

QY-9315P 主板

LINUX 用户手册

版本号：V 1.3

杭州启扬智能科技有限公司出品

杭州启扬智能科技有限公司版权所有

2008 年 11 月

目 录

一、准备工作

- 1、主机要求
- 2、硬件配置
- 3、网络准备
- 4、相关文件准备

二、下载和使用 redboot

- 1、下载 redboot
- 2、串口设置
- 3、配置和使用redboot

三、下载 Linux 内核和文件系统

- 1、下载文件系统
- 2、下载内核

四、建立工具链

五、内核编译

六、添加应用程序

七、驱动测试

八、Linux 使用常见问题

一、准备工作

1、主机要求：

- a) 装有Linux系统（redhat7、8、9或其它Linux发行版）和Windows（最好为2000以上系统）；
- b) 至少500MB自由空间；
- c) 一个RS232串口；
- d) 终端通讯软件（minicom或Windows下的超级终端）。

2、硬件配置：

- a) 一根RS232串口线，两端母头的交叉线，三芯就行了（RXD，TXD和GND）。一端连PC机COM1一端连目标板的J501（我们下面的测试都在J501口上进行）接口；
- b) 接上目标板电源；
- c) 找到目标板上的J801跳线，在下面的操作步骤中要用到他们。

3、网络准备：

- a) QY-9315P评估平台支持100M网络速率，可自动分配IP，也可手动分配。用直连网线接入交换机或交叉网线直接接入PC。目标板IP和PC机IP应在同一网段。如：

PC机IP: 192.168.0.56

目标板IP: 192.168.0.55

Network Mask: 255.255.255.0

Broadcast IP: 192.168.0.255

注：以上设置仅在redboot中有效，Linux系统启动之后需要重新分配IP。

- b) 目标板和PC机主要是通过TFTP协议来通信的，因此必须要开启PC机上的TFTP服务。

现以Redhat9.0为例说明TFTP服务器的安装与设置（也可使用Windows下的tftp工具）：

如Linux系统没有安装tftp服务，可以从Redhat9.0第三张安装光盘，在路径/mnt/cdrom/RedHat/RPMS，安装以下两个RPM包：

```
rpm -ivh tftp-0.32-4.i386.rpm
```

```
rpm -ivh tftp-server-0.32-4.i386.rpm
```

```
mkdir /tftpboot
```

让TFTP自启动：

```
/sbin/chkconfig tftp on
```

让TFTP 服务有效

```
/sbin/service xinetd restart
```

如果没有/tftpboot目录，就创建它。这是TFTP服务器默认的传输目录。

```
mkdir /tftpboot
```

chmod 777 /tftpboot有关windows下TFTP的设置请看第三节下载Linux内核和文件系统的相关内容。

4、相关文件准备：在开始下面操作之前，请确认您已经有了下列文件：

download.exe Windows下的烧写redboot 的工具

redboot.bin 编译好的redboot文件

tftpd.exe windows下的TFTP工具（如果您Linux下操作可以不要这个工具）

ramdisk.gz 9315文件系统

zImage 编译好的9315内核

以上这些都做好之后我们就可以开始启动我们的主板了。

二、下载和使用redboot

我们的核心板出厂时已经在flash中写好了redboot，一般情况下是不需要重写redboot 的。要是发生了某种意外，您可能需要重新写入和设置redboot 。具体的操作步骤如下：

