<<IERG3810>>

<<Microcontroller and Embedded Systems Laboratory>>

Report on Experiment <<3>>

<<Flexible Static Memory Controller >>

Group: 19

Member: Chan Kai Yin (CUID: 1155124983)

Ip Tsun Yu (CUID: 1155144668)

Submission Date: 23-02-2022

Disclaimer

I declare that the assignment here submitted is original except for source material explicitly acknowledged, and that the same or related material has not been previously submitted for another course. I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the website http://www.cuhk.edu.hk/policy/academichonesty/

Jesse ______ Chan Kai Yin ______ 23-02-2022 ______

Signature Name Date

Derek ______ Ip Tsun Yu ______ 23-02-2022 ______

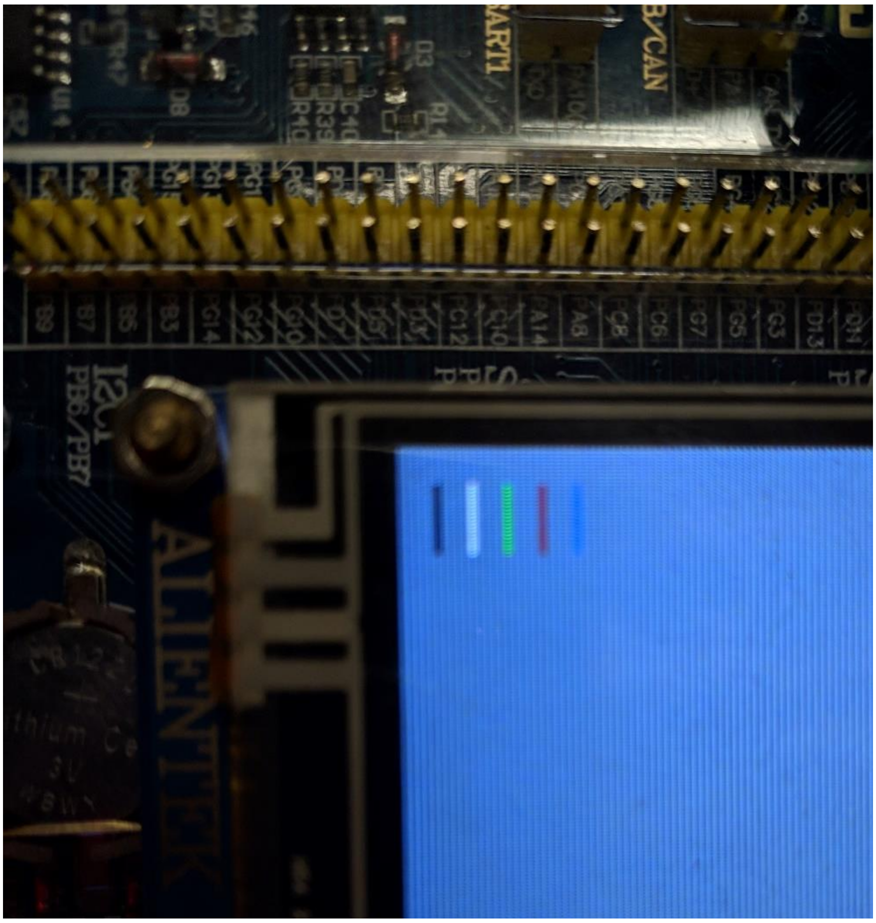
Signature Name Date

I. OBJECTIVES

- To study the interfacing of Flexible Static Memory Controller of Cortex-M3.
- To study address decoding of FSMC.
- To interface a TFT-LCD with FSMC.
- To study the control of TFT-LCD.
- To study displaying alphabet and Chinese characters on TFT-LCD.

