

A20 LCD 模块开发说明

V1.0

2013-03-15

Revision History

| Version | Date | Changes compared to previous issue |
|---------|------------|------------------------------------|
| v1.0 | 2013-03-15 | 初建版本 |
| | | |
| | | |
| | | |

目录

| | |
|-----------------------------|----|
| 1. 前言 | 6 |
| 1.1. 编写目的 | 6 |
| 1.2. 适用范围 | 6 |
| 1.3. 相关人员 | 6 |
| 2. 模块介绍 | 7 |
| 2.1. 模块功能介绍 | 7 |
| 2.2. 硬件介绍 | 7 |
| 2.3. 源码结构介绍 | 7 |
| 3. LCD 配置流程介绍 | 8 |
| 3.1. sys_config.fex | 8 |
| 3.1.1. LCD 接口参数 | 8 |
| 3.1.2. LCD 时序参数 | 9 |
| 3.1.3. LCD 功能参数 | 9 |
| 3.1.4. 电源背光及 IO | 9 |
| 3.2. lcd_panel_cfg.c | 10 |
| 3.2.1. LCD 屏的初始化 | 10 |
| 3.3. linux 阶段 LCD 配置 | 10 |
| 3.3.1. 修改屏文件 | 10 |
| 3.3.2. 编译 | 10 |
| 3.3.3. 打包 | 10 |
| 3.4. boot 阶段 LCD 配置 | 10 |
| 3.4.1. 修改屏文件 | 10 |
| 3.4.2. 编译 | 10 |
| 3.4.3. boot 驱动更新 | 11 |
| 3.5. A10 到 A20 的快速移植 | 13 |
| 3.5.1. A10 原有参数修改 | 13 |
| 3.5.2. 屏文件(lcd_panel_cfg.c) | 13 |
| 4. LCD 硬件参数说明 | 15 |
| 4.1. LCD 接口参数说明 | 15 |
| 4.1.1. lcd_if | 15 |
| 4.1.2. lcd_hv_if | 15 |
| 4.1.3. lcd_hv_s888_if | 15 |
| 4.1.4. lcd_hv_syuv_if | 16 |
| 4.1.5. lcd_cpu_if | 16 |
| 4.1.6. lcd_lvds_ch | 16 |
| 4.1.7. lcd_lvds_bitwidth | 16 |
| 4.1.8. lcd_lvds_mode | 16 |
| 4.2. LCD 时序参数说明 | 18 |
| 4.2.1. lcd_x | 18 |
| 4.2.2. lcd_y | 18 |



| | |
|--------------------------------|----|
| 4.2.3. lcd_ht | 18 |
| 4.2.4. lcd_hbp | 18 |
| 4.2.5. lcd_hspw | 18 |
| 4.2.6. lcd_vt | 18 |
| 4.2.7. lcd_vbp | 19 |
| 4.2.8. lcd_vspw | 19 |
| 4.2.9. lcd_dclk_freq | 19 |
| 4.3. LCD 其他参数说明 | 19 |
| 4.3.1. lcd_pwm_freq | 19 |
| 4.3.2. lcd_pwm_pol | 19 |
| 4.3.3. lcd_frm | 20 |
| 4.3.4. lcd_gamma_correction_en | 22 |
| 4.3.5. lcd_gamma_tbl | 23 |
| 4.4. POWER 及 IO 说明 | 23 |
| 4.4.1. lcd_power | 23 |
| 4.4.2. lcd_bl_en | 23 |
| 4.4.3. lcd_pwm | 23 |
| 4.4.4. lcd_gpio_x | 23 |
| 4.4.5. lcd_pin | 24 |
| 5. LCD 软件配置说明 | 25 |
| 5.1. 屏文件说明 | 25 |
| 5.2. 开关屏流程 | 27 |
| 5.2.1. 开关屏步骤函数说明 | 28 |
| 5.2.2. 开关屏流程函数说明 | 29 |
| 5.3. 屏的初始化 | 30 |
| 5.3.1. CPU/I80 屏的初始化 | 30 |
| 5.3.2. 使用 IO 模拟串行接口初始化 | 31 |
| 5.3.3. 使用 SPI 接口初始化 | 32 |
| 5.3.4. DSI 屏的初始化 | 32 |
| 5.4. 其它函数 | 33 |
| 5.4.1. 用户自定义函数 | 33 |
| 5.4.2. 延时函数 | 34 |
| 6. 模块开发 demo | 35 |
| 7. 模块调试 | 36 |
| 8. 附录 | 37 |
| 8.1. A20 与 HV Parallel RGB 屏 | 37 |
| 8.1.1. schematic | 37 |
| 8.1.2. sys_config.fex | 39 |
| 8.2. A20 与 HV Serial RGB 屏 | 41 |
| 8.2.1. schematic | 41 |
| 8.2.2. sys_config.fex | 43 |
| 8.2.3. lcd0_panel_cfg.c | 45 |

| | |
|--------------------------------------|----|
| 8.3. A20 与 CPU Parallel 18bit 屏----- | 46 |
| 8.3.1. schematic----- | 46 |
| 8.3.2. sys_config.fex----- | 48 |
| 8.3.3. lcd0_panel_cfg.c----- | 49 |
| 8.4. A20 与 LVDS Single Link 屏----- | 51 |
| 8.4.1. schematic----- | 51 |
| 8.4.2. sys_config.fex----- | 54 |
| 8.5. A20 与 LVDS Dual Link 屏----- | 55 |
| 8.5.1. schematic----- | 55 |
| 8.5.2. sys_config.fex----- | 57 |
| 8.6. A20+SSD2828 与 DSI 屏----- | 59 |
| 8.6.1. schematic----- | 59 |
| 8.6.2. sys_config.fexs----- | 60 |
| 8.6.3. lcd0_panel_cfg.c----- | 61 |
| 8.7. 电源和背光----- | 64 |
| 8.8. A20 LCD IO PORT 定义----- | 65 |
| 8.9. LCD CHECK LIST----- | 67 |
| 9. Declaration----- | 68 |

1. 前言

1.1. 编写目的

了解 LCD 模块在 A20 平台上的配置、使用。

1.2. 适用范围

适用于 A20 平台

1.3. 相关人员

与显示相关的软硬件开发人员。

2. 模块介绍

介绍本模块的模块功能，基本配置，目标代码的文件目录组织形式以及相关的硬件介绍。

2.1. 模块功能介绍

A20 有两路显示系统，支持双屏输出。支持的 LCD 屏接口形式及最大分辨率如表 1-1 所示。

并行数据输出，TTL 电平的屏接口，如 HV RGB 接口和 CPU/I80 接口，LCD0 从 PD 口输出，LCD1 从 PA 口输出。

串行输出，差分信号的屏接口，如 LVDS 接口，LVDS0 从 PD0-PD9 输出，LVDS1 都是从 PD10-PD19 口输出。

DSI 接口，使用 A20+SSD2828 的方式，通过 A20 的 HV RGB 接口输出，由转换 IC 转成 DSI 接口。

如果一路输出使用 Dual Link LVDS，占用了 LVDS 的所有引脚，另外一路只能使用其他接口形式输出；其他接口形式任意组合的双屏输出都支持。

表 1-1 A20 LCD 输出 IO 口

| Interface | | Solutions | LCD0 | LCD1 |
|-----------|---------------------|-------------|----------|-----------|
| HV RGB | Parallel RGB | 2048 × 1536 | PD | PA |
| | Serial RGB/CCIR | 1280 × 720 | | |
| CPU/80 | Parallel 18bit | 1280 × 720 | PD | PA |
| | Parallel 16bit | | | |
| LVDS | Single Link | 1440 × 900 | PD0-PD9 | PD10-PD19 |
| | Dual Link | 1920 × 1200 | PD0-PD19 | |
| DSI | 4Lane (A20+SSD2828) | 1920 × 1200 | PD | PA |

2.2. 硬件介绍

LCD 相关参考电路见附录。

2.3. 源码结构介绍

lichee\linux-3.3\drivers\video\sun7i\lcd

3. LCD 配置流程介绍

3.1. sys_config.fex

3.1.1. LCD 接口参数

对 LCD 接口参数的配置可以参考表 1-2。确定好接口类型，最后确保对应的一行的相关参数都有配置。每个参数的详细解释见”2.1LCD 接口参数说明”

表 1-2 LCD 接口参数配置说明

| Interface | | Parameter |
|---------------------------|--------------------------------------|---|
| RGB 接口 (lcd_if=0) | Parallel RGB (lcd_hv_if=0) | lcd_hv_clk_phase lcd_hv_sync_polarity |
| | Serial RGB (lcd_hv_if=8) | lcd_hv_clk_phase lcd_hv_sync_polarity lcd_hv_srgb_seq |
| | Dummy RGB (lcd_hv_if=10) | |
| | RGB Dummy (lcd_hv_if=11) | |
| | Serial YUV/CCIR656 (lcd_hv_if=12) | lcd_hv_syuv_seq lcd_hv_syuv_fdly |
| CPU/8080 接口 (lcd_if=1) | Parallel 18bit (lcd_cpu_if=0) | lcd_cpu_te |
| | Parallel 16bit (lcd_cpu_if=4) | |
| LVDS 接口 (lcd_if=3) | Single channel (lcd_lvds_if=0) | lcd_lvds_bitwidth lcd_lvds_mode |
| | Dual channel (lcd_lvds_if=1) | |
| DSI 接口 (lcd_if=4) | A20+SSD2828 | lcd_hv_clk_phase lcd_hv_sync_polarity |

3.1.2. LCD 时序参数

参考对应的 LCD 屏规格书，在满足 A20 的 LCD 控制器条件下，配置 LCD 时序参数。参数详细说明详见“2.2 LCD 时序参数说明”。

A20 LCD 时序参数总共有 9 个。lcd_x, lcd_y, lcd_ht, lcd_hbp, lcd_hspw, lcd_vt, lcd_vbp, lcd_vspw, lcd_dclk_freq。

在 A20 的 LCD 控制器中，必须满足如下条件：

- a、lcd_hbp > lcd_hspw;
- b、lcd_ht > lcd_x×cycle+lcd_hbp，即是 lcd_hfp>0；cycle 为每个 pixel 的 cycle 数；
- c、lcd_vbp > lcd_vspw；
- d、lcd_vt > lcd_y+lcd_vbp；即是 lcd_vfp>0。

hbp 在部分屏规格书的定义中并不包括 hspw。这种情况下，要正确配置 A20 的 LCD 控制器，hbp(A20)=hbp(panel)+hspw(panel)。vbp 的定义同 hbp。

3.1.3. LCD 功能参数

配置 LCD_FRM, LCD_GAMMA, LCD_CMAP 参数。

若 LCD 像素的格式为 RGB666，建议 lcd_frm=1；若 LCD 像素的格式为 RGB565，建议 lcd_frm=2。

若 LCD 屏需要 Gamma 校正，配置 lcd_gamma_en=1 和 lcd_gamma_tbl。

若 LCD 屏 RGB 需重新排列，配置 lcd_cmap_en=1 和 lcd_cmap_tbl。

详细说明请参见“2.3 LCD 其他参数说明”。

3.1.4. 电源背光及 IO

根据实际方案的设计原理图配置电源背光及相应 IO，详细说明请参见“2.4 POWER 及 IO 说明”。

背光参数：lcd_pwm_freq, lcd_pwm_pol, lcd_pwm_max_limit;

电源背光 IO：lcd_power (默认 power2), lcd_pwm, lcd_bl_en;

LCD 初始化 IO：lcd_gpio_x；详细说明请参见“2.4.4”；

LCD PIN:lcdd0~lcdd23, lcddclk, lcddde, lcdhsync, lcdvsync。LCD PIN 配置，所有 LCD 接口屏都必须配置 LCD PIN。LCD0 配置为 PD 口，功能选择为 2；LCD1 配置为 PA 口，功能选择为 3。

3.2. lcd_panel_cfg.c

3.2.1. LCD 屏的初始化

需要初始化的 LCD 屏，请修改 C 文件中的 LCD_panel_init 函数。不同接口的 LCD 屏，驱动提供有不同的接口函数，详细说明请参见“3.3 LCD 屏的初始化”。

3.3. linux 阶段 LCD 配置

3.3.1. 修改屏文件

```
..\lichee\linux-3.3\drivers\video\sun7i\lcd\lcd0_panel_cfg.c  
..\lichee\linux-3.3\drivers\video\sun7i\lcd\lcd1_panel_cfg.c
```

3.3.2. 编译

使用 putty login，执行

```
$ cd ~/exdroid/lichee  
$ ./build.sh -p sun7i_android
```

3.3.3. 打包

如果是 Linux 固件，执行

```
$ cd ~/exdroid/lichee;  
$ ./build.sh pack
```

如果是 android 固件，执行

```
$ cd ~/exdroid/android4.2.2;  
$ pack
```

3.4. boot 阶段 LCD 配置

3.4.1. 修改屏文件

```
...\boot\boot1\driver\drv_de\de_bsp\lcd\lcd0_panel_cfg.c  
...\boot\boot1\driver\drv_de\de_bsp\lcd\lcd1_panel_cfg.c
```

3.4.2. 编译

使用 putty login，执行

```
$ cd ~/exdroid/lichee/boot/boot1/driver/drv_de;  
$ make clean;  
$ make  
生成 drv_de.drv, 位于:  
...\lichee/tools/pack/chips/sun7i/wboot/bootfs/drv_de.drv
```

3.4.3. boot 驱动更新

有两种方法，一种方法是打包重新升级：

```
$ cd ~/exdroid/android4.2.2;  
$ pack
```

一种方法是直接替换小机里 `drv_de.drv`。

方法是先断电，接上串口，接上 USB 线，按住 PC 键盘的”1”然后开机，会进入 U 盘模式。这时打印信息如图 1-1 所示。

将...\wboot\bootfs\drv_de.drv 文件拷贝进名称为”Volumn”的磁盘里,如图 1-2 所示。最后重启系统，完成 `drv_de.drv` 的更新。



```
boot0 version : 2.0.0
reg_val=0x00000000
[DRAM 0.99] clk = 240
Found PMU:AXP221
ddr vol 1500mV
dram size =2048
Ready to disable icache.
Succeed in loading Boot1.
Jump to Boot1.
[ 0.170] boot1 version : 2.0.0
[ 0.176] script installed ok
[ 0.176] PMU: AXP221
[ 0.176] bat ratio = 100
[ 0.179] dc3 1200
[ 0.224] pll1 1008
[ 0.267] power finish
[ 0.268] key
[ 0.279] no key found
[ 0.283] dc1 3000
[ 0.283] dc2 1200
[ 0.283] dc4 1200
[ 0.284] dc5 1500
[ 0.287] flash init start
[ 1.957] flash init finish
[ 1.993] fs init ok
[ 1.995] fattype FAT16
[ 1.995] fs mount ok
[ 2.008] dram_para_set start
[ 2.012] dram_para_set end
[ 2.012] type=48
[ 2.039] 49
[ 2.040] part count = 2
[ 2.040] USB Device!!
[ 2.042] USB Connect!!
[ 2.097] uSuspend
[ 2.216] uSuspend
[ 2.274] uSuspend
[ 2.308] usb_device: Set Address 0x0x00000004
[ 2.517] usb_device: Get MaxLUN
```

图 1-1 U 盘模式



图 1-2 volumn 目录

3.5. A10 到 A20 的快速移植

如果方案是从 A10 移植到 A20 上的，可参照以下操作步骤快速地把 LCD 在 A20 平台上 work。

建议 LCD 参数在 A20 提供的 sys_config 模板上修改，模板可参考“附录”。

3.5.1. A10 原有参数修改

Lcd_hspw(A20) = lcd_hv_hspw(A10)

