[Notes]
e^2 studio Smart Configurator Plug-in,
Smart Configurator for RX

R20TS0555ES0100
Rev.1.00
Mar. 16, 2020

Outline

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

1. When using the TGIC7 and TGID7 interrupts in Normal Mode Timer or PWM Mode Timer
2. When creating a project with RX24T 64-pin FK packages
3. When using compare level of AN109 in Single Scan Mode S12AD

1. When Using the TGIC7 and TGID7 Interrupts in Normal Mode Timer or PWM Mode Timer

1.1 Applicable Products

- e^2 studio V7.3.0 (Smart Configurator Plug-in V2.0.0) or later
- Smart Configurator for RX V2.0.0 or later

1.2 Applicable Devices

- RX family:
  RX24T, RX24U groups

1.3 Details

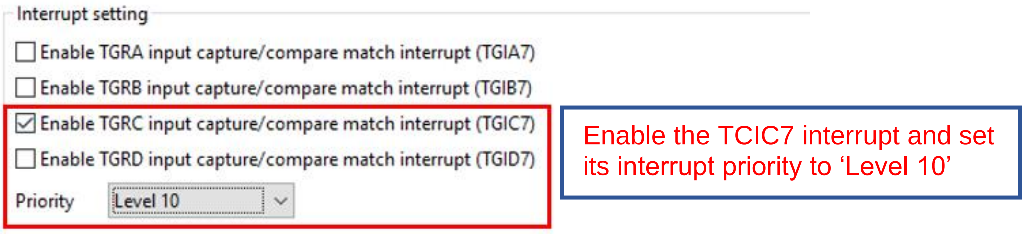
When using the TGIC7 or TGID7 interrupt in Normal Mode Timer or PWM Mode Timer, the interrupt priority setting code is generated out incorrectly under the following two conditions:

(1) When TGIA7 and TGIB7 are not used, interrupt priority setting code is incorrect with wrong macro parameter. See example 1.

(2) When TGIA7 or TGIB7 is used, interrupt priority setting code is not generated out, See example 2.

- Example 1: When TGIA7 and TGIB7 are not used, and the priority level of the "TGIC7 interrupt" is set to "Level 10"

  Code that is different from the GUI settings (Figure 1.1) is generated (Figure 1.2).



Interrupt setting

□ Enable TGRA input capture/compare match interrupt (TGIA7)

□ Enable TGRB input capture/compare match interrupt (TGIB7)

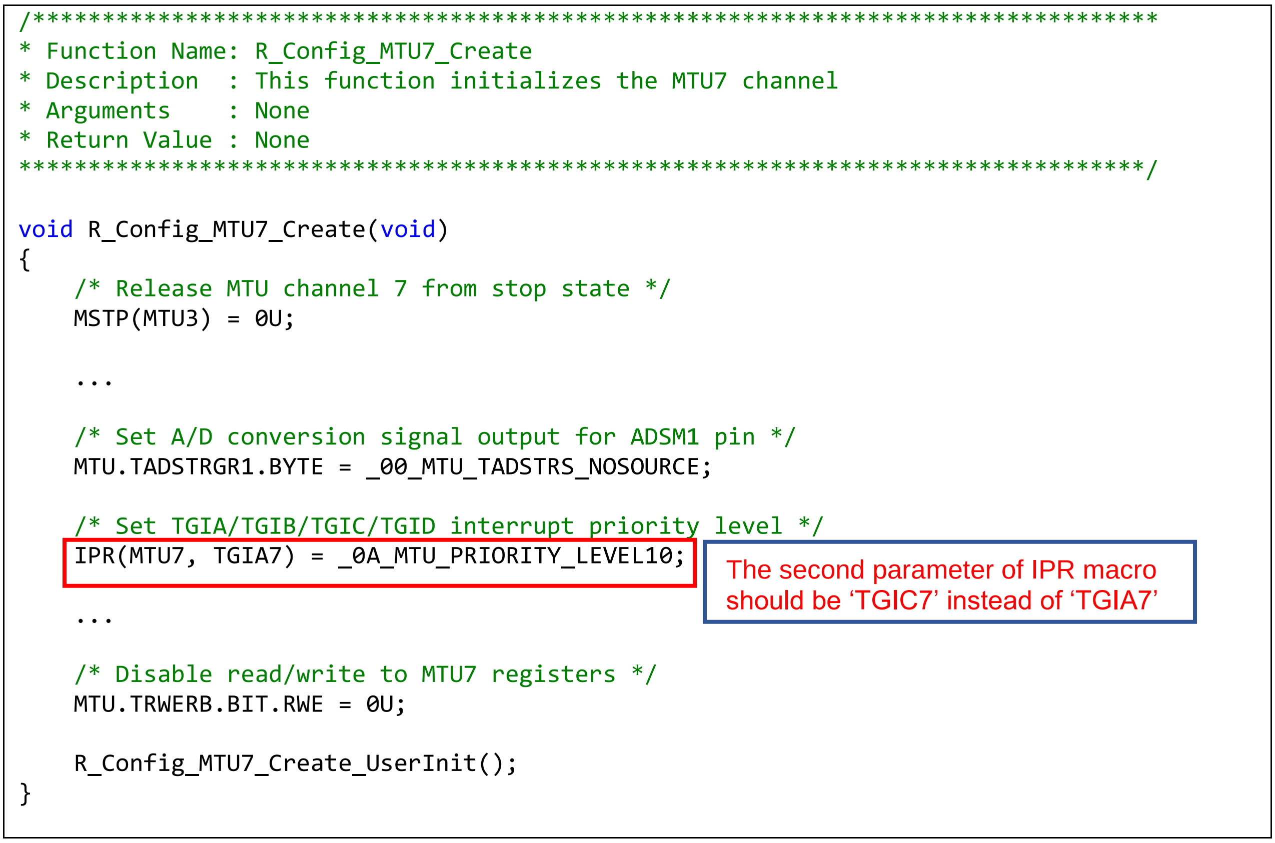
☑ Enable TGRC input capture/compare match interrupt (TGIC7)

