

【H-debugger 対応】 TOPPERS/OSEK カーネル

アプリケーションノート

H8S/2000シリーズ (BOOT-PBC) H8S/2268F KPIT-GNU [Hew] 版

 $2 \ 0 \ 0 \ 8 \ / \ 1 \ 0 \ / \ 1 \ 5$

Rev1. 00 (2008/10/15)



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 TOPPERS/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(H2268)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13
3-4. H-debugger でのプロファイル表示(DEF 7.00B 以上)
第4章 ハード構成およびシステム構成17
4-1. ハード構成17
4-2. システム構成19
4−2−1. H8S/2268F のプログラムメモリ MAP19
4-2-2. プログラムサイズの詳細 MAP 19
4 – 2 – 3.CPU 基板の I/O マップ表 20
第5章 サンプルアプリケーションの準備と動作確認
5-1. Hew での準備(ルネサス製)22
5-2. H-debugger(DEF.exe)での準備(Aone 製)27
5-3.サンプルアプリケーションを走らせる前の準備
5-4.サンプルアプリケーションを走らせます。32
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	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 「H8S/2268F」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関しては、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。

TOPPERS/0	SEK カーネル外部仕様書 ----------	株式会社ヴィッツ製
TOPPERS/0	SEK カーネル SG 取扱説明書	株式会社ヴィッツ製
TOPPERS/0	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
OSEK/VDX	Operating System Ver2.2.1	OSEK/VDX 仕様
OSEK/VDX	Binding Specification Ver1.4.2	OSEK/VDX 仕様
OSEK/VDX	OIL Specification Ver2.5	OSEK/VDX 仕様
	TOPPERS/03 TOPPERS/03 OSEK/VDX OSEK/VDX	TOPPERS/OSEK カーネル外部仕様書 TOPPERS/OSEK カーネル SG 取扱説明書 TOPPERS/OSEK カーネルアプリケーションノート OSEK/VDX Operating System Ver2.2.1 OSEK/VDX Binding Specification Ver1.4.2 OSEK/VDX OIL Specification Ver2.5

*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

Hew Version 4.04.01.001-----Renesas 製
 KPIT GNUH8[ELF] Toolchain v0801 -----KPIT 製
 に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

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。

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-2268	// 開発環境分類
+CPU 依存部	
¥H2268	// 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-2268	// 開発環境分類
+CPU 依存部	
¥H2268	// システム依存部
¥tools	// Hew 管理階層
¥h8s-kpitgnu-2268	// 開発環境分類
+ワークスペース	
¥H2268	// プロジェクト1
¥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> 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、1本のプロジェクトを用意しています。

1) プロジェクト1 ¥H2268 実行オブジェクト【H2268.x/H2268.mot】 本アプリケーションノートのサンプルプロジェクト

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(H2268)

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) 5個のタスク

	(1)	MainTask	各 Task の起動処理	
	2	Task1	センサー入力(2 点)の状態表示	
	3	Task2	PWM1/2/3の3点パルス出力	
	4	Task3	RS232C による PWM デューティーb	比変更
	5	HighPriorityTask	各タスクのスタックオーバー監視	
2)	4個	国の割込み処理		
	\bigcirc	システムタイマー	TMR0/1(16bit) 約 1ms 割込み	SysTimerInt()
	2	SCI1 受信エラー	ERI1	ErrHwSerialInt()
	3	SCI1 受信	RXI1	RxHwSerialInt()
	4	SCI1 送信終了	TXI1	TxHwSerialInt()

3) 5 個のイベント

MainEvt/T1Evt/T2Evt/T3Evt/THEvt

- 4) 2個のアラーム
 - MainCycArm/SetEvtHArm
- 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)

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) プレタスク/エラーフックルーチン

PreTaskHook/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)



①Task1(センサー入力)の起動条件 -- 基板上のSW5 ON で起動する。
 ②Task2(PWM出力)の起動条件 --- 基板上のSW4 ON で起動する。
 ③Task3(RS設定)の起動条件 ---- パソコンよりスペース入力





