RENESAS Tool News

Corrections to the User's Manual of the C Compiler Package for the M16C MCU Series

We notify corrections to the following user's manuals. They are included with the C compiler package--M3T-NC30WA--V.5.40 Release 00 through V.5.42 Release 00 for the M16C MCU series.*

- Document No. REJ10J1218-0100
  M16C/60, 30, 20, 10, Tiny, R8C/Tiny Series,
  C Compiler Package V.5.40,
  C Compiler User's Manual
- Document No. REJ10J1389-0100
  M16C/60, 30, 20, 10, Tiny, R8C/Tiny Series,
  C Compiler Package V.5.42,
  C Compiler User's Manual

*Generic name of the M16C/60, /30, /20, /10, /Tiny, and R8C/Tiny MCU series.

---

1. Document No. REJ10J1218-0100
M16C/60, 30, 20, 10, Tiny, R8C/Tiny Series, C Compiler Package V.5.40,
C Compiler User's Manual

1.1 Corrections
(1) Item c, Section 3.3.3 on Page 65
(Notes on Handling General-purpose and Address Registers)

For the description
  When changing the contents of general purpose registers (R1, R2,
  and R3, except for R0) and the address registers (A0 and A1) in
  an assembler function, it is necessary to save them on the stack
  at the beginning procedure of the assembler function and recover
  them from the stack at the ending procedure of it. However, if an

assembler function is declared by using #pragma PARAMETER /C, the codes for saving and recovering the contents of registers are created at the calling side, so that it is unnecessary to save and recover them in this assembler function.

read as follows:
Even when any of the general-purpose registers (R0, R1, R2, and R3) or the address registers (A0 and A1) is changed in its contents within an assembler function or the one specified in the #pragma PARAMTER directive, it is unnecessary to save the contents of the register on the stack and restore them from it before and after processing the assembler function.

(2) Descriptions of the /C Switch in Extended Functions on pages 134, 135, 147, 151, 152, and 188

The #pragma PARAMETER, #pragma SPECIAL, and #pragma INTCALL directives are not provided with the function of switch /C. So we ask you to remove the descriptions of the /C switch in these extended functions on the above pages.
For the full corrections to those pages, see the Web page at
http://tool-support.renesas.com/eng/toolnews/070301/tn5.htm
This Web page will be opened from March 20 on.

2. The User's Manual of the C Compiler Package V.5.42 for the M16C/60, 30,
  20,10, Tiny, and R8C/Tiny Series, Document No. REJ10J1687-0100

2.1 Corrections
(1) Item c, Section 3.3.3 on Page 60
(Notes on Handling General-purpose and Address Registers)

For the description
When changing the contents of general purpose registers (R1, R2, and R3, except for R0) and the address registers (A0 and A1) in an assembler function, it is necessary to save them on the stack at the beginning procedure of the assembler function and recover them from the stack at the ending procedure of it. However, if an assembler function is declared by using #pragma PARAMETER/C, the codes for saving and recovering the contents of registers are created at the calling side, so that it is unnecessary to save and recover them in this assembler function.

read as follows:
Even when any of the general-purpose registers (R0, R1, R2, and R3) or the address registers (A0 and A1) is changed in its contents within an assembler function or the one specified in the #pragma PARAMTER directive, it is unnecessary to save the contents of the register on the stack and restore them from it before and after processing the assembler function.

(2) Descriptions of the /C Switch in Extended Functions on pages 133, 134, 147, 151, 152, and 188

The #pragma PARAMETER, #pragma SPECIAL, and #pragma INTCALL directives are not provided with the function of switch /C. So we ask you to remove the descriptions of the /C switch in these extended functions on the above pages.
For the full corrections to those pages, see the Web page at
  http://tool-support.renesas.com/eng/toolnews/070301/tn5.htm
This Web page will be opened from March 20 on.

3. Notice
If you save and restore the contents of any register using pushm and popm instructions as described in Item c in Section 3.3.3, no problems will arise though the size of code increases by four bytes for each assembler function.