

# TOPPERS基礎実装セミナー

## (STM32F4-Discovery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

2015/12/31

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。

2015/12/31

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。

2015/12/31

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. タスクモニタの導入 | 対象: 基礎2～ |

2015/12/31

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. USBシリアルケーブルの接続
3. ソフトウェア開発環境の構築

## 実習ハードウェアの構成

- マイコンボード : STM32F4-Discovery
- USBシリアルケーブル
- USBケーブル (mini-B)
- ピンケーブル



## マイコンボード : STM32F4-Discovery

- マイコンボード : STM32F4-Discovery
  - STMicroelectronics
- プロセッサ
  - STM製 CPU:STMF407(Cortex-M4F)
  - 周波数168MHz
- 実装機能
  - 2つのUSBコネクタ
  - Audio DAC with integrated class D speaker driver
  - USERスイッチ、RESETスイッチ、4つのLED

2015/12/31

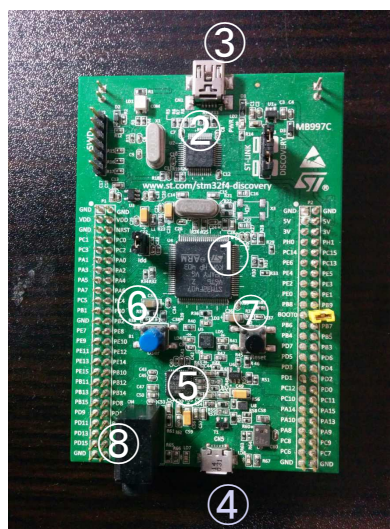
TOPPERSプロジェクト認定

7



## マイコンボード : STM32F4-Discovery

1. STM32F407
2. STM32F103
3. USB OTG FS A
4. USB OTG FS B
5. LED
6. USERスイッチ
7. リセットスイッチ
8. オーディオコネクタ



2015/12/31

TOPPERSプロジェクト認定

8

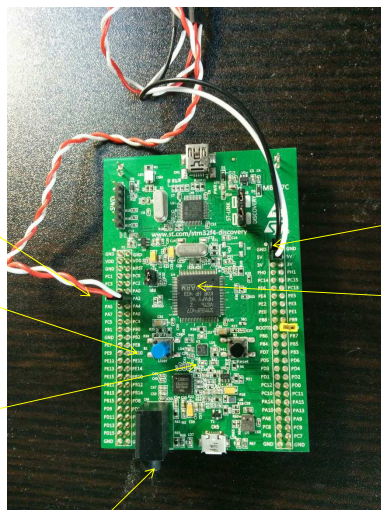


## 実習で使うハードウェア

⑩シリアル通信  
ROMモニタの表示

⑥USERスイッチ  
ポーリングと割込み  
でスイッチの検知

⑤LED  
点灯、点滅、消灯  
の実習



⑨電源供給

①CPU

④USB OTG  
フラッシュROM  
の書き込み

⑧オーディオコネクタ

2015/12/31

TOPPERSプロジェクト認定

9



## プロセッサ (STM32F407) : 概要

- STM製Cortex-M4Fをコアとしたシングルマイコンプロセッサ
  - F:ハードウェア浮動小数点
- ROM:1MB RAM:128KB+64KB
- 周辺回路:メモリマップドI/O方式
  - 入出力ポート :85本
  - タイマ :4 general purpose timer
  - シリアルI/O :2usb OTG/6uart/2CAN  
SPI:3 SSP:3 SDIO
  - A/D :2x12bit ADC/ 2x12bit DAC
  - その他 :オーディオ:専用オーディオPLL  
2系統全二重I2S

