

# 純正開発環境 HEW で 使える無償デバッグ環境

—— 専用ハードウェア不要ですぐ試せる

本章では、USB に接続するだけで純正開発環境 HEW から操作できる無償デバッグ環境の構築方法を解説します。

## 無償で使える H8 マイコンのデバッグ環境

H8SX/1655 マイコンの無償で使えるデバッグ環境は、表 1 の三つです。これらのデバッグ環境は専用のハードウェアが不要なので手軽に試せます。

統合開発環境に Eclipse を選択するとデバッグ・ソフトウェアは GDB と GDB STUB を使います。HEW を選択するとシリアル・デバッガとシリアル・モニタ（以降、モニタ）を使います。

本章ではメーカー純正統合開発環境 HEW を使ったデバッグ環境②と③の作り方と使い方を紹介します。パソコンの統合開発環境 HEW にシリアル・デバッガを追加し、マイコンにモニタを書き込みます。

②と③はコンパイラが異なりますが、HEW で GCC を使えるようにする方法は第 22 章で紹介します。

## HEW 用デバッグ環境の構築

### ● 手順 1：無償評価版 HEW をインストール

まず第 3 章で紹介している方法で無償評価版の純正統合開発環境 HEW をインストールします。ただし、2010 年 7 月時点で 64 ビット版 Windows 7 ではシリ

アル・デバッガが動作しません。32 ビット版を準備してください。

インストールできたのは「H8SX, H8S, H8 ファミリー用 C/C++ コンパイラ・パッケージ」(以下 C/C++ コンパイラ・パッケージと表す)です。このパッケージでは、図 1 にあるように、

- ① 統合開発環境 HEW (表示されている GUI, エディタなどの共通部分)
- ② H8SX, H8S, H8 ファミリー用 C/C++ コンパイラ, アセンブラ, 最適化リンクなどの実行コード生成ソフトウェア
- ③ H8SX, H8S, H8 ファミリー用シミュレータ
- ④ スタック使用量解析ツールなどのユーティリティがインストールされます。

### ● 手順 2：デバッグ用ソフトウェアの解凍

付属 CD-ROM に収録された以下のファイルをパソコンにコピーして解凍します。

¥CQ¥H8SX\_1655¥Renesas¥ モニタプログラム ¥ H8SX1655\_usb\_monitor.exe

すると図 2 に示すファイルができ、Readme がブラウザで表示されます。

Readme に従って、以下の手順で HEW にデバッグ

表 1 本書で紹介する無償で使える H8 マイコンのデバッグ環境

項番	統合開発環境 (IDE)	C/C++コンパイラ	デバッグ用ソフトウェア		専用ハードウェア	ユーザ・プログラムの書き込み先	解説
			パソコン側	マイコン側			
1	Eclipse	GCC	GDB	GDB STUB	不要	フラッシュ・メモリ	第6章
2	HEW	GCC	シリアル・デバッガ	シリアル・モニタ (モニタ)	不要	内蔵RAM	本章
3	HEW	ルネサスC/C++コンパイラ	シリアル・デバッガ	シリアル・モニタ (モニタ)	不要	内蔵RAM	本章

