

文件系统制作手册

2010-06-17 V1.4

手册内容简介：

本手册是天祥电子推出的 TX-2440A 开发板的配套手册之一，全面分析了 linux 文件系统的制作和移植过程，手册中的部分内容都会在配套的视频教程中（第十七讲，第十八讲）进行讲解。

在开始进行制作之前，需要了解文件系统的相关知识，最主要是熟悉 linux 文件系统的目录结构，还有构成文件系统的最基本的文件；建议先看视频教程学习，在具备了一定的能力后再来参考本手册来操作。

本手册分为 4 步进行制作：

一、准备制作（创建系统目录）

二、编译 busybox（制作文件系统的必备工具）

三、制作文件系统（添加必备的文件，制作镜像）

四、完善文件系统（根据需要，移植更多的应用程序）

制作好文件系统后，就可以方便的向里面添加更多的应用程序了，可以参考我写的几篇应用程序的移植手册。我也会陆续的移植更多的应用程序，制作移植手册或视频教程，如果你有好的想法，可以与我联系，我们一起开发应用程序。

说明：

由于个人能力有限，手册中难免会出现一些笔误和不足之处，如果发现问题，请及时提出，目前可以发到 QQ 群上（103105892 111874027），或发到我的博客上（blog.163.com/xgc94418297）；在学习过程中遇到的问题，可以联系我们，我们会提供技术支持。

本手册中的所有内容目前仅适用于 TX2440A 开发板，如果将其用在其他的开发板上，出现的一切问题，我们一律不提供技术支持。

最好不要从手册中直接复制代码，因为在编写手册时，有些字符可能会自动变成全角格式，在代码中是不允许有全角字符的，所以有可能会产生一些错误。我建议，最好是手动编写代码。

手册中的内容会不定期的更新，以后还会加入更丰富的内容，更新后会放到 FTP 服务器供大家下载。希望广大的 TX2440A 开发板用户能多提出宝贵的意见，我会根据大家的意见及时调整手册中的内容。

2010-06-17 相广超 制作
哈尔滨祥鹏科技有限公司
网址：www.txmcu.com
电话：0451-87572303

内核版本:

Linux-2.6.31

交叉编译器版本:

arm-linux-gcc 4.1.2

操作系统平台:

Linux -- Red Hat 9.0

开发板平台:

Arm -- TX2440A

使用的工具:

Busybox-1.15.2.tar.bz2

(源码位置: 光盘资料/linux 平台工具/)

手册中字体颜色的约定:

修改的代码用红色字体

执行的命令用红色字体, 前面加 #

添加的大段代码用蓝色字体

在终端上打印出的信息用紫色字体

出现的错误信息用绿色字体

说明的文字用红色加粗字体

版本信息:

2009-10-21	V1.0	初稿
2010-03-16	V1.1	完善了整体内容
2010-04-15	V1.2	加入了 Linux 系统下应用程序和工具使用方法 系统下驱动测试程序使用方法
2010-04-21	V1.3	新增加 串口 1 测试方法
2010-06-17	V1.4	更新了最后两部分内容

STEP 1: 准备制作

创建根文件系统目录，主要包括以下目录：

/bin, /etc, /dev, /home, /lib, /mnt, /opt, /proc, /root, /sbin, /sys, /tmp, /usr, /var.

建一个名为 root_2.6.31 的文件夹

```
#mkdir root_2.6.31
```

在该文件夹下创建以上目录

```
#mkdir bin etc dev .....
```

STEP 2: 编译 busybox

获取 busybox1.15.2 源码，在 <http://www.busybox.net/downloads/> 找到 [busybox-1.15.2.tar.bz2](http://www.busybox.net/downloads/busybox-1.15.2.tar.bz2) 这个压缩包，下载，解压：

```
#tar xjvf busybox-1.15.2
```

进入源码目录，修改 Makefile

第 164 行，`CROSS_COMPILE = arm-linux-`

第 190 行，`ARCH = arm`

执行 `#make menuconfig` 进行配置 busybox

配置选项大部分都是保持默认的，只需要注意选择以下这几个选项，其他的选项都不用动：

Busybox Setting --->

Build Options --->

[*] Build Busybox as a static binary (no shared libs)

