Outline

When using the products in the title, note the following point.
1. Transfer-completed interrupt detection type for DMA controllers
2. Using DMA controllers with the serial communications interface with FIFO (SCIFA)

1. Transfer-Completed Interrupt Detection Type for DMA Controllers

1.1 Applicable Products
- V4.0.0.26 and later versions of e² studio (V2.0.0 and later versions of the Code Generator plug-in)
- V1.00.00 and later versions of the AP4 coding assistance tool for RZ

1.2 Applicable MCUs
- RZ family: RZ/T1 group

1.3 Details
If one of the following interrupts is set for the activation source for a DMA controller and the interrupt detection type is set to “level detection”, DMA transfer interrupt cannot be used.
- External interrupt
  - IRQn
    - n: Interrupt cause number
- External DMAC request
  - DMAINTn
    - n: Request signal number
- Internal peripheral module request
  - Ether PHY
    - ETHPHYIn
      - n: Interrupt cause number
  - SCIFA Unitn
    - BRIFn
    - RXIFn
    - TXIFn
    - DRIFn
      - n: Channel number
### 1.4 Workaround

In the activation source settings for the DMA controller, set the detection type to “Rising edge”.

![Figure 1 Example of Display when IRQ0 is Selected for the Activation Source for DMAC0](image)

If SCIFA is selected for the activation source, Rising edge cannot be selected for the detection type of the applicable product. In this case, modify the following functions, and then change the detection type to Rising edge. This modification is required every time code is generated.

- void R_SCIFA\_Create(void)* in the source file r_cg_scifa.c
- void r_scifa\_drifn\_interrupt(void)* and void r_scifa\_brifn\_interrupt(void)* in the source file r_cg_scifa_user.c

*: n means the channel number.
The following is an example of the required modification when SCIFA is selected for the activation source.

- Modification for changing the setting of transmission FIFO data empty 2 (TXIF2)
  Add the processing in red.

Before modification:

```c
/**
 * Function Name: R_SCIFA2_Create
 * Description : This function initializes SCIFA2.
 * Arguments   : None
 * Return Value : None
**************************************************************************/
void R_SCIFA2_Create(void)
{
    Omitted
    /* Disable TXIF2 interrupt */
    VIC.IEC3.LONG = 0x00008000UL;

    /* Disable RXIF2 interrupt */
    VIC.IEC3.LONG = 0x00004000UL;
    Omitted
}
```

After modification:

```c
/**
 * Function Name: R_SCIFA2_Create
 * Description : This function initializes SCIFA2.
 * Arguments   : None
 * Return Value : None
**************************************************************************/
void R_SCIFA2_Create(void)
{
    Omitted
    /* Disable TXIF2 interrupt */
    VIC.IEC3.LONG = 0x00008000UL;

    /* Set interrupt detection type */
    VIC.PLS3.LONG |= 0x00008000UL;

    /* Disable RXIF2 interrupt */
    VIC.IEC3.LONG = 0x00004000UL;
    Omitted
}
```
Modification for changing the setting of reception data ready 2 (DRIF2) and break detection, or overrun 2 (BRIF2)

Modify the processing in blue to the processing in red.

Before modification:

```c
/********************************************
* Function Name: r_scifa2_drif2_interrupt
* Description  : This function is TEIF 2 or DRIF2 interrupt service routine.
* Arguments    : None
* Return Value : None
********************************************
*/
void r_scifa2_drif2_interrupt(void)
{
    Omitted
    /* Wait the interrupt signal is disabled */
    while (0U != (VIC.IRQS3.LONG & 0x00010000UL))
    {
        VIC.IEC3.LONG = 0x00010000UL;
    }
    VIC.IEN3.LONG |= 0x00010000UL;

    Omitted
}

/********************************************
* Function Name: r_scifa2_brif2_interrupt
* Description  : This function is BRIF2 or ERIF2 interrupt service routine.
* Arguments    : None
* Return Value : None
********************************************
*/
void r_scifa2_brif2_interrupt(void)
{
    Omitted
    /* Wait the interrupt signal is disabled */
    while (0U != (VIC.IRQS3.LONG & 0x00002000UL))
    {
        VIC.IEC3.LONG = 0x00002000UL;
    }
    VIC.IEN3.LONG |= 0x00002000UL;

    Omitted
}
After modification:

```c
void r_scifa2_drif2_interrupt(void)
{
    // Omitted
    VIC.PIC3.LONG = 0x00010000UL;
    VIC.IEN3.LONG |= 0x00010000UL;
    // Omitted
}

void r_scifa2_brif2_interrupt(void)
{
    // Omitted
    VIC.PIC3.LONG = 0x00020000UL;
    VIC.IEN3.LONG |= 0x00002000UL;
    // Omitted
}
```

1.5 Schedule for Fixing the Problem

This problem will be fixed in the next version.
2. Using DMA Controllers with the Serial Communications Interface with FIFO (SCIFA)

2.1 Applicable Products
- V4.0.0.26 and later versions of e² studio (V2.0.0 and later versions of the Code Generator plug-in)
- V1.00.00 and later versions of the AP4 coding assistance tool for RZ

2.2 Applicable MCUs
- RZ family: RZ/T1 group

2.3 Details
If [Data handled by DMAC] is selected in the data handling setting for the serial communications interface with FIFO (SCIFA), a code that clears an interrupt request will not be output.
2.4 Workaround

Add the code that performs interrupt termination processing to the following functions. This modification is required every time code is generated.

- void r_scifa\_txifn\_interrupt(void)* and void r_scifa\_rxifn\_interrupt(void)* in the source file r_cg_scifa.c

*: \( n \) means the channel number.

The following is an example of the required modification.

- Modification for changing the channel 2 setting
  - Add the processing in red.

Before modification:

```c
void r_scifa2\_txif2\_interrupt(void)
{
    r_scifa2\_callback\_transmitend();
}

void r_scifa2\_rxif2\_interrupt(void)
{
    r_scifa2\_callback\_transmitend();
}
```

After modification:

```c
void r_scifa2\_txif2\_interrupt(void)
{
    /* Additional processing in red */
    r_scifa2\_callback\_transmitend();
}

void r_scifa2\_rxif2\_interrupt(void)
{
    /* Additional processing in red */
    r_scifa2\_callback\_transmitend();
}
```
After modification:

```c
void r_scifa2_txif2_interrupt(void)
{
    r_scifa2_callback_transmitend();
    VIC.PIC3.LONG = 0x00008000UL;
    VIC.IEN3.LONG |= 0x00008000UL;

    /* Dummy write */
    VIC.HVA0.LONG = 0x00000000UL;
    asm("dmb");
}

void r_scifa2_rxif2_interrupt(void)
{
    r_scifa2_callback_transmitend();
    VIC.PIC3.LONG = 0x00004000UL;
    VIC.IEN3.LONG |= 0x00004000UL;

    /* Dummy write */
    VIC.HVA0.LONG = 0x00000000UL;
    asm("dmb");
}
```

2.5 Schedule for Fixing the Problem

This problem will be fixed in the next version.
## Revision History

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Mar. 1, 2017</td>
<td>First edition issued</td>
</tr>
</tbody>
</table>

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

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

Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.

The past news contents have been based on information at the time of publication. Now changed or invalid information may be included. The URLs in the Tool News also may be subject to change or become invalid without prior notice.

All trademarks and registered trademarks are the property of their respective owners.