博创科技 UP-CUP2440 经典平台
快速开始手册（LINUX 2.6.24）

V 1.0

北京博创兴业科技有限公司

2011.3

博创科技UP-CUP S2440经典
LINUX系统快速开始手册v1.0

1.目的

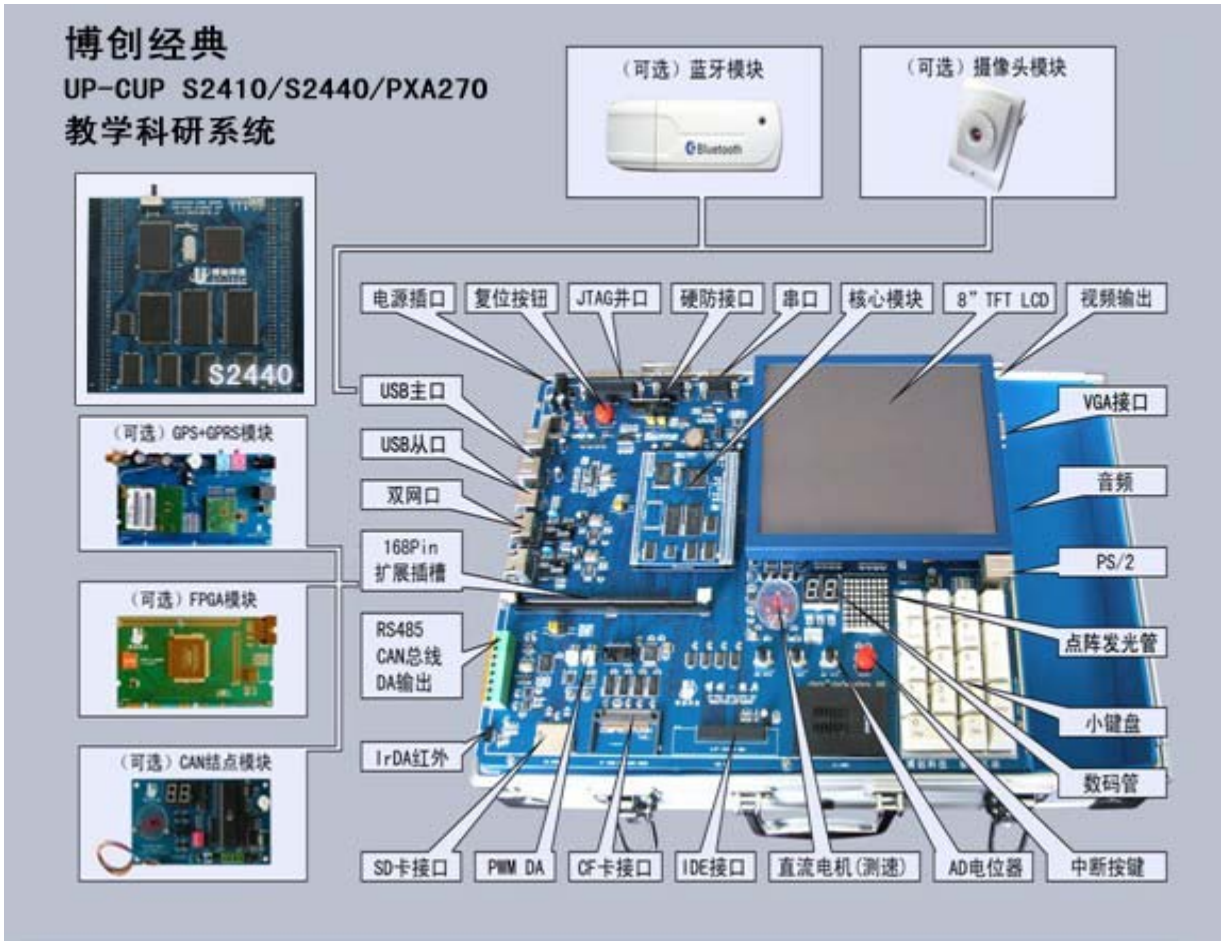
帮助初学者了解 UP-CUP2440 博创科技嵌入式教学科研平台的硬件资源，学习演示linux操作系统移植的demo程序，恢复出厂状态的方法等等.

2.对象范围

该手册适用于该教学科研平台的初次使用者，既适合于计算机，软件，电子，自动化，机电一体化等开设嵌入式专业课程的教师和学生，又适合于从事PMP、PDA、智能手机研发的硬件和软件工程师。

3.平台介绍

UP-CUP2440是博创科技推出的国内最强配置的嵌入式开发平台，可以满足不同院校的ARM9嵌入式教学要求。同时该平台可与博创PXA270 CPU和S2410核心板兼容。(S2410，PXA270和S2440三套核心板可以互换)



一、 概述

本手册是配合 UP-CUP 经典 2440 教学科研平台软件系统的烧写手册，其内容包括了软件系统中 bootloder、内核、根文件系统以及应用程序的烧写方法。可以帮助使用者快速掌握系统软件的烧写方法及了解系统的软件启动流程。此外对软件系统的更新或更改，都可以参照此文档进行。

注：本手册为 linux2.6.24 系统内核操作手册

二、资源

UP-CUP S2440 核心模块资源

* 基于 ARM9 架构的嵌入式芯片 S3C2440，主频 405MHz
* 64MB SDRAM
* 256MB Nand Flash

UP-CUP S2440 主板资源

* 8 寸 640*480TFT 真彩 LCD
* 触摸屏
* 4 个主 USB 口、1 个从 USB 口
* 1 个 UP-LINK 集成调试接口（并口）、20 针 JTAG 口
* 一个 100M 网卡
* 两个串口、1 个 RS485 串口

* 一个 VGA 接口
* CAN 总线接口
* 红外通信收发器
* 8 通道 10 位 AD 转换模块

* 10 位 DA 转换模块
* SD/MMC 接口
* IDE 硬盘接口
* CF 卡接口
* IC 卡接口
* 直流电机、带有红外线测速电路
* 2 个用户自定义 LED 数码管、1 个 8x8 点阵发光管、3 个 LED 灯
* 17 键键盘、一个中断按键
* PS2 鼠标、键盘接口

* 高性能立体声音频模块，支持放音、录音
* 麦克风接入
* 一个 168Pin 的扩展插座，硬件可无限扩展

* 可提供配套的 GPRS/GPS、FPGA、CAN 单片机、USB2.0 等扩展模块

UP-CUP S2440 软件资源：

*提供完整的 Linux 操作系统移植

*bootloader：u-boot

*操作系统： linux2.6.24

*驱动程序：提供所有板级设备的驱动程序

三、 如何恢复出厂设置

如果系统出现任何软件上的异常导致无法正常启动和运行，请参考如下步骤进行恢复系统

在 windows xp 下进行 Linux 系统烧写即恢复到出厂状态时，需要的文件在光盘中的 Linux\img 目录和 flash-uboot 目录下提供。烧写 S2440 linux 操作系统包括烧写 u-boot,kernel,root 三个步骤，除此我们还要烧写应用程序，这四个文件分别为：

u-boot.bin ---- linux 操作系统启动的 bootloader;

uImage ---- linux 操作系统内核;

root.cramfs ---- 根文件系统;

yaffs.tar.bz2 ---- 应用程序压缩包。

四、并口驱动安装

首先，将并口线连接到我们的 PC 机上，另一端连接到我们提供的 JTAG，再将 JTAG 连接到板子上的 14 针接口上。
然后，把附带光盘中 LINUX 部分下的 tools 文件夹中的整个 GIVEIO 文件夹拷贝到 C:\WINDOWS 下，并把该目录下的 giveio.sys 文件拷贝到 C:\WINDOWS\system32\drivers 下。
接着，在控制面板里，选添加硬件如图 1。

