

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

第 1 章	はじめに.....	3
第 2 章	対象デバイスについて.....	4
第 3 章	動作環境.....	9
第 4 章	変更点.....	10
4.1	変更点詳細について.....	11
4.1.1	リアルタイム・クロックの出力ソース修正.....	11
4.1.2	シリアル・アレイ・ユニットの出力ソース修正.....	12
4.1.3	PMC レジスタ設定の追加.....	14
4.1.4	UARTn の受信関数の出力ソース修正.....	14
4.1.5	競合端子の制御修正.....	14
4.1.6	A/D コンバータの GUI 表示修正.....	15
4.1.7	タイマの方形波出力について.....	15
4.1.8	ポートの TTL チェックボックスについて.....	15
4.1.9	PIOR 設定について.....	15
4.1.10	TAU1 設定について.....	15
4.1.11	UART2 について.....	16
4.1.12	キー割り込みについて.....	16
4.1.13	簡易 I2C について.....	16
4.1.14	ファイル生成モードの機能追加について.....	17
4.1.15	hdwinit()関数の変更について.....	18
4.1.16	SAU 簡易 I2C に関する API の変更.....	19
4.1.17	タイマ RD、タイマ RJ0 に関する修正と新たな制限について.....	20
4.1.18	TAU の入力パルス間隔測定に関する修正.....	22
4.1.19	電源の表記に関する修正.....	22

第 5 章	注意事項	23
5.1	注意事項一覧	23
5.2	注意事項詳細について	24
5.2.1	UART(0, 2, 3, 6)を含む UART の LIN-bus 機能について	24
5.2.2	IICA または IIC0 の拡張コード、マルチマスタ機能について	24
5.2.3	IICA または IIC0 のスレーブ送信時の注意	24
5.2.4	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	25
5.2.5	シリアル・アレイ・ユニットの制限について	25
5.2.6	ポート設定について	25
5.2.7	リアルタイム・クロックの設定について	25
5.2.8	DTC 機能を使用する時の注意	26
5.2.9	A/D コンバータの初期化関数について	26
5.2.10	UART 送信を設定した場合の初期化関数について	27
5.2.11	タイマ RD の相補 PWM モードについて	27
5.2.12	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	27
5.2.13	安全機能について	27
5.2.14	ファイルのマージ機能について	27

第1章 はじめに

Applilet3 for RL78 は、マイコン周辺機能（タイマ, UART, A/D, etc）を制御するプログラム（デバイス・ドライバ・プログラム）を GUI 設定により自動生成するツールです。各周辺の初期化処理以外にも周辺機能进行操作する関数を API（Application Programming Interface）として提供します。

第2章 対象デバイスについて

Applilet3 for RL78 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。

Applilet3 for RL78/I1A V2.01.00.02 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

Applilet3 for RL78/G12 V2.01.00.01 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

Applilet3 for RL78/G13 V2.01.00.01 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

Applilet3 for RL78/G14 V2.01.00.02 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

Applilet3 for RL78/G1A V2.01.00.01 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

Applilet3 for RL78/F12 V2.01.00.01 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

Applilet3 for RL78/L12 V2.01.00.01 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.0.01
	R01UH0330EJ0003 Rev.0.01

Applilet3 for RL78/D1A V2.01.00.01 がサポートするデバイス一覧	
ピン数	デバイス名
48pin	R5F10CGB, R5F10CGC, R5F10DGC, R5F10DGD, R5F10DGE
64pin	R5F10CLD, R5F10DLD, R5F10DLE
80pin	R5F10CMD, R5F10CME R5F10DMD, R5F10DME, R5F10DMF, R5F10DMG, R5F10DMJ
100pin	R5F10DPE, R5F10DPF, R5F10DPG, R5F10DPJ, R5F10TPJ
設計資料	
資料名	資料番号
RL78/D1A ユーザーズマニュアル ハードウェア編	R01UH0317EJ0003 Rev.0.03

第3章 動作環境

○ホストマシン

- ・ IBM PC/AT 互換機 (Windows® 7, Windows Vista®, Windows® XP)
- ・ プロセッサ : 1GHz 以上 (ハイパースレッディング, マルチコア CPU に対応)
- ・ メモリ容量 : 推奨 2GB 以上。最低 1GB 以上 (64 ビット版 Windows 7 では 2G バイト以上)
- ・ ハードディスク容量 : 空き容量 200MB 以上
- ・ ディスプレイ : 1024×768 以上の解像度, 65536 色以上
- ・ Windows OS 以外に必要なソフトウェア環境
 - .NET Framework 3.5 SP1 + 言語パック
 - Microsoft Visual C++ 2008 SP1 ランタイム・ライブラリ

