

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>

# 日立半導体技術情報

〒100-0004  
 東京都千代田区大手町2丁目6番2号  
 (日本ビル)  
 TEL (03)5201-5022 (ダイヤルイン)  
 株式会社 日立製作所 半導体グループ

製品分類	開発環境		発行番号	TN-CSX-041A	Rev.	第1版
題名	SuperH RISC engine C/C++ コンパイラパッケージ Ver.6.0、Ver.6.0A、Ver.6.0B 不具合対策版リリースのお知らせ		情報分類	①. 仕様変更 2. ドキュメント訂正追加等 3. 使用上の注意事項 4. マスク変更 5. ライン変更		
適用製品	P0700CAS6-MWR	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ユーザズマニュアル ADJ-702-304A 第1版	有効期限	
	P0700CAS6-SLR P0700CAS6-H7R	全ロット			永年	

SuperH RISC engine C/C++コンパイラパッケージを Ver. 6.0C にリビジョンアップしました。  
 次に示す製品を御使用のお客様につきましては、周知願います。

Windows 版 SuperH RISC engine C/C++コンパイラパッケージ(型名:P0700CAS6-MWR)  
 SPARC版 SuperH RISC engine C/C++コンパイラパッケージ(型名:P0700CAS6-SLR)  
 HP9000版 SuperH RISC engine C/C++コンパイラパッケージ(型名:P0700CAS6-H7R)

の Ver.6.0, Ver.6.0R1, Ver.6.0A, Ver.6.0AR1, Ver.6.0AR2, Ver.6.0B

添付:

(1) SuperH RISC engine C/C++コンパイラパッケージ V6.0C アップデート内容  
 (P0700CAS6-020822) 5枚

## SuperH RISC engine C/C++ compiler Package Ver. 6.0C アップデート内容

### 1.コンパイラ (Ver.6.0B -> Ver.6.0C)

#### 1.1 レジスタの回復順不正

##### 【内容】

CPU=SH2E または SH4 指定時、関数の末尾で他の関数を呼び出しているとき、レジスタの回復順が不正になる不具合を解決しました。

##### 【発生条件】

以下の条件をすべて満たす場合、発生することがあります。

- (1) optimize=1 を指定している。
- (2) CPU に SH2E/SH4 のいずれかを指定している。
- (3) 関数の末尾で他の関数を呼び出している。
- (4) 浮動小数点レジスタの退避/回復コードがある。
- (5) 当該関数内に上記(3)以外の関数(実行時ルーチンを含む)呼び出しがある。

##### 【対象バージョン】

Ver.6.0B

#### 1.2 pic=1 指定時実行結果不正

##### 【内容】

pic=1 指定時に外部定義関数を呼び出すプログラムを実行すると実行結果不正になる不具合を解決しました。

##### 【発生条件】

以下の条件をすべて満たす場合、発生することがあります。

- (1) CPU に SH1 以外を指定している。
- (2) pic=1 を指定している。
- (3) プログラム上で外部定義関数が呼び出されている。

##### 【対象バージョン】

Ver.6.0B

### 1.3 PR レジスタ割付不正

**【内容】**

cpu=SH2E または SH4 指定時、関数のレジスタ退避/回復命令が PR レジスタと FPU レジスタのみの場合、オブジェクト不正になる不具合を解決しました。

**【発生条件】**

以下の条件をすべて満たす場合、発生することがあります。

- (1) optimize=1 を指定している。
- (2) CPU に SH2E/SH4 のいずれかを指定している。
- (3) 退避/回復するレジスタが PR 及び FPU レジスタのみである。
- (4) 当該関数を #pragma interrupt 指定していない。

**【対象バージョン】**

Ver.6.0B

### 1.4 コードエラー

**【内容】**

コメント内に 1byte 目が '0x8e' で 2byte 目が '0xa1' 以上 '0xdf' 以下のコードがあるソースを listfile オプションを指定してコンパイルすると、6001 ~ 6007 のインターナルエラーになる不具合を解決しました。

