

# TOPPERS/ASP for EWARM sampleのコンパイルの仕方

Version 0.9.1

2013/9/08

# 本資料の位置づけ

- TOPPERS/ASP for EWARMの使い方をTOPPERSのsampleで説明したものである.
- EWARM自体の使い方は、マニュアルなどを参照のこと.
  - IARのサイトのFAQ検索などを利用可能
- TOPPERS ASPの使い方は、マニュアルなどを参照のこと.

# TOPPERS ASP for EWARMのメリット

- TOPPERS/ASP をEWARM 6.50.3で使用可能.
- GCCやCYGWIN/Mingwのインストールが不要
- GUIベースで利用で可能
- 32KB制限版でも利用可能
  - サンプル程度は利用可能. 32KBを越える場合は, 正式版が必要.

# フィードバック先

- 本プロジェクトはまだ進行形のプロジェクトであり、改善点も多い。
- 今後のためにフィードバックを希望します。
- また、実際に改善したもの・他ボードに移植して公開してよい場合も、フィードバック頂けると幸いです。

**Email:** [ewarm2013-toppers@yahoo.co.jp](mailto:ewarm2013-toppers@yahoo.co.jp)

or

[users@toppers.jp](mailto:users@toppers.jp)

# 必要なもの

- PC: WindowsXP/Vista/7
- EWARM 6.50.3
  - <http://www.iar.com/jp/Service-center/Downloads/> からダウンロード可能。
- ewgen\_sym.exe
  - cfgで使用するsymsをEWARMの生成したマップから変換
  - 公開フォルダのトップに配置
- cfg.exe
  - Windows用の 1.8.0で動作確認
- EWARM用TOPPERS/ASPリリース
  - 2013/9/8 時点のサポートターゲット
    - ST社: STM32 Discovery: STM32F100
    - Freescale社: Kinetis K70

# 設定及び注意点

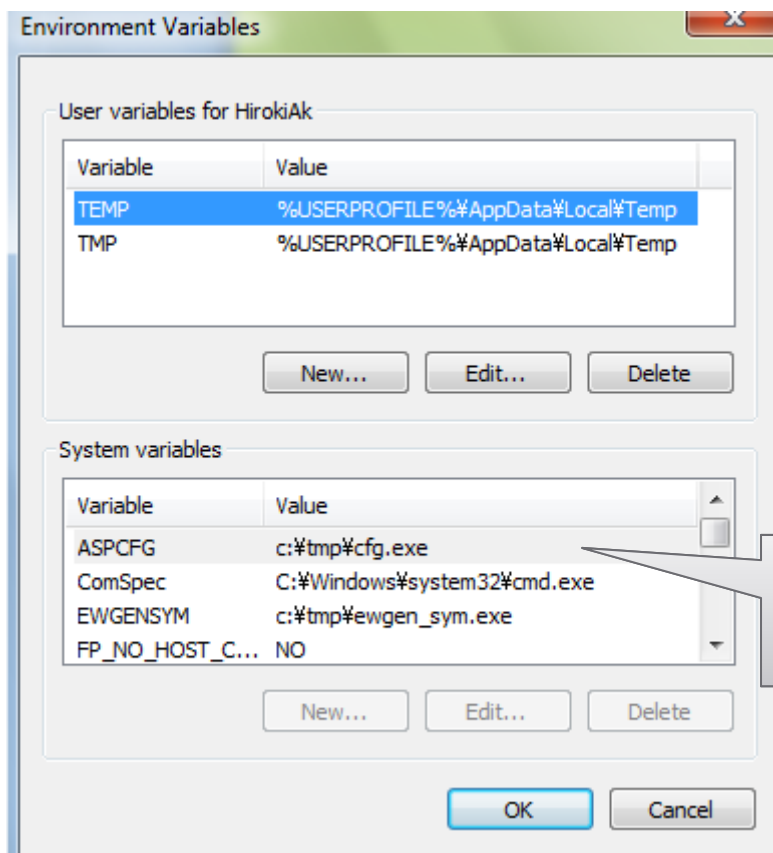
## 環境変数の設定

- `cfg.exe`  
Windowsのシステム環境変数"ASPCFG"でフルパス設定  
– 例) ASPCFG c:¥tmp¥cfg.exe
- `genmap_sym.exe`  
Windowsのシステム環境変数"EWGENSYM"でフルパスにて設定  
– 例 EWGENSYM c:¥tmp¥ewgen\_sym.exe
- 上記コマンドは, dependプロジェクトのプリビルドおよびポストビルドで呼び出して使用している.
  - `pre_depend.bat` および `post_depend.bat`

## 注意点

- 漢字コードはS-JISのみサポート.

