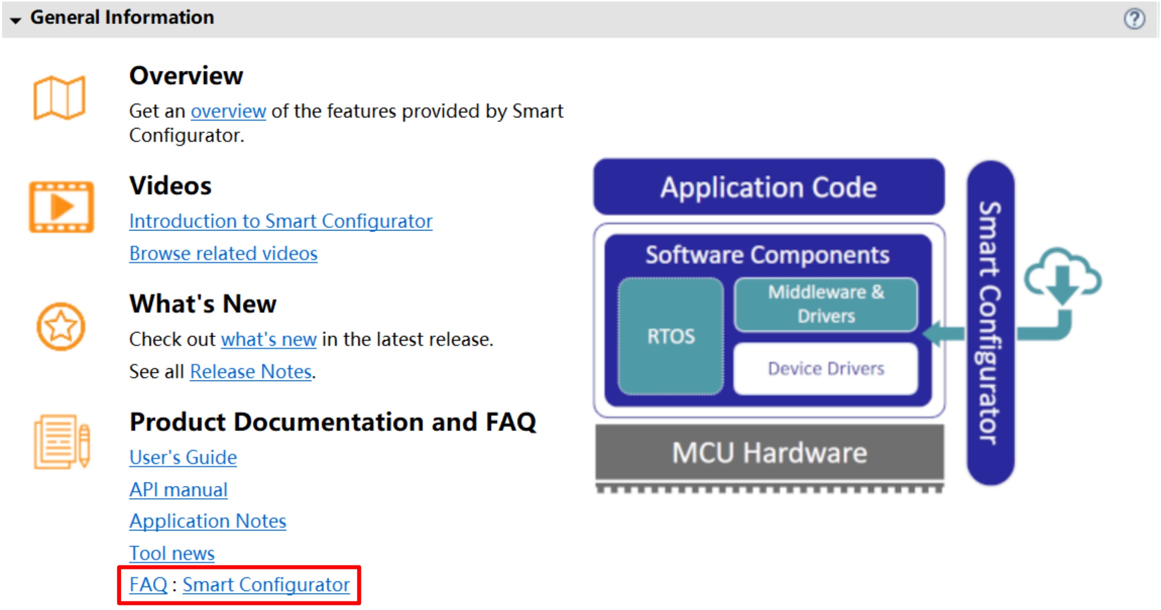


```
/* PIOR pin function assignments */
#define SMC_PIN_TxD0        0,4
#define SMC_PIN_RxD0        0,5
```

Figure 2-2 Code in pin.h

2.3.5 Support to access FAQ from [Overview] page

From Smart Configurator for RL78 V1.11.0, FAQ can be accessed from [Overview] page.



General Information

Overview

Get an overview of the features provided by Smart Configurator.

Videos

Introduction to Smart Configurator

Browse related videos

What's New

Check out what's new in the latest release.

See all Release Notes.

Product Documentation and FAQ

User's Guide

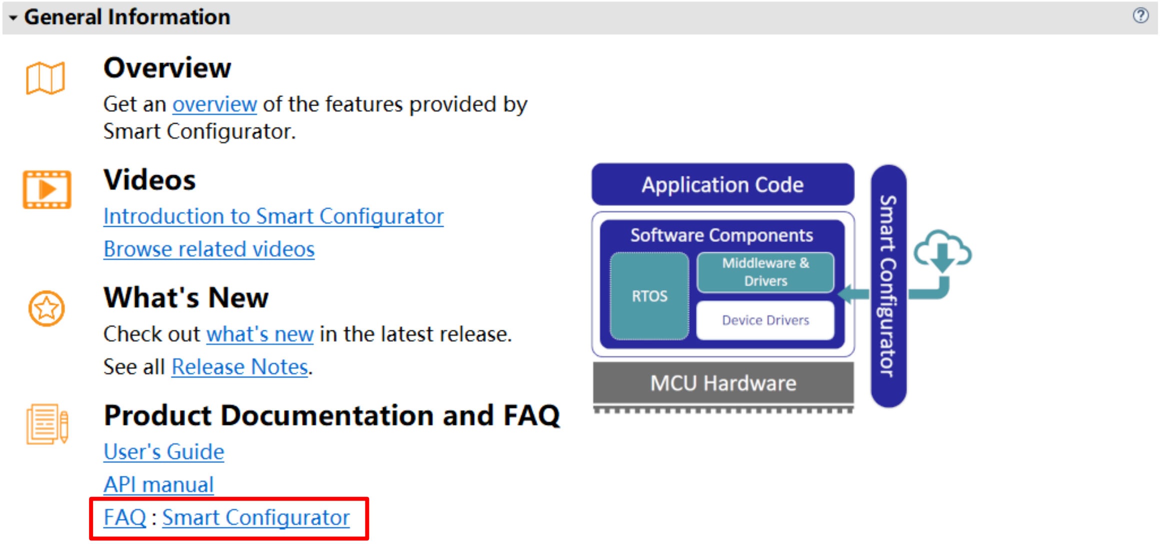
API manual

Application Notes

Tool news

FAQ : Smart Configurator

Figure 2-3 [Overview] page in e^2 studio



General Information

Overview

Get an overview of the features provided by Smart Configurator.

Videos

Introduction to Smart Configurator

Browse related videos

What's New

Check out what's new in the latest release.

See all Release Notes.

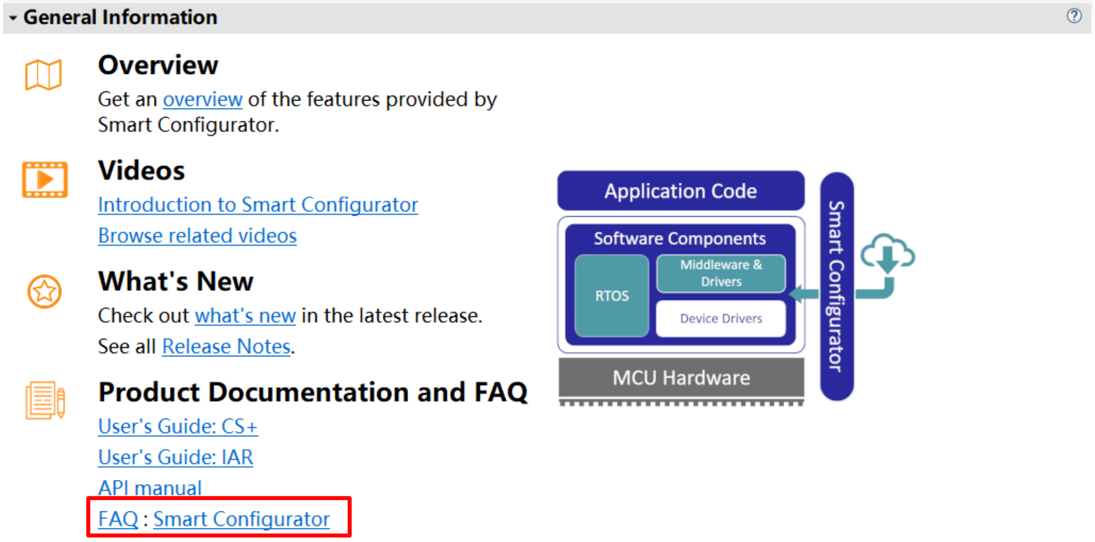
Product Documentation and FAQ

User's Guide

API manual

FAQ : Smart Configurator

Figure 2-4 [Overview] page in standalone Smart Configurator for CCRL78/IAR toolchain



General Information

Overview

Get an overview of the features provided by Smart Configurator.

Videos

Introduction to Smart Configurator

Browse related videos

What's New

Check out what's new in the latest release.

See all Release Notes.

Product Documentation and FAQ

User's Guide: CS+

User's Guide: IAR

API manual

FAQ : Smart Configurator

Figure 2-5 [Overview] page in standalone Smart Configurator for LLVM toolchain

2.3.6 Support Import/Export function by Flexible Application Accelerator component with enhancement

From Smart Configurator for RL78 V1.11.0, the function of Import/Export in [Component] page is applicable for Flexible Application Acceleration component. User can export current FAA configuration to XML file. The exported XML file can be imported to another project.
In addition, the user program added in generated Config_FAA_src.dsp file when selected "Template" function in "Custom Library" also support Import/Export function.

2.3.7 Simplify MCU/MPU Package view

From Smart Configurator for RL78 V1.11.0, when using QFP device, MCU/MPU Package view is simplified to remove the outer pad by default for easy recognition of user changes. If the user wants to check the pin input/output information, please select "I/O direction" or "I/O direction in outer pad".

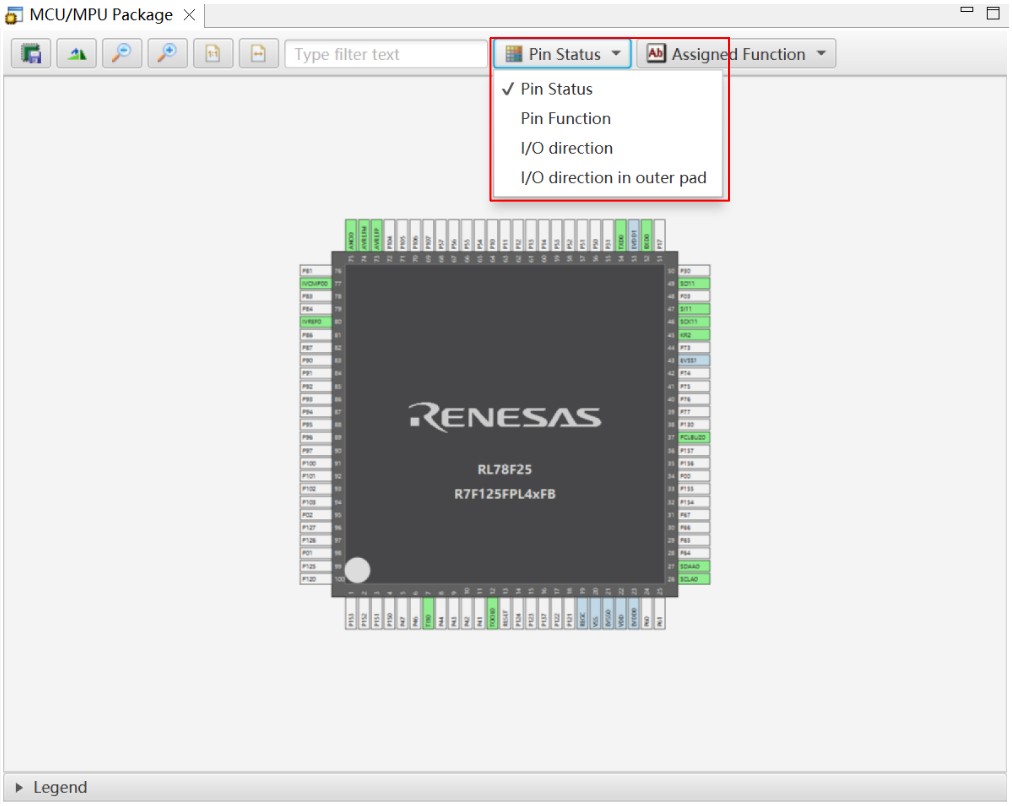


Figure 2-6 MCU/MPU Package view without outer pad setting

3. Changes

This chapter describes changes to the Smart Configurator for RL78 V1.11.0.

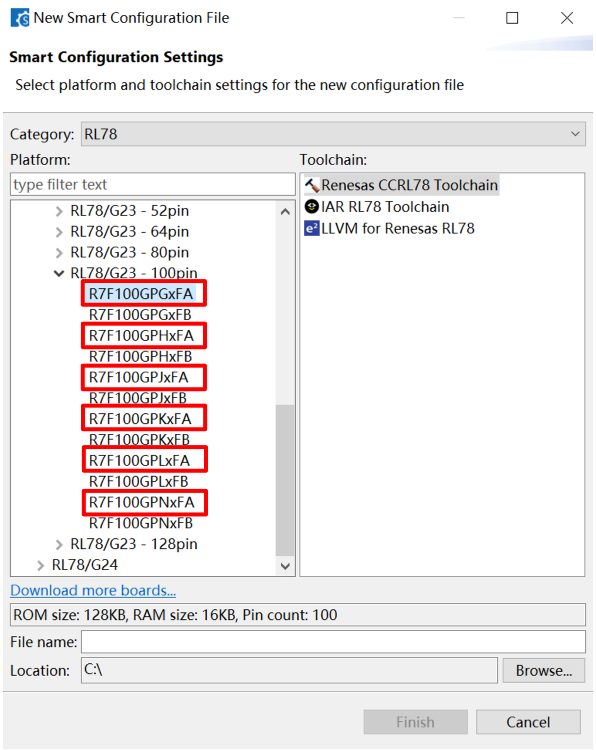
3.1 Correction of issues/limitations

Table 3-1 List of Correction of issues/limitations ✓: 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 displaying wrong top view picture in [MCU/MPU Package] page and wrong pin number information in [Pins] page	✓	–	–	–	–	–	–	–	
2	Fixed the issue of displaying the uncorrent status of unsupported components (FAA, SMS, ELCL) in [Overview] page after changing device	✓	✓	✓	✓	✓	✓	✓	✓	
3	Fixed the issue of the pin information in [Pins] page maybe wrong after changing device	✓	✓	✓	✓	✓	✓	✓	✓	
4	Fixed the issue of the pin IVREF0 and IVREF1 are conflicted among Comparator 0 - 3	–	–	–	–	–	–	✓	–	
5	Fixed the issue of the API R_Config_RTC_Get_CounterValue () and R_Config_RTC_Set_CounterValue () are unable to break out of the loop when using Alarm function	✓	–	–	–	✓	✓	✓	–	

3.1.1 Fixed the issue of displaying wrong top view picture in [MCU/MPU Package] page and wrong pin number information in [Pins] page

If the user creates a RL78/G23 100-pin plastic LQFP project with chip name as R7F100GPXxFA (X – G, J, L, H, K, N), the top view picture is wrong in [MCU/MPU Package] page and the pin number information is wrong in [Pins] page. From Smart Configurator for RL78 V1.11.0, this issue is fixed.



New Smart Configuration File

Smart Configuration Settings

Select platform and toolchain settings for the new configuration file

Category: RL78

Platform:

Toolchain:

type filter text

- RL78/G23 - 52pin
- RL78/G23 - 64pin
- RL78/G23 - 80pin
- RL78/G23 - 100pin
  - R7F100GPGxFA
  - R7F100GPGxFB
  - R7F100GPHxFA
  - R7F100GPHxFB
  - R7F100GPJxFA
  - R7F100GPJxFB
  - R7F100GPKxFA
  - R7F100GPKxFB
  - R7F100GPLxFA
  - R7F100GPLxFB
  - R7F100GPNxFA
  - R7F100GPNxFB
- RL78/G23 - 128pin
- RL78/G24

Renesas CCRL78 Toolchain
IAR RL78 Toolchain
LLVM for Renesas RL78

Download more boards...

