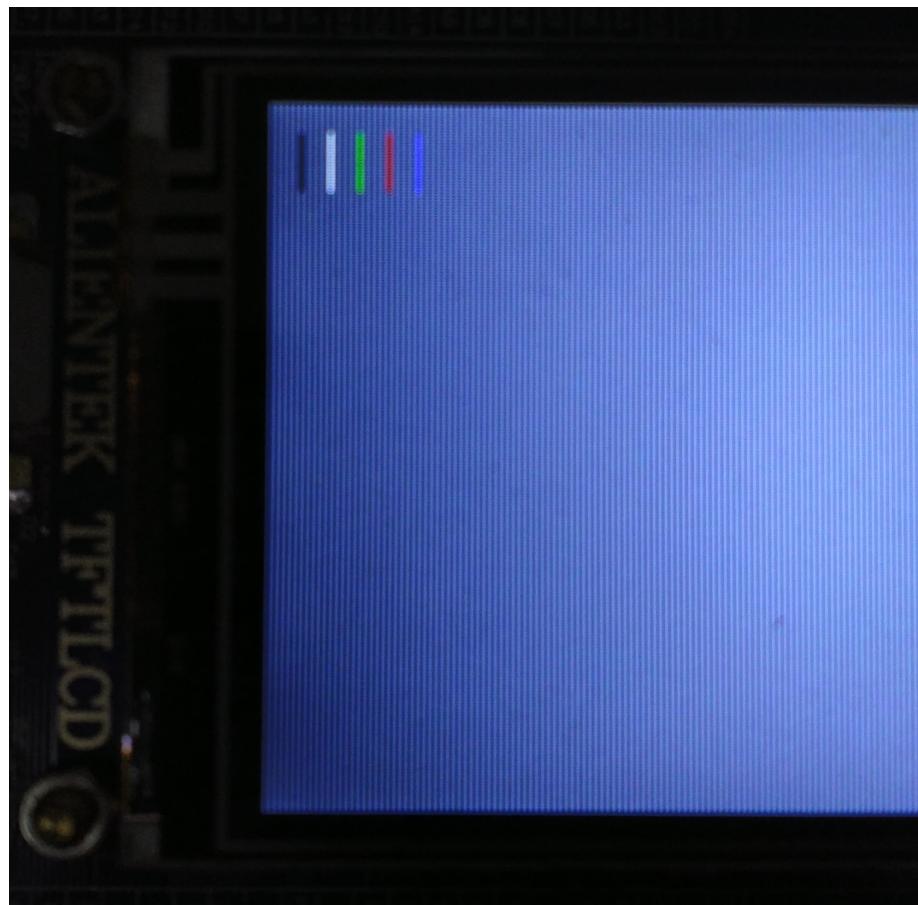


IERG3810 Lab 3 Flexible Static Memory Controller (FSMC)

HU, Han 1155107763 Session C No board number (board bought by myself)