# システム変数の設定例



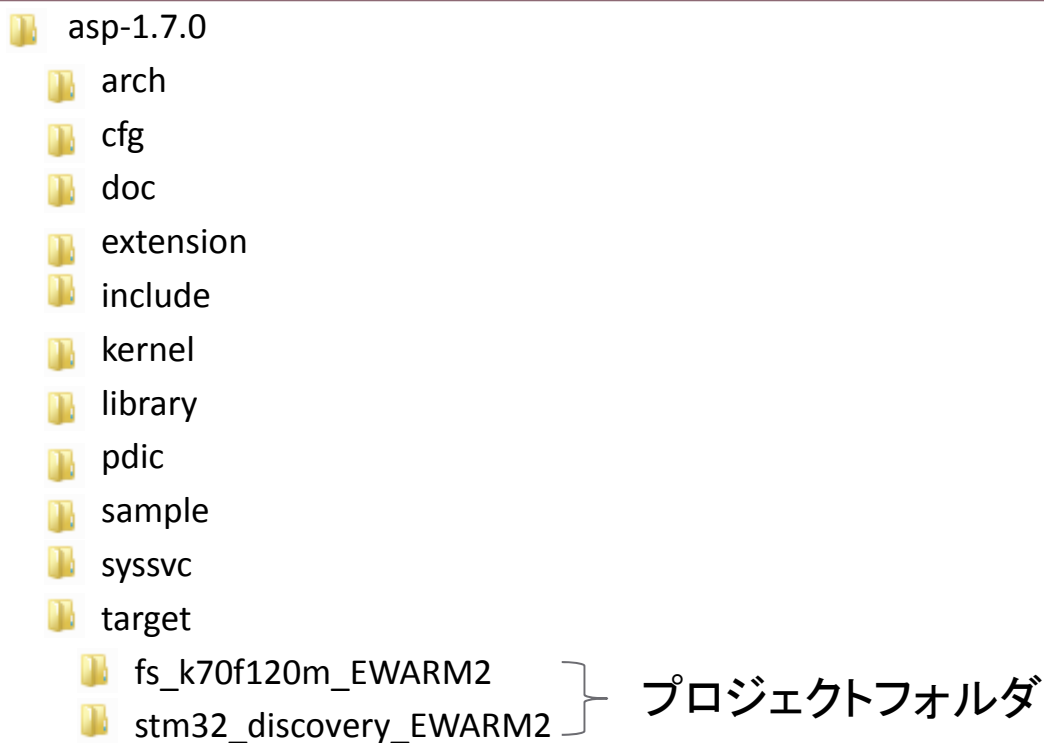
CFG.exeおよびewgen\_sym.exeのパス  
を設定する

# サンプルのコンパイル & 実行

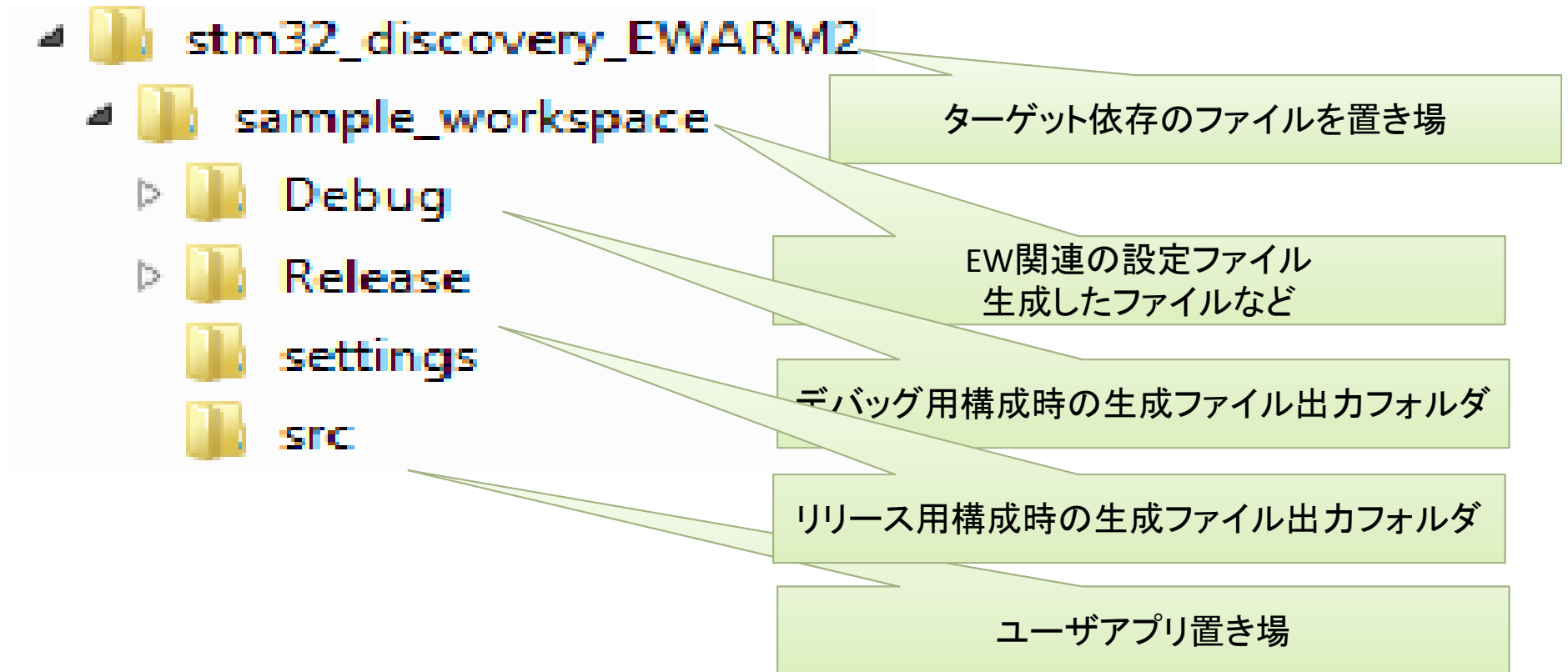


# サンプルのコンパイル & 実行

- STM32VL discoveryとTWR-K70F120Mのコンパイル手順
  - sampleの実行: stm32 value line discovery版
  - sampleの実行: K70 Tower版



# サンプルのフォルダ構成



# コンパイラオプションについて

- 動作確認したのは、
  - 最適化レベル 高 + サイズ
    - 高 + バランスなどは一部動作が規格通りではない可能性がある。

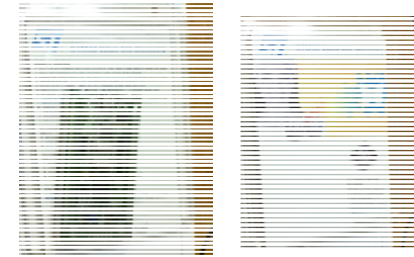
# sampleの実行: stm32 value line discovery版

# sampleの実行: stm32 value line discovery版

1. 使用機材
2. ファイル構成
3. EWARMとワークスペースのオープン
4. プロジェクトのコンパイル
5. 実行
  1. ケーブルの接続
  2. UARTボードの接続
  3. ダウンロード
  4. デバッグなど

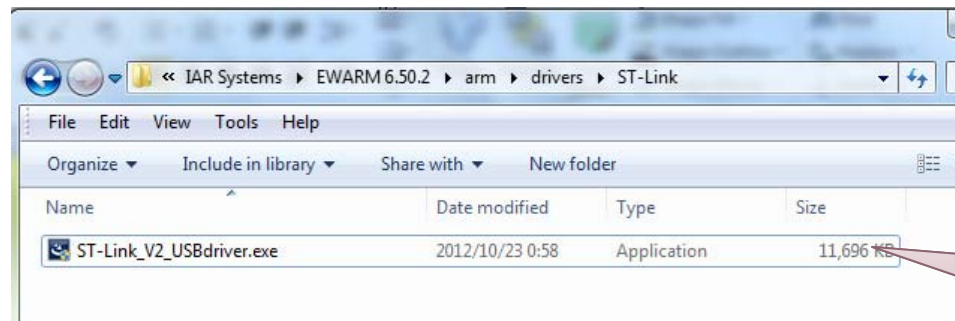
# 1. 使用機材

- STM32 Value line discovery
  - STM32F100 RBT6B
  - \* STM32は様々なDiscoveryが存在
  - \* 現状、動作確認などしているのはこのボードのみ
  - ICEとしてはST-Linkを使用
- FT2232D USBーシリアル2ch変換モジュール
  - 動作確認したものは秋月電子製のモジュール
  - <http://akizukidenshi.com/catalog/g/gM-02990/>
  - 他のUSB-UART変換モジュールでも問題ない
- その他配線用部材
- PC上
  - EWARM(評価版でも可能)
  - ターミナルソフト: TeraTermで確認
  - Cfg.exe (windows版)
  - ewgen\_sym.exe(リリースに含みます)



# その他事前準備

- ICEとして使用するST-Linkのドライバインストール
  - EWARMをインストールしたフォルダの下にインストーラあり  
arm¥drivers¥ST-link¥ST-Link\_V2\_USBdriver.exe



これをインストール

- USB-UART変換のドライバインストール
  - 対応したドライバをインストール

## 2. ファイル構成

- asp-1.7.0
  - arch
  - cfg
  - doc
  - extension
  - include
  - kernel
  - library
  - pdic
  - sample
  - sysvc
  - target
    - stm32\_discovery\_EWARM2
      - sample\_workspace
        - sample.eww
  - test
  - utils