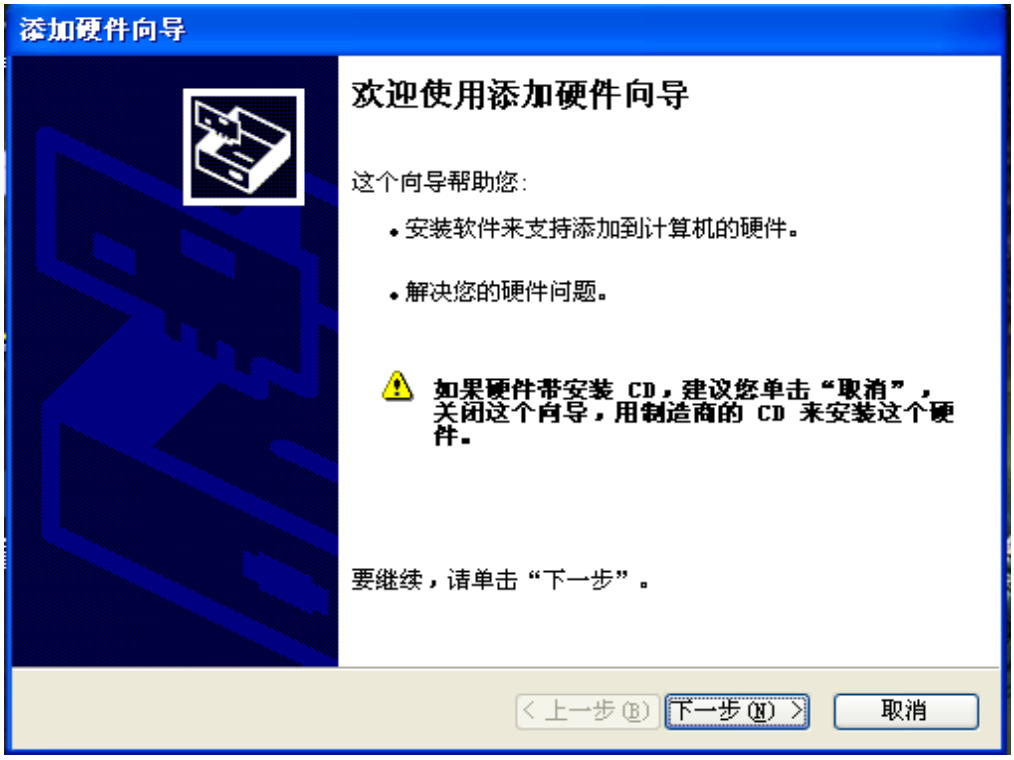


图 1

点击下一步，选“是，我已经连接了此硬件”如图 2。

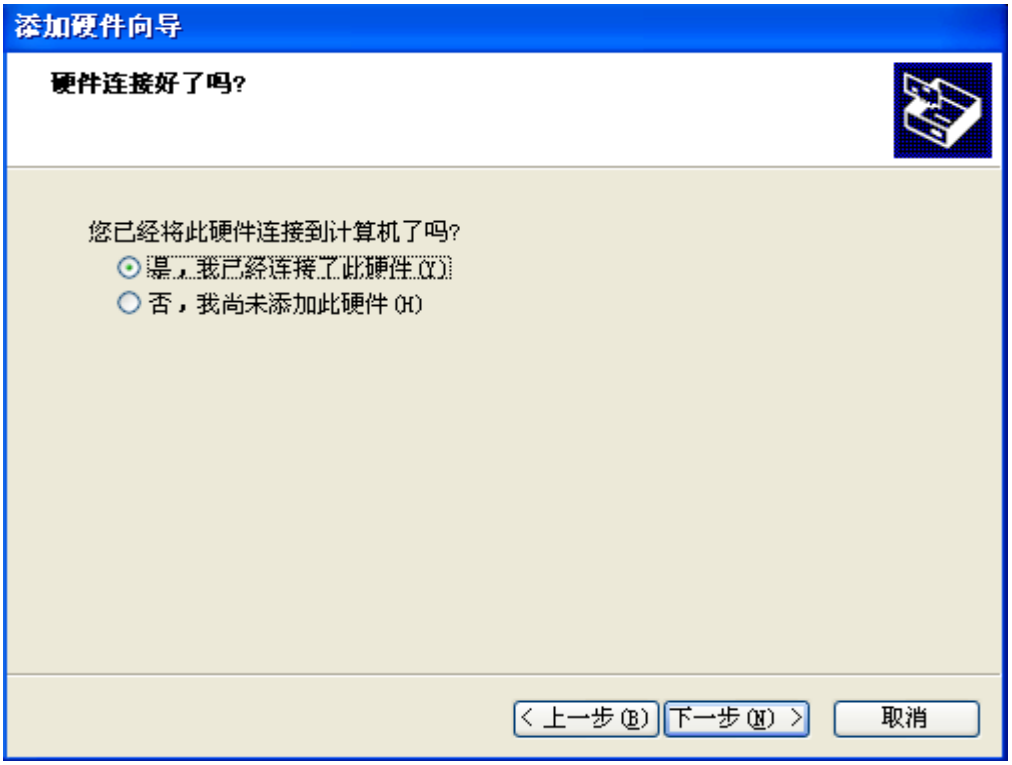


图 2

点击下一步，选中“添加新的硬件设备”，如图 3。

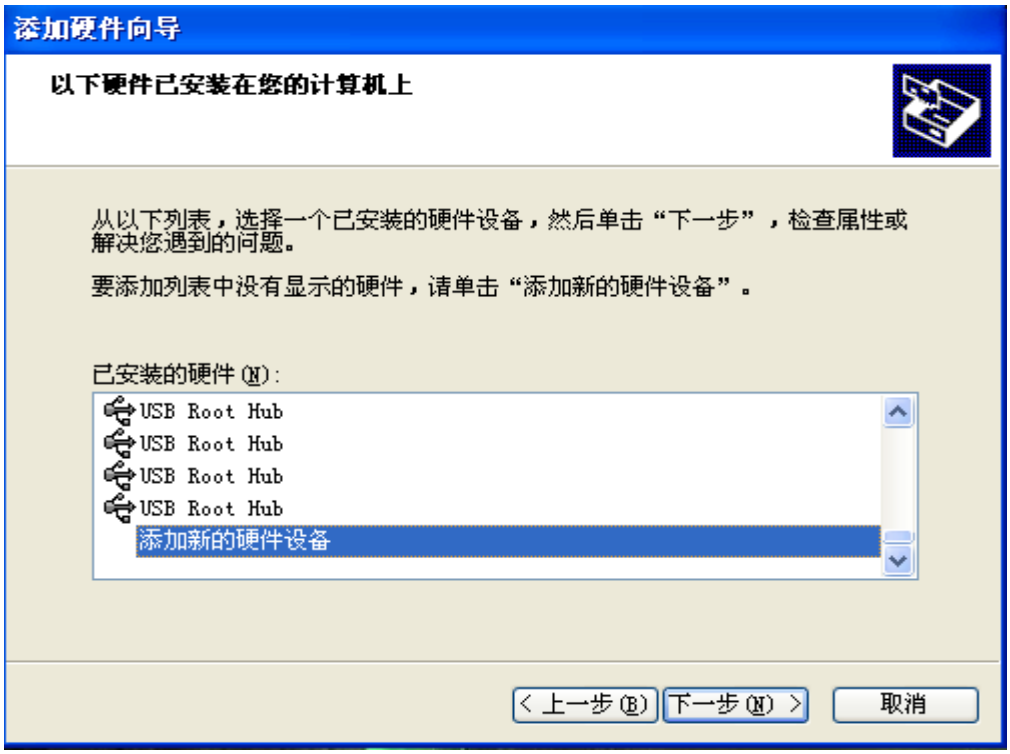


图 3

点击下一步，选中安装我手动从列表选择的硬件如图 4。

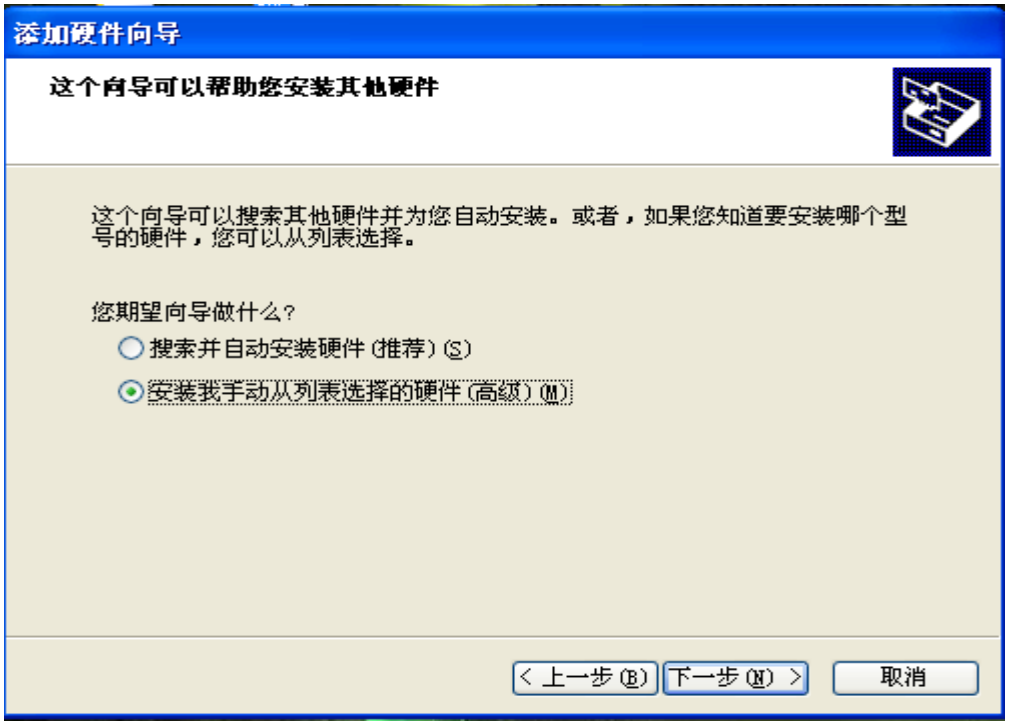


图 4

点击下一步，选择显示所有设备如图 5。

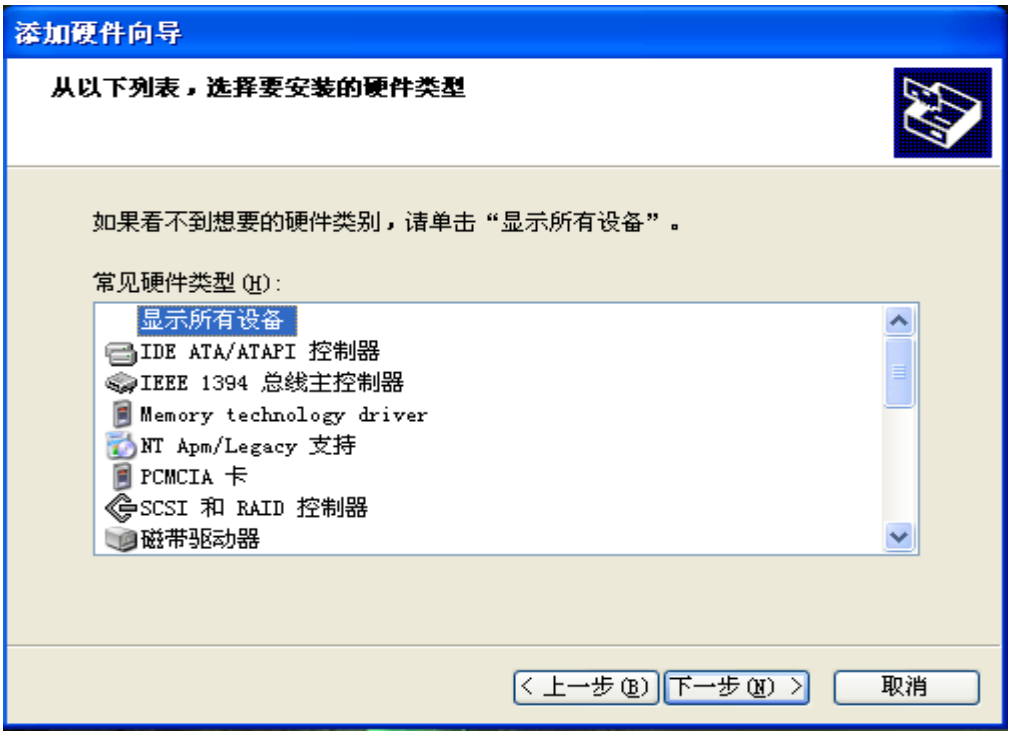


图 5

选择从磁盘安装如图 6。

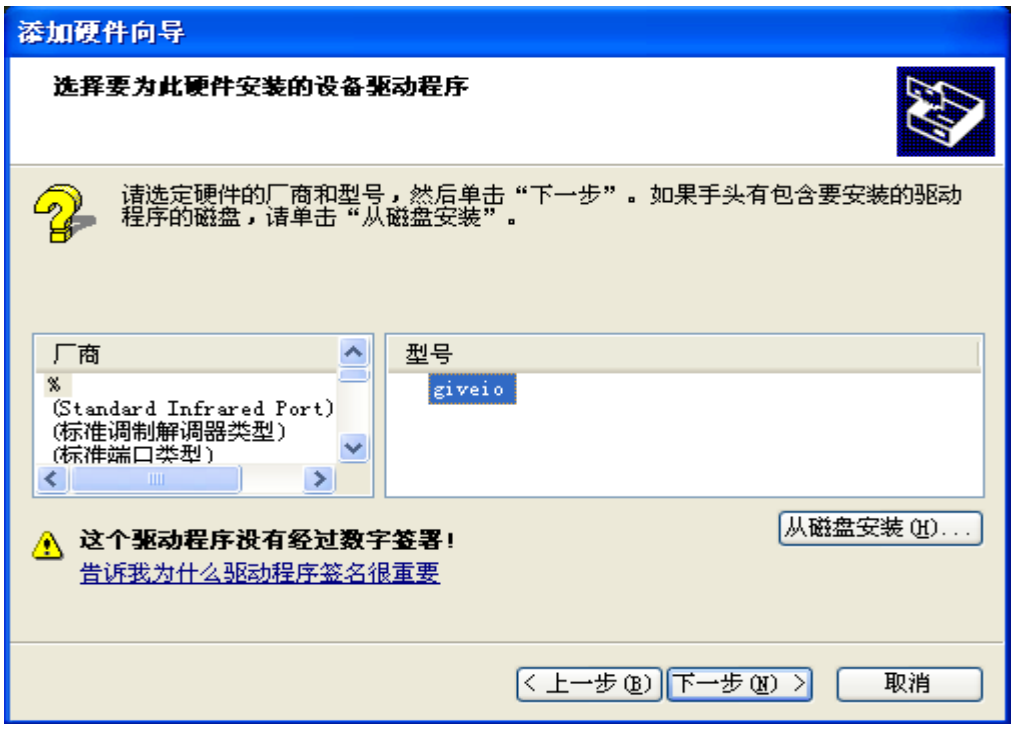


图 6

选择浏览，如图 7。

图 7

指定驱动为 C:\WINDOWS\GIVEIO\giveio.inf 文件，点击确定，如图 8 和图 9。

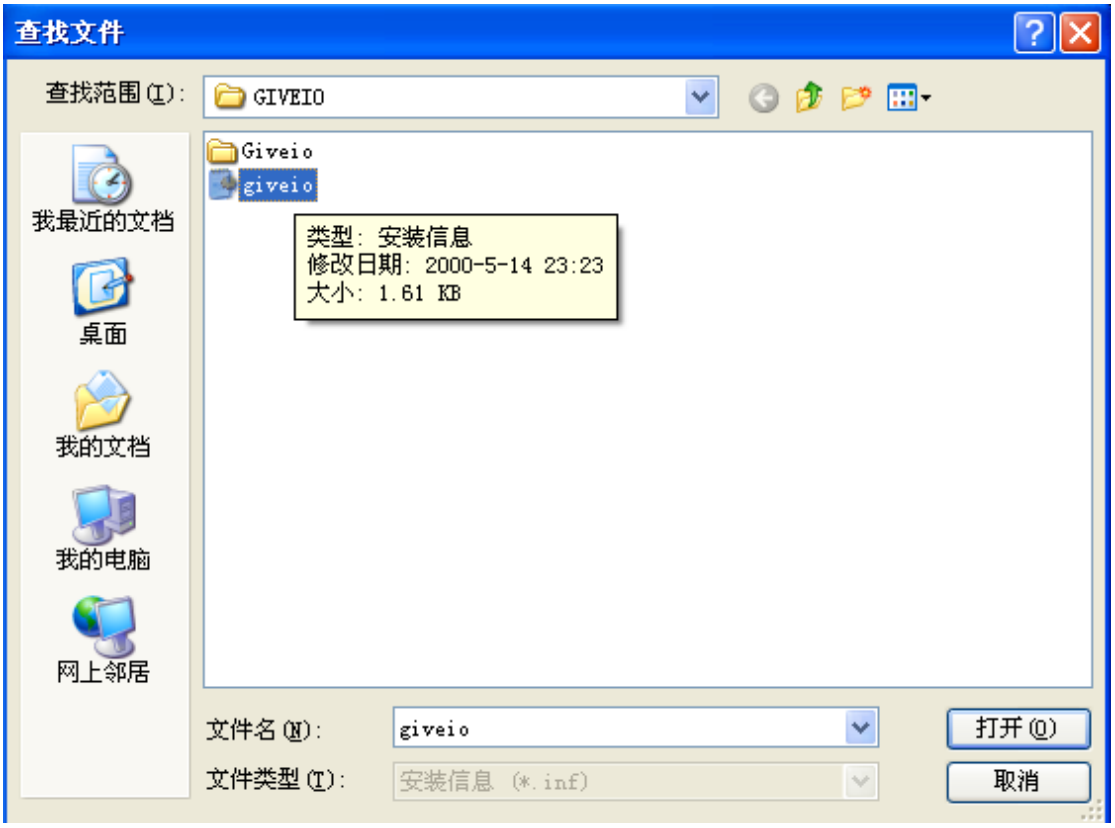


图 8

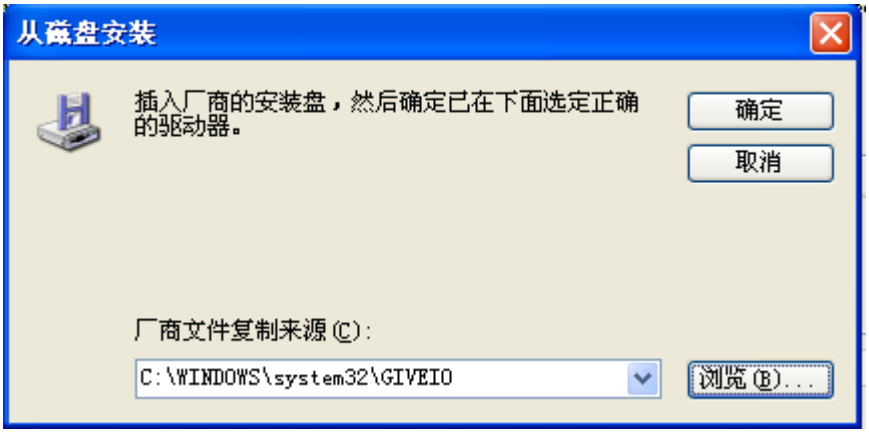


图 9

继续点击下一步，完成安装，如图 10 和图 11。

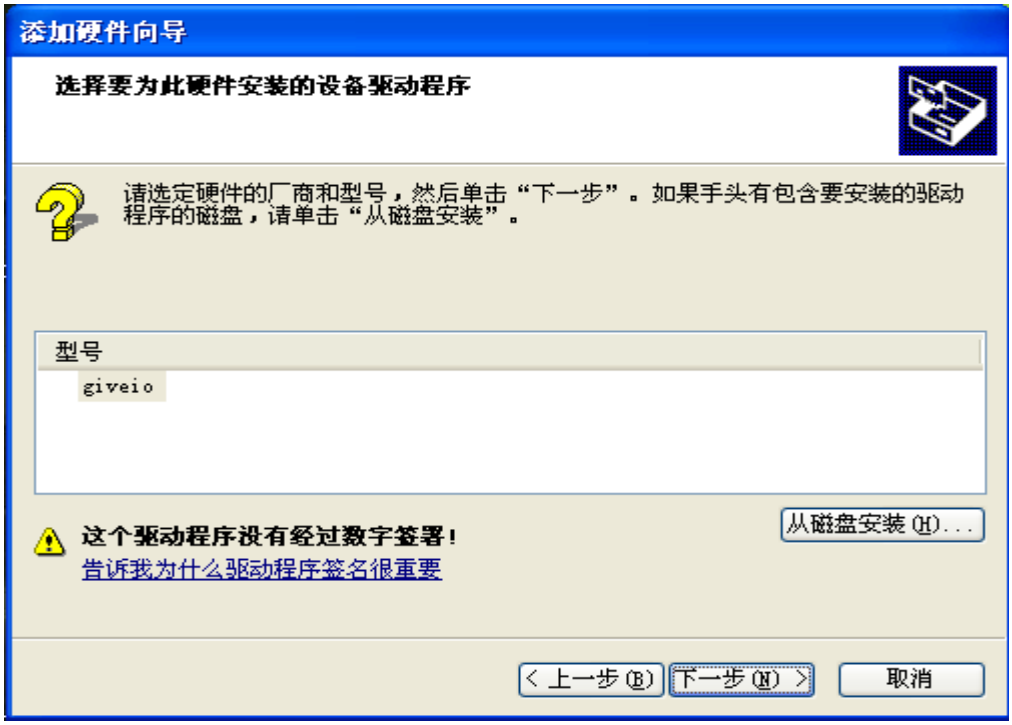

添加硬件向导
选择要为此硬件安装的设备驱动程序
请选定硬件的厂商和型号，然后单击“下一步”。如果手头有包含要安装的驱动程序的磁盘，请单击“从磁盘安装”。
型号
giveio
从磁盘安装(H)...
这个驱动程序没有经过数字签署！
告诉我为什么驱动程序签名很重要
< 上一步(B)
下一步(N) >
取消

