

【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

2008/10/02

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	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(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.....	15
3-4.	H-debugger でのプロファイル表示(DEF 7.00B 以上).....	18
第 4 章	ハード構成およびシステム構成.....	19
4-1.	ハード構成.....	19
4-2.	システム構成.....	20
4-2-1.	システムブロック図 (サンプル) .....	20
4-2-2.	CAT261 のプログラムメモリ MAP .....	21
4-2-3.	プログラムサイズの詳細 MAP .....	21
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.....	22
第 5 章	サンプルアプリケーションの準備.....	24
5-1.	Hew での準備 (ルネサス製) .....	24
5-2.	H-debugger(DEF.exe)での準備 (Aone 製) .....	29
5-3.	サンプルアプリケーションを走らせる前の準備 .....	32
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.....	34

---

---

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
<b>第7章 備考.....</b>	<b>44</b>
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るように努めます。Mail: [cat-i@aone.co.jp](mailto: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/OSEK カーネル外部仕様書                  | ----- | 株式会社ヴィッツ製   |
| 2) TOPPERS/OSEK カーネル SG 取扱説明書              | ----- | 株式会社ヴィッツ製   |
| 3) 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   |
| 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

## 第 2 章 TOPPERS/OSEK カーネルの開発階層

¥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	// <b>プロジェクト 1</b>
¥appsrc	// サンプルソース [* .c]
¥debug	// オブジェクト [* .abs]
¥h8s2612st	// <b>プロジェクト 2</b>
¥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
	// テンプレート

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個の割り込み処理

- ① システムタイマー                    TPU0(TGR0A)1ms 割り込み    SysTimerInt()
- ② 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                        TEI\_1                        TxHwSerialInt1()

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

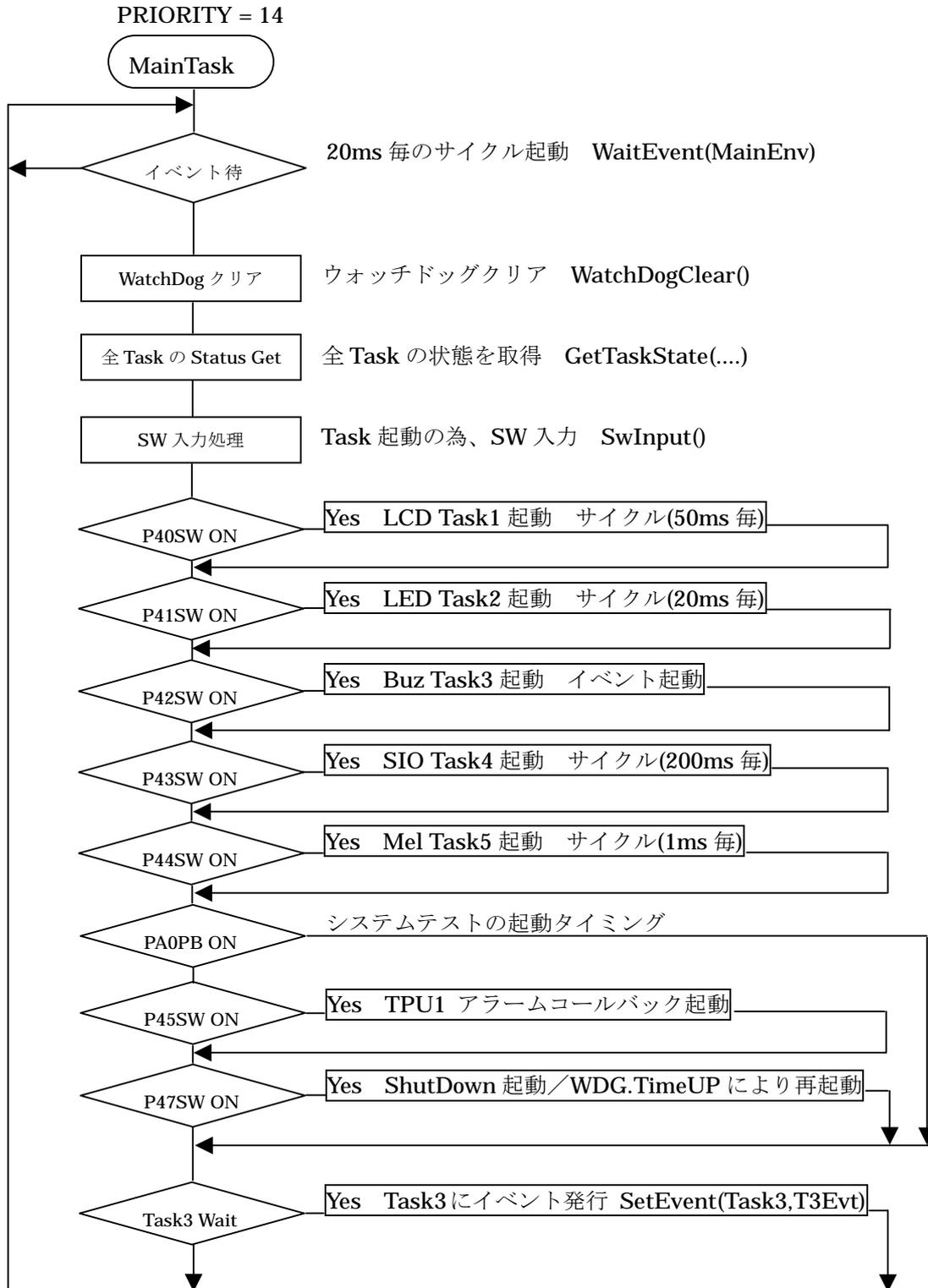
9) エラーフックルーチン

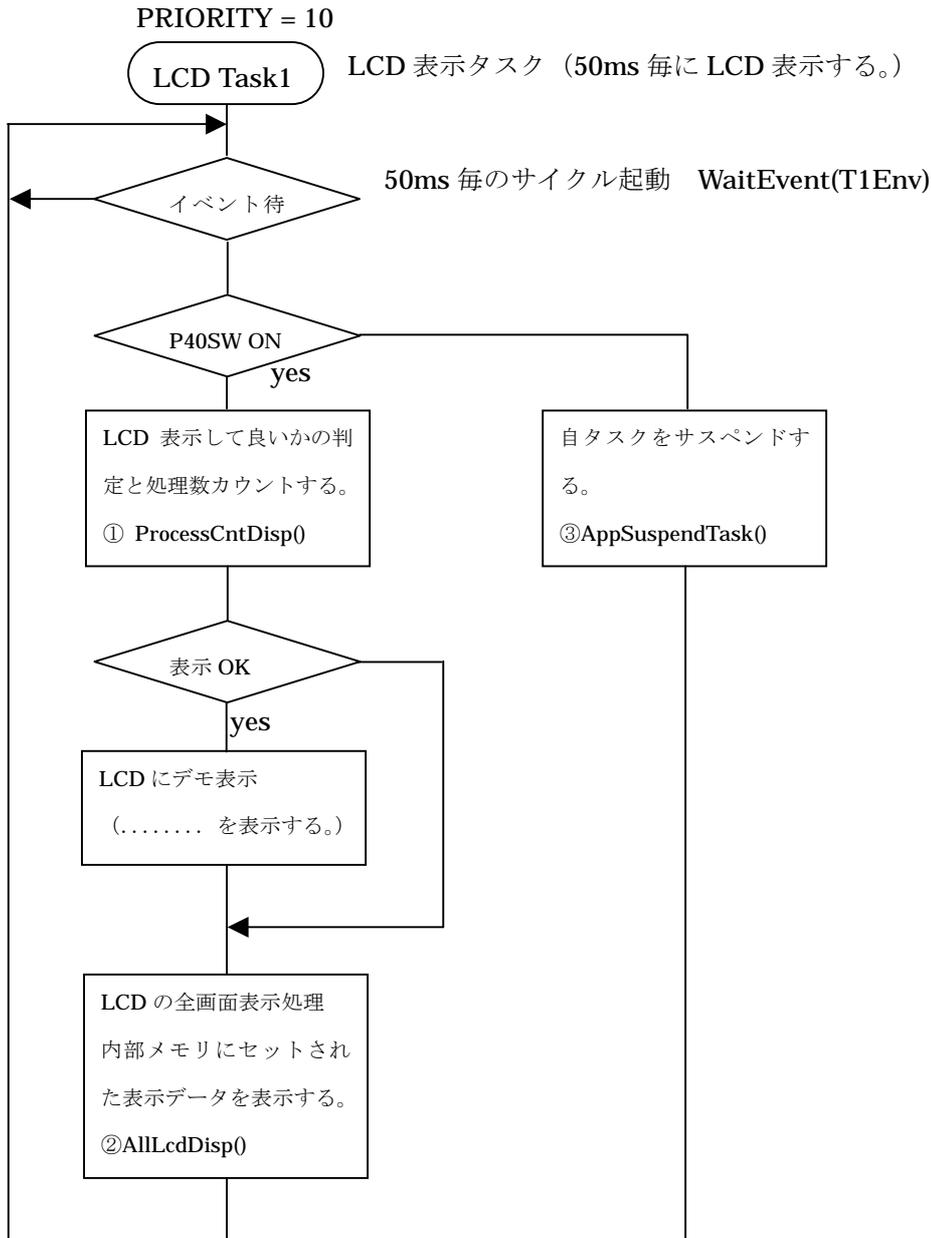
ErrorHook

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)





① ProsessCntDisp()

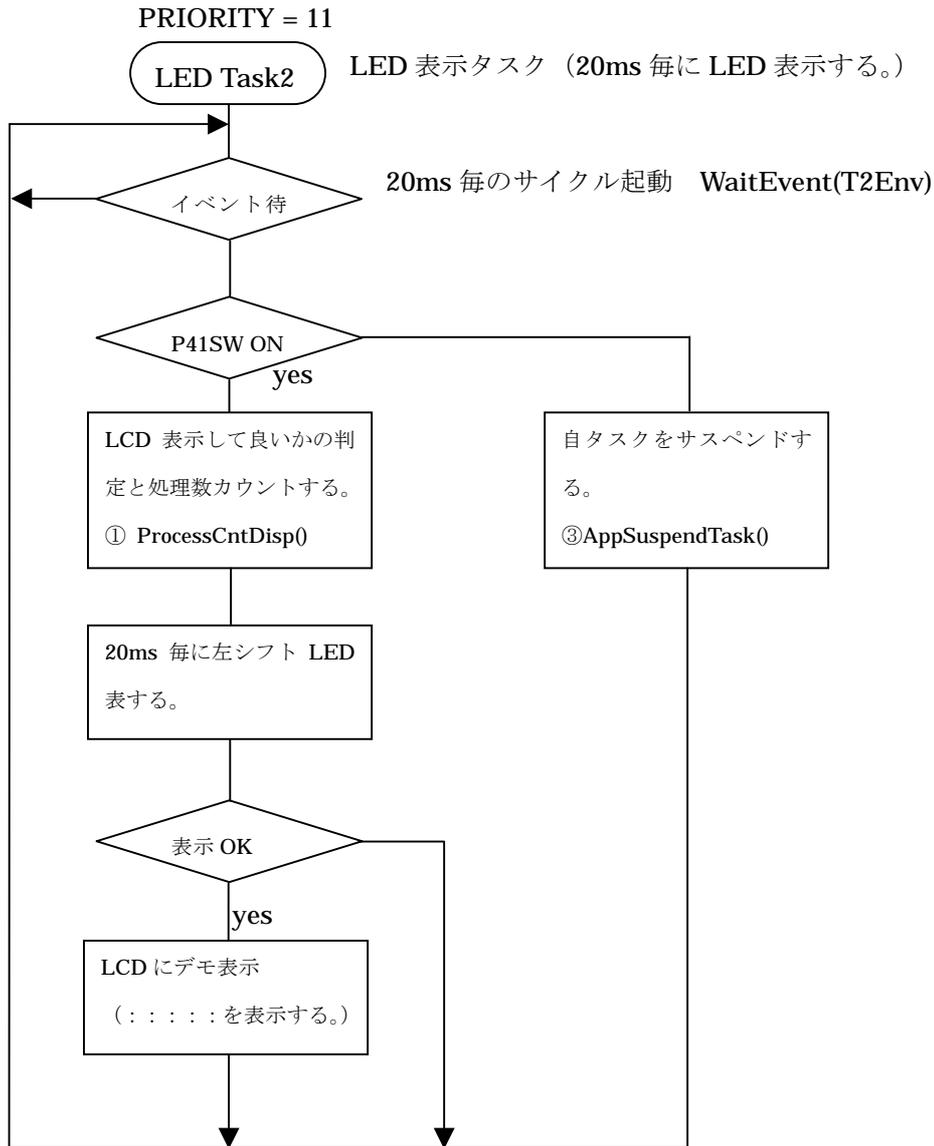
現在のアクティブ Task と自 Task の関係で、LCD デバイスを使用して良いか判断し、かつ、処理ループ数をカウント表示する。(Task1:10000~19999)

