

統合環境Hew (Ver 3.01) 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 (H8/Tiny版)

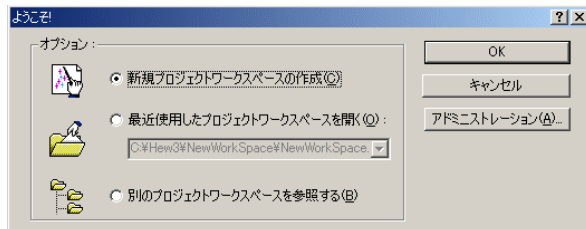
ルネサスC言語用統合環境「Hew Ver 3.01 (release1)」で H-debugger 用に新ワークスペース／プロジェクトを登録する手順方法を説明します。

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	NewWorkspace3		
プロジェクト名	Project3		
登録モジュール名	H83664.c	Cファイル	メインモジュール (アプリ用)
Hew 添付ファイル	Resetprg.c Intprg.c Dbsect.c	Cファイル	スタートアップモジュール 割込みベクターモジュール 定数転送用セクション管理宣言
CPUタイプ	H8/3664F		

1. 新ワークスペースの登録方法

“HEW” 起動させます。



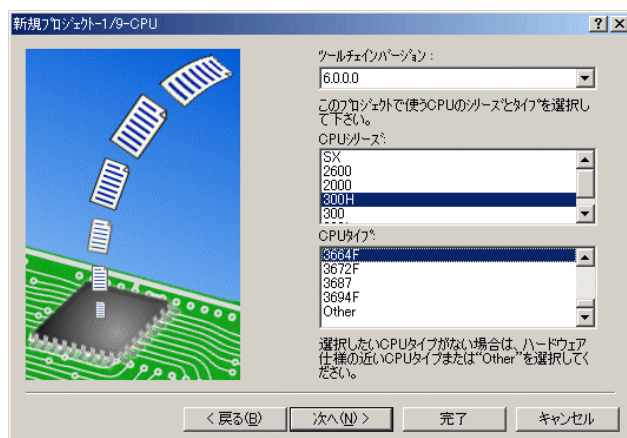
“新規プロジェクトワークスペース”をチェックしてのOKをクリックする。

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。



ワークスペース名	“NewWorkspace3”
プロジェクト名	“Project3”
ディレクトリ	“C:\Hew3\NewWorkspace3”
CPU 種別	“H8S, H8/300”
ツールチェーン	“Hitachi H8S, H8/300 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。