Lcd_vspw(A20) = lcd_hv_hspw(A10)

其余参数跟 A10 一致，可以直接使用。

3.5.2. 屏文件(lcd_panel_cfg.c)

屏文件，A20 相对 A10 有了一些变化。建议使用 A20 的默认屏文件，在上面修改初始化函数及 gamma_tbl 参数。

A20 中，几乎所有的参数都需要在 sys_config.fex 中配置，在屏文件里，只可以配置

Confidential

4. LCD 硬件参数说明

4.1. LCD 接口参数说明

4.1.1. lcd_if

Lcd Interface

设置相应值的对应含义为：

- 0: HV RGB 接口
- 1: CPU/I80 接口
- 2: Reserved
- 3: LVDS 接口
- 4: DSI 接口

4.1.2. lcd_hv_if

Lcd HV panel Interface

这个参数只有在 lcd_if=0 时才有效。定义 RGB 同步屏下的几种接口类型。
设置相应值的对应含义为：

- 0: Parallel RGB
- 8: Serial RGB
- 10: Dummy RGB
- 11: RGB Dummy
- 12: Serial YUV (CCIR656)

4.1.3. lcd_hv_s888_if

Lcd HV panel Serial RGB output Sequence

这个参数只有在 lcd_if=0 且 lcd_hv_if=1 (Serial RGB) 时才有效。
定义奇数行 RGB 输出的顺序

- 0: Odd lines R→G→B; Even line R→G→B
- 1: Odd lines B→R→G; Even line R→G→B
- 2: Odd lines G→B→R; Even line R→G→B
- 4: Odd lines R→G→B; Even line B→R→G
- 5: Odd lines B→R→G; Even line B→R→G
- 6: Odd lines G→B→R; Even line B→R→G
- 8: Odd lines R→G→B; Even line G→B→R
- 9: Odd lines B→R→G; Even line G→B→R
- 10: Odd lines G→B→R; Even line G→B→R

4.1.4. lcd_hv_syuv_if

Lcd HV panel Serial YUV output Sequence

这个参数只有在 lcd_if=0 且 lcd_hv_if=2 (Serial YUV) 时才有效。

定义 YUV 输出格式

- 0: YUYV
- 1: YVYU
- 2: UYVY
- 3: VYUY

4.1.5. lcd_cpu_if

Lcd CPU panel Interface

这个参数只有在 lcd_if=1 时才有效。

设置相应值的对应含义为：

- 0: 18bit/1cycle parallel (RGB666)
- 4: 16bit/1cycle parallel (RGB565)
- 6: 18bit/3cycle parallel (RGB666)
- 7: 16bit/2cycle parallel (RGB565)

4.1.6. lcd_lvds_ch

Lcd LVDS panel Interface

设置相应值的对应含义为：

- 0: Single Link
- 1: Dual Link

LVDS 接口的 LCD 屏，定义 1 clock pair+3/4 data pair，为 1 个 link。若有 2 clock pair，则为 dual link。

4.1.7. lcd_lvds_bitwidth

Lcd LVDS panel color depth

设置相应值对应含义为：

- 0: 8bit per color
- 1: 6bit per color

若每 link 有 4 data pair，则为 8bit per color，若每 link 有 3 data pair，则为 6bit per color。

4.1.8. lcd_lvds_mode

Lcd LVDS Mode

这个参数只有在 lcd_lvds_bitwidth=0 时才有效

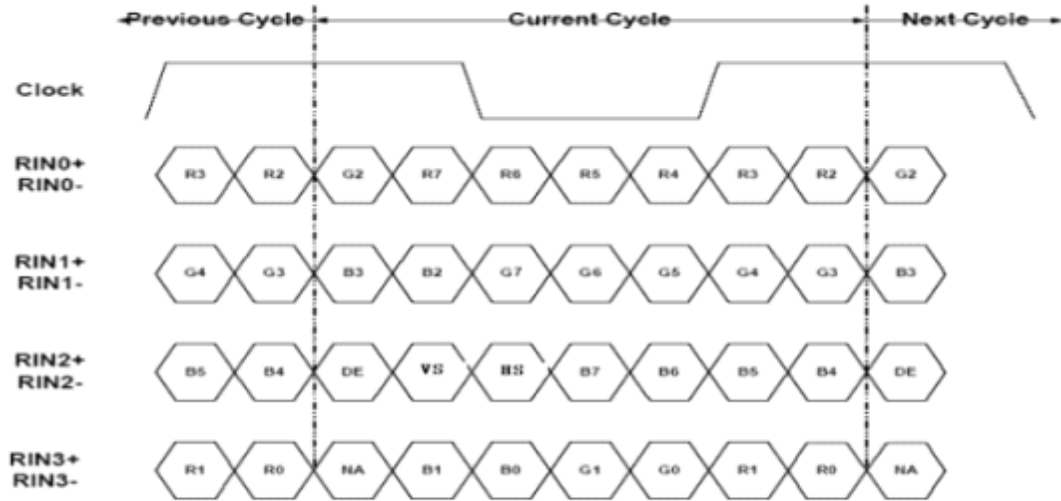
设置相应值对应含义为：

0: NS mode

1: JEIDA mode

NS mode 和 JEIDA mode 的定义如图 2-1。

JEDIA mode



NS mode

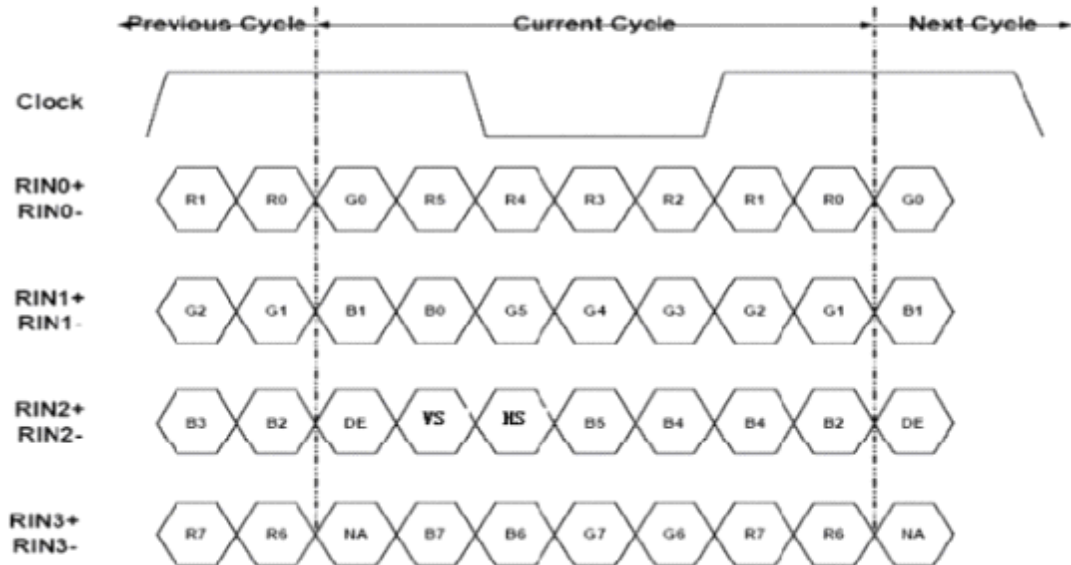


图 2-1 LVDS JEDIA mode and NS mode

4.2. LCD 时序参数说明

4.2.1. lcd_x

显示屏的水平像素点

4.2.2. lcd_y

显示屏的垂直像素点

4.2.3. lcd_ht

Horizontal Total time

指一行总的 dclk 的 cycle 个数。见图 2-2。



图 2-2 水平方向时序信号图

4.2.4. lcd_hbp

Horizontal Back Porch

指有效行间，行同步信号（hsync）开始，到有效数据开始之间的 dclk 的 cycle 个数，包括同步信号区。见图 2-2。

4.2.5. lcd_hspw

Horizontal Sync Pulse Width

指行同步信号的宽度。单位为 1 个 dclk 的时间（即是 1 个 data cycle 的时间）。见图 2-2。

4.2.6. lcd_vt

Vertical Total time

指一场的总行数。见图 2-3。

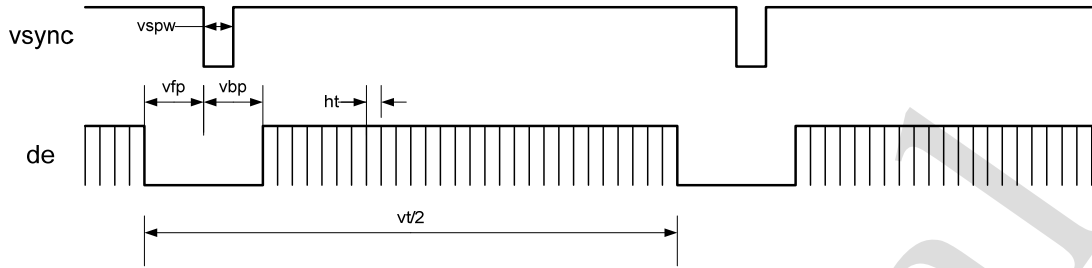


图 2-3 垂直方向时序信号图

4.2.7. lcd_vbp

Vertical Back Porch

指场同步信号（vsync）开始，到有效数据行开始之间的行数，包括场同步信号区。见图 2-3。

4.2.8. lcd_vspw

Vertical Sync Pulse Width

指场同步信号的宽度。单位为行。见图 2-3。

4.2.9. lcd_dclk_freq

Data Clock Frequency

指 PIN 总线上数据的传送频率。单位为 MHz

屏幕刷新帧数 = $(\text{lcd_dclk_freq} \times 1000 \times 1000) / (\text{ht} \times \text{vt})$

4.3. LCD 其他参数说明

4.3.1. lcd_pwm_freq

Lcd backlight PWM Frequency

这个参数配置 PWM 信号的频率，单位为 Hz。A20 PWM 的配置范围为 100Hz-100,000Hz。

4.3.2. lcd_pwm_pol

Lcd backlight PWM Polarity

这个参数配置 PWM 信号的占空比的极性。设置相应值对应含义为：

0: active high

1: active low

4.3.3. lcd_frm

Lcd Frame Rate Modulator

FRM 是解决由于 PIN 减少导致的色深问题。

这个参数设置相应值对应含义为：

0: RGB888 → RGB888 direct

1: RGB888 → RGB666 dither

2: RGB888 → RGB565 dither

有些 LCD 屏的像素格式是 18bit 色深 (RGB666) 或 16bit 色深 (RGB565)，建议打开 FRM 功能，通过 dither 的方式弥补色深，使显示达到 24bit 色深 (RGB888) 的效果。如图 2-4 和图 2-5，上图是色深为 RGB66 的 LCD 屏显示，下图是打开 dither 后的显示，打开 dither 后色彩渐变的地方过度平滑。



图 2-4 FRM 效果图



图 2-5 FRM 效果图

4.3.4. lcd_gamma_correction_en

Lcd Gamma Correction Enable

设置相应值的对应含义为：

0：Lcd 的 Gamma 校正功能关闭

1：Lcd 的 Gamma 校正功能开启

设置为 1 时，需要对 lcd_gamma_tbl [256]进行赋值。

4.3.5. lcd_gamma_tbl

Lcd Gamma Table（只能在屏文件中配置）

该参数为一个数组 `_u32 lcd_gamma_tbl[256]`;

`lcd_gamma_tbl[n] = rout<<16 | gout<<8 | bout<<0` 表示：输入 $r=n$ 时，输出 $r=rout$ ；输入 $g=n$ 时，输出 $g=gout$ ；输入 $b=n$ 时，输出 $b=bout$ 。

用户使用 Gamma 校正功能时，可以使用 `lcd0_panel_cfg.c` 中的 `LCD_cfg_panel_info` 函数中提供的模板对 `tbl` 进行赋值。

4.4. POWER 及 IO 说明

4.4.1. lcd_power

示例：`lcd_power = port:power2<1><0><default><1>`

含义：LCD 的供电定义 gpio 控制。

4.4.2. lcd_bl_en

示例：`lcd_bl_en = port:PH07<1><0><default><1>`

含义：`lcd_power` 引脚为 PH07，PH07 输出高电平时打开 LCD 背光；上下拉不使能。

第一个尖括号：功能分配；1 为输出；

第二个尖括号：内置电阻；使用 0 的话，标示内部电阻高阻态，如果是 1 则是内部电阻上拉，2 就代表内部电阻下拉。使用 `default` 的话代表默认状态，即电阻上拉。其它数据无效。

第三个尖括号：驱动能力；`default` 表驱动能力是等级 1

第四个尖括号：输出有效所需电平；LCD 背光工作时的电平，0 为低电平，1 为高电平。

4.4.3. lcd_pwm

示例：`lcd_pwm = port:PB02<2><0><default><default>`

含义：PB02 输出 PWM 信号。

A20 方案固定 PB02 为 PWM 信号输出引脚。建议使用此默认配置。

4.4.4. lcd_gpio_x

示例：`lcd_gpio_0 = port:PA06<0><0><default><default>`

含义：`lcd_gpio_0` 引脚为 PA06。

第一个尖括号：功能分配；0 为输入，1 为输出；

第二个尖括号：内置电阻；使用 0 的话，标示内部电阻高阻态，如果是 1 则是内部

电阻上拉，2 就代表内部电阻下拉。使用 default 的话代表默认状态，即电阻上拉。其它数据无效。

第三个尖括号：驱动能力；default 表驱动能力是等级 1

第四个尖括号：表示默认值；即是当设置为输出时，该引脚输出的电平，0 为低电平，1 为高电平。

A20 配置中，共有 6 个可选的 lcd_gpio 引脚，lcd_gpio_0, lcd_gpio_1, lcd_gpio_2, lcd_gpio_3, lcd_gpio_4, lcd_gpio_5。

其中，lcd_gpio_0, lcd_gpio_1, lcd_gpio_2, lcd_gpio_3, 对应于“2.3.3 使用 IO 模拟串行接口初始化”的 IO 位置。

lcd_gpio_0 和 lcd_gpio_1, 对应于“使用 SPI 接口初始化”中 SPI 的 IO 位置。使用 SPI 接口。

4.4.5. lcd_pin

示例：lccd0 = port:PD00<3><0><default><default>

含义：lccd0 这个引脚，即是 PD0，配置为 LVDS 输出。

第一个尖括号：功能分配；0 为输入，1 为输出，2 为 LCD 输出，3 为 LVDS 接口输出，7 为 disable。

第二个尖括号：内置电阻；使用 0 的话，标示内部电阻高阻态，如果是 1 则是内部电阻上拉，2 就代表内部电阻下拉。使用 default 的话代表默认状态，即电阻上拉。其它数据无效。

第三个尖括号：驱动能力；default 表驱动能力是等级 1

第四个尖括号：表示默认值；即是当设置为输出时，该引脚输出的电平，0 为低电平，1 为高电平。

LCD PIN 的配置如下：

LCD 为 HV RGB 或 DSI 屏，CPU/I80 屏时，必须定义相应的 IO 口为 LCD 输出（如果是 0 路输出，第一个尖括号为 2；如果是 1 路输出，第一个尖括号为 3）；

LCD 为 LVDS 屏时，必须定义 PD 口对应的 IO 口为 LVDS 输出（即第一个尖括号为 3）；

LCD PIN 的所有 IO，均可通过注释方式去掉其定义，显示驱动对注释 IO 不进行初始化操作。

5. LCD 软件配置说明

5.1. 屏文件说明

对 lcd0_panel_cfg.c 和 lcd1_panel_cfg.c 中提供的函数接口如图 3-1 所示。其中，LCD_cfg_panel_info，LCD_open_flow，LCD_close_flow 和 LCD_get_panel_funs_0/LCD_get_panel_funs_1 是必须包含的 4 个函数。