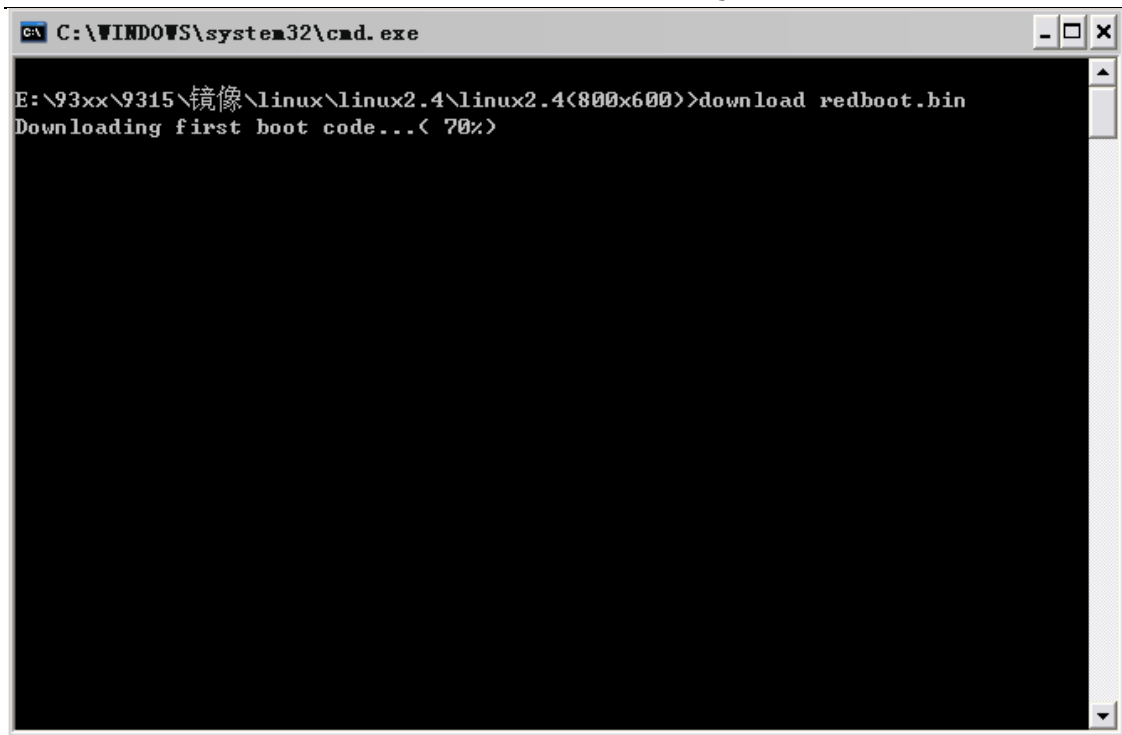
1、下载redboot

断开目标板电源，在PC机上运行download.bat如下图所示：

```

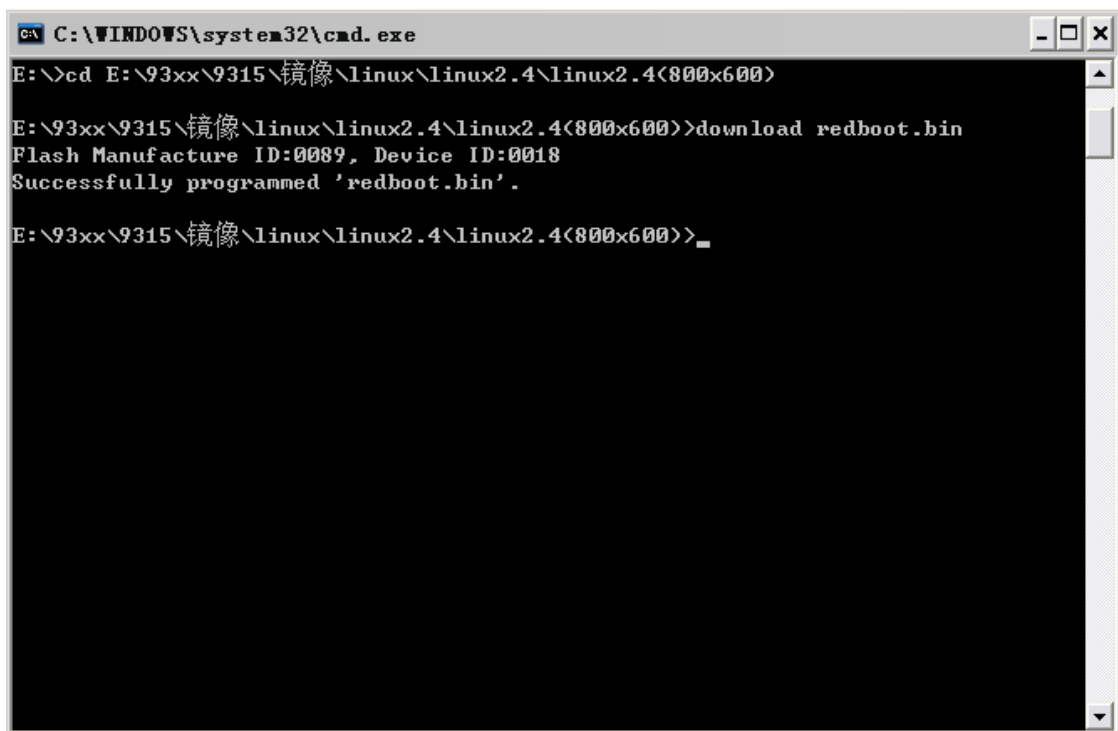
C:\WINDOWS\system32\cmd.exe - download redboot.bin
E:\>cd E:\93xx\9315\镜像\linux\linux2.4\linux2.4(800x600)
E:\93xx\9315\镜像\linux\linux2.4\linux2.4(800x600)>download redboot.bin
Waiting for the board to wakeup..._
  
```

断开J801跳线，接上目标板电源开始下载，系统提示如下图所示：



```
C:\WINDOWS\system32\cmd.exe
E:\93xx\9315\镜像\linux\linux2.4\linux2.4(800x600)>download redboot.bin
Downloading first boot code...< 70%>
```

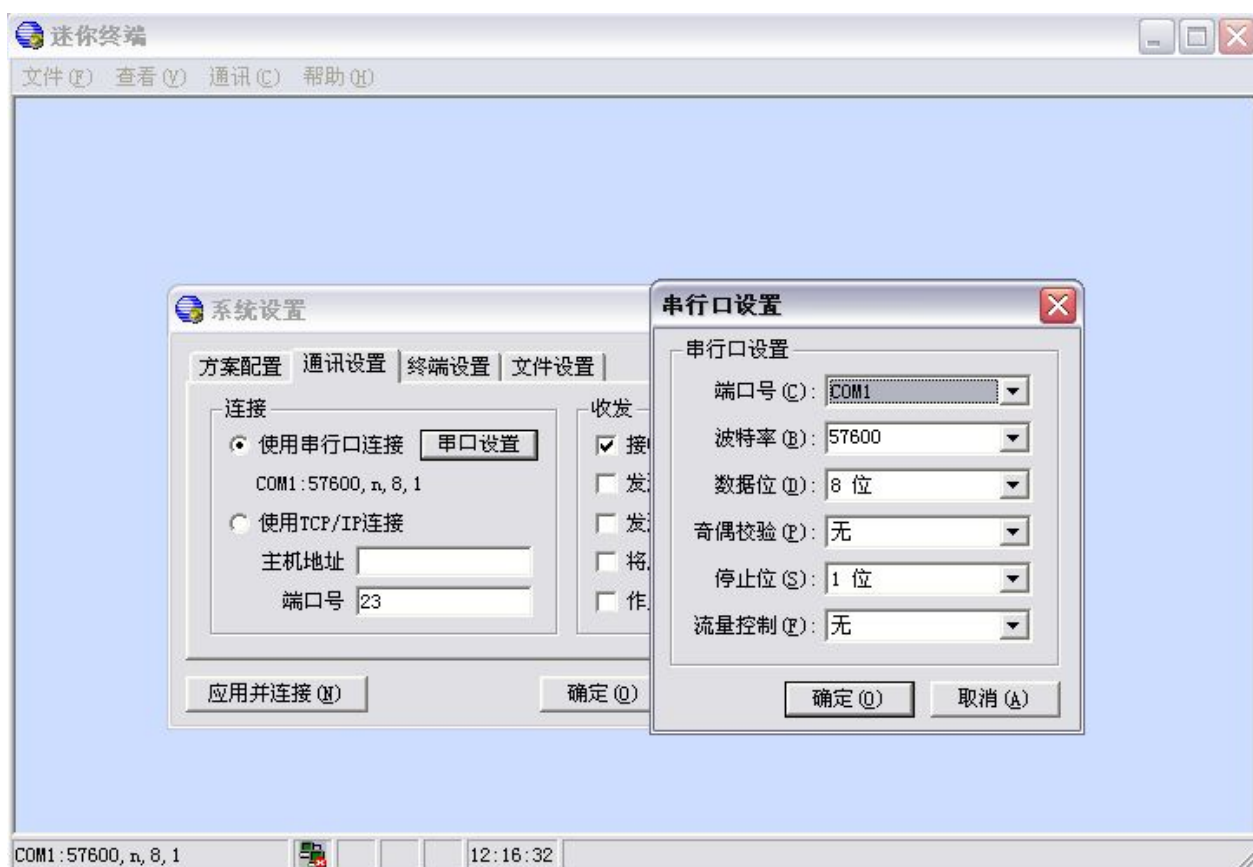
Successfully programmed 'redboot.bin' 则说明下载成功,如下图所示:



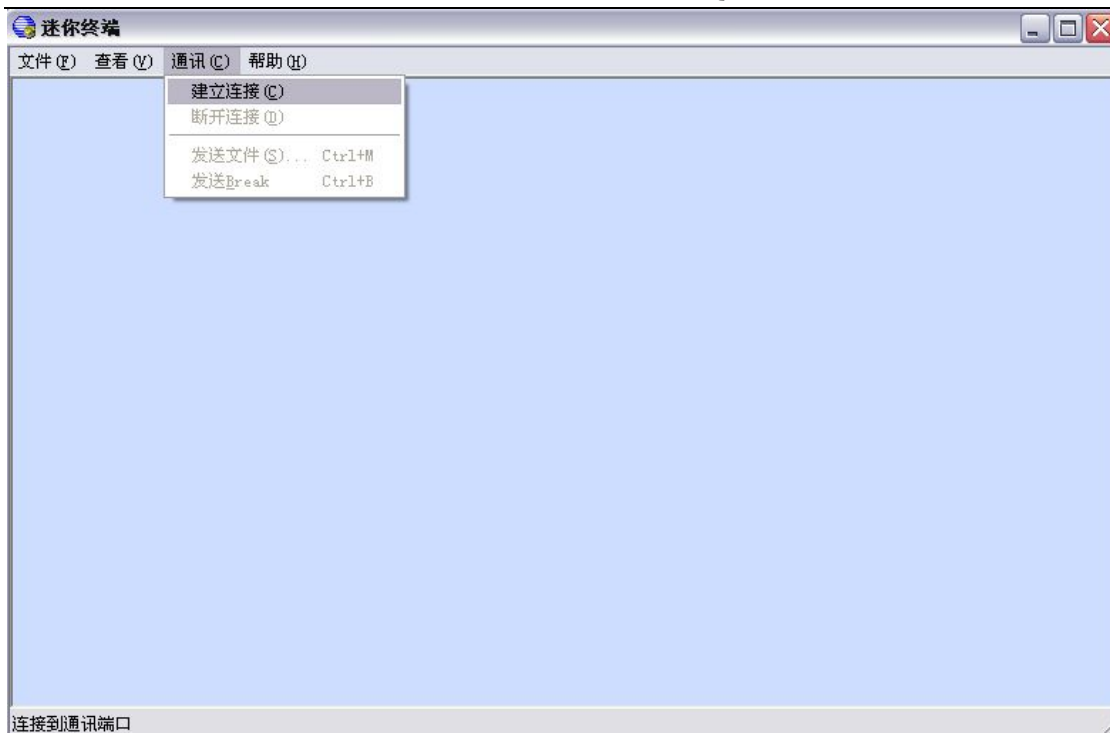
```
C:\WINDOWS\system32\cmd.exe
E:\>cd E:\93xx\9315\镜像\linux\linux2.4\linux2.4(800x600)
E:\93xx\9315\镜像\linux\linux2.4\linux2.4(800x600)>download redboot.bin
Flash Manufacture ID:0089, Device ID:0018
Successfully programmed 'redboot.bin'.
E:\93xx\9315\镜像\linux\linux2.4\linux2.4(800x600)>_
```

2、串口设置

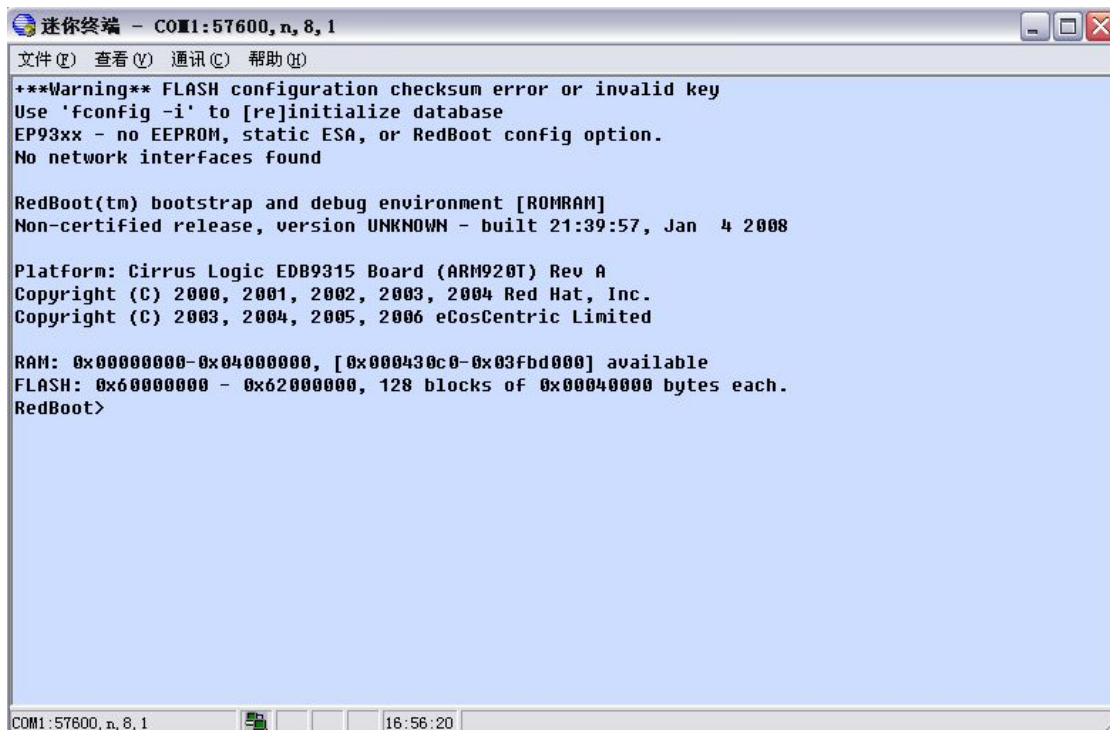
先设置PC上终端通讯程序（此处以迷你终端为例，光盘中附带此程序），打开迷你终端（光盘中附带），设置PC上相应的串口，选择[文件] → [系统设置] → [通讯设置] → 左下角[断开] → [串口设置]，波特率为[57600]，停止位为[1]位，数据位为[8]位，无奇偶校验和流控，如下图所示：



点击迷你终端工具条上的[通讯] → [建立连接]按钮建立连接，如下图所示：



断电，短接J801。接通电源就能在迷你终端下看到如下信息：



这说明我们刚才烧入的redboot能正常工作。

3、配置和使用redboot

烧好redboot之后，我们要对启动脚本进行设置。首先运行fis init进行初始化操作。

```
RedBoot>fis init -f
```

```
About to initialize [format] FLASH image system - continue (y/n)? y
```

运行fconfig -i 命令配置启动脚本如下(其中粗体字为要输入或修改的部分):

```
RedBoot> fconfig -i
```

```
Initialize non-volatile configuration - continue (y/n)? y
```

```
Run script at boot: true
```

```
Boot script: Enter script, terminate with empty line
```

```
>>fis load ramdisk.gz
```

```
>>fis load zImage
```

```
>>exec -r 0x800000 -s 0xb00000 (-s后的参数可根据ramdisk.gz的实际大小调整)
```

```
>>(回车)
```

```
Boot script timeout (1000ms resolution): 1 //等待时间
```

```
Use BOOTP for network configuration: false
```

```
Gateway IP address: 192.168.1.1
```

```
Local IP address: 192.168.1.155 //主板IP
```

```
Local IP address mask: 255.255.255.0
```

```
Default server IP address: 192.168.1.160 // PC机IP
```

```
Set eth0 network hardware address [MAC]: true
```

```
eth0 network hardware address [MAC]: 0x00:0x00:0x00:0x00:0x60:0x34
```

```
GDB connection port: 9000
```

```
Force console for special debug messages: false
```

```
Network debug at boot time: false
```

```
Update RedBoot non-volatile configuration - continue (y/n)? y
```

```
... Erase from 0x61f80000-0x61f81000: .
```

```
... Program from 0x03fbe000-0x03fbf000 at 0x61f80000:
```

```
Redboot>
```

