

2008年03月10日  
エーワン Rev 1. 00

## Hew (Ver 4. 04) +KPIT (v0801) 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 (SH7051 BOOT版)

KPIT版の統合環境「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	KPIT7051_ABS		
プロジェクト名	Project		
登録モジュール名	SH7051.c	C	メインモジュール (アプリ用)
	KpitDebugSH2.h	ヘッダ	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。)
KPIT 添付ファイル	start.asm	ASM	スタートアップモジュール
	hwinit.c	C	ハード初期化用モジュール
	vects.c	C	リセット／割込みベクターテーブル
	inhandler.c	C	割込みハンドラー用
	iodef.h	ヘッダ	I/O 定義ビットフィールド記述用
	inhandler.h	ヘッダ	割込みハンドラー用
CPUタイプ	SH7051F		

### 【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「Object」項目を変更する。
- 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。

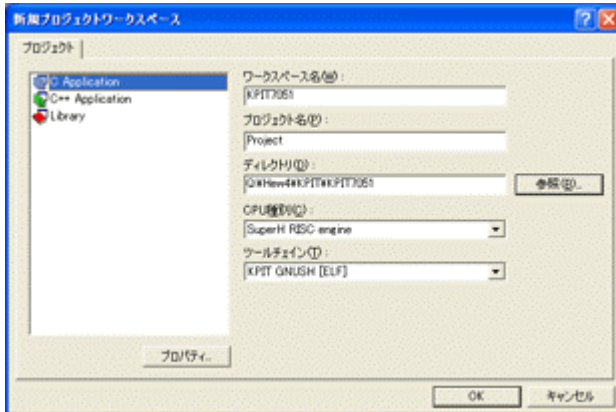
[1-1]



“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。

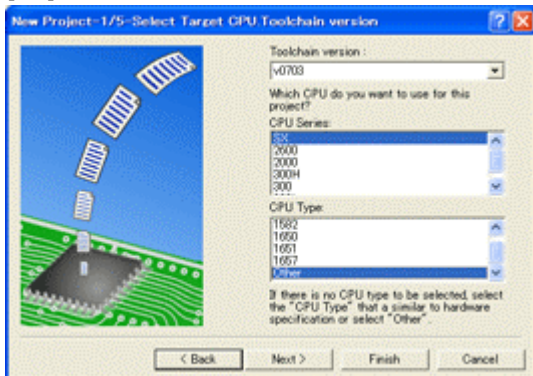
[1-2]



ワークスペース名 KPIT7051\_ABS  
プロジェクト名 Project  
ディレクトリ C:\Hew4\KPIT\KPIT7051  
CPU 種別 SuperH RISC engine  
ツールチェーン KPIT GNUSH [ELF]  
プロジェクト Application

この項目を設定確認後OKをクリックして下さい。

[1-3]

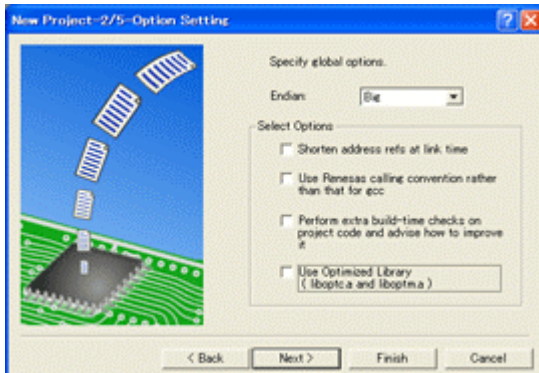


CPUスペックを選択します。

- ①SH2
- ②SH7051F

Next > をクリックします。

[1-4]

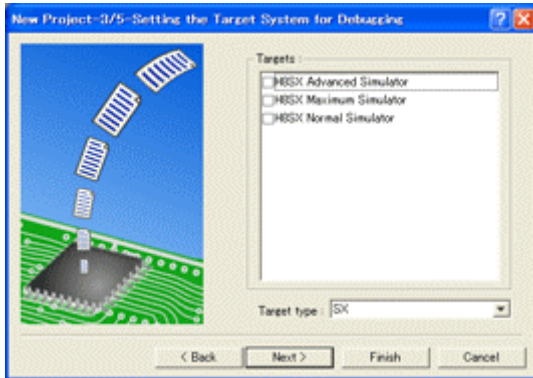


CPUオプションを選択します。

- ① 「Use Optimized Libraries」  
チェックを外します。

Next > をクリックします。

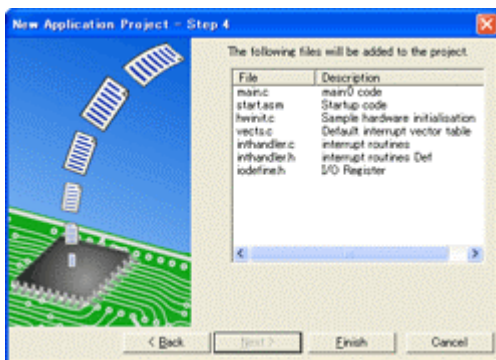
[1-5]



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、

**Next >** をクリックします。

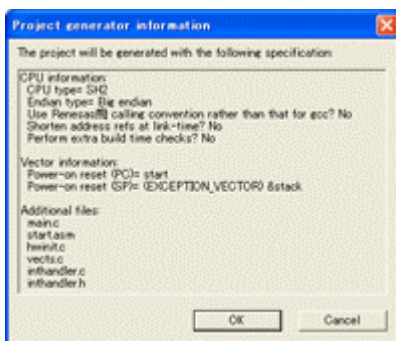
[1-6]



