

【注意事項】

R20TS0671JJ0100

Rev.1.00

2021.03.16 号

RL78 ファミリ用 C コンパイラパッケージ
(注意事項 CCRL#027)

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. pack 機能の使用に関する注意事項 (CCRL#027)

注：注意事項の後ろの番号は、注意事項の識別番号です。

1. pack 機能の使用に関する注意事項 (CCRL#027)

1.1 該当製品

CC-RL V1.02.00~V1.10.00

1.2 内容

pack 機能を使用した場合、double 型または long long 型のメンバを持つ構造体型または共用体型の変数に対し、誤った領域にデータの書き込みをする場合があります。

1.3 発生条件

次の(1)から(5)のすべてを満たす場合に、誤った領域にデータの書き込みをする可能性があります。

- (1) -pack または#pragma pack^(注 1)を使用している。
- (2) double 型^(注 2)または long long 型のメンバを持っている構造体型または共用体型の変数を使用している。
- (3) (2)のメンバを volatile 修飾していない。
- (4) (2)のメンバに、(2)と同じ型の関数の戻り値を直接代入している。
- (5) (2)のメンバの先頭番地が奇数である。

注 1：#pragma pack は CC-RL V1.05 から使用できる機能です。

注 2：-dbl_size=8 を使用した場合に該当します。

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。

[発生例]

ccrl -cpu=S3 -Onothing -pack tp.c (1)

```
/* tp.c */
struct T1 {
    signed char m1;
    long long m2;    // (2)(3)
};
struct T1 gx1;
long long f1(void) {
    return 1;
}
void main(void) {
    gx1.m2 = f1();    // (4)
}
```

この例の場合は、関数 f1() の戻り値(long long 型)を、構造体型変数 gx1 のメンバ m2(long long 型の奇数アドレス)に直接代入しているので、誤った領域へのデータの書き込みが発生します。

1.5 回避策

以下のいずれかを行うことで回避できます。

- (a) -pack および #pragma pack を使用しない。
- (b) 当該メンバを volatile で修飾する。
- (c) 関数の戻り値を自動変数に代入してから当該メンバに代入する。
- (d) 当該メンバの先頭番地が偶数アドレスになるように当該変数の配置アドレスを調整する。

1.6 恒久対策

CC-RL V1.11.00 で改修する予定です。リリース時期は未定です。

以上

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Mar.16.21	-	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。