II. DATA ANALYSIS

Experiment 3.1



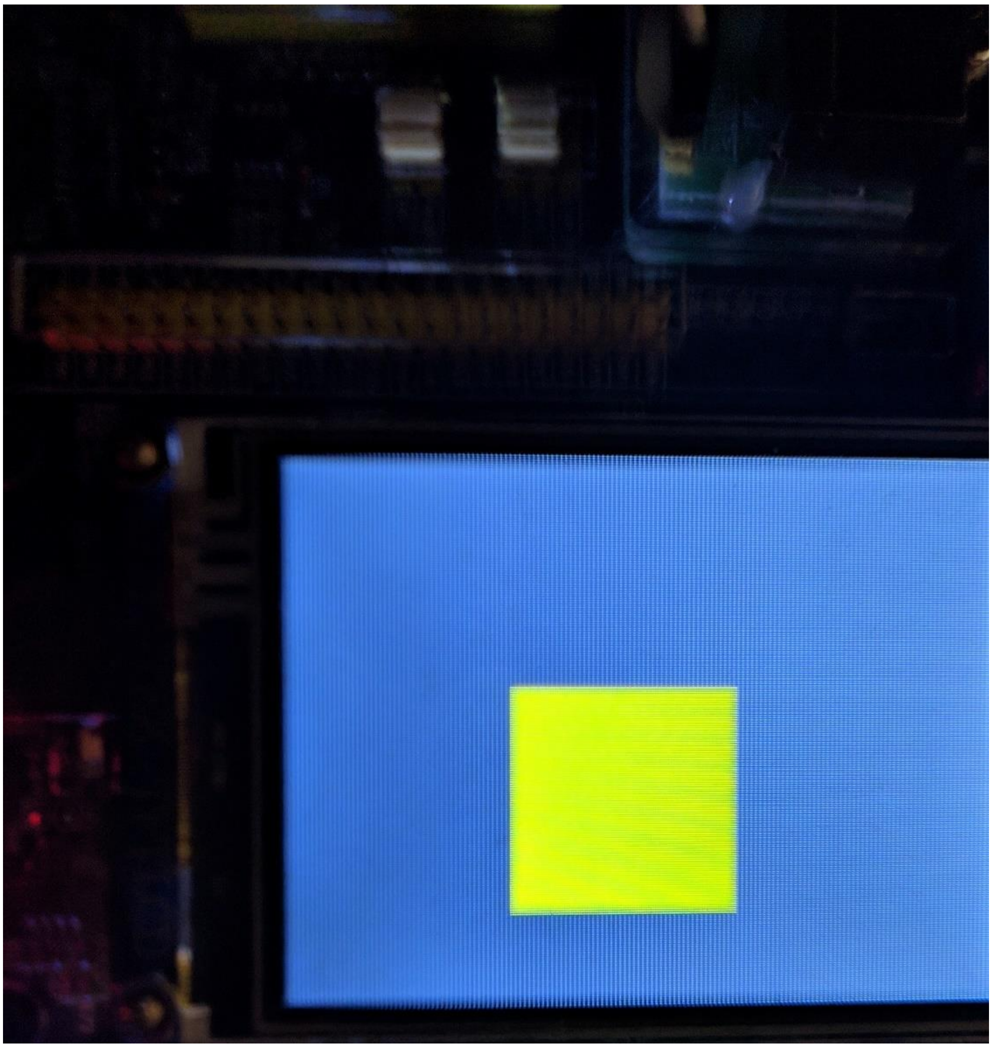
Using for loop in IERG3810_TFTLCD_DrawLine function:

```
#include "stm32f10x.h"
#include "IERG3810_LED.h"
#include "IERG3810_Buzzer.h"
#include "IERG3810_KEY.h"
#include "IERG3810_USART.h"
#include "IERG3810_Clock.h"

void IERG3810_clock_tree_init(void);
void IERG3810_USART2_init(u32, u32);
void IERG3810_USART1_init(u32, u32);
void Delay(u32);
void USART_print(u8, char *);

void Delay(u32 count){
	u32 i;
	for(i = 0; i < count; i++);
}

typedef struct{
	u16 LCD_REG;
	u16 LCD_RAM;
}	LCD_TypeDef;



#define LCD_BASE	((u32)(0x6C000000|	0x000007FE))
#define LCD			((LCD_TypeDef*)	LCD_BASE)

// Color
#define black				(u16)	0
#define white				(u16)	65535
#define green				(u16)	12256
#define red				(u16)	59554
#define blue				(u16)	415
```

```
void IERG3810_TFTLCD_WrReg(u16 regval){
        LCD->LCD_REG = regval;
}
void IERG3810_TFTLCD_WrData(u16 data){
        LCD->LCD_RAM = data;
}



void IERG3810_TFTLCD_SetParameter(void){
        IERG3810_TFTLCD_WrReg(0x01);
        IERG3810_TFTLCD_WrReg(0x11);

        IERG3810_TFTLCD_WrReg(0x3A);
        IERG3810_TFTLCD_WrData(0x55);

        IERG3810_TFTLCD_WrReg(0x29);

        IERG3810_TFTLCD_WrReg(0x36);
        IERG3810_TFTLCD_WrData(0xCA);
}




void IERG3810_TFTLCD_Init(void){
        RCC ->AHBENR |= 1 << 8;
        RCC ->APB2ENR |= 1 << 3;
        RCC ->APB2ENR |= 1 << 5;
        RCC ->APB2ENR |= 1 << 6;
        RCC ->APB2ENR |= 1 << 8;
        GPIOB ->CRL &= 0xFFFFFFF0;
        GPIOB ->CRL |= 0x00000003;

        //PORTD
        GPIOD ->CRH &= 0x00FFF000;
        GPIOD ->CRH |= 0xBB000BBB;
        GPIOD ->CRL &= 0xFF00FF00;
        GPIOD ->CRL |= 0x00BB00BB;

        //PORTE
        GPIOE ->CRH &= 0x00000000;
        GPIOE ->CRH |= 0xBBBBBBBB;
        GPIOE ->CRL &= 0x0FFFFFFF;
        GPIOE ->CRL |= 0xB0000000;

        //PORTG12
        GPIOG ->CRH &= 0xFFF0FFFF;
        GPIOG ->CRH |= 0x000B0000;
        GPIOG ->CRL &= 0xFFFFFFF0;
        GPIOG ->CRL |= 0x0000000B;



        FSMC_Bank1->BTCR[6] = 0x00000000;
        FSMC_Bank1->BTCR[7] = 0x00000000;
        FSMC_Bank1E ->BWTR[6] = 0x00000000;
        FSMC_Bank1 ->BTCR[6] |= 1 << 12;
        FSMC_Bank1 ->BTCR[6] |= 1 << 14;
        FSMC_Bank1 ->BTCR[6] |= 1 << 4;
        FSMC_Bank1 ->BTCR[7] |= 0 << 28;
        FSMC_Bank1 ->BTCR[7] |= 1 << 0;
        FSMC_Bank1 ->BTCR[7] |= 0xF << 8;
        FSMC_Bank1E ->BWTR[6] |= 0 << 28;
        FSMC_Bank1E ->BWTR[6] |= 0 << 0;
        FSMC_Bank1E ->BWTR[6] |= 3 << 8;
        FSMC_Bank1 ->BTCR[6] |= 1 << 0;

        IERG3810_TFTLCD_SetParameter();
        GPIOB ->ODR |= 1 << 0;
        //LCD_LIGHT_ON;
}

void IERG3810_TFTLCD_DrawDot(u16 x, u16 y, u16 color){
        IERG3810_TFTLCD_WrReg(0x2A);
        IERG3810_TFTLCD_WrData(x >> 8);
        IERG3810_TFTLCD_WrData(x & 0xFF);
        IERG3810_TFTLCD_WrData(0x01);
        IERG3810_TFTLCD_WrData(0x3F);
        IERG3810_TFTLCD_WrReg(0x2B);
        IERG3810_TFTLCD_WrData(y >> 8);
        IERG3810_TFTLCD_WrData(y & 0xFF);
        IERG3810_TFTLCD_WrData(0x01);
        IERG3810_TFTLCD_WrData(0xDF);
        IERG3810_TFTLCD_WrReg(0x2C);
        IERG3810_TFTLCD_WrData(color);
}

void IERG3810_TFTLCD_DrawLine(u16 x_St, u16 x_End, u16 y, u16
color){
        u16 x = 0;
        for(x = x_St; x < x_End; x++){
            IERG3810_TFTLCD_DrawDot(x,y,color);
        }
}



int main(void)
{
        IERG3810_LED_Init();
        IERG3810_TFTLCD_Init();

        Delay(2000000);



        IERG3810_TFTLCD_DrawLine(10,30,10,black);
        IERG3810_TFTLCD_DrawLine(10,30,20,white);
        IERG3810_TFTLCD_DrawLine(10,30,30,green);
        IERG3810_TFTLCD_DrawLine(10,30,40,red);
        IERG3810_TFTLCD_DrawLine(10,30,50,blue);
```

```
}
```