函数：LCD_cfg_panel_info

功能：配置 A20 的 TCON 扩展参数

原型：

```
static void LCD_cfg_panel_info(__panel_extend_para_t * info)
```

TCON 的扩展参数只能在屏文件中配置，参数的定义见“2.3 LCD 其他参数说明”。

需要 gamma 校正，或色彩映射，或亮度曲线调整（调整亮度的曲线变化，以达到让人眼感受的亮度更线性化）的 LCD 屏，在 sys_config 中将相应模块的 enable 参数置 1，lcd_gamma_en, lcd_cmap_en, lcd_bright_curve_en；并且填充 3 个系数表，lcd_gamma_tbl, lcd_bright_curve_tbl, lcd_cmap_tbl，如下所示红色代码部分。注意的是：gamma 及 bright_cure 表，模板提供了 18 段拐点值，然后再插值出所有的值（255 个）。如果觉得还不细，可以往相应表格里添加子项。cmap_tbl 的大小是固定了，不能减小或增加表的大小。

最终生成的 gamma 表项是由 rgb 三个 gamma 值组成的，各占 8bit，目前提供的模板中，三个 gamma 值是相同的。

```
static void LCD_cfg_panel_info(__panel_extend_para_t * info)
```

```
{  
    __u32 i = 0, j=0;  
    __u8 lcd_gamma_tbl[][2] =  
    {  
        // {input value, corrected value}  
        {0, 0},  
        {15, 15},  
        {30, 30},  
        {45, 45},  
        {60, 60},  
        {75, 75},  
        {90, 90},  
        {105, 105},  
        {120, 120},  
        {135, 135},  
        {150, 150},  
        {165, 165},  
        {180, 180},  
    }  
}
```

```
{195, 195},  
{210, 210},  
{225, 225},  
{240, 240},  
{255, 255},  
};
```

```
memset(info,0,sizeof(__panel_extend_para_t));
```

```
...  
}
```

函数: LCD_open_flow

功能: 定义开屏的流程

原型: static __s32 LCD_open_flow(__u32 sel)

具体说明见“3.2 开关屏流程”。

函数: LCD_close_flow

功能: 定义关屏的流程

原型: static __s32 LCD_close_flow(__u32 sel)

该函数与 LCD_open_flow 对应

函数: LCD_get_panel_funs_0/ LCD_get_panel_funs_1

功能:

原型: void LCD_get_panel_funs_0(__lcd_panel_fun_t * fun)/

void LCD_get_panel_funs_1(__lcd_panel_fun_t * fun)

该函数无需用户修改, LCD_get_panel_funs_0 只在文件 lcd0_panel_cfg.c 中定义,
LCD_get_panel_funs_1 只在文件 lcd1_panel_cfg.c 中定义。

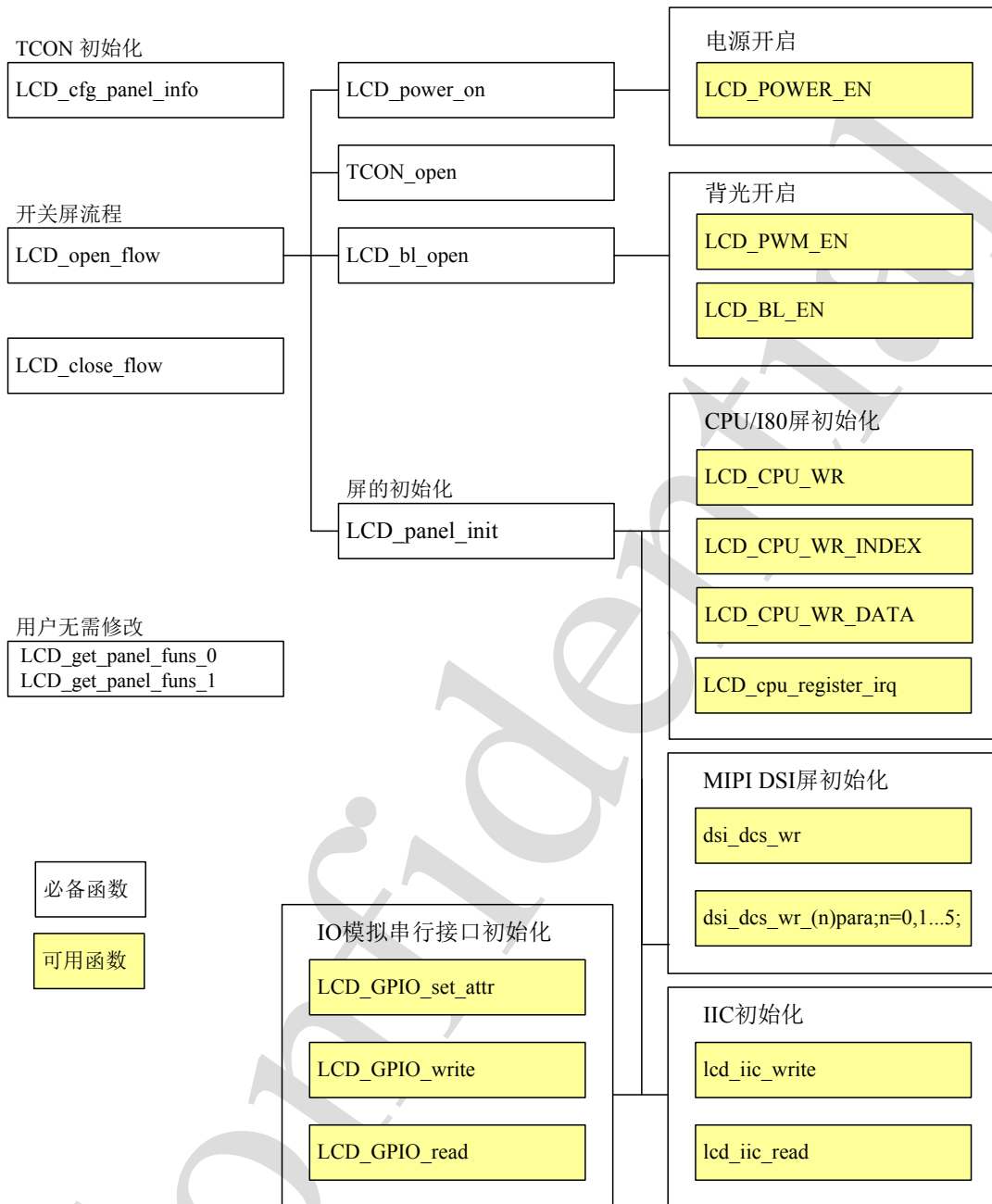


图 3-1 配屏文件中的函数列表

5.2. 开关屏流程

开关屏的操作流程如图 3-2 所示。

其中，LCD_open_flow 和 LCD_close_flow 称为开关屏流程函数，方框中的函数，如 LCD_power_on，TCON_open 等函数，称为开关屏步骤函数。

不需要进行初始化操作的 LCD 屏，LCD_panel_init 及 LCD_panel_exit 这函数可以为空。

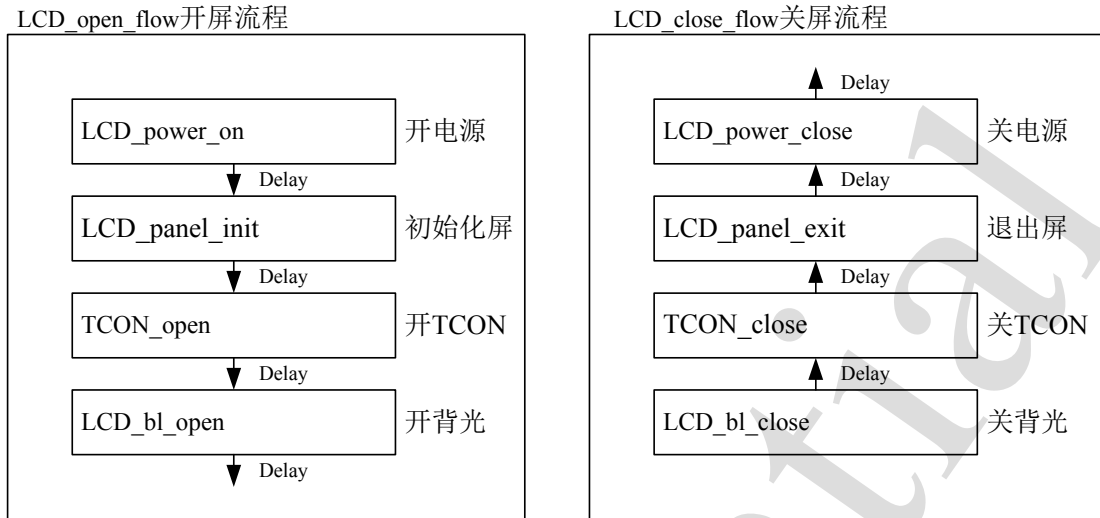


图 3-2 开关屏流程

5.2.1. 开关屏步骤函数说明

开屏的步骤函数有 LCD_panel_init, TCON_open, LCD_power_on, LCD_bl_open。

函数: LCD_panel_init

功能: 对屏初始化

原型: static void LCD_panel_init(__u32 sel)

可参考“2.3 对屏的初始化”。部分屏不需要进行初始化操作, LCD_panel_init 及 LCD_panel_exit 这两个步骤函数可以省去。

函数: TCON_open

功能: 打开 A20 LCD 控制器, 开始刷新 LCD 显示

原型: __s32 TCON0_open(__u32 sel)

该函数由显示驱动提供, 用户无需实现。

函数: LCD_power_on

功能: 打开 LCD 电源

原型: static void LCD_power_on(__u32 sel)

显示驱动提供 LCD_PWR_EN 函数可供调用, 用户也可自由实现函数内容。

函数: LCD_bl_open

功能: 打开 LCD 背光

原型: static void LCD_bl_open(__u32 sel)

显示驱动提供 LCD_PWM_EN 和 LCD_BL_EN 函数可供调用, 用户也可自由实现函数内容。

LCD_PWM_EN, LCD_BL_EN, LCD_PWR_EN 这三个函数是通过 GPIO 控制实现电源和背光的开启关闭, IO 的位置及属性定义在 sys_config.fex 文件中。

函数: LCD_PWM_EN

功能: 打开或关闭 LCD 背光调节的 PWM 信号

原型: void LCD_PWM_EN(__u32 sel, __bool b_en)

参数说明:

b_en=0: 将 PWM pin 设为输入, 并把 PWM 模块关闭

b_en=1: 将 PWM pin 设为 PWM, 并把 PWM 模块打开
对应于 sys_config.fex 文件的 lcd_pwm。

函数: LCD_BL_EN

功能: 打开或关闭 LCD 背光

原型: void LCD_BL_EN (__u32 sel, __bool b_en)

参数说明:

b_en=0: 设置 LCD 背光控制 IO 为对应电平, 关闭背光

b_en=1: 设置 LCD 背光控制 IO 为对应电平, 打开背光
对应于 sys_config.fex 文件的 lcd_bl_en;

函数: LCD_PWR_EN

功能: 打开或关闭 LCD 电源

原型: void LCD_PWR_EN(__u32 sel, __bool b_en)

参数说明:

b_en=0: 设置 LCD 电源控制 IO 为对应电平, 关闭 LCD 电源

b_en=1: 设置 LCD 电源控制 IO 为对应电平, 打开 LCD 电源
对应于 sys_config.fex 文件的 lcd_power。

关屏的步骤函数与开屏的步骤函数相对应。

5.2.2. 开关屏流程函数说明

函数: LCD_open_flow

功能: 初始化开关屏的步骤流程

原型: static __s32 LCD_open_flow(__u32 sel)

函数常用内容为:

```
static __s32 LCD_open_flow(__u32 sel)
{
    LCD_OPEN_FUNC(sel, LCD_power_on, 10);
    LCD_OPEN_FUNC(sel, LCD_panel_init, 50);
    LCD_OPEN_FUNC(sel, TCON_open, 100);
    LCD_OPEN_FUNC(sel, LCD_bl_open, 0);
    return 0;
}
```

如上, 初始化整个开屏的流程步骤为四个:

- 打开 LCD 电源, 再延迟 10ms;
- 初始化屏, 再延迟 50ms; (不需要初始化的屏, 可省掉此步骤)
- 打开 A20 TCON, 再延迟 200ms;
- 打开背光, 再延迟 0ms。

LCD_open_flow 函数只会系统初始化的时候调用一次, 执行每个 LCD_OPEN_FUNC

即是把对应的开屏步骤函数进行注册，并没有执行该开屏步骤函数。LCD_open_flow 函数的内容必须统一用 LCD_OPEN_FUNC(sel, function, delay_time)进行函数注册的形式，确保正常注册到开屏步骤中。

函数：LCD_OPEN_FUNC

功能：注册开屏步骤函数到开屏流程中

原型：void LCD_OPEN_FUNC(__u32 sel, LCD_FUNC func, __u32 delay)

参数说明：

func 是一个函数指针，其类型是：void (*LCD_FUNC) (__u32 sel)，用户自己定义的函数必须也要用统一的形式。比如：

```
void user_defined_func(__u32 sel)
{
    //do something
}
```

delay 是执行该步骤后，再延迟的时间，时间单位是毫秒。

5.3. 屏的初始化

一部分屏需要进行初始化操作，在开屏步骤函数中，对应于 LCD_panel_init 函数。在 A20 中，提供了几种方式对屏的初始化，可参加图 2-1。

对于 DSI 屏，是通过 DSI-D0 通道进行初始化。对于 CPU 屏，是通过 8080 总线的方式，使用的是 LCDIO (PD,PH) 进行初始化。这种初始化方式，其总线的引脚位置定义与 CPU 屏一致。

对于需要 SPI 或其他串行协议的方式，可使用 GPIO 引脚模拟实现。对于需要 IIC 初始化的 LCD 屏，提供 IIC 的读写函数。模拟 GPIO，IIC 的 SDA，SCL 和的引脚位置定义见于 sys_config.fex 中。

若使用 A20+SSD2828 的组合，可使用默认的 LCD_panel_init。

5.3.1. CPU/I80 屏的初始化

CPU 屏的初始化可以参考“附录 5.3.5”的实例。

显示驱动提供 5 个接口函数可供使用。如下：

函数：LCD_CPU_WR

功能：设定 CPU 屏的指定寄存器为指定的值

原型：void LCD_CPU_WR(__u32 sel, __u32 index, __u32 data)

函数内容为

```
void LCD_CPU_WR(__u32 sel, __u32 index, __u32 data)
{
    LCD_CPU_WR_INDEX(sel, index);
    LCD_CPU_WR_DATA(sel, data);
}
```

实现了 8080 总线上的两个写操作。

LCD_CPU_WR_INDEX 实现第一个写操作，这时 PIN 脚 RS (A1) 为低电平，总线

数据上的数据内容为参数 `index` 的值。

`LCD_CPU_WR_DATA` 实现第二个写操作，这时 PIN 脚 RS (A1) 为高电平，总线数据上的数据内容为参数 `data` 的值。

函数: `LCD_CPU_WR_INDEX`

功能: 设定 CPU 屏为指定寄存器

原型: `void LCD_CPU_WR(__u32 sel, __u32 index, __u32 data)`

`void LCD_CPU_WR_INDEX(__u32 sel, __u32 index);`

具体说明见 `LCD_CPU_WR`。

函数: `LCD_CPU_WR_DATA`

功能: 设定 CPU 屏寄存器的值为指定的值

原型: `void LCD_CPU_WR_DATA(__u32 sel, __u32 data);`

具体说明见 `LCD_CPU_WR`。

函数: `LCD_cpu_register_irq`

功能: 设置 LCD_GPIO PIN 脚为输入或输出模式

原型: `void LCD_CPU_register_irq(__u32 sel, void (*Lcd_cpuiqr_proc) (void))`

注册 `cpu` 屏的中断处理函数，驱动会在每个 `vblanking` 中断里调用一下用户注册的中断处理函数 `Lcd_cpuiqr_proc`。

CPU 屏的初始化对应于开屏步骤函数的 `LCD_panel_init`。在 CPU 屏 `LCD_panel_init` 函数的最后，需要进行一个操作步骤，即是：

