

この度は、統合開発環境 CubeSuite+をご使用いただきまして、誠にありがとうございます。

この添付資料では、本製品をお使いいただく上での制限事項および注意事項等を記載しております。ご使用の前に、必ずお読みくださいますようお願い申し上げます。

目次

第 1 章	対象デバイスについて.....	2
第 2 章	ユーザーズ・マニュアルについて.....	3
第 3 章	アンインストール時の選択キーワード.....	4
第 4 章	変更点.....	5
4.1	CA850 の変更点.....	5
4.1.1	CubeSuite+ 対応.....	5
第 5 章	注意事項.....	6
5.1	デバッグ情報出力オプション (-G) 指定.....	6
5.2	割り込み関数におけるR1 レジスタの取り扱い.....	6
5.3	デバッグ情報.....	6
第 6 章	制限事項.....	7
6.1	CA850 の制限事項一覧.....	7
6.2	CA850 の制限事項詳細.....	8

第1章 対象デバイスについて

CA850 がサポートする対象デバイスに関しては、WEB サイトに掲載しています。

こちらをご覧ください。

CubeSuite+製品ページ：

<http://japan.renesas.com/cubesuite+>

第2章 ユーザーズ・マニュアルについて

本製品に対応したユーザーズ・マニュアルは、次のようになります。本文書と合わせてお読みください。

マニュアル名	資料番号
CubeSuite+ V1.00.00 V850 コーディング編	R20UT0553JJ0100
CubeSuite+ V1.00.00 V850 ビルド編	R20UT0557JJ0100
CubeSuite+ V1.00.00 メッセージ編	R20UT0407JJ0100

第3章 アンインストール時の選択キーワード

本製品をアンインストールする場合は、2つの方法があります。

- ・ 統合アンインストーラを使用する(CubeSuite+自体をアンインストールする)
- ・ 個別にアンインストールする(本製品のみをアンインストールする)

個別にアンインストールを行なう場合、コントロールパネルの

- ・ 「プログラムの追加と削除」(WindowsXP の場合)
- ・ 「プログラムと機能」(Windows Vista, Windows 7 の場合)

から、「CubeSuite+ CA850 V3.50」を選択してください。

第4章 変更点

本章では、CA850 の変更点について説明します。

4.1 CA850 の変更点

CA850 V3.47 から V3.50 への変更点を説明します。

4.1.1 CubeSuite+ 対応

CubeSuite+ に対応しました。

第5章 注意事項

本章では、CA850 V3.50 の注意事項について説明します。

5.1 デバッグ情報出力オプション (-g) 指定

デバッグ情報出力オプション -g を指定した場合と指定しない場合で、命令並びやサイズが異なる可能性があります。

5.2 割り込み関数におけるr1レジスタの取り扱い

アセンブラ (as850) は、r1 レジスタを命令展開を行う際のテンポラリ・レジスタとして使用します。そのため、アセンブラ・ソース・ファイルに r1 レジスタの記述がなくても、命令展開により r1 レジスタが使用されている可能性があります。

アセンブラでの割り込み関数を記述する場合には、r1 レジスタの退避／復帰を行ってください。

5.3 デバッグ情報

関数の定義で、開始もしくは終了が #include など別のソース・ファイルで記述されている場合、正しいデバッグ情報が得られません。

また、アセンブラ・ソース・ファイルで .include / .binclude 擬似命令で指定したファイル内のコード、.macro 擬似命令で定義されたマクロ内のコード、.section 擬似命令を用いて作成したセクションに対してはデバッグ情報が出力されません。

第6章 制限事項

本章では、CA850 の制限事項について説明します。

6.1 CA850 の制限事項一覧

CA850 V3.50 における制限事項一覧を以下に示します。

No.	制限事項
4	浮動小数点定数演算の演算時精度の制限
6	関数の引数で構造体型の条件演算子の制限
7	関数の間接呼び出しの制限
8	意味不明な関数定義がエラーにならない制限
12	関数宣言で () が余計に付いている場合エラーになる制限
21	セクション配置の制限
22	変数の実体のアセンブリ・ソースにあるセクション・ファイルの制限
23	同名の外部変数の仮定義が複数ファイルにあるセクション・ファイルの制限
29	最適化オプション指定時のエラーの制限
30	最適化時のオブジェクト・サイズの制限
31	最適化時のデバッグの制限
33	構造体メンバのアドレスの制限
34	ビット・フィールドの制限
40	C ソース内における、固有シンボル、予約シンボルの参照の制限
95	浮動小数点定数と汎整数型の制限
96	入出力関数の標準ライブラリの入力変換の制限