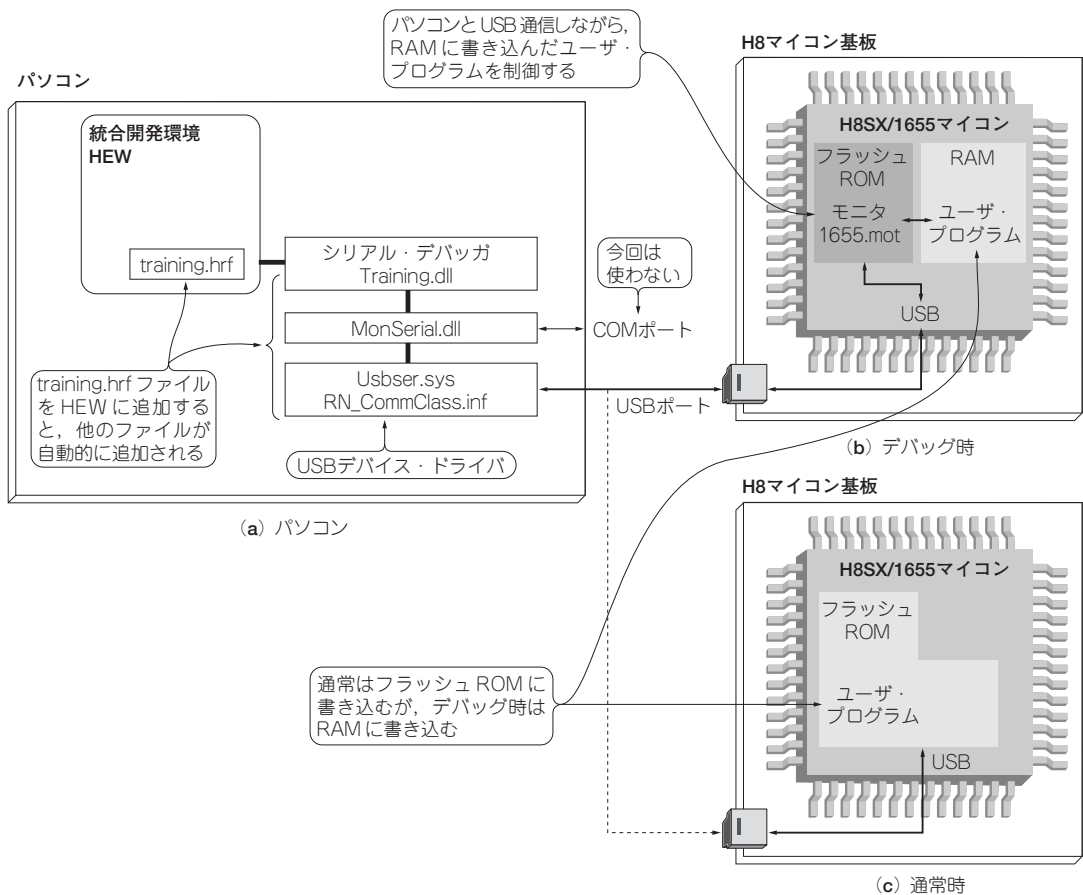


図 1 統合開発環境 HEW を使った無償のデバッグ環境

用ソフトウェアを追加します。

● 手順 3：HEW に登録する DLL のコピー

DLL 本体が格納されているフォルダ ¥dll¥monitor を、HEW がインストールされているフォルダの ¥Tools¥Renesas¥DebugComp¥Platform 内(図 3)にコピーしてください(実はどこにフォルダがあっても問題ないが、ファイルの管理しやすさからここを指定している)。

● 手順 4：HEW にシリアル・デバッガを登録

HEW を起動し、最初に表示されるダイアログで [アドミニストレーション] を選択し training.hrf を [登録] します(図 4, p.183)。登録できると Training Monitor platform が追加されます。

これでパソコン側の準備は完了です。

● 手順 5：マイコン内蔵フラッシュ・メモリにデバッグ用プログラムを書き込む

MB の J<sub>1</sub> をショートし、H8SX/1655 をフラッシュ

書き込みのブート・モードに設定します。

フラッシュ開発ツールキット FDT (Flash Development Toolkit) を使って、デバッグ用モニター・プログラム ¥monitor¥1655.mot を書き込みます。書き込み手順の詳細は「第 1 章 付属基板を動かす」を参照してください。

● 手順 6：パソコンに USB デバイス・ドライバをインストール

手順 5 で書き込んだモニターとやりとりする USB デバイス・ドライバをパソコンに追加します。

MB の J<sub>1</sub> をオープンにし、通常動作モードで起動します。MB とパソコンを USB ケーブルで接続するとパソコンに USB ポートが認識され、図 5 に示すように新しいハードウェアの検索ウィザードが開始されます。¥monitor¥inf フォルダを指定してインストールを完了します。

