Code Generator for RL78

R20UT5764EC0102
Rev.1.02
Jul.20.26

CS+ Code Generator for RL78 (CS+ for CC/CA,CX) V2.23.00,
e² studio Code Generator Plug-in V2.26.0 (Windows),
e² studio Code Generator Plug-in V1.2.0 (Linux/Mac OS),
AP4 for RL78 V1.23.00, Applilet3 for RL78 V1.26.00
Release Note

Introduction

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

Contents

1. Introduction 4
1.1 Product version 5
1.2 Operating environments 6
1.2.1 PC 6
1.2.2 Development tools 7
2. Supported devices 8
3. Changes 13
3.1 New support 13
3.1.1 Support LLVM toolchain in e² studio 13
3.2 Correction of issues/limitations 14
3.2.1 Improvement on Code Generator view display when screen scaling setting is 100% or above 15
3.2.2 Improvement on some interrupts can’t be generated in LLVM project 15
3.2.3 Improvement on error message in TAU input pulse interval measurement 16
4. History of Corrections Announced in Renesas Tool News 17
5. Restrictions 23
5.1 List of Restrictions 23
5.2 Details for Restriction 27
5.2.1 Timer array unit input clock source 27
5.2.2 24-pin device TAU0 channel 1 setting restriction 27
5.2.3 Real-time clock API function 27
5.2.4 Unit for 'Gain setting' of ΔΣ A/D CONVERTER 27
5.2.5 Restrictions on CSI continuous transfer mode 28
5.2.6 Incorrect function description in data lash library 30
5.2.7 Operation clock when fSUB and fIL are selected in TAU input pulse interval measurement 30
5.2.8 64-bit environment restrictions 30

5.2.9 Incorrect code in R_ELC_Stop() function 30
5.2.10 The trace address is incorrect 31
5.2.11 R_ELC_Stop() function build error in IAR Embedded Workbench 32
5.2.12 Generate multiple [Code Generator] categories 32
5.2.13 TRDIOA1, TRDIOB1, TRDIOC1 and TRDIOD1 pin output forced cutoff control codes are not generated in Timer RD 33
5.2.14 TI05 occupies the pin always when using TAU input pulse interval measurement 34
5.2.15 SCK10 pin setting is wrong when using CSI10 with external clock 35
5.2.16 R_COMP_Set_PowerOff() function should remove from R_Systeminit() 36
5.2.17 Pin settings are wrong when using TAU, INTP, TMRD or CSI 37
5.2.18 Incorrect content of R_RTC_Get_CounterValue() and R_RTC_Set_CounterValue() function 37
5.2.19 The information of interval time controller is wrong in WDT 44
5.2.20 Overflow time can't load correctly in WDT 44

6. Points for Caution 45
6.1 List for Cautions 46
6.2 Details for Caution 50
6.2.1 Online Help (Applilet3, AP4) 50
6.2.2 Coding rule of MISRA-C 50
6.2.3 High-speed on-chip oscillator frequency select register 50
6.2.4 Internal low-speed or internal high-speed oscillator trimming 50
6.2.5 Serial array unit 50
6.2.6 Flash memory CRC operation function (high-speed CRC) 50
6.2.7 Port mode select register (PMS) 50
6.2.8 LIN-bus function of UART 50
6.2.9 Extension code, multi-master and wakeup function of serial interface IICA and multi-master function of simple IIC 50
6.2.10 CAN controllers 50
6.2.11 Safety Functions 51
6.2.12 USB 51
6.2.13 RI78V4 project 51
6.2.14 DTC function (CS+ for CA,CX) 51
6.2.15 High Speed DTC chain transfer 52
6.2.16 Fast Mode Plus setting in IICA slave 52
6.2.17 High-speed on-chip oscillator (CS+ for CA,CX) 52
6.2.18 Pin Configurator (CS+ for CA,CX) 52
6.2.19 Version notation of RL78/G13A generation file. 52
6.2.20 Simple IIC stop condition generation 53
6.2.21 Device change function 53
6.2.22 Code Generator doesn't support C++ project 53
6.2.23 Panel display of code generator node when using multiple projects in e^2 studio 53
6.2.24 Use Trace function and Data Flash library together 54

6.2.25 Clock setting issue when using the Data Flash Library (FDL) 55
6.2.26 Switch to [Code Generator Console] after building the project 56
6.2.27 SAU and TAU GUI can’t fully display when creating projects in e^2 studio 57
6.2.28 Code Generator can’t register correctly sometimes in e^2 studio 58
6.2.29 WDT reset issue 59

Revision History 60

1. Introduction

The Code Generator for RL78 is a software tool to generate control programs (device driver programs) for peripheral modules (timers, UART, A/D, etc.). It generates device driver codes using user settings through GUI. Initialize code and API (Application Programming Interface) functions are provided. The following products are provided as code generator for RL78.

- Windows
  - Code Generator Plug-in for RL78 (IDE CS+ for CC, CS+ for CA,CX, e^2 studio)
  - Code Generator Plug-in for Common (IDE CS+ for CC, CS+ for CA,CX, e^2 studio)
  - AP4 for RL78
  - Applilet3 for RL78

- Linux/Mac OS
  - Code Generator Plug-in for RL78 (e^2 studio)

1.1 Product version

- Windows
  - CS+ Code Generator for RL78 (CS+ for CC) 2.23.00
  - CS+ Code Generator for RL78 (CS+ for CA,CX) 2.22.00
  - Code Generator Common (CS+ for CC) 1.22.00
  - Code Generator Common (CS+ for CA,CX) 1.19.00
  - e^2 studio Code Generator Plug-in 2.26.00
  - Applilet3 for RL78 1.26.00 (4.08.08.01)

Group	Version	Group	Version
RL78/D1A	V2.04.06.01	RL78/G13, G13A	V2.05.09.01
RL78/F12	V2.04.07.01	RL78/G14	V2.05.09.01
RL78/F13	V2.03.08.01	RL78/G1A	V2.04.05.01
RL78/F14	V2.03.08.01	RL78/I1A	V2.04.06.01
RL78/F15	V1.01.09.01	RL78/L12	V2.04.07.01
RL78/G12	V2.04.10.01		

  - AP4 for RL78 1.23.00 (2.10.09.01)

Group	Version	Group	Version
RL78/F1E	V1.01.08.01	RL78/H1D	V1.00.04.01
RL78/G10	V1.05.06.01	RL78/I1B	V1.03.05.01
RL78/G11	V1.02.07.01	RI78/I1C	V1.01.08.01
RL78/G1C	V1.03.05.01	RL78/I1D	V1.01.06.01
RL78/G1D	V1.01.05.01	RL78/I1E	V1.03.06.01
RL78/G1E	V1.04.05.01	RL78/L13	V1.04.07.01
RL78/G1F	V1.01.07.01	RL78/L1A	V1.01.06.01
RL78/G1G	V1.01.04.01	RL78/L1C	V1.03.04.01
RL78/G1H	V1.01.06.01		

- Linux/Mac OS
  - e^2 studio Code Generator Plug-in 1.2.0

1.2 Operating environments

1.2.1 PC

1.2.1.1 Windows

- System: x64/x86 based processor
  Windows® 11
  Windows® 10 (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)
- Required elements of the software environment other than the Windows OS: .NET Framework 4.8 plus a language pack

1.2.1.2 Linux

Code Generator Plug-in in e^2 studio 2024-07 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.

1.2.1.3 Mac OS

Code Generator Plug-in in e^2 studio 2024-07 or later is supported on Mac OS.

- System: Apple ARM-based systems-on-a-chip (SoCs)
  MacOS 26 (Tahoe)
  MacOS 15 (Sequoia)
- Memory capacity: 8 GB of RAM recommended.
- Capacity of hard disk: At least 2 GB of free space.

1.2.2 Development tools

1.2.2.1 CS+

- Integrated development environment CS+ from Renesas, V8.15.00 or later
- Renesas electronics Compiler for RL78 [CC-RL] V1.16 or later
- Renesas electronics Compiler for 78K0R [CA78K0R] V1.72 or later

1.2.2.2 e² studio (Windows), AP4 for RL78 and Applilet3 for RL78

- Integrated development environment e² studio (64-bit) from Renesas, 2025-12 or later
- Renesas electronics Compiler for RL78 [CC-RL] V1.16 or later
- Renesas GCC for RL78 V4.9 or later
- IAR Embedded Workbench for Renesas RL78 V5.10.3 or later
- Renesas LLVM for RL78 17.0.1.202506 or later

1.2.2.3 e² studio (Linux)

- Integrated development environment e² studio (64-bit) from Renesas, 2025-04 or later
- Renesas electronics Compiler for RL78 [CC-RL] V1.13 or later
- Renesas LLVM for RL78 10.0.0.202209 or later

1.2.2.4 e² studio (Mac OS)

- Integrated development environment e² studio (64-bit) from Renesas, 2025-04 or later
- Renesas LLVM for RL78 10.0.0.202310 or later

- Integrated development environment e² studio (64-bit) from Renesas, 2026-04.1 or later
- Renesas electronics Compiler for RL78 [CC-RL] V1.16.01 or later

2. Supported devices

The devices supported by the Code Generator for RL78 are listed below.

Table 2-1. Supported devices

✓: Support -: Not support

Group (HW Manual number)	PIN	Device name	CS+	e² studio	AP4	Applilet3
RL78/F12 Group (R01UH0231EJ0111)	20pin	R5F1096E, R5F1096D, R5F1096C, R5F1096B, R5F1096A, R5F10968	✓	✓	-	✓
	30pin	R5F109AE, R5F109AD, R5F109AC, R5F109AB, R5F109AA	✓	✓	-	✓
	32pin	R5F109BE, R5F109BD, R5F109BC, R5F109BB, R5F109BA	✓	✓	-	✓
	48pin	R5F109GE, R5F109GD, R5F109GC, R5F109GB, R5F109GA	✓	✓	-	✓
	64pin	R5F109LE, R5F109LD, R5F109LC, R5F109LB, R5F109LA	✓	✓	-	✓
RL78/F13 Group (R01UH0368EJ0210)	20pin	R5F10A6A, R5F10A6C, R5F10A6D, R5F10A6E	✓	✓	-	✓
	30pin	R5F10AAA, R5F10AAC, R5F10AAD, R5F10AAE, R5F10BAC, R5F10BAD, R5F10BAE, R5F10BAF, R5F10BAG	✓	✓	-	✓
	32pin	R5F10ABA, R5F10ABC, R5F10ABD, R5F10ABE, R5F10BBC, R5F10BBD, R5F10BBE, R5F10BBF, R5F10BBG	✓	✓	-	✓
	48pin	R5F10AGA, R5F10AGC, R5F10AGD, R5F10AGE, R5F10AGF, R5F10AGG, R5F10BGC, R5F10BGD, R5F10BGE, R5F10BGF, R5F10BGG	✓	✓	-	✓
	64pin	R5F10ALC, R5F10ALD, R5F10ALE, R5F10ALF, R5F10ALG, R5F10BLC, R5F10BLD, R5F10BLE, R5F10BLF, R5F10BLG	✓	✓	-	✓
	80pin	R5F10AME, R5F10AMF, R5F10AMG, R5F10BME, R5F10BMF, R5F10BMG	✓	✓	-	✓
RL78/F14 Group (R01UH0368EJ0210)	30pin	R5F10PAD, R5F10PAE	✓	✓	-	✓
	32pin	R5F10PBD, R5F10PBE	✓	✓	-	✓
	48pin	R5F10PGD, R5F10PGE, R5F10PGF, R5F10PGG, R5F10PGH, R5F10PGJ	✓	✓	-	✓
	64pin	R5F10PLE, R5F10PLF, R5F10PLG, R5F10PLH, R5F10PLJ	✓	✓	-	✓
	80pin	R5F10PME, R5F10PMF, R5F10PMG, R5F10PMH, R5F10PMJ	✓	✓	-	✓
	100pin	R5F10PPE, R5F10PPF, R5F10PPG, R5F10PPH, R5F10PPJ	✓	✓	-	✓
RL78/F15 Group (R01UH0559EJ0100)	48pin	R5F113GL, R5F113GK	✓	✓	-	✓
	64pin	R5F113LL, R5F113LK	✓	✓	-	✓
	80pin	R5F113ML, R5F113MK	✓	✓	-	✓
	100pin	R5F113PL, R5F113PK, R5F113PJ, R5F113PH, R5F113PG	✓	✓	-	✓
	144pin	R5F113TL, R5F113TK, R5F113TJ, R5F113TH, R5F113TG	✓	✓	-	✓
RL78/F1E Group (R01UH0611EJ0052)	64pin	R5F11KLE, R5F11LLE, R5F11KLF, R5F11LLF, R5F11KLG, R5F11LLG	✓	✓	✓	-

Table 2-2. Supported devices

✓: Support -: Not support

Group (HW Manual number)	PIN	Device name	CS+	e² studio	AP4	Applilet3
RL78/G10 Group (R01UH0384EJ0311)	10pin	R5F10Y14, R5F10Y16, R5F10Y17	✓	✓	✓	-
	16pin	R5F10Y44, R5F10Y46, R5F10Y47	✓	✓	✓	-
RL78/G11 Group (R01UH0637EJ0110)	10pin	R5F1051A	✓	✓	✓	-
	16pin	R5F1054A	✓	✓	✓	-
	20pin	R5F1056A	✓	✓	✓	-
	24pin	R5F1057A	✓	✓	✓	-
	25pin	R5F1058A	✓	✓	✓	-
RL78/G12 Group (Windows: R01UH0200EJ0210; Linux/Mac OS: R01UH0200EJ0231)	20pin	R5F10266, R5F10267, R5F10268, R5F10269, R5F1026A, R5F10366, R5F10367, R5F10368, R5F10369, R5F1036A	✓	✓	-	✓
	24pin	R5F10277, R5F10278, R5F10279, R5F1027A, R5F10377, R5F10378, R5F10379, R5F1037A	✓	✓	-	✓
	30pin	R5F102A7, R5F102A8, R5F102A9, R5F102AA, R5F103A7, R5F103A8, R5F103A9, R5F103AA	✓	✓	-	✓
RL78/G13 Group (Windows: R01UH0146EJ0330; Linux/Mac OS: R01UH0146EJ0351)	20pin	R5F1006A, R5F1006C, R5F1006D, R5F1006E, R5F1016A, R5F1016C, R5F1016D, R5F1016E	✓	✓	-	✓
	24pin	R5F1007A, R5F1007C, R5F1007D, R5F1007E, R5F1017A, R5F1017C, R5F1017D, R5F1017E	✓	✓	-	✓
	25pin	R5F1008A, R5F1008C, R5F1008D, R5F1008E, R5F1018A, R5F1018C, R5F1018D, R5F1018E	✓	✓	-	✓
	30pin	R5F100AA, R5F100AC, R5F100AD, R5F100AE, R5F100AF, R5F100AG, R5F101AA, R5F101AC, R5F101AD, R5F101AE, R5F101AF, R5F101AG	✓	✓	-	✓
	32pin	R5F100BA, R5F100BC, R5F100BD, R5F100BE, R5F100BF, R5F100BG, R5F101BA, R5F101BC, R5F101BD, R5F101BE, R5F101BF, R5F101BG	✓	✓	-	✓
	36pin	R5F100CA, R5F100CC, R5F100CD, R5F100CE, R5F100CF, R5F100CG, R5F101CA, R5F101CC, R5F101CD, R5F101CE, R5F101CF, R5F101CG	✓	✓	-	✓
	40pin	R5F100EA, R5F100EC, R5F100ED, R5F100EE, R5F100EF, R5F100EG, R5F100EH, R5F101EA, R5F101EC, R5F101ED, R5F101EE, R5F101EF, R5F101EG, R5F101EH	✓	✓	-	✓
	44pin	R5F100FA, R5F100FC, R5F100FD, R5F100FE, R5F100FF, R5F100FG, R5F100FH, R5F100FJ, R5F100FK, R5F100FL, R5F101FA, R5F101FC, R5F101FD, R5F101FE, R5F101FF, R5F101FG, R5F101FH, R5F101FJ, R5F101FK, R5F101FL	✓	✓	-	✓
	48pin	R5F100GA, R5F100GC, R5F100GD, R5F100GE, R5F100GF, R5F100GG, R5F100GH, R5F100GJ, R5F100GK, R5F100GL, R5F101GA, R5F101GC, R5F101GD, R5F101GE, R5F101GF, R5F101GG, R5F101GH, R5F101GJ, R5F101GK, R5F101GL	✓	✓	-	✓
	52pin	R5F100JC, R5F100JD, R5F100JE, R5F100JF, R5F100JG, R5F100JH, R5F100JJ, R5F100JK, R5F100JL, R5F101JC, R5F101JD, R5F101JE, R5F101JF, R5F101JG, R5F101JH, R5F101JJ, R5F101JK, R5F101JL	✓	✓	-	✓

