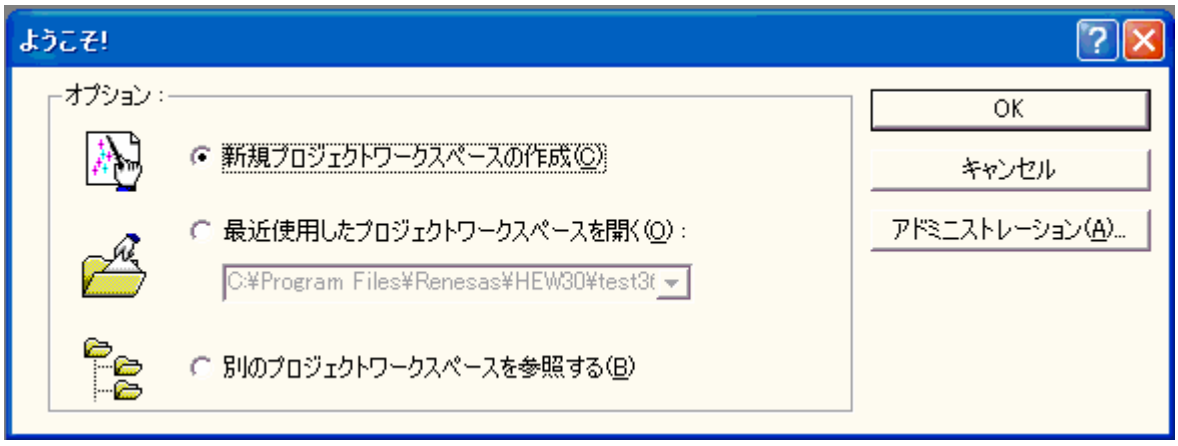
■ **ＨＥＷ4を使用したＨ８－３６７２ＣＰＵボード開発の仕方**　２００５．８．２９

２００６．３．２８　書き込みをフォースライタに変更

●ＨＥＷ4を立ち上げると下図のような表示があります。



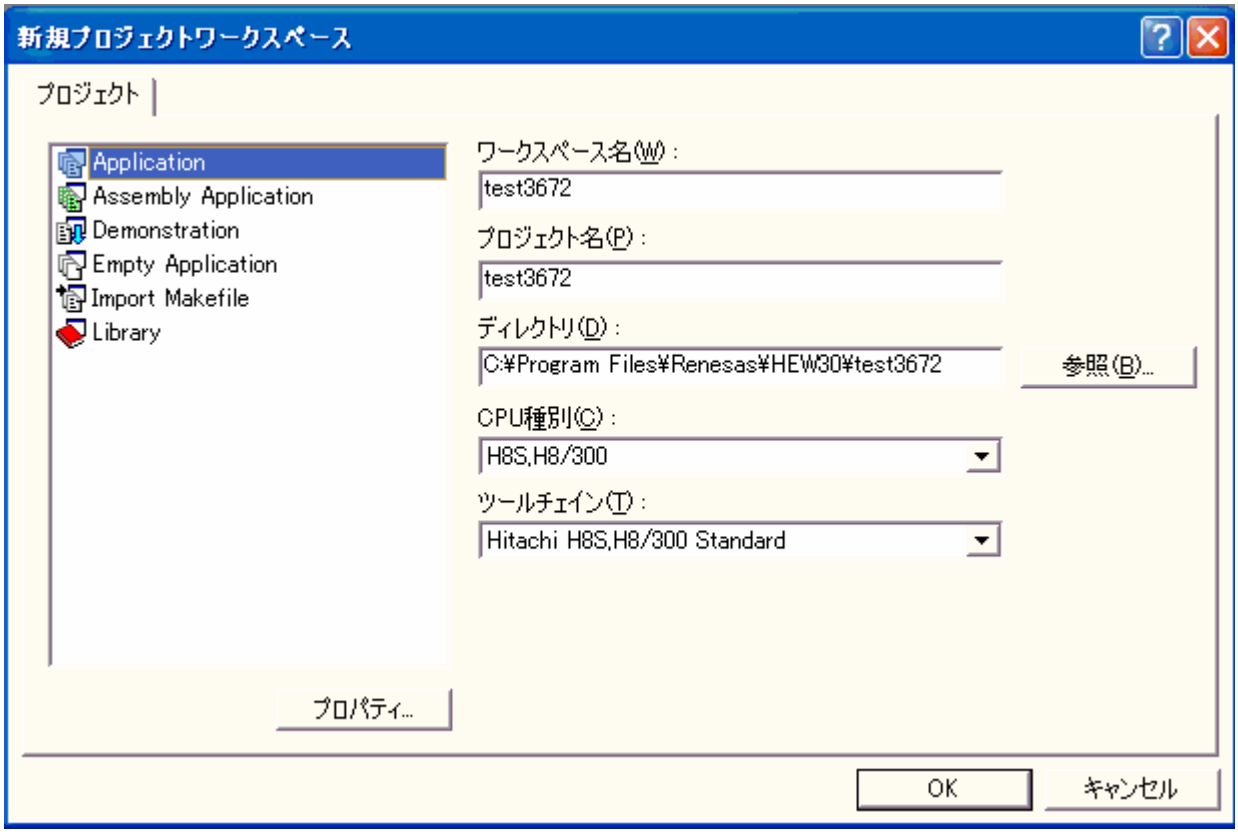
新しくプログラムを製作する場合は「新規プロジェクトワークスペース」を選択します。過去に製作したプログラムの修正、追加等は「最近使用したプロジェクトワークスペースを開く」を選択し、ワークスペースを選択します。「別のプロジェクトワークスペースを参照する」はＨＥＷのディレクトリ外にあるファイルを参照できます。

