

ルネサステクノロジ社統合開発環境「HEW」で動作する
BCH836109G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2008.7.2 第2版2008.8.6 電源3V NG



・CD構成

1. 開発環境
 - a: 開発に必要な機材
 - b: H8/36109G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

・BCH836109 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 14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

【 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BCH836109G 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

BCH836109G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
BCH836109G開発セット	BCH836109G+E8a+サンプルソフト集、HEWの操作方法等が書かれたCD+クリップ2本
価格	19,800円(税込み)

b: H8/36109Gの特徴

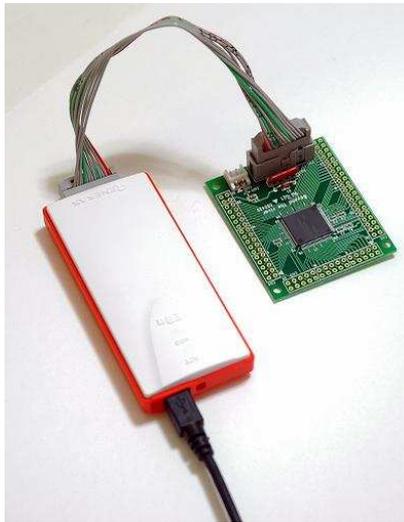
H8/36109GはH8/300HコアのTinyマイコンです。特徴として
ROM 128Kバイト、RAM5Kバイト
ROM 書き換え可能回数1000回以上
クロック、リセット回路内蔵 クロックはプログラムで可変可能。最大20MHz。
SCI 3ch内蔵
A/D 分解能10ビット、16ch内蔵
入出力ポート79本、このうち大電流ポート20本 (IoL=20mA) 入力ポート8本
RTC (リアルタイムクロック)、タイマB1, V、RC、RD、14ビットPWM、ウォッチドッグタイマ、I²Cバス等内蔵
動作電源は5V
H8/3048、H8/3052等に比べ極めてローコスト
E8aでCソースデバックが可能

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。H8/3048に比べA/D、SCIの数が拡張されていますので、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 1 0 9 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 1 0 9 G な ど) は C 、 ア セ ン ブ ラ ソ ー ス デ バ ッ ク に 対 応 し ます 。

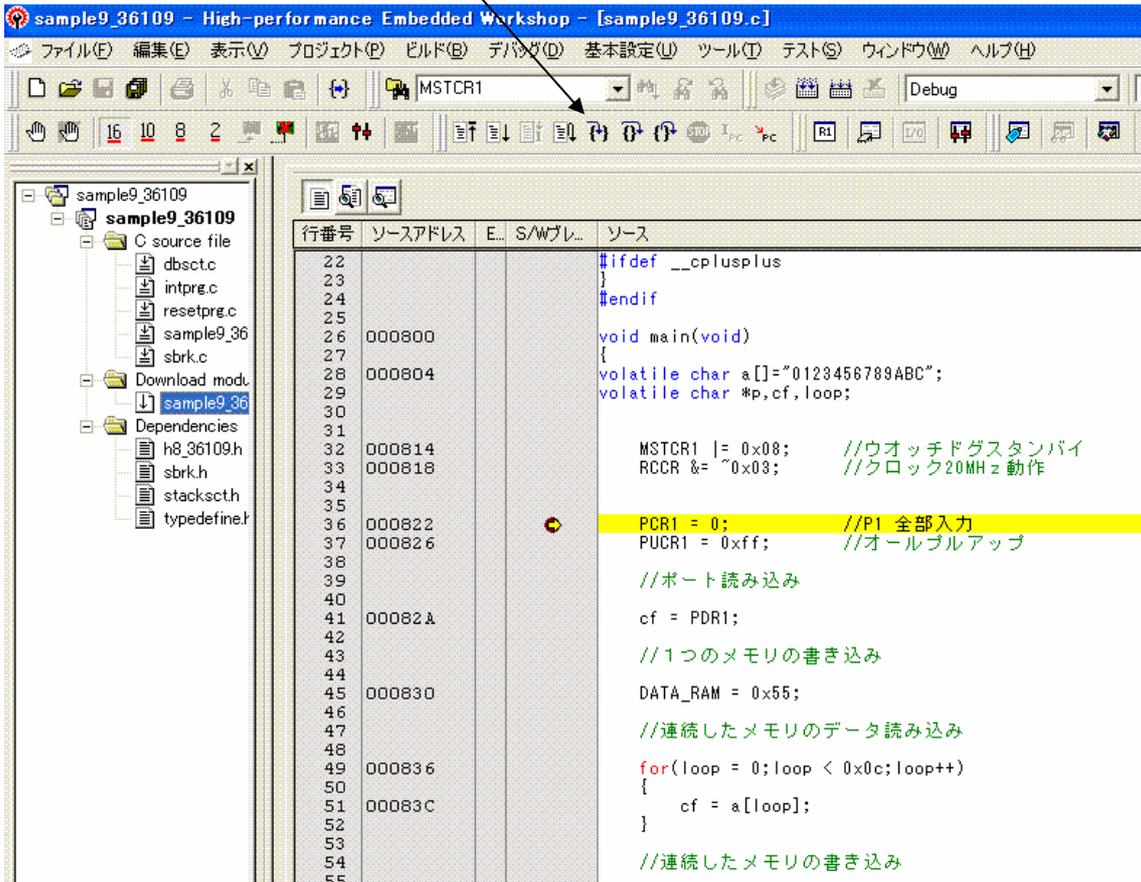
商品には
本体
U S B ケ ー ブ ル
J T A G 1 4 ピ ン ハ ー ネ ス
C D (統 合 開 発 環 境 H E W 、 C コ ン パ イ ラ 等)

