

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日

ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

RENESAS TECHNICAL UPDATE

〒100-0004 東京都千代田区大手町 2-6-2 日本ビル
株式会社 ルネサス テクノロジ
問合せ窓口 E-mail: csc@renesas.com

製品分類	開発環境	発行番号	TN-CSX-A089A/J	Rev.	第1版
題名	Renesas C/C++ Compiler Package for SuperH RISC engine family V.9.00 Release 01 へのリビジョンアップのお知らせ		情報分類	技術情報	
適用製品	ROC40700XSW09R ROC40700XSS09R ROC40700XSH09R	対象ロット等	関連資料	SuperH RISC engine C/C++コンパイラ、アセンブラ、最適化リンケージエディタコンパイラパッケージ V.9.00 ユーザーズマニュアル (RJJ10B0156-0101 Rev.1.01)	
	V.9.00				

SuperH RISC engine ファミリー用 C/C++コンパイラパッケージを V.9.00 Release 00 から V.9.00 Release 01 にリビジョンアップしました。

1. リビジョンアップ内容

1.1 新機能および機能改善

1.1.1 シミュレータ (Windows 版のみ)

- (1) SH-2 シミュレータでリトルエンディアン方式のデータ転送機能をサポートしました。
- (2) プログラムをシミュレータにダウンロードする際に、メモリリソースを自動で確保する機能を追加しました。
- (3) メモリアクセスエラー発生時に、メモリアクセスエラーが発生したアドレスを表示する機能を追加しました。

1.1.2 コンパイラ

- (1) 以下のオプションを追加しました。

- (a) 未初期化変数のメモリ割り当て順の指定

```
bss_order={declaration | definition}
```

- (b) 変数の配置指定

```
stuff[={bss | data | const}{,...}], および nostuff
```

オプションの詳細については、「コンパイラ、アセンブラ、最適化リンケージエディタ コンパイラパッケージ V.9.00 ユーザーズマニュアル Rev.1.01」を参照してください。

- (2) SH-2 のリトルエンディアン方式によるデータ転送機能をサポートしました。ただし、リトルエンディアンをサポートしていないマイコンでは使用できません。

1.2 改修内容

1.2.1 High-performance Embedded Workshop (Windows 版のみ)

ELF/DWARF2 フォーマットで作成されたロードモジュールをデバッグすると HEW が異常終了する場合がある不具合を改修しました。

詳細は 2004 年 11 月 1 日発行の RENESAS TOOL NEWS "HEW ご使用上のお願い--ELF/DWARF2 ロードモジュールのアンロードについて"を参照してください。

<http://tool-support.renesas.com/jpn/toolnews/n041101/tn5.htm>

1.2.2 シミュレータ (Windows 版のみ)

SH2A-FPU Cycle Base Simulator で RESBANK 命令を実行するとサイクル数が不正になる問題を改修しました。

詳細は 2004 年 11 月 1 日発行の RENESAS TOOL NEWS 「SuperH RISC engine ファミリ C/C++コンパイラパッケージ V.9.00 Release 00 ご使用上のお願い」を参照してください。

<http://tool-support.renesas.com/jpn/toolnews/n041101/tn7.htm>

1.2.3 コンパイラ

以降 12 点の問題を改修しました。

- (1) 以下の DSP ライブラリ関数を CPU コア SH4AL-DSP 用のプログラムで使用した場合に正しく動作しない場合がある。

FftComplex, FftReal, IfftComplex, IfftReal, FftInComplex,
 FftInReal, IfftInComplex, IfftInReal, Fir, ConvComplete,
 ConvCyclic, ConvPartial, Correlate, CorrCyclic, MatrixMult,
 MsPower, および Variance

本改修により、CPU コア SH4AL-DSP 用のライブラリファイル名が以下の通り変更になります。

指定オプション	ライブラリファイル
-pic=0 -endian=big	sh4aldspnb.lib
-pic=1 -endian=big	sh4aldsppb.lib
-pic=0 -endian=little	sh4aldspnl.lib
-pic=1 -endian=little	sh4aldsppl.lib

- (2) 条件式を含むループにおいて、ループが不正に実行される場合がある (SHC-0008)。

例 1

```

-----
int b[100];
unsigned int c=0;
void func1() {
    unsigned int i;
    for(i=0;i<=100;i++) { // 無限ループになる
        if(i != c) {
            b[i]=0;
        }
    }
}
-----

```

例 2

```

-----
int b[100];
unsigned int c=0;
void func2() {
    unsigned int i;

```

```

for(i=0;i<c;i++) {      // c=0 の時もループを1回以上まわる
    if(i != 5) {
        b[i]=0;
    }
}