6.2 CA850の制限事項詳細

CA850 V3.50 における制限事項の詳細は以下のとおりです。

No. 4 浮動小数点定数演算の演算時精度の制限

【内容】 整数へのキャストをともなうコンパイルにおいて、演算ができてしまうような浮動小数点演算を記述した場合、極微少の精度ずれが発生し、整数へのキャストにより不正な値になることがあります。なお、浮動小数点のまま扱う場合、問題は発生しません。

(例) `(long)(1.12 * 100);`

【回避策】 下記のいずれかのように変更してください。

```
float f=1.12;
(long)(f * 100);
または、
float f;
(long)(f=1.12 * 100);
```

No. 6 関数の引数で構造体型の条件演算子の制限

【内容】 引数に構造体型の条件演算子が存在する場合に正しい分岐コードが生成されません。

(例) `typedef struct {int i;}S;`
`S ss1, ss2;`
`int j;`
`void func() {`
 `func_call((j>10)?ss1:ss2);`
`}`

【回避策】 下記のように if 文に変更してください。

```
if (j > 10){
    func_call(ss1);
} else {
    func_call(ss2);
}
```


No. 7 関数の間接呼び出しの制限

【内容】 関数の間接呼び出し式が、オフセットを必要とする呼び出し方が、以下のいずれかのコンパイラ内部エラーとなります。

(メッセージ)

C2000 : internal : gen_binary() : OP_CALL : left-child's operator is wrong

C5211: syntax error at line <num> in intermediate file

(例) struct S{

```
    int dummy;
    int func_body[0x100];
}sobj;
void f() {
    ((void(*)())sobj.func_body)();
}
```

【回避策】 下記のようにオフセット計算式と呼び出し式を分けてください。

```
void f() {
    void(*fp)=(void(*)())&sobj.func_body;
    fp();
}
```

No. 8 意味不明な関数定義がエラーにならない制限

【内容】 次のような意味不明な関数定義がエラーになりません。

(例) typedef int INTFN();
INTFN f{return(0);}

【回避策】 該当するようなコーディングはしないでください。

No. 12 関数宣言で () が余計に付いている場合エラーになる制限

【内容】 関数宣言で()が余計に付いている場合、エラーになります。

(例) typedef int Int;
void f1((Int));

【回避策】 記述を以下のように変更してください。

```
typedef int Int;
void f1(Int);
```

No. 21 **セクション配置の制限**

【内 容】 #pragma section 指令による tidata セクション配置指定, または sf850 を用いて tidata セクションに配置指定した「構造体の char 型配列」や「char 型メンバへのアクセス」において, その配列, またはメンバのアクセスに使用する sst 命令/sld 命令のディスプレイメント値を越えた場合, リンカでエラーを発生します。

【回避策】 以下のいずれかの方法で回避してください。

1. リンカのエラーとなる構造体の char 型メンバや char 型配列を使用しないでください。
2. リンカのエラーとなる構造体の char 型メンバや char 型配列を tidata セクションに割り当てないでください。

No. 22 **変数の実体がアセンブリ・ソースにあるセクション・ファイルの制限**

【内 容】 変数の実体がアセンブリ・ソースにあり, その変数を C ソース上から参照しているアプリケーションにおいて, セクション・ファイルを sf850 で生成していた場合, リンク時にエラーが発生します。

【回避策】 アセンブリ・ソース中にある変数を, セクション・ファイルから削除して下さい。

No. 23 **同名の外部変数の仮定義が複数ファイルにあるセクション・ファイルの制限**

【内 容】 同名の外部変数の仮定義が複数ファイルにあるとき, セクション・ファイルを sf850 で生成していた場合, リンク時にシンボルの二重定義になる場合がある。
(メッセージ)

ld850 : fatal error: symbol "_xxxx" multiply defined.