图 10

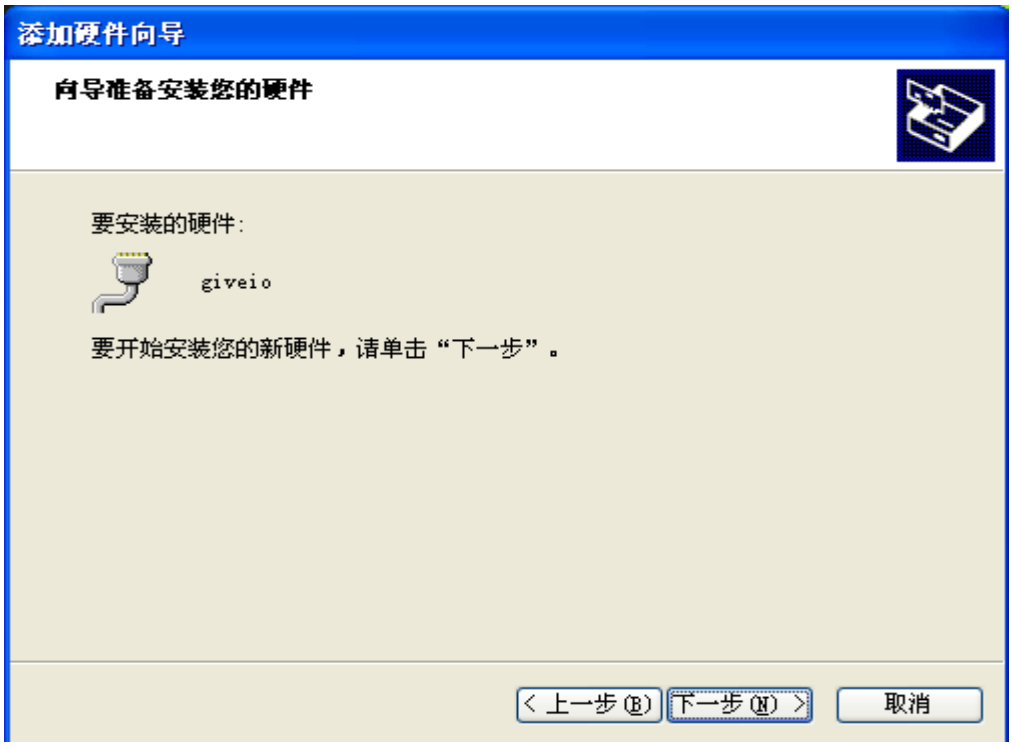

添加硬件向导
向导准备安装您的硬件
要安装的硬件：
giveio
要开始安装您的新硬件，请单击“下一步”。
< 上一步(B)
下一步(N) >
取消

图 11

三、bootloader 的烧写

在 d 盘新建一目录 bootloader，把 sjf2440.exe(在光盘 IMG 目录下）和要烧写的 u-boot.bin，linux 操作系统内核，根文件系统和应用程序压缩包拷贝到该目录下现在的文件是 u-boot.bin,该文件在光盘的 img 文件夹下。

点击“开始”中的“运行”输入 cmd，找到 sjf2440-s.exe 所在文件夹的路径，输入 sjf2440-s.exe /f:u-boot.bin 回车。进入烧写界面，如图 12。
界面会显示 CPU 的 ID：0x0032409d
这时候我们对烧写进行地址位的选择，选择 4，如图 12。

```
C:\Documents and Settings\Administrator\桌面\sjf2440>SJF2440.exe /f:u-boot.bin

+-------------------------------------+
|     SEC JTAG FLASH(SJF) v 0.1       |
|     (S3C2440X & SMDK2440 B/D)       |
+-------------------------------------+
Usage: SJF /f:<filename> /d=<delay>
> S3C2440X(ID=0x0032409d) is detected.

[SJF Main Menu]
 0:K9S1208 prog     1:28F128J3A prog   2:AM29LV800 Prog   3:Memory Rd/Wr
 4:K9F2G08 prog     5:Exit
Select the function to test:4
```

图 12

在此后出现的三次要求输入参数，第一次是让选择 Flash，选 4，然后回车，第二次是选择 JTAG 对 flash 的两种功能，也选 0，然后回车，第三次是让选择起始地址，选 0，然后回车，等待大约 3－5 分钟的烧写时间，如下图所示：

```
[K9F2G08 NAND Flash JTAG Programmer]
K9F2G08U is detected. ID=0xecda

 0 : K9F2G08 Program
 1 : K9F2G08 Print blkPage
 2 : Exit

Select the function to test : 0

[K9F2G08 NAND Flash Writing Program]

Source size:0h~3d8efh

Available target block number : 0~4096
Input target block number : 0
blockIndex = 0
..block #0 erase done
page programing : #0 ~ #123
page program done

 0 : K9F2G08 Program
 1 : K9F2G08 Print blkPage
 2 : Exit

Select the function to test : 2
```

图 13

当 u-boot 烧写完毕后选择参数 2，退出烧写。如下图所示。烧录后关闭 S2440，注意这时需要拔掉并口线与开发板的连线,否则开发板无法正常启动。

四、超级终端设置

将我们 PC 机的串口和板子的串口连接起来，打开超级终端，选择 COM1 口，配置如图 17，保证各项跟图 14 中一样。

图 14

烧写完 u-boot 之后，重启开发板，提示信息如下：

```
U-Boot 2010.03 (Mar 11 2011 - 19:20:34)
I2C:      ready
DRAM:    64 MB
Flash:    2 MB
NAND:    256 MiB
No ethernet found.

In:       serial
Out:      serial
Err:      serial
USB slave is enable!
Net:      dm9000
U-Boot 2010.03 (Mar 11 2011 - 19:20:34)
```

```
Hit any key to stop autoboot:  0
[u-boot@uptech2440]#
```

五、内核、根文件系统烧写

1、配置 bootloader

由于要通过 tftp 下载文件，所以需要配置 tftp 的网络 ip，在配置之前打开 img 下的 tftp32.exe 程序。tftp32.exe 打开后会自动寻找到 PC 机上网卡的 IP，如果 IP 不正确可以通过 IP 的下拉框寻找到本机的 IP，注意一定要保证其显示 IP 为我们 PC 机的 IP。不要随意在 tftp32 中 settings 中改变 IP，否者会导致 tftp32 无法正常使用。也可以使用虚拟机 linux 系统上的 tftp。

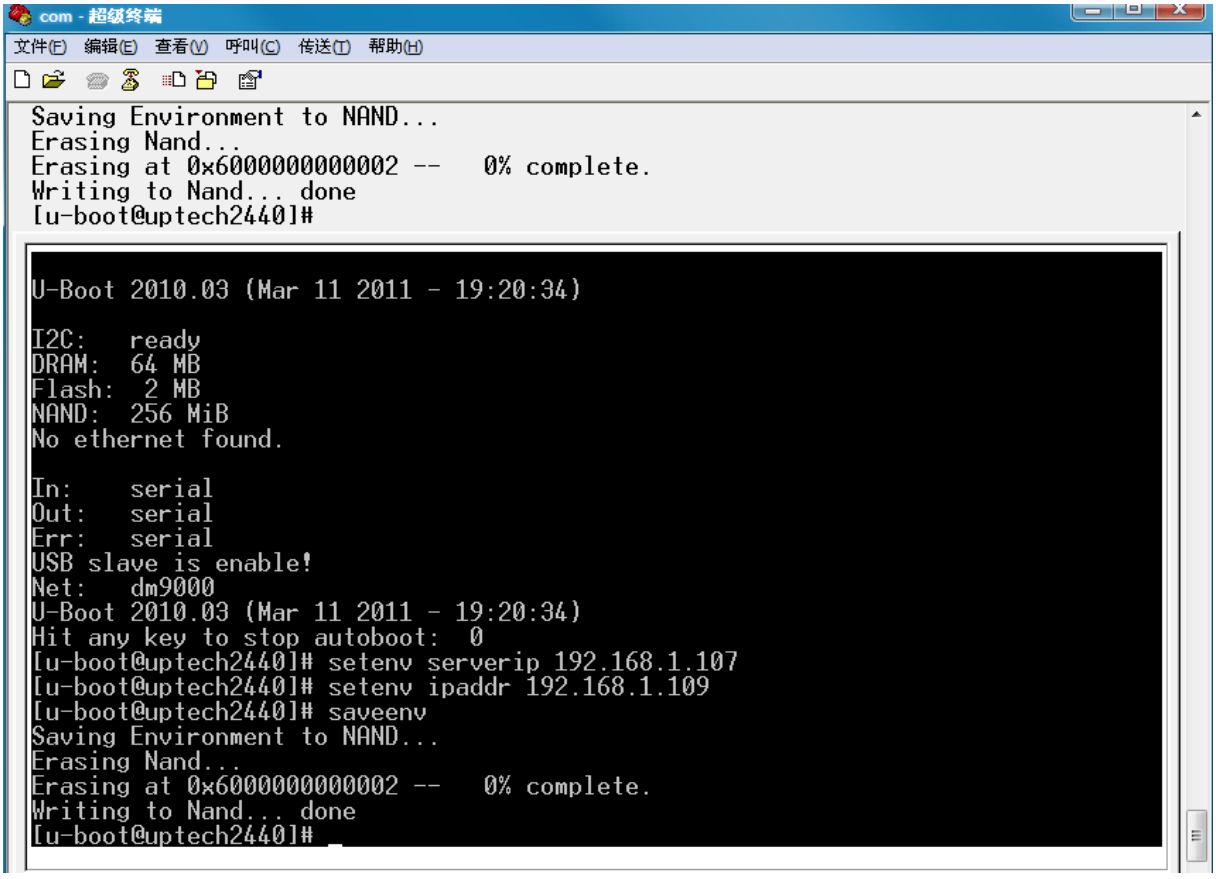
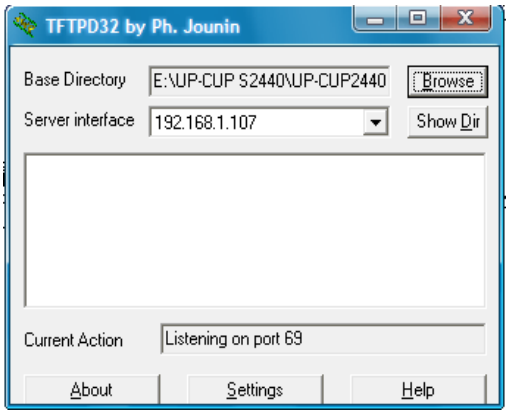


图 15

在提示符下输入 setenv serverip 192.168.1.107，此 IP 为我们 PC 机 IP，再输入 setenv ipaddr 192.168.1.109，此 IP 为我们开发板的 IP。最好输入 saveenv，保存设置，之后就不需要改动了。这些过程参考上图 15。

2、下载 u-boot 的配置脚本文件

首先在 Windows 平台下 tftp 服务的配置：双击/IMG 文件目录下的 tftpd32.exe 文件，对 Windows 下的 tftp 服务进行配置，如下图所示：



注：需要修改文件存放路径(通过 Browse)和 PC 机 IP(开发板上 serverip 应同该 IP 地址保持一致)

