

## 实验七 基于 SPI 接口的 TFT 屏幕显示接口技术

### 1. 实验要求

每位同学设计一个点阵彩色的 logo 图案，并在点阵彩色屏幕上显示该 logo 和本人的中文姓名（汉字显示，可以借鉴附录上提供的点阵生成工具，或者自己寻找设计方案）。

### 2. 设计思路

在显示中文姓名的设计目标中，采用 `etft_DisplayString` 对工具生成的汉字点阵进行显示，实现细节参考 `dr_tft2.c` 内流程，设置屏幕的显示区域后，依次对各像素点进行扫描，与点阵数据比对以确定该像素点显示字符颜色或背景颜色。需要更改的注意点为：原先显示的英文字符为  $8 \times 16$ ，调整为汉字  $16 \times 16$  时须修改对应的边界条件，

在显示图片时，采用 `Img2Lcd` 工具将图片转化为 24 位的位图数据，使用 `etft_DisplayImage` 函数进行显示。

### 3. 实验代码

修改后的 `etft_DisplayString` 函数如下（主程序内仅为简单调用）

```

26 void etft_DisplayString(const char* str, uint16_t sx, uint16_t sy, uint16_t fRGB, uint16_t bRGB)
27 {
28     uint16_t cc = 0;
29     uint16_t cx, cy;
30     while(1)
31     {
32         if(cc == 10) //名字共5个字, 即10个8*16的像素块
33             return;
34         cx = 0;
35         cy = 0;
36         //屏幕是横的, xy要对调
37         tft_SendCmd(TFTREG_WIN_MINX, sx); //x start point
38         tft_SendCmd(TFTREG_WIN_MINY, sy); //y start point
39         tft_SendCmd(TFTREG_WIN_MAXX, sx+7); //x end point
40         tft_SendCmd(TFTREG_WIN_MAXY, sy+15); //y end point
41         tft_SendCmd(TFTREG_RAM_XADDR, sx); //x start point
42         tft_SendCmd(TFTREG_RAM_YADDR, sy); //y start point
43         tft_SendIndex(TFTREG_RAM_ACCESS);
44         uint16_t color;
45         while(1)
46         {
47             if(cy >= 8)
48             {
49                 cy = 0;
50                 cx++;
51                 if(cx >= 16)
52                 { //一个字符发送完毕
53                     cc++; //下一个字符
54                     sx+=8;
55                     if(sx >= TFT_YSIZE) //越过行末
56                     {
57                         sx = 0;
58                         sy += 16;
59                     }
60                     break;
61                 }
62             }
63             if((tft_ascii[cc*16 + cx] << cy) & 0x80)
64                 color = fRGB;
65             else
66             {
67                 if((tft_ascii[cc*16 + cx] << cy) & 0x80)
68                     color = fRGB;
69                 else
70                     color = bRGB;
71                 tft_SendData(color);
72                 cy++; //y自增
73             }
74         }
75     }
76 }

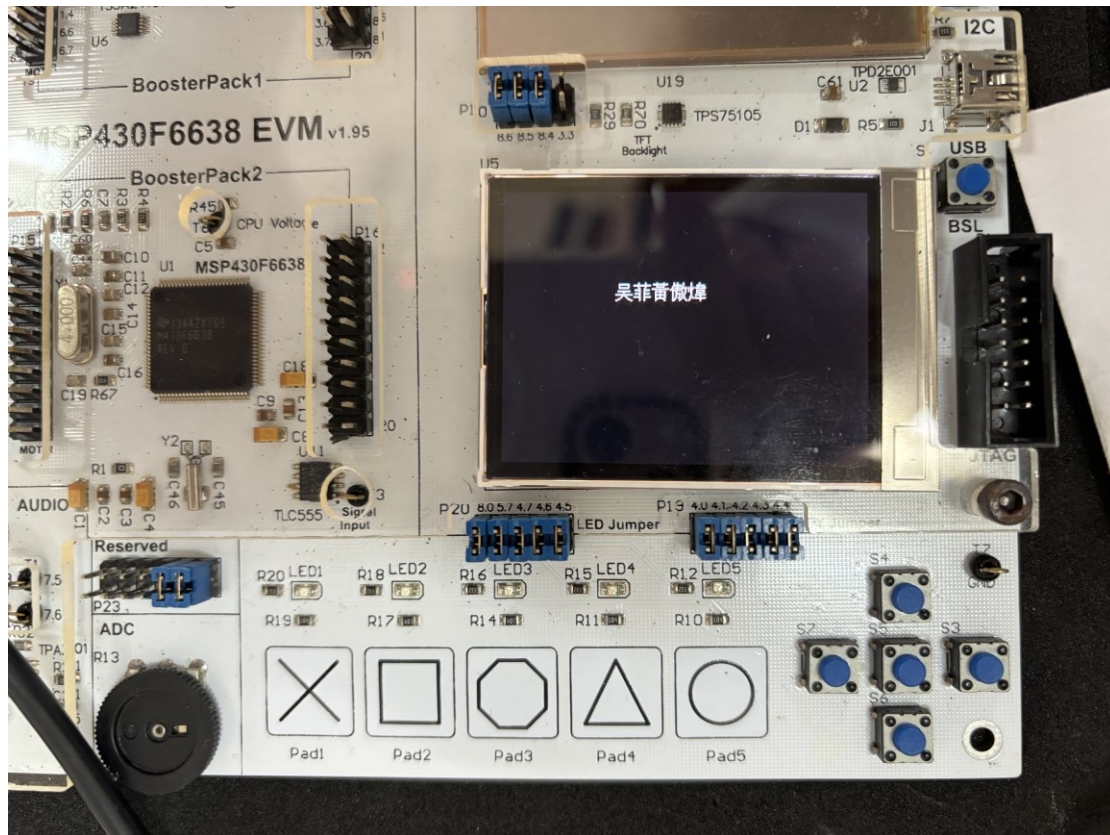
```