CPU シリーズを“300H”に選択する。
CPU タイプを“3664F”に選択する。

確認後、**次へ>**をクリックします。



CPUスペックを確認後、**次へ>**をクリックして下さい。



ここでのHewが用意したROM化支援関数は、使用しませんので全てのチェックを外して下さい。

main() 関数生成を“None”に選択する。

確認後、**次へ>**クリックして下さい。



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。

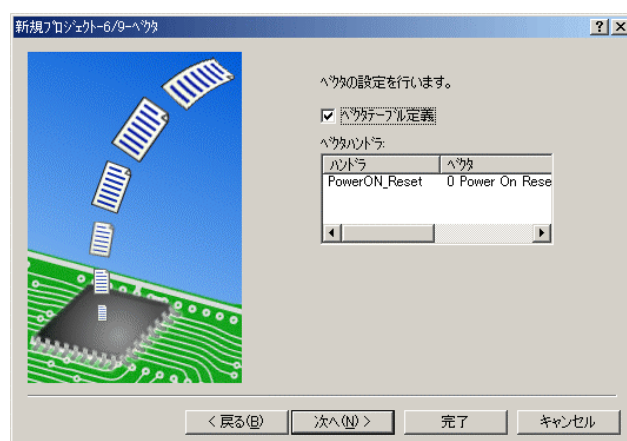
[次へ>](#)をクリックして下さい。



スタックボトムの設定です。
RAM位置を示す為「**HFF7E**」にする。

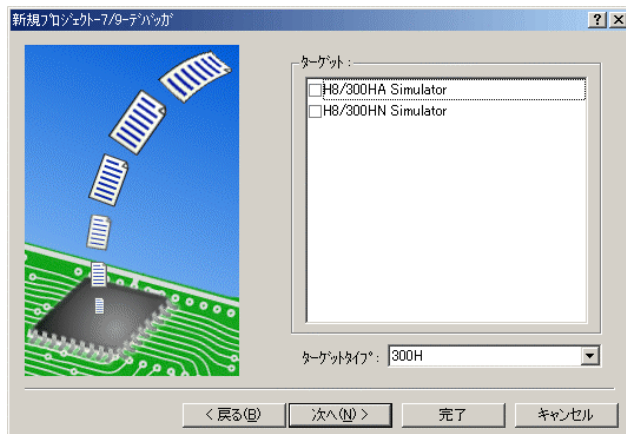
スタックサイズはデフォルト値にします。

[次へ>](#)をクリックして下さい。



ここで明示されたHew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。



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、

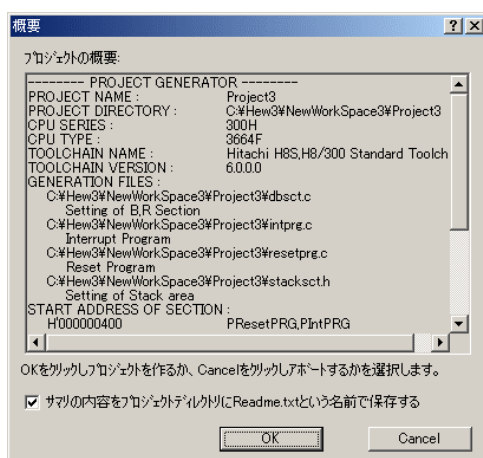
次へ>をクリックして下さい。



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。



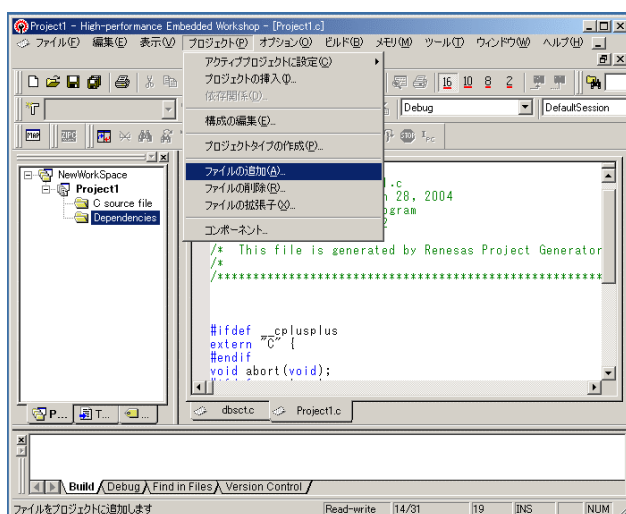
確認画面が表示されますので、**OK**をクリックして下さい。

ここまでの操作が新規プロジェクトの登録方法です。

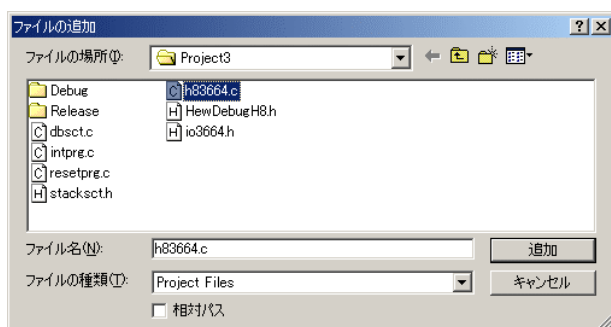
2 . プロジェクトに希望モジュール（ソースファイル）を登録する方法

準備： 作成済みのソースファイルを”C:\¥Hew3¥NewWorkSpace3¥Project3”にコピーして下さい。

H83664.c ” ¥sample¥HewH8T2_0¥H8_3664¥”
Io3664.h (左記ソースは、製品 CD の上記フォルダに入っています。)
(iodefine.h) 又は、Hew提供の I/O 定義ファイルを使用しても構いません。
HewDebugH8.h