以上で登録などの準備はすべて完了です。この時点ではユーザー・プログラムはまだマイコンに書き込まれていません。

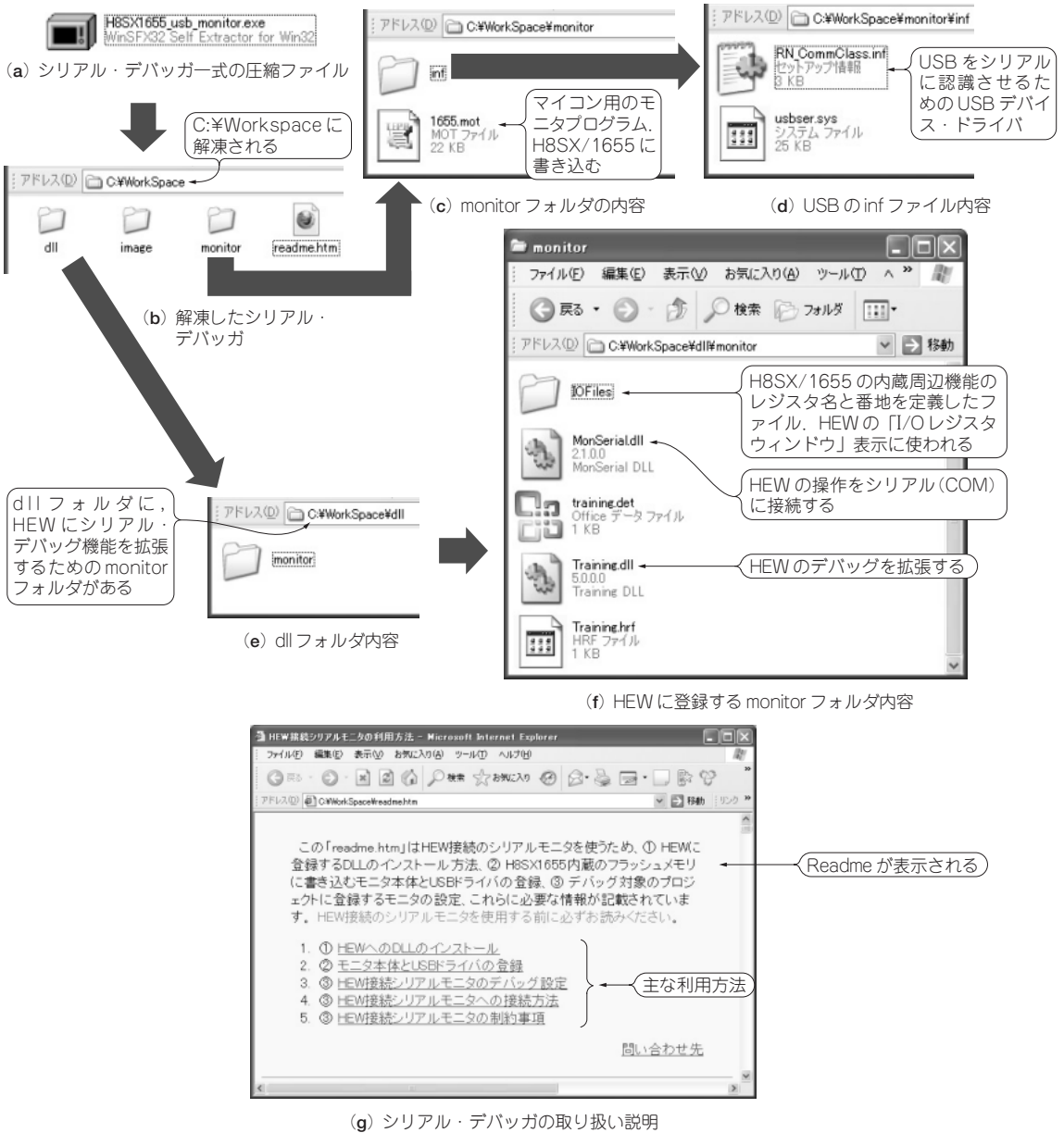


図2 CD-ROM に収録された HEW 用の無償デバッグ・プログラム

(c) のモニタを H8SX/1655 に書き込んで動作させ、(d) の USB デバイス・ドライバを割り当てる。(e) のフォルダの中身を HEW に登録すると使える

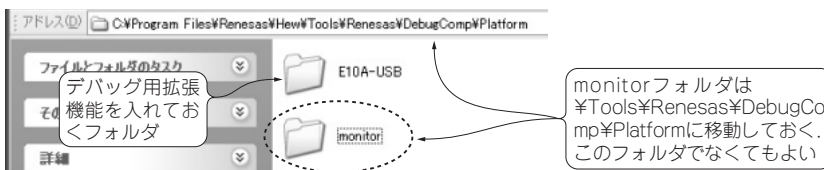
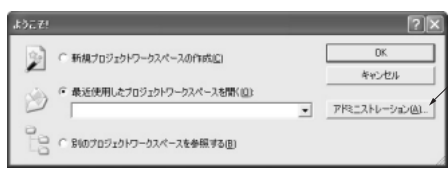
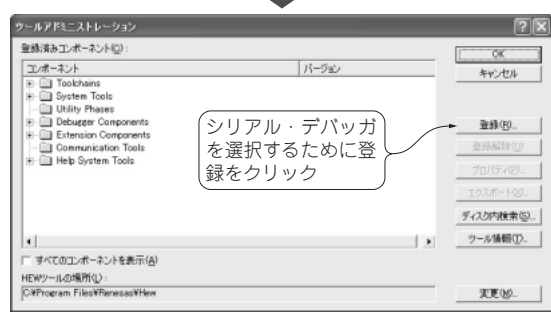


図3 DLL のコピー

HEW に新しい機能を登録 / 削除するときを使う。ここではシリアル・デバッガを登録する



(a) HEW 起動時の「ようこそ！」で「アドミニストレーション」を選択



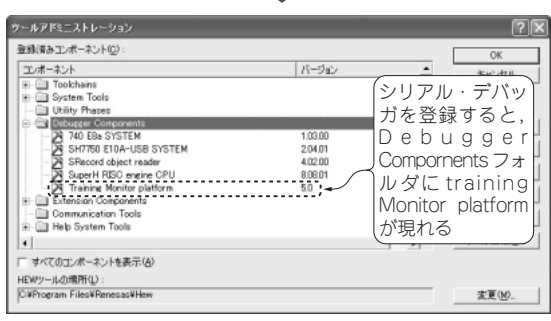
シリアル・デバッガを選択するために登録をクリック

(b) 「登録」ボタンをクリック



monitor フォルダをコピーした位置を指定して training.hrf を選択する

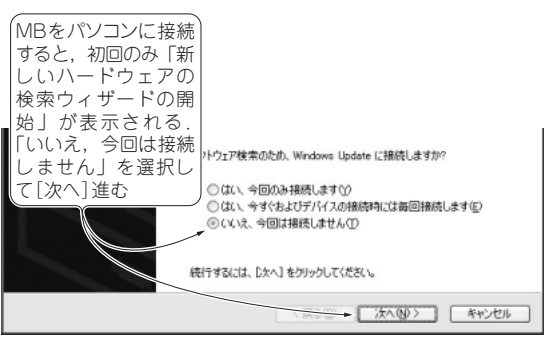
(c) シリアル・デバッガのコンポーネント [training.hrf] を選択



シリアル・デバッガを登録すると、Debugger Components フォルダに training Monitor platform が現れる

(d) シリアル・デバッガを登録した表示

図 4 シリアル・デバッガの HEW への登録



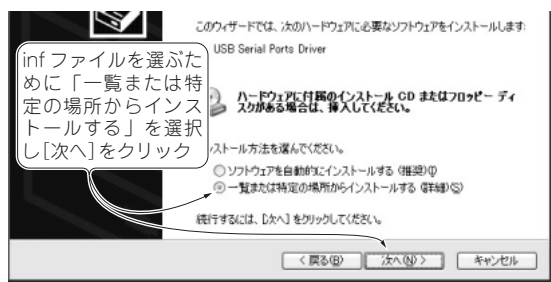
MB をパソコンに接続すると、初回のみ「新しいハードウェアの検索ウィザードの開始」が表示される。「いいえ、今回は接続しません」を選択して「次へ」進む