作成されるファイル一覧表示です。

**Finish >** をクリックします。

[1-7]



最終確認画面です。

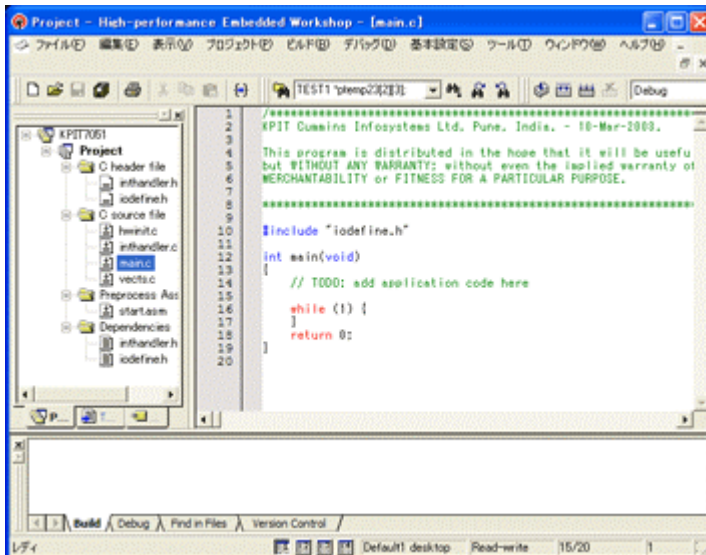
**OK** をクリックします。

ここまでの操作が新規プロジェクトの登録方法です。

## 2. プロジェクトから不要モジュール(ソースファイル)を削除します。

目的: KPI Tにより準備されたモジュールを使用しない場合に削除しておきます。

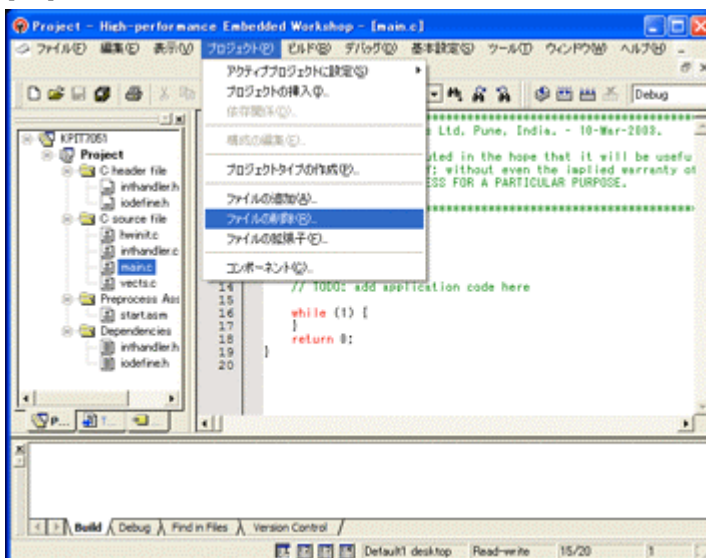
[2-1]



今回の使用例では下記1ファイルを削除します。

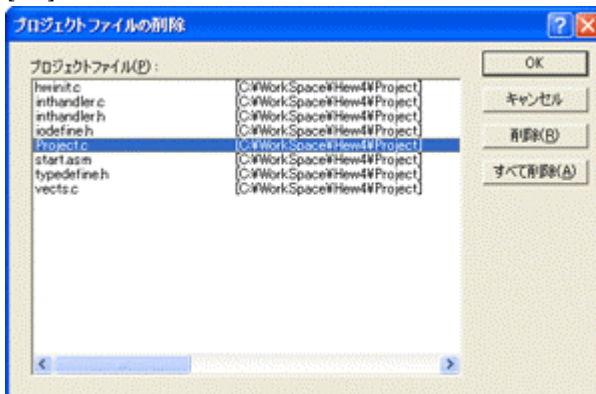
①Project.c

[2-2]



[プロジェクト] –  
[ファイルの削除] をクリックし  
ます。

[2-3]

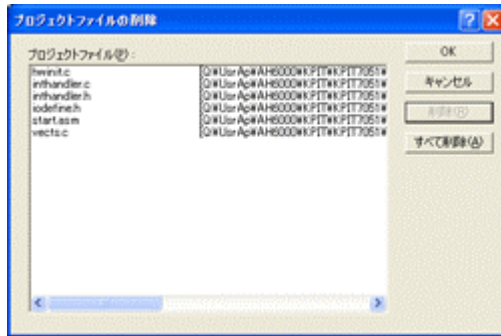


① project.c

の1ファイルを選択する。

**削除**をクリックします。

[2-4]



