

【H-debugger 対応】 TOPPERS/OSEK カーネル

アプリケーションノート

H8S/2600シリーズ (BOOT-PBC) H8S/2612F KPIT-GNU [Hew] 版

2 0 0 8 / 1 0 / 0 2

Rev1. 70 (2008/10/02)



TOPPERS/OSEK Kernel

Toyohashi Open Platform for Embedded Real-Time Systems/ OSEK Kernel Copyright (C) 2000-2003 by Embedded and Real-Time Systems Laboratory Toyohashi Univ. of Technology, JAPAN

Copyright (C) 2004 by Embedded and Real-Time Systems Laboratory

Graduate School of Information Science, Nagoya Univ., JAPAN

Copyright (C) 2004-2006 by Witz Corporation, JAPAN

Copyright (C) 2008 by A-one Corporation, JAPAN

上記著作権者は、以下の(1)~(4)の条件か、Free Software Foundation によって公表され ている GNU General Public License の Version 2 に記述されている条件を満たす場合に限 り、本ソフトウェア(本ソフトウェアを改変したものを含む.以下同じ)を使用・複製・改変・ 再配布(以下,利用と呼ぶ)することを無償で許諾する.

- (1) 本ソフトウェアをソースコードの形で利用する場合には、上記の著作 権表示、この利用条件および下記の無保証規定が、そのままの形でソー スコード中に含まれていること.
- (2) 本ソフトウェアを、ライブラリ形式など、他のソフトウェア開発に使用できる形で再配布する場合には、再配布に伴うドキュメント(利用者マニュアルなど)に、上記の著作権表示、この利用条件および下記の無保証規定を掲載すること.
- (3) 本ソフトウェアを、機器に組み込むなど、他のソフトウェア開発に使 用できない形で再配布する場合には、次のいずれかの条件を満たすこ と.
 - (a) 再配布に伴うドキュメント(利用者マニュアルなど)に、上記の著 作権表示、この利用条件および下記の無保証規定を掲載すること.
 - (b) 再配布の形態を,別に定める方法によって,TOPPERS プロジェクトに 報告すること.
- (4) 本ソフトウェアの利用により直接的または間接的に生じるいかなる損 害からも、上記著作権者および TOPPERS プロジェクトを免責すること.

本ソフトウェアは,無保証で提供されているものである.上記著作権者および **TOPPERS** プロ ジェクトは,本ソフトウェアに関して,その適用可能性も含めて,いかなる保証も行わない.ま た,本ソフトウェアの利用により直接的または間接的に生じたいかなる損害に関しても,その責 任を負わない.



目	次
---	---

第1章 概	要	5
1 - 1.	はじめに	5
1 - 2.	関連文書	5
1-3.	開発環境	5
1 - 4.	製品梱包内容	5
1 - 5.	サポート	5
第2章 T	OPPERS/OSEK カーネルの開発階層	6
第3章 サ	ンプルアプリケーションの概要(cat261)	7
3-1.	サンプルアプリケーションの構成	7
3 - 2.	動作フロー(ゼネラル)	8
3-3.	H-debugger 対応に必要なポーティング1	5
3 - 4.	H-debugger でのプロファイル表示(DEF 7.00B 以上)1	8
第4章 ハ	ード構成およびシステム構成1	9
第4章 ハ 4-1.	ード構成およびシステム構成1 ハード構成1	9 9
第4章 ハ 4-1. 4-2.	ード構成およびシステム構成	9 9 0
第4章 ハ 4-1. 4-2. 4-2	ード構成およびシステム構成	9 9 0 0
第4章 ハ 4-1. 4-2. 4-2 4-2	 ード構成およびシステム構成 ハード構成 システム構成 -1.システムブロック図(サンプル) -2. CAT261のプログラムメモリ MAP 	9 9 0 1
第4章 ハ 4-1. 4-2. 4-2 4-2 4-2	 ード構成およびシステム構成 ハード構成 システム構成 2 -1.システムブロック図(サンプル) 2 -2. CAT261のプログラムメモリ MAP 2 -3. プログラムサイズの詳細 MAP 	9 9 0 1 1
第4章 ハ 4-1. 4-2. 4-2 4-2 4-2 4-2	 ード構成およびシステム構成 ハード構成 システム構成 2 -1.システムブロック図(サンプル) 2 -2. CAT261のプログラムメモリ MAP 2 -3. プログラムサイズの詳細 MAP 2 -4. 評価ボード(AHE261)のディップ SW の設定と I/O マップ表 	9 9 0 1 1 2
第4章 ハ 4-1. 4-2. 4-2 4-2 4-2 4-2 第5章 サ	 ード構成およびシステム構成 ハード構成 システム構成 2 -1.システムブロック図(サンプル) 2 -2. CAT261のプログラムメモリ MAP 2 -3.プログラムサイズの詳細 MAP 2 -4.評価ボード(AHE261)のディップ SW の設定と I/O マップ表 2 ンプルアプリケーションの準備 	9 9 0 1 1 2 4
第4章 ハ 4-1. 4-2. 4-2 4-2 4-2 第5章 サ 5-1.	 ード構成およびシステム構成 ハード構成 システム構成 システムブロック図 (サンプル) 2 -1.システムブロック図 (サンプル) 2 -2.CAT261 のプログラムメモリ MAP 2 -3.プログラムサイズの詳細 MAP 2 -4.評価ボード(AHE261)のディップ SW の設定と I/O マップ表 ンプルアプリケーションの準備 2 Hew での準備 (ルネサス製) 2 	9 0 1 1 2 4 4
第4章 ハ 4-1. 4-2. 4-2 4-2 4-2 第5章 サ 5-1. 5-2.	 トード構成およびシステム構成 ハード構成 システム構成 システム構成 2 -1. システムブロック図 (サンプル) 2 -2. CAT261 のプログラムメモリ MAP 2 -3. プログラムサイズの詳細 MAP 2 -4. 評価ボード(AHE261)のディップ SW の設定と I/O マップ表 2 ・ンプルアプリケーションの準備 2 Hew での準備 (ルネサス製) 2 H-debugger(DEF.exe)での準備 (Aone 製) 2 	9 0 1 1 2 4 9
第4章 ハ 4-1. 4-2. 4-2 4-2 4-2 4-2 第5章 サ 5-1. 5-2. 5-3.	 ード構成およびシステム構成 ハード構成 システム構成 システム構成 2 -1.システムブロック図 (サンプル) 2 -2. CAT261 のプログラムメモリ MAP 2 -3. プログラムサイズの詳細 MAP 2 -4. 評価ボード(AHE261)のディップ SW の設定と I/O マップ表 2 ンプルアプリケーションの準備 2 Hew での準備 (ルネサス製) 2 H-debugger(DEF.exe)での準備 (Aone 製) 2 サンプルアプリケーションを走らせる前の準備 3 	9 0 1 1 2 4 9 2
第4章 ハ 4-1. 4-2. 4-2 4-2 4-2 4-2 第5章 サ 5-1. 5-2. 5-3. 5-4.	 ード構成およびシステム構成 ハード構成 システム構成 システム構成 2 1. システムブロック図 (サンプル) 2 2. CAT261 のプログラムメモリ MAP 2 3. プログラムサイズの詳細 MAP 2 4. 評価ボード(AHE261)のディップ SW の設定と I/O マップ表 2 ケンプルアプリケーションの準備 2 Hew での準備 (ルネサス製) 2 サンプルアプリケーションを走らせる前の準備 3 サンプルアプリケーションを走らせます。 	9 0 0 1 1 2 4 9 2 2



6-1. プロジェクトタイプの作成	34
6-2.新規プロジェクトを登録します。	36
第7章 備考	44
7-1. おわりに	44



第1章 概要

1-1.はじめに

本アプリケーションノートは、TOPPERS/OSEK カーネルをもとに「H-debugger」と 「CAT261」用にポーティングしたサンプルソフトです。

サンプルソフトの他アプリケーションへの利用/変更に関しての制限は一切ありませんので自 由にお使い下さい。ただし、このサンプルソフトの不具合により発生した損害に対しての責任、 及び、修正の義務は負いません。また、このサンプルソフトに関する質問の回答義務も負えませ んが、メールでのお問い合わせに対しては、弊社責任の範囲内でしたら出来るだけ御答えするよ うに努めます。Mail: cat-i@aone.co.jp

