

本文書は次に示す内容を記載しています。RH850/E1L、RH850/E1M-S グループ E1/E20 エミュレータ使用時の注意事項については、RH850/E1L、RH850/E1M-S グループの E1/E20 ユーザーズマニュアル別冊にも掲載されておりますので、あわせてお読みください。

- ・対象デバイスには該当せず、エミュレータのみが該当する制限事項
- ・対象デバイスとエミュレータの両方が該当する制限事項で、エミュレータのみが修正予定のある制限事項

なお、対象デバイスの制限事項については、次に示す文書に記載されておりますので、こちらもあわせてお読みください。

- ・対象デバイスのユーザーズマニュアル
- ・対象デバイスの制限事項文書

## 目次

1. 制限事項および仕様追加事項一覧.....	2
2. 制限事項および仕様追加事項詳細.....	3

## 1. 制限事項および仕様追加事項一覧

No	仕様変更 / 追加 / 制限事項	CS+ Ver.	
		V2.02.00 V3.00.00	V3.01.00 V3.02.00
1	内蔵フラッシュメモリ書き換え時の制限事項(クロックモニタ)	✓	✓
2	ダウンロード/アップロード機能の制限事項(外部フラッシュメモリ)	✓	✓
3	ダウンロード/アップロード機能の制限事項(進捗ダイアログ)	✓	✓
4	メモリマップ機能の制限事項	✓	-
5	メモリ機能の制限事項(CPUリセット)	✓	✓
6	周辺モジュールが使用する専用RAM領域のメモリ表示機能に関する制限事項	✓	✓
7	DMM指定によるメモリアクセスに関する制限事項	✓	✓
8	IOLレジスタ表示機能に関する制限事項	✓	✓
9	ハードウェアブレークの有効数に関する制限事項	✓	✓
10	WriteアクセスまたはRead/Writeアクセスブレーク条件設定時の制限事項	✓	✓
11	クロック設定中にブレークした場合の制限事項	✓	✓
12	リードアクセスブレークに関する制限事項	✓	✓
13	時間測定機能に関する制限事項	✓	-
14	分岐元アドレスがトレース取得されなかった場合の制限事項【注】	✓	✓
15	範囲外トレースイベントに関する制限事項1【注】	✓	✓
16	範囲外トレースイベントに関する制限事項2【注】	✓	✓
17	範囲外トレースイベントの設定で異常終了した場合の制限事項【注】	✓	✓
18	トレースの同時指定に関する制限事項【注】	✓	✓
19	デバッグコンソールに関する制限事項	✓	✓
20	マルチコア用のブート・ローダ・プロジェクトに関する制限事項	✓	-
21	周辺ガード機能が有効でIORパネルを表示した場合の制限事項	✓	✓
22	トレースの優先順位をデータ優先(ノンリアルタイム)に設定した場合の制限事項【注】	✓	✓
23	欠番(本製品では該当しないため削除しました)	-	-
24	ロードモジュール分割時の制限事項	✓	✓
25	トレース機能の制限事項(トレース設定)【注】	✓	✓
26	メインウィンドウを最大化している状態での制限事項	✓	✓
27	Pythonスクリプトの実行について	✓	✓
28	デバッグ情報関連に関する制限事項	✓	✓
29	デバッグ対象のロードモジュール内に同じ名称のファイルが複数存在する場合の制限事項	✓	✓
30	for文やインライン関数内のブレークポイント設定について	✓	✓
31	ホットプラグインの制限事項	✓	-

✓:該当する -:該当しない

【注】:トレース機能を持つデバイスを使用する場合に該当します。

## 2. 制限事項および仕様追加事項詳細

### No.1 内蔵フラッシュメモリ書き換え時の制限事項(クロックモニタ)

【内 容】 デバッグは、フラッシュメモリ書き換え時<sup>(注)</sup>にPLL の設定を変更します。このため、フラッシュメモリ書き換え時の周波数は、ご使用の周波数より高速になる可能性があります。クロックモニタ(CLMA)で設定した周波数の上限を超える場合は、フラッシュメモリ書き換えを抑止します。

