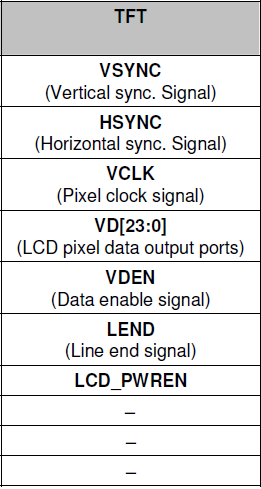
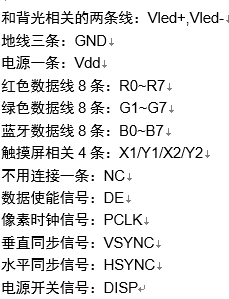
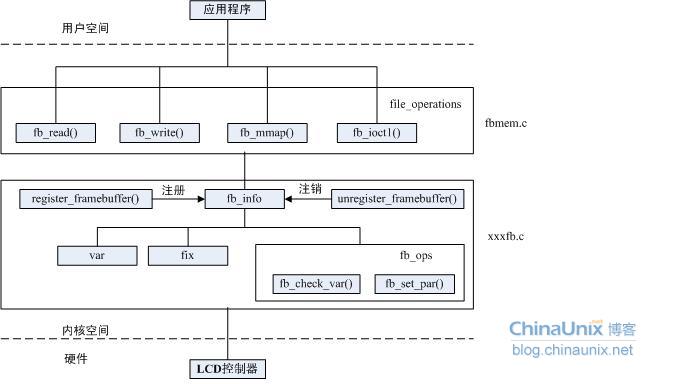
# Linux驱动\_LCD

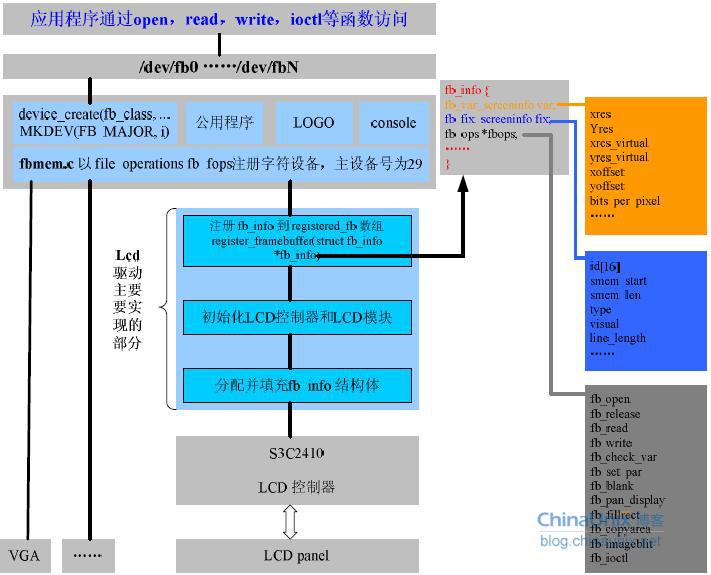
## 1. LCD物理连接

JZ2440上使用的LCD屏一共有40条线。其中：  
左边为LCD屏的pin脚，右边为CPU S3C2440的pin脚。  


其中LEDN信号不是必须的，这样两边的pin脚便可以对应起来。触摸屏另做处理。

## 2. LCD驱动架构图

[](http://blog.chinaunix.net/attachment/201209/3/27664726_1346683702gj8i.jpg)

[](http://blog.chinaunix.net/attachment/201209/4/27664726_1346764943DgTg.jpg)

## 3. LCD驱动代码分析

### 3.1 platform device注册

在和开发板相关的初始化函数smdk2440\_machine\_init中，会在总线上注册LCD platform device。

static struct platform\_device **\***smdk2440\_devices**[]** \_\_initdata **=** **{**

**&**s3c\_device\_usb**,**

**&**s3c\_device\_lcd**,**

**&**s3c\_device\_wdt**,**

**&**s3c\_device\_i2c**,**

**&**s3c\_device\_iis**,**

**&**s3c2440\_device\_sdi**,**

**};**

static void \_\_init smdk2440\_machine\_init**(**void**)**

**{**

s3c24xx\_fb\_set\_platdata**(&**smdk2440\_lcd\_cfg**);**

platform\_add\_devices**(**smdk2440\_devices**,** ARRAY\_SIZE**(**smdk2440\_devices**));**

smdk\_machine\_init**();**

**}**

其中lcd platform device相关代码如下：

#define IRQ\_LCD S3C2410\_IRQ(16) /\* 32 \*/

#define S3C24XX\_SZ\_LCD SZ\_1M

#define S3C24XX\_PA\_LCD S3C2410\_PA\_LCD

#define S3C2410\_PA\_LCD (0x4D000000)