Table 2-3. Supported devices

✓: Support -: Not support

Group (HW Manual number)	PIN	Device name	CS+	e² studio	AP4	Applilet3
RL78/G13 Group (Windows: R01UH0146EJ0330; Linux/Mac OS: R01UH0146EJ0351)	64pin	R5F100LC, R5F100LD, R5F100LE, R5F100LF, R5F100LG, R5F100LH, R5F100LJ, R5F100LK, R5F100LL, R5F101LC, R5F101LD, R5F101LE, R5F101LF, R5F101LG, R5F101LH, R5F101LJ, R5F101LK, R5F101LL	✓	✓	-	✓
	80pin	R5F100MF, R5F100MG, R5F100MH, R5F100MJ, R5F100MK, R5F100ML, R5F101MF, R5F101MG, R5F101MH, R5F101MJ, R5F101MK, R5F101ML	✓	✓	-	✓
	100pin	R5F100PF, R5F100PG, R5F100PH, R5F100PJ, R5F100PK, R5F100PL, R5F101PF, R5F101PG, R5F101PH, R5F101PJ, R5F101PK, R5F101PL	✓	✓	-	✓
	128pin	R5F100SH, R5F100SJ, R5F100SK, R5F100SL, R5F101SH, R5F101SJ, R5F101SK, R5F101SL	✓	✓	-	✓
RL78/G13A Group (R01UH0856EJ0050)	44pin	R5F140FK, R5F140FL	✓	✓	-	✓
	48pin	R5F140GK, R5F140GL	✓	✓	-	✓
	64pin	R5F140LK, R5F140LL	✓	✓	-	✓
	100pin	R5F140PK, R5F140PL	✓	✓	-	✓
RL78/G14 Group (Windows: R01UH0186EJ0330; Linux/Mac OS: R01UH0186EJ0350)	30pin	R5F104AA, R5F104AC, R5F104AD, R5F104AE, R5F104AF, R5F104AG	✓	✓	-	✓
	32pin	R5F104BA, R5F104BC, R5F104BD, R5F104BE, R5F104BF, R5F104BG	✓	✓	-	✓
	36pin	R5F104CA, R5F104CC, R5F104CD, R5F104CE, R5F104CF, R5F104CG	✓	✓	-	✓
	40pin	R5F104EA, R5F104EC, R5F104ED, R5F104EE, R5F104EF, R5F104EG, R5F104EH	✓	✓	-	✓
	44pin	R5F104FA, R5F104FC, R5F104FD, R5F104FE, R5F104FF, R5F104FG, R5F104FH, R5F104FJ	✓	✓	-	✓
	48pin	R5F104GA, R5F104GC, R5F104GD, R5F104GE, R5F104GF, R5F104GG, R5F104GH, R5F104GJ, R5F104GK, R5F104GL	✓	✓	-	✓
	52pin	R5F104JC, R5F104JD, R5F104JE, R5F104JF, R5F104JG, R5F104JH, R5F104JJ	✓	✓	-	✓
	64pin	R5F104LC, R5F104LD, R5F104LE, R5F104LF, R5F104LG, R5F104LH, R5F104LJ, R5F104LK, R5F104LL	✓	✓	-	✓
	80pin	R5F104MF, R5F104MG, R5F104MH, R5F104MJ, R5F104MK, R5F104ML	✓	✓	-	✓
	100pin	R5F104PF, R5F104PG, R5F104PH, R5F104PJ, R5F104PK, R5F104PL	✓	✓	-	✓
RL78/G1A Group (R01UH0305EJ0200)	20pin	R5F10E8A, R5F10E8C, R5F10E8D, R5F10E8E	✓	✓	-	✓
	24pin	R5F10EBA, R5F10EBC, R5F10EBD, R5F10EBE	✓	✓	-	✓
	30pin	R5F10EGA, R5F10EGC, R5F10EGD, R5F10EGE	✓	✓	-	✓
	64pin	R5F10ELC, R5F10ELD, R5F10ELE	✓	✓	-	✓
RL78/G1C Group (R01UH0348EJ0100)	32pin	R5F10JBC, R5F10KBC	✓	✓	✓	-
	48pin	R5F10JGC, R5F10KGC	✓	✓	✓	-

Table 2-4. Supported devices

✓: Support -: Not support

Group (HW Manual number)	PIN	Device name	CS+	e² studio	AP4	Applilet3
RL78/G1D Group (R01UH0515EJ0100)	48pin	R5F11AGG, R5F11AGH, R5F11AGJ	✓	✓	✓	-
RL78/G1E Group (R01UH0353EJ0101)	64pin	R5F10FLC, R5F10FLD, R5F10FLE	✓	✓	✓	-
	80pin	R5F10FMC, R5F10FMD, R5F10FME	✓	✓	✓	-
RL78/G1F Group (R01UH0516EJ0100)	24pin	R5F11B7C, R5F11B7E	✓	✓	✓	-
	32pin	R5F11BBC, R5F11BBE	✓	✓	✓	-
	36pin	R5F11BCC, R5F11BCE	✓	✓	✓	-
	48pin	R5F11BGC, R5F11BGE	✓	✓	✓	-
	64pin	R5F11BLC, R5F11BLE	✓	✓	✓	-
RL78/G1G Group (R01UH0499EJ0100)	30pin	R5F11EA8, R5F11EAA	✓	✓	✓	-
	32pin	R5F11EB8, R5F11EBA	✓	✓	✓	-
	44pin	R5F11EF8, R5F11EFA	✓	✓	✓	-
RL78/G1H Group (R01UH0575EJ0100)	64pin	R5F11FLJ, R5F11FLK, R5F11FLL	✓	✓	✓	-
RL78/H1D Group (R01UH0756EJ0080)	48pin	R5F11NGG, R5F11NGF	✓	✓	✓	-
	64pin	R5F11NLG, R5F11PLG, R5F11NLF, R5F11PLF	✓	✓	✓	-
	80pin	R5F11RMG, R5F11NMG, R5F11NMF, R5F11NME	✓	✓	✓	-
RL78/I1A Group (R01UH0169EJ0210)	20pin	R5F1076C	✓	✓	-	✓
	30pin	R5F107AC, R5F107AE	✓	✓	-	✓
	38pin	R5F107DE	✓	✓	-	✓
RL78/I1B Group (R01UH0407EJ0100)	80pin	R5F10MME, R5F10MMG	✓	✓	✓	-
	100pin	R5F10MPE, R5F10MPG	✓	✓	✓	-
RL78/I1C Group (R01UH0587EJ0210)	64pin	R5F10NLE, R5F10NLG, R5F11TLE, R5F11TLG	✓	✓	✓	-
	80pin	R5F10NME, R5F10NMG, R5F10NMJ,	✓	✓	✓	-
	100pin	R5F10NPJ, R5F10NPG	✓	✓	✓	-
RL78/I1C (512KB) Group (R01UH0889EJ0100)	80pin	R5F10NML, R5F10NML(DUAL)	✓	✓	✓	-
	100pin	R5F10NPL, R5F10NPL(DUAL)	✓	✓	✓	-

Note: RL78/I1C (512KB) User's Manual Hardware version is V1.0 Dec 2020.

Table 2-5. Supported devices

✓: Support -: Not support

Group (HW Manual number)	PIN	Device name	CS+	e² studio	AP4	Applilet3
RL78/I1D Group (R01UH0474EJ0100)	20pin	R5F11768, R5F1176A	✓	✓	✓	-
	24pin	R5F11778, R5F1177A	✓	✓	✓	-
	30pin	R5F117A8, R5F117AA, R5F117AC	✓	✓	✓	-
	32pin	R5F117BA, R5F117BC	✓	✓	✓	-
	48pin	R5F117GA, R5F117GC	✓	✓	✓	-
RL78/I1E Group (R01UH0524EJ0100)	32pin	R5F11CBC	✓	✓	✓	-
	36pin	R5F11CCC	✓	✓	✓	-
RL78/L12 Group (Windows: R01UH0330EJ0200 Linux/Mac OS: R01UH0330EJ0200)	32pin	R5F10RBC, R5F10RBA, R5F10RB8	✓	✓	-	✓
	44pin	R5F10RFC, R5F10RFA, R5F10RF8	✓	✓	-	✓
	48pin	R5F10RGC, R5F10RGA, R5F10RG8	✓	✓	-	✓
	52pin	R5F10RJC, R5F10RJA, R5F10RJ8	✓	✓	-	✓
	64pin	R5F10RLC, R5F10RLA	✓	✓	-	✓
RL78/L13 Group (Windows: R01UH0382EJ0100 Linux/Mac OS: R01UH0382EJ0100)	64pin	R5F10WLA, R5F10WLC, R5F10WLD, R5F10WLE, R5F10WLF, R5F10WLG	✓	✓	✓	-
	80pin	R5F10WMA, R5F10WMC, R5F10WMD, R5F10WME, R5F10WMF, R5F10WMG	✓	✓	✓	-
RL78/L1A Group (R01UH0636EJ0100)	80pin	R5F11MMD, R5F11MME, R5F11MMF	✓	✓	✓	-
	100pin	R5F11MPE, R5F11MPF, R5F11MPG	✓	✓	✓	-
RL78/L1C Group (R01UH0409EJ0100)	80pin	R5F110MJ, R5F110MH, R5F110MG, R5F110MF, R5F110ME, R5F111MJ, R5F111MH, R5F111MG, R5F111MF, R5F111ME	✓	✓	✓	-
	100pin	R5F110PJ, R5F110PH, R5F110PG, R5F110PF, R5F110PE, R5F111PJ, R5F111PH, R5F111PG, R5F111PF, R5F111PE	✓	✓	✓	-
RL78/D1A Group (R01UH0317EJ0003)	48pin	R5F10CGB, R5F10CGC, R5F10CGD, R5F10DGC, R5F10DGD, R5F10DGE	-	✓	-	✓
	64pin	R5F10CLD, R5F10DLD, R5F10DLE	-	✓	-	✓
	80pin	R5F10CMD, R5F10CME, R5F10DMD, R5F10DME, R5F10DMF, R5F10DMG, R5F10DMJ	-	✓	-	✓
	100pin	R5F10DPE, R5F10DPF, R5F10DPG, R5F10DPJ, R5F10TPJ	-	✓	-	✓

Note: For e² studio Linux and Mac OS version, Code Generator only support RL78/G12, RL78/G13, RL78/G14, RL78/L12, RL78/L13 group.

3. Changes

Describes the changes in this release of the Code Generator for RL78.

3.1 New support

3.1.1 Support LLVM toolchain in e^2 studio

LLVM toolchain is supported when creating projects with Code Generator for RL78/F1E, RL78/G10, RL78/G11, RL78/G1E, RL78/G1C, RL78/G1D, RL78/G1F, RL78/G1G, RL78/G1H, RL78/H1D, RL78/I1B, RL78/I1C, RL78/I1D, RL78/I1E, RL78/L1A and RL78/L1C.

3.2 Correction of issues/limitations

Table 3-1. List of Correction of issues/limitations

✓: Applicable, -: Not applicable

No	Description	Group													
		RL78/D1A	RL78/G10	RL78/G11	RL78/G12	RL78/G13, G13A	RL78/G14	RL78/G1A	RL78/G1C	RL78/G1D	RL78/G1E	RL78/G1F	RL78/G1G	RL78/G1H	RL78/H1D
1	Improvement on Code Generator view display when screen scaling setting is 100% or above	–	✓	✓	–	–	–	–	✓	✓	✓	✓	✓	✓	✓
2	Improvement on some interrupts can't be generated in LLVM project	–	–	–	✓	–	–	–	–	–	–	–	–	–	–
3	Improvement on error message in TAU input pulse interval measurement	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Table 3-2. List of Correction of issues/limitations

✓: Applicable, -: Not applicable

No	Description	Group													
		RL78/F12	RL78/F13	RL78/F14	RL78/F15	RL78/F1E	RL78/I1A	RL78/I1B	RL78/I1C	RL78/I1D	RL78/I1E	RL78/L12	RL78/L13	RL78/L1A	RL78/L1C
1	Improvement on Code Generator view display when screen scaling setting is 100% or above	–	–	–	–	✓	–	✓	✓	✓	✓	–	✓	✓	✓
2	Improvement on some interrupts can't be generated in LLVM project	–	–	–	–	–	–	–	–	–	–	–	✓	–	–
3	Improvement on error message in TAU input pulse interval measurement	–	–	–	–	–	–	–	✓	–	–	–	–	–	–

3.2.1 Improvement on Code Generator view display when screen scaling setting is 100% or above

Code Generator view may occur display issue when screen scaling setting is 100% or above.
Some controls can't display completely. This issue is improved in this release.

