CubeSuite+ Code Generator for RL78,78K0R,78K0 V1.00.06

Release Note

R20UT2243EJ0100
Rev.1.00
October 1, 2012

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 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
4.1.8 Conversion time setup of A/D correction 20
4.1.9 Changes of square wave output of a timer 20
4.1.10 Changes of the TTL check box of a port 20
4.1.11 Changes of PIOR setup 20
4.1.12 Changes of TAU1 setup 20
4.1.13 Changes of UART2 setup 21
4.1.14 Changes of key interrupt function 21
4.1.15 Changes of Simplified I2C 21
4.1.16 Additional function generation file mode 22
4.1.17 Changes of hdwinit() function 22

Chapter 5. Cautions 24
5.1 Cautions List 24
5.2 Cautions Details 25
5.2.1 Cautions of the LIN-bus function of UART2 or UART3 or UART6 25
5.2.2 Cautions of the operation for slave transmission of serial interface IICA or IIC0 25
5.2.3 Cautions of extension code, multimaster,wakeup function of serial interface IICA or IIC0 25
5.2.4 Cautions of cooperation with the linker option 26
5.2.5 Cautions of CAN controllers 26
5.2.6 Cautions of PORT 26
5.2.7 Cautions of the SNOOZE mode of Serial array unit 1 26
5.2.8 Cautions of a setup of a real-time clock 26
5.2.9 Cautions when using a DTC function 27
5.2.10 Cautions of initial function of an A/D converter 28
5.2.11 Cautions of initial function at the time of setting up UART transmission 28
5.2.12 Conversion time setup of A/D correction 28
5.2.13 Cautions of Complementary assistant PWM mode of TimerRD 28
5.2.14 Cautions of Pin Configurator 28

Chapter 6. Restrictions 29
6.1 Restrictions List 29
6.2 Restrictions Details 29
6.2.1 Restrictions of the coding rule of MISRA-C 29
6.2.2 Restrictions of High-speed on-chip oscillator frequency select register 29
6.2.3 Restrictions of internal low-speed or internal high-speed oscillator trimming 29
6.2.4 Restriction of a serial array unit 29

Chapter 7. Changes in User's Manual 30
7.1 API added and changed by V1.00.02. 30
7.1.1 The API table changed or added by RL78 Code Generator function 30
7.1.2 Details of API added by RL78 Code Generator function 34

Chapter 1. Target Devices

Below is a list of devices supported by the Code Generator for RL78/I1A V1.01.03.01	
PIN	Device name
20pin	R5F1076C
30pin	R5F107AC, R5F107AE
32pin	R5F107BC
38pin	R5F107DE
The Code Generator is based on the following documents.	
Manual Name	Document Number
RL78/I1A User's Manual: Hardware	R01UH0169JJ0100 Rev.1.00
	R01UH0169EJ0100 Rev.1.00

Below is a list of devices supported by the Code Generator for RL78/G12 V1.02.03.03	
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 is based on the following documents.	
Manual Name	Document Number
RL78/G12 User's Manual: Hardware	R01UH0200JJ0100 Rev.1.00
	R01UH0200EJ0100 Rev.1.00

Below is a list of devices supported by the Code Generator for RL78/G13 V1.03.03.05	
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 is based on the following documents.

Manual Name	Document Number
RL78/G13 User's Manual: Hardware	R01UH0146JJ0100 Rev.1.00
	R01UH0146EJ0100 Rev.1.00

Below is a list of devices supported by the Code Generator for RL78/G14 V1.01.03.06	
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 is based on the following documents.	
Manual Name	Document Number
RL78/G14 User's Manual: Hardware	R01UH0186JJ0100 Rev.1.00
	R01UH0186EJ0100 Rev.1.00