static struct resource s3c\_lcd\_resource**[]** **=** **{**

**[**0**]** **=** **{**

**.**start **=** S3C24XX\_PA\_LCD**,**

**.**end **=** S3C24XX\_PA\_LCD **+** S3C24XX\_SZ\_LCD **-** 1**,**

**.**flags **=** IORESOURCE\_MEM**,**

**},**

**[**1**]** **=** **{**

**.**start **=** IRQ\_LCD**,**

**.**end **=** IRQ\_LCD**,**

**.**flags **=** IORESOURCE\_IRQ**,**

**}**

**};**

static u64 s3c\_device\_lcd\_dmamask **=** 0xffffffffUL**;**

struct platform\_device s3c\_device\_lcd **=** **{**

**.**name **=** "s3c2410-lcd"**,**

**.**id **=** **-**1**,**

**.**num\_resources **=** ARRAY\_SIZE**(**s3c\_lcd\_resource**),**

**.**resource **=** s3c\_lcd\_resource**,**

**.**dev **=** **{**

**.**dma\_mask **=** **&**s3c\_device\_lcd\_dmamask**,**

**.**coherent\_dma\_mask **=** 0xffffffffUL

**}**

**};**

在上述代码中定义了lcd操作相关的寄存器地址和中断信息。

另外注意s3c24xx\_fb\_set\_platdata函数，该函数设置了platform device的platform\_data选项。

void \_\_init s3c24xx\_fb\_set\_platdata**(**struct s3c2410fb\_mach\_info **\***pd**)**

