

## ハードウェアブレイク(PBC)機能なし MCU 用ソースブレイクの説明

### 1. ソースブレイクとは

Cソースに直接ブレイク文「SBREAK0」を記述し、実行時その記述文に到達した場合ブレイクする方式を「ソースブレイク」と称します。

なぜ、ソースブレイク方式にしたかの理由は、この方式で無いソフトブレイクを実現させるには、1回のブレイクと実行の前後に2～3回の命令を書き換える必要があります。

プログラムエリアが「RAM」の場合は問題がないのですが、フラッシュROMの場合は、書き込み回数の制限、書き込みブロック単位（1～64Kバイト）、書き込みスピード、等の問題があるため、解決策として、必要個所に最初からブレイク文を記述しておき、そのブレイク文を有効（停止）／無効（通過）させる方式を採用しました。

### 2. ソースブレイクの原理

ソースコードにブレイク文「H8[trap #3]/SH[trapa #0x3f]」を直接実装させ、トラップ割り込み分析をモニタ側で処理し、有効時はプログラムを停止する。また、無効（通過）時は、プログラムは続行する。

#### 2-1) Cソースに記述するためのdefine文

H8/H8SX	GNU-C/IAR-C	#define SBREAK0 asm("trapa #3")
	RenesasC	#define SBREAK0 (trapa(3))
SH-2	GNU-C/IAR-C	#define SBREAK0 asm("trapa #0x3f"); ¥ asm("nop"); asm("nop"); asm("nop");
	RenesasC	#define SBREAK0 trapa(0x3f);nop0;nop0;nop0;

### 3. ソースブレイクを許可するためのDEF8K側の設定

#### <オプション>—<CPU設定>

モニタワークエリアの場所【○：設定可 △：条件あり ×：設定不可】		
スタック：size(46)		○

#### <オプション>—<環境設定>

✓	ソース(RAM)有効[SBREAK0]を使用 内蔵RAMは64byte使用
	ソフト(RAM)有効[SBREAK0]を使用しない

## 4. ソースブレークの記述例

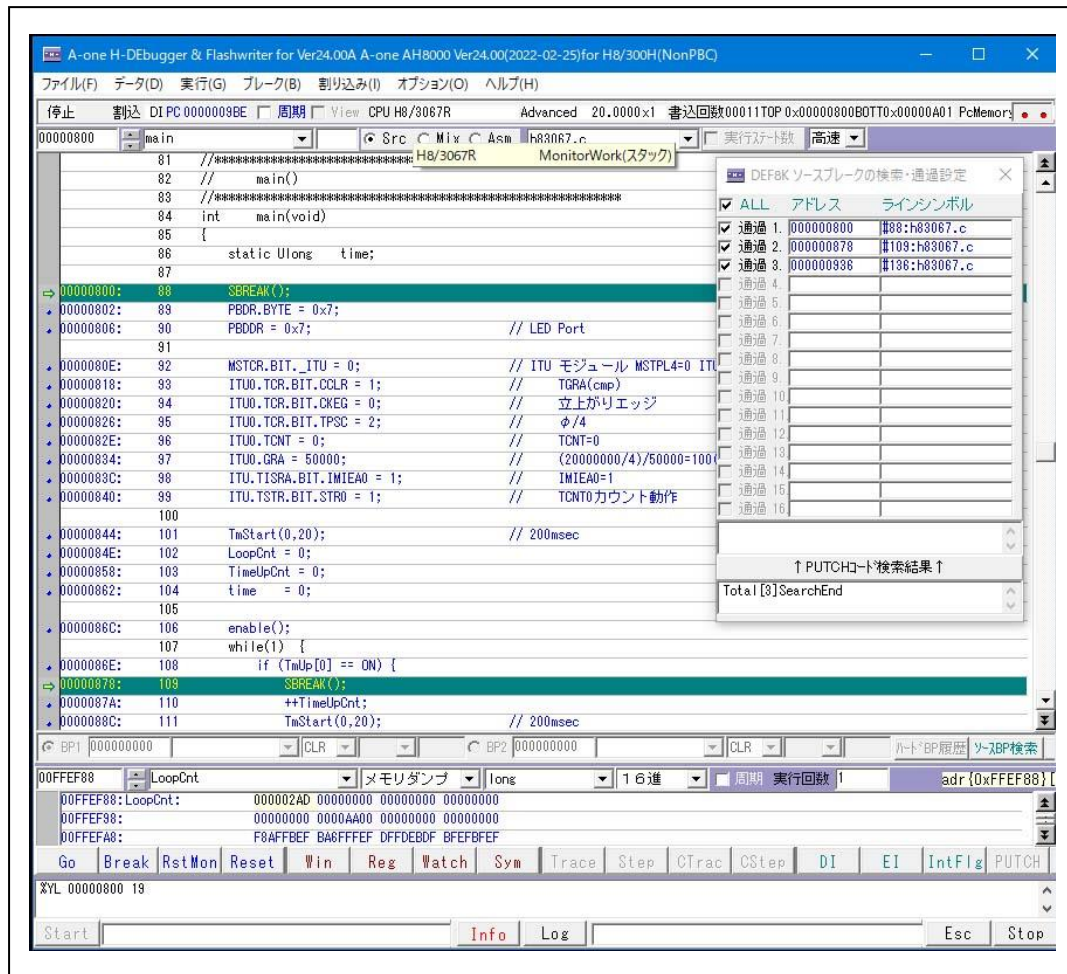
```
/**
// デバック用インライン関数
// SBREAK0を有効にする
// #define DEBUG
//
// SBREAK0を無効にする
// //#define DEBUG<-----コメントにする
//
// 【注意事項】
// ソースブレークを使用する場合は、
// bss(RAM)の先頭アドレス TOP にブレークアドレスを保存
// するため bss(RAM)の先頭アドレスを+0x40 する必要あり。
// また、DEF8K の環境設定の「ソース/ソフトブレーク設定」
// ソース(ROM)有効[SBREAK0を使用する]側にチェックする。
//
/**
#define DEBUG
#ifdef DEBUG
    #define SBREAK0 asm("trapa #3") //H8/H8S/H8SX 用 ソースブレーク
#else
    #define SBREAK0
#endif
/**
// main()
/**
int main(void)
{
    static Ulong time;

    SBREAK0;
    PBDR.BYTE = 0x7;
    PBDDR = 0x7;          //LED Port

    while(1) {
        if(TmUp[0] == ON){
            SBREAK0;
            ++TimeUpCnt;
            TmStart(0,20);    // 200msec
            if(++time >= 50){    // 10sec に 1 回
                time = 0;
            }
        }
    }
}
```