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()

自タスクのサスペンド情報を表示後、TerminateTask()を呼ぶ。

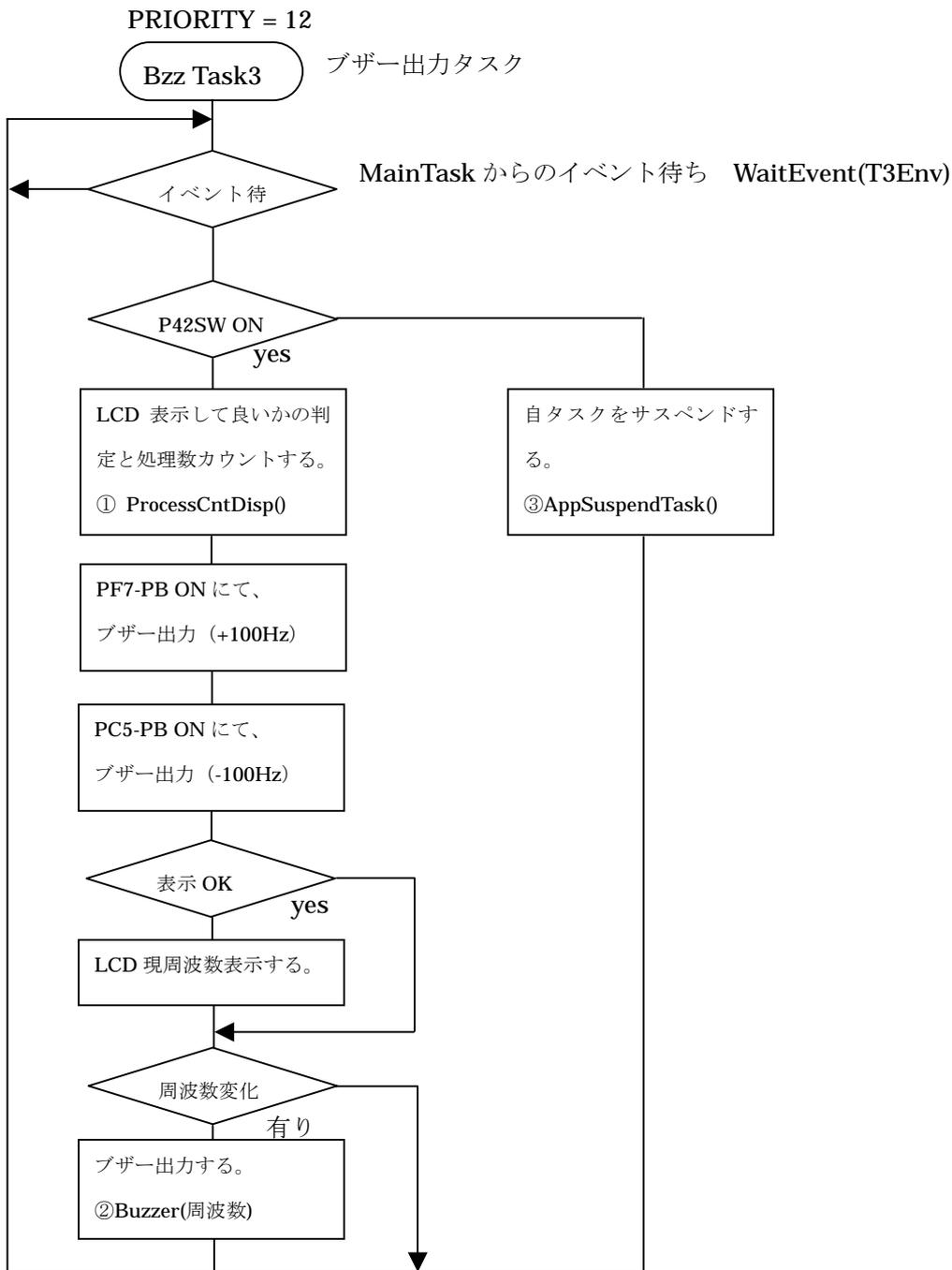


① ProsessCntDisp()

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()

自タスクのサスペンド情報を表示後、TerminateTask()を呼ぶ。



① ProsessCntDisp()

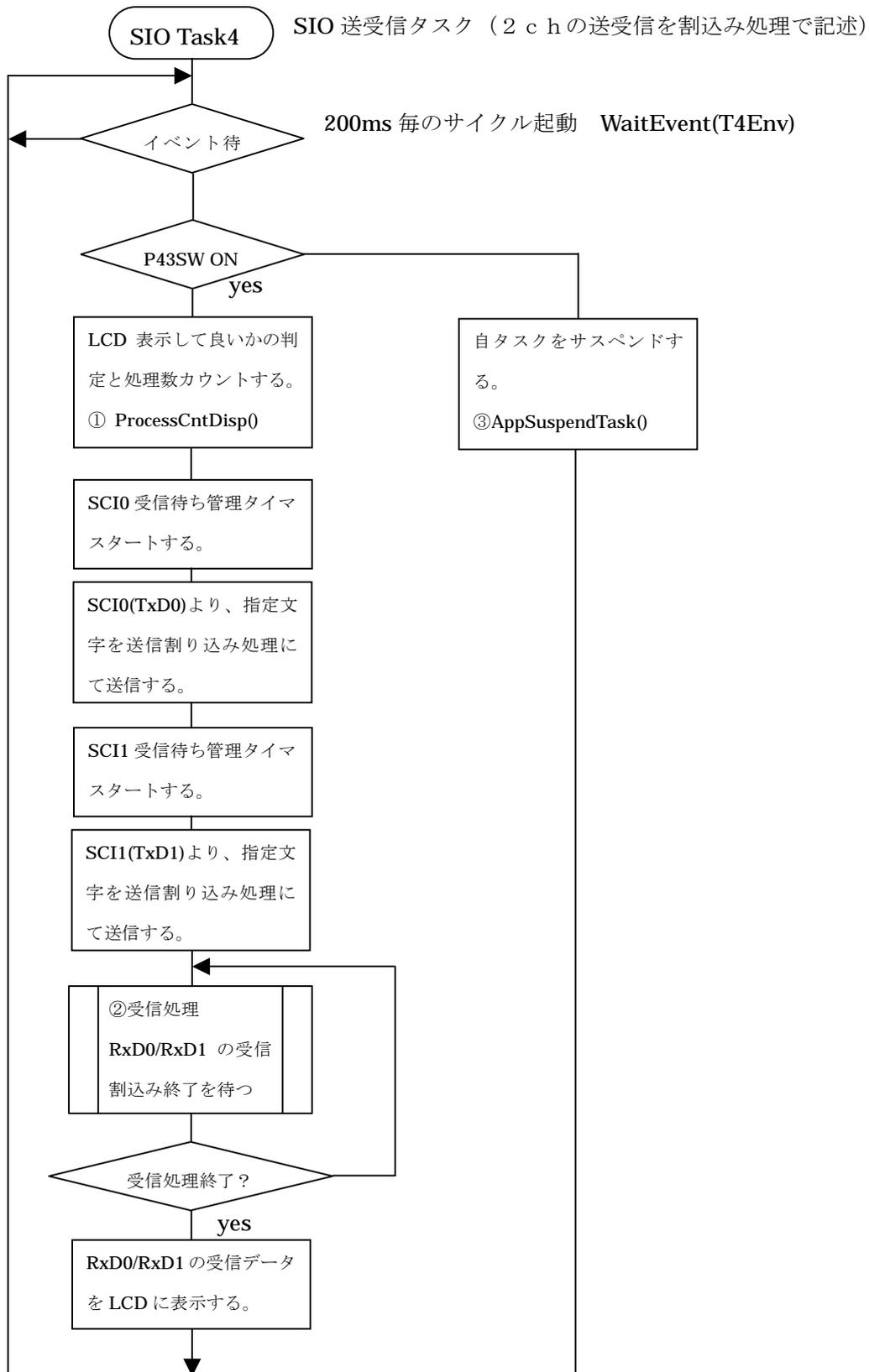
現在のアクティブ Task と自 Task の関係で、LCD デバイスを使用して良いか判断し、かつ、処理ループ数をカウント表示する。(Task3:30000~39999)

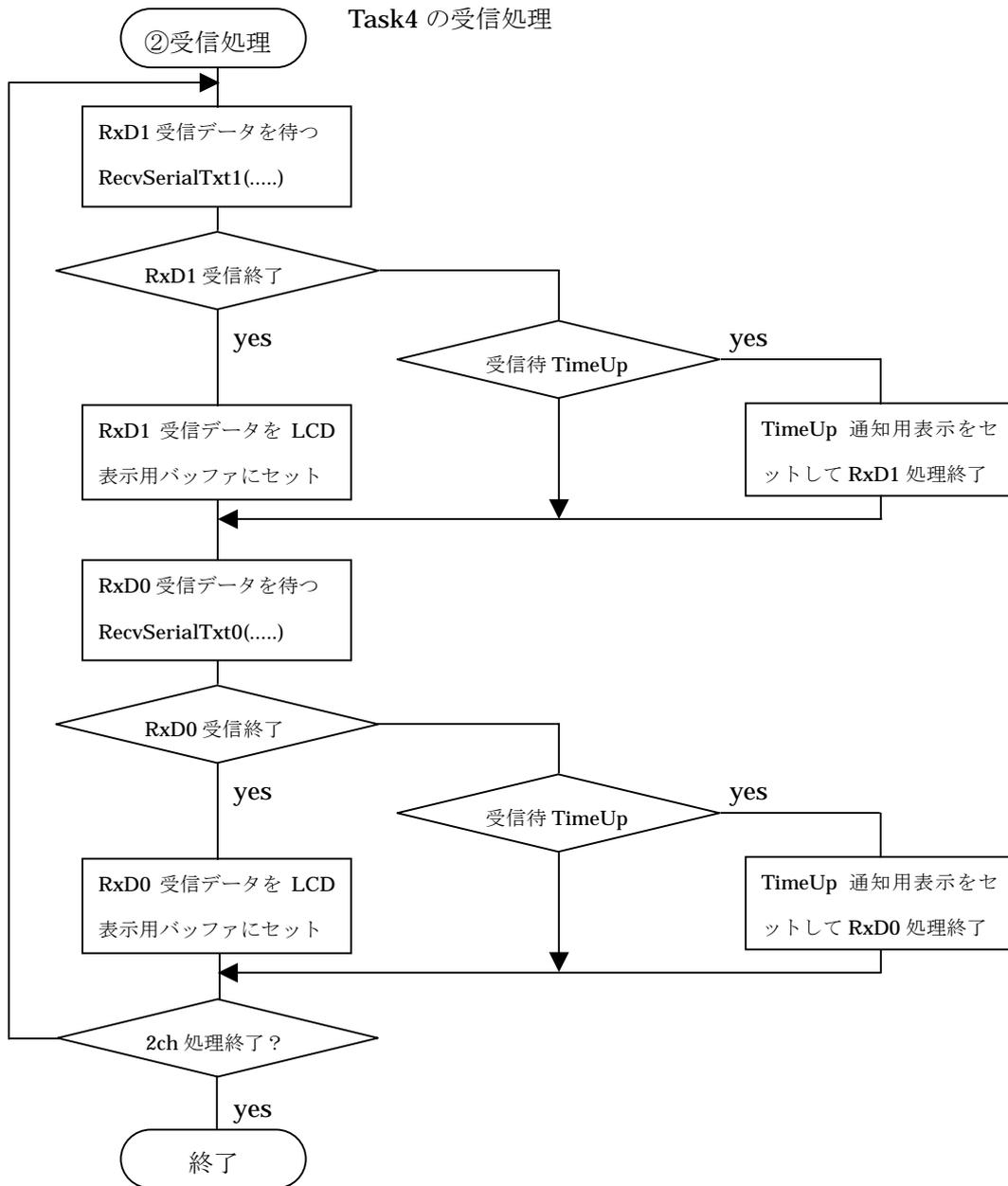
② Buzzer()

