

# Android on mini2440 简易编译安装指南

开发平台: Fedora 9

交叉编译器: arm-linux-gcc-4.3.2 with EABI

目标平台: 128M-1G 版本的 mini2440+统宝 3.5”LCD(mini2440T35)

日期: 2009-8-24

作者: 友善之臂

## Android on mini2440 简要说明:

Android 是 Google 公司开发的开放性手机操作系统平台, 它主要基于 Java 构建各种应用程序。目前 Android 已经被移植到多种嵌入式设备和平台上运行, 爱好者众多, 甚为流行。

有很多 mini2440 用户也在讨论和尝试移植 Android, 其中 shinelk 网友向我们展示了他的一些移植成果, 我们觉得这或许对于其他用户/玩家有些参考意义, 因此将其买下, 并完全开放出来供各位网友下载试用。

为了提供给各位用户更广阔的交流空间, 我们向“ARM9 之家”论坛申请了一个专门讨论 Android on mini2440 的版块, 并特意邀请 shinelk 担任版主(为期半年)。

注意: 本公司并不提供基于该版本 android 的任何技术支持和服务咨询。shinelk 本身也是一个普通网友, 仅在空余时间与大家交流和讨论他移植的 android 系统。

友善之臂保留以上文字描述的解释权。

广州友善之臂计算机科技有限公司(简称“友善之臂”)

2009-8-24

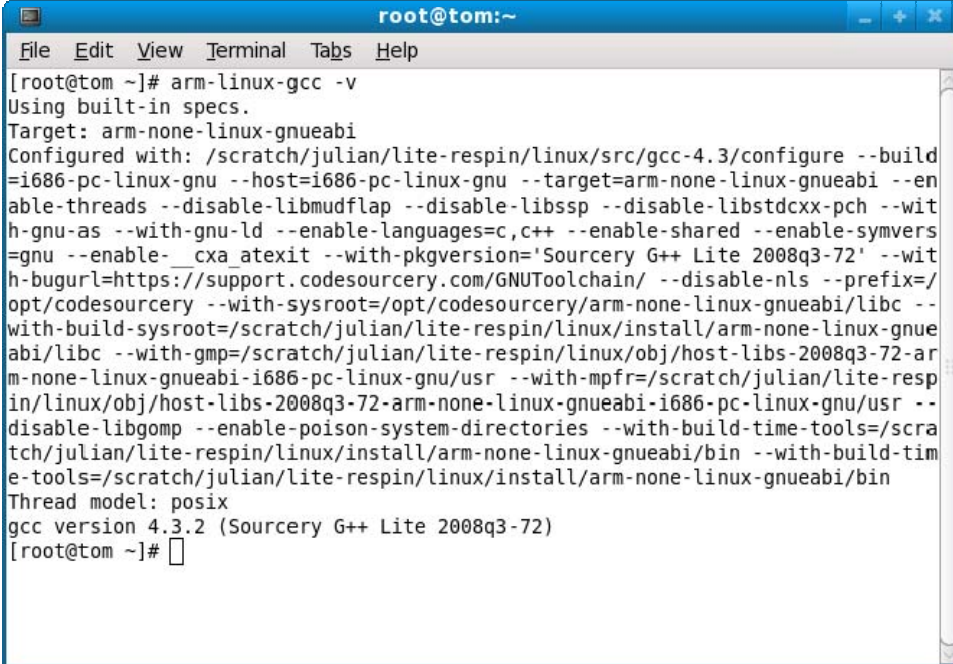
## 为何选用 128M mini2440?

因为 Android 基本文件系统内容较多, 无法直接放入容量较小的开发板运行使用。我们特意选择 128M mini2440 作为 android 基本的开发平台。

1. [检查交叉编译器版本](#)
2. [解压android内核源代码和文件系统目录](#)
3. [配置和编译android内核](#)
4. [制作yaffs2 文件系统映像](#)
5. [烧写到mini2440 开发板运行](#)
6. [各个文件下载地址](#)

