

四 iTOP-4412 开发板镜像的烧写

镜像，是一种文件形式，可以把许多文件做成一个镜像文件。说到底，镜像就是源代码编译并连接以后生成的可执行文件包，把这些镜像文件烧写到开发板的存储芯片里，开机就可以运行了。

烧写方式有两种，通过 TF 卡烧写以及使用 OTG 接口烧写。

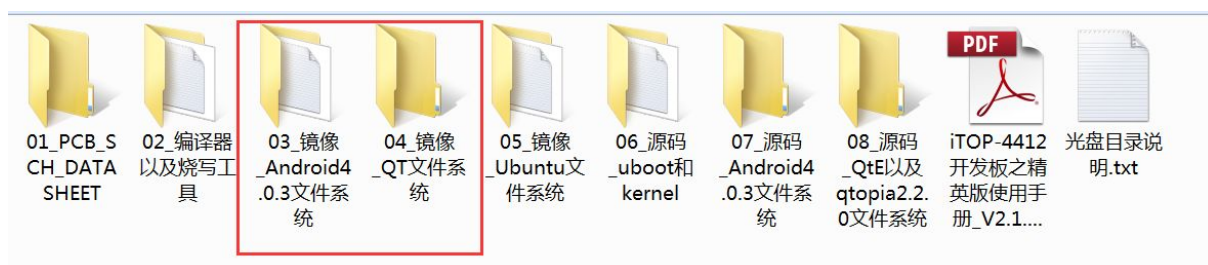
注意 1：Android 系统和 Qt 系统的烧写完全一样，是通用的。

注意 2：无论是使用 OTG 接口方式还是使用 TF 卡方式都不需要设置拨码开关，只有在出现特殊情况时才需要调整拨码开关，在本章最后一小节会提到什么情况需要重新设置拨码开关。

4.1 镜像文件说明

迅为 iTOP-4412 开发板平台支持的功能较多，镜像源码也很多，所以这一小节中先给大家介绍一下镜像文件在光盘中的位置。

开发板配套光盘，如下图所示，可以看到“03_镜像_Android4.0.3 文件系统”和“04_镜像_QT 文件系统”文件夹。Android4.0.3 和 Qt 操作系统和开发板配套的镜像都在这两个文件夹中。



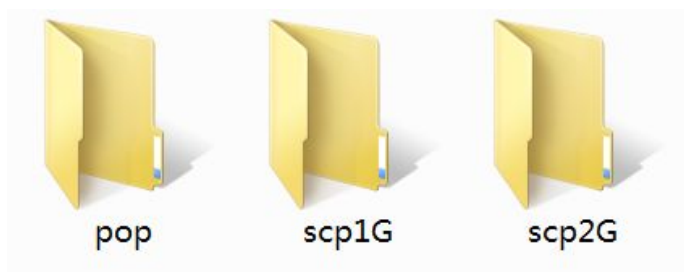
4.1.1 Android4.0.3 和三种核心板配套的镜像

Android4.0.3 系统需要的四个的镜像都在光盘的目录 “03_镜像_Android4.0.3 文件系统” 下，如下图所示，该目录分为 “uboot” ， “zImage” 和 “system” 三个文件夹。

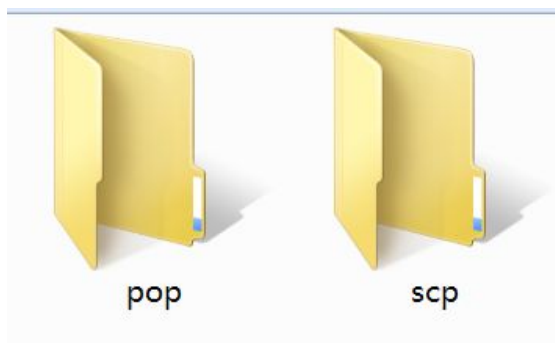


先根据手册 1.1.1 小节确定自己开发板的核心板的是 SCP 1G，SCP 2G 还是 POP。

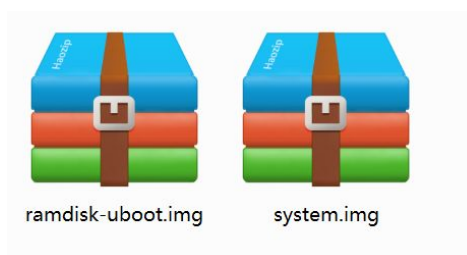
uboot 镜像文件 “u-boot-iTOP-4412.bin” ，在光盘 “03_镜像_Android4.0.3 文件系统” → “uboot” 目录下，如下图所示，分为 “pop” ， “scp1G” ， “scp2G” ，用户根据实际情况选用和核心板对应的 “u-boot-iTOP-4412.bin” 文件。



kernel 镜像文件 “zImage” ，在光盘 “03_镜像_Android4.0.3 文件系统” → “zImage” 目录下，如下图所示，分为 “pop” 和 “scp” 文件夹。SCP 1G 和 SCP 2G 核心板通用 SCP 目录下的 “zImage” 镜像文件，POP 核心板使用 “pop” 目录下的 “zImage” 镜像文件。



Android 镜像文件 “ramdisk-uboot.img” 和 “system.img” ，在光盘 “03_镜像_Android4.0.3 文件系统” → “system” 目录下，如下图所示，全部核心板通用一套镜像文件。



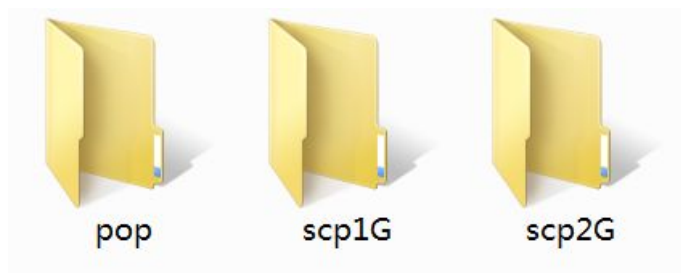
4.1.2 Qt 和三种核心板配套的镜像

Qt 系统需要的四个的镜像都在光盘的目录 “04_镜像_QT 文件系统” 下，如下图所示，该目录分为 “uboot” ， “zImage” 和 “system” 三个文件夹。

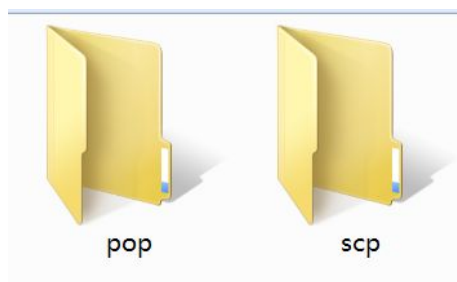


先根据手册 1.1.1 小节确定自己开发板的核心板的是 **SCP 1G**，**SCP 2G** 还是 **POP**。

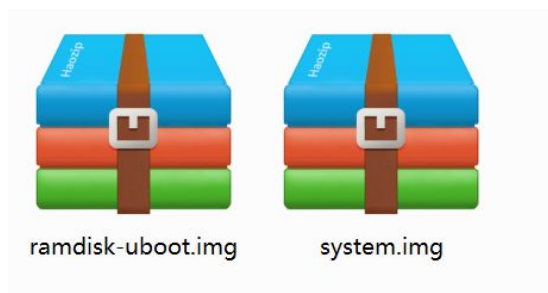
uboot 镜像文件 “u-boot-iTOP-4412.bin” ，在光盘 “04_镜像_QT 文件系统” → “uboot” 目录下，如下图所示，分为 “pop” ， “scp1G” ， “scp2G” ，用户根据实际情况选用和核心板对应的 “u-boot-iTOP-4412.bin” 文件。



kernel 镜像文件 “zImage” ，在光盘 “04_镜像_QT 文件系统” → “zImage” 目录下，如下图所示，分为 “pop” 和 “scp” 文件夹。SCP 1G 和 SCP 2G 核心板通用 SCP 目录下的 “zImage” 镜像文件，POP 核心板使用 “pop” 目录下的 “zImage” 镜像文件。



Qt 镜像文件 “ramdisk-uboot.img” 和 “system.img” ，在光盘 “04_镜像_QT 文件系统” → “system” 目录下，如下图所示，全部核心板通用一套镜像文件。



4.2 OTG 接口烧写方式

通过该方式可以烧写 Android 系统和 Qt 系统。

需要准备一根 OTG 线，绝大多数智能手机和 PC 机相连接的线都是 OTG 线，都是通用的。

这种方式比 TF 卡烧写方式要快一些，建议调试的时候使用这种方法。

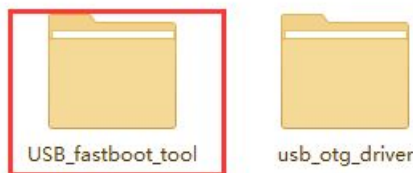
OTG 接口烧写方式也叫 fastboot 烧写方式，先介绍 OTG 烧写使用的硬件和软件平台以及烧写步骤，再介绍烧写对应的操作。

