

ルネサステクノロジ社統合開発環境「HEW」で動作する
BCH836094G用 サンプルプログラム 添付CDの使い方

有限会社ビーリバーエレクトロニクス

第1版2009.7.9



・ CD構成

1. 開発環境
 - a: 開発に必要な機材
 - b: H8 / 36094Gの特徴
2. E8aについて
3. HEWの使い方、デバックの概要
4. 事前準備
 - a: ファイルの説明、コピー
 - b: HEW + E8a起動、デバック例
 - c: 説明のアイコンが見当たらない場合

・ BCH836094 CPUボード用 サンプルプログラム

【 入門 】

1. sample 1 ポートのON、OFF
2. sample 2 A/D、I/O、SIOプログラム
3. sample 3 16ビットPWMを使用したD/Aコンバータ
4. sample 4 Vタイマーを使用した定周期割り込み
5. sample 5 省エネ動作1 サブクロック動作
6. sample 6 省エネ動作2 スタンバイモード

【 演習 】

8. sample 8 扱えるデータ型、数値範囲、浮動小数点演算
9. sample 9 ポートレジスタ、メモリの設定、データを見る

【 応用 】

10. 新しいプロジェクトワークスペースを製作する

1. 開発環境

弊社のBCH836094G CPUボードはルネサステクノロジ社の統合開発環境HEW上でプログラムを開発し、同じくルネサステクノロジ社のH-UDI(JTAG)デバッカE8aでプログラムの書き込み、Cソースコードデバックを行うものです。

従来は考えられなかったくらい安価にJTAGデバッカでCソースコードデバックが可能です。

a: 必要な機器

BCH836094Gの開発を行うためには本CPUボード+E8a+HEWの3点が最低限必要です。電源はE8aが5V、3.3V 最大300mAまで出力できますので、不足しなければ不要です。CPUは単体で10mA(10MHz動作、5V電源時)程度消費しますので、290mAくらい、外部のハードウェアに使用できます。

弊社では3点セット+サンプルプログラム集CDを同梱した「開発セット」を用意いたしております。

型名	機能、特徴
BCH836094G開発セット	BCH836094G+E8a+サンプルソフト集、HEWの操作方法等が書かれたCD

b: H8/36094Gの特徴

H8/36094GはH8/300HコアのTinyマイコンです。パッケージ48ピンQFPでH8/36109Gの省ピン、ROM、RAM版です。H836109Gのように100ピンまでI/Oを必要としない場合、ROM、RAMの容量が足りる場合、H8/36109Gよりローコストとなります。

特徴として

ROM 32Kバイト、RAM2Kバイト

ROM 書き換え可能回数1000回以上

クロック、リセット回路内蔵 クロックはプログラムで可変可能。最大20MHz。

SCI 1ch内蔵

A/D 分解能10ビット、8ch内蔵

入出力ポート31本、このうち大電流ポート8本(IoL=20mA)、入力ポート8本

タイマA、V、W、ウォッチドッグタイマ、I²Cバス等内蔵

動作電源は5V

H8/3048、H8/3052等に比べ極めてローコスト

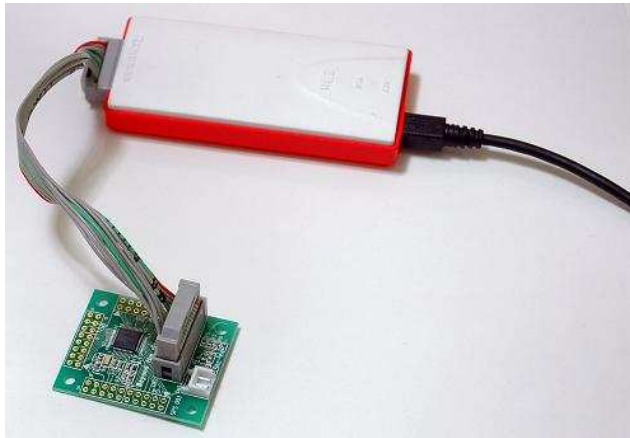
E8aでCソースデバックが可能

外部に拡張メモリを付けない用途であればH8/3048等と同等の速さ、機能を持っています。クロック内蔵ですので、外部に水晶発振子を付ける必要がありません。コンパクト、ローコストですので、より広い用途に使用することも可能です。

2 . E 8 a について



E 8 a 本体



E 8 a と B C H 8 3 6 0 9 4 G の 接 続

E 8 a は U S B 接 続 で 使 用 す る 、 ル ネ サ ス テ ク ノ ロ ジ 社 の マ イ コ ン 書 き 込 み 器 、 デ バ ッ カ で す 。 書 き 込 み 器 と し て は S H マ イ コ ン か ら T i n y マ イ コ ン ま で 書 け ま す 。 デ バ ッ カ と し て は H 8 / T i n y シ リ ー ズ で ア ド レ ス ブ レ ー ク 機 能 等 内 蔵 マ イ コ ン (H 8 / 3 6 0 9 4 G な ど) は C 、 ア セ ン ブ ラ ソ ー ス デ バ ッ ク に 対 応 し ま す 。