2015/12/31

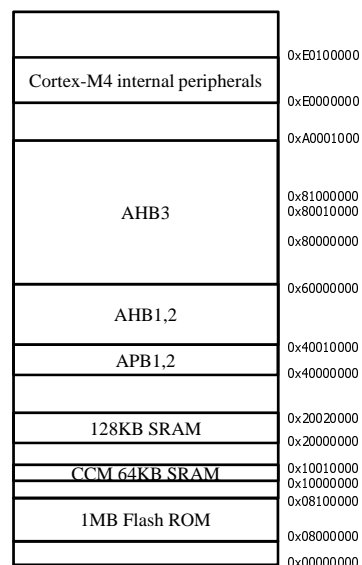
TOPPERSプロジェクト認定

10



## STM32F407: メモリマップ

- 通常は1MBのFlash ROM領域にプログラムを置き、128KB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0x2001F000～0x2001FFFFまでの4KBをROMモニタのデータ領域とし
- 0x20000000～0x2001EFFFをダウンロード領域とする



2015/12/31

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. [USBシリアルケーブルの接続](#)
3. ソフトウェア開発環境の構築

2015/12/31

TOPPERSプロジェクト認定

12



## USBシリアルケーブルの対応

- STM32F4-DiscoveryボードとSTM32-E407はRS232Cコネクタを持たないため、USBシリアルケーブルとピンコネクタを使って、UART通信を行う
- 信号レベルは3.3Vを使用する



2015/12/31

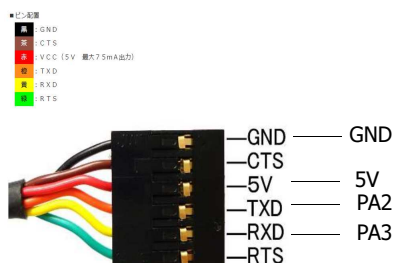
TOPPERSプロジェクト認定

13



## STM32F4-DiscoveryのUART対応

- STM32F4-DiscoveryのコネクタをUSBシリアルケーブルのGND/5V/TXD/RXDにピンケーブルを使って結線します
- PA2/PA3はUSART2用の通信端子です
- 5Vを供給しない場合は5Vを接続する必要はありません



2015/12/31

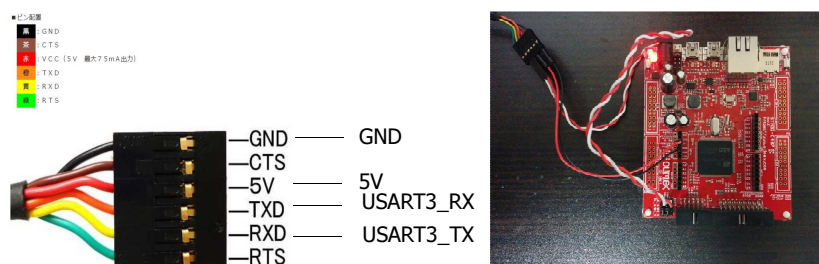
TOPPERSプロジェクト認定

14



## STM32-E407のUART対応

- STM32-E407のコネクタをUSBシリアルケーブルのGND/5V/TXD/RXDにピンケーブルを使って結線します
- USART3\_RX/USART3\_TXはUSART3用の通信端子です
- 5Vを供給しない場合は5Vを接続する必要はありません



