

HEW4を使用したH8 - 3052CPUボード開発の仕方

初版	2005.8.30
第2版	2005.11.3 FDT切り離し
第3版	2006.3.28 iodef.h未使用

ここではHEWの立ち上げ、プログラム作成、ROM化のファイル作成の方法を示します。HEWはあらかじめダウンロード、インストールしてあることが前提です。

HEWを立ち上げると下図のような表示があります。



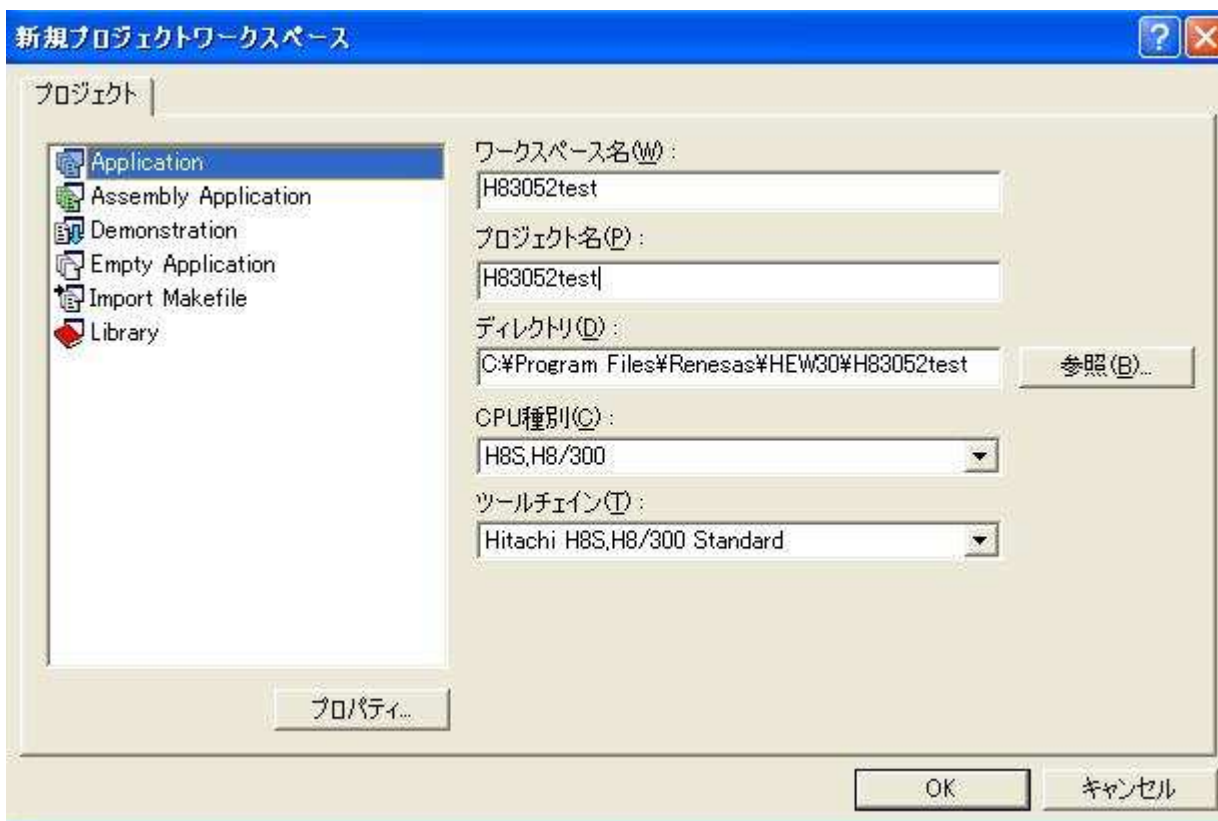
新しくプログラムを製作する場合は「新規プロジェクトワークスペース」を選択します。過去に製作したプログラムの修正、追加等は「最近使用したプロジェクトワークスペースを開く」を選択し、ワークスペースを選択します。「別のプロジェクトワークスペースを参照する」はHEWのディレクトリ外にあるファイルを参照できます。

