RL78開発環境移行ガイド

V850からRL78への移行

（コーディング編）

2017/06/21
R20UT2995JJ0101

ソフトウェア事業部ソフトウエア技術部
ルネサスシステムデザイン株式会社

BIG IDEAS
FOR EVERY SPACE

RENESAS

はじめに

- 本資料は、V850ファミリ用CコンパイラCA850用のプロジェクトやソースコードを、RL78ファミリ用CコンパイラCC-RLへ移行する際のソースコードの差異について、記述しています。

- 本資料では、V850ファミリ用CコンパイラCA850、およびRL78ファミリ用CコンパイラCC-RLを対象に説明しています。

  対象バージョンは以下の通りです。

  - CS+ V5.00.00
  - CA850 V3.50
  - CC-RL V1.04.00

アジェンダ

1．コンパイラ言語仕様

コンパイラ言語仕様

- 言語仕様の相違点(1)

項目	CA850	CC-RL	備考
言語	C言語	C言語	
言語規格	C89	C90, C99の一部機能をサポート	
エンディアン	little	little	
char	符号あり	符号なし	両者とも、オプションで符号あり/なしを切り替え可能
int	4Byte	2Byte	
double	4Byte	・-dbl_size=4 使用時 (デフォルト) 4Byte ・-dbl_size=8 使用時 8Byte	IEEE754-1985に準拠 -dbl_size=8はRL78-S3コアのみ
enum	signed int　(※1)	char signed char unsigned char signed short　　(※2)	※1：-Xenum_typeオプションで変更可能 ※2：列挙値が収まる最小型として処理する
ポインタ型	4Byte	near　2Byte far　4Byte	

コンパイラ言語仕様

- 言語仕様の相違点(2)

項目	CA850	CC-RL	備考
利用可能多バイト文字	EUC, SJIS	EUC, SJIS, UTF-8, big5, gbk	
ビットフィールド	LSBから割り付ける	LSBから割り付ける	
ビットフィールドの符号	符号あり	符号なし (※)	※：-signed_bitfield オプション使用時は符号ありとなる
ビットフィールドのメンバの型	char/short/int/long （符号あり／なし）	char/short/int/long/long long （符号あり／なし）	
境界整列	char，unsigned char 型，およびその配列　： 1バイト境界 short，unsigned short 型，およびその配列： 2バイト境界 その他（ポインタを含む）　： 4バイト境界	char，unsigned char型，signed char型，_Bool型，およびその配列　： 1バイト境界 その他（ポインタを含む）　： 2バイト境界	

コンパイラ言語仕様

- 列挙型指定子の相違点
  列挙子の範囲によって内部表現の型が変わります。
  - CA850の場合
    ・符号付き4Byte整数で扱う
    ・ただし、-Xenum_type=*string* オプションを指定した場合、指定した型で処理します。
    - char　　signed char として扱う
    - uchar　　unsigned char として扱う
    - short　　short として扱う
    - ushort　　unsigned short として扱う
  - CC-RLの場合（優先度順）
    ・-signed_charオプション指定　あり
    範囲：-128～127(0～127の場合も含む)　→　char
    範囲：0～255　→　unsigned char
    範囲：上記以外　→　signed short
    ・-signed_charオプション指定　なし
    範囲：-128～127　→　signed char
    範囲：0～255(0～127の場合も含む)→　char
    範囲：上記以外　→　signed short

コンパイラ言語仕様

- ポインタの相違点
  - CA850の場合
    - ・unsigned int 型と同じサイズ(4Byte)で扱う
  - CC-RLの場合
    - ・near属性の場合は2Byte、far属性の場合は4Byteで扱う
    - ・データや関数のnear/far属性は、オプション、およびキーワード決まります。

	オプション、キーワード	near/far属性の決定方法
(a)	-cpu	デフォルトの near/far 属性を決める
(b)	-memory_model	(a) で決めたデフォルトの near/far 属性を上書きする
(c)	-far_dom	(b) で決めた near/far 属性にROMデータのみ far 属性を上書きする
(d)	__near/__far	(a)～(c) の影響を受けず、__near、__far 記述が有効となる

コンパイラ言語仕様

- メモリモデル指定と変数、関数のnear/far属性

メモリモデル指定	near/far属性
-memory_model=small (スモール・モデル)	変数：　near属性 関数：　near属性
-memory_model=medium (ミディアム・モデル)	変数：　near属性 関数：　far属性

- ROMデータ配置先指定とconstデータのnear/far属性

メモリモデル	near/far属性
-far_romオプション指定時	constデータ：　far属性
-far_romオプション省略時	constデータ：　near属性

コンパイラ言語仕様

- ビットフィールドの相違点
  - CA850の場合
    - ・ビットフィールドのメンバの型は、次の型を使用できます。
      - – char/short/int/long (符号あり/なし)
    - ・signed、unsignedの付かないビットフィールドの符号は、符号付きで扱います。
      - – signed、unsignedの付かない型のビットフィールドの符号は、 -Xbitfieldオプションで変更可能です。
  - CC-RLの場合
    - ・ビットフィールドのメンバの型は、次の型を使用できます。
      - – char/short/int/long/long long (符号あり/なし)
    - ・signed、unsignedの付かないビットフィールドの符号は、符号なしで扱います。
      - – signed、unsignedの付かない型のビットフィールドの符号は、 -signed_bitfieldオプションで変更可能です。

2. 拡張機能

拡張機能

- 拡張機能(#pragma)、キーワードの相違点(1)

