

# 嵌入式系统工程师



# 设备模型

- 设备模型概述
- 设备模型组件
- 实例分析platform总线匹配过程

- 设备模型概述
- 设备模型组件
- 实例分析platform总线匹配过程

- 在2.5之前内核并没有一个统一的模型来管理这些设备，但随着Linux下的设备越来越多，拓扑结构也越来越复杂，一个能够对设备进行有效的统一管理就显得异常重要了
- 建立设备模型的目的就是要在内核下建立一个统一的、全局的设备树，方便设备管理
- 回顾I2C子系统

- 设备模型
- 设备模型组件
- 实例分析platform总线匹配过程

- 设备驱动程序模型建立在几个基本数据结构上，这些结构描述了总线、设备、设备驱动以及他们之间的关系
- **struct device**  
Linux系统中的每个设备由一个**struct device**代表
- **struct device\_driver**  
系统中的每个驱动程序由一个**device\_driver**对象描述
- **struct bus\_type**  
系统中总线由**struct bus\_type**描述

```
struct bus_type {  
    const char      *name;  
  
    struct bus_attribute    *bus_attrs;  
    struct device_attribute *dev_attrs;  
    struct driver_attribute *drv_attrs;  
  
    int (*match)(struct device *dev,  
                 struct device_driver *drv);  
    int (*uevent)(struct device *dev,  
                 struct kobj_uevent_env *env);  
    int (*probe)(struct device *dev);  
    int (*remove)(struct device *dev);  
    void (*shutdown)(struct device *dev);  
  
    . . . . .  
    struct bus_type_private *p;  
};
```

通过bus\_type这个结构体我们就可以自己创建一条总线

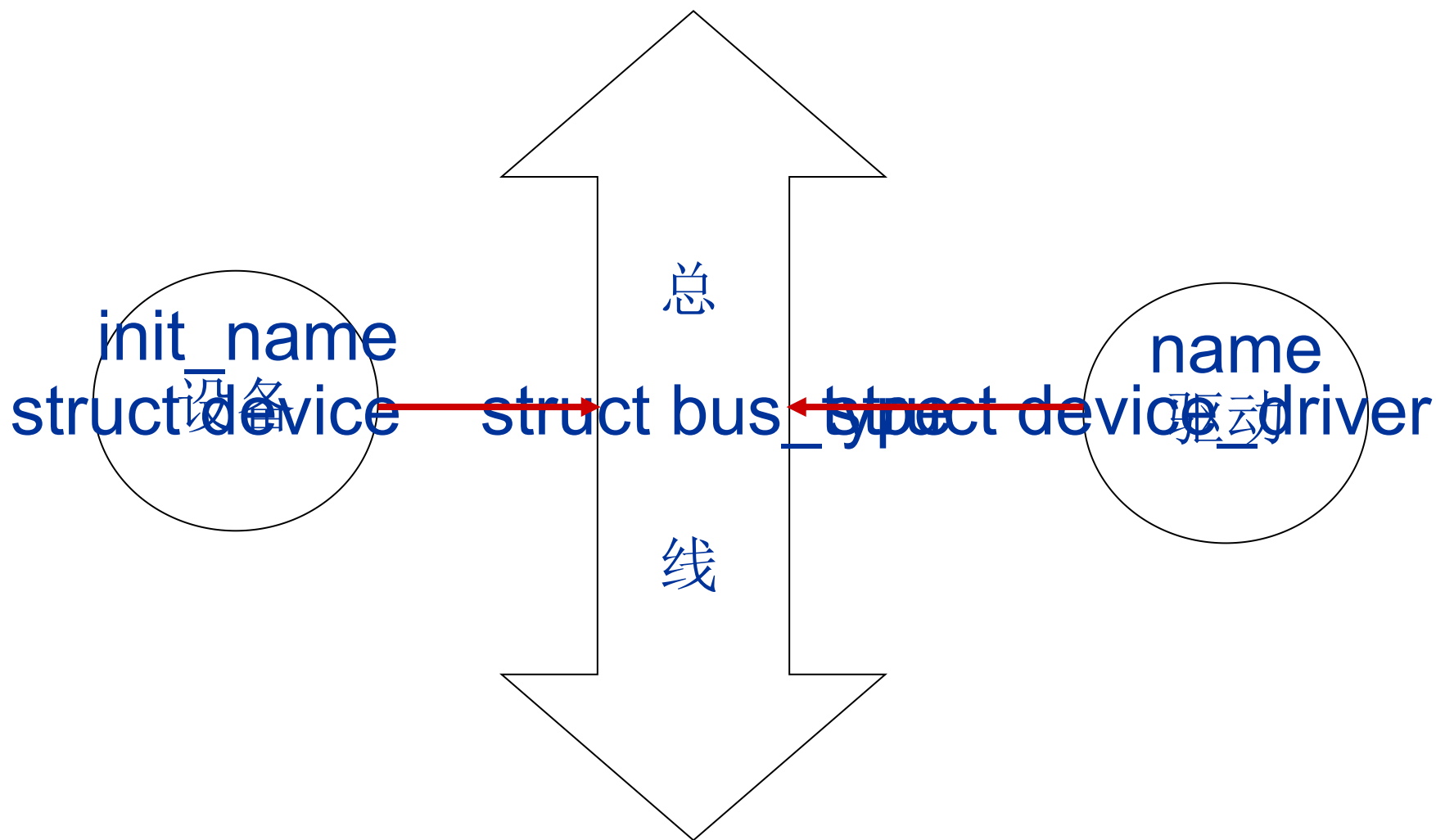


```
struct device {  
    struct device      *parent;  
  
    struct device_private  *p;  
  
    struct kobject kobj;  
    const char      *init_name;  
    struct device_type  *type;  
  
    struct mutex      mutex;  
  
    struct bus_type *bus;  
    struct device_driver *driver;  
  
    .....  
    void      (*release)(struct device *dev);  
};
```

这个结构体就表示我们具体的一个硬件设备

```
struct device_driver {  
    const char      *name;  
    struct bus_type *bus;  
    struct module   *owner;  
    const char      *mod_name;  
    bool suppress_bind_attrs;  
#if defined(CONFIG_OF)  
    const struct of_device_id *of_match_table;  
#endif  
    int (*probe) (struct device *dev);  
    int (*remove) (struct device *dev);  
    . . . . .  
};
```

通过这个结构体代表我们的驱动程序



- 总结:
  - 内核对linux下的所有设备是通过设备模型进行管理的
  - 整个Linux的设备模型是一个oo体系结构，总线、设备、驱动都是对象，由这些基本对象构建了丰富多彩的内核
- 参考 `bus_device_driver`

- 设备模型
- 设备模型组件
- 实例分析platform总线匹配过程

- 创建platform总线 (platform. c)
  - struct bus\_type
  - bus\_register
  
- 创建platform设备
  - struct device
  - device\_register
  
- 创建platform设备驱动
  - struct device\_driver
  - driver\_register

## ➤ platform设备的创建过程

platform\_device\_register

->device\_initialize

->platform\_device\_add

->device\_add

->bus\_probe\_device

->device\_attach

->bus\_for\_each\_drv

->\_\_device\_attach

->platform\_drv\_probe

->button\_probe



## ➤ platform设备驱动的创建过程

platform\_driver\_register

->driver\_register

->bus\_add\_driver

->driver\_attach

->bus\_for\_each\_dev

->\_\_driver\_attach

->driver\_probe\_device

->platform\_driver\_probe

->button\_probe





凌阳教育官方微信：Sunplusedu

Tel: 400-705-9680, BBS: [www.51develop.net](http://www.51develop.net), QQ群: 241275518