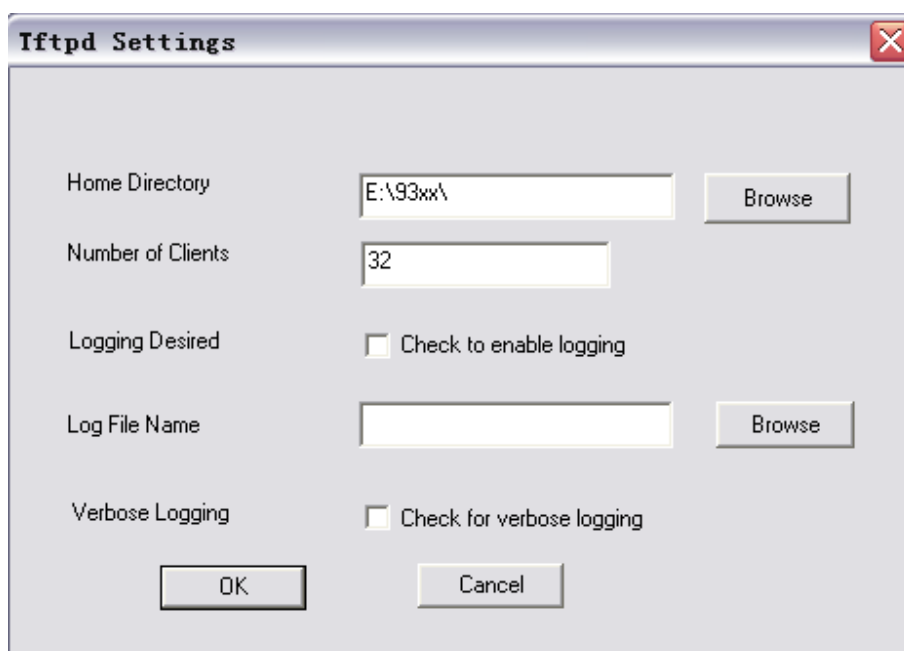
复位一下，使网络配置生效。这样，我们的启动脚本就设置好了。

三、下载 Linux 内核和文件系统：

启动脚本好了之后我们就要将内核和文件系统下载到目标板上。此时先要在PC机上开启TFTP服务，并使TFTP指向目标板所需文件。配置TFTP如下图所示：



点击Configure出现下图：



点击Browse出现下图，选中zImage和ramdisk.gz所在的目录，然后单击“确定”，然后单击“OK”完成TFTP的设置。再击TFTP下面的Start开启TFTP服务：



TFTP设置好了就可以通过TFTP下载文件了。

1、下载文件系统:

RedBoot> **load -r -v -b 0x800000 ramdisk_9315_v5.gz**

将ramdisk_9315_v5.gz 读到内存地址0x800000 \ Raw file loaded

0x00800000-0x00976304, assumed entry at 0x00800000 因为文件比较大, 可能需要一点时间。下载好之后再烧到flash 上面:

RedBoot> **fis create ramdisk.gz**

```
... Erase from 0x60040000-0x60640000: .....
... Program from 0x00800000-0x00e00000 at 0x60040000: .....
... Erase from 0x61fc0000-0x62000000: .
... Program from 0x03fbf000-0x03fff000 at 0x61fc0000: .
```

2、下载内核:

RedBoot> **load -r -v -b 0x80000 zImage**

-Raw file loaded 0x00080000-0x0014991b, assumed entry at 0x00080000

下好之后烧到flash:

RedBoot> **fis create zImage**

```
... Erase from 0x60640000-0x60740000: ....
... Program from 0x00080000-0x00180000 at 0x60640000: ....
... Erase from 0x61fc0000-0x62000000: .
... Program from 0x03fbf000-0x03fff000 at 0x61fc0000: .
```

然后用fis list 就可以查看flash 里面的文件了:

RedBoot> **fis list**

Name	FLASH addr	Mem addr	Length	Entry point
RedBoot	0x60000000	0x60000000	0x00040000	0x00000000
RedBoot config	0x61F80000	0x61F80000	0x00001000	0x00000000
FIS directory	0x61FC0000	0x61FC0000	0x00040000	0x00000000
ramdisk.gz	0x60040000	0x00800000	0x00600000	0x00800000
zImage	0x60640000	0x00080000	0x00100000	0x00080000

文件烧好之后复位目标板就能启动Linux了，迷你终端显示部分信息如下：

```

迷你终端 - COM1:57600, n, 8, 1
文件(F) 查看(V) 通讯(C) 帮助(H)
RedBoot> reset
... Resetting.
+Ethernet eth0: MAC address 0e:00:00:ea:18:f0
IP: 192.168.0.155/255.255.255.0, Gateway: 192.168.0.1
Default server: 192.168.0.68

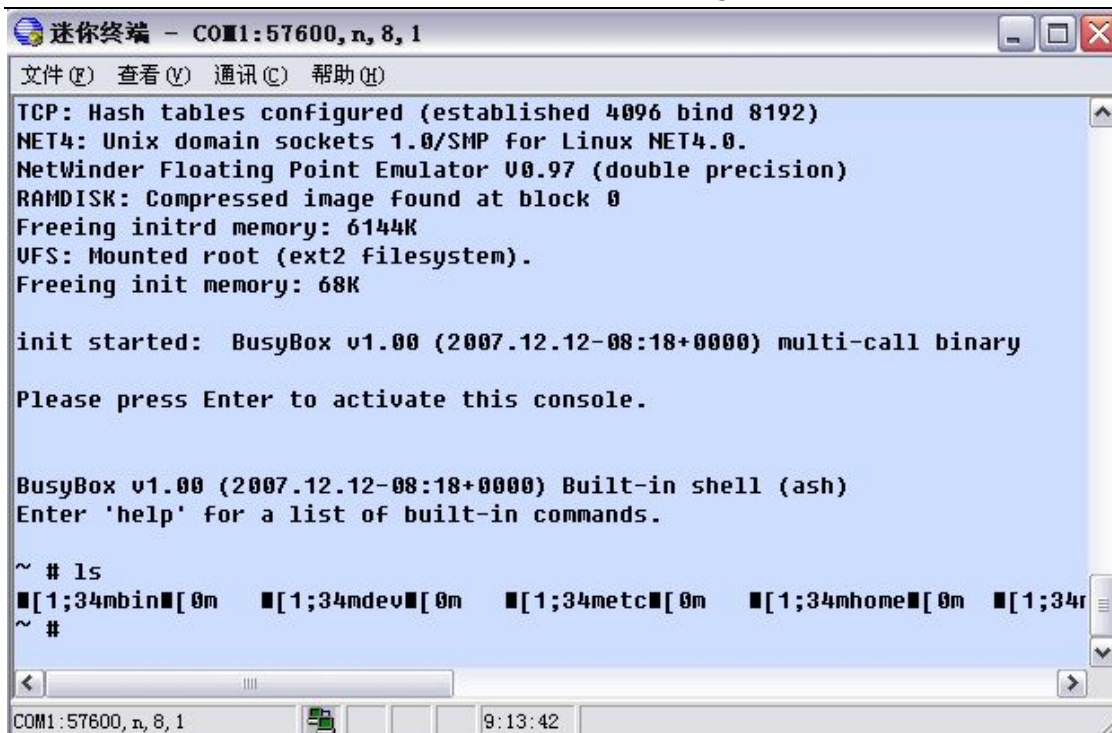
RedBoot(tm) bootstrap and debug environment [ROMRAM]
Non-certified release, version UNKNOWN - built 21:39:57, Jan 4 2008

Platform: Cirrus Logic EDB9315 Board (ARM920T) Rev A
Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.
Copyright (C) 2003, 2004, 2005, 2006 eCosCentric Limited

RAM: 0x00000000-0x04000000, [0x000430c0-0x03fbd000] available
FLASH: 0x60000000 - 0x62000000, 128 blocks of 0x00040000 bytes each.
== Executing boot script in 3.000 seconds - enter ^C to abort
RedBoot> fis load ramdisk.gz
RedBoot> fis load zImage
RedBoot> exec -r 0x8000000 -s 0x6000000
Using base address 0x00080000 and length 0x000cfd00
COM1:57600, n, 8, 1 9:12:11