使用 `LCD_CPU_register_irq` 注册 CPU 屏的中断处理函数 `Lcd_cpuiqr_proc`，该函数的内容，可以是 CPU 屏 GRAM 的 X 和 Y 坐标设置为零的操作，以保证异步屏每帧进行一次同步。

示例如下：

```
static void LCD_panel_init(__u32 sel)
{
    kgm281i0_init(sel); //initial lcd panel
    LCD_CPU_register_irq(sel,Lcd_cpuiqr_proc); //register cpu irq func
}
```

5.3.2. 使用 IO 模拟串行接口初始化

IO 模拟串行接口初始化可以参考附录 5.3.2 中的实例。

IO 的位置 (PIN 脚) 定义，默认属性 (输入输出) 定义及默认输出值在 `sys_config.fex`，具体请参考 2.5.2。

显示驱动提供 2 个接口函数可供使用。说明如下：

函数: `LCD_GPIO_read`

功能: 读取 LCD_GPIO PIN 脚上的电平

原型: `__s32 LCD_GPIO_read(__u32 sel, __u32 io_index);`

参数说明：

`io_index = 0`: 对应于 `sys_config.fex` 中的 `lcd_gpio_0`

`io_index = 1`: 对应于 `sys_config.fex` 中的 `lcd_gpio_1`

io_index = 2: 对应于 sys_config.fex 中的 lcd_gpio_2

io_index = 3: 对应于 sys_config.fex 中的 lcd_gpio_3

函数返回值为对应 IO 的输入电平，只用于该 GPIO 定义为输入的情形。

函数: LCD_GPIO_write

功能: LCD_GPIO PIN 脚上输出高电平或低电平

原型: __s32 LCD_GPIO_write(__u32 sel, __u32 io_index, __u32 data);

参数说明:

io_index = 0: 对应于 sys_config.fex 中的 lcd_gpio_0

io_index = 1: 对应于 sys_config.fex 中的 lcd_gpio_1

io_index = 2: 对应于 sys_config.fex 中的 lcd_gpio_2

io_index = 3: 对应于 sys_config.fex 中的 lcd_gpio_3

data = 0: 对应 IO 输出低电平

data = 1: 对应 IO 输出高电平

只用于该 GPIO 定义为输出的情形。

函数: LCD_GPIO_set_attr

功能: 设置 LCD_GPIO PIN 脚为输入或输出模式

原型: __s32 LCD_GPIO_set_attr(__u32 sel, __u32 io_index, __bool b_output);

参数说明:

io_index = 0: 对应于 sys_config.fex 中的 lcd_gpio_0

io_index = 1: 对应于 sys_config.fex 中的 lcd_gpio_1

io_index = 2: 对应于 sys_config.fex 中的 lcd_gpio_2

io_index = 3: 对应于 sys_config.fex 中的 lcd_gpio_3

b_output = 0: 对应 IO 设置为输入

b_output = 1: 对应 IO 设置为输出

5.3.3. 使用 SPI 接口初始化

使用 SPI 接口初始化，实质还是使用 GPIO 模拟实现，系统内部已实现了 GPIO 模拟 SPI 的操作。

IO 的位置 (PIN 脚) 定义，默认属性 (输入输出) 定义及默认输出值在 sys_config.fex，具体请参考 2.4.4。

系统提供 2 个接口函数用于 SPI 读写操作。说明如下:

函数: spi_24bit_3wire

功能: 指定 SPI 从设备，进行写寄存器操作

原型: void spi_24bit_3wire(__u32 tx);

spi_24bit_3wire(0x7000B1); //VSA=50, HAS=64 地址

spi_24bit_3wire(0x723240); //数据

5.3.4. DSI 屏的初始化

A20 支持的 DSI 方案有 A20+SSD2828。对 DSI 屏的初始化是通过 SSD2828 实现的，显示驱动实现了与 SSD2828 的通讯，用户可使用默认的 lcd_panel_init 函数即可完成屏

的初始化操作。

A20 与 SSD2828 的通讯使用的是 SPI 接口,所以,使用 DSI 屏,必须在 sys_config.fex 中定义 SPI 接口的 IO。

5.4. 其它函数

5.4.1. 用户自定义函数

函数: LCD_user_defined_func

功能: 用户可自由定义的函数

原型: static __s32 LCD_user_defined_func(__u32 sel, __u32 para1, __u32 para2, __u32 para3)

该函数是给用户作扩展使用的,可以在此函数里实现任何你想实现的代码,然后在应用层进行调用。

比如对于 3D 屏,可以在该文件里实现屏的 2D 和 3D 的切换,示例如下:

```
static __s32 LCD_user_defined_func(__u32 sel, __u32 para1, __u32 para2, __u32 para3)
{
    if(para1 == 0)
    {
        //switch to 2D mode
    }
    else
    {
        //switch to 3D mode
    }
    return 0;
}
```

在用户空间的调用代码示例如下 (切换到 3D 模式):

```
unsigned long arg[4];
int dispfh;

if((dispfh = open("/dev/disp",O_RDWR)) == -1)
{
    printf("open file /dev/disp fail. \n");
    return 0;
}
arg[0] = 0;//lcd0
arg[1] = 1;//switch to 3D mode
ioctl(dispfh, DISP_CMD_LCD_USER_DEFINED_FUNC,(unsigned long)arg);
```

5.4.2. 延时函数

驱动提供了毫秒和微秒级的延时给用户使用, 不过建议如果延时时间比较长的话可以在开关屏流程里新添新的初始化步骤. 因为在 boot 系统里延时是死等的,效率会比较低。

函数: LCD_delay_ms

功能: 延时 ms 毫秒

原型: void LCD_delay_ms(__u32 ms)

函数: LCD_delay_us

功能: 延时 us 微秒

原型: void LCD_delay_us(__u32 us)

6. 模块开发 demo

Lcd 配置可参考发布出去的源代码（屏文件）及配置文件（sys_concfig.fex），也可参考附录

Confidential

7. 模块调试

后续添加

Confidential

8. 附录

- 4.1-4.8 为 A20 常见 LCD 屏的配置举例。
- 4.9 为电源和背光的配置说明。
- 4.10 为 LCD 的 IO MAP 表
- 4.11 为 LCD Check List

8.1. A20 与 HV Parallel RGB 屏

8.1.1. schematic

HV Parallel RGB 屏有 24/18 个 data io, 有 dclk、hsync、vsync、de 四个 ctrl io。如图 4-1 是一个典型 HV Parallel RGB 屏的模组规格书的引脚定义。A20 与该 LCD 屏的引脚连接可参考图 4-2。

| Pin No. | Symbol | I/O | Function |
|---------|-------------------|-----|--|
| 1 | V _{LED-} | P | Power for LED backlight cathode |
| 2 | V _{LED+} | P | Power for LED backlight anode |
| 3 | GND | P | Power ground |
| 4 | V _{DD} | P | Power voltage |
| 5 | R0 | I | Red data (LSB) |
| 6 | R1 | I | Red data |
| 7 | R2 | I | Red data |
| 8 | R3 | I | Red data |
| 9 | R4 | I | Red data |
| 10 | R5 | I | Red data |
| 11 | R6 | I | Red data |
| 12 | R7 | I | Red data (MSB) |
| 13 | G0 | I | Green data (LSB) |
| 14 | G1 | I | Green data |
| 15 | G2 | I | Green data |
| 16 | G3 | I | Green data |
| 17 | G4 | I | Green data |
| 18 | G5 | I | Green data |
| 19 | G6 | I | Green data |
| 20 | G7 | I | Green data (MSB) |
| 21 | B0 | I | Blue data (LSB) |
| 22 | B1 | I | Blue data |
| 23 | B2 | I | Blue data |
| 24 | B3 | I | Blue data |
| 25 | B4 | I | Blue data |
| 26 | B5 | I | Blue data |
| 27 | B6 | I | Blue data |
| 28 | B7 | I | Blue data (MSB) |
| 29 | DGND | I | Digital ground |
| 30 | DCLK | I | Pixel clock |
| 31 | DISP | I | Display on/ off |
| 32 | HSYNC | I | Horizontal sync signal |
| 33 | VSYNC | I | Vertical sync signal |
| 34 | DE | I | Data enable |
| 35 | NC | - | No Connect |
| 36 | GND | P | Power ground |
| 37 | X1 | I/O | Right electrode - differential analog |
| 38 | Y1 | I/O | Bottom electrode - differential analog |
| 39 | X2 | I/O | Left electrode - differential analog |
| 40 | Y2 | I/O | Top electrode - differential analog |

并行数据引脚

同步控制引脚

图 4-1 HV Parallel RGB 屏的引脚定义

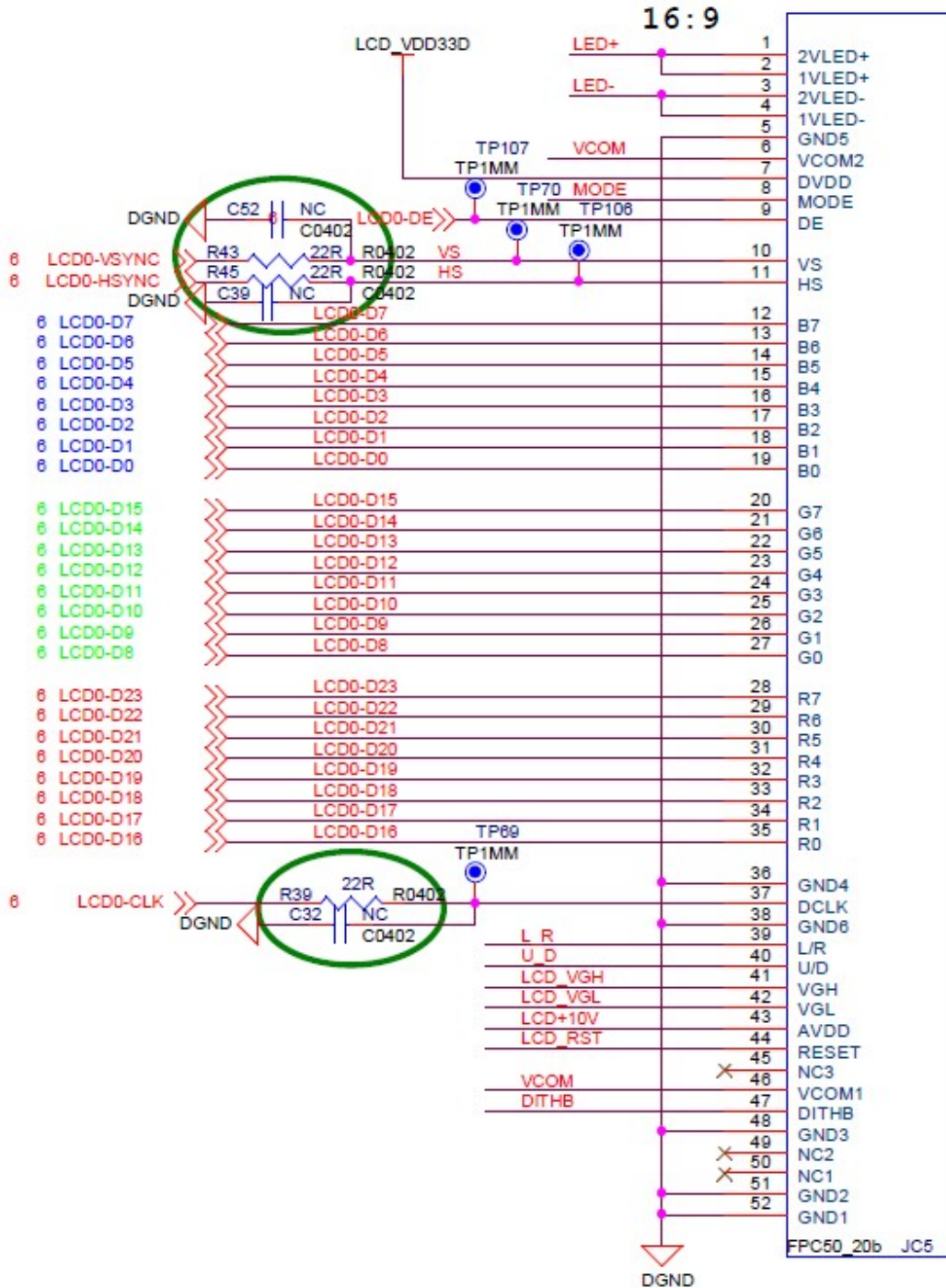


图 4-2 A20 与 HV Parallel RGB 屏连接图

8.1.2. sys_config.fex

该 LCD 屏的参数如图 4-3。从中，确定 A20 的 LCD 时序参数。如下：

- lcd_dclk_freq = 33;
- lcd_x = Horizontal display area = 800;
- lcd_y = Vertical display area = 480;

$lcd_ht = HSYNC\ period\ area = 928;$
 $lcd_hbp = HSYNC\ back\ porch + HSYNC\ width = 40 + 48 = 88;$
 $lcd_hspw = HSYNC\ width = 48;$
 $lcd_vt = VSYNC\ period\ area = 525 \times 2;$
 $lcd_vbp = VSYNC\ back\ porch + VSYNC\ width = 29 + 3 = 32;$
 $lcd_vspw = VSYNC\ width = 3;$

该 LCD 屏对应的 sys_config.fex 的配置如下。因该 LCD 屏无需屏的初始化操作，使用默认的 lcd0_panel_cfg.c 文件。

| Item | Symbol | Min. | Typ. | Max. | Unit | Note |
|-------------------------|--------|------|------|------|------|------|
| DCLK cycle time | Tclk | 25 | | | ns | |
| DCLK frequency | fclk | | 33 | 40 | MHz | |
| Horizontal display area | Thd | | 800 | | Tcph | |
| HSYNC period time | Th | | 928 | | Tcph | |
| HSYNC width | Thwh | 1 | 48 | | Tcph | |
| HSYNC back porch | Thbp | | 40 | | Tcph | |
| HSYNC front porch | Thfp | | 40 | | Tcph | |
| Vertical display area | Tvd | | 480 | | th | |
| VSYNC period time | Tv | | 525 | | th | |
| VSYNC width | Tvwh | | 3 | | th | |
| VSYNC back porch | Tvbp | | 29 | | th | |
| VSYNC front porch | Tvfp | | 13 | | th | |

