



GEC6818开发板刷机教程

作者	日期	版本	更新内容
粤嵌	2017.02.01	V1.0	初始版本

版权声明

本文档归属广州粤嵌通信科技股份有限公司（以下简称“粤嵌”）所有，并保留一切权利，非经粤嵌同意（书面形式），任何单位及个人不得擅自摘录本文档部分或者全部内容，违者将追究其法律责任。

目 录

版权声明.....	1
目 录.....	2
第一章 前言.....	3
注意事项.....	3
开发板启动顺序.....	3
第二章 使用 fastboot 烧写镜像.....	4
Windows 下使用 fastboot 烧写（推荐）.....	4
安装串口工具 secureCRT.....	4
安装 fastboot.....	6
烧写 Linux 映像.....	6
烧写 android 映像.....	8
Linux 下使用 fastboot 烧写（不推荐）.....	11
安装串口终端 minicom.....	11
安装 fastboot 工具.....	12
烧写 Linux 映像.....	13
烧写 android 映像.....	14
第三章 使用 SD 卡烧写镜像.....	16
注意事项.....	16
Windows 下制作 SD 启动卡（推荐）.....	16
Linux 下制作 SD 启动卡.....	20
使用 SD 启动卡烧写 Linux 映像.....	23
使用 SD 启动卡烧写 Android 映像.....	24

第一章 前言

开发板的嵌入式操作系统，包含 Linux 和 Android 操作系统。我们出厂时会烧写或者固化其中一个操作系统在里面。本手册讲述如何固化嵌入式操作系统到我们的开发板中。

注意事项

我们把编译好的镜像系统文件，通过 SD 或者 USB 的下载方式，固化到板载的 eMMC 存储器中（ROM），以下简称为“刷机”。

方法一：通过 fastboot 工具，USB 下载方式

方法二：通过 SD 卡方式

使用 fastboot 工具烧写 Linux 和 android 映像时，核心板必须存在 uboot（引导程序），因为烧写时需要使用 uboot 上的 fastboot 功能，

在板子不存在 uboot 时，请使用 SD 卡烧写方式。

使用 fastboot 烧写时，电脑上必须存在串口接口或者拥有 usb 转串口模块，使其连接电脑与开发板，让电脑能够通过串口与开发板通信

开发板启动顺序

6818 开发板硬件配置固定了开发板启动顺序如下：

1st: 从 TF 卡启动

2nd: 从 EMMC 启动

3rd: 从 USB 启动

开发板上电后首先从 TF 卡启动，若 SD0 插入了启动卡则从 SD 启动；如果 SD0 未插卡或者插入的不是启动卡，则启动失败；然后从板载 EMMC(SD2)启动，若 EMMC 中已经烧录固件则启动成功，否则启动失败，最后尝试从 USB 启动。

