[Notes] RX Family

Clock Synchronous Control Module for Serial NOR Flash Memory Access Firmware Integration Technology

RX Driver Package

Outline

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

1. When the functions listed in section 1.3 are used, error code returned might be incorrect.

1. When the functions listed in section 1.3 are used, error code returned might be incorrect.

1.1 Applicable Products

1) Clock Synchronous Control Module for Serial NOR Flash Memory Access (FLASH SPI FIT module)

The applicable revision numbers and document numbers are as follows:

<table>
<thead>
<tr>
<th>Revision number of the FLASH SPI FIT module</th>
<th>Document number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rev.3.03</td>
<td>R01AN2662EJ0303</td>
</tr>
<tr>
<td>Rev.3.02</td>
<td>R01AN2662EJ0302</td>
</tr>
<tr>
<td>Rev.3.01</td>
<td>R01AN2662EJ0301</td>
</tr>
<tr>
<td>Rev.3.00</td>
<td>R01AN2662EJ0300</td>
</tr>
</tbody>
</table>

2) RX Driver Package

The FLASH SPI 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 FLASH SPI FIT module are as follows:

<table>
<thead>
<tr>
<th>RX Driver Package product name</th>
<th>RX Driver Package revision number</th>
<th>Document number</th>
<th>Revision number of the included FLASH SPI FIT module</th>
</tr>
</thead>
<tbody>
<tr>
<td>RX Family</td>
<td>Rev.1.36</td>
<td>R01AN6515xx0136</td>
<td>Rev.3.03</td>
</tr>
<tr>
<td>RX Driver Package Ver.1.36</td>
<td>Rev.1.34</td>
<td>R01AN6323xx0134</td>
<td>Rev.3.03</td>
</tr>
<tr>
<td>RX Family</td>
<td>Rev.1.33</td>
<td>R01AN6073xx0133</td>
<td>Rev.3.03</td>
</tr>
<tr>
<td>RX Driver Package Ver.1.33</td>
<td>Rev.1.32</td>
<td>R01AN6013xx0132</td>
<td>Rev.3.02</td>
</tr>
<tr>
<td>RX Family</td>
<td>Rev.1.31</td>
<td>R01AN5975xx0131</td>
<td>Rev.3.02</td>
</tr>
<tr>
<td>RX Driver Package Ver.1.31</td>
<td>Rev.1.30</td>
<td>R01AN5882xx0130</td>
<td>Rev.3.02</td>
</tr>
<tr>
<td>RX Family</td>
<td>Rev.1.29</td>
<td>R01AN5826xx0129</td>
<td>Rev.3.01</td>
</tr>
<tr>
<td>RX Driver Package Ver.1.29</td>
<td>Rev.1.27</td>
<td>R01AN5600xx0127</td>
<td>Rev.3.01</td>
</tr>
<tr>
<td>RX Family</td>
<td>Rev.1.26</td>
<td>R01AN5401xx0126</td>
<td>Rev.3.01</td>
</tr>
</tbody>
</table>
### 1.2 Applicable Devices
RX110, RX111, RX113, RX130, RX13T, and RX140 groups
RX230, RX231, RX23E-A, RX23T, RX23W, RX24T, and RX24U groups
RX64M, RX65N, RX660, RX66N, RX66T, and RX671 groups
RX71M, RX72M, RX72N, and RX72T groups

### 1.3 Details
Incorrect “if” condition were found in several functions in `r_flash_spi_drvif.c`. This has resulted in the “if” condition would never be fulfilled, and thus the ‘if’ block of code would never be entered:

```c
//r_flash_spi_drvif.c
...
if (MEMDRV_SUCCESS == MEMDRV_ERR_HARD) //this is incorrect!!
{
    Return FLASH_SPI_ERR_HARD; //this will never be entered!!
}
... // Note:
// MEMDRV_SUCCESS = 0
// MEMDRV_ERR_HARD = -2
```

Table below lists the functions which contain the aforementioned bug, and the corresponding affected APIs:

<table>
<thead>
<tr>
<th>Function which contains the bug</th>
<th>Affected API</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>r_flash_spi_drvif_tx</code></td>
<td>R_FLASH_SPI_Set_Write_Protect, R_FLASH_SPI_Write_Configuration, R_FLASH_SPI_Quad_Enable, R_FLASH_SPI_Quad_Disable, R_FLASH_SPI_Write_Data_Page, R_FLASH_SPI_Erase, R_FLASH_SPI_Polling, R_FLASH_SPI_Read_Status, R_FLASH_SPI_Read_Security, R_FLASH_SPI_Read_ID, R_FLASH_SPI_Set_4byte_Address_Mode,</td>
</tr>
</tbody>
</table>

---

# Table 1.3 Function which has the bug and affected API

<table>
<thead>
<tr>
<th>Function which contains the bug</th>
<th>Affected API</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>r_flash_spi_drvif_tx</code></td>
<td>R_FLASH_SPI_Set_Write_Protect, R_FLASH_SPI_Write_Configuration, R_FLASH_SPI_Quad_Enable, R_FLASH_SPI_Quad_Disable, R_FLASH_SPI_Write_Data_Page, R_FLASH_SPI_Erase, R_FLASH_SPI_Polling, R_FLASH_SPI_Read_Status, R_FLASH_SPI_Read_Security, R_FLASH_SPI_Read_ID, R_FLASH_SPI_Set_4byte_Address_Mode,</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>r_flash_spi_drvif_tx_add</td>
<td>R_FLASH_SPI_Read_Data, R_FLASH_SPI_Read_Configuration</td>
</tr>
<tr>
<td>r_flash_spi_drvif_tx_data</td>
<td>R_FLASH_SPI_Write_Data_Page</td>
</tr>
<tr>
<td>r_flash_spi_drvif_rx</td>
<td>R_FLASH_SPI_Read_Configuration, R_FLASH_SPI_Set_Write_Protect, R_FLASH_SPI_Quad_Enable, R_FLASH_SPI_Quad_Disable, R_FLASH_SPI_Write_Configuration, R_FLASH_SPI_Read_Security, R_FLASH_SPI_Polling, R_FLASH_SPI_Read_ID, R_FLASH_SPI_Read_Status, R_FLASH_SPI_Write_Data_Page</td>
</tr>
<tr>
<td>r_flash_spi_drvif_rx_add</td>
<td>N/A</td>
</tr>
<tr>
<td>r_flash_spi_drvif_rx_data</td>
<td>R_FLASH_SPI_Read_Data</td>
</tr>
</tbody>
</table>

### 1.4 Conditions

- **In r_flash_spi_drvif_tx(),** if \( \text{R}_{MEMDRV}_{Tx}() \) returns \( \text{ret
drv} = \text{MEMDRV}_{ERR\_HARD} \), **FLASH\_SPI\_ERR\_OTHER will be returned to the caller, which is incorrect**

```c
// r_flash_spi_drvif_tx()
...
 ret_drv = R_MEMDRV_Tx(devno,&memdrv_info);

if (MEMDRV_SUCCESS == MEMDRV_ERR_HARD)
    { return FLASH_SPI_ERR_HARD; }
else if (MEMDRV_SUCCESS > ret_drv)
    { return FLASH_SPI_ERR_OTHER; }
return FLASH_SPI_SUCCESS;
```

- **In r_flash_spi_drvif_tx_add(),** if \( \text{R}_{MEMDRV}_{TxData}() \) returns \( \text{ret
drv} = \text{MEMDRV}_{ERR\_HARD} \), **FLASH\_SPI\_ERR\_OTHER will be returned to the caller, which is incorrect**

```c
// r_flash_spi_drvif_tx_add()
...
 ret_drv = R_MEMDRV_TxData(devno,&memdrv_info);

if (MEMDRV_SUCCESS == MEMDRV_ERR_HARD)
    { return FLASH_SPI_ERR_HARD; }
else if (MEMDRV_SUCCESS > ret_drv)
    { return FLASH_SPI_ERR_OTHER; }
return FLASH_SPI_SUCCESS;
```

- **In r_flash_spi_drvif_tx_data(),** if \( \text{R}_{MEMDRV}_{TxData}() \) returns \( \text{ret
drv} = \text{MEMDRV}_{ERR\_HARD} \), **FLASH\_SPI\_ERR\_OTHER will be returned to the caller, which is incorrect**

```c
// r_flash_spi_drvif_tx_data()
...
 ret_drv = R_MEMDRV_TxData(devno,&memdrv_info);

if (MEMDRV_SUCCESS == MEMDRV_ERR_HARD)
    { return FLASH_SPI_ERR_HARD; }
else if (MEMDRV_SUCCESS > ret_drv)
    { return FLASH_SPI_ERR_OTHER; }
return FLASH_SPI_SUCCESS;
```
// r_flash_spi_drvif_tx_data()
   ret_drv = R_MEMDRV_TxData(devno,&memdrv_info);

   if (MEMDRV_SUCCESS == MEMDRV_ERR_HARD)
   {
      return FLASH_SPI_ERR_HARD;
   }
   else if (MEMDRV_SUCCESS > ret_drv)
   {
      return FLASH_SPI_ERR_OTHER;
   }
   return FLASH_SPI_SUCCESS;
- In `r_flash_spi_drvif_rx()`, if `R_MEMDRV_Rx()` returns `ret_drv = MEMDRV_ERR_HARD`, `FLASH_SPI_ERR_OTHER` will be returned to the caller, which is incorrect.

```c
// r_flash_spi_drvif_rx()
...
ret_drv = R_MEMDRV_Rx(devno,&memdrv_info);
if (MEMDRV_SUCCESS == MEMDRV_ERR_HARD)
{
    return FLASH_SPI_ERR_HARD;
}
else if (MEMDRV_SUCCESS > ret_drv)
{
    return FLASH_SPI_ERR_OTHER;
}
return FLASH_SPI_SUCCESS;
```

- In `r_flash_spi_drvif_rx_add()`, if `R_MEMDRV_RxData()` returns `ret_drv = MEMDRV_ERR_HARD`, `FLASH_SPI_ERR_OTHER` will be returned to the caller, which is incorrect.

```c
// r_flash_spi_drvif_rx_add()
...
ret_drv = R_MEMDRV_RxData(devno,&memdrv_info);
if (MEMDRV_SUCCESS == MEMDRV_ERR_HARD)
{
    return FLASH_SPI_ERR_HARD;
}
else if (MEMDRV_SUCCESS > ret_drv)
{
    return FLASH_SPI_ERR_OTHER;
}
return FLASH_SPI_SUCCESS;
```

- In `r_flash_spi_drvif_rx_data()`, if `R_MEMDRV_RxData()` returns `ret_drv = MEMDRV_ERR_HARD`, `FLASH_SPI_ERR_OTHER` will be returned to the caller, which is incorrect.

```c
// r_flash_spi_drvif_rx_data()
...
ret_drv = R_MEMDRV_RxData(devno,&memdrv_info);
if (MEMDRV_SUCCESS == MEMDRV_ERR_HARD)
{
    return FLASH_SPI_ERR_HARD;
}
else if (MEMDRV_SUCCESS > ret_drv)
{
    return FLASH_SPI_ERR_OTHER;
}
return FLASH_SPI_SUCCESS;
```
1.5 Workaround

Temporary workaround: Change the ‘if’ condition from:

\[
\text{if (MEMDRV\_SUCCESS == MEMDRV\_ERR\_HARD)} \quad \text{to:}
\]

\[
\text{if (MEMDRV\_ERR\_HARD == ret\_drv)}
\]

User should upgrade to FLASH SPI FIT Rev.3.10 or later

1.6 Schedule for Fixing the Problem

This problem has been fixed in FLASH SPI FIT Rev.3.10
Revision History

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Page</th>
<th>Description</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Jun.16.23</td>
<td>-</td>
<td>First edition issued</td>
<td></td>
</tr>
</tbody>
</table>

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.

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

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.