

# Qt 开发环境搭建和移植

By Richard.Lee

[kmalloc@163.com](mailto:kmalloc@163.com)

## 一、使用 **BusyBox** 构建根文件系统

### 1. 移植 **BusyBox**

下载 busybox 的源码包，最新版本为 busybox-1.17.2

<http://www.busybox.net/downloads/>

解压并进入目录:

```
# tar jxvf busybox-1.17.2.tar.bz2
# cd busybox-1.17.2
```

修改 Makefile:

```
# gedit Makefile &
```

把 164 行修改为:

```
CROSS_COMPILE = /usr/local/arm/4.3.2/bin/arm-linux-
```

把 189 行修改为:

```
ARCH = arm
```

配置:

```
# make menuconfig
```

选上以下几项:

```
Busybox Settings --->
  Build Options --->
    [*] Build BusyBox as a static binary (no shared libs)
    [*] Build with Large File Support (for accessing files > 2 GB)
  Busybox Library Tuning --->
    [*] vi-style line editing commands
    [*] Fancy shell prompts
Linux Module Utilities --->
  [ ] Simplified modutils
  [*] insmod
  [*] rmmod
  [*] lsmod
  [*] modprobe
  [*] depmod
```

退出并保存配置。

编译:

```
# make
```

安装:

```
# make install
```

在 `busybox-1.17.2` 目录下可以找到 `_install` 子目录，这就是刚才的安装目录。

## 2. 加入必要的目录与文件

改名为最小化根文件系统目录:

```
# mv _install mini_rootfs
```

进入最小化根文件系统目录:

```
# cd mini_rootfs
```

在 `mini_rootfs` 目录里建立一个名为 `create_rootfs.sh` 的脚本文件:

```
#!/bin/sh

#root dir
mkdir bin sbin lib etc dev sys proc tmp var opt mnt usr home root media

#usr sub dir
cd usr
mkdir bin sbin lib local
#usr/local sub dir
cd local
mkdir bin sbin lib
cd ../..

#etc sub dir
cd etc
touch inittab
touch fstab
touch profile
touch passwd
touch group
touch shadow
touch resolv.conf
touch mdev.conf
touch inetd.conf
mkdir rc.d
```

```
mkdir init.d
touch init.d/rcS
chmod +x init.d/rcS
mkdir sysconfig
touch sysconfig/HOSTNAME
cd ..

#dev sub dir
cd dev
mknod console c 5 1
chmod 777 console
mknod null c 1 3
chmod 777 null
cd ..

#var sub dir
cd var
mkdir log
cd ..
```

为 create\_rootfs.sh 脚本加上可执行权限:

```
# chmod +x create_rootfs.sh
```

执行该脚本以生成最小化根文件系统所需要的目录与文件:

```
# ./create_rootfs.sh
```

拷贝交叉编译器里的动态链接库到最小化根文件系统的 lib 和 usr/lib 目录:

```
# cp -f /usr/local/arm/4.3.2/arm-none-linux-gnueabi/libc/armv4t/lib/*so* lib -a
```

```
# cp -f /usr/local/arm/4.3.2/arm-none-linux-gnueabi/libc/armv4t/usr/lib/*so*
usr/lib -a
```

etc/fstab 文件的内容为:

```
# /etc/fstab: static file system information.
#
# Use 'vol_id --uuid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way
# to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc defaults 0 0
sysfs /sys sysfs defaults 0 0
tmpfs /var tmpfs defaults 0 0
tmpfs /tmp tmpfs defaults 0 0
tmpfs /dev tmpfs defaults 0 0
```

etc/inittab 文件的内容为:

```
::sysinit:/etc/init.d/rcS
s3c2410_serial0::askfirst:/bin/sh
::ctrlaltdel:/sbin/reboot
::shutdown:/bin/umount -a -r
::restart:/sbin/init
```

etc/profile 文件的内容为:

```
# Ash profile
# vim: syntax=sh

# No core files by default
ulimit -S -c 0 > /dev/null 2>&1

USER="`id -un`"
LOGNAME=$USER
PS1='[\u@\h \W]\# '
PATH=$PATH

HOSTNAME=`/bin/hostname`

export USER LOGNAME PS1 PATH

export LD_LIBRARY_PATH=lib:/usr/lib:$LD_LIBRARY_PATH
```

etc/init.d/rcS 文件的内容为:

```
#!/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin

runlevel=S
prevlevel=N

umask 022

export PATH runlevel prevlevel

mount -a

mkdir /dev/pts
mount -t devpts devpts /dev/pts
echo /sbin/mdev > /proc/sys/kernel/hotplug
mdev -s

/bin/hostname -F /etc/sysconfig/HOSTNAME

ifconfig eth0 192.168.0.7
```