1-2. 関連文書

本アプリケーションは、下記ドキュメントを参考にして作成しました。

1)	TOPPERS/03	SEK カーネル外部仕様書 ---------	株式会社ヴィッツ製
2)	TOPPERS/03	SEK カーネル SG 取扱説明書	株式会社ヴィッツ製
3)	TOPPERS/03	SEK カーネルアプリケーションノート ----	株式会社ヴィッツ製
4)	OSEK/VDX	Operating System Ver2.2.1	OSEK/VDX 仕様
5)	OSEK/VDX	Binding Specification Ver1.4.2	OSEK/VDX 仕様
6)	OSEK/VDX	OIL Specification Ver2.5	OSEK/VDX 仕様
*OSEK/VDX が公開している仕様書は、http://www.osek-vdx.org/ よりダウンロードにより入			
手し	て下さい。		

1-3. 開発環境

1)	Hew Version 4.04.01.001	Renesas 製
2)	KPIT GNUH8[ELF] Toolchain v0801	-KPIT 製
にて	て作成しましたので各自用意をして下さい。	

1-4. 製品梱包内容

- 1) サンプルソフト用 CD (本書 PDF ファイルも含む) ---- 1 枚
 - 🔼 ・本製品の価格体系は、CD 配布の実費のみになっております。
 - ・AHE-KIT(CAT261 /AHE261)と、サンプル確認用ケーブル(AHC-SAMP1)は、
 本製品に含まれていません。(別売り)

1-5. サポート

TOPPERS/OSEK に関するサポートが必要な場合は、TOPPERS ホームページの「関連製品」 「サポート・サービス」の項をご覧になり、御利用下さい。 http://www.toppers.jp/ <-- TOPPERS ホームページ



¥toppers_osek	
¥config	// 機種依存階層
¥h8s-kpitgnu-2612	// 開発環境分類
+CPU 依存部	
¥cat261	// システム依存部
¥h8s2612st	// システム依存部
¥include	// インクルードファイル階層
¥kernel	// カーネル共通部階層
¥sample	// サンプル階層(未使用)
¥sg	// システムジェネレータ階層
¥impl_oil	// OIL 記述の実装定義部階層
¥syslib	// システムライブラリ階層
¥h8s-kpitgnu-2612	// 開発環境分類
+CPU 依存部	
¥cat261	// システム依存部
¥h8s2612st	// システム依存部
¥tools	// Hew 管理階層
¥h8s-kpitgnu-2612	// 開発環境分類
+ワークスペース	
¥cat261	// プロジェクト1
¥appsrc	// サンプルソース [*.c]
¥debug	// オブジェクト [*.abs]
¥h8s2612st	// プロジェクト2
¥appsrc	// サンプルソース [*.c]
¥debug	// オブジェクト [*.x]
¥Project	// 新規プロジェクト作成用
	<i>II テンプレート</i>

第2章 TOPPERS/OSEK カーネルの開発階層

本サンプルソフトは、2本のプロジェクトを用意しています。

1) プロジェクト1 ¥cat261 実行オブジェクト【cat261.x/cat261.mot】 本アプリケーションノートのサンプルプロジェクト

2) プロジェクト2 ¥h8s2612st 実行オブジェクト【h8s2612st.x/h8s2612st.mot】
 【株式会社ヴィッツ】殿作成のサンプルを H8S/2612 用にポーティングしたプロジェクト

・TOPPERS/OSEK カーネルアプリケーションノート【株式会社ヴィッツ製】

青字部分は、TOPPERS/OSEK 正式リリース(オープンソース)になります。その他は弊社で 改造および作成をしました。



第3章 サンプルアプリケーションの概要(cat261)

3-1. サンプルアプリケーションの構成

サンプルアプリケーションは、下記の構成にて作成しました。

1) **6**個のタスク

- ① MainTask ------ 各 Task の起動処理
- ② Task1 ------ LCD 表示タスク(50ms 毎のサイクル処理)
- ③ Task2 ------ LED 出力タスク(20ms 毎のサイクル処理)
- ④ Task3 ------ ブザー出力タスク(イベントによる起動)
- ⑤ Task4 ------ シリアル(SIO)送受信タスク(200ms 毎のサイクル処理)
- ⑥ Task5 ------ メロディタスク(1ms 毎のサイクル処理)
- 2) 8個の割込み処理
 - ① システムタイマー TPU0(TGR0A)1ms 割込み SysTimerInt() 動作確認用タイマー TPU1(TGR1A)1 ショット CounterInt() ③ SCI0 受信エラー ERI_0 ErrHwSerialInt0() ④ SCI0 受信 RXI_0 RxHwSerialInt0() ⑤ SCI0 送信終了 TEI_0 TxHwSerialInt0() ⑥ SCI1 受信エラー ERI_1 ErrHwSerialInt1() ⑦ SCI1 受信 RXI_1 RxHwSerialInt1() ⑧ SCI1 送信終了 TxHwSerialInt1() TEI_1
- 3) 6個のイベント

MainEvt/T1Evt/T2Evt/T3Evt/T4Evt/T5Evt

4) 5個のアラーム

MainCycArm/SetEvt1Arm/SetEvt2Arm/SetEvt4Arm/SetEvt5Arm

5) 2個のコールバック処理

```
CallBackArm(TimerCallBack コールバック利用のソフトタイマ)/SampleArm
```

6) 1個のウォッチドッグタイマー

WatchDogClear

7) スタートアップフックルーチン

ErrorHook

8) シャットダウンフックルーチン

ShutdownHook

9) エラーフックルーチン

ErrorHook

10)1個のアプリケーションモード

AppMode1



3-2. 動作フロー (ゼネラル)





現在のアクティブ Task と自 Task の関係で、LCD デバイスを使用して良いか判断し、 かつ、処理ループ数をカウント表示する。(Task1:10000~19999)

② AllLcdDisp()

内部メモリ「LcdBuf[LCDY][LCDX]」に登録された表示データを LCD デバイスに表示する。

③ AppSuspend()



現在のアクティブ Task と自 Task の関係で、LCD デバイスを使用して良いか判断し、 かつ、処理ループ数をカウント表示する。(Task2:20000~29999)

③ AppSuspend()





現在のアクティブ Task と自 Task の関係で、LCD デバイスを使用して良いか判断し、 かつ、処理ループ数をカウント表示する。(Task3:30000~39999)

② Buzzer()

TPU1 から、Buzzer に対して指定周波数を出力する。

③ AppSuspend()







現在のアクティブ Task と自 Task の関係で、LCD デバイスを使用して良いか判断し、 かつ、処理ループ数をカウント表示する。(Task4:40000~49999)

 ② INT8 *txt = {"AHE-ToppersCD1 for TOPPERS/OSEK by Aone"}を各 Ch から 8 文字づつ リング方式により送信し、受信した文字を LCD に表示する。

TxD1 --> RxD0 で受信し、LCD の横8 文字目から8 文字表示する。

TxD0 --> RxD1 で受信し、LCD の横1 文字目から 8 文字表示する。

③ AppSuspend()



現在のアクティブ Task と自 Task の関係で、LCD デバイスを使用して良いか判断し、 かつ、処理ループ数をカウント表示する。(Task5:50000~59999)

② AutoMelody()

Buzzer より、「ネコフンジャッタ」を演奏する。

③ AppSuspend()



3-3. H-debugger 対応に必要なポーティング

CAT261(H8S/2612)は、BOOT ポート(SCI2)使用による PBC(PC ブレークコントローラ)を 利用したデバッグモードになります。

PBC/トレースを利用するにおいて、割込みモード2と**PBC**の割込みプライオリティを7にして、他の割込みプライオリティを6以下にする必要があります。

1) ポーティング手続き1

.¥config¥h8s-kpitgnu-2612¥cat261¥sys_support.s				
hardware init hook:				
	ldc.b	#0x80, ccr		
	ldc.b	#0x06, exr		
	mov.b	#0x21, r0l		
	mov.b	r0l, @SYSCR		
	; H-debı	ıgger 用 リセット遅延	の為、 20msWait	
	; Target	CAT261-20MHz		
wait20m	s:			
	mov.w	#20,r0		
waitrst:				
	bsr	_SOFT1MS		
	dec.w	#1,r0		
	bne	waitrst		
	rts			
;/*SOFT	IMS() 1r	ms ソフトタイマー (20.0000M	1Hz) Non Wait */	
_SOFT11	MS:			
	push.w	r0	;	
	mov.w	#5000,r0	;/* 5000*4=20000cyc	*/
wait:			;	
	dec.w	#1,r0	;/* 1 clock	*/
	bne	wait:16	;/* 3 clock	*/
	pop.w	r0	;	
	rts			



<理由>

CAT261(H8S2612)の場合、H-debugger からのリセット信号が RST-IC 経由で CPU につながっている為、RESET 信号の立ち上がりに遅延が発生します。

