

个人资料



chen198746

访问: 46883次

积分: 974分

排名: 第17341名

原创: 47篇

转载: 20篇

译文: 0篇

评论: 26条

文章搜索

文章分类

ARM (11)

C# (5)

c++ (3)

linux (33)

vivi笔记 (3)

工作与生活心得 (9)

文章存档

2014年08月 (2)

2014年02月 (1)

2013年11月 (2)

2013年09月 (2)

2013年08月 (3)

展开

阅读排行

嵌入式经典面试题 不看后悔

PID控制算法 (11526)

Linux终端颜色和标题设置 (1876)

关于SRCPND 和INTPNC (1613)

即将毕业的小硕的忧虑 (1274)

研究生小论文注意事项 (1185)

uCOS移植到PIC24f芯片 (1130)

博客专家福利

【限时活动】建专辑得大奖

专访荣浩: 流程的永恒之道

当青春遇上互联网, 能否点燃你的创业梦

推荐有礼--找出您心中的技术大牛

uboot中LCD初始化之datasheet

分类: linux

2013-05-21 11:38

946人阅读

评论(1)

收藏

举报

Sc8810 DataSheet分析LCD初始化

一、基础条件

依赖代码: Sc8810_fb.c

LCD型号: NT35510_A45026N00HN

pdf中的内容如下:

7.3.6.1 Program Flow

Before configure module, set some global parameters,
Set bit[7:6] of 0x8b00_0070 to select LCD clock source;
Set bit[2:0] of 0x8b00_0060 to control the divider value of LCD clock;
Set bit[3] of 0x2090_0200 to enable LCD clock;
Set or reset bit[3] of 0x2090_0210 to reset LCDC module;
Set bit[29] of 0x8000_3008 to enable LCDC global interrupt.

Display mode

- Clear ARM interrupt and LCDC module interrupt, and enable them;
- Enable LCDC, set the bit 0 of LCDC_CTRL to '1';
- Configure layer information, includes format, size, position, address, etc;

V.0.1

Spreadtrum Communications, Inc., Confidential and Proprietary

801 of 1013

This document contains information on a product under development.
Spreadtrum reserves the right to change or discontinue this product without notice.

SPREADTRUM

SC8810 Design Specification

- Configure LCM parameters;
- Start LCDC, write '1' to bit[3] of LCDC_CTRL;
- Wait LCDC_DONE interrupt for completing a frame;
- Re-configure parameters, and start a new frame;

二、初始化分析

1、初始化部分从static int sc8810fb_probe(void * lcdbase)开始

LCD最后Application Notes中提到如下:

http://blog.csdn.net/chen198746/article/details/8954472

1/8

uboot中LCD初始化之dat (945)

BCM wifi驱动学习 (868)

earlysuspend调用过程 (840)

评论排行

一道有趣的c语言题目 (6)

拒绝平庸与堕落 (3)

uCOS移植到PIC24f芯片 (2)

嵌入式经典面试题 不看后 (2)

UNEXPECTED TOP-LE (2)

关于NO NAND FDEVICE (2)

糊涂的生活 (2)

重新生活 (1)

QT mp3 制作 (1)

mplayer Mp4播放器制作 (1)

推荐文章

* ArcGIS 构建3D动画方法

* Unity3D游戏开发之截屏保存精彩瞬间

* 大数据让生活更加糟糕

* 用Swift开发一个TODO应用

* ActionBar 样式详解 -- 样式主题 简介 ActionBar 的 icon logo 标题 菜单样式修改

* 使用Unity Render Textures实

最新评论

关于SRCPND 和INTPND stzztt: 它的特别之处就在于对当我们要把该寄存器中某个值为1的位设置为0时, 我们...

UNEXPECTED TOP-LEVEL ER chen198746: @u012430878:电脑编译环境, 电脑

nandflash中oob、ecc分析 cache_wang: 好东东!

UNEXPECTED TOP-LEVEL ER 喘不过气来: 32G内存。。。土豪。。。这么大用来干嘛? 我2g改成1024-2048, 关机了, 。

嵌入式经典面试题 不看后悔啊! ok2222991: 一. #error其实是相当有用的, 特别是用于保护一些可配置的宏。例如: #if CFG_NUM > ...

