

## Contents

Chapter 1. Target Devices .....	3
Chapter 2. User's Manuals .....	11
Chapter 3. Key Points for Selecting Uninstallation Method.....	12
Chapter 4. Changes .....	13
4.1 About Details of Changes.....	14
4.1.1 Output code changes of real-time clock.....	14
4.1.2 Output code changes of serial array unit .....	15
4.1.3 Addition of PMC register setup .....	17
4.1.4 Change of output source file name .....	18
4.1.5 Output code changes of the receiving function of UARTn.....	19
4.1.6 Control correction of a competition pin .....	19
4.1.7 GUI correction of an A/D converter .....	19
This issue has been corrected in Code Generator for RL78,78K0R,78K0 V1.00.03 .....	19
4.1.8 Conversion time setup of A/D correction .....	20
Chapter 5. Cautions.....	21
5.1 Cautions List.....	21
5.2 Cautions Details .....	22
5.2.1 About the LIN-bus function of UART2 or UART3 or UART6 .....	22
5.2.2 About the operation for slave transmission of serial interface IICA or IIC0 .....	22
5.2.3 About extension code, multimaster,wakeup function of serial interface IICA or IIC0 .....	22
5.2.4 About cooperation with the linker option .....	23
5.2.5 About CAN controllers.....	23
5.2.6 About PORT .....	23
5.2.7 About the SNOOZE mode of Serial array unit 1 .....	23
5.2.8 About a setup of a real-time clock.....	23
5.2.9 Cautions when using a DTC function.....	24
5.2.10 About initial function of an A/D converter .....	25
5.2.11 About initial function at the time of setting up UART transmission .....	25
5.2.12 Conversion time setup of A/D correction .....	25
5.2.13 About Complementary assistant PWM mode of TimerRD.....	25
Chapter 6. Restrictions.....	26
6.1 Restrictions List .....	26
6.2 Restrictions Details.....	26
6.2.1 About the coding rule of MISRA-C .....	26
6.2.2 About High-speed on-chip oscillator frequency select register .....	26
6.2.3 About internal low-speed or internal high-speed oscillator trimming register .....	26
6.2.4 About restriction of a serial array unit .....	26

Chapter 7. Changes in User's Manual ..... 27

- 7.1 Modifications in RL78 design ..... 27
  - 7.1.1 Changed description relating to the features of Code generating function..... 27
- 7.2 Modifications in 78K0R design..... 27
  - 7.2.1 Changed description relating to the features of Code generating function..... 27
- 7.3 Modifications in 78K0 design ..... 27
  - 7.3.1 Changed description relating to the features of Code generating function..... 27
- 7.4 About API added and changed by V1.00.02..... 28
  - 7.4.1 The API table changed or added by RL78 Code Generator function ..... 28
  - 7.4.2 Details of API added by RL78 Code Generator function ..... 31

## Chapter 1. Target Devices

Below is a list of devices supported by the Code Generator for RL78/I1A V1.01.01	
PIN	Device name
20pin	R5F1076C
30pin	R5F107AC, R5F107AE
32pin	R5F107BC
38pin	R5F107DE
The Code Generator for RL78/I1A V1.01.01 is based on the following documents.	
Manual Name	Document Number
RL78/I1A Preliminary User's Manual: Hardware	R01UH0169JJ0001 Rev.0.01
	R01UH0169EJ0001 Rev.0.01

Below is a list of devices supported by the Code Generator for RL78/G12 V1.02.01	
PIN	Device name
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
The Code Generator for RL78/G12 V1.02.01 is based on the following documents.	
Manual Name	Document Number
RL78/G12 Preliminary User's Manual: Hardware	R01UH0200JJ0002 Rev.0.02
	R01UH0200EJ0002 Rev.0.02

Below is a list of devices supported by the Code Generator for RL78/G13 V1.03.01	
PIN	Device name
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
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
The Code Generator for RL78/G13 V1.03.01 is based on the following documents.	
Manual Name	Document Number
RL78/G13 Preliminary User's Manual: Hardware	R01UH0146JJ0006 Rev.0.06
	R01UH0146EJ0006 Rev.0.06

Below is a list of devices supported by the Code Generator for RL78/G14 V1.01.01	
PIN	Device name
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
52pin	R5F104JC, R5F104JD, R5F104JE, R5F104JF, R5F104JG, R5F104JH, R5F104JJ
64pin	R5F104LC, R5F104LD, R5F104LE, R5F104LF, R5F104LG, R5F104LH, R5F104LJ
80pin	R5F104MF, R5F104MG, R5F104MH, R5F104MJ
100pin	R5F104PF, R5F104PG, R5F104PH, R5F104PJ
The Code Generator for RL78/G14 V1.01.01 is based on the following documents.	
Manual Name	Document Number
RL78/G14 Preliminary User's Manual: Hardware	R01UH0186JJ0001 Rev.0.01
	R01UH0186EJ0001 Rev.0.01

Below is a list of devices supported by the Code Generator for RL78/G1A V1.00.01	
PIN	Device name
25pin	R5F10E8A, R5F10E8C, R5F10E8D, R5F10E8E
32pin	R5F10EBA, R5F10EBC, R5F10EBD, R5F10EBE
48pin	R5F10EGA, R5F10EGC, R5F10EGD, R5F10EGE
64pin	R5F10ELC, R5F10ELD, R5F10ELE
The Code Generator for RL78/G1A V1.00.01 is based on the following documents.	
Manual Name	Document Number
RL78/G1A User's Manual: Hardware	R01UH0305JJ0002 Rev.0.02
	R01UH0305EJ0002 Rev.0.02

Below is a list of devices supported by the Code Generator for RL78/F12 V1.00.00	
PIN	Device name
20pin	R5F109AE, R5F109AD, R5F109AC, R5F109AB, R5F109AA
30pin	R5F109BE, R5F109BD, R5F109BC, R5F109BB, R5F109BA
32pin	R5F109BE, R5F109BD, R5F109BC, R5F109BB, R5F109BA
48pin	R5F109GE, R5F109GD, R5F109GC, R5F109GB, R5F109GA
64pin	R5F109LE, R5F109LD, R5F109LC, R5F109LB, R5F109LA
The Code Generator for RL78/F12 V1.00.00 is based on the following documents.	
Manual Name	Document Number
RL78/F12 User's Manual: Hardware	R01UH0231JJ0003 Rev.0.03
	R01UH0231EJ0003 Rev.0.03

Below is a list of devices supported by the Code Generator for 78K0R/Fx3 V1.00.01	
Nickname	Device name
78K0R/FB3	μPD78F1804, μPD78F1805, μPD78F1806, μPD78F1807
78K0R/FC3	μPD78F1808, μPD78F1809, μPD78F1810, μPD78F1811 μPD78F1812, μPD78F1813, μPD78F1814, μPD78F1815, μPD78F1816, μPD78F1817 μPD78F1826, μPD78F1827, μPD78F1828, μPD78F1829, μPD78F1830
78K0R/FE3	μPD78F1818, μPD78F1819, μPD78F1820, μPD78F1821, μPD78F1822 μPD78F1831, μPD78F1832, μPD78F1833, μPD78F1834, μPD78F1835
78K0R/FF3	μPD78F1823, μPD78F1824, μPD78F1825 μPD78F1836, μPD78F1837, μPD78F1838, μPD78F1839, μPD78F1840
78K0R/FG3	μPD78F1841, μPD78F1842, μPD78F1843, μPD78F1844, μPD78F1845
The Code Generator for 78K0R/Fx3 V1.00.01 is based on the following documents.	
Manual Name	Document Number
78K0R/Fx3 User's Manual	U19145JJ1V0UD00
	U19145EJ1V0UD00