## 1. 检查交叉编译器版本

编译此处提供的 android 内核，需要使用符合 EABI 标准的 arm-linux-gcc-4.3.2 交叉编译器，在命令行输入 arm-linux-gcc -v 可以看到版本信息如下：



```
root@tom:~  
File Edit View Terminal Tabs Help  
[root@tom ~]# arm-linux-gcc -v  
Using built-in specs.  
Target: arm-none-linux-gnueabi  
Configured with: /scratch/julian/lite-respin/linux/src/gcc-4.3/configure --build=i686-pc-linux-gnu --host=i686-pc-linux-gnu --target=arm-none-linux-gnueabi --enable-threads --disable-libmudflap --disable-libssp --disable-libstdcxx-pch --with-gnu-as --with-gnu-ld --enable-languages=c,c++ --enable-shared --enable-symvers=gnu --enable-__cxa_atexit --with-pkgversion='Sourcery G++ Lite 2008q3-72' --with-bugurl=https://support.codesourcery.com/GNUToolchain/ --disable-nls --prefix=/opt/codesourcery --with-sysroot=/opt/codesourcery/arm-none-linux-gnueabi/libc --with-build-sysroot=/scratch/julian/lite-respin/linux/install/arm-none-linux-gnueabi/libc --with-gmp=/scratch/julian/lite-respin/linux/obj/host-libs-2008q3-72-arm-none-linux-gnueabi-i686-pc-linux-gnu/usr --with-mpfr=/scratch/julian/lite-respin/linux/obj/host-libs-2008q3-72-arm-none-linux-gnueabi-i686-pc-linux-gnu/usr --disable-libgomp --enable-poison-system-directories --with-build-time-tools=/scratch/julian/lite-respin/linux/install/arm-none-linux-gnueabi/bin --with-build-time-tools=/scratch/julian/lite-respin/linux/install/arm-none-linux-gnueabi/bin  
Thread model: posix  
gcc version 4.3.2 (Sourcery G++ Lite 2008q3-72)  
[root@tom ~]#
```

## 2. 解压 android 内核源代码和文件系统目录

(1) 建立工作目录：/opt/FriendlyARM/android

(2) 进入此目录并执行解压命令：

```
#tar xvzf android-kernel-20090825.tar.gz
```

```
#tar xvzf android-fs-20090825.tar.gz
```

执行完毕，如图所示：

```
root@tom:/opt/FriendlyARM/android
File Edit View Terminal Tabs Help
kernel/drivers/clocksource/acpi_pm.c
kernel/drivers/clocksource/cyclone.c
kernel/drivers/clocksource/tcb_clksrc.c
[root@tom android]# cd kernel/
[root@tom kernel]# ls
arch          crypto        include      lib          README      sound
block        Documentation init         MAINTAINERS REPORTING-BUGS usr
config_mini2440 drivers      ipc         Makefile    samples     virt
COPYING      firmware     Kbuild      mm          scripts
CREDITS      fs          kernel      net         security
[root@tom kernel]# ls -a
.          COPYING      firmware     Kbuild      mm          scripts
..         CREDITS      fs          kernel      net         security
arch       crypto       include     lib         README      sound
block     Documentation init        MAINTAINERS REPORTING-BUGS usr
config_mini2440 drivers    ipc         Makefile    samples     virt
[root@tom kernel]# cd ../fs
[root@tom fs]# ls
bin  default.prop  init.goldfish.rc  lib  lost+found  proc  sys  var
data dev        init.rc          linuxrc  myDev      sbin  system
[root@tom fs]# cd ..
[root@tom android]# ls
fs  kernel
[root@tom android]#
```

### 3. 配置和编译 android 内核

解压出来的内核中已经包含了一个缺省的内核配置文件: config\_mini2440, 运行命令:

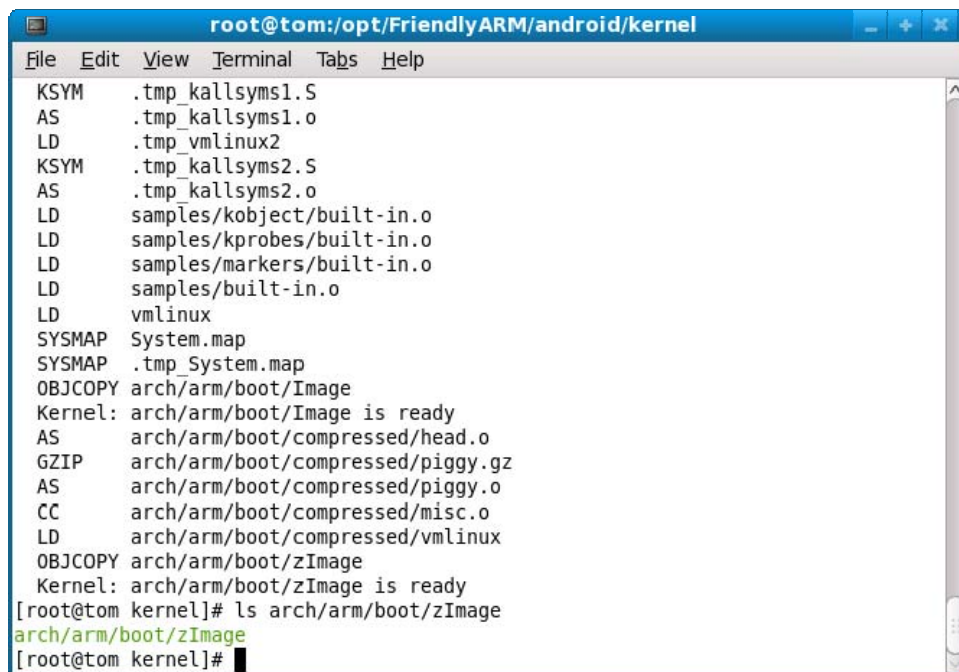
**#cp config\_mini2440 .config** (注意 config 前面有个“.”)

然后运行 make menuconfig, 不要做任何设置, 保存退出, 如图:

```
root@tom:/opt/FriendlyARM/android/kernel
File Edit View Terminal Tabs Help
..          CREDITS      fs          kernel      net          secur:
arch        crypto       include     lib         README      sound
block      Documentation init        MAINTAINERS REPORTING-BUGS usr
config_mini2440 drivers    ipc         Makefile    samples     virt
[root@tom kernel]# cd ../fs
[root@tom fs]# ls
bin  default.prop  init.goldfish.rc  lib  lost+found  proc  sys  var
data dev        init.rc          linuxrc  myDev      sbin  system
[root@tom fs]# cd ..
[root@tom android]# ls
fs  kernel
[root@tom android]# cd kernel/
[root@tom kernel]# ls
arch          crypto        include      lib          README      sound
block        Documentation init         MAINTAINERS REPORTING-BUGS usr
config_mini2440 drivers      ipc         Makefile    samples     virt
COPYING      firmware     Kbuild      mm          scripts
CREDITS      fs          kernel      net         security
[root@tom kernel]# cp config_mini2440 .config
[root@tom kernel]# make menuconfig
HOSTCC scripts/basic/fixdep
HOSTCC scripts/basic/docproc
HOSTCC scripts/kconfig/conf.o
```