Experiment 3.1: Draw lines on the TFT-LCD with different color

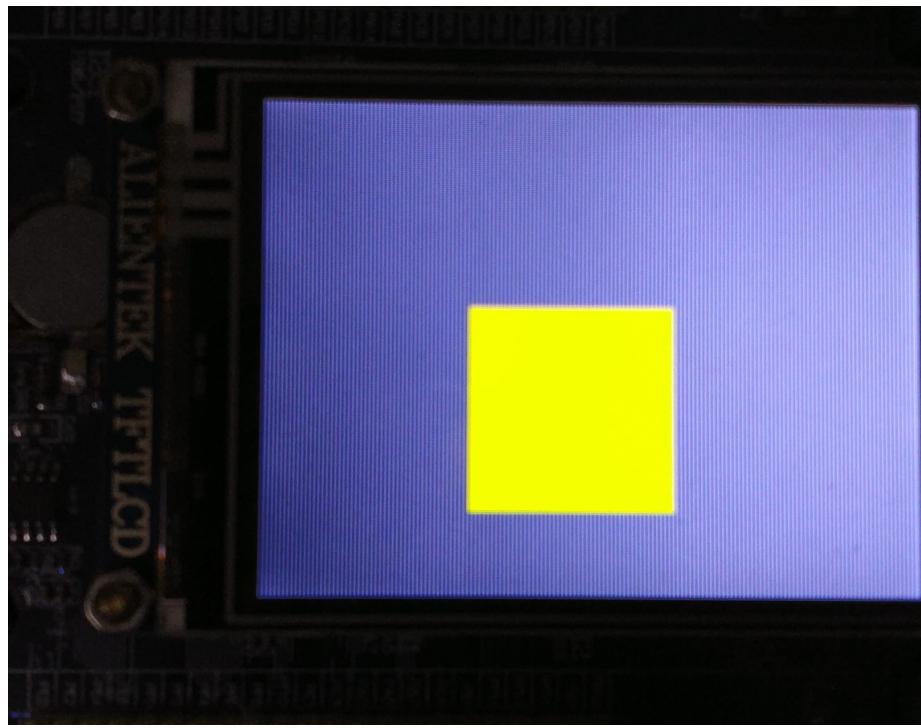
Result:



The origin $(0, 0)$ of the screen is the bottom left corner of the TFT-LCD screen (the nearest corner of the lines in the above photo).

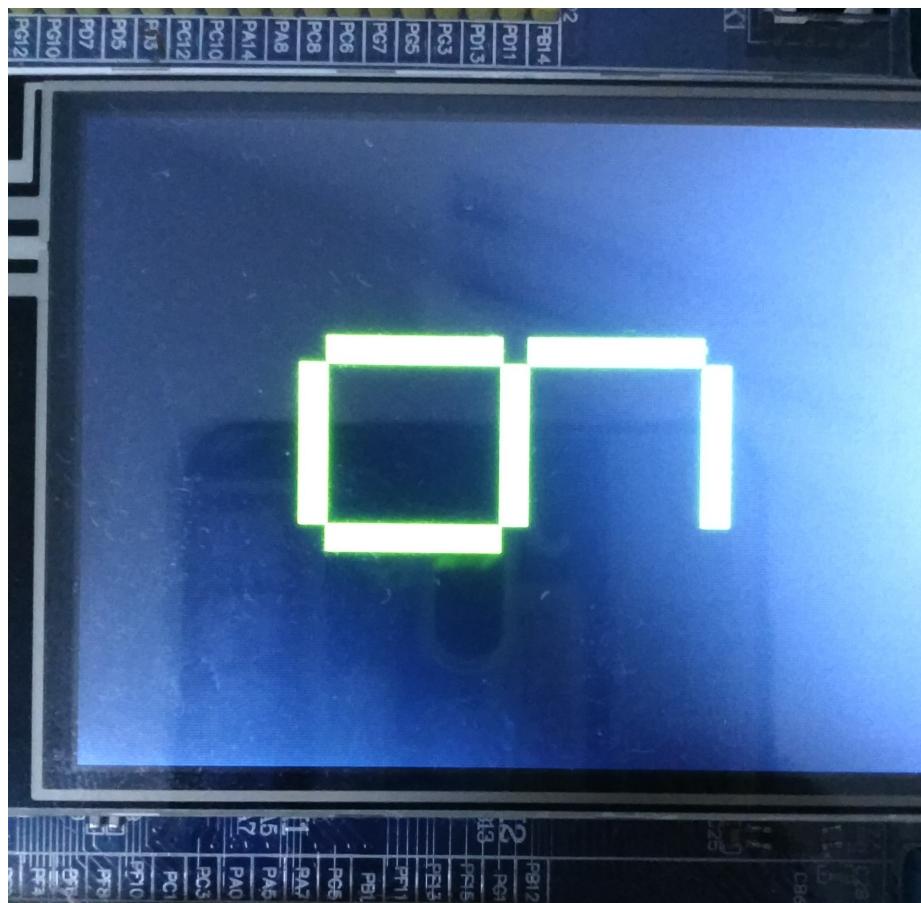
Experiment 3.2: Draw rectangle on the TFT-LCD