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。

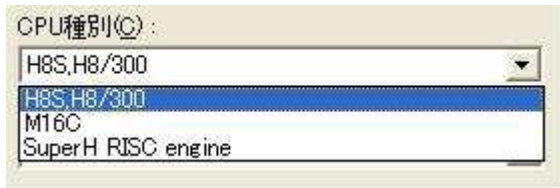
ツールチェーンはHitachi H8S、H8/300 Standardを選択してください。



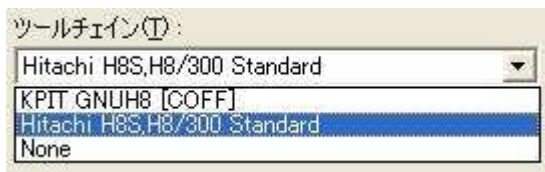
デフォルトではソースファイル等はC:\Program Files\Renesas\HEW30\に置かれます。本例ではパソコン

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トールした場合、C:\WorkSpace に置かれるようです。

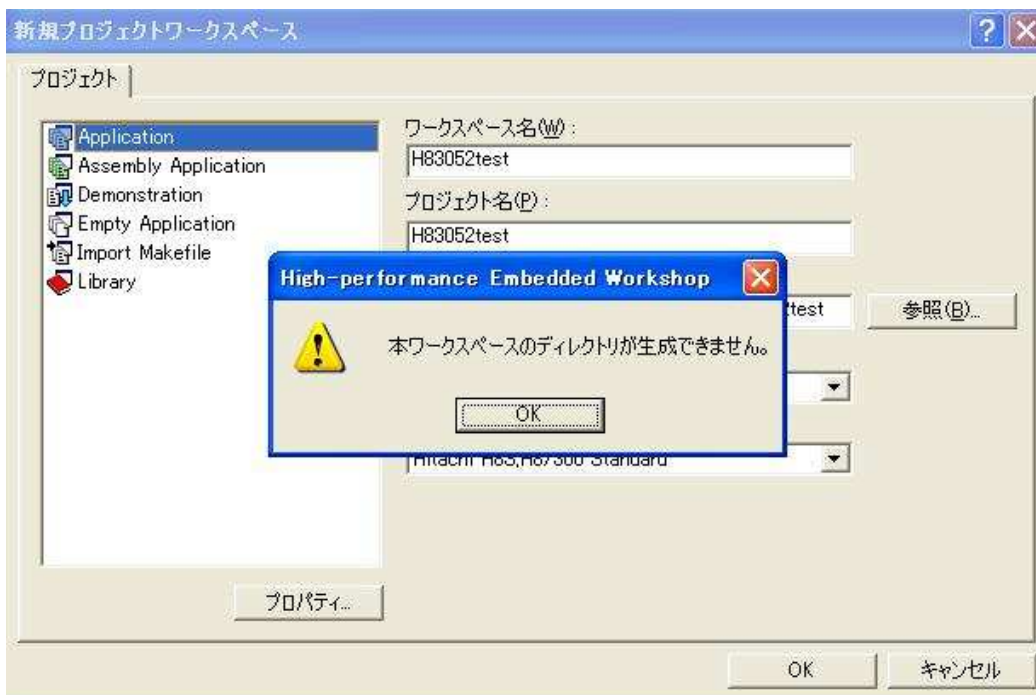
CPU種別はH8S、H8/300を選択します。他にM16C(旧三菱系マイコン)やSuperH RISC engine(SHマイコン)の開発が行える設定になっています。HEWで開発できるマイコンの種類はもっと多岐に渡ります。



