HEXファイル指定でダウンロードする場合

Hew (Ver4.04) + KPIT (v0703) 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

(SH-2/7145 H-UDI版)

KPIT版の統合環境「Hew Ver4.00」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 プロジェクト名 登録モジュール名	KPIT7145 Project SH7145.c	С	メインモジュール (アプリ用)
KPIT 添付ファイル C P Uタイプ	start.asm hwinit.c vects.c inthandler.c iodefine.h inthandler.h SH7145F	A SM C C C ヘッダ ヘッダ	スタートアップモジュール ハード初期化用モジュール リセット/割込みベクターテーブル 割込みハンドラー用 I/0 定義ビットフィールド記述用 割込みハンドラー用

1. 新ワークスペースの登録方法 "HEW" 起動させます。

[1-1]

19251	28
オプション	OK
	キャンセル
● 最近使用したプロジェクトワークスペースを聞く(2):	アドモニストレーション(日).
● ○ 割約プロジェクトワークスペースを参照する型	

"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]				
新規プロジェクトワークスペース		2 🔀	リークスペース名	KPIT7145
70/2014	ワーウスペース&〈@〉: FTT745 プロジェクト-&〈史〉: Project デイトクトリ(①): QiiHenr48KPT18KPT7145 CPU種類N(②): SuperH RSC engine ツールチェイン(①): [KPT (MUSH [ELF]	9 版②	プ vジ ェクト名 デ ィレクトリ CPU 種別 ツー・チェイン プ vジ ェクト この項目を	Project Q:¥Hew4¥KPIT¥KPIT7145 SuperH RISC engine KPIT GNUSH[ELF] Application 設定確認後 <mark>OK</mark> をクリックし
70/9	H.		て下さい。	
	OK	キャンセル		





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。

[2-1]



今回の使用例では下記1ファイル を削除します。



[2-2]



[プロジェクト] ー [ファイルの削除] をクリックし ます。



プロジェクトファイルの削除	2 🛛	(1) project c
プロジェクトファイル(ビ): Intrinite Inthandler c Inthandler h GWWork:SpaceWHew4WProject Inthandler h GWWork:SpaceWHew4WProject Pojecto CWWork:SpaceWHew4WProject Start asm CWWork:SpaceWHew4WProject Start asm CWWork:SpaceWHew4WProject Start asm CWWork:SpaceWHew4WProject CWWork:SpaceWHew4WProject	0K キャンセル 所印(B) すべて用即(A)	の1ファイルを選択する。 削除をクリックします。

[2-4] プロジェクトファイルの削除 確認画面です。 2 🗙 プロジェクトファイルモン OK H6000WKPTWKPT7051W H6000WKPTWKPT7051W H6000WKPTWKPT7051W H6000WKPTWKPT7051W H6000WKPTWKPT7051W H6000WKPTWKPT7051W AMAHRODOXYPTOXY hwinitc inthandier c inthandier h iodefine h start as m キャンセル 1<(1)(4)(4)(4) < 5

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1ファイルを"C:\Hew4\KPIT\KPIT7145\Project"にコピーして下さい。

SH7145. c HPよりダウンロードします。(GNU/gcc) KPIT7145_v0703_1.LZH





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. シンボルコンバータ「GC s ymc on v」を登録します。

- 目的:H-debugger でシンボリックデバッグする為にアブソリュートファイル【Project.x】からシンボル情報抽出します。
- 準備: KPIT 用シンボルコンバータ【GCsymconvexe】をホームページよりダウンロードし解凍後、DEF インスト ールDIR「C:¥Program Files¥Aone¥DEF」下にコピーして下さい。





SymConvert Options	n de la companya de l
Debug	 オブション 出力ファイル 依存ファイル コマンド(©): C*Program Files¥Aone¥DEF¥GCsymconv.exe オブション(②): 「\$(CONFIGDIR)¥\$(PROJECTNAME)×" ブレースホルダ(P): コンフィグレーションディレクトリ ▼ 挿入(①)

①オプションに下記内容を設定する。"\$(CONFIGDIR)¥\$(PROJECTNAME).x"

(入力ファイル名)

②OKをクリックします。

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).x"

②**\$(PROJECTNAME)**の先頭に「¥」記号を入力して下さい。(手入力) ③オプションSWを使用する場合は両端にスペースを入れてください。(手入力)

追加事項(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。
- 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。
- 4) [-i] (省略可) 重複モジュール情報を削除する。
- 5) [-g] (省略可) スタティック変数をグローバル化する。(Ver1.20x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。(Ver1.40Bから)
- 7) [-f] (省略可) 使用インクルードファイルを (View に登録する。(Ver 1. 40 Bから)

5. 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。



6. 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。

[6-2]



[Archives] タグ デフォルト状態です。(変更の必要なし)

[6-3]

File vects.o starto inthandler.o hwinito sh7145.o	File Typ Other Other Other Other Other	e Location OVHee-WKP[TVKP[T7, OVHee-WKP[TVKP[T7, OVHee-WKP[TVKP[T7, OVHee-WKP[TVKP[T7, OVHee-WKP[TVKP[T7,	Bemove Disst/de Modify_ Move up
ommon opjions \$0960.JECTINAN	Extot	metrorec - Mag *100019100	Move down

