R20TS0133EJ0100
Rev.1.00
Mar. 1, 2017

[Notes]
RX Family
SCI Multi-Mode Module Using Firmware Integration Technology Rev.1.80, RX Driver Package Ver.1.11

Outline

When using the RX Family SCI Multi-Mode Module Using Firmware Integration Technology Rev.1.80 (referred to as SCI FIT module hereafter) and RX Driver Package Ver.1.11, note the following point.

1. Transmission and reception processing of the SCI FIT module in clock-synchronous mode

1. Transmission and Reception Processing of the SCI FIT Module in Clock-Synchronous Mode

1.1 Applicable Products

- RX Family SCI Multi-Mode Module Using Firmware Integration Technology

  Revision: Rev.1.80

  The following is the document for the product.

  - RX Family SCI Multi-Mode Module Using Firmware Integration Technology Application Note

    Document number: R01AN1815EJ0180

The problem also applies to the following product which includes the above SCI FIT module*.

*: The SCI FIT module is included as r_sci_rx_v1.80.zip.

- RX Family RX Driver Package Ver.1.11

  Revision: Rev.1.11

  The following is the document for the product.

  - RX Family RX Driver Package Ver.1.11 Application Note

    Document number: R01AN3467EJ0111

1.2 Applicable MCUs

RX110, RX111, RX113, RX130, RX210, RX230, RX231, RX23T, RX24T, RX63N, RX631, RX64M, RX651, RX65N, and RX71M groups

1.3 Details

During transmission and reception processing of the SCI FIT module in clock-synchronous mode, more data than the specified number of items might be sent or received.

1.4 Condition

The problem arises when any of the following conditions is met.

(1) When receiving data
The problem arises if the time required for communication for one frame has elapsed during the period from the time dummy data for reception was written to the time the counter is decremented*.

*: If FIFO mode is used, the interval indicated by the blue dashed arrow applies.
If FIFO mode is not used, the interval indicated by the red dashed arrow applies.

```
static sci_err_t sci_receive_sync_data(sci_hdl_t const hdl,
                                       uint8_t        *p_dst,
                                       uint16_t const  length)
{
(Omit)
#if SCI_CFG_FIFO_INCLUDED
       if (true == hdl->fifo_ctrl)
       {
(Omit)
           for (cnt=0; cnt<thresh_cnt; cnt++)
           {                                                  Write dummy data
               SCI_TDR(SCI_CFG_DUMMY_TX_BYTE);
           }

           hdl->tx_cnt -= cnt;                                Decrement the counter
       }
       else
#endif
(Omit)
       {
           SCI_TDR(SCI_CFG_DUMMY_TX_BYTE);                    Write dummy data (first time)
           hdl->tx_cnt--;                                     Decrement the counter (first time)

           if (1 < length)
           {
               dummy = hdl->rom->regs->TDR;
               if (0x80 == (hdl->rom->regs->SSR.BYTE & 0x80))
               {
                   SCI_TDR(SCI_CFG_DUMMY_TX_BYTE);            Write dummy data (second time)
                   hdl->tx_cnt--;                             Decrement the counter (second time)
               }
           }
       }
(Omit)
```

(2) When sending data
The problem arises if the time required for communication for one frame has elapsed during the period from the time transmission data was written to the time the counter is decremented*.

*: If FIFO mode is used, the interval indicated by the blue dashed arrow applies.
If FIFO mode is not used, the interval indicated by the red dashed arrow applies.

```
static sci_err_t sci_send_sync_data(sci_hdl_t const hdl,
                                    uint8_t        *p_src,
                                    uint8_t        *p_dst,
                                    uint16_t const  length,
                                    bool           save_rx_data)
{
(Omit)
#if SCI_CFG_FIFO_INCLUDED
       if (true == hdl->fifo_ctrl)
       {
(Omit)
           for (cnt=0; cnt<thresh_cnt; cnt++)
           {
               SCI_TDR(*hdl->u_tx_data.buf++);   <- Write data
           }

           hdl->tx_cnt -= cnt;   <- Decrement the counter
       }
       else
#endif
(Omit)
       {
           SCI_TDR(*hdl->u_tx_data.buf++);   <- Write data

           hdl->tx_cnt--;   <- Decrement the counter
       }
(Omit)
```

1.5 Workaround

Modify functions "sci_receive_sync_data()" and "sci_send_sync_data()" in the SCI FIT module source code "r_sci_rx.c" as described below.

The details of the modification are as follows. Modify the processing in blue to the processing in red for each function.