確認画面です。

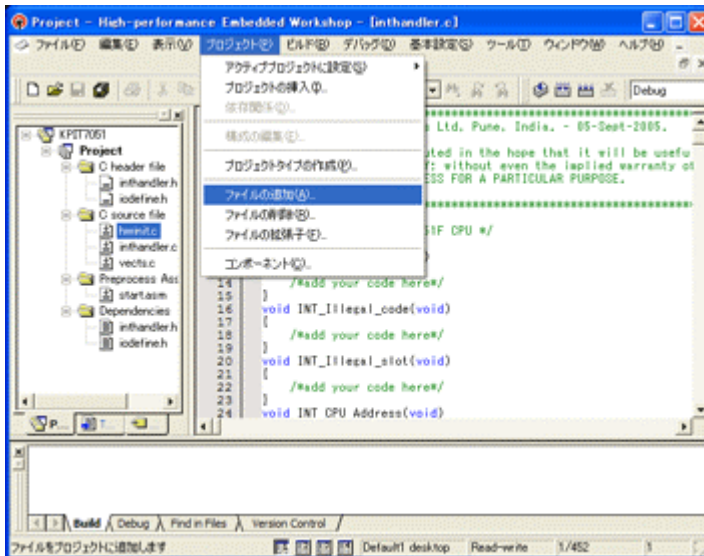
**OK**をクリックします。

### 3. プロジェクトに希望モジュール（ソースファイル）を登録します。

準備： 作成済みの 2 ファイルを”C:\Hew4\KPIT\KPIT705L\_ABS\Project”にコピーします。

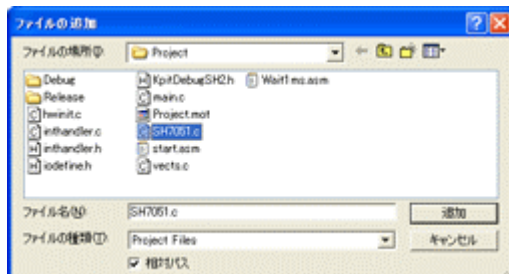
SH7051.c           HP よりダウンロードします。(GNU/g c c)  
KpitDebugSH2.h    KPIT7051\_ABS\_v0801.LZH

[3-1]



[プロジェクト]-  
[ファイルの追加]をクリックしま  
す。

[3-2]



登録ファイルを選択します。

① SH7051.c

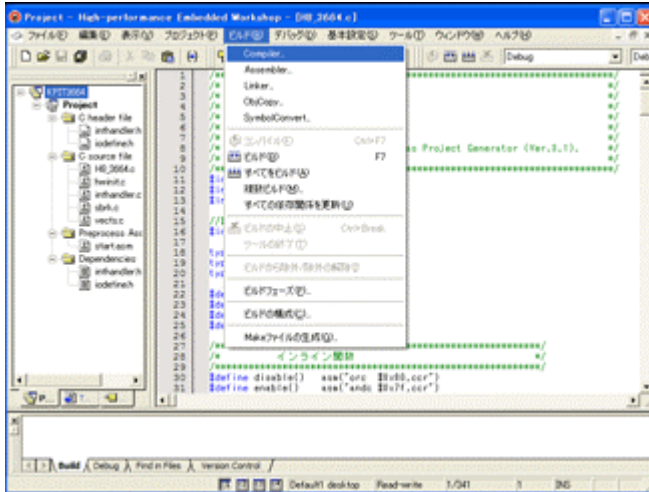
**追加**をクリックします。

この操作によりプロジェクトにモジュールを登録します。

#### 4. コンパイラオプションの確認と設定をします。

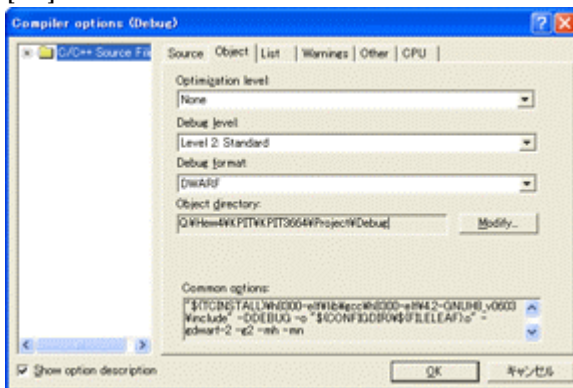
目的： H-debugger でシンボリックデバッグを可能にする為、コンパイラオプションの確認と設定をします。

[4-1]



[ビルド]-  
[Compiler]をクリックします。

[4-2]



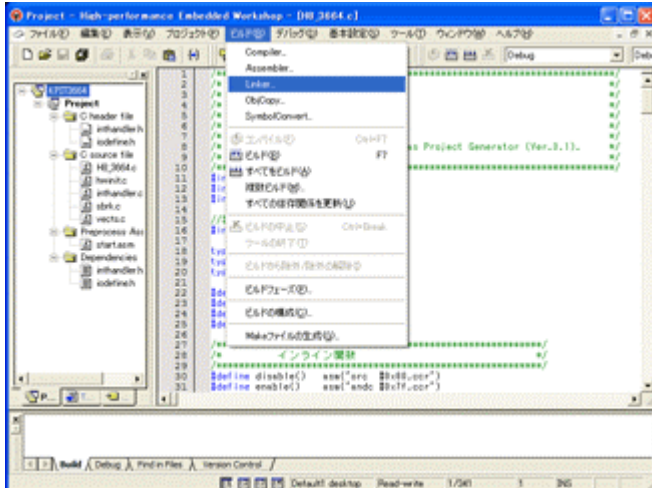
[Object] タグ  
①Optimization : None(Default)  
②Debug level : Level2:Standard(Default)  
③Debug format: **DWARF** に指定する。  
④Object directory : (Default)状態

**OK**をクリックします。

5. リンカーオプションの確認と設定をします。

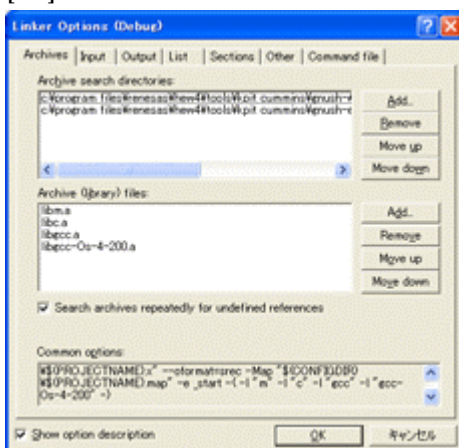
目的： H-debugger でシンボリックデバッグを可能にする為、リンカーオプションの確認と設定をします。