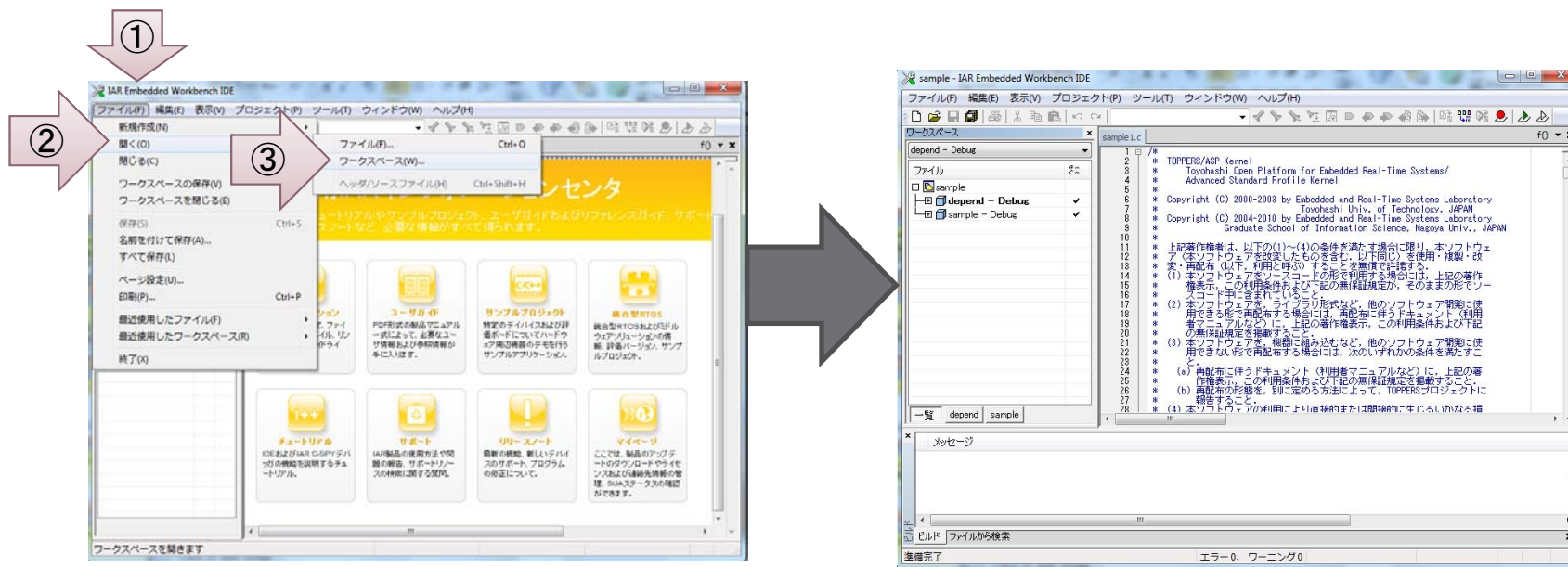
このワークスペースファイルを使用

重要なものだけ記載



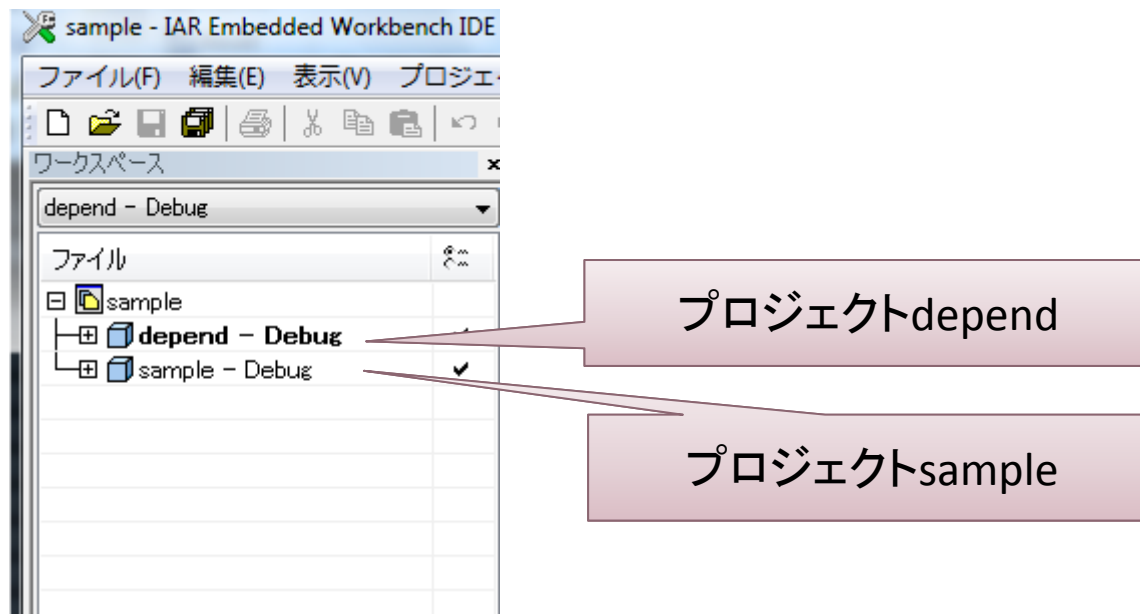
# 3. EWARMとワークスペースのオープン

- EWARMを起動し、①ファイル→②開く→③ワークスペース でsample.eww を開く。



# EWARMのワークスペース

- ワークスペースに2つのプロジェクトが存在
  - プロジェクトdepend,sampleとビルドを実施.
  - 太字のプロジェクトがアクティブとなる.



## 4. プロジェクトのコンパイル

- 以下の順番で処理するために、1つのワークスペースに2つのプロジェクトがある.

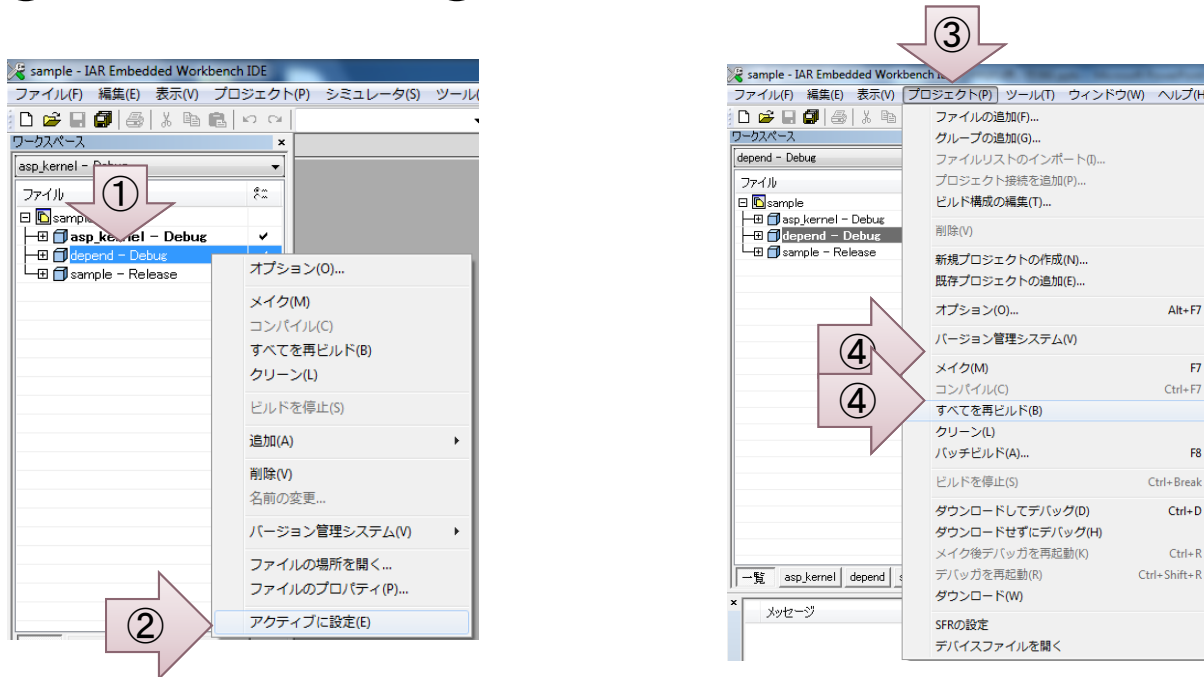
- cfgのpass1
  - sample.cfgからのcfg1\_out.cの作成
- depend プロジェクトのコンパイル
  - cfg1\_out.cのコンパイル
- cfgのpass2
  - kernel\_cfg.c kernel\_cfg.hの生成
- cfgのpass3
  - offset.hの生成
- Sampleプロジェクトのコンパイル

プロジェクトdependで実施  
バッチファイルpre\_depend.batおよび  
post\_depend.batを実行。  
この時に, ewgen\_sym.exeを使用。

プロジェクトsampleで実施

# プロジェクト:dependのコンパイル

1. アクティブプロジェクトに設定
  - ①depend を選択し右クリック → ②アクティブに設定
2. ビルドを実行
  - ③プロジェクト → ④メイクもしくはすべてを再ビルド



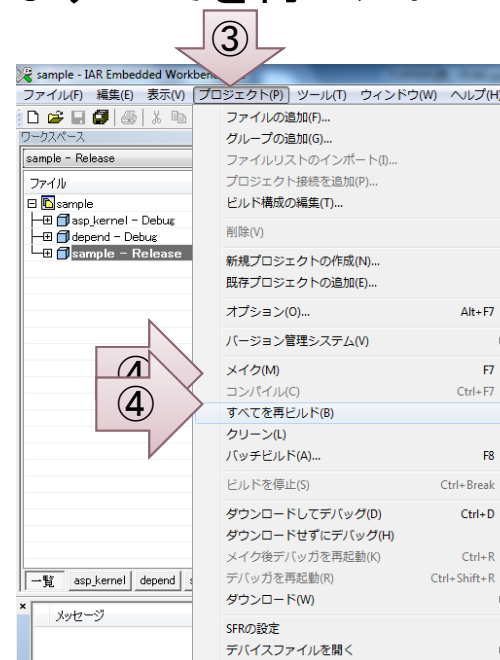
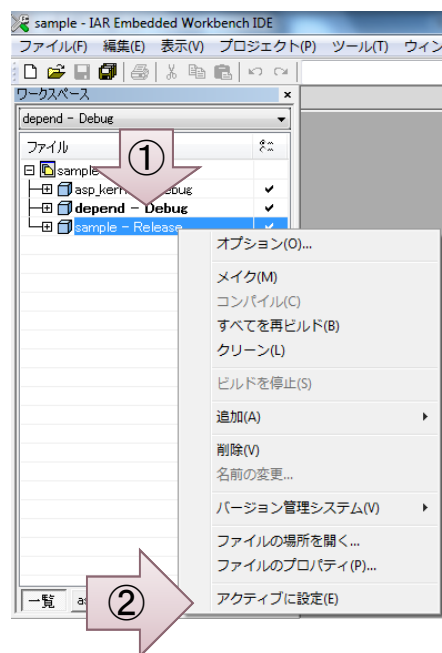
# プロジェクト: sampleのコンパイル

