# ARM 体系结构

内存：

SRAM 静态内存 容量小、价格高，优点是不需要软件初始化上电就能用

DRAM 动态内存 特点就是容量大、价格低，缺点就是需要软件初始化，上电不能直接使用

单片机中：内存需求量少，而且希望开发尽量简单，适合全部用SRAM

嵌入式系统：内存需求量大，而且没有NorFlash等可启动介质

PC机： 内存需求量大，而且软件复杂，不在乎DRAM的初始化开销，适合全部用DRAM

外存：

NorFlash：特点是容量小、价格高，优点是可以和CPU直接总线式相连，CPU上电后可以直接读取，所以用作启动介质。

NandFlash（跟硬盘一样）：特点是容量大、价格低，特点是不能总线式访问，也就是说不能上电CPU直接读取，需要CPU先运行一些初始化软件，然后通过时序接口读写。

所以一般PC机都是：BIOS（NorFlash）+很大容量的硬盘（类似于NandFlash）+大容量的DRAM（内存）

一般的单片机：很小容量的NorFlash +很小容量的SRAM

嵌入式系统：因为NorFlash很贵，所以现在很多的嵌入式系统倾向于不用NorFlash，直接用：外接的大容量Nand+外接大容量DRAM+SoC内置SRAM

S5PV210使用的启动方式是：外接的大容量Nand+外接大容量DRAM+SoC内置SRAM

实际上210的启动方式还要更好玩一些：210内置了一块96KB大小的SRAM（叫iRAM），同时还有一块内置的64KB大小的NorFlash（叫iROM）。210的启动过程大致是：

1. CPU上电后先从内部的IROM中读取预先设置的代码（BL0），执行。这一段IROM代码首先做了一些基本的初始化（CPU时钟、关看门狗。。。）（这一段IROM代码是三星出厂前设置的，三星也不知道我们板子上将来接的是什么样的DRAM，因此这一段IROM是不能负责初始化外接的DRAM的，因此这一段代码只能初始化Soc内部的东西）。然后这一段代码会判断我们选择的启动模式，（我们通过硬件跳线可以更改板子的启动模式），然后从相应的外部存储器去读取第一部分启动代码（BL1，大小为16KB）到内部的SRAM。
2. 从SRAM去运行刚上一步读取来的启动代码（BL1），然后执行，BL1就会负责初始化NandFlash，然后将BL2读取到IRAM（剩余的80KB）然后运行，BL2初始化DRAM，初始化板卡，然后将OS从Nand读取到DRAM，然后启动OS，启动过程结束。

思路：因为启动代码的大小是不定的，有些公司可能96KB就够了，有些公司可能1MB都不够，所以刚才说的2步的启动方式是有漏洞的（SRAM只有96KB）。三星的解决方案是：把启动代码分成两段（BL1和BL2），这两部分协同工作来完成启动

BL0做了什么：

1. 关看门狗
2. 初始化指令cache
3. 初始化栈
4. 初始化堆
5. 初始化块设备复制函数 device copy function
6. 设置SoC时钟系统
7. 复制BL1到内部的IRAM(16KB)
8. 检查BL1的校验和
9. 跳转到BL1去执行

wakeup status 唤醒状态（复位状态）

S5PV210的所有启动

先1st启动，通过OMpin选择启动介质

再2nd启动，从SD2

再Uart启动

再USB启动

ARM有7个基本工作模式

User：非特权模式，大部分任务执行在这种模式

System：使用和User模式相同寄存器集的特权模式

FIQ：当一个高优先级（fast）中断产生时将会进入这种模式

IRQ:当一个低优先级（normal）中断产生时会进入这种模式

Supervisor：当复位或软中断产生时将会进入这种模式

Abort：当存取异常时将会进入这种模式

Undef：当执行未定义指令时会进入这种模式

异步的：靠中断（一种特殊的异常）来实现

同步的：同步时钟

开发板刷系统

1、什么是刷系统

刷系统就是利用刷机工具，向开发版中烧录事先编译好的系统镜像，使之在开发版中运行起来。

2、串口输出的意义（做系统控制台）

串口是一种硬件通信口，很多年前的时候串口是cpu之间进行通信的主要接口，但是现在因为串口通信速率很低，所以现在串口主要用于程序输出监控、调试。

桌面电脑可以打开一个虚拟控制台，嵌入式系统一般用串口来做控制台。一般是用一根串口线连接开发板的串口和我们笔记本电脑的串口，然后在电脑上打开一个串口监视，这样开发板上的串口输出就可以在电脑上看到，还可以通过监视终端向开发板输入一些控制命令