```

発生条件：以下の条件をすべて満たした場合に発生することがあります。

1. optimize=1 オプションを使用している。
 2. ループが存在する。
 3. 2 のループ制御変数の増分値が 1 または-1 である。
 4. 2 のループ内に if 文が存在する。
 5. 以下 5a、または 5b のいずれかの条件を満たす。
 - 5a. 4 の if 文の条件式が 2.のループ制御変数と、2.のループ内で値が不変の式(例 1 の変数 c)の比較である。
 - 5b. 2 のループ制御変数の初期値または上限値がループ内で値が不変の式(例 2 の変数 c)である。
 6. 5 のループ内不変式が整数型である。
- (3) do-while ループで、正しくは 1 回で終了するものが 2 回以上ループする場合がある(SHC-0010)。

例：

```

int func() {
int count=0;
int limit=0x60000000;
do {
    count++;
    limit += 0x10000000;
} while(limit < -0x60000000); // 正しい動作では 1 回目のループ後の
// 判定で偽となり、ループを抜ける。
return (count); // count=1 が正しいが、違う値になる。
}

```

発生条件：以下の条件をすべて満たした場合に発生することがあります。

1. optimize=1 オプションを使用している。
2. do-while ループが存在する。
3. 2 のループ制御変数の型が int, signed int, long, または signed long である。
4. 2 のループ判定式が、定数値との大小比較(<,<=,>,>=)である。
5. 2 のループの比較演算子、制御変数の初期値、更新値、および判定式の比較値が以下 5a ~ 5d のいずれかの条件を満たす。
 - 5a. 判定式が、ループ制御変数<定数 の場合に以下全てに該当する。
 - ・更新値が正
 - ・比較値<=初期値
 - ・0x00000000<=(比較値-初期値-1)<=0x7FFFFFFF

5 b. 判定式が、ループ制御変数<=定数 の場合に以下全てに該当する。

- ・更新値が正
- ・比較値<初期値
- ・ $0x00000000 \leq (\text{比較値} - \text{初期値} - 1) \leq 0x7FFFFFFF$

5 c. 判定式が、ループ制御変数>定数 の場合に以下全てに該当する。

- ・更新値が負
- ・初期値<=比較値
- ・ $0x00000000 \leq (\text{初期値} - \text{比較値} - 1) \leq 0x7FFFFFFF$

5 d. 判定式が、ループ制御変数>=定数 の場合に以下全てに該当する。

- ・更新値が負
- ・初期値<比較値
- ・ $0x00000000 \leq (\text{初期値} - \text{比較値} - 1) \leq 0x7FFFFFFF$

(4) 1 ビットの符号付きビットフィールドと 1 との比較を行ったとき、またはある比較結果に対して演算を行いその結果を 1 と比較したときに、比較結果が間違っている場合がある(SHC-0011)。

例 1 :

```

-----
struct {
    char b0:1;
} ST;
void func() {
    if (ST.b0 != 1) {
        . . . . .
    }
}
-----

```

例 2 :

```

-----
int a;
void func2() {
    int t;
    t = ((a & 0x40) == 0);
    t = t - 1;
    t = -t;
    if(~t==1) {
        a = 1;
    } else {
        a = 2;
    }
}
-----

```

発生条件：以下 1、2 のいずれかの条件を満たした場合に発生することがあります。

- 1. 以下のすべての条件を満たした場合。
 - 1a. optimize=1 を指定している。
 - 1b. 1 ビットの符号付きビットフィールドを使用している。
 - 1c. 1b.と1の比較(==,!)=)を行っている。

- 2. 以下のすべての条件を満たした場合。
 - 2a. optimize=1 を指定している。
 - 2b. 比較結果に対して以下のいずれかの演算を行っている。
比較結果と1の減算、比較結果と1の排他的論理和、
比較結果の符号反転、または比較結果のビット反転
 - 2c. 2b.の演算結果と1の比較(==,!)=)を行っている。

(5) 0 との加算および減算、または1との乗算をした結果を他の演算で使用したときに、変数の値を誤って更新する場合があります(SHC-0012)。

例：

```

-----
int a[4], b;
void func() {
    a[3 & (b-0)]=0;
}
-----
    
```

発生条件：以下の条件をすべて満たした場合に発生することがあります。

- 1. 変数と0の加算および減算、または変数と1の乗算をしている。
- 2. 1の結果を以下の演算で使用している。
加算、減算、論理演算(&,|,^)、除算、剰余算、またはシフト

(6) pack=1 を指定した構造体または共用体の double 型メンバを参照したときに、レジスタ R2 の値を誤って変更する場合があります(SHC-0013)。

例：

```

-----
#pragma pack 1
struct {
    double d;
} ST;
int t;
double d[2];
void func() {
    d[t]=ST.d;
}
-----
    
```

