

A Note on Using C Compiler Packages M3T-NC308WA and M3T-NC30WA

Please take note of the following problem in using the M3T-NC308WA and M3T-NC30WA C-compiler packages with an assembler and integrated development environment:

- On using the near or far qualifier

1. Products and Versions Concerned

C compiler packages for the M32C/80, and M16C/80 series of MCUs

M3T-NC308WA V.5.00 Release 1 through V.5.20 Release 1

C compiler packages for the M16C/60, M16C/30, M16C/Tiny, M16C/20, M16C/10, and R8C/Tiny series of MCUs

M3T-NC30WA V.5.00 Release 1 through V.5.30 Release 1

2. Description

If a near or far qualifier is placed in front of a function type System Error may arise.

2.1 Conditions

This problem occurs if the following conditions are all satisfied:

- (1) A function type A (for instance, "sample" in Example below) is declared to be typedef.
- (2) The function type A in (1) is also declared to be a pointer.
- (3) In front of another function type B pointed to by the pointer in (2) (for instance, "func" in Example below) is placed a near or far qualifier.

2.2 Example

```
-----  
typedef void    sample    (int, int); /* Condition (1) */
```

```
sample far * func;          /* Conditions (2) and (3) */
```

3. Workaround

This problem can be circumvented in either of the following ways:

- (1) Don't qualify the function type B as near or far.

```
-----  
typedef void sample (int, int);
```

```
sample * func;
```

- (2) Qualify the function type A as near or far at its typedef declaration.

```
-----  
typedef void far sample (int, int);
```

```
sample far * func;
```

4. Schedule of Fixing the Problem

We plan to fix this problem in our next release of the products.

[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.