# **设备树(DTS)是如何被Linux内核识别并最终与驱动匹配的整个过程：**

## 1启动阶段解析

Bootloader -> DTB -> 内核解析DTB -> device\_node树

Bootloader将DTB文件加载到内存特定位置内核启动时，通过 unflatten\_device\_tree() 解析DTB将DTB转换成内核内部的 device\_node 树状结构

## 2 device\_node 转换为 platform\_device

// 关键函数

of\_platform\_populate() // 遍历 device\_node 树

of\_platform\_device\_create() // 创建 platform\_device

遍历 device\_node 树中的每个节点

对符合条件的节点创建对应的 platform\_device

platform\_device 包含了设备资源信息(寄存器、中断等)

## 3 驱动与设备匹配

struct of\_device\_id {

char name[32];

char type[32];

char compatible[128];

const void \*data;

};

// 驱动中声明匹配表

static const struct of\_device\_id xxx\_of\_match[] = {

{ .compatible = "vendor,device" },

{ }

};

MODULE\_DEVICE\_TABLE(of, xxx\_of\_match);

匹配过程：

驱动通过 compatible 属性声明支持的设备

内核使用设备树节点的 compatible 属性查找驱动

找到匹配时，调用驱动的 probe 函数