

この度は、統合開発環境 CubeSuite+をご使用いただきまして、誠にありがとうございます。

この添付資料では、本製品をお使いいただく上での制限事項および注意事項等を記載しております。ご使用の前に、必ずお読みくださいますようお願い申し上げます。

## 目次

<b>第 1 章</b>	<b>対象デバイスについて</b> .....	<b>3</b>
<b>第 2 章</b>	<b>ユーザズ・マニュアルについて</b> .....	<b>11</b>
<b>第 3 章</b>	<b>アンインストール時の選択キーワード</b> .....	<b>12</b>
<b>第 4 章</b>	<b>変更点</b> .....	<b>13</b>
4.1	変更点詳細について .....	14
4.1.1	リアルタイム・クロックの出力ソース修正 .....	14
4.1.2	シリアル・アレイ・ユニットの出力ソース修正 .....	15
4.1.3	PMC レジスタ設定の追加 .....	17
4.1.4	出力ソースファイル名の変更 .....	18
4.1.5	UARTn の受信関数の出力ソース修正 .....	19
4.1.6	競合端子の制御修正 .....	19
4.1.7	A/D コンバータの GUI 表示修正 .....	19
4.1.8	A/D コンバータの変換時間修正 .....	20
4.1.9	タイマの方形波出力について .....	20
4.1.10	ポートの TTL チェックボックスについて .....	20
4.1.11	PIOR 設定について .....	20
4.1.12	TAU1 設定について .....	20
4.1.13	UART2 について .....	21
4.1.14	キー割り込みについて .....	21
4.1.15	簡易 I2C について .....	21
4.1.16	ファイル生成モードの機能追加について .....	21
4.1.17	hdwinit()関数の変更について .....	22
4.1.18	SAU 簡易 I2C に関する API の変更 .....	23
4.1.19	タイマ RD、タイマ RJ に関する修正 .....	24
4.1.20	TAU の入力パルス間隔測定に関する修正 .....	26
4.1.21	TAU1 のチャンネル 2 出力時に関する修正 .....	26
4.1.22	電源の表記に関する修正 .....	27

<b>第 5 章</b>	<b>注意事項</b> .....	<b>28</b>
5.1	注意事項一覧 .....	28
5.2	注意事項詳細について .....	29
5.2.1	UART(0, 2, 3, 6)を含む UART の LIN-bus 機能について .....	29
5.2.2	IICA または IIC0 の拡張コード、マルチマスタ、ウエイク・アップ機能について .....	29
5.2.3	IICA または IIC0 のスレーブ送信時の注意 .....	29
5.2.4	リンカ・オプションとの連携について .....	30
5.2.5	CAN コントローラ機能について .....	30
5.2.6	ポート設定について .....	30
5.2.7	シリアル・アレイ・ユニット 1 の SNOOZE モードについて .....	30
5.2.8	リアルタイム・クロックの設定について .....	30
5.2.9	DTC 機能を使用する時の注意 .....	31
5.2.10	A/D コンバータの初期化関数について .....	31
5.2.11	UART 送信を設定した場合の初期化関数について .....	32
5.2.12	A/D コンバータの変換時間設定について .....	32
5.2.13	タイマ RD の相補 PWM モードについて .....	32
5.2.14	端子配置ツールについて .....	32
5.2.15	安全機能について .....	32
5.2.16	致命的エラーについて .....	33
<b>第 6 章</b>	<b>制限事項</b> .....	<b>34</b>
6.1	制限事項一覧 .....	34
6.2	制限事項詳細 .....	34
6.2.1	MISRA-C のコーディング規約対応について .....	34
6.2.2	高速オンチップオシレータ周波数選択レジスタについて .....	34
6.2.3	高速及び低速内蔵発振器精度トリミングレジスタについて .....	34
6.2.4	シリアル・アレイ・ユニットの制限について .....	34
<b>第 7 章</b>	<b>ドキュメント訂正</b> .....	<b>35</b>
7.1	V1.00.02 で追加及び変更された API .....	35
7.1.1	RL78 コード生成機能で変更または追加された API 一覧表 .....	35
7.1.2	RL78 コード生成機能で追加された API 詳細 .....	39

## 第1章 対象デバイスについて

RL78 コード生成がサポートする対象デバイスに関しては、以下の通りです。

RL78/I1A 用コード生成 V2.00.00.04 がサポートするデバイス一覧	
ピン数	デバイス名
20pin	R5F1076C
30pin	R5F107AC, R5F107AE
32pin	R5F107BC
38pin	R5F107DE
設計資料	
資料名	資料番号
RL78/I1A ユーザーズマニュアル ハードウェア編	R01UH0169JJ0002 Rev.0.02
	R01UH0169EJ0002 Rev.0.02

RL78/G12 用コード生成 V2.00.00.07 がサポートするデバイス一覧	
ピン数	デバイス名
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
設計資料	
資料名	資料番号
RL78/G12 ユーザーズマニュアル ハードウェア編	R01UH0200JJ0100 Rev.1.00
	R01UH0200EJ0100 Rev.1.00

RL78/G13 用コード生成 V2.00.00.07 がサポートするデバイス一覧		
ピン数	デバイス名	
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	
設計資料		
	資料名	資料番号
	RL78/G13 ユーザーズマニュアル	R01UH0146JJ0100 Rev.1.00
	ハードウェア編	R01UH0146EJ0100 Rev.1.00

RL78/G14 用コード生成 V2.00.00.07 がサポートするデバイス一覧	
ピン数	デバイス名
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
設計資料	
資料名	資料番号
RL78/G14 ユーザーズマニュアル ハードウェア編	R01UH0186JJ0100 Rev.1.00
	R01UH0186EJ0100 Rev.1.00

RL78/G1A 用コード生成 V2.00.00.04 がサポートするデバイス一覧	
ピン数	デバイス名
25pin	R5F10E8A, R5F10E8C, R5F10E8D, R5F10E8E
32pin	R5F10E8A, R5F10E8C, R5F10E8D, R5F10E8E
48pin	R5F10EGA, R5F10EGC, R5F10EGD, R5F10EGE
64pin	R5F10ELC, R5F10ELD, R5F10ELE
設計資料	
資料名	資料番号
RL78/G1A ユーザーズマニュアル ハードウェア編	R01UH0305JJ0002 Rev.0.02
	R01UH0305EJ0002 Rev.0.02

RL78/F12 用コード生成 V2.00.00.07 がサポートするデバイス一覧	
ピン数	デバイス名
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
設計資料	
資料名	資料番号
RL78/F12 ユーザーズマニュアル ハードウェア編	R01UH0231JJ0003 Rev.0.03
	R01UH0231EJ0003 Rev.0.03

RL78/L12 用コード生成 V2.00.00.07 がサポートするデバイス一覧	
ピン数	デバイス名
32pin	R5F10RBC, R5F10RBA, R5F10RB8
44pin	R5F10RFC, R5F10RFA, R5F10RF8
48pin	R5F10RGC, R5F10RGA, R5F10RG8
52pin	R5F10RJC, R5F10RJA, R5F10RJ8
64pin	R5F10RLC, R5F10RLA
設計資料	
資料名	資料番号
RL78/L12 ユーザーズマニュアル ハードウェア編	R01UH0330JJ0003 Rev.0.03
	R01UH0330EJ0003 Rev.0.03

78K0R コード生成, 78K0 コード生成がサポートする対象デバイスに関しては, 以下の通りです。

78K0R/Fx3 用コード生成 V2.00.00.06 がサポートするデバイス一覧	
愛称	デバイス名
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
設計資料	
資料名	資料番号
78K0R/Fx3 ユーザーズ・マニュアル	U19145JJ1V0UD00
	U19145EJ1V0UD00

78K0R/lx3 用コード生成 V2.00.00.07 がサポートするデバイス一覧	
愛称	デバイス名
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
設計資料	
資料名	資料番号
78K0R/lx3 ユーザーズ・マニュアル	U19678JJ1V1UD00
	U19678EJ1V1UD00

78K0R/Kx3 用コード生成 V2.00.00.07 がサポートするデバイス一覧	
愛称	デバイス名
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
設計資料	
資料名	資料番号
78K0R/KE3 ユーザーズ・マニュアル	U17854JJ8V0UD00
	U17854EJ8V0UD00
78K0R/KF3 ユーザーズ・マニュアル	U17893JJ7V0UD00
	U17893EJ7V0UD00
78K0R/KG3 ユーザーズ・マニュアル	U17894JJ8V0UD00
	U17894EJ8V0UD00
78K0R/KH3 ユーザーズ・マニュアル	U18432JJ4V0UD00
	U18432EJ4V0UD00
78K0R/KJ3 ユーザーズ・マニュアル	U18417JJ3V0UD00
	U18417EJ3V0UD00

