

# X210V3 SD卡烧写教程

---



深圳市九鼎创展科技有限公司  
[www.9tripod.com](http://www.9tripod.com)



## 版权声明

本手册版权归属深圳市九鼎创展科技有限公司所有，并保留一切权力。非经九鼎创展同意(书面形式)，任何单位及个人不得擅自摘录本手册部分或全部，违者我们将追究其法律责任。

敬告：

在售开发板的手册会经常更新，请在 <http://www.9tripod.com> 网站下载最新手册，不再另行通知。



## 版本说明

版本号	日期	作者	描述
Rev.01	2013-07-24	lqm	原始版本
Rev.02	2013-8-27	lqm	增加 WINCE SD 卡量产映像说明



## 技术支持

如果您对文档有所疑问，您可以在办公时间（星期一至星期五上午 9:00~12:00；下午 1:30~6:00）拨打技术支持电话或 E-mail 联系。

网 址： [www.9tripod.com](http://www.9tripod.com)

联系电话： 0755-33133436

E-mail: [phosphor88@163.com](mailto:phosphor88@163.com)

## 销售与服务网络

公司：深圳市九鼎创展科技有限公司

地址：深圳市宝安区西乡街道宝源路宝安互联网产业基地 A 区 7 栋 301 室

邮编：518101

电话：0755-33133436

网址：<http://www.9tripod.com>

论坛：<http://bbs.9tripod.com>

<http://www.xboot.org>

淘宝：<http://armeasy.taobao.com>

QQ 群：

x6410 技术论坛： 【16073601】

x210 技术论坛 1： 【23831259】

x210 技术论坛 2： 【211127570】

x210 技术论坛 3： 【211128231】

i210 技术论坛 1： 【159144256】

i210 技术论坛 2： 【189920370】

i210 技术论坛 3： 【199358213】

# 目录

## 目录

版权声明 .....	II
第 1 章 x210v3 开发平台 SD 卡烧写教程 .....	2
1.1 拨码开关设置 .....	2
1.2 破坏开发板 linux 平台下的 bootloader, 从 SD2 启动开发板 .....	2
1.3 破坏开发板 android 平台下的 bootloader, 从 SD2 启动开发板 .....	2
1.4 破坏 WINCE 平台下的 bootloader, 从 SD2 启动开发板 .....	3
1.5 制作量产卡 .....	4
1.6 使用 uboot 通过 SD 卡更新 linux 系统 .....	7
1.7 使用 uboot 通过 SD 卡更新 android4.0 系统 .....	10
1.8 使用 uboot 通过 SD 卡更新 android2.3 系统 .....	10
1.9 使用 xboot 通过 SD 卡更新 linux 系统 .....	10
1.10 使用 xboot 通过 SD 卡更新 android4.0 系统 .....	10
1.11 在 WINXP 下快速制作量产卡 .....	10
1.12 使用 xboot 通过 SD 卡更新 WINCE 系统 .....	15
1.13 使用 eboot 通过 SD 卡更新 WINCE 系统 .....	17
1.13.1 SD 卡自动升级模式 .....	17
1.13.2 SD 卡手动升级模式 .....	19
1.14 开发板 uboot 启动参数 .....	20
1.15 nand flash 和 emmc 启动速度对照表 .....	21
1.16 WINCE 系统如何切换成 android 系统 .....	21
1.17 WINCE 系统如何切换成 linux 系统 .....	21
1.18 android 系统如何切换成 WINCE 系统 .....	21
1.19 linux 系统如何切换成 android 系统 .....	21
1.20 QT4.8 系统如何切换成 QTOPIA 系统 .....	21
1.21 QTOPIA 系统如何切换成 QT4.8 系统 .....	21
1.22 linux 或 android 平台使用 SD 卡烧写方式调试程序 .....	21
1.23 WINCE 平台开机 LOGO 制作 .....	22
1.24 WINCE 平台休眠唤醒 .....	24
1.25 WINCE 平台如何校屏 .....	24
第 2 章 其他产品介绍 .....	25
2.1 核心板系列 .....	25
2.1.1 6410 核心板 .....	25
2.1.2 210 核心板 .....	25
2.2 开发板系列 .....	25
2.2.1 6410 开发板 .....	25
2.2.2 210 开发板 .....	25



## 第1章 x210v3 开发平台 SD 卡烧写教程

x210v3 开发板支持 WINCE, QT, android2.3, android4.0 四套操作系统, 同时, 支持 uboot 和 xboot 引导 linux 和 android 内核。本文档详细描述使用 SD 卡更新映像文件的方式。如需其他方式下载, 请参考相关操作系统用户手册。

**注意: 本文仅针对 x210ii, x210v3 emmc 平台, 如果使用 nand flash, 请参考相关手册, 下文不再声明。**

### 1.1 拨码开关设置

开关	SW3				SW4			
引脚	1	2	3	4	5	6	7	8
接口	OM0	OM1	OM2	OM3	OM4	OM5	NC	NC
ON(上)	VCC	VCC	VCC	VCC	VCC	VCC	NC	NC
OFF(下)	GND	GND	GND	GND	GND	GND	NC	NC

Nand 启动时, 配置如下:

引脚	1	2	3	4	5	6	7	8
接口	OM0	OM1	OM2	OM3	OM4	OM5	NC	NC
状态	VCC	VCC	GND	GND	GND	GND	NC	NC

SD 卡或 inand 启动时, 配置如下:

引脚	1	2	3	4	5	6	7	8
接口	OM0	OM1	OM2	OM3	OM4	OM5	NC	NC
状态	VCC	GND	VCC	VCC	GND	GND	NC	NC

USB 调试模式时, 配置如下:

引脚	1	2	3	4	5	6	7	8
接口	OM0	OM1	OM2	OM3	OM4	OM5	NC	NC
状态	VCC	X	X	X	X	VCC	NC	NC

### 1.2 破坏开发板 linux 平台下的 bootloader, 从 SD2 启动开发板

有时候我们需要从 SD 卡引导 bootloader, 由于 S5PV210 芯片的 SD0 和 SD2 启动的拨码开关完全相同, 默认 CPU 会首先从 SD0 启动, 如果 SD0 没有数据, 才会从 SD2 启动。而 x210 开发板 SD0 接了 emmc, 如果 emmc 中有 bootloader, 上电后会自动引导 emmc 中的 bootloader, 这时, 如果我们需要从 SD2 启动, 就需要破坏掉 emmc 中的 bootloader。

进入 linux 系统控制台, 执行如下指令:

```
busybox dd if=/dev/zero of=/dev/mmcblk0 bs=512 seek=1 count=1 conv=sync
```

为了确保执行数据有效, 再执行 sync 同步一下, 即可破坏掉前一个块的数据。

### 1.3 破坏开发板 android 平台下的 bootloader, 从 SD2 启动开发板

进入 android 系统控制台, 执行如下指令:

```
busybox dd if=/dev/zero of=/dev/block/mmcblk0 bs=512 seek=1 count=1 conv=sync
```

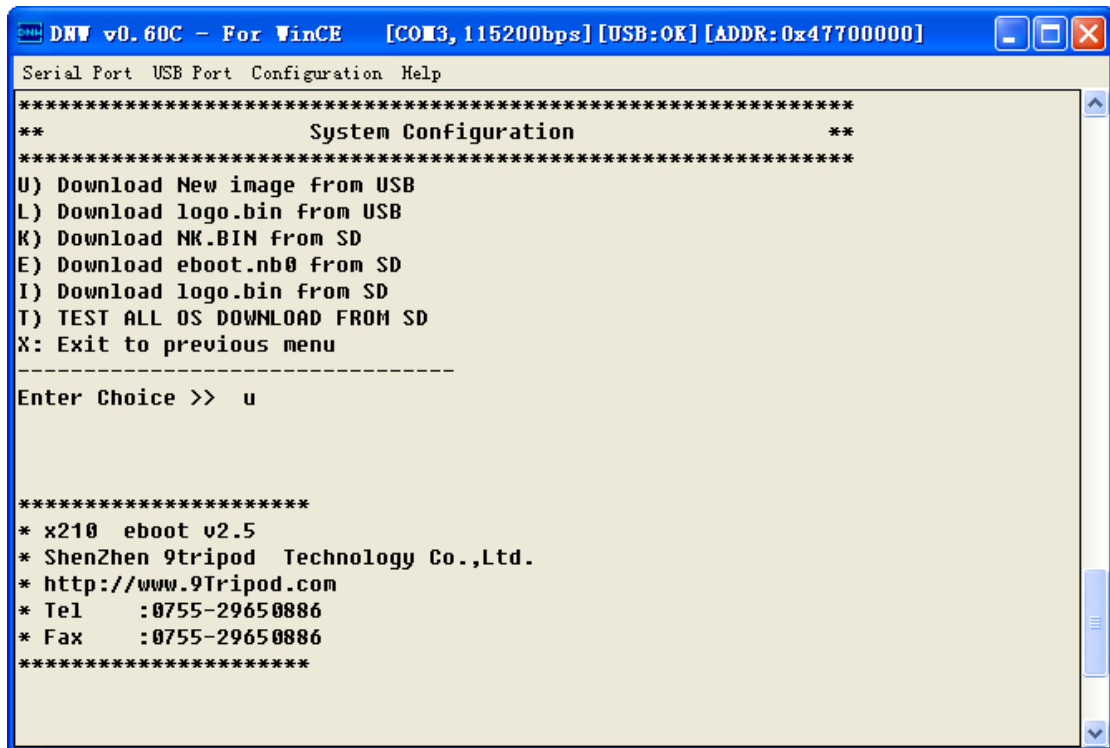
为了确保执行数据有效, 再执行 sync 同步一下, 即可破坏掉前一个块的数据。