4.2.1 硬件平台

- 1) 使用串口线连接开发板串口（精英版是靠近网口的串口 CON3）到 PC 机串口
- 2) 使用 OTG 线，将开发板 OTG 接口和 PC 的 USB 接口相连。
- 3) 连接电源，屏幕等。

4.2.2 软件平台

- 1) OTG 方式只能在 XP、WIN7 或者 WIN8 系统下实现。注意，如果用户不是在 Win7 环境下，需要参考 3.6.2 中的“cmd.exe”，使用用户系统自带的终端。
- 2) 打开串口工具（超级终端或者其它串口助手），设置默认波特率为 115200，参考“3.1 超级终端的使用”。
- 3) 将光盘中的文件夹“02_编译器以及烧写工具”→“tools”→“USB_fastboot_tool”文件夹拷贝到您方便使用的地方，因为这个文件夹会经常被用到，注意不要拷贝到中文目录下（桌面不算是中文目录），如下图所示。

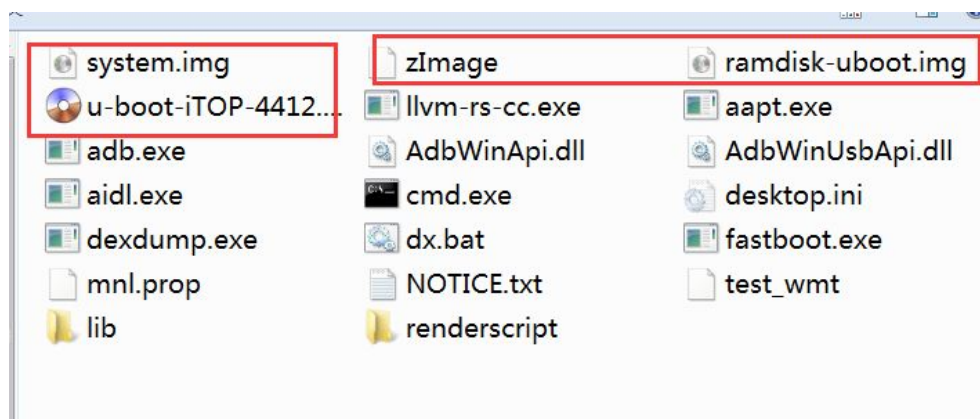


- 4) 安装 PC 机的 USB 驱动。

用户光盘"02_编译器以及烧写工具\tools\usb_otg_driver"文件夹中有 USB 驱动，这个 USB 驱动和 Android 的 ADB 驱动通用，可以参考“3.6 安卓 ADB 功能介绍”。提供 xp 和 win7 版本的 USB 驱动。

4.2.3 烧写步骤

1) 将 Android4.0.3 文件系统和核心板配套四个镜像“ramdisk-uboot.img”、“system.img”、“u-boot-iTOP-4412.bin”、“zImage”拷贝到“USB_fastboot_tool”-->“platform-tools”文件夹下面。（注意拷贝前一定要先确认自己核心板是属于哪一类，然后在对应的文件夹中拷贝出来）



1) 打开超级终端，然后上电启动开发板，按“回车”，进入 Uboot 模式，不明白 uboot 模式可以参考“2.3 uboot 模式和文件系统模式”。如下图所示，进入 uboot 模式。

```
In:      serial
Out:     serial
Err:     serial
eMMC OPEN Success.!!

!!!Notice!!!
!You must close eMMC boot Partition after all image writing!
!eMMC boot partition has continuity at image writing time.!
!So, Do not close boot partition, Before, all images is written.!

MMC read: dev # 0, block # 48, count 16 ...16 blocks read: OK
eMMC CLOSE Success.!!

Checking Boot Mode ... EMMC4.41
SYSTEM ENTER NORMAL BOOT MODE
Hit any key to stop autoboot:  0
iTOP-4412 #
```


2) 创建 eMMC 分区并格式化。如果原来已经做过此步骤，则可以跳过，不必每次烧写前都分区和格式化。在超级终端中，输入下面分区和格式化命令。

如下图所示，输入分区命令 “fdisk -c 0”。

```
iTOP-4412 # fdisk -c 0
.fdisk is completed

partition #    size(MB)    block start #    block count    partition_Id
1             1340           4862616          2744544         0x0C
2             1026           37290            2103156         0x83
3             1026          2140446          2103156         0x83
4             302           4243602          619014          0x83
iTOP-4412 #
```

如下图所示，输入命令 “fatformat mmc 0:1”。

```
iTOP-4412 # fatformat mmc 0:1
Start format MMC&d partition&d ...
Partition1: Start Address(0x4a3298), Size(0x29e0e0)
.....size checking ...
Under 8G
write FAT info: 32
Fat size : 0xa78
..Erase FAT region.....
.....
.....
..Partition1 format complete.
iTOP-4412 #
```

如下图所示，输入命令 “ext3format mmc 0:2”。

```
iTOP-4412 # ext3format mmc 0:2
Start format MMC0 partition2 ....
** Partition2 is not ext2 file-system 0 **
Partition2: Start Address(0x91aa), Size(0x201774)
Start ext2format...
Wirte 0/9block-group
Reserved blocks for jounaling : 8202
Start write addr : 0x91aa
.....
.....d_indirect_point:0xc2aa
..Wirte 1/9block-group
Reserved blocks for jounaling : 8202
Start write addr : 0x491aa
....Erase inode table(1) - 0x493ca.....
Wirte 2/9block-group
Reserved blocks for jounaling : 8202
Start write addr : 0x891aa
..Erase inode table(2) - 0x891ba.....
Wirte 3/9block-group
Reserved blocks for jounaling : 8202
```

如下图所示，输入命令 “ext3format mmc 0:3”。

```
iTOP-4412 # ext3format mmc 0:3
Start format MMC0 partition3 ....
** Partition3 is not ext2 file-system 0 **
Partition3: Start Address(0x20a91e), Size(0x201774)
Start ext2format...
Wirte 0/9block-group
Reserved blocks for jounaling : 8202
Start write addr : 0x20a91e
.....
.....d_indirect_point:0x20dale
..Wirte 1/9block-group
Reserved blocks for jounaling : 8202
Start write addr : 0x24a91e
....Erase inode table(1) - 0x24ab3e.....
Wirte 2/9block-group
Reserved blocks for jounaling : 8202
Start write addr : 0x28a91e
..Erase inode table(2) - 0x28a92e.....
Wirte 3/9block-group
Reserved blocks for jounaling : 8202
```