第二章 使用 fastboot 烧写镜像

Windows 下使用 fastboot 烧写（推荐）

安装串口工具 secureCRT

- 1、下载并安装 secureCRT 工具，打开工具，点击左上角“快速链接”按钮：

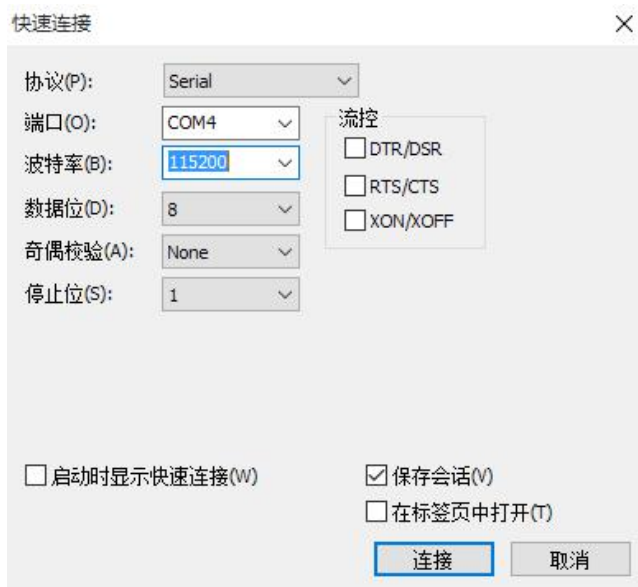


- 2、使用串口线或 USB 转串口模块连接开发板与电脑，打开 Windows 的设备管理器，查看串口端口号：

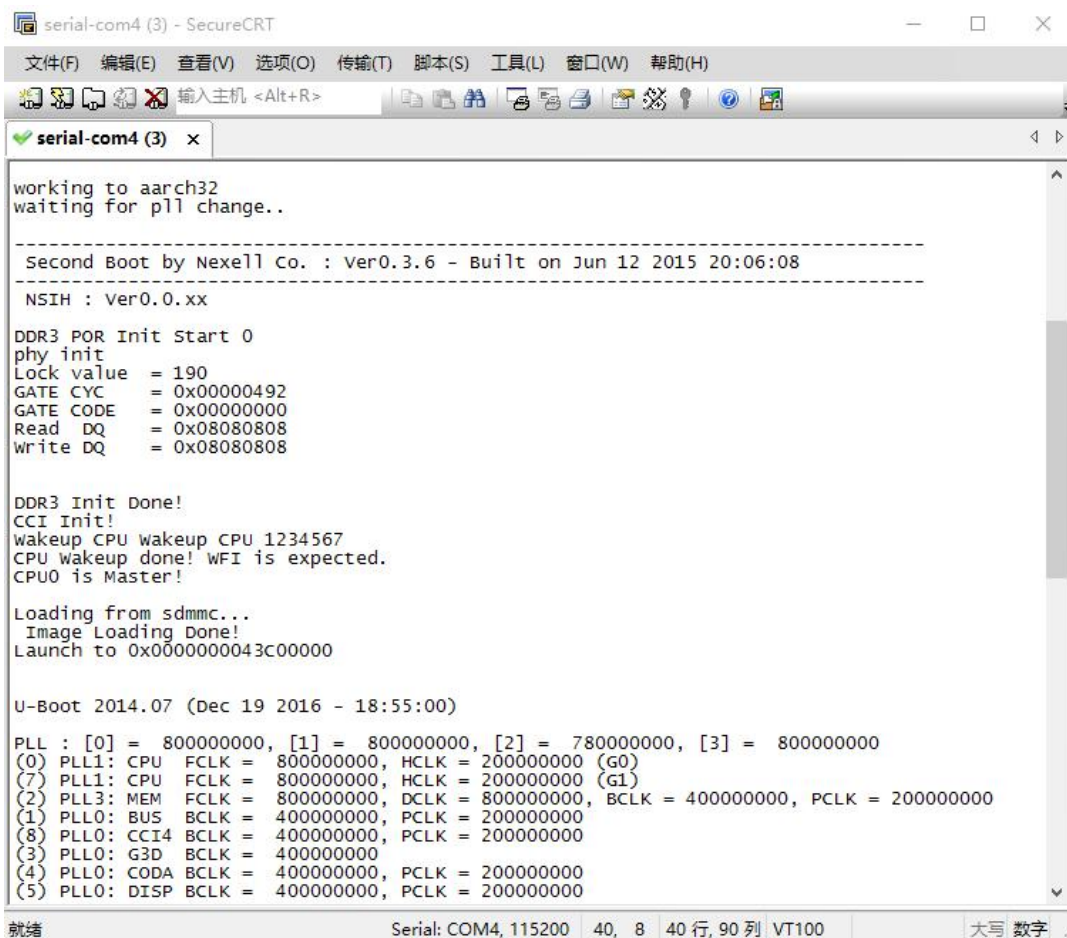


可以看到串口端口号为 COM4。

- 3、回到 secureCRT 工具界面，设置“快速链接”的配置。选择协议为 Serial，端口为 COM4，波特率为 115200，取消勾选流控 RTS/CTS：



- 4、点击连接后，打开开发板电源，secureCRT 终端输出开发板启动信息，说明 secureCRT 配置完成：



安装 fastboot

- 1、说明：在多数情况下，在 Windows 下使用 fastboot 工具烧写并不需要把 fastboot 安装到系统中，只需要解压 fastboot 工具并在解压目录中运行工具进行烧写即可。

名称	修改日期	类型	大小
 adb.exe	2010/6/10 20:13	应用程序	566 KB
 AdbWinApi.dll	2009/11/19 13:02	应用程序扩展	94 KB
 AdbWinUsbApi.dll	2011/2/24 10:00	应用程序扩展	60 KB
 fastboot.exe	2011/2/24 10:00	应用程序	971 KB

烧写 Linux 映像

- 1、使用串口线或 USB 转串口模块连接开发板与电脑。
- 2、打开 secureCRT 终端连接开发板串口。
- 3、打开开发板电源，在 secureCRT 中查看串口打印的启动信息，在 uboot 启动的 3 秒内按任意键进入 uboot 命令行模式，执行如下指令并回车：

```
fastboot
```

secureCRT 终端下将打印如下信息：

```
Fastboot Partitions:
mmc.2: ubootpak, img : 0x200, 0x78000
mmc.2: 2ndboot, img : 0x200, 0x4000
mmc.2: bootloader, img : 0x8000, 0x70000
mmc.2: boot, fs : 0x100000, 0x4000000
mmc.2: system, fs : 0x4100000, 0x2f200000
mmc.2: cache, fs : 0x33300000, 0x1ac00000
mmc.2: misc, fs : 0x4e000000, 0x800000
mmc.2: recovery, fs : 0x4e900000, 0x1600000
mmc.2: userdata, fs : 0x50000000, 0x0
Support fstype : 2nd boot factory raw fat ext4 emmc nand ubi ubifs
Reserved part : partmap mem env cmd
DONE: Logo bmp 311 by 300 (3bpp), len=280854
DRAW: 0x47000000 -> 0x46000000
Load USB Driver: android
Core usb device tie configuration done
OTG cable Connected!
-----
```

4、插入 micro USB 线连接到电脑。









5、解压 fastboot 工具压缩包到一个目录下，把 Linux 映像文件 ubootpak.bin、boot.img、qt-rootfs.img 全部复制到该目录中。

6、右键使用记事本编辑 Windows 脚本文件 auto.bat，查看烧写映像文件名是否与我们编译出来的 android 映像文件名相同，不相同则重命名 android 映像文件名。

脚本文件 auto.bat 的内容：

```
fastboot flash ubootpak ubootpak.bin
fastboot flash boot boot.img
fastboot flash system qt-rootfs.img
fastboot reboot
```

fastboot 文件夹下各文件如图：

名称	修改日期	类型	大小
 adb.exe	2010/6/10 20:13	应用程序	566 KB
 AdbWinApi.dll	2010/6/10 20:13	应用程序扩展	94 KB
 AdbWinUsbApi.dll	2010/6/10 20:13	应用程序扩展	60 KB
 auto.bat	2017/3/27 11:50	Windows 批处理...	1 KB
 boot.img	2017/2/15 10:43	光盘映像文件	16,701 KB
 fastboot.exe	2012/3/20 11:03	应用程序	969 KB
 qt-rootfs.img	2017/2/10 15:50	光盘映像文件	358,400 KB
 ubootpak.bin	2017/4/27 15:45	BIN 文件	333 KB

7、确认无误后，退出编辑，双击打开（或右键管理员权限打开）auto.bat，可以看到 Windows 下会打开命令终端，打印出如下信息：

C:\Windows\system32\cmd.exe

```
F:\work\工作\6818\gec-Linux>fastboot flash ubootpak ubootpak.bin
sending 'ubootpak' (332 KB)... OKAY
writing 'ubootpak'... OKAY

F:\work\工作\6818\gec-Linux>fastboot flash boot boot.img
sending 'boot' (16700 KB)... OKAY
writing 'boot'... OKAY

F:\work\工作\6818\gec-Linux>fastboot flash system qt-rootfs.img
sending 'system' (358400 KB)... OKAY
writing 'system'...
```

