ＨＥＸファイル指定でダウンロードする場合

２００９年０４月２４日
Ｂｙ　エーワン　Ｒｅｖ１．２３

統合環境Ｈｅｗ（Ｖｅｒ４．０４）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（ＳＨ７０４７　ＢＯＯＴ版）

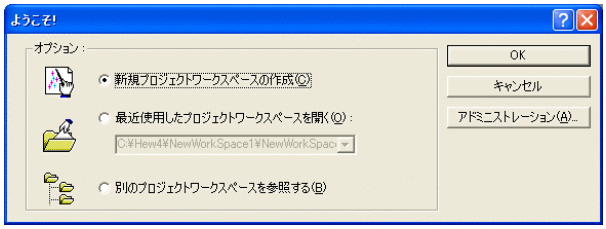
ルネサスＣ言語用統合環境「Ｈｅｗ　Ｖｅｒ４．０４」で H-debugger 用に新ワークスペース／プロジェクトを登録する手順方法を説明します。
説明を明確にするために、名前等を仮に決めて例に沿って説明を進めます。

ワークスペース名	ReneSH7047_BOOT_Hew4		
プロジェクト名	Project		
登録モジュール名	SH7047.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	SH7047F		

１．新ワークスペースの登録方法

“HEW”起動させます。

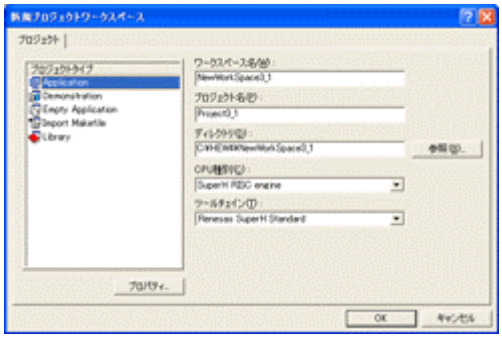
[1-1]



“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]



ﾜｰｸｽﾍﾟｰｽ名	“ReneSH7047_BOOT_Hew4
ﾌﾟﾛｼﾞｪｸﾄ名	“Project”
ﾃﾞｨﾚｸﾄﾘ	“C:¥Hew4¥Renesas”
CPU 種別	“SuperH RISC engine”
ﾂｰﾙﾁｪｲﾝ	“Renesas SuperH Standard”
ﾌﾟﾛｼﾞｪｸﾄ	Application

この項目を確認後、ＯＫをクリックして下さい。

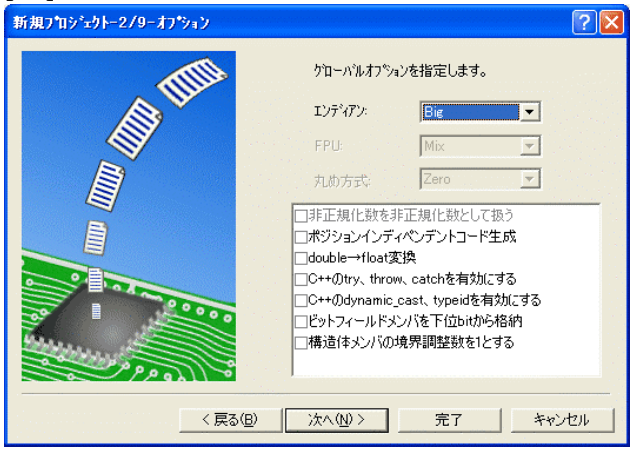
[1-3]

CPU ｼﾘｰｽﾞ を“SH-2”に選択する。
CPU ﾀｲﾌﾟ を“SH7047”に選択する。

確認後、

次へ>をクリックします。

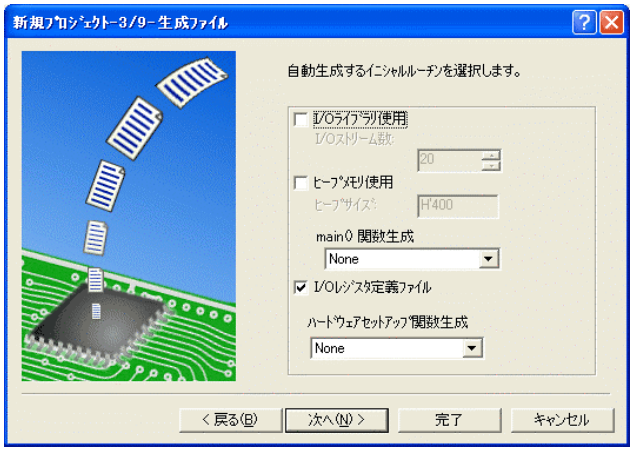
[1-4]



ＣＰＵスペックを確認後

次へ>をクリックして下さい。

[1-5]



①I/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。
③I/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。