(1) In the case of Condition (1) in 1.4

■ "sci_receive_sync_data()" in the SCI FIT module source code "r_sci_rx.c"

Before modification: Excerpt of the part of processing (two portions) that sets dummy data in the TDR register and controls the counter.

```
static sci_err_t sci_receive_sync_data(sci_hdl_t const hdl,
                                       uint8_t        *p_dst,
                                       uint16_t const  length)
{
(Omit)
#if SCI_CFG_FIFO_INCLUDED
       if (true == hdl->fifo_ctrl)
       {
(Omit)
           if (length > SCI_FIFO_FRAME_SIZE)
           {
               thresh_cnt = SCI_FIFO_FRAME_SIZE;
           }
           else
           {
               if (length < hdl->rx_dflt_thresh)
               {
                   hdl->rom->regs->FCR.BIT.RTRG = length;
               }
               thresh_cnt = length;
           }

           for (cnt=0; cnt<thresh_cnt; cnt++)
           {
               SCI_TDR(SCI_CFG_DUMMY_TX_BYTE);
           }

           hdl->tx_cnt -= cnt;
       }
       else
#endif
(Omit)
       {
           SCI_TDR(SCI_CFG_DUMMY_TX_BYTE);
           hdl->tx_cnt--;

           if (1 < length)
           {
               dummy = hdl->rom->regs->TDR;
               if (0x80 == (hdl->rom->regs->SSR.BYTE & 0x80))
               {
                   SCI_TDR(SCI_CFG_DUMMY_TX_BYTE);
                   hdl->tx_cnt--;
               }
           }
       }
(Omit)
```

After modification:

```
static sci_err_t sci_receive_sync_data(sci_hdl_t const hdl,
                                       uint8_t        *p_dst,
                                       uint16_t const  length)
{
(Omit)
#if SCI_CFG_FIFO_INCLUDED
       if (true == hdl->fifo_ctrl)
       {
(Omit)

           if (length > SCI_FIFO_FRAME_SIZE)
           {
               thresh_cnt = SCI_FIFO_FRAME_SIZE;
           }
           else
           {
               if (length < hdl->rx_dflt_thresh)
               {
                   hdl->rom->regs->FCR.BIT.RTRG = length;
               }
               thresh_cnt = length;
           }

           hdl->tx_cnt -= thresh_cnt;

           for (cnt=0; cnt<thresh_cnt; cnt++)
           {
               SCI_TDR(SCI_CFG_DUMMY_TX_BYTE);
           }
       }
       else
#endif
(Omit)
       {
           hdl->tx_cnt--;
           SCI_TDR(SCI_CFG_DUMMY_TX_BYTE);
       }
(Omit)
```

(2) In the case of Condition (2) in 1.4

■ "sci_send_sync_data()" in the SCI FIT module source code "r_sci_rx.c"

Before modification: Excerpt of the part of processing (two portions) that sets data in the TDR register and controls the counter.

```
static sci_err_t sci_send_sync_data(sci_hdl_t const hdl,
                                    uint8_t        *p_src,
                                    uint8_t        *p_dst,
                                    uint16_t const  length,
                                    bool           save_rx_data)
{
(Omit)
#if SCI_CFG_FIFO_INCLUDED
       if (true == hdl->fifo_ctrl)
       {
(Omit)
           thresh_cnt = hdl->rom->regs->FCR.BIT.RTRG;

           for (cnt=0; cnt<thresh_cnt; cnt++)
           {
               SCI_TDR(*hdl->u_tx_data.buf++);
           }

           hdl->tx_cnt -= cnt;
       }
       else
#endif
(Omit)
       {
           SCI_TDR(*hdl->u_tx_data.buf++);
           hdl->tx_cnt--;
       }
(Omit)
```

After modification:

```
static sci_err_t sci_send_sync_data(sci_hdl_t const hdl,
                                    uint8_t        *p_src,
                                    uint8_t        *p_dst,
                                    uint16_t const  length,
                                    bool           save_rx_data)
{
(Omit)
#if SCI_CFG_FIFO_INCLUDED
       if (true == hdl->fifo_ctrl)
       {
(Omit)
           thresh_cnt = hdl->rom->regs->FCR.BIT.RTRG;

           hdl->tx_cnt -= thresh_cnt;

           for (cnt=0; cnt<thresh_cnt; cnt++)
           {
               SCI_TDR(*hdl->u_tx_data.buf++);
           }
       }
       else
#endif
(Omit)
       {
           hdl->tx_cnt--;
           SCI_TDR(*hdl->u_tx_data.buf++);
       }
(Omit)
```

1.6 Schedule for Fixing the Problem

- SCI FIT module

  This problem will be fixed in the next revision.

- RX Family RX Driver Package

  The SCI FIT module (Rev.1.90) modified in accord with this note will be included in Ver.1.12 of the RX Family RX Driver Package, which will be the next release.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Mar. 1, 2017	-	First edition issued

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061 Japan
Renesas Electronics Corporation

■Inquiry
https://www.renesas.com/contact/