ツールチェーンは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。



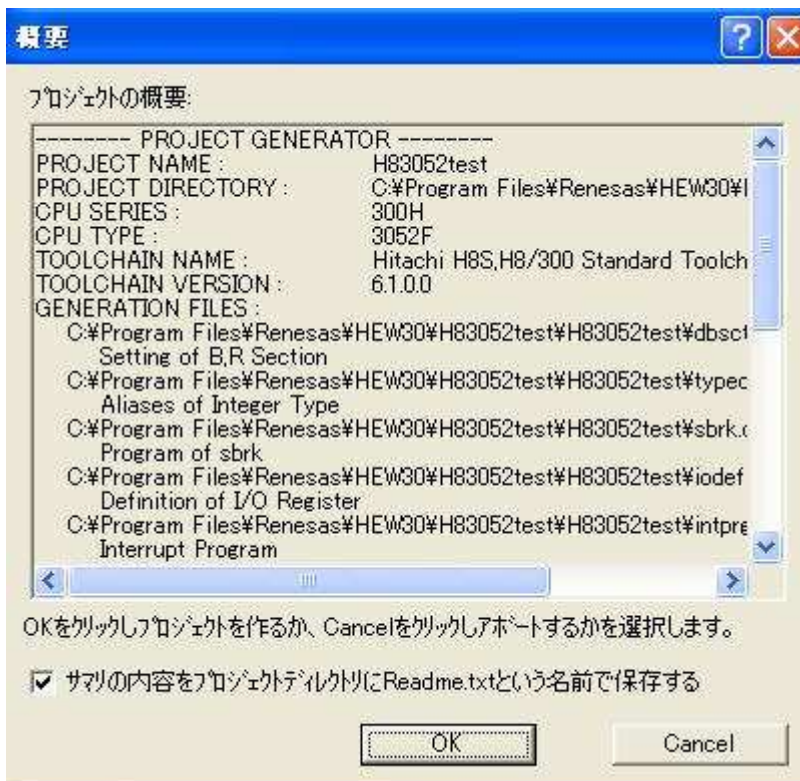
OKをクリックすると以下の画面が現れます。

CPUシリーズは300Hを選択します。

CPUタイプは3052Fを選択します。



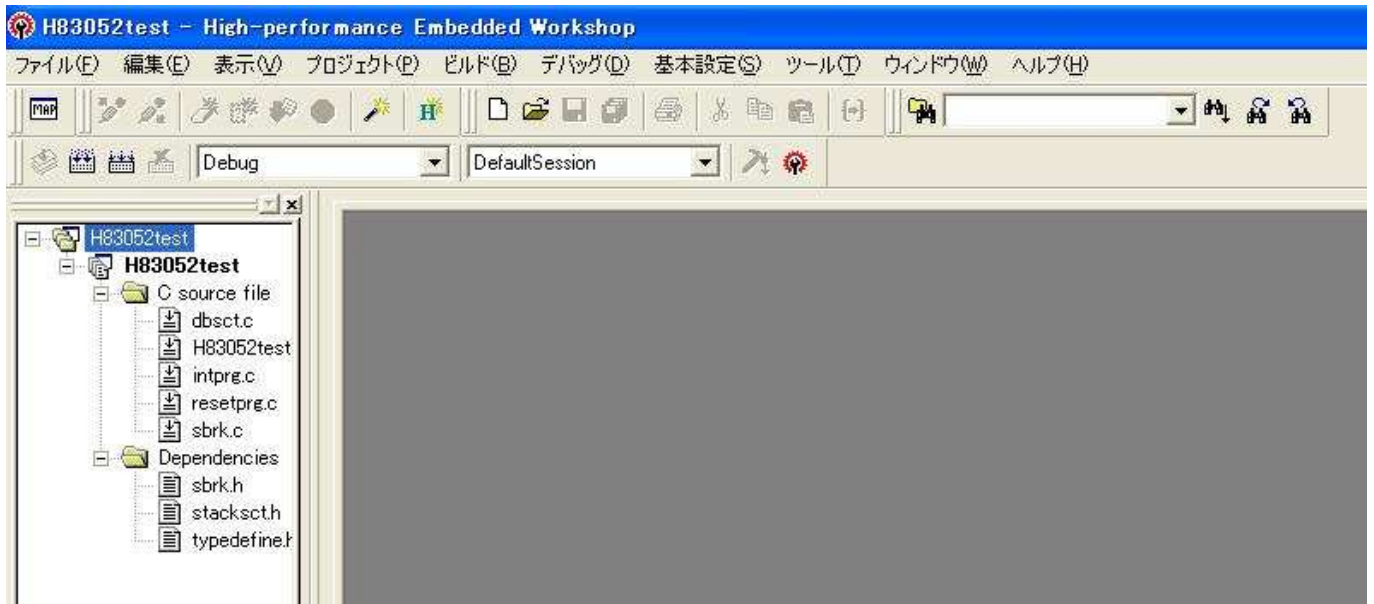
ここで「次へ」を選択するとより詳細なHEWの設定ができますが、ここでは必要ないので、「完了」をクリックしてください。以下の画面が現れます。



