[Notes]
e² studio Smart Configurator Plug-in, Smart Configurator for RX

R20TS0434ES0100
Rev.1.00
Jun. 01, 2019

Outline

When using the e² studio Smart Configurator Plug-in and Smart Configurator for RX, note the following points.

1. When using self-diagnosis function of 12-bit A/D converter in Single Scan Mode
2. When using Serial Peripheral Interface clock synchronous mode in slave transmit
3. When using I²C Bus Interface with Fast-mode Plus enabled

1. When Using Self-diagnosis Function of 12-bit A/D Converter in Single Scan Mode

1.1 Applicable Products

- e² studio V6.2.0 (Smart Configurator Plug-in V1.3.0) or later
- Smart Configurator for RX V1.3.0 or later

1.2 Applicable Devices

- RX family:
  RX230 and RX231 groups

1.3 Details

When using self-diagnosis function of 12-bit A/D converter in Single Scan Mode, sampling time is not generated. This problem applies to both Rotation mode and Fixed mode of self-diagnosis function.

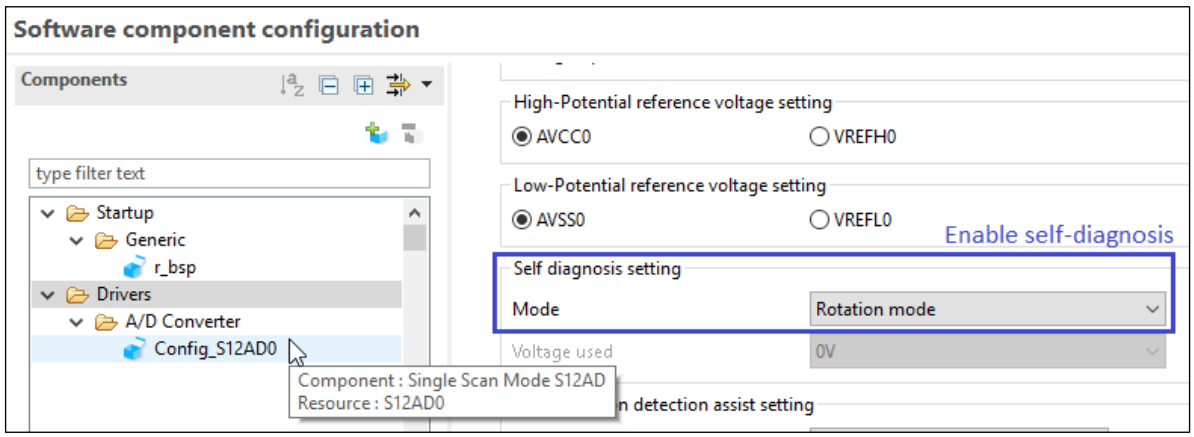

Figure 1. Example of enabling self-diagnosis function by Rotation mode

1.4 Workaround

Set sampling time by initializing A/D Sampling State Register 0 (ADSSTR0) in the function below.

Source file : < Single Scan Mode S12AD configuration name >_user.c
Function : R_< Single Scan Mode S12AD configuration name >_Create_UserInit

For setting A/D Sampling State Register 0 sampling time, refer to the following.

https://www.renesas.com/search/keyword-search.html#genre=document&q=r01uh0496

RX230 Group, RX231 Group User's Manual
43.3.6 Analog Input Sampling Time and Scan Conversion Time
50.5 A/D Conversion Characteristics

In the example below, ADSSTR0 register is initialized 0x0BU (equivalent to 11 states). The added code is written in red.

Before modification:

```
/***********************************************************************
* Function Name: R_Config_S12AD0_Create_UserInit
* Description  : This function adds user code after initializing
*                the S12AD0 channel
* Arguments    : None
* Return Value : None
***********************************************************************/
void R_Config_S12AD0_Create_UserInit(void)
{
    /* Start user code for user init. Do not edit comment generated here */
    /* End user code. Do not edit comment generated here */
}
```

After modification:

```
/***********************************************************************
* Function Name: R_Config_S12AD0_Create_UserInit
* Description  : This function adds user code after initializing
*                the S12AD0 channel
* Arguments    : None
* Return Value : None
***********************************************************************/
void R_Config_S12AD0_Create_UserInit(void)
{
    /* Start user code for user init. Do not edit comment generated here */
    /* Set sampling time */
    S12AD.ADSSTR0 = 0x0BU;
    /* End user code. Do not edit comment generated here */
}
```

1.5 Schedule for Fixing the Problem

This problem will be fixed in the next version. (Scheduled to be released in July 2019.)

