

【注意事項】

R20TS0515JJ0100

Rev.1.00

2019.12.01 号

R32C シリーズ用 C コンパイラパッケージ

M32C シリーズ用 C コンパイラパッケージ

M16C シリーズ, R8C ファミリー用 C/C++コンパイラパッケージ

概要

タイトルに記載している製品の使用上の注意事項を連絡します。

1. atan2f 関数、atan2 関数の戻り値が不正となる注意事項 (R32C シリーズ用 C コンパイラパッケージのみ)
2. 戻り値の型が構造体型、または共用体型である関数の注意事項

1. atan2f 関数、atan2 関数の戻り値が不正となる注意事項

1.1 該当製品

R32C シリーズ用 C コンパイラパッケージ V.1.01 Release 00 ~ V.1.02 Release 01

1.2 内容

atan2f または atan2 関数を使用して演算を行うと、errno に EDOM を設定して関数の戻り値が不正な値 0 となる場合があります。

1.3 発生条件

以下の(1)~(3)のすべての条件を満たす時に、errno に EDOM を設定して関数の戻り値が不正な値 0 となる場合があります。

(1) 以下(1-1)または(1-2)のいずれかを満たす。

(1-1) C ソース・プログラムで atan2f 関数を使用している

(1-2) C ソース・プログラムで atan2 関数を使用し、かつ以下の(1-2-1)~(1-2-3)いずれかのコンパイルオプションを指定して、コンパイル結果での呼び出し関数が atan2f になる

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

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

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

(2) 以下の(2-1)または(2-2)のいずれかの標準ライブラリを使用している。

(2-1) nc100fpu.lib (注1)

(2-2) nc100i16fpu.lib (注2)

注1: コンパイルオプション-fUF (-fuse_FPU)を指定している場合は自動的に使用します。

注2: コンパイルオプション-fUF (-fuse_FPU)かつ-fint_16(-fl16)を指定している場合は自動的に使用します。

(3) atan2f または atan2 関数の第 1 実引数を x、第 2 実引数 y として以下の(3-1)および(3-2)をすべて満たす。

(3-1) y の絶対値が、1.08420202e-19F 以上である

(3-2) x を y で除算した商(x/y)の絶対値が、[0.0F, 2.44140625e-4F]の範囲内である

1.4 発生例

以下に、発生例を記します。赤文字が発生条件の該当箇所です。

【C ソース】（標準ライブラリ nc100fpu.lib または nc100i16fpu.lib を使用している場合）

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

4~5 行目：

atan2f 関数の第 2 実引数に指定する変数 y に 1.08420202e-19F より大きい値を設定しているため、発生条件(3-1)を満たします。

また、atan2f 関数の第 1 実引数を第 2 実引数で除算した商の絶対値が[0.0F, 2.44140625e-4F]の範囲内であるため、発生条件(3-2)を満たします。

6 行目：

atan2f 関数を使用しているため発生条件(1)を満たします。

1.5 回避策

下記(1)、(2)のいずれかにより回避可能です。

(1) atan2 関数を使用している場合は、発生条件(1-2-1)~(1-2-3)のコンパイルオプションを使用しないでください。

(2) 発生条件(2-1)または(2-2)の標準ライブラリを使用しないでください。

1.6 恒久対策

改修の予定はありません。

2. 戻り値の型が構造体型、または共用体型である関数の注意事項

2.1 該当製品

- R32C シリーズ用 C コンパイラパッケージ V.1.00 Release 01 ~ V.1.02 Release 01
- M32C シリーズ用 C コンパイラパッケージ V.1.00 Release 1 ~ V.5.42 Release 00
- M16C シリーズ,R8C ファミリ用 C/C++コンパイラパッケージ V.1.00 Release 1 ~ V.6.00 Release 00

2.2 内容

戻り値の型が構造体型、または共用体型である関数を使用する場合、戻り値を格納するための領域を指すアドレスが正しく設定されず、戻り値の格納時に RAM 領域の内容が破壊される可能性があります。

2.3 発生条件

以下の(1)~(4)のすべての条件を満たす場合に、(1)の関数の戻り値を格納する領域を指すアドレスが、(4-1)の左辺の領域を指すアドレス、または(4-2)の return 文による戻り値を格納する領域を指すアドレスに誤って設定されます。そのため(1)の関数内の処理における戻り値の格納によって RAM 領域の内容が破壊される可能性があります。

- (1) 戻り値の型が構造体型または共用体型である関数を呼び出している。
- (2) 発生条件(1)の関数の戻り値に対して、. 演算子を使用して値の一部のみを読み出している。
例 1 : f().m
例 2 : f().m[2]
- (3) 発生条件(2)で読み出した値の型は、構造体型または共用体型である。
- (4) 発生条件(2)で読み出した値を(4-1)または(4-2)のオペランドで使用している。
(4-1) 代入演算子(=演算子)の右オペランド
例 : x = f().m;
(4-2) return 文のオペランド
例 : return f().m;

2.4 発生例

以下に、発生例を記します。赤文字が発生条件の該当箇所です。

【C ソース】

```
1:   typedef struct struct_st_m {
2:   ~省略~
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; // 発生条件(1)(2)(3)(4-2)
14:  }
```

13 行目 :

戻り値の型が構造体型 st_f2 である関数 f2 を呼び出しているため発生条件(1)を満たします。

戻り値に対して、演算子を使用して値の一部のみを読み出しているため発生条件(2)を満たしており、読み出した値の型は構造体型 st_m であるため発生条件(3)を満たします。

また、読み出した値を return 文のオペランドで使用しているため発生条件(4-2)を満たします。

2.5 回避策

発生条件(1)で呼び出す関数の戻り値と同じ型で volatile 指定された変数を別途用意してください。
変数に戻り値を一旦代入してから、代入された値を読み出してください。

【C ソース】（変更前）

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

【C ソース】（変更後）

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

2.6 恒久対策

改修の予定はありません。

以上

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Dec.01.19	-	新規発行

本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。

過去のニュース内容は発行当時の情報をもとにしており、現時点では変更された情報や無効な情報が含まれている場合があります。

ニュース本文中の URL を予告なしに変更または中止することがありますので、あらかじめご承知ください。

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24 (豊洲フォレシア)

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。