SIM8500 Android10 LCD 移植文档

智能模组

芯讯通无线科技(上海)有限公司
上海市长宁区临虹路289号3号楼芯讯通总部大楼
电话：86-21-31575100
技术支持邮箱：support@simcom.com
官网：www.simcom.com

名称：	SIM8500_android10_lcd 移植文档
版本：	1.00
日期:	2022.03.08
状态：	已发布

版权声明

芯讯通无线科技(上海)有限公司
上海市长宁区临虹路289号3号楼芯讯通总部大楼
电话：86-21-31575100
邮箱：simcom@simcom.com
官网：www.simcom.com

了解更多资料，请点击以下链接：
http://cn.simcom.com/download/list-230-cn.html

技术支持，请点击以下链接：
http://cn.simcom.com/ask/index-cn.html 或发送邮件至 support@simcom.com

关于文档

版本历史

版本	日期	作者	备注
1.00	2022.03.08	郝夏	第一版

适用范围

本文档适用于 SIMCom SIM8500 系列。

目录

目录

1 介绍

1.1 本文目的

本文基于 SIM8500（展锐 SL8541E）Android10 平台的 lcd 驱动的移植方法。主要介绍在该平台适配一个新的 LCD 所需要的改动，以及整个流程。本文会以 jd9366 为例，从软件层面，进行详细介绍。

1.2 参考文档

[1] 34100_Yocto 平台 Kernel4.14LCD 客制化指导手册 V1.0

1.3 术语和缩写

2 屏幕厂商需要提供的资料

2.1 屏幕厂商需要提供的资料

- ✧ 屏 IC Data Sheet；
- ✧ 初始化代码(init_data),用于编写 u-boot 初始化文件和.dtsi 文件（kernel）；
- ✧ 从 data sheet 中或者向屏厂获取 width、height、hsync、hfp、hbp、vsync、vfp、vbp 参数；

2.2 代码目录

- ✧ 下边以 **jd9366** 为例介绍厂家提供的相关参数放置在代码中的位置：

source/bsp/bootloader/u-boot15/drivers/video/sprd/lcd:
├── panel_cfg.h
├── Makefile
└── lcd_jd9366_truly_mipi_hd.c
source/bsp/bootloader/u-boot15/include/configs:
└── sl8541e_1h10_32b.h
source/kernel/kernel4.14/arch/arm/boot/dts/：
├── sl8541e-1h10_32b-overlay.dts
├── sl8541e-1h10_32b.dts
├── lcd
│ ├── NR
│ ├── .
│ └── lcd_jd9366_truly_mipi_hd.dtsi

3 UBOOT 新屏配置

3.1 LCD UBOOT 新屏配置介绍

Uboot 部分的 LCD 控制，主要包括 3 个部分，配置 LCD 驱动文件、配置 LCD 编译选项、添加编译规则。下面以 sl8541e-1h10_32b 这块 board 为例，添加 jd9366 显示模组。

3.2 LCD driver 文件配置

3.2.1 添加驱动文件

- 拷贝已有驱动文件，并且按照以下命名规范重新命名 LCD 驱动文件：
  lcd_[DriverIC]_[ModuleVendor]_[BusType]_[Resolution]

  例如：lcd_jd9366_truly_mipi_hd.c
  文件路径：source/bsp/bootloader/u-boot15/drivers/video/sprd/lcd

3.2.2 修改驱动文件

- 将驱动文件中原 DDIC 名全部重新命名。当前这块 LCD 重命名为 jd9366_xxx。

- 配置初始化 code（由屏幕厂商提供）为下图格式。

```
static uint8_t init_data[] = {
        0x39, 0x00, 0x00, 0x04, 0xFF, 0x98, 0x81, 0x03,
        0x23, 0x00, 0x00, 0x02, 0x01, 0x08,
        0x23, 0x00, 0x00, 0x02, 0x02, 0x00,
        0x23, 0x00, 0x00, 0x02, 0x03, 0x73,
        0x23, 0x00, 0x00, 0x02, 0x04, 0x73,
        0x23, 0x00, 0x00, 0x02, 0x05, 0x14,
        0x23, 0x00, 0x00, 0x02, 0x06, 0x06,
        0x23, 0x00, 0x00, 0x02, 0x07, 0x02,
```

