[Notes]
C Compiler Package for RH850 Family

R20TS0583EJ0100
Rev.1.00
Jun. 01, 2020

Outline

When using the C compiler package for RH850 CC-RH, note the following point.

1. Note on writing the constant value 0 to a 2-byte area (No.29)

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

1. Note on Writing the Constant Value 0 to a 2-Byte Area (No.29)

1.1 Applicable Products

CC-RH V1.00.00 to V2.02.00

1.2 Details

When you compile a source code that writes the constant value "0" to a 2-byte long memory area, a code that writes to an address shifted by 1 byte off the intended address is generated.

Executing this code leads to an unintended execution result or generates a misaligned access exception.

If this is the case, a warning message with either of the following numbers is output at the time of building.

W0550010 (V1.02.00 or earlier)

W0550019 (V1.03.00 or later)

1.3 Conditions

This problem arises if the following conditions are all met:

(1) Either a structure type variable having a 2-byte type element (member) or a 2-byte type array is used. (Here, "2-byte type" also includes a 16-bit bit field and __fp16 type.)

(2) The constant value "0" is assigned to the 2-byte element (1) by using either of the following formats:
(2-1) Through a pointer ((*p).m or p->m format)
(2-2) A subscript operator whose element specification is not a constant expression (p[i] format)

(3) In (2), the value is assigned to an element other than the first member of the type (1).

Note: The above condition might apply when a packing value is specified as 2 via an Xpack option or #pragma pack directive, in which case a 4-byte or 8-byte member is also accessed in units of 2 bytes.

1.4 Examples

Below is an example of the error. The parts corresponding to the error conditions are shown in red.

ccrh -Osize tp.c

```
/* tp.c */
struct ST {
  char m1;
  char m2;
  short m3;       /* (1),(3) */
  short m4;       /* (1),(3) */
};

void func(struct ST *pst) {
  int i;
  for (i=0;i<10;++i) {
    (pst + i)->m2 = 1;
    (pst + i)->m3 = 0;    /* (2-1) */
    pst[i].m4 = 0;        /* (2-2) */
  }
}
```

When the constant value "0" is written to members m3 and m4, the value is written to an address shifted by 1 byte.

1.5 Workaround

Change the relevant members to a type other than the 2-byte type.

If the above condition applies due to the use of the structure type packing function, remove the packing.

1.6 Schedule for Fixing the Problem

This problem will be fixed in CC- RH V2.03.00. The release date has not yet been determined.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jun.01.20	-	First edition issued

Corporate Headquarters

TOYOSU FORESIA, 3- 2- 24 Toyosu,
Koto-ku, Tokyo 135- 0061, Japan
www.renesas.com

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/

Trademarks

TS Colophon 4.1