

**CS+用 RL78コード生成 (CS+ for CA, CX)
CS+用 RL78コード生成 (CS+ for CC)
e2 studio (Code Generator プラグイン)
RL78コード生成支援ツール AP4
RL78コード生成支援ツール Applilet3
ご使用上のお願い**

CS+用 RL78コード生成 (CS+ for CA, CX)、CS+用 RL78コード生成 (CS+ for CC)、 e2 studio (Code Generator プラグイン)、RL78コード生成支援ツール AP4 および RL78コード生成支援ツール Applilet3 の使用上の注意事項を連絡します。

1. クロック発生回路(PLL回路動作時)の注意事項

対象: RL78/D1A, RL78/F13, RL78/F14, RL78/G1C および RL78/L1Cグループ

2. ポート4のP40 設定時の注意事項

対象: RL78/D1A, RL78/F12, RL78/F13, RL78/F14, RL78/G10, RL78/G12, RL78/G13, RL78/G14, RL78/G1A, RL78/G1C, RL78/G1E, RL78/G1F, RL78/G1G, RL78/I1A, RL78/I1B, RL78/I1D, RL78/L1C, RL78/L12 および RL78/L13グループ

3. UART0 および UARTF 生成コードの注意事項

対象: RL78/F12グループ

1. クロック発生回路(PLL回路動作時)の注意事項

1.1 該当製品

- CS+用 RL78コード生成 (CS+ for CA,CX) V2.01.00以降
- CS+用 RL78コード生成 (CS+ for CC) V2.07.00以降
- e2 studio V2.2.0.13以降 (Code Generator プラグイン V1.0.1以降)
- RL78コード生成支援ツール AP4 V1.06.00以降
- RL78コード生成支援ツール Applilet3 V1.07.00以降

1.2 該当マイコン

RL78/D1A, RL78/F13, RL78/F14, RL78/G1C および RL78/L1Cグループ

1.3 内容

クロック発生回路にて、PLL回路動作時の生成コードに誤りがあります。
PLL制御レジスタ(PLLCTL)を設定後、ウェイトが必要です。

1.4 回避策

r_cg_cgc.c にある関数 void R_CGC_Create(void) の出力コードを下記に従い修正してください。なお、コード生成後は常に修正が必要です。
なお、該当マイコンにより出力コードが異なります。

修正前(該当マイコン RL78/D1A, RL78/F13 および RL78/F14グループ):

```
-----  
void R_CGC_Create(void)  
{  
.....  
    /* Set fPLL */  
    PLLCTL = _40_CGC_LOCKUP_WAIT_8 | _00_CGC_PLL_BELOW_32MHZ  
            | _00_CGC_PLL_DIVISION_2 | _00_CGC_PLL_MULTIPLY_X12;  
    PLLON = 1U;          /* <- faulty sequence */  
  
    /* Change the waiting time according to the system */  
    for (w_count = 0U; w_count <= CGC_PLLWAITTIME; w_count++)  
    {  
        NOP();  
    }  
  
    while ((PLLSTS & 0x80) == 0U)  
    {  
        ;  
    }  
.....  
}
```

修正後(該当マイコン RL78/D1A, RL78/F13 および RL78/F14グループ):

```
-----  
void R_CGC_Create(void)  
{  
.....  
    /* Set fPLL */  
    PLLCTL = _40_CGC_LOCKUP_WAIT_8 | _00_CGC_PLL_BELOW_32MHZ  
            | _00_CGC_PLL_DIVISION_2 | _00_CGC_PLL_MULTIPLY_X12;
```

```

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

```

```

PLLON = 1U; /* <- correct sequence */
/* That is, modify the order of */
/* processing. */

```

```

while ((PLLSTS & 0x80) == 0U)
{
    ;
}

```

```

.....
}

```

修正前(該当マイコン RL78/G1C, RL78/L1Cグループ):

```

void R_CGC_Create(void)

```

```

{

```

```

.....

```

```

/* Set fPLL */

```

```

DSCCTL = _04_CGC_PLL_DIVISION_2 | _00_CGC_PLL_MULTI_12;

```

```

MCKC = _00_CGC_FIH_DIVISION_2;

```

```

/* Change the waiting time according to the system */

```

```

for (w_count = 0U; w_count <= CGC_FPLLWAITTIME; w_count++)

```

```

{

```

```

    NOP();

```

```

}

```

```

DSCCTL |= _01_CGC_PLL_OPERATION_ON;

```

```

/* <- faulty sequence */

```

```

.....
}

```

修正後(該当マイコン RL78/G1C, RL78/L1Cグループ):

```

void R_CGC_Create(void)