【回避策】 同名の外部変数の仮定義が複数ある場合は, 外部変数を参照するファイルでは, 必ず extern 宣言してください。

No. 29 **最適化オプション指定時のエラーの制限**

【内 容】 コンパイル時に指定する最適化オプションの最適化レベルを高くすることにより, コンパイル中に実行されるフェーズ (最適化機能やコンパイル機能など) が多くなります。-Ot オプションを指定した場合, このフェーズ間で生成される中間ファイルが巨大になり, fatal error を起こす場合があります。

【回避策】 最適化オプションのレベルを下げて (-Os など) コンパイルして下さい。

No. 30 **最適化時のオブジェクト・サイズの制限**

【内 容】 最適化を指定し, デバッグ情報を含むオブジェクト・ファイルのサイズは, 非常に大きくなる場合があります。

【回避策】 最適化レベルを下げるか, デバッグしたいファイルだけにデバッグ情報を出力する “-g オプション” をつけてコンパイルするようにしてください。

No. 31 最適化時のデバッグの制限

【内容】 最適化を指定して、ソース・デバッグを行う際は次の制限事項があります。

1. 変数の値を参照した際、正しい値でなく、計算途中の一時的な値が得られることがあります。
2. 配列の一部、構造体の要素、ユーザー定義型のポインタ変数がレジスタに割りつけられた場合、デバッガの変数ウインドウ等にて変数の表示／変更が不正になることがあります。
3. 自動変数の配列の一部、構造体の要素が使用されない場合、領域が削除されることがあります。この場合、デバッガの変数ウインドウ等にて変数の表示／変更が不正になることがあります。変更の場合、スタックを破壊する可能性があります。

【回避策】 ありません。

No. 33 構造体メンバのアドレスの制限

【内容】 構造パッキングを行って、以下のいずれかの条件に該当する場合、

1. ミス・アライン・アクセスに対応していないデバイスを使用している
2. ミス・アライン・アクセスに対応したデバイスで、ミス・アライン・アクセスを禁止している

データのアクセスは、デバイスのデータ・アライメントに従い、アドレスをマスクしてアクセスされるため、構造体メンバのアドレスでのアクセスで、データの消失や切り捨てが生じます。

```
(例) struct test {  
    char c;    /* offset 0 */  
    int i;     /* offset 1-4 */  
} test;  
int *ip, i;  
void func(){  
    i = *ip; /* マスクされたアドレスでアクセスされる */  
}  
void func2(){  
    ip = &(test.i);  
}
```

【回避策】 ありません。

No. 34 ビット・フィールドの制限

【内容】 構造パッキングにおけるビット・フィールドへのアクセスにおいて、ビット・フィールドへのアクセスではビット・フィールドの幅がメンバの型以下の場合、メンバの型で読み込むため、オブジェクトの外部（データのない領域）もアクセスします。通常、正常に実行されますが、I/O がマッピングされていると、不正アクセスとなる場合があります。

```
(例) struct S {  
    int x:21;  
} subj; /* 3 バイト */  
subj.x = 1;
```

【回避策】 ありません。

No. 40 C ソース内における、固有シンボル、予約シンボルの参照の制限

【内 容】 C ソース内において、__gp_DATA などのターゲット固有シンボル、__stext などの予約シンボルを参照することができません。

【回避策】 C ソース内では、ターゲット固有シンボルや予約シンボルを使用しないで下さい。

No. 95 浮動小数点定数と汎整数型の制限

【内 容】 浮動小数点型の値を汎整数型に型変換する場合、整数部の値で表現できる範囲を signed int/signed long 型の値域として、それを越えて unsigned int/unsigned long 型に変換する場合には、コンパイル・エラーとなることがあります。

E2519: invalid has occurred at compile time.

【 例 】

```
unsigned int ui = 2147483647.0;          /* OK */
unsigned int ui = 2147483648.0;          /* エラー */
```

【回避策】 整数型にしてください。

【 例 】

```
unsigned int ui = 2147483648;
```

No. 96 入出力関数の標準ライブラリの入力変換の制限

【内 容】 入出力関数の標準ライブラリの関数 printf, sprintf, vprintf, vsprintf の入力変換で、変換指定子 "g,G"に対して、指定した精度を+1 します。

【 例 】

```
printf("%.2g", 12.3456789);
/* 12 となるべきですが、12.3 となってしまいます。*/
```

【回避策】 ありません。

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。

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連して発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 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。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<http://japan.renesas.com/inquiry>