

AM335x Linux 下 LCD 屏幕的配置

-- Steven Liu

最近是收到了不少人提出的 LCD 屏幕显示问题，多数问题都是出现在 TFT 屏幕的时序配置上面，正好趁着这个机会，回忆了一下之前的 LCD 屏幕的配置流程，这里主要分析一下 Linux 的 kernel driver 对 LCD 的配置流程和配置方法，以 SDK6.0 为例，给大家做个参考。

关于 LCD 配置的一些基础信息：

首先谈谈关于 AM335x 的 LCD 控制器。

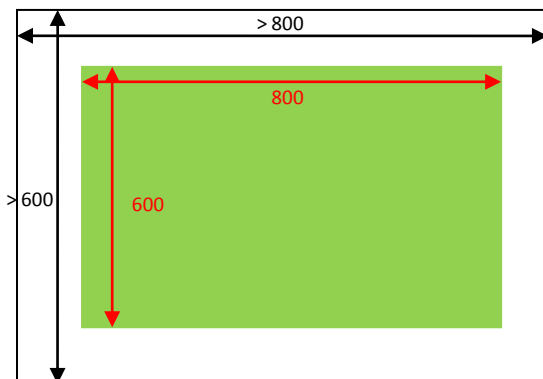
1. 关于分辨率：阅读过 datasheet 的人都知道，我们 AM335x 目前的 LCDC 能够支持的最大屏幕分辨率为 2048*2048，但不代表任意小于该分辨率的设置都可以支持。

限制：行像素必须要是 16 的整数倍。

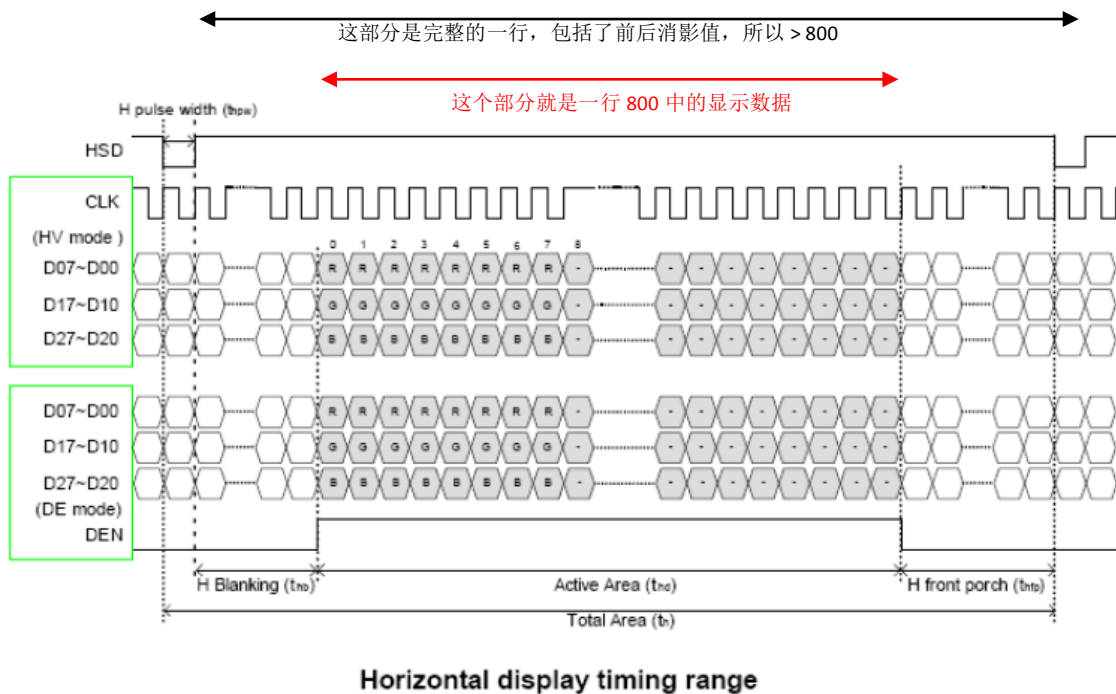
可以参考 AM335x 的 TRM 手册，13.5.1.11 章节关于 RASTER_TIMING_0 寄存器的解释，[9-4]pp11sb 设置了行像素的第 4 位到第 9 位，3-0 这四位是无法设置的，所以配置出来的就是 16 的整数倍。列像素则没有这个限制，RASTER_TIMING_1 寄存器的[9-0]可完整的设置列像素的低 9 位。因此，当你发现你配置的屏幕出现了一条缝没有填满，或者是一些像素点被整个覆盖的时候，就要查一下你的行像素是否是 16 的整数倍。