①センサ入力 ----基板上の SW2/SW3 をセンサ替りに使用する。
 ②Task1 の終了条件 --基板上の SW5 OFF で終了とする。



🎢 エーワン株式会社

①**Task2**の終了条件 -- 基板上の SW4 ON で終了とする。



🎢 エーワン株式会社

<パソコンからの指示コード>

\bigcirc	'Q'キー	 PWM1 センタ+1	(-50 \sim 50)
2	'W'キー	 PWM1 センタ-1	(-50 \sim 50)
3	'A'キー	 PWM2 パーセント+1	(-100~100)
4	'S'キー	 PWM2 パーセント-1	(-100~100)
5	'Z'キー	 PWM3 パーセント+1	(-100~100)
6	'X'キー	 PWM3 パーセント-1	(-100~100)
\overline{O}	Space	 メニュー再表示	
8	'E'キー	 Task3 の終了指示	



🎢 エーワン株式会社



3-3. H-debugger 対応に必要なポーティング

H8S/2268F は、BOOT ポート(SCI0)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-renesas-2268¥H2268¥sys_support.src				
hardware	e_init_hoo	ok:		
	ldc.b	#H'80, ccr		
	ldc.b	#H'06, exr		
	mov.b	#H'21, r0l		
	mov.b	r0l, @SYSCR		
	; H-debu	ıgger 用 リセット遅延	の為、 20msWait	
	; Target	19.6608MHz		
wait20m	s:			
	mov.w	#20,r0		
waitrst:				
	bsr	_SOFT1MS		
	dec.w	#1,r0		
	bne	waitrst		
	rts			
;/*SOFT1	MS() 1r	ns ソフトタイマー (19.6608M	1Hz) Non Wait */	
_SOFT1N	MS:			
	push.w	r0	;	
	mov.w	#4915,r0	;/* 4915*4=19660cyc	*/
wait:			;	
	dec.w	#1,r0	;/* 1 clock	*/
	bne	wait:16	;/* 3 clock	*/
	pop.w	r0	;	
	rts			



<理由>

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-renesas-2268¥H2268¥sys_config.c			
/***********	*********	*/	
/* ターゲットシステム依存の初期化		*/	
/**************************************	**********	*/	
void sys_initialize(void)			
{			
	/* PBC = 7 その他 6 以下とする debug 対応	*/	
IPRA = 0;	/* PBC 以外のプライオリティを 0 にする。	*/	
IPRB = 0;	/*使用する割り込みは、後の1/0初期化で	で */	
IPRC = 0;	/*プライオリティを設定する。	*/	
IPRD = 0;			
IPRE &= 0xf0;	/* < PBC	*/	
IPRF $= 0;$			
IPRG = 0;			
IPRI = 0;			
IPRJ = 0;			
IPRK $= 0;$			
IPRL = 0;			
IPRM = 0;			
IPRO = 0;			
}			

<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-renesas-2268¥cpu_support.src			
;H 7>6 に変更する。	H-debugger 対応の為		
EAK_WSK	.DEFINE	H 06	

<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-2268¥sys_timer.h		
#define USE_PROI	FILE	
#ifdef USE_PROF	FILE	
#define l	PROFILESIZE	512
#define I	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 __TaskProFile[PROFILESIZE];

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。



<注記>

①NMI 信号を H-debugger と接続する為、基板改造が必要です。
 ②FWE 信号による SCI0 のスイッチング回路を無効にする必要があります。

U3-B74HC125の6番ピンをカットする。

<デバッグに必要なオプションケーブル>

()AHC-FREE-KIT	ターゲット側をフリーにしたケーブル(別売品)	1本
②AHC-RS232C-TTL	RS232C ドライバー付きケーブル(別売品)	1本



- 1) CPU 基板(HSB8S2268F)の J1 と AH7000 をケーブル(AHC-FREE-KIT)で接続します。
- 2) AH7000 とパソコンを USB ケーブルで接続します。
- 3) CPU 基板(HSB8S2268F)の J5 とパソコンをケーブル(AHC-RS232C-TTL)で接続します。
- 4) CPU 基板に電源 (DC +5.0V)が供給できるように接続します。

[接続応用例]





