

一、qt的版本介绍

按照不同的图形界面来划分，分为四个版本：

Win32版：适用于windows平台；

X11版：适用于各种X系统的Linux和Unix平台；

Mac版：适用于苹果的MacOS；















Embedded版：适用于具有帧缓冲（Frame buffer）的linux平台；

Qtopia是基于qt开发的一个软件平台，Qtopia是构建于Qt/E之上的一系列应用程序，在这个平台上我们可以开发各种应用程序。2008年， TrollTech公司被Nokia收购后，Qtopia被重新命名为Qt Extended。 Nokia在推出了Qt Extended的最新版Qt Extended 4.4.3之后的2009年3月3日，决定停止Qt Extended的后续开发，转而全心投入Qt的产品开发，并逐步会将一部分Qt Extended 的功能移植到Qt的开发框架中。

所以总的来说，QT也就三种：面向桌面的x11、面向嵌入式的Qt/E、以及面向嵌入式带各种应用程序的Qtopia桌面系统！！

比如是Qt的各种版本介绍：

1.qt的安装程序（包含了QtCreator， QtAssistant）

 qt-opensource-windows-x86-msvc2012_64_opengl-5.2.1.exe	04-Feb-2014 16:14	553M	Details
 qt-opensource-windows-x86-msvc2012_64-5.2.1.exe	04-Feb-2014 16:12	556M	Details
 qt-opensource-windows-x86-msvc2012-5.2.1.exe	04-Feb-2014 16:09	541M	Details
 qt-opensource-windows-x86-msvc2010_opengl-5.2.1.exe	04-Feb-2014 16:07	517M	Details
 qt-opensource-windows-x86-msvc2010-5.2.1.exe	04-Feb-2014 16:05	518M	Details
 qt-opensource-windows-x86-mingw48_opengl-5.2.1.exe	04-Feb-2014 16:03	634M	Details
 qt-opensource-windows-x86-android-5.2.1.exe	04-Feb-2014 16:01	1.1G	Details
 qt-opensource-mac-x64-ios-5.2.1.dmg	04-Feb-2014 15:58	909M	Details
 qt-opensource-mac-x64-clang-5.2.1.dmg	04-Feb-2014 15:55	378M	Details
 qt-opensource-mac-x64-android-ios-5.2.1.dmg	04-Feb-2014 15:54	1.3G	Details
 qt-opensource-mac-x64-android-5.2.1.dmg	04-Feb-2014 15:49	851M	Details
 qt-opensource-linux-x86-android-5.2.1.run	05-Feb-2014 11:31	848M	Details
 qt-opensource-linux-x86-5.2.1.run	05-Feb-2014 11:29	371M	Details
 qt-opensource-linux-x64-android-5.2.1.run	04-Feb-2014 15:47	846M	Details
 qt-opensource-linux-x64-5.2.1.run	04-Feb-2014 15:44	368M	Details

Windows--x86--msvc2012_64_opengl :表示window平台，msvc2012可以与VS2012进行结合使用，64位应用程序，要求电脑装的是64位操作系统；opengl表示支持openGL的绘图模式
Windows--x86--mingw48_opengl : window平台下的mingw48编译器进行编译。

2.Qt的库的源码包

可以根据需要下载源码包：

↑ Parent Directory	-		
📁 qtxmlpatterns-opensource-src-5.2.1.zip	04-Feb-2014 16:29	3.1M	Details
📁 qtxmlpatterns-opensource-src-5.2.1.tar.xz	04-Feb-2014 16:29	1.3M	Details
📁 qtxmlpatterns-opensource-src-5.2.1.tar.gz	04-Feb-2014 16:29	1.8M	Details
📁 qtxmlpatterns-opensource-src-5.2.1.7z	04-Feb-2014 16:29	1.3M	Details
📁 qtx11extras-opensource-src-5.2.1.zip	04-Feb-2014 16:29	52K	Details
📁 qtx11extras-opensource-src-5.2.1.tar.xz	04-Feb-2014 16:29	29K	Details
📁 qtx11extras-opensource-src-5.2.1.tar.gz	04-Feb-2014 16:29	36K	Details
📁 qtx11extras-opensource-src-5.2.1.7z	04-Feb-2014 16:29	29K	Details
📁 qtwinextras-opensource-src-5.2.1.zip	04-Feb-2014 16:29	1.0M	Details
📁 qtwinextras-opensource-src-5.2.1.tar.xz	04-Feb-2014 16:29	784K	Details
📁 qtwinextras-opensource-src-5.2.1.tar.gz	04-Feb-2014 16:29	881K	Details
📁 qtwinextras-opensource-src-5.2.1.7z	04-Feb-2014 16:29	782K	Details
📁 qtwebkit-opensource-src-5.2.1.zip	04-Feb-2014 16:29	67M	Details

