[Notes]
RX Driver Package, RX Family RSPI Module Using Firmware Integration Technology

R20TS0720EJ0100
Rev.1.00
Jul.01, 2021

Overview

When using the RX Driver Package and RX Family RSPI Module Using Firmware Integration Technology (referred to as RSPI FIT module hereafter), note the following point.

1. Notes on Calling the R_RSPI_Write, R_RSPI_Read, or R_RSPI_WriteRead Function

1. Notes on Calling the R_RSPI_Write, R_RSPI_Read, or R_RSPI_WriteRead Function

1.1 Applicable Products

(1) RX Family RSPI Module Using Firmware Integration Technology

The relevant revisions and documents are as shown in "Table 1.1, RSPI FIT Module Applicable Products".

Table 1.1 RSPI FIT Module Applicable Products

Revision of the RSPI FIT module	Document number
Rev.1.00	R01AN1817EJ0100
Rev.1.20	R01AN1817EJ0120
Rev.1.30	R01AN1817EJ0130
Rev.1.40	R01AN1817EJ0140
Rev.1.50	R01AN1817EJ0150
Rev.1.60	R01AN1817EJ0160
Rev.1.70	R01AN1817EJ0170
Rev.1.80	R01AN1817EJ0180
Rev.2.00	R01AN1817EJ0200
Rev.2.01	R01AN1817EJ0201
Rev.2.02	R01AN1817EJ0202
Rev.2.03	R01AN1817EJ0203
Rev.2.04	R01AN1817EJ0204
Rev.2.05	R01AN1817EJ0205
Rev.3.00	R01AN1817EJ0300

These notes also apply to the following RX Driver Package products that include the above RSPI FIT module[*1].

1: The RSPI FIT module is included as r_rspi_rx_v.**.zip (*.** is the revision number).

(2) RX Driver Package

The product name and revision of the relevant RX Driver Package products and the revision of the included RSPI FIT module are as shown in "Table 1.2 Products Which Include the RSPI FIT Module".

Table 1.2 Products Which Include the RSPI FIT Module

Product name of the RX Driver Package	Revision of the RX Driver Package	Document number	Revision of the included RSPI FIT module
RX110, RX111, RX113, and RX231 Group RX Driver Package Ver.1.01	Rev.1.01	R01AN2670EJ0101	Rev.1.40
RX110, RX111, RX113, RX130, RX231, and RX23T Group RX Driver Package Ver.1.02	Rev.1.02	R01AN3159EJ0102	Rev.1.40
RX110, RX111, RX113, RX130, RX230, RX231, RX23T, and RX24T Group RX Driver Package Ver.1.03	Rev.1.03	R01AN3233EJ0103	Rev.1.40
RX64M, RX71M Group RX Driver Package Ver.1.02	Rev.1.04	R01AN2606EJ0104	Rev1.30
RX Family RX Driver Package Ver.1.11	Rev.1.11	R01AN3467EJ0111	Rev.1.50
RX Family RX Driver Package Ver.1.12	Rev.1.12	R01AN3651EJ0112	Rev.1.60
RX Family RX Driver Package Ver.1.13	Rev.1.13	R01AN3859EJ0113	Rev.1.70
RX Family RX Driver Package Ver.1.14	Rev.1.14	R01AN4191EJ0114	Rev.1.70
RX Family RX Driver Package Ver.1.15	Rev.1.15	R01AN4372EJ0115	Rev.1.70
RX Family RX Driver Package Ver.1.16	Rev.1.16	R01AN4471EJ0116	Rev.1.80
RX Family RX Driver Package Ver.1.17	Rev.1.17	R01AN4572EJ0117	Rev.1.80
RX Family RX Driver Package Ver.1.18	Rev.1.18	R01AN4659EJ0118	Rev.1.80
RX Family RX Driver Package Ver.1.19	Rev.1.19	R01AN4677EJ0119	Rev.2.00
RX Family RX Driver Package Ver.1.20	Rev.1.20	R01AN4794EJ0120	Rev.2.01
RX Family RX Driver Package Ver.1.21	Rev.1.21	R01AN4843EJ0121	Rev.2.02
RX Family RX Driver Package Ver.1.22	Rev.1.22	R01AN4873EJ0122	Rev.2.03
RX Family RX Driver Package Ver.1.23	Rev.1.23	R01AN4976EJ0123	Rev.2.03
RX Family RX Driver Package Ver.1.24	Rev.1.24	R01AN5267EJ0124	Rev.2.04
RX Family RX Driver Package Ver.1.25	Rev.1.25	R01AN5371EJ0125	Rev.2.04

