

## 为 SAM926x 编译 Qtopia-2.2.0

文档编号	MAN3016A_CH				
文档版本	Rev. A				
文档摘要	详细描述了为 SAM926x 编译 qtopia-2.2.0 的步骤				
关键词	SAM926x GUI Linux Qtopia				
创建日期	2010-07-29	创建人员	Dracula	审核人员	<a href="#">Hotislandn</a>
文档类型	公开发布/开发板配套文件				
版权信息	<a href="#">Mcuzone</a> 原创文档，转载请注明出处				

### 更新历史

版本	时间	更新	作者
Rev. A	2010-07-29	初始创建	Dracula

www.mcuzone.com

微控电子 乐微电子  
杭州市登云路 639 号 2B143  
销售 TEL: +86-571-89908193  
支持 TEL: 18913989166  
FAX: +86-571-89908193  
[www.mcuzone.com](http://www.mcuzone.com) [www.atarm.com](http://www.atarm.com)

# 1. 概述

[Qtopia](#) 是 Trolltech 公司为采用嵌入式 Linux 操作系统的消费电子设备而开发的综合应用平台, Qtopia 包含完整的应用层、灵活的用户界面、窗口操作系统、应用程序启动程序以及开发框架。

Qtopia 作为一个开源的 UI 系统,在嵌入式 Linux 下得到了广泛的应用,也得到了很多应用开发人员的支持.因此,在系统设计中选择 qtopia 作为 UI,可以方便地整合很多用 qt 开发的应用。

[MDK9263-G](#) 是 [mcuzone](#) 开发的一款基于 [ATMEL AT91SAM9263](#) (以下简称 SAM9263)的 ARM9 开发板。

SAM9263 内置 LCD 控制器,可以方便地连接 TFT LCD 模块. 以下例子中将连接一块 4.3”的 LCD 模块,分辨率为 480 x 272, 该 LCD 模块包含触摸屏。

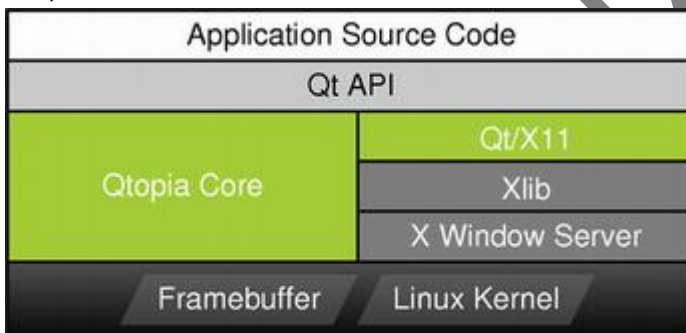
本文的目标是编译 qtopia-2.2.0 的 PDA 版本,运行于 SAM9263-G,并使用触摸屏操作。

开发机选用基于 [virtualbox](#) 的 [ubuntu\\_10.04](#), 目标板的 Linux kernel 版本为 [2.6.30](#). 工具链使用 [arm-none-linux-gnueabi](#). Linux 开发机需要有 internet 连接。

## 2. 编译 Qtopia

### 2.1 Qtopia-2.2.0

Qtopia 的大致架构如下:



在 SAM926x 上运行 Qtopia, 可以通过 Linux 下的 frame buffer, 而不需要依赖庞大的 X.

SAM9263 的 frame buffer 驱动在 Linux 源码中已经存在,直接编译进来即可。

触摸屏可以通过 tslib 来支持。

### 2.2 编译

Qtopia 有一些的依赖,可以根据实际需要按顺序编译。

编译之前,需要设置环境。

创建一个工作目录,用于编译:

```
cust@u1004:/samlinux$ mkdir qtopia
cust@u1004:/samlinux$ cd qtopia/
cust@u1004:/samlinux/qtopia$ pwd
/samlinux/qtopia
cust@u1004:/samlinux/qtopia$
```

如此,工作目录即是/samlinux/qtopia.

在工作目录下创建一个文件夹,用于安装编译后生成的各种文件:

```
mkdir _install
```

设置一些环境变量:

```
export PREFIX=$PWD/_install
export BUILD_DIR=$PWD
export HOST=arm-none-linux-gnueabi
```

设置编译相关的工具变量(注意,在此之前 arm-none-linux-gnueabi 工具链必须已经安装并在用户 PATH 中):

```
export LD=$HOST-ld
export CC=$HOST-gcc
export CXX=$HOST-g++
export CPP="$HOST-gcc -E"
export OBJDUMP=$HOST-objdump
export AS=$HOST-as
export AR=$HOST-ar
export RANLIB=$HOST-ranlib
export STRIP=$HOST-strip
export NM=$HOST-nm
export CFLAGS="-mtune=arm926ej-s -mcpu=arm926ej-s -I$PREFIX/include -I$PREFIX/usr/include"
export CPPFLAGS="-march=armv5te -mcpu=arm926ej-s -I$PREFIX/include -I$PREFIX/usr/include"
export LDFLAGS="-L$PREFIX/lib -Wl,-rpath,$PREFIX/lib -L$PREFIX/usr/lib"
export PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig
```