如下图所示，输入命令 “ext3format mmc 0:4” 。

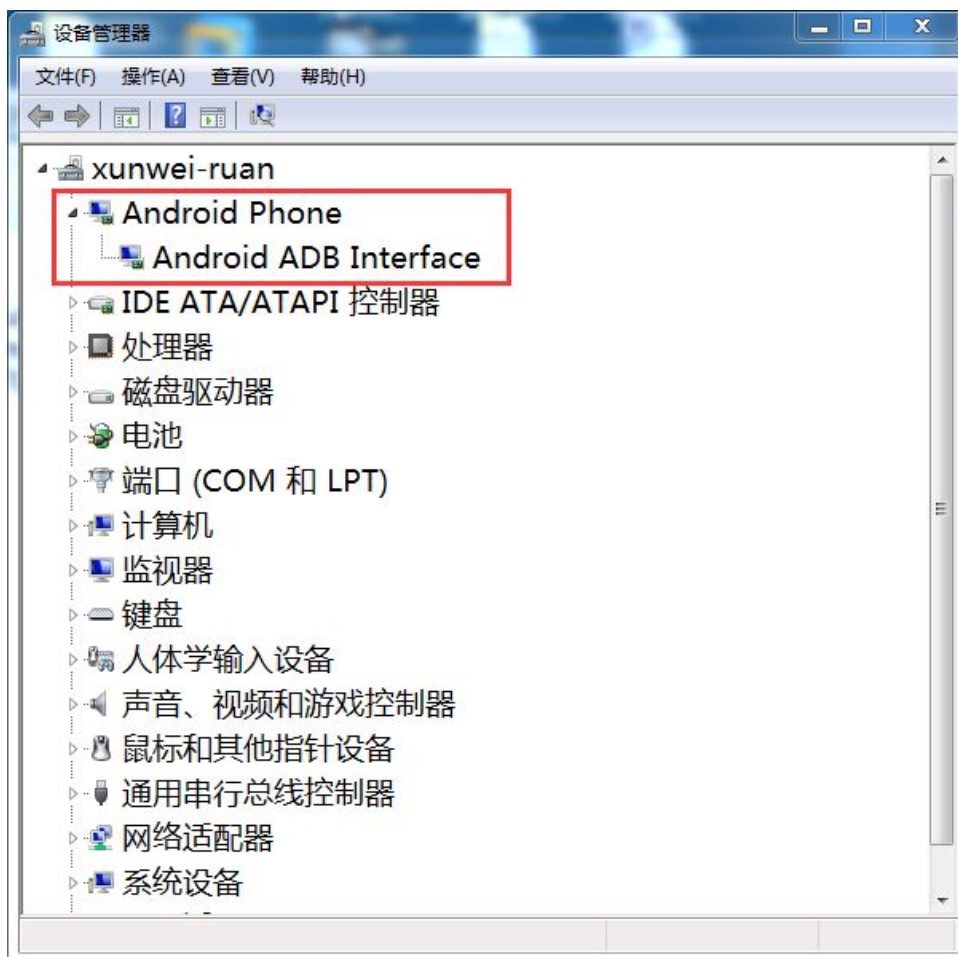
```
iTOP-4412 # ext3format mmc 0:4
Start format MMC0 partition4 ....
** Partition4 is not ext2 file-system 0 **
Partition4: Start Address(0x40c092), Size(0x97206)
Start ext2format...
Wirte 0/3block-group
Reserved blocks for jounaling : 4102
Start write addr : 0x40c092
.....Erase inode table(0) - 0x40c142.....
.....d_indirect_point:0x40fb12
..Wirte 1/3block-group
Reserved blocks for jounaling : 4102
Start write addr : 0x44c092
....Erase inode table(1) - 0x44c142.....
Wirte 2/3block-group
Reserved blocks for jounaling : 4102
Start write addr : 0x48c092
..Erase inode table(2) - 0x48c0a2.....
iTOP-4412 #
```

如下图所示，在超级终端中，输入命令 “fastboot”

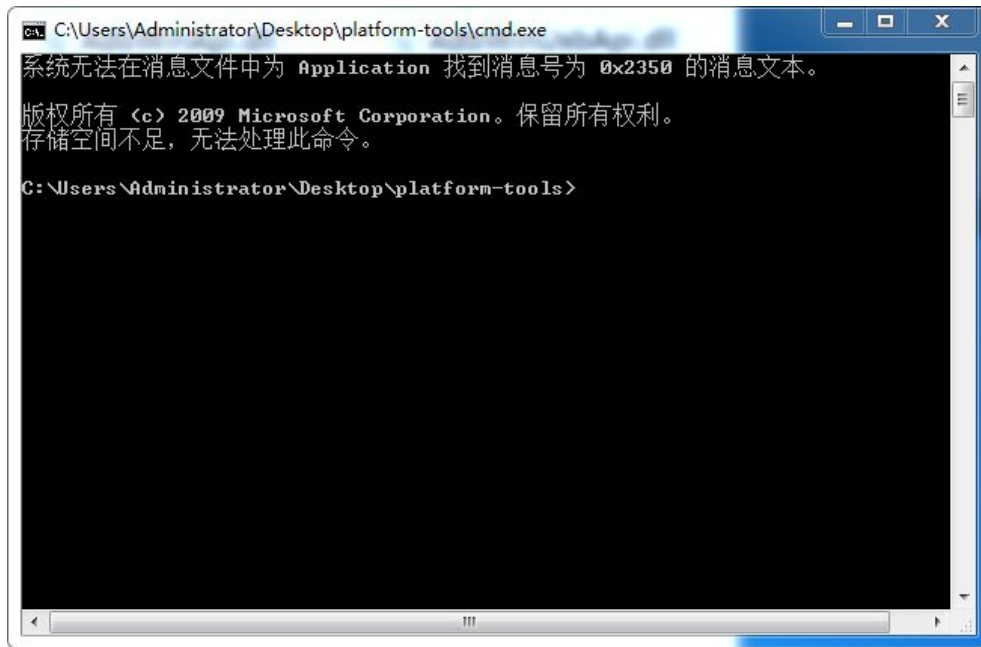
```
iTOP-4412 # fastboot
[Partition table on MoviNAND]
ptn 0 name='bootloader' start=0x0 len=N/A (use hard-coded info. (
ptn 1 name='kernel' start=N/A len=N/A (use hard-coded info. (cmd:
ptn 2 name='ramdisk' start=N/A len=0x300000(~3072KB) (use hard-co
ptn 3 name='Recovery' start=N/A len=0x600000(~6144KB) (use hard-c
ptn 4 name='system' start=0x1235400 len=0x402EE800(~1051578KB)
ptn 5 name='userdata' start=0x41523C00 len=0x402EE800(~1051578KB)
ptn 6 name='cache' start=0x81812400 len=0x12E40C00(~309507KB)
ptn 7 name='fat' start=0x94653000 len=0x53C1C000(~1372272KB)
```


注意，fastboot 命令需要与 PC 上的 USB_fastboot_tool 工具配套使用，而且 fastboot 命令需要进入 uboot 模式中才能使用。

3) 如下图所示，检查一下 PC 是否识别了设备，需要用 OTG 线将开发板 OTG 接口和 PC 的 USB 接口相连。



4) 在 PC 机上运行 “USB_fastboot_tool” --> “platform-tools” 文件夹中的文件 “cmd.exe” (cmd.exe 可执行文件是 Windows 自带的命令行工具，光盘里面的是 Win7 下的，如果提示版本不兼容，请使用你自己系统里面的 cmd.exe 工具，具体可以参考 3.6.2.1 cmd.exe 程序)，如下图所示。