图 4-3

HV Parallel RGB 屏参数

```

lcd_if           = 0
lcd_hv_if       = 0
lcd_hv_clk_phase = 0
lcd_hv_sync_polarity = 0

lcd_dclk_freq   = 33
lcd_x           = 800
lcd_y           = 480
lcd_ht         = 928
llcd_hbp       = 88
lcd_hspw       = 48
lcd_vt         = 1050
    
```


lcd_vbp = 32
lcd_vspw = 3

lcdd0 = port:PD00<2><0><3><default>
lcdd1 = port:PD01<2><0><3><default>
lcdd2 = port:PD02<2><0><3><default>
lcdd3 = port:PD03<2><0><3><default>
lcdd4 = port:PD04<2><0><3><default>
lcdd5 = port:PD05<2><0><3><default>
lcdd6 = port:PD06<2><0><3><default>
lcdd7 = port:PD07<2><0><3><default>
lcdd8 = port:PD08<2><0><3><default>
lcdd9 = port:PD09<2><0><3><default>
lcdd10 = port:PD10<2><0><3><default>
lcdd11 = port:PD11<2><0><3><default>
lcdd12 = port:PD12<2><0><3><default>
lcdd13 = port:PD13<2><0><3><default>
lcdd14 = port:PD14<2><0><3><default>
lcdd15 = port:PD15<2><0><3><default>
lcdd16 = port:PD16<2><0><3><default>
lcdd17 = port:PD17<2><0><3><default>
lcdd18 = port:PD18<2><0><3><default>
lcdd19 = port:PD19<2><0><3><default>
lcdd20 = port:PD20<2><0><3><default>
lcdd21 = port:PD21<2><0><3><default>
lcdd22 = port:PD22<2><0><3><default>
lcdd23 = port:PD23<2><0><3><default>
lcdclk = port:PD24<2><0><3><default>
lcdde = port:PD25<2><0><3><default>
lcdhsync = port:PD26<2><0><3><default>
lcdvsync = port:PD27<2><0><3><default>

8.2. A20 与 HV Serial RGB 屏

8.2.1. schematic

HV Serial RGB 屏有 8 个 data io, 有 dclk、hsync、vsync、de 四个 ctrl io。常见的 HV Serial RGB 屏可同时支持 RAW RGB、Dummy RGB、Serial YUV/CCIR, 其引脚定义与 HV Serial RGB 屏一致。对这一类屏的配置都可参考该例子。

如图 4-4 是一个典型 HV Parallel RGB 屏的模组规格书的引脚定义。A20 与该 LCD 屏的引脚连接可参考图 4-5。

| PIN NO. | Symbol | Description |
|---------|--------|---|
| 1~8 | D0~D7 | — 串行数据引脚 |
| 9 | DCLK | Data clock input |
| 10 | VSYNC | — 同步控制引脚 |
| 11 | HSYNC | Horizontal sync input |
| 12 | SCL | Serial command clock input |
| 13 | SDA | — 初始化接口引脚 |
| 14 | CSB | Serial communication chip select |
| 15 | VDDIO | Input I/O power supply |
| 16 | AGND | Analog ground for source driver |
| 17 | NC | No Connection |
| 18 | FB | Main boost regulator feedback input |
| 19 | NC | No Connection |
| 20 | VLED | Supply voltage for LED backlight |
| 21 | DRV | Gate signal for the power transistor of the boost converter |
| 22 | VDD | Charge Pump power supply |
| 23 | GND | Digital ground. |
| 24 | C1N | Pins to connect capacitors for power circuitry |
| 25 | C1P | Pins to connect capacitors for power circuitry |
| 26 | NC | NO connection |
| 27 | C2N | Pins to connect capacitors for power circuitry |
| 28 | C2P | Pins to connect capacitors for power circuitry |
| 29 | VDD3 | Charge Pump circuit reference voltage |
| 30 | C3N | Pins to connect capacitors for power circuitry |
| 31 | C3P | Pins to connect capacitors for power circuitry |

图 4-4 HV Serial RGB 屏的引脚定义

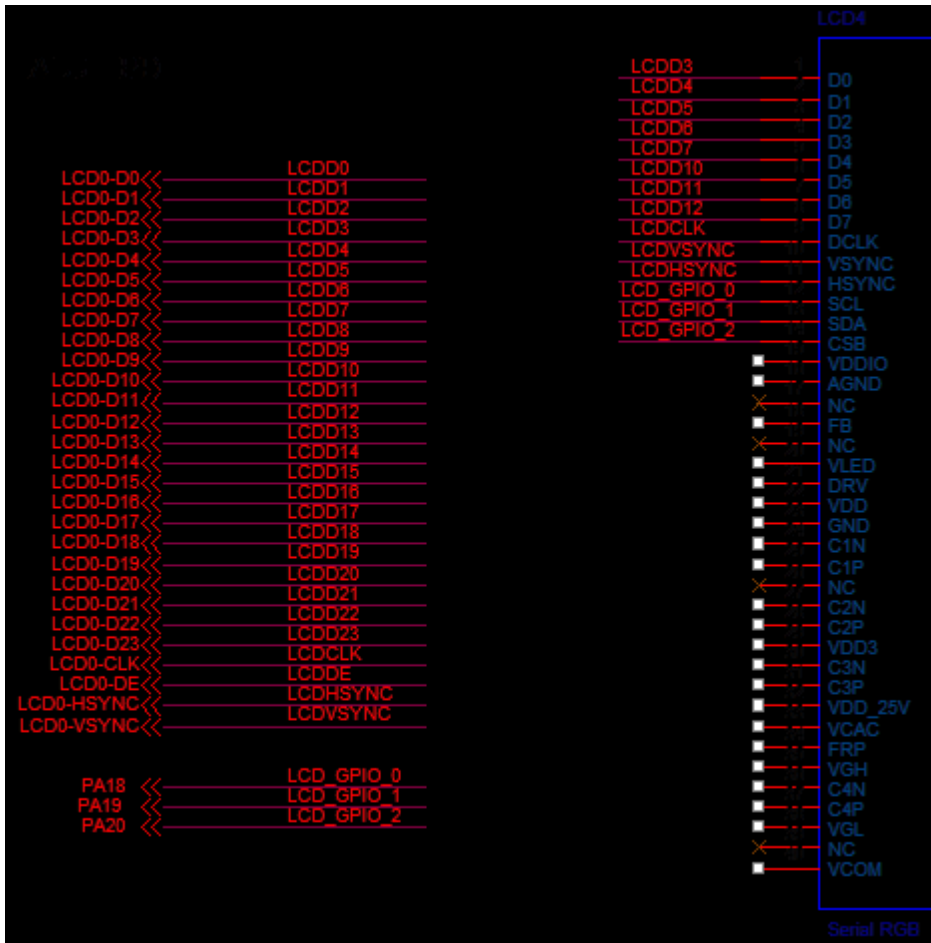


图 4-5 A20 与 HV Parallel RGB 屏连接图

8.2.2. sys_config.fex

该 LCD 屏参数如图 4-6 所示。采用 Non-Interlace 的配置，参数配置可参考 HV parallel RGB 例子。区别于 HV parallel RGB，该 LCD 屏 3 dclk cycle 为一个像素。

```
lcd_hv_if = 8;
```

```
lcd_x = 480 / 3 = 160;
```

```
lcd_ht > lcd_x * cycle + lcd_hbp = 160 * 3 + 100 = 580, 取 lcd_ht = 617。
```

该 LCD 屏需要初始化接口，配置相应的三个 lcd_gpio_x，如下，定义这 3 个 PIN 的默认状态为输出高电平。

该 LCD 屏只需 lcdd3, lcdd4, lcdd5, lcdd6, lcdd7, lcdd10, lcdd11, lcdd12, lcddclk, lcdhsync, lcdvsync, PD 口的其他 io 可以不加定义。

sys_config.fex 文件中配置如下。

| Parameter | Symbol | Interface | | | (*)Non-Interface | | | Unit |
|---------------------------------|-----------------------|-----------|-------|-------|------------------|------|------|------|
| | | Min. | Typ. | Max. | Min. | Typ. | Max. | |
| Vertical display area | t_{vd} | 240 | | | 240 | | | H |
| VSYNC period time | t_v | 247.5 | 262.5 | 277.5 | 247 | 262 | 277 | H |
| VSYNC pulse width | t_{vpw} | 1 DCLK | 1H | 6H | 1 DCLK | 1H | 6H | |
| (*)VSYNC Blanking (t_{vb}) | Odd field t_{vbo} | 6 | 13 | 21 | 6 | 13 | 21 | H |
| | Even field t_{vbe} | 6.5 | 13.5 | 21.5 | | | | |
| VSYNC Front porch (t_{vfp}) | Odd field t_{vfpo} | 1.5 | 9.5 | 16.5 | 1 | 9 | 16 | H |
| | Even field t_{vfpe} | 1 | 9 | 16 | | | | |

| Parameter | Symbol | Value | | | Unit |
|-------------------------|-----------|-------|------|------|------|
| Horizontal display area | t_{hd} | 480 | | | DCLK |
| DCLK frequency | f_{clk} | Min. | Typ. | Max | |
| | | 8.1 | 9.7 | 11.3 | Mhz |
| 1 Horizontal Line | t_h | 617 | | | DCLK |
| HSYNC pulse width | t_{hpw} | Min. | 1 | | |
| | | Typ. | 1 | | |
| | | Max. | 96 | | |
| HSYNC blanking | t_{hb} | 84 | 100 | 115 | |
| HSYNC front porch | t_{hfp} | 53 | 37 | 22 | |

图 4-6 HV Serial RGB 屏参数

```

lcd_if           = 0
lcd_hv_if       = 8
lcd_hv_clk_phase = 0
lcd_hv_sync_polarity = 0

lcd_dclk_freq   = 10
lcd_x           = 160
lcd_y           = 240
lcd_ht          = 1234
l1cd_hbp       = 100
lcd_hspw        = 20
lcd_vt          = 262
lcd_vbp         = 13
lcd_vspw        = 3

lcd_gpio_0      = port:PA18<1><0><default><1>
lcd_gpio_1      = port:PA19<1><0><default><1>
lcd_gpio_2      = port:PA20<1><0><default><1>

lcdd3           = port:PD03<2><0><default><default>
lcdd4           = port:PD04<2><0><default><default>
lcdd5           = port:PD05<2><0><default><default>
lcdd6           = port:PD06<2><0><default><default>
    
```

| | |
|----------|-------------------------------------|
| lcdd7 | = port:PD07<2><0><default><default> |
| lcdd10 | = port:PD10<2><0><default><default> |
| lcdd11 | = port:PD11<2><0><default><default> |
| lcdd12 | = port:PD12<2><0><default><default> |
| lcdclk | = port:PD24<2><0><default><default> |
| lcdhsync | = port:PD26<2><0><default><default> |
| lcdvsync | = port:PD27<2><0><default><default> |

8.2.3. lcd0_panel_cfg.c

如下代码，修改 lcd_panel_init 函数。使用 LCD_GPIO_write 对 LCD 屏进行初始化操作。

```
static void LCD_panel_init(__u32 sel)
{
    lcd_panel_t27p06_init();
}
static void lcd_panel_t27p06_init(void)
{
    t27p06_spi_wr(0x055f);
    LCD_delay_ms(5);
    t27p06_spi_wr(0x051f);//reset
    LCD_delay_ms(10);
    t27p06_spi_wr(0x055f);
    LCD_delay_ms(50);
    t27p06_spi_wr(0x2b01);//exit standby mode
    t27p06_spi_wr(0x0009);//vcomac
    t27p06_spi_wr(0x019f);//vcomdc
    t27p06_spi_wr(0x040b);//8-bit rgb interface
    t27p06_spi_wr(0x1604);//default gamma setting  2.2
}

#define t27p06_spi_scl_1 LCD_GPIO_write(0,0,1);
#define t27p06_spi_scl_0 LCD_GPIO_write(0,0,0);
#define t27p06_spi_sdi_1 LCD_GPIO_write(0,1,1);
#define t27p06_spi_sdi_0 LCD_GPIO_write(0,1,0);
#define t27p06_spi_cs_1    LCD_GPIO_write(0,2,1);
#define t27p06_spi_cs_0    LCD_GPIO_write(0,2,0);

static void t27p06_spi_wr(__u32 value)
{
```

```
    __u32 i;  
    t27p06_spi_cs_1;  
    t27p06_spi_sdi_1;  
    t27p06_spi_scl_1;  
    LCD_delay_us(10);  
    t27p06_spi_cs_0;  
    LCD_delay_us(10);  
    for(i=0;i<16;i++)  
    {  
        LCD_delay_us(10);  
        t27p06_spi_scl_0;  
        if(value & 0x8000)  
            t27p06_spi_sdi_1;  
        else  
            t27p06_spi_sdi_0;  
        value <<= 1;  
        LCD_delay_us(10);  
        t27p06_spi_scl_1;  
    }  
    LCD_delay_us(10);  
    t27p06_spi_sdi_1;  
    t27p06_spi_cs_1;  
}
```

8.3. A20 与 CPU Parallel 18bit 屏

8.3.1. schematic

CPU 屏采用 I80 总线协议，具有 RD、WR、DS、CS 4 个总线控制信号。RD 是读控制，WR 是写控制，DS 是 Command/Data 选择，CS 是屏的片选。

CPU Parallel 屏有 18/16 个总线数据 IO，CPU Serial 屏有 6/8/9 个总线数据 io。

如图 4-7 是一个典型 CPU Parallel 18bit 屏的模组规格书的引脚定义。A20 与该 LCD 屏的引脚连接可参考图 4-8。

| Pin No | Symbol | Function | Notes |
|--------|----------|--|--------|
| 1-3 | NC | Not Connect | |
| 4 | IM3 | 16/18 bit select pin | |
| 5 | NC | Not Connect | |
| 6 | RESET | Reset pin. | |
| 7 | VSYNC | Vertical synchronization signal input pin | |
| 8 | HSYNC | Horizontal synchronization signal input pin | |
| 9 | DOTCLK | Dot clock signal input used in the RGB interface circuit | |
| 10 | DEN | Enable signal pin used in RGB interface circuit | |
| 11-28 | DB17-DB0 | ata bus | 总线数据引脚 |
| 29 | RD | Read enable clock input pin | |
| 30 | WR | Write enable clock input pin | |
| 31 | DC | ly data select pin | 总线控制引脚 |
| 32 | CS | Chip select pin | |
| 33 | VSS | Gound | |
| 34 | VCC | Power supply | |
| 35 | LED- | The LED power supply (-) | |
| 36 | LED+ | The LED power supply (+) | |

图 4-7 CPU Parallel 18bit 屏引脚定义

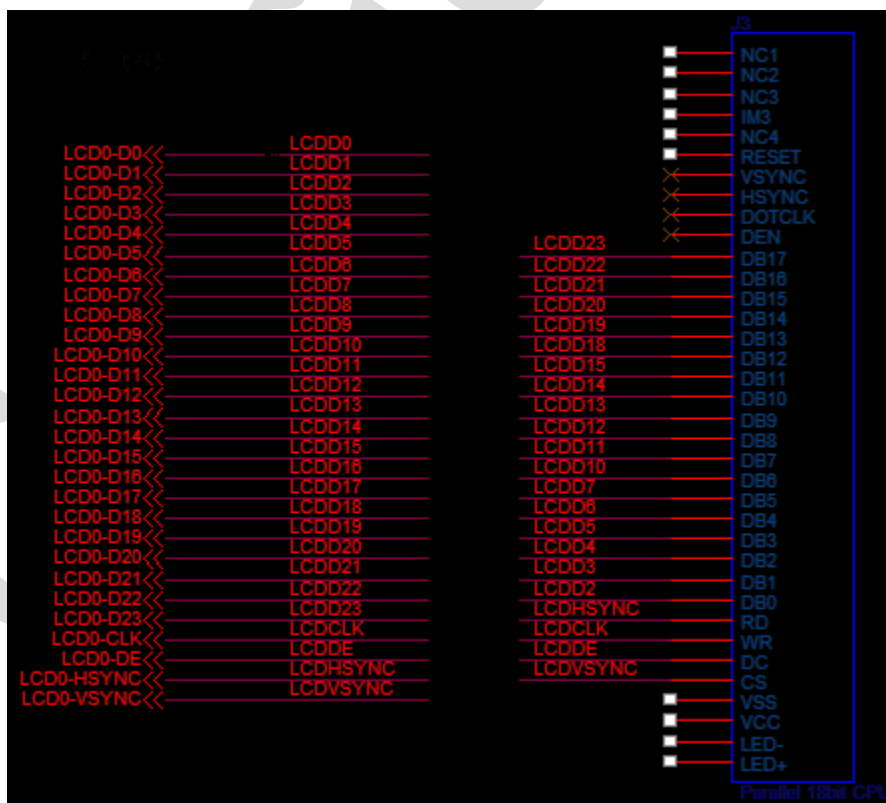


图 4-8 A20 与 CPU Parallel 18bit 屏连接图

8.3.2. sys_config.fex

该 LCD 屏的 sys_config.fex 配置如下。该 LCD 屏为 Paralle 18bit，没有 TE 引脚，故 lcd_cpu_if=0,lcd_cpu_te=0。