ROM size: 128KB, RAM size: 16KB, Pin count: 100

File name:

Location: C:\

Figure 3-1 The chips which are 100-pin plastic LQFP

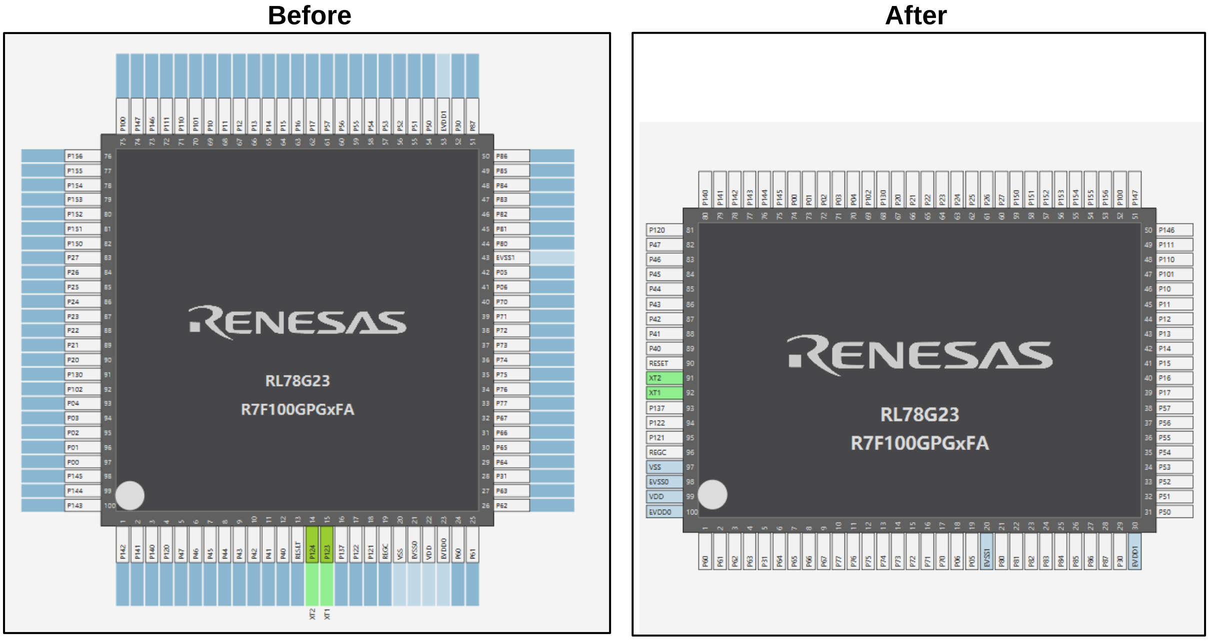
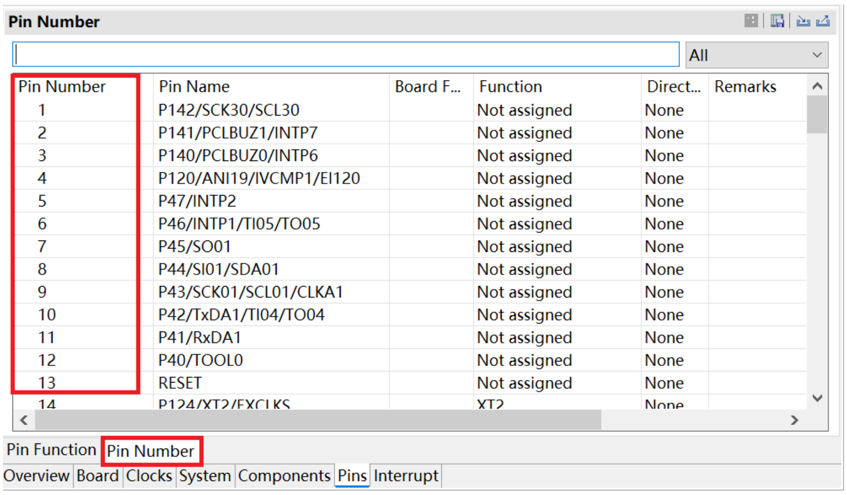


Figure 3-2 The top view picture in [MCU/MPU Package] page

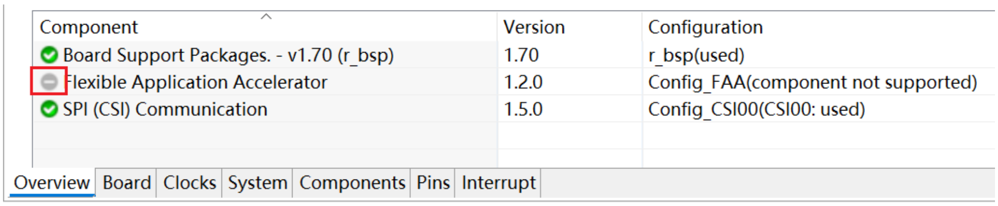


Pin Number	Pin Name	Board F...	Function	Direct...	Remarks
1	P142/SCK30/SCL30		Not assigned	None	
2	P141/PCLBUZ1/INTP7		Not assigned	None	
3	P140/PCLBUZ0/INTP6		Not assigned	None	
4	P120/ANI19/IVCMP1/EI120		Not assigned	None	
5	P47/INTP2		Not assigned	None	
6	P46/INTP1/TI05/TO05		Not assigned	None	
7	P45/SO01		Not assigned	None	
8	P44/SI01/SDA01		Not assigned	None	
9	P43/SCK01/SCL01/CLKA1		Not assigned	None	
10	P42/TxDA1/TI04/TO04		Not assigned	None	
11	P41/RxDA1		Not assigned	None	
12	P40/TOOL0		Not assigned	None	
13	RESET		Not assigned	None	
14	P124/XT2/EXCLKS		XT2	None	

Figure 3-3 The pin number information in [Pins] page

3.1.2 Fixed the issue of displaying the uncorrent status of unsupported components (FAA, SMS, ELCL) in [Overview] page after changing device

After changing device from the one in which FAA (Flexible Application Accelerator), SMS (SNOOZE Mode Sequencer) or ELCL is supported and added to the one which doesn't support these unsupported components, the status of FAA, SMS and ELCL are wrong in [Overview] page. They should be displayed with gray-off icon. From Smart Configurator for RL78 V1.11.0, this issue is fixed.

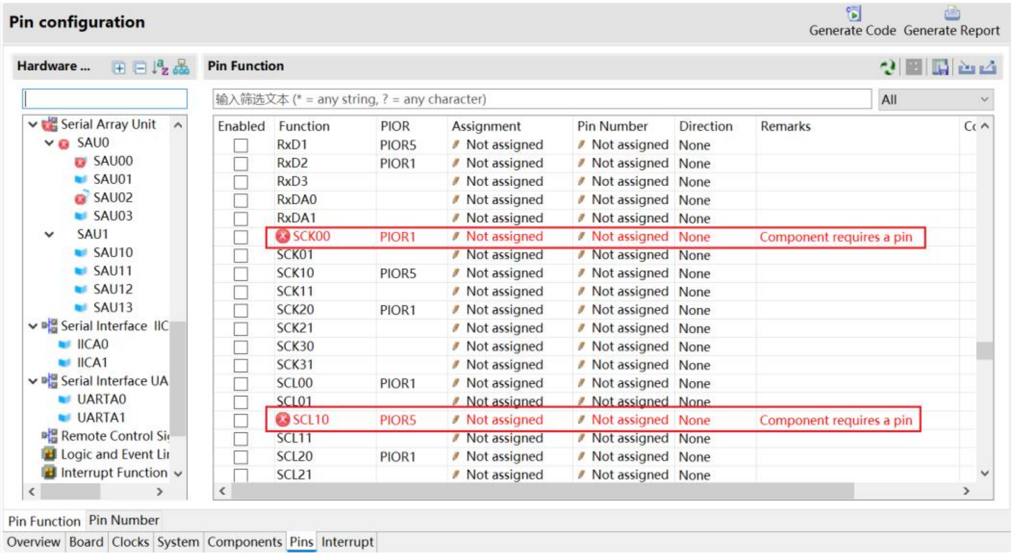


Component	Version	Configuration
✓ Board Support Packages. - v1.70 (r_bsp)	1.70	r_bsp(used)
⊖ Flexible Application Accelerator	1.2.0	Config_FAA(component not supported)
✓ SPI (CSI) Communication	1.5.0	Config_CSI00(CSI00: used)

Figure 3-4 The unsupported components should display with gray-off icon

3.1.3 Fixed the issue of the pin information in [Pins] page maybe wrong after changing device

If the user changes the device when the Smart Configurator editor is closed in e^2 studio, the pin information in [Pins] page maybe wrong after changing device. From Smart Configurator for RL78 V1.11.0, this issue is fixed.



Pin configuration

Hardware ...

- Serial Array Unit
  - SAU0
    - SAU00
    - SAU01
    - SAU02
    - SAU03
  - SAU1
    - SAU10
    - SAU11
    - SAU12
    - SAU13
- Serial Interface IIC
  - IICA0
  - IICA1
- Serial Interface UA
  - UARTA0
  - UARTA1
- Remote Control Si
- Logic and Event Li
- Interrupt Function

Pin Function

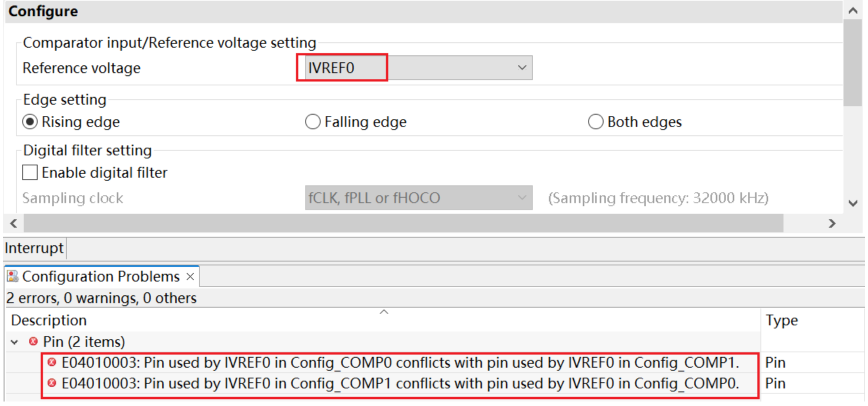
输入筛选文本 (* = any string, ? = any character)   All

Enabled	Function	PIOR	Assignment	Pin Number	Direction	Remarks	Cc
□	RxD1	PIOR5	Not assigned	Not assigned	None		
□	RxD2	PIOR1	Not assigned	Not assigned	None		
□	RxD3		Not assigned	Not assigned	None		
□	RxDA0		Not assigned	Not assigned	None		
□	RxDA1		Not assigned	Not assigned	None		
□	SCK00	PIOR1	Not assigned	Not assigned	None	Component requires a pin	
□	SCK01		Not assigned	Not assigned	None		
□	SCK10	PIOR5	Not assigned	Not assigned	None		
□	SCK11		Not assigned	Not assigned	None		
□	SCK20	PIOR1	Not assigned	Not assigned	None		
□	SCK21		Not assigned	Not assigned	None		
□	SCK30		Not assigned	Not assigned	None		
□	SCK31		Not assigned	Not assigned	None		
□	SCL00	PIOR1	Not assigned	Not assigned	None		
□	SCL01		Not assigned	Not assigned	None		
□	SCL10	PIOR5	Not assigned	Not assigned	None	Component requires a pin	
□	SCL11		Not assigned	Not assigned	None		
□	SCL20	PIOR1	Not assigned	Not assigned	None		
□	SCL21		Not assigned	Not assigned	None		

Figure 3-5 Incorrect pin assignment in [Pins] page

3.1.4 Fixed the issue of the pin IVREF0 and IVREF1 are conflicted among Comparator 0 - 3

If the user uses the pin IVREF0 in Comparator 0 - 3 at the same time, Smart Configrator displays pin conflict message. If the user uses the pin IVREF1 in Comparator 2 - 3 at the same time, Smart Configrator displays pin conflict message. These pin conflict messages should be removed. From Smart Configurator for RL78 V1.11.0, this issue is fixed.



Configure

Comparator input/Reference voltage setting

Reference voltage: IVREF0

Edge setting

● Rising edge ○ Falling edge ○ Both edges

Digital filter setting

□ Enable digital filter

Sampling clock: fCLK, fPLL or fHOCO (Sampling frequency: 32000 kHz)

Configuration Problems

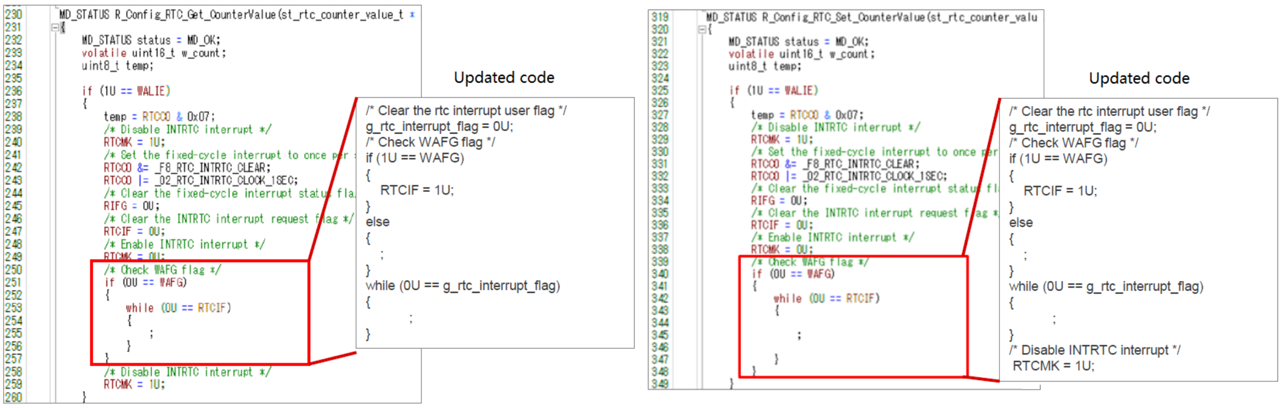
2 errors, 0 warnings, 0 others

Description	Type
Pin (2 items)	
E04010003: Pin used by IVREF0 in Config_COMP0 conflicts with pin used by IVREF0 in Config_COMP1.	Pin
E04010003: Pin used by IVREF0 in Config_COMP1 conflicts with pin used by IVREF0 in Config_COMP0.	Pin

Figure 3-6 The pin conflict messages

3.1.5 Fixed the issue of the API R_Config_RTC_Get_CounterValue () and R_Config_RTC_Set_CounterValue () are unable to break out of the loop when using Alarm function

The driver code in API R_Config_RTC_Get_CounterValue () and R_Config_RTC_Set_CounterValue () are unable to break out of the loop when using Alarm function. From Smart Configurator for RL78 V1.11.0, this issue is fixed.



```
MD_STATUS R_Config_RTC_Get_CounterValue(st_rtc_counter_value_t *
{
    MD_STATUS status = MD_OK;
    volatile uint16_t w_count;
    uint8_t temp;

    if (1U == WALIE)
    {
        temp = RTCC0 & 0x07;
        /* Disable INTRTC interrupt */
        RTCMK = 1U;
        /* Set the fixed-cycle interrupt to once per
        RTCC0 &= _F8_RTC_INTRTC_CLEAR;
        RTCC0 |= _02_RTC_INTRTC_CLOCK_1SEC;
        /* Clear the fixed-cycle interrupt status fla
        RIFG = 0U;
        /* Clear the INTRTC interrupt request flag */
        RTCIF = 0U;
        /* Enable INTRTC interrupt */
        RTCMK = 0U;
        /* Check WAFG flag */
        if (0U == WAFG)
        {
            while (0U == RTCIF)
            {
                ;
            }
        }
        /* Disable INTRTC interrupt */
        RTCMK = 1U;
    }
```

Updated code

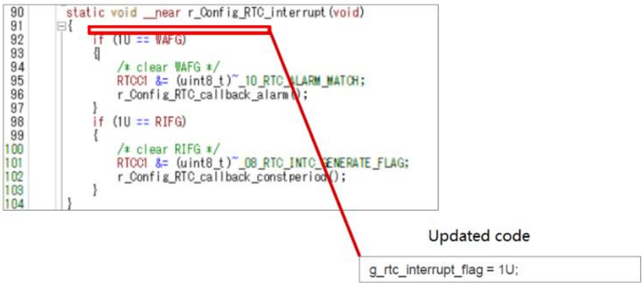
```
/* Clear the rtc interrupt user flag */
g_rtc_interrupt_flag = 0U;
/* Check WAFG flag */
if (1U == WAFG)
{
   RTCIF = 1U;
}
else
{
   ;
}
while (0U == g_rtc_interrupt_flag)
{
          ;
}
```

```
MD_STATUS R_Config_RTC_Set_CounterValue(st_rtc_counter_valu
{
    MD_STATUS status = MD_OK;
    volatile uint16_t w_count;
    uint8_t temp;

    if (1U == WALIE)
    {
        temp = RTCC0 & 0x07;
        /* Disable INTRTC interrupt */
        RTCMK = 1U;
        /* Set the fixed-cycle interrupt to once per
        RTCC0 &= _F8_RTC_INTRTC_CLEAR;
        RTCC0 |= _02_RTC_INTRTC_CLOCK_1SEC;
        /* Clear the fixed-cycle interrupt status fl
        RIFG = 0U;
        /* Clear the INTRTC interrupt request flag *
        RTCIF = 0U;
        /* Enable INTRTC interrupt */
        RTCMK = 0U;
        /* Check WAFG flag */
        if (0U == WAFG)
        {
            while (0U == RTCIF)
            {

                ;

            }
        }
    }
```

Updated code

```
/* Clear the rtc interrupt user flag */
g_rtc_interrupt_flag = 0U;
/* Check WAFG flag */
if (1U == WAFG)
{
   RTCIF = 1U;
}
else
{
   ;
}
while (0U == g_rtc_interrupt_flag)
{
          ;
}
/* Disable INTRTC interrupt */
 RTCMK = 1U;
```

Figure 3-7 The driver code with red frame is updated



```
static void __near r_Config_RTC_interrupt(void)
{
    if (1U == WAFG)
    {
        /* clear WAFG */
        RTCC1 &= (uint8_t)~_10_RTC_ALARM_MATCH;
        r_Config_RTC_callback_alarm();
    }
    if (1U == RIFG)
    {
        /* clear RIFG */
        RTCC1 &= (uint8_t)~_08_RTC_INTC_GENERATE_FLAG;
        r_Config_RTC_callback_constperiod();
    }
}
```

Updated code

```
g_rtc_interrupt_flag = 1U;
```

Add code for setting flag.

