

## 【注意事項】

R20TS0127JJ0100

Rev.1.00

2017.01.16 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. 値が 1 ずつ減少する演算式を含むループの注意事項 (No.42)

注: 注意事項の後ろの番号は、注意事項の識別番号です。

## 1. 値が 1 ずつ減少する演算式を含むループの注意事項 (No.42)

## 1.1 該当製品

CC-RX V2.05.00

## 1.2 内容

ループ内において、ループの繰り返しごとに値が 1 ずつ減少する演算式を記述した場合に、そのループの繰り返し回数が不正に 1 回少なくなる場合があります。

## 1.3 発生条件

以下の条件をすべて満たす時に、(2)のループで現象が発生する場合があります。

- (1) -optimize=2 または -optimize=max を指定しているか、最適化オプションを指定していない。

- (2) (2-1)~(2-5)をすべて満たすループが存在する。

(2-1) break 文、goto 文 および return 文のいずれも含まない。

(2-2) ループ制御変数<sup>(注1)</sup>が (2-2-a) および (2-2-b)の両条件を満たす。

(2-2-a) 初期値 および 終了判定値が定数である<sup>(注2)</sup>。

(2-2-b) 増分値が 1 である<sup>(注3)</sup>。

(2-3) ループ内にループ帰納変数<sup>(注4)</sup>を含み、かつループの繰り返しごとに演算結果が 1 ずつ減少する加算式 または 減算式が存在する。

(2-4) (2-3)の初回演算結果は「(ループ繰り返し回数) - 1」である。

(2-5) (2-3)のループ帰納変数が(2-5-a) および (2-5-b)の両条件を満たす。

(2-5-a) 初期値が定数である<sup>(注2)</sup>。

(2-5-b) 増分値が 1 または -1 である<sup>(注5)</sup>。

注 1: ループの終了条件を制御する変数です。

注 2: 静的に定数とわかる場合も含みます。

注 3: 静的に 1 とわかる場合も含みます。

注 4: ループごとに一定値ずつ増加 または 減少する変数です。

ループ制御変数は必ずループ帰納変数となります。

注 5: 静的に 1 または -1 とわかる場合も含みます。

## 1.4 発生例

以下に、1.3 項(2) の発生例を記します。赤文字が発生条件の該当箇所です。

```

1:  int result;
2:  int sub(void)
3:  {
4:      volatile int v1, v2;
5:      int i;                // 変数 i はループ制御変数、かつループ帰納変数
6:      for (i=1; i<=8; i++) { // 条件(2-2)(2-5)
7:          if (i != 8) {
8:              v1 = 8-i;     // 条件(2-3)(2-4)
9:          }
10:         v2 = i;
11:     }
12:     return v2;
13: }
14:
15: void main(void)
16: {
17:     result = sub();
18: }

```

- 6～11 行目の 8 回ループする for ループ文内には break 文、goto 文 および return 文のいずれも含まないため、条件(2-1)に該当します。
- 6 行目において、ループ制御変数 i は、初期値が 1、終了判定値が 8 の定数、増分値は 1 であるため、条件(2-2)に該当します。
- ループ制御変数 i はループごとに 1 ずつ増加するため、ループ帰納変数でもあります。さらに、ループ内の 8 行目にループ帰納変数 i を含み、かつループの繰り返しごとに演算結果が 1 ずつ減少する演算式「v1 = 8-i;」が存在しますので、条件(2-3)に該当します。
- 8 行目の演算式の初回演算結果は 7 となるため、条件(2-4)に該当します。
- ループ帰納変数 i の初期値は 1、増分値は 1 であるため、条件(2-5)に該当します。

上記のとおり、6 行目～11 行目の for ループ文は、1.3 項(2)に該当します。従って、1.3 項(1)にも該当している場合、for ループ文は正しくは 8 回のところ 7 回しかループしません。17 行目の変数 result も 8 ではなく 7 となります。

## 1.5 回避策

以下のいずれかにより回避可能です。

- (1) 最適化オプション-optimize=0 または-optimize=1 を指定する。
- (2) ループの終了判定値を volatile 修飾した変数に置き換える。

```

5:  int i;
6:  volatile int j = 8
7:  for (i=1; i<=j; i++) {

```

## 1.6 恒久対策

V2.06.00 で改修します。(1 月 20 日公開予定)

以上

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2017.1.16	-	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

[http:// www.renesas.com/contact/](http://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。