

A Note on Using the C Compiler Package for the R8C Family and the M16C Family of MCUs

Please take note of the following problem in using the C compiler package for the R8C family and the M16C family of MCUs:

- With the function for automatically generating variable vector tables

The C compiler package for the R8C family and the M16C family is a set of the following three C compiler packages:

- the C compiler package for the M16C series and R8C family(M3T-NC30WA)
- the C compiler package for the M32C series (M3T-NC308WA)
- the C compiler package for the R32C series

1. Products and Versions Concerned

- (1) The C compiler package for the M16C series and R8C family(M3T-NC30WA)
V.5.30 Release 1 through V.5.45 Release 00A
- (2) The C compiler package for the M32C series (M3T-NC308WA)
V.5.20 Release 1 through V.5.41 Release 01A
- (3) The C compiler package for the R32C series
V.1.01 Release 00 through V.1.02 Release 00

2. Description

When the linkage editor automatically generates a variable vector table, the codes on the table may incorrectly be placed.

NOTICE:

Even if the codes are placed incorrectly, these errors cannot be checked in the map file.

3. Conditions

The conditions under which the problem arises are depending on the version of your C compiler package as follows:

Group A

If your product and its version fall in the following, see Section 3.1 below:

- (1) The C compiler package for the M16C series and R8C family (M3T-NC30WA)
V.5.30 Release 1 through V.5.30 Release 02
- (2) The C compiler package for the M32C series (M3T-NC308WA)
V.5.20 Release 1 through V.5.20 Release 02

Group B

If your product and its version fall in the following, see Section 3.2 below:

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

3.1 Conditions for Group A

The problem may arise if the following conditions are all satisfied:

- (1) The function for automatically generating variable vector tables is in effect. (See NOTE 1.)
This is the case where the -fMVT option is used in the compiler, the assembler, and the linker.

If you are using real-time OS M3T-MR30/4, M3T-MR308/4, or M3T-MR100/4, this problem does not arise because the compiler's function for automatically generating variable vector tables is not used.

- (2) The software interrupt vector 0 for a variable vector table is selected.
- (3) In the generated variable vector table exists an unoccupied area, which contains no value.

If an unoccupied area exists, the following warning message is dispatched at linking:

Warning (***) : The free area's address in vector table isn't specified. (See NOTE 2.)

In either of the following cases, all the areas are filled with values, so this problem does not arise:

- (a) The link option -VECT is selected.

(b) The global label whose name is "dummy_int" is defined.

NOTES:

1. The section name of the automatically generated vector table is "__NC_rvector".
2. The symbol *** in the above message denotes a linker name, ln30 or ln308, depending on the linker you are using.

3.2 Conditions for Group B

The problem may arise if the following conditions are all satisfied:

- (1) The function for automatically generating variable vector tables is in effect. (See NOTE 3.)

This is the case where an interrupt vector number is specified in the definition of the interrupt handling function. That is, vect= is used in #pragma interrupt, or assembler directive command .RVECTOR is used in assembly language.

If you are using real-time OS M3T-MR30/4, M3T-MR308/4, or M3T-MR100/4, this problem does not arise because the compiler's function for automatically generating variable vector tables is not used.

- (2) The software interrupt vector 0 for a variable vector table is selected.

- (3) In the generated variable vector table exists an unoccupied area, which contains no value.

If an unoccupied area exists, the following warning message is dispatched at linking:

Warning (***) : The free area's address in vector table isn't specified. (See NOTE 4.)

In either of the following cases, all the areas are filled with values, so this problem does not arise:

- (a) The link option -VECT is selected.
- (b) The global label whose name is "dummy_int" or "__dummy_int" is defined.

NOTES:

3. The section name of the automatically generated vector table is "vector".
4. The symbol *** in the above message denotes a linker name, ln30, ln308, or ln100, depending on the linker you are using.

4. Example

An example in the C compiler package for the M16C series and R8C family

V.5.45 Release 00A (in Group B) is shown below.

<intp.c>

```
-----  
#pragma interrupt  _brk(vect=0) /* Conditions (1), and (2) */  
void _brk(void);  
void _brk(void){}  
-----
```

<main.c>

```
-----  
void main()  
{  
}  
-----
```

Command line input

```
nc30 intp.c main.c -R8C -ln30 "-order program=00E000,vector=00FEDC"
```

In the above example, the incorrect codes are generated in the Motorola S-record format file as follows:

Incorrect codes

```
-----  
S0030000FC  
S20900E000ECFDEDBFFB86  
S20900E006F300E000003D  <- Software interrupt vector 0 begins.  
S804000000FB  
-----
```

Here, the software interrupt vector 0 does not begin at address 00FEDCh but at address 00E007h (at the end of the program section).

The correct codes, in which the software interrupt vector 0 begins at address 00FEDCh, are as follows:

Correct codes

```
-----  
S0030000FC  
S20900E000ECFDEDBFFB86  
S20500E006F321  
S20800FEDC00E000003D  <- Software interrupt vector 0 begins.  
S804000000FB  
-----
```

Even if incorrect codes are generated in the Motorola S-record format file, however, the vector section begins at its correct address in the map file, as shown below.

```

-----
#####
# (2) SECTION INFORMATION          #
#####
# SECTION          ATR TYPE  START  LENGTH ALIGN MODULENAME
program           REL CODE   00E000 000005 2   intp
                  REL CODE   00E006 000001 2   main
vector            REL ROMDATA 00FEDC 000100    intp
-----

```

5. Workarounds

To avoid this problem, use any of the following methods:

- (1) Specify the values of the unoccupied areas in the variable vector table by using the link option `-VECT`.
- (2) Define the function `"dummy_int"` in assembly language.
- (3) Define the function `"_dummy_int"` in C language if your product is included in Group B.

Example of method (2):

```

-----
.glb dummy_int
dummy_int:
    reit
-----

```

Example of method (3):

```

-----
#pragma interrupt _dummy_int()
void _dummy_int(void);
void _dummy_int(void){}
-----

```

6. Notices

- (1) If you use the start-up file written in C language, the `"_dummy_int"` function is defined in the `fvector.c` file. So this problem does not arise.
- (2) If you use the start-up file in assembly language, the `"dummy_int"` function is defined in the `ncrt0.a30` file. So this problem does not arise also.
- (3) If you do not use the start-up file included in the C compiler

package, beware that the problem may arise.

7. Schedule of Fixing the Problem

We plan to fix this problem in the next version of each component package.

[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.