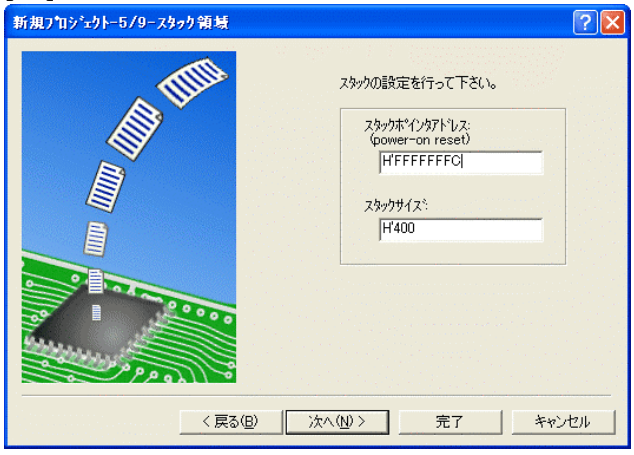
確認後、次へ>クリックして下さい。

[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>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、「H'FFFFFFFC」にする。

スタックサイズはデフォルト値でよい。

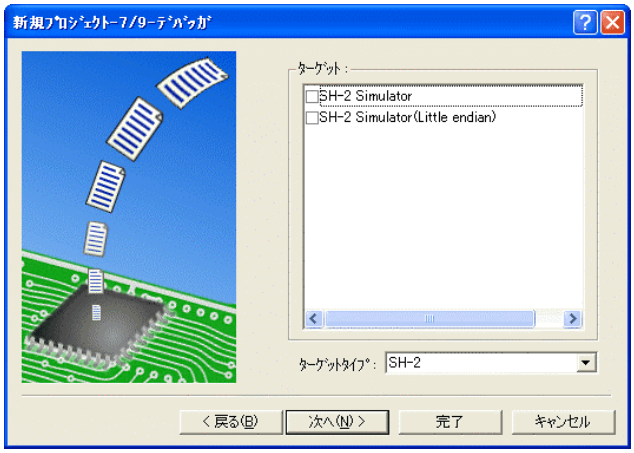
次へ>をクリックして下さい。

[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。

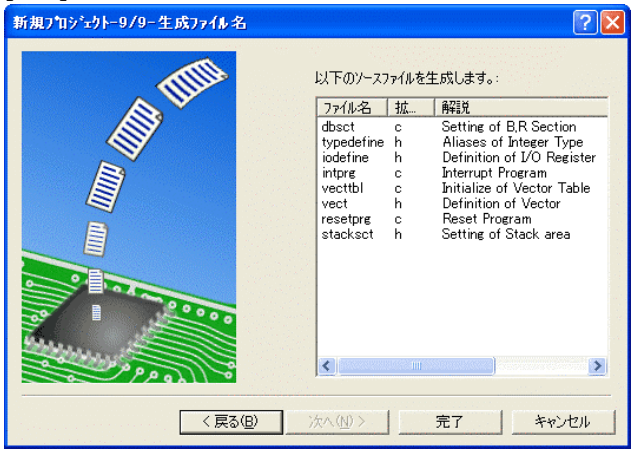
[1-9]



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>クリックして下さい。

[1-10]