**{**

struct s3c2410fb\_mach\_info **\***npd**;**

npd **=** kmalloc**(sizeof(\***npd**),** GFP\_KERNEL**);**

**if** **(**npd**)** **{**

memcpy**(**npd**,** pd**,** **sizeof(\***npd**));**

s3c\_device\_lcd**.**dev**.**platform\_data **=** npd**;**

**}** **else** **{**

printk**(**KERN\_ERR "no memory for LCD platform data\n"**);**

**}**

**}**

/\* 480x272 \*/

static struct s3c2410fb\_mach\_info smdk2440\_lcd\_cfg \_\_initdata **=** **{**

**.**regs **=** **{**

**.**lcdcon1 **=** S3C2410\_LCDCON1\_TFT16BPP **|** \

S3C2410\_LCDCON1\_TFT **|** \

S3C2410\_LCDCON1\_CLKVAL**(**0x04**),**

**.**lcdcon2 **=** S3C2410\_LCDCON2\_VBPD**(**1**)** **|** \

S3C2410\_LCDCON2\_LINEVAL**(**271**)** **|** \

S3C2410\_LCDCON2\_VFPD**(**1**)** **|** \

S3C2410\_LCDCON2\_VSPW**(**9**),**

**.**lcdcon3 **=** S3C2410\_LCDCON3\_HBPD**(**1**)** **|** \

S3C2410\_LCDCON3\_HOZVAL**(**479**)** **|** \

S3C2410\_LCDCON3\_HFPD**(**1**),**

**.**lcdcon4 **=** S3C2410\_LCDCON4\_HSPW**(**40**),**

**.**lcdcon5 **=** S3C2410\_LCDCON5\_FRM565 **|**

S3C2410\_LCDCON5\_INVVLINE **|**

S3C2410\_LCDCON5\_INVVFRAME **|**

S3C2410\_LCDCON5\_PWREN **|**

S3C2410\_LCDCON5\_HWSWP**,**

**},**

**.**gpccon **=** 0xaaaaaaaa**,**

**.**gpccon\_mask **=** 0xffffffff**,**

**.**gpcup **=** 0xffffffff**,**

**.**gpcup\_mask **=** 0xffffffff**,**

**.**gpdcon **=** 0xaaaaaaaa**,**

**.**gpdcon\_mask **=** 0xffffffff**,**

**.**gpdup **=** 0xffffffff**,**

**.**gpdup\_mask **=** 0xffffffff**,**

**.**fixed\_syncs **=** 1**,**

**.**type **=** S3C2410\_LCDCON1\_TFT**,**

**.**width **=** 480**,**

**.**height **=** 272**,**

**.**xres **=** **{**

**.**min **=** 480**,**

**.**max **=** 480**,**

**.**defval **=** 480**,**

**},**

**.**yres **=** **{**

**.**max **=** 272**,**

**.**min **=** 272**,**

**.**defval **=** 272**,**

**},**

**.**bpp **=** **{**

**.**min **=** 16**,**

**.**max **=** 16**,**

**.**defval **=** 16**,**

**},**

**};**

### 3.2 platform driver注册

int \_\_devinit s3c2410fb\_init**(**void**)**

**{**

**return** platform\_driver\_register**(&**s3c2410fb\_driver**);**

**}**

static void \_\_exit s3c2410fb\_cleanup**(**void**)**

**{**

platform\_driver\_unregister**(&**s3c2410fb\_driver**);**

**}**

static struct platform\_driver s3c2410fb\_driver **=** **{**

**.**probe **=** s3c2410fb\_probe**,**

**.**remove **=** s3c2410fb\_remove**,**

**.**suspend **=** s3c2410fb\_suspend**,**

**.**resume **=** s3c2410fb\_resume**,**

**.**driver **=** **{**

**.**name **=** "s3c2410-lcd"**,**

**.**owner **=** THIS\_MODULE**,**

**},**

**};**

### 3.3 probe函数

当platform device和platform driver匹配上时，s3c2410fb\_probe函数被调用。

(1)从lcd注册的platform\_device中获取一些有用的信息，操作寄存器和中断号。

//获取s3c24xx\_fb\_set\_platdata设置的smdk2440\_lcd\_cfg结构体

mach\_info **=** pdev**->**dev**.**platform\_data**;**

**if** **(**mach\_info **==** **NULL)** **{**

dev\_err**(&**pdev**->**dev**,**"no platform data for lcd, cannot attach\n"**);**

**return** **-**EINVAL**;**

**}**

//获取控制lcd的5个寄存器lcdcon1~lcdcon5内容

mregs **=** **&**mach\_info**->**regs**;**

//获取中断号

irq **=** platform\_get\_irq**(**pdev**,** 0**);**

**if** **(**irq **<** 0**)** **{**

dev\_err**(&**pdev**->**dev**,** "no irq for device\n"**);**

**return** **-**ENOENT**;**

**}**

(2)为fd\_info结构体分配内存空间

//分配一个fb\_info结构体,第一参数不为0表示，额外多申请的空间

//用来存放额外的数据，这里用来存放s3c2410fb\_info额外的数据

//比如:clk，resource，io,irq\_base,drv\_type等额外信息

fbinfo **=** framebuffer\_alloc**(sizeof(**struct s3c2410fb\_info**),** **&**pdev**->**dev**);**

**if** **(!**fbinfo**)** **{**

**return** **-**ENOMEM**;**

**}**

framebuffer\_alloc源码如下：

struct fb\_info **\***framebuffer\_alloc**(**size\_t size**,** struct device **\***dev**)**

**{**

//计算一个long型多少个Byte

#define BYTES\_PER\_LONG (BITS\_PER\_LONG/8)

//计算fb\_info结构体按Long型对齐还差几个字节

#define PADDING (BYTES\_PER\_LONG - (sizeof(struct fb\_info) % BYTES\_PER\_LONG))

//计算fb\_info的大小

int fb\_info\_size **=** **sizeof(**struct fb\_info**);**

struct fb\_info **\***info**;**

char **\***p**;**

//假如size不为0，fb\_info的长度需要按四字节对齐

**if** **(**size**)**

fb\_info\_size **+=** PADDING**;**

//开辟fb\_info及其额外内容的空间

p **=** kzalloc**(**fb\_info\_size **+** size**,** GFP\_KERNEL**);**

**if** **(!**p**)**

**return** **NULL;**

info **=** **(**struct fb\_info **\*)** p**;**

//额外的开辟的空间指向fb\_info的par指针

**if** **(**size**)**

info**->**par **=** p **+** fb\_info\_size**;**

//platform\_device的device结构体赋给fb\_info的device指针

info**->**device **=** dev**;**

#ifdef CONFIG\_FB\_BACKLIGHT

mutex\_init**(&**info**->**bl\_curve\_mutex**);**

#endif

**return** info**;**

#undef PADDING

#undef BYTES\_PER\_LONG

**}**

(3)设置一些指针

//info指向了创建fb\_info时额外多申请内存空间的首地址

info **=** fbinfo**->**par**;**

//info的fb指针设置为fbinfo地址

info**->**fb **=** fbinfo**;**

//info的dev指向platform\_device的dev

info**->**dev **=** **&**pdev**->**dev**;**

//相当于pdev->dev->driver\_data = fbinfo

platform\_set\_drvdata**(**pdev**,** fbinfo**);**

(4)设置fb\_info的相关参数，关闭lcd信号输出

//设置identification string为s3c2410fb

strcpy**(**fbinfo**->**fix**.**id**,** driver\_name**);**

//获取lcdcon1~lcdcon5的内容

memcpy**(&**info**->**regs**,** **&**mach\_info**->**regs**,** **sizeof(**info**->**regs**));**