3.Qt 安装包的下载

下载网址：<http://qt-project.org/downloads>

如果想下载Qt的以前版本，可以在上面网址的最下面的一栏有个Archive for old versions，进去就可以下载qt的以前版本了。



Other downloads

- ↓ **Visual Studio Add-in 1.1.11 for Qt4** (supports versions 2005, 2008 and 2010, does not work with the Express edition)
- ↓ **Visual Studio Add-in 1.2.2 for Qt5** (supports versions 2008, 2010 and 2012, does not work with the Express edition)
- ↓ **Visual Studio Add-in 1.2.3 Alpha for Qt5** (supports versions 2008, 2010, 2012 and 2013, does not work with the Express edition)
- ↓ **Qt Installer Framework 1.4.0**
- ↓ **Qt Installer Framework 1.5.0**
- ↓ **Qt repositories at qt.gitorious.org**
- ↓ **Archive for old versions**

二、嵌入式Qt开发环境的搭建

1. 预备知识

嵌入式qt开发环境的安装方法很多，qt的版本也很多。有点让人不知所措，不知该按哪种方法去安装。刚开始总不知其几个版本之间的关系。现在也只知道其大概的一个关系，下面理清一下：据我所知的qt版本有：

- 1、t-x11-opensource-src-4.3.2.tar.gz
- 2、qt-embedded-linux-opensource-src-4.4.0-rc1.tar.bz2
- 3、qt-everywhere-opensource-src-4.8.5.tar.gz
- 4、QtSdk-offline-linux-x86-v1.2.1.run
- 5、qt-creator-linux-x86-opensource-2.5.0.bin

这些版本真有些让人眼花缭乱。后经过一番的研究和请教，终于对这几个版本之间的关系有点眉目了。

首先网上有人说安装 1和2，就可以了，因为1可以提供Qt designer 和qybf，然后2可以提供嵌入式编译库，两者安装后就可以搭建一个嵌入式qt开发环境了，其实没错，不过已经过时了，因为现在有了Qt creator。Qt creator 就包含了 Qt designer和qybf，因此我们仅仅安装5的版本就可以替代1的安装。不过2也还是要安装，不过了，像2那种版本的编译库现在已经有了更加齐全的库替代，那就是3了。而4就是5的升级版了，它不仅包含5拥有的所有东西，而且能开发嵌入式应用程序，不过暂时还没测试。

构建Qt/Embedded的交叉编译环境需要3个Qt开发环境：

Qt的桌面开发环境：在linux中全安装qt，用它编译好的程序后，可以直接用“./”运行。

Qt/Embedded(x86)环境：在宿主机平台模拟编译和调试目标机程序的环境，带有qvfb功能，在PC机上可以模拟应用程序在arm开发板上的运行情况。用此编译好的程序，只用“./”命令不可以运行，必须加上参数 -qvfb -qws。

Qt/Embedded (ARM) 交叉编译环境：编译好的程序，用于在arm平台上运行。

首先要准备好交叉编译所需要的软件包：

- (1) arm-linux-gcc-4.4.3-20100728.tar.gz (交叉编译环境)
- (2) qt-everywhere-opensource-src-4.8.5.tar.gz (Qt的源码)
- (3) qt-creator-linux-x86_64-opensource-2.5.2.bin (我的系统是64位，所以选择x86_64)
- (4) tslib-1.4.tar.gz (触摸屏校验工具)

其中 (2) 和 (3) 在<http://download.qt-project.org/archive/> 进行下载。

2.环境的搭建

2.1 系统环境

主机操作系统：ubuntu10.04

主机编译器：gcc4.4.3,

交叉编译器：arm-linux-gcc-4.3.3

2.2 安装交叉编译器

(1)首先以root用户登入

(2)复制arm-linux-gcc-4.3.2.tgz到根目录下tmp文件夹里

(3)解压命令tar xvzf arm-linux-gcc-4.3.2 -C /

注意以上命令必须要有-C而且是大写的后边有个空格也要注意

(4)配置下编译环境路径 在控制台下输入 gedit /root/.bashrc 等一会出来文本编辑器后在文件最后 (最后一行) 加上 下面代码

```
export PATH=/opt/FriendlyARM/toolschain/4.4.3/bin:$PATH
```