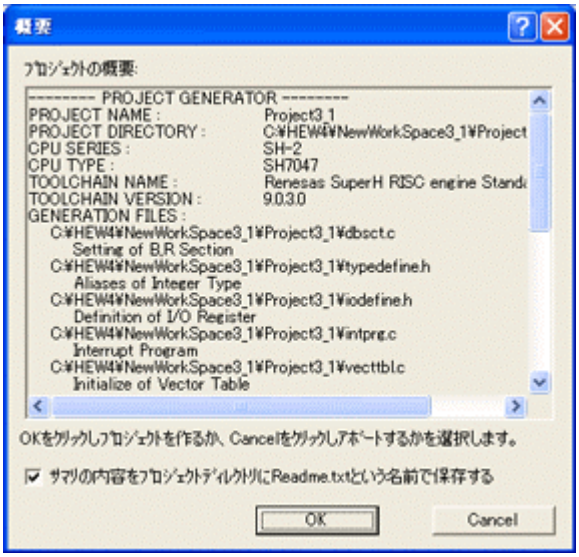


ここで最終になります。

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。

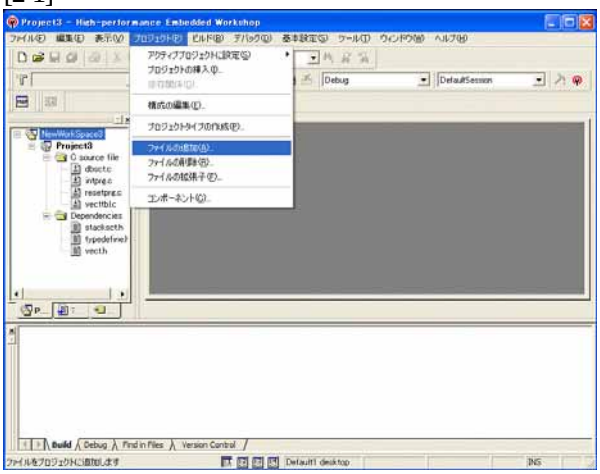
２．プロジェクトに希望モジュール（ソースファイル）を登録する方法

準備：　作成済みの２ファイルを”C:¥Hew4¥Renesas¥ReneSH7047_BOOT_Hew4¥Project”にコピーします。

SH7047.c　　HP よりダウンロードします。（ルネサスC）

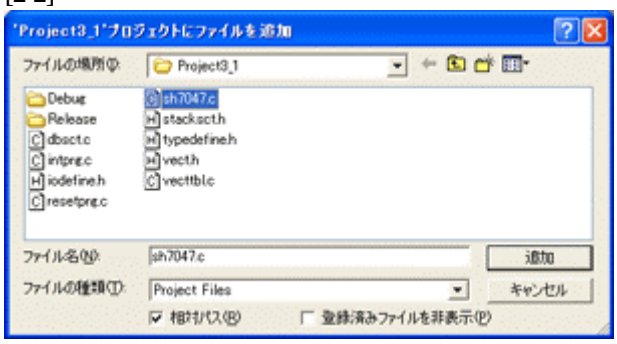
HewDebugSH2.h　　ReneSH7047_BOOT_Hew4.LZH

[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。

Sh7047.c

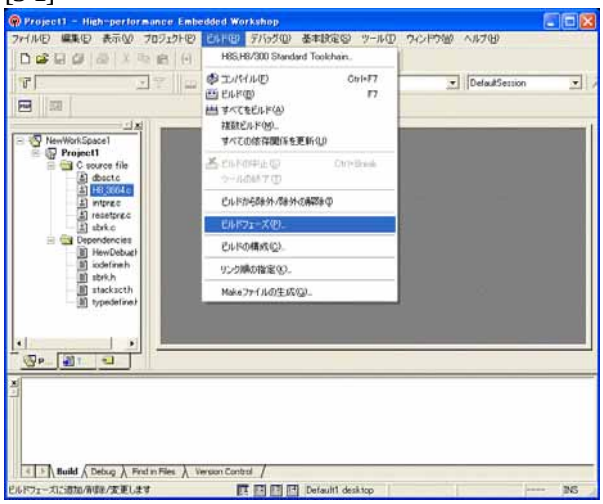
選択後、

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。

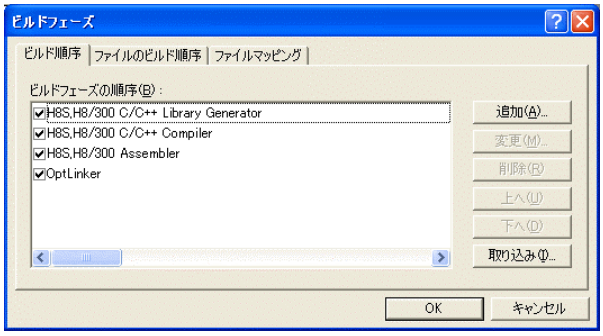
３．シンボルコンバータ「ＨＣｓｙｍｃｏｎｖ」を登録する。

[3-1]



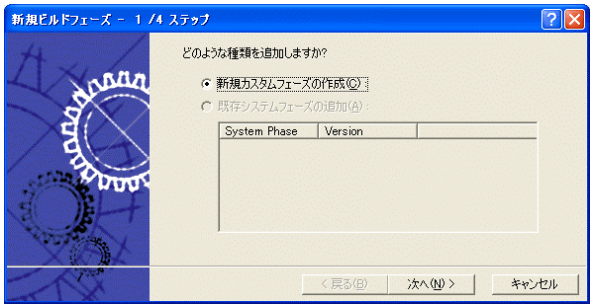
［ビルド］−
［ビルドフェーズ］をクリックします。

[3-2]



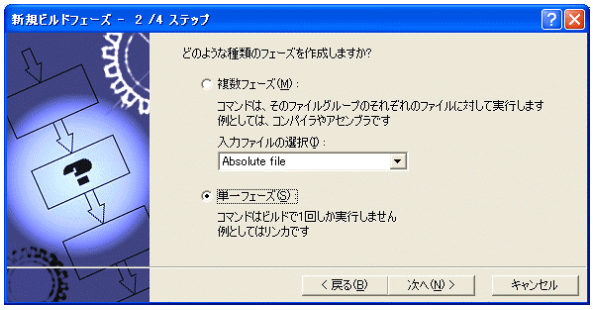
追加をクリックします。

[3-3]



次へ>をクリックします。

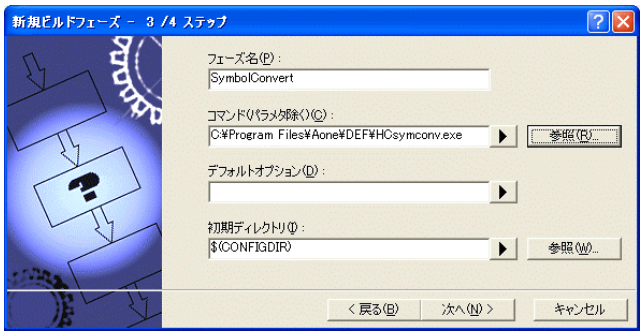
[3-4]



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。

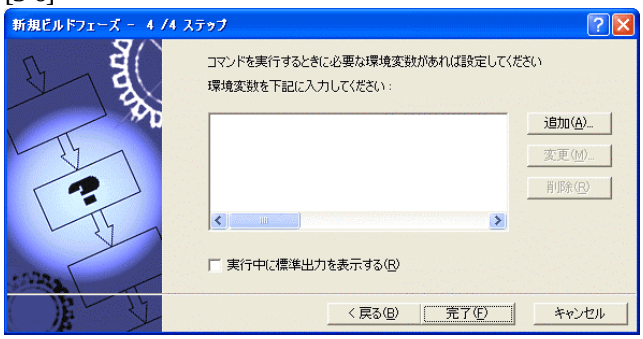
[3-5]