[5-1]



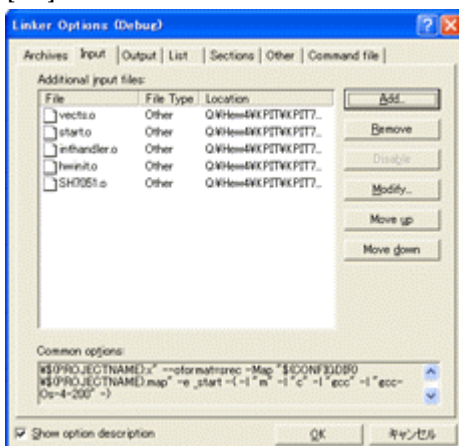
[ビルド]-  
[Linker]をクリックします。

[5-2]



[Archives] タグ  
デフォルト状態です。(変更の必要なし)

[5-3]

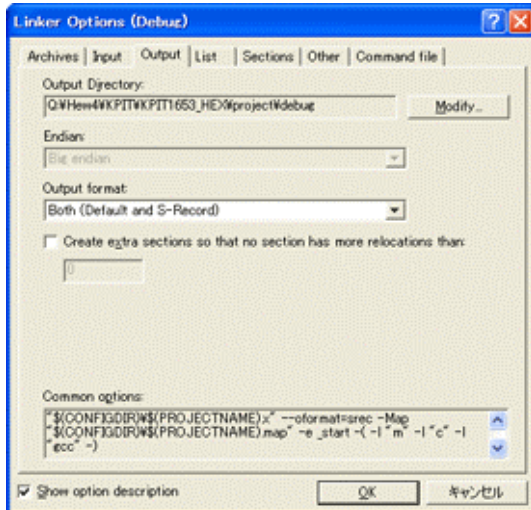


[Input] タグ  
基本的には何も設定しなくて良いですが、各モジュールのリンク順番を指定したい場合に全モジュールをここで指定します。

- ① vects.o
- ② start.o
- ③ inthandler.o
- ④ hwinit.o
- ⑤ SH7051.o



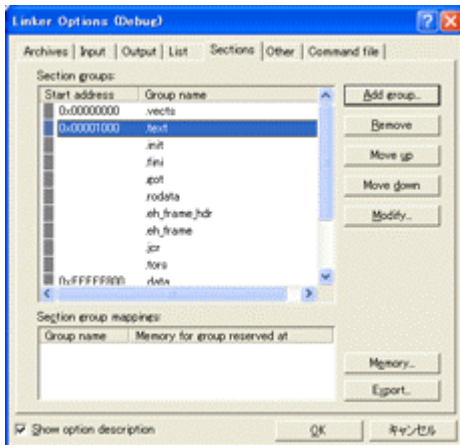
[5-4]



[Output] タグ

- ①Output Directory : (Default)
  - ②Endian : Big endian(Default)
  - ③Output format :Both(Default and S-Record)
- すべて、デフォルトです。

[5-5]

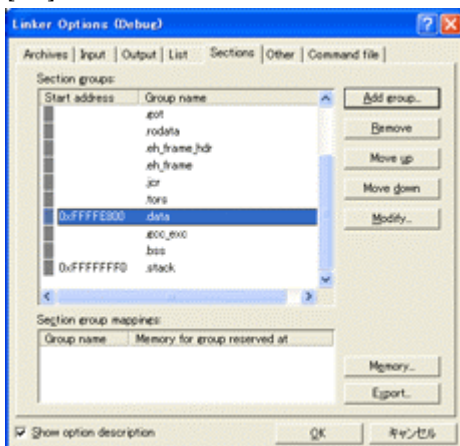


[Sections] タグ

.text セクションの開始アドレスを確認します。  
(モニターエリアを空ける為)

- ①.text セクションの開始アドレスが **【0x1000】** になっているのを確認します。 .

[5-6]

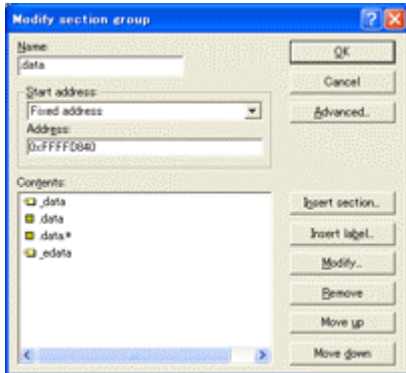


[Sections] タグ

.data セクションのアドレスを変更します。

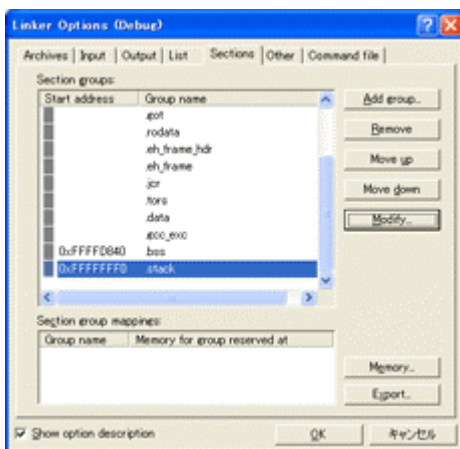
- ①.data セクションを選択します。
- ② **【Modify】** PB をクリックします。

[5-7]