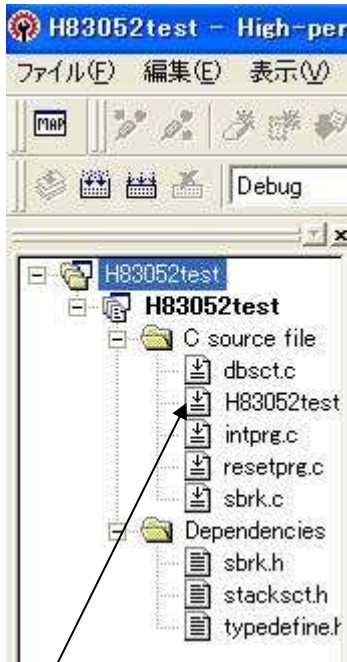
今までの設定の確認ができます。また、Readme.txt というファイル名で保存されるため後からの確認も容易です。以下の項目等、入力と相違ないか確認します。

```
PROJECT NAME :      H83052test
CPU SERIES :      300H
CPU TYPE :      3052F
```

OKをクリックすると下図のような画面が表示されます。

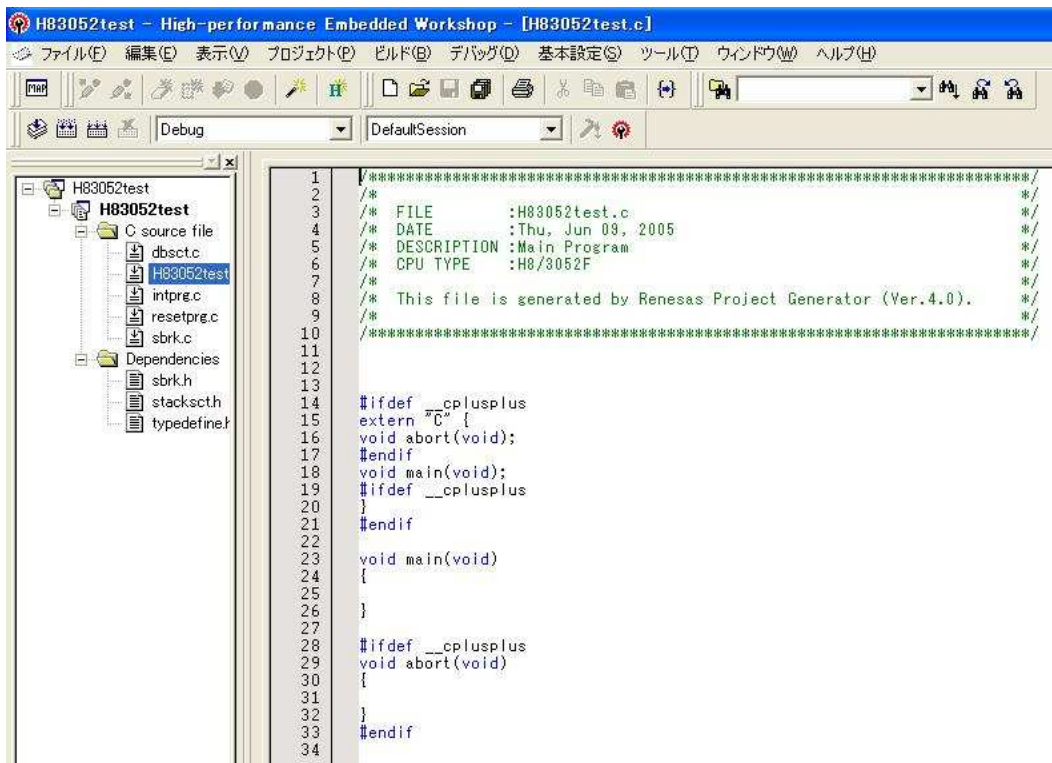


ここまでです。以下のファイルが自動的に生成されています。ユーザーがプログラムを書き込むのは「H83052test」です。他のファイルに対する知識は特に必要ありません。ファイル名をダブルクリックするとエディタが開かれます。