//清空lcdcon1的ENVID位，ENVID：LCD信号输出使能位

info**->**regs**.**lcdcon1 **&=** **~**S3C2410\_LCDCON1\_ENVID**;**

//读出目前lcdcon1寄存器内容

lcdcon1 **=** readl**(**S3C2410\_LCDCON1**);**

//LCD信号输出禁止

writel**(**lcdcon1 **&** **~**S3C2410\_LCDCON1\_ENVID**,** S3C2410\_LCDCON1**);**

//关闭背光

s3c2410\_gpio\_setpin**(**S3C2410\_GPB0**,** 0**);** // back light control

//s3c2410fb\_info的mach\_info设置为smdk2440\_lcd\_cfg

info**->**mach\_info **=** pdev**->**dev**.**platform\_data**;**

//设置framebuffer类型 Packed Pixels

fbinfo**->**fix**.**type **=** FB\_TYPE\_PACKED\_PIXELS**;**

fbinfo**->**fix**.**type\_aux **=** 0**;**

fbinfo**->**fix**.**xpanstep **=** 0**;**

fbinfo**->**fix**.**ypanstep **=** 0**;**

fbinfo**->**fix**.**ywrapstep **=** 0**;**

//无硬件加速

fbinfo**->**fix**.**accel **=** FB\_ACCEL\_NONE**;**

fbinfo**->**var**.**nonstd **=** 0**;**

//set values immediately

fbinfo**->**var**.**activate **=** FB\_ACTIVATE\_NOW**;**

//smdk2440\_lcd\_cfg height 272

fbinfo**->**var**.**height **=** mach\_info**->**height**;**

//smdk2440\_lcd\_cfg weight 480

fbinfo**->**var**.**width **=** mach\_info**->**width**;**

fbinfo**->**var**.**accel\_flags **=** 0**;**

fbinfo**->**var**.**vmode **=** FB\_VMODE\_NONINTERLACED**;**

//设置和framebuffer相关的操作函数

fbinfo**->**fbops **=** **&**s3c2410fb\_ops**;**

fbinfo**->**flags **=** FBINFO\_FLAG\_DEFAULT**;**

//设置假调色板

fbinfo**->**pseudo\_palette **=** **&**info**->**pseudo\_pal**;**

//480

fbinfo**->**var**.**xres **=** mach\_info**->**xres**.**defval**;**

//480

fbinfo**->**var**.**xres\_virtual **=** mach\_info**->**xres**.**defval**;**

//272

fbinfo**->**var**.**yres **=** mach\_info**->**yres**.**defval**;**

//272

fbinfo**->**var**.**yres\_virtual **=** mach\_info**->**yres**.**defval**;**

//16 每个像素16bit

fbinfo**->**var**.**bits\_per\_pixel **=** mach\_info**->**bpp**.**defval**;**

//根据VBPD等值设置对应的参数

//6

fbinfo**->**var**.**upper\_margin **=** S3C2410\_LCDCON2\_GET\_VBPD**(**mregs**->**lcdcon2**)** **+** 1**;**

//4

fbinfo**->**var**.**lower\_margin **=** S3C2410\_LCDCON2\_GET\_VFPD**(**mregs**->**lcdcon2**)** **+** 1**;**

//2

fbinfo**->**var**.**vsync\_len **=** S3C2410\_LCDCON2\_GET\_VSPW**(**mregs**->**lcdcon2**)** **+** 1**;**

//2

fbinfo**->**var**.**left\_margin **=** S3C2410\_LCDCON3\_GET\_HFPD**(**mregs**->**lcdcon3**)** **+** 1**;**

//11

fbinfo**->**var**.**right\_margin **=** S3C2410\_LCDCON3\_GET\_HBPD**(**mregs**->**lcdcon3**)** **+** 1**;**

//1

fbinfo**->**var**.**hsync\_len **=** S3C2410\_LCDCON4\_GET\_HSPW**(**mregs**->**lcdcon4**)** **+** 1**;**

