

【注意事項】

RX600、RX200 シリーズ

RIIC を使った I<sup>2</sup>C シングルマスタ制御ソフトウェア

概要

RX600、RX200 シリーズ RIIC を使った I<sup>2</sup>C シングルマスタ制御ソフトウェアの使用上の注意事項を連絡します。

1. SCL クロックを低速で使用する場合の注意事項
2. RIIC チャネル 0 以外を使用する場合の注意事項

1. SCL クロックを低速で使用する場合の注意事項

1.1 該当製品

- RX600、RX200 シリーズ RIIC を使った I<sup>2</sup>C シングルマスタ制御ソフトウェア Rev.1.02 および Rev.1.03

1.2 該当 MCU

RX62N、RX63N、RX63T、RX210 および RX21A グループ

1.3 内容

正常な最終データ受信では、マイコンが 8 ビットのデータを受信した後、1 ビットの NACK を送信します。しかし、異常時は最終データ受信後に NACK を送信しません。

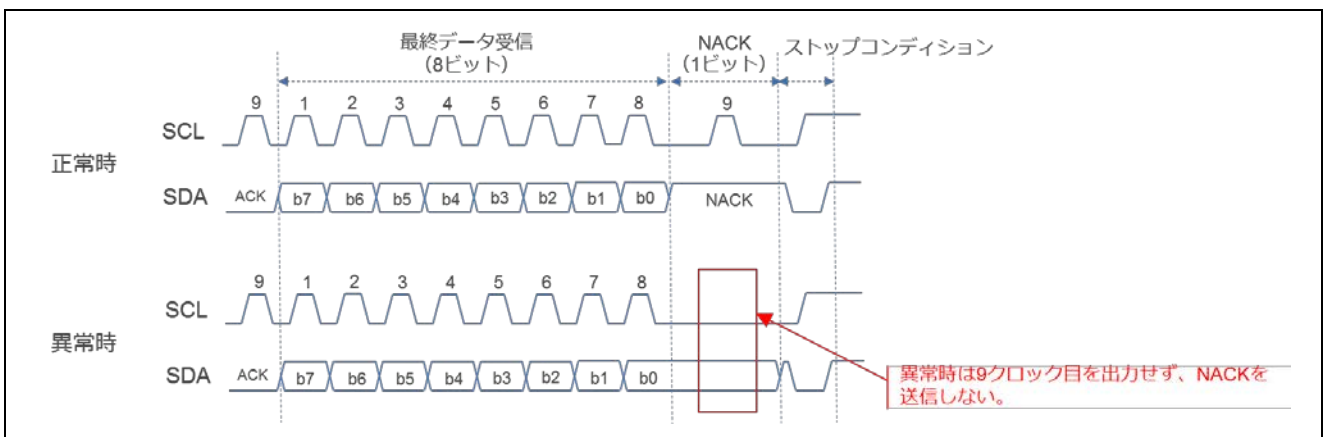


図 1 正常時と異常時の最終データ受信波形

1.4 発生条件

以下の全ての条件を満たす場合に発生します。

- (1) ソフトウェアの制御プロトコルをマスタ受信またはマスタ複合にした場合。
- (2) システムクロック (ICLK) > 周辺モジュールクロック (PCLK) > SCL クロックの場合<sup>(注)</sup>。

ICLK と PCLK に対して、SCL クロックの差分が大きいほど発生確率が高くなります。

注：(参考値) ICLK=100MHz、PCLK=50MHz、SCL クロック ≤ 100KHz の場合に発生します。

## 1.5 回避策

以下の関数について、赤字のとおりソースコードを変更してください。

r\_iic\_drv\_sfr\_rxXXX.c<sup>注</sup>の r\_iic\_drv\_receive\_end\_setting()関数

注：XXX は使用する RX マイコン名です。例えば r\_iic\_drv\_sfr\_rx63t.c です。

[回避策]

```
#define R_IIC_ICMR3_ACKWP_CLR    (uint8_t)(0xEF)
#define R_IIC_ICMR3_ACKBT_SET    (uint8_t)(0x08)
#define R_IIC_ICMR3_ACKBT_CLR    (uint8_t)(0xF7)

void r_iic_drv_receive_end_setting(r_iic_drv_info_t * pRIic_Info)
{
    /* Creates the register pointer for the specified RIIC channel. */
    volatile uint8_t * const    pcICMR3    = ICMR3_ADR(pRIic_Info->ChNo);

    /* Sets ICMR3.ACKBT bit. */
    *pcICMR3 |= R_IIC_ICMR3_ACKWP_SET;
    *pcICMR3 |= R_IIC_ICMR3_ACKBT_SET;
    *pcICMR3 &= R_IIC_ICMR3_ACKWP_CLR;

    /* Clears ICMR3.WAIT bit. */
    *pcICMR3 &= R_IIC_ICMR3_WAIT_CLR;
    if (R_IIC_ICMR3_WAIT_CLR != *pcICMR3)
    {
        nop();
    }
}
```