Below is a list of devices supported by the Code Generator for RL78/G1A V1.00.03.01	
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 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 V1.00.02.03	
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 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 RL78/L12 V1.00.01.02	
PIN	Device name
32pin	R5F10RBC, R5F10RBA, R5F10RB8
44pin	R5F10RFC, R5F10RFA, R5F10RF8
48pin	R5F10RGC, R5F10RGA, R5F10RG8
52pin	R5F10RJC, R5F10RJA, R5F10RJ8
64pin	R5F10RLC, R5F10RLA
The Code Generator is based on the following documents	
Manual Name	Document Number
RL78/L12 User's Manual: Hardware	R01UH0330JJ0001 Rev.0.01
	R01UH0330EJ0001 Rev.0.01

Below is a list of devices supported by the Code Generator for 78K0R/Fx3 V1.00.03.02	
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 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/Ix3 V1.00.03.03	
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 is based on the following documents.	
Manual Name	Document Number
78K0R/Ix3 User's Manual	U19678JJ1V1UD00
	U19678EJ1V1UD00

Below is a list of devices supported by the Code Generator for 78K0R/Kx3 V1.00.03.02	
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 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 V1.00.03.03	
Nickname	Device name
78K0R/KE3-A	μPD78F1016, μPD78F1017, μPD78F1018

The Code Generator 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 V1.00.03.02	
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 is based on the following documents	
Manual Name	Document Number
78K0R/Kx3-L User's Manual	U19291JJ3V0UD00
	U19291EJ2V0UD00
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 V1.00.03.03	
Nickname	Device name
78K0R/LF3	μPD78F1500, μPD78F1501, μPD78F1502
78K0R/LG3	μPD78F1503, μPD78F1504, μPD78F1505
78K0R/LH3	μPD78F1506, μPD78F1507, μPD78F1508

The Code Generator 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/Ix2 V1.00.02.02	
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 is based on the following documents.	
Manual Name	Document Number
78K0/Ix2 User's Manual	U19353JJ3V0UD00
	U19353EJ3V0UD00

Below is a list of devices supported by the Code Generator for 78K0/Kx2-L V1.00.02.02	
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 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+ V1.03.00 RL78 Design	R20UT2136EJ0100
CubeSuite+ V1.03.00 78K0R Design	R20UT2137EJ0100
CubeSuite+ V1.03.00 78K0 Design	R20UT2138EJ0100
CubeSuite+ V1.03.00 Message	R20UT2147EJ0100

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 V1.00.05 to V1.00.06

No	内容	Corresponds of code generation													
		RL78/L12 V1.00.01.02	RL78/F12 V1.02.02.03	RL78/G1A V1.00.03.01	RL78/G12 V1.02.03.03	RL78/G13 V1.03.03.05	RL78/G14 V1.01.03.06	RL78/I1A V1.01.03.01	78K0R/Fx3 V1.00.03.02	78K0R/Ix3 V1.00.03.03	78K0R/Kx3 V1.00.03.02	78K0R/Kx3-A V1.00.03.03	78K0R/Kx3-L V1.00.03.02	78K0R/Lx3 V1.00.03.03	78K0/Ix2 V 1.00.02.02	78K0/Kx2-L V 1.00.02.02
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	/	/	-	/	/	/	/	/	/	/	/	/	/	/	/
9	Changes of square wave output of a timer	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
10	Changes of the TTL check box of a port	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
11	Changes of PIOR setup	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
12	Changes of TAU1 setup	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
13	Changes of UART2 setup	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
14	Changes of key interrupt function	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
15	Changes of Simplified I2C	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
16	Additional function generation file mode	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
17	Changes of hdwinit() function	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

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

The following MCUs of the RL78 family have been supported:

- RL78/L12 groups

4.1 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_Iic00MasterStatusFlag & _04_SAU_IIC_SENDED_ADDRESS_FLAG) == 0U)
{
    rxadr = SIO00;
    SCR00 &= ~_C000_SAU_RECEPTION_TRANSMISSION;
    SCR00 |= _4000_SAU_RECEPTION;
    g_Iic00MasterStatusFlag |= _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 Gnerator RL78/G13 V1.00.02.