resolv.conf 文件的内容为:

```
nameserver 202.96.128.166
```

etc/inetd.conf 是 inetd 的配置文件，它的内容为：

```
# It is generally considered safer to keep these off.
echo stream tcp nowait root internal
echo dgram udp wait root internal
daytime stream tcp nowait root internal
daytime dgram udp wait root internal
time stream tcp nowait root internal
time dgram udp wait root internal

# These are standard services.
ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd
telnet stream tcp nowait root /sbin/telnetd /sbin/telnetd
```

etc/sysconfig/HOSTNAME 文件的内容为你的英文名字。

etc/passwd、etc/group、etc/shadow 文件的内容为空。

etc/mdev.conf 是 mdev 设备文件系统的配置文件，可用不做。

OK，算是做好了～～

为了正常使用，你还需要在它的 dev 目录里建立如下两个设备文件：

```
cd dev
mknod console c 5 1
chmod 777 console
mknod null c 1 3
chmod 777 null
cd ..
```

## 二、移植触摸屏库 tslib

1. 下载相关工具与最新的 tslib 源码包：

```
# sudo apt-get install autoconf
# sudo apt-get install automake
# sudo apt-get install libtool
```

RHEL 不需要下载以上文件。

2. 运行 autogen.sh 脚本：

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

3. 配置:

```
# ./configure --host=arm-linux --prefix=/opt/tslib  
ac_cv_func_malloc_0_nonnull=yes --enable-inputapi=no
```

4. 编译:

```
# make
```

5. 安装:

```
# make install
```

安装后, 主机 /opt/tslib 存在如下目录:

```
bin etc include lib
```

拷贝主机 /opt/tslib 目录至开发板 /opt/ 目录。

7. 在开发板运行触摸屏校正:

(1) 在运行触摸屏校正之前, 需要设置一些环境变量, 以下是我的开发板的 /etc/profile 文件的内容:

```
# Ash profile  
# vim: syntax=sh  
  
# No core files by default  
ulimit -S -c 0 > /dev/null 2>&1  
  
USER="`id -un`"  
LOGNAME=$USER  
PS1='[\u@\h \W]\# '  
PATH=$PATH  
  
HOSTNAME=`/bin/hostname`  
  
export USER LOGNAME PS1 PATH  
  
export TSLIB_ROOT=/opt/tslib  
export TSLIB_TSDEVICE=/dev/event0  
export TSLIB_CONFFILE=$TSLIB_ROOT/etc/ts.conf  
export TSLIB_PLUGINDIR=$TSLIB_ROOT/lib/ts  
export TSLIB_CALIBFILE=/etc/pointercal  
export TSLIB_CONSOLEDEVICE=none  
export TSLIB_FBDEVICE=/dev/fb0  
export TS_INFO_FILE=/sys/class/input/input0/uevent  
  
export LD_LIBRARY_PATH=lib:/usr/lib:$TSLIB_ROOT/lib:$LD_LIBRARY_PATH
```

(2) 运行触摸屏校正:

```
# cd /opt/tslib/bin  
# ./ts_calibrate
```

报错 “No raw modules loaded.”，原因是缺省时，/opt/tslib/etc/ts.conf 中所有的 raw module 都注释掉了，打开第一项 “module\_raw input” 即可。

(3) 重新运行，应该成功；点击屏幕上出现的需要校正的 “+”，校正结果存放在 /etc/pointercal 文件中。

### 三、移植 Qt-embedded 4.5.2 for 2440

请先完成

使用 **BusyBox** 构建根文件系统 完成根文件系统的构建！

移植触摸屏库 **tslib** 完成 **tslib** 的移植与安装！

请使用 OpenMoko 提供的交叉编译器，如果使用友善之臂公司或勤研或 CodeSourcery 提供的交叉编译器来编译 Qt/E 4.5.2，可能会导致 Qt 程序运行时出现段错误！

<http://qtextended.org/modules/mydownloads/visit.php?lid=38>

安装新的交叉编译器：

```
# tar zxvf arm920t-eabi.tgz -C /
```

会安装至 /opt/toolchains/arm920t-eabi/ 目录。

把 "/opt/toolchains/arm920t-eabi/arm-angstrom-linux-gnueabi/lib" 目录下的 libz.so libz.so.1 libz.so.1.2.3 三个动态链接库文件拷贝至开发板 /lib 目录，因为 Qt 要用到！

换个新的终端执行：

```
# export PATH=/opt/toolchains/arm920t-eabi/bin/:$PATH
```

因为要使用新的交叉编译器！

下载 Qt/E 4.5.2 源码包：

<http://get.qt.nokia.com/qt/source/qt-embedded-linux-opensource-src-4.5.2.tar.gz>

解压 Qt/E 源码包并进入目录：

```
# tar zxvf qt-embedded-linux-opensource-src-4.5.2.tar.gz
```

```
# cd qt-embedded-linux-opensource-src-4.5.2
```

配置：

```
# ./configure -prefix /opt/arm-qt4 -embedded arm -xplatform qws/linux-arm-g+  
+ -depths 16 -qt-mouse-tslib -I/opt/tslib/include -L/opt/tslib/lib
```