[プロジェクト]-[ファイルの追加]をクリックします。



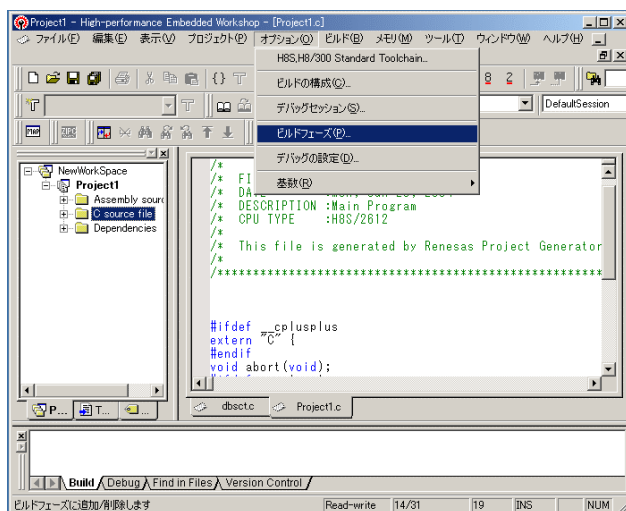
下記 1 ファイルを指定して下さい。

h83664.c

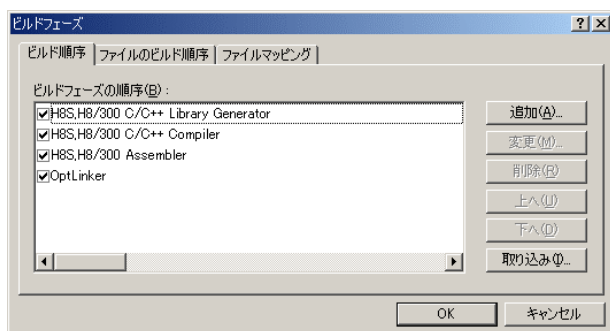
選択後、**追加**をクリックします。

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。

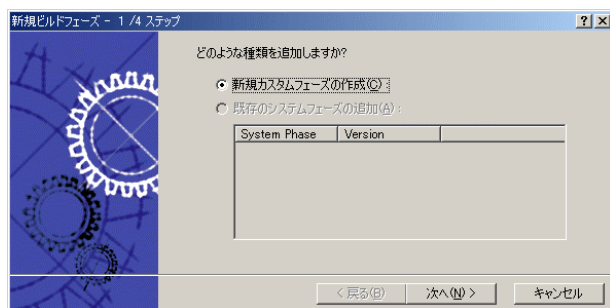
3. シンボルコンバータ「HC symconv」を登録する。



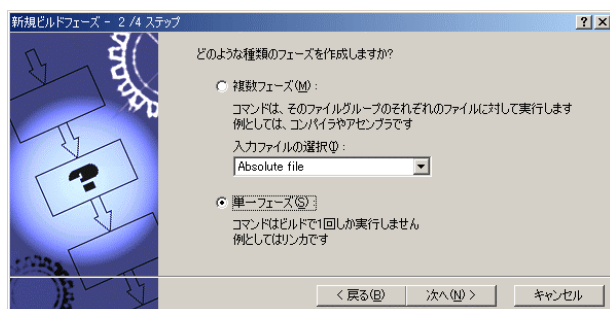
[オプション]-[ビルドフェーズ]をクリックします。



追加をクリックします。

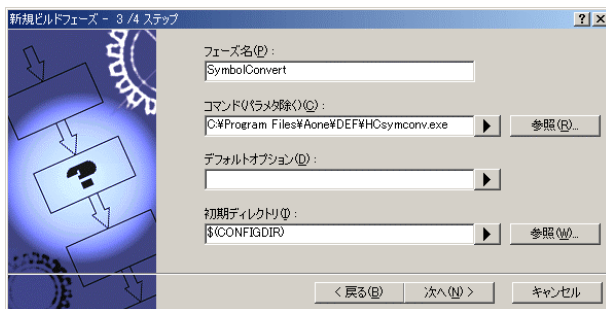


次へ>をクリックします。

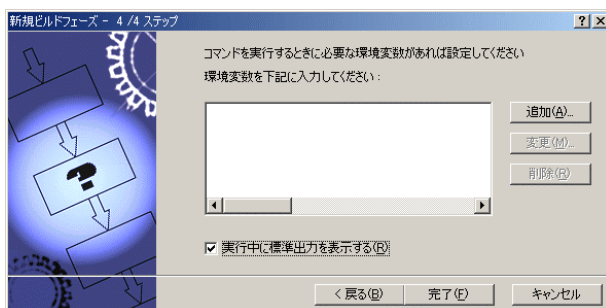


単一フェーズ側にチェックをします。

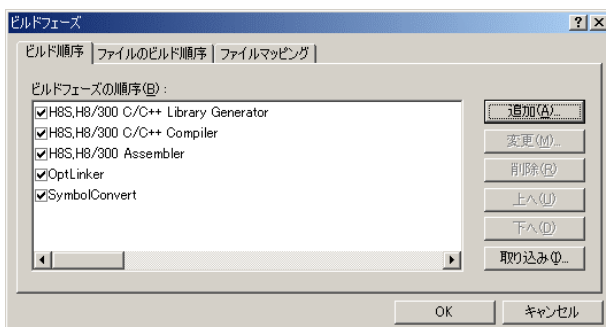
次へ>をクリックします。



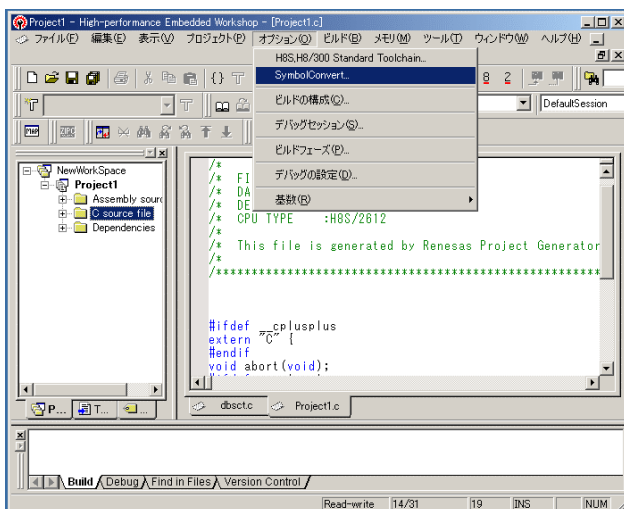
フェーズ：SymbolConvert
 コマンド：C:\ProgramFiles\Aone\DEF\HCSymconv.exe を選択する。
 (デフォルト)
 初期ディレクトリ：\$(CONFIGDIR)
 次へ>をクリックします。



