

南京航空航天大学

# 嵌入式系统综合实验报告

**Qtopia** 在 **S3C2440** 开发板上的移植

040630520

彭立勋

2009.05

## 一、实验目的

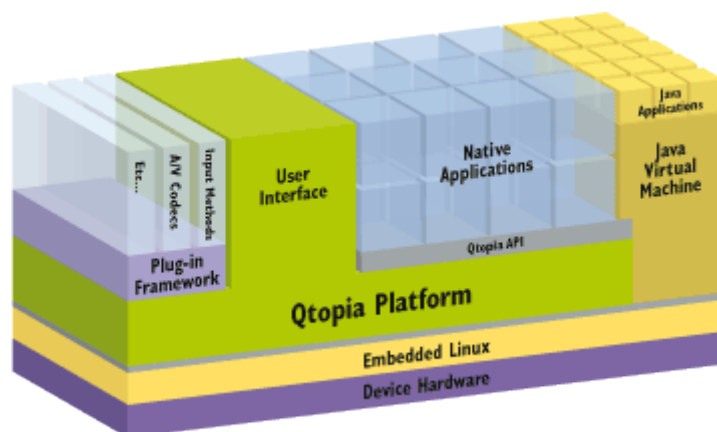
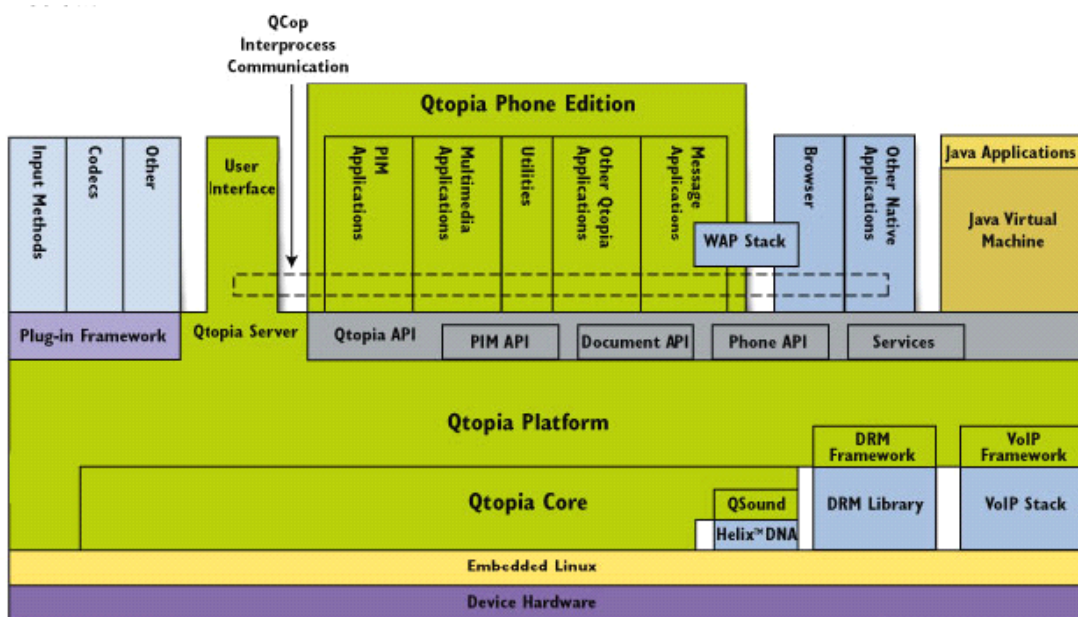
1. 熟悉ARM体系结构；
2. 熟悉Qtopia图形环境。

## 二、实验内容

将Qtopia图形环境移植到FriendlyARM QQ2440开发板。

## 三、预备知识

Qtopia的体系结构



## 四、实验设备及工具

硬件：QQ2440开发板、用于ARM7TDMI的JTAG仿真器、PC机Pentum100以上

---

软件: PC 机操作系统 win98、Win2000 或 WinXP、ARM ADS1.2 集成开发环境、仿真器驱动程序、实验原理及说明

## 五、实验步骤

### 1. 设置开发环境

下载 Qt/Embedded 的源代码包: arm-qtopia.tgz (用于在 ARM 上运行的 Qtopia 开发包, 支持 USB 鼠标。)

