东月之神

在单纯的观念里面，生命就容易变得比较深刻！

- 目录视图
- 摘要视图
- 订阅

专家坐阵，Javascript实战分享 微信开发学习路线高级篇上线 免费公开课平台正式上线啦 有奖征文：云服务器使用初体验

初探linux子系统集之led子系统(三)

分类： 初探linux子系统集 2014-07-15 19:20 1203人阅读 评论(7) 收藏 举报

世界杯结束了，德国战车夺得了大力神杯，阿根廷最终还是失败了。也许3年，5年，或者10年后，人们就不知道巴西世界杯的亚军是谁，但是总是会记得冠军是谁。就像什么考试，比赛，第一永远会被人们所记住，所以我们都想去追寻第一，渴望第一，在一次次的追寻中，成者为王败者为寇。而处在第一的位置，永远担心下面的会超越自己，从而活得很累，而第二永远想争取第一，也活得很累，有时候，想想，人一生当中，成功真的就那么重要吗？富有真的那么重要吗？采菊东篱下，悠然见南山不是也很有诗意吗？说了好多，还是继续写led子系统吧。

前面写了很多关于led子系统的相关知识，现在终于可以开始分析leds-gpio.c这个驱动了。

注册了platform驱动。

platform_driver_register(&gpio_led_driver);

platform总线就不多说了，在自己的平台下添加platform device就可以了。

当device和dirver匹配后，就会调用driver的probe函数，这里调用的是下面这个函数。

```
static int __devinit gpio_led_probe(struct platform_device *pdev)
{
        structgpio_led_platform_data *pdata = pdev->dev.platform_data;
        struct gpio_leds_priv*priv;
        int i, ret = 0;

        if (pdata &&pdata->num_leds) {
                priv =kzalloc(sizeof_gpio_leds_priv(pdata->num_leds),
                                GFP_KERNEL);
                if (!priv)
                        return-ENOMEM;

                priv->num_leds= pdata->num_leds;
                for (i = 0;i < priv->num_leds; i++) {
                        ret= create_gpio_led(&pdata->leds[i],
                                             &priv->leds[i],
                                             &pdev->dev,pdata->gpio_blink_set);
                        if(ret < 0) {
                                /*On failure: unwind the led creations */
```

```
                                        for(i = i - 1; i >= 0; i--)
                                                delete_gpio_led(&priv->leds[i]);
                                        kfree(priv);
                                        returnret;
                                }
                        }
                } else {
                        priv =gpio_leds_create_of(pdev);
                        if (!priv)
                                return-ENODEV;
                }

                platform_set_drvdata(pdev,priv);

                return 0;
}
```

获取platform里的device的数据，然后create_gpio_led，这里可以注册很多歌led，具体的leds-gpio的platform数据可以参考

http://blog.csdn.net/eastmoon502136/article/details/37569789。

接着看一下create_gpio_led这个函数。

```
static int __devinit create_gpio_led(const struct gpio_led*template,
                struct gpio_led_data*led_dat, struct device *parent,
                int (*blink_set)(unsigned,int, unsigned long *, unsigned long *))
{
                int ret, state;

                led_dat->gpio = -1;

                /* skip leds thataren't available */
                if(!gpio_is_valid(template->gpio)) {
                        printk(KERN_INFO"Skipping unavailable LED gpio %d (%s)\n",
                                        template->gpio,template->name);
                        return 0;
                }

                ret =gpio_request(template->gpio, template->name);
                if (ret < 0)
                        return ret;

                led_dat->cdev.name= template->name;
                led_dat->cdev.default_trigger= template->default_trigger;
                led_dat->gpio =template->gpio;
                led_dat->can_sleep= gpio_cansleep(template->gpio);
                led_dat->active_low= template->active_low;
                led_dat->blinking =0;
                if (blink_set) {
                        led_dat->platform_gpio_blink_set= blink_set;
                        led_dat->cdev.blink_set= gpio_blink_set;
                }
                led_dat->cdev.brightness_set= gpio_led_set;
                if(template->default_state == LEDS_GPIO_DEFSTATE_KEEP)
                        state =!!gpio_get_value(led_dat->gpio) ^ led_dat->active_low;
                else
                        state =(template->default_state == LEDS_GPIO_DEFSTATE_ON);
                led_dat->cdev.brightness= state ? LED_FULL : LED_OFF;
                if(!template->retain_state_suspended)
```

```
                led_dat->cdev.flags|= LED_CORE_SUSPENDRESUME;

        ret =gpio_direction_output(led_dat->gpio, led_dat->active_low ^ state);
        if (ret < 0)
                goto err;

        INIT_WORK(&led_dat->work,gpio_led_work);

        ret =led_classdev_register(parent, &led_dat->cdev);
        if (ret < 0)
                goto err;

        return 0;
err:
        gpio_free(led_dat->gpio);
        return ret;
}
```

```
struct gpio_led_data {
        struct led_classdevcdev;
        unsigned gpio;
        struct work_structwork;
        u8 new_level;
        u8 can_sleep;
        u8 active_low;
        u8 blinking;
        int(*platform_gpio_blink_set)(unsigned gpio, int state,
                        unsignedlong *delay_on, unsigned long *delay_off);
};
```

申请gpio，以及对于一些变量和函数指针的赋值，最后注册led设备。

关于应用层的调用:

比如我们在platform设备中注册了

```
Static struct gpio_led gpio_leds[] = {
        {
                .name=”my-led”,
                .default_trigger= “timer”,
                .gpio= 30,
                .active_low= 1,
                .default_state= LEDS_GPIO_DEFSTATE_OFF,
}
};
```

那么在/sys/class/leds/下会有my-led目录，在目录下面会创建两个文件delay_on和delay_off。

可以通过

echo 100 > /sys/class/leds/my-led/delay_on

echo 100 > /sys/class/leds/my-led/delay_off

来控制闪烁的时间。

cat /sys/class/leds/my-led/delay_on

cat /sys/class/leds/my-led/delay_off

来获取当前的delay_on和delay_off的值

对于led子系统就简单的介绍到这里了。

版权声明：本文为博主东月之神原创文章，未经博主允许不得转载。

- 上一篇初探linux子系统集之led子系统(二)
- 下一篇初探linux子系统集之i2c子系统(一)

顶
0

踩
0

猜你在找

查看评论
* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

个人资料

eastmoon502136
- 访问：326851次
- 积分：4115
- 等级：
- 排名：第3840名
- 原创：119篇
- 转载：0篇
- 译文：0篇
- 评论：219条

个性签名

别驻足，梦想要不停追逐；别认输，熬过黑夜才有日出。要记住，成功就在下一步；路很苦，汗水是最美的书！

文章搜索

搜索

文章分类

- android(13)
- linux(21)
- 证券投资(6)
- linux总线驱动(19)
- OK6410(arm11)(9)
- 单片机(1)
- c(7)
- c++(1)
- 网络(3)
- 数据结构(2)
- 算法(3)
- 人生顿悟(6)
- 电子小玩意(1)
- 深入学习国嵌实验(10)
- linux内核源码0.11学习摘录(4)
- 产品(6)
- 杂感(2)
- 初探linux子系统集(5)

文章存档

- 2015年07月(1)
- 2015年05月(2)
- 2014年07月(6)
- 2014年03月(1)
- 2013年12月(7)
- 2013年11月(1)
- 2013年08月(2)
- 2013年07月(2)
- 2013年06月(2)
- 2013年05月(1)
- 2013年04月(3)
- 2013年03月(11)
- 2013年02月(4)
- 2013年01月(11)
- 2012年12月(2)
- 2012年11月(5)
- 2012年10月(12)
- 2012年09月(7)
- 2012年08月(16)
- 2012年07月(16)
- 2012年06月(7)

阅读排行

- 和菜鸟一起学linux之bluez学习记录1(51429)
- 和菜鸟一起学linux之wifi学习记录(14127)

- 和菜鸟一起学电子小玩意之四轴飞行器原理(13001)
- 和菜鸟一起学linux之V4L2摄像头应用流程(10208)
- 和菜鸟一起学android4.0.3源码之硬件gps简单移植(8974)
- 和菜鸟一起学linux之DBUS基础学习记录(7661)
- 和菜鸟一起学android4.0.3源码之红外遥控器适配(7354)
- 和菜鸟一起学android4.0.3源码之USB wifi移植心得(6734)
- 和菜鸟一起学android4.0.3源码之bluetooth移植心得(6338)
- 和菜鸟一起学android4.0.3源码之wifi的简单分析(6029)

评论排行

- 和菜鸟一起学android4.0.3源码之wifi的简单分析(24)
- 和菜鸟一起学android4.0.3源码之bluetooth移植心得(21)
- 和菜鸟一起学linux之wifi学习记录(19)
- 和菜鸟一起学android4.0.3源码之硬件gps简单移植(12)
- 和菜鸟一起学linux之V4L2摄像头应用流程(11)
- 和菜鸟一起学android4.0.3源码之wifi direct的简单分析(11)
- 和菜鸟一起学android4.0.3源码之鼠标光标绘制简略版(9)
- 和菜鸟一起学android4.0.3源码之USB wifi移植心得(8)
- 和菜鸟一起学OK6410之最简单字符驱动(7)
- 和菜鸟一起学linux之dlna的学习记录(7)

推荐文章

- * 美团Android DEX自动拆包及动态加载简介
- *Android实战技巧之四十四：Hello,Native!
- * 如何面试Python后端工程师？
- *Android自定义控件之仿汽车之家下拉刷新
- *【android】音乐播放器之service服务设计
- *【FastDev4Android框架开发】Android MVP开发模式详解(十九)