Experiment 3.2

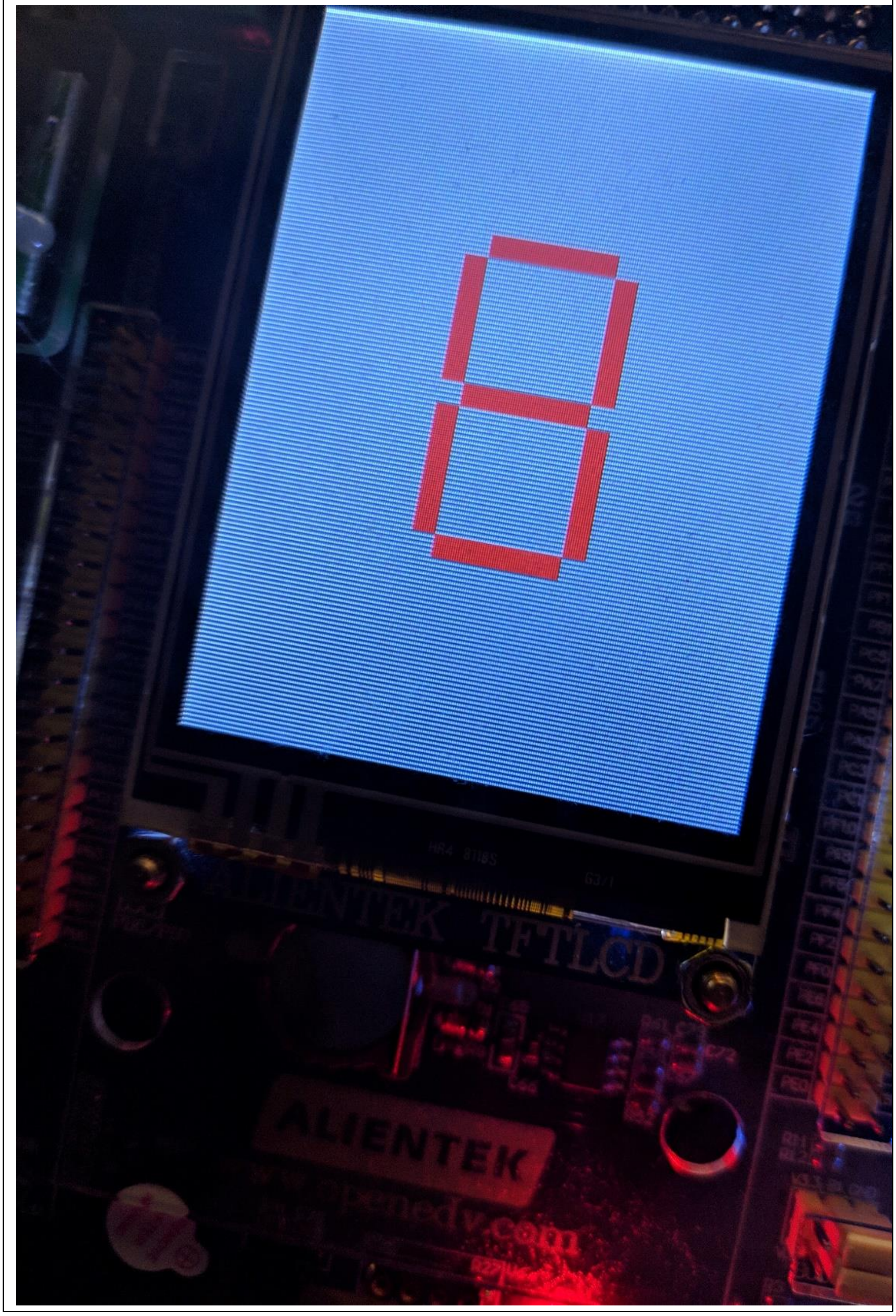


```
#include "stm32f10x.h"
#include "IERG3810_LED.h"
#include "IERG3810_Buzzer.h"
#include "IERG3810_KEY.h"
#include "IERG3810_USART.h"
#include "IERG3810_Clock.h"

void IERG3810_clock_tree_init(void);
void IERG3810_USART2_init(u32, u32);
void IERG3810_USART1_init(u32, u32);
void Delay(u32);
void USART_print(u8, char *);

void Delay(u32 count){
        u32 i;
        for(i = 0; i < count; i++);
}

typedef struct{
        u16 LCD_REG;
        u16 LCD_RAM;
}       LCD_TypeDef;



#define LCD_BASE       ((u32)(0x6C000000|       0x000007FE))
#define LCD                    ((LCD_TypeDef*)  LCD_BASE)

// Color
#define black                          (u16)       0
#define white                          (u16)       65535
#define green                          (u16)       12256
#define red                            (u16)       59554
#define blue                           (u16)       415
#define yellow                         (u16)       65504

void IERG3810_TFTLCD_WrReg(u16 regval){
        LCD->LCD_REG = regval;
}
void IERG3810_TFTLCD_WrData(u16 data){
        LCD->LCD_RAM = data;
}

void IERG3810_TFTLCD_SetParameter(void){
        IERG3810_TFTLCD_WrReg(0x01);
        IERG3810_TFTLCD_WrReg(0x11);

        IERG3810_TFTLCD_WrReg(0x3A);
```

```
	IERG3810_TFTLCD_WrData(0x55);

	IERG3810_TFTLCD_WrReg(0x29);

	IERG3810_TFTLCD_WrReg(0x36);
	IERG3810_TFTLCD_WrData(0xCA);
}



void IERG3810_TFTLCD_Init(void){
	RCC ->AHBENR |= 1 << 8;
	RCC ->APB2ENR |= 1 << 3;
	RCC ->APB2ENR |= 1 << 5;
	RCC ->APB2ENR |= 1 << 6;
	RCC ->APB2ENR |= 1 << 8;
	GPIOB ->CRL &= 0xFFFFFFF0;
	GPIOB ->CRL |= 0x00000003;

	//PORTD
	GPIOD ->CRH &= 0x00FFF000;
	GPIOD ->CRH |= 0xBB000BBB;
	GPIOD ->CRL &= 0xFF00FF00;
	GPIOD ->CRL |= 0x00BB00BB;

	//PORTE
	GPIOE ->CRH &= 0x00000000;
	GPIOE ->CRH |= 0xBBBBBBBB;
	GPIOE ->CRL &= 0x0FFFFFFF;
	GPIOE ->CRL |= 0xB0000000;

	//PORTG12
	GPIOG ->CRH &= 0xFFF0FFFF;
	GPIOG ->CRH |= 0x000B0000;
	GPIOG ->CRL &= 0xFFFFFFF0;
	GPIOG ->CRL |= 0x0000000B;
	FSMC_Bank1->BTCR[6] = 0x00000000;
	FSMC_Bank1->BTCR[7] = 0x00000000;
	FSMC_Bank1E ->BWTR[6] = 0x00000000;
	FSMC_Bank1 ->BTCR[6] |= 1 << 12;
	FSMC_Bank1 ->BTCR[6] |= 1 << 14;
	FSMC_Bank1 ->BTCR[6] |= 1 << 4;
	FSMC_Bank1 ->BTCR[7] |= 0 << 28;
	FSMC_Bank1 ->BTCR[7] |= 1 << 0;
	FSMC_Bank1 ->BTCR[7] |= 0xF << 8;
	FSMC_Bank1E ->BWTR[6] |= 0 << 28;
	FSMC_Bank1E ->BWTR[6] |= 0 << 0;
	FSMC_Bank1E ->BWTR[6] |= 3 << 8;
	FSMC_Bank1 ->BTCR[6] |= 1 << 0;

	IERG3810_TFTLCD_SetParameter();
	GPIOB ->ODR |= 1 << 0;
	//LCD_LIGHT_ON;
}

void IERG3810_TFTLCD_DrawDot(u16 x, u16 y, u16 color){

	IERG3810_TFTLCD_WrReg(0x2A);
	IERG3810_TFTLCD_WrData(x >> 8);
	IERG3810_TFTLCD_WrData(x & 0xFF);
	IERG3810_TFTLCD_WrData(0x01);
	IERG3810_TFTLCD_WrData(0x3F);
	IERG3810_TFTLCD_WrReg(0x2B);
	IERG3810_TFTLCD_WrData(y >> 8);
	IERG3810_TFTLCD_WrData(y & 0xFF);
	IERG3810_TFTLCD_WrData(0x01);
	IERG3810_TFTLCD_WrData(0xDF);
	IERG3810_TFTLCD_WrReg(0x2C);
	IERG3810_TFTLCD_WrData(color);
}

void IERG3810_TFTLCD_DrawLine(u16 x_St, u16 x_End, u16 y, u16 
color){
        u16 x = 0;
        for(x = x_St; x < x_End; x++){
                IERG3810_TFTLCD_DrawDot(x,y,color);
        }
}

void IERG3810_TFTLCD_FillRectangle(u16 color, u16 start_x, u16
length_x, u16 start_y, u16 length_y){
        u32 index = 0;
        IERG3810_TFTLCD_WrReg(0x2A);
        IERG3810_TFTLCD_WrData(start_x >> 8);
        IERG3810_TFTLCD_WrData(start_x & 0xFF);
        IERG3810_TFTLCD_WrData((start_x + length_x - 1) >> 8);
        IERG3810_TFTLCD_WrData((start_x + length_x - 1) & 0xFF);

        IERG3810_TFTLCD_WrReg(0x2B);
        IERG3810_TFTLCD_WrData(start_y >> 8);
        IERG3810_TFTLCD_WrData(start_y & 0xFF);
        IERG3810_TFTLCD_WrData((start_y + length_y - 1) >> 8);
        IERG3810_TFTLCD_WrData((start_y + length_y - 1) & 0xFF);

        IERG3810_TFTLCD_WrReg(0x2C);
        for(index = 0; index < length_x * length_y; index ++ ){
                IERG3810_TFTLCD_WrData(color);
        }
}

int main(void)
{
        IERG3810_LED_Init();
        IERG3810_TFTLCD_Init();

        Delay(2000000);
        IERG3810_TFTLCD_FillRectangle(yellow,100,100,100,100);

}
```

Experiment 3.3

Draw digit '8' (7-segment format only) at the center of the screen:



Using “SevenSegments.H” for stating the seven segments for each digit:

```
#ifndef SevenSegments_H
#define SevenSegments_H

const unsigned char Se_Seg[10][7]={

{0x1,0x1,0x1,0x1,0x1,0x1,0x0},//0
{0x0,0x1,0x1,0x0,0x0,0x0,0x0},//1
{0x1,0x1,0x0,0x1,0x1,0x0,0x1},//2
{0x1,0x1,0x1,0x1,0x0,0x0,0x1},//3
{0x0,0x1,0x1,0x0,0x0,0x1,0x1},//4
{0x1,0x0,0x1,0x1,0x0,0x1,0x1},//5
{0x1,0x0,0x1,0x1,0x1,0x1,0x1},//6
{0x1,0x1,0x1,0x0,0x0,0x0,0x0},//7
{0x1,0x1,0x1,0x1,0x1,0x1,0x1},//8
{0x1,0x1,0x1,0x1,0x0,0x1,0x1},//9
};

#endif
```

