[Notes]
C Compiler Package for RL78 Family

R20TS0331EJ0100
Rev.1.00
Jul. 16, 2018

Outline

When using the CC-RL C compiler package for the RL78 family, note the following point.

1. Point for caution regarding the static variable declaration of an array, structure, or union that has an initializer (CCRL#019)

   Note: The number which follows the description of a precautionary note is an identifying number for the precaution.

1. Point for caution regarding the static variable declaration of an array, structure, or union that has an initializer (CCRL#019)

1.1 Applicable Products

CC-RL V1.00.00 to V1.06.00

1.2 Details

Assume that a function contains multiple array-type, structure-type, or union-type static variable declarations with initializers specified. If the address of a variable in the upper-layer block is specified as the initializer of a variable in the lower-layer block, an internal error might occur or invalid code might be generated.

1.3 Conditions

An internal error occurs when conditions (1) to (3) below are met and either of conditions (4) and (5) is met:

An invalid code may be generated rather than an internal error when conditions (1) to (4) and (6) below are met:

(1) The function contains a variable declaration that meets all the following conditions:

(1-1) static specification is contained.

(1-2) The variable is the array type, structure type, or union type.

(1-3) An initializer is specified.

(2) A variable is declared with the same conditions as (1) in a lower-layer block in the block that contains the variable declaration in (1).

(3) The address of the variable in (1) is specified for the initializer of the variable in (2).

(4) The first block of the function contains the variable declaration in (1).

(5) CC-RL V1.03.00 or a later version is used, and the variable declaration in (1) is written in a lower-layer block than the first block of the function.

(6) A variable that meets the conditions in (1) is declared by one of the following methods:

(6-1) The variable is declared before the variable in (1) in the block in (4).

(6-2) The variable is declared in a block that was specified before the block in (2).

(6-3) The variable is declared after the variable in (2) in the block in (2).

(6-4) CC-RL V1.02.00 or an earlier version is used, and the variable is declared in a lower-layer block in the block in (2).

1.4 Example

The following is an example of the problem. Characters in red are the parts that correspond to the conditions.

[Example of internal error generation]

```
 1   typedef struct A {
 2     short *objId;
 3   } TEST_PARAM;
 4
 5   int sub(TEST_PARAM * aParam);
 6   int var;
 7
 8   int main(void)
 9   {
10     static short obj[] = { 1 };      // Conditions (1) and (4)
11     {
12       static TEST_PARAM t ={ obj }; // Conditions (2) and (3)
13       sub(&t);
14     }
15     return 0;
16   }
```

- Line 10: Conditions (1) and (4) are met because array-type static variable "obj" with the initializer specified is declared in the first block (lines 9 to 16) of the main function.
- Line 12: Condition (2) is met because structure-type static variable "t" with the initializer specified is declared in the lower-layer block (lines 11 to 14) in the block (lines 9 to 16) containing the variable declaration of line 10.

  In addition, condition (3) is met because the address of array-type static variable "obj" (line 10) is specified as the initializer.

[Example of invalid code generation]

```
1   typedef struct A {
2     short *objId;
3   } TEST_PARAM;
4
5   int sub(TEST_PARAM * aParam);
6   int var;
7
8   int main(void)
9   {
10    static short obj[] = { 1 };      // Conditions (1) and (4)
11    {
12      static short dmy[] = { 2 };    // Condition (6-2)
13      var = dmy[0];
14    }
15    {
16      static TEST_PARAM t ={ obj }; // Conditions (2) and (3)
17      sub(&t);
18    }
19    return 0;
20  }
```

Line 10: Conditions (1) and (4) are met because array-type static variable "obj" with the initializer specified is declared in the first block (lines 9 to 20) of the main function.

Line 16: Condition (2) is met because structure-type static variable "t" with the initializer specified is declared in the lower-layer block (lines 15 to 18) in the block (lines 9 to 20) containing the variable declaration of line 10.

In addition, condition (3) is met because the address of the variable in line 10 is specified as the initializer.

Line 12: Condition (6-2) is met because, in addition to the variable declaration in line 10, array-type static variable "dmy" with the initializer specified is declared in the block (lines 11 to 14). This block was specified before the block containing the variable declaration (line 16) that meets the condition in (2).

In this case, structure-type variable "t" is initialized with the address of array-type variable "dmy", instead of the address of array-type variable "obj".

1.5 Workaround

To avoid this problem, take either of the following steps:

(1) Remove the static specification from the variable declaration in (2) under Conditions.

(2) Change the initializer in (3) under Conditions to an assignment expression.

1.6 Schedule for Fixing the Problem

The problem will be fixed in CC-RL V1.07.00. This information will be available from July 20.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jul. 16, 2018	-	First edition issued

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061 Japan

Renesas Electronics Corporation

■Inquiry
https://www.renesas.com/contact/