#### 【PLL 設定の変更】

PLL0 を分周なしで使用します。

注:フラッシュメモリ書き換えは、下記の操作時に発生します。

- a. 内蔵フラッシュメモリへのダウンロード
- b. メモリパネル操作による内蔵フラッシュメモリの変更
- c. ソフトウェアブレークの設定・解除
- d. ソフトウェアブレーク成立後の再実行(ステップ含む)

【回避策】 デバッグによるクロック周波数の変更が問題になる場合、プロパティパネルの「フラッシュ書き込み時にクロックを変更する」を「いいえ」に設定してください。

### No.2 ダウンロード/アップロード機能の制限事項(外部フラッシュメモリ)

【内 容】 外部フラッシュへのダウンロードは対応していません。

【回避策】 申し訳ありませんが、回避策はありません。

### No.3 ダウンロード/アップロード機能の制限事項(進捗ダイアログ)

【内 容】 ダウンロード中に表示される[進捗状況]ダイアログで[キャンセル]ボタンを押した後[キャンセル]ボタンが押された状態になってからも、ダイアログが閉じるまで暫く時間が掛かります。

これはキャンセル処理が動作しているためですので、ダイアログが閉じるまでお待ちください。

【回避策】 申し訳ありませんが、回避策はありません。

### No.4 メモリマップ機能の制限事項

【内 容】 外部メモリ領域は対応していません。

【回避策】 申し訳ありませんが、回避策はありません。

### No.5 メモリ機能の制限事項(CPU リセット)

【内 容】 [ダウンロード後に CPU をリセットする]を[はい]および[CPUリセット後に指定シンボル位置まで実行する]を[はい]に設定している場合はダウンロード後に指定シンボル位置まで実行されます。ただし、デバッグ・ツールとの接続およびダウンロードを[デバッグ・ツールへダウンロード]メニューで実施した場合、リセットから指定シンボル位置まで実行している間のメモリパネル文字色はピンク色(リアルタイム RAM モニタ状態)になりません。

【回避策】 「接続」と「ダウンロード」を別々に実行してください。一度停止した場合、次の実行からはピンク色になります。

### No.6 周辺モジュールが使用する専用 RAM 領域のメモリ表示機能に関する制限事項

【内 容】 周辺モジュールが使用する専用 RAM 領域のメモリ表示機能は対応していません。

【回避策】 ウォッチパネルで直接アドレスを指定してアクセス可能です。

例: \*((int\*)0xFFC62230)

#### No.7 DMM 指定によるメモリアクセスに関する制限事項

【内 容】 DMM 指定によるメモリアクセスは、LocalRAM 領域および GlobalRAM 領域に対してのみ有効です。

【回避策】 申し訳ありませんが、回避策はありません。

#### No.8 IO レジスタ表示機能に関する制限事項

【内 容】 IO レジスタの表示には対応していますが、ビット表示には対応していません。

【回避策】 ウォッチパネルにビット IO を登録することによりビット IO の内容を表示可能です。

#### No.9 ハードウェアブレイクの有効数に関する制限事項

【内 容】 ハードウェアブレイクは CPU1、PCU の合計で 12 本までしか設定できません。

【回避策】 申し訳ありませんが、回避策はありません。

#### No.10 Write アクセスまたは Read/Write アクセスブレイク条件設定時の制限事項

【内 容】 Write アクセスまたは Read/Write アクセスブレイク条件設定時に、ビット操作命令で RMW(Read Modify Write)の Write サイクルが発生してもブレイクが発生しません。

【回避策】 申し訳ありませんが、回避策はありません。

#### No.11 クロック設定中にブレイクした場合の制限事項

【内 容】 クロック設定中にブレイクした場合、フラッシュメモリの書き換えができません。

【回避策】 クロック設定中にブレイクした状態で下記の操作を実施する場合は、プロパティパネルの「フラッシュ書き込み時にクロックを変更する」を「いいえ」に設定してください。