BOOT-PBC 仕様でのデバッガ処理は、RESET 立ち上げ後、20ms 毎に最大 20 回まで NMI が 認識するまでターゲットに対して割り込を要求します。

その 20ms 分 CPU の走行を停止させる為、上記プログラムのようにソフトタイマーを入れます。 なお、「DEF-CPU 設定にて 200ms 遅延回路を使用する」に設定した場合は、RESET 立ち上 げしてから、200ms 経過後 NMI 要求を開始します。

2) ポーティング手続き2

.¥config¥h8s-kpitgnu-2612¥cat261¥sys_config.c			
***************************************	:/		
*	:/		
***************************************	:/		
* PBC = 7 その他 6 以下とする debug 対応	*/		
* PBC 以外のプライオリティを 0 にする。	*/		
* < PBC	*/		
	_config.c ************************************		

<理由>

ここで、PBC以外のプライオリティを全てゼロ「0」にする。割込みイネーブルの場合、exr レジスタをゼロ「0」にしますので、不用割り込みを発生させないための処置です。

使用する割込み要素は、後ほど各内部レジスタの初期化処理でプライオリティと共に設定されま す。



3)ポーティング手続き3

.¥config¥h8s-kpitgnu-2612¥cpu_support.s ;H 7-->6 に変更する。 H-debugger 対応の為 .equ EXR_MSK, 0x06

<理由>

OS カーネル内でもブレークおよびトレース実行を可能にする為、割込みディセーブルをする為の exr レジスタマスクを「6」にします。

その他、[exr]設定部分を全てを見直す。



3-4. H-debugger でのプロファイル表示(DEF 7.00B 以上)

ターゲット側の RAM を使用して、各 Task のプロファイル表示をします。

1) ポーティング手続き

.¥syslib¥h8s-kpitgnu-2612¥sys_timer.h	
#define USE_PROFILE	
#ifdef USE_PROFILE	
#define PROFILESIZE	1024
#define PROFILEPRETASK	1
#endif	

- 「USE_PROFILE」を有効にすると、プロファイル表示が可能になります。不要になった場合は、コメントアウトして下さい。
- ②「PROFILESIZE」は、ターゲット側で確保する RAM のバイト数になります。

アプリケーションに応じて調整して下さい。

③「USE_PROFILE」を有効にする事により、下記変数が確保されます。

UINT8 TaskProFlg;

UINT16 TaskProFileIdx;

 ④システムタイマー割り込みのタイミングで現 Running 中の TaskID を「_TaskProFile」に 順次記憶していきます。(リングバッファ)

⑤サンプリング場所は、「プレタスクフックルーチン」と「ISR(SysTimerInt)」の2箇所です。 ⑥プロファイル表示させたい場合は、

DEF メニュー<データ>-<プロファイル表示>をクリックして下さい。



[「]更新」PBをクリックしますと最新データを表示します。

「周期更新」にチェックしますと、常時最新状態をターゲットからオンザフライ機能により、 データを収集し、Task 状態を表示します。



第4章 ハード構成およびシステム構成

4-1. ハード構成

この解説書を進めるにあたり、下記ハード構成の準備をお願いします。



[接続例]





4-2. システム構成

サンプルのシステムブロック図およびメモリマップと I/O 表を記述します。

[評価ボード回路図] AHE261_man_sch.pdf

[CAT261回路図] CAT261-C3P.pdf / CAT261 取説.pdf

4-2-1. システムブロック図 (サンプル)





4-2-2. CAT261 のプログラムメモリ MAP

CPU 内部 H8S/2612 フラッシュ ROM 128Kbyte	0x0ベクタテーブル① 0x200デバッグモニタ開始番地 0x800アプリケーションプログラム開始番地②
	0x1FFFFFROM 最終
CPU 内部 RAM1	0xFFE000 アプリケーション使用 RAM の開始番地③
約 4Kbyte	0xFFEFBE 初期スタックポインタ位置(0xFFEFBE)④
CPU 内部 RAM2 64byte	0xFFFFCO デバッグモニタが使用する RAM 開始番地
	」 0xFFFFFFRAM 最終(デバッガが使用する)

<セクション名>

① ベクタテーブル	.vects
②アプリケーションプログラムの開始番地	.text,.init,.rodataetc
③アプリケーション使用 RAM の開始番地	.data,.bssetc
④初期スタックポインタ位置	.stack

4-2-3. プログラムサイズの詳細 MAP

開始番地	サイズ	分類
0x200	0x600	デバッガモニタ(ファーム)
0x800	0x186C	アプリケーション(mainetc)
0x206C	0x1C04	割込みハンドラ、内部 I/O 関係の関数
0x3C70	0x2D8C	TOPPERS/OSEK カーネル
0x69FC	0x43F	C ライブラリ、ROM テーブル、etc
0x6E3B		最終アドレス



SW11 の設定		ON			OFF			
SW11-1		シリア TXD0,RXD	シリアル I/O の TXD0,RXD1 を折り返し			シリアル I/O の TXD0,RXD1 間オープン		
SW11-2		シリア TXD1,RXD	シリアル I/O の TXD1,RXD0 を折り返し		シリアル I/O の TXD1,RXD0 間オープン			
LCD関係								
ポートアト゛レス	ポートシンボ	ル ピン番号	ヒ゜ンシンホ゛ル	方向	句		信号名	
		CN1-7B	PD0	出ス	<u>ታ</u>	D0	LCD-module	
		1-7A	PD1	出フ	<u>ታ</u>	D1		
OvffffOc	סחחס	1-6B	PD2	出力		D2		
OXIIIIOC	IDDR	1-6A	PD3	出フ	<u>ታ</u>	D3		
		1-5B	PD4	出フ	<u></u>	D4		
		1-5A	PD5	出フ	<u>ታ</u>	D5		
		1-4B	PD6	出フ	<u>ታ</u>	D6		
		1-4A	PD7	出フ	力	D7		
		CN 2 -9A	PC6	出フ	<u>ታ</u>	RS	LCD-module	
		-9B	PC7	出フ	<u>ታ</u>	E	LCD-module	
0xffff0b	PCDR					未使	用 	
	I ODR					未使	用	
				未使用		用		
						未使	用	
						未使	用	
		-14A	P07			未使	用	

4-2-4. 評価ボード(AHE261)のディップ SW の設定と I/O マップ表

LED関係							
ポートアドレス	ポートシンボル	レ ピン番号 ピンジンボル 方向 信号名			信号名		
		CN1-14A	P10	出力	LED1		
0xffff00		1-14B	P11	出力	LED2		
	P1DR	1-15A	P12	出力	LED3		
		1-15B	P13	出力	LED4		
		1-16A	P14	出力	LED5		
		1-16B	P15	出力	LED6		
		1-17A	P16	出力	LED7		
		1-17B	P17	出力	LED8		

ブザー関係(タイマ/TPU3 TIOCA3 アウトプットコンペア出力)						
ポートアドレス	ポートシンボル	ピン番号	ヒ゜ンシンホ゛ル	仕様		
0xfffe88	TGRA_3	CN2-10A	PB0/TIOCA3	PWMモード パルス出力		



押しボタンスイッチ関係						
ポートアト・レス ポートシンボル ピン番号 ピンシンボル 方向 信号名						
0xffffb9	POATA	CN1-3B	PA0	入力	SW12 PB[PA0]	
0xffffbb	PORTC	2-7A	PC2	入力	SW13 PB[PC2]	
		2-8B	PC5	入力	SW14 PB[PC5]	
0xffffbe	POATF	2-5B	PF7	入力	SW15 PB[PF7]	
[[[

トグルスイッチ関係						
ポートアト・レス	ポートシンボル	ピン番号	ヒ゜ンシンホ゛ル	方向	信号名	
		CN1-13B	P40	入力	SW16 SW[P40]	
0xffffb3 I		1-13A	1-13A P41 入力	入力	SW17 SW[P41]	
	PORT 4	1-12B	P42	入力	SW18 SW[P42]	
		1-12A	P43	入力	SW19 SW[P43]	
		1-11B	P44	入力	SW20 SW[P44]	
		1-11A	P45	入力	SW21 SW[P45]	
		1-10B	P46	入力	SW22 SW[P46]	
		1-10A	P47	入力	SW23 SW[P47]	







第5章 サンプルアプリケーションの準備

5-1. Hew での準備(ルネサス製)

- 1) TOPPERS のホームページより、OSEK カーネル最新リリースをダウンロードして下さい。 URL:http://www.toppers.jp/osek-os.html
- 2) ダウンロードした「osek_os-x.x.lzh」を、適当なディレクトリに置き解凍して下さい。
- 3) サンプルアプリケーション用 CD の「.¥toppers_osek_Kpit_2612」の指定ファイルを、 OSEK カーネルにコピーします。