6、在 secureCRT 终端下，会打印出如下信息，说明烧写成功：


```
-----
starting download of 340732 bytes
downloading 337920 -- 99% complete.
downloading of 340732 bytes to 0x48000000 (0xf8000000) finished
Flash : ubootpak
head boot dev = 2
update mmc.2 type 2ndboot = 0x200(0x1) ~ 0x532fc(0x29a): Done
Flash : ubootpak - DONE
Starting download of 17101032 bytes
downloading 16930816 -- 99% complete.
downloading of 17101032 bytes to 0x48000000 (0xf8000000) finished
Flash : boot
Flash : boot - DONE
Starting download of 367001600 bytes
downloading 363332096 -- 99% complete.
downloading of 367001600 bytes to 0x48000000 (0xf8000000) finished
Flash : system
write image to 0x4100000(0x20800), 0x15e00000(0x20800)
Flash : system - DONE
resetting ...
```

7、烧写完成后，Windows 命令框会自动退出，按下重启键重新启动开发板。在 uboot 启动的 3 秒内按任意键进入 uboot 命令行模式，执行如下指令，设置系统启动环境变量，保存后重新启动即烧写成功：

```
setenv bootcmd "ext4load mmc 2:1 0x48000000 uImage;bootm 0x48000000"
save
```

执行完以上指令，即可正常启动 Linux 系统了。每执行一条指令，在液晶屏上都会有相应的界面提示，用户可以很清晰的观察升级的状态。

烧写 android 映像

- 1、使用串口线或 USB 转串口模块连接开发板与电脑。
- 2、打开 secureCRT 终端连接开发板串口。
- 3、打开开发板电源，在 secureCRT 中查看串口打印的启动信息，在 uboot 启动的 3 秒内按任意键进入 uboot 命令行模式，执行如下指令并回车：

```
fastboot
```

secureCRT 终端下将打印如下信息：

```
Fastboot Partitions:
mmc.2: ubootpak, img : 0x200, 0x78000
mmc.2: 2ndboot, img : 0x200, 0x4000
mmc.2: bootloader, img : 0x8000, 0x70000
mmc.2: boot, fs : 0x100000, 0x4000000
mmc.2: system, fs : 0x4100000, 0x2f200000
mmc.2: cache, fs : 0x33300000, 0x1ac00000
mmc.2: misc, fs : 0x4e000000, 0x8000000
```

```
mmc.2: recovery, fs : 0x4e900000, 0x1600000
mmc.2: userdata, fs : 0x50000000, 0x0
Support fstype : 2nd boot factory raw fat ext4 emmc nand ubi ubifs
Reserved part : partmap mem env cmd
DONE: Logo bmp 311 by 300 (3bpp), len=280854
DRAW: 0x47000000 -> 0x46000000
Load USB Driver: android
Core usb device tie configuration done
OTG cable Connected!
-----
```

4、插入 micro USB 线连接到电脑。

5、解压 fastboot 工具压缩包到一个目录下，把 android 映像文件 ubootpak.bin、boot.img、system.img、cache.img、userdata.img 全部复制到该目录中。

6、右键使用记事本编辑 Windows 脚本文件 auto.bat，查看烧写映像文件名是否与我们编译出来的 android 映像文件名相同，不相同则重命名 android 映像文件名。脚本文件 auto.bat 的内容：

```
fastboot flash ubootpak ubootpak.bin
fastboot flash boot boot.img
fastboot flash system system.img
fastboot flash cache cache.img
fastboot flash userdata userdata.img
```

fastboot 文件夹下各文件如图：

名称	修改日期	类型	大小
 adb.exe	2010/6/10 20:13	应用程序	566 KB
 AdbWinApi.dll	2010/6/10 20:13	应用程序扩展	94 KB
 AdbWinUsbApi.dll	2010/6/10 20:13	应用程序扩展	60 KB
 auto.bat	2016/12/17 12:53	Windows 批处理...	1 KB
 boot.img	2017/2/24 17:31	光盘映像文件	16,557 KB
 cache.img	2017/2/13 12:27	光盘映像文件	8,753 KB
 fastboot.exe	2012/3/20 11:03	应用程序	969 KB
 system.img	2017/3/3 11:22	光盘映像文件	429,066 KB
 ubootpak.bin	2017/4/27 15:45	BIN 文件	333 KB
 userdata.img	2017/3/3 11:23	光盘映像文件	95,642 KB

7、确认无误后，退出编辑，双击打开（或右键管理员权限打开）auto.bat，可以看到 Windows 下会打开命令终端，打印出如下信息：

```
-----
Starting download of 340732 bytes
downloading 337920 -- 99% complete.
downloading of 340732 bytes to 0x48000000 (0xf8000000) finished
Flash : ubootpak
head boot dev = 2
update mmc.2 type 2ndboot = 0x200(0x1) ~ 0x532fc(0x29a): Done
Flash : ubootpak - DONE
Starting download of 16953576 bytes
downloading 16784896 -- 99% complete.
downloading of 16953576 bytes to 0x48000000 (0xf8000000) finished
Flash : boot
Flash : boot - DONE
Starting download of 439362724 bytes
downloading 434970112 -- 99% complete.
downloading of 439362724 bytes to 0x48000000 (0xf8000000) finished
Flash : system
Flash : system - DONE
Starting download of 8962352 bytes
downloading 8873472 -- 99% complete.
downloading of 8962352 bytes to 0x48000000 (0xf8000000) finished
Flash : cache
Flash : cache - DONE
Starting download of 97936888 bytes
downloading 96958464 -- 99% complete.
downloading of 97936888 bytes to 0x48000000 (0xf8000000) finished
Flash : userdata
Flash : userdata - DONE
```

