

TOPPERS基礎実装セミナー

(LPC2388版:基本)

開発環境編

TOPPERSプロジェクト
教育ワーキング・グループ

2014/03/14

TOPPERSプロジェクト認定

1



本教材の利用条件

NEXCESS基礎コース01 組込みソフトウェア開発技術の基礎

Copyright (C) 2006-2007 by 名古屋大学 組込みソフトウェア技術者人材養成プログラム
Copyright (C) 2006-2007 by 本田晋也

上記著作権者は、以下の(1)～(4)の条件を満たす場合に限り、本コンテンツ(本コンテンツを改変・翻訳したものを含む、以下同じ)を使用・複製・改変・翻訳・再配布(以下、利用と呼ぶ)することを無償で許諾する。

- (1) この枠内の著作権表記等が、そのままの形でコンテンツ中に含まれていること。
- (2) 本コンテンツを再配布する場合には、再配布の形態等を、以下のウェブサイトから報告すること。
<http://www.nces.is.nagoya-u.ac.jp/NEXCESS/REPORT/>
- (3) 本コンテンツを改変・翻訳する場合には、コンテンツを改変・翻訳した旨の記述を、コンテンツ中に含めること。また、改変・翻訳者の著作権表記等は、この枠内の著作権表記等とは別に行うこと。
- (4) 本コンテンツ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、上記著作権者を免責すること。

※ 本コンテンツの一部は、文部科学省 科学技術振興調整費により、名古屋大学 組込みソフトウェア技術者人材養成プログラム(NEXCESS)の一環として作成しました。

※ 本コンテンツ中に記載されている商品名やサービス名などは、各社の商標または登録商標です。

2014/03/14

TOPPERSプロジェクト認定

2



本ドキュメントに関して

1. 著作権に関する表記

＜TOPPERS基礎実装セミナー(LPC2388版:基本)1日目＞

Copyright (C) 2006-2007 by 名古屋大学 組込みソフトウェア技術者人材養成プログラム

Copyright (C) 2006-2007 by 本田晋也 名古屋大学

Copyright (C) 2007-2013 by 竹内良輔 (株)リコー

上記著作権者は、以下の(1)～(3)の条件を満たす場合に限り、本ドキュメント(本ドキュメント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) 本ドキュメントを利用する場合には、上記の著作権表示、この利用条件および以下の無保証規定が、そのままの形でドキュメント中に含まれていること。

(2) 本ドキュメントを改変する場合には、ドキュメントを改変した旨の記述を、改変後のドキュメント中に含めること。ただし、改変後のドキュメントがTOPPERSプロジェクト指定の開発成果物である場合には、この限りではない。

(3) 本ドキュメント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。また、本ドキュメントのユーザまたはエンドユーザからのいかなる理由に基づく請求からも、上記著作権者およびTOPPERSプロジェクトを免責すること。

本ドキュメントは、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、本ドキュメントに関して、特定の使用目的に対する適合性も含めて、いかなる保障もしない。また、本ドキュメント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。

2. 本ドキュメントに関するご意見・ご提言・ご感想・ご質問等がありましたら、TOPPERSプロジェクト事務局までE-Mailにてご連絡ください。

3. 本ドキュメントの内容は、内容の改善や適正化の目的で予告無く改定することがあります。

本ドキュメントでは、Microsoft社のClip Art Galleryコンテンツを使用しています。

TRONは"The Real-time Operating system Nucleus"の略称です。ITRONは"Industrial TRON"の略称です。
μITRONは"Micro Industrial TRON"の略称です。TOPPERS/JSPはToyohashi Open Platform for Embedded Real-Time System/Just Standard Profile Kernelの略称です。」

本ドキュメント中の商品名及び商標名は、各社の商標または登録商標です。

2014/03/14

TOPPERSプロジェクト認定

3



目次

- | | |
|-----------------------|----------|
| 1. 開発環境の確認 | 対象: 基礎1～ |
| 2. ROMモニタ | 対象: 基礎1～ |
| 3. TOPPERS/ASPカーネルの導入 | 対象: 基礎2～ |
| 4. タスクモニタの導入 | 対象: 基礎2～ |

2014/03/14

TOPPERSプロジェクト認定

4



開発環境の確認

対象: 基礎1～

1. [実習ハードウェアの構成](#)
2. ソフトウェア環境の構築

2014/03/14

TOPPERSプロジェクト認定

5



実習ハードウェアの構成

- マイコンボード : LPC2388
- シリアルケーブル
- 5V電源
- SDカード(基礎3で必要)



2014/03/14

TOPPERSプロジェクト認定

6



マイコンボード : LPC2388

- マイコンボード : LPC2388
 - (株)日昇テクノロジー製
- プロセッサ
 - NPX製 CPU:ARM7TDMI-S
 - 周波数72MHz
- 実装機能
 - 2つのUART、ETHER-NET、USEホスト、デバイス
 - SDカード、RTC、JTAG
 - USERスイッチ、RESETスイッチ、8つのLED

2014/03/14

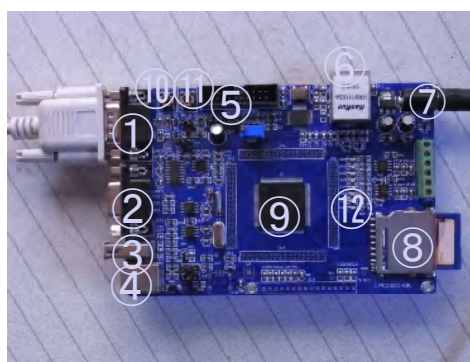
TOPPERSプロジェクト認定

7



マイコンボード : LPC2388の詳細

1. UART0コネクタ
2. UART1コネクタ
3. USBホスト
4. USBデバイス
5. JTAG
6. 10/100ETHER-LAN
7. 5V電源コネクタ
8. SDカードソケット
9. LPC2388
10. USERスイッチ(SW2)
11. リセットスイッチ(SW1)
12. LED



