

【注意事項】

R20TS0462JJ0100

Rev.1.00

2019.08.01 号

CS+用 RX コード生成

e² studio Code Generator プラグイン

RX コード生成支援ツール AP4

概要

タイトルに記載している製品の使用上の注意事項を連絡します。

1. I²C バスインタフェースの NACK 受信転送中断機能使用時の注意事項
2. スリープモード使用時の注意事項

1. I²C バスインタフェースの NACK 受信転送中断機能使用時の注意事項

1.1 該当製品

- CS+用 RX コード生成 V1.00.00 (CS+ for CC V2.01) 以降
- Code Generator プラグイン V1.00.00 (e² studio V2.1.0) 以降
- AP4 for RX V1.00.00 以降

1.2 該当デバイス

- RX ファミリ：
RX110、RX111、RX113、RX130、RX230、RX231、RX23T、RX24T、RX24U、
RX64M、RX651、RX65N、および RX71M グループ

1.3 内容

I²C バスインタフェースで NACK 受信転送中断機能を使用する場合、NACK 受信時のエラー割り込み処理が正しくありません。そのため動作に以下の影響があります。

- ・ マスタモード時は、通信再開のコードが生成されるため、通信が中断できません。
- ・ スレーブモード時は、不要な処理が実行されます。

➤ 不具合箇所

マスタモード

```

/*****
* Function Name: r_Config_RIIC0_error_interrupt
* Description  : This function is EEI0 interrupt service routine
* Arguments    : None
* Return Value : None
*****/
void r_Config_RIIC0_error_interrupt(void)
{
    volatile uint8_t dummy;
    ...
    else if ((1U == RIIC0.ICIER.BIT.NAKIE) && (1U == RIIC0.ICSR2.BIT.NACKF))
        ...
        RIIC0.ICSR2.BIT.NACKF = 0U; ← このコードで通信が再開される
        r_Config_RIIC0_callback_receiveerror(MD_ERROR3);
    }
    ...
}

```

スレーブモード

```

/*****
* Function Name: r_Config_RIIC0_error_interrupt
* Description  : This function is EEI0 interrupt service routine
* Arguments    : None
* Return Value : None
*****/
void r_Config_RIIC0_error_interrupt(void)
{
    volatile uint8_t dummy;
    ...
    else if ((1U == RIIC0.ICIER.BIT.NAKIE) && (1U == RIIC0.ICSR2.BIT.NACKF))
    {
        dummy = RIIC2.ICDRR; ← これらのコードは不要
        RIIC0.ICSR2.BIT.NACKF = 0U; ←
        r_Config_RIIC0_callback_receiveerror(MD_ERROR3);
    }
    ...
}

```

1.4 回避策

以下ソースファイルのエラー割り込み関数内にある、NACK 受信時のエラー処理を修正してください。

注意：生成コードは、再度コード生成を行うと修正前の状態に戻りますので、ソースファイルの修正はコード生成を行う度に実施してください。

- ・ ソースファイル : r_cg_riic_user.c
- ・ 関数 : void r_riicn_error_interrupt(void)

n=チャンネル番号

以下に RX64M グループの場合の修正例を記します。修正箇所を**赤文字**で示します。

また、NACK 受信時のユーザプログラム例を記しました。ご参照のうえ追記してください。

➤ マスタモードでチャンネル 0 の場合

修正方法

```

/*****
* Function Name: r_Config_RIIC0_error_interrupt
* Description  : This function is EEI0 interrupt service routine
* Arguments    : None
* Return Value : None
*****/
void r_Config_RIIC0_error_interrupt(void)
{
    volatile uint8_t dummy;
    ...
    else if ((1U == RIIC0.ICIER.BIT.NAKIE) && (1U == RIIC0.ICSR2.BIT.NACKF))
    {
        RIIC0.ICSR2.BIT.NACKF = 0U; ← この行を削除してください
        r_Config_RIIC0_callback_receiveerror(MD_ERROR3);
    }
    ...
}

```

NACK 受信時のユーザプログラム例 :

NACK 受信時に、次の通信の準備を行うためのサンプルプログラムです。

```
void NackReceive(void)
{
    volatile uint8_t dummy;

    /* RIICによる通信割り込みを禁止します */
    R_Config_RIIC0_Stop();

    /* ストップコンディションを発行します */
    RIIC0.ICSR2.BIT.STOP = 0U;
    RIIC0.ICCR2.BIT.SP = 1U;

    /* マスタ受信時はダミーリードを実施します */
    dummy = RIIC0.ICDRR;

    /* ストップコンディション発行を待ちます */
    while (0U == RIIC0.ICSR2.BIT.STOP);

    /* ステータスフラグをクリアします */
    RIIC0.ICSR2.BIT.NACKF = 0U;
    RIIC0.ICSR2.BIT.STOP = 0U;

    /* RIICによる通信割り込みを許可します */
    R_Config_RIIC0_Start();
}
```

