

QY-9263K 评估板

LINUX2.6 用户手册

版本号：V 1.0

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 2 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 u-boot

- 1、下载u-boot
- 2、配置和使用u-boot

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

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RS2323 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J7(我们下面的测试都在J7口上进行)接口；
- b) 接上目标板电源；
- c) 找到核心板上的J0跳线，在下面的操作步骤中要用到他们。

3、网络准备：

- a) QY-9263K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

- 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

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：

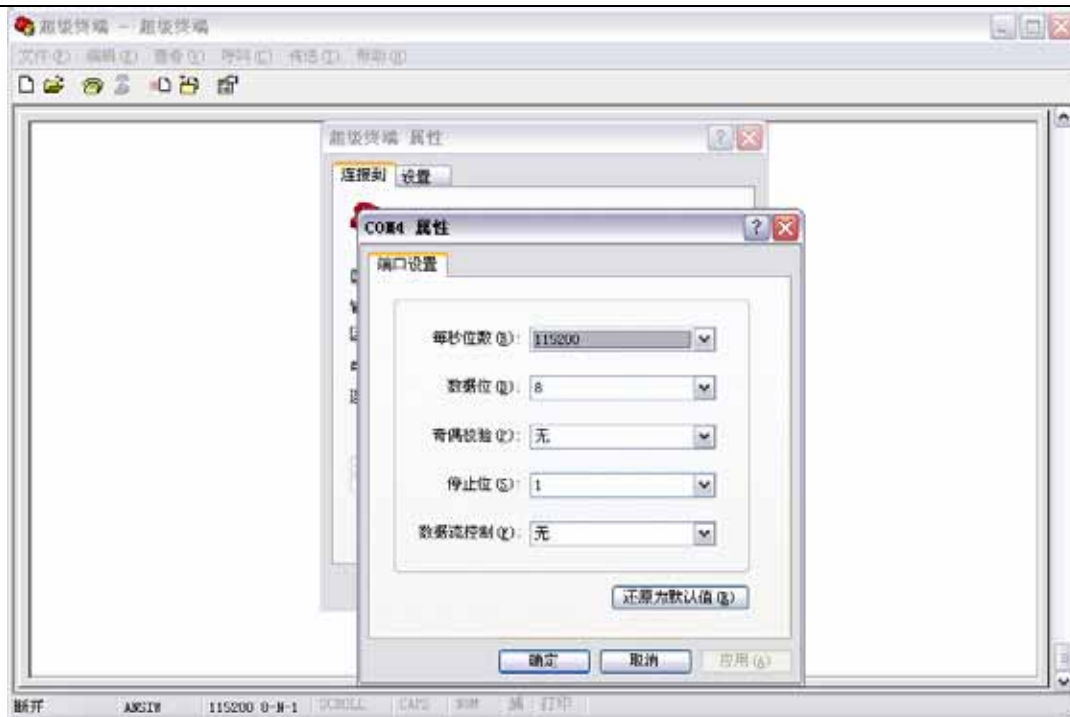
SAM-BA 2.6	windows 下的烧写u-boot 的工具
dataflash_sam9263.bin	在uboot之前的启动程序
u-boot.bin	编译好的u-boot 文件
tftpd.exe	windows 下的TFTP 工具（当然，如果您在linux 下操作可以不要这个工具）
ramdisk.gz	SAM9263的文件系统
ulmage	编译好的SAM9263内核