Figure 3-8 The API r_Config_RTC_interrupt () is updated

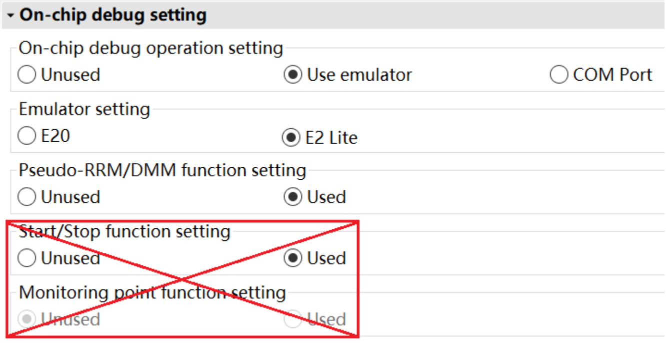
3.2 Specification changes

Table 3-2 List of Specification changes ✓: 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 changing "debug monitor" specification	–	–	✓	–	–	✓	–	–	
2	Improvement for opening "My Renesas log-in" dialog when the user want to download RL78 Software Integration System modules without logging-in to "My Renesas" account	✓	✓	✓	✓	✓	✓	✓	✓	
3	Improvement for updating the spec of Timer RD Clock in [Clock] page	–	✓	–	✓	–	–	–	✓	
4	Improvement for using alarm interrupt INTRTC when using alarm detection function	✓	–	–	–	✓	✓	✓	–	
5	Improvement for changing compiler property and debugger property according to "Pseudo-RRM/DMM function setting"	✓	✓	✓	✓	✓	✓	✓	✓	
6	Improvement for high speed on chip oscillator clock cannot be used as the PLL input clock	–	–	–	–	–	–	–	✓	
7	Improve blinky sample project for e^2 studio	✓	✓	✓	✓	✓	✓	✓	✓	

3.2.1 Improvement for changing "debug monitor" specification

Remove "Start/Stop function setting" and "Monitoring point function setting" from [System] page. When "On-chip debug operation setting" is changed, compiler property will be changed after generating code.



On-chip debug setting

On-chip debug operation setting
○ Unused ● Use emulator ○ COM Port

Emulator setting
○ E20 ● E2 Lite

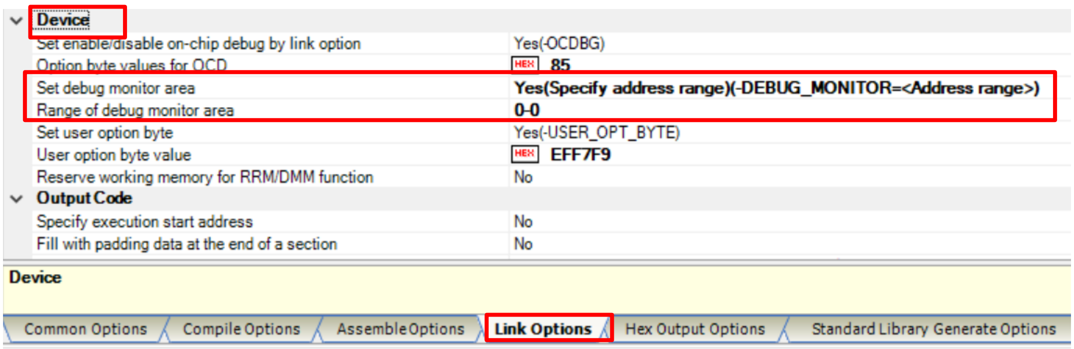
Pseudo-RRM/DMM function setting
○ Unused ● Used

Start/Stop function setting
○ Unused ● Used

Monitoring point function setting
● Unused ○ Used

Figure 3-9 Remove "Start/Stop function setting" and "Monitoring point function setting" from [System] page

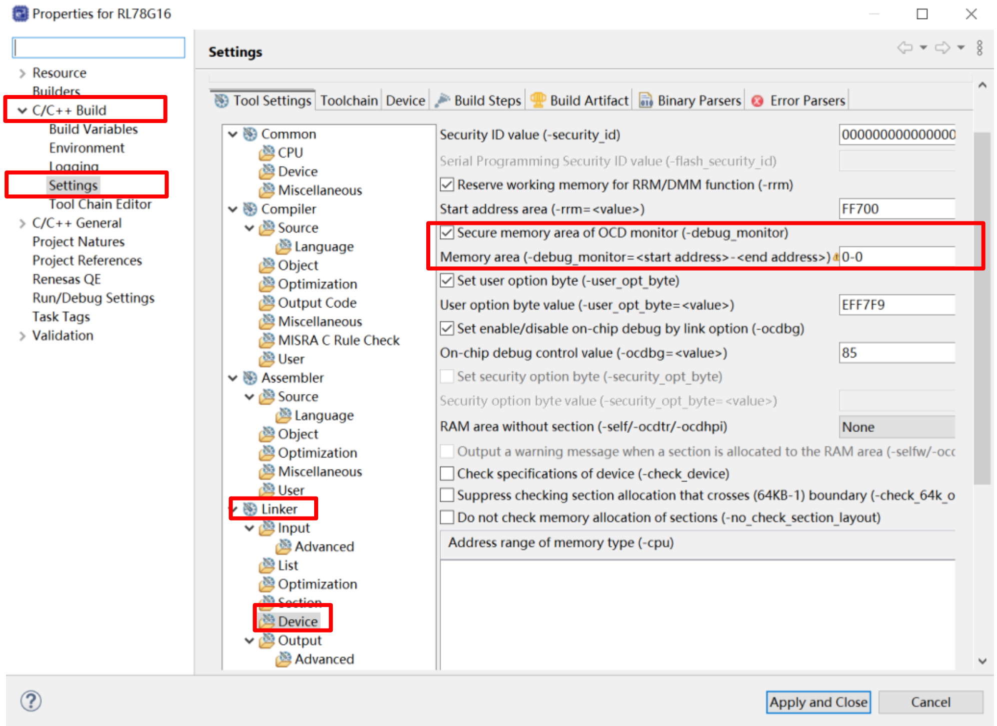
Smart Configurator will set "debug monitor" when setting "On-chip debug operation setting" to "Use emulator" or "COM Port".



Device	
Set enable/disable on-chip debug by link option	Yes(-OCDBG)
Option byte values for OCD	HEX 85
Set debug monitor area	Yes(Specify address range)(-DEBUG_MONITOR=<Address range>)
Range of debug monitor area	0-0
Set user option byte	Yes(-USER_OPT_BYTE)
User option byte value	HEX EFF7F9
Reserve working memory for RRM/DMM function	No
Output Code	
Specify execution start address	No
Fill with padding data at the end of a section	No

Device

Figure 3-10 Set compiler property "debug monitor" in CS+



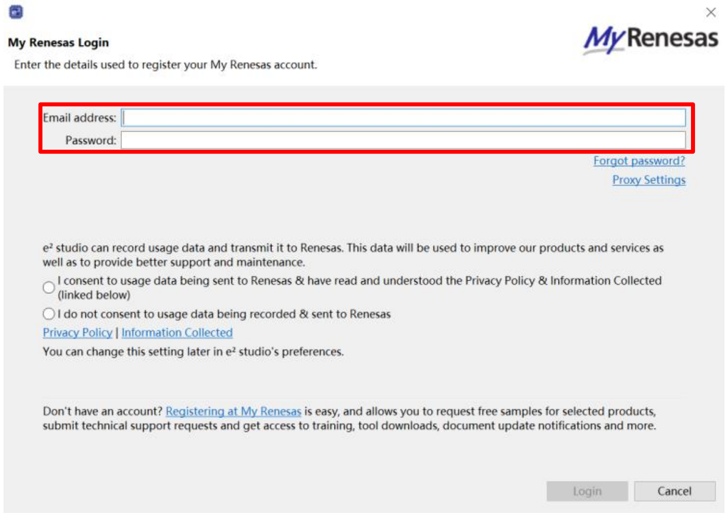
Security ID value (-security_id) 000000000000000
Serial Programming Security ID value (-flash_security_id)
☑ Reserve working memory for RRM/DMM function (-rrm)
Start address area (-rrm=<value>) FF700
☑ Secure memory area of OCD monitor (-debug_monitor)
Memory area (-debug_monitor=<start address>-<end address>) 0-0
☑ Set user option byte (-user_opt_byte)
User option byte value (-user_opt_byte=<value>) EFF7F9
☑ Set enable/disable on-chip debug by link option (-ocdbg)
On-chip debug control value (-ocdbg=<value>) 85
□ Set security option byte (-security_opt_byte)
Security option byte value (-security_opt_byte=<value>)
RAM area without section (-self/-ocdtr/-ocdhpi) None
□ Output a warning message when a section is allocated to the RAM area (-selfw/-occ
□ Check specifications of device (-check_device)
□ Suppress checking section allocation that crosses (64KB-1) boundary (-check_64k_o
□ Do not check memory allocation of sections (-no_check_section_layout)
Address range of memory type (-cpu)

Figure 3-11 Set compiler property "debug monitor" in e^2 studio

Note: The supported compilers only include CCRL.

3.2.2 Improvement for opening "My Renesas log-in" dialog when the user want to download RL78 Software Integration System modules without logging-in to "My Renesas" account

From Smart Configurator for RL78 V1.11.0, when user downloads RL78 Software Integration System Modules after launching Smart Configurator, "My Renesas log-in" dialogue will automatically display if user does not log-in to "My Renesas" account. The user can download RL78 Software Integration System Modules after "My Renesas" account is logged-in.



My Renesas Login

Enter the details used to register your My Renesas account.

Email address:

Password:

Forgot password?

Proxy Settings

e² studio can record usage data and transmit it to Renesas. This data will be used to improve our products and services as well as to provide better support and maintenance.

○ I consent to usage data being sent to Renesas & have read and understood the Privacy Policy & Information Collected (linked below)

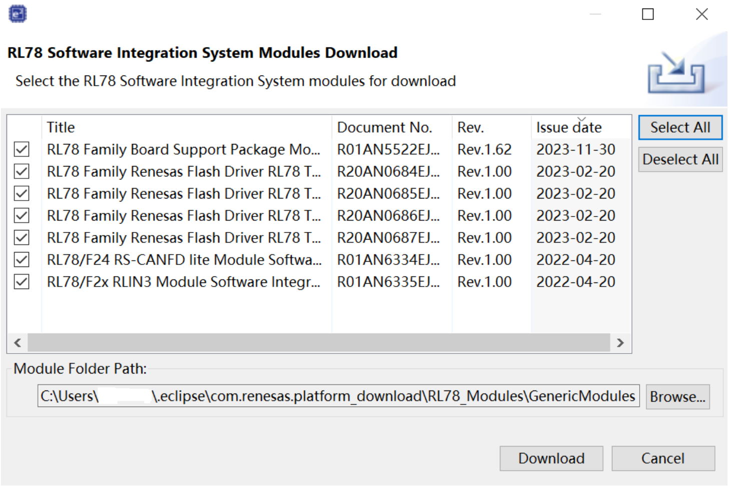
○ I do not consent to usage data being recorded & sent to Renesas

Privacy Policy | Information Collected

You can change this setting later in e² studio's preferences.

Don't have an account? Registering at My Renesas is easy, and allows you to request free samples for selected products, submit technical support requests and get access to training, tool downloads, document update notifications and more.

Figure 3-12 "My Renesas log-in" dialogue



RL78 Software Integration System Modules Download

Select the RL78 Software Integration System modules for download

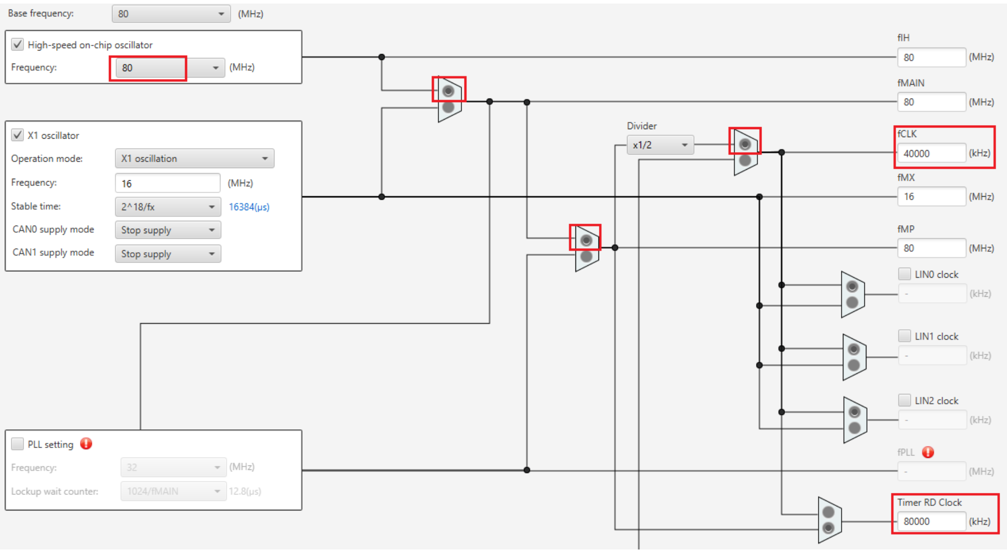
	Title	Document No.	Rev.	Issue date
☑	RL78 Family Board Support Package Mo...	R01AN5522EJ...	Rev.1.62	2023-11-30
☑	RL78 Family Renesas Flash Driver RL78 T...	R20AN0684EJ...	Rev.1.00	2023-02-20
☑	RL78 Family Renesas Flash Driver RL78 T...	R20AN0685EJ...	Rev.1.00	2023-02-20
☑	RL78 Family Renesas Flash Driver RL78 T...	R20AN0686EJ...	Rev.1.00	2023-02-20
☑	RL78 Family Renesas Flash Driver RL78 T...	R20AN0687EJ...	Rev.1.00	2023-02-20
☑	RL78/F24 RS-CANFD lite Module Softwa...	R01AN6334EJ...	Rev.1.00	2022-04-20
☑	RL78/F2x RLIN3 Module Software Integr...	R01AN6335EJ...	Rev.1.00	2022-04-20

Module Folder Path: C:\Users\ \.eclipse\com.renesas.platform_download\RL78_Modules\GenericModules

Figure 3-13 "RL78 Software Integration System Modules Download" dialogue

3.2.3 Improvement for updating the spec of Timer RD Clock in [Clock] page

When the user selects 80 MHz or 64 MHz from fIH to Timer RD, fCLK should come from fIH. When the user selects 80 MHz or 64 MHz from fPLL to Timer RD, fCLK should come from fPLL.