初始化code在mipi_dsi_send_cmds函数中将强制转换为dsi_cmd_desc结构体。

```
struct dsi_cmd_desc {
    uint8_t data_type; //mipi包的类型
    uint8_t wait;      //延时时间(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
};
```

➢ 配置 panel_info 结构体（参数由屏幕厂商提供）

```
static struct panel_info jd9366_info = {
        /* common parameters */
        .lcd_name = "lcd_jd9366_truly_mipi_hd",
        .type = SPRD_PANEL_TYPE_MIPI,
        .bpp = 24,
        //.fps = 60,
        .width = 800,
        .height = 1280,

        /* DPI specific parameters */
        .pixel_clk = 48000000,
        .rgb_timing = {
                .hfp = 24,
                .hbp = 24,
                .hsync = 4,
                .vfp = 8,
                .vbp = 8,
                .vsync = 4,
        },

        /* MIPI DSI specific parameters */
        .phy_freq = 500000,
        .lane_num = 4,
        .work_mode = SPRD_MIPI_MODE_VIDEO,
        .burst_mode = PANEL_VIDEO_BURST_MODE,
        .nc_clk_en = false,
};
```

说明:

lcd_name: lcd 节点名，必须和 kernel 中的 lcd dts 节点名称保持一致，否则 kernel 无法匹配到正确的 lcd 驱动。

可通过展锐计算文档“33340_unisoc lcd(mipi) fps-phy_feq 计算_c_v1.05”计算出 phy_freq 和 pixel_clk;

➢ 实现 panel_ops 回调函数

```
static struct panel_ops jd9366_ops = {
        .init = jd9366_init,
        .read_id = jd9366_readid,
        .power = jd9366_power,
};
```

➢ 注册 panel driver

```
struct panel_driver jd9366_truly_driver = {
        .info = &jd9366_info,
        .ops = &jd9366_ops,
};
```

3.3 LCD 添加编译规则

- 找到并打开驱动文件所在目录的 Makefile：

  source/bsp/bootloader/u-boot15/drivers/video/sprd/lcd/Makefile。

- 在 Makefile 中添加编译规则。

```
obj-$(CONFIG_LCD_JD9366_TRULY_MIPI_HD) += lcd_jd9366_truly_mipi_hd.o
```

  其中 CONFIG_LCD_ILI9881C_TRULY_MIPI_HD 在 board 头文件中定义。

- 找到并打开屏幕配置头文件：

  source/bsp/bootloader/u-boot15/drivers/video/sprd/lcd/panel_cfg.h。

- 在 panel_cfg.h 中添加结构体声明。

```
extern struct panel_driver jd9366_truly_driver;
```

- 在 panel_cfg.h 中添加 panel 结构体:

  supported panel 实例化:

```
#ifdef CONFIG_LCD_JD9366_TRULY_MIPI_HD
        {
                .lcd_id = 0x9366,
                .drv = &jd9366_truly_driver,
        },
#endif
```

