紫光展锐
UNISOC

LCD移植介绍

WWW.UNISOC.COM

紫 光 展 锐 科 技

数字世界的生态承载者

修改历史

版本号	日期	注释
V1.0	2019/5/1	初稿
V1.1	2020/6/1	1. 文档名从《UNISOC相关芯片平台LCD移植文档介绍》修改为《LCD移植介绍》 2. 更新样式，优化结构，完善内容
V1.2	2020/12/4	1. 对“相关内容说明”章节进行补充说明 2. 更新样式，优化结构，完善内容

关键字

关键字： LCD移植、uboot、kernel。

测试平台介绍

- 本文档以SC9863A项目为例，对展锐平台LCD移植做了介绍。

本文档使用软硬件信息如下：

- 软件分支：sprdroid8.1_trunk_18b
- 硬件平台：sp9863a_3h10

目录

01

Uboot移植说明

uboot移植说明 1/7

- **添加LCD编译选项**
  - 打开uboot的配置头文件include/configs/sp9863a_3h10.h
  - 添加LCD对应的宏

```
+#define CONFIG_LCD_SPRD_NT35596_BOE_MIPI_FHD
```

- **添加驱动文件**
  - 拷贝一份现有驱动。uboot的LCD驱动文件在drivers/video/sprd/lcd/目录下，将文件名重命名为 lcd_sprd_nt35596_boe_mipi_fhd.c
  - 修改文件以下内容：
    - ✓配置panel_info结构体
    - ✓填充初始化参数init_data
    - ✓配置readid函数
    - ✓将新屏添加到supported_panel数组
    - ✓添加编译规则，将新的驱动编译到内核