ここでは「新規プロジェクト、、」を選択し、ＯＫをクリックします。

●ワークスペース名、プロジェクト名を記入します。同じでも、異なっていてもよいようです。

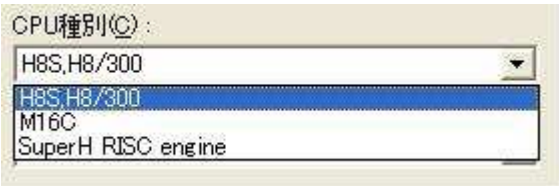
ＣＰＵ種別はＨ８Ｓ、Ｈ８／３００を選択してください。

ツールチェインはＨｉｒａｃｈｉ　Ｈ８Ｓ、Ｈ８／３００　Ｓｔａｎｄａｒｄを選択してください。



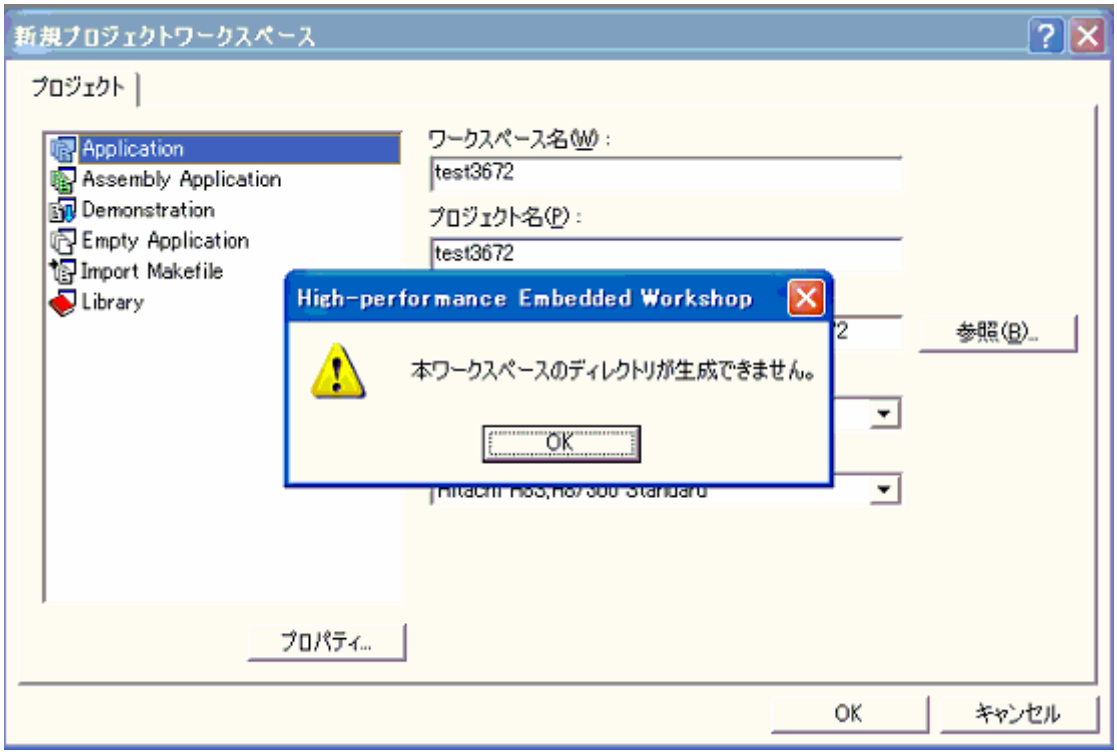
デホルトではソースファイル等はC：￥Ｐｒｏｇｒａｍ　Ｆｉｌｅｓ￥Ｒｅｎｅｓａｓ￥ＨＥＷ３０￥に置かれます。本例ではパソコンに既にＨＥＷ3がインストールされていたためこうなるようです。ＨＥＷのバージョンアップは現存するＨＥＷｘｘのディレクトリ中に上書きされるようです。新規にダウンロードしてインスツールした場合、C:\ＷｏｒｋＳｐａｃｅに置かれるようです。