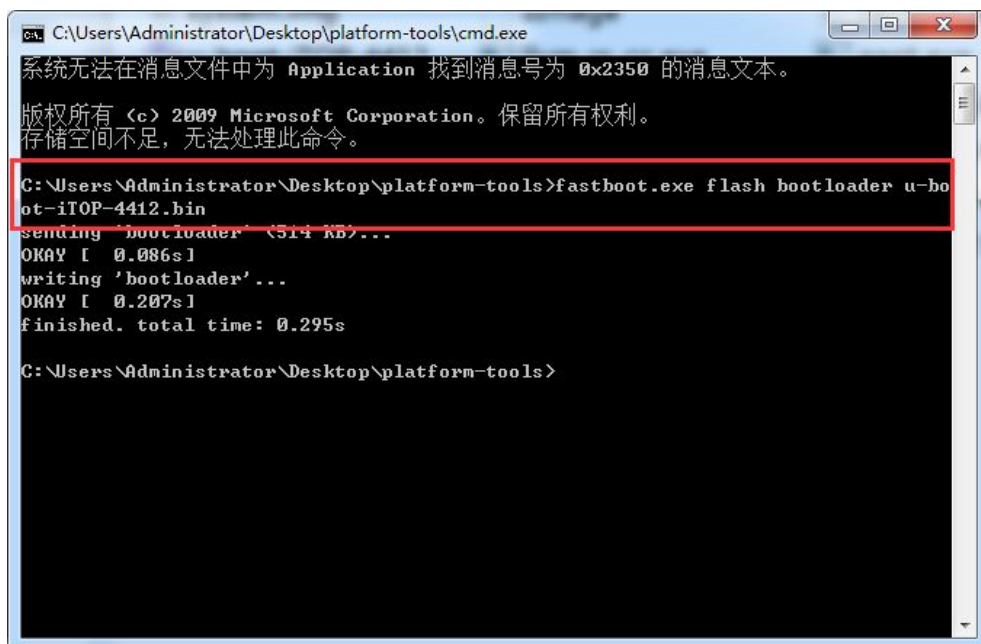


5) 在 Windows 命令行中，输入下面的命令：

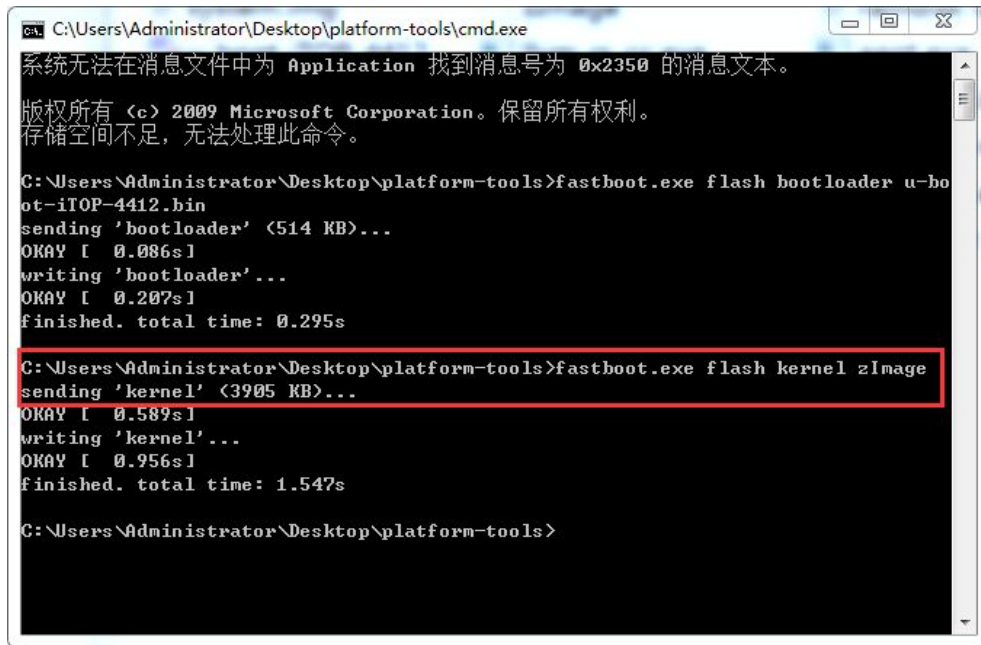
如下图所示，输入烧写 uboot 命令

`"fastboot.exe flash bootloader u-boot-iTOP-4412.bin"`

特别提醒，不建议用户烧写“u-boot-iTOP-4412.bin”这个文件，可跳过此步骤，因为出厂前已经烧写过这个镜像文件了。



如下图所示，输入烧写 zImage 内核命令 `"fastboot.exe flash kernel zImage"`。



```
C:\Users\Administrator\Desktop\platform-tools\cmd.exe
系统无法在消息文件中为 Application 找到消息号为 0x2350 的消息文本。
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。
存储空间不足，无法处理此命令。

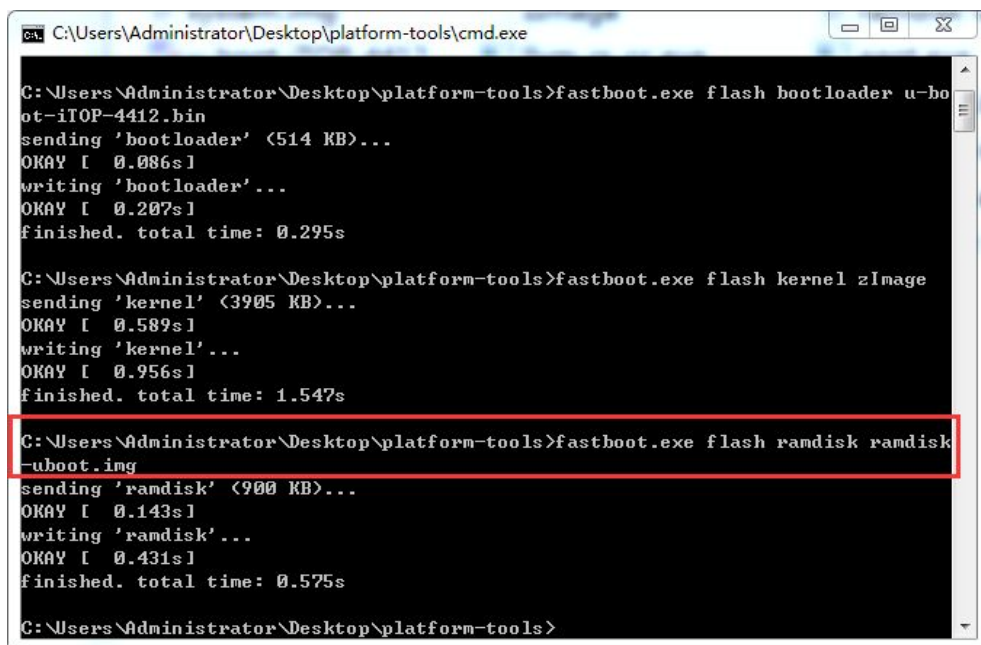
C:\Users\Administrator\Desktop\platform-tools>fastboot.exe flash bootloader u-bo
ot-iTOP-4412.bin
sending 'bootloader' (514 KB)...
OKAY [ 0.086s]
writing 'bootloader'...
OKAY [ 0.207s]
finished. total time: 0.295s

C:\Users\Administrator\Desktop\platform-tools>fastboot.exe flash kernel zImage
sending 'kernel' (3905 KB)...
OKAY [ 0.589s]
writing 'kernel'...
OKAY [ 0.956s]
finished. total time: 1.547s

C:\Users\Administrator\Desktop\platform-tools>
```

如下图所示，输入烧写 ramdisk 命令

“fastboot.exe flash ramdisk ramdisk-uboot.img”



```
C:\Users\Administrator\Desktop\platform-tools\cmd.exe

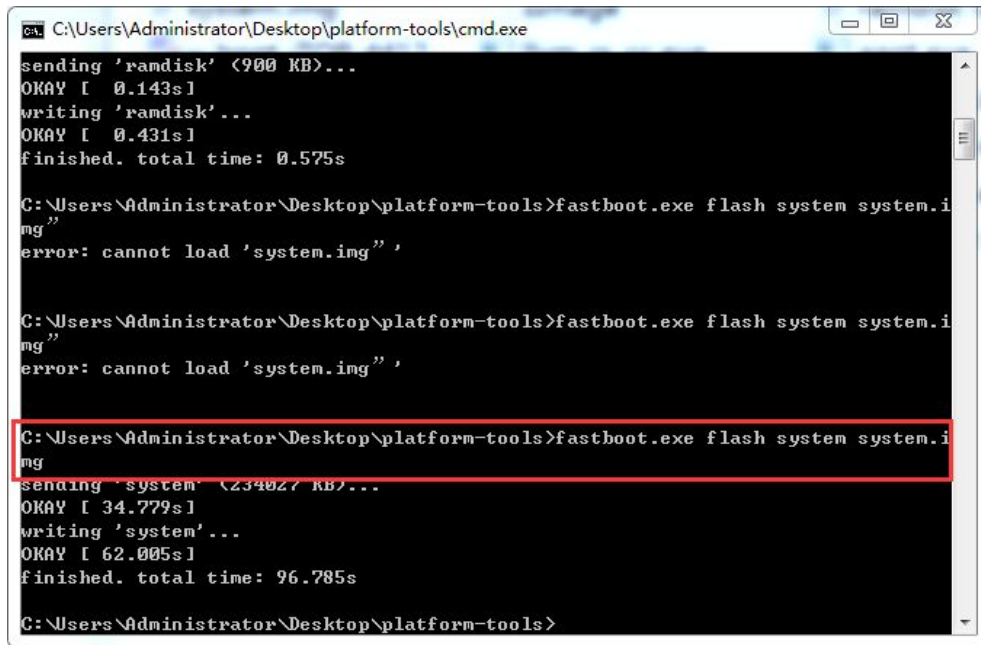
C:\Users\Administrator\Desktop\platform-tools>fastboot.exe flash bootloader u-bo
ot-iTOP-4412.bin
sending 'bootloader' (514 KB)...
OKAY [ 0.086s]
writing 'bootloader'...
OKAY [ 0.207s]
finished. total time: 0.295s

C:\Users\Administrator\Desktop\platform-tools>fastboot.exe flash kernel zImage
sending 'kernel' (3905 KB)...
OKAY [ 0.589s]
writing 'kernel'...
OKAY [ 0.956s]
finished. total time: 1.547s

C:\Users\Administrator\Desktop\platform-tools>fastboot.exe flash ramdisk randisk
-uboot.img
sending 'ramdisk' (900 KB)...
OKAY [ 0.143s]
writing 'ramdisk'...
OKAY [ 0.431s]
finished. total time: 0.575s

C:\Users\Administrator\Desktop\platform-tools>
```