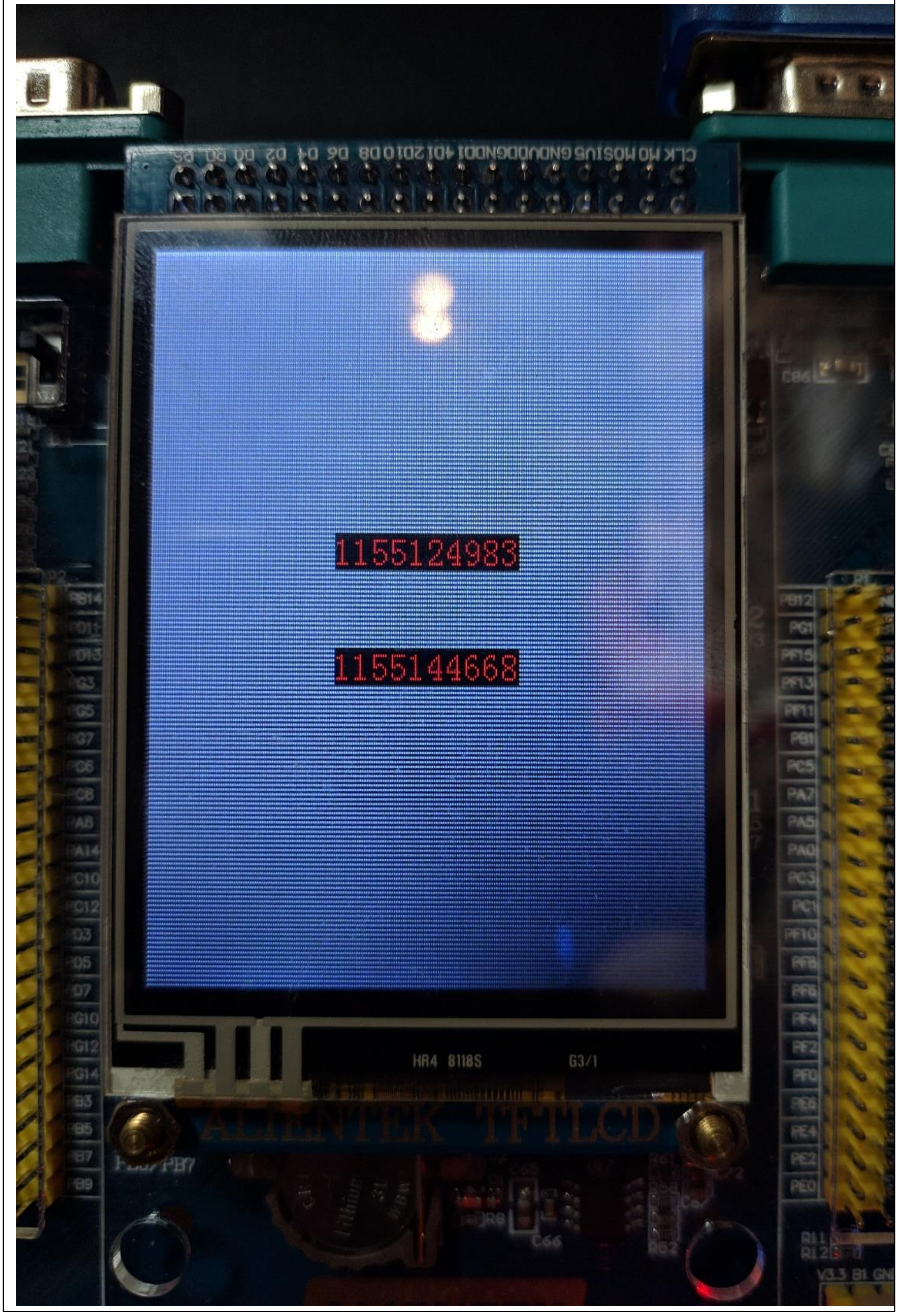
Count down from 9 to 0:

```
#include "stm32f10x.h"
#include "IERG3810_LED.h"
#include "IERG3810_Buzzer.h"
#include "IERG3810_KEY.h"
#include "IERG3810_USART.h"
#include "IERG3810_Clock.h"
//#include "FONT.h"
#include "SevenSegments.h"

void IERG3810_clock_tree_init(void);
void IERG3810_USART2_init(u32, u32);
void IERG3810_USART1_init(u32, u32);
void Delay(u32);
void USART_print(u8, char *);

void Delay(u32 count){
  u32 i;
  for(i = 0; i < count; i++);
}

typedef struct{
  u16 LCD_REG;
  u16 LCD_RAM;
}LCD_TypeDef;



#define LCD_BASE       ((u32)(0x6C000000|       0x000007FE))
#define LCD                     ((LCD_TypeDef*)  LCD_BASE)

// Color
#define black                           (u16)       0
#define white                           (u16)       65535
#define green                           (u16)       12256
#define red                             (u16)       59554
#define blue                            (u16)       415
#define yellow                          (u16)       65504

void IERG3810_TFTLCD_WrReg(u16 regval){
  LCD->LCD_REG = regval;
}
void IERG3810_TFTLCD_WrData(u16 data){
  LCD->LCD_RAM = data;
}

void IERG3810_TFTLCD_SetParameter(void){
  IERG3810_TFTLCD_WrReg(0x01);
  IERG3810_TFTLCD_WrReg(0x11);

  IERG3810_TFTLCD_WrReg(0x3A);
  IERG3810_TFTLCD_WrData(0x55);
  IERG3810_TFTLCD_WrReg(0x29);

  IERG3810_TFTLCD_WrReg(0x36);
  IERG3810_TFTLCD_WrData(0xCA);
}


void IERG3810_TFTLCD_Init(void){
  RCC ->AHBENR |= 1 << 8;
  RCC ->APB2ENR |= 1 << 3;
  RCC ->APB2ENR |= 1 << 5;
  RCC ->APB2ENR |= 1 << 6;
  RCC ->APB2ENR |= 1 << 8;
  GPIOB ->CRL &= 0xFFFFFFF0;
  GPIOB ->CRL |= 0x00000003;

  //PORTD
  GPIOD ->CRH &= 0x00FFF000;
  GPIOD ->CRH |= 0xBB000BBB;
  GPIOD ->CRL &= 0xFF00FF00;
  GPIOD ->CRL |= 0x00BB00BB;

  //PORTE
  GPIOE ->CRH &= 0x00000000;
  GPIOE ->CRH |= 0xBBBBBBBB;
  GPIOE ->CRL &= 0x0FFFFFFF;
  GPIOE ->CRL |= 0xB0000000;

  //PORTG12
  GPIOG ->CRH &= 0xFFF0FFFF;
  GPIOG ->CRH |= 0x000B0000;
  GPIOG ->CRL &= 0xFFFFFFF0;
  GPIOG ->CRL |= 0x0000000B;


  FSMC_Bank1->BTCR[6] = 0x00000000;
  FSMC_Bank1->BTCR[7] = 0x00000000;
  FSMC_Bank1E ->BWTR[6] = 0x00000000;
  FSMC_Bank1 ->BTCR[6] |= 1 << 12;
  FSMC_Bank1 ->BTCR[6] |= 1 << 14;
  FSMC_Bank1 ->BTCR[6] |= 1 << 4;
  FSMC_Bank1 ->BTCR[7] |= 0 << 28;
  FSMC_Bank1 ->BTCR[7] |= 1 << 0;
  FSMC_Bank1 ->BTCR[7] |= 0xF << 8;
  FSMC_Bank1E ->BWTR[6] |= 0 << 28;
  FSMC_Bank1E ->BWTR[6] |= 0 << 0;
  FSMC_Bank1E ->BWTR[6] |= 3 << 8;
  FSMC_Bank1 ->BTCR[6] |= 1 << 0;

  IERG3810_TFTLCD_SetParameter();
  GPIOB ->ODR |= 1 << 0;
  //LCD_LIGHT_ON;
}

void IERG3810_TFTLCD_DrawDot(u16 x, u16 y, u16 color){

  IERG3810_TFTLCD_WrReg(0x2A);
  IERG3810_TFTLCD_WrData(x >> 8);
  IERG3810_TFTLCD_WrData(x & 0xFF);
  IERG3810_TFTLCD_WrData(0x01);
  IERG3810_TFTLCD_WrData(0x3F);
  IERG3810_TFTLCD_WrReg(0x2B);
  IERG3810_TFTLCD_WrData(y >> 8);
  IERG3810_TFTLCD_WrData(y & 0xFF);
  IERG3810_TFTLCD_WrData(0x01);
  IERG3810_TFTLCD_WrData(0xDF);
  IERG3810_TFTLCD_WrReg(0x2C);
  IERG3810_TFTLCD_WrData(color);
}

void IERG3810_TFTLCD_DrawLine(u16 x_St, u16 x_End, u16 y, u16 color){
  u16 x = 0;
  for(x = x_St; x < x_End; x++){
        IERG3810_TFTLCD_DrawDot(x,y,color);
  }
}

void IERG3810_TFTLCD_FillRectangle(u16 color, u16 start_x, u16
  length_x, u16 start_y, u16 length_y){
  u32 index = 0;
  IERG3810_TFTLCD_WrReg(0x2A);
  IERG3810_TFTLCD_WrData(start_x >> 8);
  IERG3810_TFTLCD_WrData(start_x & 0xFF);
  IERG3810_TFTLCD_WrData((start_x + length_x - 1) >> 8);
  IERG3810_TFTLCD_WrData((start_x + length_x - 1) & 0xFF);

  IERG3810_TFTLCD_WrReg(0x2B);
  IERG3810_TFTLCD_WrData(start_y >> 8);
  IERG3810_TFTLCD_WrData(start_y & 0xFF);
  IERG3810_TFTLCD_WrData((start_y + length_y - 1) >> 8);
  IERG3810_TFTLCD_WrData((start_y + length_y - 1) & 0xFF);

  IERG3810_TFTLCD_WrReg(0x2C);
  for(index = 0; index < length_x * length_y; index ++ ){
        IERG3810_TFTLCD_WrData(color);
  }
}

void IERG3810_TFTLCD_SevenSegment(u16 color, u16 start_x, u16
  start_y, u8 digit){

  //u16 color, u16 start_x, u16 length_x, u16 start_y, u16 length_y
  IERG3810_TFTLCD_FillRectangle(white,80, 80, 80,160);

  // offset
  start_x = start_x + 2.5;
  start_y = start_y + 10;
  if(Se_Seg[digit][0] == 0x1){ // a
          IERG3810_TFTLCD_FillRectangle(color, 10 + start_x, 55, 130 +
  start_y, 10);
  }
  if(Se_Seg[digit][1] == 0x1){ // b
          IERG3810_TFTLCD_FillRectangle(color,65 + start_x,10,75 +
  start_y,55);
  }
  if(Se_Seg[digit][2] == 0x1){
          IERG3810_TFTLCD_FillRectangle(color,65 + start_x,10,10 +
  start_y,55);
  }
  if(Se_Seg[digit][3] == 0x1){
          IERG3810_TFTLCD_FillRectangle(color,10 + start_x,55,0 +
  start_y,10);
  }
  if(Se_Seg[digit][4] == 0x1){
          IERG3810_TFTLCD_FillRectangle(color,0 + start_x,10,10 +
  start_y,55);
  }
  if(Se_Seg[digit][5] == 0x1){
          IERG3810_TFTLCD_FillRectangle(color,0 + start_x,10,75 +
  start_y,55);
  }
  if(Se_Seg[digit][6] == 0x1){
          IERG3810_TFTLCD_FillRectangle(color,10 + start_x,55,65 +
  start_y,10);
  }

}

void CountFrom9(void){

  u16 i= 9;
  while(1){

      IERG3810_TFTLCD_SevenSegment(red,80,80,i);
```

```
        Delay(10000000);
        IERG3810_TFTLCD_FillRectangle(white,80, 80, 80,160);
        if (i == 0){ // 0
                break;
        }
        i--;
  }

}

int main(void)
{
  IERG3810_LED_Init();
  IERG3810_TFTLCD_Init();

  Delay(2000000);

  //IERG3810_TFTLCD_SevenSegment(red, 80, 80, 8);

  CountFrom9();

}
```