Base frequency: 80 (MHz)

☑ High-speed on-chip oscillator
Frequency: 80 (MHz)

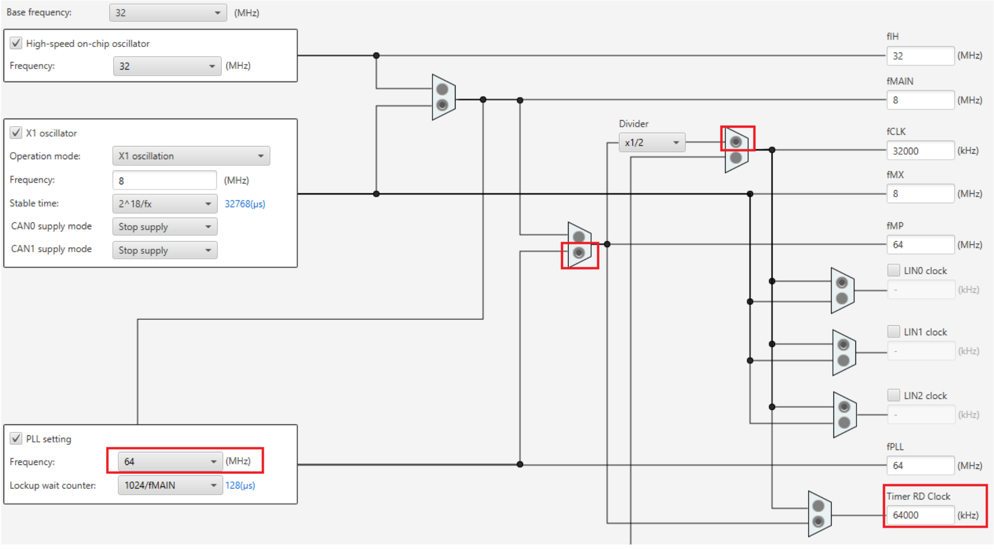
☑ X1 oscillator
Operation mode: X1 oscillation
Frequency: 16 (MHz)
Stable time: 2^18/fx 16384(μs)
CAN0 supply mode Stop supply
CAN1 supply mode Stop supply

□ PLL setting
Frequency: 32 (MHz)
Lockup wait counter: 1024/fMAIN 12.8(μs)

Divider x1/2

fIH 80 (MHz)
fMAIN 80 (MHz)
fCLK 40000 (kHz)
fMX 16 (MHz)
fMP 80 (MHz)
□ LIN0 clock - (kHz)
□ LIN1 clock - (kHz)
□ LIN2 clock - (kHz)
fPLL - (MHz)
Timer RD Clock 80000 (kHz)

Figure 3-14 fCLK should come from fIH when 80 MHz or 64 MHz from fIH to Timer RD



Base frequency: 32 (MHz)

☑ High-speed on-chip oscillator
Frequency: 32 (MHz)

☑ X1 oscillator
Operation mode: X1 oscillation
Frequency: 8 (MHz)
Stable time: 2^18/fx 32768(μs)
CAN0 supply mode Stop supply
CAN1 supply mode Stop supply

☑ PLL setting
Frequency: 64 (MHz)
Lockup wait counter: 1024/fMAIN 128(μs)

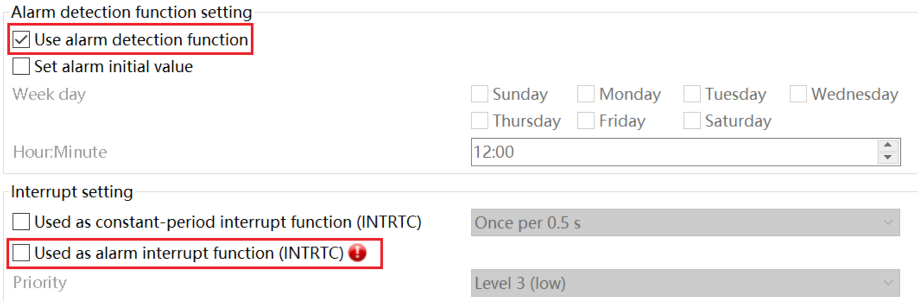
Divider x1/2

fIH 32 (MHz)
fMAIN 8 (MHz)
fCLK 32000 (kHz)
fMX 8 (MHz)
fMP 64 (MHz)
□ LIN0 clock - (kHz)
□ LIN1 clock - (kHz)
□ LIN2 clock - (kHz)
fPLL 64 (MHz)
Timer RD Clock 64000 (kHz)

Figure 3-15 fCLK should come from fPLL when 80 MHz or 64 MHz from fPLL to Timer RD

3.2.4 Improvement for using alarm interrupt INTRTC when using alarm detection function

When Alarm function is used, the interrupt function is mandatory. So, when checking "Use alarm detection function", "Used as alarm interrupt function (INTRTC)" changes to enabled (keep current spec) and there is an error icon after "Used as alarm interrupt function (INTRTC)" if it is unchecked.



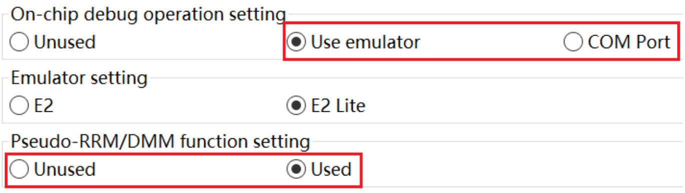
Alarm detection function setting
☑ Use alarm detection function
□ Set alarm initial value
Week day □ Sunday □ Monday □ Tuesday □ Wednesday □ Thursday □ Friday □ Saturday
Hour:Minute 12:00

Interrupt setting
□ Used as constant-period interrupt function (INTRTC) Once per 0.5 s
□ Used as alarm interrupt function (INTRTC) ❗
Priority Level 3 (low)

Figure 3-16 An error icon after "Used as alarm interrupt function (INTRTC)"

3.2.5 Improvement for changing compiler property and debugger property according to "Pseudo-RRM/DMM function setting"

When "Pseudo-RRM/DMM function setting" setting is changed and "On-chip debug operation setting" isn't selected "Unused", compiler property and debugger property will be changed after generating code.



On-chip debug operation setting
○ Unused ● Use emulator ○ COM Port

Emulator setting
○ E2 ● E2 Lite

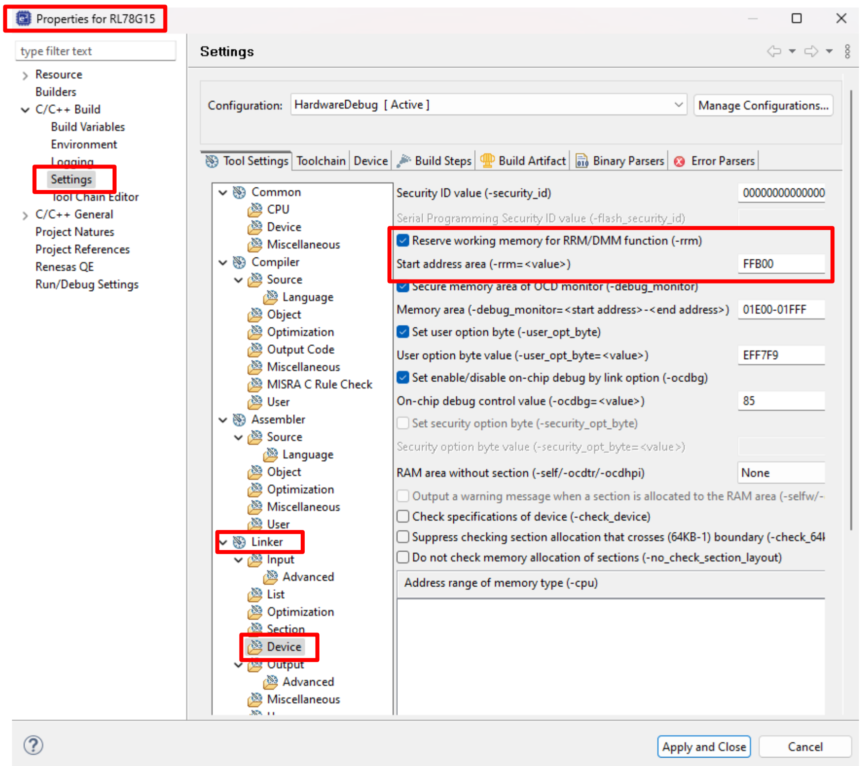
Pseudo-RRM/DMM function setting
○ Unused ● Used

Figure 3-17 Set "Pseudo-RRM/DMM function setting" in [System] page

- Compiler property

Smart Configurator will check "Reserve working memory for RRM/DMM function (-rrm)" and set "Start address (-rrm=<value>)" when setting "Pseudo-RRM/DMM function setting" to "Used".

Note 1: The "Start address (-rrm=<value>)" values only in RL78/G15 (FFB00) and RL78/G16 (FF700).

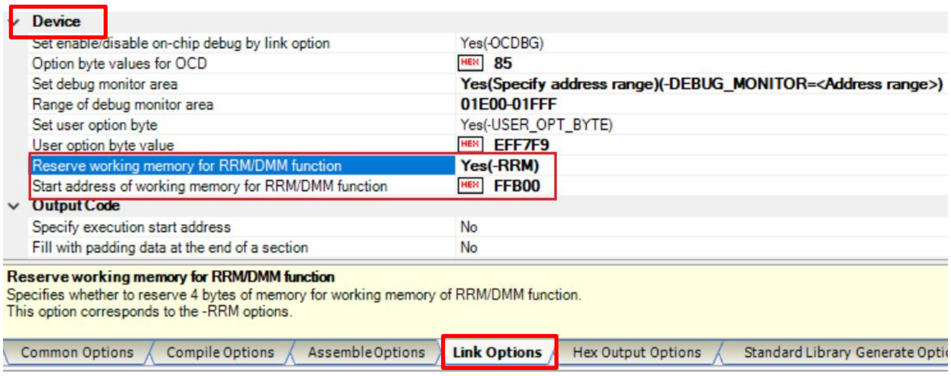


Settings

Configuration: HardwareDebug [Active]   Manage Configurations...

Security ID value (-security_id)	00000000000000
Serial Programming Security ID value (-flash_security_id)	
☑ Reserve working memory for RRM/DMM function (-rrm)	
Start address area (-rrm=<value>)	FFB00
☑ Secure memory area of OCD monitor (-debug_monitor)	
Memory area (-debug_monitor=<start address>-<end address>)	01E00-01FFF
☑ Set user option byte (-user_opt_byte)	
User option byte value (-user_opt_byte=<value>)	EFF7F9
☑ Set enable/disable on-chip debug by link option (-ocdbg)	
On-chip debug control value (-ocdbg=<value>)	85
□ Set security option byte (-security_opt_byte)	
Security option byte value (-security_opt_byte=<value>)	
RAM area without section (-self/-ocdtr/-ocdhpi)	None
□ Output a warning message when a section is allocated to the RAM area (-selfw/-	
□ Check specifications of device (-check_device)	
□ Suppress checking section allocation that crosses (64KB-1) boundary (-check_64k	
□ Do not check memory allocation of sections (-no_check_section_layout)	
Address range of memory type (-cpu)	

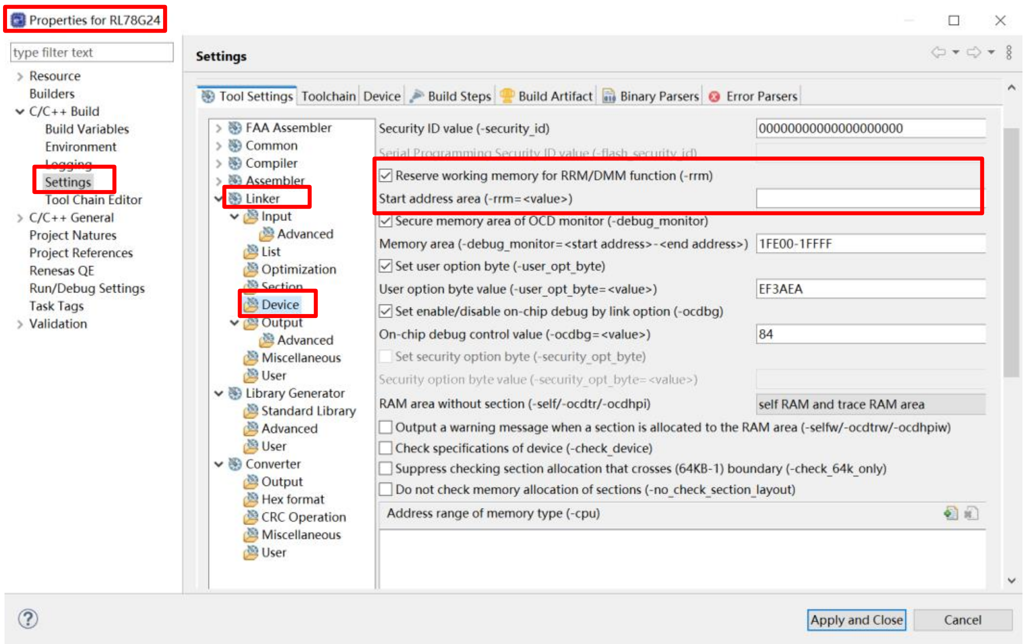
Figure 3-18 Set compiler property "Start address (-rrm=<value>)" in e^2 studio for RL78/G15 and RL78/G16



Device	
Set enable/disable on-chip debug by link option	Yes(-OCDBG)
Option byte values for OCD	85
Set debug monitor area	Yes(Specify address range)(-DEBUG_MONITOR=<Address range>)
Range of debug monitor area	01E00-01FFF
Set user option byte	Yes(-USER_OPT_BYTE)
User option byte value	EFF7F9
Reserve working memory for RRM/DMM function	Yes(-RRM)
Start address of working memory for RRM/DMM function	FFB00
Output Code	
Specify execution start address	No
Fill with padding data at the end of a section	No

Reserve working memory for RRM/DMM function
Specifies whether to reserve 4 bytes of memory for working memory of RRM/DMM function.
This option corresponds to the -RRM options.