2. When using Serial Peripheral Interface clock synchronous mode in slave transmit

2.1 Applicable Products

- e² studio V7.4.0 (Smart Configurator Plug-in V2.1.0) or later
- Smart Configurator for RX V2.1.0 or later

2.2 Applicable Devices

- RX family:
  RX66T and RX72T groups

2.3 Details

When using SPI Clock Synchronous Mode component in Slave transmit only, group AL0 interrupt SPEI0 is not generated even if error interrupt (SPEI0) has been enabled in GUI.

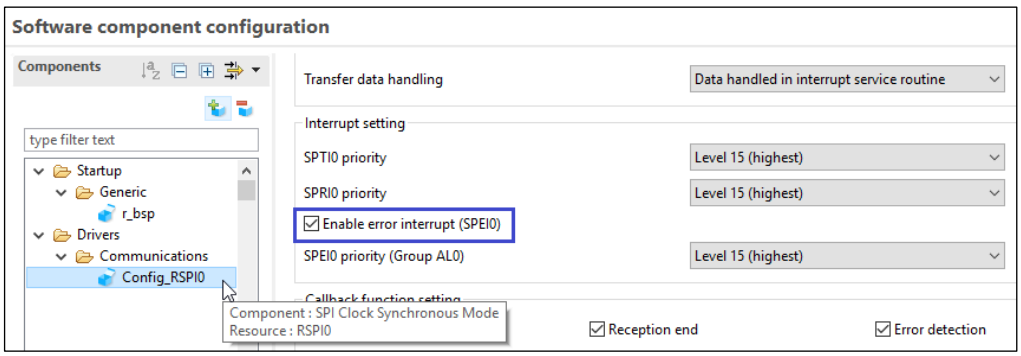

Figure 2. Error interrupt is enabled in SPI Clock Synchronous Mode

Error location:

Source file : r_cg_hardware_setup.c
Function : R_Systeminit

```
/***********************************************************************
* Function Name: R_Systeminit
* Description  : This function initializes every configuration
* Arguments    : None
* Return Value : None
***********************************************************************/
void R_Systeminit(void)
{
    …
    /* Register undefined interrupt */
    R_BSP_InterruptWrite(BSP_INT_SRC_UNDEFINED_INTERRUPT,
                         (bsp_int_cb_t)r_undefined_exception);
                                          ⬅ Interrupt setting code is not generated.
    /* Disable writing to MPC pin function control registers */
    MPC.PWPR.BIT.PFSWE = 0U;
    MPC.PWPR.BIT.B0WI = 1U;
    …
}
```

2.4 Workaround

Add code in the function below to register group AL0 interrupt SPEI0.

Source file : < SPI Clock Synchronous Mode configuration name >_user.c
Function : R_< SPI Clock Synchronous Mode configuration name >_Create_UserInit

Before modification:

```
/***********************************************************************
* Function Name: R_Config_RSPI0_Create_UserInit
* Description  : This function adds user code after initializing RSPI0
* Arguments    : None
* Return Value : None
***********************************************************************/
void R_Config_RSPI0_Create_UserInit(void)
{
    /* Start user code for user init. Do not edit comment generated here */
    /* End user code. Do not edit comment generated here */
}
```

After modification:

```
/***********************************************************************
* Function Name: R_Config_RSPI0_Create_UserInit
* Description  : This function adds user code after initializing RSPI0
* Arguments    : None
* Return Value : None
***********************************************************************/
void R_Config_RSPI0_Create_UserInit(void)
{
    /* Start user code for user init. Do not edit comment generated here */
    /* Register group AL0 interrupt SPEI0 (RSPI0) */
    R_BSP_InterruptWrite(BSP_INT_SRC_AL0_RSPI0_SPEI0,
                         (bsp_int_cb_t)r_Config_RSPI0_error_interrupt);
    /* End user code. Do not edit comment generated here */
}
```

2.5 Schedule for Fixing the Problem

This problem will be fixed in the next version. (Scheduled to be released in July 2019.)

3. When using I²C Bus Interface with Fast-mode Plus enabled

3.1 Applicable Products

- e² studio V6.0.0 (Smart Configurator Plug-in V1.2.0) or later
- Smart Configurator for RX V1.2.0 or later

3.2 Applicable Devices

- RX family:
  RX64M, RX651, RX65N and RX71M groups

3.3 Details

When using RIIC0 in the I²C Master Mode or I²C Slave Mode, initialization of Fast-Mode Plus Enable bit (ICFER.FMPE) is incorrect when baud rate is changed from Fast-mode Plus (400kbps to 1 Mbps) transmission rate to slower transmission rate, for example Fast-mode transmission rates (100kbps to 400 kbps) or Standard-mode (up to 100 kbps).

