

Note on Using Renesas Peripheral Driver Library for RX62N/RX621 Groups and Peripheral Driver Generator - -With Using Serial Communication Interface (SCIa) of RX62N/RX621 Groups--

When using Renesas Peripheral Driver Library for the RX62N/RX621 groups and Peripheral Driver Generator, take note of the following problem:

- With using serial communication interface (SCIa) of RX62N/RX621 groups
-

1. Products and Versions Concerned

- RX62N, RX621 Group Renesas Peripheral Driver Library V.1.02
- Peripheral Driver Generator V.2.01 and later

2. Description

When an ERI interrupt is generated by an SCI interrupt source, the SSR.MPBT bit, which transfers the multiprocessor bit, may be cleared if the serial communication interface (SCIa) is used.

3. Condition

This problem arises if an ERI interrupt is generated by an SCI interrupt source, and the call-back function for reception errors is not used.

4. Workaround

To avoid this problem, modify the statement for clearing the SSR.MPBT bit.

This statement is found:

- (1) In the Interrupt_SCI.c file included in the peripheral driver library.
- (2) In the source¥RX¥RX62N¥i_src folder under the directory where the peripheral driver generator has been installed.

An example of modification is described below, which is common to both

the products concerned.

Example for SCIA Channel 0:

Source code before modified

```
-----  
void Interrupt_SCI0_ERI0(void)  
{  
    /* Will the user handle the errors? */  
    if (rpd1_SCI_RX_Error_callback_func[0] != PDL_NO_FUNC)  
    {  
        /* Call the error handler */  
        rpd1_SCI_RX_Error_callback_func[0]();  
    }  
    else  
    {  
        /* Clear the error flags */  
        SCI0.SSR.BYTE = (uint8_t)(BIT_7 | BIT_6);  
    }  
}
```

Source code after modified

```
-----  
void Interrupt_SCI0_ERI0(void)  
{  
    /* Will the user handle the errors? */  
    if (rpd1_SCI_RX_Error_callback_func[0] != PDL_NO_FUNC)  
    {  
        /* Call the error handler */  
        rpd1_SCI_RX_Error_callback_func[0]();  
    }  
    else  
    {  
        /* Modification begins here */  
        /* Clear the SSR error flags;  
        Preserve MPBT(b0) and write 1 to reserved bits (b6 and b7)*/  
        SCI0.SSR.BYTE = (uint8_t)((BIT_7 | BIT_6) |  
            (SCI0.SSR.BYTE & (uint8_t)BIT_0));  
        /* Modification ends here */  
    }  
}
```

5. Schedule of Fixing Problem

We plan to fix this problem in a version published later.

[Disclaimer]

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.

© 2010-2016 Renesas Electronics Corporation. All rights reserved.