Peripheral Function	Filename	
	Befoe	After
Clock Generator	r_cgc.c	r_cg_cgc.c
	r_cgc_user.c	r_cg_cgc_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

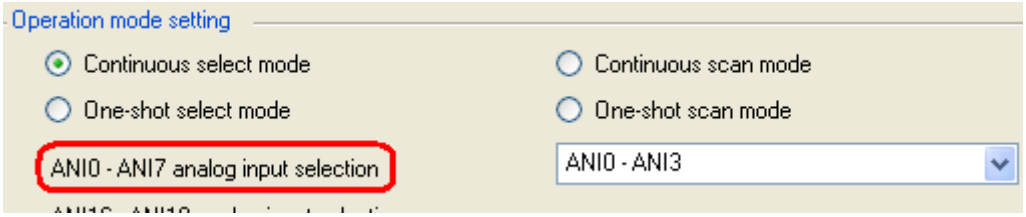
After : P50/SDA11
P30/SCL11

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

4.1.9 Changes of square wave output of a timer

If code is generated so that timers TAUx (x is 1 to 7) of an 80-, 100-, or 128-pin MCU can output square wave, the values of the TOM1 and TOL1 registers, which control TAUx, are not set but those of the TOM0 and TOL0 registers are set.

This issue has been corrected in Code Generator for RL78,78K0R,78K0 V1.00.06

4.1.10 Changes of the TTL check box of a port

There is no check box which sets TTL as P10 and P11 with 30-pin MCU.

This issue has been corrected in Code Generator for RL78,78K0R,78K0 V1.00.06

4.1.11 Changes of PIOR setup

In the code for setting registers PIOR01 and PIOR04 to 1s in an arrangement of pin assignments, incorrect pins are assigned to INTP10 and INTP11 as follows:

Incorrect:	Correct:
P110 assigned to INTP10	P100 assigned to INTP10
P111 assigned to INTP11	P110 assigned to INTP11

This issue has been corrected in Code Generator for RL78,78K0R,78K0 V1.00.06

4.1.12 Changes of TAU1 setup

If code is generated in an 80- or 100-pin MCU, no one except "interval" can be selected in the functional selection of timer TAU1.

This issue has been corrected in Code Generator for RL78,78K0R,78K0 V1.00.06

4.1.13 Changes of UART2 setup

If the code is generated for making settings of UART2 and any of the ports except 13 and 14, an error arises in building it.
Example:
If you use UART2 and set ports 10, 11, and 12 to the output state, the following code is generated; however, the last "|" is unnecessary:
PMC1 = ・・・ | _80_PMCn7_NOT_USE | ;
If build is performed including this code, an error arises. It must be read as follows:
PMC1 = ・・・ | _80_PMCn7_NOT_USE ;

This issue has been corrected in Code Generator for RL78,78K0R,78K0 V1.00.06

4.1.14 Changes of key interrupt function

If you make settings of the key interrupt flag and the triggering edge, the settings cannot properly be reflected to the KRCTL register by the generated code.

Example:
If you select Use among from the key flag pull-down list and Falling Edge among from the triggering edge pull-down list, the code generator generates the following incorrect codes.

```
KRCTL |= _00_KR_FLAG_UNUSED;
KRCTL |= _01_KR_EDGE_RISING;
```

The correct codes are as follows:

```
KRCTL |= _01_KR_FLAG_USED;
KRCTL |= _00_KR_EDGE_FALLING;
```

