[Notes]

RX Family

RSPI Module Firmware Integration Technology,

RX Driver Package

R20TS0590EJ0100
Rev.1.00
Jul. 01, 2020

Outline

When using the products in the title, note the following point.

1. Notes on callback functions when RSPI and DMAC are combined

1. Notes on Callback Functions When RSPI and DMAC Are Combined

1.1 Applicable Products

(1) RSPI module Firmware Integration Technology (RSPI FIT module)

The applicable revision numbers and document numbers are as follows.

Table 1.1 RSPI FIT module applicable products

RSPI FIT module revision number	Document number
Rev.2.05	R01AN1827EJ0205
Rev.2.04	R01AN1827EJ0204
Rev.2.03	R01AN1827EJ0203
Rev.2.02	R01AN1827EJ0202
Rev.2.01	R01AN1827EJ0201
Rev.2.00	R01AN1827EJ0200

(2) RX Driver Package

The RSPI FIT module in (1) is also included in the RX Driver Package.
The product names and revision numbers of the applicable RX Driver Package and the revision numbers of the RSPI FIT module are as follows.

Table 1.2 Products which include the RSPI FIT module

RX Driver Package product name	RX Driver Package revision number	Document number	Revision number 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.1.25	Rev.1.25	R01AN5371EJ0125	Rev.2.05
RX Family RX Driver Package Ver.1.24	Rev.1.24	R01AN5267EJ0124	Rev.2.04
RX Family RX Driver Package Ver.1.23	Rev.1.23	R01AN4976EJ0123	Rev.2.03
RX Family RX Driver Package, Ver.1.22	Rev.1.22	R01AN4873EJ0122	Rev.2.03
RX Family RX Driver Package Ver.1.20	Rev.1.20	R01AN4794EJ0120	Rev.2.01
RX Family RX Driver Package Ver.1.19	Rev.1.19	R01AN4677EJ0119	Rev.2.00

1.2 Applicable Devices

RX110, RX111, RX113, and RX130 groups

RX230, RX231, RX23E-A, RX23W, RX23T, RX24T, and RX24U groups

RX64M, RX651, RX65N, and RX66T groups

RX71M, RX72T, RX72M, and RX72N groups

1.3 Details and Conditions

We have identified an error in the processing of the DMAC callback function in the sample program. Therefore, either or both of the following problems may occur if you create a callback function according to the sample program code and perform any one of the R_RSPI_Write, R_RSPI_Read, and R_RSPI_WriteRead functions, thus combining RSPI with DMAC.

(1) Second communication does not start.

If the program lacks processing for writing 0 to the SPE bit, the second and subsequent communications will not start.

(2) Missing data occurs in transmission or reception.

An internal transmit buffer empty interrupt request may persist. In this case, missing transmission or reception data may occur in the subsequent communications.

1.4 Workarounds

If you use it in combination with DMAC, add each of the following processing to the callback function.

The following description is based on the sample program for RSPI FIT module Rev.2.05 as an example.

(1) Workaround for when the second communication does not start

If you use it in combination with DMAC, write 0 to the SPE bit in the user DMAC callback function, which is called at the end of communication.

Before modification

```
void DMA_CallBack_R(void)
{
    volatile dmaca_return_t     ret_dmaca;
    dmaca_stat_t                p_stat_dmaca;

    my_rspi_handle->channel = RSPI_CHANNEL;

    /* check DMA end */
    /*** DMACA transfer end check ***/
    ret_dmaca = R_DMACA_Control(DMACA_CH1, DMACA_CMD_STATUS_GET,
(dmaca_stat_t*)&p_stat_dmaca);
    if (DMACA_SUCCESS != ret_dmaca)
    {
        return;
    }

    if (false != (p_stat_dmaca.dtif_stat))
    {
        ret_dmaca = R_DMACA_Control(DMACA_CH1, DMACA_CMD_DTIF_STATUS_CLR,
(dmaca_stat_t*)&p_stat_dmaca);
        R_RSPI_IntSpriIerClear(my_rspi_handle);
        R_RSPI_IntSpriDmacdtcFlagSet(my_rspi_handle, RSPI_SET_TRANS_STOP);
    }

    if (false != (p_stat_dmaca.esif_stat))
    {
        ret_dmaca = R_DMACA_Control(DMACA_CH1, DMACA_CMD_ESIF_STATUS_CLR,
(dmaca_stat_t*)&p_stat_dmaca);
    }

    transfer_busy = false;
    return;
}
```

