全志科技
Allwinner Technology

A83T

显示驱动模块使用说明书

文档履历

版本号	日期	制/修订人	内容描述
V1.0	2014-10-16		初始版本

目录

1. 引言

1.1.　　编写目的

让显示应用开发人员了解显示驱动的接口及使用流程，快速上手，进行开发。

1.2.　　适用范围

适用于 A83T 及以后芯片。

1.3.　　相关人员

应用开发人员。

2. 驱动模块概述

本模块主要处理显示相关功能，主要功能如下：

- 支持 linux 标准的 framebuffer 接口
- 支持 lcd(hv/lvds/cpu/dsi)输出
- 支持多图层叠加混合处理
- 支持多种显示效果处理（alpha, colorkey, 图像增强，亮度/对比度/饱和度/色度调整）
- 支持智能背光调节
- 支持多种图像数据格式输入(arg,yuv)
- 支持图像缩放处理
- 支持截屏
- 支持图像转换

3. 驱动模块的使用

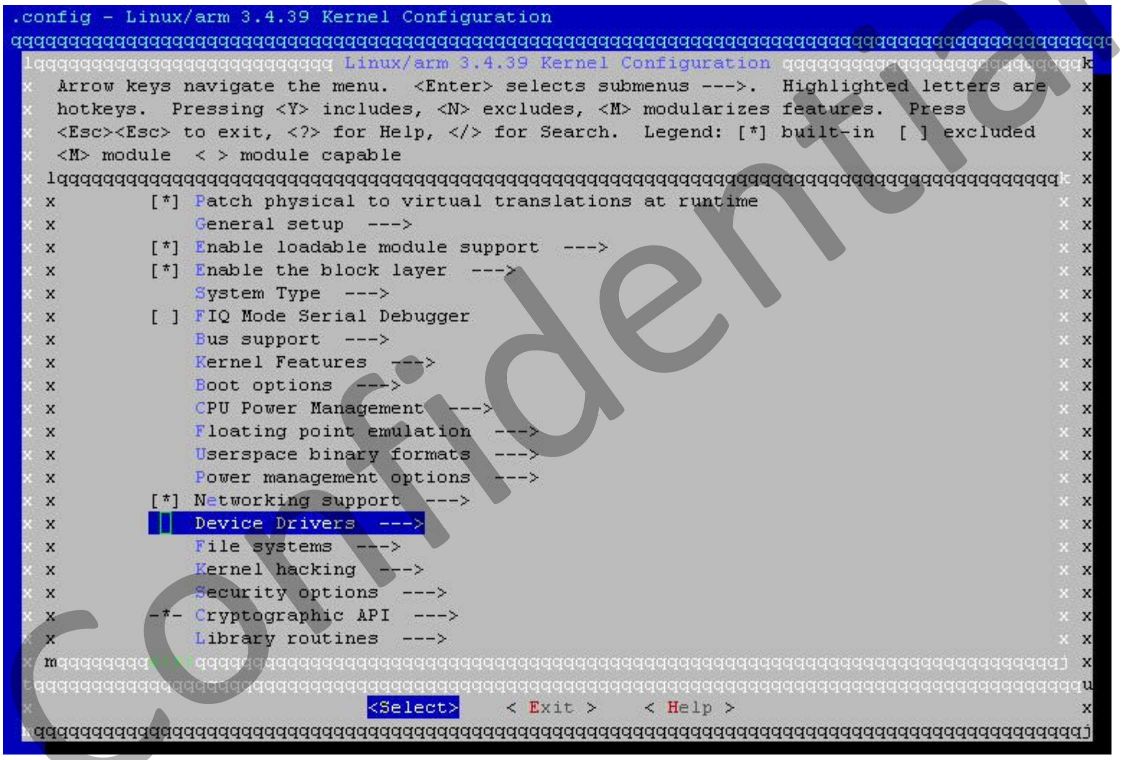
3.1.　　驱动模块适用范围

本驱动适用于 2 路显示，支持 LCD（HV/CPU/LVDS/DSI），支持 HDMI。

3.1.1.驱动模块配置信息

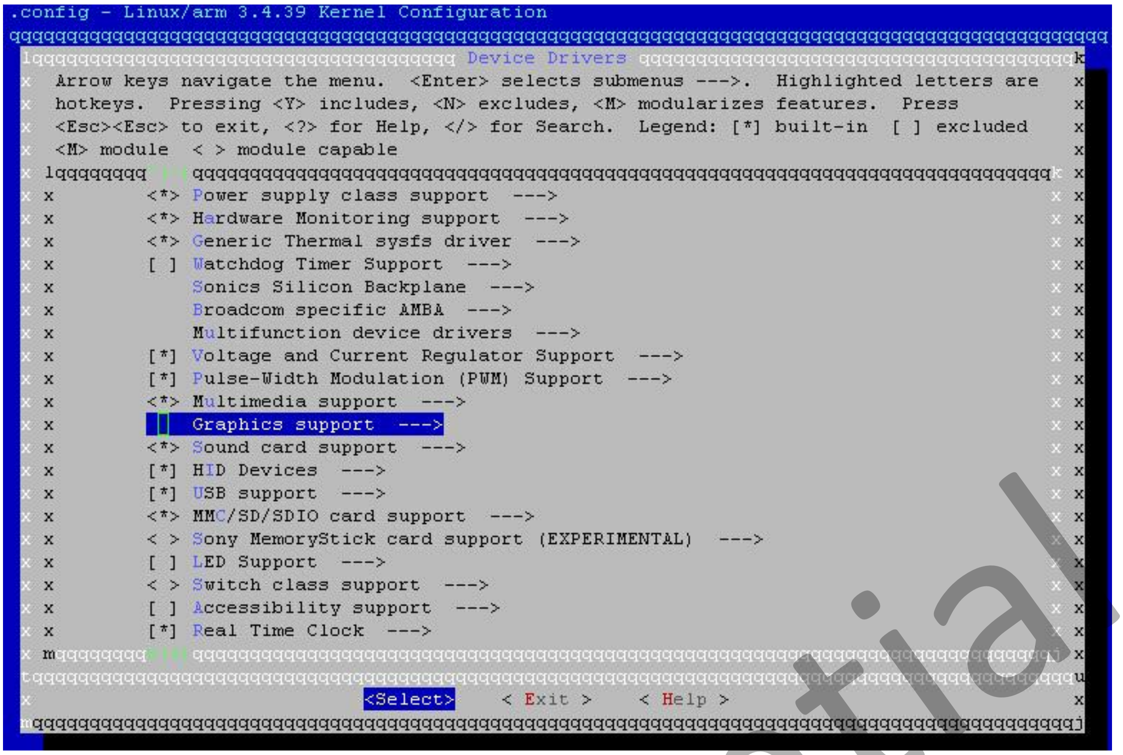
Menuconfig 配置

在 linux 目录下，#make ARCH=arm menuconfig，进入配置界面



```
.config - Linux/arm 3.4.39 Kernel Configuration
 lqqqqqqqqqqqqqqqqqqqqqqqqqq Linux/arm 3.4.39 Kernel Configuration qqqqqqqqqqqqqqqqqqqqqqqqqqk
 x  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are  x
 x  hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press         x
 x  <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  x
 x  <M> module  < > module capable                                                          x
 x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
 x x         [*] Patch physical to virtual translations at runtime                      x x
 x x             General setup  --->                                                    x x
 x x         [*] Enable loadable module support  --->                                   x x
 x x         [*] Enable the block layer  --->                                           x x
 x x             System Type  --->                                                      x x
 x x         [ ] FIQ Mode Serial Debugger                                               x x
 x x             Bus support  --->                                                      x x
 x x             Kernel Features  --->                                                  x x
 x x             Boot options  --->                                                     x x
 x x             CPU Power Management  --->                                             x x
 x x             Floating point emulation  --->                                         x x
 x x             Userspace binary formats  --->                                         x x
 x x             Power management options  --->                                         x x
 x x         [*] Networking support  --->                                               x x
 x x             Device Drivers  --->                                                   x x
 x x             File systems  --->                                                     x x
 x x             Kernel hacking  --->                                                   x x
 x x             Security options  --->                                                 x x
 x x         -*- Cryptographic API  --->                                                x x
 x x             Library routines  --->                                                 x x
 x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
 tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
 x                          <Select>    < Exit >    < Help >                                 x
 mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

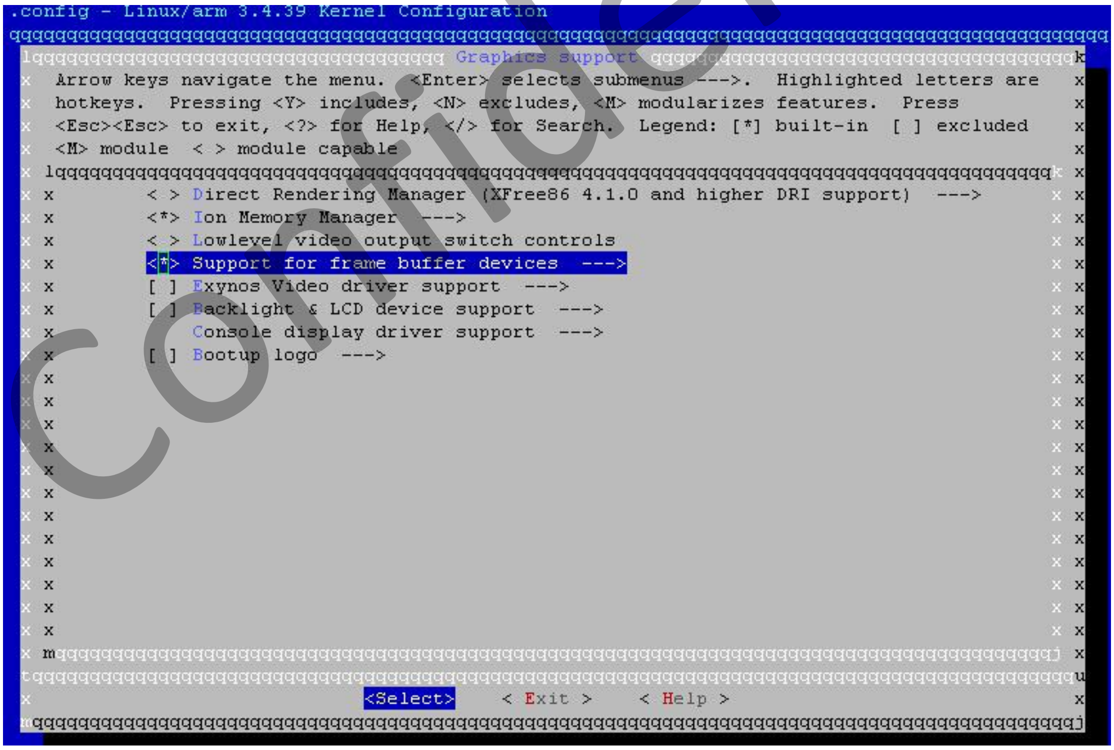
图 1 menuconfig-device drivers



