

## 基于 AT91SAM9261/ AT91SAM9263 的 QT 移植

### 一、准备工作

开发环境：RedHat Linux 9.0+ARM-LINUX

QT 软件包：tmake-1.13.tar.gz qt-embedded-2.3.10.tar.gz qt-x11-2.3.2.tar.gz

### 二、解压安装包并设置环境变量

接下来的工作都在统一目录下执行，如/home/9261qt

#### 1、安装 Tmake

```
tar -xzf tmake-1.13.tar.gz
export TMAKEDIR=$PWD/tmake-1.13
```

#### 2、安装 Qt 2.3.2

```
tar -xzf qt-x11-2.3.2.tar.gz
export QT2DIR=$PWD/qt-2.3.2
```

#### 3、安装 Qt/Embedded

```
tar -xzf qt-embedded-2.3.10-free.tar.gz
export QTEDIR=$PWD/qt-2.3.10
```

### 三、编译 QT

#### 1. 编译 Qt2.3.2

```
cd $QT2DIR
export TMAKEPATH=$TMAKEDIR/lib/linux-g++
export QTDIR=$QT2DIR
export PATH=$QTDIR/bin:$PATH
export LD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH
./configure -no-xft
make
mkdir $QTEDIR/bin
cp bin/uic $QTEDIR/bin/
```

注：./configure 是对 Qt 进行配置，编译完成后需要将生成的/bin/uic 复制到 \$QTEDIR 下新创建的目录 bin 中，因为在随后编译 Qt/Embedded 的时候会用到这个工具。

#### 2. 编译 Qvfb

```
export TMAKEPATH=$TMAKEDIR/lib/linux-g++
export QTDIR=$QT2DIR
export PATH=$QTDIR/bin:$PATH
export LD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH
cd $QTEDIR/tools/qvfb
tmake -o Makefile qvfb.pro
make
mv qvfb $QTEDIR/bin/
```

注：qvfb 工具用来生成 Virtual framebuffer，这是一个非常有用的工具，它可以模拟在开发板上的显示情况，如果在 Virtual framebuffer 中运行没有问题的话，可以直接通过交叉编译在开发板上运行。

#### 3. 编译 Qt/Embedded

```
cd $QTEDIR
```

```
export TMAKEPATH=$TMAKEDIR/lib/qws/linux-x86-g++
export QTDIR=$QTEDIR
export PATH=$QTDIR/bin:$PATH
export LD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH
./configure -no-xft -qvfb -depths 4,8,16,32
make
```

#### 四、PC 上运行测试

- 1、在主机的 framebuffer 上运行

```
cd $QTEDIR/examples/launcher
./launcher -qws
```

- 2、在 Virtual framebuffer 上运行

```
export QTDIR=$QTEDIR
export PATH=$QTEDIR/bin:$PATH
export LD_LIBRARY_PATH=$QTEDIR/lib:$QT2DIR/lib:$LD_LIBRARY_PATH
cd $QTEDIR/examples/launcher
qvfb -width 640 -height 480 &
sleep 10
./launcher -qws
```

#### 五、QT 移植到 ARM 开发板上

```
cd $QTEDIR
export TMAKEPATH=$TMAKEDIR/lib/qws/linux-arm-g++
export QTDIR=$QTEDIR
export PATH=$QTDIR/bin:$PATH
export LD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH
./configure -xplatform linux-arm-g++ -no-xft -no-qvfb -depths 4,8,16,32
make
```

这步完成后，我们会在/\$QTEDIR/lib/目录下面看到 libqte.so, libqte.so.2, libqte.so.2.3 , libqte.so.2.3.10 这四个文件，我们可以使用 file 命令来查看这个库文件是否是我们在开发板上跑的库。

```
file libqte.so.2.3.10
libqte.so.2.3.10: ELF 32-bit LSB shared object, ARM, version 1 (ARM), not stripped
```

有了这个库以后我们就可以把它拷贝到我们的开发板中相应的库目录下面，这里我们选择了开发板上的/usr/lib 目录，将/\$QTEDIR/lib/下的 libqte.so\*复制到/usr/lib 目录下。

#### 六、ARM 运行测试程序