

統合環境Hew (Ver 4. 04) 添付スタートアップ関数を使用した場合の
新ワークスペースおよびプロジェクトを登録する方法
(SH7145 BOOT版)

ルネサスC言語用統合環境「Hew Ver 4. 04」で H-debugger 用に新ワークスペース/
プロジェクトを登録する手順方法を説明します。
説明を明確にするために、名前等を仮に決めて例に沿って説明を進めます。

ワークスペース名	ReneSH7145_BOOT_ABS_Hew4		
プロジェクト名	Project		
登録モジュール名	SH7145.c	Cファイル	メインモジュール (アプリ用)
	HewDebugSH2.h	ヘッダファイル	ソフトパーツ用定義ファイル (ソフトパーツを使用しない場合は不要です。)
Hew添付ファイル	Resetprg.c	Cファイル	スタートアップモジュール
	Intprg.c		割込みハンドラモジュール
	Dbsct.c		定数転送用セクション管理宣言
	Vecttbl.c		ベクター定義モジュール
	iodefine.h	ヘッダファイル	I/O 定義ビットフィールド記述用
CPUタイプ	SH7145F		

 【ポイント】

「SH-2 BOOT版」の場合、Hewデフォルト設定から変更に必要な箇所は、

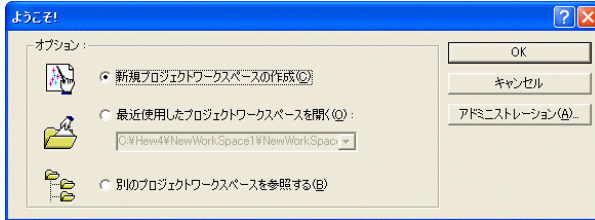
- 1) コンパイラの「最適化」を外す。
- 2) モニタ (ファーム) エリアの確保の為、セクションアドレスを変更をする。

の2点になります。

1. 新ワークスペースの登録方法

“HEW” 起動させます。

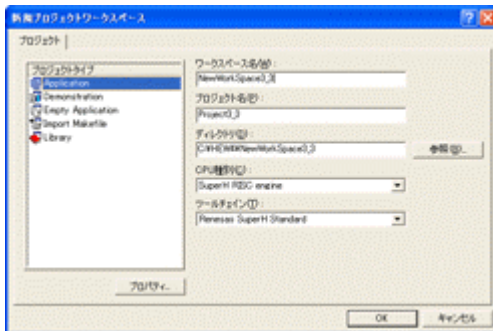
[1-1]



“新規プロジェクトワークスペース”をチェックしての **OK** をクリックする。

もしくは、**キャンセル**後に、[ファイル]-[新規ワークスペース]をクリックします。

[1-2]



ワークスペース名	“ReneSH7145_BOOT_ABS_Hew4”
プロジェクト名	“Project”
ディレクトリ	“C:\Hew4\Renesas”
CPU 種別	“SuperH RISC engine”
ツールチェーン	“Renesas SuperH Standard”
プロジェクト	Application

この項目を確認後、**OK** をクリックして下さい。

[1-3]



CPUシリーズを“SH-2”に選択する。
CPUタイプを“SH7145F”に選択する。

確認後、

次へ> をクリックします。

[1-4]



CPUスペックを確認後

次へ> をクリックして下さい。

[1-5]



- ① I/O ライブラリを使用しませんのでチェックを外して下さい。
- ② ヒープメモリを使用しませんのでチェックを外して下さい。
main() 関数生成は"None"に選択する。
- ③ I/O レジスタ定義ファイルは使用しますのでチェックして下さい。
ハードウェアセットアップ関数生成は "None" に選択する。

確認後、**次へ>**をクリックして下さい。

[1-6]



C言語ライブラリの選択です。この例では、**その他ライブラリを使用しません。**

次へ>をクリックして下さい。

[1-7]



スタックボトムの設定です。**Default** のままでも構いませんが、RAM 使用の節約の為、「**HFFFFFFFC**」にする。

スタックサイズはデフォルト値でよい。

次へ>をクリックして下さい。

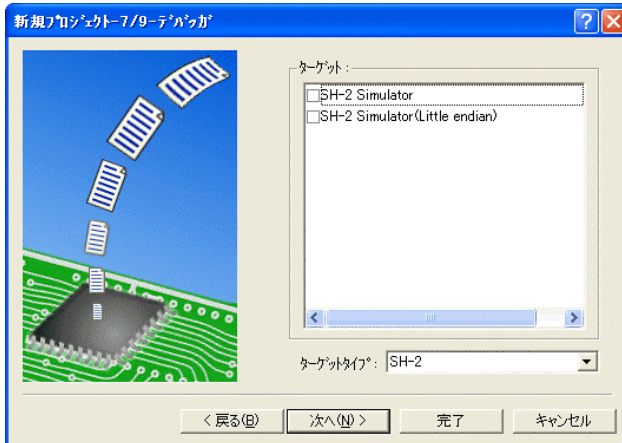
[1-8]