<CD 側> <DL した OSEK 側> .¥toppers_osek_Kpit_2612¥config¥h8s-kpitgnu-2612 --> .¥toppers_osek¥config .¥toppers_osek_Kpit_2612¥syslib¥h8s-kpitgnu-2612 --> .¥toppers_osek¥syslib .¥toppers_osek_Kpit_2612¥tools¥h8s-kpitgnu-2612 --> .¥toppers_osek¥tools 上記ディレクトリ下の全ファイルを、OSEK カーネルにディレクトリごと全コピーして下 さい。

- 4) Hew を起動します。
 ・ Hew Version 4.04.01.001-----Renesas 製
 ・ KPIT GNUH8[ELF] Toolchain v0801 -----KPIT 製
- 5) Hew メニューの<ファイル>-<ワークスペースを開く>でワークスペースを開きます。
 - ・".¥toppers_osek¥tools¥h8s-kpitgnu-2612¥h8s-kpitgnu-2612.hws"を指定します。

・ディレクトリ情報が変わりますので、下記ウォーニングが表示されますが、気にせず 「はい」を指定して下さい。

ウォーニ	25	[5-1-1]
1	このワークスペースのディレクトリが移動されました。 旧ディレクトリ: Q.¥Hew4¥TOPPERS¥toppers_osek-1_1¥tools¥h8s-renesas 新ディレクトリ: C.¥HEW4¥toppers_osek-1_1¥tools¥h8s-renesas ワークスペースを聞きますか?	
	「ほいの」 いいえい キャンセル	



 アクティブプロジェクトが「cat261」になっていることを確認します。cat261になってい ない場合は、Hew メニューの<プロジェクト>-<アクティブプロジェクトに設定>で 「cat261」を指定して下さい。



7)現 KPIT 版では、ディレクトリの相対指定[..¥]が出来ない為、インクルードディレクトリの 変更が必要です。

Hew メニューの<ビルド>-<Compiler>をクリックして下さい。





[5-1-2-2] <Brows>指定が早いかと思 います。



Hew メニューの<ビルド>-<Assembler>をクリックして下さい。



[5-1-2-3]

全てのディレクトリ を現ワークスペース のディレクトリに変 更して下さい。

Hew メニューの<ビルド>-<Linker>をクリックして下さい。

Linker Options (Debug)	2 🛛	[5-1-2-4]
Archives Input Output List Sections Other Command	file	
C¥program files¥renesas¥hew¥tools¥kpit cummins¥gnush-el C¥program files¥renesas¥hew¥tools¥kpit cummins¥gnush-el	<u>A</u> dd Bemove	<input/> タグをクリックします。
<	Move yp Move dogn	
Archive (library) files:		
liboptc.a	Add_	
libecca	Remove	
	Move up	
	Mo <u>v</u> e down	
Search archives repeatedly for undefined references		
✓ Use Optimised Libraries		
Common options:		
"\$(CONFIGDIR) %\$(PROJECTNAME)×"oformat=srec -Map \$(CONFIGDIR) %\$(PROJECTNAME)×"oformat=srec -Map \$(CONFIGDIR) %\$(PROJECTNAME).map" -e_start -l "optc" -l "optm" -l "gc	xo" 💙	
Show option description	**>セル	



- アクティブプロジェクトが「h8s2612st」に変更して、7)項と同じく<Compiler>と
 <Assembler>と<Linker>のディレクトリを変更して下さい。
- <u>▲</u> 新規プロジェクトの作成時に必要です。
- 9)アクティブプロジェクトが「Project」に変更して、7)項と同じく<Compiler>と
 <Assembler>と<Linker>のディレクトリを変更して下さい。



- 10) アクティブプロジェクトを「cat261」に戻します
- 11) Hew メニューの<ビルド>-<すべてをビルド>をクリックして下さい。



12) ビルド結果



Build Finished「0 Errors 0 Warnings」になれば成功です。



5-2. H-debugger(DEF.exe)での準備(Aone 製)

- 1) H-debugger コントロールソフト「DEF.exe」を起動します。
- 2) DEFメニューの<オプション>-<環境設定>の「本体機種設定」が正しい機種と COM ポートの選択をされているか確認して下さい。
- 3) DEF メニュー<オプション>-<CPU 設定>をクリックします。

I DEF CPU 設定 X	[5-2-1]
CPUタイプ CPUシリーズ名 H8S/2612F	CPU シリーズ名 : H8S/2612F
CPU物作モード Advanced	
水晶発振子クロック(MHz 20.0000 I	クロック(MHz) : 20.0000
周波数通倍率(PLL回路) x1 ▼	
モニタワークエリアの場所 固定番地:adr(0xffffc0)siz -	
 デバッグモード(H8/Tiny) エミュレーション(分岐トレース,(低消費電力可能) ローザーモード(起動時RAMが書き換わらない) 	- モニタワークエリアの場所:固定番地
ブートモード設定 ブート時ポーレート 19200 <u>・</u>	
ユーザブログラムモード時のイレース設定 ダウンロード時のイレース設定 (* オールイレース * セレクトイレース 消去ブロック 0 ~ (10道) 「 ダウンロード時に内臓RAMの退産/復帰をしない。	
アレイ 及び着にのする情報(第ペクエック)にで有効 アリセット運動防止タイマ200msを使用しない。 リセット運動防止タイマ200msを使用しない。 リセットペクタを作取しない。(4800時用) 日本本体からのリセット出力を使用しない。(4+UDD 命令による例外割込みをデバッガから通知する。	リセット遅延防止タイマ使用しない:チェック
外部RAM時のBSC設定スクリプト指定 「許可 参照」	
ターゲット間の通信仕様	
1382400 BPSで同期クロックの通信をします。	 最低上記4項目を設定後、「設定」にクリック
詳細情報 設定	
	◎ しより。

4) DEF メニュー<ファイル>-<アブソリュートファイル設定>をクリックします。



[5-2-2]

左図のように 3 箇所にチェックしま す。



5) DEF 画面、左下隅の「Start」をクリックします。(ターゲット側の電源は ON の事)

■ A-one H-DEbugger & Flashwriter Ver6.80D	[5-2-3]
アドルビーテーター Alter アレークロー まりとかゆ オフラコンロー ヘルフィロー 調査課題 割込 PC	
C Src C Mix C Ass	
	1
	-
	-
	¥
CQL) Char 💌	163
	2
	T
Go Break AstNon Reset Win Rez Watch Sym Trace S	Ster
Start Esc St	op

6)評価キット(AHE261/CAT261)に登録してあるプログラムによっては、下図の様な確認画面 が表示されますが無視して「OK」をクリックして下さい。

DEF N	2
(モニタワークエリア(固定番地側酸定)と現モニタ酸定が適合していません。 再ダウンロードして下さい。 [OK]でDEFの処理を継続します。
	OK キャンセル

[5-2-4]

モニタのワークエリアを固定番 地指定しましたので、スタック 方式でモニタが入っていた場合 この様な表示になります。

7) ターゲット側と正常な通信を確立しますと下図の様な画面になります。

A-one H-DEbugger & Flashy	writer Ver6.80D A-	one AH7	000 V 🗐 🗖	X	[5-2-5]
ファイル(E) データ(D) 実行(G) ブレ	ーク(B) 割り込みの オ	プション(ロ)	ヘルプ(円)		
停止 割込 6 PC 000000000		U H85/26	12F Adv		
00000800	· C Src	C Hix	• Asm		
00000800: 7407 00FF EFBE		nov.1	10xffefbe:32,s	±	
. 00000806: 5400 088C		jmp	80x88c:24	-	
0000080A: 5470		rts			
- 0000080C: 0130 6DF0		stn. I	er0-er3,8-sp		
• 00000810: 0120 6DF4		stn. I	er4-er6,8-sp	. 193	
• 00000814: 0120 6076		Idn. I	8spt,er4-er6	- 68	
- 00000818: 0130 6D/8		Idn. I	8sp+.erU-er3	- 333	
- 00000816: 0570 00000815: 0120 CDE0		rte	and and Bran	- 188	
00000012, 0130 60F0		of a l	erd-erd B-op	-	
00000828: 5500 1518		ier	80×1518:24	Ŧ	
(* BP1 00000000	VCLR VNon V	1	BP2 000000000	T	
	 (なし) 	•	char		
				±	
				Ŧ	
Go Brook Ret Non Reco	Uin Rog	Vatch	Sun Trac	T	
do Dreak Astaon Kese	Net Net	racen	078 11 ac	0	
Start			- Eas Ci	-	
otart			Esc St	up	