Result:

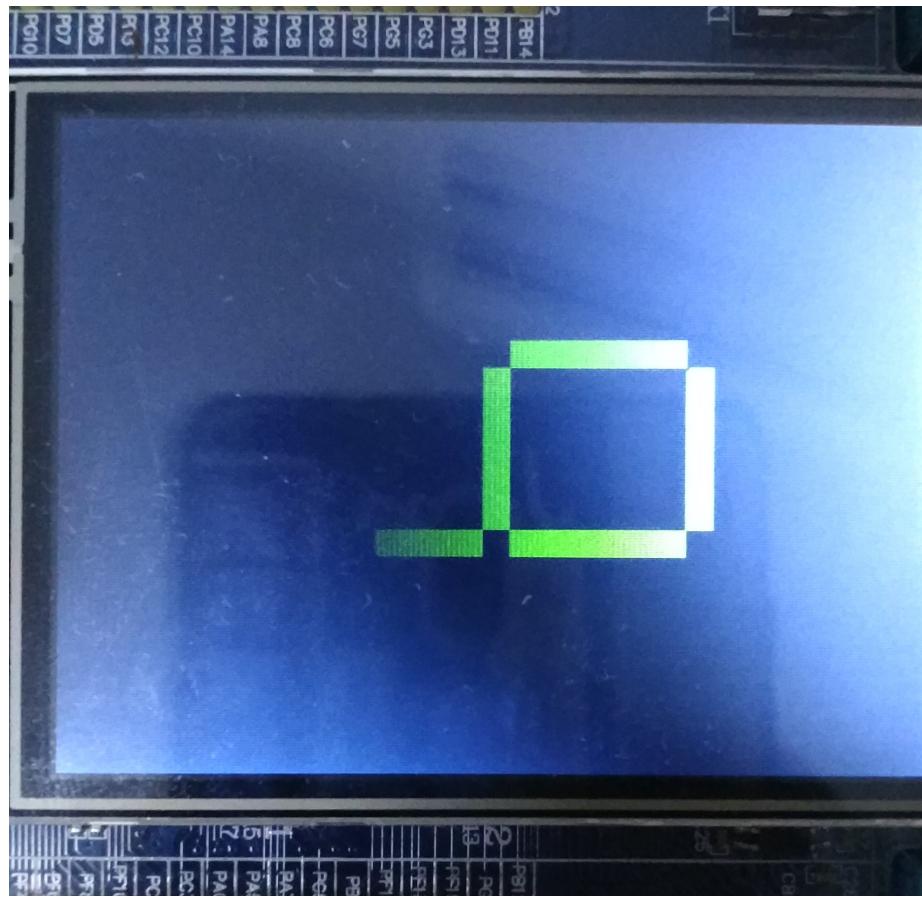


Experiment 3.3: Draw a counting down digit on the TFT-LCD screen

Result with displayed digit:



Result when changing digits (halfway):