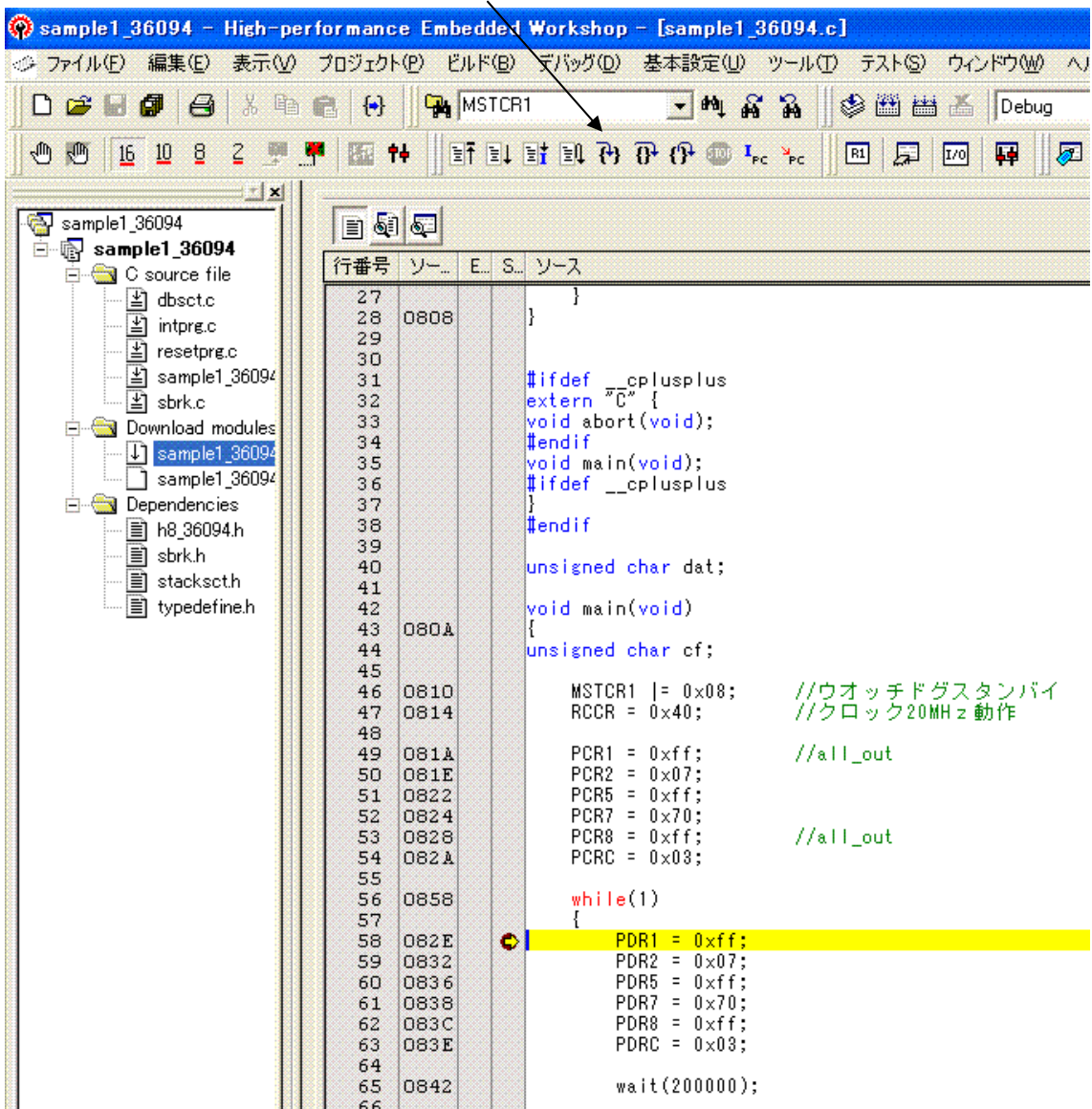
商品には
本体
U S B ケ ー ブ ル
J T A G 1 4 ピ ン ハ ー ネ ス
C D (統 合 開 発 環 境 H E W 、 C コ ン パ イ ラ 等)

が 同 梱 さ れ て いて 、 ソ フ ト ウ ェ ア を イ ン ス ト ー ル し て H E W を 立 ち 上 げ 、 プ ロ グ ラ ム 作 成 、 コ ン パ イ ル 、 書 き 込 み 、 デ バ ッ ク が で き ま す 。 U S B バ ス パ ワ ー で 最 大 3 0 0 m A ま で マ イ コ ン に 電 源 (3 . 3 V 、 5 V) を 供 給 で き ま す の で 、 そ の 範 囲 の ハ ー ド ウ ェ ア で あ れ ば 特 に 電 源 を 用 意 す る 必 要 が あ り ま せ ン 。 電 源 の O N , O F F は H E W か ら 行 い ま す 。

3. HEW (ヒューと発音します) の使い方、デバックの概要



例では sample1_36094 のプログラムをデバックしています。黄色の線は現在の実行行がそこで停止していることを示します。例えば { } マークをクリックすると 1 行ずつ実行していきます。



6 行実行すると、基板上的 LED D 2 が消灯します。

56	0858		<code>while(1)</code>
57			<code>{</code>
58	082E	●	<code> PDR1 = 0xff;</code>
59	0832		<code> PDR2 = 0x07;</code>
60	0836		<code> PDR5 = 0xff;</code>
61	0838		<code> PDR7 = 0x70;</code>
62	083C		<code> PDR8 = 0xff;</code>
63	083E		<code> PDR0 = 0x03;</code>
64			
65	0842	→	<code> wait(200000);</code>
66			
67	0846		<code> PDR1 = 0x00;</code>
68	084A		<code> PDR2 = 0x00;</code>
69	084C		<code> PDR5 = 0x00;</code>
			<code> ---</code>

このときPDR0(ポートC)の内容はI/Oの表示ウィンドウで確認できます。データが変化した場合、赤色表示されます。データは0x03を書き込んだのに0xffと表示されている理由はポートCがPC0、PC1の2ビットしかなく、他のビットは無条件に「1」と読み込まれるためです。

Name	Address	Value
PCR8	FFEB	FF
PDRB	FFDD	FE
PDR0	FFDE	FF
PCRC	FFEE	FF
Timer A		
Timer V		