2015/12/31

TOPPERSプロジェクト認定

15



## 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. USBシリアルケーブルの接続
3. [ソフトウェア環境の構築](#)

2015/12/31

TOPPERSプロジェクト認定

16





## 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）



2015/12/31

TOPPERSプロジェクト認定

17



## 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)

2015/12/31

TOPPERSプロジェクト認定

18



## 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

2015/12/31

TOPPERSプロジェクト認定

19



## 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

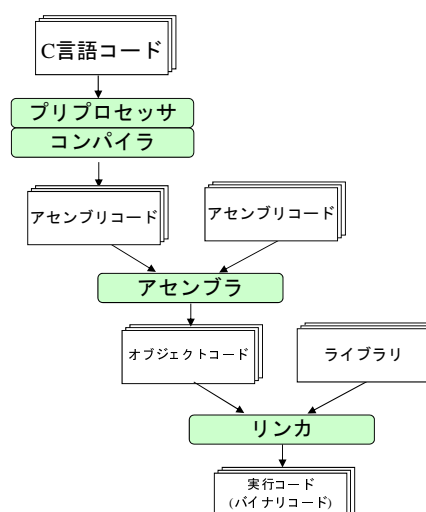
- プリプロセッサされたC言語コードをアセンブリコードへ変換

### 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



2015/12/31

TOPPERSプロジェクト認定

20



## 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

2015/12/31

TOPPERSプロジェクト認定

21



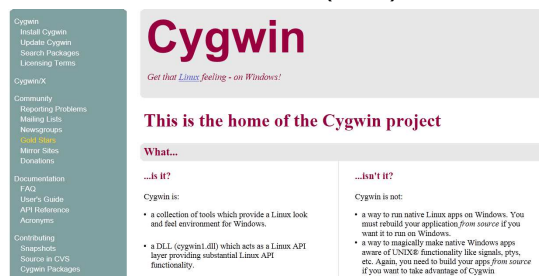
## Cygwinのインストール

- 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



2015/12/31

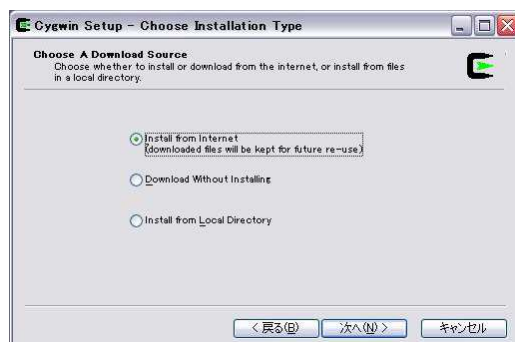
TOPPERSプロジェクト認定

22



## 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



2015/12/31

TOPPERSプロジェクト認定

23



## 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)



2015/12/31

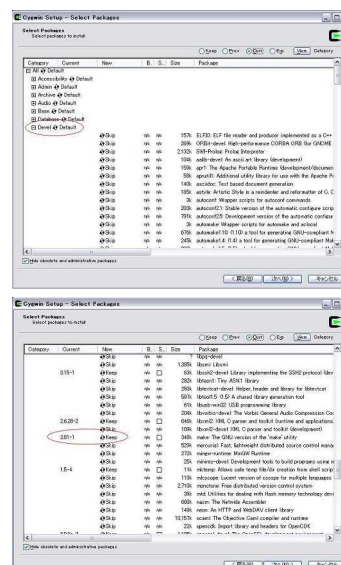
TOPPERSプロジェクト認定

24



## 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



2015/12/31

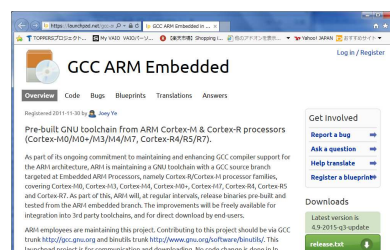
TOPPERSプロジェクト認定

25



## GCC ARM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、手順が複雑なため、ここではGCC ARMのバイナリインストールを行います
- <https://launchpad.net/gcc-arm-embedded/+download>からダウンロードしてください



2015/12/31

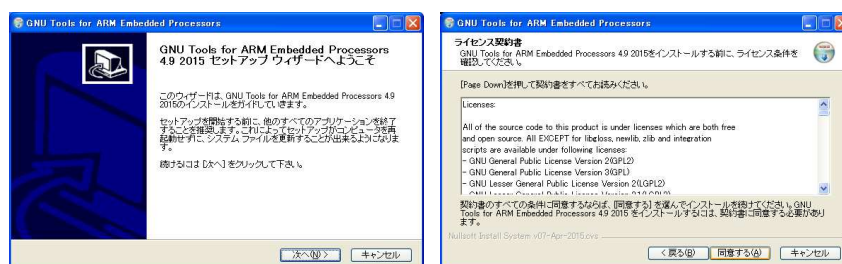
TOPPERSプロジェクト認定

26



## GCC ARMのインストール

- ・ダウンロードしたインストールプログラムを実行する
- ・言語をJapaneseにて、設定の通りインストールします



2015/12/31

TOPPERSプロジェクト認定

27



## GCC ARMのインストール

- ・インストールディレクトリを設定すると、インストールが始まります
- ・Cygwinから実行できるように「Add path to environment variable」をチェックします