重新打开一个终端，使上面的配置文件生效；输入命令 arm-linux-gcc -v; 如果出现下面界面，就代表交叉编译器安装成功，会输出gcc的版本号；

```

root@lppwrrr:~# arm-linux-gcc -v
Using built-in specs.
Target: arm-none-linux-gnueabi
Configured with: /opt/FriendlyARM/mini2440/build-toolschain/working/src/gcc-4.4.3/configure --build=i386-build_redhat-linux-gnu --host=i386-build_redhat-linux-gnu --target=arm-none-linux-gnueabi --prefix=/opt/FriendlyARM/toolschain/4.4.3 --with-sysroot=/opt/FriendlyARM/toolschain/4.4.3/arm-none-linux-gnueabi//sys-root --enable-languages=c,c++ --disable-multilib --with-arch=armv4t --with-cpu=arm920t --with-tune=arm920t --with-float=soft --with-pkgversion=ctng-1.6.1 --disable-sjlj-exceptions --enable-cxx-exceptions --with-gmp=/opt/FriendlyARM/toolschain/4.4.3 --with-mpfr=/opt/FriendlyARM/toolschain/4.4.3 --with-ppl=/opt/FriendlyARM/toolschain/4.4.3 --with-cloog=/opt/FriendlyARM/toolschain/4.4.3 --with-mpc=/opt/FriendlyARM/toolschain/4.4.3 --with-local-prefix=/opt/FriendlyARM/toolschain/4.4.3/arm-none-linux-gnueabi//sys-root --disable-nls --enable-threads=posix --enable-symvers=gnu --enable-c99 --enable-long-long --enable-target-optspace
Thread model: posix
gcc version 4.4.3 (ctng-1.6.1)
root@lppwrrr:~#

```

2.3 安装触摸屏校验工具

把tslib-1.4.tar.gz放到一个目录下，比如/opt目录下。接着执行命令如下：

```
#tar zxvf tslib-1.4.tar.gz
```

执行命令cd tslib，进入tslib当前目录,接着执行命令如下：

```
#./autogen.sh
```

然后接着执行配置命令：

```
#./configure --prefix=/usr/local/tslib --host=arm-linux-ac_cv_func_malloc_0_nonnull=yes
```