再执行命令: **make zImage** 开始编译内核, 执行完毕, 如图所示, 将在 arch/arm/boot

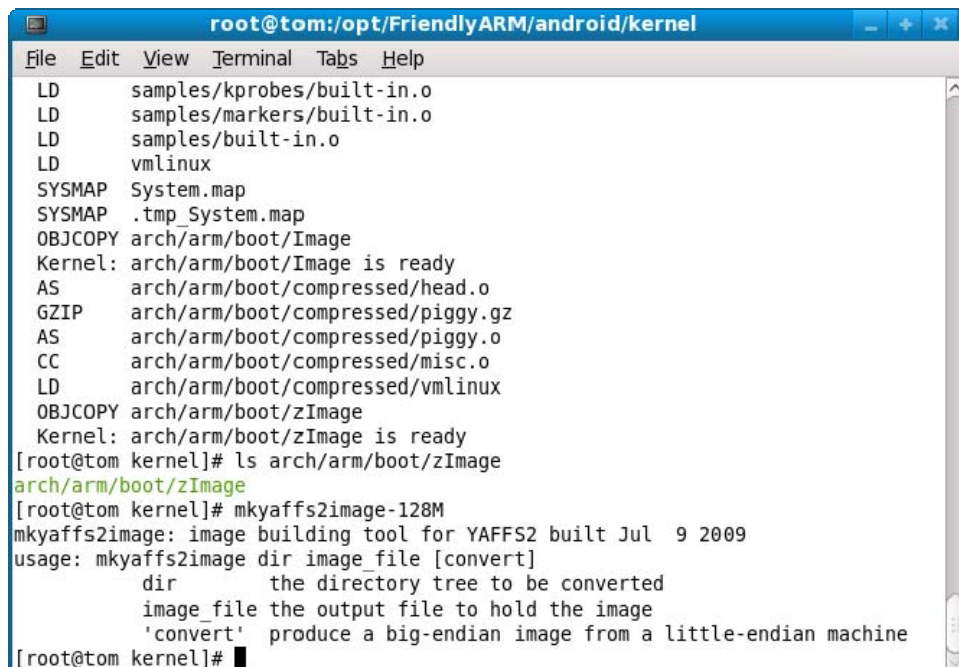
目录下生产 zImage 文件，如图：



```
root@tom:/opt/FriendlyARM/android/kernel
File Edit View Terminal Tabs Help
KSYM .tmp_kallsyms1.S
AS .tmp_kallsyms1.o
LD .tmp_vmlinux2
KSYM .tmp_kallsyms2.S
AS .tmp_kallsyms2.o
LD samples/kobject/built-in.o
LD samples/kprobes/built-in.o
LD samples/markers/built-in.o
LD samples/built-in.o
LD vmlinux
SYSMAP System.map
SYSMAP .tmp_System.map
OBJCOPY arch/arm/boot/Image
Kernel: arch/arm/boot/Image is ready
AS arch/arm/boot/compressed/head.o
GZIP arch/arm/boot/compressed/piggy.gz
AS arch/arm/boot/compressed/piggy.o
CC arch/arm/boot/compressed/misc.o
LD arch/arm/boot/compressed/vmlinux
OBJCOPY arch/arm/boot/zImage
Kernel: arch/arm/boot/zImage is ready
[root@tom kernel]# ls arch/arm/boot/zImage
arch/arm/boot/zImage
[root@tom kernel]#
```

#### 4. 制作 yaffs2 文件系统映象

要把目标文件系统目录 fs 制作成可以烧写的 yaffs2 文件系统映象文件，需要使用 mkyaffs2image-128M 功能，在命令行输入：mkyaffs2image-128M 应该出现如下信息：



```
root@tom:/opt/FriendlyARM/android/kernel
File Edit View Terminal Tabs Help
LD samples/kprobes/built-in.o
LD samples/markers/built-in.o
LD samples/built-in.o
LD vmlinux
SYSMAP System.map
SYSMAP .tmp_System.map
OBJCOPY arch/arm/boot/Image
Kernel: arch/arm/boot/Image is ready
AS arch/arm/boot/compressed/head.o
GZIP arch/arm/boot/compressed/piggy.gz
AS arch/arm/boot/compressed/piggy.o
CC arch/arm/boot/compressed/misc.o
LD arch/arm/boot/compressed/vmlinux
OBJCOPY arch/arm/boot/zImage
Kernel: arch/arm/boot/zImage is ready
[root@tom kernel]# ls arch/arm/boot/zImage
arch/arm/boot/zImage
[root@tom kernel]# mkyaffs2image-128M
mkyaffs2image: image building tool for YAFFS2 built Jul  9 2009
usage: mkyaffs2image dir image_file [convert]
        dir          the directory tree to be converted
        image_file   the output file to hold the image
        'convert'    produce a big-endian image from a little-endian machine
[root@tom kernel]#
```