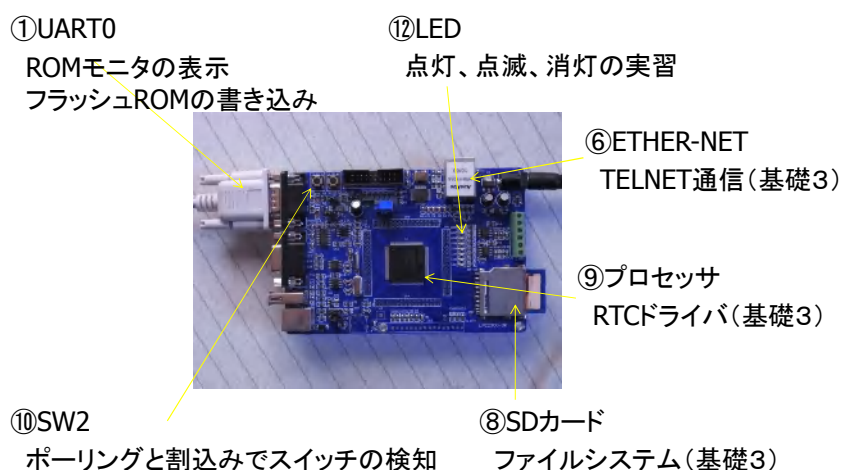
2014/03/14

TOPPERSプロジェクト認定

8



実習で使うハードウェア



2014/03/14

TOPPERSプロジェクト認定

9



プロセッサ(LPC2388):概要

- NXP製ARM7DTMI-Sをコアとしたシングルマイコンプロセッサ
 - T:thumb/D:debug/M:multipier/I:ICE
- ROM:512KB RAM:64KB 16KB: ETHER 16KB:DMA
- 周辺回路:メモリマップドI/O方式
 - 入出力ポート :104本
 - タイマ :4 general purpose timer
 - シリアルI/O :ether usb device/host/OTG:4uart/2CAN
SPI:2 SSP:3 I2S
 - A/D :10 ADC/ 10 DAC
 - その他 :SD/MMC PWM RTC等

2014/03/14

TOPPERSプロジェクト認定

10



LPC2388:メモリマップ

- 通常は512KBのFlash ROM領域にプログラムを置き、64KBのSRAM領域をデータとしてプログラム開発を行う
- 今回、演習用にROMモニタを使用する
- Flash ROM領域にROMモニタプログラムを置き、SRAM中の0x4000F000～0x4000FFFFまでの4KBをROMモニタのデータ領域とし
- 0x4000000～0x4000EFFFをダウンロード領域とする

AHB Peripherals	0xF0000000
APB Peripherals	0xE0000000
64KB External Memory Bank1	0x81010000
	0x81000000
64KB External Memory Bank0	0x80010000
	0x80000000
BOOT FLASH ROM	
16KB EtherNet RAM	0x7FE00000
16KB USB RAM	0x7FD00000
	0x40010000
64KB SRAM	0x40000000
	0x00080000
512KB Flash ROM	0x00000000

2014/03/14

TOPPERSプロジェクト認定

11



開発環境の確認

対象: 基礎1～

1. 実習ハードウェアの構成
2. [ソフトウェア環境の構築](#)

2014/03/14

TOPPERSプロジェクト認定

12



クロス開発環境：ターゲットシステムとホストシステム

組み込みシステムはクロス開発により開発する

クロス開発

- 開発環境（ホスト）と実行環境（ターゲット）が異なる開発形式
- 機械語も異なる

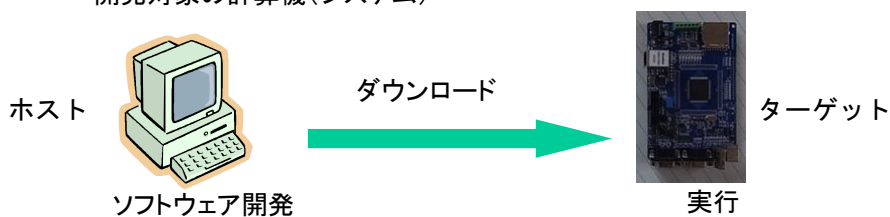
↔ セルフ開発

ホスト

- ソフトウェアを開発する（開発環境を実行する）計算機

ターゲット（ターゲットプロセッサ）

- 開発対象の計算機（システム）



2014/03/14

TOPPERSプロジェクト認定

13



ソフトウェア環境の構築

- LPC2388開発環境
 - Cygwin : UNIX互換環境
 - GNUARM : コンパイラ、アセンブラ、リンカ
 - Flash Magic : フラッシュメモリアイタ
- その他のソフトウェア
 - エディタ (TeraPad, サクラエディタ, Xyzzy等)
 - ターミナルソフトウェア (TeraTermPro等)
 - PDFリーダー (Acrobat Reader等)

2014/03/14

TOPPERSプロジェクト認定

14



組込みソフトウェア開発に使われるプログラミング言語

C言語

- ハードウェアを直接操作するプログラミングが可能であるため、組込みソフトウェア開発では、最も使われている

アセンブリ言語

- DSPなどの特殊なプロセッサで使われる場面が多い
- コンパイラが扱えない特殊命令を直接記述して、性能を出す

C++言語

- 利用は広がっているが、まだ限定的
- オーバーヘッドが大きい
- どのような実行コードになるか見えにくい

2014/03/14

TOPPERSプロジェクト認定

15



C言語のツールチェーン

C言語コードを実行コードに変換するためのツール群

コンパイルドライバ

- 実行コードを生成するまでの一連の処理を実行
 - プリプロセッサ, コンパイラ, アセンブラ, リンカを呼び出す

プリプロセッサ

- #includeやマクロを展開

コンパイラ

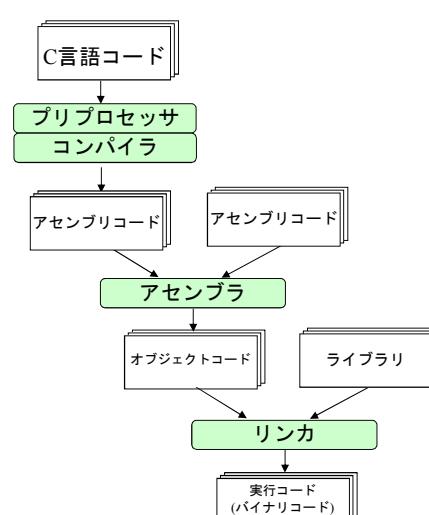
- プリプロセッサされたC言語コードをアセンブリコードへ変換

アセンブラ

- アセンブリコードをオブジェクトコード(機械語プログラム)に変換

リンカ

- 複数のオブジェクトコードとライブラリをリンクし実行コードを生成する



2014/03/14

TOPPERSプロジェクト認定

16



GCCのツールチェーン

プリプロセッサ : `cpp`

- `#include`やマクロを展開する

コンパイラ : `gcc`

- オブジェクトコードを生成するまでの一連の処理(プリプロセッサ, コンパイラ, アセンブラ, リンカの呼び出し)を行う場合は, コンパイラドライバとも呼ばれる

アセンブラ : `as`

- アセンブリ言語記述を機械語プログラムに変換する.

