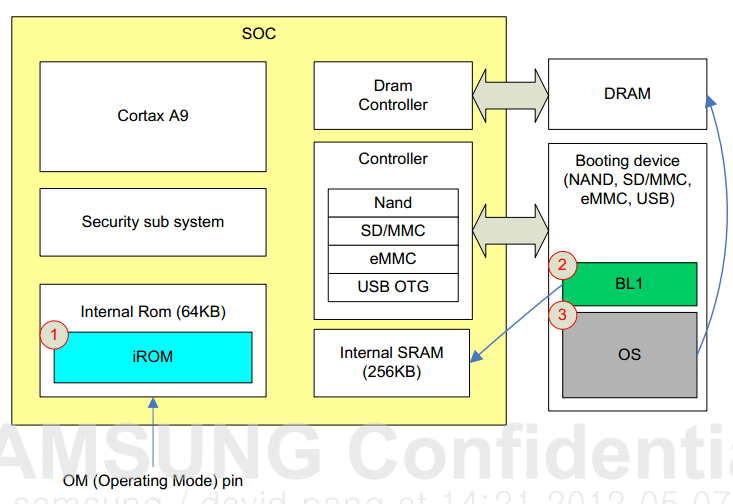
MCU就是单片机

高端的ARM就是MPU

MPU多了两个部件

高速缓存（cache）和 内存管理单元（MMU）



iROM-->BL1--->BL2（uboot前14k）---->uboot---->zImage---->挂接文件系统

IRAM IRAM DDR3

1  Linux把所有设备看成文件    
2  做驱动就是做那几个函数，例如：led\_open()， led\_read() 等等  
3  设备文件 里面包含 主设备号和次设备号  
4  设备文件（节点） 就是 上层应用和底层驱动的桥梁  
5  驱动做好以后，要通过register\_chrdev()注册到系统，这样那几个函数就和设备文件关联起来了

6 上层应用涌过调用系统函数open, read等，再通过设备文件，间接找到led\_open(), led\_read()等并执行

---------------------------------------------------------------------

uboot是个最大的裸机程序

BIOS

一个是‘board' 板级的代码

一个是'CPU' 目录里是对应的不同处理器的初始化代码

还有一个'drivers' 驱动

内核是驱动，uboot怎么也有驱动？？？

UBOOT检测内存，存储等设备，并且设置处理器等设备的时钟等等

u-boot.lds

uboot首先要执行 start.s 中的汇编程序

先执行文件 start.s

再执行cpu\_init.s 这个文件里的汇编，主要功能就是初始化DDR内存mem\_ctrl\_asm\_init\_ddr3。

lowlevel\_init.s

 start.s里调用了 函数 lowlevel\_init，’mem\_ctrl\_asm\_init\_ddr3‘被’lowlevel\_init‘调用

ldr pc, \_start\_armboot，把函数start\_armboot的地址，赋值给pc指针，start\_armboot这个函数是个C函数

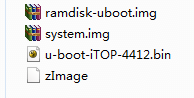
1 uboot其实就是个大程序，大的裸机程序

2 uboot 和Linux内核不一样，他没有多任务调度的代码，虽然可以挂接文件系统，也可以有网络协议栈

3 uboot最初执行的是汇编，集中在那三个汇编文件当中，然后就是C代码了

Linux必须要挂接一个文件系统

Linux是在运行的最后才挂载文件系统



4挂 1和2

4 也就是zImage是内核

1 和 2 就是 Android

在学驱动，学来学去，实际都包含在zImage里面