発生条件：以下の条件をすべて満たした場合に発生することがあります。

- 1. pack=1 を指定した構造体または共用体がある。
- 2. 1.の構造体または共用体に double 型のメンバが存在する。
- 3. cpu=sh4,sh4a,sh2afpu を指定している。

4. size または unaligned=runtime オプションを使用している。

5. 実行時ルーチン_pack1_ld64 の呼び出しがある。

(7) 定数乗算および定数除算を共に含む式において、演算結果が間違っている場合がある(SHC-0015)。

例：

```
-----
unsigned int a=65536;
unsigned int b;
void func() {
    b=(a*65536)/8; // b=0 ((65536*65536)/8 0/8=0) が正しい
}
// 結果だが、b=65535<<13 と置き換えられる。
-----
```

発生条件：以下の条件をすべて満たした場合に発生することがあります。

1. unsigned 型の式と定数の乗算式が存在する。
2. 1 を含む式を 1 の定数の正の約数で除算している。
3. 1 の乗算結果がその式の型の最大値を超える。

(8) 被シフト数のビットサイズ未満の定数シフトを複数回行い、そのシフト数の合計が被シフト数のビットサイズ以上になった場合、演算結果が間違っている場合がある(SHC-0016)。

例：

```
-----
int x,y;
void func() {
    x=y<<31<<1; // シフト数の合計は 32 で、int 型のビットサイズ
                // 以上になる
}
-----
```

発生条件：以下の条件をすべて満たした場合に発生することがあります。

1. cpu オプションのパラメータに sh1, sh2, sh2e, または sh2dsp 以外を指定している。
2. 以下 2a、2b のいずれかの条件を満たす。
 - 2a. 左シフトもしくは 2 のべき乗の乗算が 2 回以上連続し、かつ個々のシフト数、乗数は被シフト数のビットサイズ未満である。
 - 2b. 右シフトもしくは 2 のべき乗の除算が 2 回以上連続し、かつ個々のシフト数、乗数は被シフト数のビットサイズ未満である。
3. 2 のシフト数と乗数の値の合計が被シフト数のビットサイズ以上となる。

(9) 関数の最後が関数呼び出しの場合、呼び出し先関数のアドレスをレジスタ R0 に誤って書き込む場合がある。

例：

```
-----
char a[3];
void func2() {}
#pragma section A
void func() {
    ++a[2];
}
-----
```

```
func2();
}
```

発生条件：以下の条件をすべて満たした場合に発生することがあります。

1. 呼び出し元関数の最後で関数を呼び出している。
 2. 呼び出し先関数の定義が 1 と同一ファイル内にある。
 3. 2 の関数定義と 1 の関数呼び出しは別セクションにある、もしくは 4096 バイト以上離れた位置にある。
 4. 呼び出し元関数内に 1 以外の関数呼び出しがない。
- (10) ポインタを介してビットフィールドメンバに対して演算を行った場合、その演算結果が間違っている場合がある (SHC-0023)。

例：

```
typedef union {
    unsigned int a;
    unsigned int b:32;
} UN;
UN un;
void func() {
    int *p=(int *)&un.a;
    un.b=1;
    *p+=1;
}
```

発生条件：以下の条件をすべて満たした場合に発生することがあります。

1. 構造体または共用体型を使用している。
 2. 1. のメンバを直接参照(例では un.b)している。
 3. 2. と同じ領域のメンバ(例では un.a)を指すポインタ変数(例では p)が存在する。
 4. 3. のポインタはローカル変数である。
 5. 3. のポインタの間接参照(例では *p)が存在する。
 6. 当該領域の値を更新する。
 7. 構造体または共用体自身の参照を関数内で行っていない。
- (11) 構造体または共用体コピーを行った場合、スタック領域が必要以上に確保される場合がある (SHC-0021)。

例：

```
typedef struct {
    unsigned char c;
} ST0;
typedef struct {
    ST0 s;
} ST;
extern ST A[1000];
```



```
extern unsigned short i;
void func(ST *d) {
    A[i-1].s=d->s; // スタックが A[1000]分余分に確保される
}
-----
```

発生条件：以下の条件をすべて満たした場合に発生することがあります。

1. 構造体または共用体型のメンバを持つ構造体または共用体配列が存在する。
2. 1 の配列の要素数は 2 以上である。
3. 構造体または共用体代入文が存在する。
4. 3 の左辺は 1 の構造体または共用体配列の構造体または共用体型メンバである。

(12) C4098、および C4099 インターナルエラーが発生する場合がある(SHC-0007, SHC-0017, SHC-0022, SHC-0025)。