This issue has been corrected in Code Generator for RL78,78K0R,78K0 V1.00.06

4.1.15 Changes of Simplified I2C

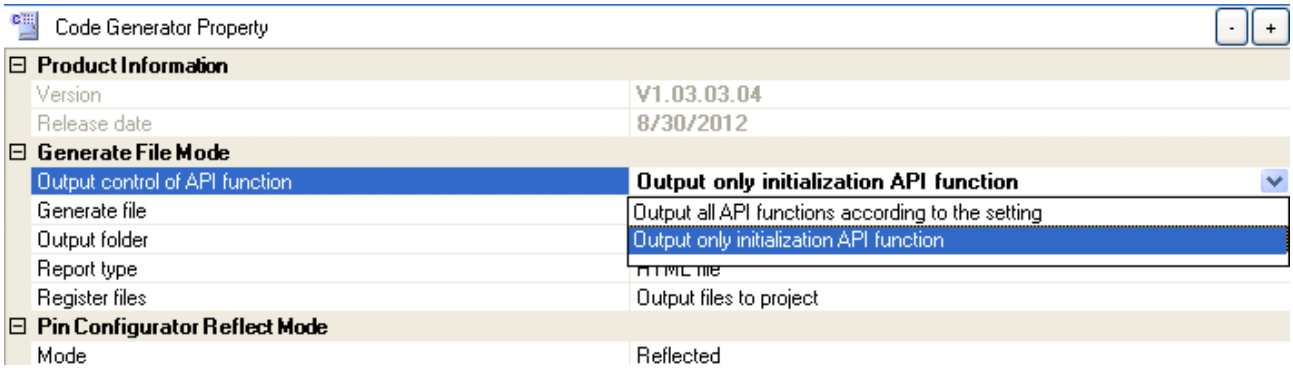
I When a receiving byte is set to 1 by Simplified I2C, it does not operate normally.

This issue has been corrected in Code Generator for RL78,78K0R,78K0 V1.00.06

4.1.16 Additional function generation file mode

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 V1.00.06



4.1.17 Changes of hdwinit() function

We have changed the initial code for the hdwinit() and main() functions.

```
void hdwinit(void)
{
    DI();
    R_Systeminit();
    EI();
}
```

The above code has been changed to the code given below. Accordingly, interrupts are not enabled within the hdwinit function.

```
void hdwinit(void)
{
    DI();
    R_Systeminit();
}
```

Interrupts are now enabled within the main() function.

```
/***********************************************************************************************************************
* Function Name: main
* Description  : This function implements main function.
***********************************************************************************************************************/
void main(void)
{
    R_MAIN_UserInit();
    /* Start user code. Do not edit comment generated here */
    while (1U)
    {
        ;
    }
    /* End user code. Do not edit comment generated here */
}
```

```
/***********************************************************************************************************************
* Function Name: R_MAIN_UserInit
* Description  : This function adds user code before implementing main function.
***********************************************************************************************************************/
void R_MAIN_UserInit(void)
{
    /* Start user code. Do not edit comment generated here */
    EI();
    /* End user code. Do not edit comment generated here */
}
```

When an old project is used in code generation, the definitions of variables within the main function may lead to errors.

[Old project]

```
void main(void)
{
    /* Start user code. Do not edit comment generated here */
    char c;
    while (1U)
    {
        ...
```

[When an old project is loaded into CubeSuite+V1.03.00 and used for code generation]

```
void main(void)
{
    R_MAIN_UserInit();
    /* Start user code. Do not edit comment generated here */
    char c;         <- error!!
    while (1U)
    {
        ...
```

In that case, use { }.

```
void main(void)
{
    R_MAIN_UserInit();
    /* Start user code. Do not edit comment generated here */
    {               <- add “{“
      char c;       <- not error!
      while (1U)
      {
          ...
          }
    }               <- add “}”
```