●ＣＰＵ種別はＨ８Ｓ、Ｈ８／３００を選択します。他にＭ１６Ｃ（旧三菱系マイコン）やＳｕｐｅｒＨ　ＲＩＳＣ　ｅｎｇｉｎｅ（ＳＨマイコン）の開発が行える設定になっています。ＨＥＷで開発できるマイコンの種類はもっと多岐に渡ります。



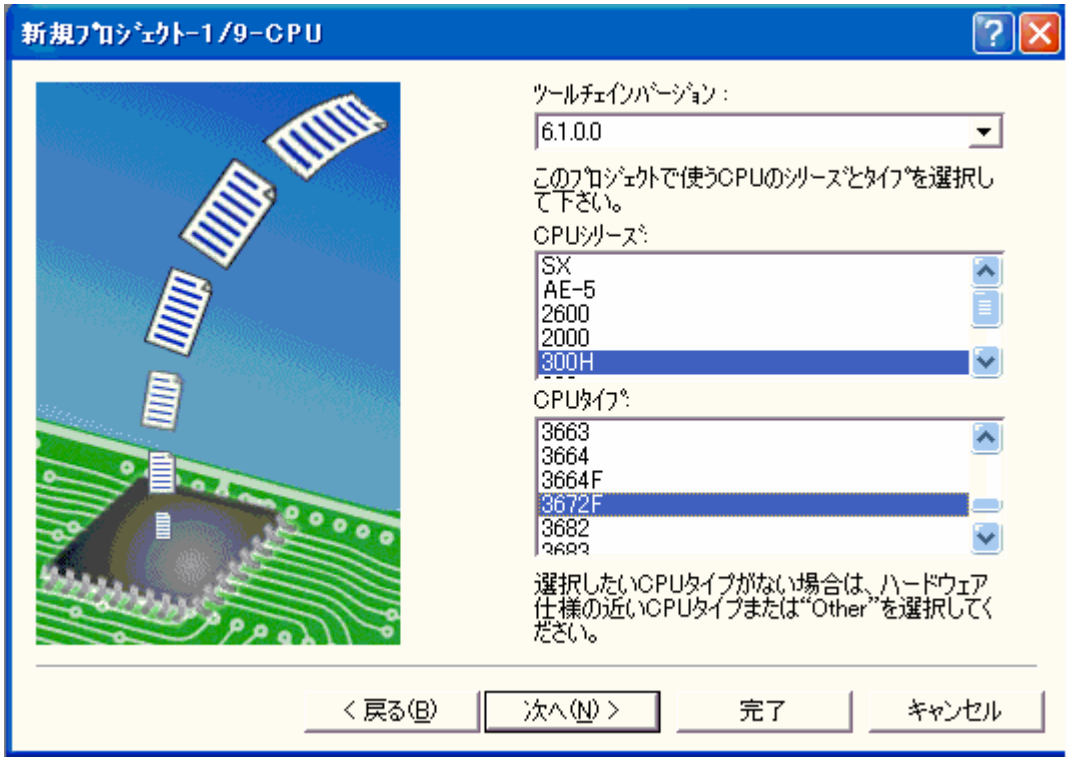
●ツールチェインはＨｉｔａｃｈｉ　Ｈ８Ｓ、Ｈ８／３００　Ｓｔａｎｄａｒｄを選択します。他にＫＰＩＴ社のＧＮＵ　Ｈ８が使用できるようです。ＫＰＩＴ社はＬｉｎｕｘ上で動作するＧＣＣ　Ｃコンパイラをｗｉｎｄｏｗｓ上で使用できるように変換して無料でダウンロードできるようにしている会社です。このＣコンパイラの動作はＨＥＷがインストールされていることが前提のようです。ＨＥＷ４の英語バージョンはＫＰＩＴ社のサイトからダウンロードできます。