```

Linux起来之后就能看到shell了,这样整个系统已经成功启动了。



```

迷你终端 - COM1:57600, n, 8, 1
文件(F) 查看(V) 通讯(C) 帮助(H)
TCP: Hash tables configured (established 4096 bind 8192)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
NetWinder Floating Point Emulator V0.97 (double precision)
RAMDISK: Compressed image found at block 0
Freeing initrd memory: 6144K
UFS: Mounted root (ext2 filesystem).
Freeing init memory: 68K

init started: BusyBox v1.00 (2007.12.12-08:18+0000) multi-call binary

Please press Enter to activate this console.

BusyBox v1.00 (2007.12.12-08:18+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

~ # ls
[1;34mbin][0m [1;34mdev][0m [1;34metc][0m [1;34mhome][0m [1;34r
~ #
    
```

四、建立工具链

在PC机上编译目录板所用的Linux内核和其他应用程序时要用到交叉编译工具，我们的主板光盘中有做好的交叉编译工具链。我们的交叉编译工具是

arm-linux-gcc-3.4.tar.bz2，用如下命令打开压缩包：

```
# tar jxvf arm-linux-gcc-3.4.tar.bz2
```

然后把它移到/usr/local/arm 目录下

```
# mv usr/local/arm /usr/local/
```

```
# mv opt/buildroot /opt
```

这样，我们的工具链就装好了。

五、编译内核

主板光盘中有配置好的内核文件linux-2.6.8.tar.gz 压缩包，用如下命令打开：

```
# tar zxvf linux-2.6.8.tar.gz
```

进入Linux 目录：

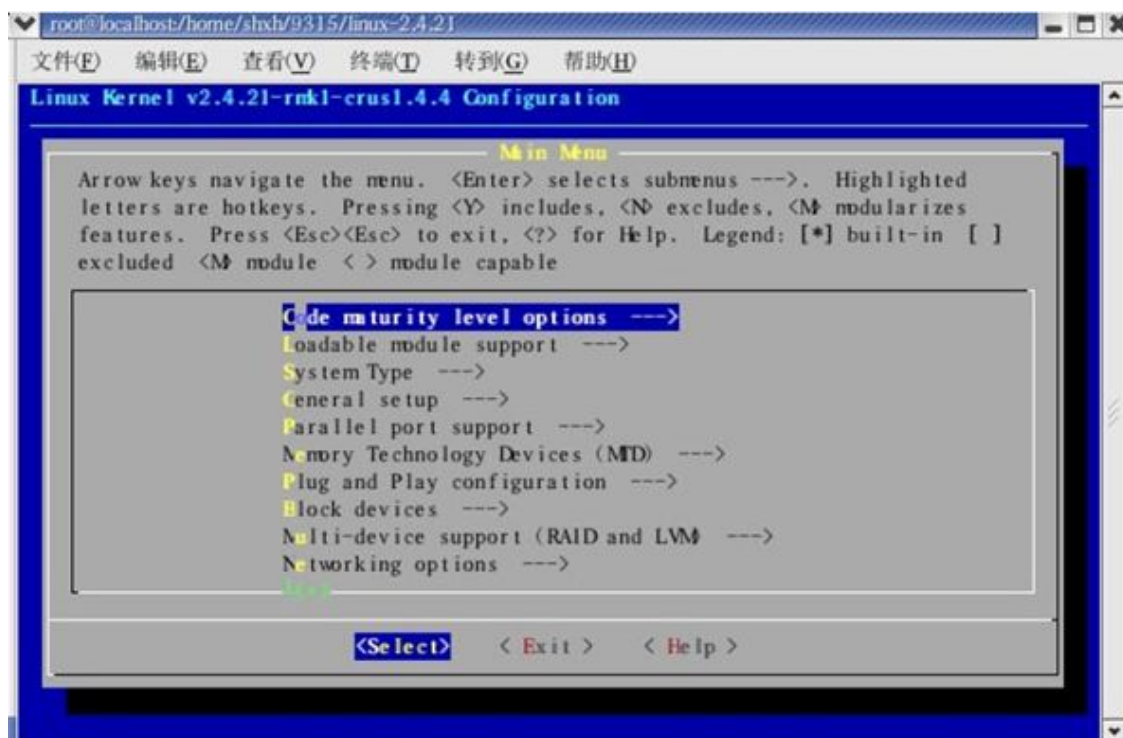
```
# cd linux-2.6.8
```

设置交叉编译工具路径等环境变量：

```
# source setenv
```

然后用make menuconfig 选择内核配置：

make menuconfig 您也可以对内核功能选项做必要的调整，以满足您的需要，好了之后保存退出。



最后就可以编译内核了，用命令make zImage：

```
#make zImage
```

编译时间比较长，依您的机器而定。完了之后我们就可以在arch/arm/boot/目录下找到编好的内核zImage 了。

六. 添加应用程序

当您需要在主板上扩展软件功能的时候您可能需要添加额外的应用程序或驱动程序。下面就简单介绍一下如何在主板上添加程序。

在进行操作之前我们假设您已经有了程序的源代码了。Linux 是一个开源的操作系统，大多程序的源代码都能在网上开源社区里找到。如国内的 <http://www.chinaunix.com/> 、 <http://www.linuxforum.net/> 国外的有

<http://www.linux.org/>、<http://www.linux.com/> 和<http://www.linuxforum.com/> 等，您可以在这些地方找到您所需要的东西，如果您有什么问题也可以在这些论坛里得到帮助。

应用程序的编译：

在 PC 机上为我们的目标版开发应用程序一定要进行交叉编译，一般来讲我们的 PC 机和目标版的架构是不一样的。

比如一个 Hello World 程序代码如下：

```
#include
main() {
printf("Hello World ! \n");
}
```