2015/12/31

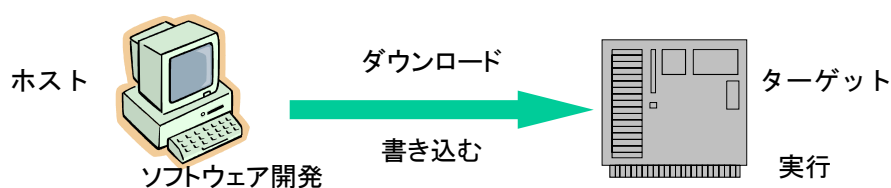
TOPPERSプロジェクト認定

28



## DfuSeDemoのインストール

- STM32F4の各ボード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
  - DfuSeDemo
- DfuSeDemoはSTM32F4ボード内のFlashROMにプログラムを書き込むためのツールです



2015/12/31

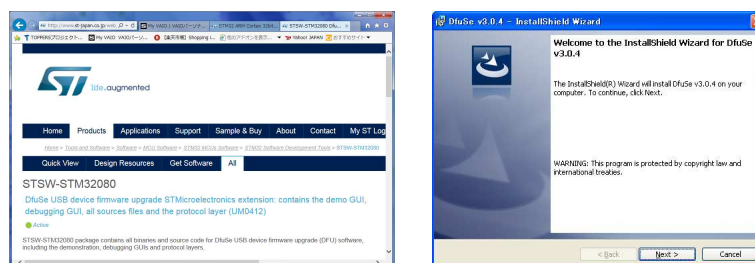
TOPPERSプロジェクト認定

29



## DfuSeDemoのインストール

- STMのサイトよりSTMSW-STM32080をダウンロードする
- Zipファイルを展開して  
DfuSe\_Demo\_V3.0.4\_Setup.exeを実行する。「Welcome」の表示がでたらNextへ