TPU1 から、Buzzer に対して指定周波数を出力する。

③ AppSuspend()

自タスクのサスペンド情報を表示後、TerminateTask()を呼ぶ。





① ProsessCntDisp0

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)

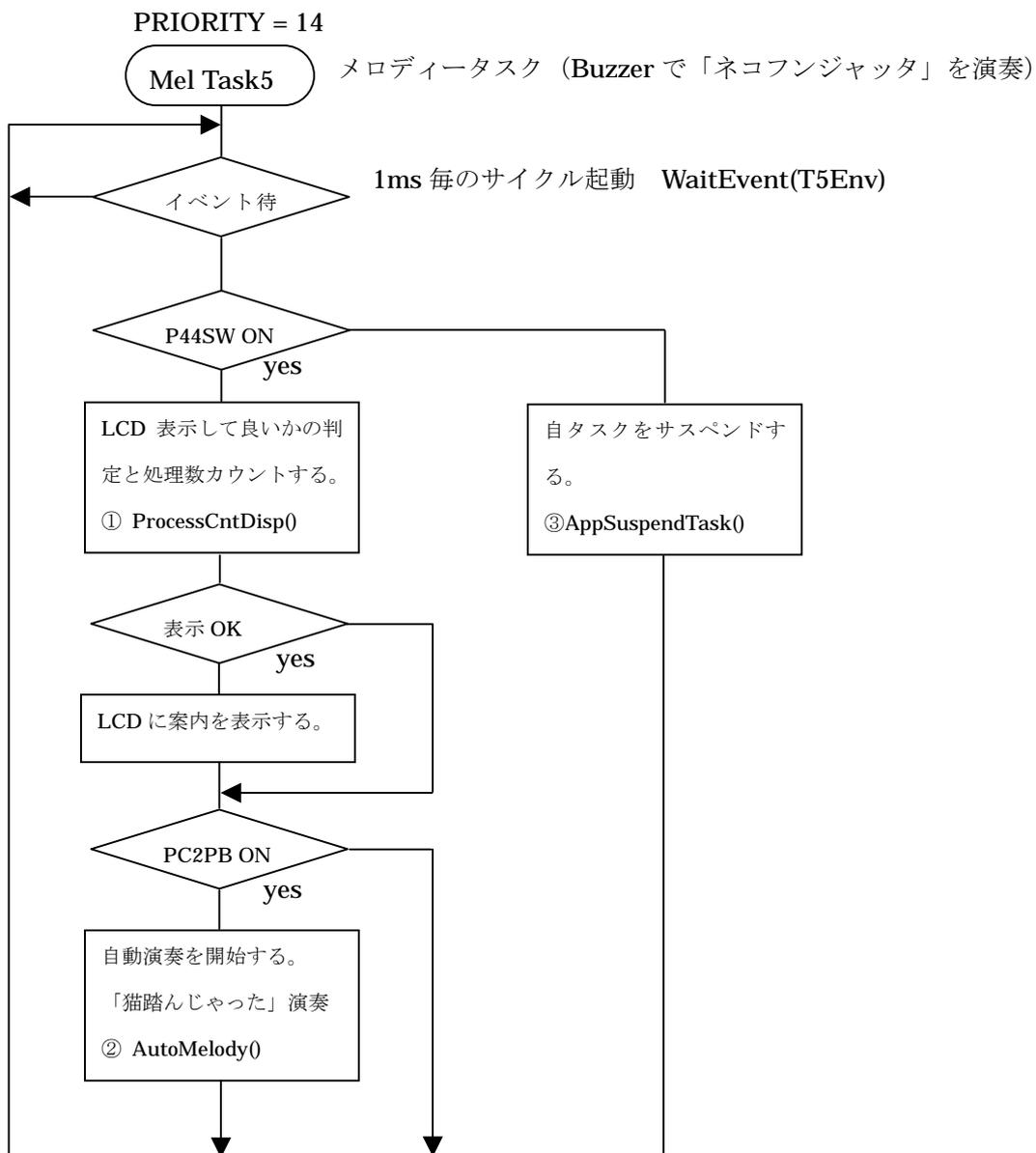
② INT8 \*txt = {"AHE-ToppersCD1 for TOPPERS/OSEK by Aone"}を各 Ch から 8 文字づつリング方式により送信し、受信した文字を LCD に表示する。

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 AppSuspend0

自タスクのサスペンド情報を表示後、TerminateTask()を呼ぶ。



① ProsessCntDisp()

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()

自タスクのサスペンド情報を表示後、TerminateTask()を呼ぶ。

### 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-debugger 用 リセット遅延の為、20msWait
    ; Target CAT261-20MHz
wait20ms:
    mov.w    #20,r0
waitrst:
    bsr      _SOFT1MS
    dec.w    #1,r0
    bne     waitrst
    rts

;/*SOFT1MS() 1ms ソフトタイマー (20.0000MHz) Non Wait */
_SOFT1MS:
    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 仕様でのデバッグ処理は、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

/*****/
/*   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;
    IPRC  = 0;
    IPRD  = 0;
    IPRE &= 0xf0;                /* <---- PBC                               */
    IPRF  = 0;
    IPRG  = 0;
    IPRH  = 0;
    IPRJ  = 0;
    IPRK  = 0;
    IPRM  = 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。割り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。

使用する割り込み要素は、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
```

① 「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;
```

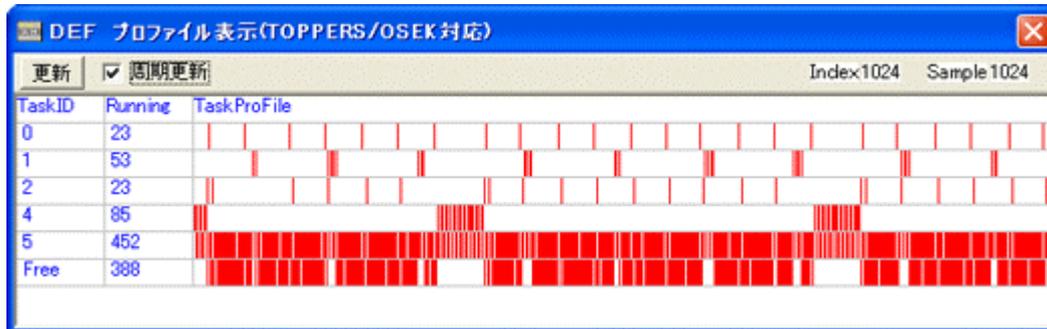
④ システムタイマー割り込みのタイミングで現 Running 中の TaskID を「\_TaskProFile」に順次記憶していきます。(リングバッファ)

