

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	制限事項の解除	5
4.2	コンパイラの変更点	5
4.2.1	条件式に対する行番号情報の改善.....	5
4.2.2	著作権表記の変更	5
4.2.3	コンパイラ名のマクロの追加.....	5
第5章	注意事項	6
5.1	アセンブラの注意事項	6
5.1.1	ユーザ・オプション・バイトの注意事項.....	6
5.2	メモリ・バンク配置支援ツールの注意事項.....	7
5.2.1	リンク・エラー時の注意事項.....	7
5.2.2	関数情報ファイルに関する注意事項.....	7
5.2.3	コード・サイズに関する注意事項.....	7
5.3	BANKNUM演算子の注意事項.....	8
5.4	-MAオプション指定の注意事項.....	8
5.5	ブートフラッシュの再リンク機能の注意事項.....	8
第6章	制限事項	9
6.1	CA78K0の制限事項.....	9
6.1.1	CA78K0の制限事項一覧.....	9
6.1.2	CA78K0の制限事項詳細.....	10
6.2	メモリ・バンク配置支援ツールの制限事項.....	16
6.2.1	メモリ・バンク配置支援ツールの制限事項一覧.....	16
6.2.2	メモリ・バンク配置支援ツールの制限事項詳細.....	17
第7章	ドキュメント訂正	18
7.1	コーディング編のドキュメント訂正事項.....	18
7.1.1	[10章 注意事項]に関する説明追加	18
7.2	ビルド編のドキュメント訂正事項.....	18
7.2.1	[コンパイル・オプション]タブの[メモリ・モデル]カテゴリに関する説明追加.....	18

第1章 対象デバイスについて

CA78K0 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+ Ver.1.00.00 78K0 コーディング編	R20UT0551JJ0100
CubeSuite+ Ver.1.00.00 78K0 ビルド編	R20UT0555JJ0100
CubeSuite+ Ver1.00.00 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+ CA78K0 V1.20」を選択してください。

第4章 変更点

本章では、78K0 用ビルド・ツールの V1.11 から V1.20 の変更点について説明します。

なお、以下の変更によってバージョンアップでコードが変わる可能性があります。

4.1 制限事項の解除

以下の制限事項を解除しました。

No.75 ##演算子を使ったマクロ展開でエラーとなる制限

【内容】 ## 演算子の直後が、関数形式マクロのパラメータでなく、大小英字でも下線 `_` でもない時、E0803 エラーが出る場合があります。

演算子の直前のトークンと直後のトークンを連結して、別のトークンを生成する時は、E0711,E0301 などの E0803 以外のエラーが出る場合があります。

No.76 アセンブラ・ソースに割り込み関数のシンボル情報が出ない制限

【内容】 以下の条件をすべて満たすときに、リンク時に E3405 エラーになります。

- (1) #pragma interrupt による割り込み関数のベクタテーブルの生成指定がある。
- (2) 同一ソース内に割り込み関数の定義がない。
- (3) -no, アセンブラ・ソース・モジュール・ファイル出力(-a or -sa)、デバッグ情報出力(-g)オプションを有効にする。

4.2 コンパイラの変更点

コンパイラの変更点について説明します

4.2.1 条件式に対する行番号情報の改善

- 条件式に対する行番号情報を改善しました。

4.2.2 著作権表記の変更

- 起動時の表記（実行開始メッセージ）
- インクルード・ファイル

4.2.3 コンパイラ名のマクロの追加

- `__CA78K0__`(コンパイラ名マクロ)を追加しました。

第5章 注意事項

本章では、CA78K0 V1.20 の注意事項について説明します。

5.1 アセンブラの注意事項

5.1.1 ユーザ・オプション・バイトの注意事項

-gb オプションは 80H~84H 番地にユーザ・オプション・バイトを設定します。

アセンブラ・ソース・ファイルと-gb オプションと重複してユーザ・オプション・バイトを設定する場合は、-gb オプションが優先されます。

ただし、どちらもユーザ・オプション・バイトを設定しなければ、デバイス・ファイルの初期値が設定されます。

また、アセンブラ・ソース・ファイルでユーザ・オプション・バイトと QB-MINI2 (愛称: MINICUBE2) のソフトウェア・ブレークを設定する場合、以下のようにコード・セグメントを設定してください。ユーザ・オプション・バイトに対するコード・セグメントの指定がない場合、デバイス・ファイルの初期値が設定されるため、エラーになります。

