

要旨

本書は、RL78/F13, RL78/F14, RL78/F15, RL78/F1A 等が持つホットプラグイン機能を用いたデバッグ方法を説明するものです。

CS+ for CA, CX, Renesas Flash Programmer, E1 エミュレータ、ホットプラグアダプタ (R0E000010ACB00)、CPU ボード (QB-R5F10BMG-TB) を使用し、プログラム作成→ビルド→デバッグの順にツール設定方法や接続手順等を具体的に説明します。

CS+ for CC をご使用の場合も、ホットプラグインの接続方法は同じなので、本書の手順に沿ってご使用ください。ただし、ホットプラグイン初期化関数のみ異なるのでご注意ください。

備考 1 : CS+の操作方法は、CS+のユーザーズマニュアルをご確認ください。

2 : E1 エミュレータとユーザシステムとの接続回路や使用するユーザ資源の情報については、エミュレータ ユーザーズマニュアル 別冊 (文書番号 : R20UT1994) をご確認ください。

3 : Renesas Flash Programmer の操作方法は、Renesas Flash Programmer のユーザーズマニュアルをご確認ください。

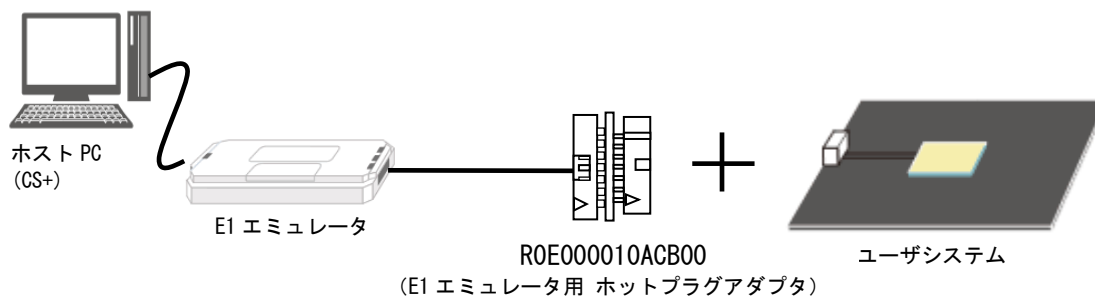


図 1. システムの概要

目次

1.	ホットプラグインを使用したデバッグの概要.....	3
1.1	ホットプラグインの特徴.....	3
1.2	ホットプラグイン接続後のデバッグ.....	4
2.	ホットプラグイン機能を用いたデバッグ方法.....	5
2.1	作業手順の概要.....	5
2.2	実践.....	6
2.2.1	ホットプラグインが可能なプログラムの作成.....	6
2.2.2	ビルドの実行.....	13
2.2.3	プログラムの書き込みと実行.....	17
2.2.4	ホットプラグイン接続.....	18
2.2.5	デバッグ.....	22
2.2.6	ユーザシステムとの切断.....	25
3.	注意事項.....	26
3.1	デバッグ DTC 動作クロックに関する注意事項.....	26
3.2	DTC の保留命令に関する注意事項.....	26
3.3	データアクセスイベントに関する注意事項.....	26
3.4	32bit 長以上の変数アクセスに関する注意事項.....	26
3.5	スタンバイモードに関する注意事項.....	27
3.6	リセットに関する注意事項.....	27
3.7	RAM の使用に関する注意事項.....	27

1. ホットプラグインを使用したデバッグの概要

1.1 ホットプラグインの特徴

ホットプラグインとは、ユーザシステム動作中に“プログラム動作を継続したまま”“リセットをかけることなく”“プログラムの内容を変更せずに”デバッガを接続できる機能です。

ユーザシステムの検査時や出荷後に不良が発生した際、ホットプラグインにより不良発生状態を保持したままデバッグすることができます。

主な特徴は以下のとおりです。

●特徴1 不良発生状態を保持

- ・プログラムの動作を継続したまま接続可能
- ・リセットを発生させずに接続可能
- ・プログラムの内容を変更せずに接続可能

●特徴2 プログラムをセキュリティ機能で保護

- ・デバッガ接続時に 10 バイトの ID コード認証

●特徴3 動作を継続したまま解析

- ・デバッガ接続後、CPU を止めずに DTC 方式の RAM モニタが可能

●特徴4 更なる詳細解析

- ・強制ブレークが使用可能
- ・ブレーク後、ソフトウェアブレークやイベントブレークの設定が可能

※ホットプラグインは、不良解析を目的とした機能なので、ユーザプログラム開発の初期段階では通常のオンチップデバッグ機能を使用してください。

1.2 ホットプラグイン接続後のデバッグ

ホットプラグインで動作中のマイコンへデバッガを接続した後は、デバッグ専用の DTC を使用した RRM や DMM を使用して CPU を動作させたままデバッグを行うことができます。

従来の RL78 マイコンではデバッグ時のメモリアクセスに CPU を使用していましたが、デバッグ DTC の搭載により CPU を使用せずにデバッグ時のメモリアクセスが可能となりました。

DTC 方式 RRM/DMM でアクセス可能な領域は以下の通りです。

[読み出し/書き込み可能な資源]

- ・ RAM
- ・ SFR

[読み出しのみ可能な資源]

- ・ データフラッシュ (読み出し許可時のみ)
- ・ ミラー領域
- ・ 汎用レジスタ

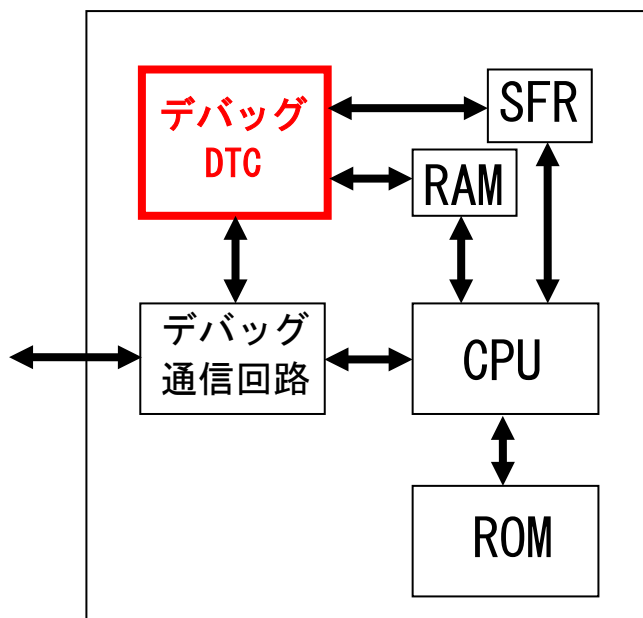


図 2. デバッグ DTC の概念図

[用語の説明]

- ・ DTC : データトランスファコントローラ。CPU を使わずにメモリとメモリの間でデータを転送する機能
- ・ RRM : リアルタイム RAM モニタ。CPU を停止せずに RAM/SFR の読み出しを行う機能
- ・ DMM : ダイナミックメモリモディフィケーション。CPU を停止せずに RAM/SFR へ書き込みを行う機能

2. ホットプラグイン機能を用いたデバッグ方法

ホットプラグイン機能を用いたデバッグ方法を説明します。

2.1 作業手順の概要

以下の手順で作業を行います。

CS+をインストールしてから作業を行ってください。

① プログラムの作成

: ホットプラグインを使用するためには、ユーザプログラム上にてホットプラグイン機能初期化関数を実行する必要があります。
ここでは、ホットプラグイン機能初期化関数含めたユーザプログラムの作成方法を説明します。