```
.config - Linux/arm 3.4.39 Kernel Configuration
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
 lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq Device Drivers qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
 x  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are  x
 x  hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press        x
 x  <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded   x
 x  <M> module  < > module capable                                                          x
 x lqqqqqqqq^(-)qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
 x x        <*> Power supply class support  --->                                        x x
 x x        <*> Hardware Monitoring support  --->                                       x x
 x x        <*> Generic Thermal sysfs driver  --->                                      x x
 x x        [ ] Watchdog Timer Support  --->                                            x x
 x x            Sonics Silicon Backplane  --->                                          x x
 x x            Broadcom specific AMBA  --->                                            x x
 x x            Multifunction device drivers  --->                                      x x
 x x        [*] Voltage and Current Regulator Support  --->                             x x
 x x        [*] Pulse-Width Modulation (PWM) Support  --->                              x x
 x x        <*> Multimedia support  --->                                                x x
 x x            Graphics support  --->                                                  x x
 x x        <*> Sound card support  --->                                                x x
 x x        [*] HID Devices  --->                                                       x x
 x x        [*] USB support  --->                                                       x x
 x x        <*> MMC/SD/SDIO card support  --->                                          x x
 x x        < > Sony MemoryStick card support (EXPERIMENTAL)  --->                      x x
 x x        [ ] LED Support  --->                                                       x x
 x x        < > Switch class support  --->                                              x x
 x x        [ ] Accessibility support  --->                                             x x
 x x        [*] Real Time Clock  --->                                                   x x
 x mqqqqqqqqv(+)qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
 tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
 x                        <Select>    < Exit >    < Help >                                  x
 mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

图 2 menuconfig-graphics support

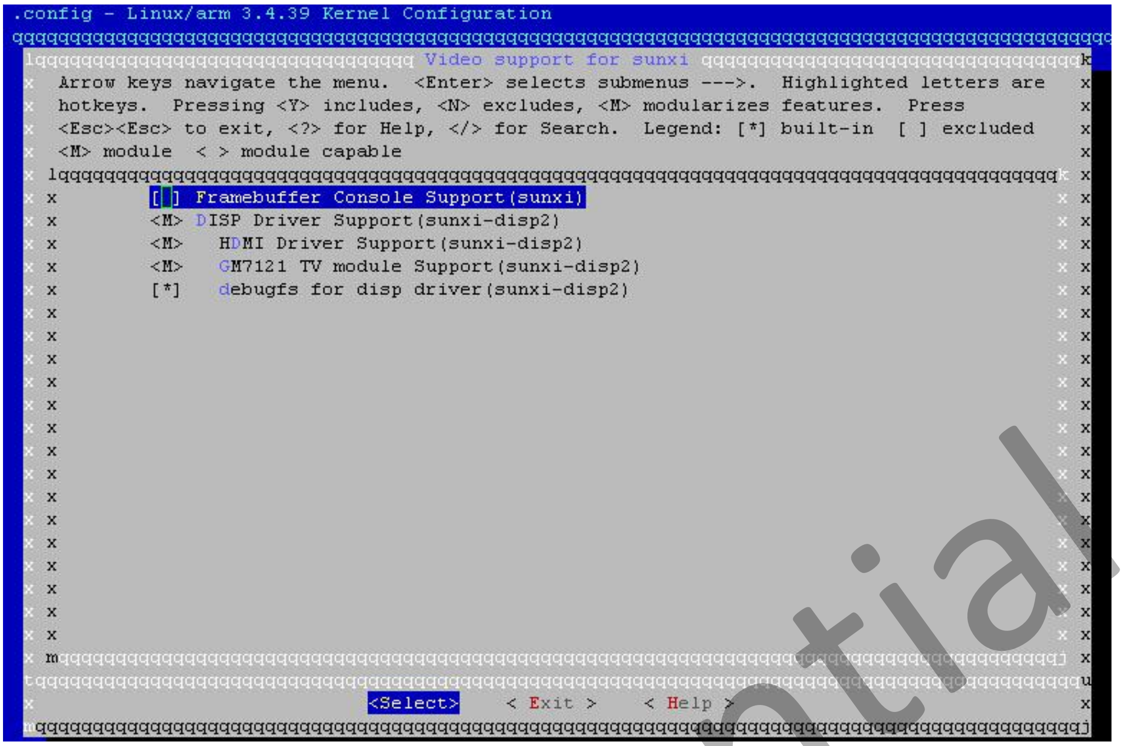
Framebuffer 驱动：



```
.config - Linux/arm 3.4.39 Kernel Configuration
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
 lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq Graphics support qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
 x  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are  x
 x  hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press        x
 x  <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded   x
 x  <M> module  < > module capable                                                          x
 x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
 x x        < > Direct Rendering Manager (XFree86 4.1.0 and higher DRI support)  --->   x x
 x x        <*> Ion Memory Manager  --->                                                x x
 x x        < > Lowlevel video output switch controls                                   x x
 x x        <*> Support for frame buffer devices  --->                                  x x
 x x        [ ] Exynos Video driver support  --->                                       x x
 x x        [ ] Backlight & LCD device support  --->                                    x x
 x x            Console display driver support  --->                                    x x
 x x        [ ] Bootup logo  --->                                                       x x
 x x                                                                                    x x
 x x                                                                                    x x
 x x                                                                                    x x
 x x                                                                                    x x
 x x                                                                                    x x
 x x                                                                                    x x
 x x                                                                                    x x
 x x                                                                                    x x
 x x                                                                                    x x
 x x                                                                                    x x
 x x                                                                                    x x
 x x                                                                                    x x
 x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
 tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
 x                        <Select>    < Exit >    < Help >                                  x
 mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

图 3 menuconfig-support for frame buffer devices

Disp/HDMI 驱动：



```
.config - Linux/arm 3.4.39 Kernel Configuration
 ---------------------------- Video support for sunxi ----------------------------
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are
  hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press
  <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded
  <M> module  < > module capable

              [ ] Framebuffer Console Support(sunxi)
              <M> DISP Driver Support(sunxi-disp2)
              <M>   HDMI Driver Support(sunxi-disp2)
              <M>   GM7121 TV module Support(sunxi-disp2)
              [*]   debugfs for disp driver(sunxi-disp2)

                       <Select>    < Exit >    < Help >
```

图 4 menuconfig －disp/hdmi driver support(sunxi)

Sys_config.fex 显示相关配置

在 A83T 中，我们做到了通过只配置屏参数，就可以完成换一个屏，不需要去修改代码（认证了的屏）。

```
;----------------------------------------------------------------------
;disp init configuration
;
;disp_mode                                    (0:screen0<screen0,fb0>;
1:screen1<screen1,fb0>)
;screenx_output_type   (0:none; 1:lcd; 3:hdmi;)
;screenx_output_mode   (used for hdmi output, 0:480i 1:576i 2:480p 3:576p
4:720p50)
;                      (5:720p60 6:1080i50 7:1080i60 8:1080p24 9:1080p50
10:1080p60)
;fbx format             (0:ARGB 1:ABGR 2:RGBA 3:BGRA 5:RGB565 8:RGB888
12:ARGB4444 16:ARGB1555 18:RGBA5551)
;fbx_width,fbx_height (framebuffer horizontal/vertical pixels, fix to
output resolution while equal 0)
;lcdx_backlight                          (lcd    init    backlight,the
range:[0,256],default:197
;----------------------------------------------------------------------
[disp_init]
```

