

iTOP-4412-驱动-linux 显卡驱动的修改方法

本文档介绍 4412 开发板的显卡分辨率修改方法。

iTOP-4412 开发板支持 4.3 寸、7 寸和 9.7 寸等 lcd 显示屏。其中 4.3 寸屏是用的 cpu 直接出来的 RGB 信号，7 寸屏和 9.7 寸屏是用的 LVDS 信号，硬件上使用了一个 RGB 转 LVDS 的芯片实现的。我们来看下显示驱动，显示驱动在内核的 “drivers/video/samsung” 目录下面，这个驱动是三星提供好的，只讲下需要修改的几个文件。

首先是关于屏幕的分辨率的修改，因为不同的屏幕分辨率，频率以及其他一些硬件参数是不同的，所以我们需要根据这些参数去配置 cpu 的显示控制器，关于这些参数是在

“driversvideo/samsung/s3cfb_wa101s.c” 这个文件，打开这个文件我们可以看到这个文件主要就是定义了一个类型是 s3cfb_lcd 的变量 wa101，屏幕的硬件参数（分辨率，时钟频率以及其它）就是保存在这个变量里面，现在我们来看下这个变量结构类型的定义：

```
struct s3cfb_lcd {  
    int width;  
    int height;  
    int bpp;  
    int freq;  
    struct s3cfb_lcd_timing timing;  
    struct s3cfb_lcd_polarity polarity;  
    void (*init_ldi)(void);  
    void (*deinit_ldi)(void);  
};
```

其中的 width 和 height 指屏幕的分辨率，freq 是时钟频率，bpp 是数据位。timing 是屏幕的其他一些参数，timing 的类型定义如下：

```
struct s3cfb_lcd_timing {  
    int h_fp;  
    int h_bp;  
    int h_sw;  
    int v_fp;  
    int v_fpe;  
    int v_bp;  
    int v_bpe;
```

```
int v_sw;  
};
```

这个结构代表屏幕的左间距，右间距，水平同步信号宽度，垂直同步信号的有效行数等屏幕的硬件参数，这些参数可以通过查看屏幕的数据手册获得。

下面是 polarity 变量，他的定义如下：

```
struct s3cfb_lcd_polarity {  
int rise_vclk;  
int inv_hsync;  
int inv_vsync;  
int inv_vden;  
};
```

这个变量代表时钟行场的极性。

通过修改这个文件里面的这些参数就可以设置 cpu 的显示控制器来支持我们使用的 lcd 屏幕了。

下面我们来看一下 lcd 的控制文件：arch/arm/mach-exynos/setup-fb-s5p.c 在这个文件的 s3cfb_cfg_gpio 函数完成 LCD 数据引脚初始化，驱动能力设为最高 S5P_GPIO_DRVSTR_LV4；管脚驱动能力，S5P_GPIO_DRVSTR_LV1-4 四个等级选择，并且设置 LVDS 芯片的使能引脚输出高：

```
void s3cfb_cfg_gpio(struct platform_device *pdev)  
{  
int err;  
s3cfb_gpio_setup_24bpp(EXYNOS4_GPF0(0), 8, S3C_GPIO_SF0(2), S5P_GPIO_DRVSTR_LV4);  
s3cfb_gpio_setup_24bpp(EXYNOS4_GPF1(0), 8, S3C_GPIO_SF1(2), S5P_GPIO_DRVSTR_LV4);  
s3cfb_gpio_setup_24bpp(EXYNOS4_GPF2(0), 8, S3C_GPIO_SF2(2), S5P_GPIO_DRVSTR_LV4);  
s3cfb_gpio_setup_24bpp(EXYNOS4_GPF3(0), 4, S3C_GPIO_SF3(2), S5P_GPIO_DRVSTR_LV4);  
#if 1 // TC4  
//LVDS_PWDN  
err = gpio_request(EXYNOS4_GPL1(0), "GPL1_0");  
if (err) {  
printk(KERN_ERR "failed to request GPL1 for "  
"lcd power control\n");  
return err;  
}  
gpio_direction_output(EXYNOS4_GPL1(0), 1);  
s3c_gpio_cfgpin(EXYNOS4_GPL1(0), S3C_GPIO_OUTPUT);
```

```
gpio_free(EXYNOS4_GPL1(0));  
#endif  
}
```

然后是时钟控制函数，完成时钟的使能和关闭：

```
int s3cfb_clk_on(struct platform_device *pdev, struct clk **s3cfb_clk)  
{  
    struct clk *sclk = NULL;  
    struct clk *mout_mpll = NULL;  
    struct clk *lcd_clk = NULL;  
    u32 rate = 0;  
    int ret = 0;  
    lcd_clk = clk_get(&pdev->dev, "lcd");  
    if (IS_ERR(lcd_clk)) {  
        dev_err(&pdev->dev, "failed to get operation clk for fimd\n");  
        goto err_clk0;  
    }  
    ret = clk_enable(lcd_clk);  
    if (ret < 0) {  
        dev_err(&pdev->dev, "failed to clk_enable of lcd clk for fimd\n");  
        goto err_clk0;  
    }  
    clk_put(lcd_clk);  
    sclk = clk_get(&pdev->dev, "sclk_fimd");  
    if (IS_ERR(sclk)) {  
        dev_err(&pdev->dev, "failed to get sclk for fimd\n");  
        goto err_clk1;  
    }  
    if (soc_is_exynos4210())  
        mout_mpll = clk_get(&pdev->dev, "mout_mpll");  
    else  
        mout_mpll = clk_get(&pdev->dev, "mout_mpll_user");  
    if (IS_ERR(mout_mpll)) {  
        dev_err(&pdev->dev, "failed to get mout_mpll for fimd\n");  
        goto err_clk2;  
    }  
    ret = clk_set_parent(sclk, mout_mpll);  
    if (ret < 0) {  
        dev_err(&pdev->dev, "failed to clk_set_parent for fimd\n");  
        goto err_clk2;  
    }  
}
```

```
}
ret = clk_set_rate(sclk, 800000000);
if (ret < 0) {
dev_err(&pdev->dev, "failed to clk_set_rate of sclk for fimd\n");
goto err_clk2;
}
dev_dbg(&pdev->dev, "set fimd sclk rate to %d\n", rate);
clk_put(mout_mpll);
ret = clk_enable(sclk);
if (ret < 0) {
dev_err(&pdev->dev, "failed to clk_enable of sclk for fimd\n");
goto err_clk2;
}
*s3cfb_clk = sclk;
return 0;
err_clk2:
clk_put(mout_mpll);
err_clk1:
clk_put(sclk);
err_clk0:
clk_put(lcd_clk);
return -EINVAL;
}

int s3cfb_clk_off(struct platform_device *pdev, struct clk **clk)
{
struct clk *lcd_clk = NULL;
lcd_clk = clk_get(&pdev->dev, "lcd");
if (IS_ERR(lcd_clk)) {
printk(KERN_ERR "failed to get ip clk for fimd0\n");
goto err_clk0;
}
clk_disable(lcd_clk);
clk_put(lcd_clk);
clk_disable(*clk);
clk_put(*clk);
*clk = NULL;
return 0;
err_clk0:
clk_put(lcd_clk);
```

```
return -EINVAL;
}
void s3cfb_get_clk_name(char *clk_name)
{
strcpy(clk_name, "sclk_fimd");
}
```

然后是 s3cfb_backlight_on 函数，这个是使能屏幕显示，s3cfb_backlight_off 关闭屏幕显示。

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 10 月