嵌入式经典面试题 不看后悔啊! Sha21: 这些问题看起来虽然简单,但是这里的每一道题都是看出功力的深厚的。

一道有趣的c语言题目 chen198746: @hanmengaidudu:呵呵 是的, 很老的东西了

一道有趣的c语言题目 韩半仙: 这是在考短路或和逻辑或的问题, 只要理解短路, 就没问题了

earlysuspend调用过程 韩半仙: 写的很详细, 一下明白了详细的过程, 我也刚毕业参加工作, 楼主有时间交流哦, QQ: 492459593

uboot中LCD初始化之datasheet cnjenkins: 您好! 我需要sc8810 design specification, 可以给我发一份吗? 我的邮箱是cn...

7.3.6.1 Program Flow

Before configure module, set some global parameters,
Set bit[7:6] of 0x8b00_0070 to select LCD clock source;
Set bit[2:0] of 0x8b00_0060 to control the divider value of LCD clock;
Set bit[3] of 0x2090_0200 to enable LCD clock;
Set or reset bit[3] of 0x2090_0210 to reset LCDC module;
Set bit[29] of 0x8000_3008 to enable LCDC global interrupt.

对应代码static void hw_early_init(struct sc8810fb_info *fb), 如下:

//select LCD clock source

__raw_bits_and(~(1<<6), GR_PLL_SRC); //pll_src=96M

__raw_bits_and(~(1<<7), GR_PLL_SRC);

//set LCD divdior

__raw_bits_and(~(1<<0), GR_GEN4); //div=0

__raw_bits_and(~(1<<1), GR_GEN4);

__raw_bits_and(~(1<<2), GR_GEN4);

//enable LCD clock

__raw_bits_or(1<<3, AHB_CTL0);

//LCD soft reset

__raw_bits_or(1<<3, AHB_SOFT_RST);

mdelay(10);

__raw_bits_and(~(1<<3), AHB_SOFT_RST);

//LCD enable LCDC global interrupt

__raw_bits_and(~(1<<0), LCDC_IRQ_EN);

配置好上述寄存器之后, 执行到显示模式, datasheet描述拉如下:

Display mode

- Clear ARM interrupt and LCDC module interrupt, and enable them;

- Enable LCDC, set the bit 0 of LCDC_CTRL to '1';

对应代码如下:

__raw_bits_or((1<<0), LCDC_IRQ_CLR);

执行子函数lcdc_mcu_init, 其内容如下:

//panel reset

panel_reset();

http://blog.csdn.net/chen198746/article/details/8954472

2/8

```
//LCDC module enable
```

```
reg_val |= (1<<0);
```

```
/*FMARK mode*/
```

```
reg_val |= (1<<1); // no f-mark
```

```
/*FMARK pol*/
```

```
//reg_val |= (1<<2);
```

```
__raw_writel(reg_val, LCDC_CTRL);
```

2、为了区分同一个厂家的LCD模块，由于其读取的ID是相同的，因此需要读取电压AD值来区分不同型号

ADC_Init(); //为了读取电压值

```
ANA_REG_OR(ANA_AGEN, AGEN_ADC_EN);
```

```
ANA_REG_OR(ANA_CLK_CTL, ACLK_CTL_AUXAD_EN | ACLK_CTL_AUXADC_EN);
```

```
ANA_REG_OR(ADC_CTRL, ADC_EN_BIT);
```