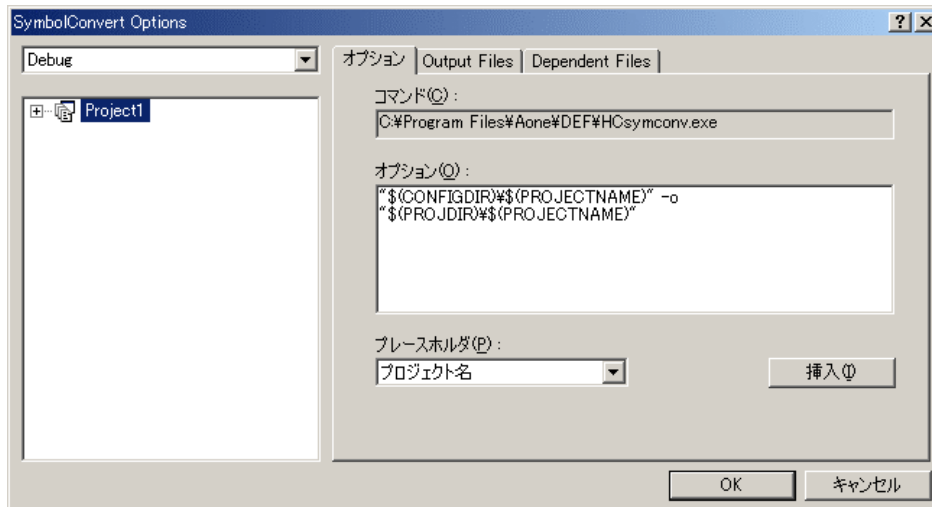
完了をクリックします。



OKをクリックします。



[オプション]-[SymbolConvert]をクリックします。



オプションに下記内容を設定する。

"\$(CONFIGDIR)¥\$(PROJECTNAME)" -o "\$(PROJDIR)¥\$(PROJECTNAME)"
 (入力ファイル名) (出力先名)

注意事項

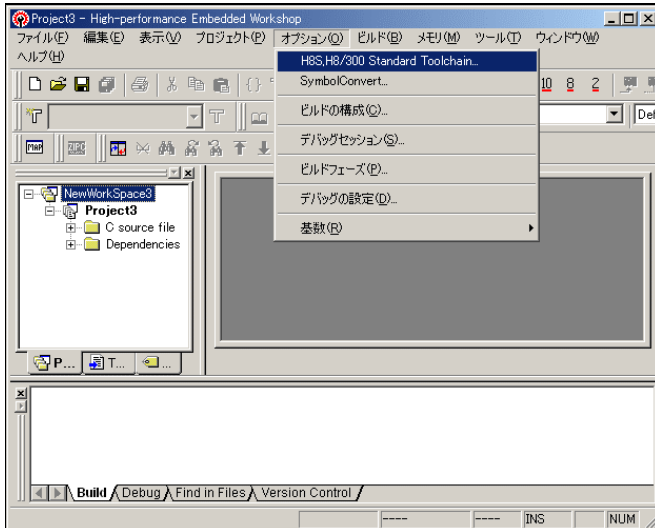
- 1) ディレクトリ名に ‘ ’ スペースを使用している場合は、“ダブルクォートで囲んで下さい。
 “\$(CONFIGDIR)¥\$(PROJECTNAME)” -o “\$(PROJDIR)¥\$(PROJECTNAME)”
- 2) \$(PROJECTNAME)の先頭に「¥」記号を挿入して下さい。(手入力)
- 3) オプションSW「-o」の両端には、スペースを入れてください。(手入力)
- 4) この設定例は、後説明の「*.mot」ファイルの生成されるディレクトリと同じ場所にシンボルコンバータが生成する「*.sym/*.lin」を置く為の指定です。
 <コンフィグレーションDIR>に生成させたい場合は、
 “\$(CONFIGDIR)¥\$(PROJECTNAME)”
 の指定のみで構いません。
 この場合は「*.mot」の生成場所を同じく<コンフィグレーションDIR>にして下さい。

追加事項 (HCSymconv.exe スイッチ説明)

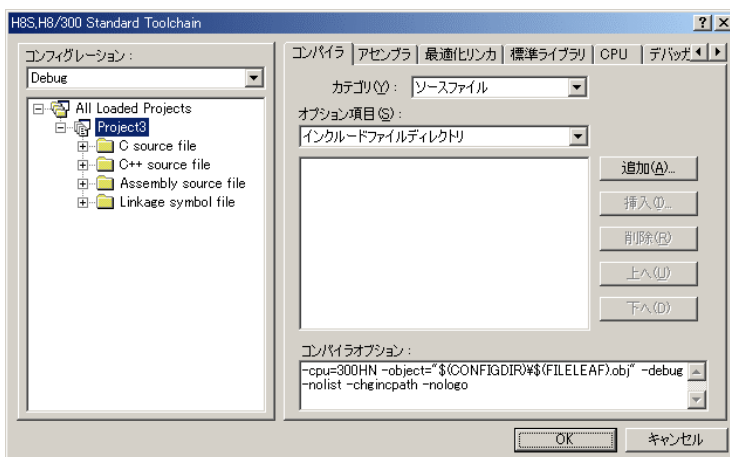
- 1) [-o] (省略可) 出力ファイル名を指定
- 2) [-r] (省略可) モジュール毎のディレクトリ情報を作成しない。ELF専用(Ver 3. 2xxから)
- 3) [-s] (省略可) ラインシンボル情報をソート (アドレス順) しない。(Ver 3. 2xxから)
- 4) [-i] (省略可) 重複モジュール情報を削除する。(Ver 3. 3xxから)
- 5) [-g] (省略可) スタティック変数をグローバル化する。(Ver 3. 6xxから)
- 6) [-m] (省略可) 重複モジュール情報をCソースにマージする。(Ver 3. 80Bから)
- 7) [-f] (省略可) 使用インクルードファイルをCViewに登録する。(Ver 3. 80Bから)