```
disp_init_enable         = 1
disp_mode                = 0

screen0_output_type      = 1
screen0_output_mode      = 4

screen1_output_type      = 3
screen1_output_mode      = 4

fb0_format               = 0
fb0_width                = 0
fb0_height               = 0

fb1_format               = 0
fb1_width                = 0
fb1_height               = 0

lcd0_backlight           = 50
lcd1_backlight           = 50

;--------------------------------------------------------------------
;lcd0 configuration

;lcd_if:                  0:hv(sync+de); 1:8080; 2:ttl; 3:lvds; 4:dsi;
5:edp
;lcd_x:                 lcd horizontal resolution
;lcd_y:                 lcd vertical resolution
;lcd_width:             width of lcd in mm
;lcd_height:            height of lcd in mm
;lcd_dclk_freq:         in MHZ unit
;lcd_pwm_freq:          in HZ unit
;lcd_pwm_pol:           lcd backlight PWM polarity
;lcd_pwm_max_limit      lcd backlight PWM max limit(<=255)
;lcd_hbp:               hsync back porch
;lcd_ht:                hsync total cycle
;lcd_vbp:               vsync back porch
;lcd_vt:                vysnc total cycle
;lcd_hspw:              hsync plus width
;lcd_vspw:              vysnc plus width
;lcd_lvds_if:           0:single link;  1:dual link
;lcd_lvds_colordepth:   0:8bit; 1:6bit
;lcd_lvds_mode:         0:NS mode; 1:JEIDA mode
;lcd_frm:                  0:disable; 1:enable rgb666 dither; 2:enable
rgb656 dither
```

```
;lcd_gamma_en          lcd gamma correction enable
;lcd_bright_curve_en   lcd bright curve correction enable
;lcd_cmap_en           lcd color map function enable
;lcdgamma4iep:         Smart Backlight parameter, lcd gamma vale * 10;
;                        decrease it while lcd is not bright enough;
increase while lcd is too bright
;-------------------------------------------------------------------
[lcd0_para]
lcd_used            = 1

lcd_driver_name     = "lt070me05000"
lcd_bl_0_percent    = 0
lcd_bl_40_percent   = 23
lcd_bl_100_percent  = 100
lcd_if              = 4
lcd_x               = 1200
lcd_y               = 1920
lcd_width           = 94
lcd_height          = 150
lcd_dclk_freq       = 156
lcd_pwm_used        = 1
lcd_pwm_ch          = 0
lcd_pwm_freq        = 50000
lcd_pwm_pol         = 1
lcd_pwm_max_limit   = 255
lcd_hbp             = 80
lcd_ht              = 1320
lcd_hspw            = 20
lcd_vbp             = 20
lcd_vt              = 1960
lcd_vspw            = 1
lcd_frm             = 0
lcd_cmap_en         = 0
lcd_dsi_if          = 2
lcd_dsi_lane        = 4
lcd_dsi_format      = 0
lcd_dsi_te          = 0
lcd_gamma_en        = 0
lcd_bright_curve_en = 0
lcdgamma4iep        = 22

lcd_bl_en           = port:PD24<1><0><default><1>
lcd_power           = "vcc-dsi-18"
lcd_power1          = "vcc-lcd-0"
```

```
lcd_power2           = ″vcc_dsi″

lcd_gpio_0           = port:PD25<1><0><default><0>
lcd_gpio_1           = port:PD26<1><0><default><0>
lcd_io_regulator1    = ″vcc-pd″

;----------------------------------------------------------------------
;hdmi configuration
;----------------------------------------------------------------------
[hdmi_para]
hdmi_used            = 0

;----------------------------------------------------------------------
;pwm config
;----------------------------------------------------------------------
[pwm0_para]
pwm_used             = 0
pwm_positive         = port:PD28<2><0><default><default>
```

3.2.   驱动模块信息

源码结构介绍