Below is a list of devices supported by the Code Generator for 78K0R/lx3 V1.00.01	
Nickname	Device name
78K0R/IB3	μPD78F1201, μPD78F1203
78K0R/IC3	μPD78F1211(38pin), μPD78F1213(38pin), μPD78F1211(44pin), μPD78F1213(44pin)
78K0R/ID3	μPD78F1213(48pin), μPD78F1214(48pin), μPD78F1215(48pin) μPD78F1223, μPD78F1224,μPD78F1225
78K0R/IE3	μPD78F1233, μPD78F1234,μPD78F1235
The Code Generator for 78K0R/lx3 V1.00.01 is based on the following documents.	
Manual Name	Document Number
78K0R/lx3 User's Manual	U19678JJ1V1UD00
	U19678EJ1V1UD00

Below is a list of devices supported by the Code Generator for 78K0R/Kx3 V1.00.01	
Nickname	Device name
78K0R/KE3	μPD78F1142/A, μPD78F1143/A, μPD78F1144/A, μPD78F1145/A, μPD78F1146/A
78K0R/KF3	μPD78F1152/A, μPD78F1153/A, μPD78F1154/A, μPD78F1155/A, μPD78F1156/A
78K0R/KG3	μPD78F1162/A, μPD78F1163/A, μPD78F1164/A, μPD78F1165/A, μPD78F1166/A, μPD78F1167/A, μPD78F1168/A
78K0R/KH3	μPD78F1174/A, μPD78F1175/A, μPD78F1176/A, μPD78F1177/A, μPD78F1178/A
78K0R/KJ3	μPD78F1184A, μPD78F1185A, μPD78F1186A, μPD78F1187A, μPD78F1188A
The Code Generator for 78K0R/Kx3 V1.00.01 is based on the following documents	
Manual Name	Document Number
78K0R/KE3 User's Manual	U17854JJ8V0UD00
	U17854EJ8V0UD00
78K0R/KF3 User's Manual	U17893JJ7V0UD00
	U17893EJ7V0UD00
78K0R/KG3 User's Manual	U17894JJ8V0UD00
	U17894EJ8V0UD00
78K0R/KH3 User's Manual	U18432JJ4V0UD00
	U18432EJ4V0UD00
78K0R/KJ3 User's Manual	U18417JJ3V0UD00
	U18417EJ3V0UD00

Below is a list of devices supported by the Code Generator for 78K0R/Kx3-A V1.00.01	
Nickname	Device name
78K0R/KE3-A	μPD78F1016, μPD78F1017, μPD78F1018
The Code Generator for 78K0R/Kx3-A V1.00.01 is based on the following documents	
Manual Name	Document Number
78K0R/Kx3-A User's Manual	U19653JJ1V0UD
	U19653EJ1V0UD



Below is a list of devices supported by the Code Generator for 78K0R/Kx3-L V1.00.01	
Nickname	Device name
78K0R/KC3-L	μPD78F1000(44pin), μPD78F1001(44pin), μPD78F1002(44pin),μPD78F1003(44pin), μPD78F1001(48pin), μPD78F1002(48pin), μPD78F1003(48pin)
78K0R/KD3-L	μPD78F1004, μPD78F1005, μPD78F1006
78K0R/KE3-L	μPD78F1007, μPD78F1008, μPD78F1009
78K0R/KF3-L	μPD78F1010, μPD78F1011, μPD78F1012
78K0R/KG3-L	μPD78F1013, μPD78F1014
The Code Generator for 78K0R/Kx3-L V1.00.01 is based on the following documents	
Manual Name	Document Number
78K0R/Kx3-L User's Manual	U19291JJ2V0UD00
	U19291EJ3V0UD00
78K0R/KF3-L User's Manual	U19459JJ1V0UD00
	U19459EJ1V0UD00
78K0R/KG3-L User's Manual	U19460JJ1V0UD00
	U19460EJ1V0UD00

Below is a list of devices supported by the Code Generator for 78K0R/Lx3 V1.00.01	
Nickname	Device name
78K0R/LF3	μPD78F1500, μPD78F1501, μPD78F1502
78K0R/LG3	μPD78F1503, μPD78F1504, μPD78F1505
78K0R/LH3	μPD78F1506, μPD78F1507, μPD78F1508
The Code Generator for 78K0R/Lx3 V1.00.01 is based on the following documents	
Manual Name	Document Number
78K0R/Lx3 User's Manual	U19155JJ3V0UD
	U19155EJ3V0UD

Below is a list of devices supported by the Code Generator for 78K0/lx2 V1.00.00	
Nickname	Device name
78K0/IY2	μPD78F0740, μPD78F0741, μPD78F0742, μPD78F0750, μPD78F0751, μPD78F0752
78K0/IA2	μPD78F0743, μPD78F0744, μPD78F0753, μPD78F0754
78K0/IB2	μPD78F0745, μPD78F0746, μPD78F0755, μPD78F0756
The Code Generator for 78K0/lx2 V1.00.00 is based on the following documents.	
Manual Name	Document Number
78K0/lx2 User's Manual	U19353JJ3V0UD00
	U19353EJ3V0UD00

Below is a list of devices supported by the Code Generator for 78K0/Kx2-L V1.00.00	
Nickname	Device name
78K0/KY2-L	μPD78F0550, μPD78F0551, μPD78F0552, μPD78F0555, μPD78F0556, μPD78F0557
78K0/KA2-L	μPD78F0560, μPD78F0561, μPD78F0562, μPD78F0565, μPD78F0566, μPD78F0567
78K0/KB2-L	μPD78F0571, μPD78F0572, μPD78F0573, μPD78F0576, μPD78F0577, μPD78F0578
78K0/KC2-L	μPD78F0581(44pin), μPD78F0582(44pin), μPD78F0583(44pin), μPD78F0581(48pin), μPD78F0582(48pin), μPD78F0583(48pin), μPD78F0586(44pin), μPD78F0587(44pin), μPD78F0588(44pin), μPD78F0586(48pin), μPD78F0587(48pin), μPD78F0588(48pin)
The Code Generator for 78K0/Kx2-L V1.00.00 is based on the following documents.	
Manual Name	Document Number
78K0/Kx2-L User's Manual	U19111JJ2V1UD
	U19111EJ2V1UD

## Chapter 2. User's Manuals

Please read the following user's manuals together with this document.

Manual Name	Document Number
CubeSuite+ V1.01.00 RL78 Design	R20UT0728EJ0100
CubeSuite+ V1.00.00 78K0R Design	R20UT0547EJ0100
CubeSuite+ V1.00.00 78K0 Design	R20UT0546EJ0100
CubeSuite+ V1.01.00 Message	R20UT0736EJ0100

## Chapter 3. Key Points for Selecting Uninstallation Method

There are two ways to uninstall this product.

- Use the integrated uninstaller (uninstalls CubeSuite+)
- Use separate uninstaller (uninstalls this product only)

To use the separate uninstaller, select the following from the Control Panel:

- Add/Remove Programs (Windows XP)
- Programs and Features (Windows Vista, Windows 7)

Then select "CubeSuite+ Code Generator for RL78\_78K".

## Chapter 4. Changes

This chapter describes change from V1.00.04 to V1.00.05

No	内容	Corresponds of code generation													
		RL78/F12	RL78/G1A	RL78/G12	RL78/G13	RL78/G14	RL78/I1A	78K0R/Fx3	78K0R/lx3	78K0R/Kx3	78K0R/Kx3-A	78K0R/Kx3-L	78K0R/Lx3	78K0/lx2	78K0/Kx2-L
		V1.00.00	V1.00.01	V1.02.01	V1.03.01	V1.01.01	V1.01.01	V1.00.01	V1.00.01	V1.00.01	V1.00.01	V1.00.01	V1.00.01	V1.00.00	V1.00.00
1	Output code changes of real-time clock	/	/	-	-	-	-	/	/	/	/	/	/	/	/
2	Output code changes of serial array unit	/	/	-	-	-	-	/	/	/	/	/	/	/	/
3	Addition of PMC register setup	/	/	-	-	-	-	/	/	/	/	/	/	/	/
4	Change of output source file name	/	/	-	-	-	-	/	/	/	/	/	/	/	/
5	Output code changes of the receiving function of UARTn	/	/	-	-	-	-	/	/	/	/	/	/	/	/
6	Control correction of a competition pin	/	/	/	-	/	/	/	/	/	/	/	/	/	/
7	GUI correction of an A/D converter	/	/	-	-	-	/	/	/	/	/	/	/	/	/
8	Conversion time setup of A/D correction	/	○	/	/	/	/	/	/	/	/	/	/	/	/

○ : Correspondence, -: Not correspondence(finish of correction), /: Outside of function

The following MCUs of the RL78 family have been supported:

- RL78/F12 and RL78/G1A groups

## 4.1 About Details of Changes

### 4.1.1 Output code changes of real-time clock

a) The output code of R\_RTC\_Set\_ConstPeriodInterruptOff() was changes.