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 パ 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 。

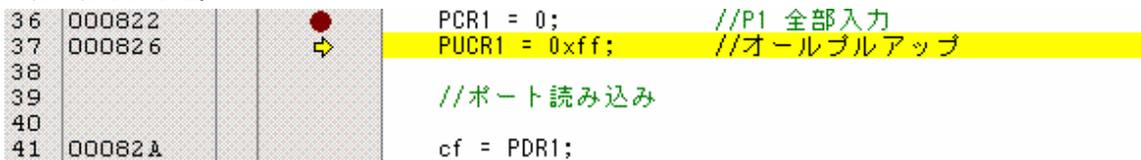
3. HEW (ヒューと発音します) の使い方、デバックの概要



例では sample9_36109 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。



1 行実行しました。



このとき P D R 1 (ポート 1) の内容は I / O の表示ウィンドウで確認できます。

Name	Address	Value
I/O Ports		
PDR1	FFFFD4	FB
PCR1	FFFFE4	FF
PUCR1	FFFFD0	08
PMR1	FFFFE0	00
PDR2	FFFFD5	FE

また、2行進めます。41行の実行でPDR1の内容はcfレジスタにも入れられました。

39			//ポート読み込み
40			
41	00082A		cf = PDR1;
42			
43			//1つのメモリの書き込み
44			
45	000830	⇒	DATA_RAM = 0x55;
46			
47			//連続したメモリのデータ読み込み
48			

あらかじめ「ウォッチ」ウィンドウを開いて「cf」を登録してありましたので、H'fbという値=先ほどのI/Oポートウィンドウと同じ値が得られています。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。

Name	Value
cf	H'fb '.' { FFF..
loop	H'00 '.' { FFF..

Watch1 Watch2 Watch3 Watch4

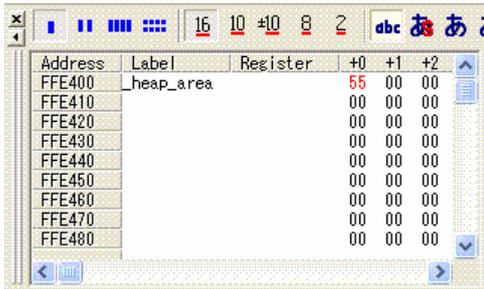
1行進めます。45行目の命令はDATA_RAMに0x55を書き込んでいます。

42			
43			//1つのメモリの書き込み
44			
45	000830		DATA_RAM = 0x55;
46			
47			//連続したメモリのデータ読み込み
48			
49	000836	⇒	for(loop = 0; loop < 0x0c; loop++)

DATA_RAMのアドレスは0xffe400です(プログラムの頭で定義しています)。

```
#define DATA_RAM (*(volatile unsigned char *)0xffe400)
```