最后执行编译命令和安装命令如下：

1. make
2. make install

如果以上步骤没错，则可以在/usr/local/tslib下出现我们需要的库文件，这些库文件我们移植时会用到。

编译期间出现的错误：./autogen.sh: 4: autoreconf: not found

解决办法：sudo apt-get install autoconf automake libtool

2.4 安装QtCreator

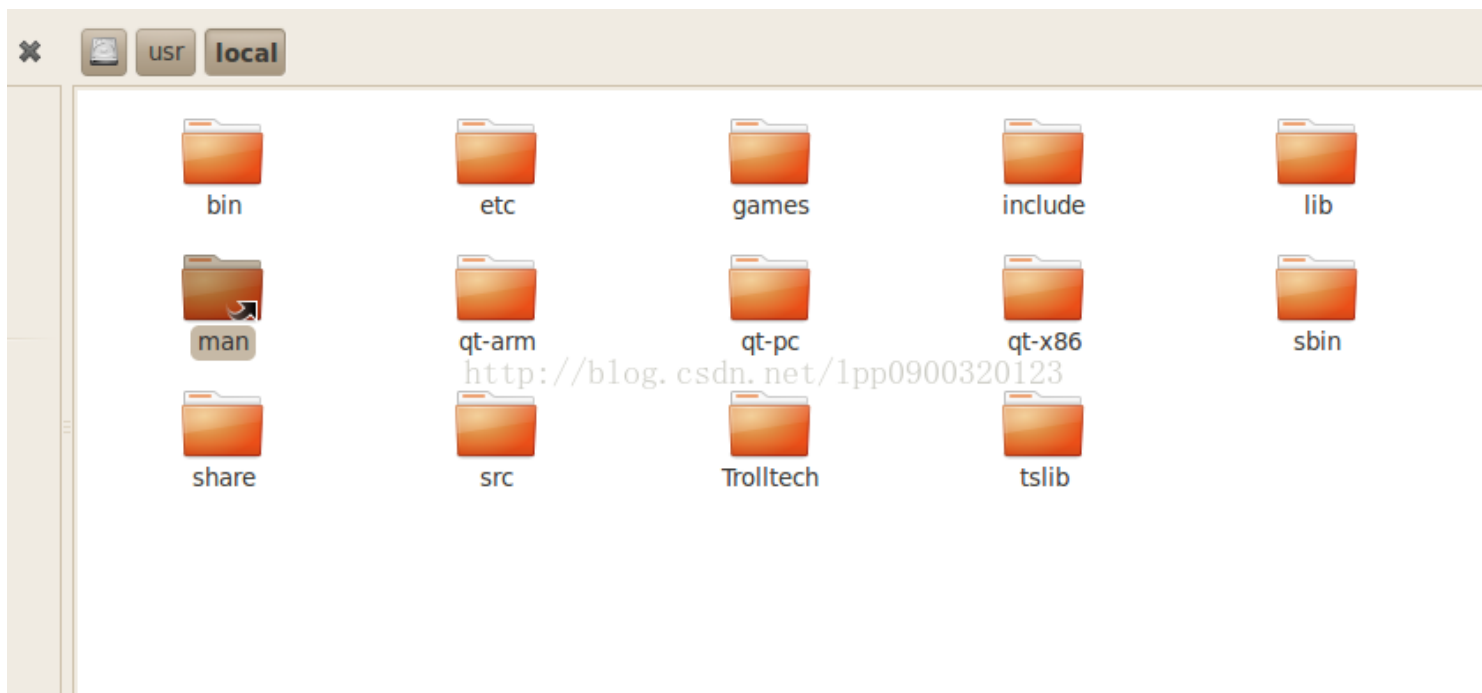
qt-creator-linux-x86_64-opensource-2.5.2.bin 拷贝到linux系统中，在终端中把路径切换到qt-creator-linux-x86_64-opensource-2.5.2.bin所在的路径，运行命令：./qt-creator-linux-x86_64-opensource-2.5.2.bin 就会出现图形安装界面；下面的安装步骤就和我们在windows下安装软件差不多。

注意：下的版本要和你的操作系统相匹配，32位的系统下对应32位的应用程序；

2.5 编译Qt库源码

2.5.1 解压

将源码qt-everywhere-opensource-src-4.8.5.tar.gz 进行解压，并复制三份到/usr/local目录下，并分别命名为qt-pc，qt-x86，qt-arm。



2.5.2 编译PC版

进入qt-pc目录，执行如下操作：

```
#./configure  
# make  
# make install
```

编译过程中出现的问题：

出错：Basic XLibfunctionality test failed!

这个错误主要是因为，XLib库没有安装，所以在编译X11平台时，应该保证，以下命令正确执行：sudo apt-get install libx11-dev libxext-dev libxtst-dev

注意：可以使用 make confclean 命令 把以前的配置信息清除掉，然后再重新配置。

2.5.3 编译x86版

进入qt-x86目录，执行如下操作：

```
# ./configure -embedded x86 -qt-gfx-qvfb -qt-kbd-qvfb -qt-mouse-qvfb  
# make  
# make install
```

安装过程比较长，没有碰到过错误。

编译安装PC版中的 qvfb：

进入qt-pc/tools/qvfb/目录，执行如下操作：

```
#make
```


编译完毕，将qt-pc/bin目录下的qvfb文件复制到/usr/local/Trolltech/QtEmbedded-4.8.5/bin目录。

编译qvfb时，出现了如下错误：

x11keyfaker.cpp:48:34: 致命错误：X11/extensions/XTest.h：没有那个文件或目录
编译中断。make: *** [.obj/release-shared/x11keyfaker.o] 错误 1