将驱动结构体关联到 supported_panel 中。在开机时，uboot 会枚举此结构体中的屏幕，直到某个 driver 的 readid 函数返回成功。

3.4 LCD 编译选项配置

LCD Driver 文件添加和配置完成后，就可以在 board 配置文件中打开添加的新的 LCD 模组了：

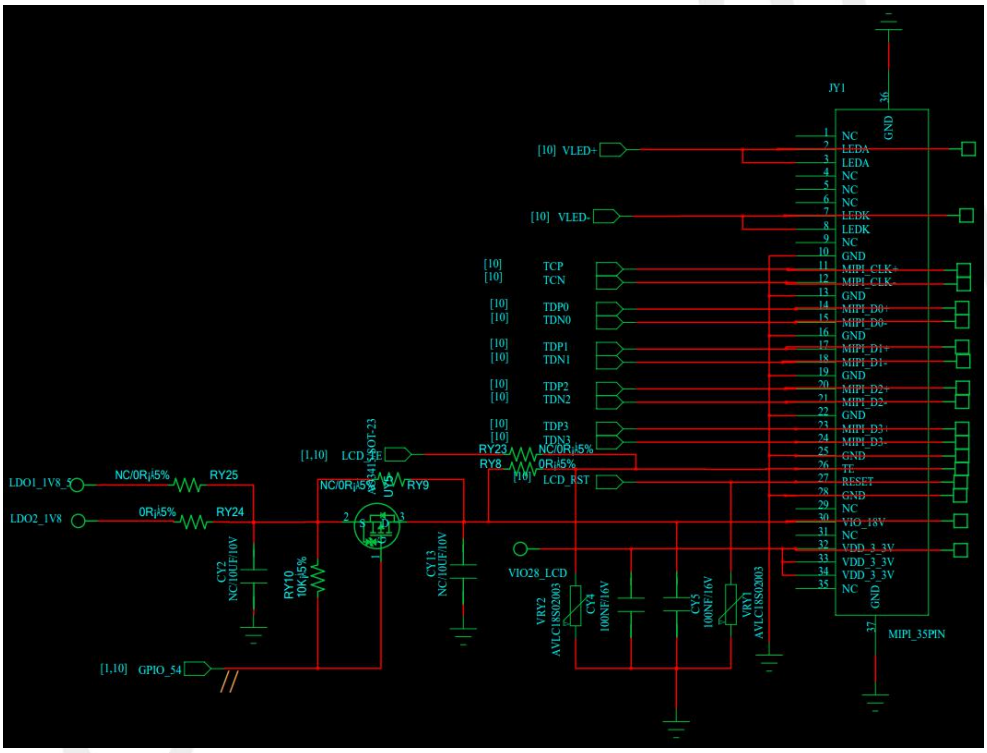
- 编译选项通常放在 uboot 的 board 配置头文件中
  source/bsp/u-boot15_sprdroidq/include/configs/sl8541e_1h10_32b.h
- 编辑 panel 配置信息

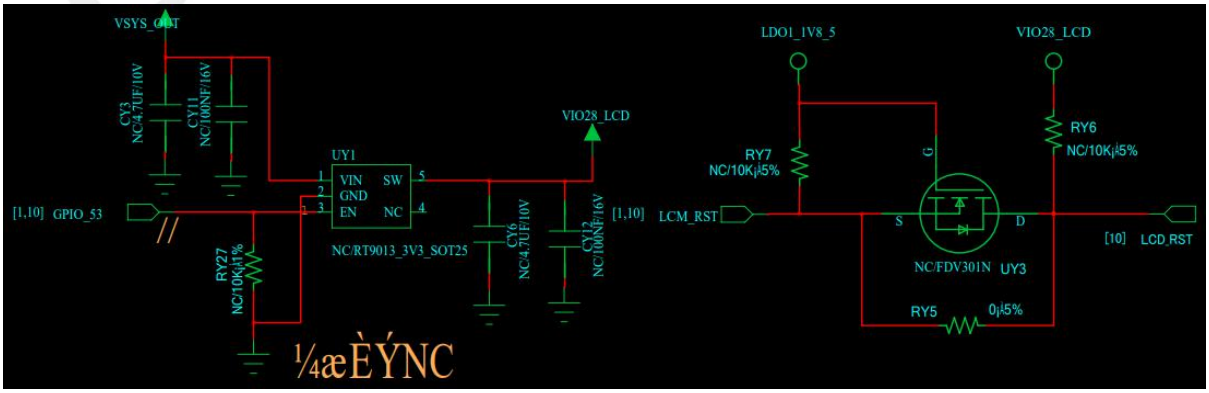
添加配置宏：#define CONFIG_LCD_JD9366_TRULY_MIPI_HD

```
/*lcd config*/
#define CONFIG_LCD
#define CONFIG_CMD_BMP
#define CONFIG_SPLASH_SCREEN
#define CONFIG_VIDEO_SPRD
#define CONFIG_DPU_LITE_R1P0
#define CONFIG_DSIH_SPRD_CTRL_RXP0
#define CONFIG_DPHY_SPRD_SHARKLE
#define CONFIG_LCD_ILI9881C_TRULY_MIPI_HD
#define CONFIG_LCD_JD9366_TRULY_MIPI_HD
#define CONFIG_LCM_GPIO_RSTN      50
#define CONFIG_SYS_WHITE_ON_BLACK
#define LCD_BPP LCD_COLOR16
```