ソフトウェア検索のため、Windows Update に接続しますか?  
 はい、今のみ接続します  
 はい、今すぐおよびデバイス接続時には毎回接続します  
 いいえ、今回は接続しません

続行するには、「次へ」をクリックしてください。



(a) 新しいハードウェア検索ウィザードが開始



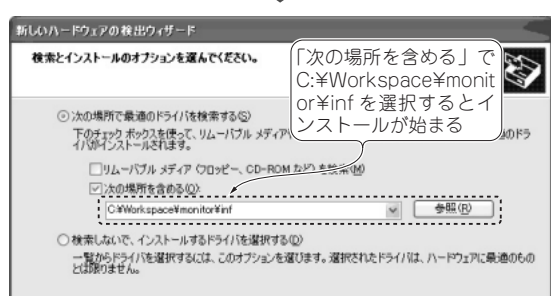
inf ファイルを選ぶために「一覧または特定の場所からインストールする」を選択し「次へ」をクリック

ソフトウェアを自動的にインストールする (推奨)  
 一覧または特定の場所からインストールする (詳細)

続行するには、「次へ」をクリックしてください。



(b) シリアル・デバッガに使う USB



「次の場所を含める」で C:\Workspace\monitor\inf を選択するとインストールが始まる

この場所を含める  
 C:\Workspace\monitor\inf

検索しない、インストールするドライブを選択する  
 一覧からドライブを選択するには、このオプションを選びます。選択されたドライブは、ハードウェアに最適なものと異なります。

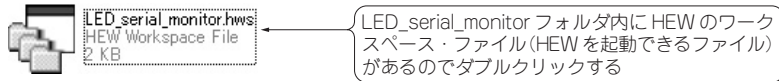
(c) 参照場所を指定

図 5 パソコンへの USB デバイス・ドライバ登録

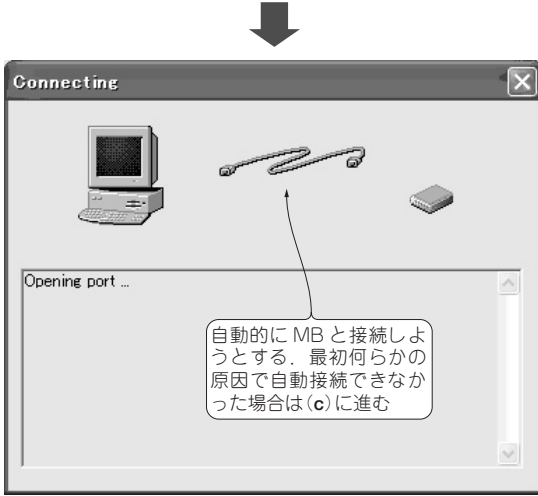
## HEW 用デバッグ環境の使い方

● H8 マイコン基板を USB 接続するだけで準備 OK 試しに LED 点滅プログラムを動かしてみましょう。付属 CD-ROM の add\workspace にある LED\_serial\_monitor フォルダを、C:\workspace フォルダにコピーして使います (図 6)。

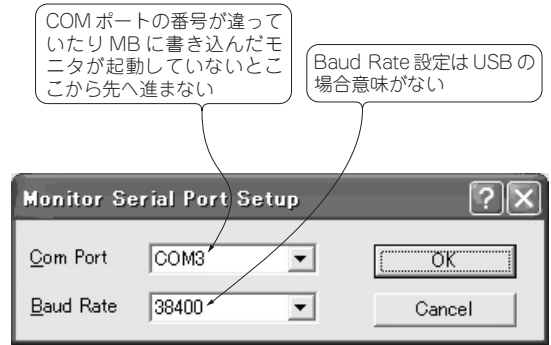
USB に MB を初めて接続すると図 7 のように自動的にモニタへの接続が行われます。もし、自動的に接続が行われない、または再度接続するときは、[接続] アイコンをクリックするか、デバッグ・メニューにある [接続] を実行します。



(a) テスト用プロジェクト LED\_serial\_monitor をダブルクリック



(b) シリアル・デバッガが自動的に起動



(c) COM ポートの選択



(d) デバッガ起動アイコン

図 6 HEW のシリアル・デバッガと H8 マイコン基板 (MB) を接続する

COM ポートの選択から進まない場合は、MB を再起動してパソコンの COM ポート番号を確認し、再実行します。なお、ボー・レートの設定値は USB 接続の場合、意味がありません。接続されると HEW のメニューにデバッグ関係のアイコンが表示されます。