- ③Start address: **【Fixed address】** に選択します。
- ④Address: **.data** セクションの先頭アドレスを指定します。  
ソースブレイクを使用する場合 **【0xFFFFFFD840】**  
ソースブレイクを使用しない場合 **【0xFFFFFFD800】**  
<DEF バージョン 6.50A より>  
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して下さい。
- ⑤**OK** をクリックします。

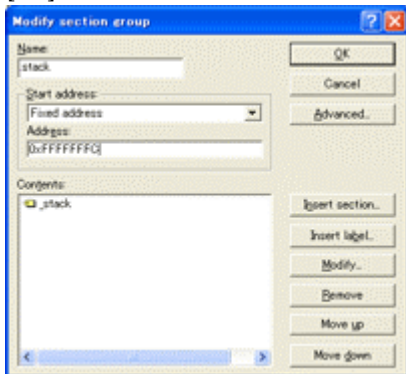
[5-8]



[Sections] タグ  
**.stack** セクションのアドレスを指定します。  
ここでの指定値は、スタックポインタへの初期設定値になります。

- ①**.stack** セクションを選択します。
- ② **【Modify】 PB** をクリックします。

[5-9]

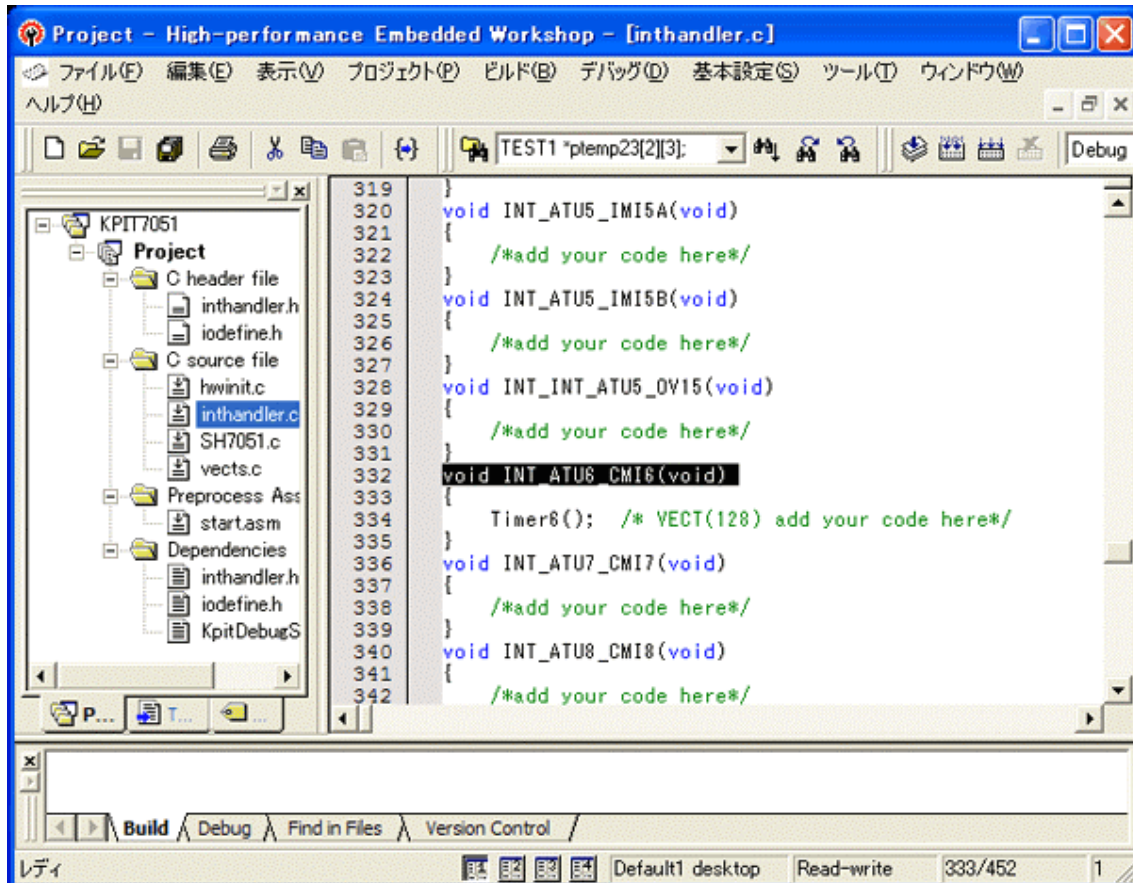


- ③Address: を「0xFFFFFFF0C」に変更します。
- ④**OK** をクリックします。

## 6. 割り込みハンドラへ登録します。

目的： 今回説明に使用したモジュール「SH7051.c」は、Timer6（ベクター128）の割り込みを使用していますので、割り込みハンドラへ登録します。

[6-1]