解决方法：

sudo apt-get install xorg-dev

如果makefile中没有-IXtst选项，则修改makefile 中 LIBS 选项：添加 -IXtst选项

2.5.4 编译ARM版

必须加上“-prefix /usr/local/Trolltech/QtEmbedded-4.8.5-arm ”参数，不然安装后不在QtEmbedded-4.8.5-arm文件夹下，而是覆盖了QtEmbedded-4.8.5。

```
# ./configure \  
-prefix /usr/local/Trolltech/QtEmbedded-4.6.3-arm \  
-opensource \  
-confirm-license \  
-release -shared \  
-embedded arm \  
-xplatform qws/linux-arm-g++ \  
-depths 16,18,24 \  
-fast \  
-optimized-qmake \  
-pch \  
-qt-sql-sqlite \  
-qt-libjpeg \  
-qt-zlib \  
-qt-libpng \  
-qt-freetype \  
-little-endian -host-little-endian \  
-no-qt3support \  
-no-libtiff -no-libmng \  
-no-opengl \  
-no-mmx -no-sse -no-sse2 \  
-no-3dnow \  
-no-openssl \  
-no-webkit
```

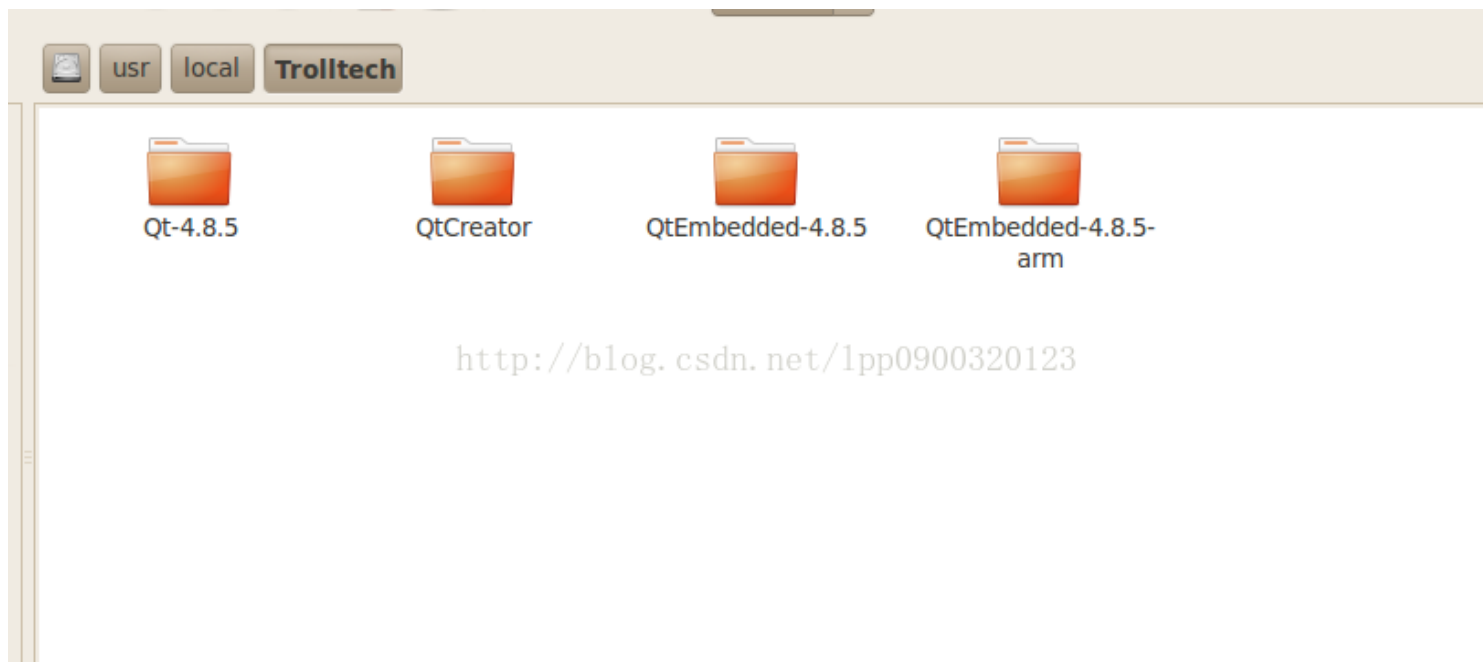
```
-no-qvfb \  
-no-phonon \  
-no-nis \  
-no-opengl \  
-no-cups \  
-no-glib \  
-no-xcursor -no-xfixes -no-xrandr -no-xrender \  
-no-separate-debug-info \  
-nomake examples -nomake tools -nomake docs  
-qt-mouse-tslib -I/usr/local/tslib/include -L/usr/local/tslib/lib
```