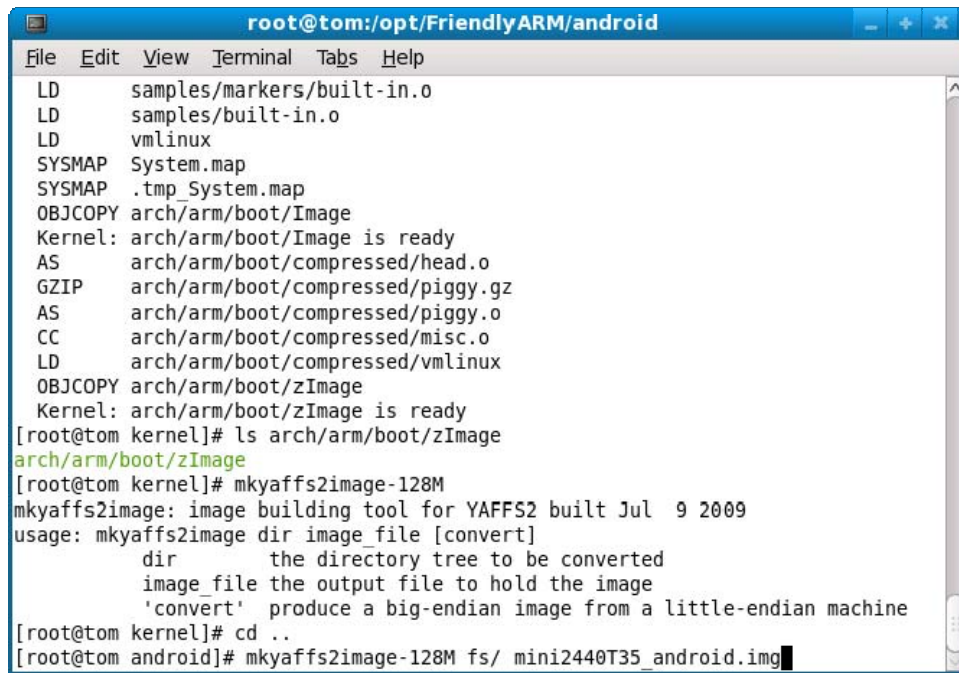
在/opt/FriendlyARM/android 目录下执行：

**#mkyaffs2image-128M fs mini2440T35\_android.img**

将会把 fs 目录制作成可烧写映象文件：mini2440T35\_android.img

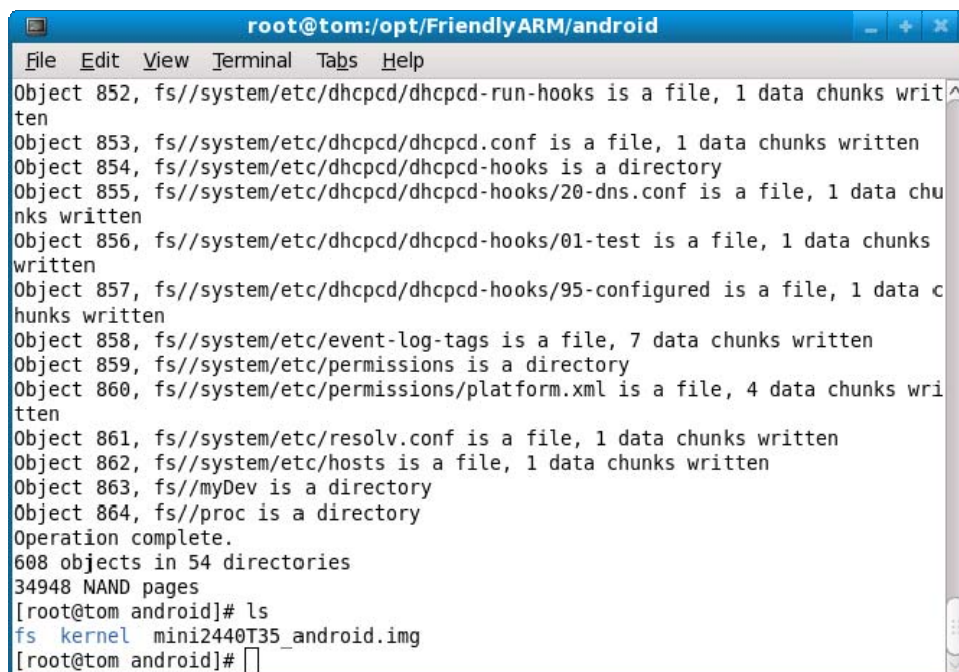


如图：



```
root@tom:/opt/FriendlyARM/android
File Edit View Terminal Tabs Help
LD      samples/markers/built-in.o
LD      samples/built-in.o
LD      vmlinux
SYSMAP  System.map
SYSMAP  .tmp_System.map
OBJCOPY arch/arm/boot/Image
Kernel: arch/arm/boot/Image is ready
AS      arch/arm/boot/compressed/head.o
GZIP    arch/arm/boot/compressed/piggy.gz
AS      arch/arm/boot/compressed/piggy.o
CC      arch/arm/boot/compressed/misc.o
LD      arch/arm/boot/compressed/vmlinux
OBJCOPY arch/arm/boot/zImage
Kernel: arch/arm/boot/zImage is ready
[root@tom kernel]# ls arch/arm/boot/zImage
arch/arm/boot/zImage
[root@tom kernel]# mkyaffs2image-128M
mkyaffs2image: image building tool for YAFFS2 built Jul  9 2009
usage: mkyaffs2image dir image_file [convert]
       dir          the directory tree to be converted
       image_file   the output file to hold the image
       'convert'    produce a big-endian image from a little-endian machine
[root@tom kernel]# cd ..
[root@tom android]# mkyaffs2image-128M fs/ mini2440T35_android.img
