【注意事項】
RX ファミリ
RSPI モジュール Firmware Integration Technology,
RX Driver Package

R20TS0667JJ0100
Rev.1.00
2021.05.16 号

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. 高速モードを使用する場合の注意事項

1. 1. 高速モードを使用する場合の注意事項

1.1 該当製品

(1) RSPI モジュール Firmware Integration Technology（RSPI FIT モジュール）

該当するリビジョンおよびドキュメントは、以下のとおりです。

表 1.1 RSPI FIT モジュール該当製品一覧

RSPI FIT モジュールのリビジョン	資料番号
Rev.3.00	R01AN1827JJ0300
Rev.2.05	R01AN1827JJ0205
Rev.2.04	R01AN1827JJ0204
Rev.2.03	R01AN1827JJ0203
Rev.2.02	R01AN1827JJ0202
Rev.2.01	R01AN1827JJ0201
Rev.2.00	R01AN1827JJ0200

(2) RX Driver Package

(1)の RSPI FIT モジュールは、RX Driver Package にも同梱されています。
該当する RX Driver Package の製品名、リビジョン、および同梱している RSPI FIT モジュールのリビジョンおよびドキュメントは、以下のとおりです。

表 1.2 RSPI FIT モジュール同梱製品一覧

RX Driver Package の製品名	RX Driver Package のリビジョン	資料番号	同梱している RSPI FIT モジュールのリビジョン
RX ファミリ RX Driver Package Ver.1.29	Rev.1.29	R01AN5826JJ0129	Rev.3.00
RX ファミリ RX Driver Package Ver.1.28	Rev.1.28	R01AN5761JJ0128	Rev.3.00
RX ファミリ RX Driver Package Ver.1.27	Rev.1.27	R01AN5600JJ0127	Rev.3.00
RX ファミリ RX Driver Package Ver.1.26	Rev.1.26	R01AN5401JJ0126	Rev.2.05
RX ファミリ RX Driver Package Ver.1.25	Rev.1.25	R01AN371JJ0125	Rev.2.05
RX ファミリ RX Driver Package Ver.1.24	Rev.1.24	R01AN5267JJ0124	Rev.2.04

RX ファミリ RX Driver Package Ver.1.23	Rev.1.23	R01AN4976JJ0123	Rev.2.03
RX ファミリ RX Driver Package Ver.1.22	Rev.1.22	R01AN4873JJ0122	Rev.2.03
RX ファミリ RX Driver Package Ver.1.20	Rev.1.20	R01AN4794JJ0120	Rev.2.01
RX ファミリ RX Driver Package Ver.1.19	Rev.1.19	R01AN4677JJ0119	Rev.2.00

1.2 該当デバイス

RX110、RX111、RX113、および RX130 グループ

RX230、RX231、RX23E-A、RX23W、RX23T、RX24T、および RX24U グループ

RX64M、RX651、RX65N、RX66T、および RX66N グループ

RX71M、RX72T、RX72M、および RX72N グループ

1.3 内容と発生条件

(1) バッファオーバフローが発生します。

高速モードに設定し、かつ他の周辺割り込みなどにより、データレジスタからデータをリードするタイミングが遅れた場合、バッファオーバフローが発生します。

(2) 受信データを正しく受け取れない場合があります。

高速モードに設定し、かつ他の周辺割り込みなどにより受信バッファフル割り込みの発生が遅れた場合、送信エンプティ割り込みで SPDR レジスタをダミーリードしてしまうため、受信データを取りこぼす可能性があります。

1.4　回避策

高速モードを使用する際は、以下の処理を追加してください。

(1)　バッファオーバフローを発生させないための回避策

RX110、RX111、RX113 グループは RSPCK 自動停止機能許可ビットがないため不要です。

ファイル：¥src¥smc_gen¥r_rspi_rx¥src¥r_rspi_defaults.h

修正前

```
  #define RSPI_SPCR2_DEF       (RSPI_SPCR2_SPPE_DEF | RSPI_SPCR2_SPOE_DEF |
RSPI_SPCR2_SPIIE_DEF | RSPI_SPCR2_PTE_DEF | ¥
                               RSPI_SPCR2_SCKASE_DEF)
```

