

Linux3.0.8平台搭建移植文档——lcd驱动移植

1. lcd驱动移植—背光灯

1) 修改背光灯 platform_device 资源

修改 platform_device 中的 platform_data (即 `platform_pwm_backlight_data`) 的参数如下:

#vi arch/arm/mach-s5pv210/mach-smdkv210.c

```
static struct platform_pwm_backlight_data smdkv210_backlight_data = {
    .pwm_id          = 3,
    .max_brightness   = 255,
    .dft_brightness   = 255,
    .pwm_period_ns    = 2500000,
    .init             = smdkv210_backlight_init,
    .exit             = smdkv210_backlight_exit,
};
```

2) 配置背光灯选项

#make menuconfig

System Type -->

[*] PWM device support

Device Drivers -->

Graphics support --->

...

[*] Backlight & LCD device support --->

--- Backlight & LCD device support

< > Lowlevel LCD controls

<*> Lowlevel Backlight controls

< > Generic (aka Sharp Corgi) Backlight Driver

<*> Generic PWM based Backlight Driver

3) make

将在 arch/arm/boot/ 下生成编译好的可执行程序 `zImage` 下载到开发板即可

4) 测试背光灯驱动是否正常运行

我们可以调节一下背光的亮度，背光灯驱动在 `sys` 目录下为我们提供了操作接口，找到如下文件：

`cd /sys/devices/platform/s3c24xx-pwm.3/pwm-backlight.0/backlight/pwm-backlight.0/`

显示当前亮度: #cat brightness

设置背光亮度: #echo 50 > brightness //背光灯亮度范围为0~255

2. lcd驱动移植—frambuffer

1) 修改 frambuffer 的 platform_device 资源

修改扫描参数 (即 smdkv210_fb_win0变量), 具体修改如下:

#vi arch/arm/mach-s5pv210/mach-smdkv210.c

```
static struct s3c_fb_pd_win smdkv210_fb_win0 = {
    .win_mode = {
        .left_margin    = 13,
        .right_margin   = 120,           //右边界长度修改为120
        .upper_margin   = 7,
        .lower_margin   = 5,
        .hsync_len      = 3,
        .vsync_len      = 1,
        .xres            = 800,
        .yres            = 480,
    },
    .max_bpp            = 32,
    .default_bpp        = 32,           //位深设置为32
};
```

2) 配置 framebuffer 选项

#make menuconfig

Device Drivers -->

Graphics support --->

...

<*> Support for frame buffer devices --->

*** Frame buffer hardware drivers ***

< > Epson S1D13XXX framebuffer support

<*> Samsung S3C framebuffer support

[] Debug register writes

...

console display driver support --->

<*> Framebuffer Console support

[*] Map the console to the primary display device

[] Framebuffer Console Rotation (NEW)

[] Select compiled-in fonts (NEW)

3) make

将在 arch/arm/boot/ 下生成编译好的可执行程序 **zImage** 下载到开发板即可，现象为不再花屏，也可以向屏幕输出一个字符串进行验证： `echo "hello" > /dev/tty0`

3. lcd驱动移植—开机logo功能实现

1) 开机 logo 的驱动代码准备

#vi drivers/video/s3c-fb.c

在 `s3c_fb_probe` 函数的结尾部添加以下代码：

```
static int __devinit s3c_fb_probe(struct platform_device *pdev)
{
    ....
    ....
    platform_set_drvdata(pdev, sfb);
    pm_runtime_put_sync(sfb->dev);

    /* show the logo*/
    #if defined(CONFIG_LOGO)
        if (fb_prepare_logo( sfb->windows[pd->default_win]->fbinfo, FB_ROTATE_UR)) {
            printk("[valor lion]: Start display and show logo\n");
            /* Start display and show logo on boot */
            fb_set_cmap(&sfb->windows[pd->default_win]->fbinfo->cmap,
sfb->windows[pd->default_win]->fbinfo);
            fb_show_logo(sfb->windows[pd->default_win]->fbinfo, FB_ROTATE_UR);
        }
    #endif
    /*end*/
    return 0;
    ....
}
```

在上面添加的代码中使用了“default_win”，所以要为找到 `smdkv210_lcd0_pdata` 结构体添加一个成员：

#vi arch/arm/mach-s5pv210/mach-smdkv210.c

```
static struct s3c_fb_platdata smdkv210_lcd0_pdata __initdata = {
    .win[0]          = &smdkv210_fb_win0,
    .default_win = 0,
    .vidcon0         = VIDCON0_VIDOUT_RGB | VIDCON0_PNRMODE_RGB,
    .vidcon1         = VIDCON1_INV_HSYNC | VIDCON1_INV_VSYNC,
```

```
.setup_gpio      = s5pv210_fb_gpio_setup_24bpp,  
};
```

2) 配置 logo 驱动支持

#make menuconfig

```
Device Drivers -->  
  Graphics support --->  
    [*] Bootup logo --->  
      [ ]   Standard black and white Linux logo  
      [ ]   Standard 16-color Linux logo  
      [*]   Standard 224-color Linux logo
```

3) make

将在 arch/arm/boot/ 下生成编译好的可执行程序 **zImage** 下载到开发板即可，现象为左上角会现示一只小企鹅

注意：

运行 qt 程序提示以下错误：

```
[root@sunplusedu /home]#Cannot create semaphore /tmp/qtembedded-0/QtEmbedded-0 'd'  
Error 38 Function not implemented  
Cannot get display lock
```

answer:

这个问题发生的主要原因是因为 linux kernel 不支持信号量的相关操作，也就是说在 kernel 的编译配置中没有信号量相关的选项给选上，所以导致不能正常创建！编译时配置内核如下选项：

```
General setup --->  
  ...  
  [*] System V IPC  
  ...
```