//设置红绿蓝数据在16个bit中的偏移和大小

fbinfo**->**var**.**red**.**offset **=** 11**;**

fbinfo**->**var**.**green**.**offset **=** 5**;**

fbinfo**->**var**.**blue**.**offset **=** 0**;**

fbinfo**->**var**.**transp**.**offset **=** 0**;**

fbinfo**->**var**.**red**.**length **=** 5**;**

fbinfo**->**var**.**green**.**length **=** 6**;**

fbinfo**->**var**.**blue**.**length **=** 5**;**

fbinfo**->**var**.**transp**.**length **=** 0**;**

//480\*272所占用的内存大小480\*272\*2

fbinfo**->**fix**.**smem\_len **=** mach\_info**->**xres**.**max **\***

mach\_info**->**yres**.**max **\***

mach\_info**->**bpp**.**max **/** 8**;**

//palette\_buffer[i] = 0x80000000，清空调色板

**for** **(**i **=** 0**;** i **<** 256**;** i**++)**

info**->**palette\_buffer**[**i**]** **=** PALETTE\_BUFF\_CLEAR**;**

其中S3C2410\_LCDCON1定义如下,开机的时候会完成静态映射。

static struct map\_desc s3c244x\_iodesc**[]** \_\_initdata **=** **{**

IODESC\_ENT**(**CLKPWR**),**

IODESC\_ENT**(**TIMER**),**

IODESC\_ENT**(**WATCHDOG**),**

IODESC\_ENT**(**LCD**),**

**};**

#define S3C2410\_LCDCON1 S3C2410\_LCDREG(0x00)

#define S3C2410\_LCDREG(x) ((x) + S3C24XX\_VA\_LCD)

#define S3C24XX\_VA\_LCD S3C2410\_ADDR(0x00300000)

#define S3C2410\_ADDR(x) ((void \_\_iomem \_\_force \*)0xF0000000 + (x))

s3c2410fb\_ops定义如下：

static struct fb\_ops s3c2410fb\_ops **=** **{**

**.**owner          **=** THIS\_MODULE**,**

**.**fb\_check\_var   **=** s3c2410fb\_check\_var**,** //设置可变参数

**.**fb\_set\_par     **=** s3c2410fb\_set\_par**,** //设置固定参数及lcdcon寄存器

**.**fb\_blank       **=** s3c2410fb\_blank**,**   //设置是否使能LCD控制器

**.**fb\_setcolreg   **=** s3c2410fb\_setcolreg**,** //设置RGB颜色，实现伪颜色表

**.**fb\_fillrect    **=** cfb\_fillrect**,**       //画一个矩形

**.**fb\_copyarea    **=** cfb\_copyarea**,**//Copy data from area to another

**.**fb\_imageblit   **=** cfb\_imageblit**,** //Draws a image to the display

**};**

(5)标记LCD要使用的物理地址，并申请LCD中断

//标记S3C24XX\_VA\_LCD后的1M空间被使用了

**if** **(!**request\_mem\_region**((**unsigned long**)**S3C24XX\_VA\_LCD**,** SZ\_1M**,** "s3c2410-lcd"**))** **{**

ret **=** **-**EBUSY**;**

**goto** dealloc\_fb**;**

**}**

dprintk**(**"got LCD region\n"**);**

//申请lcd中断，处理函数为s3c2410fb\_irq，私有信息为s3c2410fb\_info

ret **=** request\_irq**(**irq**,** s3c2410fb\_irq**,** IRQF\_DISABLED**,** pdev**->**name**,** info**);**

**if** **(**ret**)** **{**

dev\_err**(&**pdev**->**dev**,** "cannot get irq %d - err %d\n"**,** irq**,** ret**);**

ret **=** **-**EBUSY**;**

**goto** release\_mem**;**

**}**

(6)使能时钟

info**->**clk **=** clk\_get**(NULL,** "lcd"**);**

**if** **(!**info**->**clk **||** IS\_ERR**(**info**->**clk**))** **{**

printk**(**KERN\_ERR "failed to get lcd clock source\n"**);**

ret **=** **-**ENOENT**;**

**goto** release\_irq**;**

**}**

clk\_enable**(**info**->**clk**);**

lcd的时钟会在开机的时候加入到一个链表中。使能时钟就是将S3C2410\_CLKCON寄存器LCD对应的bit位置1。

**.**name **=** "lcd"**,**

**.**id **=** **-**1**,**

**.**parent **=** **&**clk\_h**,**

**.**enable **=** s3c2410\_clkcon\_enable**,**

**.**ctrlbit **=** S3C2410\_CLKCON\_LCDC**,**

int clk\_enable**(**struct clk **\***clk**)**

**{**

**if** **(**IS\_ERR**(**clk**)** **||** clk **==** **NULL)**

**return** **-**EINVAL**;**

clk\_enable**(**clk**->**parent**);**

mutex\_lock**(&**clocks\_mutex**);**

**if** **((**clk**->**usage**++)** **==** 0**)**

**(**clk**->**enable**)(**clk**,** 1**);**

mutex\_unlock**(&**clocks\_mutex**);**

**return** 0**;**

**}**

int s3c2410\_clkcon\_enable**(**struct clk **\***clk**,** int enable**)**

**{**

unsigned int clocks **=** clk**->**ctrlbit**;**

unsigned long clkcon**;**

clkcon **=** \_\_raw\_readl**(**S3C2410\_CLKCON**);**

**if** **(**enable**)**

clkcon **|=** clocks**;**

**else**

clkcon **&=** **~**clocks**;**

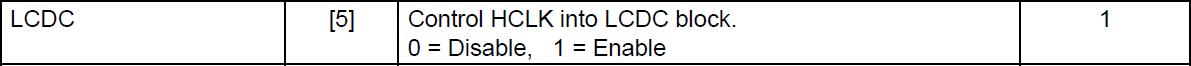
/\* ensure none of the special function bits set \*/