Experiment 3.4

The SIDs of me and my group mate:



```
1155124983

1155144668
```

```
#include "stm32f10x.h"
#include "IERG3810_LED.h"
#include "IERG3810_Buzzer.h"
#include "IERG3810_KEY.h"
#include "IERG3810_USART.h"
#include "IERG3810_Clock.h"
#include "FONT.H"
#include "SevenSegments.h"

void IERG3810_clock_tree_init(void);
void IERG3810_USART2_init(u32, u32);
void IERG3810_USART1_init(u32, u32);
void Delay(u32);
void USART_print(u8, char *);

void Delay(u32 count){
	u32 i;
	for(i = 0; i < count; i++);
}

typedef struct{
	u16 LCD_REG;
	u16 LCD_RAM;
}	LCD_TypeDef;



#define LCD_BASE	((u32)(0x6C000000|	0x000007FE))
#define LCD			((LCD_TypeDef*)	LCD_BASE)

// Color
#define black				(u16)	0
#define white				(u16)	65535
#define green				(u16)	12256
#define red				(u16)	59554
#define blue				(u16)	415
#define yellow				(u16)	65504
```

```
void IERG3810_TFTLCD_WrReg(u16 regval){
        LCD->LCD_REG = regval;
}
void IERG3810_TFTLCD_WrData(u16 data){
        LCD->LCD_RAM = data;
}

void IERG3810_TFTLCD_SetParameter(void){
        IERG3810_TFTLCD_WrReg(0x01);
        IERG3810_TFTLCD_WrReg(0x11);

        IERG3810_TFTLCD_WrReg(0x3A);
        IERG3810_TFTLCD_WrData(0x55);

        IERG3810_TFTLCD_WrReg(0x29);

        IERG3810_TFTLCD_WrReg(0x36);
        IERG3810_TFTLCD_WrData(0xCA);
}


void IERG3810_TFTLCD_Init(void){
        RCC ->AHBENR |= 1 << 8;
        RCC ->APB2ENR |= 1 << 3;
        RCC ->APB2ENR |= 1 << 5;
        RCC ->APB2ENR |= 1 << 6;
        RCC ->APB2ENR |= 1 << 8;
        GPIOB ->CRL &= 0xFFFFFFF0;
        GPIOB ->CRL |= 0x00000003;

        //PORTD
        GPIOD ->CRH &= 0x00FFF000;
        GPIOD ->CRH |= 0xBB000BBB;
        GPIOD ->CRL &= 0xFF00FF00;
        GPIOD ->CRL |= 0x00BB00BB;

        //PORTE
```

```
    GPIOE ->CRH &= 0x00000000;
    GPIOE ->CRH |= 0xBBBBBBBB;
    GPIOE ->CRL &= 0x0FFFFFFF;
    GPIOE ->CRL |= 0xB0000000;

    //PORTG12
    GPIOG ->CRH &= 0xFFF0FFFF;
    GPIOG ->CRH |= 0x000B0000;
    GPIOG ->CRL &= 0xFFFFFFF0;
    GPIOG ->CRL |= 0x0000000B;



    FSMC_Bank1->BTCR[6] = 0x00000000;
    FSMC_Bank1->BTCR[7] = 0x00000000;
    FSMC_Bank1E ->BWTR[6] = 0x00000000;
    FSMC_Bank1 ->BTCR[6] |= 1 << 12;
    FSMC_Bank1 ->BTCR[6] |= 1 << 14;
    FSMC_Bank1 ->BTCR[6] |= 1 << 4;
    FSMC_Bank1 ->BTCR[7] |= 0 << 28;
    FSMC_Bank1 ->BTCR[7] |= 1 << 0;
    FSMC_Bank1 ->BTCR[7] |= 0xF << 8;
    FSMC_Bank1E ->BWTR[6] |= 0 << 28;
    FSMC_Bank1E ->BWTR[6] |= 0 << 0;
    FSMC_Bank1E ->BWTR[6] |= 3 << 8;
    FSMC_Bank1 ->BTCR[6] |= 1 << 0;

    IERG3810_TFTLCD_SetParameter();
    GPIOB ->ODR |= 1 << 0;
    //LCD_LIGHT_ON;
}

void IERG3810_TFTLCD_DrawDot(u16 x, u16 y, u16 color){

    IERG3810_TFTLCD_WrReg(0x2A);
    IERG3810_TFTLCD_WrData(x >> 8);
    IERG3810_TFTLCD_WrData(x & 0xFF);
    IERG3810_TFTLCD_WrData(0x01);
    IERG3810_TFTLCD_WrData(0x3F);
    IERG3810_TFTLCD_WrReg(0x2B);
    IERG3810_TFTLCD_WrData(y >> 8);
    IERG3810_TFTLCD_WrData(y & 0xFF);
    IERG3810_TFTLCD_WrData(0x01);
    IERG3810_TFTLCD_WrData(0xDF);
    IERG3810_TFTLCD_WrReg(0x2C);
    IERG3810_TFTLCD_WrData(color);
}

void IERG3810_TFTLCD_DrawLine(u16 x_St, u16 x_End, u16 y, u16 color){
    u16 x = 0;
    for(x = x_St; x < x_End; x++){
        IERG3810_TFTLCD_DrawDot(x,y,color);
    }
}

void IERG3810_TFTLCD_FillRectangle(u16 color, u16 start_x, u16 length_x, u16
start_y, u16 length_y){
    u32 index = 0;
    IERG3810_TFTLCD_WrReg(0x2A);
    IERG3810_TFTLCD_WrData(start_x >> 8);
    IERG3810_TFTLCD_WrData(start_x & 0xFF);
    IERG3810_TFTLCD_WrData((start_x + length_x - 1) >> 8);
    IERG3810_TFTLCD_WrData((start_x + length_x - 1) & 0xFF);

    IERG3810_TFTLCD_WrReg(0x2B);
    IERG3810_TFTLCD_WrData(start_y >> 8);
    IERG3810_TFTLCD_WrData(start_y & 0xFF);
    IERG3810_TFTLCD_WrData((start_y + length_y - 1) >> 8);
    IERG3810_TFTLCD_WrData((start_y + length_y - 1) & 0xFF);

    IERG3810_TFTLCD_WrReg(0x2C);
    for(index = 0; index < length_x * length_y; index ++ ){
        IERG3810_TFTLCD_WrData(color);
    }
}
```