4 . ツール(ライブラリ)の設定

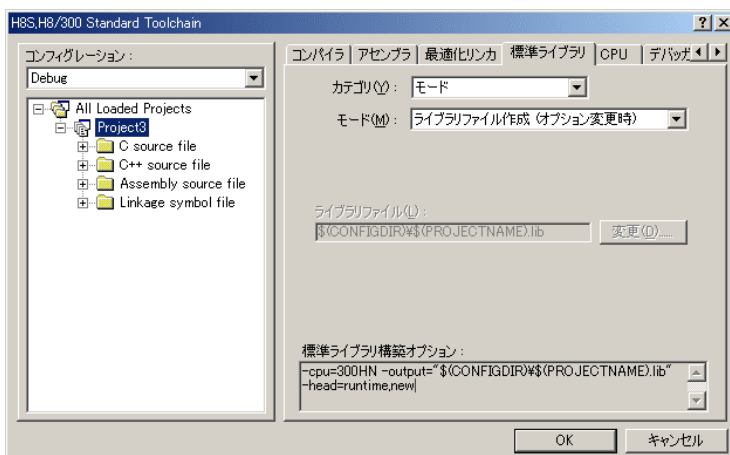
HEWは、プロジェクトごとにC言語用ライブラリを作成する仕様になっています。
ライブラリを作成および設定の確認をします。