## 1. アクティブプロジェクトに設定

①sample を選択し右クリック → ②アクティブに設定

## 2. ビルドを実行

③プロジェクト → ④メイクもしくはすべてを再ビルド



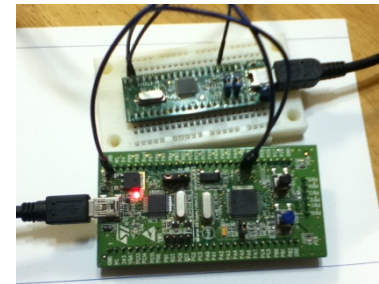
# 5. 実行

1. 接続
2. ターミナルソフトの設定
3. ダウンロード
4. 実行

# 5-1. 接続

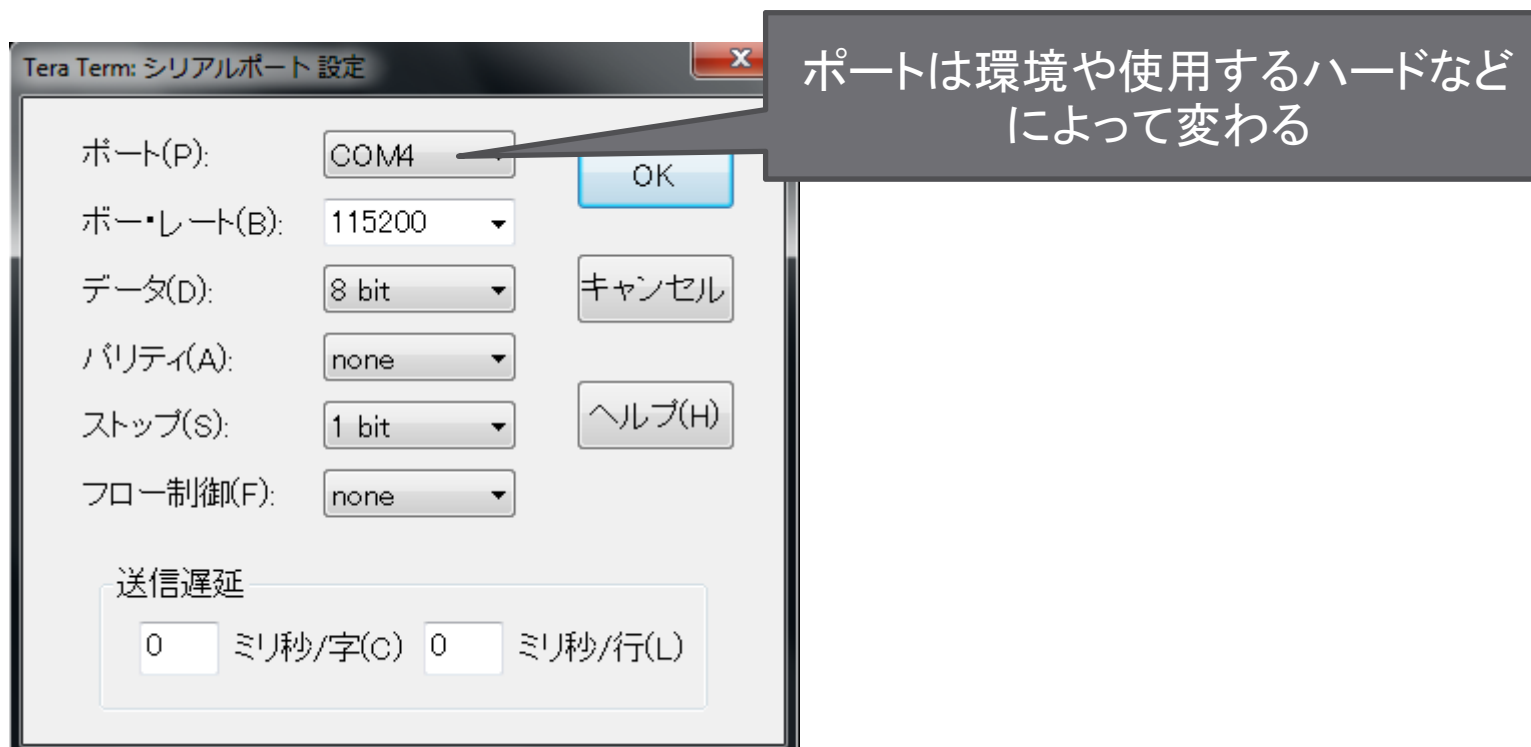
- STM32 Discovery とPCをUSB接続
- USB-UART変換をPCとUSB接続
- STM32 DiscoveryのUART1を使用
  - TX: PA9/RX: PA10 に出力される
  - 通信設定
    - ボーレート: 115200
    - データ: 8ビット
    - パリティ: NONE
    - ストップ: 1ビット
    - フロー制御: NONE