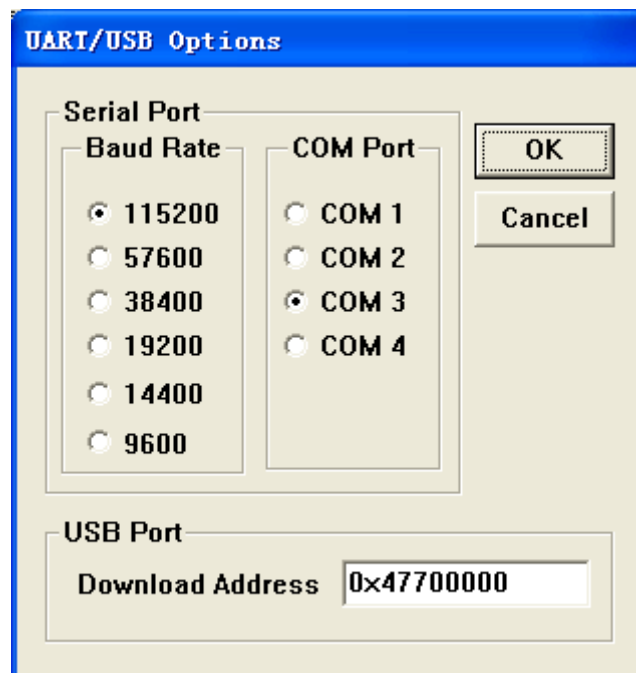
## 1.4 破坏 WINCE 平台下的 bootloader，从 SD2 启动开发板

第一步：在 eboot 5 秒倒计时的时候，按下 PC 机的空格键，进入 eboot 菜单；

第二步：确保 DNW 的地址为 0x47700000，按下 u，如下图所示：



如果地址不是 0x47700000，点击 DNW 的 Configuration，选择 Options，在最下面的 Download Address 一栏设置为 0x47700000 即可：



第三步：点击 DNW 的 USB Port->Uboot，选择 erase.nb0 文件，烧写完成后，即清掉了 eboot，这时，我们就可以从 SD2 通道通过外部 SD 卡启动了。



## 1.5 制作量产卡

第一步：准备一张不小于 2GB 的 SD 卡，并插到装有 ubuntu 操作系统的 PC 机上；

第二步：删除 SD 卡的所有分区。

在 Linux 的终端窗口，使用 `fdisk /dev/sdb` 命令删除原来所有分区，`sdb` 为系统为 SD 卡分配的设备节点。注意，具体由节点名称而定，有可能是 `sdc,sde` 等。使用如下指令查询设备节点：

```
cat /proc/partitions
```

示例如下：

```
[root@lqm mass-production]# cat /proc/partitions
```

```
major minor  #blocks  name
```

```
8          0   36700160 sda
8          1    512000 sda1
8          2   36187136 sda2
253         0   34144256 dm-0
253         1    2031616 dm-1
8          16    3879936 sdb
8          17    3875840 sdb1
```

```
[root@lqm mass-production]#
```

```
[root@lqm mass-production]# fdisk /dev/sdb
```

```
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').
```

```
Command (m for help): d
```

```
Selected partition 1
```

```
Command (m for help): w
```

```
The partition table has been altered!
```

```
Calling ioctl() to re-read partition table.
```

```
WARNING: Re-reading the partition table failed with error 16: 设备或资源忙.
```

```
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
```

```
Syncing disks.
```

```
[root@lqm mass-production]#
```

输入 `d`，表示删除分区，输入 `w` 表示保存已经修改的分区信息。至此，原 `/dev/sdb1` 被删除。拔掉 SD 卡，再插入 PC 机上，查询设备节点：

```
[root@lqm mass-production]# cat /proc/partitions
```

```
major minor  #blocks  name
```





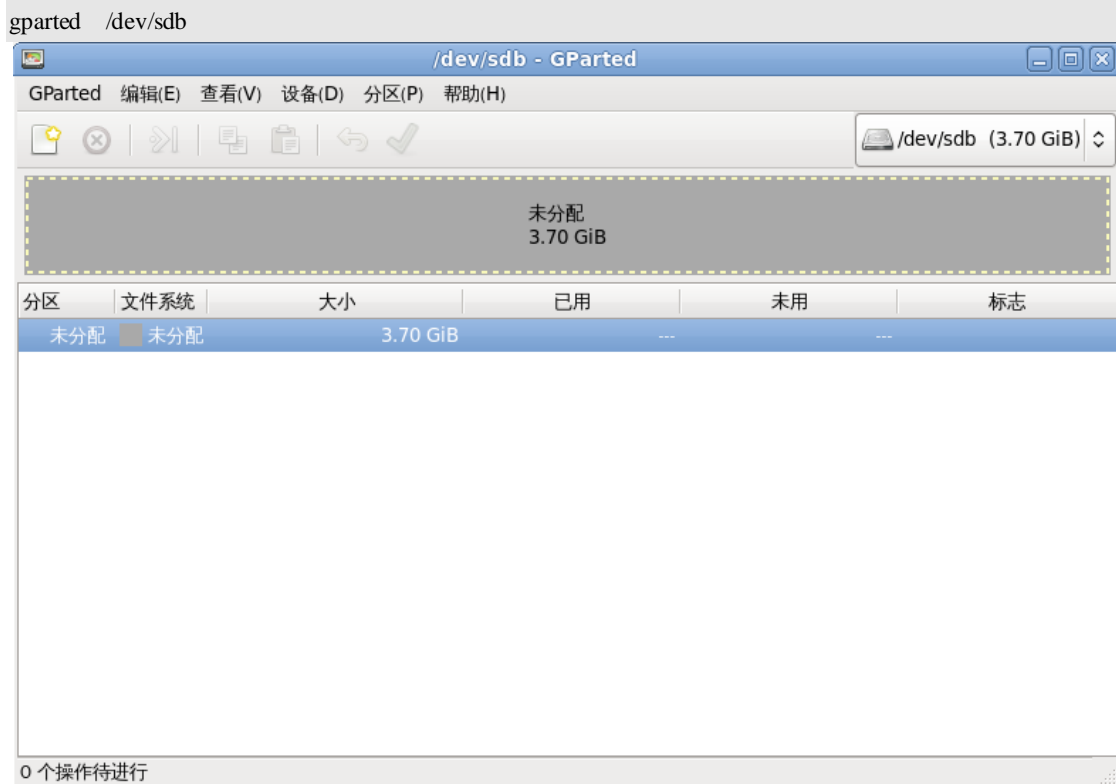
```
8      0   36700160 sda
8      1    512000 sda1
8      2   36187136 sda2
253    0   34144256 dm-0
253    1    2031616 dm-1
8     16   3879936 sdb
```

```
[root@lqm mass-production]#
```

注意必须拔掉后再插入，否则仍然会提示存在/dev/sdb1 节点，会造成出错。

第三步：使用 **gparted** 工具给 SD 卡预留 256M 空间，用于存放 **uboot** 或 **xboot** 映像。

使用如下命令打开 SD 卡分区表：



选择分区->新建，预留 256M 空间给 **uboot** 或 **xboot**，剩下的分区使用 **fat32** 格式，如下图所示：



0 个操作待进行

点击添加，选择菜单中的应用全部操作，完成 SD 卡的分区。

第四步：将 SD 卡剩余的空间格式化为 fat32 格式

```
sudo ./mkfs.vfat /dev/sdb1
```

第五步：将使用 xboot 升级需要的映像文件 qt-update.bin，android-update.bin 文件拷贝到 SD 卡根目录，同时在 SD 卡根目录下新建一个名为 backdoor 的文件，内容为空即可。

**说明：此步仅用于使用 xboot 更新文件系统，如果使用 uboot，可跳过此步。**

第六步：在 SD 卡根目录下新建名为 x210 的目录，并将需要更新的映像拷贝到该目录。

为了以防读者混淆，这里列出各自系统需要更新的文件：

android2.3 系统：

uboot.bin zImage x210.img

android4.0 系统：

uboot.bin zImage-android x210.img

linux+QT4.8 系统：

uboot.bin zImage-qt rootfs\_qt4.ext3

linux+QTOPIA 系统：

uboot.bin zImage-qt rootfs\_qt4.ext3

**注意，linux+QTOPIA 系统的文件系统，原光盘中名称为 rootfs\_qtopia.ext3，需要将该文件重命名后再拷贝到 SD 卡 x210 目录。**

**说明：此步仅用于使用 uboot 更新文件系统，如果使用 xboot，可跳过此步。**

第七步：将 WINCE 需要更新的文件 eboot.nb0，logo.bin，nk.bin 以及配置文件 updata 拷贝到 SD 卡根目录。

**说明：如果仅需制作 WINCE 的升级卡，可省去前六步。如果无需升级 WINCE，可跳过此步。**

第八步：将对应的 bootloader(uboot.bin，xboot.bin 或 9tripod\_boot.nb0，三者选其一)烧写到 SD 卡，即完成量产卡的制作。