```
drivers
├──video                                      显示驱动目录
│   ├──fbmem.c                                framebuffer core
│   ├──sunxi                                  display driver for sunxi
│   │   ├──disp2/                             disp2 的目录
│   │   │   ├──disp
│   │   │   │   ├──dev_disp.c                     display driver 层
│   │   │   │   ├──dev_fb.c                       framebuffer driver 层
│   │   │   │   ├──de                             bsp 层
│   │   │   │   │   ├──disp_lcd.c disp_manager.c ..
│   │   │   │   │   ├──disp_al.c                  al 层
│   │   │   │   │   │   └──lowlevel_sun*i/            lowlevel 层
│   │   │   │   │   │   ├──de_lcd.c de_rtmx.c...
│   │   │   │   │   └──disp_sys_int.          OSAL 层,与操作系统相关层
│    │    │    │    ├──lcd/                                    lcd driver
│    │    │    │    │    │—lcd_src_interface.c               与display驱动的接口
│    │    │    │    │    │—default_panel.c ...              平台已经支持的屏驱动
include
├──video                                                    video header dir
│    ├──sunxi_display2.h                                      display  header  file
```

编译方式

可以编译为单独的模块，也可以编译到内核。

## 3.3.	驱动模块加载

disp.ko　hdmi.ko

## 3.4.	驱动模块加载资源要求

由于模块依赖需要先加载 disp.ko，后加载 hdmi.ko。

## 3.5.	驱动模块调试测试

可以通过 cat 设备节点查看显示信息，也可以通过 dump 寄存器内容，分析运行状态等。

## 3.6.	其他注意事项

无

## 3.7.	使用示例

Android 通过调用 hwc 中的标准接口来和显示驱动模块通信。
在次阶段中完成下一帧中后端寄存器的设置。

4. LCD 屏的配置

4.1. LCD 时序图

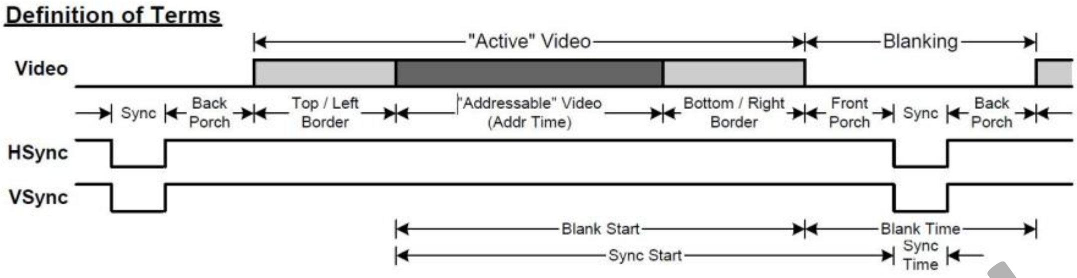


图 5 LCD 时序图

在Blanking分区，分为Front Porch ，Sync和Back Porch。
当进入Blanking分区时会触发一个VB中断，这时进入Front Porch，在中断里面设置set ready 然后硬件识别到该ready后,开始load reg 经过,start delay时间后，触发load finish中断，发送Vsync信号给上层。这时进入Sync阶段，经过VSPW时间后，进入Back Porch阶段。

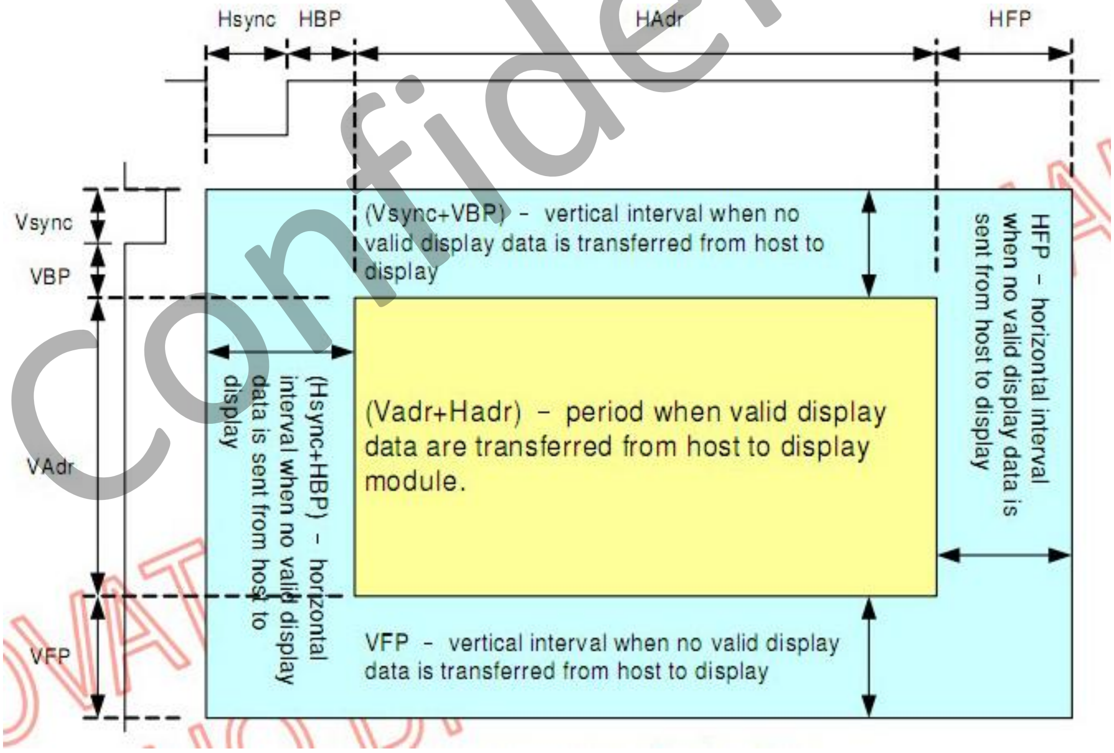

Hsync
HBP
HAdr
HFP
Vsync
VBP
VAdr
VFP
(Vsync+VBP) - vertical interval when no valid display data is transferred from host to display
(Hsync+HBP) - horizontal interval when no valid display data is sent from host to display
(Vadr+Hadr) - period when valid display data are transferred from host to display module.
HFP - horizontal interval when no valid display data is sent from host to display
VFP - vertical interval when no valid display data is transferred from host to display

图 6 LCD 时序图

正常的TFT一行的显示周期是前消隐+实际点输出+后消隐
HBP、HFP代表前后消隐所需要的周期
如果前消隐设置小了,后消隐设置大了,LCD控制器的实际输出就会被当做消隐而不会实际显示出来,看到的效果就是图像左移,反之图像右移
如果前后消隐都设置小了,理论上第二行的图像可能会被当做第一行的图像显示,

照成屏幕歪斜不同步,但是有些 TFT 中内部的时序电路会自动补上缺少的时钟,所以也不一定会看到不同步的画面

场的消隐同理

至于为什么要消隐是为了兼容 CRT 显示器的显示原理,CRT 显示器每一个扫描行完成后,电子枪需要回扫,这段时间不能显示,所以这段时间的视频信号需要暂停一下,就是消隐。

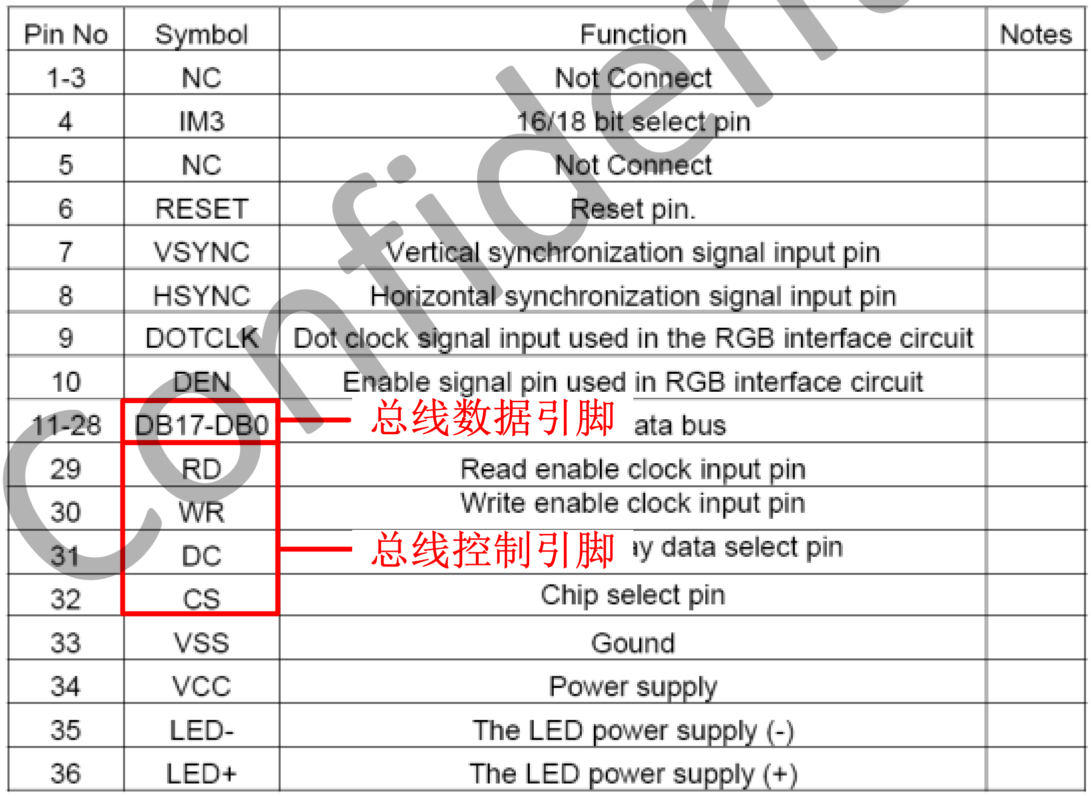
4.2. A83T 与 CPU Parallel 18bit 屏

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 7 是一个典型 CPU Parallel 18bit 屏的模组规格书的引脚定义。A83T 与该 LCD 屏的引脚连接可参考图 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	
29	RD	Read enable clock input pin	
30	WR	Write enable clock input pin	
31	DC	ıy data select pin	
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 7 CPU Parallel 18bit 屏引脚定义

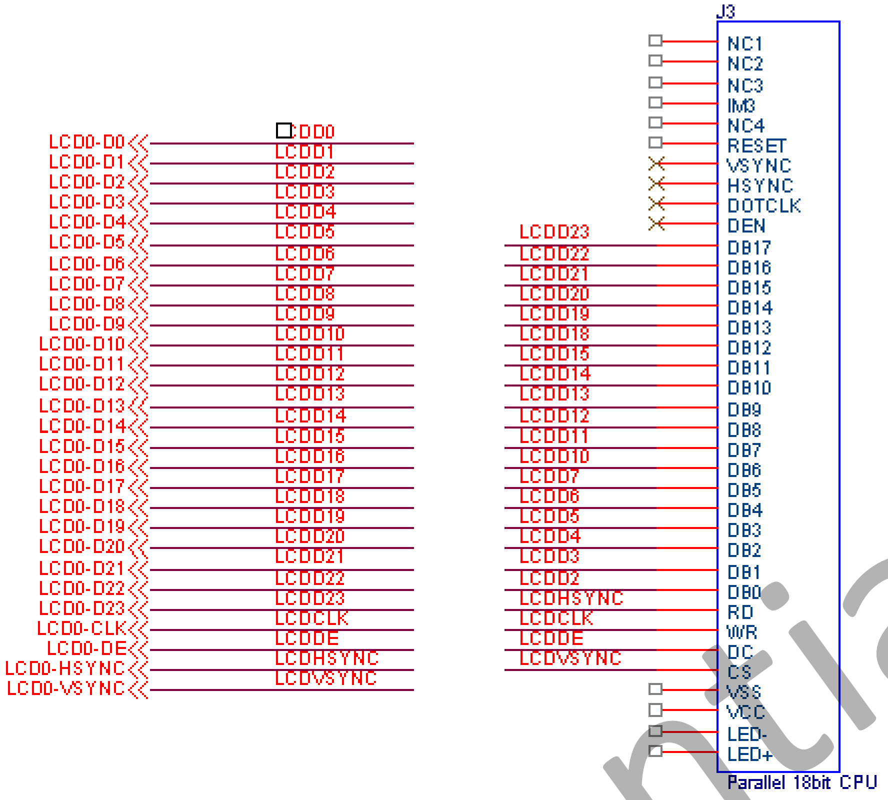


图 8 A83T 与 CPU Parallel 18bit 屏连接图

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 A83T 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          = 500
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>
```

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_orgin);

}



static void lcd_panel_kgm281e0_orgin()

{

    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);
```

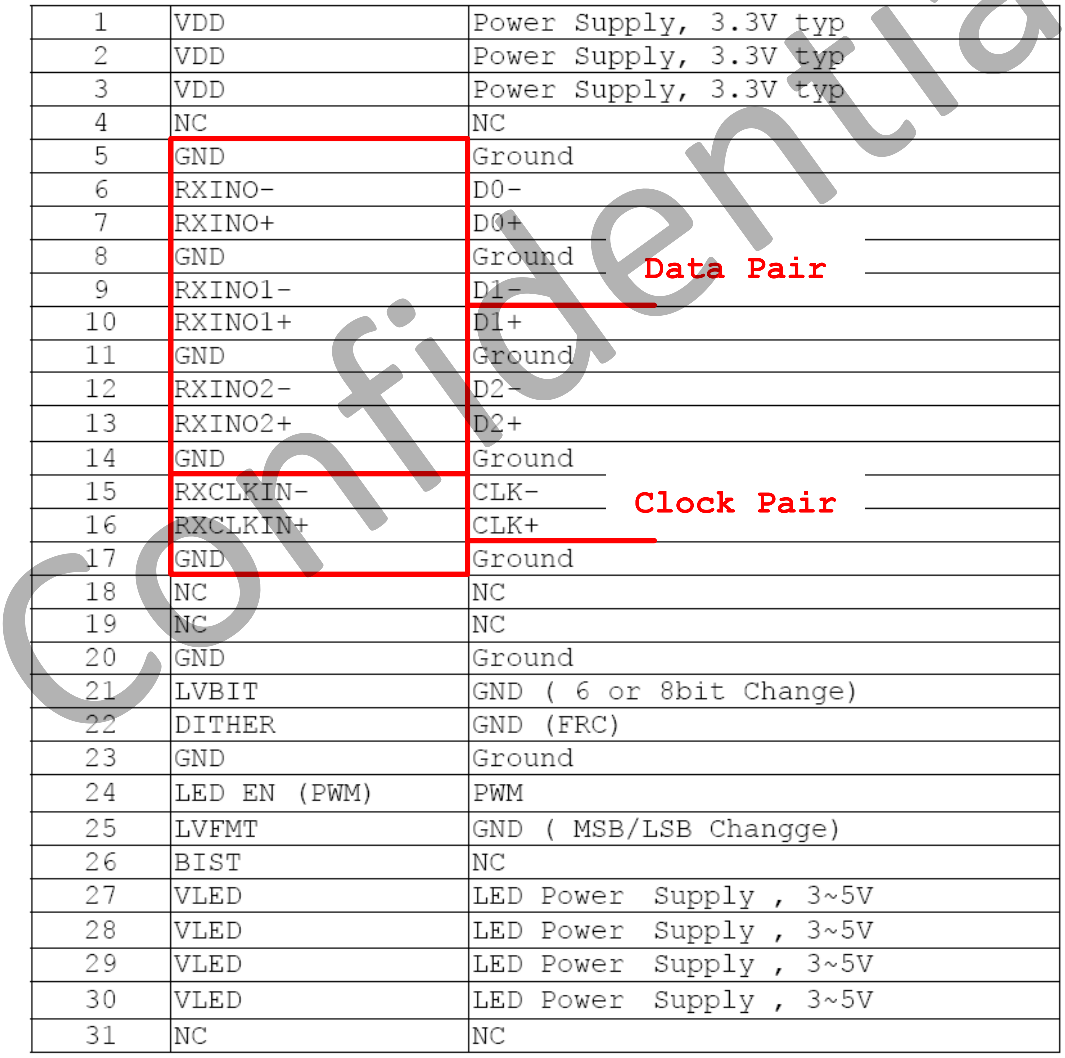
```
    LCD_CPU_WR(sel,0x0007, 0x0173);     //