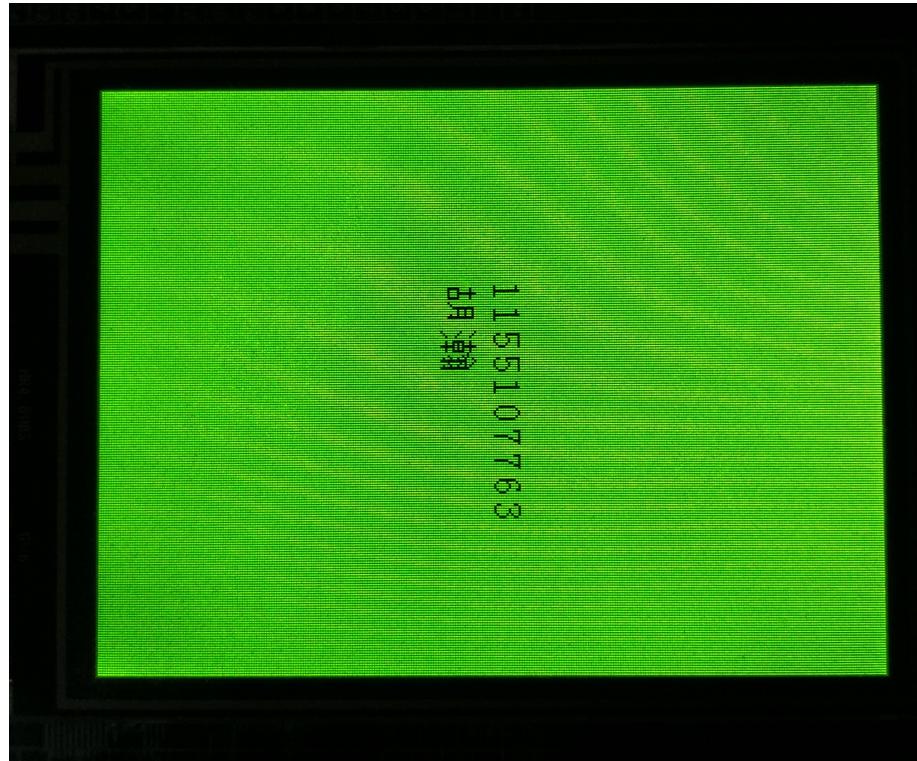


Solution to changing digits:

Using `FillRectangle()` subroutine to first overwrite the previously displayed digit and refresh the whole screen and then use `SevenSegment()` to draw the digit above the background.

Experiment 3.4 & 3.5: Show alphabets and Chinese characters on TFT-LCD

Result:

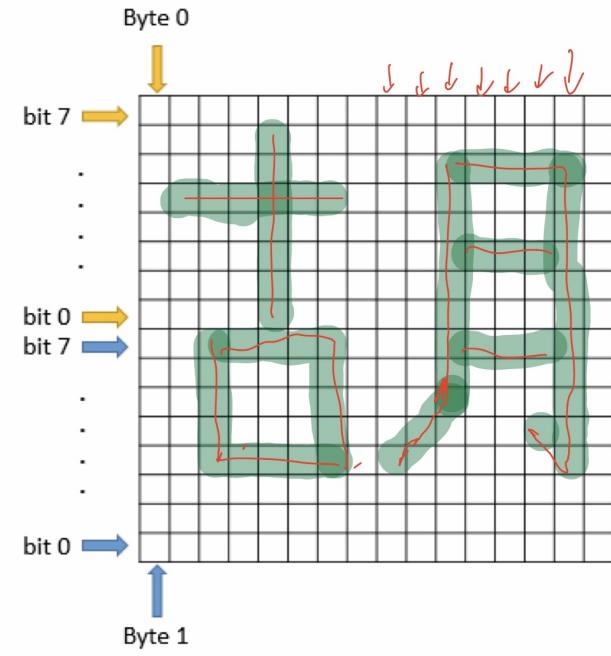


The details for designing the CFONT.H and the character matrix:

```
1 #ifndef __CFONT_H
2 #define __CFONT_H
3 const unsigned char chi_1616[2][32]={
4
{0x00,0x00,0x00,0x00,0x10,0xF8,0x10,0x88,0x7F,0x88,0x10
,0x88,0x10,0xF8,0x00,0x00,0x00,0x08,0x00,0x10,0x3F,0xE0
,0x24,0x80,0x24,0x80,0x24,0x90,0x3F,0xF8,0x00,0x00}, // HU
5
{0x41,0x01,0x20,0x82,0x10,0x44,0x00,0x00,0x13,0xE4,0x12
,0xA4,0x7E,0xBF,0x12,0xA4,0x13,0xE4,0x08,0x00,0x0A,0x08
,0x24,0x92,0x47,0xFF,0x25,0x08,0x14,0x92,0x0F,0xFF}
//HAN
6 };
7 #endif
8
```