烧写 uboot.bin：



确保 SD 卡插到运行 linux 系统的 PC 机上，使用 uboot 的 sd\_fusing.sh 脚本将 uboot.bin 文件烧写到 SD 卡上。注意，android2.3, android4.0, linux 的 uboot 都有细小差异，请烧写指令的 uboot.bin 文件。

```
cd uboot/sd_fusing/
```

首次烧录时，查询在 uboot/sd\_fusing 目录下是否存在 mkb11 文件，如果不存在，在当前目录下 make 一下，就生成了。再执行如下指令

```
sudo ./sd_fusing.sh /dev/sdb
```

这时，uboot 就烧到 SD 卡中了。

烧写 xboot.bin:

注意，android4.0 和 QT 的 xboot 不一样，请烧写指定的 xboot。运行如下命令：

```
sudo ./x210-irom-sd.sh /dev/sdb xboot.bin
```

示例如下：

```
[root@lqm mass-production]# ./x210-irom-sd.sh /dev/sdb release/xboot.bin
```

```
记录了 24084+1 的读入
```

```
记录了 24085+0 的写出
```

```
12331520 字节(12 MB)已复制, 12.0886 秒, 1.0 MB/秒
```

```
^_^ The image is fused successfully
```

```
[root@lqm mass-production]#
```

烧写 9tripod\_boot.nb0:

不公布此方法，请读者自行研究源码。

## 1.6 使用 uboot 通过 SD 卡更新 linux 系统

第一步：将制作好的量产卡插到开发板的 SD2 卡槽

第二步：确保开发板的拨码开关拨到 OM[5:0]=001101

第三步：确保开发板 emmc 中存在能够通过 SD 卡升级的 uboot，如果不支持，可以先使用 USB 下载一个支持 SD 卡升级的 uboot，或者干脆将 emmc 中的 uboot 破坏掉，开发板会自动判别从外部 SD 卡启动。

第四步：按住开发板的 LEFT 键开机，这时，uboot 会从 SD 卡的 x210 目录读取 uboot.bin, zImage-qt, rootfs\_qt4.ext3 三个文件并升级。升级完成后将自动重启。

使用 SD 卡升级的串口打印信息如下：

```
OK
```

```
U-Boot 1.3.4 (Jul 23 2013 - 15:16:39) for x210
```

```
CPU: S5PV210@1000MHz(OK)
```

```
APLL = 1000MHz, HclkMsys = 200MHz, PclkMsys = 100MHz
```

```
MPLL = 667MHz, EPLL = 96MHz
```

```
HclkDsys = 166MHz, PclkDsys = 83MHz
```

```
HclkPsys = 133MHz, PclkPsys = 66MHz
```

```
SCLKA2M = 200MHz
```

```
Serial = CLKUART
```

```
Board: X210
```



```
DRAM:      512 MB
Flash:     8 MB
SD/MMC: 3776MB
*** Warning - using default environment

In:        serial
Out:       serial
Err:       serial
[LEFT DOWN] update mode
[Fusing Image from SD Card.]
[Partition table on MoviNAND]
ptn 0 name='bootloader' start=0x0 len=N/A (use hard-coded info. (cmd: movi))
ptn 1 name='kernel' start=N/A len=N/A (use hard-coded info. (cmd: movi))
ptn 2 name='ramdisk' start=N/A len=0x300000(~3072KB) (use hard-coded info. (cmd: movi))
ptn 3 name='system' start=0xB11E00 len=0x79C4A00(~124690KB)
ptn 4 name='userdata' start=0x84D6800 len=0x401CDC00(~1050423KB)
ptn 5 name='cache' start=0x486A4400 len=0x19234200(~411856KB)
ptn 6 name='fat' start=0x618D8600 len=0x89AE5200(~2255764KB)
Partition: bootloader, File: /x210/uboot.bin
Partition1: Start Address(0x58a4), Size(0x75612c)
reading /x210/uboot.bin
409600 (0x00064000) bytes read
flashing 'bootloader'
Writing BL1 to sector 1 (16 sectors).. checksum : 0xece97
writing bootloader.. 49, 1024
MMC write: dev # 0, block # 49, count 1024 ... 1024 blocks written: OK
completed
partition 'bootloader' flashed
Partition: kernel, File: /x210/zImage-qt
Partition1: Start Address(0x58a4), Size(0x75612c)
reading /x210/zImage-qt
3616768 (0x00373000) bytes read
flashing 'kernel'
writing kernel.. 1073, 8192
MMC write: dev # 0, block # 1073, count 8192 ... 8192 blocks written: OK
completed
partition 'kernel' flashed
Partition: system, File: /x210/rootfs_qt4.ext3
Partition1: Start Address(0x58a4), Size(0x75612c)
reading /x210/rootfs_qt4.ext3
125829120 (0x07800000) bytes read
flashing 'system'
```



MMC write: dev # 0, block # 22671, count 249381 ... 249381 blocks written: OK

partition 'system' flashed

Partition: userdata

erasing(formatting) 'userdata'

Start format MMC0 partition3 ....

\*\* Partition3 is not ext4 file-system> 0 \*\*

Partition3: Start Address(0x426b4), Size(2100846)

Creating filesystem with parameters:

Size: 1075633152

Block size: 4096

Blocks per group: 32768

Inodes per group: 7295

Inode size: 256

Journal blocks: 4103

Label:

Blocks: 262605

Block groups: 9

Reserved block group size: 71

Created filesystem with 11/65655 inodes and 8593/262605 blocks

.....

Write Done.

partition 'userdata' erased

Partition: cache

erasing(formatting) 'cache'

Start format MMC0 partition4 ....

\*\* Partition4 is not ext4 file-system> 0 \*\*

Partition4: Start Address(0x243522), Size(823713)

Creating filesystem with parameters:

Size: 421741056

Block size: 4096

Blocks per group: 32768

Inodes per group: 6436

Inode size: 256

Journal blocks: 1608

Label:

Blocks: 102964

Block groups: 4

Reserved block group size: 31

Created filesystem with 11/25744 inodes and 3330/102964 blocks

.....

Write Done.

partition 'cache' erased

reset...



## 1.7 使用 uboot 通过 SD 卡更新 android4.0 系统

第一步：将制作好的量产卡插到开发板的 SD 卡槽 2；

第二步：确保开发板的拨码开关拨到 OM[5:0]=001101

第三步：确保 emmc 中烧写的 uboot 是对应能够更新 android4.0 系统的映像，否则破坏掉 emmc 的第一个块；

第四步：按住 LEFT 键后开机，即进入升级过程。升级完成后开发板会自动重启。

## 1.8 使用 uboot 通过 SD 卡更新 android2.3 系统

第一步：将制作好的量产卡插到开发板的 SD 卡槽 2；

第二步：确保开发板的拨码开关拨到 OM[5:0]=001101

第三步：确保 emmc 中烧写的 uboot 是对应能够更新 android2.3 系统的映像，否则破坏掉 emmc 的第一个块；

第四步：按住 LEFT 键后开机，即进入升级过程。升级完成后开发板会自动重启。

## 1.9 使用 xboot 通过 SD 卡更新 linux 系统

第一步：将制作好的量产卡插到开发板的 SD 卡槽 2；

第二步：确保开发板的拨码开关拨到 OM[5:0]=001101

第三步：按着 LEFT 键 + POWER 键开机，直到出现选择菜单界面；再通过上下键选择菜单，这里选择第二项，update system；按确认键即开始升级。

## 1.10 使用 xboot 通过 SD 卡更新 android4.0 系统

第一步：将制作好的量产卡插到开发板的 SD 卡槽 2；

第二步：确保开发板的拨码开关拨到 OM[5:0]=001101

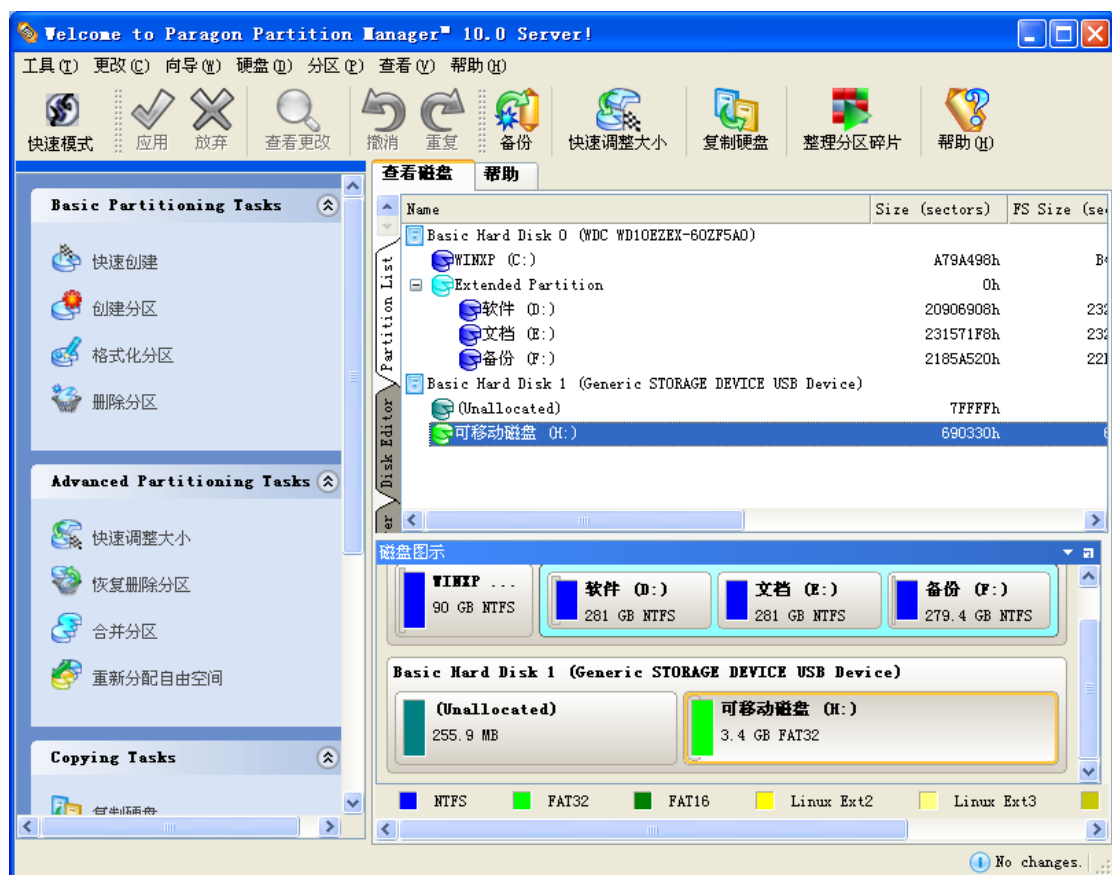
第三步：按着 LEFT 键 + POWER 键开机，直到出现选择菜单界面；再通过上下键选择菜单，这里选择第二项，update system；按确认键即开始升级。

