RENESAS Tool News

RENESAS TOOL NEWS on March 16, 2007: 070316/tn4

A Note on Using the C Compiler Package --M3T-NC308WA-- for the M32C MCU Series

Please take note of the problem described below in using the C compiler package--M3T-NC308WA--for the M32C MCU series.*
*Generic name of the M32C/80, and M16C/80 series.

1. Product and Versions Concerned

The C compiler package for the M32C MCU series
V.5.40 Release 00 through V.5.41 Release 01

2. Problem

If a section greater than 64K bytes in size is initialized using a C-language startup file generated by the High-performance Embedded Workshop, an assemble error will arise.

3. Condition

This problem occurs if a section greater than 64K bytes in size is initialized using the macro function scopy_f() or sclear_f() in a C-language start-up file.

4. Workaround

We provide the modified startup file, in which have been fixed this Problem.

So download the problem-fixed zip file from the Web page at
http://tool-support.renesas.com/eng/toolnews/070316/tn4.htm
(Available on and after April 5 on).

If you are using V.5.41 Release 01,* decompress the downloaded zip file to obtain the initsct.c and initsct.h files; then copy them to the following two directories:**

- Hew¥System¥Pg¥Renesas¥M32C¥V5_41_1¥Generate¥c_start¥m32c9x
- Hew¥System¥Pg¥Renesas¥M32C¥V5_41_1¥Generate¥c_start¥other

*If you are using V.5.40 Release 00 or V.5.41 Release 00, read
V5_40_0 or V5_41_0 for V5_41_1 respectively in the above.
**These are the directories when you have installed the product
by default.

After these procedures, the above problems will be resolved if you use the startup file in C language generated when a new workspace is created

For the already-created workspaces, modify the existing initsct.c and initsct.h files as follows:

- In the sclear_f() macro

```
  #define  sclear_f(X,Y)  _asm("  .initsct  "X","Y"¥n"¥
           "  push.w  #(sizeof  "X")>>16¥n"¥
           "  push.w  #(sizeof  "X")&0ffffH¥n"¥
           "  pusha  (topof  "X")¥n"¥    ----- (1)
           "  .stk  8¥n"¥
           "  .glb  _bzero¥n"¥
           "  .call  _bzero,G¥n"¥
           "  jsr.a  _bzero¥n"¥         ----- (2)
           "  add.l  #8H,SP¥n"¥         ----- (3)
           "  .stk  -8");
```

(1) Add the topof operator here.
(2) Since the last of (2) is wrapped to (3), end it not with ");
but with ¥n"¥, which concatenates character strings.
(3) Add an 8-byte adjustment code of the stack and stack information.

- In the scopy_f() macro

```
 #define  scopy_f(X,Y)  _asm("  .initsct  "X","Y"¥n"¥
           "  .initsct  "X"I,rom"Y"¥n"¥
           "  push.w  #(sizeof  "X")  >>  16¥n"¥
           "  push.w  #(sizeof  "X")  &  0ffffH¥n"¥
           "  pusha  (topof  "X")¥n"¥    ----- (1)
           "  pusha  (topof  "X"I)¥n"¥   ----- (2)
           "  .stk  12¥n"¥
           "  .glb  _bcopy¥n"¥
           "  .call  _bcopy,G¥n"¥
```

```
        "  jsr.a  _bcopy¥n"¥          ----- (3)
        "  add.l  #0cH,SP¥n"¥          ----- (4)
        "  .stk  -12");               ----- (5)
```

(1) Add the topof operator here.
(2) Add the topof operator here.
(3) Since the last of (3) is wrapped to (4), end it not with ");
but with ¥n"¥, which concatenates character strings.
(4) Add a 12-byte adjustment code of the stack pointer.
(5) Add stack information by .stk.

5. Schedule of Fixing the Problem

We plan to fix this problem in the next release 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.