さらに{ }をクリックすると、wait(20000)関数に飛びます。

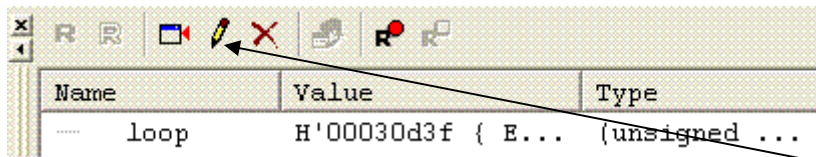
21			<code>void wait(unsigned long loop)</code>
22			<code>{</code>
23			
24	0800	→	<code> while(loop != 0)</code>
25			<code> {</code>
26	0802		<code> loop--;</code>
27			<code> }</code>
28	0808		<code>}</code>

{ } と { } の違いは、{ } はどんどん関数の中に入っていきますが、{ } はあくまでも中には入りません。この場合、67行へ行きます。

wait関数の中の例えばloopの値の変化は「ウォッチ」ウィンドウでloopを登録し、実行のたびに変わることが確認できます。

初めは200000 = H'30d40ですが、loop--を実行するたびに-1減算されていくのが分かります。

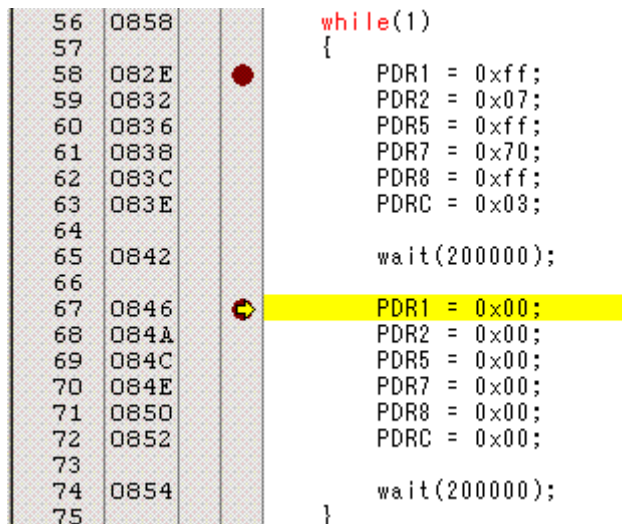
Name	Value	Type
loop	H'00030d40	{ E... (unsigned ...



ここでは200000回もループさせてもしょうがないので、loopに0を設定してみます。鉛筆マークをクリックすると変数の数値が変更できます。



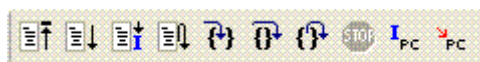
wait関数を抜けてmain関数に戻ってきました。



ブレークポイントを67行に設定し、「実行」によりそこまで飛ばしても同じです。ブレークポイントは希望する行にマウスの印を移動させダブルクリックすることにより、設定、解除ができます。(茶色のダブルクリックで設定、解除)

カーソルがある行はその命令の実行前です。1行でも過ぎてからその行が実行されます。

↓ 実行 (現在のアドレスからブレークポイントで止まる)



リセット、0番地から実行 (プログラムダウンロード後、初めの1回は必ずこれです)

72行まで { } で実行。74行は未実行です。

67	0846	PDR1 = 0x00;
68	084A	PDR2 = 0x00;
69	084C	PDR5 = 0x00;
70	084E	PDR7 = 0x00;
71	0850	PDR8 = 0x00;
72	0852	PDRC = 0x00;
73		
74	0854	wait(200000);

ここで、再びPDRCの値を見ると

Name	Address	Value
PCR8	FFEB	FF
PDRB	FFDD	FE
PDRC	FFDE	FC
PCRC	FFEE	FF
Timer A		
Timer V		

下2ビットが0で値は0xFCが読めています。B'11111100 = H'FC。

以上がデバックの概要です。BCH836094G+E8a+HEWでこのように簡単にCソースデバックができます。もちろん、アセンブラでもできます。

Cソース画面 混合 アセンブラ画面選択

E...	S...	逆ア...	オブジェクトコー...	ラベル	逆アセンブリ
		082E	F8FF		MOV.B #H'FF,ROL
		0830	38D4		ROL,@H'FFFFD4:8
		0832	F007		MOV.B #H'07,ROH
		0834	30D5		ROL,@H'FFFFD5:8
		0836	38D8		ROL,@H'FFFFD8:8
		0838	F070		MOV.B #H'70,ROH
		083A	30DA		ROL,@H'FFFFDA:8
		083C	38DB		ROL,@H'FFFFDB:8
		083E	F003		MOV.B #H'03,ROH
		0840	30DE		ROL,@H'FFFFDE:8
		0842	0FE0		MOV.L ER6,ER0
		0844	55BA		BSR @_wait:8
		0846	1888		SUB.B ROL,ROL
		0848	38D4		ROL,@H'FFFFD4:8
		084A	38D5		ROL,@H'FFFFD5:8
		084C	38D8		ROL,@H'FFFFD8:8
		084E	38DA		ROL,@H'FFFFDA:8
		0850	38DB		ROL,@H'FFFFDB:8
		0852	38DE		ROL,@H'FFFFDE:8
		0854	0FE0		MOV.L ER6,ER0

4 . 事前準備

a : ファイルの説明、コピー

入門者の方はプログラムの解説の前にC言語の解説書、例えば「はじめてのC」改訂版第5版「ANSI C 対応」 椋田實 著 技術評論社 があると便利です。

2進数、16進数変換、オームの法則等、ソフト、ハードの基礎的な部分は「ドキュメント」内にあるpdfファイル「組み込みマイコンの基礎 、 」をご参照願います。ファイル名 : mycon1.pdf、mycon2.pdf。