## 5. DEF8K の使用例

### 5-1) 全体表示例

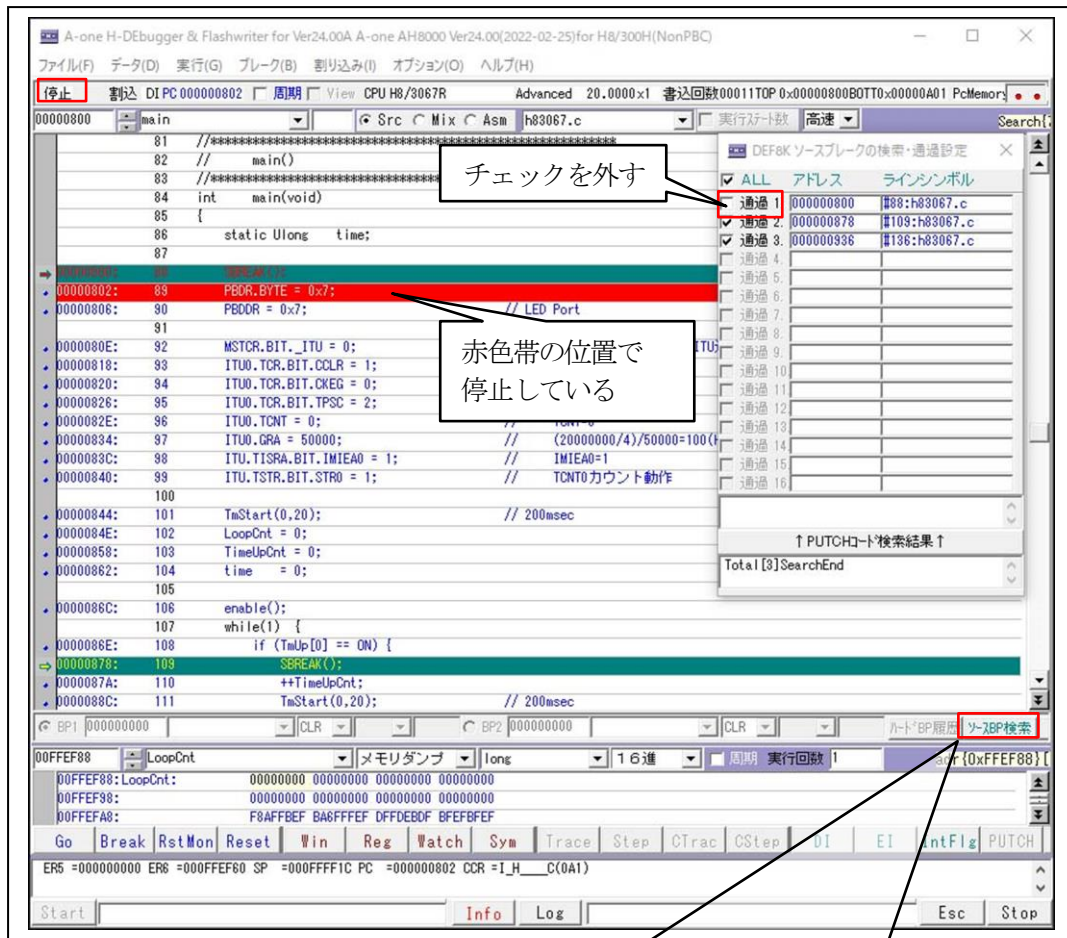


- 青緑網掛け・黄色文字： ソースブレークの通過指定を意味します。  
 青緑網掛け・赤色文字： ソースブレークの停止指定を意味します。  
 茶色網掛け・白色文字： 16ポイント超過分のソースブレークポイントを意味します。  
 このポイントは、通過指定はできません。

### 5-2) ソースブレークの制限事項

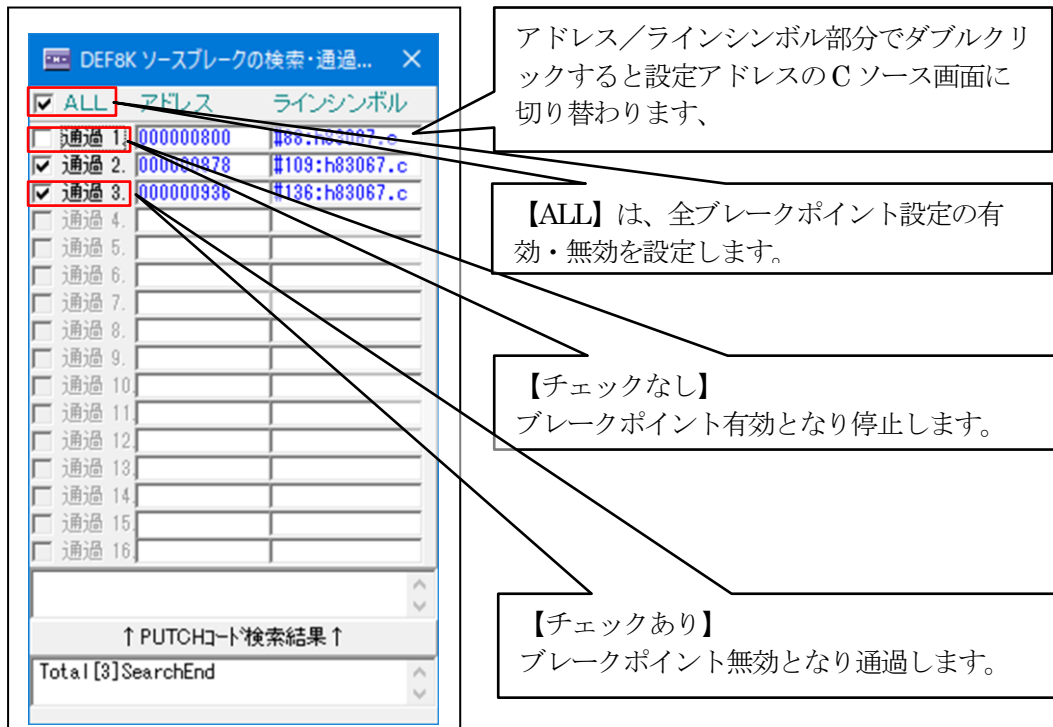
- ・C言語記述のみです。
- ・通過設定のために、ターゲット側の内蔵RAMを64バイト使用します。
- ・管理できるポイントは、16ポイントです。
- ・通過判定の処理時間に約5～15μs必要です。
- ・モニタワークエリアは「スタック方式」に限ります。

### 5-3) ブレークを有効にした場合の停止表示例



【ソース BP 検索】PB クリックにより「ソースブレークの検索・通過設定」が開く

### 5-4) ソースブレークの検索・通過設定の操作



## 6. 注意事項

- ・本文書の著作権は、エーワン（株）が保有します。
- ・本文書を無断での転載は一切禁止します。
- ・本文書に記載されている内容についての質問やサポートはお受けすることが出来ません。
- ・本文章に関して、ルネサス エレクトロニクス社への問い合わせは御遠慮願います。
- ・本文書の内容に従い、使用した結果、損害が発生しても、弊社では一切の責任を負わないもの  
とします。
- ・本文書の内容に関して、万全を期して作成しましたが、ご不審な点、誤りなどの点がありまし  
たら弊社までご連絡くだされば幸いです。
- ・本文書の内容は、予告なしに変更されることがあります。

〒486-0852

愛知県春日井市下市場町 6-9-20

エーワン株式会社

<https://www.aone.co.jp>