➤ スレーブモードでチャンネル0の場合

修正方法

```

/*****
* Function Name: r_Config_RIIC0_error_interrupt
* Description  : This function is EEI0 interrupt service routine
* Arguments    : None
* Return Value : None
*****/
void r_Config_RIIC0_error_interrupt(void)
{
    volatile uint8_t dummy;
    ...
    else if ((1U == RIIC0.ICIER.BIT.NAKIE) && (1U == RIIC0.ICSR2.BIT.NACKF))
    {
        dummy = RIIC2.ICDRR;
        RIIC0.ICSR2.BIT.NACKF = 0U;
        r_Config_RIIC0_callback_receiveerror(MD_ERROR3);
    }
    ...
}

```



NACK 受信時のユーザプログラム例 :

NACK 受信時に、次の通信の準備を行うためのサンプルプログラムです。

```

void NackReceive(void)
{
    volatile uint8_t dummy;

    /* RIICによる通信割り込みを禁止します */
    R_Config_RIIC0_Stop();

    /* ダミーリードを実施します */
    dummy = RIIC0.ICDRR;

    /* ストップコンディション発行を待ちます */
    while (0U == RIIC0.ICSR2.BIT.STOP);

    /* ステータスフラグをクリアします */
    RIIC0.ICSR2.BIT.NACKF = 0U;
    RIIC0.ICSR2.BIT.STOP = 0U;

    /* RIICによる通信割り込みを許可します */
    R_Config_RIIC0_Start();
}

```

1.5 恒久対策

改修予定はありません。

2. スリープモード使用時の注意事項

2.1 該当製品

- CS+用 RX コード生成 V1.00.00 (CS+ for CC V2.01) 以降
- Code Generator プラグイン V1.00.00 (e² studio V2.1.0) 以降
- AP4 for RX V1.00.00 以降

2.2 該当デバイス

- RX ファミリ :
RX64M、RX651、RX65N、および RX71M グループ

2.3 内容

スリープモード復帰クロックソース切り替えを有効にした場合、R_LPC_Sleep 関数のクロックソースのエラー判定に誤りがあり、スリープモードへの移行が正常に行われません。不正な動作の詳細は、表 2.1 をご参照ください。

表 2.1 R_LPC_Sleep 関数の不正動作

対象 デバイス	スリープモード移行時のクロックソース				
	MAINCLK	SUBCLK	PLLCLK	HOCO	LOCO
RX64M RX651 RX65N RX71M	2	3	1	2	3
<p>[不正動作]</p> <ol style="list-style-type: none"> 1. スリープモードに移行できないクロックソースで、スリープモードへ移行してしまう。 2. スリープモードに移行できないクロックソースで、スリープモードに移行してしまい、戻り値が正常終了(MD_OK)で返ってくる。 3. スリープモードに移行後、戻り値が異常終了(MD_ERROR1)で返ってくる。 					

2.4 回避策

以下ソースファイルのスリープモード復帰クロックソース切り替え有効時のエラー判定処理を修正してください。

注意：生成コードは、再度コード生成を行うと修正前の状態に戻りますので、ソースファイルの修正はコード生成を行う度に実施してください。

- ・ ソースファイル : r_cg_lpc.c

- ・ 関数 : void R_LPC_Sleep(void)

➤ 動作クロックがサブクロックおよび LOCO 以外の場合、エラーコードを設定して実行を終了するように修正してください。修正箇所を**赤文字**で示します。

修正前

```

/*****
 * Function Name: R_LPC_Sleep
 * Description  : This function enables sleep mode.
 * Arguments   : None
 * Return Value : status -
 *              MD_OK or MD_ERROR1
 *****/
MD_STATUS R_LPC_Sleep(void)
{
    ...
    /* When RSTCKEN is enable, the possible clock source is HOCO or
       main clock oscillator */
    if ((1U == SYSTEM.RSTCKCR.BIT.RSTCKEN) &&
        (SYSTEM.SCKCR3.BIT.CKSEL != _0100_LPC_CLOCKSOURCE_HOCO) &&
        (SYSTEM.SCKCR3.BIT.CKSEL != _0200_LPC_CLOCKSOURCE_MAINCLK))
    {
        status = MD_ERROR1;
    }
    ...
}

```

修正後

```

/*****
 * Function Name: R_LPC_Sleep
 * Description  : This function enables sleep mode.
 * Arguments   : None
 * Return Value : status -
 *              MD_OK or MD_ERROR1
 *****/
MD_STATUS R_LPC_Sleep(void)
{
    ...
    /* When RSTCKEN is enable, the possible clock source is LOCO or
       sub clock oscillator */
    if ((1U == SYSTEM.RSTCKCR.BIT.RSTCKEN) &&
        (SYSTEM.SCKCR3.BIT.CKSEL != 0U) &&
        (SYSTEM.SCKCR3.BIT.CKSEL != 3U))
    {
        status = MD_ERROR1;
        return status;
    }
    ...
}

```

2.5 恒久対策

改修予定はありません。

以上

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Aug.01.19	-	新規発行

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。

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。

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24 (豊洲フォレシア)

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。