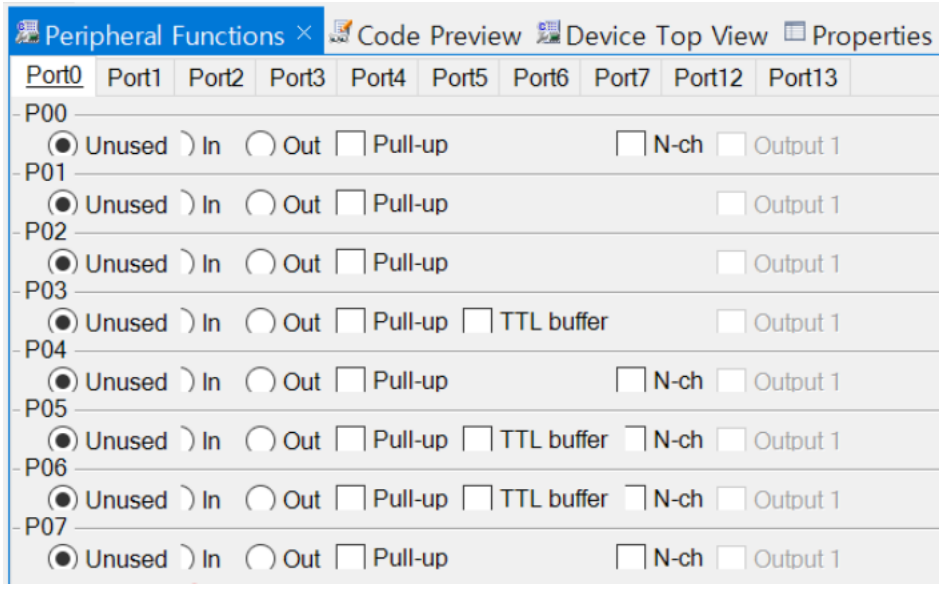


Figure 3-1 Code Generator view (before)

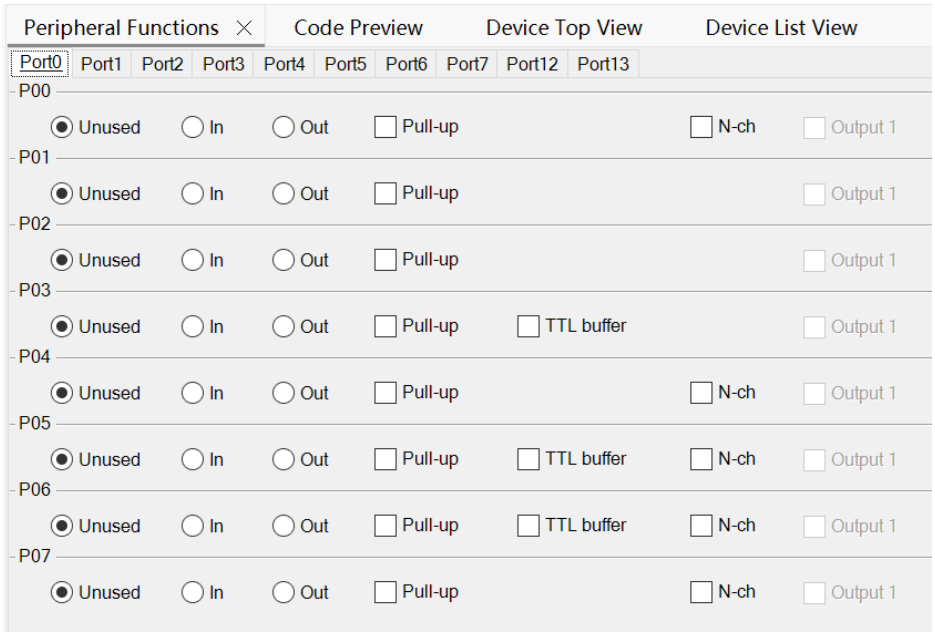


Figure 3-2 Code Generator view (after)

3.2.2 Improvement on some interrupts can't be generated in LLVM project

When using some peripherals, incorrect code will be generated and interrupts can't generate correctly.
Please refer to the document number R20TS1170 of RENESAS TOOL NEWS.
This restriction is removed from this release.

3.2.3 Improvement on error message in TAU input pulse interval measurement

When using TAU input pulse interval measurement and other TAU functions at the same time, if the user changes the setting many times, it may display an error message. This issue is improved in this release.

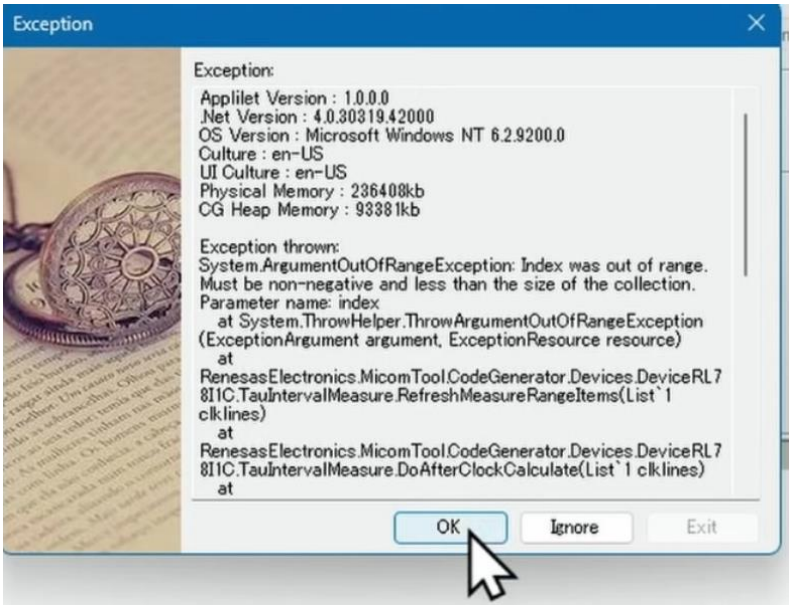


Figure 3-3 The error message

4. History of Corrections Announced in Renesas Tool News

This section is a summary of corrections announced in Renesas Tool News.

Issue Date	Document No.	Description	Device Concerned	Fixed version
May 21, 2012	120521/tn2	With generating codes for the R5F1007x and R5F1017x MCUs, RL78/G13 group	RL78/G13	CS+ V1.00.06
Aug. 01, 2012	120801/tn3	Problems arising in Applilet3 for RL78/G13 and Applilet3 for RL78/G14	RL78/G13, RL78/G14	CS+ V1.00.06
Sep. 01, 2012	120901/tn1	With using the code generator for the RL78/G12 group	RL78/G12	CS+ V1.00.06
Feb. 01, 2013	130201/tn1	With using the code generator for the RL78/G14 group of MCUs	RL78/G14	CS+ V2.00.00
Jul. 01, 2013	130701/tn1	When edited source codes disappear	RL78/F12, RL78/F13, RL78/F14, RL78/F15, RL78/G10, RL78/G12, RL78/G13, RL78/G14, RL78/G1A, RL78/G1C, RL78/G1D, RL78/G1E, RL78/G1F, RL78/G1G, RL78/G1H, RL78/I1A, RL78/I1D, RL78/I1E, RL78/L12, RL78/L13, RL78/L1C	CS+ V2.11.00
		When the port cannot be set properly	RL78/G1A	CS+ V2.00.01
Aug. 01, 2013	130801/tn1	With using the code generator for the RL78/G12 group of MCUs	RL78/G12	CS+ V2.00.01
Oct. 16, 2013	131016/tn1	2. When a RL78/G13 product in a 100-pin package is selected	RL78/G13	CS+ V2.03.00
		3. With the key input interrupt setting	RL78/L12	CS+ V2.03.00
		4. With A/D converter operation setting	RL78/G1A	CS+ V2.03.00
		5. When the timer KB20 is in use	RL78/L13	CS+ V2.03.00
Apr. 16, 2014	140416/tn5	With selecting the 20-pin, 30-pin, or 32-pin package for the RL78/F13 or RL78/F14 group	RL78/F13, RL78/F14	CS+ V2.04.00
		With using the remote control carrier wave mask signal in the RL78/L12 or RL78/L13 group	RL78/L12, RL78/L13	CS+ V2.04.00
		With processing to reflect the pin configurator when the A/D converter is set in the RL78/G12 group	RL78/G12	CS+ V2.04.00
		With the case when ports that are not available in the MCU are displayed in the RL78/G14 group	RL78/G14	CS+ V2.04.00

Issue Date	Document No.	Description	Device Concerned	Fixed version
Jul. 01, 2014	140701/tn1	With setting port 2	RL78/L13	CS+ V2.07.00
		With setting an interval timer	RL78/G10, RL78/G12, RL78/G13, RL78/G14, RL78/G1A, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C, RL78/I1A	CS+ V2.07.00
Aug. 16, 2014	140816/tn1	With setting of P20 and P21 of port2	RL78/L1C	CS+ V2.05.00
		With setting of port1	RL78/G14	CS+ V2.05.00
Nov, 1, 2014	141101/tn2	1. Point for Caution on Settings for CPU Stack Pointer Monitoring	RL78/F13	CS+ V2.07.00
		2. Point for Caution on Writing to the Serial Flag Clear Trigger Register (SIR) When Using 3-wire Serial (CSI) Transfer	RL78/F14	CS+ V2.07.00
Dec. 16, 2014	141216/tn3	1. Code Generated for Comparator Settings	RL78/I1A	CS+ V2.07.00
		2. DTC Settings	RL78/F13, RL78/F14	CS+ V2.07.00
		3. Setting the Voltage Detection Circuit to "Interrupt Mode"	RL78/L12, RL78/I1A, RL78/G1A, RL78/F13, RL78/F14	CS+ V2.07.00
		4. Saving Projects with Settings for the A/D Convertor	RL78/L1C	CS+ V2.07.00
		5. Reflection of Pin Configurations in Generated Code	RL78/G12, RL78/ G13, RL78/G14	CS+ V2.07.00
Jul. 16, 2015	150716/tn2	1. Clock Generation Circuit (PLL Circuit Operation)	RL78/F13, RL78/F14, RL78/G1C, RL78/L1C	CS+ V2.11.00 AP4 V1.10.00 Applilet3 V1.10.00
		2. Setting P40 of Port 4	RL78/F12, RL78/F13, RL78/F14, RL78/G10, RL78/G12, RL78/G13, RL78/G14, RL78/G1A, RL78/G1C, RL78/G1E, RL78/G1F, RL78/G1G, RL78/I1A, RL78/I1D, RL78/L1C, RL78/L12, RL78/L13	CS+ V2.11.00 AP4 V1.10.00 Applilet3 V1.10.00
		3. Code Generated for UART0 and UARTF	RL78/F12	CS+ V2.11.00 Applilet3 V1.10.00

Issue Date	Document No.	Description	Device Concerned	Fixed version
Nov. 16, 2015	151116/tn2	1. Indication of Channels of Serial Interface IICA	RL78/G14	CS+ V2.11.00 Applilet3 V1.10.00
		2. Procedure for Setting the PLL Clock	RL78/F13, RL78/F14, RL78/F15	CS+ V2.11.00 Applilet3 V1.10.00
Jan. 16, 2016	160116/tn5	Transfer of data with a length of 10 or more bits through an element of a serial array unit configured as a CSI or data with a length of 16 bits through an element configured as a UART	RL78/F12, RL78/F13, RL78/F14, RL78/F15, RL78/D1A	CS+ V2.11.00 Applilet3 V1.10.00
Feb. 16, 2016	160216/tn5	1. Using the error interrupt of serial array unit 4 as UART4 or DALI4	RL78/I1A	CS+ V2.11.00 Applilet3 V1.10.00
		2. Using serial array unit 4 as DALI4	RL78/I1A	CS+ V2.11.00 Applilet3 V1.10.00
Mar. 16, 2016	160316/tn1	Pin settings for the IICA serial interface when setting the PIOR to change the assignment of pin functions	RL78/G12	CS+ V2.11.00 Applilet3 V1.10.00
Jun. 16, 2016	R20TS0038EJ0100	Scan Mode of A/D Converter	RL78/F12, RL78/F13, RL78/F14, RL78/F15, RL78/G1A	CS+ V2.12.00 Applilet3 V1.11.00
Aug. 01, 2016	R20TS0045EJ0100	Peripheral I/O redirection register 0 (PIOR0)	RL78/G1F	CS+ V2.12.00 AP4 V1.11.00
Mar. 1, 2017	R20TS0139EJ0100	1. Input of Ports P10 and P11	RL78/G13 (20/24/25pin product)	CS+ V2.14.00 Applilet3 V1.13.00
		2. Port Settings Related to Reset Processing	RL78/F12 (20pin product)	CS+ V2.14.00 Applilet3 V1.13.00

Issue Date	Document No.	Description	Device Concerned	Fixed version
Dec. 16, 2017	R20TS0244EJ0100	When Continuous Transfer Mode is Selected in the CSI Configuration	RL78/D1A, RL78/F12, RL78/F13, RL78/F14, RL78/F15	CS+ V2.16.00 Applilet3 V1.15.00
			RL78/H1D, RL78/I1C, RL78/L12, RL/78L13	CS+ V2.21.00, AP4 V1.20.00 Applilet3 V1.20.00
Mar. 16, 2018	R20TS0290EJ0100	When Opening a Project for RL78/G11 Created by a Previous Version of Code Generator	RL78/G11 (20-pin R5F1056A)	CS+ V2.16.00 AP4 V1.15.00
May. 16, 2018	R20TS0313EJ0100	Writing to Port-Related Registers for Unused Pins	RL78/I1D	CS+ V2.16.00 AP4 V1.15.00
Nov. 16, 2018	R20TS0370EJ0100	When setting the Serial UART4	RL78/I1A	CS+ V2.17.00 Applilet3 V1.16.00
Jun. 1, 2019	R20TS0432EJ0100	1. PLL clock setting of clock generator	RL78/F13, RL78/F14, RL78/F15	CS+ V2.18.00 Applilet3 V1.17.00
		2. RTC operation clock setting of clock generator	RL78/F13, RL78/F14, RL78/F15	CS+ V2.18.00 Applilet3 V1.17.00
			RL78/D1A	CS+ V2.19.00 Applilet3 V1.18.00

Issue Date	Document No.	Description	Device Concerned	Fixed version
2019/08/01	R20TS0459EJ0100	1. When using IICA0 or IICA1 as a Single Master System	RL78/G10, RL78/G11, RL78/G12, RL78/G13, RL78/G14, RL78/F12, RL78/F13, RL78/F14, RL78/F15	CS+ V2.19.00 AP4 V1.18.00 Applilet3 V1.18.00
			RL78/H1D, RL78/I1C, RL78/L12, RL78/L13	CS+ V2.21.00, AP4 V1.20.00 Applilet3 V1.20.00
			RL78/G1A, RL78/G1C, RL78/G1D, RL78/G1F, RL78/G1H, RL78/I1A, RL78/I1B, RL78/L1A, RL78/L1C	Not supported
		2. When using the R_ADC_Set_ADChannel() function in the A/D converter	RL78/D1A, RL78/G1A, RL78/G1F, RL78/I1D	CS+ V2.19.00 AP4 V1.18.00 Applilet3 V1.18.00
2019/09/16	R20TS0472EJ0100	1. When using the data flash library	RL78/D1A, RL78/G11, RL78/G12, RL78/G13, RL78/G14, RL78/F12, RL78/F13, RL78/F14, RL78/F15	CS+ V2.19.00 AP4 V1.18.00 Applilet3 V1.18.00
2020/02/01	R20TS0545EJ0100	1. Callback function setting of CSI and UART 2. Operation that cancels pin function assignment of CSI and UART	RL78/G11	CS+ V2.21.00, AP4 V1.20.00
			RL78/I1E	CS+ V2.22.00, AP4 V1.21.00
2020/02/01	R20TS0544EJ0100	1. When using the trace function of the on-chip debug setting	RL78/F15	CS+ V2.21.00, Applilet3 V1.20.00
2020/05/16	R20TS0571EJ0100	1. User option byte (000C1H/010C1H) LVD off setting values	RL78/G13A	CS+ V2.21.00 Applilet3 V1.20.00

Issue Date	Document No.	Description	Device Concerned	Fixed version
2023/03/01	R20TS0926EJ0100	1. Notes on Using the Data Flash Library by Using the Data Flash API Generated by a Code Generator	RL78/F12, RL78/F13, RL78/F14, RL78/F15, RL78/G13, RL78/G13A, RL78/G14	Not supported
2025/09/20	R20TS1170EC0100	1. Notes when using LLVM compiler	RL78/L13	CS+ V2.23.00 Applilet4 V1.23.00
			RL78/G12 (30pin)	CS+ V2.23.00 Applilet3 V1.26.00

5. Restrictions

This section describes the restriction regarding the Code Generator for RL78.