3、find_adapt_from_readid读取LCD的ID值，并进行挂载，从而才能实现特定的LCD屏幕初始化。

for循环依次挂载，并相应初始化后，再读取ID值。

```
int i;
```

```
uint32_t id;
```

```
for(i = 0; i < (sizeof(lcd_panel))/(sizeof(lcd_panel[0])); i++) {
```

```
    //first ,try mount
```

```
    mount_panel(fb, lcd_panel[i].panel);
```

```
    //hw init to every panel
```

```
    hw_init(fb);
```

```
    //readid
```

```
    if(fb->panel->ops->lcd_readid) {
```

```
        id = fb->panel->ops->lcd_readid(fb->panel);
```

```
        FB_PRINT("[The panel id is 0x%x]\n", id);
```

```
    } else {
```

```
        id = lcd_readid_default(fb->panel);
```

```
    }
```

```
//if the id is right?

if(id == lcd_panel[i].lcd_id) {

    FB_PRINT("[The lcd panel return id is 0x%x]\n", id);

    save_lcd_id_to_kernel(id);

    return i;

}

}

return -1;
```

4、分配显存

```
fb->smem_start = (uint32_t)lcdbase;

fb->smem_len = fb->panel->width * fb->panel->height;
```

5、这里根据datasheet的操作如下：

- Configure layer information, includes format, size, position, address, etc;

V.0.1

Spreadtrum Communications, Inc., Confidential and Proprietary

801 of 1013

*This document contains information on a product under development.
Spreadtrum reserves the right to change or discontinue this product without notice.*



SC8810 Design Specification

- Configure LCM parameters;

初始化代码中是先进行LCM参数配置，再进行图层信息的配置，个人觉得这里初始化顺序允许颠倒。

代码中LCM参数初始化对应static void hw_init(struct sc8810fb_info *fb)，如下：

```
/* only MCU mode is supported currently */

if (LCD_MODE_RGB == fb->panel->mode)

    return;

//panel reset

panel_reset(fb->panel);

/* set lcdc-lcd interface parameters */

lcdc_lcm_configure(fb);

/* set timing parameters for LCD */
```

```
lcdc_update_lcm_timing(fb->register_timing);
```

其中lcdc_lcm_configure(fb);内容如下:

```
uint32_t reg_val = 0;

/* CS1 bus mode [BIT0]: 8080/6800 */

switch (fb->panel->info.mcu->bus_mode) {

case LCD_BUS_8080:

    break;

case LCD_BUS_6800:

    reg_val |= (1 << 8);

    break;

default:

    break;

}

/* CS1 bus width [BIT1:0] */

switch (fb->panel->info.mcu->bus_width) {

case 8:

    break;

case 9:

    reg_val |= ((1 << 9) | (1 << 12));

    break;

case 16:

    reg_val |= (2 << 9);

    break;

case 18:

    reg_val |= ((3 << 9) | (1 << 12));

    break;

case 24:

    reg_val |= ((4 << 9) | (2 << 12));

    break;

default:

    break;

}
```

```
reg_val |= (1 << 16);
```

```
__raw_writel(reg_val, LCM_CTRL);
```

其中lcdc_update_lcm_timing(fb->register_timing);内容如下:

```
__raw_writel(value, LCM_PARAMETER1);
```

6、以上初始化LCM，根据之前的datasheet知道接着初始化OSD，这之前需要先初始化屏幕，然后再初始化OSD， fb->panel->ops->lcd_init(fb->panel);

7、初始化OSD和IMG总共6层

```
__raw_bits_and(~(1<<0),LCDC_IMG_CTRL);
```

```
__raw_bits_and(~(1<<0),LCDC_OSD2_CTRL);
```

```
__raw_bits_and(~(1<<0),LCDC_OSD3_CTRL);
```

```
__raw_bits_and(~(1<<0),LCDC_OSD4_CTRL);
```

```
__raw_bits_and(~(1<<0),LCDC_OSD5_CTRL);
```

```
reg_val |= (1 << 0);
```

```
reg_val |= (1 << 2);
```

```
reg_val |= (5 << 3); //RGB565
```

```
reg_val |= (2 << 7); //B2B3B0B1
```

```
__raw_writel(reg_val, LCDC_OSD1_CTRL);
```

```
/* OSD1 layer base */
```

```
reg_val = fb->smem_start;
```

```
/*OSD1 layer alpha value*/
```

```
__raw_writel(0xff, LCDC_OSD1_ALPHA);
```

```
/*OSD1 layer size*/
```

```
reg_val = ( fb->panel->width & 0xffff) | (( fb->panel->height & 0xffff )<<16);
```

```
__raw_writel(reg_val, LCDC_OSD1_SIZE_XY);
```

```
/*OSD1 layer start position*/
```

```
__raw_writel(0, LCDC_OSD1_DISP_XY);
```

```
/*OSD1 layer pitch*/

reg_val = ( fb->panel->width & 0xffff );

__raw_writel(reg_val, LCDC_OSD1_PITCH);
```