## 1.11 在 WINXP 下快速制作量产卡

第一步：准备一张不小于 2G 的 SD 卡，通过读卡器接到装有 Windows XP 系统的 PC 机上；

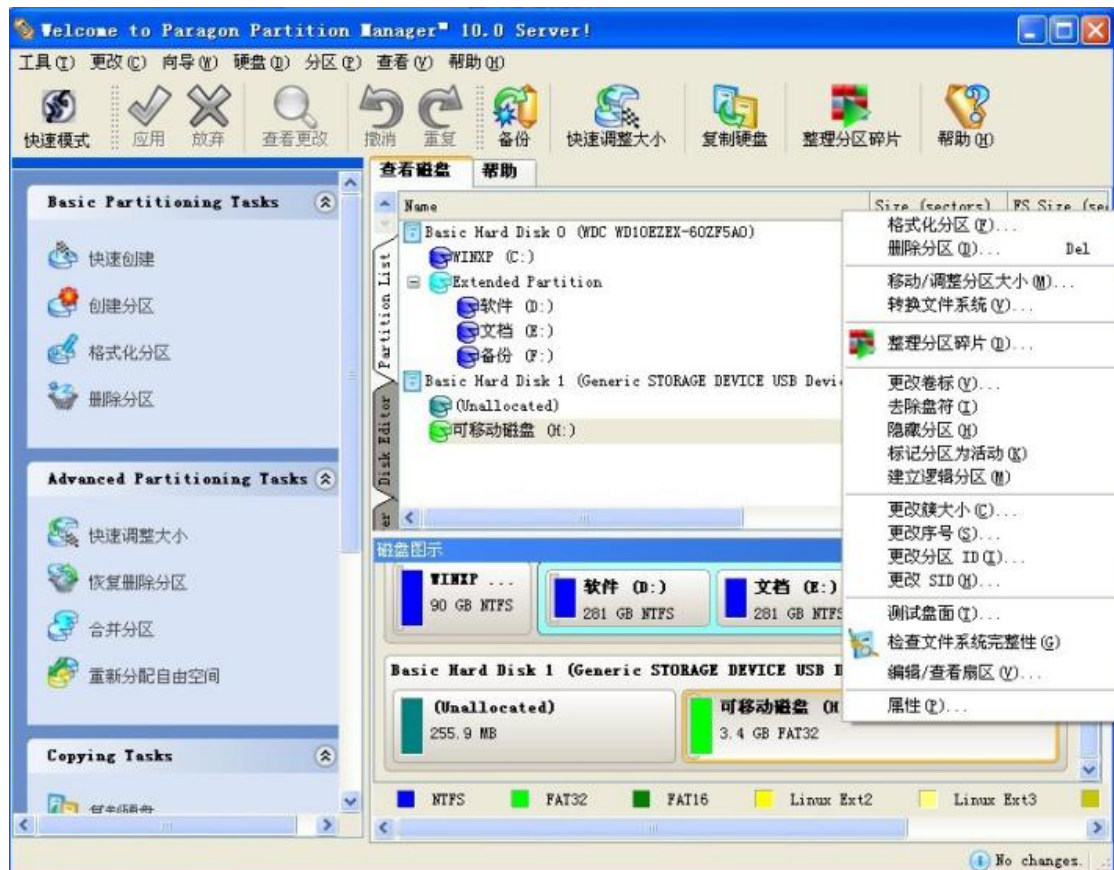
第二步：给 SD 卡分区

从光盘中找到 PartitionManager.exe 工具并打开，界面如下：

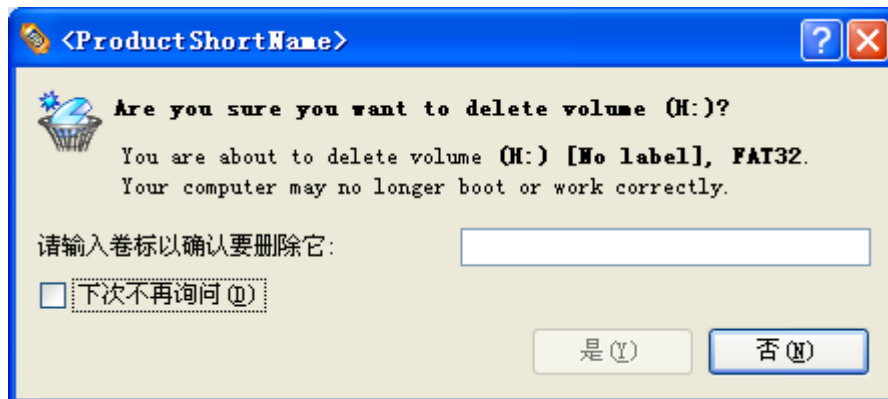


上图中H盘即为我们插入的SD卡分区。我们需要使用这个工具给SD卡预留一些空间，用于存放 uboot 或 xboot。首先我们右键点击最下面图标的可移动磁盘(H:)，点击删除分区，如下图：





软件会弹出确认界面：




勾上下次不再询问，点击是，即删除了原来的分区。这时，SD 卡就只剩一个分区了。再次右键点击该分区，点击创建分区：







<ProductShortName>

 Are you sure you want to create a new partition on disk 1?  
You are about to create a new partition in (Unallocated), 3.6 GB area.  
Please select size, position and file system of the new partition.

Basic Hard Disk 1 (Generic STORAGE DEVICE USB Device)

  New Volume (I:)  
3.6 GB NTFS

创建新分区为: 主分区

请指定新分区大小: 3681 7 MB - 3,781 MB


请指定分区前面的自由空间大小: 100 0 MB - 3,773 MB

请指定分区后面的自由空间大小: 0 0 MB - 3,773 MB

请为新分区选择文件系统(S): NTFS

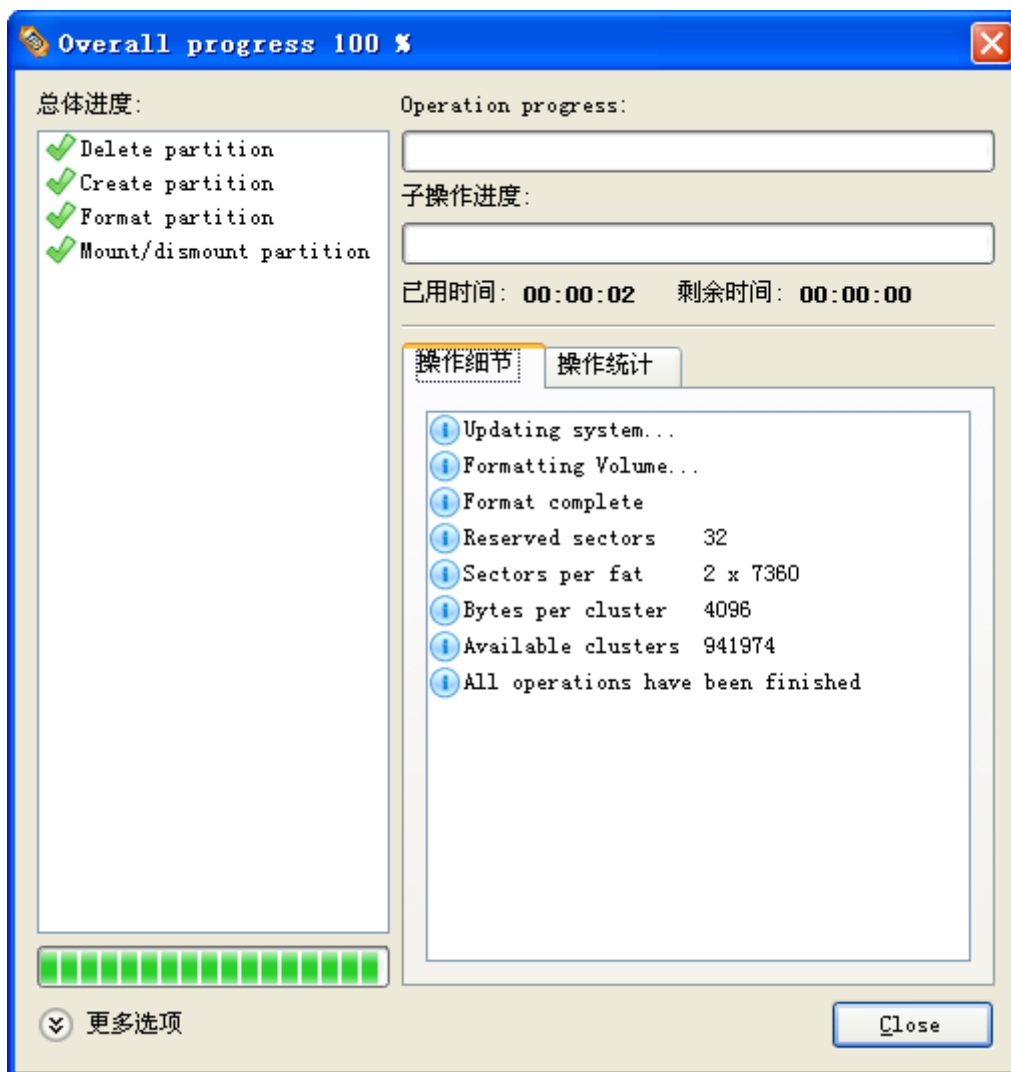
请输入新卷标: New Volume

请指定盘符(Q): I:

 更多选项

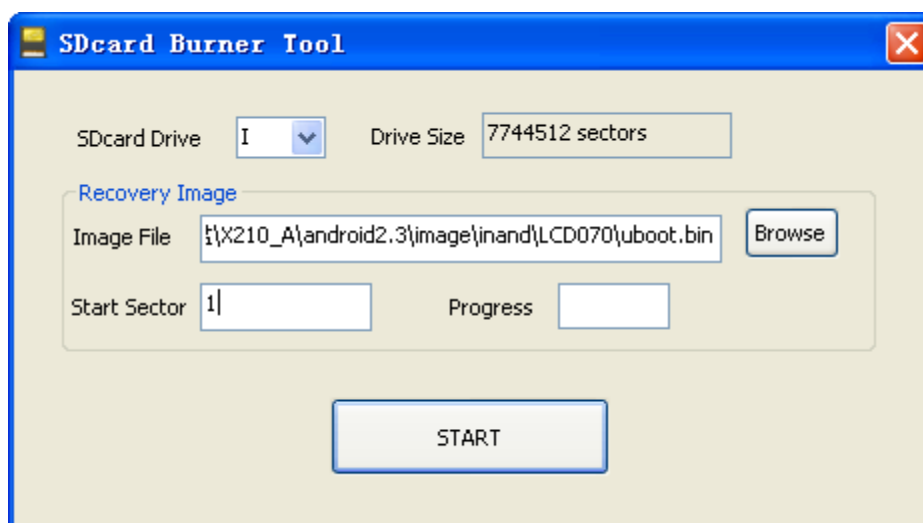
是(Y) 否(N)

在请指定分区前面的自由空间大小一栏中填入我们需要预留的空间,我们这里填100M,留给uboot或xboot已经足够用了。在请为新分区选择文件系统中选择FAT32,点击是,然后再到主界面快捷菜单栏点击应用即可。成功分区后提示如下:



第三步：将 uboot 或 xboot 映像文件烧写到 SD 卡的预留空间

烧写 uboot：从光盘中找到 SDCardBurner.exe 工具并打开，在 Sdcard Drive 中找到对应 SD 卡的盘符，这时右侧的 Drive Size 会自动显示出 SD 卡的扇区数。在 Image File 一栏，点击右侧的 Browse，找到我们需要下载的 uboot.bin 映像文件，在 Start Sector 中填 1，注意千万不要填 0！



点击 START，稍等几秒，烧写完成后会提示 Fusing image done。再将 Start Sector 改为 49，再次点击 START，待烧写完成，这时，可以从 SD 卡启动的 uboot 即制作完成。

说明：注意，烧写 uboot 到 SD 卡中时，uboot 需要将 uboot 映像的前 8K 写到 SD 卡 1 扇区开始的位置，同时需要将 uboot 的整个映像写到第 49 扇区开始的位置，这时 uboot 才能从 SD 卡启动。详细可参考 uboot 的 sd\_fusing 目录下的烧写脚本。Uboot 的前 8K 可以通过脚本提取出来，这时我们烧写 uboot 就需要指定两个文件，操作起来有点麻烦。为此，我们制作了修改 uboot 前 8K 的程序，并已集成到最新的源码包中，这样最终生成的 uboot 映像，前 8K 我们已经做过修改，那么使用 Sdcard Burner Tool 工具，就只需选择同一个 uboot.bin 文件，只不过需要将该文件同时烧写到第 1 个和第 49 个扇区的位置。

烧写 xboot：相对烧写 uboot，烧写 xboot 只需将 xboot.bin 文件烧写到第一个扇区开始的位置就可以了。同样先选择 SD 卡的盘符，再找到 xboot.bin 映像文件，最后将 Start Sector 改为 1，点击 START 即可烧写成功。

第四步：拷贝升级文件

对于 uboot，在 SD 卡的根目录建立名为 x210 的目录，并将需要升级的映像文件拷贝到 x210 目录即可；

对于 xboot，将 android-update.bin 文件或 qt-update.bin 文件拷贝到 SD 卡根目录，同时在根目录下新建一个名为 backdoor 的空文件即可。

对于 eboot，无需进行前面的操作，只需将要更新的 eboot.nb0,logo.bin,nk.bin 文件拷贝到 FAT32 格式的 SD 卡根目录，再将配置文件 updata 拷贝到 SD 卡根目录即可。

至此，量产卡制作成功。有了 WINXP 下的制作量产卡的方式，我们平时调试量产，就方便多了。

## 1.12 使用 xboot 通过 SD 卡更新 WINCE 系统

注意，使用 xboot 更新 WINCE 系统，xboot 为重新定制的映像文件，与前面的 xboot 映像有区别。目前使用 xboot 通过 SD 卡更新 WINCE 系统时，无法同时更新 LOGO 和 NK，需要量产的用户可以通过配置文件首先更新 eboot 和 NK，更新完后再使用已经更新到 emmc 中的 eboot 更新 LOGO。为了便于用户使用一张 SD 卡更新所有操作系统，我们对配置文件做了调整，使用 xboot 更新映像时，需要从 SD 卡中读取名为 9tripod.ini 的文件，而使用 eboot 更新映像时，需要从 SD 卡中读取名为 updata 的文件，这时，我们只需将 9tripod.ini 中打开更新 eboot 和 NK，在 updata 文件中打开更新 logo 即可。



第一步：参考上一章节的步骤，给 SD 卡分区；

第二步：使用 SDCardBurner.exe 或 x210\_Fusing\_Tool.exe 将 xboot\_e.bin 文件烧写到 SD 卡。这里我们将更新 WINCE 映像用的 xboot 重命名为 xboot\_e.bin，以防混淆。双击 x210\_Fusing\_Tool.exe：



点击 Browse，找到 xboot\_m.bin 文件，点击 Add，该映像将自动添加到下面的 PATH 栏：



点击 START，稍等片刻，烧写完成后，会有如下提示



点击确定，这时，xboot 已经烧写到 SD 卡。

第三步：在 SD 卡根目录新建 9tripod.ini 文件，内容如下：

```
//www.9tripod.com
eboot=T
logo=T
nk=T
```

注意，使用 xboot 更新 eboot 时，无法同时更新 LOGO 和 NK，如果我们将 LOGO 和 NK 都设置为 T，则程序会自动忽略更新 LOGO。将需要更新的映像设置为 T，不需要更新的映像设置为 F 即可。

第四步：将制作好的 SD 卡插到 x210v3 开发板的 SD2 插槽，按住 LEFT 键开机，默认 7 寸的 LCD 屏上会有蓝屏显示，并有相关的文字信息提示。更新完映像后，系统会自动重启。

提示一：如果您参考 x210v3 设计的产品并没有保留 LEFT 按键，或者您使用的 x210v3，但是不想按住 LEFT 升级，只需启动自动升级，烧写 xboot\_a.bin 即可。

提示二：如果您想快速更新 eboot,logo,nk，可以在 SD 卡中建立 9tripod.ini 文件，并将 eboot 和 nk 设置为 T，然后在 SD 卡中建立 updata 文件，并将 logo 设置为 T，这时使用 xboot 更新映像时，会首先将 eboot 和 NK 烧写到 inand，更新完后机器会自动重启，这时 xboot 的任务已经完成，重启后 eboot 得到执行，它会通过 updata 文件判断需要更新的映像。但是通过 eboot 更新映像时，必须按住 LEFT 键才能更新，这里无法去除判断 LEFT 键，否则程序会永无止境的更新下去。

提示三：如果您产品上没有保留 LEFT 键，您可以准备两张 SD 卡，第一张将 logo 设置为 T，eboot 和 NK 设置为 0，这时用于更新 LOGO。第二张卡将 eboot 和 NK 设置为 T，LOGO 为 T 和为 F 都可以，程序会自动忽略掉。这时，eboot 和 NK 得到更新，整个系统更新完成。