Figure 3-19 Set compiler property "Start address (-rrm=<value>)" in CS+ for RL78/G15 and RL78/G16



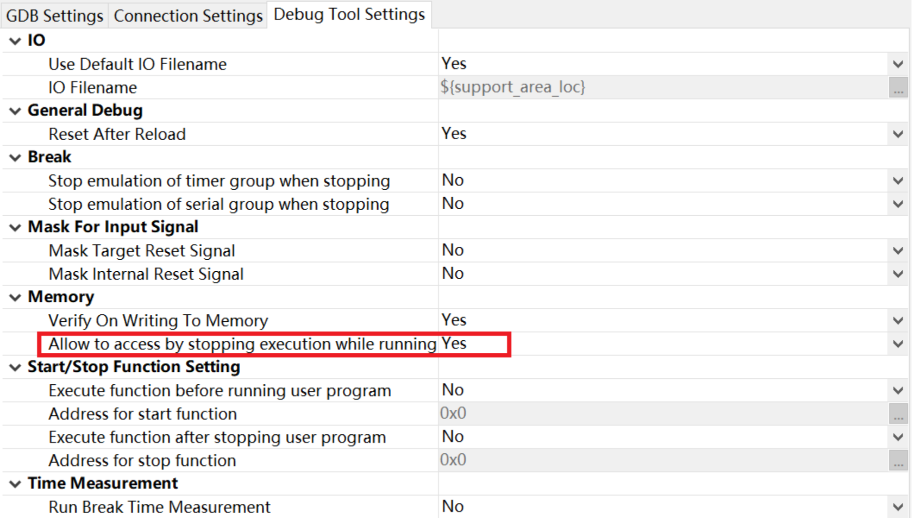
Security ID value (-security_id) 00000000000000000000
Serial Programming Security ID value (-flash_security_id)
☑ Reserve working memory for RRM/DMM function (-rrm)
Start address area (-rrm=<value>)
☑ Secure memory area of OCD monitor (-debug_monitor)
Memory area (-debug_monitor=<start address>-<end address>) 1FE00-1FFFF
☑ Set user option byte (-user_opt_byte)
User option byte value (-user_opt_byte=<value>) EF3AEA
☑ Set enable/disable on-chip debug by link option (-ocdbg)
On-chip debug control value (-ocdbg=<value>) 84
□ Set security option byte (-security_opt_byte)
Security option byte value (-security_opt_byte=<value>)
RAM area without section (-self/-ocdtr/-ocdhpi) self RAM and trace RAM area
□ Output a warning message when a section is allocated to the RAM area (-selfw/-ocdtrw/-ocdhpiw)
□ Check specifications of device (-check_device)
□ Suppress checking section allocation that crosses (64KB-1) boundary (-check_64k_only)
□ Do not check memory allocation of sections (-no_check_section_layout)
Address range of memory type (-cpu)

Figure 3-20 Set compiler property "Start address (-rrm=<value>)" in e^2 studio except RL78/G15 and RL78/G16

- Debugger property

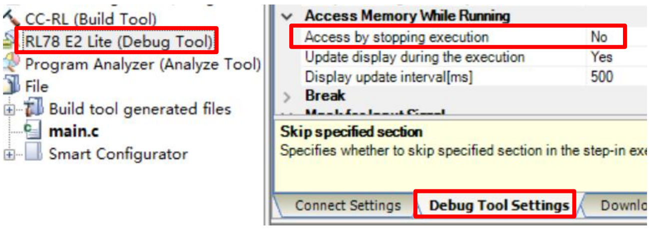
Smart Configurator will set "Allow to access by stopping execution while running" or "Access by stopping execution" to "Yes" when setting "Pseudo-RRM/DMM function setting" to "Used".

Note 1: The supported target emulators include E2 Lite, E2, E20, COM port.
Note 2: The supported compilers include CCRL and LLVM.
Note 3: If the CS+ debugging tool/e^2 studio Debug Configurations and Smart Configurator debugger don't match, Smart Configurator will output a message (Warning 070003) and won't change the related property.
Note 4: Smart Configurator can judge E2 Lite/E2/E20/COM port emulators separately in e^2 studio. But if e^2 studio Debug Configurations and Smart Configurator debugger match, Smart Configurator will modify related property in E2 Lite/E2/E20 emulators at the same time.



IO	
Use Default IO Filename	Yes
IO Filename	${support_area_loc}
General Debug	
Reset After Reload	Yes
Break	
Stop emulation of timer group when stopping	No
Stop emulation of serial group when stopping	No
Mask For Input Signal	
Mask Target Reset Signal	No
Mask Internal Reset Signal	No
Memory	
Verify On Writing To Memory	Yes
Allow to access by stopping execution while running	Yes
Start/Stop Function Setting	
Execute function before running user program	No
Address for start function	0x0
Execute function after stopping user program	No
Address for stop function	0x0
Time Measurement	
Run Break Time Measurement	No

Figure 3-21 Set "Access by stopping execution" in e^2 studio

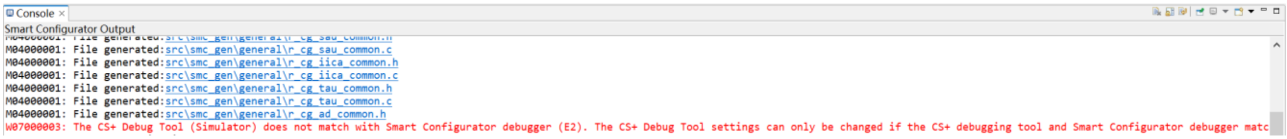


CC-RL (Build Tool)
RL78 E2 Lite (Debug Tool)
Program Analyzer (Analyze Tool)
File
Build tool generated files
main.c
Smart Configurator

Access Memory While Running	
Access by stopping execution	No
Update display during the execution	Yes
Display update interval[ms]	500
Break	

Skip specified section
Specifies whether to skip specified section in the step-in exe

Figure 3-22 Set "Access by stopping execution" in CS+



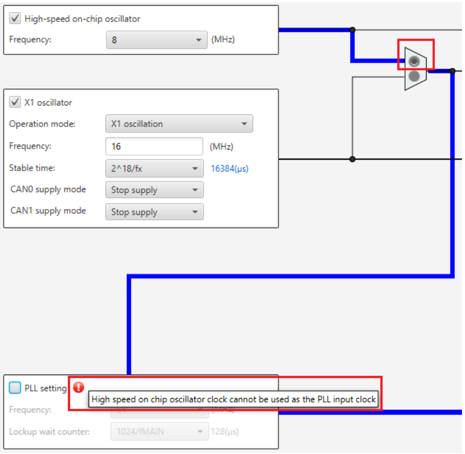
Console
Smart Configurator Output

```
M04000001: File generated:src\smc_gen\general\r_cg_sau_common.h
M04000001: File generated:src\smc_gen\general\r_cg_sau_common.c
M04000001: File generated:src\smc_gen\general\r_cg_iica_common.h
M04000001: File generated:src\smc_gen\general\r_cg_iica_common.c
M04000001: File generated:src\smc_gen\general\r_cg_tau_common.h
M04000001: File generated:src\smc_gen\general\r_cg_tau_common.c
M04000001: File generated:src\smc_gen\general\r_cg_ad_common.h
W07000003: The CS+ Debug Tool (Simulator) does not match with Smart Configurator debugger (E2). The CS+ Debug Tool settings can only be changed if the CS+ debugging tool and Smart Configurator debugger matc
```

Figure 3-23 The message when CS+ debugging tool/e^2 studio Debug Configurations and Smart Configurator debugger don't match

3.2.6 Improvement for high speed on chip oscillator clock cannot be used as the PLL input clock

The User’s Manual (R01UH1061EJ0050) has an error about PLL input clock. Clocks that can be used as the PLL input clock are the X1 clock or an external main system clock. High speed on chip oscillator clock can't be used as the PLL input clock. Smart Configurator for RL78 V1.11.0 adds the judgment about PLL input clok. If setting high speed on chip oscillator clock as the PLL input clock, an error icon will display after "PLL setting". The User’s Manual (R01UH1061EJ0100) will fix this error.



☑ High-speed on-chip oscillator
Frequency: 8 (MHz)

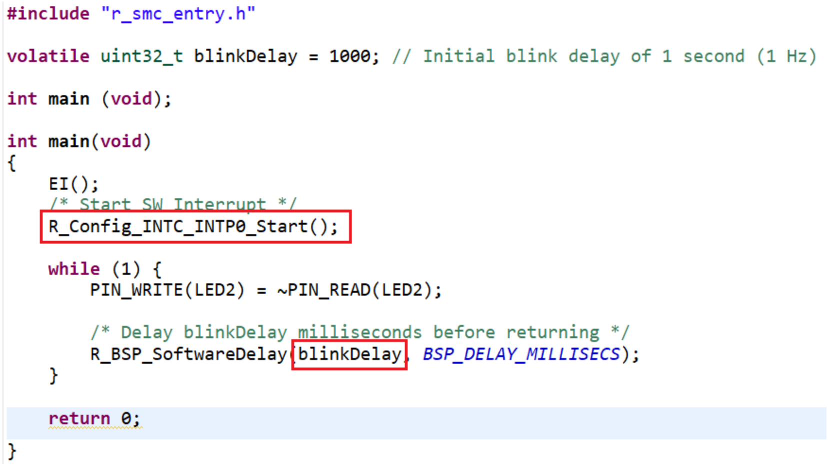
☑ X1 oscillator
Operation mode: X1 oscillation
Frequency: 16 (MHz)
Stable time: 2^18/fx 16384(μs)
CAN0 supply mode Stop supply
CAN1 supply mode Stop supply

□ PLL setting
High speed on chip oscillator clock cannot be used as the PLL input clock
Frequency:
Lockup wait counter: 1024/fMAIN 128(μs)

Figure 3-24 An error icon after "PLL setting" when high speed on chip oscillator clock is input clock

3.2.7 Improve blinky sample project for e^2 studio

From Smart Configurator for RL78 V1.11.0, blinky sample project is improved to detect if the selected board has a switch, the user can use switch to change blinky frequency.



```
#include "r_smc_entry.h"

volatile uint32_t blinkDelay = 1000; // Initial blink delay of 1 second (1 Hz)

int main (void);

int main(void)
{
    EI();
    /* Start SW Interrupt */
    R_Config_INTC_INTP0_Start();

    while (1) {
        PIN_WRITE(LED2) = ~PIN_READ(LED2);

        /* Delay blinkDelay milliseconds before returning */
        R_BSP_SoftwareDelay(blinkDelay, BSP_DELAY_MILLISECS);
    }

    return 0;
}
```

Figure 3-25 Improved code of blinky sample project

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 V1.11.0.

5.1 List of Limitation

Table 5-1 List of Limitation ✓: 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 Flexible Application Accelerator (FAA) component does not support LLVM project	–	–	–	–	–	–	✓	–	
6	Note on not using Flexible Application Accelerator (FAA) component in Mac OS and Linux	–	–	–	–	–	–	✓	–	
7	Note on not using SNOOZE Mode Sequencer (SMS) component in Mac OS and Linux	✓	–	–	–	✓	–	–	–	
8	Note on the build error message such as "Section ".bss_ATFA300" cannot be placed on the "TRACERAM area""	–	✓	–	✓	–	–	–	✓	
9	Note on the setting of subsystem clock pin operation is wrong when fCLK is from fMP and fSL is from fIL	–	✓	–	✓	–	–	–	✓	
10	Note on 16 bit capture mode with channel 0 and 1 can’t be used together with 16 bit count mode with channel 2 and 3	✓	–	–	–	✓		✓	–	

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.

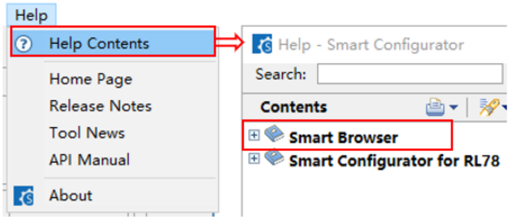


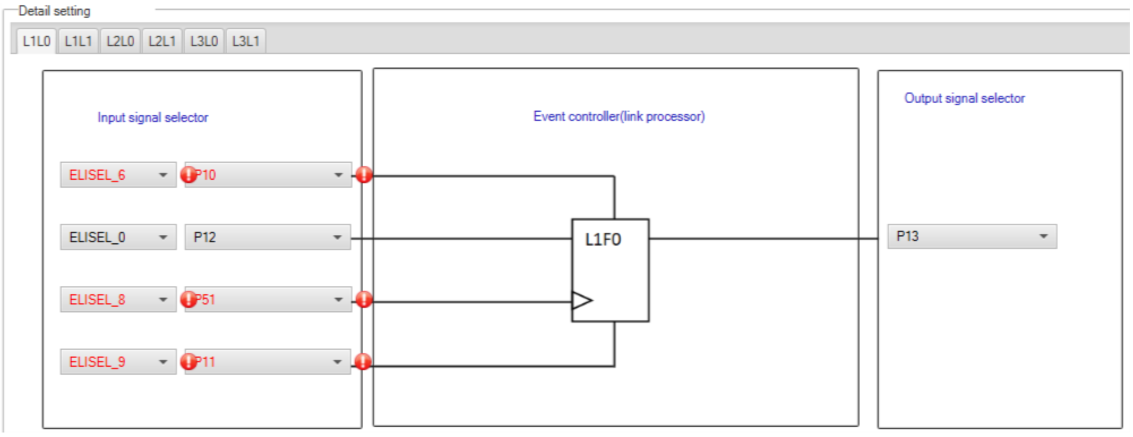
Figure 5-1 Extra help issue

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.

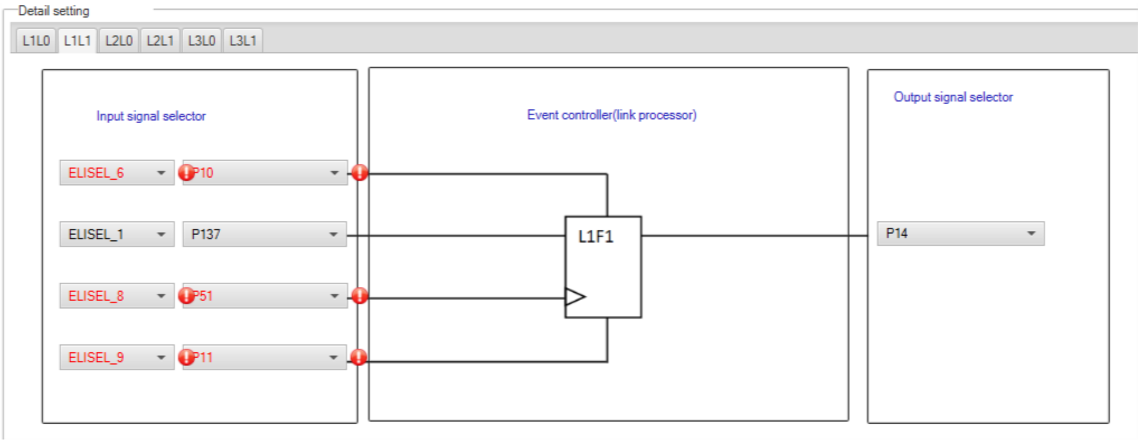
[Workaround]
Make reference to the hardware manual and set the selectable event signal though warning appeared in GUI, the waring is no impact for the code generation.

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



ELISEL_6 P10
ELISEL_0 P12
ELISEL_8 P51
ELISEL_9 P11
P13

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



ELISEL_6 P10
ELISEL_1 P137
ELISEL_8 P51
ELISEL_9 P11
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

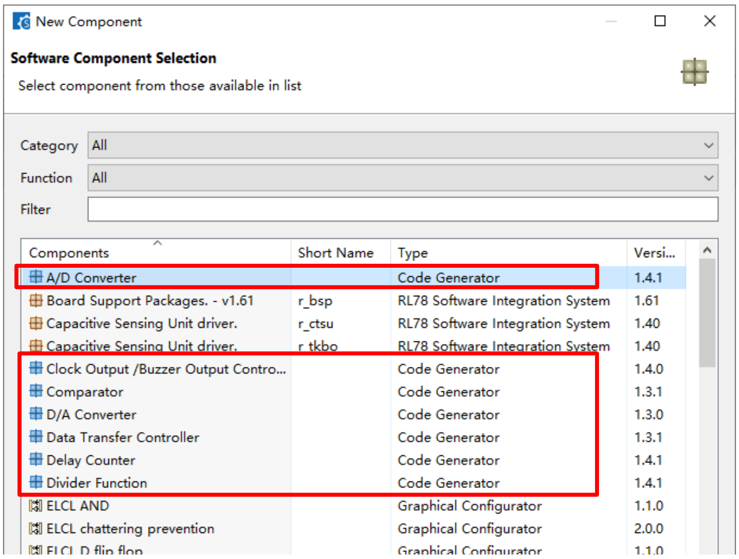
In the following ELCL modules, it is not possible to set “no selection (fixed to 0)” as the input signal of the logic cell block and “negative logic output (inverted)” as the output level of the event signal.