例)

```
-----*.asm-----  
OCN_SBR  CSEG  AT 07EH  
          DB  0FFH  
          DB  0FFH  
;  
OPT_SET  CSEG  OPT_BYTE      ; コード・セグメントを指定してください。  
          DB  30H  
          DB  00H  
          DB  00H  
          DB  00H  
          DB  02H  
END
```

5.2 メモリ・バンク配置支援ツールの注意事項

5.2.1 リンク・エラー時の注意事項

CA78K0 のリンカ(LK78K0)エラー “multiple symbol define” が発生するプロジェクトで、メモリ・バンク配置支援ツールを使用した場合、メモリ・バンク配置支援ツールはエラーメッセージは出力せずに処理を継続しますが、その際に生成する関数情報ファイル、各種情報ファイルの内容は不正なものとなります。

5.2.2 関数情報ファイルに関する注意事項

プロジェクトにおいて「メモリ・バンク配置オプション」の「関数情報ファイルを出力する」プロパティを「はい」にしている場合、ビルド実行時にソースおよびプロジェクトに変更がない場合でも、必ず関数情報ファイル(*.fin)を出力するため、そのファイルを使ってコンパイルおよびリンクが実行され、ビルドがスキップされません。

関数情報ファイルの自動生成が不要な場合には、「関数情報ファイルを出力する」または、「メモリ・バンク配置支援ツールを使用する」プロパティを「いいえ」にしてください。

5.2.3 コード・サイズに関する注意事項

メモリ・バンク配置支援ツールが出力する関数情報ファイル、配置情報ファイル、オブジェクト情報ファイル内のコード・サイズ情報は関数のみのコード・サイズです。

const データなどは含まませんので、[メモリ・バンク配置オプション]タブ の [マージン]カテゴリで、空き領域を調整してください。

5.3 BANKNUM演算子の注意事項

アセンブラ・ソースで BANKNUM 演算子を使用する場合は、対象となるシンボルの再配置属性が"UNIT" or "UNITP"の場合は別ファイルにしてください。

```
-----a.asm-----
EXTRN _TABLE
PUBLIC _BANKNO
C1 CSEG
_BANKNO :
        :
        MOV A, #BANKNUM _TABLE
        :
END
-----b.asm-----
C2 CSEG UNITP
_TABLE :
        DB 11H
        DB 12H
END
```

5.4 -maオプション指定の注意事項

-ma オプションで指定するファイルのフォルダ名またはファイル名に','(カンマ)が含まれている場合、コンパイラはカンマを区切り子と判断するため適切にファイルを認識できません。

【回避策】

カンマをフォルダ名およびファイル名に含めないでください。

5.5 ブート-フラッシュの再リンク機能の注意事項

再リンク機能を使用する際には、入力するロード・モジュール・ファイル (LMF) 内のセグメントに対してマージディレクティブ指定は行わないでください。

第6章 制限事項

本章では、CA78K0 の制限事項について説明します。

6.1 CA78K0の制限事項

CA78K0 V1.20 における制限事項一覧を以下に示します。

6.1.1 CA78K0の制限事項一覧

(1) アセンブラ部の制限事項一覧

No.	制限事項
1	構造化アセンブリ言語記述で範囲が交差する制御文がエラーとなる制限 (PM+版のみ)
5	saddr.bit の値を持つビット・シンボルを EQU 定義する際、saddr の部分に最適化の影響を受けるレーベルが記述されるとエラー等が不正になる制限

(2) コンパイラ部の制限事項一覧

No.	制限事項
16	signed 型のビット・フィールドを符号無しビット・フィールドとして処理する制限
43	入出力関数の標準ライブラリの出力変換で、動作が不正となる制限
44	int/short 型の最小値-32768 のサイズが 4 となる制限
47	関数定義の識別子の型に対して、仮引数型と関数定義の識別子の型とが合わずに、エラーとなる制限
48	関数定義の識別子並びで、宣言していない仮引数を int 型とせずエラーとなる制限
49	#演算子が正しく展開できない制限

6.1.2 CA78K0の制限事項詳細

CA78K0 V1.20 における制限事項の詳細は以下のとおりです。

(1) アセンブラ部の制限事項詳細

No. 1 構造化アセンブラ言語記述で範囲が交差する制御文がエラーとなる制限 (PM+版のみ)