● ユーザ・プログラムは RAM に書き込むのが簡単

MB を USB 接続でデバッグできる環境が整ったので、実際に使ってみましょう。

シリアル・デバッガを利用したデバッグには、作成したユーザ・プログラムをどこに記憶するかによって、二つの方法があります。

- (1) RAM に記憶する
- (2) モニタと共にフラッシュ・メモリに記憶する

RAM に記憶 (ダウンロード) する使い方は手軽なので、その手順を説明します。

ユーザ・プログラムの作成には図 8 に示すベクタ領域と内蔵 RAM の一部 32K バイトを自由に使えます

す。H8 マイコン基板 MB にタッチ・パネル LCD 付き拡張基板 TB を接続すると外部 RAM も使えます。

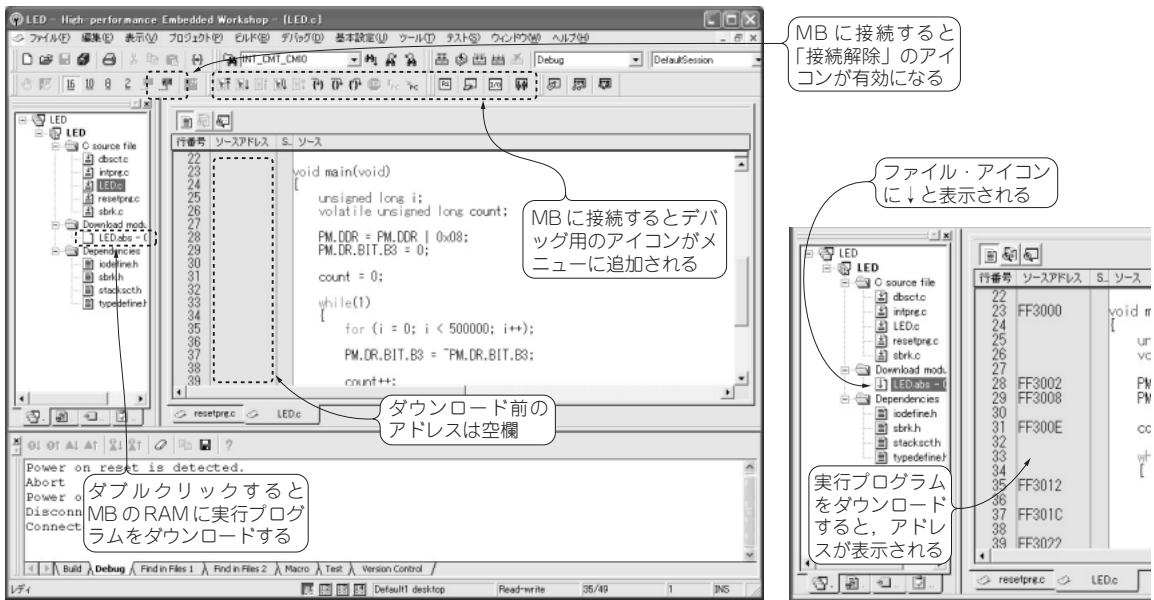
■ 使用上の注意

● その 1 : 関数や変数を RAM に配置しないと暴走する

プログラムのメモリ配置には注意が必要です。HEW の新規プロジェクト作成でプログラムやセクションのメモリ配置を自動で設定すると、プログラムの実行コード (セクション P, PResetPRG, PIntPRG) や固定データ (セクション C, C\$DSEC, C\$BSEC)、初期値ありデータの初期値 (セクション D) がすべてフラッシュ ROM 領域になっています。

メモリ配置状況は図 9(a) (p.187) のように最適化リンカのセクションのオプションで確認できます。シリアル・デバッガを利用するにはダウンロード可能な RAM 領域に変更します。

もし、ダウンロードできない領域を使ったとしても



(a) 起動画面

(b) MBのROMにユーザ・プログラムをダウンロード後の画面



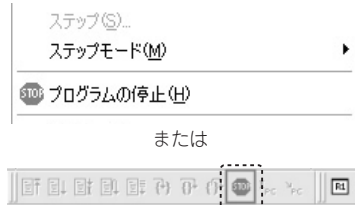
ダウンロード中に表示されるウィンドウ。バーが進まないなどの不具合発生時は、MBをUSBから抜き、シリアル・デバッガの「接続を解除」する

(c) ダウンロード時に不具合が発生したら「接続を解除」して不具合を解消する



H8SX/1655をいったんリセットし、ダウンロードしたりリセット・ベクタを読み込んでから実行を開始する。LEDが点滅する

実行しているプログラムを停止するには「プログラムの停止」または「Stop」アイコンを使う



(d) ダウンロードしたプログラムの実行

(e) プログラムの停止

図7 LED点滅プログラムの実行

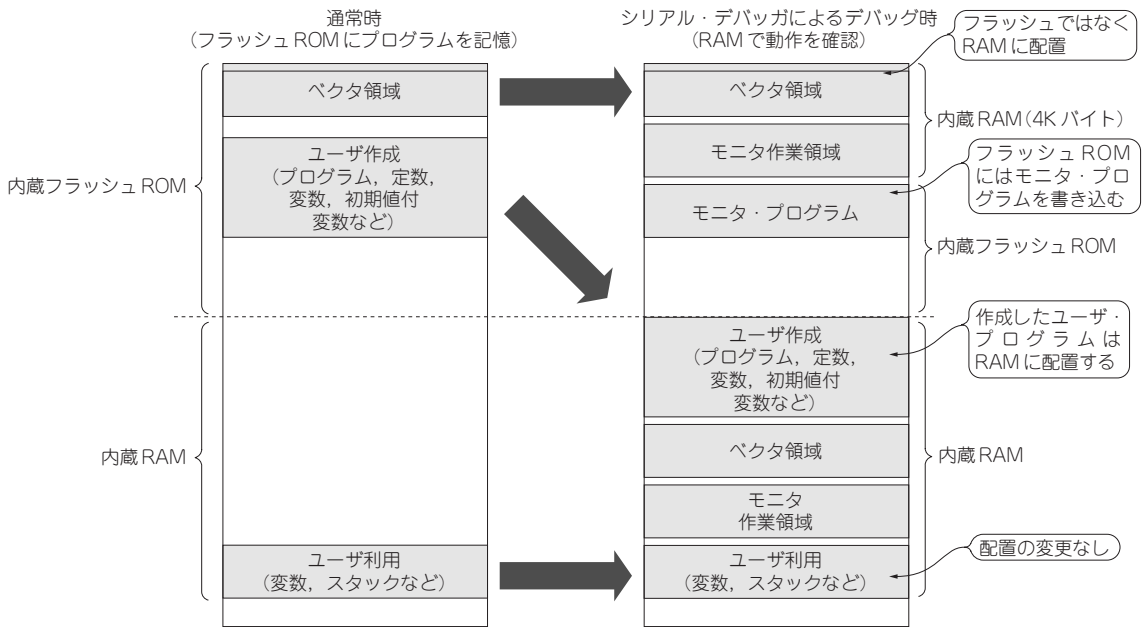
ダウンロード時に警告は発生しませんが、実行するとモニタが暴走する可能性があります。