8) DEF メニュー<ファイル>-<ダウンロード>をクリックします。

.¥toopers_osek¥tools¥h8s-kpitgnu-2612¥tools¥cat261¥debug まで降ります。





[5-2-7] ダウンロード中は、このようなインジケータ表示します。

9) ダウンロードが成功しますと、下図のような DEF 画面になります。

(P± 数2.6 PC 000000000 「 MMK 「 View CPU HBS/2012F Advance 2000001 (ま)と200000000 ・・・ 10000100 (ま) start (P Src ∩ Mix C Ass start.src) 「 WerMant (2012) * 10000101 (ま) start (P Src ∩ Mix C Ass start.src) 「 WerMant (2012) * 10000101 (ま) start (P Src ∩ Mix C Ass start.src) 「 WerMant (2012) * 10000101 (ま) start (P Src ∩ Mix C Ass start.src) 「 WerMant (2012) * 10000101 (ま) start (P Src ∩ Mix C Ass start.src) 「 WerMant (2012) * 10000101 (ま) start (P Src ∩ Mix C Ass start.src) 「 WerMant (2012) * 10000102 (t) start (P Src ∩ Mix C Ass start.src) * 10000102 (t) start (P Src ∩ Mix C Ass start.src) * 10000102 (t) start (P Src ∩ Mix C Ass start.src) * 10000102 (t) start (P Src ∩ Mix C Ass start.src) * 10000102 (t) start (P Src ∩ Mix C Ass start.src) * 10000102 (t) start (P Src ∩ Mix C Ass start.src) * 10000102 (t) start (P Src ∩ Mix C Ass start.src) * 10000102 (t) start (P Src ∩ Mix C Ass start.src) * 10000102 (t) start (P Src ∩ Mix C Ass start.src) * 10000102 (t) start (P Src ∩ Mix Src) * 10000102 (t) start (P Src ∩ Mix Src) * 100000102 (t) start (P Src ∩ Mix Src) <th>I A-one H</th> <th>DEbuge</th> <th>er & Flashwriter Ver6.800 A-one AH7000 Ver6.80(2008-03-17) for H8S.</th> <th>[5-2-8</th>	I A-one H	DEbuge	er & Flashwriter Ver6.800 A-one AH7000 Ver6.80(2008-03-17) for H8S.	[5-2-8
0000000 ● tart ● Grc ● Mix ● Arm start.src ● NorWeinty (204) 0000000 ● tarburt ● Grc ● Mix ● Arm start.src ● NorWeinty (204) 0000000 ● tarburt ● Grc ● Mix ● Arm start.src ● NorWeinty (204) 0000000 ● tarburt ● Grc ● Mix ● Arm start.src ● NorWeinty (204) 0000000 ● tarburt ● Grc ● Mix ● Arm start.src ● NorWeinty (204) 00000000 ● tarburt ● Grc ● Mix ● Arm start.src ● NorWeinty (204) 00000000 ● tarburt ● Grc ● Mix ● Arm start.src ● NorWeinty (204) 00000000 ● tarburt ● Grc ● Mix ● Arm start.src ● NorWeinty (204) 00000000 ● tarburt ● Grc ● Mix ● Arm start.src ● NorWeinty (204) 000000000 ● tarburt ● Grc ● Mix ● Arm start.src ● NorWeinty (204) 000000000 ● tarburt ● Grc ● Mix ● Arm start.src ● NorWeinty (204) 00000000 ● tarburt ● Grc ● Mix ● Arm start.src ● NorWeinty (204) 00000000 ● tarburt ● Grc ● Mix ● Arm start.src ● NorWeinty (204) 00000000 ● tarburt ● Grc ● Mix ● Arm start.src ● NorWeinty (204) 00000000 ● tarburt ● Grc ● Mix ● Arm start.src ● NorWeinty (204) 00000000 ● tarburt ● Grc ● Mix ● Arm start.src ● NorWeinty (204) 00000000 ● tarburt ● Grc ● Mix ● Arm start.src ● NorWeinty (204) 00000000 ● tarburt ● Grc ●	停止 割	62 6 PC	200000000 T MM T View CPU H85/2512F Advance 200000 x1 #32/65500041 TOP 0x0000080	
10005001 1 55 1000 0 1 1 1000 の単体出し (0 でない場合) 10015001 1 5 - グットバードウェアに依存して必要な対称的と地理がある場合 70 1 1 5 hardware_init_book の単体出し (0 でない場合) 10 1 5 - グットバードウェアに依存して必要な対称的と地理がある場合 71 1 1 15 hardware_init_book という関助を用意する。 10005001 2 3 barchware_init_book / **********************************	00000800	start	G Src C Mix C Ass start.src V NorWait5(055)	
	00000800:	65	wow.1 &_stack_top. sp	
		88	*	
		67	1	
Source State		68	; hardware_init_hook の呼出し (0 でない場合)	
70 : 3 - ブックハードワミアに依存して必要な別類にと思想のある場合 71 : 13. hardware_init_book という関数を用意する. 72 : 00000102: 23 sov.1 Dardware_init_book, end : sov時令にてシンボルが000場合、2*1となる 00000102: 74 bes hardware_init_book,r 00000102: 75 Jer 78 hardware_init_book,r: 77 ************************************		63		
100000000: 20 sov.1 bardware_init_book_end : sov@中和にてシンボルが000場合、Z=1となる 10000000: 74 bee handware_init_book_end 10000000: 75 jsr @handware_init_book_r 10000000: 75 jsr @handware_init_book 10 handware_init_book_r 10000000: 82 sov.1 @_bes_start.end 10000000: 82 sov.1 @_bes_start.end 100000000: 82 sov.1 @_bes_end.end 100000000: 83 sob.end.end 100000000: 81 sob.end.end 100000000: 82 sov.1 @_bes_end.end 1000000000 100000000: 81 sob.end.end 100000000 100000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 1000000 1000000 10000000 1000000 1000000 1000000 10000000 10000000 1000000 1000000 1000000 10000000 1000000 1000000 10000000 1000000 10000000 1000000 1000000 10000000 10000000 10000000 1000000 10000000 10000000 1000000 10000000 10000000 10000000 1000000 10000000 10000000 1000000 1000000 10000000 10000000 1000000 1000000 1000000 1000000 1000000 1000000 1000000 10000000 1000000 1000000 10000000 1000000 1000000 1000000 1000000 10000000 10000000 1000000 10000000 10000000 1000000 1000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 1000000 10000000 1000000 10000000 10000000 1000000 1000000 1000000 10000000 10000000 10000000 1000000 10000000 10000000 1000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 100000000		70	: ターグットハードウェアに依存して必要な活動を危速力のる場合	
D0000000: 74 bea hardware_init_hook, erd : movmfwhlにてシンボルがのの場合、Z*1となる D0000000: 74 bea hardware_init_hook,r D0000000: 75 jsr Whardware_init_hook,r D0000000: 76 hardware_init_hook,r T7		72	; 14. naroware_init_nook < (// DBBX SHSM > 0.	
D0000000: 74 beq hardware_init_hook_r D0000000: 75 jsr Mardware_init_hook 76 hardware_init_hook 77 78 ; 79 ; 79 ; 79 ; 79 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ; 70 ;	. 001008081	73	nov.1 Deardware init book, eril : nov命令にてジンボル作のの場合 アロとなる	
B0000000: 25 Jsr Bhardware_init_hook 78 hardware_init_hook 77 77 77 78 79 79 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70 70	. 000008CC:	74	beg hardware init hook r	
78 hardware_init_hook_rt: 77 78 78 : 78 : 78 : 78 : 78 : 78 : 78 : 78 : 78 : 78 : 78 : 78 : 78 : 78 : 78 : 78 : 79 : 78 : 79 : 79 : 79 : 79 : 79 : 79 : 70000800: : 80 : 70000800: : 81 : 71 : 82 : 70000800: : 82 : 91 : 92 : 93 :	. 00000800:	75	jsr Bhardware_init_hook	
