

# BEAGLEBOARDで QT/EMBEDDED



---

2012.01.21 TOMO

<http://ore-kb.net>



# 目次

- 目的
- Beagleboardって？
- Qt/Embeddedって？
- やりたいこと
- 必要なもの
- 作業の流れ
- ARMのクロスビルド環境を作る
- ARMツールチェーンの導入
- Graphics SDKの導入
- Qt/Embeddedの導入
- ARM用Ubuntuの導入
- BeagleBoardの起動
- SGXライブラリの導入
- サンプルの実行

# BeagleBoardって？

- テキサス・インスツルメンツ社が販売するARMのリファレンス基板
- オープンソース&オープンハード
- 性能の割に安価(BeagleBoard-xMはDigikeyで\$149)
  - Texas Instruments DM3730 (Cortex A8 1GHz)
  - 512MB SDRAM
  - USB 2.0 HOST (HS) × 4
  - USB 2.0 OTG (HS) × 1
  - UART
  - Ethernet (10/100)
  - DVI-D/S-VIDEO/Audio
  - 5V DC
  - <http://beagleboard.org/>
  - <http://ja.wikipedia.org/wiki/BeagleBoard>

# Qt/Embeddedって？

- 組み込み用の軽量なQt
- フレームバッファへ直接で描画するので、ウィンドウシステム不要
- 既存のQtソースをビルドし直すだけで、見た目や操作性(ほぼ)そのまま、組み込み機器で動く！

# やりたいこと

- BeagleBoard-xMでQt(Embedded)アプリを動かしたい！
- Qtは最新の4.8.0にしたい！
- OpenGL(ES)も動かしたい！

# 必要なもの

- BeagleBoard-xM
- 電源用ACアダプタ(5V/1A以上 2.1φセンター**プラス**)
  - USB OTGから電源供給する場合は不要
- 2GB以上のmicroSDカード
  - たまにダメなカードがあります。使用中にエラーが出る場合は、違うカードで試してみましょう
  - BeagleBoard付属の4GB microSDを使ってもいいです
- データコピー用のUSBメモリ
- USBマウス
- USBキーボード
- USB-シリアル変換ケーブル(オス)
- HDMI-HDMIケーブル(DVI-D-HDMIでも可)
- イーサネットケーブル
- インターネット環境
- ビルド用PC

# ARMのクロスビルド環境を作る

- 環境依存を避けるため、母艦となるPCにUbuntu の仮想環境を構築し、その中にクロスビルド環境を作ります
- 無償の仮想環境としては、この2つが有名
  - VMWare Player
  - Oracle VirtualBox ※今回はVirtualBoxで説明します
- Ubuntuのサイトから、「Ubuntu 10.04 LTS (32bit) 」の「仮想ハードディスクイメージ」または「Remix CD」をダウンロード
  - <http://www.ubuntulinux.jp/download/>
  - 新しいバージョンもありますが、SDKが対応してなかったりするので、あえて10.04を選びます
  - 割り当てるメモリは1～2GB、コア数は母艦PCのコア数-1くらい

# 指針

- 今回は、ホームディレクトリに「cross」ディレクトリを作成し、そこにクロスビルド関連のファイルを環境をインストールします
  - ~/cross
    - Qt/4.8.0                      Qt/Embeddedソースコード
    - ti/SDK\_BIN                      TI Graphics SDK
    - arm-2009q1                      ARM用コンパイラ
  - /opt/qt-4.8.0-arm              ARM用Qt/Embeddedライブラリ作成先



# 注意点(1)

- 現行のQtはTIのSGXライブラリ1.6に対応していないので、ソースの一部にパッチを当てる必要がある
  - `$(QTDIR) /src/3rdparty/powervr`
    - `pvr2d.h`
    - `wsegl.h`
  - `$(QTDIR) /src/plugins/gfxdrivers/powervr/QWSWSEGL`
    - `pvrqwssegl.c`

## 参照:

[http://e2e.ti.com/support/dsp/omap\\_applications\\_processors/f/447/t/101221.aspx](http://e2e.ti.com/support/dsp/omap_applications_processors/f/447/t/101221.aspx)

## 注意点(2)

- コンパイル時にプリプロセッサ設定が必要
  - 「QT\_QWS\_CLIENTBLIT」
  - 「QT\_NO\_QWS\_CURSOR」

参照:

```
$ (QTDIR) /src/plugins/gfxdrivers/powervr/QWSW  
SEGL/README
```

# ARMツールチェーンの導入(1)

- 「Sourcery G++ Lite 2009q1-203 for ARM GNU/Linux」から
  - <https://sourcery.mentor.com/sgpp/lite/arm/portal/release858>
- 「IA32 GNU/Linux Installer」をダウンロード
  - インストールの手順は全て、上記リンク先の「Getting Started Guide」で解説されています
  - 新しいバージョンもありますが、TIのSDKではこのバージョン指定です