2015/12/31

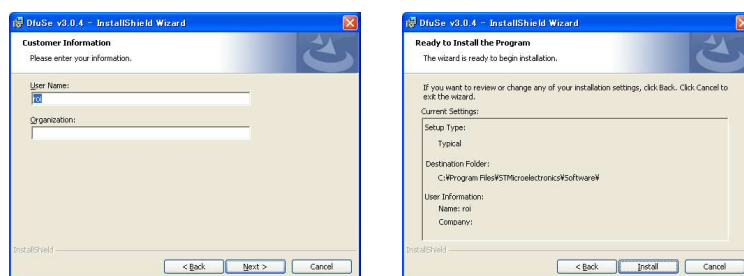
TOPPERSプロジェクト認定

30



## DfuSeDemoのインストール

- Customer Information、「Next」を押す
- Ready to Install the Program、「Next」を押す



2015/12/31

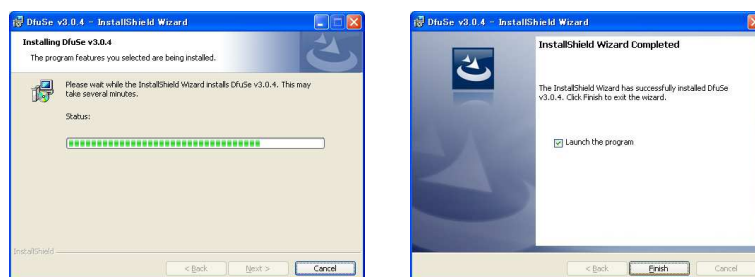
TOPPERSプロジェクト認定

31



## DfuSeDemoのインストール

- インストールが始まります。
- 終了画面で「Finish」を押すとインストールが終了します



2015/12/31

TOPPERSプロジェクト認定

32





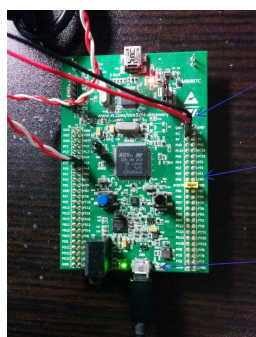
# ROMモニタ

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

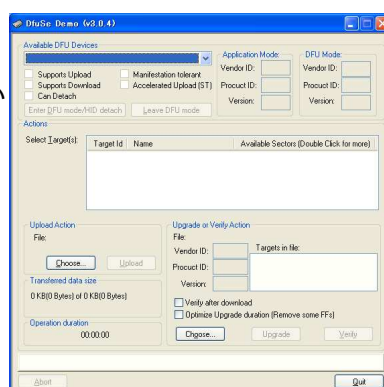
- Windowsのスタートメニューから「DfuSeDemo」を起動する
- USBを接続して、BOOT0とVDDをショートし、RESETキーを押します



電源が入っていることを確認

ショート

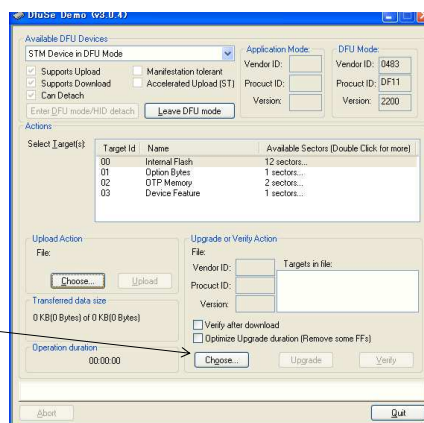
USBを接続



## ROMモニタを選択する

- ボードを認識したら、DFU Mode等の表示がでます
- Chooseでrommon\_discovery.dfsを選択します

Chooseを押して、  
dfuファイルを選択



2015/12/31

TOPPERSプロジェクト認定

35

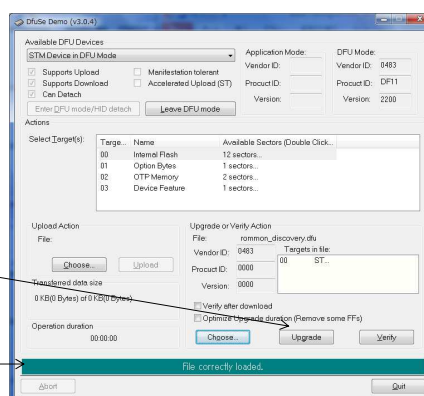


## ROMモニタを書き込む

- UpgradeやVerifyのボタンが選択可能になります
- Upgradeボタンを押して、書き込みを行います

Upgradeを押して、  
書き込み

このバーに進捗が  
表示される



2015/12/31

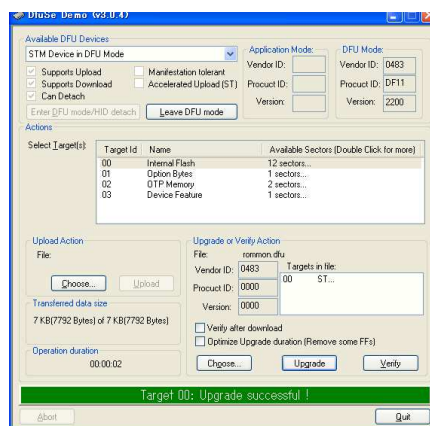
TOPPERSプロジェクト認定

36



## ROMモニタの書き込み

- Successfulの表示で、書き込み終了です。



2015/12/31

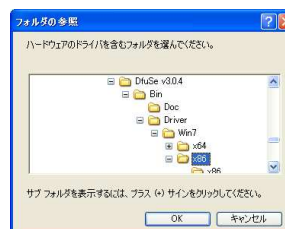
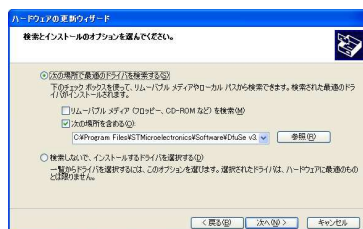
TOPPERSプロジェクト認定