由开发板执行，常用的串口监视软件有：超级终端、SecureCRT、minicom

1. 安装USB转串口线驱动

本来电脑都是有串口的（DB9接口），但是现在大家的笔记本电脑都没有串口了，所以串口连接线用不了了，办法是使用USB转串口线，这种线插入电脑后是需要安装驱动的，安装驱动后会在电脑上形成一个串口（叫USB转串口），这样就相当于你的电脑有了一个串口，可以通过这个串口来监视开发板串口的输出。

注意1：windows对USB设备的管理是和USB口有关的，你每次把usb转串口线插到一个口上，这样COM口号码是不变的，方便我们后期使用。如果每次胡乱更改插口，得到的COM口号会变化。

注意2：COM口号码是可以改的，还可以强制占用显示“已使用”的COM号，一般改成COM4以内就可以了。

1. 使用SecureCRT串口监视软件

下载、解压、注册

打开SecureCRT.exe后，建立一个Serial连接，开始监视串口，注意流控一定要去掉。

然后开机就可以看到串口信息了。

1. 破坏iNand中的bootloader以从SD2启动。

参见第四部分LED实验。

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

然后执行第二条命令sync

解释：这句话的意思就是把板载的iNand的第一个扇区擦出用全0替代。这样我们板载的iNand的bootloader的开始第一个扇区就被破坏了，将来启动时IROM还是会先从iNand中读取前16KB，然后计算校验和，这是因为第一个扇区被擦除了，所以校验和不通过，所以启动失败（会SD2去执行2nd启动）

注意：破坏板载的iNand的bootloader后，不插外部SD卡，启动是串口得到：SD checksum Error

1. 制作启动SD卡（其实就是烧录uboot到SD卡中）

有两种烧写方法：一种是在windows上用刷卡工具去制作SD卡（类似小车系统镜像，只不过这里是uboot镜像）。另一种是在linux中使用dd命令。这里使用第一种。

制作完SD卡后将SD卡插入开发板SD2通道（注意不是SD3），然后开机就可以进入uboot界面了，在uboot开机自动启动倒数三秒之内迅速按下电脑回车键，打断自动启动。（否则会自动启动iNand中的android）

1. fastboot介绍

fastboot是uboot中用来快速下载镜像的一个命令，fastboot同时还是windows的一个软件。

fastboot下载要注意1：fastboot使用USB线进行数据传输，所以fastboot执行之前要先连接开发板和主机之间的usb线

fastboot下载要注意2：需要安装驱动。

1. fastboot常用命令：

fastboot devices 命令用来查看当前连接设备

fastboot flash xxx 命令用来烧录的

fastboot reboot 烧录完成后用来重启系统

在cmd中使用下列命令来烧录

fastboot flash bootloader android4.0/uboot.bin 烧uboot

fastboot flash kernel android4.0/zImage-android 烧linux kernel

fastboot flash system android4.0/x210.img 烧android ROM

uboot的参数设置：set boot 'movi read kernel 30008000; bootm 30008000'(默认就是这个，不设置也行)

开发板刷系统3

1、烧录linux+QT(原有系统存在的情况下不需要烧录SD卡启动，可以直接硬盘装机，直接在SecureCRT重新启动，3s内按下任意按键，进入uboot，输入fastboot，然后cmd 使用fastboot +以下命令进行烧写)

fastboot flash bootloader QT/uboot\_inand.bin 烧uboot

fastboot flash kernel QT/zImage-qt 烧linux kernel

fastboot flash system QT/rootfs\_qt4.ext3 烧rootfs

uboot的参数不用特意设置（因为我们刷了专为linux+QT定制的uboot，这个uboot中 估计已经预制好了所有的启动参数）

2、烧录android2.3

fastboot flash bootloader android2.3/uboot.bin 烧uboot

fastboot flash kernel android2.3/zImage-android 烧linux kernel

fastboot flash system android2.3/x210.img 烧android ROM

1、原理图是电路原理设计图，各个电路中的部件的逻辑连接图

2、PCB图是我们用来制作电路板，并且电路板进行原件焊接，做成最终产品的生产性图纸。

现有原理图设计，根据原理图来设计PCB图，PCB图是对原理图的一种实现，原理图和 PCB图一般都有硬件工程师实现。但是原理图设计时软件工程师会有一定程度参与。PCB 图的设计完全是由硬件工程师实现。