}
```

4.3. A83T 与 LVDS Single Link 屏

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 9 是一个典型 LVDS Single Link 屏的模组规格书的引脚定义。A83T 与该 LCD 屏的引脚连接可参考图 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-
10	RXINO1+	D1+
11	GND	Ground
12	RXINO2-	D2-
13	RXINO2+	D2+
14	GND	Ground
15	RXCLKIN-	CLK-
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 Changge)
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 9 LVDS Single Link 屏引脚定义

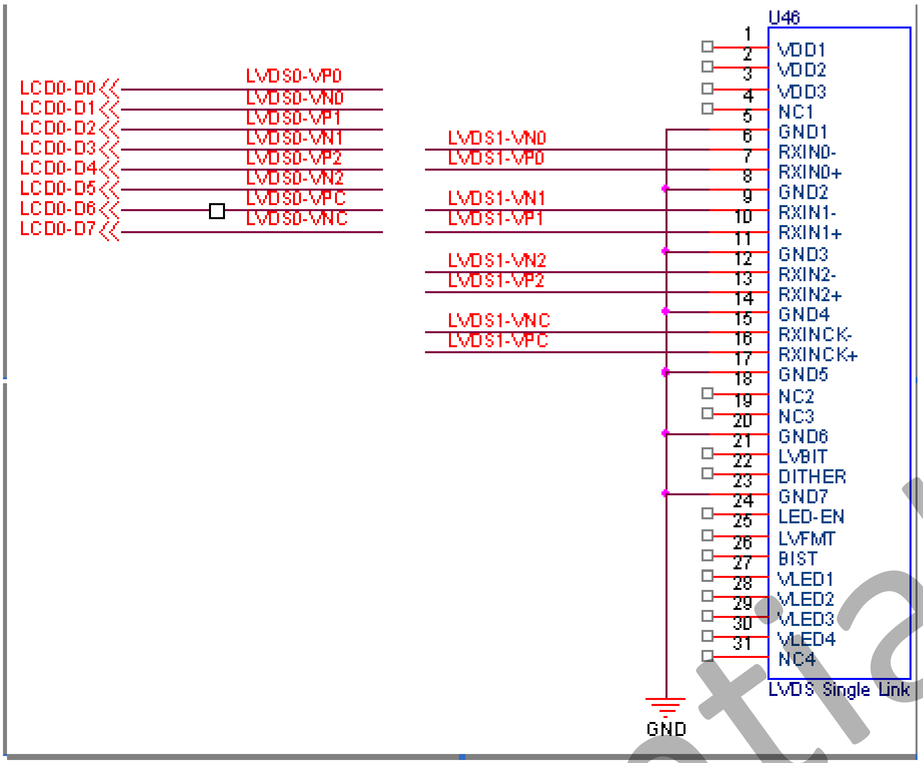


图 10 A83T 与 CPU Single Link 屏连接图

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 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 A83T 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				SYNBOL	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 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

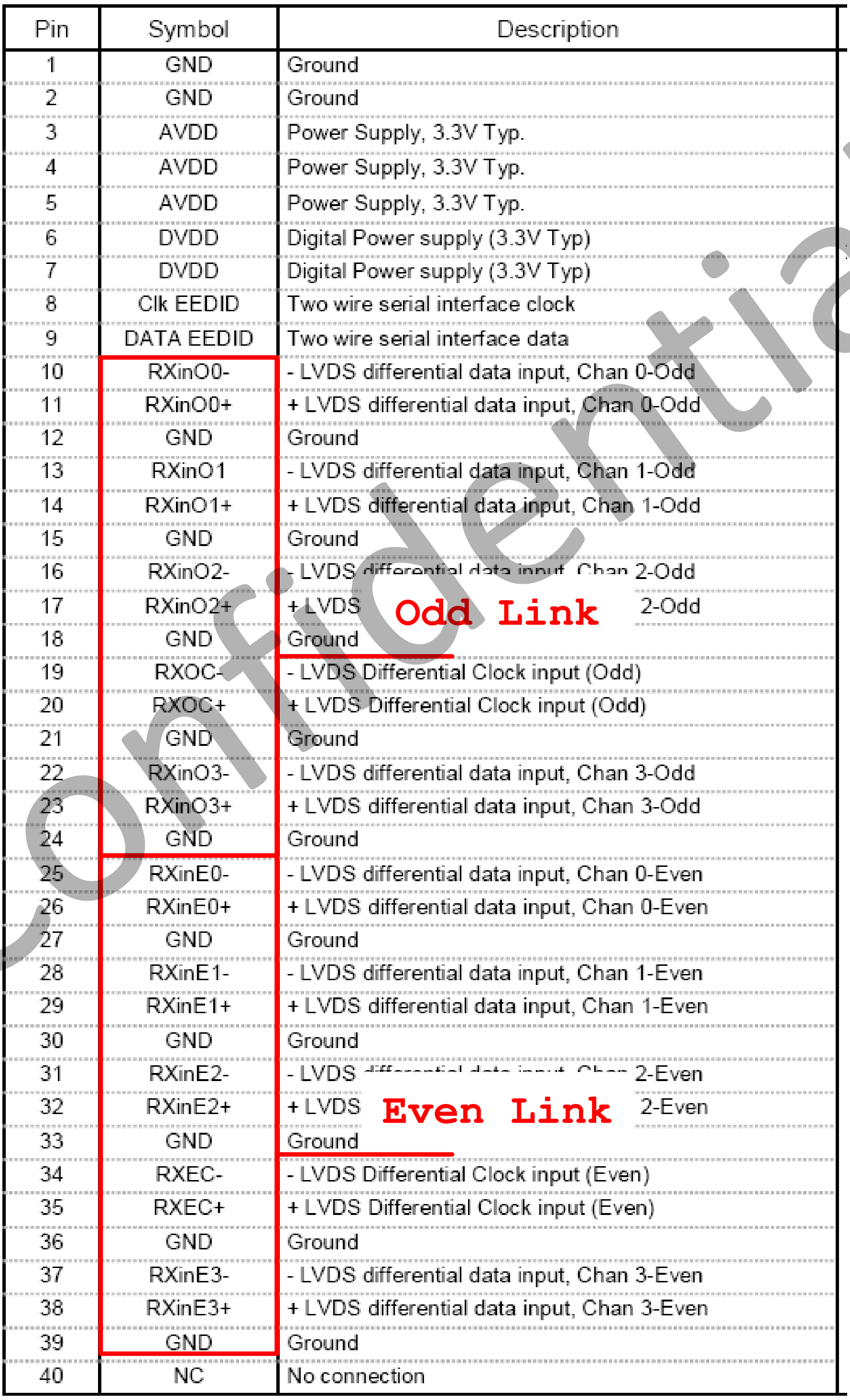
图 11 LVDS Single Link 屏参数

