

2008年03月10日
エーワン Rev 1. 00

Hew (Ver 4. 04) +KPIT-GNU (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
(H8SX/1653 H-UDI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	KPIT1653_ABS		
プロジェクト名	Project		
登録モジュール名	H8SX1653.c	C	メインモジュール (アプリ用)
	KpitDebugH8_2.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。)
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
	sbrk.c	C	ヒープメモリー用
	iodef.h	ヘッダファイル	I/O 定義ビットフィールド記述用 (一部修正しました。)
	inhandler.h	ヘッダファイル	割込みハンドラー用
CPUタイプ	H8SX/1653F		

 【ポイント】

「H8SX H-UDI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。

の1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。

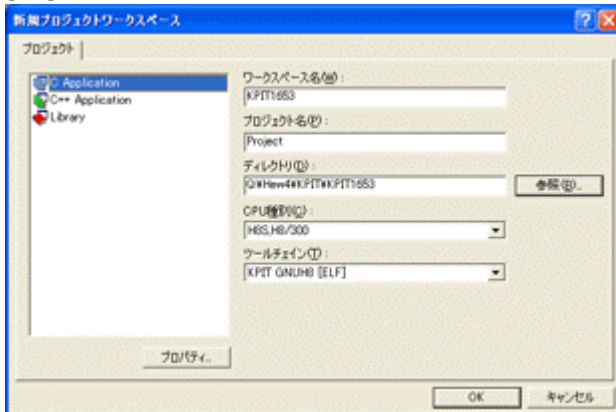
[1-1]



“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。

または、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。

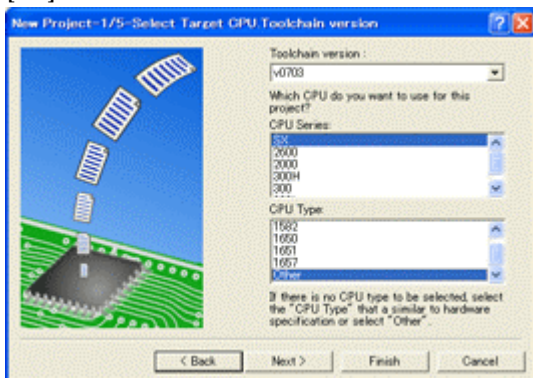
[1-2]



ワークスペース名 KPIT1653_ABS
プロジェクト名 Project
ディレクトリ C:\Hew4\KPIT\KPIT1653
CPU 種別 H8S, H8/300
ツールチェーン KPIT GNUH8 [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]

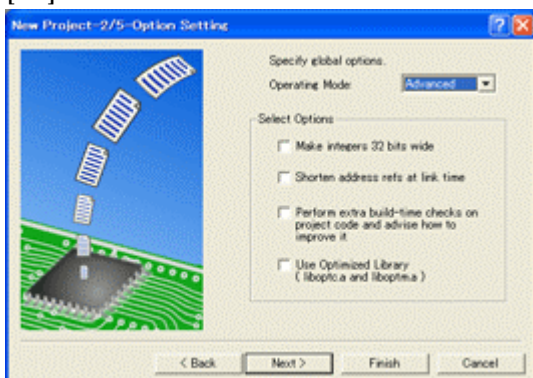


CPUスペックを選択します。

- ①SX
- ②Other

Next をクリックします。

[1-4]

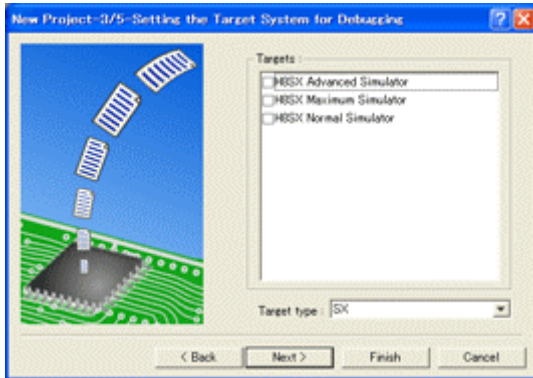


CPUオプションを選択します。

- ① 「Operation Mode」
<Advanced>を選択します
- ② 「Use Optimized Libraries」
チェックを外します。

Next をクリックします。

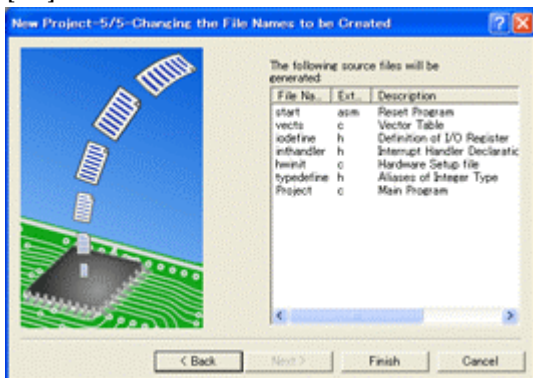
[1-5]