37



## USBドライバが正しくインストールできない場合

- コントロールパネルの「ソフトウェアドライバーの更新」からドライバを更新する
- 次の場所を含めるで、インストール先のBin/Driverからドライバのディレクトリを選択して、ドライバを更新する



2015/12/31

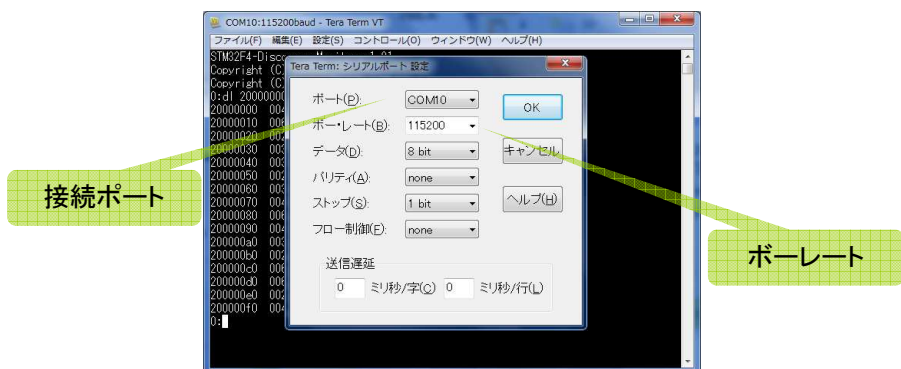
TOPPERSプロジェクト認定

38



## 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



2015/12/31

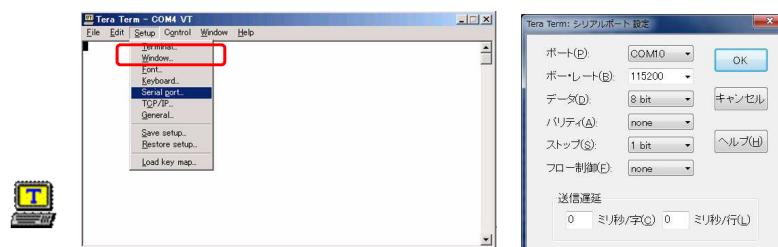
TOPPERSプロジェクト認定

39



## 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 : 115200bps, 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



2015/12/31

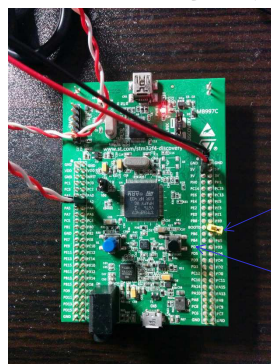
TOPPERSプロジェクト認定

40



## ROMモニタの操作方法

- BOOT0のジャンパーピンを外す
- RESETキー押下で起動します
- 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