① inhandler.c を選択します。

② void INT\_ATU6\_CMI6(void) { Timer6(); } の関数を記述します。

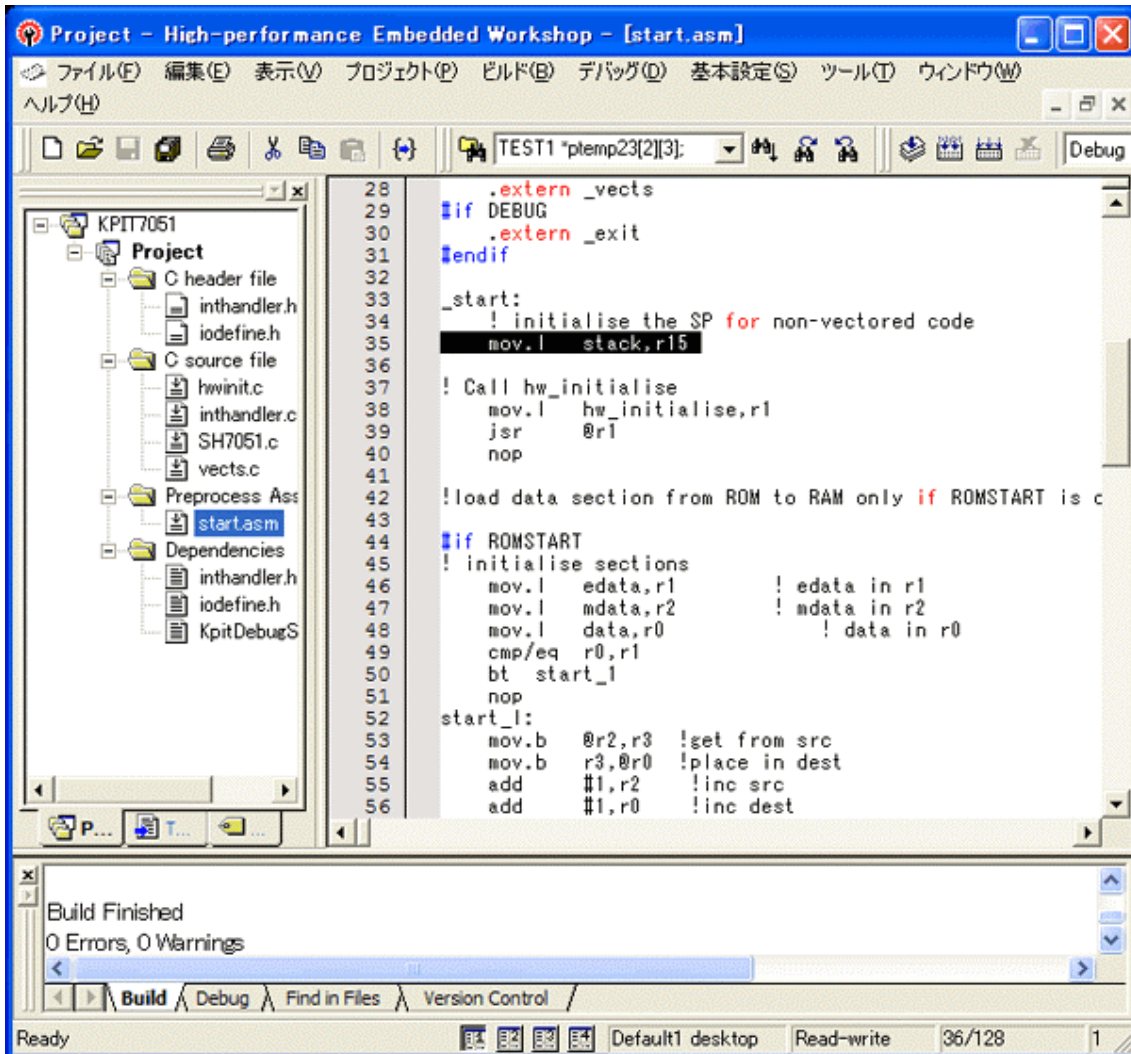
### 【注意】

① 「vects.c」の256ベクターに「INT\_Dummy」が登録されています。これは0x400番地を超えますし、意味の無い登録ですので削除します。（このサンプルでは、削除済みです。）

## 7. スタートアップ「start.asm」の説明です。

目的： スタートアップ「start.asm」に実際は不要なソースコードがありますが、その補足説明です。

[7-1]



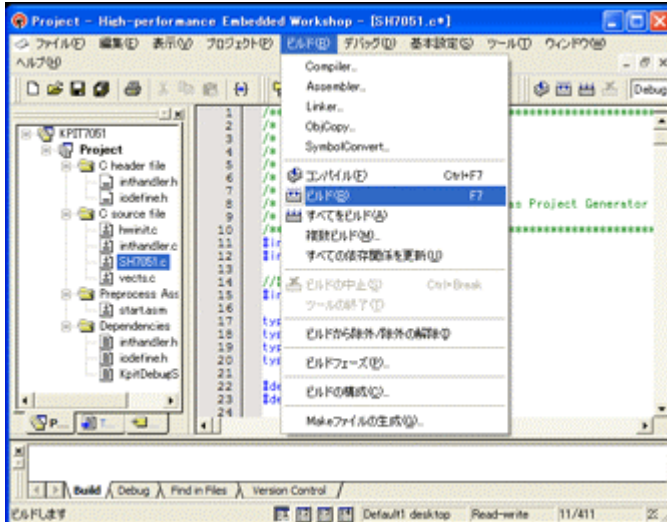
①35行目に「mov.l stack, r15」と記述してあります。本来SH-2の場合は、ベクタ1にスタックポインタ値が格納されており、リセット解除時にハード側にて設定されます。しかし、デバッガ側でリセットコマンドを発行した場合、リセット解除後、NMIを起動し、PC値をベクター0値に書き換えています。つまりリセット解除後、NMI起動までにタイムラグがある為、ターゲットCPUは若干実行しますのでスタックポインタ値が変わってしまいます。この行のように再設定するのはSP値の復帰になりますので意味のあることになります。又、デバッグ終了時でも、このソース行を残しておいても問題ありません。

②他の方法としては、今迄弊社が推奨しているソフトタイマー1～200msを入れる方法もあります。この方法ですとNMI起動まではソフトタイマ処理中ですのでスタックポインタも内部I/Oレジスタの内容も書き変わりません。  
どちらにするかは、それぞれの事情によって選択して下さい。