② ビルド

: ホットプラグインを使用するには、ユーザ RAM の確保やオンチップデバッグオプションバイトの設定が必要です。ここではビルドオプションでこれらを設定する方法を説明します。



③ プログラムの書き込みと実行

: Renesas Flash Programmer でヘキサファイルを書き込み、実機動作させます。



④ ホットプラグイン接続

: E1 エミュレータを使用して、ホットプラグイン接続を実行します。



⑤ デバッグ

: ホットプラグイン接続した状態のデバッグ機能について説明します。

2.2 実践

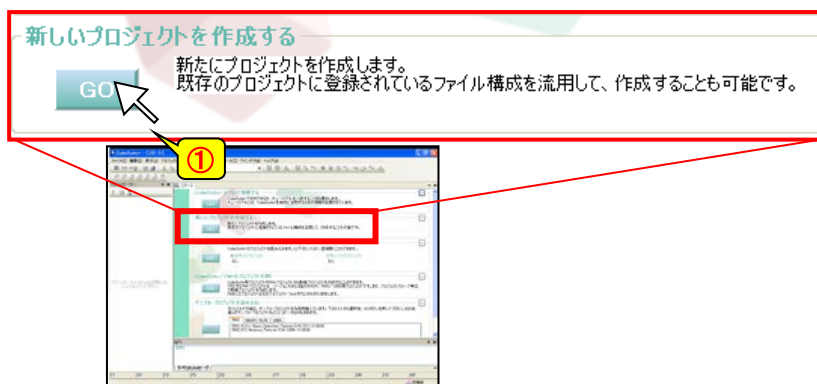
2.2.1 ホットプラグインが可能なプログラムの作成

(1) CS+の起動

[スタート]→[すべてのプログラム]→[Renesas Electronics CS+]→[CS+ for CA, CX]を選択してCS+を起動します。

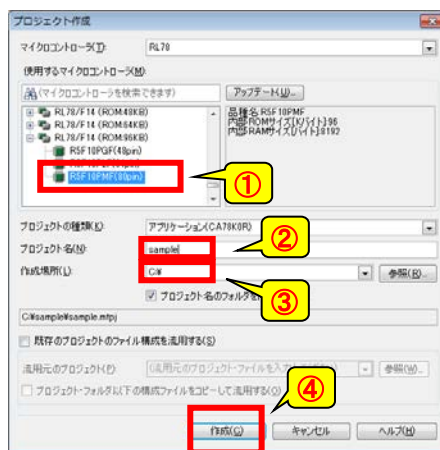
(2) プロジェクトの作成

- ・スタートパネルより新規プロジェクトを作成します。



①作成開始 「新しいプロジェクトを作成する」の[GO]ボタンを押下してください。

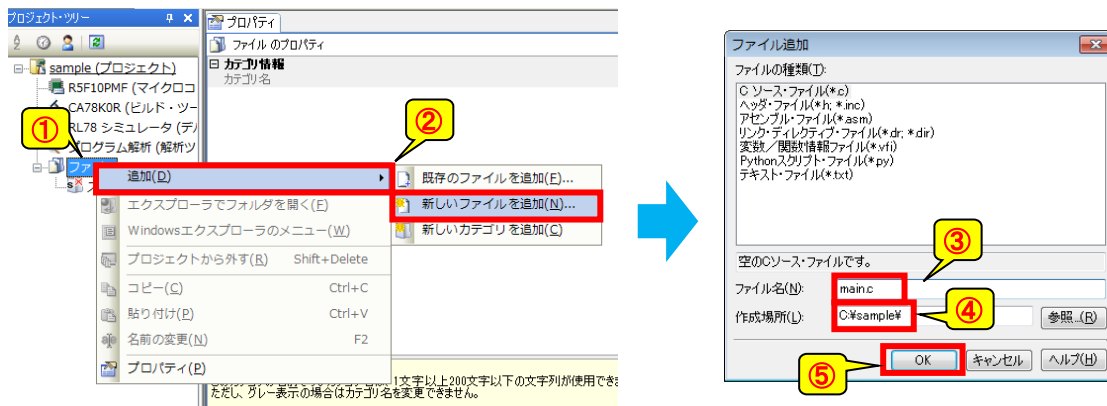
- ・プロジェクトの設定を行います。



① [使用するマイクロコントローラ]の指定	使用するデバイス名を選択します。ここでは「R5F10PMF (80pin)」を選択しています。
② [プロジェクト名]の指定	任意の指定が可能です。ここでは「sample」を指定しています。
③ [作成場所]の指定	任意の指定が可能です。ここでは「C:\¥」を指定しています。
④プロジェクトの生成	[作成]ボタンを押下してください。

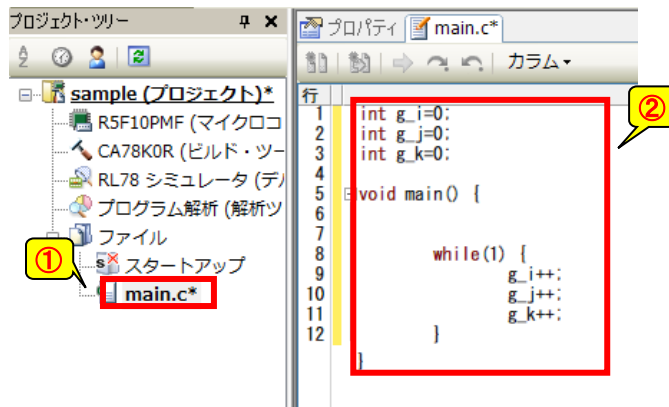
(3) ユーザプログラムの作成

・ main.c ファイルを作成します。



① 「ファイル」のコンテキストメニューを開く	プロジェクトツリー内の「ファイル」を選択後、右クリックしてください。
②新規ファイルを追加	「追加」→「新しいファイルを追加」を選択してください。
④ [ファイル名]の指定	任意の指定が可能です。ここでは「main.c」を指定しています。
⑤ [作成場所]の指定	任意の指定が可能です。ここでは「C:\sample\」を指定しています。
⑤ファイル生成	[OK] ボタンを押下してください。

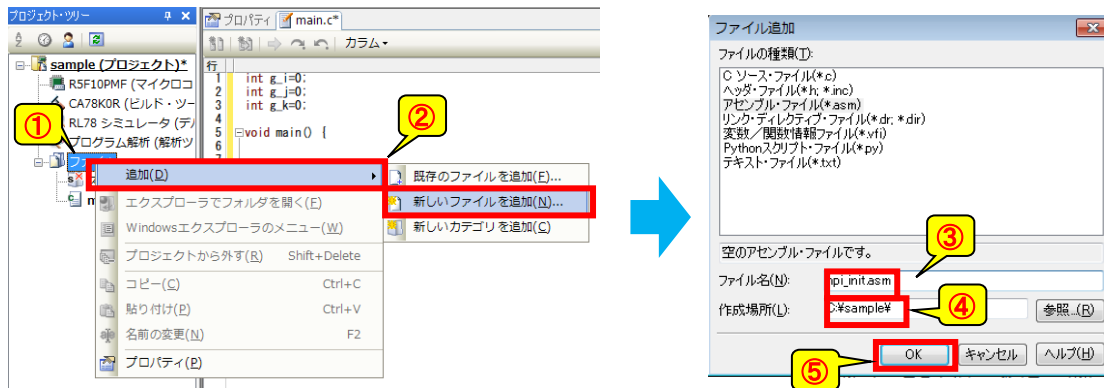
・ main.c ファイルをエディットします。



① 「main.c」を開く	プロジェクトツリー内の「main.c」をダブルクリックしてください。
②main 関数のエディット	下記内容を main.c ファイルにエディットしてください。 (下記内容をコピー&ペーストしてください。)
<pre> int g_i=0; int g_j=0; int g_k=0; void main() { while(1) { g_i++; g_j++; g_k++; } } </pre>	