6、在 secureCRT 终端下，会打印出如下信息，说明烧写成功：

```
Core usb device tie configuration done
OTG cable connected!
-----
Starting download of 340728 bytes
downloading 337920 -- 99% complete.
downloading of 340728 bytes to 0x48000000 (0xf8000000) finished
Flash : ubootpak
head boot dev = 2
update mmc.2 type 2ndboot = 0x200(0x1) ~ 0x532f8(0x29a): Done
Flash : ubootpak - DONE
Starting download of 16953576 bytes
downloading 16784896 -- 99% complete.
downloading of 16953576 bytes to 0x48000000 (0xf8000000) finished
Flash : boot
Flash : boot - DONE
Starting download of 443044932 bytes
downloading 438615040 -- 99% complete.
downloading of 443044932 bytes to 0x48000000 (0xf8000000) finished
Flash : system
Flash : system - DONE
Starting download of 8962352 bytes
downloading 8873472 -- 99% complete.
downloading of 8962352 bytes to 0x48000000 (0xf8000000) finished
Flash : cache
Flash : cache - DONE
Starting download of 97924564 bytes
downloading 96946176 -- 99% complete.
downloading of 97924564 bytes to 0x48000000 (0xf8000000) finished
Flash : userdata
Flash : userdata - DONE
```

7、烧写完成后，Windows 命令框会自动退出，按下重启键重新启动开发板。在 uboot 启动的 3 秒内按任意键进入 uboot 命令行模式，执行如下指令，设置系统启动环境变量，保存后重新启动即烧写成功：

```
setenv bootcmd "ext4load mmc 2:1 0x48000000 uImage;ext4load mmc 2:1 0x49000000
root.img.gz; bootm 0x48000000"

save
```

Linux 下使用 fastboot 烧写（不推荐）

安装串口终端 minicom

1、使用如下指令安装：

```
sudo apt-get install minicom
```

2、如果是使用 USB 转串口模块，目前市面上大多都是 pl2303 方案，需要输入如下命令查询驱动是否正常加载：

```
lsmod |grep pl2303
```

返回如下信息则加载正常：

```
lqm@lqm:~$ lsmod |grep pl2303
pl2303 11756 1
usbserial 33100 3 pl2303
```

3、查看串口设备名：

```
dmesg | tail -f
```

返回：

```
ERROR! H2M_MAILBOX still hold by MCU. command fail
---> RTMPFreeTxRxRingMemory
<--- RTMPFreeTxRxRingMemory
RTUSB disconnect successfully
usb 2-4: USB disconnect, address 3
pl2303 ttyUSB0: pl2303 converter now disconnected from ttyUSB0
pl2303 2-4:1.0: device disconnected
usb 2-4: new full speed USB device using ohci_hcd and address 5
pl2303 2-4:1.0: pl2303 converter detected
usb 2-4: pl2303 converter now attached to ttyUSB0
exit 0
```

其中 ttyUSB0 就是串口设备名。

4、输入命令配置串口参数：

```
sudo minicom -s
```

选择 Serial port setup,

- 设置串口终端：选择 A，输入正确的串口终端，如果直接使用串口，通常设置为 ttyS0，如果使用 USB 转串口，通常设置为 ttyUSB0，回车。ttyS0 以及 ttyUSB0 中的 0 代表串口 0。

- 设置波特率以及数据位：选择 E，输入 115200 8N1，回车。115200 代表波特率，8N1 代表 8 个数据位，1 个停止位。
- 设置流控：选择 F 和 G，都设置为 No，表示不使用流控，回车。

选择 Save setup as df1 保存设置。

5、常见使用问题：

非正常关闭 minicom，会在/var/lock 下创建几个文件 LCK*，这几个文件阻止了 minicom 的运行，将它们删除后即可恢复正常。

安装 fastboot 工具

- 1、执行如下指令安装 fastboot:

```
sudo apt-get install android-tools-fastboot
```

2、在使用 fastboot 时，需要获取 root 权限，为了解除这个限制，可以使用如下方法，让使用 fastboot 时不需要管理员权限。

到/etc/udev/rules.d/目录下，新建 51-android.rules 文件，内容如下：

（注意， OWNER 里面填的” gec” 务必换成自己 ubuntu 系统的用户名）

```
# adb protocol on passion (Nexus One)
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", ATTR{idProduct}=="4e12",
MODE="0666",OWNER="gec"
# adb protocol on crespo/crespo4g (Nexus S)
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", ATTR{idProduct}=="4e22",
MODE="0666",OWNER="gec"
# fastboot protocol on crespo/crespo4g (Nexus S)
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", ATTR{idProduct}=="4e20",
MODE="0666",OWNER="gec"
# fastboot protocol on stingray/wingray (Xoom)
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", ATTR{idProduct}=="708c",
MODE="0666",OWNER="gec"
# fastboot protocol on maguro/toro (Galaxy Nexus)
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", ATTR{idProduct}=="4e30",
MODE="0666",OWNER="gec"
# fastboot protocol on x210/x4412/x6818
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", ATTR{idProduct}=="0002",
MODE="0666",OWNER="gec"
```