## 8. ビルドを実行します。

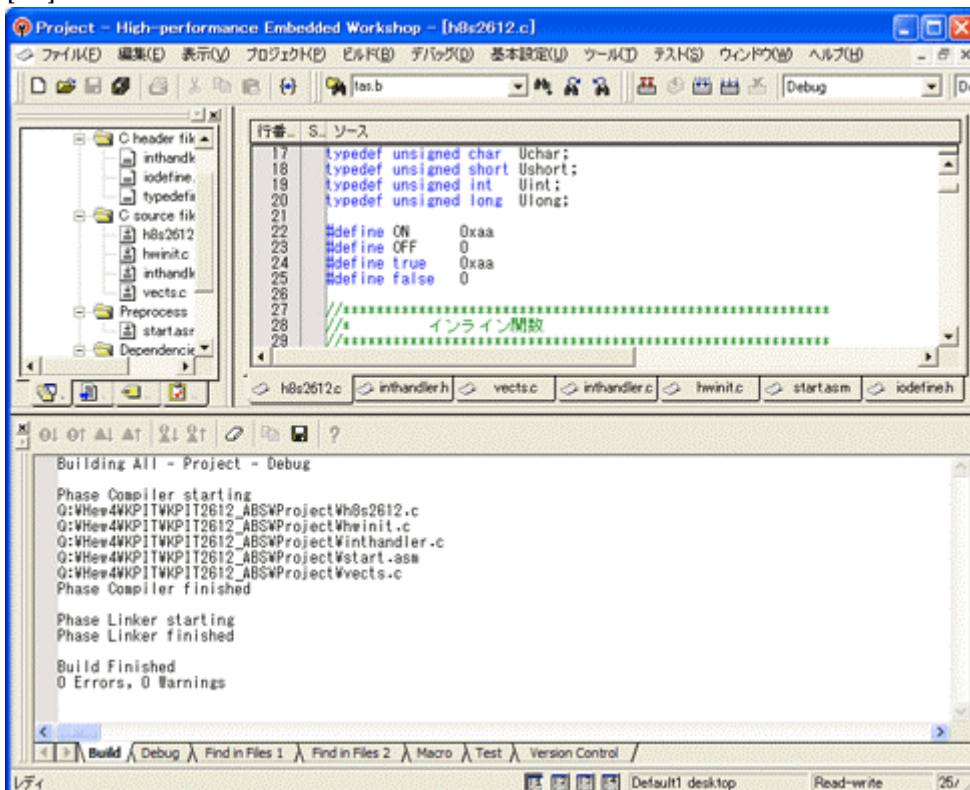
目的： コンパイル／アセンブリ／リンクロケートを実行させる為、ビルドを実行します。

[8-1]



[ビルド]-  
[ビルド]をクリックします。

[8-2]

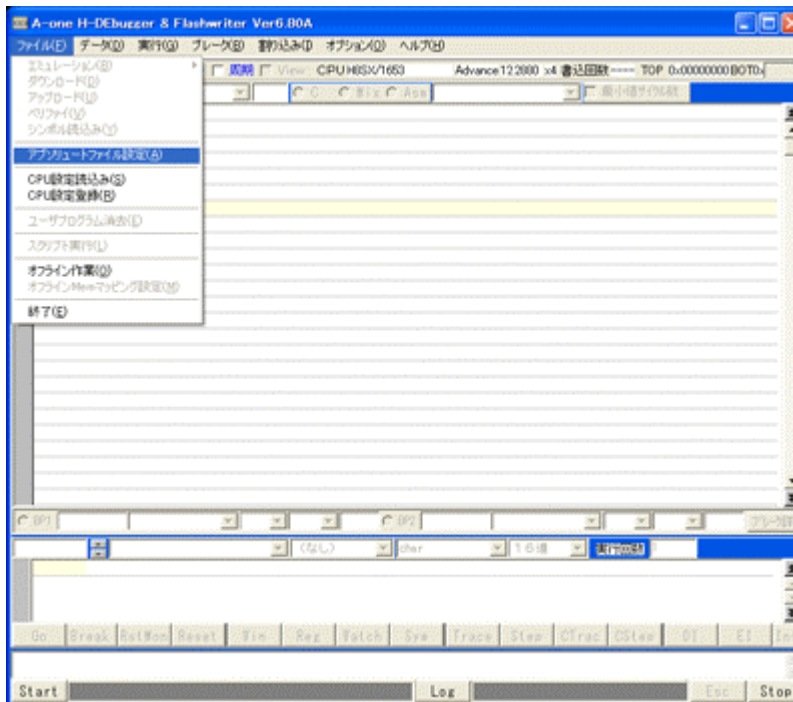


↑のように「0 Errors, 0 Warnings」になれば成功です。

## 9. 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。

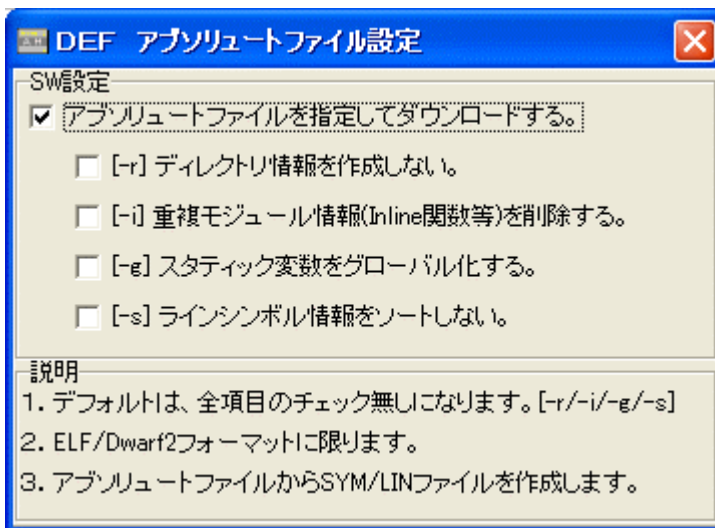
[9-1]