リンカ : `ld`

- 複数の機械語プログラムとライブラリをリンクし最終イメージを生成する

2014/03/14

TOPPERSプロジェクト認定

17



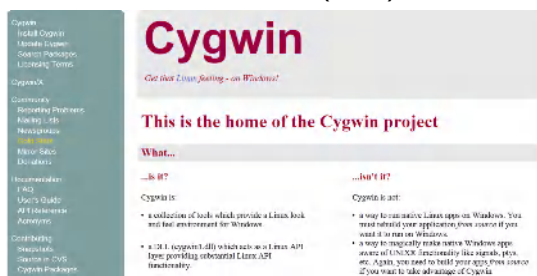
Cygwinのインストール

- CygwinのサイトからCygwin 1.5.x以降のバージョンをダウンロードし、Windows-PC (XP / 7 / 8)にインストールします (makeのバージョンは3.81以降をお勧め)

Cygwinサイト → <http://www.cygwin.com/>

注意 トラブル回避のために、すでにCygwinをインストール済みの方は、バージョンの確認をお願いします

bash上で、`uname -a`(return)がCygwinのバージョン問い合わせ、
`make -ver`(return)がmakeのバージョン表示です



2014/03/14

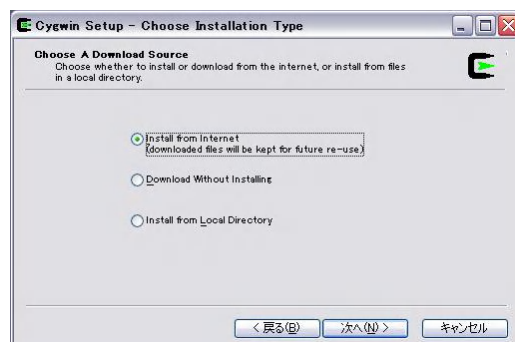
TOPPERSプロジェクト認定

18



Cygwinのインストール

- setup.exeを起動して、インターネット経由のダウンロードインストールまたはダウンロード後インストールのどちらかを選択できます



2014/03/14

TOPPERSプロジェクト認定

19



Cygwinのインストール

- マルチバイト文字およびスペースを含まないディレクトリにインストールします(例:C:\cygwin)



2014/03/14

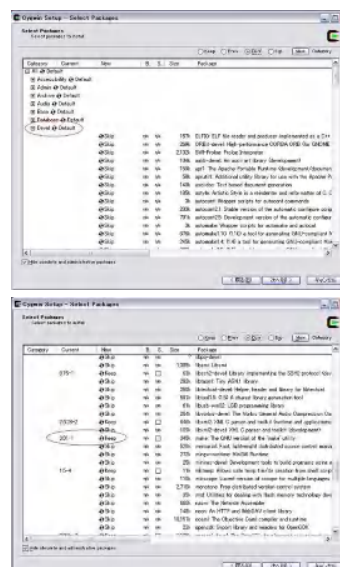
TOPPERSプロジェクト認定

20



Cygwinのインストール

- makeのバージョンは3.81-1を選択します
- makeコマンドはmakeファイルの記述に従って、ビルドの手順を指定するコマンドです



2014/03/14

TOPPERSプロジェクト認定

21



ARM-GCCのインストール

- GCCのインストールは、GCCのソースコードをダウンロードして、インストールが可能ですが、手順が複雑なため、ここではARM-GCCのバイナリインストールを行います
- 指定の(bu-2.16.1 gcc-4.0.2-c-c++ nl-1.14.0 gi-6.4.exe)をダウンロードしてください



2014/03/14

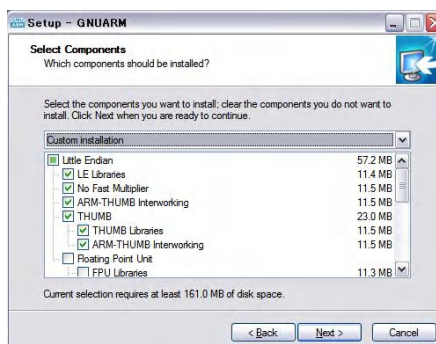
TOPPERSプロジェクト認定

22



ARM-GCCのインストール

- インストールディレクトリはUNIX風に
C:\cygwin\usr\local\bin
にインストールします
- LPC2388で使われているARM7にはlittle Endian, Floating Point Unitなし, THUMBコードの対応を行うためにこのダイアログの設定としてください



2014/03/14

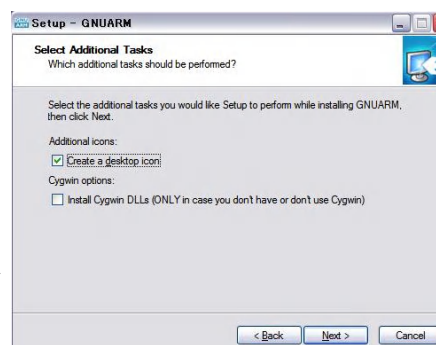
TOPPERSプロジェクト認定

23



ARM-GCCのインストール

- Cygwinはインストール済みのため、"install Cygwin DLLs..."は選択しないでください
- インストール終了時に、GNU ARMインストールディレクトリに対するWindows環境変数(パス)登録を確認されますが、パスを登録する必要はありません



2014/03/14

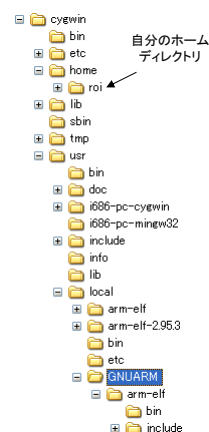
TOPPERSプロジェクト認定

24



PATHの設定

- 自分のHOMEディレクトリ
の.bash_profileの内容を修正します
- .bash_profile中のexportコマンドにイン
ストールしたARMGCCのパス
(/usr/local/GNUARM/bin)を追加しま
す
- Cygwinの起動後コンパイラ等のコマ
ンドが使用できるようになります
 - arm-elf-gcc コンパイラ
 - arm-elf-ld リンカ
 - arm-elf-ar ライブラリアン



```
export PATH=${PATH}:/usr/local/GNUARM/bin
```

2014/03/14

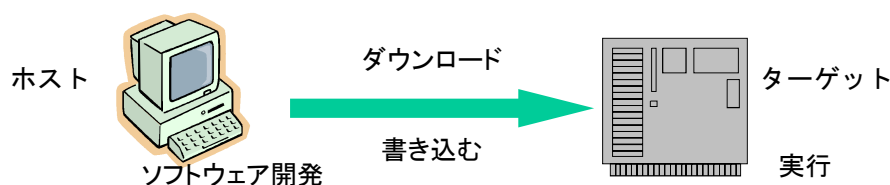
TOPPERSプロジェクト認定

25



Flash Magicのインストール

- LPC2388にプログラムを書き込むためにパソコンに書き
込みツールをインストールする
 - Flash Magic
- Flash MagicはLPC2388内のFlashROMにプログラムを書
き込むためのツールです