Product name of the RX Driver Package	Revision of the RX Driver Package	Document number	Revision of the included RSPI FIT module
RX Family RX Driver Package Ver.1.26	Rev.1.26	R01AN5401EJ0126	Rev.2.05
RX Family RX Driver Package Ver.1.27	Rev.1.27	R01AN5600EJ0127	Rev.3.00
RX Family RX Driver Package Ver.1.28	Rev.1.28	R01AN5761EJ0128	Rev.3.00
RX Family RX Driver Package Ver.1.29	Rev.1.29	R01AN5826EJ0129	Rev.3.00
RX Family RX Driver Package Ver.1.30	Rev.1.30	R01AN5882EJ0130	Rev.3.00

1.2 Applicable Devices

RX110, RX111, RX113, RX130, RX231, RX23E-A, RX23W, RX23T, RX24T, RX24U, RX64M, RX65N, RX651, RX66T, RX66N, RX71M, RX72T, RX72M, RX72N, RX210*1, RX62N*1, RX62T*1, RX63N*1, and RX631*1 Group

*1: RSPI FIT module Rev.2.01 or later does not support RX210, RX62N, RX62T, RX63N and RX631 groups.

1.3 Details

When the R_RSPI_Write, R_RSPI_Read, or R_RSPI_WriteRead function is called with the RSPI FIT module set to master mode operation, the RSPI function stops after the final data is received from a serial transfer without waiting for the SSL negation delay or next-access delay.

1.4 Conditions

This phenomenon occurs when the R_RSPI_Write, R_RSPI_Read, or R_RSPI_WriteRead function is called when the module is set to master mode operation.

1.5 Workaround

For RSPI FIT module versions earlier than Rev.2.00, modify the processing of the rspi_tx_rx_common function in the r_rspi_rx.c source code. For RSPI FIT module Rev.2.00 or later, modify the processing of the rspi_rx_common function in the r_rspi_rx.c source code.

The details of the modification are as follows. Add the codes in red.

Before modification (versions earlier than Rev.2.00)

```
    /* Check for last data.  */
     if(rx_count == g_rspi_tcb[channel].xfr_length)
     {   /* Last data was transferred. */
         (*g_rspi_channels[channel]).SPCR.BIT.SPRIE = 0;  /* Disable SPRI
interrupt. */
         (*g_rspi_channels[channel]).SPCR.BIT.SPE   = 0;  /* Disable RSPI. */
         if (0 == (*g_rspi_channels[channel]).SPCR.BIT.SPE)
         {
            nop();
         }

         #if RSPI_CFG_REQUIRE_LOCK == 1
         /* Release lock for this channel. */
         R_BSP_HardwareUnlock((mcu_lock_t)(BSP_LOCK_RSPI0 + channel));
         #endif

         /* Transfer complete. Call the user callback function passing
pointer to the result structure. */
         if((FIT_NO_FUNC != g_rspi_handles[channel].pcallback) && (NULL !=
g_rspi_handles[channel].pcallback))
         {
            g_rspi_cb_data[channel].handle = &(g_rspi_handles[channel]);
            g_rspi_cb_data[channel].event_code = RSPI_EVT_TRANSFER_COMPLETE;

g_rspi_handles[channel].pcallback((void*)&(g_rspi_cb_data[channel]));
         }
    }
```