⑤ サンプルング場所は、「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。

[3-4-1]表示例



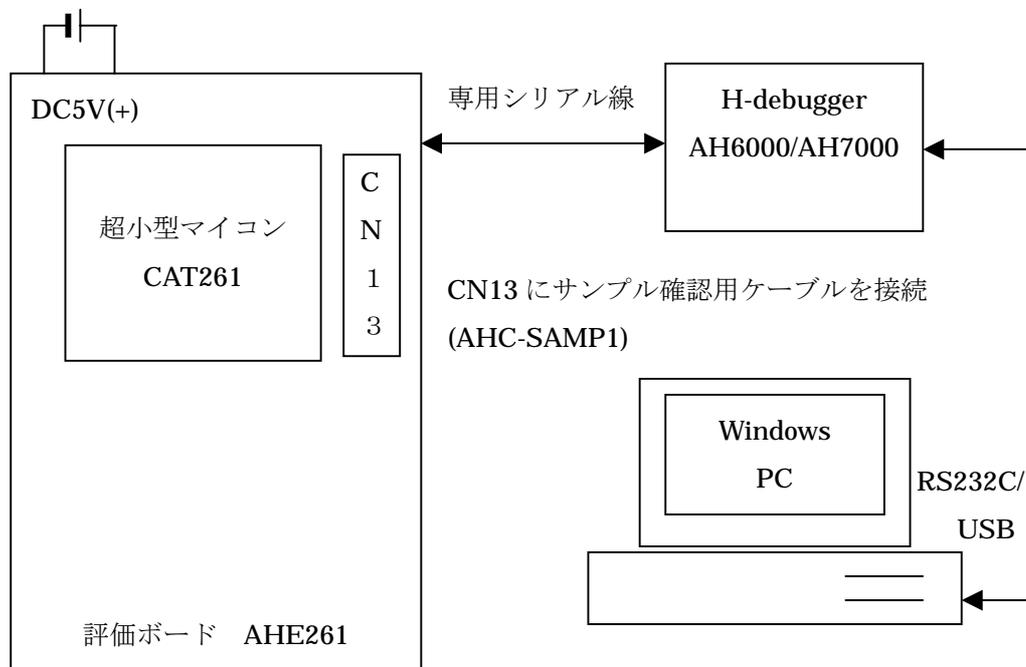
「更新」PB をクリックしますと最新データを表示します。

「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。

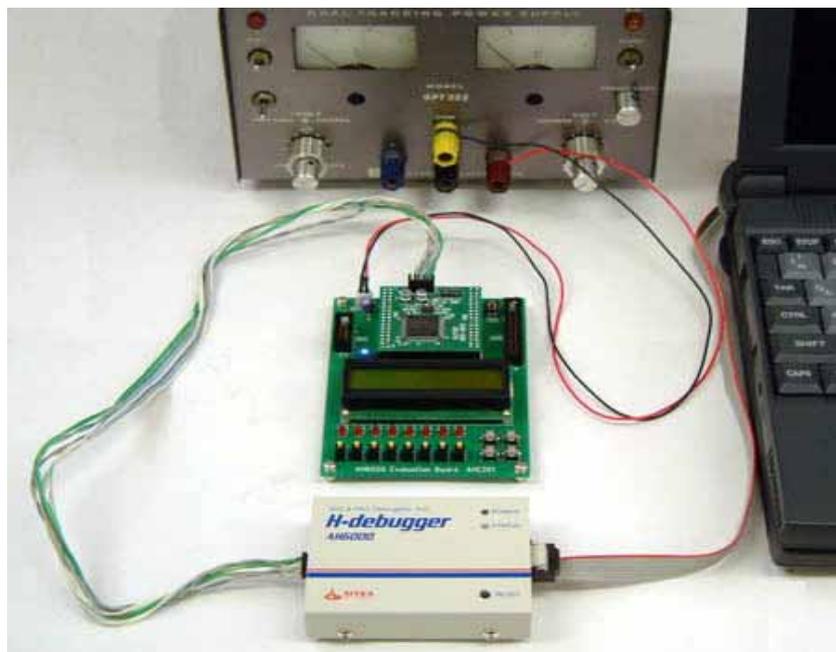
## 第4章 ハード構成およびシステム構成

### 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]



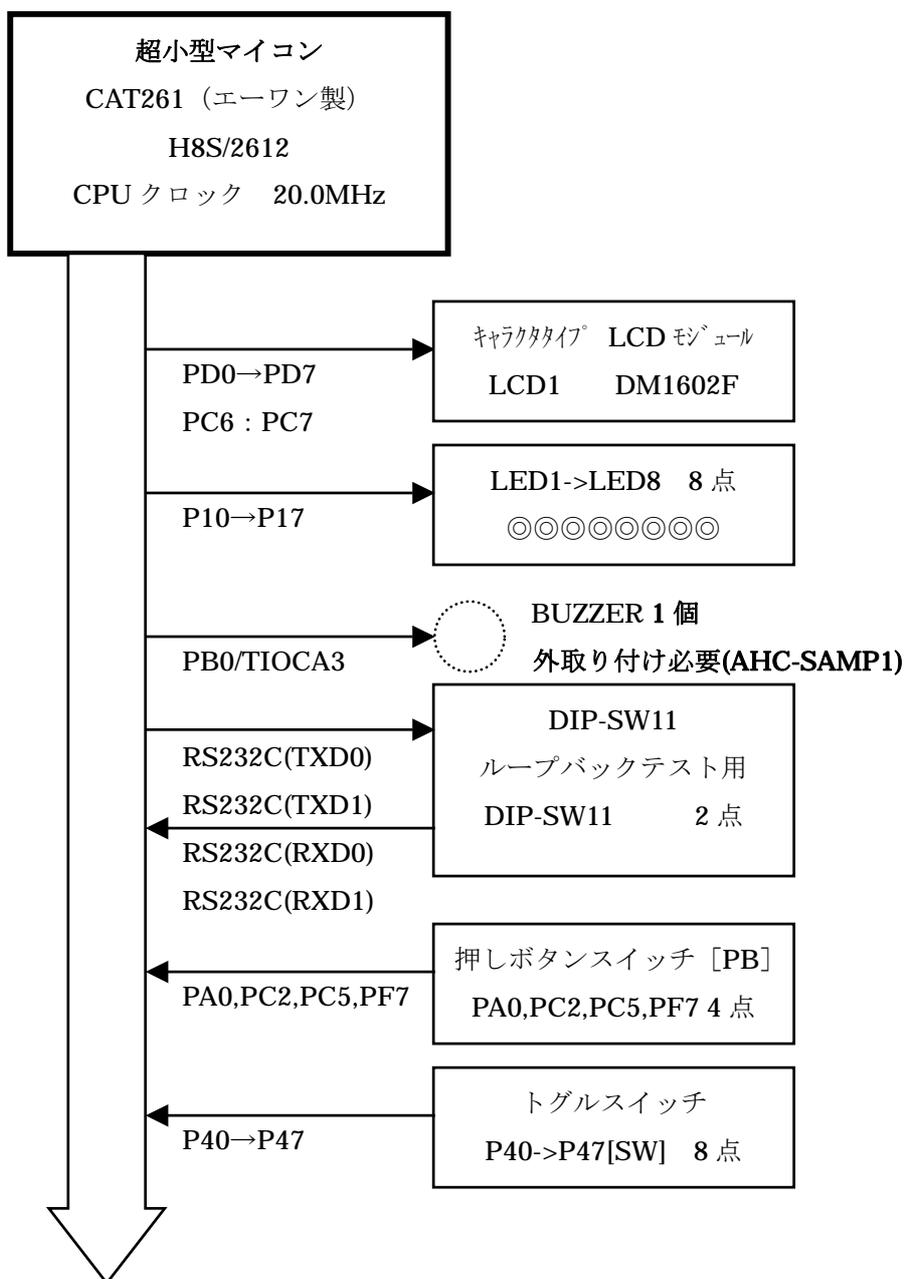
## 4-2. システム構成

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



<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

#### 4-2-3. 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 (main....etc)
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

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

SW11 の設定		ON		OFF	
SW11-1		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					
ポートアドレス	ポートシンボル	ピン番号	ピンシンボル	方向	信号名
0xffff0c	PDDR	CN1-7B	PD0	出力	D0 LCD-module
		1-7A	PD1	出力	D1
		1-6B	PD2	出力	D2
		1-6A	PD3	出力	D3
		1-5B	PD4	出力	D4
		1-5A	PD5	出力	D5
		1-4B	PD6	出力	D6
0xffff0b	PCDR	CN2-9A	PC6	出力	RS LCD-module
		-9B	PC7	出力	E LCD-module
					未使用
		-14A	P07		未使用

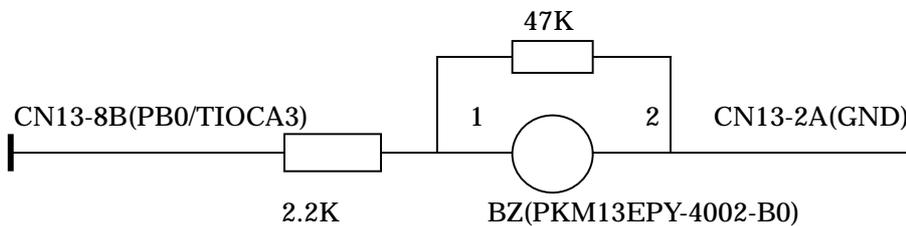
LED関係					
ポートアドレス	ポートシンボル	ピン番号	ピンシンボル	方向	信号名
0xffff00	P1DR	CN1-14A	P10	出力	LED1
		1-14B	P11	出力	LED2
		1-15A	P12	出力	LED3
		1-15B	P13	出力	LED4
		1-16A	P14	出力	LED5
		1-16B	P15	出力	LED6
		1-17A	P16	出力	LED7
		1-17B	P17	出力	LED8

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	ピンシンボル	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					
ポートアドレス	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					
ポートアドレス	ポートシンボル	ピン番号	ピンシンボル	方向	信号名
0xffffb3	PORT 4	CN1-13B	P40	入力	SW16 SW[P40]
		1-13A	P41	入力	SW17 SW[P41]
		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]

外部ブザー回路図 (型式 AHC-SAMP1)別売りで用意しております。



## 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

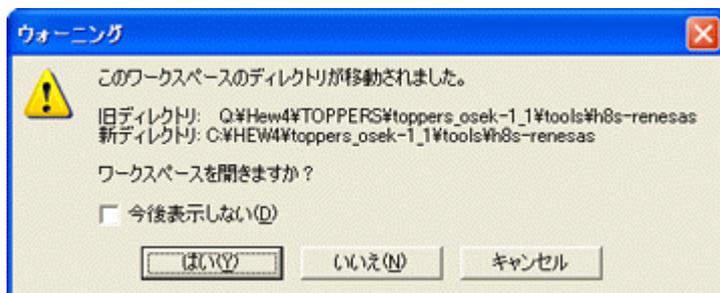
上記ディレクトリ下の全ファイルを、OSEK カーネルにディレクトリごと全コピーして下さい。

4) Hew を起動します。

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

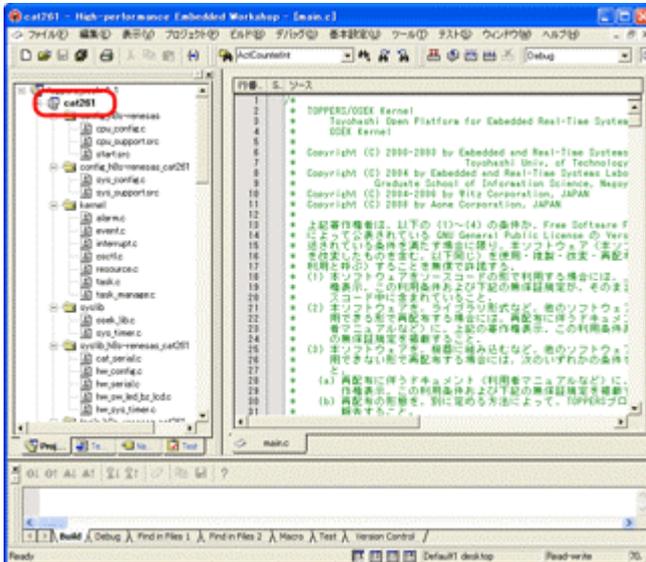
5) Hew メニューの<ファイル>-<ワークスペースを開く>でワークスペースを開きます。

- ・ ".¥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。



[5-1-1]

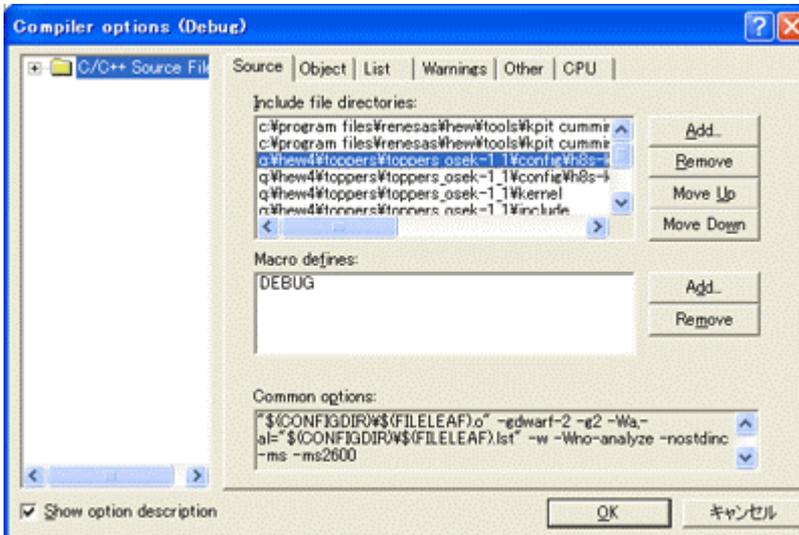
- 6) アクティブプロジェクトが「**cat261**」になっていることを確認します。**cat261** になっていない場合は、**Hew** メニューの<プロジェクト>-<アクティブプロジェクトに設定>で「**cat261**」を指定して下さい。



[5-1-2]

- 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。

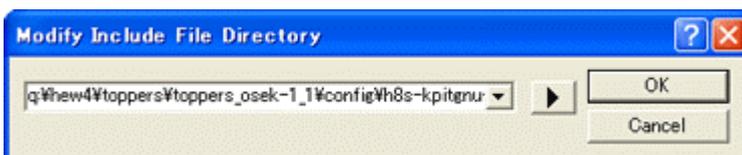
**Hew** メニューの<ビルド>-<Compiler>をクリックして下さい。



[5-1-2-1]