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。

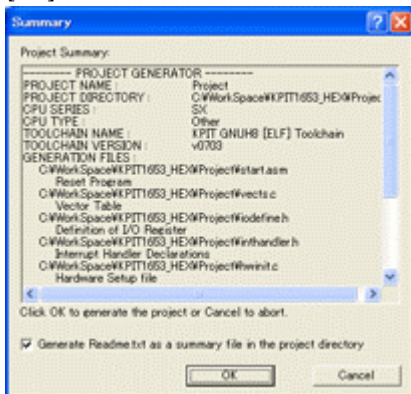
[1-6]



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。

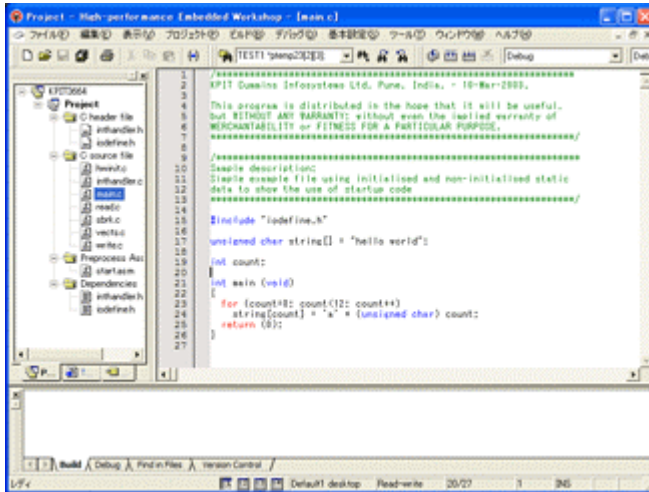
OK をクリックします。

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： K P I 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。

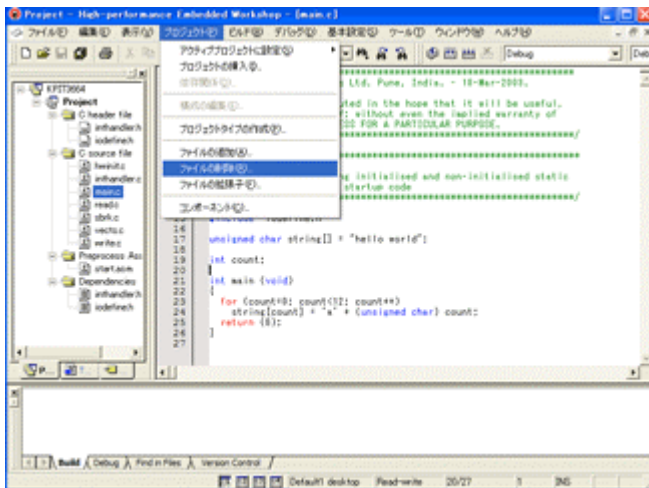
[2-1]



今回の使用例では下記1 ファイルを削除します。

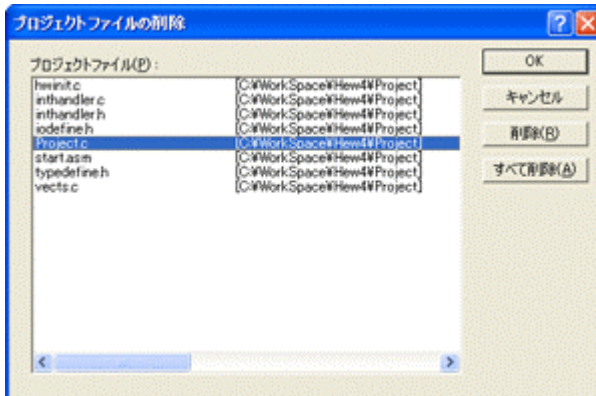
① Project.c

[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]