```
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        = 1288
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>
```

4.4. A83T 与 LVDS Dual Link 屏

schematic

如图 12 是一个典型 LVDS Single Link 屏的模组规格书的引脚定义。A83T 与该 LCD 屏的引脚连接可参考图 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 [illegible] 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 [illegible] 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 12 LVDS Dual Link 屏引脚定义

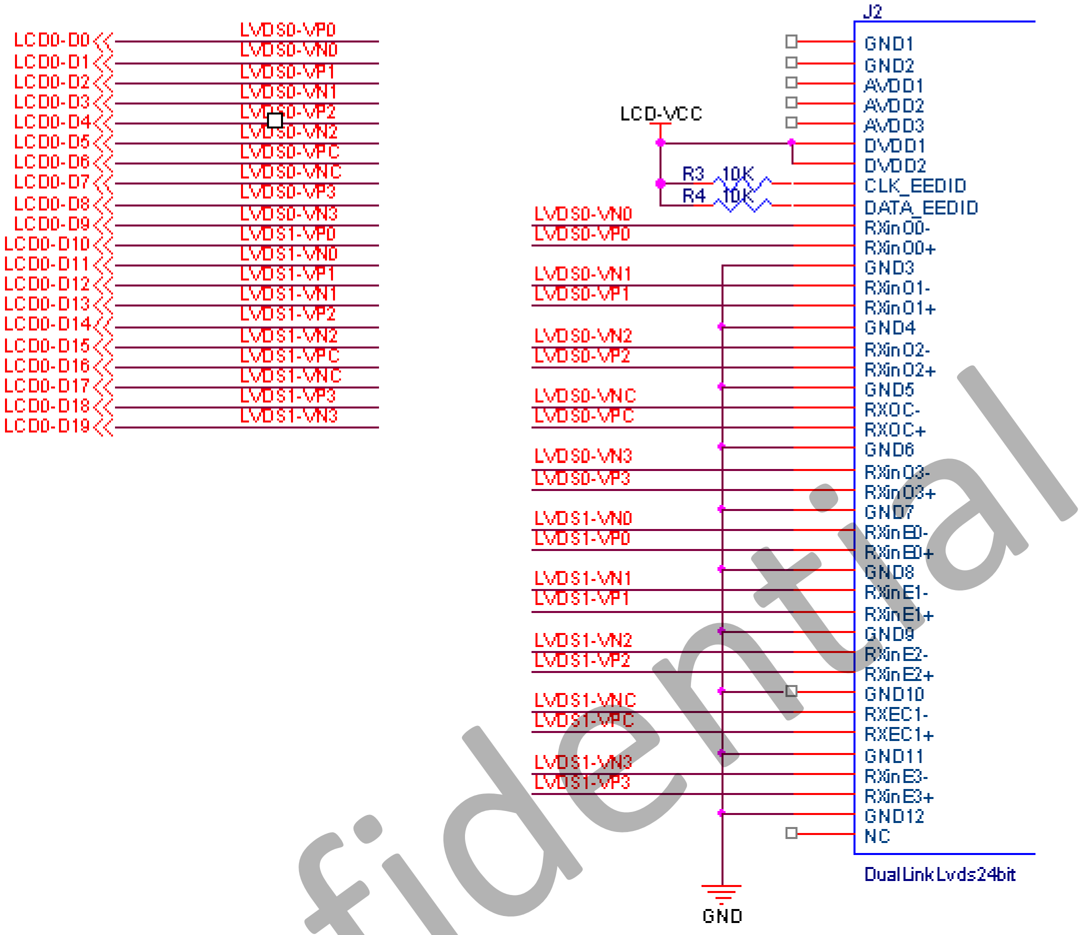


图 13 A83T 与 LVDS Dual Link 屏连接图

sys_config.fex

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

该 LCD 屏参数如图 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 A83T 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。

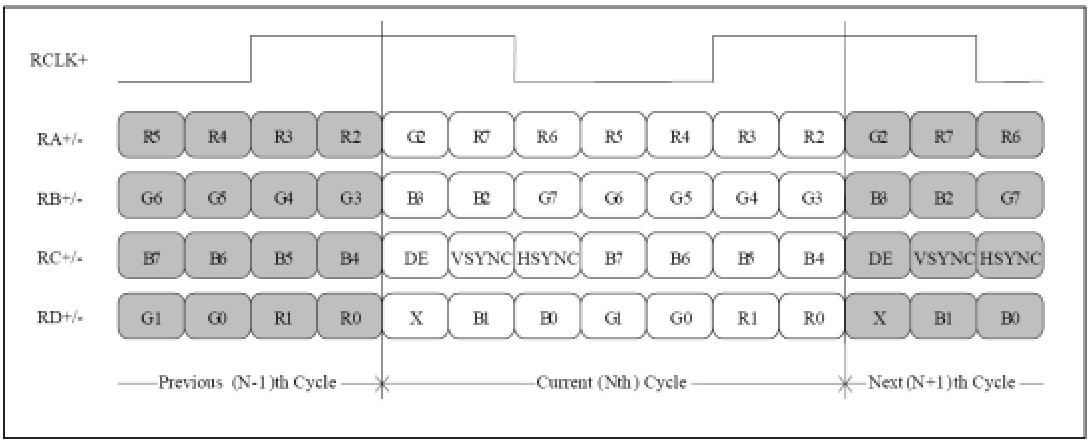
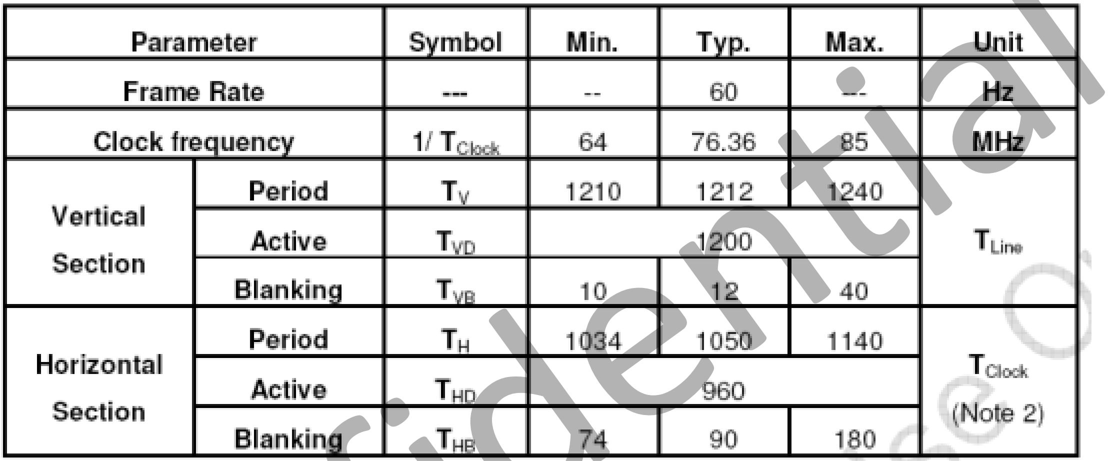

RCLK+
RA+/-
R5 R4 R3 R2 G2 R7 R6 R5 R4 R3 R2 G2 R7 R6
RB+/-
G6 G5 G4 G3 B3 B2 G7 G6 G5 G4 G3 B3 B2 G7
RC+/-
B7 B6 B5 B4 DE VSYNC HSYNC B7 B6 B5 B4 DE VSYNC HSYNC
RD+/-
G1 G0 R1 R0 X B1 B0 G1 G0 R1 R0 X B1 B0
Previous (N-1)th Cycle
Current (Nth) Cycle
Next (N+1)th Cycle

图 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 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
```

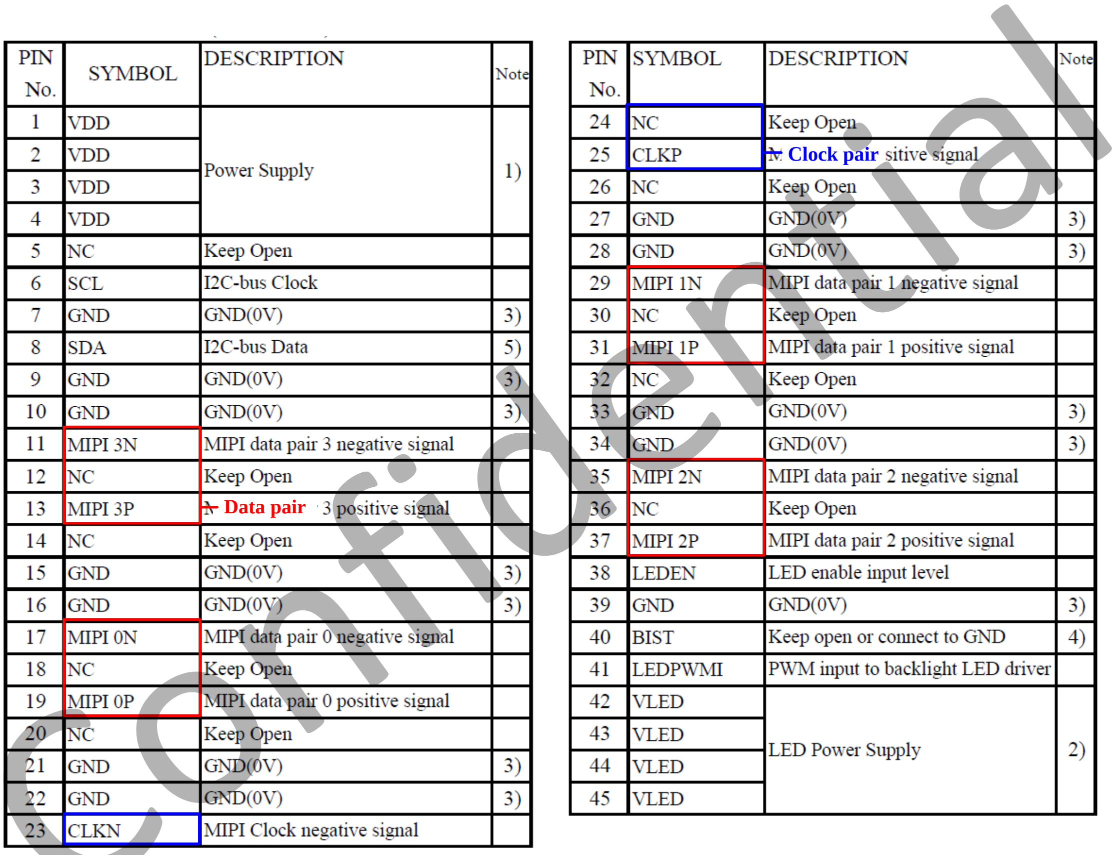
```
lcd_hbp         = 20

lcd_hspw        = 10

lcd_vt           = 1212

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>
```