そこで、「メモリ」ウィンドウで0xffe400を見てみると、ちゃんと0x55が書き込まれていることが分かります。



1行進めます。

47			//連続したメモリのデータ読み込み
48			
49	000836	⇒	for(loop = 0; loop < 0x0c; loop++)
50			{
51	00083C		cf = a[loop];
52			}

1行進めるたびにc fのデータとl o o pのデータが変化するのが「ウオッチ」ウィンドウで分かります。

47			//連続したメモリのデータ読み込み
48			
49	000836		for(loop = 0; loop < 0x0c; loop++)
50			{
51	00083C	⇒	cf = a[loop];
52			}

l o o p = 4 のときに c f = 0 x 3 3 = アスキーコード 3 が得られています。以下の配列のデータを読んでいます。

```
volatile char a[]="0123456789ABC";
```

(cf=a[loop]はカーソル上でまだ実行されていない。よってl o o p = 4 に対して1個前のデータ3が得られている)



loopは長いのでブレークポイント（茶色のダブルクリックで設定、解除）を設定し、実行。先に進みます。

実行（ブレークポイントで止まる）

```

47 //連続したメモリのデータ読み込み
48
49 000836 for(loop = 0; loop < 0x0c; loop++)
50 {
51 00083C     cf = a[loop];
52 }
53 //連続したメモリの書き込み
54
55
56 000860 p = RENZ_RAM;

```

連続してメモリに書き込んでいます。

```

54 //連続したメモリの書き込み
55
56 000860 p = RENZ_RAM;
57 000866 *p = 0x41; p++;
58 00086E *p = 0x42; p++;
59 000874 *p = 0x43; p++;
60 00087A *p = 0x44; p++;
61 000880 *p = 0x45; p++;
62 000886 *p = 0x00;
63
64 00088A while(1)

```

#define RENZ_RAM ((volatile char *)0xffe410) (プログラム初めの宣言)

0xffe4100 から連続して書き込まれたことが確認できます。

Address	Label	Register	+0	+1	+2	+3	+4	+5	+6
FFE400	_heap_area		55	00	00	00	00	00	00
FFE410			41	42	43	44	45	00	00
FFE420			00	00	00	00	00	00	00
FFE430			00	00	00	00	00	00	00
FFE440			00	00	00	00	00	00	00
FFE450			00	00	00	00	00	00	00
FFE460			00	00	00	00	00	00	00
FFE470			00	00	00	00	00	00	00
FFE480			00	00	00	00	00	00	00

以上がデバックの概要です。BCH836109G+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。

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
BCH836109G.pdf	104 KB	Adobe Acrobat Doc...
H8_36109G_CIR.pdf	19 KB	Adobe Acrobat Doc...
mycon1.pdf	51 KB	Adobe Acrobat Doc...
mycon2.pdf	177 KB	Adobe Acrobat Doc...
rjj09b0217_h836109hm.pdf	7,764 KB	Adobe Acrobat Doc...
パスワードのご通知H836109G.p...	5 KB	Adobe Acrobat Doc...
開発セットH836109.pdf	6,823 KB	Adobe Acrobat Doc...

上から

BCH836109G.pdf	取扱説明書
H8_36109G_CIR.pdf	回路図
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
rjj09bxxxx	H8/36109 ハードウェアマニュアル
パスワードのご案内	バージョンアップ等ユーザー専用コーナーパスワード
開発セットH836109G.pdf	このマニュアルです。



サンプルソフトの中身、samplex_36109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sample1_36109を例にデバックの概要を説明します。

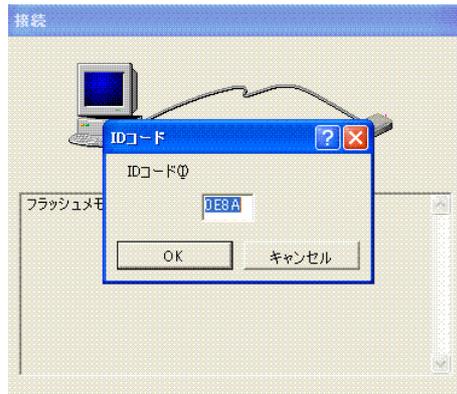
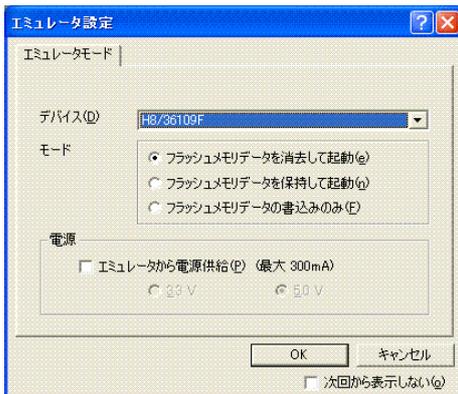
b : H E W + E 8 a 起動、デバック例