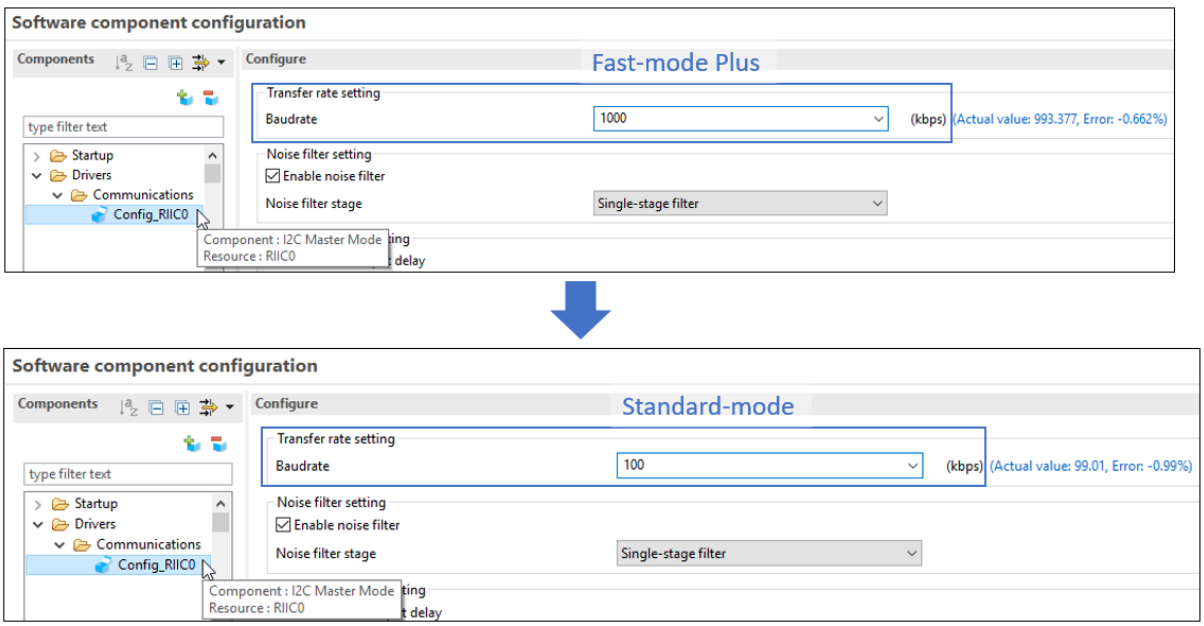

Figure 3. Change transmission rate from Fast-mode Plus to Standard-mode

Error location (red portion in the example below):

Source file : < I2C Master Mode configuration name >.c
Function : R_< I2C Master Mode configuration name >_Create

```
/***********************************************************************
* Function Name: R_Config_RIIC0_Create
* Description  : This function initializes the RIIC0 channel
* Arguments    : None
* Return Value : None
***********************************************************************/
void R_Config_RIIC0_Create (void)
{
    …
    /* Set ICFER */
    RIIC0.ICFER.BYTE = _00_IIC_TIMEOUT_FUNCTION_DISABLE |
                       _02_IIC_MASTER_ARBITRATION_ENABLE |
                       _00_IIC_NACK_ARBITRATION_DISABLE |
                       _10_IIC_NACK_SUSPENSION_ENABLE |
                       _20_IIC_NOISE_FILTER_USED |
                       _80_IIC_FASTPLUS_MODE_ENABLE;
    …
}
```

3.4 Workaround

Add code in the function below to modify setting of ICFER.FMPE bit.

The modification example is shown below. The added code is written in red.

Source file : < I2C Master Mode configuration name >_user.c
Function : R_< I2C Master Mode configuration name >_Create_UserInit

Before modification:

```
/***********************************************************************
* Function Name: R_Config_RIIC0_Create_UserInit
* Description  : This function adds user code after initializing RIIC0
channel
* Arguments    : None
* Return Value : None
***********************************************************************/
void R_Config_RIIC0_Create (void)
{
    /* Start user code for user init. Do not edit comment generated here */
    /* End user code. Do not edit comment generated here */
}
```

After modification:

```
/***********************************************************************
* Function Name: R_Config_RIIC0_Create_UserInit
* Description  : This function adds user code after initializing RIIC0
channel
* Arguments    : None
* Return Value : None
***********************************************************************/
void R_Config_RIIC0_Create (void)
{
    /* Start user code for user init. Do not edit comment generated here */
    /* Set to No FM+ */
    RIIC0.ICFER.BIT.FMPE = 0U;
    /* End user code. Do not edit comment generated here */
}
```

3.5 Schedule for Fixing the Problem

This problem will be fixed in the next version. (Scheduled to be released in July 2019.)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jun.01.19	-	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. 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 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.

© 2019. Renesas Electronics Corporation. All rights reserved.

TS Colophon 4.0