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。

4-2-1. H8S/2268F のプログラムメモリ MAP

CPU 内部	0x0ベクタテーブル①
H8S/2268F フラッシュ ROM 256Kbyte	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
	0x7FFFFFROM 最終
CPU 内部 RAM	0xFFB000 アプリケーション使用 RAM の開始番地③
約 16Kbyte	0xFFEFBF 初期スタックポインタ位置(0xFFEFBE)④
CPU 内部 RAM2	0xFFFFCO デバッグモニタが使用する RAM 開始番地
64byte	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,.rodata....etc
- ③アプリケーション使用 RAM の開始番地------ .data,.bss....etc
- ④初期スタックポインタ位置------.stack

4-2-2. プログラムサイズの詳細 MAP

開始番地	サイズ	分類
0x800	0xD9E	アプリケーション (mainetc)
0x159E	0x1532	割込みハンドラ、内部 I/O 関係の関数
0x2AD0	0x29E4	TOPPERS/OSEK カーネル
0x54B4	0x503	C ライブラリ、ROM テーブル、etc
0x59B7		最終アドレス



4-2-3. CPU 基板の I/O マップ表

CPU 基板(HSB8S2268F)の SW 関係						
	基板シンボル	ホ [°] ートシンホ [°] ル		方向	用途	
	CPU 基板	CPU 基板				
	SW2	P70		入力	センサー入力0	
	SW3	P71		"	センサー入力1	
	SW4	P72		"	Task2 の起動	
	SW5	P73		"	Task1 の起動	

CPU 基板(HSB8S2268F)の PWM 関係					
	基板シンボル	ホ [°] ートシンホ [°] ル	J5 シンホ [*] ル	方向	用途
	CPU 基板	CPU 基板	CPU 基板		
	LED-D1	P10/TIOCA0	J5-10	出力	サーボ (PWM1)
	LED-D2	P11/TIOCB0	J5-09	出力	右モータ(PWM2)
	LED-D3	P12/TIOCC0	J5-08	出力	左モータ(PWM3)
	LED-D4	P13	J5-07	出力	右モータ方向 0=正
	LED-D5	P14	J5-06	出力	左モータ方向 0= 正
	LED-D6				空き
	LED-D7	P16	J5-04	出力	LED0
	LED-D8	P17	J5-03	出力	LED1

CPU 基板(HSB8S2268F)の RS232C			J5	適用	ケーブル	AHC-RS232C-TTL(別売)
	ホ゜ートシンホ゛ル	ピン番号			方向	信号名
	CPU 基板	CPU 基板				
		1	空き			
		2	空き			
		3	空き			
		4	空き			
	P34/RxD1	14 (13)	黒白		入力	SCI1-RXD(TTL)
	P33/TxD1	15 <mark>(16)</mark>	赤白		出力	SCI1-TXD(TTL)
	VCC	31 <mark>(32)</mark>	赤		VCC	+5.0V
	VCC		空き			
	GND	33 <mark>(34)</mark>	黒		GND	GND
	GND		空き			

(nn)実コネクタ番号--コネクタ逆挿しの為、番号に注意!!



CPU	CPU 基板(HSB8S2268F)のデバッガ関係 J1 適用ケーブル AHC-FREE-KIT					
	H-debugger	H-debugger コマクロ 死 旦	基板側	方向	信号名	
		コイソク留方				
	SD1	1	J1-17	入力	P31/RxD0	
	GND	2	J1-14	GND	GND	
	SD2	3	J1-15	出力	P30/TxD0	
	GND	4	J1-12	GND	GND	
	SCK	5	J1-19	入出力	P32/SCK0	
	GND	6	J1-16	GND	GND	
	NMI	7	J1-09	入力	/NMI	
	VCC	8	J1-20	+5V	VCC	
	/RES	9	J1-01	入力	/RES	
	FWE	10	J1-03	入力	FWE	



第5章 サンプルアプリケーションの準備と動作確認

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_2268」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_2268¥config¥h8s-kpitgnu-2268 --> .¥toppers_osek¥config
.¥toppers_osek_Kpit_2268¥syslib¥h8s-kpitgnu-2268 --> .¥toppers_osek¥syslib
.¥toppers_osek_Kpit_2268¥tools¥h8s-kpitgnu-2268 --> .¥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-kpitgmu-2268¥h8s_kpitgnu_2268.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。