GNDも接続



## 5-2 ターミナルソフトの設定

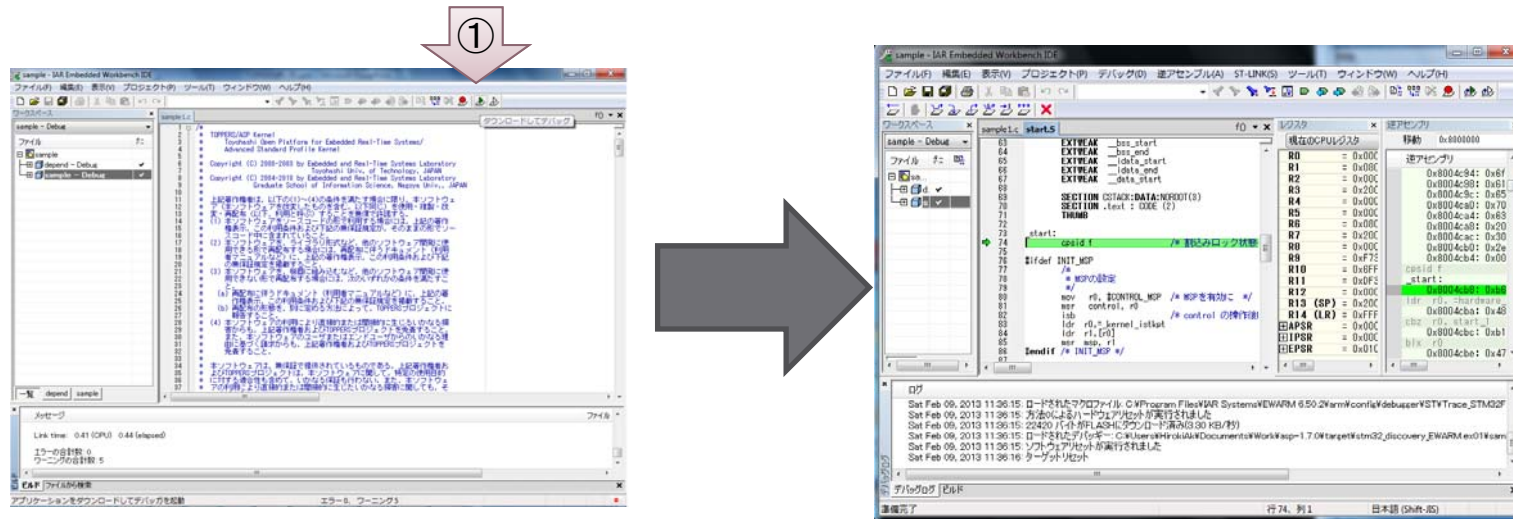
- Tera Termでの設定例





# 5-3 ダウンロード

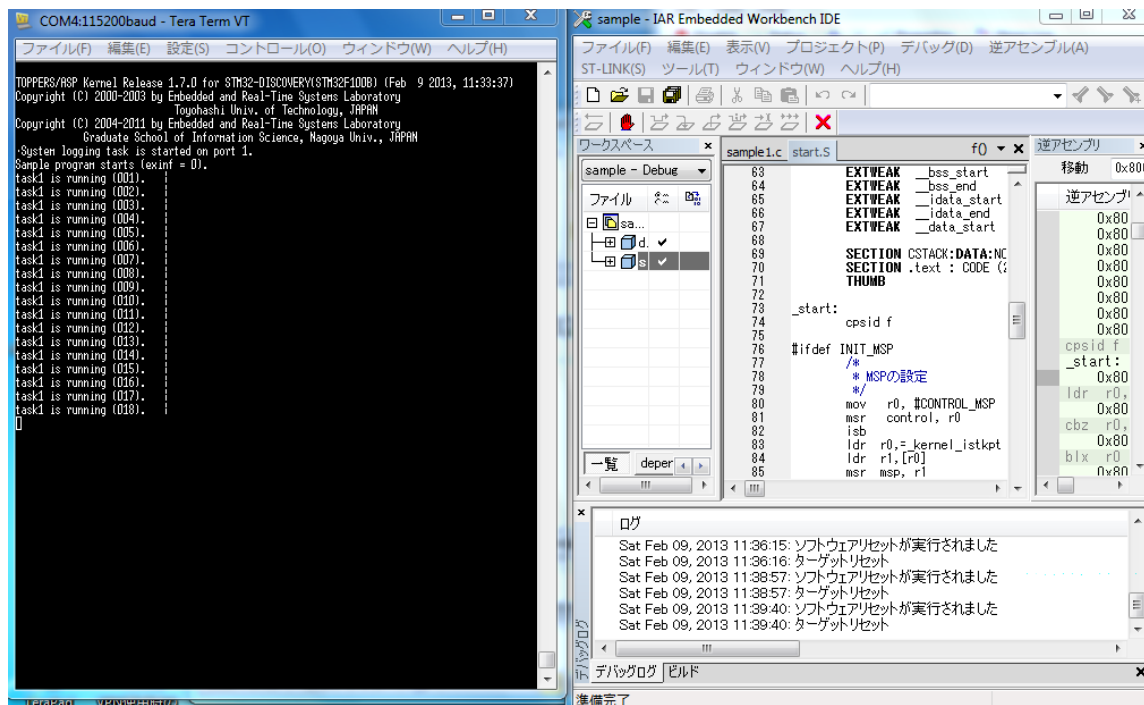
## ①「ダウンロードしてデバッグ」を実行



リセットされた状態で停止する

# 5-4 実行

①実行 を押すと、TeraTerm画面にログが表示される。



# sampleの実行：K70 Tower版

# sampleの実行：K70 Tower版

1. 使用機材
2. ファイル構成
3. EWARMとワークスペースのオープン
4. プロジェクトのコンパイル
5. 実行
  1. ケーブルの接続
  2. UARTボードの接続
  3. ダウンロード
  4. デバッグなど

# 1. 使用機材

- TWR-K70F120M: Kinetis K70 120 MHz Tower System Module
  - PK70FN1M0VMJ12

- PC上

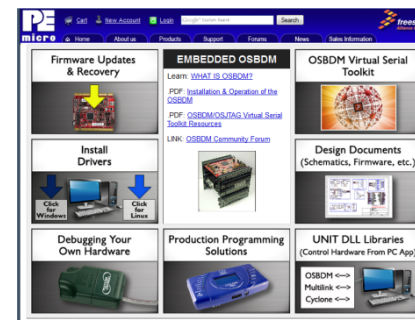
- EWARM(評価版でも可能)
- ターミナルソフト: ここではTeraTermを使用
- Cfg.exe (windows版)
- ewgen\_sym.exe(リリースに含む)



- UARTをUSBで通信するためのP&E OSJTAG Virtual Serial Toolkit

- <http://www.pemicro.com/osbdm/index.cfm>

- ICE: I-jet
- USBケーブル



ここからダウンロード可能

## 2. ファイル構成

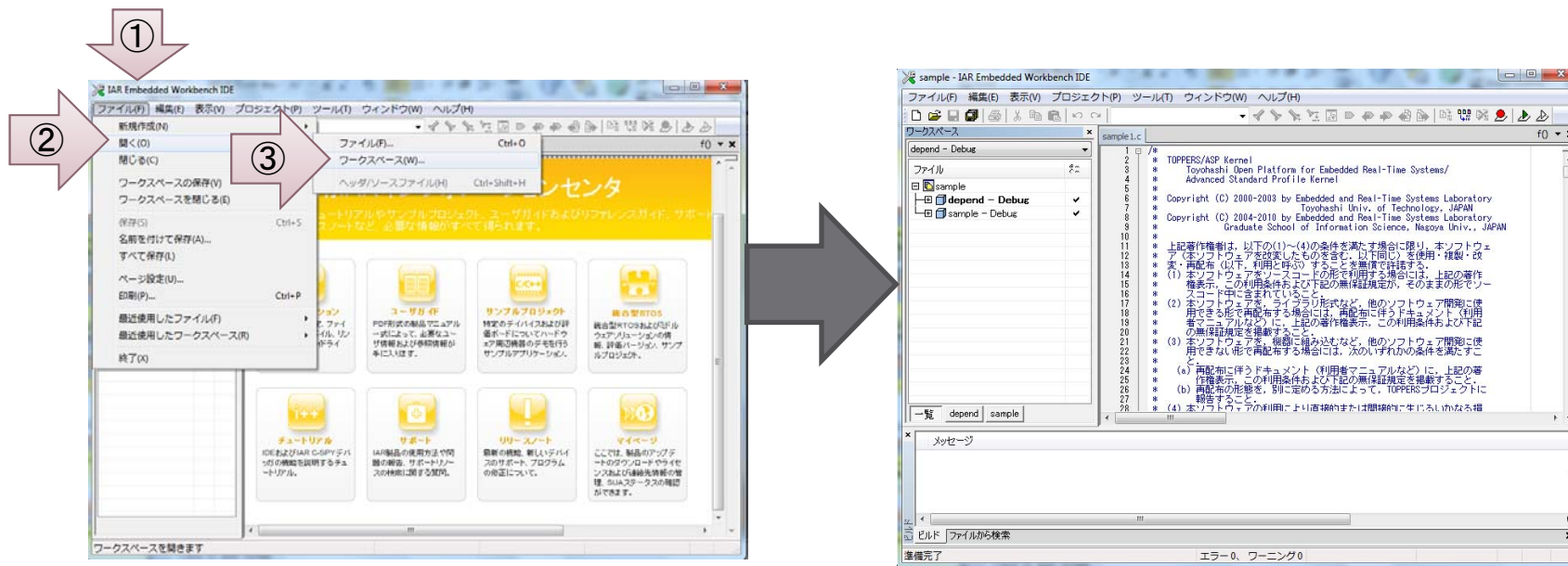
- asp-1.7.0
  - arch
  - cfg
  - doc
  - extension
  - include
  - kernel
  - library
  - pdic
  - sample
  - syssvc
  - target
    - fs\_k70f120m\_EWARM2
      - sample\_workspace
        - sample.eww
      - stm32\_discovery\_EWARM
    - test
    - utils

このワークスペースファイルを使用

重要なものだけ記載

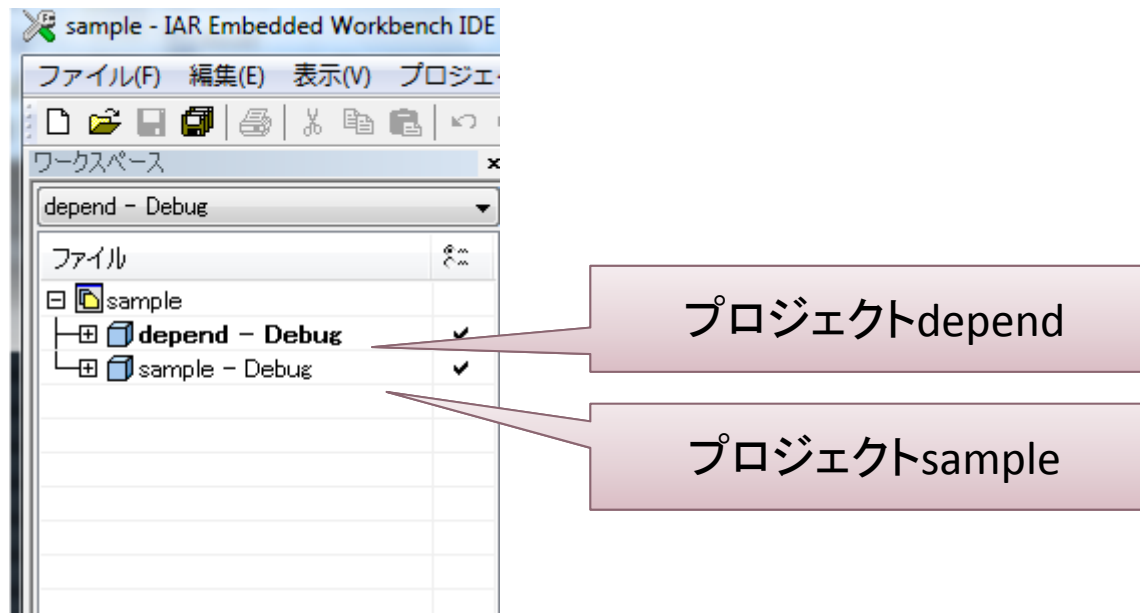
# 3. EWARMとワークスペースのオープン

- EWARMを起動し、①ファイル→②開く→③ワークスペース でsample.eww を開く。



# EWARMのワークスペース

- ワークスペースに2つのプロジェクトがある
  - プロジェクトdepend,sampleとビルドを実施.
  - 太字のプロジェクトがアクティブとなる.