编译：

```
# make
```

出错，缺少 zlib 库，如果是 ubuntu 的话，可以通过以下的命令来安装 zlib-dev：

```
(ubuntu)# apt-get install zlib1g-dev
```

```
(RedHat)# yum -y install zlib*
```

再编译:  
`# make`

安装:  
`# make install`

下载 `/opt/arm-qt4` 目录到开发板 `/opt/` 目录，可去掉一些例子与字体。

需要设置一些环境变量，以下是我的开发板的 `/etc/profile` 文件:

```
# Ash profile
# vim: syntax=sh

# No core files by default
ulimit -S -c 0 > /dev/null 2>&1

USER="\`id -un`"
LOGNAME=$USER
PS1='\u@\h \W]\# '
PATH=$PATH

HOSTNAME=`/bin/hostname`

export USER LOGNAME PS1

export TSLIB_ROOT=/opt/tslib
export TSLIB_TSDEVICE=/dev/event0
export TSLIB_CONFFILE=$TSLIB_ROOT/etc/ts.conf
export TSLIB_PLUGINDIR=$TSLIB_ROOT/lib/ts
export TSLIB_CALIBFILE=/etc/pointercal
export TSLIB_CONSOLEDEVICE=none
export TSLIB_FBDEVICE=/dev/fb0
export TS_INFO_FILE=/sys/class/input/input0/uevent

export QTDIR=/opt/arm-qt4
export QPEDIR=/opt/arm-qt4
export QWS_KEYBOARD=TTY:/dev/tty1
export QWS_MOUSE_PROTO=Tslib:/dev/event0

export PATH=$QTDIR/bin:$PATH

export QT_QWS_FONTDIR=/opt/arm-qt4/lib/fonts

export LD_LIBRARY_PATH=lib:/usr/lib:$TSLIB_ROOT/lib:$QTDIR/lib:
$LD_LIBRARY_PATH
```

在开发板上运行 Qt 自带的测试例子:

`# /opt/arm-qt4/demos/embedded/fluidlauncher/fluidlauncher -qws`



QFontDatabase: Cannot find font directory /opt/arm-qt4/lib/font - is Qt installed correctly

解决办法:

```
export QT_QWS_FONTDIR=/opt/arm-qt4/lib/fonts
```

## 四、程序编译：

1. 编写环境变量设置文件（方便不同环境切换）：

PC 机的 ~/.bashrc 中添加:

```
export PATH=$PATH:/usr/local/arm/4.1.2/bin
```

vim ~/setenv-qt-x11.sh（对应于 PC 版 Qt）：

```
PATH=/usr/local/Trolltech/Qt-4.6.0/bin:$PATH
```

```
LD_LIBRARY_PATH=/usr/local/Trolltech/Qt-4.6.0/lib:$LD_LIBRARY_PATH
```

```
CPLUS_INCLUDE_PATH=
```

vim ~/setenv-qt-x86.sh:

```
QTEDIR=/usr/local/Trolltech/QtEmbedded-4.6.0
```

```
PATH=/usr/local/Trolltech/QtEmbedded-4.6.0/bin:$PATH
```

```
LD_LIBRARY_PATH=/usr/local/Trolltech/QtEmbedded-4.6.0/lib:$LD_LIBRARY_PATH
```

vim ~/setenv-qt-arm.sh:

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

```
#QTEDIR=/usr/local/Trolltech/QtEmbedded-4.6.2-arm
```

```
QTEDIR=/opt/arm-qt4
```

```
CROSS_COMPILER=/opt/toolchains/arm920t-eabi
```

```
PATH=$QTEDIR/bin:$PATH
```

```
LD_LIBRARY_PATH=$QTEDIR/lib:$LD_LIBRARY_PATH
```

```
CPLUS_INCLUDE_PATH=$CROSS_COMPILER/arm-angstrom-linux-gnueabi/include/c++:
```

```
$CROSS_COMPILER/arm-angstrom-linux-gnueabi/include/c++/arm-angstrom-linux-gnueabi:
```

```
$CPLUS_INCLUDE_PATH
```

PC 版:

```
$ make distclean
```

```
$ source ~/.setenv-qt-x11.sh
```

```
$ qmake
```

```
$ make
```

arm 版:

```
$ make distclean
```

```
$ source ~/.setenv-qt-arm.sh
```

```
$ qmake
```

```
$ make
```

## 五、程序开发

### 1. 下载 Qt Creator :

Qt Creator 2.0.1 Binary for Linux/X11 32-bit (45 MB)

<http://get.qt.nokia.com/qtcreator/qt-creator-linux-x86-opensource-2.0.1.bin>

## **2. Qt Creator 的安装**

`./qt-creator-linux-x86-opensource-2.0.1.bin`

## **3. 使用 Qt Creator 进行开发**

<http://blog.csdn.net/yafeilinux/archive/2010/03/29/5429530.aspx>