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に変更して下さい。

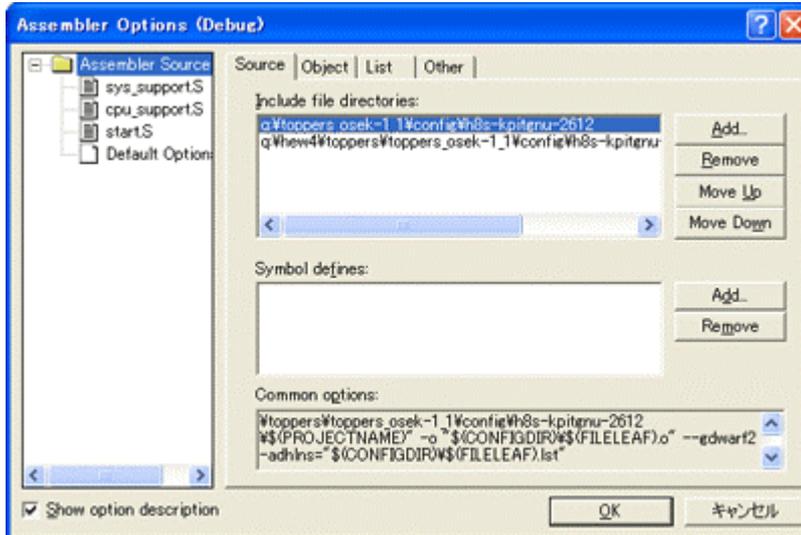
マウスでのダブルクリックで変更できます。



[5-1-2-2]

<Brows>指定が早いと思います。

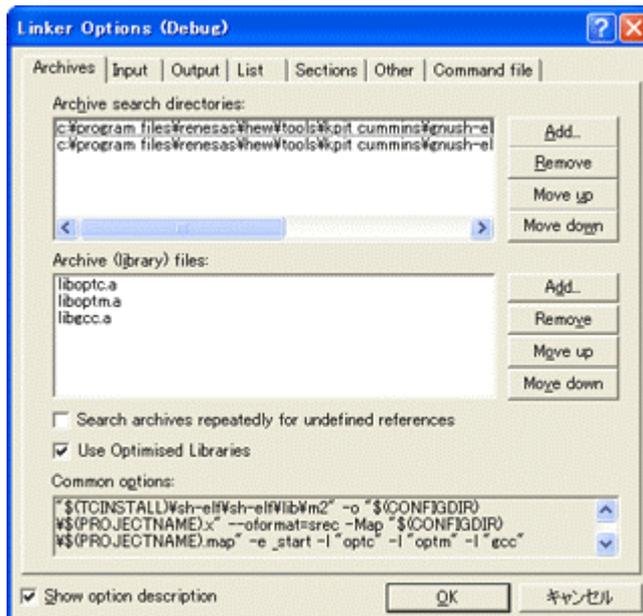
Hew メニューの<ビルド>-<Assembler>をクリックして下さい。



[5-1-2-3]

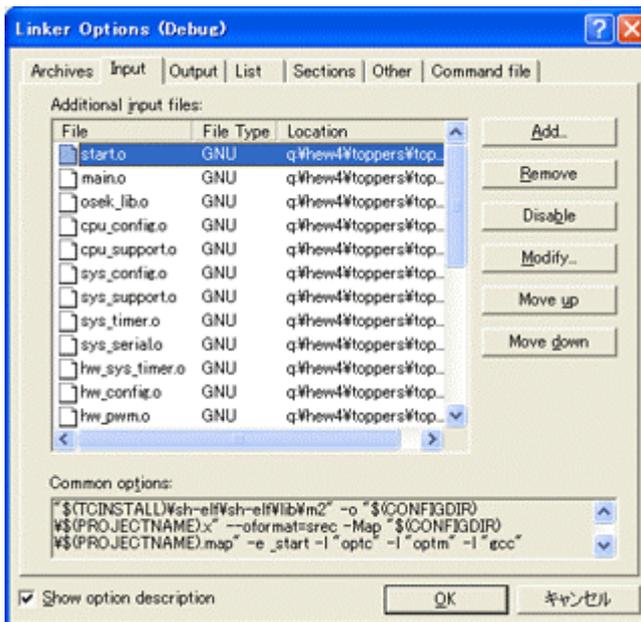
全てのディレクトリを現ワークスペースのディレクトリに変更して下さい。

Hew メニューの<ビルド>-<Linker>をクリックして下さい。



[5-1-2-4]

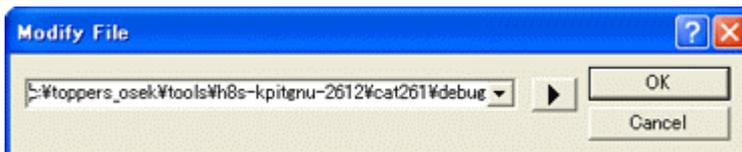
<Input>タグをクリックします。



[5-1-2-5]

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。

マウスでのダブルクリックで変更できます。



[5-1-2-6]

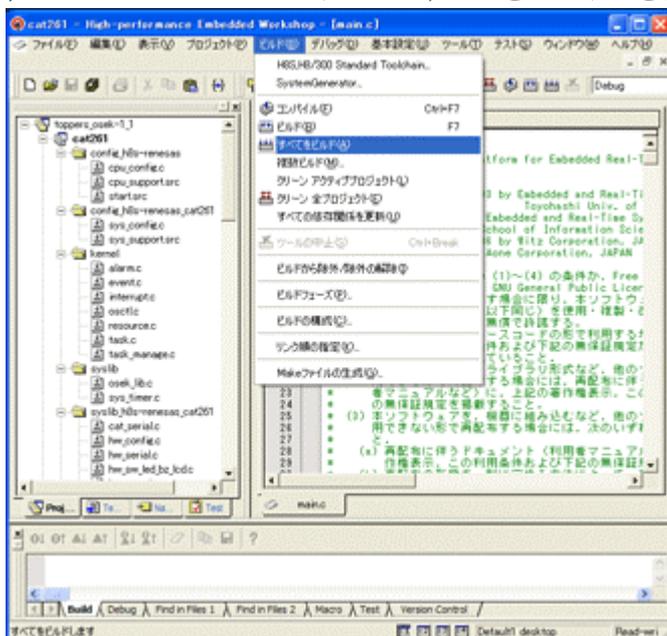
- 8) アクティブプロジェクトが「h8s2612st」に変更して、7) 項と同じく<Compiler>と<Assembler>と<Linker>のディレクトリを変更して下さい。

 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。

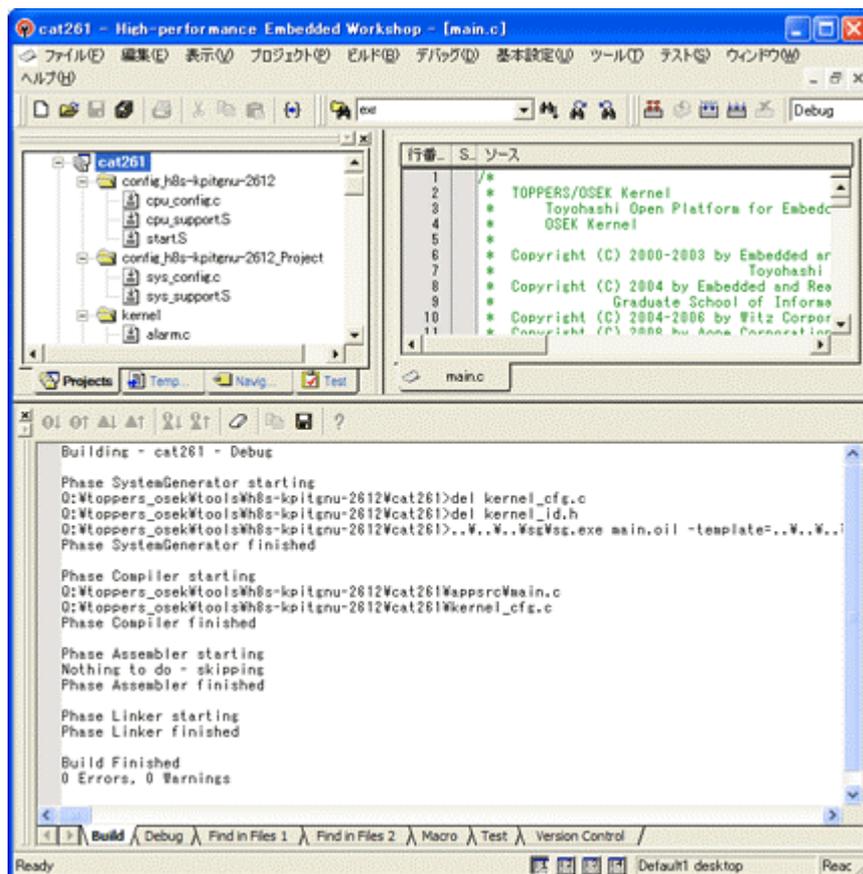
1 0) アクティブプロジェクトを「cat261」に戻します

1 1) 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。



[5-1-3]

1 2) ビルド結果

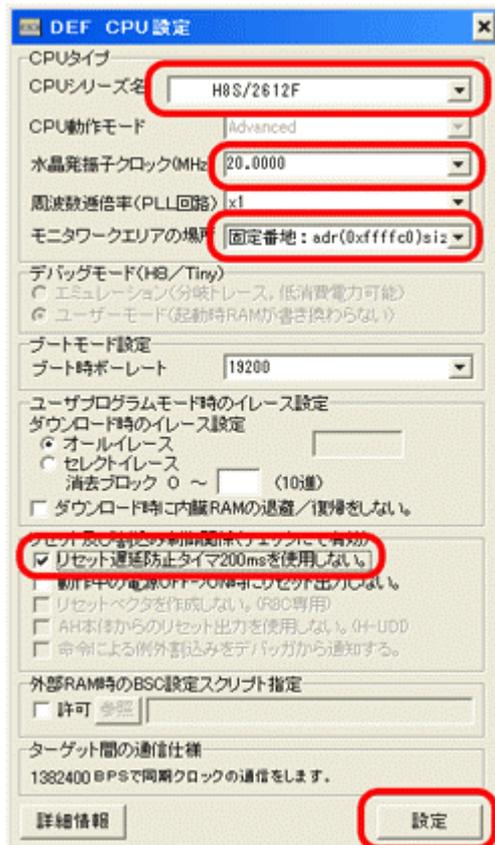


[5-1-4]

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。



[5-2-1]

CPU シリーズ名 : H8S/2612F

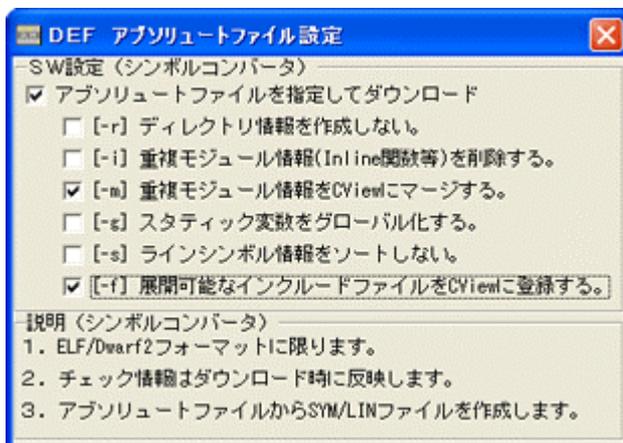
クロック (MHz) : 20.0000

モニターワークエリアの場所 : 固定番地

リセット遅延防止タイム使用しない : チェック

最低上記4項目を設定後、「設定」をクリック  
します。

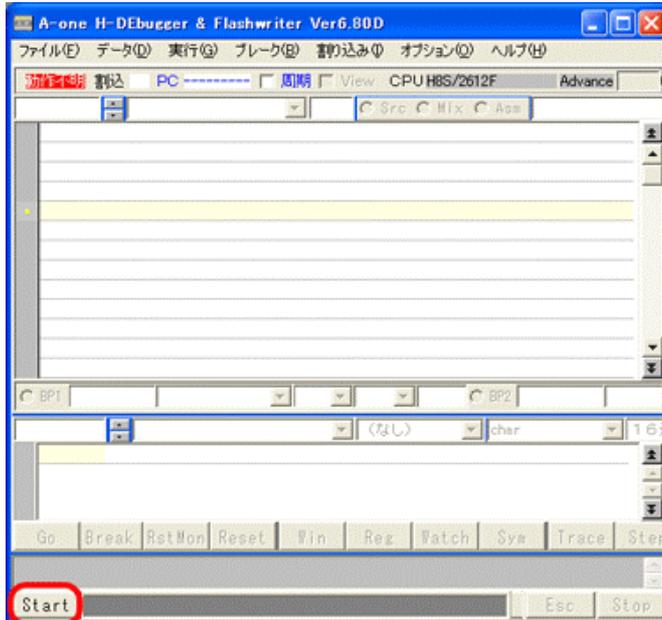
- 4) DEF メニュー<ファイル>-<アブソリュートファイル設定>をクリックします。