2015/12/31

TOPPERSプロジェクト認定

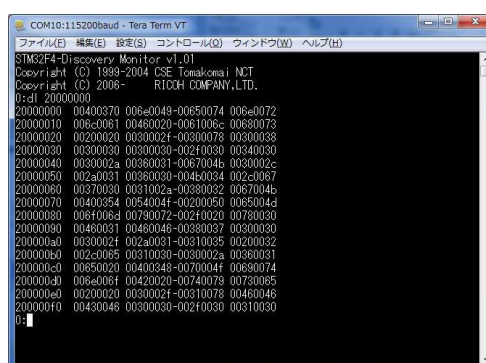
41



## 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



2015/12/31

TOPPERSプロジェクト認定

42



## 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  
0x20000000番地と  
0x20000004番地に  
0x87654321と  
0x12345678を書き込む

```

COM10:115200baud - 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)
20000000 00200020 00300024-00310078 00480048
200000f0 00430048 00300030-002f0030 00310030
0: el 20000000
20000000:00400370:87654321
20000004:008e0049:12345678
20000008:00850074:
0: el 20000000
20000000 87654321 12345678-00650074 006e0072
20000010 006e0061 00480020-0061006c 00680073
20000020 00200020 00300024-00300078 00300038
20000030 00300030 00300030-002f0030 00340030
20000040 0030002a 00380031-0067004b 0030002c
20000050 002a0031 00380030-00460034 002c0067
20000060 00370030 0031002a-00380032 0067004b
20000070 00400354 0054004f-00200050 0085004d
20000080 006f006d 00790072-002f0020 00780030
20000090 00480031 00480048-00380037 00300030
200000a0 0030002f 002a0031-00310035 00200032
200000b0 002e0085 00310030-0030002a 00380031
200000c0 00850020 00400348-0070004f 00680074
200000d0 008e006f 00420020-00740079 00780065
200000e0 00200020 00300024-00310078 00480048
200000f0 00430048 00300030-002f0030 00310030
0:
  
```

2015/12/31

TOPPERSプロジェクト認定

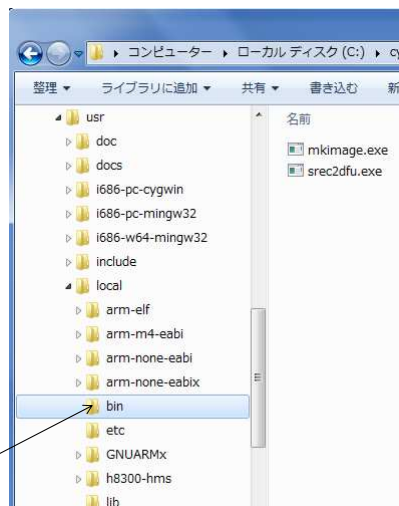
43



## srec2dfuのセットアップ

- SRECファイルをDFUファイルに変換するコマンド  
srec2dfuをインストールする
  - srec2dfuは第一引数にSRECファイル(拡張子srec)を指定するとSRECをDFUに変換し、拡張子dfuファイルをつくる
- srec2dfu.exeを/usr/local/binにすると、コマンドで実行できるようになる

cygwin/usr/local/bin  
にコピー



2015/12/31

TOPPERSプロジェクト認定

44



## STM32F401-NucleoのROMモニタ書き込み

- Nucleoでは、DFU用のUSBコネクタが実装されていないため、ROMモニタを使用する場合は、ST-LINKを用いてrommon\_nucleo.srecをFLASH-ROMに書き込んでください

## 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](#)

## 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



2015/12/31

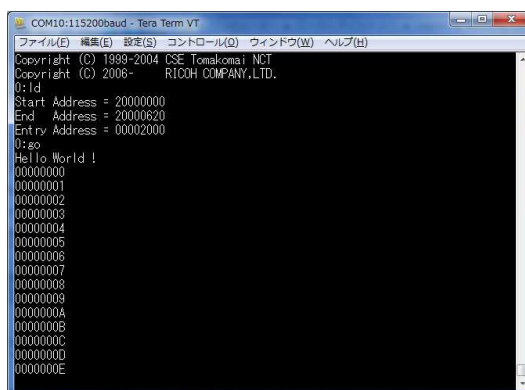
TOPPERSプロジェクト認定

47



## 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



2015/12/31

TOPPERSプロジェクト認定

48





# 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**

## 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**

2015/12/31

TOPPERSプロジェクト認定

51



## タスクモニタの使用方法

- 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] [[lowmask]
        TASK      [cycle time(ms)]
        PORT      [no] [[logno] [portaddress]
Help
```

2015/12/31

TOPPERSプロジェクト認定

52



## 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
- デバイスドライバを設定後、操作してください
- set halfコマンドでLEDが接続されているポートDのポートレジスタ(0x40020c18/0x40020c1a)に直接書き込み、動作を確認する

0x40020c18に0xf000を書き込み：全LED点灯

## 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]

## 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

The screenshot shows the ASP TASK Monitor interface. On the left, three labels with arrows point to specific commands in the terminal: 'display task' points to 'mon/display task', 'set task 4' points to 'set task 4', and 'task activate' points to 'task activate'. The terminal output shows a table of task states and the execution of the activate command for task 6.