①フェーズ：SymbolConvert
②コマンド：
C:¥ProgramFiles¥Aone¥DEF¥
HCsymconv.exe を選択する。
③初期ディレクトリ：$(CONFIGDIR)

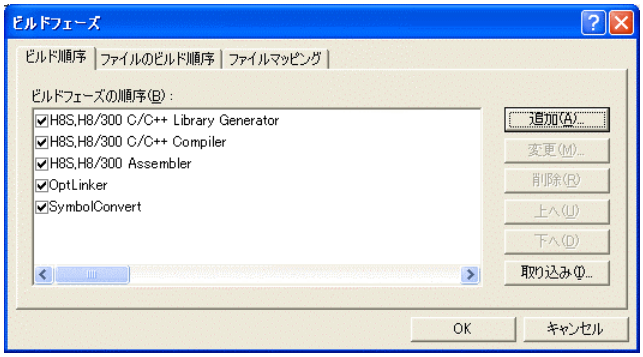
次へ>をクリックします。

[3-6]



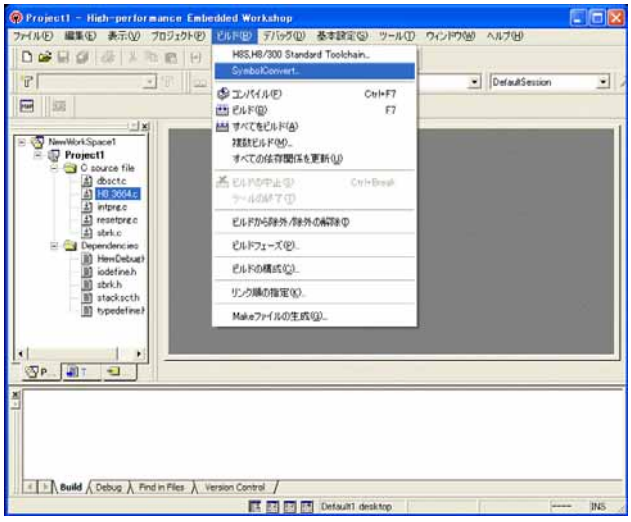
完了をクリックします。

[3-7]



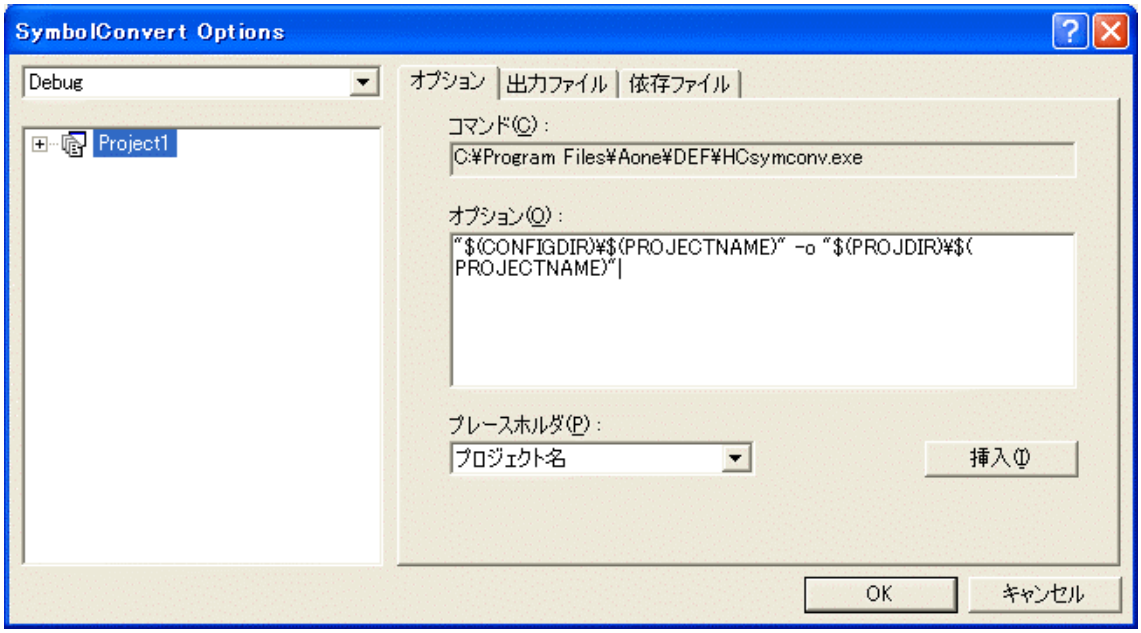
OKをクリックします。

[3-8]



[ビルド]-
[ＳｙｍｂｏｌＣｏｎｖｅｒｔ]をクリックします。

[3-9]