[5-2-2]

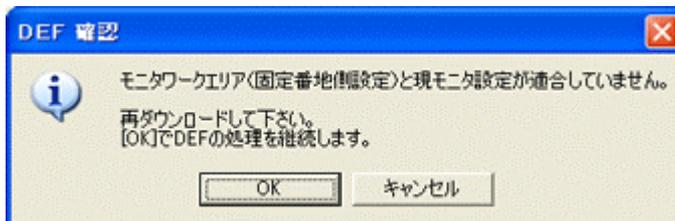
左図のように 3 箇所にチェックしま  
す。

- 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)



[5-2-3]

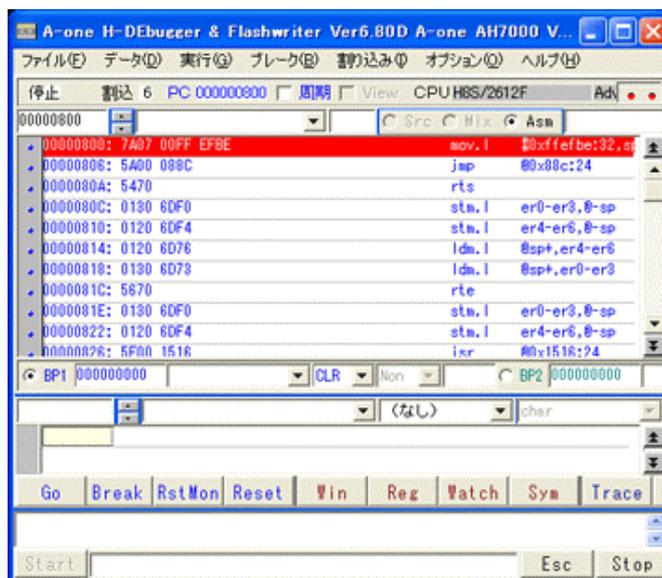
- 6) 評価キット(AHE261/CAT261)に登録してあるプログラムによっては、下図の様な確認画面が表示されますが無視して「OK」をクリックして下さい。



[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。

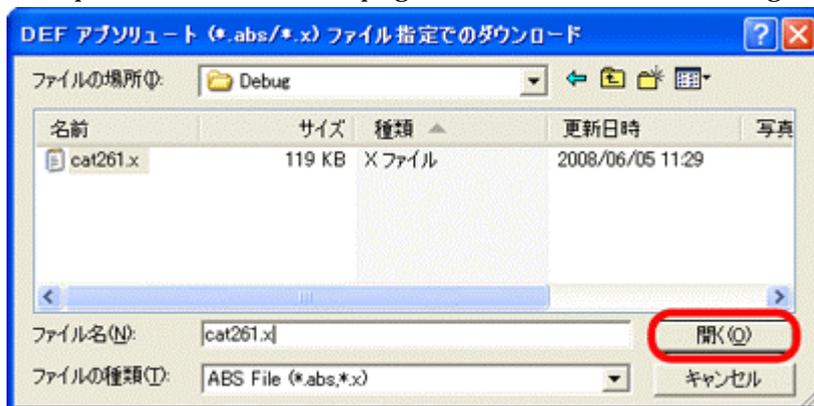
- 7) ターゲット側と正常な通信を確立しますと下図の様な画面になります。



[5-2-5]

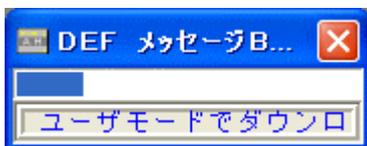
8) DEF メニュー<ファイル>-<ダウンロード>をクリックします。

.¥toopers\_osek¥tools¥h8s-kpitgnu-2612¥tools¥cat261¥debug まで降ります。



[5-2-6]

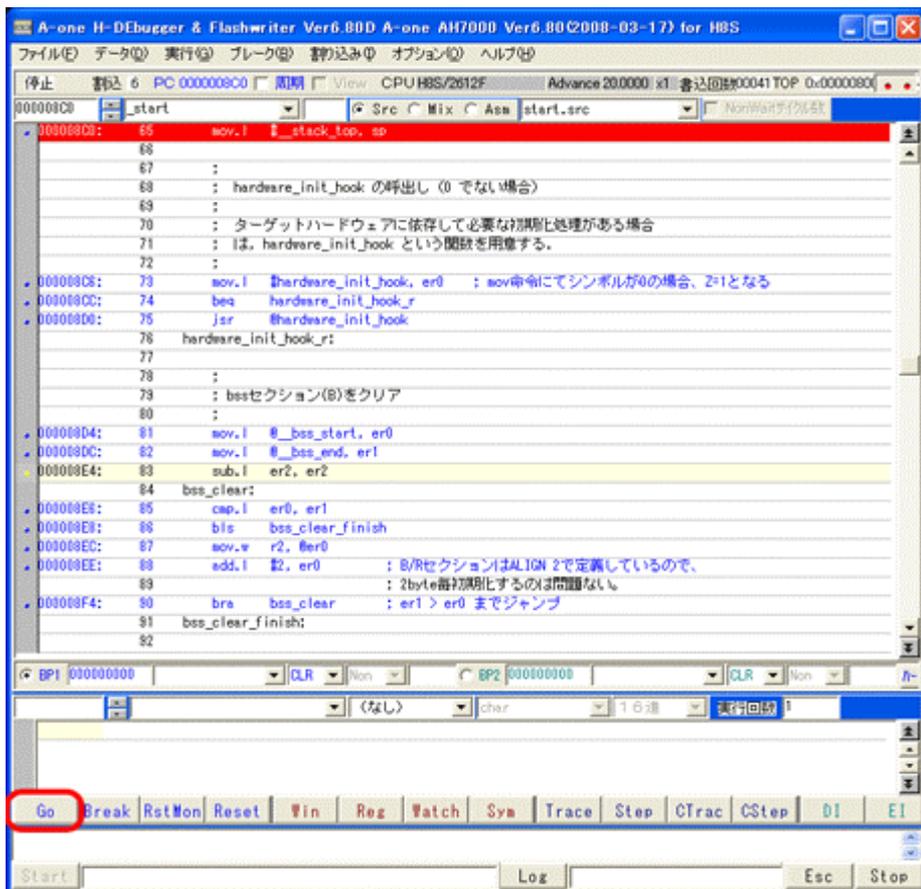
cat261.x を選択後  
「開く」をクリック  
します。



[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。



[5-2-8]

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 for CAT261[P4xS]	[5-4-1]
--------------------------------------	---------

(2) 評価ボードの SW-P40 を ON(上側)にする。

LCDTask1 [xxxxx] .....	[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]	[5-4-5]
AHE-ToppersCD1 f	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 にしてみてください。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]	[5-4-6]
ネフンジャッタ PC2 {S}	MelTask5 が起動します。

PC2-PB を ON しますと、ブザーより「ネフンジャッタ」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	[5-4-8]
E_OK	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

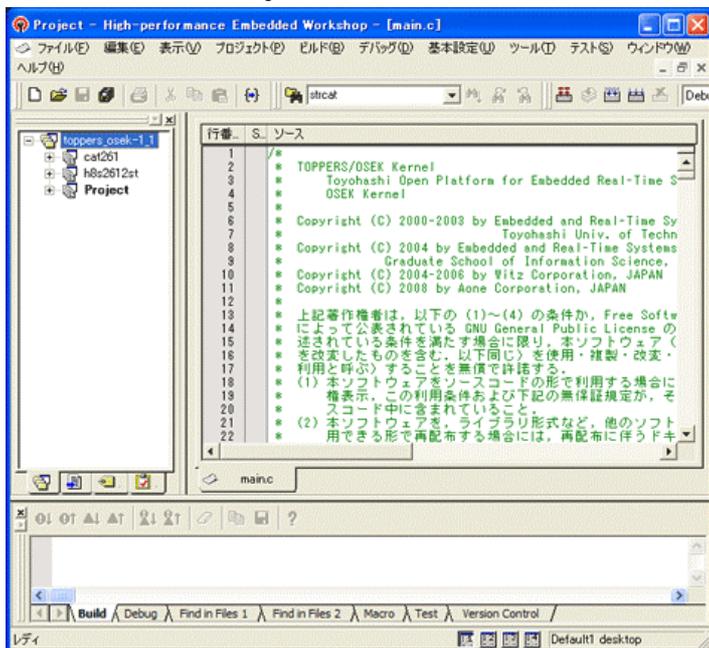
### 6-1. プロジェクトタイプの作成

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。

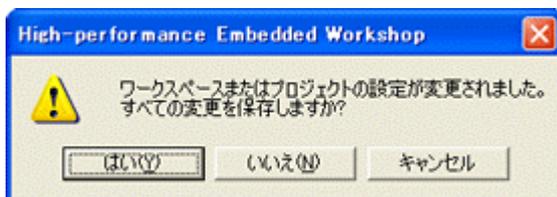
**(必ず Compiler] と Assembler と Linker のディレクトリが変更済みである事!!)**

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。



[6-1-1]

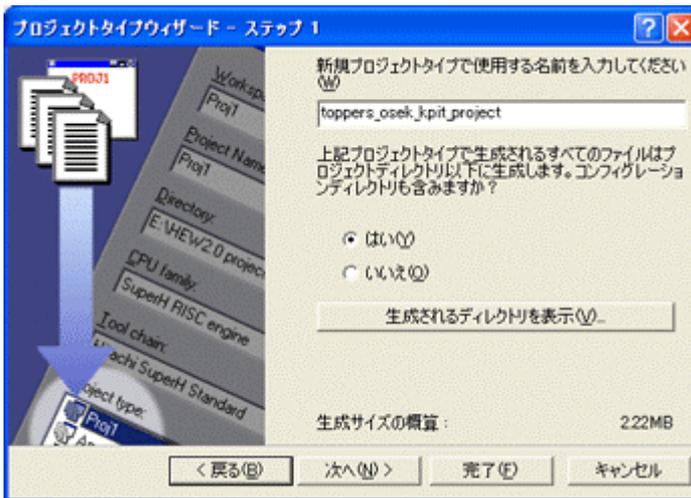
2) Hew メニュー<プロジェクト>-<プロジェクトタイプの作成>をクリックします。