**Before:**

```
void R_RTC_Set_ConstPeriodInterruptOff(void)
{
    RTCC0 &= _88_RTC_INTRTC_CLEAR;
    RTCIF = 0U;          /* clear INTRTC interrupt flag */
}
```

**After:**

```
void R_RTC_Set_ConstPeriodInterruptOff(void)
{
    RTCC0 &= _F8_RTC_INTRTC_CLEAR;
    RTCC1 &= (uint8_t)~_08_RTC_INTC_GENERATE_FLAG;
    RTCIF = 0U;          /* clear INTRTC interrupt flag */
}
```

b) The output code of R\_RTC\_Interrupt( ) was changes.

- When an alarm interrupt function is checked.

**Before:**

```
__interrupt void R_RTC_Interrupt(void)
{
    R_RTC_Callback_Alarm();
}
```

**After:**

```
__interrupt static void r_rtc_interrupt(void)
{
    if (1U == WAFG)
    {
        RTCC1 &= (uint8_t)~_10_RTC_ALARM_MATCH;    /* clear WAFG */
        r_rtc_callback_alarm();
    }
}
```

- When a constant-period interruption function is checked

**Before:**

```
__interrupt void R_RTC_Interrupt(void)
{
    R_RTC_Callback_ConstPeriod();
}
```

**After:**

```
__interrupt static void r_rtc_interrupt(void)
{
    if (1U == RIFG)
    {
        RTCC1 &= (uint8_t)~_08_RTC_INTC_GENERATE_FLAG; /* clear RIFG */
        r_rtc_callback_constperiod();
    }
}
```

a) and b) issues has been corrected in Code Generator for RL78,78K0R,78K0 V1.00.02.

## 4.1.2 Output code changes of serial array unit

- a) The following function which was not supported by SAU1 was added.

```
void R_SAU0_Set_SnoozeOn(void)
```

```
void R_SAU0_Set_SnoozeOff(void)
```

- b) The function of simple IIC was changed.

**Before:**

```
void R_IIC00_StartCondition(void)
{
    SO0 &= ~_0001_SAU_CH0_DATA_OUTPUT_1; /* clear IIC00 SDA */
    SOE0 |= _0001_SAU_CH0_OUTPUT_ENABLE; /* enable IIC00 output */
    SO0 &= ~_0100_SAU_CH0_CLOCK_OUTPUT_1; /* clear IIC00 SCL */
    SS0 |= _0001_SAU_CH0_START_TRG_ON; /* enable IIC00 */
}
```

**After:**

```
void R_IIC00_StartCondition(void)
{
    volatile uint8_t w_count;

    SO0 &= ~_0001_SAU_CH0_DATA_OUTPUT_1; /* clear IIC00 SDA */

    /* Wait for 5us */
    for (w_count = 0U; w_count <= IIC00_WAITTIME; w_count++)
    {
        NOP();
    }
    SO0 &= ~_0100_SAU_CH0_CLOCK_OUTPUT_1; /* clear IIC00 SCL */
    SOE0 |= _0001_SAU_CH0_OUTPUT_ENABLE; /* enable IIC00 output */
    SS0 |= _0001_SAU_CH0_START_TRG_ON; /* enable IIC00 */
}
```

**Before:**

```
void R_IIC00_StopCondition(void)
{
    ST0 |= _0001_SAU_CH0_STOP_TRG_ON; /* disable IIC00 */
    SOE0 &= ~_0001_SAU_CH0_OUTPUT_ENABLE; /* disable IIC00 output */
    SO0 &= ~_0001_SAU_CH0_DATA_OUTPUT_1; /* clear IIC00 SDA */
    SO0 |= _0100_SAU_CH0_CLOCK_OUTPUT_1; /* set IIC00 SCL */
    SO0 |= _0001_SAU_CH0_DATA_OUTPUT_1; /* set IIC00 SDA */
}
```

**After:**

```
void R_IIC00_StopCondition(void)
{
    volatile uint8_t w_count;

    ST0 |= _0001_SAU_CH0_STOP_TRG_ON; /* disable IIC00 */
    SOE0 &= ~_0001_SAU_CH0_OUTPUT_ENABLE; /* disable IIC00 output */
    SO0 &= ~_0001_SAU_CH0_DATA_OUTPUT_1; /* clear IIC00 SDA */
    SO0 |= _0100_SAU_CH0_CLOCK_OUTPUT_1; /* set IIC00 SCL */

    /* Wait for 5us */
    for (w_count = 0U; w_count <= IIC00_WAITTIME; w_count++)
    {
        NOP();
    }
    SO0 |= _0001_SAU_CH0_DATA_OUTPUT_1; /* set IIC00 SDA */
}
```

c) The interrupt handler function of simple IIC was corrected.

- It was made not to take out an error with the last byte's NACK.

**Before:**

```
if ((SSR00 & _0002_SAU_PARITY_ERROR) == 0x0002U)
{
    R_IIC00_Callback_Master_Error(MD_NACK);
}
```

**After:**

```
if (((SSR00 & _0002_SAU_PARITY_ERROR) == 0x0002U) && (g_iic00_tx_count != 0U))
{
    r_iic00_callback_master_error(MD_NACK);
}
```

- Deletion of an unnecessary code

**Before:**

```
if ((g_lic00MasterStatusFlag & _04_SAU_IIC_SENDED_ADDRESS_FLAG) == 0U)
{
    rxadr = SIO00;
    SCR00 &= ~_C000_SAU_RECEPTION_TRANSMISSION;
    SCR00 |= _4000_SAU_RECEPTION;
    g_lic00MasterStatusFlag |= _04_SAU_IIC_SENDED_ADDRESS_FLAG;
    SIO00 = 0xFFU;
}
```

**After:**

```
if ((g_iic00_master_status_flag & _04_SAU_IIC_SENDED_ADDRESS_FLAG) == 0U)
{
    ST0 |= _0001_SAU_CH0_STOP_TRG_ON;
    SCR00 &= ~_C000_SAU_RECEPTION_TRANSMISSION;
    SCR00 |= _4000_SAU_RECEPTION;
    SS0 |= _0001_SAU_CH0_START_TRG_ON;
    g_iic00_master_status_flag |= _04_SAU_IIC_SENDED_ADDRESS_FLAG;
    SIO00 = 0xFFU;
}
```

a) , b) and c) issues has been corrected in Code Generator for RL78,78K0R,78K0 V1.00.02.



### 4.1.3 Addition of PMC register setup

A setup of the PMC register was added about the combination terminal of each circumference.

[The terminal to which a PMC register setup was added]

- 20,24,25,30, 32pin devices

P00/ANI17/TI00/TxD1

P01/ANI16/TO00/RxD1

- other devices

P02/ANI17/SO10/TxD1

P03/ANI16/SI10/RxD1/SDA10

This issue has been corrected in Code Generator for RL78,78K0R,78K0 V1.00.02

## 4.1.4 Change of output source file name

The filename outputted from Code Generator RL78/G13 V1.00.02.