●下例「本ワークスペースのディレクトリが生成できません。」というエラーはワークスペース名が既にディレクトリにある場合のエラーです。名称を変えてください。



ここでは「ｔｅｓｔ３６７２ａ」と変更しました。

●変更後、ＯＫをクリックすると以下の画面が現れます。
ＣＰＵシリーズは３００Ｈを選択します。
ＣＰＵタイプは３６７２Ｆを選択します。



●ここで「次へ」を選択するとより詳細なＨＥＷの設定ができますが、ここでは必要ないので、「完了」をクリックしてください。以下の画面が現れます。

●今までの設定の確認ができます。また、Ｒｅａｄｍｅ．ｔｘｔというファイル名で保存されるため後からの確認も容易です。以下の項目等、入力と相違ないか確認します。

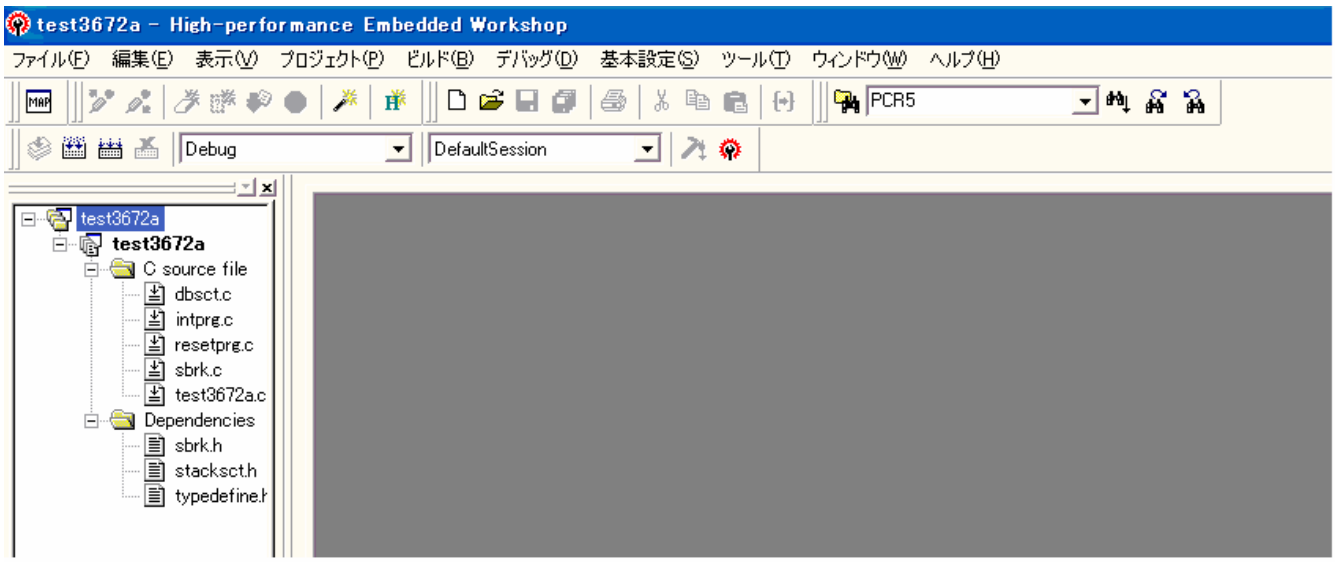
PROJECT NAME : test3672a
PROJECT DIRECTORY : C:\Program Files\Renesas\HEW30\test3672a\test3672a