ウォーコ	25	[5-1-1]
	このワークスペースのディレクトリが移動されました。	
4	旧ディレクトリ: C.¥toppers¥toppers_osek_Kpit¥tools¥h8s-kpitgnu-2268 新ディレクトリ: C.¥toppers_osek¥tools¥h8s-kpitgnu-2268	
	ワークスペースを聞きますか?	
	□ 今後表示しない(型)	
	(いたの) キャンセル	



 アクティブプロジェクトが「H2268」になっていることを確認します。H2268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 「H2268」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	
Archive search directories: C#program files¥renesas¥hew¥tools¥kpit cummins¥gnush~el C#program files¥renesas¥hew¥tools¥kpit cummins¥gnush~el	Add	<input/> タグをクリックします。
	Move yp	
<	Move down	
Archive (jbrary) files:		
liboptc.a	Add.	
libecc.a	Remove	
	Mgve up	
	Moge down	
Search archives repeatedly for undefined references		
✓ Use Optimised Libraries		
Common options:		
"\$(TCINSTALU¥sh-elf¥sh-elf¥lib¥m2" -o "\$(CONFIGDIR) ¥\$(PROJECTNAME)x"oformat≋srec -Map "\$(CONFIGDIR) ¥\$(PROJECTNAME).map" -e_start -l "optc" -l "optm" -l "go	ד 🖌	
Show option description	**>セル	



File	File Type	Location A	Add.	全てのディレクトリを現ワー
start.o	GNU	q¥hew4¥toppers¥top_	Bamaua	ペースのディレクトリレ相違。
maino	GNU	q¥hew4¥toppers¥top_	Nemove	「 ハリノイレクトリこ相座/
Cosek_lib.o	GNU GNU	q#hew4#toppers#top_ g#hew4#toppers#top_	Disable	った場合は現ディレクトリに
cpu_support.o	GNU	q¥hew4¥toppers¥top_	Modify	して下さい
sys_supporto sys_supporto sys_serialo hw_sys_timer.o hw_config.o	GNU GNU GNU GNU GNU GNU	q¥hew4¥toppers¥top. q¥hew4¥toppers¥top. q¥hew4¥toppers¥top. q¥hew4¥toppers¥top. q¥hew4¥toppers¥top.	Move yp Move gown	設定は、「\$(CONFIGDIR)*.o」 ましたので変更する必要は無い と思います。
Common options: "\$(TCINSTALL)WsH \$\$(PROJECTNAM) \$(PROJECTNAM) \$(PROJECTNAM)	n-elf¥sh-elf E).x"ofor E).map" -e tion	VIIbVm2" -o "\$(CONFIGDIR) mat=srec - Map "\$(CONFIGI _start -I "optc" -I "optm" -I 	NR) ▲ "ecc" マ キャンセル	マウスでのダブリクリックでゑ できます。

▲ 新規プロジェクトの作成時に必要です。

8) 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と確認をして下さい。

Cancel

- 9) アクティブプロジェクトを「H2268」に戻します
- 10) 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。



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 Ver7.00D		[5-2-3]
ジャートの All 10	C MN C View CPUH85/2268F A	dvance 19.6608 x1	
	C Src C Mix C Asa	<u>.</u> г.	
		1	
		-1	
		3	
CEPI	Y Y C BP2	×	
	💌 (tal.) 💌 char	💌 16jl 💌 💈	
		2	
		*	
Go Break RstWon	Reset Vin Res Vatch Sym. T	race Step CTrac	
		(A) (A)	
Start	Log	Esc Stop	

6) 最初だけモニタ不在の通知が表示されます。

DEF 📽 🛛	[5-2-4]
 ブートモードかモニター不在です。 <ファイル>-<ブートロード>-<モニタブログラム>を実行して下さい。 又は、「OK」クリックにて、<ブートロード>-<モニタブログラム>を実行します。 [OK]でDEFの処理を継続します。 	
OK ++>セル	「OK」を