clkcon **&=** **~(**S3C2410\_CLKCON\_IDLE**|**S3C2410\_CLKCON\_POWER**);**

\_\_raw\_writel**(**clkcon**,** S3C2410\_CLKCON**);**

**return** 0**;**

**}**



(7)分配显存

/\* Initialize video memory \*/

ret **=** s3c2410fb\_map\_video\_memory**(**info**);**

**if** **(**ret**)** **{**

printk**(** KERN\_ERR "Failed to allocate video RAM: %d\n"**,** ret**);**

ret **=** **-**ENOMEM**;**

**goto** release\_clock**;**

**}**

dprintk**(**"got video memory\n"**);**

具体实现代码如下：

static int \_\_init s3c2410fb\_map\_video\_memory**(**struct s3c2410fb\_info **\***fbi**)**

**{**

dprintk**(**"map\_video\_memory(fbi=%p)\n"**,** fbi**);**

//framebuffer缓存大小，多分配PAGE\_SIZE

fbi**->**map\_size **=** PAGE\_ALIGN**(**fbi**->**fb**->**fix**.**smem\_len **+** PAGE\_SIZE**);**

//A = dma\_alloc\_writecombine(B,C,D,GFP\_KERNEL);

//A：内存的虚拟起始地址，在内核要用此地址来操作所分配的内存

//B: struct device指针，可以平台初始化里指定，主要是dma\_mask之类

//dma\_mask是设备DMA能访问的内存范围

//C: 实际分配大小，传入dma\_map\_size即可

//D: 返回的内存物理地址，dma就可以用

//A和D是一一对应的，只不过，A是虚拟地址，而D是物理地址。

//dma\_alloc\_writecombine分配的内存物理上是连续的并禁止C(Cacheable)域.

//dma\_alloc\_writecombine分配出来的内存不使用缓存，但是会使用写缓冲区。

fbi**->**map\_cpu **=** dma\_alloc\_writecombine**(**fbi**->**dev**,** fbi**->**map\_size**,**

**&**fbi**->**map\_dma**,** GFP\_KERNEL**);**

fbi**->**map\_size **=** fbi**->**fb**->**fix**.**smem\_len**;**

**if** **(**fbi**->**map\_cpu**)** **{**

/\* prevent initial garbage on screen \*/

dprintk**(**"map\_video\_memory: clear %p:%08x\n"**,**

fbi**->**map\_cpu**,** fbi**->**map\_size**);**

memset**(**fbi**->**map\_cpu**,** 0xf0**,** fbi**->**map\_size**);**

fbi**->**screen\_dma **=** fbi**->**map\_dma**;**

fbi**->**fb**->**screen\_base **=** fbi**->**map\_cpu**;**

fbi**->**fb**->**fix**.**smem\_start **=** fbi**->**screen\_dma**;**

dprintk**(**"map\_video\_memory: dma=%08x cpu=%p size=%08x\n"**,**

fbi**->**map\_dma**,** fbi**->**map\_cpu**,** fbi**->**fb**->**fix**.**smem\_len**);**

**}**

**return** fbi**->**map\_cpu **?** 0 **:** **-**ENOMEM**;**

**}**

(8)设置寄存器

ret **=** s3c2410fb\_init\_registers**(**info**);**

s3c2410fb\_init\_registers代码如下：

static int s3c2410fb\_init\_registers**(**struct s3c2410fb\_info **\***fbi**)**

**{**

unsigned long flags**;**

local\_irq\_save**(**flags**);**

//设置lcd对应的gpio口为lcd总线

modify\_gpio**(**S3C2410\_GPCUP**,** mach\_info**->**gpcup**,** mach\_info**->**gpcup\_mask**);**

modify\_gpio**(**S3C2410\_GPCCON**,** mach\_info**->**gpccon**,** mach\_info**->**gpccon\_mask**);**

modify\_gpio**(**S3C2410\_GPDUP**,** mach\_info**->**gpdup**,** mach\_info**->**gpdup\_mask**);**

modify\_gpio**(**S3C2410\_GPDCON**,** mach\_info**->**gpdcon**,** mach\_info**->**gpdcon\_mask**);**

