

实验三 基于液晶显示屏的人机接口实验

一、 实验目的

- 1. 掌握液晶屏的使用及其电路设计方法；
- 2. 掌握 S3C2410X 处理器的 LCD 控制器的使用；
- 3. 通过实验掌握液晶显示文本及图形的方法与程序设计；
- 4. 通过实验掌握触摸屏（TSP）的设计与控制方法。

二、 实验内容

- 1. 掌握液晶屏作为人机接口界面的设计方法，并编写程序实现；
- 2. 编程实现触摸屏坐标转换为液晶对应坐标；
- 3. 编程实现由液晶屏和触摸屏构成的可以互动的人机界面，至少实现 3 屏。

三、 实验原理

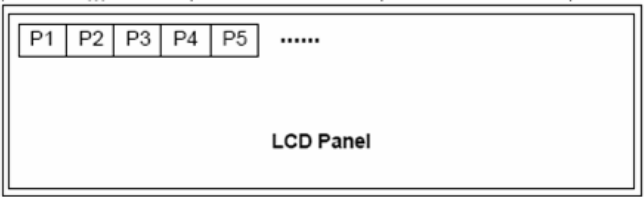
- 1. 液晶显示屏（LCD）
型号 LQ080V3DG01 外形尺寸 183×141×14 mm 重量 390g
像素 320 × 240 点 距 0.2535 X0.2535 mm 色彩 262144
电压 5V（25℃） 对比度 250 附加带驱动逻辑



- 2. 像素在内存中的表示（LCD）

(BSWP = 0, HWSWP = 0)

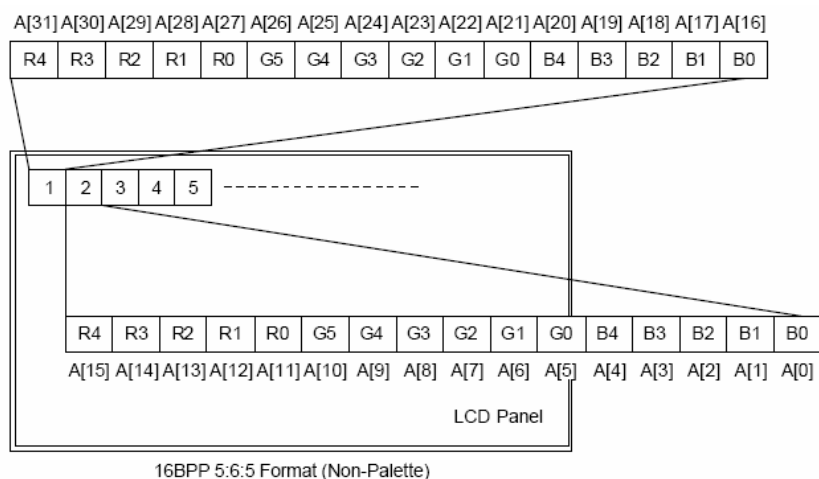
	D[31:16]	D[15:0]
000H	P1	P2
004H	P3	P4
008H	P5	P6
...		



VD Pin Connections at 16BPP

(5:6:5)

VD	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RED	4	3	2	1	0	NC									NC									NC
GREEN									5	4	3	2	1	0										
BLUE																		4	3	2	1	0		

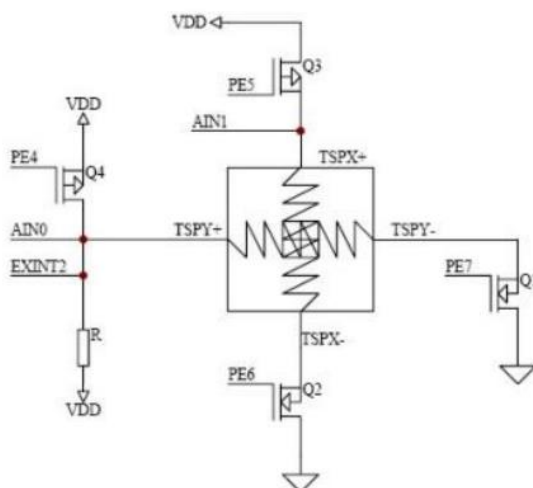


LCD 屏的尺寸决定了 HOZVAL 和 LNEVAL 的值，且一般为 HOZVAL=水平显示尺寸-1，LINEVAL=垂直显示尺寸-1。

通过以上原理图，可知一个像素采用了 16 位即两个字节进行表示，这种表示方法主要跟显存大小有关，因为一般每个像素由 RGB 三元素组成，即为一个三元向量，如果每一个元素都用一个字节表示，则一个像素就需要 24 位进行表示。但是针对 16 位 TFT 屏，所拥有的显存受限，因此采用 5：6：5 的编码表示方法。