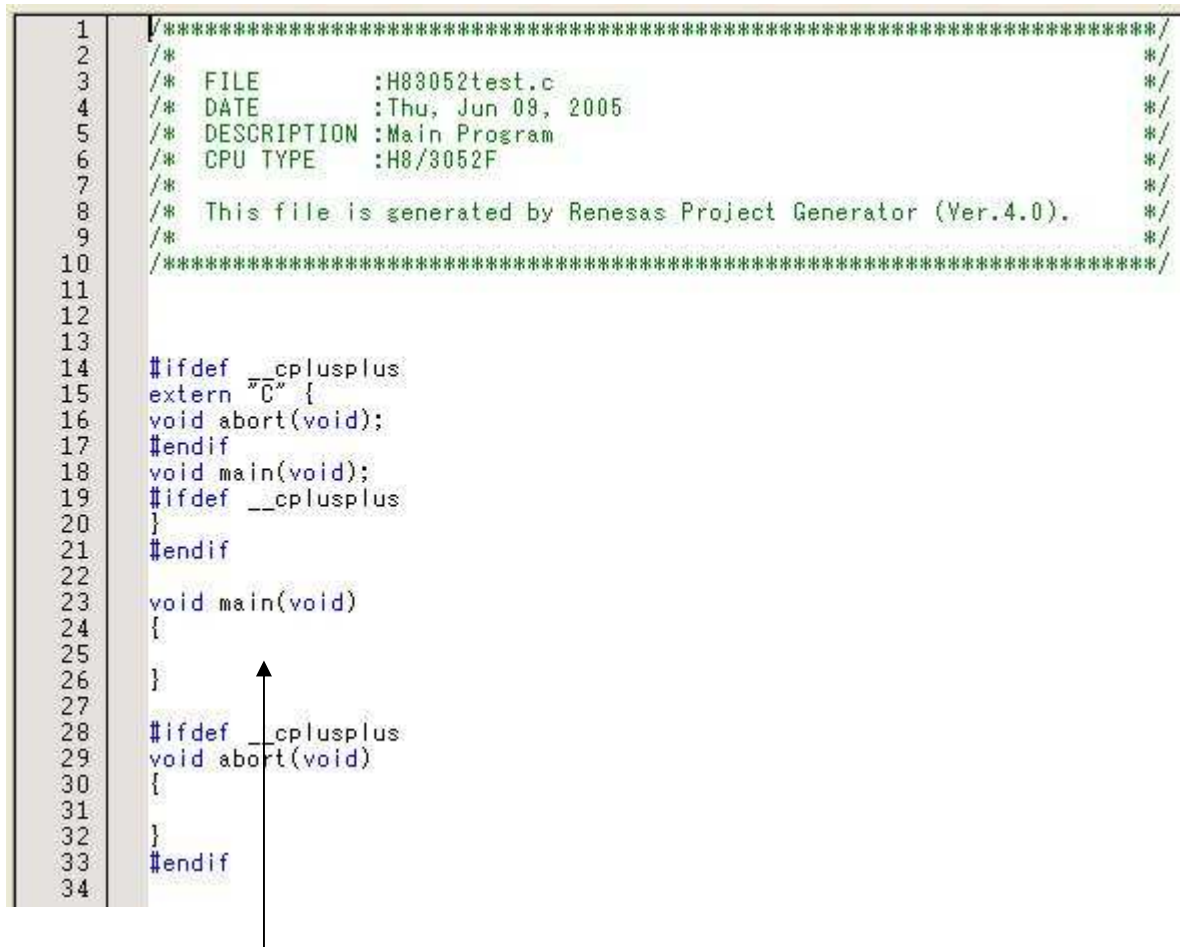
このファイルをダブルクリックする。

「H83052test」のダブルクリック結果



「H83052test.c」の詳細

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



ここにユーザープログラムを記入する。

例としてH8 - 3052のP6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以下の行番号です。

```

14 #define P6DDR (*(volatile unsigned char ...)
15 #define P6     (*(volatile unsigned char ...)
****
28 P6DDR = 0xff;
29 while(1){
30     P6 = 0;
31     P6 = 0xff;
32 }