○開発ツール

製品名	バージョン
ルネサスエレクトロニクス 78K用 IAR Embedded Workbench	V1.30以上
KPIT開発環境 GNURL78	V13.01以上

第4章 変更点

本章では、Applilet3 for RL78 の変更点について説明します。

No	内容	該当するApplilet3							
		RL78/L12 V2.01.00.01	RL78/F12 V2.01.00.01	RL78/G1A V2.01.00.01	RL78/G12 V2.01.00.01	RL78/G13 V2.01.00.01	RL78/G14 V2.01.00.02	RL78/H1A V2.01.00.02	RL78/D1A V2.01.00.01
1	リアルタイム・クロックの出カソース修正	/	/	/	-	-	-	-	-
2	シリアル・アレイ・ユニットの出カソース修正	/	/	/	-	-	-	-	-
3	PMCレジスタ設定の追加	/	/	/	-	-	-	/	/
4	UARTnの受信関数の出カソース修正	/	/	/	-	-	-	-	-
5	競合端子の制御修正	/	/	/	/	-	/	/	/
6	A/DコンバータのGUI表示修正	/	/	/	-	-	-	/	/
7	タイマの方形波出力について	/	/	/	/	-	/	/	/
8	ポートのTTLチェックボックスについて	/	/	/	/	-	/	/	/
9	PIOR設定について	/	/	/	/	/	-	/	/
10	TAU1設定について	/	/	/	/	/	-	/	/
11	UART2について	/	/	/	/	/	-	/	/
12	キー割り込みについて	/	/	/	-	/	/	/	/
13	簡易I2Cについて	/	/	/	-	-	-	/	/
14	ファイル生成モードの機能追加について	○	○	○	○	○	○	○	○
15	hdwinit()関数の変更について	○	○	○	○	○	○	○	○
16	SAU簡易I2Cに関するAPIの変更	/	/	/	/	/	/	/	/
17	タイマRD、タイマRJ0に関する修正	/	/	/	/	/	-	/	/
18	TAUの入カパルス間隔測定に関する修正	-	-	-	-	-	-	-	-
19	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)共に修正済みです。

4.1.3 PMCレジスタ設定の追加

各周辺のコム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

修正済みです。

4.1.4 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;

修正済みです。

4.1.5 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

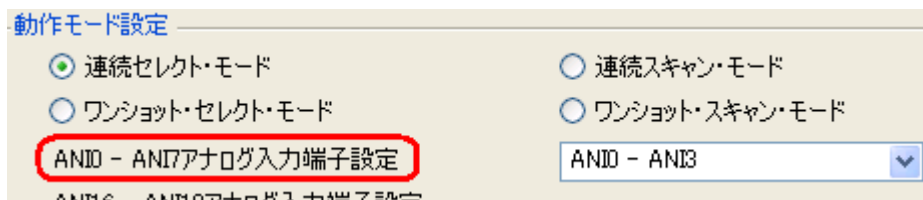
修正後: P50 SDA11

P30 SCL11

修正済みです。

4.1.6 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。



修正済みです。

4.1.7 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。

修正済みです

4.1.8 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。

修正済みです

4.1.9 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

修正済みです

4.1.10 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。

修正済みです

4.1.11 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;`

修正済みです

4.1.12 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;`

修正済みです

4.1.13 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。

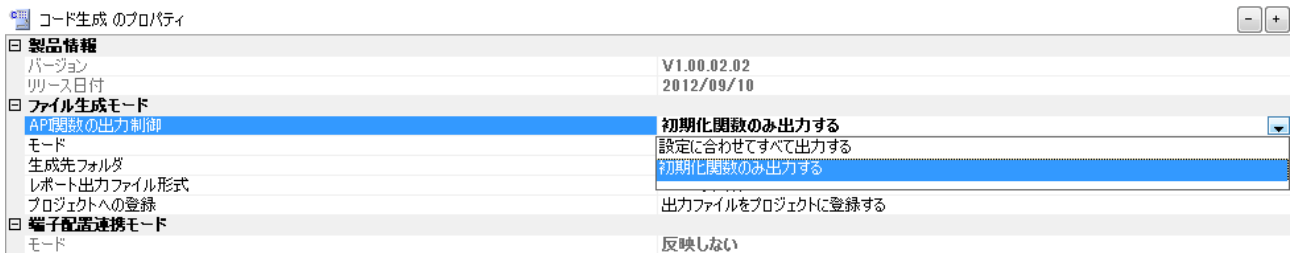
修正済みです

4.1.14 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” → 従来のとおり、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。