如果放弃配置，则使用命令：`# make confclean`

编译：`# make`

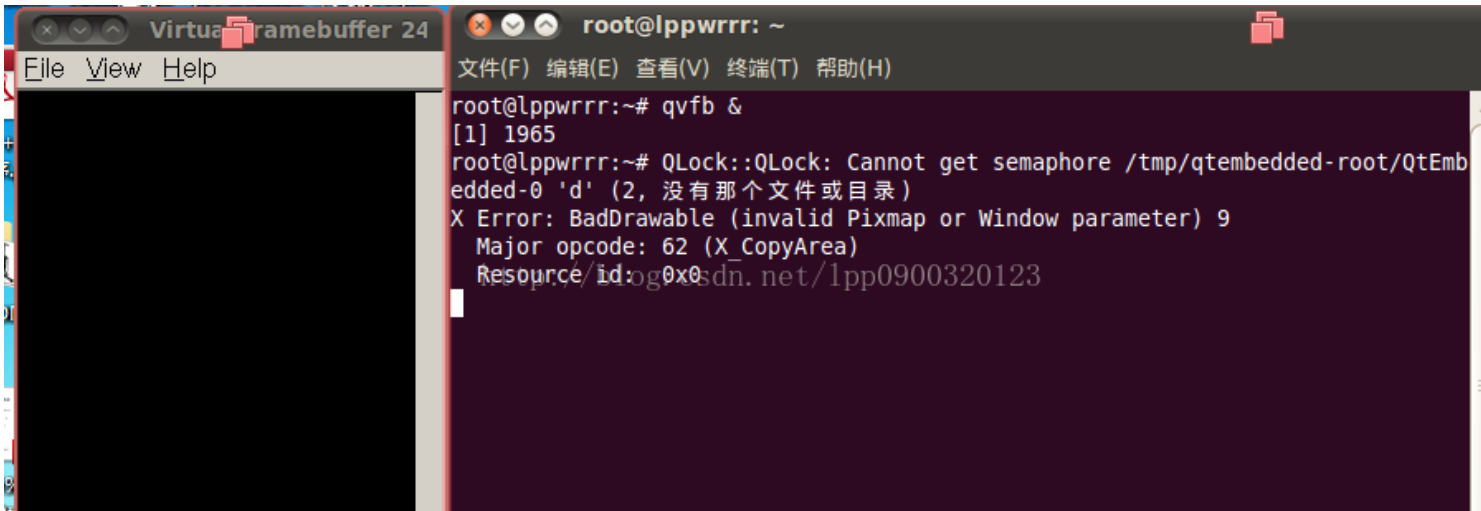
安装：`# make install`

安装完成后，在/usr/local/Trolltech目录中有三个文件夹：Qt-4.8.5、QtEmbedded-4.8.5、QtEmbedded-4.8.5-arm。



2.6 安装完后

2.6.1 qvfb的测试



运行qvfb, 提示了如下错误, 在网上也查了很多种解决办法, 最后还是没能解决; 有的说要重新编译内核, 并带上参数; 觉得比较麻烦, 就没有去尝试; 不过这个错误并不影响我们仿真模拟Qt4程序。