在提示符下输入 tftp 0x030008000 up2440.img，如图 16。

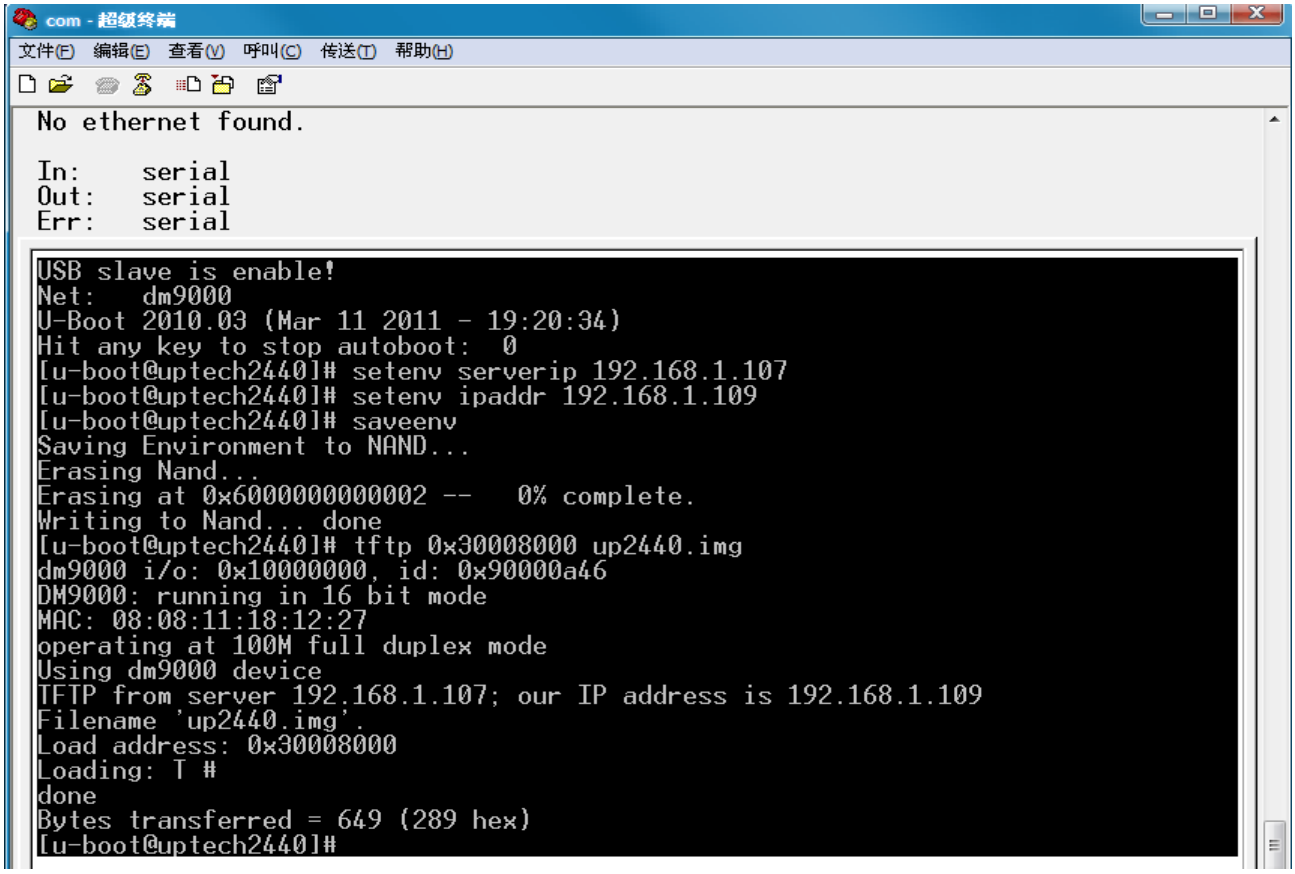


图 16

执行配置脚本，使用 u-boot 的 source 命令，如图 17。

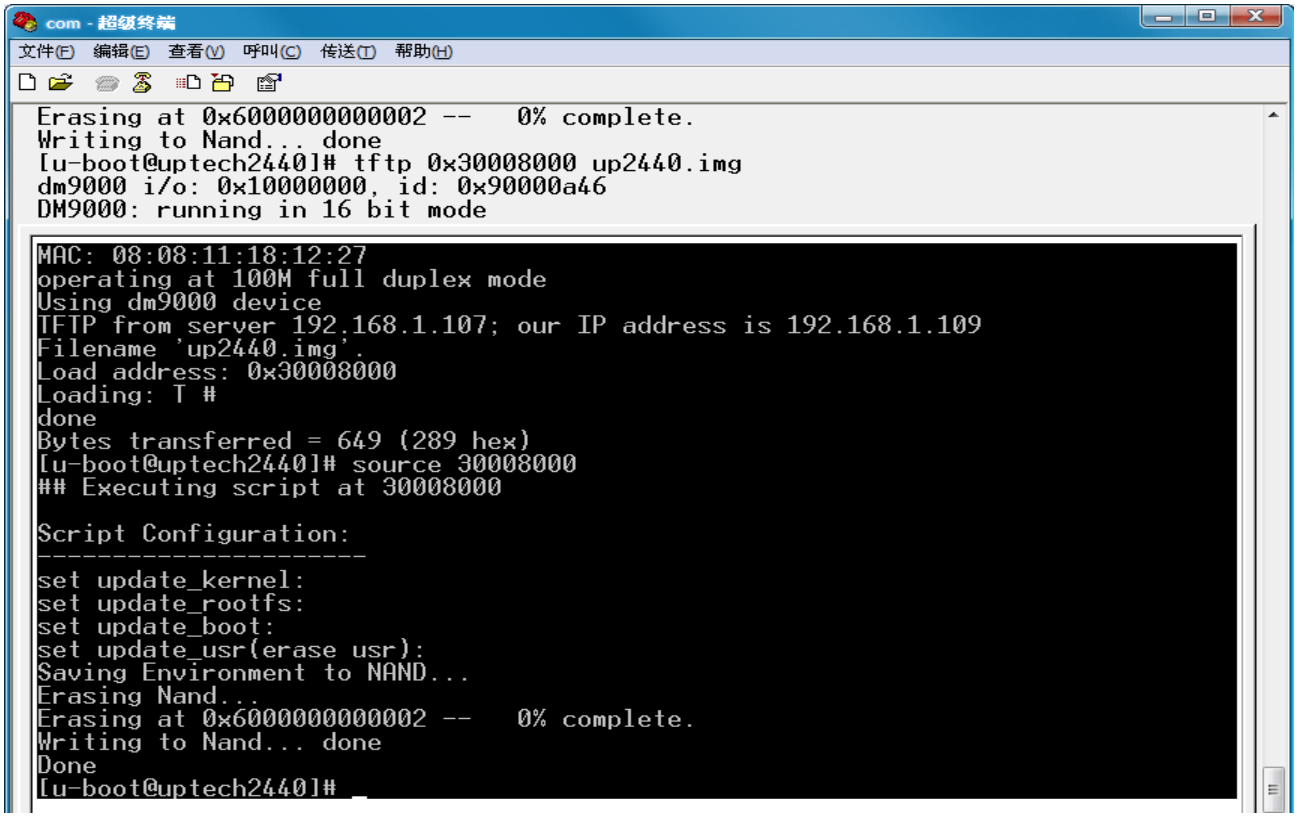


图 17

设置之后，我们可以通过 printenv 指令，查看 u-boot 的环境变量中已经添加了新的变量，如图 17。

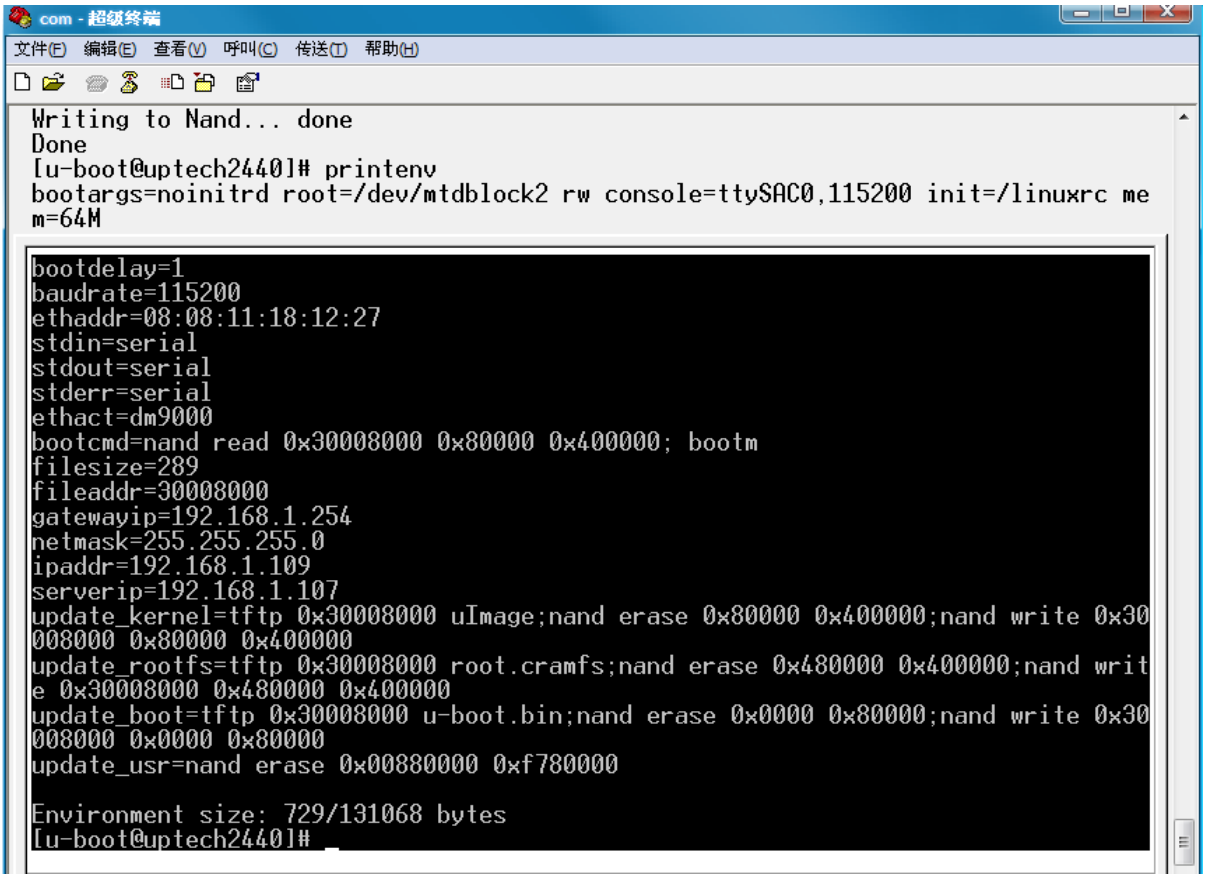


图 17

这时，我们的 u-boot 的配置就结束了。现在除了你要修改 ip 地址之外，就不需要再更改其它的配置了。

3、烧写内核

在配置好 u-boot 之后，执行烧写内核的命令，输入 run update_kernel 指令，如图 18。

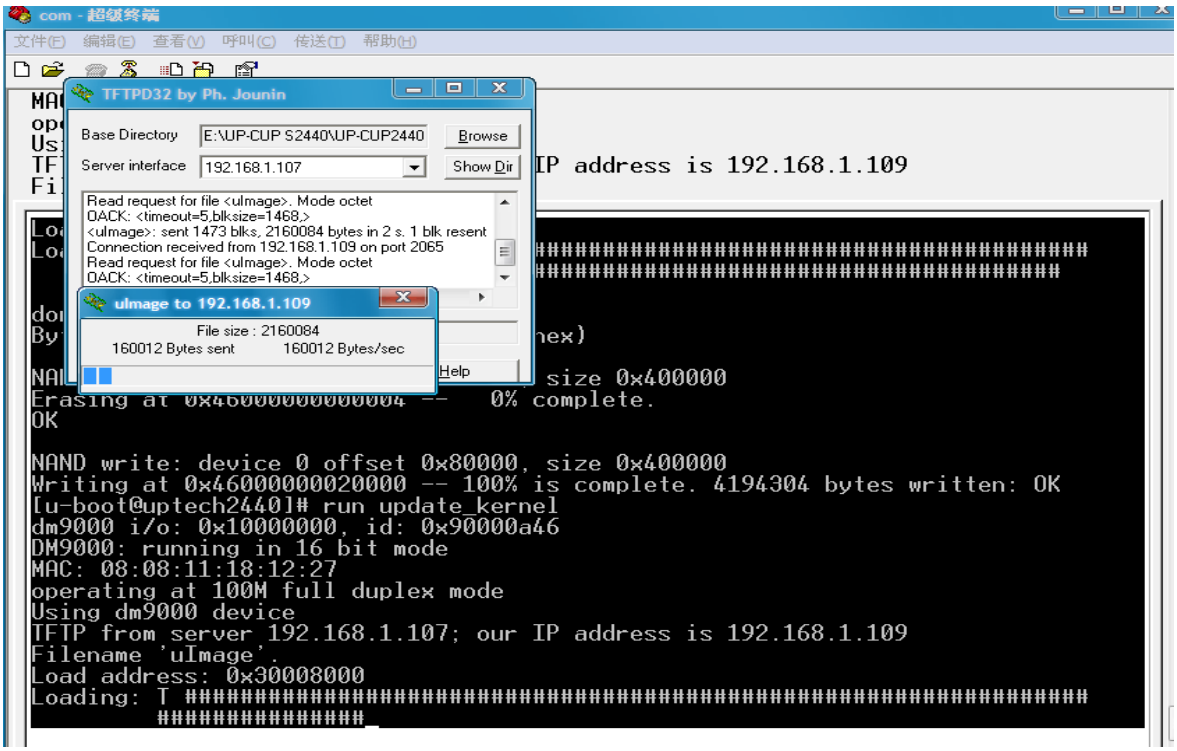


图 18

4、烧写根文件系统

执行烧写根文件系统的命令，输入 run update_rootfs 指令，如图 19

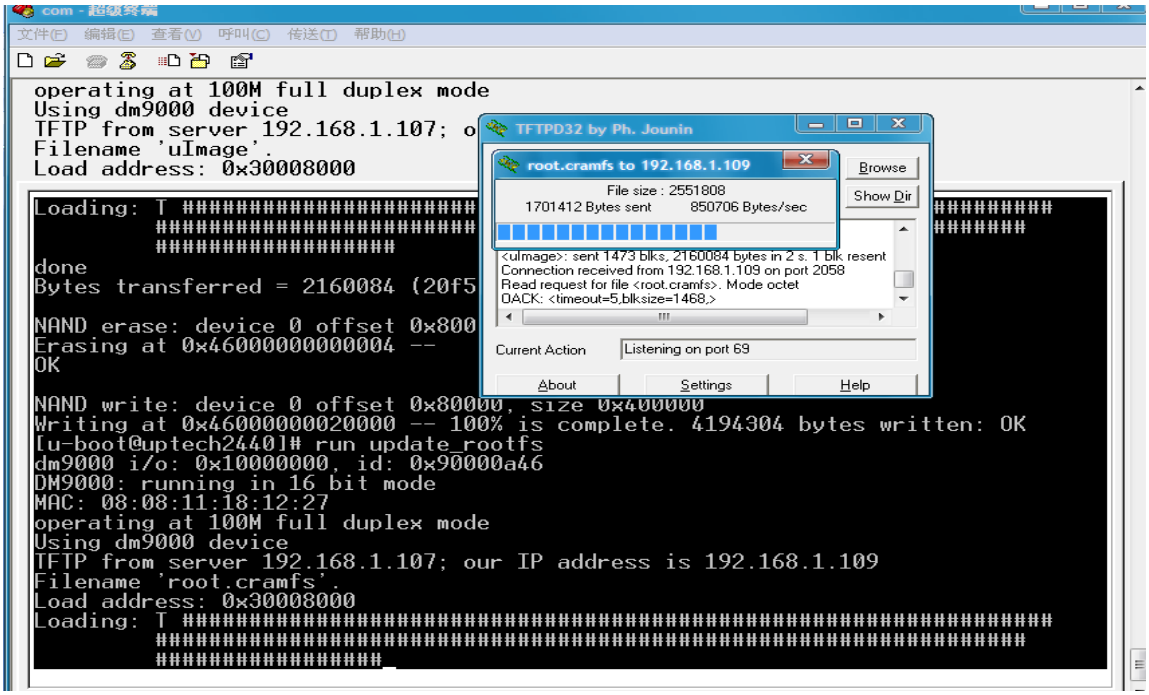


图 19

5、应用程序下载

在 u-boot 下输入 boot 启动 linux 系统，此时不要关闭 tftp32 软件，保证应用程序跟 tftp32 在同一路径下,在 linux 中的 root 目录下输入如 tftp –g –r run_exp.tar.bz2 192.168.1.107（IP 为 tftp32 软件上显示的 PC 机 IP），回车即可将想要使用的应用程序下载至开发板。

6、U-boot 使用说明

除了前几个使用的比较多的命令之外，其它的也就是；

（1）通过 tftp 下载文件，默认是下载的”uImage”文件。命令模式如下:tftp 0x30008000 filename

（2）下载到内存中，如果要保存的话，需要写 flash

（3）boot 执行环境变量中的 bootcmd 命令。

（4）bootm 从内存的 0x30008000 处开始执行。

（5）setenv 设置或者新建环境变量，例如:setenv newenv xxxx xxxx

（6）printenv 显示当前的环境变量到控制台

（7）saveenv 保存环境变量

（8）version 显示版本号和日期。

具体内容可以通过 help 指令查看 u-boot 下各指令。

启动开发板,输入:

```
setenv bootcmd nand read 0x30008000 0x80000 0x400000\; bootm
savenv
boot
```