【内容】 制御文を#ifdef~#endif 分割または交差するように囲んだ場合、#ifdef が真の時に制御文がエラーになります。

(例)

```

switch(mode)
#ifdef stsw
case 1:
break
#endif
default:
break
ends

```

← #ifdef~#else/#endif の範囲
← case~次の case/default/ends までの範囲

【回避策】 ネスト(入れ子)の場合には、正しく処理されます。制御文の範囲を交差させないようにソースを書き直してください。

(例)

```

#ifdef stsw
switch(mode)
case 1:
break
default:
break
ends
#else
switch(mode)
default:
break
ends
#endif

```

← #ifdef~#else/#endif の範囲
← case~次の case/default/ends までの範囲

No. 5 **saddr.bit** の値を持つビット・シンボルを EQU 定義する際、**saddr** の部分に最適化の影響を受けるレーベルが記述されるとエラー等が不正になる制限

【内 容】 **saddr.bit** の値を持つビット・シンボルを EQU 定義する際に、**saddr** の部分に最適化の影響を受けるレーベルが記述されると、不正な処理を行います。

以下のケースで、不正な処理をします。

(1) **saddr.bit** が 0FD20H の時、レーベルがパス 1 で領域外、パス 2 で領域内となる場合、EQU 定義行に対し、パス 1 ではエラーを出力しますが、パス 2 では出力されず、不正なオブジェクトを生成します。

(2) **saddr** が 0FF1FH の時、レーベルがパス 1 で領域内、パス 2 で領域外となる場合、EQU 定義行に対し、パス 1 ではエラーになりませんが、パス 2 ではエラーを出力します。この EQU シンボルを参照した後に定義するレーベルに対し、以下のアセンブラエラーが出力されます。

「F410 Phase error」

また、このレーベルを参照するとオブジェクトが不正になります。

【回避策】 ありません。

(2) コンパイラ部の制限事項詳細

No. 16 signed 型のビット・フィールドを符号無しビット・フィールドとして処理する制限

【内容】 signed 型のビット・フィールドを、符号無しビット・フィールドとして処理します。

【回避策】 ありません。

No. 43 入出力関数の標準ライブラリの出力変換で、動作が不正となる制限

【内容】 関数 printf, sprintf, vprintf, vsprintf の出力変換で、以下のような場合に動作が不正となります。

変換指定子 "d, i, o, u, x および X" に対して、".2" のように精度を指定した場合に、0 フラグを無視しません。

(例)

```
#include <stdio.h>

void func()
{
    printf("%04.2d\n", 77);
}
```

※(不正な動作) "0077" となります

(正しい動作) " 77" となります

変換指定子 "g, G" に対して、指定した精度+1 を精度とします。

(例)

```
#include <stdio.h>

void func()
{
    printf("%.2g", 12.3456789);
}
```

※ (不正な動作) "12.3" となります

(正しい動作) "12" となります

【回避策】 ありません。

No. 44 int/short 型の最小値-32768 のサイズが 4 となる制限

【内 容】 int/short 型の最小値-32768 のサイズが 4 となります。

(例)

```
int x;
void func()
{
    x = sizeof(-32768);
}
```

※(不正な動作) x の値が 4 となります

(正しい動作) x の値が 2 となります

【回避策】 (-32767-1) と記述してください。

No. 47 関数定義の識別子の型に対して、仮引数型と関数定義の識別子の型と合わずに、エラーとなる制限

【内 容】 関数定義の識別子の型に対して実引数拡張をしていないため、仮引数型と関数定義の識別子の型と適合せずに、E0747 エラーとなります。

(例)

```
int fn_char(int);
int fn_char(c)
char c;
{
    return 98;
}
```

【回避策】 仮引数型と関数定義の識別子の型を合わせてください。

No. 48 関数定義の識別子並びで、宣言していない仮引数を int 型とせずエラーとなる制限

【内 容】 関数定義の識別子並びで、宣言していない仮引数を int 型とせず E0706 エラーとなります。

(例)

```
void func(x1, x2, f, x3, lp, fp)
int (*fp)( );
long *lp;
float f;
{
    :
}
```

【回避策】 関数定義の仮引数はすべて宣言してください。

No. 49 #演算子が正しく展開できない制限

【内 容】 以下のいずれかの条件で正しく展開できません。

(条件1) #演算子で「'''」を正しく展開できずに、コンパイル時にエラーとなります。

(条件1の例)

```
#include <string.h>
#define str( a) (# a)
int x;
void func()
{
    if (strcmp(str(''), "¥''") == 0) x++;
}
```