□ Enable TGRD input capture/compare match interrupt (TGID7)

Priority: Level 10

Enable the TCIC7 interrupt and set its interrupt priority to 'Level 10'

Figure 1.1 GUI configuration for TGIC7 interrupt in Normal Mode Timer

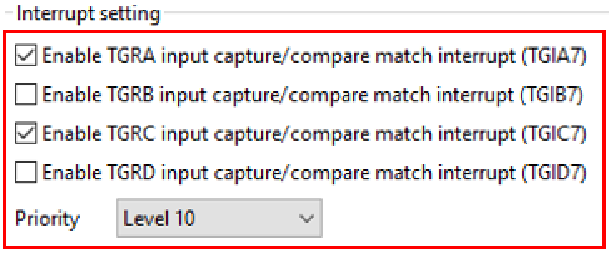


```
/*******************************************************************************
* Function Name: R_Config_MTU7_Create
* Description  : This function initializes the MTU7 channel
* Arguments    : None
* Return Value : None
*******************************************************************************/

void R_Config_MTU7_Create(void)
{
    /* Release MTU channel 7 from stop state */
    MSTP(MTU3) = 0U;

    ...

    /* Set A/D conversion signal output for ADSM1 pin */
    MTU.TADSTRGR1.BYTE = _00_MTU_TADSTRS_NOSOURCE;

    /* Set TGIA/TGIB/TGIC/TGID interrupt priority level */
    IPR(MTU7, TGIA7) = _0A_MTU_PRIORITY_LEVEL10;

    ...

    /* Disable read/write to MTU7 registers */
    MTU.TRWERB.BIT.RWE = 0U;

    R_Config_MTU7_Create_UserInit();
}
```

The second parameter of IPR macro should be ‘TGIC7’ instead of ‘TGIA7’

Figure 1.2 Wrong code generated for the TGIC7 interrupt priority settings in Normal Mode Timer

- Example 2: When TGIA7 is used, and the priority level of the “TGIC7 interrupt” is set to “Level 10” Code for interrupt priority setting of TGIC7 is not generated out (Figure 1.4).



Interrupt setting

☑ Enable TGRA input capture/compare match interrupt (TGIA7)

☐ Enable TGRB input capture/compare match interrupt (TGIB7)

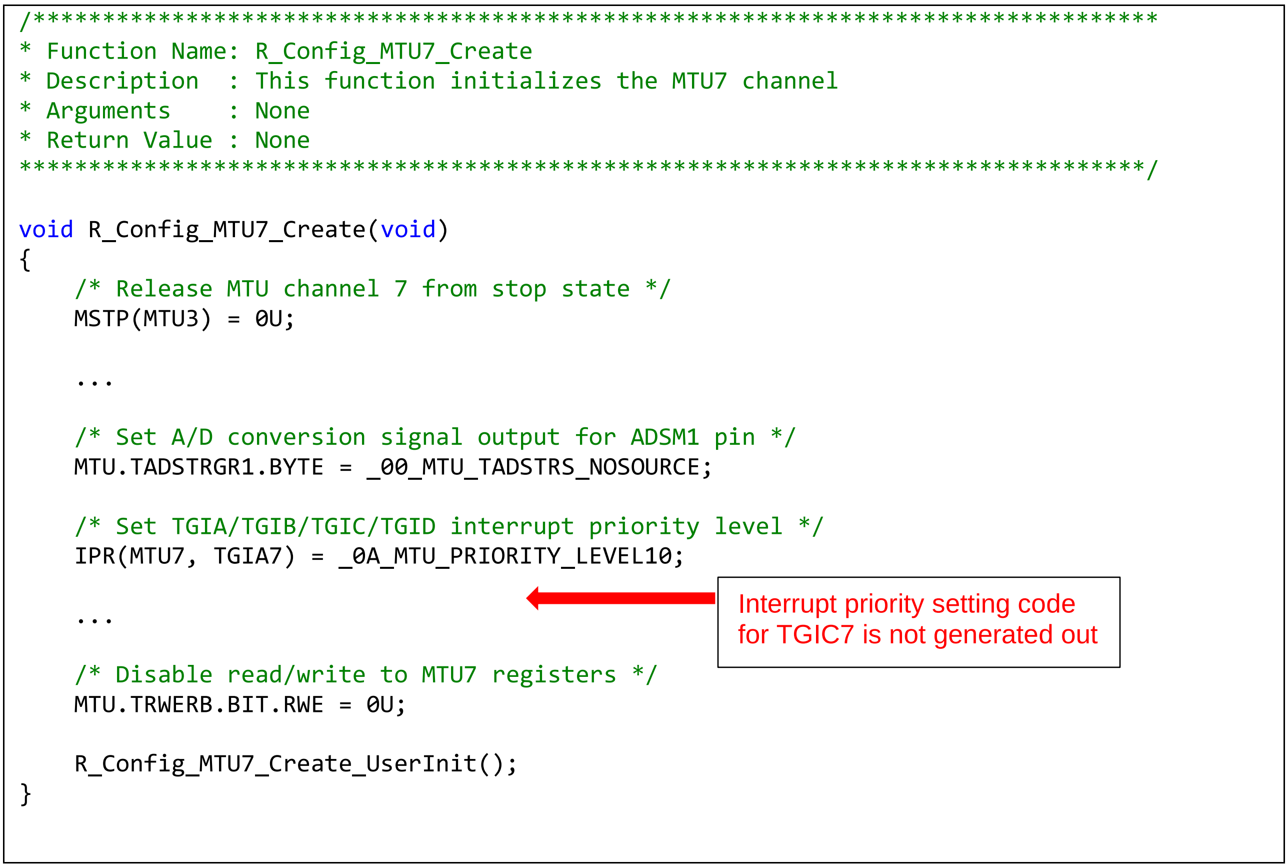
☑ Enable TGRC input capture/compare match interrupt (TGIC7)