检查下环境变量:

```
cust@u1004:/samlinux/qtopia$ echo $PREFIX
/samlinux/qtopia/_install
cust@u1004:/samlinux/qtopia$ echo $CC
arm-none-linux-gnueabi-gcc
cust@u1004:/samlinux/qtopia$
```

## 2.2.1 tslib

[tslib](#) 是一个开源的程序,能够为触摸屏驱动获得的采样提供诸如滤波、去抖、校准等功能,通常作为触摸屏驱动的适配层,为上层的应用提供了一个统一的接口。

下载源码并展开源码包:

```
cd $BUILD_DIR
wget http://download.berlios.de/tslib/tslib-1.0.tar.bz2
tar jxvf tslib-1.0.tar.bz2
cd tslib-1.0
生成配置文件,并进行 configure:
./autogen.sh
echo ac_cv_func_malloc_0_nonnull=yes > arm-linux.cache
./configure --host=$HOST --prefix=$PREFIX --cache-file=arm-linux.cache --enable-inputapi=no
```

编译并安装:

```
make install-strip
```

```
cd ..
```

生成在安装目录下的对应库文件:

```
cust@u1004:/samlinux/qtopia$ ls _install/lib/
libts-0.0.so.0 libts-0.0.so.0.1.1 libts.1a libts.so pkgconfig ts
```

## 2.2.2 zlib

```
cd $BUILD_DIR
```

```
wget http://zlib.net/zlib-1.2.5.tar.gz
```

```
tar zxvf zlib-1.2.5.tar.gz
```

```
cd zlib-1.2.5
```

```
./configure --prefix=$PREFIX --shared
```

```
make install
```

```
cd ..
```

## 2.2.3 jpeg

下载源码,展开并配置:

```
cd $BUILD_DIR
```

```
wget http://files.chumby.com/source/ironforge/build733/jpeg-6b.tar.gz
```

```
tar zxvf jpeg-6b.tar.gz
```

```
cd jpeg-6b
```

```
export LD=""
```

```
export CC=""
```

```
export CXX=""
```

```
export CPP=""
```

```
export OBJDUMP=""
```

```
export AS=""
```

```
export AR=""
```

```
export RANLIB=""
```

```
export STRIP=""
```

```
export NM=""
```

```
export CFLAGS=""
```

```
export CPPFLAGS=""
```

```
export LDFLAGS=""
```

```
./configure --host=$HOST --build=$BUILD --prefix=$PREFIX --enable-shared --enable-static
```

为了正确编译,需要修改 Makefile,具体修改如下:

```
22 # The name of your C compiler:
23 CC= arm-none-linux-gnueabi-gcc
24
```

```

62 # library (.a) file creation command
63 AR= arm-none-linux-gnueabi-ar rc
64 # second step in .a creation (use "touch" if not needed)
65 AR2= arm-none-linux-gnueabi-ranlib
66 # installation program

```

修改完成后使用如下命令安装,并恢复环境变量.

```

make install
export LD=$HOST-ld
export CC=$HOST-gcc
export CXX=$HOST-g++
export CPP="$HOST-gcc -E"
export OBJDUMP=$HOST-objdump
export AS=$HOST-as
export AR=$HOST-ar
export RANLIB=$HOST-ranlib
export STRIP=$HOST-strip
export NM=$HOST-nm
export CFLAGS="-mtune=arm926ej-s -mcpu=arm926ej-s -I$PREFIX/include -I$PREFIX/usr/include"
export CPPFLAGS="-march=armv5te -mcpu=arm926ej-s -I$PREFIX/include -I$PREFIX/usr/include"
export LDFLAGS="-L$PREFIX/lib -Wl,-rpath,$PREFIX/lib -L$PREFIX/usr/lib"
export PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig
cd ..

```

## 2.2.4 libpng

```

cd $BUILD_DIR
wget http://image_magick.veidrodix.com/image_magick/delegates/libpng-1.2.43.tar.bz2
tar jxvf libpng-1.2.43.tar.bz2
cd libpng-1.2.43
./configure --host=$HOST --build=$BUILD --prefix=$PREFIX
make install
cd ..