※(不正な動作) コンパイル時にエラーとなります

(正しい動作) if (strcmp(("¥''"), "¥''") == 0) x++; となります

(条件2) #演算子とネストを含むマクロを正しく展開できません。

(条件2の例)

```
#define str(a) #a
#define xstr(a) str(a)
#define EXP 1
char *p;
void func()
{
    p = xstr(12EEXP);
}
```

※(不正な動作) “p = ("12E1");”となります。

(正しい動作) “p = ("12EEXP");”となります。

【回避策】 ありません。

6.2 メモリ・バンク配置支援ツールの制限事項

メモリ・バンク配置支援ツールにおける制限事項一覧を以下に示します。

6.2.1 メモリ・バンク配置支援ツールの制限事項一覧

No.	制限事項
1	__flashf 関数に関する制限
2	オプション指定に関する制限
3	リンク・ディレクティブに関する制限
4	callt 領域に関する制限

6.2.2 メモリ・バンク配置支援ツールの制限事項詳細

メモリ・バンク配置支援ツールの制限事項の詳細は以下の通りです。

No. 1 `__flashf` 関数に関する制限

【内容】 `__flashf` 関数は共通領域に配置しなければなりません。メモリ・バンク配置支援ツールは `__flashf` 関数を判別できないため、`__flashf` 関数が共通領域以外に配置され、ビルド時にエラーとなる可能性があります。

【回避策】 `__flashf` 関数を含むファイルは、C ソースファイルのプロパティを開き、[ビルド設定]タブ の [メモリ・バンク]カテゴリの [配置する領域] で、"共通領域"を指定してください。

No. 2 オプション指定に関する制限

【内容】 メモリ・バンク配置支援ツールはコンパイル・オプション、アセンブル・オプション、リンク・オプションの「その他の追加オプション」で指定できるオプションを解析していません。

【回避策】 ビルド・ツール・プロパティで指定できるオプションは、「その他の追加オプション」の欄に記述しないようにしてください。

No. 3 リンク・ディレクティブに関する制限

【内容】 リンク・ディレクティブ・ファイルには対応していません。メモリ・バンク配置支援ツールは、デバイス・ファイルで定義されているメモリ領域にもとづいて配置を行います。（例えば、`const` データや初期値データなど関数以外をバンク領域に配置した場合にも対応していません。）

【回避策】 [メモリ・バンク配置オプション]タブ の [マージン]カテゴリで、各領域の空き領域を調整してください。

No. 4 `callt` 領域に関する制限

【内容】 メモリ・バンク配置支援ツールでは、`#pragma section` 指定、または疑似命令の発行により、`@@CALT` のセクション名が変更された場合、参照情報ファイルに出力する“`callt` 領域を利用して呼び出された関数の参照回数”が不正値となります。

【回避策】 `@@CALT` のセクション名を変更しないでください。

第7章 ドキュメント訂正

本章では、CubeSuite+のドキュメントの訂正について説明します。

また、Help にも同様の内容が記載されておりますので、ドキュメントと同様に読み替えてください。

7.1 コーディング編のドキュメント訂正事項

7.1.1 [10章 注意事項]に関する説明追加

【場 所】 858 ページ → [(6) va_start マクロ使用時]

【追加後】 関数により第一引数のオフセットが異なるため、stdarg.h に定義されている va_start マクロの動作は保証されません。

以下のようにマクロを使い分けてください。

- 第一引数を指定する場合は、va_starttop マクロを使用してください。
- 第二引数以降を指定する場合は、va_start マクロを使用してください。

バンク関数呼び出しルーチンを経由する関数は、以下のようにマクロを使い分けてください。

- 第一引数を指定する場合は、va_starttop_banked マクロを使用してください。
- 第二引数以降を指定する場合は、va_start_banked マクロを使用してください。

7.2 ビルド編のドキュメント訂正事項

7.2.1 [コンパイル・オプション]タブの[メモリ・モデル]カテゴリに関する説明追加

【場 所】 209 ページ → [フラッシュ用オブジェクトを出力する]

【追加後】 フラッシュ用オブジェクトを出力するかどうかを選択します。
コンパイラのオプション-zf に相当します。

「はい(-zf)」を選択した場合

- [コンパイル・オプション]タブの[スタートアップ]カテゴリの[標準のスタートアップを使用する]プロパティにて「いいえ」が選択されていない場合は、「はい(フラッシュ領域用)」に設定します。
- [リンク・オプション]タブの[デバイス]カテゴリの[フラッシュ・スタート・アドレスを設定する]プロパティを「いいえ」に設定します。

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>