5.1 List of Restrictions

Table 5-1. List of Points of Restriction (1/2)

✓: Applicable, -: Not applicable

No	Description	Group													
		RL78/D1A	RL78/G10	RL78/G11	RL78/G12	RL78/G13, G13A	RL78/G14	RL78/G1A	RL78/G1C	RL78/G1D	RL78/G1E	RL78/G1F	RL78/G1G	RL78/G1H	RL78/H1D
1	Timer array unit input clock source	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	24-pin device TAU0 channel 1 setting restriction	-	-	-	-	-	-	-	-	-	-	✓	-	-	-
3	Real-time clock API function	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-
4	Unit for 'Gain setting' of ΔΣ A/D CONVERTER	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
5	Restrictions on CSI continuous transfer mode	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	-
6	Incorrect function description in data lash library	-	-	✓	✓	✓	✓	-	-	-	-	-	-	-	-
7	Operation clock when fSUB and fIL are selected in TAU input pulse interval measurement	-	-	-	-	✓	✓	-	-	-	-	-	-	-	-
8	64-bit environment restrictions	✓	-	-	-	-	-	-	-	-	-	-	-	-	-
9	Incorrect content of R_ELC_Stop() function	-	-	-	-	-	-	-	-	-	-	✓	✓	✓	-
10	The trace address is incorrect	-	-	-	-	-	-	-	-	-	-	✓	-	-	-
11	R_ELC_Stop() function build error in IAR Embedded Workbench	-	-	-	-	-	✓	-	-	-	-	-	-	-	-
12	Generate multiple [Code Generator] categories	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
13	TRDIOA1, TRDIOB1, TRDIOC1 and TRDIOD1 pin output forced cutoff control codes are not generated in Timer RD	-	-	-	-	-	-	-	-	-	-	✓	-	-	-
14	TI05 occupies the pin always when using TAU input pulse interval measurement	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	SCK10 pin setting is wrong when using CSI10 with external clock	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	R_COMP_Set_PowerOff() function should remove from R_Systeminit()	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	Pin settings are wrong when using TAU, INTP, TMRD or CSI	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	Incorrect content of R_RTC_Get_CounterValue() and R_RTC_Set_CounterValue() function	✓	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 5-2. List of Points of Restriction (2/2)

✓: Applicable, -: Not applicable

No	Description	Group													
		RL78/D1A	RL78/G10	RL78/G11	RL78/G12	RL78/G13, G13A	RL78/G14	RL78/G1A	RL78/G1C	RL78/G1D	RL78/G1E	RL78/G1F	RL78/G1G	RL78/G1H	RL78/H1D
19	The information of interval time controller is wrong in WDT	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	Overflow time can’t load correctly in WDT	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 5-3. List of Points of Restriction (1/2)

✓: Applicable, -: Not applicable

No	Description	Group													
		RL78/F12	RL78/F13	RL78/F14	RL78/F15	RL78/F1E	RL78/I1A	RL78/I1B	RL78/I1C	RL78/I1D	RL78/I1E	RL78/L12	RL78/L13	RL78/L1A	RL78/L1C
1	Timer array unit input clock source	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-
2	24-pin device TAU0 channel 1 setting restriction	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	Real-time clock API function	-	-	-	-	-	-	-	-	-	✓	-	-	-	-
4	Unit for 'Gain setting' of ΔΣ A/D CONVERTER	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	Restrictions on CSI continuous transfer mode	-	-	-	-	-	✓	✓	-	✓	✓	-	-	✓	✓
6	Incorrect function description in data lash library	✓	✓	✓	✓	-	-	-	-	-	✓	-	-	-	-
7	Operation clock when fSUB and fIL are selected in TAU input pulse interval measurement	✓	-	-	-	-	-	-	-	-	-	-	-	-	-
8	64-bit environment restrictions	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	Incorrect content of R_ELC_Stop() function	-	-	-	-	-	-	-	-	-	-	✓	✓	✓	-
10	The trace address is incorrect	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	R_ELC_Stop() function build error in IAR Embedded Workbench	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	Generate multiple [Code Generator] categories	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
13	TRDIOA1, TRDIOB1, TRDIOC1 and TRDIOD1 pin output forced cutoff control codes are not generated in Timer RD	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	TI05 occupies the pin always when using TAU input pulse interval measurement	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
15	SCK10 pin setting is wrong when using CSI10 with external clock	-	-	-	-	-	-	-	-	-	-	-	✓	-	-
16	R_COMP_Set_PowerOff() function should remove from R_Systeminit()	-	-	-	-	-	-	-	-	-	-	-	✓	-	-
17	Pin settings are wrong when using TAU, INTP, TMRD or CSI	-	✓	-	-	-	-	-	-	-	-	-	-	-	-
18	Incorrect content of R_RTC_Get_CounterValue() and R_RTC_Set_CounterValue() function	✓	✓	✓	✓	-	-	-	-	-	-	-	-	-	-

Table 5-4. List of Points of Restriction (2/2)

✓: Applicable, -: Not applicable

No	Description	Group													
		RL78/F12	RL78/F13	RL78/F14	RL78/F15	RL78/F1E	RL78/I1A	RL78/I1B	RL78/I1C	RL78/I1D	RL78/I1E	RL78/L12	RL78/L13	RL78/L1A	RL78/L1C
19	The information of interval time controller is wrong in WDT	-	-	-	✓	-	-	-	-	-	-	-	-	-	-
20	Overflow time can't load correctly in WDT	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-

5.2 Details for Restriction

5.2.1 Timer array unit input clock source

When the clock source of a timer input is set as a RTC1HZ output by setup of a timer array unit, a setup about the output of the RTC1HZ terminal of a real-time clock becomes invalid. The code which outputs RTC1HZ then is not generated.

[Workaround] When you set to a RTC1HZ signal by setup of a timer array unit, please choose a setup which uses a real-time clock and add the code which outputs RTC1HZ.

5.2.2 24-pin device TAU0 channel 1 setting restriction

In the 24-pin device, interval timer is only selectable for the TAU0 channel 1 setting.

[Workaround] There is no workaround.
In the 32-pin device, other timer functions besides “Interval timer” are selectable for the TAU0 channel 1 setting. Refer to the setting to make a correction.

5.2.3 Real-time clock API function

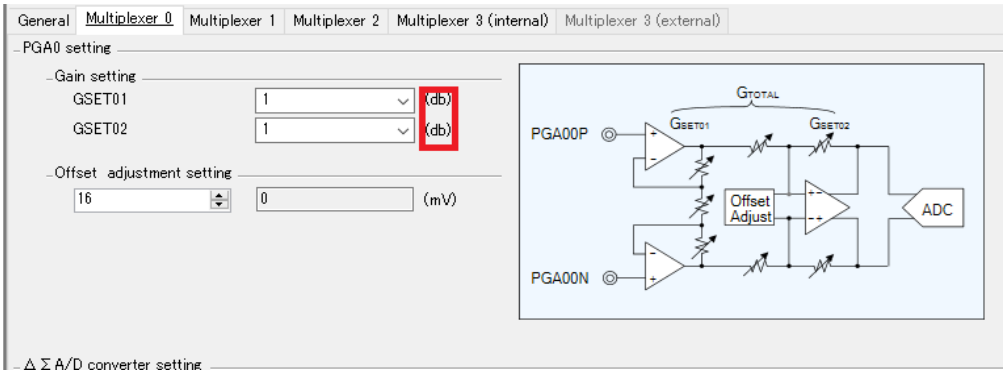
An unnecessary wait time code is output in the R_RTC_Set_AlarmOn().

```
/* Change the waiting time according to the system */
for (w_count = 0U; w_count < RTC_WAITTIME_2FRTC; w_count++)
{
   NOP();
}
```

[Workaround] There is no workaround.
I Delete the wait time code in the R_RTC_Set_AlarmOn () function after generating the code.

5.2.4 Unit for 'Gain setting' of ΔΣ A/D CONVERTER

The unit of Multiplexer 0/1/2/3(Internal)/3(external) are 'db' but it should be 'Gain'.



[Workaround] Please interpret ‘db’ as ‘Gain’ when use GSET01 and-or GSET02.

5.2.5 Restrictions on CSI continuous transfer mode

When CSI is used in continuous transfer mode, 2 bytes are received even if 1 is specified in the function argument.

[Workaround] Change the code in the red frame below. If code generation is executed again after changing the code, the code will be overwritten and deleted, so be caution.

[R_CSIn_Receive() function] case with CSI00

[Before]

```
MD_STATUS R_CSI00_Receive(uint8_t * const rx_buf, uint16_t rx_num)
{
    MD_STATUS status = MD_OK;

    if (rx_num < 1U)
    {
        status = MD_ARGERROR;
    }
    else
    {
        SMR00 |= _0001_SAU_BUFFER_EMPTY;
        g_csi00_rx_length = rx_num;    /* receive data length */
        g_csi00_rx_count = 0U;         /* receive data count */
        gp_csi00_rx_address = rx_buf;  /* receive buffer pointer */
        SIO00 = 0xFFU;    /* start receive by dummy write */
    }

    return (status);
}
```

[After]

```
MD_STATUS R_CSI00_Receive(uint8_t * const rx_buf, uint16_t rx_num)
{
    MD_STATUS status = MD_OK;

    if (rx_num < 1U)
    {
        status = MD_ARGERROR;
    }
    else
    {
        if ( 1U == rx_num )
        {
            SMR00 &= ~_0001_SAU_BUFFER_EMPTY;
        }
        else
        {
            SMR00 |= _0001_SAU_BUFFER_EMPTY;
        }
        g_csi00_rx_length = rx_num;    /* receive data length */
        g_csi00_rx_count = 0U;         /* receive data count */
        gp_csi00_rx_address = rx_buf;  /* receive buffer pointer */
        SIO00 = 0xFFU;    /* start receive by dummy write */
    }

    return (status);
}
```

[R_CSIn_Send_Receive() function] case with CSI00

[Before]

```
MD_STATUS R_CSI00_Send_Receive(uint8_t * const tx_buf, uint16_t tx_num, uint8_t * const rx_buf)
{
    MD_STATUS status = MD_OK;

    if (tx_num < 1U)
    {
        status = MD_ARGERROR;
    }
    else
    {
        g_csi00_send_length = tx_num;     /* send data length */
        g_csi00_tx_count = tx_num;        /* send data count */
        gp_csi00_tx_address = tx_buf;     /* send buffer pointer */
        gp_csi00_rx_address = rx_buf;     /* receive buffer pointer */
        SMR00 |= _0001_SAU_BUFFER_EMPTY;
        CSIMK00 = 1U;                     /* disable INTCSI00 interrupt */

        if (0U != gp_csi00_tx_address)
        {
            SIO00 = *gp_csi00_tx_address;    /* started by writing data to SDR[7:0] */
            gp_csi00_tx_address++;
        }
        else
        {
            SIO00 = 0xFFU;
        }

        g_csi00_tx_count--;
        CSIMK00 = 0U;                     /* enable INTCSI00 interrupt */
    }

    return (status);
}
```

[After]

```
MD_STATUS R_CSI00_Send_Receive(uint8_t * const tx_buf, uint16_t tx_num, uint8_t * const rx_buf)
{
    MD_STATUS status = MD_OK;

    if (tx_num < 1U)
    {
        status = MD_ARGERROR;
    }
    else
    {
        g_csi00_send_length = tx_num;     /* send data length */
        g_csi00_tx_count = tx_num;        /* send data count */
        gp_csi00_tx_address = tx_buf;     /* send buffer pointer */
        gp_csi00_rx_address = rx_buf;     /* receive buffer pointer */
        if ( 1U == tx_num )
        {
            SMR00 &= ~_0001_SAU_BUFFER_EMPTY;
        }
        else
        {
            SMR00 |= _0001_SAU_BUFFER_EMPTY;
        }
        CSIMK00 = 1U;                     /* disable INTCSI00 interrupt */

        if (0U != gp_csi00_tx_address)
        {
            SIO00 = *gp_csi00_tx_address;    /* started by writing data to SDR[7:0] */
            gp_csi00_tx_address++;
        }
        else
        {
            SIO00 = 0xFFU;
        }

        g_csi00_tx_count--;
        CSIMK00 = 0U;                     /* enable INTCSI00 interrupt */
    }

    return (status);
}
```

5.2.6 Incorrect function description in data lash library

There is an erroneous in the description of the R_FDL_BlankCheck () function and the R_FDL_Iverify () function. The description in Code Generator RL78 API Reference (P736) is the correct explanation. Please refer to it.

[Workaround] The code is not affected.

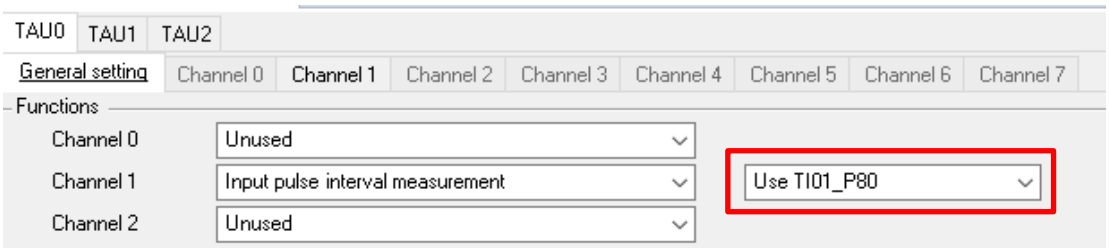
5.2.7 Operation clock when fSUB and fIL are selected in TAU input pulse interval measurement

When input pulse interval measurement is specified with TAU and fSUB and fIL are selected, the division ratio is fixed to fclk/2^8. Due to the fixed operation clock, the intended detection accuracy may not be achieved in the safety function frequency detection.

[Workaround] After code generation, change the operating clock (f_{MCK}) of the timer mode register from CK00 to CK01. If code generation is executed again after changing the code, the code will be overwritten and deleted, so be caution.

5.2.8 64-bit environment restrictions

After loading project (e^2 studio project or AP3 project) which is saved on 32-bit environment in 64-bit environment, TAU0 channel1 input selection (refer to the red box in the figure below) can't be kept.



[Workaround] The user should confirm this GUI setting after loading project in such a case.

5.2.9 Incorrect code in R_ELC_Stop() function

The content of API R_ELC_Stop() should be fixed as "*sfr_addr = _00_ELC_EVENT_LINK_OFF;", but it generates other code sometimes.

[Workaround] Change the code in the red frame below. If code generation is executed again after changing the code, the code will be overwritten and deleted, so be caution.