77 78 : 78 : bost2292 m 2/(8) € 2007 60 : bost2292 m 2/(8) € 2007 60 : mov.1 0_bos_start.er0 000008021 62 mov.1 0_bos_start.er0 00000825: 65 cmp.1 eft. 00000825: 67 mov.v r2. 6er0 00000825: 67 mov.v r2. 6er0 000008261: 68 mot.erf.inish :2 000008261: 69 mot.erf.erf.erf.erf.erf.erf.erf.erf.erf.erf		76	hardware_init_hook_r:	
78 : 79 : bost2クション(8)をクリア 80 : 00000800: 82 81 mov.1 84 bos_clear: 00000825: 85 85 cap.1 00000825: 85 83 mov.1 00000825: 85 84 bos_clear: 00000825: 85 85 cap.1 00000825: 85 85 cov.v.2. 00000825: 85 84 bos_clear finish 00000825: 85 83 : 2bytemiximus/Lif Schild 200000000 83 : 2bytemiximus/Lif Schild 2000000000 83 : 2bytemiximus/Lif Schild 2000000000 9100000000 ● CLR ● Non ▼ 92 000000000 ● CLR ● Non ▼ 16:31 ▼ 92 00000000 ● CLR ● Non ▼ 16:31 ▼ 92 ● Clear 93 ● CLR ● Non ▼ 94 ● Clear 95 ● Clear <t< td=""><td></td><td>77</td><td></td><td></td></t<>		77		
73 : bost2クション(8)をクリア 80 : 00008004: 81 80 : 00008002: 82 80 : 0000802: 82 83 : 0000802: 63 0000802: 63 0000802: 63 0000802: 63 0000802: 63 0000802: 63 0000802: 63 0000802: 63 0000802: 63 0000802: 63 : 2.000100000 : 2.0001000000 : 2.0001000000 : 0.0000000 : 0.0000000 : 0.0000000 : 0.0000000 : 0.0000000 : 0.0000000 : 0.0000000 : 0.0000000 : 0.0000000 : 0.0000000 : 0.0000000 : 0.0000000 :		78	3	
BOUDDERDC: 80 :		73	; besセクション(B)をクリア	
■ 00000000: Signature Signature Signature ■ 00000000: Signature Signature Signature ■ 00000000: Signature Signature Signature ■ 000000000 Signature Signature Signature ■ 0000000000 Signature Signature		80	1	
D00008E4: 83 mob.1 er2, er2 84 bos_clear: 00008E4: 85 mob.1 er2, er2 84 bos_clear: 00008E5: 85 mob.1 er2, er1 00008E5: 85 mob.1 er2, er1 00008E5: 85 mob.1 bos_clear_finish 00008E5: 88 mob.1 E2, er0 100008E5: 89 mod.1 E2, er0 100008E5: 89 mod.1 E2, er0 100008E5: 89 mod.1 E2, er0 100008E5: 99 mob.2000 100008E5: 99 mob.200 10000000 10008E5: 99 mob.20000 10000000 10008E5: 99 mob.2000000 10008E5: 99 mob.200000000 10008E5: 99 mob.20000000 10008E5: 99 mob.200000 10000000 10008E5: 99 mob.20000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 100000000	. 00000804:	81	mov. I 8_bss_start, er0	
DOUDDER: 00 DECEMPTION OF C. 072 Stand of C. 072 DOUDDEE: 00 DECEMPTION DOUDDEE: 00 DECEMPTION DOUDDEE: 00 edd.1 12, er0 : B/RtzウションはALIGN 2で定論していたので、 S3 edd.1 12, er0 : B/RtzウションはALIGN 2で定論していたので、 S4 edd.1 12, er0 : B/RtzウションはALIGN 2で定論していたので、 S4 edd.1 12, er0 : B/RtzウションはALIGN 2 edd. S4 edd.1 12, er0 : B/RtzウションはALIGN 2 edd. S4 edd.1 12, er0 : B/RtzウションはALIGN 2 edd.1 12, edd.	· DOTOLSDC:	82	nov.1 8_bts_end, eri	
D00008E5: 85 cmp.i er0, er1 D00008E5: 85 bis bos_clear_finish D00008E5: 87 mov.v r2, 8er0 D0008E5: 87 mov.v r2, 8er0 D0008E5: 88 edd.1 起, er0 : BVR世グションは4LIGN 2で定義していたので、 S3 edd.1 起, er1 > er0 までジャンプ S3 bra bos_clear : er1 > er0 までジャンプ S1 bos_clear_finish: S2 C BP1 00000000	000003E4:	9.4	sub-r erz, erz	
D00008E5: 88 bis bos_clear_finish D00008E5: 89 add.1 12, er0 : B/Rtt クションはALION 2で定論していたので、 S0 add.1 12, er0 : B/Rtt クションはALION 2で定論していたので、 S0 bra bos_clear : er1 > er0 までジャンプ S1 bos_clear_finish: S2 CR ▼ Non ▼	. DOTOTRES:	85	cost of the set	
D00008EE: 87 mov.v r2, 8er0 D0008EE: 88 edd.1 起, er0 : BVRE2グションは4104 2で定体していらので、 83 edd.1 起, er0 : 20xte器が規制とすらのX3問題ないな 00008F4: 80 bra bas_clear : er1 > er0 までジャンゴ 91 bos_clear_finish: 32 00000000 ・ CLR ・Non ・	. 000008E8;	18	bis bis clear finish	
D00008EE: 88 edd.1 起, er0 : B/Rセンジョンはおしび 2で定頭していたので、 S3 : 2 botem#3308EL するCA 3FE188 なし 5 D00008F4: 80 bre bos_clear : er1 > er0 までジャンゴ S1 bos_clear_finish: S2 C BP1 00000000 CLR ●Non ● CBP2 000000000 ● CCR ● Non ● R C BP2 000000000 ● CCR ● Non ● R C BP2 000000000 ● CCR ● Non ● R C BP2 000000000 ● CCR ● Non ● R C BP2 000000000 ● CCR ● Non ● R C BP2 000000000 ● CCR ● Non ● R C BP2 000000000 ● CCR ● Non ● R C BP2 000000000 ● CCR ● Non ● R C BP2 000000000 ● CCR ● Non ● R C BP2 000000000 ● CCR ● Non ● R C BP2 000000000 ● CCR ● Non ● R C BP2 000000000 ● CCR ● Non ● R C BP2 000000000 ● CCR ● DI ■ EI	. D00008EC:	87	acv.y r2, 6er0	
83 : 2byte#XXXMPL*5.07.307888421% 000008F4: 80 91 bos_clear_finish: 92 00000000 ● CLR ● Non ● CLR ● CLR ● Non ● CLR ● CLR ● CLR ● CLR ● DI ● CLR ● CLR ● CLR ● DI ● CLR ● CLR ● CLR ● DI ● CLR ● CLR ● CLR ● DI ● CLR ● CLR ● CLR ● CLR ● DI ● CLR	. 000008EE:	88	edd.1 \$2, er0 : B/RセクションはALIGN 2で定義しているので、	
D00000F4: 50 bre bos_clear ; erl > erl までジャンゴ S1 bos_clear_finish: S2 CLR ▼ Non ▼		89	: 2byte器初期化するのは問題ない。	
S1 bos_clear_finish: S2 ■ OLR Non Non CR2 00000000 ■ CLR Non Antipation Antipati	. 001008F4:	80	bra bss_clear ; er1 > er0 までジャンプ	
S2 © BPI D0000000 ▼CLR ▼Non ▼ CBP2 000000000 ▼CLR ▼Non ▼ No © CLR ▼Non ▼ CBP2 000000000 ▼CLR ▼Non ▼ No © CLR ▼Non ▼ CBP2 000000000 ▼ CLR ▼Non ▼ No © CLR ▼Non ▼ CBP2 000000000 ▼ CLR ▼Non ▼ No © CLR ▼Non ▼ CBP2 000000000 ▼ CLR ▼Non ▼ No © CLR ▼Non ▼ CBP2 000000000 ▼ CLR ▼Non ▼ No © CLR ▼Non ▼ CBP2 000000000 ▼ CLR ▼Non ▼ No © CLR ▼Non ▼ CBP2 000000000 ▼ CLR ▼Non ▼ No © CLR ▼Non ▼ CBP2 0000000000000 ▼ CLR ▼Non ▼ No © CLR ▼Non ▼ CBP2 0000000000 ▼ CLR ▼Non ▼ No © CLR ▼Non ▼ CBP2 000000000 ▼ CLR ▼Non ▼ No © CLR ▼Non ▼ CBP2 000000000 ▼ CLR ▼Non ▼ No © CLR ▼Non ▼ CBP2 000000000 ▼ CLR ▼ Non ▼ No © CLR ▼ Non ▼ CLR ▼ No © CLR ▼ NO		91	bss_clear_finish:	
© BP1 00000000	100	\$2		
Go Break Rst Mon Reset Win Reg Watch Sym Trace Step CTrac CStep DI EI	@ BP1 01000	0000	▼ CLR ▼ Non ▼ C 692 00000000 ▼ CLR ▼ Non ▼ //~	
Go Break Rst Mon Reset Win Reg Watch Sym Trace Step CTrac CStep DI EI				
Go Break RstMon Reset Win Reg Watch Sym Trace Step CTrac CStep DI EI	1521			
Go Break RstMon Reset Win Reg Watch Sym Trace Step CTrac CStep DI EI			X	
Go Break RstMon Reset Win Reg Watch Sym Trace Step CTrac CStep DI EI			1	
Go Break Rst Mon Reset Win Reg Watch Sym Trace Step CTrac CStep DI EI			*	
	Go Bre	eak Rst	Mon Reset Win Reg Watch Sym Trace Step CTrac CStep DI EI	
		and the party of the		
Class Law East Star				
LOS ESC STOP	Start		Log Esc Stop	