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。

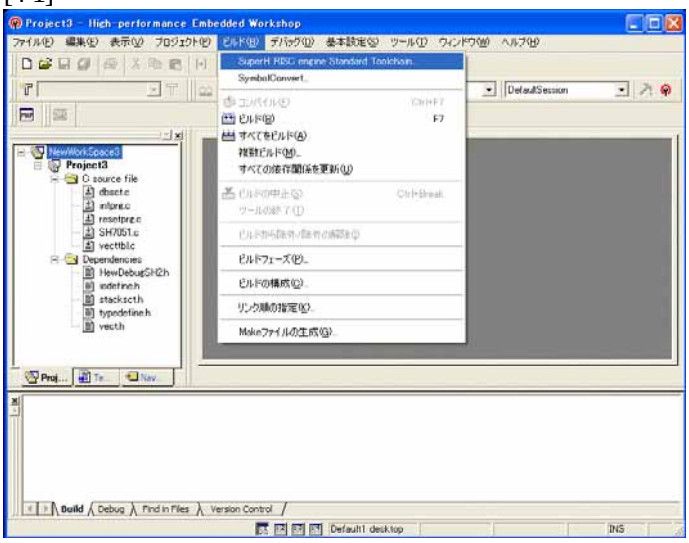
追加事項（ＨＣｓｙｍｃｏｎｖ．ｅｘｅ　スイッチ説明）

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(Ｖｅｒ３．２ｘｘ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。(Ｖｅｒ３．２ｘｘから）
４）［－ｉ］（省略可）重複モジュール情報を削除する。(Ｖｅｒ３．３ｘｘから）
５）［－ｇ］（省略可）スタティック変数をグローバル化する。(Ｖｅｒ３．６ｘｘ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。(Ｖｅｒ３．８０Ｂ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。(Ｖｅｒ３．８０Ｂから）

４．ツール（ライブラリ）の設定

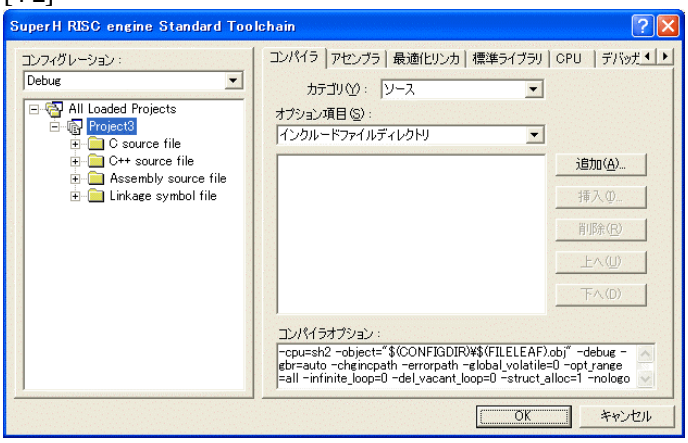
ＨＥＷは、プロジェクトごとにＣ言語用ライブラリを作成する仕様になっています。
ライブラリを作成および設定の確認をします。

[4-1]



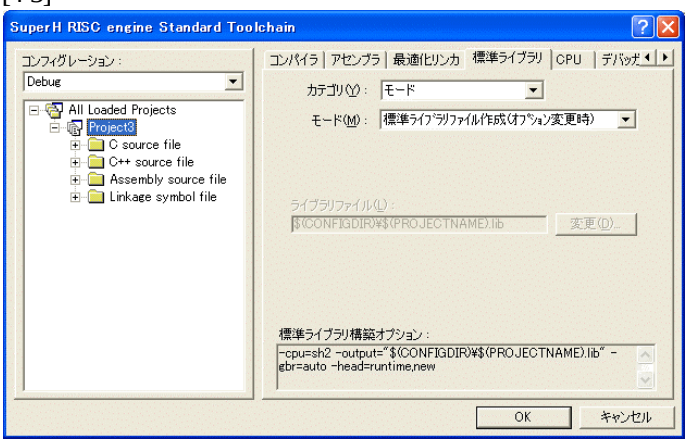
[ビルド]-
[SuperH RISC engine Standard Toolchain]をクリックします。

[4-2]



「標準ライブラリ」タグをクリックする。

[4-3]



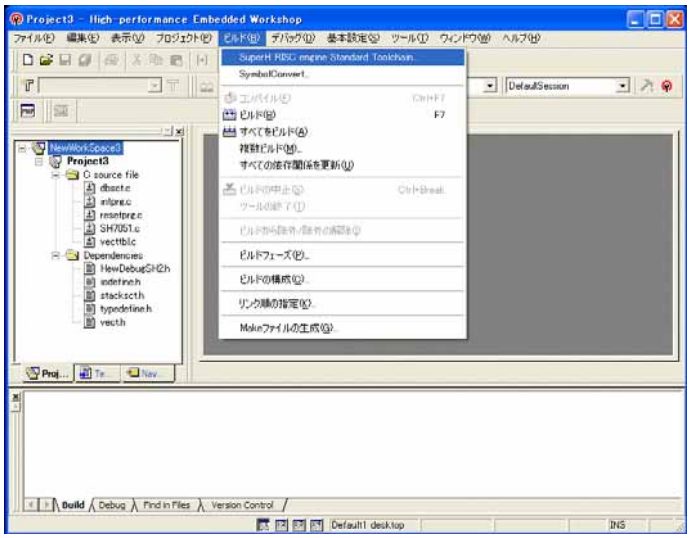
カテゴリのモードが
「ライブラリファイル作成（オプション変更時)」指定になっている事を確認します。