```

安装后的效果:

```

cust@u1004:/samlinux/qtopia$ ls _install/lib/
libjpeg.a  libjpeg.so.62  libpng12.la  libpng12.so.0.43.0  libpng.so  1
libjpeg.la  libjpeg.so.62.0.0  libpng12.so  libpng.a  libpng.so.3  1
libjpeg.so  libpng12.a  libpng12.so.0  libpng.la  libpng.so.3.43.0  1
cust@u1004:/samlinux/qtopia$ file _install/lib/libpng12.so.0.43.0
_install/lib/libpng12.so.0.43.0: ELF 32-bit LSB shared object, ARM, version 1 (SYSV),
cust@u1004:/samlinux/qtopia$ file _install/lib/libjpeg.so.62.0.0
_install/lib/libjpeg.so.62.0.0: ELF 32-bit LSB shared object, ARM, version 1 (SYSV),
cust@u1004:/samlinux/qtopia$

```

## 2.2.5 e2fsprogs

```
cd $BUILD_DIR
wget ftp://ftp.eenet.ee/pub/FreeBSD/distfiles/e2fsprogs-1.39.tar.gz
tar zxvf e2fsprogs-1.39.tar.gz
cd e2fsprogs-1.39
make distclean
./configure --host=$HOST --enable-elf-shlibs --with-cc=$CC --with-linker=$LD --prefix=$PREFIX
make install
```

安装时可能出现如下错误:

```
mkdir /samlinux/qtopia/_install/man/man5
INSTALL /samlinux/qtopia/_install/sbin/e2fsck
LINK /samlinux/qtopia/_install/sbin/fsck.ext2
LINK /samlinux/qtopia/_install/sbin/fsck.ext3
INSTALL_DATA /samlinux/qtopia/_install/man/man8/e2fsck.8
INSTALL_DATA /samlinux/qtopia/_install/man/man5/e2fsck.conf.5
/usr/bin/install: cannot stat `e2fsck.conf.5': No such file or directory
make[1]: *** [install] Error 1
make[1]: Leaving directory `/samlinux/qtopia/e2fsprogs-1.39/e2fsck'
make: *** [install-progs-recursive] Error 1
```

运行如下命令,继续安装:

```
touch ./e2fsck/e2fsck.conf.5
make install
```

又有另一个错误:

```
make[1]: Entering directory `/samlinux/qtopia/e2fsprogs-1.39/doc'
MKINSTALLDIRS /samlinux/qtopia/_install/info
INSTALL_DATA /samlinux/qtopia/_install/info/libext2fs.info*
/usr/bin/install: cannot stat `libext2fs.info*': No such file or directory
make[1]: *** [install-doc-libs] Error 1
make[1]: Leaving directory `/samlinux/qtopia/e2fsprogs-1.39/doc'
make: [install-doc-libs] Error 2 (ignored)
if test ! -d e2fsck && test ! -d debugfs && test ! -d misc && test ! -d ext2ed ;
cust@u1004:/samlinux/qtopia/e2fsprogs-1.39$
```

但是此时编译 qtopia 所需要的 libuuid 已经生成:

```
cust@u1004:/samlinux/qtopia/e2fsprogs-1.39$ ls ../_install/lib/libuuid*
../_install/lib/libuuid.so  ../_install/lib/libuuid.so.1  ../_install/lib/libuuid.so.1.2
cust@u1004:/samlinux/qtopia/e2fsprogs-1.39$ file ../_install/lib/libuuid.so.1.2
../_install/lib/libuuid.so.1.2: ELF 32-bit LSB shared object, ARM, version 1 (SYSV), dynamic
cust@u1004:/samlinux/qtopia/e2fsprogs-1.39$
```

手动复制 uuid 的头文件:

```
cp ./lib/uuid/uuid.h $PREFIX/include
cd ..
```

## 2.2.6 qtopia

下载 qtopia 源码并展开:

```
cd $BUILD_DIR
wget http://ftp.roedu.net/mirrors/ftp.trolltech.com/qtopia/source/qtopia-free-src-2.2.0.tar.gz
tar zxvf qtopia-free-src-2.2.0.tar.gz
```

方便起见,将为 ARM 编译的源码包改名:

```
mv qtopia-free-2.2.0 qtopia-free-2.2.0-arm
```

从产品光盘中将 patch 文件 qtopia220\_patch\_mcuzone20100729.zip 放到当前的工作目录下,并展开:

```

cust@u1004:/samlinux/qtopia$ ls *.zip
qtopia220_patch_mcuzone20100729.zip
cust@u1004:/samlinux/qtopia$ unzip qtopia
unzip: cannot find or open qtopia, qtopia.zip or qtopia.ZIP.
cust@u1004:/samlinux/qtopia$ unzip qtopia220_patch_mcuzone20100729.zip
Archive:  qtopia220_patch_mcuzone20100729.zip
  creating:  patch/
  inflating: patch/ablabe1.h
  inflating: patch/abtable_p.h
  inflating: patch/buttoneditorialog.h
  inflating: patch/inputmethods.cpp
  inflating: patch/keyboard.h
  inflating: patch/minefield.h
  inflating: patch/numberentry_p.h
  inflating: patch/packagewizard.h
  inflating: patch/patch_file.sh
  inflating: patch/polished.h
  inflating: patch/qconfig-qpe.h
  inflating: patch/qdawg.cpp
  inflating: patch/qmake.conf
  inflating: patch/qmemoryfile_unix.cpp.1
  inflating: patch/qmemoryfile_unix.cpp.2
  inflating: patch/qpixmapcache.cpp
  inflating: patch/qvaluestack.h
  inflating: patch/qvfbview.cpp
  inflating: patch/qwindowsystem_qws.h
  inflating: patch/thumbnailview_p.h
  inflating: patch/tmake.conf
  inflating: patch/videoviewer.cpp
  inflating: patch/vobject.cpp
  inflating: patch/wavplugin.cpp
cust@u1004:/samlinux/qtopia$ ls
e2fsprogs-1.39      jpeg-6b      libpng-1.2.43.tar.bz2
e2fsprogs-1.39.tar.gz  jpeg-6b.tar.gz  patch
_install          libpng-1.2.43  qtopia220_patch_mcuzone20100729.zip