提示四：在没有 xboot 参与的情况下，使用 eboot 更新各映像，LOGO 和 NK 可以同时更新。

## 1.13 使用 eboot 通过 SD 卡更新 WINCE 系统

### 1.13.1 SD 卡自动升级模式

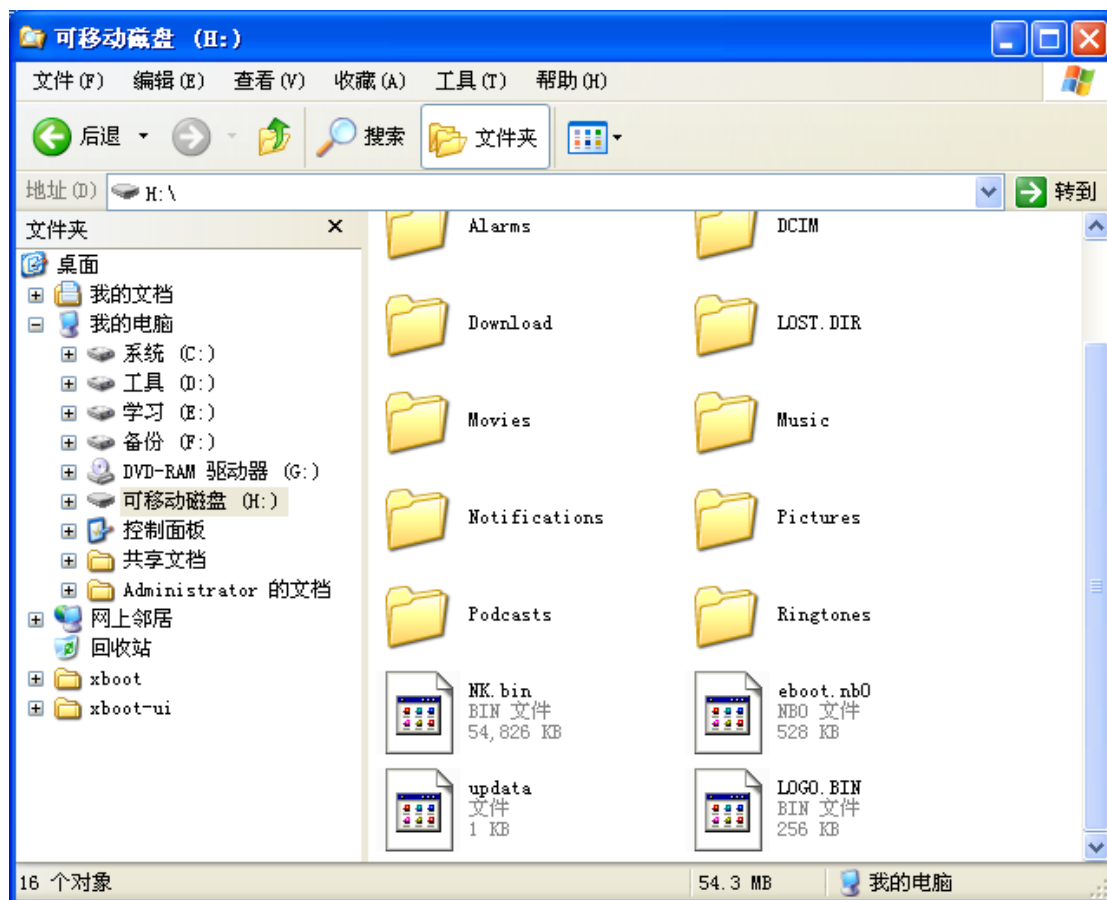
#### 1、前期准备

开发板的系统中必须先烧写有支持SD卡升级的BOOTLOADER，也就是这个版本的 9tripod\_boot.nb0 镜像。升级 9tripod\_boot.nb0 的方法，请参考《X210-II WinCE 平台用户手册》相关章节。

#### 2、准备SD卡

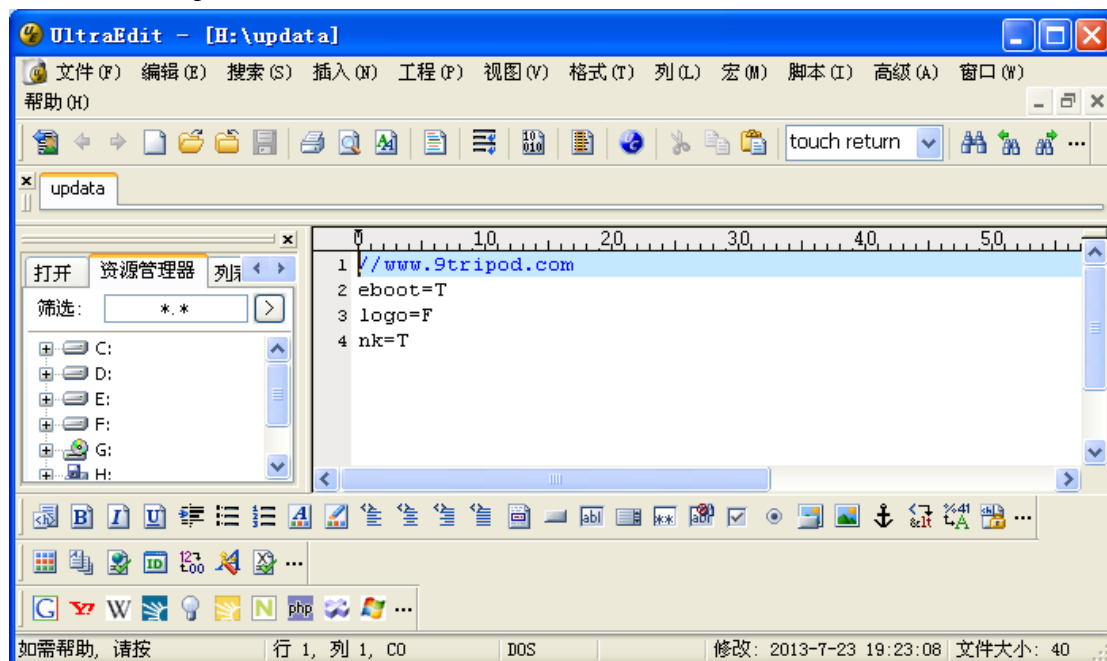
格式化成 FAT32 文件系统，复制升级的系统镜像文件和配置文件 updata 到 SD 卡到根目录下，如图 logo.bin，eboot.nb0，nk.bin 和 updata 三个文件。





注意，SD 卡中的 eboot.nb0 文件，对应光盘映像中的 9tripod\_boot.nb0 文件，并不是编译出来的 eboot.nb0 文件。

配置文件 update 内容如下：



logo=t 表示要升级 logo.bin，如果不升级，则改成 f，即 logo=f 即可。其他类似，根据用户自己的需要来改写配置文件。



### 3、插入 SD 卡

将 SD 卡插入开发板 SD 卡插槽 2

### 4、开机升级

按住 LEFT 键（即 SD 卡右边第一按键），同时按电源开关上电。待升级界面出现才可以松手。此时，LCD 屏幕进入蓝色升级界面。系统将根据配置文件 updata 里的配置来升级。如果直接进入系统，没有进入蓝色升级界面，说明没有按住 LEFT 键。

#### 1.13.2 SD 卡手动升级模式

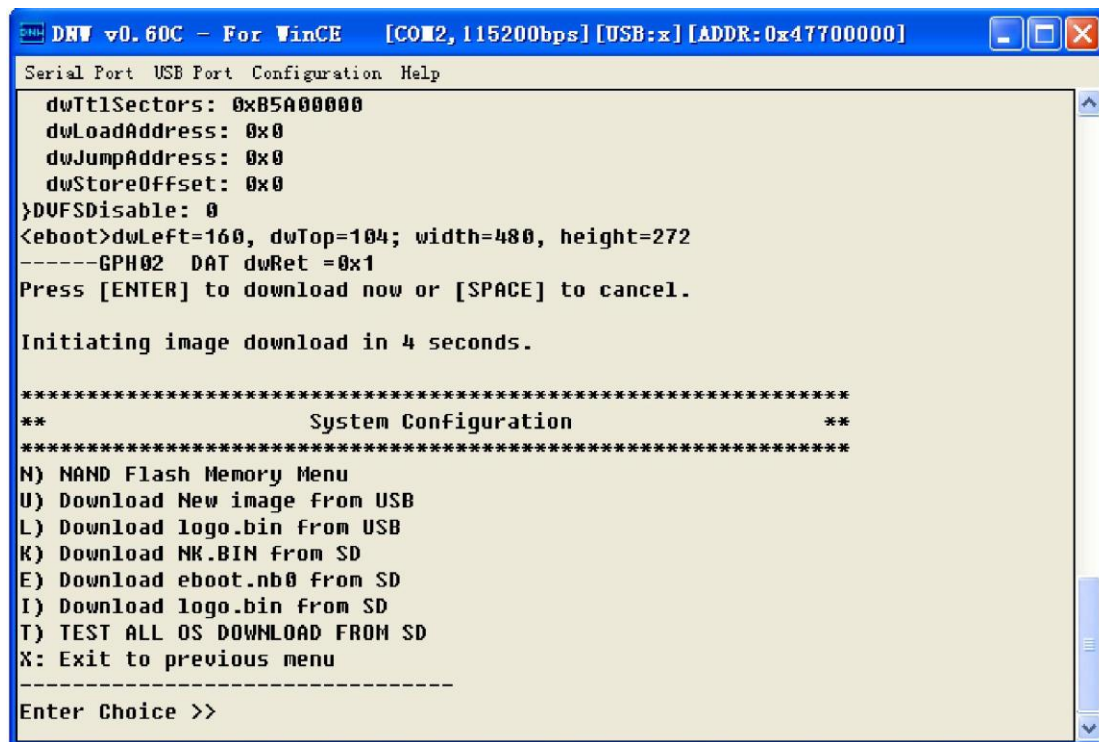
升级方法如下：

##### 1、插入 SD 卡

准备好有 LOGO.BIN 等镜像文件的 SD 卡，放到 SD 卡槽 2 中。

##### 2、进入 EBOOT 菜单

开机后按空格，进入 EBOOT 功能菜单，如下图：



```
*****
** System Configuration **
*****

N) NAND Flash Memory Menu
U) Download New image from USB
L) Download logo.bin from USB
K) Download NK.BIN from SD
E) Download eboot.nb0 from SD
I) Download logo.bin from SD
T) TEST ALL OS DOWNLOAD FROM SD
X: Exit to previous menu
-----
```