# ARMツールチェインの導入(2)

インストーラを取得

```
wget
```

```
https://sourcery.mentor.com/sgpp/lite/arm/portal/package4573/public/arm-none-linux-gnueabi/arm-2009q1-203-arm-none-linux-gnueabi.bin
```

```
sudo dpkg-reconfigure -plow dash
```

「いいえ」を選択

# Graphics SDKの導入(1)

インストールスクリプトを取得

```
wget https://github.com/RobertCNelson/stable-  
kernel/raw/master/create_sgx_package.sh
```

実行属性を付加

```
chmod a+x create_sgx_package.sh
```

ダウンロード(30分くらいかかる)

```
./create_sgx_package.sh
```

# Graphics SDKの導入(2)

: (十数分後)

GFX\_4\_03\_00\_02\_libs.tar.gz

GFX\_Linux\_SDK.tar.gz

Graphics\_SDK\_setuplinux\_4\_03\_00\_02.bin

SDK

SDK\_BIN

create\_sgx\_package.sh

「SDK\_BIN」を「~/cross/ti」にコピー

```
mkdir ~/cross/ti
```

```
cp -a SDK_BIN ~/cross/ti
```

GFX\_4\_03\_00\_02\_libs.tar.gzはBeagleBoardで使用するので、一旦USBメモリ (/media/PENDRIVE) へコピー

```
sudo cp GFX_4_03_00_02_libs.tar.gz /media/PENDRIVE/
```

# Qt/Embeddedの導入(1)

Qt 4.8.0 for Embedded Linux のダウンロード

```
wget http://get.qt.nokia.com/qt/source/qt-  
everywhere-opensource-src-4.8.0.tar.gz
```

```
tar zvxf qt-everywhere-opensource-src-  
4.8.0.tar.gz
```

```
mkdir ~/cross/qt
```

```
cp -a qt-everywhere-opensource-src-4.8.0  
~/cross/qt/4.8.0
```

# Qt/Embeddedの導入(2)

## パッチの適用 (1)

```
wget http://ore-kb.net/dl/qt/patch-qt-4.8.0-  
sgx-1.6.tar.gz
```

```
tar zvxf patch-qt-4.8.0-sgx-1.6.tar.gz
```

「~/cross/qt/4.8.0/src/3rdparty/powervr」に  
「patch-qt-4.8.0-sgx-1.6」内の「pvr2d.h, wsegl.h」  
を上書きコピー

このファイルは、SGX SDKのincludeディレクトリ  
「SDK\_BIN/Graphics\_SDK\_setuplinux\_4\_03\_00\_02/inc  
lude」  
にあるファイルと同一



# Qt/Embeddedの導入(3)

## パッチの適用(2)

「~/cross/qt/4.8.0/src/plugins/gfxdrivers/pow  
ervr/QWSWSEGL」に

「patch-qt-4.8.0-sgx-1.6」内の

pvrqwsseg1.c

を上書きコピー

以下のソースを参考に、4.8.0用の差分を作成してます

[#257] qt\_wseg1\_1.6.tar.gz

<https://gforge.ti.com/gf/project/gleslayer/docman/?action=DocmanFileEdit&id=257>

# Qt/Embeddedの導入(4)

BeagleBoard用mkspecの追加

```
wget http://ore-kb.net/dl/qt/beagle-qt.tar.gz  
tar zxvf beagle-qt.tar.gz
```

configure用スクリプト

```
cp beagle-qt/configure-beagle-g1.sh  
~/cross/qt/4.8.0
```

BeagleBoard用mkspec

```
cp -a beagle-qt/linux-beagle-g1-g++  
~/cross/qt/4.8.0/mkspecs/qws/
```

# Qt/Embeddedの導入(5)

x86用gccも必要

```
sudo apt-get install g++
```

configureを実行

```
./configure-beagle-gl.sh
```

ビルド

```
make -j4
```

インストール

```
sudo make install
```

# Qt/Embeddedの導入(6)

動作確認用に一部のサンプルをビルド

```
~/cross/qt/4.8.0/examples/opengl
```

```
qmake
```

```
make -j4
```

```
sudo make install
```