- 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 Flexible Application Accelerator (FAA) component does not support LLVM project

In Smart Configurator for RL78 V1.7.0 or later, Flexible Application Accelerator component was not supported for LLVM project. Though the user can add Flexible Application Accelerator component under LLVM project, 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 Flexible Application Accelerator (FAA) component in Mac OS and Linux

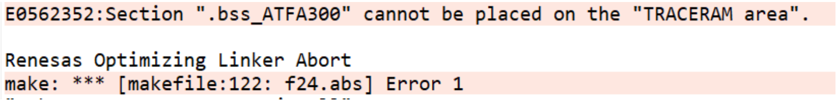
In Smart Configurator for RL78 V1.10.0 or later, Flexible Application Accelerator component was not supported in Mac OS and Linux. Though the user can add Flexible Application Accelerator component in Mac OS and Linux, but the generated Flexible Application Accelerator source code can’t be built successfully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

5.2.7 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.

5.2.8 Note on the build error message such as "Section ".bss_ATFA300" cannot be placed on the "TRACERAM area""

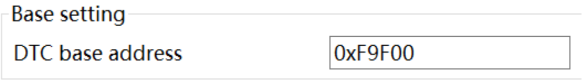
When the user uses DTC component, the generated code build might fail due to some section address overlaps.



```
E0562352:Section ".bss_ATFA300" cannot be placed on the "TRACERAM area".

Renesas Optimizing Linker Abort
make: *** [makefile:122: f24.abs] Error 1
```

Figure 5-5 Build error message

[Workaround]
The user should change the DTC base address to avoid such section overlap error.

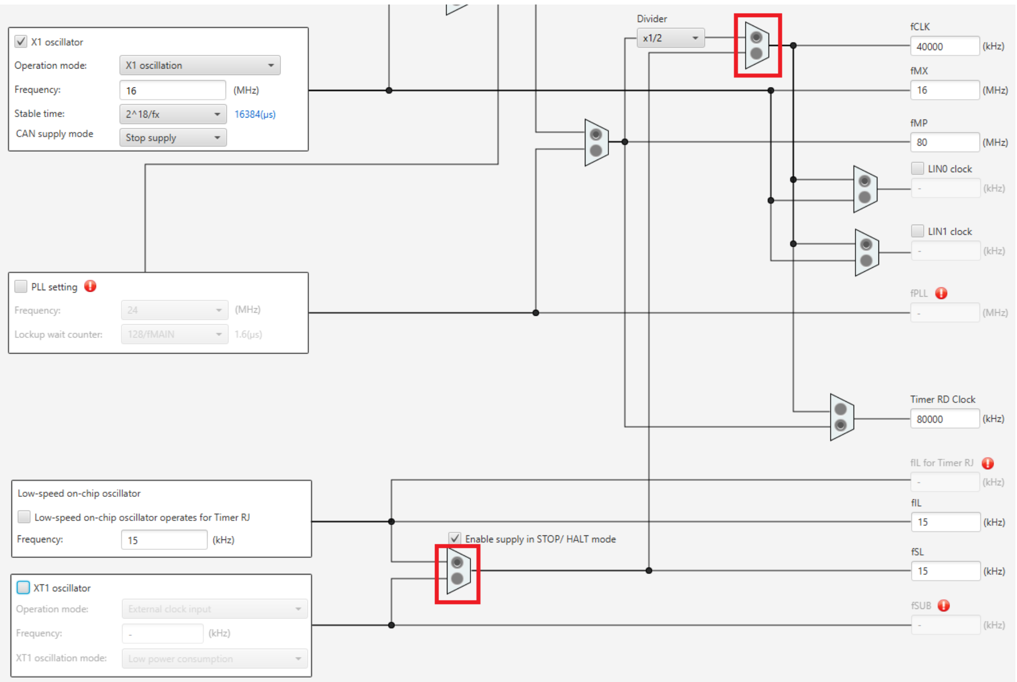


Base setting
DTC base address: 0xF9F00

Figure 5-6 DTC base address setting

5.2.9 Note on the setting of subsystem clock pin operation is wrong when fCLK is from fMP and fSL is from fIL

When fCLK is from fMP and fSL is from fIL, the generated code about CMC.[EXCLKS, OSCSELS] is wrong in r_bsp_config.h.



☑ X1 oscillator
Operation mode: X1 oscillation
Frequency: 16 (MHz)
Stable time: 2^18/fx 16384(μs)
CAN supply mode: Stop supply

□ PLL setting
Frequency: 24 (MHz)
Lockup wait counter: 128/fMAIN 1.6(μs)

Low-speed on-chip oscillator
□ Low-speed on-chip oscillator operates for Timer RJ
Frequency: 15 (kHz)

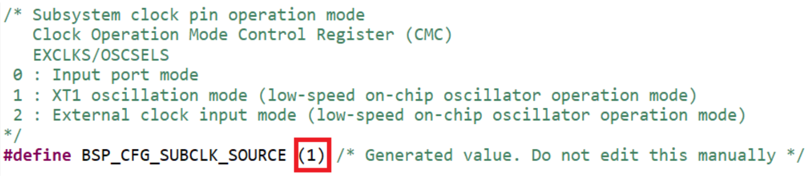
□ XT1 oscillator
Operation mode: External clock input
Frequency: - (kHz)
XT1 oscillation mode: Low power consumption

Divider: x1/2
☑ Enable supply in STOP/ HALT mode

fCLK: 40000 (kHz)
fMX: 16 (MHz)
fMP: 80 (MHz)
□ LIN0 clock: - (kHz)
□ LIN1 clock: - (kHz)
fPLL: - (MHz)
Timer RD Clock: 80000 (kHz)
fIL for Timer RJ: - (kHz)
fIL: 15 (kHz)
fSL: 15 (kHz)
fSUB: - (kHz)

Figure 5-7 The generated code is wrong when fCLK is from fMP and fSL is from fIL

[Workaround]
The user should change the macro defination in r_bsp_config.h manually.

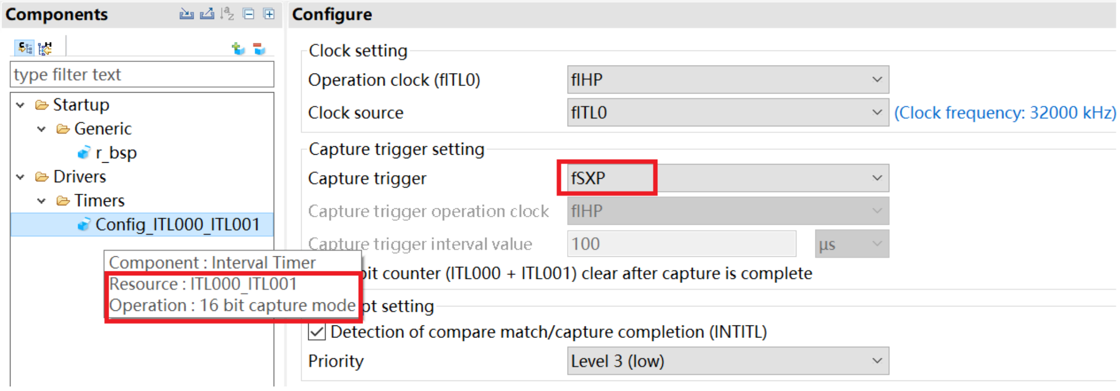


```
/* Subsystem clock pin operation mode
   Clock Operation Mode Control Register (CMC)
   EXCLKS/OSCSELS
 0 : Input port mode
 1 : XT1 oscillation mode (low-speed on-chip oscillator operation mode)
 2 : External clock input mode (low-speed on-chip oscillator operation mode)
*/
#define BSP_CFG_SUBCLK_SOURCE (1) /* Generated value. Do not edit this manually */
```

Figure 5-8 Change the value of macro definition to 1 manually

5.2.10 Note on 16 bit capture mode with channel 0 and 1 can't be used together with 16 bit count mode with channel 2 and 3

If the user wants to use 16 bit count mode with channel 2 and 3 when the user selects 16 bit capture mode and doesn't set "ITLCMP01 compare match interrupt" as "Capture trigger", Smart Configurator can't support it now.



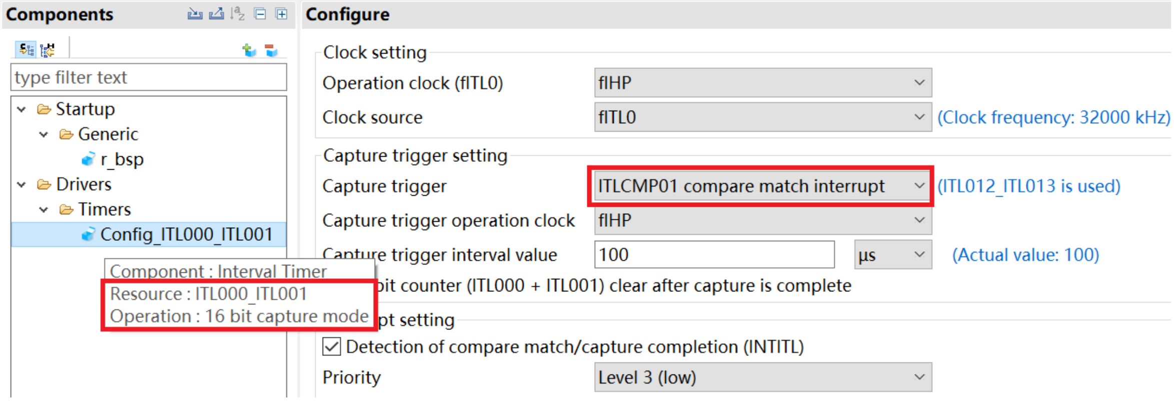
Component : Interval Timer
Resource : ITL000_ITL001
Operation : 16 bit capture mode

Clock setting
Operation clock (fITL0): fIHP
Clock source: fITL0 (Clock frequency: 32000 kHz)
Capture trigger setting
Capture trigger: fSXP
Capture trigger operation clock: fIHP
Capture trigger interval value: 100 µs
bit counter (ITL000 + ITL001) clear after capture is complete
pt setting
☑ Detection of compare match/capture completion (INTITL)
Priority: Level 3 (low)

Figure 5-9 Set Capture trigger as fSXP in 16 bit capture mode

[Workaround]
Change the code in the following steps manually. If code generation is executed again after changing the code, the code will be overwritten and deleted, so be caution.

Step1: Create a component - Interval Timer (ITL000, ITL001) as 16bit Capture Mode. Set "ITLCMP01 compare match interrupt" as "Capture trigger" and generate code.

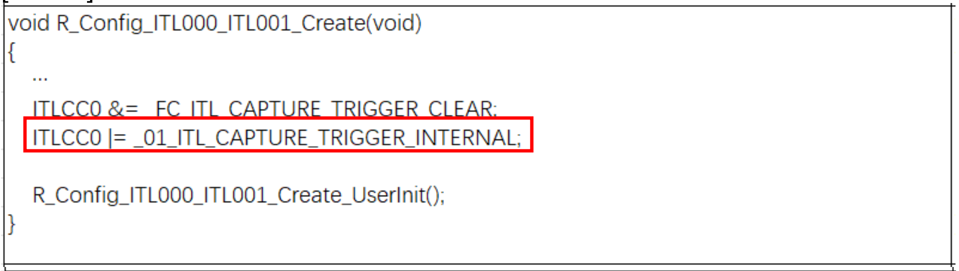


Component : Interval Timer
Resource : ITL000_ITL001
Operation : 16 bit capture mode

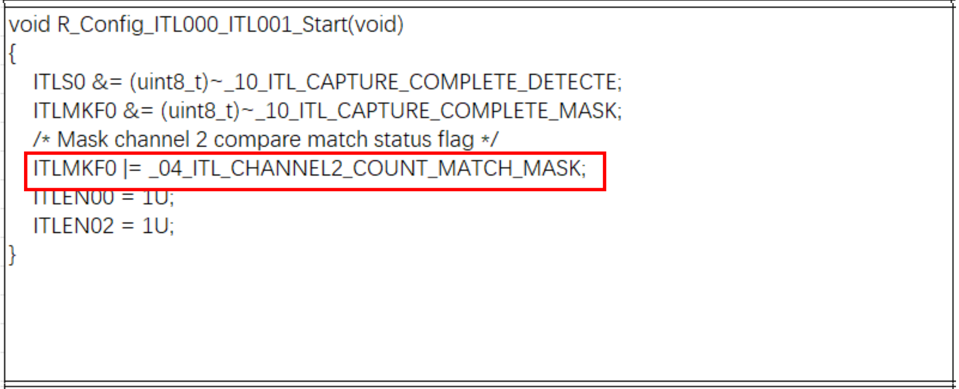
Clock setting
Operation clock (fITL0): fIHP
Clock source: fITL0 (Clock frequency: 32000 kHz)
Capture trigger setting
Capture trigger: ITLCMP01 compare match interrupt (ITL012_ITL013 is used)
Capture trigger operation clock: fIHP
Capture trigger interval value: 100 µs (Actual value: 100)
bit counter (ITL000 + ITL001) clear after capture is complete
pt setting
☑ Detection of compare match/capture completion (INTITL)
Priority: Level 3 (low)

Figure 5-10 Set "ITLCMP01 compare match interrupt" as "Capture trigger"

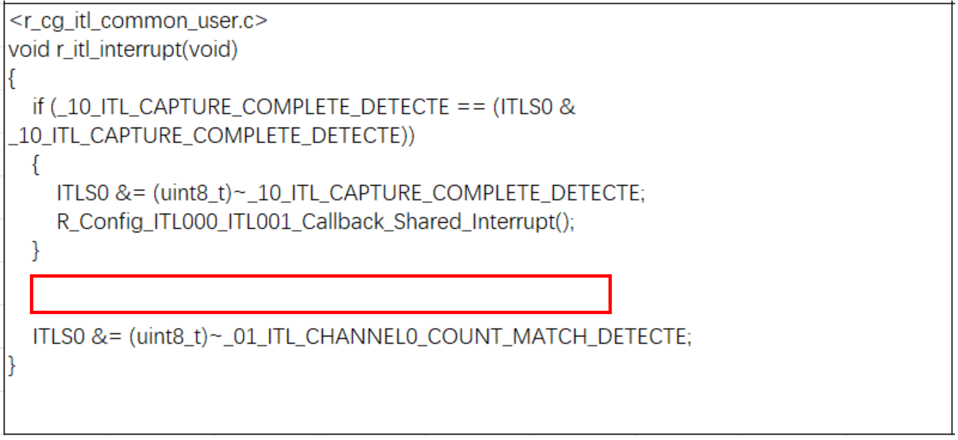
Step2: Modify driver code manually.
[Before]



```
void R_Config_ITL000_ITL001_Create(void)
{
    ...
    ITLCC0 &= _FC_ITL_CAPTURE_TRIGGER_CLEAR;
    ITLCC0 |= _01_ITL_CAPTURE_TRIGGER_INTERNAL;

    R_Config_ITL000_ITL001_Create_UserInit();
}
```

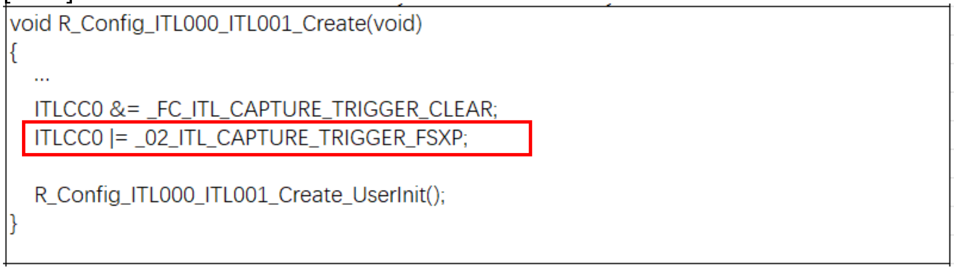


```
void R_Config_ITL000_ITL001_Start(void)
{
    ITLS0 &= (uint8_t)~_10_ITL_CAPTURE_COMPLETE_DETECTE;
    ITLMKF0 &= (uint8_t)~_10_ITL_CAPTURE_COMPLETE_MASK;
    /* Mask channel 2 compare match status flag */
    ITLMKF0 |= _04_ITL_CHANNEL2_COUNT_MATCH_MASK;
    ITLEN00 = 1U;
    ITLEN02 = 1U;
}
```