ここで明示されたHeader作成スタートアップ関数を使用しますので、デフォルトの状態で、

次へ>をクリックして下さい。

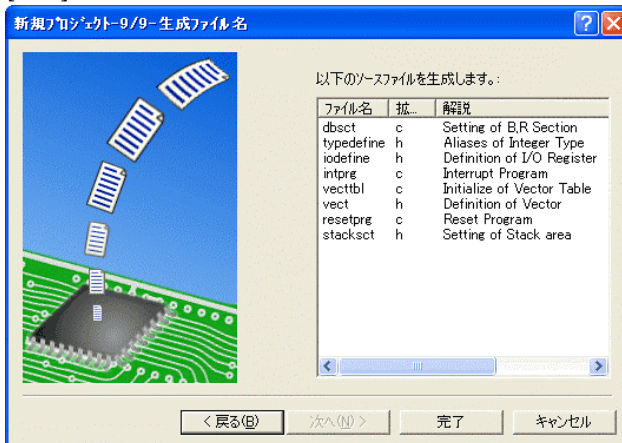
[1-9]



シミュレータの設定ですが使用しませんのでチェック無しの状態で、

次へ>をクリックして下さい。

[1-10]

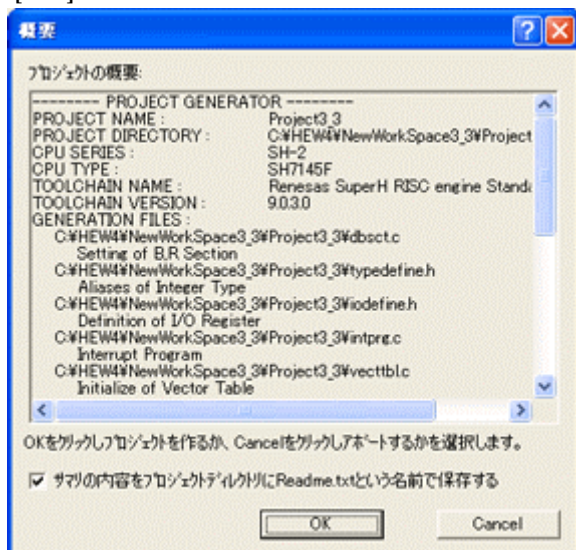


ここで最終になります。

使用するCモジュールを表示します。

この状態で完了をクリックして下さい。

[1-11]



確認画面が表示されますので、

OKをクリックして下さい。

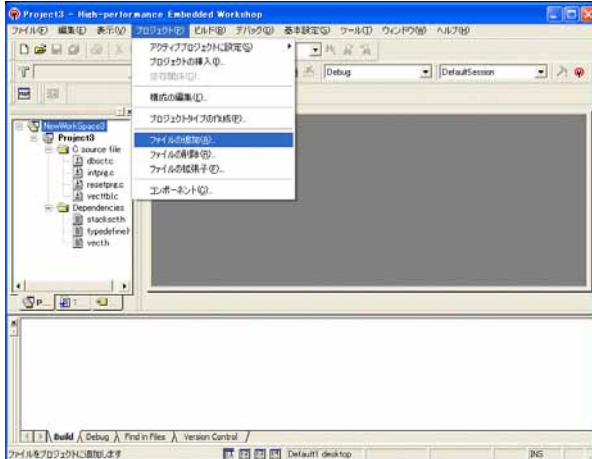
ここまでの操作が新規プロジェクトの登録方法です。

2. プロジェクトに希望モジュール(ソースファイル)を登録する方法

準備: 作成済みの2ファイルを”C:\Hew4\Renesas\ReneSH7145_BOOT_ABS_Hew4\Project”にコピーします。

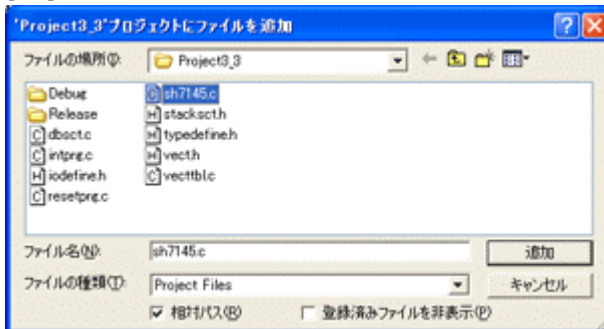
SH7145.c HP よりダウンロードします。(ルネサスC)
HewDebugSH2.h ReneSH7145_BOOT_Hew4.LZH

