プログラムをＲＡＭ上でデバッグする場合の説明

Rev1.50
DEF バージョン 6.30A 仕様より
DEF バージョン 7.10A 仕様より

【対象ＣＰＵ】

1）H8/300H、H8S シリーズ、H8SX シリーズ、SH-2 シリーズが対象になります。

【機能】

1）BSC（バスステートコントローラ）による拡張 RAM でのデバッグに対応しました。
2）PBC/UBC 無しタイプの CPU 品種でもプログラムメモリがＲＡＭの場合、C ソース／Asn ソース上に直接ソフトブレークが張れます。
3）内蔵 RAM 利用の場合は、BSC 設定の準備は不要になります。

【デバッグ開始前の準備】

1）BSC（バスステートコントローラ）設定のスクリプトファイルを作成する。

例）ファイル名＜H83069-BSC.log＞

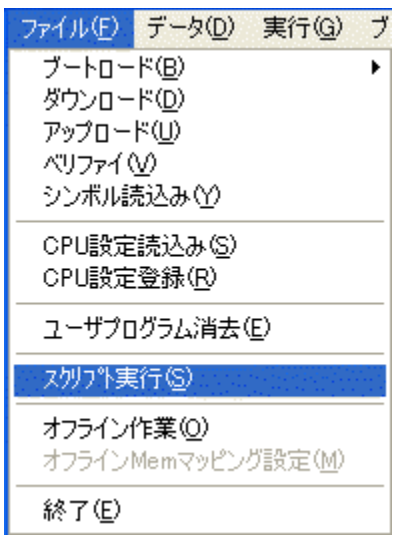
```
//  H8 用(H8/3069F)バスステートコントローラ初期設定
//        エリア 2:SRAM 256Kb 16bit 0x400000
//  コメントは、コマンド実行ラインに記述しないで下さい。

// バス幅コントロールレジスタ  CS2 エリア:16bit
<S  ABWCR  0xfb
// ポート 1 データディレクションレジスタ  A7,A6,A5,A4,A3,A2,A2,A0
<S  P1DDR  0xff
// ポート 2 データディレクションレジスタ  A15,A14,A13,A12,A11,A10,A9,A8
<S  P2DDR  0xff
// ポート 5 データディレクションレジスタ  A19,A18,A17,A16
<S  P5DDR  0xf
// ポート 8 データディレクションレジスタ  CS2 出力端子
<S  P8DDR  0x4
```

★CPU 品種用スクリプトファイル例は、ホームページで公開しています。

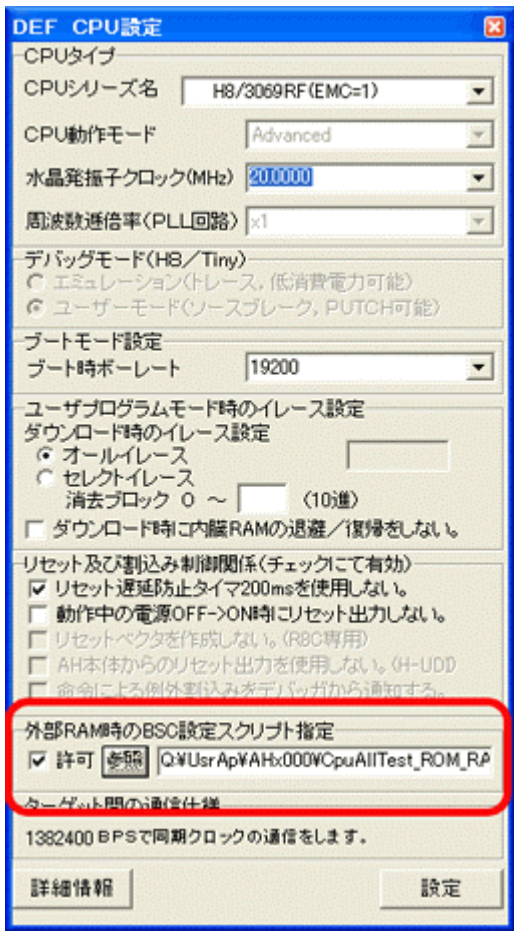
<S { 8 ビットアクセス} {レジスタ名} {データ}
<SS {16 ビットアクセス}
<SL {32 ビットアクセス}
<SQ {8～32 ビットアクセス} ＜－ 内部登録されているシンボルタイプ（ﾋﾞｯﾄ長）を使用する
// コメント行
注意 コマンド行には、コメント記述をしないで下さい。

