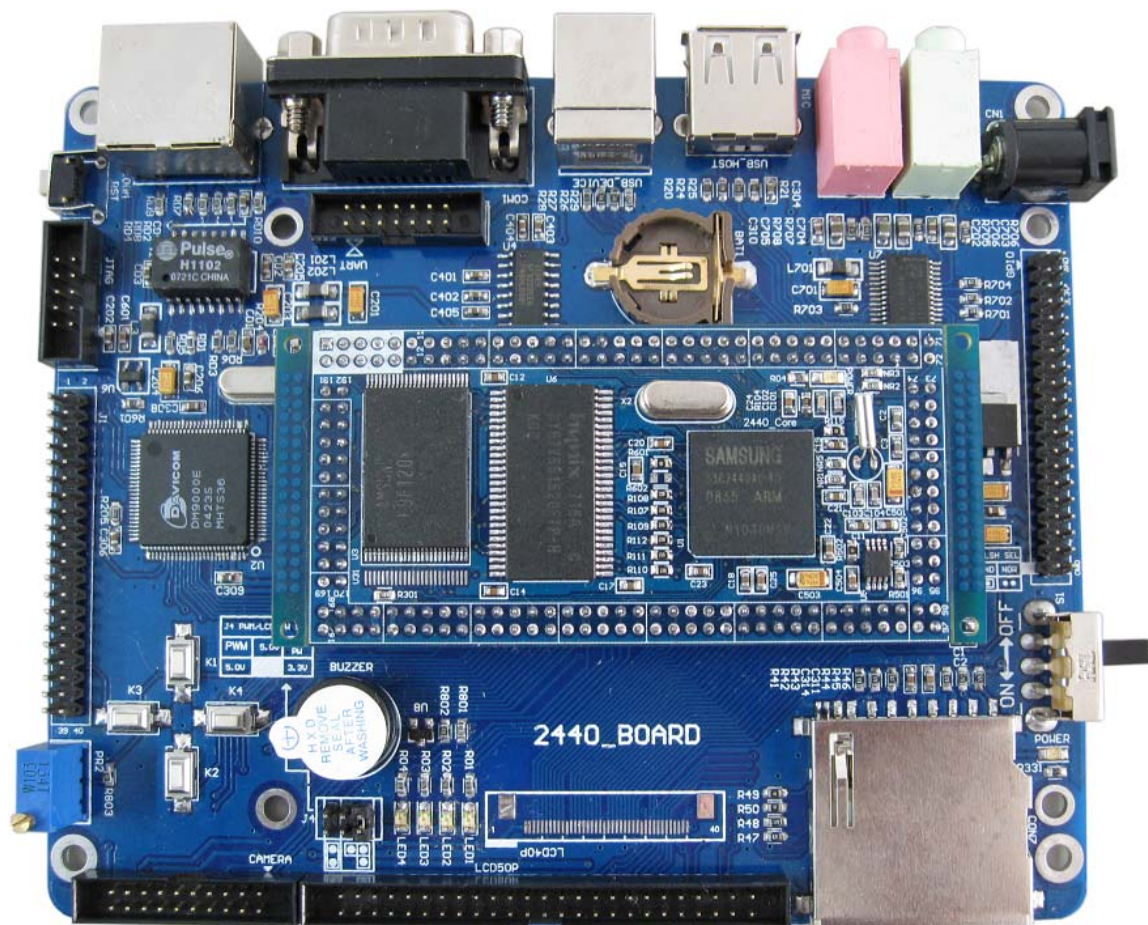


## EM2440-III

### Linux Development Manual



## **1. Introduction**

### **1.1. About this Manual**

This manual is intended to provide the user with an overview of the board and benefits, complete features specifications, and set up procedures. It contains important safety information as well.

### **1.2. Feedback and Update to this Manual**

To help our customers make the most of our products, we are continually making additional and updated resources available on the Boardcon website ([www.armdesigner.com](http://www.armdesigner.com)).

These include manuals, application notes, programming examples, and updated software and hardware. Check in periodically to see what's new!

When we are prioritizing work on these updated resources, feedback from customers is the number one influence. If you have questions, comments, or concerns about your product or project, please do not hesitate to contact us at [support@armdesigner.com](mailto:support@armdesigner.com).

### **1.3. Limited Warranty**

Boardcon warrants this product to be free of defects in material and workmanship for a period of one year from date of buy. During this warranty period Boardcon will repair or replace the defective unit in accordance with the following process:

A copy of the original invoice must be included when returning the defective unit to Boardcon. This limited warranty does not cover damages resulting from lightning or other power surges, misuse, abuse, abnormal conditions of operation, or attempts to alter or modify the function of the product.

This warranty is limited to the repair or replacement of the defective unit. In no event shall Boardcon be liable or responsible for any loss or damages, including but not limited to any lost profits, incidental or consequential damages, loss of business, or anticipatory profits arising from the use or inability to use this products.

Repairs made after the expiration of the warranty period are subject to a repair charge and the cost of return shipping. Please contact Boardcon to arrange for any repair service and to obtain repair charge information.

<b>CHAPTER 1 BUILD CROSS-COMPILE ENVIRONMENT .....</b>	<b>4</b>
<b>CHAPTER 2 COMPILING U-BOOT .....</b>	<b>6</b>
2.1 Decompressing u-boot.....	6
2.2 Customizing u-boot.....	6
2.3 Configuring u-boot.....	7
2.4 Compiling u-boot.....	7
<b>CHAPTER 3 COMPILING LINUX-2.6 KERNEL .....</b>	<b>9</b>
<b>CHAPTER 4 CUSTOM YOUR ROOT FILE SYSTEM.....</b>	<b>15</b>
<b>CHAPTER 5 LINUX EXPERIMENT .....</b>	<b>17</b>
5.1 Experiment of program termination.....	17
5.2 Experiment of program auto-run configuration.....	17

## Chapter 1 Build Cross-compile Environment

The cross\_compiler is the main environment under Linux. We introduce a process building a development environment which can compile arm-linux kernel, driver and application under RedHat 9.0.

Copy the compression package "EABI\_4.3.3\_EmbedSky\_20090812.tar.bz2" from the directory "Linux" to the directory "/opt/Boardcon/" in Linux system, and decompress the package under the current directory: ( the following commands are executed in PC ).

```
#cd /opt/Boardcon
```

```
#tar xvfj EABI_4.3.3_EmbedSky_20090812.tar.bz2 -C /
```

After the upper operation, the compiler has been installed under the directory "4.3.3" and "crosstool\_3.4.5\_softfloat" of the directory of "/opt/Boardcon". The mkyaff2image will be installed automatically under the directory "/usr/sbin/":

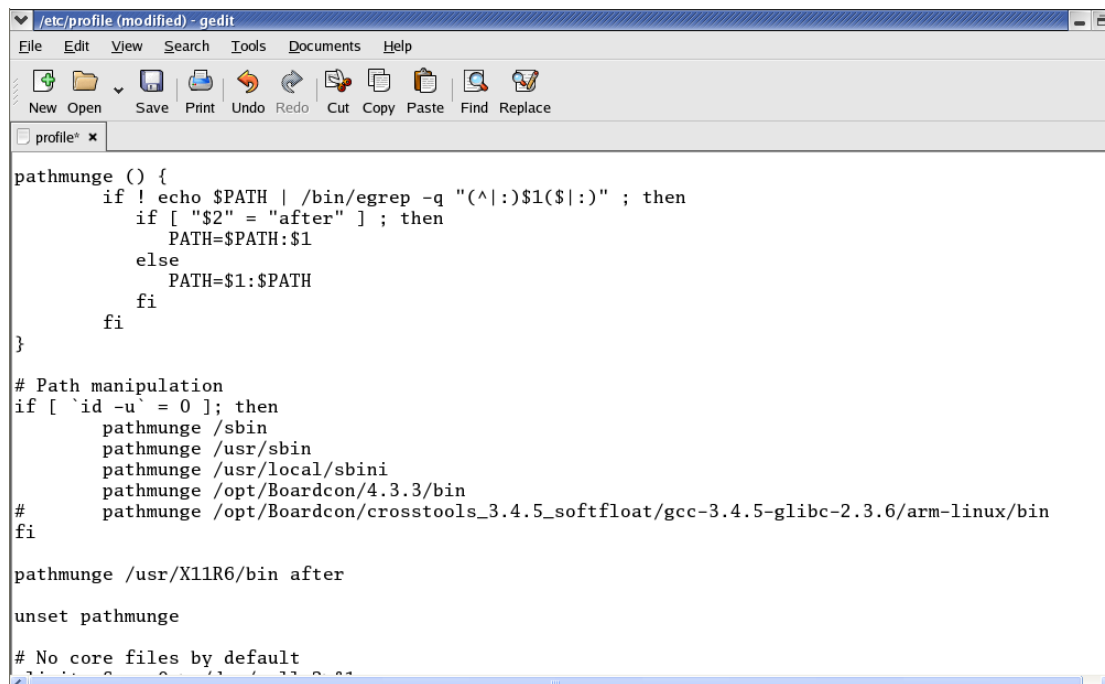
The cross-compile compiler of 4.3.3 is used to compile kernel, and application. The cross-compile compiler of crosstool\_3.4.5\_softfloat is used to compile the u-boot.

Execute the command:

```
#gedit /etc/profile
```

Add the following information in "profile":

(the following frame contains the added information. If the user tries to use the cross-compiler of a certain version, please remove its prefix "#", and add "#" to the head of other versions. The lines highlighted with blue underline as following diagram is required to be added the prefix "#")



```
pathmunge () {
    if ! echo $PATH | /bin/egrep -q "^(^|:)$1($|:)" ; then
        if [ "$2" = "after" ] ; then
            PATH=$PATH:$1
        else
            PATH=$1:$PATH
        fi
    fi
}

# Path manipulation
if [ `id -u` = 0 ]; then
    pathmunge /sbin
    pathmunge /usr/sbin
    pathmunge /usr/local/sbin
    pathmunge /opt/Boardcon/4.3.3/bin
    pathmunge /opt/Boardcon/crosstools_3.4.5_softfloat/gcc-3.4.5-glibc-2.3.6/arm-linux/bin
fi

pathmunge /usr/X11R6/bin after

unset pathmunge

# No core files by default
```

Save "profile" and execute the following command to set a default cross-compiler:

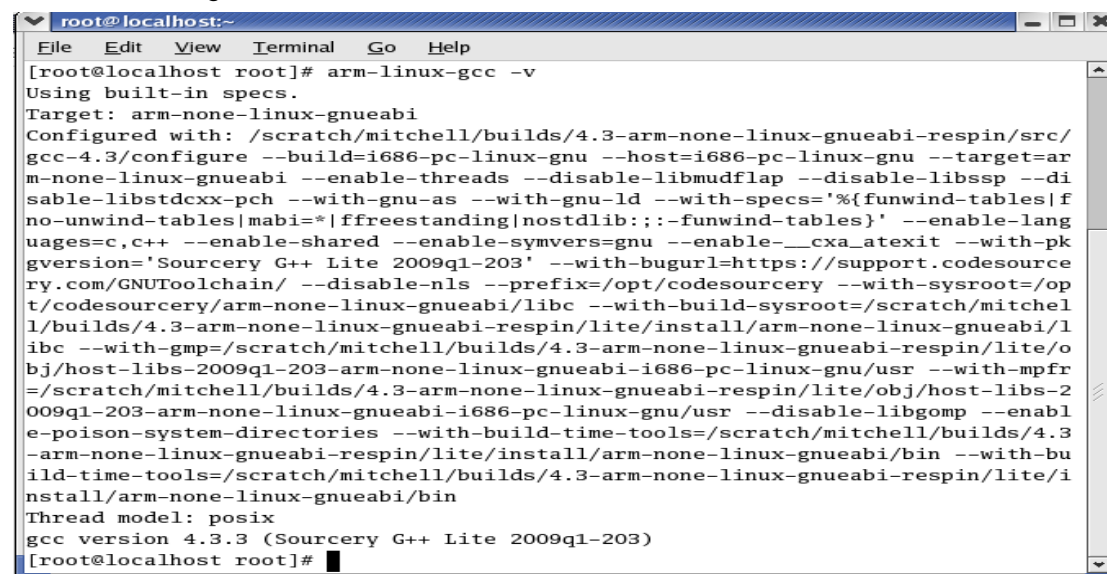
```
#source /etc/profile
```

Execute the following command to check if the cross-compiler has been installed

successfully and check the revised version:

**#arm-linux-gcc -v**

Get the following information:



```
root@localhost:~  
File Edit View Terminal Go Help  
[root@localhost root]# arm-linux-gcc -v  
Using built-in specs.  
Target: arm-none-linux-gnueabi  
Configured with: /scratch/mitchell/builds/4.3-arm-none-linux-gnueabi-respin/src/  
gcc-4.3/configure --build=i686-pc-linux-gnu --host=i686-pc-linux-gnu --target=ar  
m-none-linux-gnueabi --enable-threads --disable-libmudflap --disable-libssp --di  
sable-libstdcxx-pch --with-gnu-as --with-gnu-ld --with-specs='%{funwind-tables|f  
no-unwind-tables|mabi=*|ffreestanding|nostdlib:::-funwind-tables}' --enable-lang  
uages=c,c++ --enable-shared --enable-symvers=gnu --enable-__cxa_atexit --with-pk  
gversion='Sourcery G++ Lite 2009q1-203' --with-bugurl=https://support.codesourc  
ery.com/GNUToolchain/ --disable-nls --prefix=/opt/codesourcery --with-sysroot=/op  
t/codesourcery/arm-none-linux-gnueabi/libc --with-build-sysroot=/scratch/mitche  
ll/builds/4.3-arm-none-linux-gnueabi-respin/lite/install/arm-none-linux-gnueabi/l  
ibc --with-gmp=/scratch/mitchell/builds/4.3-arm-none-linux-gnueabi-respin/lite/o  
bj/host-libs-2009q1-203-arm-none-linux-gnueabi-i686-pc-linux-gnu/usr --with-mpfr  
=/scratch/mitchell/builds/4.3-arm-none-linux-gnueabi-respin/lite/obj/host-libs-2  
009q1-203-arm-none-linux-gnueabi-i686-pc-linux-gnu/usr --disable-libgomp --enabl  
e-poison-system-directories --with-build-time-tools=/scratch/mitchell/builds/4.3  
-arm-none-linux-gnueabi-respin/lite/install/arm-none-linux-gnueabi/bin --with-bu  
ild-time-tools=/scratch/mitchell/builds/4.3-arm-none-linux-gnueabi-respin/lite/i  
ninstall/arm-none-linux-gnueabi/bin  
Thread model: posix  
gcc version 4.3.3 (Sourcery G++ Lite 2009q1-203)  
[root@localhost root]#
```

The cross-compiler version might be frequently changed. Use the previous command “gedit /etc/profile” to modify the file “/etc/profile”, and use it to validate “source /etc/profile”. Then execute the command “arm-linux-gcc -v” to check the revised version of cross-compiler.

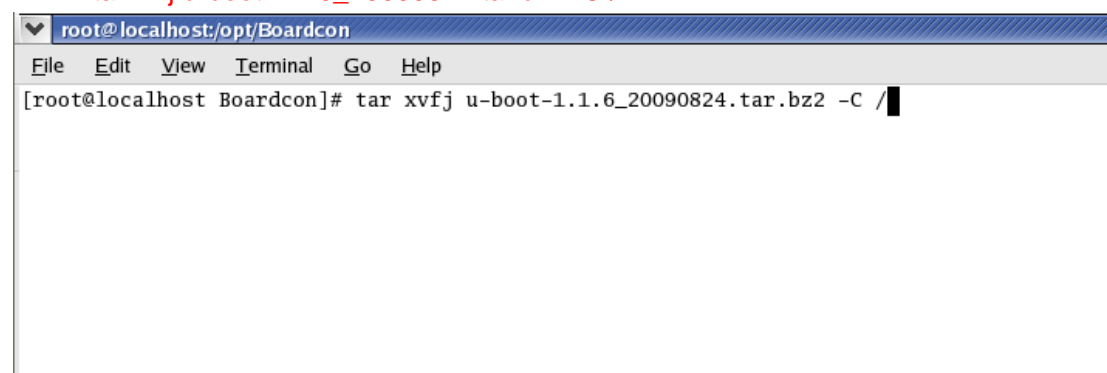
## Chapter 2 Compiling u-boot

The file “u-boot-1.1.6\_20090824.tar.bz2” in CD-ROM is the source code package of u-boot for EM2440-III. Make sure the Cross-compiler of crosstools\_3.4.5\_softfloat is installed properly.