78K0R/Kx3-A 用コード生成 V2.00.00.07 がサポートするデバイス一覧	
愛称	デバイス名
78K0R/KE3-A	μPD78F1016, μPD78F1017, μPD78F1018
設計資料	
資料名	資料番号
78K0R/Kx3-A ユーザーズ・マニュアル	U19653JJ1V0UD
	U19653EJ1V0UD



78K0R/Kx3-L 用コード生成 V2.00.00.07 がサポートするデバイス一覧	
愛称	デバイス名
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
設計資料	
資料名	資料番号
78K0R/Kx3-L ユーザーズ・マニュアル	U19291JJ3V0UD00
	U19291EJ2V0UD00
78K0R/KF3-L ユーザーズ・マニュアル	U19459JJ1V0UD00
	U19459EJ1V0UD00
78K0R/KG3-L ユーザーズ・マニュアル	U19460JJ1V0UD00
	U19460EJ1V0UD00

78K0R/Lx3 用コード生成 V2.00.00.07 がサポートするデバイス一覧	
愛称	デバイス名
78K0R/LF3	μPD78F1500, μPD78F1501, μPD78F1502
78K0R/LG3	μPD78F1503, μPD78F1504, μPD78F1505
78K0R/LH3	μPD78F1506, μPD78F1507, μPD78F1508
設計資料	
資料名	資料番号
78K0R/Lx3 ユーザーズ・マニュアル	U19155JJ3V0UD
	U19155EJ3V0UD

78K0/Ix2 用コード生成 V2.00.00.04 がサポートするデバイス一覧	
愛称	デバイス名
78K0/IY2	μPD78F0740, μPD78F0741, μPD78F0742, μPD78F0750, μPD78F0751, μPD78F0752
78K0/IA2	μPD78F0743, μPD78F0744, μPD78F0753, μPD78F0754
78K0/IB2	μPD78F0745, μPD78F0746, μPD78F0755, μPD78F0756 ただし、32pin はサポートしていません。
設計資料	
資料名	資料番号
78K0/Ix2 ユーザーズ・マニュアル	U19353JJ3V0UD00
	U19353EJ3V0UD00

78K0/Kx2-L 用コード生成 V2.00.00.04 がサポートするデバイス一覧	
愛称	デバイス名
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)
設計資料	
資料名	資料番号
78K0/Kx2-L ユーザーズ・マニュアル	U19111JJ2V1UD
	U19111EJ2V1UD

## 第2章 ユーザーズ・マニュアルについて

本製品に対応したユーザーズ・マニュアルは、次のようになります。本文書と合わせてお読みください。

マニュアル名	資料番号
CubeSuite+ V1.03.00 RL78 設計編	R20UT2136JJ0100
CubeSuite+ V1.03.00 78K0R 設計編	R20UT2137JJ0100
CubeSuite+ V1.03.00 78K0 設計編	R20UT2138JJ0100
CubeSuite+ V2.00.00 メッセージ編	R20UT2448JJ0100

## 第3章 アンインストール時の選択キーワード

本製品をアンインストールする場合は、2つの方法があります。

- ・ 統合アンインストーラを使用する(CubeSuite+自体をアンインストールする)
- ・ 個別にアンインストールする(本製品のみをアンインストールする)

個別にアンインストールを行なう場合、コントロールパネルの

- ・ 「プログラムの追加と削除」(WindowsXP の場合)
- ・ 「プログラムと機能」(Windows Vista, Windows 7, Windows 8 の場合)

から、「CubeSuite+ Code Generator for RL78\_78K」を選択してください。

## 第4章 変更点

本章では、RL78, 78K0R, 78K0 コード生成 V1.00.06 から V2.00.00 への変更点について説明します。

No	内容	該当するコード生成													
		RL78/L12	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
		V2.00.00.07	V2.00.00.07	V2.00.00.04	V2.00.00.07	V2.00.00.07	V2.00.00.04	V2.00.00.06	V2.00.00.07	V2.00.00.07	V2.00.00.07	V2.00.00.07	V2.00.00.07	V2.00.00.04	V2.00.00.04
1	リアルタイム・クロックの出力ソース修正	/	/	/	-	-	-	/	/	/	/	/	/	/	/
2	シリアル・アレイ・ユニットの出力ソース修正	/	/	/	-	-	-	/	/	/	/	/	/	/	/
3	PMCレジスタ設定の追加	/	/	/	-	-	-	/	/	/	/	/	/	/	/
4	出力ソースファイル名の変更	/	/	/	-	-	-	/	/	/	/	/	/	/	/
5	UARTnの受信関数の出力ソース修正	/	/	/	-	-	-	/	/	/	/	/	/	/	/
6	競合端子の制御修正	/	/	/	/	-	/	/	/	/	/	/	/	/	/
7	A/DコンバータのGUI表示修正	/	/	/	-	-	-	/	/	/	/	/	/	/	/
8	A/Dコンバータの変換時間設定修正	/	/	-	/	/	/	/	/	/	/	/	/	/	/
9	タイマの方形波出力について	/	/	/	/	-	/	/	/	/	/	/	/	/	/
10	ポートのTTLチェックボックスについて	/	/	/	/	-	/	/	/	/	/	/	/	/	/
11	PIOR設定について	/	/	/	/	/	-	/	/	/	/	/	/	/	/
12	TAU1設定について	/	/	/	/	/	-	/	/	/	/	/	/	/	/
13	UART2について	/	/	/	/	/	-	/	/	/	/	/	/	/	/
14	キー割り込みについて	/	/	/	-	/	/	/	/	/	/	/	/	/	/
15	簡易I2Cについて	/	/	/	-	-	-	/	/	/	/	/	/	/	/
16	ファイル生成モードの機能追加について	○	○	○	○	○	○	○	○	○	○	○	○	○	○
17	hdwinit()関数の変更について	○	○	○	○	○	○	○	○	○	○	○	○	○	○
18	SAU簡易I2Cに関するAPIの変更	/	/	/	/	/	/	-	-	-	-	-	-	/	/
19	タイマRD、タイマRJに関する修正	/	/	/	/	/	-	/	/	/	/	/	/	/	/
20	TAUの入カパルス間隔測定に関する修正	-	-	-	-	-	-	-	-	-	-	-	-	/	/
21	TAU1のチャンネル2出力時に関する修正	/	/	/	/	/	/	-	/	/	/	/	/	/	/
22	電源の表記に関する修正	/	/	/	-	/	/	/	/	/	/	/	/	/	/

○：対象項目(修正項目)， -：対象外(修正済み)， /：非対象項目

## 4.1 変更点詳細について

### 4.1.1 リアルタイム・クロックの出力ソース修正

a) R\_RTC\_Set\_ConstPeriodInterruptOff( )の出力コードを修正しました。

**修正前 :**

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

**修正後 :**

```
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) R\_RTC\_Interrupt( )の出力コードを修正しました。

- ・ アラーム割り込み機能をチェックした時

**修正前 :**

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

**修正後 :**

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

- ・ 定周期割り込み機能をチェックした時

**修正前 :**

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

**修正後 :**

```
__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), b)共にRL78, 78K0R, 78K0コード生成 V1.00.02で修正済みです。

## 4.1.2 シリアル・アレイ・ユニットの出力ソース修正

- a) SAU1でサポートされていなかった下記関数を追加しました。

```
void R_SAU0_Set_SnoozeOn(void)

void R_SAU0_Set_SnoozeOff(void)
```

- b) 簡易IICを使用時の関数を修正しました。

**修正前 :**

```
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 */
}
}
```

**修正後 :**

```
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 */
}
}
```

**修正前 :**

```
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 */
}
}
```

**修正後 :**

```
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) 簡易IICの割り込みハンドラ関数を修正しました。

- ・最終バイトのNACKでエラーを出さないようにした。

**修正前:**

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

**修正後:**

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

- ・不要なコードの削除

**修正前:**

```
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;
}
```

**修正後:**

```
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), c)共にRL78, 78K0R, 78K0コード生成 V1.00.02で修正済みです。



### 4.1.3 PMCレジスタ設定の追加

各周辺の兼用端子設定コードについて、PMCレジスタの設定コードを追加しました。

[PMCレジスタ設定を追加した端子]