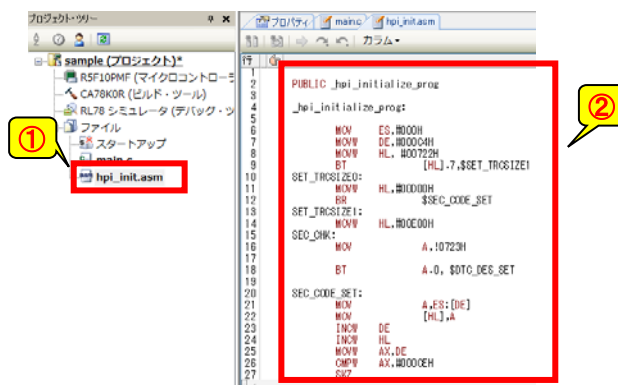
(4) ホットプラグイン用初期設定関数の作成

- ・ホットプラグイン用初期化関数 (hpi_initialize_prog) 定義ファイルを作成します。プロジェクトに hpi_init.asm ファイルを追加してください。



① 「ファイル」のコンテキストメニューを開く	プロジェクトツリー内の「ファイル」を選択後、右クリックしてください。
② 新規ファイルを追加	「追加」→「新しいファイルを追加」を選択してください。
③ [ファイル名]の指定	任意の指定が可能です。ここでは「hpi_init.asm」を指定しています。
④ [作成場所]の指定	任意の指定が可能です。ここでは「C:\sample\」を指定しています。
⑤ ファイル生成	[OK] ボタンを押下してください。

- ・ホットプラグイン用初期化関数 (hpi_initialize_prog) をエディットします。



① hpi_init.asm を開く	プロジェクトツリー内の「hpi_init.asm」をダブルクリックしてください。
② ホットプラグイン初期化関数のエディット	次ページの内容を hpi_init.asm ファイルにエディットしてください。(次の内容をコピー&ペーストしてください。)

※下記内容は、ホットプラグイン初期化関数です。デバッグ専用 DTC の初期化/有効化を行っています。
そのままユーザプログラムへコピーしてください。

[初期化関数の使用方法]

- ・ RAM の初期化完了後、なるべく早いタイミングで呼び出してください。
CPU リセット解除後 400ms 内に初期化関数を実行完了してください。実行完了しない場合、ホットプラグイン後の端子リセット/POC リセット発生でデバッグがハングアップします。
- ・ リセット後 1 回のみ呼び出しとしてください。
- ・ 初期化内で SFR を操作している箇所がありますので、SFR ガード機能を OFF にしてから呼び出してください。
- ・ 掲載の初期化関数では、関数実行完了まで割り込みを禁止しています。割り込み禁止としない場合は” for disable interrupt” のコメントがある 3 行を削除してください。(割り込み許可とする場合、初期化関数実行完了までの時間やスタック使用量にご注意ください。)

CS+ for CA, CX の場合

```

PUBLIC _hpi_initialize_prog

_hpi_initialize_prog:
    PUSH    PSW        ;for disable interrupt
    DI          ;for disable interrupt
    PUSH    AX
    PUSH    DE
    PUSH    HL
    MOV     A, ES
    PUSH    AX

    MOV     ES, #000H
    MOVW   DE, #000C4H
    MOVW   HL, #00722H
    BT     [HL], 7, $SET_TRCSIZE1
SET_TRCSIZE0:
    MOVW   HL, #00D00H
    BR     $SEC_CODE_SET
SET_TRCSIZE1:
    MOVW   HL, #00E00H
SEC_CHK:
    MOV     A, !0723H
    BT     A, 0, $DTC_DES_SET

SEC_CODE_SET:
    MOV     A, ES: [DE]
    MOV     [HL], A
    INCW   DE
    INCW   HL
    MOVW   AX, DE
    CMPW   AX, #000CEH
    SKZ
    BR     $SEC_CODE_SET
DTC_DES_SET:
    MOV     ES, #00FH
    MOV     L, #00EH
    MOVW   AX, #0FFADH
    MOVW   [HL], AX
    MOV     L, #010H
DTC_SET:
    MOV     [HL+00H], #013H
    MOV     [HL+01H], #001H
    MOV     [HL+02H], #00AH
    MOV     [HL+03H], #00AH
    MOV     [HL+04H], #000H
    MOV     A, H
    MOV     [HL+05H], A
    MOV     [HL+06H], #023H
    MOV     [HL+07H], #007H
    MOV     [HL+08H], #000H
    
```

```

MOV    [HL+09H], #001H
MOV    [HL+0AH], #001H
MOV    [HL+0BH], #001H
MOV    [HL+0CH], #023H
MOV    [HL+0DH], #007H
MOV    [HL+0EH], #0ADH
MOV    [HL+0FH], #0FFH
MOV    [HL+10H], #011H
MOV    [HL+11H], #001H
MOV    [HL+12H], #010H
MOV    [HL+13H], #010H
MOV    [HL+14H], #0ADH
MOV    [HL+15H], #0FFH
MOV    [HL+16H], #000H
MOV    A, H
MOV    [HL+17H], A
MOV    [HL+18H], #003H
MOV    [HL+19H], #001H
MOV    [HL+1AH], #010H
MOV    [HL+1BH], #010H
MOV    [HL+1CH], #000H
MOV    A, H
MOV    [HL+1DH], A
MOV    [HL+1EH], #0ADH
MOV    [HL+1FH], #0FFH
SET1   DTCEN

POP    AX
MOV    ES, A
POP    HL
POP    DE
POP    AX
POP    PSW      ;for disable interrupt

RET
END

```

CS+ for CC の場合

```

.PUBLIC _hpi_initialize_prog

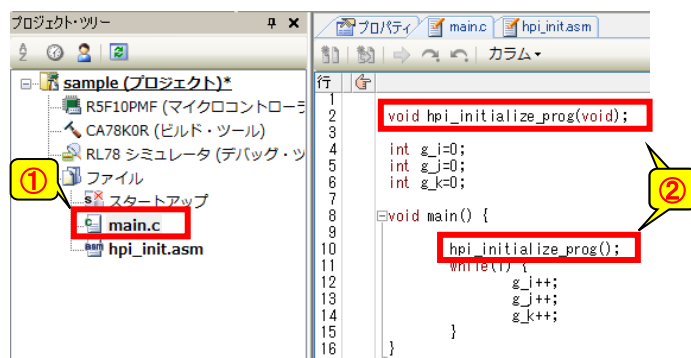
_hpi_initialize_prog:
    PUSH PSW      ;for disable interrupt
    DI           ;for disable interrupt
    PUSH AX
    PUSH DE
    PUSH HL
    MOV A, ES
    PUSH AX
    MOV ES, #0x000
    MOVW DE, #0x000C4
    MOVW HL, #0x00722
    BT [HL], 7, $SET_TRCSIZE1
    SET_TRCSIZE0:
    MOVW HL, #0x00D00
    BR $SEC_CODE_SET
SET_TRCSIZE1:
    MOVW HL, #0x00E00
SEC_CHK:
    MOV A, !0x0723
    BT A, 0, $DTC_DES_SET
SEC_CODE_SET:
    MOV A, ES: [DE]
    MOV [HL], A
    INCW DE
    INCW HL
    MOVW AX, DE
    CMPW AX, #0x000CE
    SKZ
    BR $SEC_CODE_SET
DTC_DES_SET:
    MOV ES, #0x00F
    MOV L, #0x00E
    MOVW AX, #0x0FFAD
    MOVW [HL], AX

