

RENESAS TOOL NEWS on September 16, 2004: RSO-M3T-CC32R-040916D

A Note on Using C Compiler Package M3T-CC32R

Please take note of the following problem in using the M3T-CC32R, a C-compiler package for the M32 family of MCUs:

- On initializing a two-dimensional array of type char using an initializer
-

1. Versions Concerned

M3T-CC32R V.1.00 Release 1 through V.4.30 Release 00

2. Description

If a two-dimensional array of type char is declared and at the same time initialized using a specific type of initializer that contains string literals, the compiler will tell the following error message:

```
cg32r: "xxxx", line XX: internal error: illegal IL, size of initializer is larger than  
name size.
```

Here, xxxx and XX denote a file name and a line number respectively.

2.1 Conditions

This problem occurs if the following conditions are all satisfied:

- (1) A two-dimensional array of type char (except for type pointer) is declared with an initializer.
- (2) In the declaration of the array in (1), its size is neglected.
- (3) The initializer in (1) contains two or more string literals (for example, "ab" and "cd").
- (4) Among the string literals in (3), the first is enclosed with braces.

2.2 Examples

1. Statically Initialized Array

Source file: sample1.c

```
-----  
char array1[][2] = { /* Conditions (1) and (2) */  
    {"ab"}, /* Conditions (3) and (4) */  
    "cd", /* Condition (3) */  
    "ef" /* Condition (3) */  
};  
-----
```

2. Dynamically Initialized Array

Source file: sample2.c

```
-----  
extern void array_func(char [][][6]);  
void func2(void)  
{  
    char array2[][6] = { /* Conditions (1) and (2) */  
        {"5678"}, /* Conditions (3) and (4) */  
        {"1234"}, /* Condition (3) */  
    };  
    array_func(array2);  
}  
-----
```

3. Workaround

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

- (1) Specify the size of the array.

Circumvention of source file sample1.c

```
-----  
-----  
char array1[3][2] = { /* Size of 3 specified */  
    {"ab"},  
    "cd",  
    "ef"  
};  
-----  
-----
```

Circumvention of source file sample2.c

```
-----
extern void array_func(char [][][6]);
void func2(void)
{
    char array2[2][6] = { /* Size of 2 specified */
        {"5678"},
        {"1234"},
    };
    array_func(array2);
}
-----
-----
```

(2) Remove the braces enclosing the first string literal.

Circumvention of source file sample1.c

```
-----
-----
char array1[][2] = {
    "ab", /* Braces removed */
    "cd",
    "ef"
};
-----
-----
```

Circumvention of source file sample2.c

```
-----
-----
extern void array_func(char [][][6]);
void func2(void)
{
    char array2[][6] = {
        "5678", /* Braces removed */
        {"1234"},
    };
    array_func(array2);
}
-----
-----
```

4. **Schedule of Fixing the Problem**

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

[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.