以上这些都做好之后我们就可以开始启动我们的主板了。

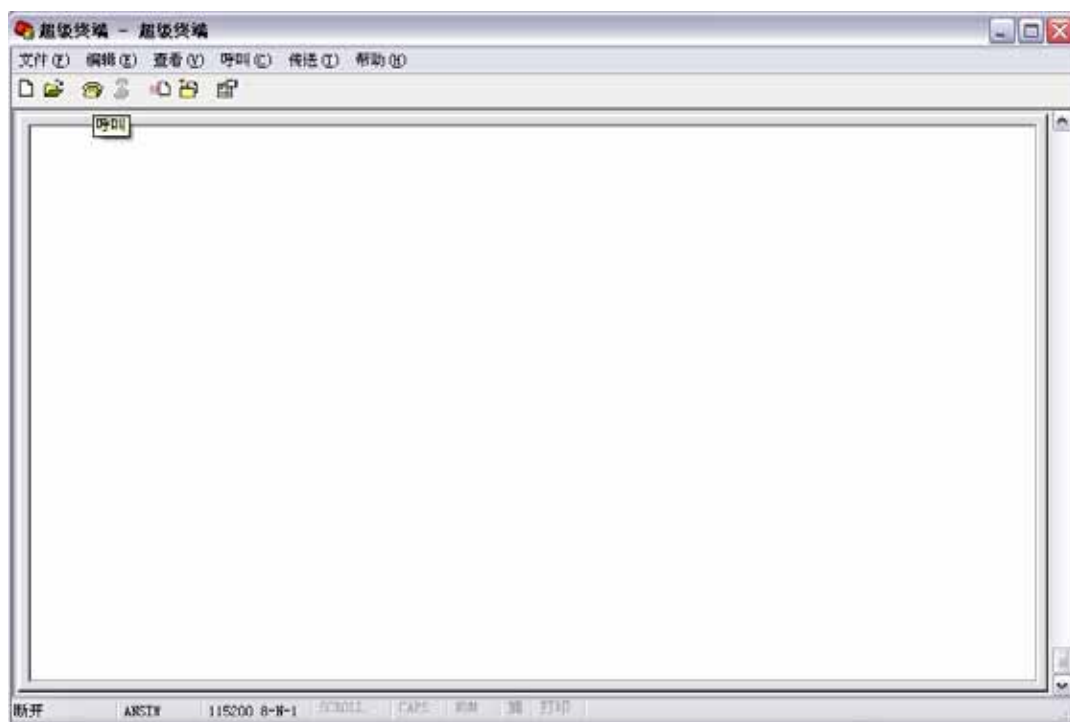
二、下载和使用u-boot

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

- (1)打开超级终端，新建一个连接，设置 PC 上相应串口波特率为[115200]，[1]位停止位，无奇偶校验和流控。如下图所示：

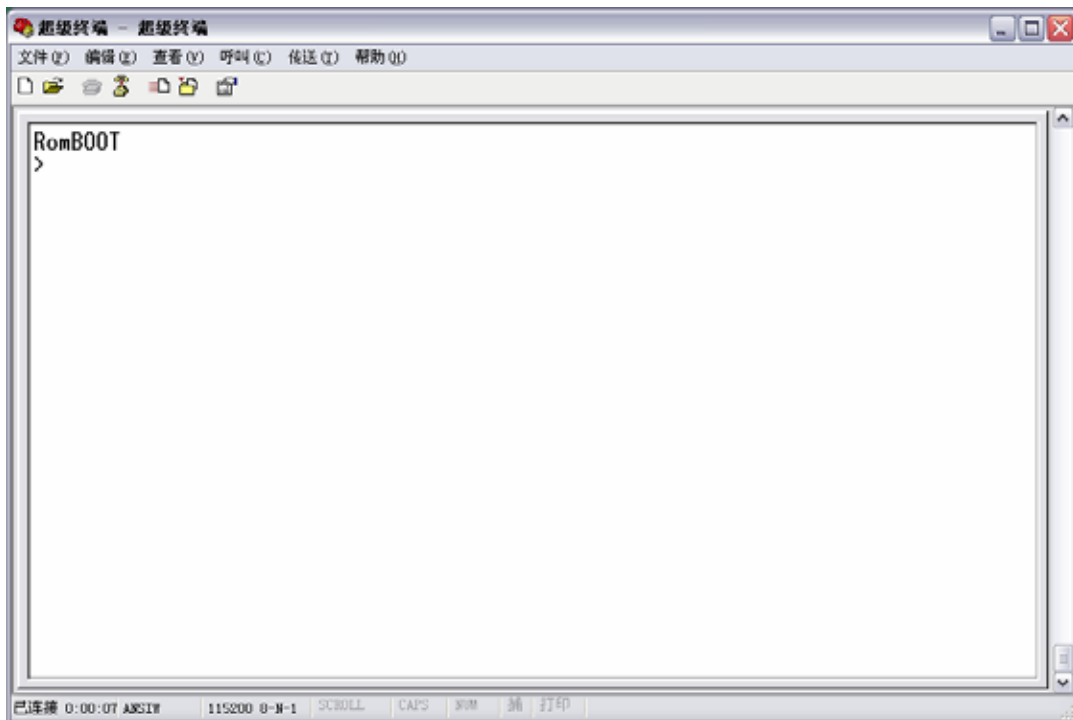


(2) 点击工具条上的[呼叫]按钮或通过主菜单中的呼叫选项建立连接：



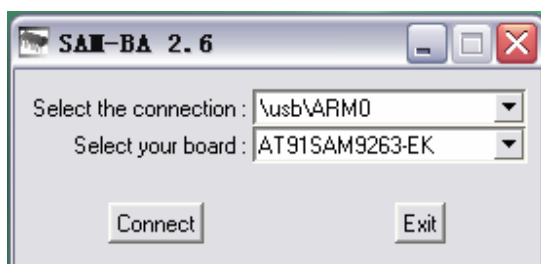
- (3) 将核心板上的J0跳线帽摘除,使能CPU内部SAM-BA下载功能,将交叉串口线分别一端接PC机上的COM1口,另一端接QY-9263K板的调试串口(J7端口)。(注意不要使用其他的串口通讯程序打开PC上QY-9263K主板相连接的串口)。给QY-9263K主板上电。

