RENESAS Tool News

A Note on Using the C Compiler Packages for the M16C MCU Family
--With Initializing a Member of a Structure Using an Expression Containing the sizeof Operator--

Please take note of the following problem in using the C compiler packages for the M16C MCU family:

- With initializing a member of a structure using an expression containing the sizeof operator

1. Products and Versions Concerned

(1) The C compiler package for the R32C series V.1.01 Release 00
(2) The C compiler package for the M32C series, M3T-NC308WA (NOTE 1)
 V.5.10 Release 1 through V.5.41 Release 01
(3) The C compiler package for the M16C series, M3T-NC30WA (NOTE 2)
 V.5.10 Release 1 through V.5.44 Release 00

NOTES:
1. The M32C series is the generic name of the M32C/80 and M16C/80 series.
2. The M16C series is the generic name of the M16C/60, /30, /20, /10, /Tiny, and R8C/Tiny series.

2. Description

If you initialize a member of a structure using an expression containing the sizeof operator, the compiler's system error or Windows' error may arise.

2.1 Conditions

This problem occurs if the following conditions are all satisfied:
(1) Compile options -Oconst(-OC) and -O3(-O) are selected, or any one of the options -O4, -O5, -OR_MAX, and -OS_MAX is selected.

(2) A const-qualified structure is defined outside of a function.
(3) A member of the structure in (2) is initialized using
   an expression containing the sizeof operator.
(4) The member of the structure in (2) is at the top of the structure.
(5) An expression references the member of the structure in (2)
   using the dot (.) operator.

2.2 Example

```
--------------------------------------------------------------------------
   struct S{
        int  m;                      /* Condition (4) */
   };

   const struct S t = {2+sizeof(int)};  /* Conditions (2) and (3) */

   int  n;

   void    func(void)
   {
        n = t.m;                     /* Condition (5) */
   }
--------------------------------------------------------------------------
```

3. Workaround
Initialize the member of the structure involved using a constant.

```
--------------------------------------------------------------------------
   struct S{
        int  m;
   };

   const struct S t = {4};  /* Initialized using a constant */

   int  n;

   void    func(void)
   {
        n = t.m;
   }
--------------------------------------------------------------------------
```

4. Schedule of Fixing the Problem
We plan to fix this problem in the next release of the products.