```

```

MOV L, #0x010
DTC_SET:
MOV [HL+0x00], #0x013
MOV [HL+0x01], #0x001
MOV [HL+0x02], #0x00A
MOV [HL+0x03], #0x00A
MOV [HL+0x04], #0x000
MOV A, H
MOV [HL+0x05], A
MOV [HL+0x06], #0x023
MOV [HL+0x07], #0x007
MOV [HL+0x08], #0x000
MOV [HL+0x09], #0x001
MOV [HL+0x0A], #0x001
MOV [HL+0x0B], #0x001
MOV [HL+0x0C], #0x023
MOV [HL+0x0D], #0x007
MOV [HL+0x0E], #0x0AD
MOV [HL+0x0F], #0x0FF
MOV [HL+0x10], #0x011
MOV [HL+0x11], #0x001
MOV [HL+0x12], #0x010
MOV [HL+0x13], #0x010
MOV [HL+0x14], #0x0AD
MOV [HL+0x15], #0x0FF
MOV [HL+0x16], #0x000
MOV A, H
MOV [HL+0x17], A
MOV [HL+0x18], #0x003
MOV [HL+0x19], #0x001
MOV [HL+0x1A], #0x010
MOV [HL+0x1B], #0x010
MOV [HL+0x1C], #0x000
MOV A, H
MOV [HL+0x1D], A
MOV [HL+0x1E], #0x0AD
MOV [HL+0x1F], #0x0FF
SET1 !DTCEN
POP AX
MOV ES, A
POP HL
POP DE
POP AX
POP PSW          ;for disable interrupt
RET
    
```

- ・ main 関数内でホットプラグイン用初期化関数 (hpi_initialize_prog) を呼び出す記述をします。



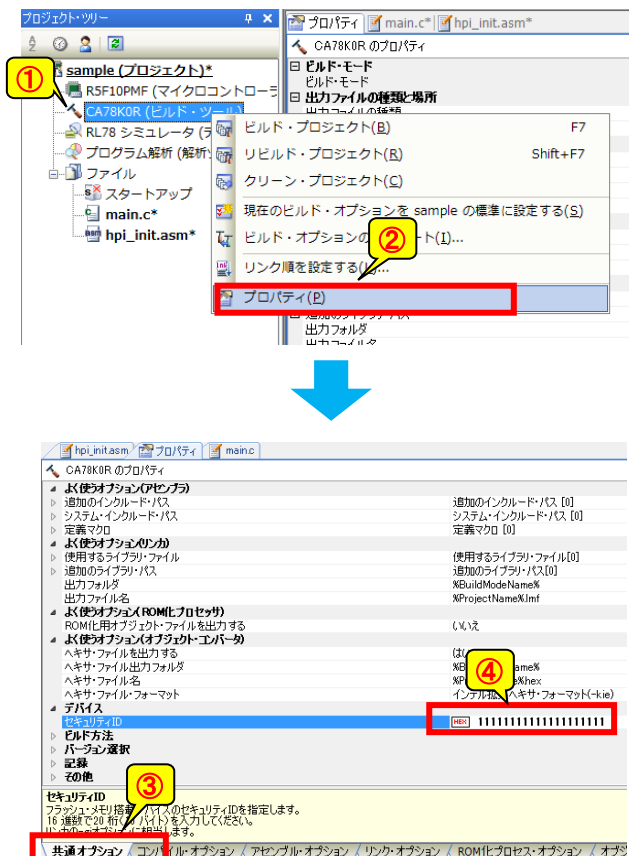
① 「main.c」を開く	プロジェクトツリー内の「main.c」をダブルクリックしてください。
②ホットプラグイン初期化関数を記述	下記内容を main.c ファイル内に追加してください。
<pre> void hpi_initialize_prog(void); int g_i=0; int g_j=0; int g_k=0; void main() { hpi_initialize_prog(); while(1) { g_i++; g_j++; g_k++; } } </pre> <p>今回の追加内容</p> <p>今回の追加内容</p>	

2.2.2 ビルドの実行

(1) セキュリティ ID の設定

・セキュリティ ID の設定を行います。

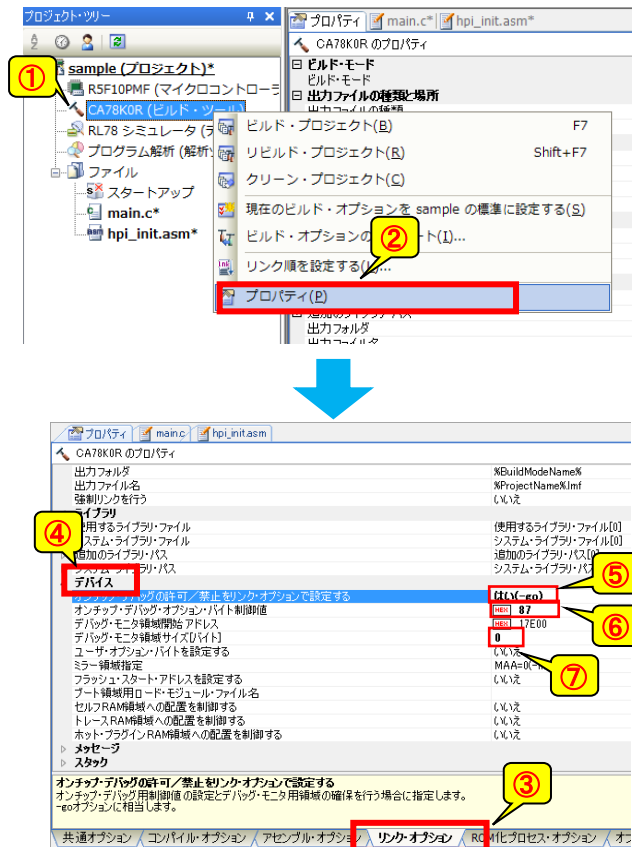
ビルドツールのプロパティにある共通オプションで設定します。



<p>① CA78K0R(ビルドツール)」のコンテキストメニューを開く</p>	<p>プロジェクトツリー内の「CA78K0R(ビルドツール)」を選択後、右クリックしてください。</p>
<p>②プロパティを開く</p>	<p>コンテキストメニュー内の「プロパティ」を選択してください。</p>
<p>③[共通オプション]タブを開く</p>	<p>メインパネルにある[共通オプション]タブを選択してください。</p>
<p>④「セキュリティ ID」の設定</p>	<p>「デバイス」項目にある[セキュリティ ID]を指定します。任意の指定が可能です。ここでは「11111111111111111111」を指定しています。</p>

(2) オプションバイトの設定

- ・オンチップデバッグオプションバイトの設定を行います。
ビルドツールのプロパティにあるリンクオプションで設定します。



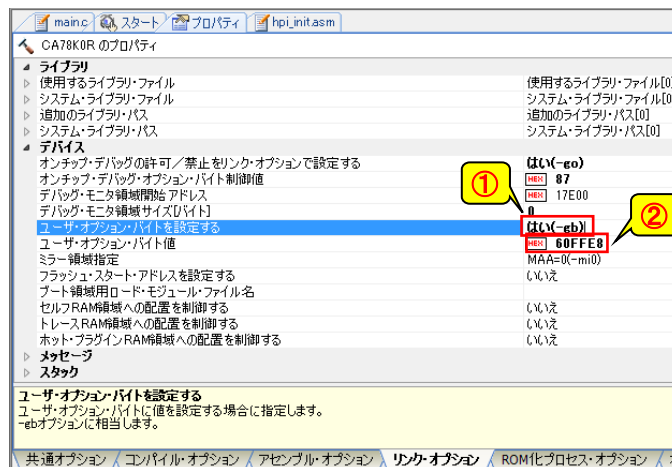
① 「CA78K0R(ビルドツール)」のコンテキストメニューを開く	プロジェクトツリー内の「CA78K0R(ビルドツール)」を選択後、右クリックしてください。
② プロパティを開く	コンテキストメニュー内の「プロパティ」を選択してください。
③ [リンクオプション]タブを開く	メインパネルにある[リンクオプション]タブを選択してください。
④ 「デバイス」項目を開く	「デバイス」項目を開いてください。
⑤ [オンチップデバッグの許可/禁止をリンクオプションで設定する]	「はい」を選択してください。
⑥ [オンチップデバッグオプションバイト制御値]	「87」を指定してください。
⑦ [デバッグモータ領域サイズ]	「0」を指定してください。 ※ホットプラグインでない通常のオンチップデバッグを行う際はサイズを指定します。

