RENESAS Tool News

A Note on Using C Compiler Package for R8C and M16C Families
--With Calculating Stack Usage--

Please take note of the following problem in using C Compiler Package for R8C and M16C Families:

- With calculating stack usage

The product is a set of three compiler packages: the C compiler package for the R32C series, the C compiler package for the M32C series (M3T-NC308WA), and the C compiler package for the M16C series and the R8C family (M3T-NC30WA).

1. Products and Versions Concerned
(1) C compiler package for the M16C series and the R8C family (M3T-NC30WA)
V.5.40 Release 00 through V.5.45 Release 00
(2) C compiler package for the M32C series (M3T-NC308WA)
V.5.40 Release 00 through V.5.41 Release 01
(3) C compiler package for the R32C/100 series
V.1.01 Release 00 and V.1.02 Release 00

2. Description
If stack usage is calculated by using stack-analyzing tool Call Walker
or STK Viewer, a value less than the actual stack usage may be displayed.

2.1 Conditions
This problem arises if all the conditions in any of the following
five condition groups are satisfied:

Condition group 1
(1) Any of the products tond versions concerned is used.
(2) Optimizing option -OR or -OR_MAX (-ORM) is used.
(3) Optimizing option -Ono_asmopt (-ONA) is not used.

Condition group 2

(1) In the C compiler package for the M16C series and the R8C family
(M3T-NC30WA), any of the versions concerned is used.

(2) Compile option -fauto_over_255 (-fAO2) is used.

(3) A function exists in which the total number of bytes of the auto
variables exceeds 255. Here, the auto variables include the
temporary variables automatically generated by the compiler.

Condition group 3

(1) In the C compiler package for the R32C/100 series, either
V.1.01 Release 00 or V.1.02 Release 00 is used.

(2) The #pragma TASK preprocessing directive is issued to a function.

Condition group 4

(1) In the C compiler package for the R32C/100 series, either
V.1.01 Release 00 or V.1.02 Release 00 is used.

(2) The #pragma INTERRUPT preprocessing directive is issued to
a function.

(3) In the preprocessing directive in (2), /B is not used.

(4) Within the function in (2) exist auto variables, which include the
temporary variables automatically generated by the compiler.

Condition group 5

(1) In the C compiler package for the R32C/100 series, either
V.1.01 Release 00 or V.1.02 Release 00 is used.

(2) The #pragma preprocessing directive INTHANDLER or INTCALL is
issued to a function.

(3) Within the function in (2) exist auto variables, which include the
temporary variables automatically generated by the compiler.

2.2 Example

An example is shown below where the C compiler package for the R32C/100
series V.1.01 Release 00 is used, and all the conditions in Condition
group 1 are satisfied.

```
----------------------------------------
long long val1;
long long val2;
long long val3;
long long val4;
long long val5;
long long val6;
void func1(void)
```

```
{
    val1 = 1;
    val2 = 2;
    val3 = 3;
    val4 = 4;
}
void func2(void)
{
    val1 = 1;
    val2 = 2;
    val3 = 3;
    val5 = 5;
}
void func3(void)
{
    val1 = 1;
    val2 = 2;
    val6 = 6;
}
```

In the above example, the correct numbers of bytes of the three functions are as follows:

    func1   12 bytes
    func2   12 bytes
    func3   8 bytes

However, they are displayed as shown below when stack usage is calculated by Call Walker.

    func1   4 bytes
    func2   4 bytes
    func3   4 bytes

3. Workaround

When Call Walker or STK Viewer is used, estimate that stack usage is greater than the size that is displayed and then determine the required size of the stack area after evaluating it deliberately.

4. Schedule of Fixing the Problem

We plan to fix this problem when we release the following versions of the products concerned:

 - C compiler package for the M16C series and the R8C family
     V.5.45 Release 01
 - C compiler package for the M32C series V.5.42 Release 00
 - C compiler package for the R32C/100 series V.1.02 Release 01