[*] Build with Large File Support (for accessing files > 2GB)

Installation Options --->

(./_install) Busybox installation prefix

进入这个选项，输入 busybox 的安装路径，如： `../rootfs`

Busybox Library Tuning --->

[*] vi-style line editing commands

[*] Fancy shell prompts

要选择这个选项：“Fancy shell prompts”，否则挂载文件系统后，无法正常显示命令提示符：“`[\u@\h \W]#`”

配置完后，执行 `#make make install`

然后就会在上一级目录下生成 **rootfs** 文件夹，里面包含几个文件夹 **/bin /sbin /usr linuxrc**

把这些文件全部复制到刚建好的 **root_2.6.31** 目录下，

```
#cp -rf * ../root_2.6.31
```

STEP 3: 制作文件系统

在 **dev** 目录下，创建两个设备节点：

```
#mknod console c 5 1
```

```
#mknod null c 1 3
```

进入 **etc** 目录，添加文件：

拷贝 **Busybox-1.15.2/examples/bootfloopy/etc/*** 到当前目录下。

```
# cp -r ../../busybox-1.15.2/examples/bootfloopy/etc/* ./
```

包括文件：**fstab init.d inittab profile**

拷贝 **/etc/passwd, /etc/group, /etc/shadow** 到当前目录下。

```
# cp /etc/passwd ./
```

```
# cp /etc/group ./
```

```
# cp /etc/shadow ./
```

把 **passwd** 文件中的第一行：**root:x:0:0:root:/root:/bin/bash** 中的 **/bin/bash**，改成 **/bin/ash**

因为文件系统的 **bin** 目录下没有 **bash** 这个命令，而是用 **ash** 代替 **bash**，所以在用用户名密码登录的时候(如 **telnet**)，会出现 “**cannot run /bin/bash: No such file or directory**” 的错误。

修改 **inittab** 文件：

内容如下：

```
console::sysinit:/etc/init.d/rcS
ttyS0::respawn:-/bin/sh
s3c2410_serial0::askfirst:-/bin/sh
::once:/usr/sbin/telnetd -l /bin/login
::ctrlaltdel:/bin/umount -a -r
```

修改 **fstab** 文件，内容如下：

```
proc          /proc      proc      defaults    0    0
tmpfs         /tmp       tmpfs     defaults    0    0
```

```
sysfs      /sys      sysfs      defaults    0    0
tmpfs      /dev      tmpfs      defaults    0    0
var         /dev      tmpfs      defaults    0    0
```

修改 **init.d/rcS** 文件，内容如下：

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

```
PATH=/bin:/sbin:/usr/bin:/usr/sbin
```

```
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
```

```
mkdir -p /var/lock
```

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

修改 **profile** 文件，内容如下：

```
# 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`
```

```
echo "Processing /etc/profile..."  
echo "Done"
```

```
export USER LOGNAME PS1 PATH
```

新建 **sysconfig** 文件夹，在里面新建 **HOSTNAME** 文件，内容为：TX2440A

拷贝库文件：

进入 **lib** 目录，拷贝交叉编译器的库文件到 **lib** 目录下

```
#cp -f -a  
/opt/toolchains/arm920t-eabi/arm-angstrom-linux-gnueabi/lib/*.so* ./
```

由于交叉编译器的库文件比较多，体积较大，所以需要裁剪一下库文件，只保留常用的库文件，具体的裁剪方法可以参考我们提供的文件系统的源码。

到这里，文件系统的框架已经做好，我们需要把文件系统制作成可以烧写进 Nandflash 中的 yaffs2 镜像文件。

使用 **mkyaffs2image** 工具，制作 yaffs2 镜像，这个工具是在 yaffs2 源码包中 **utils** 目录下的 **mkyaffs2image.c** 文件编译得到的，如果使用的是大页 NAND，需要修改源码。