Enter Choice >>

红色字体部分的功能菜单是从 SD 卡读取镜像文件。需提前准备好要下载的镜像文件到 SD 卡中，并插入开发板卡槽 2。

### 3、选择从 SD 卡升级 LOGO 文件

如果从 SD 中升级 LOGO 则输入 I，即可自动完成 LOGO 升级。升级完后，需要重启。

```

DNW v0.60C - For WinCE [COM2, 115200bps] [USB:x] [ADDR: 0x47700000]
Serial Port USB Port Configuration Help
L) Download logo.bin from USB
K) Download NK.BIN from SD
E) Download eboot.nb0 from SD
I) Download logo.bin from SD
T) TEST ALL OS DOWNLOAD FROM SD
X: Exit to previous menu
-----
Enter Choice >> i

System ready!
Preparing for download...
Please send the LOGO.BIN through sd.
LOGO bin image
ERROR: Checksum failure (expected=0x159843 computed=0x1E68743)
[VerifyChecksum] LOGO.BIN...
OEMLaunch!
+OEMLaunch. dwLaunchAddr = 0x3fc08
g_ImageType = IMAGE_TYPE_LOGO
dwImageLength = 0x3fc08
dwNumBlocks = 0x2
WriteData: succes to write block (0x6).
WriteData: succes to write block (0x7).
INFO: logo.bin image stored to Smart Media. Please Reboot. Halting...
    
```

其他镜像从 SD 卡升级的方法与 LOGO 升级一样，这里不再重复。

## 1.14 开发板 uboot 启动参数

x210ii,x210v3 开发板有 nand flash, emmc 两种不同配置，同时有多种操作系统可供选择，他们的启动参数并不完全相同，读者切换操作系统后，有必要有动设置启动参数。详细列表如下：

x210 开发平台		
nand flash 与 inand 各操作系统启动参数设置		
nand	android2.3	setenv bootargs "root=/dev/mtdblock4 rootfstype=yafs2 init=/init console=ttySAC0,115200" setenv bootcmd "nand read C0008000 600000 400000;bootm C0008000"
	android4.0	setenv bootcmd "nand read C0008000 600000 400000;bootm C0008000"
	QT4.8	setenv bootargs "root=/dev/mtdblock4 rw init=/linuxrc rootfstype=jffs2 console=ttySAC2,115200" setenv bootcmd "nand read C0008000 600000 500000; bootm C0008000"
	QTOPIA	同 QT4.8
inand	android2.3	setenv bootcmd "movi read kernel 30008000;bootm 30008000"
	android4.0	setenv bootcmd "movi read kernel 30008000;bootm 30008000"
	QT4.8	setenv bootcmd "movi read kernel 30008000;bootm 30008000" setenv bootargs "console=ttySAC2,115200 root=/dev/mmcblk0p2 rw init=/linuxrc rootfstype=ext3"
	QTOPIA	同 QT4.8





### 1.15 nand flash 和 emmc 启动速度对照表

x210ii 开发平台				
nand flash 与 inand 各操作系统启动速度对照表				
	android2.3	android4.0	QT4.8	QTOPIA
nand	50 秒	74 秒	70 秒	30 秒
inand	18 秒	23 秒	11 秒	7 秒

### 1.16 WINCE 系统如何切换成 android 系统

烧有 WINCE 系统的开发板，如需切换成 android 或 linux 系统，首先需将拨码开关的 OM5 拨到 1，利用 DNW 通过 USB 下载相应的 uboot 到 emmc，之后再通过 SD 卡或是 DNW，fastboot 之类的工具下载相应的映像。

### 1.17 WINCE 系统如何切换成 linux 系统

同上。

### 1.18 android 系统如何切换成 WINCE 系统

烧有 android 或是 linux 系统的开发板，如需切换成 WINCE 系统，有两种方法。

方法一：将拨码开关的 OM5 拨到 1，利用 DNW 通过 USB 下载相应的 eboot 到 emmc，再通过 DNW 或是 SD 卡更新 NK。

方法二：破坏 emmc 的最前面的块，阻止 uboot 启动，再通过 WINCE 的启动卡从 SD 通道 2 启动，并通过 SD 卡升级。

如果您不会制作 WINCE 启动卡，请使用方法一。

### 1.19 linux 系统如何切换成 android 系统

第一步：更换对应的 uboot

第二步：使用最新的 uboot 执行如下指令重新分区：

```
fdisk -c 0
```

第三步：使用最新的 uboot 通过 SD 卡或是 DNW 更新最新的映像文件

第四步：如果更换系统时出现卡死在解压内核的位置，请修改 uboot 启动参数，如果没有出现，可忽略此步。

### 1.20 QT4.8 系统如何切换成 QTOPIA 系统

只需使用 SD 卡或 fastboot 更新文件系统即可。

### 1.21 QTOPIA 系统如何切换成 QT4.8 系统

同上。

### 1.22 linux 或 android 平台使用 SD 卡烧写方式调试程序

前面章节介绍了 linux 或是 android 平台通过 SD 卡一键更新映像的方法，使用起来确实简单明了，整个升级过程一气呵成，非常适合批量生产或是不想研究底层系统的朋友。但是对于调试底层驱动，完全没有必要为更新一个映像而将 bootloader,内核，文件系统一起更新。不用怕，SD 卡也可以使用指令调试。

使用 SD 卡通过 sdfuse 指令烧写指定映像，需要借助于调试串口。确保不要按下 LEFT



菜单，直接启动 uboot，进入命令行，直接在 uboot 命令行下输入相关指令即可。注意，这里输入命令行和 fastboot 有差异，fastboot 需要通过 cmd 窗口输入，sdfuse 是直接在 uboot 命令行输入，请读者不要混淆。

烧写相关映像时，首先需准备一张格式化为 FAT32 的 SD 卡，在 SD 卡根目录下建一个 x210 的目录，并将需要更新的映像文件存放在该目录，然后将 SD 卡插到开发板的 SD 卡槽 2。

```
sdfuse flash bootloader uboot.bin
```

烧写 linux 或 android 的 bootloader，需使用各自对应的 uboot

```
sdfuse flash kernel zImage-android
```

烧写 android4.0 的内核，需使用 android4.0 对应的 uboot

```
sdfuse flash kernel zImage
```

烧写 android2.3 的内核

```
sdfuse flash system x210.img
```

烧写 android 文件系统

```
sdfuse flash system rootfs_qt4.ext3
```

烧写 linux 文件系统，如果需烧写 QTOPIA 文件系统，请将文件系统名重命名为 rootfs\_qt4.ext3

```
sdfuse info
```