4.5. A83T 与 DSI Video mode 屏

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 16 是一个典型 DSI Video Mode 屏的模组规格书的引脚定义。A83T 与该 LCD 屏的引脚连接可参考图 17。



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

图 16 MIPI DSI Video Mode 引脚定义

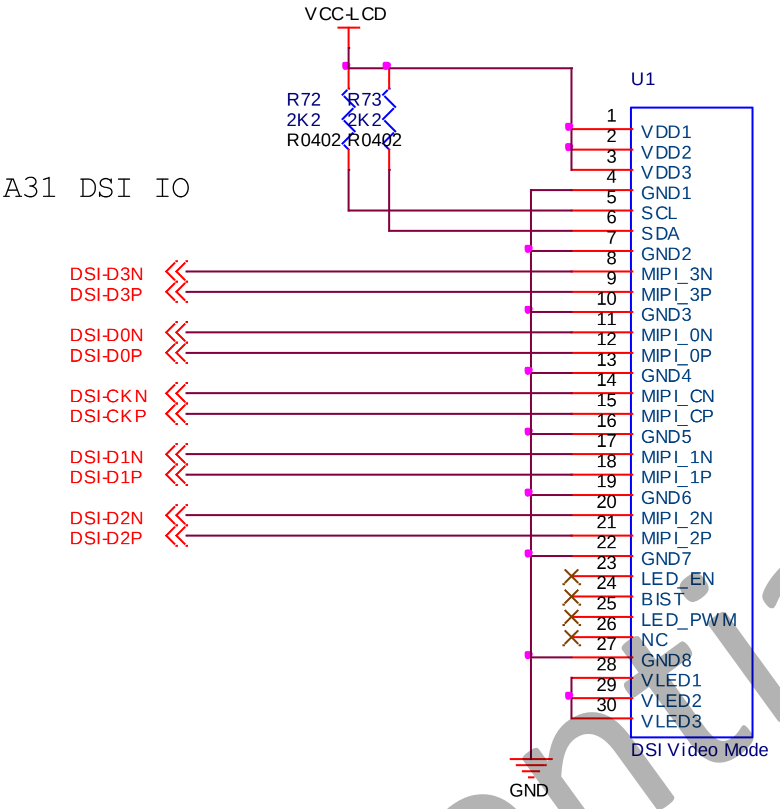


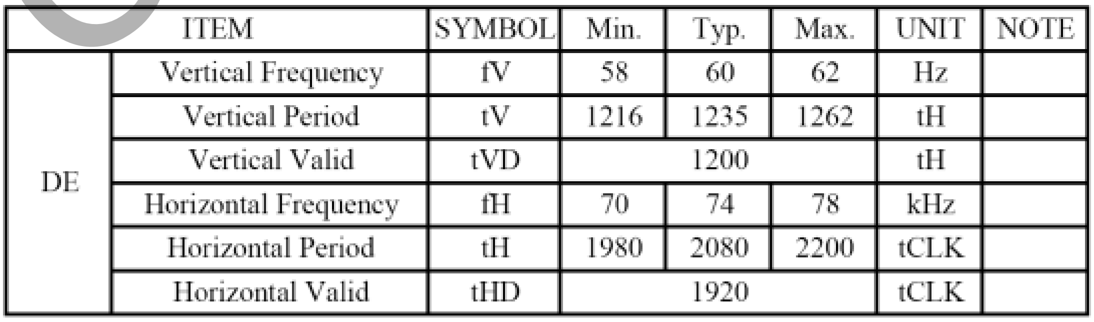
图 17 A83T 与 MIPI DSI Video Mode 连接图

sys_config.fex

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 18 所示。时序参数配置与 LVDS 类似。注意 DSI 屏中，须满足 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 18 MIPI DSI Video Mode 屏参数

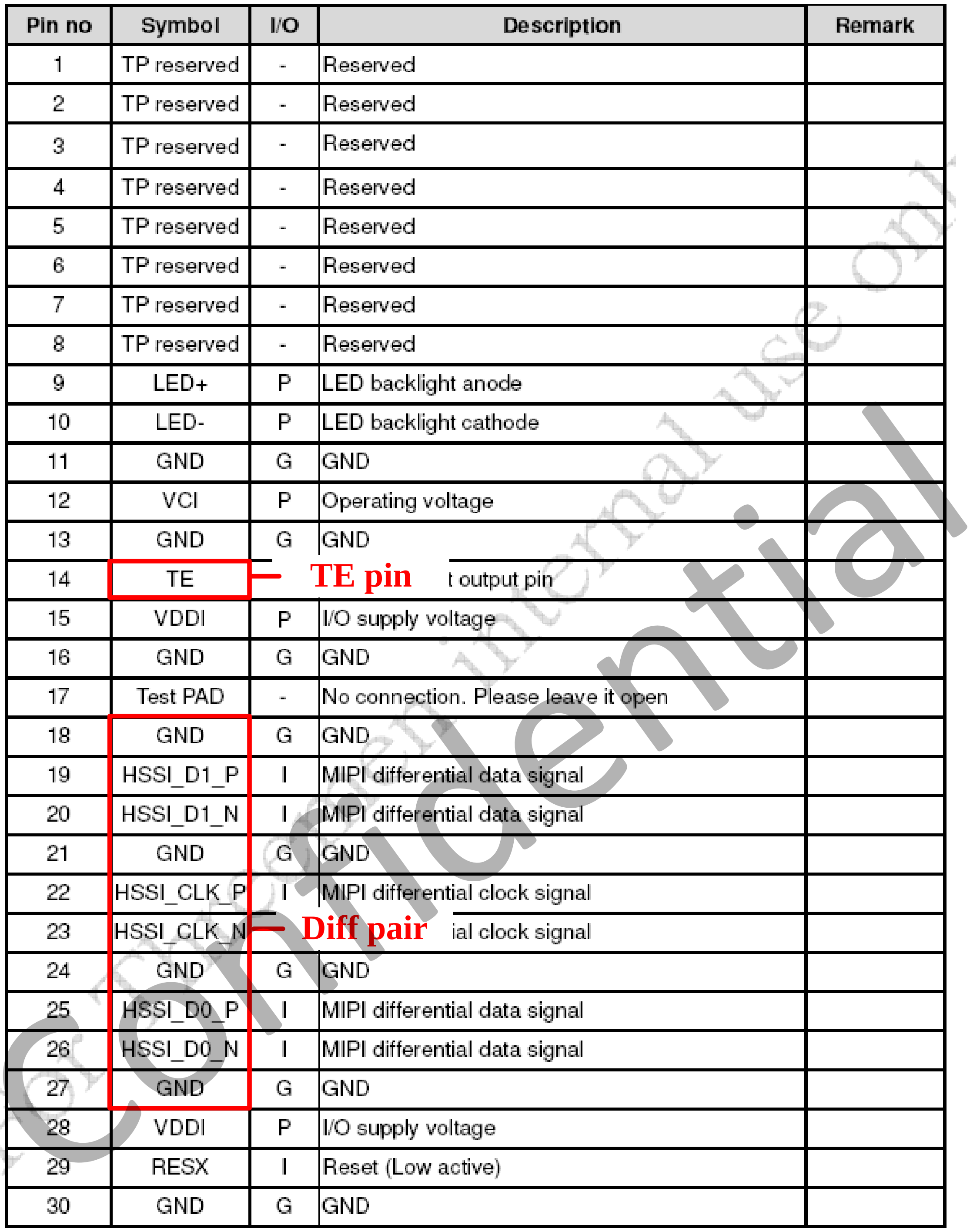
```
lcd_if              = 4
```

```
lcd_dsi_if          = 0

lcd_dsi_lane       = 4

lcd_dsi_format      = 0

lcd_dsi_eotp       = 0



lcd_dclk_freq       = 154

lcd_x             = 1920

lcd_y             = 1200

lcd_ht             = 2080

lcd_hbp             = 80

lcd_hspw            = 20

lcd_vt             = 1235

lcd_vbp             = 20

lcd_vspw            = 10
```

4.6. A83T 与 DSI Command mode 屏

schematic

如图 19 是一个典型 DSI Command Mode 屏的模组规格书的引脚定义。A83T 与该 LCD 屏的引脚连接可参考图 20。



Pin no	Symbol	I/O	Description	Remark
1	TP reserved	-	Reserved	
2	TP reserved	-	Reserved	
3	TP reserved	-	Reserved	
4	TP reserved	-	Reserved	
5	TP reserved	-	Reserved	
6	TP reserved	-	Reserved	
7	TP reserved	-	Reserved	
8	TP reserved	-	Reserved	
9	LED+	P	LED backlight anode	
10	LED-	P	LED backlight cathode	
11	GND	G	GND	
12	VCI	P	Operating voltage	
13	GND	G	GND	
14	TE		t output pin	
15	VDDI	P	I/O supply voltage	
16	GND	G	GND	
17	Test PAD	-	No connection. Please leave it open	
18	GND	G	GND	
19	HSSI_D1_P	I	MIPI differential data signal	
20	HSSI_D1_N	I	MIPI differential data signal	
21	GND	G	GND	
22	HSSI_CLK_P	I	MIPI differential clock signal	
23	HSSI_CLK_N		ial clock signal	
24	GND	G	GND	
25	HSSI_D0_P	I	MIPI differential data signal	
26	HSSI_D0_N	I	MIPI differential data signal	
27	GND	G	GND	
28	VDDI	P	I/O supply voltage	
29	RESX	I	Reset (Low active)	
30	GND	G	GND	