[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.0
- 2 start.o
- 3 in than dle no
- ④ hwinit.o
- 5 SH7145.0

[6-4]



[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。

[6-5]

Section groups:			
Start address	Group name	- 1	Bog thorb"
0.00000400	Mechi Mechi	- 10	Remove
	ind.	- 13	
	fini		Move yp
8	.pot		Manual Annual
8	rodata		Hove goen
8	.eh_frame_hdr		Modify_
	.eh_frame		
	jer .		
8	Aore	1	
	data	. E .	
Section group map	ones		
Group name	Memory for group reserved at		
			Memory_

 [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「0x1000」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「0x500」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。

[6-6]



③Address: を「0x500」に変更します。
 ④OKをクリックします。



hives bout Ou	tout List Sections c	ther Comm	und file
faction mount	and the second by	reser 1 comm	and the l
Start address	Group name	~	Add group.
	igot rodata		Bemove
	.eh.frame.jkdr .eh.frame		Move up
	jer tors		Move gown
0xFFFFE000	data		Modify_
0.FFFFFFF0	.goo_exo .bee .stack		
<	-	× ×	
legtion group maps	ines		
Group name	Memory for group reserved	t at	
			Mgmory_
			Eport.

[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。

[6-8]



③Start address: 【Fixed address】に選択します。
 ④Address: .bss セクションの先頭アドレスを指定します。
 【OxFFFFE000】

ソースブレーク使用の指定の場合でも内臓RAMは使用 しません。

⑤Advanced をクリックします。

[6-9]



⑥「Reserve memory for section mapping」をチェックする。
⑦Map address: 【Label】にする。
⑧Label: 【_mdata】

⑨OKをクリックします。

初期値を ROM->RAM にコピーする為の重要 な設定です。

[6-10]



[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。

[6-11]

jame	QK
Start address:	Cancel
Fixed address	Advanced.
Address	CHERRICAL STREET
DEFFFFFFC	
Sontjenitis	
🖬 "stack	lgsert section
	Incert label.
	Modify.
	Bemove
	Move up
	Move down

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

7.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「SH7145.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登録します。



① inthandler.c を選択します。

②void INT_MTU0_TGIA0(void) { Timer0(); }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。)

8. 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。

[8-1]



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。 デバッグ途中(スタックポインタが変化後)で、「start.asm」からプログラムを走らせたくなった 場合、このソース行があればハードリセットしなくてもスタックポインタは初期化されますので、 記述しておいても良いでしょう。

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。

9. 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/GCsymonv 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。

[9-2]



↑のように「0 Errors, 0 Warnings」になれば成功です。

10.DEFでの確認

[10-1]

A-one H-DEbugger & Flashwriter Ver6.00A A-one AH7000 Ve	г нирр 🔲 🗖 🔀
ファイルビン テータ(1) 美行(1) フレーク(1) 割り込み(1) オフジョン(1) ヘルン(CONSTRUCT OF CONSTRUCT DOT CONSTRUCT DO NO.
TOTAL STOCK COULDED A STOCK CPOSH/145F0HODD	0000910P 0x00000400B010x00003C6B PcMemor .
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU	konWart罗名GL道友
* 00000500: 35 mov.1 stack.r15	<u>±</u>
35	
37 ! Call hw_initialise	
00000504: 38 mov.1 hw_initialise.rl	DEF レジスタ SH-2
00000004: 33 JSF @FI	PC 000000502 PR 000000C30
40 hop	R0 000000A02 R8 000000000
41 42 Lload data section from ROM to	R1 00000004 R9 000041039
42 :10a0 data section from to	R2 0FFFF4010 R10 000000000
44 Hif ROMSTART	R3 0FFFF8A52 R11 000000000
45 Linitialise sections	R4 000000001 R12 000850014
• 00000508: 46 mov.l edata.rl !	R5 0FFFF4000 R13 04F6C6400
 0000050A: 47 mov.1 mdata.r2 ! 	R6 000000000 R14 0FFFFFFB4
+ 0000050C: 48 mov.l data.r0	R7 0FFFF4804 SP 0FFFFFFFC
• 0000050E: 49 cmp/eq r0,r1	MACH OFFFFFFFF MACL 000000000
• 00000512: 50 bt start_1	GBB 000000011 YBR 00000000
• 00000512: 51 nop	SB 00F1 3210 T FRIEND
52 start_1:	
100000514 - 52 march Br2 r2 Last from	<u>•</u>
CLR CLR CLR CLR	CLR
000032E0 StringROM ・メモリダンフォ char	行回数 1
000032E01 StripeR0M: 68.65.6C.6C.6E.20.77	ID 29 Ibello world(ROM)
00002201 0011101000 00 00 00 00 00 00 00 00 00	
000000000000000000000000000000000000000	*
Go Break RstMon Reset Win Reg Watch Sym	Step DI EI IntFlg PUTCH
000032F0: 00 00 00 FF FF E0 1C	0 • 🔺
	~
Start	Esc Stop

①500H番地にスタックポインタの設定コードがあります。
 ②SP値が「**0xFFFFFFC**」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。