案例：之前有人反映过 1366*768 这个像素显示出来的就是多了一条缝，因为 1366 不能被 16 整除，只能配成 1360*768，或者 1376*768。如果遇到这种情况，要不就换屏幕分辨率；要不就是配置成比原始的分辨率大，然后再在应用层上，通过 QT 的设置等方式来显示显示的区域。PS：个人会推荐前者，。

2. 关于像素时钟：使用时一定要注意你在配置一个分辨率时，像素时钟 pixel clock 是否能满足你的 LCD 屏幕的要求。简而言之，你的像素时钟，一定要能 cover 住你当前显示所需值。提到这个是因为之前有人和我说我们的像素时钟配置很受限，其实不然，他配置很灵活，要注意选对时钟源，我之前写过一篇 [《AM335x 关于 LCD 屏幕的配置》](#) 里面讲述的了如果配置出任意的频率，可以参考这个，这里不再赘述。
3. 关于屏幕的显示：简要的示意图如下，如果说你要显示的是一个 800*600 的显示，LCD 的控制器实际上是刷出了一个比 800*600 的分辨率更大的空间（大出来的部分就是前后消影值）。各个厂家的屏幕这里是不一样的。



类似的，你能在 LCD 屏幕上面看到的时序如下所示



有了上述的概念后，下面我们就开始配置 LCD 屏幕的流程。这里以 Linux 的 SDK 6.0 为例，带大家来看一下我们是如何配置的一个 LCD 屏幕的。使用的设备是 GPEVM 开发板。

首先，在 EZSDK_06_00_00_00/board-support/linux-3.2.0-psp04.06.00.11/arch/arm/mach-omap2/board-am335xevm.c 文件中，以下几个函数是必须要了解的。

```
/* General Purpose EVM */
static struct evm_dev_cfg gen_purp_evm_dev_cfg[] = {
    {am335x_rtc_init, DEV_ON_BASEBOARD, PROFILE_ALL},
    {clkout2_enable, DEV_ON_BASEBOARD, PROFILE_ALL},
    {enable_ecap0, DEV_ON_DGHTR_BRD, (PROFILE_0 | PROFILE_1 |
        PROFILE_2 | PROFILE_7)},
    {lcdc_init, DEV_ON_DGHTR_BRD, (PROFILE_0 | PROFILE_1 |
        PROFILE_2 | PROFILE_7)},
    {mfd_tscadc_init, DEV_ON_DGHTR_BRD, (PROFILE_0 | PROFILE_1 |
        PROFILE_2 | PROFILE_7)},
}
```

这里就是配置 GPEVM 板时调用的 LCDC 的初始化函数。

```
static void lcdc_init(int evm_id, int profile)
{
    struct da8xx_lcdc_platform_data *lcdc_pdata;
    setup_pin_mux(lcdc_pin_mux);

    if (conf_disp_pll(300000000)) {
        pr_info("Failed configure display PLL, not attempting to\n");
        register LCDC\n");
        return;
    }
    switch (evm_id) {
        case GEN_PURP_EVM:
        case GEN_PURP_DDR3_EVM:
            lcdc_pdata = &TFC_S9700RTWV35TR_01B_pdata;
            break;
        case EVM_SK:
            lcdc_pdata = &NHD_480272MF_ATXI_pdata;
            break;
        default:
            pr_err("LCDC not supported on this evm (%d)\n", evm_id);
            return;
    }
}
```

这里首先配置了 LCD 的管脚 PINMUX，然后通过对板子的型进行判断，从而决定配置。这里需注意处理逻辑，确保配置调用的类型是你预期要设置的。

```

/* Module pin mux for LCD */
static struct pinmux_config lcdc_pin_mux[] = {
    {"lcd_data0.lcd_data0",      OMAP_MUX_MODE0 | AM33XX_PIN_OUTPUT
                                     | AM33XX_PULL_DISA},
    {"lcd_data1.lcd_data1",      OMAP_MUX_MODE0 | AM33XX_PIN_OUTPUT

