

嵌入式家园- JLink 烧写 Nand Flash 步骤指南

贺工 Hanson He, 阿里旺旺: 嵌入式家园
QQ:844698468, MSN:embedclub@hotmail.com

上海嵌入式家园-开发板商城

联系人: 贺工 (嵌入式高级系统工程师)

电话: 13917077026 传真: 021-50793997

QQ: 844698468

MSN: embedclub@hotmail.com

Email: embedclub@qq.com

网址: <http://embedclub.taobao.com/>

地址: 上海市浦东新区建中路 126 号

定位嵌入式服务提供商, 专业嵌入式 ARM 开发板商城, 承接各种嵌入式外包项目, 提供嵌入式培训, 提供方案设计, 项目预研指导, 提供技术支持服务, 涉及操作系统平台包括 Linux、Android、Windows CE 5.0/6.0、Windows Mobile 5.0/6.0 等; 硬件平台: S3C6410/S3C2440/S3C2410、TI OMAP 3530、PXA270、EP9315、i.MX21、i.MX31 等。

嵌入式家园-开发板商城主营嵌入式开发板及开发配件, 包括:

ARM9 系列: 友善之臂 mini2440/micro2440, 飞凌 OK2440/FL2440/TE2440, 天嵌 TQ2440, 广嵌 GEC2440 开发板

ARM11 系列: 友善之臂 mini6410/tiny6410, 华天正科技 Real6410, 飞凌 OK6410/FL6410, 友坚恒天 idea6410/UT-6410

S5PV210 系列: 友坚恒天 UT-S5PC100、UT-S5PC210 开发板

OMAP3530 系列: 天漠 Devkit8000、SBC8100 开发板

Zigbee 系列: 天运 CC2430 开发板

Cortex-M3 系列: 火牛 STM32 开发板

嵌入式开发仿真器: JLink、OpenJTAG 和 USB 转串口线

以及各种嵌入式开发配件 (GPS、GPRS、WiFi、CMOS 摄像头、USB 摄像头、3G 模块、VGA 转接板、AV 摄像头等), 欢迎广大客户来电咨询、洽谈!

我是使用笔记本的, 请问贺工还有什么配件需要购买吗? 如何实现在线仿真调试呢?

贺工解答:

请问你的笔记本或者台式机有串口和并口嘛?

如果没有, 你必须购买 USB 转串口线, 实现串口通讯。因为串口打印消息, 是嵌入式开发必备的, 否则寸步难行。

如果你的开发电脑没有并口的话, 在进行 ARM 编程调试的时候, 需要购买 JLINK, 实现在线仿真调试。

另外, 如果你想从事 Linux 学习开发的话, 推荐购买 OpenJTAG, 用于在线仿真调试 Linux 系统程序, 包括 uboot, 内核, 驱动和应用程序。

USB 转串口线 20 元

<http://item.taobao.com/item.htm?id=1344814485>

JLINK V7 117 元

<http://item.taobao.com/item.htm?id=2476313191>

JLINK V8 168 元

<http://item.taobao.com/item.htm?id=4034932401>

OpenJTAG 179 元

<http://item.taobao.com/item.htm?id=2050665211>

一、前期准备:

硬件连接:

- 1、JLink 的 JTAG 排线插上开发板的 JTAG 插座上, 注意排线的方向, 一定要吻合!
- 2、JLink 另一端, 可以通过 USB 延长线, 连接到 PC (笔记本) 的 USB 插口上。
- 3、开发板上电。
- 4、分别拷贝光盘中的 init.bin、u-boot.bin_openjtag、u-boot.bin_utu2440 至 PC 机的 E 盘下。

安装 JLink 驱动:

双击光盘中的 Setup_JLinkARM_V402.exe, 即可完成 JLink 驱动安装。

二、烧写 uboot 至扬创 utu2440/YC2440 的 Nand Flash:

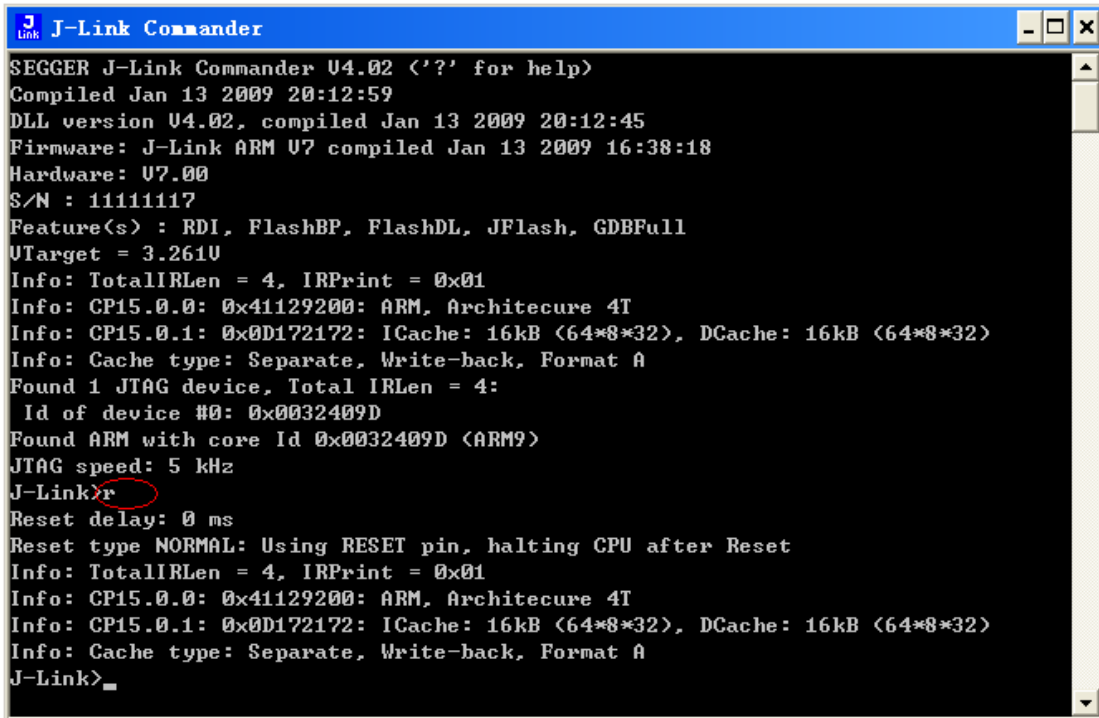
打开 J-Link Commander:

具体方法是: 开始->所有程序->SEGGER->J-Link ARM V4.02->J-Link Commander

在 J-Link Commander 命令行中, 依次执行下面步骤:

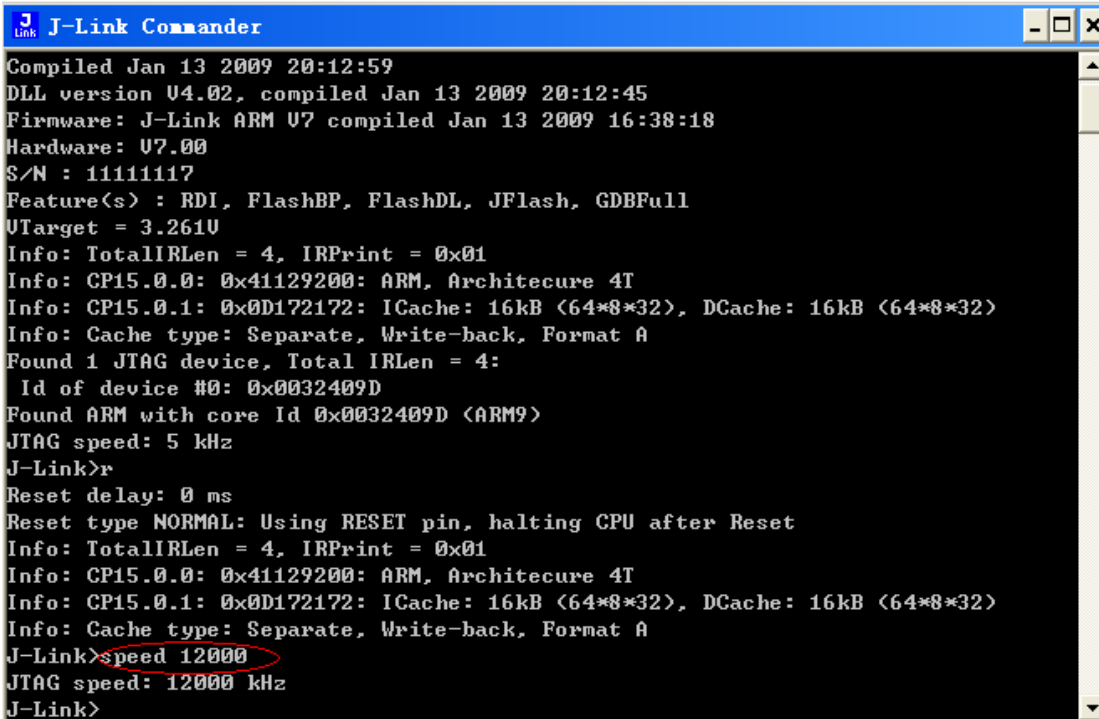
- 1、在 J-Link Commander 命令行中输入: **r**

//这个必不可少, 为了避免提示 CPU is not halted! 的错误, 正确找到 CPU, 可以多输入几次 r, 回车



```
SEGGER J-Link Commander V4.02 ('?' for help)
Compiled Jan 13 2009 20:12:59
DLL version V4.02, compiled Jan 13 2009 20:12:45
Firmware: J-Link ARM V7 compiled Jan 13 2009 16:38:18
Hardware: V7.00
S/N : 11111117
Feature(s) : RDI, FlashBP, FlashDL, JFlash, GDBFull
UTarget = 3.261V
Info: TotalIRLen = 4, IRPrint = 0x01
Info: CP15.0.0: 0x41129200: ARM, Architecture 4T
Info: CP15.0.1: 0x0D172172: ICache: 16kB (64*8*32), DCache: 16kB (64*8*32)
Info: Cache type: Separate, Write-back, Format A
Found 1 JTAG device, Total IRLen = 4:
  Id of device #0: 0x0032409D
Found ARM with core Id 0x0032409D (ARM9)
JTAG speed: 5 kHz
J-Link>r
Reset delay: 0 ms
Reset type NORMAL: Using RESET pin, halting CPU after Reset
Info: TotalIRLen = 4, IRPrint = 0x01
Info: CP15.0.0: 0x41129200: ARM, Architecture 4T
Info: CP15.0.1: 0x0D172172: ICache: 16kB (64*8*32), DCache: 16kB (64*8*32)
Info: Cache type: Separate, Write-back, Format A
J-Link>
```

2、speed 12000 //设置 TCK 为 12M, 提高下载速度



```
Compiled Jan 13 2009 20:12:59
DLL version V4.02, compiled Jan 13 2009 20:12:45
Firmware: J-Link ARM V7 compiled Jan 13 2009 16:38:18
Hardware: V7.00
S/N : 11111117
Feature(s) : RDI, FlashBP, FlashDL, JFlash, GDBFull
UTarget = 3.261V
Info: TotalIRLen = 4, IRPrint = 0x01
Info: CP15.0.0: 0x41129200: ARM, Architecture 4T
Info: CP15.0.1: 0x0D172172: ICache: 16kB (64*8*32), DCache: 16kB (64*8*32)
Info: Cache type: Separate, Write-back, Format A
Found 1 JTAG device, Total IRLen = 4:
  Id of device #0: 0x0032409D
Found ARM with core Id 0x0032409D (ARM9)
JTAG speed: 5 kHz
J-Link>r
Reset delay: 0 ms
Reset type NORMAL: Using RESET pin, halting CPU after Reset
Info: TotalIRLen = 4, IRPrint = 0x01
Info: CP15.0.0: 0x41129200: ARM, Architecture 4T
Info: CP15.0.1: 0x0D172172: ICache: 16kB (64*8*32), DCache: 16kB (64*8*32)
Info: Cache type: Separate, Write-back, Format A
J-Link>speed 12000
JTAG speed: 12000 kHz
J-Link>
```

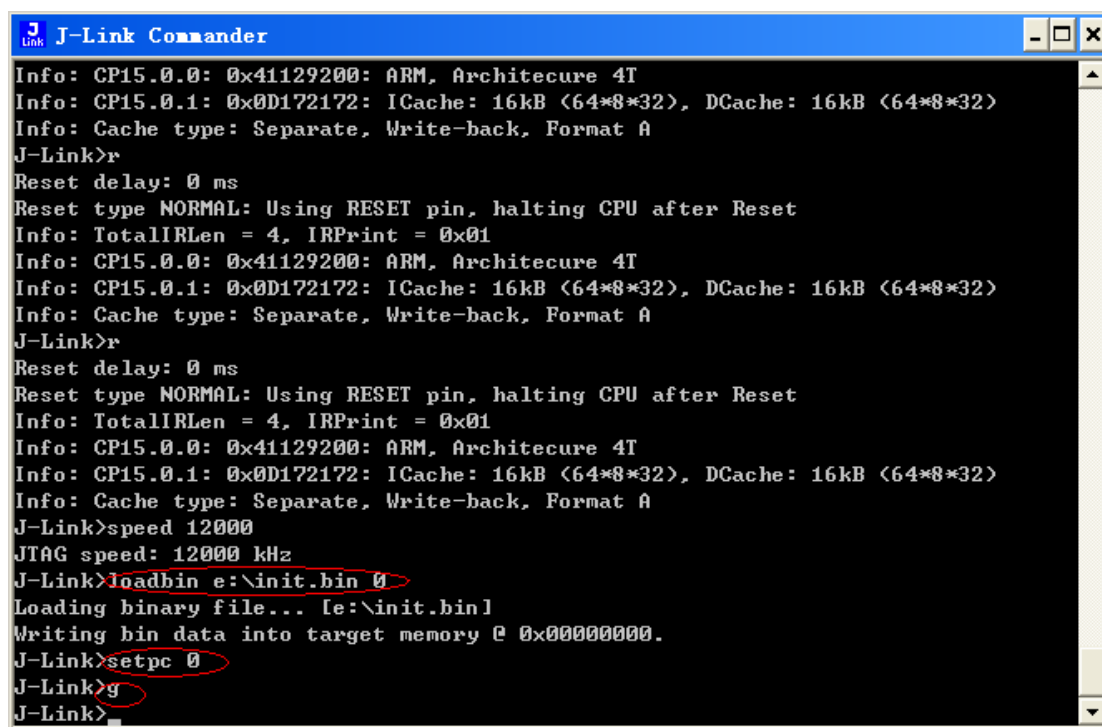
3、下载并运行 init.bin, 用来初始化 SDRAM

如果准备烧写 Nand Flash, 则将跳线打到 Nand Flash 这则, 从 Nand Flash 启动:

loadbin e:\init.bin 0 // 这个是 Jlink 特制的初始化 SDRAM 代码, 下载到 NandFlash 前 4KB 处。

setpc 0

g



```
J-Link Commander
Info: CP15.0.0: 0x41129200: ARM, Architecture 4T
Info: CP15.0.1: 0x0D172172: ICache: 16kB (64*8*32), DCache: 16kB (64*8*32)
Info: Cache type: Separate, Write-back, Format A
J-Link>r
Reset delay: 0 ms
Reset type NORMAL: Using RESET pin, halting CPU after Reset
Info: TotalIRLen = 4, IRPrint = 0x01
Info: CP15.0.0: 0x41129200: ARM, Architecture 4T
Info: CP15.0.1: 0x0D172172: ICache: 16kB (64*8*32), DCache: 16kB (64*8*32)
Info: Cache type: Separate, Write-back, Format A
J-Link>r
Reset delay: 0 ms
Reset type NORMAL: Using RESET pin, halting CPU after Reset
Info: TotalIRLen = 4, IRPrint = 0x01
Info: CP15.0.0: 0x41129200: ARM, Architecture 4T
Info: CP15.0.1: 0x0D172172: ICache: 16kB (64*8*32), DCache: 16kB (64*8*32)
Info: Cache type: Separate, Write-back, Format A
J-Link>speed 12000
JTAG speed: 12000 kHz
J-Link>loadbin e:\init.bin 0
Loading binary file... [e:\init.bin]
Writing bin data into target memory @ 0x00000000.
J-Link>setpc 0
J-Link>g
J-Link>
```

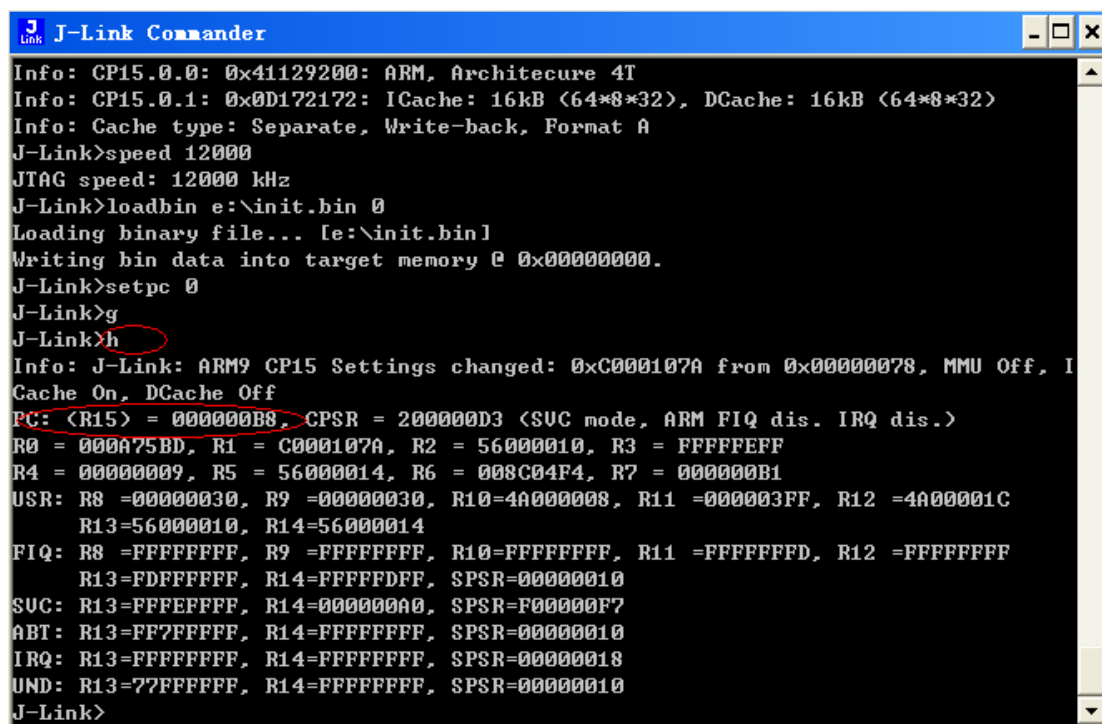
4、下载 u-boot.bin_openjtag 至 SDRAM 并运行

h

loadbin e:\u-boot.bin_openjtag 0x33f80000 //u-boot.bin_openjtag 为 Jlink 特制的 u-boot.bin, 大家可以将 JLINK 光盘中自带的 \example\u-boot\u-boot.bin 直接重命名为 u-boot.bin_openjtag。