「OK」をクリックします。





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 &	Flashwriter Ver7.00D A-one AH7000 1	/er6.80(2008 📜 migVh8s-renesas-22684	[5-2-5
グリルビデータログ 奥円 (後止 1053、DL PC 000		Advance 19,6608	
0000000 mm	- FORCE BILL OF ASS	A+A	
00101202+ FFFF	and b Dyf	6-8, e71 (- -
. 00001804: FFFF	nov.b Divi	f:8.r71 <	D
. 00100805: FFFF	nov,b Dixf	f:8.r71 <	D
. 00000808; FFFF	nov.b D0xf	f:8.r71 <	D
. 0000080A: FFFF	mov.b D0xf	f:8.r71 <	1>
. 0010080C: FFFF	nov.b 20xf	f:8,r71 <	D
. 0030080E: FFFF	nov.b D0xf	f:8.r71 <	1)
. 00000810: FFFF	mov.b Doxf	f:8,r7) <	1D
. 00000812: FFFF	mov.b 20xf	f:8,r71 <	1D
. 00000814: FFFF	nov.b D0xf	f:8,r71 <	1>
. 00000816: FFFF	nov.b \$0xf	f:8,r71 <	D
. 00000818: FFFF	mov.b I0×f	f:8.r71 <	D
. 0000081A: FFFF	mov.b D0xf	f:8,r71 <	1>
• 0000081C: FFFF	nov.b 10xf	f:8,r7) <	1D
 D000081E: FFFF 	mov.b B0xf	(:8,r7) <	1>
• 00000820: FFFF	mov.b 20xf	f:8,r71 <	1>
 00000822: FFFF 	mov.b D0xf	f:8,r7) <	1> ·
• 00000824: FFFF	mov.b 20×f	f:8,r7) <	1> =
@ BP1 00000000	CLR Non C BP2	00000000	
-	 (なし) char 	💌 16湯	X
			*
			-
			*
Go Break RstMon	Reset Tin Reg Tatch Sy	Trace Step	CFraci
Internet in the second s			100
Start		Los Erel	Stop
DUBLIC		LUE ESC	OC OP

8) DEF メニュー<ファイル>-<ダウンロード>をクリックします。

ユーザモードでダウンロ

.¥toopers_osek¥tools¥h8s-kpitgnu-2268¥tools¥H2268¥debug まで降ります。

DEF アブソリュート/ヘキサファイル指定でのダウンロード 🛛 🛛					[5-2-6]
ファイルの場所の:	Debug		- 🗢 🗈 👉 📰		
名前	サイズ	種類 🔺	更新日時		H2268.x を選択後
E H2268.x	107 KB	X ファイル	2008/10/14 1	7:45	「開く」をクリック
					します。
< ファイル名(N):	H2268.x			0	
ファイルの種類(工):	ABS File (*.abs,*.x)		· ++>	也儿	
				11	
DEF איז	z−⊅в 🚺 [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。

=	A-one I	H- DEbu	seer & Flashw	riter Ver	7.00D A-one AH7000 Ver6.80(2008-03	-17) for HOS	[5-2-8]
77	YND :	データ(ロ)	実行(3) ブレー	-ク(日) 評	1込みゆ オブション(2) ヘルプ(2)		
11	F止 1	前达 6	PC 000000800 [T RUN IT	View CPU H85/2268F Advance 19.660	8 x1 書込回鉄00003TOP 0x00000800	
000	00800	\$71:	start.S		IF Src ⊂ Mix ⊂ Asm start.S	💌 🔽 NonWait#(9685	
	00000800	: 21	aov.1	I_stack,	4P	1	L
		72					1
		73	:	turne lelt	had (1985-14) (1) (20-12-14) (40-05)		
		74	; naro	mare_mit	THORE ON A THE C. (I. CLARCH MAR)		
		76	: 3-	ゲットハー	- ドウェアに体存して必要な抑制化体理がある規	<u>A</u>	
		17	: 13.	hardware	init hook という関数を用意する。		
		78	1	A			
	00000806	: 79	nov. I	Ihardean	e_init_hook, er0 ; mov命句にてシンボルが	0の場合、2=1となる	
	D000080C	: 80	beq	hardware	_init_hook_r		
	DODODBOE	: 81	jar	Shardvan	e_init_hook		
		82	hardware_in	ait_hook_r			
		83	:	40.752 cm.*	(お271111)(レナス、/00~しうまた)		
	-		, oace	165545	256040103-2 (voli10-120)		1
	00000812	: 86	acv. I	I data.e	d		
	00000818	: 87	eov. I	1 edata.	erl		
	0000081E	: 88	nov. I	I_adata.	er2		
	00000824	: 89	cap.1	er8,er1			
	00000826	: 90	beq	start_1			
		\$1	start_1:				
•	00000828	: 92	wov.b	Ber2, r31	;get from src		
	0000082A	33	sov.b	r31,8er0	Iplace in dest		
	00000820	1 04	inc. I	al,erz	tine and		
	00000830	: 36	cap. I	er0.er1	tdest ## edata?		
	00000882	: \$7	boe	start		-	
6	RP1 0000	00000		- IGR			1
1000		L.I					4
1000							
							-
							1
-	-			1			1
L	Go B	reak R	stmon Reset	Tin	Reg Watch Sym Trace Sto	p Cirac CStep DI EI	
						-	<u> </u>
-	1-						4
181	arti				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) CPU 基板上の SW2/3/4/5 を OFF の状態にします。