CPU SERIES :　　300H

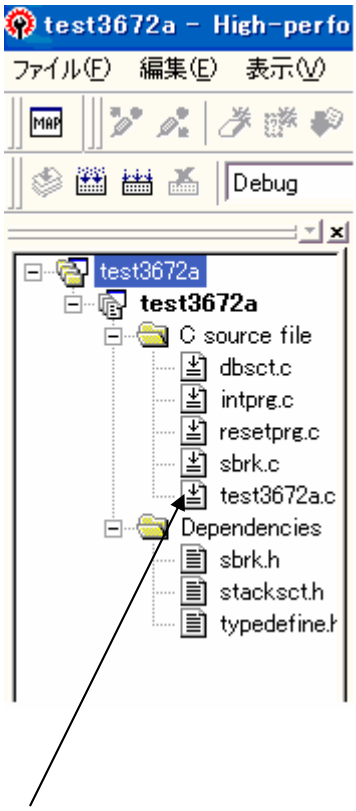
CPU TYPE :3672F

TOOLCHAIN NAME :　　Hitachi H8S,H8/300 Standard Toolchain

●問題なければＯＫをクリックすると下図のような画面が表示されます。

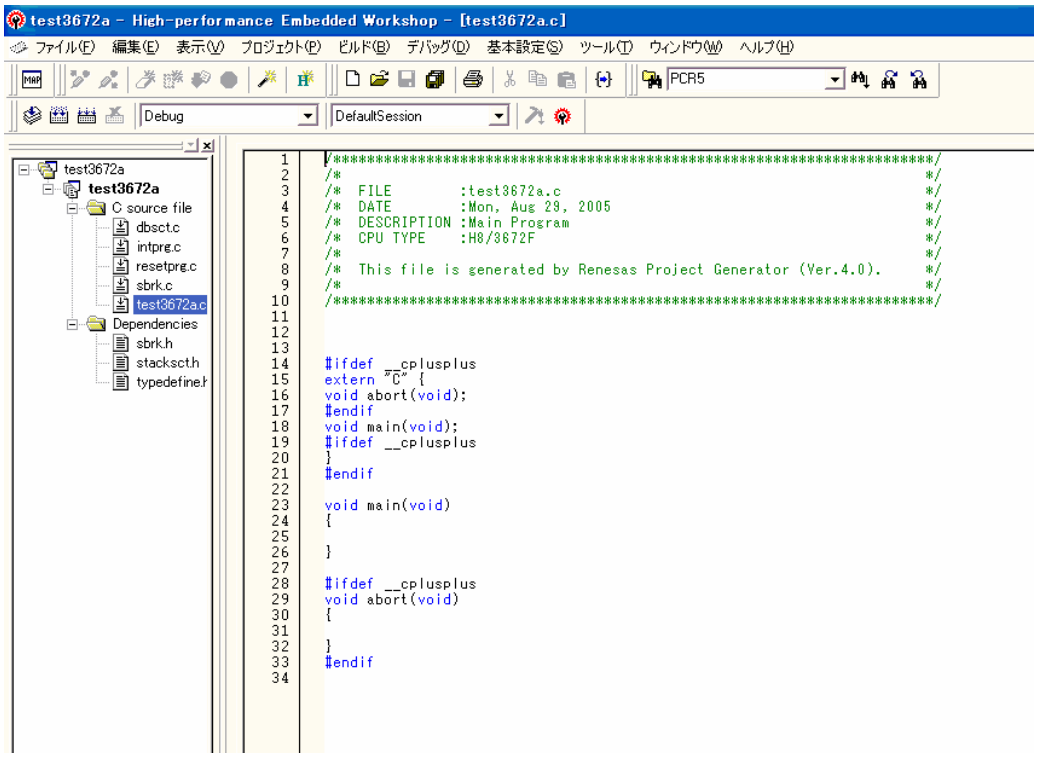


●ここまでですでに以下のファイルが自動的に生成されています。ユーザーがプログラムを書き込むのは「ｔｅｓｔ３６７２ａ」です。他のファイルに対する知識は今回は特に必要ありません。ファイル名をダブルクリックするとエディタが開かれます。



このファイルをダブルクリックする。

●「ｔｅｓｔ３６７２ａ」のダブルクリック結果



●「ｔｅｓｔ３６７２ａ．ｃ」の詳細

もっぱら、ｖｏｉｄ　ｍａｉｎ（ｖｏｉｄ）関数内にユーザープログラムを書きます。

```
/*****************************************************************************/
/*                                                                           */
/*  FILE        :test3672a.c                                                 */
/*  DATE        :Mon, Aug 29, 2005                                           */
/*  DESCRIPTION :Main Program                                                */
/*  CPU TYPE    :H8/3672F                                                    */
/*                                                                           */
/*  This file is generated by Renesas Project Generator (Ver.4.0).           */
/*                                                                           */
/*****************************************************************************/


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

void main(void)
{

}

#ifdef __cplusplus
void abort(void)
{

}
#endif

```

ここにユーザープログラムを記入する。

●例としてＨ８－３６７２のＰ１をＯＮ、ＯＦＦさせるプログラムを作成してみます。以下がプログラム全容です。エディタで追記した部分は赤字の部分です。

```
/************************************************************************/
/*                                                                      */
/*  FILE        :test3672a.c                                            */
/*  DATE        :Mon, Aug 29, 2005                                      */
/*  DESCRIPTION :Main Program                                           */
/*  CPU TYPE    :H8/3672F                                               */
/*                                                                      */
/*  This file is generated by Renesas Project Generator (Ver.4.0).      */
/*                                                                      */
/************************************************************************/


#define PCR1 *(volatile unsigned char *)0xffe4)
#define P1   *(volatile unsigned char *)0xffd4)

void wait(int);


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

void main(void)
{

        PCR1 = 0xff;

        while(1)
        {
                P1 = 0;
                wait(500);
                P1 = 0xff;
                wait(500);

        }

}

void wait(int time)
{
int loop;
```

```
        while(time != 0)
        {
                for(loop = 0;loop < 100;loop++)
                {

                }
                time--;
        }

}

#ifdef __cplusplus
void abort(void)
{

}
#endif
```