[Before]

```
void R_ELC_Stop(uint32_t event)
{
    volatile uint32_t                  w_count;
    volatile uint8_t __no_bit_access * sfr_addr;

    sfr_addr = &ELSELR00;

    for (w_count = 0U; w_count < ELC_DESTINATION_COUNT; w_count++)
    {
        if (0x1U == ((event >> w_count) & 0x1U))
        {
            *sfr_addr = _04_ELC_EVENT_LINK_TAU05;
        }

        sfr_addr++;
    }
}
```

[After]

```
void R_ELC_Stop(uint32_t event)
{
    volatile uint32_t                  w_count;
    volatile uint8_t __no_bit_access * sfr_addr;

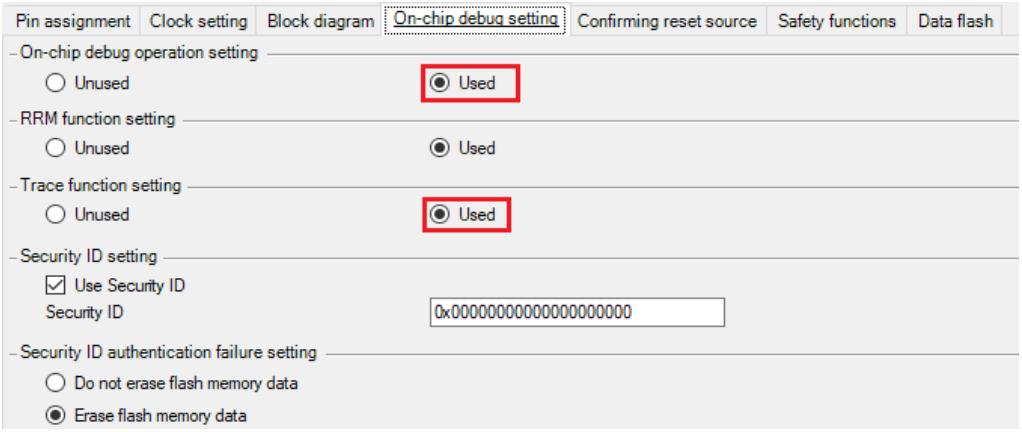
    sfr_addr = &ELSELR00;

    for (w_count = 0U; w_count < ELC_DESTINATION_COUNT; w_count++)
    {
        if (0x1U == ((event >> w_count) & 0x1U))
        {
            *sfr_addr = _00_ELC_EVENT_LINK_OFF;
        }

        sfr_addr++;
    }
}
```

5.2.10 The trace address is incorrect

When OCD and trace is used (as following picture) and IAR compiler is selected, the trace address should be 0xFED00 in r_cg_main.c.



[Workaround] Change the code in the red frame below. If code generation is executed again after changing the code, the code will be overwritten and deleted, so be caution.

Before:

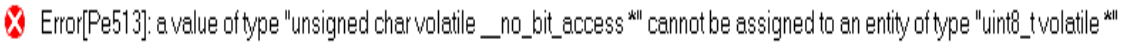
```
/* Secure trace RAM area */
__no_init __root unsigned char ocdtraceram[1024] @ 0xFE300;
```

After:

```
/* Secure trace RAM area */
__no_init __root unsigned char ocdtraceram[1024] @ 0xFED00;
```

5.2.11 R_ELC_Stop() function build error in IAR Embedded Workbench

When creating IAR project of RL78/G14 and using ELC function, it has build error as following picture:



[Workaround] Add the code in the red frame below. If code generation is executed again after changing the code, the code will be overwritten and deleted, so be caution.

```
void R_ELC_Stop(uint32_t event)
{
    volatile uint32_t   w_count;
    volatile uint8_t __no_bit_access * sfr_addr;

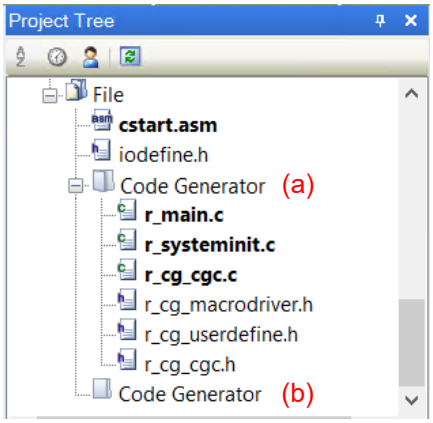
    sfr_addr = &ELSELR00;

    for (w_count = 0U; w_count < ELC_DESTINATION_COUNT; w_count++)
    {
        if (((event >> w_count) & 0x1U) == 0x1U)
        {
            *sfr_addr = _00_ELC_EVENT_LINK_OFF;
        }

        sfr_addr++;
    }
}
```

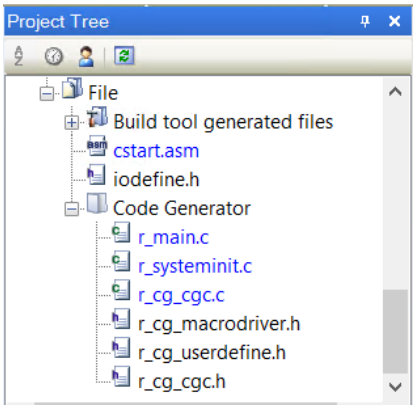
5.2.12 Generate multiple [Code Generator] categories

There are cases where multiple [Code Generator] categories are generated, as shown below.



[Workaround] You can remove unwanted [Code Generator] categories by following the steps below.

1. Select the [Code Generator] category in (a) and remove it from the project.
2. Make sure it has been removed from your project and run the generated code.



5.2.13 TRDIOA1, TRDIOB1, TRDIOC1 and TRDIOD1 pin output forced cutoff control codes are not generated in Timer RD

When the “PWM mode (up to 6 PWM outputs)”, “Reset synchronous PWM mode” or “Complementary PWM mode” functions of Timer RD are selected, the TRDIOA1, TRDIOB1, TRDIOC1 and TRDIOD1 pin output forced cutoff control codes are not generated after enabled by selecting “Comparator 0 output”, “INTP0 pin input” or “Event input from ELC”.



[Workaround] Add OPDF1 register code according to requirement as it does in the red frame below. If code generation is executed again after adding the code, the code will be overwritten and deleted, so be caution.

The macro definition of OPDM1 register in r_cg_tmrd.h.

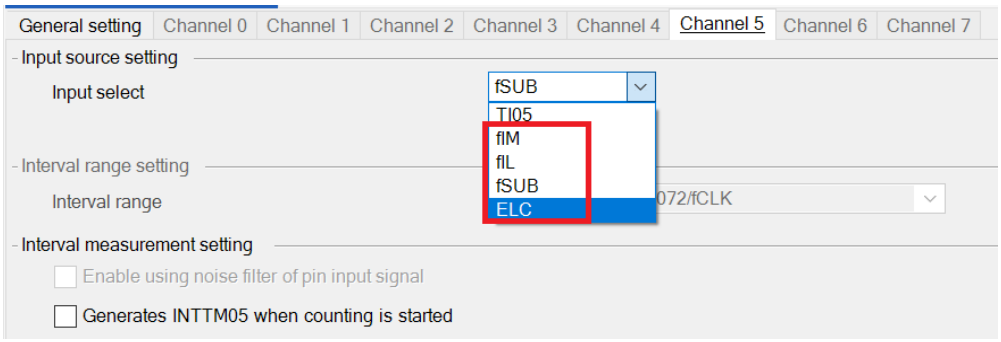
```
/*
    PWMOP cutoff control register m (OPDFm)
*/
/* TRDIODm pin output forced cutoff control (DFDm1,DFDm0) */
#define _00_TRD_PWMOP_TRDIODm_PROHIBIT              (0x00U) /* forced cutoff prohibited */
#define _40_TRD_PWMOP_TRDIODm_HIIMPEDANCE           (0x40U) /* hi-impedance output */
#define _80_TRD_PWMOP_TRDIODm_LOW                   (0x80U) /* low Level output */
#define _C0_TRD_PWMOP_TRDIODm_HIGH                  (0xC0U) /* high Level output */
/* TRDIOCm pin output forced cutoff control (DFCm1,DFCm0) */
#define _00_TRD_PWMOP_TRDIOCm_PROHIBIT              (0x00U) /* forced cutoff prohibited */
#define _10_TRD_PWMOP_TRDIOCm_HIIMPEDANCE           (0x10U) /* hi-impedance output */
#define _20_TRD_PWMOP_TRDIOCm_LOW                   (0x20U) /* low Level output */
#define _30_TRD_PWMOP_TRDIOCm_HIGH                  (0x30U) /* high Level output */
/* TRDIOBm pin output forced cutoff control (DFBm1,DFBm0) */
#define _00_TRD_PWMOP_TRDIOBm_PROHIBIT              (0x00U) /* forced cutoff prohibited */
#define _04_TRD_PWMOP_TRDIOBm_HIIMPEDANCE           (0x04U) /* hi-impedance output */
#define _08_TRD_PWMOP_TRDIOBm_LOW                   (0x08U) /* low Level output */
#define _0C_TRD_PWMOP_TRDIOBm_HIGH                  (0x0CU) /* high Level output */
/* TRDIOAm pin output forced cutoff control (DFAm1,DFAm0) */
#define _00_TRD_PWMOP_TRDIOAm_PROHIBIT              (0x00U) /* forced cutoff prohibited */
#define _01_TRD_PWMOP_TRDIOAm_HIIMPEDANCE           (0x01U) /* hi-impedance output */
#define _02_TRD_PWMOP_TRDIOAm_LOW                   (0x02U) /* low Level output */
#define _03_TRD_PWMOP_TRDIOAm_HIGH                  (0x03U) /* high Level output */
```

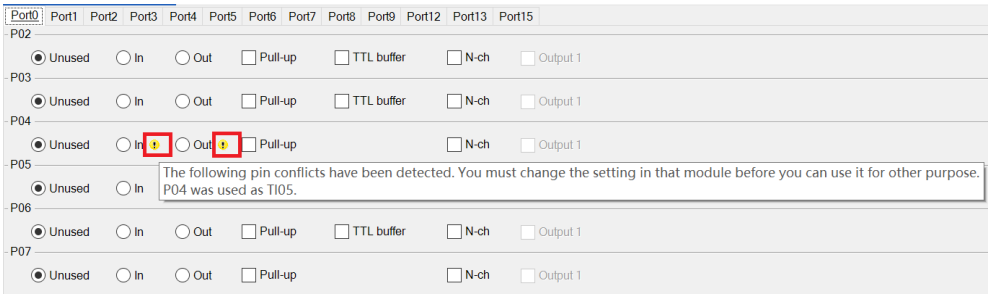
The TMRD create function in r_cg_tmrd.c.

```
void R_TMRD0_Create(void)
{
    TRD0EN = 1U;    /* enables input clock supply */
    PWMOPEN = 1U;   /* enables input clock supply */
    TRDSTR |= _04_TMRD_TRD0_COUNT_CONTINUES | _08_TMRD_TRD1_COUNT_CONTINUES;
    TRDSTR &= (uint8_t)~(_02_TMRD_TRD1_COUNT_START | _01_TMRD_TRD0_COUNT_START);
    ......
    OPCTL0 |= _08_TRD_PWMOP_COMP_CUTOFF;
    OPDF0 = _40_TRD_PWMOP_TRDIODm_HIIMPEDANCE | _10_TRD_PWMOP_TRDIOCm_HIIMPEDANCE | _04_TRD
    OPDF1 = _40_TRD_PWMOP_TRDIODm_HIIMPEDANCE | _10_TRD_PWMOP_TRDIOCm_HIIMPEDANCE |
            _04_TRD_PWMOP_TRDIOBm_HIIMPEDANCE | _01_TRD_PWMOP_TRDIOAm_HIIMPEDANCE;
    OPEDGE = _00_TRD_PWMOP_RELEASE_TRDIOC0_R;
    /* Set TRDIOB0 pin */
    POM1 &= 0xDFU;
    P1 &= 0xDFU;
    PM1 &= 0xDFU;
```

5.2.14 TI05 occupies the pin always when using TAU input pulse interval measurement

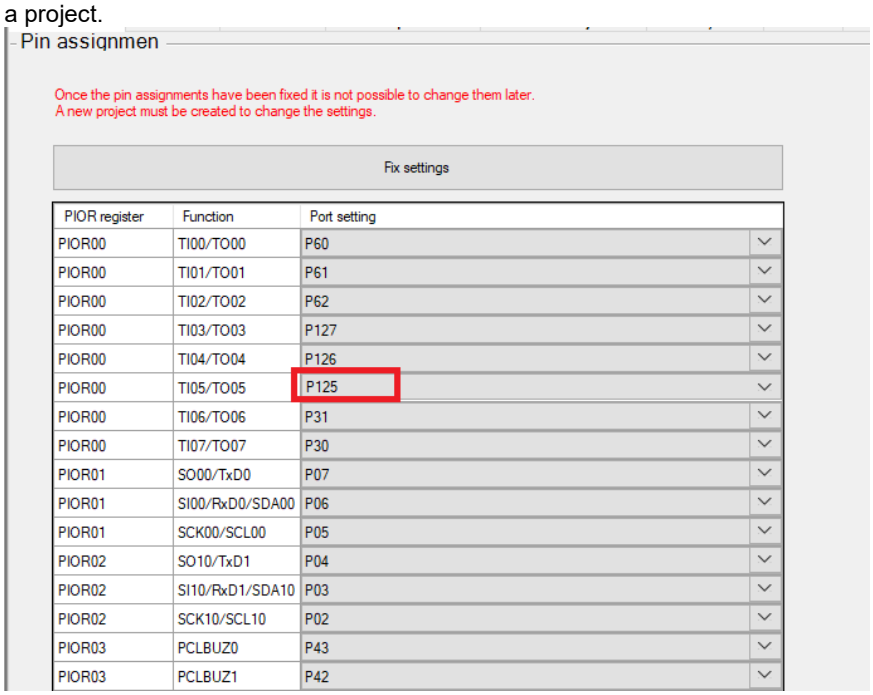
When selecting other than TI05 as the input source in TAU input pulse interval measurement, alternative I/O port is not free.



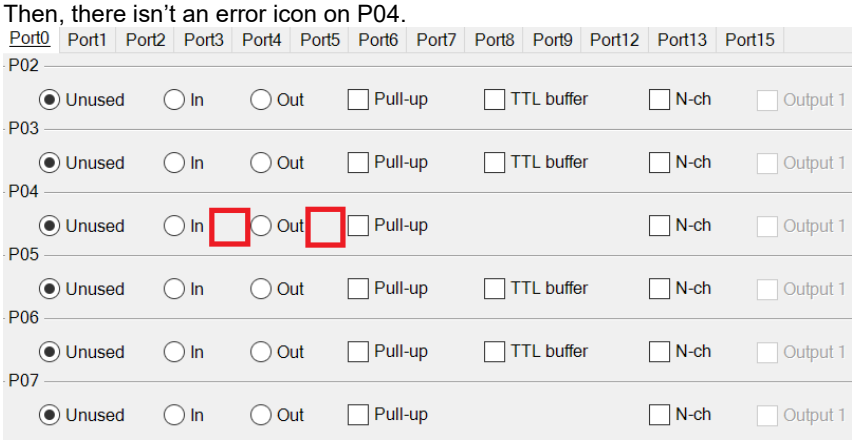


[Workaround] Set TI05 to the pin which wouldn't be used.

For example, if the user wants to use P04 in PORT, please set TI05 to P125 when creating a project.

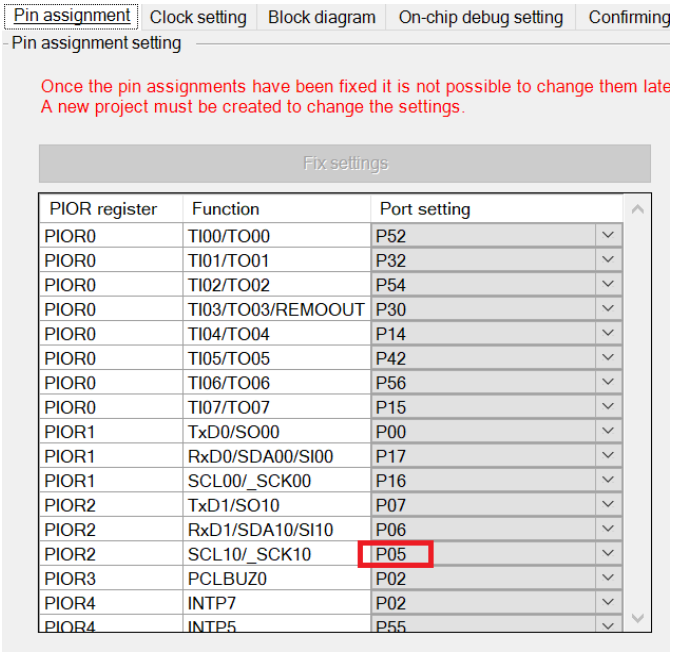


Then, there isn't an error icon on P04.