2）作成したスクリプトファイルを実行させ確認をする。＜ファイルメニュ＞



3）スクリプトファイル実行後、拡張 RAM が正しく読み書き出来るか「メモリフィル」等で確認する。

4）作成したスクリプトファイルを CPU 設定に登録する。



許可を「チェックレ」後、「参照」PB を押下し、作成したスクリプトファイルを登録する。

5）スクリプトファイル登録による効果
・ユーザプログラムのダウンロード時の、開始と終了後に登録された内容を実行します。
・【RstMon】と【Reset】を実施後、登録された内容を実行します。

6）スタートアップ関数に「BSC 設定プログラム」を登録する。

```
例) ファイル名＜startupE1.c＞  GNU/gcc
//      プロトタイプ宣言
void StartUp(void)         __attribute__ ((section (".startup")));
void SoftWait(Ushort ms)    __attribute__ ((section (".startup")));
void Wait1ms()             __attribute__ ((section (".startup")));

void    StartUp()
{
    asm("mov.l #0xffff1e,sp");          // ｽﾀｯｸﾎﾟｲﾝﾀ 設定
    BCR     |= 0x4;                     // BCR   EMC=1(3069 特有)
//
//  H8 用(H8/3069F)バスステートコントローラ初期設定
//       エリア 2:SRAM 256Kb 16bit 0x400000
//
// バス幅コントロールレジスタ  CS2 エリア:16bit
    ABWCR = 0xfb;
// ポート 1 データディレクションレジスタ  A7,A6,A5,A4,A3,A2,A2,A0
    P1DDR = 0xff;
// ポート 2 データディレクションレジスタ  A15,A14,A13,A12,A11,A10,A9,A8
    P2DDR = 0xff;
// ポート 5 データディレクションレジスタ  A19,A18,A17,A16
   P5DDR = 0xf;
// ポート 8 データディレクションレジスタ  CS2 出力端子
   P8DDR = 0x4;

   SoftWait(1);    // ←――ポイント２ 1ms Wait(ブート I/F の場合必要-Reset 解除時ソフトタイマ推奨タイプ)
   main();
}
```

例) ファイル名＜resetprg.c＞　ルネサスC

```
#pragma section ResetPRG
void    SoftWait(short ms);
void    Wait1ms(void);
__entry(vect=0) void PowerON_Reset(void)
{
//  set_imask_ccr((_UBYTE)1);
//
//  H8用(H8/3069F)バスステートコントローラ初期設定
//       エリア2:SRAM 256Kb 16bit 0x400000
//
    BSC.BCR.BIT.EMC = 1;            // BCR   EMC=1
// バス幅コントロールレジスタ  CS2エリア:16bit
    BSC.ABWCR.BYTE = 0xfb;
// ポート1データディレクションレジスタ  A7,A6,A5,A4,A3,A2,A2,A0
    P1DDR = 0xff;
// ポート2データディレクションレジスタ  A15,A14,A13,A12,A11,A10,A9,A8
    P2DDR = 0xff;
// ポート5データディレクションレジスタ  A19,A18,A17,A16
    P5DDR = 0xf;
// ポート8データディレクションレジスタ  CS2出力端子
    P8DDR = 0x4;

    _INITSCT();   // <――ポイント１

//  _CALL_INIT();           // Remove the comment when you use global class object
//  _INIT_IOLIB();          // Remove the comment when you use SIM I/O
//  errno=0;               // Remove the comment when you use errno
//  srand((_UINT)1);        // Remove the comment when you use rand()
//  _s1ptr=NULL;            // Remove the comment when you use strtok()
//  HardwareSetup();        // Remove the comment when you use Hardware Setup
//  set_imask_ccr((_UBYTE)0);

    SoftWait(1);  // <――ポイント２ 1ms Wait(ブートI/Fの場合必要-Reset解除時ソフトタイマ推奨タイプ)
    main();

//  _CLOSEALL();            // Remove the comment when you use SIM I/O
//  _CALL_END();            // Remove the comment when you use global class object
    sleep();
}
```