This issue has been corrected in Code Generator for RL78,78K0R,78K0 V1.00.06

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/I12 V1.00.01.02	RL78/F12 V1.00.02.03	RL78/G1A V1.00.03.01	RL78/G12 V1.02.03.03	RL78/G13 V1.03.03.05	RL78/G14 V1.01.03.06	RL78/I1A V1.01.03.01	78K0R/Fx3 V1.00.03.02	78K0R/Ix3 V1.00.03.03	78K0R/Kx3 V1.00.03.02	78K0R/Kx3-A V1.00.03.03	78K0R/Kx3-L V1.00.03.02	78K0R/Lx3 V1.00.03.03	78K0/Ix2 V 1.00.02.02	78K0/Kx2-L V 1.00.02.02
1	Cautions of the LIN-bus function of UART0, UART2, UART3, UART6 or UARTF.	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
2	Cautions of the operation for slave transmission of serial interface IICA or IIC0.	○	○	○	○	○	○	○	/	○	○	○	○	○	○	○
3	Cautions of extension code, wakeup function and multimaster of serial interface IICA or IIC0	/	/	/	-	-	-	-	/	○	○	○	○	○	○	○
4	Cautions of cooperation with the linker option	/	/	/	/	/	/	/	/	/	/	/	/	/	○	○
5	Cautions of CAN controllers	/	/	/	/	/	/	/	○	/	/	/	/	/	/	/
6	Cautions of PORT	/	/	/	-	-	-	-	/	/	/	/	/	/	/	/
7	Cautions of the SNOOZE mode of Serial array unit 1	/	/	/	-	-	-	-	/	/	/	/	/	/	/	/
8	Cautions of setup of a real-time clock	/	/	/	/	○	○	○	/	/	/	/	/	/	/	/
9	Cautions when using a DTC function	/	/	/	/	/	○	/	/	/	/	/	/	/	/	/
10	Cautions of initial function of an A/D converter	/	/	/	-	-	-	/	/	/	/	/	/	/	/	/
11	Cautions of initial function at the time of setting up UART transmission	/	/	/	-	-	-	/	/	/	/	/	/	/	/	/
12	Conversion time setup of A/D correction	/	/	-	/	/	/	/	/	/	/	/	/	/	/	/
13	Cautions of Complementary assistant PWM mode of TimerRD	/	/	/	/	/	-	/	/	/	/	/	/	/	/	/
14	Cautions of Pin Configurator	○	○	○	○	○	○	○	/	/	/	/	/	/	/	/

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

5.2 Cautions Details

5.2.1 Cautions of 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 Cautions of 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 Cautions of 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 Cautions of 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 Cautions of CAN controllers

The code generator is not supporting the CAN Controllers.
[Workaround] There is no workaround.

5.2.6 Cautions of 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 Cautions of 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.

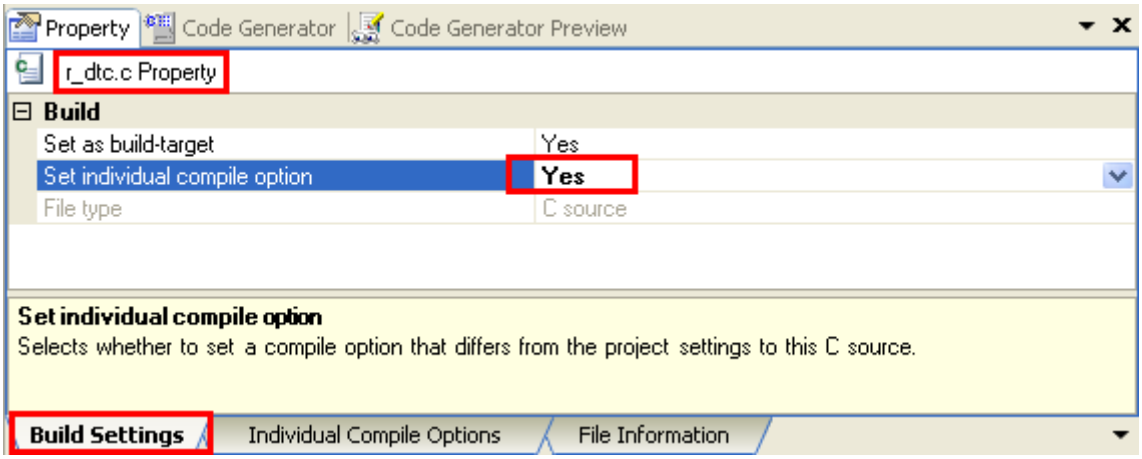
5.2.8 Cautions of a setup of a real-time clock