```

```

1  /*****
2  /*
3  /* FILE      :H83052test.c
4  /* DATE      :Thu, Jun 09, 2005
5  /* DESCRIPTION:Main Program
6  /* CPU TYPE  :H8/3052F
7  /*
8  /* This file is generated by Renesas Project Generator (Ver.4.0).
9  /*
10 /*****
11
12 #include "iodefine.h"
13
14 #define P6DDR (*(volatile unsigned char *)0xFFFFC9) /* P6DDR Address */
15 #define P6     (*(volatile unsigned char *)0xFFFFCB) /* P6 Address */
16
17 #ifdef __cplusplus
18 extern "C" {
19 void abort(void);
20 #endif
21 void main(void);
22 #ifdef __cplusplus
23 }
24 #endif
25
26 void main(void)
27 {
28     P6DDR = 0xff;           //P6を出力ポートに設定
29     while(1){
30         P6 = 0;           //P6に0を出力
31         P6 = 0xff;       //P6に0xffを出力
32     }
33
34 }
35
36 #ifdef __cplusplus
37 void abort(void)
38 {
39
40 }
41 #endif
42

```

順番に説明します。

P6のレジスタの絶対値アドレス等を設定しています。

```

14 #define P6DDR (*(volatile unsigned char ...)
15 #define P6     (*(volatile unsigned char ...)

```

P6のアドレス定義を書きます。

プログラムの部分です。

```
28 P6DDR = 0xff;  
29 while(1){  
30     P6 = 0;  
31     P6 = 0xff;  
32 }
```

28行はP6のポートを全て出力に設定しています。

29行は無限ループです。

30行はP6に0を出力しています。

31行はP6に0xffを出力しています。各ビットで見ると1を出力しています。

30行と31行が無限ループですので、P6のポートが0と1を繰り返します。

ここまで書けたら一度ファイルをセーブしておきます。(各アイコンの位置はHEWのバージョン、画面設定により異なりますが、絵が同じであれば動作も同じです)



ここをクリックしてセーブ

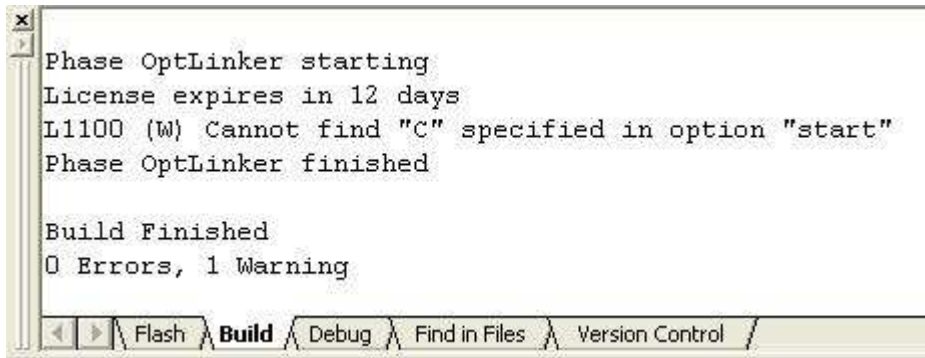
コンパイル(ビルド)します。



ビルド
コンパイル

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。初めてのビルドは数分待たされることがあります。

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



```
Phase OptLinker starting
License expires in 12 days
L1100 (W) Cannot find "C" specified in option "start"
Phase OptLinker finished

Build Finished
0 Errors, 1 Warning
```

ここで 1 Warning とはデフォ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。

ここまでできたら後はプログラムを CPU 内蔵フラッシュROM に書き込み、動作させてみます。

別ファイルで

フォースライタ「FWR3052」での書き込み方

ルネサステクノロジ社純正「FDT」による書き込み方

両方の解説がありますので参考にして書き込んでみてください。

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