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 iodefine.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、インストゥ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。

ようこそ!		? 🛛
ーオプション	ŧ.	ОК
	◎ 新規プロジェクトワークスペースの作成(○)	キャンセル
- Q	○ 最近使用したプロジェクトワークスペースを開く(②):	アドミニストレーション(A)
	C¥Program Files¥Renesas¥HEW30¥H830(	
A A A	○ 別のブロジェクトワークスペースを参照する(B)	

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。

新規プロジェクトワークスペース		? 🛛
新規プロジェクトワークスペース プロジェクト Application の Assembly Application の Empty Application で Import Makefile ・ Library	ワークスペース名(W): [H83052test プロジェクト名(P): [H83052test] ディレクトリ(D): [C¥Program Files¥Renesas¥HEW30¥H83052test] CPU種別(C): [H85,H8/300 ▼ ツールチェイン(T): [Hitachi H85,H8/300 Standard] ▼	② X 参照(B)
ブロパティ	ОК	キャンセル

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HEWx x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ゥ 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(S H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。

H8S,H8/300	
H8S,H8/300	
M16C	
SuperH RISC engine	75

ッールチェインは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がインストゥ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。

ツールチェイン(①:	
Hitachi H8S,H8/300 Standard	*
KPIT GNUH8 [COFF]	
Hitachi H8S,H8/300 Standard	
None	

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。

Application		ワークスペース名(W):		
Assembly Application		H83052test		
Demonstration		プロジェクト名(P)		
	13	H83052test	-	
Library	High-perf	sh-performance Embedded Workshop 🛛 🔀		
		OK III	Ŀ	

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。

新規ブロジュクトー1/9ーCPU	2 🔀
	ツールチェインバージョン
	6.1.0.0
	このプロジェクトで使うCPUのシリーズとタイプを選択し て下さい。 CPUNU_ブ
	AE-5
	2600
	3008
	CPU%/7%
	3048F
	3060
1 2000	3061
	aueo E
Contraction of the second	選択したいCPUタイプがない場合は、ハードウェア 仕様の近いCPUタイプまたは"Other"を選択してく ださい。
< 戻る( <u>B</u> )	次へ(11) > 売了 キャンセル

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。

	SION :	Hitachi H8S,H8/3 6.1.0.0	300 Standard Toolch
3ENERATION FIL C:¥Program Fi C:¥Program Fi Aliases of 1 C:¥Program Fi Program of C:¥Program fi Definition o C:¥Program Fi Interrupt Pr	165 : es¥Renesasi 3,R Section es¥Renesasi nteger Type es¥Renesasi sbrk es¥Renesasi f 1/0 Registi es¥Renesasi ogram	¥HEW30¥H83052tes ¥HEW30¥H83052tes ¥HEW30¥H83052tes ¥HEW30¥H83052tes #HEW30¥H83052tes ¥HEW30¥H83052tes	st¥H83052test¥dbsct st¥H83052test¥typec st¥H83052test¥sbrk.c st¥H83052test¥iodef st¥H83052test¥intpre
<	10		) 🔉

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: CPU SERIES: CPU TYPE:
- H 8 3 0 5 2 t e s t
- ES: 300H : 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「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。



## 「H83052test」のダブルクリック結果

🖗 H83052test - High-performa	ance Embedded Workshop - [H83052test.c]
◇ ファイル(E) 編集(E) 表示(V) ブ	ロジェクト(P) ビルド(B) デバッグ(D) 基本設定(S) ツール(T) ウインドウ(W) ヘルプ(H)
] 🔤    🏏 🍂   🥭 👘 🌒	/ # D # # # # # # # # # # # # # # # # #
😂 🖽 📇 👗 Debug	▼ DefaultSession ▼ 2 @
H83052test H83052test H83052test G Source file H83052test H83052test H83052test H83052test Sorkc Bependencies Sorkh Stackscth Stackscth	<pre> 1</pre>

「H83052test.c」の詳細

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

```
1
 123
                                                                             */ */
      1*
         FILE :H83052test.c
DATE :Thu, Jun 09, 2005
DESCRIPTION :Main Program
      /*
      /*
 4
 5
      18
                                                                             */
      18
 6
         CPU TYPE
                      :H8/3052F
                                                                             */
 7
      18
                                                                             */
 8
      /* This file is generated by Renesas Project Generator (Ver.4.0).
                                                                             */
 9
      1*
                                                                             */
10
      11
12
13
14
      ‡ifdef __cplusplus
extern <sup>™</sup>C<sup>™</sup> {
15
16
17
      void abort(void);
      #endif
18
      void main(void);
19
20
21
22
23
24
25
26
27
28
29
30
      #ifdef __cplusplus
      #endif
      void main(void)
      {
              4
      }
      #ifdef __cplusplus
yoid abort(void)
      {
31
32
33
      #endif
34
```

```
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 }
```

```
123
      18
                                                                       */
      18
         FILE
                    :H83052test.c
                                                                       */
         DATE :Thu, Jun 09, 2005
DESCRIPTION :Main Program
      /*
                                                                       */
 4567
     18
                                                                        */
                    :H8/3052F
      1*
         CPU TYPE
                                                                        */
      1*
 8
      1*
         This file is generated by Renesas Project Generator (Ver.4.0).
                                                                       */
                                                                       */
 9
      1*
10
      11
      #include "iodefine.h"
12
13
14
      #define P6DDR (*(volatile unsigned char *)0xFFFFC9) /* P6DDR Address */
15
      #define P6
                  (*(volatile unsigned char *)0xFFFFCB) /* P6
                                                              Address */
16
     #ifdef __cplusplus
extern "C" {
17
18
      void abort(void);
19
20
21
22
23
      #endif
      void main(void);
      #ifdef __cplusplus
24
25
      #endif
26
27
      void main(void)
      {
28
         P6DDR = 0xff;
                                   //P6を出力ボートに設定
29
         while(1){
30
                    P6 = 0;
                                   //P61こ0を出力
                    P6 = 0 \times ff;
31
                                   //P61こ0xffを出力
         }
32
33
34
      }
35
36
      #ifdef __cplusplus
37
      void abort(void)
38
39
40
      #endif
41
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。

2 8 P6DDR = 0xff; 2 9 while(1){ 3 0 P6=0; 3 1 P6=0xff; 3 2 }

28行はP6のポートを全て出力に設定しています。
 9行は無限ループです。
 0行はP6に0を出力しています。
 1行はP6に0×ff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。
 0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。

🏟 H83	3052te	est – Hie	h-perfo	mance	Embe
@ 7P	イル(E)	編集( <u>E</u> )	表示⊙	プロジェ	クト(円)
MAP		1. 3	# <b>#</b> •	) 🎢	H
		🔏 🛛 Det	bug		-
6	I H8305	i2test			
ב   יב	ビルド ノパイル				

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 O Errors, 1 Warning

 $| \rangle$  Flash  $\lambda$  Build  $\langle$  Debug  $\lambda$  Find in Files  $\lambda$  Version Control

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