☐ Enable TGRD input capture/compare match interrupt (TGID7)

Priority: Level 10

Enable the TGIA7 interrupt and TCIC7 interrupt and set its interrupt priority to ‘Level 10’

Figure 1.3 GUI configuration for TGIA7 and TGIC7 interrupt in Normal Mode Timer



```
/*******************************************************************************
* Function Name: R_Config_MTU7_Create
* Description  : This function initializes the MTU7 channel
* Arguments    : None
* Return Value : None
*******************************************************************************/

void R_Config_MTU7_Create(void)
{
    /* Release MTU channel 7 from stop state */
    MSTP(MTU3) = 0U;

    ...

    /* Set A/D conversion signal output for ADSM1 pin */
    MTU.TADSTRGR1.BYTE = _00_MTU_TADSTRS_NOSOURCE;

    /* Set TGIA/TGIB/TGIC/TGID interrupt priority level */
    IPR(MTU7, TGIA7) = _0A_MTU_PRIORITY_LEVEL10;

    ...

    /* Disable read/write to MTU7 registers */
    MTU.TRWERB.BIT.RWE = 0U;

    R_Config_MTU7_Create_UserInit();
}
```

Interrupt priority setting code
for TGIC7 is not generated out

Figure 1.4 No generated code for the TGIC7 interrupt priority settings in Normal Mode Timer

1.4 Workaround

In the generated file, do either of the following.

Workaround varies depending on whether the “TGIA7” or “TGIB7” interrupt is being used.

(1) When the TGIA7 and TGIB7 interrupts are not used.

Manually change the second parameter of IPR macro from “TGIA7” to “TGIC7”. See example 1.

(2) When the TGIA7 or TGIB7 interrupt is used.

Add “IPR(MTU7, TGIC7)”, IPR macro with “TGIC7” specified as the second parameter. See example 2.

- Source file: “<*Configuration-name*>.c”
- Function: "void R_<*Configuration-name*>_Create(void)"

The <*Configuration-name*> varies depending on the selected component of Normal Mode Timer or PWM Mode Timer.

Note: When code is generated again, generated code returns to the state before modification. Therefore, modify the source file each time you generate code.

Below are examples of modification when the <*Configuration-name*> is Config_MTU7 and the priority level of the “TGIC7 interrupt” is set to “Level 10”.

■ Example 1: When the TGIA7 and TGIB7 interrupts are not used

```
/*********************************************************************************
* Function Name: R_Config_MTU7_Create
* Description  : This function initializes the MTU7 channel
* Arguments    : None
* Return Value : None
*********************************************************************************/

void R_Config_MTU7_Create(void)
{
    /* Release MTU channel 7 from stop state */
    MSTP(MTU3) = 0U;
    ...

    /* Set A/D conversion signal output for ADSM1 pin */
    MTU.TADSTRGR1.BYTE = _00_MTU_TADSTRS_NOSOURCE;

    /* Set TGIA/TGIB/TGIC/TGID interrupt priority level */
    IPR(MTU7, TGIC7) = _0A_MTU_PRIORITY_LEVEL10;

    ...
    /* Disable read/write to MTU7 registers */
    MTU.TRWERB.BIT.RWE = 0U;

    R_Config_MTU7_Create_UserInit();
}
```

Manually change the second parameter of IPR macro from ‘TGIA7’ to TGIC7’

- Example 2: When the TGIA7 or TGIB7 interrupt is used

```
/*********************************************************************************
* Function Name: R_Config_MTU7_Create
* Description  : This function initializes the MTU7 channel
* Arguments    : None
* Return Value : None
*********************************************************************************/

void R_Config_MTU7_Create(void)
{
    /* Release MTU channel 7 from stop state */
    MSTP(MTU3) = 0U;
    ...

    /* Set A/D conversion signal output for ADSM1 pin */
    MTU.TADSTRGR1.BYTE = _00_MTU_TADSTRS_NOSOURCE;

    /* Set TGIA/TGIB/TGIC/TGID interrupt priority level */
    IPR(MTU7, TGIA7) = _0A_MTU_PRIORITY_LEVEL10;

    IPR(MTU7, TGIC7) = _0A_MTU_PRIORITY_LEVEL10;

    ...
    /* Disable read/write to MTU7 registers */
    MTU.TRWERB.BIT.RWE = 0U;

    R_Config_MTU7_Create_UserInit();
}
```

Manually add the TGIC7 interrupt priority setting codes

1.5 Schedule for Fixing the Problem

This problem will be fixed in the following versions. (Scheduled to be released in April 2020.)