- a. 再ダウンロードなどフラッシュメモリの書き換えを伴う操作
- b. ソフトウェアブレイクの設定および解除

また、クロック設定処理にソフトウェアブレイクを設定しないでください。

#### No.12 リードアクセスブレイクに関する制限事項

【内 容】 SYSCALL 命令に対しては、リードアクセスブレイク条件を設定できません。

【回避策】 申し訳ありませんが、回避策はありません。

#### No.13 時間測定機能に関する制限事項

【内 容】 任意区間の実行時間の測定はできません。

【回避策】 実行停止までの実行時間の計測を使用してください。

#### No.14 分岐元アドレスがトレース取得されなかった場合の制限事項

【内 容】 分岐トレースは、分岐命令実行時に取得する分岐先アドレスから次の分岐命令実行時に取得する分岐元アドレスまでの命令を補完して表示します。トレース開始/終了イベントを使用する場合、設定によっては補完に必要な分岐情報が取得できないため、トレースが表示できません。

【回避策】 申し訳ありませんが、回避策はありません。

トレース開始/終了イベントの設定位置を分岐命令が含まれる位置に変更してください。

**No.15 範囲外トレースイベントに関する制限事項 1**

【内 容】 範囲外トレースイベントにおいて、開始(下限)と終了(上限)の 2 点を設定しないとトレースできません。

【回避策】 申し訳ありませんが、回避策はありません。

**No.16 範囲外トレースイベントに関する制限事項 2**

【内 容】 範囲外トレースイベントは 1 区間しか設定できません。

【回避策】 申し訳ありませんが、回避策はありません。

**No.17 範囲外トレースイベントの設定で異常終了した場合の制限事項**

【内 容】 範囲外トレースイベントを設定した状態で USB 切断等の理由によりデバッグ・ツールとの接続が切断された場合、再接続しても範囲外トレースイベントが有効になりません。また、その後のトレースイベント設定はできません。

【回避策】 USB 切断等の理由によりデバッグ・ツールとの接続が切断された後、一度プロジェクトを閉じ、プロジェクトを開いてください。その後、デバッグ・ツールと接続しダウンロードすれば範囲外トレースイベントの設定が可能となります。

**No.18 トレースの同時指定に関する制限事項**

【内 容】 区間トレースと範囲外トレースの同時使用はできません。

【回避策】 申し訳ありませんが、回避策はありません。

**No.19 デバッグコンソールに関する制限事項**

【内 容】 デバッグコンソールには対応していません。

【回避策】 申し訳ありませんが、回避策はありません。

**No.20 マルチコア用のブート・ローダ・プロジェクトに関する制限事項**

【内 容】 マルチコア用のブート・ローダ・プロジェクトには対応していません。

【回避策】 申し訳ありませんが、回避策はありません。

**No.21 周辺ガード機能が有効で IOR パネルを表示した場合の制限事項**

【内 容】 周辺ガード機能が有効で IOR パネルを表示した場合、周辺ガード機能で設定したレジスタも表示してしまうため、ガードエラーが発生します。よって、周辺ガード機能を有効した状態で IOR パネルを表示しないでください。

【回避策】 周辺ガード機能で設定した以外のレジスタを表示する場合、ウォッチパネルに登録して表示するようにしてください。

**No.22 トレースの優先順位をデータ優先(ノンリアルタイム)に設定した場合の制限事項**

【内 容】 トレースの優先度をデータ優先に設定した場合は、トレースを停止する機能(トレースフルストップ機能)は使用できません。

【回避策】 トレースを停止する機能を使用する場合は、トレースの優先度をスピード優先(リアルタイム)でご使用ください。

#### No.24 ロードモジュール分割時の制限事項

【内 容】 CG-RH コンパイラを使用してロードモジュールを分割生成した場合に、下記の制限があります。

- a. ソースレベルデバッグができません。
- b. 2 番目以降の出力ファイルは、デバッグ・ツールに自動登録されません。

【回避策】 ロードモジュールは分割しないでください。

#### No.25 トレース機能の制限事項(トレース設定)

【内 容】 トレース機能を持たないデバイスでもデバッグ・ツール設定タブのプロパティにおいてトレース設定を表示しますが、トレース機能を使用することはできません。プロパティ項目は変更しないでください。