可以使用我们提供的编译好的 **mkyaffs2image** 工具（位置：光盘资料/linux 平台工具/mkyaffs2image.tar.gz）

把解压出来的可执行文件复制到 **/usr/bin** 目录下，

```
#cp mkyaffs2image /usr/bin
```

制作文件系统镜像：

```
#mkyaffs2image root_2.6.31 root_2.6.31.bin
```

root_2.6.31.bin 就是 yaffs2 文件系统镜像，要先修改它的权限，否则该文件无法被复制：

```
#chmod 777 root_2.6.31.bin
```

把 **root_2.6.31.bin** 下载到 nandflash 中的文件系统分区中

如果文件系统能正常挂载，终端会打印出：

```
Please press Enter to activate this console. (按回车)
```

```
Processing /etc/profile...
```

```
Done
```

```
[root@TX2440A /]#
```


STEP 4: 完善文件系统

在/mnt 目录下创建以下文件夹，用来挂载 NFS，SD 卡和 U 盘

```
nfs sd udisk1 udisk2 udisk3 udisk4
```

设置开发板 IP 地址：在 init.d/rcS 文件中加入：

```
ifconfig lo 127.0.0.1
```

```
ifconfig eth0 192.168.1.10 netmask 255.255.255.0 up
```

```
route add default gw 192.168.1.1
```

这样在开机时，就自己设置了 IP 地址为:192.168.1.10

在/etc/profile 中加入：

```
alias ll='ls -l'
```

```
alias mntnfs='mount -t nfs -o nolock'
```

```
alias mntudl='mount /dev/sdal /mnt/udisk1'
```

```
alias umntudl='umount /mnt/udisk1'
```