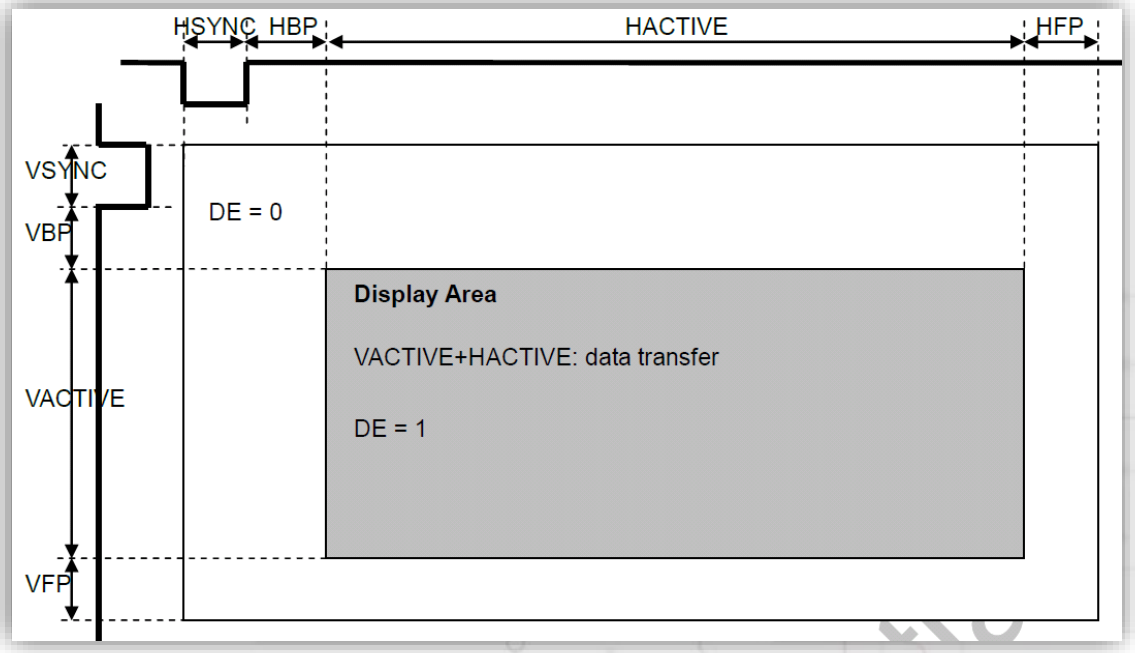
uboot移植说明 2/7

- 配置panel_info结构体

```
static struct panel_info nt35596_info = {
        /* common parameters */
        .lcd_name = "lcd_nt35596_boe_mipi_fhd",
        .type = SPRD_PANEL_TYPE_MIPI,
        .bpp = 24,                  // Bits per Pixel
        .width = 1080,              // 屏幕分辨率：宽
        .height = 1920,             // 屏幕分辨率：长

        /* DPI specific parameters */
        .pixel_clk = 153600000, /*Hz*/ // 像素时钟
        .rgb_timing = {             // Porch参数
                .hfp = 176,         // Horizontal front porch
                .hbp = 16,          // Horizontal back porch
                .hsync = 10,        // Horizontal sync
                .vfp = 32,          // vertical front porch
                .vbp = 32,          // vertical back porch
                .vsync = 4,         // vertical sync
        },

        /* MIPI DSI specific parameters */
        .phy_freq = 1000000, /*Kbps*/   // Mipi速率
        .lane_num = 4,              // Mipi data lane
        .work_mode = SPRD_MIPI_MODE_VIDEO, // MIPI传输模式
        .burst_mode = PANEL_VIDEO_BURST_MODE,
        .nc_clk_en = false,
};
```



uboot移植说明 3/7

- **初始化参数**

  把LCD厂商提供的初始化参数填充到init_data数组，结构如下:

```
static uint8_t init_data[] = {
        0x23, 0x00, 0x00, 0x02, 0xFF, 0x01,
        0x23, 0x00, 0x00, 0x02, 0xFB, 0x01,
        0x23, 0x00, 0x00, 0x02, 0x00, 0x01,
        0x23, 0x00, 0x00, 0x02, 0x01, 0x55,
        ......
        0x13, 0x78, 0x00, 0x01, 0x11,
        0x13, 0x64, 0x00, 0x01, 0x29,
        CMD_END
};
```

init_data的数据结构如下:

```
struct dsi_cmd_desc {
    uint8_t data_type; // mipi包的类型
    uint8_t wait; // 延时等待的时间（ms）
    uint8_t wc_h; // 数据包长度高8位
    uint8_t wc_l;  // 数据包长度低8位
    uint8_t payload[ ]; // data数据
};
```

0x23(data type), 0x00(延时时间), 0x00(数据包长度高8位), 0x02(数据长度低8位), 0xFF(data), 0x01(data)

uboot移植说明 4/7

紫光展锐 UNISOC

●Data数据类型

Table 16 Data Types for Processor-sourced Packets

Data Type, hex	Data Type, binary	Description	Packet Size
0x01	00 0001	Sync Event, V Sync Start	Short
0x11	01 0001	Sync Event, V Sync End	Short
0x21	10 0001	Sync Event, H Sync Start	Short
0x31	11 0001	Sync Event, H Sync End	Short
0x08	00 1000	End of Transmission packet (EoTp)	Short
0x02	00 0010	Color Mode (CM) Off Command	Short
0x12	01 0010	Color Mode (CM) On Command	Short
0x22	10 0010	Shut Down Peripheral Command	Short
0x32	11 0010	Turn On Peripheral Command	Short
0x03	00 0011	Generic Short WRITE, no parameters	Short
0x13	01 0011	Generic Short WRITE, 1 parameter	Short
0x23	10 0011	Generic Short WRITE, 2 parameters	Short
0x04	00 0100	Generic READ, no parameters	Short
0x14	01 0100	Generic READ, 1 parameter	Short
0x24	10 0100	Generic READ, 2 parameters	Short
0x05	00 0101	DCS Short WRITE, no parameters	Short
0x15	01 0101	DCS Short WRITE, 1 parameter	Short
0x3D	11 1101	Packed Pixel Stream, 12-bit YCbCr, 4:2:0 Format	Long
0x0E	00 1110	Packed Pixel Stream, 16-bit RGB, 5-6-5 Format	Long
0x1E	01 1110	Packed Pixel Stream, 18-bit RGB, 6-6-6 Format	Long
0x2E	10 1110	Loosely Packed Pixel Stream, 18-bit RGB, 6-6-6 Format	Long
0x3E	11 1110	Packed Pixel Stream, 24-bit RGB, 8-8-8 Format	Long
0xX0 and 0xXF, unspecified	XX 0000 XX 1111	DO NOT USE All unspecified codes are reserved	