対策としてプログラムがダウンロード可能な領域に納まったかをバリファイ機能でチェックすることをお

勧めます。図10(p.187)の「バリファイ」オプションに利用できるメモリの範囲を指定するとビルド時にチェックしてくれます。

番 地	メモリ・マップ	容量 [バイト]	ダウンロード
H'000000~ H'0003FF	ベクタ領域 (H'FFFA000~H'FFFA3FFと同じ内容、フラッシュROMの領域だがRAMで代替)	1K	○
H'000400~ H'000FFF	モニタの作業領域 (H'FFFA400~H'FFFAFFFと同じ内容、フラッシュROMの領域だがRAMで代替)	3K	×
H'001000~ H'07FFFF	内蔵フラッシュROM (モニタが10Kバイト程度を使っているが、残りは空いている)	509K	×
H'080000~ H'FF1FFF	外部空間 (TBやSBを接続すると使う。詳細は第9章のColumn図Aを参照)		○
H'FF2000~ H'FF9FFF	内蔵RAM (デバッグしたいプログラムやデータを配置できる領域)	32K	○
H'FFA000~ H'FFAFFF	内蔵RAM (H'0000000~H'000FFFFと同じ内容。ベクタ領域とモニタの作業領域)	4K	○ (ベクタ領域と同じ内容)
H'FFB000~ H'FFBFFF	内蔵RAM (デバッグしたいプログラムやデータを配置できる領域)	4K	○
H'FFC000~ H'FFFFFF	内蔵周辺機能など	16K	×

(a) モニタ動作時のメモリ・マップ



(b) 通常とモニタ動作時の相違

図8 シリアル・デバッガ利用時のメモリ・マップ

## ● その2：デバッグ中にUSBや一部の割り込みが使えない

シリアル・デバッガがH8SX/1655の機能の一部を占有しているために使えないものがあります。プログラムを作成したりデバッグしたりするときに注意しましょう。

まずUSBは使えません。デバッグ用ソフトウェア(シリアル・デバッガとモニタ)がUSB通信を利用しているためです。

割り込み機能は制御モード2に固定です。制御モード0は使えません。また優先レベルも最も高い7は表2に示すようにダウンロードしたプログラムの実行を停止するために使っています。ユーザが利用すると実行停止できなくなる可能性があります。ユーザが利用できる優先レベルは1~6です。

さらに、モニタが利用している表3に示す例外(割り込み)は利用できません。

● その3：パソコンに直接につながないと動かない

シリアル・デバッガはハブに対応していません。パソコンと H8SX/1655 マイコンの接続は外付けのハブを使わずパソコン本体の USB ポート(ルート・ハブ)に接続してください。

外付けハブを経由すると[リセット後実行]を選択しても実行しなかったり、[Stop]で停止を2回選択すると動作し始めたりするなどの現象が発生します。

■ 機能

● 開始番地を指定して実行

図 11 の手順でプログラム・カウンタ(PC)開始番地を設定し実行します。プログラム・カウンタを設定するには二つの方法があります。

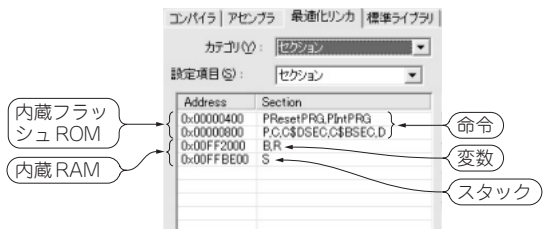
- ① [表示]-[CPU]-[レジスタ]で CPU 内部レジスタを表示し、PC をダブルクリックで編集
- ② ソース・プログラム上で右クリックしメニューを表示し [現在のカーソル位置に PC を設定] を選択

● ブレークポイントの設定・解除

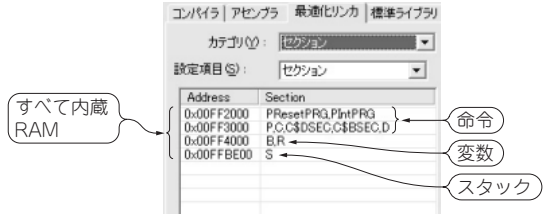
プログラムの実行を目的の位置で停止するには、図 12 のようにブレークポイントを設定します。シリアル・デバッガではソフトウェア・ブレークポイントが使えます。[S/W ブレークポイント]の欄でダブルクリックします。設定できるのは最大4カ所です。これは H8SX/1655 内蔵の UBC(ユーザブレーク・コントローラ)で実現していて、そのチャンネル数が4のためです。