2) パソコン側のソフト「Tera Term」を起動します。



3)「Tera Term」メニューの<Setup>-<Serial port>をクリックします。

Tera Term: Serial p	ort setup		[5-3-2]
Port: Baud rate: Data: Parity: Stop: Flow control:	COM1 • 9600 • 8 bit • none • 1 bit •	OK Cancel <u>H</u> elp	Port 以外の通信仕様を左図のように 設定して下さい。
0 mse	c/ <u>c</u> har 0 r	nsec/line	「OK」をクリックします。



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

「Tera Term」画面は下画面のようになります。

🗏 Tera Term - COM1 VT 📃 🗖 🔀	[5-4-1]
<u>Eile E</u> dit <u>S</u> etup C <u>o</u> ntrol <u>W</u> indow <u>H</u> elp	
TOPPERS/OSEK 1.1 for H8S2268	
~	

2) センサン入力タスク(Task1)を起動します。

CPU 基板上の SW5 を ON にします。

「Tera Term」画面は下画面のようになります。



センサ入力値を16進数で表示します。

[5-4-2]

センサ代わりに SW1/SW2 を代用します。

3) PWM 出力タスク(Task2)を起動します。

①**CPU** 基板上の **SW4** を **ON** にします。

「Tera Term」画面は下画面のようになります。





4) RS 設定タスク(Task3)を起動します。

①「Tera Term」よりスペースキーを入力します。

「Tera Term」画面は下画面のようになります。

🗏 Tera Term – COM1 VT 💦 🔲 🔀	[5-4-4]
Eile Edit Setup Control Window Help	
TOPPERS/OSEX 1.1 for H8S2268	
Taski Activate Ext.Contact.Value 0x00 Taski Suspended	
Task2 Activate	
Task3 Activate Task3 Menu '0' key : PVM1 Offset(-50~ 50) +1 'V' key : Offset(-50~ 50) -1 'A' key : Percent(-100~100) +1 'S' key : Percent(-100~100) -1 'Z' key : PWM3 Percent(-100~100) +1 'X' key : Percent(-100~100) -1 Space Key: Menu Disp 'E' key : Task3 Suspended Now PVM1 Percent 000 Now PVM3 Percent 000 Now PVM3 Percent 000	
②PWM1(サーボ)を出力します。	<cpu j5-10="" 基板「hsb8s2268f」の=""></cpu>
QとW キーを押して下さい。数値にも	らとずいて PWM 出力します。
③ PWM2 (右モータ)を出力します。	<cpu j5-09="" 基板「hsb8s2268f」の=""></cpu>
AとS キーを押して下さい。数値にも	らとずいて PWM 出力します。
+ 数値は正転、- 数値は逆転します。	<cpu led-d4="" 基板「hsb8s2268f」の=""></cpu>
④PWM3(左モータ)を出力します。	<cpu j5-08="" 基板「hsb8s2268f」の=""></cpu>
ZとX キーを押して下さい。数値にも	らとずいて PWM 出力します。
+ 数値は正転、- 数値は逆転します。	<cpu led-d5="" 基板「hsb8s2268f」の=""></cpu>
パルス出力確認としてオシロ等で波形計測	リしてみて下さい。