烧写 Linux 映像

- 1、使用串口线或 USB 转串口模块连接开发板与电脑。
- 2、执行如下指打开 minicom:

```
minicon
```

- 3、打开开发板电源，在 minicom 中查看串口打印的启动信息，在 uboot 启动的 3 秒内按任意键进入 uboot 命令行模式，执行如下指并回车:

```
fastboot
```

minicom 终端下将打印如下信息:

```
Fastboot Partitions:
mmc.2: ubootpak, img : 0x200, 0x78000
mmc.2: 2ndboot, img : 0x200, 0x4000
mmc.2: bootloader, img : 0x8000, 0x70000
mmc.2: boot, fs : 0x100000, 0x4000000
mmc.2: system, fs : 0x4100000, 0x2f200000
mmc.2: cache, fs : 0x33300000, 0x1ac00000
mmc.2: misc, fs : 0x4e000000, 0x800000
mmc.2: recovery, fs : 0x4e900000, 0x1600000
mmc.2: userdata, fs : 0x50000000, 0x0
Support fstype : 2nd boot factory raw fat ext4 emmc nand ubi ubifs
Reserved part : partmap mem env cmd
DONE: Logo bmp 311 by 300 (3bpp), len=280854
DRAW: 0x47000000 -> 0x46000000
Load USB Driver: android
Core usb device tie configuration done
OTG cable Connected!
-----
```

- 4、插入 micro USB 线连接到电脑。
- 5、另外开启一个 ubuntu 命令行终端，输入如下指令更新 Linux 系统各个部分需要的映像:

```
fastboot flash ubootpak ubootpak.bin

fastboot flash boot boot.img

fastboot flash system qt-rootfs.img
```

- 6、烧写完成后，Windows 命令框会自动退出，按下重启键重新启动开发板。在 uboot 启动的 3 秒内按任意键进入 uboot 命令行模式，执行如下指令，设置系统启动环境变量，保存后重新启动即烧写成功:

```
setenv bootcmd "ext4load mmc 2:1 0x48000000 uImage;bootm 0x48000000"
setenv bootargs "lcd=at070tn92 tp=gs1x680-linux root=/dev/mmcblk0p2 rw rootfstype=ext4"
save
```

执行完以上指令，即可正常启动 Linux 系统了。每执行一条指令，在液晶屏上都会有相应的界面提示，用户可以很清晰的观察升级的状态。

烧写 android 映像

- 1、使用串口线或 USB 转串口模块连接开发板与电脑。
- 2、执行如下指打开 minicom:

```
minicon
```

- 3、打开开发板电源，在 minicom 中查看串口打印的启动信息，在 uboot 启动的 3 秒内按任意键进入 uboot 命令行模式，执行如下指并回车:

```
fastboot
```

minicom 终端下将打印如下信息:

```
Fastboot Partitions:
mmc.2: ubootpak, img : 0x200, 0x78000
mmc.2: 2ndboot, img : 0x200, 0x4000
mmc.2: bootloader, img : 0x8000, 0x70000
mmc.2: boot, fs : 0x100000, 0x4000000
mmc.2: system, fs : 0x4100000, 0x2f200000
mmc.2: cache, fs : 0x33300000, 0x1ac00000
mmc.2: misc, fs : 0x4e000000, 0x800000
mmc.2: recovery, fs : 0x4e900000, 0x1600000
mmc.2: userdata, fs : 0x50000000, 0x0
Support fstype : 2nd boot factory raw fat ext4 emmc nand ubi ubifs
Reserved part : partmap mem env cmd
DONE: Logo bmp 311 by 300 (3bpp), len=280854
DRAW: 0x47000000 -> 0x46000000
Load USB Driver: android
Core usb device tie configuration done
OTG cable Connected!
-----
```

- 4、插入 micro USB 线连接到电脑。

- 5、另外开启一个 `ubuntu` 命令行终端，输入如下指令更新 `android` 系统各个部分需要的映像：

```
fastboot flash ubootpak ubootpak.bin
fastboot flash boot boot.img
fastboot flash system system.img
fastboot flash cache cache.img
fastboot flash userdata userdata.img
```

- 6、烧写完成后，`Windows` 命令框会自动退出，按下重启键重新启动开发板。在 `uboot` 启动的 3 秒内按任意键进入 `uboot` 命令行模式，执行如下指令，设置系统启动环境变量，保存后重新启动即烧写成功：

```
setenv bootcmd "ext4load mmc 2:1 0x48000000 uImage;ext4load mmc 2:1 0x49000000
root.img.gz; bootm 0x48000000"
save
```

执行完以上指令，即可正常启动 `android` 系统了。每执行一条指令，在液晶屏上都会有相应的界面提示，用户可以很清晰的观察升级的状态。

第三章 使用 SD 卡烧写镜像

注意事项

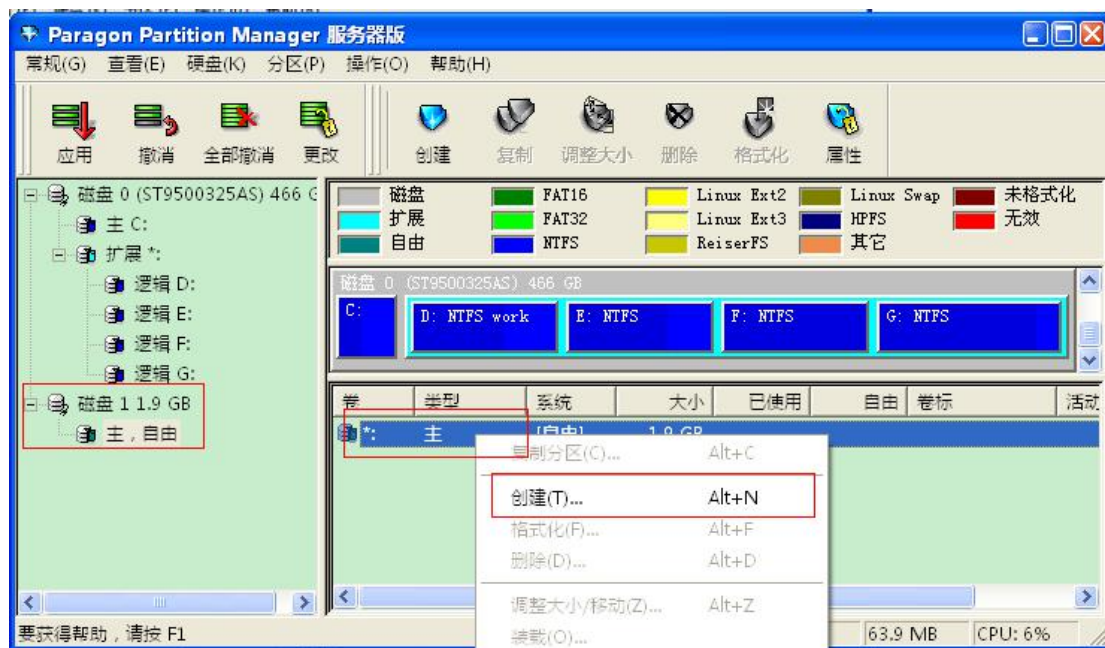
使用 SD 卡烧写必须准备一张不小于 2GB 的 micro SD 卡以及 micro SD 卡读卡器。使用 SD 卡烧写会清空 SD 卡中原有内容，请提前做好备份。