- (4) 在超级终端上出现提示符: ROMBOOT>, 如下图

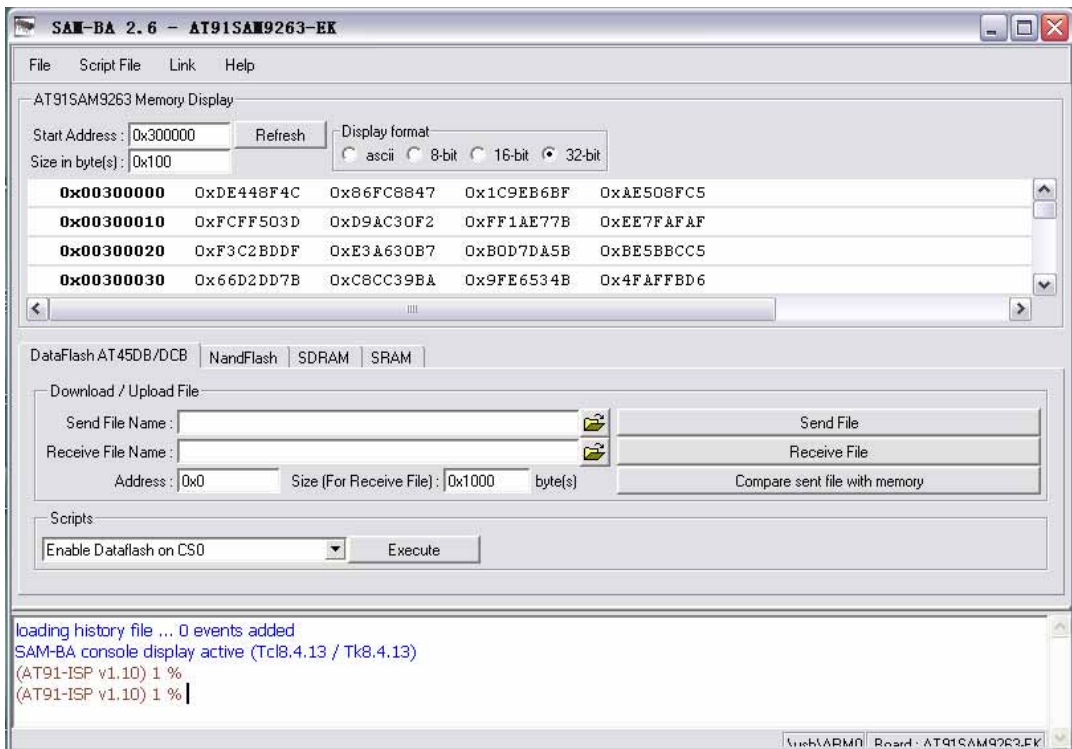


- (5) 将 USB 设备线一端接 PC 机上的 USB 口 ,另一端接 QY-9263K 板的调试 USB 设备接口(J20 端口)。此时, 在 PC 上会提示有 USB 设备插入, 如果需要安装驱动, 请按照默认安装。

- (6) 运行 SAM-BA.exe , 如下



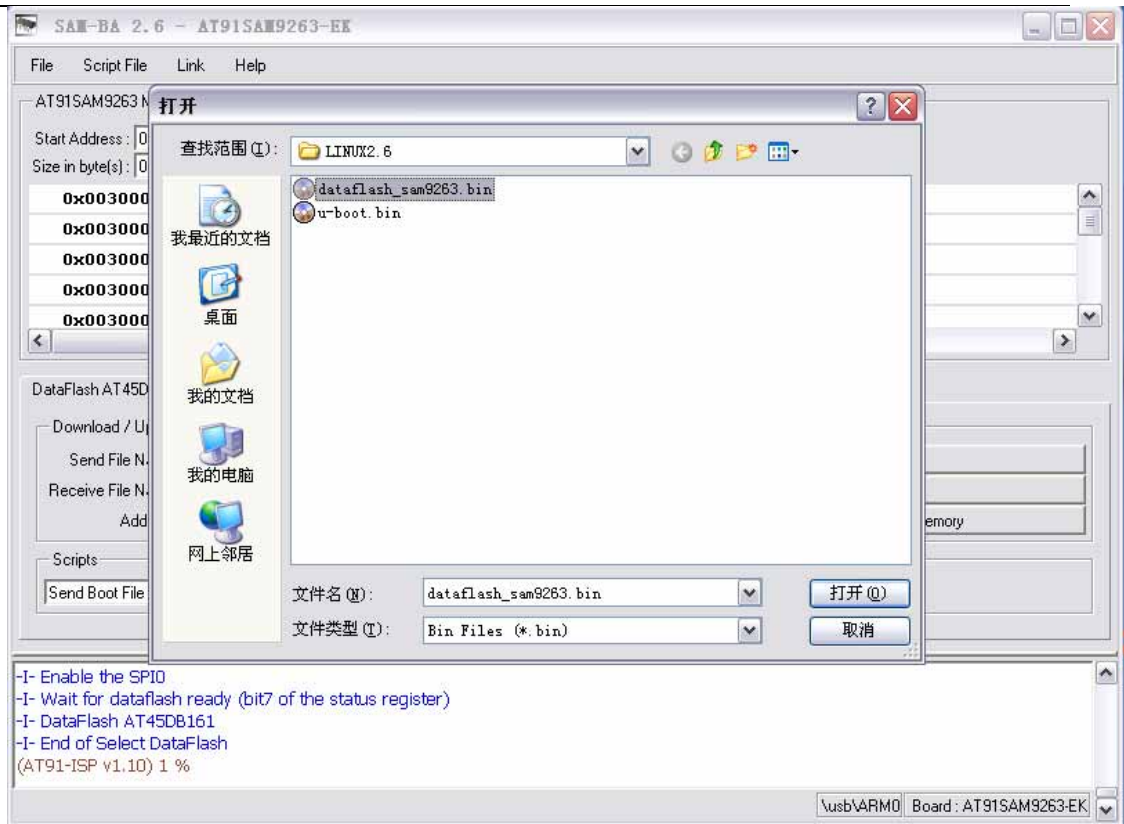
如果在连接中出现 “\usb\ARM0 ”，则说明 USB 连接已经正常。 主板选择 AT91SAM9263-EK 后，点 CONNECT 按钮进入主界面。



