

# 迅为电子Linux驱动教程

## ——驱动注册

北京迅为电子有限公司





# 本节目标

- 驱动注册
  - 很重要，牢牢掌握，以后写任何Linux驱动都会用到
- 实验
  - 在mini\_linux\_module的基础上添加驱动注册部分



## 驱动注册 ——头文件

- 驱动注册使用结构体platform\_driver，该结构体在头文件“vim include/linux/platform\_device.h”中
- 驱动注册platform\_driver\_register，驱动卸载函数platform\_driver\_unregister也在这个头文件中
  - 这两个函数的参数都只有结构体platform\_driver



# 驱动注册

## ——注册结构体

```
struct platform_driver {  
    int (*probe)(struct platform_device *);  
    int (*remove)(struct platform_device *);  
    void (*shutdown)(struct platform_device *);  
    int (*suspend)(struct platform_device *, pm_message_t state);  
    int (*resume)(struct platform_device *);  
    struct device_driver driver;  
    const struct platform_device_id *id_table;  
};  
  
extern int platform_driver_register(struct platform_driver *);  
extern void platform_driver_unregister(struct platform_driver *);
```

119,8

44%



# 驱动注册

## ——注册结构体

- 驱动常见的几种状态，初始化，移除，休眠，复位
  - 就像PC一样，有的驱动休眠之后无法使用，有的可以使用；有的系统唤醒之后，驱动需要重新启动才能正常工作，也有直接就可以使用等等
- probe函数
  - platform\_match函数匹配之后，驱动调用的初始化函数
- remove函数
  - 移除驱动函数
- suspend函数
  - 悬挂（休眠）驱动函数



# 驱动注册

## ——注册结构体

- resume函数
  - 休眠后恢复驱动
- device\_driver数据结构的两个参数
  - name和注册的设备name要一致
  - owner一般赋值THIS\_MODULE



# 实验

- 在mini\_linux\_module的基础上添加驱动注册部分
- 编译，在开发板上加载和卸载驱动



谢谢！