5.2.15 SCK10 pin setting is wrong when using CSI10 with external clock

When setting SCK10 to P05 and using CS10 with external clock, the SCK10 pin setting is wrong.





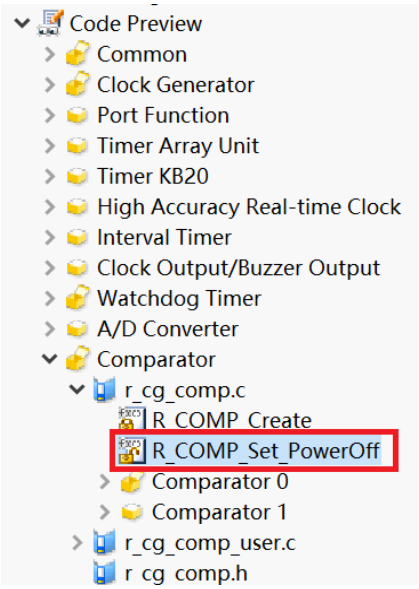
[Workaround] Change pin setting manually in Create(). If code generation is executed again after adding the code, the code will be overwritten, so be caution.

[Before]
```
/* Set SCK10 pin */
PFSEG6 &= 0xFDU;
PM1 |= 0x20U;
```

[After]
```
/* Set SCK10 pin */
PFSEG6 &= 0xFDU;
PM0 |= 0x20U;
```

5.2.16 R_COMP_Set_PowerOff() function should remove from R_Systeminit()

When the user locks R_COMP_Set_PowerOff() on Code preview tree. This API will be called by R_Systeminit().




```
void R_Systeminit(void)
{
    PIOR = 0x00U;
    R_CGC_Get_ResetSource();
    R_CGC_Create();
    R_WDT_Create();
    R_COMP_Create();
    R_COMP_Set_PowerOff();
    IAWCTL = 0x00U;
}
```

[Workaround] Remove R_COMP_Set_PowerOff() manually. If code generation is executed again after adding the code, the code will be overwritten, so be caution.

5.2.17 Pin settings are wrong when using TAU, INTP, TMRD or CSI

When using an alternative function as input/output which is occupied on P120 or P125, Code Generator lacks setting PMC register. This error occurs in the following situations:

- The chips: Group A in User's Manual. It includes R5F10A6A, R5F10A6C, R5F10A6D, R5F10A6E, R5F10AAA, R5F10AAC, R5F10AAD, R5F10AAE, R5F10ABA, R5F10ABC, R5F10ABD, R5F10ABE, R5F10AGA, R5F10AGC, R5F10AGD, R5F10AGE, R5F10ALC, R5F10ALD, R5F10ALE
- The pins: TI07/INTP4/TRDIOD0/SO01/TO07 (P120)
  TI03/INTP1/TO03/TRDIOB0/SSI01 (P125)

[Workaround] Change pin setting manually in Create(). If code generation is executed again after adding the code, the code will be overwritten, so be caution.
For example:
Example for used as alternative function on P120 case.
[Before]

```
/* Set SO01 pin */
PSRSEL &= 0xF7U;
P12 |= 0x01U;
PM12 &= 0xFEU;
```

[After]

```
/* Set SO01 pin */
PSRSEL &= 0xF7U;
PMC12 &= 0xFEU;
P12 |= 0x01U;
PM12 &= 0xFEU;
```

Example for used as alternative function on P125 case.
[Before]

```
/* Set TO03 pin */
P12 &= 0xDFU;
PM12 &= 0xDFU;
```

[After]

```
/* Set TO03 pin */
PMC12 &= 0xDFU;
P12 &= 0xDFU;
PM12 &= 0xDFU;
```

5.2.18 Incorrect content of R_RTC_Get_CounterValue() and R_RTC_Set_CounterValue() function

When reading or writing to the counter while generation of the alarm interrupt is enabled, the constant-period interrupt should be set to once per 1 second. Current R_RTC_Get_CounterValue() and R_RTC_Set_CounterValue() function don't support this.

[Workaround] Change the content of R_RTC_Get_CounterValue() and R_RTC_Set_CounterValue() function manually. If code generation is executed again after adding the code, the code will be overwritten, so be caution.

r_cg_rtc.c:

[Before]

```
/***********************************************************************
Global variables and functions
***********************************************************************/
/* Start user code for global. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */
```

[After]

```
/***********************************************************************
Global variables and functions
***********************************************************************/
/* Start user code for global. Do not edit comment generated here */
volatile uint8_t g_rtc_interrupt_flag;    /* rtc interrupt user flag */
/* End user code. Do not edit comment generated here */
```

R_RTC_Get_CounterValue():

[Before]

```
MD_STATUS status = MD_OK;
uint16_t  w_count;

RTCC1 |= _01_RTC_COUNTER_PAUSE;

/* Change the waiting time according to the system */
for (w_count = 0U; w_count < RTC_WAITTIME; w_count++)
{
    NOP();
}

if (0U == RWST)
{
    status = MD_BUSY1;
}
else
{
    counter_read_val->sec = SEC;
    counter_read_val->min = MIN;
    counter_read_val->hour = HOUR;
    counter_read_val->week = WEEK;
    counter_read_val->day = DAY;
    counter_read_val->month = MONTH;
    counter_read_val->year = YEAR;

    RTCC1 &= (uint8_t)~_01_RTC_COUNTER_PAUSE;

    /* Change the waiting time according to the system */
    for (w_count = 0U; w_count < RTC_WAITTIME; w_count++)
    {
        NOP();
    }

    if (1U == RWST)
    {
        status = MD_BUSY2;
    }
}

return (status);
```

[After]

```
MD_STATUS status = MD_OK;
uint16_t  w_count;
uint8_t temp;
uint8_t temp1 = RTCC0;

if (1U == WALIE)
{
    temp = RTCC0 & 0x07;
    /* Disable INTRTC interrupt */
    RTCMK = 1U;
    /* Set the fixed-cycle interrupt to once per second */
    temp1 &= _F8_RTC_INTRTC_CLEAR;
    temp1 |= _02_RTC_INTRTC_CLOCK_1;
    RTCC0 = temp1;
    /* Clear the fixed-cycle interrupt status flag */
    RIFG = 0U;
    /* Clear the INTRTC interrupt request flag */
    RTCIF = 0U;
    /* Enable INTRTC interrupt */
    RTCMK = 0U;
    /* 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;
}
```

```
RTCC1 |= _01_RTC_COUNTER_PAUSE;

/* Change the waiting time according to the system */
for (w_count = 0U; w_count < RTC_WAITTIME; w_count++)
{
    NOP();
}

if (0U == RWST)
{
    status = MD_BUSY1;
}
else
{
    counter_read_val->sec = SEC;
    counter_read_val->min = MIN;
    counter_read_val->hour = HOUR;
    counter_read_val->week = WEEK;
    counter_read_val->day = DAY;
    counter_read_val->month = MONTH;
    counter_read_val->year = YEAR;

    RTCC1 &= (uint8_t)~_01_RTC_COUNTER_PAUSE;

    /* Change the waiting time according to the system */
    for (w_count = 0U; w_count < RTC_WAITTIME; w_count++)
    {
        NOP();
    }

    if (1U == RWST)
    {
        status = MD_BUSY2;
    }
    else if (1U == WALIE)
    {
        /* Recover the setting of fixed-cycle interrupt after read counter value */
        RTCC0 &= _F8_RTC_INTRTC_CLEAR;
        RTCC0 |= temp;
        /* Clear the fixed-cycle interrupt status flag */
        RIFG = 0U;
        /* Clear the INTRTC interrupt request flag */
        RTCIF = 0U;
        /* Enable INTRTC interrupt */
        RTCMK = 0U;
    }
}

return (status);
```

R_RTC_Set_CounterValue():

[Before]

```
MD_STATUS status = MD_OK;
uint16_t  w_count;

RTCC1 |= _01_RTC_COUNTER_PAUSE;

/* Change the waiting time according to the system */
for (w_count = 0U; w_count < RTC_WAITTIME; w_count++)
{
    NOP();
}

if (0U == RWST)
{
    status = MD_BUSY1;
}
else
{
    SEC = counter_write_val.sec;
    MIN = counter_write_val.min;
    HOUR = counter_write_val.hour;
    WEEK = counter_write_val.week;
    DAY = counter_write_val.day;
    MONTH = counter_write_val.month;
    YEAR = counter_write_val.year;
    RTCC1 &= (uint8_t)~_01_RTC_COUNTER_PAUSE;

    /* Change the waiting time according to the system */
    for (w_count = 0U; w_count < RTC_WAITTIME; w_count++)
    {
        NOP();
    }

    if (1U == RWST)
    {
        status = MD_BUSY2;
    }
}

return (status);
```

[After]

```
MD_STATUS status = MD_OK;
uint16_t  w_count;
uint8_t temp;
uint8_t temp1 = RTCC0;

if (1U == WALIE)
{
    temp = RTCC0 & 0x07;
    /* Disable INTRTC interrupt */
    RTCMK = 1U;
    /* Set the fixed-cycle interrupt to once per second */
    temp1 &= _F8_RTC_INTRTC_CLEAR;
    temp1 |= _02_RTC_INTRTC_CLOCK_1;
    RTCC0 = temp1;
    /* Clear the fixed-cycle interrupt status flag */
    RIFG = 0U;
    /* Clear the INTRTC interrupt request flag */
    RTCIF = 0U;
    /* Enable INTRTC interrupt */
    RTCMK = 0U;
    /* 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;
}

RTCC1 |= _01_RTC_COUNTER_PAUSE;

/* Change the waiting time according to the system */
for (w_count = 0U; w_count < RTC_WAITTIME; w_count++)
{
    NOP();
}

if (0U == RWST)
{
    status = MD_BUSY1;
}
else
{
    SEC = counter_write_val.sec;
    MIN = counter_write_val.min;
    HOUR = counter_write_val.hour;
    WEEK = counter_write_val.week;
    DAY = counter_write_val.day;
    MONTH = counter_write_val.month;
    YEAR = counter_write_val.year;
    RTCC1 &= (uint8_t)~_01_RTC_COUNTER_PAUSE;

    /* Change the waiting time according to the system */
    for (w_count = 0U; w_count < RTC_WAITTIME; w_count++)
    {
        NOP();
    }
```

```
    if (1U == RWST)
    {
        status = MD_BUSY2;
    }
    else if (1U == WALIE)
    {
        /* Recover the setting of fixed-cycle interrupt after write counter value */
        RTCC0 &= _F8_RTC_INTRTC_CLEAR;
        RTCC0 |= temp;
        /* Clear the alarm detection status flag */
        WAFG = 0U;
        /* Clear the fixed-cycle interrupt status flag */
        RIFG = 0U;
        /* Clear the INTRTC interrupt request flag */
        RTCIF = 0U;
        /* Enable INTRTC interrupt */
        RTCMK = 0U;
    }
}

return (status);
```

r_cg_rtc_user.c

[Before]

```
/***********************************************************************
Global variables and functions
***********************************************************************/
/* Start user code for global. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */
```

[After]

```
/***********************************************************************
Global variables and functions
***********************************************************************/
/* Start user code for global. Do not edit comment generated here */
extern volatile uint8_t g_rtc_interrupt_flag;
/* End user code. Do not edit comment generated here */
```

r_rtc_interrupt():

[Before]

```
static void __near r_rtc_interrupt(void)
{
    ...
}
```

[After]

```
static void __near r_rtc_interrupt(void)
{
    g_rtc_interrupt_flag = 1U;
    ...
}
```

5.2.19 The information of interval time controller is wrong in WDT

When using WDT, please notice the information of interval time controller is wrong.

[Current]



[Correct]

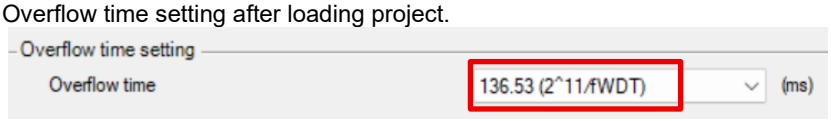


5.2.20 Overflow time can’t load correctly in WDT

When the overflow time is set to 2^13/fWDT, 2^14/fWDT, or 2^16/fWDT and the project is saved, the overflow time setting can’t load correctly

[Current]

Overflow time setting after loading project.



[Workaround]

Please re-set overflow time by manual after loading the project.

6. Points for Caution

This section describes points for caution regarding the Code Generator for RL78.

6.1 List for Cautions

Table 6-1. List of Points for Caution (1/2)

✓: Applicable, -: Not applicable

No	Description	Group													
		RL78/D1A	RL78/G10	RL78/G11	RL78/G12	RL78/G13, G13A	RL78/G14	RL78/G1A	RL78/G1C	RL78/G1D	RL78/G1E	RL78/G1F	RL78/G1G	RL78/G1H	RL78/H1D
1	Online Help (Applilet3, AP4)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	Coding rule of MISRA-C.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	High-speed on-chip oscillator frequency select register	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	Internal low-speed or internal high-speed oscillator trimming	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	Serial array unit	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	Flash memory CRC operation function (high-speed CRC)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
7	Port mode select register (PMS)	✓	✓	-	-	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
8	LIN-bus function of UART	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
9	Extension code, multi-master and wakeup function of serial interface IICA and multi-master function of simple IIC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
10	CAN controllers	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	Safety Functions	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
12	USB	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	RI78V4 project	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	DTC function (CS+ for CA,CX)	✓	-	-	-	✓	-	-	-	-	-	-	-	-	-
15	High Speed DTC chain transfer	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	Fast Mode Plus setting in IICA slave	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
17	high-speed on-chip oscillator (CS+ for CA,CX)	-	-	✓	✓	✓	✓	-	-	-	-	-	-	-	-
18	Pin Configurator (CS+ for CA,CX)	-	-	✓	✓	✓	✓	-	-	-	-	-	-	-	-
19	Version notation of RL78/G13A generation file.	-	-	-	-	✓	-	-	-	-	-	-	-	-	-
20	Simple IIC stop condition generation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓
21	Device change function	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
22	Code Generator doesn't support C++ project	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
23	Panel display of code generator node when using multiple projects in e^2 studio	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
24	Use Trace function and Data Flash library together	-	-	-	-	-	✓	-	-	-	-	-	-	-	-
25	Clock setting issue when using the Data Flash Library (FDL)	-	-	-	-	✓	✓	-	-	-	-	-	-	-	-
26	Switch to [Code Generator Console] after building the project	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
27	SAU and TAU GUI can't fully display when creating projects in e^2 studio	✓	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 6-2. List of Points for Caution (2/2)

✓: Applicable, -: Not applicable

No	Description	Group													
		RL78/D1A	RL78/G10	RL78/G11	RL78/G12	RL78/G13, G13A	RL78/G14	RL78/G1A	RL78/G1C	RL78/G1D	RL78/G1E	RL78/G1F	RL78/G1G	RL78/G1H	RL78/H1D
28	Code Generator can't register correctly sometimes in e^2 studio	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
29	WDT reset issue	✓	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 6-3. List of Points for Caution (1/2)