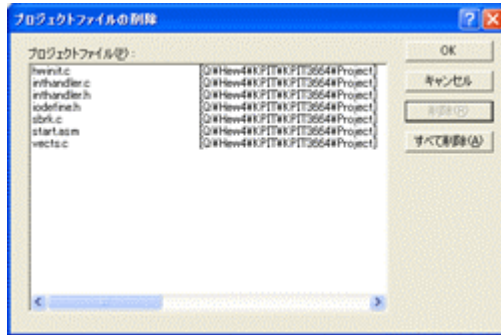


① project.c

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。

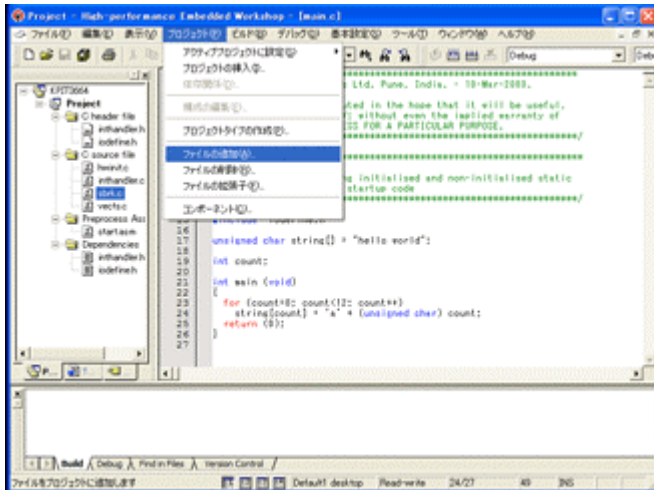
OKをクリックします。

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3ファイルを”C:\Hew4\KPIT\KPIT1653_ABS\Project”にコピーします。

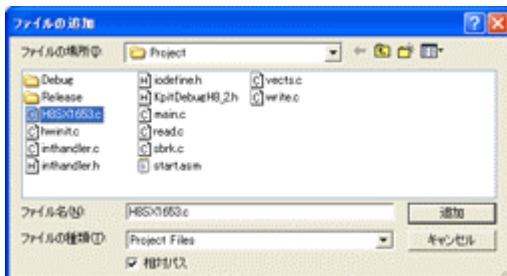
H8SX1653.c HP よりダウンロードします。(GNU/gcc)
KpitDebugH8_2.h KPIT1653_ABS_v0801.LZH
iodefine.h 一部修正しましたので Copy して下さい。

[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 H8SX1653.c

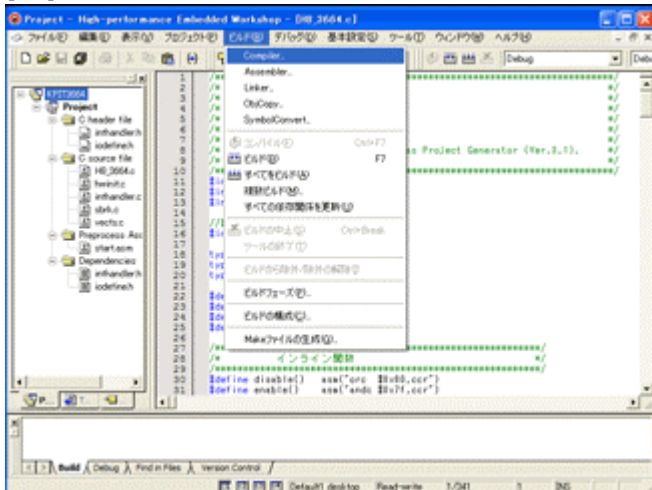
追加をクリックします。

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。

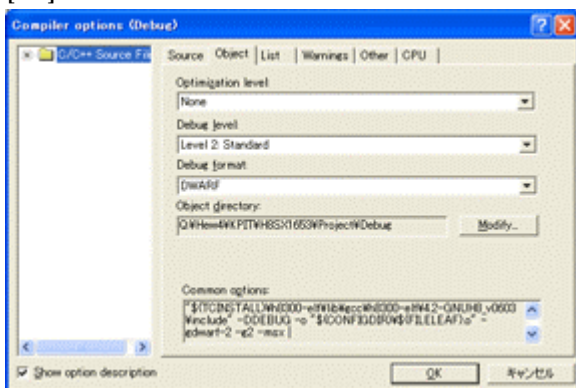
目的： H-debugger でシンボリックデバッグを可能にする為、コンパイラオプションの確認と設定をします。