```

将获得 patch 文件夹, 里面包含了完整的修改过后的文件,直接覆盖即可,不需要使用 patch.

修改 patch 脚本,添加可执行属性:

```
chmod 555 ./patch/patch_file.sh
```

进入源码目录:

```
cd qtopia-free-2.2.0-arm
```

先 patch 文件:

```
././patch/patch_file.sh
```

然后设置编译需要用到环境变量:

```
export TMAKEDIR=$BUILD_DIR/qtopia-free-2.2.0-arm/tmake/bin/
```

```
export QPEDIR=$BUILD_DIR/qtopia-free-2.2.0-arm/qtopia
```

```
export QTDIR=$BUILD_DIR/qtopia-free-2.2.0-arm/qt2
```

```
cp $QPEDIR/src/qt/qconfig-qpe.h $QTDIR/src/tools
```

```
cd $QPEDIR/src/libraries/qtopia
```

```
cp custom-linux-ipaq-g++.cpp custom-linux-arm-g++.cpp
```

```
cp custom-linux-ipaq-g++.h custom-linux-arm-g++.h
```



```
cp $BUILD_DIR/qtopia-free-2.2.0-arm/qtopia/etc/defaultbuttons-generic.conf
$BUILD_DIR/qtopia-free-2.2.0-arm/qtopia/etc/defaultbuttons-arm.conf
```

配置:

```
cd $BUILD_DIR/qtopia-free-2.2.0-arm
export QT2_CFG="-no-opengl -no-xft -no-sm -platform 'linux-g++' -I/usr/include
-I$BUILD_DIR/qtopia-free-2.2.0-arm/qt2/src/3rdparty/freetype/include/"
export DQT_CFG="-thread -no-xft -I/usr/work/qte/freetype-2.1.10/_install/include/freetype2
-L/usr/work/qte/freetype-2.1.10/_install/lib -lfreetype -lz -lXft"
export QTE_CFG="-embedded -no-xft -qconfig qpe -depths 16,24 -no-qvfb -system-jpeg -gif -release -platform
linux-g++ -xplatform linux-arm-g++ -thread -tslib -I$PREFIX/include -L$PREFIX/lib "
export QPE_CFG="-edition pda -displaysize 480x272 -no-qtopiadesktop -release -platform linux-g++ -xplatform
linux-arm-g++ -I$PREFIX/include -L$PREFIX/lib "
echo 'yes' | ./configure
```