```

执行完毕结果如图：



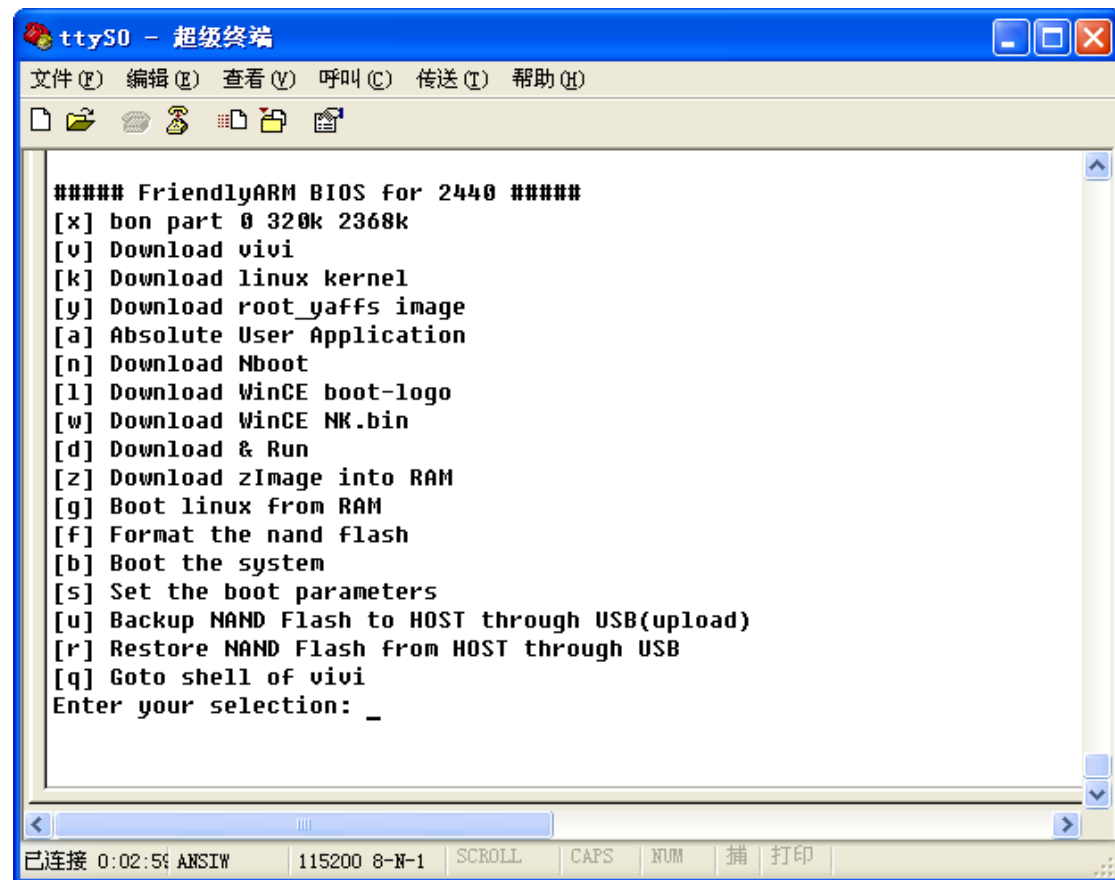
```
root@tom:/opt/FriendlyARM/android
File Edit View Terminal Tabs Help
Object 852, fs//system/etc/dhcpd/dhcpd-run-hooks is a file, 1 data chunks written
Object 853, fs//system/etc/dhcpd/dhcpd.conf is a file, 1 data chunks written
Object 854, fs//system/etc/dhcpd/dhcpd-hooks is a directory
Object 855, fs//system/etc/dhcpd/dhcpd-hooks/20-dns.conf is a file, 1 data chunks written
Object 856, fs//system/etc/dhcpd/dhcpd-hooks/01-test is a file, 1 data chunks written
Object 857, fs//system/etc/dhcpd/dhcpd-hooks/95-configured is a file, 1 data chunks written
Object 858, fs//system/etc/event-log-tags is a file, 7 data chunks written
Object 859, fs//system/etc/permissions is a directory
Object 860, fs//system/etc/permissions/platform.xml is a file, 4 data chunks written
Object 861, fs//system/etc/resolv.conf is a file, 1 data chunks written
Object 862, fs//system/etc/hosts is a file, 1 data chunks written
Object 863, fs//myDev is a directory
Object 864, fs//proc is a directory
Operation complete.
608 objects in 54 directories
34948 NAND pages
[root@tom android]# ls
fs kernel mini2440T35_android.img
[root@tom android]#
```