如下图所示，输入烧写 system 文件系统命令 “fastboot.exe flash system system.img”



```
C:\Users\Administrator\Desktop\platform-tools\cmd.exe
sending 'ramdisk' (900 KB)...
OKAY [ 0.143s]
writing 'ramdisk'...
OKAY [ 0.431s]
finished. total time: 0.575s

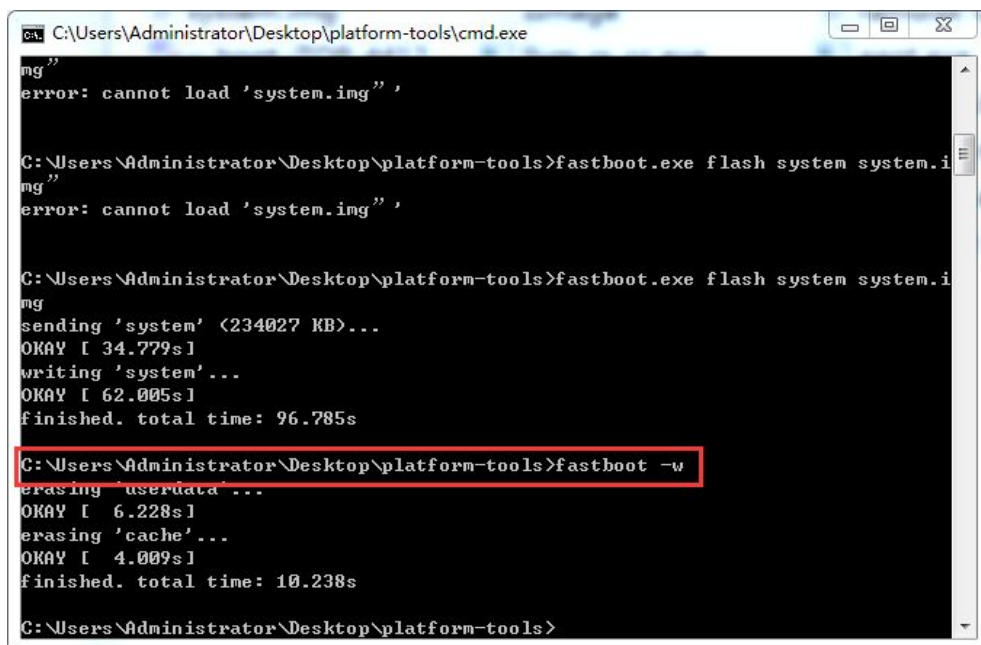
C:\Users\Administrator\Desktop\platform-tools>fastboot.exe flash system system.i
mg"
error: cannot load 'system.img' '

C:\Users\Administrator\Desktop\platform-tools>fastboot.exe flash system system.i
mg"
error: cannot load 'system.img' '

C:\Users\Administrator\Desktop\platform-tools>fastboot.exe flash system system.i
mg
sending 'system' (234027 KB)...
OKAY [ 34.779s]
writing 'system'...
OKAY [ 62.005s]
finished. total time: 96.785s

C:\Users\Administrator\Desktop\platform-tools>
```

如下图所示，输入擦除命令 “fastboot -w” 。



```
C:\Users\Administrator\Desktop\platform-tools\cmd.exe
mg"
error: cannot load 'system.img' '

C:\Users\Administrator\Desktop\platform-tools>fastboot.exe flash system system.i
mg"
error: cannot load 'system.img' '

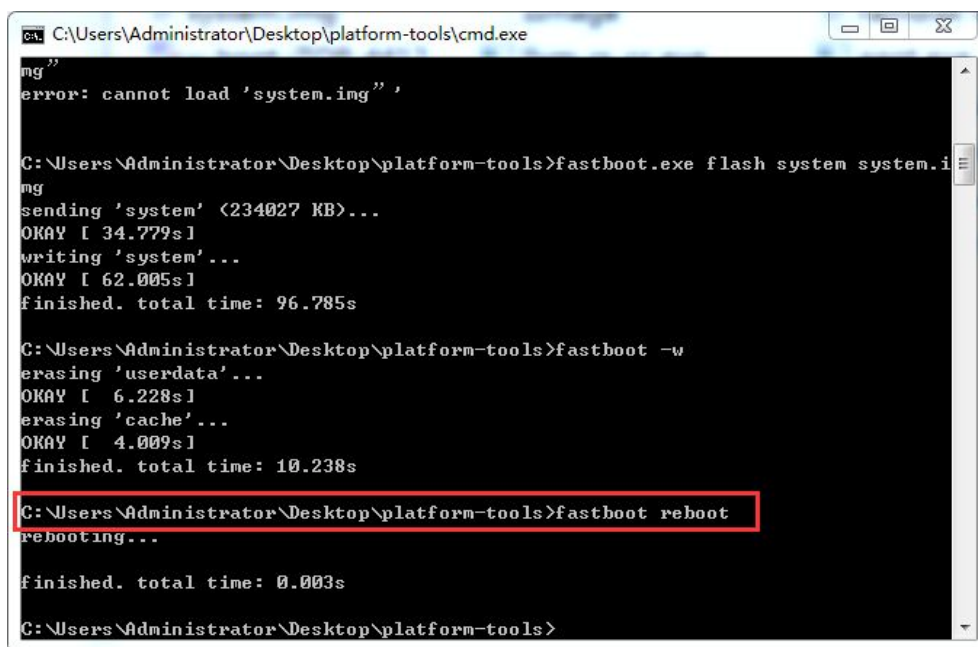
C:\Users\Administrator\Desktop\platform-tools>fastboot.exe flash system system.i
mg
sending 'system' (234027 KB)...
OKAY [ 34.779s]
writing 'system'...
OKAY [ 62.005s]
finished. total time: 96.785s

C:\Users\Administrator\Desktop\platform-tools>fastboot -w
erasing 'userdata'...
OKAY [ 6.228s]
erasing 'cache'...
OKAY [ 4.009s]
finished. total time: 10.238s

C:\Users\Administrator\Desktop\platform-tools>
```

注意：上述 fastboot.exe flash 命令可以分开执行，只烧写单个的镜像。

6) 在 Windows 命令行中，输入重启开发板命令 “fastboot reboot” 。



```
C:\Users\Administrator\Desktop\platform-tools\cmd.exe
mg''
error: cannot load 'system.img''

C:\Users\Administrator\Desktop\platform-tools>fastboot.exe flash system system.i
mg
sending 'system' (234027 KB)...
OKAY [ 34.779s]
writing 'system'...
OKAY [ 62.005s]
finished. total time: 96.785s

C:\Users\Administrator\Desktop\platform-tools>fastboot -w
erasing 'userdata'...
OKAY [ 6.228s]
erasing 'cache'...
OKAY [ 4.009s]
finished. total time: 10.238s

C:\Users\Administrator\Desktop\platform-tools>fastboot reboot
rebooting...

finished. total time: 0.003s

C:\Users\Administrator\Desktop\platform-tools>
```

输入重启命令之后，开发板会重启，超级终端会打印启动信息，第一次 Android 启动需要解压和安装一些初始化文件，会花费的时间长一点。第一次启动完成之后，再次启动速度就会快一些。

4.3 TF 卡烧写方式

使用该方式能够烧写 Android 系统和 QT 系统。

TF 卡存储容量最少要 2G 以上。

4.3.1 制作可以烧写的 TF 卡

注意：制作可以烧写的 TF 卡需要用到 Ubuntu 系统，用户需要学习 Ubuntu 的使用，Ubuntu 的安装和使用可以参考 3.2、3.3 以及 3.4 小节。另外使用这种方式要求核心板的 uboot 可以正常启动，如果核心板的 uboot 无法启动，请参考使用手册“4.4 开发板出厂前首次烧写”

本节制作 TF 卡的方法需要核心板的 eMMC 能够正常启动打印信息。

使用 TF 卡之前，必须要先分区。制作 TF 卡需要在 PC 机的 Ubuntu 系统下，分 3 个步骤来完成。这里需要注意的是，TF 卡制作完成后，就可长期使用，不用每次重新制作，另外如果烧坏了 eMMC 的 uboot，那么将无法进行分区，就要参考 4.4 小节先给 TF 卡分区。

1) 给 TF 卡分区。需要将 TF 卡先插入开发板，然后再启动开发板并进入 Uboot 模式（如何进入可参考前面 2-5 节），然后在超级终端中，依次输入下列烧写命令：

— `fdisk -c 1`

注意上面的分区命令的是参数“1”，代表的是 tf 卡，如果是“0”则代表是 eMMC。如果是 2G 的 TF 卡，请将命令“`fdisk -c 1`”改为“`fdisk -c 1 300 300 300`”。

— `fatformat mmc 1:1`

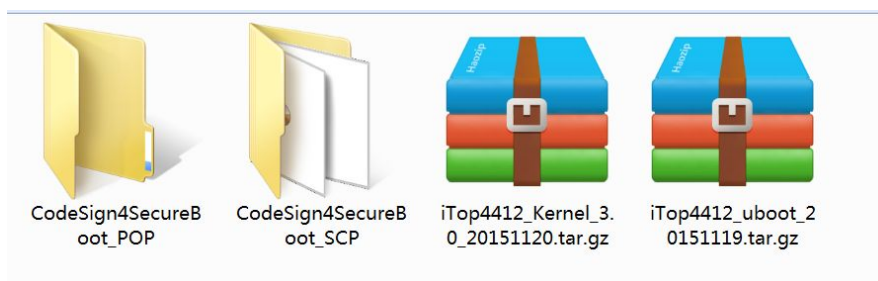
— `ext3format mmc 1:2`

— `ext3format mmc 1:3`

— `ext3format mmc 1:4`

执行完上面的命令之后，就要将 uboot 烧写到 TF 卡。

2) 如下图所示，在光盘文件夹“06_源码_uboot 和 kernel”中有一个压缩包“iTop4412_uboot_xxx.tar.gz”，压缩包文件名中的“xxx”代表不确定，“xxx”表示日期，日期信息在系统升级后会变更。