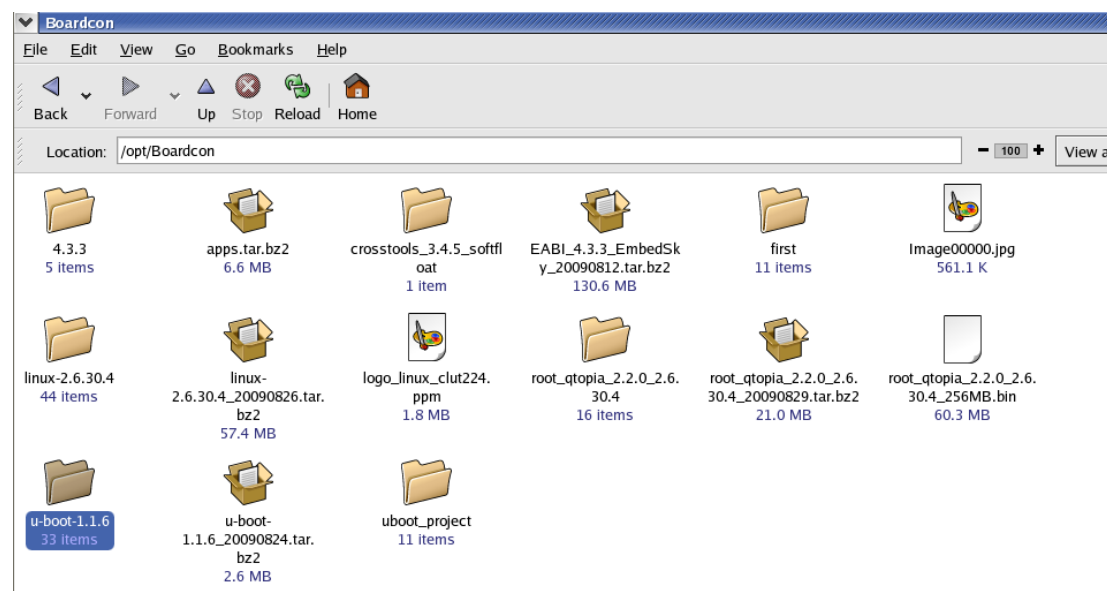
### 2.1 Decompressing u-boot

Execute the decompression command in PC:

```
# tar xvfj u-boot-1.1.6_20090824.tar.bz2 -C /
```



The file is auto-decompressed to the directory “/opt/Boardcon/u-boot-1.1.6”:



### 2.2 Customizing u-boot

(1) Support 3.5 inch LCD:

Open the file \u-boot-1.1.6\include\configs\ EmbedSky.h, and modify the program in the 112 line as follows:

```
#define LCD_TFT W35
```

(2) Support 7.0 inch LCD:

Open the file \u-boot-1.1.6\include\configs\ EmbedSky.h, and modify the program in the 112 line as follows:

```
#define LCD_TFT TFT800480
```

Normally, you just need modify the program according to the size of your LCD.

## 2.3 Configuring u-boot

Execute the command:

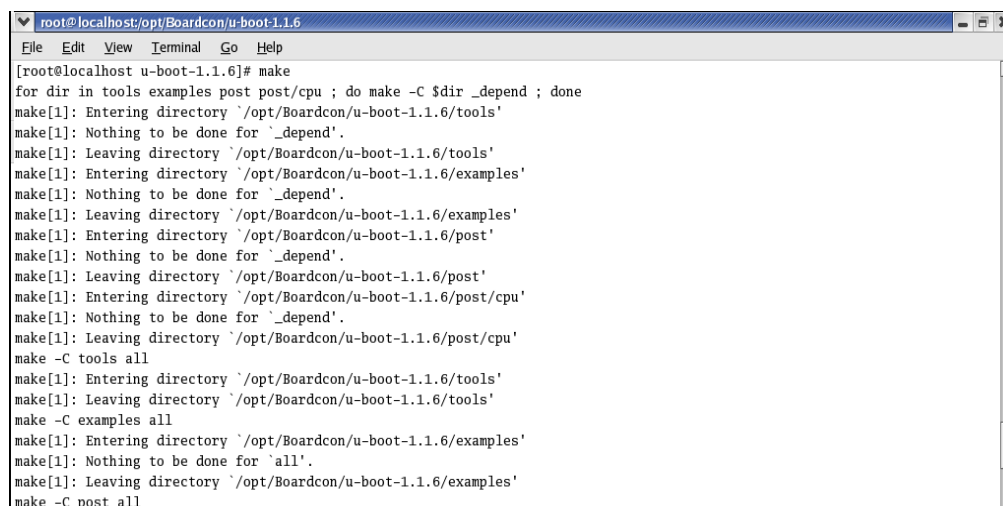
```
# make EmbedSky_config
```



## 2.4 Compiling u-boot

Execute the command:

```
# make
```



Compiling is complete, and the file u-boot.bin under current directory is the image to be burnt into the board.



```

root@localhost:opt/Boardcon/u-boot-1.1.6
File Edit View Terminal Go Help
r - miiphyutil.o
r - s_record.o
r - serial.o
r - soft_i2c.o
r - soft_spi.o
r - spartan2.o
r - spartan3.o
r - usb.o
r - usb_kbd.o
r - usb_storage.o
r - virtex2.o
r - xilinx.o
r - crc16.o
r - xyzModem.o
r - cmd_mac.o
r - ../modules/bmp2rgb.module
make[1]: Leaving directory `/opt/Boardcon/u-boot-1.1.6/common'
UNDEF_SYM=`/opt/Boardcon/crosstools_3.4.5_softfloat/gcc-3.4.5-glibc-2.3.6/arm-linux/bin/arm-linux-objdump -x lib_generic/libg
eneric.a board/EmbedSky/libEmbedSky.a cpu/arm920t/libarm920t.a cpu/arm920t/s3c24x0/libs3c24x0.a lib_arm/libarm.a fs/cramfs/li
bcramfs.a fs/jffs2/libjffs2.a net/libnet.a rtc/librtc.a drivers/libdrivers.a drivers/nand/libnand.a drivers/nand_legacy/libna
nd_legacy.a drivers/lcd/liblcd.a modules/usb.module modules/wince.module common/libcommon.a |sed -n -e 's/^(__u_boot_cmd_
*\)/-u\1/p'|sort|uniq`; \
    cd /opt/Boardcon/u-boot-1.1.6 && /opt/Boardcon/crosstools_3.4.5_softfloat/gcc-3.4.5-glibc-2.3.6/arm-linux/bin/arm-lin
ux-ld -Bstatic -T /opt/Boardcon/u-boot-1.1.6/board/EmbedSky/u-boot.lds -Ttext 0x3D80000 $UNDEF_SYM cpu/arm920t/start.o \
--start-group lib_generic/libgeneric.a board/EmbedSky/libEmbedSky.a cpu/arm920t/libarm920t.a cpu/arm920t/s3c2
4x0/libs3c24x0.a lib_arm/libarm.a fs/cramfs/libcramfs.a fs/jffs2/libjffs2.a net/libnet.a rtc/librtc.a drivers/libdrivers.a dr
ivers/nand/libnand.a drivers/nand_legacy/libnand_legacy.a drivers/lcd/liblcd.a modules/usb.module modules/wince.module common
/libcommon.a --end-group -L /opt/Boardcon/crosstools_3.4.5_softfloat/gcc-3.4.5-glibc-2.3.6/arm-linux/bin/../lib/gcc/arm-linux
/3.4.5 -lgcc \
    -Map u-boot.map -o u-boot
/opt/Boardcon/crosstools_3.4.5_softfloat/gcc-3.4.5-glibc-2.3.6/arm-linux/bin/arm-linux-objcopy --gap-fill=0xff -O srec u-boot
u-boot.srec
/opt/Boardcon/crosstools_3.4.5_softfloat/gcc-3.4.5-glibc-2.3.6/arm-linux/bin/arm-linux-objcopy --gap-fill=0xff -O binary u-bo
ot u-boot.bin
[root@localhost u-boot-1.1.6]#

```



## Chapter 3 Compiling Linux-2.6 Kernel

Caution: The cross-compiler of 4.3.3 is needed when compiling Linux kernel of version 2.6. Make sure the compiler of correct version has been installed successfully.

```
[root@localhost Boardcon]# arm-linux-gcc -v
Using built-in specs.
Target: arm-none-linux-gnueabi
Configured with: /scratch/mitchell/builds/4.3-arm-none-linux-gnueabi-respin/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-lin
tdxch-pch --with-gnu-as --with-gnu-ld --with-specs='%{funwind-tables|fno-unwind-tables|abi=*|fprestanding|nostdlib:::funwi
nd-tables}' --enable-languages=c,c++ --enable-shared --enable-symvers=gnu --enable-__cxa_atexit --with-pkgversion='Sourcery G
++ Lite 2009q1-203' --with-bugurl=https://support.codesourcery.com/GNUToolchain/ --disable-nls --prefix=/opt/codesourcery --w
ith-sysroot=/opt/codesourcery/arm-none-linux-gnueabi/libc --with-build-sysroot=/scratch/mitchell/builds/4.3-arm-none-linux-gn
ueabi-respin/lite/install/arm-none-linux-gnueabi/libc --with-gmp=/scratch/mitchell/builds/4.3-arm-none-linux-gnueabi-respin/l
ite/obj/host-libs-2009q1-203-arm-none-linux-gnueabi-686-pc-linux-gnu/usr --with-mpfr=/scratch/mitchell/builds/4.3-arm-none-
linux-gnueabi-respin/lite/obj/host-libs-2009q1-203-arm-none-linux-gnueabi-686-pc-linux-gnu/usr --disable-libgomp --enable-po
ison-system-directories --with-build-time-tools=/scratch/mitchell/builds/4.3-arm-none-linux-gnueabi-respin/lite/install/arm-ne
one-linux-gnueabi/bin --with-build-time-tools=/scratch/mitchell/builds/4.3-arm-none-linux-gnueabi-respin/lite/install/arm-ne
one-linux-gnueabi/bin
Thread model: posix
gcc version 4.3.3 (Sourcery G++ Lite 2009q1-203)
[root@localhost Boardcon]#
```

If the compiler version 4.3.3 has not been successfully installed, please refer to “Chapter 1” to install the compiler.

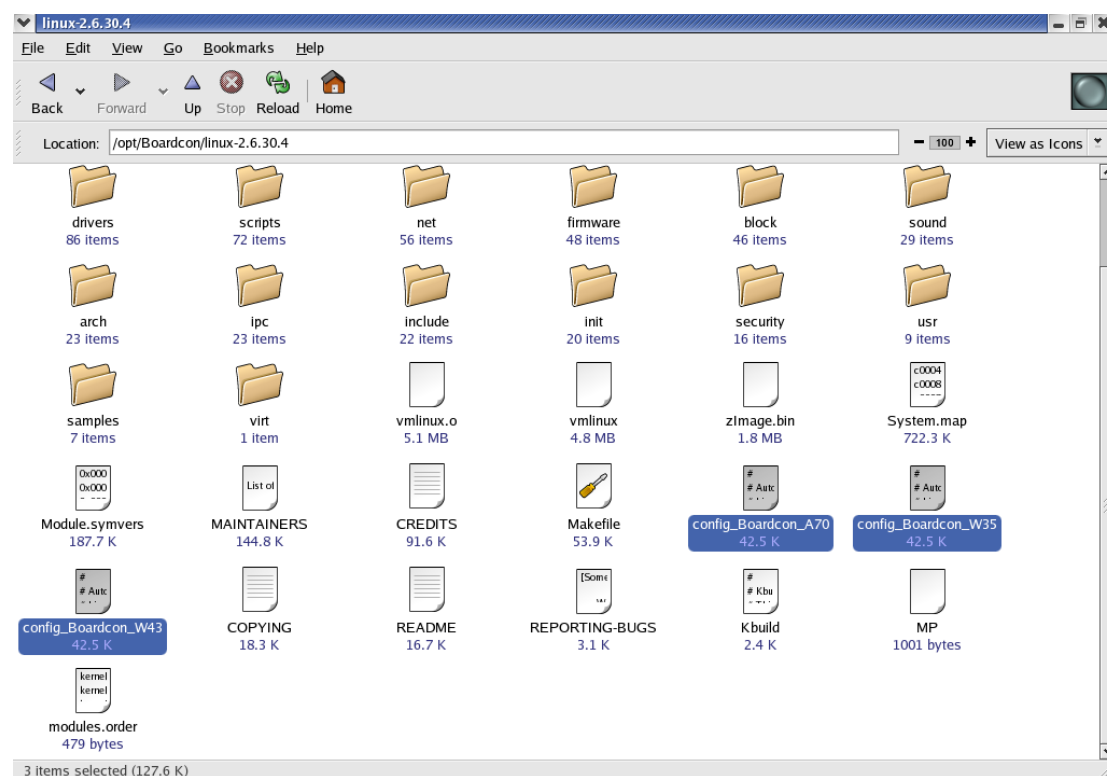
The Linux kernel source code is under the directory "Linux/linux-2.6.30.4\_20090826.tar.bz2". Execute the command `#tar xvfj kernel-2.6.13.tar.bz2 -C /` to decompress it to the directory `/opt/Boardcon/linux-2.6.30.4`:

```
root@localhost:opt/Boardcon
File Edit View Terminal Go Help

[root@localhost Boardcon]# tar xvfj linux-2.6.30.4_20090826.tar.bz2 -C /
opt/Boardcon/linux-2.6.30.4/
opt/Boardcon/linux-2.6.30.4/.gitignore
opt/Boardcon/linux-2.6.30.4/.mailmap
opt/Boardcon/linux-2.6.30.4/arch/
opt/Boardcon/linux-2.6.30.4/arch/.gitignore
opt/Boardcon/linux-2.6.30.4/arch/alpha/
opt/Boardcon/linux-2.6.30.4/arch/alpha/boot/
opt/Boardcon/linux-2.6.30.4/arch/alpha/boot/bootloader.lds
opt/Boardcon/linux-2.6.30.4/arch/alpha/boot/bootp.c
opt/Boardcon/linux-2.6.30.4/arch/alpha/boot/bootpz.c
opt/Boardcon/linux-2.6.30.4/arch/alpha/boot/head.S
opt/Boardcon/linux-2.6.30.4/arch/alpha/boot/main.c
opt/Boardcon/linux-2.6.30.4/arch/alpha/boot/Makefile
opt/Boardcon/linux-2.6.30.4/arch/alpha/boot/misc.c
opt/Boardcon/linux-2.6.30.4/arch/alpha/boot/tools/
opt/Boardcon/linux-2.6.30.4/arch/alpha/boot/tools/mkbb.c
opt/Boardcon/linux-2.6.30.4/arch/alpha/boot/tools/objstrip.c
opt/Boardcon/linux-2.6.30.4/arch/alpha/defconfig
opt/Boardcon/linux-2.6.30.4/arch/alpha/include/
```

The default files are under the decompression directory:

- config\_Boardcon\_W35 the Donghua 3.5 inch LCD default configuration file.
- config\_Boardcon\_A70 is the 7 inch LCD default configuration file.
- config\_Boardcon\_W43 is the Donghua 4.3 inch LCD default configuration

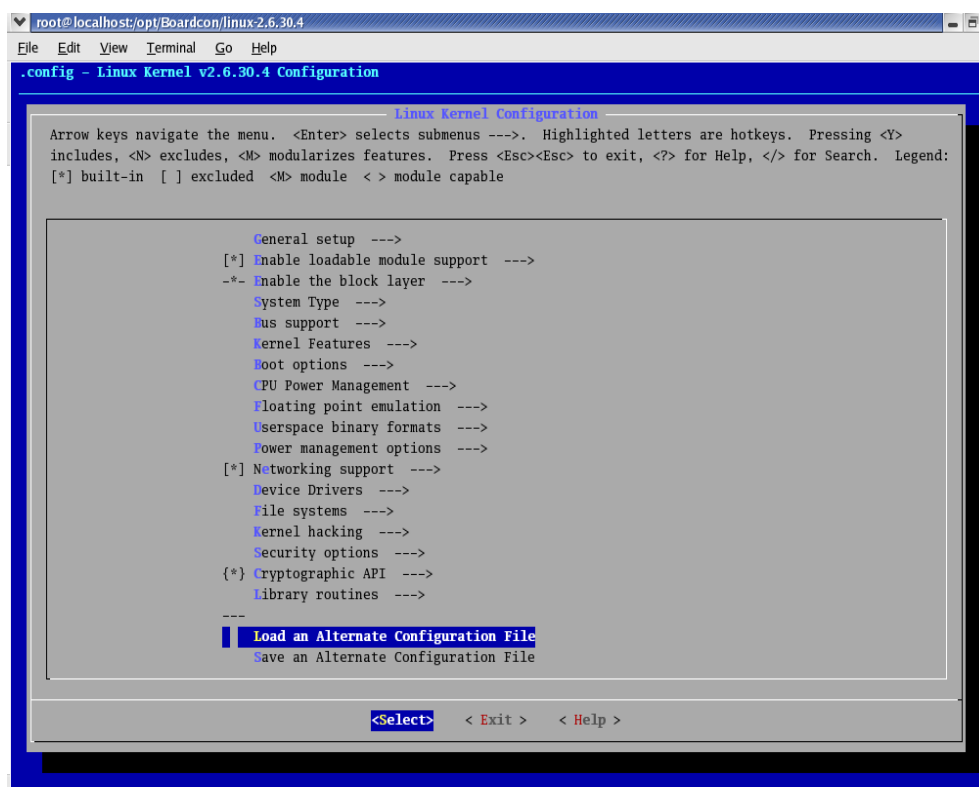


The following steps illustrate the process compiling Linux kernel:

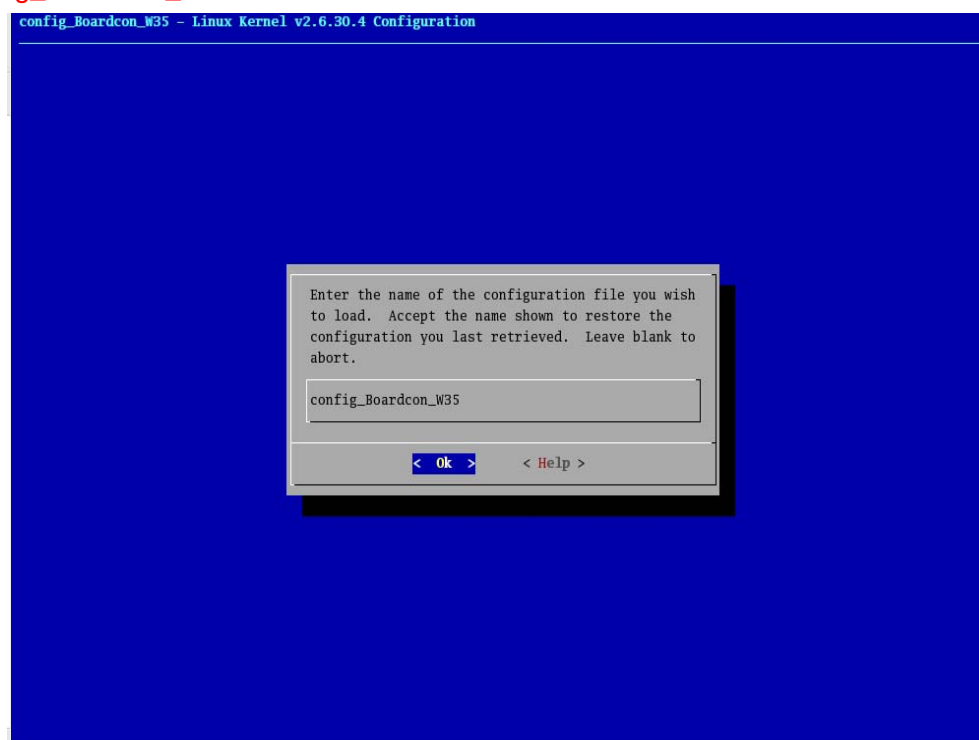
Step1. Input “make menuconfig” to start configuring Linux kernel:

```
root@localhost:/opt/Boardcon/linux-2.6.30.4
File Edit View Terminal Go Help
[root@localhost linux-2.6.30.4]# make menuconfig
HOSTCC scripts/kconfig/lxdialog/checklist.o
HOSTCC scripts/kconfig/lxdialog/inputbox.o
HOSTCC scripts/kconfig/lxdialog/menubox.o
```

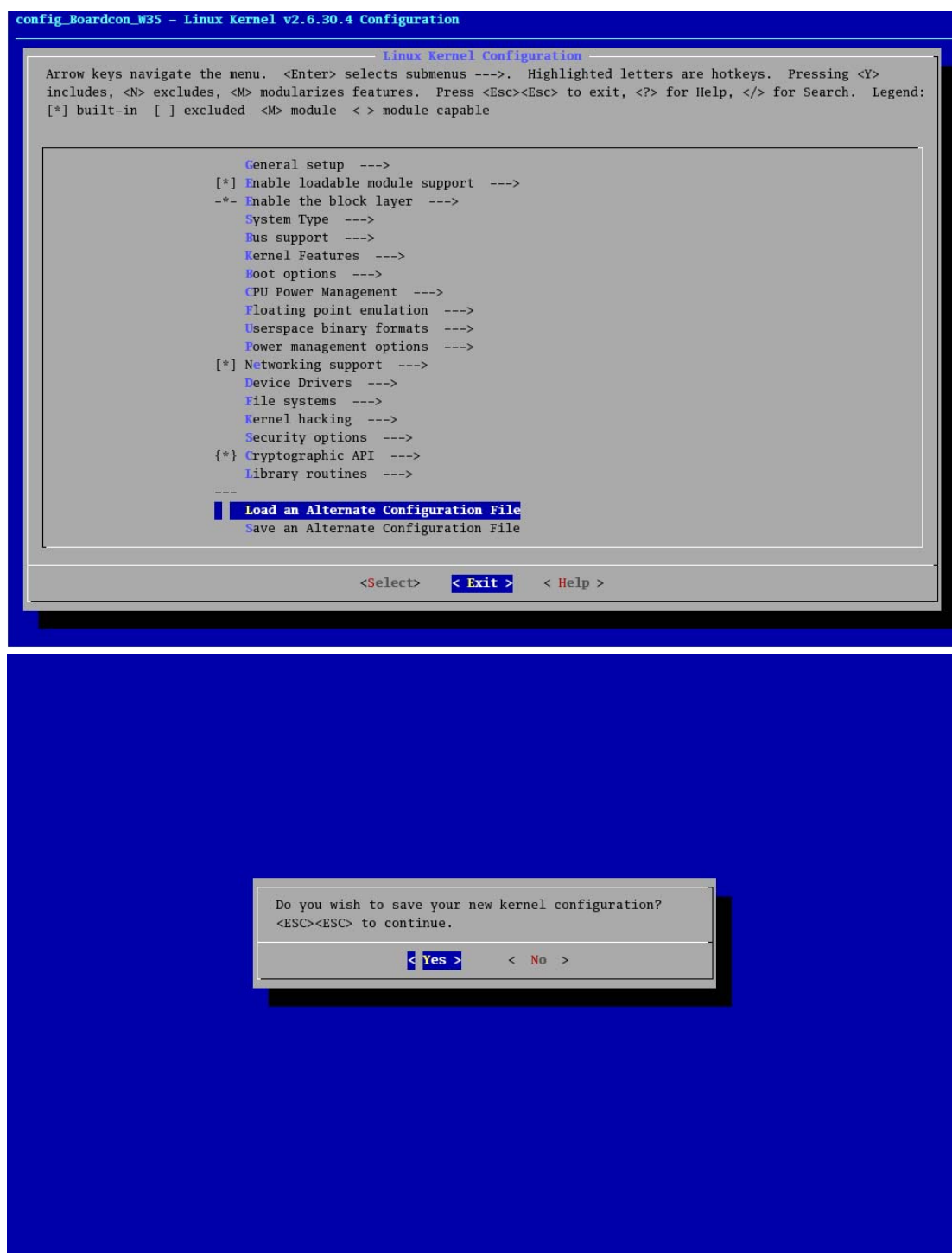
Step2, select “Load an Alternate Configuration File”:



Step3. Input the configuration file name according to the LCD type you are using. Here we use the configuration file of Donghua 3.5 inch LCD for example. Click “OK” after input complete. If you use the 7.0 inch LCD, please select the configuration file “config\_Boardcon\_A70”.



Step4. Go back to main menu. Select “<Exit>” to save the configuration and exit:



Step6. Input “make zImage” and press return-key to start compiling:

```

root@localhost/opt/Boardcon/linux-2.6.30.4
File Edit View Terminal Go Help
HOSTCC scripts/kconfig/ldialog/util.o
HOSTCC scripts/kconfig/ldialog/yesno.o
HOSTCC scripts/kconfig/mconf.o
HOSTLD scripts/kconfig/mconf
scripts/kconfig/mconf arch/arm/Kconfig

*** End of Linux kernel configuration.
*** Execute 'make' to build the kernel or try 'make help'.

[root@localhost linux-2.6.30.4]# make menuconfig
scripts/kconfig/mconf arch/arm/Kconfig
#
# configuration written to config_Boardcon_W35
#

*** End of Linux kernel configuration.
*** Execute 'make' to build the kernel or try 'make help'.

[root@localhost linux-2.6.30.4]# make zImage
make: *** No rule to make target 'zImage'. Stop.
[root@localhost linux-2.6.30.4]# make zImage
CHK include/linux/version.h
make[1]: `include/asm-arm/mach-types.h' is up to date.
CHK include/linux/utsrelease.h
SYMLINK include/asm -> include/asm-arm
CALL scripts/checksyscalls.sh
<stdin>:1097:2: warning: #warning syscall fadvise64 not implemented
<stdin>:1265:2: warning: #warning syscall migrate_pages not implemented
<stdin>:1321:2: warning: #warning syscall pselect6 not implemented
<stdin>:1325:2: warning: #warning syscall ppoll not implemented
<stdin>:1365:2: warning: #warning syscall epoll_pwait not implemented
CHK include/linux/compile.h

```

Step7. The compiling is complete:

```

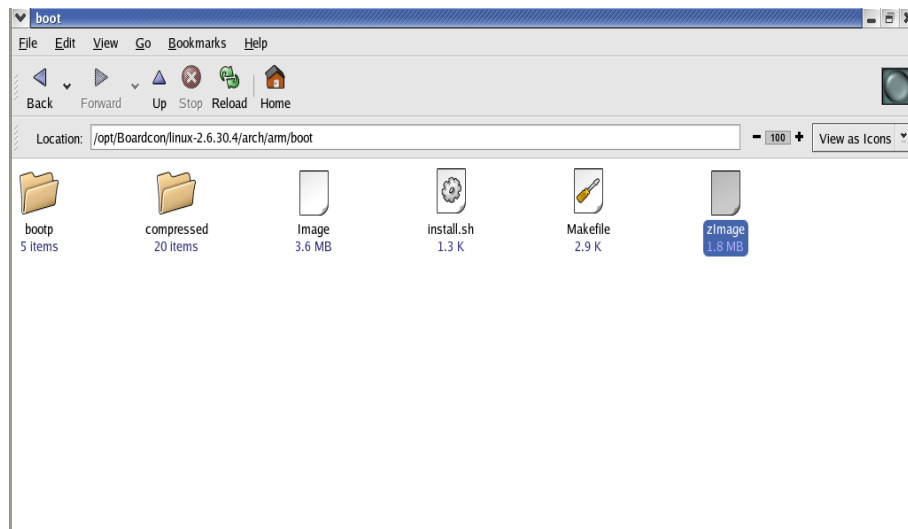
root@localhost/opt/Boardcon/linux-2.6.30.4
File Edit View Terminal Go Help

*** End of Linux kernel configuration.
*** Execute 'make' to build the kernel or try 'make help'.

[root@localhost linux-2.6.30.4]# make zImage
make: *** No rule to make target 'zImage'. Stop.
[root@localhost linux-2.6.30.4]# make zImage
CHK include/linux/version.h
make[1]: `include/asm-arm/mach-types.h' is up to date.
CHK include/linux/utsrelease.h
SYMLINK include/asm -> include/asm-arm
CALL scripts/checksyscalls.sh
<stdin>:1097:2: warning: #warning syscall fadvise64 not implemented
<stdin>:1265:2: warning: #warning syscall migrate_pages not implemented
<stdin>:1321:2: warning: #warning syscall pselect6 not implemented
<stdin>:1325:2: warning: #warning syscall ppoll not implemented
<stdin>:1365:2: warning: #warning syscall epoll_pwait not implemented
CHK include/linux/compile.h
Kernel: arch/arm/boot/Image is ready
Kernel: arch/arm/boot/zImage is ready
[root@localhost linux-2.6.30.4]# make zImage
CHK include/linux/version.h
make[1]: `include/asm-arm/mach-types.h' is up to date.
CHK include/linux/utsrelease.h
SYMLINK include/asm -> include/asm-arm
CALL scripts/checksyscalls.sh
<stdin>:1097:2: warning: #warning syscall fadvise64 not implemented
<stdin>:1265:2: warning: #warning syscall migrate_pages not implemented
<stdin>:1321:2: warning: #warning syscall pselect6 not implemented
<stdin>:1325:2: warning: #warning syscall ppoll not implemented
<stdin>:1365:2: warning: #warning syscall epoll_pwait not implemented
CHK include/linux/compile.h
Kernel: arch/arm/boot/Image is ready
Kernel: arch/arm/boot/zImage is ready
[root@localhost linux-2.6.30.4]#

```

The kernel image file “zImage” is automatically created under the directory “/opt/Boardcon/linux-2.6.30.4/arch/arm/boot”.



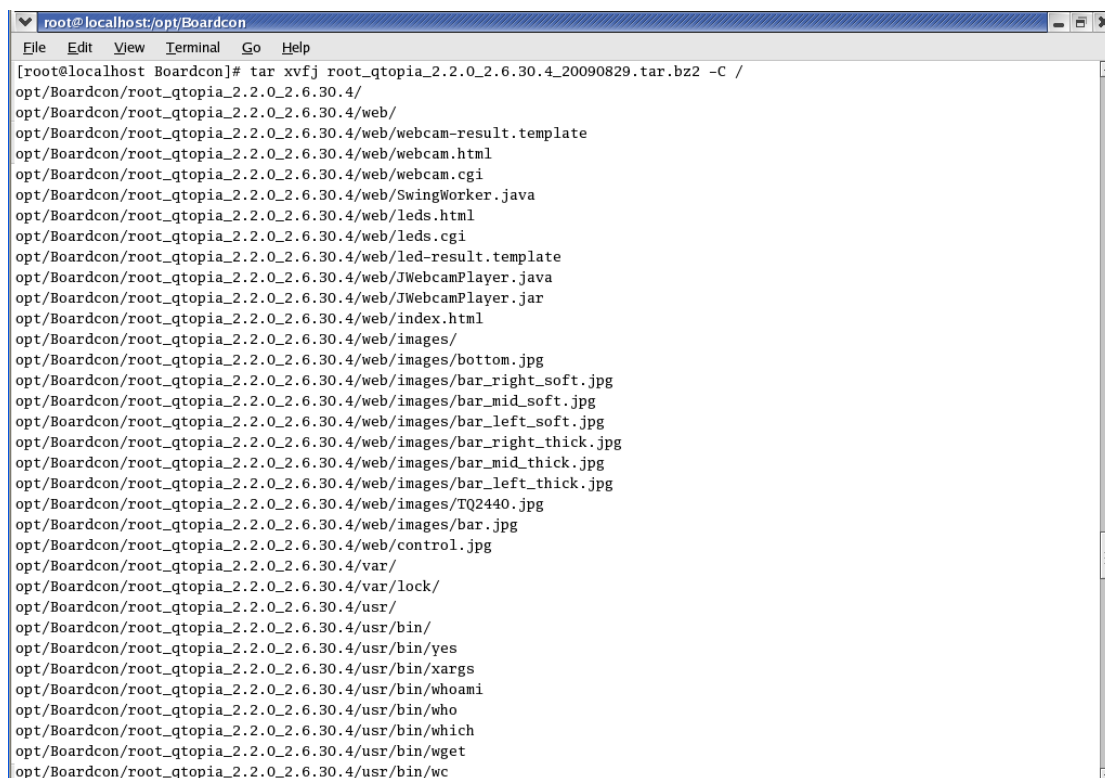
This file zImage is linux kernel image that used to be burn into the board.

## Chapter 4 Custom Your Root File System

The root file system is loaded by Linux when initializing. The user can execute the command “root=” to set the device that is corresponding to the root file system. The root file system has supported in our CD( linux directory), so you can modify it according to your applications.