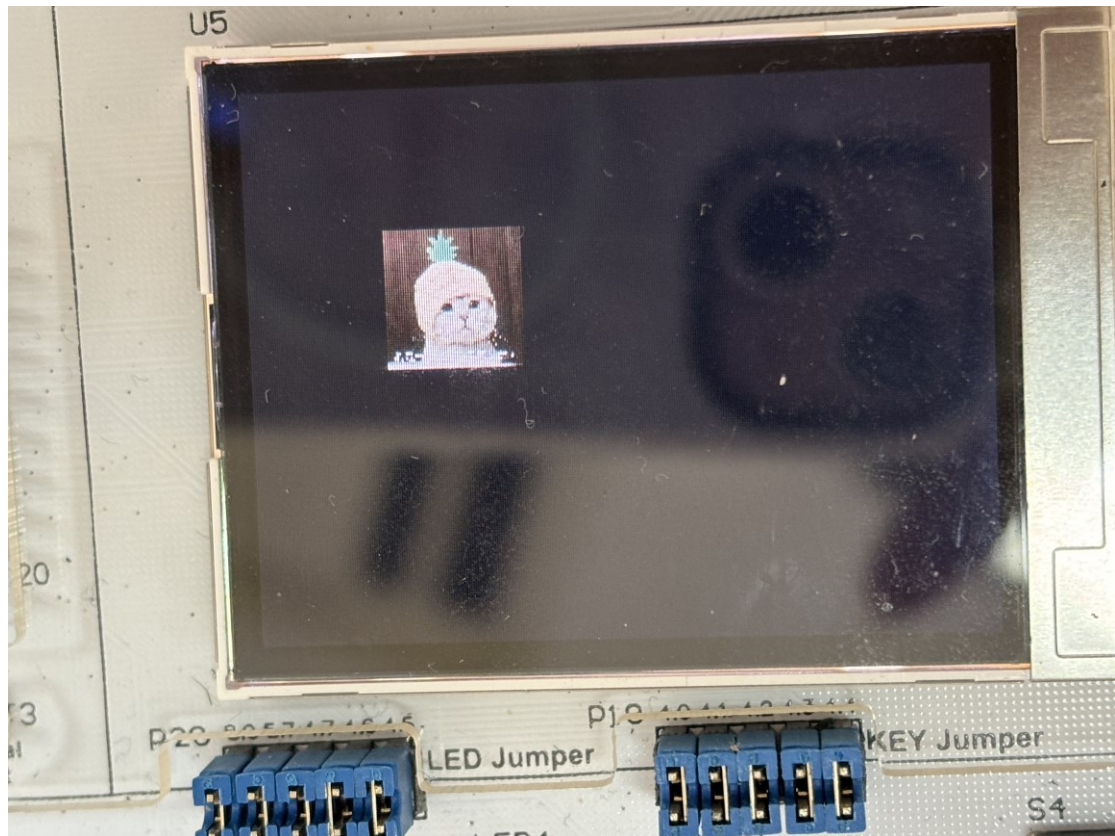
```

34 int main( void )
35 {
36
37     // Stop watchdog timer to prevent time out reset
38     WDTCTL = WDTPW + WDTHOLD;
39
40     _DINT();
41
42     initClock();
43     initTFT();
44
45     _EINT();
46
47     etft_AreaSet(0,0,319,239,0);
48
49
50     while(1){
51         etft_DisplayString(" ",100,80,65535,0); //这是显示文字
52         __delay_cycles(MCLK_FREQ*3);
53         etft_DisplayImage(img,60,60, 60, 60); //这是显示图片
54         __delay_cycles(MCLK_FREQ*3);
55     }
56 }

```

#### 4. 实验结果





如图，可分别显示汉字和图片。

## 5. 分析与思考

在汉字的显示中，需要注意显示屏的  $x$  坐标与  $y$  坐标需要对调，否则显示出的结果会旋转  $90^\circ$ 。在图片的显示中生成位图点阵时，，需要注意显示像素的扫描顺序为从左到右，从下到上。在调试过程中持续出现显示结果乱码和仅显示一小块白色高亮的情况，经实验发现，减小生成位图点阵的大小（验证可行的大小为  $60 \times 60$ ）即可成功显示。