再次 patch:

```
./../patch/patch_file.sh
```

运行如下命令开始编译并安装:

```
make;make install
```

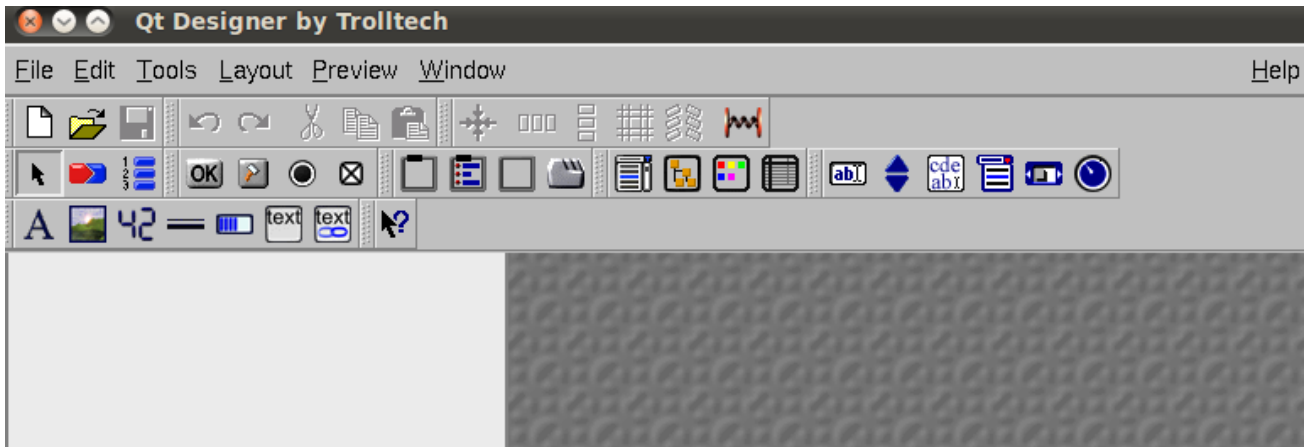
安装后的用于在 ARM 上运行的库文件:

```
cust@u1004:/samlinux/qtopia/qtopia-free-2.2.0-arm$ ls ../_install/lib/
e2initrd_helper  libcom_err.so.2  libmediaplayer.so.2  libqmstroke.so.2
fonts           libcom_err.so.2.1  libmediaplayer.so.2.2  libqmstroke.so.2.2
libamr.so       libe2p.so        libmediaplayer.so.2.2.0  libqmstroke.so.2.2.0
libamr.so.1     libe2p.so.2     libopenobex.so        libqpepim1.so
libamr.so.1.0  libe2p.so.2.3  libopenobex.so.2     libqpepim1.so.2
libamr.so.1.0.0  libext2fs.so    libopenobex.so.2.2   libqpepim1.so.2.1
libavcodec.so  libext2fs.so.2  libopenobex.so.2.2.0  libqpepim1.so.2.1.0
libavcodec.so.2  libext2fs.so.2.4  libpng12.a           libqpepim.so
libavcodec.so.2.2  libgsm.so      libpng12.la          libqpepim.so.1
libavcodec.so.2.2.0  libgsm.so.1    libpng12.so         libqpepim.so.1.6
libavformat.so  libgsm.so.1.0  libpng12.so.0       libqpepim.so.1.6.3
libavformat.so.2  libgsm.so.1.0.0  libpng12.so.0.43.0  libqpe.so
libavformat.so.2.2  libjpeg.a      libpng.a            libqpe.so.1
libavformat.so.2.2.0  libjpeg.la    libpng.la           libqpe.so.1.5
libblkid.so     libjpeg.so     libpng.so           libqpe.so.1.5.3
libblkid.so.1   libjpeg.so.62  libpng.so.3        libqte.so
libblkid.so.1.0  libjpeg.so.62.0.0  libpng.so.3.43.0  libqte.so.2
libcom_err.so   libmediaplayer.so  libqmstroke.so     libqte.so.2.3
cust@u1004:/samlinux/qtopia/qtopia-free-2.2.0-arm$ file ../_install/lib/libqpe.so.1.5.3
../_install/lib/libqpe.so.1.5.3: ELF 32-bit LSB shared object, ARM, version 1 (SYSV), dyn
cust@u1004:/samlinux/qtopia/qtopia-free-2.2.0-arm$
```

以及生成的 x86 版本的开发工具:

```
cust@u1004:/samlinux/qtopia/qtopia-free-2.2.0-arm$ ls qt2/bin/
designer moc qvfb syncqt uic
cust@u1004:/samlinux/qtopia/qtopia-free-2.2.0-arm$ file qt2/bin/designer
qt2/bin/designer: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
not stripped
cust@u1004:/samlinux/qtopia/qtopia-free-2.2.0-arm$
```

尝试在 ubuntu 上启动 designer:



### 3. 使用 Qtopia

前面只是将 qtopia 及其依赖进行了编译,要在板子上运行,还需要一个测试环境: 一个可以运行的 Linux 系统. 同时,由于 qtopia 是 RGB,而 SAM9263 的 LCDC 是 IBGR,因此测试中使用了一块转换板.

Memory Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RGB 5-6-5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
LCD Controller BGR Format	I	B4	B3	B2	B1	B0	G4	G3	G2	G1	G0	R4	R3	R2	R1	R0
LCD Controller Interface	D18	D23	D22	D21	D20	D19	D15	D14	D13	D12	D11	D7	D6	D5	D4	D3
Display Interface	R5	R4	R3	R2	R1	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1

转换板的具体连接方法可以参考 atmel 的文档 [doc6300](#).

### 3.1 编译 kernel

因为使用了 4.3" 的 TFT LCD, kernel 中需要做一些修改.

首先是 board-sam9263.c 中关于 LCD 的部分:

```

265 /*
266 * LCD Controller
267 */
268 #if defined(CONFIG_FB_ATMEL) || defined(CONFIG_FB_ATMEL_MODULE)
269 static struct fb_videomode at91_tft_vga_modes[] = {
270     {
271         .name           = "TX09D50VM1CCA @ 60",
272         .refresh        = 60,
273         .xres           = 480, .yres           = 272,
274         .pixclock       = KHZ2PICOS(9000),
275
276         .left_margin    = 2, .right_margin   = 2,
277         .upper_margin   = 2, .lower_margin  = 2,
278         .hsync_len     = 41, .vsync_len    = 10,
279
280         .sync           = FB_SYNC_HOR_HIGH_ACT | FB_SYNC_VERT_HIGH_ACT,
281         .vmode          = FB_VMODE_NONINTERLACED,
282     },
283 };