解压源代码:

```
#tar xvzf arm-qtopia.tgz -C /opt/FriendlyARM/qq2440
```

将开发板自带的 arm-linux-gcc-3.3.2.tgz 编译器解压到/usr/local/arm/3.3.2, 并且设置环境变量。

### 2. 编译 Qt/Embedded

因为配置编译 Qt/Embedded 的过程比较复杂, 为了方便, 我把配置和编译的步骤制作成一个 build-all 脚本, 内容如下:

```
#!/bin/bash
```

```
tar xfvz tmake-1.11.tar.gz
```

```
tar xfvz qt-embedded-2.3.7.tar.gz
```

```
tar xfvz qtopia-free-1.7.0.tar.gz
```

```
tar xfvz qt-x11-2.3.2.tar.gz
```

```
tar xvzf konqueror-embedded.tar.gz
```

```
mv konqueror-embedded konq
```

```
mv tmake-1.11 tmake
```

```
mv qt-2.3.7/ qt
```

```
mv qtopia-free-1.7.0 qtopia
```

```
mv qt-2.3.2 qt-x11
```

```
cd qt-x11
```

```
export QTDIR=$PWD
```

```
echo yes | ./configure -static -no-xft -no-opengl -no-sm
```

```
make -C src/moc
```

```
cp src/moc/moc bin
```

```
make -C src
```

```
make -C tools/designer
```

```
make -C tools/qvfb
```

```
cp tools/qvfb/qvfb bin
```

```
strip bin/uic bin/moc bin/designer bin/qvfb
```

```
cd ..
```

```
cp qt-x11/bin/* qt/bin
```

```
rm -fr qt-x11
```

```
export QTDIR=$PWD/qt
```

```
export QPEDIR=$PWD/qtopia
```

```
export TMAKEDIR=$PWD/tmake
```

```
export TMAKEPATH=$TMAKEDIR/lib/qws/linux-arm-g++
```

```
export PATH=$QTDIR/bin:$QPEDIR/bin:$TMAKEDIR/bin:$PATH
cd qt
make clean
cp ../qtopia/src/qt/qconfig-qpe.h src/tools/
(echo yes ; echo no) | ./configure -system-jpeg -gif -system-libpng -system-zlib -platform linux-
arm-g++ -qconfig qpe -depths 16,24,32
make -C src
cd ..
cd qtopia/src
./configure -platform linux-arm-g++
make
cd -
```

```
export AR=arm-linux-ar
export STRIP=arm-linux-strip
export RANLIB=arm-linux-ranlib
export CXX=arm-linux-g++
export CCC=arm-linux-c++
export CC=arm-linux-gcc
export CROSS_COMPILE=1
export PATH=$PATH:/usr/local/arm/3.3.2/bin
export INSTALL=/usr/local/arm/3.3.2
cd konq
./configure --host=arm-linux --target=arm-linux --enable-embedded --enable-qt-embedded --
enable-qpe --with-gui=qpe --disable-debug --enable-ftp --enable-static --disable-shared --disable-
mt --with-extra-libs=/usr/local/arm/3.3.2/lib:$QPEDIR/lib --with-extra-
include=/usr/local/arm/3.3.2/include:$QPEDIR/include --without-ssl --with-qt-dir=$QTDIR --
with-qt-includes=$QTDIR/include --with-qt-libraries=$QTDIR/lib --with-qtopia-dir=$QPEDIR
make
cd -
```

```
#cd /opt/FriendlyARM/QQ2440/arm-qtopia
```

```
#!/build-all
```

```
#!/mktarget_qtopia (制作更新文件系统用的 Qtopia 压缩包，将生成 target_qtopia.tgz，脚本内  
容如下)
```

```
#!/bin/sh
```

```
export QTDIR=$PWD/qt
export QPEDIR=$PWD/qtopia
export TMAKEDIR=$PWD/tmake
export TMAKEPATH=$TMAKEDIR/lib/qws/linux-generic-g++
export PATH=$QTDIR/bin:$QPEDIR/bin:$TMAKEDIR/bin:$PATH
```

```
echo "Copy qt - lib & fonts..."
```

```

mkdir -p ./opt/qt/lib/fonts
cp -a $QTDIR/lib/lib* ./opt/qt/lib

echo "Copy qtopia - apps bin etc help i18n lib pics plugins services sounds"
mkdir -p ./opt/qtopia
export TARGET_QPE=./opt/qtopia
cp -a $QPEDIR/bin $TARGET_QPE
cp -a $QPEDIR/etc $TARGET_QPE
cp -a $QPEDIR/help $TARGET_QPE
cp -a $QPEDIR/i18n $TARGET_QPE
cp -a $QPEDIR/lib $TARGET_QPE
cp -a $QPEDIR/pics $TARGET_QPE
cp -a $QPEDIR/plugins $TARGET_QPE
cp -a $QPEDIR/services $TARGET_QPE
cp -a $QPEDIR/sounds $TARGET_QPE

echo "tar zcvf target_qtopia ./opt"
tar zcvf target_qtopia.tgz ./opt
rm -rf ./opt