local\_irq\_restore**(**flags**);**

//设置lcd寄存器

writel**(**fbi**->**regs**.**lcdcon1**,** S3C2410\_LCDCON1**);**

writel**(**fbi**->**regs**.**lcdcon2**,** S3C2410\_LCDCON2**);**

writel**(**fbi**->**regs**.**lcdcon3**,** S3C2410\_LCDCON3**);**

writel**(**fbi**->**regs**.**lcdcon4**,** S3C2410\_LCDCON4**);**

writel**(**fbi**->**regs**.**lcdcon5**,** S3C2410\_LCDCON5**);**

//设置帧缓冲寄存器，把分配的显存物理地址告知

s3c2410fb\_set\_lcdaddr**(**fbi**);**

dprintk**(**"LPCSEL = 0x%08lx\n"**,** mach\_info**->**lpcsel**);**

writel**(**mach\_info**->**lpcsel**,** S3C2410\_LPCSEL**);**

//停止临时调色板

dprintk**(**"replacing TPAL %08x\n"**,** readl**(**S3C2410\_TPAL**));**

writel**(**0x00**,** S3C2410\_TPAL**);**

//打开背光控制

s3c2410\_gpio\_cfgpin**(**S3C2410\_GPB0**,** S3C2410\_GPB0\_OUTP**);** // back light control

s3c2410\_gpio\_pullup**(**S3C2410\_GPB0**,** 0**);**

s3c2410\_gpio\_setpin**(**S3C2410\_GPB0**,** 1**);** // back light control, enable

/\* probably not required \*/

msleep**(**10**);**

//LCD信号输出使能

fbi**->**regs**.**lcdcon1 **|=** S3C2410\_LCDCON1\_ENVID**;**

writel**(**fbi**->**regs**.**lcdcon1**,** S3C2410\_LCDCON1**);**

**return** 0**;**

**}**

s3c2410fb\_set\_lcdaddr代码如下：

static void s3c2410fb\_set\_lcdaddr**(**struct s3c2410fb\_info **\***fbi**)**

**{**

struct fb\_var\_screeninfo **\***var **=** **&**fbi**->**fb**->**var**;**

unsigned long saddr1**,** saddr2**,** saddr3**;**

//这里使用的都是分配出来的物理地址

saddr1 **=** fbi**->**fb**->**fix**.**smem\_start **>>** 1**;**

saddr2 **=** fbi**->**fb**->**fix**.**smem\_start**;**

saddr2 **+=** **(**var**->**xres **\*** var**->**yres **\*** var**->**bits\_per\_pixel**)/**8**;**

saddr2**>>=** 1**;**

saddr3 **=** S3C2410\_OFFSIZE**(**0**)** **|** S3C2410\_PAGEWIDTH**((**var**->**xres **\*** var**->**bits\_per\_pixel **/** 16**)** **&** 0x7ff**);**

dprintk**(**"LCDSADDR1 = 0x%08lx\n"**,** saddr1**);**

dprintk**(**"LCDSADDR2 = 0x%08lx\n"**,** saddr2**);**

dprintk**(**"LCDSADDR3 = 0x%08lx\n"**,** saddr3**);**

writel**(**saddr1**,** S3C2410\_LCDSADDR1**);**

writel**(**saddr2**,** S3C2410\_LCDSADDR2**);**

writel**(**saddr3**,** S3C2410\_LCDSADDR3**);**

**}**

(9)检查fb\_info中的var参数

ret **=** s3c2410fb\_check\_var**(&**fbinfo**->**var**,** fbinfo**);**

(10)向framebuffer子系统注册自己

ret **=** register\_framebuffer**(**fbinfo**);**

**if** **(**ret **<** 0**)** **{**

printk**(**KERN\_ERR "Failed to register framebuffer device: %d\n"**,** ret**);**

**goto** free\_video\_memory**;**

**}**

(11)创建/sys/devices/platform/s3c2410-lcd/debug文件，方便调试。

device\_create\_file**(&**pdev**->**dev**,** **&**dev\_attr\_debug**);**

相关代码如下：

static DEVICE\_ATTR**(**debug**,** 0666**,**

s3c2410fb\_debug\_show**,**

s3c2410fb\_debug\_store**);**

static int s3c2410fb\_debug\_show**(**struct device **\***dev**,** struct device\_attribute **\***attr**,** char **\***buf**)**