**【対象バージョン】**

Ver.6.0, Ver.6.0A, Ver6.0B

### 1.5 文字列不正エラー

**【内容】**

以下のソースを英語 OS 上でかつ C++ でコンパイルした際に不当にエラー 2138 を出力する不具合を解決しました。

**【ソース】**

```
char b[] = "¥xec""¥x01";
```

**【対象バージョン】**

Ver.6.0, Ver.6.0A, Ver6.0B

## 1.6 セクション割付不正

### 【内容】

以下のように宣言の後にセクションを変更した定義が存在すると、セクションが切り替わらない不具合を解決しました。

### 【ソース】

```
extern int a;
```

```
#pragma section OTHER
```

```
int a = 1;
```

### 【対象バージョン】

Ver.6.0, Ver.6.0A, Ver.6.0B

## 1.7 FDIV チップバグ対策

### 【内容】

SH7055RF FPU FDIV 関連不具合(TN-SH7-402A)を対策しました。

cpu=SH2E、extra=a=2000 オプション指定時に上記チップ不具合を回避したオブジェクトを出力します。

### 【対象バージョン】

Ver.6.0, Ver.6.0A, Ver.6.0B

## 2. 最適化リンケージエディタ (Ver. 7.0A -> Ver.7.1.06)

### 2.1 rename 指定時のデバッグ情報不正

#### 【内容】

form=relocate 指定時に rename を指定すると、当該セクション内シンボルのデバッグ情報が削除される不具合を解決しました。

#### 【発生条件】

下記条件を全て満たす時、発生します。

(1) form=relocate を指定している。

(2) rename を指定している。

## 2.2 const、リテラルデータ最適化不正

### 【内容】

`optimize=string_unify` を指定すると不正にシンボルの統合最適化を実施する不具合を解決しました。

### 【発生条件】

下記条件を全て満たす時、発生する可能性があります。

- (1) C ソースファイルを `goptimize` オプション指定でコンパイルする。
- (2) 最適化リンカで、`optimize=string_unify` を指定する。

## 2.3 binary ファイル入力指定時のセクション属性不正

下記条件全てを満たした際に、セクションの属性が不正となる不具合を解決しました。

- (1) `object` ファイルと `binary` ファイルを入力
- (2) `object` ファイル内でサイズ 0 のセクションを定義
- (3) `binary` オプションでサイズ 0 のセクションを指定
- (4) ファイルの入力指定順：(2)の `object` ファイル -> (3)の `binary` ファイル

## 2.4 relocate ファイル生成時の異常終了

下記条件全てを満たした際に異常終了する不具合を解決しました。

- (1) 入力 `object` ファイルの先頭：  
`goptimize` オプション指定ありのファイル  
入力 `object` ファイルの 2 番目以降の入力：  
`goptimize` オプション指定なしのファイルもしくはアセンブラ出力ファイル
- (2) 出力形式に `relocate` ファイルを指定
- (3) `profile` オプション指定あり
- (4) `optimize` オプション指定あり

## 2.5 複数関数が同一リテラルを指す場合のリテラル参照不正

下記条件全てを満たした際に参照するリテラルの値が不正となる不具合を解決しました。

- (1) `goptimize` オプション指定ありの `object` ファイルを入力
- (2) (1)の `object` ファイル内で、複数関数が同一リテラルを参照
- (3) レジスタ退避/回復コード最適化を指定

## 2.6 最適化抑止指定時の不当な最適化

下記条件を全て満たした際に、最適化抑止範囲が有効にならない不具合を解決しました。

- (1) `goptimize` オプション指定ありのファイルを入力
- (2) `optlnk` で最適化を指定
- (3) `absolute_forbid` オプションを指定
- (4) 最適化抑止範囲を二つ以上記述(二つ目以降の指定が無効となる)

## 2.7 relocate ファイル生成時の Object 不正

下記条件を全て満たした際に不正な Object コードが生成される不具合を解決しました。

- (1) 入力に `relocate` ファイルを指定
- (2) 出力に `relocate` ファイルを指定
- (3) `delete`,もしくは `rename` オプションを指定

## 2.8 レジスタ退避/回復最適化指定時の Object 不正

レジスタ退避/回復コード最適化を指定した際、下記条件のいずれかを満たす場合、不正な Object コードが出力される場合がある不具合を修正しました。

- (1) ファイル内の最後の関数にリテラルが存在しない。
- (2) レジスタ回復コード列の直後に分岐先が存在する。
- (3) 最適化対象関数に `SUBC` 命令が含まれており、その `SUBC` 命令自体がレジスタ番号の変換対象である。

## 2.9 Internal error 解決

以下の Internal error となる問題点を解決しました。

(1703),(1704),(3304),(7041),(7707)

## 3. フォーマットコンバータ (1.0B -> 1.0.04)

### 3.1 圧縮フォルダ内のファイルの入力

圧縮フォルダ内のファイルを入力しようとした場合、ファイルを認識できない (G3002 エラー)不具合を解決しました。