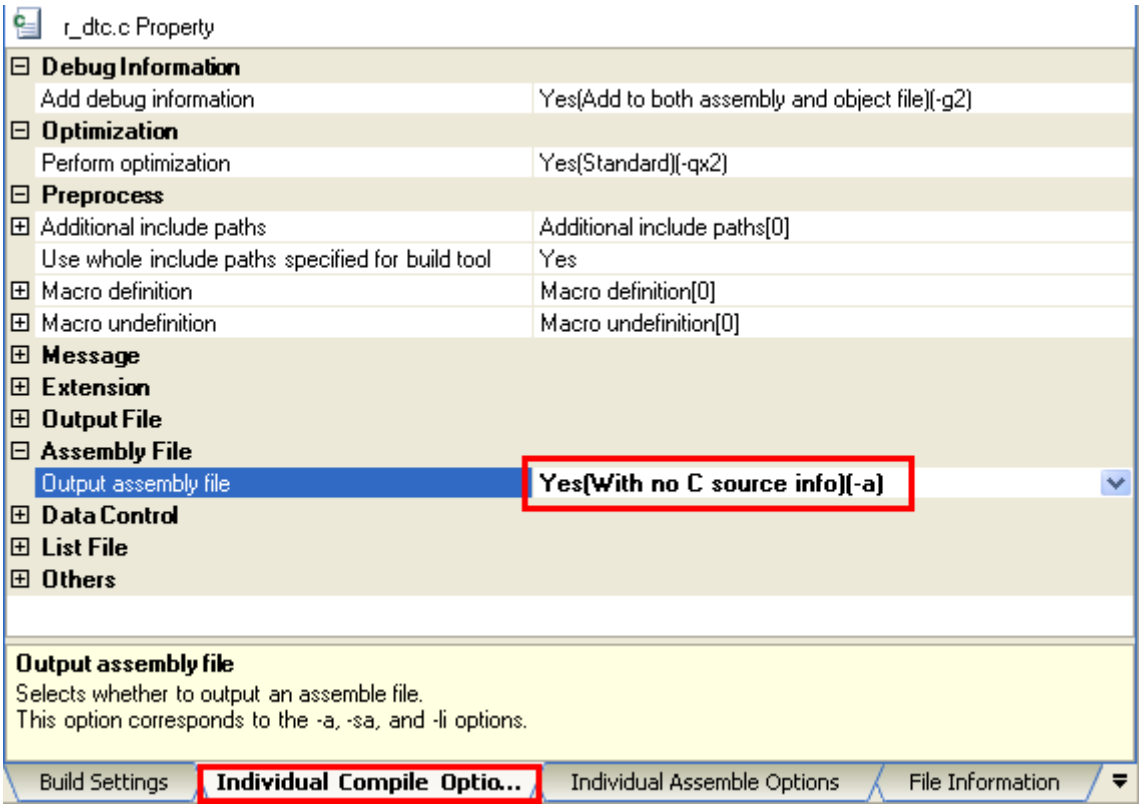
When a clock sauce is set to 15(fIL)kHz on device, clock function cannot be used. However, it is displayed on GUI that clock function seems to be used with 15(fIL)kHz. Please do not set up clock function.
[Workaround] There is no workaround

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 sauce 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 Cautions of 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. RL78 Code Generator is corrected in V1.00.06.

5.2.11 Cautions of 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. RL78 Code Generator is corrected in V1.00.06.

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 Cautions of 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. RL78 Code Generator is corrected in V1.00.06.

5.2.14 Cautions of Pin Configurator

The Pin Configurator tool of RL78 was supported from CubeSuite+V1.03.00.
However, there is the following restriction.
- 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.

In the above-mentioned case, please edit terminal information with pin configurator.

[Workaround] There is no workaround.

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/L12 V1.00.01.02	RL78/F12 V1.00.02.03	RL78/G1A V1.00.03.01	RL78/G12 V1.02.03.03	RL78/G13 V1.03.03.05	RL78/G14 V1.01.03.06	RL78/I1A V1.01.03.01	78K0R/Fx3 V1.00.03.02	78K0R/Ix3 V1.00.03.03	78K0R/Kx3 V1.00.03.02	78K0R/Kx3-A V1.00.03.03	78K0R/Kx3-L V1.00.03.02	78K0R/Lx3 V1.00.03.03	78K0/Ix2 V1.00.02.02	78K0/Kx2-L V1.00.02.02
1	Restrictions of the coding rule of MISRA-C.	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
2	Restrictions of High-speed on-chip oscillator frequency select register	○	○	○	○	○	○	/	/	/	/	/	/	/	/	/
3	Restrictions of internal low-speed or internal high-speed oscillator trimming	○	○	○	○	○	○	○	○	/	○	/	/	/	/	/
4	Restriction of a serial array unit	/	/	/	/	/	/	○	/	/	/	/	/	/	/	/

○ : Correspondence, /: Outside of function

6.2 Restrictions Details

6.2.1 Restrictions of 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 Restrictions of 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 Restrictions of internal low-speed or internal high-speed oscillator trimming

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

6.2.4 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 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.

7.1.1 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/I1A V1.01.00
	Before change	After change			
Interrupt	R_INTCn_Interrupt	r_intcn_interrupt	○	○	○
	R_KEY_Interrupt	r_key_interrupt	○	○	○
Serial	R_UARTn_Interrupt_Send	r_uartn_interrupt_send	○	○	○
	R_UARTn_Interrupt_Receive	r_uartn_interrupt_receive	○	○	○
	R_UARTn_Interrupt_Error	r_uartn_interrupt_error	○	○	○
	R_UARTn_Callback_SendEnd	r_uartn_callback_sendend	○	○	○
	R_UARTn_Callback_SendEnd	r_uartn_callback_receiveend	○	○	○
	R_UARTn_Callback_SendEnd	r_uartn_callback_error	○	○	○
	R_UARTn_Callback_SoftwareOverRun	r_uartn_callback_softwareoverrun	○	○	○
	R_CSImn_Interrupt	r_csimn_interrupt	○	○	○
	R_CSImn_Callback_SendEnd	r_csimn_callback_sendend	○	○	○
	R_CSImn_Callback_ReceiveEnd	r_csimn_callback_receiveend	○	○	○
	R_CSImn_Callback_Error	r_csimn_callback_error	○	○	○
	R_IICmn_Interrupt	r_iicmn_interrupt	○	○	○
	R_IICmn_Callback_Master_SendEnd	r_iicmn_callback_master_sendend	○	○	○
	R_IICmn_Callback_Master_ReceiveEnd	r_iicmn_callback_master_receiveend	○	○	○
	R_IICmn_Callback_Master_Error	r_iicmn_callback_master_error	○	○	○
	New addition	R_DALIn_Create	/	/	○
	New addition	R_DALIn_Start	/	/	○
	New addition	R_DALIn_Stop	/	/	○
	New addition	R_DALIn_Send	/	/	○
	New addition	R_DALIn_Receive	/	/	○
	New addition	r_dalin_interrupt_send	/	/	○
	New addition	r_dalin_interrupt_receive	/	/	○

	New addition	r_dalin_interrupt_error	/	/	○
	New addition	r_dalin_callback_sendend	/	/	○
	New addition	r_dalin_callback_receiveend	/	/	○
	New addition	r_dalin_callback_error	/	/	○
	New addition	r_dalin_callback_softwareoverrun	/	/	○
	R_IICAn_Interrupt	r_iican_interrupt	○	○	○
	R_IICAn_Callback_Master_SendEnd	r_iican_callback_master_sendend	○	○	○
	R_IICAn_Callback_Master_ReceiveEnd	r_iican_callback_master_receiveend	○	○	○
	R_IICAn_Callback_Master_Error	r_iican_callback_master_error	○	○	○
	R_IICAn_Callback_Slave_SendEnd	r_iican_callback_slave_sendend	○	○	○
	R_IICAn_Callback_Slave_ReceiveEnd	r_iican_callback_slave_receiveend	○	○	○
	R_IICAn_Callback_Slave_Error	r_iican_callback_slave_error	○	○	○
	R_IICAn_Callback_GetStopCondition	r_iican_callback_getstopcondition	○	○	○

