アブソリュートファイル指定でダウンロードする場合 DEF Ver6.80Aより

2008年03月10日 エーワン Rev1.00

Hew (Ver4.04) + 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

(H8**S**/2612 BOOT版)

KPIT版の統合環境「Hew Ver4.04」で H-debugger 用に新ワークスペース/プロジ エ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	KPIT2612_ABS		
プロジェクト名	Project		
登録モジュール名	h8s2612. c	С	メインモジュール(アプリ用)
	KpitDebugH8.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	С	ハード初期化用モジュール
	vects.c	С	リセット/割込みベクターテーブル
	inthandler.c	С	割込みハンドラー用
	sbrk.c	С	ヒープメモリー用
	iodefine.h	ヘッダファイル	I/0 定義ビットフィールド記述用
	inthandler.h	ヘッダファイル	割込みハンドラー用
CPUタイプ	H8S/2612F		



「H8S 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。



"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。

加ブロジェクトワークスペース		2 🗙	ワークスへ。
709291⊧ ©C++ Application ©C++ Application € Library	ワークスペース名(他): [XPT2652 プロジョント名(型): [Project ディレクドリ(型): [2PH5440971785/172852 CPU感的(型): [HES,HE/200 ウールチェイン(型): [XPTT GAULIE (ELF]		アジュ デルク CPU種 ツールチェイ アジュ このり
20/0	¥		て下さ

名 KPIT2612_ABS 名 Project C:¥Hew4¥KPIT¥KPIT2612 H8S, H8/300 KPIT GNUH8[ELF] Application

を設定確認後OKをクリックし

•



Perform extra build-time checks or project code and advise how to intercore a

Use Optimized Library (liboptc.a and liboptm.a)

< Back Next > Finish Cancel





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。



[1-7]



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: KPIT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。



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] ー [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]		2 🗙	1
プログェクトファイル使): Invinite infhundler:c infhundler:b idotic startaan vectsc	OWHere 4KPTWKPT3654Project OWHere 4KPTWKPT3654Project OWHere 4KPTWKPT3654Project OWHERE 4KPTWKPT3654Project OWHERE 4KPTWKPT3654Project OWHERE 4KPTWKPT3654Project OWHERE 4KPTWKPT3654Project	0K #+>/Us R3005 # <cbrb#(a)< td=""><td>確認画面です。</td></cbrb#(a)<>	確認画面です。
c			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"Q: ¥Hew4¥KPIT¥KPIT2612_ABS¥Project"にコピーします。

 h8s2612.c
 HPよりダウンロードします。(GNU/gcc)

 KpitDebugH8.h
 KPIT2612_ABS_v0801.LZH



[プロジェクト]-[ファイルの追加]をクリックします。

[3-2]



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、コンパイラーオプションの確認 と設定をします。



[ビルド]-[Compiler]をクリックします。

[4-2]

C/C++ Source File	Source Object List Warnings Other CPU	
	Optimigation level	
	None	*
	Debug jevel	
	Level 2 Standard	+
	Debug format	
	DwARF	
	Object girectory:	
	Q.WHem4WK.PITWK.PIT3664WProjectWDebug Model	y
	Common options: "31TCINSTALLY#K000-effitExeccifil000-effit2-GNUH8.v06 Winclude: = ODEIL00 - or 31000F3DDRW8/FILELEAF1or - getwert-2 = mh - mi	00 <u>~</u>
	jedwart-2 -e2 -mh -mn	
and a brief of the state of the state of the		

[Object] タグ ①Optimization:None(Default) ②Debug level: Level2:Standard(Default) ③Debug format:DWRAF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。



[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 チェック覧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 inthandler.o
- ④ hwinit.o
- (5) sbrk.o
- 6 h8s2612.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。
 (モニターエリアを空ける為)

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「0x800」に変更します。
 ④OK をクリックします。



Section groups:				1977
Start address	Group name			Add group.
	rodata			Bemove
	.eh.frame.jkdr .eh.frame		1	Move up
	jor have			Move gown
0x00FFE000	data			Modify_
0x00FFEFC0	.goojexo bas .stack			
<		2		
Segtion group maps	pings Memory for entern reserved at		-	
Group name	memory for group reserves as			
				Memory.
Group name	Memory for group reserved at			Memory.