3) 使用 SSH 工具（参考使用手册 3.3.5 安装和使用 SSH 软件），拷贝压缩包“iTop4412_uboot_xxx.tar.gz”到 PC 机的 Ubuntu 系统中，然后解压压缩包，得到文件夹“iTop4412_uboot”，进入“iTop4412_uboot”文件夹，如下图所示。

```

root@ubuntu: /home/topeet/android4.0/iTop4412_uboot
root@ubuntu: /home/topeet/android4.0# ls
iTop4412_uboot  iTop4412_uboot_20151119.tar.gz
root@ubuntu: /home/topeet/android4.0# cd iTop4412_uboot
root@ubuntu: /home/topeet/android4.0/iTop4412_uboot# ls
all00_padding.bin  E4412_N.bl1.bin  lib_nios  onenand_ip1
api                examples         lib_nios2 paddingaa
board              fs               lib_ppc   post
build_uboot.sh     include          lib_sh    README
common             lib_arm          lib_sparc readme.txt
config.mk          lib_avr32        MAINTAINERS rules.mk
COPYING            lib_blackfin     MAKEALL   sdfuse
cpu                libbftd          Makefile  sdfuse
CREDITS            lib_generic      mkb12     tc4_cmm.cmm
disk               lib_i386         mkconfig  tools
doc                lib_m68k         mkuboot   uboot_readme.txt
drivers            lib_microblaze   nand_spl
E4212              lib_mips          net
root@ubuntu: /home/topeet/android4.0/iTop4412_uboot#

```

4) 拷贝光盘中文件夹中对应核心板的镜像 “u-boot-iTOP-4412.bin” 到上一步解压出来的文件夹 “iTop4412_uboot” 中，如下图所示。

```

root@ubuntu: /home/topeet/android4.0/iTop4412_uboot
common          lib_arm          lib_sparc      readme.txt
config.mk       lib_avr32        MAINTAINERS    rules.mk
COPYING         lib_blackfin     MAKEALL        sdfuse
cpu             libbftd          Makefile       sdfuse
CREDITS         lib_generic      mkb12          tc4_cmm.cmm
disk            lib_i386         mkconfig       tools
doc             lib_m68k         mkuboot        uboot_readme.txt
drivers         lib_microblaze   nand_spl
E4212           lib_mips          net
root@ubuntu: /home/topeet/android4.0/iTop4412_uboot# ls
all00_padding.bin  E4412_N.bl1.bin  lib_nios  onenand_ip1
api                examples         lib_nios2 paddingaa
board              fs               lib_ppc   post
build_uboot.sh     include          lib_sh    README
common             lib_arm          lib_sparc readme.txt
config.mk          lib_avr32        MAINTAINERS rules.mk
COPYING            lib_blackfin     MAKEALL   sdfuse
cpu                libbftd          Makefile  sdfuse
CREDITS            lib_generic      mkb12     tc4_cmm.cmm
disk               lib_i386         mkconfig  tools
doc                lib_m68k         mkuboot   uboot_readme.txt
drivers            lib_microblaze   nand_spl
E4212              lib_mips          net
u-boot-iTOP-4412.bin
root@ubuntu: /home/topeet/android4.0/iTop4412_uboot#

```

5) 在 Ubuntu 命令行中输入命令 “df -l” ，查看一下系统有哪些盘符。

```

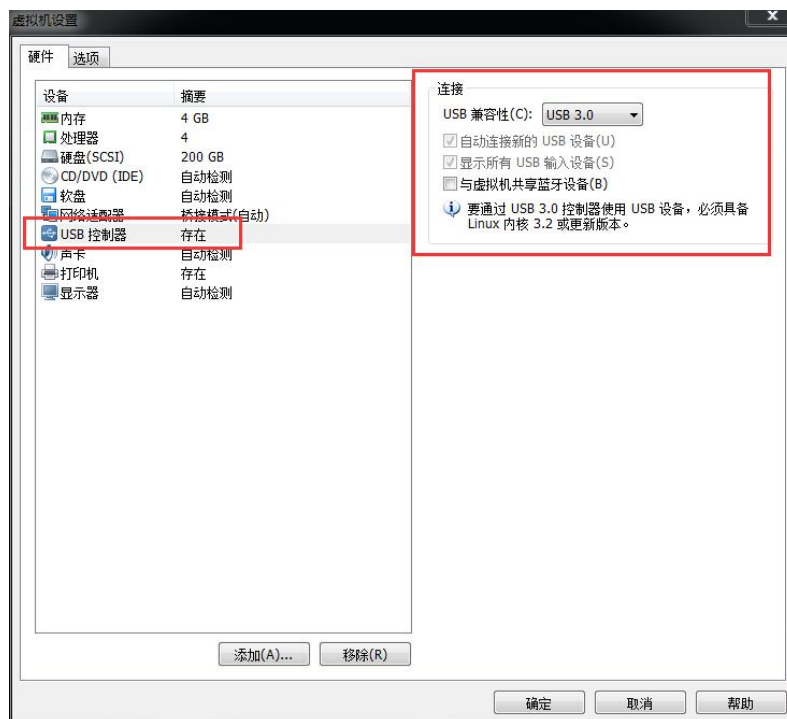
root@ubuntu: /home/topeet/android4.0/iTop4412_uboot# df -l
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda1       60894268 10242120  47558900  18% /
udev            2013744      4    2013740    1% /dev
tmpfs           809208       808    808400    1% /run
none            5120         0      5120      0% /run/lock
none            2023012     200    2022812    1% /run/shm
root@ubuntu: /home/topeet/android4.0/iTop4412_uboot#

```

6) 接着使用读卡器将 TF 卡连接到 PC 机的 Ubuntu 系统下，如下图所示，Ubuntu 系统识别 TF 卡后，可能提示下面的信息。



7) 在虚拟机 VMware Workstation 选项“虚拟机 M”，进入“虚拟机设置”，如下图所示，根据 USB 接口选择一下版本，如果是 USB3.0 则使用 USB3.0。



8) TF 卡连接之后到 Ubuntu 之后，再次使用 Linux 命令“df -l”查看盘符。将第二次查看的盘符和第一次查看的盘符对比一下，就会发现 Ubuntu 系统中多出了盘符，这个盘符就是 TF 卡的盘符，盘符名称在接下来的 Linux 命令中会用到，如下图所示。

```

root@ubuntu: /home/topeet/android4.0/iTop4412_uboot
root@ubuntu: /home/topeet/android4.0/iTop4412_uboot# df -l
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda1        60894268 10242120  47558900   18% /
udev             2013744      4    2013740    1% /dev
tmpfs            809208      808    808400    1% /run
none              5120        0     5120     0% /run/lock
none             2023012     200    2022812    1% /run/shm
root@ubuntu: /home/topeet/android4.0/iTop4412_uboot# df -l
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda1        60894268 10242424  47558596   18% /
udev             2013744      4    2013740    1% /dev
tmpfs            809208      832    808376    1% /run
none              5120        0     5120     0% /run/lock
none             2023012     200    2022812    1% /run/shm
/dev/sdb2        1034996     34112    948308    4% /media/37b2a37e-d68c-4259-7467-
a77513424650
/dev/sdb1         6053656      4    6053652    1% /media/0000-3333
/dev/sdb3         302612     16584    270408    6% /media/befaf008-2317-8fe3-fbaf-
677c60cccc98
/dev/sdb4         302612     16584    270408    6% /media/9ec90b96-3da4-a970-a2c0-
0e8d409bad67
root@ubuntu: /home/topeet/android4.0/iTop4412_uboot#

```

9) 在执行下面这条命令的时候，要特别特别注意！一定要分清楚，哪个盘符是 TF 卡的盘符，如果不清楚，请务必先拔掉 TF 卡，看清楚哪些盘符是属于 Ubuntu 系统的硬盘盘符，看清楚哪些盘符是 Ubuntu 系统的硬盘盘符后，再插入 TF 卡，分辨出哪个盘符是新增加的盘符，新增加的盘符才是 TF 卡的盘符。