一般显存位于内存起始地址（0x3000 0000）偏移 16M（0x0100 0000）的地方。计算得到显存的大小：320×240×2 = 153600 字节

3. 触摸屏的等效电路



四、 实验步骤

1. 验证示例源码

- 拷贝整个实验例程源码目录到本地磁盘自己的工作目录下;
 - 使用 μ Vision IDE for ARM 通过 ULINK2 仿真器连接实验板, 打开实验例程目录 11_LCD_Test 子目录下的 LCD_Test.Uv2 例程, 编译链接工程;
 - 调试程序, 观察运行结果;
 - 打开实验例程目录 07_TSP_Test 子目录下的 TSP_Test.Uv2 例程, 编译链接工程;
 - 调试程序, 观察运行结果。
2. 设计实现自己的人机互动界面程序
- 拷贝示例实验源码工程;
 - 设计程序, 实现由液晶屏和触摸屏构成的可以互动的人机界面, 至少实现 3 屏。

五、 实验结果

具体实验效果是, 首先进入的是首页, 展现了由两个横线隔开的三个模块, 第一个模块是 “Xidian University”, 第二个模块是 “Name: Zhihao Li ID:21009290012”, 第三个模块是 “Welcome to the main page!”, 点击右边的方框 “Right”, 然后进入第二个页面, 其中第一个模块是 “Xidian University”, 第二个模块是 “Name: Zhihao Li”, 第三个模块是 “Welcome to the second page!”, 点击右边的方框 “Right”, 然后进入第三个页面, 其中第一个模块是 “Xidian University”, 第二个模块是 “ID:21009290012”, 第三个模块是 “Welcome to the third page!”。

在每个页面上, 页面左边都有一个方框 “Left”, 可以实现向左切换页面, 并且切换到最后一张页面会重写切回主页, 因此实现了循环页面的功能。

六、 程序说明

在程序核心代码如下:

1. 修改 tsp_int 函数, 加入中断处理调度程序。

g_nPosX 和 g_nPosY 用来获得触摸屏收到的触摸坐标, 根据触摸的位置相应的做出页面调度, 显示不同的页面。

这里我根据触摸位置进行了两个判断, 如果 g_nPosX 在 [400, 450] 之间, g_nPosY 在 [150, 250] 之间, 表示触摸到的是 “Left” 按钮, 所以进行 “show” 指针数组的循环递减操作, 实现的功能是向左切换页面。

如果 g_nPosX 在 [400, 550] 之间, g_nPosY 在 [850, 1000] 之间表示触摸到的是 “Right” 按钮, 所以进行 “show” 指针数组的循环递增操作, 实现的功能是向右切换页面。

```
1. void (*show[3])(void) = {show_first, show_second, show_third};
2. /*****
3. * name:      tsp_int
4. * func:      Touch screen interrupt handler (ADC)
5. * para:      none
6. * ret:       none
7. * modify:
8. * comment:
9. *****/
```

```

10. void __irq tsp_int(void)
11. {
12.     int i;
13.     UINT32T szPos[40];
14.     rINTSUBMSK |= (BIT_SUB_ADC | BIT_SUB_TC); // Mask sub interrupt (ADC and TC)
15.     // TC(Touch screen Control) Interrupt
16.     if( rADCTSC & 0x100)
17.     {
18.         rADCTSC&=0xff; // Set stylus down interrupt
19.     }
20.     else
21.     {
22.         uart_printf(" Stylus Down!!\n");
23.         szPos[30] = g_nPosX;
24.         szPos[34] = g_nPosY;
25.
26.         // <X-Position Read>
27.         //Hi-Z,AIN5,GND,Ext vlt,Pullup Dis,Normal,X-position
28.         rADCTSC = (0<<8)|(0<<7)|(1<<6)|(1<<5)|(0<<4)|(0<<3)|(0<<2)|(1);
29.         //adc input ain5
30.         rADCCON = (1<<14)|(39<<6)|(5<<3)|(0<<2)|(1<<1)|(0);
31.         rADCDAT0;
32.         delay(10);
33.         for(i = 0,g_nPosX=0; i<nSampleNo; i++)
34.         {
35.             while(!(0x8000 & rADCCON)); // Check ECFLG
36.             szPos[i] = (0x3ff & rADCDAT0);
37.             g_nPosX += szPos[i];
38.         }
39.         g_nPosX = g_nPosX/nSampleNo;
40.         // g_nPosX = (g_nPosX-45)*640/685;
41.         uart_printf(" X-Posion[AIN5] is %04d\n", g_nPosX);
42.
43.         // <Y-Position Read>
44.         //GND,Ext vlt,Hi-Z,AIN7,Pullup Dis,Normal,Y-position
45.         rADCTSC = (0<<8)|(1<<7)|(0<<6)|(0<<5)|(1<<4)|(0<<3)|(0<<2)|(2);
46.         //adc input ain7
47.         rADCCON = (1<<14)|(39<<6)|(7<<3)|(0<<2)|(1<<1)|(0);
48.         rADCDAT1;
49.         delay(10);
50.         for(i = 0,g_nPosY=0; i<nSampleNo; i++)
51.         {