初めに、E 8 a と U S B、E 8 a 1 4 ピンコネクタと H C H 8 3 6 1 0 9 G 基板を接続してください。
H E W を立ち上げます。

H E W 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、s a m p l e 1 _ 3 6 1 0 9 ホルダの中の s a m p l e 1 _ 3 6 1 0 9 . h w s 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。

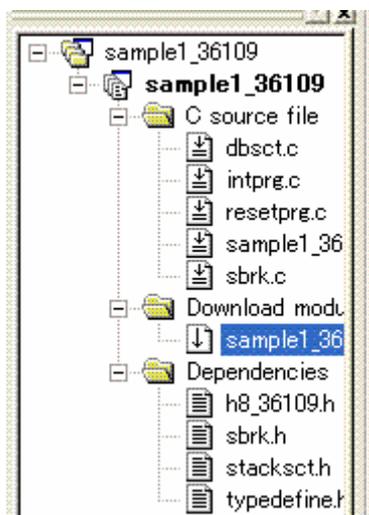
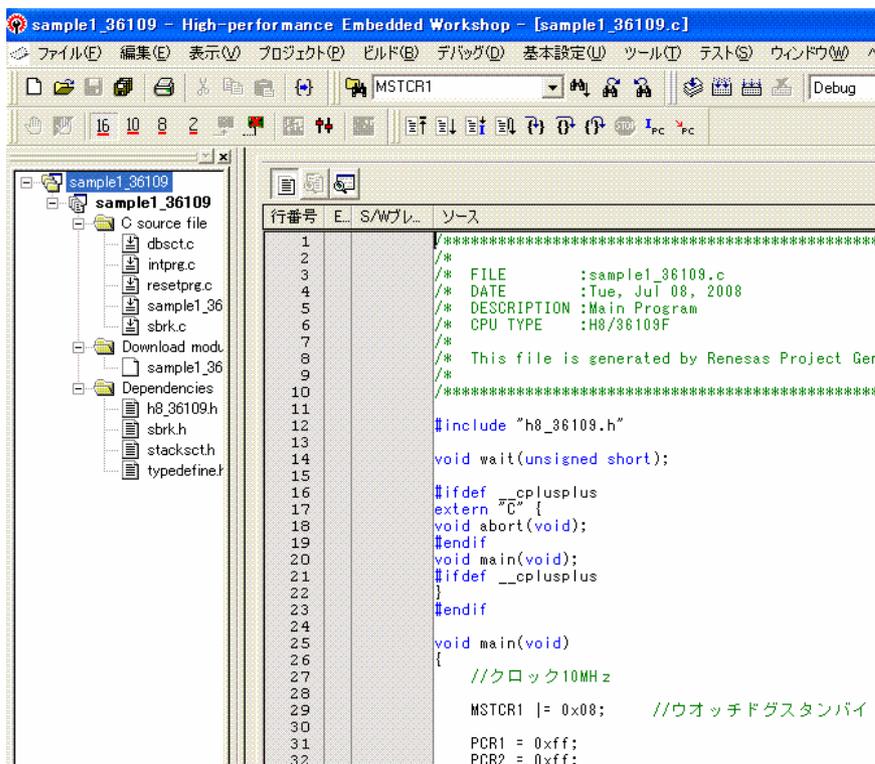


エミュレータ設定画面が立ちあがりますので、モードは「フラッシュメモリデータを消去して起動」でも、それ以外でも OK です。エミュレータ (E 8 a) から電源を供給する場合、クリックして、5 V を選択し、「OK」をクリックします。



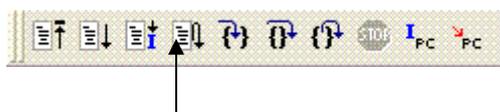
E 8 a の I D コードを聞いてきますので「OK」をクリックします。

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「sample1_36109」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。