## 1.6 恒久対策

次期バージョンで改修予定です。なお、改修時期は未定です。

## 2. RIIC チャネル 0 以外を使用する場合の注意事項

### 2.1 該当製品

- RX600、X200 シリーズ RIIC を使った I<sup>2</sup>C シングルマスタ制御ソフトウェア Rev.1.02 および Rev.1.03

### 2.2 該当 MCU

RX62N、RX63N、RX63T、RX210 および RX21A グループ

### 2.3 内容

正常なデータ受信では、マイコンが 8 ビットのデータを受信した後、1 ビットの ACK を送信します（最終データの場合は NACK）。しかし、異常時はデータを受信した後、常に NACK を送信します。そのため、2 バイト目のデータ受信から最終データ受信までの間、スレーブデバイスから不定値を受信します<sup>(注)</sup>。

注：スレーブデバイスが NACK 受信後に送信するデータについては、スレーブデバイスの仕様をご確認ください。

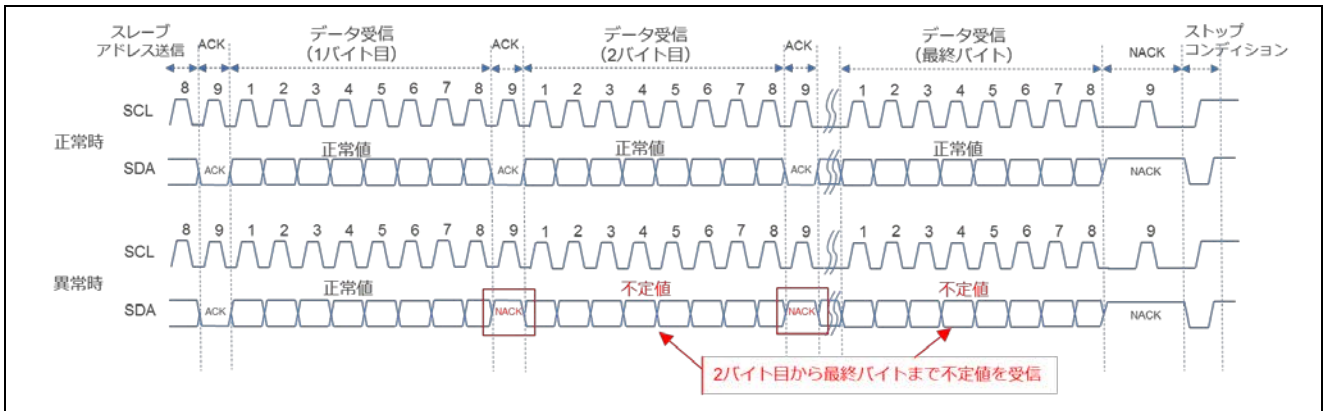


図 2 RIIC チャンネル 0 以外を使用した場合の正常時と異常時の通信波形

## 2.4 発生条件

以下の全ての条件を満たす場合に発生します。

- (1) ソフトウェアの制御プロトコルをマスタ受信またはマスタ複合にした場合。
- (2) RIIC チャンネル 0 以外を使用した場合。
- (3) 最終データ受信処理において、ICDRR レジスタリードから ICMR3.ACKBT ビットを“1”にするまでの期間に、RIIC 以外の割り込みが発生した場合。
- (4) 上記 (3) の割り込み処理中にストップコンディションの発行が完了した場合。
- (5) 上記 (4) のストップコンディション発行後、再びデータ受信を実行した場合。

## 2.5 回避策

以下の関数について、赤字のとおりソースコードを変更してください。

r\_iic\_drv\_sfr\_rxXXX.c<sup>注</sup>の r\_iic\_drv\_next\_comm\_setting()関数

注：XXX は使用する RX マイコン名です。例えば r\_iic\_drv\_sfr\_rx63t.c です。

[回避策]

```
void r_iic_drv_next_comm_setting(r_iic_drv_info_t * pRIic_Info)
{
    /* Creates the register pointer for the specified RIIC channel. */
    volatile uint8_t * const pcICSR2 = ICSR2_ADR(pRIic_Info->ChNo);
    volatile uint8_t * const pcICMR3 = ICMR3_ADR(pRIic_Info->ChNo);

    /* Checks the internal mode. */
    if ((R_IIC_MODE_READ == g_iic_InternalInfo[pRIic_Info->ChNo].Mode) ||
        (R_IIC_MODE_COMBINED == g_iic_InternalInfo[pRIic_Info->ChNo].Mode))
    {
        /* Clears ICMR3.RDRFS bit.*/
        /* Clears ICMR3.ACKBT bit. */
        *pcICMR3 |= R_IIC_ICMR3_ACKWP_SET;
        *pcICMR3 = R_IIC_ICMR3_INIT;
    }
    (中略)
}
```

## 2.6 恒久対策

次期バージョンで改修予定です。なお、改修時期は未定です。

以上

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2017.12.01	-	新規発行
1.01	2018.05.16	2	1.5 回避策の内容変更

ルネサスエレクトロニクス株式会社  
〒135-0061 東京都江東区豊洲 3-2-24 (豊洲フォレシア)

## ■総合お問い合わせ先

<https://www.renesas.com/contact/>

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。

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。