ポイント１　「_INITSCT()」は、Dセグメントから Rセグメントにコピーする処理が入っています。Dセグメントを外部拡張RAMMに配置している場合は、この関数処理前にBSC（バスステートコントローラ）の初期化が必要です。

ポイント２　ブートI/FのCPU品種の場合、リセット解除後NMIを起動するまでCPUは走行します。「main()」へ飛ぶまではROM上プログラムであることが望ましいので、ソフトタイマでNMI起動までの時間を調整します。
リセット解除の遅延はハードに依存しますので、「SoftWait(n)」で調整して下さい。
ダウンロード後も　リセット－＞NMI　のシーケンスを実行しますので、**NMI起動まではROM走行が必要ですので必ず「SoftWait(n)」**を入れて下さい。

【ソフトタイマが必要な理由】

１．ダウンロード前でRAMエリアが不定な場合、RAMエリアに走行してしまい暴走となり、モニタとの通信に必要なI/O等が書き換わり正常通信ができなくなり二度と立ち上げることができなくなる可能性があるためです。

２．初期段階ではプログラムのバグによりＲＡＭのプログラムエリアが書き換る可能性があるためです。

★CPU品種用スタートアップ関数例は、ホームページで公開しています。

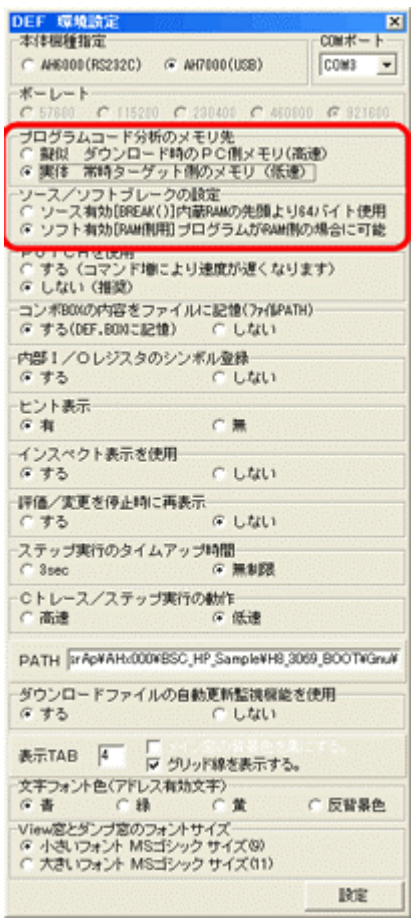
備考

・スタートアップ関数は、ROM側に配置する必要があります。(SoftWait()関数も必要)
・Hewの場合は、「resetprg.c」内に記述すれば良いかと思います。
・Hewにおいて、「_INITSCT()」等のライブラリーを使用している場合、RAM側に配置したプログラムの変更・追加をすることにより、ライブラリー配置が変更されてしまい、「resetprg.c」のプログラムが固定されない場合があります。対策としては、後記のようにライブラリーアドレスを固定化する方法があります。

【ＨｏｗＴｏ】

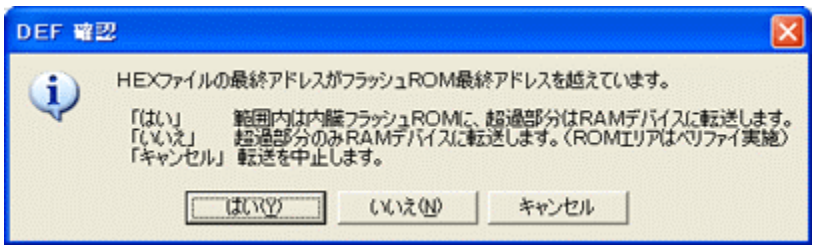
1）プログラムデバッグの初期段階で暴走等の原因により、プログラムのRAMエリアを書き換えてしまうバグが潜んでいる可能性がある場合は【環境設定】の「プログラムコード分析のメモリ先」を「実体」側に指定して下さい。(安定するまで)