Windows 下制作 SD 启动卡（推荐）

1、注意事项：WinPM 这个软件是为 windowsXP 而设置的，使用 windows7 或者 windows8 或者 windows10 的同学记得选择该软件，右键 —> 兼容性建议 —> 尝试建议性操作 --> 启动程序）

2、整个流程是在 Windows 下要先通过 WinPM 这个软件来对 SD 卡进行分区，然后再使用 IROM_Fusing_Tool_6818.exe 的软件，将我们编译好的 bootloader 固化到 SD 卡里面，然后再选择从 SD 卡启动我们的开发板进行烧写。以下是详细步骤：

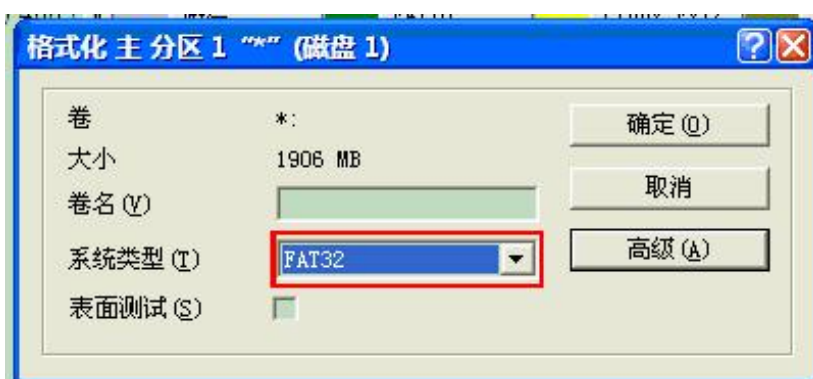
- 1) 打开 VinPM，这个软件位于光盘文件的 Tools 目录下面。
- 2) 选择 SD 卡，注意：选择你正确的 SD 卡盘符