✓: Applicable, -: Not applicable

No	Description	Group													
		RL78/F12	RL78/F13	RL78/F14	RL78/F15	RL78/F1E	RL78/I1A	RL78/I1B	RL78/I1C	RL78/I1D	RL78/I1E	RL78/L12	RL78/L13	RL78/L1A	RL78/L1C
1	Online Help (Applilet3, AP4)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	Coding rule of MISRA-C.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	High-speed on-chip oscillator frequency select register	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
4	Internal low-speed or internal high-speed oscillator trimming	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	Serial array unit	-	-	-	-	-	✓	-	-	-	-	-	-	-	-
6	Flash memory CRC operation function (high-speed CRC)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
7	Port mode select register (PMS)	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓
8	LIN-bus function of UART	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
9	Extension code, multi-master and wakeup function of serial interface IICA and multi-master function of simple IIC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
10	CAN controllers	-	✓	✓	✓	✓	-	-	-	-	-	-	-	-	-
11	Safety Functions	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
12	USB	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
13	RI78V4 project	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	DTC function (CS+ for CA,CX)	-	✓	✓	✓	✓	-	-	-	-	-	-	-	✓	-
15	High Speed DTC chain transfer	-	✓	✓	✓	-	-	-	-	-	-	-	-	-	-
16	Fast Mode Plus setting in IICA slave	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
17	high-speed on-chip oscillator (CS+ for CA,CX)	✓	✓	✓	-	-	✓	-	-	-	-	✓	-	-	-
18	Pin Configurator (CS+ for CA,CX)	✓	✓	✓	✓	-	✓	-	-	-	-	✓	-	-	-
19	Version notation of RL78/G13A generation file.	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	Simple IIC stop condition generation	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	-	✓	✓	✓
21	Device change function	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
22	Code Generator doesn't support C++ project	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
23	Panel display of code generator node when using multiple projects in e^2 studio	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
24	Use Trace function and Data Flash library together	-	✓	✓	✓	-	-	-	-	-	✓	-	-	-	-
25	Clock setting issue when using the Data Flash Library (FDL)	✓	✓	✓	✓	-	-	-	-	-	✓	-	-	-	-
26	Switch to [Code Generator Console] after building the project	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
27	SAU and TAU GUI can't fully display when creating projects in e^2 studio	-	-	-	✓	-	-	-	-	-	-	-	-	-	-

Table 6-4. List of Points for Caution (2/2)

✓: Applicable, -: Not applicable

No	Description	Group													
		RL78/F12	RL78/F13	RL78/F14	RL78/F15	RL78/F1E	RL78/I1A	RL78/I1B	RL78/I1C	RL78/I1D	RL78/I1E	RL78/L12	RL78/L13	RL78/L1A	RL78/L1C
28	Code Generator can't register correctly sometimes in e^2 studio	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
29	WDT reset issue	✓	✓	✓	✓	-	-	-	-	-	-	-	-	-	-

6.2 Details for Caution

6.2.1 Online Help (Applilet3, AP4)

Applilet3 and AP4 do not support online help.

6.2.2 Coding rule of MISRA-C

Compliance with the MISRA-C (Guidelines for the Use of the C Language in Vehicle Based Software) coding convention is not supported for source code output by the code generator.

6.2.3 High-speed on-chip oscillator frequency select register

The code generator is not equivalent to a setup of high-speed on-chip oscillator frequency select register.

6.2.4 Internal low-speed or internal high-speed oscillator trimming

The code generator is not equivalent to a setup of internal low-speed or internal high-speed oscillator trimming register.

6.2.5 Serial array unit

The code generator is not equivalent to a setup of single-wire UART mode and DMX512 communication.

6.2.6 Flash memory CRC operation function (high-speed CRC)

The code generator does not correspond to a flash memory CRC operation function (high-speed CRC). Please refer to application note r01an0736.
https://www.renesas.com/document/apn/rl78g13-safety-function-flash-memory-crc-operation-function

6.2.7 Port mode select register (PMS)

The code generator does not correspond to a port mode select register (PMS).

6.2.8 LIN-bus function of UART

The code generator does not support the LIN-bus functions of serial interface UART0, UART2, UART3, UART6 or UARTF.

6.2.9 Extension code, multi-master and wakeup function of serial interface IICA and multi-master function of simple IIC

The code generator is not supporting the extension code, multi-master and wakeup function of serial interface IICA. It isn't supporting the multi-master function of simple IIC also.

6.2.10 CAN controllers

The code generator is not supporting the CAN Controllers.

6.2.11 Safety Functions

The code generator does not support the USB host, USB function.

6.2.12 USB

The code generator does not support the USB host, USB function.

6.2.13 RI78V4 project

The Code generator can't be used in a project of RI78V4. But code generator is shown to a project of RI78V4.
Even if a code is generated, RI78V4 will be an unsupported purpose build error.

6.2.14 DTC function (CS+ for CA,CX)

When DTC is used, the following warning message is displayed and an object file is not generated.
CC78K0R warning W0837: Output assembler source file, not object file.
[Workaround]
Set up the following individual option of building.

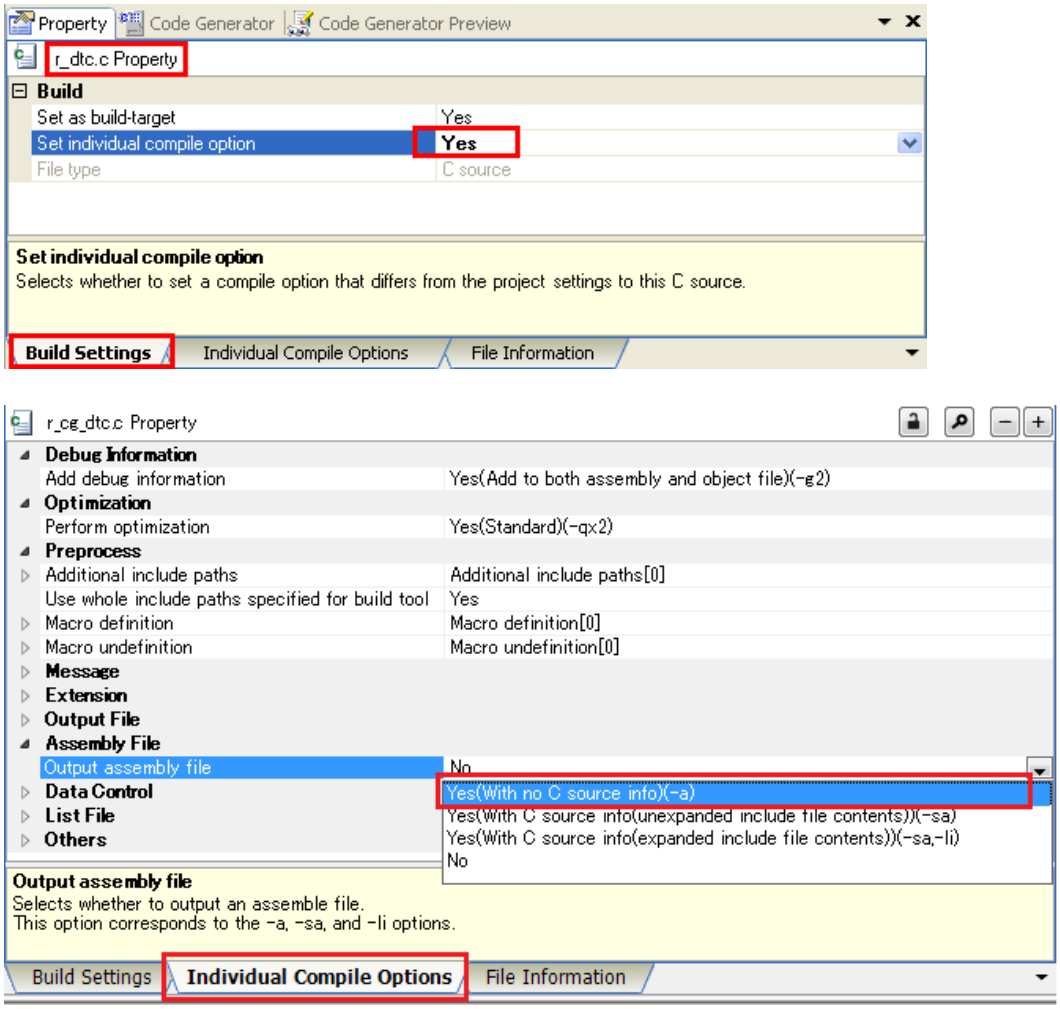


Figure 6-1 CS+ individual option of building

6.2.15 High Speed DTC chain transfer

Although there are chain transfer setting items of High Speed DTC, code corresponding to chain transfer is not supported.

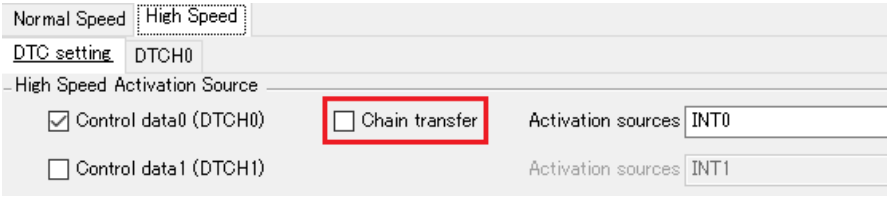


Figure 6-2 High Speed DTC

[Workaround] It cannot be used for chain transfer.

6.2.16 Fast Mode Plus setting in IICA slave

If the Fast Mode Plus is set when using the IICA slave, IICA Low level range setting register (IICWLn, n= channel number), and IICA High level range setting register (IICWHLn) are not set correctly.

[Workaround] There is no workaround.
After doing code generator, please rewrite the numerical value of the register setting of IICWLn, IICWHn in the R_IICAn_Create function. I depend on a system for the numerical value. Please change device UM to reference.

6.2.17 High-speed on-chip oscillator (CS+ for CA,CX)

When a high-speed on-chip oscillator clock is set up by CubeSuite+ RL78, 78K0R, and 78K0 code generator V2.01.00 or earlier, If it is read by CubeSuite+V2.03.00, a clock frequency setup of a high-speed on-chip oscillator may not be right.

[Workaround] Re-set up the frequency right in that case.

6.2.18 Pin Configurator (CS+ for CA,CX)

There is a pin which is not reflected even if it performs reflection to pin configurator from code generator.
Even if it sets up using a code generator PIOR function, it is not reflected to pin configurator.

[Workaround] Edit terminal information with pin configurator.

6.2.19 Version notation of RL78/G13A generation file.

The device name output in the version of the file generated by RL78/G13A is output as “RL78/G13” instead of “RL78/G13A”.

```
/***********************************************************************
* File Name    : r_main.c
* Version      : CodeGenerator for RL78/G13 V2.05.04.02 [20 Nov 2019]
* Device(s)    : R5F140PL
* Tool-Chain   : CCRL
* Description  : This file implements main function.
* Creation Date: 2019/11/27
***********************************************************************/
```

Figure 6-3 RL78/G13A version file

6.2.20 Simple IIC stop condition generation

Using simple IIC (master) to transmit data, after all data are transmitted to the slave, a stop condition should be generated, and the bus is released. Otherwise, subsequent data transmission can't start correctly. However, there is no generation of stop condition in code generator generated code because the code generator can't know when slave process completes.

[Workaround] The user should manually check slave process completion and set stop condition in main() function.

For example:

```
void main(void)
{
    ....
    R_IICA0_Slave_Receive(IICA_rx_buff,5U);
    R_IIC00_Master_Send(16U, IIC0_tx_buff, 5U);
    while(IICA_Receive_flag != 1U);//After IICA0 receive all data, IICA_Receive_flag will become 1.
    R_IIC00_StopCondition();//Trigger stop condition signal.
    ....
}
```

Figure 6-4 The code in main()

6.2.21 Device change function

The Code Generator can only support device change function within same group and same pin-count devices. If the devices selected before and after change are in the same group and the number of pins is the same, the Code Generator can be changed with current setting successfully.
Otherwise, after change, Code Generator setting is restored to the initial state.

6.2.22 Code Generator doesn't support C++ project

The Code Generator doesn't support C++ project. So, please avoid using Code Generator in C++ project. Otherwise, maybe you can see a build error when C++ project build.

6.2.23 Panel display of code generator node when using multiple projects in e^2 studio

For the case there are multiple projects in e^2 studio project tree, after reopening one of "Peripheral Functions", "Code Preview", "Device Top View" and "Device List View" panels, the contents on the panel are uncertain.

[Workaround]
Double click any node under [Code Generator], then all these panels content will be refreshed to match the selected project.

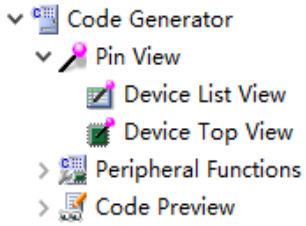


Figure 6-5 [Code Generator] node

6.2.24 Use Trace function and Data Flash library together

If Trace function is used, the user can't select data flash library function. In fact, RAM area used by data flash library and Trace function are not overlapped. They can be used at the same time.

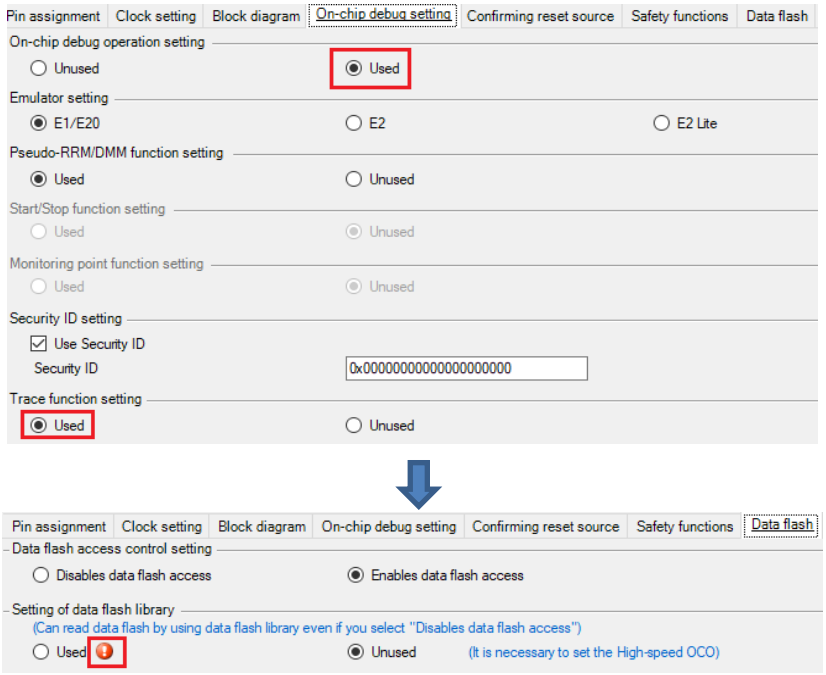


Figure 6-6 Trace and Data Flash library can't work together in Code Generator

[Workaround for IAR project]

1. Create a project using IAR toolchain.
2. In [On-chip debug setting] page of [Clock Generator], select Trace function as used.
3. Click [Generate Code].
4. In the code generation path, find and open "r_main.c" file.
5. Manually change the file: move the secured trace RAM code into user code area, so that this code can be reserved in next time code generation.

```
/* Start user code for global. Do not edit comment generated here */
/* Secure trace RAM area */
__no_init __root uint8_t ocdtraceram[1024] @ 0xF4300;
/* End user code. Do not edit comment generated here */

/* Set option bytes */
#pragma location = "OPTBYTE"
__root const uint8_t opbyte0 = 0xFFU;
#pragma location = "OPTBYTE"
__root const uint8_t opbyte1 = 0xFFU;
#pragma location = "OPTBYTE"
__root const uint8_t opbyte2 = 0xF8U;
#pragma location = "OPTBYTE"
__root const uint8_t opbyte3 = 0x84U;

/* Set security ID */
#pragma location = "SECUID"
__root const uint8_t secuid[10] =
    {0x00U, 0x00U, 0x00U, 0x00U, 0x00U, 0x00U, 0x00U, 0x00U, 0x00U, 0x00U};
/* Secure trace RAM area */
__no_init __root uint8_t ocdtraceram[1024] @ 0xF4300;
```

Figure 6-7 Move the secured trace RAM code into user code area

6. In [On-chip debug setting] page of [Clock Generator], select Trace function as unused, then select Data flash library as used in [Data flash] page of [Clock Generator].

