

[Note]

R20TS0652EJ0100

Rev.1.00

Jan. 16, 2021

C Compiler Package for RL78 Family (CCRL#026)

Overview

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

1. Using the -Ointermodule option (CCRL#026)

Note: The number following the note is an identification number for the note.

1. Using the -Ointermodule option (CCRL#026)

1.1 Applicable products

CC-RL V1.01.00 to V1.09.00

1.2 Details

When the -Ointermodule option is used, access to static variables may be deleted incorrectly.

1.3 Conditions

If all of the conditions from (1) to (8) are met, access to a variable in condition (7) may be deleted incorrectly.

- (1) -Ointermodule or-Owhole_program is specified. ^(Note 1)
- (2) -Onothing is not specified.
- (3) There is a structure-type or union-type having a pointer-type member.
- (4) The pointer-type member in (3) is not const-qualified.
- (5) There is a const-qualified static variable^(Note 2) of the structure-type or union-type in (3).
- (6) The initial value of the pointer-type member (3) of the static variable in (5) is the address of a variable.
- (7) The variable with the address in (6) is a static variable^(Note 2) that is not const-qualified.
- (8) There is a const-qualified pointer-type static variable^(Note 2) whose initial value is the address of the static variable in (5).

Note 1: When -Owhole_program is specified, -Ointermodule is also implicitly specified.

Note 2: A static variable corresponds to a global variable or a 'static' variable.

1.4 Examples

An example of the problem is shown below. The parts corresponding to the error conditions are shown in red.

Example

ccrl -cpu=S3 -Osize -Ointermodule tp.c (1) (2)

```

/* tp.c */
int GGG; /* (7) */
typedef struct { /* (3) */
    int* mmm; /* (4) */
}Str;
const Str SSS = { /* (5) */
    &GGG /* (6) */
};
const Str* PPP = &SSS; /* (8) */

int func(void) {
    GGG = 1;
    *(PPP->mmm) = 2;
    return GGG;
}

```

In this example, although function func() is supposed to return 2 because PPP->mmm points to the address of the variable GGG, it returns 1.

1.5 Workaround

You can avoid this problem by one of the following methods:

- (a) Do not specify either -Ointermodule or -Owhole_program.
- (b) Specify -Onothing.
- (c) Remove the const qualifier from the structure-type or union-type static variable in condition (5).
- (d) Remove the const qualifier from the pointer-type static variable in condition (8).

1.6 Schedule for fixing the problem

This problem will be fixed in CC-RL V1.10.00. This version will be released in January 2021.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jan.16.21	-	First edition issued

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 URL in the Tool News also may be subject to change or become invalid without prior notice.

Corporate Headquarters

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

www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:

www.renesas.com/contact/