## 4. プロジェクトのコンパイル

- 以下の順番で処理するために、1つのワークスペースに2つのプロジェクトが存在

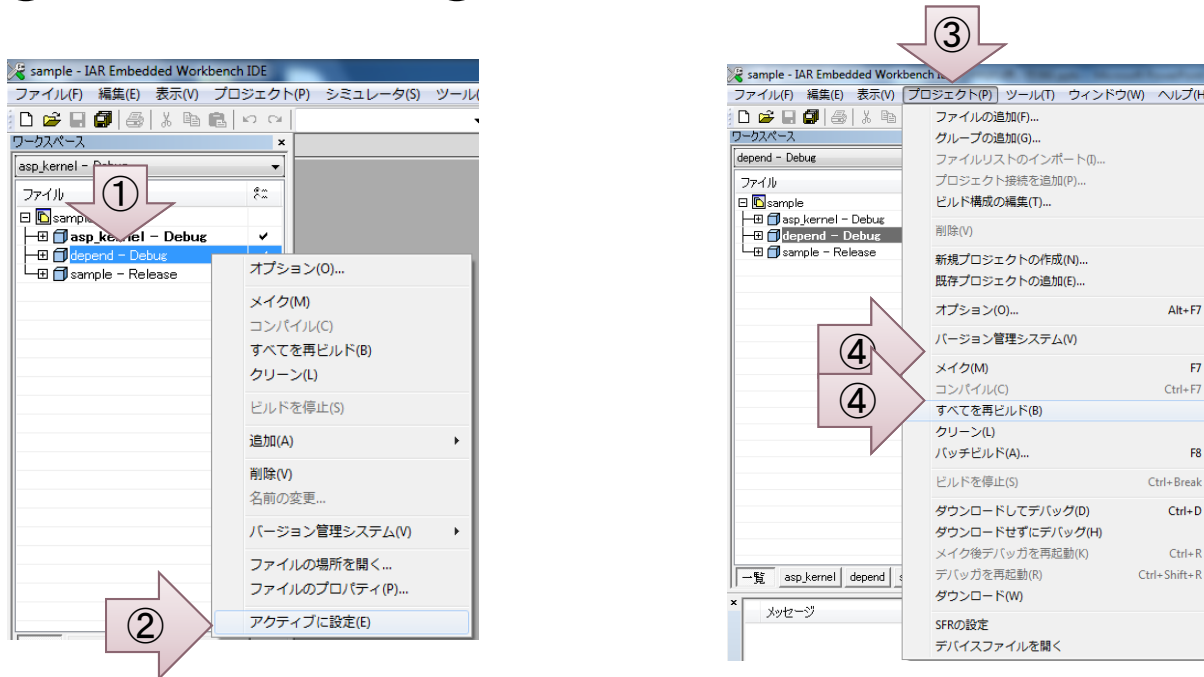
- cfgのpass1
  - sample.cfgからのcfg1\_out.cの作成
- depend プロジェクトのコンパイル
  - cfg1\_out.cのコンパイル
- cfgのpass2
  - kernel\_cfg.c kernel\_cfg.hの生成
- cfgのpass3
  - offset.hの生成
  
- Sampleプロジェクトのコンパイル

プロジェクトdependで実施  
バッチファイルpre\_depend.batおよび  
post\_depend.batを実行。  
この時に, ewgen\_sym.exeを使用。

プロジェクトsampleで実施

# プロジェクト:dependのコンパイル

1. アクティブプロジェクトに設定
  - ①depend を選択し右クリック → ②アクティブに設定
2. ビルドを実行
  - ③プロジェクト → ④メイクもしくはすべてを再ビルド



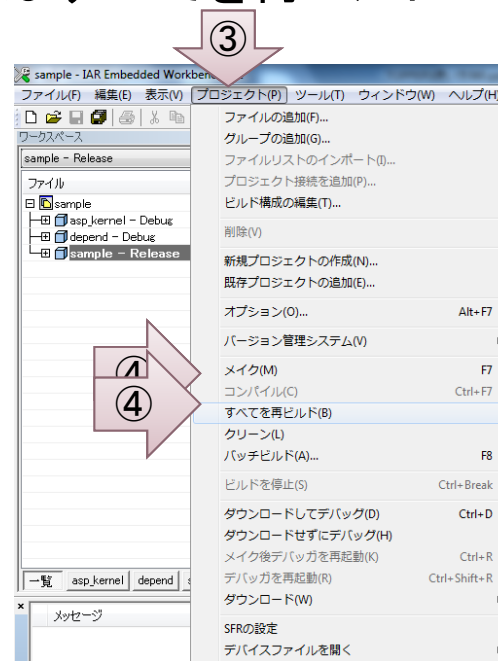
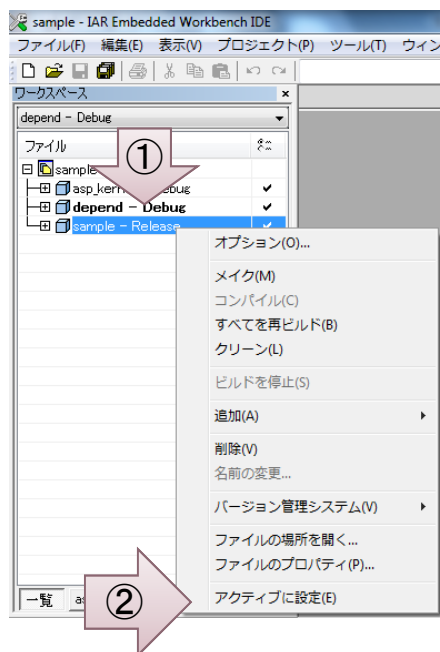
# プロジェクト: sampleのコンパイル

## 1. アクティブプロジェクトに設定

①sample を選択し右クリック → ②アクティブに設定

## 2. ビルドを実行

③プロジェクト → ④メイクもしくはすべてを再ビルド



# 5. 実行

1. 接続
2. ターミナルソフトの設定
3. ダウンロード
4. 実行

## 5-1. 接続

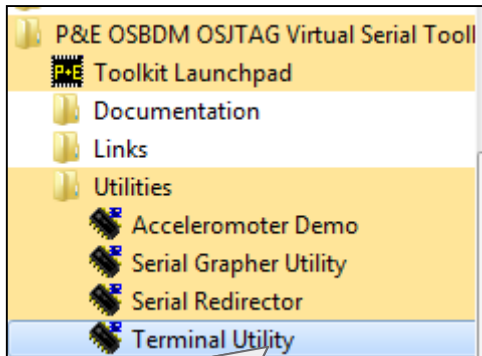
- ① TWR-K70F120MとI-jet接続し、I-jetをPCと接続
- ② TWR-K70F120M とPCでUSB接続



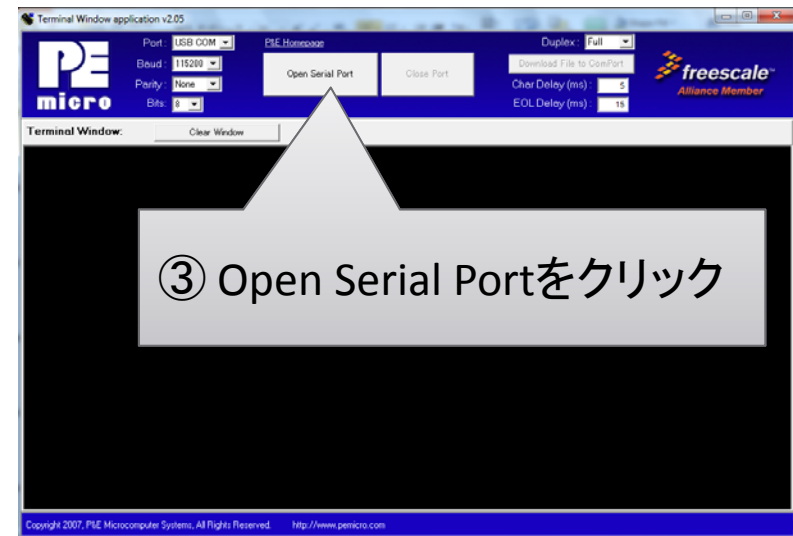
## 5-2 ターミナルソフトの設定

- ① P&E OSBDM Virtual Serial ToolからTerminal Utilityを起動
- ② 設定
- ③ Open Serial Portをクリック