3.5 LCD 供电配置

- 相关原理图：





- 由原理图可知，需要把 gpio53，gpio54 拉高，所以需要在 jd9366_power 函数中，在上电的时候将其拉高；

```
static int jd9366_power(int on)
{
        if (on) {
                regulator_set_voltage("vddcamio",1800);
                regulator_enable("vddcamio");
                mdelay(5);

                sprd_gpio_request(NULL, 53);
                sprd_gpio_direction_output(NULL, 53, 1);
                mdelay(5);

                sprd_gpio_request(NULL, 54);
                sprd_gpio_direction_output(NULL, 54, 1);
                mdelay(5);

                sprd_gpio_request(NULL, CONFIG_LCM_GPIO_RSTN);
                sprd_gpio_direction_output(NULL, CONFIG_LCM_GPIO_RSTN, 1);
                mdelay(5);
                sprd_gpio_direction_output(NULL, CONFIG_LCM_GPIO_RSTN, 0);
                mdelay(50);
                sprd_gpio_direction_output(NULL, CONFIG_LCM_GPIO_RSTN, 1);
                mdelay(20);
                pr_err("simcom reset1!\n");
        } else {
                sprd_gpio_direction_output(NULL, CONFIG_LCM_GPIO_RSTN, 0);
                mdelay(5);
                pr_err("simcom reset0!\n");
        }

        return 0;
}
```

4 Kernel 配置介绍

4.1 LCD DTS 文件配置

4.1.1 添加 dts 文件

拷贝已有驱动文件，并且按照以下命名规范重新命名 LCD 驱动文件：
lcd_[DriverIC]_[ModuleVendor]_[BusType]_[Resolution]

例如：lcd_jd9366_truly_mipi_hd.dtsi
文件路径：source/kernel/kernel4.14/arch/arm/boot/dts/lcd

4.1.2 添加 dts 文件

- 将 dtsi 文件中原 DDIC 名全部重新命名。当前这块 LCD 重命名为 jd9366_xxx。

- 配置初始化 code（由屏幕厂商提供）为如图 3-2 格式。

```
sprd,initial-command = [
        39 00 00 04 FF 98 81 03
        23 00 00 02 01 08
        23 00 00 02 02 00
        23 00 00 02 03 73
        23 00 00 02 04 73
        23 00 00 02 05 14
        23 00 00 02 06 06
        23 00 00 02 07 02
        23 00 00 02 08 05
```

```
struct dsi_cmd_desc {
        uint8_t data_type; //mipi包的类型
        uint8_t wait;      //延时时间(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
};
```

