RENESAS Tool News

RENESAS TOOL NEWS on November 1, 2014: 141101/tn2

Note on Using V2.05.00 of Code Generator for RL78 with CS+ (CS+ for CA and CX)

Please note the following points for caution when using V2.05.00 of the code generator for RL78 with CS+ (CS+ for CA and CX).

1. Point for Caution on Settings for CPU Stack Pointer Monitoring (Applicable When Using Products of the RL78/F13 Group)
2. Point for Caution on Writing to the Serial Flag Clear Trigger Register (SIR) When Using 3-wire Serial (CSI) Transfer (Applicable When Using Products of the RL78/F12 Group)

---

1. Applicable Product
   V2.05.00 of code generator for RL78 when used with CS+ (CS+ for CA and CX).

2. Items
 2.1 Point for Caution on Settings for CPU Stack Pointer Monitor Function

   MCUs to which this notice applies: RL78/F13 group

   The order of statements in the procedure for setting the registers for CPU stack pointer monitor function* is erroneous as shown below.
   (1) SPM control register (SPMCTRL) <-- This register should be set after step (3).
   (2) SP overflow address setting register (SPOFR)
   (3) SP underflow address setting register (SPUFR)
   Note: CPU stack pointer monitor function is a security function of the MCU.

   Workarounds:
   Modify the R_Systeminit() function in r_systeminit.c to the correct order of settings.

   Example of modification when setting the SPOFR register to 0xFFFE and

the SPUFR register to 0x0000.

Before Modification

```
SPMCTRL = 0x80U;
SPOFR = 0xFFFEU;
SPUFR = 0x0000U;
```

After Modification

```
SPOFR = 0xFFFEU;
SPUFR = 0x0000U;
SPMCTRL = 0x80U; <--  This sets the SPM enable bit (bit 7) of the
                      SPMCTR register to 1.
```

2.2 Point for Caution on Writing to the Serial Flag Clear Trigger Register (SIR) When Using 3-wire Serial (CSI) Transfer

MCUs to which this notice applies: RL78/F12 group

There was an erroneous statement in the code for writing to the serial flag clear trigger registers listed below. Bit 2 is set to 1, but the correct setting for bit 2 is 0.

SIR00, SIR02, SIR10, and SIRS0 registers

Workarounds:

Modify the R_CSIxx_Create() function in r_cs_serial.c so that bit 2 is correctly set to 0 in the SIR00, SIR02, SIR1, and SIRS0 registers.

Correction example: When channel 0 of SAU1 is used by CSI20

Before Modification

```
void R_CSI20_Create(void)
{
 . . . . . .
    SIR10 = _0004_SAU_SIRMN_FECTMN | _0002_SAU_SIRMN_PECTMN |
          _0001_SAU_SIRMN_OVCTMN;    /* clear error flag */
 . . . . . .
```

After Modification

Delete "| _0001_SAU_SIRMN_OVCTMN" as shown below.

```
SIR10 = _0004_SAU_SIRMN_FECTMN | _0002_SAU_SIRMN_PECTMN;
```

3. Schedule for Fixing the Problem

This problem will be fixed in a later revision of the product.

[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.