3) 预留 250M 的空间给 uboot。



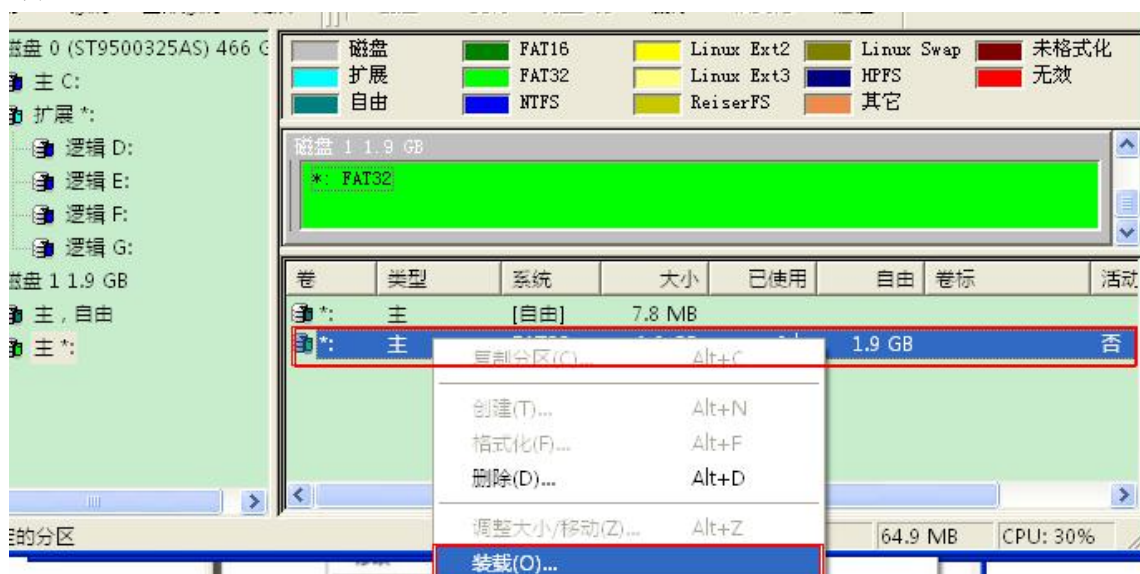
4) 选择是 FAT32 的格式



5) 选择确定



6) 选择剩余的未分配的空间，右键选择装载



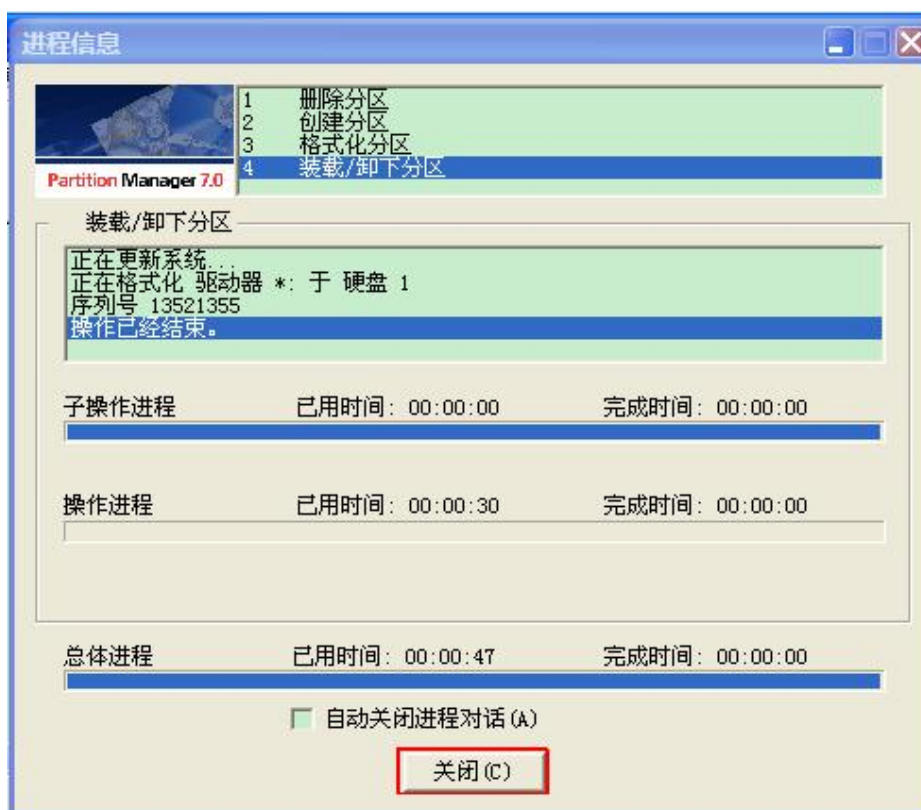
7) 系统会给我们新建的这个盘分一个盘符，我设置为 K，点击确定即可



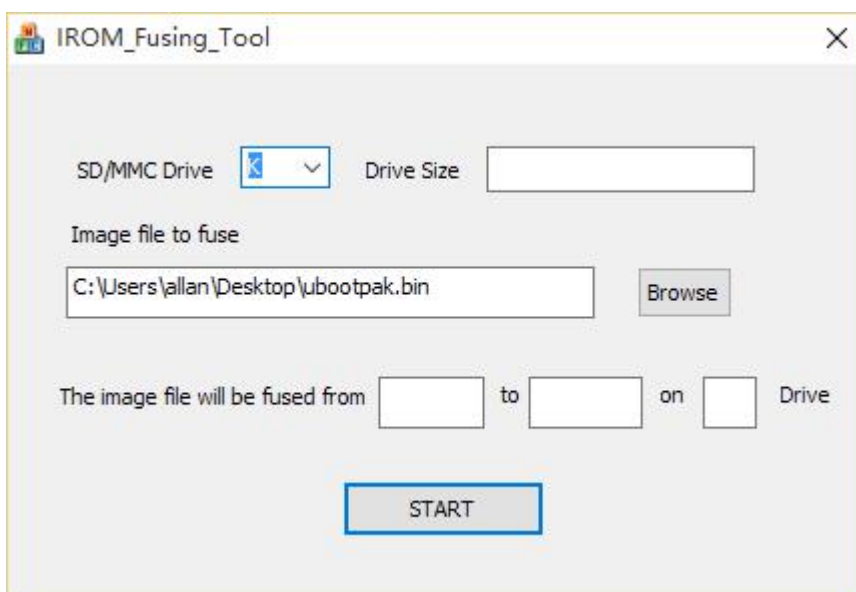
8) 对我们的 SD 卡分区好之后，点击应用点击确定，就可以对我们的 SD 卡进行分区，建立分区表信息。



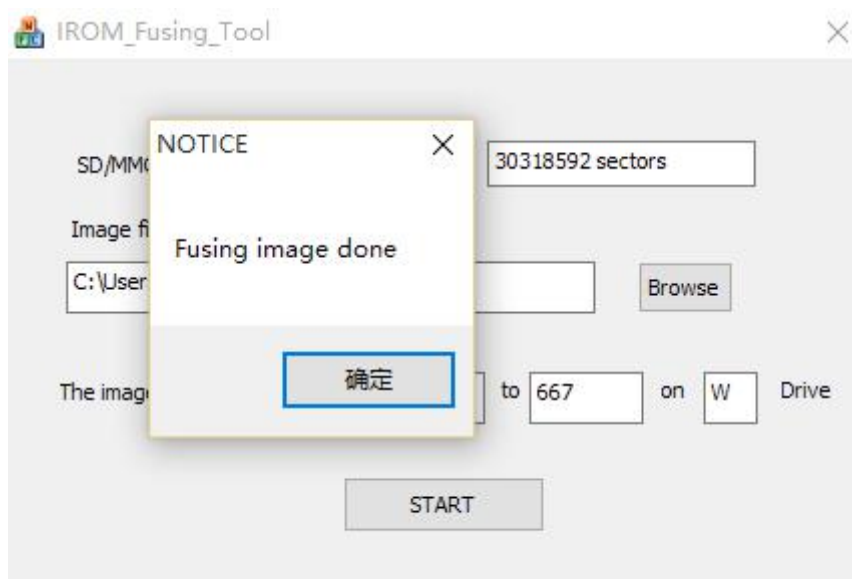
9) 分区完成之后, 点击关闭即可。



10) 对 SD 卡分好区之后, 以管理员权限运行 IROM_Fusing_Tool_6818.exe 工具。点击 Browse 浏览到已经编译好的 uboot, 所需烧写的 uboot 文件在 out/release 目录下, 文件名字为: ubootpak.bin, 如下图所示:



11) 选择我们需要烧写的正确的盘符（我们上面设置 SD 的盘符是 K，这里选择 K）。然后点击 START，显示 Fusing image done 表示烧写完成。完成 uboot