Step 1: Decompress “root\_qtopia\_2.2.0\_2.6.30.4\_20090829.tar.bz2” under the directory “Linux” to “/opt/Boardcon”. Execute the command :

**# tar xvfj root\_qtopia\_2.2.0\_2.6.30.4\_20090829.tar.bz2 -C /**

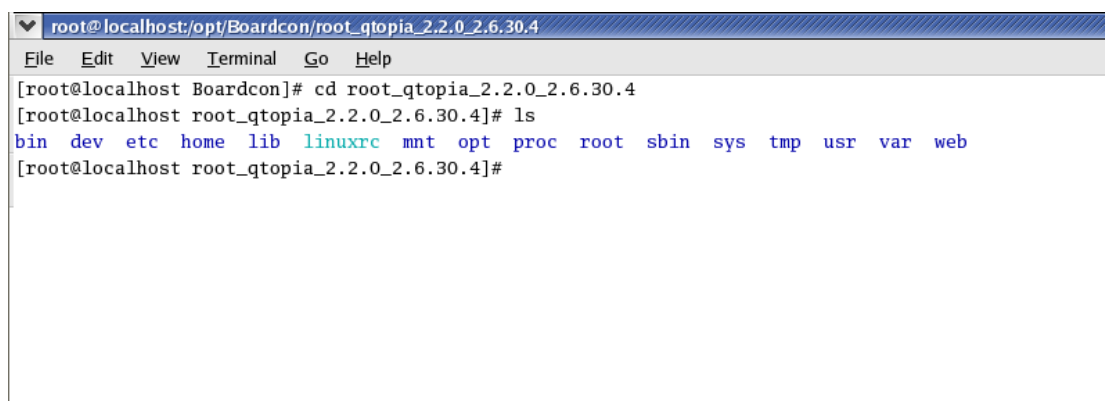


```

root@localhost:opt/Boardcon
File Edit View Terminal Go Help
[root@localhost Boardcon]# tar xvfj root_qtopia_2.2.0_2.6.30.4_20090829.tar.bz2 -C /
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/webcam-result.template
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/webcam.html
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/webcam.cgi
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/SwingWorker.java
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/leds.html
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/leds.cgi
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/led-result.template
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/JWebcamPlayer.java
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/JWebcamPlayer.jar
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/index.html
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/images/
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/images/bottom.jpg
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/images/bar_right_soft.jpg
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/images/bar_mid_soft.jpg
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/images/bar_left_soft.jpg
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/images/bar_right_thick.jpg
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/images/bar_mid_thick.jpg
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/images/bar_left_thick.jpg
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/images/TQ2440.jpg
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/images/bar.jpg
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/web/control.jpg
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/var/
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/var/lock/
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/usr/
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/usr/bin/
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/usr/bin/yes
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/usr/bin/xargs
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/usr/bin/whoami
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/usr/bin/who
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/usr/bin/which
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/usr/bin/wget
opt/Boardcon/root_qtopia_2.2.0_2.6.30.4/usr/bin/wc

```

Step 2: Enter into the directory root\_qtopia\_2.2.0\_2.6.30.4, and modify the file system according to your applications.



```

root@localhost:opt/Boardcon/root_qtopia_2.2.0_2.6.30.4
File Edit View Terminal Go Help
[root@localhost Boardcon]# cd root_qtopia_2.2.0_2.6.30.4
[root@localhost root_qtopia_2.2.0_2.6.30.4]# ls
bin dev etc home lib linuxrc mnt opt proc root sbin sys tmp usr var web
[root@localhost root_qtopia_2.2.0_2.6.30.4]#

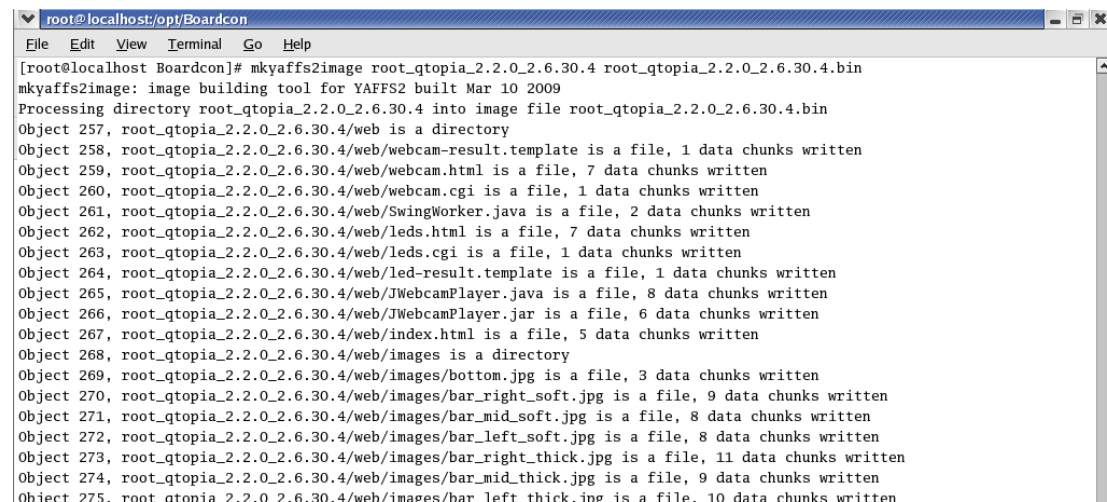
```

Step 3: Get out of the directory root\_qtopia\_2.2.0\_2.6.30.4, execute the command:



#mkyaffs2image root\_qtopia\_2.2.0\_2.6.30.4 root\_qtopia\_2.2.0\_2.6.30.4.bin

Then, we get the root file system image “root\_qtopia\_2.2.0\_2.6.30.4.bin” that used to be burn into the board.



```
root@localhost:opt/Boardcon
File Edit View Terminal Go Help
[root@localhost Boardcon]# mkyaffs2image root_qtopia_2.2.0_2.6.30.4 root_qtopia_2.2.0_2.6.30.4.bin
mkyaffs2image: image building tool for YAFFS2 built Mar 10 2009
Processing directory root_qtopia_2.2.0_2.6.30.4 into image file root_qtopia_2.2.0_2.6.30.4.bin
Object 257, root_qtopia_2.2.0_2.6.30.4/web is a directory
Object 258, root_qtopia_2.2.0_2.6.30.4/web/webcam-result.template is a file, 1 data chunks written
Object 259, root_qtopia_2.2.0_2.6.30.4/web/webcam.html is a file, 7 data chunks written
Object 260, root_qtopia_2.2.0_2.6.30.4/web/webcam.cgi is a file, 1 data chunks written
Object 261, root_qtopia_2.2.0_2.6.30.4/web/SwingWorker.java is a file, 2 data chunks written
Object 262, root_qtopia_2.2.0_2.6.30.4/web/leds.html is a file, 7 data chunks written
Object 263, root_qtopia_2.2.0_2.6.30.4/web/leds.cgi is a file, 1 data chunks written
Object 264, root_qtopia_2.2.0_2.6.30.4/web/led-result.template is a file, 1 data chunks written
Object 265, root_qtopia_2.2.0_2.6.30.4/web/JWebcamPlayer.java is a file, 8 data chunks written
Object 266, root_qtopia_2.2.0_2.6.30.4/web/JWebcamPlayer.jar is a file, 6 data chunks written
Object 267, root_qtopia_2.2.0_2.6.30.4/web/index.html is a file, 5 data chunks written
Object 268, root_qtopia_2.2.0_2.6.30.4/web/images is a directory
Object 269, root_qtopia_2.2.0_2.6.30.4/web/images/bottom.jpg is a file, 3 data chunks written
Object 270, root_qtopia_2.2.0_2.6.30.4/web/images/bar_right_soft.jpg is a file, 9 data chunks written
Object 271, root_qtopia_2.2.0_2.6.30.4/web/images/bar_mid_soft.jpg is a file, 8 data chunks written
Object 272, root_qtopia_2.2.0_2.6.30.4/web/images/bar_left_soft.jpg is a file, 8 data chunks written
Object 273, root_qtopia_2.2.0_2.6.30.4/web/images/bar_right_thick.jpg is a file, 11 data chunks written
Object 274, root_qtopia_2.2.0_2.6.30.4/web/images/bar_mid_thick.jpg is a file, 9 data chunks written
Object 275, root_qtopia_2.2.0_2.6.30.4/web/images/bar_left_thick.png is a file. 10 data chunks written
```

## Chapter 5 Linux Experiment

Before the following examples, you should first burn the linux image. You can refer to “How to Burn the Image” for details.

### 5.1 Experiment of program termination

There are 2 methods to terminate a program:

Method 1: In terminal console, press “Ctrl” and hold, and then press “C”.

Method 2: If the program is running background, you can first execute the command “ps” to search the process ID of the program, and then execute the command “kill” to terminal it.

### 5.2 Experiment of program auto-run configuration

The user can configure the booting script or other system settings to set auto-run. The booting script is under the directory “/etc/init.d/rcS”, the settings is shown as the following contents ( the following contents might have some tiny differences with the actual ones ):

```
#!/bin/sh

PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/bin:    #Set the default execution
path

runlevel=S                                #User level. Here is: Single
prevlevel=N
umask 022
export PATH runlevel prevlevel

#
#  Trap CTRL-C &c only in this shell so we can interrupt subprocesses.
#
trap ":" INT QUIT TSTP
hwclock -s                                #Synchronize Linux clock and RTC

mknod /dev/pts/0 c 136 0
ln -s /dev/v4l/video0 /dev/video0
ln -s /dev/fb/0 /dev/fb0                    #The symbol link of FrameBuffer
ln -s /dev/vc/0 /dev/tty1
ln -s /dev/sound/dsp /dev/dsp                #The symbol link of sound
device
ln -s /dev/sound/mixer /dev/mixer
ln -s /dev/input/tsraw0 /dev/h3600_tsraw
```

```

#Set the common temporary directory
mount -t proc none /proc
mount -t tmpfs none /tmp
mount -t tmpfs none /var

mkdir -p /var/lib
mkdir -p /var/run
mkdir -p /var/log

/etc/rc.d/init.d/netd start           #Start telnet/ftp service
/etc/rc.d/init.d/httpd start         #Start web server
/etc/rc.d/init.d/leds start          #Start LED

ifconfig lo 127.0.0.1                #IP address of local loop device
route add default gw 192.168.1.2    #Set gateway
ifconfig eth0 192.168.1.6 up         #Set and enable the IP
address of network card

/bin/qtopia &                        #Run Qt/Embedded after start-up

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

```

### 5.3 Experiment of setting and saving system real-time clock

Time in Linux is modified by executing the command “date”. The command “hwclock” is used to synchronize S3C2440 internal clock and Linux system clock:

- The format of “date” command: month date hour minute year. For example 2007-08-28 12:30, the command is “date -s 082812302007”.

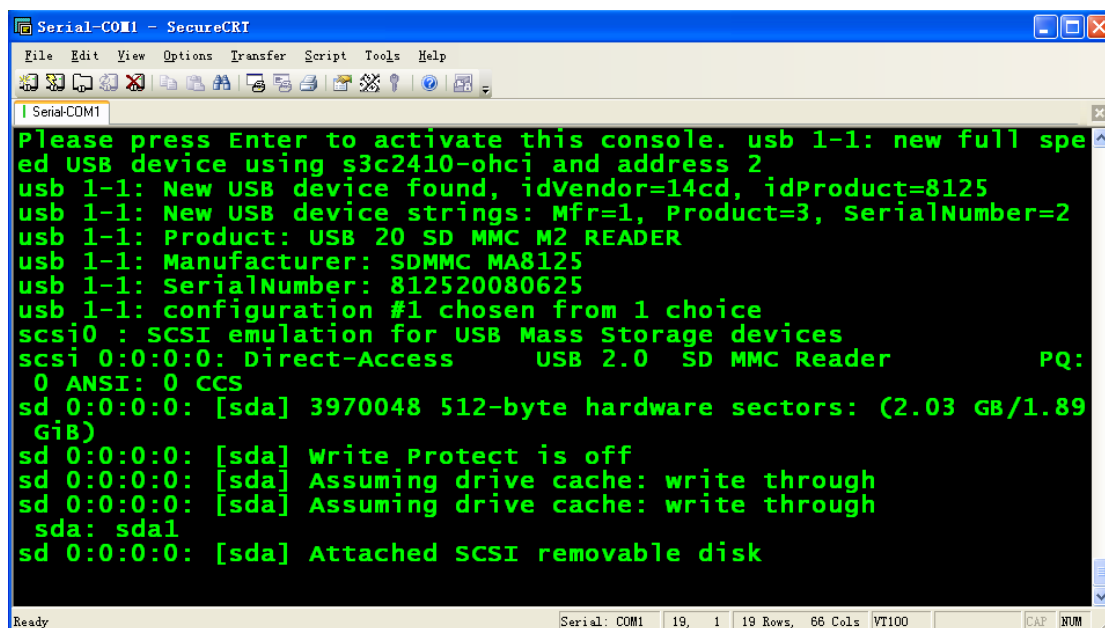
- Execute the command “hwclock -w” to save the time to S3C2440 internal RTC. Execute the command “hwclock -s” when Linux start-up to recover the RTC time. You can also add the command to the directory “etc/init.d/rcS” to set auto-run when start-up.

**Caution:** We have added the command “hwclock -s” to the file “rcS” in factory.

### 5.4 Experiment of USB mobile storage device test

After the USB mobile storage device is inserted, the hyper-terminal appears prompt information, as shown in the following diagram. According to the prompt, you could mount the USB mobile storage device under the directory “/mnt/udisk”:

The prompt of U-disk insertion is shown in the following diagram:



```

Serial-COM1 - SecureCRT
File Edit View Options Transfer Script Tools Help
Serial-COM1
Please press Enter to activate this console. usb 1-1: new full speed
usb 1-1: New USB device found, idVendor=14cd, idProduct=8125
usb 1-1: New USB device strings: Mfr=1, Product=3, SerialNumber=2
usb 1-1: Product: USB 20 SD MMC M2 READER
usb 1-1: Manufacturer: SDMMC MA8125
usb 1-1: SerialNumber: 812520080625
usb 1-1: configuration #1 chosen from 1 choice
scsi0 : SCSI emulation for USB Mass Storage devices
scsi 0:0:0:0: Direct-Access    USB 2.0  SD MMC Reader          PQ:
0 ANSI: 0 CCS
sd 0:0:0:0: [sda] 3970048 512-byte hardware sectors: (2.03 GB/1.89
GiB)
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] Assuming drive cache: write through
sda: sda1
sd 0:0:0:0: [sda] Attached SCSI removable disk
Ready
Serial: COM1 19, 1 19 Rows, 66 Cols VT100 CAP RTM

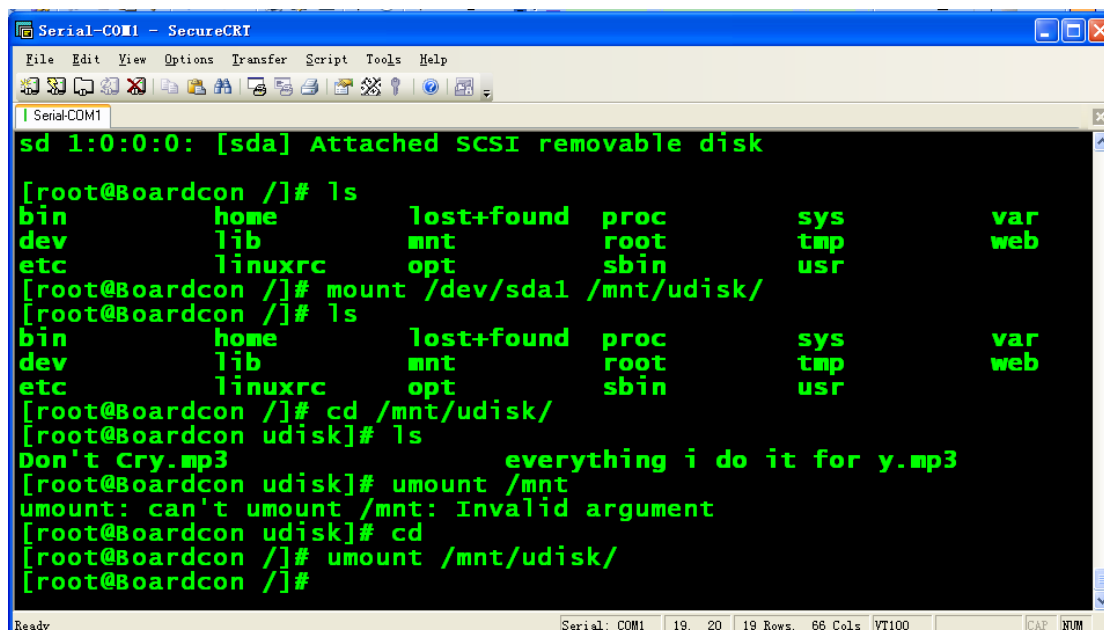
```

Execute the mount command:

```
#mount /dev/sda1 /mnt/udisk
```

And execute the umount command:

```
#umount /mnt/udisk
```



```

Serial-COM1 - SecureCRT
File Edit View Options Transfer Script Tools Help
Serial-COM1
sd 1:0:0:0: [sda] Attached SCSI removable disk
[root@Boardcon /]# ls
bin      home      lost+found  proc      sys      var
dev      lib       mnt        root      tmp      web
etc      linuxrc  opt        sbin     usr
[root@Boardcon /]# mount /dev/sda1 /mnt/udisk/
[root@Boardcon /]# ls
bin      home      lost+found  proc      sys      var
dev      lib       mnt        root      tmp      web
etc      linuxrc  opt        sbin     usr
[root@Boardcon /]# cd /mnt/udisk/
[root@Boardcon udisk]# ls
Don't Cry.mp3      everything i do it for y.mp3
[root@Boardcon udisk]# umount /mnt
umount: can't umount /mnt: Invalid argument
[root@Boardcon udisk]# cd
[root@Boardcon /]# umount /mnt/udisk/
[root@Boardcon /]#
Ready
Serial: COM1 19, 20 19 Rows, 66 Cols VT100 CAP RTM

```

## 5.5 Experiment of SD card test

Just like using U-disk, the following information appears after the SD card insertion:

```

etc          linuxrc      opt          sbin         usr
[root@Boardcon /]# mount /dev/sda1 /mnt/udisk/
[root@Boardcon /]# ls
bin          home          lost+found   proc         sys          var
dev          lib           mnt         root         tmp         web
etc          linuxrc      opt          sbin         usr
[root@Boardcon /]# cd /mnt/udisk/
[root@Boardcon udisk]# ls
Don't Cry.mp3          everything i do it for y.mp3
[root@Boardcon udisk]# umount /mnt
umount: can't umount /mnt: Invalid argument
[root@Boardcon udisk]# cd
[root@Boardcon /]# umount /mnt/udisk/
[root@Boardcon /]# usb 1-1: USB disconnect, address 3
mmc0: new MMC card at address 0001
mmcblk0: mmc0:0001 000000 980 MiB
mmcblk0: p1
[root@Boardcon /]#