➢ dts 配置属性详细说明下图：

```
/ { lcds {
        lcd_jd9366_truly_mipi_hd: lcd_jd9366_truly_mipi_hd {

                sprd,dsi-work-mode = <1>;
                sprd,dsi-lane-number = <4>;
                sprd,dsi-color-format = "rgb888";

                sprd,phy-bit-clock = <500000>;
                sprd,phy-escape-clock = <20000>;

                sprd,width-mm = <107>;
                sprd,height-mm = <172>;

                sprd,esd-check-enable = <0>;
                sprd,esd-check-mode = <1>;
                sprd,esd-check-period = <1000>;

                sprd,reset-on-sequence = <1 5>, <0 50>, <1 20>;
                sprd,reset-off-sequence = <0 20>;

                sprd,initial-command = [
                        39 00 00 02 E0 00
                        39 00 00 02 E1 93
                        39 00 00 02 E2 65
                        39 00 00 02 E3 F8
                        39 00 00 02 09 10
                        39 00 00 02 2D 03
                        39 00 00 02 E0 00
                        05 78 00 01 11
                        05 78 00 01 29
                        ];
                sprd,sleep-in-command = [
                        13 0A 00 01 28
                        13 78 00 01 10
                        ];
                sprd,sleep-out-command = [
                        13 78 00 01 11
                        13 64 00 01 29
                        ];
                display-timings {
                        timing0 {
                                clock-frequency = <48000000>;
                                hactive = <800>;
                                vactive = <1280>;
                                hback-porch = <24>;
                                hfront-porch = <24>;
                                vback-porch = <8>;
                                vfront-porch = <8>;
                                hsync-len = <4>;
                                vsync-len = <4>;
                        };
                };
        };
;};
```

可通过展锐计算文档“33340_unisoc lcd(mipi) fps-phy_feq 计算_c_v1.05”计算出 phy_freq 和 pixel_clk；

说明：当前 dts 节点 lcd_jd9366_truly_mipi_hd 要和 uboot 配置中的 lcd_name 要保持一致，否则 kernel 无法匹配到正确的 lcd 驱动。

➢ 引用配置好的 dts 文件。

找到并打开 board 配置头文件，并且添加到 board 配置文件中：
source/kernel/kernel4.14/arch/arm/boot/dts/sl8541e-1h10_32b.dts

```
#include "lcd/lcd_jd9366_truly_mipi_hd.dtsi"
```

通过#include 将需要适配的 lcd 外设与该 board 关联上，这样 panel 驱动就会根据 uboot 传上来的 lcd_name 去"/lcds"节点下查找相应的 lcd 子节点，从而完成 lcd 的匹配。

➢ 配置和 LCD 模组相关的 GPIO。

找到并打开 board 配置头文件，并且添加到 board 配置文件中：
source/kernel/kernel4.14/arch/arm/boot/dts/sl8541e-1h10_32b.dts

```
&dsi {
        status = "okay";
        #address-cells = <1>;
        #size-cells = <0>;

        panel: panel {
                compatible = "sprd,generic-mipi-panel";
                #address-cells = <1>;
                #size-cells = <0>;
                reg = <0>;

                power-supply = <&vddcamio>;
                avdd-gpio = <&ap_gpio 53 GPIO_ACTIVE_HIGH>;
                //avee-gpio = <&ap_gpio 54 GPIO_ACTIVE_HIGH>;
                reset-gpio = <&ap_gpio 50 GPIO_ACTIVE_HIGH>;
                port {
                        reg = <1>;
                        panel_in: endpoint {
                                remote-endpoint = <&dphy_out>;
                        };
                };
        };
};
```

目前支持 3 个 GPIO，分别为 reset-gpio、avdd-gpio 和 avee-gpio；avdd-gpio 和 avee-gpio 是根据原理图实际情况去进行配置，相关的供电就使用 avdd-gpio 和 avee-gpio 在内核中进行操作；电压为 2.8V 则使用 avdd-gpio 配置，电压为 1.8V 则使用 avee 进行配置；