CPU 屏规格书没有行列的时序参数，只需满足 A20 的 LCD 控制器要求即可。时序参数确定流程如下：

- a、先取 lcd_hspw=10;
- b、lcd_hbp>hspw，取 lcd_hbp=20;
- C、lcd_ht>lcd_x×cycle+lcd_hbp=240×1+20，取 lcd_ht=300;
- d、同理 lcd_vspw=10;lcd_vbp=20;lcd_vt=500;
- e、lcd_dclk_freq=lcd_ht×lcd_vt×fps=300×500×60=9M。

该 CPU 屏为 18bit，为达到 24bit 的显示效果，lcd_frm=1。

```

lcd_if          = 1
lcd_cpu_if      = 0
lcd_cpu_te      = 0

lcd_dclk_freq   = 9
lcd_x           = 240
lcd_y           = 320
lcd_ht          = 300
lcd_hbp         = 20
lcd_hspw        = 10
lcd_vt          = 1000
lcd_vbp         = 20
lcd_vspw        = 10

lcd_frm         = 1

lcdd2           = port:PD02<2><0><default><default>
lcdd3           = port:PD03<2><0><default><default>
lcdd4           = port:PD04<2><0><default><default>
lcdd5           = port:PD05<2><0><default><default>
lcdd6           = port:PD06<2><0><default><default>
lcdd7           = port:PD07<2><0><default><default>
lcdd10          = port:PD10<2><0><default><default>
lcdd11          = port:PD11<2><0><default><default>
lcdd12          = port:PD12<2><0><default><default>
lcdd13          = port:PD13<2><0><default><default>

```



```

lcdd14           = port:PD14<2><0><default><default>
lcdd15           = port:PD15<2><0><default><default>
lcdd16           = port:PD16<2><0><default><default>
lcdd19           = port:PD19<2><0><default><default>
lcdd20           = port:PD20<2><0><default><default>
lcdd21           = port:PD21<2><0><default><default>
lcdd22           = port:PD22<2><0><default><default>
lcdd23           = port:PD23<2><0><default><default>
lcdclk           = port:PD24<2><0><default><default>
lcdde            = port:PD25<2><0><default><default>
lcdhsync        = port:PD26<2><0><default><default>
lcdvsync        = port:PD27<2><0><default><default>

```

8.3.3. lcd0_panel_cfg.c

如下代码，修改 lcd_panel_init 函数。使用 LCD_CPU_register_irq 注册一个每帧对 CPU 屏写坐标原点的操作，使用 LCD_CPU_WR 对 CPU 屏进行初始化操作。

```

static void LCD_panel_init(__u32 sel)
{
    lcd_panel_kgm281e0_init(sel);
    LCD_CPU_register_irq(sel, lcd_panel_kgm281e0_organ);
}

static void lcd_panel_kgm281e0_organ()
{
    LCD_CPU_WR(sel, 0x0050, 0); //
    LCD_CPU_WR(sel, 0x0051, 239); //
    LCD_CPU_WR(sel, 0x0052, 0); //
    LCD_CPU_WR(sel, 0x0053, 319); //
    LCD_CPU_WR(sel, 0x0020, 0); // GRAM horizontal Address
    LCD_CPU_WR(sel, 0x0021, 0); // GRAM Vertical Address
    LCD_CPU_WR_INDEX(sel, 0x0022);
}

static void lcd_panel_kgm281e0_init(__u32 sel)
{
    LCD_CPU_WR(sel, 0x0000, 0x0001);
    LCD_CPU_WR(sel, 0x0001, 0x0100); // set SS and SM bit
    LCD_CPU_WR(sel, 0x0002, 0x0400); // set 1 line inversion
    LCD_CPU_WR(sel, 0x0003, 0x10B0); // vertical scan
    LCD_CPU_WR(sel, 0x0004, 0x0000); // Resize register
}

```

```

LCD_CPU_WR(sel,0x0008, 0x0202); //
LCD_CPU_WR(sel,0x0009, 0x0000); //
LCD_CPU_WR(sel,0x000A, 0x0000); // FMARK function
LCD_CPU_WR(sel,0x000C, 0x0000); // RGB interface setting
LCD_CPU_WR(sel,0x000D, 0x0000); // Frame marker Position
LCD_CPU_WR(sel,0x000F, 0x0000); // RGB interface polarity
LCD_CPU_WR(sel,0x0010, 0x0000); //
LCD_CPU_WR(sel,0x0011, 0x0007); // DC1[2:0], DC0[2:0], VC[2:0]
LCD_CPU_WR(sel,0x0012, 0x0000); // VREG1OUT voltage
LCD_CPU_WR(sel,0x0013, 0x0000); // VDV[4:0] for VCOM amplitude
LCD_delay_ms(50); //
LCD_CPU_WR(sel,0x0010, 0x17B0); //
LCD_CPU_WR(sel,0x0011, 0x0001); // DC1[2:0], DC0[2:0], VC[2:0]
LCD_delay_ms(50); // Delay 50ms
LCD_CPU_WR(sel,0x0012, 0x013C); // VREG1OUT voltage
LCD_delay_ms(50); // Delay 50ms
LCD_CPU_WR(sel,0x0013, 0x1300); // VDV[4:0] for VCOM amplitude
LCD_CPU_WR(sel,0x0029, 0x0012); // VCM[4:0] for VCOMH
LCD_delay_ms(50); // Delay 50ms

LCD_CPU_WR(sel,0x0020, 0x0000); // GRAM horizontal Address
LCD_CPU_WR(sel,0x0021, 0x0000); // GRAM Vertical Address
LCD_CPU_WR(sel,0x002B, 0x0020); //

LCD_CPU_WR(sel,0x0030, 0x0000); // Gamma Control
LCD_CPU_WR(sel,0x0031, 0x0306); // Gamma Control
LCD_CPU_WR(sel,0x0032, 0x0200); // Gamma Control
LCD_CPU_WR(sel,0x0035, 0x0107); // Gamma Control
LCD_CPU_WR(sel,0x0036, 0x0404); // Gamma Control
LCD_CPU_WR(sel,0x0037, 0x0606); // Gamma Control
LCD_CPU_WR(sel,0x0038, 0x0105); // Gamma Control
LCD_CPU_WR(sel,0x0039, 0x0707); // Gamma Control
LCD_CPU_WR(sel,0x003C, 0x0600); // Gamma Control
LCD_CPU_WR(sel,0x003D, 0x0807); // Gamma Control

LCD_CPU_WR(sel,0x0050, 0x0000); //
LCD_CPU_WR(sel,0x0051, 0x00EF); //
LCD_CPU_WR(sel,0x0052, 0x0000); //
LCD_CPU_WR(sel,0x0053, 0x013F); //

LCD_CPU_WR(sel,0x0060, 0x2700); // GS, NL[5:0], SCN[5:0]
LCD_CPU_WR(sel,0x0061, 0x0001); // NDL, VLE, REV

```

```
LCD_CPU_WR(sel,0x006A, 0x0000); // VL[8:0]

LCD_CPU_WR(sel,0x0080, 0x0000); //
LCD_CPU_WR(sel,0x0081, 0x0000); //
LCD_CPU_WR(sel,0x0082, 0x0000); //
LCD_CPU_WR(sel,0x0083, 0x0000); //
LCD_CPU_WR(sel,0x0084, 0x0000); //
LCD_CPU_WR(sel,0x0085, 0x0000); //

LCD_CPU_WR(sel,0x0090, 0x0013); //
LCD_CPU_WR(sel,0x0092, 0x0000); //
LCD_CPU_WR(sel,0x0093, 0x0003); //
LCD_CPU_WR(sel,0x0095, 0x0110); //
LCD_CPU_WR(sel,0x0097, 0x0000); //
LCD_CPU_WR(sel,0x0098, 0x0000); //

LCD_CPU_WR(sel,0x0007, 0x0001); // Display Control 1
LCD_delay_ms(50);
LCD_CPU_WR(sel,0x0007, 0x0021); // Display Control 1
LCD_CPU_WR(sel,0x0007, 0x0023); // Display Control 1
LCD_delay_ms(50);
LCD_CPU_WR(sel,0x0007, 0x0173); //
}
```

8.4. A20 与 LVDS Single Link 屏

8.4.1. schematic

LVDS 屏接口分为 Single Link 和 Dual Link 两种接口。LVDS 屏使用 LVDS 差分信号，LVDS Single Link 具有 1 clock pair 和 3/4 data pair。

如图 4-9 是一个典型 LVDS Single Link 屏的模组规格书的引脚定义。A20 与该 LCD 屏的引脚连接可参考图 4-10。

| | | |
|----|--------------|-------------------------|
| 1 | VDD | Power Supply, 3.3V typ |
| 2 | VDD | Power Supply, 3.3V typ |
| 3 | VDD | Power Supply, 3.3V typ |
| 4 | NC | NC |
| 5 | GND | Ground |
| 6 | RXINO- | D0- |
| 7 | RXINO+ | D0+ |
| 8 | GND | Ground |
| 9 | RXINO1- | D1- Data Pair |
| 10 | RXINO1+ | D1+ |
| 11 | GND | Ground |
| 12 | RXINO2- | D2- |
| 13 | RXINO2+ | D2+ |
| 14 | GND | Ground |
| 15 | RXCLKIN- | CLK- Clock Pair |
| 16 | RXCLKIN+ | CLK+ |
| 17 | GND | Ground |
| 18 | NC | NC |
| 19 | NC | NC |
| 20 | GND | Ground |
| 21 | LVBIT | GND (6 or 8bit Change) |
| 22 | DITHER | GND (FRC) |
| 23 | GND | Ground |
| 24 | LED EN (PWM) | PWM |
| 25 | LVFMT | GND (MSB/LSB Change) |
| 26 | BIST | NC |
| 27 | VLED | LED Power Supply , 3~5V |
| 28 | VLED | LED Power Supply , 3~5V |
| 29 | VLED | LED Power Supply , 3~5V |
| 30 | VLED | LED Power Supply , 3~5V |
| 31 | NC | NC |

图 4-9 LVDS Single Link 屏引脚定义

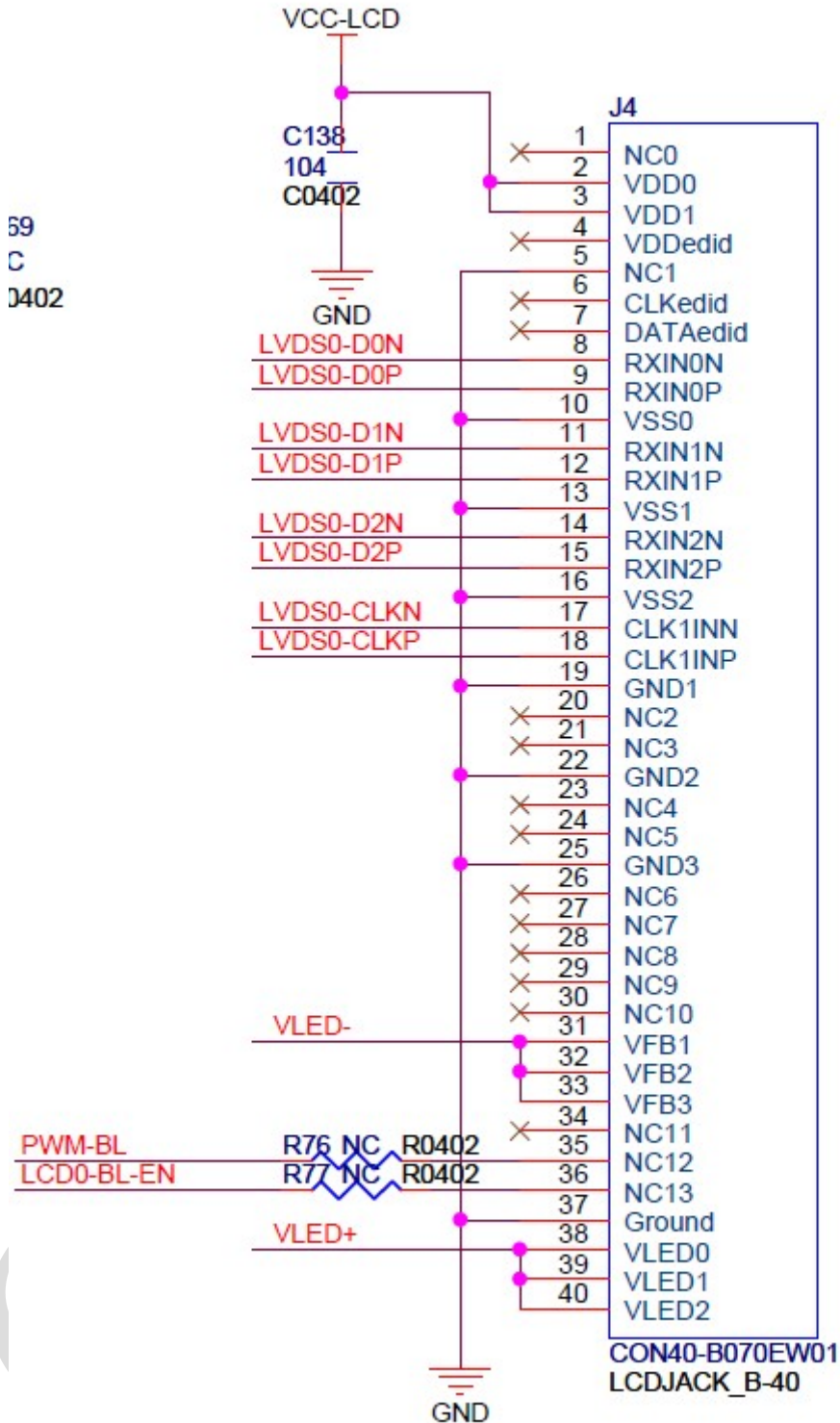


图 4-10 A20 与 CPU Single Link 屏连接图

8.4.2. sys_config.fex

该 LCD 屏有 3 data pair，为 18bit 色深，18bit 色深不区分模式。故 lcd_lvds_colordepth=1, lcd_lvds_mode=0.