2.6.2 QtCreator编译前的配置

X86的调试编译与调试

(1) 打开QtCreator, 工具---->选项--->构建和运行 进行设置, 在Qt版本菜单下, 手动设置选择qmake路径。



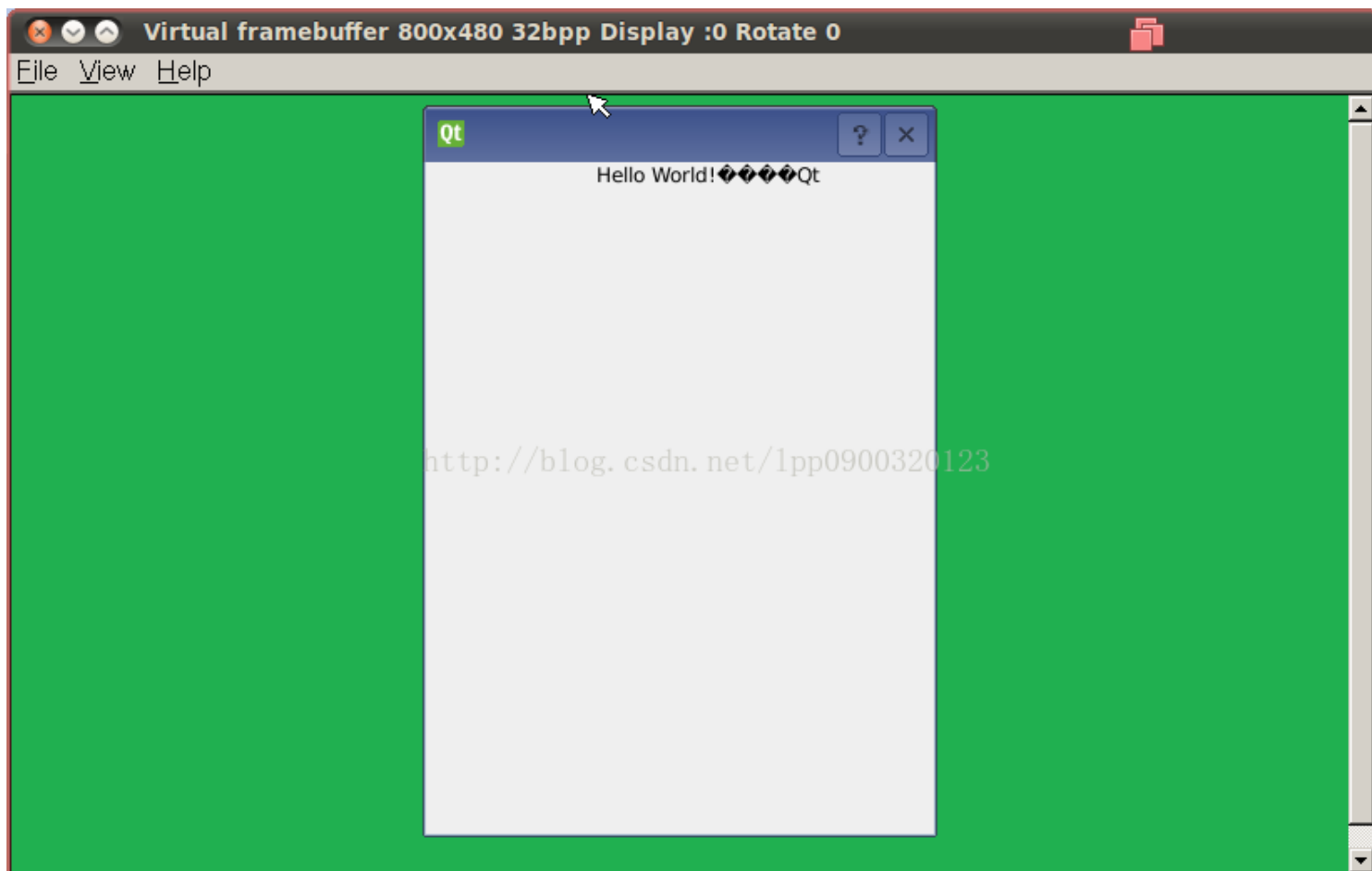
(2) 打开QtCreator 对其进行设置, 点击左边的项目, 选择运行, 对其进行设置, 在参数里添加 -qws参数



(3) 编译与运行

现在终端 输入命令：`qvfb -width 800 -height 480 &` 打开虚拟缓冲端；
然后就可以在QtCreator对程序进行编译、运行了；

运行结果如下：



由于自己没有ARM版，现在移植的那部分还没有做，等以后自己做的的话，自己会贴上去的。

总结

由于嵌入式实验的原因，自己想用Qt4编写一个俄罗斯方块的游戏，并把它移到ARM版上；就开始了Qt的交叉编译环境的搭建；自己足足花了4天半的时间；之前在CentOS上进行搭建的，但是出现了很多错误，并且这些错误的解决办法网上给的都是Ubuntu的解决办法，在第三天时，用CentOS实在搭不下去了，于是就换成了Ubuntu10.04，接下来虽然也遇到错误，但在网上也都能搜到相似的错误，唯有编译qvfb时，遇到的麻烦最多。可能我们会做一件我们觉得很难得事情或者之前我们从未涉及的领域，这时我们不要做无头的苍蝇，到处乱撞，应该有目的去撞；心里一定要知道完成这件事的大概流程，和需要我们拥有哪些知识，然后一步一步地去做。