Peripheral Function	Filename	
	Before	After
Clock Generator	r_cg.c	r_cg_cg.c
	r_cg_user.c	r_cg_cg_user.c
Port	r_port.c	r_cg_port.c
	r_port_user.c	r_cg_port_user.c
Interrupt	r_intc.c	r_cg_intc.c
	r_intc_user.c	r_cg_intc_user.c
Serial	r_serial.c	r_cg_serial.c
	r_serial_user.c	r_cg_serial_user.c
A/D Converter	r_adc.c	r_cg_adc.c
	r_adc_user.c	r_cg_adc_user.c
D/A Converter	r_dac.c	r_cg_dac.c
	r_dac_user.c	r_cg_dac_user.c
Timer	r_timer.c	r_cg_timer.c
	r_timer_user.c	r_cg_timer_user.c
Watchdog Timer	r_wdt.c	r_cg_wdt.c
	r_wdt_user.c	r_cg_wdt_user.c
Real-time Clock	r_rtc.c	r_cg_rtc.c
	r_rtc_user.c	r_cg_rtc_user.c
Interval Timer	r_it.c	r_cg_it.c
	r_it_user.c	r_cg_it_user.c
Clock Output	r_pclbuz.c	r_cg_pclbuz.c
	r_pclbuz_user.c	r_cg_pclbuz_user.c
DMA Controller	r_dmac.c	r_cg_dmac.c
	r_dmac_user.c	r_cg_dmac_user.c
Voltage Detector	r_lvd.c	r_cg_lvd.c
	r_lvd_user.c	r_cg_lvd_user.c

#### 4.1.5 Output code changes of the receiving function of UARTn

The output code of the receiving function of UARTn was corrected. The following is a case of UART0.

```
[ r_cg_serial.c ]
MD_STATUS R_UART0_Receive(uint8_t * const rx_buf, uint16_t rx_num)
{
    MD_STATUS status = MD_OK;

    if (rx_num < 1U)
    {
        status = MD_ARGERROR;
    }
    else
    {
        g_uart0_rx_count = 0U;
        g_uart0_rx_length = rx_num;
        gp_uart0_tx_address = rx_buf;
    }

    return (status);
}
```

**Before :** gp\_uart0\_tx\_address = rx\_buf;

**After :** gp\_uart0\_rx\_address = rx\_buf;

This issue has been corrected in Code Generator for RL78,78K0R,78K0 V1.00.03

#### 4.1.6 Control correction of a competition pin

Control of the competition pin when simple-I2C of 24 and 25 pin device of RL78/G13 is set up was corrected.

**Before :** P17/SDA11

P30/SCL11

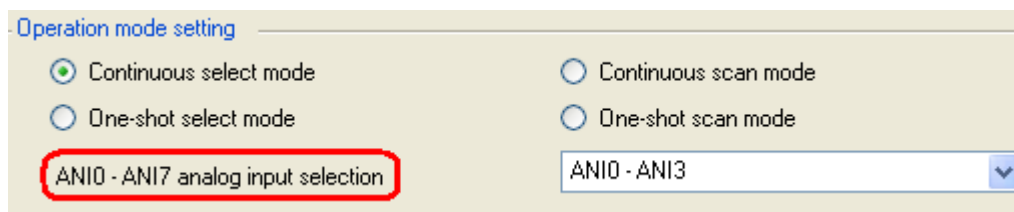
**After :** P50/SDA11

P30/SCL11

This issue has been corrected in Code Generator for RL78,78K0R,78K0 V1.00.03

#### 4.1.7 GUI correction of an A/D converter

In the A/D converter, it corrected so that the message displayed with the fixed value as the number of analog input channels might be dynamically displayed according to a number of channels.



This issue has been corrected in Code Generator for RL78,78K0R,78K0 V1.00.03

#### 4.1.8 Conversion time setup of A/D correction

In the A/D converter, it corrected so that the message displayed with the fixed value as the number of analog input channels might be dynamically displayed according to a number of channels. In the RL78/G1A A/D converter, since conversion time was not able to be set up, it corrected that an A/D converter could not be used.

This issue has been corrected in Code Generator for RL78,78K0R,78K0 V1.00.05

## Chapter 5. Cautions

This section describes cautions for using Code Generator for RL78,78K0R,78K0.

### 5.1 Cautions List

No.	Description	Corresponds of code generation												
		RL78/F12	RL78/G1A	RL78/G12	RL78/G13	RL78/G14	RL78/I1A	78K0R/Fx3	78K0R/Ix3	78K0R/Kx3	78K0R/Kx3-A	78K0R/Kx3-L	78K0R/Lx3	78K0/Kx2-L
		V1.00.00	V1.00.01	V1.02.01	V1.03.01	V1.01.01	V1.01.01	V1.00.01	V1.00.01	V1.00.01	V1.00.01	V1.00.01	V1.00.01	V1.00.00
1	About the LIN-bus function of UART0, UART2, UART3, UART6 or UARTE.	○	○	○	○	○	○	○	○	○	○	○	○	○
2	About the operation for slave transmission of serial interface IICA or IIC0.	○	○	○	○	○	○	/	○	○	○	○	○	○
3	About extension code, multimaster, wakeup function of serial interface IICA or IIC0	/	/	-	-	-	-	/	○	○	○	○	○	○
4	About cooperation with the linker option	/	/	/	/	/	/	/	/	/	/	/	/	○
5	About CAN controllers	/	/	/	/	/	/	○	/	/	/	/	/	/
6	About PORT	/	/	-	-	-	-	/	/	/	/	/	/	/
7	About the SNOOZE mode of Serial array unit 1	/	/	-	-	-	-	/	/	/	/	/	/	/
8	About a setup of a real-time clock	/	/	/	○	○	○	/	/	/	/	/	/	/
9	Cautions when using a DTC function	/	/	/	/	○	/	/	/	/	/	/	/	/
10	About initial function of an A/D converter	/	/	○	○	○	/	/	/	/	/	/	/	/
11	About initial function at the time of setting up UART transmission	/	/	○	○	○	/	/	/	/	/	/	/	/
12	Conversion time setup of A/D correction	/	-	/	/	/	/	/	/	/	/	/	/	/
13	About Complementary assistant PWM mode of TimerRD	/	/	/	/	○	/	/	/	/	/	/	/	/

○ : Correspondence, -: Not correspondence, /:Outside of function.

## 5.2 Cautions Details

### 5.2.1 About the LIN-bus function of UART2 or UART3 or UART6

The code generator is not supporting the LIN-bus functions of serial interface UART2 or UART3 or UART6.

[Workaround] There is no workaround.

### 5.2.2 About the operation for slave transmission of serial interface IICA or IIC0

The code generator is not supporting the extension code, multimaster, wakeup function of serial interface IIC.

[Workaround] There is no workaround.

### 5.2.3 About extension code, multimaster,wakeup function of serial interface IICA or IIC0

During slave transmission, if the master receiver does not return an ACK after the final data is received, then the error API IICA\_SlaveErrorCallback(MD\_NACK) will be called, regardless of whether the actual slave transmission process ended. For this reason, the program will not terminate normally.

[Work-around]

If the master being communicated with does not return an ACK after the final data reception, change IICA\_SlaveHandler's internal code as follows. (So that it does not check for an ACK after the final data is received. The figure below for the serial interface IICA.)

```
void IICA_SlaveHandler(void)
{
  ...
  if (TRC0 == 1U)
  {
    if (ACKD0 == 0U)
    {
      IICA_SlaveErrorCallback(MD_NACK);
    }
    else
    {
      if (glicaTxCnt > 0U)
      {
        IICA = *gplicaTxAddress;
        gplicaTxAddress++;
        glicaTxCnt--;
      }
      else
      {
        IICA_SlaveSendEndCallback();
        WREL0 = 1U;
      }
    }
  }
}
```

if ((ACKD0 == 0U) && (glicaTxCnt != 0))

[Workaround] RL78 Code Generator is corrected in V1.00.02.

### 5.2.4 About cooperation with the linker option

The setting of on the chip debugging of the code generation is not coordinated with " Set user option byte" of link-option.

[Workaround] There is no workaround.

### 5.2.5 About CAN controllers

The code generator is not supporting the CAN Controllers.