首先移到 `hello.c` 文件所在的目录，我们用如下命令对它进行交叉编译：

```
#/usr/local/arm/3.4/bin/arm-linux-gcc -O3 -o hello.out hello.c 其中
```

`/usr/local/arm/3.4/bin/arm-linux-gcc` 是交叉编译工具的路径，这条命令会生成一个 `hello.out` 文件，这个文件就是我们得到的可执行文件（跟 windows 下的 .exe 文件一样，只不过这个文件不能在 PC 机上运行，它是为目标板编译的）。这样我们就得到了可以在主板上执行的程序了，下面讲解如何将它下载到目标板上去执行。

下载并运行：

当我们得到所需要的应用程序之后，我们就要想办法将它弄到主板上运行了。这里也有两种方法可以实现。

1、通过 TFTP 下载到目标板

这样做首先要求我们的目标板跑起 Linux，并用网线和 PC 机相连。在 PC 机上开启 tftp 服务，将 `hello.out` 文件放到 tftp 目录下，给目标板设置好 IP 后就可以用如下命令下载程序了：


```
~#tftp -g -r hello.out 192.168.0.56
```

这里 `hello.out` 是我们前面编译好的 `hello world` 文件，`192.168.0.56` 是我的 PC 机的IP。然后你就可以用 `ls` 命令看到当前目录下有 `hello.out` 这个文件了。运行它就能看到打印出来的信息了：

```
~#chmod 777 hello.out
```

```
~#./ hello.out
```

```
Hello world !
```

这种方法十分方便我们调试程序。

2、编入文件系统

```

~ # ifconfig eth0 192.168.0.55
~ # tftp -g -r a.out 192.168.0.56
~ # ls
a.out      dev      funky.mp3 lib      proc      tmp      var
bin        etc      home     mt       sbin     usr
~ # ./a.out
sh: ./a.out: Permission denied
~ # chmod 777 a.out
~ # ./a.out
Hello World !
~ #

```

当然，如果需要，您也可以将您的应用程序直接编入文件系统，这样您就可以把它和文件系统一起烧到 `flash` 上了。文件系统在光盘的 `image` 目录下，名为 `ramdisk.gz` 首先解开压缩：

```
$ gunzip ramdisk.gz
```

映像文件挂载

```
$ mount -o loop ramdisk /mnt/your_ramdisk_directory
```

对 `/mnt/your_ramdisk_directory` 目录进行操作

```
$ cd /mnt/your_ramdisk_directory
```

```
$ do_whatever_you_want(create directories,files ... //加入您的应用程序
```

到您的映像文件目录下

```
$ cd where_your_ramdisk_file_is
```

卸装文件系统

```
$ umount /mnt/your_ramdisk_directory
```

压缩文件系统，生成最终的文件系统映像

```
$ gzip ramdisk
```

这样，您的应用程序就做到文件系统上了，您可以重新烧写文件系统到目录板上，您的应用程序就能在目标板上运行了。

七. 驱动测试:

QY-9315P Linux BSP 包含了以下接口驱动程序:

- 两路三线 RS232
- 两路 RS485（与三线 RS232 复用）
- LCD
- 触摸屏接口
- 三路 USB host 接口
- 网络接口
- 音频接口
- 16 路 IO 接口
- 硬盘及 CF 卡接口
- 实时时钟
- 蜂鸣器
- 看门狗

串口驱动（RS232）测试:

串口驱动在内核中的配置

```
Character devices --->
```

```
Serial drivers --->
```

```

<> 8250/16550 and compatible serial support
[ ] Console on 8250/16550 and compatible serial port
[ ] RightSky PCM 9861 support
(4) Maximum number of non-legacy 8250/16550 serial ports
[*] Extended 8250/16550 serial driver options
[*] Support more than 4 legacy serial ports
[*] Support for sharing serial interrupts
[ ] Autodetect IRQ on standard ports (unsafe)
[*] Support special multiport boards
[ ] Support RSA serial ports
--- Non-8250 serial port support
<*> EP93xx serial port support
[*] Support for console on EP93xx serial port
< > Device driver for SST 25LF0x0 serial flash on Cirrus Logic EP93xx boards
    
```

8250/16550 和 EP93xx 分别对应扩展 9 线串口和 CPU 自带的 3 线串口。

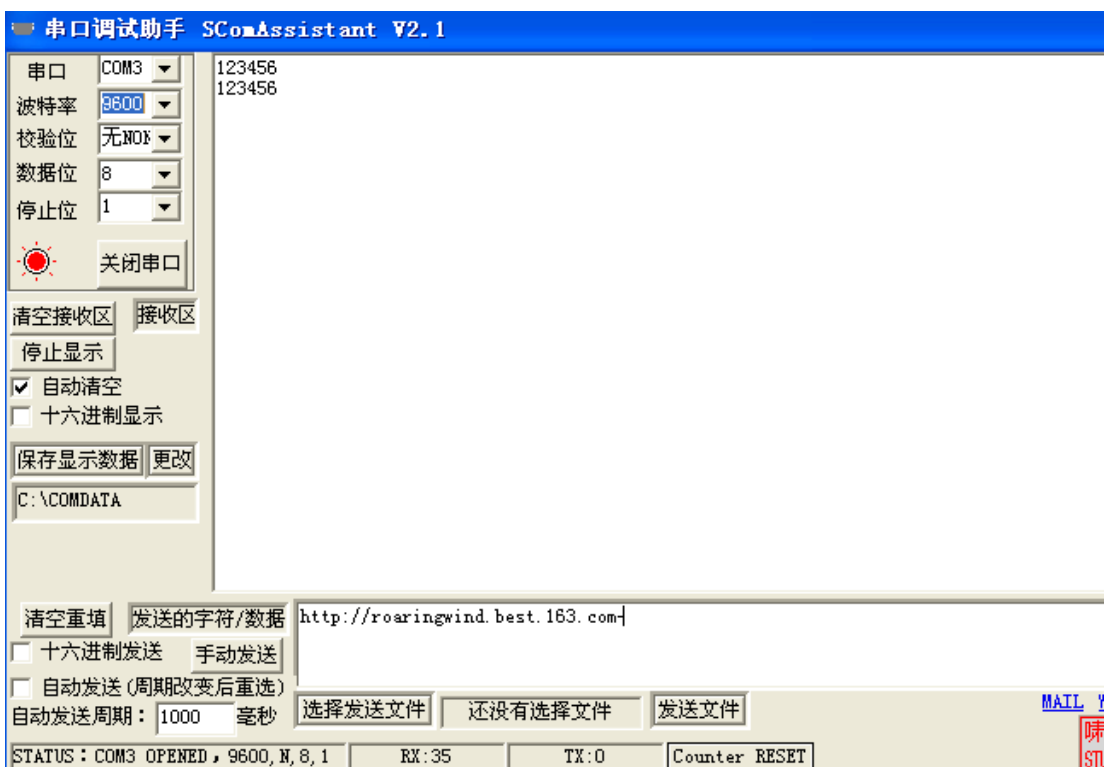
J502 对应的串口为两路 3 线串口，Linux 下设备名为 ttyAM1、ttyAM2。