10) 进入文件夹 “iTop4412_uboot” 中。在 Ubuntu 命令行中，执行 Linux 命令：

“./mkuboot /dev/sdx”，mkuboot 是 uboot 源码文件夹中的一个脚本，下图中脚本命令的对象是上图中识别的 “sdb”。

(注意，sdx 用前面查看盘符，多出来的盘符名代替，不要带数字，比如 df -l 看到的 tf 卡是 /dev/sdb0，这个 0 不要带，直接写 sdb)

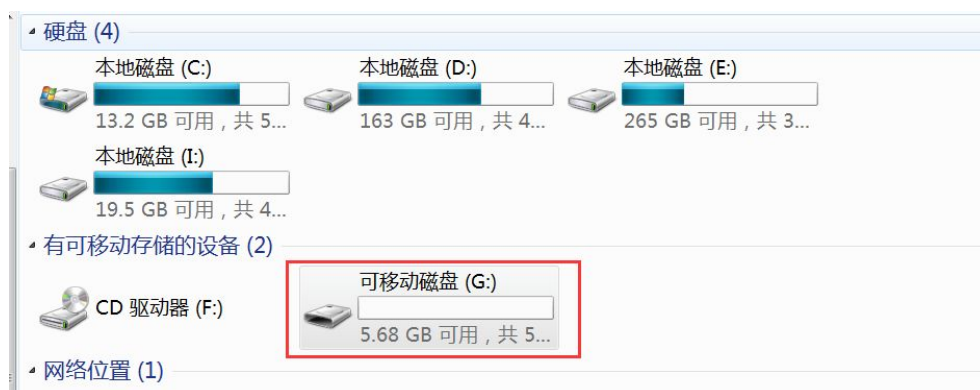
```

root@ubuntu: /home/topeet/android4.0/iTop4412_uboot# ls
all00_padding.bin  E4412_N.b11.bin  lib_nios  onenand_ip1
api                examples         lib_nios2  paddingaa
board              fs              lib_ppc    post
build_uboot.sh     include         lib_sh     README
common            lib_arm         lib_sparc  readme.txt
config.mk          lib_avr32       MAINTAINERS  rules.mk
COPYING           lib_blackfin    MAKEALL     sdfuse
cpu               libfdt         Makefile    sdfuse.g
CREDITS           lib_generic     mkbl2      tc4_cmm.cmm
disk              lib_i386        mkconfig   tools
doc               lib_m68k        mkuboot    u-boot-iTOP-4412.bin
drivers           lib_microblaze  nand_spl   uboot_readme.txt
E4212             lib_mips        net
root@ubuntu: /home/topeet/android4.0/iTop4412_uboot# ./mkuboot /dev/sdb
Fuse iTOP-4412 trustzone uboot file into SD card
/dev/sdb reader is identified.
u-boot-iTOP-4412.bin fusing...
1029+1 records in
1029+1 records out
527104 bytes (527 kB) copied, 5.47041 s, 96.4 kB/s
u-boot-iTOP-4412.bin image has been fused successfully.
Eject SD card
root@ubuntu: /home/topeet/android4.0/iTop4412_uboot#

```

11) 检查可以烧写的 TF 卡是否制作成功。

检查的方法很简单，使用读卡器连接 TF 卡到 Win7 系统上，如果在 Win7 系统上发现 TF 卡的存储空间减少了 2G 以上，那么这个 TF 卡就制作成功了。如下图所示，8G 的 TF 卡只剩 5.68G。



4.3.2 使用 TF 卡烧写

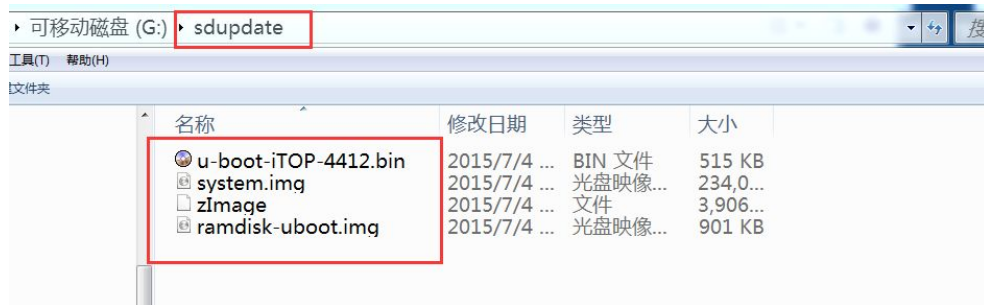
在 Win7 系统和 Ubuntu 系统下，都可以使用 TF 卡烧写。这里以 Win7 为例，Ubuntu 的也是一样，将镜像文件拷贝到 tf 卡中的“sdupdate”文件夹中。

Win7 下 TF 卡烧写步骤如下。

1) 将制作完成的 TF 卡接入 PC 机的 Win7 或者 Ubuntu 系统中，在 TF 卡上建立文件夹“sdupdate”。注意，文件夹名字一定要使用“sdupdate”。如下图所示。



2) 拷贝相应的镜像文件到 TF 卡的文件夹“sdupdate”中，如下图所示。



3) 将 TF 卡先插入开发板中，进入超级终端的 UBOOT 模式，如下图所示。

```

eMMC OPEN Success!!!

!!!Notice!!!
!You must close eMMC boot Partition after all image writing!
!eMMC boot partition has continuity at image writing time.!
!So, Do not close boot partition, Before, all images is written.!

MMC read: dev # 0, block # 48, count 16 ...16 blocks read: OK
eMMC CLOSE Success!!!

Checking Boot Mode ... EMMC4.41
SYSTEM ENTER NORMAL BOOT MODE
Hit any key to stop autoboot:  0
iTOP-4412 #

```

4) 输入烧写命令 “sdfuse flashall” 。

这是一个全部烧写的命令，就是将 “sdupdate” 中全部的镜像烧写到开发板中，如下图所示。

```

iTOP-4412 # sdfuse flashall
SD sclk_mmc is 400K HZ
SD sclk_mmc is 50000K HZ
SD sclk_mmc is 50000K HZ
[Fusing Image from SD Card.]
.fdisk is completed

partition #    size(MB)    block start #    block count    partition_Id
1             1340           4862616          2744544         0x0C
2             1026           37290            2103156         0x83
3             1026           2140446          2103156         0x83
4             302            4243602          619014          0x83
>>>part_type : 2
Partition1: Start Address(0x4a3298), Size(0x29e0e0)
.....size checking ...
Under 8G
write FAT info: 32
Fat size : 0xa78
..Erase FAT region.....
.....

```

烧写命令 “sdfuse flashall” 可以用下面替代的烧写命令：

- sdfuse flash bootloader u-boot-iTOP-4412.bin
- sdfuse flash kernel zImage

— sdfuse flash ramdisk ramdisk-uboot.img

— sdfuse flash system system.img

替代的烧写命令允许单条执行，在执行单条烧写命令的时候，只烧写相应的单个镜像文件。例如，执行烧写命令“sdfuse flash kernel zImage”，就只会更新 Linux 内核“zImage”镜像，而不会影响其它的镜像。

5) 等待烧写完成，最后在超级终端中，输入以下命令：

— reset (在超级终端中执行该命令会重启开发板)

4.3.3 FormatTool.exe 格式化工具

可以烧写的 tf 会少一部分空间，如果想恢复成的正常的 tf 卡，可以使用 FormatTool.exe 格式化工具。

在网盘目录“iTOP4412 开发板资料汇总（不含光盘内容）\iTOP-4412 开发板所需 PC 软件（工具）\09-盘符格式化工具”文件夹下有压缩包“FormatTool.zip”，解压之后得到“FormatTool.exe”，双击打开，界面如下所示。



注意，需要选择正确的盘符，如果选择硬盘盘符，软件也会将其中的数据格式化。选择盘符之后，点击确定，完成格式化，TF 卡恢复正常。

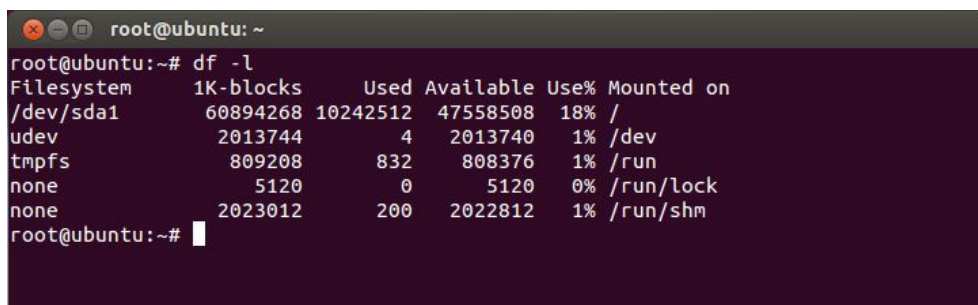
4.4 开发板出厂前首次烧写

在前面几个小节中，介绍制作 TF 卡的时候，都需要依赖核心板的 eMMC 能够启动 uboot，当核心板的 uboot 被烧写了错误的文件，导致损坏，那么就需要使用本节的方法来解决。