[4-1]



[ビルド]-
[Compiler]をクリックします。

[4-2]



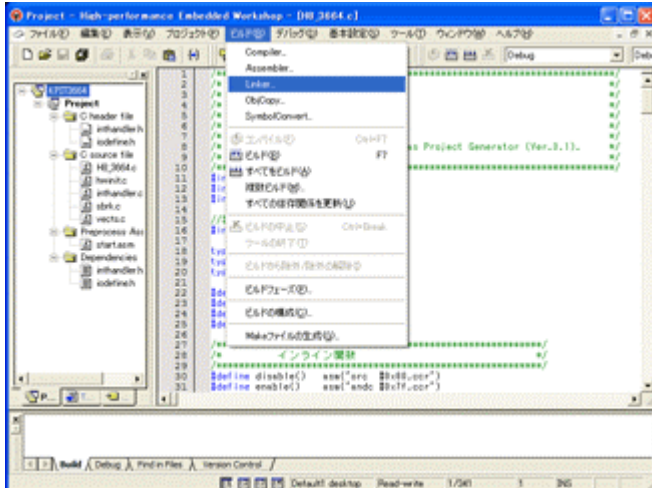
[Object] タグ
① Optimization : None(Default)
② Debug level : Level2:Standard(Default)
③ Debug format: **DWARF** に指定する。
④ Object directory : (Default)状態

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。

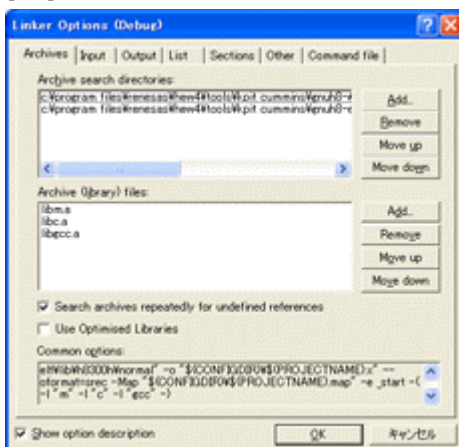
目的： H-debugger でシンボリックデバッグを可能にする為、リンカーオプションの確認と設定をします。

[5-1]



[ビルド]-
[Linker]をクリックします。

[5-2]

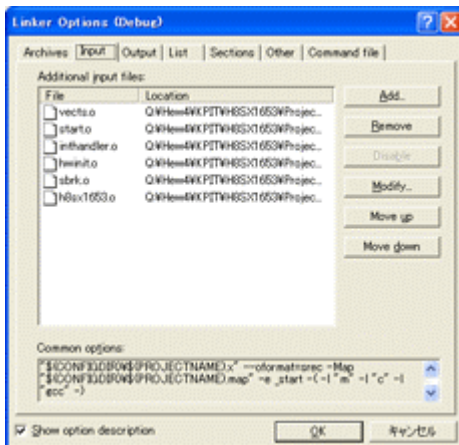


[Archives] タグ
デフォルト状態です。(変更の必要なし)

【HowTo】

「Archive search directories」の情報が何らかの原因により変わってしまった場合、「Use Optimised Libraries」のチェックを付けて外しますとデフォルト状態に戻ります。

[5-3]

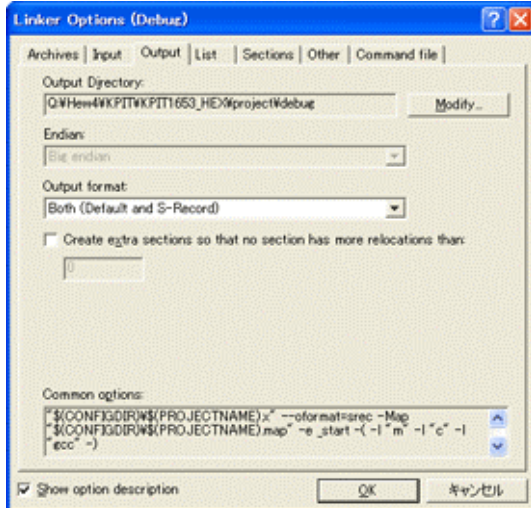


[Input] タグ

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 inhandler.o
- ④ hwinit.o
- ⑤ sbrk.o
- ⑥ H8S1653.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]