```

`#!/mktarget_konq` (制作更新文件系统用的浏览器压缩包，将生成 `target_konq.tgz`，脚本内容如下)

```

#!/bin/sh

mkdir -p ./opt/qtopia/bin
mkdir -p ./opt/qtopia/apps/Applications
mkdir -p ./opt/qtopia/pics
mkdir -p ./opt/kde/share/apps/khtml/css
mkdir -p ./opt/kde/share/config

cp konq/konq-embed/src/konqueror ./opt/qtopia/bin
cp konq/konq-embed/src/konqueror.png ./opt/qtopia/pics
cp konq/konq-embed/src/konqueror.desktop ./opt/qtopia/apps/Applications
cp konq/konq-embed/kdesrc/khtml/css/html4.css ./opt/kde/share/apps/khtml/css
cp konq/konq-embed/kdesrc/kdecore/charsets ./opt/kde/share/config

tar zcvf target_konq.tgz ./opt
rm -rf ./opt

```

### 3.编译 Qt 的 Hello World

```

#cd /opt/FriendlyARM/qq2440/arm-qtopia
#. set-env
#cd hello
hello.cpp

```

```

#include "hello.h"
#include <qlabel.h>

HelloForm::HelloForm( QWidget* parent, const char* name, WFlags fl):
HelloBaseForm(parent, name, fl)
{
}
HelloForm::~HelloForm()
{
}

void HelloForm::SayHello()
{
    MessageLabel->setText("Hello, Qtopia world!");
}

hello.h
#ifndef HELLOFORM_H
#define HELLOFORM_H
#include "hello_base.h"

class HelloForm : public HelloBaseForm
{
public:
    HelloForm( QWidget* parent = 0, const char* name = 0, WFlags fl = 0 );
    virtual ~HelloForm();
protected:
    virtual void SayHello();
};

#endif // HELLOFORM_H

main.cpp
#include "hello.h"
#include <qtopia/qpeapplication.h>

int main( int argc, char *argv[] )
{
    QPEApplication a( argc, argv );

    HelloForm f;
    a.showMainWidget( &f );

    return a.exec();
}

```

```
#make
```

在/opt/FriendlyARM/qq2440/arm-qtopia/qtopia/bin 目录下生成 hello 可执行文件

#### 4.把 hello 下载到目标板并运行

(1)复制文件到 Windows 目录下

把刚刚编译生成的 hello 可执行文件复制到 Windows 下,同时把 hello/目录中的 hello.desktop 也复制到 Windows 下的某个目录。