Port: USB COM  
Baud: 115200  
Parity: None  
Bits: 8



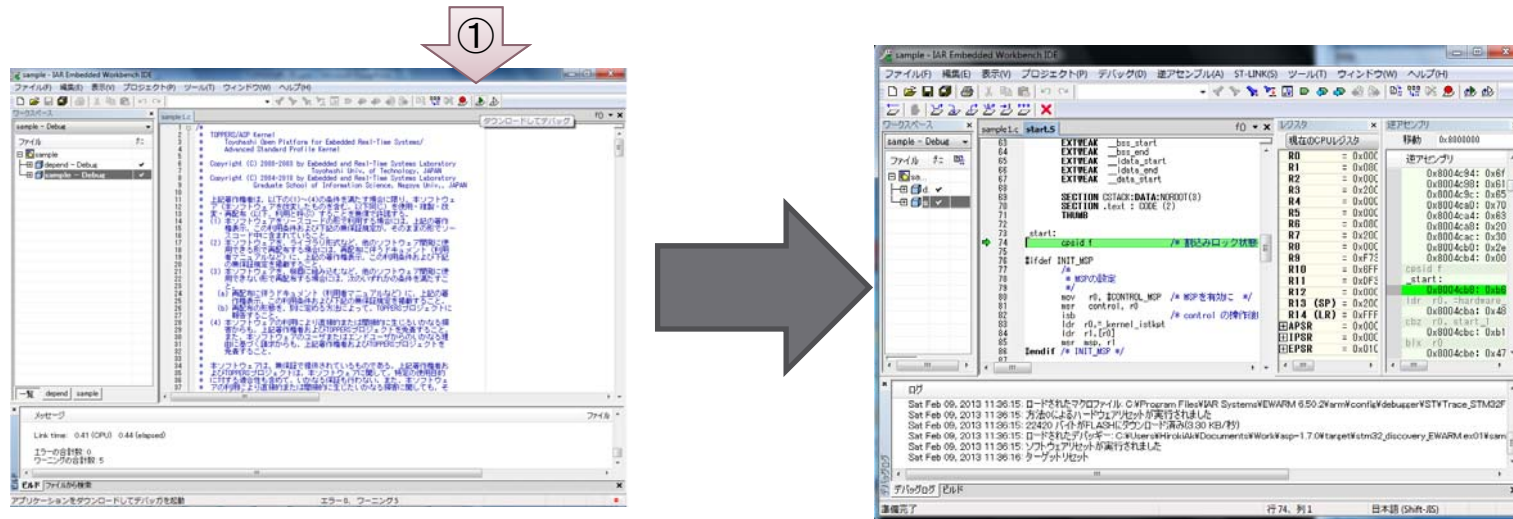
① Terminal Utilityを起動



# 5-3 ダウンロード

プロジェクト: sample を有効にして、

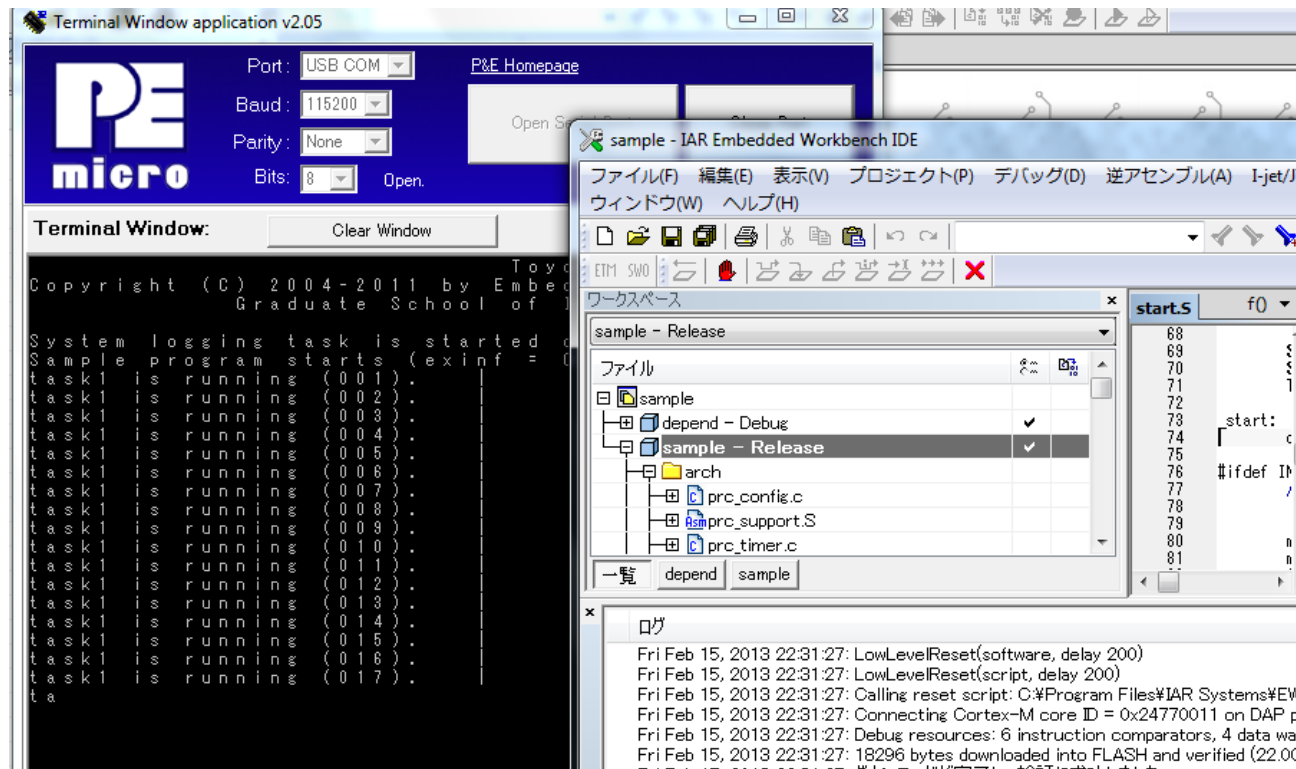
①「ダウンロードしてデバッグ」を実行



リセットされた状態で停止する

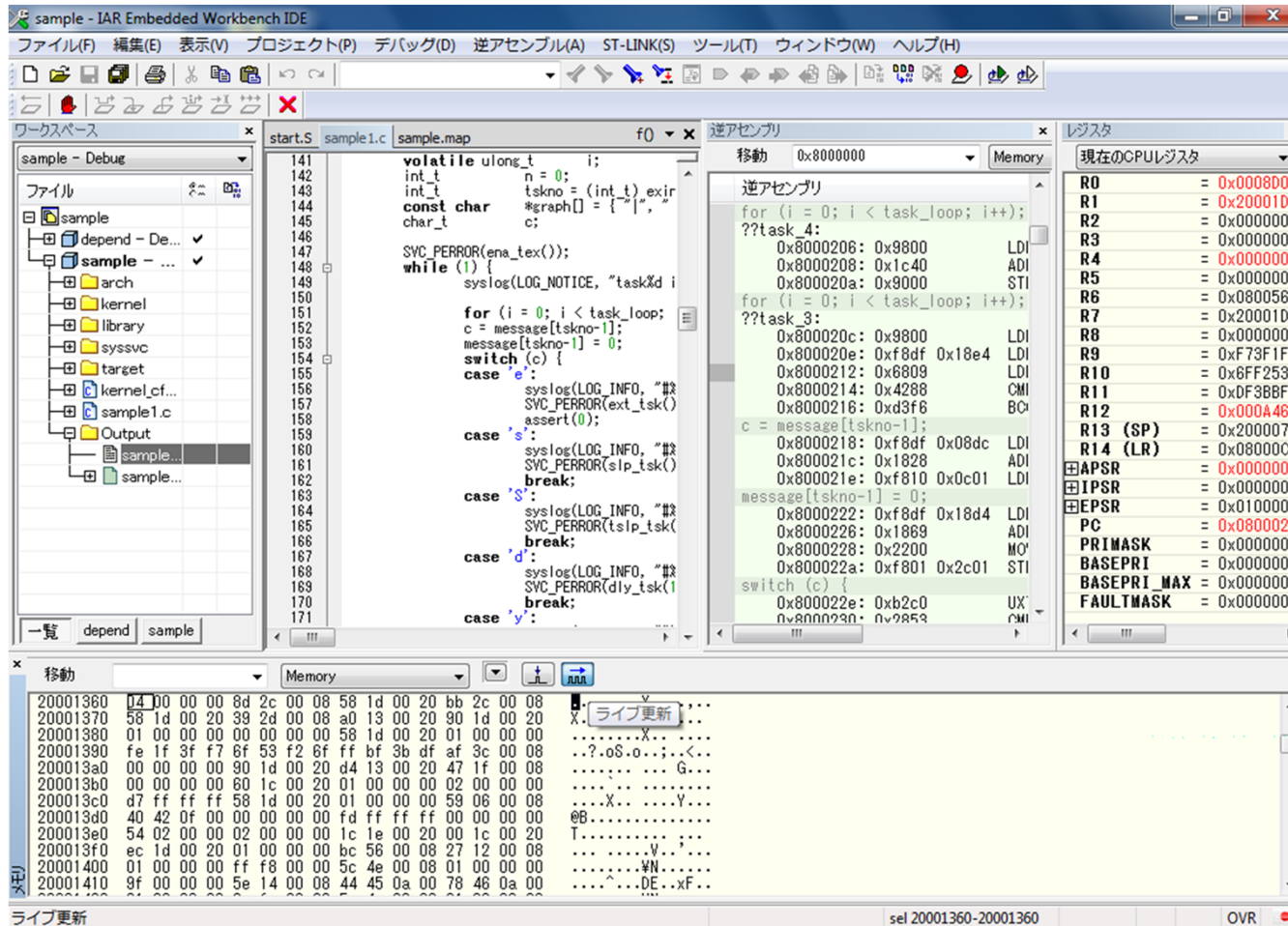
# 5-4 実行

①実行 を押すと、Terminal 画面にログが表示される



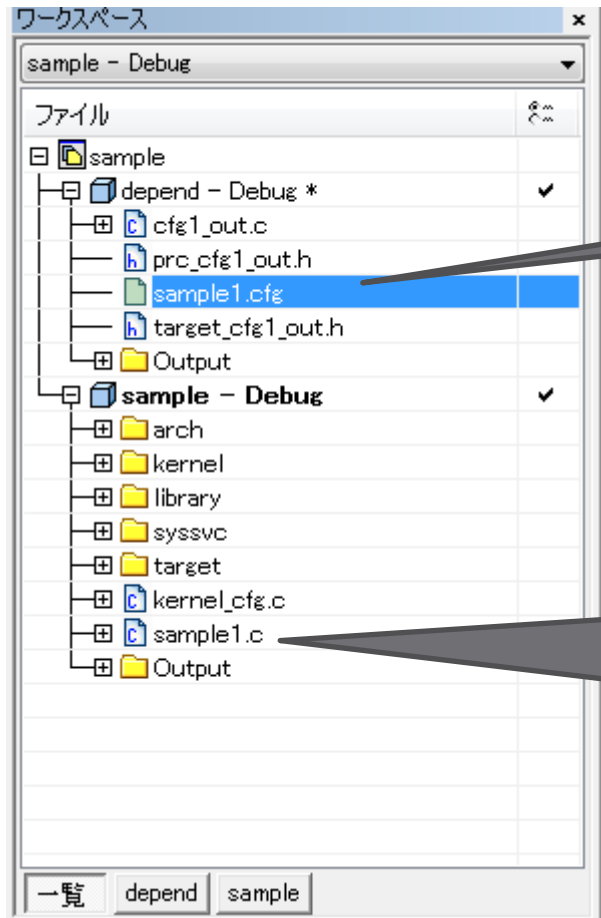


# デバッグ画面の一例



# 新規アプリを作る場合

# 新規のアプリを作る場合



Cfgファイルの内容を変更する

Cfgファイルのファイル名を変更した場合は、  
pre\_build.batとpost\_build.batの変更する  
(詳細は次スライド)

作成したCファイルをここに指定。  
ヘッダのパスはプロジェクトオプションで指定  
する。

# コンフィギュレータでのパスの設定

## pre\_build.bat

```
@ECHO OFF
SETLOCAL

REM アプリのcfgファイルをパス付きで指定
SET APP_CFG=src¥sample1.cfg

SET CURRENT_DIR=%1
...
CHDIR %CURRENT_DIR%

%ASPCFG% --pass 1 --kernel asp -l. -
l..¥..¥..¥.....
ENDLOCAL
```

変更箇所

必要なインクルードパス  
を追加する

## post\_build.bat

```
