

## iTOP-4412-驱动-设备注册分析

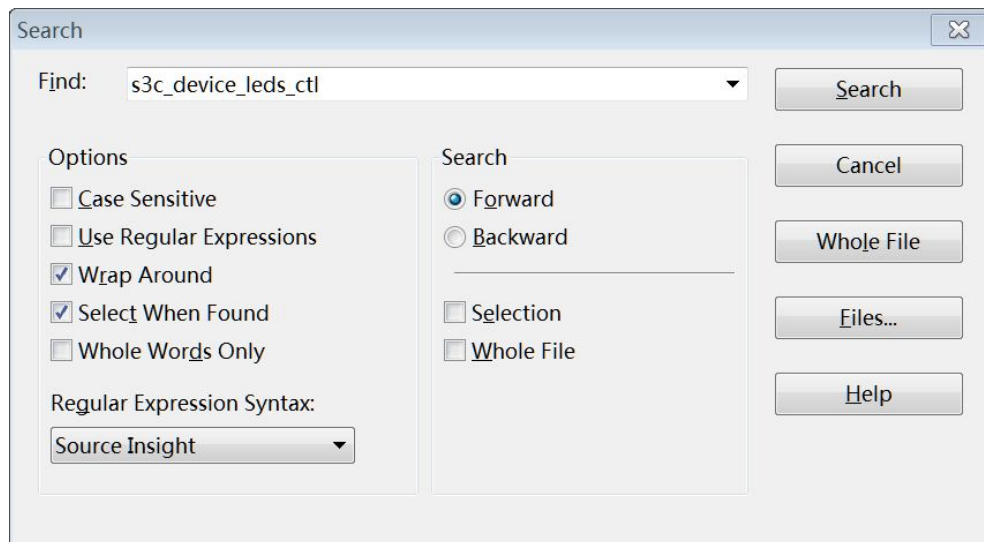
本文档主要介绍 4412 内核设备注册的过程。

在驱动实验手册“实验 06 设备注册”章节，我们把设备注册到了平台文件上，但设备具体是怎么挂到平台文件上的呢？这次我们要探究一下注册设备的结构体“platform\_device”的来龙去脉。

### 1.1 平台文件注册设备分析

打开“source insight”软件，并加载好 4412 内核源码（可参考“iTOP-4412 开发板之精英版使用手册\_V2.6”中的“3.5.2 使用 Source Insight 查看内核代码”章节）。

打开平台文件——arch/arm/mach-exynos/mach-itop4412.c。按“ctrl+f”按键，打开搜索对话框。输入“s3c\_device\_leds\_ctl”，如下图。点击“Search”



```
struct platform_device s3c_device_gps = {
    .name = "si_gps",
    .id = -1,
};

#ifdef CONFIG_MAX485_CTL
struct platform_device s3c_device_max485_ctl = {
    .name = "max485_ctl",
    .id = -1,
};
#endif

#ifdef CONFIG_LEDS_CTL
struct platform_device s3c_device_leds_ctl = {
    .name = "leds",
    .id = -1,
};
#endif

#ifdef CONFIG_BUZZER_CTL
struct platform_device s3c_device_buzzer_ctl = {
    .name = "buzzer_ctl",
    .id = -1,
};
#endif
```

如上图所示，是平台文件中定义的 LEDS 设备注册的结构体，再次搜索。

```
    &s3c_device_adc,
#endif
    &exynos4_busfreq,
#ifdef CONFIG_SWITCH_GPIO
    &headset_switch_device,
#endif
    &s3c_device_gps,
    //&smm6260_modem,

#ifdef CONFIG_MAX485_CTL
    &s3c_device_max485_ctl,
#endif

#ifdef CONFIG_LEDS_CTL
    &s3c_device_leds_ctl,
#endif

#ifdef CONFIG_BUZZER_CTL
    &s3c_device_buzzer_ctl,
#endif

#ifdef CONFIG_ADC_CTL
    &s3c_device_adc_ctl,
#endif
```

可以查看到，如下图所示，是 LEDS 设备注册结构体的引用。

```
#ifdef CONFIG_LEDS_CTL
    &s3c_device_leds_ctl,
#endif
```