至此，SD 启动卡的制作已经完成。

Linux 下制作 SD 启动卡

1、输入如下命令查看 linux 系统原有的设备节点。

```
cat /proc/partitions
```

返回：

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

2、使用读卡器连接 SD 卡与电脑，输入如下命令查看设备节点，经过对比原有设备节点获得 SD 卡设备节点名为 sdb

```
cat /proc/partitions
```

返回：

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

3、输入如下命令，修改 SD 卡中的分区

```
sudo fdisk /dev/sdb
```

返回：

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

输入 **d** 并回车，删除所有分区。返回：

```
Selected partition 1
```

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

输入 **w** 并回车，保存所有已经修改的分区信息。返回：

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

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

```
Syncing disks.
```

拔掉 SD 卡，再插入 PC 机上，输入如下命令查看现有的设备节点：

```
cat /proc/partitions
```

返回：

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

至此，SD 卡原分区 `/dev/sdb1` 被删除。

4、使用 gparted 工具给 SD 卡预留 256M 空间，用于存放 uboot 映像。使用如下命令安装 gparted 工具：

```
sudo apt-get install gparted
```

使用如下命令使用 **gparted** 打开 SD 卡分区表：

```
sudo gparted /dev/sdb
```

弹出：



5、选择分区->新建，预留 256M 空间给 uboot，剩下的分区使用 fat32 格式，设置如下图所示，设置完成后点击添加，选择菜单中的应用全部操作，完成 SD 卡的分区。



6、进入映像生成目录 out/release，执行如下指令烧写 uboot 到 SD 卡：（注意，这里/dev/sdb 为 SD 卡的节点，可查询节点名称后再执行下面的烧写脚本。）

```
sudo ./x6818-sdmmc.sh /dev/sdb ubootpak.bin
```

返回如图：

```
594+1 records in
595+0 records out
304640 bytes (305 kB) copied, 0.485139 s, 628 kB/s
^^ The image is fused successfully
```

这时，该 SD 卡就可以引导开发板启动 uboot 了。

使用 SD 启动卡烧写 Linux 映像

1、使用读卡器连接已经制作好的 SD 启动卡与电脑，SD 启动卡的制作方法请参考上文《Linux 下制作 SD 启动卡》或《Windows 下制作 SD 启动卡》。

2、在 SD 卡中，新建名为 `gec-Linux` 的文件夹，把编译好的的目标文件 `ubootpak.bin`、`boot.img`、`qt-rootfs.img`、`env.txt` 四个文件复制并粘贴到该目录下，如图：

名称	修改日期	类型	大小
 <code>boot.img</code>	2017/2/15 10:43	光盘映像文件	16,701 KB
 <code>env.txt</code>	2016/12/16 19:31	文本文档	1 KB
 <code>qt-rootfs.img</code>	2017/2/10 15:50	光盘映像文件	358,400 KB
 <code>ubootpak.bin</code>	2017/4/27 15:45	BIN 文件	333 KB

3、`env.txt` 为传给 `uboot` 的启动参数，他记录了硬件模块的名字，以及系统启动时文件加载的地址。`env.txt` 内容如下：

```
*****linux3.4.39 & qt5.4 *****
bootcmd=ext4load mmc 2:1 0x48000000 uImage;bootm 0x48000000
bootargs=lcd=at070tn92 tp=gs1x680-linux root=/dev/mmcblk0p2 rw rootfstype=ext4
ubootpak=1
boot=1
system=1
userdata=0
cache=0
```

4、将 SD 卡插到开发板的 SD0 卡槽，打开开发板电源，这时系统将会自动升级。等待数秒后，开发板会自动进行烧写。烧写完毕将会自动重启，进入 Linux 系统。

说明：默认 `uboot` 会自动判断 SD 卡的 `gec-Linux` 目录中的映像文件是否有更新，一旦发现映像和烧写到开发板上的不同，则会自动更新映像，否则不会再次更新；**如果需要强制进入升级模式，开机上电时按住返回键（K2，调试串口中会打印 `force update`）即可重新进入 SD 卡烧写模式，开始自动烧写。**

使用 SD 启动卡烧写 Android 映像

1、使用读卡器连接已经制作好的 SD 启动卡与电脑，SD 启动卡的制作方法请参考上文《Linux 下制作 SD 启动卡》或《Windows 下制作 SD 启动卡》。

2、在 SD 卡中，新建名为 GEC-android 的文件夹，把编译的目标文件生成目录 out/release 中的 ubootpak.bin, boot.img, system.img, userdata.img 以及 cache.img, env.txt 六个文件复制并粘贴到该目录下，如图：

名称	修改日期	类型	大小
 boot.img	2017/2/16 10:47	光盘映像文件	16,557 KB
 cache.img	2017/2/13 12:27	光盘映像文件	8,753 KB
 env.txt	2017/2/15 18:18	文本文档	1 KB
 system.img	2017/2/16 10:48	光盘映像文件	428,650 KB
 ubootpak.bin	2017/2/16 10:49	BIN 文件	334 KB
 userdata.img	2017/2/16 10:49	光盘映像文件	95,630 KB

3、env.txt 为传给 uboot 的启动参数，他记录了硬件模块的名字，以及系统启动时文件加载的地址。env.txt 内容如下：

```
bootcmd=ext4load mmc 2:1 0x48000000 uImage;ext4load mmc 2:1 0x49000000 root.img.gz;bootm
0x48000000
bootargs=lcd=at070tn92 tp=ft5x06 cam=OV5645
```

4、将 SD 卡插到开发板的 SD0 卡槽，打开开发板电源，这时系统将会自动升级。等待数秒后，开发板会自动进行烧写。烧写完毕将会自动重启，进入 android 系统。

说明：默认 uboot 会自动判断 SD 卡的 GEC-android 目录中的映像文件是否有更新，一旦发现映像和烧写到开发板上的不同，则会自动更新映像，否则不会再次更新；**如果需要强制进入升级模式，开机上电时按住返回键（K2，调试串口会打印 force update）即可重新进入 SD 卡烧写模式，开始自动烧写。**