2014/03/14

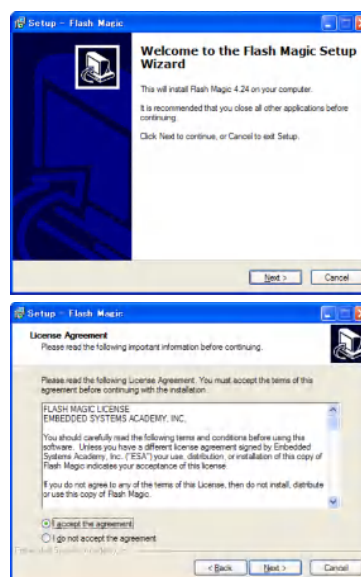
TOPPERSプロジェクト認定

26



Flash Magicのインストール

- FlashMagic.exeを実行する
- 「Next」を押すと、ライセンス画面が表示される。
- 「I accept the agreement」を選択して、「Next」ボタンを押す



2014/03/14

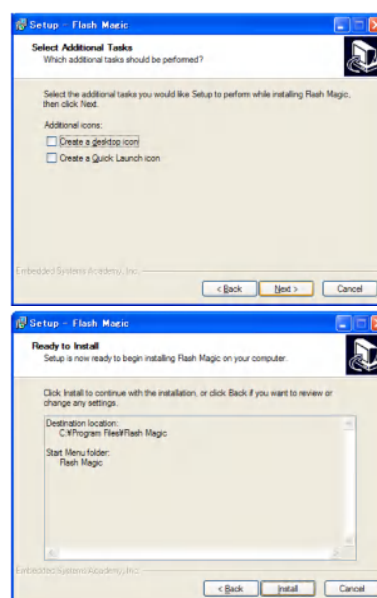
TOPPERSプロジェクト認定

27



Flash Magicのインストール

- インストール先を変更せず、「Next」を押す
- メニュー・フォルダを変更せず、「Next」を押す



2014/03/14

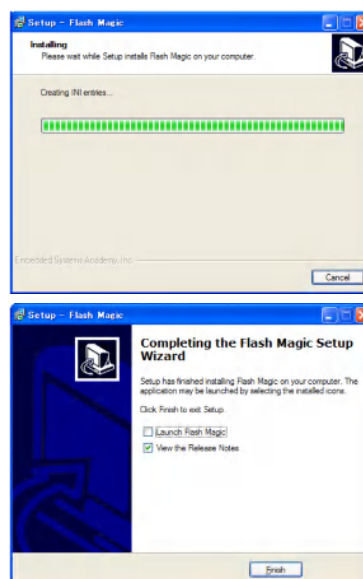
TOPPERSプロジェクト認定

28



Flash Magicのインストール

- インストール終了画面で「Finish」を押すとインストールが終了します



ROMモニタ

対象: 基礎1～

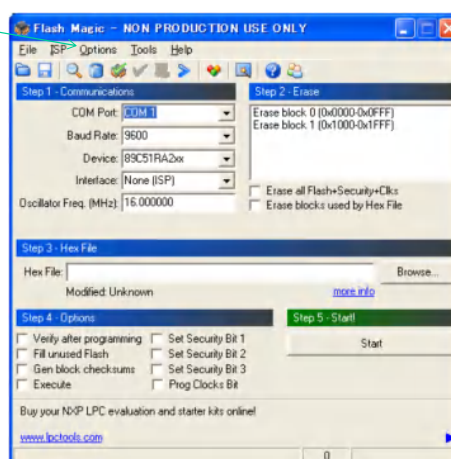
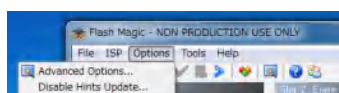
1. [ROMモニタを書き込む](#)
2. ダウンロードと実行

ROMモニタをボードに書き込む

- Windowsのスタートメニューから「Flash Magic」を起動する

「Option」メニュー

最初に起動した時は、
左の画面が表示されます
「Option」⇒「Advanced
Options」を選択してくだ
さい



2014/03/14

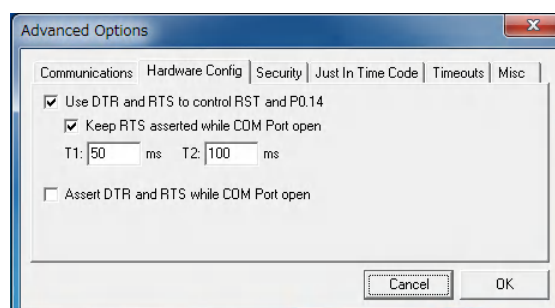
TOPPERSプロジェクト認定

31



Flash Magicの初期設定

- Advanced Optionsメニュー中の「Hardware Config」メニューを表示して画面の通りの設定を行う



2014/03/14

TOPPERSプロジェクト認定

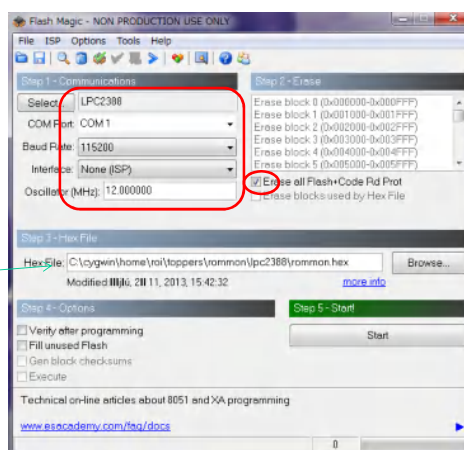
32



rommon.hexを選択する

- メイン画面のSTEP1/2/3を設定する
- STEP3では
../rommon/lpc2388/rommon.hexを選択する

rommon.hex
ファイルの選
択



2014/03/14

TOPPERSプロジェクト認定

33

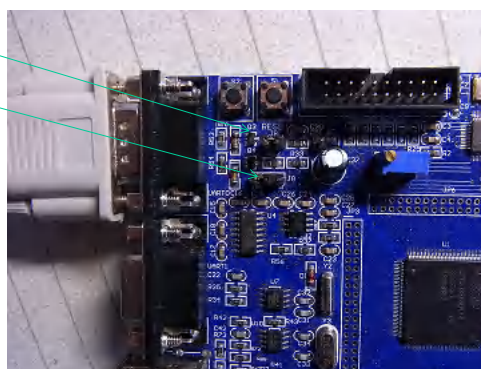


Flash Magicでプログラムを書き込む

- LPC2388ボードのUART0ポートにRS232Cケーブルを接続し、J8ピンを接続、J9ピンの左の2つのピンを接続する
- ボードの電源を入れてメニューの「Start」ボタンを押すと書き込みが始まります