【回避策】 申し訳ありませんが、回避策はありません。

#### No.26 メインウィンドウを最大化している状態での制限事項

【内 容】 メインウィンドウを最大化している状態で、ステータスバーのコア切り替えコンボ・ボックスのリストを表示するとリスト用の部品がタスクバーの裏に隠れて選択できない状態になります。

【回避策】 タスクバーの設定を「自動的に隠す」に設定するか、タスクバーの表示位置を「右」、「左」、「上」のどれかに設定することで回避可能です。

#### No.27 Python スクリプトの実行について

【内 容】 フック処理設定で Python スクリプトを実行する場合、以下のコマンド以外は使用できません。

```
debugger.Register.GetValue  
debugger.Register.SetValue  
debugger.Memory.GetValue  
debugger.Memory.SetValue
```

【回避策】 Python コンソールの Hook 関数によって登録できるフック機能を使用してください。

**No.28 デバッグ情報関連に関する制限事項**

CC-RH コンパイラが生成するデバッグ情報にいくつかの制限があります。

この制限により発生するデバッグ機能の現象を以下に記載します。

なお、これらはコンパイラの生成したデバッグ情報と、生成コードとのズレによる現象であり、

CC-RH コンパイラの生成したコードの実行結果には問題ありません。

**No.28-1 ソースレベルステップ機能**

【内容1】現象1: デバッグ上でソースレベルステップを実行時に、実行しないはずの行を実行しているように見える場合があります。

【例 1-1】 以下のようなプログラムを記述した場合、生成コード上は(\*1)の for ループ終了後に(\*2)の位置で停止し、変数“i”の値によって次に実行する文へ分岐します。

しかし、本制限に該当する場合には、(\*2)の位置での変数“i”の値に関わらず(\*3)の位置へPC 矢印が移動してしまうように見えます。

```
void main()
{
    int i = 0;
    int j = 0;
    for (j = 0; j < 10; j++) {          <-( *1)
        i = atoi("100");
    }
    if (i != 100) {                   <-( *2)
        i = atoi("100");             <-( *3)
    }
    return;
}
```

発生条件:

- a. (\*2)に該当する行が制御文(if, for, switch など)である。
- b. (\*2)の直前に実行されるプログラムが以下のいずれかである。
  - 制御文
  - ラベル、goto、return 文
  - 三項演算、論理演算、否定演算

【回避策】 申し訳ありませんが、回避策はありません。

【内容2】現象2: デバッガ上でソースレベルステップを実行時に、実行しないはずの行を実行しているように見える場合があります。

【例 1-2】 以下のようなプログラムを記述した場合、生成コード上は GetCount()関数の初回呼び出し時には(\*1)の位置から(\*2)の case 文へ分岐します。しかし、本制限に該当する場合には、(\*2)の case 文の直前の実行行の位置(\*3)へ PC 矢印が移動してしまうように見えます。

```
enum Count { ZERO, ONE, TWO };

enum Count GetCount()
{
    static enum Count value = ZERO;

    switch (value) {                <-(*1)
    case TWO:
        value = ZERO;
        break;
    case ONE:
        value = TWO;                <-(*3)
        break;
    case ZERO:
    default:
        value = ONE;                <-(*2)
        break;
    }
    return value;
}
```

発生条件:

分岐先の命令の内容が以下のいずれかに該当する場合に発生します。

- a. 定数値の取得を行っている。
- b. 変数のアドレスを算出している。

【回避策】 申し訳ありませんが、回避策はありません。



**No.28-2 変数情報表示機能に関する制限**

【内容】現象 3: 関数内で同じ名称の変数を複数定義している場合、各行で停止時に参照できる変数の情報が期待値と異なる場合があります。なお、本現象はコンパイル時の最適化レベル<sup>(注)</sup>に依存します。

注: 設定項目(プロパティ)の位置

[ビルド・ツール] - [共通オプション] - [よく使うオプション(リンク)]。

【例】以下のようなプログラムを記述した場合、生成コード上は、

(\*1)の位置で char 型の変数“a”が最も内側のスコープに生存し、

(\*2)の位置で int 型の変数“a”が最も内側のスコープに生存します。

しかし、本制限に該当する場合には、(\*1)、(\*2)のどちらの位置からも一方の変数の情報しか参照できません。

```
void main()
{
    int a = 100;
    {
        char a = 'A';
        a++;          <-(*1)
    }
    a++;             <-(*2)
}
```