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。
デフォルトのままで

OKをクリックする。

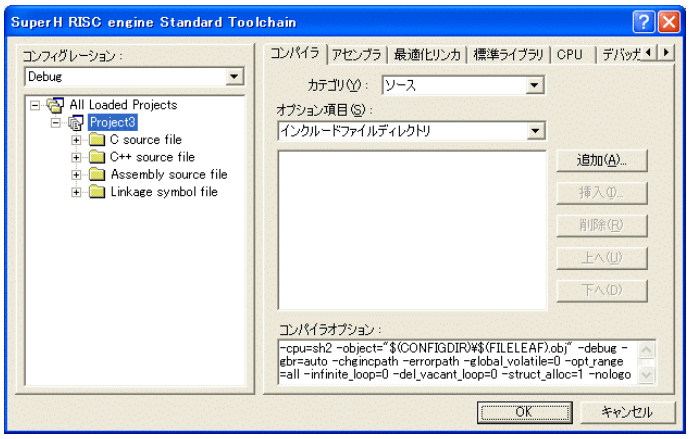
５．ツール（リンカ）の設定

[5-1]



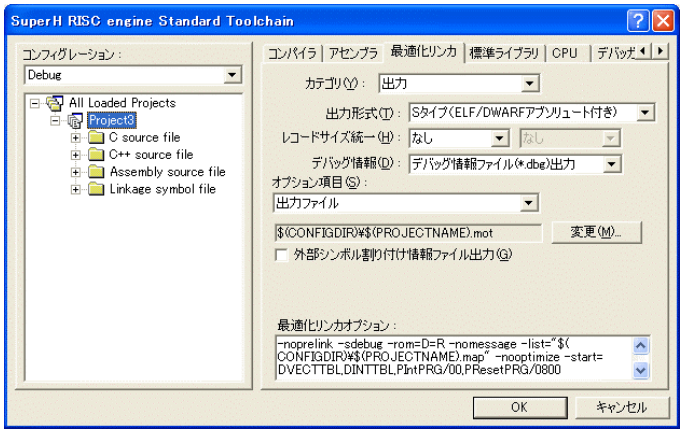
[ビルド]-
[SuperH RISC engine Standard Toolchain]をクリックします。

[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)」を選択する。
③デバッグ情報の
「デバッグ情報ファイル(*.dbg)出力」を選択する。
④「出力ファイル」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。

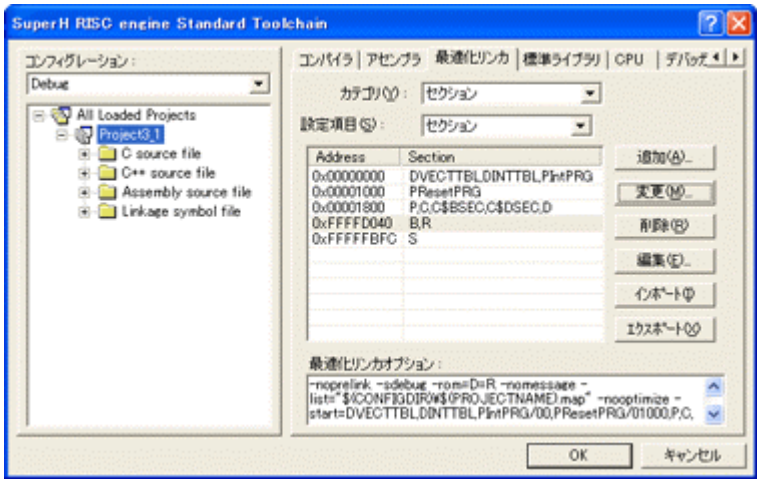
[5-4]

相対パス「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。

[5-5]



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。

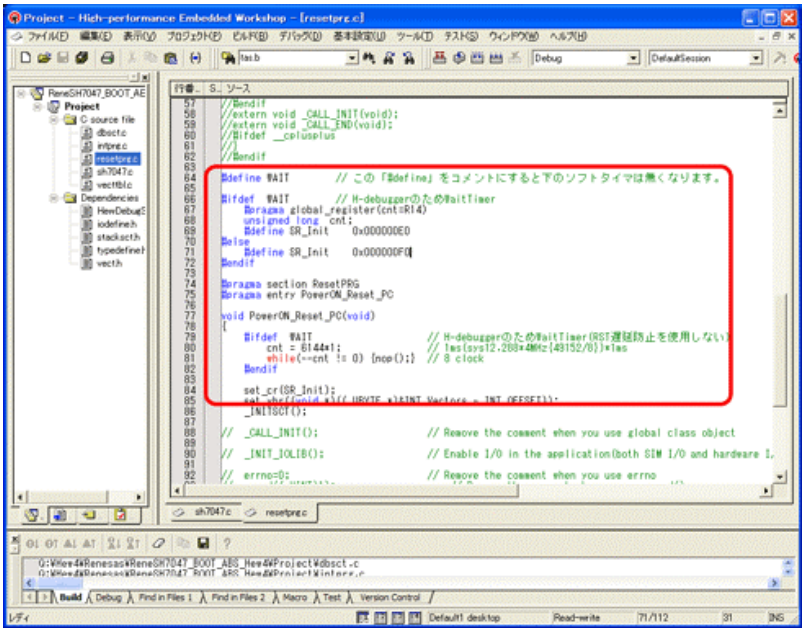
0x00000000	DVECTTBL, DINTTBL, PIntPRG
0x00001000	PResetPRG
0x00001800	P, C, C$BSEC, C$DSEC, D
0xFFFFD040	B, R
0xFFFFFBFC	S

