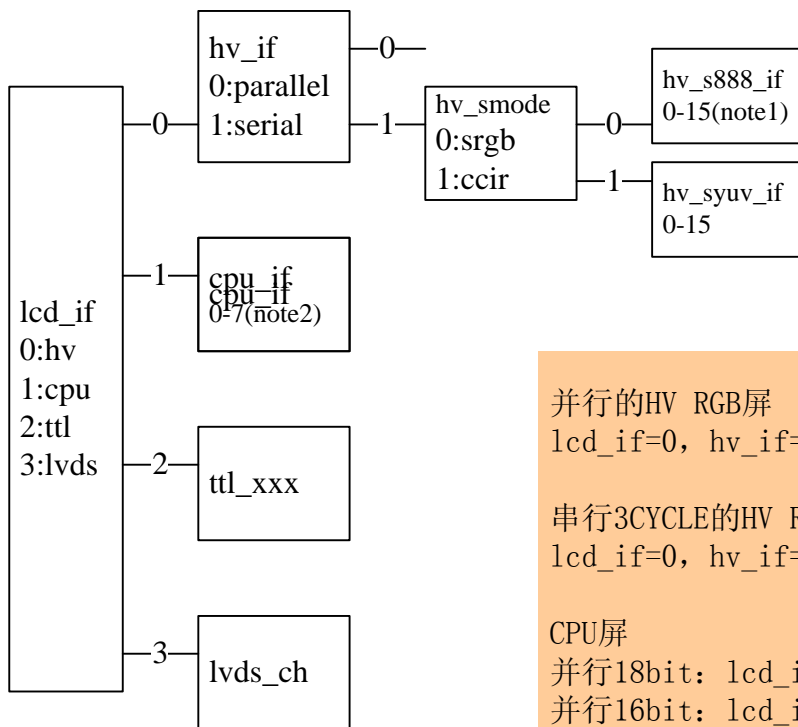


配屏事项

- 1、接口
- 2、时序
- 3、电源与背光
- 4、IO口配置
- 5、模组初始化



并行的HV RGB屏

lcd_if=0, hv_if=0

串行3CYCLE的HV RGB屏

lcd_if=0, hv_if=1, hv_smode=0

CPU屏

并行18bit: lcd_if=1, cpu_if=0

并行16bit: lcd_if=1, cpu_if=4

LVDS屏

单通道: lcd_if=3, lvds_ch=0

双通道: lcd_if=3, lvds_ch=1

Note1:

hv_srgb_if为串行的SRGB像素格式

高两位对应奇数行

00: R→G→B

01: B→R→G

10: G→B→R

11: R→G→B

低两位对应偶数行

00: R→G→B

01: B→R→G

10: G→B→R

11: R→G→B

常见参数说明

lcd_swap 颜色变换

0: 正常情况

1: RED和BLUE通路交换

lcd_frm 色深抖动开关

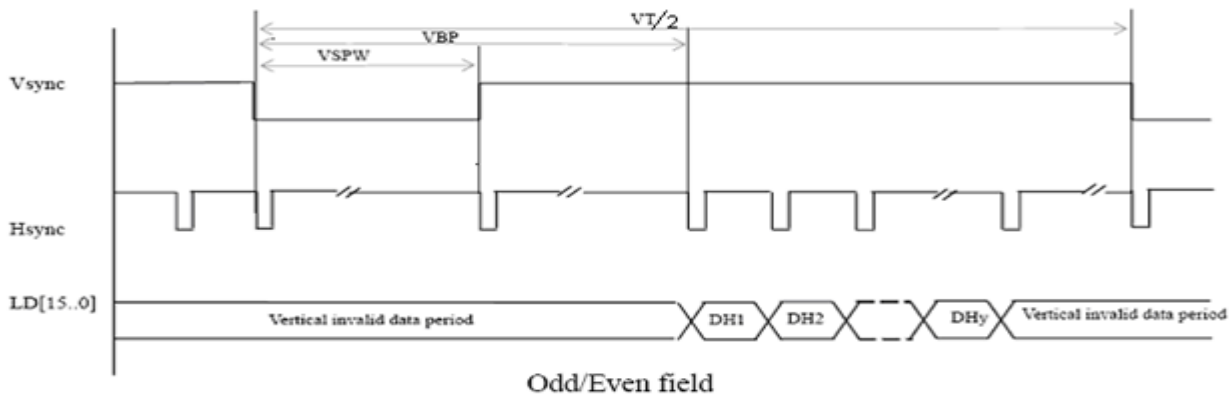
0: 关闭, 数据直通

1: 24位输入18位抖动输出

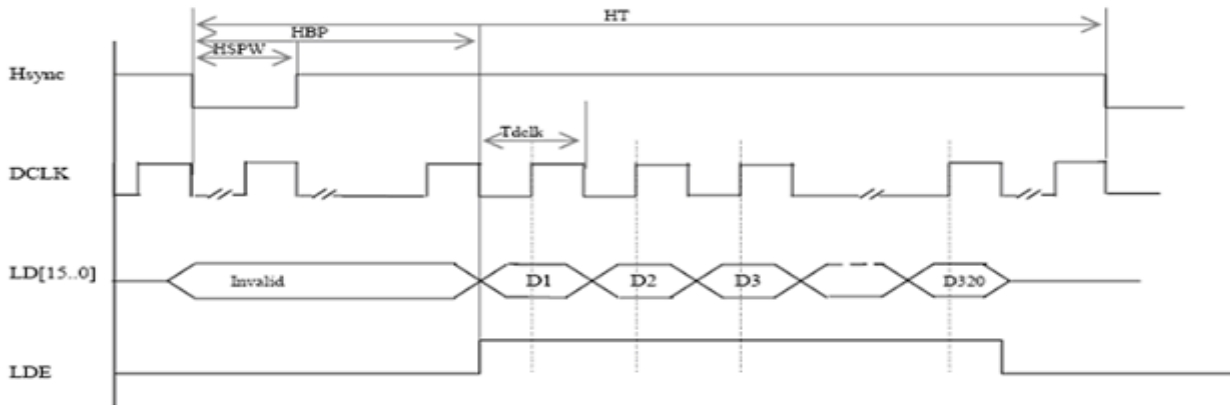
2: 24位输入16位抖动输出

1.屏接口IF

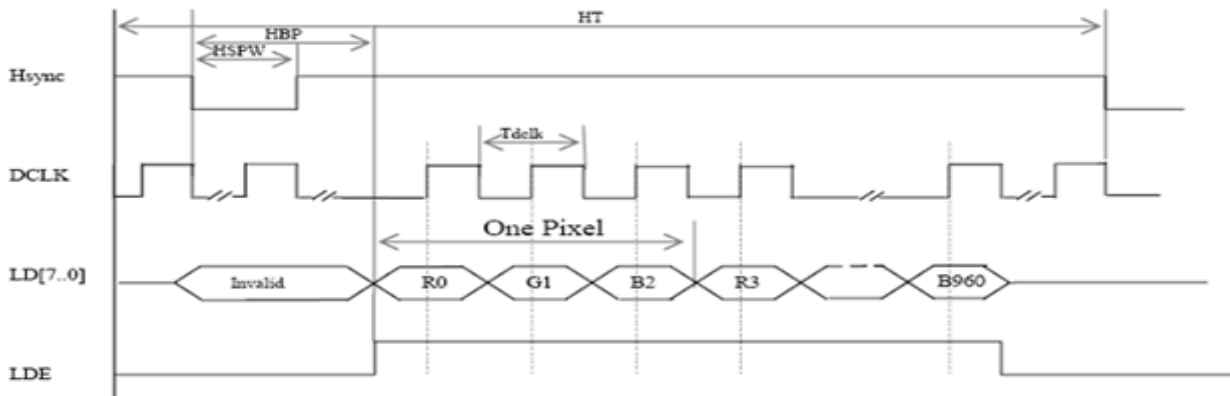
Verital Timing:



Parallel Mode Horizontal Timing:



Serial Mode Horizontal Timing:



时序参数说明（建议按此顺序配置）

x
y

hspw hspw>0
vspw vspw>0

hbp hbp>hspw
vbp vbp>vspw

ht para: ht>hbp+x+3
seri: ht>hbp×3+x+3
vt vt/2>vbp+y+2

dclk_freq
dclk_freq=ht×vt×30/10⁶

2.时序

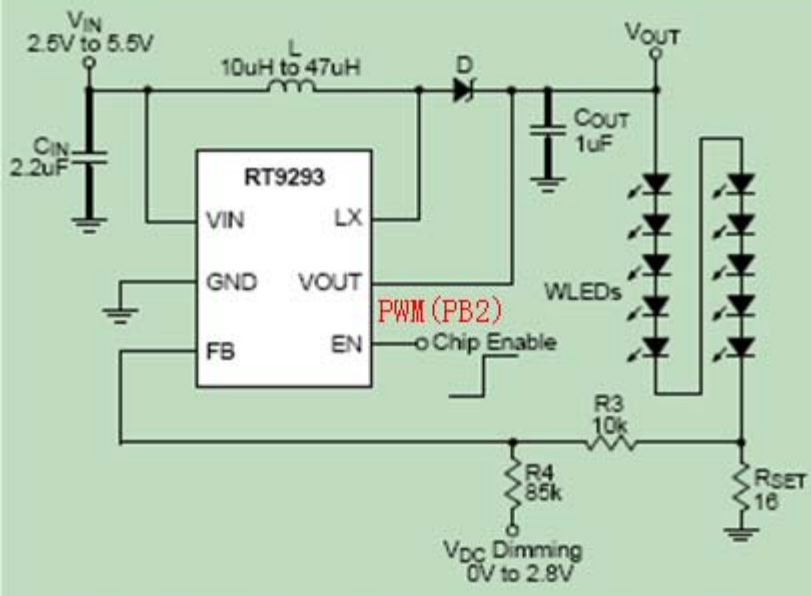


Figure 2. Dimming Control Using a DC Voltage

背光常见的两种电路，对应两种配置
 上边图特征是RT9293-EN连接到PWM (PB2)
 下边图特征是RT9293-EN连接到普通IO口

两种电路的参数配置如左边所示
 红色para1为连接对应的普通IO口
 红色para2为<0>时，低电平有效
 为<1>时，高电平有效

如若用到PMU的IO，请到lcd0_panel_cfg.c修改程序

```

lcd_bl_en_used = 0
lcd_bl_en =
lcd_power_used = 1
lcd_power = port:PH08<1><0><default><0>
lcd_pwm_used = 1
lcd_pwm = port:PB02<2><default><default><default>
  
```

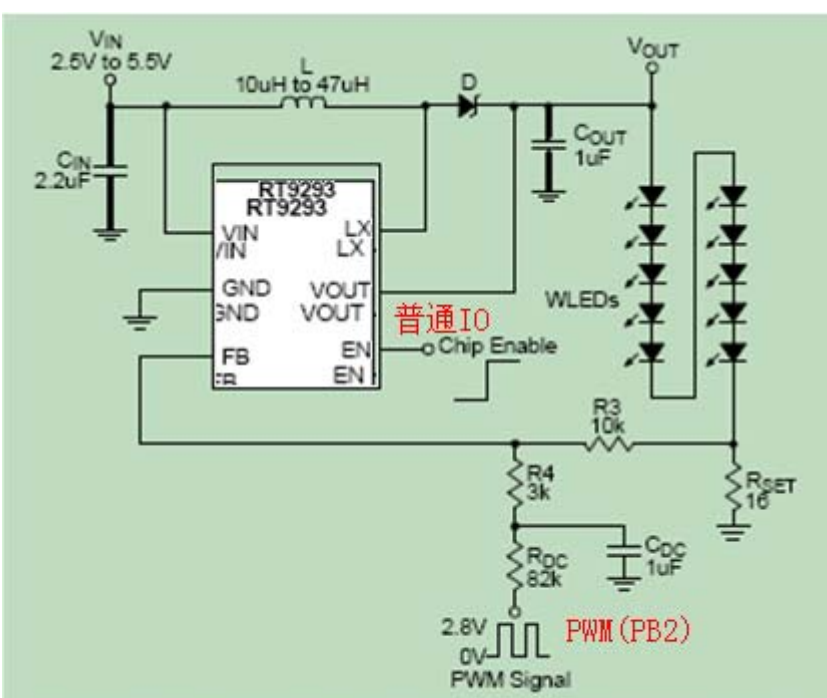


Figure 3. Dimming Control Using a Filtered PWM Signal

```

lcd_bl_en_used = 1
lcd_bl_en = port:PI10<1><0><default><0>
lcd_power_used = 1
lcd_power = port:PH08<1><0><default><0>
lcd_pwm_used = 1
lcd_pwm = port:PB02<2><default><default><default>
  
```

3.电源与背光

1、lcd_gpio_0-3为配屏用的IO申求，不需要的IO请留空，如
lcd_gpio_3

2、lcdd0-lcdd23, lcdclk, lcdde, lcdhsync, lcdvsync为lcd控制器输出的IO申求，不需要的LCD IO请用分号注释去，如lcdd0, lcdd1

```
lcd_gpio_0          = port:PI10<1><0><default><0>
lcd_gpio_1          = port:PI10<1><0><default><0>
lcd_gpio_2          = port:PI10<1><0><default><0>
lcd_gpio_3          =

;lcdd0              = port:PD00<2><0><default><default>
;lcdd1              = port:PD01<2><0><default><default>
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>
;lcdd8              = port:PD08<2><0><default><default>
;lcdd9              = port:PD09<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>
;lcdd17             = port:PD17<2><0><default><default>
lcdd18              = port:PD18<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>
```

4、IO口配置

常见的模组的初始化有两种

1、用lcd_gpio_0-3模拟串行时序实现，多见于HV屏；

函数在lcd_panel_cfg.c文件中自由实现；

2、用lcdc的io总线读写操作实现，见于CPU屏；

BSP提供3个接口函数；

```
void LCD_CPU_WR_INDEX(__u32 index); //ca pin is low
```

```
void LCD_CPU_WR_DATA(__u32 data); //ca pin is high
```

```
void LCD_CPU_WR(__u32 index, __u32 data); //combine 2func above
```

5、模组初始化