uboot移植说明 5/7

紫光展锐 UNISOC

- **配置readid函数**

```
static int sprd_nt35596_readid(void)
{
        struct sprd_dsi *dsi = &dsi_device;
        uint8_t read_buf[4] = {0};

        mipi_dsi_lp_cmd_enable(dsi, true);
        mipi_dsi_set_max_return_size(dsi, 1);
        mipi_dsi_dcs_read(dsi, 0xF4, read_buf, 1);

        if (read_buf[0] == 0x96) {
                pr_info("Eric: sprd nt35596 read id success!\n");
                return 0;
        }

        pr_err("sprd nt35596 read id failed!\n");
        return -1;
}
```

1. 设置mipi为low power模式
2. 设置ID的返回个数为1
3. 设置ID寄存器为0xF4，ID值存储到read_buf数组
4. 判断是否等于0x96
5. 匹配成功则返回0，否则返回-1

uboot移植说明 6/7

- **关联到内核**

在配置头文件里添加结构体声明，并初始化supported_panel结构体。

```
diff --git a/drivers/video/sprd/lcd/panel_cfg.h b/drivers/video/sprd/lcd/panel_cfg.h
index 21c0e7a..7186ba8 100755
--- a/drivers/video/sprd/lcd/panel_cfg.h
+++ b/drivers/video/sprd/lcd/panel_cfg.h
@@ -25,8 +25,15 @@ extern struct panel_driver nt35597_boe_driver;
 extern struct panel_driver nt35597_fpga_driver;
 extern struct panel_driver jd9161_xxx_driver;
 extern struct panel_driver rm67191_edo_driver;
+extern struct panel_driver sprd_nt35596_boe_driver;

 static struct panel_cfg supported_panel[] = {
+#ifdef CONFIG_LCD_SPRD_NT35596_BOE_MIPI_FHD
+       {
+               .lcd_id = 0x96,
+               .drv = &sprd_nt35596_boe_driver,
+       },
+#endif
 #ifdef CONFIG_LCD_ILI9881C_XXX_MIPI_HD
        {
                .lcd_id = 0x9881,
```

uboot移植说明 7/7

- **添加编译规则**

在drivers/video/sprd/lcd/Makefile里添加编译规则。

```
diff --git a/drivers/video/sprd/lcd/Makefile b/drivers/video/sprd/lcd/Makefile
index 9444bbc..be1065f 100755
--- a/drivers/video/sprd/lcd/Makefile
+++ b/drivers/video/sprd/lcd/Makefile
@@ -6,7 +6,8 @@ obj-$(CONFIG_LCD_ST7701_COE_MIPI_WVGA) += lcd_st7701_coe_mipi_wvga.o
 obj-$(CONFIG_LCD_NT35532_TRULY_MIPI_FHD) += lcd_nt35532_truly_mipi_fhd.o
 obj-$(CONFIG_LCD_NT35695_TRULY_MIPI_FHD) += lcd_nt35695_truly_mipi_fhd.o
 obj-$(CONFIG_LCD_NT35596_BOE_MIPI_FHD) += lcd_nt35596_boe_mipi_fhd.o
+obj-$(CONFIG_LCD_SPRD_NT35596_BOE_MIPI_FHD) += lcd_sprd_nt35596_boe_mipi_fhd.o
 obj-$(CONFIG_LCD_NT35597_BOE_MIPI_HD) += lcd_nt35597_boe_mipi_hd.o
 obj-$(CONFIG_LCD_NT35597_FPGA_MIPI_2K) += lcd_nt35597_fpga_mipi_2k.o
 obj-$(CONFIG_LCD_JD9161_XXX_MIPI_WVGA) += lcd_jd9161_xxx_mipi_wvga.o
```