[2-1]



[プロジェクト]-
[ファイルの追加]をクリックします。

[2-2]



下記1ファイルを指定して下さい。

Sh7145.c

選択後、

追加をクリックします。

この操作によりプロジェクトにモジュールが登録されました。

3 . コンパイラの設定

コンパイラの「最適化」を外す

[3-1]



①[ビルド] -
[H8S, H8/300Standard Toolchain]
をクリックします。

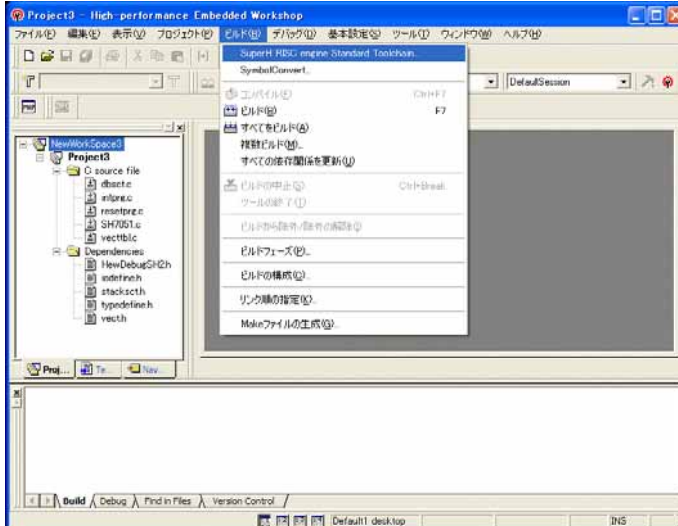
②「コンパイラ」を選択
③カテゴリ「最適化」を選択
④「最適化」のチェックを外す。

⑤OKをクリックする。

4 . ツール(ライブラリ)の設定

HEWは、プロジェクトごとにC言語用ライブラリを作成する仕様になっています。
ライブラリを作成および設定の確認をします。

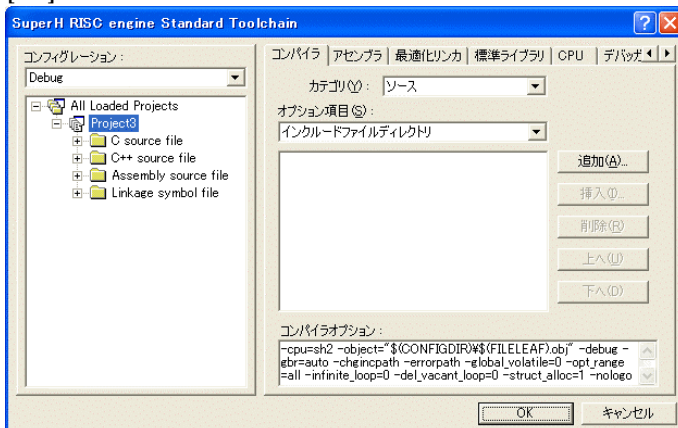
[4-1]



[ビルド]-

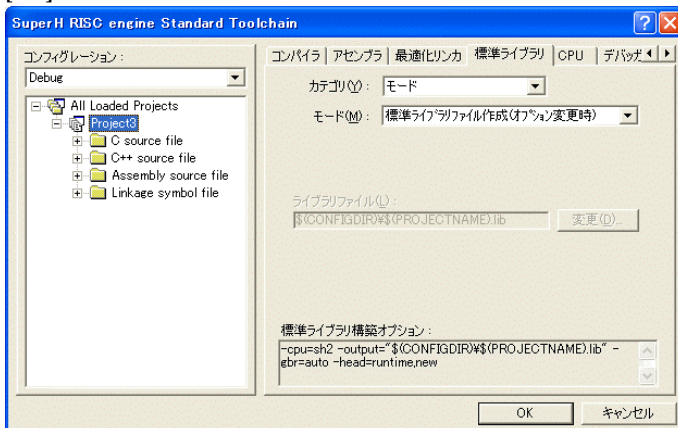
[SuperH RISC engine Standard Toolchain]をクリックします。