1.2.4 アセンブラ

CPU シリーズ SH-2A、または SH2A-FPU 用のプログラムを作成する場合に発生する以降 3 点の問題を改修しました。

(1) 拡張命令や分岐命令(PC 相対)のディスプレイメント値が正しくない場合がある。

例：

```
-----
.ALIGN 4
NOP
MOV.L @(AA,R1),R0
BRA LABEL
.ALIGN 4
BT LABEL
.....
LABEL: NOP
AA: .EQU 4
-----
```

発生条件：以下の条件をすべて満たした場合に発生することがあります。

1. cpu=sh2a または cpu=sh2afpu オプションを指定している。
2. ある遅延分岐命令の前に " MOV @(disp,Rn) " (disp:12 が指定可であるが指定されていない命令)が奇数個存在する。
3. 2 の遅延分岐命令の次が.ALIGN 制御命令または.ORG 制御命令である。
4. 3 により 2 バイトのアドレス補正が行われる。
5. 3 の次が拡張命令、または分岐命令(PC 相対)である。

(2) アドレス分岐の分岐先が間違っている場合がある。

例：

```
-----
.SECTION SEC1, CODE
MOV.L @(AA,R1),R0 ; 2 バイト命令
NOP
BRA L3
-----
```

```

        .ALIGN    4           ; ロケーションカウンタ 補正あり
        BF      L1           ; ディレイスロット命令でない
L3:    NOP
        .SECTION SEC2, CODE
L1:    NOP
AA:    .EQU      4
-----

```

発生条件：以下の条件をすべて満たした場合に発生することがあります。

1. cpu=sh2a または cpu=sh2afpu オプションを指定している。
 2. ある遅延分岐命令の前に " MOV @(disp,Rn) " (disp:12 が指定可であるが指定されていない命令)が奇数個存在する。
 3. 2 の遅延分岐命令の次が .ALIGN 制御命令または .ORG 制御命令である。
 4. 3 により MOV @(disp,Rn)命令数×2 バイトの補正がある。
 5. 3 の次がアドレス分岐命令である。
- (3) 偶奇演算子\$even2, および\$oadd2 を使用した演算の結果が正しくない場合がある。

1.2.5 最適化リンケージエディタ

以降 5 点の問題を改修しました。

- (1) 共通コード統合最適化(optimize=same_code)が有効な場合、生成されたデバッグ情報を使ってデバッガでステップ実行を行うと間違った関数へ分岐する場合がある。
- (2) .stack 制御命令を、定義シンボルには記述せずに参照シンボルのみに指定した場合に、その情報が sni ファイルに出力されない。
- (3) 共通コード統合最適化が有効な場合、リロケータブルファイルを入力すると間違ったコードが生成される場合がある。
発生条件：以下の条件をすべて満たす場合、発生する可能性があります。
 1. コンパイル時に goptimize オプションを指定している。
 2. 共通コード統合最適化(optimize=same_code)が有効である。
 3. 最適化リンケージエディタ V.8.00.03 以降のバージョンを使用している。
 4. リンク時に、リロケータブルファイルを入力している。
 5. 4 のリロケータブルファイルのコードが、2 の最適化により 4n+2 バイト(4 の倍数でないサイズ)で共通化される。
- (4) 定数および文字列統合最適化が有効な場合、不適切なエラー(L2330)が出力される場合がある。
発生条件：以下の条件をすべて満たす場合、発生する可能性があります。
 1. コンパイル時に goptimize オプションを指定している。
 2. 定数および文字列統合最適化(optimize=string_unify)が有効である。
 3. コンパイル時に abs16 オプションを指定している、または#pragma abs16 宣言されている const 指定変数がある。
- (5) 以下に該当する場合に内部エラーが発生する。
 1. 未参照シンボル削除最適化(optimize=symbol_delete)が有効な場合に出力ファイルを output オプションを使用して複数ファイルに分割するように指定している(内部エラー(7041)発生)。
 2. profile オプション使用時に、binary オプションを指定している(内部エラー(L4001)発生)。

2. リビジョンアップと購入方法

2.1 リビジョンアップ（無償）

該当製品をお持ちの場合、無償でリビジョンアップできます。

(1) Windows 版: R0C40700XSW09R の場合

オンラインでリビジョンアップできます。

以下の URL からダウンロードしてください。

<http://www.renesas.com/jpn/products/mpumcu/tool/download/crosstool/sh/v9/shcv90001.html>

(2) R0C40700XSS09R(Solaris 版) および R0C40700XSH09R(HP-UX 版)の場合

以下の情報を最寄りのルネサス販売または特約店までご連絡ください。

製品型名：

Solaris 版: R0C40700XSS09R

HP-UX 版: R0C40700XSH09R

バージョン番号：V.9.00

リリース番号：Release 01

2.2 新規購入

ご注文の際には、以下の情報を最寄りのルネサス販売または特約店までご連絡ください。

製品型名：

Windows 版: R0C40700XSW09R

Solaris 版: R0C40700XSS09R

HP-UX 版: R0C40700XSH09R

バージョン番号：V.9.00

リリース番号：Release 01

以上