名前 ▲	サイズ	種類	更新日時
 BCH836094G.pdf	97 KB	Adobe Acrobat Doc...	2009/07/14 18:47
 H8_36094G_CIR.pdf	13 KB	Adobe Acrobat Doc...	2009/07/14 18:45
 COMポートの調べ方.pdf	313 KB	Adobe Acrobat Doc...	2006/03/28 18:36
 mycon1.pdf	51 KB	Adobe Acrobat Doc...	2006/11/14 18:15
 mycon2.pdf	177 KB	Adobe Acrobat Doc...	2006/11/14 18:14
 開発セットH836094.pdf	8,123 KB	Adobe Acrobat Doc...	2009/07/14 18:49
 パスワードH836109G.pdf	5 KB	Adobe Acrobat Doc...	2009/07/14 18:53
 rjj09b0278_h836094hm.pdf	7,465 KB	Adobe Acrobat Doc...	2009/07/14 18:55

上から

BCH836094G.pdf	取扱説明書
H8_36094G_CIR.pdf	回路図
COMポートの調べ方	COMポートの調べ方
mycon1.pdf	組み込みマイコンの基礎
mycon2.pdf	組み込みマイコンの基礎
開発セット H836094.pdf	このマニュアルです。
パスワード H836109G.pdf	バージョンアップ等ユーザー専用コーナーパスワード (H8/36109G と兼用)
rjj09bxxxx	H8/36094 ハードウェアマニュアル



サンプルソフトの中身、samplex_36094をWorkSpaceに移します。WorkSpaceはHEWをインストールするとデフォルトでC : に自動的に作成されるワークスペースです。プログラムはこの中に格納されます。もし、ご自身で設定したものがあればそちらに写します。

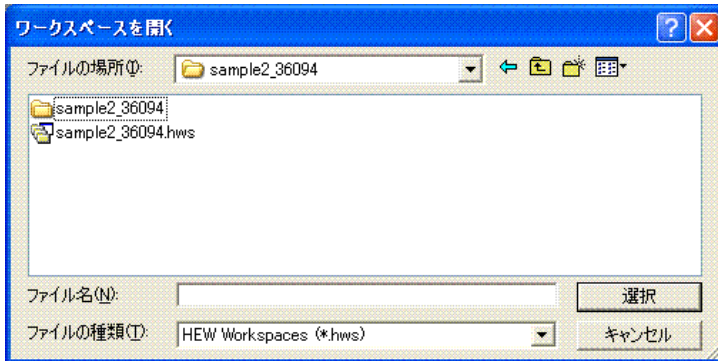
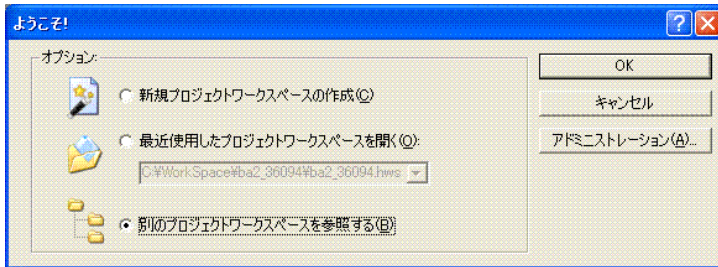
HEW、E8aは添付CDに従い、インストールを行います。

次にsample2_36094を例にデバックの方法を説明します。

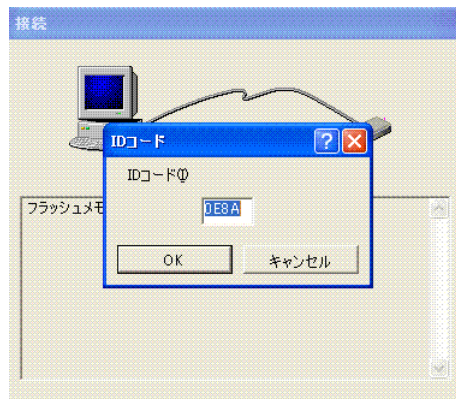
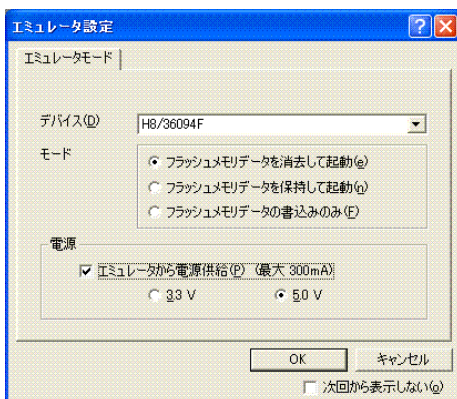
b : HEW + E 8 a 起動、デバック例

初めに、E 8 a と USB、E 8 a 14 ピンコネクタと HCH 8 3 6 0 9 4 G 基板を接続してください。
HEW を立ち上げます。

HEW は始めに使用するプロジェクトを聞いてきますので、「別のプロジェクトワークスペースを参照する」を選択し、sample2_36094 ホルダの中の sample2_36094.hws をダブルクリックして選択してください。一度使用すると、「最近使用したプロジェクトワークスペースを開く」でも選択できます。