## 5. 烧写到 mini2440 开发板运行

以上我们分别制作生产了内核文件：**zImage**，和文件系统映像文件：**mini2440T35\_android.img**，下面我们通过开发板自带的 **supervivi** 把它们烧写到开发板并运行。

**注意：**安装 **USB 驱动和串口终端设置**，以及 **USB 下载烧写说明**，请见 **mini2440 用户手册**。

首先，把开发板的 S2 开关设置到 NOR 一次，开机启动进入 BIOS 模式(也就是 supervivi 模式)，128M 版本的 mini2440 的 BIOS 输出如图：



```
##### FriendlyARM BIOS for 2440 #####
[x] bon part 0 320k 2368k
[v] Download vivi
[k] Download linux kernel
[y] Download root_yaffs image
[a] Absolute User Application
[n] Download Nboot
[l] Download WinCE boot-logo
[w] Download WinCE NK.bin
[d] Download & Run
[z] Download zImage into RAM
[g] Boot linux from RAM
[f] Format the nand flash
[b] Boot the system
[s] Set the boot parameters
[u] Backup NAND Flash to HOST through USB(upload)
[r] Restore NAND Flash from HOST through USB
[q] Goto shell of vivi
Enter your selection: _
```

在此菜单下：

- (1) 先输入“x”格式化系统
- (2) 然后再选择“v”开始下载 bootloader，可以选 vboot.bin，也可以选择 supervivi-128M
- (3) 再选择“k”开始下载 android 内核：zImage
- (4) 最后选择“y”开始下载 mini2440T35\_android.img 文件系统文件

烧写完毕，可以使用“b”命令启动系统，也可以把 S2 开关拨到 Nand Flash 一侧，复位启动系统。

第一次启动系统时，会出现校正界面，依次点“十”字型的中间位置，直到校正接触，稍等片刻，就可以看到“大钟”界面了，如图：



此版本的 android 支持以下特性:

1. 比较准确的触摸屏
2. SD 卡自动挂载（开机之前必须先插入 SD 卡）
3. 按键驱动

板子上的按键对应功能关系		
K6	K4	K5
返回	菜单	右键
K3	K2	K1
音量减	中间按键	音量加

4. DM9000 网卡驱动
5. busybox 采用 1.13.3
6. 音频输入与输出：可播放 mp3,并可录音
7. 命令行截图工具：snapshot

## 6. 各个文件下载地址

(1)二进制映像文件(包含 supervivi-128M, zImage, mini2440T35\_android.img)

<http://www.arm123.com.cn/android/android-images-shinelk-20090825.tgz>

(2)文件系统包

<http://www.arm123.com.cn/android/android-fs-20090825.tgz>

(3)内核源代码包

<http://www.arm123.com.cn/android/android-kernel-20090825.tgz>

(4) 符合 EABI 标准的交叉编译器 arm-linux-gcc-4.3.2

<http://www.arm123.com.cn/linux/arm-linux-gcc-4.3.2.tgz>