华清远见 FARSIGHT 嵌入式培训专家

嵌入式学院 华清远见旗下品牌

Platform设备驱动浅析

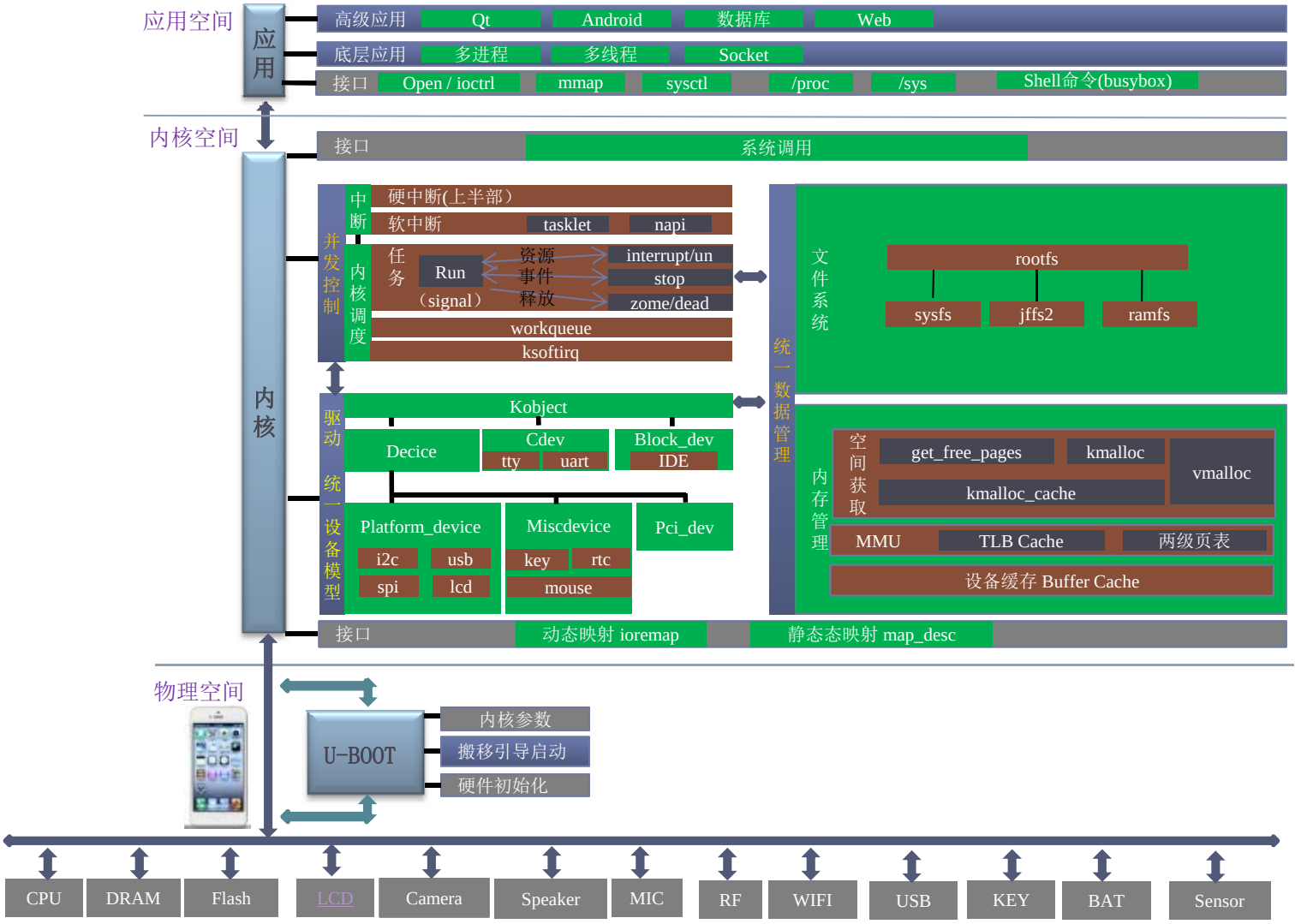
袁祖刚

内容提纲

嵌入式学院
华清远见旗下品牌

- 驱动是如何管理设备的
- 设备 驱动 总线
- 添加注册platform设备
- 实例

架构


应用空间
应用
高级应用
Qt
Android
数据库
Web
底层应用
多进程
多线程
Socket
接口
Open / ioctrl
mmap
sysctl
/proc
/sys
Shell命令(busybox)
内核空间
接口
系统调用
内核
并发控制
中断
硬中断(上半部)
软中断
tasklet
napi
内核调度
任务
Run
（signal）
资源
事件
释放
interrupt/un
stop
zome/dead
workqueue
ksoftirq
驱动统一设备模型
Kobject
Decice
Cdev
tty
uart
Block_dev
IDE
Platform_device
i2c
usb
spi
lcd
Miscdevice
key
rtc
mouse
Pci_dev
统一数据管理
文件系统
rootfs
sysfs
jffs2
ramfs
内存管理
空间获取
get_free_pages
kmalloc
vmalloc
kmalloc_cache
MMU
TLB Cache
两级页表
设备缓存 Buffer Cache
接口
动态映射 ioremap
静态态映射 map_desc
物理空间
U-BOOT
内核参数
搬移引导启动
硬件初始化
CPU
DRAM
Flash
LCD
Camera
Speaker
MIC
RF
WIFI
USB
KEY
BAT
Sensor

驱动是如何管理设备的

嵌入式学院
华清远见旗下品牌

```
struct platform_device {
    const char  * name;
    u32     id;
    struct device   dev;
    u32     num_resources;
    struct resource * resource;
}
struct resource {
    const char *name;
    unsigned long start, end;
    unsigned long flags;
    struct resource *parent, *sibling, *child;
};
```

类似面向对象中的继承

平台设备把板子看成一个设备，它包含了许多资源设备。 并建立虚拟总线，把driver和device自动匹配起来。
它实现了主机和驱动相分离

```
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;
};
```

```
struct  bus_type   platform_bus_type  =  {   //虚拟总线
    .name  =  “platform”,
    .match =     platform_match,
};
static int platform_match(struct device * dev, struct device_driver * drv)
{
    struct platform_device *pdev =
                                container_of(dev, struct platform_device, dev);
    //当设备名和驱动名相同则匹配成功
     return (strncmp(pdev->name, drv->name, BUS_ID_SIZE) == 0);
}
```

设备 驱动 总线

嵌入式学院
华清远见旗下品牌

```
struct device {
    struct klist        klist_children;
    struct klist_node   knode_parent;
    struct klist_node   knode_driver;
    struct klist_node   knode_bus;
    struct device    * parent;

    struct kobject kobj;
    char    bus_id[BUS_ID_SIZE
    struct bus_type * bus;
    struct device_driver *driver;
    void        *driver_data;
   ……
};
```

```
struct device_driver {
    const char      * name;
    struct bus_type     * bus;
    ……
    struct module       * owner;

    int (*probe)    (struct device * dev);
    int (*remove)   (struct device * dev);
    void    (*shutdown) (struct device * dev);
    int (*suspend)  (struct device * dev);
    int (*resume)   (struct device * dev, u32 level);
};
```

用Device描述设备属性，
Driver描述对设备控制行为。
Bus连接二者建立通讯。

```
struct bus_type {
    const char      * name;
    struct kset         drivers;
    struct kset         devices;
    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     (*hotplug) (struct device *dev);
};
```

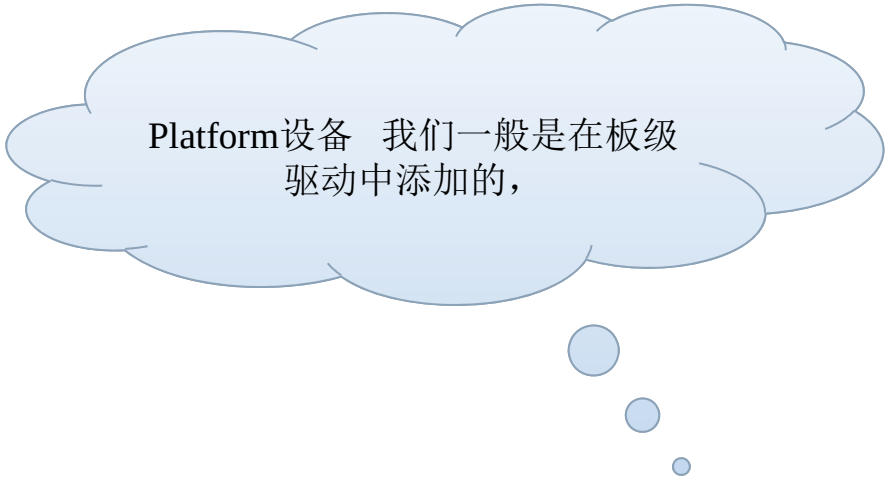
添加注册platform设备

嵌入式学院
华清远见旗下品牌

```
static struct platform_device *smdk2410_devices[] __initdata = {
                &s3c_device_lcd,
                &s3c_device_wdt,
                &s3c_device_i2c0,
                &s3c_device_iis,
};
static struct resource s3c_lcd_resource[] = {
                [0] = {
                    .start = S3C24XX_PA_LCD,
                    .end   = S3C24XX_PA_LCD + S3C24XX_SZ_LCD - 1,
                    .flags = IORESOURCE_MEM,
                },
                [1] = {
                    .start = IRQ_LCD,
                    .end   = IRQ_LCD,
                    .flags = IORESOURCE_IRQ,
                }
};
static u64 s3c_device_lcd_dmamask = 0xffffffffUL;
struct platform_device s3c_device_lcd = {
                .name                           = "s3c2410-lcd",
                .id                             = -1,
                .num_resources      = ARRAY_SIZE(s3c_lcd_resource),
                .resource           = s3c_lcd_resource,
                .dev            = {
                    .dma_mask               = &s3c_device_lcd_dmamask,
                    .coherent_dma_mask = 0xffffffffUL
                }
};
EXPORT_SYMBOL(s3c_device_lcd);
```

```
static void __init smdk2410_init(void)
{
   platform_add_devices(smdk2410_devices,ARRAY_SIZE(smdk2410_devices));
    smdk_machine_init();
}

MACHINE_START(SMDK2410, "SMDK2410")
                .phys_io        = S3C2410_PA_UART,
                .boot_params    = S3C2410_SDRAM_PA + 0x100,
                .map_io         = smdk2410_map_io,
                .init_irq       = s3c24xx_init_irq,
                .init_machine   = smdk2410_init,
                .timer = &s3c24xx_timer,
MACHINE_END
```



实例

嵌入式学院
华清远见旗下品牌

Device设备的类别描述：

1. 平台设备

```
struct   resource   s3c_lcd_resource[]={
   [0]={
      .start = S3C24XX_PA_LCD,
      .end  = S3C24XX_PA_LCD + S3C24XX_SZ_LCD -1,
      .flags= IORESOURCE_MEM,
   },
   [1]={
      .start  = IRQ_LCD,
      .end   = IRQ_LCD,
      .flags = IORESOURCE_IRQ,
   }
};
struct   platform_device    s3c_device_lcd ={
   .name         ="s3c2410-lcd",
   .id             =-1,
   .num_resources = ARRAY_SIZE(s3c_lcd_resource),
   .resource          = s3c_lcd_resource,
};
```

2. 显示设备(framebuffer)

```
struct fb_info{
   struct fb_pixmap pixmap;    /*图像硬件mapper* /
   struct fb_cmap cmap;        /*当前的颜色表*/
   struct fb_videomode *mode;   /*当前的显示模式*/
   struct fb_ops   *fbops;
   struct device *dev;
}
staticstruct fb_ops my2440fb_ops=
{
   .owner          = THIS_MODULE,
   .fb_set_par     = my2440fb_set_par,/*设置fb_info中的参数*/
   .fb_setcolreg  = my2440fb_setcolreg,/*设置颜色表*/
};
```

Driver 设备驱动：

```
struct   platform_driver    lcd_fb_driver=
{
   .probe      = lcd_fb_probe,            /*FrameBuffer设备探测* /
   .remove    = __devexit_p(lcd_fb_remove),/*FrameBuffer设备移除*/
   .suspend  = lcd_fb_suspend,          /*FrameBuffer设备挂起*/
   .resume    = lcd_fb_resume,          /*FrameBuffer设备恢复*/
   .driver      =
   {
      .name ="s3c2410-lcd",
      .owner = THIS_MODULE,
   },
};
```

接口与实现相分离，与c++中虚类类似。

就像公司先把框架搭起来，明确各岗位的职能，人可以后续再招聘

Bus总线通讯方式： 平台虚拟总线platform_bus,它会匹配device和driver

LCD有两种设备描述，表示设备有两项职能。

这就像公司里某人是科长，同时又兼任项目经理是一样的。