[オンチップデバッグオプションバイト設定について]

ホットプラグインを使用する場合、オンチップデバッグオプションバイトには0x87を設定する必要があります。ホットプラグイン検出には低速オンチップオシレータを使用するため、この設定を行うと低速オンチップオシレータが動作し、ユーザプログラムによる停止ができなくなります。ただし、スタンバイモード中はレジスタ設定により低速オンチップオシレータを停止可能です。

- ・ユーザオプションバイトの設定を行います。

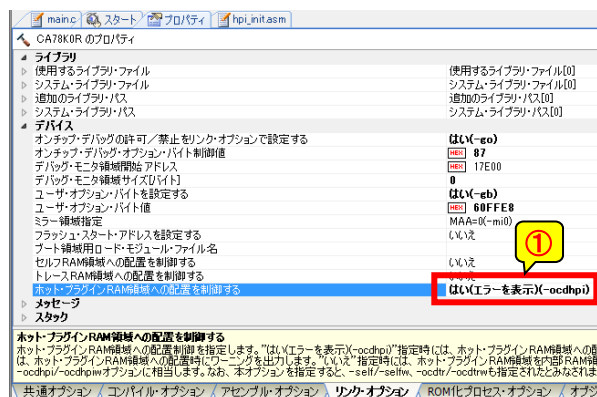
ビルドツールのプロパティにあるリンクオプションで設定します。



①[ユーザオプションバイトを設定する]	「はい」を選択してください。
②[ユーザオプションバイト値]	任意の指定が可能です。ここでは「60FFE8」を指定しています。

(3) ホットプラグイン用 RAM の確保

- ・ホットプラグインで使用する RAM 空間をユーザプログラムが使用しないようにビルドツールのプロパティにあるリンクオプションで設定します。
※ホットプラグイン用 RAM 領域への RAM 確保が不必要な品種の場合は、この項目 CubeSuite+にありません。設定不要ですので次の手順に進んでください。



①[ホットプラグイン RAM 領域への配慮を制御する]	「はい(エラーを表示)」を選択してください。
-----------------------------	------------------------

(4) ビルドの実行

- ・エディットとビルドツールのオプション設定が終了したのでビルドを実行します。

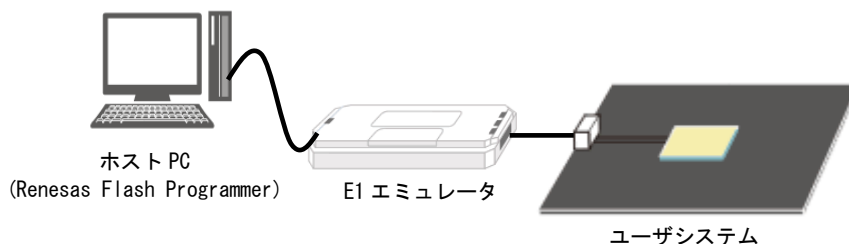


①リビルドの実行	メニューボタン内にある[リビルド]ボタンを押下してください。
----------	--------------------------------

2.2.3 プログラムの書き込みと実行

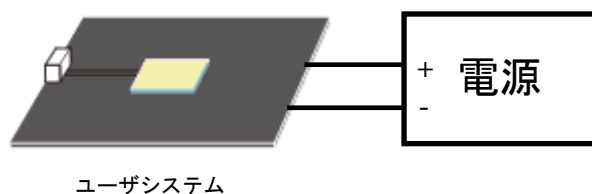
(1) プログラムの書き込み

- ・ Renesas Flash Programmer (ライター) にて、作成したヘキサファイルを対象デバイスに書き込みます。
Renesas Flash Programmer の使用方法は、[スタート]→[すべてのプログラム]→[Renesas Electronics Utilities]→[書き込みツール] →[Renesas Flash Programmer Vx.xx]内にあるユーザーズマニュアルをご参照ください。



(2) プログラムの実行

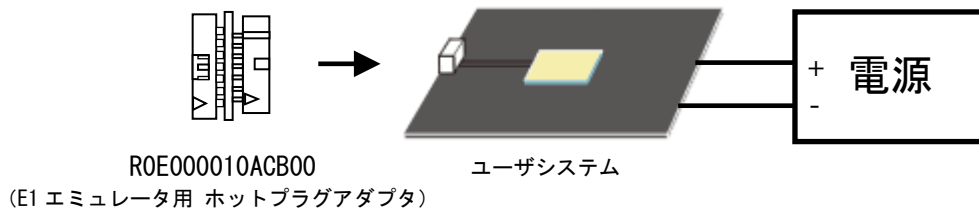
- ・ ユーザシステムに電源を供給し実機動作を行います。



2.2.4 ホットプラグイン接続

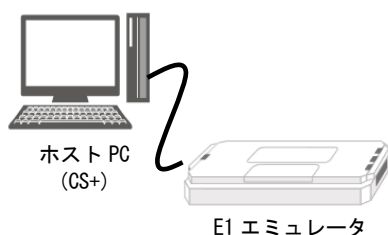
(1) ホットプラグアダプタ (ROE000010ACB00) とユーザシステムの接続

- ・最初に、必ず ROE000010ACB00 をユーザシステムに装着します。
ROE000010ACB00 は、E1 エミュレータと接続する際に必ず GND から接続できるような機構となっています。



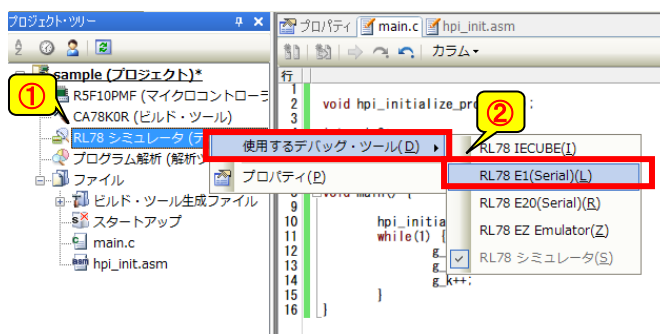
(2) E1 エミュレータとホスト PC との接続

- ・E1 エミュレータとホスト PC を接続します。



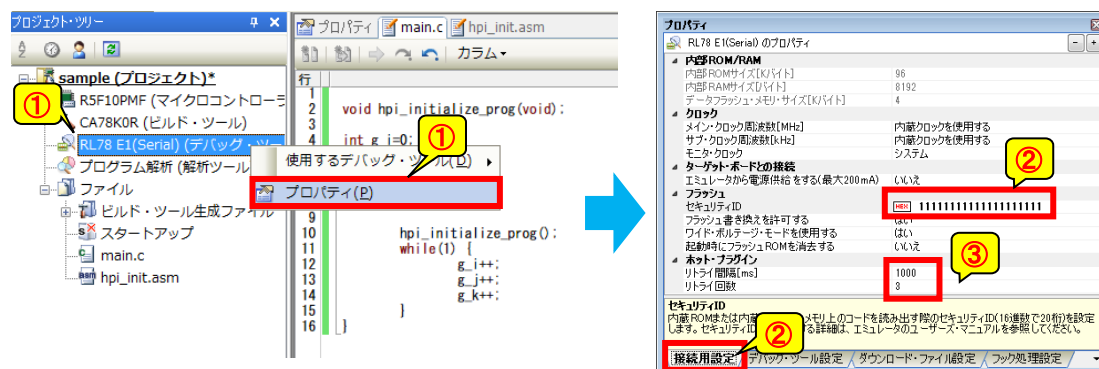
(3) デバッグ機能の設定

- ・CS+にて、デバッグツールを選択します。



① 「デバッグツール」のコンテキストメニューを開く	プロジェクトツリー内の「デバッグツール」を選択後、右クリックしてください。
② デバッグツールの選択	「使用するデバッグツール」→「RL78 E1 (Serial)」を選択してください。