(7) 重新短接核心板上的跳线帽 J0,

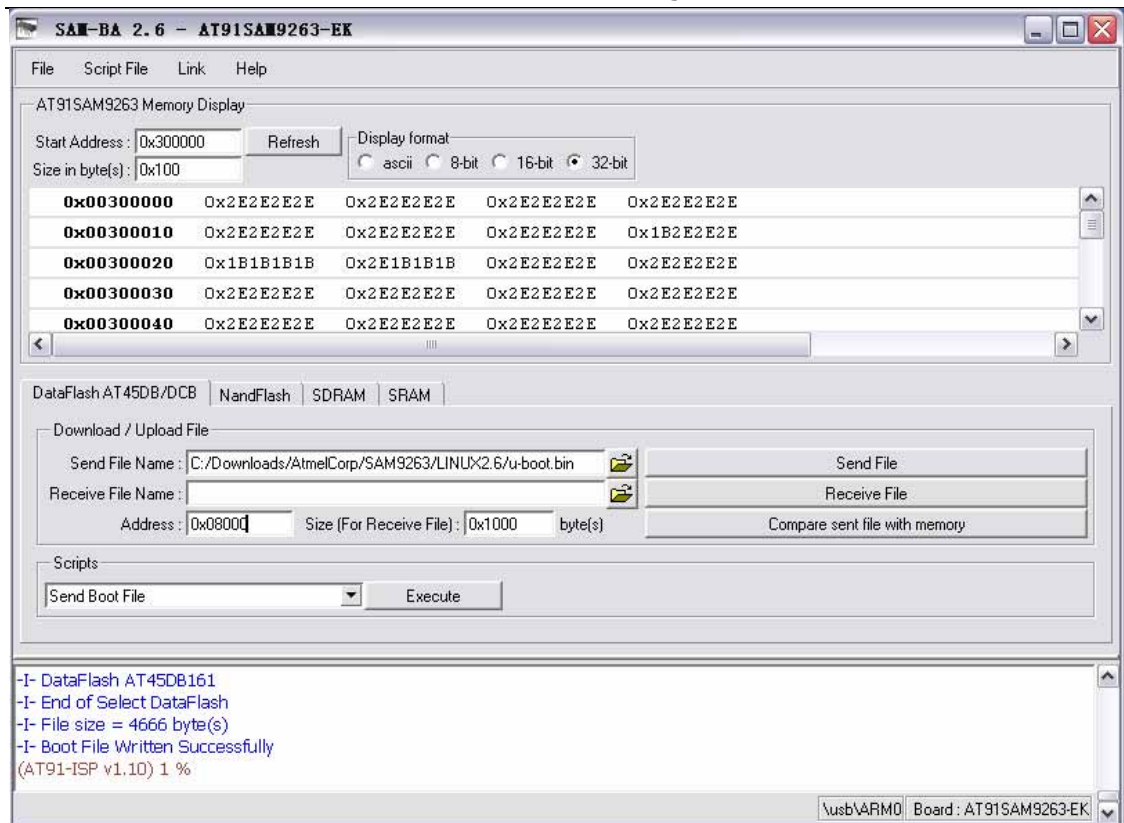
在 Scripts 下拉菜单中选择 Enale Dataflash on CS0, 点击 Execute 按钮 ,

在 Scripts 下拉菜单中选择 Send Boot File, 点击 Execute 按钮 ,出现打开对话框 , 如下图 :