# Qt/Embeddedの導入(7)

BeagleBoardへ持って行くため、圧縮してUSBメモリ  
(/media/PENDRIVE)にコピー

```
cd /opt
```

```
sudo tar cvzf /media/PENDRIVE/qt-4.8.0-  
arm.tar.gz qt-4.8.0-arm
```

# ARM用Ubuntuの導入(8)

Ubuntu 11.10 プレビルドイメージのダウンロード

```
sudo apt-get install xz-utils
wget http://rcn-ee.net/deb/rootfs/oneiric/ubuntu-11.10-
r3-minimal-armel.tar.xz

tar xJf ubuntu-11.10-r3-minimal-armel.tar.xz

ls ubuntu-11.10-r3-minimal-armel
armel-rootfs-201112230927.tar
initrd.img-3.1.6-x6
vmlinuz-3.1.0-psp3
initrd.img-3.1.0-psp3
setup_sdcard.sh
vmlinuz-3.1.6-x6
```

# ARM用Ubuntuの導入(9)

母艦にmicroSDを挿入し、プレビルドイメージを書き込む  
microSDがどのデバイスに割り当たったかは、  
「`sudo fdisk -l`」で確認できる(ここでは`/dev/sdb`)

```
sudo apt-get install uboot-mkimage wget pv  
dosfstools btrfs-tools parted
```

```
sudo ./setup_sdcard.sh --mmc /dev/sdb --uboot  
beagle_xm
```

:

```
Are you 100% sure, on selecting [/dev/sdb] (y/n)? y
```

# BeagleBoardの起動

BeagleBoardのシリアルポートと母艦PCとをUSBシリアルケーブルで接続、TeraTermPro等のターミナルソフトでシリアルポートを開きます。  
(115,200bps)

そして、先ほど作成したmicroSDをBeagleBoardに挿入して電源ON

ユーザ名:ubuntu

パスワード:temppwd

タイムゾーン変更

```
sudo dpkg-reconfigure tzdata
```

Asia→Tokyoの順で選択



# Qtライブラリの導入

USBメモリを挿入

```
mkdir /media/usbmem
```

```
sudo mount /dev/sda1 /media/usbmem
```

```
mkdir ~/work
```

```
cd ~/work
```

```
cp /media/usbmem/qt-4.8.0-arm.tar.gz ./
```

```
tar xzvf qt-4.8.0-arm.tar.gz
```

```
sudo cp -a qt-4.8.0-arm /opt/
```

```
export
```

```
LD_LIBRARY_PATH=/lib:/usr/lib:/usr/local/lib:/opt/qt-4.8.0-arm/lib
```

# WSEGLライブラリの設定

以下の内容で「/etc/powervr.ini」を作成

```
[default]  
WindowSystem=libpvrQSWSEGL.so
```

改行コードはLFのみ

# フレームバッファ割り当ての変更

エディタで「uEnv.txt」を開き、一部編集する

```
sudo vi /boot/uboot/uEnv.txt
```

```
vram=${vram}
```

↓

```
vram=16M omapfb.vram=0:8M,1:4M,2:4M
```

初期値では一部サンプルが動きません。

(example/opengl/textures とか)

FATパーティションなので、Windowsからも編集可能

# SGXライブラリの導入

```
cd ~/work
```

```
tar xvzf
```

```
/media/usbmem/GFX_4_03_00_02_libs.tar.gz
```

```
install-SGX.sh
```

インストールが終わったらリブートする

```
sudo reboot
```

# サンプルの実行

```
cd ~/opt/qt-4.8.0-  
arm/examples/opengl/helloogl_es2
```

```
./helloogl_es2 -qws -display powervr
```

1番目に起動する実行ファイルは、引数に「-qws」を付ける必要があります

# 参考

Beagleboard-xM で遊ぼうのこーなー (ソフト編)

<http://legacyos.homeip.net/beagleboard2/>

SGXドライバのインストール (1)

<http://matsu-world.qee.jp/?p=498>

Try! 組み込みLinux

<http://aspectrt.web.fc2.com/>

OpenGL ES について

<http://www.asahi-net.or.jp/~yw3t-trns/opengl/opengles>

# 参考

Linux OS上のGCC 4でQtを使う

<http://www.02.246.ne.jp/~torutk/cxx/qt/QtOnLinuxAndGCC4.html>

SGXDbg

<http://processors.wiki.ti.com/index.php/SGXDbg>

Building Qt

[http://processors.wiki.ti.com/index.php/Building\\_Qt](http://processors.wiki.ti.com/index.php/Building_Qt)

# 問題・課題

- キーボードが使えない
- マウスが使えない
- QGLWidgetを使うと画面が乱れる



ご清聴ありがとうございました。