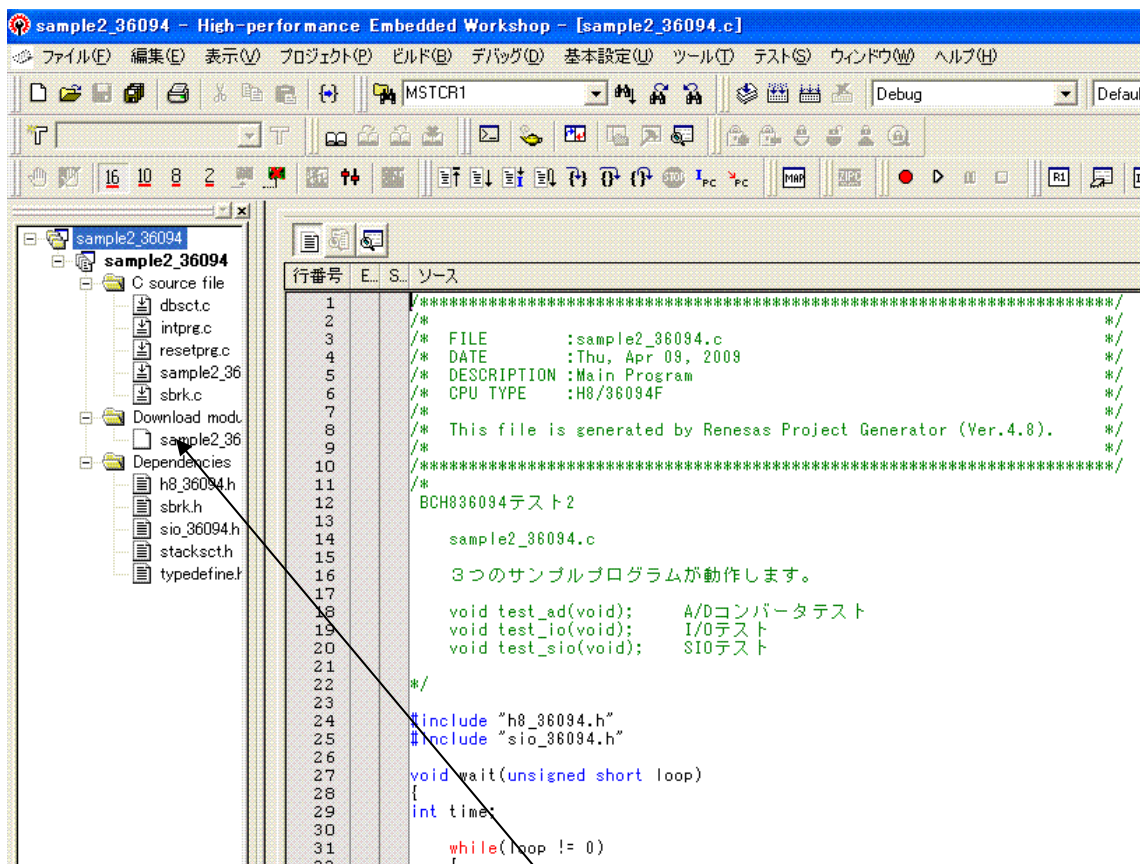



エミュレータ設定画面が立ちあがりますので、モードは「フラッシュメモリデータを消去して起動」でも、それ以外でも OK です。エミュレータ (E 8 a) から電源を供給する場合、クリックして、5 V を選択し、「OK」をクリックします。



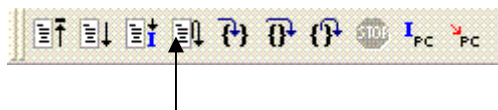
E 8 a の ID コードを聞いてきますので「OK」をクリックします。

下記のように左に使用しているプログラム、ヘッダ名等、右にプログラム自身が表示されればOKです。

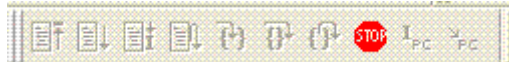


Download modulesにある「sample2_36094」をダブルクリックしてプログラムをダウンロードします。正常にダウンロードされると無地だったファイルマークに  マークが付きます。

「リセットして実行」をクリックするとプログラムが実行されます。



プログラムのリアルタイム動作中は以下のような表示になります。

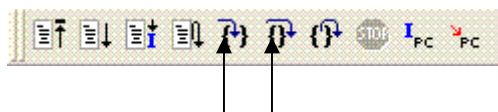


赤丸は「STOP」です。ブレークポイントまで動作、ステップ動作を行う場合、「STOP」をクリックして停止状態にしてから行います。

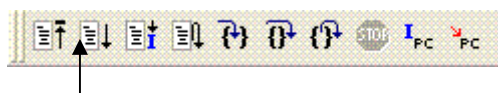
ブレークポイントを設定したい場合、

25		void main(void)
26	000800	{
27		//クロック10MHz
28		
29	000804	MSTCR1 = 0x08; //ウォッチドグスタンバイ
30		
31	000808	S/Wブレークポイント = 0xff;
32	00080C	PCR2 = 0xff;
33	000810	PCR3 = 0xff;
34	000814	PCR5 = 0xff;
35	000818	PCR7 = 0xff;

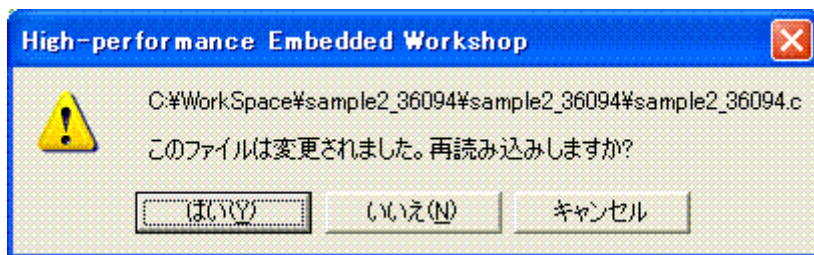
ここをマウスでダブルクリックすると茶色い丸が出て設定できます。解除もそこをダブルクリックします。「リセットして実行」をクリックし、それ以降、1行ずつステップ動作をさせるためにはここをクリックします。