图 19 MIPI DSI Video Mode 屏引脚定义

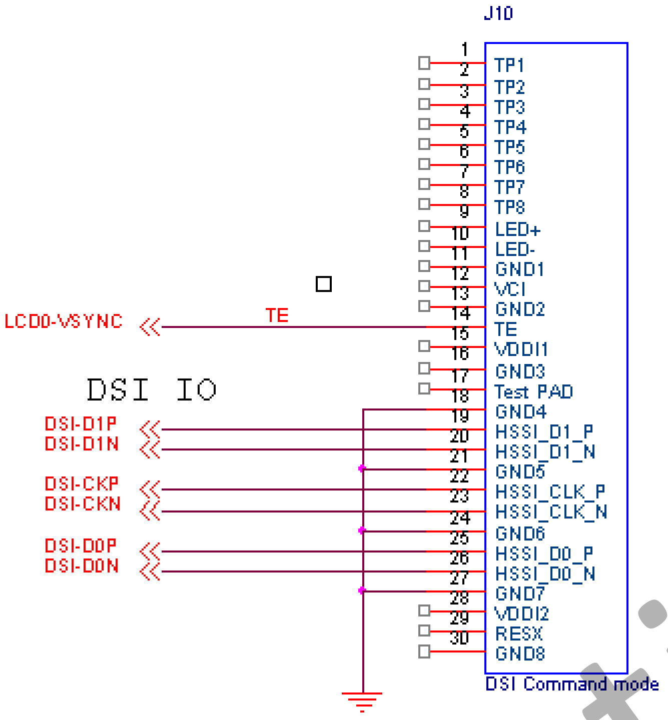


图 20 A83T 与 MIPI DSI Command Mode 屏连接图

sys_config.fex

该 LCD 屏，具有 TE 引脚，接口参数配置如下：lcd_if = 4，lcd_dsi_if=1，lcd_dsi_te=1。

DSI command mode 的 LCD 屏，一般只有 lcd_x，lcd_y。其时序参数配置可参考 CPU 屏，差异于 CPU 屏，要求 lcd_ht > lcd_x+80。该 LCD 屏配置如下。

```
lcd_if              = 4

lcd_dsi_if              = 1

lcd_dsi_lane            = 2

lcd_dsi_format          = 0

lcd_dsi_eotp            = 0

lcd_dsi_te              = 1



lcd_dclk_freq           = 36

lcd_x               = 480
```

```
lcd_y              = 800
lcd_ht             = 600
lcd_hbp            = 20
lcd_hspw           = 10
lcd_vt             = 1000
lcd_vbp            = 20
lcd_vspw           = 10
```

lcd0_panel_cfg.c

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

```
static void LCD_panel_init(__u32 sel)
{
    lcd_panel_a050vw02_init(sel);
}
static void lcd_panel_a050vw02_init(__u32 sel)
{
    __u8 const Enable_Manufacture_Command      =   0xf0;
    __u8 const Display_Option_Control          =   0xb1;
    __u8 const Inversion_Driving_Control       =   0xbc;

    dsi_dcs_wr_5para(sel,Enable_Manufacture_Command,0x55,0xaa,0x52,0x08,0x00);
    dsi_dcs_wr_2para(sel,Display_Option_Control,0xff,0x00);
    dsi_dcs_wr_3para(sel,Inversion_Driving_Control,0x05,0x05,0x05);
    dsi_dcs_wr_5para(sel,Enable_Manufacture_Command,0x55,0xaa,0x52,0x
08,0x01);

    LCD_delay_ms(120);

    dsi_dcs_wr_0para(sel,DSI_DCS_EXIT_SLEEP_MODE);

    LCD_delay_ms(120);

    dsi_dcs_wr_0para(sel,DSI_DCS_SET_DISPLAY_ON);



    dsi_dcs_wr_1para(sel,DSI_DCS_SET_TEAR_ON,0x00);

    dsi_dcs_wr_1para(sel,DSI_DCS_SET_PIXEL_FORMAT,0x77);

}
```

4.7. 电源和背光

LCD 供电，在 A83T 方案中，有两套背光控制，一是外部 LCD 自带，需要通过向 LCD 发送指令控制背光；而是使用内部 PWM 来控制。

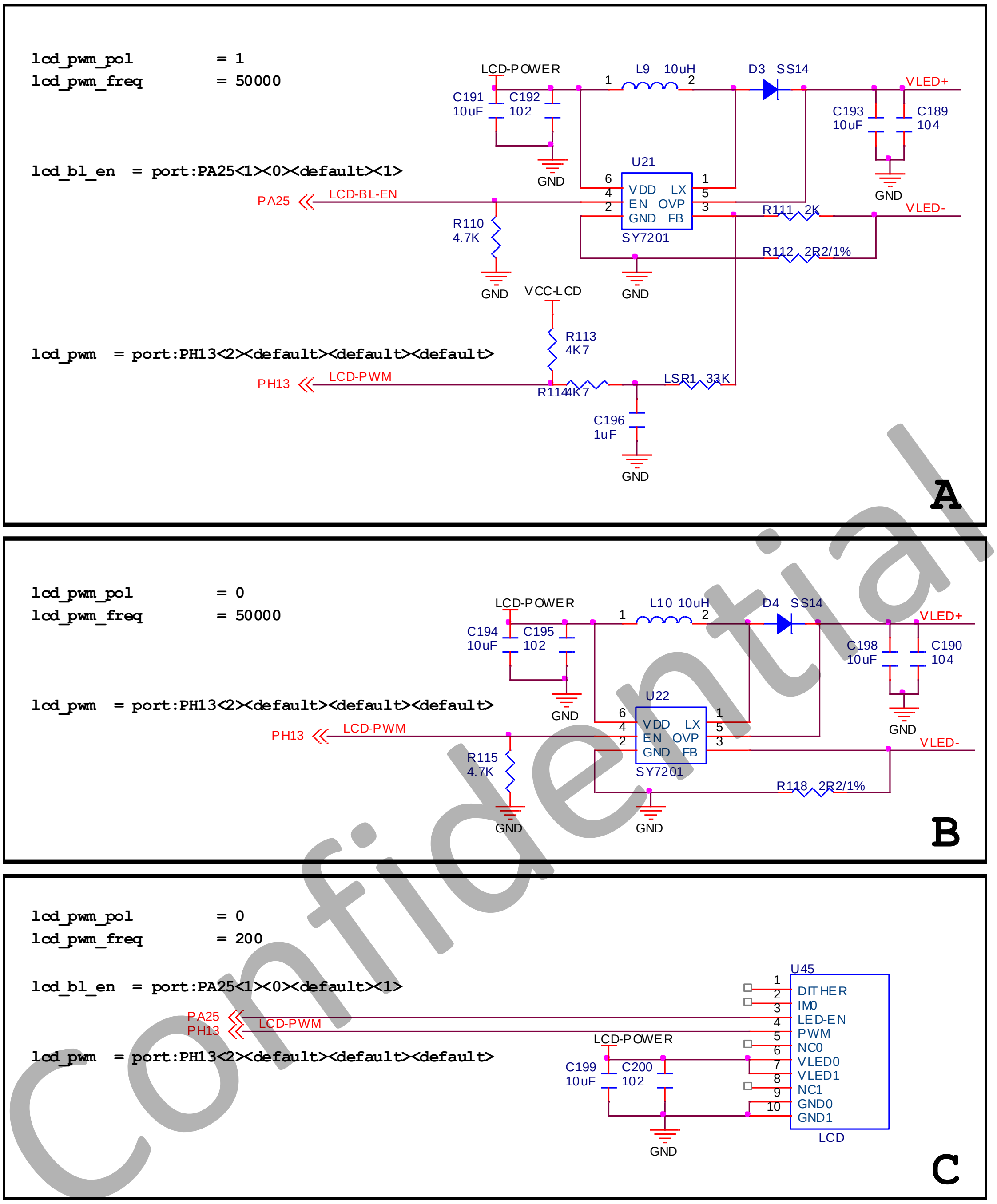
内部 PWM 控制 LCD 背光，常见有三种电路，如图

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。



```
lcd_pwm_pol        = 1
lcd_pwm_freq       = 50000

lcd_bl_en  = port:PA25<1><0><default><1>

lcd_pwm  = port:PH13<2><default><default><default>
```

```
lcd_pwm_pol        = 0
lcd_pwm_freq       = 50000

lcd_pwm  = port:PH13<2><default><default><default>
```

```
lcd_pwm_pol        = 0
lcd_pwm_freq       = 200

lcd_bl_en  = port:PA25<1><0><default><1>

lcd_pwm  = port:PH13<2><default><default><default>
```

图 21 常用 LCD 背光电路