[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]



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。 [0xFFE000]

<DEF バージョン 6.50A より>

ソースブレークを使用する場合は、内臓 RAM2の 先頭から 64 バイト[0x40]使用しますが、モニタワーク方 式をスタック方式にする必要があります。

⑤OK 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.stack セクションを選択します。 ②【Modify】PBをクリックします。

[5-10]



③Address: を「0xFFEFBE」に変更します。

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「**h8s2612.c**」は、**Timer0**(ベクター32)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。



① inthandler.c を選択します。

②void INT_TGIA0_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。



[ビルド]-

[ビルド]をクリックします。





↑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。



2) 設定します。



[8-3]

ファイルの場所(1):	C Debug	- 🖛 🖬 -			
名前		サイズ	種類 🔺	更新日時	
🗊 Project x		39 KB	× ファイル	2008/03/07 16:23	
<	T.				
ファイル名(N):	Projectx			■■((_)	
ファイルの種類(工):	ABS File (* abs	(x*,x)		 キャンセル 	

<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) <s t a r t. a sm>の確認

[8-4]

A-one H-DEbugger & Flashwriter Ver6,00A A-one AH7000 Ver5,90(2006-10-01 ファイル(な) デッカ(の) まだ(な) ゴルッカ(の) まわえる(の オブルマン(の) ヘルゴル)	1) for H8S	- 🗆 🛛			
19正 新た U PC 000000806 V JAJAA) View CPU H85/2612F Advance 200000 x	(1 書达回敏 0000210P 0x0000800B010x000036B1 PcMe	smor 🙍 🔹			
00000800 🕂 179:start.asm 💽 🔍 C C Mix C Asm start.asm	NonWaitサイクル教				
• 00000800: 79 mov.1 #_stack,er7		*			
80 ; call the hardware initialiser		<u> </u>			
<u> 00000806: 81 jsr @_hw_initialise</u>					
82 #ifdef ROMSTART					
83 ; get the boundaries for the .data section	on initialisation				
* 0000080A: 84 mov.1 #_data,er0					
*00000810: 85 mov.1 #_edata.er1					
* 00000816: 86 mov.1 #_mdata,er2					
* 0000081G: 87 cmp.1 er0,er1					
*0000081E: 88 beq start_1					
89 start_1:	_ DEF レジスタ H8S	×			
• 00000820; 90 mov.b @er2,r31 ;get from src	PC - 000000806 CCR - 0E1 IIH	ī 📥 🗕			
• 00000822; 91 mov.b r31,eer0 ;place in dest	ER0 - 000FFE8AA ER4 - 059C60000	-			
1 00000824; 82 Inc.1 #1,erz ; Inc.src	FR1 V 000FFE95A FR5 V 0FFFF0000	- 1			
• 00000020; 33 Inc. I #I,er0 ; Inc dest	FR2 V 000002526 FR6 V 000FFEFB6	-			
- 00000020; 34 Cmp.1 eru,eri ;dest edata;	FR3 > 000FFE840 SP > 000FFEFBE	-			
0000002A- 00 bite start_1					
97 Handif //POWSTADT		-			
		ㅋ			
, zero out uss		<u>*</u>			
← BP1 000000000	▼ CLR ▼ アーメート*BP履歴 ソースBP	検索			
00002DB4 🚆 StringROM 🛛 メモリダンブラ char 💌 16雄	i 💌 実行回数 1				
00002DB4: StringROM: 68 65 6C 6C 6F 20 77 6F 72 6C 64 28	52 4F 4D 29 [hello.world(ROM)	±			
000002DC4: 00 83 5E 83 43 83 7D 8B 4E 93 AE 83	49 83 93 28 .タイマ起動オン(Ŧ			
Go Break RstMon Reset Win Reg Watch Sym Trace Step	CTrac CStep DI EI IntFls PUT	СН 📃			
00002DC4: 00 83 5E 83 43 83 7D 8B 4E 93 AE 83 49 83 93 28 しタイマ記動オン(
	- ,	~			
Start Log	Esc	Stop			

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」用の設定作業が終了です。

以上