4.1.15 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
}
```

4.1.16 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。

修正済みです

4.1.17 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) 修正済みです。

4.1.18 TAUの入力パルス間隔測定に関する修正

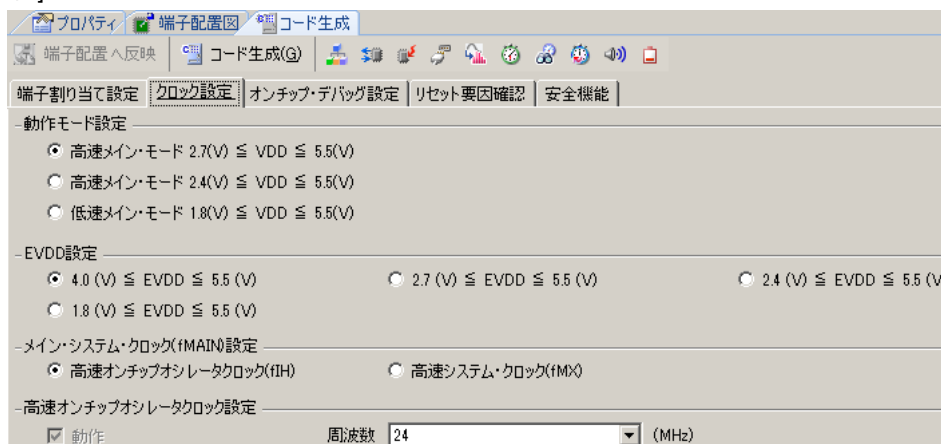
TAUの入力パルス間隔測定機能、ハイ/ロウ幅測定機能を使用時、割り込み関数r_taux_channelx_interrupt()で正しい測定値を取得できなかったのを修正しました。

修正済みです

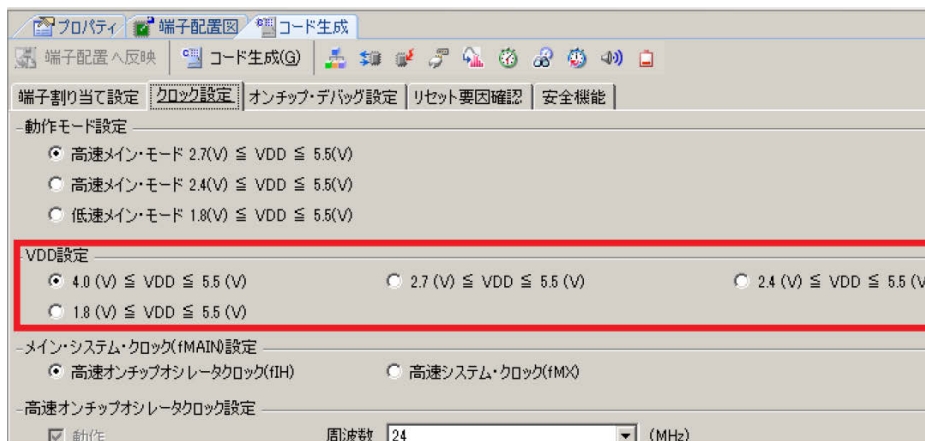
4.1.19 電源の表記に関する修正

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]



修正済みです

第5章 注意事項

本章では、Applilet3 for RL78 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Applilet3							
		RL78/L12 V2.01.00.01	RL78/F12 V2.01.00.01	RL78/G1A V2.01.00.01	RL78/G12 V2.01.00.01	RL78/G13 V2.01.00.01	RL78/G14 V2.01.00.02	RL78/H1A V2.01.00.02	RL78/D1A V2.01.00.01
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、マルチマスタ機能について	○	○	○	○	○	○	⊖	○
3	IICAまたはIIC0のスレーブ送信時の注意	/	/	/	-	-	-	≡	-
4	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	○	○	○	○	○	○	≠	○
5	シリアル・アレイ・ユニットの制限について	/	/	/	/	/	/	⊖	/
6	ポート設定について	/	/	/	-	-	-	≡	-
7	リアルタイム・クロックの設定について	/	/	/	/	-	-	≡	/
8	DTC機能を使用する時の注意	/	/	/	/	/	○	≠	/
9	A/Dコンバータの初期化関数について	/	/	/	-	-	-	≠	/
10	UART送信を設定した場合の初期化関数について	/	/	/	-	-	-	≠	/
11	タイマRDの相補PWMモードについて	/	/	/	/	/	-	≠	/
12	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	○	○	○	○	○	○	⊖	○
13	安全機能について	○	○	○	○	○	○	⊖	○
14	ファイルのマージ機能について	○	○	○	○	○	○	⊖	○

○：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。

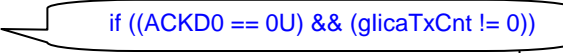
5.2.3 IICAまたはIIC0のスレーブ送信時の注意

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;
            }
        }
    }
}
```