2）ソフトブレークを有効にする場合は【環境設定】の「ソース／ソフトブレーク設定」を「ソフト有効」側に指定して下さい。



【ダウンロード説明】

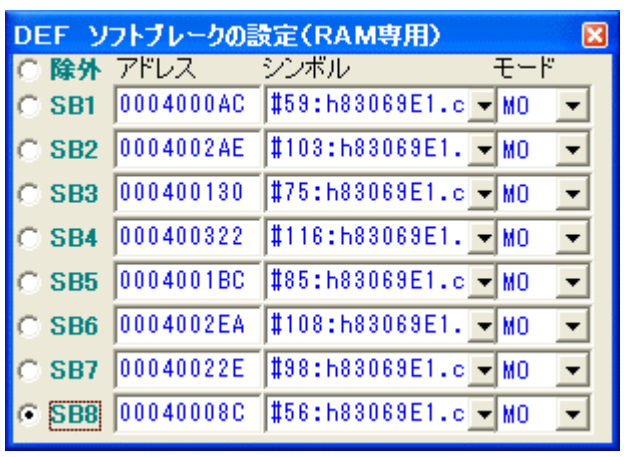
1）外部拡張RAMに配置したプログラムの【ダウンロード】を実施しますと、下記問い合わせが表示されます。



・【はい】　　フラッシュROMとRAMエリアに転送します。
・【いいえ】　　フラッシュROMエリアは「ベリファイ」を実施し、RAMエリアのみに転送します。
・【キャンセル】ダウロードを中止します。

