设备树

用来描述硬件信息的文件

主要针对ARM架构，x86架构用ACPI来描述硬件资源。

**dts**：device tree source，设备树源码

**dtsi**：device tree source include，设备树源码头文件

**dtc**：device tree compiler，设备树编译器

**dtb**：device tree blob，设备树源码编译后的文件



1、如何编译设备树？

内核源码编译后，在内核源码路径下/scripts/dtc/有一个dtc可执行文件，此为dtc编译器。

编译： **dtc -I dts -O dtb -o xxx.dtb xxx.dts**

反编译：**dtc -I dtb -O dts -o xxx.dts xxx.dtb**

2、设备树语法

标签**:** 设备名称**@**设备地址

如：uart: serial@02288000

同级节点下节点名称不能相同

**/dts-v1/; //**第一行表示dts文件的版本

**/{ //** 根节点 **/**，只能这个名

**uart: serial@02288000{ //** 子节点

**node1\_child{**

**};**

**};**

**led: gpio@22020101{ //** 子节点

**node2\_child{**

**};**

**};**

**};**

**reg属性**

reg = <address1 length1 address2 length2……>;

reg = <0x02200000 0x4000

0x02205000 0x4000

>;

**#address-cells**和**#size-cells**用来描述子节点中的reg信息中的地址和长度信息。

如：

**node1{**

**#address-cells = <1>;**

**#size-cells = <0>;**

**node1\_child{**

**reg = <0>;**

**};**

**};**

**model属性**

如：model = “wm8960-audio”;

model = “this is linux board”;

**compatible属性**

非常重要的属性，用来和驱动进行匹配的，匹配成功之后，会调用驱动中的probe函数

如：compatible = “xvnwei”, “xvnwei-board”;

**status属性**

设备状态，只能为以下值：

okay 设备可用

disabled 设备不可用

fail 设备不可用并检测到了错误

fail-sss 设备不可用并检测到了错误，sss为检测到的错误内容

如：status = “okay”;

**device\_type属性**

只用于描述cpu和memory节点

如：device\_type = “memory”;

·根节点 **/** 下的两个特殊节点：**aliases**和**chosen**

批量定义别名：

**aliases{**

**mmc0 = &sdmmc0;** //第一种写法

**mmc1 = &sdmmc1;**

**mmc2 = &sdhci;**

**serial0 = “/simple@fe000000/serial@11c500”;** //第二种写法

**};**

uboot给内核传递bootargs参数，可以使用chosen节点设置bootargs参数

**chosen{**

**bootargs = “root=/dev/nfs rw nfsroot=192.168.1.1 console=ttyS0,115200”;**

**}；**