以下是开发板完整的启动信息：

```
NAND read: device 0 offset 0x80000, size 0x300000
 3145728 bytes read: OK
## Booting kernel from Legacy Image at 30008000 ...
    Image Name:    Linux-2.6.24.4
    Created:       2011-03-14   1:51:59 UTC
    Image Type:    ARM Linux Kernel Image (uncompressed)
    Data Size:     2160020 Bytes =  2.1 MB
    Load Address: 30008000
    Entry Point:  30008040
    Verifying Checksum ... OK
    XIP Kernel Image ... OK
OK

Starting kernel ...

Uncompressing
Linux.....
..... done, booting the kernel.
Linux version 2.6.24.4 (gxj@localhost.localdomain) (gcc version 3.4.6) #419 Mon Mar
```

```
14 09:51:27 CST 2011
CPU: ARM920T [41129200] revision 0 (ARMv4T), cr=c0007177
Machine: SMDK2440
Memory policy: ECC disabled, Data cache writeback
CPU S3C2440A (id 0x32440001)
S3C244X: core 405.000 MHz, memory 101.250 MHz, peripheral 50.625 MHz
S3C24XX Clocks, (c) 2004 Simtec Electronics
CLOCK: Slow mode (1.500 MHz), fast, MPLL on, UPLL on
CPU0: D VIVT write-back cache
CPU0: I cache: 16384 bytes, associativity 64, 32 byte lines, 8 sets
CPU0: D cache: 16384 bytes, associativity 64, 32 byte lines, 8 sets
Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 16256
Kernel command line: noinitrd root=/dev/mtdblock2 rw console=ttySAC0,115200
init=/linuxrc mem=64M
irq: clearing pending ext status 00080300
irq: clearing subpending status 00000003
irq: clearing subpending status 00000002
PID hash table entries: 256 (order: 8, 1024 bytes)
timer tcon=00500000, tcnt a4ca, tcfg 00000200,00000000, usec 00001e57
Console: colour dummy device 80x30
temp=0
console [ttySAC0] enabled
Dentry cache hash table entries: 8192 (order: 3, 32768 bytes)
Inode-cache hash table entries: 4096 (order: 2, 16384 bytes)
Memory: 64MB = 64MB total
Memory: 59940KB available (3996K code, 772K data, 132K init)
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
net_namespace: 64 bytes
NET: Registered protocol family 16
S3C2410 Power Management, (c) 2004 Simtec Electronics
S3C2440: Initialising architecture
S3C2440: IRQ Support
S3C2440: Clock Support, DVS off
S3C24XX DMA Driver, (c) 2003-2004,2006 Simtec Electronics
DMA channel 0 at c4800000, irq 33
DMA channel 1 at c4800040, irq 34
DMA channel 2 at c4800080, irq 35
DMA channel 3 at c48000c0, irq 36
SCSI subsystem initialized
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
```

```
NET: Registered protocol family 23
Bluetooth: Core ver 2.11
NET: Registered protocol family 31
Bluetooth: HCI device and connection manager initialized
Bluetooth: HCI socket layer initialized
NET: Registered protocol family 2
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 2048 (order: 2, 16384 bytes)
TCP bind hash table entries: 2048 (order: 1, 8192 bytes)
TCP: Hash tables configured (established 2048 bind 2048)
TCP reno registered
NetWinder Floating Point Emulator V0.97 (double precision)
fuse init (API version 7.9)
yaffs Mar 14 2011 08:35:20 Installing.
io scheduler noop registered
io scheduler anticipatory registered (default)
io scheduler deadline registered
io scheduler cfq registered
<7>s3c2410fb: devinit
Console: switching to colour frame buffer device 80x30
fb0: s3c2410fb frame buffer device
Serial: 8250/16550 driver $Revision: 1.90 $ 4 ports, IRQ sharing enabled
<3><3><6>s3c2440-uart.0: s3c2410_serial0 at MMIO 0x50000000 (irq = 70) is a
S3C2440
s3c2440-uart.1: s3c2410_serial1 at MMIO 0x50004000 (irq = 73) is a S3C2440
s3c2440-uart.2: s3c2410_serial2 at MMIO 0x50008000 (irq = 76) is a S3C2440
PPP generic driver version 2.4.2
PPP Deflate Compression module registered
PPP BSD Compression module registered
SLIP: version 0.8.4-NET3.019-NEWTTY (dynamic channels, max=256) (6 bit
encapsulation enabled).
CSLIP: code copyright 1989 Regents of the University of California.
SLIP linefill/keepalive option.
DM9000: dm9k_init_module
bwsconk:7ffc

Board init for dm9000a finished!
<DM9KS> I/O: c480e300, VID: 90000a46
eth0: at 0xc480e300 IRQ 18
eth0: Ethernet addr: 08:08:11:18:12:27
Linux video capture interface: v2.00
Uniform Multi-Platform E-IDE driver Revision: 7.00alpha2
ide: Assuming 50MHz system bus speed for PIO modes; override with idebus=xx
```

```
iomap:0xc4810020register the ide hw ok!!
iomap:0xc4814020register the ide hw ok!!
<4>Driver 'sd' needs updating - please use bus_type methods
S3C24XX NAND Driver, (c) 2004 Simtec Electronics
<3><6>s3c2440-nand s3c2440-nand: Tacls=3, 29ns Twrph0=7 69ns, Twrph1=3 29ns
NAND device: Manufacturer ID: 0xec, Chip ID: 0xda (Samsung NAND 256MiB 3,3V
8-bit)
NAND_ECC_NONE selected by board driver. This is not recommended !!
Scanning device for bad blocks
Bad eraseblock 0 at 0x00000000
Bad eraseblock 1 at 0x00020000
Bad eraseblock 660 at 0x05280000
Bad eraseblock 1072 at 0x08600000
Bad eraseblock 1791 at 0x0dfe0000
Creating 4 MTD partitions on "NAND 256MiB 3,3V 8-bit":
0x00000000-0x00080000 : "u-boot"
0x00080000-0x00480000 : "kernel"
0x00480000-0x00880000 : "rootfs"
0x00880000-0x10000000 : "user"
<3><3><6>s3c2410-spi s3c2410-spi.0: registering uptech_spi
s3c2410-spi0 driver initial
usbmon: debugfs is not available
<7>s3c2410-ohci s3c2410-ohci: s3c2410_start_hc:
s3c2410-ohci s3c2410-ohci: S3C24XX OHCI
s3c2410-ohci s3c2410-ohci: new USB bus registered, assigned bus number 1
s3c2410-ohci s3c2410-ohci: irq 42, io mem 0x49000000
the USB device you want to use    needs 0 MA
the power of the USB interface avaliable is 0 MA
usb usb1: configuration #1 chosen from 1 choice
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 2 ports detected
usb usb1: Product: S3C24XX OHCI
usb usb1: Manufacturer: Linux 2.6.24.4 ohci_hcd
usb usb1: SerialNumber: s3c24xx
Initializing USB Mass Storage driver...
usb 1-1: new full speed USB device using s3c2410-ohci and address 2
the USB device you want to use    needs 100 MA
the power of the USB interface avaliable is 500 MA
usb 1-1: configuration #1 chosen from 1 choice
hub 1-1:1.0: USB hub found
hub 1-1:1.0: 4 ports detected
usb 1-1: Product: Generic USB Hub
usb 1-1: Manufacturer: ALCOR
```

```
usbcore: registered new interface driver usb-storage
USB Mass Storage support registered.
usbcore: registered new interface driver microtekX6
usbcore: registered new interface driver gspca
drivers/usb/media/gspcav1-20071224/gspca_core.c: gspca driver 00.60.00 registered
mice: PS/2 mouse device common for all mice
S3C24XX RTC, (c) 2004,2006 Simtec Electronics
<6>s3c2410-rtc s3c2410-rtc: rtc disabled, re-enabling
s3c2410-rtc s3c2410-rtc: rtc core: registered s3c as rtc0
i2c /dev entries driver
<3>s3c24xx_i2c_init
s3c2440-i2c s3c2440-i2c: slave address 0x10
s3c2440-i2c s3c2440-i2c: bus frequency set to 98 KHz
IRQ FINDED!
add bus to i2c core!
s3c2440-i2c s3c2440-i2c: i2c-0: S3C I2C adapter
S3C2410 Watchdog Timer, (c) 2004 Simtec Electronics
<6>s3c2410-wdt s3c2410-wdt: watchdog inactive, reset disabled, irq enabled
Bluetooth: HCI USB driver ver 2.9
usbcore: registered new interface driver hci_usb
s3cmci_init
<3><3>s3cmci_probe
mapped channel 2 to 0
s3c2440-sdi s3c2410-sdi: powered down.
s3c2440-sdi s3c2410-sdi: initialisation done.
s3cmci_probe end
s3c2440-sdi s3c2410-sdi: running at 0kHz (requested: 0kHz).
s3c2440-sdi s3c2410-sdi: running at 198kHz (requested: 197kHz).
s3c2440-sdi s3c2410-sdi: running at 198kHz (requested: 197kHz).
s3c2440-sdi s3c2410-sdi: running at 198kHz (requested: 197kHz).
usbcore: registered new interface driver usbhid
drivers/hid/usbhid/hid-core.c: v2.6:USB HID core driver
TCP cubic registered
NET: Registered protocol family 1
NET: Registered protocol family 17
IrCOMM protocol (Dag Brattli)
s3c2440-sdi s3c2410-sdi: powered down.
Bluetooth: L2CAP ver 2.9
Bluetooth: L2CAP socket layer initialized
Bluetooth: SCO (Voice Link) ver 0.5
Bluetooth: SCO socket layer initialized
Bluetooth: RFCOMM socket layer initialized
Bluetooth: RFCOMM TTY layer initialized
```

```
Bluetooth: RFCOMM ver 1.8
Bluetooth: BNEP (Ethernet Emulation) ver 1.2
Bluetooth: BNEP filters: protocol multicast
Bluetooth: HIDP (Human Interface Emulation) ver 1.2
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
ieee80211: 802.11 data/management/control stack, git-1.1.13
ieee80211: Copyright (C) 2004-2005 Intel Corporation <jketreno@linux.intel.com>
s3c2410-rtc s3c2410-rtc: hctosys: invalid date/time
VFS: Mounted root (cramfs filesystem) readonly.
Freeing init memory: 132K
Warning: unable to open an initial console.
yaffs: dev is 32505859 name is "mtdblock3"
yaffs: passed flags ""
yaffs: Attempting MTD mount on 31.3, "mtdblock3"
yaffs: auto selecting yaffs2
block 593 is bad
block 1005 is bad
block 1724 is bad
temp=0
temp=0

up-tech login:
输入用户名为 root，密码为空

up-tech：/#
```

六、运行应用程序

系统启动后进入、root 目录下面，如下图所示：

```
up-tech:~ #
up-tech:~ #
up-tech:~ #
up-tech:~ #ls
lost+found    test_QT       test_basic    test_modules
up-tech:~ #
```

test_QT:为简单的 QT 演示程序，包含 DA、LED、DC-MOTOR、音视频播放器、视频监控等测试菜单，目录如下图所示：

```
up-tech:~ #ls test_QT/
Qt-embedded-4.4.0  app                s3c2440_ts.ko      test.sh
up-tech:~ #
```

test_basic:开发板基本资源的测试，目录如下所示：

```
up-tech:~ #ls test_basic/
test_485                test_canchat            test_mega8_ps2key
test_IDE                test_da                 test_mega8_ps2mouse
test_INT                test_dc-motor           test_mplayer
test_SD                 test_irda               test_sound
test_VGA                test_led                test_usb
test_ad                 test_mega8_4*4keyboard  test_video
test_blue               test_mega8_iccard       test_wireless_rt73
up-tech:~ #
```

test_modules:为开发板 168 针扩展插槽及 DB9 串口等扩展的模块的测试，目录如下所示：

```
up-tech:~ #ls test_modules/
test-fpga         test_gprs         test_miniprint    test_usb2.0
test_can_module   test_gps          test_rfid         test_zigbee
test_fingermap    test_labelscan    test_sensor
up-tech:~ #
```

下面逐个介绍各个测试步骤：

1、 开发板基本资源的测试

目录为 test_basic 的测试。

1.1da 测试：

实验目录为：/root/test_basic/test_da

先进到 test_basic 目录，再进到 test_da 目录，执行测试脚本 test.sh ，然后用万用表测 da 输出的电压，为 2.0V 左右。

```
up-tech:~ #ls
lost+found    system        test_QT       test_basic    test_modules
up-tech:~ #cd test_basic/
up-tech:~/test_basic #ls
test_485                test_canchat            test_mega8_ps2key
test_IDE                test_da                 test_mega8_ps2mouse
test_INT                test_dc-motor           test_mplayer
test_SD                 test_irda               test_sound
test_VGA                test_led                test_usb
test_ad                 test_mega8_4*4keyboard  test_video
test_blue               test_mega8_iccard       test_wireless_rt73
up-tech:~/test_basic #cd test_da/
up-tech:~/test_basic/test_da #ls
da_main         s3c2440-da.ko  test.sh
up-tech:~/test_basic/test_da #./test.sh
s3c2440-da      device initialized
Current Voltage is 2.000000 v
up-tech:~/test basic/test da #
```

1.2can 回环测试：

实验目录为：/root/test_basic/ test_canchat

进到 test_canchat 目录，执行测试脚本 test.sh ，本测试程序用的是 CAN 的回环测试方式，所以键盘输入会回显到控制终端上，如下图所示，输入 df 回车，回显 df

```
up-tech:~/test_basic #cd test_canchat/
up-tech:~/test_basic/test_canchat #./test.sh
can initialized
can recieve thread begin.

Press "\q!" to quit!

Press Enter to send!
df
df
gh
gh
```

1.3 485 回环测试：

实验目录为：/root/test_basic/test_485

进到 test_485 目录，执行测试脚本 test.sh ，测试效果如下，会连续发送数据，可用接收端接收数据。

```
up-tech:~/test_basic #cd test_485/
up-tech:~/test_basic/test_485 #./test.sh
s3c2440_485     device initialized
s3c2440 485 device open!
####s3c2440 485 device ready to send####
0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz0123456789:;<
>?@ABCDEFGHIJKLMNOPQRSs3c2440 485 device release!
```

1.4 IDE 硬盘测试：

首先将 IDE 硬盘插入实验板的插槽“ATA-44 HDD”插槽后，UP-CUP2440 中关于 IDE 的启动信息如下显示：

```
Uniform Multi-Platform E-IDE driver Revision: 7.00alpha2
ide: Assuming 50MHz system bus speed for PIO modes; override with

idebus=xx
oldval_bwscon:0x22111912newval_bwscon:0x22111952iomap:0xc4810020tmp is

0x00
tmp22 is 0x01
iomap:0xc4812010register the ide hw ok!!
oldval_bwscon:0x22111952newval_bwscon:0x22111952iomap:0xc4814020tmp is

0x01
tmp22 is 0x01
iomap:0xc4816010register the ide hw ok!!
hda: HTS541040G9AT00, ATA DISK drive
hdc: HTS541040G9AT00, ATA DISK drive
ide0 at 0xc4810020-0xc4810027,0xc481201c on irq 50
ide1 at 0xc4814020-0xc4814027,0xc481601c on irq 50 (shared with ide0)
```

```
hda: max request size: 128KiB
hda: 78140160 sectors (40007 MB) w/7539KiB Cache, CHS=65535/16/63
hda: cache flushes supported
 hda: hda1 hda2 hda3 < hda5 >
hdc: max request size: 128KiB
hdc: 78140160 sectors (40007 MB) w/7539KiB Cache, CHS=65535/16/63
hdc: cache flushes supported
 hdc: hdc1 hdc2 hdc3 < hdc5 >
```

查看硬盘分区

```
up-tech:~ # ls /dev/hd
hda   hda1  hda2  hda3  hda5
```

挂载硬盘。

```
up-tech:~ # fdisk -l
Disk /dev/hda: 40.0 GB, 40007761920 bytes
240 heads, 63 sectors/track, 5168 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes

   Device Boot      Start         End      Blocks   Id System
/dev/hda1   *           1        1355    10243768+  7 HPFS/NTFS
/dev/hda2            1356        3262    14416920   c Win95 FAT32 (LBA)
/dev/hda3            3263        5168    14409360   f Win95 Ext'd (LBA)
/dev/hda5            3263        5168    14409328+  7 HPFS/NTFS
[root@UP_2440 ~]# mount -t vfat /dev/hda2 /mnt/udisk/
```

注：只能挂载 FAT32 格式

测试硬盘

```
up-tech:~ # cd /mnt/udisk/
up-tech:/mnt/udisk # ls
??2007         ?????          Recycled       img-2410linux
up-tech:/mnt/udisk # mkdir uptech
 ??2007         ?????          Recycled       img-2410linux  uptech
```

注：在/root/test_basic/test_IDE 目录中的测试脚本只是根据上面的操作列出了一个简单的测试脚本，若用户的分区是其他设备名，相应的更改挂载命令中的设备名即可。

```
up-tech:~/test_basic #cd test_IDE/
up-tech:~/test_basic/test_IDE #ls
test.sh
up-tech:~/test_basic/test_IDE #more test.sh
mount -t vfat /dev/hda2 /mnt/udisk
ls /mnt/udisk
up-tech:~/test_basic/test_IDE #
```