●(\*1)の位置でのウォッチパネル上の表示結果

“a”            'A' (0x41) "signed char" "0xfefb1004" //期待値  
または “a” 100 (0x00000064) "int"            "0xfefb1000"

●(\*2)の位置でのウォッチパネル上の表示結果

“a”            'B' (0x42) "signed char" "0xfefb1004"  
または “a” 100 (0x00000064) "int"            "0xfefb1000" //期待値

発生条件:

コンパイル時に最適化を行うと発生することがあります。

【回避策】コンパイル時の最適化レベルを「デバッグ優先」にしてください。

**No.29 デバッグ対象のロードモジュール内に同じ名称のファイルが複数存在する場合の制限事項**

【内 容】 デバッグ対象のロードモジュール内に同じ名称のファイルが複数存在する場合、エディタで行アドレスが正しく表示されません。また、イベント機能が正しく設定されません。

フォルダ構成例:

```
C:\Work\CS+\ProjA\ProjA.mtpj\Src\main.c      -> A.abs
C:\Work\CS+\ProjB\ProjB.mtpj\Src\main.c      -> B.abs
```

上記 2 つのロードモジュールを同時にデバッグする場合

※上記例では複数のロードモジュールを使用していますが、同一ロードモジュール内にファイルが複数存在する場合も同様です。

発生条件: ファイルのコンパイルディレクトリ以降の相対パス(ファイル名含む)が一致する

CS+上でビルドする場合:

コンパイルディレクトリ=プロジェクトファイル(\*.mtpj<sup>(注)</sup>)位置

注: サブプロジェクトの場合は "\*.mtpj"ではなく、 "\*.mtsp"になります

MakeFile 等でビルドする場合:

コンパイルディレクトリ=カレントディレクトリ

【回避策】 以下のいずれかを実施いただくことでデバッグ中も同名ファイルが区別できるようになります。

a. コンパイルディレクトリからの相対パスが異なるようにフォルダ構成を変更してください。

```
Before : ProjA\Src\main.c
        : ProjB\Src\main.c
After  : ProjA\SrcA\main.c
        : ProjB\SrcB\main.c
```

→変更を実施することで相対パスはそれぞれ

"SrcA\main.c"

"SrcB\main.c"になります。

b. デバッグ対象ファイルに同名ファイルがなくなるようファイル名を変更してください。

```
Before : ProjA\Src\main.c
        : ProjB\Src\main.c
After  : ProjA\Src\mainA.c
        : ProjB\Src\mainB.c
```

**No.30 for 文やインライン関数内のブレークポイント設定について**

【内 容】 以下のプログラムをCソースに記述するとソースプログラム 1 行に対する命令が複数箇所に配置されますがエディタ上にはそのうちの 1 箇所のアドレスのみを表示しています。

この行に対しブレークポイントを設定した場合、ブレークするのはエディタに表示したアドレス実行時のみとなります。

a. インライン関数<sup>(注)</sup>

b. テンプレート関数

c. for 文、do-while 文の先頭行

注: 最適化によりインライン展開された関数も含む

【回避策】 申し訳ありませんが、回避策はありません。

## ご注意書き

1. 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・システムの設計において、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て、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。
2. 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。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因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に対する侵害に関し、当社は、何らの責任を負うものではありません。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。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等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。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
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、産業用ロボット等  
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、各種安全装置等  
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図し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損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の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。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、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、社会的損害等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、お客様の機器・システムとしての出荷保証を行ってください。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、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の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い合わせください。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。
9. 本資料に記載されている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・システムに使用することはできません。また、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用用途に使用しないでください。当社製品または技術を輸出する場合は、「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。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
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禁じます。

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が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

営業お問い合わせ窓口

<http://www.renesas.com>

営業お問い合わせ窓口の住所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 エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24 (豊洲フォレシア)

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