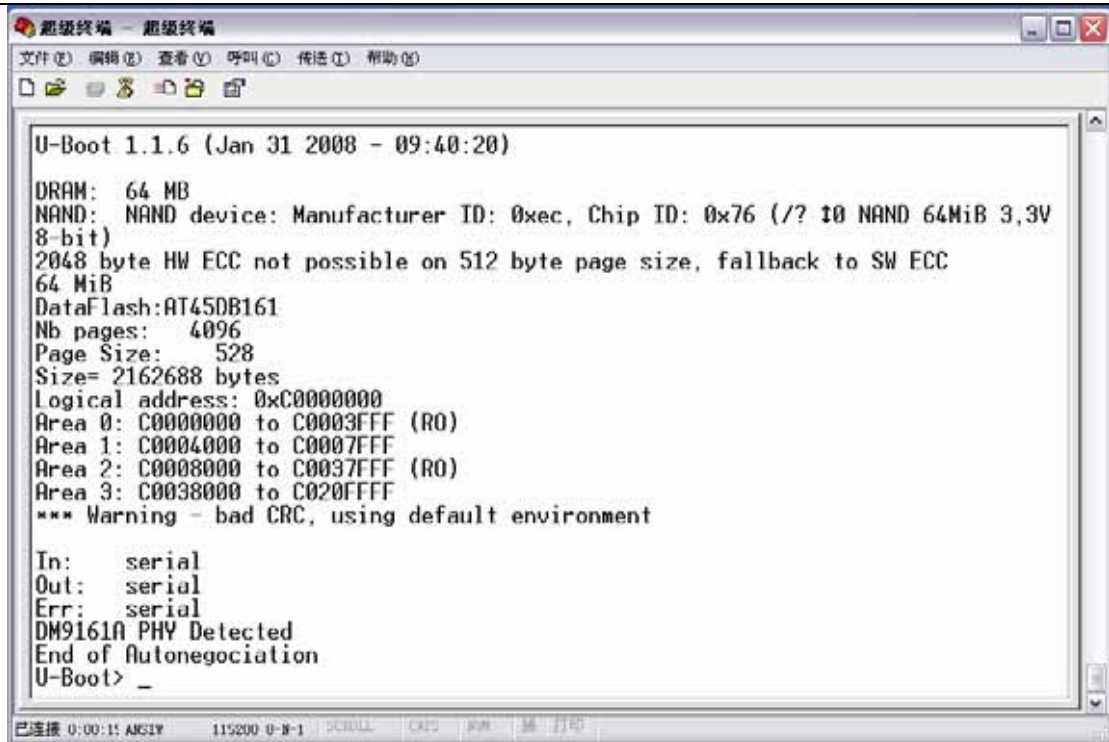
选择 dataflash_sam9263.bin 后打开，SAM-BA 会自动将文件写入 Dataflash 中。

(8) 修改 Address 框内容为 0x8000，在 Send File Name 框中，选择 u-boot.bin，如下图：



点击 Send File 按钮将 u-boot 写入 Dataflash。这个过程可能会持续一段时间。请等候。

(9) 发送完毕后，关闭 SAM-BA，断开 USB 设备线，按主板上的复位按钮 K1。这时候观察超级终端可以看到启动信息。



```

U-Boot 1.1.6 (Jan 31 2008 - 09:40:20)

DRAM: 64 MB
NAND: NAND device: Manufacturer ID: 0xec, Chip ID: 0x76 (/? 10 NAND 64MiB 3,3V
8-bit)
2048 byte HW ECC not possible on 512 byte page size, fallback to SW ECC
64 MiB
DataFlash:AT45DB161
Nb pages: 4096
Page Size: 528
Size= 2162688 bytes
Logical address: 0xC0000000
Area 0: C0000000 to C0003FFF (R0)
Area 1: C0004000 to C0007FFF (R0)
Area 2: C0008000 to C000BFFF (R0)
Area 3: C000C000 to C000FFFF (R0)
*** Warning - bad CRC, using default environment

In: serial
Out: serial
Err: serial
DM9161A PHY Detected
End of Autonegotiation
U-Boot> _
  
```

这时候，说明u-boot已经烧写成功了！

3、配置和使用u-boot

烧好u-boot 之后，我们要对启动脚本进行设置。依次执行如下命令：

- 1, setenv serverip 192.168.0.28 （服务器地址）
- 2, setenv ipaddr 192.168.0.128
- 3, setenv ethaddr 12:34:56:78:90:22 （arm板的MAC地址）
- 4, setenv bootargs mem=64M console=ttyS0 115200 root=/dev/ram0
initrd=0x21100000,0x600000 ip=192.168.0.128:192.168.0.28:192.168.0.1:
255.255.255.0:::eth0:off(0x21100000是ramdisk在sdram中的地址 ,0x600000
是它的大小,ip后面设置分别为目标板IP:服务器IP:网关IP:地址掩码)
- 5, setenv nf_kernel nand read 22200000 0 200000 (22200000是内核在sdram
中的地址)
- 6, setenv nf_ramdisk nand read 21100000 300000 600000 (300000是ramdisk
在nand中的偏移地址)
- 7, setenv boot bootm 22200000
- 8, setenv bootcmd run nf_kernel\;run nf_ramdisk\;run boot
- 9, saveenv (保存参数)