[4-2]



「標準ライブラリ」タグをクリックする。

[4-3]



カテゴリのモードが「ライブラリファイル作成 (オプション変更時)」指定になっている事を確認します。

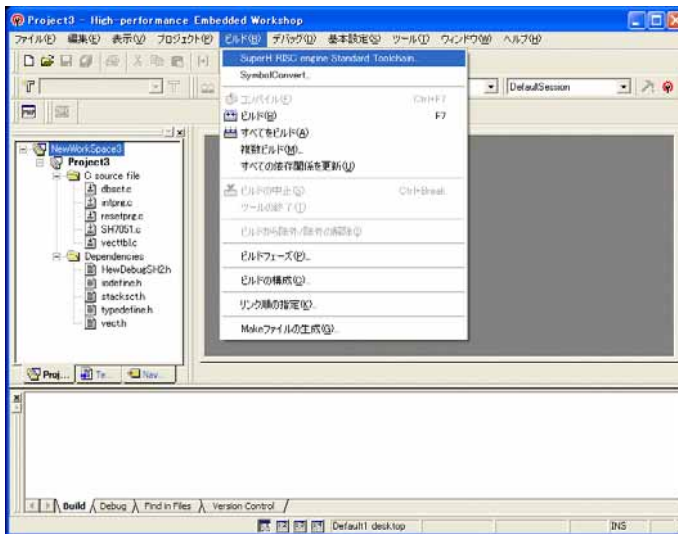
この指定によりオプション変更時のみライブラリを作成する事になります。

デフォルトのまま

OKをクリックする。

5 . ツール(リンカ)の設定

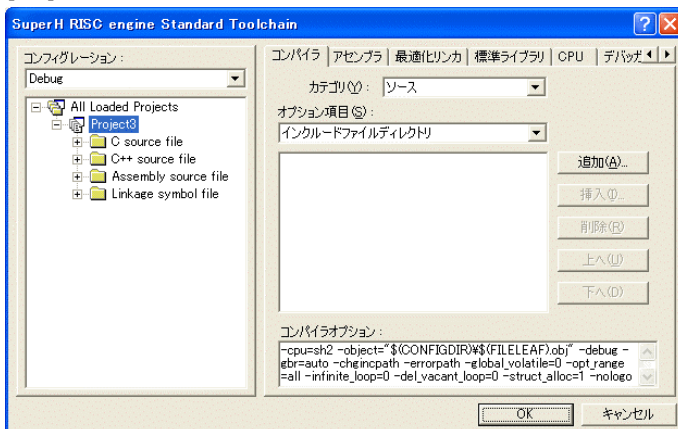
[5-1]



[ビルド]-

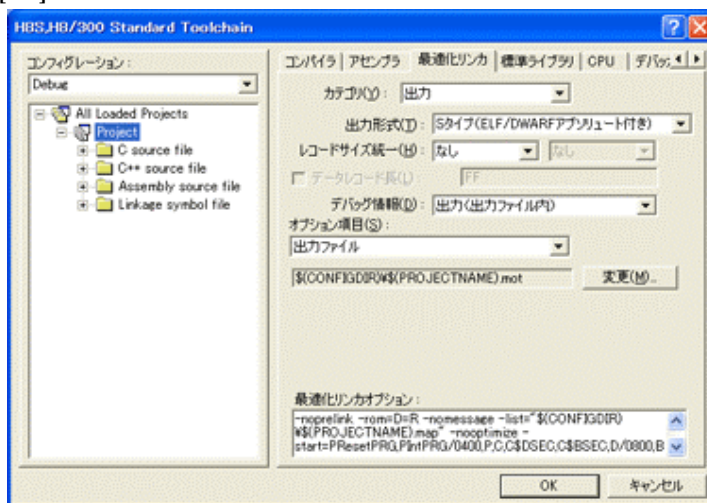
[SuperH RISC engine Standard Toolchain]をクリックします。

[5-2]



「最適化リンカ」タグをクリックする。

[5-3]