```

PINMUX 配置，一定确保要调用

```

struct da8xx_lcdc_platform_data TFC_S9700RTWV35TR_01B_pdata = {
    .manu_name      = "ThreeFive",
    .controller_data = &lcd_cfg,
    .type           = "TFC_S9700RTWV35TR_01B",
};

```

这里就是对屏幕的配置，Timing 参数在相应的结构体中

```

static struct lcd_ctrl_config lcd_cfg = {
    &disp_panel,
    .ac_bias      = 255,
    .ac_bias_intrpt = 0,
    .dma_burst_sz = 16,
    .bpp          = 32,
    .fdd          = 0x80,
    .tft_alt_mode = 0,
    .stn_565_mode = 0,
    .mono_8bit_mode = 0,
    .invert_line_clock = 1,
    .invert_frm_clock = 1,
    .sync_edge      = 0,
    .sync_ctrl      = 1,
    .raster_order   = 0,
};

```

LCD 的一些基本配置信息，对应到 RASTER_CTRL 寄存器。

下面，也是最重要的修改的 Timing 配置参数：

EZSDK_06_00_00_00/board-support/linux-3.2.0-psp04.06.00.11/drivers/video/da8xx-fb.c

```

static struct da8xx_panel known_lcd_panels[] = {

```

在该结构体中

```

/* ThreeFive S9700RTWV35TR */
[2] = {
    .name = "TFC_S9700RTWV35TR_01B",
    .width = 800,
    .height = 480,
    .hfp = 39,
    .hbp = 39,
    .hsw = 47,
    .vfp = 13,
    .vbp = 29,
    .vsw = 2,
    .pxl_clk = 30000000,
    .invert_pxl_clk = 0,
},

```

这里就是对于该屏幕配置的结构体，其中的时序参数的含义可以参考 AM335x 的 TRM 手册 13.5.1 章节的 RASTER_TIMING_1~3；实际的配置值是要根据使用的屏幕手册来进行的。这里是基本上每个屏幕都要检查配置的地方。

那么下面让我们来看看这些参数的配置是怎样得来的。

首先，你要确保你拿到你自己屏幕的完整的数据手册，以下面的手册作为一个样例。如果你发现其中的参数缺失了，那么联系你的屏幕供应商，或者是自行百度谷歌，一定要拿到这个手册，才可以进行相关的配置。否则，自己慢慢试的话，只能祝你好运了。

GPEVM 板的屏幕手册：

<http://www.elinux.org/images/d/dc/TFC-S9700RTWV35TR-01B.pdf>

在屏幕手册中，一定可以在 AC Characteristics 章节找到当前屏幕的 Timing 信息：

5.3. AC CHARACTERISTICS

Item Sy	mbol	Min.	Typ.	Max.	Unit	Note
DCLK cycle	time	25	-	-	ns	
DCLK frequency	fclk	-	30	40	MHz	pxl_clk: 像素时钟
DCLK pulse duty	Tcwh	40	50	60	%	
VSD setup time	Tvst	8	-	-	ns	
VSD hold time	Tvhd	8	-	-	ns	
HSD setup time	Thst	8	-	-	ns	
HSD hold time	Thhd	8	-	-	ns	
Data setup time	Tdsu	8	-	-	ns	
Data hold time	Tdhd	8	-	-	ns	
DE setup time	Tesu	8	-	-	ns	
DE hold time	Tehd	8	-	-	ns	
Horizontal display area	thd	-	800	-	Tcph	width: 行宽
HSD period time	th	-	928	-	Tcph	
HSD pulse width	thpw	1	48	-	Tcph	hsw * = 48 -1
HSD back porch	thb	-	40	-	Tcph	hbp * = 40 -1
HSD front porch	thfp	-	40	-	Tcph	hfp * = 40 -1
Vertical display area	tvd	-	480	-	th	height: 列宽 (高度)
VSD period time	tv	-	525	-	th	
VSD pulse width	tpw	-	3	-	th	vsw * = 3 -1
VSD back porch	tvb	-	29	-	th	vbp = 29
VSD front porch	tvfp	-	13	-	th	vfp = 13

上述标红的部分就是我们当前所设置出来的配置。

```
/* ThreeFive S9700RTWV35TR */
[2] = {
    .name = "TFC_S9700RTWV35TR_01B",
    .width = 800,
    .height = 480,
    .hfp = 39,
    .hbp = 39,
    .hsw = 47,
    .vfp = 13,
    .vbp = 29,
    .vsw = 2,
    .pxl_clk = 30000000,
    .invert_pxl_clk = 0,
},
```