左がステップイン、右がステップオーバーですが、違いは途中に関数があった場合、関数の中まではいるのがステップイン、入らず次行に行くのがステップオーバーです。用途により使い分けます。この地点からブレークポイントを新たに設定して、そこまで動作させるのは、「実行」をクリックします。



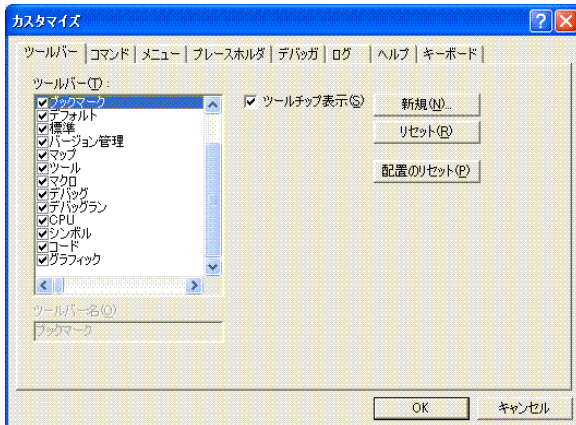
なお、エディタはH E Wの画面上でも行えますが、使い慣れたエディタを使用することも可能で、H E W側では常にファイルの変更を監視しており、セーブされると自動的に以下のような表示が現れ、再読み込みが行えます。



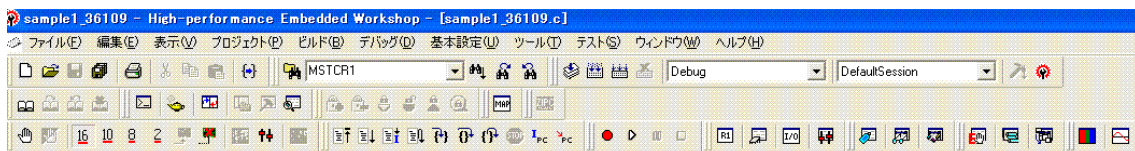
使い慣れたエディタでファイルを開いていて、H E W側のエディタで内容を変更してセーブする場合にも警告があり、一つのファイルの内容が異なる工夫があります。

c : 説明のアイコンが見当たらない場合

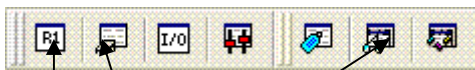
ツールバー画面上に各種アイコン等を表示、消去するのは「基本設定」->「カスタマイズ」



で選択します。例として、全部選択すると、以下のように表示されます。



本取説でよく使うウィンドウは



レジスタ、メモリ、ウォッチなどです。

【 入門 】

1. sample1 ポートのON,OFF

【 概要 】

全ポートのON、OFFプログラムです。以下がプログラム全文です。出荷検査のときに使用しているルーチンです。ここでは解説のための番号が入っています。

【 プログラムソース 】

```
/*
*****/
/*
*/
/* FILE      :sample1_36094.c          */
/* DATE      :Thu, Apr 09, 2009       */
/* DESCRIPTION :Main Program          */
/* CPU TYPE   :H8/36094F              */
/*
*/
/* This file is generated by Renesas Project Generator (Ver.4.8). */
/*
*/
/*
*****/
/*
*/
BCH836094 sample1

        ポート ON,OFF

*/

#include "h8_36094.h"

void wait(unsigned long loop)
{
    while(loop != 0)
    {
        loop--;
    }
}

#ifdef __cplusplus
extern "C" {
void abort(void);
#endif
void main(void);
#ifdef __cplusplus
}
#endif
```

```

unsigned char dat;

void main(void)
{
unsigned char cf;

MSTCR1 |= 0x08;           //ウォッチドグスタンバイ
RCCR = 0x40;             //クロック 20MHz 動作

PCR1 = 0xff;             //all_out
PCR2 = 0x07;
PCR5 = 0xff;
PCR7 = 0x70;
PCR8 = 0xff;             //all_out
PCRC = 0x03;

while(1)
{
PDR1 = 0xff;
PDR2 = 0x07;
PDR5 = 0xff;
PDR7 = 0x70;
PDR8 = 0xff;
PDRC = 0x03;

wait(200000);

PDR1 = 0x00;
PDR2 = 0x00;
PDR5 = 0x00;
PDR7 = 0x00;
PDR8 = 0x00;
PDRC = 0x00;

wait(200000);
}
}

#ifdef __cplusplus
void abort(void)

```

```
{
```

```
}
```

```
#endif
```

【 解説 】

```
#include "h8_36094.h"
```

H 8 / 3 6 0 9 4 のレジスタのアドレスとサイズを定義しているファイルをインクルードしています。これによりプログラマは各々のレジスタの絶対番地、サイズをプログラム中に書く必要がなくなります。また、それらの記入ミスからは開放されません。

```
void wait(unsigned long loop);
```

wait 関数です。 から呼ばれている関数です。loop が 0 になるまでの時間、関数の中でループしますので、loop に設定する数により時間が変化します。

```
#ifdef __cplusplus
```

```
extern "C" {
```

```
void abort(void);
```

```
#endif
```

```
void main(void);
```

```
#ifdef __cplusplus
```

```
}
```

```
#endif
```

H E W でワークスペースプロジェクトを作成したときに、自動的に作成されます。特に使用しません。

```
void main(void)
```

ここからが電源 ON によりスタートアッププログラムの次に動作するメインのプログラムです。

```
MSTCR1 |= 0x08; //ウォッチドグスタンバイ
```

```
RCCR = 0x40; //クロック 20MHz 動作
```