①カテゴリの「出力」を選択する。

②出力形式の

「Sタイプ(ELF/DWARFアブソリュート付き)」を確認する。

③デバッグ情報の

「出力(出力ファイル内)」を確認する。

④ オプション項目の

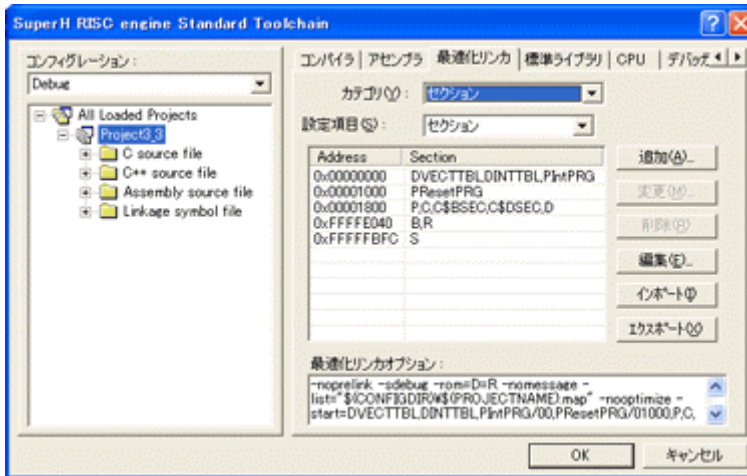
「出力ファイル」

`${CONFIGDIR}$(PROJECTNAME).mot`

を確認する。

全てデフォルトです。

[5-4]



カテゴリの「セクション」を選択する。

下図のようにセクション指定をする。
(モニタエリア確保のため)

OKをクリックする。

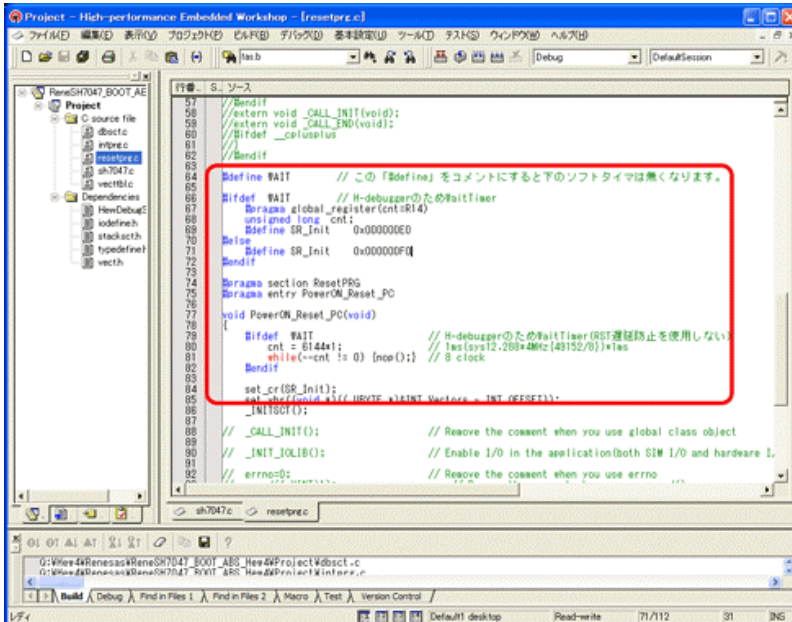
0x00000000	DVECTTBL, DINTTBL, PIntPRG
0x00001000	PResetPRG
0x00001800	P, C, C\$BSEC, C\$DSEC, D
0xFFFFE040	B, R
0xFFFFFBFC	S

(重要) 「Bセクション」0xFFFFE040 (内臓RAM先頭 0xFFFFE000) は、ソースブレイク使用又は、モニタワーク方式 (固定番地) の場合です。ソースブレイク用エリアとモニタワーク方式 (固定番地) のエリアを共用していますので、DEFバージョン6.50xから、ソースブレイクを使用したい場合は、モニタワーク方式 (スタック) に選択して下さい。

6 . スタートアップおよびベクターの変更と追加

1) <resetprg. c>の変更

[6-1]



①左図のようなソース行を追加しますとリセット時ソフトタイマ挿入になります。

このソースを追加することにより④項のメリットがあります。

②デバッグ中は、割り込みマスクレベルを14にします。「2) 項参照」

③ソフトタイマが不必要な場合は「#define WAIT」をコメントにしてください。

④ソフトタイマによるメリット

- DEFの「R s t M o n」操作時「m a i n ()」プログラムまで走行しないため初期設定等による内部レジスタが汚れない。
この例ですと **1ms** ですがリセット遅延が無い場合は **CPU 設定** でリセット遅延なしの指定をすれば「**20us**」タイマ値で機能します。
- S P 値がベクタ1の設定値のままである。

2) ブレークおよびトレース/ステップ実行について (マスクレベルを14にする理由)

UBCブレーク割り込みのプライオリティは、15になっていますのでCPUのSRレジスタの割り込みマスクレベルを14以下に設定する必要があります。下記方法のどちらかを都合に応じて実施して下さい。