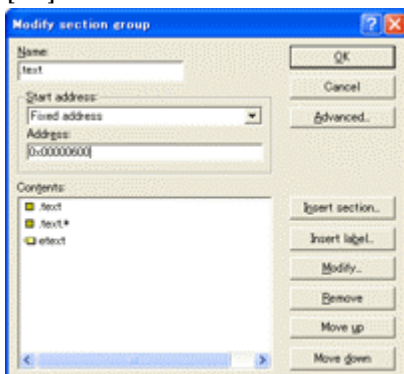


[Sections] タグ

.text セクションの開始アドレスを変更します。
デフォルトで「0x400」になっています。
「0x400」番地から可能ですが、DTC ベクターテーブル
を考慮して「0x600」番地に変更します。

- ①.text セクションを選択します。
- ② [Modify] PB をクリックします。

[5-6]



③Address: を「0x600」に変更します。

④OK をクリックします。

[5-7]

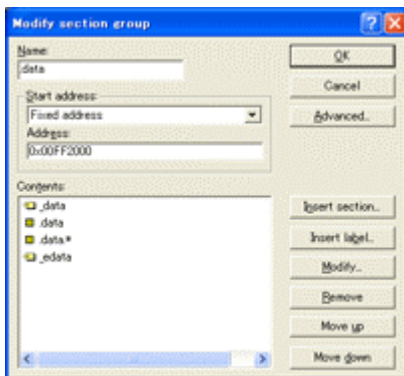


[Sections] タグ

.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。

[5-8]



③ Start address: **【Fixed address】** に選択します。

④ Address: .bss セクションの先頭アドレスを指定します。
【0xFF2000】

ソースブレイク使用の指定の場合でも内臓RAMは使用しません。

⑤ **【OK】** をクリックします。

[5-9]