●順番に説明します。

```
    1  /************************************************************************
    2  /*
    3  /*  FILE        :test3672a.c
    4  /*  DATE        :Mon, Aug 29, 2005
    5  /*  DESCRIPTION :Main Program
    6  /*  CPU TYPE    :H8/3672F
    7  /*
    8  /*  This file is generated by Renesas Project Generator (Ver.4.0).
    9  /*
   10  /************************************************************************
   11
   12
   13  #define PCR1 *((volatile unsigned char *)0xffe4)
   14  #define P1   *((volatile unsigned char *)0xffd4)
   15
   16  void wait(int);
   17
   18
   19  #ifdef __cplusplus
   20  extern "C" {
   21  void abort(void);
   22  #endif
   23  void main(void);
   24  #ifdef __cplusplus
   25  }
   26  #endif
   27
   28  void main(void)
   29  {
   30
   31      PCR1 = 0xff;
   32
   33      while(1)
   34      {
   35          P1 = 0;
   36          wait(500);
   37          P1 = 0xff;
   38          wait(500);
   39
   40      }
   41
   42  }
```

●Ｐ１のレジスタの絶対値アドレス等を設定しています。

```
13 #define PCR1   (*(volatile unsigned char ...
14 #define P1     (*(volatile unsigned char ...
```

使用するＰ１の絶対アドレスを書きます。これを書くことにより、以降　Ｐ１と書けば０ｘｆｆｄ４のアドレスに自動的に変換されます。つまり＃ｄｅｆｉｎｅとはＰ１＝０ｘｆｆｄ４という意味です。
途中の　ｖｏｌａｔｉｌｅはコンパイラ最適化のときに無視されないように書いています。無条件で書いておけば間違いありません。
unsigned　ｃｈａｒ　はポートが８ビットで００～０ｘｆｆまで扱う、という宣言です。

さて、実はコンパイラは　ｉｏｄｅｆｉｎｅ．ｈ　というヘッダファイルが各ＣＰＵ毎に用意されています。基本的には＃ｉｎｃｌｕｄｅ　”ｉｏｄｅｆｉｎｅ．ｈ”と宣言するだけで上記のような記入無しでＰ１等使用できるようになります。理由はｉｏｄｅｆｎｅ．hの中でＰ１の定義がなされているからです。しかし、その定義が普段自分が使い慣れた形でない場合や、定義もれがあって十分にプログラムが書けない場合があります。そこで本例ではあえてｉｏｄｅｆｉｎｅ．hを使用していません。使い方に興味のある方は別ファイル「ＨＥＷを使用したＨ８－３０５２ＣＰＵボード開発方法」をダウンロードしていただければ使用例を見ることができます。

●ｖｏｉｄ　ｗａｉｔ（ｉｎｔ）

Ｐ１をＯＮ、ＯＦＦさせる時間間隔をこのウエイトプログラムで決めています。その関数宣言で、書かない場合、ｍａｉｎ関数より前にｗａｉｔ関数を書けばエラーになりません。

●プログラムの部分です。

```
void main(void)
{
        PCR1 = 0xff;

        while(1)
        {
                P1 = 0;
                wait(500);
                P1 = 0xff;
                wait(500);

        }

}
```

ＰＣＲ１＝０ｘｆｆはＰ１ポートを全て出力として設定しています。後はＰ１＝０を出力し、ｗａｉｔ（５００）時間待ちます。Ｐ１＝０ｘｆｆを出力し再びｗａｉｔ（５００）時間待ちます。ｗｈｉｌｅ（１）はその括弧中を無限に繰り返します。
Ｐ１をオシロスコープで確認すれば’０’、’１’＝０Ｖ、５Ｖと繰り返しているはずです。またｗａｉｔ（５００）の引数５００を小さくすれば時間間隔が短く、多くすれば長くなるはずです。

●ウエイト（時間待ち）プログラムです。

```
29  void main(void)
30  {
31
32      PCR1 = 0xff;
33
34      while(1)
35      {
36          P1 = 0;
37          wait(500);
38          P1 = 0xff;
39          wait(500);
40
41      }
42
43  }
44
45  void wait(int time)
46  {
47  int loop;
48
49      while(time != 0)
50      {
51          for(loop = 0;loop < 100;loop++)
52          {
53
54          }
55          time--;
56      }
57
58  }
59
```

プログラムはｔｉｍｅが0になるまで繰り返します。ｌｏｏｐが0から９９まで変化しまが、そのたびにｔｉｍｅは1引かれます。ｔｉｍｅの引数は５００ですから、４９９，４９８，４９７、、、とｔｉｍｅ－－により減算されるはずです。