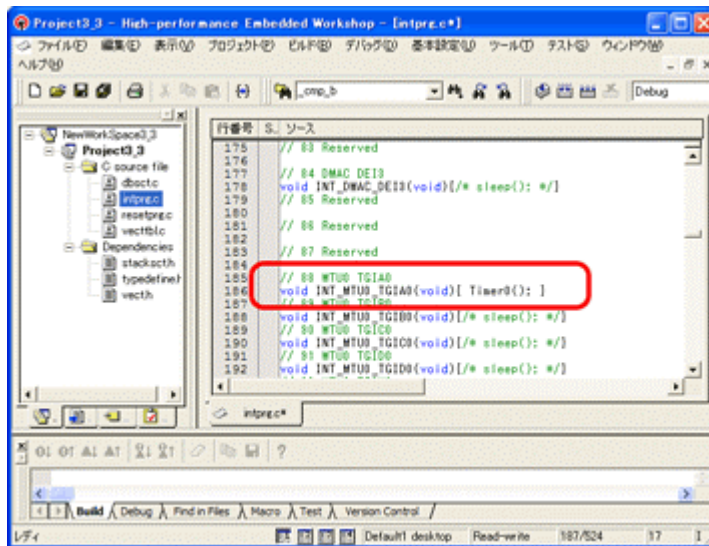
なお、トレース/ステップはUBCを利用したソフト判定での実施となります。

【方法①】 「set_imask(14)」を記述する。

【方法②】 DEF操作でのショートPB「DI」をクリックする。

3) <intprg.c>の変更

[6-3]

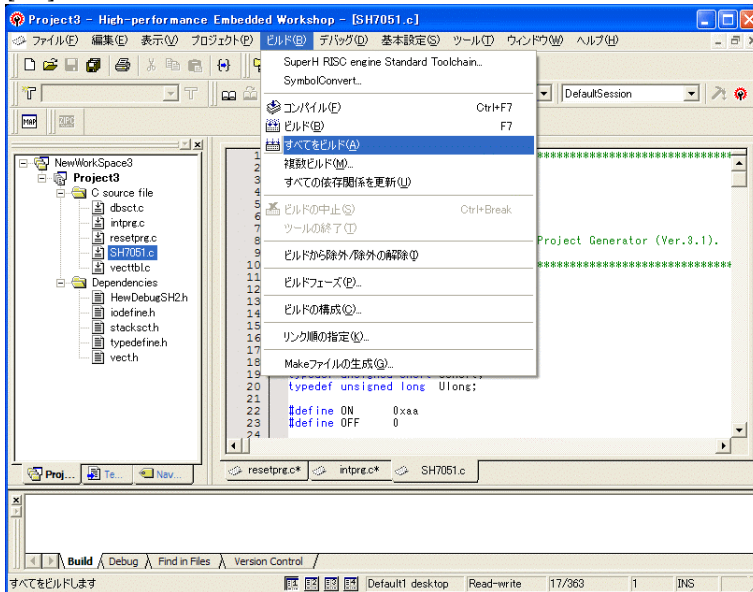


今回使用の「SH7144.C」は、MTU0のTGIA0割り込みを使用した例ですのでベクターを設定します。

- ① 「vector 88」に「Timer0();」関数を登録します。

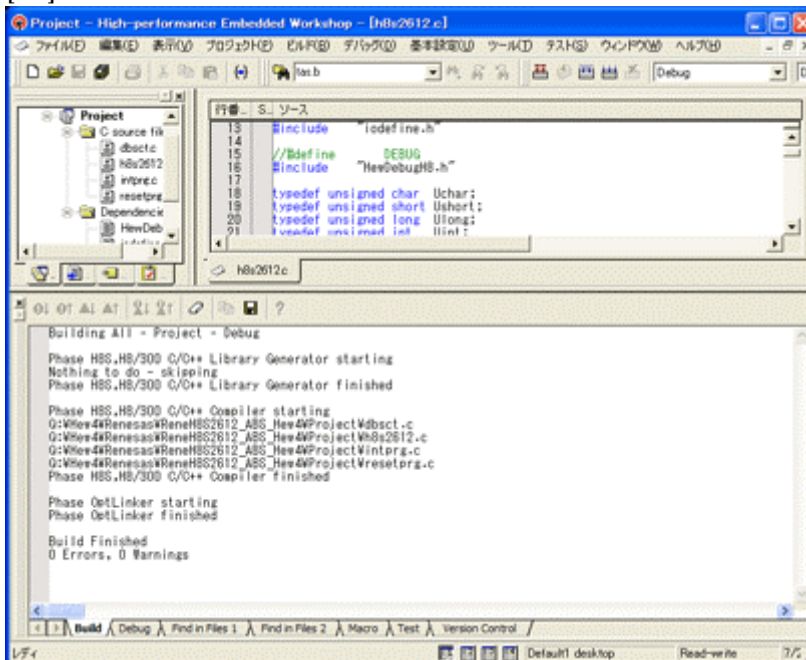
7. ビルドの実行

[7-1]



[ビルド] -
[すべてをビルド]をクリック
します。

[7-2]

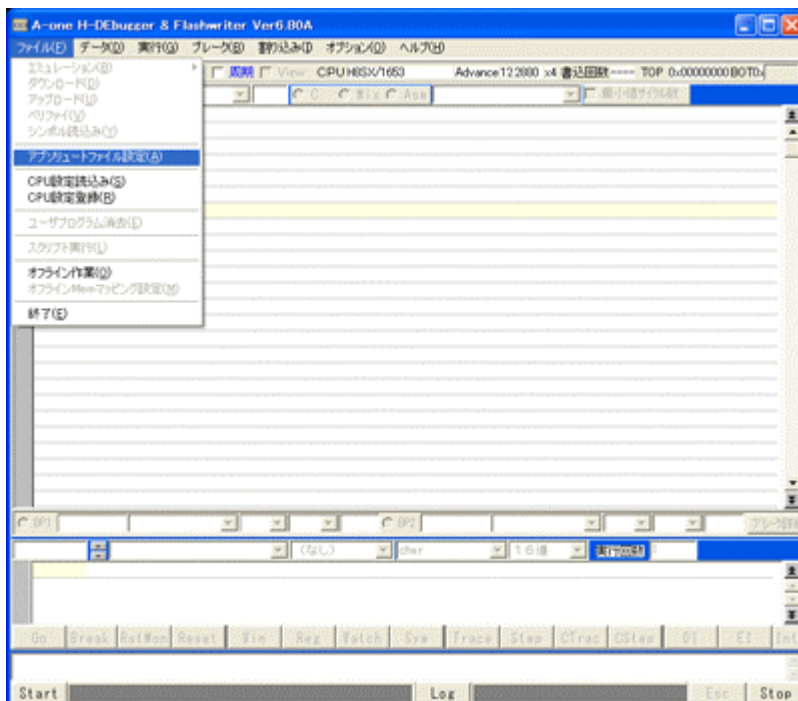


「0 Error 0 Warnings」
になり作業終了です。

8. DEFでの確認

1) アブソリュートファイル指定でのダウンロードを指定する。

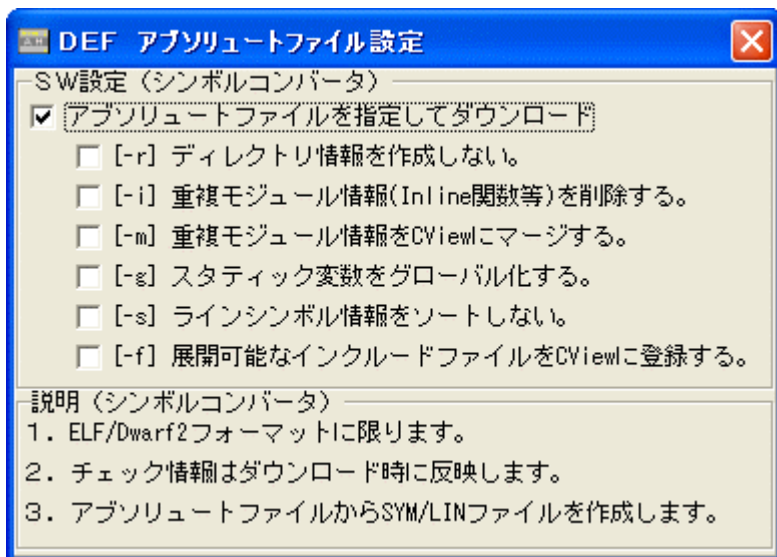
[8-1]



<ファイル>—
<アブソリュート設定>
を指定します。

2)

[8-2]



左画面のように
「チェック 」を入れて
下さい。

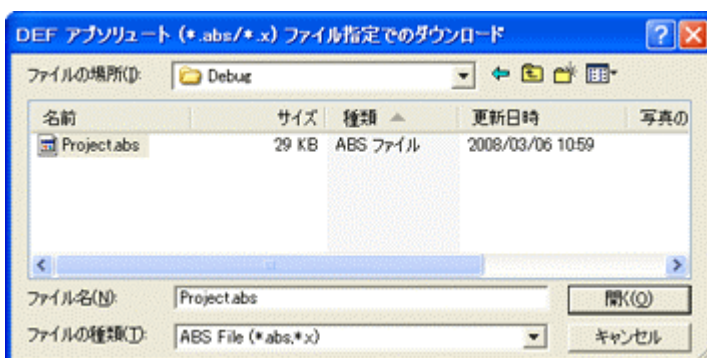
Hewデフォルト設定の状態
で使用する場合は、
[r]を**チェックしない**で下
さい。

アブソリュートファイルから
直接
「*SYM*.LIN」ファイル
を作成します・

この設定は記憶します。

3) ダウンロードします。

[8-3]



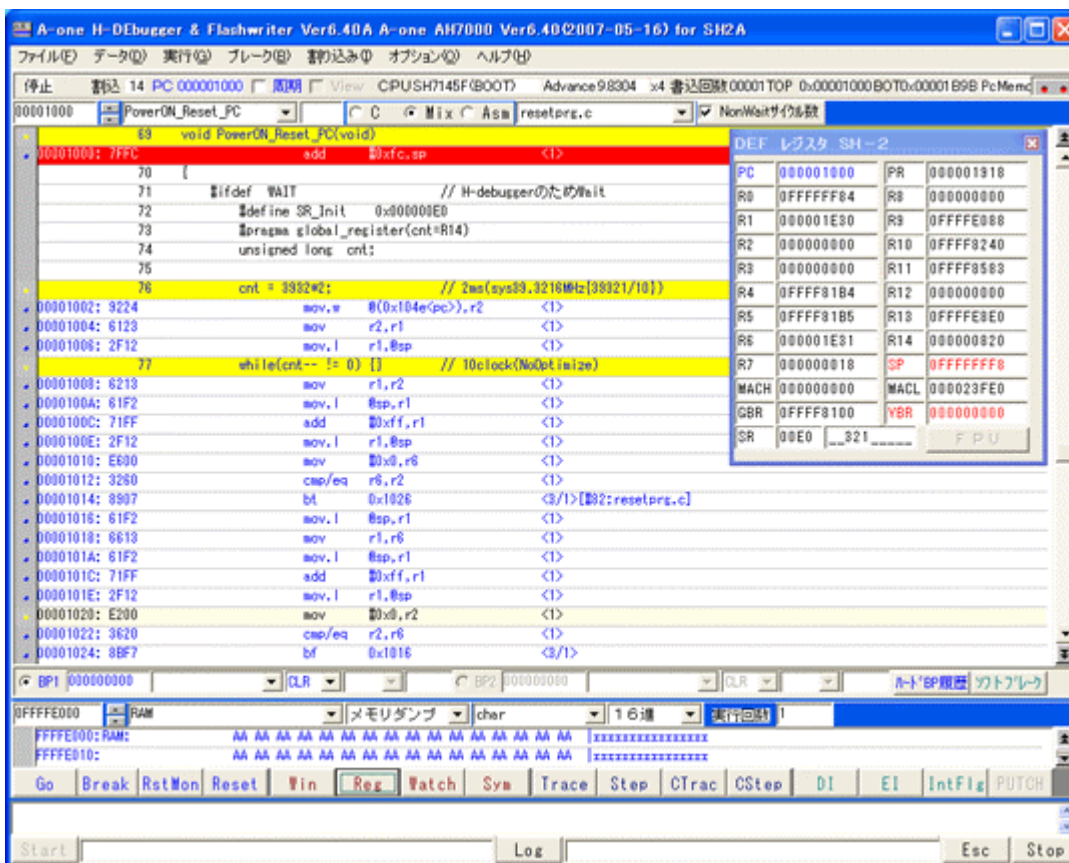
<ファイル>—
<ダウンロード>
を指定します。

左画面の通り、ファイル指定後
「開く」をクリックしますとダウ
ンロードを開始します。

<Debug>ホルダーがデフォルト
位置になります。

4) <resetprg.c>の確認

[8-1]



- ①0番地のリセットベクター値「0x1000」が確認できます。
- ②ソフトタイマの逆アセンブラ表示です。(参考まで)
- ③SR値およびSP値がリセット状態であることが確認できます。

これで「H-Debugger」用の設定作業が終了です。

以上