

【注意事項】

R20TS0065JJ0100

Rev.1.00

2016.08.01 号

RL78 ファミリ用 C コンパイラパッケージ

概要

RL78 ファミリ用 C コンパイラパッケージ CC-RL の使用上の注意事項を連絡します。

1. シフト演算の結果が不正となる注意事項 (CCRL#011)

注: 注意事項の後ろの番号は、注意事項の識別番号です。

1. シフト演算の結果が不正となる注意事項 (CCRL#011)

1.1 該当製品

CC-RL V1.00.00 ~ V1.01.00

1.2 内容

シフト演算の結果が不正となる場合があります。

1.3 発生条件

以下のすべての条件を満たす時に発生します。

(1) -Onothing オプションを指定していない。

(2) シフト演算をしている。

(3) (2)のシフト演算の結果を 8 ビット以下のデータ領域^{注1}に直接代入しているか、または、16 ビット以上の整数型で、volatile でなく、かつローカルな変数を経由して 8 ビット以下のデータ領域^{注1}に代入している。

注1: 変数、ビットフィールド、引数、戻り値の領域を表します。

(4) (2)の被シフト数は、8 ビットで表現できる値であることがシフト演算時に確定している。

(5) (2)のシフト数は、8 以上かつ被シフト数の整数拡張後のビット幅未満であることがシフト演算時に確定している。

発生例: -Onothing オプションを指定していない場合

```
signed char c;
void func(void){
    signed int a = -1; /* 発生条件(4) */
    signed char b = 8; /* 発生条件(5) */
    c = a >> b;      /* 発生条件(2)(3)(4)(5) */
}
```

1.4 回避策

以下のいずれかの方法で回避できます。

- (1) `-Onothing` オプションを指定する。
- (2) シフト演算の結果を、`volatile` 修飾型のテンポラリ変数を経由して対象のデータ領域に代入する。
- (3) シフト演算の処理を関数化^{注1}する。

注1：関数は、`#pragma noline` を使用してインライン展開を抑止してください。

- (4) 被シフト数またはシフト数の変数への代入を、関数外で行う。

回避例：回避策(2)の場合

```
signed char c;
void func(void){
    signed int a = -1;
    signed char b = 8;
    volatile int d = a >> b; /* 回避策(2) */
    c = d;                    /* 回避策(2) */
}
```

1.5 恒久対策

CC-RL V1.02.00 で改修済みです。

以上

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2016.08.01	-	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://japan.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。