[Sections] タグ

.stack セクションのアドレスを指定します。

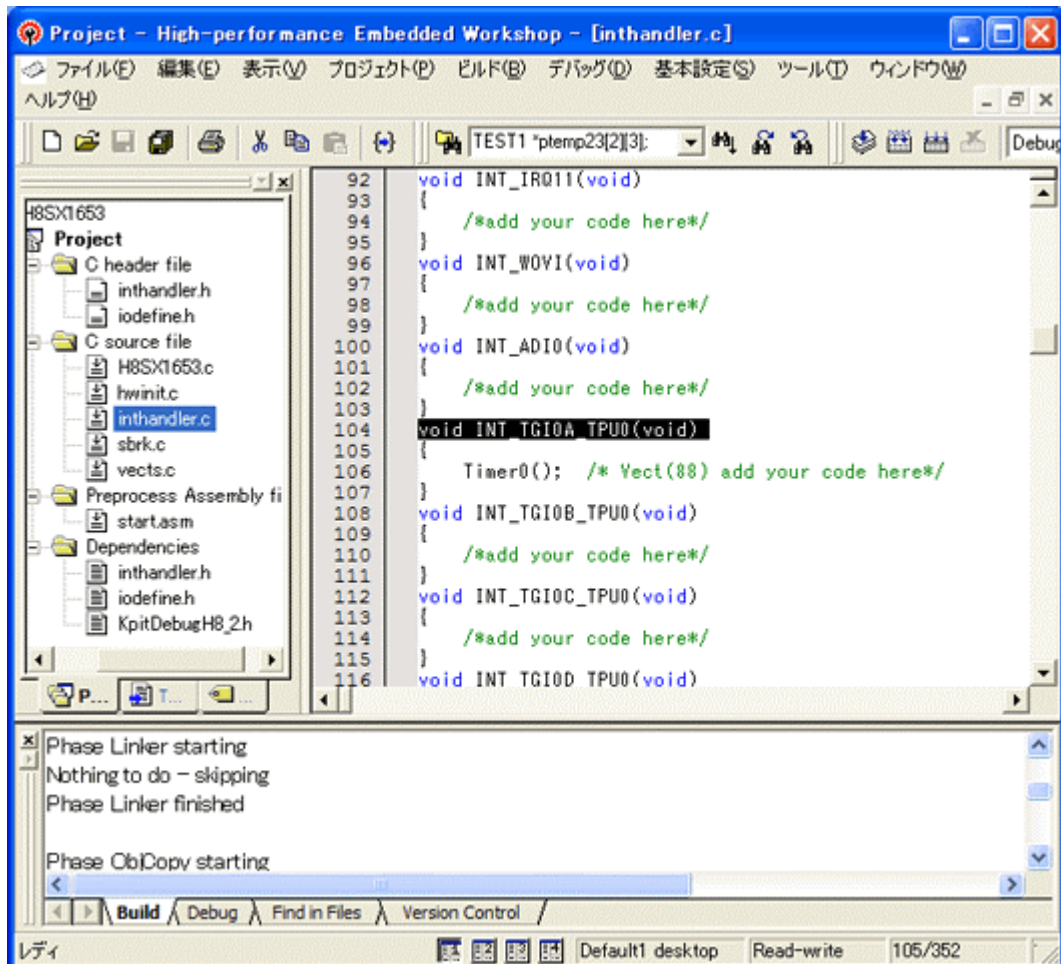
ここでの指定値は、スタックポインタへの初期設定値になります。

- ① デフォルト値「**0xFFC000**」を確認する。
- ② **【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「H8SX1653.c」は、Timer0（ベクター88）の割り込みを使用していますので、割り込みハンドラへの登録をします。

[6-1]



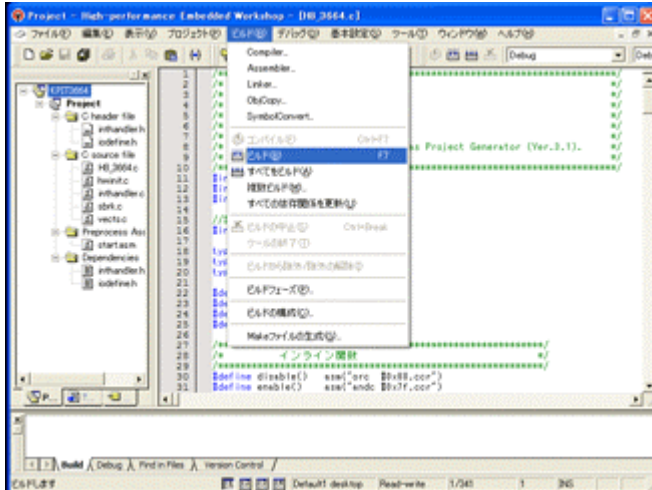
① inthandler.c を選択します。

② void INT_TG10A_TPU0(void) { Timer0(); } の関数を記述します。

7. ビルドを実行します。

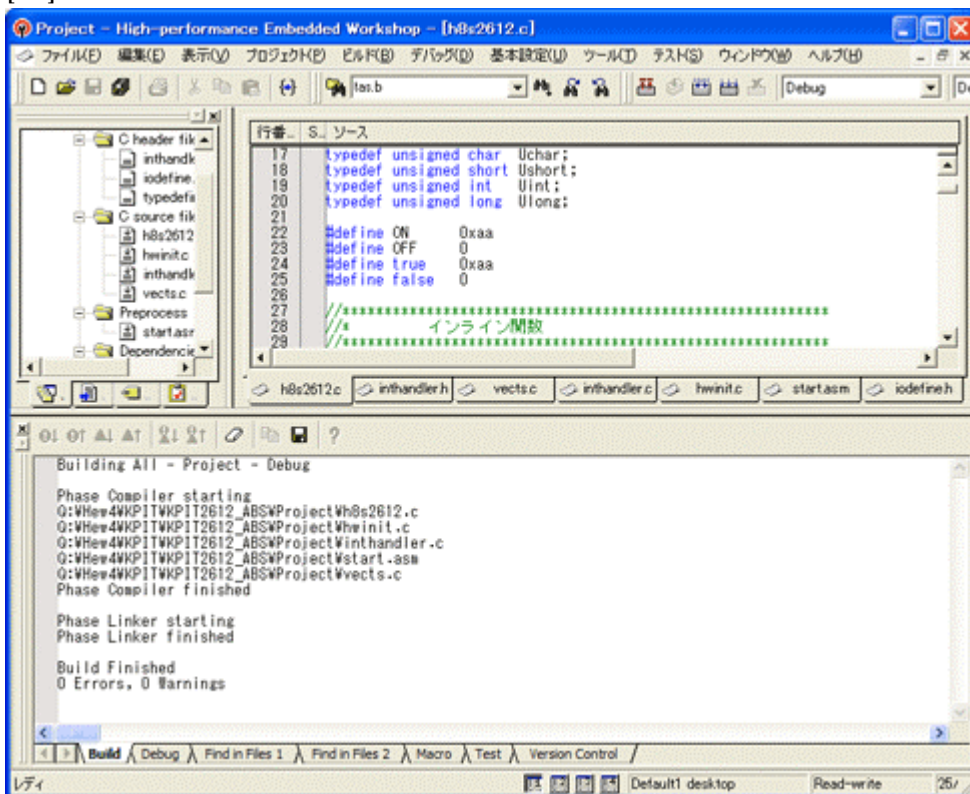
目的： コンパイル／アセンブリ／リンクロケートを実行させる為、ビルドを実行します。

