

基础6-Linux 设备树 (Device Tree)

一、设备树起源

为了减少内核中的板级细节代码垃圾！

二、设备树的组成和结构

2.1、设备树由一系列被命名的节点 (Node) 和属性 (Property) 组成，而节点本身可包含子节点。所谓属性，其实就是成对出现的名

称和值。在设备树中，可描述的信息包括（原先这些信息大多被硬编码在内核中）：
CPU的数量和类别。
内存基地址和大小。
总线和桥。
外设连接。
终端控制器和中断使用情况。
GPIO控制器和GPIO使用情况。
时钟控制器和时钟使用情况。
它基本上就是画一颗电路板上CPU、总线、设备组成的树，Bootloader会将这棵树传递给内核，然后内核可以识别这棵树，并根据它展开出Linux内核中的platform_device、i2c_client、spi_device等设备，而这些设备用到的内存，IRQ等资源，也将传递给了内核，内核会将这些资源绑定给展开的相应的设备。

2.2、DTS、DTC和DTB等

基本上，在ARM Linux中，一个.dts文件对应一个ARM的设备，一般放置在arch/arm/boot/dts目录中。1个dts文件+n个dtsi文件，它们编译而成的dtb文件就是真正的设备树。

2.2.1、DTS

soc厂商会把soc公共的特性和多块开发板公用的特性提炼为dtsi，而dts则负责描述某个具体的产品（开发板）的特性。dts直接或间接的包含多个dtsi（类似于c语言的头文件），就体现了一个完整的产品（开发板）所有的特性。以Ti公司的boneblack为例，其组成为：
[root@localhost dts]# tree

├── aks-cdu.dts
├── am335x-boneblack.dtb
├── am335x-boneblack.dts
├── am335x-bone-common.dtsi
├── am335x-bone.dtb
├── am335x-bone.dts
├── am335x-evm.dtb
├── am335x-evm.dts
├── am335x-evmsk.dtb
├── am335x-evmsk.dts
├── am335x-tester.dtb
├── am335x-tester.dts
└── am33xx.dtsi

chp06 ▶ deviceTree ▶ DTSource3.8.13

名称	修改日期	类型	大小
am33xx.dtsi	2018/1/29 9:38	DTSI 文件	15 KB
am335x-bone.dts	2018/1/29 9:38	DTS 文件	1 KB
am335x-boneblack.dts	2018/1/29 9:38	DTS 文件	2 KB
am335x-bone-common.dtsi	2018/1/29 9:38	DTSI 文件	11 KB
skeleton.dtsi	2018/1/29 9:38	DTSI 文件	1 KB
tps65217.dtsi	2018/1/29 9:38	DTSI 文件	2 KB

2.2.2、DTC (Device Tree Compiler)

DTC是将.dts编译为.dtb的工具。DTC的源代码位于内核的scripts/dtc目录中。

DTC除了可以编译dts文件外，还可以“反汇编”.dtb为.dts文件，其指令格式为：
./scripts/dtc /dtc -I dtb -O dts -o xxx.dts arch/arm/boot/dts/xxx.dtb

2.2.3、DTB (Device Tree Blob)

2.2.4、绑定 (Binding)