

[Notes]

R20TS0065EJ0100

Rev.1.00

Aug. 01, 2016

C Compiler Package for RL78 Family

Outline

When using the CC-RL C Compiler Package for the RL78 family, take note of the problem described in this note regarding the following point.

1. The result of shift operation becoming incorrect (CCRL#011)

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

1. The Result of Shift Operation Becoming Incorrect (CCRL#011)

1.1 Applicable Product

CC-RL V1.00.00 to V1.01.00

1.2 Details

The results of shift operations may become incorrect.

1.3 Conditions

This problem arises if the following conditions are all met.

- (1) The -Onothing option is not designated.
- (2) The program includes a shift operation.
- (3) The result of the shift operation is directly assigned to a data area*1 with less than 8 bits or is assigned to a data area*1 with less than 8 bits via a local variable that is an integer-type variable with 16 or more bits and not declared volatile.

*1: The area of a variable, bit field, argument, or return value.

- (4) The size of the thing to be shifted in (2) is fixed so that the result can be correctly expressed in 8 bits at the time of the shift operation.
- (5) The size of the shift in (2) is fixed to a number which is 8 or more and less than the number of bits after integer extension to suit the size of the shift in the shift operation.

Example: Code satisfying the conditions when the -Onothing option is also not designated

```
signed char c;
void func(void){
    signed int a = -1; /* Condition (4) */
    signed char b = 8; /* Condition (5) */
    c = a >> b;      /* Conditions (2)(3)(4)(5) */
}
```

1.4 Workaround

To avoid this problem, do any of the following.

- (1) Designate the -Onothing option.
- (2) Assign the result of the shift operation to the target data area via a temporary variable having the volatile attribute.
- (3) Separate processing of the shift operation into a function*1.

*1: Use the #pragma noline to inhibit inline expansion of the function.

- (4) Assign the variable for the thing to be shifted or the size of the shift to a variable outside the function where the shifting takes place.

Example of workaround (2)

```
signed char c;
void func(void){
    signed int a = -1;
    signed char b = 8;
    volatile int d = a >> b; /* Workaround (2) */
    c = d;                    /* Workaround (2) */
}
```

1.5 Schedule for Fixing the Problem

The problem has been already rectified in CC-RL V1.02.00.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Aug. 01, 2016	-	First edition issued

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

■Inquiry

<http://www.renesas.com/contact/>

Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.

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.

All trademarks and registered trademarks are the property of their respective owners.