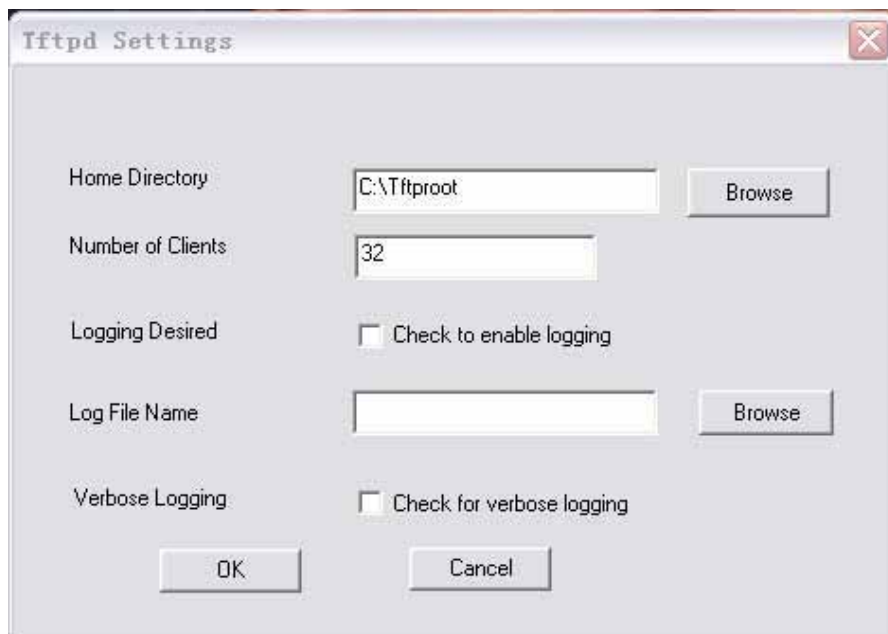
复位一下，使网络配置生效。这样，我们的启动脚本就设置好了。

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 uImage 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tftpd 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、下载内核：

```
U-Boot> tftp 22200000 ulmage
```

```
TFTP from server 192.168.0.28; our IP address is 192.168.0.188
```

```
Filename 'ulmage'.
```

```
Load address: 0x22200000
Loading: T #####
#####T
#####
done
Bytes transferred = 1446676 (161314 hex)
```

2、下载文件系统：

```
U-Boot> tftp 21100000 ramdisk.gz
TFTP from server 192.168.0.28; our IP address is 192.168.0.188
Filename 'ramdisk.gz'.
Load address: 0x21100000
Loading: T #####
#####T
#####
done
Bytes transferred = 4211555 (404363 hex)
```

3、写入NAND FLASH：

- 1, nand erase (擦除整块FLASH)
- 2, nand write 22200000 0 200000
- 3, nand write 21100000 300000 600000 (300000是在nandflash里面的偏移地址,600000是ramdisk.gz的大小,这两个数值可以根据实际情况修改,但是在前面参数设置里面设置的参数也要做相应的修改)

4、等待下载完毕上电重启

linux 起来之后就能看到shell 了：这样整个系统已经成功启动了。

```
RomBOOT
>
```

```
U-Boot 1.1.6 (Jan 31 2008 - 09:40:20)
```

```
DRAM: 64 MB
NAND: NAND device: Manufacturer ID: 0xec, Chip ID: 0x76 (/_0 NAND 64MiB 3,3V 8-bit)
2048 byte HW ECC not possible on 512 byte page size, fallback to SW ECC
64 MiB
DataFlash:AT45DB161
Nb pages: 4096
```



QY-9263K LINUX 用户手册

```
Page Size: 528
Size= 2162688 bytes
Logical address: 0xC0000000
Area 0: C0000000 to C0003FFF (RO)
Area 1: C0004000 to C0007FFF
Area 2: C0008000 to C0037FFF (RO)
Area 3: C0038000 to C020FFFF
In: serial
Out: serial
Err: serial
DM9161A PHY Detected

End of Autonegociation

Hit any key to stop autoboot: 0

NAND read: device 0 offset 0x0, size 0x200000
2097152 bytes read: OK

NAND read: device 0 offset 0x300000, size 0x500000
5242880 bytes read: OK
## Booting image at 22200000 ...
Image Name: Linux-2.6.20
Image Type: ARM Linux Kernel Image (uncompressed)
Data Size: 1446612 Bytes = 1.4 MB
Load Address: 20008000
Entry Point: 20008000
Verifying Checksum ... OK
OK

Starting kernel ...

Uncompressing Linux..... done, booting the kernel.
Linux version 2.6.20 (root@QY-SVR) (gcc version 4.0.0 (DENX ELDK 4.1 4.0.0)) #48 Fri
Feb 22 17:20:24 CST 2008
CPU0: ARM926EJ-S [41069265] revision 5 (ARMv5TEJ), cr=00053177
Machine: Atmel AT91SAM9263-EK
Ignoring unrecognised tag 0x54410008
Memory policy: ECC disabled, Data cache writeback
Clocks: CPU 198 MHz, master 99 MHz, main 18.432 MHz
CPU0: D VIVT write-back cache
CPU0: I cache: 16384 bytes, associativity 4, 32 byte lines, 128 sets
CPU0: D cache: 16384 bytes, associativity 4, 32 byte lines, 128 sets
Built 1 zonelists. Total pages: 16256
Kernel command line: mem=64M console=ttyS0 115200 root=/dev/ram0
initrd=0x21100000,0x800000
AT91: 160 gpio irqs in 5 banks
PID hash table entries: 256 (order: 8, 1024 bytes)
Console: colour dummy device 80x30
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: 53644KB available (2656K code, 254K data, 120K init)
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
NET: Registered protocol family 16
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 2
```



QY-9263K LINUX 用户手册

```
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 2048 (order: 1, 8192 bytes)
TCP bind hash table entries: 1024 (order: 0, 4096 bytes)
TCP: Hash tables configured (established 2048 bind 1024)
TCP reno registered
checking if image is initramfs...it isn't (no cpio magic); looks like an initrd
Freeing initrd memory: 8192K
NetWinder Floating Point Emulator V0.97 (double precision)
JFFS2 version 2.2. (NAND) (C) 2001-2006 Red Hat, Inc.
io scheduler noop registered
io scheduler anticipatory registered (default)
atmel_lcdfb_probe ...
LCD CLOCK: 0xc02c56c4Hz
atmel_lcdfb atmel_lcdfb.0: 600KiB frame buffer at 21100000 (mapped at ffc00000)
Console: switching to colour frame buffer device 80x30
atmel_lcdfb atmel_lcdfb.0: fb0: Atmel LCDC at 0x00700000 (mapped at c4850000), irq
26
atmel_usart.0: ttyS0 at MMIO 0xfeffee00 (irq = 1) is a ATMEL_SERIAL
atmel_usart.1: ttyS1 at MMIO 0xffff8c000 (irq = 7) is a ATMEL_SERIAL
atmel_usart.2: ttyS2 at MMIO 0xffff90000 (irq = 8) is a ATMEL_SERIAL
RAMDISK driver initialized: 16 RAM disks of 16384K size 1024 blocksize
loop: loaded (max 8 devices)
nbd: registered device at major 43
macb macb: detected PHY at address 0 (ID 0181:b8a0)
eth0: Atmel MACB at 0xffffbc000 irq 21 (12:34:45:67:89:90)
NAND device: Manufacturer ID: 0xec, Chip ID: 0x76 (Samsung NAND 64MiB 3,3V 8-bit)
Scanning device for bad blocks
Creating 2 MTD partitions on "NAND 64MiB 3,3V 8-bit":
0x00000000-0x04000000 : "Partition 1"
0x04000000-0x04000000 : "Partition 2"
mtd: partition "Partition 2" is out of reach -- disabled
atmel_spi atmel_spi.0: Atmel SPI Controller at 0xfffa4000 (irq 14)
mtd_dataflash spi0.0: AT45DB161x (2112 KBytes)
at91_ohci at91_ohci: AT91 OHCI
at91_ohci at91_ohci: new USB bus registered, assigned bus number 1
at91_ohci at91_ohci: irq 29, io mem 0x00a00000
usb usb1: configuration #1 chosen from 1 choice
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 2 ports detected
Initializing USB Mass Storage driver...
usbcore: registered new interface driver usb-storage
USB Mass Storage support registered.
udc: at91_udc version 3 May 2006
mice: PS/2 mouse device common for all mice
ads7846 spi0.3: touchscreen, irq 31
input: ADS784x Touchscreen as /class/input/input0
i2c /dev entries driver
Advanced Linux Sound Architecture Driver Version 1.0.14rc1 (Tue Jan 09 09:56:17 2007
UTC).
AC97C regs = FFFA0000
AC97C irq = 18
<6>ALSA device list:
#0: Atmel AC97 Controller at 0xfffa0000, irq 18
TCP cubic registered
NET: Registered protocol family 1
NET: Registered protocol family 17
RAMDISK: Compressed image found at block 0
VFS: Mounted root (ext2 filesystem).
Freeing init memory: 120K
INIT: version 2.86 booting
/ $
```

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在 arm-2007-01-21.iso里，用如下命令安装：

```
mount arm-2007-01-21.iso /mnt/ -o loop
cd /mnt
./install -d /opt/arm
```

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-2.6.20.tar.gz 压缩包，用如下命令打开：

```
# tar zxvf linux-2.6.20.tar.gz
```

进入Linux 目录：

```
# cd linux-2.6.20
```

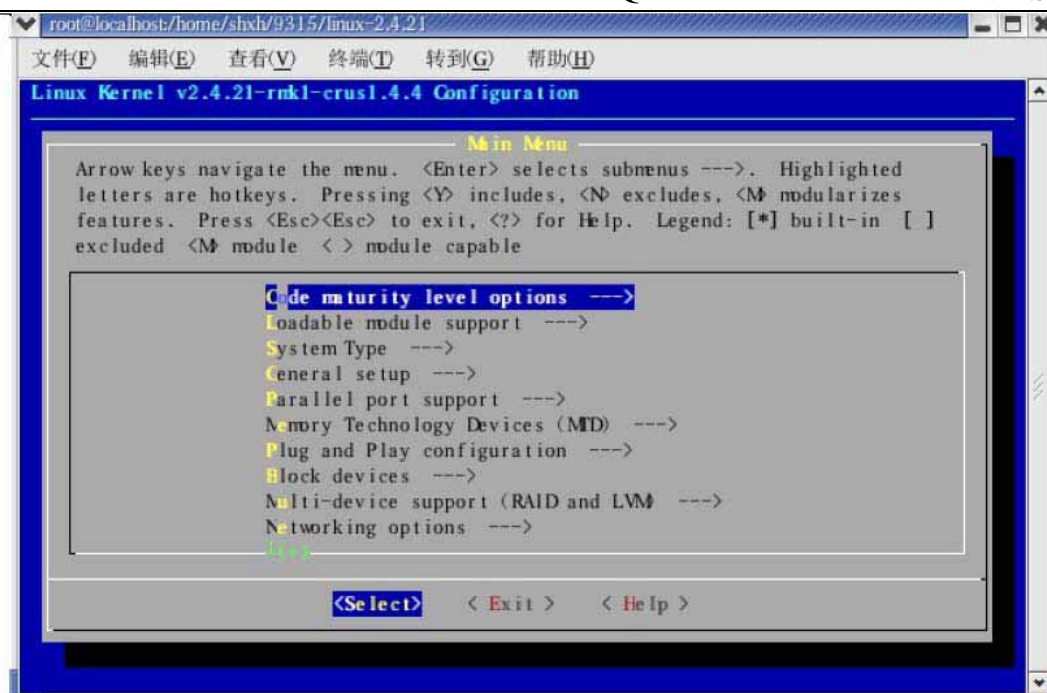
设置环境变量：

```
# 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。



保存好了之后就可以编译内核了，用命令make uImage：
#make uImage

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uImage 了。

六. 添加应用程序

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：

```
#arm-linux-gcc hello.c
```

其中arm-linux-gcc 是交叉编译工具的路径(请确保已经用命令source setnev设置了

环境变量), 这条命令会生成一个a.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)。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。当然, 在我们提供的压缩包的userland 目录下也有相关的几个例子程序。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放到主板上来运行了。这里列举了三种实现的方法。

1. 通过挂载NFS运行

通过目标板挂载服务器的NFS目录, 可以在目标板上直接操作NFS目录下的任何文件。

一. 服务器端设置

1. 在服务器上运行 setup, 进入 System services, 确认 nfs 选项被选中。
2. 创建/home/share 目录
3. 编辑/etc/exports 文件, 输入如下文字:

```
/home/share          192.168.0.*(rw, sync)
```

其中 192.168.0.*表示允许网段为 192.168.0 的任何 IP 登陆, 请确认网段为您目前所在网段。

4. 重新启动服务器

二. 目标板设置:

进入串口控制台, 输入命令:

```
mount -t nfs -o nolock 192.168.0.254:/home/share /mnt
```

注释:

192.168.0.254 为服务器 IP 地址, /home/share 为服务器的 NFS 目录

/mnt 为目标板的挂载目录

进入/mnt 目录, 即可看到服务器上/home/share 目录下的任何文件。

在该目录下用 ./a.out 等命令运行应用程序。

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

```
~#chmod 777 a.out
```

```
~#./a.out
```

```
Hello world !
```

这种方法十分方便我们调试程序。

3. 编入文件系统

您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
```

映像文件挂装

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

对/mnt/tmp目录进行操作

```
$ cd /mnt/tmp
```

```
$ 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/tmp
```

压缩文件系统，生成最终的文件系统映像

```
$ gzip -c -v9 ramdisk > /tftpboot/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标板上，您的应用程序就能在目标板上运行了。

杭州启扬智能科技有限公司

电话：0571-87858811 87858822

传真：0571-87858822

网址：www.qiyangtech.com

地址：杭州市西湖区山水人家诗家谷 5 幢 1 单元 1 楼

邮编：310013