- e^2 studio V7.8.0
- Smart Configurator for RX V2.5.0

2. When Creating a Project with RX24T 64-pin FK Packages

2.1 Applicable Products

- e^2 studio V7.6.0 (Smart Configurator Plug-in V2.2.1) or later
- Smart Configurator for RX V2.2.1 or later

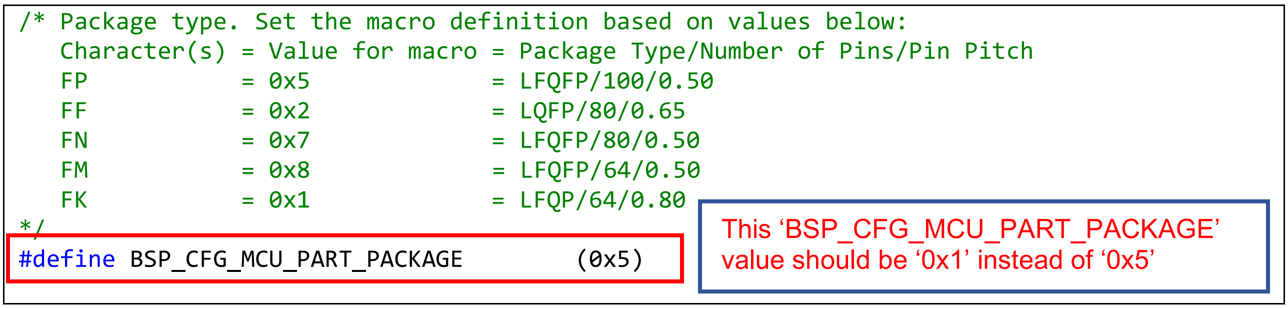
2.2 Applicable Devices

- RX family:
  RX24T group

2.3 Details

When you create a project with RX24T 64-pin FK packages (R5F524TAAxFK and R5F524T8AxFK) and generate out codes, the ‘BSP_CFG_MCU_PART_PACKAGE’ macro value in the r_bsp_config.h file is set wrongly. (It should be ‘0x1’ instead of ‘0x5’.)

Error location:



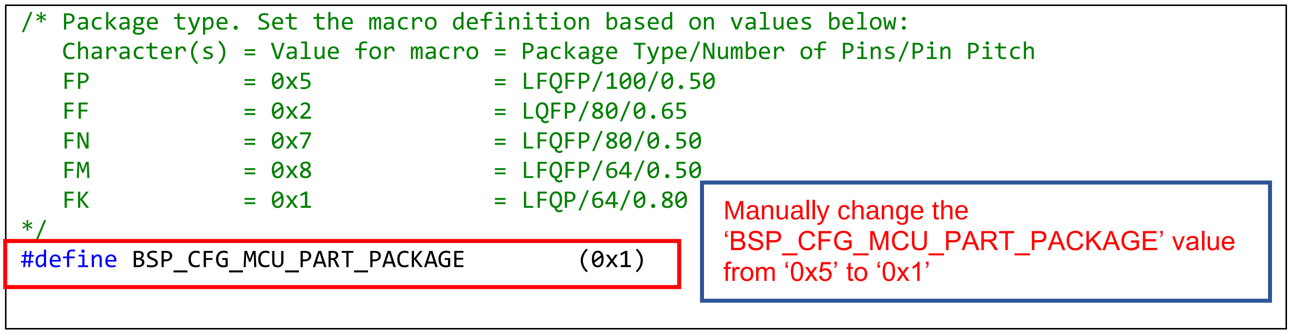
```
/* Package type. Set the macro definition based on values below:
   Character(s) = Value for macro = Package Type/Number of Pins/Pin Pitch
   FP           = 0x5             = LFQFP/100/0.50
   FF           = 0x2             = LQFP/80/0.65
   FN           = 0x7             = LFQFP/80/0.50
   FM           = 0x8             = LFQFP/64/0.50
   FK           = 0x1             = LFQP/64/0.80
*/
#define BSP_CFG_MCU_PART_PACKAGE        (0x5)
```

This ‘BSP_CFG_MCU_PART_PACKAGE’ value should be ‘0x1’ instead of ‘0x5’

2.4 Workaround

In the r_bsp_config.h file generated out under project path ‘src/smc_gen/r_config’, manually change the ‘BSP_CFG_MCU_PART_PACKAGE’ macro value from ‘0x5’ to ‘0x1’ as shown below.