●対応デバイス 20,24,25,30, 32pin

P00/ANI17/TI00/TxD1

P01/ANI16/TO00/RxD1

●その他 デバイス

P02/ANI17/SO10/TxD1

P03/ANI16/SI10/RxD1/SDA10

RL78, 78K0R, 78K0コード生成 V1.00.02で修正済みです。

## 4.1.4 出力ソースファイル名の変更

RL78/G13 コード生成において V1.00.02 から出力するファイル名が変更されました。

周辺機能	ファイル名	
	変更前	変更後
クロック発生回路	r_cg.c	r_cg_cg.c
	r_cg_user.c	r_cg_cg_user.c
ポート	r_port.c	r_cg_port.c
	r_port_user.c	r_cg_port_user.c
割り込み	r_intc.c	r_cg_intc.c
	r_intc_user.c	r_cg_intc_user.c
シリアル	r_serial.c	r_cg_serial.c
	r_serial_user.c	r_cg_serial_user.c
A/Dコンバータ	r_adc.c	r_cg_adc.c
	r_adc_user.c	r_cg_adc_user.c
D/Aコンバータ	r_dac.c	r_cg_dac.c
	r_dac_user.c	r_cg_dac_user.c
タイマ	r_timer.c	r_cg_timer.c
	r_timer_user.c	r_cg_timer_user.c
ウォッチドッグ・タイマ	r_wdt.c	r_cg_wdt.c
	r_wdt_user.c	r_cg_wdt_user.c
リアルタイム・クロック	r_rtc.c	r_cg_rtc.c
	r_rtc_user.c	r_cg_rtc_user.c
インターバル・タイマ	r_it.c	r_cg_it.c
	r_it_user.c	r_cg_it_user.c
クロック出力	r_pclbuz.c	r_cg_pclbuz.c
	r_pclbuz_user.c	r_cg_pclbuz_user.c
DMAコントローラ	r_dmac.c	r_cg_dmac.c
	r_dmac_user.c	r_cg_dmac_user.c
電圧検出回路	r_lvd.c	r_cg_lvd.c
	r_lvd_user.c	r_cg_lvd_user.c

#### 4.1.5 UARTnの受信関数の出力ソース修正

UARTnの受信関数の出力ソースを修正しました。下記はUART0の場合。

```
[ 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);
}
```

修正前: gp\_uart0\_tx\_address = rx\_buf;

修正後: gp\_uart0\_rx\_address = rx\_buf;

RL78, 78K0R, 78K0コード生成 V1.00.03で修正済みです。

#### 4.1.6 競合端子の制御修正

RL78/G13の24/25ピンデバイスの簡易I2Cを設定した時の競合端子の制御を修正しました。

修正前: P17 SDA11

P30 SCL11

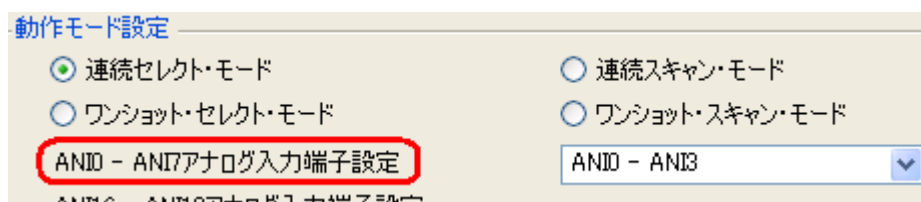
修正後: P50 SDA11

P30 SCL11

RL78, 78K0R, 78K0コード生成 V1.00.03で修正済みです。

#### 4.1.7 A/DコンバータのGUI表示修正

A/Dコンバータにおいて、アナログ入力チャンネルの数として固定値で表示されたメッセージをチャンネル数に合わせて表示するように修正しました。



RL78, 78K0R, 78K0 コード生成 V1.00.03 で修正済みです。

#### 4.1.8 A/Dコンバータの変換時間修正

RL78/G1A A/Dコンバータにおいて、変換時間が設定できないためにA/Dコンバータが使えないのを修正しました。

RL78, 78K0R, 78K0コード生成 V1.00.05で修正済みです

#### 4.1.9 タイマの方形波出力について

80ピン、100ピンおよび128ピンマイコンのタイマTAU1~TAU7への方形波出力を設定するためのコードを出力した場合、本来はTAUxを制御するTOM1およびTOL1レジスタが設定されなければなりません、TOM0およびTOL0レジスタが設定されます。

RL78, 78K0R, 78K0コード生成 V1.00.06で修正済みです

#### 4.1.10 ポートのTTLチェックボックスについて

30ピンマイコンでP10, P11にTTLを設定するチェックボックスがありません。

RL78, 78K0R, 78K0コード生成 V1.00.06で修正済みです

#### 4.1.11 PIOR設定について

端子割り当て設定で、レジスタPIOR01 および PIOR04 を1に設定するコードを出力した場合、INTP10およびINTP11の端子割り当てに以下のとおり誤りがあります。

正:	誤:
P100 → INTP10	P110 → INTP10
P110 → INTP11	P111 → INTP11

RL78, 78K0R, 78K0コード生成 V1.00.06で修正済みです

#### 4.1.12 TAU1設定について

80ピンおよび100ピンマイコンのコード生成で、タイマTAU1機能選択で "インターバル" しか選択できません。

RL78, 78K0R, 78K0コード生成 V1.00.06で修正済みです

#### 4.1.13 UART2について

80ピンマイコンのUART2設定とポート13および14以外のポートを設定するためのコードを出力した場合、ビルド時にエラーが発生します。

例：

UART2を設定し、ポート10, 11および12を出力へ設定すると以下のコード最後の"|"が不要です。

誤: PMC1 = . . . | \_80\_PMCn7\_NOT\_USE |;

正: PMC1 = . . . | \_80\_PMCn7\_NOT\_USE;

RL78, 78K0R, 78K0コード生成 V1.00.06で修正済みです

#### 4.1.14 キー割り込みについて

キー割り込みフラグおよび検出エッジの設定が正しくレジスタへ反映されません

例えば、キー割り込みフラグのリストから「使用する」を選択、検出エッジのリストから「立ち下がリエッジ」を選択すると、誤って以下のように出力します。

誤: KRCTL |= \_00\_KR\_FLAG\_UNUSED;

KRCTL |= \_01\_KR\_EDGE\_RISING;

正: KRCTL |= \_01\_KR\_FLAG\_USED;

KRCTL |= \_00\_KR\_EDGE\_FALLING;

RL78, 78K0R, 78K0コード生成 V1.00.06で修正済みです

#### 4.1.15 簡易I2Cについて

簡易I2Cで受信バイトを1にした場合、正常に動作しません。

RL78, 78K0R, 78K0コード生成 V1.00.06で修正済みです

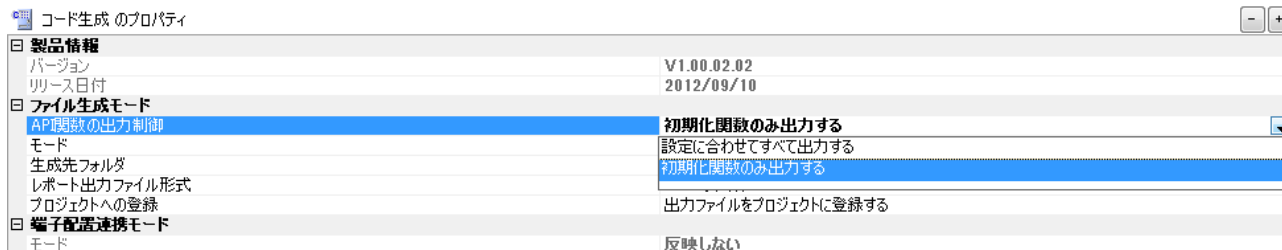
#### 4.1.16 ファイル生成モードの機能追加について

コード生成のプロパティで、「API関数の出力制御」を追加しました。

“設定に合わせて全て出力する” → 従来とおり、GUIの設定に合わせて必要なAPIを出力します。

“初期化関数のみ出力する” → GUIの設定にかかわらず、初期化関数(Create関数)のみ出力します。  
割り込み関数などは、ユーザが自由に設定してください。