该 LCD 屏参数如图 4-11 所示。时序参数配置与 HV parallel RGB 类似。区别于 HV parallel RGB，该 LCD 屏参数没有指定 Back Porch 和 Sync Width。根据 A20 时序要求，配置如下。

lcd_ht>lcd_x×cycle+lcd_hbp 得 lcd_hbp<64。取 lcd_hbp=20;

lcd_hbp>lcd_hspw，取 lcd_hspw=10。

同理，lcd_vbp=20， lcd_vspw=10。

LCD IO 必须配置为 LVDS，即第一个尖括号内容为 3。

sys_config.fex 文件中配置如下。

| ITEM | | | SYMBOL | MIN | TYP | MAX | UNIT | |
|---------------|------------|------------|------------------------|------------------|-----|-------|------|------------------|
| LCD Timing | Frame Rate | | - | TBD | 60 | TBD | Hz | |
| | DCLK | | Frequency | f _{CLK} | TBD | 66.77 | TBD | MHz |
| | DENA | Horizontal | Horizontal total time | t _H | TBD | 864 | TBD | t _{CLK} |
| | | | Horizontal Active time | t _{HA} | TBD | 800 | TBD | t _{CLK} |
| | | | Horizontal Blank time | t _{HB} | TBD | 64 | TBD | t _{CLK} |
| | Vertical | Vertical | Vertical total time | t _V | TBD | 1288 | TBD | t _H |
| | | | Vertical Active time | t _{VA} | TBD | 1200 | TBD | t _H |
| | | | Vertical Blank time | t _{VB} | TBD | 8 | TBD | t _H |

图 4-11 LVDS Single Link 屏参数

```

lcd_if          = 3
lcd_lvds_if     = 0
lcd_lvds_colordepth = 1
lcd_lvds_mode   = 0
    
```

```

lcd_dclk_freq  = 67
lcd_x          = 800
lcd_y          = 1200
lcd_ht        = 864
lcd_hbp       = 20
lcd_hspw      = 10
lcd_vt        = 2576
lcd_vbp       = 20
lcd_vspw      = 10
    
```

```

lcdd0          = port:PD00<3><0><default><default>
lcdd1          = port:PD01<3><0><default><default>
lcdd2          = port:PD02<3><0><default><default>
    
```

| | |
|-------|-------------------------------------|
| lcdd3 | = port:PD03<3><0><default><default> |
| lcdd4 | = port:PD04<3><0><default><default> |
| lcdd5 | = port:PD05<3><0><default><default> |
| lcdd6 | = port:PD06<3><0><default><default> |
| lcdd7 | = port:PD07<3><0><default><default> |

8.5. A20 与 LVDS Dual Link 屏

8.5.1. schematic

如图 4-12 是一个典型 LVDS Single Link 屏的模组规格书的引脚定义。A20 与该 LCD 屏的引脚连接可参考图 4-13.



| Pin | Symbol | Description |
|-----|------------|---|
| 1 | GND | Ground |
| 2 | GND | Ground |
| 3 | AVDD | Power Supply, 3.3V Typ. |
| 4 | AVDD | Power Supply, 3.3V Typ. |
| 5 | AVDD | Power Supply, 3.3V Typ. |
| 6 | DVDD | Digital Power supply (3.3V Typ) |
| 7 | DVDD | Digital Power supply (3.3V Typ) |
| 8 | Clk EEDID | Two wire serial interface clock |
| 9 | DATA EEDID | Two wire serial interface data |
| 10 | RXinO0- | - LVDS differential data input, Chan 0-Odd |
| 11 | RXinO0+ | + LVDS differential data input, Chan 0-Odd |
| 12 | GND | Ground |
| 13 | RXinO1 | - LVDS differential data input, Chan 1-Odd |
| 14 | RXinO1+ | + LVDS differential data input, Chan 1-Odd |
| 15 | GND | Ground |
| 16 | RXinO2- | - LVDS differential data input, Chan 2-Odd |
| 17 | RXinO2+ | + LVDS Odd Link 2-Odd |
| 18 | GND | Ground |
| 19 | RXOC- | - LVDS Differential Clock input (Odd) |
| 20 | RXOC+ | + LVDS Differential Clock input (Odd) |
| 21 | GND | Ground |
| 22 | RXinO3- | - LVDS differential data input, Chan 3-Odd |
| 23 | RXinO3+ | + LVDS differential data input, Chan 3-Odd |
| 24 | GND | Ground |
| 25 | RXinE0- | - LVDS differential data input, Chan 0-Even |
| 26 | RXinE0+ | + LVDS differential data input, Chan 0-Even |
| 27 | GND | Ground |
| 28 | RXinE1- | - LVDS differential data input, Chan 1-Even |
| 29 | RXinE1+ | + LVDS differential data input, Chan 1-Even |
| 30 | GND | Ground |
| 31 | RXinE2- | - LVDS differential data input, Chan 2-Even |
| 32 | RXinE2+ | + LVDS Even Link 2-Even |
| 33 | GND | Ground |
| 34 | RXEC- | - LVDS Differential Clock input (Even) |
| 35 | RXEC+ | + LVDS Differential Clock input (Even) |
| 36 | GND | Ground |
| 37 | RXinE3- | - LVDS differential data input, Chan 3-Even |
| 38 | RXinE3+ | + LVDS differential data input, Chan 3-Even |
| 39 | GND | Ground |
| 40 | NC | No connection |

图 4-12 LVDS Dual Link 屏引脚定义

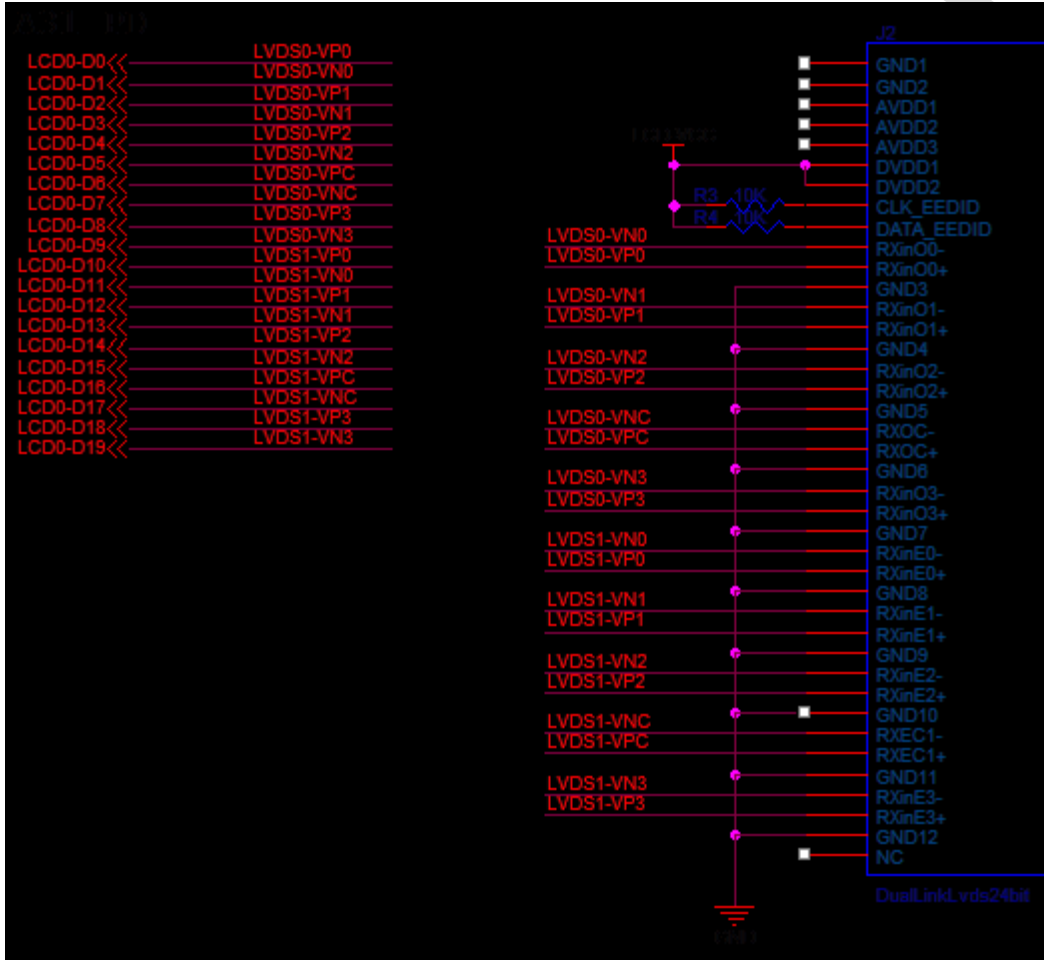


图 4-13 A20 与 LVDS Dual Link 屏连接图

8.5.2. sys_config.fex

该 LCD 屏每个 Link 各有 4 data pair, 为 24bit 色深, 格式如图 4-14, 为 JEIDA 模式。故 `lcd_lvds_colordepth=0`, `lcd_lvds_mode=1`。

该 LCD 屏参数如图 4-15 所示。在该 LCD 屏中, `clock frequency` 和 `Horizontal Period` 指的单个 link 上的时钟频率和 cycle 个数。

在 A20 中, `lcd_dclk_freq`, `lcd_ht` 是指两个 link 合并后的时钟频率和 cycle 个数。故 `lcd_dclk_freq = 76.36x2 = 153`, `lcd_ht = 1050x2=2100`。

sys_config.fex 文件中配置如下。

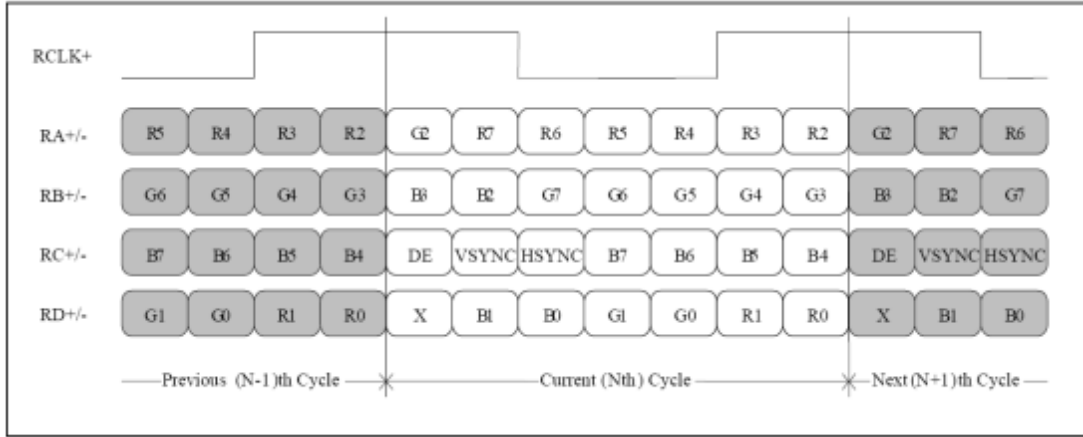


图 4-14 LVDS Dual Link 屏数据排布格式

| Parameter | Symbol | Min. | Typ. | Max. | Unit | |
|--------------------|---------------|----------|-------|------|------|-------------------------|
| Frame Rate | --- | -- | 60 | --- | Hz | |
| Clock frequency | $1/T_{Clock}$ | 64 | 76.36 | 85 | MHz | |
| Vertical Section | Period | T_V | 1210 | 1212 | 1240 | T_{Line} |
| | Active | T_{VD} | 1200 | | | |
| | Blanking | T_{VB} | 10 | 12 | 40 | |
| Horizontal Section | Period | T_H | 1034 | 1050 | 1140 | T_{Clock} (Note 2) |
| | Active | T_{HD} | 960 | | | |
| | Blanking | T_{HB} | 74 | 90 | 180 | |

图 4-15 LVDS Dual Link 屏参数

```

lcd_if                = 3
lcd_lvds_if           = 1
lcd_lvds_colordepth   = 0
lcd_lvds_mode         = 0

lcd_dclk_freq         = 153
lcd_x                  = 1920
lcd_y                  = 1200
lcd_ht                 = 2100
lcd_hbp                = 20
lcd_hspw               = 10
lcd_vt                 = 2424
    
```

lcd_vbp = 10
lcd_vspw = 5

lcdd0 = port:PD00<3><0><default><default>
lcdd1 = port:PD01<3><0><default><default>
lcdd2 = port:PD02<3><0><default><default>
lcdd3 = port:PD03<3><0><default><default>
lcdd4 = port:PD04<3><0><default><default>
lcdd5 = port:PD05<3><0><default><default>
lcdd6 = port:PD06<3><0><default><default>
lcdd7 = port:PD07<3><0><default><default>
lcdd8 = port:PD08<3><0><default><default>
lcdd9 = port:PD09<3><0><default><default>
lcdd10 = port:PD10<3><0><default><default>
lcdd11 = port:PD11<3><0><default><default>
lcdd12 = port:PD12<3><0><default><default>
lcdd13 = port:PD13<3><0><default><default>
lcdd14 = port:PD14<3><0><default><default>
lcdd15 = port:PD15<3><0><default><default>
lcdd16 = port:PD16<3><0><default><default>
lcdd17 = port:PD17<3><0><default><default>
lcdd18 = port:PD18<3><0><default><default>
lcdd19 = port:PD19<3><0><default><default>

8.6. A20+SSD2828 与 DSI 屏

8.6.1. schematic

MIPI DSI 屏分为 Video mode 和 Command mode 两种。两种模式的使用一样的 PIN 定义，都具有 1 clock pair 和 N data pair，N 为 1，2，3，4。CLK 是 DDR，在上下沿对 DATA 采样。

如图 4-16 是一个典型 DSI Video Mode 屏的模组规格书的引脚定义。A20 与该 LCD 屏的引脚连接可参考图 4-17。

| PIN No. | SYMBOL | DESCRIPTION | Note | PIN No. | SYMBOL | DESCRIPTION | Note | |
|---------|---------|----------------------------------|------|---------|---------|-----------------------------------|-------------------------------|----|
| 1 | VDD | Power Supply | 1) | 24 | NC | Keep Open | | |
| 2 | VDD | | | | 25 | CLKP | MIPI Clock pair sitive signal | |
| 3 | VDD | | | | 26 | NC | Keep Open | |
| 4 | VDD | | | | 27 | GND | GND(0V) | 3) |
| 5 | NC | Keep Open | | 28 | GND | GND(0V) | 3) | |
| 6 | SCL | I2C-bus Clock | | 29 | MIPI 1N | MIPI data pair 1 negative signal | | |
| 7 | GND | GND(0V) | 3) | 30 | NC | Keep Open | | |
| 8 | SDA | I2C-bus Data | 5) | 31 | MIPI 1P | MIPI data pair 1 positive signal | | |
| 9 | GND | GND(0V) | 3) | 32 | NC | Keep Open | | |
| 10 | GND | GND(0V) | 3) | 33 | GND | GND(0V) | 3) | |
| 11 | MIPI 3N | MIPI data pair 3 negative signal | | 34 | GND | GND(0V) | 3) | |
| 12 | NC | Keep Open | | 35 | MIPI 2N | MIPI data pair 2 negative signal | | |
| 13 | MIPI 3P | MIPI Data pair 3 positive signal | | 36 | NC | Keep Open | | |
| 14 | NC | Keep Open | | 37 | MIPI 2P | MIPI data pair 2 positive signal | | |
| 15 | GND | GND(0V) | 3) | 38 | LEDEN | LED enable input level | | |
| 16 | GND | GND(0V) | 3) | 39 | GND | GND(0V) | 3) | |
| 17 | MIPI 0N | MIPI data pair 0 negative signal | | 40 | BIST | Keep open or connect to GND | 4) | |
| 18 | NC | Keep Open | | 41 | LEDPWMI | PWM input to backlight LED driver | | |
| 19 | MIPI 0P | MIPI data pair 0 positive signal | | 42 | VLED | LED Power Supply | 2) | |
| 20 | NC | Keep Open | | 43 | VLED | | | |
| 21 | GND | GND(0V) | 3) | 44 | VLED | | | |
| 22 | GND | GND(0V) | 3) | 45 | VLED | | | |
| 23 | CLKN | MIPI Clock negative signal | | | | | | |