Modification example:



```
/* Package type. Set the macro definition based on values below:
   Character(s) = Value for macro = Package Type/Number of Pins/Pin Pitch
   FP           = 0x5             = LFQFP/100/0.50
   FF           = 0x2             = LQFP/80/0.65
   FN           = 0x7             = LFQFP/80/0.50
   FM           = 0x8             = LFQFP/64/0.50
   FK           = 0x1             = LFQP/64/0.80
*/
#define BSP_CFG_MCU_PART_PACKAGE        (0x1)
```

Manually change the ‘BSP_CFG_MCU_PART_PACKAGE’ value from ‘0x5’ to ‘0x1’

2.5 Schedule for Fixing the Problem

This problem will be fixed in the following versions. (Scheduled to be released in April 2020.)

- e^2 studio V7.8.0
- Smart Configurator for RX V2.5.0

3. When Using Compare Level of AN109 in Single Scan Mode S12AD

3.1 Applicable Products

- e^2 studio V6.2.0 (Smart Configurator Plug-in V1.3.0) or later
- Smart Configurator for RX V1.3.0 or later

3.2 Applicable Devices

- RX family:
  RX71M group

3.3 Details

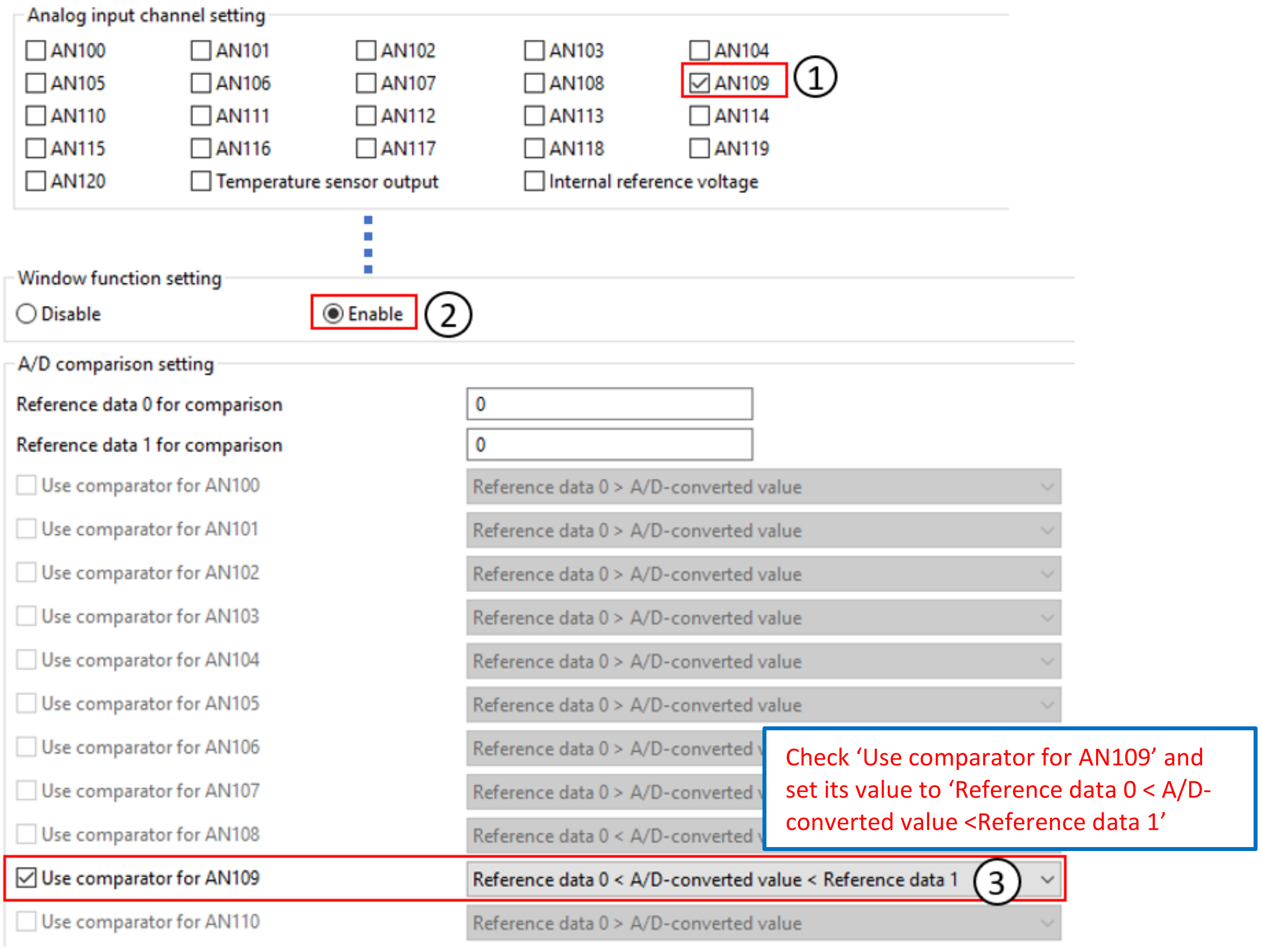
When using comparator for AN109 in Single Scan Mode S12AD, and setting its value to one of the following, the corresponding register setting code cannot be generated.

(1) Reference data 0 < A/D-converted value

See Figure 3.1 for GUI configuration.

(2) Reference data 0 < A/D-converted value < Reference data 1

See Figure 3.2 for GUI configuration.