**{**

**return** snprintf**(**buf**,** PAGE\_SIZE**,** "%s\n"**,** debug **?** "on" **:** "off"**);**

**}**

static int s3c2410fb\_debug\_store**(**struct device **\***dev**,** struct device\_attribute **\***attr**,**

const char **\***buf**,** size\_t len**)**

**{**

**if** **(**mach\_info **==** **NULL)**

**return** **-**EINVAL**;**

**if** **(**len **<** 1**)**

**return** **-**EINVAL**;**

**if** **(**strnicmp**(**buf**,** "on"**,** 2**)** **==** 0 **||**

strnicmp**(**buf**,** "1"**,** 1**)** **==** 0**)** **{**

debug **=** 1**;**

printk**(**KERN\_DEBUG "s3c2410fb: Debug On"**);**

**}** **else** **if** **(**strnicmp**(**buf**,** "off"**,** 3**)** **==** 0 **||**

strnicmp**(**buf**,** "0"**,** 1**)** **==** 0**)** **{**

debug **=** 0**;**

printk**(**KERN\_DEBUG "s3c2410fb: Debug Off"**);**

**}** **else** **{**

**return** **-**EINVAL**;**

**}**

**return** len**;**

**}**

可以通过操作/sys/devices/platform/s3c2410-lcd/debug打开和关闭debug功能。  
查询当前状态：

# cat /sys/devices/platform/s3c2410-lcd/debug

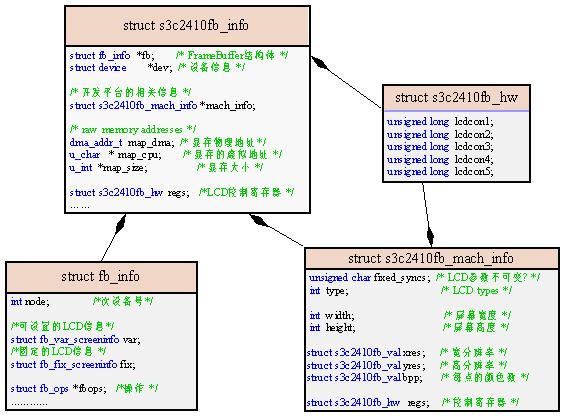
off

改变状态：

# echo 1 > /sys/devices/platform/s3c2410-lcd/debug

# cat /sys/devices/platform/s3c2410-lcd/debug

on



framebuffer mmap

## 5.sysfs快速调试功能：

在sysfs下创建调试文件：

static DEVICE\_ATTR**(**rawdata**,** S\_IRUGO**|**S\_IWUSR**,** gtp\_sysfs\_rawdata\_show**,** gtp\_sysfs\_rawdata\_store**);**

s32 gtp\_sysfs\_init**(**void**)**

**{**

    s32 ret **;**

    debug\_kobj **=** kobject\_create\_and\_add**(**"gtp"**,** **NULL)** **;**

    //SET\_INFO\_LINE\_INFO("Starting initlizing gtp\_debug\_sysfs");

    if **(**debug\_kobj **==** **NULL)**

**{**

        GTP\_ERROR**(**"%s: subsystem\_register failed\n"**,** \_\_func\_\_**);**

        return **-**ENOMEM**;**

**}**

    ret **=** sysfs\_create\_file**(**debug\_kobj**,** **&**dev\_attr\_rawdata**.**attr**);**

    if **(**ret**)**

**{**

        GTP\_ERROR**(**"%s: sysfs\_create\_rawdata\_file failed\n"**,** \_\_func\_\_**);**

        return ret**;**

**}**

    return 0 **;**

**}**

其中读写函数为：

static ssize\_t gtp\_sysfs\_rawdata\_show**(**struct device **\***dev**,**struct device\_attribute **\***attr**,** char **\***buf**)**

**{**

//当read /sys/gtp/rawdata，会打印on/off

**return** snprintf**(**buf**,** PAGE\_SIZE**,** "%s\n"**,** debug **?** "on" **:** "off"**);**

**}**

static ssize\_t gtp\_sysfs\_rawdata\_store**(**struct device **\***dev**,**struct device\_attribute **\***attr**,** char **\***buf**,** size\_t len**)**

**{**

//当write /sys/gtp/rawdata，数据会存放在buf中，长度为len

**if(**strnicmp**(**buf**,** "on"**,** 2**)** **==** 0 **||**

strnicmp**(**buf**,** "1"**,** 1**)** **==** 0**)**

**{**

debug **=** 1**;**

printk**(**KERN\_DEBUG "s3c2410fb: Debug On"**);**

**}** **else** **if** **(**strnicmp**(**buf**,** "off"**,** 3**)** **==** 0 **||**

strnicmp**(**buf**,** "0"**,** 1**)** **==** 0**)**

**{**

debug **=** 0**;**

printk**(**KERN\_DEBUG "s3c2410fb: Debug Off"**);**

**}** **else**

**{**

**return** **-**EINVAL**;**

**}**

**return** len**;**

**}**