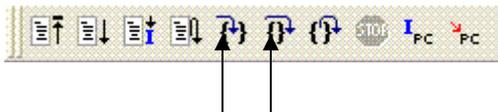


赤丸は「STOP」です。ブレークポイントまで動作、ステップ動作を行う場合、「STOP」をクリックして停止状態にしてから行います。

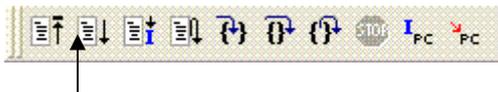
ブレークポイントを設定したい場合、



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。

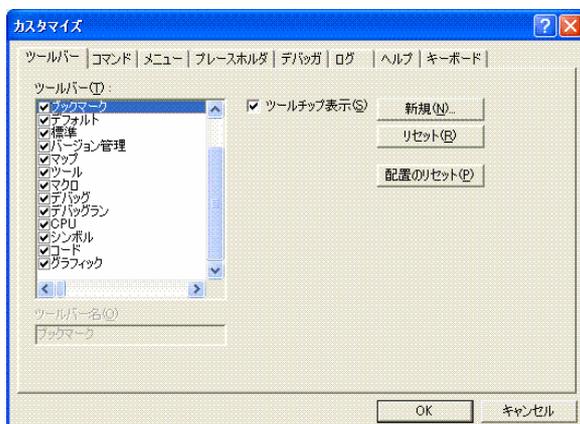


左がステップイン、右がステップオーバーですが、違いは途中に関数があった場合、関数の中まではいるのがステップイン、入らず次行に行くのがステップオーバーです。用途により使い分けます。この地点からブレークポイントを新たに設定して、そこまで動作させるのは、「実行」をクリックします。



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 】

```
/*
*****/
/*
*/
/* FILE      :sample1_36109.c          */
/* DATE      :Tue, Jul 08, 2008       */
/* DESCRIPTION :Main Program          */
/* CPU TYPE   :H8/36109F              */
/*
*/
/* This file is generated by Renesas Project Generator (Ver.4.8). */
/*
*/
*****/

#include "h8_36109.h"

void wait(unsigned short);

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

void main(void)
{
    //クロック 10MHz

    MSTCR1 |= 0x08;          //ウォッチドグスタンバイ

    PCR1 = 0xff;
    PCR2 = 0xff;
    PCR3 = 0xff;
    PCR5 = 0xff;
    PCR7 = 0xff;
    // PCR8 = 0xff;      JTAG PORT
    PCRC = 0xff;
}
```

```

PCRD = 0xff;
PCRE = 0xff;
PCRG = 0xff;
PCRH = 0xff;

    while(1)
    {
        PDR1 = 0;
        PDR2 = 0;
        PDR3 = 0;
        PDR5 = 0;
        PDR7 = 0;
//        PDR8 = 0;
        PDRC = 0;
        PDRD = 0;
        PDRE = 0;
        PDRG = 0;
        PDRH = 0;
        wait(300);

        PDR1 = 0xff;
        PDR2 = 0xff;
        PDR3 = 0xff;
        PDR5 = 0xff;
        PDR7 = 0xff;
//        PDR8 = 0xff;
        PDRC = 0xff;
        PDRD = 0xff;
        PDRE = 0xff;
        PDRG = 0xff;
        PDRH = 0xff;
        wait(300);
    }
}

```

```

void wait(unsigned short wtime)
{
    unsigned short sdata;

    while(wtime != 0)
    {
        sdata = 100;
        while(sdata != 0)
        {
            sdata--;

```

```

    }
    wtime--;
}
}
}

```

```

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

```

【 解説 】

```
#include "h8_36109.h"
```

H 8 / 3 6 1 0 9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す。

```
void wait(unsigned short);
```

wait 関数の宣言を行っています。 から書かれている関数ですが、書き方として main 関数より前に書くか、後を書く場合はこのように関数の形を宣言する必要があります。

```

#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; //ウオッチドグスタンバイ
```

ウオ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。

```
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、3、5、7、C、D、E、G、Hすべてに0を出力しています。