【ソフトブレーク設定】

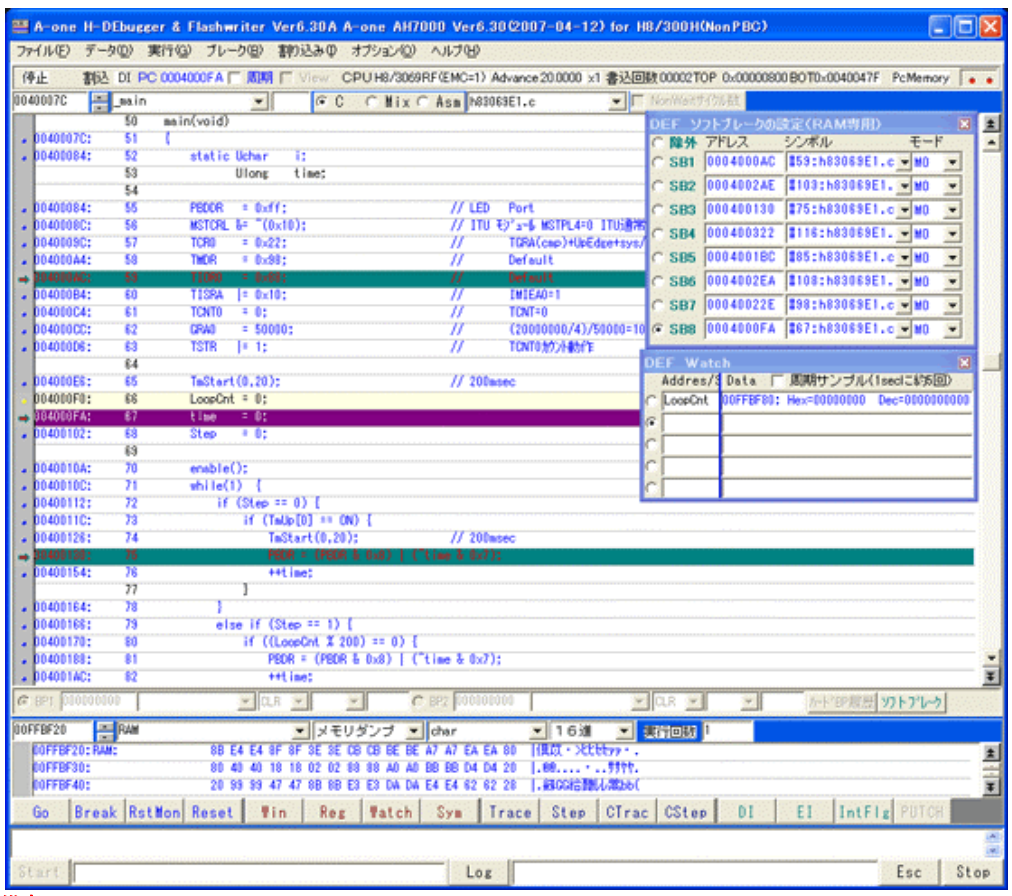
1）RAMに配置したプログラムのみにソフトブレークを直接設定が出来ます。



・「SB1->SB8」を選択してから、CViewの画面上で「ダブルクリック」しますとソフトブレークの設定が出来ます。
・「除外」を選択しますと、CViewの画面上で「ダブルクリック」がソフトブレーク設定から除外されます。ハードブレークを設定する場合に選択して下さい。

【ソフトブレーク設定画面例】

1）DEF にて「ソフトブレーク設定」をした画面です。



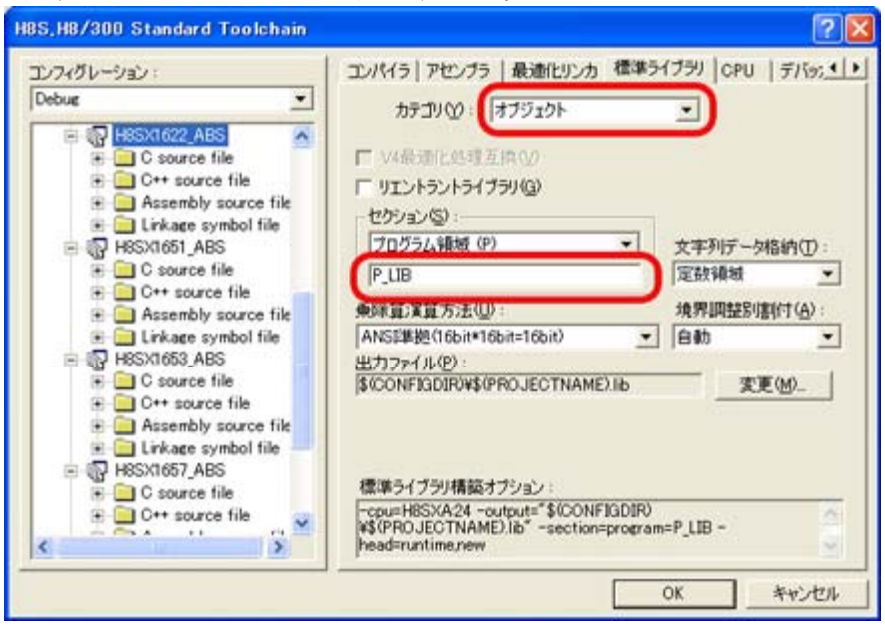
備考

・PBC 無しタイプの CPU の場合でも、プログラムが RAM での実行時は【Trace/Step】が可能になります。

・プログラムを内蔵 RAM 側に配置した場合でも同じく機能します。（この場合は BSC 設定は不要です）

【Ｈｅｗ４　ライブラリーアドレスの固定化】

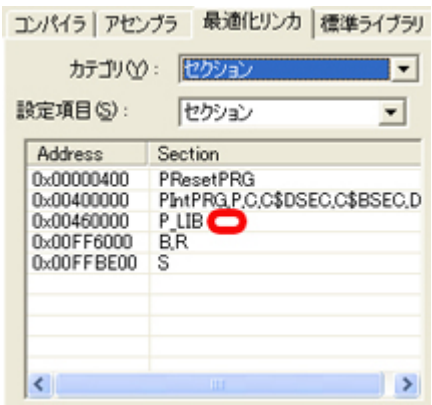
1）標準ライブラリーのセクション名を定義します。



＜標準ライブラリ＞
カテゴリー：オブジェクトを選択します。

セクション名を定義します。（例　P_LIB）

2）定義したセクション名をアドレス割付します。



このように「標準ライブラリー」の開始アドレスを固定化します。

以上