

RENESAS TOOL NEWS on July 16, 2015: 150716/tn2

Notes on Using the CS+ Code Generator for RL78 (CS+ for CA and CX), the CS+ Code Generator for RL78 (CS+ for CC), the e2 studio (Code Generator Plug-in), the AP4 Coding Assistance Tool for the RL78, and the Applilet3 Coding Assistance Tool for the RL78

When using the CS+ Code Generator for RL78 (CS+ for CA and CX), the CS+ Code Generator for RL78 (CS+ for CC), the e2 studio (Code Generator Plug-in), the AP4 coding assistance Tool for RL78, and the Applilet3 coding assistance tool for RL78, take note of the problems on the following points that are described in this note.

1. Clock Generation Circuit (PLL Circuit Operation)

Applicable MCUs: RL78/D1A, RL78/F13, RL78/F14, RL78/G1C, and RL78/L1C groups

2. Setting P40 of Port 4

Applicable MCUs: RL78/D1A, RL78/F12, RL78/F13, RL78/F14, RL78/G10, RL78/G12, RL78/G13, RL78/G14, RL78/G1A, RL78/G1C, RL78/G1E, RL78/G1F, RL78/G1G, RL78/I1A, RL78/I1D, RL78/L1C, RL78/L12, and RL78/L13 groups

3. Code Generated for UART0 and UARTF

Applicable MCUs: RL78/F12 group

1. Clock Generation Circuit (PLL Circuit Operation)

1.1 Products Concerned

- V2.01.00 and later versions of the CS+ Code Generator for RL78 (CS+ for CA and CX)
- V2.07.00 and later versions of the CS+ Code Generator for RL78 (CS+ for CC)
- V2.2.0.13 and later versions of the e2 studio (V1.0.1 and later version of the Code Generator Plug-in)
- V1.06.00 and later versions of the AP4 coding assistance tool for RL78*
- V1.00.00 and later versions of the Applilet3 coding assistance tool

for RL78

*: This note also applies to the following products.

- V1.00.00 and later versions of the Application Leading Tool which is a coding assistance tool for RL78

Note: The Application Leading Tool for RL78 is listed separately because its name has been changed to AP4 for RL78 from V1.06.00 (the latter are the newer versions of the former).

1.2 Applicable MCUs

RL78/D1A, RL78/F13, RL78/F14, RL78/G1C, and RL78/L1C groups

1.3 Description

Generated code includes an error when the PLL circuit is operating as the clock generation circuit. A wait is required immediately after setting the PLL control register (PLLCTL).

1.4 Workaround

Modify the code output for void R_CGC_Create(void) in the way shown below, according to the device you are using. The function is in the r_cg_cg.c file. This modification is required every time code is generated.

The output code differs with the MCU.

Before modification (for the RL78/D1A, RL78/F13, and RL78/F14 groups):

```
-----  
void R_CGC_Create(void)  
{  
.....  
    /* Set fPLL */  
    PLLCTL = _40_CGC_LOCKUP_WAIT_8 | _00_CGC_PLL_BELOW_32MHZ  
            | _00_CGC_PLL_DIVISION_2 | _00_CGC_PLL_MULTIPLY_X12;  
    PLLON = 1U;          /* <- faulty sequence */  
  
    /* Change the waiting time according to the system */  
    for (w_count = 0U; w_count <= CGC_PLLWAITTIME; w_count++)  
    {  
        NOP();  
    }  
  
    while ((PLLSTS & 0x80) == 0U)  
    {  
        ;  
    }  
}
```

```
}
```

```
.....
```

```
}
```

After modification (for the RL78/D1A, RL78/F13, and RL78/F14 groups):

void R_CGC_Create(void)

```
{
```

```
.....
```

```
    /* Set fPLL */
```

```
    PLLCTL = _40_CGC_LOCKUP_WAIT_8 | _00_CGC_PLL_BELOW_32MHZ  
            | _00_CGC_PLL_DIVISION_2 | _00_CGC_PLL_MULTIPLY_X12;
```

```
    /* Change the waiting time according to the system */
```

```
    for (w_count = 0U; w_count <= CGC_PLLWAITTIME; w_count++)
```

```
    {
```

```
        NOP();
```

```
    }
```

```
        PLLON = 1U;    /* <- correct sequence          */
```

```
                    /* That is, modify the order of */
```

```
                    /* processing.                */
```

```
    while ((PLLSTS & 0x80) == 0U)
```

```
    {
```

```
        ;
```

```
    }
```

```
.....
```

```
}
```

Before modification (for the RL78/G1C and RL78/L1C groups):

void R_CGC_Create(void).

```
{
```

```
.....
```

```
    /* Set fPLL */
```

```
    DSCCTL = _04_CGC_PLL_DIVISION_2 | _00_CGC_PLL_MULTI_12;
```

```
    MCKC = _00_CGC_FIH_DIVISION_2;
```

```
    /* Change the waiting time according to the system */
```

```
    for (w_count = 0U; w_count <= CGC_FPLLWAITTIME; w_count++)
```

```
    {
```

```
        NOP();
```

```
}
```

```
DSCCTL |= _01_CGC_PLL_OPERATION_ON;
```

```
/* <- faulty sequence */
```

```
.....  
}
```

After modification (for the RL78/G1C and RL78/L1C groups):

```
void R_CGC_Create(void)
```

```
{
```

```
.....
```

```
/* Set fPLL */
```

```
DSCCTL = _04_CGC_PLL_DIVISION_2 | _00_CGC_PLL_MULTI_12;
```

```
MCKC = _00_CGC_FIH_DIVISION_2;
```

```
/* Change the waiting time according to the system */
```

```
for (w_count = 0U; w_count <= CGC_FPLLWAITTIME; w_count++)
```

```
{
```

```
    NOP();
```

```
}
```

```
DSCCTL |= _01_CGC_PLL_OPERATION_ON;
```

```
/* Change the waiting time according to the system */
```

```
/* <- correct sequence */
```

```
for (w_count = 0U; w_count <= CGC_FPLLWAITTIME; w_count++)
```

```
{
```

```
    NOP();
```

```
}
```

```
.....  
}
```

1.5 Schedule for Fixing the Problem

This problem will be fixed in the next version.