RL78, 78K0R, 78K0コード生成 V1.00.06で追加されました



#### 4.1.17 hdwinit()関数の変更について

hdwinit(), main()関数の初期コードを変更しました。

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

これを下記のように変更しました。hdwinit内では割り込み許可しません。

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

割り込み許可を main()関数内で行うようにしました。

```
/*
*****
* 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 */
}
```

旧プロジェクトを使ってコード生成を行う場合、main関数内の変数定義でエラーになることがあります。

旧プロジェクト

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

旧プロジェクトをCubeSuite+V1.03.00でコード生成した場合

```
void main(void)
{
    R_MAIN_UserInit();
    /* Start user code. Do not edit comment generated here */
    char c;          ← エラーとなる
    while (1U)
    {
        ...
    }
}
```

その場合は、中括弧{ } を追加して修正してください。

```
void main(void)
{
    R_MAIN_UserInit();
    /* Start user code. Do not edit comment generated here */
    {                ← 中括弧を追加
        char c;      ← エラーにならない
        while (1U)
        {
            ...
        }
    }                ← 中括弧を追加
}
```

RL78, 78K0R, 78K0コード生成 V1.00.06で変更されました

#### 4.1.18 SAU簡易I2Cに関するAPIの変更

SAUの簡易IICのR\_IICmn\_StartCondition、R\_IICmn\_StopCondition関数で、I2C規格のスタート・コンディション、ストップ・コンディションの生成に必要なウェイトを確保するコードを追加しました。

RL78, 78K0R, 78K0コード生成 V2.00.00で修正済みです

#### 4.1.19 タイマRD、タイマRJ0に関する修正と新たな制限について

##### a) タイマRDに関する修正

RL78/G14デバイスで、高速オンチップオシレータクロックに64MHzを指定したとき、タイマRDのPWMモードを正しい周期、デューティ比で設定できません。また、PWMを使用したときの兼用端子はPWM以外の機能を使えないことを知らせるためにポート設定画面には "!" が表示されますが、タイマRDが使用するポートに "!" が表示されません。

##### b) タイマRJ0に関する修正

RL78/G14デバイスのタイマRJ0パルス周期測定モードの不具合を修正しました。

例えば、カウントソース任意に設定、TRJIO0極性を任意に設定、カウント値設定で0xffffを指定してコード生成を行ったときに出力される割り込みハンドラを下記のように修正しました。

[出力されたソース]

```
__interrupt static void r_tmr_rj0_interrupt(void)
{
    if ((TRJCR0 & _20_TMRJ_UNDERFLOW_OCCUR) != 0U)
    {
        g_tmrj0_underflow_count += 1U;
        TRJCR0 &= (uint8_t)~_20_TMRJ_UNDERFLOW_OCCUR;
    }

    if ((TRJCR0 & _10_TMRJ_ACTIVE_EDGE_UNRECEIVED) != 0U)
    {
        g_tmrj0_width = (uint32_t)(g_tmrj0_trj_count - TRJ0 + 1U +
            (g_tmrj0_underflow_count * (_FFFF_TMRJ_TRJ0_VALUE + 1U)));
        g_tmrj0_trj_count = (uint32_t)TRJ0;
        g_tmrj0_underflow_count = 0U;
        TRJCR0 &= (uint8_t)~_10_TMRJ_ACTIVE_EDGE_UNRECEIVED;
    }
}
```

[修正した部分]

```
g_tmrj0_width = (uint32_t)(_FFFF_TMRJ_TRJ0_VALUE - TRJ0 + 1U +
    (g_tmrj0_underflow_count * (_FFFF_TMRJ_TRJ0_VALUE + 1U)));
```

修正する前の計算式にある g\_tmrj0\_trj\_count の部分を "設定したカウント値" にしました。

カウント値 0xffff の場合は \_FFFF\_TMRJ\_TRJ0\_VALUE という名前を設定します。

上記 a), b)共にRL78, 78K0R, 78K0コード生成 V2.00.00で修正済みです



## c) タイマRJ0に関する制限の追加

RL78/G14デバイスのタイマRJ0パルス幅測定モードの割り込み関数に不具合があります。

例えば、カウントソースfCLK、TRJIO0極性”L”レベル幅を測定、カウント値設定で100を指定してコード生成を行ったときに出力される割り込みハンドラは下記のように修正してください。

[出力されたソース]

```
__interrupt static void r_tmr_rj0_interrupt(void)
{
    if ((TRJCR0 & _20_TMRJ_UNDERFLOW_OCCUR) != 0U)
    {
        g_tmrj0_underflow_count += 1U;
        TRJCR0 &= (uint8_t)~_20_TMRJ_UNDERFLOW_OCCUR;
    }

    if ((TRJCR0 & _10_TMRJ_ACTIVE_EDGE_UNRECEIVED) != 0U)
    {
        g_tmrj0_width = (uint32_t)(_0064_TMRJ_TRJ0_VALUE - TRJ0 + 1U +
            (g_tmrj0_underflow_count * (_0064_TMRJ_TRJ0_VALUE + 1U)));
        g_tmrj0_underflow_count = 0U;
        TRJCR0 &= (uint8_t)~_10_TMRJ_ACTIVE_EDGE_UNRECEIVED;
    }
    /* Start user code. Do not edit comment generated here */
    /* End user code. Do not edit comment generated here */
}
```

[修正する部分]

**volatile uint32\_t g\_tmrj0\_trj\_count = 0U;** (グローバル変数を追加する)

```
__interrupt static void r_tmr_rj0_interrupt(void)
{
    if ((TRJCR0 & _20_TMRJ_UNDERFLOW_OCCUR) != 0U)
    {
        g_tmrj0_underflow_count += 1U;
        TRJCR0 &= (uint8_t)~_20_TMRJ_UNDERFLOW_OCCUR;
    }

    if ((TRJCR0 & _10_TMRJ_ACTIVE_EDGE_UNRECEIVED) != 0U)
    {
        g_tmrj0_width = (uint32_t)(g_tmrj0_trj_count - TRJ0 + 1U +
            (g_tmrj0_underflow_count * (_0064_TMRJ_TRJ0_VALUE + 1U)));
        g_tmrj0_trj_count = (uint32_t)TRJ0;
        g_tmrj0_underflow_count = 0U;
        TRJCR0 &= (uint8_t)~_10_TMRJ_ACTIVE_EDGE_UNRECEIVED;
    }
    /* Start user code. Do not edit comment generated here */
    /* End user code. Do not edit comment generated here */
}
```

c) は次バージョンで改修する予定です。

#### 4.1.20 TAUの入力パルス間隔測定に関する修正

TAUの入力パルス間隔測定機能、ハイ/ロウ幅測定機能を使用時、割り込み関数r\_taux\_channelx\_interrupt()で正しい測定値を取得できなかったのを修正しました。

RL78, 78K0R, 78K0コード生成 V2.00.00で修正済みです

#### 4.1.21 TAU1のチャンネル2出力時に関する修正

TAU1のチャンネル2で出力を伴う機能(方形波出力等)を使用時、TO12(P4.6)の端子設定コードを修正しました。

修正前 :

```
/* Set TO12 pin */  
P4   &= 0xEFU;  
PM4  &= 0xEFU;  
TOS1 |= 0x04U;
```

修正後 :

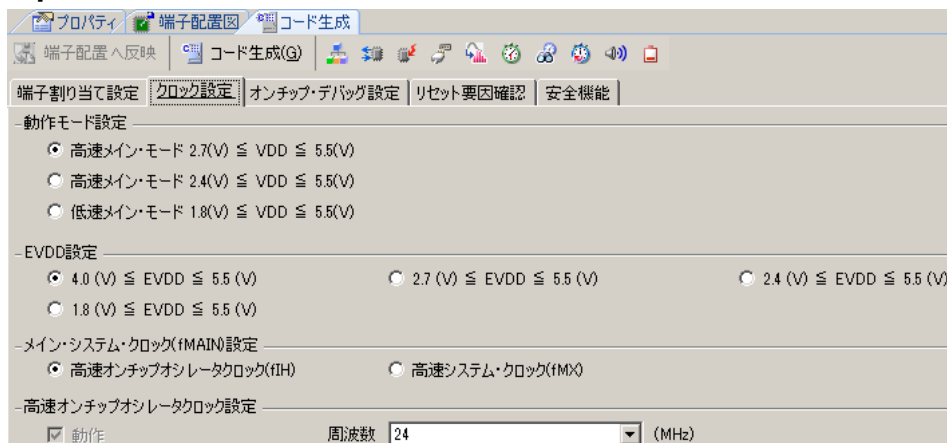
```
/* Set TO12 pin */  
P4   &= 0xBFU;  
PM4  &= 0xBFU;  
TOS1 |= 0x04U;
```