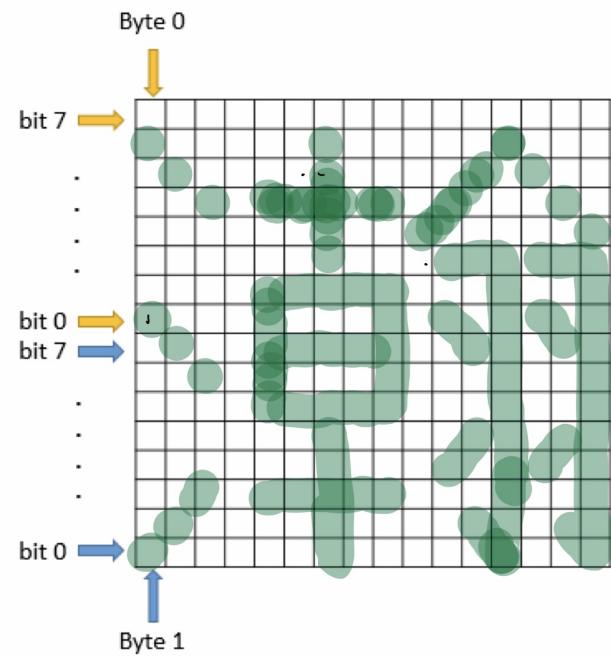
Character dot matrix for 胡:

Figure 1-18 Chinese Character Dot Matrix



Character dot matrix for 瀚:

Figure 1-18 Chinese Character Dot Matrix



Experiment 3.6: Characters overlay on the TFT-LCD

Result:



From the above result that we can observe that my CUID displayed does not disobey either the left part nor the right part of the screen.

For completing the overlay mission, I use `DrawDot()` subroutine instead of the original `ShowChar()` subroutine. By this way, we only change the color of dot which is the location of the character and skip the part which should be covered with `bgcolor` in `ShowChar()` subroutine.

The controlling function is still the same as the `ShowChar()` subroutine and the dots that should be modified are still determined by the character matrix.

```
1 void IERG3810_TFTLCD_ShowChar_overlay(u16 x,u16 y,u8
2     ascii, u16 color){
3     u8 i, j;
4     u8 index;
5     u8 height = 16, length = 8;
6     if (ascii < 32 || ascii > 127) return;
7     ascii -= 32;
8     IERG3810_TFTLCD_WrReg(0x2A);
9     IERG3810_TFTLCD_WrData(x >> 8);
10    IERG3810_TFTLCD_WrData(x & 0xFF);
11    IERG3810_TFTLCD_WrData((length + x - 1) >> 8);
12    IERG3810_TFTLCD_WrData((length + x - 1) & 0xFF);
13    IERG3810_TFTLCD_WrReg(0x2B);
14    IERG3810_TFTLCD_WrData(y >> 8);
15    IERG3810_TFTLCD_WrData(y & 0xFF);
16    IERG3810_TFTLCD_WrData((height + y - 1) >> 8);
17    IERG3810_TFTLCD_WrData((height + y - 1) & 0xFF);
18
19    IERG3810_TFTLCD_WrReg(0x2C); //LCD_WriteRAM_Prepare();
```

```

19     for (j = 0; j < height / 8; j++)
20     {
21         for (i = 0; i < height / 2; i++)
22         {
23             for (index = 0; index < length; index++)
24             {
25                 if ((asc2_1608[ascii][index*2+1-j] >> i) &
26                     0x01)
27                 {
28                     if (j == 0)
29                     {
30                         IERG3810_TFTLCD_DrawDot(x+index, y+i,
31                                     color);
32                     }
33                     else if (j == 1)
34                     {
35                         IERG3810_TFTLCD_DrawDot(x+index, y+i+7,
36                                     color);
37                     }
38                 }
39             }

```

Advantages over the provided `ShowChar()` subroutine which uses the `FillRectangle()` mechanism:

- This subroutine does not violate the color background and is adaptive on changing background colors (e.g. photos) while not causing obvious artifact effects.

Disadvantages over `showChar()`:

- Each time we call the `DrawDot()` subroutine, we need to go through all writing REG (Column Address, Page Address and Memory write) and writing corresponding data about the position information and color information. This process will consume more steps and time to execute than the `ShowChar()`.