2. Setting P40 of Port 4

2.1 Products Concerned

- V2.01.00 and later versions of the CS+ Code Generator for RL78 (CS+ for CA and CX)
- V2.07.00 and later versions of the CS+ Code Generator for RL78

(CS+ for CC)

- V2.1.0.21 and later versions of the e2 studio
(V1.0.0 and later version of Code Generator Plug-in)
- V1.06.00 and later versions of the AP4 coding assistance tool
for RL78*
- V1.00.00 and later versions of the Applilet3 coding assistance tool
for RL78

*: This note also applies to the following products.

- V1.00.00 and later versions of the Application Leading Tool
which is a coding assistance tool for RL78

Note: The Application Leading Tool for RL78 is listed separately
because its name has been changed to AP4 for RL78 from V1.06.00
(the latter are the newer versions of the former).

2.2 Applicable MCUs

RL78/D1A, RL78/F12, RL78/F13, RL78/F14, RL78/G10, RL78/G12, RL78/G13,
RL78/G14, RL78/G1A, RL78/G1C, RL78/G1E, RL78/G1F, RL78/G1G, RL78/I1A,
RL78/I1D, RL78/L1C, RL78/L12, and RL78/L13 groups

2.3 Description

Generated code has an error when P40 is set such that the on-chip
pull-up resistor for P40 is not connected even though this is included
in the settings of the on-chip pull-up resistors for port 4.

The code to set the pull-up resistor option register (PU4) of P40 is not
generated.

2.4 Workaround

Modify the code output for void R_PORT_Create(void) in the function
r_cg_port.c to include setting of the pull-up resistor option register
(PU4). This modification is required every time code is generated.

Before modification:

```
-----  
void R_PORT_Create(void)  
{  
    /* <- faulty sequence */  
    R_PORT_Create_UserInit();  
}
```

After the modification:

```
-----  
void R_PORT_Create(void)
```

```

{
    PU4 = _00_PUn0_PULLUP_OFF; /* <- correct sequence */
        /* That is, modify the */
        /* order of processing. */

    R_PORT_Create_UserInit();
}
-----

```

2.5 Schedule for Fixing the Problem

This problem will be fixed in the next version.

3. Code Generated for UART0 and UARTF

3.1 Products Concerned

- V2.01.00 and later versions of the CS+ Code Generator for RL78 (CS+ for CA and CX)
- V2.07.00 and later versions of the CS+ Code Generator for RL78 (CS+ for CC)
- V2.2.0.13 and later versions of the e2 studio (V1.0.1 and later versions of the Code Generator Plug-in)
- V1.02.00 and later versions of the Applilet3 coding assistance tool for RL78

3.2 Applicable MCUs

RL78/F12 group

3.3 Description

- (a) Generated code has an error when unit 0 of the serial array unit is used as UART0 and its configuration is set to transmission or transmission and reception. Unnecessary code is output to the function void R_UART0_Creat(void) which is in r_cg_serial.c.
- (b) Generated code has an error in the setting of the LTXD0 pin when the asynchronous serial interface LIN-UART (UARTF) is set for transmission or transmission and reception. Incorrect code is output to the function void R_UARTF0_Create(void) which is in r_cg_serial.c.

3.4 Workarounds

(a) Workaround for 3.3(a)

Modify the code output for void R_UART0_Create(void) in the way shown below. The function is in the r_cg_serial.c file. This modification is required every time code is generated.

Before modification:

```
-----  
void R_UART0_Create(void)  
{  
.....  
    /* Set RxD0 pin */  
    PMX1 |= 0x01U;          /* <- faulty code */  
    PM1 |=0x02U;  
.....  
}
```

After modification:

```
-----  
void R_UART0_Create(void)  
{  
.....  
    /* Set RxD0 pin */  
                                /* <- deleted code */  
    PM1 |=0x02U;  
.....  
}
```

(b) Workaround for 3.3(b)

Modify the code output for void R_UARTF0_Create(void) in the way shown below. The function is in the r_cg_serial.c file. This modification is required every time code is generated.

Before modification:

```
-----  
void R_UARTF0_Create(void)  
{  
.....  
    /* Set LTXD0 pin */  
    PM5 |= 0x02U;  
    PMX2 = 0xFEU;          /* <- faulty code */  
.....  
}
```

After the modification:

```
-----  
void R_UARTF0_Create(void)  
{
```

```
.....  
/* Set LTXD0 pin */  
PM5 |= 0x02U;  
PMX2 &= 0xFEU;          /* <- correct code */  
.....  
}  
-----
```

3.5 Schedule for Fixing the Problem

This problem will be fixed in the next version.

[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.