修正後（赤字部分を追加）

```
#if (!defined(BSP_MCU_RX110) && !defined(BSP_MCU_RX111)
&& !defined(BSP_MCU_RX113))
  #define RSPI_SPCR2_SCKASE_ENABLE (0x10) /* 1: Enables the RSPCK auto-stop function
*/
#else
#define RSPI_SPCR2_SCKASE_ENABLE (0x00)
#endif


  #define RSPI_SPCR2_DEF       (RSPI_SPCR2_SPPE_DEF | RSPI_SPCR2_SPOE_DEF |
RSPI_SPCR2_SPIIE_DEF | RSPI_SPCR2_PTE_DEF | ¥
                               RSPI_SPCR2_SCKASE_DEF)
  #define RSPI_SPCR2_HIGH_SPEED   (RSPI_SPCR2_SPPE_DEF |
RSPI_SPCR2_SPOE_DEF | RSPI_SPCR2_SPIIE_DEF | RSPI_SPCR2_PTE_DEF | ¥
                                RSPI_SPCR2_SCKASE_ENABLE)
```

ファイル：¥src¥smc_gen¥r_rspi_rx¥src¥r_rspi_defaults.h

修正前

```
static rspi_ctrl_reg_values_t g_ctrl_reg_values[] =
{
    RSPI_SPCR_DEF,          // Control Register (SPCR)
    RSPI_SSLP_DEF,          // Slave Select Polarity Register (SSLP)
    RSPI_SPPCR_DEF,         // Pin Control Register (SPPCR)
    RSPI_SPSCR_DEF,         // Sequence Control Register (SPSCR)
    RSPI_SPBR_DEF,          // Bit Rate Register (SPBR)
    RSPI_SPDCR_DEF,         // Data Control Register (SPDCR)
    RSPI_SPCKD_DEF,         // Clock Delay Register (SPCKD)
    RSPI_SSLND_DEF,         // Slave Select Negation Delay Register (SSLND)
    RSPI_SPND_DEF,          // Next-Access Delay Register (SPND)
    RSPI_SPCR2_DEF,         // Control Register 2 (SPCR2)
    RSPI_SPDCR2_DEF,        // Data Control Register 2 (SPDCR2)
```

修正後(赤字部分を追加)

```
static rspi_ctrl_reg_values_t g_ctrl_reg_values[] =
{
    RSPI_SPCR_DEF,          // Control Register (SPCR)
    RSPI_SSLP_DEF,          // Slave Select Polarity Register (SSLP)
    RSPI_SPPCR_DEF,         // Pin Control Register (SPPCR)
    RSPI_SPSCR_DEF,         // Sequence Control Register (SPSCR)
    RSPI_SPBR_DEF,          // Bit Rate Register (SPBR)
    RSPI_SPDCR_DEF,         // Data Control Register (SPDCR)
    RSPI_SPCKD_DEF,         // Clock Delay Register (SPCKD)
    RSPI_SSLND_DEF,         // Slave Select Negation Delay Register (SSLND)
    RSPI_SPND_DEF,          // Next-Access Delay Register (SPND)
#if RSPI_CFG_HIGH_SPEED_READ == 1
        RSPI_SPCR2_HIGH_SPEED, // Control Register 2 (SPCR2) High-speed mode
#else
        RSPI_SPCR2_DEF,     // Control Register 2 (SPCR2) Default-speed mode
#endif
    RSPI_SPDCR2_DEF,        // Data Control Register 2 (SPDCR2)
```

修正前

```
#if    RSPI_NUM_CHANNELS > 1
       RSPI_SPCR_DEF, // Control Register (SPCR)
       RSPI_SSLP_DEF,          // Slave Select Polarity Register (SSLP)
       RSPI_SPPCR_DEF,         // Pin Control Register (SPPCR)
       RSPI_SPSCR_DEF,         // Sequence Control Register (SPSCR)
       RSPI_SPBR_DEF,          // Bit Rate Register (SPBR)
       RSPI_SPDCR_DEF,         // Data Control Register (SPDCR)
       RSPI_SPCKD_DEF,         // Clock Delay Register (SPCKD)
       RSPI_SSLND_DEF,         // Slave Select Negation Delay Register (SSLND)
       RSPI_SPND_DEF,          // Next-Access Delay Register (SPND)
       RSPI_SPCR2_DEF,         // Control Register 2 (SPCR2)
       RSPI_SPDCR2_DEF,        // Data Control Register 2 (SPDCR2)
#endif
```