全新的 TF 卡并不能直接用于烧写镜像，全新的 TF 卡需要经过分区、烧写 uboot 等步骤后才能用于烧写。

下面具体介绍如何制作可以烧写的 TF 卡，首先确定 TF 卡是 FAT32 格式的。

1) 在 Ubuntu 命令行中输入 Linux 命令 “df -l” ，查看一下 Ubuntu 系统有哪些盘符。如下图所示。



```
root@ubuntu: ~  
root@ubuntu:~# df -l  
Filesystem      1K-blocks    Used Available Use% Mounted on  
/dev/sda1        60894268 10242512  47558508  18% /  
udev             2013744      4    2013740   1% /dev  
tmpfs            809208      832    808376   1% /run  
none              5120         0       5120   0% /run/lock  
none             2023012     200    2022812   1% /run/shm  
root@ubuntu:~#
```

2) 接着使用读卡器将 TF 卡连接到 PC 机的 Ubuntu 系统下，Ubuntu 系统识别 TF 卡后，再次使用 Linux 命令 “df -l” 查看盘符。将第二次查看的盘符和第一次查看的盘符对比一下，就会发现 Ubuntu 系统中多出一个盘符，这个盘符就是 TF 卡的盘符，TF 卡盘符名称在接下来的 Linux 命令中会用到，如下图所示。

```

root@ubuntu:~# df -l
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda1        60894268 10242512  47558508  18% /
udev             2013744      4    2013740   1% /dev
tmpfs            809208      832    808376   1% /run
none              5120         0      5120    0% /run/lock
none             2023012     200    2022812   1% /run/shm
root@ubuntu:~# df -l
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda1        60894268 10242560  47558460  18% /
udev             2013744      4    2013740   1% /dev
tmpfs            809208      820    808388   1% /run
none              5120         0      5120    0% /run/lock
none             2023012     200    2022812   1% /run/shm
/dev/sdb1         7767164      4    7767160   1% /media/disk
root@ubuntu:~#

```

3) 拷贝用户光盘 “06_源码_uboot 和 kernel” 文件夹中的文件 “iTop-4412_uboot_xxx” 到 Ubuntu 系统中解压，得到文件夹 “iTop4412_uboot”。

拷贝对应核心板的文件 “u-boot-iTOP-4412.bin” 到前面解压出来的文件夹 “iTop4412_uboot” 中。进入前面解压出来的文件夹 “iTop4412_uboot” 中，如下图所示。

```

root@ubuntu:~# cd /home/topeet/android4.0/iTop4412_uboot
root@ubuntu:/home/topeet/android4.0/iTop4412_uboot# ls
all00_padding.bin  E4412_N.bl1.bin  lib_nios  onenand_ip1
api                examples         lib_nios2  paddingaa
board              fs               lib_ppc    post
build_uboot.sh     include          lib_sh     README
common             lib_arm          lib_sparc  readme.txt
config.mk           lib_avr32        MAINTAINERS  rules.mk
COPYING             lib_blackfin     MAKEALL     sdfuse
cpu                 libfdt           Makefile    sdfuse.g
CREDITS             lib_generic      mklbl2     tc4_cmm.cmm
disk                lib_i386         mkconfig    tools
doc                 lib_m68k         mkuboot     u-boot-iTOP-4412.bin
drivers             lib_microblaze   nand_spl   uboot_readme.txt
E4212              lib_nips         net
root@ubuntu:/home/topeet/android4.0/iTop4412_uboot#

```

4) 烧写文件 “u-boot-iTOP-4412.bin” 到 TF 卡，具体操作如下。

在执行下面这条命令的时候，要特别特别注意！一定要分清楚，哪个盘符是 TF 卡的盘符，如果不清楚，请务必先拔掉 TF 卡，看清楚哪些盘符是属于 Ubuntu 系统的硬盘盘符，看清楚哪些盘符是 Ubuntu 系统的硬盘盘符后，再插入 TF 卡，分辨出哪个盘符是新增加的盘符，新增加的盘符才是 TF 卡的盘符。

在 Ubuntu 命令行中，执行命令 “./mkuboot /dev/sdx”（“sdx” 就是前面查到的 TF 卡盘符名，不要带数字，比如 df -l 看到的 tf 卡是/dev/sdb0，这个 0 不要带，直接写 sdb）

需要注意的是，上面的命令需要在文件夹 “iTop4412_uboot_xxx” 中执行。这个文件夹是解压 uboot 源码之后得到的文件夹。

如果这里识别为“/dev/sdb1”，那么如下图所示，使用命令“./mkuboot /dev/sdb”，不要带后面盘符的编号。

```

root@ubuntu:/home/topeet/android4.0/iTop4412_uboot# ls
all00_padding.bin  E4412_N.bl1.bin  lib_nios  onenand_ipl
api                examples         lib_nios2 paddingaa
board             fs              lib_ppc   post
build_uboot.sh    include         lib_sh    README
common           lib_arm         lib_sparc readme.txt
config.mk         lib_avr32       MAINTAINERS rules.mk
COPYING          lib_blackfin    MAKEALL   sdfuse
cpu              libbdt         Makefile  sdfuse.g
CREDITS          lib_generic     mkb12     tc4_cmm.cmm
disk             lib_i386       mkconfig  tools
doc              lib_m68k       mkuboot   u-boot-iTOP-4412.bin
drivers          lib_microblaze nand_spl  uboot_readme.txt
E4412           lib_mips       net
root@ubuntu:/home/topeet/android4.0/iTop4412_uboot# ./mkuboot /dev/sdb
Fuse iTOP-4412 trustzone uboot file into SD card
/dev/sdb reader is identified.
u-boot-iTOP-4412.bin fusing...
1029+1 records in
1029+1 records out
527104 bytes (527 kB) copied, 5.39526 s, 97.7 kB/s
u-boot-iTOP-4412.bin image has been fused successfully.
Eject SD card
root@ubuntu:/home/topeet/android4.0/iTop4412_uboot#

```

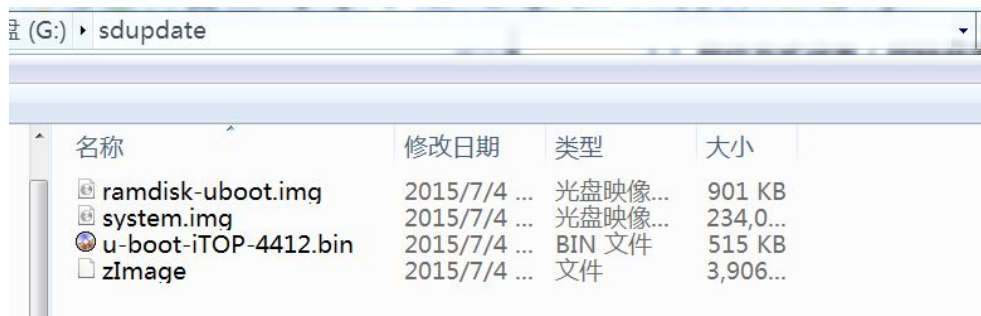
5) 将开发板的拨码开关置于 **TF 卡启动模式**，先插入经过上一步处理的 TF 卡，启动开发板，进入 uboot 模式，对 TF 卡进行分区，具体操作如下。

在超级终端中，执行下面的命令。

- fdisk -c 1 300 300 300
- fatformat mmc 1:1
- ext3format mmc 1:2
- ext3format mmc 1:3
- ext3format mmc 1:4

6) 到上一步，TF 卡就制作好了。下面继续介绍如何使用制作好的 TF 卡。

7) 将 TF 卡插入 Window 双系统，新建“sdupdate”，将 uboot 等四个文件拷贝到“sdupdate”文件夹中，也可以选择只拷贝 uboot 文件，如下图所示。



8) 将 TF 卡插入开发板，设置开发板为 tf 卡启动模式，启动开发板。对 eMMC 进行格式化分区命令以及擦除命令，最后使用命令“sdfuse flashall”，就可以将 uboot、内核以及文件系统更新到开发板的 eMMC 中，这里也可以选择只更新 uboot。

9) 烧写镜像完成之后，将开发板设置为 eMMC 启动模式，开发板就可以正常启动 uboot，然后可以使用 fastboot 烧写或者 tf 卡烧写内核或者文件系统的镜像，烧写方法参考 4.2 或者 4.3 小节。