[Workaround] There is no workaround.

### 5.2.6 About PORT

There are notes in the port setting of RL78/G13(R5F100LJ, R5F100LK, R5F100LL).

Please do not use a item of P43, P52, P53, and P54 ("TTL buffer" or "N-ch").

[Workaround] RL78 Code Generator is corrected in V1.00.02.

### 5.2.7 About the SNOOZE mode of Serial array unit 1

The code generation of RL78/G13 in not supporting the SNOOZE mode of serial array unit 1.

[Workaround] RL78 Code Generator is corrected in V1.00.02.

### 5.2.8 About a setup of a real-time clock

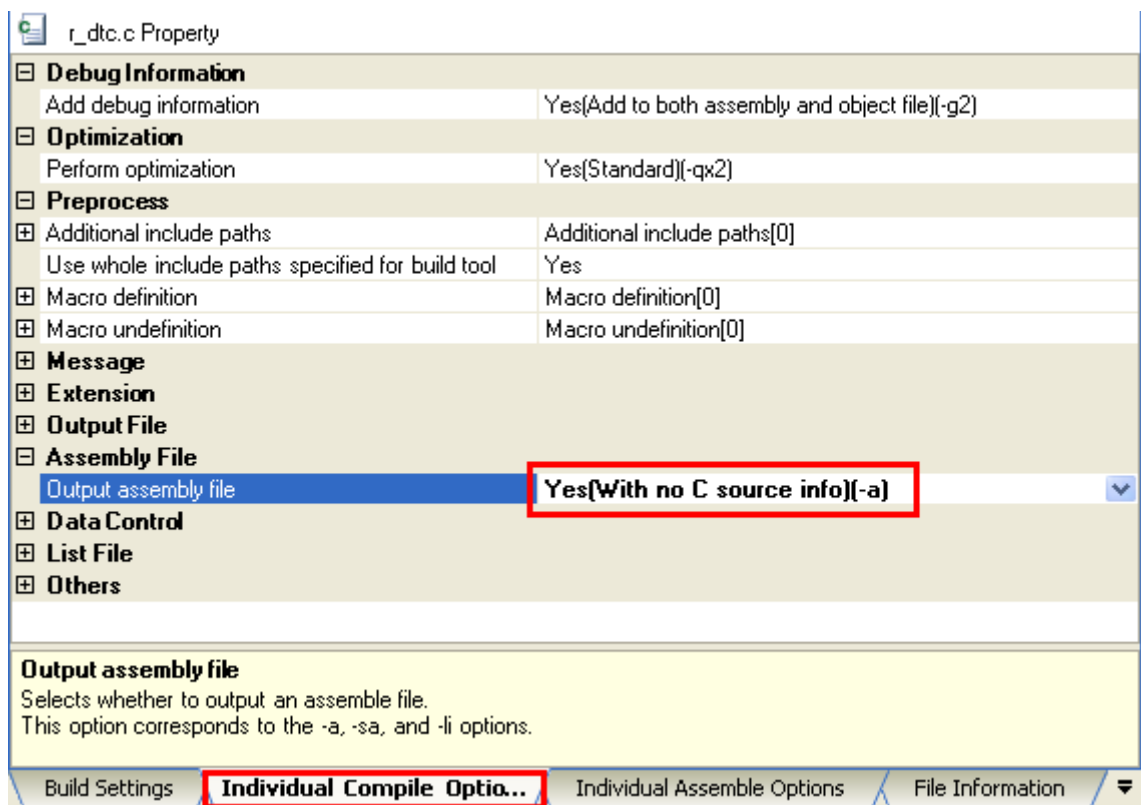
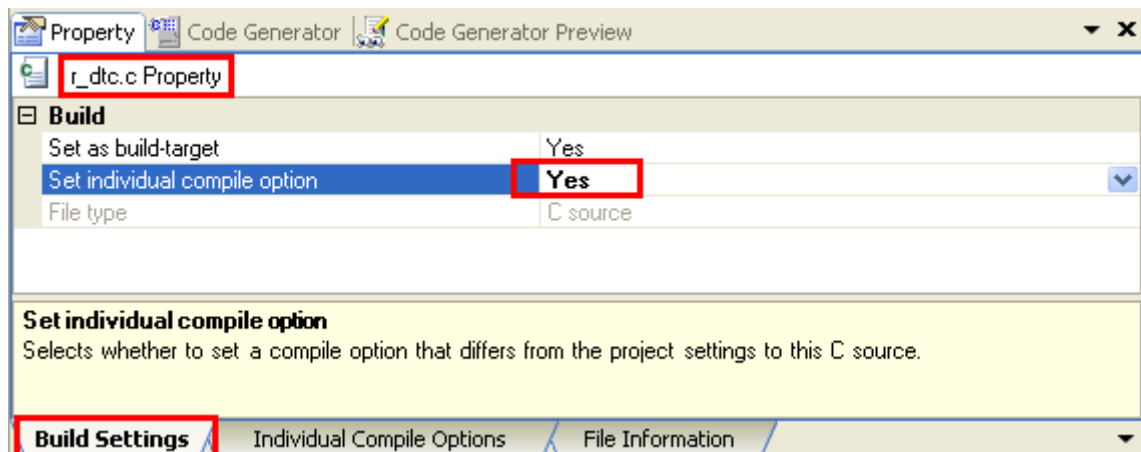
When a clock sauce is set to 15(fIL)kHz on device, clock function cannot be used. However, it is displayed on GUI that clock function seems to be used with 15(fIL)kHz. Please do not set up clock function.

[Workaround] There is no workaround

### 5.2.9 Cautions when using a DTC function

When DTC is used, please set up the following individual option of building. The DATA section is added for DTC to output source file "r\_cg\_dtc.c." Unless the individual option is set up, the following warning message is displayed and an object file is not generated.

( CC78K0R warning W0837: Output assembler source file , not object file )



[Workaround] There is no workaround



### 5.2.10 About initial function of an A/D converter

After making the port 2 a setup which does not compete with an A/D converter, the initialization function at the time of setting up an analog input terminal by an A/D converter has an error. Source code outputted by R\_ADC\_Create() "PM2 |= 0x??" The value of 0x?? has an error.

[Workaround] Please set up an A/D converter before setting up the port 2. The right value will be reflected if the port 2 is finally set up. It is due to be corrected by the code generator of the following version.

### 5.2.11 About initial function at the time of setting up UART transmission

The source code of a SDRmn register setup is not outputted to initialization function R\_UARTn\_Create() at the time of choosing only UART transmission.

[Workaround] There is no workaround. It is due to be corrected by the code generator of the following version.

### 5.2.12 Conversion time setup of A/D correction

Conversion time of the A/D converter of RL78/G1A cannot be set up. Therefore, an A/D converter cannot be used.

[Workaround] There is no workaround. RL78 Code Generator is corrected in V1.00.05.

### 5.2.13 About Complementary assistant PWM mode of TimerRD

When TimerRD Complementary PWM mode is used using a high-speed system clock by clock setup of RL78/G14, it is necessary to change a setup of an option byte. RL78/G14 512 pages of R01UH0186JJ0100 Rev.1.00 edited by user's manual hardware Please refer to Notes 1.

[Workaround] There is no workaround. It is due to be corrected by the code generator of the following version.

## Chapter 6. Restrictions

This section describes the restrictions for the Code Generator for RL78,78K0R,78K0.