```
wait(300);
```

wait () 関数に300というパラメータを渡し、時間を製作しています。

```
PDR1 = 0xff;
```

ポート1、2、3、5、7、C、D、E、G、Hすべてに0xffを出力しています。端子は5Vになります。while (1) で囲まれているので、それを繰り返します。

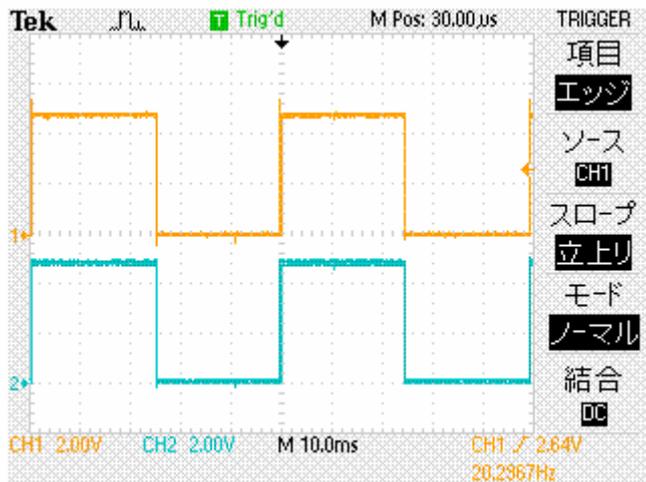
```
void wait(unsigned short wtime)
```

300から1つずつ引いて、0になるまでプログラムを抜けません。特に単位はありませんが、これで時間を作っています。

```
#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。

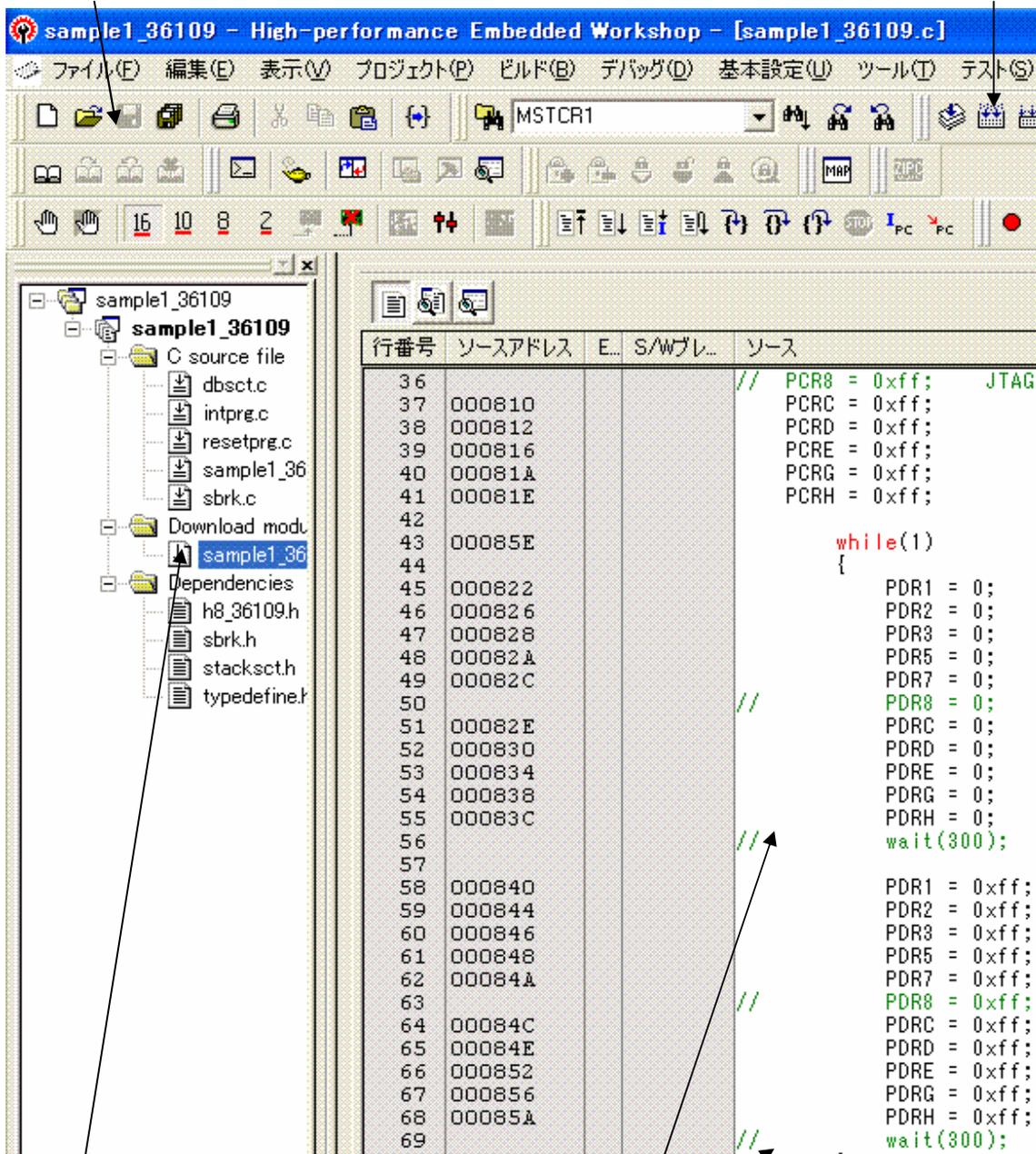
ポートをオシロスコープで観測すると20Hz程度の波形が繰り返していることが分かります。