alias 命令的作用是给一个命令取个别名，可以把较长的命令简化成简单易记的短命令，可以根据自己的需要添加这里的内容。

向文件系统加入更多的应用程序，可以参考我们提供的移植手册

嵌入式 WEB 服务器：参考《web 服务器 Boa 移植手册》

嵌入式音乐播放器：参考《madplay 音乐播放器移植手册》

嵌入式 FTP 服务器：参考《嵌入式 FTP 服务器(vsftpd)移植手册》

屏幕截图工具：参考《嵌入式 linux 截图工具移植》

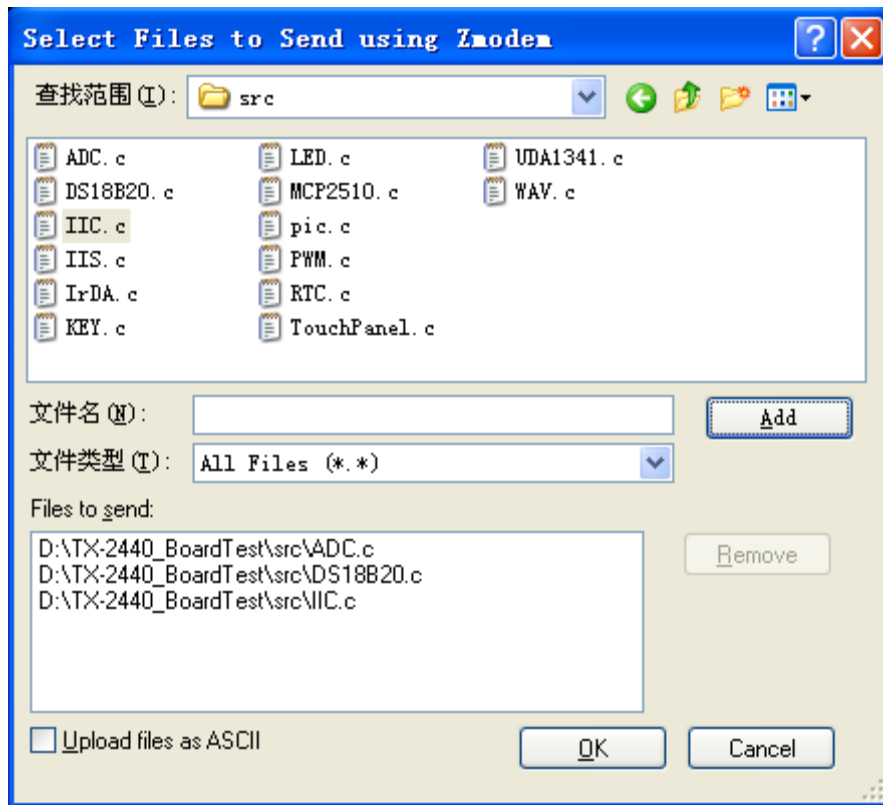
USB 摄像头监控：参考《基于 TX2440A 开发板的网络远程视频监控系统》

图形文件系统的制作，可以参考《Qttopia4.2.4 手机平台移植手册》

Linux 系统下应用程序和工具使用方法:

1. 串口下载文件 (PC 到开发板)

输入命令: **rz**, 会弹出一个对话框, 选择要下载的文件, 点击 Add, 加入到下面的框中 (可以添加多个文件), 点击 OK 进行下载



```
[root@TX2440A /]#
[root@TX2440A /]# rz
rz ready. Type "sz file ..." to your modem program

starting zmodem transfer. Press Ctrl+C to cancel.
Transferring ADC.c...
  100%    2 KB    2 KB/s 00:00:01    0 Errors
Transferring DS18B20.c...
  100%    5 KB    5 KB/s 00:00:01    0 Errors
Transferring IIC.c...
  100%   12 KB   12 KB/s 00:00:01    0 Errors

Šrz 3.48 01-27-98 finished.
[root@TX2440A /]# █
```


2. 串口上传文件（开发板到 PC）

方式 1:

输入命令: **sx 文件名**, 选择菜单栏中的 **Transfer->Receive Xmodem**, 打开接收文件对话框, 选择上传文件的路径, 输入要保存的文件名, 点 Receive, 即可上传。

比如, 我要上传开发板上的 IIC.c 这个文件到我的文档中

输入命令: **sx IIC.c**



```
[root@TX2440A /]# sx IIC.c
Sending IIC.c, 104 XMODEM blocks. Start your local XMODEM receive.
Starting xmodem transfer. Press Ctrl+C to cancel.
Transferring D:\Backup\我的文档\IIC.c...
    13312 bytes   1024 bytes/sec 00:00:13    0 Errors
sx 3.48 01-27-98 finished.
[root@TX2440A /]#
```

方式 2:

输入命令: **sz 文件名**, 就会把这个文件传送到 SecureCRT 的 Download 目录下,

默认路径是: **C:\Program Files\VanDyke Software\SecureCRT\download**

这种传输方式的速度较方式 1 快, 建议使用这种方式传输小文件

3. 播放音乐

输入命令: `madplay 歌曲名`, 比如: `madplay /home/xgc/beyond-HKTK.mp3`

4. 抓取 LCD 图像

输入命令: `gsnpa 要保存图片文件名 /dev/fb0`, 比如: `gsnap pic.jpg /dev/fb0`

5. 挂载 U 盘

插入 U 盘到任意一个 USB 口, 等待弹出信息, 查看 U 盘的设备名称 (第一次插入, 设备名默认为 `sda1`), 输入命令: `mount /dev/sda1 /mnt/udisk1`, 然后到 `/mnt/udisk1` 目录下查看 U 盘内容。取消挂载用: `umount /mnt/udisk1`

也可以使用命令 `mntud1` 来挂载 U 盘, 也可以在 `profile` 文件中使用 `alias` 命令来自定义挂载命令。

注意: 部分 U 盘的设备名不是 `sda1`, 可能是 `sda` 或是没有设备名, 这样使用上面的挂载命令就不能成功挂载 U 盘, 这就需要查看 `dev` 目录下 U 盘的设备名来改写挂载 U 盘命令

6. 挂载 SD 卡

插入 SD 卡, 等待弹出信息, 查看 SD 卡设备名称 (默认为 `mmcblk0p1`), 输入命令: `mount /dev/mmcblk0p1 /mnt/sd`, 进入 `/mnt/sd` 目录查看 SD 卡的内容。