```

ASP TASK Monitor Release 1.2.0 for stm32f4-discovery(Cortex-M4) (Nov 29 2015, 18:44:29)
Copyright (C) 2009-2008 by GJ Business Division RICOH COMPANY,LTD. JAPAN
mon/display task
cur id pri state pc stack inistack inisize
* 001 4 WAITING 20004e77 20006830 200065c8 1024
  * 003 3 RUNNING 2000c099 20006830 20006830 2048
  003 10 DORMANT 20000231 00000000 2000d1c8 4096
  004 10 DORMANT 20000231 00000000 2000d1c8 4096
  005 10 DORMANT 20000231 00000000 2000d1c8 4096
  006 5 DORMANT 200007a1 00000000 2000f1c8 4096

mon/set task 6
006(DORMANT)>task activate
execute set_task(6) :: result = E_OK !
Sample program starts (euid = 0).
task1 is running (001).
006(WAITING)>task1 is running (002).
task1 is running (002).
task1 is running (004).
task1 is running (005).
task1 is running (006).
  
```

2015/12/31

TOPPERSプロジェクト認定

55



## 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

The screenshot shows the ASP TASK Monitor interface with a series of 'task terminate' and 'set task' commands being executed. The terminal output shows the results of these commands, including 'E\_OK' and the state of the tasks.

```

006(WAITING)>task terminate
execute ter_task(6) :: result = E_OK !
006(DORMANT)>set task 5
005(RUNNABLE)>task terminate
execute ter_task(5) :: result = E_OK !
005(DORMANT)>set task 4
004(RUNNABLE)>task terminate
execute ter_task(4) :: result = E_OK !
004(DORMANT)>set task 3
003(RUNNABLE)>task terminate
execute ter_task(3) :: result = E_OK !
003(DORMANT)>display task
003(DORMANT)>
cur id pri state pc stack inistack inisize
* 001 4 WAITING 20004e77 20006830 200065c8 1024
  * 003 3 RUNNING 2000c099 20006830 20006830 2048
  003 10 DORMANT 20000231 00000000 2000d1c8 4096
  004 10 DORMANT 20000231 00000000 2000d1c8 4096
  005 10 DORMANT 20000231 00000000 2000d1c8 4096
  006 5 DORMANT 200007a1 00000000 2000f1c8 4096
  
```

2015/12/31

TOPPERSプロジェクト認定

56



## 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となってログ表示しなくなる

log mode 4  
(l m 4)

```

mon/d t
cur id pri state pc stack inistack inisize
001 4 WAITING 20004e77 20006930 200065c8 1024
* mon 3 RUNNING 2000c090 2000c8c8 2048
003 10 DORMANT 20000231 00000000 2000c1c8 4096
004 10 DORMANT 20000231 00000000 2000a1c8 4096
005 10 DORMANT 20000231 00000000 2000a1c8 4096
006 5 DORMANT 200007a1 00000000 2000f1c8 4096

mon/s t 4
004(DORMANT)>s t 6
006(DORMANT)>t a
execute act_task(6) :: result = E_OK !
Sample program starts (exinf = 0).
task1 is running (001). |
006(WAITING)/task1 is running (002). |
(task1 is running (002). |
task1 is running (004). |
mtask1 is running (005). |
task1 is running (006). |
task1 is running (007). |
set logmask=LOG_WARNING logmask=LOG_EMERG !
006(WAITING)>
  
```