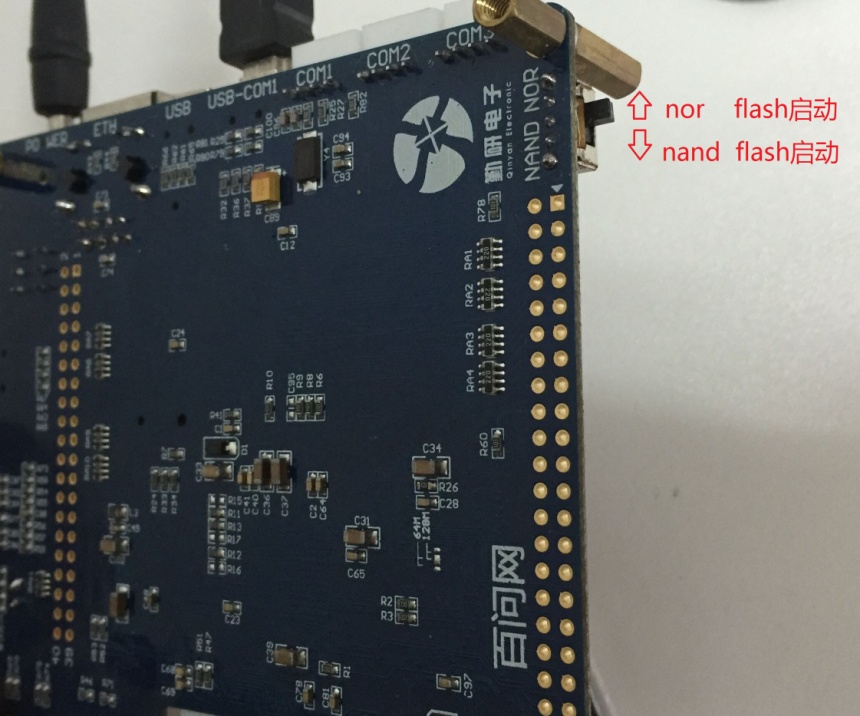
准备工作

**a.连接PC和开发板**

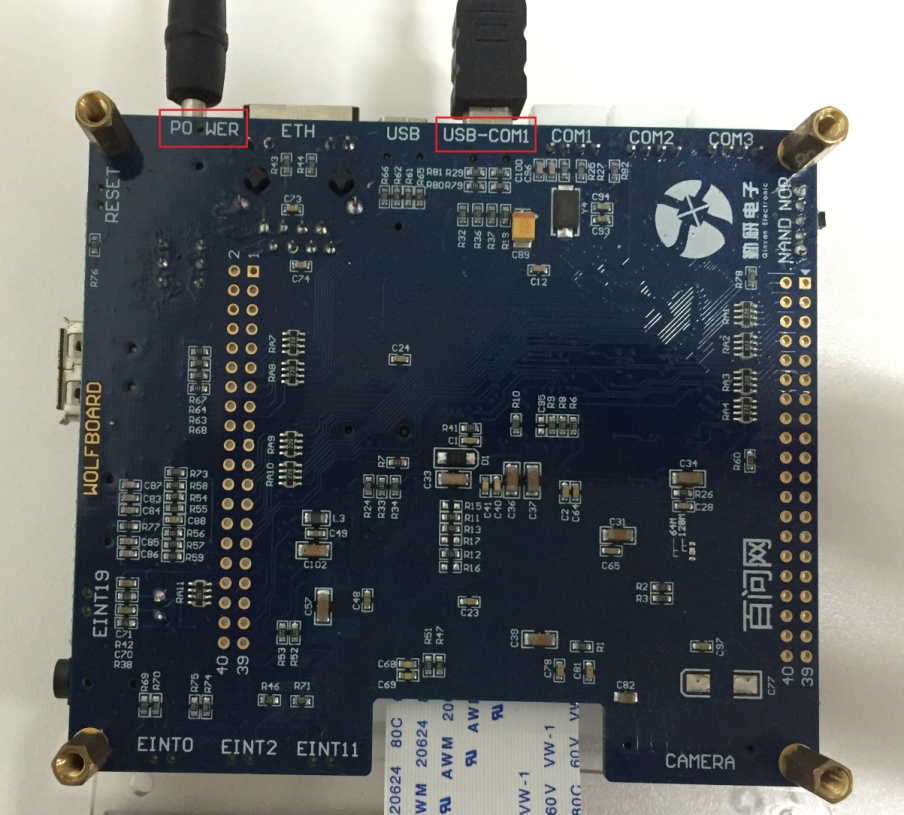
1.假定已经使用openjtag或者easyopenjtag将光盘\bin\u-boot.bin烧写到nor flash中。