图 4-16 MIPI DSI Video Mode 引脚定义

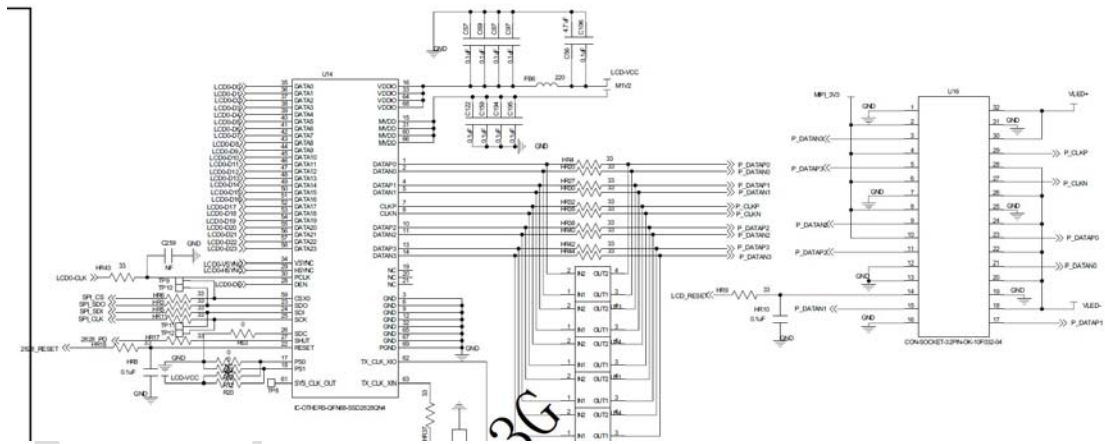


图 4-17 A20+SSD2828 与 MIPI DSI Video Mode 连接图

8.6.2. sys_config.fexs

该 LCD 屏具有 4 pair data，像素格式为 RGB888，故 `lcd_dsi_lane = 4`，`lcd_dsi_format = 0`。

该 LCD 屏参数如图 4-18 所示。时序参数配置与 LVDS 类似。注意 DSI 屏中，须满 A20 LCD 模块开发说明

足 lcd_hbp>40, lcd_vbp>10。

DSI 屏无须配置 LCD IO。sys_config.fex 文件配置如下。

| | ITEM | SYMBOL | Min. | Typ. | Max. | UNIT | NOTE |
|----|----------------------|--------|------|------|------|------|------|
| DE | Vertical Frequency | fV | 58 | 60 | 62 | Hz | |
| | Vertical Period | tV | 1216 | 1235 | 1262 | tH | |
| | Vertical Valid | tVD | 1200 | | | tH | |
| | Horizontal Frequency | fH | 70 | 74 | 78 | kHz | |
| | Horizontal Period | tH | 1980 | 2080 | 2200 | tCLK | |
| | Horizontal Valid | tHD | 1920 | | | tCLK | |

图 4-18 MIPI DSI Video Mode 屏参数

```

lcd_if           = 4
lcd_hbp         = 128
lcd_ht          = 948
lcd_vbp         = 80
lcd_vt          = 2280
lcd_vspw        = 50
lcd_hspw        = 64
lcd_hv_if       = 0
lcd_hv_smode    = 0
lcd_hv_s888_if  = 0
lcd_hv_syuv_if  = 0
lcd_lvds_ch     = 0
lcd_lvds_mode   = 0
lcd_lvds_bitwidth = 0
lcd_lvds_io_cross = 0
lcd_cpu_if      = 0
lcd_frm         = 1
lcd_io_cfg0     = 0x04000000
    
```

8.6.3. lcd0_panel_cfg.c

如下代码，修改 lcd_panel_init 函数。使用 lp079x01_init 函数对 MIPI DSI 屏进行初始化操作。

```

static void LCD_panel_init(__u32 sel)
{
    __panel_para_t *info = kmalloc(sizeof(__panel_para_t), GFP_KERNEL |
    __GFP_ZERO);
    
```

```
lcd_get_panel_para(sel, info);
if(info->lcd_if == LCD_IF_HV2DSI)
{
    lcd_2828_rst(0);
    lcd_panel_rst(0);
    lcd_2828_pd(0);
    LCD_delay_ms(20);
    lcd_2828_rst(1);
    lcd_panel_rst(1);
    LCD_delay_ms(50);
    lp079x01_init();
}
kfree(info);
return;
}
void lp079x01_init(void)
{
    spi_24bit_3wire(0x7000B1); //VSA=50, HAS=64
    spi_24bit_3wire(0x723240);

    spi_24bit_3wire(0x7000B2); //VBP=30+50, HBP=56+64
    spi_24bit_3wire(0x725078);

    spi_24bit_3wire(0x7000B3); //VFP=36, HFP=60
    spi_24bit_3wire(0x72243C);

    spi_24bit_3wire(0x7000B4); //HACT=768
    spi_24bit_3wire(0x720300);

    spi_24bit_3wire(0x7000B5); //VACT=1240
    spi_24bit_3wire(0x720400);

    spi_24bit_3wire(0x7000B6);
    //todo, cfg to 18bpp packed
    spi_24bit_3wire(0x72000B); //0x720009:burst mode, 18bpp packed
                                //0x72000A:burst mode,
18bpp loosely packed
                                //0x72000B:burst mode, 24bpp

    spi_24bit_3wire(0x7000DE); //no of lane=4
    spi_24bit_3wire(0x720003);
```

```
spi_24bit_3wire(0x7000D6); //RGB order and packet number in blanking period
spi_24bit_3wire(0x720005);

spi_24bit_3wire(0x7000B9); //disable PLL
spi_24bit_3wire(0x720000);

spi_24bit_3wire(0x7000BA); //lane speed=560
spi_24bit_3wire(0x72C015); //may modify according to requirement, 500Mbps to
560Mbps, (n+1)*12M

spi_24bit_3wire(0x7000BB); //LP clock
spi_24bit_3wire(0x720008);

spi_24bit_3wire(0x7000B9); //enable PPL
spi_24bit_3wire(0x720001);

spi_24bit_3wire(0x7000c4); //enable BTA
spi_24bit_3wire(0x720001);

spi_24bit_3wire(0x7000B7); //enter LP mode
spi_24bit_3wire(0x720342);

spi_24bit_3wire(0x7000B8); //VC
spi_24bit_3wire(0x720000);

spi_24bit_3wire(0x7000BC); //set packet size
spi_24bit_3wire(0x720000);

spi_24bit_3wire(0x700011); //sleep out cmd

LCD_delay_ms(200);
spi_24bit_3wire(0x700029); //display on

LCD_delay_ms(200);
spi_24bit_3wire(0x7000B7); //video mode on
spi_24bit_3wire(0x72030b);
}
```

8.7. 电源和背光

LCD 供电，在 A20 方案中，使用 gpio 控制。

LCD 背光，常见有三种电路，如图

A 电路中，PWM 占空比越高，背光越暗，配置 `lcd_pwm_pol=1`，为反极性。建议 `lcd_pwm_freq=50000`，须配置 `lcd_bl_en` 和 `lcd_pwm` 两个 PIN。

B 电路中，PWM 占空比越高，背光越亮，配置 `lcd_pwm_pol=0`，为正极性。
`lcd_pwm_freq` 频率由背光 IC 决定；须配置 `lcd_pwm` 一个 PIN。

C 电路中，PWM 占空比越高，背光越亮，配置 `lcd_pwm_pol=0`，为正极性。

`lcd_pwm_freq` 频率由 LCD 屏决定；一般须配置 `lcd_bl_en` 和 `lcd_pwm` 两个 PIN。

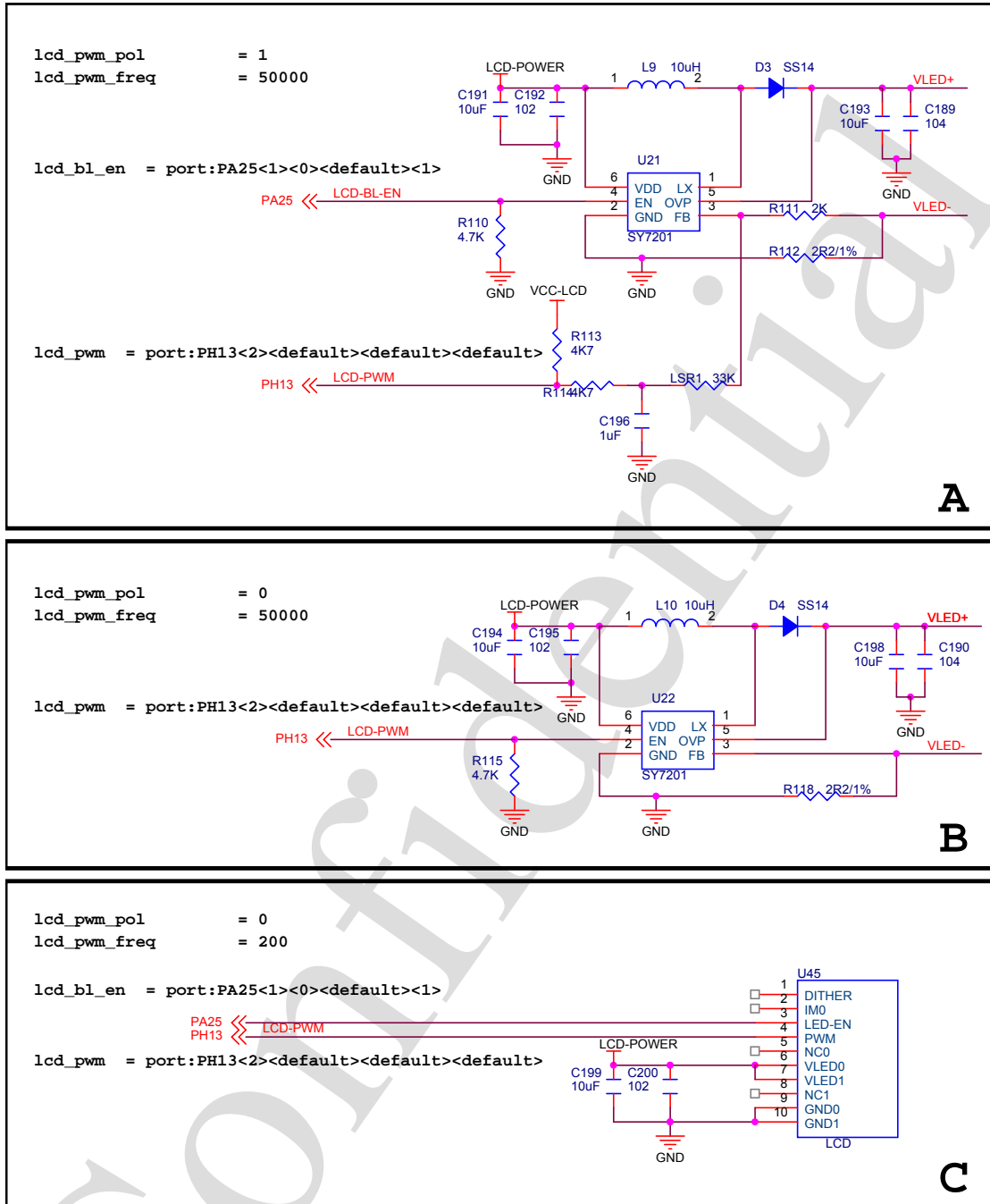


图 4-24 常用 LCD 背光电路

8.8. A20 LCD IO PORT 定义

表 5-1 A20 LCD IO 定义

| | | | |
|--|-----------|------------|---------|
| | 同步 RGB 接口 | CPU/I80 接口 | LVDS 接口 |
|--|-----------|------------|---------|

| IO | Para | SetRGB | | | CCR | Para | Para | Set1 | | Set2 | | Single | Dual |
|---------|------|--------|-----|-----|-----|-------|-------|-------|-------|------|------|--------|-------|
| | RGB | | | | 666 | 16bit | 16bit | 16bit | 16bit | | Link | Link | |
| | | 1st | 2nd | 3rd | | | | 1st | 2n | 1s | 2n | | |
| PD27 | | VSYNC | | | | | | CS | | | | | |
| PD26 | | HSYNC | | | | | | RD | | | | | |
| PD25 | | DCLK | | | | | | WR | | | | | |
| PD24 | | DE | | | | | | RS | | | | | |
| PD23 | R | | | | | R | R | | | | | | |
| PD22 | R | | | | | R | R | | | | | | |
| PD21 | R | | | | | R | R | | | | | | |
| PD20 | R | | | | | R | R | | | | | | |
| PD19 | R | | | | | R | R | | | | | 1-D3N | E-D3N |
| PD18 | R | | | | | R | G | | | | | 1-D3P | E-D3P |
| PD17 | R | | | | | | | | | | | 1-CKN | E-CKN |
| PD16 | R | | | | | | | | | | | 1-CKP | E-CKP |
| PD15 | G | | | | | G | G | | | | | 1-D2N | E-D2N |
| PD14 | G | | | | | G | G | | | | | 1-D2P | E-D2P |
| PD13 | G | | | | | G | | | | | | 1-D1N | E-D1N |
| PD12 | G | D1 | D2 | D3 | D | G | G | R | G | R | G2 | 1-D1P | E-D1P |
| PD11 | G | D1 | D2 | D3 | D | G | G | R | G | R | G1 | 1-D0N | E-D0N |
| PD10 | G | D1 | D2 | D3 | D | G | G | R | G | R | G0 | 1-D0P | E-D0P |
| PD9 | G | | | | | | | | | | | 0-D3N | O-D3N |
| PD8 | G | | | | | | | | | | | 0-D3P | O-D3P |
| PD7 | B | D1 | D2 | D3 | D | B | B | R | B | R | B4 | 0-CKN | O-CKN |
| PD6 | B | D1 | D2 | D3 | D | B | B | R | B | R | B3 | 0-CKP | O-CKP |
| PD5 | B | D1 | D2 | D3 | D | B | B | R | B | G | B2 | 0-D2N | O-D2N |
| PD4 | B | D1 | D2 | D3 | D | B | B | G | B | G | B1 | 0-D2P | O-D2P |
| PD3 | B | D1 | D2 | D3 | D | B | B | G | B | G | B0 | 0-D1N | O-D1N |
| PD2 | B | | | | | B | | G | B | | | 0-D1P | O-D1P |
| PD1 | B | | | | | | | | | | | 0-D0N | O-D0N |
| PD0 | B | | | | | | | | | | | 0-D0P | O-D0P |
| DSI-D3N | | | | | | | | | | | | | |
| DSI-D3P | | | | | | | | | | | | | |
| DSI-D2N | | | | | | | | | | | | | |
| DSI-D2P | | | | | | | | | | | | | |
| DSI-CKN | | | | | | | | | | | | | |
| DSI-CKP | | | | | | | | | | | | | |
| DSI-D1N | | | | | | | | | | | | | |
| DSI-D1P | | | | | | | | | | | | | |
| DSI-D0N | | | | | | | | | | | | | |
| DSI-D0P | | | | | | | | | | | | | |

8.9. LCD CHECK LIST

LCD CHECK LIST

方案:

屏规格:

日期:

| | | DVDD | AVDD | VHG | VGL | VCOM |
|------|--------------------------------|--------------------------------|------|--------------------------------|-----|------|
| 电压检查 | 开屏状态 <input type="checkbox"/> | | | | | |
| | 关屏状态 <input type="checkbox"/> | | | | | |
| 效果检查 | 位置渐变图 <input type="checkbox"/> | 色深灰阶图 <input type="checkbox"/> | | 文字图 <input type="checkbox"/> | | |
| | 色块图 <input type="checkbox"/> | 亮度调节 <input type="checkbox"/> | | 快慢开关屏 <input type="checkbox"/> | | |
| 签名 | | | | | | |

9. Declaration

This(A20 LCD 模块开发说明) is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner.

The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.