

【注意事項】

R20TS0377JJ0100

Rev.1.00

RL78 ファミリー用 C コンパイラパッケージ

2018.12.16 号

概要

RL78 ファミリー用 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. 構造体または共用体の整数型メンバの初期値がアドレス定数の場合の注意事項(CCRL#022)

注: 注意事項の後ろの番号は、注意事項の識別番号です。

1. 構造体または共用体の整数型メンバの初期値がアドレス定数の場合の注意事項 (CCRL#022)

1.1 該当製品

CC-RL V1.01.00 ~ V1.07.00

1.2 内容

構造体または共用体の整数型メンバの初期値がアドレス定数の場合に初期値データが不正となり、結果として不正動作となる可能性があります。

1.3 発生条件

以下の(1)~(3)の条件をすべて満たす場合に初期値データが不正となります。

- (1) 構造体型または共用体型の静的変数が整数型メンバを持つ。
- (2) (1)の整数型メンバの初期値がアドレス定数である。
- (3) (1)の整数型メンバのサイズが初期値のアドレス定数のサイズ^(注1)より大きい。

注1: アドレス定数のサイズは、オブジェクトを指すポインタのサイズと同じです。
初期化子の中でキャストをしても、(3)の発生条件に影響しません。

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】

```

1  unsigned char data; // 発生条件(3)
2  struct _st {
3      unsigned long ldata; // 発生条件(1)(3)
4  } st1[] = { 0x12345678, (unsigned long)&data, 0x12345678 }; // 発生条件(2)

```

3 行目: 構造体配列 st1 が整数型メンバ ldata を持つため発生条件(1)に該当します。

4 行目: 構造体メンバ ldata の初期化子にアドレス定数&data を記述しているため発生条件(2)に該当します。

1 行目、3 行目: 変数 data を near 領域に配置しているため、アドレス定数のサイズは near ポインタのサイズと同じで 2 バイトとなります。構造体メンバ ldata のサイズは 4 バイトであるため、発生条件(3)に該当します。

初期化子のアドレス定数は unsigned long 型にキャストしていますが、キャストの有無は発生条件(3)に影響しません。

【出力アセンブラコード】

1	<code>_st1:</code>	
2	<code>.DB4 0x12345678</code>	
3	<code>.DB2 LOWW(_data)</code>	; 4 バイト分の領域が確保されていない
4	<code>.DB4 0x12345678</code>	

整数型メンバの初期値として、アドレス定数のサイズ分（2 バイト分）の領域しか確保されないため、以降のデータをアクセスすると不正な動作となります。

1.5 回避策

次のいずれかにより回避可能です。

- (1) データポインタ、関数ポインタ、および整数型データを取り扱う変数を別管理とする。

1	<code>unsigned char data;</code>
2	<code>struct _st1 {</code>
3	<code> unsigned long ldata;</code>
4	<code>} st1[] = {0x12345678, 0x12345678};</code>
5	<code>struct _st2 {</code>
6	<code> unsigned char * p;</code>
7	<code>} st2[] = {&data};</code>

- (2) 初期値のアドレス定数に記述している変数を `far` 変数にする。^(注1)

1	<code>__far unsigned char data;</code>
2	<code>struct _st {</code>
3	<code> unsigned long ldata;</code>
4	<code>} st[] = {0x12345678, (unsigned long)&data, 0x12345678};</code>

注1：本回避策は、整数型メンバが `long long` 型以外の場合に有効です。
また、`far` 変数を使用することによりコードサイズが大きくなります。

- (3) 発生条件(1)のメンバの初期化子にアドレス定数を記述せず、代入式で初期値を設定する。^(注2)

1	<code>unsigned char data;</code>
2	<code>struct _st {</code>
3	<code> unsigned long ldata;</code>
4	<code>} st[] = {0x12345678, 0, 0x12345678};</code>
5	<code> :</code>
6	<code> :</code>
7	<code> st[1].ldata = (unsigned long)&data;</code>

注2：`const` 変数に対しては、代入式による初期値の設定はできません。

(4) 共用体でポインタをメンバに持つ場合に限り、C99 規格の要素指示子を使用してアドレス定数の初期値を設定する。^(注3)

1	<code>unsigned char data;</code>
2	<code>union _st {</code>
3	<code> unsigned long ldata;</code>
4	<code> unsigned char * p;</code>
5	<code>} st1[] = { 0x12345678, {.p = {&data}}, 0x12345678 };</code>

注3：本回避策は、C99 規格を指定（オプション-lang=c99 指定）している場合に有効です。
 オプション-lang=c99 を指定していない場合は、コンパイル時にエラーになります。
 また、要素指示子で指定したメンバの型で初期値が設定されるため、メンバが near ポインタの場合は、空き領域に上位アドレスが設定されません。

1.6 恒久対策

CC-RL V1.08.00 で改修する予定です。

以上

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2018.12.16	-	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。