

[Notes]

R20TS0518EJ0100

Rev.1.00

Dec. 16, 2019

RX Family

Simple I²C Module Firmware Integration Technology,

RX Driver Package

Outline

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

1. Invalid bit rate by the "R_SCI_IIC_Open" function

1. Invalid Bit Rate by the "R_SCI_IIC_Open" Function

1.1 Applicable Products

- (1) Simple I²C module Firmware Integration Technology (Simple I²C FIT module)

The applicable revision numbers and document numbers are as follows.

Table 1.1 Simple I²C FIT module applicable products

Simple I ² C FIT module revision number	Document number
Rev.2.43	R01AN1691EJ0243

- (2) RX Driver Package

The Simple I²C 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 Simple I²C FIT module are as follows.

Table 1.2 Products that include the Simple I²C FIT module

RX Driver Package product name	RX Driver Package revision number	Document number	Revision number of the included Simple I ² C FIT module
RX Family RX Driver Package, Ver.1.22	Rev.1.22	R01AN4873EJ0122	Rev.2.43

1.2 Applicable Devices

- RX72M group

1.3 Details

In the "sci_iic_set_frequency" function that is called within the "R_SCI_IIC_Open" function, an unexpected value may be set in the bit rate register (BRR) and the clock select bit (SMR.CKS) in the serial mode register. As a result, an invalid bit rate is set.

1.4 Conditions

This error occurs when all of the following conditions are met:

- One of SCI7 to SCI9 in the SCli module is used in simple I²C mode.
- The frequencies of the PCLKA and PCLKB peripheral module clocks are different.

1.5 Workaround

Add settings for SCI7 to SCI9 in the SCli module to the "r_sci_iic_rx72m.c" source file. The parts to be added are indicated in red in the corrected source file below.

- Before modification

Line number	Source code
988	static void sci_iic_set_frequency (sci_iic_info_t *
989	p_sci_iic_info)
-	{
1003	Omitted
1003	if ((SCI_IIC_NUM_CH10 == p_sci_iic_info->ch_no)
1004	(SCI_IIC_NUM_CH11 == p_sci_iic_info->ch_no))
1004	{
1005	brr_n_tmp = brr_n;
1006	brr_value = (uint32_t) ((double) ((double) ((double)
1007	BSP_PCLKA_HZ / (brr_n_tmp * (prom->bitrate)))) - 0.1);
1008	}
1008	else
1009	{
1010	brr_n_tmp = brr_n;
1011	brr_value = (uint32_t) ((double) ((double) ((double)
1012	BSP_PCLKB_HZ / (brr_n_tmp * (prom->bitrate)))) - 0.1);
1012	}
-	Omitted
1046	if ((SCI_IIC_NUM_CH10 == p_sci_iic_info->ch_no)
1047	(SCI_IIC_NUM_CH11 == p_sci_iic_info->ch_no))
1048	{
1048	brr_n_tmp = brr_n;
1049	brr_value = (uint32_t) ((double) (((double)
1050	BSP_PCLKA_HZ / (brr_n_tmp * (prom->bitrate)))) - 0.1);
1051	}
1051	else
1052	{
1053	brr_n_tmp = brr_n;
1054	brr_value = (uint32_t) ((double) (((double)
1055	BSP_PCLKB_HZ / (brr_n_tmp * (prom->bitrate)))) - 0.1);
1055	}
-	Omitted
1066	pregs->SMR.BYTE = cks_value_tmp; /* Sets SMR */
1067	pregs->BRR = brr_value; /* Sets BRR */
1068	} /* End of function sci_iic_set_frequency() */

- After modification

Line number	Source code
988	static void sci_iic_set_frequency (sci_iic_info_t *
989	p_sci_iic_info)
-	{
-	Omitted
1003	if ((SCI_IIC_NUM_CH7 == p_sci_iic_info->ch_no) ¥
1004	(SCI_IIC_NUM_CH8 == p_sci_iic_info->ch_no) ¥
1005	(SCI_IIC_NUM_CH9 == p_sci_iic_info->ch_no) ¥
1006	(SCI_IIC_NUM_CH10 == p_sci_iic_info->ch_no) ¥
1007	(SCI_IIC_NUM_CH11 == p_sci_iic_info->ch_no))
1008	{
1009	brr_n_tmp = brr_n;
1010	brr_value = (uint32_t) ((double) ((double) ((double)
	BSP_PCLKA_HZ / (brr_n_tmp * (prom->bitrate)))) - 0.1);
1011	}
1012	else
1013	{
1014	brr_n_tmp = brr_n;
1015	brr_value = (uint32_t) ((double) ((double) ((double)
	BSP_PCLKB_HZ / (brr_n_tmp * (prom->bitrate)))) - 0.1);
1016	}
-	Omitted
1050	if ((SCI_IIC_NUM_CH7 == p_sci_iic_info->ch_no) ¥
1051	(SCI_IIC_NUM_CH8 == p_sci_iic_info->ch_no) ¥
1052	(SCI_IIC_NUM_CH9 == p_sci_iic_info->ch_no) ¥
1053	(SCI_IIC_NUM_CH10 == p_sci_iic_info->ch_no) ¥
1054	(SCI_IIC_NUM_CH11 == p_sci_iic_info->ch_no))
1055	{
1056	brr_n_tmp = brr_n;
1057	brr_value = (uint32_t) ((double) (((double)
	BSP_PCLKA_HZ / (brr_n_tmp * (prom->bitrate)))) - 0.1);
1058	}
1059	else
1060	{
1061	brr_n_tmp = brr_n;
1062	brr_value = (uint32_t) ((double) (((double)
	BSP_PCLKB_HZ / (brr_n_tmp * (prom->bitrate)))) - 0.1);
1063	}
-	Omitted
1074	pregs->SMR.BYTE = cks_value_tmp; /* Sets SMR */
1075	pregs->BRR = brr_value; /* Sets BRR */
1076	} /* End of function sci_iic_set_frequency() */

1.6 Schedule for Fixing the Problem

This problem will be fixed in the next version Rev.2.45 ^(Note). (Scheduled to be released in 2020.)

Note: Rev.2.44 will not be released.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Dec.16.19	-	First edition issued

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.

The past news contents have been based on information at the time of publication. Now changed or invalid information may be included.

URLs in 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/

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.