[7-1]



[ビルド]-
[ビルド]をクリックします。

[7-2]

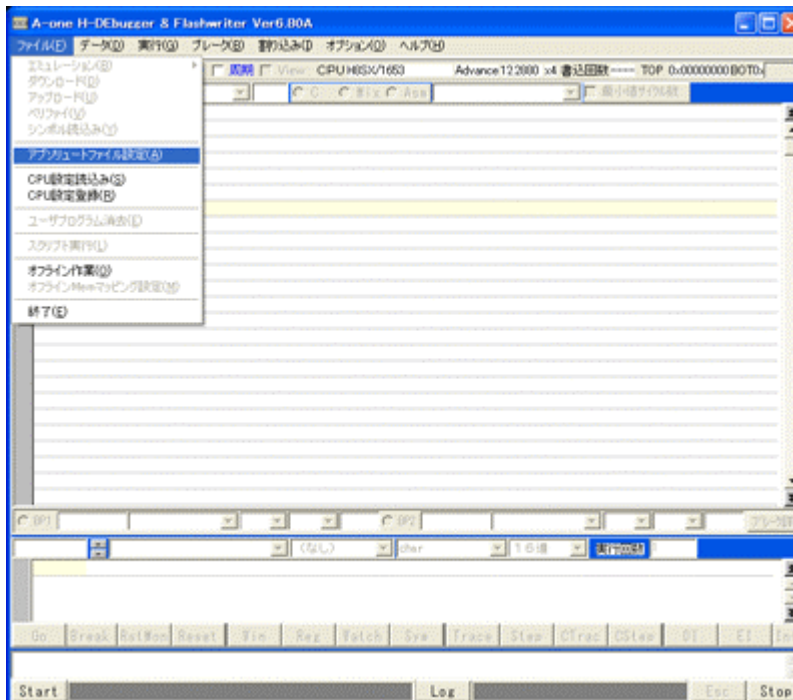


↑のように「0 Errors, 0 Warnings」になれば成功です。

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。

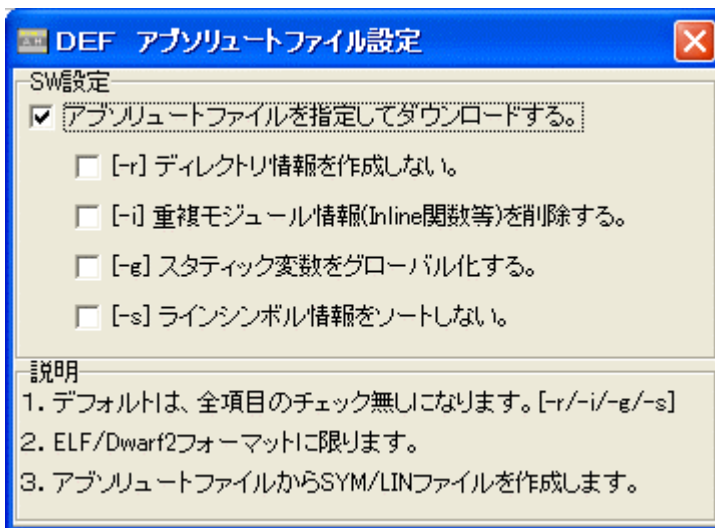
[8-1]



<ファイル>—
<アブソリュート設定>
を指定します。

2) 設定します。

[8-2]