【回避策】 修正済みです。

5.2.4 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

Applilet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。

【回避策】 ありません。

5.2.5 シリアル・アレイ・ユニットの制限について

Appliletは、1線UARTモード及びDMX512通信の設定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】 修正済みです。

5.2.7 リアルタイム・クロックの設定について

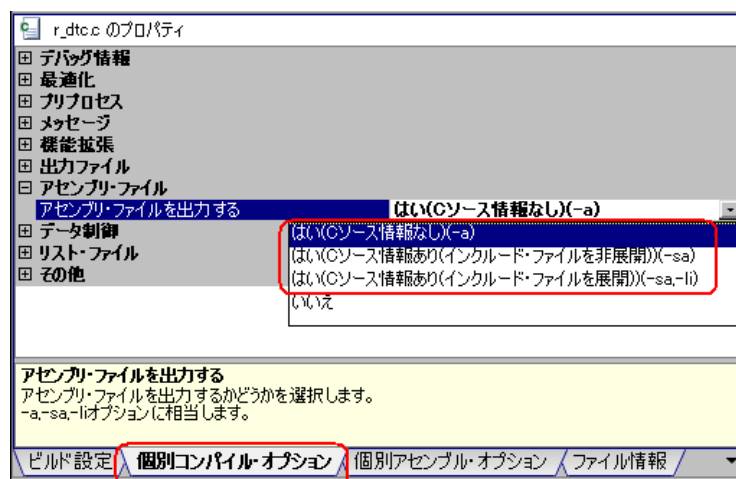
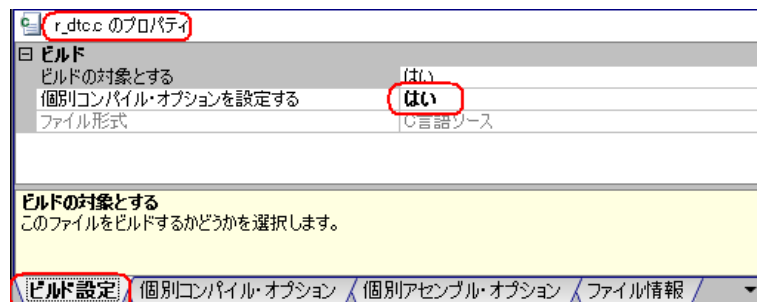
デバイスでは、クロック・ソースを15(fIL)kHzにした時、時計機能は使えません。しかし、GUI上では設定できるように表示されます。この場合、時計機能の設定を行わないでください。

【回避策】 修正済みです。

5.2.8 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。

5.2.9 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】 修正済みです。

5.2.10 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】 修正済みです。

5.2.11 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

5.2.12 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

Appliletは、低速及び高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。

【回避策】 ありません。

5.2.13 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。

5.2.14 ファイルのマージ機能について

コード生成でファイル生成モードを“ファイルをマージする”設定したとき、編集したソースコードが消える場合があります。

例えば、下記のコメントとコメントとの間にソースコードを書けば、マージされます。

```
/* Start user code. Do not edit comment generated here */
```

```
/* End user code. Do not edit comment generated here */
```

しかし、編集したソースコード中の "{"と"}"の数が合わない場合（コメントを含む）で、コード生成を行うと編集したソースコードが消される場合があります。

【回避策】 ありません。

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/>