setpc 0x33f80000

g



```
J-Link Commander
Info: CP15.0.0: 0x41129200: ARM, Architecture 4T
Info: CP15.0.1: 0x0D172172: ICache: 16kB (64*8*32), DCache: 16kB (64*8*32)
Info: Cache type: Separate, Write-back, Format A
J-Link>speed 12000
JTAG speed: 12000 kHz
J-Link>loadbin e:\init.bin 0
Loading binary file... [e:\init.bin]
Writing bin data into target memory @ 0x00000000.
J-Link>setpc 0
J-Link>g
J-Link>h
Info: J-Link: ARM9 CP15 Settings changed: 0xC000107A from 0x00000078, MMU Off, I
Cache On, DCache Off
PC: (R15) = 000000B8, CPSR = 200000D3 (SUC mode, ARM FIQ dis. IRQ dis.)
R0 = 000A75BD, R1 = C000107A, R2 = 56000010, R3 = FFFFFFFF
R4 = 00000009, R5 = 56000014, R6 = 008C04F4, R7 = 000000B1
USR: R8 = 00000030, R9 = 00000030, R10 = 4A000008, R11 = 000003FF, R12 = 4A00001C
R13 = 56000010, R14 = 56000014
FIQ: R8 = FFFFFFFF, R9 = FFFFFFFF, R10 = FFFFFFFF, R11 = FFFFFFFD, R12 = FFFFFFFF
R13 = FFFFFFFF, R14 = FFFFFFFF, SPSR = 00000010
SUC: R13 = FFFFFFFF, R14 = 000000A0, SPSR = F00000F7
ABT: R13 = FF7FFFFFFF, R14 = FFFFFFFF, SPSR = 00000010
IRQ: R13 = FFFFFFFF, R14 = FFFFFFFF, SPSR = 00000018
UND: R13 = 77FFFFFFF, R14 = FFFFFFFF, SPSR = 00000010
J-Link>
```

```
J-Link Commander
J-Link>loadbin e:\init.bin 0
Loading binary file... [e:\init.bin]
Writing bin data into target memory @ 0x00000000.
J-Link>setpc 0
J-Link>g
J-Link>h
Info: J-Link: ARM9 CP15 Settings changed: 0xC000107A from 0x00000078, MMU Off, I-Cache On, D-Cache Off
PC: <R15> = 000000B8, CPSR = 200000D3 (SVC mode, ARM FIQ dis. IRQ dis.)
R0 = 000A75BD, R1 = C000107A, R2 = 56000010, R3 = FFFFFFFF
R4 = 00000009, R5 = 56000014, R6 = 008C04F4, R7 = 000000B1
USR: R8 = 00000030, R9 = 00000030, R10 = 4A000008, R11 = 000003FF, R12 = 4A00001C
R13 = 56000010, R14 = 56000014
FIQ: R8 = FFFFFFFF, R9 = FFFFFFFF, R10 = FFFFFFFF, R11 = FFFFFFFD, R12 = FFFFFFFF
R13 = FDFFFFFFFF, R14 = FFFFFFFD, SPSR = 00000010
SVC: R13 = FFFEFFFF, R14 = 000000A0, SPSR = F00000F7
ABT: R13 = FF7FFFFFFF, R14 = FFFFFFFF, SPSR = 00000010
IRQ: R13 = FFFFFFFF, R14 = FFFFFFFF, SPSR = 00000018
UND: R13 = 77FFFFFF, R14 = FFFFFFFF, SPSR = 00000010
J-Link>loadbin e:\u-boot.bin_openjtag 0x33f80000
Loading binary file... [e:\u-boot.bin_openjtag]
Writing bin data into target memory @ 0x33f80000.
J-Link>setpc 0x33f80000
J-Link>g
J-Link>
```

这个时候，u-boot.bin_openjtag 已经在 SDRAM 中的 0x33f80000 开始运行了。

在超级终端中看到 u-boot.bin_openjtag 启动消息：OpenJTAG>

```
hanson - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
U-Boot 1.1.6 (Oct 20 2008 - 23:44:07)
DRAM: 64 MB
Flash: 0 kB
NAND: 64 MiB
*** Warning - bad CRC, using default environment
In: serial
Out: serial
Err: serial
Use these steps to program the image to flash:
1. In OpenOCD
Run the 'halt' command to halt u-boot
Run the 'load_image <file> <address>' command to load file to
Run the 'resume' command to resume u-boot
2. In u-boot, use the flash commands to program the image to flash
Or, use the tftp or nfs command to download file, and then program
OpenJTAG>
```

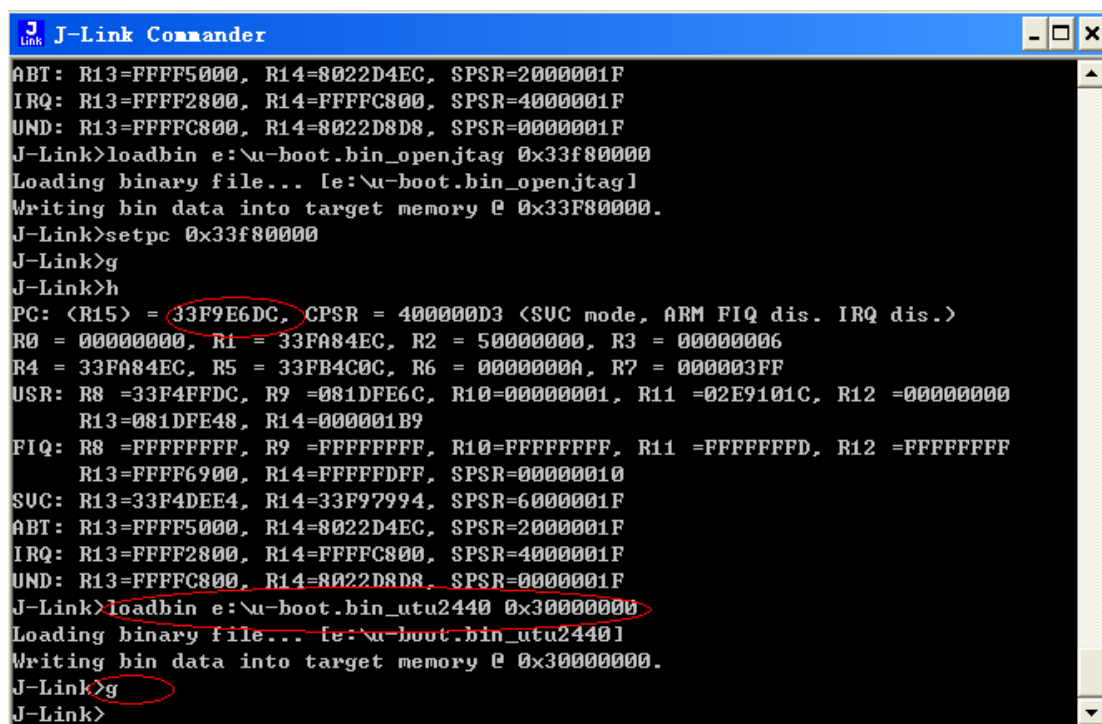
5、下载扬创开发板配套的 u-boot.bin_u2440 至 SDRAM

接着在 J-Link Commander 中继续输入：

h

loadbin e:\u-boot.bin_u2440 0x30000000 //u-boot.bin_u2440 为扬创开发板提供的 u-boot.bin。

g



```
J-Link Commander
ABT: R13=FFFF5000, R14=8022D4EC, SPSR=2000001F
IRQ: R13=FFFF2800, R14=FFFC800, SPSR=4000001F
UND: R13=FFFC800, R14=8022D8D8, SPSR=0000001F
J-Link>loadbin e:\u-boot.bin_openjtag 0x33f80000
Loading binary file... [e:\u-boot.bin_openjtag]
Writing bin data into target memory @ 0x33f80000.
J-Link>setpc 0x33f80000
J-Link>g
J-Link>h
PC: <R15> = 33F9E6DC, CPSR = 400000D3 (SVC mode, ARM FIQ dis. IRQ dis.)
R0 = 00000000, R1 = 33FA84EC, R2 = 50000000, R3 = 00000006
R4 = 33FA84EC, R5 = 33FB4C0C, R6 = 0000000A, R7 = 000003FF
USR: R8 = 33F4FFDC, R9 = 081DFE6C, R10=00000001, R11 =02E9101C, R12 =00000000
R13=081DFE48, R14=000001B9
FIQ: R8 =FFFFFFFF, R9 =FFFFFFFF, R10=FFFFFFFF, R11 =FFFFFFFD, R12 =FFFFFFFF
R13=FFFF6900, R14=FFFFFDFF, SPSR=00000010
SVC: R13=33F4DEE4, R14=33F97994, SPSR=6000001F
ABT: R13=FFFF5000, R14=8022D4EC, SPSR=2000001F
IRQ: R13=FFFF2800, R14=FFFC800, SPSR=4000001F
UND: R13=FFFC800, R14=8022D8D8, SPSR=0000001F
J-Link>loadbin e:\u-boot.bin_utu2440 0x30000000
Loading binary file... [e:\u-boot.bin_utu2440]
Writing bin data into target memory @ 0x30000000.
J-Link>g
J-Link>
```