で解説したwait関数(行番号 56、69)を//でコメントにして波形の変化をみます。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。

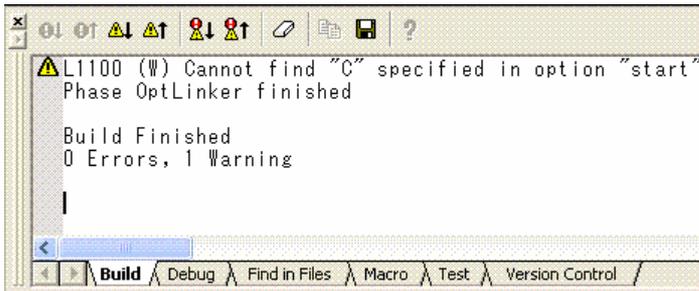
コンパイルする



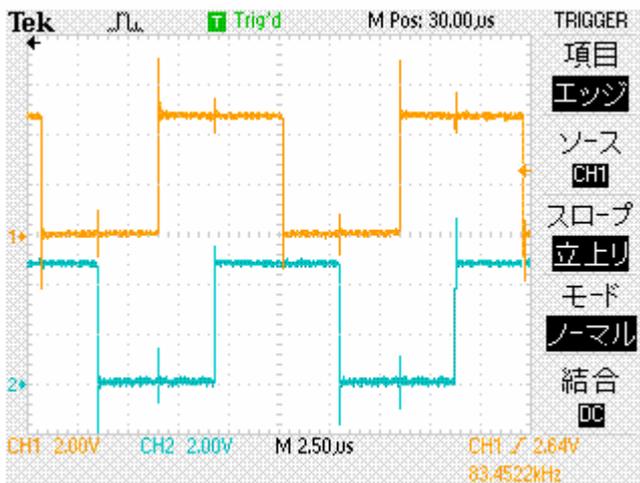
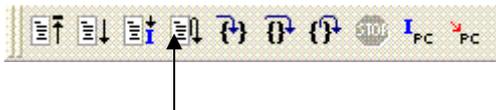
ダウンロード

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。



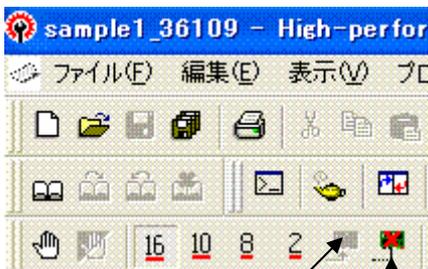
Download modulesをダブルクリックして新しいプログラムをダウンロードします。
 「リセットして実行」をクリックするとプログラムが実行されます。



周波数が83kHzとなり、さきほどの20Hzの4150倍も速い波形となりました。

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、BCH836109G基板側の電源を切りたいときは「接続解除」をクリックし、電源を切りま
 す。「接続」はこの逆で、電源を入れてから「接続」をクリックしてください。



接続 接続解除

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 : support@beriver.co.jp

有限会社ビーリバーエレクトロニクス ©Beyond the river Inc 20080704