

【注意事項】

R20TS0334JJ0100

Rev.1.00

2018.07.16 号

RX ファミリ用 C/C++コンパイラパッケージ

概要

RX ファミリ用 C/C++コンパイラパッケージ CC-RX の使用上の注意事項を連絡します。

1. 初期化子の指定がある配列型、構造体型、または共用体型の `static` 変数宣言に関する注意事項 (No.49)

注： 注意事項の後ろの番号は、注意事項の識別番号です

1. 初期化子の指定がある配列型、構造体型、または共用体型の `static` 変数宣言に関する注意事項 (No.49)

1.1 該当製品

CC-RX V2.00.00～V2.08.00

1.2 内容

初期化子の指定がある配列型、構造体型、または共用体型の `static` 変数宣言が関数内に複数あり、下位階層ブロックにある変数の初期化子に、上位階層ブロックにある変数のアドレスを指定すると内部エラーが発生、または不正なコードを生成する可能性があります。

1.3 発生条件

下記(1)～(3)の条件を全て満たし、かつ(4)または(5)のいずれかの条件を満たす場合に内部エラーが発生します。また、下記(1)～(4)の条件を全て満たし、かつ(6)の条件を満たす場合に、不正なコードを生成する可能性があります。

- (1) 関数内に(1-1)～(1-3)の条件を全て満たす変数宣言をする。

- (1-1) `static` 指定をする
- (1-2) 配列型、構造体型、または共用体型の変数である
- (1-3) 初期化子の指定をする

- (2) (1)の変数宣言を記述したブロック内で下位階層ブロックに(1)と同じ条件の変数宣言をする。

- (3) (2)の変数の初期化子に(1)の変数のアドレスを指定する。

- (4) (1)の変数宣言を関数先頭のブロックに記述する。

- (5) V2.05.00 以降のバージョンを使用し、かつ(1)の変数宣言を関数先頭のブロックより下位階層ブロックに記述する。

- (6) (1)を満たす変数を、(6-1)～(6-4)のいずれかの記述で変数宣言をする。

- (6-1) (4)のブロック内で(1)の変数より前に変数宣言を記述する
- (6-2) (2)のブロックより前に記述したブロックに変数宣言を記述する
- (6-3) (2)のブロック内で(2)の変数より後に変数宣言を記述する
- (6-4) V2.04.00 以前のバージョンを使用し、かつ(2)のブロック内で下位階層ブロックに変数宣言を記述する

1.4 発生例

以下に、不正なコードを生成する例を記します。赤文字が発生条件の該当箇所です。

```

1:  typedef struct A {
2:     short *objId;
3: } TEST_PARAM;
4:
5:  int sub(TEST_PARAM * aParam);
6:  int var;
7:
8:  int main(void)
9:  {
10:     static short obj[] = { 1 };      // 条件(1)(4)
11:     {
12:         static short dmy[] = { 2 };  // 条件(6-2)
13:         var = dmy[0];
14:     }
15:     {
16:         static TEST_PARAM t = { obj }; // 条件(2)(3)
17:         sub(&t);
18:     }
19:     return 0;
20: }
```

- ・ 10 行目：main 関数の先頭ブロック(9～20 行目)内に初期化子の指定がある配列型の static 変数” obj” の宣言があるため、条件(1)と条件(4)に該当します。
- ・ 16 行目：10 行目の変数宣言が存在するブロック(9～20 行目)内で、下位階層ブロック(15～18 行目)に初期化子の指定がある構造体型の static 変数” t” の宣言があるため、条件(2)に該当します。また、初期化子として 10 行目の配列型変数” obj” のアドレスを指定しているため、条件(3)に該当します。
- ・ 12 行目：10 行目の変数宣言とは別に、初期化子の指定がある配列型の static 変数” dmy” の宣言があり、条件(2)に該当する 16 行目の変数宣言があるブロックより前に記述したブロック(11～14 行目)内に存在しているため、条件(6-2)に該当します。
この場合、構造体型変数” t” は配列型変数” obj” ではなく配列型変数” dmy” のアドレスで初期化されます。

1.5 回避策

以下のいずれかにより回避可能です。

- (1) 発生条件(2)の変数宣言で `static` 指定を外す。
- (2) 発生条件(3)の初期化子を代入式に変更する。

1.6 恒久対策

CC-RX V3.00.00 で改修します。

以上

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2018.07.16	-	新規発行

ルネサスエレクトロニクス株式会社

〒135-0061 東京都江東区豊洲 3-2-24 (豊洲フォレシア)

■総合お問い合わせ先

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

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。

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。