```

Execute the command “**#mount /dev/mmcblk0p1 /mnt/sd**” to mount SD card under the directory of “/mnt/sd”, and execute the command “**#umount /mnt/sd**”.

```

[root@Boardcon /]# ls
bin          home          lost+found   proc         sys          var
dev          lib           mnt         root         tmp         web
etc          linuxrc      opt          sbin         usr
[root@Boardcon /]# cd /mnt/udisk/
[root@Boardcon udisk]# ls
Don't Cry.mp3          everything i do it for y.mp3
[root@Boardcon udisk]# umount /mnt
umount: can't umount /mnt: Invalid argument
[root@Boardcon udisk]# cd
[root@Boardcon /]# umount /mnt/udisk/
[root@Boardcon /]# usb 1-1: USB disconnect, address 3
mmc0: new MMC card at address 0001
mmcblk0: mmc0:0001 000000 980 MiB
mmcblk0: p1

[root@Boardcon /]# mount /dev/mmcblk0p1 /mnt/sd/
[root@Boardcon /]# umount /mnt/sd/
[root@Boardcon /]#

```

## 5.6 Experiment of USB camera capturing screen test

EM2440-III Development Board supports most USB cameras sold in market, for example the USB camera produced by Smics. After the camera is inserted to USB interface, the following information appears, and you can find the device name under the directory “/dev/v4l/”:

[illegible]

Use the software `sapccat` you can get images captured by camera. Execute the command: `sapccat -p 100ms -N 3`

The screenshot displays a Windows terminal application with the title bar "超级终端 - 超级终端". The menu bar includes options like "文件(F)", "编辑(E)", "查看(V)", "呼叫(C)", "传送(T)", and "帮助(H)". Below the menu bar are icons for file operations. The main area shows a shell session:

```
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]# usb 1-1: new full speed USB device using s3c2410-ohci and address 8  
drivers/usb/media/gspca/gspca_core.c: USB SPCA5XX camera found.(ZC3XX)  
  
[root@EmbedSky /]# ls /dev/v4l/  
video0  
[root@EmbedSky /]# spcacadat -s 320x240 -p 100ms -N 1 -o
```

The status bar at the bottom indicates connection details: "已连接 3:14:23 ANSIW 115200 8-N-1 SCROLL CAPS NUM 捕 打印".

Command illustration: “-s” represents solution; “-p” represents the time interval between two captures; “-N” represents how many images to capture; “-o” represents over writing the former images named “SpcaPict.jpg” and saving the new one.

```

Available Resolutions width 192  height 144
Available Resolutions width 176  height 144
Format asked 15 check 8
VIDIOCSPIC brightness=32768 hue=0 color=0 contrast=32768 whiteness=0depth=12 palette=15
VIDIOCGPCT brightness=32768 hue=0 color=0 contrast=32768 whiteness=0depth=12 palette=15
grabbing method default MMIO asked
VIDIOCGMBUF size 2457616 frames 2 offsets[0]=0 offsets[1]=1228808
Waiting ... for Incoming Events. Ctrl_c to stop !!!!
picture jpeg SpcaPict.jpg
GRABBER going out !!!!
unmapping frame buffer
close video_device
freeing output buffer 0
freeing output buffer 1
freeing output buffer 2
freeing output buffer 3
[root@EmbedSky /]# ls
SpcaPict.jpg  home  opt  tmp
bin          lib   proc  usr
dev          linuxrc root  var
etc          mnt /sbin  web
[root@EmbedSky /]# _

```

Execute the command “**spccat -h**” to get more information of the parameters:

```

GRABBER going out !!!!
unmapping frame buffer
close video_device
freeing output buffer 0
freeing output buffer 1
freeing output buffer 2
freeing output buffer 3
[root@EmbedSky /]# ls
SpcaPict.jpg  home  opt  tmp
bin          lib   proc  usr
dev          linuxrc root  var
etc          mnt /sbin  web
[root@EmbedSky /]# spccat -h
usage: cdse [-h -d -g -s -P -p -N -o]
-h      print this message
-d      /dev/videoX      use videoX device
-g      use read method for grab instead mmap
-f      video format default yuv others options are r16 r24 r32 yuv jpg
-s      widthxheight    use specified input size
-P      /dev/partportX    use partportX device
-p      x ms           take a picture every x ms minimum is set to 50ms
-N      take a N pictures every p ms and stop
-o      overwrite picture, each picture come with the same name SpacPict.jpg
[root@EmbedSky /]#

```

## 5.7 Experiment of sound card test

Madplay is a MP3 player running on console, with various control modes. Execute the command “**madplay -h**” to get more help information:

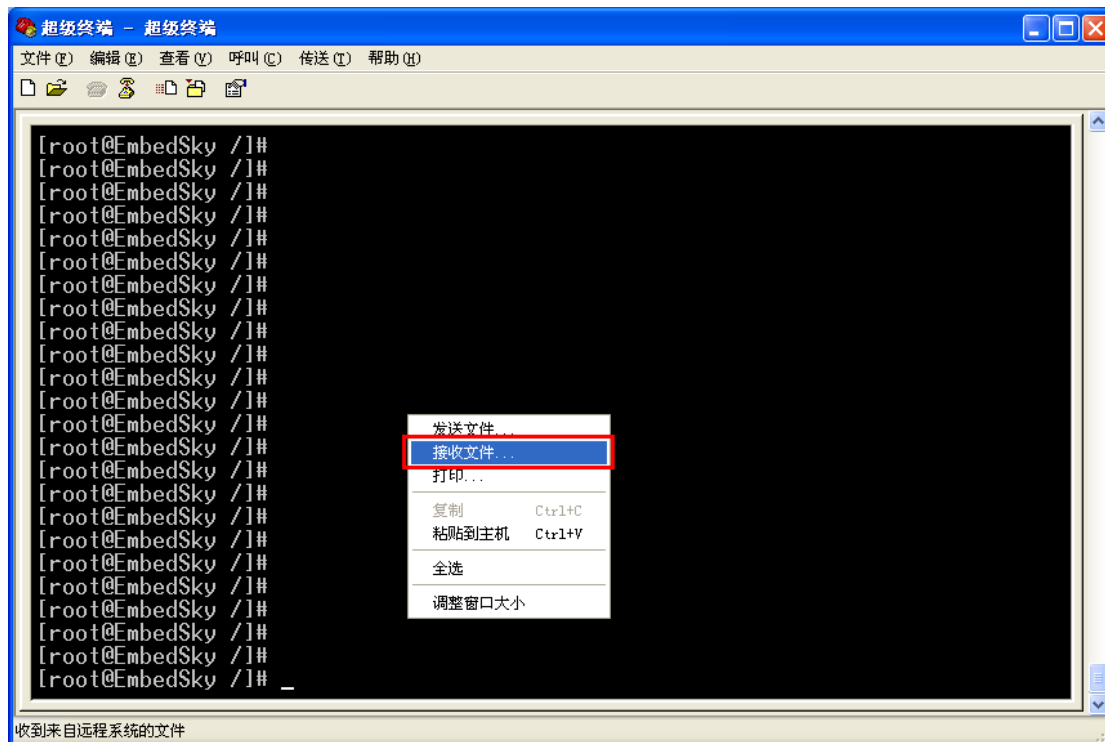




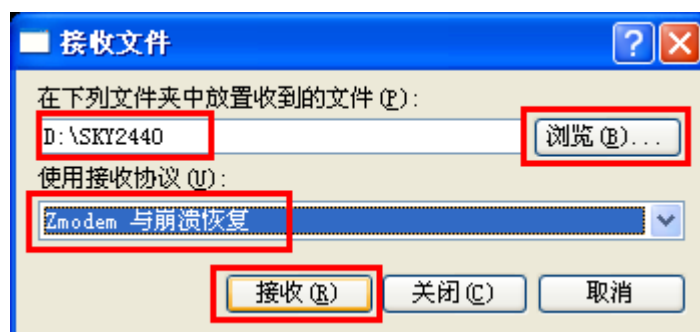
The operation is introduced as follows steps:

- 1) send files to PC:

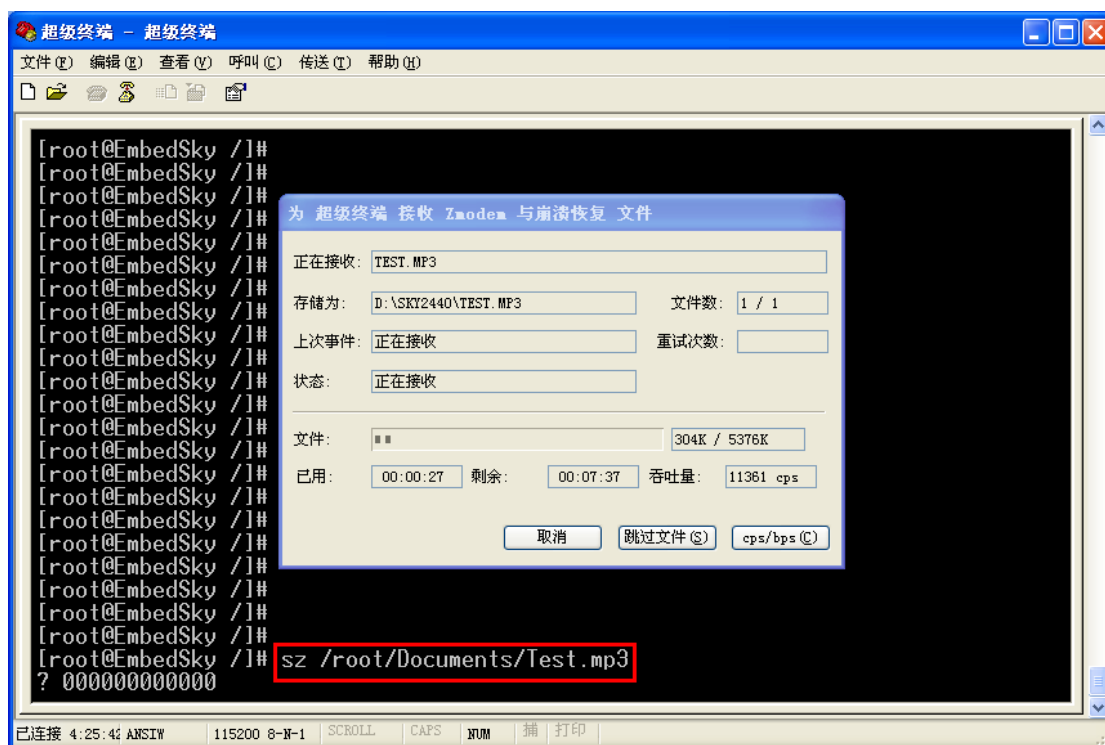
Step1, click mouse right button in hyper-terminal interface, and select “receive files”:



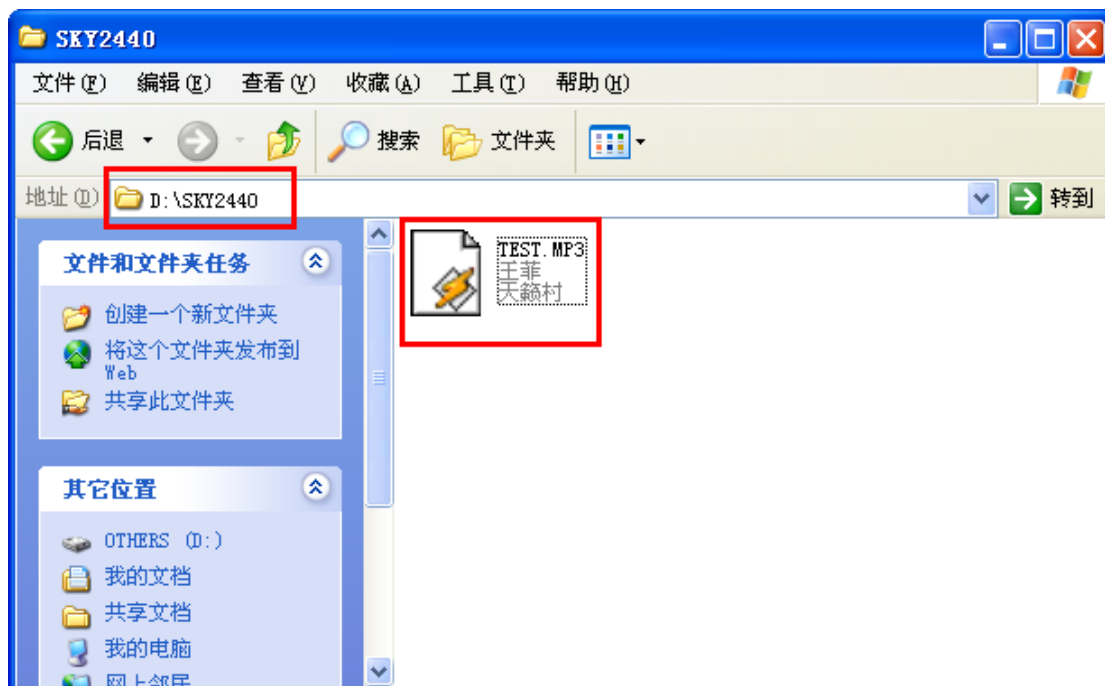
Step2, the interface “Receive files” pops up. Configure the interface as the following diagram and click “Receive” to continue:



Step3, enter the command “sz /root/Documents/Test.mp3” in hyper-terminal to start transmitting “Test.mp3” under the directory “/root/Documents/” to PC:

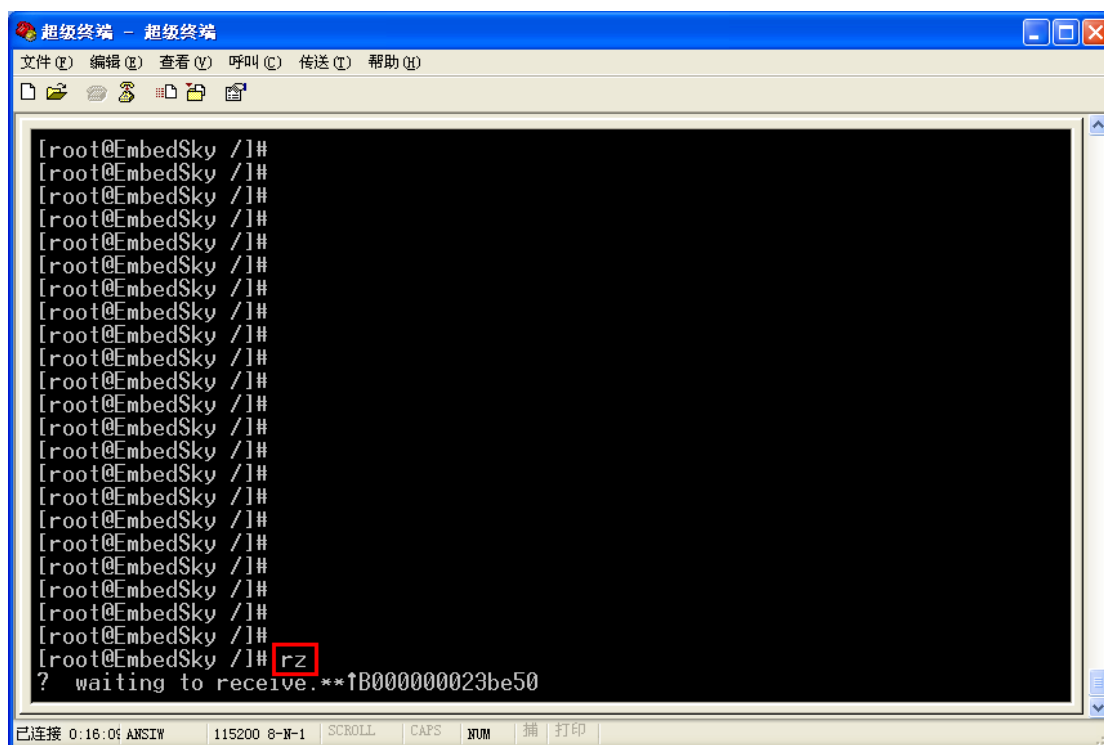


Step4, the file is automatically saved to the directory that you have just set after the transmission is over:

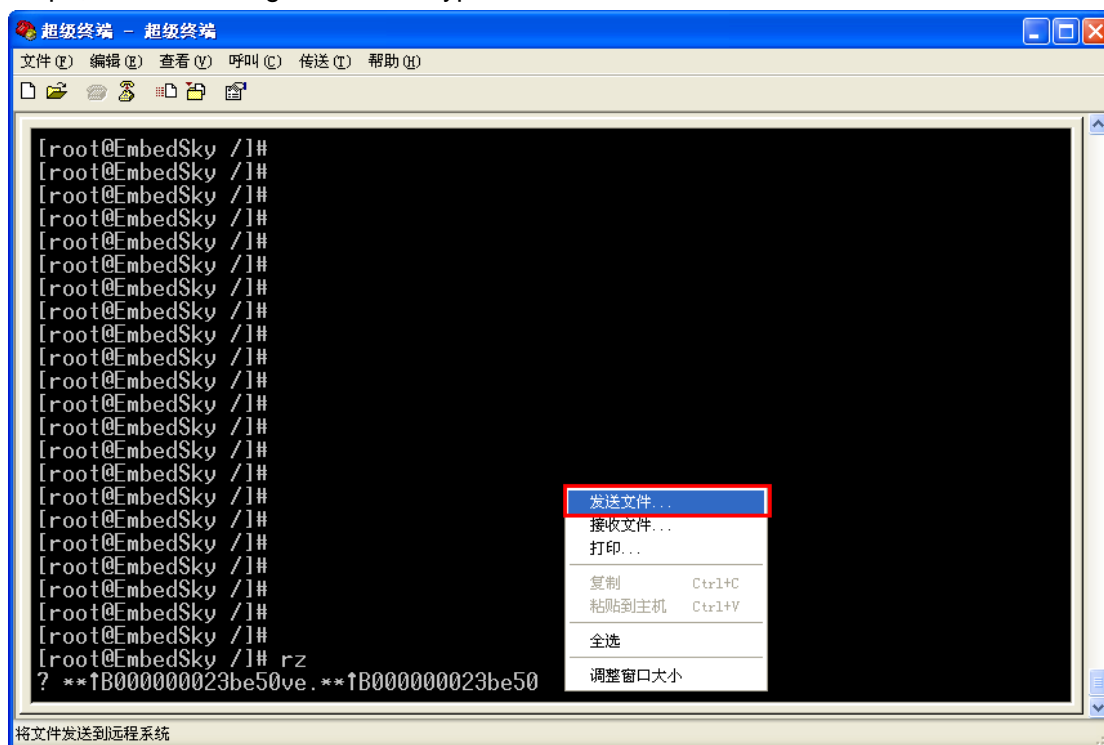


## 2). Transmit files to platform:

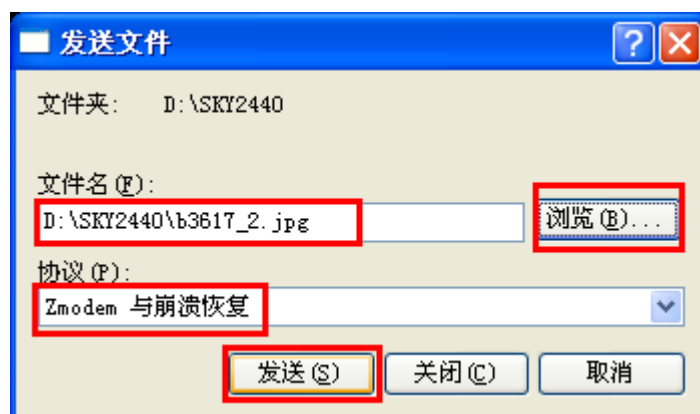
Step1, enter the command “rz” in hyper-terminal to start receiving files from PC:



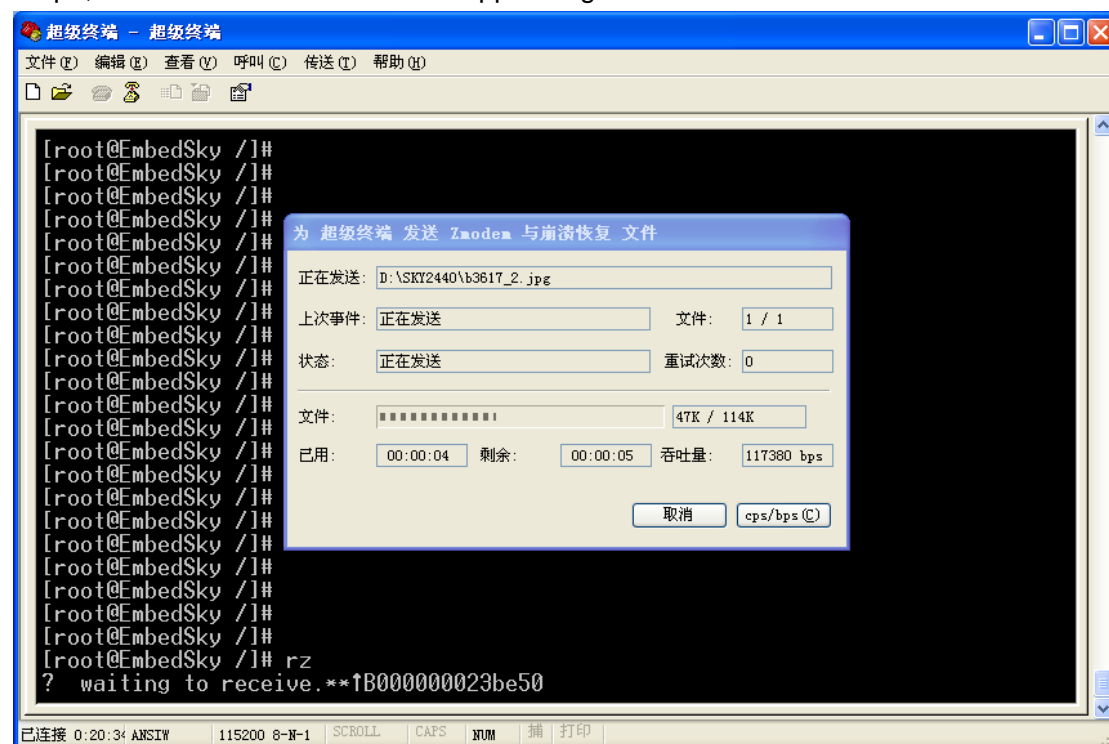
Step2, click mouse right button in hyper terminal and select "Send files":



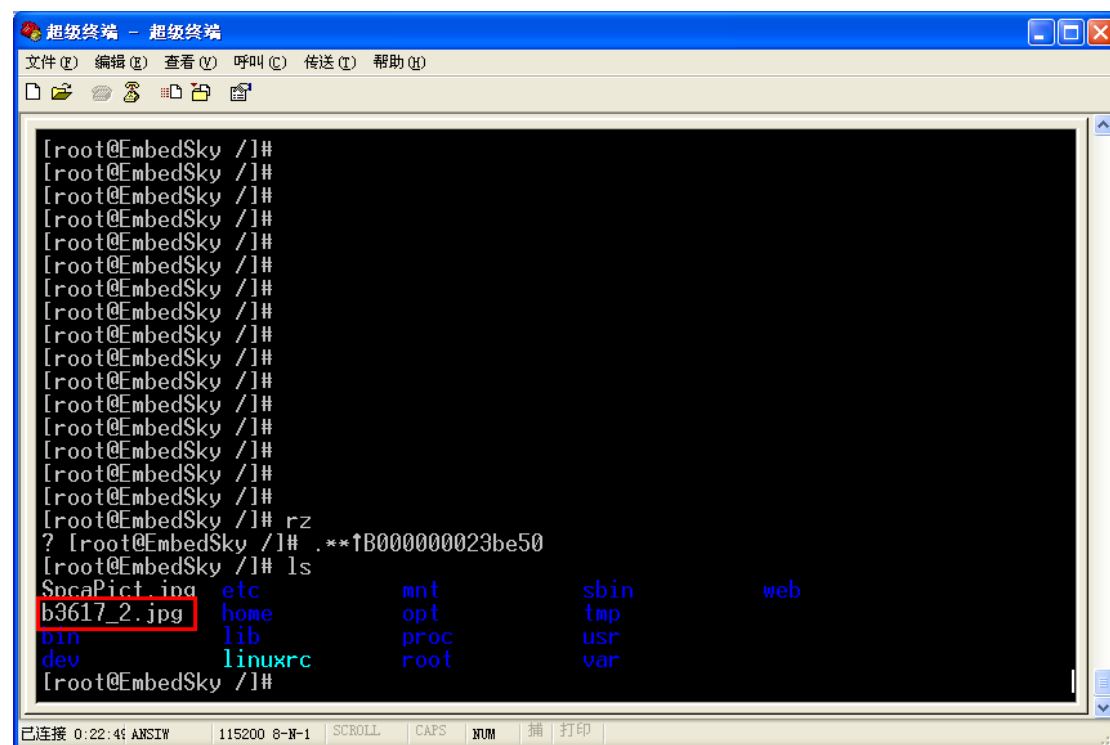
Step3, click the button “Browse files” in the pop-up interface “Send files” and locate the file for transmission, as shown in the following diagram:



Step4, click the button “Send” in the upper diagram to start transmission:



Step5, the name of the file transmitted is in the following red frame:



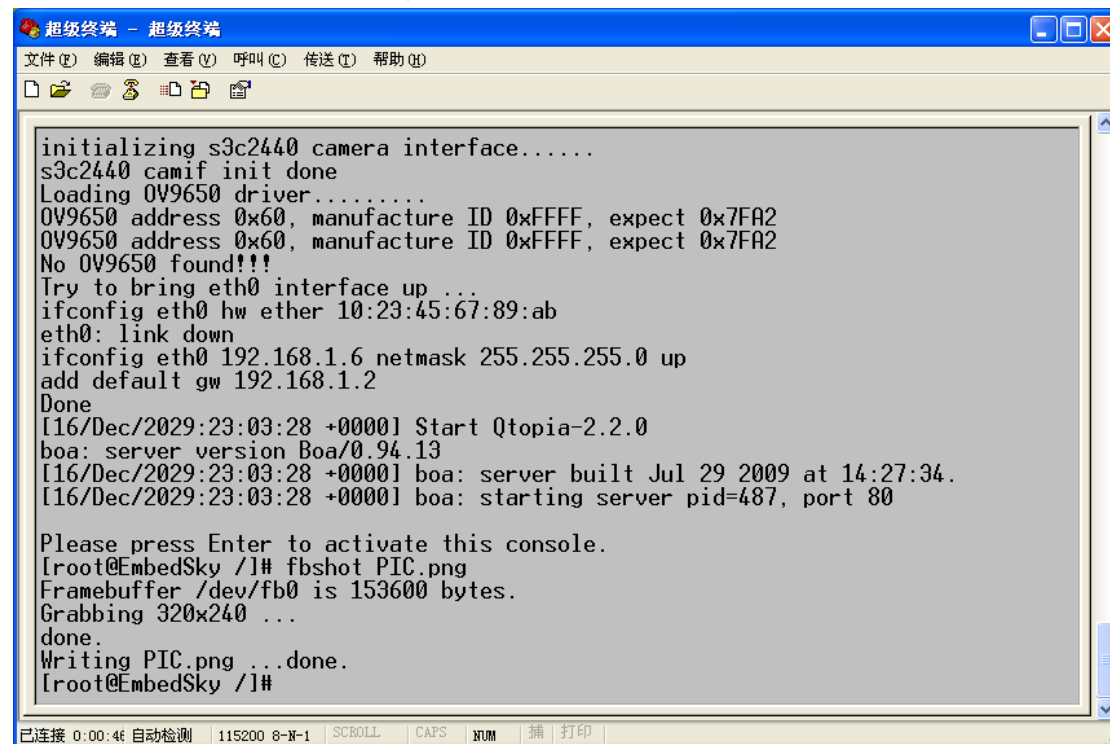
```

[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]# rz
? [root@EmbedSky /]# .**†B000000023be50
[root@EmbedSky /]# ls
SpcaPict.jpg  etc          mnt          sbin          web
b3617_2.jpg  home         opt          tmp
bin          lib          proc         usr
dev          linuxrc     root         var
[root@EmbedSky /]#

```

## 5.9 Experiment of screen capture

Execute the command “fbshot” to capture screen and save the image into the png format.  
The command: **# fbshot PIC.png**



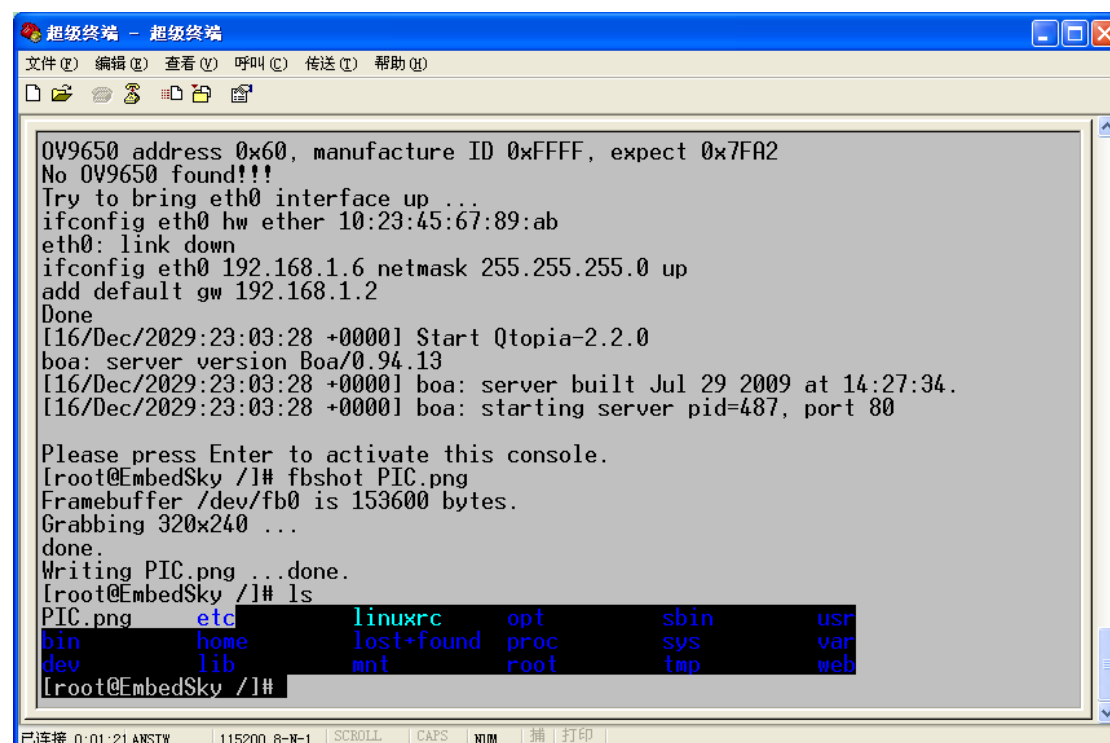
```

initializing s3c2440 camera interface.....
s3c2440 camif init done
Loading OV9650 driver.....
OV9650 address 0x60, manufacture ID 0xFFFF, expect 0x7FA2
OV9650 address 0x60, manufacture ID 0xFFFF, expect 0x7FA2
No OV9650 found!!!
Try to bring eth0 interface up ...
ifconfig eth0 hw ether 10:23:45:67:89:ab
eth0: link down
ifconfig eth0 192.168.1.6 netmask 255.255.255.0 up
add default gw 192.168.1.2
Done
[16/Dec/2029:23:03:28 +0000] Start Qtopia-2.2.0
boa: server version Boa/0.94.13
[16/Dec/2029:23:03:28 +0000] boa: server built Jul 29 2009 at 14:27:34.
[16/Dec/2029:23:03:28 +0000] boa: starting server pid=487, port 80

Please press Enter to activate this console.
[root@EmbedSky /]# fbshot PIC.png
Framebuffer /dev/fb0 is 153600 bytes.
Grabbing 320x240 ...
done.
Writing PIC.png ...done.
[root@EmbedSky /]#

```

The captured image is under the root directory:



```

0V9650 address 0x60, manufacture ID 0xFFFF, expect 0x7FA2
No 0V9650 found!!!
Try to bring eth0 interface up ...
ifconfig eth0 hw ether 10:23:45:67:89:ab
eth0: link down
ifconfig eth0 192.168.1.6 netmask 255.255.255.0 up
add default gw 192.168.1.2
Done
[16/Dec/2029:23:03:28 +0000] Start Qtopia-2.2.0
boa: server version Boa/0.94.13
[16/Dec/2029:23:03:28 +0000] boa: server built Jul 29 2009 at 14:27:34.
[16/Dec/2029:23:03:28 +0000] boa: starting server pid=487, port 80

Please press Enter to activate this console.
[root@EmbedSky /]# fbshot PIC.png
Framebuffer /dev/fb0 is 153600 bytes.
Grabbing 320x240 ...
done.
Writing PIC.png ...done.
[root@EmbedSky /]# ls
PIC.png  etc          linuxrc  opt       sbin      usr
bin      home         lost+found  proc     sys       var
dev      lib          mnt       root     tmp       web
[root@EmbedSky /]#

```

## 5.10 Experiment of user LED test

### 1) LED server

After system start-up, the user LED service application “/etc/rc.d/init.d/leds” executes automatically. The application calls a script of led-player to create a pipe file under the directory “/tmp”. The user LED flash mode varies according to the parameters send to this pipe:

```
> #echo 0 0.5 > /tmp/led-control
```