1.5 开发板中断按键测试：

实验目录为：/root/test_basic/test_INT

进入实验目录为：/root/test_basic/test_INT 目录，执行测试脚本以后，按下开发板上的中断按键，会触发中断，测试效果如下图所示：

```
up-tech:~/test_basic #cd test_INT/
up-tech:~/test_basic/test_INT #./test.sh
s3c2440_int: module license 'unspecified' taints kernel.
enter interrupt 5 !
enter interrupt 5 !
s3c2440-int int05   initialized
up-tech:~/test_basic/test_INT #enter interrupt 5 !
enter interrupt 5 !
```

1.6 直流电机测试：

实验目录为：/root/test_basic/test_dc-motor

进入实验目录为：/root/test_basic/test_dc-motor 目录，执行测试脚本以后，开发板上的直流电机会按照设定的速度转动，测试效果如下图所示：

```
up-tech:~/test_basic #cd test_dc-motor/
up-tech:~/test_basic/test_dc-motor #./te
-bash: ./te: No such file or directory
up-tech:~/test_basic/test_dc-motor #./test.sh
s3c2440-dc-motor        device initialized
S3c2440 DC Motor device open now!
setpwm = -512
setpwm = -511
```

1.7 SD 卡测试：

插入 UP-CUP2440 型设备配套 SD 卡,此时会有如下提示

```
up-tech:~# s3c2440-sdi s3c2410-sdi: running at 0kHz (requested: 0kHz).
s3c2440-sdi s3c2410-sdi: running at 198kHz (requested: 197kHz).
s3c2440-sdi s3c2410-sdi: running at 198kHz (requested: 197kHz).
s3c2440-sdi s3c2410-sdi: running at 198kHz (requested: 197kHz).
s3c2440-sdi s3c2410-sdi: running at 198kHz (requested: 197kHz).
s3c2440-sdi s3c2410-sdi: running at 198kHz (requested: 197kHz).
s3c2440-sdi s3c2410-sdi: running at 198kHz (requested: 197kHz).
s3c2440-sdi s3c2410-sdi: running at 198kHz (requested: 197kHz).
s3c2440-sdi s3c2410-sdi: running at 16875kHz (requested: 25000kHz).
s3c2440-sdi s3c2410-sdi: running at 16875kHz (requested: 25000kHz).
mmc0: new SD card at address 0002
mmcblk0: mmc0:0002 SD128 124672KiB
mmcblk0: p1
```

根据提示”mmcblk0:p1”，用户可将 mmcblk0p1 挂载至/mnt/sdcard 目录下,可对对其进行 mkdir,cd，rm 等操作。

```
up-tech:~# mount /dev/mmcblk0p1 /mnt/sdcard/
up-tech:~# cd /mnt/sdcard
up-tech:/mnt/sdcard # mkdir uptech
up-tech:/mnt/sdcard # ls
uptech
up-tech:/mnt/sdcard #
```

拔下 SD 卡系统提示如下所示。

```
up-tech:/mnt/sdcard # cd mmc0: card 0002 removed
s3c2440-sdi s3c2410-sdi: powered down.
```

注：/root/test_basic/test_SD 卡目录，只是简单的挂载 SD 卡进行测试，若是挂载的是其他设备，相应的要更改挂载命令。

```
up-tech:~/test_basic #cd test_SD/
up-tech:~/test_basic/test_SD #ls
test.sh
up-tech:~/test_basic/test_SD #more test.sh
mount /dev/mmcblk0p1 /mnt/sdcard
ls /mnt/sdcard

up-tech:~/test_basic/test_SD #
```

1.8 红外通信测试：

实验目录为：/root/test_basic/test_irda,实验前确保 PC 机已经连接红外接收器设备并装好驱动，具体介绍见实验指导书红外通信章节。

进入实验目录为：/root/test_basic/test_irda 目录，执行测试脚本以后，进行文件传输，测试效果如下图所示，具体测试步骤见实验指导书红外通信章节。

```
up-tech:~/test_basic #cd test_irda/
up-tech:~/test_basic/test_irda #ls
irattach        irxfer          test.sh
irda.sh         s3c2440_ir.ko   test.txt
up-tech:~/test_basic/test_irda #./test.sh
tmp is 0xc2
tmp is 0xf4
tmp is 0xc4
tmp is 0x00
ULCON2:0x3
ULCON2:0x40
init irda
Send files to and receive files from win95
```

1.9 VGA 测试：

实验目录为：/root/test_basic/test_VGA,实验前确保已经连接电脑显示器到 VGA 接口。

进入实验目录为：/root/test_basic/test_ VGA 目录，执行测试脚本以后，开发板 LCD 上面显示的内容将同步显示在 VGA 显示器上，测试效果如下图所示。

```
up-tech:~/test_basic #cd test_VGA
up-tech:~/test_basic/test_VGA #ls
setvga   test.sh
up-tech:~/test_basic/test_VGA #./test.sh
Version ID is 0x25
```

1.10 LED 测试：

实验目录为：/root/test_basic/test_led

进入实验目录为：/root/test_basic/test_ led 目录，执行测试脚本以后，开发板 LED 数码管上面显示设置的数字，测试效果如下图所示。

```
up-tech:~/test_basic #cd test_led/
up-tech:~/test_basic/test_led #./test.sh
insmod: cannot insert 's3c2440-led.ko': File exists
led device open sucess!
will enter TUBE LED  ,please waiting ..............DOT buffer is ff
DOT buffer is ff00

DOT buffer is c0
DOT buffer is c090
DOT buffer is f9
DOT buffer is f980
DOT buffer is a4
DOT buffer is a4f8
```

1.11 AD 测试：

实验目录为：/root/test_basic/test_ad，注意 AD 的驱动和触摸屏的驱动冲突，要进行此试验，请确保不要加载触摸屏的驱动，若加载了，请重启开发板后再进行此实验。

进入实验目录为：/root/test_basic/test_ ad 目录，执行测试脚本以后，旋转开发板上的 3 路 AD 旋钮，将显示相应采集到的 AD 值，测试效果如下图所示。

```
up-tech:~/test_basic #cd test_ad/
up-tech:~/test_basic/test_ad #./test.sh
add s3c2440_adc ok!

Press Enter key exit!
a0=  3.2968      a1=  0.0000      a2=  1.5759
```

1.12 蓝牙测试：

实验目录为：/root/test_basic/test_blue，注意先插入蓝牙模块，同时 PC 机插入另

一模块并安装好驱动，再进行此实验，具体配置参考实验指导书中蓝牙实验，本目录下只是简单的测试脚本。

```
up-tech:~/test_basic #cd test_blue/
up-tech:~/test_basic/test_blue #./test.sh
init bluetooth
Starting Bluetooth:                                        fail
Can't get device info: No such device
Device is not available: No such device
ifconfig: SIOCSIFADDR: No such device
up-tech:~/test_basic/test_blue #more test.sh
./bluzmodule.sh
hciconfig hci0 up
hcitool scan
ifconfig bnep0 192.168.1.109 up
up-tech:~/test_basic/test_blue #
```

1.13 无线网卡测试：

实验目录为：/root/test_basic/test_wireless_rt73，注意先插入无线网卡模块，再进行此实验，具体配置参考实验指导书中无线网卡实验，本目录下只是简单的加载无线网卡的驱动。

```
up-tech:~/test_basic #cd test_wireless_rt73/
up-tech:~/test_basic/test_wireless_rt73 #ls
rt73.ko  test.sh
up-tech:~/test_basic/test_wireless_rt73 #./test.sh
usbcore: registered new interface driver rt73
up-tech:~/test_basic/test_wireless_rt73 #more test.sh
insmod rt73.ko
#ifconfig eth0 down
#ifconfig rausb0 192.168.1.199 up
#route add default gw 192.168.1.254 dev rausb0
#ifconfig -a
#iwlist scanning
```

1.14 U 盘测试：

实验目录为：/root/test_basic/test_usb，注意先插入 u 盘模块，再进行此实验，当从 USB 口插入设备后，显示如下信息。

```
up-tech：/ #usb 1-1: new full speed USB device using s3c2410-ohci and address 2
usb 1-1: configuration #1 chosen from 1 choice
scsi0 : SCSI emulation for USB Mass Storage devices
usb 1-1: Product: USB Mass Storage Device
usb 1-1: Manufacturer: Myson Century, Inc.
usb 1-1: SerialNumber: 100
scsi 0:0:0:0: Direct-Access     ST940211 3A              3.01 PQ: 0 ANSI: 0
CCS
sd 0:0:0:0: [sda] 78140160 512-byte hardware sectors (40008 MB)
```

```
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] 78140160 512-byte hardware sectors (40008 MB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Assuming drive cache: write through
 sda:<7>usb-storage: queuecommand called
 sda1
sd 0:0:0:0: [sda] Attached SCSI disk
sd 0:0:0:0: Attached scsi generic sg0 type 0
```

进入/root/test_basic/test_usb，查看一下，执行脚本文件./test.sh，可以看到U盘内的设备，如下图所示：

```
up-tech:~/test_basic #cd test_usb/
up-tech:~/test_basic/test_usb #./test.sh
2410-S??????.pdf                    Mediaplay
2410-S?????@2007.05.29.pdf          Qtopia.tar.bz2
270 dvp ?????????                   UP-TECH PXA270-S????.pdf
270???? WINCE?????.doc              UP-TECH PXA270A????.pdf
270CL_BSP_20080927                  UP-TECHPXA270-S-LINUX?????V1.0.pdf
270cl                               UP-TECHPXA270-S??????_v1.0.pdf
?? 082.jpg                          UP-TECHPXA270A-LINUX?????V5.0.pdf
?? 088.jpg                          UP-TECHPXA270A_linux??????_v1.2.pdf
?? ????.txt                         adcapp
???                                 brick
????                                dalianpeixun
?????                               gprs
????????.doc                        gprs.exe
ArrayLed                            gpxtext
Cheese.wmv                          img
EXE Files2410                       led
EXE Files270                        nk.bin
I2C??????.pdf                       tftp??2410-s linux??.pdf
JFlash MM                           ut2410classic.pbxml
Linux????.txt
```

1.15 视频播放器测试：

实验目录为：/root/test_basic/test_mplayer

进入 /root/test_basic/test_mplayer 目录，执行测试脚本./test.sh，会在 LCD 上播放倒霉熊视频，并有相应的音频输出

```
up-tech:~/test_basic #cd test_mplayer/
up-tech:~/test_basic/test_mplayer #./test.sh
<7>dma10: s3c2410_request_dma: client=I2SSDO, dev=00000000
mapped channel 10 to 2
mapped channel 9 to 1
UDA1341 audio driver initialized
MPlayer 1.0pre4-2.95.3 (C) 2000-2004 MPlayer Team

CPU: ARM
Reading config file /home/samba/tmp/mplayer/etc/mplayer/mplayer.conf:
ectory
```

1.16 音频播放器测试：

实验目录为：/root/test_basic/test_sound

进入 /root/test_basic/test_sound 目录，执行测试脚本./test.sh，在音频接口会输出西游记歌曲

```
up-tech:~/test_basic/test_mplayer #cd ../test_sound/
up-tech:~/test_basic/test_sound #./test.sh
insmod: cannot insert 'soundcore.ko': File exists
insmod: cannot insert 's3c2440-uda1341.ko': File exists
MPEG Audio Decoder 0.14.2 (beta) - Copyright (C) 2000-2001 Robert Leslie
audio open
     Title: 通天大道宽又阔(续集)-片头曲崔京浩领唱
audio_sync
audio_sync
```

1.17 视频监控测试：

实验目录为：/root/test_basic/test_video，首先确保开发板连接了中星微的 USB 摄像头。

进入 /root/test_basic/test_video 目录，执行测试脚本./test.sh，LCD 会显示监控画面

```
up-tech:~/test_basic/test_sound #cd ../test_video/
up-tech:~/test_basic/test_video #./test.sh
video /dev/video0 caputure: 320x240
frame buffer: 640x480,  16bpp, 0x96000byte
```

1.18 4*4 矩阵键盘测试：

实验目录为：/root/test_basic/test_mega8_4*4keyboard

进入 /root/test_basic/test_mega8_4*4keyboard 目录，执行测试脚本./test.sh，然后按下开发板的 4*4 矩阵按键，会读出相应的键值，如下图所示。

```
up-tech:~/test_basic #cd test_mega8_4\*4keyboard/
up-tech:~/test_basic/test_mega8_4*4keyboard #./test.sh
no PS/2 device found on PS/2 Port 0!
no PS/2 device found on PS/2 Port 1!
input: s3c2410_mouse as /class/input/input0
input: s3c2410_keyboard as /class/input/input1
initialization success!
keyboard is opened
keyboard opend!
which key you press is 4
which key you press is 5
```

1.19 IC 卡测试：

实验目录为：/root/test_basic/test_mega8_iccard，首先确保开发板 IC 卡座插入了

IC卡

进入 /root/test_basic/test_mega8_iccard 目录，执行测试脚本./test.sh，会读出IC卡内容，如下图所示。

```
up-tech:~/test_basic/test_mega8_4*4keyboard #cd ../test_mega8_iccard/
up-tech:~/test_basic/test_mega8_iccard #./test.sh
insmod: cannot insert 'mega8.ko': File exists
iccard open sucess!
Write 96 bytes data to /dev/Mega8-icc

Read 96 bytes data from /dev/Mega8-icc

      !   "   #   $   %   &   '
  (   )   *   +   ,   -   .   /
  0   1   2   3   4   5   6   7
  8   9   :   ;   <   =   >   ?
  @   A   B   C   D   E   F   G
  H   I   J   K   L   M   N   O
  P   Q   R   S   T   U   V   W
  X   Y   Z   [   \   ]   ^   _
  `   a   b   c   d   e   f   g
  h   i   j   k   l   m   n   o
  p   q   r   s   t   u   v   w
  x   y   z   {   |   }   ~
up-tech:~/test basic/test mega8 iccard #
```

1.20 PS2 键盘测试：

实验目录为：/root/test_basic/test_mega8_ps2key，首先确保开发板连接了 PS2 键盘

进入 /root/test_basic/test_mega8_ps2key 目录，执行测试脚本./test.sh，会切换到 PS2 键盘控制终端，如下图所示。

```
up-tech:~/test_basic/test_mega8_iccard #cd ../test_mega8_ps2key/
up-tech:~/test_basic/test_mega8_ps2key #more test.sh
insmod mega8.ko
getty /dev/tty1 115200
up-tech:~/test basic/test mega8 ps2key #./test.sh
```

可使用：getty /dev/s3c2410_serial0 115200　　切换会原来的控制终端

1.21 PS2 鼠标测试：

实验目录为：/root/test_basic/test_mega8_ps2mouse，首先确保开发板连接了 PS2 鼠标

进入 /root/test_basic/test_mega8_ps2mouse 目录，执行测试脚本./test.sh，移动鼠标，会读取相应的坐标数据，如下图所示。

```
up-tech:~/test_basic/test_mega8_ps2key #cd ../test_mega8_ps2mouse/
up-tech:~/test_basic/test_mega8_ps2mouse #more test.sh
insmod mega8.ko
hexdump /dev/event1
up-tech:~/test_basic/test_mega8_ps2mouse #./test.sh
insmod: cannot insert 'mega8.ko': File exists
```

2、 扩展模块测试

实验目录为:/root/test_modules

```
up-tech:~/test_basic #cd ../test_modules/
up-tech:~/test_modules #ls
test-fpga        test_gprs        test_miniprint   test_usb2.0
test_can_module  test_gps         test_rfid        test_zigbee
test_fingermap   test_labelscan   test_sensor
up-tech:~/test_modules #
```

2.1 FPGA 模块测试：

实验目录为：/root/modules/test-fpga，首先确保 FPGA 模块连接到开发板的扩展插槽上。

进入 /root/modules/test-fpga 目录，执行测试脚本./test.sh，会显示如下数据，同时 FPGA 模块上指示灯会按照测试程序设定的频率闪烁，如下图所示。

```
up-tech:~/test_modules #cd test-fpga/
up-tech:~/test_modules/test-fpga #./test.sh
fpga    device initialized
mmap offset=0x0, protect=0x7f.
mmap physical=0x20000, protect=0x7f.
mmap end=0x40017000, start=0x40016000.
mmap offset=0x0, protect=0x7f.
mmap offset=0x0, protect=0x7f.
mmap offset=0x0, protect=0x7f.
mmapvma=0xc3559518,vma->vm_start=0x40016000, physical=0x20000,vsize=0x1000, vma->vm_page
prot=0x73
up-tech:~/test_modules/test-fpga #
```

2.2 CAN 扩展模块测试：

实验目录为：/root/modules/test_can_module，首先确保 CAN 模块连接到开发板左下方的 CAN 接口及电源接口上。

进入 /root/modules/test_can_module 目录，执行测试脚本./test.sh，会显示如下数据，如下图所示。

```
up-tech:~/test_modules #cd test_can_module/
up-tech:~/test_modules/test_can_module #./test.sh
can initialized
add s3c2440_adc ok!
    +-----------------------------------------------------------+
    |   Please input 1,2,3,4,5,or 6 to respectively operate     |
    |  led1,led2,led3,buzzer,numeral transitor or step motor    |
    +-----------------------------------------------------------+
```

选择 1,指定操作 LED1,输入 1 控制灯亮,选择 0,控制灯灭,输入 2,退出进入下一个选择

```
    +-------------------------------------------------------+
    |   Please input 1,2,3,4,5,or 6 to respectively operate   |
    |  led1,led2,led3,buzzer,numeral transitor or step motor  |
    +-------------------------------------------------------+
1
Input '0' or '1' to set led1,'2' to exit this cycle
1

0x81,1
Input '0' or '1' to set led1,'2' to exit this cycle
2
```

输入 4,选择控制蜂鸣器,输入 1,声响,输入 0,声灭,输入 2,推出进入下一项选择.

```
    +-------------------------------------------------------+
    |   Please input 1,2,3,4,5,or 6 to respectively operate   |
    |  led1,led2,led3,buzzer,numeral transitor or step motor  |
    +-------------------------------------------------------+
4
Input 1 to start buzzer 0 to stop buzzer,or 2 to exit this cycle
1
Input 1 to start buzzer 0 to stop buzzer,or 2 to exit this cycle
0
Input 1 to start buzzer 0 to stop buzzer,or 2 to exit this cycle
2
```

输入 5 可看到数码管变化

```
    +-------------------------------------------------------+
    |   Please input 1,2,3,4,5,or 6 to respectively operate   |
    |  led1,led2,led3,buzzer,numeral transitor or step motor  |
    +-------------------------------------------------------+
5
Input a number to light numeral transistor
Input '0xaa' to put out numeral transistor
```

输入 6,控制 AD 采集.通过转动开发板平台上 ADC POT1 可观察到效果

```
    +-------------------------------------------------------+
    |   Please input 1,2,3,4,5,or 6 to respectively operate   |
    |  led1,led2,led3,buzzer,numeral transitor or step motor  |
    +-------------------------------------------------------+
6
The sample value is:   -199
The sample value is:   -324
The sample value is:   -127
The sample value is:    208
```

2.3 指纹识别模块测试:

实验目录为：/root/modules/test_fingermap，首先确保指纹识别模块连接到开发板168扩展插槽上。

进入 /root/modules/test_fingermap 目录，执行测试脚本./test.sh，会显示如下数据，如下图所示。

```
up-tech:~/test_modules #cd test_fingermap/
up-tech:~/test_modules/test_fingermap #./test.sh
tmp is 0xc0
tmp is 0xc0

selected clock c0417158 (pclk) quot 329, calc 9588
 *****try to opselected clock c0417158 (pclk) quot 54, calc 57528
selected clock c0417158 (pclk) quot 54, calc 57528
en com1*****
 *****open com1 success*****
 ********set COM1 succeed! **********
fingermap inited !
<fingermap control shell>
 [1]  Get fingermap record num
 [2]  User Enroll
 [3]  User Auth
 [4]  Delete fingermap record
 [5]  Clear all fingermap record
 [0]  Print the end of the context an exit
 [**] help menu
 [0]  print the end of the context an exit
shell>
```

按照菜单，输入1可选择记录号码。

```
shell> 1

get num
finger num  =1
shell>
```

按照菜单,输入2注册用户.

```
shell> 2
Please input user's fingerprint!
get one finger
get another finger
creat img  ok and saved

shell>
```

按照菜单,输入3匹配用户指纹.

```
shell> 3
Please input user's fingerprint!
auth ok id =2 match score = 76
```

2.4 GPRS模块测试：

实验目录为：/root/modules/test_gprs，首先确保 GPRS 模块连接到开发板左下方的 168 扩展插槽上。

进入 /root/modules/test_gprs 目录，执行测试脚本./test.sh，会显示如下数据，如下图所示。

```
up-tech:~/test_modules/test_gprs #./test.sh
insmod: cannot insert 'mega8.ko': File exists
insmod: cannot insert 's3c2440-tty2.ko': File exists
  Default baudrate is 9600 bps. If not, please enter the baudrateselected clock c041715
(pclk) quot 54, calc 57528
selected clock c0417158 (pclk) quot 329, calc 9588
 as a parameter
kbd init...

read modem

<gprs control shell>
 [1]  give a call
 [2]  respond a call
 [3]  hold a call
 [4]  send a msg
 [5]  change baudrate
 [6]  exit
 [**] help menu
keyshell>
```

按照菜单，输入 1 可选择拨号功能。并在其中输入电话号码（通过开发板 4*4 键盘输入），即可打出电话。输入 3 可挂断电话，具体现象如下图所示。

```
keyshell>$: 1



you select to gvie a call, please input number:15811043080

calling......
keyshell>$: 3
```

如要验证通话效果可连接耳机和话筒来实现。

注意，此时数字为开发板 4*4 键盘输入。在 GPRS 模块背面插入手机卡，并插上天线。

2.5 GPS 模块测试：

实验目录为：/root/modules/test_gps，首先确保 GPS 模块连接到开发板左下方的 168 扩展插槽上。

进入 /root/modules/test_gps 目录，执行测试脚本./test.sh，会显示如下数据，如下图所示。

```
up-tech:~/test_modules/test_gprs #cd ../test_gps
up-tech:~/test_modules/test_gps #./test.sh
insmod: cannot insert 's3c2440-tty2.ko': File exists
selected clock c0417158 (pclk) quot 329, calc 9588
Default baudrate is 480selected clock c0417158 (pclk) quot 658, calc 4801
0 bps. If not, please enter baudrate as a parameter
read modem
```

```
Default baudrate is 4800 bps. If not, please enter baudrate as a temp=0
parameter
read modem
$GPGGA,,,,,,0,00,,,,,,,*66
DATE     : 0-00-00
TIME     :  00:00:00
Latitude :     0.0000
Longitude:     0.0000
high     :     0.0000
STATUS   :
$GPRMC,,V,,,,,,,,,,N*53
6
DATE     : 1945--10--44
TIME     :  06:-44:-43
Latitude :     0.0000 ,
Longitude:     0.0000 ,
high     :     0.0000
STATUS   : V
$GPGGA,,,,,,0,00,,,,,,,*66
DATE     : 1945--10--44
TIME     :  06:-44:-43
Latitude :     0.0000 ,
Longitude:     0.0000 ,
high     :     0.0000
STATUS   : V
```

默认波特率 B4800

根据信号的强弱，程序刷新正确信息的时间不同，可能需要几分钟来显示正确信息。

2.6 条码扫描模块测试：

实验目录为：/root/modules/test_labelscan，首先确保条码扫描模块的串口连接到开发板上方的串口 1 上。

注意：在扫描前需要将条码扫描器初始化，具体做法是：

1.打开条形码模块的用户手册翻到第 2 页,扫描左上角[设置]条码

2.翻到第 3 页,扫一下[接口选择]条码,之后再扫[RS-232]

3.翻到第 7 页，先扫[传输波特率],再扫[57600bps]

4.扫第 8 页右下角的退出

5.此时条码模块初始化设置完毕,即可进行接下来的实验步骤

进入 /root/modules/test_labelscan 目录，执行测试脚本./test.sh，会显示如下数据，如下图所示。

```
up-tech:~/test_modules #cd test_labelscan/
up-tech:~/test_modules/test_labelscan #./test.sh
tmp is 0xc0
tmp22 is 0xc1

creat database:test.db

creat table:table merchandise(contents:id name price)

and add two records

        name            id              price
        ------------------------------------------------------------
        ibm             277325073               483
        sun             433443          400

selected clock c0417158 (pclk) quot 329, calc 9588
selected clock c0417158 (pclk) quot 54, calc 57528
```

```
creat database:test.db

creat table:table merchandise(contents:id name price)

and add two records

<miniprint control shell>
 [1]  select all the records in table merchandise
 [2]  select the the record which you known its name
 [3]  select the record by the BarCode Scanner
 [4]  delete record
 [5]  add record
 [**] help menu
 [0]  print the end of the context an exit
shell>
```

选择 1,查看历史记录

```
shell> 1

      name          id            price
      ------------------------------------------------------------
      ibm           277325073             483
      sun           433443        400
```

输入 5 添加一条记录.

```
shell> 5
please enter the info of the record you want to add!
id:9787508327549
```

```
get: 9787508327549

name:linux

price:48

shell>
shell>
```

按照菜单,输入 3 按照条形码寻找相应的记录.(注意是用条码模块扫描标签)

```
shell> 3
enter the id:
read:14   9787508327549

393738373530383332373534 39a00000000000000000000000000
       name          id            price
       ------------------------------------------------------------
       linux        9787508327549    48


shell>
```

输入 1,查看记录.

```
shell> 1

       name          id            price
       ------------------------------------------------------------
       ibm           277325073             483
       sun           433443         400
       linux          9787508327549
              48
```

2.7 微型打印机模块测试:

实验目录为:/root/modules/test_miniprint,首先确保微型打印机模块的串口连接到开发板上方的串口 1 上。

注意:微型打印机波特率设置 119200(跳线设置参考微打使用手册)

进入 /root/modules/test_miniprint 目录,执行测试脚本./test.sh,会显示如下数据,如下图所示。

```
up-tech:~/test_modules #cd test_miniprint/
up-tech:~/test_modules/test_miniprint #./test.sh
insmod: cannot insert 's3c2440-tty.ko': File exists
selected clock c0417158 (pclk) quot 54, calc 57528
selected clock c0417158 (pclk) quot 164, calc 19176
selected clock c0417158 (pclk) quot 164, calc 19176

<miniprint control shell>
 [1]  test the miniprint
 [**] help menu
 [0]  print the end of the context an exit
shell>
```

输入 1，打印机打印出“博创科技 嵌入互动”的字样。

2.8 射频模块测试：

实验目录为：/root/modules/test_rfid，首先确保射频模块连接到开发板下方 168 针插槽上。

进入 /root/modules/test_rfid 目录，执行测试脚本./test.sh，会显示如下数据，如下图所示。

```
up-tech:~/test_modules #cd test_rfid/
up-tech:~/test_modules/test_rfid #./test.sh
insmod: cannot insert 's3c2440-tty2.ko': File exists
  Default baudrate is 38400 bps. If not, please enter the baudratselected clock c0417158
(pclk) quot 658, calc 4801
selected clock c0417158 (pclk) quot 81, calc 38586
e as a parameter

<ICcardReader control shell>
 [1]  detect ICCard mode.
 [2]  get ICCard status.
 [3]  read default data
 [4]  write default data
 [5]  read  data from the block which you choose
 [6]  write data  to  the block which you choose
 [0]  exit.
 [**] help menu.
keyshell>
```

按照菜单，输入 1 可选择搜索 IC 卡模式，将 IC 卡靠近射频模块，即可进行搜索。

```
keyshell>$: 1
bb ff 1 7 0 4 7 6e ec e8 0 0 1c 0 1 0 54 1c 8 0 0 0 0 0 1 0
Find a ICcard!
The ICcard NO.: 7 6e ec e8
waiting for your next command!
keyshell>
```

2.9 传感器模块测试：

实验目录为：/root/modules/test_sensor，首先确保传感器模块连接到开发板下方

168 针插槽上。

进入 /root/modules/test_sensor 目录，执行测试脚本./test.sh，会显示如下数据，如下图所示。

```
up-tech:~/test_modules #cd test_sensor/
up-tech:~/test_modules/test_sensor #./test.sh
insmod: cannot insert 's3c2440-tty2.ko': File exists
selected clock c0417158 (pclk) quot 81, calc 38586
selected clock c0417158 (pclk) quot 26, calc 117187

<sensor_board control shell>
  [10] ~ [20] enter command
  [00] exit
shell>
```

这里提示输入命令[10]~[20]，具体的命令的意义请参考本文的串口协议部分，这里执行其中一个作为演示

```
<sensor_board control shell>
  [10] ~ [20] enter command
  [00] exit
shell>10
we get cmd:0x10
we get data: 2 1 0
```

获取数据为 26.0 表示的是当前的温度值

2.10 USB2.0 扩展模块测试：

实验目录为：/root/modules/test_usb2.0，首先确保 USB2.0 扩展模块连接到开发板下方 168 针插槽上。

进入 /root/modules/test_usb2.0 目录，执行测试脚本./test.sh，会显示如下数据，如下图所示。

```
up-tech:~/test_modules #cd test_usb2.0/
up-tech:~/test_modules/test_usb2.0 #ls
isp116x-hcd.ko  test.sh          uImage
up-tech:~/test_modules/test_usb2.0 #./test.sh
116x: driver isp116x-hcd, 03 Nov 2005
up-tech:~/test_modules/test_usb2.0 #
```