(重要)　「Ｂセクション」０ｘＦＦＦＤ０４０（内臓ＲＡＭ先頭　０ｘＦＦＦＤ０００）は、ソースブレーク使用する場合の例です。
ＤＥＦバージョン６．５０ｘから、ソースブレー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」をコメントにして下さい。

④ソフトタイマによるメリット

・ＤＥＦの「Ｒｓｔ Ｍｏｎ」操作時「ｍａｉｎ ()」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。

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。

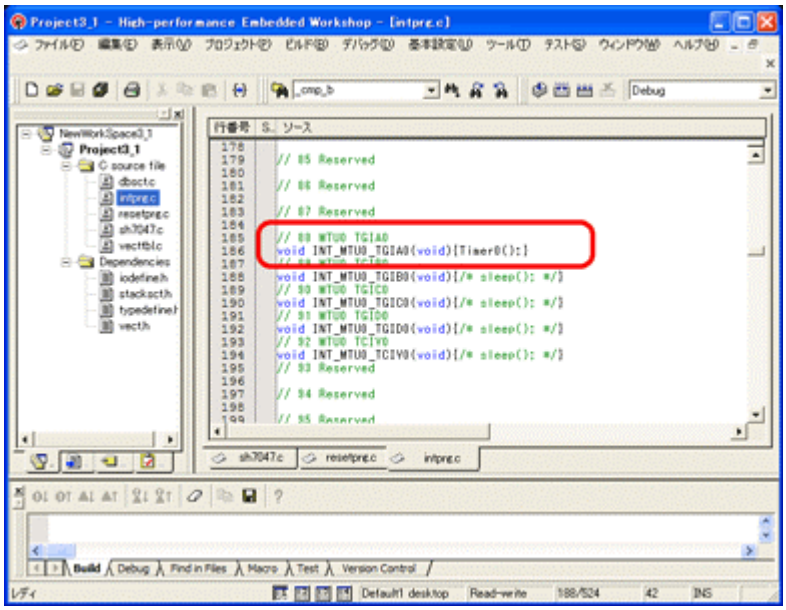
なお、トレース／ステップはＵＢＣを利用したソフト判定での実施となります。

【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]

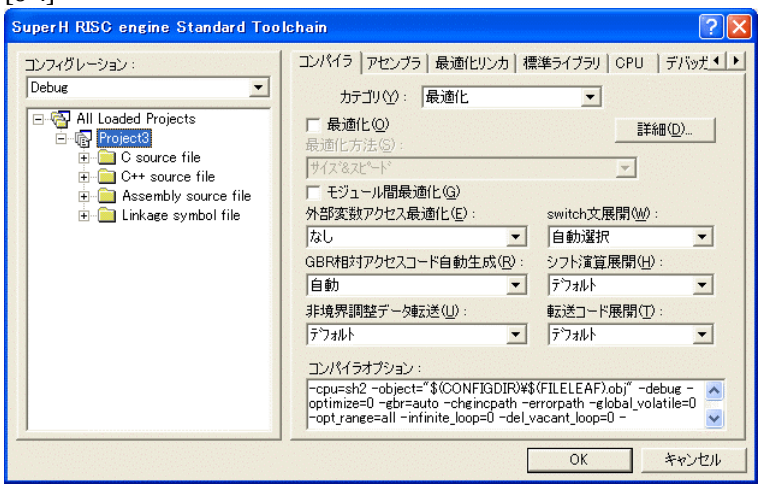


今回使用の「SH7047.C」は、MTU0 の TGIA0 割り込みを使用した例ですのでベクターを設定します。

①「vector 88」に「Timer0();」関数を登録します。

４）　コンパイラの「最適化」を外す

[6-4]



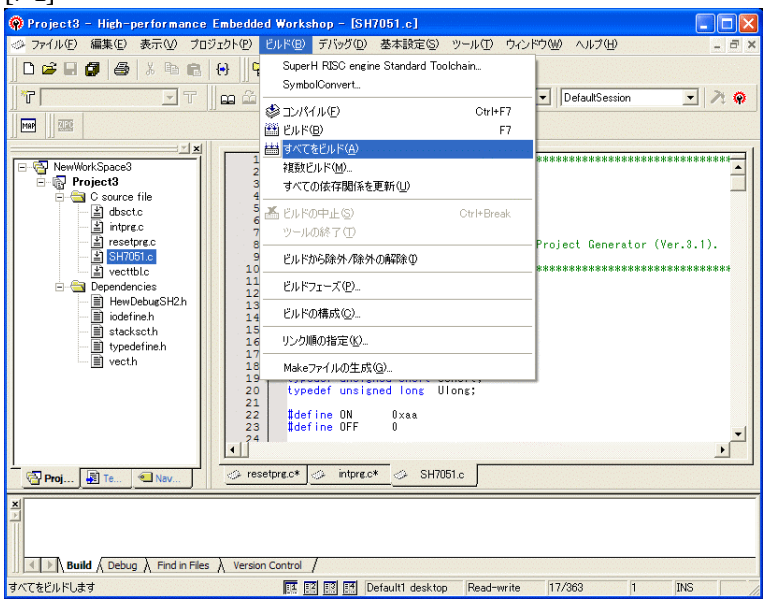
①[ビルド]-[SuperH RISC engine Standard Toolchain]をクリックします。