項目	CA850	CC-RL	備考
セクションの切り替え指定	#pragma section #pragma text	#pragma section	CC-RLでは、変数、関数セクションともに、#pragma section で指定してください。
短いアクセス命令セクションへの切り替え指定	#pragma section tidata	#pragma saddr __saddr	
周辺I/O レジスタ、SFRへのアクセス	#pragma ioreg	#include “iodefine.h”	CC-RLでは、ビットアクセスを含むSFRへの参照用に、iodefine.hで記号定数を定義しています。 iodefineのインクルードは手動で指定してください。
関数のインライン展開	#pragma inline	#pragma inline __inline	
アセンブラ埋め込み機能(1)	__asm( ); _asm( );	#pragma inline_asm	CC-RLでは、__asm( )/_asm( )を通常の関数呼出として扱います。
アセンブラ埋め込み機能(2)	#pragma asm #pragma endasm	なし	

拡張機能

- 拡張機能(#pragma)、キーワードの相違点(2)

項目	CA850	CC-RL	備考
構造体パッキング	#pragma pack([1][2][4][8]) -Xbyte -Xpack	-pack	CC-RLでは、オプションでのみ指定可能
割り込み関数の作成	#pragma interrupt __interrupt __multi_interrupt	#pragma interrupt #pragma interrupt_brk	RL78ファミリでは、__multi_interruptの機能必要なし
割り込みレベルの制御	__set_il(,);	なし	CC-RLでは、関数を用意していないので割り込み優先度を変更するにはSFRを操作する必要がある。
割り込み禁止関数指定	#pragma block_interrupt	なし	

拡張機能

- 拡張機能(#pragma)、キーワードの相違点(3)

項目	CA850	CC-RL	備考
リアルタイムOS対応機能	#pragma rtos_task	#pragma rtos_task #pragma rtos_interrupt	
セクションアドレスの取得	__s __e	__sectop __secend	CC-RLでは、セクションの先頭アドレス / 末尾のアドレス +1 を参照。 CA850では、セクション名の先頭に文字列を追加したシンボルを生成。

拡張機能

- セクション指定の相違点
  - CA850の場合
    - 変数のセクション指定
      #pragma section セクション種別　“作成するセクション名”　begin
      …
      #pragma section セクション種別　“作成するセクション名”　end
    - 関数のセクション指定
      #pragma text “作成するセクション名” 関数名
      または
      #pragma text “作成するセクション名”
  - CC-RLの場合
    - #pragma section セクション種別　変更セクション名
      – セクション種別：{text | const | data | bss }

拡張機能

- 割り込み関数の相違点
  - CA850の場合
    - ・割り込み/例外ハンドラの指定：
      #pragma interrupt　割り込み要求名 関数名 [配置方法]
      __interrupt 関数定義、または関数宣言
    - ・多重割り込み/例外ハンドラを指定：
      #pragma interrupt　割り込み要求名 関数名 [配置方法]
      __multi_interrupt 関数定義、または関数宣言
  - CC-RLの場合
    - ・ RL78ファミリでは、割り込み時のPC、PSWはスタックに退避されるため、通常割り込み、多重割り込みの区別はありません。
    - ・ハードウェア割り込みハンドラを指定：
      #pragma interrupt　割り込みハンドラ名 (割り込み仕様)
    - ・ソフトウェア割り込みハンドラを指定：
      #pragma interrupt_brk　割り込みハンドラ名 (割り込み仕様)

3．組み込み関数

組み込み関数

- CA850とCC-RLの組み込み関数の相違点(1)

項目	CA850	CC-RL	備考
飽和演算	__satadd	なし	
飽和減算	__satsub	なし	
ハーフワード・データのバイト・スワップ	__bsh	なし	
ワード・データのバイト・スワップ	__bsw	なし	
ワード・データのハーフワード・スワップ	__hsw	なし	
バイト・データの符号拡張	__sxb	なし	
ハーフワード・データの符号拡張	__sxh	なし	
mul 命令を用いて乗算結果の上位32ビットを変数に代入する命令	__mul32	なし	
mulu 命令を用いて符号なし乗算結果の上位32 ビットを変数に代入する命令	__mul32u	なし	
論理左シフト付きフラグ条件の設定	__sasf	なし	

4．マクロ

マクロ

- CA850とCC-RLのマクロの相違点(1)

CA850	CC-RL	備考
__CA850 __CA850__	__CCRL __CCRL__	コンパイラ名称のマクロ
__v800 __v800__ __v850 __v850__ __v850e __v850e__ __v850e2 __v850e2__	__RL78__ __RL78_S1__ __RL78_S2__ __RL78_S3__ __RL78_SMALL__ __RL78_MEDIUM__	対象コアのマクロ
CPUマクロ名 「品種指定名」で示される文字列の先頭と末尾に“__”を付けたもの	なし	対象CPUのマクロ

マクロ

- CA850とCC-RLのマクロの相違点(2)

CA850	CC-RL	備考
__DATE__ __TIME__	__DATE__ __TIME__	
__FILE__ __LINE__	__FILE__ __LINE__	
__STDC__	__STDC__	
__CHAR_SIGNED__	__SCHAR	
__CHAR_UNSIGNED__	__UCHAR	
__reg32__ __reg26__ __reg22__	__RL78_SMALL__ __RL78_MEDIUM__	コンパイラモードのマクロ CA850：-regオプション CC-RL：-memory_modelオプション

参考資料

- 詳細は、以下のユーザーズマニュアルを参照し変更してください

- CC-RLコンパイラ　ユーザーズマニュアル

https://www.renesas.com/search/keyword-search.html#genre=document&q=r20ut3123

- CubeSuite+ V1.00.00 統合開発環境 ユーザーズマニュアル V850 コーディング編

https://www.renesas.com/search/keyword-search.html#genre=document&q=r20ut0553

BIG IDEAS FOR EVERY SPACE

ルネサス システムデザイン株式会社