[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。

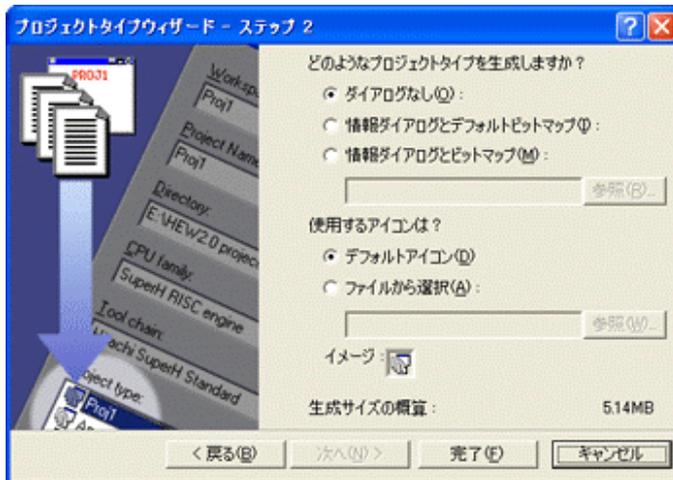


[6-1-3]

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。



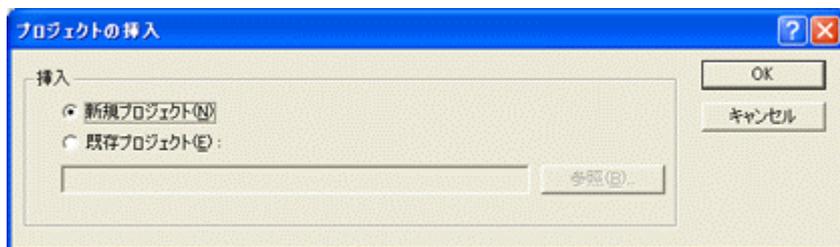
[6-1-4]

左図の様に「デフォルト」のままで、

「完了」をクリックします。

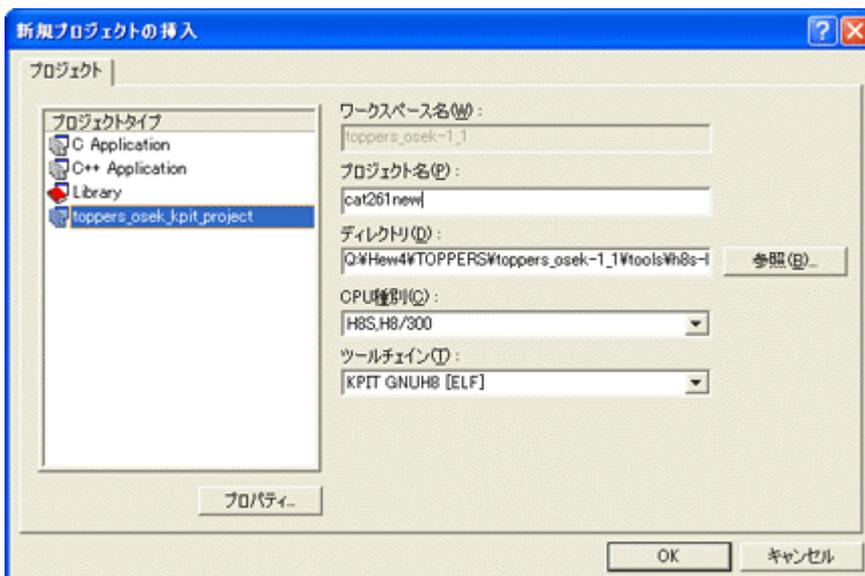
6-2. 新規プロジェクトを登録します。

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。



[6-2-1]  
新規プロジェクト側チェックにて「OK」をクリックします。

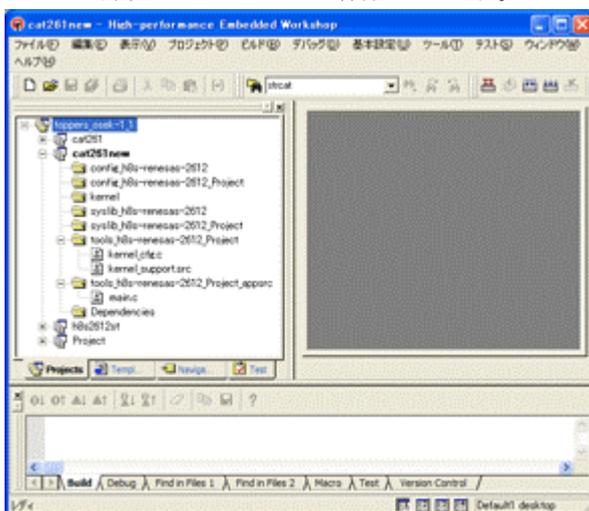
2) プロジェクト名を登録します。



[6-2-2]

- 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。



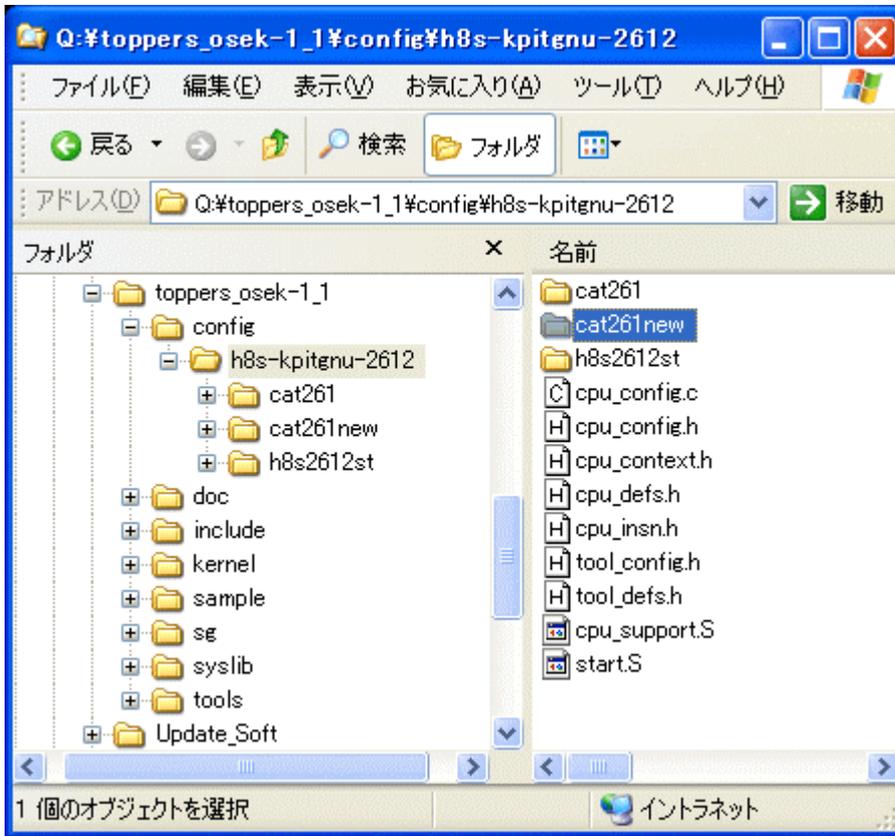
[6-2-3]

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。

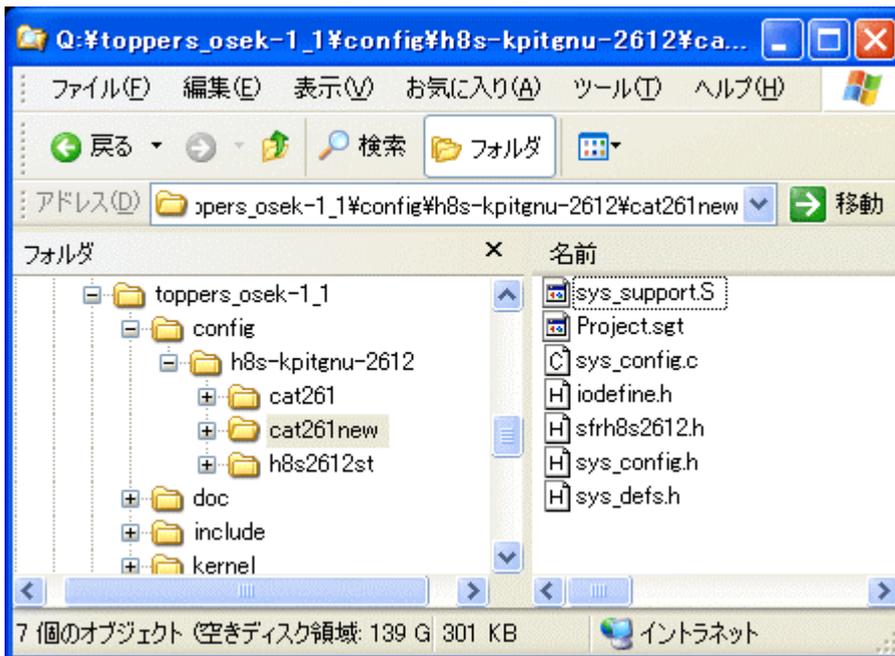
① .¥toppers\_osek¥config¥h8s-kpitgnu-2612 の下に、