这个时候扬创 u-boot.bin_utu2440 被下载到 SDRAM 的 0x30000000 去了。

6、通过 u-boot.bin_openjtag 将 u-boot.bin_utu2440 烧写到 Nand Flash:

在超级终端中输入:

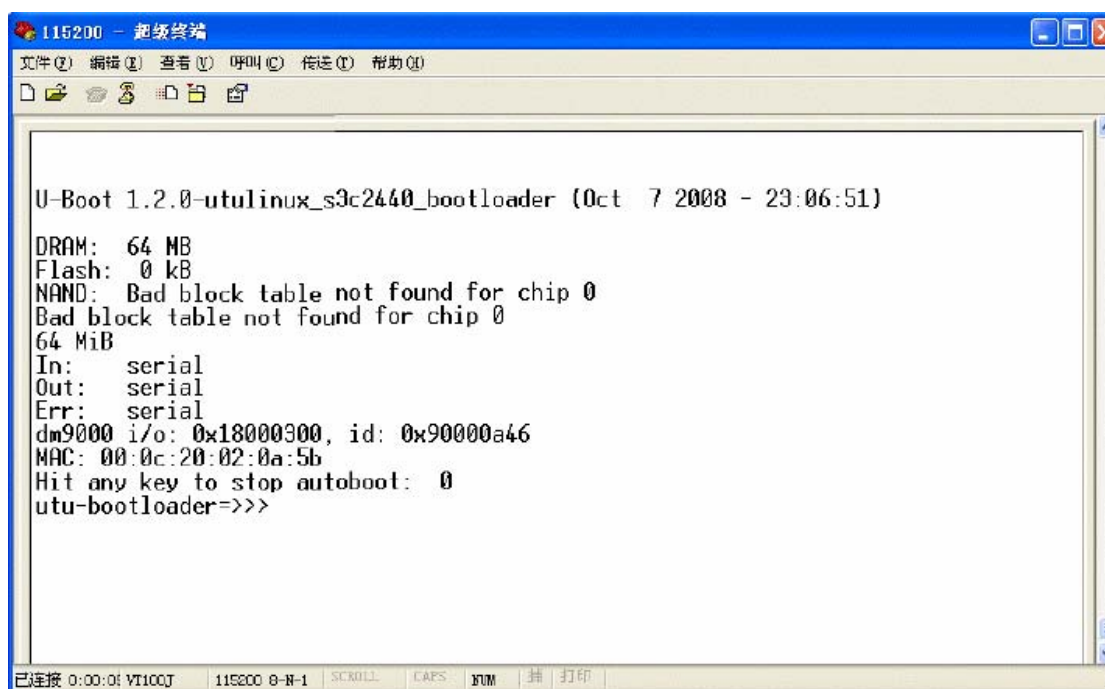
nand scrub // 擦除整个 Nand Flash

然后, 输入 y , 按下回车键: (注意: 这里输入 y 是没有显示的, 直接再回车就可以了);

nand erase 0 30000 //擦除从 0 地址开始的大小为 0x30000 (192KB) 的 Nand Flash 扇区

nand write.jffs2 30000000 0 30000 //把前面下载到的 0x30000000 的程序烧写到 Nand Block 0

这时先从开发板中 JTAG 接口处拔掉 JLINK 连接线, 然后, 复位后, 就可以在超级终端中看到扬创 utu2440 的 u-boot 启动, utu-bootloader=>>>



```
U-Boot 1.2.0-utulinux_s3c2440_bootloader (Oct 7 2008 - 23:06:51)

DRAM: 64 MB
Flash: 0 kB
NAND: Bad block table not found for chip 0
Bad block table not found for chip 0
64 MiB
In: serial
Out: serial
Err: serial
dm9000 i/o: 0x18000300, id: 0x90000a46
MAC: 00:0c:20:02:0a:5b
Hit any key to stop autoboot: 0
utu-bootloader=>>>
```

这时，就可以按照扬创开发板开发指南－《YC2440 开发指南-utuLinux2.6.24 篇-20090310.pdf》分别来烧写内核和文件系统了。run install-bootloader, run install-kernel 复位，按下空格，run install-filessystem

三、烧写 supervivi 至 mini2440/micro2440 至 Nor Flash:

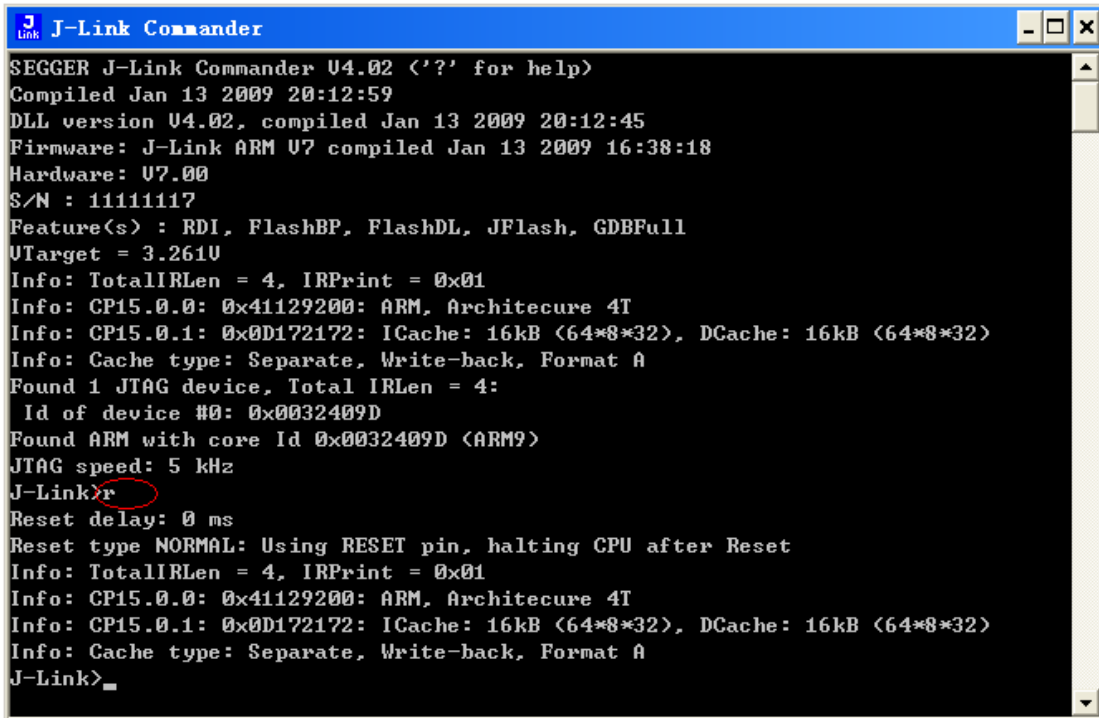
打开 J-Link Commander:

具体方法是：开始->所有程序->SEGGER->J-Link ARM V4.02->J-Link Commander

在 J-Link Commander 命令行中，依次执行下面步骤：

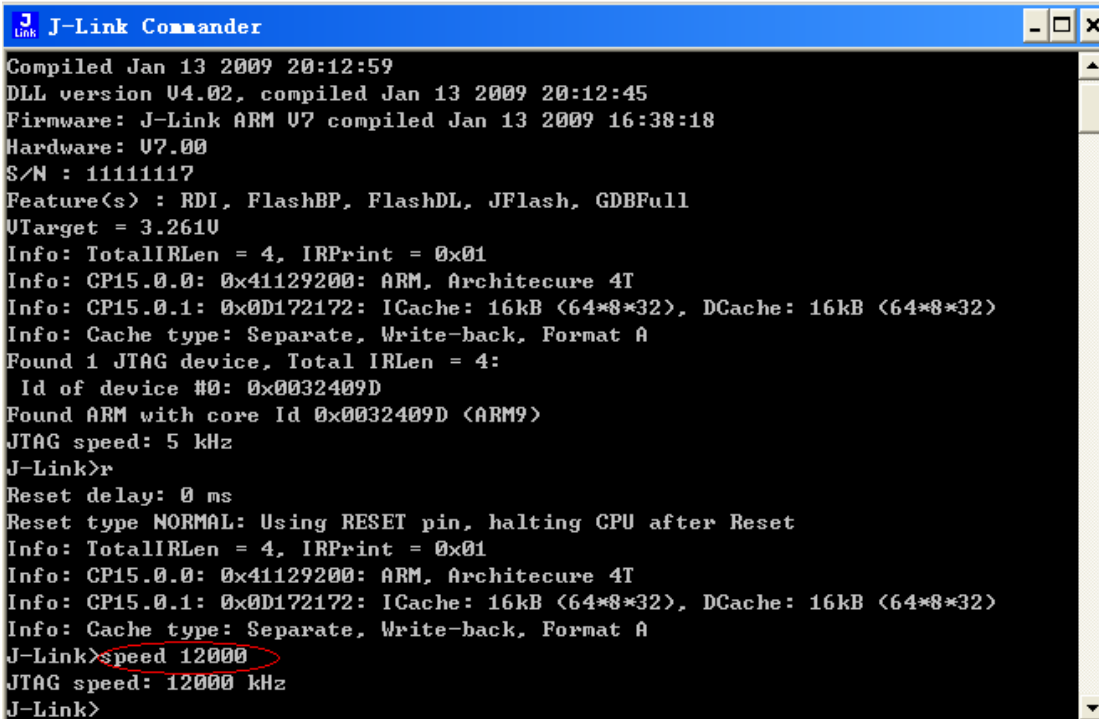
1、在 J-Link Commander 命令行中输入：**r**

//这个必不可少，为了避免提示 CPU is not halted!的错误，正确找到 CPU，可以多输入几次 r，回车



```
SEGGER J-Link Commander V4.02 ('?' for help)
Compiled Jan 13 2009 20:12:59
DLL version V4.02, compiled Jan 13 2009 20:12:45
Firmware: J-Link ARM V7 compiled Jan 13 2009 16:38:18
Hardware: V7.00
S/N : 11111117
Feature(s) : RDI, FlashBP, FlashDL, JFlash, GDBFull
UTarget = 3.261V
Info: TotalIRLen = 4, IRPrint = 0x01
Info: CP15.0.0: 0x41129200: ARM, Architecture 4T
Info: CP15.0.1: 0x0D172172: ICache: 16kB (64*8*32), DCache: 16kB (64*8*32)
Info: Cache type: Separate, Write-back, Format A
Found 1 JTAG device, Total IRLen = 4:
  Id of device #0: 0x0032409D
Found ARM with core Id 0x0032409D (ARM9)
JTAG speed: 5 kHz
J-Link>r
Reset delay: 0 ms
Reset type NORMAL: Using RESET pin, halting CPU after Reset
Info: TotalIRLen = 4, IRPrint = 0x01
Info: CP15.0.0: 0x41129200: ARM, Architecture 4T
Info: CP15.0.1: 0x0D172172: ICache: 16kB (64*8*32), DCache: 16kB (64*8*32)
Info: Cache type: Separate, Write-back, Format A
J-Link>
```

2、speed 12000 //设置 TCK 为 12M, 提高下载速度



```
Compiled Jan 13 2009 20:12:59
DLL version V4.02, compiled Jan 13 2009 20:12:45
Firmware: J-Link ARM V7 compiled Jan 13 2009 16:38:18
Hardware: V7.00
S/N : 11111117
Feature(s) : RDI, FlashBP, FlashDL, JFlash, GDBFull
UTarget = 3.261V
Info: TotalIRLen = 4, IRPrint = 0x01
Info: CP15.0.0: 0x41129200: ARM, Architecture 4T
Info: CP15.0.1: 0x0D172172: ICache: 16kB (64*8*32), DCache: 16kB (64*8*32)
Info: Cache type: Separate, Write-back, Format A
Found 1 JTAG device, Total IRLen = 4:
  Id of device #0: 0x0032409D
Found ARM with core Id 0x0032409D (ARM9)
JTAG speed: 5 kHz
J-Link>r
Reset delay: 0 ms
Reset type NORMAL: Using RESET pin, halting CPU after Reset
Info: TotalIRLen = 4, IRPrint = 0x01
Info: CP15.0.0: 0x41129200: ARM, Architecture 4T
Info: CP15.0.1: 0x0D172172: ICache: 16kB (64*8*32), DCache: 16kB (64*8*32)
Info: Cache type: Separate, Write-back, Format A
J-Link>speed 12000
JTAG speed: 12000 kHz
J-Link>
```

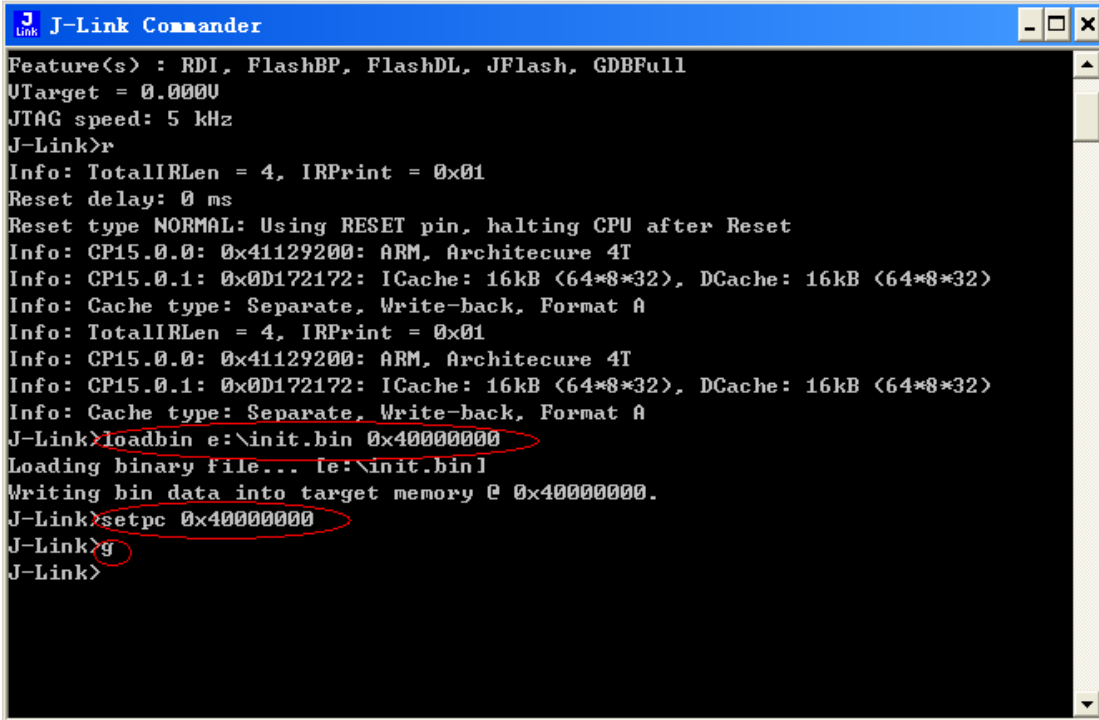
3、下载并运行 init.bin, 用来初始化 SDRAM

如果准备烧写 Nor Flash, 则将跳线拨至 Nor Flash 这则, 从 Nor Flash 启动:

loadbin e:\init.bin 0x40000000

setpc 0x40000000

g



```
J-Link Commander
Feature(s) : RDI, FlashBP, FlashDL, JFlash, GDBFull
VTarget = 0.000V
JTAG speed: 5 kHz
J-Link>r
Info: TotalIRLen = 4, IRPrint = 0x01
Reset delay: 0 ms
Reset type NORMAL: Using RESET pin, halting CPU after Reset
Info: CP15.0.0: 0x41129200: ARM, Architecture 4T
Info: CP15.0.1: 0x0D172172: ICache: 16kB (64*8*32), DCache: 16kB (64*8*32)
Info: Cache type: Separate, Write-back, Format A
Info: TotalIRLen = 4, IRPrint = 0x01
Info: CP15.0.0: 0x41129200: ARM, Architecture 4T
Info: CP15.0.1: 0x0D172172: ICache: 16kB (64*8*32), DCache: 16kB (64*8*32)
Info: Cache type: Separate, Write-back, Format A
J-Link>loadbin e:\init.bin 0x40000000
Loading binary file... [e:\init.bin]
Writing bin data into target memory @ 0x40000000.
J-Link>setpc 0x40000000
J-Link>g
J-Link>
```

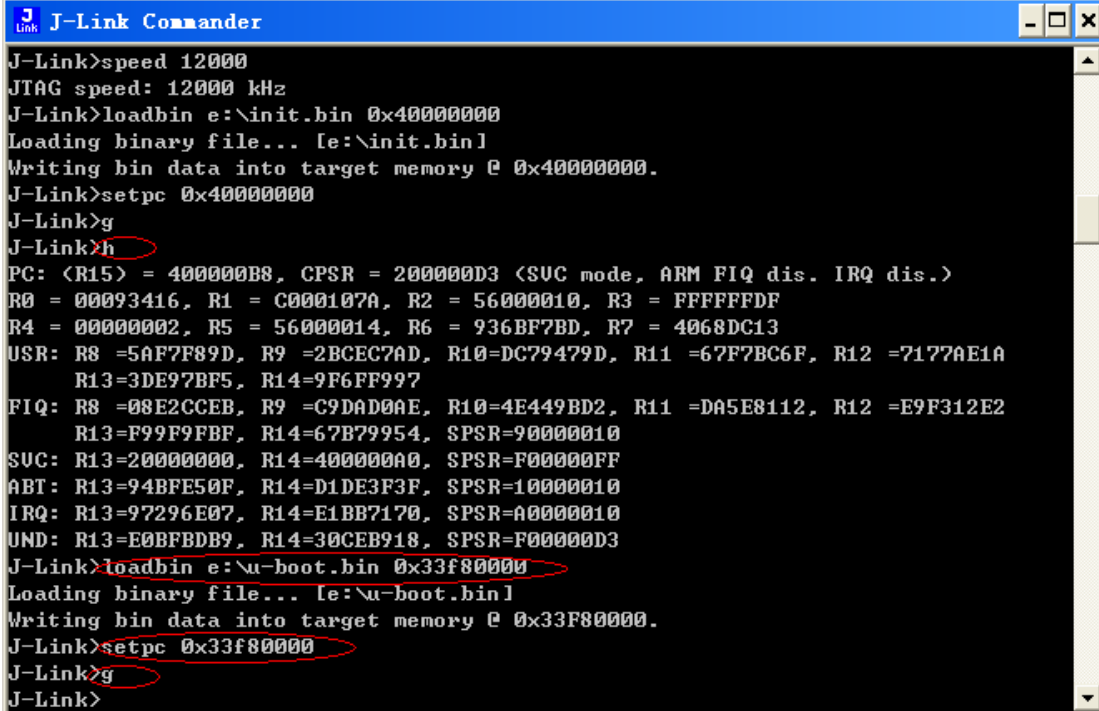
4、下载 u-boot.bin 至 SDRAM 并运行

h

loadbin e:\u-boot.bin 0x33f80000 //u-boot.bin 为 Jlink 特制的 u-boot.bin, 大家可以直接使用 JLINK 光盘中自带的\example\u-boot\u-boot.bin。

setpc 0x33f80000

g

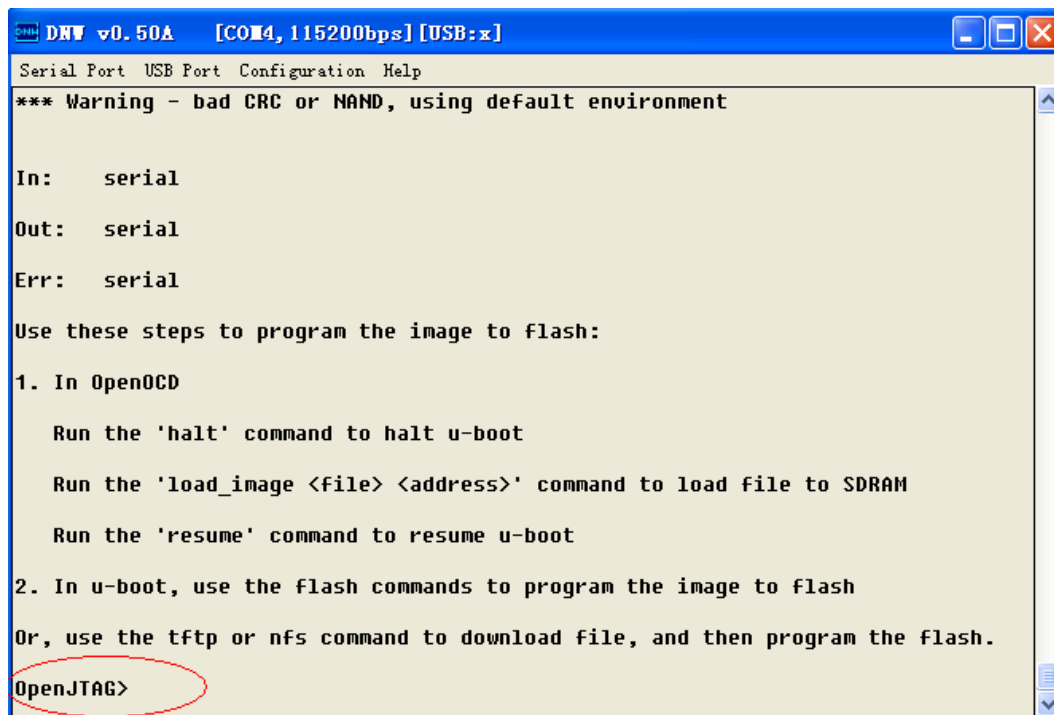


```
J-Link Commander
J-Link>speed 12000
JTAG speed: 12000 kHz
J-Link>loadbin e:\init.bin 0x40000000
Loading binary file... [e:\init.bin]
Writing bin data into target memory @ 0x40000000.
J-Link>setpc 0x40000000
J-Link>g
J-Link>h
PC: <R15> = 400000B8, CPSR = 200000D3 (SUC mode, ARM FIQ dis. IRQ dis.)
R0 = 00093416, R1 = C000107A, R2 = 56000010, R3 = FFFFFFFD
R4 = 00000002, R5 = 56000014, R6 = 936BF7BD, R7 = 4068DC13
USR: R8 = 5AF7F89D, R9 = 2BCEC7AD, R10=DC79479D, R11 = 67F7BC6F, R12 = 7177AE1A
R13=3DE97BF5, R14=9F6FF997
FIQ: R8 = 08E2CCEB, R9 = C9DAD0AE, R10=4E449BD2, R11 = DA5E8112, R12 = E9F312E2
R13=F99F9FBF, R14=67B79954, SPSR=90000010
SUC: R13=20000000, R14=400000A0, SPSR=F00000FF
ABT: R13=94BFE50F, R14=D1DE3F3F, SPSR=10000010
IRQ: R13=97296E07, R14=E1BB7170, SPSR=A0000010
UND: R13=E0BFBDB9, R14=30CEB918, SPSR=F00000D3
J-Link>loadbin e:\u-boot.bin 0x33f80000
Loading binary file... [e:\u-boot.bin]
Writing bin data into target memory @ 0x33f80000.
J-Link>setpc 0x33f80000
J-Link>g
J-Link>
```

随着蜂鸣器“嘟”的一声，这个时候，u-boot.bin 已经在 SDRAM 中的 0x33f80000 开始运行

了。

在超级终端或 DNW 或 SecureCRT 等串口调试工具中，将看到 u-boot.bin 启动消息：
OpenJTAG>



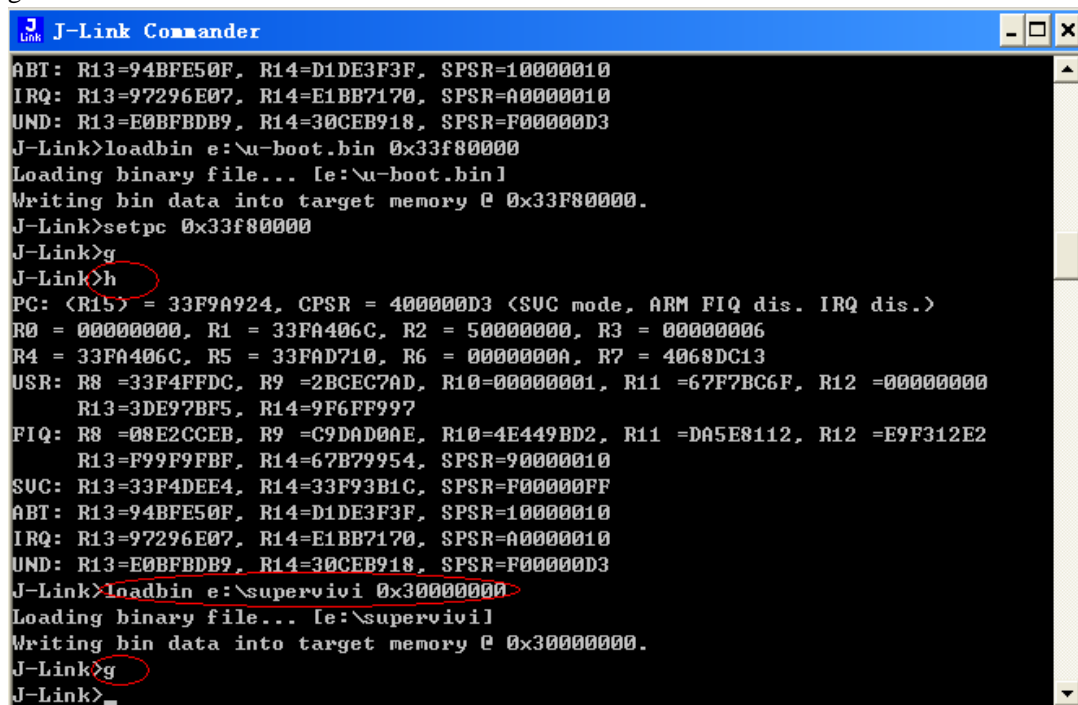
5、下载 mini2440/micro2440 开发板配套的 supervivi 至 SDRAM

接着在 J-Link Commander 中继续输入：

h

loadbin e:\supervivi 0x30000000 //其中 supervivi 为友善之臂 mini2440/micro2440 开发板光盘
中提供的 Bootloader 镜像文件。

g



这个时候 supervivi 被下载到 SDRAM 的 0x30000000 去了。

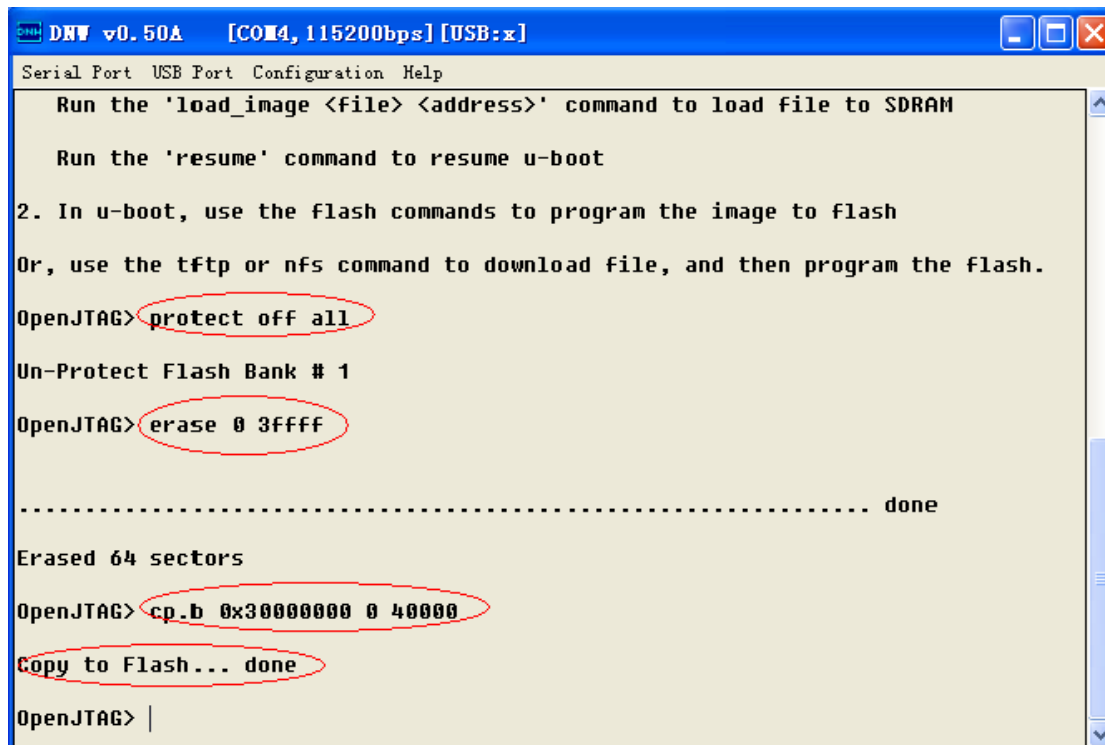
6、通过 u-boot.bin 将 supervivi 烧写到 Nor Flash:

在超级终端或 DNW 或 SecureCRT 等串口调试工具中输入:

```
protect off all //解锁
```

```
erase 0 3ffff // 擦除从 0 地址开始的大小为 0x40000(256KB)的 Nor Flash 扇区（大小  
可为擦除块的整数倍，可以运行 flinfo 命令查看）
```

```
cp.b 0x30000000 0 40000 //将前面下载到 0x30000000 的 supervivi 烧写到 Nor Flash 去。（注  
意到 supervivi 的大小为 251KB，因此，我们选择擦除的大小为 256KB。上面用的 3ffff, 40000  
都是 256KB，如果你的镜像比较小，请自行设置）。
```



这时，supervivi 已经烧写到开发板中 Nor Flash 0 地址开始的大小为 256KB 的块中。

接着，关闭开发板电源开关，分别从开发板上拔掉 JLINK 的 JTAG 端，从电脑上拔掉 JLINK 的 USB 端，保持开发板上的跳线仍在 Nor Flash 这则，复位开发板，这时将在 DNW 中看到 supervivi 的启动菜单界面，读者就可以按照《mini2440-um-20090719.pdf》用户手册的第三章，利用 supervivi 下载系统镜像的方法来安装 Linux 系统了。

四、烧写 stepldr.nb0(4KB), eboot.nb0(256KB)

1、烧写 stepldr.nb0 的方法与 u-boot 的方法一致;

```
speed 12000
```

```
loadbin e:\init.bin 0
```

```
setpc 0
```

```
g
```

```
h
```

```
loadbin e:\u-boot.bin_openjtag 0x33f80000
```

```
setpc 0x33f80000
```

g

这个时候, u-boot.bin_openjtag 已经在 SDRAM 中的 0x33f80000 开始运行了。

在超级终端中看到 OpenJTAG:>消息

接着在 J-Link Commander 中继续输入:

h

loadbin e:\stepldr.nb0 0x30000000

g

这个时候 stepldr.nb0 被下载到 SDRAM 的 0x30000000 去了。

将 stepldr.nb0 烧写到 Nand Flash:

在 u-boot 里执行, 超级终端中输入:

nand erase 0 1000 //擦除从 0 地址开始的大小为 0x1000 (4KB) 的 Nand Flash 扇区

nand write.jffs2 30000000 0 1000 //把前面下载到的 0x30000000 的程序烧写到 Nand Block 0

2、接着烧写 eboot.nb0 到 Block 2 开始的 256KB。

在 J-Link Commander 中继续输入:

h

loadbin e:\eboot.nb0 0x30000000

g

这个时候 boot.nb0 被下载到 SDRAM 的 0x30000000 去了。

将 stepldr.nb0 烧写到 Nand Flash:

在 u-boot 里执行, 超级终端中输入:

nand erase 8000 40000 //擦除从 0x8000 地址(即 Block 2,1 个 BLOCK 是 16K, BLOCK2 的地址就是 16K*2=32768=0x8000.)开始的大小为 0x40000 (256KB) 的 Nand Flash 扇区

nand write.jffs2 30000000 8000 40000 //把前面下载到的 0x30000000 的程序烧写到 Nand Block 2

五、烧写扬创开发板套餐 5 的 wince 的 BootLoader 映像文件: nboot1st.bin, NBOOT2nd.bin, YC2440_umon.bin

(1)、烧写 NBOOT1st.bin0 的方法与 u-boot 的方法一致;

speed 12000

loadbin e:\init.bin 0

setpc 0

g

h

loadbin e:\u-boot.bin_openjtag 0x33f80000

setpc 0x33f80000

g

这个时候，u-boot.bin_openjtag 已经在 SDRAM 中的 0x33f80000 开始运行了。
在超级终端中看到 OpenJTAG:>消息

接着在 J-Link Commander 中继续输入：

```
h
loadbin e:\NBOOT1st.bin 0x30000000
g
```

这个时候 NBOOT1st.bin 被下载到 SDRAM 的 0x30000000 去了。

将 NBOOT1st.bin 烧写到 Nand Flash：

在 u-boot 里执行，超级终端中输入：

```
nand erase 0 1000 //擦除从 0 地址开始的大小为 0x1000（4KB）的 Nand Flash 扇区
nand write.jffs2 30000000 0 1000 //把前面下载到的 0x30000000 的程序烧写到 Nand Block 0
```

（2）、接着烧写 nboot2nd.bin 到 Block 2 开始的 160KB。

在 J-Link Commander 中继续输入：

```
h
loadbin e:\nboot2nd.bin 0x30000000
g
```

这个时候 nboot2nd.bin 被下载到 SDRAM 的 0x30000000 去了。

将 nboot2nd.bin 烧写到 Nand Flash：

在 u-boot 里执行，超级终端中输入：

```
nand erase 8000 28000 //擦除从 0x8000 地址(即 Block 2,1 个 BLOCK 是 16K, BLOCK2 的地址就是 16K*2=32768=0x8000.)开始的大小为 0x28000（160KB）的 Nand Flash 扇区
nand write.jffs2 30000000 8000 28000 //把前面下载到的 0x30000000 的程序烧写到 Nand Block 2
```

（3）、接着烧写 YC2440_umon.bin（套餐 T43A(配套三星 480x272 分辨率 4.3 寸横屏)）到 Block 12 开始的 32KB。

在 J-Link Commander 中继续输入：

```
h
loadbin e:\YC2440_umon.bin 0x30000000
g
```

这个时候 YC2440_umon.bin 被下载到 SDRAM 的 0x30000000 去了。

将 YC2440_umon.bin 烧写到 Nand Flash：

在 u-boot 里执行，超级终端中输入：

```
nand erase 30000 8000 //擦除从 0x8000 地址(即 Block 2,1 个 BLOCK 是 16K, BLOCK 12 的地址就是 16K*12=0x30000.)开始的大小为 8000（32KB）的 Nand Flash 扇区
nand write.jffs2 30000000 30000 8000 //把前面下载到的 0x30000000 的程序烧写到 Nand Block
```

12

烧写完毕，关闭电源开关，拔下 JLINK，开发板复位，系统引导正常，进入 YC2440_umon 界面：

'U' for USBMON

接着按下键盘“u”键，进入下载界面：

Select Menu

[0] Download & Run

[1] Download Only

[2] Test SDRAM

[3] Change The Console UART Ch.

[4] Clear unused area in SDRAM

后面的操作，请参考扬创开发板用户手册，依次下载 nk.nb0, eboot.nb0

六、烧写扬创开发板套餐 T35B(奇美 3.5 寸触摸屏 320×240) 的 wince 的 BootLoader 映像文件：nboot1st.bin, NBOOT2nd.bin, YC2440_umon.bin

(1)、烧写 NBOOT1st.bin0 的方法与 u-boot 的方法一致；

speed 12000

loadbin e:\init.bin 0

setpc 0

g

h

loadbin e:\u-boot.bin_openjtag 0x33f80000

setpc 0x33f80000

g

这个时候，u-boot.bin_openjtag 已经在 SDRAM 中的 0x33f80000 开始运行了。

在超级终端中看到 OpenJTAG:>消息

接着在 J-Link Commander 中继续输入：

h

loadbin e:\NBOOT1st.bin 0x30000000

g

这个时候 NBOOT1st.bin 被下载到 SDRAM 的 0x30000000 去了。

将 NBOOT1st.bin 烧写到 Nand Flash：

在 u-boot 里执行，超级终端中输入：

nand erase 0 1000 //擦除从 0 地址开始的大小为 0x1000（4KB）的 Nand Flash 扇区

nand write.jffs2 30000000 0 1000 //把前面下载到的 0x30000000 的程序烧写到 Nand Block 0

(2)、接着烧写 nboot2nd.bin 到 Block 2 开始的 160KB。

在 J-Link Commander 中继续输入：

h

loadbin e:\nboot2nd.bin 0x30000000

g

这个时候 nboot2nd.bin 被下载到 SDRAM 的 0x30000000 去了。

将 nboot2nd.bin 烧写到 Nand Flash：

在 u-boot 里执行，超级终端中输入：

nand erase 8000 18000 //擦除从 0x8000 地址(即 Block 2,1 个 BLOCK 是 16K, BLOCK2 的地址就是 $16K * 2 = 32768 = 0x8000$.)开始的大小为 0x18000 (96KB) 的 Nand Flash 扇区

nand write.jffs2 30000000 8000 18000 //把前面下载到的 0x30000000 的程序烧写到 Nand Block 2

(3)、针对 utu2440-F 套餐 T35C(配套奇美 320x240 分辨率 3.5 寸横屏)烧写 YC2440_umon.bin 步骤：

烧写 YC2440_umon.bin 至 Nand Flash Block 8 开始的 32KB。

在 J-Link Commander 中继续输入：

h

loadbin e:\YC2440_umon.bin 0x30000000

g

这个时候 YC2440_umon.bin 被下载到 SDRAM 的 0x30000000 去了。

将 YC2440_umon.bin 烧写到 Nand Flash：

在 u-boot 里执行，超级终端中输入：

nand erase 20000 8000 //擦除从 0x8000 地址(即 Block 2,1 个 BLOCK 是 16K, BLOCK 8 的地址就是 $16K * 8 = 0x20000$.)开始的大小为 8000 (32KB) 的 Nand Flash 扇区

nand write.jffs2 30000000 20000 8000 //把前面下载到的 0x30000000 的程序烧写到 Nand Block 8

烧写完毕，关闭电源开关，拔下 JLINK，开发板复位，系统引导正常，进入 YC2440_umon 界面：

'U' for USBMON

接着按下键盘“u”键，进入下载界面：

Select Menu

[0] Download & Run

[1] Download Only

[2] Test SDRAM

[3] Change The Console UART Ch.

[4] Clear unused area in SDRAM

后面的操作，请参考扬创开发板用户手册，依次下载 nk.nb0, eboot.nb0

七、烧写飞凌开发板 OK2440/FL2440/TE2440 的 BootLoader 程序：

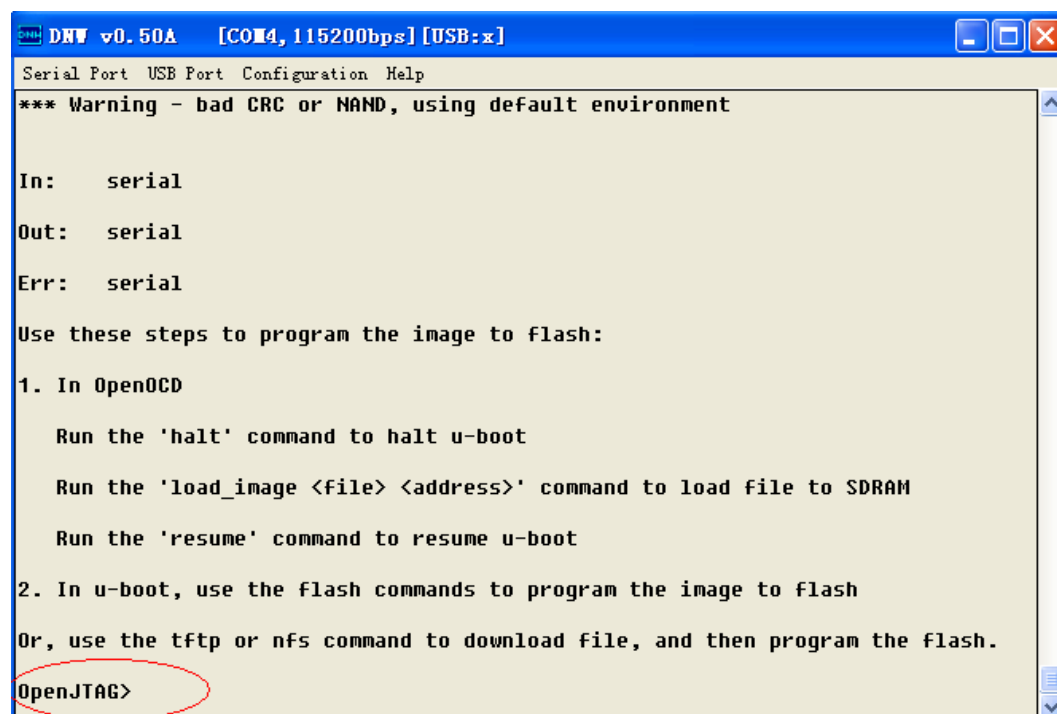
前面的步骤一样，依次是：

连接超级终端，打开 J-Link Commander 命令行，依次输入：

```
r
speed 12000
loadbin e:\init.bin 0
setpc 0
g
h
loadbin e:\u-boot.bin_openjtag 0x33f80000
setpc 0x33f80000
g
h
```

这是在超级终端显示 u-boot.bin_openjtag 运行的信息

OpenJTAG>



接下来，下载 OK2440/FL2440/TE2440 开发板配套的 BootLoader 程序： 2440bootV5.1.bin 至 SDRAM。

在 J-Link Commander 中继续输入：

h

loadbin e:\2440bootV5.1.bin 0x30000000 //其中 2440bootV5.1.bin 为 OK2440/FL2440/TE2440 开发板配套的 BootLoader 镜像文件。

g

这个时候 2440bootV5.1.bin 被下载到 SDRAM 的 0x30000000 去了。

继续, 通过 u-boot.bin_openjtag 将 2440bootV5.1.bin 烧写到 Nand Flash:

在超级终端中输入:

nand scrub // 擦除整个 Nand Flash

然后, 输入 y, 按下回车键: (注意: 这里输入 y 是没有显示的, 直接再回车就可以了);

nand erase 0 20000 //擦除从 0 地址开始的大小为 0x30000 (192KB) 的 Nand Flash 扇区

nand write.jffs2 30000000 0 20000 //把前面下载到的 0x30000000 的程序烧写到 Nand Block 0

这时先从开发板中 JTAG 接口处拔掉 JLINK 连接线, 然后, 复位后, 就可以在超级终端中看到 OK2440/FL2440/TE2440 开发板的 2440bootV5.1.bin 启动,

```
+-----+
|  TE/OK2440 Bootloader VER-5.1          www.witech.com.cn |
|-----|
|-----|
|CPU ID is 0x32440001                               |
|FCLK=400MHz,HCLK=100MHz,PCLK=50MHz, CPU is running at 400MHz|
|UPLL=96MHz, UCLK=48MHz                               |
|Serial port 0, Baud rate is 115200.                |
|OS image stored in NAND Flash.                      |
|Autoboot delay is 1 seconds.                        |
+-----+
```

按下空格键, 进入 2440bootV5.1.bin 命令行界面:

```
+-----+
|                               Select Menu                               |
+-----+
|[0] Download & Run                               |
|[1] Download to flash                           |
|[2] Download From UART                           |
|[3] Boot My Application                           |
|[4] Boot Linux                                   |
|[5] Boot Wince                                   |
|[6] Erase Flash Partition                         |
|[7] Config parameters                           |
```

后续, 就可以按照飞凌用户手册来下载内核 zImage 和根文件系统镜像文件。