(2)使用 rz 下载文件到开发板

在开发板串口终端输入“rz”命令开始接收从串口发来的文件。

然后点鼠标右键,在弹出的菜单中选择“发送文件”,选择 hello,开始向开发板传送文件。



使用同样的方法下载 hello.desktop 到开发板里面

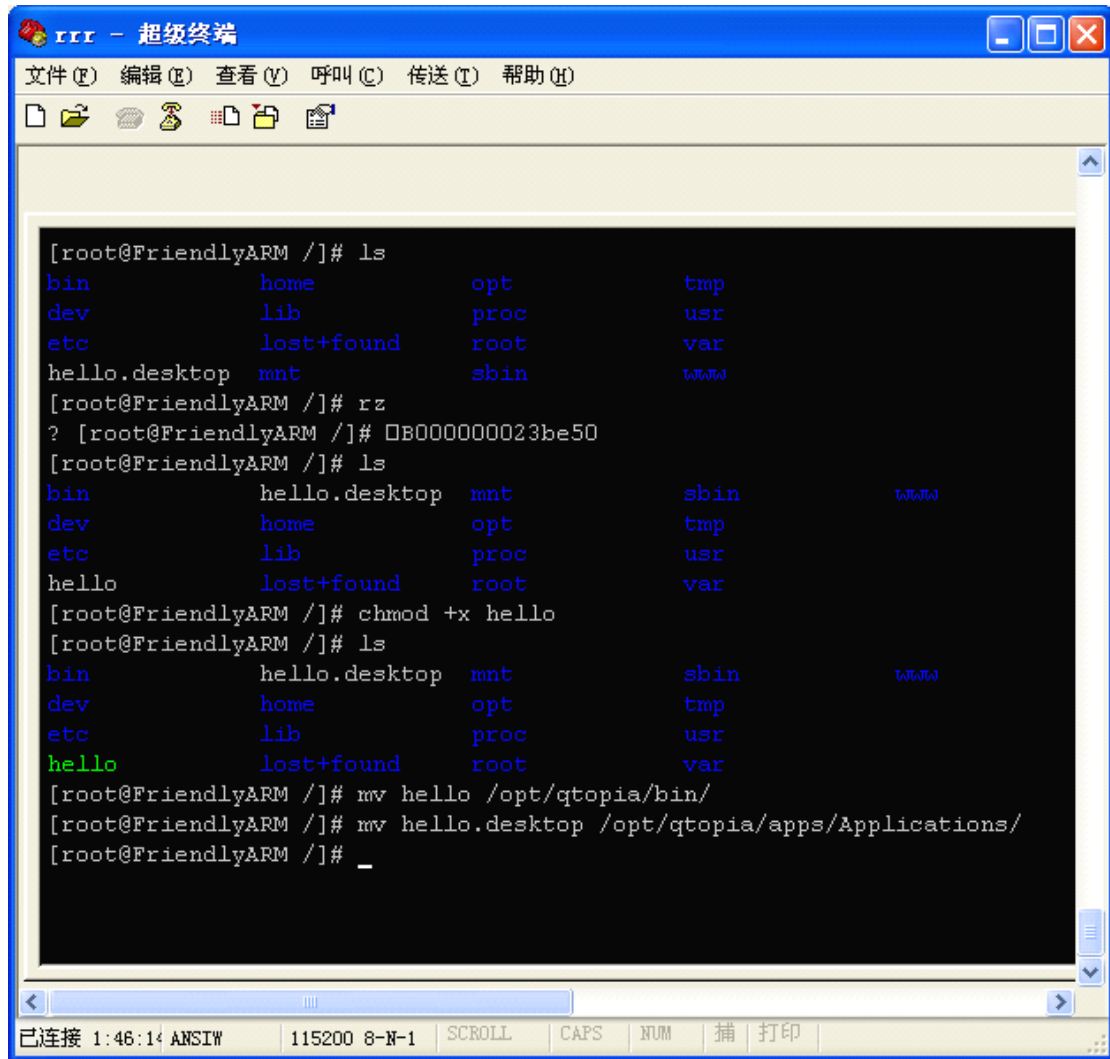
(3)改变 hello 文件的执行权限

通过串口现在到开发板的文件是没有执行权限的,所以需要先使用 chmod 命令改变它的执行权限,再把它放到正确的目录里面。

```
#chmod +x hello
```

```
#mv hello /opt/qtopia/bin
```

```
#mv hello.desktop /opt/qtopia/apps/Applications
```

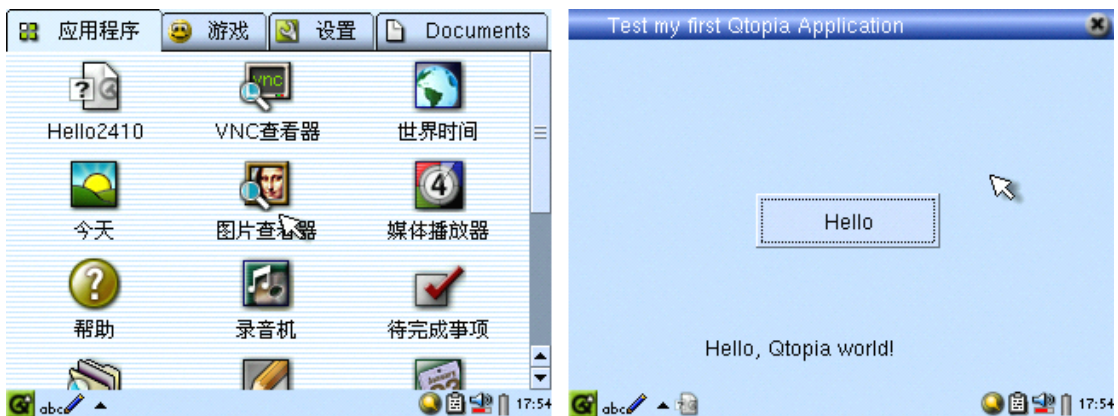


```
[root@FriendlyARM /]# ls
bin             home            opt             tmp
dev             lib             proc           usr
etc             lost+found     root           var
hello.desktop  mnt            sbin           777777
[root@FriendlyARM /]# rz
? [root@FriendlyARM /]# 0B000000023be50
[root@FriendlyARM /]# ls
bin             hello.desktop  mnt            sbin           777777
dev             home           opt            tmp
etc             lib            proc           usr
hello           lost+found    root           var
[root@FriendlyARM /]# chmod +x hello
[root@FriendlyARM /]# ls
bin             hello.desktop  mnt            sbin           777777
dev             home           opt            tmp
etc             lib            proc           usr
hello           lost+found    root           var
[root@FriendlyARM /]# mv hello /opt/qtopia/bin/
[root@FriendlyARM /]# mv hello.desktop /opt/qtopia/apps/Applications/
[root@FriendlyARM /]# _
```

已连接 1:46:14 ANSIIW 115200 8-N-1 SCROLL CAPS NUM 捕 打印

(4)在开发板上运行 hello  
重新启动开发板，就可以看到 hello 图标，鼠标点击运行它就可以看到界面。

## 六、实验结果



Qtopia成功被刷到开发板，并且可以运行Hello World测试程序。



## 八、实验心得

更熟悉了ARM硬件体系结构，也熟悉了Qtopia环境的构架，对今后从事ARM+Linux方面的开发积累了很好的经验。