ShortPB[RstMon]On でこの View 画面になります



5-3. サンプルアプリケーションを走らせる前の準備

- 1)評価ボード「AHE261」のSW「P40~P47」の8点全てOFF(下側)にします。
- 2)評価ボード「AHE261」の DIP-SW11 の 1·2 の 2 点を ON(上側) にします。
- 3) 評価ボード「AHE261」の CN13 にブザーの付いたコネクタ(AHC-SAMP1)を装着します。 (無い場合は、未装着でも構いません)
- 4) CPU 基板「CAT261」の DIP-SW1 の 1・2・3・4 の 4 点を ON(下側)にします。

5-4. サンプルアプリケーションを走らせます。

- 1) DEF のショート PB「Go」をクリックします。[5-2-8]図を参照
 - (1) 評価ボードの LCD に下図のようになれば正常です。

TOPPERS/OSEK 1.1	[5-4-1]
for CAT261[P4xS]	

(2) 評価ボードの SW-P40 を ON(上側) にする。

[5-4-2] LCD Task1 が起動します。

(3) 評価ボードの SW-P41 を ON(上側) にする。

LEDTask2 [xxxxx]

[5-4-3] LED Task2 が起動します。

(4) 評価ボードの SW-P42 を ON(上側) にする。

BzzTask3 [xxxxx] xxxxHz[+PF7-PC5]

[5-4-4] Bzz Task3 が起動します。(ブザー出力タスク)

PF7-PB ON で、+100Hz の周波数を出力します。
PC5-PB ON で、-100Hz の周波数を出力します。 (100~1100Hz)
TPU3 TIOCA3 アウトプットコンペア出力を利用しています。



(5) 評価ボードの SW-P43 を ON(上側) にする。

SioTask4 [xxxxx] AHE-ToppersCD1 f

[5-4-5] SioTask4 が起動します。

{"AHE-ToppersCD1 for TOPPERS/OSEK by Aone"};

SIO 通信が正常送受信を継続している間は、上記文字が流れて表示します。 <動作>

①SCI-TxD0より8文字送信 -> SCI-RxD1で受信 -> 受信データをLCD表示
 ①SCI-TxD1より8文字送信 -> SCI-RxD0で受信 -> 受信データをLCD表示

評価ボード「AHE261」の DIP-SW11 の 1・2 の 2 点を ON(上側) にすることにより、 SCI-2CH のループテストになります。

試しにどちらかを OFF にしてみて下さい。タイムアップ表示されます。

(6) 評価ボードの SW-P44 を ON(上側) にする。

MelTask5 [xxxxx] ネコフンシ゛ャッタ PC2{S}

[5-4-6] MelTask5 が起動します。

PC2-PBをONしますと、ブザーより「ネコフンジャッタ」の曲が流れます。

(6) 評価ボードの SW-P44・P43・P42・P41・P40 と順番に OFF(下側) にする。

LcdTask1 suspend

[5-4-7] OFF 順番に各タスクが Suspend していきます。

(6) 評価ボードの SW-P47 を ON(上側) にして、PA0-PB を ON にします。

ShutDown	WaitWDG
E_OK	

[5-4-8] シャットダウンフックルーチンが実行されます。 このサンプルは、ウォッチドッグタイマーを起動させています ので、「1.68sec」後にリセットされ再起動します。

以上です。



第6章 新規プロジェクトを追加する場合の手順例

6-1. プロジェクトタイプの作成

TOPPERS/OSEK 下で新規プロジェクトを追加する場合、Hew 設定を簡略化するための手順案 を記述します。

▲ cat261 用のプロジェクトテンプレートは作成してありますので、下記手順でプロジェクト タイプ(カスタム)を作成して下さい。

(必ず Compiler]と Assembler と Linker のディレクトリが変更済みである事!!)

1) 空プロジェクト「Project」をアクティブプロジェクトに指定します。



2) Hew メニュー<プロジェクト>-<プロジェクトタイプの作成>をクリックします。

High-pe	rformance	Embedded Wo	rkshop	X
1	ワークスペーン すべての変更	スまたはプロジェクト(「を保存しますか?	の設定が変更されま	ました。
	an M	いいえい	キャンセル	

[6-1-2] 保存を促すメッセージです。 「はい」をクリックします。





[6-1-3]

プロジェクト新規登録時に指 定するタイプ名を入力する。 ex)toppers_osek_kpit_project

はい(Y)側をチェック

「次へ」をクリックします。



左図の様に「デフォルト」のま まで、

「完了」をクリックします。



6-2. 新規プロジェクトを登録します。

1) Hew メニュー<プロジェクト>-<プロジェクトの挿入>をクリックします。

挿入 ○ <u>新規プロジェクト(N)</u> ○ 既存プロジェクト(E):	今照(B)	OK キャンセル	新規プロジェク 側チェックにて 「OK」をク! クします。
プロジェクト名を登録 新 <u>規プロジェクトの</u> 邦入	とします。	?(X [6-2-2]
プロジェクトタイプ ○ C Application ○ C++ Application ● Library ■ toppers_osek_kpit_project プロパ5	ワークスペース名(W): 10ppers_ooek=1_1 プロジェクト名(P): cat261new ディレクトリ(Q): (0¥Hew44*TOPPERS¥toppers_osek=1_1¥tools¥h8s=1 CPU能的(C): H85,H8/300 ▼ ツールチェイン(D): [KPIT GNUH8 [ELF] ▼	李照(图)	

 ①プロジェクトタイプを前項で登録した「toppers_osek_kpit_project」を指定します。
 ②プロジェクト名に任意な目的プロジェクト名を入力します。後の説明でプロジェクト名が 必要になりますので、ここでは新規プロジェクト名を「cat261new」とします。
 ③上記設定で「OK」をクリックします。

3) Hew に新規プロジェクトが作成されます。



[6-2-3]

左図のように、新規プロジェクトが Hew に登録されます。



- 4)新規プロジェクト用に新規ディレクトリ作成とファイルをコピーします。
 - ① .¥toopers_osek¥config¥h8s-kpitgnu-2612 の下に、

新規プロジェクト名「cat261new」のディレクトリを作成します。



②作成した「cat261new」に、「cat261」下の全ファイルをコピーします。





① .¥toopers_osek¥syslib¥h8s-kpitgnu-2612の下に、

新規プロジェクト名「cat261new」のディレクトリを作成します。





5)システムジェネレータ用バッチファイルの一部を変更します。

のファイルを何らかのエディタで開きます。

@REM SG 実行バッチファイル

<元ファイル> [6-2-5-1]

@REM カーネルコンフィグレーション
del kernel_cfg.c
del kernel_id.h
...¥..¥sg¥sg.exe main.oil
-template=..¥..¥..¥config¥h8s-kpitgnu-2612¥Project¥Project.sgt
-I..¥..¥sg¥impl_oil -os=ECC2

<変更ファイル> [6-2-5-2]

@REM SG 実行バッチファイル

@REM カーネルコンフィグレーション

del kernel_cfg.c

del kernel_id.h

..¥..¥..¥sg¥sg.exe main.oil -template=..¥..¥..¥config¥h8s-kpitgnu-2612¥cat261new¥Project.sgt

-I..¥..¥..¥sg¥impl_oil -os=ECC2

上記の様に、Project を新規プロジェクト名「cat261new」に変更します。



6)新規プロジェクト「cat261new」にユニット「ソースファイル」を登録します。

①フォルダ名「config_h8-kpitgnu-2612」に登録

Hew メニュー<プロジェクト>-<ファイルの追加>をクリックします。

.¥toppers_osek¥config¥h8s-kpitgnu-2612 に移動します。

'cat261new'プロ:	ジェクトにファイルを追加 ? 🔀	[6-2-6-1]
ファイルの場所(1):	🔁 h8s-kpitenu-2612 💽 🗢 🖆 📰•	①cpu_config.c
Cat261	H) cpu_defs.h	<pre>②cpu_support.s</pre>
h8s2612st cpu_config.c H cpu_config.h H cpu_contexth	H tool_configh H tool_defsh m cpu_supportS m startS	③start.s
ファイル名(N):	「startS" "cpu_config.c" "cpu_supportS" 注意加	の3ファイルを「追
ファイルの種類(工):	Project Files チャンセル	加」します。
	☑ 相対パス(B) 「 登録済みファイルを非表示(P)	