[オプション]-
[H8S,H8/300Standard Toolchain]を
クリックします。



[標準ライブラリ]タグをクリッ
クする。

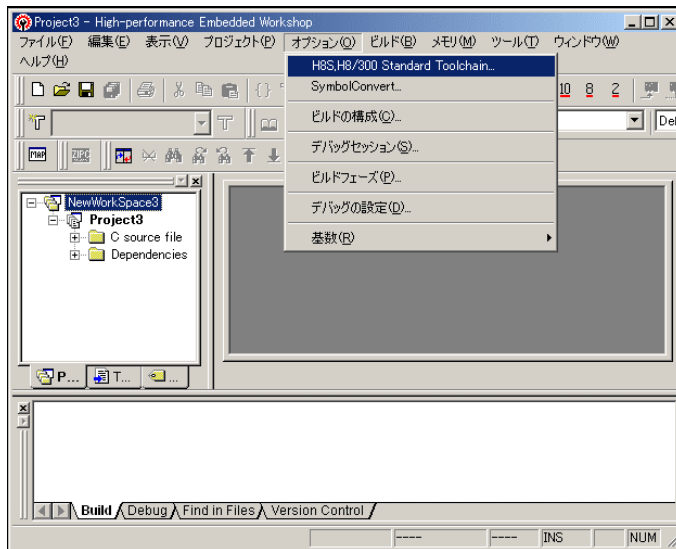


カテゴリのモードが
「ライブラリファイル作成 (オ
プション変更時)」指定になっ
ている事を確認します。

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。

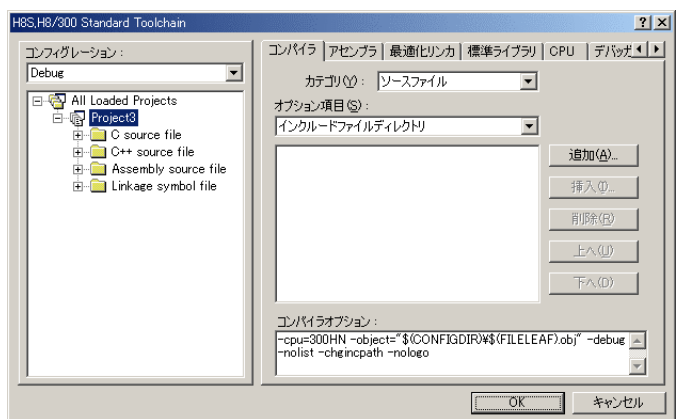
OKをクリックする。

5 . ツール（リンク）の設定

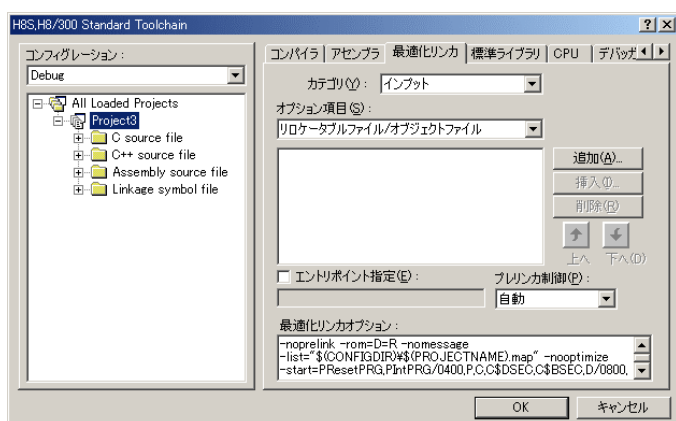


[オプション]-

[H8S, H8/300Standard Toolchain]を
クリックします。



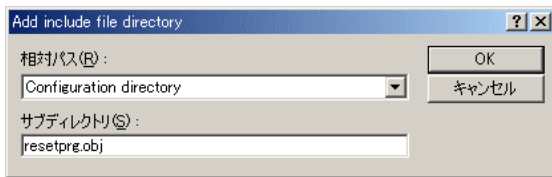
「最適化リンク」タグをクリックする。



カテゴリの「インプット」を選択する。

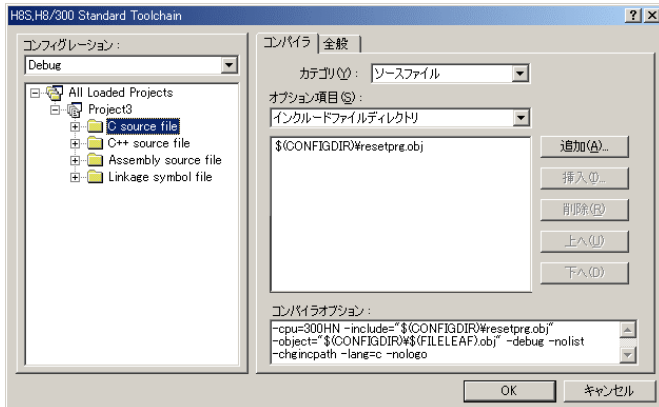
オプション項目の
「リロケータブルファイル／オブジ
ェクトファイル」を選択する。

追加をクリックする。



相対パス:「**Configuration directory**」に選択します。
サブディレクトリ:「**resetprg.obj**」と入力します。

OKをクリックします。

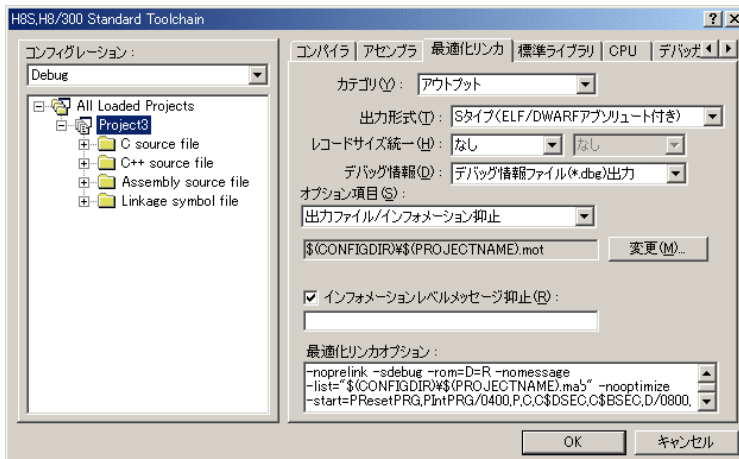


ファイルがセットされたのを確認します。

この指定は、“最初にこのリスト順にモジュールをリンクしなさい”との指示になります。

(DEF の C View にてアドレス順に他モジュールを表示させたい場合は、追加設定してください。)

(重要) 特に「**resetprg.obj**」の先頭に **SP** を設定するコードがありますので先頭を **0 x 8 0 0** 番地にする為の設定ですが、後説明のセクション指定で「**PRResetPRG**」の先頭番地を **0 x 8 0 0** 番地にしていますので必要ありませんが、同セクション名の他モジュールを追加した場合でも、必ず最初にリンクさせるための手続きです。



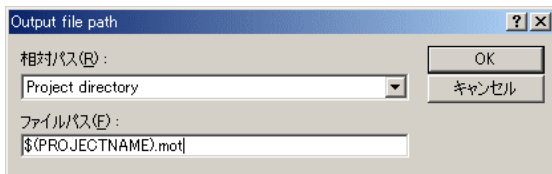
カテゴリ:
「アウトプット」を選択する。

出力形式:
「S タイプ (ELF/DWARF アブソリュート付き)」を選択する。

デバッグ情報:
「デバッグ情報ファイル (*.dbg) 出力」を選択する。

「出力ファイル／インフォメーション抑止」の**変更**をクリックします。

(重要) シンボリックデバッグを可能にするために必要な設定です。

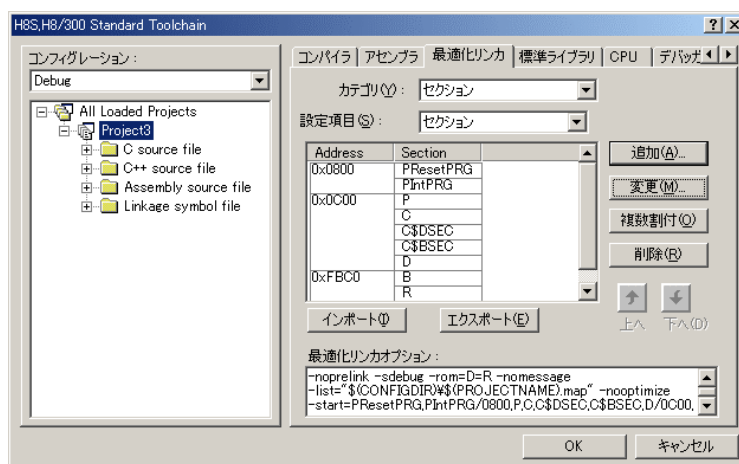


相対パス:を「Project directry」に設定します。

OKをクリックする。

(重要) この指定は、HEXファイルをCソースファイルのある同じディレクトリに置くための設定です。絶対条件として、「*.mot/*.sym/*.lin」は、同じ場所に置く必要があります。