<ファイル>—  
<アブソリュート設定>  
を指定します。

2) 設定します。

[9-2]



左画面のように  
「チェック」を入れて下さい。

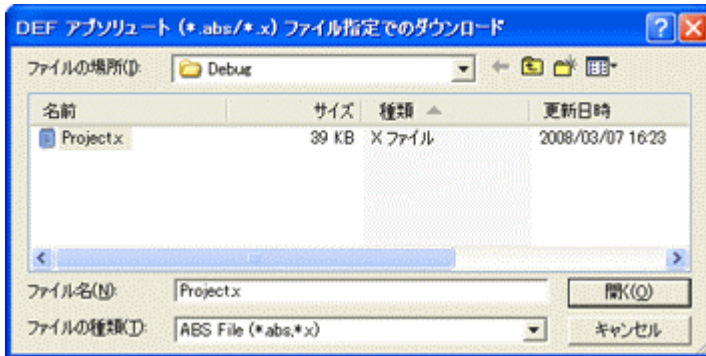
He wデフォルト設定の状態で使用  
する場合は、  
[r]をチェックしないで下さい。

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。

[9-3]



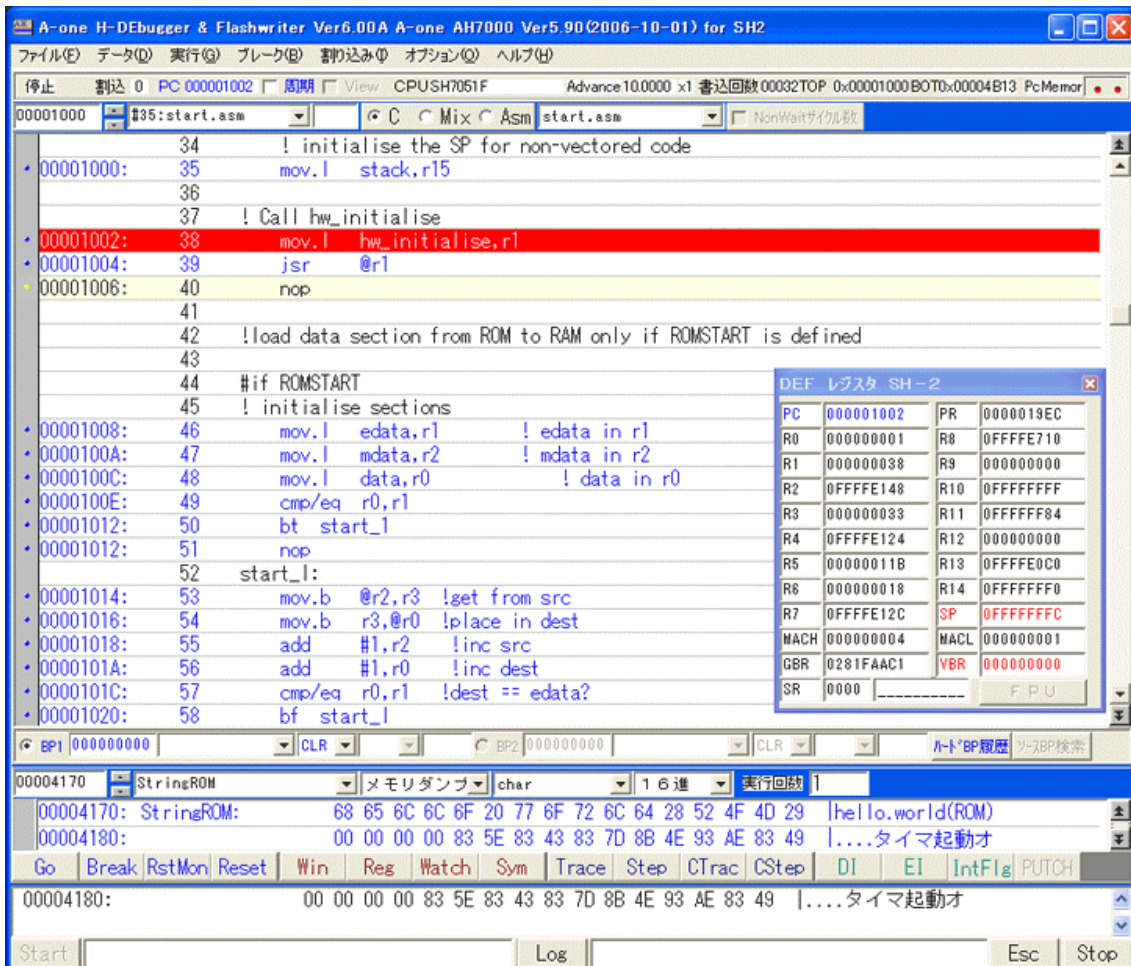
<ファイル>—  
<ダウンロード>  
を指定します。

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) <start.asm>の確認

[9-4]



- ① 1000H 番地にスタックポインタの設定コードがあります。
- ② SP値が「0xFFFFFFFF」値になっているのが確認できます。

これで「H-Debugger」用の設定作業が終了です。

以上