这里，需要特别注意的是，上述我标星号*的地方都做了一个减一的操作，其他地方都是直接等于原值，这是为什么，大家看看AM335x 的 TRM 手册上的 Timing 寄存器就知道了。

在 AM335x 的 TRM 手册 13.5.1 章节的 RASTER_TIMING_1~3 寄存器描述中：

由于寄存器中 hfp, hbp, hsw, vsw 这四个寄存器的值，会自加 1，所以在配置的时候，一定要注意把屏幕手册中的值自减 1，这样你配置出来的参数才是当前屏幕手册中的参数。否则，你配出来的参数值就和你的屏幕手册要求的 Timing 数值不一致。至于时序不对的后果，想必不用我多说。看 RP 吧，据说也有可以正常显示的（感觉略不靠谱），也有颜色显示出问题失真的。总之，

尽量按照标准的配置流程来，若是出现了问题，就按照以上的流程来重新核对配置一遍。确保配置的正确才是王道。

Table 13-24. RASTER_TIMING_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	hbp	R/W	0h	Horizontal Back Porch Lowbits Bits 7:0 of the horizontal back porch field Encoded value (from 1-1024) used to specify the number of pixel clock periods to add to the beginning of a line transmission before the first set of pixels is output to the display (programmed value plus 1) Note that pixel clock is held in its inactive state during the beginning of the line wait period in passive display mode, and is permitted to transition in active display mode
23-16	hfp	R/W	0h	Horizontal Front Porch Lowbits Encoded value (from 1-1024) used to specify the number of pixel clock periods to add to the end of a line transmission before line clock is asserted (programmed value plus 1) Note that pixel clock is held in its inactive state during the end of line wait period in passive display mode, and is permitted to transition in active display mode
15-10	hsw	R/W	0h	Horizontal Sync Pulse Width Lowbits Bits 5:0 of the horizontal sync pulse width field Encoded value (from 1-1024) used to specify the number of pixel clock periods to pulse the line clock at the end of each line (programmed value plus 1) Note that pixel clock is held in its inactive state during the generation of line clock in passive display mode, and is permitted to transition in active display mode
9-4	pplsb	R/W	0h	Pixels-per-line LSB[9:4] Encoded LSB value (from 1-1024) used to specify the number of pixels contained within each line on the LCD display (programmed to value minus one) $PPL = 11'b\{pplmsb, ppls b, 4'b1111\} + 1$ ex: $pplmsb=1'b1$, $pppls b=6'0001000$ $PPL = 11'b\{1, 000100, 1111\} + 1 = 1104d$ (decimal) pixels per line In other words, $PPL = 16 * (\{pplmsb, ppls b\} + 1)$
3	pplmsb	R/W	0h	Pixels-per-line MSB[10] Needed in order to support up to 2048 ppl
2-0	Reserved	R	0h	

Table 13-25. RASTER_TIMING_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	vbp	R/W	0h	Vertical Back Porch Value (from 0-255) use to specify the number of line clock periods to add to the beginning of a frame before the first set of pixels is output to the display Note that line clock transitions during the insertion of the extra line clock periods
23-16	vfp	R/W	0h	Vertical Front Porch Value (from 0-255) used to specify the number of line clock periods to add to the end of each frame Note that the line clock transitions during the insertion of the extra line clock periods
15-10	vsw	R/W	0h	Vertical Sync Width Pulse In active mode (lcdtft=1), encoded value (from 1-64) used to specify the number of line clock periods to set the lcd_fp pin active at the end of each frame after the (vfp) period elapses The number of clock cycles is programmed value plus one The frame clock is used as the VSYNC signal in active mode In passive mode (lcdtft=0), encoded value (from 1-64) used to specify the number of extra line clock periods to insert after the vertical front porch (vfp) period has elapsed Note that the width of lcd_fp is not affected by vsw in passive mode and line clock transitions during the insertion of the extra line clock periods (programmed value plus one)
9-0	lpp	R/W	0h	Lines Per Panel Encoded value (programmed value range of {0:2047} represents an actual range of {1:2048}) used to specify the number of lines per panel It represents the total number of lines on the LCD (programmed value plus one) Bit 10 of this field is in RASTER_TIMING_2