After modification (Add the parts shown in red.)

```
void DMA_CallBack_R(void)
{
    volatile dmaca_return_t     ret_dmaca;
    dmaca_stat_t                p_stat_dmaca;

    my_rspi_handle->channel = RSPI_CHANNEL;

    /* check DMA end */
    /*** DMACA transfer end check ***/
    ret_dmaca = R_DMACA_Control(DMACA_CH1, DMACA_CMD_STATUS_GET,
(dmaca_stat_t*)&p_stat_dmaca);
    if (DMACA_SUCCESS != ret_dmaca)
    {
        return;
    }

    if (false != (p_stat_dmaca.dtif_stat))
    {
        ret_dmaca = R_DMACA_Control(DMACA_CH1, DMACA_CMD_DTIF_STATUS_CLR,
(dmaca_stat_t*)&p_stat_dmaca);
        R_RSPI_IntSpriIerClear(my_rspi_handle);
        RSPI0.SPCR.BIT.SPE    = 0;
        R_RSPI_IntSpriDmacdtcFlagSet(my_rspi_handle, RSPI_SET_TRANS_STOP);
    }

    if (false != (p_stat_dmaca.esif_stat))
    {
        ret_dmaca = R_DMACA_Control(DMACA_CH1, DMACA_CMD_ESIF_STATUS_CLR,
(dmaca_stat_t*)&p_stat_dmaca);
    }

    transfer_busy = false;
    return;
}
```

(2) Workaround for when missing data occurs in transmission or reception

If you use it in combination with DMAC, write 0 to the SPTIE bit in the user DMAC callback function, which is called at the completion of DMAC transmission.

Before modification

```
void DMA_CallBack_W(void)
{
    volatile dmaca_return_t     ret_dmaca;
    dmaca_stat_t                p_stat_dmaca;

    my_rspi_handle->channel = RSPI_CHANNEL;

    /* check DMA end */
    /*** DMACA transfer end check ***/
    ret_dmaca = R_DMACA_Control(DMACA_CH0, DMACA_CMD_STATUS_GET,
(dmaca_stat_t*)&p_stat_dmaca);
    if (DMACA_SUCCESS != ret_dmaca)
    {
        return;
    }

    if (false != (p_stat_dmaca.dtif_stat))
    {
        ret_dmaca = R_DMACA_Control(DMACA_CH0, DMACA_CMD_DTIF_STATUS_CLR,
(dmaca_stat_t*)&p_stat_dmaca);
        R_RSPI_IntSptiIerClear(my_rspi_handle);
        R_RSPI_IntSptiDmacdtcFlagSet(my_rspi_handle, RSPI_SET_TRANS_STOP);
    }

    if (false != (p_stat_dmaca.esif_stat))
    {
        ret_dmaca = R_DMACA_Control(DMACA_CH0, DMACA_CMD_ESIF_STATUS_CLR,
(dmaca_stat_t*)&p_stat_dmaca);
    }

    return;
}
```

After modification (Add the parts shown in red.)

```
void DMA_CallBack_W(void)
{
    volatile dmaca_return_t     ret_dmaca;
    dmaca_stat_t                p_stat_dmaca;

    my_rspi_handle->channel = RSPI_CHANNEL;

    /* check DMA end */
    /*** DMACA transfer end check ***/
    ret_dmaca = R_DMACA_Control(DMACA_CH0, DMACA_CMD_STATUS_GET,
(dmaca_stat_t*)&p_stat_dmaca);
    if (DMACA_SUCCESS != ret_dmaca)
    {
        return;
    }

    if (false != (p_stat_dmaca.dtif_stat))
    {
        ret_dmaca = R_DMACA_Control(DMACA_CH0, DMACA_CMD_DTIF_STATUS_CLR,
(dmaca_stat_t*)&p_stat_dmaca);
        RSPI0.SPCR.BIT.SPTIE = 0;
        R_RSPI_IntSptiIerClear(my_rspi_handle);
        R_RSPI_IntSptiDmacdtcFlagSet(my_rspi_handle, RSPI_SET_TRANS_STOP);
    }

    if (false != (p_stat_dmaca.esif_stat))
    {
        ret_dmaca = R_DMACA_Control(DMACA_CH0, DMACA_CMD_ESIF_STATUS_CLR,
(dmaca_stat_t*)&p_stat_dmaca);
    }

    return;
}
```

1.5 Schedule for Fixing the Problem

This problem will be fixed in the next version.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jul.01.20	-	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 4.1