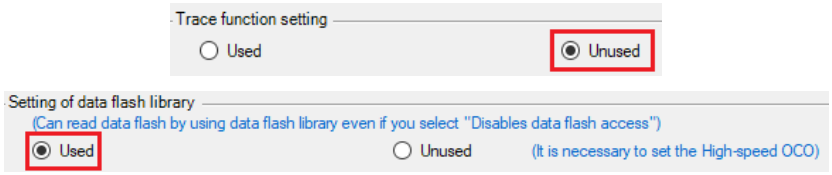


Figure 6-8 Unuse Trace and use Data Flash library in Code Generator

7. Click [Generate Code] again.

By this time, the generated code can secure the Trace RAM correctly and use the data flash library together.

[Workaround for CS+ CC-RL project]

1. Create a project in CS+.
2. In [On-chip debug setting] page of [Clock Generator], select Trace function as unused, then select Data flash library as used in [Data flash] page of [Clock Generator] (refer to Figure 6-8).
3. Click [Generate Code]
4. In [CC-RL property] > [Link Options], select [Yes(Warning message)(-OCDTRW)]. Although there is warning message output, the project can build and run successfully.

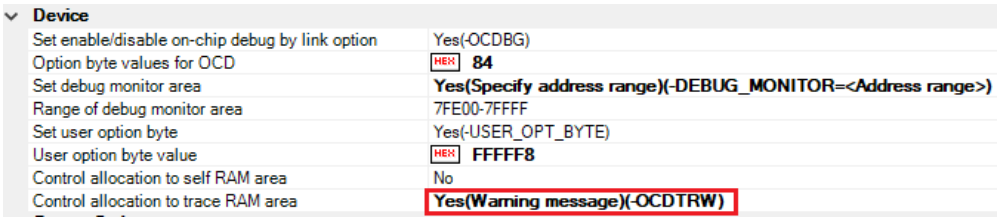


Figure 6-9 Select [Yes(Warning message)(-OCDTRW)] in CS+ [CCRL property]

6.2.25 Clock setting issue when using the Data Flash Library (FDL)

When [Main system clock (fMAIN) setting] in [Clock setting] is [High-speed system clock (fMX)], and [High-speed OCO clock setting] and [CPU and peripheral clock setting] are set to different frequencies, the data flash library might not operate properly.

About the workaround, please refer to the document numbered R20TS0926 of RENESAS TOOL NEWS.

6.2.26 Switch to [Code Generator Console] after building the project

When the user double-clicks the Textbox, a message about the valid input value range should output in the console. However, the user can't see this message after building the project.

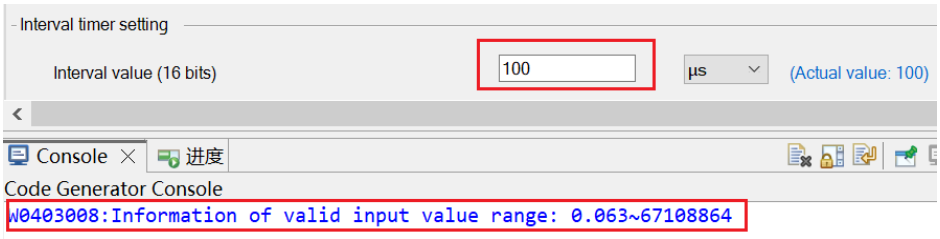


Figure 6-10 The message about the valid input value range

[Workaround]

Set "Code Generator Console" on top manually.

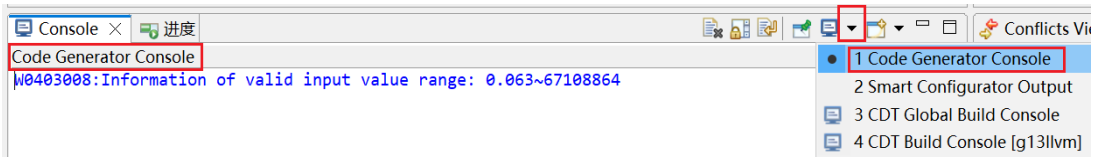


Figure 6-11 Set "Code Generator Console" manually

6.2.27 SAU and TAU GUI can't fully display when creating projects in e² studio

When the user opens SAU or TAU, it may not have a scroll bar and some controls can't display.

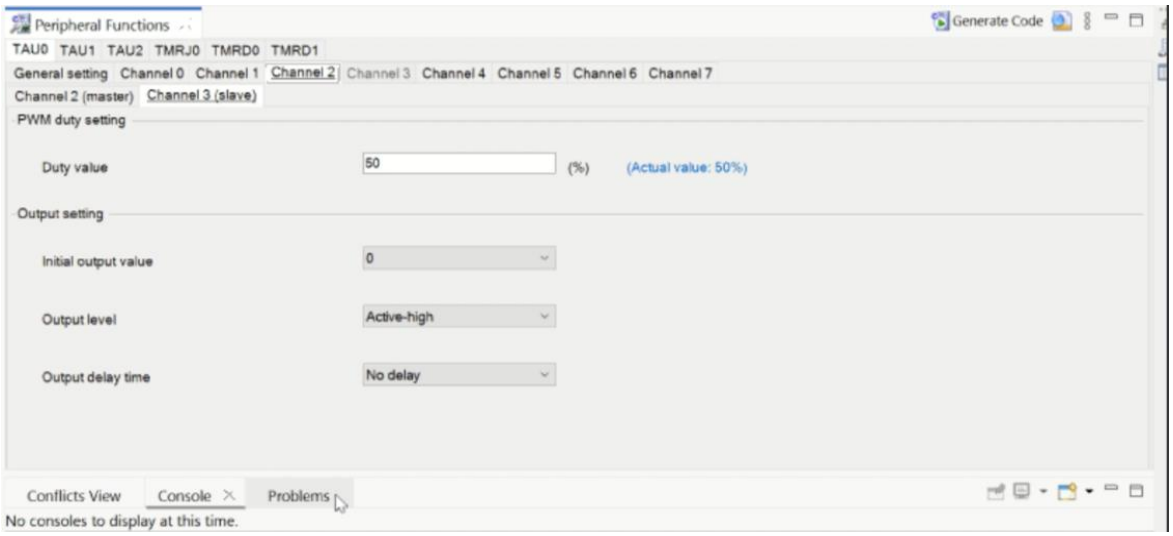


Figure 6-12 The scroll bar disappears

[Workaround]

Click to other peripherals (for example: from TAU0 to TAU1) then go back, the scroll bar will exist and all controls can display.

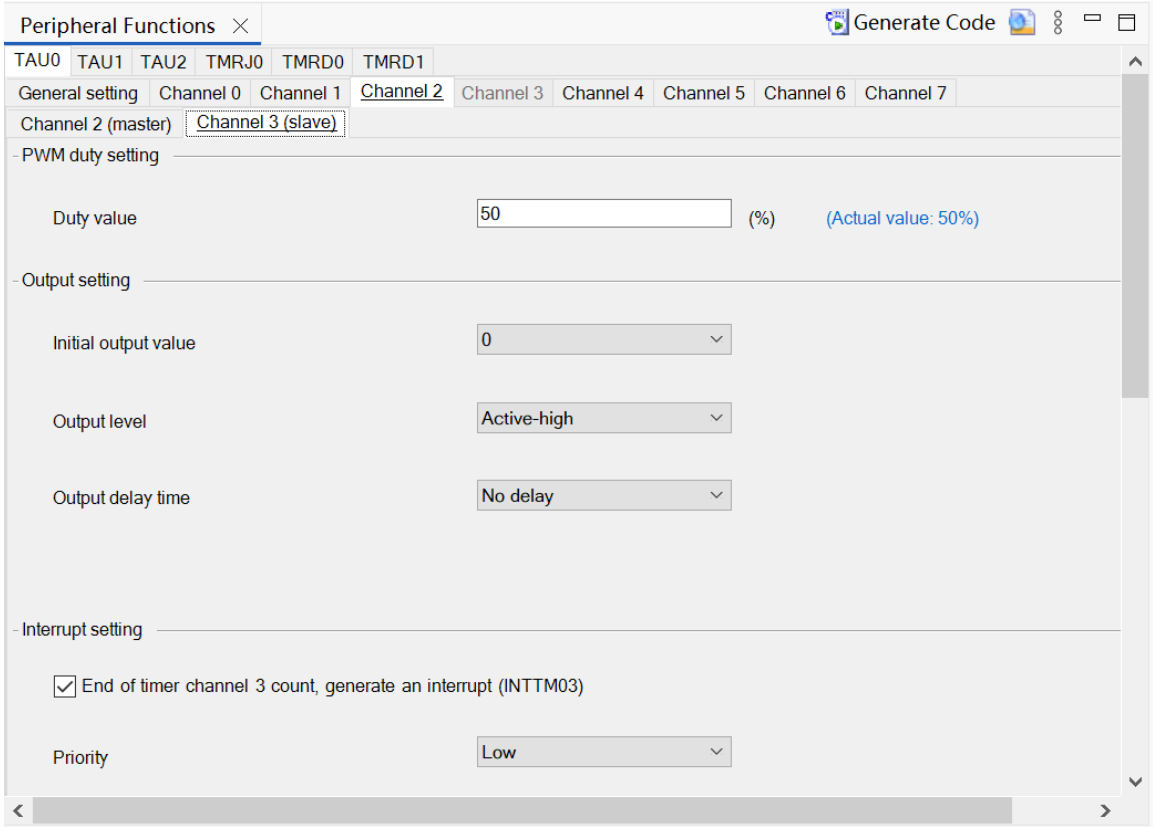


Figure 6-13 The scroll bar exists

6.2.28 Code Generator can't register correctly sometimes in e² studio

When user selects 'Use Peripheral Code Generator' during creating project, there may be no Code Generator node in project tree seldom.

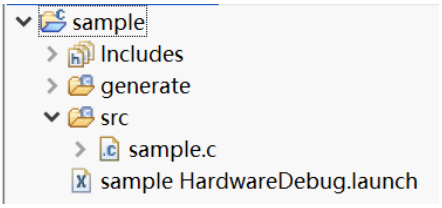


Figure 6-14 No Code Generator node

[Workaround]
Step1: Close e² studio.
Step2: Open e2studio.ini and write "-Dcom.renesas.cg.comMachineRegister" into it.

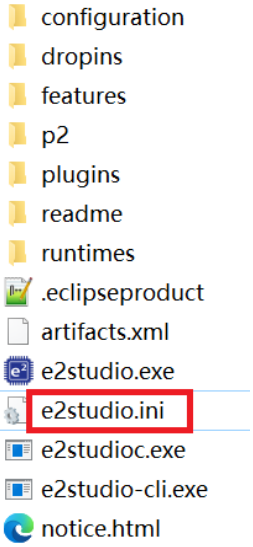


Figure 6-15 Open e2studio.ini

```
--launcher.defaultAction
openFile
-startup
plugins/org.eclipse.equinox.launcher_1.7.0.v20250717-1148.jar
--launcher.library
plugins/org.eclipse.equinox.launcher.win32.win32.x86_64_1.2.1300.v20250331-1702
-vm
plugins/org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916/jre/bin
-vmargs
-XX:CompileCommand=exclude org.eclipse.jdt.internal.core.dom.rewrite.ASTRewriteAnalyzer::getExtendedRange
-Declipse.p2.max.threads=10
-Dosgi.requiredJavaVersion=21
-Dosgi.dataAreaRequiresExplicitInit=true
-Dorg.eclipse.swt.graphics.Resource.reportNonDisposed=true
-Declipse.e4.inject.javax.warning=false
-Dorg.slf4j.simpleLogger.defaultLogLevel=off
-XX:+UseG1GC
-XX:+UseStringDeduplication
-Dsun.java.command=E2Studio
-Xms512m
-Xmx3g
--add-modules=ALL-SYSTEM
-Djava.security.manager=allow
-Dswt.autoScale=quarter
-Dswt.autoScale.method=smooth
-Dcom.renesas.cg.comMachineRegister
```

Figure 6-16 Write "-Dcom.renesas.cg.comMachineRegister" into e2studio.ini

Step3: Double click CGMRegister.exe in installer path.

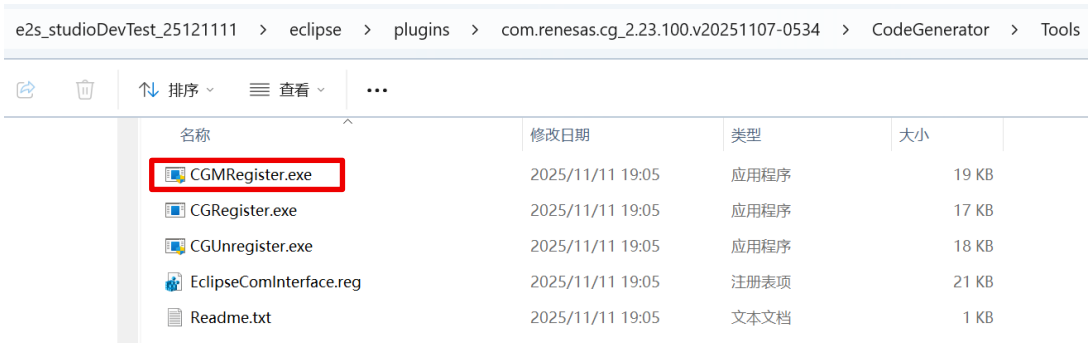


Figure 6-17 Run CGMRegister.exe

Step4: Re-start e^2 studio.

6.2.29 WDT reset issue

Wrong WDT-RESET may assert when WDT counter is cleared (writing 0xAC to WDTC register) at just 50% of count-overflow-value and when window open period of watchdog timer is set to 75%. Please refer to the document TN-RL*-A068A/E for details.

Revision History

Rev.	Section	Description
1.00	-	First edition issued
1.01	5.Restrictions	Add 5.2.18 Incorrect content of R_RTC_Get_CounterValue() and R_RTC_Set_CounterValue() function
		Add 5.2.19 The information of interval time controller is wrong in WDT
	6.Points for Caution	Add 6.2.29 WDT reset issue
1.02	5.Restrictions	Add 5.2.20 Overflow time can’t load correctly in WDT

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

Trademarks

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

Contact information

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

© 2026 Renesas Electronics Corporation. All rights reserved.