2.选择“nand / nor启动开关”拨到nor flash启动，如图（为了方便说明，下图为去掉LCD后的图片，用户可以不用去掉LCD）：



3.用一根USB线连接“电脑的USB口”和 “JZ2440开发板的USB转串口接口”其中，当板子的LCD朝上时， “JZ2440开发板的USB转串口接口”（在PCB丝印中标有USB-COM1）。

此时开发板的连接图为（为了方便说明，下图为去掉LCD后的图片，用户可以不用去掉LCD）：



**1.1 运行nor flash里面的uboot**

经过前面的准备工作，按下开发板电源开关，开发板就开始运行nor flash里面的uboot程序，串口的打印信息如下：

U-Boot 1.1.6 (Nov 26 2010 - 08:49:16)

DRAM: 64 MB

Flash: 2 MB

NAND: 256 MiB

\*\*\* Warning - bad CRC or NAND, using default environment

In: serial

Out: serial

Err: serial

UPLLVal [M:38h,P:2h,S:2h]

MPLLVal [M:5ch,P:1h,S:1h]

CLKDIVN:5h

+---------------------------------------------+

| S3C2440A USB Downloader ver R0.03 2004 Jan |

+---------------------------------------------+

USB: IN\_ENDPOINT:1 OUT\_ENDPOINT:3

FORMAT: <ADDR(DATA):4>+<SIZE(n+10):4>+<DATA:n>+<CS:2>

NOTE: Power off/on or press the reset button for 1 sec

in order to get a valid USB device address.

输入空格键

Hit any key to stop autoboot: 3

此时“Hit any key to stop autoboot: 3”，表明uboot进入自动启动linux系统的三秒倒计时期间（此后统一称为三秒倒计时）。

按住键盘的“空格键”，将出现如下菜单界面：

##### 100ask Bootloader for OpenJTAG #####

[n] Download u-boot to Nand Flash

[o] Download u-boot to Nor Flash // 如果是Nand Flash启动的话，这个菜单项没有

[k] Download Linux kernel uImage

[j] Download root\_jffs2 image

[y] Download root\_yaffs image

[d] Download to SDRAM & Run

[z] Download zImage into RAM

[g] Boot linux from RAM

[f] Format the Nand Flash

[s] Set the boot parameters

[b] Boot the system

[r] Reboot u-boot

[q] Quit from menu

**1.1 uboot菜单**

各菜单项的作用从名字上即可看出来，对于“Download”开头的菜单项，它用来启动USB下载，选择这些菜单项后，要在主机上使用USB方式的DNW工具发送文件。

各菜单项说明如下：

|  |  |
| --- | --- |
| [n] Download u-boot to Nand Flash | 输入n，下载裸机代码或uboot到nand flash里面 |
| [o] Download u-boot to Nor Flash | 输入o，更新nor flash里面的u-boot，如果是Nand Flash启动的话，这个菜单项没有 |
| [k] Download Linux kernel uImage | 输入k，下载linux内核uImage |
| [j] Download root\_jffs2 image | 输入j，烧写JFFS2文件系统映象 |
| [y] Download root\_yaffs image | 输入y，烧写YAFFS文件系统映象 |
| [d] Download to SDRAM & Run | 输入d，下载文件到SDRAM并运行 |
| [z] Download zImage into RAM | 输入z，下载zImage到RAM中 |
| [g] Boot linux from RAM | 输入g，从RAM中启动 |
| [f] Format the Nand Flash | 输入f，格式化nand flash，格式化后，需要重新烧写整个系统 |
| [s] Set the boot parameters | 输入s，设置boot参数 |
| [b] Boot the system | 输入b，启动系统 |
| [r] Reboot u-boot | 输入r，重启uboot |
| [q] Quit from menu | 输入q，从uboot菜单切换到uboot命令操作模式 |

**1.2 uboot命令**

在uboot菜单界面输入字母q切换到uboot命令操作模式，即可输入各种命令，比如下载文件到内存，擦除、读写Flash，运行内存、NOR Flash、NAND Flash中的程序，查看、修改、比较内存中的数据等。

uboot命令可以理解为一段代码。每个uboot命令，对应不同的执行代码，通过正确输入命令字符串，就可以运行对应的命令。

由于uboot命令中，如果输入的字符串只与一个命令名“匹配”，则可以使用较短的输入去替代带该命令的完整输入，如输入tftp实际是运行tftpboot命令，输入print实际是运行printenv命令，输入bd实际是运行bdinfo命令。

输入一个命令后，必须按下回车键，命令才有效。若想再次运行上一次的命令，可以直接输入回车。

U-Boot接受的数据都是16进制，输入时可以省略前缀0x、0X。

下面按照一般uboot的移植顺序，介绍常用的uboot命令：

**（1）帮助命令help**

运行help命令可以看到U-Boot中所有命令的作用，如果要查看某个命令的使用方法，运行“help 命令名”，比如“help bootm”。

可以使用“?”来代替“help”，比如直接输入“?”、“? bootm”。

**（2）复位**

“reset”命令可以重启开发板。

**（3）内存操作命令**

在uboot中成功移植sdram后，就可以使用内存操作命令。

常用的内存操作命令有：查看内存命令md、修改内存命令md、填充内存命令mw、拷贝命令cp，比较cmp。这些命令都可以带上后缀“.b”、“.w”或“.l”，表示以字节、字(2个字节)、双字(4个字节)为单位进行操作。比如“cp.l 30000000 31000000 2”将从开始地址0x30000000处，拷贝2个双字到开始地址为0x31000000的地方。

md命令用法为“md[.b, .w, .l] address [count]”，表示以字节、字或双字(默认为双字)为单位，显示从地址address开始的内存数据，显示的数据个数为count。

mm命令用法为“mm[.b, .w, .l] address”，表示以字节、字或双字(默认为双字)为单位，从地址address开始修改内存数据。执行mm命令后，输入新数据后回车，地址会自动增加，Ctrl+C退出。

mw命令用法为“mw[.b, .w, .l] address value [count]”，表示以字节、字或双字(默认为双字)为单位，往开始地址为address的内存中填充count个数据，数据值为value。

cp命令用法为“cp[.b, .w, .l] source target count”，表示以字节、字或双字(默认为双字)为单位，从源地址source的内存拷贝count个数据到目的地址的内存。

cmp命令用法为“cmp [.b, .w, .l] addr1 addr2 count”，表示以字节、字或双字(默认为双字)为单位，比较地址addr1与addr2的count个数据。

**（4）NOR Flash操作命令**

常用的命令有查看Flash信息的flinfo命令、加/解写保护命令protect、擦除命令erase。由于NOR Flash的接口与一般内存相似，所以一些内存命令可以在NOR Flash上使用，比如读NOR Flash时可以使用md、cp命令，写NOR Flash时可以使用cp命令(cp根据地址分辨出是NOR Flash，从而调用NOR Flash驱动完成写操作)。

直接运行“flinfo”即可看到NOR Flash的信息，有NOR Flash的型号、容量、各扇区的开始地址、是否只读等信息。比如对于JZ2440V2开发板，flinfo命令的结果如下：

OpenJTAG> flinfo

Bank # 1: MXIC MX29LV160B FLASH (16 x 16) Size: 2 MB in 35 Sectors

AMD Standard command set, Manufacturer ID: 0xC2, Device ID: 0x2249

Erase timeout: 30000 ms, write timeout: 100 ms

Sector Start Addresses:

00000000 RO 00004000 RO 00006000 RO 00008000 RO 00010000 RO

00020000 RO 00030000 RO 00040000 00050000 00060000

00070000 00080000 00090000 000A0000 000B0000

000C0000 000D0000 000E0000 000F0000 00100000

00110000 00120000 00130000 00140000 00150000

00160000 00170000 00180000 00190000 001A0000

001B0000 001C0000 001D0000 001E0000 001F0000

其中的RO表示该扇区处于写保护状态，只读。

对于只读的扇区，在擦除、烧写它之前，要先解除写保护。最简单的命令为“protect off all”，解除所有NOR Flash的写保护。

erase命令常用的格式为“erase start end”──擦除的地址范围为start至end、“erase start +len”──擦除的地址范围为start至(start + len – 1)，“erase all”──表示擦除所有NOR Flash。

注意：其中的地址范围，刚好是一个扇区的开始地址到另一个(或同一个)扇区的结束地址。比如要擦除 MX29LV160B的前5个扇区，执行的命令为“erase 0 0x3fff”，而非“erase 0 0x4000”。

**（5）NAND Flash操作命令**

NAND Flash操作命令只有一个：nand，它根据不同的参数进行不同操作，比如擦除、读取、烧写等。

“nand info”查看NAND Flash信息。

“nand erase [clean] [off size]”擦除NAND Flash。加上“clean”时，表示在每个块的第一个扇区的OOB区加写入清除标记；off、size表示要擦除的开始偏移地址和长度，如果省略off和size，表示要擦除整个NAND Flash。

“nand read[.jffs2] addr off size”从NAND Flash偏移地址off处读出size个字节的数据，存放到开始地址为addr的内存中。是否加后缀“.jffs”的差别只是读操作时的ECC较验方法不同。

“nand write[.jffs2] addr off size”把开始地址为addr的内存中的size个字节数据，写到NAND Flash的偏移地址off处。是否加后缀“.jffs”的差别只是写操作时的ECC较验方法不同。

“nand read.yaffs addr off size”从NAND Flash偏移地址off处读出size个字节的数据(包括OOB区域），存放到开始地址为addr的内存中。

“nand write.yaffs addr off size”把开始地址为addr的内存中的size个字节数据(其中有要写入OOB区域的数据），写到NAND Flash的偏移地址off处。

“nand dump off”，将NAND Flash偏移地址off的一个扇区的数据打印出来，包括OOB数据。

**（6）环境变量命令**

“printenv”命令打印全部环境变量，通常输入字符串print来运行printenv命令。“printenv name1”打印名字为name1”的环境变量。

“setenv name value”设置名字为name的环境变量的值为value。

“setenv name”删除名字为name的环境变量。

“saveenv”每次在使用setenv命令设置环境变量后，都需要使用saveenv来保存环境变量到nand flash中。

上面的设置、删除操作只是在内存中进行，“saveenv”将更改后的所有环境变量写入NAND Flash中。

下面演示一下这几个命令。

OpenJTAG> setenv TEST 1 //新建一个TEST环境变量，并设置值为1

OpenJTAG> saveenv

Saving Environment to NAND...

Erasing Nand...Writing to Nand... done

OpenJTAG> print TEST

TEST=1

OpenJTAG> setenv TEST

OpenJTAG> saveenv

Saving Environment to NAND...

Erasing Nand...Writing to Nand... done

OpenJTAG> print TEST

## Error: "TEST" not defined

最后输出##Error: “TEST” not defined表明环境变量TEST删除成功。

**（7）串口下载命令**

U-Boot支持串口下载，相关命令有：loadb、loads、loadx、loady。

前几个串口下载命令使用方法相似，以loadx命令为例，它的用法为“loadx [ off ] [ baud ]”。中括号“[]”表示里面的参数可以省略，off表示文件下载后存放的内存地址，baud表示使用的波特率。如果baud参数省略，则使用当前的波特率；如果off参数省略，存放的地址为配置文件中定义的宏CFG\_LOAD\_ADDR。

**（8）网络命令和网络相关环境变量**

U-Boot支持网络下载，相关命令有：ping、tftpboot和nfs。

网络命令相关的环境变量有ipaddr、serverip、netmask和ethaddr。

ipaddr表示开发板网卡自己的ip地址。

serverip 表示服务器的ip地址。

netmask表示子网掩码。

ethaddr表示开发板网卡自己的设备地址。

输入print打印如下（此时只列出网络命令相关的环境变量）：

OpenJTAG> print

……

ipaddr=192.168.7.17

serverip=192.168.7.11

netmask=255.255.255.0

ethaddr=08:00:3e:26:0a:5b

注意，在网络下载前一般需要根据实际网络设置环境变量ipaddr、serverip、netmask。

ping 命令是向网络主机发送 ICMP的 ECHO\_REQUEST请求，一般测试网络连接状态，用法为“ping pingAddress”，pingAddress为网络中的ip地址。

注意：

由于uboot源码限制，只能使用开发板去ping 电脑的网卡，然后接受电脑的网卡发送过来的信息。如果使用电脑的网卡去ping 开发板，uboot是不会应答数据，从而可能造成“网络连接失败”的假象。

tftpboot命令使用TFTP协议从服务器下载文件，服务器的IP地址为环境变量serverip。通过help tftpboot可知，用法为“tftpboot [loadAddress] [bootfilename]”，loadAddress表示文件下载后存放的内存地址，bootfilename表示要下载的文件的名称。如果loadAddress省略，存放的地址为配置文件中定义的宏CFG\_LOAD\_ADDR；如果bootfilename省略，则使用单板的IP地址构造一个文件名，比如单板IP为192.168.1.17，则缺省的文件名为C0A80711.img。

nfs命令使用NFS协议下载文件，通过help nfs可知，用法为“nfs [loadAddress] [host ip addr:bootfilename]”。loadAddress、bootfilename的意义与tftpboot命令一样，host ip addr表示服务器的IP地址，默认为环境变量serverip。

下载文件成功后，U-Boot会自动创建或更新环境变量filesize，它表示下载的文件的长度，可以在后续命令中使用“$(filesize)”来引用它。

**（9）启动命令和启动相关环境变量**

启动命令：bdinfo,iminfo,bootm,go,nboot。

网络命令相关的环境变量有bootdelay 、bootcmd和bootargs。

bootdelay表示自启动系统的延时时间，一般设置为3秒。

bootcmd表示默认的自启动系统的命令。

bootargs表示启动系统时，传递给系统的启动参数。

OpenJTAG> print

……

bootdelay=3

bootcmd=nand read.jffs2 0x30007FC0 kernel; bootm 0x30007FC0

bootargs=noinitrd root=/dev/mtdblock3 init=/linuxrc console=ttySAC0

“bdinfo”命令可以检查开发板的机器ID和CPU的ID。

“iminfo”命令可以查看内存中内核信息。

不带参数的“boot”、“bootm”命令都是执行环境变量bootcmd所指定的命令。

“bootm [addr [arg ...]]”命令启动存放在地址addr处的U-Boot格式的映像文件(使用U-Boot目录tools下的mkimage工具制作得到)，[arg ...]表示参数。如果addr参数省略，映像文件所在地址为配置文件中定义的宏CFG\_LOAD\_ADDR。

“go addr [arg ...]”与bootm命令类似，启动存放在地址addr处的二进制文件， [arg ...]表示参数。

“nboot [[[loadAddr] dev] offset]”命令将NAND Flash设备dev上偏移地址off处的映像文件复制到内存loadAddr处，然后，如果环境变量autostart的值为“yes”，就启动这个映像。如果loadAddr参数省略，存放地址为配置文件中定义的宏CFG\_LOAD\_ADDR；如果dev参数省略，则它的取值为环境变量bootdevice的值；如果offset参数省略，则默认为0。