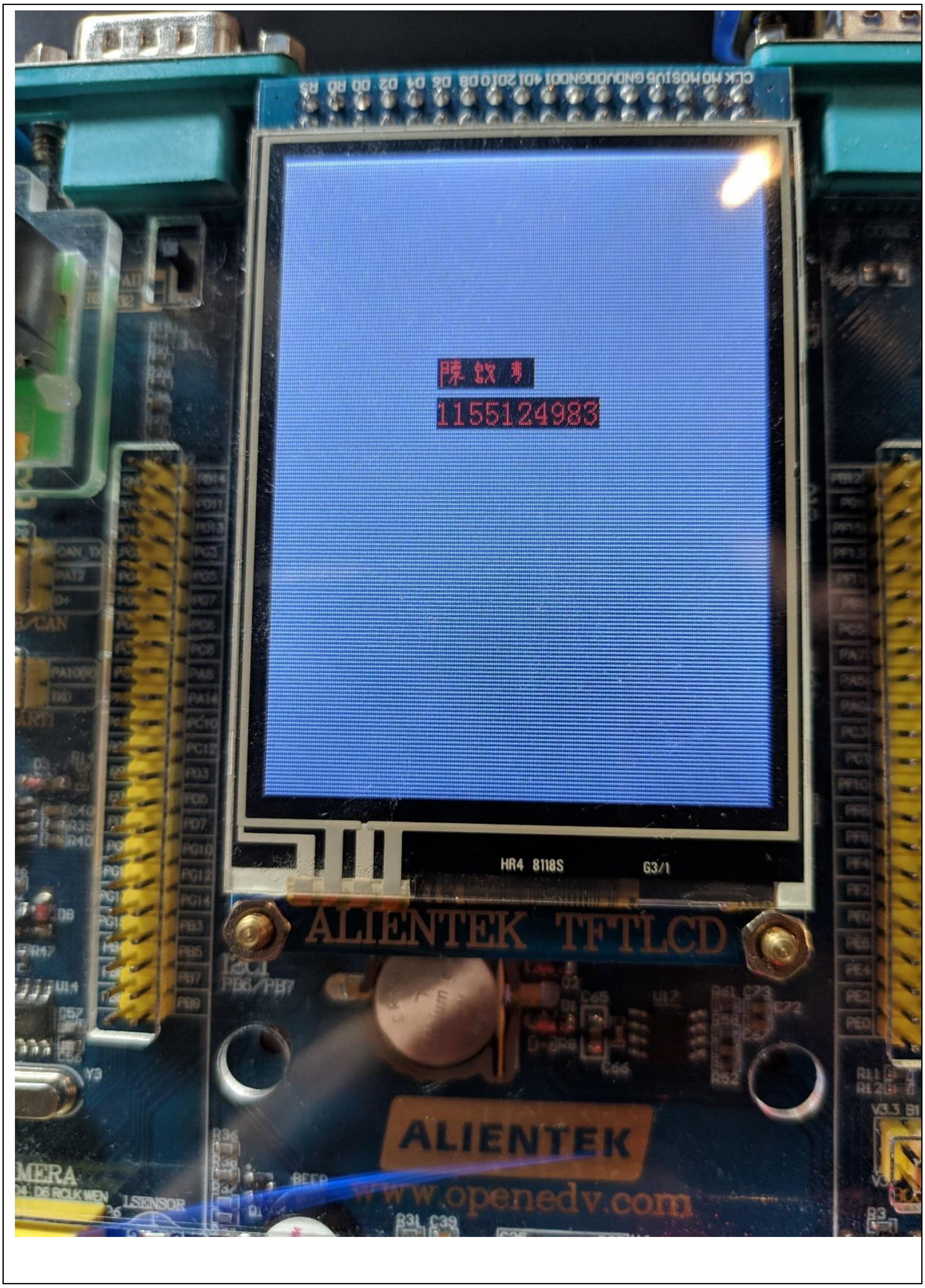
```
void IERG3810_TFTLCD_SevenSegment(u16 color, u16 start_x, u16 start_y, u8
digit){

        //u16 color, u16 start_x, u16 length_x, u16 start_y, u16 length_y
        IERG3810_TFTLCD_FillRectangle(white,80, 80, 80,160);

        // offset
        start_x = start_x + 2.5;
        start_y = start_y + 10;



        if(Se_Seg[digit][0] == 0x1){ // a
                IERG3810_TFTLCD_FillRectangle(color, 10 + start_x, 55, 130 +
start_y, 10);
        }
        if(Se_Seg[digit][1] == 0x1){ // b
                IERG3810_TFTLCD_FillRectangle(color,65 + start_x,10,75 +
start_y,55);
        }
        if(Se_Seg[digit][2] == 0x1){
                IERG3810_TFTLCD_FillRectangle(color,65 + start_x,10,10 +
start_y,55);
        }
        if(Se_Seg[digit][3] == 0x1){
                IERG3810_TFTLCD_FillRectangle(color,10 + start_x,55,0 +
start_y,10);
        }
        if(Se_Seg[digit][4] == 0x1){
                IERG3810_TFTLCD_FillRectangle(color,0 + start_x,10,10 +
start_y,55);
        }
        if(Se_Seg[digit][5] == 0x1){
                IERG3810_TFTLCD_FillRectangle(color,0 + start_x,10,75 +
start_y,55);
        }
        if(Se_Seg[digit][6] == 0x1){
                IERG3810_TFTLCD_FillRectangle(color,10 + start_x,55,65 +
start_y,10);
        }

}

void CountFrom9(void){

        u16 i= 9;
        while(1){

            IERG3810_TFTLCD_SevenSegment(red,80,80,i);
            Delay(10000000);
            IERG3810_TFTLCD_FillRectangle(white,80, 80, 80,160);
            if (i == 0){ // 0
                break;
            }
            i--;
        }

}

void IERG3810_TFTLCD_ShowChar(u16 x, u16 y, u8 ascii, u16 color, u16
bgcolor)
{
        u8 i,j;
        u8 index;
        u8 height = 16, length = 8;
        if(ascii < 32 || ascii >127) return;
        ascii -= 32;

        IERG3810_TFTLCD_WrReg(0x2A);
        IERG3810_TFTLCD_WrData(x >> 8);
        IERG3810_TFTLCD_WrData(x & 0xFF);
        IERG3810_TFTLCD_WrData((x + length - 1) >> 8);
        IERG3810_TFTLCD_WrData((x + length - 1) & 0xFF);
```

```
        IERG3810_TFTLCD_WrReg(0x2B);
        IERG3810_TFTLCD_WrData(y >> 8);
        IERG3810_TFTLCD_WrData(y & 0xFF);
        IERG3810_TFTLCD_WrData((y + height - 1) >> 8);
        IERG3810_TFTLCD_WrData((y + height - 1) & 0xFF);
        IERG3810_TFTLCD_WrReg(0x2C);

        for(j = 0; j < height /8; j++){
                for(i = 0; i < height /2; i++){
                        for(index = 0; index < length ; index ++){
                                if( (asc2_1608[ascii][index * 2 + 1 - j] >> i) & 0x01 ){
                                        IERG3810_TFTLCD_WrData(color);
                                }else{
                                        IERG3810_TFTLCD_WrData(bgcolor);
                                }
                        }
                }
        }
}

void IERG3810_ShowSID(char* str, u8 y, u16 color, u16 bgcolor)
{
        u8 x = 0;
        u8 i = 0;
        u8 x_inc = 8;
        while(str[i] !='\0'){
                //u8 send_ascii = str[i];
                IERG3810_TFTLCD_ShowChar(80+x_inc*i, y, (int) str[i], color,
bgcolor);
                i++;
        }
}



int main(void)
{
        IERG3810_LED_Init();
```

```
	IERG3810_TFTLCD_Init();

	Delay(2000000);

	//IERG3810_TFTLCD_SevenSegment(red, 80, 80, 8);
	//CountFrom9();

	IERG3810_ShowSID("1155124983", 180, red, black);
	IERG3810_ShowSID("1155144668", 130, red, black);

}
```

Experiment 3.5

Due to my groupmate is lacking behind. Here shows the result of mine:



```
#include "stm32f10x.h"
#include "IERG3810_LED.h"
#include "IERG3810_Buzzer.h"
#include "IERG3810_KEY.h"
#include "IERG3810_USART.h"
```

```
#include "IERG3810_Clock.h"
#include "FONT.H"
#include "SevenSegments.h"
#include "CFONT.H"

void IERG3810_clock_tree_init(void);
void IERG3810_USART2_init(u32, u32);
void IERG3810_USART1_init(u32, u32);
void Delay(u32);
void USART_print(u8, char *);

void Delay(u32 count){
        u32 i;
        for(i = 0; i < count; i++);
}

typedef struct{
        u16 LCD_REG;
        u16 LCD_RAM;
}       LCD_TypeDef;



#define LCD_BASE        ((u32)(0x6C000000|        0x000007FE))
#define LCD                     ((LCD_TypeDef*)  LCD_BASE)

// Color
#define black                           (u16)       0
#define white                           (u16)       65535
#define green                           (u16)       12256
#define red                             (u16)       59554
#define blue                            (u16)       415
#define yellow                          (u16)       65504

void IERG3810_TFTLCD_WrReg(u16 regval){
        LCD->LCD_REG = regval;
}
void IERG3810_TFTLCD_WrData(u16 data){
        LCD->LCD_RAM = data;
}

void IERG3810_TFTLCD_SetParameter(void){
        IERG3810_TFTLCD_WrReg(0x01);
        IERG3810_TFTLCD_WrReg(0x11);

        IERG3810_TFTLCD_WrReg(0x3A);
        IERG3810_TFTLCD_WrData(0x55);

        IERG3810_TFTLCD_WrReg(0x29);

        IERG3810_TFTLCD_WrReg(0x36);
        IERG3810_TFTLCD_WrData(0xCA);
}



void IERG3810_TFTLCD_Init(void){
        RCC ->AHBENR |= 1 << 8;
        RCC ->APB2ENR |= 1 << 3;
        RCC ->APB2ENR |= 1 << 5;
        RCC ->APB2ENR |= 1 << 6;
        RCC ->APB2ENR |= 1 << 8;
        GPIOB ->CRL &= 0xFFFFFFF0;
        GPIOB ->CRL |= 0x00000003;

        //PORTD
        GPIOD ->CRH &= 0x00FFF000;
        GPIOD ->CRH |= 0xBB000BBB;
        GPIOD ->CRL &= 0xFF00FF00;
        GPIOD ->CRL |= 0x00BB00BB;

        //PORTE
        GPIOE ->CRH &= 0x00000000;
        GPIOE ->CRH |= 0xBBBBBBBB;
        GPIOE ->CRL &= 0x0FFFFFFF;
        GPIOE ->CRL |= 0xB0000000;
```

```
	//PORTG12
	GPIOG ->CRH &= 0xFFF0FFFF;
	GPIOG ->CRH |= 0x000B0000;
	GPIOG ->CRL &= 0xFFFFFFF0;
	GPIOG ->CRL |= 0x0000000B;



	FSMC_Bank1->BTCR[6] = 0x00000000;
	FSMC_Bank1->BTCR[7] = 0x00000000;
	FSMC_Bank1E ->BWTR[6] = 0x00000000;
	FSMC_Bank1 ->BTCR[6] |= 1 << 12;
	FSMC_Bank1 ->BTCR[6] |= 1 << 14;
	FSMC_Bank1 ->BTCR[6] |= 1 << 4;
	FSMC_Bank1 ->BTCR[7] |= 0 << 28;
	FSMC_Bank1 ->BTCR[7] |= 1 << 0;
	FSMC_Bank1 ->BTCR[7] |= 0xF << 8;
	FSMC_Bank1E ->BWTR[6] |= 0 << 28;
	FSMC_Bank1E ->BWTR[6] |= 0 << 0;
	FSMC_Bank1E ->BWTR[6] |= 3 << 8;
	FSMC_Bank1 ->BTCR[6] |= 1 << 0;

	IERG3810_TFTLCD_SetParameter();
	GPIOB ->ODR |= 1 << 0;
	//LCD_LIGHT_ON;
}

void IERG3810_TFTLCD_DrawDot(u16 x, u16 y, u16 color){

	IERG3810_TFTLCD_WrReg(0x2A);
	IERG3810_TFTLCD_WrData(x >> 8);
	IERG3810_TFTLCD_WrData(x & 0xFF);
	IERG3810_TFTLCD_WrData(0x01);
	IERG3810_TFTLCD_WrData(0x3F);
	IERG3810_TFTLCD_WrReg(0x2B);
	IERG3810_TFTLCD_WrData(y >> 8);
	IERG3810_TFTLCD_WrData(y & 0xFF);
```

```
    IERG3810_TFTLCD_WrData(0x01);
    IERG3810_TFTLCD_WrData(0xDF);
    IERG3810_TFTLCD_WrReg(0x2C);
    IERG3810_TFTLCD_WrData(color);
}

void IERG3810_TFTLCD_DrawLine(u16 x_St, u16 x_End, u16 y, u16 color){
    u16 x = 0;
    for(x = x_St; x < x_End; x++){
        IERG3810_TFTLCD_DrawDot(x,y,color);
    }
}

void IERG3810_TFTLCD_FillRectangle(u16 color, u16 start_x, u16 length_x, u16 start_y, u16 length_y){
    u32 index = 0;
    IERG3810_TFTLCD_WrReg(0x2A);
    IERG3810_TFTLCD_WrData(start_x >> 8);
    IERG3810_TFTLCD_WrData(start_x & 0xFF);
    IERG3810_TFTLCD_WrData((start_x + length_x - 1) >> 8);
    IERG3810_TFTLCD_WrData((start_x + length_x - 1) & 0xFF);

    IERG3810_TFTLCD_WrReg(0x2B);
    IERG3810_TFTLCD_WrData(start_y >> 8);
    IERG3810_TFTLCD_WrData(start_y & 0xFF);
    IERG3810_TFTLCD_WrData((start_y + length_y - 1) >> 8);
    IERG3810_TFTLCD_WrData((start_y + length_y - 1) & 0xFF);

    IERG3810_TFTLCD_WrReg(0x2C);
    for(index = 0; index < length_x * length_y; index ++ ){
        IERG3810_TFTLCD_WrData(color);
    }
}

void IERG3810_TFTLCD_SevenSegment(u16 color, u16 start_x, u16 start_y, u8 digit){
    //u16 color, u16 start_x, u16 length_x, u16 start_y, u16 length_y