```

```

{

```

```

.....
/* Set fPLL */
DSCCTL = _04_CGC_PLL_DIVISION_2 | _00_CGC_PLL_MULTI_12;
MCKC = _00_CGC_FIH_DIVISION_2;

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

DSCCTL |= _01_CGC_PLL_OPERATION_ON;

/* Change the waiting time according to the system */
/* <- correct sequence */

for (w_count = 0U; w_count <= CGC_FPLLWAITTIME; w_count++)
{
    NOP();
}
.....
}
-----

```

1.5 恒久対策

次期バージョンで改修する予定です。

2. ポート4のP40 設定時の注意事項

2.1 該当製品

- CS+用 RL78コード生成 (CS+ for CA,CX) V2.01.00以降
- CS+用 RL78コード生成 (CS+ for CC) V2.07.00以降
- e2 studio V2.1.0.21以降 (Code Generator プラグイン V1.0.0以降)
- RL78コード生成支援ツール AP4 V1.06.00以降
- RL78コード生成支援ツール Applilet3 V1.07.00以降

2.2 該当マイコン

RL78/D1A, RL78/F12, RL78/F13, RL78/F14, RL78/G10, RL78/G12, RL78/G13, RL78/G14, RL78/G1A, RL78/G1C, RL78/G1E, RL78/G1F, RL78/G1G, RL78/I1A, RL78/I1B, RL78/I1D, RL78/L1C, RL78/L12 および RL78/L13グループ

2.3 内容

ポート4の内蔵プルアップ抵抗の設定において、P40のみ内蔵プルアップを接続しない設定とした場合、生成コードに誤りがあります。

P40 のプルアップ抵抗オプション・レジスタ(PU4)を設定するコードが生成されません。

2.4 回避策

r_cg_port.c にある関数 void R_PORT_Create(void) の出力コードにプルアップ抵抗オプション・レジスタ(PU4)の設定を追記してください。なお、コード生成後は常に修正が必要です。

修正前:

```
-----  
void R_PORT_Create(void)  
{  
    /* <- faulty sequence */  
    R_PORT_Create_UserInit();  
}
```

修正後:

```
-----  
void R_PORT_Create(void)  
{  
    PU4 = _00_PUn0_PULLUP_OFF; /* <- correct sequence */  
    /* That is, modify the */  
    /* order of processing. */  
  
    R_PORT_Create_UserInit();  
}
```

2.5 恒久対策

次期バージョンで改修する予定です。

3. UART0 および UARTF 生成コードの注意事項

3.1 該当製品

- CS+用 RL78コード生成 (CS+ for CA,CX) V2.01.00以降
- CS+用 RL78コード生成 (CS+ for CC) V2.07.00以降
- e2 studio V2.2.0.13以降 (Code Generator プラグイン V1.0.1以降)
- RL78コード生成支援ツール Applilet3 V1.07.00以降

3.2 該当マイコン

RL78/F12グループ

3.3 内容

(a) シリアル・アレイ・ユニットのユニット0をUART0として使用時、送信機能または送受信機能に設定した場合、生成コードに誤りがあります。

r_cg_serial.c にある関数 void R_UART0_Create(void) に不要なコードが出力されます。

(b) アシクロナス・シリアル・インタフェースLIN-UART(UARTF)を送信機能または送受信機能に設定した場合、LTXD0端子の設定において生成コードに誤りがあります。

r_cg_serial.c にある関数 void R_UARTF0_Create(void) に不正なコードが出力されます。

3.4 回避策

(a) 3.3(a)の回避策

r_cg_serial.c にある関数 void R_UART0_Create(void) の出力コードを下記に従い修正してください。なお、コード生成後は常に修正が必要です。

修正前:

```
-----  
void R_UART0_Create(void)  
{  
.....  
  /* Set RxD0 pin */  
  PMX1 |= 0x01U;          /* <- faulty code */  
  PM1 |=0x02U;  
.....  
}
```

修正後:

```
-----  
void R_UART0_Create(void)  
{  
.....  
  /* Set RxD0 pin */  
                                     /* <- deleted code */  
  PM1 |=0x02U;  
.....  
}
```

(b) 3.3(b)の回避策

r_cg_serial.c にある関数 void R_UARTF0_Create(void) の出力コードを下記に従い修正してください。なお、コード生成後は常に修正が必要です。

修正前:

```
-----  
void R_UARTF0_Create(void)  
{  
.....
```

```
/* Set LTXD0 pin */
PM5 |= 0x02U;
PMX2 = 0xFEU;          /* <- faulty code */
.....
}
```

修正後:

```
void R_UARTF0_Create(void)
{
.....
/* Set LTXD0 pin */
PM5 |= 0x02U;
PMX2 &= 0xFEU;        /* <- correct code */
.....
}
```

3.5 恒久対策

次期バージョンで改修する予定です。

[免責事項]

過去のニュース内容は発行当時の情報をもとにしており、現時点では変更された情報や無効な情報が含まれている場合があります。ニュース本文中のURLを予告なしに変更または中止することがありますので、あらかじめご承知ください。