5) 各タスクの Running の状態を見てみます。

更新	▼ 周期便	町			Index352	Sample 512
ask1D	Running	TaskProFile				
0	54	111111111				
1	53	1111111111		mmm		
2	57				1111111111	
3	54					
4	27	1111111	TTTT		TTTTT	
Free	267	Think the state of	nininini	mininin	rin î ninî din	inininininini

①DEF メニュー<データ>-<プロファイル表示>をクリックします。

ト トタスクの Run 状態を見るこ こができます。

6)作者より

本サンプルの利用により、**TOPPERS/OSEK** カーネル下でのユーザーアプリケーションプ ログラムの作成に少しでもお役に立てれば幸いです。



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。

▲ H2268 用のプロジェクトテンプレーを作成しましたので、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	2またはプロジェクトへ 「を保存しますか?	の設定が変更されま	もした。
	ant	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。

プロジェクトの挿入	? 🔀	[6-2-1]
#入	OK	新規プロジェクト
 ・ ・ ・		側チェックにて
		「OK」をクリッ
		クします。

2) プロジェクト名を登録します。

ブロジェクト			
プロジェクトタイプ			
Provide the second seco	プロジェクト名①:		
Library	H2268new		
da obberaloser ichtilbioleer	ディレクトリ(型):		
	C¥toppers_osek¥tools¥h8s-kpitgnu-2268¥H2268ne	参照(<u>B</u>)	
	CPU種別(C):		
	H85,H8/300		
	ツールチェイン①:		
	(

 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「H2268new」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-2268 の下に、

新規プロジェクト名「**H2268new**」のディレクトリを作成します。



②作成した「H2268new」に、「H2268」下の全ファイルをコピーします。





① .¥toopers_osek¥syslib¥h8s-kpitgnu-2268 の下に、

新規プロジェクト名「**H2268new**」のディレクトリを作成します。



②作成した「H2268new」に、「H2268」下の全ファイルをコピーします。

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。

.¥toopers_osek¥tools¥h8s-kpitgnu-2268¥H2268new¥call_sg.bat

のファイルを何らかのエディタで開きます。

<元ファイル> [6-2-5-1]

@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-2268¥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-2268¥H2268new¥Project.sgt

-I..¥..¥..¥sg¥impl_oil -os=ECC2

上記の様に、Project を新規プロジェクト名「H2268new」に変更します。

6)新規プロジェクト「H2268new」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_h8s-kpitgnu-2268」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-2268 に移動します。

'H2268new'プロS	フェクトにファイルを追加 ? 🔀	[6-2-6-1]
ファイルの場所の:	🔁 h8s-kpitenu-2268 🗾 🗢 🗈 📸 📰	<pre>①cpu_config.c</pre>
H2268	H cpu_insnh	<pre>②cpu_support.s</pre>
H2200new Couconfiec H cpu_confie H cpu_contexth H cpu_defsh	H tool_contech H tool_defsh E cpu_supportS E start.S	③start.s
ファイル名(N):	『startS" "cpu_config.c" "cpu_supportS" 追加	の 3 ファイルを「追
ファイルの種類(①)	Project Files ・ キャンセル	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。

②フォルダ名「config_h8s-kpitgnu-2268_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-2268¥H2268new に移動します。

'H2268new'プロジ	ジェクトにファイルを追加 🔹 💽 🔀	[6-2-6-3]
ファイルの場所の:	🔁 H2268new 💽 🖛 🛅 🐨 📰 -	$①$ sys_config.c
sys_config.c H iodefine.h H strh8s2268.h H sys_config.h H sys_defs.h sys_support.S		②sys_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 ・ 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。