- ・E1 エミュレータ接続時の認証用セキュリティ ID を設定します。また、ホットプラグインのリトライ設定を行います。

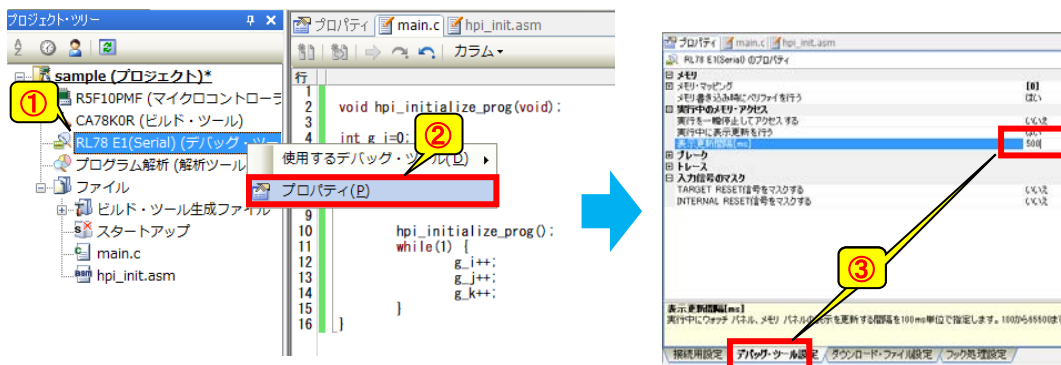


① デバッグツール」のプロパティパネルを開く	プロジェクトツリー内の「デバッグツール」を選択後、右クリックしてください。
② 認証用のセキュリティ ID の入力	[接続用設定]タブ内の「フラッシュ」項目にある[セキュリティ ID]を指定します。 ビルドツールのオプションで設定した値と同じ値を入力します。ここでは「11111111111111111111」を指定してください。
③ ホットプラグインのリトライ設定	CPU が STOP モード中などで低速オンチップオシレータが停止しているために、ホットプラグイン接続に失敗した場合のリトライ間隔とリトライ回数を設定します。 今回のプログラムでは低速オンチップオシレータを停止しないので、デフォルト設定(1000ms、3回)のままで使用しています

[ホットプラグインのリトライ設定について]

- ・リトライ間隔にはプログラム内の STOP モード最大継続時間を入力してください。
ただし、リトライ処理中(リトライ間隔×リトライ回数)はデバッガが操作できない状態となります。
このため、リトライ処理の時間が極端に長くないようご注意ください。

- RAM モニタ間隔を設定します。



①「デバッグツール」のコンテキストメニューを開く	プロジェクトツリー内の「デバッグツール」を選択後、右クリックしてください。
②「プロパティ」を開く	「プロパティ」を選択してください。
③[表示更新間隔]の指定	[デバッグツール]タブ内の「実行中のメモリアクセス」項目にある[表示更新間隔]を指定します。 100msc 以上の値が指定可能です。 ここでは「500」を指定しています。

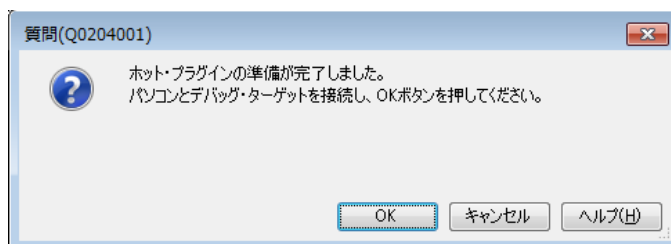
(4) ホットプラグイン接続の実行

- ホットプラグイン接続の準備を行います。



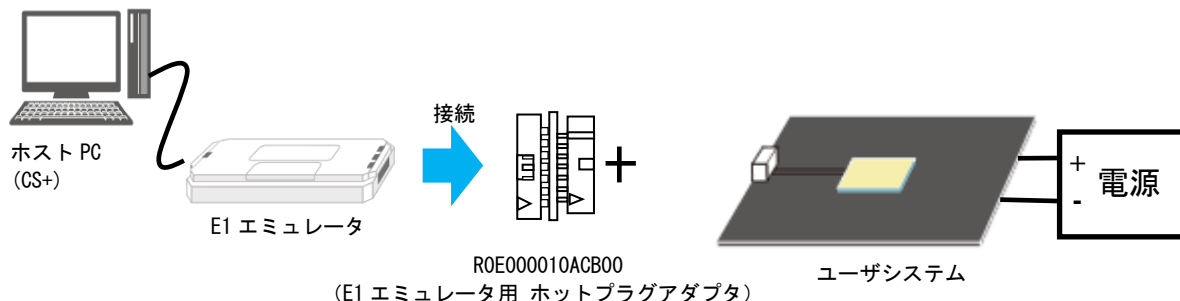
①デバッガをホットプラグイン準備状態へ移行	「デバッグ」メニューの「ホットプラグイン」を選択してください。
-----------------------	---------------------------------

- 下記ダイアログが表示されることを確認します。



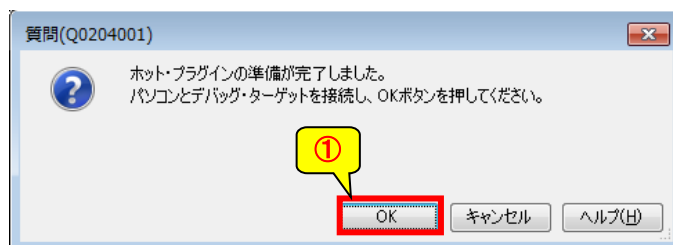
・ E1 エミュレータとホットプラグアダプタ (ROE000010ACB00) との接続

E1 と ROE000010ACB00 を接続します。E1 エミュレータ側とホットプラグアダプタ側でコネクタをまっすぐに接続してください。



・ ホットプラグイン接続

接続後にデバッガの[OK]ボタンを押下してください。



①ホットプラグイン接続

[OK]ボタンを押下してください。

[ホットプラグイン接続時に使用できない機能]