测试方法：在 PC 上运行“串口调试助手”之类工具软件，在板子上运行

```
~ # stty -F /dev/ttyAM1 9600 （设置波特率）
```

```
~ # echo 123456 >/dev/ttyAM1 （发送测试）
```

可以看到串口调试工具里收到“123456”字符。



在板子上运行

```
~ # stty -F /dev/ttyAM1 raw
```

```
~ # cat /dev/ttyAM1 （接收测试）
```

用串口调试工具发送数据，可以看到板子上可以接收到数据。

```
~ # cat /dev/ttyAM1
http://roaringwind.best.163.com
```

其他串口只需要更改设备名称，可进行同样测试。

用程序操作串口的方法，可以参照光盘中驱动测试程序目录下的源码。

RS485 驱动测试:

RS485 在内核中的配置

Character devices --->

```
[*] Ep93xx PCI04 Support
[*] Ep93xx RS485 Support
[*] Ep93xx BUZZER Support
```

测试 RS485 需要 485 设备，没有 485 设备的情况下，可以将主板上的两路 485 直连（用跳线帽将 J503 1、2 脚 3、4 脚短接），测试 RS485 接口。运行 ./rs485_test 0 是测试串口 1（ttyAM1）发送，串口 2（ttyAM2）接收，执行 ./rs485_test 1 是测试串口 2 发送，串口 1 接收。

执行 ./rs485_test 0 ，输入要发送的字符，回车后，可以看到终端上显示接收到的字符。

```
/usr/test/ # ./rs485_test 0
```

```
rs485 test ...
```

```
12jk
```

```
12jk
```

```
sadfsfsfds
```

```
sadfsfsfds
```

```
/usr/test/ # ./rs485_test 1
```

```
rs485 test ...
```

```
asjkldf
```

```
asjkldf
```

```
asdfklasf
```

```
asdfklasf
```

LCD 驱动测试:

Framebuffer 驱动在内核中的配置

Graphics support --->

```
[[*] support for frame buffer devices
[*]   EP93xx frame buffer support
      EP93xx frame buffer display (LCD 800x600 display) --->
      EP93xx frame buffer depth (16bpp true color) --->
< > Virtual Frame Buffer support (ONLY FOR TESTING!)
      console display driver support --->
      logo configuration --->
```

在 EP93xx frame buffer display (LCD 800x600 display) ---> 选项下，我们提供了常见的一些分辨率的选项，但是这些设置并不适应所有 LCD，有可能要根据具体的屏来进行调整。

将 LCD 与主板按正确方式连接（连接方式参照光盘中提供的主板与相应屏的连接方法），启动主板完成后，会看到 LCD 上或者显示器上显示 Linux 的 LOGO——小企鹅。

运行 QT 示例程序:

```
~ # cd /usr/qt/demo/canvas
```

```
/usr/qt/demo/canvas # ./canvas -qws
```

可以查看 LCD 显示图形效果。

触摸屏接口驱动测试:

触摸屏驱动在内核中的配置

Input device support --->

```

(768) Vertical screen resolution
< > Joystick interface
<*> Touchscreen interface
(1024) Horizontal screen resolution
(768) Vertical screen resolution
    
```

触摸屏使用之前需要校准。正确连接触摸屏，执行 `ts_calibrate`，按照 LCD 上的提示依次点击屏幕上五个点，可以生成校准数据文件 `/etc/pointercal`。

```
~ # ts_calibrate
```

```
xres = 800, yres = 600
```

```
Took 1 samples...
```

```
Top left : X = 941 Y = 856
```

```
Took 1 samples...
```

```
Top right : X = 971 Y = 3300
```

```
Took 1 samples...
```

```
Bot right : X = 3223 Y = 3307
```

```
Took 1 samples...
```

```
Bot left : X = 3239 Y = 875
```

```
Took 1 samples...
```

```
Center : X = 2026 Y = 2106
```

```
-196.413452 -0.001582 0.287105
```

```
-155.544556 0.219606 -0.000592
```

```
Calibration constants: -12872152 -103 18815 -10193768 14392 -38 65536
```

触摸屏校准完成后，可以运行 `ts_test`、`ts_harvest` 等触摸屏测试程序。触摸屏校准及测试采用的 `ts_lib` 源码在光盘中源码目录下。

运行 QT demo 程序：

```
/usr/qt/demo/canvas # export QWS_MOUSE_PROTO='TPanel:/dev/misc/ep93xx_ts'
```

```
/usr/qt/demo/canvas # ./canvas -qws
```

可以在 QT 中使用触摸屏。

USB host 接口测试:

USB 驱动配置

USB support --->

将 USB 鼠标接入主板 USB 接口 J505 的任一接口,可以看到发现 USB 设备的提示:

```
~ # usb 1-1: USB disconnect, address 2
usb 1-1: new low speed USB device using address 3
drivers/usb/input/hid-core.c: ctrl urb status -32 received
input: USB HID v1.10 Mouse [USB OpticalWheel Mouse] on usb-ep93xx_usb-1
```

运行 qt 示例程序

```
~ # export QWS_MOUSE_PROTO='USB:/dev/input/mouse0'
```

```
~ # cd /usr/qt/demo/canvas
```

```
/usr/qt/demo/canvas # ./canvas -qws
```

可以看到 USB 鼠标可以使用了。

将 U 盘插入 USB 接口 J505 的任一接口,同样可以看到法相 USB 设备的提示:

```
usb 1-1: new full speed USB device using address 2
Initializing USB Mass Storage driver...
usbcore: registered new driver usb-storage
USB Mass Storage support registered.
scsi0 : SCSI emulation for USB Mass Storage devices
Vendor: Kingston   Model: DataTraveler 2.0   Rev: 1.00
Type:   Direct-Access                ANSI SCSI revision: 02
SCSI device sda: 3905407 512-byte hdwr sectors (2000 MB)
sda: Write Protect is off
sda: assuming drive cache: write through
sda:<7>usb-storage: queuecommand called
sda1
Attached scsi removable disk sda at scsi0, channel 0, id 0, lun 0
```

挂载 U 盘

```
~ # mount -t vfat /dev/sda1 /mnt
```

```
~ # cd mnt
```

```
/mnt # ls
```

```
l.avi                Drivers              manual
???.cdr_???         TAV_16.1_Guidebook.pdf  superuser
Autorun.inf         TAV_32Bit_Kingston.exe  uptitle.txt
DDR FOR 128         ite8888             vga
DDR FOR 256         lan
```

可以看到 U 盘中的内容。

网络接口驱动测试:

将主板的网络接口用直连网线与路由器、交换机连接或者交叉线与 PC 机连接，可以测试主板的网络驱动。

主板启动完成后，设置主板 IP 与 PC 机同一网段，测试与主机连接

```
~ # ifconfig eth0 192.168.0.222
```

```
~ # ping 192.168.0.16
```

```
PING 192.168.0.16 (192.168.0.16): 56 data bytes
```

```
84 bytes from 192.168.0.16: icmp_seq=0 ttl=128 time=1.1 ms
```

```
84 bytes from 192.168.0.16: icmp_seq=1 ttl=128 time=0.5 ms
```

```
84 bytes from 192.168.0.16: icmp_seq=2 ttl=128 time=0.3 ms
```

主板设置是启动后自动获取 IP，需要所在网络中有 DHCP 服务，如需启动后自动设置 IP，可以在/etc/rc.sysinit 文件中将

```
if [ "x`grep ip= /proc/cmdline`" = "x" ]
```

```
then
```

```
udhcpc -i eth0 -b -p /var/run/udhcpc.eth0.pid >/dev/null 2>&1
```

```
fi
```

替换为 ifconfig eth0 192.169.0.222（要设的 IP 地址）

音频接口测试:

运行

```
~ # aplay wav 文件.wav
```

或者

```
~ # madplay mp3 文件.mp3
```

可以听到播放的声音。

GPIO/矩阵键盘接口测试:

主板上 J504 接口为 GPIO/矩阵键盘接口,在内核配置选项中,可以选择用 GPIO 的驱动或者矩阵键盘的驱动。

Character devices --->

```

[ * ] EP93xx GPIO Configure (EP93xx GPIO Support) --->
[ * ] EP93xx PC104 Support

(X) EP93xx GPIO Support
( ) EP93xx 8x8 Matrix Keypad
    
```

默认是配成 GPIO 模式。

对 GPIO 的操作可以参照测试程序 gpio_test 中的源码。

矩阵键盘测试:

如将内核中 GPIO 选项配置成矩阵键盘模式,则可以对矩阵键盘进行测试。

将矩阵键盘与主板连接,运行键盘测试程序,可以看到当有键按下时,程序可以读出键值。

```

/usr/test # ./keypad_test
Watting event ...
Keyboard key, the code is : 1      Press down.
Watting event ...
Watting event ...
Keyboard key, the code is : 1      Press up.
    
```

Watting event ...

Watting event ...

Keyboard key, the code is : 4 Press down.

Watting event ...

Watting event ...

Keyboard key, the code is : 4 Press up.

Watting event ...

Watting event ...

矩阵键盘按键与键值的映射关系可以修改内核中 `drivers/char/ep93xx_keypad.c` 中

“`static unsigned int const KeyPadTable[KeyPad_SIZE] =`” 的内容。

IDE 接口测试:

将 CF 卡插入 CF 卡插座 J402 中，启动主板，可以在调试信息中看到发现 CF 卡

`hda: Flash Card, ATA DISK drive`

`device only supports MDMA ? (we're hosed)`

`Using anticipatory io scheduler`

`ide0 at 0x800-0x807,0x406 on irq 40`

`hda: max request size: 128KiB`

`hda: 254464 sectors (130 MB) w/0KiB Cache, CHS=994/8/32`

`hda: hda1`

挂载 CF，可以查看 CF 卡中文件。

```
~ # mount -t vfat /dev/hda1 /mnt
```

```
~ # cd /mnt
```

```
/mnt # ls
```

```
???0.bmp                      ???1.bmp                      format.com
```

```
???1.bmp                      ???1.bmp                      gyjfp.exe
```

同样将硬盘用 40 芯排线与主板上 J401 连接，也可以对 IDE 接口测试。

如果同时使用硬盘与 CF 卡，需将硬盘设置成 slave 模式。

实时时钟:

RTC 在内核中配置:

Character devices --->

```
[*] Ep93xx RTC support
```

运行实时时钟测试程序:

```
#cd /usr/test/  
/usr/test/#./rtc_test  
Do you want to Reset Time(Y/N)y  
year(2000-2099):2008  
month(1-12):10  
date(1-31):9  
week(1-7):4  
hour(0-23):12  
minute(0-59):59  
second(0-59):00  
Your input string is =1073759044 34384 1 33948 -1090519660 33595 68524  
Now is = '9-Oct-2002008 12:59:00 Thursday '  
Now is = '9-Oct-2002008 12:59:01 Thursday '  
Now is = '9-Oct-2002008 12:59:02 Thursday '  
Now is = '9-Oct-2002008 12:59:03 Thursday '  
Now is = '9-Oct-2002008 12:59:04 Thursday '  
Now is = '9-Oct-2002008 12:59:05 Thursday '  
Now is = '9-Oct-2002008 12:59:06 Thursday '  
Now is = '9-Oct-2002008 12:59:07 Thursday '
```

设置实时时钟时间，断电重启，再次运行 `rtc_test`，可以发现时间是新设置的时间。

蜂鸣器测试:

Character devices --->

```
[*] Ep93xx BUZZER Support
```

运行蜂鸣器测试程序

```
#cd /usr/test/
```

```
/usr/test/#./buzzer_test
```

可以听到蜂鸣器响声。

看门狗测试:

Character devices --->

Watchdog Cards --->

```
[*] Watchdog Timer Support
[*] Disable watchdog shutdown on close
[*] EP93XX WATCHDOG
--- Watchdog Device Drivers
< > Software watchdog
--- USB-based Watchdog Cards
< > Berkshire Products USB-PC Watchdog
```

内核中看门狗喂狗最大间隔设置为 30s，一旦开启看门狗，30s 之内不喂狗的话，

系统就会自动重启。运行测试程序

```
~#cd /usr/test/
```

```
/usr/test/#./wdt_test
```

程序一直运行，板子不会重启，用 CTRL+C 结束程序后，主板会在 30s 以内重启。

八. Linux 常见问题:

1.jffs2文件系统:

Flash的分区信息由redboot生成。在内核启动时可以看到如下的启动信息。

```
0x00000000-0x00040000 : "RedBoot"  
0x00040000-0x00800000 : "ramdisk.gz"  
0x00800000-0x00940000 : "zImage"  
0x00940000-0x01fc0000 : "unallocated"  
0x01fc0000-0x01fff000 : "FIS directory"  
0x01fff000-0x02000000 : "RedBoot config"
```

其中unallocated（未分配）即为可以使用的空间。

执行 `mount -t jffs2 /dev/mtdblock3 /mnt/flash`

在/mnt/flash目录下的内容就是写在flash中，掉电可以保存的。

第一次挂载应先执行

```
~ # flash_eraseall /dev/mtd3, 以擦除该分区上原有的内容。
```

2.自启动应用程序:

自启动应用程序可以添加在/etc/rc.sysinit 中

“`stty -F /dev/ttyAM0 ospeed 57600`” 句之前。

杭州启扬智能科技有限公司

电话：0571-87858811 / 87858822

传真：0571-89935912

支持：0571-89935913

E-MAIL: supports@qiyangtech.com

网址： <http://www.qiyangtech.com>

地址： 杭州市西湖科技园西园一路 8 号 2 幢 5 层

邮编： 310013