The 4 user LEDs run marquee at 0.5-second interval after the command running:



超级终端 - 超级终端

文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

```

[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
[root@EmbedSky /]#
echo 0 0.5 > /tmp/led-control
type is 0, period is 0.500000

[root@EmbedSky /]# _

```

已连接 0:40:26 ANSI 115200 8-N-1 SCROLL CAPS NUM 插 打印

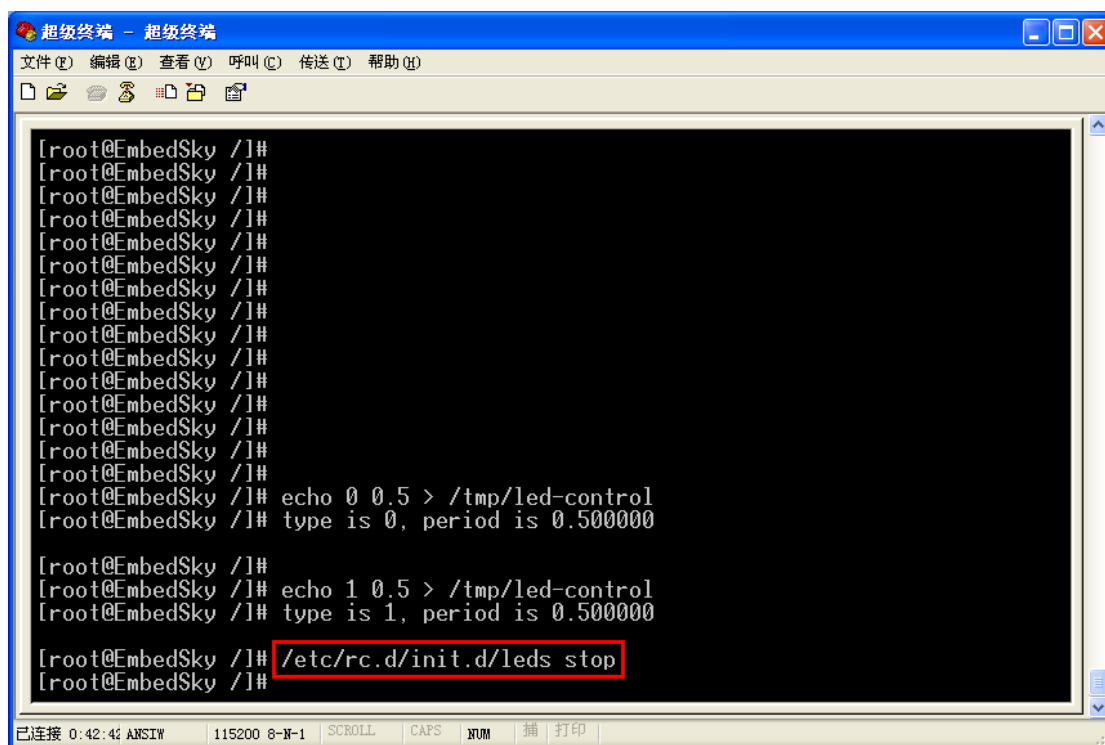
```
➤ #echo 1 0.5 > /tmp/led-control
```

The 4 user LEDs run accumulator at 0.5-second interval after the command running:

[illegible]

```
➤ #/etc/rc.d/init.d/leds stop
```

The 4 user LEDs stop flashing after the command running:



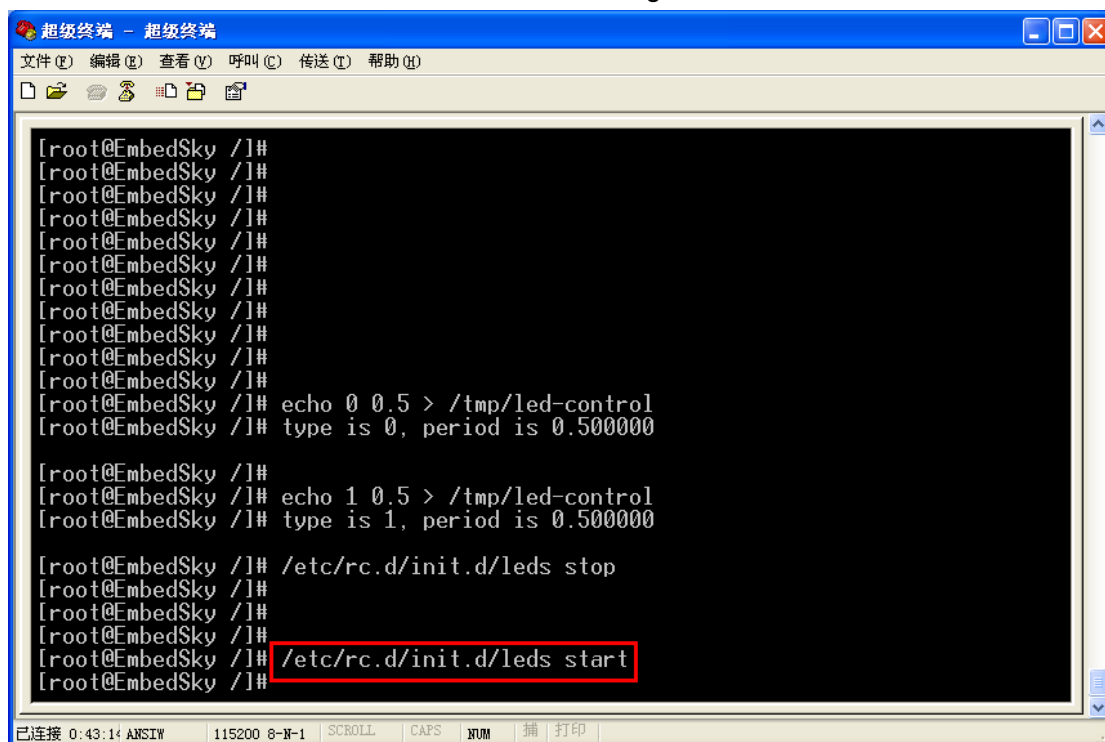
The screenshot shows a terminal window titled "超级终端 - 超级终端" (Super Terminal - Super Terminal). The terminal displays the following commands and output:

```
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]# echo 0 0.5 > /tmp/led-control  
[root@EmbedSky /]# type is 0, period is 0.500000  
  
[root@EmbedSky /]#  
[root@EmbedSky /]# echo 1 0.5 > /tmp/led-control  
[root@EmbedSky /]# type is 1, period is 0.500000  
  
[root@EmbedSky /]# /etc/rc.d/init.d/leds stop  
[root@EmbedSky /]#
```

The command `/etc/rc.d/init.d/leds stop` is highlighted with a red box. The status bar at the bottom shows "已连接 0:42:41 ANSIIW 115200 8-N-1 SCROLL CAPS NUM 捕 打印".

➤ `/etc/rc.d/init.d/leds start`

The 4 user LEDs re-flash after the command running:



The screenshot shows a terminal window titled "超级终端 - 超级终端" (Super Terminal - Super Terminal). The terminal displays the following commands and output:

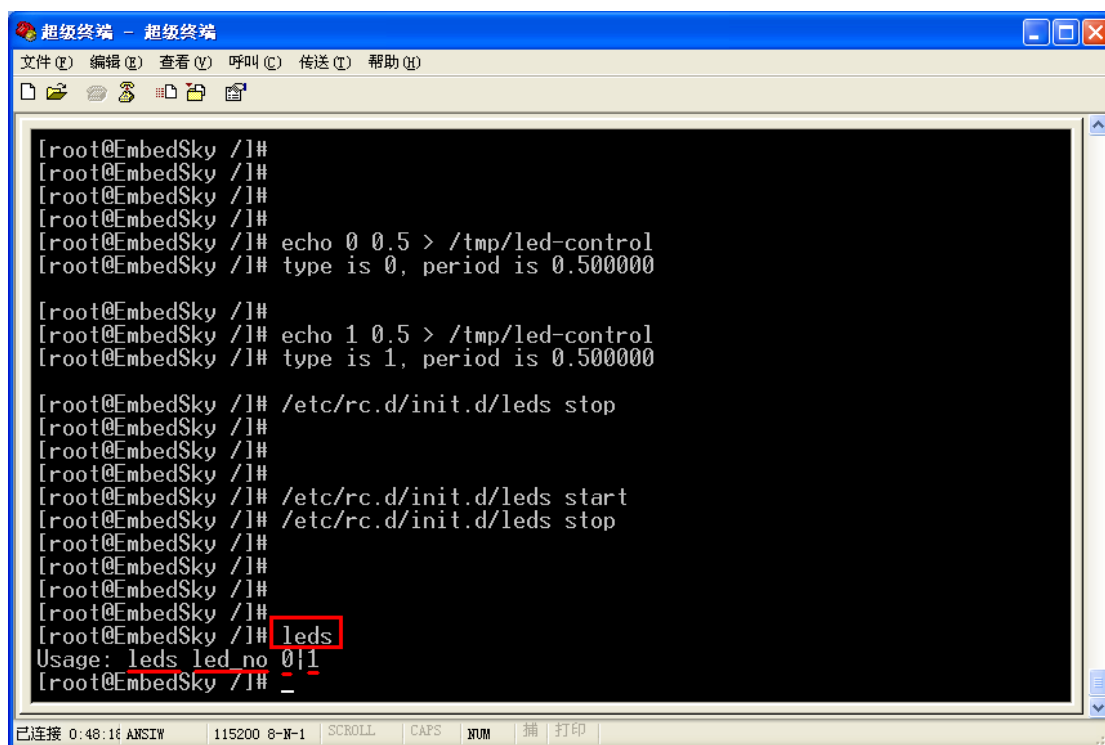
```
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]# echo 0 0.5 > /tmp/led-control  
[root@EmbedSky /]# type is 0, period is 0.500000  
  
[root@EmbedSky /]#  
[root@EmbedSky /]# echo 1 0.5 > /tmp/led-control  
[root@EmbedSky /]# type is 1, period is 0.500000  
  
[root@EmbedSky /]# /etc/rc.d/init.d/leds stop  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]# /etc/rc.d/init.d/leds start  
[root@EmbedSky /]#
```

The command `/etc/rc.d/init.d/leds start` is highlighted with a red box. The status bar at the bottom shows "已连接 0:43:14 ANSIIW 115200 8-N-1 SCROLL CAPS NUM 捕 打印".

## 2) control LED separately

The application `/sbin/leds` can control LED separately. You must stop led-player before run this application. Execute the command `/etc/rc.d/init.d/leds stop` to stop led-player.

"leds" usage method:



```
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]# echo 0 0.5 > /tmp/led-control  
[root@EmbedSky /]# type is 0, period is 0.500000  
  
[root@EmbedSky /]#  
[root@EmbedSky /]# echo 1 0.5 > /tmp/led-control  
[root@EmbedSky /]# type is 1, period is 0.500000  
  
[root@EmbedSky /]# /etc/rc.d/init.d/leds stop  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]# /etc/rc.d/init.d/leds start  
[root@EmbedSky /]# /etc/rc.d/init.d/leds stop  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]#  
[root@EmbedSky /]# leds  
Usage: leds led_no 0|1  
[root@EmbedSky /]#
```

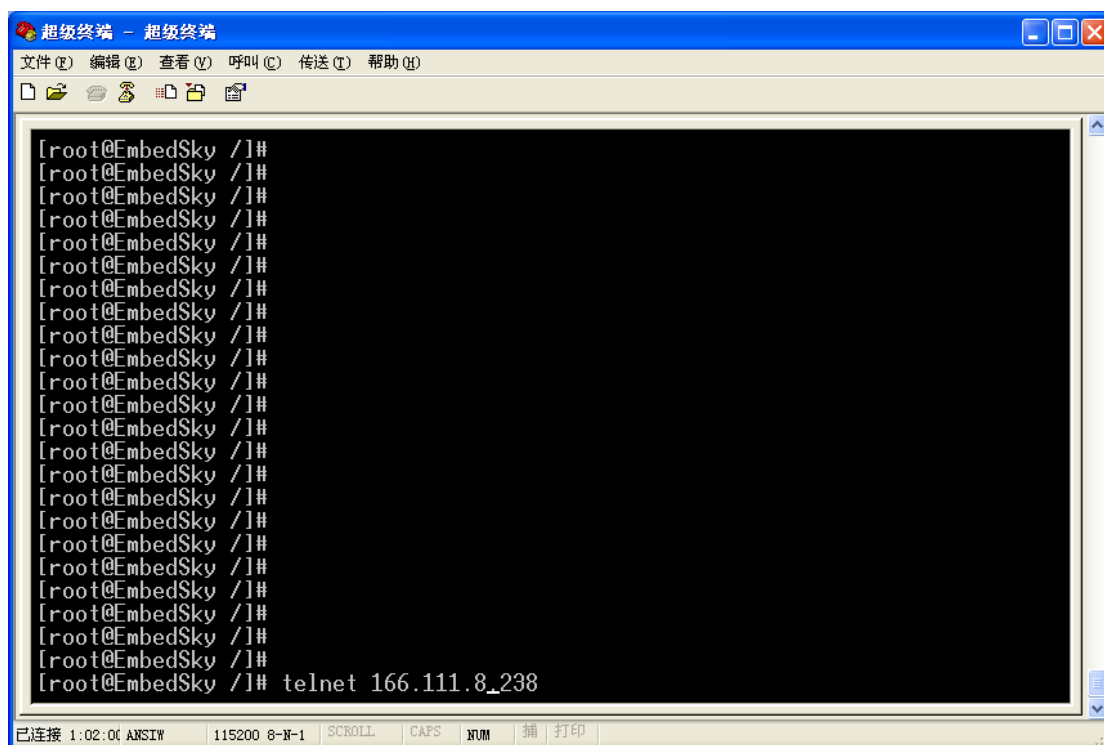
“led\_no” is the LED sequence number ( 0, 1, 2, 3 ); The value “0” and “1” represent operation switching off and switching on.

For example “leds 3 1” means switch on LED3.

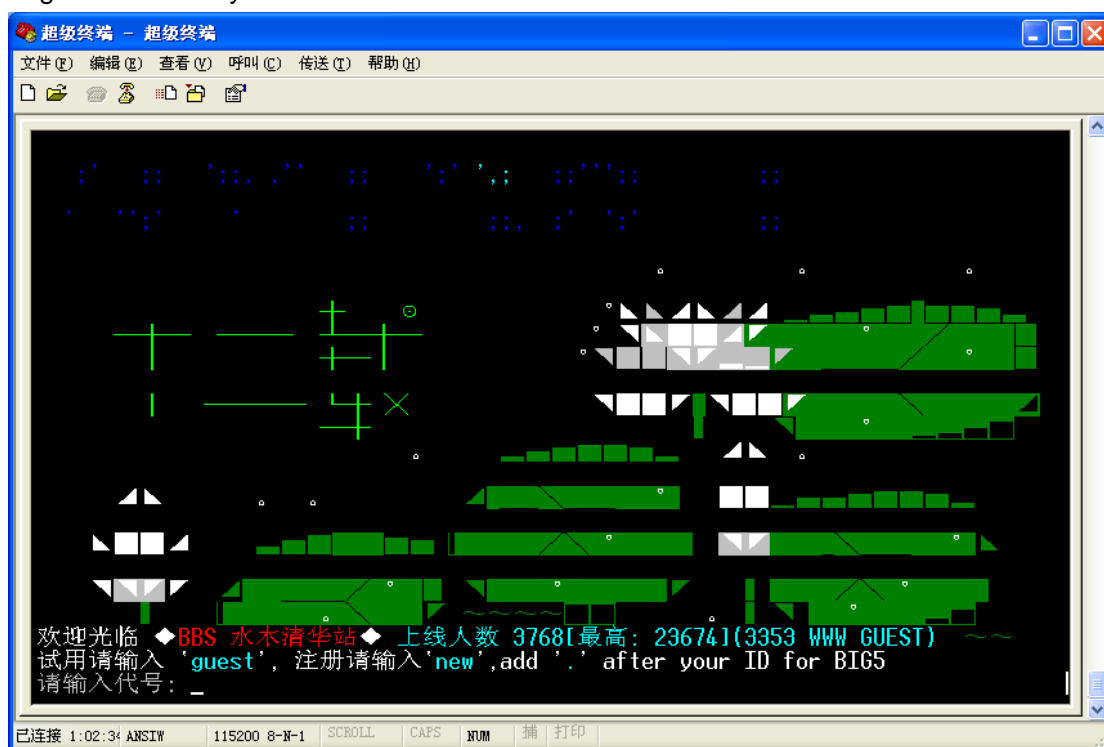
### 5.11 Experiment of log in BBS in telnet

Telnet is a kind of commonly used remote login tool. We can use telnet to log in other telnet servers from the platform. If the platform is connected to internet, you could use telnet command to log in external BBS.