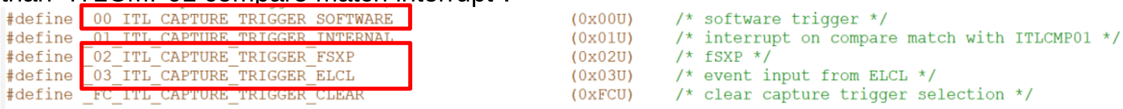
```
<r_cg_itl_common_user.c>
void r_itl_interrupt(void)
{
    if (_10_ITL_CAPTURE_COMPLETE_DETECTE == (ITLS0 &
_10_ITL_CAPTURE_COMPLETE_DETECTE))
    {
        ITLS0 &= (uint8_t)~_10_ITL_CAPTURE_COMPLETE_DETECTE;
        R_Config_ITL000_ITL001_Callback_Shared_Interrupt();
    }

    ITLS0 &= (uint8_t)~_01_ITL_CHANNEL0_COUNT_MATCH_DETECTE;
}
```

[After]

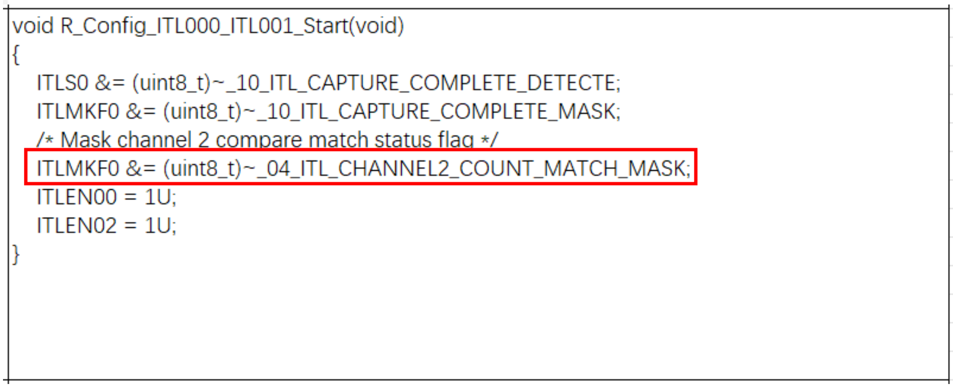


```
void R_Config_ITL000_ITL001_Create(void)
{
    ...
    ITLCC0 &= _FC_ITL_CAPTURE_TRIGGER_CLEAR;
    ITLCC0 |= _02_ITL_CAPTURE_TRIGGER_FSXP;

    R_Config_ITL000_ITL001_Create_UserInit();
}
```

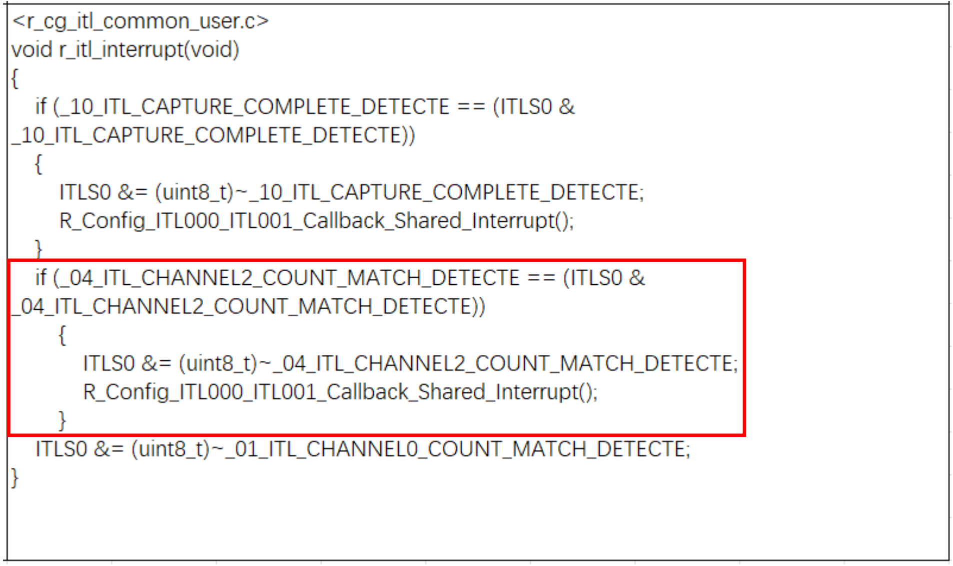
Note: Change the driver code above in red frame to the following settings to set the capture trigger other than "ITLCMP01 compare match interrupt".



```
#define _00_ITL_CAPTURE_TRIGGER_SOFTWARE      (0x00U)   /* software trigger */
#define _01_ITL_CAPTURE_TRIGGER_INTERNAL      (0x01U)   /* interrupt on compare match with ITLCMP01 */
#define _02_ITL_CAPTURE_TRIGGER_FSXP          (0x02U)   /* fSXP */
#define _03_ITL_CAPTURE_TRIGGER_ELCL          (0x03U)   /* event input from ELCL */
#define _FC_ITL_CAPTURE_TRIGGER_CLEAR         (0xFCU)   /* clear capture trigger selection */
```



```
void R_Config_ITL000_ITL001_Start(void)
{
    ITLS0 &= (uint8_t)~_10_ITL_CAPTURE_COMPLETE_DETECTE;
    ITLMKF0 &= (uint8_t)~_10_ITL_CAPTURE_COMPLETE_MASK;
    /* Mask channel 2 compare match status flag */
    ITLMKF0 &= (uint8_t)~_04_ITL_CHANNEL2_COUNT_MATCH_MASK;
    ITLEN00 = 1U;
    ITLEN02 = 1U;
}
```



```
<r_cg_itl_common_user.c>
void r_itl_interrupt(void)
{
    if (_10_ITL_CAPTURE_COMPLETE_DETECTE == (ITLS0 &
_10_ITL_CAPTURE_COMPLETE_DETECTE))
    {
        ITLS0 &= (uint8_t)~_10_ITL_CAPTURE_COMPLETE_DETECTE;
        R_Config_ITL000_ITL001_Callback_Shared_Interrupt();
    }
    if (_04_ITL_CHANNEL2_COUNT_MATCH_DETECTE == (ITLS0 &
_04_ITL_CHANNEL2_COUNT_MATCH_DETECTE))
        {
            ITLS0 &= (uint8_t)~_04_ITL_CHANNEL2_COUNT_MATCH_DETECTE;
            R_Config_ITL000_ITL001_Callback_Shared_Interrupt();
        }
    ITLS0 &= (uint8_t)~_01_ITL_CHANNEL0_COUNT_MATCH_DETECTE;
}
```

6. Points for Caution

This section describes points for caution regarding the Smart Configurator for RL78 V1.11.0.

6.1 List of Caution

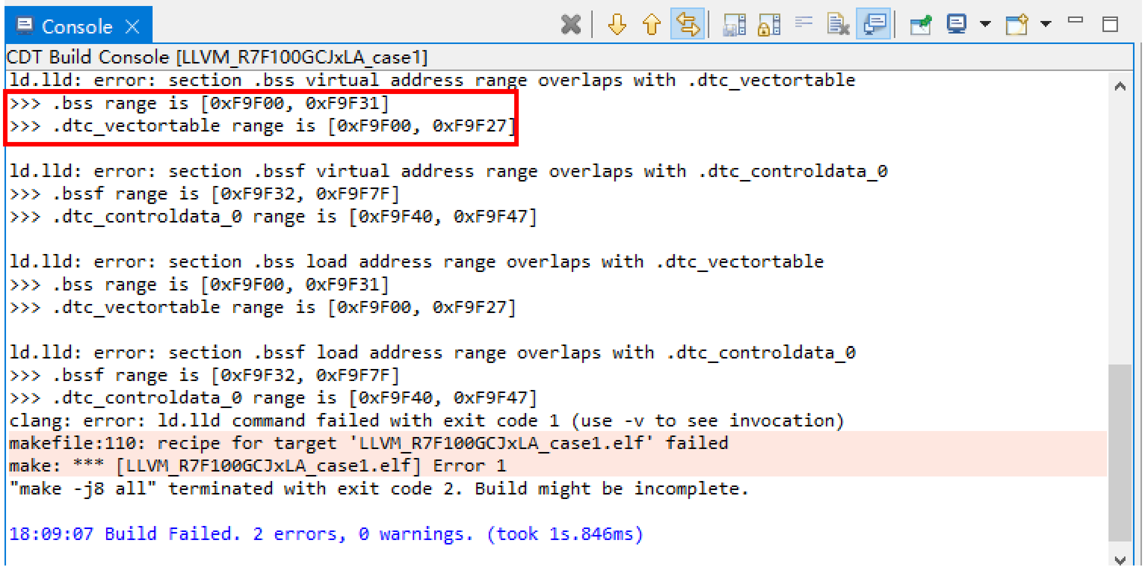
Table 6-1 List of Caution **✓: Applicable, -: Not Applicable**

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 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	−	−	−	−	−	−	✓	−	

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"

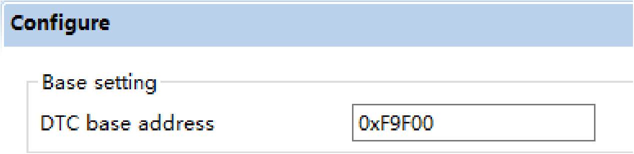
When the user uses many components and DTC component together, the generated code build might fail due to some section address overlaps.



```
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.



DTC base address: 0xF9F00

Figure 6-2 DTC base address setting

6.2.2 Note on the installation of the Smart Configurator

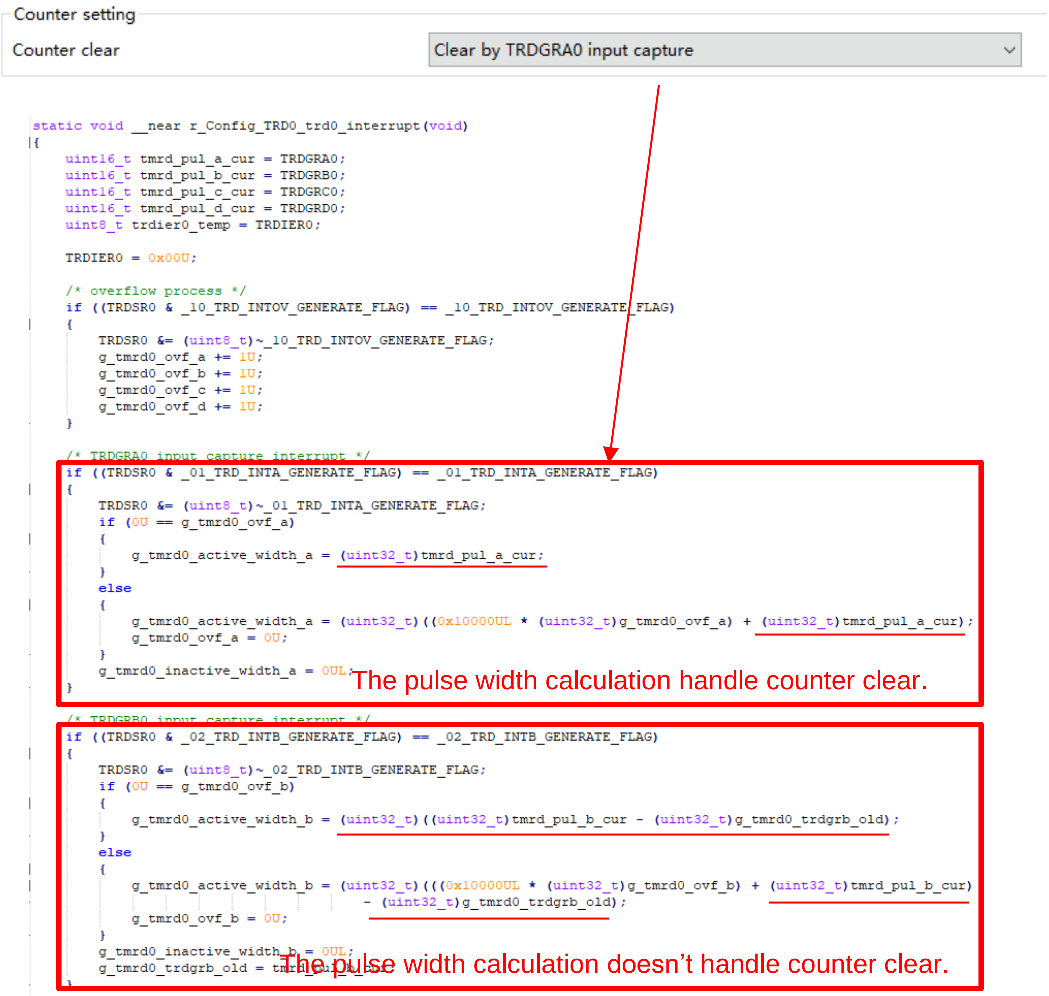
Do not set more than 64 characters for the installation directory.
The user might see an error message "The specified path is too long" and will not be able to install Smart Configurator.

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.

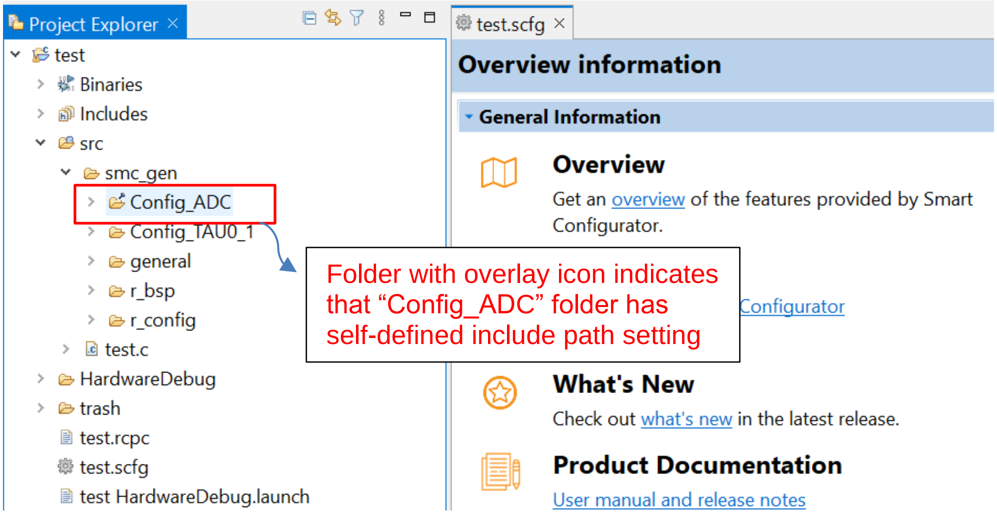
The pulse width calculation doesn't handle counter clear.

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

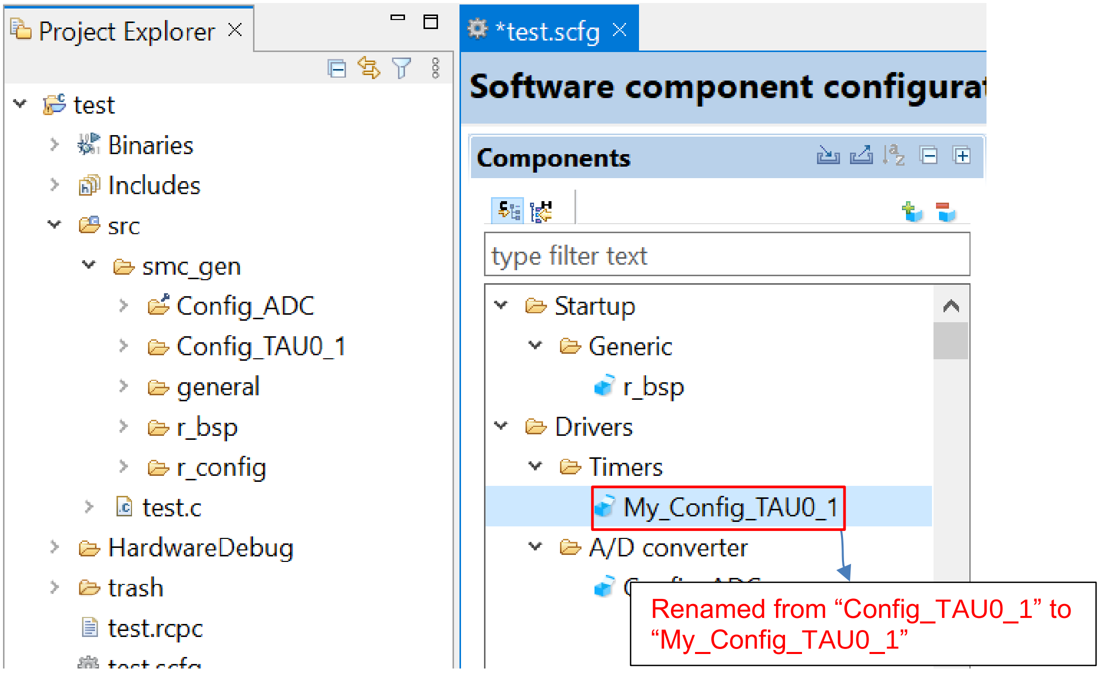
When renaming the added component's configuration in e^2 studio Smart Configurator project that has self-defined include path setting for any folder or file, include path setting for that folder or file will keep the old name setting after code generation. This will cause build error when compiling the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



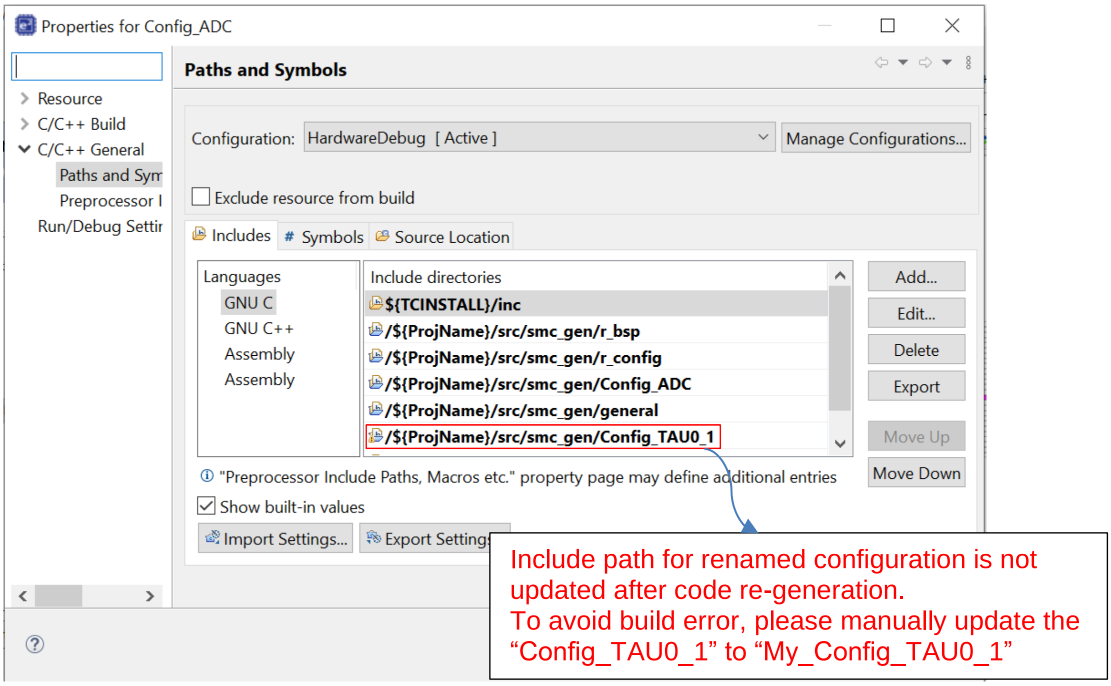
Folder with overlay icon indicates that "Config_ADC" folder has self-defined include path setting

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



Properties for Config_ADC

Paths and Symbols

Configuration: HardwareDebug [Active]

□ Exclude resource from build

Languages	Include directories
GNU C	${TCINSTALL}/inc
GNU C++	/${ProjName}/src/smc_gen/r_bsp
Assembly	/${ProjName}/src/smc_gen/r_config
Assembly	/${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

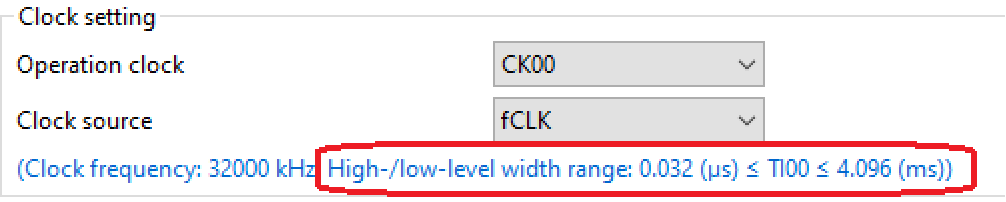
☑ Show built-in values

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

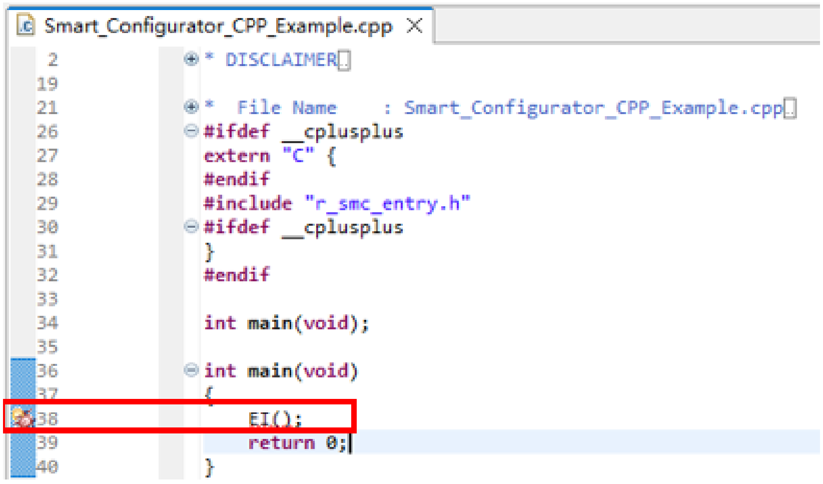
Clock source: fCLK

(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.



Smart_Configurator_CPP_Example.cpp

```
* 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 V1.4.0 or before version, "Release Notes" and "Tools News" in the help menu cannot access the correct URL. This issue has been fixed from this version.
Please access the URL below directly for Smart Configurator for RL78 V1.4.0 or before version.
Release Notes: https://www.renesas.com/rl78-smart-configurator-release-note
Tool News: https://www.renesas.com/rl78-smart-configurator-tn-notes

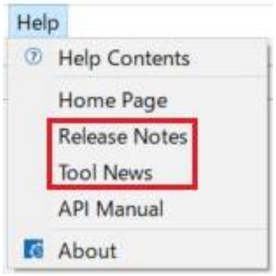


Figure 6-9 Release Notes and Tool News in Smart Configurators

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 */
```

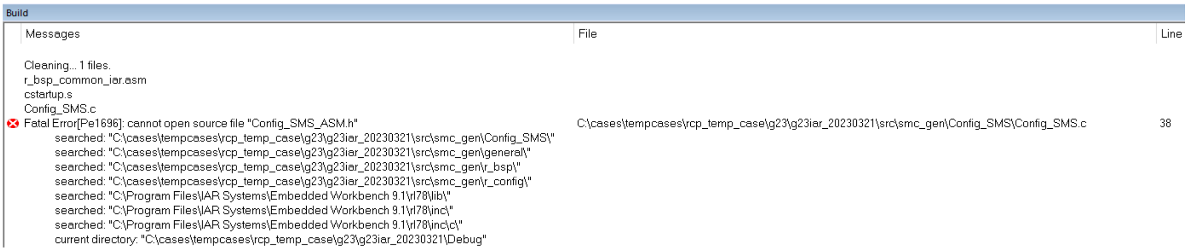
6.2.10 Note on IAR build error when using SMS component

When using 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)

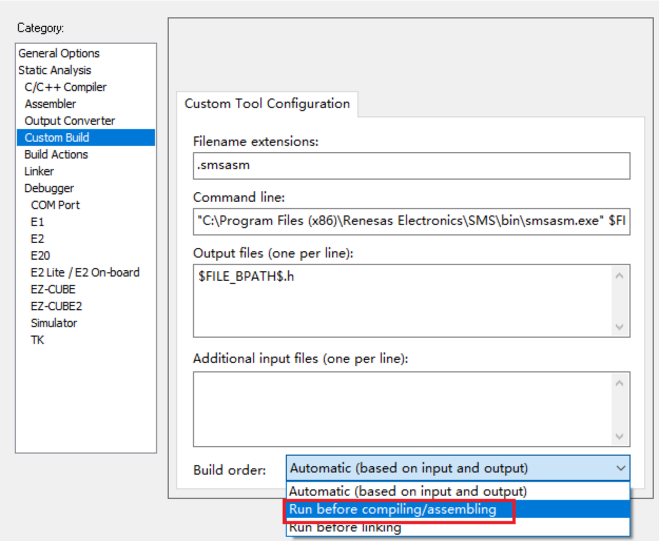
2) When using IAR Embedded workbench V4.21, make “Run this tool before all other tools” checked (refer to Figure 6-12)

The above setting can eliminate this build error.



```
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

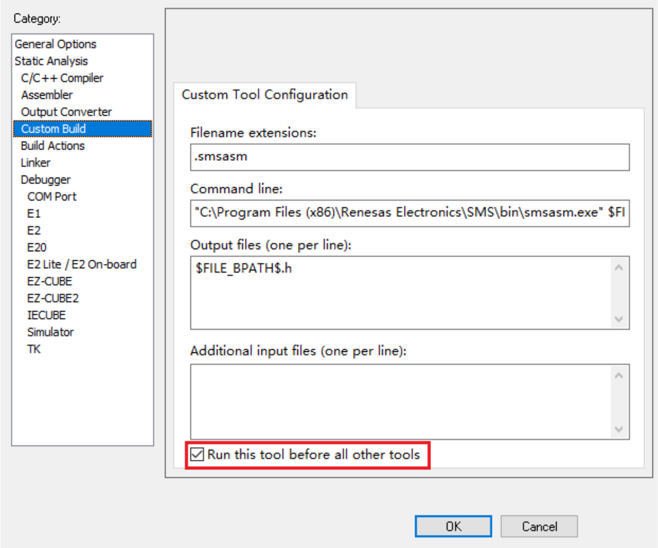
Command line: "C:\Program Files (x86)\Renesas Electronics\SMS\bin\smsasm.exe" $FI

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

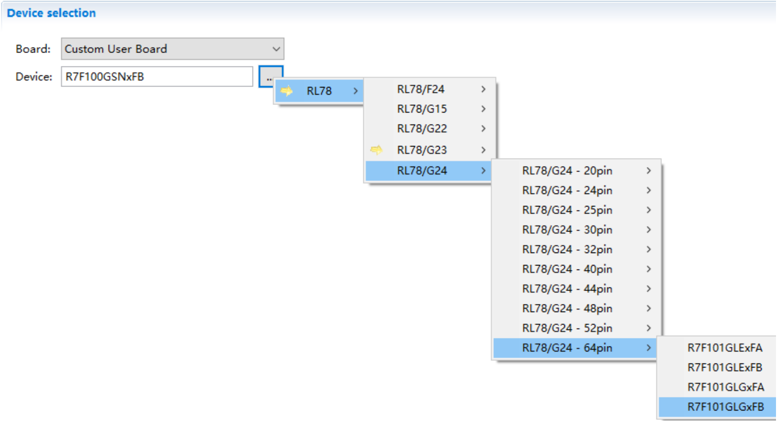
Additional input files (one per line):

☑ 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]

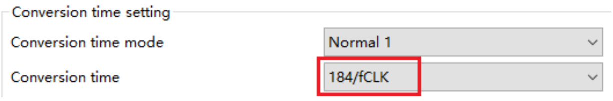
After performing [Change device] (for example, change from RL78/G23 to RL78/G24), the A/D conversion time setting can't be kept. The user should take note to reconfirm the conversion time setting as he wants.



Board: Custom User Board

Device: R7F100GSNxFB

Figure 6-13 [Change device] operation



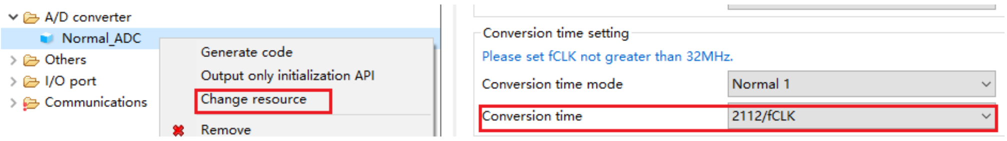
Conversion time setting

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.

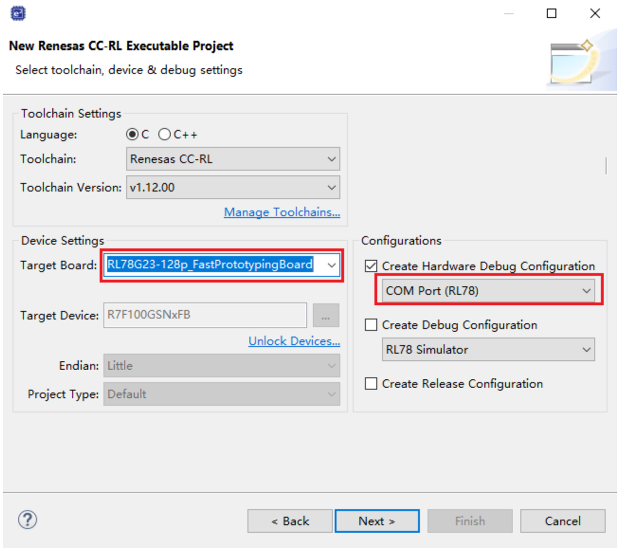
Conversion time mode: Normal 1

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

Device Settings

Target Board: RL78G23-128p_FastPrototypingBoard

Target Device: R7F100GSNxFB

Endian: Little

Project Type: Default

Configurations

☑ Create Hardware Debug Configuration

COM Port (RL78)

☐ Create Debug Configuration

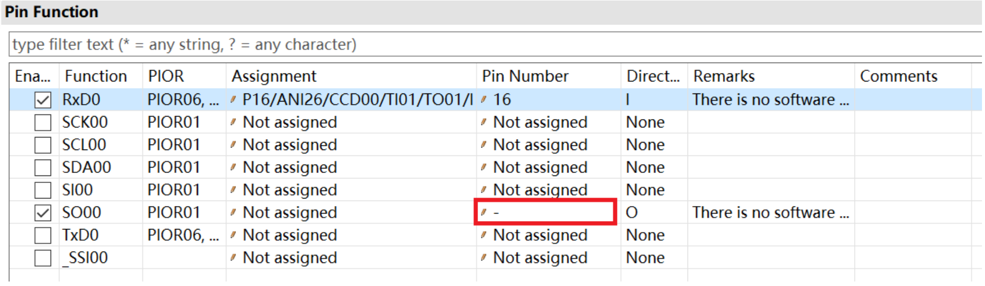
RL78 Simulator

☐ 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

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.

Corporate Headquarters

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

www.renesas.com

Trademarks

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/.