J8ピン

J9ピン



2014/03/14

TOPPERSプロジェクト認定

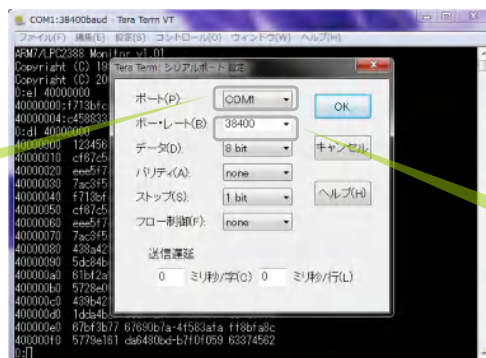
34



ROMモニタの操作方法 : TeraTermシリアルポート

- ROMモニタはUARTを使用して、ボードに対して種々の設定や確認を行うことができます
- TeraTermは, UARTから送られてくるデータを表示する
- TeraTermのシリアルポート設定で設定を行います

接続ポート



ボーレート

2014/03/14

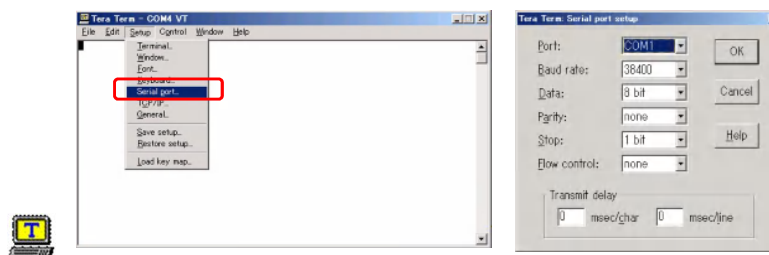
TOPPERSプロジェクト認定

35



ROMモニタの操作方法 : TeraTermシリアルポート

- TeraTermのメニューのSetup→Serial port で設定
 - Port は接続しているCOMポートの番号に設定
 - Baud rate : 38400bps, Data : 8bit
 - Parity : none, Stop : 1bit, Flow control : なし
- 設定後メニューのSetup→Save setupで設定を保存



2014/03/14

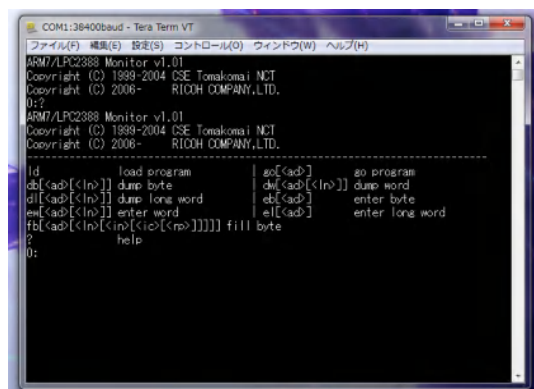
TOPPERSプロジェクト認定

36



ROMモニタの操作方法

- 電源を切り、J8ピンをオフ、J9ピンの右2つのピンを接続する
- TeraTermを立ち上げる
- 電源を入れる



2014/03/14

TOPPERSプロジェクト認定

37



ROMモニタの操作方法

- ROMモニタを起動する前は常に
 - J8、J9が書き込み側ではないか？
 - TeraTermを立ち上げたか？
 - ボードの電源が入っているか？
 を確認して、一度リセットボタンを押す
- コマンドを使ってモニタを操作する
 - ? Enterで、使用可能なコマンドの一覧を表示する



2014/03/14

TOPPERSプロジェクト認定

38



操作方法:読み出し

- dl コマンドで4バイト単位のメモリDUMPができます
- db コマンドは1バイト単位、dw コマンドは2バイト単位

dl <address> return

2014/03/14

TOPPERSプロジェクト認定

39



操作方法:書き込み

- el コマンドにて4バイト単位でのデータ書き込みができます
- el アドレスで、データ入力モードに移行し、データを入力すると設定アドレスにデータを書き込みます
- '.' returnにて、データ入力モードを終了します

RAM領域の
0x40000000番地と
0x40000004番地に
0x87654321と
0x12345678を書き込む

2014/03/14

TOPPERSプロジェクト認定

40



ROMモニタ

対象: 基礎1～

1. ROMモニタを書き込む
2. [ダウンロードと実行](#)

2014/03/14

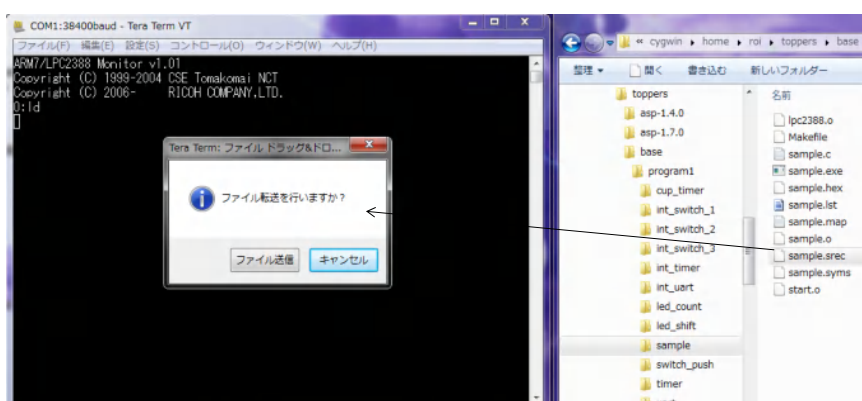
TOPPERSプロジェクト認定

41



実行: ダウンロード

- ・ モニタ上でld(return) を実行しダウンロードモードに設定
- ・ SRECフォーマットのダウンロードファイルをエクスプローラからTeraTermにドラックする