RL78, 78K0R, 78K0コード生成 V2.00.00で修正済みです

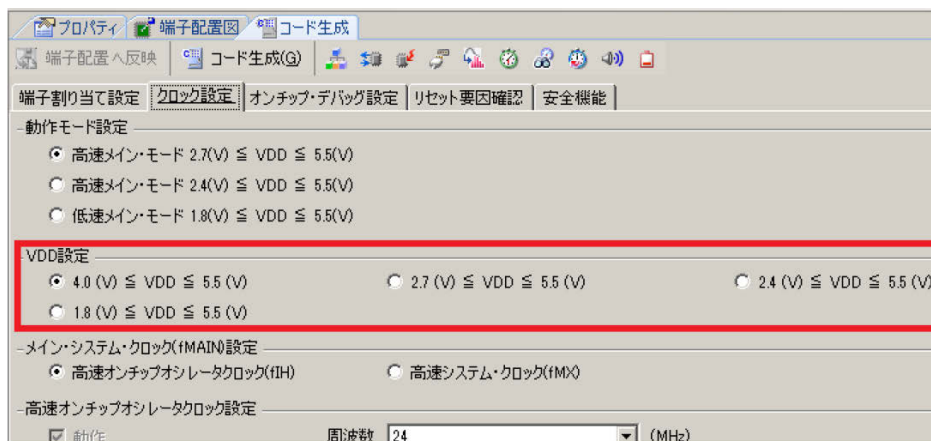
#### 4.1.22 電源の表記に関する修正

RL78/G12デバイスにはEVDDはありませんが、GUI上で表記が残っていたのを修正しました。

[修正前]



[修正後]



RL78, 78K0R, 78K0コード生成 V2.00.00で修正済みです

## 第5章 注意事項

本章では、RL78, 78K0R, 78K0 コード生成 の注意事項について説明します。

### 5.1 注意事項一覧

No	内容	該当するコード生成														
		RL78/L12	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
		V2.00.00.07	V2.00.00.07	V2.00.00.04	V2.00.00.07	V2.00.00.07	V2.00.00.07	V2.00.00.04	V2.00.00.06	V2.00.00.07	V2.00.00.07	V2.00.00.07	V2.00.00.07	V2.00.00.07	V2.00.00.04	V2.00.00.04
1	UART(0,2,3,6,F)を含むUARTのLIN-bus機能について	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
2	IICAまたはIIC0の拡張コード、マルチマスタ、ウエイク・アップ機能について	○	○	○	○	○	○	○	/	○	○	○	○	○	○	○
3	IICAまたはIIC0のスレーブ送信時の注意	/	/	/	-	-	-	-	/	○	○	○	○	○	○	○
4	リンカ・オプションとの連携について	/	/	/	/	/	/	/	/	/	/	/	/	/	/	○
5	CANコントローラ機能について	/	/	/	/	/	/	/	○	/	/	/	/	/	/	/
6	ポート設定について	/	/	/	-	-	-	-	/	/	/	/	/	/	/	/
7	シリアル・アレイ・ユニット1のSNOOZEモードについて	/	/	/	-	-	-	-	/	/	/	/	/	/	/	/
8	リアルタイム・クロックの設定について	/	/	/	/	○	○	○	/	/	/	/	/	/	/	/
9	DTC機能を使用する時の注意	/	/	/	/	/	○	/	/	/	/	/	/	/	/	/
10	A/Dコンバータの初期化関数について	/	/	/	-	-	-	/	/	/	/	/	/	/	/	/
11	UART送信を設定した場合の初期化関数について	/	/	/	-	-	-	/	/	/	/	/	/	/	/	/
12	A/Dコンバータの変換時間設定について	/	/	-	/	/	/	/	/	/	/	/	/	/	/	/
13	タイマRDの相補PWMモードについて	/	/	/	/	/	-	/	/	/	/	/	/	/	/	/
14	端子配置ツールについて	○	○	○	○	○	○	○	/	/	/	/	/	/	/	/
15	安全機能について	○	○	○	○	○	○	○	/	/	/	/	/	/	/	/
16	致命的エラーについて	/	/	/	/	/	/	/	○	/	/	/	/	/	/	/

○：対象項目， -：非対象項目(修正済み)， /：機能対象外

## 5.2 注意事項詳細について

### 5.2.1 UART(0, 2, 3, 6)を含むUARTのLIN-bus機能について

コード生成は、シリアル・インタフェースUART(0, 2, 3, 6,F)を含むUARTのLIN- bus機能に対応していません。

【回避策】 ありません。

### 5.2.2 IICAまたはIIC0の拡張コード、マルチマスタ、ウエイク・アップ機能について

コード生成は、シリアル・インタフェースIICAまたはIIC0の拡張コード、マルチマスタ、ウエイク・アップ機能に対応していません。

【回避策】 ありません。

### 5.2.3 IICAまたはIIC0のスレーブ送信時の注意

スレーブ送信時にマスタ受信が最終データ受信後にACKを返さない場合、実際のスレーブ送信処理が終了していてもエラーAPIのIICA\_SlaveErrorCallback(MD\_NACK)がコールされます。そのため、プログラムが正常終了しません。

【回避策】

通信相手のマスタが最終データ受信後にACKを返さない場合はIICA\_SlaveHandler内を下記のように変更してください。(最終データ送信後にACKをチェックしないようにします。下図はシリアル・インタフェースがIICAの場合)

```
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))

【回避策】 RL78 コード生成の V1.00.02 で修正済みです。

### 5.2.4 リンカ・オプションとの連携について

リンカ・オプション(ビルド・ツール → プロパティ → リンカ・オプションタブ)の項目にある、デバイス → 「ユーザ・オプション・バイトを設定する」とコード生成の設定は連携をしていません。

【回避策】      ありません。

### 5.2.5 CANコントローラ機能について

コード生成は、CAN機能に対応していません。

【回避策】      ありません。

### 5.2.6 ポート設定について

RL78/G13(R5F100LJ, R5F100LK, R5F100LL)のポート設定に注意点があります。

・ P43, P52, P53, P54の設定項目にある「TTLバッファ」及び「N-ch」を使用しないでください

【回避策】      RL78 コード生成は、V1.00.02 で修正済みです。

### 5.2.7 シリアル・アレイ・ユニット1のSNOOZEモードについて

RL78/G13のコード生成は、シリアル・アレイ・ユニット1のSNOOZEモードに対応していません。

【回避策】      RL78 コード生成の V1.00.02 で修正済みです。

### 5.2.8 リアルタイム・クロックの設定について

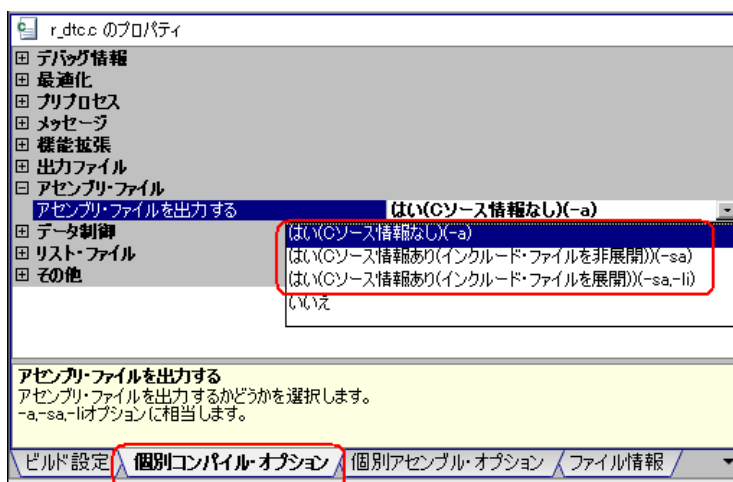
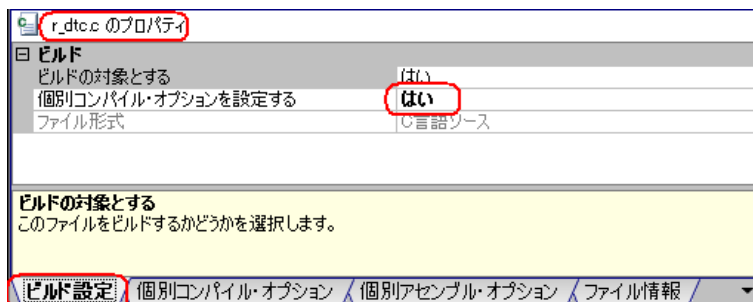
デバイスでは、クロック・ソースを15(fIL)kHzにした時、時計機能は使えません。しかし、GUI上では設定できるように表示されます。この場合、時計機能の設定を行わないでください。

【回避策】      ありません。

### 5.2.9 DTC機能を使用する時の注意

DTCを使用する場合、ビルド時に下記の個別オプションを設定してください。出力ソース「r\_cg\_dtc.c」でDTC用にDATAセクションを追加しています。個別オプションを設定しないと、ワーニングが表示され、(CC78K0R warning W0837: Output assembler source file, not object file)

オブジェクト・ファイルが生成されません。



【回避策】 ありません。

### 5.2.10 A/Dコンバータの初期化関数について

ポート2をA/Dコンバータと競合しない設定にした後に、A/Dコンバータでアナログ入力端子を設定した場合の初期化関数に誤りがあります。R\_ADC\_Create()で出力されるコード“PM2 |= 0x??;”の0x??の値に誤りがあります。

【回避策】 ポート2の設定を行う前にA/Dコンバータを設定してください。最後にポート2の設定をすると正しい値が反映されます。RL78 コード生成のV1.00.06で修正済みです。

### 5.2.11 UART送信を設定した場合の初期化関数について

UART送信のみを選択した場合の初期化関数 R\_UARTn\_Create()にSDRmnレジスタ設定のコードが不足します。

【回避策】      ありません。RL78 コード生成の V1.00.06 で修正済みです。

### 5.2.12 A/Dコンバータの変換時間設定について

RL78/G1AのA/Dコンバータの変換時間が設定できません。そのためA/Dコンバータが使用できません。

【回避策】      ありません。RL78 コード生成の V1.00.05 で修正済みです。

### 5.2.13 タイマRDの相補PWMモードについて

RL78/G14のクロック設定で高速システムクロックを使いタイマRD相補PWMモードを使用した時にはオプションバイトの設定を変更する必要があります。RL78/G14 ユーザーズマニュアル ハードウェア編 R01UH0186JJ0100 Rev.1.00の519ページ 注意事項1を参照してください。

【回避策】      ありません。RL78 コード生成の V1.00.06 で修正済みです

### 5.2.14 端子配置ツールについて

CubeSuite+V1.03.00よりRL78の端子配置ツールをサポートしました。しかし、下記の制限があります。

- ・コード生成から端子配置への反映を実行しても反映されない端子がある
- ・コード生成でPIOR機能を使い値を設定しても、端子配置へ反映されない

上記の場合、端子配置ツールで端子情報を編集してください

【回避策】      ありません。次回リリースで修正することを検討しています。

### 5.2.15 安全機能について

安全機能のRAM/パリティ・エラー検出機能については未対応です。

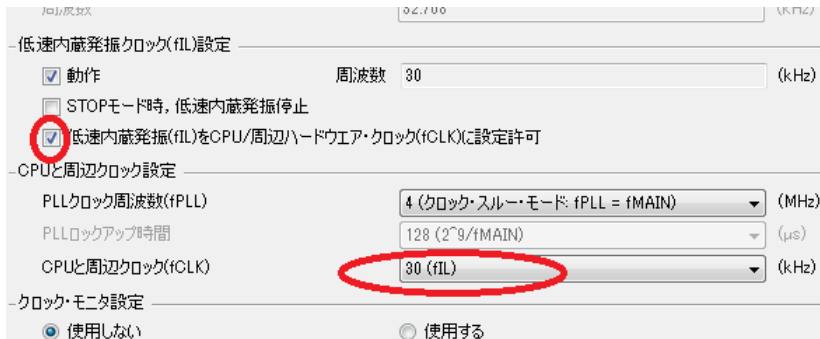
【回避策】      ありません。



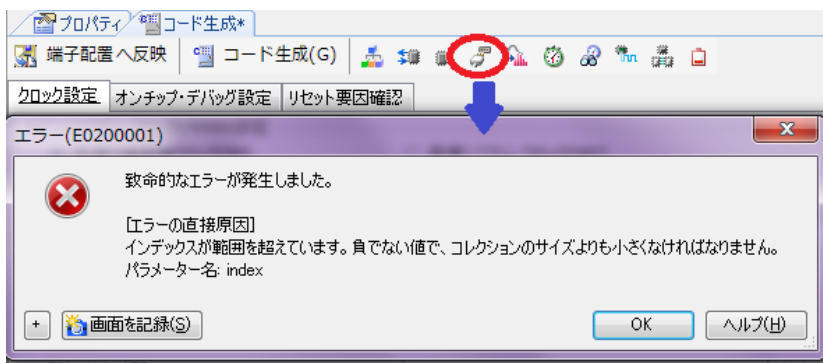
## 5.2.16 致命的エラーについて

78K0R/Fx3のコード生成で下記の操作を行った場合、致命的なエラーが発生します。

## a) クロック

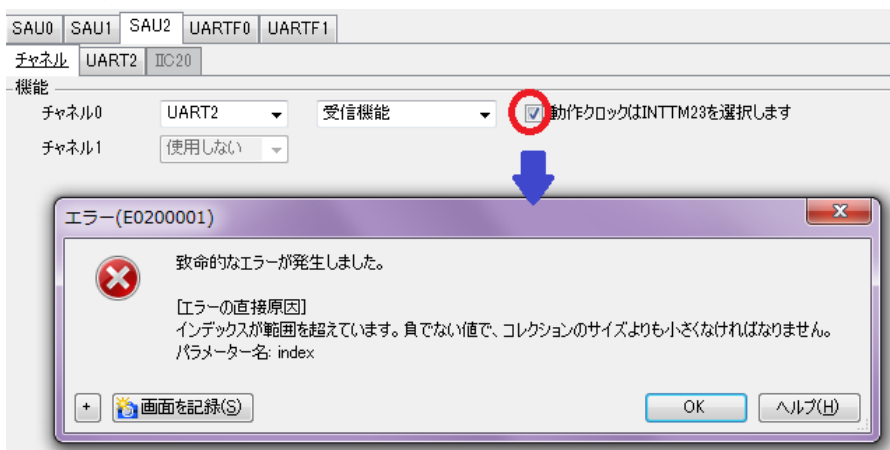


・シリアルパネルを選択すると例外が発生



## b) タイマ

- ・タイマTAU2のチャンネル3をなんらかのモードで割り込みを使用
- ・シリアルSAU0,SAU1,SAU2のいずれかのシリアルを、「動作クロックはINTTM23を選択します」にチェックすると例外が発生



上記 a), b)共に回避策はありません。次バージョンで改修の予定です。

## 第6章 制限事項

本章では、RL78, 78K0R, 78K0 コード生成 の制限事項について説明します。

### 6.1 制限事項一覧

No.	内容	該当するコード生成													
		RL78/L12	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/Kx2-L
1	MISRA-Cのコーディング規約対応について	○	○	○	○	○	○	○	○	○	○	○	○	○	○
2	高速オンチップオシレータ周波数選択レジスタについて	○	○	○	○	○	○	/	/	/	/	/	/	/	/
3	低速及び高速内蔵発振器精度トリミングレジスタについて	○	○	○	○	○	○	○	/	○	/	/	/	/	/
4	シリアル・アレイ・ユニットの制限について	/	/	/	/	/	/	○	/	/	/	/	/	/	/

○：対象項目， /：機能対象外

### 6.2 制限事項詳細

#### 6.2.1 MISRA-Cのコーディング規約対応について

コード生成から出力されるソース・コードは、自動車向け組み込みC言語用ガイドラインMISRA-Cのコーディング規約に対応していません。

#### 6.2.2 高速オンチップオシレータ周波数選択レジスタについて

コード生成は、高速オンチップオシレータ周波数選択レジスタの設定に対応していません。

#### 6.2.3 高速及び低速内蔵発振器精度トリミングレジスタについて

コード生成は、高速及び低速内蔵発振器精度トリミングレジスタの設定に対応していません。

#### 6.2.4 シリアル・アレイ・ユニットの制限について

コード生成は、1線UARTモード及びDMX512通信の設定に対応していません。

## 第7章 ドキュメント訂正

本章では、CubeSuite+のドキュメントの訂正について説明します。

また、Help にも同様の内容が記載されておりますので、ドキュメントと同様に読み替えてください。

### 7.1 V1.00.02で追加及び変更されたAPI

前版のRL78 設計編（資料番号：R20UT0548JJ0100）に変更及び追加されたAPIについて説明します。重要な変更ですので、本文書でも掲載します。

#### 7.1.1 RL78コード生成機能で変更または追加されたAPI一覧表

周辺機能	API関数名		RL78/G13 V1.03.00	RL78/G14 V1.01.00	RL78/1A V1.01.00
	変更前	変更後			
割り込み	R_INTCn_Interrupt	<a href="#">r_intcn_interrupt</a>	○	○	○
	R_KEY_Interrupt	<a href="#">r_key_interrupt</a>	○	○	○
シリアル	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>	○	○	○
	新規追加	<a href="#">R_DALIn_Create</a>	/	/	○
	新規追加	<a href="#">R_DALIn_Start</a>	/	/	○
	新規追加	<a href="#">R_DALIn_Stop</a>	/	/	○
	新規追加	<a href="#">R_DALIn_Send</a>	/	/	○
	新規追加	<a href="#">R_DALIn_Receive</a>	/	/	○

新規追加	r_dalin_interrupt_send	/	/	○
新規追加	r_dalin_interrupt_receive	/	/	○
新規追加	r_dalin_interrupt_error	/	/	○
新規追加	r_dalin_callback_sendend	/	/	○
新規追加	r_dalin_callback_receiveend	/	/	○
新規追加	r_dalin_callback_error	/	/	○
新規追加	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	○	○	○

○ : 対象項目, / : 機能対象外

周辺機能	API関数名		RL78/G13 V1.03.00.	RL78/G14 V1.01.00	RL78/11A V1.01.00
	変更前	変更後、または新規			
A/Dコンバータ	R_ADC_Interrupt	r_adc_interrupt	○	○	○
タイマ	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	/	○	/
	新規追加	R_TMR_KB_Create	/	○	/
	新規追加	R_TMR_KBm_Start	/	○	/
	新規追加	R_TMR_KBm_Stop	/	○	/
	新規追加	R_TMR_KBm_Set_PowerOff	/	○	/
	新規追加	R_TMR_KBm_ForcedOutput_Start	/	○	/
	新規追加	R_TMR_KBm_ForcedOutput_Stop	/	○	/
	新規追加	R_TMR_KBm_Create_UserInit	/	○	/
	新規追加	r_tmr_kbm_interrupt	/	○	/
	新規追加	R_TMR_KC0_Create	/	○	/
	新規追加	R_TMR_KC0_Start	/	○	/
	新規追加	R_TMR_KC0_Stop	/	○	/
	新規追加	R_TMR_KC0_Set_PowerOff	/	○	/
	新規追加	R_TMR_KC0_Create_UserInit	/	○	/
	新規追加	r_tmr_kc0_interrupt	/	○	/

○ : 対象項目, / : 機能対象外

周辺機能	API関数名		RL78/G13 V1.03.00.	RL78/G14 V1.01.00	RL78/11A V1.01.00
	変更前	変更後、または新規			
ウォッチドック・タイマ	R_WDT_Interrupt	<a href="#">r_wdt_interrupt</a>	○	○	○
リアルタイム・クロック	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>	○	○	○
インターバル・タイマ	R_IT_Interrupt	<a href="#">r_it_interrupt</a>	○	○	○
コンパレータ	R_COMPn_Interrupt	<a href="#">r_compn_interrupt</a>	/	○	○
DMA コントローラ	R_DMAn_Interrupt	<a href="#">r_dmacn_interrupt</a>	○	○	○
電圧検出回路	R_LVD_Interrupt	<a href="#">r_lvd_interrupt</a>	○	○	○
プログラマブル・ゲイン・アンプ	新規追加	<a href="#">R_PGA_Create</a>	/	/	○
	新規追加	<a href="#">R_PGA_Start</a>	/	/	○
	新規追加	<a href="#">R_PGA_Stop</a>	/	/	○
	新規追加	<a href="#">R_PGA_Create_UserInit</a>	/	/	○

○ : 対象項目, / : 機能対象外

## 7.1.2 RL78コード生成機能で追加されたAPI詳細

### 【API名】 R\_DALIn\_Create

シリアル・インタフェース（DALI）用チャンネルの初期化処理を行います。

備考：本 API 関数は、R\_SAUm\_Create の内部関数として位置づけられているため、通常、ユーザの処理プログラムから呼び出す必要はありません。

#### 【所属】

r\_cg\_serial.c

#### 【指定形式】

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

備考: n はチャンネル番号を意味します。

#### 【引数】

なし

#### 【戻り値】

なし

### 【API名】 R\_DALIn\_Start

DALI 通信を待機状態にします。

#### 【所属】

r\_cg\_serial.c

#### 【指定形式】

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

備考: n はチャンネル番号を意味します。

#### 【引数】

なし

#### 【戻り値】

なし

### 【API名】 R\_DALIn\_Stop

DALI 通信を終了します。

#### 【所属】

r\_cg\_serial.c

#### 【指定形式】

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

備考: n はチャンネル番号を意味します。

#### 【引数】

なし

#### 【戻り値】

なし

**【API 名】 R\_DALIn\_Send**

データの DALI 送信を開始します。

- 備考 1. 本 API 関数では、引数 txbuf で指定されたバッファから 1 バイト単位の DALI 送信を引数 txnum で指定された回数だけ繰り返し行います。
2. DALI 送信を行う際には、本 API 関数の呼び出し以前に R\_DALIn\_Start を呼び出す必要があります。

**[所属]**

r\_cg\_serial.c

**[指定形式]**

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

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

備考: n はチャンネル番号を意味します。

**[引数]**

I/O	引数	説明
I	uint8_t *txbuf;	送信するデータを格納したバッファへのポインタ
I	uint16_t txnum;	送信するデータの総数

**[戻り値]**

マクロ	説明
MD_OK	正常終了
MD_ARGERROR	引数の指定が不正

**【API 名】 R\_DALIn\_Receive**

データの DALI 受信を開始します。

- 備考 1. 本 API 関数では、1 バイト単位の DALI 受信を引数 rxnum で指定された回数だけ繰り返し行い、引数 rxbuf で指定されたバッファに格納します。
2. 実際の DALI 受信は、本 API 関数の呼び出し後、R\_DALIn\_Start を呼び出すことにより開始されます。

**[所属]**

r\_cg\_serial.c

**[指定形式]**

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

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

備考: n はチャンネル番号を意味します。

**[引数]**

I/O	引数	説明
0	uint8_t *rxbuf;	受信するデータを格納したバッファへのポインタ
I	uint16_t rxnum;	受信するデータの総数

**[戻り値]**

マクロ	説明
MD_OK	正常終了
MD_ARGERROR	引数の指定が不正



**【API 名】 r\_dalin\_interrupt\_send**

DALI 送信完了割り込み INTSTDLn の発生に伴う処理を行います。

備考：本 API 関数は、DALI 送信完了割り込み INTSTDLn に対応した割り込み処理として呼び出されます。

## [所属]

r\_cg\_serial\_user.c

## [指定形式]

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

備考: n はチャンネル番号を意味します。

## [引数]

なし

## [戻り値]

なし

**【API 名】 r\_dalin\_interrupt\_receive**

DALI 受信完了割り込み INTSRDLn の発生に伴う処理を行います。

備考：本 API 関数は、DALI 受信完了割り込み INTSRDLn に対応した割り込み処理として呼び出されます。

## [所属]

r\_cg\_serial\_user.c

## [指定形式]

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

備考: n はチャンネル番号を意味します。

## [引数]

なし

## [戻り値]

なし

**【API 名】 r\_dalin\_interrupt\_error**

DALI 受信エラー割り込み INTSREDLn の発生に伴う処理を行います。

備考：本 API 関数は、DALI 受信エラー割り込み INTSREDLn に対応した割り込み処理として呼び出されます。

## [所属]

r\_cg\_serial\_user.c

## [指定形式]

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

備考: n はチャンネル番号を意味します。

## [引数]

なし

## [戻り値]

なし

**【API 名】 r\_dalin\_callback\_sendend**

DALI 送信完了割り込み INTSTDLn の発生に伴う処理を行います。

備考：本 API 関数は、DALI 送信完了割り込み INTSTDLn に対応した割り込み処理 r\_dalin\_interrupt\_send のコールバック・ルーチン (R\_DALIn\_Send の引数 txnum で指定された数のデータ送信が完了した際の処理) として呼び出されます。

## [所属]

r\_cg\_serial\_user.c

## [指定形式]

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

備考: n はチャンネル番号を意味します。

## [引数]

なし

## [戻り値]

なし

**【API 名】 r\_dalin\_callback\_receiveend**

DALI 受信完了割り込み INTSRDLn の発生に伴う処理を行います。

備考：本 API 関数は、DALI 受信完了割り込み INTSRDLn に対応した割り込み処理 r\_dalin\_interrupt\_receive のコールバック・ルーチン (R\_DALIn\_Receive の引数 rxnum で指定された数のデータ受信が完了した際の処理) として呼び出されます。

## [所属]

r\_cg\_serial\_user.c

## [指定形式]

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

備考: n はチャンネル番号を意味します。

## [引数]

なし

## [戻り値]

なし

**【API 名】 r\_dalin\_callback\_error**

DALI 受信エラー割り込み INTSREDLn の発生に伴う処理を行います。

備考：本 API 関数は、DALI 受信エラー割り込み INTSREDLn に対応した割り込み処理 r\_dalin\_interrupt\_error のコールバック・ルーチンとして呼び出されます。

**[所属]**

r\_cg\_serial\_user.c

**[指定形式]**

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

```
void r_dalin_callback_error ( uint8_t err_type );
```

備考: n はチャンネル番号を意味します。

**[引数]**

I/O	引数	説明
0	uint8_t err_type;	DALI受信エラー割り込みの発生要因 00000xx1B : オーバラン・エラー 00000x1xB : パリティ・エラー 000001xxB : フレーミング・エラー

**[戻り値]**

なし

**【API 名】 r\_dalin\_callback\_softwareoverrun**

オーバラン・エラーの検出に伴う処理を行います。

備考：本 API 関数は、DALI 受信完了割り込み INTSRDLn に対応した割り込み処理 r\_dalin\_interrupt\_receive のコールバック・ルーチン (R\_DALIn\_Receive の引数 rxnum で指定された数以上のデータを受信した際の処理) として呼び出されます。

**[所属]**

r\_cg\_serial\_user.c

**[指定形式]**

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

```
void r_dalin_callback_softwareoverrun ( uint16_t rx_data );
```

備考: n はチャンネル番号を意味します。

**[引数]**

I/O	引数	説明
0	uint16_t rx_data;	受信したデータ (R_DALIn_Receive の引数 rxnum で指定された数以上に受信したデータ)

**[戻り値]**

なし

**【API 名】 R\_TMR\_KB\_Create**

16 ビット・タイマ KBm の機能を制御するうえで必要となる初期化処理を行います。

[所属]

r\_cg\_timer.c

[指定形式]

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

[引数]

なし

[戻り値]

なし

**【API 名】 R\_TMR\_KBm\_Start**

16 ビット・タイマ KBm のカウント処理を開始します。

[所属]

r\_cg\_timer.c

[指定形式]

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

備考: m はユニット番号を意味します。

[引数]

なし

[戻り値]

なし

**【API 名】 R\_TMR\_KBm\_Stop**

16 ビット・タイマ KBm のカウント処理を終了します。

[所属]

r\_cg\_timer.c

[指定形式]

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

備考: m はユニット番号を意味します。

[引数]

なし

[戻り値]

なし

**【API 名】 R\_TMR\_KBm\_Set\_PowerOff**

16 ビット・タイマ KBm に対するクロック供給を停止します。

備考：本 API 関数の呼び出しにより、16 ビット・タイマ KBm はリセット状態へと移行します。このため、本 API 関数の呼び出し後、制御レジスタへの書き込みは無視されます。

## [所属]

r\_cg\_timer.c

## [指定形式]

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

備考: m はユニット番号を意味します。

## [引数]

なし

## [戻り値]

なし

**【API 名】 R\_TMR\_KBm\_ForcedOutput\_Start**

強制出力停止機能に使用するトリガ信号の入力を許可します。

## [所属]

r\_cg\_timer.c

## [指定形式]

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

備考: m はユニット番号を意味します。

## [引数]

なし

## [戻り値]

なし

**【API 名】 R\_TMR\_KBm\_ForcedOutput\_Stop**

強制出力停止機能に使用するトリガ信号の入力を禁止します。

## [所属]

r\_cg\_timer.c

## [指定形式]

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

備考: m はユニット番号を意味します。

## [引数]

なし

## [戻り値]

なし

**【API 名】 R\_TMR\_KBm\_Create\_UserInit**

16 ビット・タイマ KBm に関するユーザ独自の初期化処理を行います。

備考：本 API 関数は、R\_TMR\_KB\_Create のコールバック・ルーチンとして呼び出されます。

## [所属]

r\_cg\_timer\_user.c

## [指定形式]

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

備考: m はユニット番号を意味します。

## [引数]

なし

## [戻り値]

なし

**【API 名】 r\_tmr\_kbm\_interrupt**

タイマ割り込みの発生に伴う処理を行います。

備考：本 API 関数は、タイマ割り込みに対応した割り込み処理として呼び出されます。

## [所属]

r\_cg\_timer\_user.c

## [指定形式]

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

備考: m はユニット番号を意味します。

## [引数]

なし

## [戻り値]

なし

**【API 名】 R\_TMR\_KC0\_Create**

16 ビット・タイマ KC0 の機能を制御するうえで必要となる初期化処理を行います。

## [所属]

r\_cg\_timer.c

## [指定形式]

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

## [引数]

なし

## [戻り値]

なし

**【API 名】 R\_TMR\_KC0\_Start**

16 ビット・タイマ KC0 のカウント処理を開始します。

## [所属]

r\_cg\_timer.c

## [指定形式]

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

## [引数]

なし

## [戻り値]

なし

**【API 名】 R\_TMR\_KC0\_Stop**

16 ビット・タイマ KC0 のカウント処理を終了します。

## [所属]

r\_cg\_timer.c

## [指定形式]

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

## [引数]

なし

## [戻り値]

なし

**【API 名】 R\_TMR\_KC0\_Set\_PowerOff**

16 ビット・タイマ KC0 に対するクロック供給を停止します。

備考：本 API 関数の呼び出しにより、16 ビット・タイマ KC0 はリセット状態へと移行します。このため、本 API 関数の呼び出し後、制御レジスタへの書き込みは無視されます。

## [所属]

r\_cg\_timer.c

## [指定形式]

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

## [引数]

なし

## [戻り値]

なし

**【API 名】 R\_TMR\_KC0\_Create\_UserInit**

16 ビット・タイマ KC0 に関するユーザ独自の初期化処理を行います。

備考：本 API 関数は、R\_TMR\_KC0\_Create のコールバック・ルーチンとして呼び出されます。

## [所属]

r\_cg\_timer\_user.c

## [指定形式]

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

## [引数]

なし

## [戻り値]

なし

**【API 名】 r\_tmr\_kc0\_interrupt**

タイマ割り込みの発生に伴う処理を行います。

備考：本 API 関数は、タイマ割り込みに対応した割り込み処理として呼び出されます。

## [所属]

r\_cg\_timer\_user.c

## [指定形式]

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

## [引数]

なし

## [戻り値]

なし

**【API 名】 R\_PGA\_Create**

プログラマブル・ゲイン・アンプの機能を制御するうえで必要となる初期化処理を行います。

## [所属]

r\_cg\_pga.c

## [指定形式]

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

## [引数]

なし

## [戻り値]

なし



**【API 名】 R\_PGA\_Start**

プログラマブル・ゲイン・アンプの動作を開始します。

## [所属]

r\_cg\_pga.c

## [指定形式]

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

## [引数]

なし

## [戻り値]

なし

**【API 名】 R\_PGA\_Stop**

プログラマブル・ゲイン・アンプの動作を停止します。

## [所属]

r\_cg\_pga.c

## [指定形式]

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

## [引数]

なし

## [戻り値]

なし

**【API 名】 R\_PGA\_Create\_UserInit**

プログラマブル・ゲイン・アンプに関するユーザ独自の初期化処理を行います。

備考：本 API 関数は、R\_PGA\_Create のコールバック・ルーチンとして呼び出されます。

## [所属]

r\_cg\_pga\_user.c

## [指定形式]

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

## [引数]

なし

## [戻り値]

なし

## ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、  
家電、工作機械、パーソナル機器、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、  
防災・防犯装置、各種安全装置等  
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍用用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続きを行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。  
総合お問合せ窓口：<http://japan.renesas.com/contact/>