

SSD20X Rootfs下烧录mac地址

概述

一、开机自动执行SD卡中的脚本

二、mac烧写脚本

脚本工作原理

烧写脚本

概述

注意：本文主要介绍，在系统启动后通过SD卡或U盘的方式，批量烧录mac地址。

上电进入系统后，自启SD卡中的脚本，通过脚本文件往boot下的环境变量写mac地址。每次写完往后加1。就可以实现批量烧录mac地址了。

只要在不重新烧录uboot的情况下，更新固件，mac地址是不会被覆盖的。

参考：

 [rootfs_burn_mac.zip](#)

一、开机自动执行SD卡中的脚本

我们可以在/customer/demo.sh中添加相应的自启脚本。系统启动后就自动执行burn_mac.sh进行烧写mac地址。

```

insmod /config/modules/4.9.84/ntfs.ko
insmod /config/modules/4.9.84/usb-common.ko
insmod /config/modules/4.9.84/usbcare.ko
insmod /config/modules/4.9.84/ehci-hcd.ko
insmod /config/modules/4.9.84/usb-storage.ko
insmod /config/modules/4.9.84/usbhid.ko
insmod /config/modules/4.9.84/mdrv_crypto.ko

#burn mac
mount /dev/mmcblk0p1 /sdcard -o iocharset=utf8
if [ ! $? == 0 ]; then
    echo "mount fail!"
    exit 0
fi

/sdcard/burn_mac.sh

sync

umount /sdcard

exit 0

#kernel mod list

```

二、mac烧写脚本

脚本工作原理

脚本执行后，读取/sdcard/mac_current中的mac的起始地址，通过/etc/fw_setenv向uboot中的环境变量，写入mac地址，并将写入的地址记录在/sdcard/mac_record.txt中。每次烧写完成后，向后加1，在烧写一下块核心板时，从新的mac地址开始。并且在烧录mac地址前，校验当前设备是否已经烧录过地址了（具体可自定义判断规则），防止重复烧录。

烧写脚本

烧写脚本如下：

```
1  #!/bin/sh
2  MAC_CURNNET=/sdcard/mac_current
3  MAC_RECORD=/sdcard/mac_record.txt
4
5  burn_ok() {
6      echo 14 > /sys/class/gpio/export
7      echo out > /sys/class/gpio/gpio14/direction
8
9      for i in 1 2 3
10     do
11         echo 1 > /sys/class/gpio/gpio14/value
12         sleep 0.5
13         echo 0 > /sys/class/gpio/gpio14/value
14         sleep 0.5
15     done
16
17     echo "-----> burn_ok"
18 }
19
20 burn_fail() {
21     echo 14 > /sys/class/gpio/export
```

将mac_record.txt、mac_current、burn_mac.sh拷贝到SD卡中。

mac_record.txt: 新建一个空的txt, 用于记录烧写的mac地址。

mac_current: 存放起始地址文件, 例如mac_current里写入88:A9:A7:00:77:40, 则会从88:A9:A7:00:77:40开始写入mac地址。

上电后就自动读取SD卡中的脚本, 进行烧录MAC地址。