● 内蔵周辺機能レジスタ値の参照・変更

プログラムの動作を調べるときに、ポート端子の状態を確認・変更したいことがあります。このとき威力を発揮するのが図 13 の I/O レジスタの表示・編集機



(a) 初期設定



(b) シリアル・デバッガ利用時

図 9 シリアル・デバッガを使うときのメモリ(セクション)の設定



図 10 利用可能な範囲にあるかをチェックするベリファイ・オプション

表 2 割り込み優先レベルと利用目的

優先レベル	目的
7	モニタがユーザ・プログラム実行停止に占有して利用。ユーザが利用するとプログラム実行の強制停止ができなくなる可能性あり
6~1	ユーザが自由に使える割り込み
0	割り込みとしては機能しない(割り込み要求をDMAC起動要因とするなどの場合に使う)

表 3 モニタが利用しているためユーザが利用できない例外要因

バクタ番号	要因	モニタの用途
4	不当命令	ユーザ・プログラムの暴走検出
5	トレース	ステップ・イン, ステップ・オーバ, ステップ・アウト制御
7	NMI割り込み	強制実行停止
12	CPUアドレス・エラー	ユーザ・プログラムの暴走検出
14	UBC ブレーク割り込み	ブレークポイント, ステップ・イン, ステップ・オーバ, ステップ・アウト制御
234	USBINTN2	ユーザ・プログラム実行の強制停止

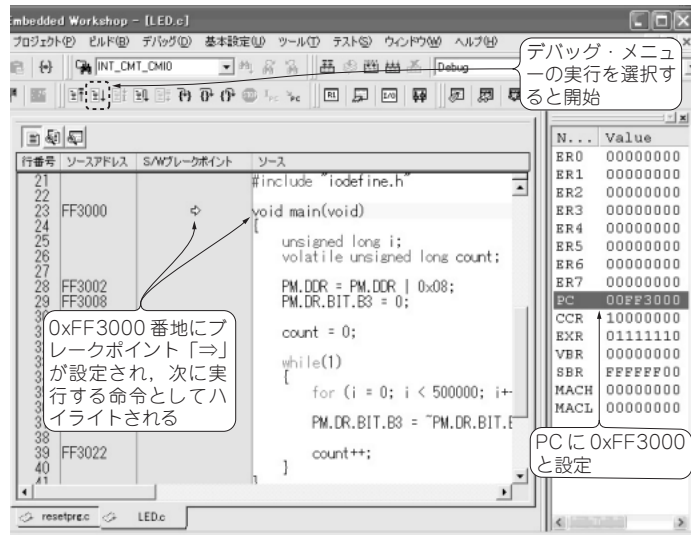




(a) CPU 内部レジスタの表示設定



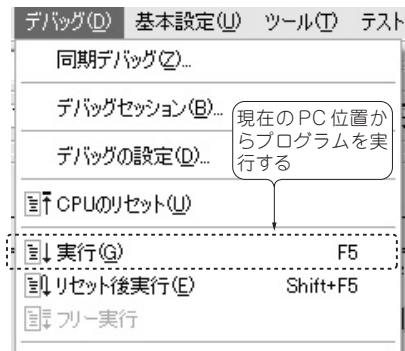
(b) プログラム・カウンタPCを設定



(c) PC 設定後の表示



(d) ソース位置の指定でPCを設定する



(e) 現在のPC位置からプログラムを実行する

図 11 プログラム・カウンタPCの位置を設定しそこから実行

能です。

内蔵周辺機能が持つレジスタはメモリ・マップに割り付けられているので、メモリを参照することで確認できます。しかし周辺機能レジスタが名称で呼べるように工夫していますから、ぜひ分かりやすい専用ウィンドウを使ってください。

「IO レジスタ」ウィンドウを開いたらそのウィンドウ内で右クリックし H8SX/1655 用の I/O ファイル (h8sx1655.io) をロードします。これは最初の 1 回だけです。

I/O ファイルはテキストで構成されています。レジスタ名称を変更したり、内蔵周辺機能だけでなく外部に接続した周辺機能を追加したりできます。

### ● 変数の参照・変更

図 14 のウォッチ・ウィンドウで変数の参照・変更が行えます。

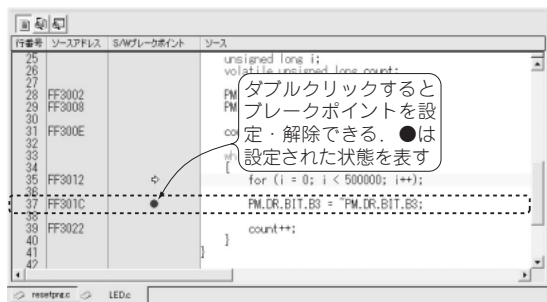
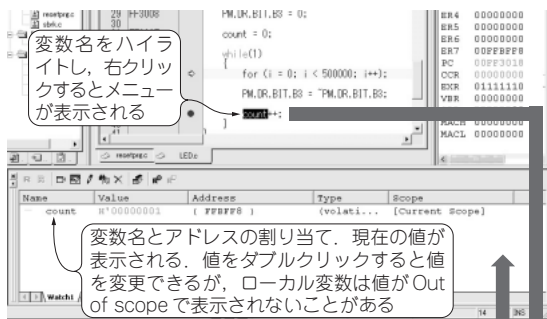
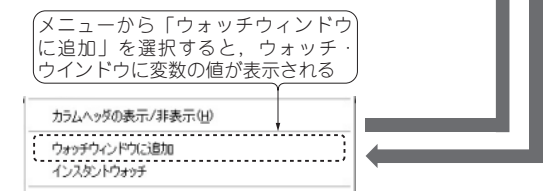