显示 sdfuse 相关信息

```
sdfuse flashall
```

烧写 linux 或 android 的 bootloader,内核和文件系统

```
sdfuse erase userdata
```

擦除 userdata 分区

```
sdfuse erase cache
```

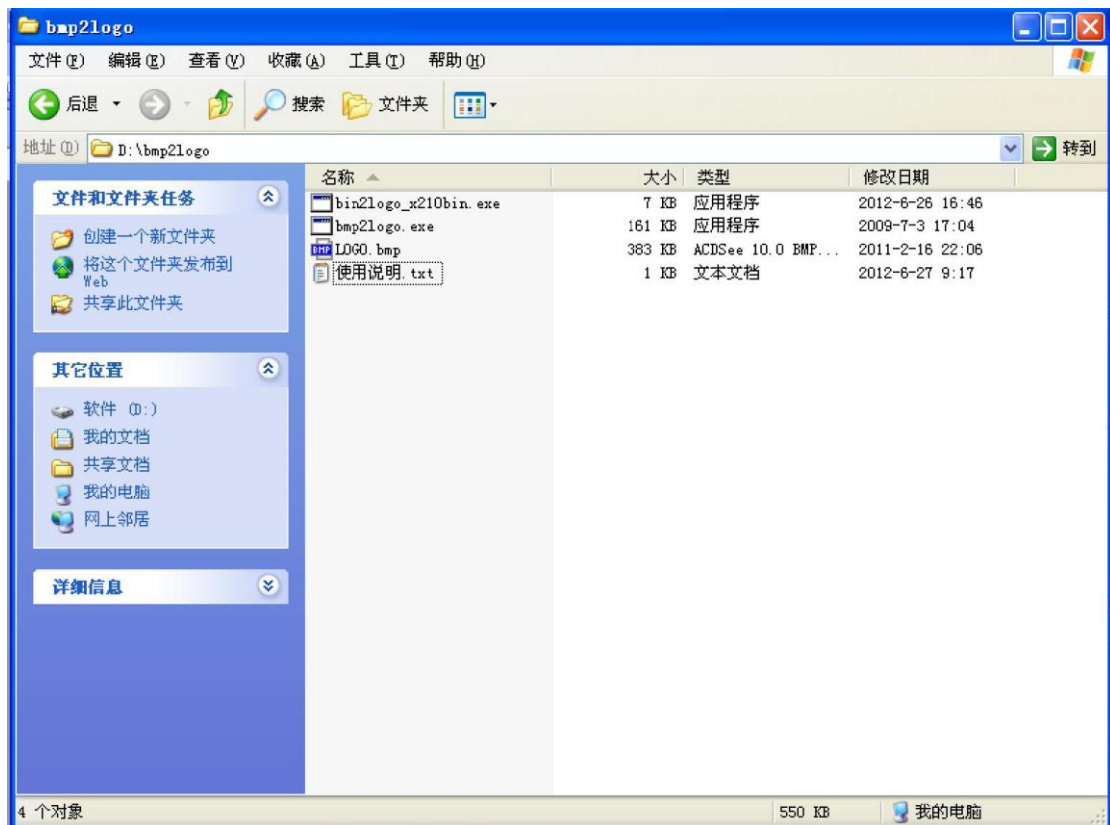
擦除 cache 分区

## 1.23 WINCE 平台开机 LOGO 制作

用户会根据自己的需求更改开机画面。也就是这里所说的开机 LOGO 文件。现提供 x210 开发板支持的 LOGO 文件制作方法如下：

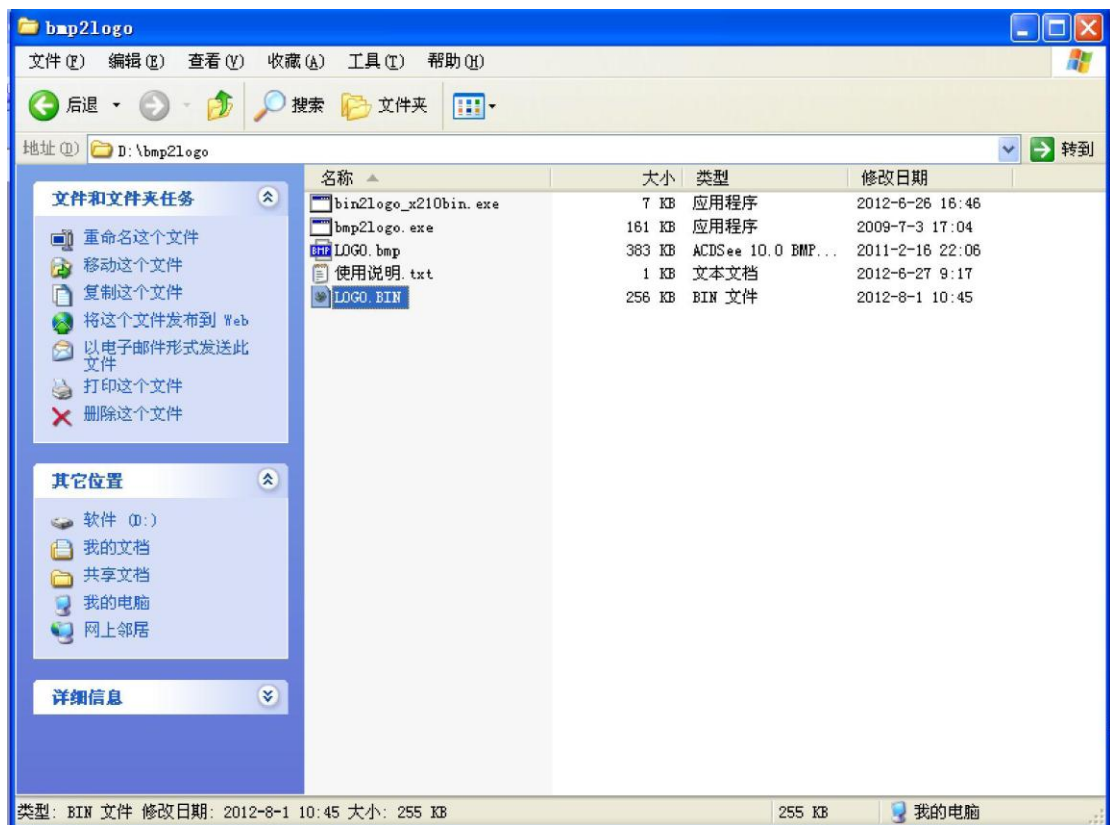
### 1、工具和原始 LOGO 文件放到同一文件夹

把 LOGO 制作工具和 LOGO.bmp 源文件放到同一文件夹中，如下图所示：



## 2、运行bmp2logo.exe

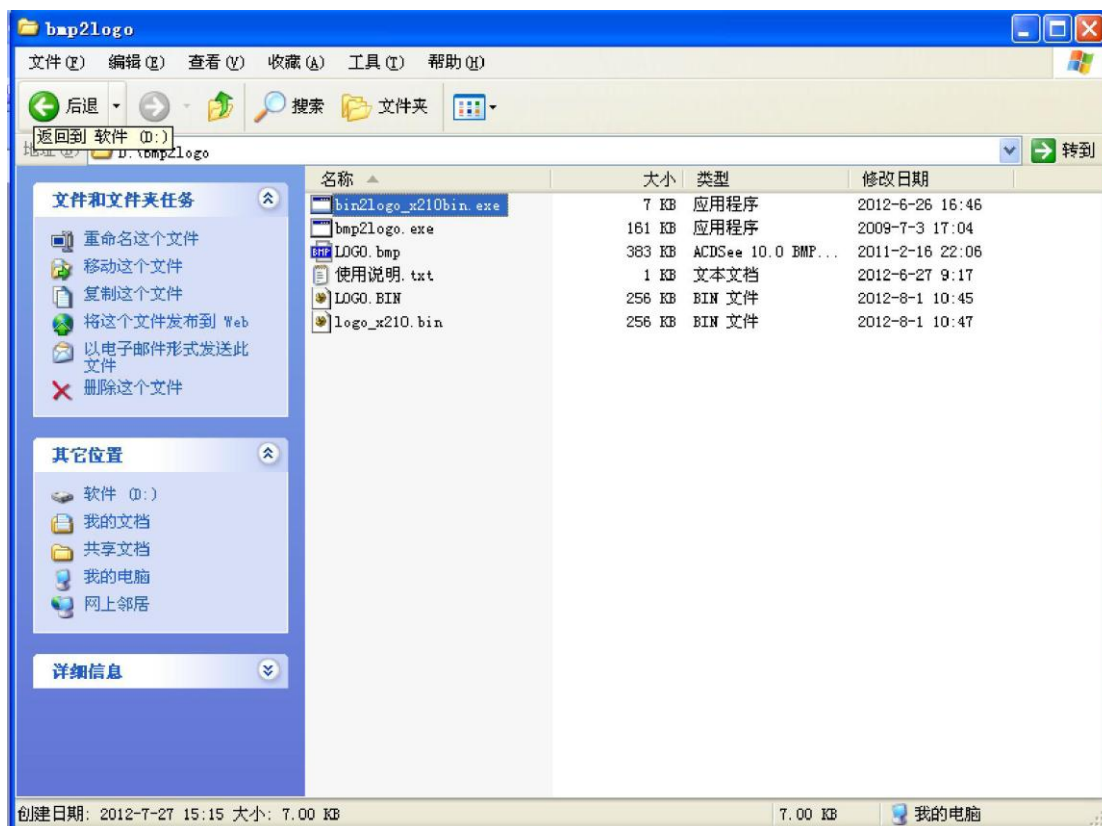
双击运行 bmp2logo.exe，在当前文件夹中会生成 LOGO.BIN 文件，如下图



## 3、运行bin2logo\_x210bin.exe



当前目录生成logo\_x210.bin。这才是X210 开发板所支持的LOGO 升级文件。如下图所示：



注意，将该文件拷贝到SD卡中通过SD卡升级时，需要将logo\_x210.bin 重命名为logo.bin。

## 1.24 WINCE 平台休眠唤醒

注意，WINCE 平台下休眠唤醒的方式和 linux 或 android 有点不一样，对应源码包已经开放，读者可以根据自己的需求调整源码包。休眠唤醒功能支持电阻屏套餐和电容屏套餐，如电容屏套餐无法唤醒，请使用最新的 touch.dll 文件。

休眠：进入 WINCE 主界面后，点击左下脚的开始按钮，选择挂起即可。

唤醒：开发板的左键或下键均可唤醒 WINCE 系统。

## 1.25 WINCE 平台如何校屏

进入 WINCE 主界面后，使用 USB 鼠标点击我的设备，控制面板，双击笔针，点击校准，然后点击再校准，会出现校准界面。电阻屏套餐使用笔针或指尖依次点击十字架，直至完成校准。电容屏套餐用手指的肉体点击十字架，切勿使用指甲点击。



## 第2章 其他产品介绍

### 2.1 核心板系列

#### 2.1.1 6410 核心板

X6410CV10

#### 2.1.2 210 核心板

X210CV10

X210CV3

G210CV10

I210CV10

### 2.2 开发板系列

#### 2.2.1 6410 开发板

x6410 开发板

#### 2.2.2 210 开发板

x210v3 开发板

g210 开发板

i210 开发板

说明：产品详细规格，以及更多其他产品请关注九鼎创展官方网站和论坛。