

[Notes]

R20TS0515EJ0100

Rev.1.00

Dec. 01, 2019

C Compiler Package for R32C Series

C Compiler Package for M32C Series

C/C++ Compiler Package for M16C Series and R8C Family

Outline

When using the products in the title, note the following points.

1. Problem with the atan2f and atan2 functions returning incorrect values (C Compiler Package for R32C Series only)
2. Function whose return value is of structure type or union type

1. Problem with the atan2f and atan2 Functions Returning Incorrect Values

1.1 Applicable Products

C Compiler Package for R32C Series V.1.01 Release 00 to V.1.02 Release 01

1.2 Details

Arithmetic operation by using the atan2f or atan2 function may produce an incorrect return value (0) with EDOM as errno.

1.3 Conditions

This problem (incorrect return value 0 with EDOM as errno) may arise if all of the following three conditions are met.

(1) One of the following conditions, (1-1) or (1-2), is met.

(1-1) The C source program uses the atan2f function.

(1-2) The C source program uses the atan2 function, any of the following compilation options, (1-2-1) to (1-2-3), is specified, and the calling function in the compilation result is atan2f.

(1-2-1) -fD32 (-fdouble_32)

(1-2-2) -ORM (-OR_MAX)

(1-2-3) -OSM (-OS_MAX)

(2) Either of the following standard library functions, (2-1) or (2-2), is used.

(2-1) nc100fpu.lib^(Note 1)

(2-2) nc100i16fpu.lib^(Note 2)

Note 1: Automatically used when compilation option -fUF (-fuse_FPU) is specified.

Note 2: Automatically used when compilation options -fUF (-fuse_FPU) and -fint_16(-fl16) are specified.

(3) Both of the following conditions, (3-1) and (3-2), are met when the first actual argument of the atan2f or atan2 function is x and the second actual argument is y.

(3-1) The absolute value of y is 1.08420202e-19F or more.

(3-2) The absolute value of the quotient (x/y) is within the range of [0.0F, 2.44140625e-4F].

1.4 Examples

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

[C source] (When the standard library nc100fpu.lib or nc100i16fpu.lib is used.)

```

1:  volatile float x, y, z;
2:  void main(void)
3:  {
4:      x = 3.06658759e-23F; // Condition(3-2)
5:      y = 1.53329391e-19F; // Condition(3-1) (3-2)
6:      z = atan2f(x, y);    // Condition(1)
7:  }
```

Lines 4 and 5:

Condition (3-1) is met because a value larger than 1.08420202e-19F is specified for variable y (which is specified as the second actual argument of the atan2f function).

Also, condition (3-2) is met because the absolute value of the quotient to divide the first actual argument of the atan2f function by the second actual argument is within the range of [0.0F, 2.44140625e-4F].

Line 6:

Condition (1) is met because the atan2f function is used.

1.5 Workaround

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

- (1) When using the atan2 function, do not use compilation options (1-2-1) to (1-2-3).
- (2) Do not use standard libraries (2-1) and (2-2).

1.6 Schedule for Fixing the Problem

There is no schedule for fixing this problem.

2. Function Whose Return Value is of Structure Type or Union Type

2.1 Applicable Products

- C Compiler Package for R32C Series V.1.00 Release 01 to V.1.02 Release 01
- C Compiler Package for M32C Series V.1.00 Release 1 to V.5.42 Release 00
- C/C++ Compiler Package for M16C Series and R8C Family V.1.00 Release 1 to V.6.00 Release 00

2.2 Details

When using a function whose return value is of the structure or union type, the address specifying the area for storing the return value may not be set correctly, and the content in the RAM area may be destroyed when storing the return value.

2.3 Conditions

When all of the following conditions (1) to (4) are met, the address specifying the area for storing the return value of the function in (1) is incorrectly set as the address specifying the area of the left side in (4-1), or the address specifying the area for storing the return value by the return statement in (4-2). Therefore, storing the return value in the function processing in (1) may destroy the content of the RAM area.

- (1) A function whose return value is of the structure or union type is called.
- (2) Only part of the return value of the function in condition (1) is read by using a . operator.
 Example 1: `f().m`
 Example 2: `f().m[2]`
- (3) The value read in condition (2) is of the structure or union type.
- (4) The value read in condition (2) is used in operand (4-1) or (4-2).
 (4-1) Right operand of an assignment operator (= operator)
 Example: `x = f().m;`
 (4-2) Operand of a return statement
 Example: `return f().m;`

2.4 Examples

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

[C source]

```

1:  typedef struct struct_st_m {
2:  -Omitted-
3:  }st_m;
4:  typedef struct struct_st_f2 {
5:      unsigned char uc;
6:      st_m m;
7:  }st_f2;
8:
9:  st_f2 f2(void);
10:
11:  st_m f1(void)
12:  {
13:      return f2().m; // Condition(1) (2) (3) (4-2)
14:  }
```

Line 13:

Condition (1) is met because the f2 function whose return value is of structure type st_f2 is called.

Condition (2) is met because only part of the return value is read by using a . operator, and condition (3) is met because the read value is of structure type st_m.

Condition (4-2) is met because the read value is used for an operand of the return statement.

2.5 Workaround

Prepare a separate variable for which volatile is specified by the same type as the return value of the function called in condition (1).

Assign a return value to the variable, and then read the assigned value.

[C source] (Before modification)

1:	st_m f1(void)
2:	{
3:	return f2().m;
4:	}

[C source] (After modification)

1:	st_m f1(void)
2:	{
3:	volatile st_f2 temp;
4:	temp = f2();
5:	return temp.m;
6:	}

2.6 Schedule for Fixing the Problem

There is no schedule for fixing this problem.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Dec.01.19	-	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 URLs 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/.