4.2 Kernel 背光配置

4.2.1 配置 LCD 背光属性

➢ 找到并打开 overlay board 配置头文件，添加 sprd_backlight 属性：
文件名：sl8541e-1h10_32b-overlay.dts
文件路径：source/kernel/kernel4.14_sprdroidq/arch/arm/boot/dts

```
pwm_backlight: sprd_backlight {
        compatible = "pwm-backlight";
        pwms = <&pwms 1 20000>;
        pwm-names = "backlight";
        brightness-levels = <
                                                  2   2   2   2
                  2   2   2   3   4   5   6   7   8   9
                 10  11  12  13  14  15  16  17  18  19
                 20  21  22  23  24  25  26  27  28  29
                 30  31  32  33  34  35  36  37  38  39
                 40  41  42  43  44  45  46  47  48  49
                 50  51  52  53  54  55  56  57  58  59
                 60  61  62  63  64  65  66  67  68  69
                 70  71  72  73  74  75  76  77  78  79
                 80  81  82  83  84  85  86  87  88  89
                 90  91  92  93  94  95  96  97  98  99
                100 101 102 103 104 105 106 107 108 109
                110 111 112 113 114 115 116 117 118 119
                120 121 122 123 124 125 126 127 128 129
                130 131 132 133 134 135 136 137 138 139
                140 141 142 143 144 145 146 147 148 149
                150 151 152 153 154 155 156 157 158 159
                160 161 162 163 164 165 166 167 168 169
                170 171 172 173 174 175 176 177 178 179
                180 181 182 183 184 185 186 187 188 189
                190 191 192 193 194 195 196 197 198 199
                200 201 202 203 204 205 206 207 208 209
                210 211 212 213 214 215 216 217 218 219
                220 221 222 223 224 225 226 227 228 229
                230 231 232 233 234 235 236 237 238 239
                240 241 242 243 244 245 246 247 248 249
                250 251
                >;
        default-brightness-level = <33>;
};
```

3.2.2 引用 LCD 背光属性

引用 backlight 节点

```
&panel {
        sprd,backlight = <&pwm_backlight>;
};
```

5 编译

- 编译 uboot:

修改完 uboot 下的驱动需要 make bootloader。
下载通过 fastboot flash uboot u-boot-sign.bin

- 编译 kernel:

修改完 kernel 下的驱动需要 make bootimage，make dtboimage。
下载通过 fastboot flash boot boot.img
fastboot flash dtbo dtbo.img

6 DEBUG

- 读取 ID 成功的 LOG:

```
pr_info("jd9366 read id success!\n");
pr_err("simcom0 jd9366 read id read_buf[0]=0x%x,read_buf[1]=ox%x!\n",read_buf[0],read_buf[1]);
```

- 读取 ID 失败的 LOG:

```
pr_err("simcom1 jd9366 read id read_buf[0]=0x%x,read_buf[1]=ox%x!\n",read_buf[0],read_buf[1]);
pr_err("jd9366 read id failed!\n");
```

- 以上 log 可以到 uboot 下的.c 文件中查看相应的代码：

```
static int jd9366_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, 2);
        mipi_dsi_dcs_read(dsi, 0x04, read_buf, 2);

        if ((read_buf[0] == 0x93) && (read_buf[1] == 0x66)) {
                pr_info("jd9366 read id success!\n");
                pr_err("simcom0 jd9366 read id read_buf[0]=0x%x,read_buf[1]=0x%x!\n",read_buf[0],read_buf[1]);
                return 0;
        }
        pr_err("simcom1 jd9366 read id read_buf[0]=0x%x,read_buf[1]=0x%x!\n",read_buf[0],read_buf[1]);
        pr_err("jd9366 read id failed!\n");
        return -1;
}
```

7 QA

Q1：uboot 阶段，LCD 未被点亮，需要如何排查？
A1：1）添加 log，查看 lcd id 是否读取正常，保证走正常的 init 流程；（具体读哪个寄存器，以及读出来的寄存器值是什么，都需要和 LCD 厂商确认。）
2）查看 porch 值、clk、freq 是否正常。

Q2：kernel 阶段，LCD 未被点亮，需要如何排查？
A2：查看 porch 值、clk、freq 是否正常。clk、freq 值需要使用展锐的计算文档算出响应的值。

Q3：LCD 无法显示 ，还有什么排查思路？
A3：根据原理图，相应的供电也需要进行配置。可以使用万用表，量一下相应的引脚，确认供电是否正常。

Q4：LCD 可以显示，但是显示的图像不完整，有黑边，是什么原因？
A4：这个不是 LCD 显示问题，是开机图片的大小问题，和实际的 LCD 分辨率不一致导致，需要更换相应分辨率的开机图片即可解决。