```

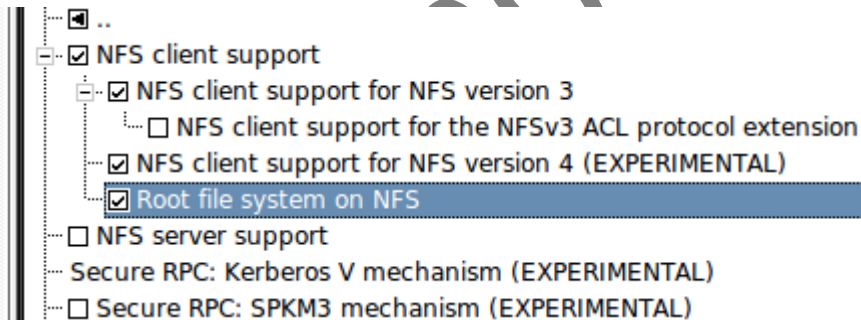
以及系统时钟:

```

55 static void __init ek_map_io(void)
56 {
57     /* Initialize processor: 16.367 MHz crystal */
58     at91sam9263_initialize(18432000); // 16367660
59
60     /* DGBU on ttyS0. (Rx & Tx only) */
61     at91_register_uart(0, 0, 0);

```

然后在内核配置中打开 nfs 的支持:



重新编译内核,并生成 u-boot 可以启动的 ulmage.

### 3.2 rootfs

由于前面只是编译了 qtopia 的部分, 而 Linux 启动本身所需要的 rootfs 不在本文的讨论范围之内.

对 rootfs 的要求就是使用 arm-none-linux-gnueabi 编译,并在/lib 下有对应的库文件.

### 3.3 准备 NFS

用前面编译的新的 kernel 启动系统,并登录.

首先需要设置网络,将开发板与路由器通过网线连接.设置好开发板的 IP. 这里假定开发板 IP 为 192.168.1.100, ubuntu Linux 开发机 IP 为 192.168.1.11.

ubuntu 开发机上的 nfs server 根目录为/samlinux/nfsroot,将前面编译生成的文件全部复制到 nfs 根目录下的某个文件夹,比如 qt:

```
cust@u1004:/samlinux/qtopia$ cd _install/
cust@u1004:/samlinux/qtopia/_install$ ls
apps bin etc help i18n include info lib man pics plugins sbin services share sounds
cust@u1004:/samlinux/qtopia/_install$ mkdir /samlinux/nfsroot/qt
cust@u1004:/samlinux/qtopia/_install$ cp -a * /samlinux/nfsroot/qt
cust@u1004:/samlinux/qtopia/_install$ ls /samlinux/nfsroot/qt
apps bin etc help i18n include info lib man pics plugins sbin services share sounds
cust@u1004:/samlinux/qtopia/_install$
```

然后在开发板上通过 NFS 将 qtopia 的文件 mount.

首先创建 mount 点,比如:

```
[root@mcuzone /]#mkdir -p /samlinux/qtopia/_install
```

然后通过 nfs 进行 mount:

```
[root@mcuzone /]#mount -t nfs 192.168.1.11:/samlinux/nfsroot/qt /samlinux/qtopia
/_install/ -o nolock
[root@mcuzone /]#df
Filesystem            1k-blocks      Used Available Use% Mounted on
ubi0_0                57712         47892    9820   83% /
tmpfs                 512            0        512    0% /dev
tmpfs                 4096            8       4088    0% /var/volatile
192.168.1.11:/samlinux/nfsroot/qt
16508576             4340256   11329728   28% /samlinux/qtopia/_install
[root@mcuzone /]#
```

mount 后的效果:

```
[root@mcuzone _install]#pwd
/samlinux/qtopia/_install
[root@mcuzone _install]#ls
apps      help      info      pics      services
bin       i18n     lib       plugins   share
etc       include  man       sbin      sounds
[root@mcuzone _install]#
```

### 3.4 准备运行环境

在运行 qt 之前,还需要准备一些环境,包括环境变量设置,触摸屏校准等.

先看下 qpe 的依赖:

```
[root@mcuzone _install]#pwd
/samlinux/qtopia/_install
[root@mcuzone _install]#ldd ./bin/qpe
libqtopia2.so.2 => not found
libqtopia.so.1 => not found
libqpe.so.1 => not found
libqte.so.2 => not found
libpng12.so.0 => not found
libts-0.0.so.0 => /usr/lib/libts-0.0.so.0 (0x40027000)
libz.so.1 => not found
libuuid.so.1 => not found
libjpeg.so.62 => not found
libstdc++.so.6 => /usr/lib/libstdc++.so.6 (0x40031000)
libm.so.6 => /lib/libm.so.6 (0x4010b000)
libgcc_s.so.1 => /lib/libgcc_s.so.1 (0x401ad000)
libc.so.6 => /lib/libc.so.6 (0x401c1000)
libcrypt.so.1 => /lib/libcrypt.so.1 (0x40309000)
libdl.so.2 => /lib/libdl.so.2 (0x40345000)
/lib/ld-linux.so.3 (0x40000000)
[root@mcuzone _install]#
```