Analog input channel setting

□ AN100	□ AN101	□ AN102	□ AN103	□ AN104
□ AN105	□ AN106	□ AN107	□ AN108	☑ AN109 ①
□ AN110	□ AN111	□ AN112	□ AN113	□ AN114
□ AN115	□ AN116	□ AN117	□ AN118	□ AN119
□ AN120	□ Temperature sensor output		□ Internal reference voltage	

Window function setting

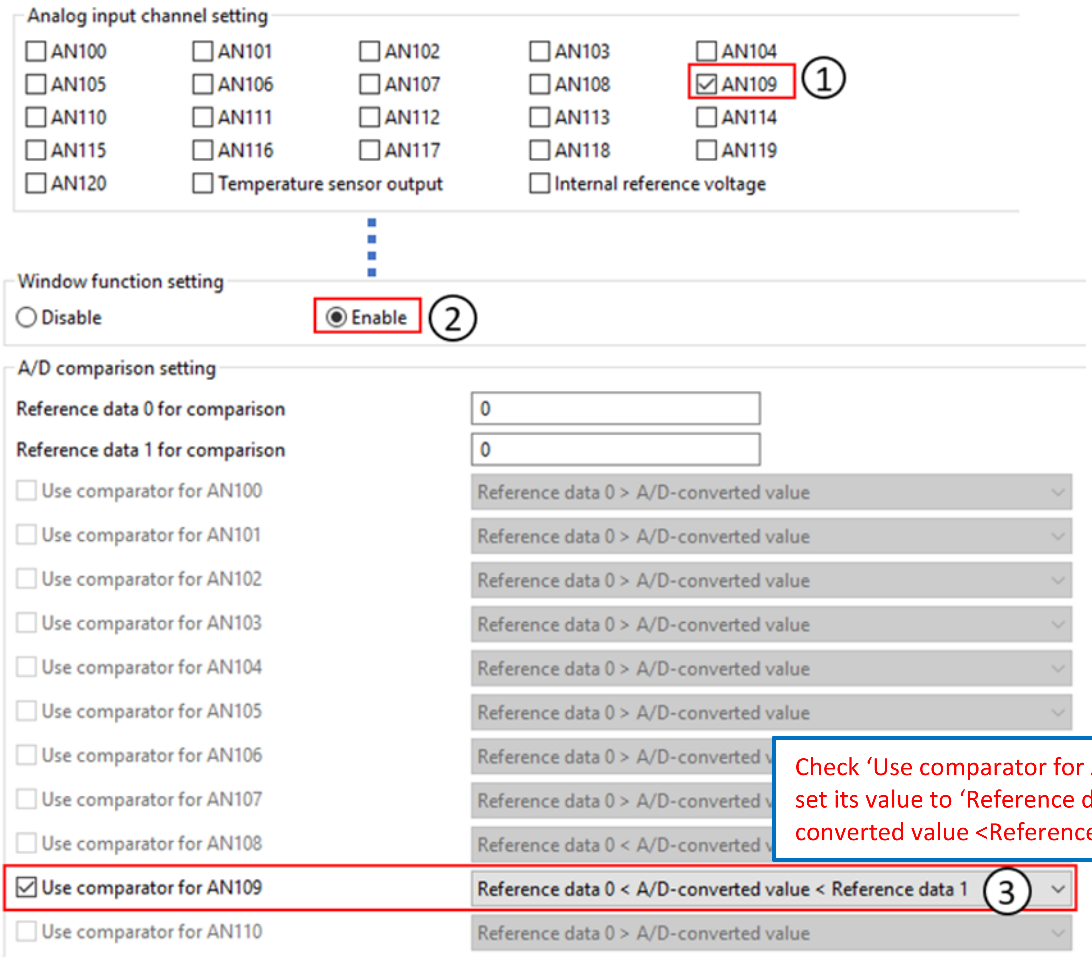
○ Disable ● Enable ②

A/D comparison setting

Reference data 0 for comparison	0
Reference data 1 for comparison	0
□ Use comparator for AN100	Reference data 0 > A/D-converted value
□ Use comparator for AN101	Reference data 0 > A/D-converted value
□ Use comparator for AN102	Reference data 0 > A/D-converted value
□ Use comparator for AN103	Reference data 0 > A/D-converted value
□ Use comparator for AN104	Reference data 0 > A/D-converted value
□ Use comparator for AN105	Reference data 0 > A/D-converted value
□ Use comparator for AN106	Reference data 0 > A/D-converted v
□ Use comparator for AN107	Reference data 0 > A/D-converted v
□ Use comparator for AN108	Reference data 0 < A/D-converted v
☑ Use comparator for AN109	Reference data 0 < A/D-converted value < Reference data 1 ③
□ Use comparator for AN110	Reference data 0 > A/D-converted value

Check ‘Use comparator for AN109’ and set its value to ‘Reference data 0 < A/D-converted value <Reference data 1’

Figure 3.1 GUI configurations for setting compare level to ‘Reference data 0 < A/D-converted value’



Analog input channel setting

□ AN100 □ AN101 □ AN102 □ AN103 □ AN104
□ AN105 □ AN106 □ AN107 □ AN108 ☑ AN109
□ AN110 □ AN111 □ AN112 □ AN113 □ AN114
□ AN115 □ AN116 □ AN117 □ AN118 □ AN119
□ AN120 □ Temperature sensor output □ Internal reference voltage