2014/03/14

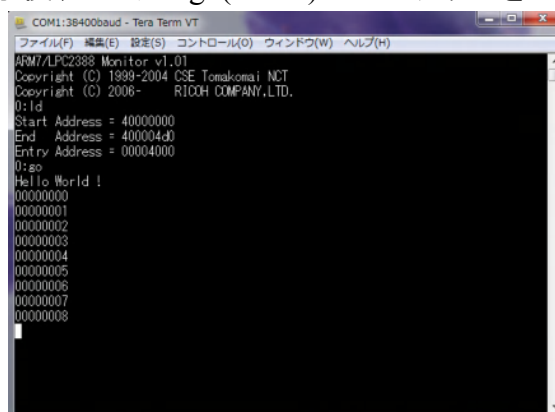
TOPPERSプロジェクト認定

42



実行:ダウンロード開始、実行開始

- ダイアログのファイル転送をクリックするとダウンロードを開始する
- ダウンロード終了後、モニタのgo(return) でプログラムを実行する



```
COM1:38400baud - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(C) ウィンドウ(W) ヘルプ(H)
ARM/LPC2388 Monitor v1.01
Copyright (C) 1999-2004 CSE Tomakomai NCT
Copyright (C) 2006- RICOH COMPANY,LTD.
0:ld
Start Address = 40000000
End Address = 400004d0
Entry Address = 00004000
0:go
Hello World!
00000000
00000001
00000002
00000003
00000004
00000005
00000006
00000007
00000008
```

TOPPERS/ASPカーネルの導入

対象:基礎2～

TOPPERS/ASPのダウンロード

- TOPPERSプロジェクトのWebからasp-1.7.0の個別パッケージをダウンロードします
- 同様にARMアーキテクチャ・GCC依存部パッケージ asp_arch_arm_gcc-1.7.0をダウンロードします

ARMアーキテクチャ・GCC依存部パッケージ (担当: 名古屋大学)

パッケージ	ディレクトリ	対応する非依存部のバージョン	リリース日
asp_arch_arm_gcc-1.7.0.tar.gz	arch/arm_gcc/common/target/at91skyeye_gcc	1.7.*	2011-05-19
asp_arch_arm_gcc-1.4.0.tar.gz	arch/arm_gcc/common/arch/arm_gcc/at91sam7s/target/at91skyeye_gcc/target/bic090_gcc	1.4.*	2009-05-18
asp_arch_arm_gcc-1.3.2.tar.gz	arch/arm_gcc/common/arch/arm_gcc/at91sam7s/target/at91skyeye_gcc/target/bic090_gcc	1.3.*	2008-08-27

TOPPERS/ASPカーネル 個別パッケージのダウンロード

TOPPERS/ASPカーネルの最新リリースの個別パッケージを配布しています。各ターゲットカーネル非依存部パッケージ、依存部パッケージ (ARMアーキテクチャ依存部、ターゲット・ビルドツールの依存部(PD/C)パッケージ)を個別にダウンロードできます。

ターゲットごとに必要なソースコードを一つにまとめた簡易パッケージは、[こちら](#)から手に入ります。

一般公開以前のバージョン (Release 1.3.0以前) に対応した個別パッケージは、公開版として提供していませんが、[こちら](#)からダウンロードできます。

※ ゲット非依存部

TOPPERS/ASPカーネル ターゲット非依存部パッケージ (担当: 名古屋大学)	リリース日
asp-1.8.0.tar.gz	2012-12-26
asp-1.7.0.tar.gz	2011-05-09

2014/03/14

TOPPERSプロジェクト認定

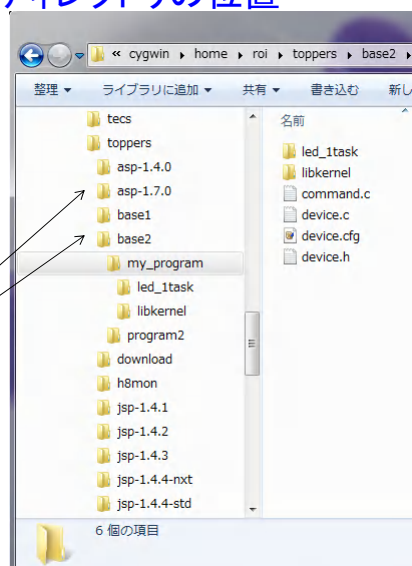
45



TOPPERS/ASPを構築するディレクトリの位置

- TOPPERS/ASPの開発ディレクトリは基礎2、基礎3で共通に使用します
- 2日目の実習プログラムのディレクトリ(base2)と並列の位置にasp-1.7.0のディレクトリを作成してください

base2の実習ディレクトリとasp-1.7.0は並列に作成



2014/03/14

TOPPERSプロジェクト認定

46



TOPPERS/ASPの解凍

- 開発用のディレクトリにasp-1.7.0.tar.gz、asp_arch_gcc-1.7.0.tar.gzと、添付の開発環境asp-1.7.0-030913.tar.gzを置きます
- asp-1.7.0.tar.gzとasp_arch_arm_gcc-1.7.0.tar.gzの解凍後、解凍されたaspディレクトリ名をasp-1.7.0に変更します
- その後、asp-1.7.0-030913.tar.gzを解凍します

```
$ ls
asp_arch_arm_gcc-1.7.0.tar.gz asp-1.7.0.tar.gz asp-1.7.0-070812.tar.gz
$ tar zxvf asp-1.7.0.tar.gz
$ tar zxvf asp_arch_arm_gcc-1.7.0.tar.gz
$ mv asp asp-1.7.0
$ tar zxvf asp-1.7.0-030912.tar.gz
```

2014/03/14

TOPPERSプロジェクト認定

47



コンフィギュレータの構築

- Webより、コンフィギュレータをダウンロードしasp-1.7.0フォルダの下に置きます
- LINUXやCygwinでは、cfgの下でMakefileを用いてコンフィギュレータのビルドを行う
- コンフィギュレータの構築はCygwinのバージョンにも関係するため、構築済みの実行形式を利用してもよい

```
$ tar zxvf cfg-1.8.0.tar.gz
$ cd cfg
$ ./configure
$ make
```

TOPPERS新世代カーネル用コンフィギュレータ

TOPPERS新世代カーネル用コンフィギュレータの最新リリースを配布しています。コンフィギュレータ自体についてはこちらを、構築方法等についてはアーカイブに含まれる README.txt をご覧ください。

最新のリリース			
リリース名	タイプ	サイズ	リリース日
コンフィギュレータ Release 1.8.1	tar.gz(EUC-LF)	139KB	2013-04-05
コンフィギュレータ Release 1.8.1(Windows用バイナリ)	zip	10.8MB	2013-04-05
Release 1.8.0Cの通り			

過去のリリース			
リリース名	タイプ	サイズ	リリース日
コンフィギュレータ Release 1.8.0	tar.gz(EUC-LF)	139KB	2012-12-26
コンフィギュレータ Release 1.8.0(Windows用バイナリ)	zip	10.7MB	2012-12-26
Release 1.8.0Cの通り			
コンフィギュレータ Release 1.8.0	tar.gz(EUC-LF)	91KB	2012-05-09
コンフィギュレータ Release 1.8.0(Windows用バイナリ)	zip	1.0MB	2012-05-09
コンフィギュレータ Release 1.8.0(Cygwin用バイナリ)	tar.gz	650KB	2012-05-09