该引用在结构体数组\*smdk4x12\_devices[]中，该结构体数组中引用了大量的设备注册的结构体。

```
static struct platform_device *smdk4x12_devices[] __initdata = {

};
```

```
#if defined (CONFIG_S5P_SYSTEM_MMU) && defined(CONFIG_TC4_GB)
extern struct platform_device s5p_device_sysmmu;
#endif
static struct platform_device *smdk4412_devices[] __initdata = {
    &s3c_device_adc,
};

static struct platform_device *smdk4x12_devices[] __initdata = {
#ifdef CONFIG_ANDROID_PMEM
    &pmem_device,
    &pmem_gpu1_device,
#endif
    /* Samsung Power Domain */
    &exynos4_device_pd[PD_MFC],
    &exynos4_device_pd[PD_G3D],
    &exynos4_device_pd[PD_LCD0],
    &exynos4_device_pd[PD_CAM],
    &exynos4_device_pd[PD_TV],
    &exynos4_device_pd[PD_GPS],
    &exynos4_device_pd[PD_GPS_ALIVE],
#ifdef CONFIG_VIDEO_EXYNOS_FIMC_IS
    &exynos4_device_pd[PD_ISP],
#endif
};
```

搜索关键词 “smdk4x12\_devices” 找到如下信息，可以看到 platform\_add\_devices() 设备注册函数调用了结构体数组 \*smdk4x12\_devices[]，对设备进行注册。

```
#endif
#ifdef CONFIG_TC4_GB
    exynos_sysmmu_init();
#endif
    smdk4x12_gpio_power_init();

    platform_add_devices(smdk4x12_devices, ARRAY_SIZE(smdk4x12_devices));
    if (soc_is_exynos4412())
        platform_add_devices(smdk4412_devices, ARRAY_SIZE(smdk4412_devices));

#ifdef CONFIG_FB_S3C
    exynos4_fimd0_setup_clock(&s5p_device_fimd0.dev, "mout_mpll_user",
        800 * MHZ);
#endif
#ifdef CONFIG_S3C64XX_DEV_SPI
    if 0 //remove by cym 20130529
        sclk = clk_get(spi0_dev, "dout spi0");
```

发现 “smdk4x12\_devices” 是 “platform\_add\_devices()” 中的参数，在 “platform\_add\_devices” 上右键，选择 “Jump To Definition”，跳转到如下界面（跳转到的文件是：driver/base/platform.c）。

```
int platform_get_irq_byname(struct platform_device *dev, const char *name)
{
    struct resource *r = platform_get_resource_byname(dev, IORESOURCE_IRQ,
                                                       name);

    return r ? r->start : -ENXIO;
}
EXPORT_SYMBOL_GPL(platform_get_irq_byname);

/**
 * platform_add_devices - add a numbers of platform devices
 * @devs: array of platform devices to add
 * @num: number of platform devices in array
 */
int platform_add_devices(struct platform_device **devs, int num)
{
    int i, ret = 0;

    for (i = 0; i < num; i++) {
        ret = platform_device_register(devs[i]);
        if (ret) {
            while (--i >= 0)
                platform_device_unregister(devs[i]);
            break;
        }
    }

    return ret;
}
```

“platform\_add\_devices()” 函数定义如下所示，循环调用

“platform\_device\_register” 设备注册的函数，很明显这个函数也是可以单独使用。

```
int platform_add_devices(struct platform_device **devs, int num)
{
    int i, ret = 0;

    for (i = 0; i < num; i++) {
        ret = platform_device_register(devs[i]);
        if (ret) {
            while (--i >= 0)
                platform_device_unregister(devs[i]);
            break;
        }
    }

    return ret;
}

EXPORT_SYMBOL_GPL(platform_add_devices);
```

“platform\_add\_devices()” 的功能是向 linux 模型中添加多个平台设备。通过调用

“platform\_device\_register()” 实现向内核添加一个平台设备 devs[i]。

由此可以分析出调用关系：

```
s3c_device_leds_ctl-->*smdk4x12_devices[-->platform_add_devices(-->platform_device_register()).
```

我们也可以直接使用 “platform\_device\_register()” 来注册设备。

后面我们提供一个例程，在 module 中调用该函数，来注册一个设备。

## 1.2 例程

platform\_device\_register()注册设备的测试程序代码如下。

```
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>
#include <linux/platform_device.h>
#include <linux/fb.h>
#include <linux/backlight.h>
#include <linux/err.h>
#include <linux/pwm.h>
#include <linux/slab.h>
#include <linux/miscdevice.h>
#include <linux/delay.h>
#include <linux/gpio.h>
#include <mach/gpio.h>
#include <plat/gpio-cfg.h>

static void leds_release(struct device * dev)
{
}

static struct platform_device leds_device = {
    .name          = "my_code_led",
    .id            = -1,
    .dev = {
        .release = leds_release, ////必须向内核提供一个 release 函数,否则释放的时候会报错
    },
};

static int __init iTop4412_platform_add_device_init(void)
{
}
```

```

return platform_device_register(&leds_device);
}

static void __exit iTop4412_platform_add_device_exit(void)
{
    platform_device_unregister(&leds_device);
}

```

把驱动程序 “platform\_device\_test.c” 和 Makefile 文件上传到同一目录，执行 “make” 命令编译驱动。如下图。

```

root@ubuntu:/home/topeet/neo/device# ls
Makefile  platform_device_test.c
root@ubuntu:/home/topeet/neo/device# make
make -C /home/topeet/android4.0/iTop4412_Kernel_3.0 M=/home/topeet/neo/device modules
make[1]: Entering directory `/home/topeet/android4.0/iTop4412_Kernel_3.0'
CC [M] /home/topeet/neo/device/platform_device_test.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/topeet/neo/device/platform_device_test.mod.o
LD [M] /home/topeet/neo/device/platform_device_test.ko
make[1]: Leaving directory `/home/topeet/android4.0/iTop4412_Kernel_3.0'
root@ubuntu:/home/topeet/neo/device# ls
Makefile                platform_device_test.ko
modules.order           platform_device_test.mod.c
Module.symvers          platform_device_test.mod.o
platform_device_test.c  platform_device_test.o
root@ubuntu:/home/topeet/neo/device#

```

通过 U 盘挂载、tftp 或者 nfs 功能把 “platform\_device\_test.ko” 文件上传到开发板。

先使用 “ls /sys/devices/platform/” 命令查看平台设备。如下图。

```

[root@iTOP-4412]# ls /sys/devices/platform/
adc_ctl          s3c-pl1330.2      samsung-audio
alarm            s3c-sdhci.2       samsung-audio-idma
android_pmem.0   s3c-sdhci.3       samsung-i2s.0
android_pmem.1   s3c-usb gadget    samsung-i2s.4
arm-pmu.0        s3c2410-wdt       samsung-keypad
bt-sysfs         s3c2440-i2c.1     samsung-kmsg
buzzer_ctl       s3c2440-i2c.3     samsung-pd.0
dw_mmc           s3c2440-i2c.4     samsung-pd.1
exynos-busfreq   s3c2440-i2c.5     samsung-pd.2
exynos-usb-switch s3c2440-i2c.7     samsung-pd.5
exynos4412-adc   s3c24xx-pwm.1     samsung-pd.6
gpio-keys        s3c64xx-rtc       samsung-pd.7
i2c-gpio.0       s3c64xx-spi.2     samsung-rp
ion-exynos       s5p-ehci          serial8250
leds             s5p-fimg2d        si_gps
max485_ctl       s5p-pmic          snd-soc-dummy
mt3326-gps       s5p-sysmmu.15     soc-audio
power            s5p-tvout-cec     switch-gpio.0
power.0          s5p-tvout-hpd     tc4-regulator-consumer
reg-dummy        s5pv210-uart.0    uevent
regulatory.0     s5pv210-uart.1    wlan_ar6000_pm_dev.1

```

使用 “insmod platform\_device\_test.ko ” 命令加载 ko 文件。然后再次查看设备，如下图。



```
[root@iTOP-4412]#
[root@iTOP-4412]# insmod platform_device_test.ko
[root@iTOP-4412]# ls /sys/devices/platform/
adc_ctl          s3c-pl1330.1      s5pv210-uart.3
alarm            s3c-pl1330.2      samsung-audio
android_pmem.0   s3c-sdhci.2       samsung-audio-idma
android_pmem.1   s3c-sdhci.3       samsung-i2s.0
arm-pmu.0        s3c-usb gadget     samsung-i2s.4
bt-sysfs         s3c2410-wdt       samsung-keypad
buzzer_ctl       s3c2440-i2c.1     samsung-kmsg
dw_mmc           s3c2440-i2c.3     samsung-pd.0
exynos-busfreq   s3c2440-i2c.4     samsung-pd.1
exynos-usb-switch s3c2440-i2c.5     samsung-pd.2
exynos4412-adc   s3c2440-i2c.7     samsung-pd.5
gpio-keys        s3c24xx-pwm.1     samsung-pd.6
i2c-gpio.0       s3c64xx-rtc       samsung-pd.7
ion-exynos       s3c64xx-spi.2     samsung-rp
leds             s5p-ehci          serial8250
max485_ctl       s5p-fimg2d        si_gps
mt3326-gps       s5p-pmic          snd-soc-dummy
my_code_led      s5p-sysmmu.15     soc-audio
power            s5p-tvout-cec     switch-gpio.0
power.0          s5p-tvout-hpd     tc4-regulator-consumer
reg-dummy        s5pv210-uart.0    uevent
```

对比之后，可以看到新注册的设备 “my\_code\_led”，表明我们成功注册了设备。

使用 “rmmod platform\_device\_test” 命令卸载刚加载的 ko 文件，再次查看平台设备。如下图。

```
[root@iTOP-4412]#
[root@iTOP-4412]# rmmod platform_device_test
[root@iTOP-4412]# ls /sys/devices/platform/
adc_ctl          s3c-pl1330.2      samsung-audio
alarm            s3c-sdhci.2       samsung-audio-idma
android_pmem.0   s3c-sdhci.3       samsung-i2s.0
android_pmem.1   s3c-usb gadget     samsung-i2s.4
arm-pmu.0        s3c2410-wdt       samsung-keypad
bt-sysfs         s3c2440-i2c.1     samsung-kmsg
buzzer_ctl       s3c2440-i2c.3     samsung-pd.0
dw_mmc           s3c2440-i2c.4     samsung-pd.1
exynos-busfreq   s3c2440-i2c.5     samsung-pd.2
exynos-usb-switch s3c2440-i2c.7     samsung-pd.5
exynos4412-adc   s3c24xx-pwm.1     samsung-pd.6
gpio-keys        s3c64xx-rtc       samsung-pd.7
i2c-gpio.0       s3c64xx-spi.2     samsung-rp
ion-exynos       s5p-ehci          serial8250
leds             s5p-fimg2d        si_gps
max485_ctl       s5p-pmic          snd-soc-dummy
mt3326-gps       s5p-sysmmu.15     soc-audio
power            s5p-tvout-cec     switch-gpio.0
power.0          s5p-tvout-hpd     tc4-regulator-consumer
reg-dummy        s5pv210-uart.0    uevent
regulatory.0     s5pv210-uart.1    wlan ar6000 pm dev.1
```

设备 “my\_code\_led” 消失，说明注册的设备已经被注销。

测试完毕。

## 联系方式

北京迅为电子有限公司致力于嵌入式软硬件设计，是高端开发平台以及移动设备方案提供商；基于多年的技术积累，在工控、仪表、教育、医疗、车载等领域通过 OEM/ODM 方式为客户创造价值。

iTOP-4412 开发板是迅为电子基于三星最新四核处理器 Exynos4412 研制的一款实验开发平台，可以通过该产品评估 Exynos 4412 处理器相关性能，并以此为基础开发出用户需要的特定产品。

本手册主要介绍 iTOP-4412 开发板的使用方法，旨在帮助用户快速掌握该产品的应用特点，通过对开发板进行后续软硬件开发，衍生出符合特定需求的应用系统。

如需平板电脑案支持，请访问迅为平板方案网“<http://www.topeet.com>”，我司将有能力为您提供全方位的技术服务，保证您产品设计无忧！

本手册将持续更新，并通过多种方式发布给新老用户，希望迅为电子的努力能给您的学习和开发带来帮助。

迅为电子

2018 年 1 月