很多库都没有找到,需要更新环境变量:

```
[root@mcuzone _install]#pwd
/samlinux/qtopia/_install
[root@mcuzone _install]#env
TSLIB_TSDEVICE=/dev/event1
USER=root
HOSTNAME=mcuzone
LD_LIBRARY_PATH=/lib:/usr/lib
OLDPWD=/
HOME=/home/root
TSLIB_FBDEVICE=/dev/fb0
TSLIB_PLUGINDIR=/usr/lib/ts
TSLIB_CONSOLEDEVICE=none
LOGNAME=root
TERM=vt100
PATH=/bin:/sbin:/usr/bin:/usr/sbin
TSLIB_CONFFILE=/etc/ts.conf
SHELL=/bin/sh
PWD=/samlinux/qtopia/_install
TSLIB_CALIBFILE=/var/volatile/pointercal
[root@mcuzone _install]#pwd
/samlinux/qtopia/_install
[root@mcuzone _install]#export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$PWD/lib
[root@mcuzone _install]#echo LD_LIBRARY_PATH
LD_LIBRARY_PATH
[root@mcuzone _install]#echo $LD_LIBRARY_PATH
/lib:/usr/lib:/samlinux/qtopia/_install/lib
[root@mcuzone _install]#
```

再次查看 qpe 的库依赖:

```
[root@mcuzone _install]#ldd ./bin/qpe
libqtopia2.so.2 => /samlinux/qtopia/_install/lib/libqtopia2.so.2 (0x40027000)
libqtopia.so.1 => /samlinux/qtopia/_install/lib/libqtopia.so.1 (0x40071000)
libqpe.so.1 => /samlinux/qtopia/_install/lib/libqpe.so.1 (0x400ce000)
libqte.so.2 => /samlinux/qtopia/_install/lib/libqte.so.2 (0x401ae000)
libpng12.so.0 => /samlinux/qtopia/_install/lib/libpng12.so.0 (0x405b1000)
libts-0.0.so.0 => /usr/lib/libts-0.0.so.0 (0x405f0000)
libz.so.1 => /samlinux/qtopia/_install/lib/libz.so.1 (0x405fa000)
libuuid.so.1 => /samlinux/qtopia/_install/lib/libuuid.so.1 (0x40622000)
libjpeg.so.62 => /samlinux/qtopia/_install/lib/libjpeg.so.62 (0x4062e000)
libstdc++.so.6 => /usr/lib/libstdc++.so.6 (0x4065f000)
libm.so.6 => /lib/libm.so.6 (0x40739000)
libgcc_s.so.1 => /lib/libgcc_s.so.1 (0x407db000)
libc.so.6 => /lib/libc.so.6 (0x407ef000)
libcrypt.so.1 => /lib/libcrypt.so.1 (0x40937000)
libdl.so.2 => /lib/libdl.so.2 (0x40973000)
/lib/ld-linux.so.3 (0x40000000)
[root@mcuzone _install]#
```

在校准触摸屏之前,需要设置 tslib 的一些环境变量.



首先将/samlinux/qtopia/\_install/etc 下的 ts.conf 复制到/etc 下,并打开 module raw:

```
# Uncomment if you wish to use the linux input layer event interface
module_raw input

# Uncomment if you're using a Sharp Zaurus SL-5500/SL-5000d
# module_raw collie

# Uncomment if you're using a Sharp Zaurus SL-C700/C750/C760/C860
# module_raw corqi
```

确定触摸屏的设备号以及 frame buffer 设备号:

```
[root@mcuzone _install]#cat /proc/bus/input/devices
I: Bus=0019 Vendor=0001 Product=0001 Version=0100
N: Name="gpio-keys"
P: Phys=gpio-keys/input0
S: Sysfs=/class/input/input0
U: Uniq=
H: Handlers=event0
B: EV=3
B: KEY=30000 0 0 0 0 0 0 0 0

I: Bus=0000 Vendor=0000 Product=0000 Version=0000
N: Name="ADS784x Touchscreen"
P: Phys=spi0.3/input0
S: Sysfs=/class/input/input1
U: Uniq=
H: Handlers=mouse0 event1
B: EV=b
B: KEY=400 0 0 0 0 0 0 0 0 0
B: ABS=1000003

[root@mcuzone _install]#
```

```
[root@mcuzone _install]#ls /dev/fb*
/dev/fb0
[root@mcuzone _install]#
```

根据以上信息,设置如下环境变量:

```
export TSLIB_PLUGINDIR=/samlinux/qtopia/_install/lib/ts
export TSLIB_TSDEVICE=/dev/event1
export TSLIB_CONFFILE=/etc/ts.conf
export TSLIB_FBDEVICE=/dev/fb0
export TSLIB_CONSOLEDEVICE=none
export TSLIB_CALIBFILE=/etc/pointercal
```