After modification (versions earlier than Rev.2.00)

```
    /* Check for last data.  */
     if(rx_count == g_rspi_tcb[channel].xfr_length)
     {   /* Last data was transferred. */
         (*g_rspi_channels[channel]).SPCR.BIT.SPRIE = 0;  /* Disable SPRI
interrupt. */
        while(1==(*g_rspi_channels[channel]).SPSR.BIT.IDLNF)
        {
           nop();
        }

        (*g_rspi_channels[channel]).SPCR.BIT.SPE   = 0;  /* Disable RSPI. */
        if (0 == (*g_rspi_channels[channel]).SPCR.BIT.SPE)
        {
           nop();
        }

        #if RSPI_CFG_REQUIRE_LOCK == 1
        /* Release lock for this channel. */
        R_BSP_HardwareUnlock((mcu_lock_t)(BSP_LOCK_RSPI0 + channel));
        #endif

        /* Transfer complete. Call the user callback function passing
pointer to the result structure. */
        if((FIT_NO_FUNC != g_rspi_handles[channel].pcallback) && (NULL !=
g_rspi_handles[channel].pcallback))
        {
           g_rspi_cb_data[channel].handle = &(g_rspi_handles[channel]);
           g_rspi_cb_data[channel].event_code = RSPI_EVT_TRANSFER_COMPLETE;

g_rspi_handles[channel].pcallback((void*)&(g_rspi_cb_data[channel]));
        }
    }
```

Before modification (Rev.2.00 or later)

```
    /* Check for last data.  */
     if (rx_count == g_rspi_tcb[channel].xfr_length)
     {   /* Last data was transferred. */
#if RSPI_CFG_HIGH_SPEED_READ == 1
        (*g_rspi_channels[channel]).SPCR.BIT.SPTIE = 0;  // Disable SPTI
interrupt.
#endif
        (*g_rspi_channels[channel]).SPCR.BIT.SPRIE = 0;  // Disable SPRI
interrupt.
        (*g_rspi_channels[channel]).SPCR.BIT.SPE   = 0;  // Disable RSPI.
        if (0 == (*g_rspi_channels[channel]).SPCR.BIT.SPE)
        {
           nop();
        }
        #if RSPI_CFG_REQUIRE_LOCK == 1
        /* Release lock for this channel. */
        R_BSP_HardwareUnlock((mcu_lock_t)(BSP_LOCK_RSPI0 + channel));
        #endif


        /* Transfer complete. Call the user callback function passing
pointer to the result structure. */
        if ((FIT_NO_FUNC != g_rspi_handles[channel].pcallback) && (NULL !=
g_rspi_handles[channel].pcallback))
        {
           g_rspi_cb_data[channel].handle = &(g_rspi_handles[channel]);
           g_rspi_cb_data[channel].event_code = RSPI_EVT_TRANSFER_COMPLETE;

g_rspi_handles[channel].pcallback((void*)&(g_rspi_cb_data[channel]));
        }
    }
```

After modification (Rev.2.00 or later)

```
    /* Check for last data.  */
     if (rx_count == g_rspi_tcb[channel].xfr_length)
     {   /* Last data was transferred. */
#if RSPI_CFG_HIGH_SPEED_READ == 1
        (*g_rspi_channels[channel]).SPCR.BIT.SPTIE = 0;  // Disable SPTI
interrupt.
#endif
        (*g_rspi_channels[channel]).SPCR.BIT.SPRIE = 0;  // Disable SPRI
interrupt.
        while(1==(*g_rspi_channels[channel]).SPSR.BIT.IDLNF)
        {
           nop();
        }
        (*g_rspi_channels[channel]).SPCR.BIT.SPE   = 0;  // Disable RSPI.
        if (0 == (*g_rspi_channels[channel]).SPCR.BIT.SPE)
        {
           nop();
        }

        #if RSPI_CFG_REQUIRE_LOCK == 1
        /* Release lock for this channel. */
        R_BSP_HardwareUnlock((mcu_lock_t)(BSP_LOCK_RSPI0 + channel));
        #endif

        /* Transfer complete. Call the user callback function passing
pointer to the result structure. */
        if ((FIT_NO_FUNC != g_rspi_handles[channel].pcallback) && (NULL !=
g_rspi_handles[channel].pcallback))
        {
           g_rspi_cb_data[channel].handle = &(g_rspi_handles[channel]);
           g_rspi_cb_data[channel].event_code = RSPI_EVT_TRANSFER_COMPLETE;

g_rspi_handles[channel].pcallback((void*)&(g_rspi_cb_data[channel]));
        }
    }
```

1.6 Schedule for fixing the problem

(1) RSPI FIT module

This problem will be fixed in a later version.

(2) RX Driver Package

The upcoming release of the RX Family RX Driver Package will include an RSPI FIT module modified in accord with this note.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jul.01.21	-	First edition issued

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.

Trademarks

TS Colophon 4.2