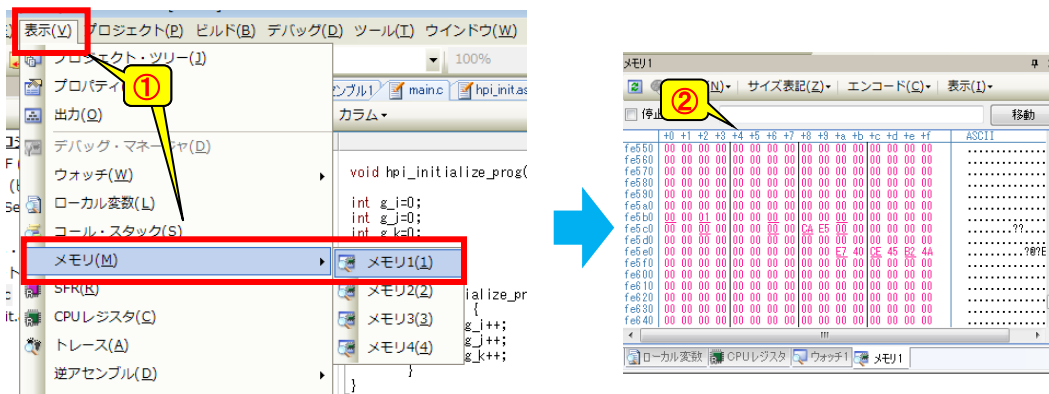
以下の機能はホットプラグイン接続時に使用できません。ホットプラグイン接続後に一度強制ブレークをしてから使用してください。

- ・ トレース機能
- ・ 端子マスク機能
- ・ イベントブレーク機能/シーケンシャル・ブレーク機能
- ・ ソフトウェアブレーク機能
- ・ 停止時のタイマ/シリアル系周辺エミュレーション動作/停止設定

2.2.5 デバッグ

(1) メモリ表示更新機能

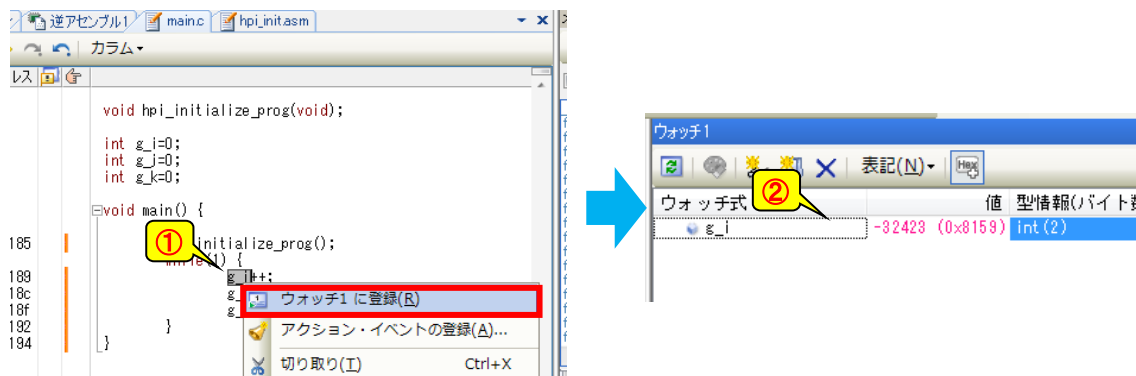
- ・メモリパネルを開いてください。



①メモリパネルを開く	「表示」メニューを開いて[メモリ]→[メモリ 1]を選択してください。
②RAM モニタ機能が動作しているメモリパネルの確認	メモリパネルの値がピンク色になり、定期的に表示が更新されます。

(2) グローバル変数の表示

- ・ソース上から変数を登録してください。



①グローバル変数をウォッチパネルに登録	main.c ソース内にある「g_i」をドラッグし、右クリックしてください。その後「ウォッチ1に登録」を選択してください。
②ウォッチ登録されている変数の確認	変数の表示が定期的に更新されています。

(3) 強制ブレーク

- ・プログラムをブレークさせます。

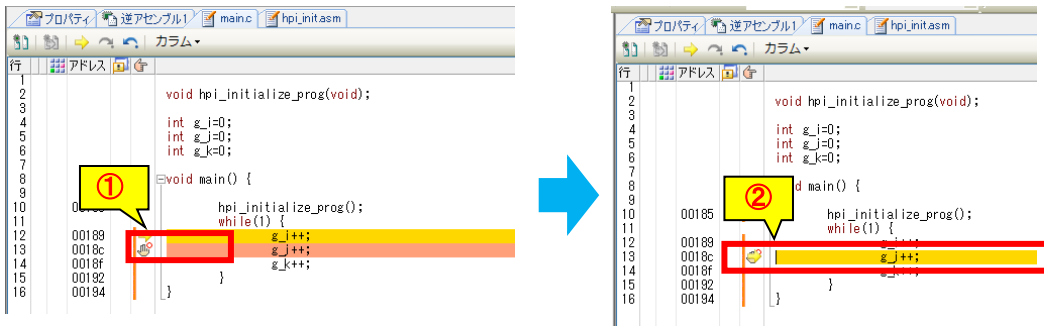


①プログラムの強制ブレーク	メニュー上にある「停止」ボタンを押下してください。
---------------	---------------------------

(4) ブレーク後のデバッグ機能

- ・ソフトウェアブレーク

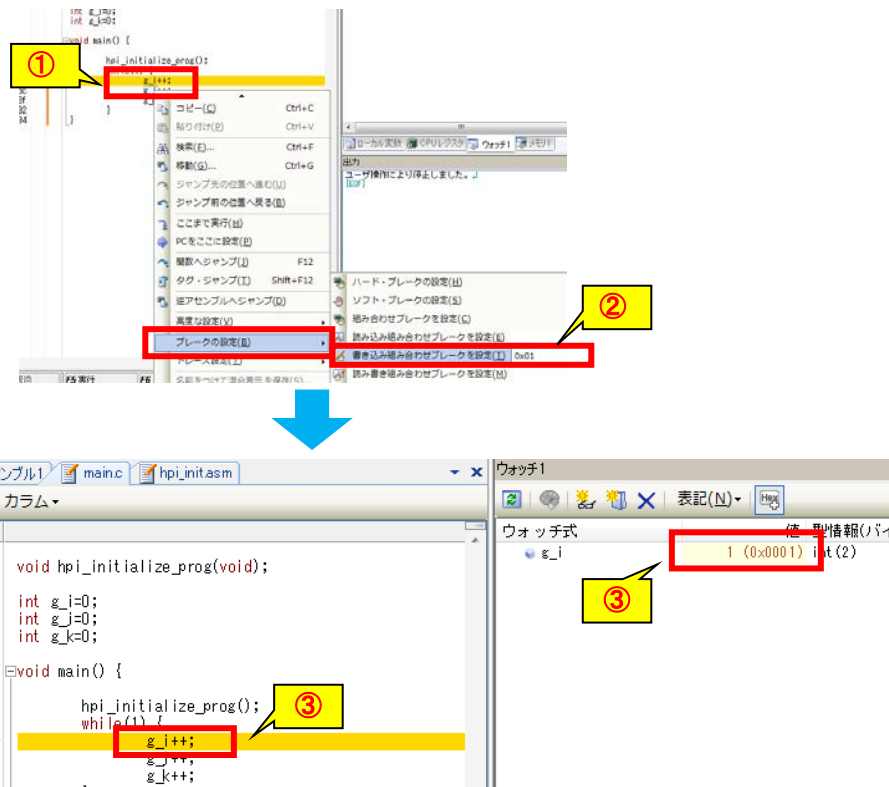
プログラム実行をした際、設定個所でブレークが発生します。



①ソフトウェアブレークの設定	main.c ソース上にある「g_j++;」行のブレーク設定行をクリックしてください。
②ソフトウェアブレークの確認	プログラムを実行しソフトウェアブレークが発生することを確認します。

・アクセスブレーク

変数に特定の値を Write した場合、Read した場合、Write/Read した場合にブレークすることができる機能です。



①コンテキストメニューを開く	main.c ソース上で「g_i」をドラックし、右クリックしてください。
②アクセスブレークの設定	「ブレークの設定」→「書き込みみ合わせブレークを設定」でブレークする値を指定します。任意の指定が可能です。ここでは「0x01」を指定してください。
③アクセスブレークの確認	アクセスブレークが発生するとプログラムが止まりアクセスした場所と値が示されます。