ウォッチドグタイマーは電源投入時、有効です。ここでは使用しませんので切っています。

あと、H 8 / 3 6 0 9 4 G は電源投入時、1 0 M H z 動作となっていますので、最大クロックの 2 0 M H z に上げています。

```
PCR1 = 0xff;
```

ポートの初期設定を行います。ポートは電源投入時、全て入力になっています。出力として使用する場合、ポートコントロールレジスタの対応ビットに '1' を設定します。ポート全部を出力にする場合、0 x f f になります。ポート 1、2、3、5、7、C、D、E、G、H すべてを出力にしています。ポート 8 はデバック (E 8 a) で使用するポートなので設定しません。

```
PDR1 = 0;
```

ポート 1、2、5、7、8、C すべてに 0 を出力しています。

L E D D 2 は P D R C = 0 で点灯します。

```
wait(200000);
```

wait()関数に200000というループ回数を渡し、時間を製作しています。

```
PDR1 = 0xff;
```

ポート1、2、5、7、8、Cすべてに0xffを出力しています。端子は5Vになります。

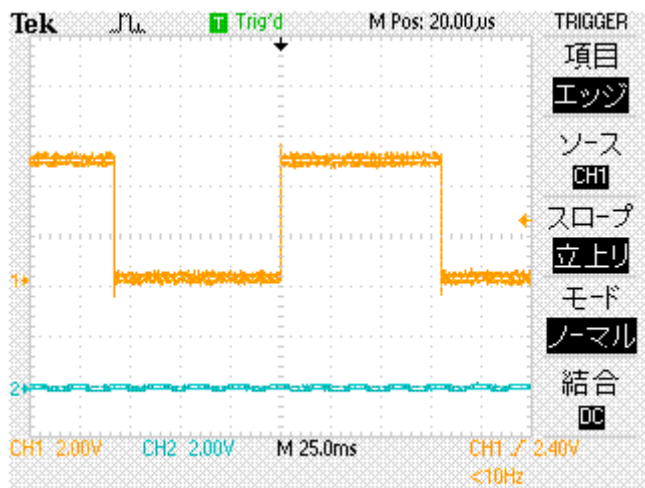
LED D2はPDR C = 0xffで消灯します。

while(1)で囲まれているので、それを繰り返します。

```
#ifdef __cplusplus
void abort(void)
{
}
#endif
```

HEWでワークスペースプロジェクトを作成したときに、自動的に作成されます。特に使用しません。

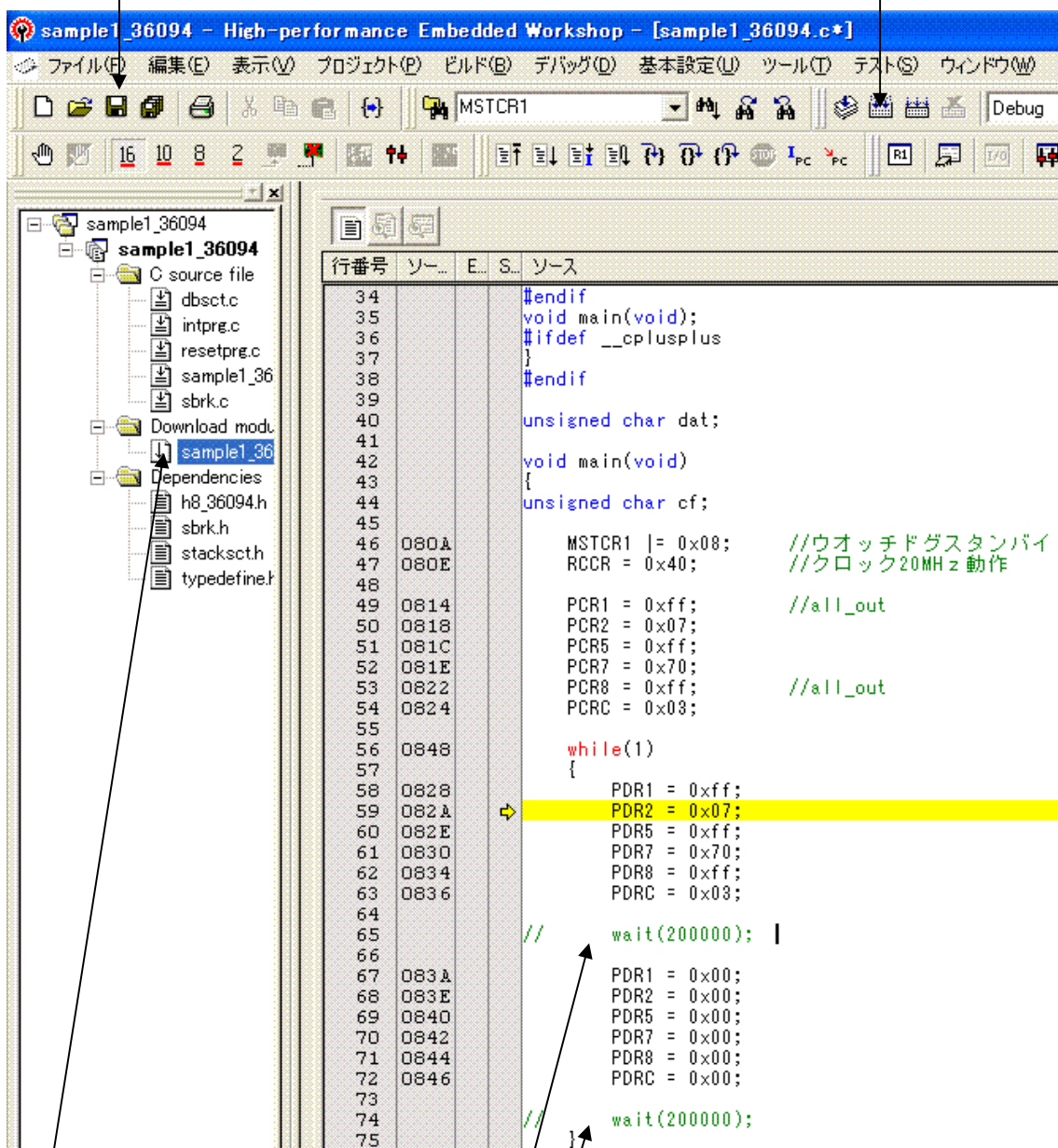
ポートをオシロスコープで観測すると6Hz程度の波形が繰り返していることが分かります。