```

```

52.         while(!(0x8000 & rADCCON)); // Che
        ck ECFLG
53.         szPos[i] = (0x3ff & rADCDA1);
54.         g_nPosY += szPos[i];      }
55.         g_nPosY = g_nPosY/nSampleNo;
56.         // g_nPosY = (g_nPosY-70)*480/340;
57.         uart_printf(" Y-Posion[AIN7] is %04d\n", g_nPosY);
58.         //GND,AIN,Hi-z,AIN,Pullup En,Normal,Waiting mode
59.         rADCTSC=(1<<8)|(1<<7)|(1<<6)|(0<<5)|(1<<4)|(0<<3)|(0<<2)|(3);
60.
61.         if(g_nPosX>400 && g_nPosX<=550 && g_nPosY<=250 && g_nPosY>=150){
62.             j = (j - 1 + 3) % 3;
63.             show[j]();
64.         }
65.         else if(g_nPosX>=400 && g_nPosX<=550 && g_nPosY<=1000 && g_nPosY>=850){
66.             j = (j + 1) % 3;
67.             show[j]();
68.         }
69.     }

```

2. 修改 LCD 显示函数 color_lcd_test

每一次触摸屏检测到有触摸，则就会根据触摸的位置调用 color_lcd_test 函数，然后 color_lcd_test 函数会根据触摸的位置调度不同的页面，来作出相应的反应。

```

1.  /*****
2.  * name:      Lcd_show_init()
3.  * func:      LCD initialize function
4.  * para:      none
5.  * ret:      none
6.  * modify:
7.  * comment:  初始化页面配置，用于每个页面的统一配置
8.  *****/
9.  void Lcd_show_init(void)
10. {
11.     Lcd_Clear(0xff);
12.
13.     Lcd_Draw_Box(20, 100, 60, 140, 0x00);
14.     Lcd_DspAscII6x8(30, 120, 0x00, "Left");
15.
16.     Lcd_Draw_Box(260, 100, 300, 140, 0x00);
17.     Lcd_DspAscII6x8(270, 120, 0x00, "Right");
18.
19.     Lcd_Draw_HLine(100, 200, 80, 0x00);
20.     Lcd_DspAscII6x8(130, 100, 0x00, "Xidian University");
21.     Lcd_Draw_HLine(100, 200, 120, 0x00);

```

```

22. }
23.
24. /*****
25. * fuc: show page function
26. * comment: 个性化地配置页面，用于每个页面的个性化配置
27. *****/
28. void show_first(void)
29. {
30.     Lcd_show_init();
31.     Lcd_DspAscII6x8(80, 130, 0x00, "Name: Zhihao Li ID: 21009290012");
32.     Lcd_DspAscII6x8(80, 140, 0x00, "Welcome to the main page!");
33. }
34.
35. void show_second(void)
36. {
37.     Lcd_show_init();
38.
39.     Lcd_DspAscII6x8(80, 130, 0x00, "Name: Zhihao Li");
40.     Lcd_DspAscII6x8(80, 140, 0x00, "Welcome to the second page!");
41. }
42.
43. void show_third(void)
44. {
45.     Lcd_show_init();
46.
47.     Lcd_DspAscII6x8(80, 130, 0x00, "ID: 21009290012");
48.     Lcd_DspAscII6x8(80, 140, 0x00, "Welcome to the third page!");
49. }

```

七、 心得体会

在这次实验中，我在实践中掌握了如何操作触摸屏、LCD 显示的功能，通过学习实例工程代码，我加深了对中断程序的理解，尤其是中断程序和 I/O 复用的工作原理，变得更加深刻。同时对 LCD 屏显示坐标和触摸坐标的转换，能够实现熟练的运用。