@ECHO OFF
SETLOCAL

REM アプリのcfgファイルをパス付きで指定
SET APP_CFG=src¥sample1.cfg

SET CURRENT_DIR=%1
SET LIST_DIR=%2
SET EXE_DIR=%3

CHDIR %CURRENT_DIR%
...
%EWGENSYM% -i %LIST_DIR%¥cfg1_out.map -o cfg1_out.syms

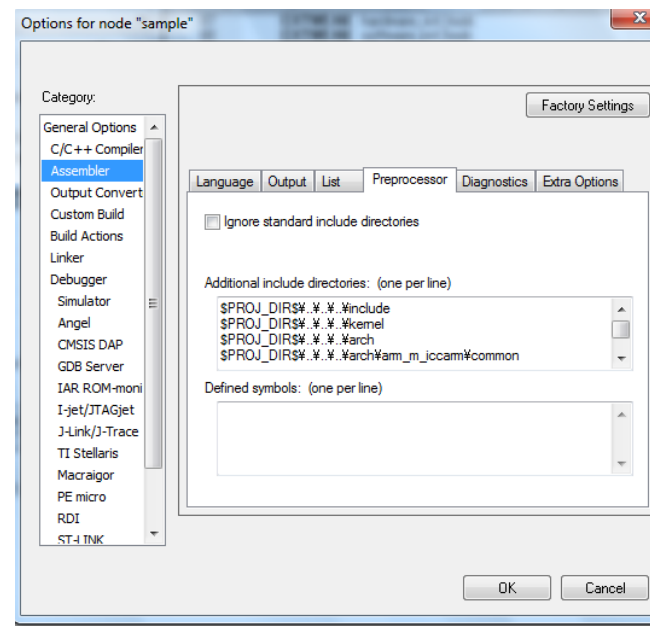
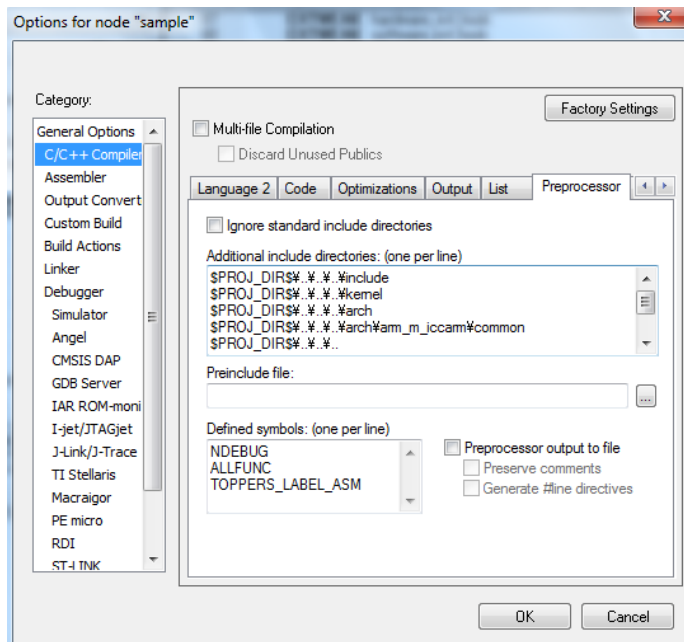
XCOPY /Y %EXE_DIR%¥cfg1_out.srec .¥

%ASPCFG% --pass 2 --kernel asp -l. -l..¥..¥..¥.....
%ASPCFG% --pass 3 --kernel asp -l. -l..¥..¥..¥.....
ENDLOCAL
```

必要なインクルードパス  
を追加する

# コンパイラへのインクルードパスの追加

- プロジェクトsampleのC/C++, アセンブラで必要なインクルードディレクトリを追加.



# 改版履歷

# 改版履歴

- Version 0.5.0 2013/2/9 ASP 1.7.0のSTM32 DISCOVERY用として作成
- Version 0.6.0 2013/2/9 ASP\_KERNELを別コンパイルにした資料に変更
- Version 0.7.0 2013/2/15 K70についてsampleを作成
- Version 0.7.2 2013/2/18 cfg/genmap\_symを変更、ファイル構成追加
- Version 0.9.0 2013/3/13 EWARM 6.50.3で動作確認し改版
- Version 0.9.1 2013/9/8 表現の統一