### 6.1 Restrictions List

No	Description	Corresponds of code generation													
		RL78/F12	RL78/G1A	RL78/G12	RL78/G13	RL78/G14	RL78/I1A	78K0R/Fx3	78K0R/Ix3	78K0R/Kx3	78K0R/Kx3-A	78K0R/Kx3-L	78K0R/Lx3	78K0/Ix2	78K0/Kx2-L
1	About the coding rule of MISRA-C.	○	○	○	○	○	○	○	○	○	○	○	○	○	○
2	About High-speed on-chip oscillator frequency select register	○	○	○	○	○	/	/	/	/	/	/	/	/	/
3	About internal low-speed or internal high-speed oscillator trimming register	○	○	○	○	○	○	/	○	/	/	/	/	/	/
4	About restriction of a serial array unit	/	/	/	/	/	○	/	/	/	/	/	/	/	/

○ : Correspondence, /: Outside of function

### 6.2 Restrictions Details

#### 6.2.1 About the 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.

#### 6.2.2 About High-speed on-chip oscillator frequency select register

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

#### 6.2.3 About internal low-speed or internal high-speed oscillator trimming register

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

#### 6.2.4 About restriction of a serial array unit

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

## Chapter 7. Changes in User's Manual

This section describes errata in CubeSuite+ documentation. The same content is also contained in the Help file, and should be replaced by this content.

### 7.1 Modifications in RL78 design

This section describes modifications in User's Manual RL78 Design(document #R20UT0548EJ01000).

#### 7.1.1 Changed description relating to the features of Code generating function

Location: Page 10

Before change: Remarks 1. Source code output by the Code Generator conforms to the MISRA-C (Guidelines for the Use of the C Language in Vehicle Based Software) coding convention.

After change: Remarks 1. Compliance with the MISRA-C (Guidelines for the Use of the C Language in Vehicle Based Software) coding convention is planned for source code output by the code generator.

### 7.2 Modifications in 78K0R design

This section describes modifications in User's Manual 78K0R Design(document #R20UT0547EJ01000).

#### 7.2.1 Changed description relating to the features of Code generating function

Location: Page 10

Before change: Remarks 1. Source code output by the Code Generator conforms to the MISRA-C (Guidelines for the Use of the C Language in Vehicle Based Software) coding convention.

After change: Remarks 1. Compliance with the MISRA-C (Guidelines for the Use of the C Language in Vehicle Based Software) coding convention is planned for source code output by the code generator.

### 7.3 Modifications in 78K0 design

This section describes modifications in User's Manual 78K0 Design(document #R20UT0546EJ01000).

#### 7.3.1 Changed description relating to the features of Code generating function

Location: Page 10

Before change: Remarks 1. Source code output by the Code Generator conforms to the MISRA-C (Guidelines for the Use of the C Language in Vehicle Based Software) coding convention.

After change: Remarks 1. Compliance with the MISRA-C (Guidelines for the Use of the C Language in Vehicle Based Software) coding convention is planned for source code output by the code generator.

### 7.4 About API added and changed by V1.00.02.

API changed and added to the User's Manual RL78 Design(document #R20UT0548EJ01000) of the front version is explained.Since it is an important change, this document is also published.

#### 7.4.1 The API table changed or added by RL78 Code Generator function

Peripheral Function	API function name		RL78/G13 V1.03.00.	RL78/G14 V1.01.00	RL78/1A V1.01.00
	Before change	After change			
Interrupt	R_INTCn_Interrupt	<a href="#">r_intcn_interrupt</a>	○	○	○
	R_KEY_Interrupt	<a href="#">r_key_interrupt</a>	○	○	○
Serial	R_UARTn_Interrupt_Send	<a href="#">r_uartn_interrupt_send</a>	○	○	○
	R_UARTn_Interrupt_Receive	<a href="#">r_uartn_interrupt_receive</a>	○	○	○
	R_UARTn_Interrupt_Error	<a href="#">r_uartn_interrupt_error</a>	○	○	○
	R_UARTn_Callback_SendEnd	<a href="#">r_uartn_callback_sendend</a>	○	○	○
	R_UARTn_Callback_ReceiveEnd	<a href="#">r_uartn_callback_receiveend</a>	○	○	○
	R_UARTn_Callback_Error	<a href="#">r_uartn_callback_error</a>	○	○	○
	R_UARTn_Callback_SoftwareOverRun	<a href="#">r_uartn_callback_softwareoverrun</a>	○	○	○
	R_CSImn_Interrupt	<a href="#">r_csimn_interrupt</a>	○	○	○
	R_CSImn_Callback_SendEnd	<a href="#">r_csimn_callback_sendend</a>	○	○	○
	R_CSImn_Callback_ReceiveEnd	<a href="#">r_csimn_callback_receiveend</a>	○	○	○
	R_CSImn_Callback_Error	<a href="#">r_csimn_callback_error</a>	○	○	○
	R_IICmn_Interrupt	<a href="#">r_iicmn_interrupt</a>	○	○	○
	R_IICmn_Callback_Master_SendEnd	<a href="#">r_iicmn_callback_master_sendend</a>	○	○	○
	R_IICmn_Callback_Master_ReceiveEnd	<a href="#">r_iicmn_callback_master_receiveend</a>	○	○	○
	R_IICmn_Callback_Master_Error	<a href="#">r_iicmn_callback_master_error</a>	○	○	○
	New addition	<a href="#">R_DALIn_Create</a>	/	/	○
	New addition	<a href="#">R_DALIn_Start</a>	/	/	○
	New addition	<a href="#">R_DALIn_Stop</a>	/	/	○
	New addition	<a href="#">R_DALIn_Send</a>	/	/	○
	New addition	<a href="#">R_DALIn_Receive</a>	/	/	○
	New addition	<a href="#">r_dalin_interrupt_send</a>	/	/	○
	New addition	<a href="#">r_dalin_interrupt_receive</a>	/	/	○
	New addition	<a href="#">r_dalin_interrupt_error</a>	/	/	○
New addition	<a href="#">r_dalin_callback_sendend</a>	/	/	○	
New addition	<a href="#">r_dalin_callback_receiveend</a>	/	/	○	
New addition	<a href="#">r_dalin_callback_error</a>	/	/	○	
New addition	<a href="#">r_dalin_callback_softwareoverrun</a>	/	/	○	

○ : Correspondence, /: Not correspondence

Peripheral Function	API function name		RL78/G13 V1.03.00.	RL78/G14 V1.01.00	RL78/1A V1.01.00
	Before change	After change			
Serial	R_IICAn_Interrupt	<a href="#">r_iican_interrupt</a>	○	○	○
	R_IICAn_Callback_Master_SendEnd	<a href="#">r_iican_callback_master_sendend</a>	○	○	○
	R_IICAn_Callback_Master_ReceiveEnd	<a href="#">r_iican_callback_master_receiveend</a>	○	○	○
	R_IICAn_Callback_Master_Error	<a href="#">r_iican_callback_master_error</a>	○	○	○
	R_IICAn_Callback_Slave_SendEnd	<a href="#">r_iican_callback_slave_sendend</a>	○	○	○
	R_IICAn_Callback_Slave_ReceiveEnd	<a href="#">r_iican_callback_slave_receiveend</a>	○	○	○
	R_IICAn_Callback_Slave_Error	<a href="#">r_iican_callback_slave_error</a>	○	○	○
	R_IICAn_Callback_GetStopCondition	<a href="#">r_iican_callback_getstopcondition</a>	○	○	○
A/D Converter	R_ADC_Interrupt	<a href="#">r_adc_interrupt</a>	○	○	○
Timer	R_TAUm_Channeln_Interrupt	<a href="#">r_taum_channeln_interrupt</a>	○	○	/
	R_TAUm_Channeln_Higher8bits_Interrupt	<a href="#">r_taum_channeln_higher8bits_interrupt</a>	○	○	/
	R_TMR_RDn_Interrupt	<a href="#">r_tmr_rdn_interrupt</a>	/	○	/
	R_TMR_RG0_Interrupt	<a href="#">r_tmr_rg0_interrupt</a>	/	○	/
	R_TMR_RJ0_Interrupt	<a href="#">r_tmr_rj0_interrupt</a>	/	○	/
	New addition	<a href="#">R_TMR_KB_Create</a>	/	○	/
	New addition	<a href="#">R_TMR_KBm_Start</a>	/	○	/
	New addition	<a href="#">R_TMR_KBm_Stop</a>	/	○	/
	New addition	<a href="#">R_TMR_KBm_Set_PowerOff</a>	/	○	/
	New addition	<a href="#">R_TMR_KBm_ForcedOutput_Start</a>	/	○	/
	New addition	<a href="#">R_TMR_KBm_ForcedOutput_Stop</a>	/	○	/
	New addition	<a href="#">R_TMR_KBm_Create_UserInit</a>	/	○	/
	New addition	<a href="#">r_tmr_kbm_interrupt</a>	/	○	/
	New addition	<a href="#">R_TMR_KC0_Create</a>	/	○	/
	New addition	<a href="#">R_TMR_KC0_Start</a>	/	○	/
	New addition	<a href="#">R_TMR_KC0_Stop</a>	/	○	/
	New addition	<a href="#">R_TMR_KC0_Set_PowerOff</a>	/	○	/
	New addition	<a href="#">R_TMR_KC0_Create_UserInit</a>	/	○	/
	New addition	<a href="#">r_tmr_kc0_interrupt</a>	/	○	/