8、最后设置LCDC和LCM的尺寸

```
/*LCDC workplane size*/

set_lcdsize(fb->panel);

/*LCDC LCM rect size*/

set_lcmrect(fb->panel);
```

9、清除LCD，并设置开机Logo

```
lcd_clear (NULL, 1, 1, NULL);  /* dummy args */

lcd_enable ();//未实现
```

10、通过阅读datasheet后面的屏幕初始化，刷屏如下：

- Start LCDC, write '1' to bit[3] of LCDC_CTRL;
- Wait LCDC_DONE interrupt for completing a frame;
- Re-configure parameters, and start a new frame;

通过代码查找，定位到static void real_refresh(struct sc8810fb_info *fb)

```
fb->panel->ops->lcd_invalidate(fb->panel);

/* set timing parameters for LCD */

lcdc_update_lcm_timing(fb->gram_timing);

__raw_bits_or((1<<3), LCDC_CTRL); /* start refresh */

while(!__raw_readl(LCDC_IRQ_RAW) & (1<<0)); // wait util done

__raw_bits_or((1<<0), LCDC_IRQ_CLR);

/* set timing parameters for LCD */
```

来源： android uboot

```
lcdc_update_lcm_timing(fb->register_timing);
```

三、总结

不难看出，整个屏幕初始化程序的编写基本都是依据datasheet中的lcd application notes编写的，因此如果以后需要编写初始化程序，完全可以按照datasheet中的应用来编写或者分析。

上一篇 [ubuntu下安装Fctix输入法](#)
下一篇 [Linux Kernel初始化分析](#)

顶 0 踩 0

主题推荐 uboot application interface sizeof switch

猜你在找

- | | |
|----------------------------------------|------------------------------|
| Linux GPIO - gpio.c(GPIO各个接口的实现) | 实现uboot 命令自动补全 及 修正设置环境变量时自启 |
| FL2440 Linux kernel + yaffs2根文件移植过程（一） | Linux3.4内核Nand Flash驱动的移植 |
| 华硕主板COM口问题 | Source Insight中的条件编译 |
| uboot下简单小实验LED跑马灯裸机uboot下运行 | struct nand_ecclayout 之解义， |
| u-boot nand flash read/write cmd | U-BOOT LCD分析 |

查看评论

1楼 [cnjenkins](#) 2013-06-04 15:56发表



您好！我需要sc8810 design specification，可以给我发一份吗？
我的邮箱是cnjenkins@163.com

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

- | | | | | | | | | | | | | |
|-----------|---------------|------------|----------------|---------|-----------|------------|------------|------------|--------|-----------|--------|-------|
| 全部主题 | Hadoop | AWS | 移动游戏 | Java | Android | iOS | Swift | 智能硬件 | Docker | OpenStack | | |
| VPN | Spark | ERP | IE10 | Eclipse | CRM | JavaScript | 数据库 | Ubuntu | NFC | WAP | jQuery | |
| BI | HTML5 | Spring | Apache | .NET | API | HTML | SDK | IIS | Fedora | XML | LBS | Unity |
| Splashtop | UML | components | Windows Mobile | Rails | QEMU | KDE | Cassandra | CloudStack | FTC | | | |
| coremail | OPhone | CouchBase | 云计算 | iOS6 | Rackspace | Web App | SpringSide | Maemo | | | | |
| Compuware | 大数据 | aptch | Perl | Tornado | Ruby | Hibernate | ThinkPHP | HBase | Pure | Solr | | |
| Angular | Cloud Foundry | Redis | Scala | Django | Bootstrap | | | | | | | |