将 U 盘插入 USB2.0 扩展模块的 USB 接口，会打印如下语句：

```
up-tech:/mnt/nfs/SRC/exp/module/usb2.0/usb2.0 # usb 2-1: new full speed USB
device using isp116x-hcd and address 2
the USB device you want to use  needs 50 MA
the power of the USB interface avaliable is 500 MA
usb 2-1: configuration #1 chosen from 1 choice
scsi3 : SCSI emulation for USB Mass Storage devices
usb 2-1: Product: DataTraveler 2.00000000437
usb 2-1: Manufacturer: Kingston
usb 2-1: SerialNumber: 0000000437
scsi 3:0:0:0: Direct-Access     Kingston DataTraveler 2.0 1.00 PQ: 0 ANSI: 2
sd 3:0:0:0: [sda] 984576 512-byte hardware sectors (504 MB)
sd 3:0:0:0: [sda] Write Protect is off
sd 3:0:0:0: [sda] Assuming drive cache: write through
sd 3:0:0:0: [sda] 984576 512-byte hardware sectors (504 MB)
sd 3:0:0:0: [sda] Write Protect is off
sd 3:0:0:0: [sda] Assuming drive cache: write through
 sda:<7>usb-storage: queuecommand called
 sda4
sd 3:0:0:0: [sda] Attached SCSI removable disk
sd 3:0:0:0: Attached scsi generic sg0 type 0

up-tech:/mnt/nfs/SRC/exp/module/usb2.0/usb2.0 #usb 2-2: new full speed USB
device using isp116x-hcd and address 3
the USB device you want to use  needs 50 MA
the power of the USB interface avaliable is 500 MA
usb 2-2: configuration #1 chosen from 1 choice
scsi4 : SCSI emulation for USB Mass Storage devices
usb 2-2: Product: DataTraveler 2.00000000437
usb 2-2: Manufacturer: Kingston
usb 2-2: SerialNumber: 0000000437
scsi 4:0:0:0: Direct-Access     Kingston DataTraveler 2.0 1.00 PQ: 0 ANSI: 2
sd 4:0:0:0: [sda] 984576 512-byte hardware sectors (504 MB)
sd 4:0:0:0: [sda] Write Protect is off
sd 4:0:0:0: [sda] Assuming drive cache: write through
sd 4:0:0:0: [sda] 984576 512-byte hardware sectors (504 MB)
sd 4:0:0:0: [sda] Write Protect is off
sd 4:0:0:0: [sda] Assuming drive cache: write through
 sda:<7>usb-storage: queuecommand called
 sda4
sd 4:0:0:0: [sda] Attached SCSI removable disk
sd 4:0:0:0: Attached scsi generic sg0 type 0
```

2.11zigbee 协调器模块测试：

实验目录为：/root/modules/test_zigbee，首先确保 zigbee 协调器模块的串口连接到开发板串口 1 上。

进入 /root/modules/test_zigbee 目录，执行测试脚本./test.sh，会显示如下数据，如下图所示。

```
up-tech:~/test_modules #cd test_zigbee/
up-tech:~/test_modules/test_zigbee #./test.sh
insmod: cannot insert 's3c2440-tty.ko': File exists
<7>selected clock c0417158 (pclk) quot 164, calc 19176
selected clock c0417158 (pclk) quot 26, calc 117187

<zigbee ctrl remote control shell>
 [1]  open      led1
 [2]  close     led1
 [3]  open      led2
 [4]  close     led2
 [**] help      menu
 [0]  exit
shell>
```

输入 1,zigbee 节点模块上 sensor board LED1 亮；输入 2，zigbee 节点模块上 sensor board LED1 灭；同理可控制 LED2 的亮与灭.

```
<zigbee ctrl remote control shell>
 [1]  open     led1
 [2]  close    led1
 [3]  open     led2
 [4]  close    led2
 [**] help     menu
 [0]  exit
shell> 1

we get cmd 1
shell> 2

we get cmd 2
shell> 3

we get cmd 3
shell> 4

we get cmd 4
shell>
```

3、 QT 演示程序

```
up-tech:~ #ls
lost+found    system        test_QT       test_basic    test_modules
up-tech:~ #cd test_QT/
up-tech:~/test_QT #ls
Qt-embedded-4.4.0  app                s3c2440_ts.ko      test.sh
up-tech:~/test_QT #
```

直接执行测试脚本，如下所示：

```
up-tech:~/test_QT #./test.sh
aiting for initializing...
tmp is 0xaa
tmpnew is :0xcc
s3c2440-dc-motor       device initialized
s3c2440_led    device initialized
s3c2440-da     device initialized
<7>dma10: s3c2410_request_dma: client=I2SSDO, dev=00000000
mapped channel 10 to 2
dma2: s3c2410_dma_request:757: ls=0, cur=00000000, 00000000 00000000
dma2: s3c2410_dma_request:757: DCSRC=00000000, DISRC=00000000, DSTAT=00000000
DMT=00, DCON=00000000
dma10: s3c2410_dma_request : requesting irq 35
s3c2410_dma_request: channel initialised, c044254c
s3c2410_dma_devconfig: source=1, hwcfg=00000003, devaddr=55000010
s3c2410_dma_devconfig: mem source, devaddr=55000010, hwcfg=3
s3c2410_dma_config: chan=10, xfer_unit=2, dcon=a0800000
s3c2410_dma_config: Initial dcon is a0800000
s3c2410_dma_config: New dcon is a0800000
s3c2410_dma_config: dcon now a0900000
s3c2410_dma_set_buffdone_fn: chan=c044254c, callback rtn=bf024394
s3c2410_dma_setflags: chan=c044254c, flags=00000002
dma9: s3c2410_request_dma: client=I2SSDI, dev=00000000
mapped channel 9 to 1
dma1: s3c2410_dma_request:757: ls=0, cur=00000000, 00000000 00000000
dma1: s3c2410_dma_request:757: DCSRC=00000000, DISRC=00000000, DSTAT=00000000
DMT=00, DCON=00000000
dma9: s3c2410_dma_request : requesting irq 34
s3c2410_dma_request: channel initialised, c04424c0
s3c2410_dma_devconfig: source=0, hwcfg=00000003, devaddr=55000010
s3c2410_dma_devconfig: hw source, devaddr=55000010, hwcfg=3
s3c2410_dma_config: chan=9, xfer_unit=2, dcon=a2900000
s3c2410_dma_config: Initial dcon is a2900000
```

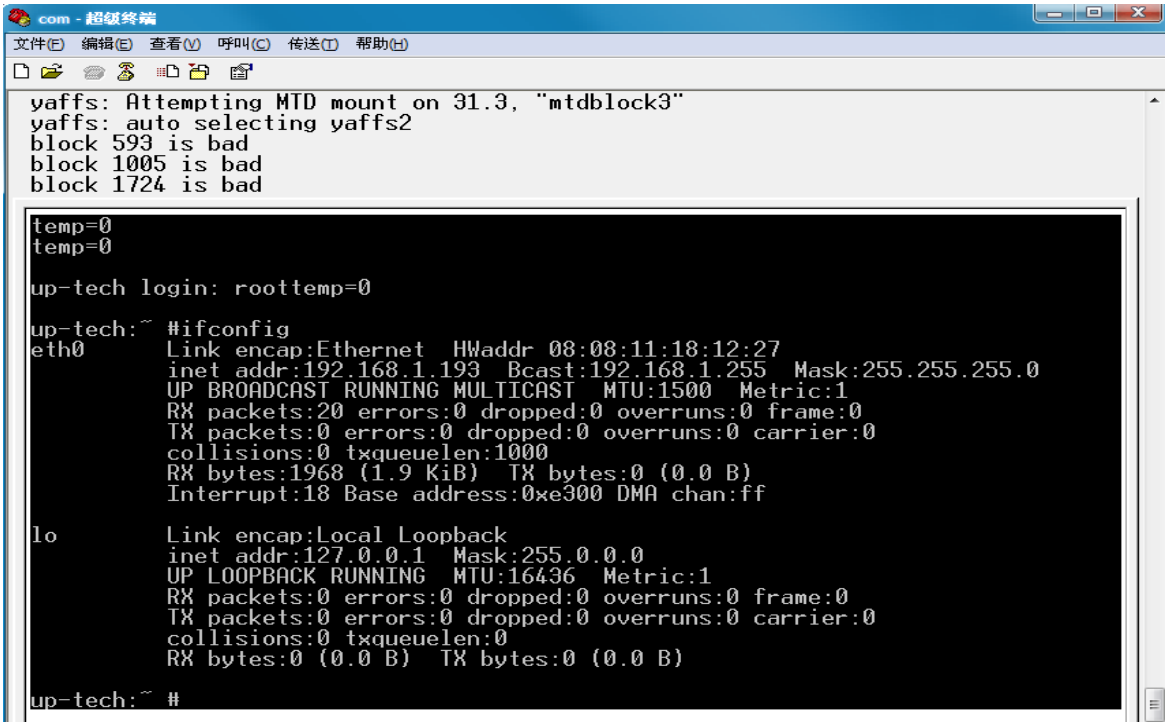
```
s3c2410_dma_config: New dcon is a2900000
s3c2410_dma_config: dcon now a2900000
s3c2410_dma_set_buffdone_fn: chan=c04424c0, callback rtn=bf0243e0
s3c2410_dma_setflags: chan=c04424c0, flags=00000002
UDA1341 audio driver initialized
set environment...
run qt process ING...
```

会出现相应的 QT 演示界面，可逐个点击进行演示。

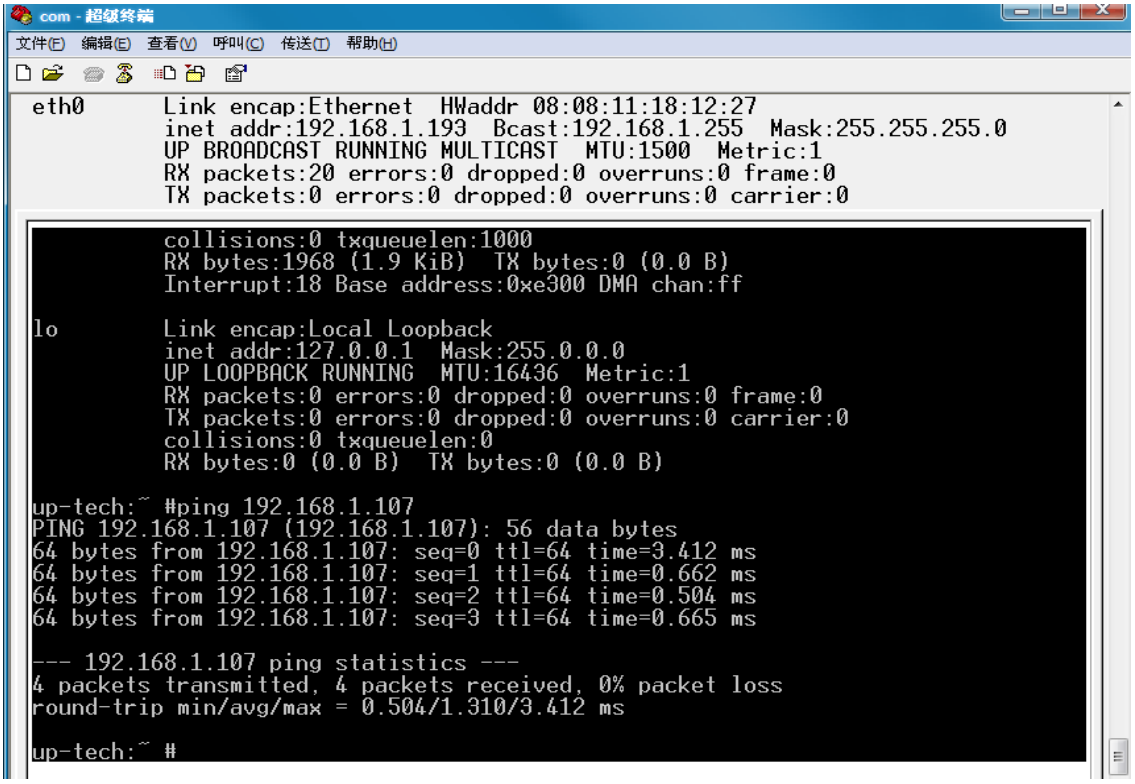
4、 其他

4.1、测试网络

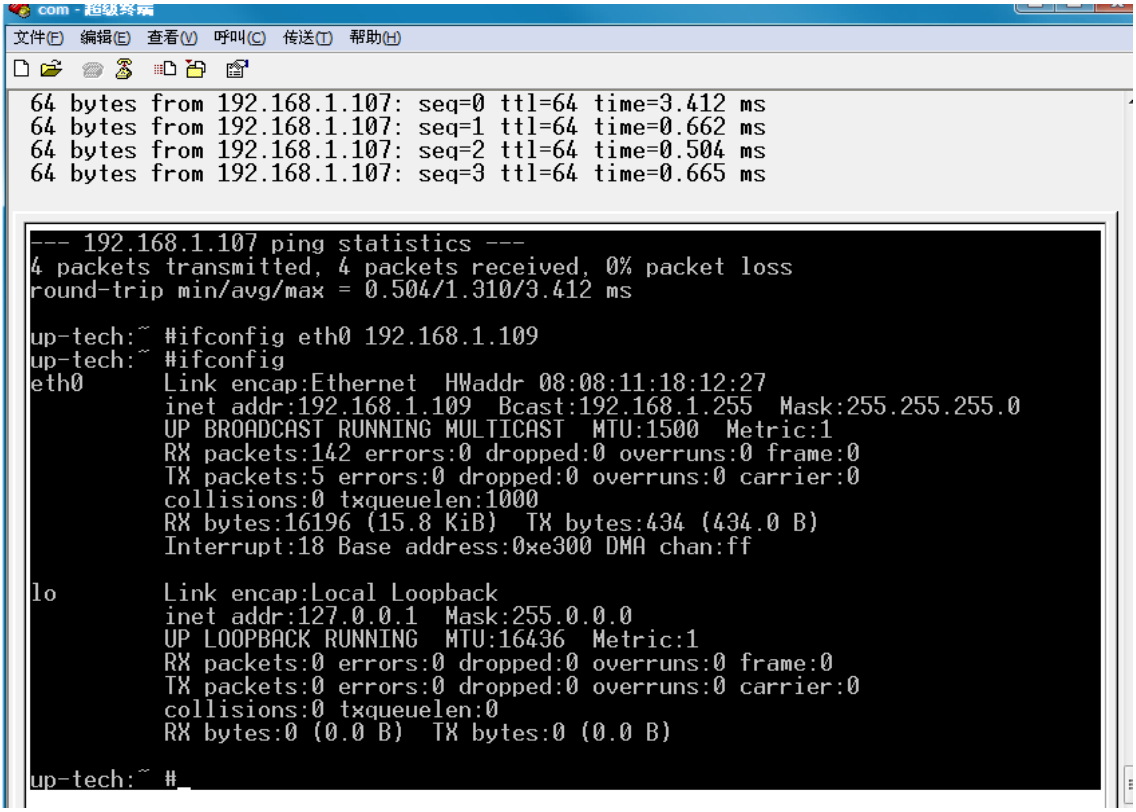
首先配置 IP 地址。输入 ifconfig 查看 IP，如下图。



测试网络连接，输入 ping 192.168.1.107，IP 为 PC 机 IP，如下图。注意要保证开发板与 PC 机的 IP 在同一网段，同时要关闭各类防火墙。



如果您想改动 IP，可以使用 ifconfig eth0 192.168.1.109，eth0 后为您想要的 IP。查看一下，如下图



4.6、测试触摸屏

进入/root/test_QT 目录，如图 38，执行脚本./test_touchscreen.sh
会出现矫屏程序，按照所示的“+”，给予触摸屏矫正。
up-tech:~/test_QT #./test_touchscreen.sh

```
xres = 320, yres = 240
Took 22 samples...
Top left : X =  810 Y =  778
Took 17 samples...
Top right : X =  148 Y =  774
Took 1 samples...
Bot right : X =   85 Y =  792
Took 10 samples...
Bot left : X =  146 Y =  238
```