○ : Correspondence, /: Not correspondence

Peripheral Function	API function name		RL78/G13 V1.03.00.	RL78/G14 V1.01.00	RL78/1A V1.01.00
	Before change	After change			
Watchdog timer	R_WDT_Interrupt	<a href="#">r_wdt_interrupt</a>	○	○	○
Real-time Clock	R_RTC_Interrupt	<a href="#">r_rtc_interrupt</a>	○	○	○
	R_RTC_Callback_ConstPeriod	<a href="#">r_rtc_callback_constperiod</a>	○	○	○
	R_RTC_Callback_Alarm	<a href="#">r_rtc_callback_alarm</a>	○	○	○
Interval timer	R_IT_Interrupt	<a href="#">r_it_interrupt</a>	○	○	○
Comparator	R_COMPn_Interrupt	<a href="#">r_compn_interrupt</a>	/	○	○
DMA Controoler	R_DMAn_Interrupt	<a href="#">r_dmacn_interrupt</a>	○	○	○
Voltage Detecor	R_LVD_Interrupt	<a href="#">r_lvd_interrupt</a>	○	○	○
Programmable Gain Amplifier	New addition	<a href="#">R_PGA_Create</a>	/	/	○
	New addition	<a href="#">R_PGA_Start</a>	/	/	○
	New addition	<a href="#">R_PGA_Stop</a>	/	/	○
	New addition	<a href="#">R_PGA_Create_UserInit</a>	/	/	○

○ : Correspondence, /: Not correspondence

## 7.4.2 Details of API added by RL78 Code Generator function

### [API name] **R\_DALIn\_Create**

Performs initialization of the serial interface (DALI) channel.

**Remark** This API function is used as an internal function of R\_SAUm\_Create. For this reason, there is normally no need to call it from a user program.

#### [Classification]

r\_cg\_serial.c

#### [Syntax]

```
void R_DALIn_Create ( void );
```

**Remark** n is the channel number.

#### [Argument(s)]

None.

#### [Return value]

None.

### [API name] **R\_DALIn\_Start**

Sets DALI communication to standby mode.

#### [Classification]

r\_cg\_serial.c

#### [Syntax]

```
void R_DALIn_Start ( void );
```

**Remark** n is the channel number.

#### [Argument(s)]

None.

#### [Return value]

None.

### [API name] **R\_DALIn\_Stop**

Ends DALI communication.

#### [Classification]

r\_cg\_serial.c

#### [Syntax]

```
void R_DALIn_Stop ( void );
```

**Remark** n is the channel number.

#### [Argument(s)]

None.

#### [Return value]

None.

**[API name] R\_DALIn\_Send**

Starts DALI data transmission.

**Remark 1.** This API function repeats the byte-level DALI transmission from the buffer specified in parameter txbuf the number of times specified in parameter txnum.

**2.** When performing a DALI transmission, R\_DALIn\_Start must be called before this API function is called.

**[Classification]**

r\_cg\_serial.c

**[Syntax]**

```
#include "r_cg_macrodriver.h"
```

```
MD_STATUS R_DALIn_Send ( uint8_t *txbuf, uint16_t txnum );
```

**Remark** n is the channel number.

**[Argument(s)]**

I/O	Argument	Description
I	uint8_t *txbuf;	Pointer to a buffer storing the transmission data
I	uint16_t txnum;	Total amount of data to send

**[Return value]**

Macro	Description
MD_OK	Normal completion
MD_ARGERROR	Invalid argument specification

**[API name] R\_DALIn\_Receive**

Starts DALI data reception.

**Remark 1.** This API function performs byte-level DALI reception the number of times specified by the parameter rxnum, and stores the data in the buffer specified by the parameter rxbuf.

**2.** Actual DALI reception starts after this API function is called, and R\_DALIn\_Start is then called.

**[Classification]**

r\_cg\_serial.c

**[Syntax]**

```
#include "r_cg_macrodriver.h"
```

```
MD_STATUS R_DALIn_Receive ( uint8_t *rxbuf, uint16_t rxnum );
```

**Remark** n is the channel number.

**[Argument(s)]**

I/O	Argument	Description
O	uint8_t *rxbuf;	Pointer to a buffer to store the received data
I	uint16_t rxnum;	Total amount of data to receive

**[Return value]**

Macro	Description
MD_OK	Normal completion
MD_ARGERROR	Invalid argument specification



**[API name] r\_dalin\_interrupt\_send**

Performs processing in response to the DALI transmission end interrupt INTSTDLn.

**Remark** This API function is called as the interrupt process corresponding to the DALI transmission end interrupt INTSTDLn.

**[Classification]**

r\_cg\_serial\_user.c

**[Syntax]**

```
void r_dalin_interrupt_send ( void );
```

**Remark** n is the channel number.

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] r\_dalin\_interrupt\_receive**

Performs processing in response to the DALI reception end interrupt INTSRDLn.

**Remark** This API function is called as the interrupt process corresponding to the DALI reception end interrupt INTSRDLn.

**[Classification]**

r\_cg\_serial\_user.c

**[Syntax]**

```
void r_dalin_interrupt_receive ( void );
```

**Remark** n is the channel number.

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] r\_dalin\_interrupt\_error**

Performs processing in response to the DALI reception error interrupt INTSREDLn.

**Remark** This API function is called as the interrupt process corresponding to the DALI reception error interrupt INTSREDLn.

**[Classification]**

r\_cg\_serial\_user.c

**[Syntax]**

```
void r_dalin_interrupt_error ( void );
```

**Remark** n is the channel number.

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] r\_dalin\_callback\_sendend**

Performs processing in response to the DALI transmission end interrupt INTSTDLn.

**Remark** This API function is called as the callback routine of interrupt process R\_DALIn\_Interrupt\_Send corresponding to the DALI transmission end interrupt INTSTDLn (performed when number of transmission data specified by R\_DALIn\_Send parameter txnum has been completed).

**[Classification]**

r\_cg\_serial\_user.c

**[Syntax]**

```
void r_dalin_callback_sendend ( void );
```

**Remark** n is the channel number.

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] r\_dalin\_callback\_receiveend**

Performs processing in response to the DALI reception end interrupt INTSRDLn.

**Remark** This API function is called as the callback routine of interrupt process R\_DALIn\_Interrupt\_Receive corresponding to the DALI reception end interrupt INTSRDLn (performed when number of received data specified by R\_DALIn\_Receive parameter rxnum has been completed).

**[Classification]**

r\_cg\_serial\_user.c

**[Syntax]**

```
void r_dalin_callback_receiveend ( void );
```

**Remark** n is the channel number.

**[Argument(s)]**

None.

**[Return value]**

None.

[API name] **r\_dalin\_callback\_error**

Performs processing in response to the DALI reception error interrupt INTSREDLn.

**Remark** This API function is called as the callback routine of interrupt process R\_DALIn\_Interrupt\_Error corresponding to the DALI reception error interrupt INTSREDLn.