2.2.6 ユーザシステムとの切断

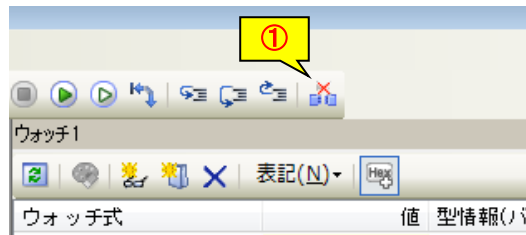
(1) デバッグツールの切断

デバッグツールを切断してください。

※1 切断する場合は、プログラムを停止する必要があります。

※2 切断時にマイコンのリセットが発生します。デバッグツール切断後、エミュレータ接続中はリセット状態が継続します。

※3 切断後、再度ホットプラグインする場合は、E1 をユーザシステムから外してマイコンを一度リセットしてください。



①デバッグツールの切断

プログラムをブレークさせ、メニューにある「切断」ボタンを押下してください。

(2) ユーザシステムの立ち下げ

- ・ユーザシステムの電源を落としてください。

(3) E1 エミュレータの取り外し

- ・E1 エミュレータとホスト PC 間の USB ケーブルを取り外します。
- ・その後、E1 エミュレータとユーザシステムを取り外してください。

(4) CS+の終了

- ・CS+を終了してください。

3. 注意事項

3.1 デバッグ DTC 動作クロックに関する注意事項

デバッグ DTC の動作クロック供給設定は、通常の DTC と同じ DTCGEN ビットで行います。このため、ホットプラグインおよび DTC 方式 RRM/DMM を使用する際に、DTCGEN ビットを 0 に設定しないようにしてください。


3.2 DTC の保留命令に関する注意事項

ホットプラグインおよび DTC 方式 RRM/DMM は DTC を使用して実現しているため、DTC 保留命令が連続する場合は DTC の起動が保留されます。DTC の起動が保留され続けるとホットプラグイン接続や RRM/DMM に失敗します。

[DTC 保留命令]

- a) 無条件分岐命令
 - b) コールリターン命令
 - c) 条件付分岐命令
 - d) コードフラッシュ領域へのリードアクセス命令
 - e) IFxx、MKxx、PRxx、PSW へのビット操作命令とオペランドに ES レジスタを含んだ 8 ビット操作命令
- 特に C 言語で無限ループ処理を行う場合、以下のようにアセンブラ展開されるので注意が必要です。


```
void main (void){
    while(1){}
}
```



```
_main:
    br main
```

このような場合は無限ループ内に nop 命令を使用するなどして該当命令が連続しないようにしてください。

```
void main (void){
    while(1){
        NOP();
    }
}
```



```
_main:
    nop
    br main
```

3.3 データアクセスイベントに関する注意事項

DTC 方式 RRM/DMM を使用する場合、RRM/DMM 対象の変数や SFR に対してデータアクセスイベントが設定されていると RRM/DMM のアクセスに対して以下のイベントが発生します。

- ・ イベントブレイク (シーケンシャル含む)
- ・ トレース開始/終了イベント

データアクセスイベントを変数/SFR に設定する場合は、同じ変数/SFR に対して DTC 方式 RRM/DMM を使用しないでください。

3.4 32bit 長以上の変数アクセスに関する注意事項

DTC 方式 RRM/DMM で使用している DTC のデータサイズは最大 16bit です。

このため、32bit 長以上の変数を読み出し/書き込みと、CPU による変数への書き込みが競合すると、誤った読み出し値や誤った書き込み値となることがあります。

例えば 32bit 長の変数を読み出す場合、上位 16bit と下位 16bit の読み出しの間に CPU によって変数に書き込みが行われると、上位 16bit が書き込み前データで下位 16bit が書き込み後データとなった値を読み出します。

3.5 スタンバイモードに関する注意事項

スタンバイモードに関して以下の注意事項があります。

- ・ホットプラグイン接続前

接続前に MCU がスタンバイモードに入っていて、低速オンチップオシレータと DTC の動作クロックのいずれかが停止している場合、ホットプラグイン接続ができません。

スタンバイモードに入っていることが原因でホットプラグイン接続できない場合はリトライ設定を活用してください。

- ・ホットプラグイン接続後

ホットプラグイン接続後に STOP モードに入った場合、かつ DTC の動作クロックが高速オンチップオシレータでない場合、STOP モード中は DTC 方式 RRM/DMM を保留します。(STOP モード復帰後に再開します。)

なお、デバッグはリトライ設定の「リトライ間隔」で指定した時間を DTC 方式 RRM/DMM の最大保留時間として扱っています。(リトライ間隔分 STOP モードから復帰しない場合、エラーとなります。)

3.6 リセットに関する注意事項

リセットに関して以下の注意事項があります。

- ・プログラム停止中の POR 回路によるリセット

ホットプラグイン後、プログラム停止中に POR 回路によるリセットが発生すると、リセット発生から 900ms 程度ユーザプログラムが実行後に停止します。

※ホットプラグイン用初期化プログラム実行完了後に停止可能となるため。

- ・プログラム実行中の POR 回路によるリセットと端子リセット

ホットプラグイン後、プログラム実行中に POR 回路によるリセット、もしくは端子リセットが発生すると以下の設定が無効になります。

-ハードウェアブレーク設定(イベント無効)

-トレース機能(イベント無効、トレース OFF)

-入力信号のマスク設定(マスク設定が全て OFF)

また、ソフトウェアブレーク使用時に POR 回路によるリセット、もしくは端子リセットが発生し、再度ホットプラグイン接続するまでの間にソフトウェアブレークを設定した命令を実行すると 0xFF コード実行によるリセットが発生します。

なお、端子リセットに関しては、端子リセットのマスク機能をホットプラグイン接続してから強制ブレーク後に使用することができます。このため、上記の動作が問題となる場合は端子リセットマスク機能を使用してください。

- ・プログラム実行中の内部リセット(POR 回路によるリセットと端子リセットを除く)

ホットプラグイン後、プログラム実行中に内部リセットが発生すると DTC 方式 RRM/DMM が中断されます。この場合、リトライ間隔として設定された時間後に DTC 方式 RRM/DMM を再度開始します。

3.7 RAM の使用に関する注意事項

デバイス品種によってはホットプラグイン接続やDTC方式RRM/DMM、トレース機能を使用するためにRAM領域確保が必要なものがあります。詳細は、各デバイスのユーザーズマニュアルハードウェア編（オンチップ・トレース）をご参照ください。

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<https://www.renesas.com>

お問合せ先

<https://www.renesas.com/contact>

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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2013.03.29	—	初版発行
2.00	2013.08.20	4	「ホットプラグイン接続後のデバッグ」追加
		9	ホットプラグイン初期化関数を一部変更 「初期化関数の使用方法」を追加
		13	「オンチップデバッグオプションバイト設定について」を追加
		18	ホットプラグインのリトライ設定に関する記載を追加
		20	「ホットプラグイン接続時に使用できない機能」を追加
		24	「ユーザシステムとの切断」を見直し
		25	「注意事項」を追加
3.00	2016.08.26	—	CubeSuite+をCS+に変更
		1	RL78/F15、RL78/F1Aを追加
		10	CS+ for CC用のホットプラグイン初期化関数を追加

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレス（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違っていると、内部ROM、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

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て、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事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。
総合お問合せ窓口：<http://japan.renesas.com/contact/>