02

Kernel
移植说明

Kernel移植说明 1/2

●dtsi文件内容介绍(1)

```
/ { lcds {
        lcd_sprd_nt35596_mipi_fhd: lcd96@96 {

                work-mode = <1>;                    // mipi工作模式。
                bpp = <24>;                         // 一个像素点占用的位数
                lane-number = <4>;                  // mipi lane的数量
                need-check-esd = <0>;               // esd check未打开
                esd-timeout = <2000>;               // esd超时时间ms
                esd-check-reg = <0x0A>;             // esd check 寄存器
                esd-return-code = <0x9C>;           // esd匹配的返回值
                fps = <60>;                         // 帧率
                width-mm = <68>;                    // 屏幕宽度(mm)
                height-mm = <121>;                  // 屏幕高度(mm)
                panel-name = "boe_sprd_nt35596_mipi_fhd"; // panel name
                power-on-sequence = <               // 上电时序
                                        4  1  10
                                        5  1  20
                                        50  1  5
                                        50  0  5
                                        50  1  5
                                        50  0  5
                                        50  1  20
                                        >;
                power-off-sequence = <              // 下电时序
                                        5  0  20
                                        4  0  10
                                        50  0  20
                                        >;
        };
};};
```

拷贝一份dtsi文件并重命名为：lcd_sprd_nt35596_mipi_fhd.dtsi。

dtsi文件路径：arch/arm64/boot/dts/sprd/lcd/

- work-mode：
  - 0 - cmd mode
  - 1 - video mode
- need-check-esd：
  - 0 – 关闭esd功能
  - 1 – ESD_MODE_VIDEO
  - 2 – ESD_MODE_CMD_STOP_DPU
  - 3 – ESD_MODE_WAIT_TE
- panel-name:
  - 字符串会写入/sys/class/display/panel0/name节点
- power-on-sequence & power-off-sequence：
  - 三个数字分别代表：gpio，level，delay

Kernel移植说明 2/2

●dtsi文件内容介绍(2)

```
/ { lcds {
        lcd_sprd_nt35596_mipi_fhd: lcd96@96 {
                init-data = [           // 初始化code
                        23 00 00 02 FF 01
                        23 00 00 02 FB 01
                        23 00 00 02 00 01
                        23 00 00 02 01 55
                        ...
                        13 78 00 01 11
                        13 64 00 01 29
                        ];
                sleep-in =  [           // 进入休眠code
                        13 0A 00 01 28
                        13 78 00 01 10
                        ];
                sleep-out = [           // 唤醒屏幕code
                        13 78 00 01 11
                        13 64 00 01 29
                        ];
                display-timings {               // timing配置
                        clock-frequency = <1000000>;    // dphy clock
                        hactive = <1080>;       // 屏幕分辨率：宽
                        vactive = <1920>;       // 屏幕分辨率：长
                        hback-porch = <16>;     // horizontal back porch
                        hfront-porch = <176>;   // horizontal front porch
                        vback-porch = <32>;     // vertical back porch
                        vfront-porch = <32>;    // vertical front proch
                        hsync-len = <10>;       // horizontal sync
                        vsync-len = <4>;        // vertical sync
                };
        };
};};
```

- 引用dts文件

  arch/arm64/boot/dts/sprd/sp9863a-common.dtsi

```
#include "lcd/lcd_sprd_nt35695_mipi_fhd.dtsi"
```

- 添加panel属性

```
panel_if: sprd-panel-if {
        compatible = "sprd-panel-if";
        #address-cells = <1>;
        #size-cells = <0>;
        panel-drivers = <
                &lcd_nt35596_mipi_fhd
                &lcd_nt35695_mipi_fhd
                &lcd_sprd_nt35695_mipi_fhd
                &lcd_nt35532_2_mipi_fhd
                &lcd_rm67191_mipi_fhd
                &lcd_dummy_mipi_hd
        >;
```

03

编译烧录说明

编译烧录说明

1. 编译uboot

```
make bootloader -j10
```

2. 烧录u-boot-sign.bin

```
fastboot flash uboot u-boot-sign.bin
```

3. 编译Kernel

```
make bootimage -j10
```

4. 烧录boot.img

```
fastboot flash boot boot.img
```

04

验证方法介绍

编译烧录说明

1. 抓取串口log，找到如下信息：

```
[sprdfb][sprd_nt35596_readid] Eric: read_buf[0] = 150, read_buf[1] = 0, r
ead_buf[2] = 0, read_buf[3] = 0
[sprdfb][sprd_nt35596_readid] Eric: sprd nt35596 read id success!
[sprdfb][sprd_panel_probe] attach panel 0x96 success
```