[Classification]

r\_cg\_serial\_user.c

[Syntax]

```
#include "r_cg_macrodriver.h"

void r_dalin_callback_error ( uint8_t err_type );
```

**Remark** n is the channel number.

[Argument(s)]

I/O	Argument	Description
O	uint8_t err_type;	Trigger for DALI reception error interrupt 00000xx1B: Overrun error 00000x1xB: Parity error 000001xxB: Framing error

[Return value]

None.

[API name] **r\_dalin\_callback\_softwareoverrun**

Performs processing in response to detection of overrun error.

**Remark** This API function is called as the callback routine of interrupt process R\_DALIn\_Interrupt\_Receive corresponding to the DALI reception end interrupt INTSRDLn (process performed when the amount of data received is greater than the parameter rxnum specified for R\_DALIn\_Receive).

[Classification]

r\_cg\_serial\_user.c

[Syntax]

```
#include "r_cg_macrodriver.h"

void r_dalin_callback_softwareoverrun ( uint16_t rx_data );
```

**Remark** n is the channel number.

[Argument(s)]

I/O	Argument	Description
O	Uint16_rx_data;	Receive data (greater than the parameter rxnum specified for R_DALIn_Receive)

[Return value]

None.

**[API name] R\_TMR\_KB\_Create**

Performs initialization necessary to control the 16-bit timer KBm functions.

**[Classification]**

r\_cg\_timer.c

**[Syntax]**

```
void R_TMR_KB_Create ( void );
```

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] R\_TMR\_KBm\_Start**

Starts the count for 16-bit timer KBm.

**[Classification]**

r\_cg\_timer.c

**[Syntax]**

```
void R_TMR_KBm_Start ( void );
```

**Remark m** is the unit number.

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] R\_TMR\_KBm\_Stop**

Ends the count for 16-bit timer KBm.

**[Classification]**

r\_cg\_timer.c

**[Syntax]**

```
void R_TMR_KBm_Stop ( void );
```

**Remark m** is the unit number.

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] R\_TMR\_KBm\_Set\_PowerOff**

Halts the clock supplied to the 16-bit timer KBm.

**Remark** Calling this API function changes the 16-bit timer KBm to reset status. For this reason, writes to the control registers after this API function is called are ignored.

**[Classification]**

r\_cg\_timer.c

**[Syntax]**

```
void R_TMR_KBm_Set_PowerOff ( void );
```

**Remark** m is the unit number.

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] R\_TMR\_KBm\_ForcedOutput\_Start**

Enables input of the trigger signal used for the forced output stop function.

**[Classification]**

r\_cg\_timer.c

**[Syntax]**

```
void R_TMR_KBm_ForcedOutput_Start ( void );
```

**Remark** m is the unit number.

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] R\_TMR\_KBm\_ForcedOutput\_Sop**

Disables input of the trigger signal used for the forced output stop function.

**[Classification]**

r\_cg\_timer.c

**[Syntax]**

```
void R_TMR_KBm_ForcedOutput_Stop ( void );
```

**Remark** m is the unit number.

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] R\_TMR\_KBm\_Create\_UserInit**

Performs user-defined initialization relating to the 16-bit timer KBm.

**Remark** This API function is called as the R\_TMR\_KB\_Create callback routine.

**[Classification]**

r\_cg\_timer\_user.c

**[Syntax]**

```
void R_TMR_KBm_Create_UserInit ( void );
```

**Remark** m is the unit number.

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] r\_tmr\_kbm\_interrupt**

Performs processing in response to the timer interrupt.

**Remark** This API function is called as the interrupt process corresponding to the timer interrupt.

**[Classification]**

r\_cg\_timer\_user.c

**[Syntax]**

```
void r_tmr_kbm_interrupt ( void );
```

**Remark** m is the unit number.

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] R\_TMR\_KC0\_Create**

Performs initialization necessary to control the 16-bit timer KC0 functions.

**[Classification]**

r\_cg\_timer.c

**[Syntax]**

```
void R_TMR_KC0_Create ( void );
```

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] R\_TMR\_KC0\_Start**

Starts the count for 16-bit timer KC0.

**[Classification]**

r\_cg\_timer.c

**[Syntax]**

```
void R_TMR_KC0_Start ( void );
```

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] R\_TMR\_KC0\_Stop**

Ends the count for 16-bit timer KC0.

**[Classification]**

r\_cg\_timer.c

**[Syntax]**

```
void R_TMR_KC0_Stop ( void );
```

**Remark m** is the unit number.

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] R\_TMR\_KC0\_Set\_PowerOff**

Halts the clock supplied to the 16-bit timer KC0.

**Remark** Calling this API function changes the 16-bit timer KC0 to reset status. For this reason, writes to the control registers after this API function is called are ignored.

**[Classification]**

r\_cg\_timer\_user.c

**[Syntax]**

```
void R_TMR_KC0_Set_PowerOff ( void );
```

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] R\_TMR\_KC0\_Create\_UserInit**

Performs user-defined initialization relating to the 16-bit timer KC0.

**Remark** This API function is called as the R\_TMR\_KC0\_Create callback routine.

**[Classification]**

r\_cg\_timer\_user.c

**[Syntax]**

```
void R_TMR_KC0_Create_UserInit ( void );
```

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] r\_tmr\_kc0\_interrupt**

Performs processing in response to the timer interrupt.

**Remark** This API function is called as the interrupt process corresponding to the timer interrupt.

**[Classification]**

r\_cg\_timer\_user.c

**[Syntax]**

```
void r_tmr_kc0_interrupt ( void );
```

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] R\_PGA\_Create**

Performs initialization necessary to control programmable gain amplifiers functions.

**[Classification]**

r\_cg\_pga.c

**[Syntax]**

```
void R_PGA_Create ( void );
```

**[Argument(s)]**

None.

**[Return value]**

None.



**PI name] R\_PGA\_Start**

Starts the operation of programmable gain amplifier.

**[Classification]**

r\_cg\_pga.c

**[Syntax]**

```
void R_PGA_Start ( void );
```

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] R\_PGA\_Stop**

Ends the operation of programmable gain amplifier.

**[Classification]**

r\_cg\_pga.c

**[Syntax]**

```
void R_PGA_Stop ( void );
```

**[Argument(s)]**

None.

**[Return value]**

None.

**[API name] R\_PGA\_Create\_UserInit**

Performs user-defined initialization relating to the programmable gain amplifiers.

**Remark** This API function is called as the R\_PGA\_Create callback routine.

**[Classification]**

r\_cg\_pga\_user.c

**[Syntax]**

```
void R_PGA_Create_UserInit ( void );
```

**[Argument(s)]**

None.

**[Return value]**

None.

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of 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. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may 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.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.  
"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.  
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.  
"Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate the safety of the final products or system 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.  
(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.  
(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.



### SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

Refer to "<http://www.renesas.com/>" for the latest and detailed information.

#### Renesas Electronics America Inc.

2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.  
Tel: +1-408-588-6000, Fax: +1-408-588-6130

#### Renesas Electronics Canada Limited

1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada  
Tel: +1-905-898-5441, Fax: +1-905-898-3220

#### Renesas Electronics Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K  
Tel: +44-1628-585-100, Fax: +44-1628-585-900

#### Renesas Electronics Europe GmbH

Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-65030, Fax: +49-211-6503-1327

#### Renesas Electronics (China) Co., Ltd.

7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

#### Renesas Electronics (Shanghai) Co., Ltd.

Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China  
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

#### Renesas Electronics Hong Kong Limited

Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2886-9318, Fax: +852-2886-9022/9044

#### Renesas Electronics Taiwan Co., Ltd.

13F, No. 363, Fu Shing North Road, Taipei, Taiwan  
Tel: +886-2-8175-9600, Fax: +886-2-8175-9670

#### Renesas Electronics Singapore Pte. Ltd.

1 harbourFront Avenue, #06-10, Keppel Bay Tower, Singapore 098632  
Tel: +65-6213-0200, Fax: +65-6278-8001

#### Renesas Electronics Malaysia Sdn.Bhd.

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

#### Renesas Electronics Korea Co., Ltd.

11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141