Window function setting

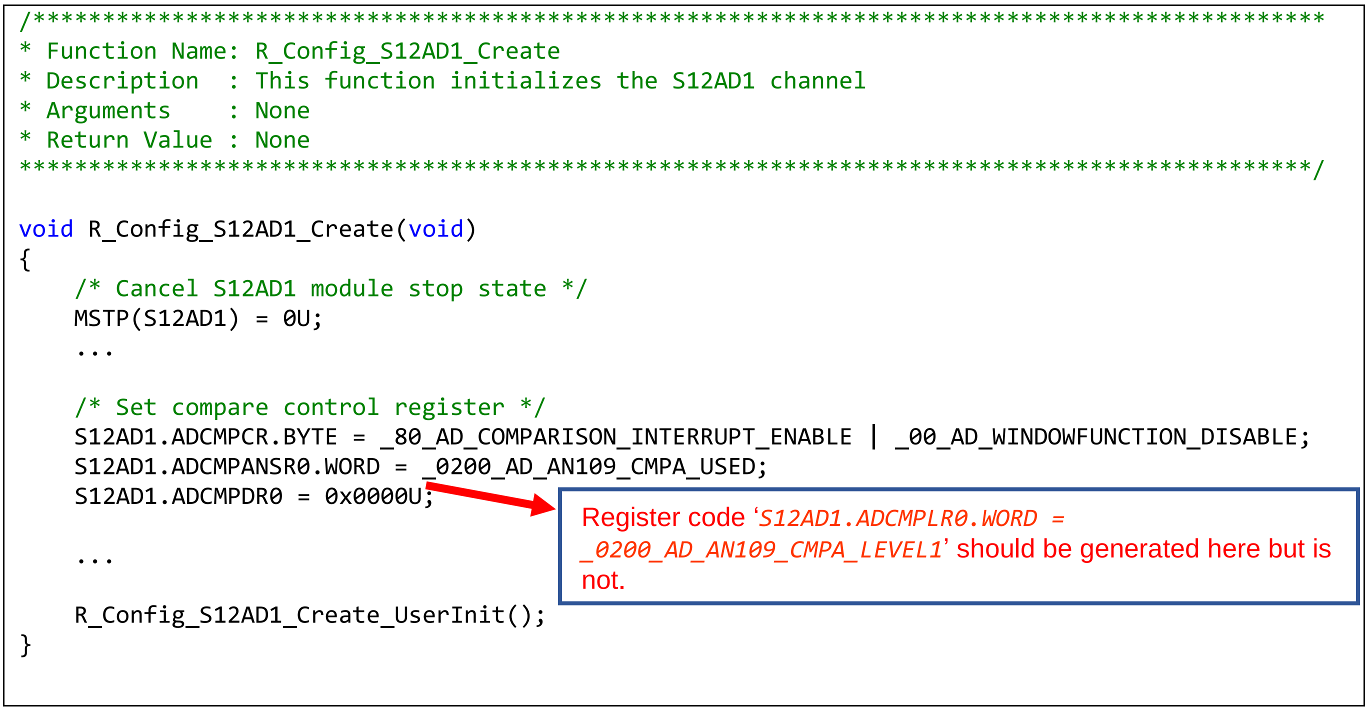
○ Disable ● Enable

A/D comparison setting

Reference data 0 for comparison	0
Reference data 1 for comparison	0
□ Use comparator for AN100	Reference data 0 > A/D-converted value
□ Use comparator for AN101	Reference data 0 > A/D-converted value
□ Use comparator for AN102	Reference data 0 > A/D-converted value
□ Use comparator for AN103	Reference data 0 > A/D-converted value
□ Use comparator for AN104	Reference data 0 > A/D-converted value
□ Use comparator for AN105	Reference data 0 > A/D-converted value
□ Use comparator for AN106	Reference data 0 > A/D-converted v
□ Use comparator for AN107	Reference data 0 > A/D-converted v
□ Use comparator for AN108	Reference data 0 < A/D-converted v
☑ Use comparator for AN109	Reference data 0 < A/D-converted value < Reference data 1
□ Use comparator for AN110	Reference data 0 > A/D-converted value

Check ‘Use comparator for AN109’ and set its value to ‘Reference data 0 < A/D-converted value <Reference data 1’

Figure 3.2 GUI configurations for setting compare level to ‘Reference data 0 < A/D-converted value < Reference data 1’



```
/****************************************************************************************
* Function Name: R_Config_S12AD1_Create
* Description  : This function initializes the S12AD1 channel
* Arguments    : None
* Return Value : None
****************************************************************************************/

void R_Config_S12AD1_Create(void)
{
    /* Cancel S12AD1 module stop state */
    MSTP(S12AD1) = 0U;
    ...

    /* Set compare control register */
    S12AD1.ADCMPCR.BYTE = _80_AD_COMPARISON_INTERRUPT_ENABLE | _00_AD_WINDOWFUNCTION_DISABLE;
    S12AD1.ADCMPANSR0.WORD = _0200_AD_AN109_CMPA_USED;
    S12AD1.ADCMPDR0 = 0x0000U;

    ...

    R_Config_S12AD1_Create_UserInit();
}
```

Register code ‘*S12AD1.ADCMPLR0.WORD = _0200_AD_AN109_CMPA_LEVEL1*’ should be generated here but is not.

Figure 3.3 Location of expected generated code when compare level is ‘Reference data 0 < A/D-converted value’ or ‘Reference data 0 < A/D-converted value < Reference data 1’

3.4 Workaround

Manually add register code ‘*S12AD1.ADCMPLR0.WORD = _0200_AD_AN109_CMPA_LEVEL1*’ into the generated file.