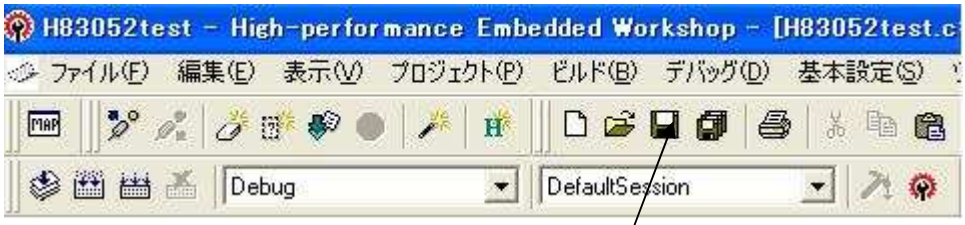
```
void wait(int time)
{
int loop;

        while(time != 0)
        {
                for(loop = 0;loop < 100;loop++)
                {

                }
                time--;
        }

}
```

ここまで書けたら一度ファイルをセーブしておきます。

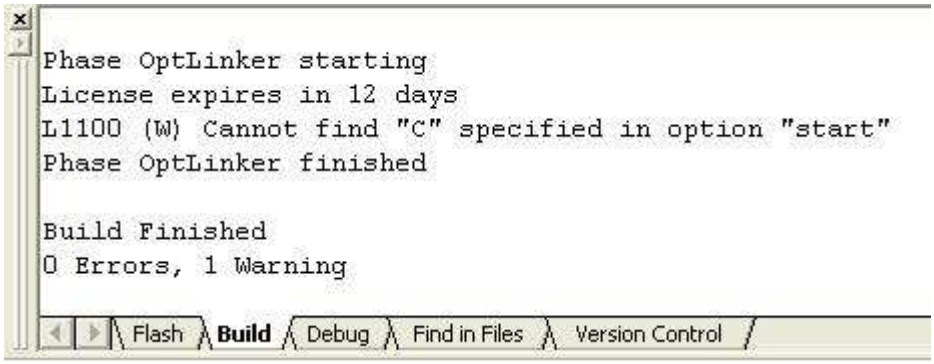


ここをクリックしてセーブ

●コンパイル（ビルド）します。

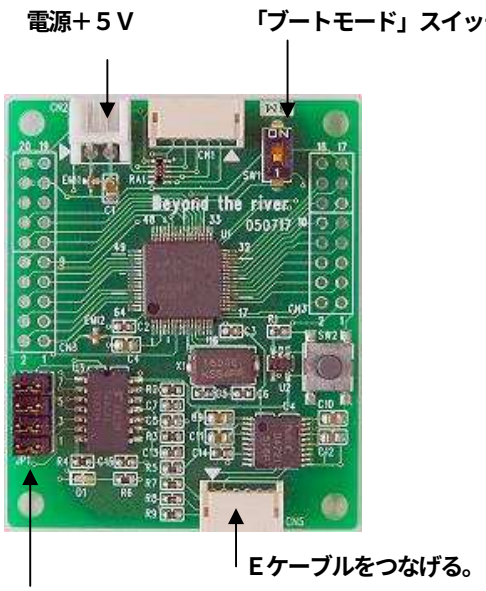
●コンパイルはリンクしません。プログラム書き始めの頃（エラーが多い場合）はコンパイルでエラー確認したほうがデバックが早いです。
●ビルドはリンクまで行いＲＯＭ書き込み可能ファイル形式　ＭＯＴファイルを出力します。最終的にはビルドを行いＲＯＭに書き込みます。

ビルドの結果、０　Ｅｒｒｏｒｓ、１　Ｗａｒｎｉｎｇの結果が得られればＯＫです。（下図）



ここで１　Ｗａｒｎｉｎｇとはデホルトでメモリマップに指定されているセクションの一つであるｃが使用されていないことによるもので、本件では気にする必要はありません。ユーザが書いたプログラムに問題がある場合、ここから数値が増えることになります。これは修正し、０エラー、１ワーニングの状態までもっていく必要があります。ここまできたらフォースライタで作成したプログラムをＲＯＭに書き込みます。