図 12 ブレークポイントの設定と解除



(a) ウォッチ・ウィンドウと変数表示例



(b) ウォッチ・ウィンドウへの変数登録

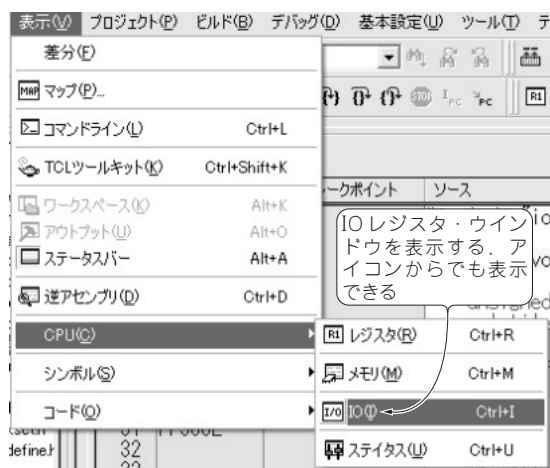
図 14 変数の参照と変更

## デバッグできたら ROM に書き込む

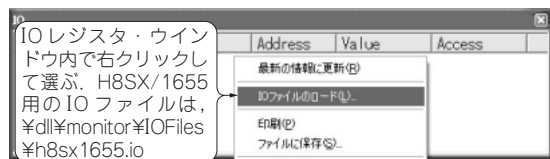
デバッグできたプログラムは、H8SX/1655 内蔵フラッシュ ROM に書き込んで自律動作させます。シリアル・デバッグでデバッグしていたときは内蔵 RAM に記憶するようセクションの先頭番地を設定していましたが、今度はこれを内蔵フラッシュ ROM に戻す必要があります。

### ● ワンクリックで通常用とデバッグ用のビルド・オプションを切り替える方法

HEW には「ビルド・コンフィグレーション」という機能があり、異なるビルド・オプションを持つことが



(a) 周辺機能レジスタの表示



(b) I/O 設定ファイルのロード



(c) 「IO ウィンドウ」の表示

図 13 周辺機能レジスタの表示方法

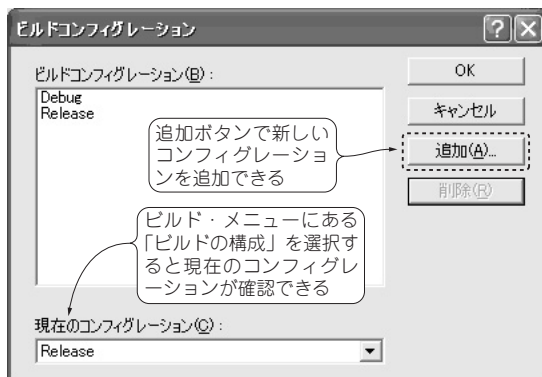


(a) コンフィグレーションの変更

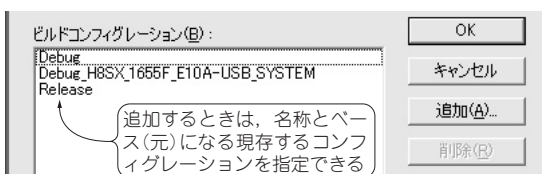


(b) コンフィグレーション単位に設定できるオプション

図 15 二つのコンフィグレーション切り替えでROM書き込みとデバッグを簡単に選べる



(a) コンフィグレーションの追加



(b) 例

図 16 コンフィグレーションの追加

できます。ビルド・コンフィグレーションを図 15 に示すように「Debug」から「Release」に変更します。もう一度最適化リンクの「セクション」を開いて見るとフラッシュROMのアドレスが設定されています。このままビルドしFDTを使って書き込みます。

別のビルドのオプションを設定したコンフィグレーションを持ちたいときは、図 16 に示すように追加することもできます。ビルド・メニューにある[ビルドの構成]で追加ボタンを押し、名称を入力します。

〈天野 利幸〉

# トランジスタ技術

役にたつエレクトロニクスの総合誌

通常号定価780円 毎月10日発売



CQ出版社

『トランジスタ技術』は、実用性を重視したエレクトロニクス技術の専門誌です。現場で通用する、電子回路技術、パソコン周辺技術、マイコン応用技術、半導体技術、計測/制御技術を、具体的かつ実践的な内容で、実験や製作を通して解説します。

大きな特徴の一つは、毎号80ページ以上の特集記事です。質、量ともに単行本に匹敵する内容で、保存していつまでも活用することができます。もう一つの特徴は、基礎に重点をおいた連載記事と最新技術を具体的に解説する特設記事です。重要なテーマには十分なページを割いて、理解しやすく解説しています。このほか、製作/実験記事、最新デバイスの評価記事など、役立つ実用的な情報を満載しています。

ウェブ・サイト(<http://toragi.cqpub.co.jp/>)からできること

最新号やバック・ナンバの購入

メルマガ配信

過去記事検索

年間購読