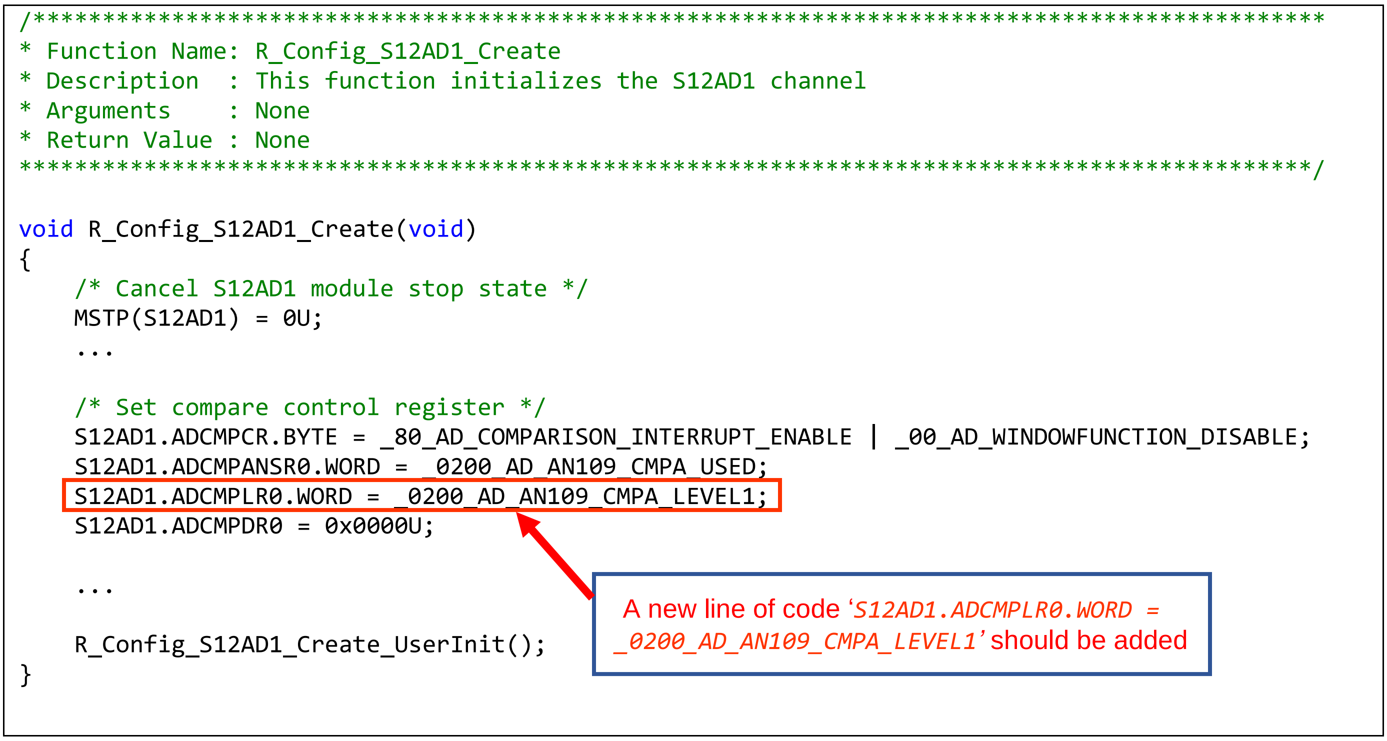
- Source file: “<*Configuration-name*>.c”
- Function: "void R_<*Configuration-name*>_Create(void)"

The <*Configuration-name*> varies depending on the selected component of Single Scan Mode S12AD.

> Note: When code is generated again, generated code returns to the state before modification. Therefore, modify the source file each time you generate code.

Below is the example of modification when the <*Configuration-name*> is Config_S12AD1 (initial value).

Modification example:



```
/***********************************************************************************************
* Function Name: R_Config_S12AD1_Create
* Description  : This function initializes the S12AD1 channel
* Arguments    : None
* Return Value : None
***********************************************************************************************/

void R_Config_S12AD1_Create(void)
{
    /* Cancel S12AD1 module stop state */
    MSTP(S12AD1) = 0U;
    ...

    /* Set compare control register */
    S12AD1.ADCMPCR.BYTE = _80_AD_COMPARISON_INTERRUPT_ENABLE | _00_AD_WINDOWFUNCTION_DISABLE;
    S12AD1.ADCMPANSR0.WORD = _0200_AD_AN109_CMPA_USED;
    S12AD1.ADCMPLR0.WORD = _0200_AD_AN109_CMPA_LEVEL1;
    S12AD1.ADCMPDR0 = 0x0000U;

    ...

    R_Config_S12AD1_Create_UserInit();
}
```

A new line of code ‘*S12AD1.ADCMPLR0.WORD = _0200_AD_AN109_CMPA_LEVEL1*’ should be added

3.5 Schedule for Fixing the Problem

This problem will be fixed in the following versions. (Scheduled to be released in April 2020.)

- e^2 studio V7.8.0
- Smart Configurator for RX V2.5.0

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Mar.16.20	-	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.

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

Trademarks

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/

TS Colophon 4.1