左画面のように
「チェック」を入れて下さい。

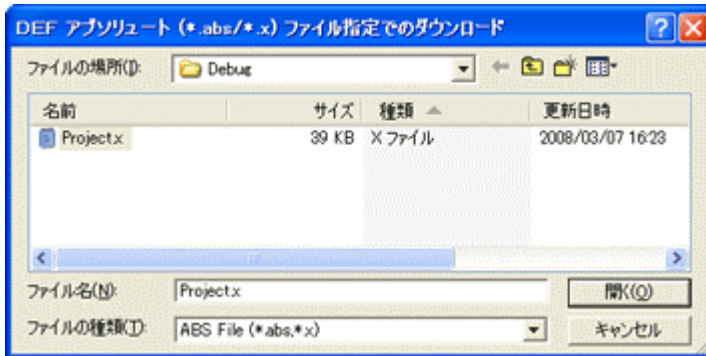
He wデフォルト設定の状態で使用
する場合は、
[r]をチェックしないで下さい。

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。

[8-3]



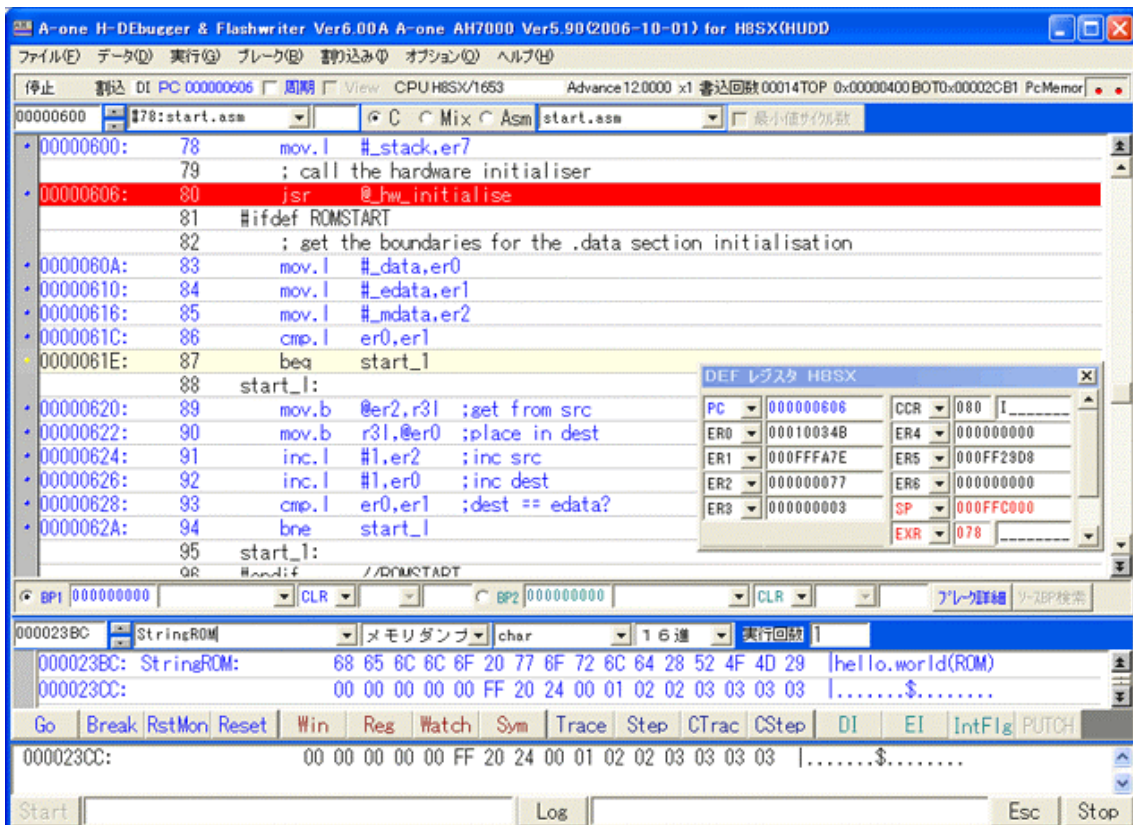
<ファイル>—
<ダウンロード>
を指定します。

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) <start.asm>の確認

[8-4]



①600H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」用の設定作業が終了です。

以上