2014/03/14

TOPPERSプロジェクト認定

48



TOPPERS/ASPのワークスペースの作成

- asp-1.7.0ディレクトリの下にOBJ/LPC2388/SAMPLE1ディレクトリを作成します
- 教材では基礎3講座用のSAMPLE1のプログラムが用意されていますが、解説のため、これを削除し再生成します
- SAMPLE1ディレクトリに入り、configureコマンドを使用してSAMPLE1プログラムを生成します
- 以下のファイルが生成されます
 - Makefile, sample1.cfg, sample1.c, sample1.h

```
$ cd ../OBJ/LPC2388_GCC/SAMPLE1
$ rm *
$ ../../configure -T lpc2388_gcc
$ ls
Makefile sample1.c sample1.cfg sample1.h
```



タスクモニタの導入

対象: 基礎2～



タスクモニタの重要性1

- 中規模システムでは、複数のサブシステム分離開発する。サブシステムの結合を行うと種々の原因でいろいろな問題が発生する。ひとつのサブシステムの不具合により、別のサブシステムが誤動作する可能性もある
- 評価専門の部署で商品評価を行う。デバッグ環境の整っていない環境で、不具合の一時調査を行う必要がある
- いろいろな開発、評価の部署で問題が発生した場合、商品に近い開発形態で問題解決する手段が必要となる



商品形態でタスクモニタを実装することにより、問題調査を行うことができる

2014/03/14

TOPPERSプロジェクト認定

51



タスクモニタの重要性2

- 中期規模システムでは、結合環境でサブシステムや専用ミドルウェアの状態表示やモード設定のためのインターフェイスがあったほうが、問題解決を行いやすい
- 開発メーカ供給のデバッガやICEでは、ユーザーシステムに特化した機能を組み込めない
 - このような開発環境は基本的に改造ができない



ソース公開されているタスクモニタに、ユーザーシステム専用の機能を組み込むみ改造することにより、開発効率を上げることができる

2014/03/14

TOPPERSプロジェクト認定

52



タスクモニタの取り込み1

- 教育WGで開発したタスクモニタをサンプルプログラムに取り込みを行う
- program2ディレクトリ以下の3つのファイルをasp-1.7.0/pdic/lpc23xxディレクトリの下にコピーする
 - device.cfg/device.c/device.h
- タスクモニタを取り込みには以下の変更が必要です
 - タスク化するため、コンフィギュレーションファイルの変更を行います
 - インクルードパスの追加と部品となるファイルの追加を行います
 - Makefileの修正を行います

2014/03/14

TOPPERSプロジェクト認定

53



タスクモニタの取り込み2(デバイスドライバ)

- この演習で使用するデバイス(LED、ディップスイッチ、プッシュスイッチ)のデバイスドライバを取り込みます
- 基礎1で使用したデバイスドライバをdeviceディレクトリ以下に用意してあります

関数名	型	引数	機能
led_init	void	inptr_t exinf	LEDデバイスの初期化
led_in	uint8_t	void	LEDの設定値の取り出し
led_out	void	uint8_t led_data	LED点灯、消灯を設定
switch_dip_init	void	inptr_t exinf	ディップスイッチの初期化
switch_dip_sense	uint8_t	void	ディップスイッチの状態取得
switch_push_init	void	inptr_t exinf	プッシュスイッチの初期化
switch_push_state	uint8_t	void	プッシュスイッチの状態取得

2014/03/14

TOPPERSプロジェクト認定

54



デバイスドライバを取り込む

- C言語の標準ライブラリを使用するため、LIBSに-lcの設定を追加します

```

130 #
131 # 共通コンパイルオプションの定義
132 #
133 COPTS := $(COPTS) -g
134 ifndef OMIT_WARNING_ALL
135 COPTS := $(COPTS) -Wall
136 endif
137 ifndef OMIT_OPTIMIZATION
138 COPTS := $(COPTS) -O2
139 endif
140 CDEFS := $(CDEFS)
141 INCLUDES := -I. -I$(SRCDIR)/include -I$(SRCDIR)/arch -I$(SRCDIR) $(INCLUDES)
142 LDFLAGS := $(LDFLAGS)
143 LIBS := $(LIBS) -lc $(CXXLIBS)
144 CFLAGS = $(COPTS) $(CDEFS) $(INCLUDES)

```

2014/03/14

TOPPERSプロジェクト認定

55



タスクモニタとデバイスドライバを取り込み

- Makefileにmonitorディレクトリ中のMakefile.configをインクルードすることによりパスや部品の設定が行える
- デバイスドライバのプログラム(device.c)とmonitorのボード依存部(armv4.c)をシステムサービスとして取り込む

```

167 #
168 # ミドルウェアの Makefile のインクルード
169 #
170 include $(SRCDIR)/monitor/Makefile.config
171
172 #
173 # システムサービスに関する定義
174 #
175 SYSSVC_DIR := $(SYSSVC_DIR) $(SRCDIR)/sysvc $(SRCDIR)/library $(SRCDIR)/pdic/lpc23xx
176 SYSSVC_ASMOBS := $(SYSSVC_ASMOBS)
177 SYSSVC_COBJS := $(SYSSVC_COBJS) banner.o syslog.o serial.o logtask.o ¥
178 log_output.o vasylog.o t_perror.o strerror.o ¥
179 armv4.o device.o $(CXXRTS)
180 SYSSVC_CFLAGS := $(SYSSVC_CFLAGS)
181 SYSSVC_LIBS := $(SYSSVC_LIBS)
182 INCLUDES := $(INCLUDES) -I$(SRCDIR)/pdic/lpc23xx

```

2014/03/14

TOPPERSプロジェクト認定

56



タスクモニタの取り込み(CFGファイルの修正)

- sample1.cfgにmonitor.cfgとdevice.cfgを取り込む
- サンプルプログラムが自動実行しないようにTA_ACTを変更

```

INCLUDE("target_timer.cfg");
INCLUDE("syssvc/syslog.cfg");
INCLUDE("syssvc/banner.cfg");
INCLUDE("syssvc/serial.cfg");
INCLUDE("syssvc/logtask.cfg");
INCLUDE("monitor/monitor.cfg");
INCLUDE("pdic/lpc23xx/device.cfg");

#include "sample.h"
CRE_TSK(TASK1, { TA_NULL, 1, task, MID_PRIORITY, STACK_SIZE, NULL });
CRE_TSK(TASK2, { TA_NULL, 2, task, MID_PRIORITY, STACK_SIZE, NULL });
CRE_TSK(TASK3, { TA_NULL, 3, task, MID_PRIORITY, STACK_SIZE, NULL });
CRE_TSK(MAIN_TASK, { TA_NULL, 0, main_task, MAIN_PRIORITY, STACK_SIZE, NULL });
DEF_TEX(TASK1, { TA_NULL, tex_routine });
DEF_TEX(TASK2, { TA_NULL, tex_routine });
DEF_TEX(TASK3, { TA_NULL, tex_routine });
CRE_CYC(CYCHDR1, { TA_NULL, 0, cyclic_handler, 2000, 0 });
CRE_ARM(ALMHDR1, { TA_NULL, 0, alarm_handler });