1. 丝印图是PCB图设计中，所有原件的外框的框图。丝印图其实就是整个电路板的实体的位置图纸，与软件和功能无关，与生产制造有关，对软件工程师来说，可以不予理会。

S5PV210数据手册导读

数据手册的正确读法：先通读概览，然后根据需要去查询

学会利用文档目录

利用好pdf文档阅读的搜索功能。

补充-x210的dnw刷机

之前是SD卡刷机，这里讲usb启动刷机（dnw刷机），sd卡刷机简单，dnw较麻烦

有人刷错了uboot，错刷成了nand版本的uboot（还有些确实没有刷错uboot），这时候在uboot中执行了movi write u-boot 0x30000000,再次开机碰到诡异现象：串口输入乱码、SD卡不启动开发板变砖头了。这时候解决思路就是dnw刷机。

刷机准备：

1. 看x210 dnw刷机，按照其中进行操作。
2. 事先把开发板启动方式拨到usb启动。OM5打到VCC就能从usb启动。
3. 两个刷机文件x210\_usb.bin和uboot.bin准备好。

刷机

刷机分两步：第一步刷x210\_usb.bin ，dnw配置下载地址是0xd0020010;第二步刷uboot.bin,dnw刷机地址是ox23e00000。这时串口终端有信息打印出来，回车进入uboot命令行，这时候可以松开开机键。（事先连接上串口，防止来不及按回车进入uboot），uboot启动起来后，先fdisk -c 0 去重新分区，然后再fastboot。。。。。

补充-linux下使用dd命令刷写uboot

SD卡接入linux中

首先：一个USB设备插到电脑后，要么可以在windows中识别，此时不可以在虚拟机linux中识别；要么能在linux中识别，但是在windows中又不见了，默认一般都是连在windows中的，所以在linux中是找不到的

如果你需要将该设备连接到linux中，需要在VMware软件的菜单“虚拟机->可移动设备”中点选该设备，选择连接。

连接进liunx中，在命令行中：ls /dev/sd\*查看，和之前比对，就知道我们的SD卡在linux中的设备编号了，一般是/dev/sdb,也有可能是sdc或者其他。我们后面的实验是基于这个设备号是/dev/sdb的，如果不是，后面的实验要相应的做修改。

制作SD启动卡

执行./nand\_fusing.sh /dev/sdb1

使用SD卡启动

先确定开发板是从SD卡启动的

然后确定板子内的iNand中的uboot是被破坏的，不能启动的

SD卡插入SD2通道，启动即可，启动后，（看有没有必要fdisk -c 0 来分区（分区数要对））使用fastboot继续刷机。

刷机常见问题处理

1、屏幕显示异常，原因是刷成了旧版本的开发板镜像，解决方案就是使用新版本镜像

2、dnw驱动装不上：多试几次，禁用数字签名

3、fastboot驱动装不上：多试几次，windows联网自动安装，用360手机助手之类的工具安装。

4、找不到拨码开关，不知道如何设置USB启动

5、dnw下载uboot.bin时进度条不动，卡住：禁用数字签名+右键管理员权限运行dnw软件，换电脑，使用linux下dnw软件

6、dnw下载uboot.bin后串口没有任何信息输出：可能性1：串口插错了；可能性2：uboot.bin用错了。解决方案：换串口插试一试（正确的是用串口2），用正确的uboot.bin。

7、执行了movi write u-boot 0x30000000 后变砖了：解决方案：dnw刷机。

8、windows不提示安装驱动：如果开发板中就在运行uboot或者linux+QT，这时候插上USB线主机windows也不会发现开发板；当我们uboot下执行了fastboot命令后主机就会提示时装fastboot驱动，当开发版运行了andriod系统后主机就会提示装fastboot驱动；当开发板设置了USB启动并且重启时主机就会提示装dnw驱动。

9、android系统刷机成功后启动不成功：最大的可能性还是镜像用得不对。

linux下dnw软件的使用

linux下dnw的获取

1. 源码包获取
2. 编译和安装

开发板连接到linux下

1. 开发板设置USB启动，开机，windows设备管理器中提示发现新硬件
2. VMware菜单：虚拟机->可移动设备->Samsung S5PC110 SEC Test B/D,点击连接
3. 成功连接后的现象：1、windows下设备管理没了；2、ls /dev/secbulk0 设备节点自动出现

使用dnw进行下载uboot.bin

1. dnw -a 0xd0020010 x210\_usb.bin
2. dnw -a 0x23e00000 uboot.bin

成功标志：SecureCRT中成功看到了uboot的启动信息，并且进入了uboot命令行。

uboot启动后的刷机注意事项：

1. 正确的uboot启动后一定要先fdisk -c 0 进行分区，然后再fastboot
2. uboot的参数中bootcmd和bootargs一定要正确。

set bootcmd 'movi read kernel 30008000;bootm 30008000'

set bootargs console=ttySAC2,115200 root = /dev/mmcblk0p2 rw init=/linuxrc rootfstype=ext3