新規プロジェクト名「cat261new」のディレクトリを作成します。



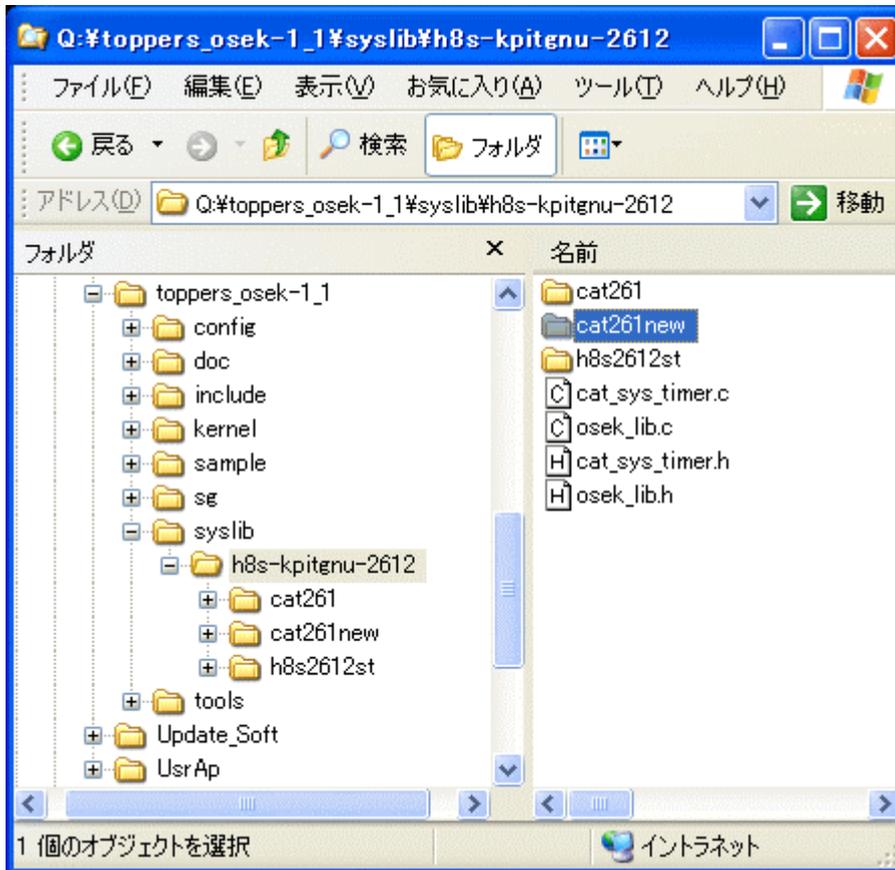
[6-2-4-1]  
<追加1>

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



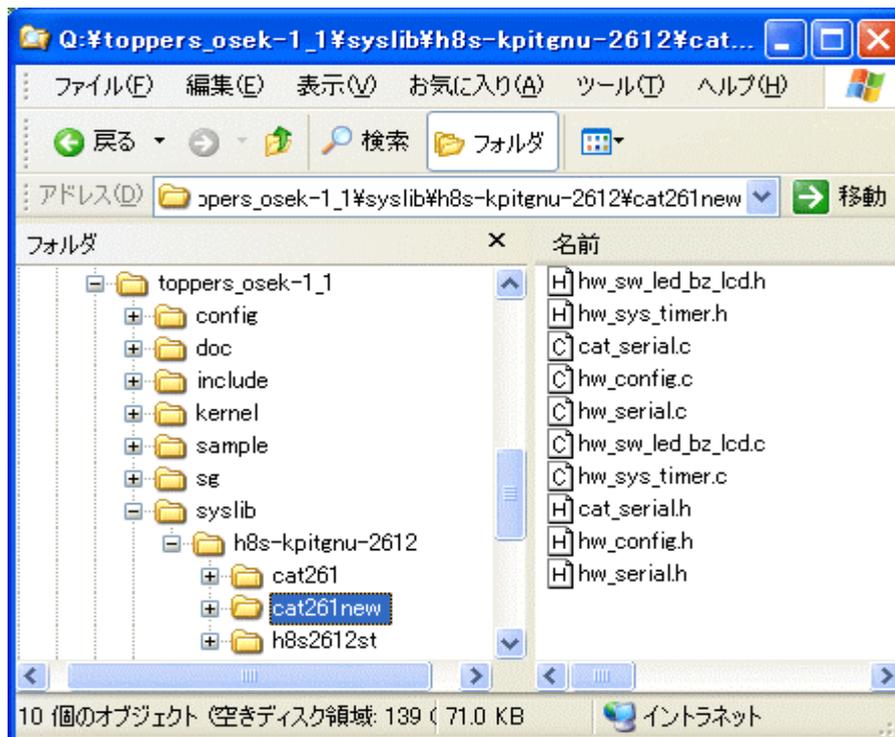
[6-2-4-2]  
コピー確認

- ① .¥toppers\_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。



[6-2-4-3]  
<追加 2 >

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



[6-2-4-4]  
コピー確認

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。

`.\tools\h8s-kpitgnu-2612\cat261new\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.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.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。

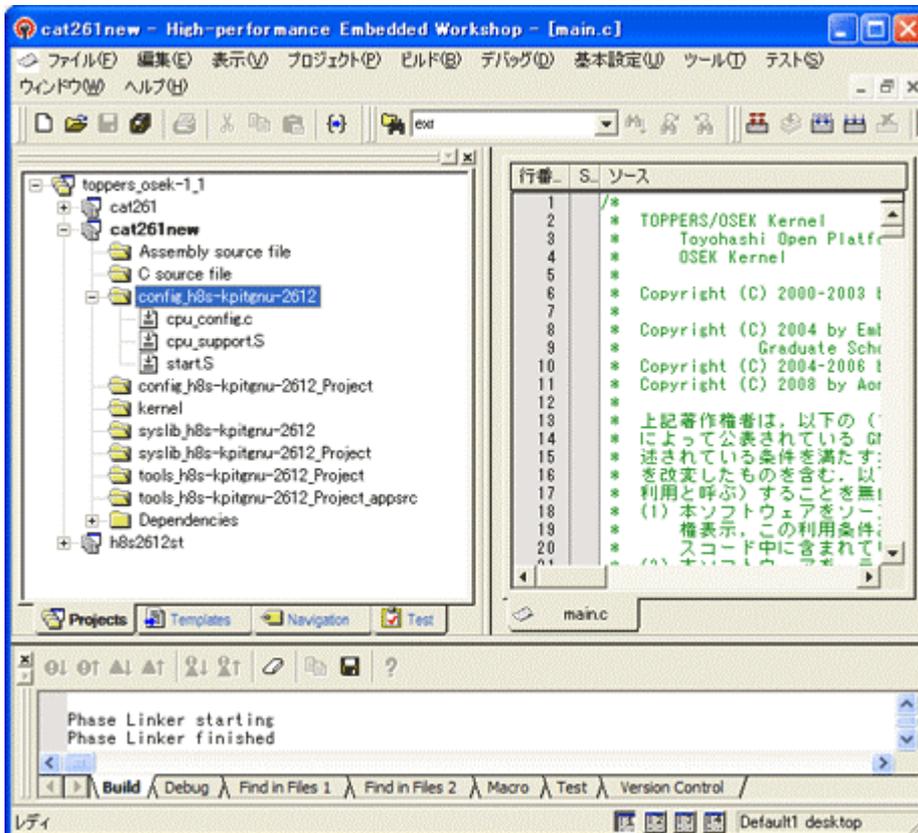


[6-2-6-1]

- ①cpu\_config.c
- ②cpu\_support.s
- ③start.s

の 3 ファイルを「追加」します。

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、マウドロップにて目的フォルダに移動して下さい。



[6-2-6-2]

移動後の状態

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。



[6-2-6-3]

- ①sys\_config.c
- ②sys\_supt.S

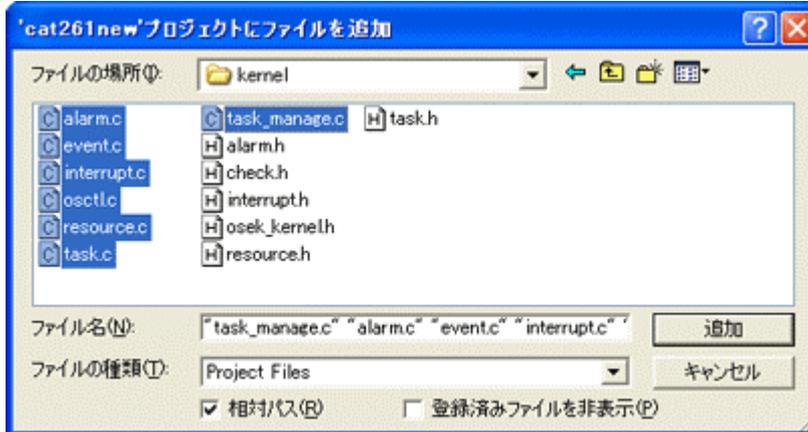
の 2 ファイルを「追加」します。

前項と同じ様に、別フォルダに入りますので、マウสด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。



[6-2-6-4]

- ①alarm.c
- ②event.c
- ③interrupt.c
- ④osctl.c
- ⑤resource.c
- ⑥task.c
- ⑦task\_manage.c

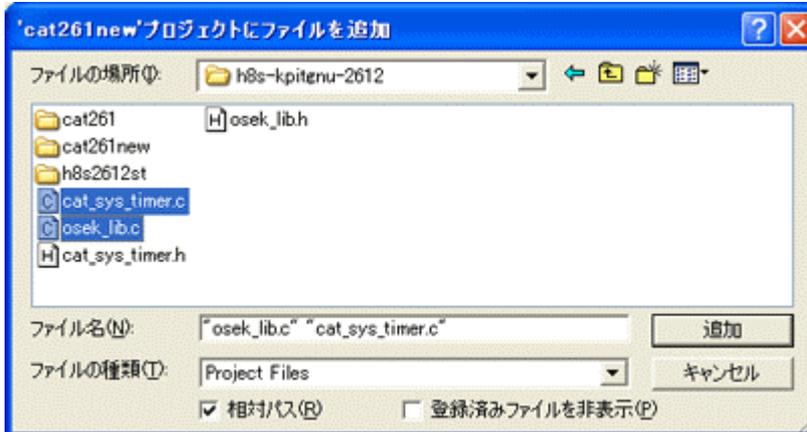
の 7 ファイルを「追加」します。

前項と同じ様に、別フォルダに入りますので、マウสด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。



[6-2-6-5]

- ①sys\_timer.c
- ②osek\_lib.c

の 2 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สด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。



[6-2-6-6]

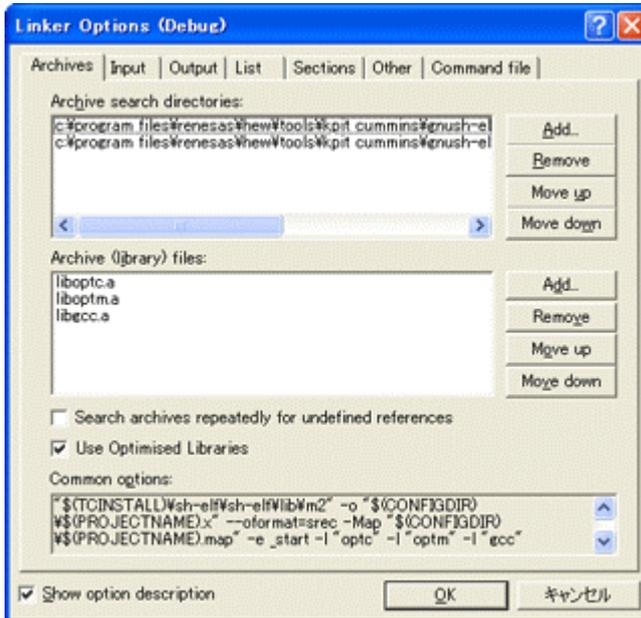
- ①cat\_serial.c
- ②hw.config.c
- ③hw\_serial.c
- ④hw\_sw\_led\_bz\_lcd.c
- ⑤hw\_sys\_timer.c

の 5 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สดロップで目的フォルダに移動して下さい。

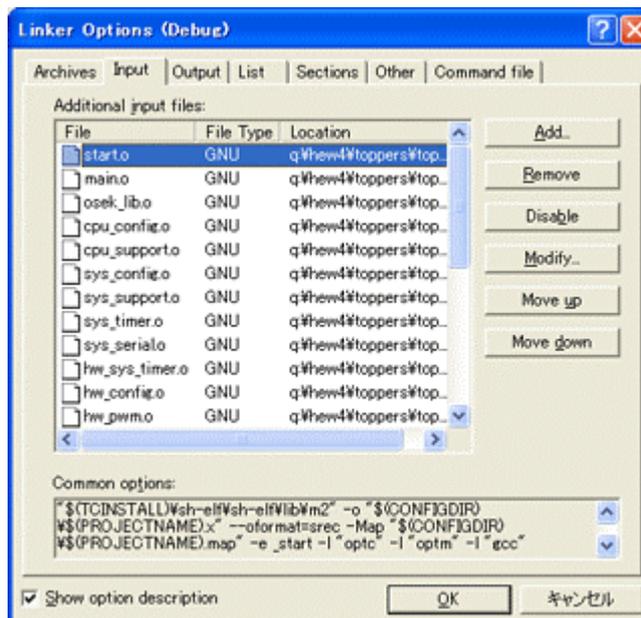
7) 新規プロジェクト「cat261new」のリンク順番を指定します。

Hew メニューの<ビルド>-<Linker>をクリックして下さい。



[6-2-7-1]

<Input>タブをクリックします。

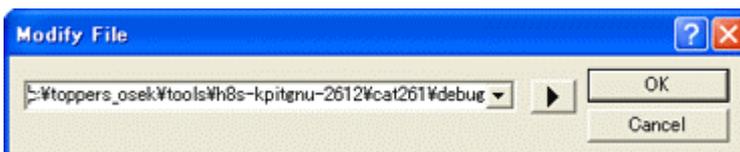


[6-2-7-2]

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ひます。

マウスでのダブルクリックで変更できます。



[6-2-7-3]

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ください。

## 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いと思いますので、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>