で解説したwait関数(行番号 65、74)を//でコメントにして波形の変化を見ます。プログラムの変更、セーブ、コンパイル、ダウンロードの手順を追って説明します。

ファイルを保存する。

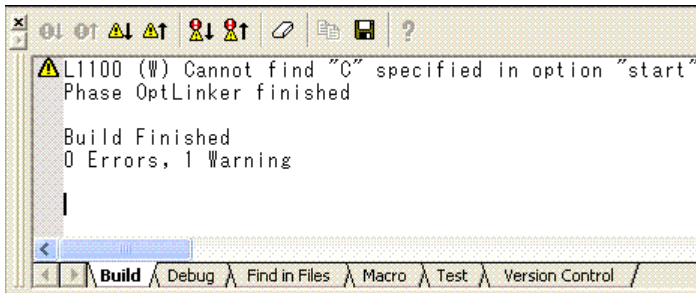
コンパイルする



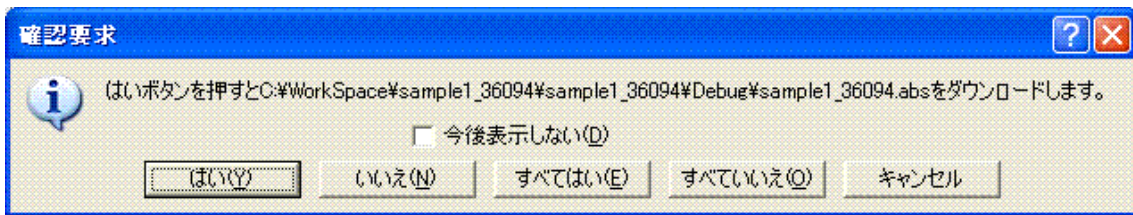
ダウンロード

ここに//を書いてコメントとする。コメントは色が緑に変わります。

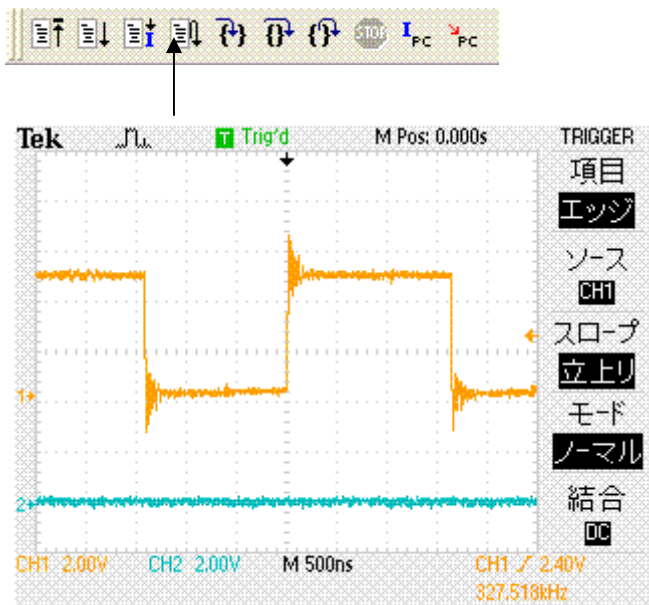
Build画面で0 ErrorsならOK。0以上のエラーはなにか間違いがあります。画面をさかのぼって、エラー行あたりを修正します。1 Warning(警告)は問題ありません。



Download modulesをダブルクリックして新しいプログラムをダウンロードします。あるいはエラーでない場合は以下の画面が出ますので、はい(Y)をクリックします。



「リセットして実行」をクリックするとプログラムが実行されます。



周波数が327KHzとなり、さきほどの6Hzの54500倍も速い波形となりました。

このようにプログラムの修正は「プログラムの修正、セーブ、コンパイル、ダウンロード」という手順で行います。

一度、BCH836094G基板側の電源を切りたいときは「接続解除」をクリックし、電源を切ることができます。「接続」はこの逆で、電源を入れてから「接続」をクリックします。



H E W動作中にターゲットの電源を切ったり、S L E E P命令の実行時等にH E Wがタイムアウトを出力し、ソフトウェアの再立ち上げを要求する場合がありますが、一度「接続解除」し、再度「接続」することにより、H E W自身の再立ち上げが不要になる場合もあります。

以下省略

-
- 1．本文章に記載された内容は弊社有限会社ビーリバーエレクトロニクスの調査結果です。
 - 2．本文章に記載された情報の内容、使用結果に対して弊社はいかなる責任も負いません。
 - 3．本文章に記載された情報に誤記等問題がありましたらご一報いただけますと幸いです。
 - 4．本文章は許可なく転載、複製することを堅くお断りいたします。

〒350-1213 埼玉県日高市高萩 1 1 4 1 - 1

TEL 042 (985) 6982

FAX 042 (985) 6720

Homepage : <http://beriver.co.jp>

e-mail : info@beriver.co.jp

有限会社ビーリバーエレクトロニクス

©Beyond the river Inc 20090714