Use telnet to log in BBS “ARM Forum”:

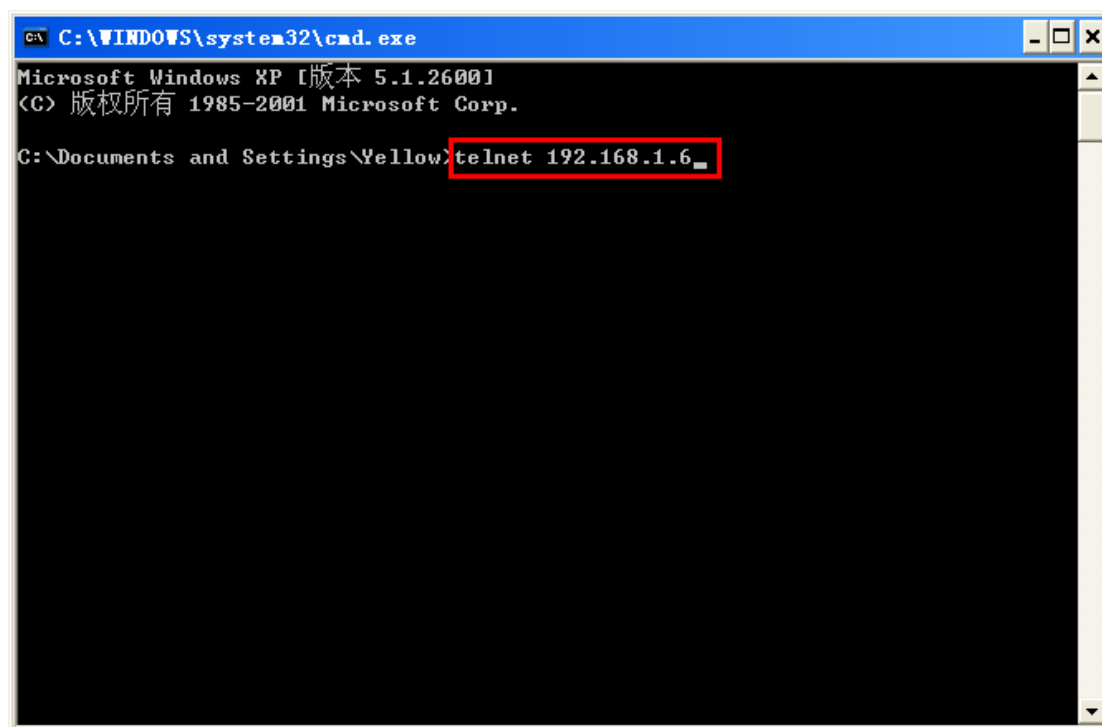


Log in successfully:

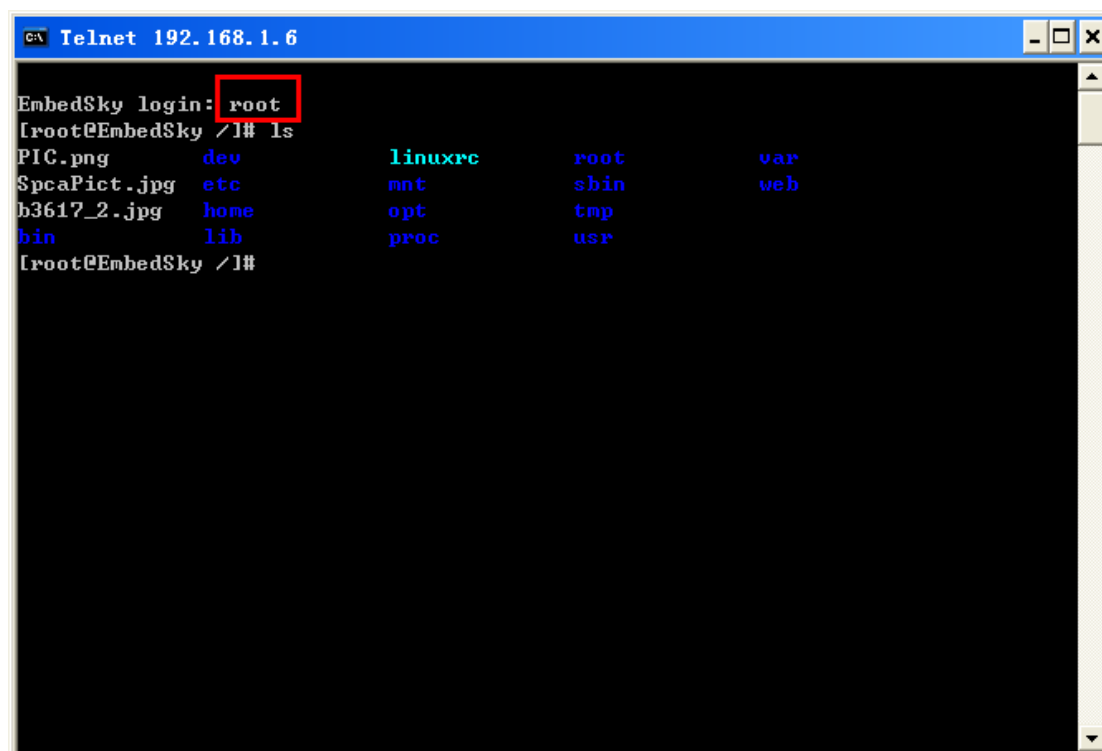


### 5.12 Experiment of remote login platform by telnet

The user can directly log in platform after system starts up. Enter the command “telnet 192.168.1.6” in command window in Windows OS, and press return-key to continue:



Enter "root" in the following log-in interface to enter into the system:



### 5.13 Experiment of FTP remote file transmission

We can use the ftp application contained in Linux or Windows to log in remote host and transmit file, if the remote host support ftp service and the authority is available.

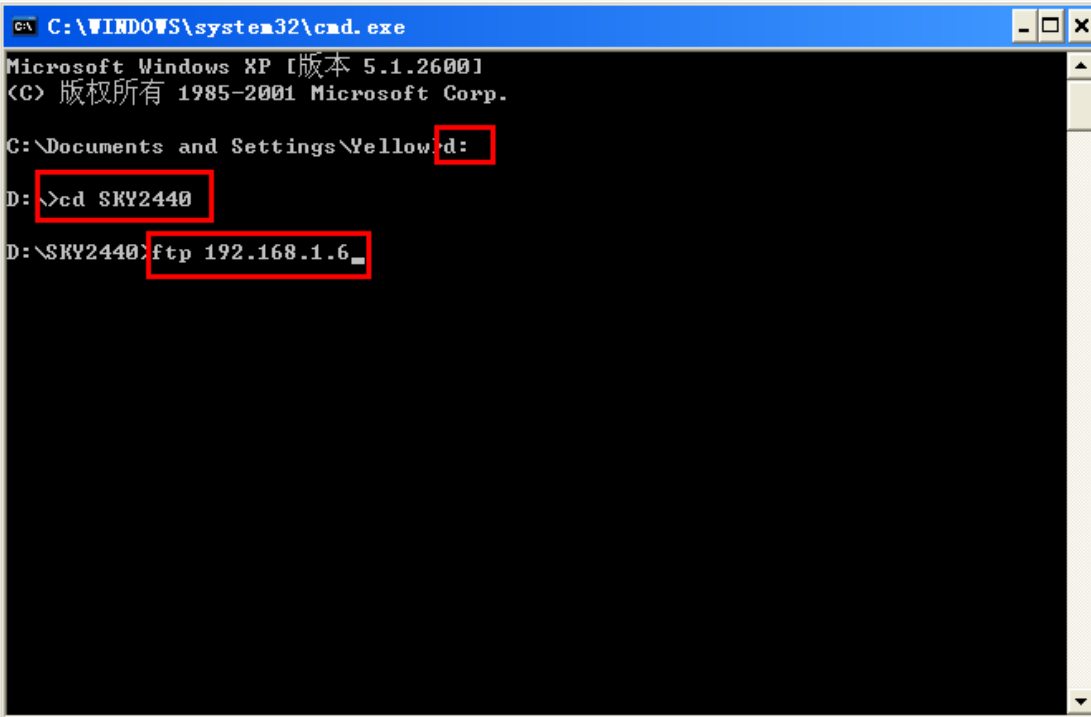
EM2440-III provides ftp application and ftp service. Here we make a test to log in platform from PC command window and send files to the platform.

**Caution:** Make sure the transmission file is under the same directory with ftp and it is available.

Here we use the file “PPMM.jpg” for transmission.

After the transmission is over, the file “PPMM.jpg” is added to the directory “/home/sky”.

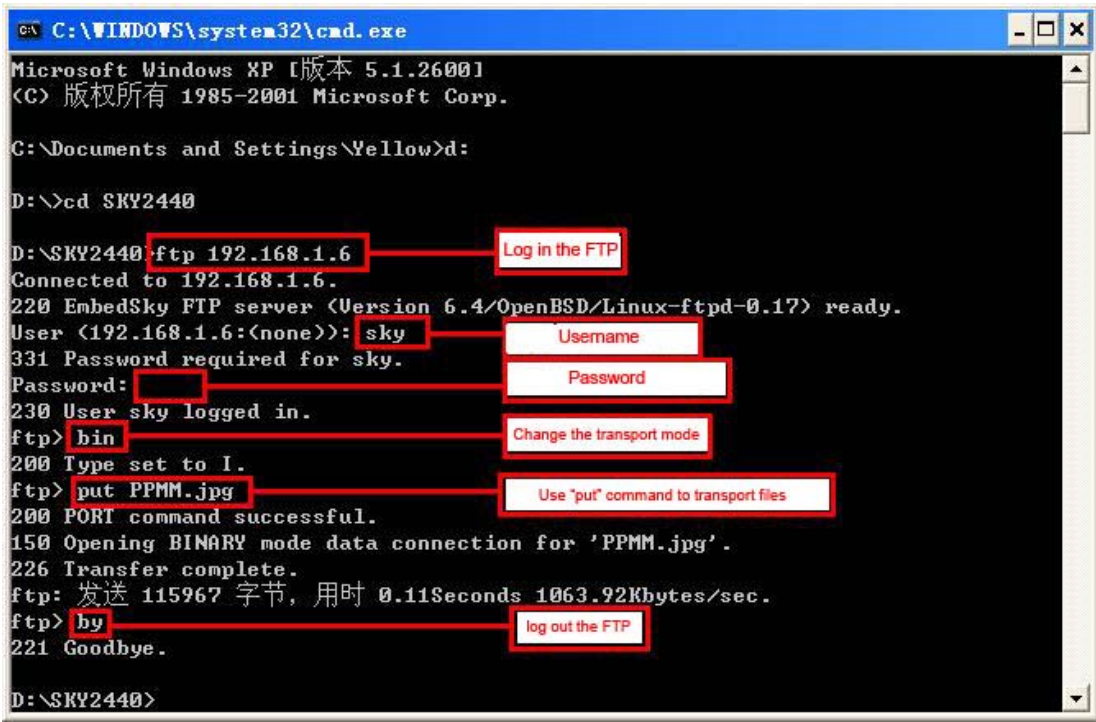
Enter “ftp 192.168.1.6” in command window and press return-key to continue:



```
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\Yellow>d:
D:\>cd SKY2440
D:\SKY2440>ftp 192.168.1.6
```

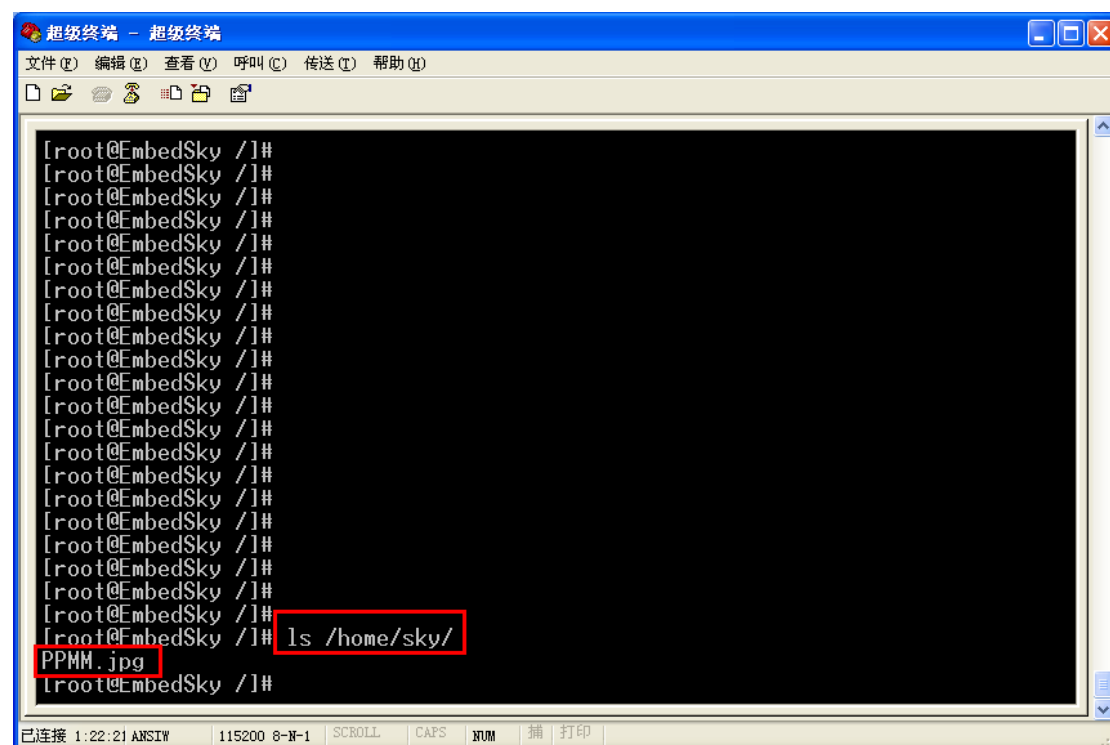
Operating as the following diagram:



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

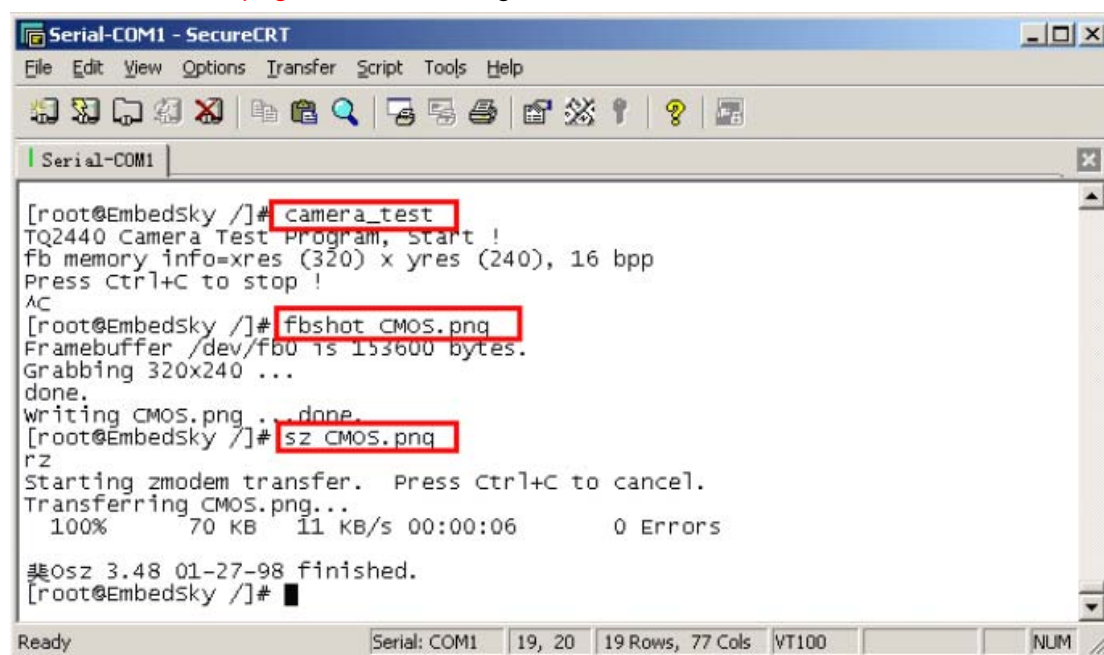
C:\Documents and Settings\Yellow>d:
D:\>cd SKY2440
D:\SKY2440>ftp 192.168.1.6
Connected to 192.168.1.6.
220 EmbedSky FTP server (Version 6.4/OpenBSD/Linux-ftpd-0.17) ready.
User (192.168.1.6:(none)): sky
331 Password required for sky.
Password:
230 User sky logged in.
ftp> bin
200 Type set to I.
ftp> put PPMM.jpg
200 PORT command successful.
150 Opening BINARY mode data connection for 'PPMM.jpg'.
226 Transfer complete.
ftp: 发送 115967 字节, 用时 0.11Seconds 1063.92Kbytes/sec.
ftp> bye
221 Goodbye.
D:\SKY2440>
```

Enter the command “ls /home/sky”, you can find the file “PPMM.jpg”



### 5.14 Experiment of CMOS camera capturing screen test

The CMOS camera must be inserted to the camera interface before starting the board. We can execute the command “#camera\_test” in the console, and use the command “fbshot XXX.png” to save the image.



The following picture is the captured image:



