iTOP-4412 开发板 LCD 的屏幕驱动

大家好今天我们来讲一下 iTOP-4412 开发板 LCD 的屏幕驱动，iTOP-4412 开发板支持 4.3 寸，7 寸，9.7 寸的 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，我们这支只讲下我们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。

iTIO-4412 开发板内核启动时 LCD 会显示 logo，关于这个 logo 是保存在“drivers/video/samsung/iTop-4412.h”文件，打开这个文件，会看到里面指示定义了一个数组 iBitmapData_q，这个数组的内容就是要显示的 logo。我们修改 logo，就需要准备一张 480x640 的 bmp 图片然后使用 Image2LCD 软件转换成数组，把 iBitmapData_q 里面的内容用新生成的数组替换掉。

有可能我们自己制作的 logo 没有显示在屏幕的最中央，那我们需要修改下文件“drivers/video/samsung/ s3cfb_ops.c”，在这个文件找到函数：s3cfb_draw_logo

```
int s3cfb_draw_logo(struct fb_info *fb)
{
#ifdef CONFIG_FB_S5P_SPLASH_SCREEN
    struct fb_fix_screeninfo *fix = &fb->fix;
    struct fb_var_screeninfo *var = &fb->var;
#if 0
    struct s3c_platform_fb *pdata = to_fb_plat(fbdev->dev);
    memcpy(fbdev->fb[pdata->default_win]->screen_base,
            LOGO_RGB24, fix->line_length * var->yres);
#else
    //u32 height = var->yres / 3;
    u32 line = fix->line_length;
    u32 i, j;
```

```
    u32 index;
    u32 top,left;
    const unsigned char *pLog =NULL;



    memset(fb->screen_base, 0x00, var->yres * line);
    printk("\n CPU type: \n");
    if(soc_is_exynos4412()){
        printk(" Exynos 4412\n");
        pLog = iBitmapData_q;
    }else{
        printk("Exynos 4212\n");
        pLog = iBitmapData;
    }


    top = 170;
    left = 230;

    index = 0;
    for (i = 0; i < 480; i++) {
        for (j = 0; j < 640; j++) {
```

```
                memset(fb->screen_base + (i + top) * line + (j + left) * 4 + 0, pLog[index], 1);//B
                memset(fb->screen_base + (i + top) * line + (j + left) * 4 + 1, pLog[index+1], 1);//G
                memset(fb->screen_base + (i + top) * line + (j + left) * 4 + 2, pLog[index+2], 1);//R
                memset(fb->screen_base + (i + top) * line + (j + left) * 4 + 3, 0x00, 1);
                index += 3;
            }
        }

#endif
#endif

    return 0;
}
```

修改这个函数里面的 top 和 left 就可以控制图片在屏幕显示的位置了。

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_SFN(2), S5P_GPIO_DRVSTR_LV4);
    s3cfb_gpio_setup_24bpp(EXYNOS4_GPF1(0), 8, S3C_GPIO_SFN(2), S5P_GPIO_DRVSTR_LV4);
    s3cfb_gpio_setup_24bpp(EXYNOS4_GPF2(0), 8, S3C_GPIO_SFN(2), S5P_GPIO_DRVSTR_LV4);
    s3cfb_gpio_setup_24bpp(EXYNOS4_GPF3(0), 4, S3C_GPIO_SFN(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。