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。

'H2268new'プロジ	テクトにファイルを追加]		? 🛛	[6-2-6-4]
ファイルの場所(1):	bernel	ŀ	- 🗢 🗈 🖶 📰	•	(1)alarm.c
Balarmo	Calask_manage.c) task.h			<pre>②event.c</pre>
interrupt.c	H check.h				③interrupt.c
le osctic Cresource.c	비 interrupth 비 osek_kernelh				④osctl.c
🕑 task.c	H resource.h				⑤resource.c
ファイル名(N):	"task_manage.c" "ala	rm c‴event.c″″i	iterrupt.c"	追加	<pre>⑥task.c</pre>
ファイルの種類(工):	Project Files		- +	やンセル	\textcircled{T} task_manage.c
	▼相対パス(B)	□ 登録済みファ	イルを非表示(P)		の7ファイルを
					「追加」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。

④フォルダ名「syslib_h8s-kpitgnu-2268」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-2268 に移動します。

'H2268new'ブロ:	フェクトにファイルを追加 🕜 🔀	[6-2-6-5]
ファイルの場所の:	🔁 h8s-kpitenu-2268 📃 🗢 🗈 📸 📰	①sys_timer.c
H2268	H sys_serialh	②sys_serial.c
Cosek_lib.c Cosek_lib.c Cosek_serial.c Cosys_timer.c Hosek_lib.h	H sys_timer.n	③osek_lib.c
		の3ノアイルを「迫加」」ます
ファイル名(12):	"sys_timer.c" "osek_lib.c" "sys_serial.c")Bto	加] しより。
ファイルの種類(①):	Project Files ・ キャンセル	
	☑ 相対パス(B) □ 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-2268_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-2268¥H2268new に移動します。

'H2268new'ブロジェクトにファイルを追加 🕜 🔀	[6-2-6-6]
ファイルの場所①: Carl H2268new 💽 🗢 🖆 🖽	①hw_config.c
Chw.config.c H) hw.config.h	<pre>②hw_led.c</pre>
Chw.pwm.c H)hw.pwm.h	③hw_pwm.c
Chw_serialc Hhw_serialh	<pre>④hw_serial.c</pre>
Mw_sys_timer.c Hhw_sys_timer.h	⑤hw_sw.c
ファイル名(N): 「hw_sys_timer.c" "hw_config.c" "hw_led.c" "hw_pwr 追加	$6 hw_sys_timer.c$
ファイルの種類①: Project Files ・ キャンセル	の 6 ファイルを「追
✓ 相対パス(B) 「登録済みファイルを非表示(P)	加」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「H2268new」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)	
rchives Input Output List Sections Other Command	file
Archive search directories:	
c¥program files¥renesas¥hew¥tools¥kpit cummins¥gnush-el	Add
c-sprogram messrenesassnewstooisskipit cumminssgnusri-e	Remove
	Move up
< >	Move down
Archive (library) files:	
liboptc.a	Add.
liboptma libecc.a	Remove
	Mgve up
	Move down
Search archives repeatedly for undefined references	
✓ Use Optimised Libraries	
Common options:	
"\$(TCINSTALL)Vsh-elfVsh-elfVlibVm2" -o "\$(CONFIGDIR) V\$(PROJECTNAME)x"oformatistrec -Map "\$(CONFIGDIR) V\$(PROJECTNAME).map" -e_start -I "opto" -I "optm" -I "opt	xc‴
Show option description QK	キャンセ

<Input>タグをクリックします。

starto	GNU	althew4¥toppers¥top	-	<u>M</u> 00.
Imaino	GNU	g¥hew4¥toppers¥top_		Remove
]osek_lib.o	GNU	q¥hew4¥toppers¥top_		
cpu_config.o	GNU	q¥hew4¥toppers¥top_		Disable
cpu_support.o	GNU	q¥hew4¥toppers¥top_		Modify
sys_config.o	GNU	q¥hew4¥toppers¥top		moony_
sys_support.o	GNU	q¥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_	~	
1		>		

[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]
\$(CONFIGDIR)¥start.o OK Cancel	
▲ 変更が必要な場合は、「h2268new」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-2268_Project_appsrc」

ディレクトリ「.**¥toppers_osek¥tools¥h8s-kpitgnu-2268¥H2268new¥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-2268¥H2268new**」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 っておりますので、皆様の協力を御願い申し上げます。

2008年10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