A Note on Using C Compiler Package M3T-NC30WA
--On Inverting the Sign of a Signed Variable--

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

- On inverting the sign of a signed variable

1. **Versions Concerned**
   M3T-NC30WA V.5.00 Release 1 through V.5.30 Release 1

2. **Description**
   Inverting the sign of a signed variable may result in incorrect code being generated if the variable is less than zero.

2.1 **Conditions**
   This problem occurs if all the following conditions are satisfied:

   (1) Any of the optimizing options -O1, -O2, -O3, -O4, -O5, -OR, and -OS is selected at compilation.

   (2) A signed variable is compared with zero in the controlling expression of an "if" statement.

   (3) The type of the variable in (2) is any of these, signed char, signed int, and signed short.

   (4) If the variable is less than zero in the comparison in (2), in the true statement exists only an expression where the above-mentioned variable is assigned to another with its sign being inverted.

2.2 **Example**
3. **Workaround**

Place a dummy asm function immediately after the controlling expression.

```c
signed char s1,s2;
void func(void)
{
    if(s1 < 0)        /* Condition (2) */
        asm();        /* Dummy asm function placed */
    s2 = -s1; /* Condition (4) and (5) */
}
```

In the above example, -s1 is assigned to s2 even if the controlling expression is unsatisfied.

4. **Schedule of Fixing the Problem**

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