

RENESAS TOOL NEWS on July 1, 2005: RSO-M3T-NC30WA-050701D

A Note on Using the C-Compiler Package M3T-NC30WA V.5.30 Release 02

Please take note of the following problem in using the C-compiler package M3T-NC30WA V.5.30 Release 02 (this compiler package is used for the M16C/60, M16C/30, M16C/20, M16C/10, M16C/Tiny, and R8C/Tiny series of MCUs):

- On assigning a near-qualified variable of 64-bit data type to a far-qualified variable of the same type

1. Description

If a near-qualified variable of 64-bit data type* is assigned to a far-qualified variable of the same type, the values in the lower 3rd and 4th bytes of an 8-byte data piece cannot be done.

* Omission of the qualifier is interpreted as qualified to be near.

Example:

```
-----  
far long long l2;  
long long l1;
```

```
void func(void)  
{  
    l2 = l1;  
}
```

2. Workaround

When assigning a near-qualified variable of 64-bit data type to a far-qualified variable of the same type, transfer the former variable 2 bytes at a time using asm functions as follows:

- If the variables to reference are auto variables or arguments, use format "\$\$", and if they are external variables, use format "\$\$" or "\$@".

far long long l2;

```
void func(long long l1)
{
    asm(" ste.w $$[FB],$@",l1,l2);
    asm(" ste.w $$+2[FB],$@+2",l1,l2);
    asm(" ste.w $$+4[FB],$@+4",l1,l2);
    asm(" ste.w $$+6[FB],$@+6",l1,l2);
}
```

3. **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.

© 2010-2016 Renesas Electronics Corporation. All rights reserved.