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。

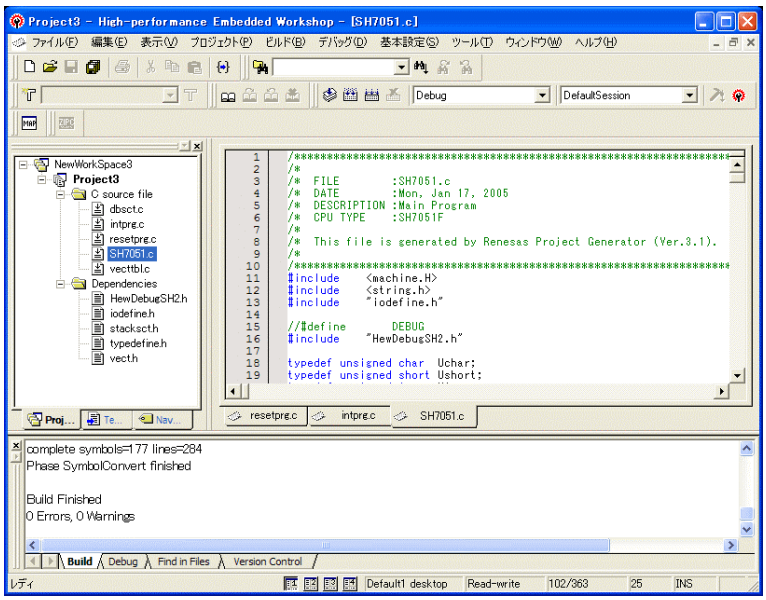
７．ビルドの実行

[7-1]



［ビルド］－
[すべてをビルド]をクリックします。

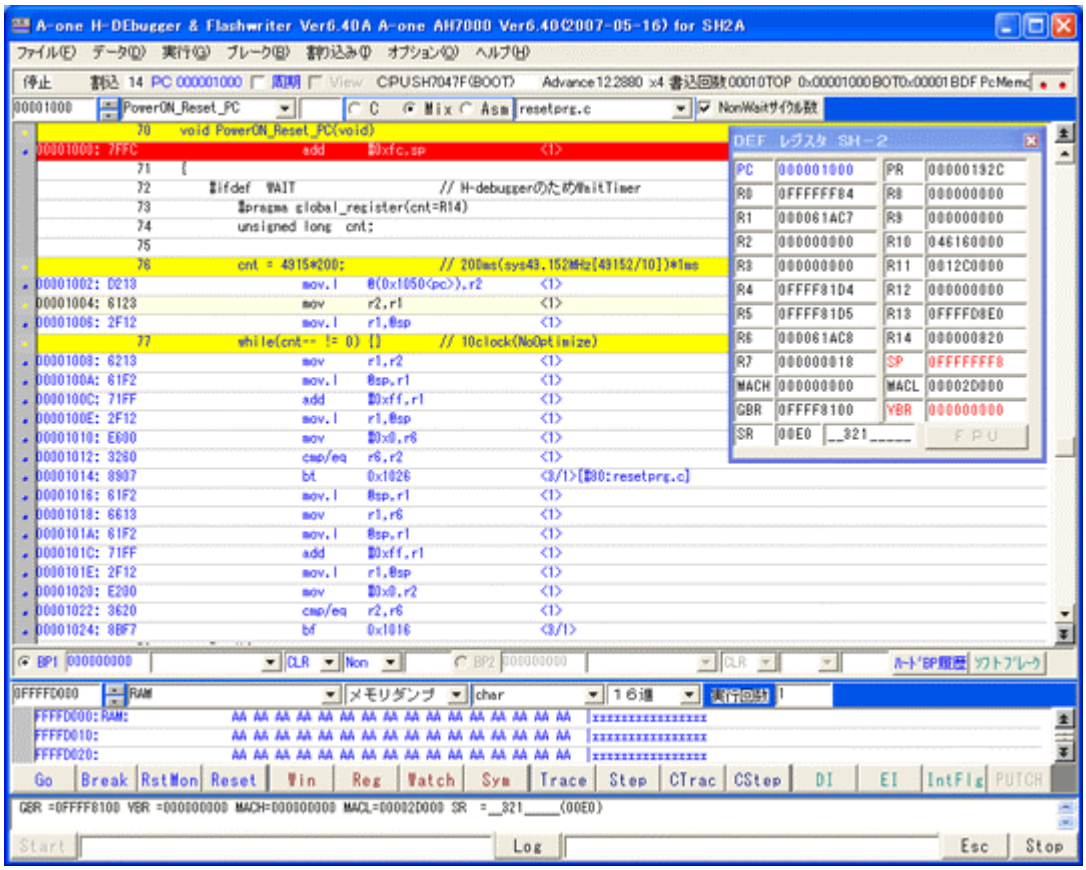
[7-2]



「0 Error　0 Warnings」になり作業終了です。

８．ＤＥＦでの確認

[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