修正後(赤字部分を追加)

```
#if    RSPI_NUM_CHANNELS > 1
       RSPI_SPCR_DEF, // Control Register (SPCR)
       RSPI_SSLP_DEF,          // Slave Select Polarity Register (SSLP)
       RSPI_SPPCR_DEF,         // Pin Control Register (SPPCR)
       RSPI_SPSCR_DEF,         // Sequence Control Register (SPSCR)
       RSPI_SPBR_DEF,          // Bit Rate Register (SPBR)
       RSPI_SPDCR_DEF,         // Data Control Register (SPDCR)
       RSPI_SPCKD_DEF,         // Clock Delay Register (SPCKD)
       RSPI_SSLND_DEF,         // Slave Select Negation Delay Register (SSLND)
       RSPI_SPND_DEF,          // Next-Access Delay Register (SPND)
#if RSPI_CFG_HIGH_SPEED_READ == 1
           RSPI_SPCR2_HIGH_SPEED, // Control Register 2 (SPCR2) High-speed mode
#else
           RSPI_SPCR2_DEF,     // Control Register 2 (SPCR2) Default-speed mode
#endif
       RSPI_SPDCR2_DEF,        // Data Control Register 2 (SPDCR2)
#endif
```

修正前

```
#if    RSPI_NUM_CHANNELS >2
       RSPI_SPCR_DEF, // Control Register (SPCR)
       RSPI_SSLP_DEF,          // Slave Select Polarity Register (SSLP)
       RSPI_SPPCR_DEF,         // Pin Control Register (SPPCR)
       RSPI_SPSCR_DEF,         // Sequence Control Register (SPSCR)
       RSPI_SPBR_DEF,          // Bit Rate Register (SPBR)
       RSPI_SPDCR_DEF,         // Data Control Register (SPDCR)
       RSPI_SPCKD_DEF,         // Clock Delay Register (SPCKD)
       RSPI_SSLND_DEF,         // Slave Select Negation Delay Register (SSLND)
       RSPI_SPND_DEF,          // Next-Access Delay Register (SPND)
       RSPI_SPCR2_DEF,         // Control Register 2 (SPCR2)
       RSPI_SPDCR2_DEF,        // Data Control Register 2 (SPDCR2)
#endif
```

修正後(赤字部分を追加)

```
#if    RSPI_NUM_CHANNELS >2
       RSPI_SPCR_DEF, // Control Register (SPCR)
       RSPI_SSLP_DEF,          // Slave Select Polarity Register (SSLP)
       RSPI_SPPCR_DEF,         // Pin Control Register (SPPCR)
       RSPI_SPSCR_DEF,         // Sequence Control Register (SPSCR)
       RSPI_SPBR_DEF,          // Bit Rate Register (SPBR)
       RSPI_SPDCR_DEF,         // Data Control Register (SPDCR)
       RSPI_SPCKD_DEF,         // Clock Delay Register (SPCKD)
       RSPI_SSLND_DEF,         // Slave Select Negation Delay Register (SSLND)
       RSPI_SPND_DEF,          // Next-Access Delay Register (SPND)
#if RSPI_CFG_HIGH_SPEED_READ == 1
            RSPI_SPCR2_HIGH_SPEED, // Control Register 2 (SPCR2) High-speed mode
#else
            RSPI_SPCR2_DEF,     // Control Register 2 (SPCR2) Default-speed mode
#endif
       RSPI_SPDCR2_DEF,        // Data Control Register 2 (SPDCR2)
#endif
```

(2)　受信データを正しく受け取るための回避策

ファイル：¥src¥smc_gen¥r_rspi_rx¥src¥r_rspi_defaults.h

修正前

```
R_BSP_ATTRIB_STATIC_INTERRUPT void rspi_spti0_isr(void)
{
    if (RSPI_TRANS_MODE_SW == g_rspi_tcb[0].data_tran_mode)
    {
        g_rxdata[0] = RSPI0.SPDR.LONG; // Read rx-data register into temp buffer.

            /* If master mode then disable further spti interrupts on first transmit.
            If slave mode then we do two transmits to fill the double buffer,
            then disable spti interrupts.
            The receive interrupt will handle any remaining data. */
#if RSPI_CFG_HIGH_SPEED_READ == 0
            if ((RSPI0.SPCR.BIT.MSTR) || (g_rspi_tcb[0].tx_count > 0))
            {
```