2. 读取节点：cat /proc/cmdline

```
earlycon=sprd_serial,0x70100000,115200n8 console=ttyS1,115200n8 loglevel=
1 init=/init root=/dev/ram0 rw androidboot.hardware=s9863a3h10 swiotlb=64
k lcd_id=ID96 lcd_base=9dfd2000 lcd_size=1920x1080 pixel_clock=153600000
  sysdump_magic=85500000 modem=shutdown ltemode=lcsfb rfboard.id=0 rfhw.i
d=0 crystal=2 32k.less=1  androidboot.verifiedbootstate=green androidboot
.flash.locked=1 androidboot.vbmeta.device=PARTUUID=1.0 androidboot.vbmeta
```

3. 读取节点：cat /sys/class/display/panel0/name

```
s9863a3h10:/sys/class/display/panel0 # cat name
boe_sprd_nt35596_mipi_fhd
```

此处与dts中panel-name的内容相同。

05

常见问题及相关内容介绍

常见问题

1. 灭屏休眠后闪白屏

✓解决方法：以SL8521E为例

背光驱动代码太旧，更新代码sc2721_bltcled_bl.c（SL8521E / sprdroid4.4_sfphone_17f_rls1）

2. kernel crash，无法开机（将默认mipi接口改为spi接口）

log中包含“ion_buffer_create, carveout_fb, alloc size: 1540096 failed with freelist, ret -12”信息。

✓解决方法：计算1540096×4=0x5E0000，修改sp9820e-common-mipi.dtsi文件内fb_reserved的reg内容。

```
fb_reserved: fb@8fb3e000 {
        reg = <0x8fb3e000 0x005e0000>;
};
```

相关内容说明 1/3

紫光展锐 UNISOC

- **频率尺寸计算方式**
  - ✓屏幕尺寸，根据width-mm和height-mm计算而来。如果dts中没有这两个属性，则系统会配置为默认数值。
  - ✓screen_size等于width-mm和height-mm的平方和，再开方，单位转换为英寸。
- **dpi clk 配置方法**

```
static struct panel_info ili9881c_info = {
    /* common parameters */
    .lcd_name = "lcd_ili9881c_3lane_mipi_fhd",
    .type = SPRD_PANEL_TYPE_MIPI,
    .bpp = 24,
    .width = 720,
    .height = 1280,

    /* DPI specific parameters */
    .pixel_clk = 64000000,
    .rgb_timing = {
        .hfp = 52,
        .hbp = 26,
        .hsync = 25,
        .vfp = 5,
        .vbp = 9,
        .vsync = 2,
    },
```

- ✓ pixel_clk=(width+hporch)*(height+vporch)*fps =(720+103)*(1280+16)*60=63996480
- ✓ dpi clock时钟源参数：

```
static uint32_t dpi_clk_src[] = {
    96000000,
    100000000,
    128000000,
    153600000,
    192000000
};
```

- ✓ 按照128M / 2 = 64M来算，128M的2分频，最接近63996480。
- ✓ 所以，pixel_clk=64M正合适。

相关内容说明 2/3

紫光展锐 UNISOC

●计算mipi速率的方法

根据porch、帧率、lane个数、dividor，来计算mipi速率。

1. 计算dpi所需速率

dpi_need_clk = (Hsync + Hbp + Hactive + Hfp) * (Vsync + Vbp + Vactive + Vfp) * fps

2. dpi_clk_src的数值是dpi_need_clk的整数倍，即：

dpi_clk_src = (dpi_need_clk * dividor)

从global_dispc.c的dpi_clk_src数组里选择合适的时钟源。

3. 计算MIPI_clk

MIPI_clk * lane * 0.9 > dpi_clk_src / dividor * (8 + 8 + 8)

相关内容说明 3/3

紫光展锐 UNISOC

- **ESD Check流程**

1. 初始化等待队列wq_te_esd
2. 初始化并启动延时工作队列esd_work， 每隔esd-timeout时间调度一次。
3. 调度esd_work， 进入esd_work_func()函数， 执行等待队列， 判断条件te_esd_flag是否为true。te_esd_flag的真假由TE中断处理函数sprd_disc_isr()决定。
4. 500ms内， 如果te_esd_flag是真，则重新调度esd_work；否则超时需执行esd recovery。

谢 谢