开始校准,运行如下命令:

```
[root@mcuzone _install]#pwd
/samlinux/qtopia/_install
[root@mcuzone _install]#./bin/ts_calibrate
xres = 480, yres = 272
Took 16 samples...
Top left : X = 3517 Y = 3111
Took 18 samples...
Top right : X = 515 Y = 2999
Took 20 samples...
Bot right : X = 483 Y = 832
Took 20 samples...
Bot left : X = 3519 Y = 873
Took 15 samples...
Center : X = 2012 Y = 1952
491.287842 -0.125886 0.000841
284.567627 0.001973 -0.078086
Calibration constants: 32197040 -8250 55 18649424 129 -5117 65536
[root@mcuzone _install]#
```

生成了校准文件:

```
[root@mcuzone _install]#ls /etc/pointerca
/etc/pointerca
[root@mcuzone _install]#
```

校准完成运行测试程序检验效果:

```
[root@mcuzone _install]#pwd
/samlinux/qtopia/_install
[root@mcuzone _install]#./bin/ts_test
2179.750528: 220 192 7500
2179.780516: 220 192 7500
2179.780516: 220 192 7500
2179.800504: 220 192 7500
2179.800504: 220 192 7500
2179.810471: 220 191 7500
2179.830463: 220 191 7500
```

设置 qt 的环境变量:

```
export QTDIR=/samlinux/qtopia/_install
export QPEDIR=$QTDIR
export PATH=$QPEDIR/bin:$PATH
export QWS_MOUSE_PROTO=TPanel:/dev/ts
export QWS_SIZE=480x272
export QWS_DISPLAY=LinuxFb:mmWidth35:mmHeight45:0
```

创建一些文件夹:

```
mkdir -p $QTDIR/etc/keytabs
mkdir -p $QTDIR/etc/themes
mkdir -p $QTDIR/services/TimeMonitor
mkdir -p /home/root/Applications/qtmail
ln -s /var/volatile/ /var/spool
mkdir /var/spool/at
```

### 3.5 字体和资源文件

在 linux 主机上复制一个时区信息文件夹到 nfs 目录下:

```
cust@u1004:/samlinux/nfsroot/qt$ pwd
/samlinux/nfsroot/qt
cust@u1004:/samlinux/nfsroot/qt$ cp -a /usr/share/zoneinfo/America/ .
cust@u1004:/samlinux/nfsroot/qt$
```

这样在开发板上也可以看到这个文件夹:

```
[root@mcuzone _install]#pwd
/samlinux/qtopia/_install
[root@mcuzone _install]#ls
America  etc      include  man      sbin     sounds
apps     help    info     pics     services
bin      i18n   lib      plugins  share
[root@mcuzone _install]#
```

将此文件夹复制到开发板的对应目录下(/usr/share/zoneinfo/America):

```
[root@mcuzone _install]#mkdir -p /usr/share/zoneinfo/
[root@mcuzone _install]#mv America/ /usr/share/zoneinfo/
[root@mcuzone _install]#ls /usr/share/zoneinfo/
America Asia
[root@mcuzone _install]#
```

使用相同的方法,将 ubuntu 上的/usr/share/zoneinfo/zone.tab 也复制到开发板的对应目录:

```
[root@mcuzone _install]#ls /usr/share/zoneinfo/
America Asia zone.tab
[root@mcuzone _install]#
```

在 qtopia-2.2.0 的源码目录(qt2/lib/fonts)下复制字体文件到 nfs 对应目录下:

```
cust@u1004:/samlinux/qtopia/qtopia-free-2.2.0-arm/qt2/lib/fonts$ cp -a * /samlinux/nfsroot/qt/lib/fonts/
cust@u1004:/samlinux/qtopia/qtopia-free-2.2.0-arm/qt2/lib/fonts$
```

在开发板上就能看到对应的字体了:

```
[root@mcuzone _install]#ls lib/fonts/
5x7.bdf smoothmono_180_50_t15.qpf
README smoothmono_180_50_t5.qpf
UTBI____.pfa smoothmono_180_50i.qpf
UTB____.pfa smoothmono_180_50i_t10.qpf
UTI____.pfa smoothmono_180_50i_t15.qpf
UTRG____.pfa smoothmono_180_50i_t5.qpf
Vera.ttf smoothmono_180_75.qpf
VeraBI.ttf smoothmono_180_75_t10.qpf
VeraBd.ttf smoothmono_180_75_t15.qpf
VeraCopyright.txt smoothmono_180_75_t5.qpf
VeraTt.ttf smoothmono_180_75i.qpf
```

并不是所有字体都需要,但是测试的时候所有的都复制,应用中应做裁剪,以节省空间.

## 3.6 运行 qtopia

通过如下命令启动 qtopia:

```
[root@mcuzone _install]#pwd
/samlinux/qtopia/_install
[root@mcuzone _install]#./bin/qpe
```

在欢迎界面做些选择之后就会进入桌面.



Notes 程序显示中文 txt 文件的效果图:

