

# u-boot 烧写说明

——By Team Mcuzone

首先，打开超级终端（开始->程序->附件->通讯->超级终端），新建一个连接：



图 1

选择串口，本站提供的 9200EK 使用的是 USB 转 UART 芯片 ARK3116，请先安装驱动，然后在设备管理器里面查看串口号，注意选择正确的串口号：



图 2

本站提供的 9200EK 上，DBGU 和 UART0 都是通过 USB 转串口芯片 ARK3116 来实现的，请确认当前跳线在 DBGU 位置上。

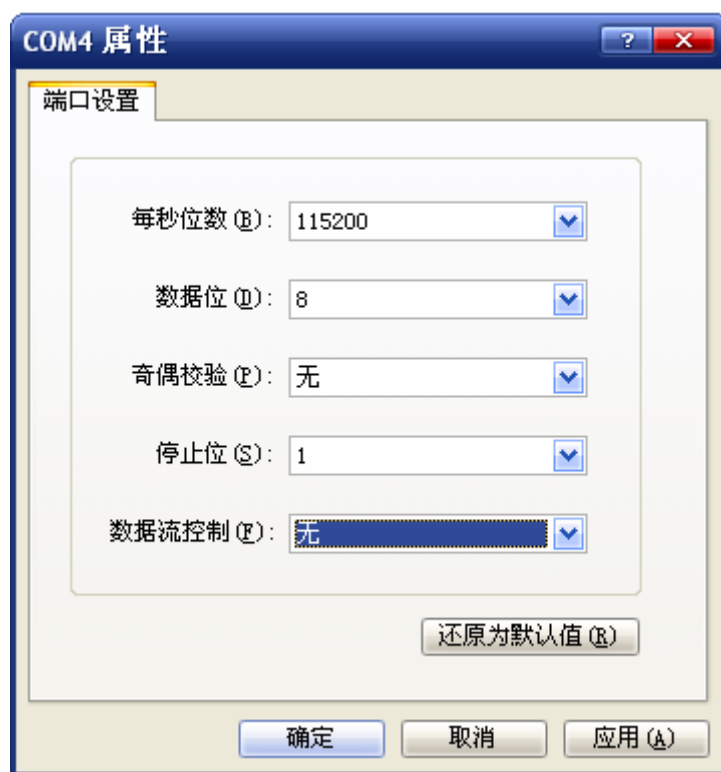


图 3

按照上图所示，选择波特率和其他设置。

把核心板上的 FLASHBOOT 跳线悬空，打开电源开关，将在超级终端窗口接收到字符 C:

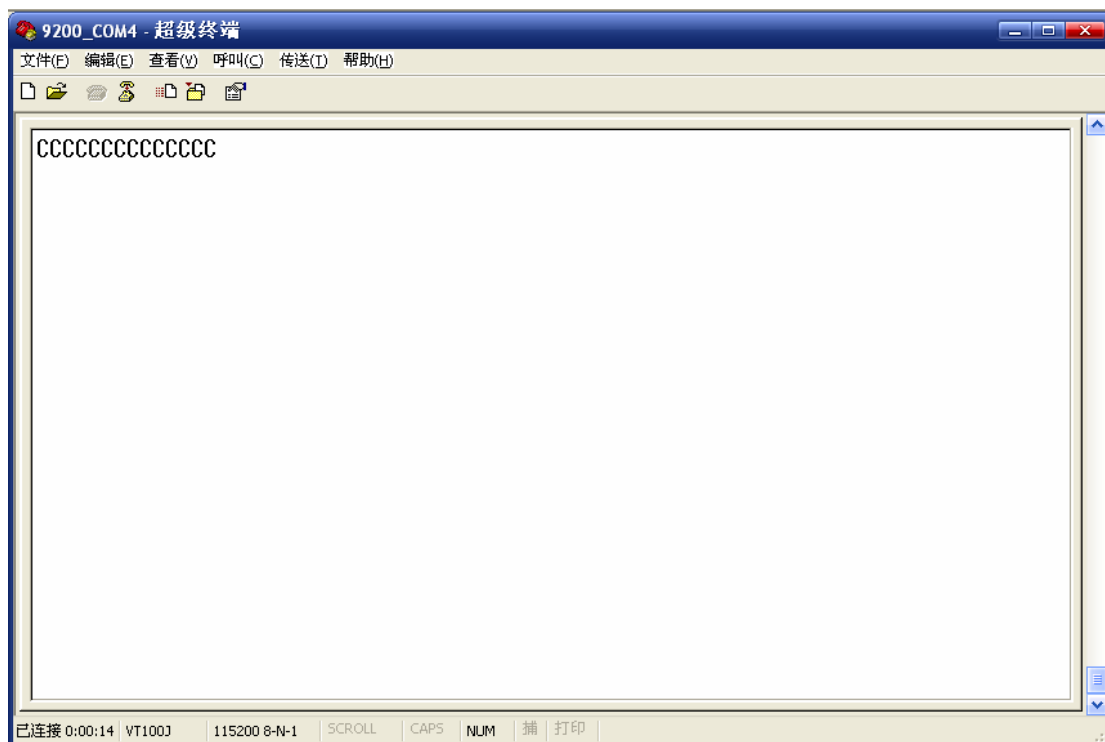


图 4

然后以 Xmodem 协议发送 9200loader.bin 文件，如下图：



图 5

很快下载完成，然后出现以下提示，9200loader.bin 完成对系统的初始化，同时输出字符 C：

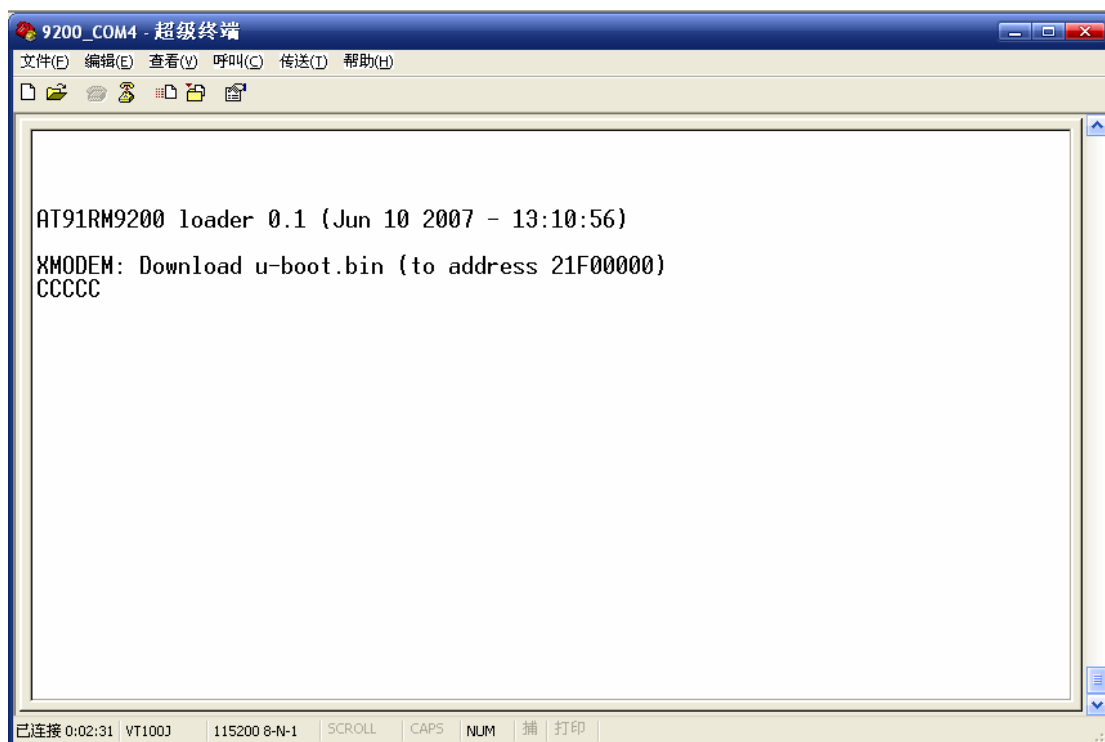


图 6

接下来继续以 Xmodem 协议发送 u-boot.bin，由于 u-boot.bin 相对较大，下载时间相对较长：



图 7

差不多大半分钟后可以下载完成。

下载完成后输出如下信息：

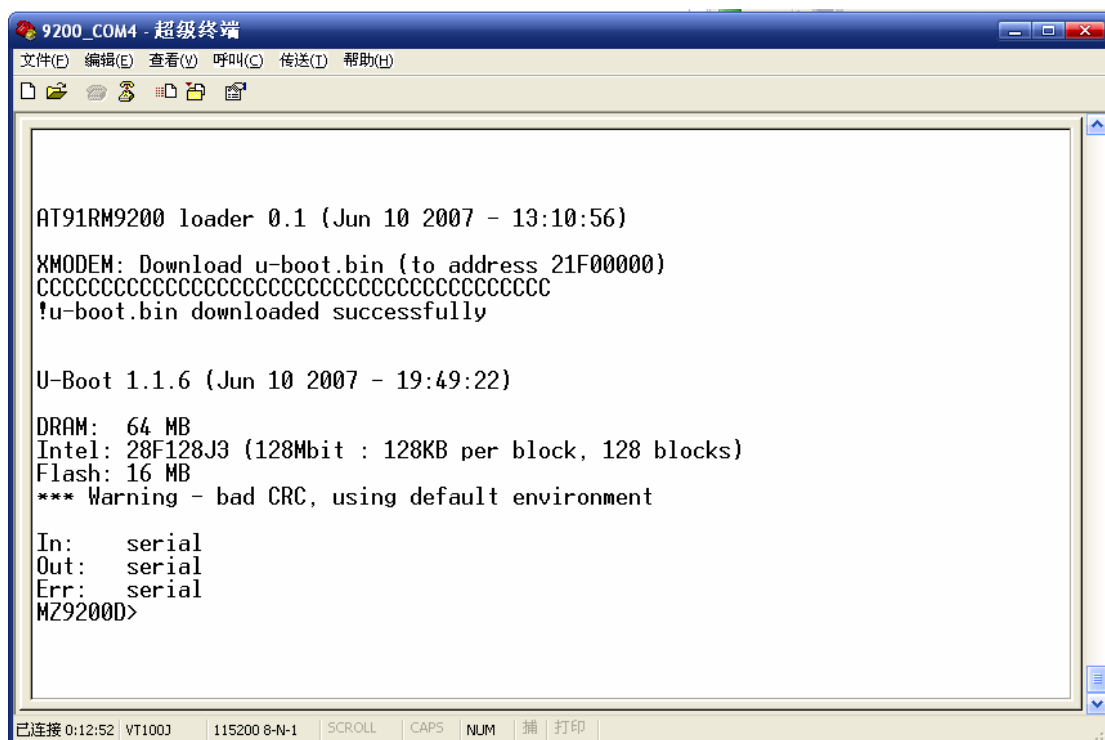


图 8

以上代码都运行在 SDRAM 中，为了能在开机后直接运行代码，我们需要把代码写入到 FLASH，可以用 cp 命令来实现。

先启动 loadb 命令：

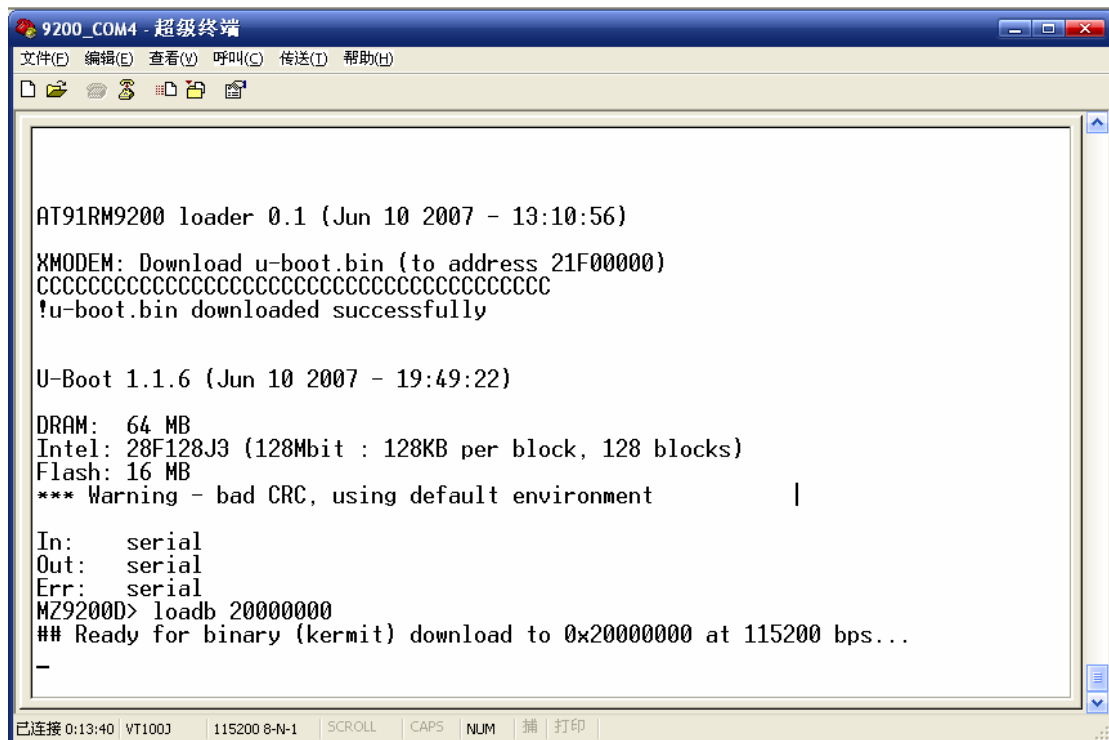


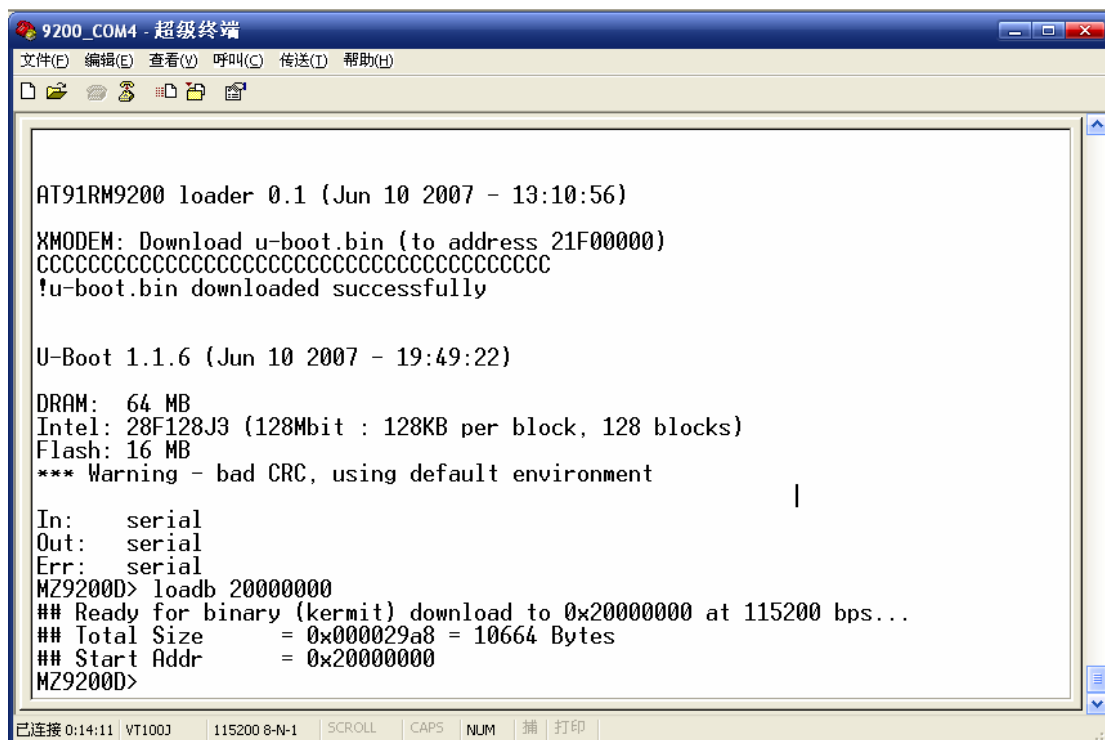
图 9

以 Kermit 协议发送 9200flashloader.bin



图 10

发送完成，输入以下信息：



```
AT91RM9200 loader 0.1 (Jun 10 2007 - 13:10:56)

XMODEM: Download u-boot.bin (to address 21F00000)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
!u-boot.bin downloaded successfully

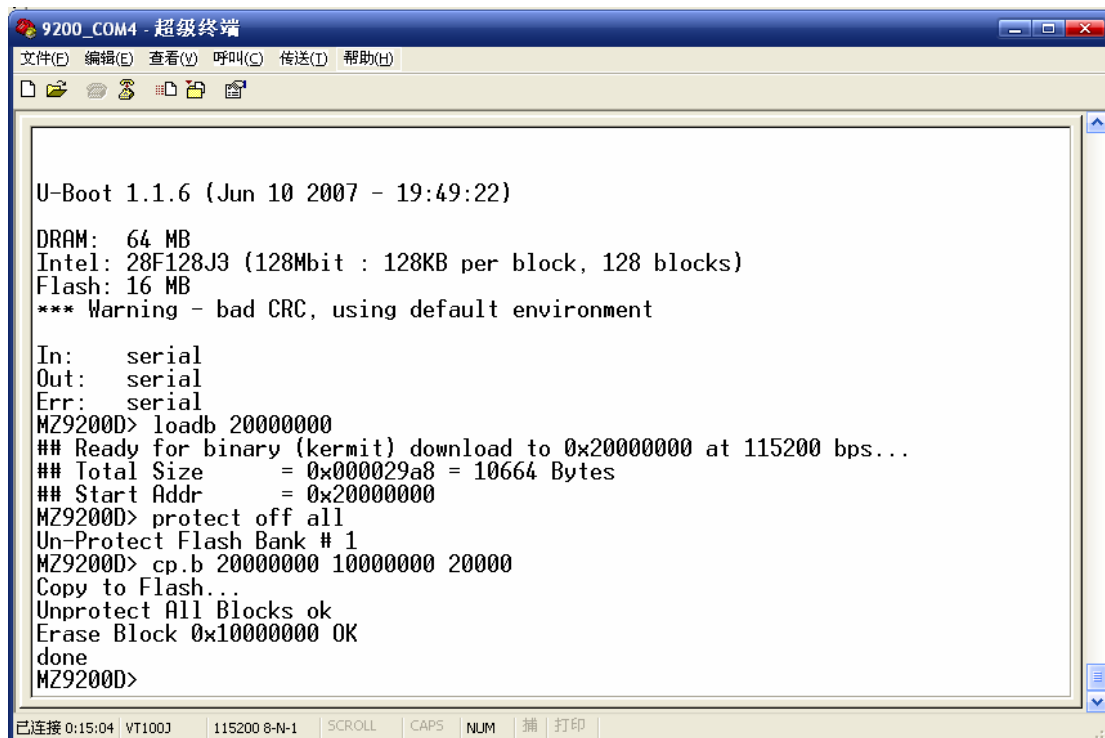
U-Boot 1.1.6 (Jun 10 2007 - 19:49:22)

DRAM: 64 MB
Intel: 28F128J3 (128Mbit : 128KB per block, 128 blocks)
Flash: 16 MB
*** Warning - bad CRC, using default environment

In: serial
Out: serial
Err: serial
MZ9200D> loadb 20000000
## Ready for binary (kermit) download to 0x20000000 at 115200 bps...
## Total Size      = 0x000029a8 = 10664 Bytes
## Start Addr      = 0x20000000
MZ9200D>
```

图 11

输入 protect off all 命令，去掉 FLASH 写保护，然后使用 cp 命令把下载下来的 9200flashloader.bin 复制到 FLASH 内：



```
U-Boot 1.1.6 (Jun 10 2007 - 19:49:22)

DRAM: 64 MB
Intel: 28F128J3 (128Mbit : 128KB per block, 128 blocks)
Flash: 16 MB
*** Warning - bad CRC, using default environment

In: serial
Out: serial
Err: serial
MZ9200D> loadb 20000000
## Ready for binary (kermit) download to 0x20000000 at 115200 bps...
## Total Size      = 0x000029a8 = 10664 Bytes
## Start Addr      = 0x20000000
MZ9200D> protect off all
Un-Protect Flash Bank # 1
MZ9200D> cp.b 20000000 10000000 20000
Copy to Flash...
Unprotect All Blocks ok
Erase Block 0x10000000 OK
done
MZ9200D>
```

图 12

接下来下载并复制 u-boot.bin 到 FLASH 区段：



图 13

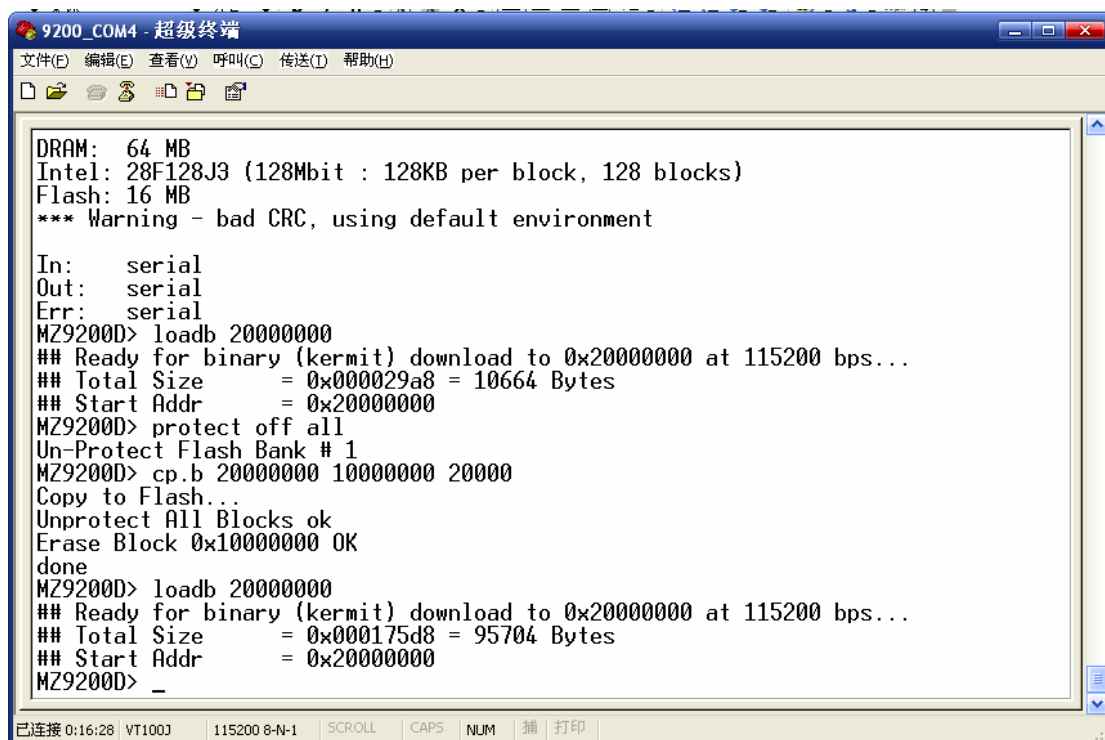
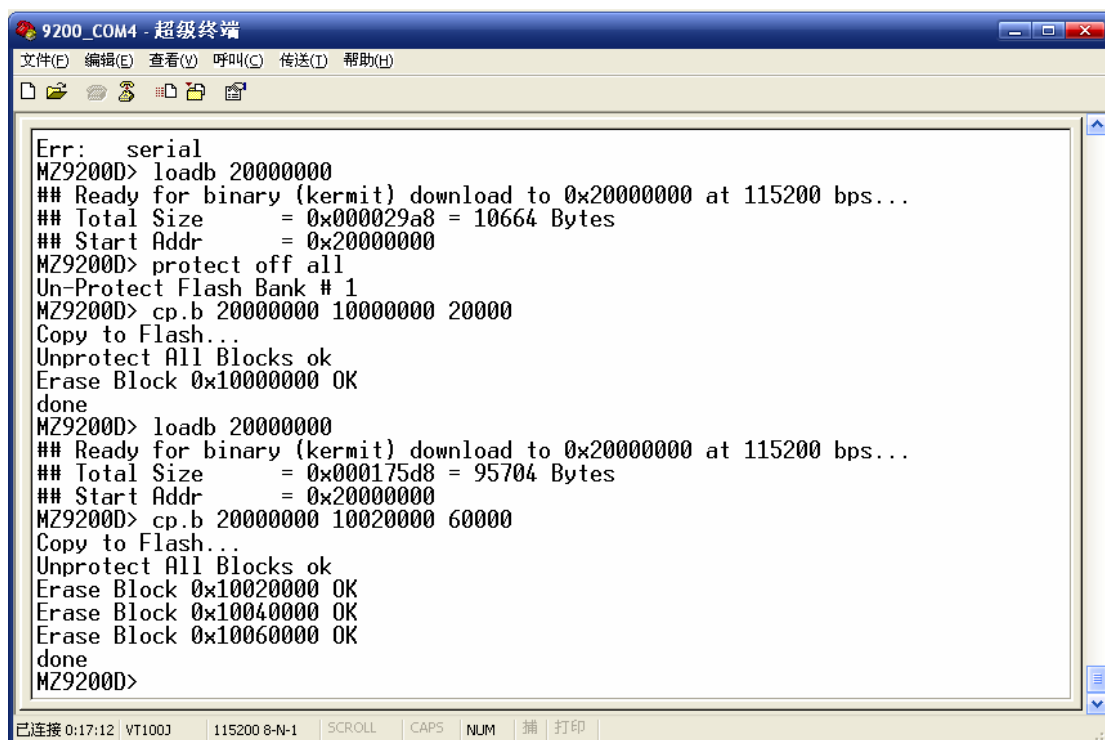


图 14

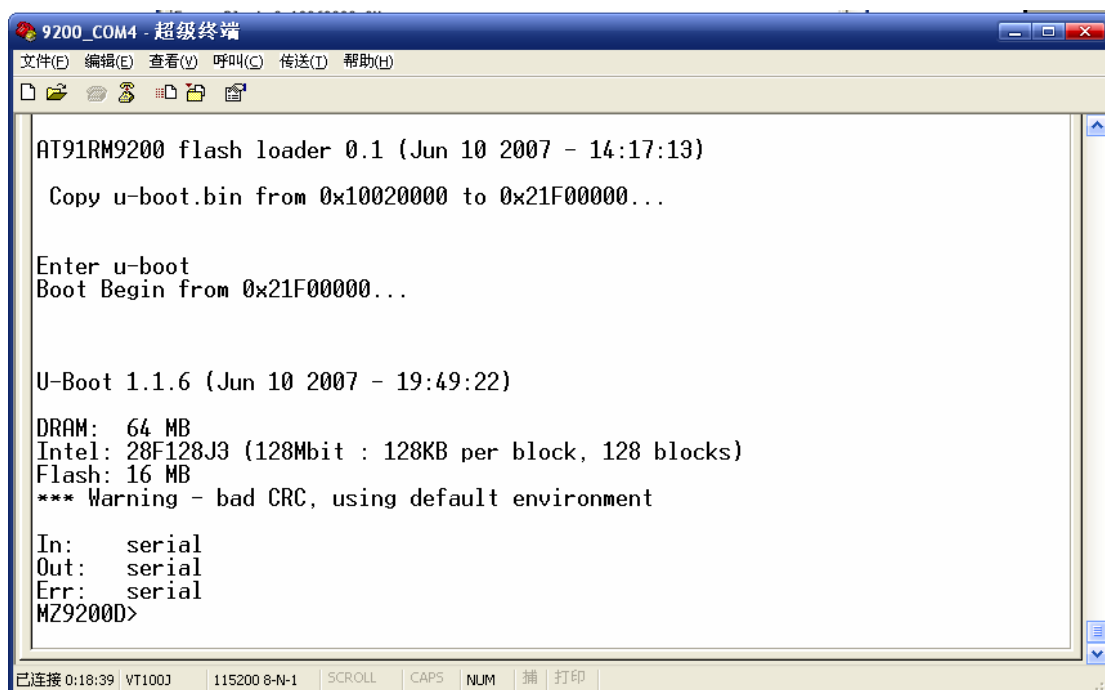
复制到 FLASH:



```
Err: serial
MZ9200D> loadb 20000000
## Ready for binary (kermit) download to 0x20000000 at 115200 bps...
## Total Size      = 0x000029a8 = 10664 Bytes
## Start Addr      = 0x20000000
MZ9200D> protect off all
Un-Protect Flash Bank # 1
MZ9200D> cp.b 20000000 10000000 20000
Copy to Flash...
Unprotect All Blocks ok
Erase Block 0x10000000 OK
done
MZ9200D> loadb 20000000
## Ready for binary (kermit) download to 0x20000000 at 115200 bps...
## Total Size      = 0x000175d8 = 95704 Bytes
## Start Addr      = 0x20000000
MZ9200D> cp.b 20000000 10020000 60000
Copy to Flash...
Unprotect All Blocks ok
Erase Block 0x10020000 OK
Erase Block 0x10040000 OK
Erase Block 0x10060000 OK
done
MZ9200D>
```

图 15

完成以上操作后，把 FLASHBOOT 跳线短上，按下复位键，可以看到如下内容：



```
AT91RM9200 flash loader 0.1 (Jun 10 2007 - 14:17:13)

Copy u-boot.bin from 0x10020000 to 0x21F00000...

Enter u-boot
Boot Begin from 0x21F00000...

U-Boot 1.1.6 (Jun 10 2007 - 19:49:22)

DRAM: 64 MB
Intel: 28F128J3 (128Mbit : 128KB per block, 128 blocks)
Flash: 16 MB
*** Warning - bad CRC, using default environment

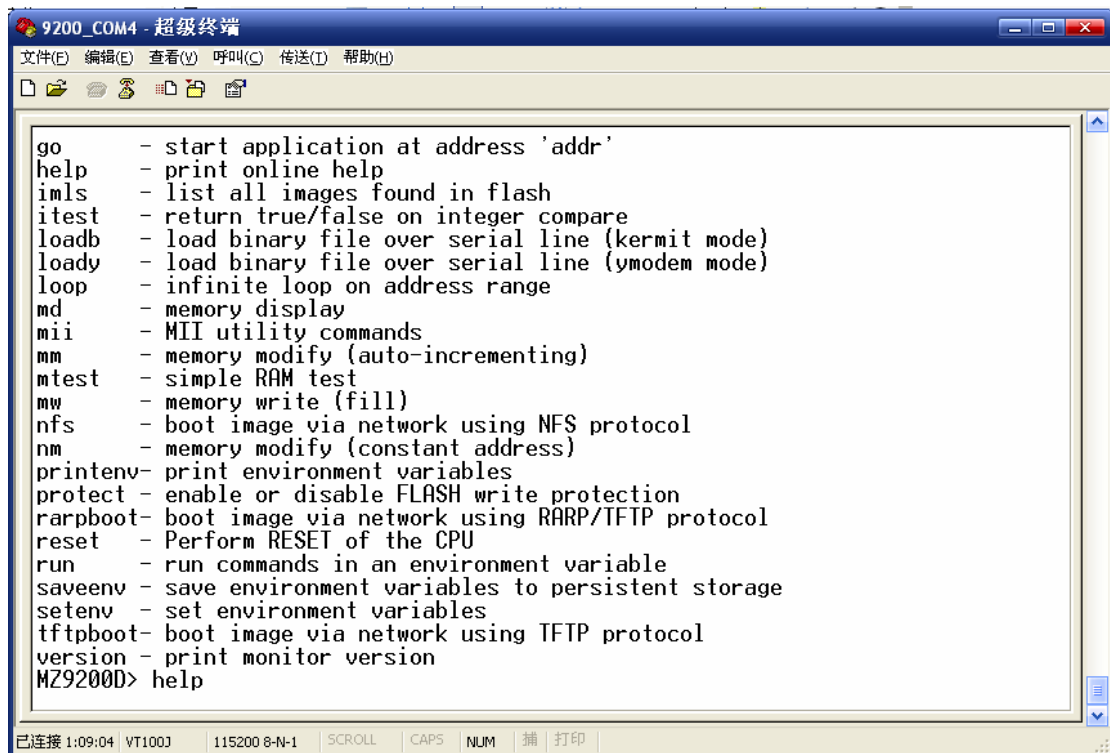
In: serial
Out: serial
Err: serial
MZ9200D>
```

图 16

说明 u-boot 已经被成功写入到了 FLASH，以后上电即可运行。



在 MZ9200 提示符下，可以用 `help` 或者 `?` 来查看 u-boot 所带的命令，每个命令都有说明，如果需要更详细的命令说明，可以查阅相关资料。



```
9200_COM4 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

go      - start application at address 'addr'
help    - print online help
imls    - list all images found in flash
itest   - return true/false on integer compare
loadb   - load binary file over serial line (kermit mode)
loady   - load binary file over serial line (ymodem mode)
loop    - infinite loop on address range
md      - memory display
mii     - MII utility commands
mm      - memory modify (auto-incrementing)
mtest   - simple RAM test
mw      - memory write (fill)
nfs     - boot image via network using NFS protocol
nm      - memory modify (constant address)
printenv- print environment variables
protect - enable or disable FLASH write protection
rarpboot- boot image via network using RARP/TFTP protocol
reset   - Perform RESET of the CPU
run     - run commands in an environment variable
saveenv - save environment variables to persistent storage
setenv  - set environment variables
tftpboot- boot image via network using TFTP protocol
version - print monitor version
MZ9200> help
```

图 17