○ : Correspondence, /: Not correspondence

Peripheral Function	API function name		RL78/G13 V1.03.00.	RL78/G14 V1.01.00	RL78/I1A V1.01.00
	Before change	After change			
A/D Converter	R_ADC_Interrupt	r_adc_interrupt	○	○	○
Timer	R_TAUm_Channeln_Interrupt	r_taum_channeln_interrupt	○	○	/
	R_TAUm_Channeln_Higher8bits_Interrupt	r_taum_channeln_higher8bits_interrupt	○	○	/
	R_TMR_RDn_Interrupt	r_tmr_rdn_interrupt	/	○	/
	R_TMR_RG0_Interrupt	r_tmr_rg0_interrupt	/	○	/
	R_TMR_RJ0_Interrupt	r_tmr_rj0_interrupt	/	○	/
	New addition	R_TMR_KB_Create	/	○	/
	New addition	R_TMR_KBm_Start	/	○	/
	New addition	R_TMR_KBm_Stop	/	○	/
	New addition	R_TMR_KBm_Set_PowerOff	/	○	/
	New addition	R_TMR_KBm_ForcedOutput_Start	/	○	/
	New addition	R_TMR_KBm_ForcedOutput_Stop	/	○	/
	New addition	R_TMR_KBm_Create_UserInit	/	○	/
	New addition	r_tmr_kbm_interrupt	/	○	/
	New addition	R_TMR_KC0_Create	/	○	/
	New addition	R_TMR_KC0_Start	/	○	/
	New addition	R_TMR_KC0_Stop	/	○	/
	New addition	R_TMR_KC0_Set_PowerOff	/	○	/
	New addition	R_TMR_KC0_Create_UserInit	/	○	/
	New addition	r_tmr_kc0_interrupt	/	○	/

○ : Correspondence, /: Not correspondence

Peripheral Function	API function name		RL78/G13 V1.03.00.	RL78/G14 V1.01.00	RL78/I1A V1.01.00
	Before change	After change			
Watchdog timer	R_WDT_Interrupt	r_wdt_interrupt	○	○	○
Real-time Clock	R_RTC_Interrupt	r_rtc_interrupt	○	○	○
	R_RTC_Callback_ConstPeriod	r_rtc_callback_constperiod	○	○	○
	R_RTC_Callback_Alarm	r_rtc_callback_alarm	○	○	○
Interval timer	R_IT_Interrupt	r_it_interrupt	○	○	○
Comparator	R_COMPn_Interrupt	r_compn_interrupt	/	○	○
DMA Controoler	R_DMACn_Interrupt	r_dmacn_interrupt	○	○	○
Voltage Detecor	R_LVD_Interrupt	r_lvd_interrupt	○	○	○
Programmable Gain Amplifier	New addition	R_PGA_Create	/	/	○
	New addition	R_PGA_Start	/	/	○
	New addition	R_PGA_Stop	/	/	○
	New addition	R_PGA_Create_UserInit	/	/	○

○ : Correspondence, /: Not correspondence

7.1.2 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_DALI*n*_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_DALI*n*_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_DALI*n*_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 00000*xx*1B: Overrun error 00000*x*1*x*B: Parity error 000001*xx*B: 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 KB*m* 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. 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.
2. 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.
3. 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.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). 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 for which it is not intended. 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 which the product is not intended by Renesas Electronics.
6. 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.
7. 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 systems manufactured by you.
8. 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.
9. 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. You should not use Renesas Electronics products or 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. When exporting the Renesas Electronics 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.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
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.

RENESAS

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: +44-1628-651-700, Fax: +44-1628-651-804

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.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

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

Colophon 2.2