修正後(赤字を追加)

```
R_BSP_ATTRIB_STATIC_INTERRUPT void rspi_spti0_isr(void)
{
    if (RSPI_TRANS_MODE_SW == g_rspi_tcb[0].data_tran_mode)
    {
        if (0 == g_rspi_tcb[0].tx_count)
        {
            g_rxdata[0] = RSPI0.SPDR.LONG; // Read rx-data register into temp buffer.
        }
            /* If master mode then disable further spti interrupts on first transmit.
            If slave mode then we do two transmits to fill the double buffer,
            then disable spti interrupts.
            The receive interrupt will handle any remaining data. */
#if RSPI_CFG_HIGH_SPEED_READ == 0
            if ((RSPI0.SPCR.BIT.MSTR) || (g_rspi_tcb[0].tx_count > 0))
            {
```

修正前

```
R_BSP_ATTRIB_STATIC_INTERRUPT void rspi_spti1_isr(void)
{
    if (RSPI_TRANS_MODE_SW == g_rspi_tcb[1].data_tran_mode)
    {
        g_rxdata[1] = RSPI1.SPDR.LONG; // Read rx-data register into temp buffer.

            /* If master mode then disable further spti interrupts on first transmit.
            If slave mode then we do two transmits to fill the double buffer,
            then disable spti interrupts.
            The receive interrupt will handle any remaining data. */
#if RSPI_CFG_HIGH_SPEED_READ == 0
            if ((RSPI1.SPCR.BIT.MSTR) || (g_rspi_tcb[1].tx_count > 0))
            {
```

修正後(赤字を追加)

```
R_BSP_ATTRIB_STATIC_INTERRUPT void rspi_spti1_isr(void)
{
    if (RSPI_TRANS_MODE_SW == g_rspi_tcb[1].data_tran_mode)
    {
        if(0 == g_rspi_tcb[1].tx_count)
        {
            g_rxdata[1] = RSPI1.SPDR.LONG; // Read rx-data register into temp buffer.
        }
            /* If master mode then disable further spti interrupts on first transmit.
            If slave mode then we do two transmits to fill the double buffer,
            then disable spti interrupts.
            The receive interrupt will handle any remaining data. */
#if RSPI_CFG_HIGH_SPEED_READ == 0
            if ((RSPI1.SPCR.BIT.MSTR) || (g_rspi_tcb[1].tx_count > 0))
            {
```

修正前

```
R_BSP_ATTRIB_STATIC_INTERRUPT void rspi_spti2_isr(void)
{
    if (RSPI_TRANS_MODE_SW == g_rspi_tcb[2].data_tran_mode)
    {
        g_rxdata[2] = RSPI2.SPDR.LONG; // Read rx-data register into temp buffer.
            /* If master mode then disable further spti interrupts on first transmit.
            If slave mode then we do two transmits to fill the double buffer,
            then disable spti interrupts.
            The receive interrupt will handle any remaining data. */
#if RSPI_CFG_HIGH_SPEED_READ == 0
            if ((RSPI2.SPCR.BIT.MSTR) || (g_rspi_tcb[2].tx_count > 0))
            {
```

修正後(赤字を追加)

```
R_BSP_ATTRIB_STATIC_INTERRUPT void rspi_spti2_isr(void)
{
    if (RSPI_TRANS_MODE_SW == g_rspi_tcb[2].data_tran_mode)
    {
        g_rxdata[2] = RSPI2.SPDR.LONG; // Read rx-data register into temp buffer.
        if (0 == g_rspi_tcb[2].tx_count)
        {
            g_rxdata[2] = RSPI2.SPDR.LONG; // Read rx-data register into temp buffer.
        }
            /* If master mode then disable further spti interrupts on first transmit.
            If slave mode then we do two transmits to fill the double buffer,
            then disable spti interrupts.
            The receive interrupt will handle any remaining data. */
#if RSPI_CFG_HIGH_SPEED_READ == 0
            if ((RSPI2.SPCR.BIT.MSTR) || (g_rspi_tcb[2].tx_count > 0))
            {
```

1.5　恒久対策

今後のバージョンで改修予定です。

以上

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Mey.16.21	-	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ものですが、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。

TS Colophon 4.2