    IERG3810_TFTLCD_FillRectangle(white,80, 80, 80,160);

    // offset
    start_x = start_x + 2.5;
    start_y = start_y + 10;



    if(Se_Seg[digit][0] == 0x1){ // a
            IERG3810_TFTLCD_FillRectangle(color, 10 + start_x, 55, 130 +
start_y, 10);
    }
    if(Se_Seg[digit][1] == 0x1){ // b
            IERG3810_TFTLCD_FillRectangle(color,65 + start_x,10,75 +
start_y,55);
    }
    if(Se_Seg[digit][2] == 0x1){
            IERG3810_TFTLCD_FillRectangle(color,65 + start_x,10,10 +
start_y,55);
    }
    if(Se_Seg[digit][3] == 0x1){
            IERG3810_TFTLCD_FillRectangle(color,10 + start_x,55,0 +
start_y,10);
    }
    if(Se_Seg[digit][4] == 0x1){
            IERG3810_TFTLCD_FillRectangle(color,0 + start_x,10,10 +
start_y,55);
    }
    if(Se_Seg[digit][5] == 0x1){
            IERG3810_TFTLCD_FillRectangle(color,0 + start_x,10,75 +
start_y,55);
    }
    if(Se_Seg[digit][6] == 0x1){
            IERG3810_TFTLCD_FillRectangle(color,10 + start_x,55,65 +
start_y,10);
    }
```

```
}

void CountFrom9(void){

    u16 i= 9;
    while(1){

        IERG3810_TFTLCD_SevenSegment(red,80,80,i);
        Delay(10000000);
        IERG3810_TFTLCD_FillRectangle(white,80, 80, 80,160);
        if (i == 0){ // 0
            break;
        }
        i--;
    }

}

void IERG3810_TFTLCD_ShowChar(u16 x, u16 y, u8 ascii, u16 color, u16
bgcolor)
{
    u8 i,j;
    u8 index;
    u8 height = 16, length = 8;
    if(ascii < 32 || ascii >127) return;
    ascii -= 32;

    IERG3810_TFTLCD_WrReg(0x2A);
    IERG3810_TFTLCD_WrData(x >> 8);
    IERG3810_TFTLCD_WrData(x & 0xFF);
    IERG3810_TFTLCD_WrData((x + length - 1) >> 8);
    IERG3810_TFTLCD_WrData((x + length - 1) & 0xFF);

    IERG3810_TFTLCD_WrReg(0x2B);
    IERG3810_TFTLCD_WrData(y >> 8);
    IERG3810_TFTLCD_WrData(y & 0xFF);
    IERG3810_TFTLCD_WrData((y + height - 1) >> 8);
    IERG3810_TFTLCD_WrData((y + height - 1) & 0xFF);
    IERG3810_TFTLCD_WrReg(0x2C);

    for(j = 0; j < height /8; j++){
        for(i = 0; i < height /2; i++){
            for(index = 0; index < length ; index ++){
                if( (asc2_1608[ascii][index * 2 + 1 - j] >> i) & 0x01 ){
                    IERG3810_TFTLCD_WrData(color);
                }else{
                    IERG3810_TFTLCD_WrData(bgcolor);
                }
            }
        }
    }
}

void IERG3810_TFTLCD_ShowChinChar(u16 x, u16 y, u8 ind_char, u16 color,
u16 bgcolor)
{
    u8 i,j;
    u8 index;
    u8 height = 16, length = 16;

    IERG3810_TFTLCD_WrReg(0x2A);
    IERG3810_TFTLCD_WrData(x >> 8);
    IERG3810_TFTLCD_WrData(x & 0xFF);
    IERG3810_TFTLCD_WrData((x + length - 1) >> 8);
    IERG3810_TFTLCD_WrData((x + length - 1) & 0xFF);

    IERG3810_TFTLCD_WrReg(0x2B);
    IERG3810_TFTLCD_WrData(y >> 8);
    IERG3810_TFTLCD_WrData(y & 0xFF);
    IERG3810_TFTLCD_WrData((y + height - 1) >> 8);
    IERG3810_TFTLCD_WrData((y + height - 1) & 0xFF);
    IERG3810_TFTLCD_WrReg(0x2C);

    for(j = 0; j < height /8; j++){
        for(i = 0; i < height /2; i++){
            for(index = 0; index < length ; index ++){
                if( (chi_1616[ind_char][index * 2 + 1 - j] >> i) & 0x01 ){
                    IERG3810_TFTLCD_WrData(color);
                }else{
                    IERG3810_TFTLCD_WrData(bgcolor);
                }
            }
        }
    }
}

void IERG3810_ShowSID(char* str, u8 y, u16 color, u16 bgcolor)
{
    u8 x = 0;
    u8 i = 0;
    u8 x_inc = 8;
    while(str[i] !='\0'){
        //u8 send_ascii = str[i];
        IERG3810_TFTLCD_ShowChar(80+x_inc*i, y, (int) str[i], color,
bgcolor);
        i++;
    }
}

void IERG3810_ShowName(u8 y, u16 color, u16 bgcolor)
{
    u8 x = 0;
    u8 i = 0;
    u8 x_inc = 16;

    for(i = 0; i < 3; i++){
    IERG3810_TFTLCD_ShowChinChar(80+x_inc*i, y,i,color,bgcolor);
    }
}

int main(void)
```

```
{
	IERG3810_LED_Init();
	IERG3810_TFTLCD_Init();

	Delay(2000000);

	//IERG3810_TFTLCD_SevenSegment(red, 80, 80, 8);
	//CountFrom9();

	IERG3810_ShowSID("1155124983", 180, red, black);
	//IERG3810_ShowSID("1155144668", 130, red, black);
	IERG3810_ShowName(200,red, black);

}
```

III. DISCUSSION

Exp 3.1

<you shall know the coordination of the origin (0, 0) of the screen. Where is it?>
Top left.

IV. SUMMARY

We learn displaying alphabet and Chinese characters on TFT-LCD

V. DIVISION OF WORK

<Lab work: Jesse & Derek, Report writing: Jesse>

VI. REFERENCES