ファイル選択後「追加」をクリックしますと、Hew ツリーでは一旦「Assembly souce file」 と「C souce file」のフォルダに入りますので、マウスドロップにて目的フォルダに移動し て下さい。

😡 cat261 new - High-performance Embedded Workshop - [main.c]	[6-2-6-2]
◇ ファイル(E) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) 基本設定(U) ツール(T) テスト(S) ウィンドウ(W) ヘルプ(H) _ □ □ ×	
▋▙▆▆▓▆▓▆▓▆▋₩	移動後の状
toppers_osek-1_1 cat251 cat25 cat251 cat25 cat251 cat251 cat251 cat251	態
Image: Second secon	
Phase Linker starting Phase Linker finished	



②フォルダ名「config_h8-kpitgnu-2612_Project」に登録

Hew メニュー<プロジェクト>-<ファイルの追加>をクリックします。

.¥toppers_osek¥config¥h8s-kpitgnu-2612¥cat261new に移動します。

'cat261new'プロ	ジェクトにファイルを追加 🔹 💽 🔀	[6-2-6-3]
ファイルの場所(1):	🔁 cat261new 💽 🔶 🖆 📰 🕇	$①$ sys_config.c
sys_config.c iodefine.h strh8s2612.h sys_config.h sys_config.h sys_defs.h sys_support.S		② sys_suport.s の 2 ファイルを「追
ファイル名(11):	「"sys_supportS" "sys_config.c" 注意加	加」します。
ファイルの種類(工):	Project Files ++>ンセル	
	▼相対パス(R) 「登録済みファイルを非表示(P)	

前項と同じ様に、別フォルダに入りますので、マウスドロップで目的フォルダに移動して下 さい。

③フォルダ名「kernel」に登録

Hew メニュー<プロジェクト>-<ファイルの追加>をクリックします。

.¥toppers_osek¥kernel に移動します。

'cat261new'フロ	ジェクトにファイルを追加 ? 🔀	[6-2-6-4]
ファイルの場所中:	🎦 kernel 💽 🗢 🖆 🔠	①alarm.c
alarm.c	Gtask_manage.c H task.h	2)event.c
interrupt.c	H check.h	③interrupt.c
resource.c	H interrupth H osek_kernelh	④osctl.c
C task.c	H] resource.h	⑤resource.c
ファイル名(N):	『task_managecで "alarmc "eventc "interruptc ' isto	⑥task.c
ファイルの種類(工):	Project Files ・ キャンセル	⑦task_manage.c
	▼相対パス(R) 「登録済みファイルを非表示(P)	の7ファイルを

前項と同じ様に、別フォルダに入りますので、マウスドロップで目的フォルダに移動して下 さい。



④フォルダ名「syslib_h8s-kpitgnu-2612」に登録

Hew メニュー<プロジェクト>-<ファイルの追加>をクリックします。

.¥toppers_osek¥syslib¥h8s-kpitgnu-2612に移動します。

'cat261new'プロ:	ラェクトにファイルを追加	? 🛛	[6-2-6-5]
ファイルの場所(1):	🔁 h8s-kpitenu-2612 💌 🗢 🗈 🖻	* -	①sys_timer.c
Cat261 cat261new h8s2612st cat_sys_timer.c cat_sys_timer.h cat_sys_timer.h	H osek_libh		② osek_lib.c の 2 ファイルを「追
ファイル名(11):	f"osek_lib.c" "cat_sys_timer.c"	追加	加」します。
ファイルの種類(工):	Project Files	キャンセル	
	▼ 相対パス(R) 「 登録済みファイルを非表示(P))	

前項と同じ様に、別フォルダに入りますので、マウスドロップで目的フォルダに移動して下 さい。

⑤フォルダ名「syslib_h8s-kpitgnu-2612_Project」に登録

Hew メニュー<プロジェクト>-<ファイルの追加>をクリックします。

.¥toppers_osek¥syslib¥h8s-kpitgnu-2612¥cat261new に移動します。



前項と同じ様に、別フォルダに入りますので、マウスドロップで目的フォルダに移動して下 さい。



7)新規プロジェクト「cat261new」のリンク順番を指定します。

Hew メニューの<ビルド>-<Linker>をクリックして下さい。

ker Options (Debug)	
chives Input Output List Sections Other Command	file
Archive search directories:	
c¥program files¥renesas¥hew¥tools¥kpit cummins¥gnush-el	Add
c-sprogram meswrenesaswneww.couswupit.comminiswgnusin-ei	Bemove
	Move up
c >	Move dogs
Archive (library) files:	
liboptc.a	Add
libecc.a	Remove
	Mgve up
	Move down
Search archives repeatedly for undefined references	
✓ Use Optimised Libraries	
Common options:	
"\$(TCINSTALL)Vsh-elfVsh-elfVlbVm2" -o "\$(CONFIGDIP) V\$(PROJECTNAME)x"oformat=srec -Map "\$(CONFIGDIP) V\$(PROJECTNAME).map" -e_start -l "optc" -l "optm" -l "gc	xc‴
Show option description OK	1 **`/*

<Input>タグをクリックします。

File	File Type	Location	<u>A</u> dd.
a starto	GNU	g#new4#toppers#top	Remove
Dosek libo	GNU	gifthem4Wtoppers#top_	
Courcentine	GNU	a¥hew4¥toppers¥top	Disable
cpu support.o	GNU	g¥hew4¥toppers¥top_	M. 17.
sys config.o	GNU	g¥hew4¥toppers¥top	Modify_
sys_support.o	GNU	g¥hew4¥toppers¥top.	Move up
sys_timer.o	GNU	q¥hew4¥toppers¥top	
sys_serialo	GNU	q¥hew4¥toppers¥top_	Move down
hw_sys_timer.o	GNU	q¥hew4¥toppers¥top	
hw_config.o	GNU	q¥hew4¥toppers¥top	
hw_pwm.o	GNU	q¥hew4¥toppers¥top_	
¢		>	
ommon options: \$(TCINSTALL)¥sh \$(PROJECTNAME \$(PROJECTNAME	-elfVsh-elfV)x"oforn	/lib¥m2" -o "\$(CONFIGDI) nat=srec -Map "\$(CONFIG start =l _onto" =l _ontm"	ນ IDIRV

[6-2-7-2]

[6-2-7-1]

全てのディレクトリを現ワークス ペースのディレクトリと相違があ った場合は現ディレクトリに変更 して下さい。

設定は、「\$(CONFIGDIR)*.o」にし ましたので変更する必要は無いか と思います。

マウスでのダブリクリックで変更 できます。

Modify File	[6-2-7-3]
5:¥toppers osek¥tools¥h8s-kpitgnu-2612¥cat261¥debug ▼ ► OK	
Cancel	
変更が必要な場合は、「CAT261new」のディレクトリに変更す	る 。



▲ 以上の作業で、新規プロジェクトの追加作業は終了です。目的のプロジェクト仕様に合わせた「main.c」を作成して下さい。

フォルダ名「tools_h8s-kpitgnu-2612_Project_appsrc」

ディレクトリ「.¥toppers_osek¥tools¥h8s-kpitgnu-2612¥cat261new¥appsrc」に空ファイル として用意してあります。

8) OS 定義および Task の追加やプライオリティを変更したい場合は、ディレクトリ

「.**¥toppers_osek¥tools¥h8s-kpitgnu-2612¥cat261new**」にある、「main.oil」のテキストフ ァイルを変更して下さい。

定義仕様に関しては、「OPPERS/OSEK カーネル SG 取扱説明書」株式会社ヴィッツ製をご覧下さい。

第7章 備考

7-1.おわりに

本アプリケーションノートは、いたらない所が多々有ると思います。意味不明な箇所がありま したら、遠慮なくメールにて申し付け下さい。積極的に改訂し、より判り易いノートにしたく思 っていますので、皆様の御協力を御願い申し上げます。

2008年6月 著者

〒486-0852
愛知県春日井市下市場町6-9-20
エーワン株式会社
Tel 0568-85-8511
Fax 0568-85-8501
E-mail cat-i@aone.co.jp
URL http://www.aone.co.jp