HCsymconvで出力ファイルを「Configuration directory」にした場合は、上記の指定も「Configuration directory」にして下さい。今回の使用例は、「Project directry」になっています。



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。

0x00000800	PResetPRG
	PIntPRG
0x00000C00	P
	C
	C\$DSEC
	C\$BSEC
	D
0x0000F7C0	B
	R
0x0000FE7E	S

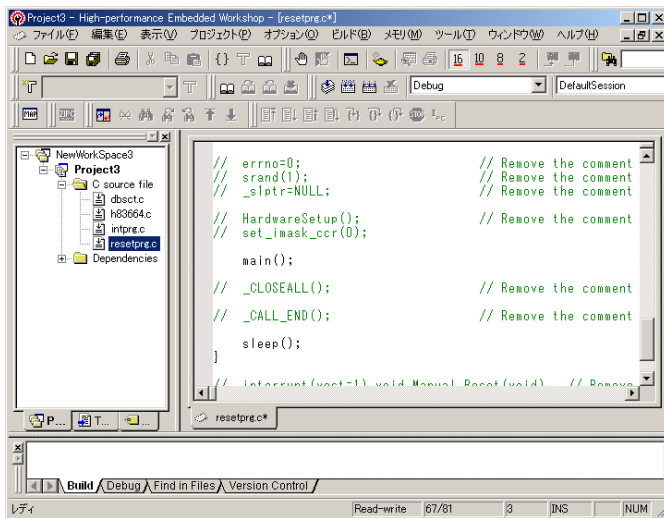
OKをクリックする。

(重要) 「Bセクション」0xF7C0 (0xF780) は、ソースブレイク使用時の例です。

DEFバージョン6.50xから、ソースブレイクを使用する場合は、モニタワーク方式をスタック方式に選択する必要があります。

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



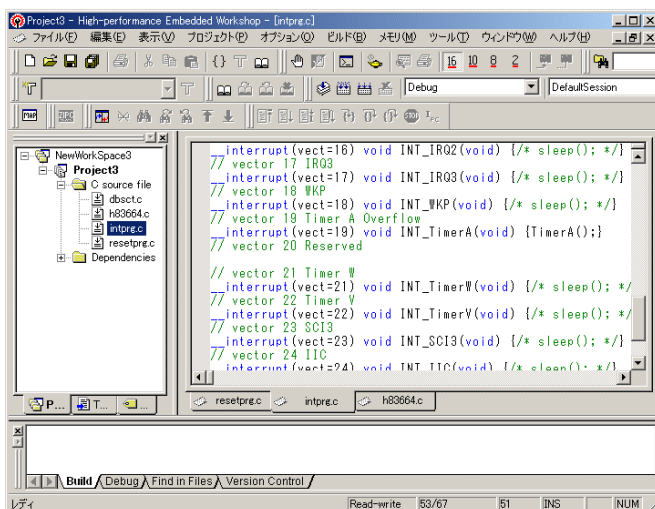
Main()の1行上の

①set_imask_ccr(0)をコメントにします。

(理由)