●書き込み事前準備　「ＢＣＨ８３６７２」　ＣＰＵボードの設定



ＪＰ１　１－２，３－４のショートバーを抜く

●パソコンとＣＰＵボード間の接続を確認します。

前提条件として、

１．先に指定したパソコンのＣＯＭポートからＣＰＵボードにＥケーブルが接続されている必要があります。

２．「ＢＣＨ８３６７２」ＣＰＵボードはブートモードに設定されていて、電源が投入されている必要があります。具体的には

あ）ＪＰ１のショートピン　４つありますが、１－２、３－４間のショートバー２つを抜いてください。５－６、７－８間はショートのままです。

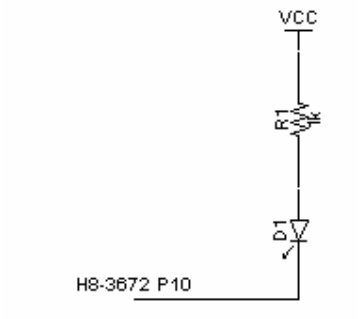
い）ブートモードスイッチをＯＮにしてください。

う）上記までできましたら電源を入れてください。赤いＬＥＤが点灯します。しない場合上記を再度確認してください。

書き込むプログラムを選択します。さきほどビルドしたＨ８３６７２ａ．ｍｏｔファイルは普通はＣ：￥ＷｏｒｋＳｐａｃｅ￥の中にあります。フォースライタでのＢＣＨ８３６７２への書き込みについては別ファイルをダウンロード願います。　デバック、動作中「ＢＣＨ８３６７２」の基板上の赤いＬＥＤは点灯しっぱなしです。Ｅケーブルを抜くと通電中ＬＥＤは消灯します。

●波形観測

ＲＯＭに焼いたら動作していますから、たとえば下記のような回路をポートに接続すればウエイトありの場合、ＬＥＤが点滅するのが確認できます。

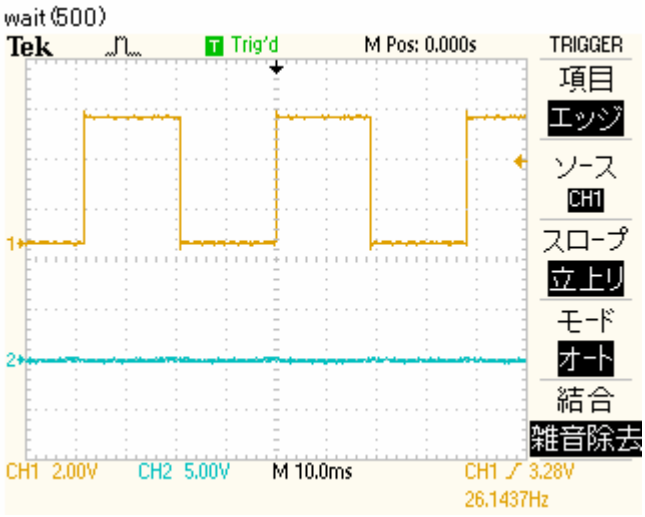


●Ｐ１０観測波形（上のオレンジ色の波形、下の青の波形は本件には関係ありません）

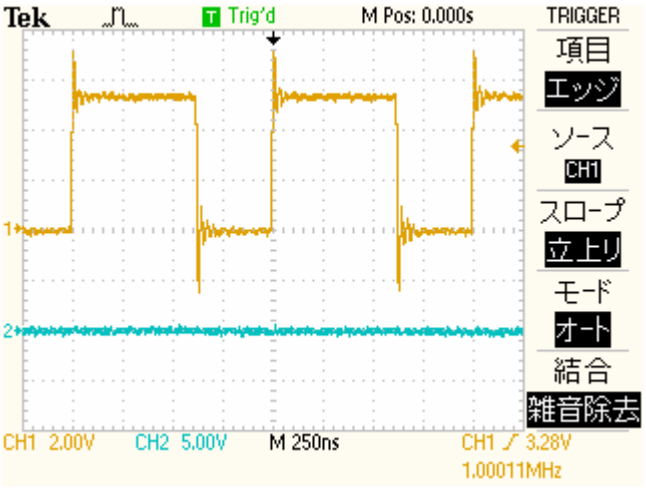
㊤：ウエイト（５００）　　２６．１４３７Ｈｚの波形が得られている。ＬＥＤ等でも確認できる。

㊦：ウエイトなし　　約１ＭＨｚの波形が得られている。ソフトウエア（Ｃ言語）で作れる発振周波数の限界付近。

㊤



㊦ウエイトなし



ご注意

■ＨＥＷは株式会社ルネサステクノロジ社の商標登録です。

■Ｗｉｎｄｏｗｓは米国マイクロソフト社の登録商標です。

■ＧＣＣ（ＧＮＵ　Ｃｏｍｐｉｌｅｒ　Ｃｏｌｌｅｃｔｉｏｎ）Ｃ　はＧＮＵプロジェクトによるフリーＣコンパイラです。

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 1141-1

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