也可以使用命令 `mntsd` 来挂载 SD 卡, 取消挂载用: `umount /mnt/sd`

注意: 部分 SD 卡的设备名不是 `mmcblk0p1`, 这个跟 SD 的分区有关, 一般的 SD 卡默认有一个分区 `p1`, 它的第一个分区就是 `mmcblk0p1`, 这个命令就是挂载 SD 卡的第一个分区; 有的 SD 卡没有分区, 那么它的设备名就是 `mmcblk0`, 这就需要根据 SD 卡的设备名来改写挂载 SD 卡命令。

还可以对 SD 卡进行分区, 使用 `fdisk` 命令, 具体使用方式请查看 `fdisk` 的帮助, 对 SD 卡分完区后, 它的设备名就变为 `mmcblk0p1` 了

7. 使用 telnet 登录开发板

打开 windows 下的命令行窗口, 输入: `telnet 192.168.1.10` ← 开发板的 IP 地址
输入用户名 (`root`) 和密码 (`111111`), 即可远程登录到开发板, 输入: `exit`, 退出 telnet 登录

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>telnet 192.168.1.10
```

```
Telnet 192.168.1.10

TX2440A login: root
Password:
Processing /etc/profile...
Done
[root@TX2440A /root]# cd ..
[root@TX2440A /]# ls
ADC.c      dev        linuxrc    proc       tmp
DS18B20.c  etc        lost+found root        usr
IIC.c      home       mnt       /sbin       var
bin        lib        opt        sys
[root@TX2440A /]#
```

8. 重启、关机命令

重启: **reboot**

关机: **halt**

9. 关闭、开启 LCD 背光

关闭背光: `echo 0 > /dev/TX2440-backlight`

开启背光: `echo 1 > /dev/TX2440-backlight`

10. 查看、修改 IP 地址

输入命令: `ifconfig eth0`, 查看开发板 IP 地址

如果要修改 IP 地址, 就在这个命令后面加上要修改的 IP 地址

11. 串口 1 测试方法

输入命令: `getty /dev/s3c2410_serial1 115200`

命令说明: `getty` 是改变控制台命令, `/dev/s3c2410_serial1` 是串口 1 的设备文件名, 115200 是串口的波特率

把串口线接到串口 1 上, 按回车键

终端上提示输入用户名和密码

TX2440A login: `root`

Password: `111111` ← (注意密码不回显)

这样控制台就交给了串口 1, 你可以在串口 1 的命令提示符下输入命令来操作开发板

如果要退出串口 1, 输入命令: `exit`, 把串口线接到串口 0 上, 这样控制台又交给了串口 0

```
[root@TX2440A /]# getty /dev/s3c2410_serial1 115200
```

```
TX2440A login: root
```

```
Password:
```

```
Processing /etc/profile...
```

```
Done
```

```
[root@TX2440A /root]# █
```

系统下的驱动测试程序使用方法:

1. LED

命令格式: `led 0|1` ←控制 LED 全亮或全灭

`Led (1-4) 0|1` ←控制单个 LED 亮或灭

`led 1` 4 个 LED 全亮

led 0 4 个 LED 全灭
led 2 0 第二个 LED 灭
led 3 1 第三个 LED 亮

2. KEY

命令格式: key,

执行 key, 按下按键, 在终端下打印出按下的键值

3. ADC

命令格式: adc 0 | 1 ←选择 AD 通道

adc 0 查看 0 通道 AD 转换值

adc 1 查看 1 通道 AD 转换值

4. DS18B20

命令格式: ds18b20

执行 ds18b20, 在终端上打印出温度值

5. BEEP

命令格式: buzzer

执行 buzzer, 输入蜂鸣器发声频率, 输入 0 退出程序

6. CAMERA

命令格式: camera

在开机之前, 插入 COMS 摄像头模块

执行 camera, 在 LCD 上显示出摄像头捕捉的画面

7. GPS (20100612 更新的内核中包含有 GPS 模块驱动)

命令格式: gps

连接 GPS 模块 (参考 GPS 模块手册, 用 5P 短线连接开发板的 GPS 接口)

执行 gps, 在终端上打印出 GPS 接收到的原始数据