この例ですと main()関数内で I/O 初期化および割り込み許可をしていますので、ここでは割り込み禁止状態にしておきます。

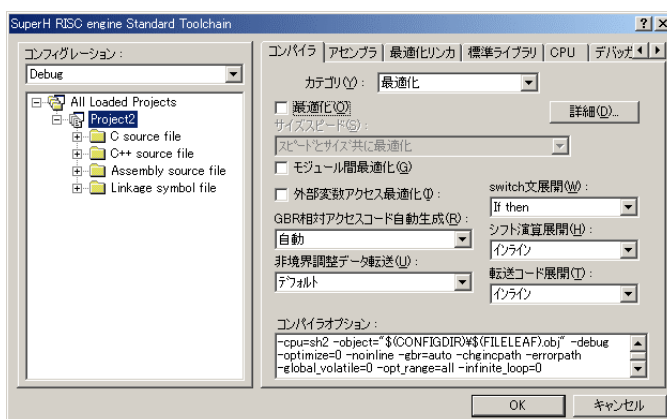
2) <intprg.c>の変更



今回使用の「H83664.c」は、Timer A の **Overflow** 割り込みを使用した例ですのでベクターを設定をします。

①「vector 19」に「TimerA0;」関数を登録します。

3) コンパイラの「最適化」を外す



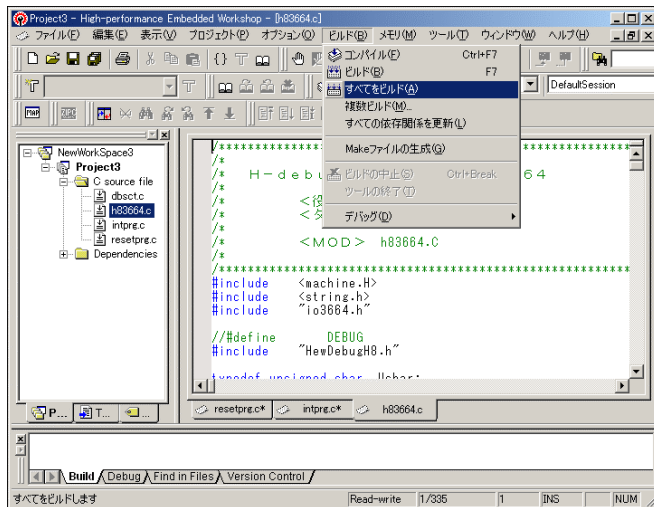
①Toolchain の「コンパイラ」を選択

②カテゴリ「最適化」を選択

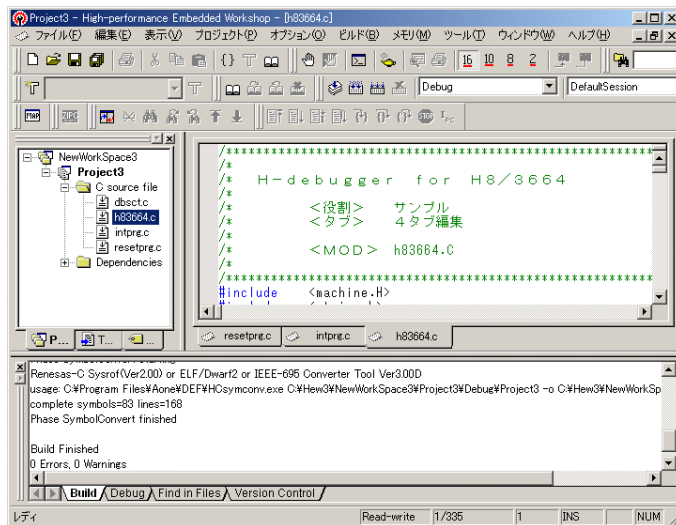
③「最適化」のチェックを外す。

④OKをクリックする。

7. ビルドの実行

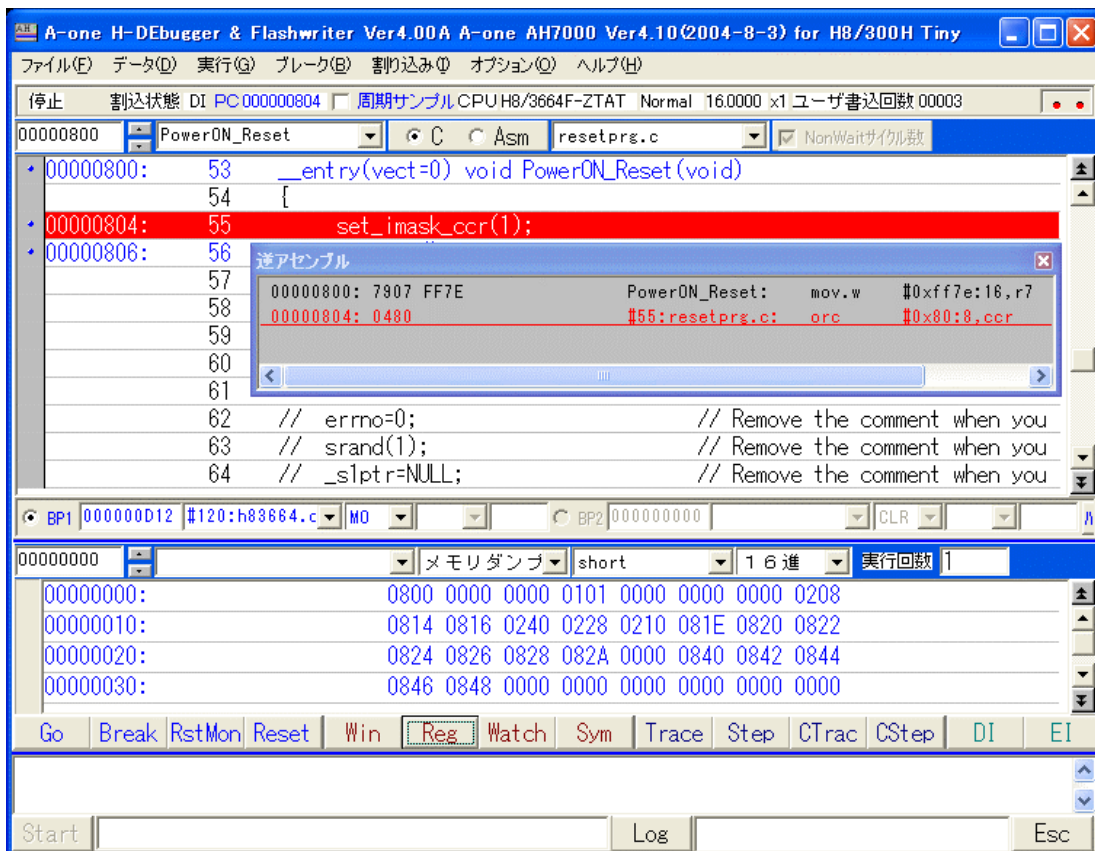


「ビルド」-「すべてをビルド」をクリックします。



「0 Error 0 Warnings」になり作業終了です。

8. DEFでの確認



上記画面の様に、

①800H番地にスタックポインタの設定プログラムが確認できます。

これで「H-Debugger」用に設定ができたのが確認できました。

以上