```

追加

修正

2014/03/14

TOPPERSプロジェクト認定

57



タスクモニタのソースの確認

- 以下のファイルはmonitor/Makefile.configによりビルド時に自動的に取り込まれます
- タスクモニタの入出力先はstddev.cで関数設定されていますので、関数を入れ替えれば、種々のデバイスに対応が可能です

標準入出力の追加	モニタ基本部の追加	ARMV4依存部の追加
stdio/stddev.c	monitor.c	arch/armv4.c
stdio/printf.c	task_expansion.c	
stdio/sprintf.c		
stdio/scanf.c		
stdio/sscanf.c		

2014/03/14

TOPPERSプロジェクト認定

58

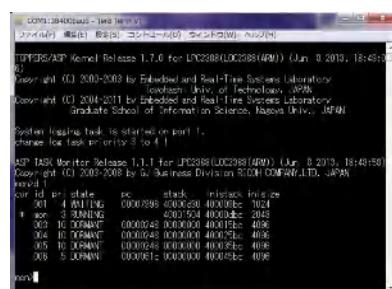


ビルドとタスクモニタの起動を確認

- makeコマンドにて、再ビルドします
- プログラム書き込み後、電源投入でタスクモニタが起動します

```
$ make realclean
$ make depend
$ make
```

構成変更となったため、
realcleanから始める



2014/03/14

TOPPERSプロジェクト認定

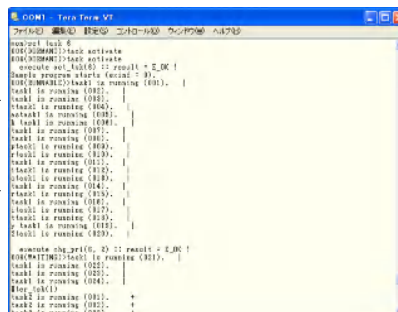
59



タスクモニタでのタスク優先度の注意

- タスクモニタの優先度は3で通常はコンソールからの入力待ちで待ち状態になっており、コンソールに入力があると起床して指定されたコマンドを実行する
- メインタスクの優先度2以上に設定すると、メインタスクが優先的にサンプルプログラム操作を行い、タスクモニタの操作ができなくなる

選択されているタスクがメインタスクの状態、task priority 2 (return)の短縮コマンド tp 2 (return)を発行して優先度2に変更



2014/03/14

TOPPERSプロジェクト認定

60



タスクモニタの使用法

- タスクモニタが起動されてコンソールにモニタの起動ログを表示し、コマンド待ちを示す `mon>` が表示される
- `help`と入力するとコマンド一覧が表示される

```

mon>help
  Display BYTE      [start address] [end address]
          HALF      [start address] [end address]
          WORD       [start address] [end address]
          TASK
          REGISTER
  Set      BYTE      [start address]
          HALF      [start address]
          WORD       [start address]
          COMMAND    [mode] mode=1 or (2)
          SERIAL     [port id]
          TASK        [task id]
  Task     ACTIVATE   (act_tsk)
          TERMINATE  (ter_tsk)
          SUSPEND    (sus_tsk)
          RESUME     (res_tsk)
          PRIORITY   [pri] (chg_pri)
  Log      MODE       [logmask] [logmask]
          TASK        [cycle time(ms)]
          PORT        [no] [logno] [portaddress]
  Help

```

2014/03/14

TOPPERSプロジェクト認定

61



タスクモニタからのメモリ操作

- タスクモニタから直接、LEDを操作します
- `set word`コマンドでLEDが接続されているポート0のポートレジスタ(0x3fffc058/0x3fffc05c)に直接書き込み、動作を確認する

0x3fffc040をOUT設定:設定済

2014/03/14

TOPPERSプロジェクト認定

62



タスクモニタによるタスク操作

- ・ タスクモニタではタスクに対して次の操作が可能
 - タスクの起動 (act_tsk) : タスクを実行可能状態に
 - タスクの終了 (ter_tsk) : タスクを休止状態に
 - タスクのサスペンド (sus_tsk) : タスクを強制待ち状態に
 - タスクのレジューム (rsm_tsk) : タスクを強制待ち状態から復帰
 - 優先度の変更 (chg_pri) : タスクの優先度を変更する
- ・ タスクの操作する前に操作対象のタスクを指定する必要がある
 - タスクIDは kernel_id.h で定義されている値
 - 一度設定すると、それ以後のタスク操作は指定したタスクに対して行われる

```
mon> set task [タスクID]
```

2014/03/14

TOPPERSプロジェクト認定

63



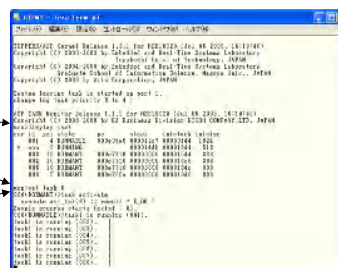
タスクの起動

- ・ タスクモニタからサンプルプログラムを実行する
- ・ 入力はタスクモニタが取り込むため、操作はできない
- ・ display task : タスクの状態を表示
- ・ set task 6 : ID番号6(メインタスク)のタスクを指定
- ・ task activate : タスクの起動
- ・ コマンドは最初の一文字で短縮可能

display task

set task 4

task activate



2014/03/14

TOPPERSプロジェクト認定

64



タスクの終了

- サンプルプログラムは3～6までの4つのタスクで実行している
- task terminate : 現在選ばれているタスク(6)を終了
- set task とtask terminateで、3～5のタスクを終了させる

リセットは直接実行版では、
リセットボタン押下。

ROMモニタが再起動するので、ダウンロードから再実行させる

[illegible]

Syslogの表示レベルの変更

- サンプルプログラムはsyslogのLOG_NOTICEで表示を行っている、タスクモニタからsyslogの表示レベルを変更できる
- 表示を無視して、log mode 4 (return)を入力すると、表示レベルがLOG_WARNINGとなってログ表示なくなる

[illegible]